

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Л. С. Ямпольський
О. І. Лісовиченко
В. В. Олійник

НЕЙРОТЕХНОЛОГІЇ ТА НЕЙРОКОМП’ЮТЕРНІ СИСТЕМИ

Затверджено
Вченою радою Національного технічного уні-
верситету України “КПІ” як підручник для
студентів, які навчаються за спеціальністю
“Автоматизація та комп’ютерно-інтегровані
технології”

Київ
Дорадо-Друк
2016

УДК 004.8
ББК 32.813
Я 57

Рецензенти:

В. М. Казак, д-р техн. наук, проф.

С. І. Осадчий, д-р техн. наук, проф.

П. І. Бідюк, д-р техн. наук, проф.

Гриф наданий Вченою радою НТУУ “КПІ” (протокол № 10 від 07.12.2015)

Авторські права на твір зареєстровані Державною Службою інтелектуальної власності України (Свідоцтво № 58776 від 24.02.2015)

Ямпольський Л. С., Лісовиченко О. І., Олійник В. В.
Я 57 Нейротехнології та нейрокомп’ютерні системи: підручник / Л. С. Ямпольський, О. І. Лісовиченко, В. В. Олійник. – К.: «Дорадо-Друк», 2016. – 576 с.: іл. – Бібліогр.: с. 537-551.
ISBN 978-966-2077-69-8

Підручник є продовженням циклу розробленої авторами вперше в Україні навчальної літератури нового покоління з широким впровадженням сучасних інформаційних технологій в електронній версії, виконаній у PDF-форматі, призначеній для Internet-розповсюдження, оснащеної тестуючими та пошуковими системами і лабораторними практикумами і супроводжуваній мультимедійною анімацією складних для сприйняття процесів функціонування нейронних моделей прикладних технологічних та організаційно-технічних систем.

Для студентів, викладачів, аспірантів і науковців, які займаються проблематикою нейротехнологій та їх прикладними аспектами.

УДК 004.8
ББК 32.813

ISBN 978-966-2077-69-8

© Л. С. Ямпольський, 2016

© О. І. Лісовиченко, 2016

© В. В. Олійник, 2016

Шановний читачу!

Основні засади, на яких формувався мультимедійний підручник “Нейро-технології та нейрокомп’ютерні системи”, були запозичені з розроблених за участю авторів попередніх праць, які увійшли в мультимедійний комп’ютерно-інтегрований комплекс засобів дистанційно-віртуального навчання (МКІЗ ДВН) для ряду навчальних дисциплін. Отже, підручник, який Ви тримаєте в руках, є продовженням циклу розробленої вперше в Україні навчальної літератури нового покоління, започаткованої авторами з 2004 року під загальною назвою:

МУЛЬТИМЕДІЙНИЙ КОМП’ЮТЕРНО-ІНТЕГРОВАНІЙ КОМПЛЕКС ЗАСОБІВ ДИСТАНЦІЙНО-ВІРТУАЛЬНОГО НАВЧАННЯ З ВИКОРИСТАННЯМ ІНТЕРНЕТ-ТЕХНОЛОГІЙ

Ямпольський Л. С., Мельничук П. П., Самотокін Б. Б.
Лісовиченко О. І., Остапченко К. Б., Поліщук М. М.,
Ткач М. М., Ткач Б. П.

Характеристика проблеми та мета роботи

Протягом 2004–2011 років авторами був підготовлений і виданий “Мультимедійний КІКЗ ДВН”, який складається з 6 підручників [72, 73, 78] з грифом МОН України, укомплектованих мультимедійними електронними версіями та захищених Державним департаментом інтелектуальної власності МОН України та Державною службою інтелектуальної власності України як багатофункціональні комп’ютерно-інтегровані засоби дистанційно-віртуального навчання з використанням інтернет-технологій.

Мета роботи полягала у створенні на рівні світових стандартів цілісних КІКЗ ДВН з використанням INTERNET-ресурсів для забезпечення користувача дружнім інтерфейсом і багатофункціональним середовищем навчально-методично-наукової інформації про певну проблемну галузь. *Дружній інтерфейс* підвищує інтерес користувача до використання МКІКЗ ДВН у процесі навчання за рахунок впровадження мультимедійних та інтернет-технологій, а також удосконаленого середовища подання матеріалу і пошуку інформації. *Багатофункціональне середовище* дає можливість вивчати теоретичний матеріал, виконувати лабораторно-дослідні практикуми і проектно-розрахункові роботи, здійснювати самоконтроль знань користувача та атестацію з боку викладача.

Реалізація МКІКЗ ДВН у такому обсязі можливостей вимагала створення *семантично узгоджених інструментальних засобів*, на основі яких функціонують і синхронізуються окремі складові “Мультимедійного КІКЗ ДВН”. Можна констатувати, що як за повнотою реалізації і комплексного охоплення складових навчального процесу, так і за науково-методичним рівнем представлення подібних багатофункціональних аналогів в Україні немає.

У “Мультимедійному КІКЗ ДВН” втілено багаторічний досвід науково-педагогічної діяльності авторів зі створення сучасної системи науково-методичного забезпечення зв’язаних логічною схемою викладання ключових навчальних дисциплін програм підготовки фахівців, які навчаються за напрямками: “Комп’ютерні науки”, “Прикладна математика”, “Інформатика”, “Системна інженерія”, “Автоматизація та комп’ютерно-інтегровані технології”, “Менеджмент”. Вищерозглянута робота уявляє собою єдиний комплекс навчальної літератури для бакалаврів, спеціалістів і магістрів відповідних напрямків підготовки і надає можливість формувати останню як неперервний взаємопов’язаний процес поширення і поглиблення певної галузі знань за єдиним пер-

сональним автоматизованим учбовим місцем студента – ПАУМС. При цьому стає можливим використовувати як сукупності окремих розділів для профілювання студента, так і їх елементи для загальноосвітньої базової підготовки фахівців. Окремо необхідно відзначити проблемно-орієнтований аспект спрямованості МКІКЗ ДВН: при його створенні в основу була покладена реалізація за окремими напрямками знань, об'єднаними навчальним планом у відповідні напрями підготовки студентів (наприклад, проектно-технологічний, схемо- та системотехнічний тощо). Все це підтверджує універсальний характер підходу, покладеного авторами роботи в розробку “Мультимедійного КІКЗ ДВН”.

Реалізація концептуальної моделі КІКЗ ДВН

Структура мультимедійного середовища МКІКЗ ДВН – ПАУМС є загальною для всіх мультимедійних підручників, виданих за участю авторів, і наведена на рис. 2П (див. стор. 6 даної книги).

Семантична узгодженість

Ø **Означення 2МК.** *Семантика об'єкта у мультимедійному середовищі МКІКЗ ДВН (його семантична характеристика)* – це частина інформації про об'єкт, якою подаються властивості навчаючого і відображаючого об'єкта. Зокрема, *семантика топоніма* – змістовне значення назви навчального або відображаючого об'єкта. *Цифровий опис об'єкта* – формалізоване уявлення в цифровому вигляді даних про об'єкт, яке містить рубрикацію й позначення складових навчальних і відображуючих об'єктів, їх семантику і відношення об'єктів у відповідності з *класифікатором*. Перелік семантичних характеристик задається *класифікатором об'єктів* мультимедійного середовища МКІКЗ ДВН.

Ø **Означення 3МК.** *Семантична узгодженість об'єктів* мультимедійного середовища МКІКЗ ДВН – це співпадання або логічна відповідність *семантик* пов'язаних навчаючих і відображуючих об'єктів.

Загальні властивості мультимедійного комплексу

Ø *реалізація звичайних* (паперових) та *мультимедійних* (на дисковому носії) варіантів підручників з певних галузей знань;

Ø *використання сучасних технологій* подання навчального матеріалу у HTML-, PDF- форматах, придатних до Інтернет-розповсюдження;

Ø *використання мультимедійних засобів* (2D анімація – Flash, GIF та 3D анімація – 3D Max Studio, Maya, 3D Canvas) подання графічного матеріалу;

Ø *запровадження інтерактивного інтерфейсу* в процесі дистанційно-віртуального навчання;

Ø *організація ефективного контекстного пошуку* під час роботи з матеріалом на дисковому носії;

Ø *інтеграція лабораторно-дослідних практикумів* у середовище КІКЗ ДВН;

Ø *впровадження самоконтролю остаточних знань* студента (користувача) з можливістю використання процедур покращення рівня підготовленості;

Ø *автоматизоване атестаційне оцінювання знань* студента (користувача) викладачем, в тому числі за класичною та за 100-бальною системами;

Ø *придатність (адаптованість)* до застосування в on-line (потоківих) технологіях, а саме – мультимедійні комп'ютерно-інтегровані засоби ДВН, що містить комплекс, можуть бути відтворені в режимі розширеної інтернет-трансляції навчального матеріалу.

Властивості мультимедії в комплексах підручників

1. На відміну від домінуючого в Україні використання *лінійного мультимедійного методу навчання* без впливу слухачів на якість процесу подання і засвоєння навчального матеріалу, запропонований “Мультимедійний КІКЗ ДВН ” як *інтерактивна навчальна система, заснована на нелінійному методі мультимедійної технології - гіпермедії*, забезпечує одночасну зі змістовим текстовим навчальним матеріалом роботу анімаційної комп’ютерної графіки.

2. Впроваджений в навчальний процес метод *медіа-конвергенції*, що домінує в підручниках як *механізм поєднання (семантичної узгодженості) на одній платформі різних форм просторового сприйняття відеоінформації*.

За своїми особливостями створені мультимедії поділяються на форми:

1. За принципом дії

Ø із статичним відтворенням пояснень, характерних рис, особливостей досліджуваного об’єкта;

Ø з динамічним відтворенням зміни станів, послідовності дії досліджуваного об’єкта;

Ø комбіновані із статично-динамічним відтворенням як пояснень, характерних рис, так і відтворення зміни станів досліджуваного об’єкта;

Ø з інтелектуалізованою інтерактивною взаємодією та емуляцією реалізації досліджуваного об’єкта у дослідних практикумах.

2. За об’єктом дослідження

Ø подання складних динамічних систем;

Ø відтворення математичного апарату;

Ø подання математичних моделей процесів;

Ø відтворення методів і алгоритмів;

Ø відтворення схем, часових діаграм.

Особливості тестуючих режимів

Використовуються алгоритми проходження тестів у трьох режимах тестування: *навчаючому, закріплюючому (або фіксуєчому) та атестаційному*:

Ø **Означення 4МК. Навчаючий** – режим, за яким користувач на підставі загального ознайомлення з теоретичним розділом (поданий у вигляді “теорії” у функції «Help») відповідає на ключові запитання «Тестуючого меню»;

Ø **Означення 5МК. Закріплюючий (фіксуєчий)** – режим, за яким користувачу пропонуються поглиблені тестові набори з можливостями: *використання спроб виправлення відповідей* у процесі поточного проходження тестування за його результатами; *покращення при бажанні користувача показників тестування* після його проходження шляхом повторного ініціювання тих фрагментів тесту, на які дано невірні відповіді. В останньому випадку функцією «Help» для користувача забезпечується виклик фрагментів теорії, які відповідають контексту тесту з невірною відповіддю;

Ø **Означення 6МК. Атестаційний** - режим, відповідний захисту користувачем теоретичного курсу викладачу. Для користувача це остаточна перевірка рівня засвоєння навчального матеріалу, коли підказки і донавчання неможливі, а сформований системою тестування (СТ) результат використовується викладачем для атестації.

Формування тестів

Для будь-якого режиму тестування тести, в залежності від характеру

теоретичного матеріалу, засвоєння якого користувачем перевіряється СТ, різняться за *формою подання* і *принципом формування* тестуючого запиту.

Як *форми представлення тестів* використовуються:

Ø *прості* (відповіді “Так” або “Ні” на запитання) – при реалізації навчального режиму тестування (головним чином у [72]);

Ø *ускладнені* (відповіді *N* з *M* запитань) з підвищеною бальністю кожної вірної відповіді такої форми – при реалізації закріплюючого (фіксуємого) режиму тестування (переважно у [73] та [78]);

Ø *з інтерактивним пересуванням запропонованих відповідей* в порядку, який визначає вірну послідовність виконання певних дій щодо досліджуваного об’єкту (впроваджено у [73] та [78]).

За *принципом формування тестуючого запиту*:

Ø *стандартні фразеологічні* – надаються або вимагаються до відповіді вислови природною мовою;

Ø *математичного походження* – надаються або вимагаються до відповіді математичні залежності;

Ø *графічного походження* – у вигляді фрагментів функціональних схем, алгоритмів, конструктивів об’єктів тестування (як при формуванні запитів, так і у відповідях при тестуванні).

Реалізація лабораторно-дослідних практикумів

Концептуальні форми організації подання і проведення *лабораторно-дослідних практикумів* (ЛДП):

1. ЛДП з *локально інтегрованими розрахунково-проектними роботами*, яким обмеженою мірою передували або супроводжували прості СТ (переважно у [72]).

2. ЛДП із *складними дослідними комплексами* з поданням поглиблених теоретичних відомостей і передуючою СТ (як правило, із закріплюючим режимом тестування) і можливістю проведення всебічних досліджень об’єкта за тематикою відповідного розділу підручника (характерне для [73]).

3. Найпотужніші *лабораторно-дослідні стенди* для моделювання дослідження процесів у складних організаційно-технічних, технологічних, логістичних системах (реалізовано у [78]). Стенди укомплектовано як поглибленим теоретичним матеріалом, презентаціями з мультимедійним відтворенням динамічних процесів у складних об’єктах, СТ із закріплюючим та атестаційним режимами тестування, так і емуляторами ЛДП для проведення досліджень об’єктів різної природи, в тому числі, із застосуванням засад штучного інтелекту.

Впровадження результатів роботи

Мультимедійний комплекс [72] демонструвався у Верховній Раді України під час слухань з питань освіти і науки в травні 2004 року.

Складові “Мультимедійного КІКЗ ДВН ” [72, 73, 78] впроваджено у навчальні процеси в: НТУ України “КПІ”, Житомирському державному технологічному університеті, Міжрегіональній академії управління персоналом (м. Київ), а також передані для впровадження в Технічний університет Софії (Болгарія), Технічний університет м. Клермонта (Франція), Ліберецький технічний університет (Чеська республіка), Саратовський та Новосибірський державні технічні університети (Росія) в межах виконання спільного проекту TEMPUS (Подвійний магістерський ступінь з автоматизації/мехатроніки країн ЄС – країн партнерів: номер проекту 517138- TEMPUS-1-2011-1-CZ- TEMPUS -JPCR).

Автори вдячні:

*створювачам теорії нейротехнології і розробникам нейронних сіток –
за теоретичний і прикладний матеріал до підручника,
рецензентам – за критику і слушні зауваження
щодо його структури і змісту,
студентам і аспірантам – за увагу під час навчання і допомогу
при підготовці підручника до видання,
нашим дружинам – за терпіння і розуміння*

*Ні доби без рядка
Пліній Старший Гай*

ПЕРЕДМОВА ЗАГАЛЬНА

*Якщо бути, то бути першим
Валерій Чкалов*

*Те, що ви робите в житті, повинно бути тим,
що ви любите,
і треба любити те, чим ви займаєтесь
Рей Дуглас Бредбері*

Актуальність створення гнучких комп'ютерно-інтегрованих комплексів навчання. Глобалізація процесів, які відбуваються у світі, не обминула й освітянську галузь, що проявилось в кількісному і якісному зростанні знання-орієнтованих інформаційних потоків, які мають бути опановані студентами. Тому з урахуванням перспектив наукового та технологічного розвитку України важливе місце в діяльності вітчизняних вищих навчальних закладів посідає удосконалення навчального процесу, реформування програм підготовки та перепідготовки спеціалістів, що ускладнюється суперечністю між зростаючим обсягом нової навчально-наукової інформації та обмеженими термінами навчання. Ці протиріччя характеризують зміст соціального замовлення на створення нових форм і методів ефективних технологій навчання у поєднанні з посиленням самостійної навчальної і наукової роботи студентів.

Все це складає *необхідні умови* для розробки і впровадження нових відповідних методів і форм проведення навчального процесу з метою його інтенсифікації. Постійне підвищення якості освіти, осучаснення її змісту та форм організації навчально-виховного процесу належать до важливих напрямів державної політики, визначених “Національною доктриною розвитку освіти”.

З іншого боку, саме глобалізація обумовила й створення *достатніх умов* для модернізації навчального процесу шляхом впровадження в освіту інформаційно-комунікаційних технологій, що сприяє виникненню нових засобів, методів і форм навчання на базі комп'ютерних технологій (рис. 1П).

Загальноприйняті форми навчання вимагають присутності лектора (учителя) і не надають необхідної мобільності процесу навчання. Крім того, учні, які мають фізичні вади, а також ті, хто з інших різних причин не може регулярно відвідувати аудиторні заняття і хоче опанувати певні дисципліни або їх розділи самостійно (екстерном), потребують докорінних змін у формах і методах організації навчального процесу з наданням можливості використання *дистанційно-віртуального навчання*. Це викликало нагальну необхідність у створенні сучасних *мультимедійних комплексів комп'ютерно-інтегрованих засобів (КІКЗ) дистанційно-віртуального навчання (ДВН)* [72; 73; 78] з використанням комп'ютерних технологій відображення і подання інформації на усіх етапах навчального процесу.



Рис. 1П. Необхідні та достатні умови впровадження мультимедійних комплексів

Означення ПЗ.1. *Дистанційно-віртуальне навчання* — це навчання, яке виконується з використанням технологічних режимів клієнт-сервер (на віддалених ресурсах Internet-середовища) і клієнт-клієнт (на необхідних ресурсах клієнтської машини) та *організоване* таким чином, що користувач може розглядати КІКЗ ДВН як єдине велике операційне середовище, що суттєво спрощує процедуру проходження етапів навчання.

На сьогодні майже не існує, окрім наведених вище, розроблених КІКЗ ДВН, які б задовольняли вимоги подання матеріалу з урахуванням індивідуальних особливостей студентів у вивченні матеріалу, наявності систем контролю знань з метою підвищення креативного мислення, а також систем розширеного пошуку інформації з можливістю її отримання не тільки за ключовими словами.

Таким чином, *метою створення* КІКЗ ДВН було забезпечення користувача дружнім інтерфейсом і багатофункціональним середовищем навчально-методично-наукової інформації про певну проблемну галузь. Дружній інтерфейс підвищує інтерес користувача до використання КІКЗ ДВН у процесі навчання за рахунок впровадження мультимедійних та інтернет-технологій, а також удосконаленого середовища подання матеріалу і пошуку інформації. Багатофункціональне середовище дає можливості вивчення теоретичного матеріалу, виконання лабораторно-дослідницьких практикумів ієрархічного за структурою та пріоритетністю матеріалу і проектно-розрахункових робіт, самоконтролю користувача та атестацію з боку викладача.

Реалізація КІКЗ ДВН в такому обсязі можливостей потребує семантичноузгоджених інструментальних засобів, на основі яких функціонують та синхронізуються окремі складові частини.

Постановка питання. У створених мультимедійних КІКЗ ДВН мультимедійне середовище (рис. 2П) структурується певними наборами *навчально-науково-методичних об'єктів* навчального процесу (лекційний матеріал, тестові завдання, лабораторні практикуми, глосарії, джерела знань, презентації тощо), з одного боку, і *мультимедійних засобів комп'ютерних технологій відображення і подання інформації*, на основі яких функціонують і синхронізуються окремі складові мультимедійного комплексу на усіх етапах процесу навчання, з другого боку.

Можливість використання створеного мультимедійного середовища в якості оболонки для широкого впровадження в навчальний процес усіх видів освіти (дошкільну і позашкільну, загально-середню, професійно-технічну, і звичайно, вищу та післядипломну – перепідготовки і підвищення кваліфікації) пов'язана з виконанням необхідної умови *семантичної узгодженості* інтегровуваних у мультимедійне середовище КІКЗ. Це вимагає, в першу чергу, чіткого визначення поняття семантичної узгодженості в аспекті застосування його до означеної проблеми.

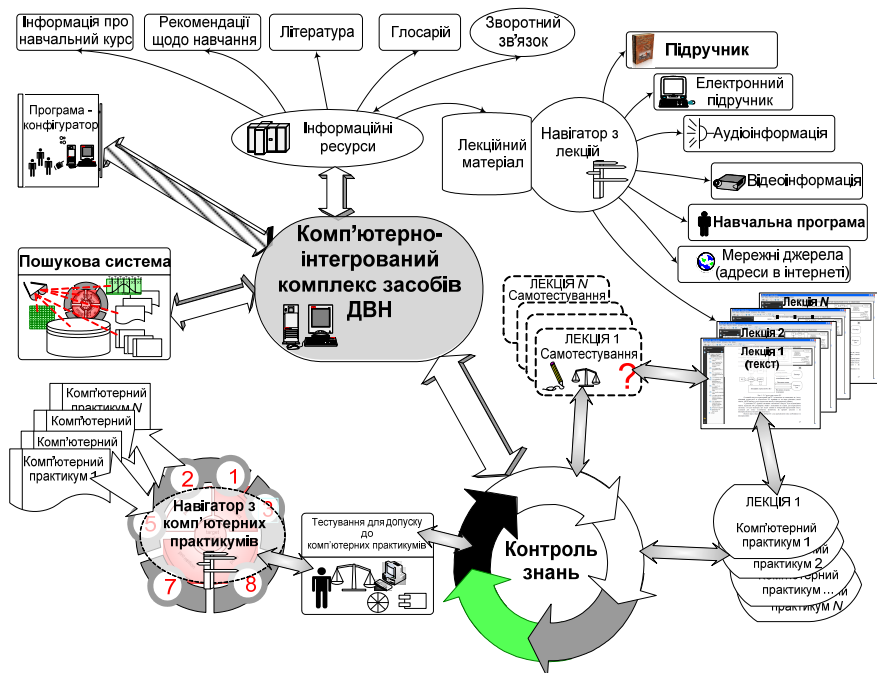


Рис. 2П. Структура мультимедійного середовища КІКЗ ДВН

Означення семантичної узгодженості КІКЗ ДВН. Всі засоби, які інтегруються в наведене на рис. 2П мультимедійне середовище, в свою чергу, можуть об'єднуватися у набори міждисциплінарного обміну і розглядатися як *об'єктно-орієнтована база даних*, підпорядковувана наборам *семантичних правил* (об'єкти, атрибути, зв'язки між ними тощо) і організована ("закодована") у відповідності із умовами, заданими вимогами навчального процесу.

Поняття семантики при цьому спирається на те, що будь-який навчальний або відображуючий його засіб-об'єкт володіє як сутнісно-інформаційними, так і функціонально-описовими властивостями. У відповідності з цим мультимедійне середовище КІКЗ ДВН складається з двох типів знання-орієнтованих об'єктів: навчаючих і відображуючих. *Навчаючі об'єкти* (наприклад, підручник – навчально-науково-методичний інформаційний вузол, лекція – сутнісний навчальний компонент, практик – закріплюючий навчальний модуль, тестуюча система – модуль оцінки глибини засвоєння знань тощо) характеризуються як певними наборами знань, які визначають галузь об'єкта вивчення, так і послідовністю їх проходження в процесі на-

вчання. *Відображуючі об'єкти* володіють визначеним набором атрибутів і описують деякий *формат подання навчаючих об'єктів* або *мультимедійний засіб їх відображення*, наприклад: HTML-, PDF-формати, придатні до Internet-розповсюдження; 2D анімація – Flash, GIF та 3D анімація – 3D Max Studio, Maya, 3D Canvas для *подання графічного матеріалу* тощо. Між навчаючими і відображаючими об'єктами можуть існувати зв'язки різноманітного типу, які дозволяють форматувати набір КІКЗ ДВН для будь-якої спорідненої дисципліни. Таким чином, при формуванні об'єктно-орієнтованого на певну навчальну дисципліну мультимедійного середовища запроваджується можливість формування запитів навчально-науково-методичного інформаційного пошуку, заснованого на семантиці навчаючих і відображаючих онтологій (в даному випадку онтологія — це спроба всеосягаючої і детальної формалізації визначеного мультимедійного середовища за допомогою концептуальної схеми).

Означення ПЗ.2. *Семантика об'єкта у мультимедійному середовищі КІКЗ ДВН* (його *семантична характеристика*) – це частина інформації про об'єкт, якою подаються властивості навчаючого і відображаючого об'єкта. Наприклад, *семантика топоніма* – змістовне значення назви навчального або відображаючого об'єкта. *Цифровий опис об'єкта* – формалізоване уявлення в цифровому вигляді даних про об'єкт, яке містить рубрикацію й позначення складових навчальних і відображаючих об'єктів, їх семантику і відношення об'єктів у відповідності з *класифікатором*. Перелік семантичних характеристик задається *класифікатором об'єктів* мультимедійного середовища КІКЗ ДВН.

Означення ПЗ.3. *Семантична узгодженість об'єктів* мультимедійного середовища КІКЗ ДВН – це співпадання або логічна відповідність *семантик* пов'язаних навчаючих і відображаючих об'єктів.

Вимоги щодо семантичної узгодженості КІКЗ ДВН. *Семантична узгодженість* забезпечується своєчасним усуненням виникаючих протиріч в *семантичних характеристиках зв'язаних об'єктів* при формуванні або переформатуванні мультимедійного середовища під нові або споріднені навчальні дисципліни. *Метрична узгодженість об'єктів* забезпечується при введенні інформації додержанням правил подання *метрики об'єктів* всіляких класів. Метрична і семантична узгодженості об'єктів однієї дисципліни є обов'язковими і разом з тим не завжди обов'язковими для різних об'єктів (різних дисциплін, інтегровуваних у мультимедійне середовище КІКЗ ДВН).

Об'єкти класифікуються у відповідності з притаманними їм ознаками. Ознаки об'єктів поділяються на основні та змінні. *Основними* є ознаки, які однозначно визначають класифікаційне групування, в яке входить об'єкт. *Змінними* є семантичні характеристики, які не впливають на віднесення об'єкта до класифікаційних груп. Семантичні характеристики можуть бути обов'язковими і необов'язковими. Система *класифікації* і створені у відповідності до неї *класифікатори* повинні однозначно визначати належність усіх підлягаючих класифікації *об'єктів і їх семантичних характеристик* до класифікаційних групувань. При визначенні складу семантичних характеристик об'єктів мультимедійного середовища КІКЗ ДВН не включаються характеристики, значення яких можуть бути отримані автоматичним обчисленням на основі загально-описової інформації і (або) значень інших характеристик. Контроль повноти та узгодженості обов'язкових до включення *семантичних характеристик і логічних зв'язків* за умов виконаної формалізації визначеного мультимедійного середовища за допомогою концептуальної схеми може здійснюватися програмно.

Класифікація семантик повинна забезпечувати включення усіх семантичних характеристик навчальних і відображуючих об'єктів, передбачених для мультимедійного середовища даної навчальної дисципліни, а також властивостей, необхідних для віднесення об'єктів до видів певних засобів навчання. Всі *семантичні характеристики* можуть бути розбиті на групи: найменування засобу-об'єкта; кількісні характеристики; якісні характеристики; характеристики функціональних зв'язків тощо. *Семантика об'єкта повинна описувати* сутність і властивості засобу-об'єкта в обсязі, передбаченому заданими вимогами та розробленими на їх основі класифікаторами. *Семантика об'єкта повинна містити*: ідентифікатор засобу-об'єкта; коди його характеристик; значення характеристик засобу-об'єкта. *Характеристики об'єкта* можуть приймати значення: числа; символічної строки; алфавітно-цифрового коду; значення із списків можливих значень; імені файлу будь-якого виду. *Семантика об'єкта може містити* спеціальні характеристики, які несуть інформацію щодо відношень об'єктів.

Проблемно-орієнтований КІКЗ ДВН. При створенні КІКЗ ДВН в основу було покладено його реалізацію за окремими напрямкам знань, об'єднаними навчальним планом у відповідні напрямки підготовки студентів (наприклад, проектно-технологічний, схемо- та системотехнічний тощо).

У підручнику запропоновано підхід до складу та змісту, який за допомогою КІКЗ ДВН забезпечує:

- **по-перше**, використання матеріалу у навчальному процесі одразу для декількох спеціальностей;
- **по-друге**, подання всього матеріалу про властивості, підходи щодо створення і застосування нейрокомп'ютерних систем (НКС) та їхніх інтелектуалізованих версій у логічній послідовності його викладення і взаємозв'язку складових підручника без повторень і дублювання навчального матеріалу;
- **по-третє**, подання матеріалу про НКС таким чином, щоб кожна його складова мала самостійне значення і давала змогу забезпечувати навчальний процес інших споріднених спеціальностей навчально-методичним матеріалом, коли окрема частина підручника (його складові й окремі їх частини) відповідає вимогам методичного і змістовного забезпечення конкретного теоретичного курсу з відповідної навчальної дисципліни;
- **по-четверте**, поєднання на звичайному (паперовому) і електронному носіях, причому перший призначений для лекційного викладення курсу з поданням ключових вузлових аспектів проблемної галузі, тоді як електронна версія охоплює ширше коло питань щодо навчальної дисципліни з детальним її аналізом і призначена для студентів з урахуванням їх самостійної роботи з відповідною навчально-методичною літературою;
- **по-п'яте**, супроводження електронної версії підручника мультимедійною анімацією складних для сприймання студентом деяких елементів функціонування систем, поданих рисунком у статичному вигляді з текстовою формою опису принципу дії; кольорове анімаційне «оживлення» статичних зображень елементів гнучких об'єктно-орієнтованих систем (ГООС) та їх інтелектуалізованих версій дає змогу користувачеві дослідити всі необхідні для сприймання деталі функціонування різноманітних за складністю, фізичною основою та послідовністю дії компонентів як під час їх експлуатації, так і в процесі проектування, моделювання та управління з використанням сучасних нейротехнологій;
- **по-шосте**, оснащення електронної версії підручника підсистемами контролю залишкових знань і тестування (входять окремими складовими в КІКЗ ДВН), у результаті користування якими студенти мають можливість виконати самоаналіз

глибини засвоєння навчального матеріалу. *Валідність тестуючих систем* забезпечується їх апробацією і коректуванням на протязі понад десяти років при викладанні дисципліни студентами різних спеціальностей і форм навчання.

Потрібно зазначити, що створення підручника з такими властивостями для проектно-технологічного та спеціального напрямку підготовки студентів означених та інших споріднених через базову (бакалаврську) підготовку спеціальностей є актуальним внаслідок незабезпечення україномовними підручниками, тим більше - їх інтелектуалізованими електронними версіями. Крім того, у вітчизняній практиці немає, окрім розроблених раніше авторами, підручників з комплексним поданням дисциплін професійно-орієнтованої та спеціальної підготовки, а також підручників, електронні версії яких оснащено мультимедійним анімаційним супроводженням і тестуючими системами*.

У підручнику наведено основи створення нейромоделей *інтелектуалізованих комп'ютерно-інтегрованих комплексів* (ІКІК) широкого призначення, системного забезпечення формування інтелектуальних систем, а також методи і принципи моделювання складових ІКІК в задачах класифікації, кластерного аналізу, розпізнавання і відтворення спотворених образів, прогнозування, прийняття рішень в умовах невизначеності, управління тощо.

Подано термінологію та основні поняття, що застосовуються в нейротехнології та нейрокомп'ютерних системах, розглянуто поширені моделі подання знань із застосування сучасних нейротехнологій і дано їх порівняльну характеристику. Наведено приклади реалізації умовних інтелектуальних систем підтримки рішень щодо моделювання та управління технологічними і організаційно-технічними системами.

Розкрито особливості формування топологій нейросіток з різними методами їх навчання, логічних виведень і прийняття рішень в умовах лінгвістичної невизначеності при гіперпросторовому моделюванні процесів в складних і різних за своїми фізичними, структурними та функціональними властивостями об'єктних застосуваннях. Викладено основи створення інтелектуалізованих мультиагентних систем в задачах прийняття рішень щодо вибору топологій НС, задовольняючих властивості об'єкта нейромоделювання.

*Першим в Україні мультимедійним виданням був комплекс підручників: Гнучкі комп'ютеризовані системи: проектування, моделювання і управління: Підручник / Л. С. Ямпольський, П. П. Мельничук, Б. Б. Самотокін, М. М. Поліщук, М. М. Ткач, К. Б. Остапченко, О. І. Лісовиченко. – Житомир: ЖДТУ, 2005. – 680 с.

Підручник проілюстрований численними прикладами реальних систем інтелектуальної підтримки при розв'язанні задач розпізнавання образів, комівояжера, апроксимації складних функцій, прогнозування, а також управління автономними мобільними транспортними засобами в умовах невизначеностей.

Вибір принципів реалізації системи тестування. Існує досить багато систем тестування (СТ), які використовують різноманітні алгоритми проходження тесту.

В КІКЗ ДВН було запропоновано дві тестові підсистеми: самоконтролю користувача, яка безпосередньо вбудована в КІКЗ ДВН, та перевірки знань викладачем як окрема складова.

В СТ враховані наступні функціональні вимоги:

- наявність ефективних засобів розробки тестів та адміністрування;
- наочність та простота процесу самоперевірки залишкових знань;
- інформативність;
- стабільність та безвідмовність роботи.

Оскільки підручник створювався з метою забезпечення можливості і дистанційного навчання, то основним засобом подання матеріалів є формати PDF та HTML. Враховуючи цю особливість, СТ інтегрована в КІКЗ ДВН і може завантажуватися як з робочого місця користувача, так і через Інтернет.

Підсистема самоконтролю складається з модуля підготовки тесту, який використовується для можливості формування ефективної бази даних (БД) тестових запитань з варіантами відповідей, та html-файлів з елементами JavaScript для відображення в головному фреймі підручника тестових запитань до відповідних його розділів, які використовуються для подання тесту та відображення результату користувачу.

З існуючих найпоширеніших реалізацій модуля підготовки тесту таких систем за технологіями клієнт-сервер (БД зберігається на віддаленому ресурсі) та клієнт-клієнт (вся необхідна інформація знаходиться на клієнтській машині) в КІКЗ ДВН застосовано другий варіант, який надає можливість завантажувати систему на локальному комп'ютері, не приєднаному до мережі.

Наочність і простота забезпечуються гнучкою структурою і різновидами форм побудови тестів, а інформативність підтримується контекстними підказками під час проходження тесту.

Підсистема перевірки знань створена за технологією клієнт-сервер з огляду на вимоги компактності та легкого налаштування на функціонування. Означені вимоги за рахунок створення модуля підсистеми, який функціонує в режимах адміністратора та звичайного користувача з наданням таких можливостей:

- проходження тесту студентом за обраною тематикою;
- адміністративні дії викладача, пов'язані з введенням БД тестів і студентів;
- отримання звітів з рейтинговими оцінками за результатами тестувань;
- активний моніторинг процесу проходження тестування.

Особливістю реалізації підсистеми є можливість відновлення працездатності й оновлення інформації проходження тесту в разі виникнення нештатних ситуацій у функціонуванні (зависання і термінове відключення комп'ютера), що досягається такою реалізацією структури БД, в яку оперативно вноситься інформація про кожну відповідь користувача.

Особливості реалізації пошукової системи. Система пошуку інформації (СПІ) для цього КІКЗ ДВН розроблялася з урахуванням можливості реалізації таких задач:

- адаптації СПІ до будь-якого локального ресурсу;
- автоматичного оновлення БД виявлених ресурсів;

формування критеріїв пошуку з урахуванням різних видів (текст, малюнки, анімація, відео, таблиці тощо) подання інформації.

Оскільки ефективність будь-якої СПІ визначається швидкістю пошуку та якістю фільтрації даних з урахуванням їх видів, у КІКЗ ДВН використовується структурованість за видами інформації та її індексація за ключовими словами в кожному з видів. Крім того, результати в даній реалізації СПІ дають змогу подавати знайдені ресурси згідно з оцінкою входження пошукової інформації в кожний знайдений ресурс у спадному порядку.

Додаткові можливості розширеного пошуку забезпечуються використанням БД синонімів ключових слів, що дає змогу вживати як запити скорочені ключові слова та аббревіатури.

Отже, результатом розробки одного з перших в нашій державі системних проєктів КІКЗ ДВН є створення зручного за інтерфейсом і багатофункціонального за призначенням робочого місця студента і викладача, яке, враховуючи реалізовані властивості, дає змогу

використовувати його як базову оболонку навчаючої системи для різних напрямків підготовки в системі освіти. Слід зазначити, що такий підхід до створення навчальної літератури був успішно апробований авторами на прикладі створення шести підручників для різних дисциплін декількох спеціальностей.

В основу підручника покладені курси “Штучні нейронні сітки”, “Нейротехнології та нейросистеми”, “Нейротехнології та нейрокомп’ютерні системи”, які читаються авторами студентам Національного технічного університету України – “КПІ” на протязі останніх двадцяти років.

Перспективи. Наведений підхід до формування мультимедійного середовища вимагає реалізації принципів семантичної узгодженості як навчально-науково-методичних об’єктів навчального процесу, так і мультимедійних засобів комп’ютерних технологій відображення і подання навчальної інформації. Ця узгодженість повинна на *першому етапі створення* мультимедійних навчальних комплексів слугувати усуненню виникаючих протиріч в семантичних характеристиках зв’язаних об’єктів при формуванні або переформуванні мультимедійного середовища під нові або споріднені навчальні дисципліни, а в *подальшому* – створювати умови для використання оболонки мультимедійного середовища для забезпечення можливості автоматизованого формування семантично-узгодженого КІКЗ ДВН під будь-які навчальні дисципліни.

Комп’ютерний набір підручника у PDF-форматі, створення мультимедійного анімаційного супроводу графічного матеріалу та демонстраційних прикладів презентацій, лабораторних практикумів, а також тестуючих систем виконано під керівництвом Л. С. Ямпольського, О. І. Лісовиченка і В. В. Олійника, аспірантами А. М. Дзінько та Р. І. Дзіньком за участі студентів 4-5 курсів напряму підготовки 0502 “Системна інженерія” Національного технічного університету України “КПІ”, яким автори висловлюють щире подяку.

Публікації книги автори зобов’язані рекомендаціям Вченої ради НТУУ-“КПІ”, а також спонсорській підтримці ІТ-підприємця, випускника 2006 р. кафедри Технічної кібернетики нашого університету З. В. Діхтяра. Автори висловлюють вдячність професорам В. М. Казаку, С. І. Осадчому, П. І. Бідюку за копітку наукове рецензування підручника і висловлені цінні зауваження і поради, які були враховані при його доопрацюванні і сприяли підвищенню науково-методичного рівня і якості тексту.

СПИСОК СКОРОЧЕНЬ

*Зайва інформація мозком відсіюється,
щоб звільнити місце для некорисної*

Анекдот

Пишіть коротко і неясно

Наполеон I Бонапарт

ААП	– автоасоціативна пам'ять
АГК	– аналіз головних компонент
АЗПП	– алгоритм зворотного поширення похибок
АКК	– алгоритм каскадної кореляції
АЛМ	– алгоритм Левенберга-Марквардта
АМП	– автомасштабне перетворення
АОМ	– аналогова обчислювальна машина
АОП	– агентно-орієнтована підсистема
АП	– асоціативна пам'ять
АР	– агент реалізації
АРТ	– адаптивна резонансна теорія
АС	– асоціативна сітка
АФ	– активаційна функція
А _{шнс}	– архітектура штучної нейронної сітки
БД	– база даних
БЗ	– база знань
БН	– базовий нейрон
БП	– база правил
БРС	– багатошарова рекурсивна сітка
ВОЗ	– властивості та обмеження задач
ГА	– генетичний алгоритм
ГАП	– гетероасоціативна пам'ять
ГБФ	– гіпербазисна функція
ГІМАК	– гнучка інтелектуалізована мультиагентна конфігурація
ГІМАС	– гнучка інтелектуалізована мультиагентна система
ГО	– генетичний оператор
ГООС	– гнучка об'єктно-орієнтована система
ДАП	– двоспрямована асоціативна пам'ять
ДВН	– дистанційно-віртуальне навчання
ДНК	– дезоксирабінуклеїнова кислота
ДРНС	– динамічна рекурсивна нейронна сітка
ЕлО	– елементарний об'єкт
ЕлС	– елементарна складова
ЕО	– експертна оцінка
ЕОМ	– електронно-обчислювальна машина

Список скорочень

ЕРК	– електроробочар
ЕС	– експертна система
ЗНС	– згортаюча нейронна сітка
ЗПП	– зворотне поширення похибок
ІКІК	– інтелектуалізований комп'ютерно-інтегрований комплекс
ІНС	– інтелектуалізована нейронна сітка
ІМП	– інтермальне масштабне перетворення
ІЧНС	– ієрархічна частково-рекурсивна нейронна сітка
КБЗ	– критерій балансу змінних
КЗОР	– коефіцієнт завантаження обробного ресурсу
КІКЗ	– комп'ютерно-інтегрований комплекс засобів
ККНС	– каскадно-кореляційні нейронні сітки
КМЗ	– критерій мінімуму зсуву
КУПД	– конкретна унікальна послідовність дій
ЛМПС	– логічна модель поетапного синтезу
ЛРНС	– локально-рекурсивна нейронна сітка
ЛСПС	– логічна схема поетапного вибору
МАК	– мультиагентне керування
МАП	– мультиагентна підсистема
МАРТС	– мультиагентна робототехнічна система
МАС	– мультиагентна система
МАСе	– мультиагентне середовище
МГУА	– метод групового урахування аргументів
МЗПП	– метод зворотного поширення похибки
МЛВ	– механізм логічних виведень
МНК	– метод найменших квадратів
МН _{ШНС}	– метод навчання штучної нейронної сітки
МОК	– метод опуклого комбінування
МС _{ШНС}	– метод спрощення штучної нейронної сітки
НВКО	– набір вирішальних класифікаційних ознак
НІ	– нейроінформатика
НІХ	– нескінченна імпульсна характеристика
НК	– нейрокомп'ютер
НКС	– нейрокомп'ютерна система
НН ШНС	– нейро-нечітка штучна нейронна сітка
НО	– нейрообчислювач
НП	– нейропроцесор
НС	– нейронна сітка
НТ	– нейротехнологія
НФК	– нейро-фаззі контролер
НФС	– нейро-фаззі-система
ОВТД	– об'єкт вибору з табличних даних

ОІВ	– об'єкт інтелектуального вибору
ОК	– об'єкт керування
ОНС	– осциляторні нейронні сітки
ОПГ	– об'єкт побудування графіків
ПАЗ	– пам'ять з адресацією за змістом
ПЗ	– прикладна задача
ПЗП	– програмно-задаючий пристрій
ПЛІС	– програмувана логічна інтегральна схема
ПП	– правило продукції
ПДРНС	– повнозв'язна динамічна рекурсивна нейронна сітка
ПС	– прогноуюча система
ПСНС	– прямоспрямовані нейронні сітки
РБФ-сітка	– сітка радіально-базисних функцій
РБПД	– рекурсивний багат шаровий персептрон Джордана
РНС	– рекурсивна нейронна сітка
РТС	– робототехнічна система
СЗП	– сітка зустрічного поширення
СІНС	– стохастична інтелектуалізована нейронна сітка
СІХ	– скінчена імпульсна характеристика
СК	– синергетичний комп'ютер
СКП	– середньоквадратична похибка
СПІ	– система пошуку інформації
СС	– семантична сітка
С _{ШНС}	– структура штучної нейронної сітки
СТ	– система тестування
ТВ ШНС	– топологічні властивості ШНС
ТКМ	– телекомунікаційна мережа
ТП ШНС	– технологічний простір ШНС
TSK	– Такагі-Сугено-Канга нейронна сітка
УМВТТ	– узагальнена модель вибору типових топологій
УРНС	– узагальнено-регресійна нейронна сітка
ФК	– фаззі-контролер
ФН	– функція належності
ФНН	– функція нечіткої належності
ФСЗ	– формальна система знань
ФСІА	– функціонально-спеціалізований інтелектуалізований агент
ЧП	– чисельна процедура
ЧРНС	– частково-рекурсивна нейронна сітка
ШІ	– штучний інтелект
ШІС	– штучна інтелектуалізована система
ШН	– штучний нейрон
ШНС	– штучна нейронна сітка
ШНМ	– штучна нейромережа

ПЕРЕДМОВА

*В науці немає широкої стовпової дороги,
і тільки той може досягти її сяючих вершин,
хто, не боючись втоми, дереться по її кам'янистих схилах*
Карл Маркс

Штучні нейронні сітки. Вдосконалення елементної бази, яка визначає архітектуру комп'ютера, та розпаралелювання обчислень дозволяють швидко та ефективно розв'язувати задачі все зростаючої складності. Проте, такі задачі, як розпізнавання (ідентифікація) образів (наприклад, знайоме обличчя людина впізнає за 100-120 мс, в той час, як сучасний комп'ютер це робить за хвилини або години), розуміння мови та тексту, написаного від руки, та багато інших, комп'ютер, навіть за умов надзвичайної швидкодії, часто не в змозі своєчасно розв'язати. Отже, сітка нейронів, що утворює мозок людини і уявляє собою, як і комп'ютерна мережа, систему паралельної обробки інформації, в багатьох випадках виявляється ефективнішою. Ідея переходу від обробки закладеним в комп'ютері алгоритмом деяких формалізованих знань до реалізації в ньому властивих людині прийомів обробки інформації (розумової діяльності) привели до виникнення *штучних нейронних сіток* (ШНС). Основна ідея, закладена у функціонування ШНС, це послідовне перетворення сигналу паралельно працюючими елементами – *штучними нейронами* (ШН).

Означення П.1. *Штучні нейронні сітки* – математичні моделі та їх програмні і/або апаратні реалізації, виконані за принципом організації біологічних нейросіток нервових клітин живого організму, уявляють собою систему з'єднаних і взаємодіючих штучних нейронів (простих процесорів).

З ШНС все в більшій мірі пов'язують подальше підвищення продуктивності комп'ютерів, зокрема, у вигляді *нейрокомп'ютерів* (НК), основу яких і складає саме ШНС. Останніми роками сформувався новий науково-технічний напрям – *нейротехнологія* (НТ).

Означення П.2. *Нейротехнологія* – складова загальної науки нейроінформатики, задачами якої є дослідження принципів утворення, методів побудування і особливостей функціонування ШНС.

Означення П.3. *Нейроінформатика* – область наукових досліджень на перетині нейронаук та інформатики, в сферу якої входить збирання отриманих в ході нейробіологічних досліджень ре-

зультатів з їх подальшим використанням за допомогою обчислювальних моделей і спеціалізованих комп'ютерних аналітичних програмних інструментальних засобів для дослідження усіляких аспектів функціонування і будови нервових систем, а також моделювання процесів, які в них відбуваються, за допомогою ШНС.

Нейронні сітки та штучний інтелект. Нейроінформатика і нейрокібернетика є одним з напрямів *штучного інтелекту* (ШІ).

Означення П.4. *Штучний інтелект* – це напрям інформатики, метою якого є розробка програмно-апаратних засобів, які дозволяють розв'язувати традиційно інтелектуальні задачі.

В число цих задач входить створення інтелектуальних роботів, оптимальне керування, навчання і самонавчання, розпізнавання образів, прогнозування тощо [78] – тобто саме ті задачі, для яких нейронні сітки й застосовують найбільш широко.

Якщо при створенні класичних експертних систем людина-експерт повинна спочатку формалізувати свої знання (подати їх природною мовою у вигляді набору правил або шаблонів), і отримання несуперечливого і повного формулювання знань є довготривалим і трудомістким процесом, то нейросіткові експертні системи самонавчаються на базі експериментальних даних (фактів). Це робить можливим створення нейроекспертних систем за відсутності людини-експерта, наприклад, для галузей, де необхідною є діагностика.

Нейросіткові алгоритми успішно застосовуються для розв'язання складних практичних задач, які традиційно вважаються інтелектуальними – розпізнавання облич, об'єктів на зображенні, керування безпілотними літальними апаратами, діагностування захворювань та ін.

Отже, оскільки апарат нейронних сіток є формальною системою знань (див. розд 1.6) і дозволяє не тільки оперувати останніми, але й створювати нові знання за рахунок спроможності формування логічних висновків, нейротехнології є потужним інструментарієм в системах із штучним інтелектом.

Проте, між ШНС і традиційним ШІ існує така ж принципова різниця, яка є між людською свідомістю та окремою популяцією нейронних клітин у пробірці.

Дослідження щодо ШІ базуються на так званій гіпотезі символічних систем, яка припускає, що якщо створити досить об'ємну пам'ять, занести в неї доволі складно організованих структур даних і

зв'язати все це достатньо універсальними алгоритмами опрацювання, то можна отримати машину, поведінка якої та її спроможність переробляти інформацію нічим не буде відрізнятися від людської. Це, по суті, проєціювання на область ІІІ загальнофілософського принципу – закону переходу кількості у якість.

Але для створення системи ІІІ в традиційному розумінні необхідно у явному виді запрограмувати весь процес придбання і використання отриманих знань. При цьому проєктувальники такої системи втручаються у недосить досліджену на сьогодні область – їм необхідно зрозуміти, як живі істоти придбають нові знання і в якому виді вони зберігаються всередині. Очевидно, частину знань ми отримуємо з досвіду, частину у виді готових правил, а ще частина нам дістається генетично у виді рефлексів. До цього слід додати, що існують певні складності з побудуванням моделі міркувань (наприклад, каскадні умовиводи, коли кожний наступний умовивід робиться на підставі попереднього і коректується з урахуванням знову надходильної інформації), врешті решт, традиційний ІІІ зазнає значних труднощів з побудовою механізмів самомодифікування своєї внутрішньої структури, тобто з організацією процесу навчання.

Таким чином, можливості нейронних сіток в деякому сенсі перевершують традиційний ІІІ. Так, в НС досить легко організувати процес самомодифікації структури шляхом видалення або додавання окремих нейронів (див. *розділи 17, 23*). Значно легше організується й процес навчання, оскільки сітка навчається по суті як “чорна скринька” на прикладах. Необхідності у побудуванні структури даних і розробці правил роботи з ними у програміста в даному разі нема. Проте очевидно, що штучний мозок на двох-трьох нейронах не побудуєш, а зі зростанням кількості нейронів стрімко зростає практична складність навчання подібної сітки. Через це найкращим варіантом на даному етапі розвитку нейроінформатики можна вважати комбінування цих двох основних підходів: штучні нейронні сітки можна з успіхом використовувати для моделювання рецепторів і первинного опрацювання інформації, а традиційний ІІІ непогано зарекомендував себе в області найбільш абстрактних узагальнень.

Сучасні нейрокомп'ютери. Багаторічні зусилля багатьох дослідницьких груп привели до того, що на теперішній час накопичена велика кількість усіляких «правил навчання» і архітектур нейронних

сіток, їх апаратних реалізацій і прийомів використання ШНС для розв'язання прикладних задач.

Ці інтелектуальні винаходи [14] існують у вигляді “нейропарку” ШНС. Кожна сітка з нейропарку має свою архітектуру, правило навчання і розв'язує конкретний набір задач. В останнє десятиріччя докладаються серйозні зусилля для стандартизації структурних елементів і перетворення цього нейропарку в “технопарк” [49]: кожна нейросітка з нейропарку реалізована на ідеальному універсальному нейрокомп'ютері із заданою структурою. Основні правила виокремлення функціональних компонентів ідеального НК:

- відносна функціональна відокремленість – кожна компонента має чіткий набір функцій. Її взаємодія з іншими компонентами може бути описаною у вигляді невеликої кількості запитів;
- можливість взаємозаміни всіляких реалізацій будь-якої компоненти без зміни інших компонент.

Означення П.5. *Нейрокомп'ютер* – пристрій реалізації коннекціоністської ідеї нейробіоніки щодо опрацювання інформації на основі формалізованих принципів роботи природних нейросіток.

Поступово створюється ринок нейрокомп'ютерів. В теперішній час широко розповсюджені різні високопаралельні нейроприскорювачі [36] (співпроцесори) для всіляких задач. Моделей універсальних НК на ринку мало почасти тому, що більшість з них реалізовані для спецпризначень. Видаються спеціалізовані науково-технічні журнали, проводяться щорічні конференції по НК. З технічної точки зору сьогоденні нейрокомп'ютери – це обчислювальні системи з паралельними потоками однакових команд і численним потоком даних (MSIMD-архітектура). Це один з основних напрямків розвитку обчислювальних систем з масовим паралелізмом. Штучна нейросітка може передаватися від (нейро)комп'ютера до (нейро)комп'ютера так само, як і комп'ютерна програма. Більше того, на її основі можуть бути створені спеціалізовані швидкодіючі аналогові пристрої. Виділяються декілька рівней відчуження ШНС від універсального (нейро)комп'ютера [50]: від нейросітки, яка навчається на універсальному пристрої і використовує широкі можливості у маніпулюванні задачником, алгоритмами навчання і модифікації архітектури, до повного відчуження без можливостей навчання і модифікування, з використанням тільки функціонування навченої сітки.

Одним зі способів підготовки ШНС для реалізації передачі є її вербалізація: навчену ШНС мінімізують із збереженням корисних навичок.

Означення П.6. *Вербалізація* – мінімізований опис роботи синтезованої і вже навченої ШНС у вигляді декількох взаємозалежних алгебраїчних або логічних функцій. Опис мінімізованої сітки є більш компактним і часто припускає зрозумілу інтерпретацію.

Одним з основних недоліків навчуваних ШНС, з точки зору багатьох користувачів, є той факт, що з навченої сітки важко витягти явний і зрозумілий користувачу алгоритм розв'язання задачі – сама ШНС є цим алгоритмом, і якщо структура сітки є складною, то цей алгоритм незрозумілий. Проте, спеціальним чином побудована процедура спрощення і вербалізації часто дозволяє витягти явний метод розв'язання.

Отже, однією з цілей вербалізації є підготовка навченої і спрощеної НС до реалізації у програмному коді або у вигляді спеціалізованого електронного пристрою, а також для використання результатів у вигляді явних знань. Під симптомами при цьому розуміються вхідні значення НС, а під синдромами – значення на виходах сітки.

Вербалізація зазвичай здійснюється засобами спеціалізованих пакетів.

На відміну від цифрових систем, уявляючих собою комбінації процесорних і запам'ятовуючих блоків, *нейропроцесори* (НП) містять пам'ять, розподілену у зв'язках між дуже простими процесорами, які часто можуть бути поданими як формальні нейрони або блоки з однотипових формальних нейронів. Тим самим основне навантаження на виконання конкретних функцій процесорами покладається на архітектуру системи, деталі якої в свою чергу визначаються міжнейронними зв'язками.

Розробники НК прагнуть об'єднати стійкість, швидкодію і паралелізм *аналогових обчислювальних машин* (АОМ) з універсальністю сучасних комп'ютерів [15].

Ці принципи були формалізовані, що дозволило говорити про теорію ШНС. Проблематика ж НК полягає в побудові реальних фізичних пристроїв, що дозволить не просто моделювати ШНС на звичайному комп'ютері, але й так змінити принципи його роботи, що

буде можливим говорити про те, що НК працюють у відповідності з теорією ШНС.

Проблематика НК полягає у побудові реальних фізичних пристроїв, що дозволятимуть не просто моделювати ШНС на звичайному комп'ютері, але й так змінити принципи його роботи, що уможливить роботу НК у відповідності з теорією ШНС.

Означення П.7. *Коннекціонізм* – це підхід до уявлення нейропроцесорів, заснований на поданні як пам'яті, так і алгоритмів системою зв'язків (та їх вагами).

Терміни *нейробіоніка*, *нейрокібернетика*, *нейроінформатика*, *нейрокомп'ютери* увійшли у наукову обіходь порівняно нещодавно – з середини 80-х років XX століття. Проте, електронний і біологічний мозок постійно порівнювалися на протязі всієї історії існування обчислювальної техніки. Відома книга Н. Вінера “Кібернетика” [10] має підзаголовок “Керування і зв'язок у тваринному і машині”.

До перших НК відноситься й програма «Кора», розроблена у 60-х роках минулого століття під керівництвом М. М. Бонгарда [8].

Велику роль у розвитку нейрокомп'ютинга відіграла робота Розенблатта [47; 187].

Ідея нейробіоніки (створення технічних засобів на нейропринципах) стала інтенсивно реалізовуватися на початку 1980-х років.

Одним з імпульсів до розвитку НК було розв'язання наступного протиріччя: розміри елементарних компонентів комп'ютерів ставали порівнюваними з розмірами елементарних “перетворювачів інформації” в нервовій системі, до того ж було досягнуто швидкодії окремих електронних елементів у мільйони разів вищої, аніж у біологічних систем. Проте ефективність розв'язання задач, особливо пов'язаних з орієнтуванням і прийняттям рішень у природному середовищі, у “живих систем” поки недосяжно вища.

Другий імпульс розвитку НК дали наукові розробки 80-х років минулого століття з теорії ШНС.

Сутність усіх підходів НТ полягає в наступному: розробка методів створення (синтезу) нейронних схем, розв'язуючих ті чи інші задачі. Відмінність між підходами і методами – в деталях уявлень про роботу нейрона, і, звичайно, в уявленнях про роботу зв'язків. Власне, пристрої нейроінформатики уявляють собою зв'язкові системи. На відміну від цифрових мікропроцесорних систем, уявляючих

собою складні комбінації процесорних і запам'ятовуючих блоків, нейропроцесори містять пам'ять, розподілену у зв'язках між дуже простими процесорами. Тим самим основне навантаження на виконання конкретних функцій процесорами лягає на архітектуру системи, деталі якої в свою чергу визначаються міжнейронними зв'язками.

Яким чином нейросітка розв'язує задачі, багато з яких часто є неформалізованими або важко формалізованими? Як відомо, для розв'язання таких задач традиційно застосовується два основних підходи. Перший, заснований на правилах (rule-based), є характерним для експертних систем і базується на описі предметної області у вигляді набору продукційних правил (аксіом) «ЯКЩО ..., ТО ...» та правил виведення. Шукане знання представляється у цьому випадку теоремою, істинність якої доводиться шляхом побудовання ланцюжка виведення. Проте, при такому підході необхідно заздалегідь знати весь набір закономірностей, описуючих предметну область. При використанні іншого підходу, заснованого на прикладах (case-based), треба лише мати достатню кількість прикладів для настроювання адаптивної системи із заданим ступенем достовірності. Нейронні сітки уявляють собою класичний приклад такого підходу.

Відмінною особливістю біологічних систем є адаптація, завдяки якій такі системи в процесі навчання розвиваються і придбають нові властивості. Як і біологічні нейронні сітки, ШНС складаються з пов'язаних між собою елементів – *штучних нейронів* (ШН), функціональні можливості яких в тій чи іншій мірі відповідають елементарним функціям біологічного нейрона.

На роль центральної проблеми, розв'язуваною всією нейроінформатикою і нейрокомп'ютигом, А. Горбанем була запропонована проблема ефективного паралелізму [14]. Як факт відомо, що продуктивність комп'ютера зростає набагато повільніше, ніж кількість використовуваних процесорів – приблизно логарифму кількості процесорів (це набагато повільніше, ніж лінійна функція – так звана гіпотеза М. Мінського). Для подолання цього обмеження застосовується наступний підхід: для різних класів задач будуються максимально паралельні алгоритми розв'язання, які використовують яку-небудь абстрактну архітектуру (парадигму) дрібнозернистого паралелізму, а для конкретних паралельних комп'ютерів створюються засоби реалізації паралельних процесів заданої абстрактної архітектури. В ре-

зультаті з'являється ефективний апарат здійснення паралельних програм.

Нейроінформатика постачає універсальні дрібнозернисті паралельні архітектури для розв'язання усіляких класів задач. Для конкретних задач будується абстрактна нейросіткова реалізація алгоритму розв'язання, яка потім реалізується на конкретних паралельних обчислювальних пристроях. Таким чином ШНС дозволяють ефективно використовувати паралелізм.

Можна навести три основні аспекти, що сприяли, як і у випадку із штучним інтелектом [78], розвитку напряду створення й впровадження ШНС у розв'язання прикладних задач. Перш за все це *гносеологічний аспект*, який полягає в тенденції людини до пізнання самої себе, властивостей і можливостей свого мозку і, врешті решт, інтелекту. Тут найважливішим для людини є створення математичної моделі самоусвідомлення для розв'язання задач самовдосконалення, найповнішого використання можливостей, закладених у людину природою. Тому відбувається оцінювання того, що стимулює прагнення людини до пізнання через сприйняття навколишнього середовища і адаптації в ньому, і це є джерелом її прагнення до створення штучних емульованих засобів, які б відтворювали можливості розумової діяльності людини в цьому процесі.

Другим є *інтенціональний застосовний аспект*, коли знання й уявлення можливостей людини дають змогу визначити сферу та міру використання їх. Тому в цьому аспекті використання НТ спрямоване на розвиток і доповнення обмежених можливостей людини (наприклад, людина краще розв'язує задачі одновимірного розпізнавання, тоді як електронна НК потенційно може розв'язувати задачі у вимірностях і постановках, недосяжних для людини, і не тільки з позицій швидкодії та обсягів обчислень, як у разі звичайного застосування ЕОМ, а вже й з позицій інтелектуалізації задач).

Третім є *екстенціональний застосовний аспект*, коли НК розв'язує задачу нехай і не краще за людину, але дає змогу поширити її досвід і знання для розв'язання значно більшої кількості задач, ніж це дозволила б зробити реальна (обмежена) кількість експертів у даній галузі знань, яких для цього можна залучити. Це й є звичайною галуззю застосування *експертних* (ЕС) і *прогнозуючих* (ПС) систем в широкому використанні у багатьох сферах діяльності людини.

Отже, поява ШНС пов'язана з розумінням того, що істотний

мозок як складна нелінійна паралельна інформаційно-керуюча система працює інакше, ніж комп'ютер: вона спроможна до мислення, накопичення і відновлення інформації. З інженерної точки зору ШНС саме і уявляє собою паралельно розподілену систему обробки інформації, утворену тісно зв'язаними простими обчислювальними вузлами, спроможну накопичувати експериментальні знання, узагальнювати їх і робити доступними для використання у формі, зручній для інтерпретації і прийняття рішень.

За аналогією зі своїм природним прототипом ШНС володіє наступними властивостями [48]:

- адаптивне навчання – спроможність покращувати свої характеристики за рахунок закладеної у тому чи іншому алгоритмі налагодження параметрів сітки, відпрацьовуючим пред'явлені їй навчаючі послідовності або використовуючим наявний досвід;
- самоорганізація – нечутливість сітки після закінчення процесу її навчання до певних змінювань вхідних сигналів, що дозволяє застосовувати ШНС при зашумлених/спотворених або не повністю заданих даних;
- обчислення у реальному часі – спроможність здійснення нейросіткових обчислень паралельно у часі, що суттєво підвищує швидкість ШНС;
- стійкість до збоїв – збереження деяких властивостей сітки навіть у випадках руйнування значної її частини, хоча ця властивість і супроводжується певною втратою якості.

Як і у будь-якій іншій галузі знань, інтелектуальний прорив у широкому використанні НТ став можливим, тільки коли для його реалізації були дотримані необхідні і достатні умови (див. розд. “Передмова загальна”).

Тепер дослідження в області теорії і практики ШНС здійснюються в напрямках пошуку нових нелінійних елементів, спроможних реалізовувати складну колективну поведінку в ансамблі нейронів, створенні нових архітектурних сполучень нейронних сіток, пошуку областей їх застосування в системах обробки зображень, розпізнавання образів і мови, робототехніки, прийняття рішень в керуванні, оптимізації та прогнозування тощо. Значне місце в даних дослідженнях традиційно відводиться математичному моделюванню.

Все вищезгадане супроводжується наявністю об'єктивної соціальної потреби у впровадженні елементів НТ у виробництво.

Сьогодні можна виокремити наступні сфери досліджень в галузі НТ [48; 64]:

- проектування ШНС – створення нових топологій нейросіток, включаючи структурно-архітектурний синтез;
- нейроінформатика – дослідження процесів та удосконалення методів їх навчання;
- нейрокомп'ютинг – підвищення продуктивності НК за рахунок удосконалення розпаралелювання обчислень;
- нейроморфна інженерія – розв'язання прикладних проблем із застосуванням НТ.

Щодо комерційного використання на ринку ШНС і НК, то слід відзначити, що річний обсяг їх продаж наприкінці минулого століття перевищував 3 млрд. доларів, а щорічний приріст становив 50%. С початку 21-го століття з широким використанням переходу на субмікронні і нанотехнології, а також подальшими успіхами молекулярної і біомолекулярної технологій створюються умови для переходу до принципово нових архітектурних і технологічних рішень по створенню сучасних нейрокомп'ютерів.

Необхідність підготовки даного підручника з теорії ШНС і *нейрокомп'ютерних систем* (НКС), розвитку НТ в області прийняття рішень багато в чому визначається обмеженістю україномовних наукових видань з цієї проблематики. Крім того, сама ця проблематика поки не зайняла своє місце в традиційних курсах університетів і вищих технічних навчальних закладів, тоді як вже сьогоднішній рівень теоретичного розуміння і практичного використання нейронних сіток у світовій інформаційній індустрії все виразніше потребує професійних знань в цій області.

Головною метою пропонованого видання є практичне введення в сучасні методи і системи обробки інформації, об'єднані в науковій літературі терміном “обчислювальна нейронаука” (Computational Neuroscience), а також ознайомлення з перспективними підходами щодо побудування обчислювальних та інформаційних систем нових поколінь.

Важливою особливістю розглядуваної в підручнику проблематики є її міждисциплінарний характер через те, що свій внесок в становлення нейронауки внесли біологія і фізіологія вищої нервової діяльності, психологія сприйняття, дискретна математика, статистична фізика і синергетика, а також, безперечно, кібернетика і

комп'ютерне моделювання. Тому, глибоке вивчення ШНС потребує знань нейрофізіології, науки про пізнання, психології, фізики (статистичної механіки), теорії управління, теорії обчислень, проблем штучного інтелекту, статистичної математики, розпізнавання образів, комп'ютерного зору, паралельних обчислень і апаратних засобів (цифрових і аналогових). Зі свого боку, ШНС також стимулює ці дисципліни, забезпечуючи їх новими інструментальними засобами і уявленнями. Цей симбіоз життєво необхідний для дослідження НТ і НКС.

Цей підручник є одним з перших україномовних видань з проблем НТ і нейрокомп'ютерних систем, що створювало методологічні та термінологічні труднощі при її поданні як наукового та навчального джерела.

В ньому наведено огляд відомих ШНС, який має в основному ілюстраційний та пізнавальний характер. Важливу роль при цьому відіграють приклади, що пояснюють структуроутворення нейромоделей, правила їх навчання і особливості функціонування та ін. Це робить книгу дуже корисною не тільки для, аспірантів і науковців, спеціальність яких пов'язана зі створенням нейрокомп'ютерних систем, а й для студентів і користувачів, які не є спеціалістами в галузі НТ, але зацікавлені в їх використанні.

Разом з тим розглядаються й численні приклади реалізації більшості топологій ШНС, в яких механізми прийняття рішень ґрунтуються на засадах певних архітектурних і структурних реалізацій, методів навчання і застосування всіляких видів активаційних функцій, використання механізмів нейро-фаззи технологій і т. п.

В книзі наведений детальний аналіз основних топологій ШНС, в тому числі розглядаються можливості створення нейроконтролерів та використання найро-фаззи технологій для розв'язання задач розпізнавання образів, контролю та керування технологічними процесами в умовах невизначеності, зокрема, у випадках, які характеризуються надзвичайними умовами функціонування обробляючих ресурсів в складних комп'ютерно-інтегрованих системах, де необхідне опрацювання нечіткої інформації, відображення суттєво нелінійних багатопараметричних об'єктів, які важко формалізуються.

Нарешті, однією з перших в літературі, присвяченій проблематиці НТ, розглядається можливість побудування автоматизованої системи вибору топологій ШНС, адекватних вимогам модельованих прикладних задач. З метою реалізації концепції нечіткої метаіденти-

фікації в задачах вибору ШНС запропоновані класифікатор ШНС, який базується на *наборі вирішальних класифікаційних ознак* (НВКО), і *логічна схема поетапного синтезу* (ЛСПС) топології ШНС, задовольняючої властивості модельованої задачі.

Не дивлячись на всезростаючу кількість прикладів практичного застосування нових інформаційних технологій в розв'язанні і моделюванні прикладних задач, напрямок, пов'язаний із створенням НКС, все ж належить в більшій мірі майбутньому. Адже більшість створених і створюваних систем носять ще поки що експериментальний характер і дедалі все більше удосконалюються. Нові інформаційні технології, як і інші напрямки науки і техніки, які живлять розробки в галузі НТ, знаходяться в стадії становлення та інтенсивного розвитку досліджень (в тому числі, фундаментальних) як по центральній проблемі – створення НКС – так і принципам, методам, технології побудування ШНС, розробці їх архітектури, технології застосування, створенню і перевірці спеціальних мов програмування і специфікацій, орієнтованих на розв'язання задач з проблематики НТ.

Зрозуміло, що зміст книги не є настільки повним, щоб починати самостійне проектування нейросистем, але він достатній для того, щоб ознайомитись з досягненнями в галузі їх розробки і увійти в коло проблем, супроводжуваних цей процес, вибрати самостійно тип моделі ШНС стосовно конкретної предметної галузі й почати роботу з реалізації НК разом із спеціалістом у галузі системотехніки.

Користувачі напрямів підготовки “Системна інженерія” та “Автоматизація та комп'ютерно-інтегровані технології” можуть ознайомитись також з об'єктно-орієнтованою спрямованістю наведених у підручнику прикладів, що безпосередньо стосуються їх власної кваліфікації, а саме – проблемою керування і навігації в робототехніці, в гнучких комп'ютерно-інтегрованих виробництвах, розпізнавання образів у широкому розумінні цієї задачі в різних аспектах її застосування, прогнозування, стиснення даних, розв'язання оптимізаційних задач тощо.

Зазначимо також, що науковці інших споріднених спеціальностей, для яких матеріал з проблематики НТ та штучного інтелекту складає основний напрям їх наукової діяльності, мають змогу підвищити базові знання з принципів побудови ШНС як вискоєфективної моделі подання знань, а також розширити коло своїх професійних інтересів наведеними прикладами практичного застосування НКС в інтелектуалізації процесів в різних предметних галузях і різного призначення.

ВСТУП

Найважча частина – вступ

Марк Тулій Цицерон

Немає нічого, на що б не дерзнуло людство

Лукреций Кар Тит

В.1. Хронологія розвитку нейротехнології

Характеристика проблеми. На початку 40-х років минулого століття з'явилися дві фундаментальні роботи, в яких вперше формалізується поняття *штучного нейрона* (ШН) і *штучної нейронної сітки* (ШНС) – це робота У. Мак-каллоха та В. Піттса [169] про логічне числення ідей та нервової активності та фундаментальна праця Н. Вінера [10] і його колег про подання складних біологічних процесів математичними моделями (в подальшому закріпленому у терміні “кібернетика”). З появою запропонованого Ф. Розенблаттом *одношарового перцептрона* [187] (який використовують для розпізнавання образів, прогнозування тощо), а в подальшому розробленої на основі *дельта-правила – формули Відроу* і побудованого на принципово нових засадах адаптивного лінійного елемента – *Адаліни* Б. Відроу і М. Хоффа [204] значно підвищується зацікавленість на комерційному рівні у розробках в галузі НТ. Цьому в значній мірі сприяв запропонований Д. Хеббом перший *алгоритм навчання нейрона* [131].

Проте проведені у 1963 році під керівництвом А. П. Петрова в Інституті проблем передачі інформації АН СРСР дослідження задач, “важких” для перцептрона [44], явилися відправною точкою для Н. Н. Бонгарда [8], щоб “невеликою переробкою алгоритму (перцептрона) виправити його недоліки”. Ці дві роботи і стали запорукою того, що перша хвиля ейфорії відносно ШНС була пригальмована.

А наприкінці 60-х і майже до середини 70-х років минулого століття спостерігається вже різке спадання активності у розвитку НТ та їх прикладного застосування, яке було викликане спостереженими М. Мінським і С. Пайпертом [171] обмеженнями у роботі (модельовання виключно найпростіших логічних функцій, складності у навчанні та перенавчанні тощо) і пов'язаними з інваріантністю подання (так звана проблема “парності” та “один у блоці”) штучного нейрона (ШН) Мак-Каллоха і Піттса, а також перших моделей одношарового перцептрона Ф. Розенблатта і адаліни Б. Відроу і М. Хоффа. І хоча у дослідників НТ було розуміння можливості будувати багатшарові сітки з, очевидно, більш широкими властивостями, але за-

лишалось нез'ясованим питання щодо навчання схованих шарів. Проте, окремі дослідження в галузі НТ продовжувалися, і серед них, в першу чергу, треба відзначити самоорганізуючу НС – *конгнітрон* (та *неокогнітрон*), запропонований К. Фукушимою [114; 116] і призначений для інваріантного розпізнавання образів (досягається при запам'ятовуванні практично усіх станів образу).

Величезного значення також набули фундаментальні розробки Д. Хопфілда [144; 145], в яких на основі функцій Ляпунова досліджувалася проблема стійкості симетричних рекурсивних ШНС з перспективною і надзвичайно важливою для просування НТ на комерційний ринок ідеєю створення моделі НС з оберненими зв'язками, яка уявляє собою мінімізуючу енергетичну систему. Сконструйована Д. Хопфілдом ШНС, названа його ім'ям (*сітка Хопфілда*), спроможна розв'язувати складні оптимізаційні задачі, наприклад, *задачу про комівояжера*.

Крім того, необхідно відзначити ряд праць Т. Кохонена [151; 154], в яких обґрунтовується можливість створення моделей НС, які навчаються без учителя (*самоорганізуючі мapi* Кохонена) і використовуються в задачах кластеризації, візуалізації та інших для попередньої обробки даних. Подібні архітектури ШНС запам'ятовують образи, що на них надходять, асоціюючи їх з чим-небудь, а при пред'явленні деякого “асоціативного” образу добувають їх з пам'яті. Дана властивість дозволяє організувати пошук інформації не за адресою, а за її вмістом, і це може бути зроблено ШНС, навіть якщо пред'явлена асоціація буде перевернута перешкодами. Всі ці дослідження набули значного розвитку в реалізації можливості відтворення спотворених або зашумлених даних і базуються на природній властивості мозку використовувати асоціативні зв'язки при пошуку інформації за допомогою ключа – послідовності бітів, яка порівнюється зі всіма ключами інформації, що зберігається.

Якщо в моделі обчислень фон Неймана звертання до пам'яті є доступним тільки за допомогою адреси, яка не залежить від вмісту пам'яті, і більш того, якщо допущена помилка в обчисленні адреси, то може бути знайдена зовсім інша інформація, то *асоціативна пам'ять* (що адресується за змістом) є доступною по указанню заданого змісту. Нейросітки із зворотним процесом – відновлення вихідного образу з його частини або з його спотвореного (зашумленого/пошкодженого) оригіналу – дістали назву (*авто*)*асоціативної пам'яті*. Створення на ШНС *гетероасоціативної пам'яті* дозволяє реалізовувати пам'ять, яка адресується за вмістом. Значний внесок у розв'язання проблеми асоціативного навчання і створення гетеро-

авто- і двоспрямованої асоціативної пам'яті, а також відтворення принципів конкурентного навчання зробили також Дж. Андерсен [83], С. Гроссберг [122] і Б. Коско. [158].

Тільки з появою запропонованого в дисертації П. Вербоса [202] і паралельно в роботі О. І. Галушкіна [11], але спочатку залишеного поза уваги дослідниками, а в подальшому розвинутого Д. Румельхартом, Г. Хінтоном і Р. Уільямсом [188] та незалежно і одночасно С. І. Барцевим та В. О. Охоніним [5; 85] методу навчання з використанням *алгоритму зворотного поширення похибки* (АЗПП) з'явився потужний теоретичний фундамент для конструювання ефективних *багатошарових* ШНС. Розпочався період вибухової зацікавленості до ШНС, які навчаються.

Нарешті, великого значення набувають сітки з *радіально-базисними функціями* (РБФ-сітки), що уявляють собою спеціальний тип ШНС з прямими зв'язками і основне призначення яких – апроксимація та інтерполяція багатомірних функцій для розв'язання, зокрема, задач прогнозування. Вони запропоновані М. Повеллом [184] і спираються на математичну основу теорії апроксимації та інтерполяції багатовимірних функцій, раніше розробленої М. А. Айзерманом, Е. М. Браверманом і Л. І. Розоноєром [1]. Суть математичної основи базується на тому, що скільки завгодно точна апроксимація функцій досягається шляхом комбінації *радіально симетричних функцій*.

Якщо першу переконливу перевагу використання ШНС при розв'язанні складних прикладних задач доводить робота Т. Сейновського і Ч. Розенберга щодо автоматизованого навчання англійській мові [189], то сьогодні вже важко уявити область досліджень, будь то технічне або технологічне проектування, діагностування, генетика, біоніка, авіоніка, економіка, ергономіка, комунальне господарство, транспортні задачі і багато інших застосувань, де би не впроваджувалися НТ. Саме ця зацікавленість з боку користувачів спонукає дослідників в області НТ до створення численних топологій ШНС як спеціального, так і універсального призначення. Проте, багатоваріантність вибору ШНС, в свою чергу, створює утруднення у прийнятті адекватного рішення у користувача, і це, на наш погляд, і складає основну проблему використання НТ.

Ускладнення проблеми. Вище наголошувалося на тому, що після “перевідкриття” і суттєвого доопрацювання Д. Румельхартом, Г. Хінтоном і Р. Уільямсом і незалежно та одночасно С. І. Барцевим та В. О. Охоніним методу навчання з використанням АЗПП наприкінці минулого – початку теперішнього століть створилися підстави до “ренесансу” НТ із все зростаючою зацікавленістю з боку користувачів і задоволенням попиту новими розробками у топологіях ШНС.

Останнє відбувається двома шляхами:

1. За рахунок використання сполучень, модифікацій і розширень вже існуючих моделей ШНС;
2. Створенням моделей ШНС на нових архітектурних, структурних і топологічних принципах побудування і методах навчання або використання їх для обслуговування інших оптимізаційних алгоритмів, зокрема, генетичних.

Яскравим представником *першого шляху* може служити сполучення двох ШНС: Кохонена і Гроссберга з новим методом навчання – “зустрічного поширення” (ЗП), запропонованим та розвиненим Р. Хехт-Нільсеном [133]. Такий симбіоз різних ШНС дозволив покращити властивості результуючої сітки (зменшення часу навчання; можливість роботи як з неперервними, так і з двійковими сигналами; блокове побудування сітки шляхом каскадування спеціалізованих модулів тощо) і розширити сфери застосування (апроксимація функцій, розпізнавання, обробка і відновлення зображень, класифікація). Існує два різновиди двошарових ШНС ЗП: шар Кохонена – шар Гроссберга та самоорганізуюча карта Кохонена – шар Гроссберга (різниця в організації схованого шару і, відповідно, в способі навчання його нейронів) [124].

Розширенням за рахунок модифікації вузлів на вхідному шарі стандартної архітектури багатoshарового прямошарованого персептрона з методом зворотного поширення похибки стали *функціонально-пов'язані сітки, розроблені Йох-хан Пао* і детально описані в його роботі [213]. Входи комбінуються зрозумілим математичним шляхом за допомогою функцій вищого порядку (квадрати, куби, синуси, косинуси, функції LOG, MAX тощо). Із назв цих функцій вищого порядку або функціонально пов'язаних входів і впливає назва НС. В тензорній топології таких сіток у модель додаються перехресні добутки вхідних елементів, і кожна компонента вхідного образу перемножується зі всіма компонентами вхідного вектора. В топології НС функціонального розширення додаються вхідні вузли з наведеними вище функціями. Повний ефект забезпечують сітки з об'єднанням обох топологій. Ніякої нової інформації не додається, але розширене представлення входів робить сітку простішою для навчання при не занадто великій кількості входів. В протилежному випадку зростає час навчання.

Іншим прикладом розробок першого шляху розвитку ШНС є створення на базі ідеї сітки Хопфілда з додаванням спеціального модуля нелінійного перетворення нової парадигми, що працює з аналоговими сигналами – “*синергетичного комп'ютера*” (СК), запропонованого засновником синергетики Х. Хакеном [127]. Цей комп'ютер

відноситься до асоціаторів з інтерпретацією образів, які в ньому зберігаються, як локальні енергетичні мінімуми. Перевага над сіткою Хопфілда – відсутність нестійких станів завдяки наявності динамічного модуля, що збігається до одного з базових векторів.

Суттєве скорочення витрат на пам'ять і об'єм необхідних обчислень досягається в *сітці Хеммінга* [164], в якій замість образу, що зберігається у пам'яті, на виході видається тільки його номер. Тому, на відміну від сітки Хопфілда, місткість сітки Хеммінга визначається не розмірністю вхідного сигналу N , а кількістю нейронів шару L на виході. Крім того, перевагою сітки Хеммінга є те, що залежність між кількістю нейронів і кількістю з'єднань в сітці Хеммінга лінійна, тоді як у Хопфілда – квадратична, і при однаковій ємності з сіткою Хопфілда перша буде містити значно менше сполучень між нейронами.

Ще один приклад створення нової топології за рахунок використання базової моделі багат шарового персептрона прямого поширення являє собою ШНС з елементами затримки сигналу (ШНС ЗС). Основне призначення – краще опрацювання (класифікація) зсувених у часі образів (як правило, мовних сигналів), які надходять на вхідний шар із згасанням у часі. Так, оскільки корисна мовна інформація часто подається згасаючою у часі в середовищі перешкод (паузи між надходячими фонемами, зашумлення тощо), а тривалість окремих фонем може сильно коливатися, ШНС ЗС повинна однаково ефективно опрацьовувати такі фонemi, тобто бути робастною до зовнішнього зсуву надходячих на сітку образів [161; 201]. Відмінність такої сітки полягає в наявності можливості створення за рахунок елементів затримки або у самих нейронах, або між схованими шарами тимчасових копій надходячих сигналів, через що на нейрони наступного шару подаються не тільки самі образи, але й їх копії. Цього принципу дотримуються всі шари ШНС ЗС, завдяки чого і досягається робастність сітки до часових спотворень у вхідних образах.

Представником *другого шляху* може служити ШНС, яка дістала назву *машини Больцмана* [81; 135]. Якщо сітка Хопфілда збігається до локального мінімуму, то в новому різновиді ШНС збіжність має відбуватися у глобальному мінімумі, а оскільки на шляху до цього сітка проходить локальні мінімуми, то існує ймовірність “зависання” в них. Тому по мірі просування до глобального мінімуму на кожному локальному мінімумі алгоритмічно задається збільшення “умовної енергії”, достатньої для подолання локального максимуму. Надалі енергія “розгойдування” сітки поволі зменшується по мірі наближення до глобального мінімуму. Ця ідея і реалізована в машині Больцмана, сполучаючи детерміновану сітку Хопфілда з ймовірнісним/стохастичним правилом навчання, яке дістало назву відпалу.

Найчастіше цей різновид ШНС застосовується в задачах класифікації/розпізнавання невідомих або спотворених образів. Серед переваг – можливість виходу з локальних екстремумів, серед недоліків – трудомісткість пов'язаних з алгоритмом обчислень і, як наслідок, суттєві обчислювальні та часові витрати.

Трохи зменшує цей недолік *машина Коші* [179], в якій за рахунок використання в алгоритмі розподілу Коші можна скоротити час навчання; проте у порівнянні з персептроном цей час залишається занадто великим.

Класом ШНС, які яскраво доводять свою причетність до другого шляху розвитку суттєвими відмінностями від сіток Хопфілда як структурно (може мати декілька шарів з наявністю власних зворотних зв'язків у нейронів), так і особливостями навчання (матриця ваг може бути несиметричною, а навчання є контрольованим шляхом використання АЗПП), слугують неперервні і дискретні *динамічні рекурсивні сітки* (ДРС) [81; 82; 180; 188]. Наявність зворотних зв'язків між нейронами різних шарів і шару на виході включно забезпечує цьому класу ШНС додаткові позитивні властивості, яких неможна досягти в статичних багатошарових ШНС прямого поширення: можливість роботи з образами, параметри яких змінюються в часі. Дискретні ДРС поділяються на:

- *повнозв'язні* [180; 188] (застосування: аналіз та опрацювання послідовностей, ідентифікація нелінійних динамічних об'єктів);
- *частково-рекурсивні* – багатошарові персептрони, доповнені контекстним шаром, кількість нейронів якого співпадає з кількістю виходів ШНС. Останнє має місце в сітці М. Джордана [148] (застосування: розпізнавання і класифікація образів, прогнозування часових рядів, а також більш ефективно в порівнянні з ШНС прямого поширення асоціювання надходячих на вхід образів з послідовностями на виході – так зване “проковзне” вікно; недолік – повільна збіжність і виникаючі через це проблеми сталості) або в модифікованій версії цієї ШНС – *сітці Дж. Елмана* [102] з сигналами зворотних зв'язків не вихідного шару, а з виходів нейронів контекстного шару з кількістю нейронів, співпадаючою з кількістю нейронів схованого шару;
- *локально-рекурсивні* з лінійною структурою як багатошаровий персептрон або як сітка радіальної основи, в яких рекурсивність завжди обмежується одним нейроном і інтерпретується як фільтр із скінченною або нескінченною імпульсними характеристиками [174].

Представниками нового класу ШНС прямого поширення з можливістю за необхідності розширювання своєї архітектури в процесі функціонування є *каскадно кореляційні сітки* (ККС) з лінійними або/та нелінійними активаційними функціями нейронів і викорис-

танням алгоритму навчання з будь-яким правилом (дельта-правило, АЗПП тощо), запропоновані С. Фалманом і К. Леб'єром [107]. Навчання ККС повторюється до досягнення бажаної точності для всіх пар образів і базується на: покроковому інкрементному уведенні нейронів у сховані шари, які з'являються тільки після уведення нових нейронів; навчанні знов уведених нейронів при зберіганні ваг нейронів, які вже до того були у сітці. Переваги ККС: відсутність апіорного вибору архітектури сітки, бо як вихідна береться найпростіша з вхідним та вихідним шарами, а далі нарощується, поки сітка не визначить задовольняючу критеріям архітектуру; відсутність конкуренції між нейронами через їх незв'язність значно скорочує процес навчання в порівнянні з іншими топологіями сіток; з поданням нових образів може не відбуватися перенавчання всієї сітки, а тільки шляхом корекції зв'язків нейронів вхідного та схованого шарів; процес навчання сітки значно спрощується в обчислювальному сенсі, бо коректуються ваги тільки одного шару, і сигнали у сітці проходять тільки в одному напрямі. Призначення: основні переваги ККС проявляються в повній мірі при застосуванні в обчислювальних середовищах з використанням паралельних обчислень.

Проблема стабільності-пластичності є однією з найскладніших і важко розв'язуваних при побудуванні штучних систем, моделюючих сприйняття. Спосіб сприйняття зовнішнього світу живими організмами (і, насамперед, людиною) полягає у постійному оцінюванні: чи є деякий образ "новою" інформацією і, відтак, реакція на нього має бути пошуково-пізнавальною зі збереженням цього образу у пам'яті, чи цей образ є варіантом "старої" вже знайомої ситуації, і в цьому разі реакція організму повинна відповідати раніш накопиченому досвіду, а спеціальне запам'ятовування цього образу не потрібна. Таким чином, сприйняття одночасно є пластичним, адаптованим до нової інформації, і при цьому воно є стабільним, тобто не руйнує пам'ять про старі образи.

Традиційні ШНС виявилися не спроможними розв'язати проблему стабільності-пластичності. Дуже часто навчання нового образу знищує або змінює результати попереднього навчання. Наприклад, багатошаровий персептрон, який навчається за методом зворотного поширення похибки запам'ятовує весь набір навчаючої інформації, при цьому образи навчаючої вибірки пред'являються в процесі навчання багаторазово. Спробі потім навчати персептрон нового образу приводять до модифікації синаптичних зв'язків з неконтрольованим руйнуванням структури пам'яті щодо попередніх образів. Отже, персептрон не спроможний запам'ятовувати нову інформацію, і необхідно повне перенавчання НС.

Аналогічна ситуація має місце і в нейросітках Кохонена та Хеммінга, які навчаються на основі самоорганізації. Дані НС завжди видають позитивний результат при класифікації, і через це не в змозі виокремити нові образи від спотворених або зашумлених версій старих образів. В реальній ситуації НС буде наражатися під постійно змінюваним збуренням. За таких обставин ШНС, скоріше за все, не буде навчатися, і, змінюючи неперервно свої ваги, не досягатиме задовольняючих результатів.

Така нестабільність роботи ШНС в умовах змінюваної навчальної вибірки спонукала до розв'язання “дилеми стабільності-пластичності” (за С. Гроссбергом) шляхом створення двошарової *нейросітки адаптивної резонансної теорії* (АРТ) Г. Карпенером і С. Гроссбергом [94] дозволяє адаптивно навчатися при надходженні нових даних із збереженням стабільності щодо гарантії неруйнування попередньо накопичених даних, з одного боку, і достатньої пластичності відносно визначення суттєвості нових надходжень, з другого. Основне призначення АРТ – моделювання володіючих високим ступенем паралелізму архітектур самоорганізованих ШНС розпізнавання образів, відтворюючих їх біологічний і поведінковий контексти. Основні властивості сіток АРТ – стабілізація і стійкість процесу навчання (який є скінченим) і сталість процесу пошуку. В залежності від виду вхідних змінних і способу їх опрацювання розрізняють АРТ: з двійковими вхідними векторами (АРТ-1); з двійковими та неперервними вхідними векторами (АРТ-2); для моделювання біологічних процесів (АРТ-3), з двійковими вхідними векторами з використанням нечіткої логіки (Fuzzy-АРТ) [95].

Вважається, що *генетичні алгоритми* (ГА) були запропоновані та досліджені Дж. Холландом в 1975 році [142]. Проте, майже 10-ма роками раніше розроблений О. Г. Івахненко *метод групового урахування аргументів* (МГУА) [24-26] базувався на використанні одно- та багаторядних фільтруючих за певними критеріями алгоритмів саме для визначення “*моделей оптимальної складності*”, що і являли собою ніщо інше, як процедуру генетичного добору за певними ознаками якості. Відмінності МГУА від ГА полягають лише в тому, що родинні хромосоми останнього у МГУА мають вигляд складових поліноміальних попарних сполучень (“батьки”), а нові хромосоми (“діти”) – відібрані на етапах алгоритмування (кросовер, мутація) проміжні моделі складності в МГУА. Сітки, основані на МГУА [182], при складанні з m шарів можуть реалізовувати поліном степеня 2^m . “Просіюючи” на кожному з етапів багаторядного алгоритму через “*решето*” критеріїв отримані на попередньому етапі моделі (“*особини*” за термінологією ГА), можна керувати відбиранням най-

кращих. Більш того, МГУА є кращим методом для розв'язання задач ідентифікації та короткострокових прогнозувань, а для довгострокового прогнозування є не тільки найкращим, а й, можливо, унікальним методом, який забезпечує точність прогнозування за великим часом його випередження. Таким чином, як ГА, так і алгоритми МГУА є оптимізаційними і в ШНС можуть застосовуватися для пошуку оптимальних синаптичних ваг сіток зворотного поширення обмеженої розмірності [203], а також для визначення оптимальної структури ШНС під конкретну задачу [129]. Крім того, заслуговують на увагу дослідження, пов'язані з використанням ГА-підходу до *генетичного програмування* [159]. В загальному вигляді ГА можна подати наступною послідовністю кроків:

1. Подання всіх змінних, які оптимізуються, у вигляді бінарного або дійсного (в залежності від модифікації методу) ланцюжка параметрів – “хромосоми”, і далі випадковим чином створюється “популяція хромосом”;
2. Оцінювання кожної хромосоми з популяції за цільовою функцією;
3. Сорткування всіх оцінювань. Якщо кількість виконаних ітерацій не перевищує заздалегідь задане значення, то перехід до наступного кроку. Інакше рішенням є найкраща хромосома за результатами сортування на цьому кроці;
4. Відібрані на попередньому кроці найкращі представники популяції складають основу (“хромосом-батьків”) для створення за допомогою генетичних операцій (“кросовера” і “мутації”) нових хромосом (“хромосом-дітей”). Кросовер визначає рекомбінацію, за якою різні найкращі “хромосоми-батьки” даної популяції обмінюються деякими частинами бінарних ланцюжків. При цьому довільні двійкові розряди довільних хромосом можуть інвертуватися з малою ймовірністю цього процесу;
5. Оцінювання “хромосом-дітей” цільовою функцією і розміщення їх в популяцію на місця “гірших” її представників. Далі – перехід до пункту 3.

Позитивна якість ГА полягає в його універсальності – він дозволяє працювати як з неперервними, так і дискретними параметрами, і можливість його застосування не залежить від виду цільової функції; крім того, ГА спроможний знайти глобальний екстремум, тоді як інші алгоритми “зависають” на локальному екстремумі. Нарешті, ГА можна застосовувати на будь-якому етапі синтезу нечітких моделей, особливо у тих випадках, коли використання градієнтних методів неможливе.

До недоліків слід віднести досить низьку швидкість, яка значно знижується із зростанням кількості параметрів, що оптимізуються; крім того, ГА має велику кількість своїх параметрів – розмір популяції, довжина хромосоми, ймовірності кросовера і мутації, кількість “хромосом-дітей” тощо. Істотним недоліком є також явище “стогнації” як наслідок “невдалого” вибору параметрів з виродженням популяції або відсутністю прогресу.

Деякі типи ШНС володіють властивостями, які дозволяють використовувати ці сітки для стискування даних, наприклад, перед їх передаванням, зменшуючи тим самим кількість бітів інформації, які передаються. Подібні задачі виникають і в кластерному аналізі, коли різні схожі за визначеними ознаками образи об'єднуються в деякі групи або кластери, тобто здійснюється перехід від вихідного n -вимірного простору образів до m -вимірного простору кластерів, де $m < n$. Подальша робота у просторі меншої розмірності приводить до економії обчислювальних ресурсів та зменшенню об'єму необхідної пам'яті. Так, застосування ШНС для стиснення даних в системах розпізнавання мови та зображень забезпечує стиснення даних в 100 та більше разів [179].

Зрештою можна констатувати, що з найпоширеніших сфер застосування ШНС є розпізнавання образів. Це тим важливіше враховувати через те, що у більшості інших застосувань задача розпізнавання, навіть не будучи метою використання у явному вигляді, все ж припускає цю процедуру при надходячих на вхід образах з наступною їх класифікацією, кластеризацією тощо. Під кластеризацією мають на увазі розбиття множини вхідних образів на класи, коли ані кількість, ані ознаки класів не відомі. Сітка може констатувати, що вхідний образ не відноситься до жодного з виділених класів – це свідчить про те, що з'явилися нові образи з відмінними ознаками від тих, що визначають належність до певного класу. Тобто сітка виявляє нові, невідомі раніше класи образів, виконуючи функції кластеризатора.

З іншого боку, будь-яке застосування ШНС для розв'язання прикладних задач асоціюється з механізмом прийняття рішень при керуванні, рівно як і прийняття рішень, в свою чергу, базується на перебиранні можливих варіантів і виборі задовольняючого заданий показник ефективності. А це й є визначенням такого стану ШНС, яке відповідає “образу” ефективного керування [68].

Більшість практично важливих задач може бути сформульована як оптимізаційні, які забезпечують екстремум деякому наперед вибраному критерію. Однією з найбільш поширених серед них є так

звана задача комівояжера, розв'язання якої потребує значних обчислювальних ресурсів. Ця задача полягає в загальному вигляді у тому, щоб забезпечити найкоротший шлях проходження заданої кількості пунктів (точок простору), причому у кожному пункті (точці) перебуваючи не більше одного разу (наприклад, трасування печатних плат, обслуговування промисловими роботами технологічного устаткування, проектування оптимальних траєкторій руху робочих органів промислових роботів, обернена задача кінематики маніпуляційних систем з повним перебиранням можливих їх конфігурацій для забезпечення обслуговування технологічного процесу, логістичні задачі визначення оптимальних маршрутів перевезення об'єктів виробництва, товарів і послуг тощо). Це задачі, які прийнято відносити до "НР-повних" (недетермінована поліноміальність), кращим розв'язком яких є повне перебирання можливих варіантів. Оскільки для n пунктів (точок) обходу існує $n!$ варіантів обходу, то із збільшенням кількості пунктів складність розв'язання задачі різко зростає. Одне з ефективних рішень за допомогою ШНС, яке потребує менших обчислювальних витрат, до того ж зі співпадаючими результатами за методом повного перебирання, було запропоноване в роботі [145].

Проблема синтезу ефективної системи управління складними процесами уявляє собою досить складну задачу через те, що реальні процеси характеризуються, як правило, нелінійними залежностями, високим рівнем шумів та їх корелятивністю, змінними умовами функціонування тощо. Необхідність розв'язання задач управління в реальному часі висуває певні вимоги як до самих алгоритмів управління, так і до технічних засобів їх реалізації. В таких умовах найбільш ефективними виявляються методи та алгоритми, що базуються на *теорії адаптації*. Однак ці методи вимагають розробки таких математичних моделей досліджуваних об'єктів, які б не тільки застосовувалися з метою безпосереднього управління, але й для прогнозування поведінки об'єкта, що дозволяє підвищити ефективність управління шляхом упередженого коректування параметрів управління. Якщо отримання математичної моделі викликає труднощі, доцільне застосування ШНС [42; 183; 216].

Прогнозування (*екстраполяцію*) можна віднести до так званої "імпровізуючої" властивості ШНС у вигляді узагальнення і виокремлення схованих взаємозв'язків між даними на вході і виході. Статистичний аналіз процесів, які формуються під впливом великої кількості діючих по різних напрямках і часто невідомих факторів, включає

дослідження взаємозв'язків факторів як в статичному стані, так і в часі. Інформацією для вивчення взаємозв'язків слугують часові ряди показників, які характеризують розвиток об'єктів. Найбільш розповсюдженим підходом до розв'язання задач прогнозування є екстраполяція діючих в теперішній час зв'язків і закономірностей на майбутнє. Чим більш загальні припущення закладені у форму побудованої за таким принципом моделі прогнозування і чим більший клас процесів можна описати за її допомогою, тим ширші її можливості при дослідженні окремої реалізації. Проте на практиці часто виявляється, що через ті чи інші причини отримати математичну модель, яка адекватно відбиває властивості досліджуваного об'єкта, надзвичайно складно. Інакше кажучи, якщо у вхідній виборці даних є сховані закономірності їх впливу на змінювання даних на виході, то можна використовувати ШНС як прогнозуючий інструмент. А вже далі, на підставі цієї прогнозуючої функції, ШНС може приймати рішення при керуванні [87; 168; 183; 218].

В багатьох випадках буває необхідно підібрати для функції, заданої тільки таблично або графіком і відображаючої результати експериментальних даних, аналітичний вираз, відбиваючий ці залежності. Формула ж може виявитися досить складною і некоректною для вимаганої мети (наприклад, функція має бути проінтегрованою, а інтеграл від неї не виражається через елементарні функції). Тоді емпіричні формули для зображення необхідної функції $f(x)$ вибирають наближену функцію $g(x)$ як набір з функцій певного виду, вимагаючи, щоб функція $g(x)$ якомога ближче співпадала з $f(x)$ на деякому інтервалі ($a \leq x \leq b$). Нелінійна характеристика при цьому може бути довільною: від сигмоїдальної до будь-якого хвильового пакету, синуса або багаточлена. Від вибору нелінійної функції може залежати складність ШНС, але з будь-якою нелінійністю сітка залишається універсальним апроксиматором і при коректному виборі структури може досить точно апроксимувати функціонування будь-якого *неперервного автомата* [16]. Окрім того, ряд інших дослідників підтвердили, що ШНС з одним схованим шаром і з одним шаром на виході здатна апроксимувати з будь-якою наперед заданою точністю на компактній множині будь-яку неперервну функцію. Зокрема, Г. Цибенко та К. Харнік [97; 146], використовуючи теорему Ханна-Банаха, показали, що кінцева лінійна комбінація фіксованих одновимірних функцій може однозначно апроксимувати будь-яку неперервну функцію n дійсних змінних на заданому гіперкубі при досить м'яких припущеннях відносно функцій однієї змінної.

Теорема В1 (*про повноту*): будь-яка неперервна функція на замкненій обмеженій множині може бути рівномірно наближена функціями, обчислюваними нейронами ШНС, якщо активаційна функція нейрона двічі неперервно диференціювана і неперервна.

Отже, розширення інструментального “асортименту” різновидів ШНС приводить до важко розв’язуваної дилеми: дозволяючи більш “тонко” і професійно підходити до вибору моделі/топології ШНС, адекватної умовам обслуговування прикладної задачі, ускладнюється сам процес вибору адекватної розв’язуваним прикладним задачам ШНС через його багатоваріантність. За недостатньої професійної підготовленості користувача це може призводити до значних додаткових матеріальних (апаратних, обчислювальних) і/або часових витрат.

В.2. Шляхи розв’язання дилеми обґрунтованого вибору нейросіток

Постановка задачі щодо покращення ситуації. Роль і уявлення про ШНС, які й визначають зростаючий попит на застосування ШНС в різних сферах життєдіяльності людини, багатогранні [14]. З точки зору:

- машинного навчання – ШНС уявляють собою окремі випадок методів розпізнавання образів, кластеризації, дискримінантного аналізу тощо;
- математичної – навчання ШНС є багатопараметричною задачею нелінійної оптимізації;
- кібернетики – ШНС використовуються в задачах адаптивного управління і прогнозування;
- розвитку обчислювальної техніки – програмування ШНС є способом розв’язання проблеми ефективного паралелізму;
- штучного інтелекту – ШНС є основою філософської течії когнітивізму та основним напрямком у структурному підході по вивченню можливостей побудування/моделювання природного інтелекту за допомогою комп’ютерних алгоритмів.

Отже, обчислювальні системи на основі ШНС володіють якостями, яких позбавлені машини з архітектурою фон Неймана, проте притаманних мозку людини: масовий паралелізм; розподілене подання інформації та обчислень; здібність до навчання та узагальнення; адаптивність; властивість контекстуального оброблення інформації; толерантність до похибок; низьке енергоспоживання.

Апаратна реалізація ШНС – *нейрокомп’ютер* (НК) – має суттєві позитивні для широкого впровадження відмінності (як за струк-

турою, так і за класами розв'язуваних задач) від обчислювальних машин, виконаних у відповідності з традиційною архітектурою фон Неймана. У НК реалізується ідея створення аналого-цифрової ЕОМ, в якій аналогова частина виконує багатовимірні операції в пороговому базисі, а цифрова – реалізує алгоритми настроювання параметрів нейронних сіток [12].

Стратегічним напрямком розвитку НК як загального призначення, так і проблемно-орієнтованих є розробка *нейрочипів*. Дослідження в мікроелектроніці викликає переконаність в тому, що в недалекому майбутньому будуть створені більш потужні і в той же час більш дешеві нейронні ЕОМ [141].

Порівняльні характеристики нейрокомп'ютерів і традиційних комп'ютерів наведено у табл. В.1.

Таблиця В.1.

Порівняльні характеристики традиційних ЕОМ і НК

Категорія порівняння	ЕОМ традиційної архітектури	Нейрокомп'ютер
Процесор	Складний Високошвидкісний Один або декілька	Простий Низькошвидкісний Велика кількість
Пам'ять	Відокремлена від процесора Локалізована Адресація не за змістом	Інтегрована у процесор Розподілена Адресація за змістом
Обчислення	Централізовані Послідовні Програми, що зберігаються	Розподілені Паралельні Самонавчання
Надійність	Висока уразливість	Живучість
Спеціалізація	Чисельні та символічні операції	Проблеми сприйняття
Середовище функціонування	Суворо визначене Суворо обмежене	Без обмежень

Основні тези, які складають підґрунтя використання ШНС і нейромоделювання, полягають у наступному:

- нейросітка відтворює структуру і властивості нервової системи живих організмів – складається з великої кількості простих обчислювальних елементів (нейронів) і володіє більш складною поведінкою по відношенню до можливостей кожного окремого нейрона;

- нейросітка отримує на вході набір вхідних сигналів і видає відповідну до них відповідь (сигнали на виході НС), яка і є розв'язком задачі;

- як і природна біологічна нейросітка, ШНС може навчатися розв'язанню задач через наявність внутрішніх адаптивних параметрів нейронів і своєї структури і, змінюючи їх, може міняти свою поведінку;

- місце програмування займає навчання, тренування НС, і для розв'язання задачі не треба програмувати алгоритм;

- нейросітка навчається розв'язанню задачі на деякому “асоціаторі” – наборі ситуацій, кожна з яких описує значення вхідних сигналів НС і бажану для них відповідь. При цьому “асоціатор” задає набір еталонних ситуацій з відомими розв'язками, а НС при навчанні сама знаходить залежності між вхідними сигналами та відповідями.

Саме широка варіативність ШНС, наявність тонких відмінностей у принципах побудови і функціонування, відсутність строгих класифікаційних ознак і чітких рекомендацій щодо сфер застосування і розв'язуваних задач вимагають високої усвідомленості в питаннях вибору топологій ШНС у прикладного користувача, що виявляється не завжди виконуваним, перш за все, через його професійну невідповідність у галузі сучасних НТ. Отже, розрив між об'єктивною доцільністю використання новітніх досягнень в НТ для розв'язання прикладних задач і суб'єктивною неінформованістю кінцевого користувача обмежує можливості ефективнішого використання ШНС. Через це ШНС використовуються в двох варіантах: будується нейросітка, яка розв'язує певний клас задач; під кожний екземпляр задачі будується деяка нейросітка, яка знаходить квазіоптимальний розв'язок цієї задачі. Проте в обох випадках прийняття рішення покладатиметься на користувача.

Покращення ситуації нам уявляється у використанні процедур автоматизованого вибору адекватних розв'язуваним прикладним задачам ШНС:

- формуванні набору *вирішених класифікаційних ознак* (НВКО) і створенні класифікатора ШНС;

- побудуванні чіткої *логічної моделі поетапного синтезу* (ЛМПС) ШНС;

- створенні *гнучких інтелектуалізованих мультиагентних конфігурацій* (ГІМАК) *агентно-орієнтованих підсистем* (АОП) для конкретних прикладних задач, що базуються на формалізованих моделях подання знань [5; 78] з використанням НВКО та агентно-орієнтованого підходу [86; 74].

Розділ 1

Біологічні основи і поняття штучних нейронних сіток

*Детальне вивчення окремих органів
відучує розуміти життя всього організму*
Василь Ключевський

Дурість – це не відсутність розуму. Це такий розум
Генерал Лебедь

Структура зв'язків – усе, властивості елементів – ніщо
Основний лозунг нейрокомп'ютингу

1.1. Загальні відомості про біологічні нейросітки

Перш ніж перейти до розгляду моделей штучних нейронів і штучних нейронних сіток, сформулюємо загальні фактологічні положення про біологічні нейронні сітки у відповідності до [7].

Метод нейробіології. До предмета нейробіології відноситься вивчення нервової системи та її головного органа – мозку. Принциповим питанням для цієї науки є з'ясування співвідношення між *будовою* нервової системи та її *функцією*. При цьому розгляд виконується на декількох рівнях: молекулярному, клітинковому, окремого органа, організму в цілому і далі на рівні соціальної групи. Таким чином, класичний нейробіологічний підхід полягає у послідовному просуванні від елементарних форм у напрямку їх ускладнення.

Для наших практичних цілей відправною точкою буде клітинковий рівень. За сучасними уявленнями, саме на ньому сукупність елементарних молекулярних хіміко-біологічних процесів, протікаючих в окремій клітині, формує її як елементарний процесор, здатний до найпростішого опрацювання інформації.

Біологічний нейрон. Елементом клітинкової структури мозку є нервова клітина - *нейрон*. Нейрон у своїй будові має багато спільних рис з іншими клітинами біотканини: тіло нейрона оточено плазматичною мембраною, всередині якої знаходиться цитоплазма, ядро та інші складові клітини. Проте нервова клітина суттєво відрізняється від інших за своїм *функціональним призначенням*. Нейрон виконує прийом, елементарне перетворення і подальше передавання інформації до інших нейронів. Інформація переноситься у вигляді імпульсів нервової активності, які мають електрохімічну природу.

Нейрони вкрай різноманітні за формою, яка залежить від їх місцезорозташування у нервовій системі та особливостей функціонування. На рис. 1.1 наведена схема будови “типового” нейрона.

Тіло клітини містить безліч розгалужуваних відростків двох типів. Відростки першого типу, які називають *дендритами* за їх схожість з кроною дерева, слугують в якості входних каналів для нервових імпульсів від інших нейронів. Ці імпульси надходять до *сома* або *тіла* клітини розміром від 3 до 100 мікрон, викликаючи її специфічне збудження, яке потім розповсюджується по відростку на виході другого типу - *аксону*. Довжина аксонів зазвичай помітно перевищує розміри дендритів, в окремих випадках сягаючи десятків сантиметрів і навіть метрів. Гігантський аксон кальмара має товщину коло міліметра, і саме спостереження за ним дозволило виявити механізм передавання нервових імпульсів між нейронами.



Рис. 1.1. Загальна схема будови біологічного нейрона

Тіло нейрона, заповнене провідним іонним розчином, оточено мембраною завтовшки близько 75 ангстрем, володіючею низькою провідністю. Між внутрішньою поверхнею мембрани аксона і зовнішнім середовищем підтримується різниця електричних потенціалів. Це здійснюється за допомогою молекулярного механізму іонних насосів, утворюючих різну концентрацію позитивних іонів K^+ та Na^+ всередині і поза клітини. Проникливість мембрани нейрона є селективною для цих іонів. Всередині аксона клітини, перебуваючий у стані спокою, активне транспортування іонів прагне підтримувати концентрацію іонів калію більш високою, ніж іонів натрію, тоді як у рідині, оточуючій аксон, вищою є концентрація іонів Na^+ . Пасивна дифузія більш рухомих іонів калію веде до їх інтенсивного виходу з клітини, що обумовлює її загальний негативний відносно зовнішнього середовища *потенціал спокою*, який складає біля 65 мілівольт.

Під впливом стимулюючих сигналів від інших нейронів мембрана аксона динамічно змінює свою провідність. Це відбувається,

коли сумарний внутрішній потенціал перевищує граничне значення масштабу -50 мВ. Мембрана на короткий час, який сягає близько 2 мс, змінює свою полярність (деполяризується) і досягає *потенціалу дії* близько $+40$ мВ. На мікрорівні це пояснюється короткочасним підвищенням проникливості мембрани для іонів Na^+ та активним надходженням їх в аксон. У подальшому, по мірі виходу іонів калію, позитивний заряд з внутрішнього боку мембрани змінюється на негативний, і настає так званий *період рефрактерності* тривалістю близько 200 мс. На протязі цього часу нейрон є повністю пасивним, практично незмінно зберігаючи потенціал всередині аксона на рівні близько -70 мВ.

Імпульс деполяризації клітинкової мембрани, який називають *спайком*, розповсюджується уздовж аксона практично без згасання, підтримуючись локальними іонними градієнтами. Швидкість переміщення спайка є відносно невисокою і складає від 100 до 1000 сантиметрів за секунду.

Збудження нейрона у вигляді спайка передається іншим нейронам, які таким чином об'єднані у провідну сітку нервових імпульсів. Ділянки мембрани на аксоні, де розміщуються області контакту аксона даного нейрона з дендритами інших нейронів, називаються *синапсами*. В області синапсу, який має складну будову, відбувається обмін інформацією про збудження між нейронами. Механізми синаптичної передачі досить складні і різноманітні. Вони можуть мати хімічну та електричну природу. В хімічному синапсі у передаванні імпульсів приймають участь специфічні хімічні речовини – *нейромедіатори*, які викликають зміни проникливості локальної ділянки мембрани. В залежності від типу вироблюваного медіатора *синапс може володіти збуджуючою* (ефективно проводячою збудження) або *гальмівною дією*. Зазвичай на усіх відростках одного нейрона виробляється один і той же медіатор, і тому нейрон у цілому функціонально є гальмуючим або збуджуючим. Це важливе спостереження про наявність нейронів різних типів у наступних главах (когнітрон та неокогнітрон) буде суттєво використовуватися при проектуванні штучних нейронних систем.

Нейронні сітки. Взаємодіючі між собою шляхом передавання через відростки збуджені нейрони формують *нейронні сітки* (НС). Перехід від розглядання окремого нейрона до вивчення НС є природним кроком у нейробіологічній ієрархії.

Загальна кількість нейронів у центральній нервовій системі людини сягає $10^{10} - 10^{11}$, при цьому кожна нервова клітина пов'язана в середньому з $10^3 - 10^4$ іншими нейронами. Встановлено, що у голо-

вному мозку сукупність нейронів в об'ємі масштабу 1 мм^3 формує відносно незалежну локальну сітку, яка несе визначене функціональне навантаження.

Виокремлюють декілька (зазвичай три) основних типи НС, які відрізняються структурою і призначенням. Перший тип уявляє ієрархічні сітки, які часто зустрічаються в сенсорних і рухових шляхах. Інформація у таких сітках передається в процесі послідовного переходу від одного рівня ієрархії до другого.

Нейрони утворюють два характерних типи з'єднань – *конвергентні*, коли велика кількість нейронів одного рівня контактує з меншою кількістю нейронів наступного рівня, і *дивергентні*, в яких контакти встановлюються зі все більшою кількістю клітин наступних шарів ієрархії. Сполучення конвергентних і дивергентних з'єднань забезпечує багаторазове дублювання інформаційних шляхів, що є вирішним фактором надійності НС. При загибелі частини клітин нейрони, які збереглися, виявляються спроможними підтримувати функціонування сітки.

До другого типу НС відносяться *локальні сітки*, які формуються нейронами з обмеженими сферами впливу. Нейрони локальних сіток виконують опрацювання інформації в межах одного рівня ієрархії. При цьому функціонально локальна сітка уявляє собою відносно ізольовану гальмуючу або збуджуючу структуру.

Важливу роль також відіграють *НС третього виду* – так звані дивергентні сітки з одним виходом. Командний нейрон, розташований в основі такої сітки, може чинити вплив одночасно на безліч нейронів, і тому сітки з одним входом виступають узгоджувачим елементом у складному поєднанні нейросіткових систем усіх типів.

Розглянемо схематично НС, формуючу простий рефлексорний ланцюг з передачею збудження від подразника до рушійної м'язи (рис. 1.2).

Сигнал зовнішнього подразника сприймається *сенсорними* нейронами, зв'язаними з чутливими клітинами-рецепторами. Сенсорні нейрони формують перший (нижчий) рівень ієрархії. Вироблені ними сигнали передаються нейронам локальної сітки, які містять численність прямих і зворотних зв'язків зі сполученням дивергентних і конвергентних з'єднань. Характер перетвореного у локальних сітках сигналу визначає стан збудження *моторних* нейронів. Ці нейрони, які складають верхній в розглядуваній сітці рівень ієрархії, образно кажучи, “приймають рішення”, яке виражується у впливі на клітини м'язової тканини за допомогою нервово-м'язових з'єднань.

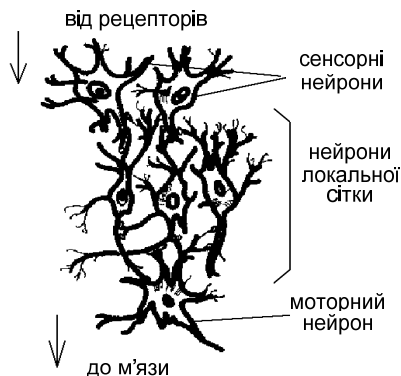


Рис. 1.2. Структура простої рефлекторної нейронної сітки

Генетична напередвизначеність структури нейронних сіток. Мозок людини уявляє собою сполучення нейроутворень, які дістали назву біологічних нейросіток (НС), тоді як в технічних дисциплінах уживанішим є термін “штучні нейросітки” (ШНС) або “штучні нейромережі” (ШНМ). Отже, свій розвиток ШНС отримали від сполучення двох наук – біології та електроніки і відносяться до технічної галузі – інформатики.

Треба відзначити, що, не дивлячись на те, що основною складовою мозку майже усіх живих істот є біологічні НС, кожна істота має свою унікальну побудову останніх у мозку. Ця *унікальність* визначається структурними та ієрархічними особливостями побудовання біологічних нейросіток і є наслідком *генної топології*; зі свого боку, ця *унікальність* визначає ступінь інтелектуальних спроможностей живої істоти. Інакше кажучи, структура основних типів НС *генетично напередвизначена*.

Можна також стверджувати, що, наприклад, серед людей *геніями, талановитими або особами з абиякими (пересічними) здібностями народжуються*, бо це саме ті якості мозку, які успадковуються через генні залучення. Інша справа, що навколишнє середовище, освіта, виховання можуть створити умови або до прояву закладених генною природою здібностей, або до не розкриття закладеного розумового потенціалу певної особистості. Так само можна стверджувати, що ніякі освіта у престижних навчальних закладах і наукових установах та елітне виховання не спроможні пересічну людину (тобто людину, яка успадкувала через генну спадковість посередні здібності) вивести в розряд талановитих або геніїв. Ці постулати є дуже важли-

вими з точки зору розуміння того, що ШНС можуть краще або гірше (за швидкістю, точністю апроксимації, якістю розпізнавання, прогнозування тощо) моделювати і відтворювати властивості застосованих задач, в першу чергу, від закладених в них особливостей базових ШН, структури, архітектури, а у другу чергу, за аналогією з біологічними НС, від впливу навколишнього середовища у вигляді залучених методів і засобів навчання, способів настроювання ваг, видів активізаційних (передатних) функцій (АФ) тощо.

Біологічна мінливість і навчання нейронних сіток. Дослідження в області порівняльної нейроанатомії говорять про те, що за фундаментальним планом будови мозок дуже мало змінився в процесі еволюції. Але не дивлячись на це, детерміновані нейронні структури демонструють властивості мінливості, які обумовлюють їх адаптацію до конкретних умов функціонування.

Генетична напередвизначеність має місце також і у відношенні властивостей окремих нейронів, таких, наприклад, як тип використовуваного нейромедіатора, форма і розмір клітини. *Мінливість на клітинному рівні* проявляється у *пластичності* синаптичних контактів. Характер метаболічної активності нейрона і властивості проникливості синаптичної мембрани можуть змінюватися у відповідь на тривалу активізацію або гальмування нейрона. Синаптичний контакт “тренується” у відповідь на умови функціонування.

Мінливість на рівні сітки пов’язана зі специфікою нейронів. Нервова тканина практично позбавлена характерної для інших типів тканин спроможності до регенерації шляхом поділу клітин. Проте нейрони демонструють спроможність до формування нових відростків і нових синаптичних контактів. Ряд експериментів з навмисним пошкодженням нервових шляхів доводить, що розвиток нейронних відгалужень супроводжується конкуренцією за володіння синаптичними ділянками. Ця властивість у цілому забезпечує сталість функціонування нейронних сіток при відносній ненадійності їх окремих компонент – нейронів.

Специфічна мінливість нейронних сіток і властивостей окремих нейронів лежить в основі їх *спроможності до навчання - адаптації до умов функціонування* - при незмінності у цілому їх морфологічної структури. Проте, слід зауважити, що розгляд мінливості та навчованості малих груп нейронів не дозволяє у цілому відповісти на питання про навчованість на рівні вищих форм психічної діяльності, пов’язаних з інтелектом, абстрактним мисленням, мовою.

1.2. Модель формального кібернетичного нейрона

Загальні зауваження. Далі розглядатимуться конкретні нейросіткові парадигми, в тому числі і ті, що виникли на початку розвитку нейроінформатики. Адже саме ці сітки стали основою для розвитку кібернетичних НС, і в їх основі лежать перші уявлення, отримані людиною про функціонування мозку. Не дивлячись ні на що, багато які з них і досі успішно застосовуються на практиці.

Для опису будь-якої НС треба визначити для неї основні положення:

- модель формального нейрона (функція перетворення тощо);
- структура зв'язків нейронів у сітці;
- архітектура побудови сіткових сполучень;
- метод навчання сітки.

В подальшому будемо дотримуватись саме такого плану при викладенні навчального матеріалу в підручнику.

Функціональні особливості штучного нейрона. Вищевикладене дозволяє провести деякі паралелі: в біологічному організмі геном складається з певних генів хромосоми, тоді як ШНС складається з певних ШН, тобто властивості штучної нейронної сітки у першу чергу визначаються *базовими штучними нейронами*, а вже далі *структура і архітектура* взаємозв'язків останніх визначає *топологію* ШНС (аналогічно топології геному в організмі).

Історично першою роботою, якою закладений теоретичний фундамент для створення штучних моделей нейронів і ШНС, прийнято вважати опубліковану у 1943 р. статтю Уоррена У. Мак-каллоха і Вальтера Піттса [173]. Головний принцип теорії Мак-каллоха і Піттса полягає у тому, що довільні явища, які відносяться до вищої нервової діяльності, можуть бути проаналізовані і зрозумілі як деяка активність у сітці, що складається з логічних елементів з двома станами (“усе або нічого”). При цьому для будь-якого логічного виразу, задовольняючого цим умовам, може бути знайдена сітка логічних елементів, яка має описувану цим виразом поведінку.

В якості моделі такого логічного елемента, який в подальшому отримав назву “формальний нейрон”, У. Мак-каллохом і В. Піттсом була запропонована схема, наведена на рис. 1.3. З сучасної точки зору, формальний нейрон уявляє собою математичну модель простого процесора, який має декілька входів і один вихід. Вектор *вхідних сигналів* або *вхідних образів* (надходячих через “дендрити”) перетворюється нейроном (блок функції перетворення) у *сигнал на виході*

або *образ на виході* (розповсюджується по “аксону”) з використанням трьох функціональних блоків: локальної пам’яті (синаптичні ваги), блока підсумовування і блока нелінійного перетворення.

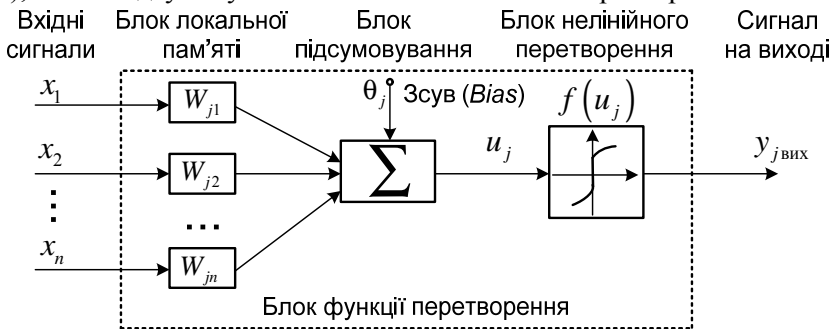


Рис. 1.3. Функціональна схема формального нейрона Маккаллоха і Піттса (пунктирною рамкою виокремлений блок функції перетворення)

Вектор локальної пам’яті містить інформацію про вагові множники W_i , з якими вхідні сигнали будуть інтерпретуватися нейроном. Ці змінні ваги є аналогом чутливості пластичних синаптичних контактів. Підбиранням ваг досягається та або інша інтегральна функція нейрона.

Вхідні сигнали X_i уявляють собою вектори надходжуваних на вхід ШН “властивостей образів”, які подаються з виходів інших ШН.

Блок локальної пам’яті шляхом порозрядного зважування відповідними вагами W_i вхідних сигналів X_i – складових надходжуваних векторів формує вхідну функцію ШН.

Блок підсумовування формує вхідний оператор (вхідну функцію) $f(x, w)$ ШН і задає вид використовуваного у ШН перетворення зважених входів:

- суму зважених входів $f(x, w) = \sum_{i=1}^n x_i w_i$;
- добуток зважених входів $f(x, w) = \prod_{i=1}^n x_i w_i$;
- максимальне значення зважених входів $f(x, w) = \max_i (x_i w_i)$;
- мінімальне значення зважених входів $f(x, w) = \min_i (x_i w_i)$.

Твердження 1.1. Оператор на виході нейрона слугує для подання стану нейрона в бажаній області значень.

Блок нелінійного перетворення – функція активації $f_a(\cdot)$ задає правило переходу ШН, який знаходиться у момент часу τ в стані $k(\tau)$, у новий стан $k(\tau+1)$ при надходженні вхідних сигналів X_i :

$$k(\tau+1) = f_a(k(\tau), f_{\text{вх}}(x, w)).$$

Означення 1.1. Активаційна функція (функція перетворення) – це вхідний оператор нейрона, який задає вид використовуваного в ньому перетворення зважених входів.

Найбільш розповсюджені АФ наводяться далі.

У більшості праць, присвячених проблематиці нейротехнології, сигнал Y на виході ШН визначається як сигнал після оператора активації. Проте при аналізі та синтезі ШНС, які містять усілякі АФ з різними областями значень і області визначень, виникає необхідність у використанні оператора на виході $f_{\text{вих}}$:

$$y_{j\text{вих}} = f\left(\sum_{i=1}^n W_{ji}x_i + \theta_j\right), \quad (1.1)$$

де y_j – сигнал на виході j -го нейрона; f – активаційна (обмежуюча) функція; n – кількість входів; W_{ji} – синаптичні ваги; x_i – вхідні сигнали ($i = \overline{1, n}$); θ_j – зсув (*Bias*) або пороговий сигнал.

Якщо прийняти $\theta_j = W_{j0}x_0$ (зазвичай $x_0=1$), вираз (1.1) приймає вид:

$$y_{j\text{вих}} = f\left(\sum_{i=0}^n W_{ji}x_i\right). \quad (1.2)$$

Слід зауважити, що будь-який формальний нейрон характеризується своєю АФ і зсувом (порогом) θ_j . Перша модель штучного нейрона, запропонована У. Мак-каллохом і В. Піттсом, використовувала тільки бінарні (жорстко обмежені) АФ, за яких сума усіх зважених входів порівнювалася з пороговим значенням θ_j , і якщо вона перевищувала це порогове значення, на виході нейрона встановлювалося логічне 1, а у протилежному випадку – логічне 0. У сучасних застосуваннях порогова функція може бути нелінійною а, відтак, вихід нейрона може приймати як дискретне значення на множині (наприклад, $\{-1, +1\}$), так і неперервне (наприклад, між -1 і $+1$ або у загальному випадку між y_{\min} і y_{\max}).

Отже, елементарна складова сітки – нейрон – має кілька входів і один вихід. Для кращого розуміння функціональних аспектів зобразимо за схемою на рис. 1.4, а структуру моделі навчуваного ШН у вигляді адаптивного нелінійного елемента, який був розроб-

лений Б. Уїдроу і М. Хоффом і дістав назву *Адалін* (Adeline – adaptive linear element [28; 29; 48; 209; 210; 213]).

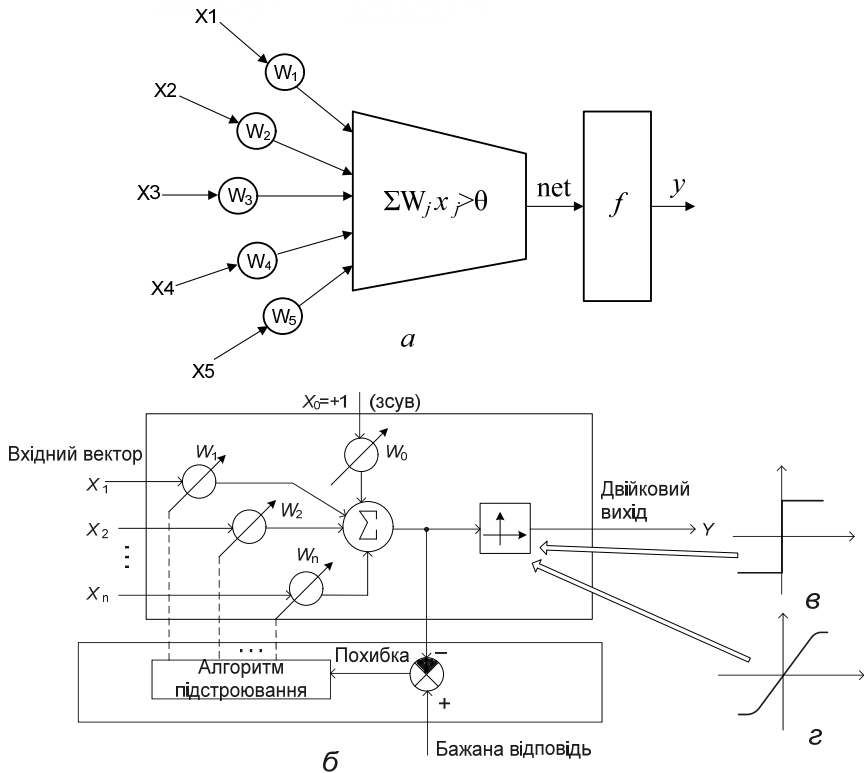


Рис. 1.4. Спрощена структура Адалін (а) (тут $x_1 \dots x_5$ – вхідні сигнали; $W_1 \dots W_5$ – синапси (вагові коефіцієнти); θ – порогова величина; f – активаційна функція; y – вихідний сигнал), уявлення нейрона у вигляді процесорного блоку (б) та варіанти нелінійностей в ньому (в, з)

Адалін складається з двох частин: підналагоджуваного лінійного підсилювача і дворівневого квантувача із жорстко обмеженою або релейною АФ.

Однією з найважливіших властивостей біологічних і штучних нейронів є їх спроможність до навчання, дозволяючи вірно реагувати на навчаючі сигнали шляхом переналагодження синаптичних ваг за певним алгоритмом.

Опишемо фрагментарно алгоритм роботи блоку функції перетворення:

Вхідні сигнали перемножуються на коефіцієнти W_i , (синап-

тичні ваги або просто ваги). Для кожного входу – свій коефіцієнт ваги.

2. Перемножені на ваги входи підсумовуються.
3. Результат підсумовування подається на вхід функції перетворення. Зазвичай $\sup_{x \in R} \left\{ |f(x)| \right\} \leq 1$.
4. Результат функції подається на вихід.

У загальному випадку для відтворення нелінійностей у нейроні можуть застосовуватися не тільки ступінчасті (рис. 1.4, в), а й похилі (рис. 1.4, з) активаційні функції.

Як вже наголошувалося вище, *ШН може приймати лише два стани*: або він збуджений, або пасивний. Тому для цих двох станів досить використовувати двійкові числа, наприклад, числа 0 і 1. Входами для нейрона є збудження, які надходять до нього дендритами. Через те, що ці сигнали є виходами попередніх нейронів, їхні величини також будуть дорівнювати 1 чи 0. Сигнали дендритів перед надходженням до нейрона підсилюються або послабляються. Це легко піддається моделюванню, при якому їх перемножують з додатними або від'ємними дійсними числами: додатні числа призначаються для посилення, а від'ємні – для послаблення сигналів. Коли оброблені в такий спосіб вхідні сигнали досягають нейрона, вони збуджують його лише тоді, якщо сума усіх величин потенціалів перевершуватиме деяке граничне значення.

У моделі ШН формується сума усіх входів, перемножених на значення синапсів, і порівнюється з *граничним значенням* θ . Порогове значення θ (часто вважається дорівнюваним нулю) також зберігається в локальній пам'яті.

Якщо сума перевищує θ , вихідний сигнал нейрона (величина сигналу аксона) встановлюється на 1, в противному випадку – на 0. Формальне подання функцій нейрона наведено на рис. 1.4, а, де x_1, x_2, x_3, x_4, x_5 – вхідні величини; w_1, w_2, w_3, w_4, w_5 – величини синапсів (ваг), якщо сума $\sum w_i x_i > \theta$, то нейрон збуджений, тобто $y = 1$.

При цьому для нейрона справедливі наступні кількісні співвідношення:

- вхідні величини x_1, x_2, \dots, x_a дорівнюють 1 чи 0;
- вагові коефіцієнти w_1, w_2, \dots, w_a – будь-які дійсні числа.

Ці вхідні сигнали, у сукупності позначені вектором \overline{X} , відповідають сигналам, що надходять у біологічні нейрони. Аналогічно

множину ваг позначають вектором \overline{W} . Блок підсумовування (рис. 1.3), що відповідає тілам біологічних елементів (СОМИ), додає зважені входи алгебраїчно, утворюючи вихід, що одержав назву *net*. У векторних позначеннях це має вигляд:

$$net = \overline{XW}.$$

У розглянутому прикладі $y = f(net - \theta)$ – сигнал на виході нейрона, в якому активаційна функція $f(x)$ може, наприклад, приймати значення:

$$f(x) = \begin{cases} 1, & \text{якщо } x > 0 \\ 0, & \text{якщо } x \leq 0 \end{cases}.$$

Тут $net = \sum w_i x_i$ – алгебраїчна сума зважених входів.

Поведінку нейрона з трьома входами можна проілюструвати наступним чином: у дорівнює точно одиниці, якщо входи $1, x_1, x_2, x_3$, перемножені відповідно на ваги θ, w_1, w_2, w_3 , в сумі перевищують 0, тобто

$$\theta \times 1 + w_1 x_1 + w_2 x_2 + w_3 x_3 > 0.$$

Для введення граничного значення $-\theta$ у загальну суму передбачається (див. рис. 1.4, б) додатковий вхід, що завжди дорівнює 1. Цей допоміжний вхід спрощує формалізацію, при якій граничне значення $-\theta$ зараховується до ваг, і опитування нормується на число 0. Цей додатковий вхід зі стандартною величиною 1 і вагою $-\theta$ одержав найменування *Bias* (зсув).

Активаційна функція $f(x)$ може в деталях варіюватися. Вище були відзначені стани нейрона двійковими величинами 0 і 1. Можуть застосовуватися й інші пари. У багатьох випадках зручніше працювати з величинами ± 1 . При цьому двійкові АФ замінюються в такий спосіб:

$$f(x) = \begin{cases} 1, & \text{якщо } x > 0 \\ -1, & \text{якщо } x \leq 0. \end{cases}$$

Наведені на рис. 1.3 та рис. 1.4 моделі формально кібернетичного нейрона зустрічається у багатьох топологіях нейросіток, проте не є догмою. Існують НС і з іншими моделями нейронів, про які детальніше буде йти мова нижче.

Активаційна функція нейрона. Формальні нейрони можуть бути об'єднані в сітки шляхом замикання виходів одних нейронів на

входи інших, і за думкою авторів моделі, така кібернетична система з обґрунтовано вибраними вагами може подавати довільну логічну функцію. Для теоретичного опису отримуваних НС пропонувалася математична мова обчислення логічних предикатів.

Треба зауважити, що сьогодні, через 50 років після роботи Мак-каллоха і Піттса, вичерпної теорії синтезу логічних сіток з довільною функцією, певно, немає. Найбільш просунутими виявилися дослідження в галузі багат шарових систем і сіток з симетричними зв'язками. Більшість моделей спираються у своїй основі на різні модифікації формального нейрона. Важливим розвитком теорії формального нейрона є перехід до аналогових (неперервних) сигналів, а також до всіляких типів нелінійних перехідних – *активаційних функцій* (або функцій перетворення).

Активаційна функція $f(x)$ може в деталях варіюватися. Вище були відзначені стани нейрона двійковими величинами 0 і 1. Можуть застосовуватися й інші пари. У багатьох випадках зручніше працювати з величинами ± 1 . При цьому двійкові АФ замінюються у такий спосіб:

$$f(x) = \begin{cases} 1, & \text{якщо } x > 0 \\ -1, & \text{якщо } x \leq 0. \end{cases}$$

В інтелектуалізованих НС (ІНС) набули застосування різні АФ $f(x)$, зображені на рис. 1.5 [4]:

а) *сигмоїдна функція*:

$$f(x) = \frac{1}{1 + e^{-cx}}.$$

Її застосування дозволяє уникнути проблем, пов'язаних з недиференційованістю АФ, і поліпшує властивості процедур навчання;

б) *функція гіперболічного тангенсу*:

$$f(x) = \text{th}(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}};$$

в) *гаусівська функція*:

$$f(x) = e^{\frac{-(x-c)^2}{2\sigma^2}};$$

г) *інверсна мультікватрична функція*:

$$f(x) = \frac{1}{\sqrt{(x-c)^2 + d^2}};$$

д) *трикутна функція* (для нейро-фаззі-систем):

$$f(x) = \begin{cases} c_1(x - x_0) & \text{для } x \in (x_0, 0) \\ -c_2(x - x_2) & \text{для } x \in (0, x_2) \\ 0 & \text{в інших випадках} \end{cases} ;$$

е) трапецієвидна функція (для нейро-фаззі-систем):

$$f(x) = \begin{cases} c_1(x - x_0) & \text{для } x \in (x_0, x_1) \\ 1 & \text{для } x \in (x_1, x_2) \\ -c_2(x - x_2) & \text{для } x \in (x_2, x_3) \\ 0 & \text{в інших випадках} \end{cases} .$$

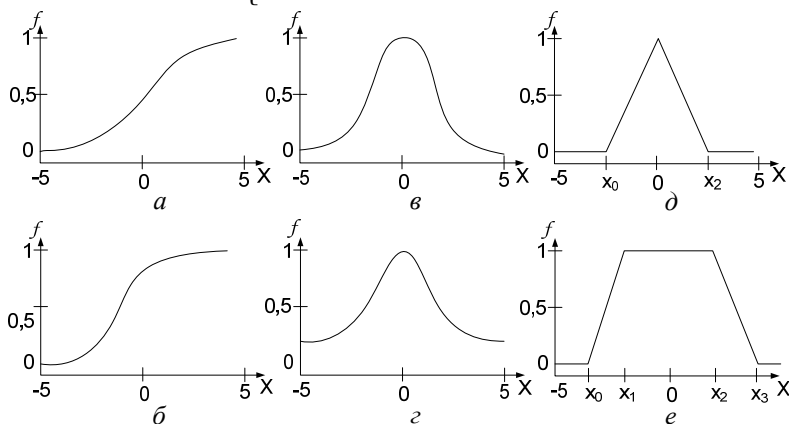


Рис. 1.5. Активаційні функції для різних типів ШНС і нейро-фаззі-систем

Як наголошувалося С. Гроссбергом [125], сигмоїдальна функція володіє вибірковою чутливістю до сигналів різної інтенсивності, що відповідає біологічним даним. Найбільша чутливість спостерігається поблизу порогу, де малі зміни сигналу *net* приводять до відчутних змін виходу. Напроти, до варіацій сигналу в областях значно вище/нижче порогового рівня сигмоїдальна функція не чутлива через те, що її похідна при великих і малих аргументах прямує до нуля.

1.3. Базові моделі штучних нейронів

Означення 1.2: Базова модель ШН – математична та її програмна і/або апаратна реалізації, на основі яких будується ШНС і в яких реалізується *функція перетворення*.

Моделі ШН залежать від прикладних застосувань, через що синтез моделі у кожному конкретному випадку є нетривіальною задачею.

Структури усіх моделей ШН умовно можна розбити на дві принципові групи: *статичні*, які не спроможні моделювати динамічні процеси; *динамічні*, які є альтернативою в цьому сенсі до статичних ШН.

Окрім формального ШН Мак-каллоха і Піттса (рис. 1.3) та Уїдроу і Хоффа (рис. 1.4) існують інші моделі ШН, деякі з них коротко розглянемо нижче. Детальний розгляд ШНС, побудованих на використанні різних моделей ШН, розглядається далі у відповідних розділах підручника.

Модель ШН К. Фукушими. Як і в моделі Мак-каллоха і Піттса в залежності від впливу надходячих сигналів на реакцію ШН, його синаптичні ваги можуть бути додатними, дорівнюваними нулю або від'ємними. Сигнали дістали назву *збуджуючих*, якщо ваги додатні ($w_{ji} > 0$), і *гальмуючих*, якщо ваги від'ємні ($w_{ji} < 0$). Проте у запропонованій К. Фукушимою моделі ШН усі синаптичні ваги і усі сигнали на вході і виході вважаються невід'ємними, тобто вони можуть дорівнювати нулю або приймати будь-яке додатне значення [114-118]. Через це в моделі К. Фукушими входи і відповідні синаптичні ваги розділяються на дві групи: збуджуючі і гальмуючі, причому збуджуючий ефект визначається зваженою сумою усіх збуджуючих входів. Синаптичні ваги зазвичай є змінними, і вони налагоджуються у процесі навчання ШНС.

Модель ШН Дж. Хопфілда. Модель Хопфілда (рис. 1.6) є динамічною структурою ШН і найпопулярнішою динамічною моделлю ШН [143; 144].

Диференціюючий ланцюжок $R_j C_j$ і нелінійний підсилювач $f(u_j)$ з сигмоїдальною АФ і двома асиметричними виходами v_j і $-v_j$, а також моделюючі синаптичні ваги резистори R_{ji} утворюють електричну схему моделі ШН Хопфілда (рис. 1.6, а). Незалежний зовнішній вхідний сигнал – струм I_j уявляє собою зсув (Bias), а реалізація додатної і від'ємної синаптичних ваг здійснюється з'єднаннями резисторів R_{ji} з $+v_i$ або з $-v_i$, відповідно.

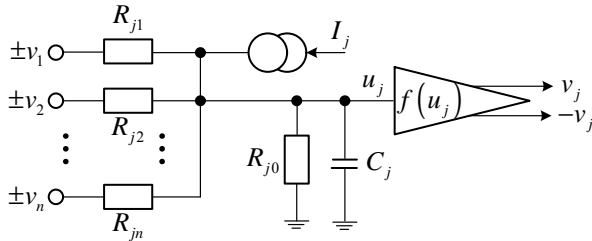
Диференційне рівняння, яким описується модель ШН за рис. 1.6, а, можна отримати, залучаючи закон Кірхгофа [48]:

$$C_j \frac{du_j}{d\tau_j} = \frac{u_j}{R_j} + \sum_{i=1}^n \frac{v_i}{R_{ji}} + I_j, \quad (1.3)$$

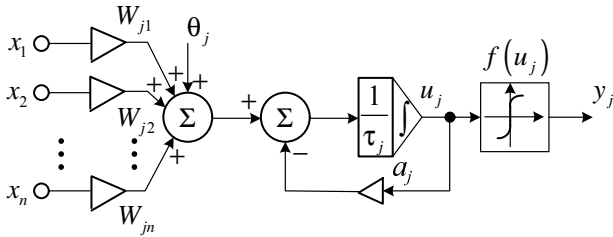
де τ_j – простийна інтегрування;

$$\frac{1}{R_j} = \frac{1}{R_{j0}} + \sum_{i=1}^n \frac{1}{R_{ji}} = H_0 + \sum_{i=1}^n H_{ji}; \quad i = \overline{1, n}, \quad j = \overline{1, n}; \quad H_{ji} - \text{визначає елек-}$$

тропровідності; $v_j = f(u_j)$.



а



б

Рис. 1.6. Електрична модель (а) динамічної нейронної клітини і структура (б) штучного нейрона Хопфілда

Для переходу до функціональної структури ШН (рис. 1.6, б) перепишемо (1.3) у більш загальній формі

$$L_{0j} \frac{du_j}{d\tau_j} = -a_j u_j + \left(\sum_{i=1}^n W_{ji} x_i + \theta_j \right), \quad (1.4)$$

де $y_j = f(u_j)$; u_j – внутрішній сигнал або внутрішній потенціал дії; $a_j = r_j / R_j$ – коефіцієнт загасання; $W_{ji} = \pm r_j / R_{ji}$ – синаптичні ваги з додатнім знаком, якщо R_{ji} з'єднаний з $(+x)$, і з від'ємним знаком якщо R_{ji} з'єднаний з $(-x)$; x_i – входні сигнали; $i = \overline{1, n}$; $\theta_j = r_j / I_j$ – сигнал зсуву (Bias).

Динаміка ШН у цілому визначається диференціальним ланцюжком $R_{ji}C_j$. Синаптичні ваги визначаються входними електропровід-

ностями $H_{ji} = 1/R_{ji}$, з'єднаними з одним із виходів $+v$ або $-v$ для j -го підсилювача, причому для додатних синаптичних ваг ($W_{ji} > 0$) необхідно з'єднання резистора R_{ji} з додатним, а від'ємних ваг ($W_{ji} < 0$) – із від'ємним виходом j -го підсилювача.

Модель ШН С. Гроссберга. Ідеї, закладені в дослідженнях Стефана Гроссберга на зорі біологічної кібернетики, покладені в основу багатьох наступних нейросіткових напрацювань. Тому доцільно розпочати розгляд ієрархічних архітектур ШНС з конфігурацій вхідних і вихідних зірок Гроссберга [121-126].

Зазвичай, нейрон у формі вхідної зірки має N входів $x_1 \dots x_N$, яким відповідають ваги $w_1 \dots w_N$, і один вихід Y , який уявляє собою зважену суму входів. Вхідна зірка навчається видавати сигнал на виході кожного разу, коли на входи надходить певний вектор. Таким чином, вхідна зірка є детектором сукупного стану своїх входів. Процес навчання подається в наступною ітераційною формою:

$$w_i(t+1) = w_i(t) + \beta(x_i - w_i(t)).$$

Темп навчання β має початкове значення масштабу $0 \dots 1$ і поступово зменшується в процесі навчання. При настроюванні нейрон вчиться усередненим навчаючим векторам.

Особливість нейронів у вигляді зірок Гроссберга полягає у локальній пам'яті: кожний нейрон у вигляді *вхідної зірки Гроссберга* пам'ятає "свій" образ та ігнорує всі інші, тоді як *зірка Гроссберга на виході* виконує протилежну функцію – функцію командного нейрона, видаючи на виходах певний вектор при надходженні сигналу на вхід. Тобто образ пам'яті пов'язується з певним нейроном, а не виникає внаслідок взаємодії множини нейронів у сітці. Цей тип нейрона має один вхід і M виходів з вагами $w_1 \dots w_M$, які навчаються за формулою:

$$w_i(t+1) = w_i(t) + \beta(Y_i - w_i(t)).$$

Рекомендується починати зміну параметра швидкості навчання β з порядку одиниці і поступово зменшувати до нуля в процесі навчання. Ітераційний процес буде збігатися до збирального образу, отриманого з сукупності навчаючих векторів.

Штучний нейрон С. Гроссберга [123] є узагальненням моделі Дж. Хопфілда і описується наступним диференціальним рівнянням [48]:

$$\tau_j \frac{du_j}{d\tau} = -a_j u_j + (\chi_j - \rho_j u_j) \left(\sum_{i=1}^n W_{ji} f_i(u_i) + \theta \right), \quad (1.5)$$

де u_j – внутрішня активність j -го нейрона; a_j, χ_j, ρ_j – динамічні константи нейрона.

На рис. 1.7 наведена спрощена функціональна модель ШН, яка відповідає виразу (1.5).

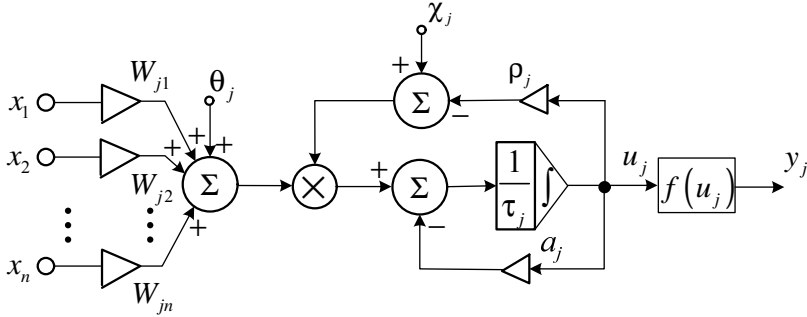


Рис. 1.7. Спрощена функціональна модель ШН С. Гроссберга

Нейрон може знаходитись в двох станах: збудженому і загальмованому, які описуються відповідними рівняннями:

$$\begin{aligned} \tau_j \frac{du_j}{d\tau} = & -a_j u_j + (\chi_{jзб} - \rho_{jзб} u_j) \left(\sum_{i=1}^{n_{зб}} W_{jiзб} f_{iзб}(u_i) + \theta_{jзб} \right) - \\ & - (\chi_{jгальм} - \rho_{jгальм} u_j) \left(\sum_{i=1}^{n_{гальм}} W_{jiгальм} f_{iгальм}(u_i) + \theta_{jгальм} \right). \end{aligned} \quad (1.6)$$

Сигма-пі-нейрон. Щоб більш точно відбивати властивості біологічного нейрона, а саме, спроможність моделювати певні синаптичні контакти, удаються до використання більш складного, аніж у лінійній моделі (1.1), перетворення вхідних сигналів нейрона – їх добутку або кореляції [182; 183]. Сигнал на виході $y_{вих}$ формується так само, як і у випадку ШН моделі У. Мак-каллока і В. Пітса:

$$y_{вих} = f(W, x) = f \left(\sum_{j=1}^m W_j \prod_{i=1}^n x_i \right),$$

де $f(*)$ – функція активації (найчастіше порогова).

Спрощена схема сигма-пі-нейрона наведена на рис. 1.8.

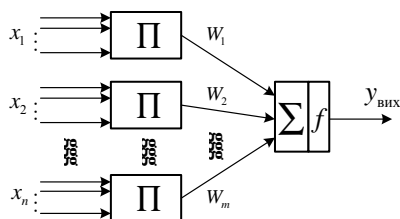


Рис. 1.8. Спрощена схема моделі сигма-пі-нейрона

1.4. Математична інтерпретація штучних нейронних сіток

Узагальнені топології ШНС. Хоча можливості розглянутих моделей ШН обмежені, штучні нейронні сітки, побудовані на їх основі володіють практично необмеженими можливостями. Штучні нейронні сітки відображають розвиток моделей, які з'явилися завдяки спробам імітування механізму мислення людини, що вже з огляду на цю обставину є формальними моделями подання знань. Для певних класів задач, що стосуються *штучного інтелекту* (ШІ) (насамперед розпізнавання, інтелектуального управління динамікою складних механічних систем тощо), нейросіткові моделі часто забезпечують більш ефективні рішення, ніж традиційні символічні підходи.

Нейронні сітки дозволяють створювати ефективне програмне забезпечення для високопаралельних комп'ютерів, де виникає відома проблема: як досягти того, щоб всі елементи одночасно і без зайвого дублювання обчислювали щось корисне? Створюючи математичне забезпечення на основі НС, можна для широкого кола задач розв'язати цю проблему.

Якщо ж перейти до більш прозаїчного рівня повсякденної роботи, ШНС – це лише сітки, які складаються з пов'язаних між собою простих елементів – формальних нейронів. Значна більшість праць з НІ присвячена перенесенню всьляких алгоритмів розв'язання задач на такі сітки.

Ядром використовуваних при цьому уявлень є ідея про те, що нейрони можна моделювати досить простими автоматами, а вся складність мозку, гнучкість його функціонування та інші найважливіші якості визначаються зв'язками між нейронами. При цьому кожний зв'язок подається як зовсім простий елемент, що слугує для передавання сигналу. Одночасно уявляється, що система зв'язків достатньо багата за своїми можливостями і досить надлишкова, щоб компенсувати бідність вибору елементів, їх ненадійність, можливі руйнування частини зв'язків.

Отже, НС є сукупністю нейронів і зв'язків між ними. Кожний нейрон виконує відносно просту функцію, але, будучи об'єднаними

численними зв'язками в єдину структуру, всі вони разом утворюють систему, здатну розв'язувати складні задачі.

Вирізняють два типи штучних нейроподібних сіток. У *сітках першого типу* вузлами є формальні елементи, що описують окремі нейрони. В *сітках другого типу* вузлами є формальні елементи, що відповідають не окремим нейронам, а особливим сукупностям їх – нейронним ансамблям.

Означення 1.3. Під *нейронним ансамблем* розуміють таку сукупність взаємозв'язаних нейронів, яка збуджується повністю при збудженні окремої її частини.

Багато дослідників вважають, що саме нейронний ансамбль, а не окремий нейрон, і є функціональною одиницею мозку як системи, що забезпечує складну діяльність людини як живої істоти.

Як об'єкт модельного опису нейронний ансамбль відрізняється від окремого нейрона двома основними особливостями. *Одна з них* полягає в тому, що початкове збудження ансамблю змінюється неперервно, а не за законом “так – ні”. Нейронний ансамбль можна описати як нелінійний перетворювач аналогової інформації, який задається набором певних статичних і динамічних характеристик.

Інша особливість нейронного ансамблю полягає в тому, що він може бути поставлений у відповідність до деякої змістовної одиниці (поняття, образу тощо), тобто елементу, який бере участь у процесі розумової діяльності людини.

Отже, нейроподібна сітка, вузли якої відповідають ансамблям, стає сіткою із семантикою, або семантичною сіткою (СС). Ця обставина значно змінює підхід до проблем синтезу сіток і змістової інтерпретації процесів, що в них відбуваються.

У найзагальніших рисах результатом функціонування нейронної сітки є сигнал, який ідентифікує належність вхідного зображення до одного з кількох класів. Завдяки властивості навчатися сітка “зберігає” ці зображення у своїй “пам'яті” (тобто сітка здійснює категоризацію зображень).

Загальна характеристика архітектури ШНС. В загальному випадку ШНС складається з декількох шарів, серед яких обов'язково є *вхідний*, отримуючий зовнішні сигнали, *шар на виході*, відбиваючий реакцію нейронів на комбінації вхідних сигналів, і у багатошарових ШНС – *сховані*. Така пошарова організація є аналогом шаруватих структур відповідних відділень мозку.

Нейронна сітка утворюється шляхом послідовного чи паралельного сполучення ряду нейронів. Так, наприклад, на рис. 1.9, а

представлена найпростіша НС з 2 нейронів.

На вхід сітки подаються сигнали x_1, x_2 , на виході формуються величини y_1, y_2 . Ваги (значення синапсів) включені в зв'язуючі лінії. При відсутності *Bias* (зсуву) опитування при двійковій АФ орієнтоване на $\theta=0$. При розрахунку кожній парі входів x_1, x_2 відповідає одна пара виходів y_1, y_2 . Через те, що усі величини двійкові (0 чи 1), одержуємо логічну функцію, передану по сітці.

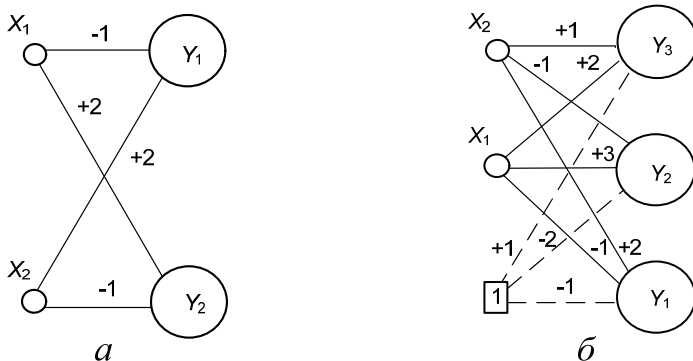


Рис. 1.9. Нейронні сітки: а – найпростіша сітка; б – сітка зі зсувом

Аналітичні вирази для розрахунку ШНС (рис. 1.9, а) мають вигляд:

$$\left. \begin{aligned} y_1 &= f(net_1 - 0), \\ y_2 &= f(net_2 - 0), \\ net_1 &= -1x_1 + 2x_2, \\ net_2 &= 2x_1 - 1x_2 \end{aligned} \right\},$$

де f – двійкова функція.

Задаючи для входів x_1, x_2 будь-які можливі величини і з'єднуючи їх згідно з наступною умовою з відповідними величинами y_1, y_2 на виході, одержуємо *табличну форму* подання ШНС:

x_1	x_2	y_1	y_2
0	0	0	0
1	0	0	1
0	1	1	0
1	1	1	1

Зв'язки між нейронами задаються у виді векторів і матриць. Ваги зручно подавати елементами матриці $W = [w_{ij}]$ розмірності $N \times M$, де N – кількість входів, M – кількість нейронів. Елемент w_{ij} відбиває зв'язок між i -м та j -м нейронами. При цьому, якщо

$w_{ij} = 0$ – зв'язок між i -м та j -м нейронами відсутній;

$w_{ij} < 0$ – гальмуючий сигнал зв'язок;

$w_{ij} > 0$ – прискорюючий сигнал (збуджуючий) зв'язок.

Тоді розглянута на рис. 1,9, а сітка являє собою логічну двійкову вектор-функцію. Вводячи *векторну форму* подання ШНС, запишемо:

$$\overline{X} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \overline{Y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

і матрицю

$$\overline{M} = \begin{pmatrix} -1 & 2 \\ 2 & -1 \end{pmatrix},$$

Одержуємо

$$\overline{net} = \overline{M} \overline{X},$$

$$\overline{y} = \overline{f}(\overline{net} - \overline{0}),$$

де $\overline{0}$ – нуль-вектор, \overline{net} – вектор сумарного імпульсу та

$$\overline{f}(\dots) = \begin{pmatrix} f(net_1 - 0) \\ f(net_2 - 0) \end{pmatrix}.$$

У розглянутому прикладі всі граничні величини $\theta=0$. У протилежному разі граничні величини відображаються через зсуви (*Bias*), як це показано на рис. 1.9, б. Цій нейронній сітці відповідає така процедура розрахунку:

$$\left. \begin{aligned} y_1 &= f(-1x_1 + 2x_2 - 1) \\ y_2 &= f(3x_1 - 1x_2 - 2) \\ y_3 &= f(2x_1 + 1x_2 + 1) \end{aligned} \right\}.$$

Відповідно у векторній формі

$$\overline{y} = \overline{f}(\overline{net} - \overline{\theta}) = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}; \quad \overline{net} = \begin{pmatrix} -1 & 2 \\ 3 & -1 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}; \quad \overline{\theta} = \begin{pmatrix} -1 \\ -2 \\ 1 \end{pmatrix}.$$

Таким чином, вхідний вектор \bar{X} двовірний, а вихідний вектор \bar{Y} так само, як і граничний вектор $\bar{\theta}$, є тривимірним, і матриця має вигляд "2 x 3". Легко побачити, що сіткою (рис. 1.9, б) задається така функція:

x_1	x_2	y_1	y_2	y_3
0	0	0	0	1
1	0	0	1	1
0	1	1	0	1
1	1	0	0	1

На основі цих прикладів можна пояснити поняття "нейронна сітка":

Твердження 1.2. Одношарова нейронна сітка задається за допомогою $n \times t$ матриці \bar{M} , елементами якої є дійсні числа, а також векторною активаційною функцією \bar{f} так, що кожному двійковому імпульс-вектору \bar{X} відповідає один вихідний вектор \bar{Y} і відповідна умова $\bar{y} = f(\bar{net} - \bar{\theta})$, де $\bar{\theta}$ – вектор граничної величини, $\bar{net} = \bar{M} \bar{X}$.

Отже, будь-яка НС може бути подана графом, у якому кожна вхідна величина зв'язана з кожною вихідною. На лініях зв'язку вказуються відповідні ваги (елементи матриці). Граничні величини можуть при цьому вноситися в граф за допомогою зсувів. Зазначимо, що нейрони лівого ряду (рис. 1.9) виконують функції розподілу вхідних сигналів і не виконують які-небудь обчислення і тому не вважаються шаром. Структури (рис. 1.9) являють собою одношарові ШНС.

Загальна характеристика структури ШНС. Прямоспрямовані НС. Вирізняють два основних класи сіток: сітки зі зворотним зв'язком, у яких сигнал на виході подається назад на вхід сітки, і сітки, у яких для кожного вхідного вектора розраховується вектор на виході, що зчитується з нейронів на виході. Тут відсутні зворотні зв'язки, й у цьому сенсі відбувається прямоспрямований потік інформації – такі сітки дістали назву *прямоспрямованих нейронних сіток* (ПНС). Сітки з такою структурою називають також "feed forward Networks" чи "hetero assotiative Networks".

Розглянуті одношарові сітки подаємо в матричній формі

$$\bar{y} = f(\bar{net} - \bar{\theta}),$$

де $\bar{net} = \bar{M} \bar{X}$; \bar{X} і \bar{Y} – вхідний і вихідний вектори; f – активаційна функція.

У 1969р. М. Мінський і С. Паперт [171; 172] показали, що є важливі логічні функції висловлювань, які не піддаються опису у формалізмах зазначених одношарових типів сіток. Однією з таких функцій є, наприклад, буліва *XOR-функція* (*Exclusiven Order* – "виключає АБО"). Однак через 10 років було доведено, що положення Мінські і Паперт втрачає силу стосовно до багатошарових сіток, коли один вихідний шар є входним для наступного нейронного шару. Якщо ці шари кілька разів включаються один за одним, виходить багатошаровий тип нейронної сітки (рис. 1.10, а, б) – *багатошаровий перцептрон MLP* (*Multilayer-Perceptron-Networks*).

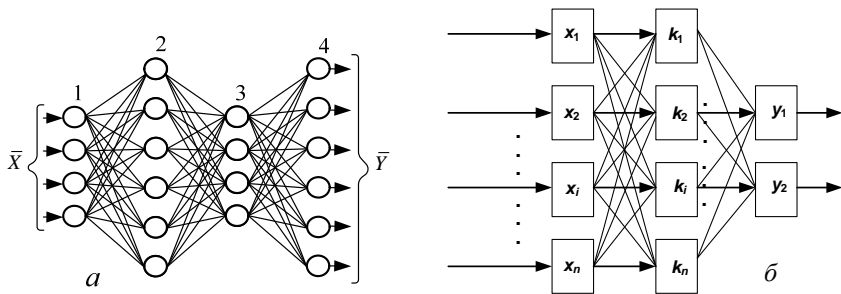


Рис. 1.10. Структура MLP (а) (де \bar{x} – вектор вхідних сигналів; \bar{y} – вектор вихідних сигналів; 1 – вхідний шар нейронів – розподільників вхідних сигналів; 2, 3 – сховані (проміжні) шари ІНС; 4 – вихідний шар нейронів) та тришарового (б) перцептрона

Ці різновиди сіток можуть являти собою усілякі функції, і для них з 1985 р. використовується прийнятний алгоритм навчання (*Backpropagation Algorithmus*).

Розглянемо спочатку (рис. 1.11, а, б) топологію *тришарової прямокутної нейронної сітки*. Тут (рис. 1.11, а) середній шар служить входом для наступного за ним шару чи нейронів, що у свою чергу являє собою вихід усієї нейронної сітки. Перший шар (вхідний), як уже було зазначено, використовується лише для розподілу вхідних сигналів. Він не виконує якихось обчислень, і його часто не вважають окремим шаром. У цій сітці всі граничні величини дорівнюють нулю (відсутній *Bias*). Вхідними величинами є x_1, x_2, x_3 . На виході нижнього шару формуються величини k_1, k_2 , і ці величини потім як вхідні надходять до верхнього шару. Цей верхній шар формує остаточні вихідні величини сітки y_1, y_2, y_3 . Проміжні нейрони за-

пам'ятовують допоміжні величини k_1, k_2 . Їх називають *схованими нейронами* (*hidden neurons*), а весь середній шар – *схованим шаром* (*hidden layer*).

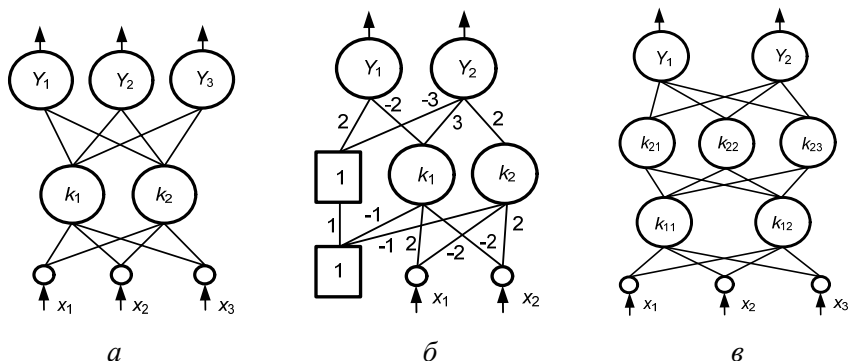


Рис 1.11. Варіанти архітектури прямоспрямованих нейронних сіток: *а* – тришарова ІНС з двома схованими нейронами; *б* – тришарова ІНС зі зсувами у вхідному і схованому шарах; *в* – нейронна сітка з двома схованими шарами

З математичної точки зору маємо нелінійну комбінацію. Важливо, що для визначення завантаження схованих нейронів застосовується АФ, тому що в протилежному разі весь опис звівся б до перемножування двох матриць, і тоді всю сітку можна було б звести до одношарової.

Для багатошарової сітки застосовано такі ж формалізми, як і для одношарової. Розглянемо тришарову сітку (рис. 1.11, *а*)

$$\bar{k} = \bar{f}(\overline{net_1} - \bar{\theta}_1),$$

де $\overline{net_1} = \overline{W_1}x$. Тут \bar{k} , \bar{x} і $\bar{\theta}_1$ – вектори відповідно схованого шару, вхідний, граничний; net_1 розраховується за матрицею $\overline{W_1}$.

Другий шар описується співвідношенням:

$$\bar{Y} = \bar{f}(\overline{net_2} - \bar{\theta}_2),$$

де $\overline{net_2} = \overline{W_2}\bar{k}$; \bar{Y} – вихідний вектор; $\overline{W_2}$ – вагова матриця другого шару.

Для пояснення розглянемо приклад (рис. 1.11, *б*), де для кожного шару додатково до ваг введено по *Bias*'у (зсуву).

Розрахувавши величини схованого шару і вихідні значення, одержимо:

$$\left. \begin{aligned} k_1 &= f(2x_1 - 2x_2 - 1) \\ k_2 &= f(-2x_1 + 2x_2 - 1) \\ y_1 &= f(-2k_1 + 0 \times k_2 + 2) \\ y_2 &= f(3k_1 + 2k_2 - 3) \end{aligned} \right\}$$

У векторній формі це можна подати так:

$$\begin{pmatrix} 1 \\ k_1 \\ k_2 \end{pmatrix} = f \left[\begin{pmatrix} 1 & 0 & 0 \\ -1 & 2 & -2 \\ -1 & -2 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} \right]; \quad \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = f \left[\begin{pmatrix} 2 & -2 & 0 \\ -3 & 3 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ k_1 \\ k_2 \end{pmatrix} \right]$$

Причому

$$\bar{k} = f(W_1 \bar{x}); \quad \bar{y} = f(W_2 \bar{k}), \quad (1.7)$$

де \bar{x} – розширений на число 1 вхідний вектор, а W_1 і W_2 – матриці.

Як бачимо, *Bias* може бути інтегрований простим розширенням вектора на координату 1, завдяки чому надалі розглядаються лише матриці, в яких не враховуються граничні вектори.

Простим перерахуванням можна визначити, що сітка (рис.1.11, б) подає такі функції (додано сховані нейрони k_1, k_2):

x_1	x_2	k_1	k_2	y_1	y_2
0	0	0	0	1	0
0	1	0	1	1	0
1	0	1	0	0	0
1	1	0	0	1	0

В загальному вигляді дворівневу сітку можна подати у формі (1.7), причому тепер: \bar{x} – вхідний вектор; \bar{k} – вектор схованого шару; \bar{y} – вектор вихідного сигналу. Можлива наявність *Bias*’у заздалегідь інтегрована у вхідний вектор, тобто передбачається, що нейрон входу x_1 як *Bias* постійно встановлений на 1, і відповідні елементи матриць W_1, W_2 мають ті самі пороги.

З приведених рівнянь одержуємо:

$$\bar{y} = f(W_2(f(W_1 \bar{x}))).$$

За аналогією можна визначити тришарову сітку

$$\bar{y} = f(W_1 f(W_2 f(W_3 \bar{x}))),$$

або n -шарову сітку

$$\bar{y} = f(W_1 f(W_2 f(W_3 \dots f(W_n \bar{x}))))).$$

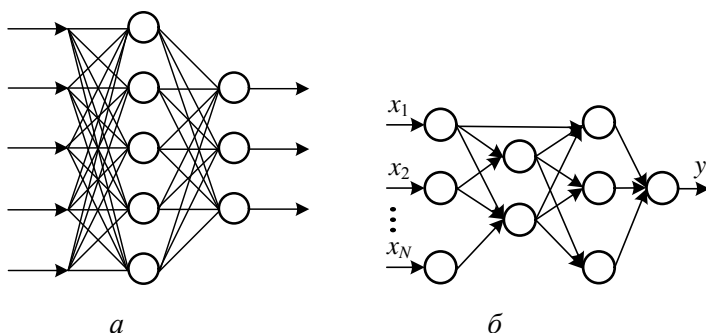


Рис. 1.12. Повнозв'язні ШНС прямого поширення: *а* – першого порядку; *б* – другого порядку

На рис. 1.11, *в* показано топологію ІНС з двома схованими шарами, побудовану за описаним вище принципом.

Отже, загалом ШНС прямого поширення припускають наявність декількох шарів зі зв'язками між нейронами різних шарів. У сітках *першого порядку* існують тільки зв'язки між двома сусідніми шарами, тобто між i -м та $(i+1)$ -м шарами. В цьому випадку кажуть, що зв'язки ШНС пошарові (наприклад, варіанти архітектури сіток на рис. 1.11). І якщо у сітці даного типу кожен нейрон i -го шару зв'язаний з кожним нейроном $(i+1)$ -го шару, сітка називається *повнозв'язаною прямого поширення* (рис. 1.12, *а*).

У сітках *другого порядку* поряд із зв'язками між нейронами сусідніх (i -го та $(i+1)$ -го) шарів присутні й зв'язки між нейронами шарів i -го та $(i+m)$ -го, де $m > 1$. Приклад такої ШНС наведений на рис. 1.12, *б*.

Нейронні сітки зворотного поширення. Сітки даного типу припускають наявність зворотних зв'язків як між нейронами окремих шарів, так і між нейронами одного шару. Використання ШНС зі зворотними зв'язками необхідне для вивчення складних динамічних об'єктів, які змінюють свій стан при надходженні нових вхідних сигналів. Такі ШНС можуть володіти властивостями, подібними до короткочасної людської пам'яті.

В ШНС з *прямими зворотними зв'язками* на вхід нейрона деякого i -го шару подається сигнал з його виходу (рис. 1.13, *а*), тобто даний нейрон підсилює або послабляє сигнал, перетворений його АФ, завдяки чому досягається його граничний активаційний стан. Охоплені зворотним зв'язком нейрони у таких ШНС дістали назву *локально-зв'язаними*.

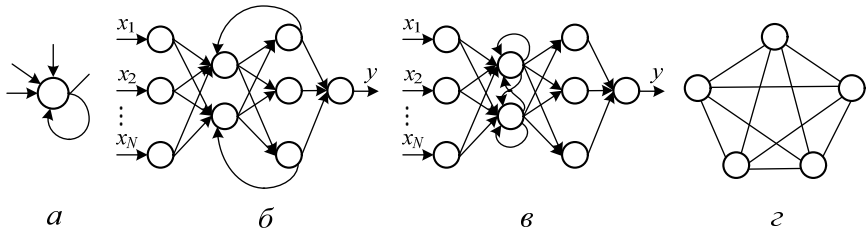


Рис. 1.13. ШНС зворотного поширення: *а* – з локально-зв’язаними нейронами; *б* – з непрямыми зворотними зв’язками; *в* – з латеральними зв’язками; *г* – повнозв’язана

У ШНС з *непрямыми зворотними зв’язками* існують зв’язки нейрона i -го шару з нейронами $(i-k)$ -го шару, де $k > 0$. При цьому одночасно можуть існувати прямі зв’язки цього ж нейрона з нейроном $(i+m)$ -го, де $m > 0$. Уведення таких зворотних зв’язків необхідне, щоб виокремити визначену особливо важливу для даної ШНС область вхідних сигналів.

У ШНС з *латеральними зв’язками* присутні зв’язки між нейронами одного шару (рис. 1.13, *в*). Такий варіант зворотних зв’язків використовується у тому випадку, якщо тільки один нейрон з даної групи нейронів має бути активним. У цьому випадку на вхід кожного нейрона надходять *гальмівний* (послаблюючий або інгібіторний) сигнал від інших нейронів і зазвичай *збуджуючий* (підсилюючий або *ексгібіторний*) сигнал власного зворотного зв’язку. Нейрон з найвищою активністю (переможець) пригальмовує інші нейрони. Тому дану топологію ще називають топологією сітки “*переможець отримує все*” (*the winner takes all-Net*).

Нарешті, *повнозв’язні* ШНС характеризуються наявністю зв’язків між усіма нейронами сітки (рис. 1.13, *г*). Прикладом такої ШНС може бути *сітка Хопфілда* (див. розд. 6.7).

1.5. Нейросітки вищого порядку

Нейросітки вищого порядку, або функціонально-пов’язані сітки були розроблені Йох-Хан Пао і детально описані в його книзі [213]. Нейросітка розширює стандартну архітектуру багатозарового перцептрона з *FeedForward* поширенням та методом *Back-Propagation* модифікацією вузлів на вхідному шарі. Входи комбінуються математичним шляхом за допомогою функцій вищого порядку, таких як квадрати, куби або синуси і розширюють сприйняття сіткою

заданої проблеми. Із назв цих функцій вищого порядку або функціонально пов'язаних входів і впливає назва нейросіток.

Існує два основні способи додавання вхідних вузлів. В *першому* у модель можуть додаватись *перехресні добутки вхідних елементів*, це називається “вхідним добутком” або “тензорною моделлю”, де кожна компонента вхідного образу перемножується зі всіма компонентами вхідного вектора. Наприклад, для сітки *BackPropagation* з трьома входами (A , B і C) перехресними добутками будуть AA , BB , CC , AB , AC та BC (елементи другого порядку). Також можуть додаватись елементи третього порядку, такі як ABC .

Другим методом як додавання до вхідних вузлів слугують *функціональні розширення базових входів*. У випадку моделі з входами A , B і C її можна перетворити у модель нейронної сітки вищого порядку зі входами: A , B , C , $\sin(A)$, $\cos(B)$, $\log(C)$, $\max(A, B, C)$ та ін. Повний ефект повинна забезпечити сітка з об'єднанням моделі тензорного та функціонального розширення.

Ніякої нової інформації не додається, але розширене представлення входів робить сітку простішою для навчання. Існують обмеження для цієї моделі: зокрема, для перетворення початкових входів необхідна обробка більшої кількості вхідних вузлів, що впливає на швидкодію сітки.

1.6. Штучна нейронна сітка як модель подання знань

Уявлення про взаємозв'язки інформації та знань. Як уже зазначалось, міжнародне товариство вступило в стадію інформатизації, що ґрунтується на двох принципово відмінних складових: *інформації та знаннях*, які є базисними підвалинами проблематики штучного інтелекту (ШІ), до якої як одна з моделей подання знань належить і нейротехнологія.

За умов істинності, вірогідності відбиття фактів інформація вважається *первинним знанням* (рис.1.14). В свою чергу, утворене за допомогою механізму логічного виведення (МЛВ) з використанням законів логіки *вторинне знання* (в подальшому називається *виведенням*, або *висновком*) також є інформацією [78].

Проте *інформація* як відбиття дійсності в принципі *не може бути утвореною* (можуть лише змінюватися спосіб і форма її відбиття), тоді як *знання* завдяки законам логіки та застосовуваним засобам (зокрема, природному та/або штучному інтелекту) саме *утворюється*. Інформація підвладна старінню, може бути суб'єктивною, недостатньою (або дезінформацією). Знання ж як істинне відбиття дійс-

ності в мисленні людини завжди об'єктивне й вірогідне. Ці взаємозв'язки інформації та знання і лежать в основі підходу до створення штучних інтелектуалізованих систем (ШІС).

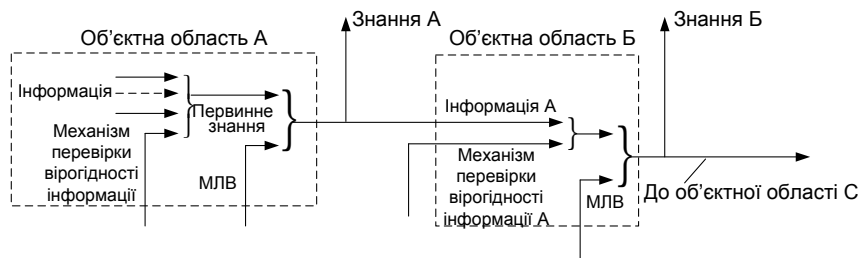


Рис 1.14. Уявлення про взаємозв'язки інформації та знань

У [58] знання визначається як “особлива форма духовного засвоєння результатів пізнання, процесу відображення дійсності, яка характеризується усвідомленням їх істинності”. З точки зору ШІ в інженерії знань це означення необхідно пов'язати з виведенням, і тоді більш привабливим є таке означення:

Означення 1.4. Знання – це формалізована перевірена інформація, на яку посилаються чи використовують в процесі виведення.

Якщо ця процедура виконується програмним шляхом, то знання – це обов'язково інформація, подана в певній формі. Якщо ж розглядати знання з точки зору розв'язування задач у деякій предметній галузі, то їх зручно поділити на дві великі категорії – *факти* та *евристику*.

Перша категорія вказує звичайно на добре відомі в даній предметній галузі обставини, тому знання цієї категорії іноді називають тентовими, маючи на увазі їх достатню освітленість у спеціальній літературі чи підручниках. Друга категорія знань ґрунтується на власному досвіді спеціаліста даної предметної галузі – експерта, накопиченому завдяки багаторічній практиці. В ШІС саме ця категорія знань відіграє вирішальну роль у підвищенні ефективності системи, оскільки до цієї категорії входять такі знання, що потребують високого інтелекту для проведення аналізу інформації, яка надходить від об'єкта досліджень, а саме, способи: зосередження, вилучення некорисних ідей, використання нечіткої інформації та ін.

Крім цих двох категорій знань, виділяють ще *метазнання* (знання про знання) та *правила* (знання про прийняття рішень).

Отже, під фактами розуміють знання типу «А це А», і вони є характерними для БД і сіткових моделей. Метазнання вказують на

знання стосовно способів використання знань, а також знання стосовно властивостей знань (що необхідно для управління базами знань (БЗ)) і виведення, ототожнення, навчання тощо. Зрештою, під правилами розуміють знання типу «ЯКЩО – ТО».

Таким чином, знання, якими заповнюється ШПС, містять знання *першого роду* – *декларативні* (загальновідомі факти, явища, закономірності-істини, визнані в даній предметній галузі й зафіксовані в книгах, статтях, довідниках тощо чи утворені іншими інтелектуальними системами знань завдяки виведенням після оброблення інформації) та знання *другого роду* – *процедурні* (емпіричні правила, інтуїтивні міркування й факти, що, як правило, не публікуються, але дають змогу досвідченому експерту навіть в умовах неповноти їх і суперечливості ефективно приймати рішення).

Успіх реалізації ШПС тим більший, чим вища питома вага знань першого роду. Якщо переважають знання другого роду, то деяким розв'язанням труднощів у ситуаціях, коли експерт не може чітко сформулювати правила прийняття рішень або не хоче передавати кому-небудь свої знання, є використання підсистеми накопичення знань. При цьому в автоматичному чи напіваавтоматичному режимі формуються емпіричні залежності з неповних даних, і підсистема накопичення знань повинна зі знань другого роду побудувати знання першого роду, створивши теорію, а потім вивести вірогідні дані. Знання звичайно мають класифікаційну ієрархічну структуру вигляду «джерело інформації – книга – навчальна література – підручник – інформатика». Кожен з елементів такої структури має різні зв'язки з елементами інших ієрархічних структур, тому є потреба в поданні всіх знань у вигляді сіток. Цей підхід характерний не тільки для декларативних, а й для процедурних знань.

Зазначимо, що знання можуть біти *неповними* (коли для доведення або спростування певного твердження не вистачає інформації), *недостовірними* (наприклад, через випадковість чинників, що впливають, через об'єктивну невизначеність), *нечіткими* (через неточну визначеність понять, якими подаються знання: наприклад, “пору́ч”, “далеко”, “важкий”, “великий” тощо).

Для створення можливості використання знань має бути вирішена проблема здобуття знань, або в більш широкому розумінні – проблема навчання. Саме з цим пов'язане добування інформації з навколишнього середовища та її накопичення в пам'яті з метою дальшого використання.

Зазначена проблема має три взаємопов'язані аспекти: функції того, хто навчається; властивості середовища – джерела інформації; взаємодія того, хто навчається, з середовищем. Як наслідок, при навчанні ставляться принаймні дві мети: добування інформації з середовища та підвищення її рівня й структурування. Доки не вивчена вся інформація, вона не може стати знанням, і дані нижчого рівня, наприклад, початкові дані, можуть бути використаними на цьому рівні. З підвищенням рівня даних універсальність їх у плані застосування зростає. Друга мета забезпечується різними засобами.

Виходячи з розглянутих вище тлумачень інформації та знання, можна сформулювати концепцію інформатизації:

Означення 1.5. *Інформатизація* – це процес максимізації використання та відтворення знань як невичерпного ресурсу розвитку при мінімізації непоповнюваного ресурсу – часу для відтворення й комунікації знань.

Це означення інформатизації, з одного боку, підтверджує еволюційність процесу суспільного розвитку, оскільки відтворення, зберігання та комунікація знань за мінімальними ресурсами часу забезпечуються саме стадією комп'ютеризації, а з іншого – дає змогу визначити рівень науково-технічного прогресу суспільства за ступенем його інформатизації.

Сучасний етап розвитку інформатики характеризується еволюцією моделей даних у напрямі переходу від традиційних (реляційна, ієрархічна, мережна) до моделей знань [32].

Визначеність ШНС до формальної системи представлення знань. Становлення нової інформаційної технології зумовлено тим, що в теорії ШІ розроблено логіко-лінгвістичні моделі, які дають змогу формалізувати конкретні змістовні знання про об'єкти керування (ОК) та процеси, що в них відбуваються, тобто ввести в ЕОМ логіко-лінгвістичні моделі поряд з математичними. Такі моделі – СС, продукційні системи, фрейми – іноді об'єднуються поняттям «програмно-апаратні засоби в системах із ШІ». Саме цим моделям зобов'язані своєю появою БЗ.

Таким чином, нова інформаційна технологія відрізняється від існуючої насамперед тим, що за допомогою спеціальних формалізмів (логіко-лінгвістичних моделей) декларативні та процедурні знання подаються в електронній формі, завдяки чому розв'язування задач за допомогою ЕОМ відбувається більш ефективно.

Етап перекладу умов задачі (ситуацій) з природної мови на формалізовану присутній в кожній науковій дисципліні. Фахівцям, які займаються дослідженням операцій, постійно доводиться описувати формальною мовою складні ситуації. Цей процес називається *моделюванням*.

Система знань, подана формальною мовою, є *формальною системою*. Вона об'єднує суто абстрактні об'єкти (не пов'язані із зовнішнім світом), визначаючи правила оперування множиною символів у виключно синтаксичному трактуванні без урахування змістового осмислення (або семантики).

Означення 1.6. *Формальна система знань (ФСЗ)* буде визначеною, якщо:

- 1) задано скінченний *алфавіт* (скінченну множину символів);
- 2) визначено процедуру побудови *формул* (або слів);
- 3) виділено деяку множину формул, які називаються *аксіомами*;
- 4) задано скінченну множину *правил виведення*, що дають змогу одержувати з деякої скінченної множини формул іншу множину формул, тобто $U_1 \text{ і } U_2 \text{ і } \dots \text{ і } U_p \rightarrow W_1 \text{ і } W_2 \text{ і } \dots \text{ і } W_n$, де U_i, W_i – формули формальної системи, а стрілка \rightarrow читається як «спричиняє» або «впливає».

Алфавіт, який заздалегідь вважається кінцевим, іноді називається *словником*.

Спосіб подання формул (п. 2 означення 1.6 формальної системи знань) визначає конкретну синтаксичну конструкцію формул (*граматику формул*) – правильно побудовані послідовності символів. На відміну від цього в п. 4 означення 1.6 задаються дозволені правила виведення формул одна з одної.

Оскільки модель ШНС служить моделлю подання знань у системах ШІ, визначимо відповідність апарата ШНС умовам формальної системи, а саме:

- **базовим алфавітом** у ШНС є *вершини* (входів, виходів, тіл нейронів, вагових настроювачів та активаційних функцій, а також базові моделі ШН та види активаційних функцій), спрямовані дуги (у прямоспрямованих та зворотноспрямованих структурах ШНС);
- правила з'єднання ребрами (спрямованими дугами) вершин є **механізмом створення складних утворень** з елементів базово-

вого алфавіту ШНС, тобто формальним апаратом подання ШНС;

- навчання ШНС (з вчителем, без вчителя або методами самоорганізації) шляхом визначення настроювальних ваг становить основу **аксіоматичного твердження** – третьої умови визначеності ШНС як формальної системи;
- нарешті, можливість ініціювання зміни вхідних векторів (образів) в ШНС в результаті їх ідентифікації, класифікації, класстеризації шляхом ототожнювання з аксіоматичними твердженнями – результатами навчання – формує **механізм логічного виведення**.

Оскільки *знання* – це формалізована інформація, яку використовують в процесі логічного виведення, то можна сказати, що нейронна сітка бере *факти* (фактичні знання про світ, подані у виді навчаючої вибірки) і в процесі навчання формує *правила* – знання, які описують знайдений нейросіткою спосіб розв’язання. Ці правила прийняття рішень можна далі витягти з нейронної сітки і записати в одному з традиційних для класичних експертних систем формалізмів подання знань – наприклад, набором продукційних правил логічного виведення.

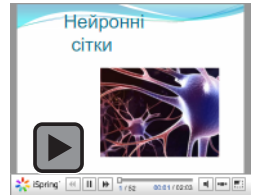
Контрольні завдання

1. Розкрити основні напрями предмета нейробіологія і схарактеризувати класичний нейробіологічний підхід.
2. Охарактеризувати структуру нейрона та функціональне призначення його складових.
3. Пояснити значення спайка в нервовій системі організму.
4. Навести характеристику трьох (зазвичай) основних типів нейросіток.
5. Дати характеристику унікальності побудови біологічних нейронів.
6. Навести й охарактеризувати моделі і функціональні особливості штучних нейронів.
7. Пояснити роль базових нейронів у виборі топології нейросіток.
8. Розкрити роль активаційної функції у формуванні сигналу на виході нейрона.
9. Розкрити структуру адаптивного нелінійного елемента і алгоритм роботи блоку функції перетворення.
10. Навести основні типи активаційних функцій (належності) і розкрити роль цих функцій у процедурах формування даних на виході нейрона.

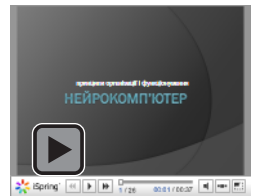
11. Охарактеризувати базову модель нейрона К. Фукушими.
12. Охарактеризувати базову модель нейрона Дж. Хопфілда.
13. Охарактеризувати базову модель нейрона С. Гроссберга.
14. Охарактеризувати базову модель сигма-пі-нейрона.
15. Дати загальну характеристику структури, архітектури і топології штучної нейросітки.
16. Навести і прокоментувати форми подання штучних нейронних сіток: аналітичну, табличну, графічну, векторну, матричну.
17. Навести й охарактеризувати основні за структурою класи нейронних сіток.
18. Розкрити зміст функціональної пов'язаності нейросіток вищого порядку.
19. Дати уявлення про взаємозв'язок інформації та знань і розкрити зміст основних категорій знань: фактів, евристик, мета-знань і правил.
20. Охарактеризувати роль декларативних і процедурних знань у складі моделей нейронних сіток.
21. Дати означення і розкрити зміст поняття формальної системи знань.
22. Розкрити чотири складові – умови відповідності моделі нейронної сітки формальній системі.

Презентації

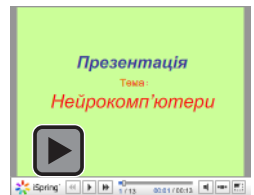
1. Нейронні сітки



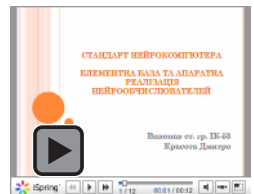
2. Принципи організації і функціонування Нейрокомп'ютер



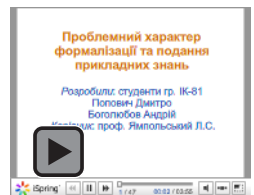
3. Нейрокомп'ютери



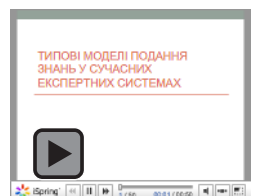
4. Стандарт НЕЙРОКОМП'ЮТЕРА



5. Проблемний характер формалізації та подання прикладних знань



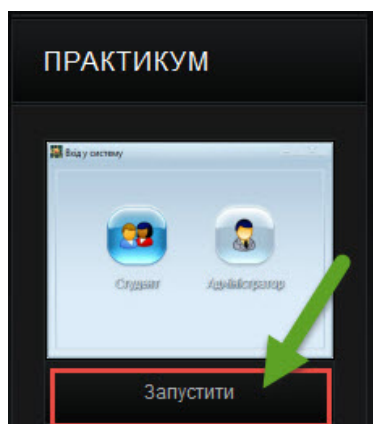
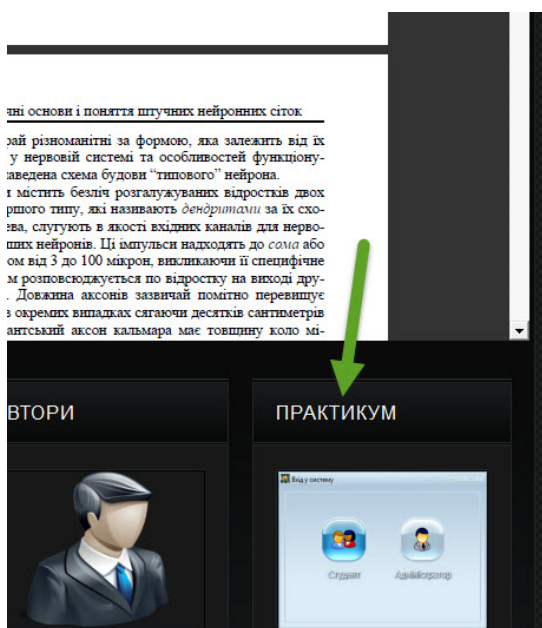
6. Типові моделі подання знань у сучасних Експертних системах



Практичні заняття

Біологічні основи і поняття штучних нейронних сіток

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 2

Класифікація штучних нейронних сіток

Свій бардак чистіший за чужий порядок
Афоризм

2.1. Формування набору вирішних класифікаційних ознак

Проведений у передмові та вступі підручника аналітичний перегляд основних напрямків розвитку НТ, а також результати численних досліджень в інших джерелах, присвячених даній проблемі, дозволяють сформуванню необхідний і достатній набір вирішальних класифікаційних ознак (НВКО) і запропонувати класифікатор ШНС [71], які в сукупності сприятимуть послідовності обґрунтування вибору відповідей “розв’язувана прикладна задача – топологічна модель ШНС” (детальніше див. розд. 26).

Означення 2.1. *Набір вирішних класифікаційних ознак ШНС* – така їх мінімально допустима сукупність, яка є необхідною для формалізації процесу подання основних властивостей ШНС з метою подальшого вибору задовольняючих топологій нейросіток і достатньою для адекватного обслуговування вимог (критеріїв оцінки) з боку прикладної розв’язуваної задачі.

Досвід роботи по створенню агентно-орієнтованого інтерпретатора ШНС дозволяє стверджувати, що наведений нижче перелік НВКО є і необхідним, і достатнім для розв’язання поставленої задачі і містить:

- *тип вхідної інформації* – аналогові і двійкові;
- *модель інтелектуального нейрона* – Мак-Каллоха–Питтса, одношаровий персептрон, Адаліна, Фукушіма, сігма-пі-нейрон, Хопфілда, Гроссберга;
- *тип активаційної функції* – лінійна, лінійні з уніполярним та біполярним насиченням, порогова, біполярна порогова, логістична, гіперболічного тангенсу, гаусова, інверсна мультикватратична, синусоїдальна з насиченням, косинусоїдальна з насиченням, трикутна, трапеціодна;
- *структуру* – прямоспрямовані, зворотного поширення (з прямими, непрямыми, латеральними зв’язками, повнозв’язні);

- *архітектуру* – одношарові (одношаровий персептрон, Адаліна, Н-Адаліна), багатошарові (багатошаровий персептрон, Мадаліна, сітка на основі МГУА тощо);
- *методи навчання* – з вчителем, без вчителя, шляхом самоорганізації, конкурентні, стохастичні, градієнтні;
- *настроювання ваг* – фіксоване і динамічне;
- *метод редукації* – при нарощуванні НС (за алгоритмами Мазарда-Надала, Мерчанда, Лі-Тафтса, каскадної кореляції), при спрощенні НС(видалення ваг, видалення нейронів);
- *критерії оцінки* – простота реалізації, швидкість та ефективність алгоритму навчання, точність апроксимації, потужність при розв'язанні прикладної задачі, кваліфікація користувача, наявність локальних мінімумів, можливість фільтрації та відновлення.

Означення 2.2. *Потужність (розмірність) НС* – її спроможність обробляти певну кількість образів. Як правило, визначається розмірністю вхідного шару НС.

2.2. Класифікація штучних нейронних сіток

Класифікація ШНС з бажаним НВКО наведена на рис. 2.1.

Означення 2.3. *Аналогова вхідна інформація* – інформація на вході ШНС, подана у формі дійсних чисел;

Означення 2.4. *Двійкова вхідна інформація* – інформація на вході ШНС у вигляді нулів та одиниць.

Отже, можна виділити два основних підходи до реалізації НС: *цифровий та аналоговий*. Перевагою аналогових реалізацій є: *висока швидкодія, надійність та економічність*. Проте сфера можливого масового використання навчуваних аналогових нейронів досить вузька. Це обумовлене великою складністю апаратної реалізації високоефективних навчальних алгоритмів і необхідністю спеціальної підготовки потенційних користувачів для оптимальної організації адаптивного процесу. У той же час, широке розповсюдження можуть отримати аналогові нейрокомп'ютери/нейросітки з фіксованою або незначно підналагоджуваною структурою зв'язків – *нейропроцесори*;

Твердження 2.1. Задача створення нейропроцесорів зводиться до навчання цифрової нейросіткової моделі необхідній поведінці на звичайному цифровому комп'ютері.

Запропонований НВКО потребує уточнення деяких тверджень, означень:

Означення 2.5. Базова модель ШН – математична та її програмна і/або апаратна реалізації, на основі яких будується ШНС;

Твердження 2.2. Архітектура ШНС визначається кількістю шарів і нейронів у шарі.

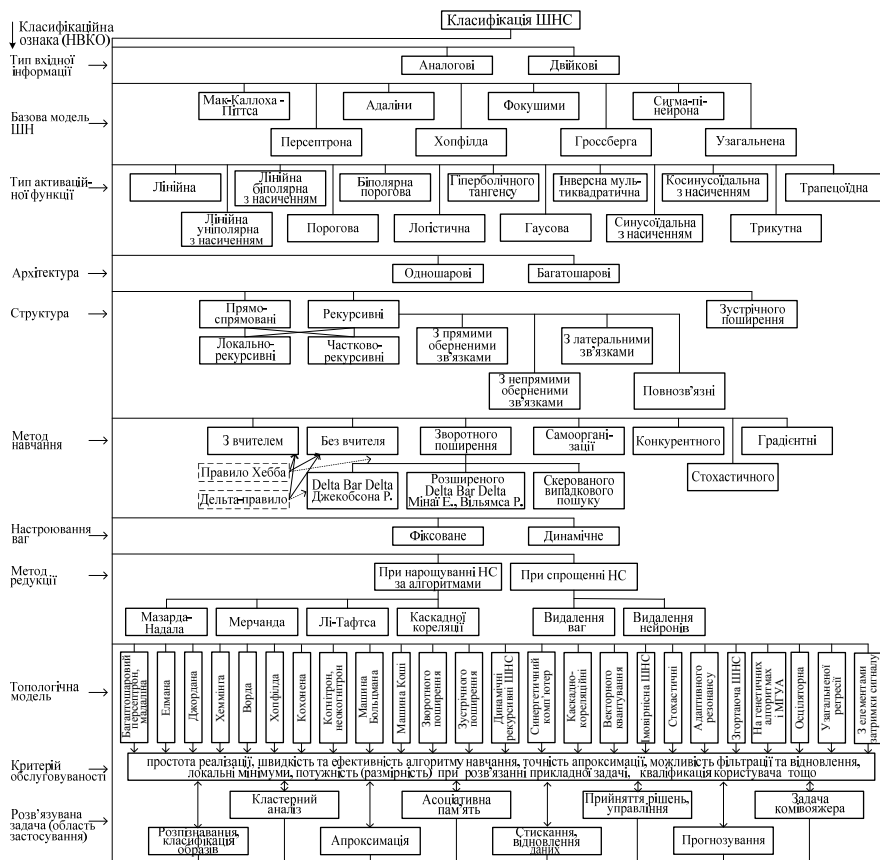


Рис. 2.1. Класифікація ШНС з бажаним НВКО

З позицій архітектурного побудування ШНС важливу роль відіграє *нелінійність активаційної функції* через те, що, якби вона не володіла даною властивістю або не входила в алгоритм роботи кожного нейрона, результат функціонування будь-якої n -шарової НС зводився б до добутку вхідного вектора X сигналів x_i на матрицю вагових коефіцієнтів. Тобто фактично така НС була б еквівалентна одношаровій НС з ваговою матрицею W єдиного шару.

Означення 2.6. Активаційна/передавальна функція (АФ) – функція $f_{ак}(*),$ яка визначає правило переходу нейрона, що знаходиться в момент часу τ в стані $g(\tau),$ у новий стан $g(\tau+1)$ при надходженні вхідних сигналів $x_i.$ Інакше кажучи, ця функція використовується для перетворення рівня активації нейрона у сигнал на виході і разом з пост-синаптичною функцією (яка застосовується спочатку) визначає тип нейрона у сітці.

В ряді топологій нейросіток АФ може залежати не тільки від вагових коефіцієнтів зв'язків $w_{ij},$ але й від часу τ_{ij} передавання сигналу по каналах зв'язку, і у загальному випадку АФ зв'язку C_{ij} від елемента u_i до елемента u_j має вигляд: $C_{ij}^* = f[w_{ij}(t), u_i^*(t - \tau_{ij})].$ Тоді для синхронної нейросітки час τ_{ij} передавання кожного зв'язку або дорівнює нулю, або фіксованій сталій $\tau.$ В асинхронній нейросітці час τ_{ij} передавання для кожного зв'язку між елементами u_j та u_j свій, але також сталий.

Порогові АФ застосовуються в основному в класичних ШНС. При побудованні нових структур ШНС дуже часто необхідно працювати як з самою АФ, так і з її першою похідною, тому виникає потреба у використанні як активаційної диференційовуваної монотонної та обмеженої функції. Особливу роль відіграють такі функції (дістали назву *сигмоїдних*) при моделюванні нелінійних залежностей між змінними на вході та виході.

Твердження 2.3. Функція $f_{ак}(*)$ називається сигмоїдальною, якщо вона є монотонно зростаючою, диференційовуваною і задовольняє умові

$$\lim_{\lambda \rightarrow -\infty} f(\lambda) = k_1, \lim_{\lambda \rightarrow \infty} f(\lambda) = k_2, k_1 < k_2.$$

До характерних сигмоїдних належать логістичні, гіперболічного тангенсу, синусоїдальна і косинусоїдальна з насиченням функції. Перевагою функції гіперболічного тангенсу перед логістичною є її симетричність відносно початку координат, що надає переваги в ряді обчислень. Слід відзначити, що *сигмоїдні функції є диференційованими по всій осі абсцис, що використовується в деяких алгоритмах навчання.* Взагалі, враховуючи, що центральна область сигмоїдальних функцій має більший коефіцієнт підсилення, легше розв'язується проблема опрацювання слабких сигналів, тоді як області функцій зі спадаючим підсиленням на додатному та від'ємному її кінцях слугують для опрацювання сильних збуджень. Отже, нейрон функціонує з більшим підсиленням в широкому діапазоні рівней вхі-

дного сигналу. Вибір конкретного типу АФ специфічний для кожного виду ШНС і залежить від властивостей розв'язуваної задачі.

Твердження 2.4. Структура ШНС визначається напрямом та охопленням зв'язків між шарами та окремими нейронами, тобто характером зв'язків.

Вибір архітектури і структури ШНС здійснюється у відповідності з особливостями і складністю модельованої задачі. Для розв'язання окремих типів задач вже існують оптимальні конфігурації ШНС, але якщо задача не може бути зведеною до будь-якого з відомих типів, доводиться синтезувати нові конфігурації. При цьому слід керуватися наступними основними правилами:

- можливості сітки зростають із збільшенням кількості нейронів у сітці, щільності зв'язків між ними і кількістю шарів;
- уведення зворотних зв'язків поряд із збільшенням можливостей сітки загострює питання щодо динамічної стійкості;
- складність алгоритмів функціонування сітки, уведення декількох типів синапсів сприяє підсиленню потужності ШНС.

Питання щодо необхідних і достатніх властивостей сітки для розв'язання прикладних задач того чи іншого роду уявляє собою цілий напрям нейрокомп'ютерної науки. Оскільки проблема синтезу ШНС в значній мірі залежить від розв'язуваної задачі, дати загальні детальні рекомендації утруднено. В більшості випадків оптимальний варіант отримують на основі інтуїтивного підбирання (обґрунтування вибору ШНС див. у розд. 26), хоча є докази того, що для будь-якого алгоритму існує нейросітка, яка може його реалізувати.

Визначення кількості проміжних шарів і кількості нейронів в них є важливим при моделюванні сітки. Зазвичай, застосовуючи архітектуру до визначених проблем, використовують загальні правила, зокрема:

- кількості входів та виходів сітки визначаються кількостями вхідних та вихідних параметрів досліджуваного об'єкту, явища, процесу, тощо. На відміну від зовнішніх шарів, кількість нейронів прихованого шару $n_{\text{прих}}$ вибирається емпіричним шляхом. В більшості випадків достатня кількість нейронів становить $n_{\text{прих}} \leq n_{\text{вх}} \leq n_{\text{вих}}$, де $n_{\text{вх}}$, $n_{\text{вих}}$ – кількість нейронів у вхідному і, відповідно, у вихідному шарах.
- якщо складність у відношенні між отриманими та бажаними даними на виході збільшується, кількість нейронів прихованого шарі повинна також збільшитись.

- якщо процес, що моделюється, може розділятися на багато етапів, потрібен додатковий прихований шар (шари). Якщо процес не розділяється на етапи, тоді додаткові шари можуть допустити пере-запам'ятовування і, відповідно, невірне загальне рішення.

Оскільки всі ШНС базуються на концепції функціонування нейронів, правил та механізмів їх з'єднань, а також передатних функцій щодо процедури активації, існує подібність між різними структурами або архітектурами нейронних сіток. Більшість змін походить з різних правил навчання. Таким чином, після того, як визначено кількість шарів і число нейронів в кожному з них, потрібно знайти значення для синаптичних ваг і порогів сітки, які спроможні мінімізувати похибку спродукованого результату. Саме для цього існують *алгоритми навчання*, де відбувається підгонка моделі сітки до наявних навчальних даних.

Твердження 2.5. Штучні нейронні сітки не програмуються у звичайному сенсі – вони *навчаються*;

Означення 2.7. *Навчання з вчителем* – таке навчання ШНС, яке припускає існування цільового вектора – бажаного виходу ШНС для кожного вхідного вектора і виконується із залученням правил навчання: Хебба, дельта-правила/його модифікацій, алгоритму зворотного поширення похибки тощо;

Твердження 2.6. Разом вхідний та цільовий вектори утворюють навчаючу пару, і зазвичай сітка навчається на деякому сполученні таких пар;

Означення 2.8. *Навчання без вчителя* – таке навчання, яке відбувається за відсутності цільового вектора, а *навчаючий алгоритм* при цьому підналагоджує ваги таким чином, щоб отримати узгоджені вектори на виході ШНС, причому подання досить близьких вхідних векторів дає однакові вектори на виході;

Означення 2.9. *Навчання з використанням алгоритму зворотного поширення похибки* – це навчання, яке, використовуючи поширення принципів дельта-правила, реалізує градієнтний метод опуклого функціоналу похибки в багатошарових ШНС із структурою прямого поширення на моделях ШН з диференційованими функціями активації;

Твердження 2.7. Похибка для конкретної топології ШНС з варіюваними значеннями синаптичних ваг і порогів сітки (так званих *вільних параметрів топології ШНС*) визначається шляхом

проходження через неї всієї навчальної множини і порівняння спродукованих на виході значень з цільовими/бажаними. Множина похибок утворює *функцію похибок*, яку можна розглядати як похибку сітки. В якості функції похибок найчастіше використовують *суму квадратів похибок*.

Твердження 2.8. Мета навчання ШНС за алгоритмом АЗПП полягає в знаходженні на багатовимірній *поверхні станів* найнижчої точки, яка відповідатиме глобальному мінімуму.

Для кращого розуміння навчання за алгоритмом АЗПП потрібно роз'яснити поняття *поверхні станів*. Кожному значенню синаптичних ваг і порогів сітки (вільних параметрів моделі кількістю N) відповідає один вимір в багатовимірному просторі. Вимір $N+1$ -ий відповідає похибці сітки. Для різноманітних сполучень ваг відповідну похибку сітки можна зобразити точкою в $N+1$ -вимірному просторі, всі ці точки утворюють деяку поверхню станів.

Поверхня станів має складну будову і досить неприємні властивості, зокрема, наявність локальних мінімумів (точки, найнижчі в своєму певному околі, але вищі від глобального мінімуму), пласкі ділянки, сідлові точки і довгі вузькі яри. Аналітичними засобами неможливо визначити розташування глобального мінімуму на поверхні станів, тому навчання нейросітки по суті полягає в дослідженні цієї поверхні. Відштовхуючись від початкової конфігурації ваг і порогів (від випадково обраної точки на поверхні), алгоритм навчання поступово відшукує глобальний мінімум. Обчислюється вектор градієнту поверхні похибок, який вказує напрямок найкоротшого спуску по поверхні з заданої точки. Якщо трошки просунутись по ньому, похибка зменшиться. Зрештою алгоритм зупиняється в нижній точці, що може виявитись лише локальним мінімумом (в ідеальному випадку - глобальним мінімумом). Складність застосування АЗПП полягає у виборі довжини кроків переходу вектора градієнта поверхні похибок, через що або зростає кількість ітерацій і разом з тим час пошуку (при замалих кроках, або *epochs*), або зростає ймовірність “перестрибування” оптимального розв’язку чи руху в невірному напрямку (при занадто великих кроках). На відміну в сенсі ітераційного навчання ШНС епоха – це один прохід по усій навчаючій множині з наступною перевіркою на контрольній множині. Ітеративний процес навчання припиняється або коли *пройдена визначена кількість епох*, або коли *похибка досягає визначеного рівня малості*, або коли *похиб-*

ка перестає зменшуватись (користувач переважно сам вибирає потрібний критерій зупинення процесу);

Означення 2.10. *Навчання шляхом самоорганізації* – таке навчання, за яким аналогічні вхідні сигнали збуджують сусідні нейрони, і такий “колективізм” може здійснюватися завдяки оберненим зв’язкам. Ваги змінюються на кожному кроці навчання, і їх змінювання залежить як від сусідніх вхідних образів, так і ймовірного розподілення, за яким пропонуються для навчання допущені вхідні образи.

Дуже важливим є питання вибору швидкості навчання ШНС.

Означення 2.11. *Швидкість навчання ШНС* – керуючий параметр деяких алгоритмів навчання, який контролює величину кроку при ітераційній корекції ваг.

Евристичні міркування щодо змінювання кроків навчання нейронів дають підстави припустити, що якщо послідовні змінювання ваги мають протилежні знаки, то це означає, що дана вага осцилює, і, отже, швидкість навчання повинна бути зменшена. І навпаки, якщо серія послідовних змінювань ваги має однакові знаки, то швидкість навчання може бути збільшена. При цьому лінійне збільшення змінювання швидкості навчання дозволяє запобігти надмірного її зростання, тоді як геометричне зменшення дозволяє простежити, що швидкість навчання завжди є позитивною. Більше того, швидкість може зменшуватись інтенсивніше на надмірно нелінійних ділянках. Як правило, в якості коефіцієнта швидкості навчання вибирають число, менше за 1 (наприклад, 0,1), яке поступово зменшується в процесі навчання.

Означення 2.12. *Сітки з фіксованими зв’язками* – такі, у яких коефіцієнти ШНС призначаються відразу, виходячи з умов задачі;

Означення 2.13. *Сітки з динамічними зв’язками* – такі, у яких в процесі навчання відбувається налагодження синаптичних ваг;

Означення 2.14. *Топологія ШНС* – це модель НС з певними топологічними властивостями (ТВ) – архітектурою, структурою, активаційною функцією, методами навчання і настроювання ваг тощо і побудована на основі базових моделей ШН з упорядкуванням методами редукції (див. розділи 17 та 23);

Твердження 2.9. *Топологічні властивості (ТВ) ШНС* не змінюються при будь-яких структурно-архітектурних перетвореннях в межах *топологічного простору ШНС* без втрати суттєвості певного виду нейросіток. Прикладом ТВ ШНС є розмірність/потужність

останніх, що визначається кількісною характеристикою: шарів, нейронів у шарі, зворотних зв'язків, входів та виходів сітки (а отже, кількістю опрацьовуваних “образів”).

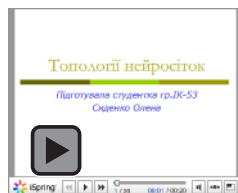
Означення 2.15. *Топологічний простір* (ТП) ШНС – множина моделей сіток будь-якої топології, в якій через НВКО в той чи інший спосіб визначені граничні значення ТВ ШНС.

Контрольні завдання

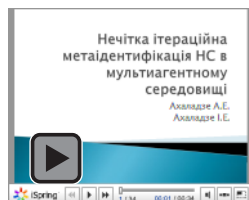
1. Дати тлумачення складових вирішального набору класифікаційних ознак штучних нейронних сіток.
2. Навести умову відповідності активаційної функції до сигмоїдального типу.
3. Прокоментувати правила, на яких ґрунтується вибір архітектури і структури штучних нейронних сіток.
4. Охарактеризувати основні алгоритми навчання нейронних сіток.
5. Проаналізувати функцію похибок як складову похибок штучної нейронної сітки для всієї навчальної множини.
6. Розкрити мету навчання штучних нейронних сіток за алгоритмом зворотного поширення похибки і розкрити сутність багатовимірної поверхні станів сітки.
7. Розкрити фізичний зміст параметра “швидкість навчання” штучної нейронної сітки.
8. Пояснити відмінність між нейросітками з фіксованими і динамічними зв'язками.

Презентації

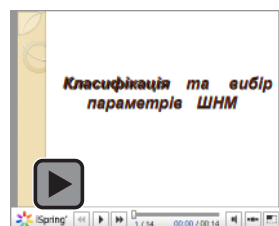
1. Топології нейросіток



2. Нечітка ітераційна метаідентифікація НС в мультиагентному середовищі



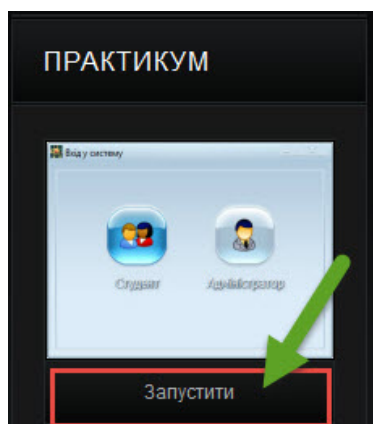
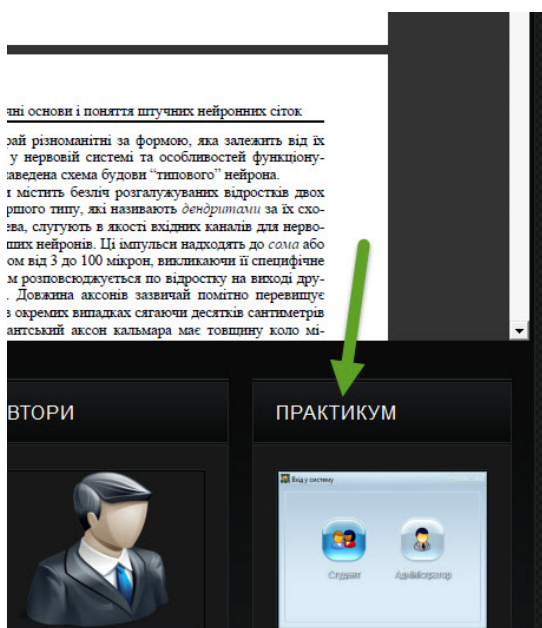
3. Класифікація та вибір параметрів ШНМ



Практичні заняття

Класифікація штучних нейронних сіток

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 3

Підготовка даних при роботі з штучними нейронними сітками

Факти – це повітря вченого

І. П. Павлов

*Я займався до цієї пори розв'язанням низки задач через те,
що при вивченні наук приклади корисніші за правила*

І. Ньютон

*Виміряти все, що піддається вимірюванню,
а що не піддається – зробити вимірюванням*

Галілео Галілей

3.1. Етапи розв'язання задач при підготовці ШНС

Обґрунтуванню вибору ШНС для розв'язання прикладних задач передують етапи підготовки сітки до виконання поставленої мети. Основними серед таких етапів є:

- збирання даних для навчання;
- підготовка і попереднє опрацювання даних;
- вибір топології НС;
- експериментальне підбирання характеристик НС;
- експериментальне підбирання параметрів навчання;
- власне навчання;
- перевірка адекватності навчання;
- коректування параметрів, остаточне навчання;
- вербалізація НС з метою подальшого використання.

Розглянемо детальніше окремі основні етапи.

3.2. Збирання даних для навчання

Вибір даних для навчання ШНС та їх опрацювання є самим складним етапом розв'язання задачі. Набір даних для навчання повинен задовольняти наступним критеріям:

- **репрезентативності** – дані повинні ілюструвати істинне положення речей у предметній галузі;
- **несуперечності** – суперечні дані у навчаючій вибірці приведуть до поганої якості навчання ШНС.

Вихідні (на вході) дані перетворюються до вигляду, в якому їх можна подати на входи сітки. Кожний запис у файлі даних назива-

ється *навчаючою парою* або *навчаючим вектором*. Навчаючий вектор містить по одному значенню на кожний вхід сітки i , в залежності від типу навчання (з учителем або без), по одному значенню для кожного виходу сітки. Навчання нейросітки на «сирому» наборі, як правило, не дає якісних результатів.

3.3. Методи попереднього опрацювання даних

Якщо виникає необхідність використовувати нейросіткові методи для розв'язання конкретних задач, то перше, з чим стикається досліджувач – це підготовка даних. Як правило, при поданні різних нейроархітектур за умовчанням припускають, що дані для навчання вже є і наведені у вигляді, доступному для НС. На практиці ж саме етап передопрацювання може стати найбільш трудомістким елементом нейросіткового аналізу. Успіх у навчанні НС також може вирішальним чином залежати від того, в якому вигляді подана інформація для навчання сітки.

Через це важливого значення набувають різні процедури нормування і методи зниження розмірності вихідних даних, дозволяючи збільшити інформативність навчаючої вибірки.

Попереднє опрацювання даних. Для побудування нейро-нечіткої ШНС (НН ШНС) необхідними є дані двох видів: опис лінгвістичних змінних на вході та виході однією з мов нечіткого моделювання (зокрема, FS-мові); навчаюча вибірка (у форматі LRN) [37].

Лінгвістична змінна визначається типом: неперервна; дискретна. Для неперервних лінгвістичних змінних терм-множина складається з нечітких множин термів з неперервними функціями належності у вигляді трикутників, трапецій, прямокутників, Pi -функцій або гауссіанів, визначених на неперервному числовому базисі (універсальна множина). Кожна з наведених форм апроксимації має свої недоліки і переваги: форми трапеції, трикутника і прямокутника найбільш економічні з точки зору машинної реалізації, тоді як форма Pi -функції є найбільш універсальною і зручною для налагоджування (оптимізації) нечітких моделей градієнтними методами.

Максимізація ентропії як мета передопрацювання. Розглянемо основний керуючий принцип, загальний для усіх етапів передопрацювання даних. Припустимо, що вихідні дані продані у числовій формі, і після відповідного нормування усі вхідні змінні і змінні на виході відображаються в одиничному кубі. Задача нейросіткового моделювання – відшукати статистично достовірні залежності

між вхідними і вихідними змінними. Єдиним джерелом інформації для статистичного моделювання є приклади з навчаючої вибірки. Чим більше біт інформації принесе приклад, тим краще використовуються наявні в нашому розпорядженні дані.

Розглянемо довільну компоненту нормованих (передопрацьованих) даних x_i^{α} . Середня кількість інформації $H(x_i^{\alpha})$, яка надходить з кожним прикладом, дорівнює ентропії розподілення $H(x_i^{\alpha})$ значень цієї компоненти. Якщо ці значення зосереджені у відносно незначній області одиничного інтервалу, інформаційний зміст такої компоненти замалий. В межах нульової ентропії, коли всі значення змінної співпадають, ця змінна не несе ніякої інформації. Навпаки, якщо значення змінної x_i^{α} рівномірно розподілені в одиничному інтервалі, інформація такої змінної максимальна.

Отже, загальний принцип передопрацювання даних для навчання полягає у максимізації ентропії входів і виходів НС.

Нормування даних. Ця процедура виконується у випадках, коли вибірка містить значення, які потребують упорядкування. Нормування робить довжини всіх векторів у сукупності даних однаковими, тобто сума квадратів складових вектора даних така сама, як і для всіх векторів сукупності даних. Вибір ненормованого зразка i відбувається не випадковим чином, а для забезпечення найкращої збіжності алгоритму – на підставі статистичних даних вибірки. Така процедура над вектором простих даних виконується з кожною змінною кожного зразка і є *етапом попереднього опрацювання*.

Як входами, так і виходами можуть бути зовсім різнорідні величини. Очевидно, що результати нейросіткового моделювання не повинні залежати від одиниць вимірювання цих величин. Тому сітка має трактувати їх значення одноманітно, а для цього усі вхідні і вихідні величини мають бути зведені до єдиного масштабу. Крім того, для підвищення швидкості і якості навчання корисно провести додаткове передопрацювання, яке вирівнюватиме розподілення значень ще до етапу навчання.

Індивідуальне нормування даних. Зведення до єдиного масштабу забезпечується нормуванням кожної змінної на діапазон розкиду її значень. В найпростішому варіанті це – лінійне перетворення $x_i^{\alpha} = (x - x_{i \min}) / (x_{i \max} - x_{i \min})$ в одиничний відрізок $x_i^{\alpha} \in [0, 1]$. Узагальнення для відображення даних в інтервал $[-1, 1]$, який рекомендується для вхідних даних, тривіальне.

Твердження 3.1. Лінійне нормування є оптимальним, коли значення змінної x_i щільно заповнюють визначений інтервал.

Проте подібний “прямолінійний” підхід далеко не завжди застосований. Так, якщо у даних є відносно рідкі викиди, набагато перевищуючі типовий розкид, саме ці викиди визначають, відповідно до попередньої формули, масштаб нормування. Це приведе до того, що основна маса значень нормованої змінної \mathcal{X} зосередиться поблизу нуля, тобто $[\mathcal{X}] \ll 1$.

Через це набагато надійніше орієнтуватися при нормуванні не на екстремальні значення, а на типові, тобто статистичні характеристики даних, такі, як середнє і дисперсія:

$$\mathcal{X} = (x_i - \bar{x}_i) / \sigma_i,$$

де

$$\mathcal{X} = P^{-1} \sum_{\alpha=1}^P x_i^{\alpha}, \quad \sigma_i^2 \equiv (P-1)^{-1} \sum_{\alpha=1}^P (x_i^{\alpha} - \bar{x}_i)^2$$

В цьому випадку основна маса даних матиме одиничний масштаб, тобто типові значення усіх змінних будуть порівнянними (рис. 3.1)

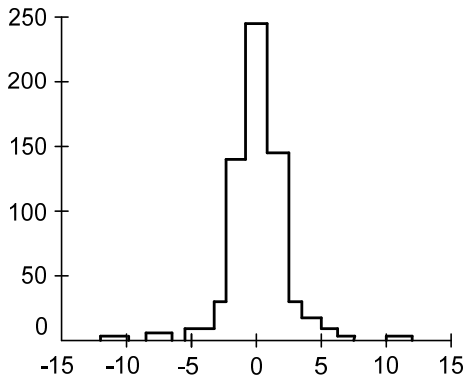


Рис. 3.1. Гістограма значень змінної за наявності рідких, але значущих за амплітудою відхилень від середнього

Проте, тепер нормовані величини не належать гарантовано одиничному інтервалу, більше того, розкид значень \mathcal{X} заздалегідь не відомий. Для вхідних даних це може бути і не важливим, але змінні на виході будуть використовуватися як еталони для нейронів на виході. У випадку, якщо нейрони на виході – сигмоїдальні, вони мо-

жуть приймати значення тільки в одиничному діапазоні. Щоб встановити відповідність між навчаючою вибіркою і нейросіткою в цьому випадку необхідно обмежити діапазон змінювання змінних.

Лінійне перетворення, наведене вище, не здатне віднормувати основну масу даних і одночасно обмежити діапазон можливих значень цих даних. Природний вихід з цієї ситуації – використати для передопрацювання даних функцію активації тих же нейронів. Наприклад, нелінійне перетворення

$$\mathcal{X}_i = f((x_i - \bar{x}_i) / \sigma_i), \quad f(\alpha) = (1 + e^{-\alpha})^{-1}$$

нормує основну масу даних, одночасно гарантуючи, що $\mathcal{X}_i \in [0, 1]$ (рис. 3.2).

Як видно з рис. 3.2, розподілення значень після такого нелінійного перетворення значно ближче до рівномірного.

Усі вищезазначені методи нормування спрямовані на те, щоб максимізувати ентропію кожного входу (виходу) відокремлено. Але, загалом, можна досягти значно більшого, максимізуючи їх *спільну* ентропію. Існують методи, які дозволяють виконувати нормування для усієї сукупності входів. Деякі з цих методів наведено в [21].

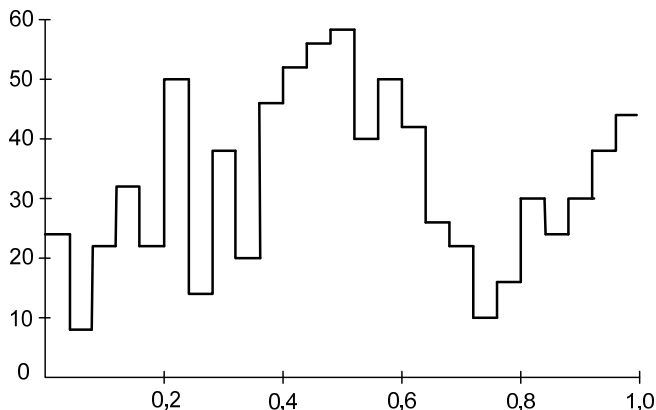


Рис. 3.2. Нелінійне нормування з використанням логістичної функції $f(\alpha) = (1 + e^{-\alpha})^{-1}$

Пониження розмірності входів. Оскільки заздалегідь невідомо, наскільки корисними є ті чи інші вхідні змінні для завбачення значень входів, виникає спокуса збільшувати кількість вхідних параметрів в надії на те, що сітка сама визначить, які з них найбільш значущі. Проте частіше за все це не приводить до очікуваних результа-

тів і збільшує, окрім того, складність навчання. Навпаки, стиснення даних, зменшення ступеня їх надмірності може суттєво полегшити наступну роботу, виокремлюючи дійсно незалежні ознаки. Скорочення розмірності навчаючої вибірки даних є однією з головних проблем при побудуванні діагностичних і розпізнаючих моделей за прецедентами, оскільки дозволяє прискорити та спростити процес побудови моделі, а також спростити її структуру [40].

Традиційно застосовуваними інструментами скорочення розмірності вибірок даних є *методи відбирання інформативних ознак* (feature selection) [18] та *методи витягання* (конструювання) *ознак* (feature extraction) [6]. З іншого боку, розмірність даних можна скоротити шляхом формування вибірки меншого розміру з вихідної вибірки [41].

Отже, можна виділити два ефективних типи алгоритмів, призначених для пониження розмірності даних з мінімальною втратою інформації:

- відбирання найбільш інформативних ознак і використання їх у процесі навчання НС;
- кодування даних на вході меншою кількістю змінних, але при цьому таких, що містять по можливості всю закладену у цих даних інформацію.

Розглянемо більш детально обидва типи алгоритмів.

Добір найбільш інформаційних ознак. Для того, щоб зрозуміти, які з вхідних змінних несуть максимум інформації, а якими можна нехтувати, необхідно або порівняти всі ознаки між собою і визначити ступінь інформативності кожної з них, або спробувати знайти визначені комбінації ознак, які найповніше відбивають основні характеристики даних на вході.

Для вибору задовольняючої комбінації вхідних змінних використовують так звані *генетичні алгоритми* [88], які добре пристосовані для задач такого типу, оскільки дозволяють виконувати пошук серед значної кількості комбінацій за наявності внутрішніх залежностей у змінних.

Стиснення інформації. Аналіз головних компонент. Самий розповсюджений метод зниження розмірності – це *аналіз головних компонент* (АГК). Традиційна реалізація цього методу наводиться в теорії лінійної алгебри. Основна ідея полягає в наступному: до даних застосовується лінійне перетворення, за яким напрямкам нових координатних осей відповідають напрямки найбільшого розкиду даних на вході. Для цього визначаються попарно ортогональні

напрямки максимальної варіації даних на вході, після чого дані проєктуються на простір меншої розмірності, породжений компонентами з найбільшою варіацією [21]. Один з недоліків класичного методу головних компонент полягає в тому, що це чисто лінійний метод, і він, відповідно, не може враховувати деякі важливі характеристики структури даних.

В теорії НС розроблені більш потужні алгоритми, здійснюючі нелінійний аналіз головних компонент [17]. Вони уявляють собою самостійну нейросіткову структуру, яку навчають видавати в якості виходів свої вхідні дані, але при цьому в її прихованому шарі міститься менше нейронів, ніж у вхідному і вихідному шарах (рис. 3.3). Нейросітки подібного роду носять назву *автоасоціативних*.

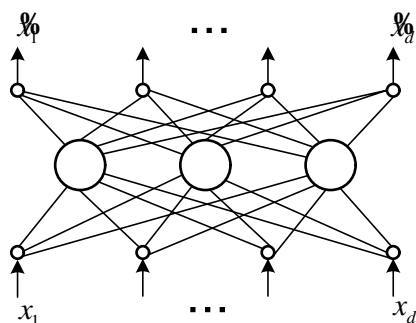


Рис. 3.3. Автоасоціативні НС з “вузьким горлом” – аналог правила навчання Ойя

Щоб відтворити свої вхідні дані, НС повинна навчитися подавати їх у більш низькій розмірності. Базовий алгоритм навчання у цьому випадку носить назву *правило навчання Ойя* для одношарової сітки. Враховуючи на те, що в такій структурі ваги з однаковими індексами в обох випадках однакові, дельта-правило навчання верхнього (а тим самими і нижнього) шару можна записати у вигляді:

$$\Delta w_{ij} = \eta y_i (x_j - \mathcal{Y}_j) = \eta y_i \left(x_j - \sum_{k=1}^Z y_k w_{kj} \right),$$

де: $y_i = \sum_{j=1}^d w_{ij} x_j$ ($i=1, \dots, Z$); $\mathcal{Y}_j = \sum_{k=1}^Z w_{kj} y_k$ ($j=1, \dots, d$); x_j – компонента вхідного вектора ($j=1, 2, \dots, d$); \mathcal{Y}_j – виходи нейрона ($j=1, 2, \dots, d$); d – кількість нейронів у вхідному та вихідному шарах (розмірність вектора ознак); y_i – вихід з i -го нейрона прихованого

шару, $i = 1, \dots, Z$ – кількість нейронів прихованого шару; h – коефіцієнт навчання; $w_{ij} = w_{kj}$ – ваги, відповідно, між вхідним – прихованим і прихованим – вихідним шарами НС.

Прихований шар такої сітки здійснює оптимальне кодування вхідних даних і містить максимально можливу при даних обмеженнях кількість інформації.

Після навчання (визначення ваг w_{ij}) зовнішній інтерфейс (рис. 3.4) може бути збережений і використаний для пониження розмірності.

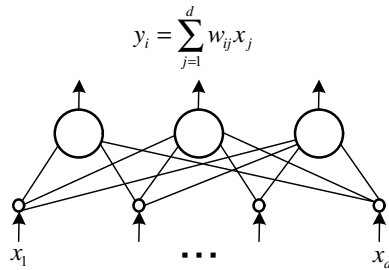


Рис. 3.4. Шар лінійних нейронів

Нелінійний аналіз головних компонент. Головна перевага нейроалгоритмів полягає в тому, що вони легко узагальнюються на випадок нелінійного стиску інформації, коли ніяких явних розв'язків вже не існує. Можна замінити лінійні нейрони в наведених вище НС нелінійними. З мінімальними видозміненнями нейроалгоритми будуть працювати і в цьому випадку, завжди відшукуючи оптимальний стиск інформації за накладеними обмеженнями. Наприклад, проста заміна лінійної функції активації нейронів на сигмоїдальну в правилі навчання Ойя

$$\Delta w_{ij} = \eta f(y_i) \left(x_j - \sum_{k=1}^Z f(y_k) w_{kj} \right)$$

веде до нової якості.

Таким чином, нейроалгоритми уявляють собою зручний інструмент нелінійного аналізу, дозволяючий відносно легко знаходити способи глибокого стиску інформації і виокремлення нетривіальних ознак.

Інтермальне масштабне перетворення. Для *інтермального масштабного перетворення* (ІМП) мінімальне значення кожної змінної розміщують у початок відліку та ділять на інтервал даних,

щоб максимальне значення кожної нової характеристики дорівнювало 1. За відсутністю якої-небудь апіорної інформації дані повинні бути масштабовані так, щоб всі змінні несли однакову інформацію щодо їх дисперсій.

Автомасштабне перетворення. При *автомасштабному перетворенні* (АМП) усувається будь-яке ненавмисне зважування, яке виникає завдяки довільності одиниць; проте АМП не є таким чутливим до викидів, як це має місце у ІМП. А оскільки АМП на одиницю дисперсії має відношення до середньоцентрованому, за яким слідує ділення на стандартне відхилення змінної з базису змінних, то при інших рівних умовах до використання перевага у більшості застосувань надається АМП, а не ІМП.

3.4. Вибір топології штучних нейронних сіток

Вибирати тип нейросітки необхідно виходячи з постави задачі та наявних даних для навчання. Для навчання з учителем необхідна наявність для кожного елемента вибірки «експертної» оцінки. Іноді отримання такої оцінки для великого масиву даних неможливе. У таких випадках природнім вибором є сітка, яка навчається без учителя, наприклад, самоорганізуюча карта Кохонена або нейронна сітка Хопфілда. При розв'язанні інших задач, таких, як прогнозування часових рядів, експертна оцінка вже міститься у даних на вході і може бути виокремлена при їх опрацюванні. У цьому випадку можна використати багатошаровий перцептрон або нейросітку Ворда.

Питанням обґрунтування вибору топології нейросіток при розв'язанні прикладних задач присвячено *розділ 25* підручника.

3.5. Експериментальне підбирання характеристик сітки

Після вибору загальної структури треба експериментально підібрати параметри сітки. Для сіток, подібних перцептроні, це буде кількість шарів, кількість блоків у схованих шарах (для сіток Ворда), наявність або відсутність обхідних з'єднань, передаткові функції нейронів. При виборі кількості шарів і нейронів в них необхідно виходити з того, що спроможності сітки до узагальнень тим вищі, чим більша сумарна кількість зв'язків між нейронами. Проте, кількість зв'язків обмежена зверху кількістю записів у навчаючих даних.

3.6. Експериментальне підбирання параметрів навчання

Після вибору конкретної топології необхідно вибрати параметри навчання нейронної сітки. Цей етап вкрай важливий для сіток, які навчаються з учителем. Від правильного вибору параметрів зале-

жити не тільки те, наскільки швидко відповіді сітки будуть збігатися до правильних відповідей. Наприклад, вибір низької швидкості навчання збільшує час збігання, проте іноді дозволяє запобігати *паралічу сітки*. Зростання часу навчання може привести як до збільшення, так і до зменшення часу збіжності в залежності від форми поверхні похибки. Виходячи з такого суперечного впливу параметрів, можна зробити висновок, що їх значення необхідно вибирати експериментально, керуючись при цьому критерієм завершення навчання (наприклад, мінімізації похибки або обмеженням за часом навчання).

3.7. Власне навчання сітки

Власне методи навчання ШНС розглядаються у наступному розділі 4. Тут обмежимося тільки загальними зауваженнями.

В процесі навчання сітка у визначеному порядку переглядає навчаючу вибірку. Порядок переглядання може бути послідовним, випадковим и т. п. Деякі сітки, які навчаються без учителя, наприклад, сітки Хопфілда переглядають вибірки тільки один раз. Інші, наприклад, сітки Кохонена, а також сітки, які навчаються з учителем, переглядають вибірку багато разів, при цьому одне повне проходження вибірки називається *епохою навчання*. При навчанні з учителем набір початкових даних поділяють на дві частини – власне *навчаючу вибірку* і *тестові дані*; принцип розділення може бути довільним. Навчаючі дані подаються сітці для навчання, а перевіірочні використовуються для розрахунку похибки сітки (перевіірочні дані ніколи для навчання сітки не використовуються). Таким чином, якщо на перевіірочних даних похибка зменшується, то сітка дійсно виконує узагальнення. Якщо ж похибка на навчаючих даних продовжує зменшуватись, а похибка на тестових даних збільшується, це означає, що сітка перестала виконувати узагальнення і просто “запам’ятовує” навчаючі дані. Це явище називається *перенавчанням* сітки або *оверфітінгом*. У таких випадках навчання зазвичай припиняють. В процесі навчання можуть виявитися інші проблеми, такі як *параліч* або попадання сітки в *локальний мінімум поверхні похибок*. Неможливо заздалегідь завбачити виявлення тієї або іншої проблеми, так само як і дати однозначні рекомендації щодо їх розв’язання.

3.8. Перевірка адекватності навчання

Навіть у випадку успішного, на перший погляд, навчання сітка не завжди навчається саме тому, чого від неї жадав створювач. Відомий випадок, коли сітка навчалась розпізнаванню зображень

танків по фотографіях, проте пізніше виявилось, що всі танки були сфотографовані на одному й тому ж фоні. В результаті сітка “навчилась” розпізнавати цей тип ландшафту, замість того, щоб “навчитися” розпізнавати танки. Таким чином, сітка “розуміє” не те, що від неї вимагалось, а те, що простіше за все узагальнити.

3.9. Вербалізація штучних нейронних сіток

Одним з основних недоліків навчуваних ШНС з точки зору багатьох користувачів є той факт, що з навченої нейронної сітки користувачу важко витягти явний і зрозумілий алгоритм розв'язання задачі – сама НС є цим алгоритмом, і якщо структура сітки складна, то цей алгоритм є незрозумілим. Проте, спеціальним чином побудована процедура спрощення і *вербалізації* часто дозволяє витягти явний метод розв'язання.

Означення 3.1. *Вербалізація* – мінімізований опис роботи синхронізованої і вже навченої НС у вигляді декількох взаємозалежних алгебраїчних або логічних функцій.

Зокрема, вербалізація здійснюється для підготовки навченої і спрощеної НС до реалізації в програмному коді або у вигляді спеціалізованого електронного (оптоелектронного) пристрою, а також для використання результатів у вигляді явних знань. Під збудниками при цьому розуміються вхідні значення НС, а під реакцією ШНС – значення на виходах нейронів. Кінцевий синдром – це значення нейросітки на виході. Вербалізація зазвичай здійснюється засобами спеціалізованих пакетів і полягає в наступному:

- використання поодиноких задач спрощення і вербалізації;
- спрощення архітектури НС;
- зменшення кількості вхідних сигналів;
- зведення параметрів НС до невеликої кількості виокремлених значень;
- зниження вимог до точності вхідних сигналів;
- формулювання явних знань у вигляді симптом-синдромної структури та явних формул формування синдромів з симптомів;
- використання алгоритмів проріджування.

Перед вербалізацією, як правило за допомогою продукційних правил, для деяких видів сіток пропонується спрощувати структуру НС шляхом *проріджування* (pruning), основна ідея якого полягає в тому, що ті елементи моделі або ті нейрони сітки, які чинять малий вплив на похибку апроксимації, можна виключити з моделі без знач-

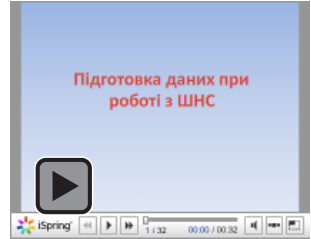
ного погіршення якості апроксимації. Але при цьому треба мати на увазі, що це є справедливим тільки для конкретної розв'язуваної задачі. Якщо ж з'являться нові статистичні дані для навчання, то провіджена сітка втратить спроможність до узагальнення, якою вона володіла б, якщо зв'язки не були б втраченими (як мінімум зворотне не було доказаним). Таким чином, мова йде про алгоритми зі втратою якості, які можуть застосовуватися для окремих задач, але не можуть застосовуватися поза залежності від задачі.

Контрольні завдання

1. Навести етапи підготовки нейронної сітки до виконання поставленої мети.
2. Охарактеризувати критерії репрезентативності і несуперечності даних для навчання нейронної сітки.
3. Розкрити основний керуючий принцип максимізації ентропії для усіх етапів передопрацювання даних для навчання нейронної сітки.
4. Прокоментувати підхід передопрацювання даних з індивідуальним їх нормуванням на діапазон розкиду значень.
5. Пояснити функції і особливості лінійного та нелінійного перетворення даних при їх підготовці до введення в нейронну сітку.
6. Охарактеризувати методи відбирання і витягання інформаційних ознак при підготовці даних до введення в нейронну сітку.
7. Розкрити метод зниження розмірності даних – аналіз головних компонент і лінійне перетворення даних.
8. Пояснити, в чому полягає базовий алгоритм навчання у випадку більш низької розмірності сітки – правило навчання Ойя.
9. Розкрити принцип нелінійного стиску інформації при підготовці до введення в нейронну сітку.
10. Охарактеризувати принцип інтермального масштабного перетворення даних.
11. Розкрити основні особливості автомасштабного перетворення при підготовці нейронної сітки до введення вхідної інформації.
12. Пояснити процедуру експериментального підбирання характеристик нейронної сітки і параметрів її навчання.
13. Навести означення і розкрити сутність вербалізації нейронної сітки.

Презентації

1. Підготовка даних



Розділ 4

Навчання нейронних сіток

Вік живи – вік вчися!

*І ти, врешті решт, досягнеш того, що, подібно мудрецю,
будеш мати право казати, що нічого не знаєш*

Козьма Прутков

...недолік розуму не слід пояснювати недоліками пам'яті

Клод Адріан Гельвецій

Навчаючи – вчуся

Луцій Анней Сенека молодший

Пороки засвоюються і без учителя

Луцій Анней Сенека молодший

4.1 Механізм навчання штучних нейронних сіток

Загальні зауваження. Для переходу до розгляду механізмів, методів і алгоритмів навчання нейронних сіток спочатку треба зазначити, що у своєму життєвому циклі НС проходять три стадії: створення (або вибору з номенклатури сіток за ознакою їх обслуговування прикладної задачі), навчання і застосування. Ця тріада може бути інтерпретована схемою на рис. 4.1.

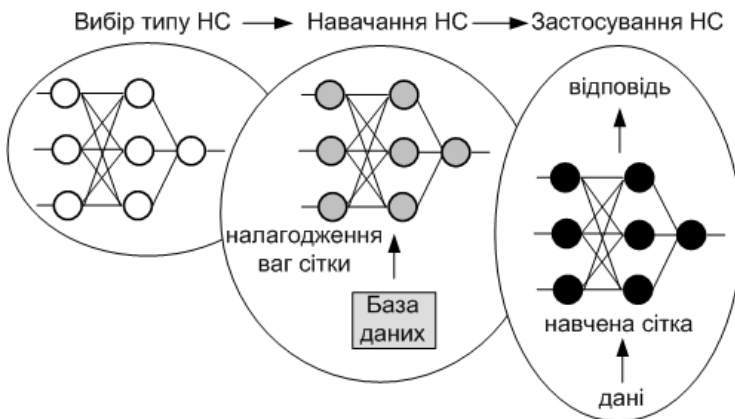


Рис. 4.1. Послідовність стадій життєвого циклу ШНС

Детальніше про етапи вибору топологій і сфер застосування ШНС мова йтиме в подальших розділах підручника. В цьому ж розділі

ділі приділимо увагу одній з основних і відмінних функцій НС – їх навчанню.

Твердження 4.1. Штучні нейронні сітки не програмуються у звичайному сенсі – вони навчаються.

Теорія навчання розглядає три фундаментальних властивості, пов’язаних з навчанням за прикладами: ємність, складність образів та обчислювальна складність.

Під *ємністю* розуміють кількість образів, які може запам’ятати ШНС, та які функції і межі прийняття рішень можуть бути в ній сформовані.

Складність образів визначає кількість навчаючих прикладів, необхідних для досягнення спроможності сітки до навчання. Занадто мала кількість прикладів може викликати “перенавченість” ШНС, коли вона добре функціонує на прикладах навчаючої вибірки, але погано – на тестових прикладах, підпорядкованих тому ж самому статистичному розподілу.

В залежності від розв’язуваної задачі у навчаючій вибірці використовують ті або інші типи даних в залежності від вхідних сигналів та сигналів на виході. *Вхідні дані прикладів навчаючої вибірки* – зображення, таблиці чисел, розподілення. Набір навчаючих прикладів формується на розсуд користувача програми моделювання ШНС індивідуально для кожної конкретної розв’язуваної задачі.

Нейросітка навчається, щоб для деякої множини входів дати бажану (або, принаймні, відповідну до неї) множину виходів. Кожна вхідна (або на виході) множина розглядається як *вектор*. Навчання здійснюється шляхом послідовного пред’явлення вхідних векторів з одночасним підналагоджуванням ваг у відповідності з визначеною процедурою. В процесі навчання ваги НС поступово стають такими, щоб кожний вхідний вектор формував відповідний вектор на виході. Технічно навчання полягає у *знаходженні коефіцієнтів зв’язків між нейронами*.

Якщо в ненавчену ШНС ввести вхідний сигнал одного з прикладів навчаючої вибірки, то сигнал на виході сітки буде суттєво відрізнятися від бажаного. Функція похибки чисельно визначає схожість усіх поточних сигналів на виході сітки та відповідних бажаних сигналів на навчаючої вибірки.

Твердження 4.2. *Мета навчання* – мінімізувати функцію похибки, тобто знайти такі значення параметрів ШНС, за яких поточні сигнали на виході сітки мінімально відрізняються від відповідних бажаних вхідних сигналів, заданих навчаючою вибіркою.

Означення 4.1. *Навчання НС* – це процес, в якому вільні параметри НС адаптуються в результаті її безперервної стимуляції зовнішнім оточенням.

Це означення процесу навчання НС припускає наступну послідовність подій:

1. В нейронну сітку надходять стимули із зовнішнього середовища.
2. В результаті першого пункту змінюються вільні параметри НС.
3. Після змінення внутрішньої структури НС відповідає на збудження вже інакше.

Вищенаведений список правил роз'яснення проблеми навчання НС асоціюється з *алгоритмом* навчання.

Означення 4.2. *Алгоритм* – поняття, під яким, у загальному випадку, розуміють деякий припис виконати точно визначену послідовність дій, спрямованих на досягнення заданої мети або розв'язання поставленої задачі.

В загальному випадку *алгоритм навчання означає процедуру*, в якій використовуються навчання для підналагоджування ваг.

Не існує універсального алгоритму навчання, придатного для усіх топологій НС. Існує лише набір засобів, представлений множиною алгоритмів навчання, кожний з яких має свої переваги. Алгоритми навчання відрізняються один від одного *способом налагодження синаптичних ваг нейронів*. Ще однією відмінною характеристикою є *спосіб зв'язку навчованої НС із зовнішнім середовищем*. В цьому контексті говорять про *парадигму навчання*, пов'язану з моделлю оточуючого середовища, в якому функціонує дана НС.

Для навчання ШНС можуть бути застосовані всілякі алгоритми. Можна виокремити дві великі групи алгоритмів – градієнтні та стохастичні.

Градієнтні алгоритми навчання сіток базуються на обчисленні частинних похідних функції похибки по параметрах сітки. Серед градієнтних розрізняють алгоритми першого і другого порядків.

В *стохастичних алгоритмах* пошук мінімуму функції похибки ведеться випадковим чином.

При навчанні сіток, як правило, використовується один з двох наступних критеріїв зупинення:

- при досягненні деякого задовольняючого малого значення функції похибки;

- при досягненні критичного значення якого-небудь параметра, наприклад, кількості ітерацій;
- у випадку розв'язання всіх прикладів навчаючої вибірки.

Вирішення задачі навчання ШНС. Для вирішення задачі навчання нейросітки необхідно виконати ряд послідовних процедур:

1. Правильно вибрати структуру НС;
2. Коректно вибрати параметри навчання, а саме:
 - крок навчання;
 - кількість параметрів для навчання;
 - кількість прикладів для навчання;
 - метод (парадигму, алгоритм) навчання;
3. Підготувати вхідні (початкові) дані.

Перед навчанням виконується *ініціалізація ШНС*, тобто при-власнювання параметрам сітки деяких початкових значень. Як правило, ці початкові значення – деякі малі випадкові числа.

Початкові дані перетворюють до вигляду, в якому їх можна подати на входи сітки. Кожний запис називається *навчальною парою/навчальною вибіркою* або *навчальним вектором*.

Твердження 4.3. Разом *вхідний та цільовий вектори утворюють навчаючу пару*, і зазвичай сітка навчається на деякому сполученні таких пар.

Навчання сітки на «сирому» наборі не дає якісних результатів. Саме через це перед роботою з НС необхідно використати способи для покращення «сприйняття» даних в сітці:

- *нормування* (виконується, коли на різні входи подаються дані різної розмірності) – розмірності всіх вхідних і вихідних даних зводяться воедино;
- *квантування* (виконується над безперервними величинами) – виділяється кінцевий набір дискретних значень;
- *фільтрація* (виконується для зашумлених даних).

Метод навчання визначається способом підналагодження цих параметрів. Спроможність навчатися – одна з головних переваг НС перед традиційними алгоритмами. Технічно навчання полягає у знаходженні коефіцієнтів зв'язків між нейронами. В процесі навчання НС спроможна виявляти складні залежності між вхідними даними і даними на виході, а також *виконувати узагальнення*. Це означає, що у випадку успішного навчання сітка може повернути вірний результат на підставі даних, які були відсутні у навчаючій вибірці, а також неповних і/або “зашумлених”/спотворених даних.

Існує багато можливостей навчання ШНС [48], серед яких основними є змінення:

- конфігурації сітки шляхом утворення нових або виключення існуючих зв'язків між нейронами;
- елементів матриці зв'язків (ваг);
- характеристик нейронів (виду і параметрів активаційної функції).

Найбільш поширеним є підхід, за яким структура НС задається апріорно, а сітка навчається шляхом підналагоджування *матриці зв'язків* (вагових коефіцієнтів) W . Від того, наскільки вдало побудована матриця, залежить ефективність даної НС. В такому випадку навчання полягає у змінюванні за визначеною процедурою елементів матриці W при послідовному пред'явленні нейросітці навчаючих векторів.

З урахуванням вищезначеного узагальнену модель навчання штучного нейрона можна подати структурою (рис. 4.2).

В процесі навчання ваги стають такими, що при надходженні вхідних сигналів/векторів НС виробляє відповідні вимагані сигнали/вектори на виході.

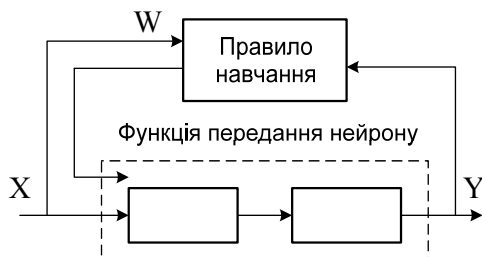


Рис. 4.2. Узагальнена модель навчання штучного нейрона

Узагальнену процедуру навчання НС можна відобразити схематично (рис. 4.3), коли з бази даних вибирається навчаюча вибірка вхідних векторів (образів).

Останні надходять по чергово на входи нейросітки, і в результаті виконання певної процедури щодо підналагоджування ваг нейронів НС, досягають мети навчання за критерієм визначеного рівня помилки у відображенні вихідного вектора по відношенню до бажаного його виду. Якщо для векторів усієї навчаючої вибірки досягнуто певного компромісу щодо відповідності векторів на виході, нейросітка вважається навченою, і процес навчання зупиняється.

Для навчання НС необхідно мати набір даних, який розбивають на три частини: навчаючу, перевірочну та тестуючу вибірки.

Навчаюча вибірка призначена для навчання НС і використовується для підбору ваг при навчанні (у більшості випадків мова йде про ітеративно збігаючі алгоритми).

Перевірочна вибірка призначена для того, щоб вчасно закінчити навчання НС, бо іноді збільшення кількості навчаючих циклів – *epoch* веде до виникнення ефекту перенавчання. При цьому НС видає чудові результати на навчаючій вибірці та все більш погіршуючі результати на перевірочній вибірці. У більшості випадків це означає, що кількість вузлів у сітці зavelика і сітка “запам’ятовує” інформацію замість того, щоб її аналізувати.

Тестуюча вибірка призначена для перевірки якості прогнозу остаточно навченої НС.

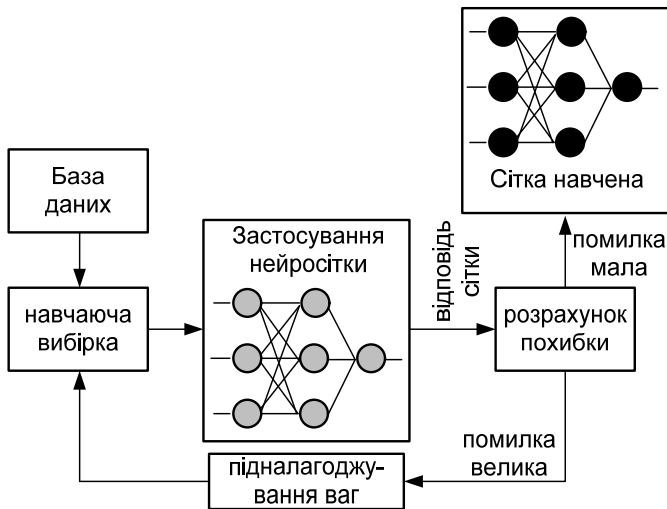


Рис. 4.3. Узагальнена схема процесу навчання НС

Як вище було зазначено, навчання можна розділити на *epoch*.

Означення 4.3. *Епоха* – це період навчання НС, за який вона один раз була навчена за допомогою кожного із спостережень у навчальній вибірці.

Підходи до навчання ШНС. Можливі, як мінімум, три підходи до навчання нейронних сіток [152]:

- *Навчання з вчителем.* На кожному кроці навчання вектор \bar{Y} , що видається сіткою, порівнюється з цільовим вектором \bar{Y}_c . Ваги змінюються за однією з формул, наприклад “Delta - Regal” – дельта-правилом, якщо $\bar{Y} \neq \bar{Y}_c$.

Означення 4.4. *Навчання з вчителем* – таке навчання ШНС, яке припускає існування цільового вектора – бажаного виходу ШНС для кожного вхідного вектора і виконується із залученням правил навчання: Хебба, дельта-правила/його модифікацій, *алгоритму зворотного поширення похибки* (АЗПП) (Backpropagation Algorithm) тощо.

- *Навчання без вчителя.* Тут точний цільовий вектор не задається, а ваги на кожному кроці трохи змінюються у довільному напрямку. За допомогою оцінювальної функції визначається чи поліпшується розрахований вихідний вектор при зміні ваг в порівнянні з попереднім значенням. Якщо поліпшення досягнуто, змінені ваги запам'ятовуються, а в протилежному разі вони “забуваються”.

Означення 4.5. *Навчання без вчителя* – таке навчання, яке відбувається за відсутності цільового вектора, а *навчаючий алгоритм* при цьому підналагоджує ваги таким чином, щоб отримати узгоджені вектори на виході ШНС, причому подання досить близьких вхідних векторів дає однакові вектори на виході.

- *Навчання шляхом самоорганізації.* Ваги змінюються на кожному кроці навчання і залежать від: а) сусідніх вхідних образів; б) ймовірного розподілу, за яким пропонуються до навчання припустимі вхідні образи.

Означення 4.6. *Навчання шляхом самоорганізації* – таке навчання, за яким аналогічні вхідні сигнали збуджують сусідні нейрони, і такий “колективізм” може здійснюватися завдяки оберненим зв'язкам. Ваги змінюються на кожному кроці навчання, і їх змінювання залежить як від сусідніх вхідних образів, так і ймовірного розподілення, за яким пропонуються для навчання допущені вхідні образи.

Означення 4.7. *Навченим станом нейронної сітки* є стан, при якому ШНС реалізує вимагану функцію.

4.2. Методи навчання штучних нейронних сіток

Навчання з учителем. Звичайно, навчання нейронної сітки здійснюється значно складніше, ніж навчання одного нейрона. На практиці використовують різні евристичні підходи, що забезпечують наближене розв'язання задачі навчання нейронної сітки за прийнятний час.

Розрізняють *навчання з учителем* (supervised learning) та *навчання без учителя* (unsupervised learning). Перший метод навчання припускає, що для кожного вхідного вектора існує цільовий вектор,

уявляючий собою вимаганий вектор на виході НС, тобто є “учитель”, який задає навчаючі пари: “вхідний вектор – вектор на виході”. При цьому для кожного навчаючого вектора на вході НС обчислюється вектор на виході сітки, останній порівнюється з відповідним вимаганим (цільовим) його значенням, і різниця (похибка) за допомогою зворотного зв’язку подається в сітку для корегування ваг у відповідності до алгоритму, який прагне мінімізувати похибку. Вектори навчаючої множини пред’являються послідовно, обчислюються похибки, і ваги підналагоджуються для кожної навчаючої пари доти, доки похибка для таких пар не досягне вимаганого (як правило, низького) рівня.

Не дивлячись на численні прикладні досягнення, навчання з учителем піддається критиці через неправдоподібність з біологічної точки зору. Важко уявити навчаючий механізм у вигляді зовнішнього “учителя” мозку, який би порівнював бажані та дійсні значення виходів, виконуючи корекцію за допомогою від’ємного зворотного зв’язку.

Навчання без учителя. Більш природним є навчання без учителя, коли нейросітці пред’являються тільки вектори вхідних сигналів, і НС сама, використовуючи деякий алгоритм навчання, налаштовувала б ваги так, щоб при пред’явленні їй досить близьких вхідних векторів сигнали на виході були б однаковими. У цьому випадку в процесі навчання виділяються статистичні властивості множини навчальних вхідних векторів і відбувається об’єднання близьких (подібних) векторів у класи. Пред’явлення сітці вектора з даного класу викликає її визначену реакцію, яка до навчання є непередбаченою. Тому в процесі навчання виходи НС повинні трансформуватися в деяку зрозумілу форму. Це не є серйозною проблемою, тому що, як правило, нескладно ідентифікувати зв’язок між вхідними векторами і відповідною реакцією сітки.

Підкріплюване навчання. Існує ще один вид навчання - *підкріплюване* (reinforcement learning), при якому також передбачається наявність учителя, що не підказує, однак, сітці правильної відповіді. Учитель тільки повідомляє, чи вірно відпрацювала НС пред’явлений образ. На підставі цього сітка корегує свої параметри, збільшуючи значення ваг зв’язків, що вірно реагують на вхідний сигнал, і зменшуючи значення інших ваг. Хоча даний вид навчання і був досліджений у ряді присвячених нейросітковій проблематиці наукових праць, він не отримав значної популярності.

Конкурентне навчання. Вид навчання, за яким всі нейрони одного шару є конкуруючими, а перевага віддається нейрону, який в більшій мірі реагує на вхідний подразниковий сигнал і вага якого тільки й підлягає підналагодженню, дістав назву *конкурентного навчання*. Нейрон, вага якого при цьому переналагоджується, в свою чергу дістав назву *нейрона-переможця* (winner-takes-all).

Правило ж підналагоджування ваг звичайно являє собою деяку модифікацію правила Хеба. У якості початкових значень вагових коефіцієнтів вибираються малі, відмінні від нуля і різні величини. При пред'явленні навчального образу реакція одного з нейронів буде найбільш сильною. Його вагові коефіцієнти підсилюються або змінюються таким чином, щоб найбільш повно відповідати пред'явленому образу, ваги ж інших (неактивних) нейронів або не змінюються, або зменшуються. Процес навчання закінчується, коли вага активного нейрона дорівнюватиме загальній сумі ваг нейронів одного шару.

Реалізація цього виду навчання є багатоваріантною.

Стохастичне навчання. Розглянуті вище методи навчання є *детерміністськими*, у яких на кожному такті за визначеним алгоритмом відбувається корекція ваг, заснована на порівнянні значень вхідних і вихідних бажаних і фактичних сигналів.

Стохастичні методи навчання базуються на псевдовипадкових змінах ваг зі збереженням тих змін, що ведуть до покращень. Для корекції ваг використовується деяка ймовірнісна функція. Великі первісні випадкові корекції зі збереженням визначених змін ваг поступово зменшують, досягаючи при цьому мети навчання. Це нагадує процес відпалу металу, коли атоми розплавленого металу, що знаходяться у хаотичному русі, при його поступовому охолодженні втрачаючи енергію, досягають нижчого з можливих енергетичних станів (глобального мінімуму). Тому для опису цього виду навчання часто використовують термін "*імітація відпалу*" (simulated annealing).

Таке стохастичне навчання використовується у машинах Больцмана та Коші.

Градiєнтні методи навчання. Багато методів навчання базуються на мінімізації деякої цільової (вартісної, енергетичної) функції I , що являє собою звичайно деяку опуклу функцію. Якщо використовувані функції активації $f_{ак}(\bullet)$ є диференційовуваними, зручно застосовувати *градієнтні методи мінімізації*. У цьому випадку корекція ваг зв'язку між i -м та j -м нейронами відбувається за правилом

$$\Delta w_{ij} = \beta \frac{\partial I}{\partial w_{ij}}, \quad (4.1)$$

де β – коефіцієнт, що впливає на швидкість навчання.

Прикладами такого вигляду навчання є *дельта-правило Відроу-Хоффа* і найбільш широко застосовуваний алгоритм АЗПП. Дельта-правило є оптимальним у визначеному сенсі для одношарових ШНС через те, що за його допомогою визначається вагова матриця, яка точно відтворює реакцію нейрона у випадку лінійно незалежних вхідних векторів. У багатошарових ШНС це правило не можна застосовувати, тому що незрозуміло, яким чином налаштовувати вагові коефіцієнти нейронів схованих шарів, що зменшували б похибку сигналу на виході НС. Ця проблема зветься “*credit assignment-проблемою*” і вирішується за допомогою алгоритму зворотного поширення помилки. Даний алгоритм може бути застосований до ШНС, що має будь-яку кількість схованих шарів. Відповідно до цього алгоритму шляхом мінімізації вихідної помилки спочатку підналагоджуються ваги останнього (на виході) шару, потім попереднього і так далі до вхідного шару. Однак, якщо дельта-правило гарантує збіжність процесу навчання, то алгоритм зворотного поширення помилки такої гарантії не надає, і навчання може застрівати в одному з локальних мінімумів.

4.3. Правила навчання нейронних сіток

Правило Д. Хебба. Як вище було показано, нейронна сітка являє собою двійкові функції, що відображають вхідні і вихідні вектори. Ці функції можуть бути математичними залежностями, способом дій, що керують інструкціями для роботів, тощо. Для побудови такої сітки необхідні вказівки з розрахунку вагових коефіцієнтів, правил розрахунку яких для більшості застосувань немає. Тому необхідні спроби пошуку різних методик, які дали б змогу виконувати розрахунки ваг для сітки з заданими функціями.

Дональд Хебб [131] висловив гіпотезу про те, що мозок збільшує підсилювальні властивості синапсів тоді, коли нейрони перед і за синапсами одночасно активізовані. У більш строгому формулюванні гіпотеза Хебба говорить:

Гіпотеза Хебба: *властивість синапсу (посилення чи уповільнення) змінюється пропорційно добутку перед- і післясинаптичної активності.*

Якщо w_{ij} вага в графічній сітці від входу x_i до виходу y_i , то алгоритм реалізації цього правила для НС у момент надходження вхідного сигналу можна представити математичною формою:

$$\Delta w_{ij} = \sigma x_i y_j \text{ (правило Хебба),} \quad (4.2)$$

де σ – коефіцієнт пропорційності (навчання), $\sigma > 0$; x_i – вхід; y_j – вихід.

В алгоритмах навчання правило Хебба використовується з незначними змінами. Замість виходу y_j приймають Δy_j , при цьому величина Δy_j дорівнює різниці між бажаним і фактичним виходами, на момент навчання. При цьому одержимо так зване *дельта-правило*:

$$\Delta w_j = \sigma x_i \Delta y_j. \quad (4.3)$$

Як бачимо, подальша корекція ваги не буде потрібна, якщо отриманий вихід узгоджується з бажаним.

В разі необхідності практичного застосування НС у загальному випадку потрібно виконати такі заходи:

- вибрати для сітки необхідну топологію;
- задати представницькі логічні функції, а саме: вхідні вектори x_1, x_2, \dots, x_n і відповідні їм вихідні вектори y_1, y_2, \dots, y_n ;
- визначити значення ваг, при яких сітка буде реалізовувати задані функції. Для визначення ваг застосувати процедуру навчання сітки;
- після навчання сітка підготовлена до роботи і багаторазового використання.

Таким чином, розробка і застосування НС містить дві важливі фази: *навчання (Learning Phase)*; *експлуатації (Recall-Phase)*.

Відповідно до досвіду фаза навчання в плані обчислень дуже трудомістка, але її здійснюють лише один раз, а фаза експлуатації за витратами на обчислення безпроблемна і може як завгодно багато повторюватися.

Вивчення ваг виробляється звичайно в наступному порядку:

Крок 1. Для усіх ваг задаються випадкові числа.

Крок 2. Вибирається випадковий вхідний вектор x_i .

Крок 3. Розраховують з вагами на поточний момент вихідний вектор y_j .

Крок 4. Порівнюють y_j з цільовим вектором $y_{jц}$. Якщо $y_j = y_{jц}$, то повторюють процедуру кроку 2. У протилежному разі

коригують ваги за однією з прийнятих формул корекції і далі діють за кроком 2.

Як формули корекції на кроці 4 можна застосувати, наприклад, вираз (4.2) або дельта-правило (4.3).

Наведене правило навчання називають “*навчанням з учителем*”. Вказівки вчителя полягають у повідомленні на кожному кроці системі точного вихідного вектора – навчання йде під наглядом учителя, якому знайомий зміст досліджуваного предмета, і він включає його в процедуру навчання.

Інший метод навчання полягає в змінюванні ваг без знання цільової величини, і якщо при навчанні ми наближаємося до мети, воно заохочується, у протилежному разі – штрафується. При комп’ютерній процедурі оцінною функцією зважується кожний акт навчання. У разі успіху ваги змінюються, у протилежному разі залишаються попередніми. Саме цей метод і дістав назву “*навчання без учителя*” (*Unsupervised Learning*).

В остаточному підсумку існують також процедури навчання, в яких ваги самоорганізуються в аспекті задачі навчання.

Дельта-правило. Дельта-правило є модифікованою зміною правила Хебба і найчастіше використовується. Базується на простій ідеї неперервної зміни синаптичних ваг для зменшення різниці (“дельта”) між значенням бажаного d_i та поточного y_i вихідного сигналу нейрона:

$$\Delta W_{ij} = \beta x_i (d_j - y_j), \quad (4.4)$$

де β – коефіцієнт, який впливає на швидкість навчання.

За правилом мінімізується середньоквадратична похибка сітки.

З виразу (4.4) видно, що якщо різниця $(d_j - y_j)$ замала, тобто реакція j -го нейрона незначно відрізняється від бажаної, то й змінювання ваги зв’язку між цими нейронами також буде незначною.

Похибка, отримана у вихідному прошарку, перетворюється похідною передатної функції і послідовно поширюється назад на попередні прошарки для корекції синаптичних ваг. Процес зворотного поширення похибок сітки триває до досягнення першого прошарку.

При використанні правила “дельта” важливою є невпорядкованість множини вхідних даних. Для впорядкованої навчальної множини результат сітки може не збігтися до бажаної точності, і сітка буде вважатись нездатною до навчання.

Розширене дельта-правило. Розширене дельта-правило знаходить застосування у багатошарових ШНС прямого поширення другого порядку. У відповідності з цим правилом змінювання ваг здійснюється наступним чином:

$$\Delta W_{ij} = \beta x_i \delta_j, \quad (4.5)$$

де

$$\delta_j = \begin{cases} f_{akj}(x, w)(d_j - y_j) & \text{якщо } j\text{-нейрон шару на виході;} \\ f_{akj}(x, w) \sum_m \delta_m w_m & \text{в іншому випадку} \end{cases}, \quad (4.6)$$

де $f(x, w)$ – похідна активаційної функції, використовувана у j -му нейроні; індекс j використовується для позначення усіх нейронів наступних шарів, пов'язаних з j -м нейроном.

4.4. Парадигми навчання перцептрона

Парадигма навчання одношарового перцептрона. Першою моделлю нейросіток вважають перцептрон Ф. Розенблатта [47] (див. розд. 1.3). Теорія перцептронів є основою для багатьох типів ШНС прямого поширення і вони є класикою для вивчення.

Одношаровий перцептрон здатний розпізнавати найпростіші образи. Окремий нейрон обчислює зважену суму елементів вхідного сигналу, віднімає значення зсуву і пропускає результат через жорстку порогову функцію, вихід якої дорівнює +1 чи -1. В залежності від значення вихідного сигналу приймається рішення:

- +1 - вхідний сигнал належить класу A ,
- -1 - вхідний сигнал належить класу B .

На рис. 4.4 показана схема нейронів, використовуваних в одношарових перцептронах, графік передатної функції і схема вирішальних областей, створених у багатовимірному просторі вхідних сигналів.

Алгоритм навчання одношарового перцептрона дотримується виконання наступних почергових кроків:

Крок 1. Ініціалізація синаптичних ваг і зсуву: синаптичні ваги приймають малі випадкові значення w_{ij} .

Крок 2. Пред'явлення сітці нового вхідного вектора

$$\overline{X} = (x_1, x_2, \dots, x_n)$$

і бажаного вихідного сигналу d . Приймаються значення $\overline{X}_l = 1$ або 0.

Крок 3. Обчислення вихідного сигналу нейрона:

$$y(t) = f\left(\sum_{i=1}^N w_i(t) x_i(t) - \theta\right). \quad (4.7)$$

Крок 4. Змінюють ваги за співвідношенням:

$$w_{ij}^n = w_{ij}^n + \Delta w_{ij}, \quad (4.8)$$

де w_{ij}^n, w_{ij}^n – попередня і нова ваги; $\epsilon_j = y_j - f(\sum_i w_{ij} x_i)$, тобто $\epsilon_j = y_j - f(\text{net}_j)$ – різниця між цільовим і фактичним виходами у точці j ; $\Delta w_{ij} = \beta x_i \epsilon_j$; x_i, y_j – вхідний і вихідний сигнали сітки; $\beta > 0$ – мале число.

Якщо сітка приймає правильне рішення, синаптичні ваги не модифікуються.

Крок 5. Повернення до кроку 2.

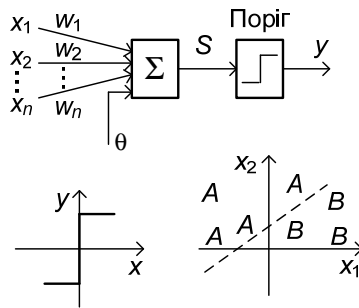


Рис. 4.4. Схема нейрона, графік передатної функції і поділяюча поверхня

Робота алгоритму завершується в момент, коли сітка для усіх вхідних векторів стане видавати правильні цільові вектори. Тут, звичайно, йдеться про варіант «навчання з учителем».

З теореми збіжності для персептрона випливає, що будь-яка класифікаційна задача, для якої при навчанні існує рішення, буде в остаточному підсумку вирішена за декілька кроків.

Збіжність процесу навчання персептрона. Якщо функцію можна подати персептроном, то алгоритм навчання сходиться за скінченну кількість кроків. У теорії нейронних сіток доведення цього виконується за глобальними функціями, які поєднують в собі інформацію про всю сітку. Ці функції порівняні з енергетичними функціями фізики. Так само, як там фізична система зазнає змін доти, поки загальна енергія буде мінімізована, так і в нейронній сітці алгоритм навчання змінює ваги доти, поки буде виконана задача навчання і

«енергетична функція» буде мінімізована.

Для персептрона ця функція, що зазнає мінімізації, вкрай проста, а саме є сумою квадратів помилок:

$$\Phi = 0,5 (\sum_i \epsilon_i^2), \quad (4.9)$$

де ϵ_i – сумарна помилка вихідних величин. Функція Φ зменшується з кожним

кроком навчання і в остаточному підсумку сходиться до 0.

Тип вхідних сигналів: бінарні чи аналогові (дійсні).

Розмірності входу і виходу обмежені при програмній реалізації тільки можливостями обчислювальної системи, на якій моделюється нейронна сітка, при апаратній реалізації – технологічними можливостями.

Області застосування: розпізнавання образів, класифікація.

Недоліки. Примітивні поділяючі поверхні (гіперплощини) дають можливість вирішувати найпростіші задачі розпізнавання.

Переваги. Програмні та апаратні реалізації моделі дуже прості. Простий і швидкий алгоритм навчання.

Парадигма навчання багатшарового персептрона. Багатшарові персептрони (рис. 4.5) дають можливість будувати більш складні поділяючі поверхні і тому більш поширені.

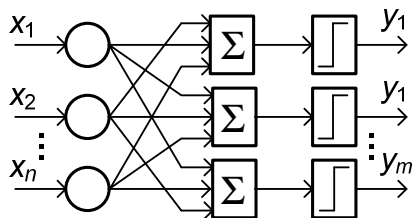


Рис. 4.5. Персептрон із багатьма виходами

Відразу після визначення недоліків персептрона Розенблатта [173; 174] розпочалися спроби створити нейросітку без властивості лінійної роздільності. Основною ідеєю являлося збільшення кількості шарів асоціативних нейронів, проте довгий час (понад десять років) не існувало алгоритму навчання таких НС. Першим алгоритмом навчання багатшарової НС був алгоритм *зворотного поширення похибки* (ЗПП) (*Error Back Propagation*). Архітектура *FeedForward BackPropagation* була розроблена на початку 1970-х років декількома незалежними авторами: Вербос (*Werbos*) [202]; Паркер (*Parker*) [178]; Румельхарт (*Rumelhart*), Хінтон (*Hinton*) та Вільямс (*Williams*) [137].

На даний час парадигма *BackPropagation* є найбільш популярною, ефективною та легкою моделлю навчання для складних багатшарових сіток. Вона використовується у різних типах застосувань і породила великий клас НС з різними структурами та методами навчання.

Модель нейрона зазнала змін. Тепер нейрони-рецептори приймають на вхід дійсні числа (зазвичай від 0 до 1), а асоціативні нейрони використовують сигмоїдальну функцію перетворення

$$f(X_{\Sigma}) = 1 / \left[1 + \left(- (X_{\Sigma} - \theta) \right) \right]. \quad (4.10)$$

Багатшаровий персептрон працює в наступному середовищі:

- областю значень функції є інтервал $(0 \dots 1)$;
- функція нескінчену кількість разів неперервно диференційовувана;
- похідна функції визначається тільки через неї саму:
 $f = f(1 - f)$.

Структура зв'язків у багатшаровому персептроні дуже схожа на структуру класичного персептрона Розенблатта. Є один шар нейронів-рецепторів і довільна кількість шарів асоційованих нейронів. Нейрони одного й того ж самого шару не зв'язані один з одним, і кожний нейрон зв'язаний з усіма нейронами наступного шару (окрім останнього шару - його виходи є виходами сітки у цілому).

Отже, типова сітка *BackPropagation* має вхідний шар, вихідний шар та принаймні один прихований шар. Теоретично, обмежень відносно числа прихованих шарів не існує, але практично застосовують один або два (рис. 4.6).

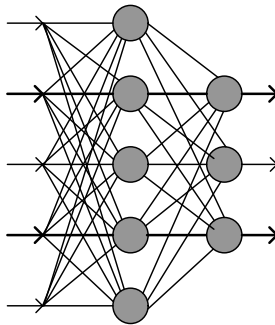


Рис. 4.6. Багатшаровий персептрон

Як видно з архітектури, k -шарова нейросітка творить ланцюжок відображень $R^{n_0} \rightarrow R^{n_i} \rightarrow \dots \rightarrow R^{n_k}$. Нейрони організовані в по-

шарову структуру з прямою передачею сигналу. Кожний нейрон сітки продукує зважену суму своїх входів, пропускає цю величину через передатну функцію і видає вихідне значення.

Існує теорема, яка стверджує, що багат шарова НС може з будь-якою заданою точністю наближати довільну кусково-неперервну та обмежену функцію (ми не будемо наводити її доведення).

Твердження 4.4. *Сітка може моделювати функцію практично будь-якої складності, причому кількість шарів і кількість нейронів у кожному шарі визначають складність функції.*

Визначення кількості проміжних шарів і кількості нейронів в них є важливим при моделюванні сітки. Більшість дослідників та інженерів, застосовуючи архітектуру до визначених проблем, використовують загальні правила, зокрема:

- кількість входів та виходів сітки визначаються кількістю вхідних та вихідних параметрів досліджуваного об'єкту, явища, процесу, тощо. На відміну від зовнішніх шарів, кількість нейронів прихованого шару $n_{\text{прих}}$ вибирається емпіричним шляхом. В більшості випадків достатня кількість нейронів становить $n_{\text{прих}} \leq n_{\text{вх}} + n_{\text{вих}}$, де $n_{\text{вх}}$, $n_{\text{вих}}$ – кількість нейронів у вхідному і, відповідно, у вихідному шарах;

- якщо складність у відношенні між отриманими та бажаними даними на виході збільшується, кількість нейронів прихованого шарі повинна також збільшитись;

- якщо процес, що моделюється, може розділятися на багато етапів, потрібен додатковий прихований шар (шари). Якщо процес не розділяється на етапи, тоді додаткові шари можуть допустити пере-запам'ятовування і, відповідно, невірне загальне рішення.

Після того, як визначено кількість шарів і число нейронів в кожному з них, потрібно знайти значення для синаптичних ваг і порогів сітки, які спроможні мінімізувати похибку спродукованого результату. Саме для цього існують алгоритми навчання, де відбувається підгонка моделі сітки до наявних навчальних даних. *Похибка для конкретної моделі* сітки визначається шляхом проходження через сітку всіх навчальних прикладів і порівняння спродукованих вихідних значень з бажаними значеннями. *Множина похибок* створює функцію похибок, значення якої можна розглядати як похибку сітки. В якості функції похибок найчастіше використовують *суму квадратів похибок*.

Для кращого розуміння алгоритму навчання сітки *Back-Propagation* потрібно роз'яснити поняття поверхні станів. Кожному значенню синаптичних ваг і порогів сітки (вільних параметрів моделі кількістю N) відповідає один вимір в багатовимірному просторі. $N+1$ -ий вимір відповідає похибці сітки. Для різноманітних сполучень ваг відповідну похибку сітки можна зобразити точкою в $N+1$ -вимірному просторі, всі ці точки утворюють деяку поверхню - поверхню станів. Мета навчання нейросітки полягає в знаходженні на багатовимірній поверхні найнижчої точки.

Поверхня станів має складну будову і досить неприємні властивості, зокрема, наявність локальних мінімумів (точки, найнижчі в своєму певному околі, але вищі від глобального мінімуму), пласкі ділянки, сідлові точки і довгі вузькі яри. Аналітичними засобами неможливо визначити розташування глобального мінімуму на поверхні станів, тому навчання нейросітки по суті полягає в дослідженні цієї поверхні. Відштовхуючись від початкової конфігурації ваг і порогів (від випадково обраної точки на поверхні), алгоритм навчання поступово відшукує глобальний мінімум. Обчислюється вектор градієнту поверхні похибок, який вказує напрямок найкоротшого спуску по поверхні з заданої точки. Якщо трошки просунутись по ньому, похибка зменшиться. Зрештою алгоритм зупиняється в нижній точці, що може виявитись лише локальним мінімумом (в ідеальному випадку - глобальним мінімумом). Складність тут полягає у виборі довжини кроків. При великій довжині кроку збіжність буде швидшою, але є небезпека перестрибнути рішення, або піти в неправильному напрямку. При маленькому кроці, правильний напрямок буде виявлений, але зростає кількість ітерацій. На практиці розмір кроку береться пропорційним крутизни схилу з деякою константою - швидкістю навчання. Правильний вибір швидкості навчання залежить від конкретної задачі і здійснюється дослідним шляхом. Ця константа може також залежати від часу, зменшуючись по мірі просування алгоритму.

Алгоритм діє ітеративно, його *кроки називаються епохами*. На кожній епосі на вхід сітки по черзі подаються всі навчальні приклади, вихідні значення сітки порівнюються з бажаними значеннями і обчислюється похибка. Значення похибки, а також градієнту поверхні станів використовують для корекції ваг, і дії повторюються. Процес навчання припиняється або коли пройдена визначена кількість епох, або коли похибка досягає визначеного рівня малості, або коли похибка перестає зменшуватись (користувач переважно сам вибирає потрібний критерій зупинення).

4.5. Метод зворотного поширення похибки

Опис методу градієнтного спуску або *методу зворотного поширення похибки* (МЗПП) (Backpropagation Algorithm) [137] можна інтерпретувати прикладом для тришарової НС, наведеним на рис. 1.10, б.

Означення 4.8. *Навчання з використанням алгоритму зворотного поширення похибки* – це навчання, яке, використовуючи поширення принципів дельта-правила, реалізує градієнтний метод опуклого функціоналу похибки в багатшарових ШНС із структурою прямого поширення на моделях ШН з диференційованими функціями активації.

Отже, метод ЗПП застосовний тільки тоді, коли всі функції нейронів сітки неперервні та диференційовувані.

Твердження 4.5. *Мета навчання ШНС за алгоритмом АЗПП* полягає в знаходженні на багатовимірній поверхні найнижчої точки, яка відповідає глобальному мінімуму.

Оскільки зворотне поширення похибки є градієнтним методом, його основний недолік полягає в тому, що він може збігатися до локального мінімуму або, при певних умовах, не збігатися зовсім. Крім того, збіжність методу сильно обумовлена вибором початкових умов оптимізації.

Узагальнена процедура методу ЗПП. Алгоритм зворотного поширення похибки застосовується для багатшарового персептрона. У сітки є безліч входів (*Inputs*), безліч виходів (*Outputs*) та безліч внутрішніх вузлів. Пронумеруємо всі вузли (входи і виходи включно) числами від 1 до N (наскрізна нумерація, поза залежності від топології шарів). Позначимо через вагу w_{ij} , розташовану на ребрі, з'єднання i -го та j -го вузлів, а через O_i – вихід i -го вузла. Якщо нам відомий навчаючий приклад (вірні відповіді сітки), то функція похибки, отримана за методом найменших квадратів, виглядає так:

$$E(\{w_{ij}\}) = \frac{1}{2} \sum_{k \in \text{Outputs}} (t_k - O_k)^2 \quad (4.11)$$

Твердження 4.6. *Похибка для конкретної топології ШНС з варіюваними значеннями синаптичних ваг і порогів сітки (так званих вільних параметрів топології ШНС) визначається шляхом проходження через неї всієї навчальної множини і порівняння спродукованих на виході значень з цільовими/бажаними. Множина похибок утворює функцію похибок, яку можна розглядати як похибку сіт-*

ки. В якості функції похибок найчастіше використовують суму квадратів похибок.

Щоб «дістатися» до мінімуму похибки, нам необхідно «рухатися» в бік, протилежний градієнту, тобто, на підставі кожної групи вірних відповідей, до кожної ваги w_{ij} додавати величину:

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}}, \quad 0 < \eta < 1.$$

Спочатку відзначимо, що w_{ij} впливає на вихід сітки тільки як частка суми $S_j = \sum_i w_{i,j} x_i$, де сума береться по входах j -го вузла. Тоді

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial S_j} \frac{\partial S_j}{\partial w_{i,j}} = x_i \frac{\partial E}{\partial S_j}; \quad (4.12)$$

$$\begin{aligned} \frac{\partial E}{\partial S_j} &= \frac{\partial E}{\partial O_j} \frac{\partial O_j}{\partial S_j} = \left(\frac{\partial}{\partial O_j} \frac{1}{2} \sum_{k \in \text{Outputs}} (t_k - o_k)^2 \right) \left(\frac{\partial \sigma(S_j)}{\partial S_j} \right) = \\ &= \left(\frac{1}{2} \frac{\partial}{\partial O_j} (t_k - o_k)^2 \right) (O_j (1 - O_j)) = -O_j (1 - O_j) (t_j - O_j). \end{aligned} \quad (4.13)$$

Якщо ж j -й вузол – не на останньому рівні, то у нього є виходи. Позначимо їх через «Нашадаок» (j). В цьому випадку

$$\frac{\partial E}{\partial S_j} = \sum_{k \in \text{Нашадаок}(j)} \frac{\partial E}{\partial S_k} \frac{\partial S_k}{\partial S_j}; \quad (4.14)$$

$$\frac{\partial S_k}{\partial S_j} = \frac{\partial S_k}{\partial O_j} \frac{\partial O_j}{\partial S_j} = w_{j,k} \frac{\partial O_j}{\partial S_j} = w_{j,k} O_j (1 - O_j). \quad (4.15)$$

Ну а $\frac{\partial E}{\partial S_j}$ – це в точності аналогічна поправка, але обчислена

для вузла наступного рівня. Оскільки ми навчилися обчислювати поправку для вузлів останнього і виражати поправку для вузла більш низького рівня через поправки більш високого, можна приступати до написання алгоритму. Саме через цю особливість обчислення поправок алгоритм називається алгоритмом зворотного поширення похибки (АЗПП).

Метод ЗПП для тришарової ШНС. Нижче наведено опис методу для тришарової сітки (рис. 1.10), але формально наведені міркування можна легко поширити на n -шарові сітки. У розглянутій сітці маємо: вхідний вектор \bar{x} , вектор \bar{k} схованого шару і вихідний

вектор \bar{y} . Для цих векторів вище були отримані співвідношення

$$\bar{y} = \bar{f}(W_1 \bar{k}); \quad (4.16)$$

$$\bar{k} = \bar{f}(W_2 \bar{x}); \quad (4.17)$$

де W_1, W_2 – матриці, які ми будемо диференціювати.

Для диференціювання як активаційну вибираємо сигмоїдну функцію

$$f(x) = \frac{1}{(1 + \exp(-cx))}, \quad (4.18)$$

де $c > 0$.

Формально при виведенні процедури навчання поводяться як для персептрона: ваги вибираються так, щоб функція помилки була мінімальною. Її подають у вигляді

$$E = (1/2) \sum_1 (z_1 - y_1), \quad (4.19)$$

де z_1 – задані цільові величини досліджуваної функції; y_1 – фактичні значення вихідного сигналу сітки, які у цілому забезпечують суму всіх квадратів помилки. Чим менше величина E , тим краще сітка, і при $E=0$ (абсолютний мінімум) сітка працює відмінно.

Зміни ваг у процесі навчання вибираються так, щоб E на кожному кроці зменшувалася, тобто процес повинний виходити на одержання мінімуму. Математично це формулюється в такий спосіб. Нехай w_{ij}^1 — елементи матриці W_1 у (4.16) і w_{ij}^2 — елементи матриці W_2 у (4.17). Тоді кожен крок навчання полягає в тому, щоб ці елементи матриць (ваги) змінювалися на прирощення

$$\Delta w_{ij}^1 = \beta \epsilon_j k_j, \quad (4.20)$$

$$\Delta w_{ij}^2 = \beta \sum_m \bar{x}_m w_{mi}^1 \bar{x}_i. \quad (4.21)$$

Тут x_m – вхідні величини; k_j – сховані величини; $\epsilon_j = z_j - y_j$ – помилка (різниця між цільовою і фактичною величинами); β – коефіцієнт навчання ($\beta > 0$ – вибирається довільно). У результаті функція помилки E поступово зменшується, кроки навчання змінюють сітку в напрямку мінімуму (локального). Розширена формула корекції відповідає формулі корекції для одношарового персептрона.

Для доведення використовується метод градієнтного спуску, при якому функцію помилки E диференціюють і показують, що при прийнятих змінах ваг величина $\Delta E < 0$ і, отже, E зменшується.

Реалізація *алгоритму навчання Backpropagation* (методу зворотної передачі помилки) виконується так:

Крок 1. Ініціалізація сітки: вагові коефіцієнти і зсуви сітки приймають малі випадкові значення.

Крок 2. Задання випадкових образів – визначення елемента навчальної множини: пари “вхід – вихід” для досліджуваної функції. Входи (x_1, x_2, \dots, x_N) , повинні розрізнятися для всіх прикладів навчальної множини; розрахунок зайнятості k_j схованих шарів.

Крок 3. Обчислення вихідного сигналу:

$$S_{i_m} = \sum_{i_{m-1}}^{N_{m-1}} w_{i_m i_{m-1}} y_{i_{m-1}} - b_{i_m}, \quad y_{i_m} = f(S_{i_m}), \quad i_m = 1, 2, \dots, N_m, \quad (4.22)$$

$$m = 1, 2, \dots, L,$$

де S – вихід суматора, w – вага зв’язку, y – вихід нейрона, b – зсув, i – номер нейрона, N – кількість нейронів у шарі, m – номер шару, L – кількість шарів, f – передатна функція.

Крок 3. Для заданих вхідних x_i і цільових z_i величин виконані корегування ваг за формулами (4.20, 4.21) або

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \epsilon_j y_i, \quad (4.23)$$

де w_{ij} – вага від нейрона i або від елемента вхідного сигналу i до нейрона j у момент часу t ; y_i – вихід нейрона i ; α – швидкість навчання; ϵ_j – значення похибки для нейрона j .

Якщо нейрон з номером j належить останньому шару, тоді

$$\epsilon_j = y_j(1 - y_j)(d_j - y_j),$$

де d_j – бажаний вихід нейрона j ; y_j – поточний вихід нейрона j .

Якщо нейрон з номером j належить одному з шарів з першого по передостанній, тоді

$$\epsilon_j = y_j(1 - y_j) \sum_k \epsilon_k w_{jk}, \quad (4.24)$$

де k пробігає всі нейрони шару з номером на одиницю більше, ніж у того, якому належить нейрон j .

Зовнішні зсуви нейронів b налаштовуються аналогічно.

Крок 4. Повернення до кроку 2.

Цей метод має ту перевагу, що для нього існує математична формалізація, яку можна застосовувати для будь-якої сітки, і не потрібно враховувати особливостей досліджуваних функцій. На жаль, кількість кроків навчання найчастіше буває значною, через що фаза навчання в обчислювальному аспекті є трудомісткою.

Хоча для тришарових сіток алгоритм зворотної передачі дає змогу подавати будь-які можливі функції, він у ряді випадків дає неправильні розв'язки. Істотний недолік цього методу в тому, що функція помилки підлягає мінімізації. При цьому, як при будь-якій мінімаксній задачі, можна попадати на локальний мінімум, тоді як для навчальної функції потрібен абсолютний мінімум. У випадках локального мінімуму одержувані величини часто близькі до шуканого значення, але все ж відрізняються від точного значення.

Іншою проблемою є *вибір кількості схованих вузлів сітки*. При виборі малої кількості можливе положення, коли навчальну функцію неможливо подати через недостатню потужність сітки. При збільшенні кількості нейронів у схованому шарі збільшується кількість незалежних змінних функції помилок, а це призводить до збільшення кількості побічних мінімумів, причому програма приводить до найближчого з них. Поряд з цим наростання кількості схованих точок збільшує час розрахунків у фазі навчання, яке і так досить великий.

Для більшості зазначених ускладнень немає теоретичних вказівок, які могли б допомогти запобігти розвитку помилкового ходу процесу навчання. Залишається лише експериментальний контроль, тобто виконання ряду пробних тестів. Для прискорення збіжності для корекції ваг w_{ij} користаються формулою

$$w_{ij}^n = \mu w_{ij}^{n-1} + (1 - \mu) \Delta w_{ij}, \quad (4.25)$$

де w_{ij}^{n-1}, w_{ij}^n – попередня і нова ваги; μ – додатне число, яке на початку процесу навчання близьке до нуля, потім у ході навчання збільшується і наприкінці наближається до одиниці; Δw_{ij} – корекція ваги.

Очевидно, що спочатку коефіцієнти корекції ваги чинять сильний вплив, який поступово послаблюється з поліпшенням значень ваг. Існує строге доведення збіжності методу зворотної передачі помилки [150].

Ефективного зниження кількості кроків навчання можна досягти, якщо вхідні і вихідні вектори підібрати так, щоб подібні вхідні вектори відтворювали також подібні вихідні вектори. Відображення з такими особливостями у певному сенсі є неперервним.

Насамкінець зазначимо, що існує багато прикладів застосування сіток, у яких вхідними чи вихідними величинами служать не (0;1) біт-послідовності, а реальні числа x в діапазоні $0 \leq x \leq 1$.

Тип вхідних сигналів: цілі чи дійсні.

Тип вихідних сигналів: дійсні з інтервалу, заданого передатною функцією нейронів.

Тип передатної функції: сигмоїдальна. В нейронних сітках застосовуються кілька варіантів сигмоїдальних передатних функцій (див. розд. 1.2): *функція Фермі* (експонентна сигмоїда); *раціональна сигмоїда*; *гіперболічний тангенс*.

Згадані функції відносяться до *однопараметричних* – значення функції залежить від аргументу й одного параметра. Також використовуються *багатопараметричні передатні функції*, наприклад:

$$f(S) = p_1 \frac{S}{|S| + p_2} + p_3, \quad (4.26)$$

де S – вхід суматора нейрона.

Сигмоїдальні функції є монотонно зростаючими і мають відмінні від нуля похідні по всій області визначення. Ці характеристики забезпечують правильне функціонування і навчання сітки.

Області застосування: розпізнавання образів, класифікація, прогнозування.

Недоліки. Багатокритеріальна задача оптимізації в методі зворотного поширення розглядається як набір однокритеріальних задач – на кожній ітерації відбуваються зміни значень параметрів сітки, що покращують роботу лише з одним прикладом навчальної вибірки. Такий підхід істотно зменшує швидкість навчання.

Переваги. Зворотне поширення – ефективний та популярний алгоритм навчання багат шарових нейронних сіток, з його допомогою вирішуються численні практичні задачі.

Після створення методу ЗПП було запропоновано багато інших методів: стохастичні, генетичні тощо. Всі вони об'єднуються ідеєю, вперше використаною у методі ЗПП: *підхід до навчання НС як до задачі багатовимірної оптимізації*.

На сьогоднішній день багат шаровий персептрон - одна з самих популярних і використовуваних НС. Причин для успіху чимало. Одна з головних переваг багат шарового персептрона – це можливість розв'язувати алгоритмічно нерозв'язувані задачі, для яких можливо скласти репрезентативний набір розв'язань. Багат шаровий персептрон функціонує дуже схоже на людину. При навчанні НС за рахунок своєї внутрішньої побудови виявляє закономірності у зв'язках вхідних образів з образами на виходах, тим самим як би “узагальнює” отриманий на навчаючій вибірці досвід. У цій спроможності до узагальнення і складається основа привабливості багато-

шарового персептрона. Ми можемо і не знати, яка залежність між вхідними образами та образами на виході, досить мати великий набір векторів, для яких відомий очікуваний результат.

Парадигма навчання *Delta Bar Delta* (модифікація парадигми навчання *BackPropagation*). Парадигма *Delta bar Delta* була розроблена Робертом Джекобсом (*Robert Jacobs*) [185], для покращення оцінки навчання стандартних сіток *FeedForward* і є модифікацією сітки *BackPropagation*.

Процедура *BackPropagation* базується на підході найкрутішого спуску, який мінімізує похибку передбачення сітки під час процесу змінювання синаптичних ваг. Стандартні оцінки навчання застосовуються на базисі “шар за шаром”, і значення моменту призначаються глобально. Моментом вважається фактор, що використовується для згладжування оцінки навчання. Момент додається до стандартної зміни ваги і є пропорційним до попередньої зміни ваги. Збіжність процедури занадто повільна для використання.

Парадигма *Delta bar Delta* має “неформальний” підхід до навчання штучних сіток, при якому *кожна вага має свій власний самонадаптований фактор навчання, і попередні значення похибки використовуються для обчислення майбутніх значень*. Знання ймовірних похибок дозволяє сітці робити інтелектуальні кроки при змінюванні ваг, але процес ускладнюється тим, що кожна вага може мати зовсім різний вплив на загальну похибку. *Джекобс запропонував поняття “здорового глузду”, коли кожна вага з'єднання сітки повинна мати власну оцінку навчання, а розмір кроку, що призначений до одної ваги з'єднання, не застосовується для всіх ваг у шару.*

Оцінка навчання ваги з'єднання змінюється на основі інформації про біжучу похибку, знайденої із стандартної *BackPropagation*. Якщо локальна похибка має однаковий знак для декількох послідовних часових кроків, оцінка навчання для цього з'єднання лінійно збільшується. Якщо локальна похибка часто змінює знак, оцінка навчання зменшується геометрично, і це гарантує, що оцінки навчання з'єднання будуть завжди додатними. *Оцінкам навчання дозволено змінюватись в часі.* Призначення оцінки навчання до кожного з'єднання, дозвіл цій оцінці навчання неперервне змінюватись з часом обумовлюють зменшення часу збіжності.

З дозволом різних оцінок навчання для кожної ваги з'єднання у сітці, пошук найкрутішого спуску може не виконуватись. Замість цього, ваги з'єднань змінюються на основі часткових похідних похиб-

ки відносно самої ваги і оцінці “кривини поверхні похибки” поблизу біжучої точки значення ваги. Зміни ваг відповідають обмеженню місцевості і вимагають інформацію від нейронів, з якими вони з’єднані.

Переваги. Парадигма *Delta Bar Delta* є спробою прискорити процес збіжності алгоритму зворотного поширення *за рахунок використання додаткової інформації про зміну параметрів і ваг під час навчання*.

Недоліки:

- навіть невелике лінійне збільшення коефіцієнта може привести до значного зростання швидкості навчання, що викликає стрибки в просторі ваг.
- геометричне зменшення коефіцієнта іноді виявляється не досить швидким.

Парадигма навчання за алгоритмом Extended Delta Bar Delta (розширення Delta Bar Delta). Елі Мінаї (Ali Minai) та Рон Вільямс (Ron Williams) [170] розробили алгоритм *Extended Delta bar Delta* як природний наслідок роботи Джекобса. В алгоритм вбудовується пам’ять з особливістю відновлення. Після кожного представлення навчальних даних епохи оцінюється накопичена похибка. Якщо похибка є меншою за попередню мінімальну похибку, ваги зберігаються у пам’яті як найкращі на цей час. Параметр допуску керує фазою відновлення. У випадку, коли поточна похибка перевищує мінімальну попередню похибку, модифіковану параметром допуску, всі значення ваг з’єднань стохастично повертаються до збереженої у пам’яті найкращої множини ваг.

Парадигма навчання “скерованим випадковим пошуком”. Парадигма скерованого випадкового пошуку (*Directed Random Search*), використовує стандартну архітектуру багатошарового персептрона з методом навчання *FeedForward*, яка не базується на алгоритмі *BackProragation* і коректує ваги випадковим чином. Для забезпечення порядку в такому процесі до випадкового кроку додається компонента напрямку, яка гарантує скерування ваг до попередньо успішного напрямку пошуку. Вплив на всі нейрони здійснюється окремо.

Для збереження ваг всередині компактної області, де алгоритм працює добре, встановлюють верхню межу величини ваги. Встановлюючи межі ваг великими, сітка може продовжувати працювати, оскільки справжній глобальний оптимум лишається невідомим. Іншою особливістю правила навчання є початкова відмінність у ви-

падковому розподілі ваг. У більшості комерційних пакетів існує рекомендоване розробником число для параметра початкової відмінності.

Парадигма випадкового пошуку має декілька важливих рис. Вона є швидкою та легкою у використанні, найкращі результати отримують, коли початкові ваги знаходяться близько до найкращих ваг. Швидкою парадигма є завдяки тому, що для проміжних нейронів похибки не обчислюються, а обчислюється лише вихідна похибка. Алгоритм надає ефект лише в малій за потужністю сітці, оскільки при збільшенні числа з'єднань, процес навчання стає довгим та важким.

Існує чотири ключові компоненти сітки з випадковим пошуком. Це є випадковий крок, крок реверсування, скерована компонента та самокорегуюча відмінність.

Випадковий крок. До кожної ваги додається випадкова величина. Вся навчальна множина пропускається через сітку, створюючи "похибку передбачення". Якщо нова загальна похибка навчальної множини є меншою за попередню найкращу похибку передбачення, біжучі значення ваг, які включають випадковий крок стають новою множиною "найкращих" ваг. Біжуча похибка передбачення зберігається як нова, найкраща похибка передбачення.

Крок реверсування. Якщо результати випадкового кроку є гіршими за попередні найкращі, випадкова величина віднімається від початкового значення ваги. Це створює множину ваг, які знаходяться в протилежному напрямку до попереднього випадкового кроку. Якщо загальна "похибка передбачення" є меншою за попередню найкращу похибку, біжучі значення ваг та біжуча похибка передбачення зберігаються як найкращі. Якщо і прямий і зворотній кроки не покращують результат, до найкращих ваг додається повністю нова множина випадкових значень і процес починається спочатку.

Скерована компонента. Для збіжності сітки створюється множина скерованих компонент, отриманих по результатах прямого та зворотного кроків. Скеровані компоненти, що відображають ланку успіхів або невдач попередніх випадкових кроків, додаються до випадкових компонент на кожному кроці процедури і забезпечують елемент "здорового глузду" до пошуку. Доведено, що додавання скерованих компонент забезпечує різке підвищення ефективності алгоритму.

Самокорегуюча відмінність. Визначається параметр початкової відмінності для керування початковим розміром випадкових кроків, що додається до ваг. Адаптивний механізм змінює параметр від-

мінності, який базується на біжучій оцінці успіху або невдачі. Правило навчання припускає, що біжучий розмір кроків для ваг у правильному напрямку збільшується для випадку декількох послідовних успіхів. Навпаки, якщо воно виявляє декілька послідовних невдач, відмінність зменшується для зменшення розміру кроку.

Переваги. Для малих та середніх нейросіток скерований випадковий пошук надає добрі результати за короткий час. Навчання є автоматичним, вимагає невеликої взаємодії з користувачем.

Недоліки. Кількість ваг з'єднань накладає практичні обмеження на розмір задачі. Якщо сітка має більше ніж 200 ваг з'єднань, скерований випадковий пошук може вимагати збільшення часу навчання, але продукувати прийнятні рішення.

4.6. Загальні зауваження щодо навчання нейронних сіток

Дані для навчання повинні бути *репрезентативними*. Навчаюча вибірка має повно і різнобічно подавати об'єкт дослідження. Щодо *нормування і центрування* даних див. розд. 3 підручника, а в разі використання *регуляризації* – розд. 22 підручника (ознакою перенавченої НС є великі значення (порядку сотень и тисяч) ваг – така нейросітка не буде нормально функціонувати на нових даних);

Треба *розділяти вибірку на Train, Test та Validation*. Перенавчання сітки, особливо на малих вибірках, приводить до максимального підстроювання НС під вибірку і втрати працездатності на реальних даних. Тому, для того, щоб контролювати узагальнюючі спроможності моделі, треба поділити всі дані на три вибірки у співвідношенні 70: 20: 10. Навчатися треба на Train, періодично перевіряючи якість моделі на Test. Для фінальної неупередженої оцінки удаються до Validation. Техніка крос-валідації, коли Train і Test декілька раз формуються по черзі довільним способом з тих самих даних, може дати хибне уявлення про задовольняючу якість роботи системи – наприклад, якщо дані взяті з різних джерел і це є критичним. Тому треба використовувати правильний Validation.

Не треба донавчати нейронну сітку в режимі он-лайн. Для звичайних штучних нейронних сіток на сучасному етапі технічного розвитку така практика є ризикованою: сітка може перенавчитися або піднагадитися під останні дані, які надійшли – и *втратить свої узагальнюючі спроможності*. Для того, щоб систему можна було використовувати на практиці, нейронну сітку треба: а) навчити; б) протестувати якість на тестових і валідаційних вибірках; в) вибрати

вдалий варіант сітки, зафіксувати її ваги; г) використати навчену нейросітку на практиці, ваги в процесі використання не міняти.

Навчати нейронні сітки краще в MATLAB та подібних дру- жніх середовищах. Існує значна кількість програмних пакетів, в основному на MATLAB та Python, які дозволяють навчати нейронні сітки, контролюючи при цьому процес навчання і тестування зі зручними засобами візуалізації і налагодження. Крім MATLAB та Python можна використати пакет STATISTICA, який містить сучасні методи навчання нейросіток і дозволяє генерувати їх у виді програмного коду на C, зручного для імплементації.

4.7. Функція оцінки роботи сітки

У тих випадках, коли вдається оцінити роботу сітки, навчання нейросіток можна подати як задачу оптимізації.

Твердження 4.7. *Оцінити роботу ШНС* — означає вказати кількісно, добре чи погано сітка розв'язує поставлені задачі.

Для цього будується функція оцінки, яка, як правило, *явно залежить від сигналів на виході* сітки і *неявно* (через функціонування) — *від усіх її параметрів*. Найпростіший і найрозповсюдженіший приклад оцінки – сума квадратів віддалей від сигналів на виходах сітки до їх вимаганих значень:

$$H = \frac{1}{2} \sum_{\tau \in V_{out}} (Z(\tau) - Z^*(\tau))^2, \quad (4.27)$$

де $Z^*(\tau)$ – бажане значення сигналу на виході.

Контрольні завдання

1. Навести і прокоментувати три стадії життєвого циклу нейронних сіток.
2. Охарактеризувати три фундаментальні властивості, пов'язані з навчанням за прикладами: ємність, складність образів та обчислювальна складність.
3. Розкрити зміст понять “навчання”, “мета процесу навчання”, “алгоритму навчання” нейронних сіток.
4. Навести і прокоментувати послідовні процедури процесу навчання нейронних сіток.
5. Навести і прокоментувати узагальнену схему процесу навчання нейронних сіток.
6. Охарактеризувати вимоги до трьох складових набору даних (*Train*, *Test* та *Validation*) при навчанні нейронних сіток.

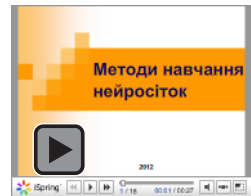
7. Розкрити три основні підходи до навчання нейронних сіток.
8. Дати характеристику основним методам навчання штучних нейронних сіток.
9. Розкрити вміст гіпотези і навести правило навчання Д. Хебба.
10. Пояснити вміст модифікації дельта-правила і його розширення.
11. Розкрити вміст парадигми навчання одношарового персептрона.
12. Розкрити вміст парадигми навчання багатошарового персептрона.
13. Охарактеризувати метод навчання зі зворотним поширенням похибки.
14. Навести узагальнену процедуру навчання за методом зворотного поширення похибки.
15. Розкрити алгоритм методу навчання зі зворотним поширенням похибки.
16. Розкрити вміст парадигми навчання Delta-Bar-Delta (модифікованого методу зворотного поширення похибки).
17. Навести парадигму навчання за алгоритмом розширення Delta-Bar-Delta.
18. Прокоментувати функцію оцінки роботи нейронної сітки.

Презентації

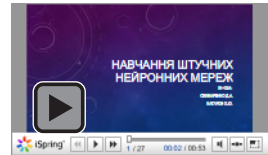
1. Нейронні сітки



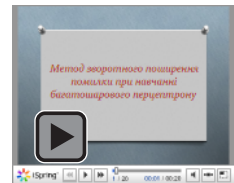
2. Методи навчання нейросіток



3. Навчання штучних нейронних мереж



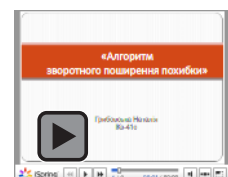
4. Метод зворотного поширення помилки при навчанні багатосарового перцептрону



5. Метод зворотного поширення похибки



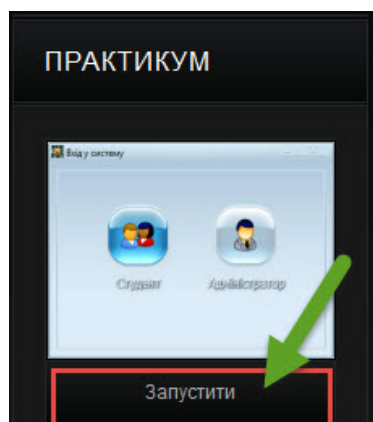
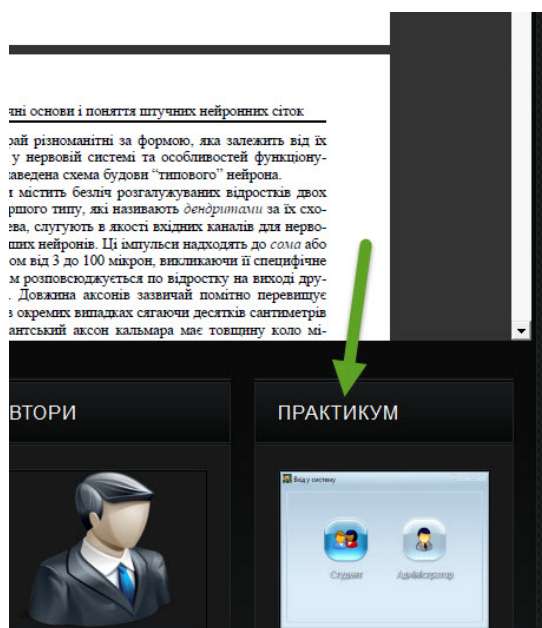
6. Алгоритм зворотного поширення похибки



Практичні заняття

Навчання нейронних сіток

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 5

Персептрон

*Іноді необхідно порушити правило,
щоб врятувати його силу*
Василь Ключевський

5.1. Визначення персептрона

Персептрон Ф. Розенблатта [47; 171] є історично першою навчуваною ШНС. Існують декілька версій персептрона. Розглянемо класичний персептрон – сітку з пороговими нейронами і вхідними сигналами, дорівнюваними нулю або одиниці.

Визначення персептрона. Персептрон має розв'язувати задачу класифікації на два класи за бінарними вхідними сигналами. Набір вхідних сигналів будемо позначати n -вимірним вектором \bar{X} . Всі елементи вектора є булівими змінними, приймаючими значення «Істина» або «Хибність»; проте іноді корисно оперувати чисельними значеннями 1 або 0, відповідно.

Твердження 5.1. Персептроном є пристрій, обчислюючий наступну функцію:

$$f = \left[\sum_{i=1}^n W_i x_i > \theta_i \right], \quad (5.1)$$

де W_i – ваги персептрона; θ_i – зсув (поріг); x_i – значення вхідних сигналів; дужки $[]$ означають перехід від булевих (логічних) до чисельних значень за наведеним вище зауваженням. В якості вхідних сигналів персептрона можуть виступати як вхідні сигнали усієї сітки (змінні x), так і значення на виході інших персептронів.

Додавши у (5.1) постійний одиничний вхідний сигнал $x_0 \equiv 1$ і поклавши $W_0 = -\theta$, персептрон можна подати у наступному виді:

$$f = \left[\sum_{i=1}^n W_i x_i > 0 \right]. \quad (5.2)$$

Отже, вираз (5.2) обчислюється одним нейроном У. Маккаллоха і В. Піттса з пороговим нелінійним перетворювачем. Каскад з декількох шарів таких нейронів дістав назву *багатошарового персептрона*. Далі у цьому розділі розглядатимуться деякі властивості персептронів. Детальне дослідження персептронів наведено в роботі [34].

5.2. Навчання персептрона

Правило Хебба. Персептрон навчають за правилом Хебба. Подаємо на вхід персептрона один приклад (образ). Якщо сигнал на виході персептрона співпадає з вірною відповіддю, то ніяких дій здійснювати не треба. У випадку похибки необхідно навчити персептрон правильно розв'язувати даний приклад. Похибки можуть бути двох типів.

Перший тип похибки – на виході персептрона 0, а вірна відповідь буде 1. Для того, щоб персептрон (5.2) видавав вірну відповідь необхідно, щоб сума у правій частині (5.2) стала більшою. Оскільки змінні x_i приймають значення 0 або 1, збільшення суми може бути досягнуто за рахунок збільшення ваг W_i . Проте нема сенсу збільшувати ваги при змінних x_i , які дорівнюють нулю. Таким чином, необхідно збільшити ваги W_i при тих змінних x_i , які дорівнюють 1. Для фіксації одиничних сигналів з x_i слід провести ту саму процедуру і на усіх інших шарах. Звідси випливає *перше правило Хебба*:

Правило Хебба 5.1: коли на виході персептрона отриманий 0, а вірна відповідь дорівнює 1, необхідно збільшити ваги зв'язків між одночасно активними нейронами. При цьому персептрон на виході вважається активним, а вхідні сигнали вважаються нейронами.

Другий тип похибки – на виході персептрона маємо 1, а вірна відповідь дорівнює нулю. Для навчання вірному розв'язку даного приклада необхідно зменшити суму у правій частині (5.2). Для цього необхідно зменшити ваги зв'язків W_i при тих змінних x_i , які дорівнюють 1 (оскільки немає сенсу зменшувати ваги зв'язків при змінних x_i , дорівнюваних нулю). Необхідно також провести цю процедуру для усіх активних нейронів попередніх шарів. В результаті отримуємо *друге правило Хебба*:

Правило Хебба 5.2: якщо на виході персептрона отримано 1, а вірна відповідь дорівнює 0, то необхідно зменшити ваги зв'язків між одночасно активними нейронами.

Таким чином, процедура навчання зводиться до послідовного перебирання усіх прикладів навчаючої множини із застосуванням правил Хебба для навчання помилково розв'язаних прикладів. Якщо після чергового циклу представлення усіх прикладів з'ясується, що усі вони розв'язані вірно, то процедура навчання завершується.

Нерозглянутими залишилися два питання. *Перше* – наскільки треба збільшувати/зменшувати ваги зв'язків при застосуванні правила Хебба. *Друге* – про збіжність процедури навчання. Відповідь на перше

запитання розглядається у розд. 4, а на друге питання приводимо результат доказу наступних теорем, наведених в роботі [34]:

Теорема 5.1 (про збіжність персептрона). Якщо існує вектор \bar{X} параметрів, за яким персептрон вірно розв'язує усі приклади з навчаючої вибірки, то при навчанні персептрона за правилом Хебба розв'язок буде знайдений за скінчену кількість кроків.

Сенс теореми полягає в тому, що якщо відносно задуманої класифікації можна знайти набір елементів, в якому існує розв'язок, то в межах цього набору він буде досягнутий за скінчений проміжок часу.

Теорема 5.2 (про «зациклювання» персептрона). Якщо не існує вектора \bar{X} параметрів, за яким персептрон вірно розв'язує усі приклади з навчаючої вибірки, то при навчанні персептрона за правилом Хебба через скінчену кількість кроків вектор ваг почне повторюватися.

Доказ цих теорем наведений в роботі [34].

Цілочисловість ваг персептронів. Розглянемо доведення наступної теореми [35]:

Теорема 5.3 (про заміну будь-якого персептрона). Будь-який персептрон (5.2) можна замінити іншим персептроном того ж виду з цілими вагами зв'язків.

Доведення. Позначимо множину прикладів одного класу (вірна відповідь дорівнює 0) через \bar{X}_0 , а іншого (вірна відповідь дорівнює 1) через \bar{X}_1 . Обчислимо максимальне і мінімальне значення суми у правій частині (5.2):

$$\varepsilon_0 = \min_{x=\bar{X}_0} - \sum_{i=1}^n W_i x_i; \quad \varepsilon_1 = \min_{x=\bar{X}_1} - \sum_{i=1}^n W_i x_i.$$

Визначимо допуск ε як мінімум з ε_0 та ε_1 . Покладемо $\delta = \varepsilon / (n+1)$, де n – кількість доданків у (5.2). Оскільки персептрон (5.2) розв'язує поставлену задачу класифікації і множина прикладів у навчаючій вибірці скінчена, то $\delta > 0$. З теорії чисел відома теорема про те, що будь-яке дійсне число можна скільки завгодно точно наблизити раціональними числами. Замінімо ваги W_i на раціональні числа так, щоб виконувалися наступні нерівності $|W_i - W'_i| < \delta$.

З цих нерівностей випливає, що при використанні ваг W'_i персептрон буде працювати з тими самими результатами, що і початковий

персептрон. Дійсно, якщо вірною відповіддю приклада є 0, маємо

$$\sum_{i=1}^n W_i x_i \leq -\varepsilon.$$

Підставивши нові ваги, отримаємо:

$$\begin{aligned} \sum_{i=1}^n W'_i x_i &= \sum_{i=1}^n (W'_i - W_i) x_i + \sum_{i=1}^n W_i x_i \leq \sum_{i=1}^n |W'_i - W_i| x_i - \\ &= -\varepsilon \leq \sum_{i=1}^n |W'_i - W_i| - \varepsilon < (n+1)\delta - \varepsilon = 0, \end{aligned}$$

і з цього випливає необхідна нерівність:

$$\sum_{i=1}^n W'_i x_i < 0. \quad (5.3)$$

Аналогічно, у випадку вірної відповіді, дорівнюваної 1, маємо

$$\sum_{i=1}^n W_i x_i \geq \varepsilon, \text{ звідки, підставивши нові ваги і поріг, отримаємо:}$$

$$\begin{aligned} \sum_{i=1}^n W'_i x_i &= \sum_{i=1}^n (W'_i - W_i) x_i + \sum_{i=1}^n W_i x_i \geq \varepsilon - \sum_{i=1}^n |W'_i - W_i| x_i \geq \\ &\geq \varepsilon - \sum_{i=1}^n |W'_i - W_i| > \varepsilon - (n+1)\delta = 0, \end{aligned}$$

і з цього випливає виконання нерівності:

$$\sum_{i=1}^n W'_i x_i > 0. \quad (5.4)$$

Нерівності (5.3) і (5.4) доводять можливість заміни усіх ваг і порога будь-якого персептрона раціональними числами. Очевидно також, що при множенні усіх ваг і порога на одне і те саме ненульове число персептрон не зміниться. Оскільки будь-яке раціональне число можна подати у як відношення цілого числа до натурального числа, отримаємо:

$$f = \left[\sum_{i=0}^n W_i x_i > 0 \right] = \left[\sum_{i=0}^n W'_i x_i > 0 \right] = \left[\sum_{i=0}^n \frac{W''_i x_i}{k_i} > 0 \right], \quad (5.5)$$

де W''_i — цілі числа. Позначимо через k добуток усіх знаменників

$k = \prod_{i=0}^n k_i$. Перемножимо усі ваги і поріг на k . Отримаємо цілочислові ваги $W'' = kW''$. З (5.3), (5.4) і (5.5) знаходимо:

$$f = \left[\sum_{i=0}^n W_i x_i > 0 \right] = \left[\sum_{i=0}^n W'_i x_i > 0 \right] = \left[\sum_{i=0}^n \frac{W''_i x_i}{k_i} > 0 \right] = \left[\sum_{i=0}^n W''_i x_i > 0 \right],$$

що і завершує доведення теореми.

Оскільки з доведеної теореми випливає, що ваги персептрона є цілими числами, то питання про вибір кроку при застосуванні правила Хебба розв'язується просто: *ваги і поріг необхідно збільшувати/зменшувати на 1.*

Проблеми “Виключного АБО”. Як вище наголошувалося, робота персептрона зводиться до класифікації (узагальнення) вхідних сигналів, які належать n -вимірному гіперпростору, за деякою кількістю класів. З математичної точки зору це відбувається шляхом розбиття гіперпростору гіперплощинами. Для одношарового персептрона:

$$\sum_{i=1}^n W_{ij} x_i = \theta_j, \quad j = 1 \dots m.$$

Кожна отримана область є областю визначення окремого класу. Кількість таких класів для персептрона не перевищує 2^m , де m – кількість його виходів. Проте не всі з класів можуть бути розділеними даною НС.

Наприклад, одношаровий персептрон, який складається з одного нейрона з двома входами, не може реалізувати логічну функцію “Виключне АБО”, тобто не здатний розділити площину (двовимірний гіперпростір) на дві напівплощини так, щоб здійснити класифікацію вхідних сигналів за класами A і B (див. *таблицю 5.1* та інтерпретацію нероздільності функції «Виключне АБО», подану рис. 5.1).

Таблиця 5.1

Логічна функція «Виключного АБО»

$x_1 \backslash x_2$	0	1
0	B	A
1	A	B

Можливість реалізації одношаровим персептроном такої функції отримала назву *проблеми “Виключного АБО”*.

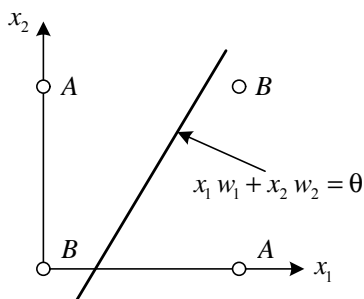


Рис. 5.1. Лінійна нероздільність функції “Виключне АБО”

Відзначимо, що функції, які не реалізуються одношаровим персептроном, називаються *лінійно нероздільними*. Розв'язання задач, що підпадають під це обмеження, полягає у застосуванні двох і більше шарових сіток або сіток з нелінійними синапсами, проте і тоді існує імовірність неможливості коректного розділення деяких вхідних сигналів.

Лінійна роздільність функцій при навчанні персептронів встановлюється наступною теоремою:

Теорема 5.4 (*про лінійну роздільність функцій*). Клас елементарних персептронів, для яких існує розв'язок до будь-якої задуманої класифікації, не є порожнім.

Ця теорема стверджує, що для будь-якої класифікації навчаючої вибірки можна підібрати такий набір (з нескінченного набору) елементарних нейронів, в якому буде здійснено розділення навчаючої послідовності за допомогою лінійного розв'язувального правила.

5.3. Багатшаровий персептрон

Двошаровість персептрона. Як вже згадувалося раніше в даному розділі, можливе використання багатшарових персептронів. Проте теореми про зациклюваність і збіжність персептрона, наведені вище, є вірними тільки при навчанні одношарового персептрона або багатшарового персептрона за умови, що навчаються тільки ті ваги персептрона, які розташовані в останньому шарі сітки. Для випадку довільного багатшарового персептрона теореми не працюють. Наступний приклад демонструє основну проблему, що виникає при навчанні багатшарових персептронів за правилом Хебба.

Нехай ваги усіх шарів персептрона в ході навчання сформувалися так, що усі приклади навчаючої множини, окрім першого, розв'язуються вірно. При цьому вірною відповіддю першого прикладу є 1. Всі вхідні сигнали персептрона останнього шару дорівнюють нулю. В такому випадку перше правило Хебба не дає результату, оскільки всі нейрони передостаннього шару не активні. Існує безліч методів, як розв'язувати цю проблему. Проте усі ці методи не є регулярними і не гарантують збіжність багатшарового персептрона до розв'язку навіть за умови, що такий розв'язок існує.

В дійсності проблема налагодження (навчання) багатшарового персептрона вирішується наступною теоремою.

Теорема 5.6 (*про двошаровість персептрона*). Будь-який багатшаровий персептрон може бути поданий у вигляді двошарового персептрона з ненавчуваними вагами першого шару.

Для доведення цієї теореми скористуємося однією теоремою з математичної логіки.

Теорема 5.7 (про диз'юнктивну нормальну форму). Будь-яка буліва функція булевих аргументів може бути представлена у вигляді диз'юнкції кон'юнкцій елементарних висловлювань і заперечень елементарних висловлювань:

$$h = \vee \left(\wedge x_i \wedge \overline{x_j} \right).$$

Нагадаємо деякі властивості диз'юнктивної нормальної форми.

Властивість 5.1. В кожний кон'юнктивний член (доданок) входять усі елементарні висловлювання або у вигляді самого висловлювання, або у вигляді його заперечення.

Властивість 5.2. При будь-яких значеннях елементарних висловлювань в диз'юнктивній нормальній формі може бути істинним не більше одного кон'юнктивного члена (доданку).

Доведення теореми про двошаровість перцептрона. З теореми про диз'юнктивну нормальну форму випливає, що будь-який багатшаровий перцептрон може бути поданий в наступному вигляді:

$$f = \left[\vee \left(\wedge x_i \wedge \overline{x_j} \right) \right].$$

З урахуванням другої властивості диз'юнктивної нормальної форми вираз (5.6) можна переписати у вигляді:

$$f = \left[\vee \left(\wedge x_i \wedge \overline{x_j} \right) \right] = \left[\sum_{i=1}^n \left(\wedge x_i \wedge \overline{x_j} \right) > 0 \right]. \quad (5.7)$$

Переведемо в арифметичну форму всі доданки у (5.7). Кон'юнкція змінюється множенням, а заперечення на різницю: $\overline{x_j} = 1 - x_j$. Виконавши цю заміну в (5.7) і звівши подібні члени, отримуємо:

$$f = \left[\sum_s W_s \prod_{i \in I_s} x_i > 0 \right], \quad (5.8)$$

де I_s – множина індексів співмножників в s -му доданку; W_s – число, яке визначає, скільки разів такий доданок зустрівся у (5.7) після заміни і розкриття дужок (кількість подібних доданків).

Замінімо i -й доданок у (5.8) перцептроном наступного виду:

$$\psi_i = \prod_{s \in I_i} x_s = \left[\sum_{s \in I_i} x_s > |I_i| - 1 \right]. \quad (5.9)$$

Підставивши (5.9) у (5.8) отримаємо вираз (5.1), тобто довільний багатошаровий персептрон представлений у вигляді (5.2) з цілочисловими коефіцієнтами. В якості персептронів першого шару використовуються персептрони виду (5.9) з ненавчуваними вагами. Теорему доведено.

Сітки Ворда. Одним із варіантів багатошарового персептрона є *сітка Ворда*, узагальнена структура якої наведена на рис. 5.2.

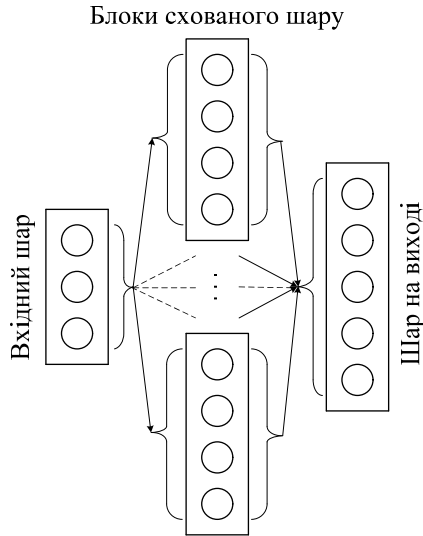


Рис. 5.2. Сітка Ворда

Означення 5.1. *Нейронна сітка Ворда* – ШНС, топологія якої характеризується тим, що внутрішні (сховані) шари нейронів розбиті на блоки.

Нейронні сітки Ворда розрізняються кількістю блоків схованого шару і наявністю або відсутністю обхідних з'єднань. Розбиття схованих шарів на блоки дозволяє використовувати всілякі передатні функції для різних блоків схованого шару. Таким чином, одні й ті самі сигнали, отримані від вхідного шару, зважуються та опрацьовуються паралельно з використанням декількох способів, а отриманий результат потім опрацьовується нейронами шару на виході. Застосування всіляких методів опрацювання для одного й того ж набору даних дозволяє стверджувати, що НС аналізує дані з різних аспектів. Практика доводить, що сітка реалізує досить добрі результати при розв'язанні задач прогнозування та розпізнавання образів. Для нейронів вхідного

шару, як правило, установлюється лінійна функція активації. Функція активації для нейронів з блоків схованого і на виході шарів визначається експериментально.

Для навчання НС Ворда можна застосовувати метод зворотного поширення похибки. Для налагодження вагових коефіцієнтів використовуються ті ж алгоритми, про які йшлося вище (розд. 5.2).

5.4. Узагальнення

Наприкінці виокремимо наступні основні властивості персептронів, які необхідно враховувати при їх використанні для розв'язання практичних задач:

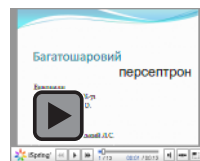
- будь-який персептрон може містити один або декілька шарів. У випадку двошарового персептрона ваги першого шару не навчаються;
- ваги будь-якого персептрона можна замінити на цілочислові;
- при навчанні за правилом Хебба після скінченої кількості ітерацій можливі два результати: персептрон навчиться або вектор ваг персептрона буде повторюватися (персептрон зациклиться).

Контрольні завдання

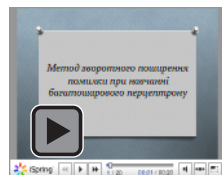
1. Дати визначення персептрона.
2. Охарактеризувати особливості навчання персептрона.
3. Пояснити джерела виникнення двох типів похибок при навчанні персептрона.
4. Прокоментувати теорему про збіжність і зациклованість навчання персептрона.
5. Прокоментувати теорему про заміну будь-якого персептрона (цілочисловість ваг персептрона).
6. Охарактеризувати проблему “виключного АБО” (лінійної нероздільності цієї функції).
7. Навести тлумачення теореми про двошаровість персептрона (з використанням теореми про диз'юнктивну нормальну форму).
8. Навести означення й охарактеризувати клас нейронних сіток Ворда.

Презентації

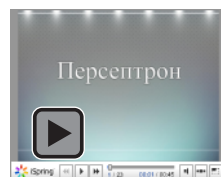
1. Багат шаровий персептрон



2. Метод зворотного поширення помилки при навчанні багат шарового персептрону



3. Персептрон



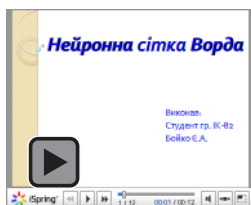
4. Персептрони та зародження штучних нейронних мереж



5. Нейронні сітки



6. Нейронна сітка Ворда



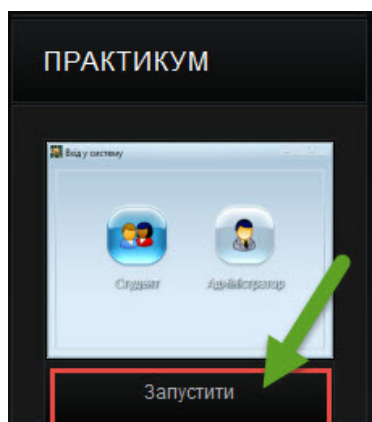
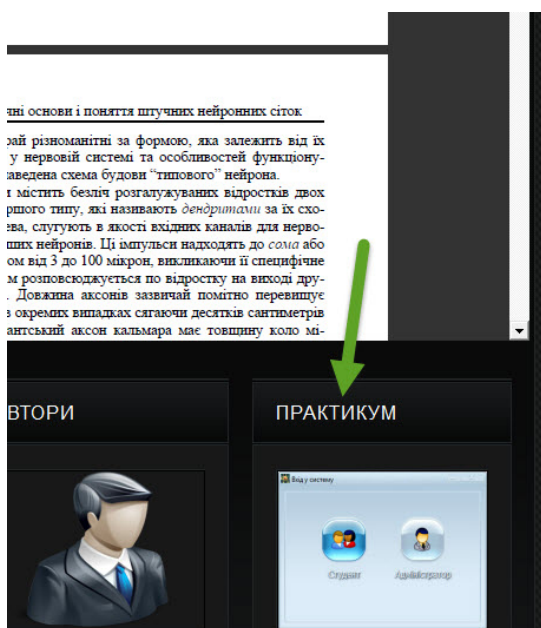
7. Персептрон



Практичні заняття

Персептрон

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Асоціативні нейронні сітки

*Часто студент перед іспитом знає предмет на двійку,
плекає надію отримати трійку, складає іспит на четвірку
і ображається, чому не поставили п'ятірку
З 50-річного досвіду викладання у ВНЗ*

Кожному — своє
Марк Тулій Цицерон

6.1. Асоціативна пам'ять нейронних сіток

Біологічні аналоги. У біологічних системах, які володіють пам'яттю, зміна нервової активності системи під впливом зовнішніх подразників залежить від дії попередніх подій та від інформації, яка зберігається у пам'яті. Процес запам'ятовування пов'язаний з утворенням сліду (узору, неграми) в мозкових структурах. Потік нервових імпульсів, які несуть інформацію про об'єкт запам'ятовування, проходить через нейронні сітки, збуджуючи на своєму шляху нервові клітини, з яких формується нейронний слід. Утворений таким чином нервовий шлях має менший опір по відношенню до інших можливих шляхів. Підвищення продуктивності нейронного сліду, що виник, можливе внаслідок властивості нейронів досить швидко адаптуватися до нервових імпульсів, які повторно проходять по тому ж самому маршруту (сліду). Механізми пам'яті забезпечують подальше зберігання збільшеної провідності нейронів, утягнутих в утворений узор.

Сліди пам'яті, які зберігають образи об'єктів, відбиваються у складних паралельно-послідовних нейронних сітках великої надмірності. В організованих випадковим чином нейронних сітках сліди пам'яті розподіляються по простору мозку також випадково.

Ситуація ще більше ускладнюється тим, що ті ж самі нейрони приймають участь у зберіганні образів різних об'єктів, які запам'ятовуються. Це означає, що слід, виникаючий при запам'ятовуванні одного об'єкта інформації, може мати загальні ланки нейронної сітки із слідами від інших об'єктів. Через це неможливо указати з впевненістю, в якій ділянці мозку знаходиться слід конкретного об'єкта інформації – образу.

У цьому випадку механізм доступу до інформації базується *не на показі місця зберігання інформації в логіко-запам'ятовуючому сере-*

довищі, а на аналізі властивостей самої шуканої інформації. В біологічних системах обробки даних таким механізмом є *механізм асоціацій*.

Вперше термін «асоціація» був уведений Дж. Локком у 1698 г. і визначений як «зв'язок, виникаючий при певних умовах між двома або більше психічними утвореннями – відчуттями, актами, сприйняттями, ідеями».

Розглядаючи природу зв'язків у мозку, Ф. Крік виокремив *прецизійні* та *асоціативні* з'єднання [65]. Перші для невеликої кількості кліток є упорядкованими. Сила зв'язків інших, характерних для значно більшої кількості кліток, неоднакова, залежить від типу розглядуваної сітки і відрегульована “на підставі досвіду” таким чином, що ті провідні шляхи, які часто активізуються укупі, підсилюються якимось чином. Саме ці з'єднання і утворюють асоціації, тобто такі зв'язки між явищами, що виклик одного з них приводить до виклику й другого.

Спосіб доступу до інформації у запам'ятовуючому середовищі, який базується на механізмі асоціації, дістав назву *асоціативного способу доступу*. Асоціативний спосіб доступу до інформації *забезпечує*:

- практично одночасний доступ до всієї інформації, яка зберігається у пам'яті;
- відносну незалежність часу пошуку інформації від ємкості пам'яті;
- занесення елементів обробки інформації безпосередньо у процесі самого доступу;
- обробку інформації безпосередньо в середовищі її зберігання.

Ці, а також ряд інших відмінних особливостей асоціативного способу доступу до інформації робить його надзвичайно перспективним у системах обробки даних.

Асоціації. Стосовно систем обробки даних і ШНС, зокрема, асоціація тлумачиться наступним чином:

Означення 6.1. *Асоціація* — це взаємозв'язок між інформацією (образом) на вході логіко-запам'ятовуючого середовища (*нейросітки*) та інформацією (образом), що зберігається в логіко-запам'ятовуючому середовищі.

Існують різні концепції *асоціативної пам'яті* (АП). Проте всі вони передбачають наявність наступних елементів:

- логіко-запам'ятовуючого середовища, яке по суті уявляє носій інформації;

- множини занесених до пам'яті інформаційних об'єктів;
- структури взаємозв'язків між інформаційними об'єктами;
- механізму інформаційних взаємодій у логіко-запам'ятовуючому середовищі.

Наведені елементи концепції АП дозволяють виявити основний підхід, відповідно до якого асоціації між інформаційними об'єктами (образами) тлумачаться як деяка абстрактна структура взаємозалежностей (відношень), наявно закодована в інформаційних об'єктах та у відповідних зв'язках між ними або у формах їх представлень.

Ці відношення містять компоненти двох типів, перші з яких задають власне інформаційні об'єкти, а другі – вид відношень. Ознаки цих відношень можуть характеризувати властивості об'єктів, дії над ними, підпорядкованість, часові ознаки тощо. Таке представлення асоціації – найбільш проста конструкція, на основі якої можна побудувати структури відношень довільної складності (подібний підхід використовується і в семантичних моделях подання знань).

В залежності від умов формування відношень між асоційованими об'єктами, асоціації можуть встановлюватися за критеріями подібності об'єктів, контрастів між ними, за суміжністю проявлень об'єктів у часі або у просторі, а також в межах визначеної сукупності властивостей, наприклад, при забезпеченні заданих просторово-часових співвідношень.

Логіка взаємодій між інформаційними об'єктами може бути реалізована на основі двох концепцій:

- формування відношень асоціації між однорідними об'єктами;
- «складового відношення», де відношення асоціації устанавлюються між якісно відмінними об'єктами.

Вид зв'язку між асоційованими об'єктами може бути різним:

- символний – зв'язок між об'єктами устанавлюється «на підставі угоди»;
- індексний – зв'язок асоціюється завдяки існуючих відношень між об'єктами;
- іконічний – зв'язок між об'єктами устанавлюється на основі фактичної подібності;
- гібридний – сполучає особливості різних видів зв'язку.

В залежності від ступеня відповідності асоційованих об'єктів розрізняють автоасоціації і гетероасоціації.

Твердження 6.1. *Автоасоціації* реалізуються за умови відповідності співставляваних об'єктів, причому об'єкт відшукується за його довільними частинами, які мають більшу або меншу кореляцію з шуканим об'єктом, або за його фрагментами у випадку, якщо вони достатні для того, щоб відрізнити цей об'єкт від інших.

Твердження 6.2. При *гетероасоціації* ініційований об'єкт структурно не відповідає жодному з шуканих об'єктів і формалізується як відповідь на специфічний ключовий об'єкт.

Асоційовані об'єкти можуть біти представленими або безпосередніми, або непрямими асоціаціями.

Твердження 6.3. *Безпосередні асоціації* за способом представлення у вигляді функціональної залежності ініціюються або логічно детермінованою послідовністю, або на підставі асоціативного зв'язку за допомогою прямих покажчиків.

Твердження 6.4. Представлення *непрямих асоціації* у вигляді функціональної залежності утворюється на основі асоціативних зв'язків за допомогою перехресних посилань.

В контексті тлумачень поняття асоціації як структури взаємозв'язків між інформаційними об'єктами відбиває наявність взаємозв'язків між даними і не має відношення до самого механізму зберігання інформації.

Асоціативні властивості у нейронній сітці. В рамках підходу, який вивчає асоціації як колективні та інтегральні змінювання у нейронній сітці (логіко-запам'ятовуючому середовищі), асоціативні властивості сітки можуть бути розглянуті з точки зору:

- по-перше, можливості колективного доступу до всієї розподіленої у НС інформації, а також паралельного опрацювання та одночасного перетворення усіх даних безпосередньо у нейронному середовищі;
- по-друге, практичної реалізації відношень між розміщеними у НС асоційованими інформаційними об'єктами. В цьому випадку властивості сітки можуть суттєво впливати на інтерпретування відношень між інформаційними об'єктами, дозволяючи по-новому підійти до дослідження цих об'єктів і взаємодій між ними.

Саме цей підхід дозволяє виокремити типи нейронних сіток, які ефективно використовуються для реалізації різних задач АП.

6.2. Асоціативна пам'ять і програмування

Під АП зазвичай розуміють деякий набір або сукупність елементів, володіючих здатністю зберігати інформацію. Доступ до цих елементів здійснюється одночасно і паралельно у відповідності зі змістом збережуваних в них даних, а не шляхом завдання адреси або розташування елемента.

Проте таке розуміння АП відбиває, по суті, лише факт наявності взаємозв'язків між даними і не має відношення до самого механізму збереження інформації. Тому для позначення такого механізму збереження інформації використовується термін “*пам'ять з адресацією за змістом*” (ПАЗ).

Як тільки акцент був зроблений на будову ПАЗ, *стало можливим спростити вимоги до самого розуміння асоціативності* і розробити пристрої, лише у деякому сенсі володіючи асоціативністю.

Так, наприклад, *перше, що було спрощено* – це припущення, що паралелізм при виконанні операцій пошуку по суті не є принциповою функціональною характеристикою.

Друге припущення пов'язане із запереченням необхідності розподіленої пам'яті, бо асоціативності в сенсі пам'яті можна досягти і без необхідності розподіляти між елементами пам'яті інформацію. На противагу цьому можна зберігати одиницю інформації цілісно у визначеній комірці, маючи лише інформацію щодо безпосередніх зв'язків даної комірки з іншими – таким чином, ми приходимо до розуміння семантичних сіток. Дані принципи також використовуються при індексуванні та пошуку у сучасних базах даних. Ясно, що в даному сенсі це припущення протирічить ідеям конективізму (на яких, власне, базуються ШНС) і плавно перетікає до ідей символізму.

Головне, що втрачається при такому припущенні – це одна з дивовижних властивостей біологічної пам'яті. Як відомо, різного роду пошкодження тканини мозку приводять до порушень функціональних характеристик пам'яті. Проте, виявилось вкрай важко виокремити в роботі окремих нейронних структур явища, пов'язані з локалізацією функцій пам'яті. Пояснення цьому феномену базується на припущенні, що у мозку сліди пам'яті представлені у вигляді просторово-розподілених структур, які формуються в результаті деякого перетворення первісних сприйняття.

І хоча при такому припущенні був загублений ряд біологічно правдоподібних властивостей, що є важливим при моделюванні мозку, але в технічному сенсі стало зрозумілим як реалізувати пам'ять, адресовану за змістом. Дякуючи цьому з'явилися ідеї щодо хешування, які

потім були реалізовані як у мовах програмування, так і при апаратній реалізації деяких процесорів.

Третє припущення пов'язане з точністю співпадання шуканої інформації. Вибірка даних на основі їх змісту завжди припускає ту чи іншу форму порівняння заданого ззовні ключа, за яким повинен здійснюватися пошук, з деякою частиною або з усією інформацією, збереженою у комірках пам'яті. Метою порівняння не завжди має бути поява інформації, співпадаючої з ключовою. Наприклад, при відшукуванні значень, які розташовані всередині заданого інтервалу. У цьому випадку ми маємо класичний спосіб використання SQL при відборі з бази даних.

Проте можливий варіант пошуку, при якому необхідно серед сукупності даних знайти ті з них, які найкращим чином (в сенсі деякої заданої міри) відповідають ключовій інформації. За такої постановки задача асоціативної вибірки дуже близька до задачі розпізнавання образів. Проте визначальним є методи, які при цьому використовуються: якщо смисл асоціативності не підлягає описанню тут припущенням, то ми маємо справу з розпізнаванням образів за допомогою ШНС, у протилежному випадку – з оптимізацією роботи баз даних (а також апаратних кешів процесорів) або способами асоціативного представлення даних (наприклад, семантичних сіток).

Четверте припущення можна пов'язати з так званою проблемою часових асоціацій, що з точки зору програмування відноситься до теорії автоматів. Ці проблеми пов'язані з розробкою методів запам'ятовування та витягання з пам'яті упорядкованих у часі послідовностей. При цьому вони можуть розгалужуватися, утворюючи вторинні альтернативні послідовності, причому перехід до однієї з них визначається змістом деякої фонові або контекстної інформації. Вказані послідовності можуть також містити замкнені цикли.

Таким чином, з точки зору програмування або символізму, по відношенню до асоціативної пам'яті є всі ті ж самі проблеми і задачі, що й у штучному інтелекті. Відмінність пролягає хіба що в тому, що у програмуванні можуть бути зроблені припущення і побудовані методи, які лише частково задовольняють розумінню асоціативної пам'яті. В той же час конективізм прагне розв'язати проблему асоціативної пам'яті, використовуючи методи, які не містять припущень в наведених вище смислах, володіють деякою стохастичністю і непередбачуваністю в сенсі роботи методу, але у кінцевому рахунку дають осмислений результат в областях розпізнавання образів та адаптаційного управління.

6.3. Моделі асоціативної пам'яті

Асоціативна нейросітка прямого поширення. Процеси, аналогічні біологічним механізмам запам'ятовування та обробки інформації, можна представити за допомогою різних моделей АП, дозволяючих відобразити відношення (асоціації) довільної складності між інформаційними об'єктами. За великим рахунком у загальному випадку можна умовно стверджувати, що будь-яка ШНС так чи інакше пов'язана з розв'язанням задачі розпізнавання образів, як і всі вони певною мірою є асоціативними – відмінність полягає лише в ступені асоціативності: повній по всіх властивостях образу або частковій по окремих фрагментах останнього.

Означення 6.2. Асоціативна пам'ять може бути визначена як система для запису, зберігання, пошуку, обробки та зчитування інформації, в якій дані (знання) про об'єкт можуть ініціалізуватися за заданим в якості пошукового *фрагментом цих даних* (знань).

Виходячи з цього означення, можна сформулювати задачі, що розв'язуються АП:

- співставлення пошукової інформації з тією, що зберігається, та доповнення її (ініціалізація) до точного опису об'єкта, тобто всієї інформації, яка є доступною для АП;
- фільтрування (корегування) пошукової інформації відносно всього об'єму інформації, яка зберігається в АП, вирішення недостовірної і на основі тієї, що залишилася, розв'язання першої задачі.

Усі відношення можна реалізувати у вигляді простих конструкцій з трьох компонентів: упорядкованої пари інформаційних об'єктів O та V , а також типу відношення $A: O \xleftarrow{A} V$. Одна з найпростіших моделей АП для відображення таких відношень показана на рис. 6.1.

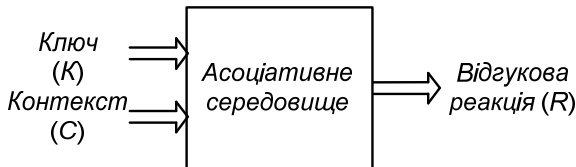


Рис. 6.1. Модель асоціативної пам'яті

Модель складається з асоціативного логіко-запам'ятовуючого середовища (нейронної сітки), зв'язаної з двома каналами введення та одним каналом виведення інформації.

На *етапі записування* (навчання) з першого каналу введення на вхід K подається вхідна інформація, а по другому каналу – ознако-

ва інформація C , яка представляє контекст, в якому вхідна інформація записується у пам'ять.

На *етапі функціонування* (зчитування) при появі ключа K (або його фрагмента) на виході пам'яті формується реакція-відповідь R , пов'язана з ключем K . Таким чином, занесену у пам'ять інформацію можна отримати із застосуванням будь-яких її фрагментів, які використовуються як пошукові. Задаючи різний контекст C , можна точніше конкретизувати інформацію, яку необхідно отримати.

Розглянемо на прикладі, як повинен бути організований процес накопичення та пошуку інформації в АП, щоб забезпечити циклічність процесу, за яким вибраний елемент інформації стає ключем для пошуку нової інформації

По трьох вхідних каналах одночасно можуть вводитися набори значень. По першому каналу в момент t подається адресна інформація $K(t)$, а по другому – ознака $C(t)$. Відгук $R(t)$ по каналу зворотного зв'язку подається також на вхід асоціативного середовища. Канал на виході слугує для видавання інформації.

При функціонуванні такої АП ключі $K(t)$ й ознаки $C(t)$ подаються через інтервали часу, які відповідають затриманню каналу зворотного зв'язку (рис. 6.2).



Рис. 6.2. Модель асоціативної пам'яті з оберненим зв'язком

Процес роботи пам'яті розглядатимемо у припущенні, що трійка $[K(t), C(t), R(t-\Delta)]$ уявляє собою єдиний статичний образ, заданий на момент часу t , причому можливий його одночасний запис у пам'ять за одну операцію. Припустимо також, що на етапі записування $R(t)$ і $K(t)$ є однаковими.

На *етапі запису* на входи АП надходять $K(t)$ і $C(t)$, при цьому на виході формується $R(t)$, ідентичний до $K(t)$. Після цього із затримкою Δ на вході формується $R(t-\Delta)$. Кожна нова трійка, яка з'являється на входах, записується у пам'ять.

На етапі здобуття даних з АП на вхід подається ключ K , пов'язаний з контекстною інформацією C , після чого ключ K можна зняти зі входу. В результаті на виході в якості відгуку з'явиться копія K . Коли на вході пам'яті з'явиться затриманий сигнал $R(t - \Delta)$, новою ключовою ознакою стає пара (C, R) , яка приводить до появи на виході наступного образу $R(t)$ і т. д. Таким чином, вибирається уся записана послідовність образів разом з контекстною інформацією.

Розглянута системна модель реалізує АП, придатну для запису та вибірки структурованих знань.

Прості лінійні моделі містять, як і персептрон, один або два шари, проте, на відміну від нього, використовує іншу парадигму навчання. Лінійна модель у загальному випадку використовується не для класифікації образів, а для їх асоціативного пошуку і відтворення.

Найпростішою асоціативною сіткою (АС) є лінійний асоціатор, який уявляє собою одношарову сітку и використовує тільки операцію скалярного добутку векторів

$$y_j = \sum_{i=1}^L W_{ij} x_i = w_j x, \quad j = \overline{1, Z}, \quad (6.1)$$

де $w_j - j$ -й рядок матриці ваг W .

Якщо у випадку персептрона (див. розд. 5) введення додаткових шарів приводить до придбання сіткою нових позитивних властивостей, це не відноситься до лінійних асоціаторів. Дійсно, якщо одношаровий лінійний асоціатор описується лінійним рівнянням з ваговою матрицею W_1 , тобто $y = W_1 x$, то додавання другого шару з ваговою матрицею W_2 продукує $v = W_2 y = W_1 W_2 x = Ux$, де $U = W_1 W_2$, і знов отримуємо лінійне перетворення. То ж багатошаровий лінійний асоціатор еквівалентний одношаровому.

Відмінною рисою лінійної моделі є її спроможність відтворювати частково зруйнований образ. Уявимо найпростішу АС прямого поширення у вигляді НС, наведеної на рис. 6.3, яка перетворює L вхідних сигналів у Z на виході. Нехай навчальну вибірку складають асоціативні пари $\{(x_p, y_p), p = \overline{1, P}\}$, кожна з яких характеризується своєю ваговою матрицею W_p розмірності $L \times Z$ і запам'ятовуються; задачею НС є видавання на виході сигналу (образу) y_p , асоційованого з x_p , якщо на її вхід надходить сигнал (образ) x , подібний до x_p , тобто віддалений від x_p на мінімальній віддалі Хеммінга.

Означення 6.3. Віддаллю Хеммінга між двома векторами, представленими у двійковій формі, називається кількість позицій (бітів), у яких вони відмінні:

$$d(x, x_p) = \min_k (x, x_k). \quad (6.2)$$

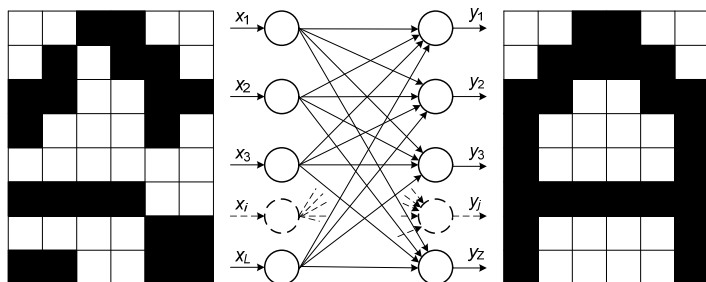


Рис. 6.3. Проста асоціативна НС прямого поширення відновлює спотворений “образ” літери “А”

Як видно з рис. 6.3, наведений приклад відновлення спотвореної літери “А”, яка подається у бінарному виді на вхід моделі. При цьому лінійна НС відводить для кожного образу не деяке визначене місце, а розподіляє інформацію щодо усіх збережуваних образах у вагових коефіцієнтах сітки. Тому при надходженні інформації на вхід такої сітки відбувається не порівняння пред’явлюваного образу із збережуваним, а лінійне перетворення вхідного сигналу, яке використовується для обчислення сигналу на виході. Саме так функціонуюча модель дістала назву *асоціативної пам’яті*.

Ємність АП обмежена її фізичними розмірами і видом використовуваного (лінійного) перетворення.

Моделі АП, які реалізуються НС, можуть бути й складнішими. Крім того, пам’ять може мати декілька входів і виходів, складатися з декількох підсистем. Дані в одному каналі можуть породжувати контекстну інформацію для іншого каналу. Етап запису може виконуватися окремо від вибірки або бути суміщеним тощо.

6.4. Алгоритми навчання асоціативної пам’яті

Асоціації або пам’ять сітки реалізується у ваговій матриці W , через що задача навчання сітки полягає у визначенні таких значень елементів матриці W , які забезпечуватимуть точне відтворення викликаного образу. Для цілей навчання використовуються правило Хебба, дельта-правило і метод найменших квадратів (або псевдоінверсії). Розглянемо особливості їх застосування при навчанні АП детальніше.

Використання правила навчання Хебба. Як відзначалося в розділі 4.3, гіпотеза Хебба формально задається виразом (4.2) у наступному виді [48]:

$$W_{ij} = \sigma x_i y_j; \quad i = \overline{1, N}; \quad j = \overline{1, Z} \quad (6.3)$$

або у матричній формі

$$W = \sigma \mathbf{y} \mathbf{x}^T, \quad (6.4)$$

де σ – коефіцієнт навчання.

Оскільки співвідношеннями (6.3) і (6.4) подається будь-яка з P навчаючих пар (x_p, y_p) , можна це представити так:

$$W_p = \sigma \mathbf{y}_p \mathbf{x}_p^T, \quad p = \overline{1, P}. \quad (6.5)$$

Тоді матриця ваг W , яка відбиває збереження у пам'яті усіх P образів, матиме вигляд:

$$W = \sum_{p=1}^P W_p. \quad (6.6)$$

Оскільки АП заснована на запам'ятовуванні об'єктів, одночасно спостережуваних, співвідношення (6.3) і (6.4) припускають і одночасне з'явлення \mathbf{x} та

У співвідношенні (6.3) синаптична вага W_{ij} між входом x_i та виходом y_j відбиває той факт, що якщо додатний/від'ємний x_i викликає додатний/від'ємний (відповідно) y_j то синаптичний зв'язок підсилюється згідно до наступного співвідношення:

$$W(k+1) = W(k) + \sigma \mathbf{y}(k) \mathbf{x}^T(k), \quad (6.7)$$

де індекс k відповідає k -м векторам на вході і на виході.

Правило (6.7) є неконтрольованим правилом навчання Хебба.

Означення 6.3. Неконтрольоване правило навчання Хебба – це правило, яке містить фактичні значення входів і виходів НС і не використовує ніякої інформації щодо вимаганої мети.

Згідно правила (6.7) змінення ваги пропорційне виникненню активності на будь-якому боці синапсу, тобто вага збільшується не тільки, коли \mathbf{x} та \mathbf{y} додатні, але й коли обидві змінні від'ємні. Іншими словами, дане правило засновано на кореляції між вхідними сигналами і сигналами на виході НС.

Якщо переписати правило (6.7) у виді

$$W(k) = W(0) + \sigma \sum_{i=1}^k \mathbf{y}(i) \mathbf{x}^T(i), \quad (6.8)$$

то легко побачити, що при тривалому пред'явленні входів, тобто із зростанням k , значення ваг можуть стати скільки завгодно великими.

Проте це не узгоджується з біологічними принципами, бо синаптичні ваги не можуть зростати до нескінченості. Цей факт є *першим недоліком* неконтрольованим правилом навчання Хебба.

Другим недоліком неконтрольованим правилом навчання Хебба, пов'язаним з першим, є відсутність у (6.7) механізму зменшення ваг, що приводить знов-таки до зростання ваг НС та її невірної реакції при впливі шумів на входи і виходи НС.

Для усунення наведених недоліків в алгоритмі (6.7) використовують механізм “незапам'ятовування” у вигляді:

$$\begin{aligned} W(k) &= W(k-1) + \sigma y(k) \mathbf{x}^T(k) - \nu W(k-1) = \\ &= (1-\nu) W(k-1) + \sigma y(k) \mathbf{x}^T(k), \end{aligned} \quad (6.9)$$

де $\nu \in (0, 1)$ – коефіцієнт незапам'ятовування.

Якщо $\nu \rightarrow 0$, алгоритм (6.9) відповідає (6.7), а при $\nu \rightarrow 1$ алгоритм (6.9) забуває стару інформацію і пам'ятає тільки останню, усуваючи тим самим нескінчене зростання вагових коефіцієнтів. Максимальне значення W_{ij}^{\max} визначається величинами ν та σ при $x_i = y_j = 1$, що має місце при максимізації швидкості навчання:

$$W_{ij}^{\max} = \frac{\sigma}{\nu}. \quad (6.10)$$

Оскільки за відсутності нових навчаючих сигналів зберезувані сіткою асоціації втрачаються, використання правила навчання з незапам'ятовуванням вимагає повторення пред'явлення сітки навчаючих послідовностей. Для запобігання такого явища слід користуватися наступним співвідношенням [48]:

$$W_{ij}(k) = W_{ij}(k-1) + \nu y_i(k) (x_j(k) - W_{ij}(k-1)), \quad (6.11)$$

яке отримало назву *правила вхідного зв'язку* [186].

Використання дельта-правила. Дане правило, називане також *правилом Відроу-Хоффа*, реалізує градієнтну процедуру мінімізації квадратичного функціоналу від помилки по всіх пред'явлених мережі образах. На відміну від неконтрольованого правила Хебба, дане правило реалізує *контрольоване навчання* через те, що використовує інформацію про бажаний сигнал на виході сітки y^{δ} .

Якщо для r -ої навчальної пари (x_p, y_p^{δ}) , $p = \overline{1, P}$ сигнал y_j на виході представлений співвідношенням (6.3), квадратичний критерій навчання має вид:

$$Q = \sum_{p=1}^P \sum_{j=1}^Z \left(y_{pj}^6 - \sum_{n=1}^N W_{ij} x_{pi} \right)^2, \quad (6.12)$$

і вагові параметри корегуються за правилом:

$$\Delta W_{ij} = -\sigma \frac{\partial Q}{\partial W_{ij}}, \quad (6.13)$$

яке з урахуванням (6.11) може бути перетворене у такий спосіб:

$$\Delta W_{ij} = \sigma \left(y_{pi}^6 - y_{pi} \right) x_{pi}, \quad (6.14)$$

чи в матричній формі

$$\Delta W_{ij} = \sigma \sum_{p=1}^P \left(y_p^6 - W x_p \right) x_p^T, \quad (6.15)$$

Перевага дельта-правила полягає в тому, що воно дозволяє оновлювати елементи вагової матриці після кожного пред'явлення вхідного образу, тобто забезпечує адаптацію нейросітки до умов, що змінюються.

Розпізнавання спотвореного образу. Якщо на вхід асоціативної НС подається зашумлений образ $\mathbf{x}_p^* = \mathbf{x}_p + \Delta \mathbf{x}_p$, то на її виході з'являється сигнал

$$\mathbf{y}_p^* = \mathbf{y}_p + \Delta \mathbf{y}_p, \quad (6.16)$$

де $\Delta \mathbf{y}_p$ – завадова складова, яка може бути подана як лінійна комбінація векторів, утворюючих ортонормальний базис

$$\Delta \mathbf{y}_p = \sum_i \xi_i \mathbf{x}_i. \quad (6.17)$$

У цьому випадку сигнал на виході НС обчислюється наступним чином:

$$\begin{aligned} \mathbf{W} \mathbf{x}_p^* &= \mathbf{W} \left(\mathbf{x}_p + \Delta \mathbf{y}_p \right) = \mathbf{y}_p + \mathbf{W} \Delta \mathbf{y}_p = \\ &= \mathbf{y}_p + \sum_i \sum_j \xi_i \mathbf{y}_i^T \mathbf{x}_j = \mathbf{y}_p + \sum_i \xi_i \mathbf{y}_i. \end{aligned} \quad (6.18)$$

Отже, сигнал на виході крім відповідного вірного (незбуреного) образу \mathbf{y}_p буде містити деякі частини інших збережуваних у сітці неспотворених образів. Вплив завади зростає, якщо збережувані образи не ортогональні.

Кожний вхідний образ \mathbf{x} відповідає у найбільшій мірі тому збережуваному образу \mathbf{x}_p , значення скалярного добутку (кореляція) з яким буде максимальним. Отже сигнал, відмінний від нуля, з'являється на виході i -го нейрона, номер якого визначається умовою:

$$i = \arg \max_k (\mathbf{x}_k^T \mathbf{x}). \quad (6.19)$$

6.5. Гетероасоціативна пам'ять

Гетероасоціативна пам'ять (ГАП) уявляє собою НС прямого поширення, що зберігає пари $\{\mathbf{x}_p, \mathbf{y}_p\}$, причому у загальному випадку $\mathbf{y}_p \neq \mathbf{x}_p$. Асоційовані пари \mathbf{x}_p та \mathbf{y}_p є векторами розмірності $N \times 1$ та $Z \times 1$, відповідно (рис. 6.3), причому $N \neq Z$. Схеми, що реалізують ГАП, наведені на рис. 6. 4.

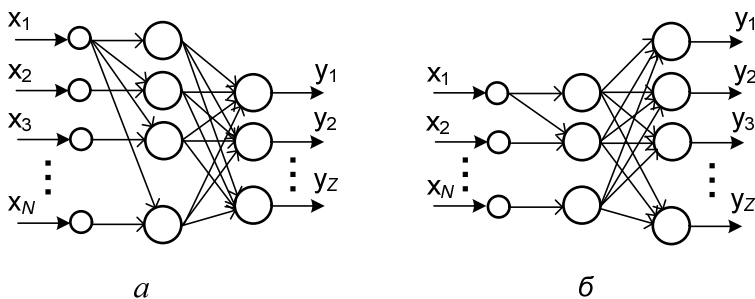


Рис. 6.4. Гетероасоціативна пам'ять ($N \neq Z$):

$a - N > Z$; $b - N < Z$

Нейросітку, зображену на рис. 6.4, *a*, застосовують при розв'язанні задач розпізнавання і прогнозування, коли необхідно визначити, до якого класу відноситься пропонований образ, а НС на рис. 6.4, *б* – при розв'язанні задач відновлення, коли за номером класу (деякому коду) необхідно відновити образ.

Для навчання мережі використовується будь-який з розглянутих вище алгоритмів.

Приклад 6.1 [48]. Розглянемо ГАП, наведену на рис. 6.4, *a* з $N = 5$, $Z = 3$ і пороговою (релейною) функцією активації, на яку надходять навчальні пари, представлені біполярними сигналами.

При використанні порогової функції активації компоненти вектора сигналу на виході НС обчислюються у такий спосіб:

$$y_j = \text{sign} \left(\sum_{i=1}^N w_{ij} x_i \right) = \begin{cases} 1 & \text{при } \sum_{i=1}^N w_{ij} x_i \geq 0; \\ -1 & \text{при } \sum_{i=1}^N w_{ij} x_i < 0. \end{cases} \quad (6.20)$$

Додамо, що вхідні образи не є ані ортогональними, ані нормалізова-

ними. Незважаючи на це, вагова матриця, що обчислювана за правилом Хебба, дає вірний результат. Обчислимо відповідно з (6.5) частинні вагові матриці, прийнявши $\sigma = 1$:

$$\mathbf{W}_1 = \mathbf{y}_1 \mathbf{x}_1^T = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} (1 \ -1 \ -1 \ 1 \ 1) = \begin{pmatrix} 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & -1 \end{pmatrix}; \quad (6.21)$$

$$\mathbf{W}_2 = \mathbf{y}_2 \mathbf{x}_2^T = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} (1 \ 1 \ 1 \ -1 \ -1) = \begin{pmatrix} 1 & 1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 \end{pmatrix}; \quad (6.22)$$

$$\mathbf{W}_3 = \mathbf{y}_3 \mathbf{x}_3^T = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} (-1 \ 1 \ 1 \ 1 \ 1) = \begin{pmatrix} 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 \end{pmatrix}. \quad (6.23)$$

Скориставшись (6. 6), одержуємо:

$$\mathbf{W} = \sum_{i=1}^3 \mathbf{W}_i = \begin{pmatrix} 3 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 3 & 3 \\ -1 & 3 & 3 & -1 & -1 \end{pmatrix}. \quad (6.24)$$

Для відшукування асоційованих образів при використанні порогової активаційної функції з урахуванням (6.20) і (6.24) обчислимо

$$\mathbf{W} \mathbf{x}_1 = \begin{pmatrix} 3 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 3 & 3 \\ -1 & 3 & 3 & -1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 7 \\ -9 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} = \mathbf{y}_1. \quad (6.25)$$

$$\mathbf{W} \mathbf{x}_2 = \begin{pmatrix} 3 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 3 & 3 \\ -1 & 3 & 3 & -1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 3 \\ -9 \\ 7 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \mathbf{y}_2. \quad (6.26)$$

$$\mathbf{W} \mathbf{x}_3 = \begin{pmatrix} 3 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 3 & 3 \\ -1 & 3 & 3 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -7 \\ 5 \\ 5 \end{pmatrix} \Rightarrow \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} = \mathbf{y}_3. \quad (6.27)$$

Отже, ГАП навчилася, тобто вірно створила і відтворила асоціації $\mathbf{y}_1 = \mathbf{y}_1$; $\mathbf{y}_2 = \mathbf{y}_2$; $\mathbf{y}_3 = \mathbf{y}_3$.

Тепер розглянемо ситуацію, коли на вхід навченої ГАП подається спотворений образ. Нехай через якісь обставини спотворився образ \mathbf{x}_2 , тобто замість \mathbf{x}_2 на вхід надходить образ $\mathbf{x}_2^* = (1 \ 1 \ -1 \ -1 \ -1)$ (перекручений біт виділений у третьому розряді). У цьому випадку (6.26) перепишемо у вигляді:

$$\mathbf{W}\mathbf{x}_2^* = \begin{pmatrix} 3 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 3 & 3 \\ -1 & 3 & 3 & -1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 3 \\ -7 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \mathbf{y}_2. \quad (6.28)$$

Таким чином, нейросітка вірно відновила спотворений образ \mathbf{x}_2^* .

6.6. Автоасоціативна пам'ять

Автоасоціативна пам'ять (ААП) є частинним випадком ГАП при $N=Z$ (рис. 6.5).

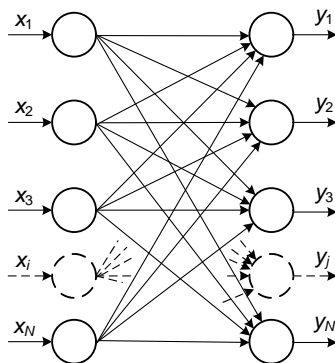


Рис. 6.5. Автоасоціативна пам'ять

Сітка зберігає пред'явлені навчальні пари (образи) (x, y) з $y = x$. Задача, розв'язувана цією пам'яттю, полягає у відновленні деякого зашумленого чи спотвореного образу $\mathbf{x}_p^* = \mathbf{x}_p + \Delta\mathbf{x}_p$, а не у створенні його асоціації з іншим образом. Це є результатом одношарової структури асоціативної пам'яті, у якій вектор сигналів на виході з'являється на виходах тих же нейронів, на які надходить вектор вхідних сигналів. Дана пам'ять має усі властивості ГАП і забезпечує точне відтворення перекрученого образу при нормалізованих (ортонормованих) вхідних сигналах. Ємність пам'яті (максимальне число збе-

режуваних образів), дорівнює N . На практиці ж звичайно використовується кількість ортогональних образів, що зберігаються, менша за N через те, що при N асоціаціях кожний стан НС є стійким і вона втрачає здатність відновлювати перекручені образи.

Як і у випадку з ГАП, навчання автоасоціативної пам'яті здійснюється за допомогою кожного з розглянутих вище методів.

Приклад 6.2 [48]. Розглянемо ААП, представлену на рис. 6.5 з $N=5$, що має порогову функцію активації. Нехай на вхід мережі подаються образи

$$\begin{aligned}\mathbf{x}_1^T &= (-1 \ -1 \ -1 \ -1 \ 1) \\ \mathbf{x}_2^T &= (1 \ -1 \ 1 \ -1 \ 1) \\ \mathbf{x}_3^T &= (-1 \ -1 \ 1 \ 1 \ -1) \\ \mathbf{x}_4^T &= (-1 \ -1 \ -1 \ 1 \ 1).\end{aligned}\tag{6.29}$$

Скориставшись тим, що вагові матриці для образів x_i ($i = \overline{1,4}$) визначаються за формулою $\mathbf{W}_i = \mathbf{x}_i \mathbf{x}_i^T$, одержуємо:

$$\begin{aligned}\mathbf{W}_1 = \mathbf{x}_1 \mathbf{x}_1^T &= \begin{pmatrix} 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix}; \mathbf{W}_2 = \mathbf{x}_2 \mathbf{x}_2^T = \begin{pmatrix} 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 \end{pmatrix}; \\ \mathbf{W}_3 = \mathbf{x}_3 \mathbf{x}_3^T &= \begin{pmatrix} 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 \end{pmatrix}; \mathbf{W}_4 = \mathbf{x}_4 \mathbf{x}_4^T = \begin{pmatrix} 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 \end{pmatrix}.\end{aligned}\tag{6.30}$$

Результуючу вагову матрицю \mathbf{W} нейросітки обчислюємо за залежністю (6.6):

$$\mathbf{W} = \sum_{i=1}^4 \mathbf{W}_i = \begin{pmatrix} 4 & 2 & 2 & -2 & 0 \\ 2 & 4 & 0 & 0 & -2 \\ 2 & 0 & 4 & 0 & -2 \\ -2 & 0 & 0 & 4 & -2 \\ 0 & -2 & -2 & -2 & 4 \end{pmatrix}.\tag{6.31}$$

Для визначення будь-якого образу, збережаного в пам'яті, обчислюємо добуток $\mathbf{W} \mathbf{x}_i$ ($i = \overline{1,4}$). Наприклад, при пред'явленні сітці образу \mathbf{x}_3 маємо:

$$\mathbf{W}\mathbf{x}_3 = \begin{pmatrix} 4 & 2 & 2 & -2 & 0 \\ 2 & 4 & 0 & 0 & -2 \\ 2 & 0 & 4 & 0 & -2 \\ -2 & 0 & 0 & 4 & -2 \\ 0 & -2 & -2 & -2 & 4 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -6 \\ -4 \\ 4 \\ 6 \\ -6 \end{pmatrix}; \quad (6.32)$$

$$\mathbf{y}_3^T = \text{sign}(\mathbf{W}\mathbf{x}_3)^T = (-1 -1 \ 1 \ 1 -1),$$

тобто сітка вірно відтворила збережений образ \mathbf{x}_3 . Аналогічні результати виходять і для інших образів. Нехай на вхід НС поданий перевернутий у третьому біті образ $\mathbf{x}_3^* = (-1 -1 -1 \ 1 -1)$ (спотворений біт виділений у третьому розряді).

Застосовуючи матрицю \mathbf{W} (6.31) до \mathbf{x}_3^* , одержуємо:

$$\mathbf{W}\mathbf{x}_3^* = \begin{pmatrix} 4 & 2 & 2 & -2 & 0 \\ 2 & 4 & 0 & 0 & -2 \\ 2 & 0 & 4 & 0 & -2 \\ -2 & 0 & 0 & 4 & -2 \\ 0 & -2 & -2 & -2 & 4 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -10 \\ -4 \\ 4 \\ 8 \\ -6 \end{pmatrix}; \quad (6.33)$$

$$\mathbf{y}_3^* = \text{sign}(\mathbf{W}\mathbf{x}_3^*) = (-1 -1 \ 1 \ 1 -1).$$

Отже, як впливає з співставлення (6.32) та (6.33), ААП правильно відновила спотворений образ.

6.7. Модель нейронної сітки Хопфілда з автоасоціативною пам'яттю

Загальні відомості. На відміну від розглянутих прямоспрямованих сіток існують НС, в яких сигнал на виході знову спрямовується на вхід сітки, через що виникає ітеративний процес – це так звані *сітки з оберненим зв'язком* (*Feed-back Nets*), або автоасоціативні НС, згадувані вище. До таких структур належать і *сітки Хопфілда* [143], які викликали новий спалах досліджень в області ШНС. Американський фізик Джон Хопфілд вперше представив свою асоціативну сітку у 1982 р. у Національній Академії Наук, і на честь запропонованого ним нового підходу до моделювання ця сіткова парадигма згадується як сітка Хопфілда. Ідеї, засновані на результатах У. Мак-каллоха і В. Піттса, Т. Кохонена і С. Гроссберга та інших науковців, Хопфілд використав для моделювання спінних слідів при дослідженні матеріалів, атоми яких володіють магнітним диполем. При моделюванні кожному диполу відповідав нейрон, орієнтація у магнітному полі відбувалася

збудженням відповідного нейрона, а сітка відбивала магнітні взаємодії поле й. Описувані в його публікаціях застосування містять *асоціативну пам'ять, аналого-цифрове перетворення, розв'язання задачі оптимізації* (про комівояжера).

Сітки Хопфілда містять лише один шар нейронів, причому кожний нейрон пов'язаний з усіма іншими і сам із собою (є повнозв'язними сітками), в результаті чого мають місце численні зворотні зв'язки. До всіх нейронів підводяться вхідні сигнали, які можуть приймати значення $+1$ або -1 . Проте, результати функціонування такої сітки залишаються справедливими і для версії ± 1 . Оскільки, крім цього, всі ваги є цілочисельними, подальші розрахунки легко виконувати. Основною областю застосування сіток Хопфілда є розпізнавання образів. Процес розпізнавання полягає в тому, що якщо задано сітку Хопфілда з вагами і пороговими значеннями, то вона перебуватиме в динамічній рівновазі за умови, якщо на її вхід поданий образ, заданий певним стартовим вектором в процесі їх генерації. Ці стани рівноваги характеризуються тим, що загальна енергія (Гамільтонова функція) системи є мінімальною. Перенесення цього положення на сітку Хопфілда приводить до “енергетичної функції”, яка буде мінімальною при наданні сітці деякого образу. Оскільки при цьому запам'ятовуються всі двійкові величини, для яких енергетична функція мінімальна, то зафіксовані мінімуми є локальними.

Проте, деякі застосування сіток потребують пошуку глобального мінімуму, тобто абсолютного мінімуму енергетичної функції. Такі застосування уявляють собою базу Больтцман-машини, яку обробляють так звані сітки Больтцмана [80].

До інших застосувань Хопфілд-ітерацій відноситься задачі оптимізації, в яких здійснюється ідентифікація енергетичної функції з деяким виразом. Цю процедуру можна розв'язати за допомогою сітки Хопфілда.

Нейронна сітка Хопфілда реалізує суттєву властивість *автоасоціативної пам'яті* (ААП) – відновлення по спотвореному (зашумленому) образу найближчого до нього еталонного. В цьому випадку вхідний вектор використовується як початковий стан сітки, і далі сітка еволюціонує відповідно до своєї динаміки. Причому будь-який приклад, що знаходиться в області тяжіння зразка, який зберігається, може бути використаний як вказівка для його поновлення. Вихідний (поновлюваний) зразок встановлюється, коли сітка досягає рівноваги.

Якщо сітка Хопфілда використовується як пам'ять, що адресується за змістом, вона має два головних обмеження: *по-перше*, кількість збережуваних та точно відтворюваних образів строго обмежена – якщо зберігається занадто багато параметрів, сітка може або збігатися до нового неіснуючого образу, відмінному від усіх запрограмованих, або не збігатися взагалі. Межа ємкості пам'яті для сітки приблизно 15% від кількості нейронів у шарі Хопфілда. *Другим обмеженням парадигми* є те, що шар Хопфілда може стати нестабільним, якщо навчальні приклади є занадто подібними.

Твердження 6.5. Зразок образу вважається нестабільним, якщо він застосовується за нульовий час і сітка збігається до деякого іншого образу з навчальної множини. Ця проблема може бути вирішена вибором більш ортогональних між собою навчальних прикладів.

Особливості побудування сітки. Структура сітки Хопфілда наведена на рис. 6.6. Вона складається з одного шару нейронів, кількість яких визначає кількість входів та виходів сітки. Вихід кожного нейрона з'єднаний із входами усіх інших нейронів. Подавання входних векторів здійснюється через окремі входи нейронів.

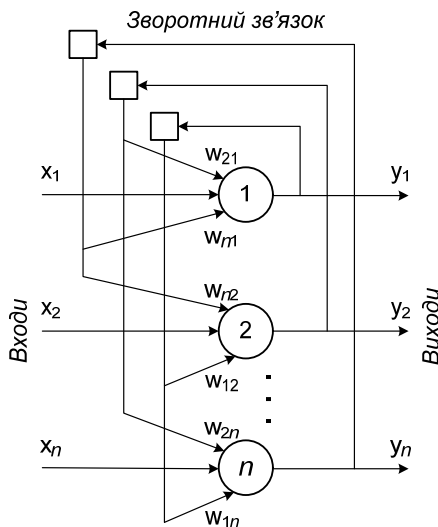


Рис. 6.6. Структура нейронної сітки Хопфілда

Сітки Хопфілда відрізняються від інших типів нейронних сіток наступними суттєвими ознаками:

- наявністю зворотних зв'язків з виходів сіток на їх входи за принципом «з усіх на всі»;

- розрахунок вагових коефіцієнтів нейронів виконується на основі вихідної інформації лише перед початком функціонування сітки, і все навчання сітки зводиться саме до цього розрахунку без урахування навчаючих ітерацій;
- при пред'явленні вхідного вектора сітка «збігається» до одного з запам'ятованих в ній еталонів, які уявляють собою множину рівноважних точок – локальних мінімумів функції енергії, що містить у собі всю структуру взаємозв'язків у сітці.

Проблема стійкості сітки Хопфілда була розв'язана після того, як Кохонен і Гроссберг довели теорему, яка визначає *необхідну і достатню умови стійкості сіток з оберненими зв'язками*:

Твердження 6.6. *Сітка з оберненими зв'язками є стійкою, якщо матриця її ваг симетрична ($w_{ij} = w_{ji}$) (необхідна умова) і має нулі на головній діагоналі ($w_{ii} = 0$) (достатня умова).*

Твердження 6.7. *Для того, щоб сітка Хопфілда була працездатною, її ваги не повинні бути всі від'ємними.*

Динамічна зміна станів сітки може виконуватися, принаймні, двома способами: *синхронно* та *асинхронно*. У першому випадку усі елементи модифікуються одночасно на кожному часовому кроці, у другому – у кожний момент часу вибирається та підлягає обробці один елемент. Цей елемент може вибиратися випадково.

Синхронний режим роботи сітки. Якщо робота сітки модулюється на одному процесорі, то при синхронному режимі послідовно переглядаються нейрони, проте їх стани запам'ятовуються окремо і не змінюються доти, доки не будуть пройдені всі нейрони сітки. Коли всі нейрони переглянуті, їх стани одночасно (тобто синхронно, звідси й назва) змінюються на нові. Таким чином, досягається моделювання паралельної роботи послідовним алгоритмом.

При реально паралельному моделюванні цей режим фактично означає, що час передачі τ_{ij} для кожного зв'язку між елементами u_i та u_j однаковий, що приводить до паралельної роботи всіх зв'язків, вони одночасно змінюють свої стани, базуючись тільки на попередньому моменті часу. Наявність таких синхронних тактів, які можна легко виокремити, і приводить до розуміння синхронного режиму. При синхронному режимі можливо (хоча далеко і на завжди спостерігається) нескінченне чергування двох станів з різною енергією – так званий динамічний атрактор. Тому синхронний режим практично для сітки Хопфілда не використовується і розглядається далі лише як основа для розуміння більш складного асинхронного режиму.

Асинхронний режим роботи сітки. Якщо моделювати роботу сітки як послідовний алгоритм, то в асинхронному режимі роботи стани нейронів в наступний момент часу змінюються послідовно: обчислюється локальне поле для першого нейрона в момент t , визначається його реакція, і нейрон встановлюється в новий стан (який відповідає його виходу в момент $t+1$), далі обчислюється локальне поле для другого нейрона з урахуванням нового стану першого, змінюється стан другого нейрона, і так далі – стан кожного наступного нейрона обчислюється з урахуванням всіх змінень станів розглянутих раніше нейронів.

По суті при послідовній реалізації сітки Хопфілда явно не видно, в чому полягає асинхронність, але це стає помітним, якщо сітка Хопфілда реалізується з паралельними обчисленнями. В цьому випадку асинхронний режим сітки Хопфілда спрощений і носить частковий випадок у порівнянні із загальним видом асинхронних сіток, де час передання τ_{ij} для кожного зв'язку між елементами u_i та u_j свій, але постійний. Щоб розглянути роботу сітки при паралельній реалізації, необхідно ввести поняття такту – мінімальний час, за який відбувається передання сигналу за зв'язком, тобто при $\tau_{ij}=1$. Тоді за проміжок часу між t та $t+1$ відбувається визначена кількість тактів N . Саме в межах часу з N тактів відбувається асинхронність проходження сигналів і виконання розрахунків. Наприклад, коли необхідно розрахувати стан нейрона № 3, треба розрахувати стан нейрона № 1 і стан нейрона № 2 та перемножити це на відповідні ваги w_{13} і w_{23} . Проте, як виявляється, щоб розрахувати стан нейрона № 2, треба знати *оновлений* стан нейрона № 1 і *старий* стан нейрона № 3, і перемножити їх на ваги w_{13} і w_{23} . Зрозуміло, що фізично неможливо обчислити стан нейрона № 1 і стан нейрона № 2 за один і той же час через те, що стан нейрона № 2 залежить від стану нейрона № 1. Тому зв'язок між нейроном № 1 і нейроном № 3 має час передання $\tau_{ij}=2$ і досягає нейрона № 3 за два такти. Саме такий відмінний час передання τ_{ij} і дозволяє говорити про сітку Хопфілда як про сітку з асинхронним режимом.

В асинхронному режимі неможливий динамічний аттрактор: не залежно від кількості запам'ятованих образів і початкового стану сітка неодмінно прийде до усталеного стану (статичного аттрактора).

Розглянемо *синхронну бінарну сітку Хопфілда* як приклад сітки з дискретними станами і дискретним часом і сформулюємо розв'язувану нею задачу наступним чином.

Нехай сітка складається з нейронів із пороговою функцією активації, виходи яких приймають значення або «0», або «1» при перевищенні зваженої суми значень входів деякого порогового рівня. Початково у сітці матрицею вагових коефіцієнтів заданий набір еталонів. Кожний еталон при цьому є точкою з кінцевої множини рівноважних точок, характеризуючих мінімуми енергії сітки (функції Ляпунова):

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} y_i y_j - \sum_{j=1}^n x_j y_j + \sum_{j=1}^n \theta_j y_j, \quad (6.34)$$

де E – штучна енергія сітки; w_{ij} – вага від виходу i -го до входу j -го нейрона; x_j , y_j – вхід і вихід j -го нейрона; θ_j – поріг j -го нейрона.

Головна властивість енергетичної функції полягає в тому, що в процесі еволюції станів нейронної сітки відповідно до рівняння (6.33) вона зменшується і *досягає локального мінімуму (атрактора)*, в якому зберігає постійну енергію. Це дозволяє розв'язувати задачі комбінаторної оптимізації, якщо вони можуть бути сформульовані як задачі мінімізації енергії.

Позначимо вектор, який описує k -й еталон, через $\mathbf{X}^{\text{ет}} = \{x_k^{\text{ет}}\}$, $k=1, \dots, K$; K – кількість еталонів.

На вхід сітки подається довільний вектор $\mathbf{X} = \{x_i\}$.

В результаті серії ітерацій сітка повинна виокремити еталон, який відповідає вхідному вектору, або дати висновок відносно невідповідності даних жодному з еталонів.

Після окремої ітерації загальна зміна енергії сітки, викликана зміною станів усіх нейронів, складе:

$$\Delta E = - \sum_{j=1}^n \left[\sum_{i \neq j} (w_{ij} y_j) + x_j - \theta_j \right] \Delta y_j, \quad (6.35)$$

де Δy_j – зміна виходу j -го нейрона після ітерації.

Аналіз виразу (6.35) доводить, що будь-яке змінювання стану нейронів або зменшить значення E , або залишить його без зміни. Другий випадок указуватиме на досягнення сіткою сталого стану і виокремлення нею еталону, який найкращим чином узгоджується із вхідним вектором.

При розпізнаванні вхідного вектору (частково представленого або спотвореного) виходи сітки будуть містити відповідний еталон, тобто $\mathbf{Y} = \mathbf{X}^{\text{ет}}$, де $\mathbf{Y} = \{y_j\}$ – вектор на виході. Інакше вектор на виході не співпадає з жодним еталоном.

Твердження 6.8. Для безпомилкової роботи сітки Хопфілда кількість запам'ятовуваних еталонів не повинна перевищувати $0,15n$.

Крім того, у випадку високого ступеня кореляції декількох еталонів можливе виникнення перехресних асоціацій при їх пред'явленні на входах сітки. Вимогу достатньої (але не необхідної) умови слабкої корельованості образів можна представити як виконання наступної нерівності

$$\sum_{k \neq j}^N |(x^k, x^j)| < n, \quad j = 1, \dots, n \quad (6.36)$$

або у вигляді більш сильної умови

$$|(x^k, x^j)| < \frac{n}{N}, \quad k = 1, \dots, n, \quad k \neq j. \quad (6.37)$$

Нейронні сітки Хопфілда з *неперервними станами* відрізняються від вищерозглянутої сітки *неперервною активаційною функцією нейронів*, за яку найчастіше вибирають *сигмоїдну* або *логістичну функцію*. Концептуально питання організації та функціонування цих сіток схожі.

Алгоритм функціонування сітки. Алгоритм навчання сітки Хопфілда суттєво відрізняється від класичних алгоритмів навчання персептронів (методи корекції похибок або зворотного поширення похибки). Відмінність полягає у тому, що замість послідовного наближення до необхідного стану з обчисленням похибок, всі коефіцієнти матриці розраховуються по одній формулі, за один цикл, після чого сітка готова до роботи. Обчислення коефіцієнтів засноване на наступному правилі: для всіх запам'ятованих образів \mathbf{X}_i матриця зв'язків повинна задовольняти рівнянню

$$\mathbf{X}_i^T = \mathbf{W}\mathbf{X}_i, \quad (6.38)$$

оскільки саме за умовою (6.38) стани сітки \mathbf{X}_i будуть стійкими, і потрапивши у такий стан сітка в ньому і залишиться.

Деякі автори відносять сітку Хопфілда до навчання без вчителя. Проте це невірно, бо навчання без вчителя передбачає відсутність інформації про те, до яких класів треба відносити стимули. Але для сітки Хопфілда без цієї інформації неможливо налагодити вагові коефіцієнти, тому тут можна говорити тільки за те, що таку сітку можна віднести до класу оптимізуючих сіток (фільтрів). Відмінною особливістю фільтрів є те, що матриця вагових коефіцієнтів настроюється детермінованим алгоритмом раз і назавжди, а потім вагові коефіцієнти більш не змінюються. Це може бути зручним для фізичного втілення

такого пристрою, бо на схемотехнічному рівні реалізувати пристрій зі змінними ваговими коефіцієнтами на порядок складніше.

Сталість сітки в процесі роботи. В сітці Хопфілда є зворотні зв'язки, через що треба розв'язувати проблему стійкості. Є багато стійких систем, наприклад, всі сітки прямого поширення, а також сучасні рекурентні сітки Джордана та Елмана, для яких не обов'язково виконувати умову на симетричність вагової матриці. Але це відбувається через те, що на зворотні зв'язки накладаються інші обмеження. У випадку ж сіток Хопфілда умова симетричності є необхідною, але не є достатньою (див. *Твердження 6.6*).

Принципова різниця між обома режимами роботи сітки полягає в тому, що в асинхронному випадку сітка обов'язково прийде до одного усталеного стану. При синхронному ж можливі ситуації з нескінченням циклічним переходом між двома різними станами.

Визначити, чи усталеним є стан нейрона, можна на підставі так званої *штучної енергії* $E_i = s_i h_i$ нейрона в даному полі. Якщо знак виходу (+1 або -1) нейрона співпадає з напрямком локального поля ($E_i = 0$), то його положення енергетично стійке, і в наступний момент часу стан нейрона залишається незмінним. В протилежному разі ($E_i = 0$) положення нейрона нестале, і він змінює свій знак, переходячи в стан $s_i(t+1) = -s_i(t)$ з енергією $E_i(t+1) < E_i(t)$.

Усталеність при асинхронному способі досягається тому, що виконується умова на загальну енергію сітки $E(t+1) \leq E(t)$. В синхронному випадку умова трохи змінюється, а саме: $E(t+1) \leq E(t-1)$. В ситуації, коли відбуваються нескінчені циклічні переходи, енергія двох різних станів відповідно дорівнює $E(t)$ і $E(t+1)$. При цьому стани $t+1$ і $t-1$, а також t і $t+2$ співпадають. Якщо утворюється такий стан, то його називають *динамічним атрактором*. Якщо ж співпадають стани t і $t+1$ атрактор називають *статичним*. У більшості випадків динамічні атрактори є небажаними, бо не відповідають будь-якій визначеній відповіді сітки.

Недоліком класичного варіанта сіток Хопфілда є їх тенденція до *усталювання в локальних, а не глобальних мінімумах сітки*. Через це запропоновані *статистичні сітки Хопфілда*, у яких цей недолік долається за рахунок утворення статистичних, а не детерміністських правил змінювання станів нейронів. При цьому для кожного j -го нейрона вводиться імовірність змінювання його стану p_j як функції від величини, на яку вихід нейрона y_j перевищує його поріг θ_j (для бінарної сітки Хопфілда):

$$\rho_j = \frac{1}{1 + \exp\left(\frac{-\Delta E_j}{\mu}\right)}, \quad (6.39)$$

де $\Delta E_j = y_j - \theta_j$; μ – параметр, який змінюється в процесі стабілізації сітки.

Тоді після початкового завдання вагових коефіцієнтів нейронів процедура пошуку мінімуму енергії сітки для усталеного вхідного вектора виглядає наступним чином:

Крок 1. Задати велике значення параметра μ .

Крок 2. Установити на виході j -го нейрона значення «1» з імовірністю ρ_j та значення «0» з імовірністю $1 - \rho_j$.

Крок 3. Поступово зменшити значення параметра μ , доки не буде досягнуто рівновагу.

Навчання сітки. Запам'ятовувані вектори повинні мати бінарний вид. Після цього відбувається розрахунок вагових коефіцієнтів за формулою:

$$w_{ij} = \frac{1}{N} \sum_{d=1 \dots m} \mathbf{X}_{id} \mathbf{X}_{jd}, \quad (6.40)$$

де N – вимірність векторів; m – кількість запам'ятовуваних векторів на виході; d – номер запам'ятовуваного вектора на виході; \mathbf{X}_{ij} – i -та компонента запам'ятовуваного j -го вектора на виході.

Вираз (6.40) може стати зрозумілішим, якщо зауважити, що матриця \mathbf{W} може бути знайдена обчислюванням зовнішнього добутку кожного запам'ятовуваного вектора із самим собою і підсумовуванням матриць, отриманих таким чином. Це можна записати у вигляді:

$$\mathbf{W} = \frac{1}{N} \sum_i \mathbf{X}_i^T \mathbf{X}_i, \quad (6.41)$$

де \mathbf{X}_i – i -й запам'ятовуваний вектор-рядок.

Розрахунок цих вагових коефіцієнтів і називається *навчанням сітки*.

Як тільки ваги задано, сітка може використовуватися для отримання запам'ятованого вектора на виході по даному вхідному вектору, який може бути частково невірним або неповним. Для цього виходам сітки спочатку надають значення цього навчального вектора, а далі сітка послідовно змінює свої стани відповідно до формули:

$$\mathbf{X}_{i+1}^T = f(\mathbf{W} \mathbf{X}_i), \quad (6.42)$$

де f – активаційна функція; поточний та наступний стани сітки, поки

стани \mathbf{X}_i та \mathbf{X}_{i+1} не співпадуть (або, у випадку синхронного режиму роботи, не співпадуть стани \mathbf{X}_{i-1} з \mathbf{X}_{i+1} та одночасно \mathbf{X}_{i-2} з \mathbf{X}_i). Саме цей процес дістав назву *конвергенції сітки*. Отриманий стійкий стан \mathbf{X}_i (*статичний атрактор*), або, можливо, у синхронному випадку пара $\{\mathbf{X}_i, \mathbf{X}_{i+1}\}$ (*динамічний атрактор*), i є відповіддю сітки на даний вхідний образ.

Приклад 6.3. Розглянемо для ілюстрації наведених співвідношень сітку Хопфілда, що складається з трьох нейронів (рис. 6.7). Усі зв'язки мають ваги, рівні 1.

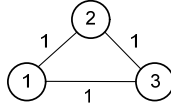


Рис. 6.7. Сітка Хопфілда з трьох нейронів

$$w_{12} = w_{21} = w_{13} = w_{31} = w_{23} = w_{32} = 1, \quad (6.43)$$

а

$$w_{11} = w_{22} = w_{33} = 0. \quad (6.44)$$

Вагова матриця має вигляд

$$w = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad (6.45)$$

Як активаційна функція прийнята сигмоїдна функція, а всі граничні величини дорівнюють 0. Для вхідного вектора

$$\bar{x} = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

розраховується вектор \bar{y} на виході:

$$\begin{aligned} \bar{y}^{(1)} &= \bar{f}(\bar{W}\bar{x}) = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}; \\ \bar{y}^{(2)} &= \bar{f}(\bar{W}\bar{y}^{(1)}) = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}; \\ \bar{y}^{(3)} &= \bar{f}(\bar{W}\bar{y}^{(2)}) = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}. \end{aligned}$$

Оскільки $\bar{y}^{(3)} = \bar{y}^{(2)}$, починаючи з третього кроку подальших змін не відбудеться. Вектор на виході \bar{y} знайдений у такий спосіб за дві ітерації і дорівнює

$$\bar{y} = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

Опрацювання візуальних образів (фільтрація та асоціативна пам'ять) – не єдина область застосування моделі Хопфілда. Динамічна процедура, описана вище, на кожному кроці знижує значення енергії НС. Це дозволяє розв'язувати комбінаторні задачі оптимізації, якщо вони можуть бути сформульовані як задачі мінімізації енергії. Класичною проблемою такого типу є задача комівояжера (комівояжер (*фр. commis voyageur, застар.*) – роз'їздний збутовий посередник), яка полягає у визначенні найвигіднішого маршруту, який проходить через N міст.

Приклад 6.4. Розв'язання задачі про комівояжера. В умовах задачі комівояжера указують критерій вигідності маршруту (найкоротший, найдешевший, сукупний тощо) і відповідні матриці віддалей і вартості переїзду між містами і т.п. Як правило, указують також, що маршрут повинен проходити через кожне місто тільки один раз – в такому випадку вибір здійснюється серед гамільтонових циклів. Сітка Хопфілда може використовуватись для розв'язання означеної задачі (треба обійти всі n міст і повернутися в початкове так, щоб довжина пройденого маршруту була мінімальною). Для цього висуваяться, наприклад, такі вимоги до сітки:

1. Сітка повинна складатися з $N = n \times n$ нейронів, які розглядатимуться як квадрат з n рядків та n стовпців.
2. Відповідь сітки повинна містити тільки по одному активному нейрону в кожному рядку і кожному стовпці.
3. Активний нейрон в першому стовпці задає перше місто маршруту, у другому стовпці – друге місто маршруту, і так далі.
4. При присвоєнні ваг повинна існувати гарантія, що НС надасть розв'язок, який відповідатиме коректному туру.

Стає зрозумілим, чому присвоєння ваг є найважливішим для досягнення задовольняючого розв'язку при використанні машини Больцмана (див. приклад 9.2, розд. 9) та сітки Хопфілда.

Джон Хопфілд та Девід Танк запропонували один з шляхів присвоєння ваг для машини Больцмана [145]. Основна ідея полягає у накладанні штрафів на некоректні тури і одночасно в мінімізації довжини туру.

Інший підхід Еміля Аартса [79] надає можливість формально встановлювати присвоєння, ув'язавши з поставленою задачею, тоді як багато інших підходів для присвоєння ваг є евристичними. Звичайно, необов'язково

припускається, що результуючий алгоритм працює добре, тому Аартс пропонує декілька модифікацій, призначених для покращення продуктивності в задачах великої розмірності.

В цілому ж, виявляється, що для розв'язання цієї задачі за допомогою нейросіткових реалізацій досить наступних простих міркувань:

- для виконання умови 2 ваги сітки повинні бути побудовані таким чином, щоб кожний нейрон перешкоджав активації інших нейронів в своєму рядку та в своєму стовпці;
- для мінімізації довжини шляху необхідно, щоб нейрон в i -му стовпці тим активніше перешкоджав активації нейронів в $i+1$ -му та $i-1$ -му стовпцях, чим більша віддаль між ними;
- для того, щоб сітка Хопфілда взагалі працювала, необхідно, щоб ваги сітки не були всі від'ємні.

Всім цим умовам задовольняє наступна формула обчислення ваги між нейроном, відповідним місту x на позиції i в маршруті, та нейроном, відповідним місту y на позиції j :

$$\begin{aligned} W_{x_i, y_j} = & -H\delta_{xy}(1-\delta_{ij}) - K\delta_{ij}(1-\delta_{xy}) - \\ & -Lg(x, y)(\delta_{i, j+1} + \delta_{i, j-1}) + R, \end{aligned} \quad (6.46)$$

де H, K, L, R – деякі константи; $g(x, y)$ – віддаль між містами x та y ; δ_{xy} – символ Кронекера, який приймає значення 1, якщо $x = y$ та значення 0 в протилежному випадку.

Означення 6.4. Символ Кронекера (або дельта Кронекера) – індикатор рівності елементів, формально: функція двох цілих змінних, яка дорівнює 1, якщо вони є рівними, та 0 в протилежному випадку:

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (6.47)$$

Наприклад, $\delta_{12} = 0$, проте $\delta_{33} = 1$.

Легко переконатися, що перший член дорівнює $-H$ для всіх зв'язків в тому ж рядку ($x = y$), крім зв'язку нейрона з самим собою (при $i = j$). Другий член дорівнює $-K$ для всіх зв'язків в тому ж стовпці ($i = j$), крім зв'язку з самим собою ($x = y$). Третій член L пропорційний віддалі між містами x та y , якщо вони є сусідніми в маршруті ($i = j-1$) або ($i = j+1$).

Якщо таку сітку привести у випадковий початковий стан, то можна очікувати, що результуючий стабільний стан і визначить субоптимальний шлях, довжина якого не дуже перевершуватиме оптимальний (сам шлях може значно відрізнятись від оптимального). Для практичного застосування сітку треба перезапустити декілька раз і вибрати найкращий шлях.

Розв'язання даної задачі цікаве не стільки своєю якістю (існують більш ефективні розв'язуючі алгоритми), скільки самим підходом до задач оптимізації: якщо можливо перевести умови деякої задачі в параметри зв'язків між нейронами, то вона може бути відносно непогано розв'язана НС без будь-якого додаткового аналізу.

6.8. Нейронна сітка Хеммінга

Загальна характеристика. У випадку, якщо необхідно визначити номер еталону, найближчого до поданого вхідного вектору, можна використати сітку Хеммінга (*Hamming*). Сітка Хеммінга є розширенням сітки Хопфілда і була розроблена Ричардом Ліппманом (*Richard Lippman*) у середині 80-х років минулого століття [166].

Сітка Хеммінга реалізує класифікатор, що базується на найменшій похибці для векторів двійкових входів, де похибка визначається віддаллю Хеммінга.

Твердження 6.9. Віддаль Хеммінга визначається кількістю бітів, які відрізняються між двома відповідними вхідними векторами фіксованої довжини: незашумленого прикладу образу і спотвореного образу.

Хеммінгова віддаль уявляє собою приклад міри схожості або, скоріше, відмінності, первісно уведеної для бінарних функцій у діадному просторі. Вона є застосовною для порівняння будь-яких упорядкованих наборів, які приймають дискретні значення і, імовірно, є найкращою з усіх відомих мір схожості між цифровими кодами. Для бінарних послідовностей хеммінгову віддаль можна визначити:

$$\rho(x, x') = bc\{(\bar{x}_i \wedge x'_i) \vee (x_i \wedge \bar{x}'_i) | i = 1, \dots, n\}, \quad (6.48)$$

де функція $bc\{*\}$ визначається як кількість елементів набору $\{*\}$, приймаючих значення логічної «1».

Вектор виходів навчальної множини є *вектором класів*, до яких належать образи. У режимі навчання вхідні вектори розподіляються до категорій, для яких відстань між зразковими вхідними векторами та поточним вхідним вектором є мінімальною. Іншими словами, сітка Хеммінга реалізує паралельне обчислення віддалей Хеммінга між пред'явлюваним вхідним вектором і збережуваними у пам'яті образами.

Задача сітки полягає у тому, щоб при пред'явленні їй деякого невідомого двійкового вхідного образу x вона видавала асоційований з ним збережуваний у сітці образ x_k . Отже, вона працює за принципом *асоціативної пам'яті*. При цьому на виході сітки з'являється не сам асоціативний образ, а його номер k у вигляді сигналу, який ви-

никає на виході k -го нейрона шару на виході i має максимальне значення, в той час як на виходах інших нейронів сигнали будуть слабкими.

На відміну від сітки Хопфілда ємність сітки Хеммінга визначається не розмірністю N вхідного сигналу, а кількістю нейронів K у шарі на виході. Перевагами цієї сітки у порівнянні з сіткою Хопфілда є менші витрати на пам'ять та обсяг обчислень.

Нейронна сітка Хеммінга (рис. 6.8) складається з вхідного та схованого (*категорій*) шарів (які утворюють сітку прямого поширення), а також шару нейронів на виході.

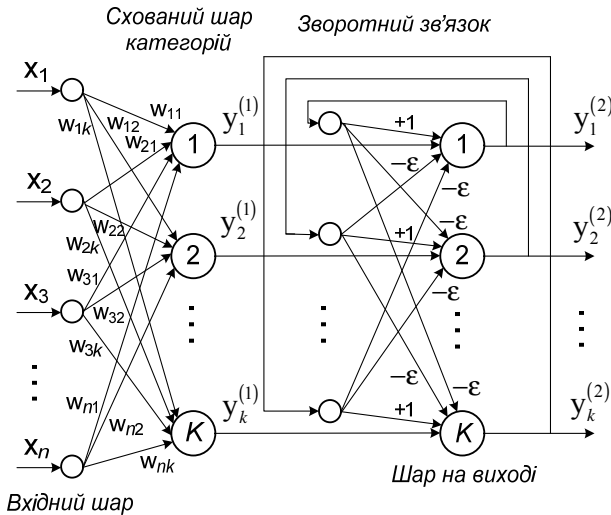


Рис. 6.8. Структура нейронної сітки Хеммінга

Кількість вузлів вхідного шару відповідає кількості окремих двійкових ознак у образі. Кількість K нейронів схованого шару (шару Хопфілда) відповідає кількості категорій або класів. Шар на виході відповідає кількості K вузлів у схованому шарі категорій, причому K – кількість еталонів. Нейрони схованого шару k синапсами з'єднані з виходами нейронів вхідного шару сітки. Виходи нейронів шару на виході зв'язані із входами решти нейронів цього шару від'ємними зворотними (інгібіторними) зв'язками $-\epsilon$. Єдиний додатний $(+1)$ зворотний зв'язок подається з виходу кожного нейрона шару на виході на його ж вхід. Сигнал з виходу схованого шару категорій, сформований через конкуренцію, надходить на входи рекурсивного шару на виході, вагова матриця якого має вигляд:

$$\mathbf{W}^{\text{вих}} = \begin{vmatrix} 1 & -\varepsilon & \dots & -\varepsilon \\ -\varepsilon & 1 & \dots & -\varepsilon \\ \dots & \dots & \dots & \dots \\ -\varepsilon & -\varepsilon & \dots & 1 \end{vmatrix}, \quad (6.49)$$

де $\varepsilon < \frac{1}{M}$; M – збережуваних у пам'яті образів.

Нейрони цього шару, який називають сіткою MAXNET, має зворотні зв'язки, тобто шар є рекурсивним, тому сигнал на виході сітки описується співвідношенням:

$$\mathbf{y}(t) = \mathbf{W}^{\text{вих}} \mathbf{y}(t-1). \quad (6.50)$$

Сітка вибирає еталон з мінімальною хеммінговою віддаллю від представленого вхідного вектору шляхом активації тільки одного виходу сітки (нейрона шару на виході), відповідного до цього еталону.

Алгоритм функціонування сітки. Розглянемо послідовність функціонування сітки Хеммінга:

Крок 1. На етапі ініціалізації (настроювання) сітки Хеммінга встановлюються наступні значення ваг нейронів схованого шару і порогу їх активаційної функції:

$$w_{ik} = \frac{x_i^{(k)}}{2}, \quad \theta_z = \frac{n}{2}, \quad (6.51)$$

де $x_i^{(k)}$ – i -й компонент k -го еталону; $i = 1, \dots, n$, $k = 1, \dots, K$.

Коефіцієнти від'ємних гальмуючих синапсів (зворотних зв'язків) нейронів шару на виході задають дорівнюючими деякій величині з інтервалу $0 < e < 1/K$, а коефіцієнти додатного зворотного зв'язку $+1$.

Крок 2. На нейрони вхідного шару подається вектор $\mathbf{X} = \{x_i\}$, $i = 1, \dots, I$.

На їх виходах формуються наступні значення (верхній індекс указує номер шару):

$$y_k^{(1)} = \sum_{i=1}^n w_{ik} x_i + \theta_j, \quad k = 1, \dots, K. \quad (6.52)$$

У відповідності до цього установлюються значення на виходах нейронів шару на виході:

$$y_k^{(2)} = y_k^{(1)}, \quad z = 1, \dots, K. \quad (6.53)$$

Крок 3. В результаті нової $(t+1)$ -ї ітерації визначаються нові стани нейронів шару на виході:

$$s_k^{(2)}(t+1) = y_k^{(2)}(t) - \varepsilon \sum_{j=1(j \neq k)}^n y_j^{(2)}(t), \quad k=1, \dots, K, \quad (6.54)$$

$$y_k^{(2)}(t+1) = f[s_k^{(2)}(t+1)], \quad k=1, \dots, Z. \quad (6.55)$$

Активаційна функція f має вигляд порогу, причому величина «приступка» повинна бути досить великою, щоб можливі значення $s_z^{(2)}$ не приводили до насичення.

Крок 4. Перевірка зміни станів нейронів шару на виході за останню ітерацію і перехід до кроку 3 у випадку, якщо спостерігалися зміни. Інакше – закінчення процедури.

Роль нейронів вхідного шару досить умовна: скориставшись один раз на кроці 2 значеннями його вагових коефіцієнтів, сітка більше не звертається до нього, через що цей шар можна взагалі усунути з сітки (може бути замінений на матрицю вагових коефіцієнтів).

На закінчення можна зробити наступне узагальнення:

- сітки Хопфілда та Хеммінга дозволяють просто та ефективно розв'язати задачу ААП – *відтворення образів за неповною та спотвореною інформацією*;

- *невисока ємність сіток* (кількість запам'ятовуваних образів) пояснюється тією обставиною, що сітки не просто запам'ятовують образи, а й дозволяють проводити їх узагальнення, (наприклад, за допомогою сітки Хеммінга можлива класифікація за критерієм максимальної правдоподібності);

- *легкість побудови програмних і апаратних моделей* роблять ці сітки привабливими у багатьох практичних застосуваннях.

Разом з тим, є декілька зауважень щодо обмежень у використанні таких сіток. Так, в разі, якщо необхідно визначити еталон, найближчий до пред'явленого вхідному вектору (наприклад, на основі хеммінгової віддалі), часто виникають проблеми, пов'язані з різницею довжин або обмеженнями на довжину послідовностей або кількість компонентів у наборах навчаючої вибірки. Тут треба роз'яснити один важливий аспект проблеми порівняння векторів різної довжини. Так, послідовність більшої довжини може уявляти собою функцію у просторі з кількістю вимірів більшою, ніж кількість вимірів у іншій функції на різницю числа параметрів у довжинах послідовностей. Це визначає можливість переходу до оцінки не тільки кількісної, але й якісної подібності (схожості) векторів, до оперування у термінах імовірності, нечітких оцінювань, зарахування вектора до якого-небудь класу.

Ці відмінності та обмеження у порівнянні векторів різної довжини можна усунути деякими способами, наприклад, за допомогою *міри схожості Танимото*.

6.9 Двоспрямована асоціативна пам'ять

Загальні відомості. Двоспрямована асоціативна пам'ять (ДАП) (англ.: *Bidirectional Associative Memory*) розроблена Бартом Коско (*Bart Kosko*) [157; 158] і є узагальненням сітки Хопфілда на випадок двошарової рекурентної структури, яка дозволяє кодувати множини двох взаємопов'язаних векторів. Як і у випадку сітки Хопфілда-Хеммінга, ДАП спроможна до узагальнень, виробляючи вірні реакції, не дивлячись на спотворені входи. Проте, на відміну від цих типів сіток і розширюючи модель Хопфілда, ДАП розв'язує задачі гетеро-асоціативної пам'яті, тобто вхідний образ може асоціюватися з іншим, не корельованим з ним чином. Реалізується це внаслідок того, що вектор на виході формується на іншому наборі нейронів, аніж відповідний до нього вхідний.

Образи, які надходять на входи \mathbf{x} , представлені N -вимірними векторами, а на виході \mathbf{y} – у загальному випадку Z -вимірними. Таким чином, як і для будь-якого типу асоціативної пам'яті, асоціації формуються у ваговій матриці \mathbf{W} розмірністю $N \times Z$. Отже, ємність ДАП не може перевищувати меншого з чисел N або Z . Більше того, реально вона є меншою меншого з цих чисел і визначається кількістю лінійно незалежних рядків (стовбців) матриці \mathbf{W} . Елемент матриці w_{ij} ($i = \overline{1, N}; j = \overline{1, Z}$), зв'язуючий i -й нейрон образу з j -м нейроном образу, використовується при аналізі сигналів, які можуть бути як дискретними, поданими у двійковій формі (бінарні або біполярні), так і неперервними, реалізуючи *дискретні* та *неперервні* ДАП. У відповідності до цього активаційні функції нейронів можуть вибиратися релейними або сигмоїдними. При використанні релейної активаційної функції у дискретній ДАП, як показав Б. Коско, більш ефективним є біполярне кодування сигналів.

Можлива *синхронна* та *асинхронна* робота ДАП.

Дана сітка має багато загального з НС адаптивної резонансної теорії та сіткою Хопфілда. Як і сітка Хопфілда, робота ДАП може бути проаналізована з енергетичної точки зору. Вводячи поняття двоспрямованої стійкості сітки Коско довів, що енергія ДАП є обмеженою і при роботі сітки не зростає. Збіжність сітки до усталеного стану, при якому стани нейронів не змінюються, характеризує її рух до енергетичного мінімуму, відповідаючому визначеному асоційованому образу. Якщо ж кількість збережуваних образів перевищує максимально допустиме, сітка видає помилковий сигнал.

На рис. 6.9 наведена базова структура ДАП, яка складається з вхідного, схованого шарів і шару на виході. Сигнали розповсюджуються у двох напрямках: від входу до виходу і зворотно. Функціонування має синхронний характер. Це означає, що якщо у першому циклі сигнали спочатку проходять в один бік для визначення стану нейронів-отримувачів, то у наступному циклі вони самі стають джерелом, надсилаючи сигнали в зворотному напрямку. Цей процес повторюється до досягнення стану рівноваги.

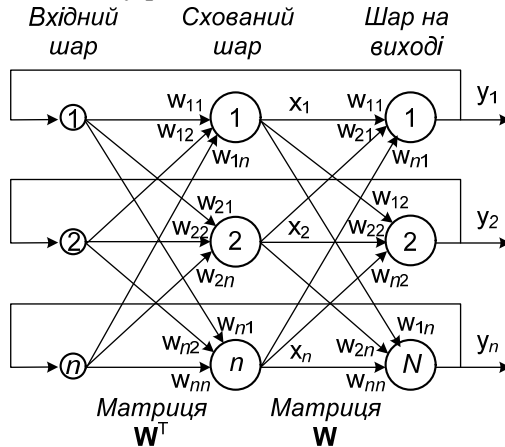


Рис. 6.9. Структура двоспрямованої асоціативної пам'яті

Функція активації нейронів має пороговий характер: вона може бути двійковою зі значеннями 1 або 0 або біполярною зі значеннями ± 1 . При нульовому сигналі збудження нейрона його поточний стан залишається дорівнюваним попередньому стану. Для забезпечення кращих характеристик сітки у режимі розпізнавання на етапі навчання використовуються тільки біполярні сигнали.

Матриця ваг W , зв'язуюча обидві частини сітки, є дійсною і несиметричною. З урахуванням симетрії зв'язків вхідного і на виході шарів НС при прямому напрямку поширення сигналів ваги описуються матрицею W , а при протилежному напрямку – матрицею W^T .

Уявимо, що вхідні навчаючі дані визначені у вигляді множини з m біполярних пар $\{(a_i, b_i)\}$, де $a_i = [a_{i1}, a_{i2}, \dots, a_{in}]$, $b_i = [b_{i1}, b_{i2}, \dots, b_{in}]$ (вектори-рядки). Цій множині співставляється множина біполярних пар $\{(x_i, y_i)\}$, де x_i – це біполярне подання a_i ($0 \rightarrow -1$, $1 \rightarrow 1$), а y_i – біполярне подання b_i .

У відповідності з означенням Б. Коско [158] матриця ваг W формується на основі множини $\{(x_i, y_i)\}$ як матриця кореляції

$$W = \sum_{i=1}^m x_i^T y_i, \quad (6.56)$$

Показано, що використання біполярних навчаючих векторів дає кращі результати на стадії розпізнавання. Визначення ваг міжнейронних зв'язків дозволяє простежити процес стабілізації стану на обох кінцях сітки.

Якщо припустити, що вихідний стан сітки був заданий парою (x_0, y_0) , то процес двоспрямованого опрацювання сигналів складається з послідовності наступних циклів:

$$\begin{aligned} f(x_0 W) = y_1 &\rightarrow f(y_1 W^T) = x_1 \rightarrow f(x_1 W^T) = y_2 \rightarrow \\ &\rightarrow f(y_2 W^T) = x_2 \rightarrow f(x_2 W^T) = y_3 \rightarrow \\ &\rightarrow \dots \dots \dots \rightarrow \\ &\rightarrow f(y_f W^T) = x_f \rightarrow f(x_f W^T) = y_f, \end{aligned} \quad (6.57)$$

в результаті чого формуються дві стабільні величини x_f та y_f , що свідчить про досягнення стабільного стану сітки. У випадку бінарного опису вихідного стану у виді (a_0, b_0) біполярним величинам (x_f, y_f) співставляються бінарні представлення (a_f, b_f) . Кожній проміжній точці (x_k, y_k) процесу можна співставити енергетичну функцію E_k , визначувану в виді

$$E_k = -x_k W y_k^T. \quad (6.58)$$

Доведено, що кожне чергове змінювання стану перехідного процесу веде до зменшення значення енергетичної функції сітки аж до досягнення локального мінімуму. Цей мінімум досягається за скінчену кількість ітерацій, і він має значення

$$E_{\min} = -x_f W y_f^T. \quad (6.59)$$

Іншими словами, будь-який інший розв'язок (в тому числі і найближче, яке відрізняється лише на 1 в сенсі міри Хеммінга від (x_f, y_f)) буде характеризуватися великим значенням енергетичної функції. При виконанні деяких додаткових умов парою (x_f, y_f) стає одна з навчаючих пар, що бере участь у формуванні матриці W , яка є найбільш подібною (найближча за мірою Хеммінга) до пари (x_0, y_0) , яка визначала вихідний стан.

Особливості функціонування сітки. Нейрони схованого шару та шару на виході () виконують функцію зваженого підсумовування входів з сигмоїдальною (логістичною) активаційною функцією f . У спрощеному вигляді як активаційну вибирають порогову функцію. До початку функціонування НС навчається з використанням набору пар

векторів \mathbf{X} і \mathbf{Y} шляхом обчислення значень вагових коефіцієнтів \mathbf{W} і \mathbf{W}^T , які реалізують відношення асоціацій. При цьому вагова матриця \mathbf{W} обчислюється як сума добутків усіх N пар векторів з навчаючої вибірки:

$$\mathbf{W} = \sum_{i=1}^N \mathbf{X}_i^T \mathbf{Y}_i, i = \overline{1, N}. \quad (6.60)$$

При розв'язанні задачі відтворення запам'ятованих асоціацій вектор \mathbf{X} або його частина короткочасно установлюються на виходах нейронів схованого шару. Вектор \mathbf{X} опрацьовується матрицею вагових коефіцієнтів \mathbf{W} нейронів шару на виході. Далі вектор \mathbf{X} вилучається, і сітка виробляє асоційований вектор \mathbf{Y} на виході нейронів вхідного шару, який надходить на входи нейронів вхідного шару та опрацьовується транспонованою матрицею \mathbf{W}^T ваг нейронів схованого шару:

$$\mathbf{Y} = f(\mathbf{X}\mathbf{W}), \mathbf{Y} = f(\mathbf{X}\mathbf{W}^T). \quad (6.61)$$

При цьому у кожному циклі відбувається уточнення вектора на виході. Процес повторюється до досягнення сталого стану сітки, за яким вектори \mathbf{X} і \mathbf{Y} не змінюються.

Функціонує ДАП у напрямку мінімізації енергії сітки (функція Ляпунова) у відповідності до значень ваг. Взагалі системи із зворотним зв'язком мають тенденцію до коливань. Вони можуть переходити від стану до стану, ніколи не досягаючи стабільності. Доведено, що ДАП безумовно стабільна при будь-яких значеннях ваг сітки.

Твердження 6.10. Сталість ДАП гарантується транспонуванням матриці вагових коефіцієнтів.

Проводячи аналогію з біологічними системами, можна відзначити, що значення вагових коефіцієнтів \mathbf{W} і \mathbf{W}^T утворюють довготривалу пам'ять, а стани нейронів – короткочасну пам'ять. Вагові коефіцієнти можуть змінюватися на більш довготривалому інтервалі часу на основі розроблених для цього методів з метою виведення сітки з локального та глобального оптимуму для первісно встановленого вектора \mathbf{X} .

Взагалі ДАП зводиться до сітки Хопфілда, якщо матриця вагових коефіцієнтів \mathbf{W} є квадратною і симетричною, тобто $\mathbf{W} = \mathbf{W}^T$.

Двоспрямована асоціативна пам'ять має обмеження на максимальну кількість збережених асоціацій, при перевищенні якої сітка може виробити невірний сигнал на виході, відтворюючи асоціації, яким не була навчена.

Для безпомилкової роботи бінарної ДАП кількість N за-

пам'ятовуваних векторів не повинна перевищувати значення $N \leq n/(2 \log_2 n)$.

Відомі оцінки, за якими ДАП може мати до 2^n стабільних станів, якщо для кожного нейрона вибирається своє ненульове порогове значення θ_i . Така сітка, яка отримала назву *негомогенної ДАП*, є розширенням гомогенної ДАП, у якій усі пороги нульові. В цьому випадку вихід нейрона приймає наступний вигляд:

$$y_i(t+1) = 1, \quad \text{якщо } s_i(t) > \theta_i,$$

$$y_i(t+1) = 1, \quad \text{якщо } s_i(t) < \theta_i,$$

$$y_i(t+1) = y_i(t), \quad \text{якщо } s_i(t) = \theta_i.$$

де $y_i \{f\}$ – вихід i -го нейрона на момент часу t .

Проте вибір цих станів визначається жорсткою процедурою. Якщо обрано N станів випадковим чином, причому

$$N < \frac{0,68n^2}{(\log_2 n + 4)^2}, \quad (6.62)$$

і якщо кожний вектор має $(4 + \log_2 n)$ компонентів, які дорівнюють «1» (інші компоненти дорівнюють «0»), то можна сформуванню негомогенну ДАП, яка зберігатиме $0,98 \log_2 n$ цих векторів як такі, що мають стабільні стани.

Створено багато видозмін ДАП, основними з яких є: неперервна ДАП (з сигмоїдами в якості функцій активації нейронів), адаптивна ДАП (із змінюванням ваг в процесі функціонування сітки), конкуруюча ДАП (з конкуренцією нейронів всередині кожного шару).

Основні достоїнства ДАП:

- структурна простота сітки, що дозволяє реалізувати її у вигляді СБІС и УБІС;
- сумісність з аналоговими схемами та оптичними системами; швидка збіжність процесу навчання і відтворення інформації.

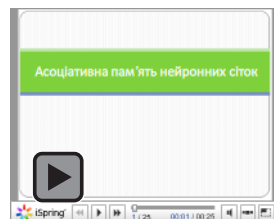
Контрольні завдання

1. Навести паралелі та відмінності між прецизійним та асоціативним зв'язками в нейросполученнях мозку.
2. Охарактеризувати асоціативний спосіб доступу до інформації у запам'ятовуючому середовищі.
3. Навести означення асоціації і виокремити основні елементи асоціативної пам'яті.
4. Розкрити зміст понять “асоціація”, “гетероасоціація”, “автоасоціація”.

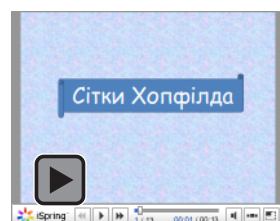
5. Охарактеризувати особливості пам'яті з адресацією за змістом (навести відмінності від адресації за адресою).
6. Проаналізувати модель асоціативної пам'яті на етапах записування (навчання) і функціонування (зчитування).
7. Розкрити вміст віддалі Хеммінга і пояснити принципи відновлення асоціативною пам'яттю спотворених образів.
8. Охарактеризувати властивості навчання асоціативної пам'яті при використанні правила Д. Хебба.
9. Охарактеризувати властивості навчання асоціативної пам'яті при використанні дельта-правила.
10. Пояснити процедуру розпізнавання асоціативною пам'яттю спотворених образів.
11. Охарактеризувати і окреслити сферу використання гетероасоціативної пам'яті.
12. Охарактеризувати і окреслити сферу використання автоасоціативної пам'яті як частинного випадку гетероасоціативної пам'яті.
13. Розкрити особливості побудовання і охарактеризувати суттєві ознаки нейронної сітки Хопфілда.
14. Прокоментувати роботу сітки Хопфілда з точки зору "Гамільтонової енергетичної функції".
15. Проаналізувати можливості динамічної (синхронної та асинхронної) зміни станів в сітці Хопфілда.
16. Розкрити алгоритм послідовності функціонування сітки Хопфілда.
17. Навести і розтлумачити використання сіток Хопфілда в розв'язанні задач оптимізації на прикладі задачі про комівояжера.
18. Дати характеристику нейронної сітки Хеммінга як розширення сітки Хопфілда (прокоментувати Хеммінгову віддаль як міру схожості або відмінності).
19. Розкрити алгоритм послідовності функціонування нейронної сітки Хеммінга.
20. Пояснити принципи функціонування двоспрямованої асоціативної пам'яті Б. Коско в умовах синхронної та асинхронної роботи.
21. Прокоментувати вимоги щодо матриці ваг, зв'язуючої обидві частини нейронної сітки Б. Коско.
22. Навести і прокоментувати умови зведення двоспрямованої асоціативної пам'яті до сітки Хопфілда.

Презентації

1. Асоціативна пам'ять нейронних сіток



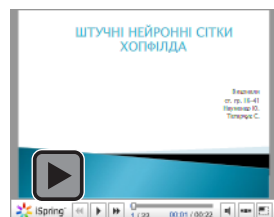
2. Сітки Хопфілда



3. Нейронні сітки асоціативної пам'яті



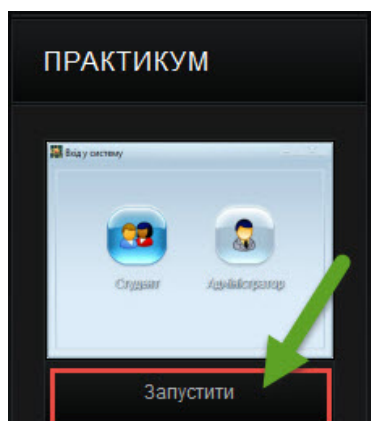
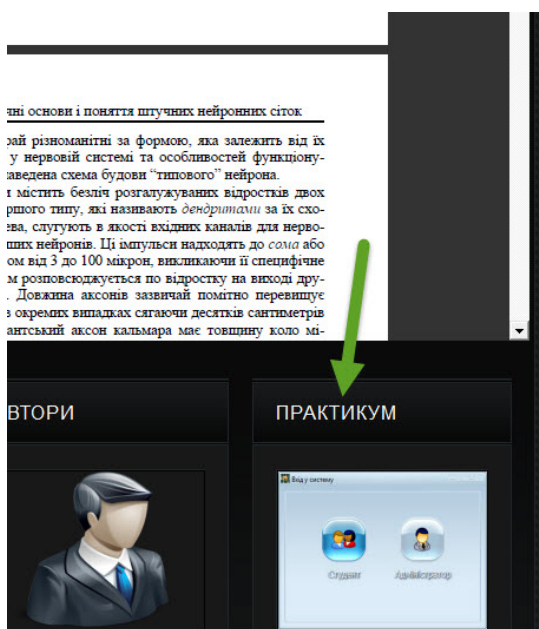
4. Штучні нейронні сітки Хопфілда



Практичні заняття

Асоціативні нейронні сітки

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 7

Нейронні сітки Кохонена

*Красивими чоловіками жінки милуються,
розумних боготворять,
в добрих влюбляються, сміливих бояться,
проте виходять заміж охоче тільки за сильних*
Василь Ключевський

Смелі все дається нелегко... Проте – все
Афоризм

7.1. Загальні відомості про нейронні сітки Кохонена

Загальні зауваження. Це окремих клас ШНС, які використовуються для розв'язання всіляких задач класифікації та похідних від них. Уявляє собою багат шарові НС прямого поширення, основним елементом яких є шар Кохонена, який складається з адаптивних лінійних суматорів (“лінійних формальних нейронів”). Для навчання цих ШНС, як правило, застосовується алгоритм зворотного поширення похибки; ці сітки можна схарактеризувати як базові НС. Вони успішно застосовуються при розв'язанні широкого кола задач, проте у деяких областях їх використання або неможливе, або неефективне через тривалий час навчання.

Для використання нейронних сіток Кохонена в задачах класифікації необхідна деяка формалізація. Кожний об'єкт, який вимагає класифікації, подається у вигляді деякого вектора, подаваного на вхід НС. *Кількість нейронів вхідного шару* визначається кількістю компонентів цього вхідного вектора (і може, наприклад, визначатися кількістю властивостей об'єкта класифікації). *Кількість же виходів* визначається кількістю класів: якщо є M класів, то й кількість нейронів у шарі на виході буде також M . Таким чином, кожний нейрон шару на виході “відповідає” за свій клас. Значення, які приймають нейрони у шарі на виході, відбивають, наскільки вектор класифікованого об'єкта на вході є близьким, за “думкою” НС Кохонена, до того чи іншого класу. Чим більша “впевненість”, що об'єкт належить до того чи іншого класу, тим більшого значення набуває нейрон відповідного класу. Іноді застосовують спеціальну функцію активації, яка робить суму виходів з усіх нейронів дорівнюваній одиниці. У такому випадку кожний вихід можна трактувати як ймовірність того, що об'єкт належить до даного класу.

Слід зазначити, що існує і більш проста реалізація НС Кохонена, яка частіше відома як “переможець отримує все”. В такому випадку кожний нейрон шару на виході може приймати значення або нуль, або одиниця. При цьому для одного вхідного вектора одиниці може дорівнювати один і тільки один нейрон шару на виході, тобто один об’єкт не може відноситися одразу до двох класів.

Сигнал в сітку Кохонена надходить одночасно на всі нейрони, ваги відповідних синапсів інтерпретуються як координати положення вузла, і сигнал на виході формується за принципом “переможець отримує все” – тобто ненульовий на виході сигнал має нейрон, найближчий (в сенсі ваг синапсів) до подаваного на вхід об’єкта. В процесі навчання ваги синапсів налаштовуються таким чином, щоб, з одного боку, вузли грат “розташовувалися” в місцях локальних згущень даних, тобто описували кластерну структуру хмари даних, а з іншого боку, щоб зв’язки між нейронами відповідали відношенням сусідства між відповідними кластерами у просторі ознак.

Зручно розглядати такі мапи як двовимірні сітки вузлів, розташованих у багатовимірному просторі. Первісно *самоорганізована мапа* уявляє собою сітку з вузлів, з’єднаних між собою зв’язками. Кохонен розглядав два варіанта з’єднання вузлів – у прямокутну та гексагональну сітку, і відмінність полягає в тому, що у прямокутній сітці кожний вузол з’єднаний з 4-ма сусідніми, а у гексагональній – з шістьма найближчими вузлами. Для двох таких сіток процес побудови сітки Кохонена відрізняється лише в тому місці, де перебираються найближчі до даного вузла сусіди.

Розглянемо наступну просту нейронну сітку Кохонена з трьома входами і двома виходами, тобто НС для класифікації всього за двома класами (рис. 7.1).

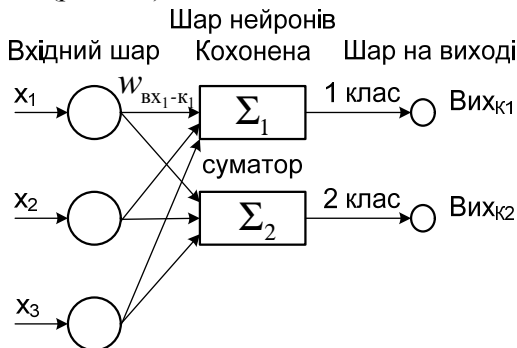


Рис. 7.1. Проста нейронна сітка Кохонена

Принцип функціонування вже навченої НС Кохонена наступний. Нейрони вхідного шару (на рис. 1 позначені великими колами) не виконують ніяких обчислень і слугують тільки точками розгалуження складових вхідних векторів. Кожний нейрон вхідного шару з'єднаний з кожним нейроном схованого шару нейронів, саме який і називають *шаром Кохонена*, окремою вагою: так, наприклад, перший нейрон вхідного шару з'єднаний з першим нейроном шару Кохонена вагою $w_{Bx_1-K_1}$. Нейрони шару Кохонена складають надходячі значення і подають їх на нейрони шару на виході: один нейрон шару Кохонена – один нейрон шару на виході:

$$\text{Вих}_K = (w_{Bx_1-K} \times x_1) + (w_{Bx_2-K} \times x_2) + (w_{Bx_3-K} \times x_3) \quad (7.1)$$

або:

$$\text{Вих}_{Km} = \sum_n^{n=1..3} (w_{Bx_n-K} \times x_n), \quad (7.2)$$

де: Вих_{Km} – це вихід m -го нейрона шару Кохонена; $x_1...x_3$ – сигнали вхідного вектора **X**.

Як правило, на виході НС ще встановлюють деякий інтерпретатор: нейрони шару Кохонена генерують сигнали Вих_{Km} , інтерпретатор вибирає максимальний сигнал і видає номер класу M , до якого необхідно віднести класифіковуваний об'єкт.

Сітка розпізнає кластери в навчальних даних і розподіляє дані до відповідних кластерів. Якщо в наступному сітка зустрічається з набором даних, несхожим ні з одним із відомих зразків, вона відносить його до нового кластеру. Якщо в даних містяться мітки класів, то сітка спроможна вирішувати задачі класифікації. Сітки Кохонена можна використовувати і в задачах, де класи відомі – перевага буде у спроможності сітки виявляти подібність між різноманітними класами.

Зазвичай прийнято вважати, що сітка Кохонена має всього два шари (рис. 7.2): вхідний і вихідний, що називають *самоорганізованою мапою*. Елементи мапи розташовуються в деякому просторі - як правило, двовимірному.

Сітка, розроблена Тойво Кохоненом [153], принципово відрізняється від розглянутих вище сіток, оскільки використовує неконтрольоване навчання, і навчальна множина складається лише із значень вхідних змінних. Таким чином, навчання сітки Кохонена є навчанням без учителя і так само, як і звичайної нейронної сітки, *полягає в підналагоджуванні ваг зв'язків між нейронами*, проте здійснюється *методом конкурентного навчання*.

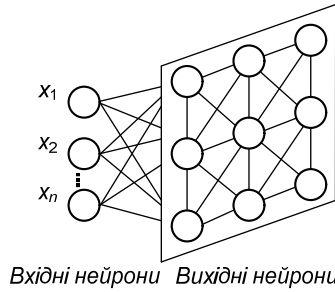


Рис. 7.2. Сітка Кохонена

7.2. Навчання нейронної сітки Кохонена

Означення 7.1. *Конкурентне навчання* – методика навчання, за якою нейрони конкурують один з одним, щоб вектор їх ваг виявився якомога ближче до вектора ознак об'єкта, який пред'являється.

Конкурентне навчання уявляє собою ітераційний процес, в якому на кожній ітерації виконується 3 дії:

- *конкуренція;*
- *об'єднання;*
- *підналагодження ваг.*

При конкуренції на вхід сітки подається вхідний вектор ознак і здійснюється пошук нейрона з найбільш близьким до нього набором ваг. Такий нейрон оголошується *переможцем*.

В процесі об'єднання навколо нейрона-переможця утворюється група (сусідство) нейронів, які будуть приймати участь в процесі навчання. Розмір групи визначається *радіусом навчання*.

На етапі підналагодження відбувається коректування ваг нейронів, розташованих в межах радіусу навчання від нейрона-переможця, таким чином, щоб їх вектори стали ближче до нього (див. рис. 7.3).

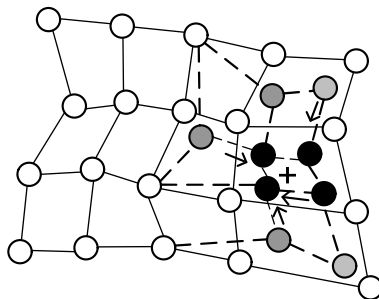


Рис. 7.3. Решітка нейросітки Кохонена

Це підвищує шанси нейрона стати переможцем на наступній ітерації при надходженні схожого прикладу.

Таким чином, сітка Кохонена навчається методом послідовних наближень. Починаючи з випадковим чином обраного вихідного розташування центрів, алгоритм поступово покращується для класифікації навчальних даних.

Проте, алгоритм може працювати і на іншому рівні. В результаті ітеративної процедури навчання сітка організовується таким чином, що елементи, які відповідають центрам, розташованим близько один від одного в просторі входів, розташовуватимуться близько один від одного і на топологічній мапі. Топологічний шар сітки можна уявити як *двовимірну решітку* (рис. 7.3), яку потрібно так відобразити в N -вимірний простір входів, щоб по можливості зберегти вихідну структуру даних.

Ясна річ, при будь-якій спробі відтворити N -вимірний простір на площині буде загублено багато деталей, але такий прийом дозволяє користувачу візуалізувати дані, що неможливо зрозуміти іншим засобом.

Основний ітераційний алгоритм Кохонена послідовно проходить ряд *epoch*, на кожній епосі опрацьовується один навчальний приклад. Вхідні сигнали - вектори дійсних чисел - послідовно пред'являються сітці. Бажані сигнали на виході не визначаються. Після пред'явлення достатнього числа вхідних векторів, синаптичні ваги сітки визначають кластери. Крім того, ваги організуються так, що топологічно близькі вузли чуттєві до схожих вхідних сигналів.

Для реалізації алгоритму необхідно визначити *міру сусідства нейронів* (окіл нейрона-переможця). На рис. 7.4 показані зони топологічного сусідства нейронів на мапі ознак у деякий момент часу t . Сусідами нейрона-переможця j (позначений чорним колом) у момент часу t вважається $NE_j(t)$ - множина нейронів. Зони сусідства зменшуються з часом.

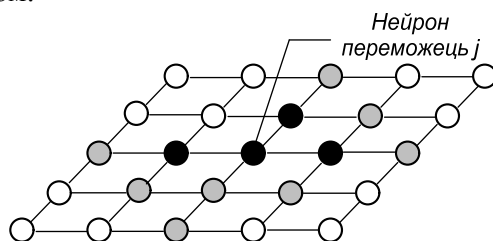


Рис. 7.4. Зони топологічного сусідства на мапі ознак у деякий момент часу t

Через те, що ваги всіх затемнених (позначені сірими колами) нейронів змінюються по-різному в залежності від їх віддаленості від нейрона-переможця (що можна простежити і з рис. 7.3), найбільш простим є вибір в якості f_{ij} деякої величини, що дорівнює одиниці при $j = i$, меншій за одиницю для затемнених (сірий колір) нейронів (тобто нейронів, які лежать в безпосередній близькості від активованого нейрона), і нулю для всіх інших, помічених світлими кружечками.

На практиці в якості функції f_{ij} вибирають функції, які використовують *евклідову метрику*

$$d = \sum_k (r_{ik} - r_{jk})^2, \quad (7.3)$$

де $r_{ik}; r_{jk}$ – координати i -го і j -го нейронів.

До найбільш уживаних потенціальних функцій відноситься дзвоноподібна функція Гаусса:

$$f_{gauss}(d, \sigma) = e^{-d^2 / 2\sigma^2}, \quad (7.4)$$

де σ^2 – дисперсія відхилення.

7.3. Побудування мапи Кохонена

Як правило, спочатку будують доволі грубу мапу (модель розбиття), поступово уточнюючи її в процесі навчання. Для цього необхідно повільно змінювати не тільки параметр α – коефіцієнт підсилення (який зазвичай на першій ітерації приймають $\alpha = 1$), але й, наприклад, параметр σ^2 у (7.4).

Одним з ефективних способів змінення цих параметрів є наступний [48]:

$$\begin{aligned} \alpha(k) &= \alpha_{(0)} \left[\alpha_{\min} / \alpha_{(0)} \right]^{k/k_{\max}}; \\ \sigma(k) &= \sigma_{(0)} \left[\sigma_{\min} / \sigma_{(0)} \right]^{k/k_{\max}}, \end{aligned} \quad (7.5)$$

де $\alpha_{(0)} \approx 0,8$; $\alpha_{\min} \ll 1$; $\sigma_{(0)} = 0,2$; $\sigma_{\min} = 0,5$ – параметри, які визначають крутість функції f_{ij} ; k_{\max} – задавана кількість ітерацій.

При реалізації сітки Кохонена спіткаються з наступними проблемами:

1. *Вибір коефіцієнта навчання* (підсилення). Цей вибір впливає як на швидкість навчання, так і на сталість отриманого розв'язку.

Із зростанням $\alpha \rightarrow 1$ збіжність алгоритму навчання зростає. Проте, у цьому випадку пред'явлення сітці всіляких вхідних векторів, які відносяться до одного класу, приведе до змінням відповідного вектора вагових коефіцієнтів. Навпаки, при $\alpha \rightarrow 0$ швидкість навчання буде повільною, проте вектор вагових коефіцієнтів, сягнувши центра класу, при надходженні на вхід сітки різних сигналів, які належать до одного класу, буде залишатися поблизу цього центру.

2. *Вибір параметра α .* Одним із шляхів прискорення процесу навчання при одночасному забезпеченні отримання сталого розв'язку є вибір параметра α : з $\alpha \rightarrow 1$ на початкових етапах навчання до $\alpha \rightarrow 0$ – на заключних. На жаль, такий підхід неможна застосовувати, коли сітка повинна неперервно підналагоджуватися до нових вхідних сигналів, які на неї надходять.

3. *Рандомізація ваг.* Ця операція може породжувати серйозні проблеми при навчанні через те, що в результаті неї вагові вектори розподіляються рівномірно по поверхні гіперсфери. Як правило ж, вхідні вектори розподілені нерівномірно і групуються на відносно малій частині поверхні гіперсфери. Тому більшість вагових векторів будуть настільки віддаленими від будь-якого вхідного вектора, що не заактивуються і стануть некорисними. Крім того, залишившихся активованих нейронів може бути замало, щоб розбити недалеко розташовані вхідні вектори на кластери.

4. *Вибір початкових значень векторів вагових коефіцієнтів нейронів.* Якщо початкові значення обрані невдало, тобто розташованими далеко від надходжуваних вхідних векторів, то нейрон не стане переможцем ні за яких вхідних сигналів, а, відтак, не навчиться.

5. *Вибір параметра віддалі σ .* Якщо спочатку параметр σ вибраний замалим або дуже швидко зменшується, то далеко розташовані нейрони не можуть впливати один на одного. І хоча дві частини у такій мапі настроюються вірно, загальна мапа буде мати *топологічний дефект* (рис. 7.5 [217]).

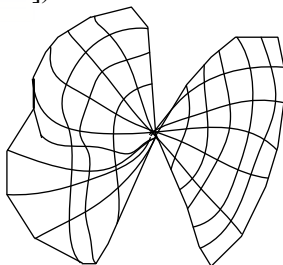


Рис. 7.5. Топологічний дефект мапи Кохонена

6. *Кількість нейронів у шарі.* Кількість нейронів у шарі Кохонена повинна відповідати кількості класів вхідних сигналів (векторів). Це може бути неприпустимим для тих задач, коли кількість класів наперед невідома.

7. *Класи вхідних сигналів.* Шар Кохонена може формувати тільки класи, які являють собою опуклі області вхідного простору.

7.4. Алгоритм функціонування сітки Кохонена

Ініціалізація сітки. Ваговим коефіцієнтам сітки надаються малі випадкові значення. Загальна кількість синаптичних ваг — $M \times N$ (див. рис. 7.2). Початкова зона сусідства наведена на рис. 7.4.

1. Пред'явлення сітці нового вхідного сигналу.

2. Обчислення відстані до всіх нейронів сітки:

відстані d_j від вхідного сигналу до кожного нейрона j визначаються за формулою:

$$d_j = \sum_{i=1}^N (x_i(t) \cdot w_{ij}(t))^2, \quad (7.6)$$

де x_i — i -ий елемент вхідного сигналу в момент часу t , $w_{ij}(t)$ — вага зв'язку від i -го елемента вхідного сигналу до нейрона j у момент часу t .

3. Вибір нейрона з найменшою відстанню: вибирається нейрон-переможець j^* , для якого відстань d_j найменша.

4. Налаштування ваг нейрона j^* і його сусідів: робиться налаштування ваг для нейрона j^* і всіх нейронів з його околу NE. Нові значення ваг:

$$w_{ij}(t+1) = w_{ij}(t) + r(t)(x_i(t) - w_{ij}(t)), \quad (7.7)$$

де $r(t)$ — швидкість навчання, що зменшується з часом (додатне число, менше одиниці).

5. Повернення до кроку 2.

В алгоритмі використовується *коефіцієнт швидкості навчання*, який поступово зменшується, для тонкішої корекції на новій епосі. В результаті позиція центру встановлюється в певній позиції, яка задовільним чином кластеризує приклади, для яких даний нейрон є переможцем.

Властивість топологічної впорядкованості досягається в алгоритмі за допомогою використання поняття околу. Окіл — це декілька нейронів, що оточують нейрон-переможець. Відповідно до швидкості навчання, розмір околу поступово зменшується, так, що спочатку до нього належить досить велике число нейронів (можливо вся

мапа), на самих останніх етапах окіл стає нульовим і складається лише з нейрона-переможця. В алгоритмі навчання корекція застосовується не тільки до нейрона-переможця, але і до всіх нейронів з його поточного околу. В результаті такої зміни околу, початкові доволі великі ділянки сітки мігрують в бік навчальних прикладів. Сітка формує грубу структуру топологічного порядку – *топологічну мапу*, при якій схожі приклади активують групи нейронів, що близько знаходяться на топологічній мапі. З кожною новою епохою швидкість навчання і розмір околу зменшуються, тим самим всередині ділянок мапи виявляються більш тонкі розходження, що зрештою призводить до точнішого налаштування кожного нейрона. Часто навчання зумисне розбивають на дві фази: більш коротку, з великою швидкістю навчання і великих околів, і більш тривалу з малою швидкістю навчання і нульовими або майже нульовими околами.

Після того, як сітка навчена розпізнаванню структури даних, її можна використовувати як засіб візуалізації при аналізі даних.

Недоліки. Сітка може бути використана для кластерного аналізу тільки в тому випадку, якщо заздалегідь відоме число кластерів.

Переваги. Сітка Кохонена здатна функціонувати в умовах перешкод через те, що число кластерів фіксоване, ваги модифікуються повільно, налаштування ваг закінчується після навчання.

Модифікації. Одна з модифікацій полягає в тому, що до сітки Кохонена додається сітка MAXNET, що визначає нейрон з найменшою відстанню до вхідного сигналу.

7.5. Квантування навчального вектора (Learning Vector-Quantization)

Сітка запропонована Тойво Кохоненом у середині 80-х рр., набагато пізніше за його початкову роботу по самоорганізованим мапам. Сітка базується на шарі Кохонена, який здатний до сортування прикладів у відповідні кластери і *використовується як для проблем класифікації, так і для кластеризації зображень* [152].

Сітка містить вхідний шар, самоорганізовану мапу Кохонена та вихідний шар. Приклад сітки зображений на рис. 7.6. Вихідний шар має стільки нейронів, скільки є відмінних категорій або класів. Мапа Кохонена має ряд нейронів, згрупованих для кожного з цих класів. Кількість елементів обробки на один клас залежить від складності відношення “вхід-вихід”. Звичайно, кожний клас буде мати однакову кількість елементів по всьому шару. Шар Кохонена навчається класифікації за допомогою навчальної множини. Сітка використо-

186

вує правила контрольованого навчання. Вхідний шар повинен містити лише один нейрон для кожного окремого вхідного параметра.

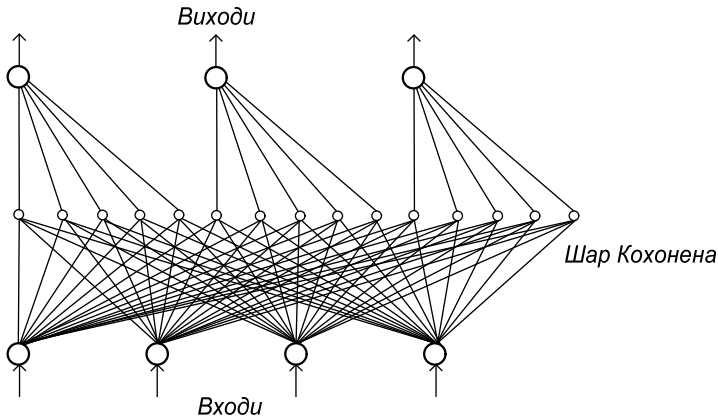


Рис. 7.6. Приклад сітки з квантуванням навчального вектора

Квантування навчального вектора класифікує вхідні дані у визначені групування, тобто відображає n -вимірний простір у m -вимірний простір (бере m входів і створює n виходів). Мапи зберігають відношення між близькими сусідами у навчальній множині так, що вхідні образи, які не були попередньо вивчені, будуть розподілені за категоріями їх найближчих сусідів у навчальних даних.

У режимі навчання, контрольована сітка використовує шар Кохонена, де обчислюється відстань від навчального вектора до кожного нейрону і найближчий нейрон оголошується переможцем. Існує лише один переможець на весь шар. Переможцю дозволено збуджувати лише один вихідний нейрон, оголошуючи клас або кластер до якого належить вхідний вектор. Якщо нейрон-переможець знаходиться у очікуваному класі навчального вектора, його ваги підсилюються у напрямку навчального вектора. Якщо нейрон-переможець не є у класі навчального вектора, ваги з'єднань зменшуються. Ця остання операція згадується як відштовхування (*repulsion*). Під час навчання окремі нейрони, що приписані до часткового класу мігрують до області, пов'язаної з їх специфічним класом.

Під час режиму функціонування, обчислюється відстань від вхідного вектора до кожного нейрону і знову найближчий нейрон оголошується переможцем. Це в свою чергу генерує один вихід, визначаючи частковий клас, знайдений сіткою.

Геометрична інтерпретація. Значне поширення отримали шари Кохонена, побудовані наступним чином: кожному (j -му) нейро-

ну співставляється точка $W_j = (w_{j1}, w_{j2}, \dots, w_{jm})$ в m -вимірному просторі (просторі сигналів) (на рис. 7.7 помічені в окремих багатокутниках).

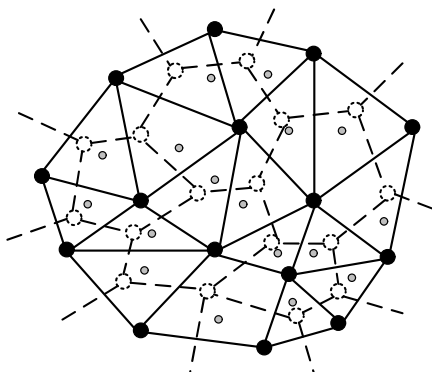


Рис. 7.7. Розділення площини на багатокутники Вороного-Діріхле для випадково вибраних точок (кожна точка помічена у своєму багатокутнику)

Для вхідного вектора $x = (x_1, x_2, \dots, x_m)$ обчислюються його евклідові віддалі $\rho_j(x)$ до точок W_j та “найближчий отримує все” – той нейрон, для якого ця віддаль мінімальна, видає одиницю, інші – нулі. Слід відзначити, що для порівняння віддалей досить обчислювати лінійну функцію сигналу:

$$\rho_j(x)^2 = \|x - W_j\|^2 = \|y\|^2 - 2 \sum_{i=1}^m w_{ji} x_i + \|x\|^2 \quad (7.8)$$

де $\|y\|$ – Евклідова довжина вектора: $\|y\|^2 = \sum_i y_i^2$. Останній доданок $\|x\|^2$ однаковий для всіх нейронів, тому для визначення найближчої точки він не потрібний.

Задача зводиться до пошуку номера найбільшого зі значень лінійних функцій:

$$j_{\max} = \arg \max_j \left\{ \sum_{i=1}^m w_{ji} x_i - \frac{1}{2} \|W_j\|^2 \right\}. \quad (7.9)$$

Таким чином, координати точки $W_j = (w_{j1}, w_{j2}, \dots, w_{jm})$ співпадають з вагами лінійного нейрона шару Кохонена (при цьому значення порогового коефіцієнта $w_{j0} = -\|W_j\|^2 / 2$).

Якщо задано точки $W_j = (w_{j1}, w_{j2}, \dots, w_{jm})$, то m -вимірний простір розбивається на відповідні багатогранники Вороного-Діріхле V_j : многогранник V_j складається з точок, які ближче до W_j , ніж до інших $W_k, (k \neq j)$ (див. рис. 7.7).

Твердження 7.1. *Діаграма Вороного скінченної множини точок S на площині уявляє таке розбиття площини, при якому кожна область цього розбиття утворює множину точок, більш близьких до одного з елементів множини S , ніж до будь-якого іншого елемента множини [45]. Названа на честь російського вченого українського походження Георгія Феодосійовича Вороного. Також відома як: мозаїка Вороного, розбиття Вороного, розбиття Діріхле.*

Функціонування сітки з квантуванням навчального вектора. Шар Кохонена складається з деякої кількості n паралельно працюючих лінійних елементів. Усі вони мають однакову кількість m входів і отримують на свої входи один і той же вектор вхідних сигналів $x = (x_1, \dots, x_m)$. На виході j -го лінійного елемента отримуємо сигнал

$$y_j = w_{j0} + \sum_{i=1}^m w_{ji} x_i, \quad (7.10)$$

де w_{ji} – ваговий коефіцієнт i -го входу j -го нейрона; w_{j0} – пороговий коефіцієнт.

Після проходження шару лінійних елементів сигнали надсилаються на опрацювання за правилом “переможець отримує все”: серед сигналів y_j на виході відшукується максимальний – його номер $j_{\max} = \arg \max_j \{y_j\}$. Остаточню на виході сигнал, пронумерований як j_{\max} , дорівнює одиниці, всі інші – нулю. Якщо ж максимум одночасно досягається для декількох j_{\max} , то або приймають всі відповідні сигнали дорівнюваними одиниці, або тільки перший у списку (за згодою).

Взагалі задача векторного квантування з k кодовими векторами W_j для заданої сукупності вхідних векторів S ставиться як задача мінімізації спотворювання при кодуванні, тобто при заміщенні кожного вектора з S відповідним кодовим вектором. Як вже вище наголошувалося, в базовому варіанті сіток Кохонена використовується метод найменших квадратів, і спотворення d обчислюється за формулою 7.3, яку можна переписати у вигляді

$$d = \sum_{j=1}^k \sum_{x \in K_j} \|x - W_j\|^2, \quad (7.11)$$

де K_j складається з тих точок $x \in S$, які є ближчими до W_j , аніж до інших W_l ($l \neq j$). Іншими словами, K_j складається з тих точок $x \in S$, які кодуються кодовим вектором W_j .

Якщо сукупність S задана і зберігається у пам'яті, то стандартним вибором у навчанні відповідної сітки Кохонена є метод К-середніх – метод розщеплювання:

- при даному виборі кодових векторів W_j (вони ж вагові вектори сітки) мінімізацією спотворення d находимо множини K_j – вони складаються з тих точок $x \in S$, які є ближчими до W_j , аніж до інших W_l ($l \neq j$);
- при даному розбитті S на множини K_j мінімізацією спотворення d находимо оптимальні позиції кодових векторів W_j – для оцінки за методом найменших квадратів це просто середні арифметичні:

$$W_j = \frac{1}{|K_j|} \sum_{x \in K_j} x, \quad (7.12)$$

де $|K_j|$ – кількість елементів в K_j .

Далі – інтегрування. Цей метод розщеплення збігається за скінчену кількість кроків і видає локальний мінімум спотворення d .

Якщо ж, наприклад, сукупність S заздалегідь не задана, або з якихось причин не зберігається у пам'яті, то широко використовується *on-line метод*. Вектори x вхідних сигналів опрацьовуються по одному, для кожного з них відшукується найближчий кодовий вектор $W_j(x)$ (саме той “переможець”, який “отримує все”). Після цього даний кодовий вектор переобчислюється за формулою

$$W_{j(x)}^{new} = W_{j(x)}^{old} (1 - \theta) + x\theta, \quad (7.13)$$

де $\theta \in (0, 1)$ – крок навчання. Інші кодові вектори на цьому кроці не змінюються.

Для забезпечення стабільності використовується *on-line метод із згасаючою швидкістю навчання*: якщо T – кількість кроків навчання, то вважають $\theta = \theta(T)$. Функцію $\theta(T) > 0$ вибирають так,

щоб $\theta(T) \rightarrow 0$ монотонно при $T \rightarrow \infty$ і щоб ряд $\sum_{T=1}^{\infty} (\theta T)$ розбігався.

Векторне квантування є набагато більш загальною операцією, аніж кластеризація, оскільки кластери повинні бути розділеними між собою, тоді як сукупності K_j для різних кодових векторів W_j не обов'язково уявляють собою роздільні кластери. З іншого боку, при наявності роздільних кластерів векторне квантування може відшукувати їх і по-різному кодувати.

Недоліки. Для складних проблем класифікації з подібними вхідними прикладами сітка вимагає великої мапи Кохонена з багатьма нейронами на клас. Це вибірково може бути подолано або вибором доцільних навчальних прикладів, або розширенням вхідного шару.

Сітка квантування навчального вектора страждає від дефекту, що деякі нейрони мають тенденцію до перемоги занадто часто, тобто налаштовують свої ваги дуже швидко, а інші постійно залишаються незадіяними. Це часто трапляється, коли їх ваги мають значення далеко від навчальних прикладів. Щоб пом'якшити цю проблему, нейрон, який перемагає занадто часто штрафується, тобто зменшуються ваги його зв'язків з кожним вхідним нейроном. Це зменшення ваг є пропорційним до різниці між частотою перемог нейрону та частотою перемог середнього нейрону.

Переваги. Алгоритм граничної корекції використовується для вдосконалення рішення навіть коли було знайдено відносно добре рішення. Алгоритм спроможний діяти, коли нейрон-переможець знаходиться у неправильному класі, а другий найкращий нейрон у правильному класі. Навчальний вектор повинен бути близько від середньої точки простору, що з'єднує ці два нейрони. Неправильний нейрон-переможець зміщується з навчального вектора, а нейрон з іншого місця посувається до навчального вектора. Ця процедура робить чіткішою межу між областями, де можлива невірна класифікація.

На початку навчання бажано відключити відштовхування. Нейрон-переможець пересувається до навчального вектора лише тоді, коли навчальний вектор та нейрон-переможець знаходяться в одному класі. Це право вибору доцільне, коли нейрон повинен оминати область, яка має відмінний клас для досягнення необхідної області.

7.6. Пружні мапи

Метод апроксимації багатовимірних даних, оснований на мінімізації “енергії пружної деформації” мапи, зануреної у простір даних, був запропонований у 1996 році А.Н. Горбанем [119], а далі розвинений їм спільно з А.Ю. Зинов'євим, А.А. Россієвим і А.А. Пітен-

ко [23]. Метод заснований на аналогії між головним різноманіттям та еластичною мембраною і пружною пластиною. У цьому сенсі він є розвитком класичної ідеї сплану (хоча пружні мапи і не є багатовимірними спланами).

Нехай задана сукупність вхідних векторів S . Так само, як і сітки векторного квантування і самоорганізуючі мапи, пружна мапа представлена як сукупність кодових векторів (вузлів) W_j у просторі сигналів. Множина даних S розділена на класи K_j , які складаються з тих точок $x \in S$, які є ближчими до W_j , аніж до інших $W_l (l \neq j)$. Спотворення кодування d з формули (7.9) може тлумачитися як сумарна енергія пружин одиничної жорсткості, зв'язуючих вектори даних з відповідними кодовими векторами.

На множині вузлів задана додаткова структура: деякі пари зв'язані “пружними зв'язками”, а деякі трійки об'єднані у “ребра жорсткості”. Позначимо множину пар, зв'язаних пружними зв'язками, через E , а множину трійок, складаючих ребра жорсткості, через G . Наприклад, у квадратних ґратах найближчі вузли (як по вертикалі, так і по горизонталі) зв'язуються пружними зв'язками, а ребра жорсткості утворюються вертикальними і горизонтальними трійками найближчих вузлів. Енергія деформування мапи складається з двох складових енергій: розтягання

$$U_E = \lambda \sum_{(W_i, W_j) \in E} \|W_i - W_j\|^2 \quad (7.14)$$

і згину

$$U_G = \mu \sum_{(W_i, W_j, W_l) \in G} \|W_i - 2W_j + W_l\|^2, \quad (7.15)$$

де λ, μ – відповідні модулі пружності.

Задача побудування пружної мапи полягає у мінімізації функціоналу

$$U = d + U_E + U_G. \quad (7.16)$$

Якщо розбиття сукупності вхідних векторів S на класи K_j фіксоване, то мінімізація U – лінійна задача з розрідженою матрицею коефіцієнтів. Тому, як і для НС векторного квантування, застосовується *метод розщеплення*: фіксуємо $\{W_j\}$ – шукаємо $\{K_j\}$ – для даних $\{K_j\}$ шукаємо $\{W_j\}$ – для даних $\{W_j\}$ шукаємо $\{K_j\}$ – Алгоритм збігається до (локального) мінімуму U .

Метод пружних мап дозволяє розв'язувати всі задачі, які розв'язують самоорганізуючі мапи Кохонена, проте, володіють більшою регулярністю і передбачливістю. При збільшенні модуля зги-

ну μ пружні мапи наближаються до лінійних головних компонент. При зменшенні обох модулів пружності вони перетворюються у Кохоненівські сітки векторного квантування. В теперішній час пружні мапи інтенсивно використовуються для аналізу багатовимірних даних у біоінформатиці.

Приклад 7.1. Розглянемо приклад розділення класів, складеного за допомогою розбиття Вороного-Діріхле. Розв'язується задача класифікації. Кількість класів може бути будь-якою. Викладемо алгоритм для двох класів: **A** і **B** (рис. 7.8).

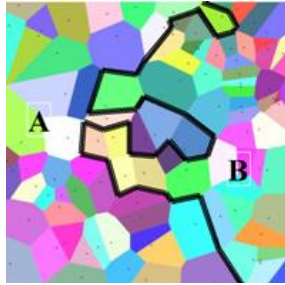


Рис. 7.8. Приклад можливого розділу класів, складеного за допомогою розбиття Вороного-Діріхле

Початково для навчання системи надходять дані, клас яких відомий. Задача: знайти для класу **A** деяку кількість k_A кодових векторів W_j^A , а для класу **B** деяку (можливо, іншу) кількість k_B кодових векторів W_l^B таким чином, щоб результуюча сітка Кохонена з $k_A + k_B$ кодовими векторами W_j^A, W_l^B (об'єднуємо обидва сімейства) здійснювала класифікацію за наступним вирішальним правилом:

Правило 7.1: якщо для вектора вхідних сигналів x найближчий кодовий вектор (“переможець”, який у шарі Кохонена “отримує все”) належить до сімейства $\{W_j^A\}$, то x належить класу **A**; якщо ж найближчий до x кодовий вектор належить до сімейства $\{W_l^B\}$, то x належатиме до класу **B**.

З кожним кодовим вектором об'єданого сімейства $\{W_j^A\} \cup \{W_l^B\}$ зв'язаний багатогранник Вороного-Діріхле. Позначимо ці багатогранники V_j^A, V_l^B відповідно. Клас **A** у просторі сигналів, відповідно до вирішального правила 7.1, відповідає об'єднанню $\cup_j V_j^A$, а клас **B** відповідає об'єднанню $\cup_l V_l^B$. Геометрія таких об'єднань багатогранників може бути досить складною.

Правило навчання сітки on-line будується на основі базового правила навчання сітки векторного квантування. Нехай на вхід сис-

теми надходить вектор сигналів x , клас якого відомий. Якщо він класифікується системою вірно, то відповідний до x кодовий вектор W трохи зсувається у бік вектора сигналу (“заохочення”):

$$W^{new} = W^{old} (1 - \theta) + x\theta.$$

Якщо ж x класифікується невірно, то відповідний до x кодовий вектор W трохи зсувається у протилежний бік від сигналу (“покарання”):

$$W^{new} = W^{old} (1 + \theta) - x\theta,$$

де $\theta \in (0, 1)$ – крок навчання. Для забезпечення стабільності використовується on-line метод зі згасаючою швидкістю навчання. Можливе також використання різних кроків для “заохочення” вірного розв’язку і для “покарання” невірного.

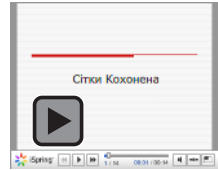
Це – найпростіша (базова) версія методу. Існує і багато інших модифікацій.

Контрольні завдання

1. Навести основні властивості нейронної сітки Кохонена.
2. Пояснити принцип функціонування нейронної сітки Кохонена.
3. Розкрити зміст поняття “Самоорганізована мапа Кохонена”.
4. Охарактеризувати особливості ітераційного процесу конкурентного навчання нейронної сітки Кохонена (конкуренція-об’єднання-підналагоджування ваг).
5. Навести умови, які лежать в основі функції f_{ij} віддаленості i -го та j -го нейронів від нейрона-переможця в мапі Кохонена.
6. Розкрити особливості побудування мапи Кохонена.
7. Прокоментувати проблеми, пов’язані з реалізацією сітки Кохонена: вибору коефіцієнтів навчання і параметрів віддалі від нейрона-переможця, рандомізації ваг тощо.
8. Навести послідовність кроків алгоритму функціонування сітки Кохонена.
9. Розкрити структурні особливості і принцип функціонування сітки з квантуванням навчального вектора.
10. Пояснити механізм розділення площини на багатокутники Вороного-Діріхле для випадково вибраних точок у просторі. Навести визначення діаграми Вороного.
11. Розкрити метод апроксимації багатовимірних даних, оснований на мінімізації “енергії пружної деформації”, і навести правило навчання on-line пружної мапи.

Презентації

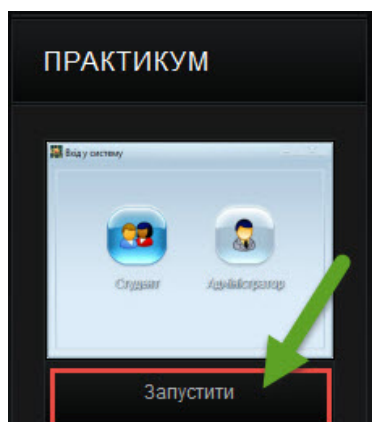
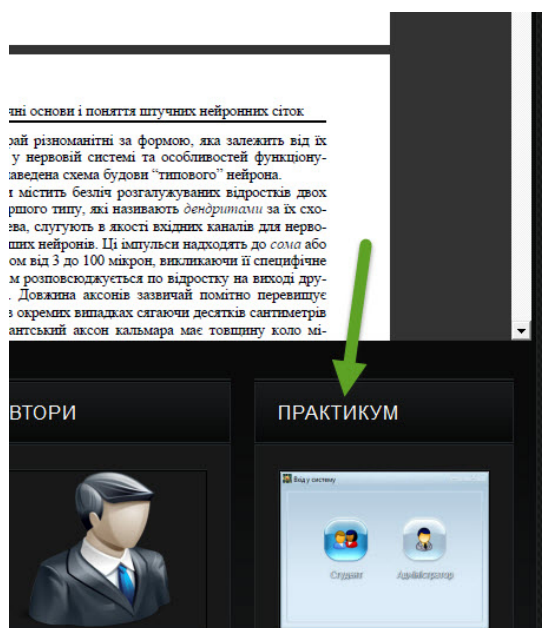
1. Сітки Кохонена



Практичні заняття

Нейронні сітки Кохонена

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 8

Сітки зустрічного поширення

*Коли людина з грошима зустрічає людину з досвідом,
людина з досвідом уходить з грошима,
а людина, у якої були гроші, уходить без грошей, але з досвідом*
Життєва мудрість

Зустрілися підлеглий та його начальник.

Підлеглий – начальнику:

– У мене для Вас дві новини, погана і добра.

– Починай з поганої.

– Мене звільнили.

– А добра?

– И Вас теж

Анекдот

8.1. Загальні відомості

Уведення до сіток зустрічного поширення. Метод навчання ШНС, отримавший назву “зустрічне поширення” (*Counter Propagation*) і запропонований і розвинений Робертом Хехт-Нільсеном [132; 133], уявляє собою сполучення методів навчання двох різних ШНС: мапи Кохонена, що самоорганізується, і зірки Гроссберга (*Grossberg Outstar*), вдало об’єднавши в одній архітектурі переваги можливості узагальнення сітки Т. Кохонена [153] і простоту навчання вихідної зірки Гроссберга [125]. Вона належить до класу сіток, що навчаються без учителя. Їх поєднання веде до властивостей, яких немає у жодної з них відокремлено. Сітка зустрічного поширення працює з двійковими векторами, що складаються з нулів і одиниць, або безперервними. У результаті навчання вхідні вектори асоціюються з вихідними і, коли сітка навчена, подача вхідних образів приводить до одержання вихідних. Правильний вихід може бути отриманий і тоді, коли вхід є неповним, трохи невірним. Гіперповерхня, одержувана в результаті функціонування сітки ЗП, унаслідок принципу безперервності, дає можливість здійснювати прогнозування. Природно, що усередині гіперпаралелепіпеда навчальних образів прогнозування буде більш точним, а при рішенні задачі екстраполяції помилка буде значно більше.

Отже, запропонована Нехт-Нільсеном *сітка зустрічного поширення* (СЗП) поєднує неконтрольований шар Кохонена із контро-

льованим шаром на виході сітки і призначена для розв'язання складних класифікацій при мінімізації кількості нейронів та часу навчання. Навчання для СЗП подібне до сітки з квантуванням навчального вектора.

Існує два різновиду варіантів сіток, які одержали назву СЗП:

- двошарова сітка прямого поширення, яка складається з шару Кохонена і шару Гроссберга;
- двошарова сітка прямого поширення, яка складається з самоорганізуючої мапи Кохонена і шару Гроссберга.

Відмінність обох варіантів полягає в організації схованого шару, і, відповідно, в способі навчання нейронів: в першому варіанті зв'язки між сусідніми нейронами шару Кохонена відсутні; в другому – такі зв'язки є, і через це нейрони схованого шару навчаються як самоорганізуюча мапа Кохонена.

Можливості СЗП перевершують можливості одношарових сіток, причому час навчання у порівнянні з методом зворотного поширення може зменшитися у сто разів. Зустрічне поширення не настільки загальновживане, як зворотне поширення, проте може давати розв'язки у тих застосуваннях, де тривала навчаюча процедура неможлива. Крім того, окрім подолання обмежень однорідних нейросіток, зустрічному поширенню притаманні інші власні корисні властивості.

Методи, які подібно до зустрічного поширення об'єднують відмінні сіткові парадигми як будівельні блоки, можуть привести до моделей сіток, більш близькими до мозку за архітектурою, аніж будь-які інші однорідні структури. Схоже на те, що саме каскадні сполучення модулів різної спеціалізації у мозку виконують необхідні обчислення.

Модель СЗП за функціонуванням схожа на стіл довідок, здібний до навчання. В процесі навчання вхідні вектори асоціюються з відповідними вихідними векторами. Ці вектори можуть бути двійковими (складаються з нулів та одиниць) або неперервними. Коли сітка навчена, прикладання вхідного вектора приводить до бажаного вихідного вектора. Узагальнююча властивість сітки дозволяє отримати вірний вихід навіть при прикладанні вхідного вектора, який може бути трохи невірним або спотвореним. Це дозволяє використовувати дану парадигму сітки для розпізнавання образів, відтворення та посилення сигналів.

Структура сітки. На рис. 8.1 показана спрощена версія прямої дії СЗП та ілюструються функціональні властивості цієї пара-

дигми. Повна двоспрямована сітка заснована на тих самих принципах.

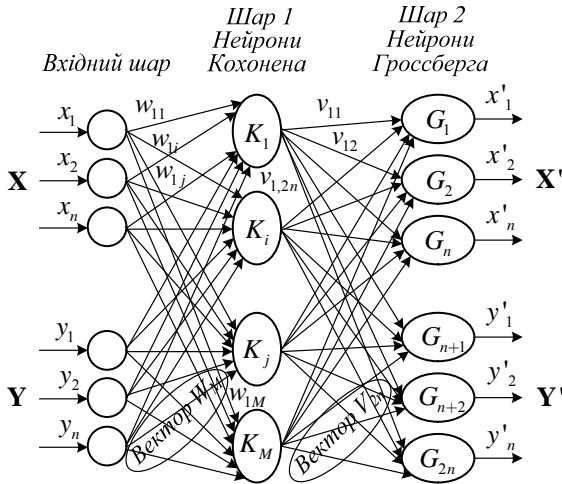


Рис. 8.1. Сітка із зустрічним розпізнаванням без зворотних зв'язків

Нейрони вхідного шару слугують лише точками розгалуження і не виконують обчислень. Кожний нейрон вхідного шару з'єднаний з кожним нейроном шару 1 (так званий шар Кохонена) окремою вагою w_{mj} . Ці ваги у цілому розглядаються як матриця ваг \mathbf{W} . Аналогічно, кожний нейрон у шарі Кохонена (шар 1) з'єднаний з кожним нейроном у шарі Гроссберга (шар 2) вагою v_{np} . Ці ваги утворюють матрицю ваг \mathbf{V} . Все це дуже нагадує сітки інших топологій, проте, відмінність полягає в операціях, виконуваних нейронами Кохонена і Гроссберга.

Як і багато інших видів нейросіток, зустрічне поширення функціонує в двох режимах: *навчання*, за яким подається вхідний вектор і ваги корегуються, щоб дати бажаний вихідний вектор; *нормальному*, за яким приймається вхідний вектор \mathbf{X} і видається вихідний вектор \mathbf{Y} .

8.2. Нормальне функціонування сітки зустрічного поширення

Шар Кохонена. У своїй найпростішій формі шар Кохонена функціонує в режимі “переможець забирає все”, тобто для даного

вхідного вектора один і тільки один нейрон Кохонена видає на виході логічну одиницю, всі інші видають нуль. Нейрони Кохонена можна сприймати як набір реле, то ж для будь-якого вхідного вектора спрацьовує одне з них.

Асоційована з кожним нейроном Кохонена множина ваг з'єднує його з кожним входом. Наприклад, на рис. 8.1 нейрон Кохонена K_1 має ваги $w_{11}, w_{21}, \dots, w_{m1}$, які утворюють ваговий вектор \mathbf{W}_1 . Вони з'єднуються через вхідний шар з вхідними сигналами x_1, x_2, \dots, x_m , що складають вхідний \mathbf{X} . Подібно до нейронів більшості видів нейросіток вихід NET кожного нейрона Кохонена є просто сумою зважених входів. Це можна виразити наступною залежністю:

$$NET_j = w_{1j} x_1 + w_{2j} x_2 + \dots + w_{mj} x_m, \quad (8.1)$$

де NET_j – це вихід NET j -го нейрона Кохонена

$$NET_j = \sum_i x_i w_{ij} \quad (8.2)$$

або у векторній формі

$$\mathbf{N} = \mathbf{XW}, \quad (8.3)$$

де \mathbf{N} – вектор виходів NET шару Кохонена.

Нейрон Кохонена з максимальним значенням NET є “переможцем”: його вихід дорівнює одиниці, у решти він дорівнює нулю.

Шар Гроссберга. Шар Гроссберга функціонує у схожій манері. Його вихід NET є зваженою сумою виходів k_1, k_2, \dots, k_n шару Кохонена, утворюючих вектор \mathbf{K} . Вектор з'єднуючих ваг, позначений через \mathbf{V} , складається з ваг $v_{11}, v_{21}, \dots, v_{np}$. Тоді вихід NET кожного нейрона Гроссберга є

$$NET_j = \sum_i k_i w_{ij}, \quad (8.4)$$

де NET_j – вихід j -го нейрона Гроссберга, або у векторній формі

$$\mathbf{Y} = \mathbf{KV}, \quad (8.5)$$

де \mathbf{Y} – вихідний вектор шару Гроссберга, \mathbf{K} – вихідний вектор шару Кохонена, \mathbf{V} – матриця ваг шару Гроссберга.

Якщо шар Кохонена функціонуватиме таким чином, що лише у одного нейрона величина NET дорівнюватиме одиниці, а у решти дорівнюватиме нулю, то лише один елемент вектора \mathbf{K} буде відмінним від нуля, і обчислення будуть дуже простими. Фактично кожний нейрон шару Гроссберга лише видає величину ваги, яка зв'яже цей нейрон з єдиним ненульовим нейроном Кохонена.

8.3. Навчання сітки зустрічного поширення

8.3.1. Навчання шару Кохонена

Шар Кохонена класифікує вхідні вектори по групах схожості. Це досягається за допомогою такого піднастроювання ваг шару Кохонена, що близькі вхідні вектори активують один і той же нейрон даного шару. Далі задачею шару Гроссберга стає отримання бажаних виходів.

Навчання Кохонена відноситься до *самонавчання*, яке відбувається *без вчителя*. Тому важко (і не треба) завбачати, який саме нейрон Кохонена буде активізуватися для заданого вхідного вектора. Необхідно лише гарантувати, щоб у результаті навчання несхожі вхідні вектори розділялися.

Попереднє опрацювання вхідних векторів. Дуже бажано (хоча і не обов'язково) *нормалізувати вхідні вектори* перед тим, як пред'являти їх сітці. Це виконується за допомогою ділення кожної компоненти вхідного вектора на його довжину. Ця довжина визначається витяганням квадратного кореня з суми квадратів компонент вектора, що в алгебраїчній формі має вигляд:

$$x'_i = \frac{x_i}{\sqrt{x_1^2 + x_2^2 + \dots + x_n^2}}. \quad (8.6)$$

Таким чином вхідний вектор перетворюється в одиничний вектор з тим самим напрямком, тобто вектор одиничної довжини у n -вимірному просторі.

Рівняння (8.6) узагальнює добре відомий випадок двох вимірювань, коли довжина вектора дорівнює гіпотенузі прямокутного трикутника, утвореного його x і y компонентами, як це впливає з відомої теореми Піфагора. На рис. 8.2, *а* такий двовимірний вектор \mathbf{V} поданий у координатах $x - y$, причому координата x дорівнює чотирьом, а координата y — трьом. Квадратний корінь з суми квадратів цих компонент дорівнює п'яти. Ділення кожної компоненти вектора \mathbf{V} на п'ять дає вектор \mathbf{V}' з компонентами $4/5$ і $3/5$, де напрямок \mathbf{V}' той самий, що й вектора \mathbf{V} , але має одиничну довжину.

На рис. 8.2, *б* зображено декілька одиничних векторів, які закінчуються у точках одиничного кола (кола одиничного радіусу), що має місце, коли у сітки лише два виходи. У випадку трьох входів вектори подавалися б стрілками, які закінчуються на поверхні одиничної сфери. Ці ж уявлення можна перенести на сітки, які мають довільну кількість входів, де кожний вектор подається стрілкою, яка за-

кінчується на поверхні одиничної гіперсфери (корисною абстракцією, яка, проте, не припускає безпосередньої візуалізації).

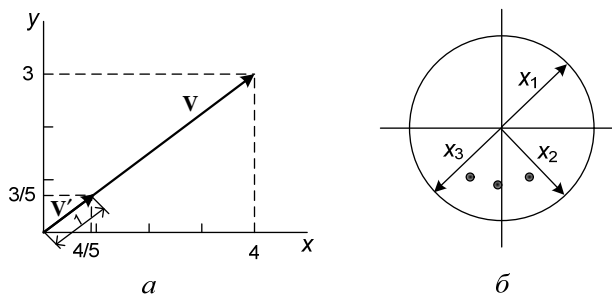


Рис. 8.2. Одиничний вхідний вектор (а) і двовимірні одиничні вектори на одиничному колі

При навчанні шару Кохонена на вхід подається нормалізований вхідний вектор. На виході нейрона з максимальним значенням зваженої суми формується рівень логічної «1». Нейрон з максимальним значенням скалярного добутку оголошується “переможцем”, при цьому процес навчання після вибору нейрона-“переможця” з ваговим вектором, найближчим до вхідного вектора, полягає у подальшому піднастроюванні (наближенні) компонентів вагового вектора обраного нейрона до пред’явленого вхідного вектора. Сітка самоорганізується таким чином, що даний нейрон Кохонена має максимальний вихід для даного вхідного вектора. Рівняння, яким описується процес навчання, має наступний вигляд:

$$\mathbf{W}_j(t+1) = \mathbf{W}_j(t) + \eta(x^k - \mathbf{W}_j(t)), \quad (8.7)$$

де $\mathbf{W}_j(t+1)$, $\mathbf{W}_j(t)$ – відповідно, нове та попереднє значення вектора ваг j -го нейрона-“переможця” для пред’явленого вхідного вектора \mathbf{X}_k , $k=1, \dots, N$; η – коефіцієнт швидкості навчання.

Кожний коефіцієнт з вагового вектора нейрона-“переможця” змінюється пропорційно різниці між його величиною та величиною входу, з яким він з’єднаний. Напрямок змінювання мінімізує різницю між ваговим коефіцієнтом і входом. З ходом навчання коефіцієнт η поступово зменшується.

На рис. 8.3 цей процес проілюстрований геометрично у двовимірному вигляді. Спочатку відшукується вектор $\mathbf{X}-\mathbf{W}_c$, для цього проводиться відрізок з кінця \mathbf{W} у кінець \mathbf{X} . Далі цей вектор укорочується множенням його на скалярну величину η , меншу за одиницю, в

результаті чого отримується вектор змінювання δ . Остаточно новий ваговий вектор \mathbf{W}_n є відрізком, спрямованим з початку координат у кінець вектора δ . Отже можна бачити, що ефект навчання полягає у обертанні вагового вектора у напрямку вхідного вектора без суттєвого змінювання його довжини.

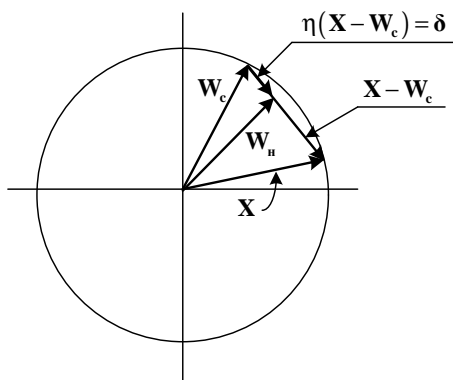


Рис. 8.3. Обертання вагового вектора у процесі навчання
(\mathbf{W}_n – вектор нових вагових коефіцієнтів; \mathbf{W}_c – вектор
старих вагових коефіцієнтів)

Змінна η є коефіцієнтом швидкості навчання, який спочатку, як правило, приймається $\sim 0,7$ і далі поступово зменшується у процесі навчання. Це дозволяє робити великі початкові кроки для швидкого грубого навчання і менші кроки при підході до остаточної величини.

Якби з кожним нейроном Кохонена асоціювався один вхідний вектор, то шар Кохонена міг би навчитися за допомогою одного обчислення на вагу. Ваги нейрона-“переможця” прирівнювалися б до компонент навчаючого вектора ($\eta=1$). Проте, як правило, навчаюча множина охоплює багато подібних один до одного вхідних векторів, і сітка повинна бути навчена активувати один і той же нейрон Кохонена для кожного з них. В цьому випадку ваги цього нейрона мають формуватися усередненням вхідних векторів, які повинні його активувати. Поступове зниження величини η зменшує дію кожного навчаючого кроку, через що остаточне значення буде середньою величиною від вхідних векторів, на яких відбувається навчання. Таким чином, ваги, асоційовані з нейроном, приймуть значення поблизу “центра” вхідних векторів, для яких даний нейрон є “переможцем”.

В результаті навчання нейрон-“переможець” буде активізуватися для сукупності асоційованих з ним вхідних векторів, середня величина яких співпадає з вектором ваг цього нейрона.

Проте існує ряд проблем навчання шару Кохонена, від розв'язання яких залежить ефективність використання нейронної СЗП у цілому.

Перш за все, через те, що нормалізовані вхідні вектори, як правило, нерівномірно розподілені по поверхні гіперсфери, більшість вагових векторів (первісно рівномірно розподілених рандомізацією ваг) будуть значно віддалені від будь-якого вхідного вектора. Тому на виходах відповідних до них нейронів постійно буде установлений рівень логічного «0», і ці нейрони виявляться некорисними. Через це активних нейронів може виявитися недостатнім для ефективного розділення близькорозташованих вхідних векторів.

Ще одна проблема полягає у складності розділення на відмінні класи множин схожих вхідних векторів у випадку, якщо первісна щільність вагових векторів в околі навчаючих векторів буде недостатньою.

Нарешті проблему може представляти також надмірно висока щільність вагових векторів поблизу несуттєво розрізнених вхідних векторів, що може привести до активізації декількох нейронів шару Кохонена, тобто до формування хибних класів вхідних векторів.

Вибір початкових значень вагових векторів. Усім вагам сітки перед початком навчання необхідно задати первісні значення. Загальноприйнята практика при роботі з нейронними сітками передбачає присвоювання вагам невеликих випадкових значень. *При навчанні шару Кохонена випадково вибрані вагові вектори необхідно нормалізувати.* Остаточні значення вагових векторів після навчання співпадають з нормалізованими вхідними векторами. Ось чому нормалізація перед початком навчання наближає вагові вектори до їх остаточних значень, скорочуючи, таким чином, навчаючий процес.

Рандомізація ваг шару Кохонена може породжувати й серйозні проблеми при навчанні через те, що у результаті її вагові вектори розподіляються рівномірно по поверхні гіперсфери. Але оскільки вхідні вектори, як правило, розподілені нерівномірно і мають тенденцію до групування на відносно невеликій частині поверхні гіперсфери, більшість вагових векторів будуть так віддалені від будь-якого вхідного вектора, що вони ніколи не дадуть найкращої відпові-

дності. Отже ці нейрони Кохонена будуть завжди мати нульовий вихід і виявляться некорисними. Більше того, ваг, що залишилися і дають найкращі відповідності, може виявитися занадто мало, щоб розділити вхідні вектори на розташовані на поверхні гіперсфери близько один до одного класи.

Припустимо, що є декілька множин вхідних векторів, причому усі множини схожі, але повинні бути розділені на різні класи. Сітка повинна бути навчена активізувати окремий нейрон Кохонена для кожного класу. Якщо початкова щільність вагових векторів в околі навчаючих векторів занадто мала, то може виявитися неможливим розділення схожих класів через недостатню кількість вагових векторів у досліджуваному околі для того, щоб приписати по одному з них до кожного класу вхідних векторів.

Навпаки, якщо декілька вхідних векторів отримано незначними змінюваннями з одного й того ж зразка і повинні бути об'єднані в один клас, то вони включатимуть один і той самий нейрон Кохонена. Якщо ж щільність вагових векторів дуже висока поблизу групи ледь відмінних вхідних векторів, то кожний вхідний вектор може активувати окремий нейрон Кохонена. Це не представляє катастрофи, бо шар Гроссберга може відобразити відмінні нейрони Кохонена в один і той самий вихід, проте це марнотратне використання нейронів Кохонена.

Найбільш бажаним розв'язком є розподілення вагових векторів у відповідності до щільності вхідних векторів, які необхідно розділити, розміщуючи тим самим більшу кількість вагових векторів в околі більшої кількості вхідних векторів. На практиці це є нездійсненим, але існує декілька методів наближеного досягнення тієї ж мети.

Одне з таких рішень, відоме під назвою *метода опуклого комбінування* (МОК) (convex combination method), полягає у тому, що усі ваги порівнюються до однієї й тієї ж самої величини:

$$w_i = \frac{1}{\sqrt{n}},$$

де n – кількість входів, а отже, кількість компонент кожного вагового вектора. Завдяки цьому усі вагові вектори співпадають і мають одиничну довжину. Кожні же компоненти входу \mathbf{X} надається значення:

$$x_i = \eta x_i + \frac{1-\eta}{\sqrt{n}},$$

де n – кількість входів. Спочатку коефіцієнт швидкості навчання η

дуже малий, внаслідок чого усі вхідні вектори мають довжину, близьку до $1/\sqrt{n}$, і майже співпадають з векторами ваг. У процесі навчання сітки η поступово зростає, наближуючись до одиниці. Це дозволяє розділяти вхідні вектори і остаточно приписує їм їх істинні значення. Вагові вектори відслідковують один або невелику групу вхідних векторів і наприкінці навчання дають бажану картину виходів. Додамо, що МОК добре працює, але уповільнює процес навчання через те, що вагові вектори піднастроюються до змінюваної мети.

Другий підхід полягає у додаванні шуму до вхідних векторів. Тим самим вони зазнають випадкових змінювань, охоплюючи кінець з кінцем ваговий вектор. Цей метод також є працездатним, але ще більш уповільнений, ніж метод опуклого комбінування.

Третій метод розпочинає з випадкових ваг, але на початковій стадії навчального процесу піднастроює усі ваги, а не тільки зв'язані з вигравшим нейроном Кохонена. Тим самим вагові вектори зсуваються ближче до області вхідних векторів. У процесі навчання корекція ваг розпочинається лише для найближчих до переможця нейронів Кохонена. Цей радіус корекції поступово зменшується, і в решті решт коректуються тільки ваги, пов'язані з вигравшим нейроном Кохонена.

Ще один метод наділяє кожний нейрон Кохонена “Почуттям справедливості”. Якщо цей нейрон стає переможцем частіше за свою законну частку часу (приблизно $1/k$, k – кількість нейронів Кохонена), він тимчасово збільшує свій поріг, що зменшує його шанси на виграш, надаючи тим самим можливість навчатися й іншим нейронам.

У багатьох застосуваннях точність результату суттєво залежить від розподілення ваг. На жаль, ефективність всіляких рішень вичерпно не оцінена і залишається проблемою.

Режим інтерполяції. Вище обговорювався алгоритм навчання, в якому для кожного вхідного вектора активувався лише один нейрон Кохонена – це так званий *метод акредитації*. Його точність обмежена тією обставиною, що вихід повністю визначається функцією лише одного нейрона Кохонена.

У *методі інтерполяції* ціла група нейронів Кохонена з найбільшими виходами може передавати свої вихідні сигнали у шар Гроссберга. Кількість нейронів у такій групі повинна вибиратися в залежності від задачі, і переконливих даних щодо оптимального розміру

групи немає. Як тільки групу визначено, множина її NET розглядається як вектор, довжина якого нормалізується на одиницю діленням кожного значення NET на корінь квадратний з суми квадратів значень NET у групі. Всі нейрони поза групою мають нульові виходи.

Метод інтерполяції здатний встановлювати більш складні відповідності і може давати більш точні результати. Проте знов таки немає переконливих даних, дозволяючих порівнювати режими інтерполяції та акредитації.

Статистичні властивості навченої сітки. Метод навчання Кохонена характеризується корисною і цікавою здібністю – здобувати статистичні властивості з множини вхідних даних. Кохоненом доведено, що для повністю навченої сітки імовірність найближчого розташування випадково обраного вхідного вектора (у відповідності з функцією щільності імовірності вхідної множини) до будь-якого заданого вагового вектора дорівнюватиме $1/k$, де k – кількість нейронів Кохонена. Це й обумовлюватиме оптимальне розподілення ваг на гіперсфері (припускається, що використовуються усі вагові вектори, а це має місце лише у тому випадку, якщо використовується один з вищеобговорюваних методів розподілення ваг).

8.3.2. Навчання шару Гроссберга

Призначенням нейронів шару Гроссберга є формування необхідних вихідних векторів після того, як нейрони шару Кохонена розділили вхідні вектори на класи. Фактично кожний нейрон шару Гроссберга лише видає значення ваги, яка зв'язує цей нейрон з нейроном-“переможцем” шару Кохонена.

На відміну від самонавчаючого шару Кохонена, шар Гроссберга навчається з *вчителем*. Відмінність від стандартної навчаючої процедури полягає у тому, що піднастроюванню підлягають тільки ті ваги нейронів шару Гроссберга, які з'єднані з ненульовим нейроном Кохонена. Використовується наступне правило:

$$\mathbf{V}_{ij}(t+1) = \mathbf{V}_{ij}(t) + \eta(\mathbf{Y}_j - \mathbf{V}_{ij}(t))K_j, \quad (8.8)$$

де K_j – вихід j -го нейрона Кохонена; y_j – j -й компонент необхідного вихідного вектора. Первісно η дорівнює 0,1 і зменшується у процесі навчання.

З (8.8) випливає, що ваги шару Гроссберга будуть збігатися до середніх величин від бажаних значень виходів, тоді як ваги шару Кохонена навчаються на середніх значеннях входів. Навчання шару

Гроссберга – це *навчання з вчителем*, алгоритм містить бажаний вихід, по якому він навчається. Шар Кохонена, який навчається без вчителя і самоорганізується, дає виходи у недетермінованих позиціях. Вони відображаються у бажані виходи шару Гроссберга.

Відмінність нейронної СЗП від інших полягає і в особливостях функціонування. У відповідності до наведеної на рис. 8.1 структури на вхід сітки подаються нормалізовані одиничні вектори \mathbf{X} і \mathbf{Y} , а на виході формуються їх нормалізовані апроксимації \mathbf{X}' і \mathbf{Y}' .

При навчанні вектори \mathbf{X} і \mathbf{Y} подаються одночасно і як вхідні, і як необхідні вихідні. При цьому вектор \mathbf{X} є вхідним для вектора \mathbf{X}' , а вектор \mathbf{Y} – для вектора \mathbf{Y}' . В результаті такого навчання здобувається однозначне відображення векторів \mathbf{X} і \mathbf{Y} на їх копії.

Після навчання в нейронній сітці зустрічного поширення реалізується властивість асоціативної пам'яті, полягаюча в тому, що пред'явлення на вхід тільки вектора \mathbf{X} (або \mathbf{Y}) при відсутності іншого приводить до породження на виході як вектора \mathbf{X}' , так і \mathbf{Y}' .

8.4. Режим нормального функціонування сітки зустрічного поширення

На рис. 8.1 наведена нейронна СЗП у цілому. В режимі нормального функціонування пред'являються вхідні вектори \mathbf{X} та \mathbf{Y} , і навчена сітка видає на виході вектори \mathbf{X}' та \mathbf{Y}' як апроксимації відповідно для \mathbf{X} та \mathbf{Y} . Вектори \mathbf{X} та \mathbf{Y} також будуть мати тенденцію до нормалізованості. У процесі навчання вектори \mathbf{X} та \mathbf{Y} подаються одночасно і як вхідні вектори сітки, і як бажані вихідні сигнали. Вектор \mathbf{X} використовується для навчання виходів \mathbf{X}' , а вектор \mathbf{Y} – для навчання виходів \mathbf{Y}' шару Гроссберга. У цілому СЗП навчається із застосуванням того ж методу, який вище описувався для сітки прямої дії. Нейрони Кохонена приймають вхідні сигнали як від векторів \mathbf{X} , так і від векторів \mathbf{Y} . Проте це відрізняється від ситуації, коли є один великий вектор, складений з векторів \mathbf{X} и \mathbf{Y} , і не впливає на алгоритм навчання.

Як результуюче отримується одиничне відображення, за яким при пред'явленні пари вхідних векторів породжується їх копія на виході. Це було б не викликало зацікавленості, якби не зазначити, що пред'явлення тільки вектора \mathbf{X} (з вектором \mathbf{Y} , дорівнюваним нулю) породжує як виходи \mathbf{X}' , так і виходи \mathbf{Y}' . Якщо F – функція, відображення \mathbf{X} у \mathbf{Y}' , то сітка апроксимує її. Також, якщо F оборотна, то

пред'явлення тільки вектора \mathbf{Y} (прирівнюючи \mathbf{X} нулю) породжує \mathbf{X}' . Унікальна властивість породжувати функцію и зворотну до неї робить СЗП корисною у ряді застосувань.

На рис. 8.1 не демонструється протитечія у сітці, по якій вона отримала свою назву. Така форма подання обрана тому, що вона також ілюструє сітку без обернених зв'язків і дозволяє узагальнити загаль-новживані поняття.

8.5. Деякі практичні застосування

Стиснення даних. Як доповнення до звичайних функцій відображення векторів зустрічне поширення виявляється корисним і у деяких менш очевидних застосовних областях. Одним з найбільш цікавих прикладів є *стиснення даних*.

Дійсно, СЗП можна використовувати для стиснення даних перед їх передачею, зменшуючи тим самим кількість бітів, які треба передавати. Припустимо, що необхідно передати деяке зображення \mathbf{S} . Його можна розбити на підзображення \mathbf{S}_{ij} , як це показано на рис. 8.4.

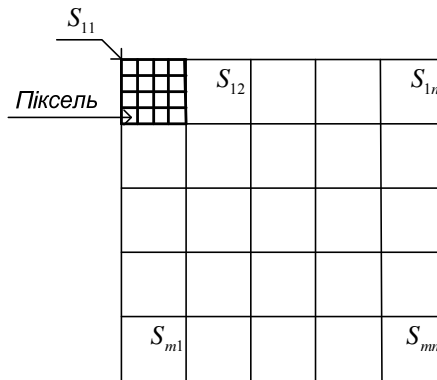


Рис. 8.4. Система стиснення зображень

В свою чергу, кожне підзображення \mathbf{S}_{ij} розбито на пікселі (найдрібніші елементи зображення). Тоді кожне підзображення \mathbf{S}_{ij} виявляється вектором, елементами якого є пікселі, з яких складається підзображення. Припустимо для спрощення, що кожний піксель – це одиниця (освітлення) або нуль (затемнення). Якщо підзображення \mathbf{S}_{ij} містить n пікселів, то для його передачі будуть необхідними n біт. Якщо припустити деякі спотворення, то для передачі типового зображення необхідна суттєво менша кількість бітів, що дозволяє пе-

редавати зображення швидше. Це стає можливим завдяки статистичного розподілення векторів підзображень. Деякі з них будуть зустрічатися частіше, тоді як інші – так рідко, що їх можна грубо апроксимувати. Метод *векторного квантування*, який при цьому використовується, знаходить більш короткі послідовності бітів, які найкращим чином представляють ці підзображення.

Векторне квантування. Сітка зустрічного поширення може бути використаною для виконання векторного квантування. Множина векторів підзображень застосовуються як навчаюча для навчання шару Кохонена за методом акредитації, коли вихід лише одного нейрона дорівнює 1. Ваги шару Гроссберга вчаться видавати бінарний код номера того нейрона Кохонена, вихід якого саме дорівнює 1. Наприклад, якщо вихідний сигнал нейрона 15 дорівнює 1 (а всі інші дорівнюють 0), то шар Гроссберга буде вчитися видавати 00...001111 (двійковий код числа 15). Це й буде найкоротшою бітовою послідовністю символів, які передаються.

На приймальному кінці ідентичним чином навчена СЗП приймає двійковий код і реалізує зворотну функцію, апроксимуючу первісне підзображення.

8.6. Загальні недоліки і переваги сіток зустрічного поширення

Недоліки. Сітка не дає можливості будувати точні апроксимації (точні відображення). Сам винахідник СЗП Роберт Хехт-Нільсон усвідомлював її обмеження наступним чином: “Звісно, СЗП поступається зворотному поширенню у більшості застосувань, пов’язаних з сітковими відображеннями. Проте її переваги полягають у простоті і утворенні задовільної статистичної моделі для свого середовища вхідних векторів”.

Крім того, до недоліків моделі слід віднести слабкий теоретичний базис модифікацій сітці зустрічного поширення.

Переваги:

- сітка зустрічного поширення *проста*. Вона надає можливість втягати статистичні властивості з множини вхідних сигналів. Кохонен довів, що для навченої сітки ймовірність того, що випадково обраний вхідний вектор буде найближчим до будь-якого заданого вагового вектора, дорівнює $1/k$, де k – кількість нейронів Кохонена;

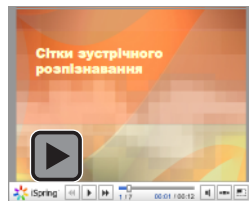
- сітка *швидко навчається*, і при правильному використанні вона: наприклад, у порівнянні зі зворотним поширенням час навчання може бути скорочений у 100 разів. То ж вона є корисною також для *швидкого моделювання систем*, де висока точність зворотного поширення примушує віддати йому перевагу в остаточному варіанті, проте необхідна *швидка початкова апроксимація*;
- спроможність у СЗП *породжувати функцію й обернену до неї* знайшло застосування у ряді систем;
- у своїх можливостях *будувати відображення* СЗП значно перевищує одношарові перцептрони.

Контрольні завдання

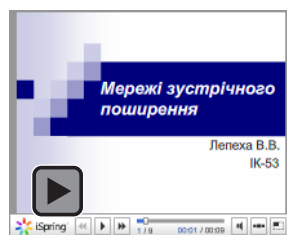
1. Навести природу методу навчання “зустрічного поширення” і розкрити роль складових “будівельних блоків” в однойменній нейронній сітці.
2. Охарактеризувати структуру і функціональні властивості версії прямої дії сітки зворотного поширення.
3. Окреслити роль шару Кохонена при нормальному функціонуванні сітки зворотного поширення.
4. Окреслити роль шару Гроссберга при нормальному функціонуванні сітки зворотного поширення.
5. Розкрити етап попереднього опрацювання вхідних векторів при навчанні шару Кохонена в сітці зворотного поширення.
6. Прокоментувати етап вибору початкових значень вагових векторів при навчанні шару Кохонена в сітці зворотного поширення.
7. Проаналізувати підходи до нормалізації вхідних векторів при навчанні шару Кохонена в сітці зворотного поширення.
8. Розкрити сутність методів акредитації та інтерполяції при навчанні шару Кохонена в сітці зворотного поширення.
9. Прокоментувати особливості навчання шару Гроссберга в сітці зворотного поширення.
10. Навести і прокоментувати основні напрямки, недоліки і переваги практичного застосування сіток зворотного поширення.

Презентації

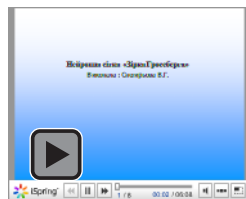
1. Сітки зустрічного розпізнавання



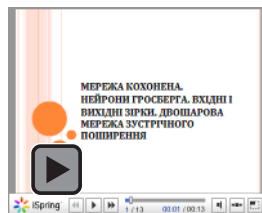
2. Мережі зустрічного поширення



3. Нейронна сітка «Зірка Гроссберга»



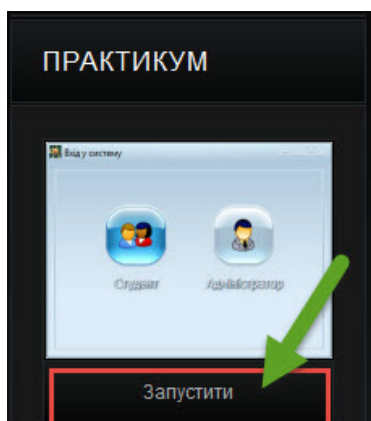
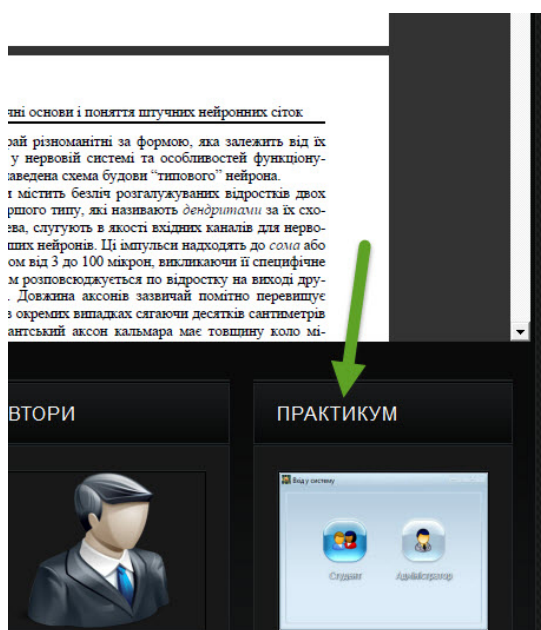
4. Мережа Кохонена. Нейрони Гроссберга. Вхідні і вихідні зірки. Двошарова мережа зустрічного поширення



Практичні заняття

Сітки зустрічного поширення

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 9

Машина Больцмана

*Математика — найбільш довершений спосіб
водити самого себе за ніс*
Альберт Ейнштейн

*Техніка давно визнала високу ціну науки
та її впливом зобов'язана своїм сучасним блискучим розвитком*
М. Є. Жуковський

9.1. Загальні відомості

Машина Больцмана (*Boltzman mashine*) є подібною за функцією та дією до сітки Хопфілда і включає поняття “*модельованого відпалу*” для пошуку глобального мінімуму у просторі станів шару образів.

Еклі Д., Хінтон Д. та Сейновські Т. у 1985 р. розробили правило больцманівського навчання [81; 135; 138; 139; 140]. Подібно до сітки Хопфілда, машина Больцмана має простір станів, який базується на вагах з'єднань у шарі образів. Процес навчання сітки, наповненої образами, включає вивчення *рельєфу простору станів*. Під час ітеративного навчання визначається краща множина значень на виході.

В процесі навчання машина Больцмана моделює відвал металу. Як і при фізичному відпалі, температура починається з вищих значень і зменшується з часом. Збільшена температура додає збільшений шумовий коефіцієнт до кожного нейрону у шарі образів. Звичайно кінцева температура є нульовою. Для досягнення кращого розв'язку доцільно на нижчих температурах додавати більше ітерацій.

Твердження 9.1. Навчаючись на високій температурі, машина Больцмана веде себе більш подібно до *випадкової моделі*, а на низьких температурах — як *детермінована модель*.

Через випадкову компоненту у відпаловому навчанні нейрон може прийняти нове значення стану, яке збільшиться швидше, аніж зменшується загальний простір станів. Імітація фізичного відпалу дозволяє, таким чином, уникнути локального мінімуму, просуваючись до глобального.

Як і у випадку Хопфілда, сітці може бути представлений частковий образ для доповнення відсутньої інформації. Обме-

женням на кількість класів є 15% від загальної кількості елементів у шарі образів (див. розд. 6).

9.2. Енергетична функція

Сітка Хопфілда може виявляти різні образи, які визначаються своїми стартовими векторами в процесі генерації.

Твердження 9.2. Якщо задана Хопфілд-сітка з вагами і граничними значеннями, то вона буде знаходитися в динамічній рівновазі за умови надходження на її вхід образу. Ці стани рівноваги характеризуються тим, що загальна енергія (*Гамільтонова функція*) системи мінімальна.

Перенесення цього положення на сітку Хопфілда приводить до “енергетичної функції”, яка буде мінімальною при надходженні до сітки деякого образу. Енергетична функція може бути визначена в такий спосіб. Нехай задана сітка Хопфілда з вагами w_{ij} і граничними величинами δ_i , тоді можемо уявити енергетичну функцію (функцію Гамільтона) для вектора y як

$$H = -1/2(\sum_i \sum_j w_{ij} y_i y_j + \sum_k \theta_k y_k), \quad (9.1)$$

а у векторній формі цей вираз буде мати вигляд

$$\bar{H} = -1/2[\bar{y}(\bar{W}\bar{y}) + \theta\bar{y}]. \quad (9.2)$$

Розраховуючи для всіх можливих вхідних векторів функцію H , що надходить до сітки, може бути розрахована енергетична функція H , одержуємо “енергетичний ландшафт” з максимумами і мінімумами. Мінімум буде відповідати випадку, коли вхідний сигнал являє собою образ. Отже, мінімуми енергетичної функції є образами сітки Хопфілда.



Рис. 9.1. Графік енергетичної функції H

На рис. 9.1 наведено криву з двома мінімумами. Рівнина на цій кривій уявляє ділянку, де наростання дорівнює нулю і величина H незмінна. Якщо енергія H володіє таким розмитим мінімумом, то існують сусідні стани з такою ж енергією. Звідси випливає висновок:

а) образ може бути локальним мінімумом (рис. 9.1);

б) образ може знаходитися в деякому діапазоні, де сусідні стани мають такого ж значення H .

Досвід показує, що в більшості випадків “рівнина” складається з двох станів, і при одній ітерації можуть виникнути коливання між цими двома станами. У випадку відсутності зазначеної вище “рівнини” ітераційний процес сходиться до образу \bar{y} за співвідношенням

$$\bar{y}^{(p+1)} = \bar{f}(W\bar{y}^{(p)} - \theta), \quad (9.3)$$

де $p = 1, 2, 3 \dots$

9.3. Імітація охолодження

Метод розрахунку Хопфілд-ітераціями. Як було показано вище, сітка Хопфілда являє собою запам'ятовуючий пристрій, що дозволяє запам'ятовування образів, тобто послідовність бітів. При цьому запам'ятовуються всі двійкові величини, для яких енергетична функція мінімальна. Фіксовані при цьому мінімуми є локальними.

У деяких застосуваннях сітки необхідний пошук глобального мінімуму, тобто абсолютного мінімуму енергетичної функції. Такі застосування являють собою, наприклад, базу Больцман-машини. До інших застосувань відносяться задачі оптимізації, в яких здійснюють ідентифікацію енергетичної функції з деяким виразом. Ця процедура дозволяє за допомогою сітки Хопфілда знайти мінімум відповідним методом.

При виконанні розрахунків Хопфілд-ітераціями необхідно враховувати можливість зависання в одному з локальних мінімумів.

Оскільки відомо, що з кожним кроком ітерації енергетична функція зменшується, то може виявитися, що виявлений локальний мінімум буде неможливо покинути, і буде загублений шанс знайти глобальний мінімум. У зв'язку з цим правило для розрахунку Хопфілд-ітераціями необхідно крокообразно змінювати так, що в кінці рахунку буде сформований алгоритм для розрахунку глобального мінімуму. Такий алгоритм має значну аналогію з методом імітаційного моделювання, який імітує охолодження речовини (наприклад кристалізацію). Звідси виникло поняття “імітація охолодження” (“Simulated Annialing”).

Процедура імітаційного моделювання процесу ітерації в сітці Хопфілда виглядає наступним способом. Нехай задана сітка Хопфілда з вагами w_{ij} і граничними коефіцієнтами θ_i . Для двійкових величин вибирають 1 і 0 (замість ± 1). Відомо, що для розрахунку локального мінімуму процес ітерацій має вигляд:

$$y_i^{(\text{in}+1)} = f(\sum_j w_{ij} y_j^{(\text{in})} - \theta_i), \quad (9.4)$$

де in – індекс ітерації.

Цю ітераційну процедуру можна замінити імітаційним процесом, що має покроковий вигляд:

Крок 1 – вибір випадкового нейрона m ;

Крок 2 – розрахунок величини

$$r_m = \sum_j w_{mj} y_j^{(\text{in})} - \theta_m, \quad (9.5)$$

де r_m – стан випадкового m -го нейрона;

Крок 3 – приймаємо

$$\begin{aligned} y_m &= 1, \text{ якщо } r > 0, \\ y_m &= 0, \text{ якщо } r \leq 0; \end{aligned} \quad (9.6)$$

Крок 4 – далі повернення до кроку (1).

В принципі імітаційна процедура забезпечує розрахунок, подібний тому, що виконувався раніше Хопфілд-ітераціями. Він відрізняється лише тим, що *детерміністський процес ітерацій* замінений *стохастичною процедурою розрахунку*. Як активаційна в обох випадках приймається двійкова функція:

$$f(x) = \begin{cases} 1, & \text{якщо } x > 0, \\ 0, & \text{якщо } x \leq 0. \end{cases} \quad (9.7)$$

Проте, і за стохастичною процедурою знаходять лише один локальний мінімум.

Для подальшої модифікації описаного алгоритму на відміну від двійкової функції (9.7) введемо як активаційну сигмоїдну функцію

$$y(r) = \frac{1}{1 + \exp(-r/T)}. \quad (9.8)$$

Тут $T > 0$ – будь-який параметр. За аналогією з імітацією при охолодженні речовини цей параметр одержав найменування “*температура*”. На рис. 9.2 наведений характер графіків функції $y(r)$ для різних величин T . При $T \rightarrow 0$ функція $y(r)$ прямує до двійкової функції (9.7), а при $T \rightarrow \infty$ одержуємо $y(r) \equiv 1/2$.

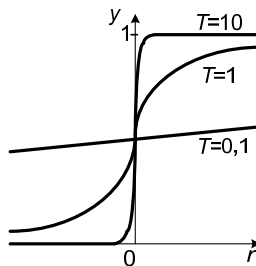


Рис. 9.2. Графіки функцій $y(r)$ для різних величин параметра T

Замінивши в наведеному вище стохастичному алгоритмі двійкову активаційну функцію (9.7) на сигмоїдну (9.8), одержимо змінену ітераційну покрокову процедуру:

Крок 1 – вибір випадкового нейрона m ;

Крок 2 – розрахунок величини

$$r_m = \sum_j w_{mj} y_{mj} - \theta_m; \quad (9.9)$$

Крок 3 – заміщуємо y_{mj} з імовірністю

$$y(r_m) = \frac{1}{1 + \exp(-r_m / T)} \quad (9.10)$$

на 1 (в іншому випадку на 0);

Крок 4 далі повернення до кроку 1.

Вибираючи T , близьке до нуля, одержуємо вихідний алгоритм із двійковою активаційною функцією, при цьому процедура розрахунку приводить до локального мінімуму. Якщо ж задати величину T дуже великою, то $y(r) \cong \sim 1/2$. Це означає, що нейрони незалежно від результатів розрахунку за кроком 2 з імовірністю $1/2$ займуть положення 1, а це свідчить про те, що всі стани сітки можуть прийняти глобальне положення. Звідси випливає, що процедура розрахунку не буде “зависати” в локальному мінімумі. Причому, безумовно, рівень енергії більше не буде знижуватися в кожному випадку.

При виборі деякого проміжного значення T зниження рівня енергії й прямування до мінімуму буде відбуватися з імовірністю, яка тим більше, чим менше значення приймає T . Одночасно можна залишити локальний мінімум, оскільки через $T > 0$ можуть прийматися всі стани. Як буде показано нижче, процедура прямує до глобального мінімуму за умови, якщо починається розрахунок з дуже великими значеннями T , а потім його поступово зменшують (*охолодження*). У початковій стадії через високі температури система буде приймати всі стани. Потім при зниженні температури енергія стохастично зменшується і відбувається “влучення” в глобальний мінімум.

Крок 3 у розглянутому вище алгоритмі для імітації може бути поданий за допомогою ЕОМ в більш зручному вигляді. Виходячи з кроку 3, одержуємо остаточну покрокову структуру алгоритму розрахунку імітації охолодження:

Крок 3.1 – вибір випадкового нейрона;

Крок 3.2 – розрахунок величини

$$r_m = \sum_j w_{mj} y_j - \theta_m;$$

Крок 3.3 – розрахунок імовірності

$$p = \frac{1}{1 + \exp(-r_m/T)}; \quad (9.11)$$

Крок 3.4 – вибір випадкового числа $0 \leq Z \leq 1$ і після заміщення:

$$y_m = 1, \quad \text{якщо } Z \leq p;$$

$$y_m = 0, \quad \text{якщо } Z > p;$$

Крок 3.5 – зменшення T , потім повернення до кроку 1.

У літературі наводиться доказ збіжності методу “імітації охолодження” у стохастичному сенсі, що відповідає досягненню абсолютного мінімуму енергетичної функції.

Наведений опис імітованого відпалу є сильно спрощеним, бо в реальності енергетичний ландшафт є простором великої вимірності, а енергетичний бар’єр між станами не має чітких меж. Це означає, що існує багато способів переходу з одного стану в інший (їх кількість зростає експоненційно із збільшенням додаваного до системи теплового шуму).

Як вище зазначалося, важливе значення має швидкість зниження температури. Якщо температура знижується занадто швидко, система не має достатньої можливості вийти з локального енергетичного мінімуму, і задовольняючий розв’язок може бути недосяжним. Якщо ж температура знижується занадто повільно, система може вийти з локального мінімуму, проте для цього їй знадобиться тривалий час, що також може бути неприйнятним.

9.4. Розподілення Больцмана

Австрійський фізик Людвіг Больцман (*Ludwig Boltzmann*) у 1871 році розробив теорію для опису фізичних властивостей матеріальних систем, що складаються з безлічі часток. Він виходив з ідеалізованого уявлення, що матерія складається з молекул, поведінка яких підпорядковане класичній механіці.

Теорія, що одержала назву “*статистики Больцмана*”, має разючу подібність із приведеними вище математичними формалізмами. Заміняючи частки “статистики Больцмана” станами сітки Хопфілда, переконаємося, що подібні вирази частково є ідентичними. Метод “імітації охолодження” буде відповідати “охолодженню речовини”, і ця аналогія дозволяє краще зрозуміти процеси “імітації охолодження” в сітках Хопфілда.

Статистика Больцмана для статистичного розподілу молекул з енергією E використовує формулу

$$p(E) = \alpha \exp(-E/(kT)), \quad (9.12)$$

де $p(E)$ – імовірність молекул з енергією E ; k – константа розмірності (константа Больцмана); T – абсолютна температура (середня температура по всіх частках).

З цієї формули випливає очевидний вираз для двох енергетичних станів E_1 і E_0 (рис 9.3):

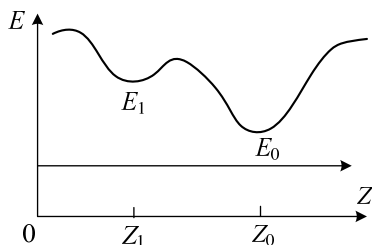


Рис. 9.3. Характер зміни енергії функції Z

$$\frac{p(E_1)}{p(E_0)} = \exp((E_1 - E_0)/(kT)). \quad (9.13)$$

Ця формула (без врахування константи k Больцмана) точно відповідає формулі, що характеризує два стани Z_0 і Z_1 нейронної сітки з відповідними до неї енергіями E_0 і E_1 . Імовірність стану сітки визначається співвідношенням:

$$\frac{p(Z_0)}{p(Z_1)} = \exp((E_1 - E_0)/T), \quad (9.14)$$

з якого випливає, що процедура охолодження, що імітується, стохастично збігається до абсолютного мінімуму.

Нехай енергетична функція сітки має вигляд, наведений на рис. 9.3. У положенні Z_1 знаходиться відносний мінімум, у положенні Z_0 – абсолютний мінімум. Відповідні величини енергії будуть дорівнювати E_1 і E_0 .

Тоді $E_0 < E_1$ чи $E_1 - E_0 > 0$. Звідси випливає

$$\frac{p(Z_0)}{p(Z_1)} = \exp((E_1 - E_0)/T) > 1 \quad (9.15)$$

або $p(Z_0) > p(Z_1)$, тобто імовірність стану абсолютного мінімуму перевищує імовірність стану відносного мінімуму.

З наведених співвідношень можна зробити висновок, що імовірність стану Z з енергією E можна подати у вигляді:

$$p(E) = \alpha \exp(-E/T), \quad (9.16)$$

де α – величина, що не залежить від E .

На рис. 9.4 подані залежності зміни імовірності $p(E)$ від величини енергії E .

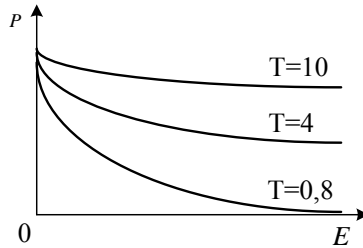


Рис. 9.4. Залежність імовірності p від величини енергії E

Процес імітації охолодження характеризується наступними кроками:

Крок 1 – як стартову величину приймають високу температуру, наприклад $T_0 = 10$. Потім з рис. 9.4 визначається $p(E) \cong \alpha$, тобто тут спостерігається сталість ймовірності. Результати ЕОМ будуть в цій ситуації перестрибувати “туди-сюди” між окремими станами (рис. 9.5).

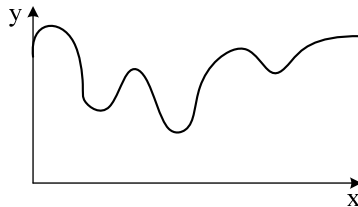


Рис. 9.5. Крива зміни енергії з декількома змінними

Крок 2 – поступово зменшують величину T . З рис. 9.4 видно, що менші стани енергії будуть більш ймовірними. Система прямує в стан з мінімальною енергією і тепер коливається “туди – сюди” між окремими локальними мінімумами;

Крок 3 – задається мала величина T , наприклад $T = 0,1$. Як видно з рис. 4, імовірність станів з великою енергією тепер близька до нуля. Система прямує до абсолютного мінімуму, охолоджується.

Приклад 9.1. (Приклад пошуку абсолютного мінімуму енергії). Розглянемо процедуру імітації охолодження для сітки Хопфілда з n нейронами для пошуку абсолютного мінімуму на ЕОМ за Pascal-програмою. Ваги і граничні величини задаються довільними в межах між -1 і $+1$. Розрахунок починається при температурі $T = 10$. Після кожного етапу розрахунку нейронів визначається величина енергії. Як тільки припиняється зменшення енергії, тобто

$E(j+1) = E(j)$, подальше зменшення температури відбувається за співвідношенням $T(j+1) = 0,9T(j)$.

У першому розрахунку було задано 15 нейронів. Після приблизно 100 ітерацій був знайдений мінімум енергії. У табл. 9.1 наведений якісний процес ітерацій. З нього видно, що на початку температура знижується інтенсивно, у той же час у кінцевій стадії вона більш повільно наближається до нуля. Енергія на початку процесу може навіть декілька зрости, а потім її закінчення прямує до абсолютного мінімуму $-7,06$ і далі вниз.

В наступних розрахунках усі ваги і порогові величини були встановлені на 1. Це має ту перевагу, що мінімальну енергію можна розрахувати вручну. Було отримано $E = -97,5$. Програма одержала іншу величину точно за 54 ітерації.

Табл. 9.1.

Якісний процес ітерацій

Кількість ітерацій	Температура	Енергія
1	10,0	0,92
10	4,8	0,92
20	2,8	0,51
30	2,6	1,90
40	1,2	2,48
50	0,6	0,86
60	0,3	-0,20
70	0,2	0,39
80	0,1	-3,79
90	0,0	-5,48
100	0,0	-7,06

Хопфілд і інші дослідники застосували метод “імітації охолодження” для розв’язування задач оптимізації. При пошуку мінімуму для деякого математичного виразу, якщо вдається цей вираз ідентифікувати з енергетичною функцією сітки Хопфілда, в результаті може бути знайдений мінімум розглядуваної функції.

9.5. Сітки Больцмана

Ці ШНС розроблені в 1985 р. [80]. Фаза вибірки здійснюється імітацією охолодження. Для цього існує процедура навчання ваг.

Топологія сітки. Структура даної сітки наведена на рис. 9.6.

Сітка Больцмана аналогічна сітці Хопфілда: є N цілком взаємозалежних нейронів зі зворотними зв’язками. В загальному випадку

не всі елементи ШНС пов'язані між собою, тобто в матриці ваг можуть бути і нульові елементи.

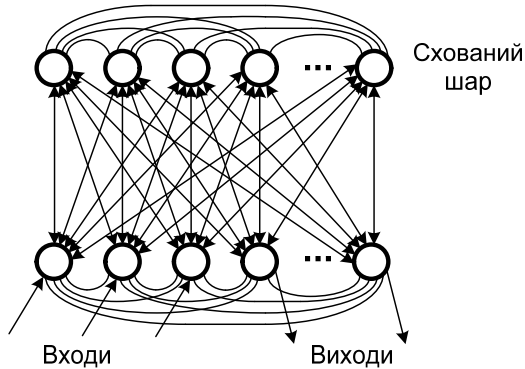


Рис. 9.6. Структура сітки Больцмана

Як і в сітці Хопфілда, для ваг справедливі співвідношення

$$w_{ij} = w_{ji}, \quad w_{ii} = 0.$$

Нейрони сітки розділені на три класи: вхідні, вихідні і сховані і приймають стани 0 і 1. Поряд зі спостережуваними елементами сітка Больцмана може містити і сховані, виконуючі функції стохастичних детекторів ознак і покращуючи завдяки цьому властивості сітки.

Функціонування сітки здійснюється наступним чином: величини, що задають, подаються на вхідні нейрони. Потім відбувається імітоване охолодження, причому так довго, поки величини на вихідних нейронах перестануть змінюватися. Під час процесу охолодження у вхідних комірках не повинно відбуватися ніяких змін. Після цього можна буде знімати вихідні величини з вихідних комірок. Очевидно, що кожному вхідному вектору \bar{X} наводиться у відповідність вектор \bar{Y} на виході і утворюється функціональна залежність

$$\bar{Y} = \bar{f}(\bar{X}), \quad (9.17)$$

яка запам'ятовується сіткою.

Сітковий вхід для j -го елемента сітки Больцмана може бути представлений виразом:

$$x_j = \sum_{i \in S_j} w_{ij} y_i - \theta, \quad (9.18)$$

де w_{ij} — вага зв'язку i -го та j -го елементів; y_i — двійкова активаційна функція на виході j ; θ — поріг (може дорівнювати нулю); S_j — множина елементів сітки, пов'язаних з j -м.

Значення y_i є бінарними (0, 1) або біполярними (-1, 1) і задаються наступним чином:

$$y_j = \begin{cases} +1 \text{ з імовірністю } p(x_j); \\ -1 \text{ з імовірністю } 1 - p(x_j). \end{cases} \quad (9.19)$$

При цьому приймається, що випадкові величини x мають сигмоїдальну функцію розподілення

$$p(x) = \frac{1}{1 + e^{-\frac{x}{T}}}. \quad (9.20)$$

Параметр T у виразі (9.20) уявляє собою керуваний параметр “температури”, який адаптується під час функціонування сітки.

Одне з багатьох застосувань цього типу сіток полягає в можливості асоціативного запам'ятовування даних.

Методика навчання сітки. Застосування цього типу сіток можливе, якщо відомі вірні ваги і порогові величини, і вони зафіксовані в пам'яті сітки. Визначення усіх ваг виконується шляхом адаптивного навчання, основна ідея якого полягає в проведенні імітаційного охолодження:

а) тільки для схованих комірок вхідні і вихідні величини досліджуваної функції вводяться у вхідні і вихідні комірки (*плюс-фаза*);

б) для всієї сітки після того, як вхідні і вихідні величини введені в сітку (*мінус-фаза*).

Обидва охолодження проводяться по чергову, і після кожного кроку ваги змінюються так, що обидва процеси збігаються. Метою процедури є в кінці кінців одержання таких значень ваг, щоб при вільному охолодженні у випадку (б) вийшло таке ж співвідношення між входом і виходом сітки, як і при (а). Якщо вдається цього досягти, то сітка буде відбивати функцію, що вивчалася. У наступному кожна реалізація запам'ятованої функції буде характеризуватися локальним мінімумом енергетичної функції. Іншими словами, усі мінімуми енергетичної функції являють собою запам'ятовані функції.

Тепер необхідно вивести формулу для корегування ваг. Нехай задана підлягаюча вивченню сіткофункція у вигляді:

$$\bar{Y}_k = \bar{f}(\bar{X}_k), \quad k = 1, 2, 3, \dots, N,$$

де \bar{X}_k – вхідний вектор; \bar{Y}_k – вектор на виході.

Кожний стан сітки можна подати у вигляді

$$z_k = (a_1, a_2, a_3, \dots, a_n); \quad (a_j \in \{0; 1\}).$$

Окремо визначаються:

Z_k^+ – стан сітки, отриманий в результаті імітації охолодження,

для випадку, коли до вхідних і вихідних комірок прикладені вхідний і вихідний вектори, і ці комірки в процесі охолодження більше не змінюються (охолодження тільки для схованих комірок, тобто плюс-фаза);

Z_k^- – стан сітки, отриманий в результаті імітації охолодження, для випадку вільного функціонування всієї сітки (мінус-фаза).

Нехай для цих випадків:

p_k^+ – імовірність для стану Z_k^+ ;

p_k^- – імовірність для стану Z_k^- .

Очевидно, що сітка може вважатися налаштованою на досліджувану функцію, якщо ваги сітки вивчені так, що

$$p_k^+ = p_k^- \quad (9.21)$$

Оскільки при охолодженні (мінус-фаза) досягається мінімум енергії, то і шукані стани досягають високої імовірності: p_k^- велика і p_k^+ також велика. Таким чином, при справедливості співвідношення (9.21) з високою імовірністю в сенсі досліджуваної функції досягнуто вірний вихідний сигнал сітки.

Існує потенційна функція, яка виконується за умови, породжуваної наступним співвідношенням:

$$p = -\sum_k \ln(p_k^+ / p_k^-) \quad (9.22)$$

Можна показати, що p є мірою відстані між станами Z_k^+ й Z_k^- і завжди $p > 0$, а в ідеальному випадку $p = 0$.

Алгоритм функціонування сітки Больцмана. Узагальнюючи вищенаведене, можна сформулювати алгоритм функціонування сітки Больцмана, який включає наступні фази:

1. Визначити змінну T , що представляє штучну температуру.
2. Пред'явити сітці множину входів і обчислити виходи та цільову функцію.
3. Дати випадкову зміну вагам і перерахувати вихід сітки та зміну цільової функції у відповідності зі зробленою зміною ваг.
4. Якщо цільова функція зменшилася, то зберегти змінені ваги. Якщо зміна ваг приводить до збільшення цільової функції, то ймовірність збереження цієї зміни обчислюється за допомогою розподілу Больцмана:

$$P(x) = \exp\left(-\frac{x}{kT}\right), \quad (9.23)$$

де $P(x)$ – імовірність зміни стану x у цільовій функції; k – константа, аналогічна константі Больцмана, що вибирається в залежності від задачі; T – штучна температура.

Константа (постійна) k Больцмана – фізична постійна, яка

визначає зв'язок між температурою та енергією. Її експериментальне значення у Міжнародній системі одиниць (СІ) дорівнює: $k = 1,3806488 \times 10^{-23}$ Дж/К.

5. Вибирається випадкове число r з рівномірного розподілу від нуля до одиниці. Якщо $P(x)$ більше, ніж r , то зміна зберігається, у противному випадку величина ваги повертається до попереднього значення.

Ця процедура дає можливість системі робити випадковий крок у напрямку, який псує цільову функцію, дозволяючи їй тим самим “виірватися” з локальних мінімумів.

Кроки 3 і 4 повторюються для кожної з ваг сітки, поступово зменшуючи температуру T , поки не буде досягнуто припустиме низьке значення цільової функції. У цей момент пред'являється інший вхідний вектор, і процес навчання повторюється. Сітка навчається на всіх векторах навчальної множини, поки цільова функція не стане припустимою для усіх з них.

Швидкість зменшення температури повинна бути зворотно пропорційна логарифму часу. При цьому сітка збігається до глобального мінімуму.

Недоліком сітки є повільний час навчання, а *перевагою* – здатність вибиратися з локальних мінімумів адаптивного рельєфу простору станів.

Модифікаціями алгоритму навчання є проведення випадкових змін не тільки для окремих ваг, але й для всіх нейронів шарів у багатошарових сітках або для всіх нейронів сітки одночасно. Ці модифікації алгоритму дають можливість скоротити загальну кількість ітерацій, а відтак – і час навчання.

Приклад 9.2. *Розв'язання задачі комівояжера.* Раніше при розгляданні сіток Хопфілда (розд. 6.7) була сформульована задача комівояжера і наведений приклад її розв'язання за допомогою нейросітки Хопфілда (див. приклад 6.4). В умові 4 до цього прикладу наголошувалося, що особливого значення набуває питання щодо призначення ваг для досягнення задовольняючого розв'язку при використанні машини Больцмана: *при присвоєнні ваг повинна існувати гарантія, що НС надасть розв'язок, який відповідатиме коректному туру.*

Розглянемо детальніше процедуру формування ваг сітки з урахуванням алгоритму “Імітації відпалу” Больцмана.

Рисунки 9.7, а та б ілюструють з'єднання, зроблені до кожного вузла НС. Вони розділені на два типи. З'єднання віддалей, для яких ваги вибрані таким чином, що якщо сітка знаходиться в стані, який відповідає туру, ці ваги будуть відбивати вартість енергії E з'єднання $(j-1)$ -го міста туру з j -м містом, а j -е місто з $(j+1)$ -м містом.

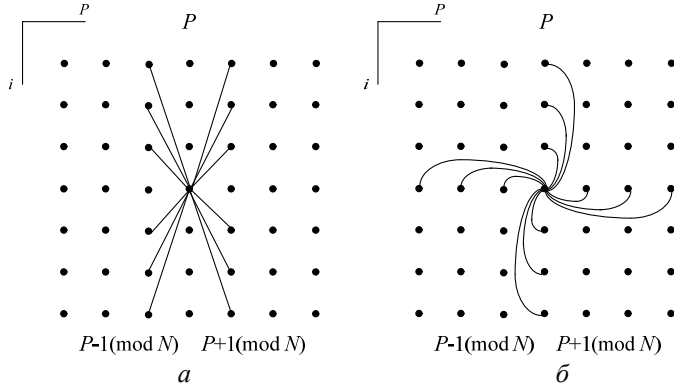


Рис.9.7. Ілюстрація зв'язків наявних (а) та виключаючих (б) з'єднань віддалей

На рис. 9.7, а зображені з'єднання віддалей. У кожного (i, j) –го вузла є забороняючі зв'язки до двох суміжних стовпців, ваги яких відбивають вартість з'єднання трьох міст.

На рис. 9.7, б наведені виключаючі з'єднання. У кожного (i, j) –го вузла є заперечуючі зв'язки до всіх елементів у тому ж рядку і стовпці.

Виключаючі з'єднання забороняють двом елементам знаходитись в одному й тому ж рядку або стовпчику в один і той же час. Виключаючі з'єднання дозволяють сітці встановлюватись в стан, відповідаючий стану туру.

Через те, що всі з'єднання до цього часу були нами визначені як забороняючі, треба тепер надати сітці деякий стимул для включення елементів за допомогою маніпулювання порогами. Інтуїтивно можна бачити, що деякі розташування, наприклад, ті, що наведені на рисунках 9.7, а та б, можуть володіти необхідним ефектом, і наступна теорема вказує, як точно вибирати ваги. Розглянемо наступний набір параметрів:

$$\begin{aligned} St &= (\Theta_{ip} : 0 \leq i, p \leq N-1); \\ Sg &= (w_{ipjq} : i \neq j \cup q \equiv p \pm 1 \pmod{N}); \\ SE &= (E_{ipjq} : (i = j \cup p \neq q) \cap (i \neq j \cup p = q)), \end{aligned} \quad (9.24)$$

де g_{in} – віддаль між i -м та p -м містами; N – кількість міст; E_{inia} – енергія між нейронами (j, q) та (i, p) ; θ_{in} – є входом елемента (i, p) , а w_{inia} – вага від елемента (j, q) -го до (i, p) -го. Зазначимо, що кожний елемент на рисунках 9.7, а та б матрично зв'язаний, тому будь-який особливий елемент ідентифікується двома підписами.

Виберемо ваги і входи таким чином, щоб:

$$\begin{aligned} \forall \Theta_{ip} \in St : \Theta_{ip} &< -\min(g_{if} + g_{iv} : f \neq v, 0 \leq f, v \leq N-1); \\ \forall w_{ipjq} \in Sg : w_{ipjq} &= -g_{ij}; \\ \forall w_{ipjq} \in SE : E_{ipjq} &< \min(\Theta_{ip}, \Theta_{iq}). \end{aligned} \quad (9.25)$$

Тоді:

1. *Виконуваність*. Коректні стани турів НС точно співвідносяться з локальним мінімумом енергії.
2. *Упорядкування*. Функція енергії є порядкозберігаючою у відповідності з довжиною туру.

Доказ є досить прямолінійним і залежить від припущення, що стани турів абсолютно точно співвідносяться з локальним мінімумом енергії. В асинхронній машині Больцмана локальні мінімуми енергій абсолютно точно співвідносяться з фіксованими точками. Недоліком алгоритму є те, що в синхронному випадку деякі локальні мінімуми енергій можуть також відповідати двом циклам, що ускладнює присвоєння ваг в синхронному випадку.

При розв'язанні задачі комівояжера функціонування НС виглядає наступним чином:

Крок 1. З урахуванням поставленої задачі за алгоритмом, наведеним вище в теорії, розраховуються ваги.

Крок 2. Подається на вхід сигнал.

Крок 3. Виконується активація нейронів за алгоритмом навчання Больцмана, за винятком того, що ваги залишаються завжди однаковими.

Крок 4. Повторюються дії кроків 2 та 3 доти, доки енергія не попаде у глобальний мінімум (тобто за період n обчислень не змінює свого стану).

Оптимізація роботи алгоритму. Розв'язуючи задачу за допомогою алгоритму "Імітації відпалу" крім правильного підбору ваг, вхідного вектора, порогових значень і розмірності, не менш важливий параметр t (температура).

При поступовому зниженні температури енергії атомів знижуються, і при цьому система в цілому прагне прийняти конфігурацію з мінімальною енергією. Коли охолодження завершено, досягається стан глобального мінімуму енергії.

При фіксованій температурі розподіл енергій системи визначається ймовірнісним фактором Больцмана.

9.6. Машина Коші

Загальні властивості. Маючи ряд суттєвих переваг у порівнянні з багатошаровим персептроном (в першу чергу, можливістю виходу з локальних екстремумів), машина Больцмана все ж не знайшла настільки широкого розповсюдження, як персептрон. Це пояснюється, насамперед, тим, що розв'язання практичних задач супроводжується суттєвими обчислювальними витратами, якими супроводжується реалізація алгоритму відпалу, а отже для досягнення сіткою стану рівноваги необхідне проведення великої кількості обчислень для кожного значення температури. Суттєвого спрощення алгоритму відпалу можна досягти, якщо замість розподілення Больцмана (9.23) використати *розподілення Коші*

$$P(x, \theta) = \frac{1}{\pi |1 + (x - \theta)^2|}, \quad (9.26)$$

де θ – центр симетричного розподілення.

Штучні НС, які використовують дане розподілення, називають *машиною Коші*. Розподілення (9.26) відноситься до “розподілення з важкими хвостами”, другі моменти яких нескінченні. Проте, використання (9.26) замість (9.23) в алгоритмі відпалу *дозволяє робити великі кроки при навчанні ШНС*, прискорюючи тим самим процедуру охолодження. При цьому значення температурного коефіцієнта $T(k)$ змінюється за правилом:

$$T(k) = \frac{T_0}{1+k}, \quad (9.27)$$

причому як стартову величину приймають високу температуру, наприклад, $T_0 = 10$.

Схожість та відмінність машин Коші і Больцмана. Можна виокремити наступні основні відмінності властивостей машин Коші і Больцмана:

- архітектура машини Коші *може містити сховані шари* (на рисунку наявний один схований шар), які виконують функції стохастичних детекторів ознак і покращують завдяки цьому властивості сітки;
- структура *має зворотні зв'язки* між нейронами, але, як і в машині Больцмана, в загальному випадку не всі елементи можуть пов'язуватися між собою, тобто *в матриці ваг можуть бути і нульові елементи*;
- машина Коші, як і машина Больцмана *є сполученням детермінованої сітки Хопфілда з ймовірнісним правилом навчання*;
- процес навчання цієї штучної нейронної мережі *є стохастичним*;
- значення вагових коефіцієнтів вибирається випадковим чином;
- процес навчання в машині Коші *заснований на використанні «моделі відпалювання»*;
- в машині Коші використовується, на відміну від машини Больцмана, суттєво спрощений алгоритм відпалу – *шляхом заміни розподілу Больцмана на розподіл Коші*;

Алгоритм навчання машини Коші. Алгоритм навчання машини Коші включає наступні кроки:

Крок 1. Усім вагам мережі привласнюються невеликі початкові значення та значення температурного коефіцієнта T_0 .

Крок 2. На спостережувані нейрони подаються навчальні пари x_p, y_p доти, поки можливі зміни станів прихованих нейронів.

Крок 3. Випадковим чином вибирається прихований нейрон і змінюється його стан з x_i на x'_i , що призводить до зміни енергії.

Якщо $\Delta E_l = E'_l - E_l$ від'ємна, то здійснюється активація l -го нейрона незалежно від його попереднього стану. Якщо ΔE_l додатна та імовірність P_l нового стану l -го нейрона більша за апіорну P_0 , то здійснюється активація l -го нейрона. Вираховується імовірність $P(x, \theta)$ на основі розподілення Коші (9.26). В іншому випадку стан l -го нейрона не змінюється.

Крок 4. Значення температурного коефіцієнта T_0 зменшується.

Крок 5. (ітераційний). Попередні кроки повторюються доти, поки не досягнуто мінімальної температури.

Крок 6. Стани всіх нейронів запам'ятовуються у вигляді вектора r_c^P .

Крок 7. Обчислюються оцінки кореляцій для всіх пар нейронів

$$r_{ij}^c = \frac{1}{P} \sum_{P+1}^P \Phi(z_i^c, h_{ci}^c). \quad (9.28)$$

Крок 8. (ітераційний). Попередні кроки повторюються для вільних нейронів на виході.

Крок 9. Обчислюються оцінки кореляцій для вільних нейронів

$$r_{ij}^f = \frac{1}{P} \sum_{P+1}^P \Phi(z_i^f, h_{fi}^f). \quad (9.29)$$

Крок 10. Корегуються вагові коефіцієнти

$$\Delta w_{ij} = \alpha [r_{ij}^c - r_{ij}^f]. \quad (9.30)$$

Процес навчання припиняється, якщо всі зміни ваг стануть нульовими або досить малими.

Особливості процесу навчання. На другому кроці роботи алгоритму одному і тому самому вхідному образу можуть відповідати кілька бажаних реакцій системи.

При виборі схеми охолодження за правилом (9.27)

$$T(k) \geq \frac{T_0}{1+k},$$

де T_0 – досить велика величина, алгоритм збігається в асимптоті з імовірністю 1. Для прискорення процесу обчислення і, як наслідок, навчання на відміну від машини Больцмана в знаменнику $\log k$ замінюється параметром k .

Особливості вибору температурного коефіцієнту T_0 . Спочатку значення T_0 повинно бути досить великим. В подальшому зна-

чення T_0 повинно поступово зменшуватись, і процес переходу мережі в наступний стан є ітераційним.

Після навчання на заданій множині образів сітка може бути застосована для класифікації (розпізнавання) невідомих або спотворених образів.

Алгоритм класифікації. Алгоритм класифікації в машини Коші включає наступні кроки:

Крок 1. Усім вихідним сигналам випадковим чином привласнюються значення 1 та -1.

Крок 2. Задається велике (початкове) значення параметра T_0 .

Крок 3. Подається вхідний образ \mathbf{X} .

Крок 4. Вибирається деякий j -ий нейрон шару на виході або прихованого шару та обчислюється вхідний сигнал

$$x_i = \sum_{i \in S_j} w_{ij} y_i.$$

Крок 5. Незалежно від поточного стану стан j -го нейрону приймається рівним 1, якщо $P(x_j) > P_0$, де P_0 – наперед задане значення імовірності.

Кроки 2 і 3 циклічно повторюються до тих пір, поки всі нейрони шару на виході і прихованого шару не змінили свого стану хоча б один раз.

Цей цикл повторюється багаторазово до досягнення рівноважного стану.

Далі зменшується температурний коефіцієнт T_0 відповідно до правила:

$$T(k+1) = \beta T(k), \text{ причому } \beta \in (0; 1);$$

$$T(k) = T_0 / (1 + t).$$

Кроки 2-5 повторюються до тих пір, поки сітка не досягне стійкого стану, що свідчить про розпізнавання нею поданого вхідного образу.

Переваги машини Коші: можливість виходу з локального екстремуму; значно зменшено час навчання у порівнянні з аналогічною машиною Больцмана.

Недоліки машини Коші: не дивлячись на підвищення швидкості навчання машиною Коші у порівнянні з машиною Больцмана, час збіжності все ще може в 100 разів перевищувати час для алгоритму зворотного поширення. Відзначимо також і можливість сіткового паралічу, який особливо небезпечний для алгоритму навчання Коші у випадку сітки з нелінійністю типу логістичної функції. Не-

скінчена *дисперсія розподілення Коші* приводить до змін ваг до необмежених величин. Далі, великі змінення ваг будуть іноді прийматися навіть у тих випадках, коли вони несприятливі, часто приводячи до сильному насичення сіткових нейронів з впливаючим через це ризиком паралічу.

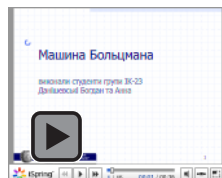
Висновки: нейросітки типу машини Коші викликають великий інтерес через те, що дозволяють моделювати деякі властивості біологічних мереж. Проте, будучи призначеною для «проскакування» локального екстремуму і визначення глобального, їй не завжди це вдається, а кількість обчислень і їх складність залишаються досить великими.

Контрольні завдання

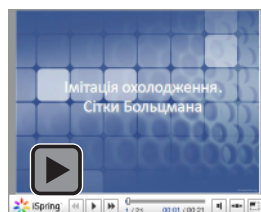
1. Розкрити термін “модельований відпал” у фізиці функціонування машини Больтцмана.
2. Пояснити, чим характеризуються стани динамічної рівноваги в Хопфілд-сітці.
3. Прокоментувати графік енергетичної (Гамільтонової) функції у випадку навчання сітки Хопфілда.
4. Проаналізувати процес “імітації охолодження” методом розрахунків Хопфілд-ітераціями.
5. Пояснити процедуру стохастичного алгоритму ітераційного покрокового імітованого відпалу.
6. Навести основні положення теорії “статистика Больтцмана” і дати їй інтерпретацію станами сітки Хопфілда.
7. Співставити сітки Больтцмана і Хопфілда та виокремити аналогію і відмінності між ними.
8. Охарактеризувати методику навчання сітки Больтцмана, загостривши увагу на плюс- та мінус-фазах охолодження.
9. Дати детальне тлумачення алгоритму функціонування сітки Больтцмана.
10. Пояснити відмінність в алгоритмах відпалу у розподіленнях Больтцмана і Коші.
11. Виокремити основні відмінності властивостей машин Коші та Больтцмана.
12. Прокоментувати послідовності кроків алгоритму навчання машини Коші.
13. Навести основні кроки алгоритму класифікації в машині Коші.
14. Розкрити переваги й недоліки машин Больтцмана і Коші.

Презентації

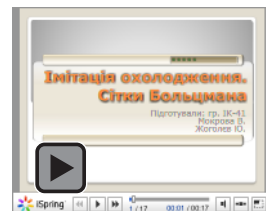
1. Машина Больцмана



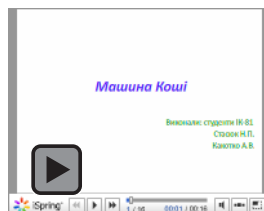
2. Імітація охолодження. Сітки Больцмана



3. Імітація охолодження. Сітки Больцмана



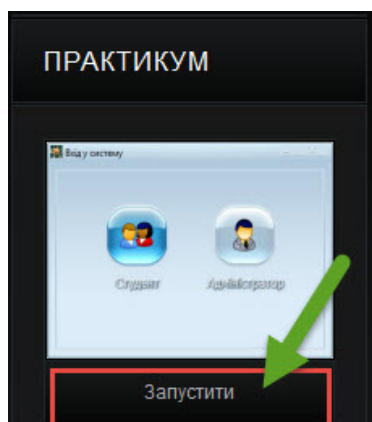
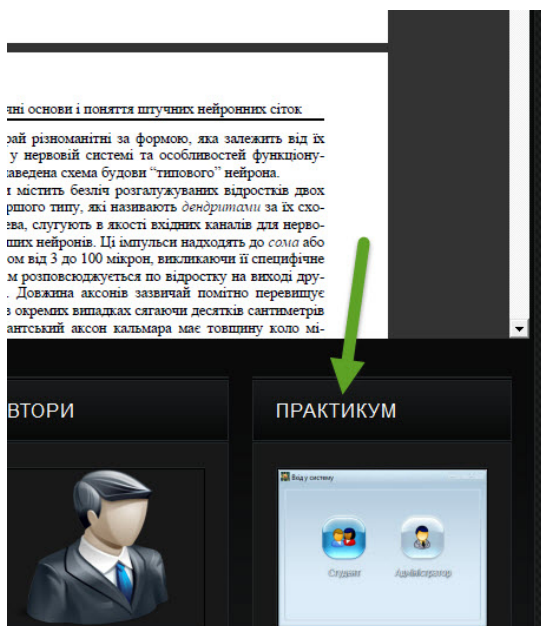
4. Машина Коші



Практичні заняття

Машина Больтцмана

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 10

Нейросітки радіально-базисних функцій

*Тому, що Ви не можете робити усе, що хочете,
бажайте тільки того, що можете зробити*
Теренцій Публій

*Майбутнє наповнено невизначеності,
але ця обманливість майбутнього є найбільшим благом*
Фукидід

10.1. Загальні відомості

Сітки *радіально-базисних функцій* (РБФ-сітки) – спеціальний тип ШНС з прямими зв'язками, запропонований В. Повелом [184]. Основне їх призначення – апроксимація та інтерполяція багатовимірних функцій для розв'язання, зокрема, задач прогнозування [89; 167]. Математичну основу РБФ-сіток, розроблену Айзерманом М. А., Браверманом Е. М. та Розоноєром Л. І. [1], складає теорія апроксимації та інтерполяції багатовимірних функцій, згідно з якою скільки завгодно точна апроксимація функцій досягається комбінуванням радіально-симетричних функцій. РБФ-сітки володіють рядом характерних властивостей [48]:

- мають архітектуру з прямими зв'язками першого порядку: від нейронів одного шару до нейронів наступного шару;
- характеризуються швидким навчанням;
- відсутні “патології” збіжності: в них, на відміну від нейросіток зворотного поширення, не виникають проблеми локальних мінімумів;
- є непоганими апроксиматорами функцій;
- потребують тривалішого часу підготовки та настроювання через необхідність виконання більш складних розрахунків.

Особливістю цих ШНС є наявність радіально-симетричного шаблонного шару.

10.2. Архітектура РБФ-сіток

Структура РБФ-сіток. Сітка радіально-базисних функцій (структура наведена на рис. 10.1) має тільки один шар схованих нейронів, кількість яких зазвичай відповідає кількості елементів у навчаючій послідовності. Крім того, вона містить один вхідний шар і

один шар на виході з одного (саме такий випадок формування зв'язків на виході схованого шару відображений на рис. 10.1) або декількох нейронів. На рис. 10.1 єдиний нейрон на виході видає значення функції, яка і складає основу РБФ-сітки за методом потенціальних функцій [1] і дозволяє подати деяку функцію $y(x)$ у вигляді суперпозиції *потенціальних або базисних функцій*:

$$y(x) = \sum_{i=1}^N \mathbf{a}_i \mathbf{h}_i(x) = \mathbf{a}^T \mathbf{f}(x),$$

де $\mathbf{a}_i = (a_1, a_2, \dots, a_N)^T$ – вектор підлягаючих визначенню параметрів;

$\mathbf{h}(x) = (h_1(x), h_2(x), \dots, h_N(x))^T$ – вектор базисних функцій.

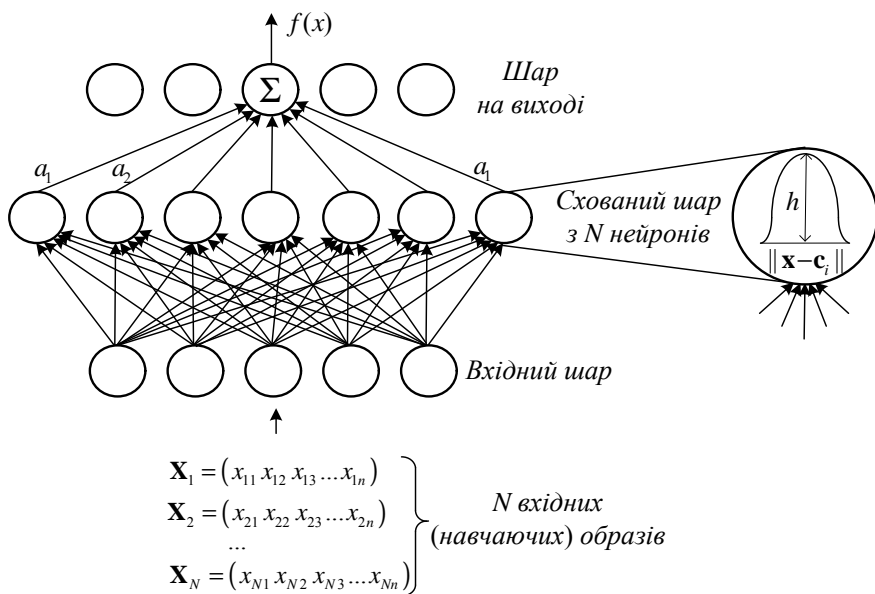


Рис. 10.1. Структура РБФ-сітки

За своєю побудовою РБФ-сітки – двошарові сітки першого порядку, причому обидва шари з'єднані ваговою матрицею \mathbf{W} . Вхідний вектор \mathbf{X} передається на нейрони схованого (шаблонного) шару, кожний з яких отримує повну інформацію щодо вхідного вектора \mathbf{X} .

Радіально-базисна сітка *характеризується трьома особливостями*:

1. Єдиний схований шар;
2. Тільки нейрони схованого шару мають нелінійну активаційну функцію;

3. Синаптичні ваги зв'язків вхідного та схованого шарів дорівнюють одиниці.

Нейрон схованого (шаблонного) шару. Кожний нейрон I схованого (шаблонного) шару розраховує значення одновимірної функції h_i (наприклад, «дзвін» Гауса):

$$h_i(x) = \exp \left[-\frac{(\|\mathbf{X} - \mathbf{C}_i\|)^2}{2\sigma_i^2} \right], \quad (10.1)$$

де \mathbf{X} – вхідний вектор; \mathbf{C}_i – вектор координат i -ї опорної точки або i -го образу навчаючої послідовності (ще називають *центрами* базисних функцій); σ_i – параметр розсіювання для одновимірної функції h_i . В якості *метрики* $\|\mathbf{X} - \mathbf{C}_i\|$ зазвичай використовуються евклідова

$$\begin{aligned} \|\mathbf{X} - \mathbf{C}_i\| &= \sqrt{(x_1 - c_{i1})^2 + (x_2 - c_{i2})^2 + \dots + (x_n - c_{in})^2} = \\ &= \left(\sum_{j=1}^N (x_j - c_{ij})^2 \right)^{1/2}, \end{aligned} \quad (10.2)$$

або манхеттенська віддаль:

$$\|\mathbf{X} - \mathbf{C}_i\| = \sum_{j=1}^N |x_j - c_{ij}|, \quad (10.3)$$

де $|x_j - c_{ij}| = (x_j - c_{ij}) \operatorname{sign}(x_j - c_{ij})$,

$$\operatorname{sign}(x_j - c_{ij}) = \begin{cases} 1, & \text{якщо } (x_j - c_{ij}) > 0; \\ 0, & \text{якщо } (x_j - c_{ij}) = 0; \\ -1, & \text{якщо } (x_j - c_{ij}) < 0. \end{cases} \quad (10.4)$$

Радіальні базисні функції h_i (10.1) виконують попередню обробку вхідних векторів, визначаючи їх близькість до центрів \mathbf{C}_i . Значення $h_i(\mathbf{X})$ на виході – це ступені подібності між вхідним вектором \mathbf{X} та окремими центрами c_i . На основі значень $h_i(\mathbf{X})$ визначається зважена сума.

Центри c_i визначаються на основі навчаючої послідовності і мають таку ж саму розмірність n , що й вхідний вектор. Експериментально визначається параметр σ_i . Принцип роботи нейрона схованого шару ілюструється рис. 10.2.

З формули (10.1) випливає: вихід нейрона схованого шару тим більший, чим ближче поточний вхід \mathbf{X} до центра c_i . Вихід $f(x)$ РБФ-сітки визначається у вигляді зваженої суми

$$f(x) = y = \sum_{i=1}^k a_i h_i(x). \quad (10.5)$$

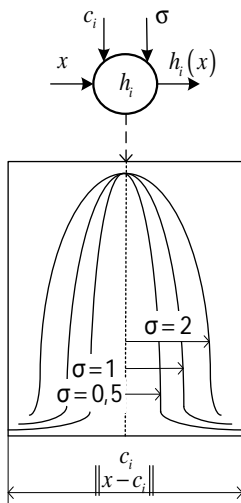


Рис. 10.2. Принцип роботи нейрона схованого шару

Нейрони шару на виході утворюють, таким чином, лінійну комбінацію виходів нейронів схованого шару. Їх функція активації або виходу – це радіальна базисна функція (10.1).

Одновимірні РБФ-сітки можуть бути узагальнені на багатовимірні функції. В цьому випадку в шарі на виході розміщуються m нейронів за кількістю апроксимовуваних (або інтерпольовуваних) функцій, причому значення на виході j -го нейрона шару на виході визначається наступним чином:

$$y_i = \frac{\sum_{i=1}^k a_{ij} h_i}{\sum_{i=1}^k h_i}. \quad (10.6)$$

В РБФ-сітках як опорні точки в найпростішому випадку можуть бути використані образи навчаючої послідовності. Відзначимо дві переваги РБФ-сіток у порівнянні з іншими нейросітками:

- функція активації або виходу схованого нейрона приймає більші значення лише у тих випадках, коли вхідний образ знаходиться поблизу від опорної точки цього нейрона. Для входів поза цієї області, «покритої» образами навчаючої послідовності, сітка формує лише невеликі значення на своїх виходах, причому для сіток з сигмоїдаль-

ними функціями активації або виходу (наприклад, в багатошарових перцептронах) ці виходи можуть бути взагалі не передбачувані;

- проста структура РБФ-сіток, які містять тільки один шар схованих нейронів, робить можливим прямий (не ітеративний) розрахунок ваг сітки. В цьому – їх значна перевага в порівнянні з іншими типами НС, які у більшості випадків використовують трудомісткі рекурентні алгоритми навчання. Заслугує також уваги можливість ініціалізації РБФ-сіток на основі алгоритмів навчання з заохоченням (наприклад, на основі алгоритму із зворотним поширенням похибки).

Розглянемо нижче спочатку математичні основи, а далі архітектури сітки, розширення у вигляді узагальнених РБФ-сіток, спеціальні алгоритми визначення опорних точок.

10.3. Математичні основи РБФ-сіток

Інтерполяція за допомогою центральних функцій. Теорія РБФ-сіток базується на математичній теорії апроксимації функцій багатьох змінних. Розглянемо спочатку проблему інтерполяції однієї функції, далі проблему апроксимації і нарешті – розширення у вигляді сіток узагальнених РБФ-функцій та гіпербазових (ГБФ) функцій.

Нехай $f: \mathbf{R}^n \rightarrow \mathbf{R}$ – функція, здійснююча перетворення з \mathbf{R}^n у \mathbf{R} , задана своїми значеннями в M опорних точках (образи навчаючої послідовності). Для кожної опорної точки c_i задається n -вимірний вхідний вектор

$$\mathbf{C}_i = (c_{i1}, c_{i2}, \dots, c_{in})'$$

та відповідне дійсне значення функції $y_i \in \mathbf{R}$. Для всіх опорних точок \mathbf{C}_i ($i = 1, 2, \dots, N$) мають виконуватись відповідно умови інтерполяції:

$$f(\mathbf{C}_i) = y_i. \quad (10.7)$$

Для розв'язання приведеної задачі інтерполяції використовується система з радіальних базисних функцій \mathbf{h}_i :

$$\mathbf{h}_i(\|\mathbf{X} - \mathbf{C}_i\|), \quad i = 1, 2, \dots, N. \quad (10.8)$$

Інакше кажучи, базисні функції $\mathbf{h}_i: \mathbf{R}^n \rightarrow \mathbf{R}$ залежать від вибору опорних точок \mathbf{C}_i . Свою назву РБФ-сітки отримали саме за цими функціями, приймаючими дійсні значення. В якості аргументу (входу) вони використовують віддаль між вхідним вектором \mathbf{X} та опорним вектором \mathbf{C}_i нейрона i схованого шару. При цьому зазвичай визначається евклідова віддаль ($L2$ -норма) між обома точками \mathbf{X} та \mathbf{C}_i в \mathbf{R}^n .

Шукана функція подається при цьому у вигляді:

$$f(x) = \sum_{i=1}^N a_i h_i(\|x - c_i\|). \quad (10.9)$$

При розв'язанні проблеми інтерполяції РБФ-сітками опорними точками і відповідно ваговими векторами w_i нейронів схованого шару є опорні вектори $C_i : w_i = C_i$, тобто i -й схований нейрон як ваговий вектор має i -й образ навчаючої послідовності. Отже, задача інтерполяції зводиться до визначення вагових коефіцієнтів w_i , $i = 1, 2, \dots, N$. На рис. 10.3 наведено приклад подання функції у вигляді накладання (лінійної суперпозиції) чотирьох радіальних одновимірних базисних/центральных функцій з однаковими дисперсіями σ^2 . Функція

$f(x) = 0,3h(\|X - 0,5\|) + 0,5h(\|X - 2\|) + 0,4h(\|X - 4\|) - 0,3h(\|X - 5\|)$ подана у вигляді зваженої суми окремих функцій Гауса з чотирма центрами $X_1 = 0,5$, $X_2 = 2$, $X_3 = 4$, $X_4 = 5$ та вагами $a_1 = 0,3$, $a_2 = 0,5$, $a_3 = 0,4$, $a_4 = -0,3$.

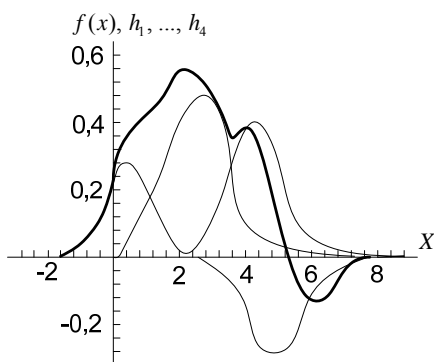


Рис. 10.3. Приклад подання функції у вигляді накладання (лінійної суперпозиції) чотирьох радіальних одновимірних базисних функцій з однаковими дисперсіями σ^2

Вагові коефіцієнти a_i можна визначити безпосередньо у вигляді розв'язку лінійної системи рівнянь для проблеми інтерполяції:

$$\sum_{i=1}^N a_i h_i(\|x_j - c_i\|) = y_j, \quad j = 1, 2, \dots, N. \quad (10.10)$$

Ми отримали систему з N рівнянь для N невідомих. Можна показати, що при виборі відомих функцій Гауса в якості центральных функцій проблема інтерполяції (10.10) безпосередньо розв'язувана.

Уведемо векторні позначення:

$$\mathbf{C} = (c_1, c_2, \dots, c_N)', \mathbf{y} = (y_1, y_2, \dots, y_N)'$$

та

$$\mathbf{H} = \begin{pmatrix} h_1(\|x_1 - c_1\|) & \dots & h_N(\|x_1 - c_N\|) \\ \dots & \dots & \dots \\ h_1(\|x_N - c_1\|) & \dots & h_N(\|x_N - c_N\|) \end{pmatrix}, \quad (10.11)$$

і тоді проблему інтерполяції (10.10) можна представити так:

$$\mathbf{H}\mathbf{C} = \mathbf{y}, \quad (10.12)$$

а її розв'язання – на основі інверсії матриці \mathbf{H} :

$$\mathbf{C} = \mathbf{H}^{-1}\mathbf{y}. \quad (10.13)$$

Проте практично цей спосіб (на основі інверсії матриці \mathbf{H}) не використовується. Реально застосовуються інші, більш ефективні способи розв'язання систем з великої кількості лінійних рівнянь, але якщо вхідні дані спотворені/зашумлені, то точна інтерполяція не має сенсу. Тому описаний вище метод необхідно модифікувати шляхом такого розв'язання системи лінійних рівнянь (10.11), яка є робастною відносно шумів (перешкод) у вхідних даних.

Простий та ефективний спосіб такого розв'язання запропонований Тихоновим А.М. і Арсеніним В.Я. [3], суть якого полягає в наступному: матриця \mathbf{H} замінюється матрицею $\mathbf{H} = \lambda\mathbf{I}$, де \mathbf{I} – одинична матриця, а λ – “невеликий” параметр. Його величина пропорційна силі шуму у вхідних даних. Коефіцієнти розв'язку за допомогою РБФ-сітки при цьому визначаються так:

$$\mathbf{C} = (\mathbf{H} + \lambda\mathbf{I})^{-1} \mathbf{y}. \quad (10.14)$$

У відповідності з (10.14) по суті розв'язується задача апроксимації, а точне розв'язання задачі інтерполяції досягається при $\lambda=0$. Якщо ж врахувати, що вхідні дані на практиці зашумлені (містять перешкоди), то розв'язок на основі (10.14) є найбільш придатним.

10.4. Узагальнені РБФ-сітки

Інтерполяція за допомогою центральних функцій та поліномів. У загальному випадку при інтерполяції за допомогою РБФ-сіток в поданні шуканої функції f додатково враховується поліноми $\mathbf{P}(\mathbf{X})$. Відповідно, загальна форма подання функції f має вигляд:

$$f(x) = \sum_{i=1}^N a_i h_i(\|x - c_i\|) + \sum_{i=1}^m d_i P_i(x), \quad m \leq N. \quad (10.15)$$

Поліноми P_i утворюють при цьому простір поліномів $\mathbf{P}_{k-1}(\mathbf{R}^n)$ у \mathbf{R} з максимальним степенем $k-1$, причому значення k задається. Для визначених центральних функцій це доповнення поліномами є необхідним для розв'язання проблеми інтерполяції. Крім того, введення поліномів сприяє згладжуванню функції f , тобто таким способом можна суттєво зменшити коливання функції між опорними точками. Між тим слід враховувати, що це не відноситься до РБФ-сіток з функціями Гауса в якості центральних функцій. Можна показати, що проблема інтерполяції є розв'язуваною і без циклічних поліномів.

У загальному випадку для розв'язання проблеми інтерполяції необхідно визначити коефіцієнти a_i та d_i , тобто $N+m$ невідомих. Умови інтерполяції $f(x_j) = y_j$, $j=1,2,\dots,N$ утворюють перші N рівнянь для визначення коефіцієнтів r_i та d_i :

$$\sum_{i=1}^N a_i h_i(\|x_j - c_i\|) + \sum_{i=1}^m d_i P_i(x_j) = y_j, \quad j=1,2,\dots,N, \quad (10.16)$$

а інші m вимог можуть вибиратися вільно. Для спрощення обчислень другий доданок у (10.16) зазвичай приймаються дорівнюваними нулю.

На основі теорії апроксимації можна стверджувати, що система рівнянь (10.16) має розв'язок, якщо h_i додатно визначені та мають порядок k в \mathbf{R}^n . Для більш простої системи з N рівнянь (10.8) центральні функції h_i додатно визначені, а, отже, система рівнянь розв'язувана. Наведемо приклади інших класів припущених центральних функцій:

$$\begin{aligned} h(z) &= \exp\left(-\left(\frac{z}{\alpha}\right)^2\right); \\ h(z) &= \frac{1}{(c^2 + z^2)^\alpha}, \quad \alpha > 0; \\ h(z) &= \sqrt{z^2 + c^2}. \end{aligned} \quad (10.17)$$

10.5. Розширення РБФ-сіток

Апроксимація за допомогою центральних функцій. Поки що розглядалося розв'язання задачі інтерполяції, коли шукана функція $f(\mathbf{X})$ повинна пройти точно через опорні точки. Вона може бути розв'язана за допомогою РБФ-сітки з радіальними базисними функ-

ціями без додаткових поліномів (рис. 10.1), яка містить один вхідний, один схований шар та один нейрон на виході. Проте, таким сіткам притаманний ряд недоліків:

- графік функцій між опорними точками може містити значні коливання, хоча умови інтерполяції виконані. Цей ефект може підсилюватися при додатковому використанні поліномів;
- точна інтерполяція при зазвичай зашумлених (спотворених) результатах вимірювання не має особливого сенсу. і тому більш обґрунтованою є апроксимація невідомої функції гладкою функцією;
- порядок лінійної системи рівнянь пропорційний кількості N застосовуваних опорних точок (образів навчаючої послідовності). Розмірність схованого шару відповідної РБФ-сітки, яка дорівнює кількості образів навчаючої послідовності, також зростає з використанням кожного нового образу. Бажано, щоб розмір сітки та витрати на розв'язання лінійної системи рівнянь були б по можливості незалежними від кількості образів або довжини навчаючої послідовності.

Вимога фіксованого розміру сітки приводить до першого поширення РБФ-сіток – так званим узагальненим РБФ-сіткам. З математичної точки зору ми стикаємося при цьому не з задачею інтерполяції, а з задачею апроксимації. При цьому не вимагається, щоб шукана функція f проходила через усі опорні точки. Необхідно лише мінімізувати сумарну похибку за всіма опорними точками при виконанні деяких вимог гладкості апроксимуючої функції.

При апроксимації першим способом за допомогою центральних функцій незалежно від кількості N образів навчаючої послідовності вибирається кількість k центрів або k схованих нейронів. Для цього, зокрема, можна використати відповідну підмножину образів навчаючої послідовності або інші опорні точки, попередньо визначені в той чи інший спосіб. Через це ці центри тепер позначаються не через \mathbf{X}_j , а через використання векторів \mathbf{t}_j (від teacher) або частіше \mathbf{W}_j . Пояснюється це тим, що координати опорних точок майже в усіх реалізаціях запам'ятовуються у вигляді ваг w_{ij} НС. При цьому, зокрема, краще виявляється схожість з іншими алгоритмами навчання.

На рис. 10.4 наведена структура узагальненої РБФ-сітки, яка містить один схований шар з фіксованою кількістю k нейронів. Дані опорних точок запам'ятовуються у вигляді ваг w_{ij} схованих нейронів.

Для k різних центрів апроксимуюча функція f приймає вигляд:

$$f(x) = \sum_{j=1}^k a_j h_j(\|x_i - w_j\|). \quad (10.18)$$

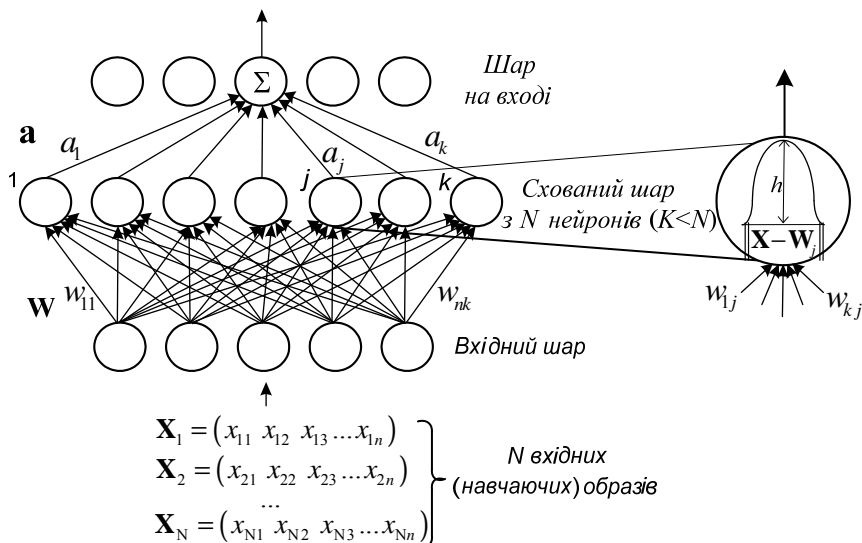


Рис. 10.4. Структура узагальненої РБФ-сітки

Для визначення k коефіцієнтів a_1, a_2, \dots, a_k маємо N рівнянь. При цьому можна використати умови інтерполяції

$$\sum_{j=1}^k a_j h_j(\|x_i - w_j\|) = y_i \quad (10.19)$$

для усіх N образів навчаючої послідовності.

Позначивши

$$H = \begin{pmatrix} h_1(\|x_1 - w_1\|) & \dots & h_k(\|x_1 - w_k\|) \\ h_1(\|x_N - w_1\|) & \dots & h_k(\|x_N - w_k\|) \end{pmatrix} \quad (10.20)$$

проблему інтерполяції можна представити так:

$$\mathbf{H}\mathbf{C} = \mathbf{y}. \quad (10.21)$$

У випадку $N > k$ ця система перевизначена і не має точного розв'язку (інтерполяція). Шляхом апроксимації можна мінімізувати суму квадратів похибок за всіма образами навчаючої послідовності, що приведе до розв'язку:

$$\mathbf{C} = \mathbf{H}^+ \mathbf{y}, \quad (10.22)$$

де \mathbf{H}^+ – псевдоінверсна матриця Мура-Пенроза відносно матриці \mathbf{H} :

$$\mathbf{H}^+ = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T. \quad (10.23)$$

У відповідності з теоремою Мічеллі інверсна матриця існує у тих випадках, коли похідна h' центральної функції монотонна, а в якості центрів w_j використовується підмножина образів \mathbf{X}_j навчаючої послідовності.

Варіаційне обчислення для розв'язання проблеми апроксимації за допомогою РБФ-сіток. Другий спосіб розв'язання проблеми апроксимації базується на варіаційному обчисленні, для чого формується функціонал:

$$H(f) = \sum_{i=1}^N (y_i - f(c_i))^2 + \lambda \|pf\|^2. \quad (10.24)$$

Задача зводиться до визначення функції f , яка мінімізує функціонал $H(f)$. Сума у цьому функціоналі містить умови інтерполяції для усіх образів \mathbf{X}_i навчаючої послідовності $i = (1, 2, \dots, N)$. Чим менша ця сума, тим краще функція f апроксимує образи навчаючої послідовності. Диференційний оператор p в (10.24) виконує функції стабілізатора для варіаційної задачі.

Таким чином, остаточний розв'язок проблеми апроксимації за допомогою узагальнених радіальних базових функцій може бути представлений у наступному вигляді: задані N образів навчаючої послідовності $\mathbf{X}_i \in \mathbb{R}^n$ та відповідні до них значення функцій $y_i \in \mathbb{R}$. Задані також k центрів $w_i \in \mathbb{R}^n$ базисних функцій h_i і порядок функції f , яка подається у вигляді:

$$f(x) = \sum_{j=1}^k a_j h_j(\|x - x_j\|). \quad (10.25)$$

Вектор невідомих коефіцієнтів $\mathbf{a} = (a_1, a_2, \dots, a_k)'$ визначається так:

$$\mathbf{a} = (\mathbf{H}'\mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}'\mathbf{y}, \quad (10.26)$$

де $'$ – знак транспонування матриці. В цьому випадку відкритим поки залишається питання щодо вибору центрів w_j . Часто ці центри вибирають просто у вигляді підмножини образів навчаючої послідовності, полягаючи, що воно рівномірно покриває всю область. Проте, слід враховувати, що положення центрів у вирішальній мірі визначає якість апроксимації.

Поширення на випадок багатьох функцій. Окрім цього прямого способу визначення ваг можна використовувати ітеративні процедури додаткового навчання нейросітки, якщо отримані резуль-

тати не є задовольняючими. Одна з них – це процедура алгоритму із зворотним поширенням похибки.

Метод радіальних базисних функцій дуже просто можна розповсюдити і на випадок багатьох функцій (багатовимірний випадок) $\mathbf{f} : \mathbf{R}^n \rightarrow \mathbf{R}^m$. Це рівносильне (еквівалентне) апроксимації множини з m окремих функцій: $\mathbf{f} : \mathbf{R}^n \rightarrow \mathbf{R}$ в припущенні, що використовуються ти ж самі опорні точки w_j для всіх функцій та ті ж самі образи \mathbf{X}_j навчаючої послідовності:

$$y_k = f_k(x) = \sum_{j=1}^k a_{jk} h_j(\|x - w_j\|). \quad (10.27)$$

Поширення лінійною частиною. Як зазначалося вище, при використанні певних центральних функцій необхідне уведення додаткового поліному степеня S в апроксимуючу функцію f , що забезпечує отримання оптимальних результатів. Проте для функцій Гауса, як зазначалося вище, це не є необхідним. У випадку $S=1$ (поліном першого степеня) це розширення РБФ-сітки є особливо привабливим. Функція f при цьому визначається так:

$$f(x) = \sum_{j=1}^k a_j h_j(\|x - w_j\|) + \sum_{s=1}^n d_s x_s + \theta, \quad (10.28)$$

де θ – поріг. Застосування різних базисних функцій приводить до теорії *гіпербазисних функцій* (ГБФ) (на рис. 10.5 наведений загальний вид ГБФ-сіток, зокрема, ГБФ-сітки з одним лінійним поліномом).

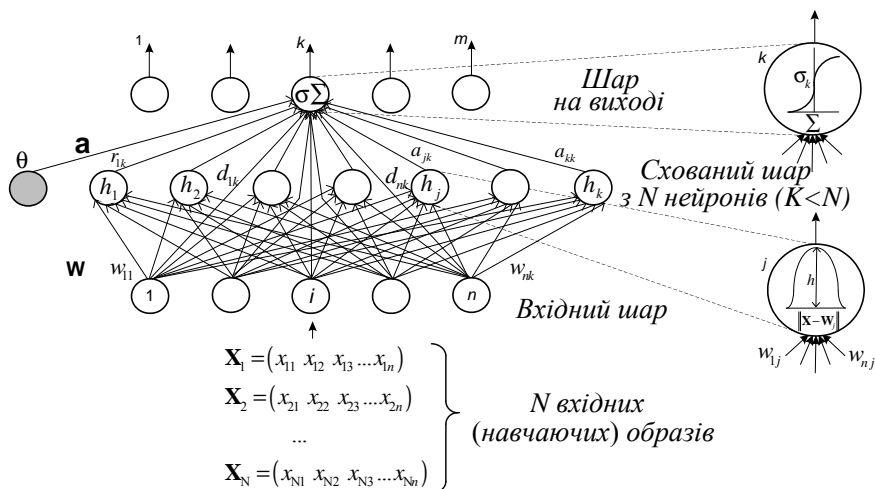


Рис. 10.5. Загальний вид ГБФ-сітки з одним лінійним поліномом

Сітки гіпербазисних функцій. При розв'язанні проблем класифікації образів часто застосовуються ШНС з сигмоїдальними функціями активації або виходу в шарі на виході. В розглядуваних досі РБФ-сітках для розв'язання задач апроксимації значення функції f (зважена сума) використовувалося в якості виходу нейрона на виході, що відповідає лінійній функції активації або виходу. Використання сигмоїдальної функції активації або виходу

$$h(x) = \left[1 + \exp \left(\frac{\|x - c_i\|}{2\sigma_i^2 - \theta} \right) \right]^{-1} \quad (10.29)$$

і в РБФ-сітках може сприяти покращенню результатів. Одне з поширень РБФ-сіток здобувається при використанні саме таких сигмоїдальних функцій активації або виходу.

Нейросітки гіпербазисних функцій уявляють собою *чергове поширення РБФ-сіток*. В них шляхом зважування аргументу $\|\mathbf{X} - \mathbf{w}_j\|$ центральної функції h забезпечується індивідуальне опрацювання кожного центра, яке може бути реалізоване різними способами:

- шляхом зважування віддалі $\|\mathbf{X} - \mathbf{w}_j\|$ деяким коефіцієнтом;
- шляхами змінювання центральної функції другим параметром P ;
- шляхом зважування окремих складових $\|\mathbf{X}_i - w_{ij}\|$, де $i = 1, 2, 3, \dots, n$ – віддалі.

10.6. Ітеративне додаткове навчання радіально- та гіпербазисних сіток

З міркувань наочності на рис. 10.5 наведені лише з'єднання для одного нейрона на виході. Лінійні складові $d_k * \mathbf{X}_k$ реалізуються безпосередньо у вигляді прямих зв'язків від входів до шару на виході. Поріг θ реалізується за допомогою додаткового нейрона з виходом 1. Інше рішення – використання зсуву (*Bias*) в нейроні на виході.

Для навчання РБФ- та ГБФ-сіток часто використовується комбінація з різних алгоритмів навчання із заохоченням і без нього. При цьому мета навчання, як звичайно, полягає в мінімізації загальної похибки сітки по всіх нейронах на виході та для усіх образів на-

навчаючої послідовності. Ця сумарна похибка сітки має вигляд:

$$E = 0,5 \sum_{k=1}^m \sum_{i=1}^N (y_{ik} - g_k(x_i))^2, \quad (10.30)$$

де m – кількість нейронів на виході (або кількість апроксимовуваних функцій); $g_k(\mathbf{X}_i)$ – дійсний вихід k -го нейрона на виході для i -го образу \mathbf{X}_i навчаючої послідовності; y_{ik} – вимаганий вихід k -го нейрона на виході для i -го образу \mathbf{X}_i навчаючої послідовності.

Вид похибки (10.30) є типовим для алгоритмів навчання нейросіток (наприклад, з тим, аби “загасити” двійку, виникаючу при диференціюванні сумарної похибки (10.30)). Виведення співвідношень ідентичний до алгоритму зворотного поширення. Для визначення змінювань ваг в процесі навчання сітки необхідні всі частинні похідні функції похибки E (10.30) за всіма вільними параметрами $a_j, c_{jk}, \theta_k, P_j$.

Вибір центрів і радіусів в РБФ-сітках. Опорні точки задають точки, через які повинна пройти апроксимуюча функція. В найпростішому випадку, як зазначалося вище, можна використовувати всі образи навчаючої послідовності для формування опорних точок і відповідно для визначення нейронів схованого шару. Проте це пов'язано зі значними витратами. Для їх зниження часто в якості центрів РБФ- і ГБФ-сіток використовується підмножина навчаючої послідовності. При цьому інколи передбачається незначне модифікування вхідних образів і ваг з'єднань («шум»). Подібний «шум» є необхідним, наприклад, у тих випадках, коли кількість k схованих нейронів перевищує кількість N образів навчаючої послідовності. Досі ми полягали, що $k \leq N$. Принципово спосіб застосовуваний і у випадку, коли $k > N$.

Для найбільш доцільного використання даних навчаючої послідовності в РБФ-сітках необхідно прийняти до уваги, що відмінності між деякими векторами навчаючої послідовності можуть бути малими або незначними. Для виявлення таких векторів можна використовувати різні методи автоматичної класифікації або кластер-аналізу. Вони дозволяють виявити кластери (групи) подібних (близьких) векторів і подати ці кластери лише одним вектором (прототипом, стереотипом, типовим представником). Досягання при цьому значне скорочення (стискання) даних – суттєве достоїнство РБФ-

сіток. В теперішній час розроблена велика кількість алгоритмів автоматичної класифікації або кластеризації. Нижче наводиться один з прикладів простих алгоритмів кластеризації на основі наступної стратегії виявлення кластерів:

- довільно вибирається 1-й елемент навчаючої послідовності, який оголошується в якості прототипу (стереотипу) 1-го кластера;
- всі вектори навчаючої послідовності, віддалі до яких від вибраного вектора не перевищує деякого порога, включаються в 1-й кластер;
- вектор навчаючої послідовності, який не може бути включеним у перший кластер, оголошується прототипом (стереотипом) 2-го кластера;
- далі процедура повторюється із залишившимися образами навчаючої послідовності доти, доки усі її образи не будуть розділені на кластери.

Ітеративний алгоритм кластеризації. Розглянемо приклад ітеративних алгоритмів кластеризації. Спочатку вибираються k різних опорних векторів \mathbf{C}_i , де $i = 1, 2, \dots, k$. Далі вектори \mathbf{X}_i навчаючої послідовності послідовно пред'являються сітці. Після кожного такого подання опорні вектори \mathbf{C}_i модифікуються у напрямку \mathbf{X}_i .

1. Ініціалізація

SELECT k опорних векторів \mathbf{C}_i

$i := 1$;

2. Правило ітерації

REPEAT

$r := 1/i$;

SELECT

$\mathbf{X}_j \mathbf{W}^T$;

$\|\mathbf{C}_i - \mathbf{X}_j\| \leq \|\mathbf{C}_k - \mathbf{X}_j\|$ для $i = 1, 2, \dots, k$, $1 \leq k \neq i \leq k$;

$\mathbf{C}_i(t+1) := \mathbf{C}_i(t) + \eta (\|\mathbf{C}_i - \mathbf{X}_j\|)$;

$i := i + 1$;

UNTIL $\mathbf{W} \leq j \leq k : \mathbf{C}_j(t+1) = \mathbf{C}_j(t)$.

Коефіцієнт корекції η зменшується із зростанням номера ітерації. Алгоритм збігається тоді, коли прототипи \mathbf{C}_i стають стабільними. При застосуванні алгоритмів кластеризації у формулі (10.6)

необхідно врахувати кількості m_i векторів в тих або інших кластерах. Модифіковані виходи розраховуються при цьому так:

$$y_i = \frac{\sum_{i=1}^k m_i h_i a_{ij}}{\sum_{i=1}^k m_i a_{ij}}. \quad (10.31)$$

Вибір параметра розсіяння σ . Параметр σ визначає розсіяння навкруги опорної точки \mathbf{C}_i . Шляхом комбінування опорних точок і параметрів σ_i бажано «покрити» весь простір образів. Поряд з вибором достатньої кількості опорних точок, які повинні бути по можливості рівномірно розподілені по простору образів, підходящий вибір значень параметрів σ_i слугує заповненню можливих порожнин.

Для визначення k найближчих сусідів в околі точки \mathbf{C}_i на практиці зазвичай застосовується *метод k найближчих сусідів*. Далі визначається середній вектор \mathbf{C}'_i . Віддаль між \mathbf{C}_i і \mathbf{C}'_i слугує потім мірою для вибору значень параметра σ_i . Для багатьох застосувань робочим правилом є вибір $\sigma = 1$.

Інший підхід до визначення оптимальної множини центральних векторів для РБФ-сіток може бути заснований на використанні *самоорганізуючих карт* (сітки Кохонена). Для цього у більшості випадків образи навчаючої послідовності попередньо нормуються за довжиною, дорівнюваною 1. Після цього випадково вибирається k з N образів навчаючої послідовності в якості початкових (стартових) значень центральних векторів і визначається скалярний добуток $\mathbf{X}'_i \mathbf{w}_j$ між вектором \mathbf{X}_i навчаючої послідовності та кожним з k центральних векторів \mathbf{w}_j , де $j = 1, 2, \dots, k$. Цей скалярний добуток є мірою схожості обох векторів, нормованих за довжиною, дорівнюваною 1. При цьому той центральний вектор \mathbf{w}_j , віддаль від якого до поточного вектора навчаючої послідовності мінімально (або для якого указаний скалярний добуток максимальний), оголошується «переможцем». Далі він зсувається на невелику величину у напрямку поточного вектора \mathbf{X}_i навчаючої послідовності:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(\mathbf{X}_i - \mathbf{w}_j(t)), \quad (10.32)$$

де α – коефіцієнт корекції. Ця процедура багаторазово повторюється по всіх образах навчаючої послідовності.

Додаткова перевага застосування самоорганізуючих карт полягає в тому, що в них враховується і близькість між нейронами схованого шару. Через це модифікується не тільки нейрон – «переможець», але й нейрони околу навкруги нього (проте в значно меншій мірі).

Розрахунок вагової матриці \mathbf{W} на виході. Після визначення опорних точок \mathbf{C}_i і параметрів розсіювання σ_i необхідно розрахувати вагову матрицю \mathbf{W} . Навчаюча послідовність містить вектори $\mathbf{V}_i = (\mathbf{X}_i, y_i)$, де \mathbf{X}_i – i -й образ навчаючої послідовності, а y_i – значення функції для нього. Частина цих векторів \mathbf{V}_i вже була використана для визначення опорних точок \mathbf{C}_i . Для визначення матриці \mathbf{W} доцільно використати алгоритм зворотного поширення. Для цього вектор \mathbf{X}_i навчаючої послідовності подається на вхід сітки, далі розраховуються активації нейронів схованого шару $h(\mathbf{X}_i)$. Вихід нейронів схованого шару представимо у вигляді вектора \mathbf{h} . Цей вектор \mathbf{h} «пропускається» через вагову матрицю

$$\mathbf{W} = (w_{ij}), \quad i = 1, 2, \dots, k, \quad j = 1, 2, \dots, m,$$

де w_{ij} – вага зв'язку нейрона i схованого шару та нейрона j шару на виході. В результаті отримується дійсний вектор \mathbf{Y}^d виходу, який порівнюється з бажаним вектором \mathbf{Y} на виході. На основі порівняння цих векторів і здійснюється корекція вагової матриці \mathbf{W} (подібно до алгоритму зворотного поширення).

Узагальнюючи, можна стверджувати, що основна відмінність РБФ-сіток полягає в такому комбінуванні радіально-симетричних функцій, яке забезпечує апроксимацію функцій з будь-яким ступенем точності.

Слід також зауважити, що РБФ-сітки володіють високою швидкістю навчання, і при їх навчанні не виникає проблем із “застряванням” у локальних мінімумах. Проте у зв'язку з тим, що при виконанні безпосередньо класифікації виконуються доволі складні обчислення, зростає час отримання результату.

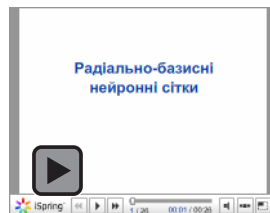
Контрольні завдання

1. Сформулювати основний висновок з теорії апроксимації та інтерполяції багатовимірних функцій Айзермана М.А., Бравермана Е.М. та Розоноера Л.І. як основи нейронних сіток радіально-базисних функцій.
2. Розкрити особливості структури радіально-базисних нейронних сіток.

3. Розкрити функції нейронів схованого (шаблонного) шару РБФ-сіток і пояснити принцип їх роботи.
4. Охарактеризувати проблему інтерполяції в РБФ-сітках.
5. Охарактеризувати проблему апроксимації в РБФ-сітках.
6. Пояснити реалізацію інтерполяції РБФ-сітками за допомогою центральних функцій та поліномів.
7. Пояснити можливості апроксимації РБФ-сітками за допомогою центральних функцій та поліномів.
8. Охарактеризувати структуру узагальненої РБФ-сітки: розкрити два способи розв'язання проблеми апроксимації – за допомогою центральних функцій та варіаційного обчислення.
9. Розкрити можливості поширення методу радіальних базисних функцій на випадок багатьох функцій.
10. Прокоментувати особливості побудовання і сфери використання нейросіток гіпербазисних функцій.
11. Навести послідовність ітеративного додаткового навчання радіально- та гіпербазисних сіток, прокоментувавши принципи вибору центрів та радіусів в сітках.
12. Розкрити особливості ітеративного алгоритму класифікації і вибору параметра розсіяння σ .
13. Навести послідовність розрахунку вагової матриці на виході РБФ-сітки.

Презентації

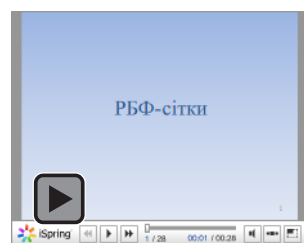
1. Радіально-базисні нейронні сітки



2. РБФ сітки



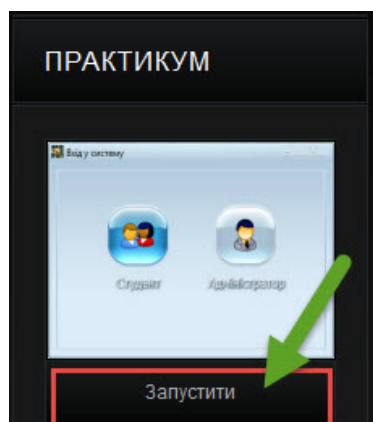
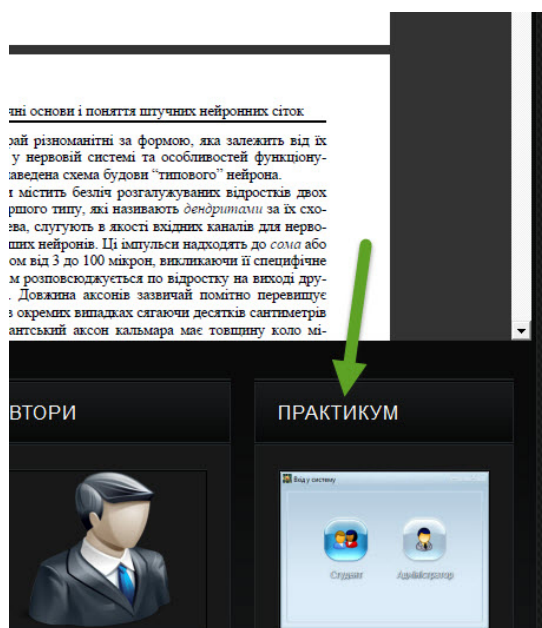
3. РБФ-сітки



Практичні заняття

Нейросітки радіально-базисних функцій

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 11

Ймовірнісні нейронні сітки

*Геній, не геній — перший не розуміє, що творить,
а другий розуміє, що він нічого не творить*
Василь Ключевський

*Зустрічаються двоє шанувальників собак, і один каже іншому:
“У мене мудрий пес, і я його назвав Мефістофелем через те,
що коли я йому кажу: Мефістофель, ти йдеш чи не йдеш?
— Він миттєво йде чи не йде”*
Анекдот

*Ймовірнісні знання —
ось межа людського розуміння*
Цицерон Марк Туллій

11.1. Ймовірнісна (стохастична) нейронна сітка

Загальна характеристика. В задачах класифікації виходи сітки можна інтерпретувати як оцінки ймовірності того, що елемент належить деякому класу, і сітка фактично вчиться оцінювати функцію щільності ймовірності. Аналогічна корисна інтерпретація може мати місце і в задачах регресії — вихід сітки розглядається як очікуване значення моделі у даній точці простору входів. Це очікуване значення пов'язане зі щільністю ймовірності спільного розподілу даних на вході та на виході.

Задача оцінювання щільності ймовірності має давню історію у математичній статистиці [178] і відноситься до області байєсової статистики. Звичайна статистика по заданій моделі розкриває нам, якою буде ймовірність того чи іншого висліджу (наприклад, що на гральній кістці шість очок буде випадати у середньому в одному випадку з шести). Байєсова статистика перевертає питання “догори ногами”: правильність моделі оцінюється по достовірним даним. У більш загальному плані байєсова статистика дає можливість оцінювати щільність ймовірності розподілення параметрів моделі по даним, які є в наявності. Для того, щоб мінімізувати помилку, вибирається модель з такими параметрами, при яких щільність ймовірності буде найбільшою.

При розв'язанні задачі класифікації можна оцінити щільність ймовірності для кожного класу, порівняти між собою ймовірності належності до відмінних класів і вибрати найбільш ймовірний. Саве

так це й відбувається, коли ми навчаємо нейронну сітку розв'язувати задачу класифікації – сітка прагне визначити (тобто апроксимувати) щільність ймовірності.

У порівнянні з лінійними методами статистики (лінійна регресія, авторегресія, лінійний дискримінант) нейронні сітки дозволяють ефективно відтворювати нелінійні залежності, які більш точно описують набори даних. З нелінійних методів класичної статистики розповсюджений, хіба що, тільки байєсівський класифікатор (див. розд. 11.2), який будує квадратичну розділяючу поверхню – а ШНС може побудувати поверхню більш високого порядку. Висока нелінійність розділяючої поверхні звичайного байєсівського класифікатора (він не використовує коваріаційні матриці класів, як класичний байєс, а аналізує локальні щільності ймовірності) потребує значної сумарної кількості прикладів для можливості оцінювання ймовірностей при кожному сполученні інтервалів значень змінних, тоді як ШНС навчається на усій вибірці даних, фрагментуючи її, що підвищує адекватність настроювання нейросітки.

При побудування нелінійних моделей (наприклад, поліноміальних) в статистичних програмах іноді необхідне ручне введення опису моделі у символьному вигляді з точністю до значень параметрів: так, при $N=10$ незалежних змінних поліном другого степеня буде містити $N \cdot (N-1)/2 = 45$ коефіцієнтів при попарних добутках змінних, 10 при самих змінних, 10 при квадратах значень змінних, тобто 66 (з урахуванням неоднорідного доданку) коефіцієнтів. При 20 змінних у вираз увійде вже 231 доданок. Введення вручну таких довгих формул вимагає тривалого часу, а крім того, зростає ризик помилкових дій при цьому. Нейронна сітка ж створюється шляхом визначення адекватної топології ШНС (див. розд. 26) – тобто базового нейрона, виду активаційної функції, структури, архітектури (кількостей шарів і нейронів в кожному шарі), що зазвичай значно швидше. А алгоритми побудування зростаючих (каскадно-кореляційних – див. розд. 17) ШНС і зовсім не потребує первісного задання їх розміру. Альтернативою ШНС при побудуванні складних нелінійних моделей є тільки метод групового урахування аргументів ([24-26], про який йшлося в розд. В.1) та, мабуть, машини опорних векторів (support vector machines).

Підходи щодо розв'язання задачі оцінки щільності ймовірності. Традиційний підхід до задачі полягає в тому, щоб побудувати оцінку для щільності ймовірності за наявними даними. Звичайно при цьому припускається, що щільність має деякий визначений вид (частіше за все, вид нормального розподілення). Після цього оцінюються

параметри моделі. Нормальне розподілення часто використовується тому, що тоді параметри моделі (середні і стандартні відхилення) можна оцінити аналітично. Але при цьому залишається питання щодо виправданості припущення про нормальність розподілення.

Другий підхід до оцінки щільності ймовірності заснований на ядерних оцінках [88; 178; 179; 192]. Міркування тут зводиться до наступного: той факт, що спостереження розташоване у даній точці простору свідчить про те, що у цій точці є деяка щільність ймовірності. Кластери з поруч розташованих точок указують на те, що у даній координаті простору щільність ймовірності є найбільшою. Поблизу спостереження є більша довіра до рівня щільності, а по мірі віддалення від нього довіра спадає прямує до нуля. За методом ядерних оцінок у точці, відповідній кожному спостереженню, розміщується деяка проста функція, потім всі вони складаються і в результаті отримується оцінка для загальної щільності ймовірності. Частіше за все в якості ядерних функцій беруться гаусові функції (з формою дзвону). Якщо навчаючих прикладів достатня кількість, то такий метод дає досить непогане наближення до істинної щільності ймовірності.

Метод апроксимації щільності ймовірності за допомогою ядерних функцій багато в чому схожий на метод РБФ, і таким чином ми природно приходимо до понять *ймовірнісної нейронної сітки* (ЙНС) та *узагальнено-регресійної нейронної сітки* (УРНС) [192; 193]. Ймовірнісні сітки призначені для задач класифікації, а узагальнено-регресійні сітки - для задач регресії.

Означення 11.1. *Ймовірнісні нейронні сітки* – вид НС для задач класифікації, де щільність ймовірності належності класам оцінюється за допомогою ядерної апроксимації. Це один з видів так званих байєсівських сіток.

Сітка ЙНС має не менше трьох шарів: вхідний, радіальний і на виході. Радіальні елементи беруться з умови – по одному на кожне навчаюче спостереження. Кожний з них уявляє гаусову функцію з центром у цьому спостереженні. Кожному класу відповідає один елемент на виході, який з'єднаний зі всіма радіальними елементами, що відносяться до його класу, а зі всіма іншими радіальними елементами він має нульове з'єднання. Таким чином, елемент на виході просто складає відгуки усіх елементів, належних до його класу. Значення сигналів на виході стають пропорційними ядерним оцінкам ймовірності належності відповідним класам, і пронормувавши їх на одиницю, ми отримуємо остаточні оцінки ймовірності належності до класів.

Базова модель ЙНС-сітки може мати дві модифікації. У *першому випадку* припускається, що пропорції класів у навчаючій множині відповідають їх пропорціям у всієї досліджуваної популяції (або так званим апіорним ймовірностям). Наприклад, якщо серед усіх людей хворими є 2%, то у навчаючій множині для сітки, діагностуючій захворювання, хворих теж має бути 2%. Якщо ж апіорні ймовірності будуть відрізнятися від пропорцій у навчаючій вибірці, то буде видавати невірний результат. В подальшому це можна буде урахувати (якщо стали відомі апіорні ймовірності), уводячи поправкові коефіцієнти для відмінних класів.

Другий варіант модифікації заснований на наступній ідеї. Будь-яка оцінка, яка формується сіткою, базується на зашумлених даних і неминуче буде приводити до окремих похибок класифікації (наприклад, результати аналізів у деяких хворих можуть бути цілком задовільними). Іноді доцільно рахувати, що деякі види похибок обходяться “дорожче” від інших (наприклад, якщо здорова людина буде діагностована як хвора, то це викличе зайві витрати на його обстеження, проте не створить загрози для життя; якщо ж не буде виявлений дійсно хворий, то це може привести до смертельного наслідку). У такій ситуації ті ймовірності, які видає сітка, слід домножити на коефіцієнти втрат, відбиваючи відносну оцінку похибок класифікації. З цією метою в ЙНС може бути доданий четвертий шар, який містить матрицю втрат. Вона умножається на вектор оцінок, отриманий в третьому шарі, після чого у якості відповіді береться клас, який має найменшу оцінку втрат. До речі, матрицю втрат можна додавати і до інших видів нейросіток, розв'язуючих задачі класифікації.

Ймовірнісні нейронні сітки мають єдиний керуючий параметр навчання (його значення вибирається користувачем) – ступінь згладжування/відхилення гаусової функції. Як і у випадку РБФ-сіток, цей параметр вибирається з міркувань, щоб капелюхи “визначену кількість разів перекривалися”: вибір занадто малих відхилень приведе до “гострих” апроксимуючих функцій і неспроможності сітки до узагальнень, а при занадто великих відхиленнях будуть втрачатися деталі. Вимагане значення неважко знайти дослідним шляхом, підбираючи його так, щоб контрольна похибка була якомога меншою. На щастя, ЙНС не дуже чутливі до вибору параметра згладжування.

Ймовірнісна нейросітка не потребує навчання у тому сенсі, який вимагається для сіток зі зворотним поширенням похибок через те, що усі параметри ЙНС (кількість елементів і значення ваг) визначаються безпосередньо навчальними даними.

Найбільш важливі переваги ЙНС-сіток полягають у тому, що значення на виході має ймовірнісний сенс (і тому його легше інтерпретувати), а також у тому, що сітка швидко навчається. При навчанні такої сітки час витрачається практично тільки на те, щоб подавати на її вхід навчаючі спостереження, і сітка працює настільки швидко, наскільки це взагалі можливо.

Суттєвим недоліком таких сіток є їх обсяг: ЙНС фактично вміщує в себе усі навчаючі дані, через що вона потребує забагато пам'яті і може повільно працювати. Отже ЙНС-сітки дуже вимогливі у відношенні ресурсів, і розв'язання деяких проблем вимагає сотень і навіть тисяч навчальних зразків, в результаті чого класифікація кожного невідомого екземпляра вимагатиме чимало часу. Проте слід пам'ятати, що якщо сітка реалізована у вигляді апаратних засобів, то обчислення частіше за все виконуються паралельно.

Ймовірнісні нейросітки особливо корисні при апробаційних експериментах (наприклад, коли необхідно вирішити, які з вхідних змінних використовувати), бо дякуючи “короткому” часу навчання можна швидко передати велику кількість пробних тестів.

Властивості стохастичних нейронних сіток. Хоча данні сітки і називаються стохастичними (probabalistic), функціонування їх, на відміну від машини Больцмана, є детермінованим. Основним їх призначенням є *стохастичне моделювання*.

Стохастичні сітки (СС), введені Дональдом Спехтом (*Donald Specht*), володіють цілим рядом нетривіальних властивостей, що робить їх особливо привабливими для розв'язання прикладних задач розпізнавання образів і класифікації. До ряду таких властивостей відносяться:

- велика швидкість навчання – досягається за рахунок відмови від застосування ітеративного оптимізаційного процесу налаштування внутрішніх параметрів;
- оптимальність, яка полягає в тому, що при наявності достатнього для конкретної задачі обсягу даних отримане рішення співпадає з рішенням, яке дається байєсівським класифікатором;
- можливість отримання ефективного рішення за наявності пропущених або зашумлених даних.

Обмежені до недавнього часу можливості обчислювальної техніки (малий обсяг пам'яті та невелика швидкодія) були причиною недостатньої уваги до даних сіток. Бурхливий розвиток інформаційних технологій дав новий поштовх як до дослідження СС, так і розширення сфер їх застосування.

Запропонована Спехтом сітка моделювала роботу байєсівського класифікатора, який забезпечує мінімум середнього ризику класифікації [193].

11.2. Байєсівський класифікатор

Властивості байєсівського класифікатора. Робота класифікатора може бути представлена наступним чином. Нехай заданий вхідний N -вимірний вектор $\mathbf{X} = (X_1, X_2, \dots, X_N)^T$, що описує об'єкти, які можуть бути віднесені до одного з k класів, які характеризуються відповідно функціями розподілення ймовірностей $f_1(X), f_2(X), \dots, f_k(X)$. Нехай p_1, p_2, \dots, p_k – апіорні ймовірності приналежності вектора \mathbf{X} до відповідного класу. Необхідно задати деякий критерій класифікації (функцію) $d(X) = C_i | i = \overline{1, k}$, що характеризує правильність віднесення вхідного вектора \mathbf{X} до одного з заданих класів $C_i | i = \overline{1, k}$. В якості такого можна взяти, наприклад, критерій мінімального ризику або ж вартісний критерій, який визначає штраф за неправильне рішення. Нехай якимось чином задані функції втрат L_1, L_2, \dots, L_k , що характеризують помилкове віднесення пред'явленого вектора до одного з класів. При цьому функція втрат L_i відповідає випадку віднесення вектора \mathbf{X} до класу i , в той час як насправді він відноситься до класу j , тобто L_i відповідає $d(X) = C_i | X \in C_j | i \neq j$. Природно, при правильній класифікації $L = 0$.

Байєсівський класифікатор оснований на порівнянні величин $p_1 L_1 f_1(X), p_2 L_2 f_2(X), \dots, p_k L_k f_k(X)$, а вхідний сигнал відносить до того класу, для якого дана величина буде найбільшою. Таким чином, вектор \mathbf{X} буде віднесений до класу C_j , якщо

$$p_1 L_1 f_1(X) > p_j L_j f_j(X) \text{ для } j = \overline{1, k}, i \neq j. \quad (11.1)$$

Використання наведеного правила припускає, що всі співмножники добутоків $p_i L_i f_i(X)$ відомі. На практиці ж справа є трохи інакшою. Якщо апіорні ймовірності $p_i | i = \overline{1, k}$ або відомі, або можуть бути достатньо просто оцінені, то функції розподілу ймовірностей $f_i(X) | i = \overline{1, k}$, як правило, не відомі, а їх оцінювання представляє досить складну задачу. Звичайно для спрощення припускають, що вони є нормальними і оцінюють параметри цих розподілів. Більш раціональним уявляється застосування непараметричних методів оцінювання, зокрема, вікон (методу) Парзена [63; 178].

Вікна Парзена. В нейронних сітках вікно Парзена - це альтернативна назва методу ядерної апроксимації [178]. Вікна Парзена застосовуються для побудови сімейства оцінок, заданих у загальному випадку відношенням

$$f_n(x) = \frac{1}{n\lambda} \sum_{i=1}^n \varphi\left(\frac{x-x_i}{\sigma}\right). \quad (11.2)$$

Тут x_i – незалежні випадкові змінні, які мають ідентичні неперервні функції розподілу; $\varphi(y)$ – функції, які задовольняють умову

$$\int_{-\infty}^{\infty} |\varphi(y)| dy < \infty; \quad \lim_{y \rightarrow \infty} |y\varphi(y)| = 0; \quad \int_{-\infty}^{\infty} \varphi(y) dy = 1; \quad (11.3)$$

де $\sigma = \sigma(n)$ – функція, що задовольняє вимоги $\lim_{n \rightarrow \infty} \sigma(n) = 0$; $\lim_{n \rightarrow \infty} n\varphi(n) = \infty$.

Власне Парзен і показав, що оцінка (11.2) асимптотично збігається до розподілу, що лежить в її основі, якщо він є гладким та неперервним.

Графічно це може бути проілюстровано наступним чином (рис. 11.1).

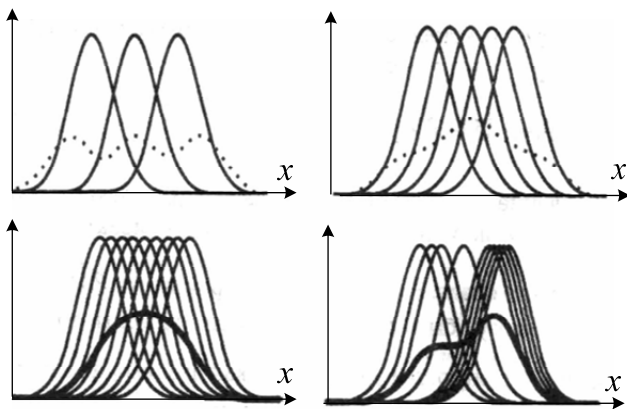


Рис. 11.1. Вікна Парзена

На осі відкладаються значення випадкової величини x . Для кожного значення x будується функція густини (наприклад, гаусівська, як на наведеному рис. 11.1), максимум якої досягається при даному значенні x . Усереднення кривих дає нову криву – усереднену функцію густини, яка із зростанням значень x , а відповідно і функцій густини, може скільки завгодно точно апроксимувати дійсну функцію густини.

Результат Парзена був узагальнений Т. Какулосом [63] на багатовимірний випадок. Для найпоширенішого на практиці багатовимірного нормального розподілу оцінка (11.2) приймає вигляд:

$$f_n(x) = (2\pi\sigma^2)^{-\frac{n}{2}} \frac{1}{n} \sum_{i=1}^n \exp\left(-\frac{(x-x_i)^T(x-x_i)}{2\sigma^2}\right) \quad (11.4)$$

Стосовно до задачі класифікації оцінка може бути записана наступним чином:

$$f_i(x) = (2\pi\sigma^2)^{-\frac{n}{2}} k_i^{-1} \sum_{j=1}^{k_i} \exp\left(-\frac{(x-x_i)^T(x-x_{ij})}{2\sigma^2}\right) \quad (11.5)$$

де n – розмірність випадкових змінних; k_i – кількість точок образів класу C_i ; x_{ij} – це j -й образ, який відноситься (належить) до i -го класу.

В оцінці (11.5) σ^2 являє собою параметр згладжування, що визначається експериментально. Зауважимо, що величина σ^2 впливає на ступінь оцінки Парзена (див. рис. 11.2), однак сама оцінка не дуже чутлива до цієї величини. Як впливає з рис. 11.2, збільшення σ веде до згладжування розділяльних гіперповерхонь. У більшості випадків цей параметр вибирають в межах $2 \leq \sigma \leq 6$ [192].

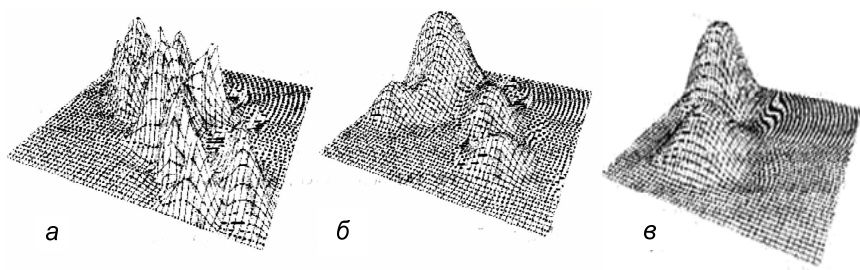


Рис. 11.2. Вплив величини σ^2 на вигляд оцінки Парзена: відповідає малому (а), збільшеному (б), ще більшому значенню σ^2 (в)

11.3. Структура стохастичної сітки

Запропонована Спехтом структура *стохастичної інтелектуалізованої нейронної сітки* (СІНС) (рис. 11.3) являє собою аналог байесівського класифікатора, що відновлює на основі формули (11.5) розподіл ймовірностей заданих класів образів. Застосування такого підходу дозволяє будувати нелінійні розділяючі поверхні, що забезпечують високу якість класифікації.

Кількість нейронів вхідного шару відповідає розмірності вхідного векторного простору.

Усі вектори, які пред'являються сітці, попередньо нормалізуються. Кількість нейронів шару аналізу образів відповідає кількості самих образів.

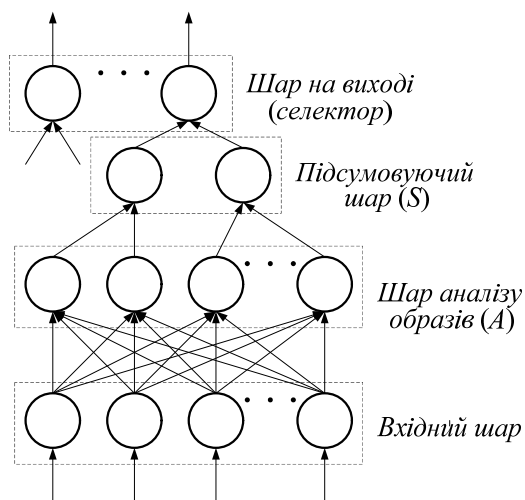


Рис. 11.3. Структура СІНС

У режимі навчання встановлюються та запам'ятовуються елементи вагової матриці \mathbf{W} . Так, якщо $X_1 \in C_i$ (1-й елемент множини образів належить класу C_i), то по закінченню режиму навчання маємо $W_1 = X_1$. Оскільки вектор \mathbf{X} нормалізований, то вектор \mathbf{W} буде також нормалізованим. Після цього встановлюється зв'язок між 1-м нейроном шару аналізу образів a_1 і j -м нейроном підсумовуючого шару S_j . Таким чином, в сутності процес навчання сітки полягає в присвоєнні відповідних значень вхідних сигналів елементам вагової матриці \mathbf{W} і встановленні зв'язків між нейронами шару аналізу образів та підсумовуючого шару.

В режимі класифікації на вхід сітки подається сигнал X і проводиться активація нейронів шару аналізу образів. При цьому обчислюється зважена сума

$$Z_j = \sum_{i=1}^N X_i W_{ij}. \quad (11.6)$$

При активації нейрону шару аналізу образів обчислюється величина

$$f(Z_j) = \exp\left\{\frac{Z_j - 1}{\sigma^2}\right\}. \quad (11.7)$$

Кількість нейронів підсумовуючого шару відповідає кількості роздільних класів. Нейрони шару аналізу образів зв'язуються з нейронами підсумовуючого шару не довільно, а в залежності від того, до якого класу відноситься даний образ.

Нейрони підсумовуючого шару просто підсумовують, у від-

повідності з (11.5), подання на їх входи m ($0 \leq m \leq M$, де M – кількість нейронів шару аналізу образів) сигналів (11.7), тобто кожний нейрон S_k даного шару обчислює

$$S_k = \sum_{p=1}^m f_p(Z_j). \quad (11.8)$$

А оскільки сигнали $f_p(Z_j)$ відносяться до одного класу, то отримані в результаті підсумовування вхідні сигнали нейрону даного шару будуть оцінками ймовірностей розподілу відповідних класів.

У загальному випадку вихідний шар представляє собою селектор, що вибирає нейрон підсумовуючого шару з максимальним значенням вихідного сигналу і відносить його до відповідного класу.

Приклад 11.1. При розв'язанні задачі дихотомії, тобто розбиття образів на два класи, нейрон вихідного шару має два вхідних сигнали, які поступають з підсумовуючого шару. Дані нейрони обчислюють добутки вихідних сигналів підсумовуючого шару та вагових коефіцієнтів C_k | $k = 1, 2$, які визначаються за формулою

$$C_k = \frac{p_2 L_2}{p_1 L_1} \times \frac{k_1}{k_2},$$

де k_1, k_2 – кількість образів, які належать першому та другому класам відповідно.

Якщо $C_1 > C_2$, пред'явлений образ відносять до першого класу, в протилежному випадку – до другого. Архітектура ЙНС для розв'язання задачі дихотомії наведена на рис. 11.4.

У цьому випадку підсумовуючий шар містить 2 нейрони, а вихідний – один (селектор).

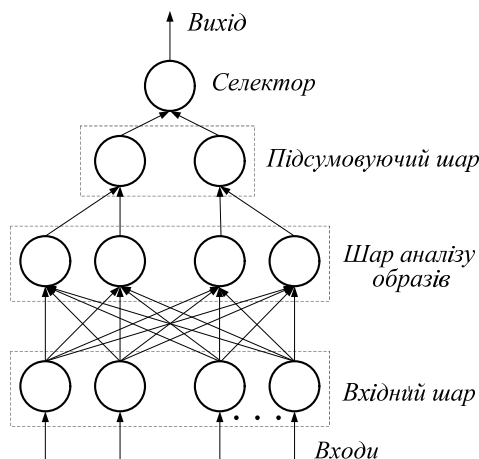


Рис. 11.4. Приклад СІНС для розв'язування задачі дихотомії

11.4. Узагальнено-регресійна нейронна сітка

Математичне визначення регресії. Взагалі *регресійний аналіз* є методом моделювання вимірюваних даних та дослідження їх властивостей. Дані складаються з пар значень *залежної змінної* (змінної відгуку) і *незалежної змінної* (пояснюючої змінної).

Означення 11.2. *Регресія* – це залежність математичного очікування (наприклад, середнього значення) випадкової величини від однієї або декількох інших випадкових величин (вільних змінних).

Твердження 11.1. *Задача регресії* – поодинокий випадок задачі апроксимації функції з неперервними вхідними значеннями.

Звичайно для НС задача регресії ставиться так: знайти найкраще наближення функції, заданої скінченим набором вхідних значень (навчаючих прикладів). Наприклад, НС дозволяють розв'язувати задачу відновлення закономірностей.

Твердження 11.2. *Головною задачею регресійного аналізу* є визначення математичних моделей досліджуваних об'єктів або явищ на основі експериментів або спостережень [51].

Ці моделі уявляють собою певні математичні співвідношення між деякими характеристиками y_1, y_2, \dots, y_M спостережуваного об'єкта або явища і обумовлюючими їх величинами x_1, x_2, \dots, x_N , які відповідно називаються залежними (на виході) і незалежними (вхідними) змінними.

Означення 11.3. *Математична модель досліджуваного об'єкта або явища* – співвідношення умовного, через наявності у спостережень випадкових збурень і відбиття досліджуваного явища з деяким наближенням, математичного очікування залежної змінної при заданих значеннях незалежних змінних:

$$f(x_1, x_2, \dots, x_N) = E\{y | x_1, x_2, \dots, x_N\}, \quad (11.9)$$

яке й дістало назву *регресії*, або *рівняння регресії у загальному виді*. Це означає, що вимірювана змінна y на виході містить деяку заваду ζ , і тоді вираз (11. 9) приймає вид:

$$y = f(x_1, x_2, \dots, x_N) + \zeta. \quad (11.10)$$

Графік функції (11.10) є *лінією регресії* y за x_1, x_2, \dots, x_N або *рівнянням регресії*.

Таким чином, припускається, що залежна змінна y є сумою значень деякої моделі $f(x_1, x_2, \dots, x_N)$ і випадкової величини ζ . Відно-

сно характеру розподілення цієї величини робляться припущення, які називаються *гіпотезою породження даних*. Для підтвердження або спростування цієї гіпотези виконуються статистичні тести, які дістали назву *аналізу залишків*.

Регресійний аналіз використовується для *прогнозування, аналізу часових рядів, тестування гіпотез та виявлення схованих взаємозалежностей в даних*.

Існує велика кількість регресійних моделей, які визначаються видом функції $f(x_1, x_2, \dots, x_N)$ та значеннями використовуваних в них коефіцієнтів v_1, v_2, \dots, v_l , які визначаються за експериментальними даними. В залежності від того, чи входять ці коефіцієнти у рівняння регресії лінійно або нелінійно, розрізняють відповідно *лінійні* або *нелінійні* за параметрами моделі.

В класичному регресійному аналізі на практиці лінія регресії найчастіше шукається у виді $y = v_0 + v_1 x_1 + v_2 x_2, \dots, v_N x_N$ лінійної функції, найкращим чином наближаючої шукану криву. Оцінювання вектора параметрів v (*параметричне оцінювання*) здійснюється шляхом мінімізації деякого наперед вибраного опуклого (звичайно, квадратичного) критерія оцінювання. Критерієм якості наближення (цільовою функцією) звичайно є середньоквадратична похибка: сума квадратів різниці значень моделі і залежної змінної для всіх значень незалежної змінної в якості аргументу. Інакше кажучи, основним методом оцінювання в лінійній регресії є *метод найменших квадратів* (МНК), коли мінімізується сума квадратів відхилень реально спостережуваних y від їх оцінок \hat{y} (маються на увазі оцінки за допомогою прямої лінії, претендуючої на представлення шуканої регресійної залежності):

$$\sum_{i=1}^M (y_k - \hat{y}_k)^2 \rightarrow \min, \quad (11.11)$$

де M – об'єм вибірки.

Цей підхід заснований на відомому факті, що у наведеному виразі (11.11) сума приймає мінімальне значення саме для випадку, коли $y = f(x_1, x_2, \dots, x_N)$.

Задача параметричного оцінювання у нелінійній регресії суттєво ускладнюється через те, що оцінка МНК для такої регресії може не існувати взагалі, а може існувати декілька оцінок МНК, які при-

водять до того ж самого значення мінімізованого критерія.

Параметри v_i є частковими коефіцієнтами кореляції; $(v_i)^2$ інтерпретується як частка дисперсії y , відповідна x_i при фіксації впливу інших предикторів, тобто вимірює індивідуальний внесок x_i у пояснення y . У випадку корелюючих предикторів виникає проблема невизначеності в оцінках, які стають залежними від порядку включення предикторів у модель. У таких випадках необхідне застосування методів аналізу кореляційного та покрокового регресійного аналізу.

Саме використання вікон Парзена дозволяє ефективно розв'язувати задачу побудування *нелінійної регресійної моделі*.

Модель функціонування УРНС [48]. Нехай $f(x,y)$ – спільна щільність розподілення випадкового вектора x та величини y . Регресія за визначенням уявляє собою умовне середнє:

$$E\{y | x\} = \frac{\int_{-\infty}^{\infty} y f(x, y) dy}{\int_{-\infty}^{\infty} f(x, y) dy}. \quad (11.12)$$

Для оцінки невідомої функції $f(x,y)$ можна застосувати різні методи. Однак одним з найбільш ефективних є непараметричне оцінювання Парзена

$$f_k(x, y) = (2\pi\sigma^2)^{-\frac{N}{2}} k^{-1} \sum_{i=1}^k \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|y - y_i\|^2}{2\sigma^2}\right), \quad (11.13)$$

де k – кількість образів; N – розмірність вектора i -го образу x_i ; σ – згладжуючий параметр.

Зі зростанням кількості k образів точність оцінки (11.13) зростає.

Використання у (11.12) замість $f(x,y)$ оцінки (11.13) приводить до наступної оцінки регресії [192]:

$$\hat{y} = \frac{\sum_{i=1}^k y_i \exp\left(-D_i^2 (2\sigma^2)^{-1}\right)}{\sum_{i=1}^k \exp\left(-D_i^2 (2\sigma^2)^{-1}\right)}, \quad (11.14)$$

де $D_i^2 = \|x - x_i\|^2$.

Запропонована Спехтом УРНС розв'язує задачу побудування узагальнених (лінійних та нелінійних) регресій. На відміну від кла-

сичного регресійного аналізу використання даного підходу не вимагає завдання функціональної залежності, зв'язуючої вхідні змінні зі змінними на виході. Сітка видає оцінку функції розподілення ймовірностей образів, які їй пред'являються.

Функціонуванню сітки передуює процедура розбиття всіх вхідних образів яким-небудь чином на визначену кількість кластерів. Якщо кількість образів є невеликою, кожний образ може утворювати свій кластер. Далі вибирається структура УРНС і здійснюється її навчання. Через те, що вираз (11.14) описує оцінку для окремого образу, при побудуванні узагальненої регресії використовують оцінку

$$y = \frac{\sum_{i=1}^k U_i \exp\left(-D_i^2 (2\sigma^2)^{-1}\right)}{\sum_{i=1}^k Q_i \exp\left(-D_i^2 (2\sigma^2)^{-1}\right)}, \quad (11.15)$$

де $U_i \equiv U_i(k) = U_i(k-1) + y_i$; $Q_i \equiv Q_i(k) = Q_i(k-1) + 1$ уявляють собою значення коефіцієнтів для i -го кластера після k спостережень.

Структура УРНС звичайно співпадає зі структурою стохастичної сітки на рис. 11.3.

Структура узагальнено-регресійної сітки. Узагальнено-регресійна нейронна сітка побудована аналогічно до ЙНС, але вона призначена для розв'язання задач регресії а не класифікації. Копіюючи всередину себе всі навчаючі спостереження, УРНС використовує їх для оцінки відгуку у довільній точці. Рахують, що кожне спостереження свідчить про деяку нашу упевненість в тому, що поверхня відгуку в деякій конкретній точці має визначену висоту, і ця упевненість спадає при відхиленні в бік від цієї точки. Остаточна оцінка на виході сітки отримується як зважене середнє виходів за всіма навчаючими спостереженнями, де величини ваг відображають віддаль від цих спостережень до точки, в якій утворюється оцінювання (і, таким чином, внесок більш близьких точок є вагомішим). З цією метою, як і у випадку ЙНС, в точку розташування кожного навчаючого спостереження розміщується гаусова ядерна функція.

Узагальнено-регресійна НС (рис. 11. 5), як і РБФ-сітки, крім вхідного з функціями розподілу вхідного вектора має *радіально-базисний шар* з кількістю нейронів, дорівнюваною або менше кількості елементів навчаючої множини, і крім того ще містить *проміжний* (лінійний) шар.

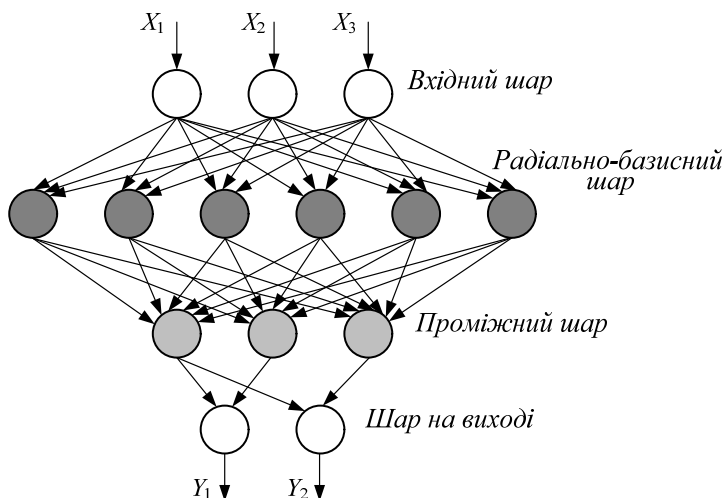


Рис. 11.5. Узагальнено-регресійна НС

Проміжний шар містить елементи, які допомагають оцінити зважене середнє за допомогою використання спеціальної процедури. Кожний вихід має у цьому шарі свій елемент, формуючий для нього зважену суму. Щоб отримати із зваженої суми зважене середнє, цю суму слід поділити на суму вагових коефіцієнтів. Останню суму обчислює спеціальний елемент другого шару. Після цього у шарі на виході виробляється власне ділення (за допомогою спеціальних елементів «ділення»). Таким чином, кількість елементів у другому проміжному шарі на одиницю більше, ніж у шарі на виході. Як правило, в задачах регресії (апроксимації) необхідно оцінити одне значення на виході, і тому другий проміжний шар містить два елемента.

Процедура функціонування УРНС полягає у наступному. Вхідні образи передаються вхідним шаром на радіально-базисний шар (аналізу образів), який містить k нейронів за кількістю кластерів. Елементи вагової матриці приймаються дорівнюваними відповідним компонентам вхідних векторів або відповідним координатам векторів центрів кластерів. При надходженні до сітки вхідного образу x обчислюється віддаль між цим образом і вектором вагових коефіцієнтів сітки, які є аргументом активаційної функції нейронів радіально-базисного шару. Звичайно як функція активації застосовується експоненційна (гаусівська) функція. В цьому випадку при застосуванні квадратичної функції відхилення ваг і входів сітки сигнали на виході

нейронів радіально-базисного шару відповідають (11.7). Далі ці сигнали надходять на проміжний (підсумовуючий) шар, який містить нейрони двох типів.

Нейрони одного типу використовуються для подання “бажаної” регресії, і на їх виходах формується сигнал $\sum f(x)\chi$, де \sum – оцінка середнього значення y для заданого $x(E\{y|x\})$; $f(x)$ – функція щільності x ; χ – деякий коефіцієнт, який залежить від використовуваних вікон Парзена.

Ваги нейронів *другого типу* дорівнюють кількості образів в одному кластері. У цих нейронах підсумовуються $f(x,y)k$. В шарі на виході здійснюється ділення сигналів на виході нейронів першого типу на сигнали на виході нейронів другого типу, в результаті чого формується оцінка регресії (11.14) або (11.15).

Процес навчання УРНС аналогічний навчанню РБФ-сітки і відзначається простотою та високою швидкістю, бо в даному випадку не використовується потактова корекція (рекурентна налагодка) елементів вагової матриці, а сітка будує оцінку регресії на підставі подаваних образів. При цьому першочергово налагоджуються центри базисних функцій (11.8), потім з фіксованими параметрами РБФ-нейронів навчається шар на виході.

Модифікація УРНС. Можна модифікувати УРНС таким чином, щоб радіальні елементи відповідали не окремим навчаючим випадкам, а їх кластерам. Це зменшує розмірність сітки та збільшує швидкість навчання. Центри для таких елементів можна вибирати за допомогою будь-якого призначеного для цієї мети алгоритму (вибірки з вибірки, K -середніх або Кохонена), і програма *ST Neural Networks* відповідним чином коректує внутрішні ваги.

Достоїнства і недоліки в УРНС в основному ті ж самі, що й у ЙНС – єдина різниця полягає в тому, що УРНС використовуються в задачах регресії, а ЙНС – в задачах класифікації. Навчається УРНС майже миттєво, проте може бути великою і повільною (хоча тут, на відміну від ЙНС, не обов'язково мати по одному радіальному елементові на кожний навчаючий приклад, їх кількість все рівно буде великою). Як і у випадку РБФ-сітки, УРНС не володіє спроможністю екстраполювати дані.

Лінійна УРНС. Відповідно до загальноприйнятого у науці принципу, якщо більш складна модель не дає кращих результатів у порівнянні з простою, то з них слід віддавати перевагу другій. В тер-

мінах апроксимації відображень найпростішою моделлю буде лінійна, в якій підгонкова функція визначається гіперповерхнею. В задачі класифікації гіперповерхня розташовується таким чином, щоб вона розділяла собою два класи (лінійна дискримінантна функція); в задачі регресії гіперповерхня повинна проходити через задані точки. Лінійна модель звичайно записується за допомогою матриці $N \times N$ і вектора зсуву розмірністю N .

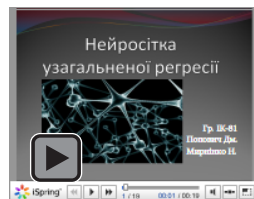
На мові НС лінійна модель подається сіткою без проміжних шарів, яка в шарі на виході містить тільки лінійні елементи (тобто елементи з лінійною функцією активації). Ваги відповідають елементам матриці, а пороги – компонентам вектора зсуву. Під час роботи сітка фактично перемножує вектор входів на матрицю ваг, а потім до отриманого вектора додає вектор зсуву.

Контрольні завдання

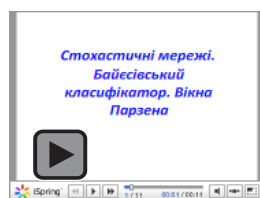
1. Дати визначення ймовірнісної нейронної сітки та охарактеризувати їх властивості та особливості структурної та архітектурної побудови.
2. Прокоментувати властивості байєсівського класифікатора.
3. Сформулювати метод ядерної апроксимації як альтернативу вікон Парзена в нейронних сітках.
4. Прокоментувати структуру стохастичної інтелектуалізованої нейронної сітки.
5. Пояснити принципи розв'язання задачі дихотомії за допомогою стохастичної інтелектуалізованої нейронної сітки.
6. Дати математичне визначення регресії і розкрити задачу регресійного аналізу.
7. Охарактеризувати структуру узагальнено-регресійної нейронної сітки.
8. Охарактеризувати процес навчання узагальнено-регресійної нейронної сітки.

Презентації

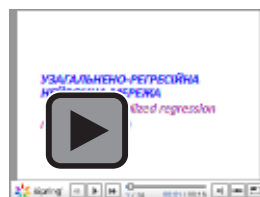
1. Нейросітка узагальненої регресії



2. Стохастичні мережі. Байєсівський класифікатор. Вікна Парзена



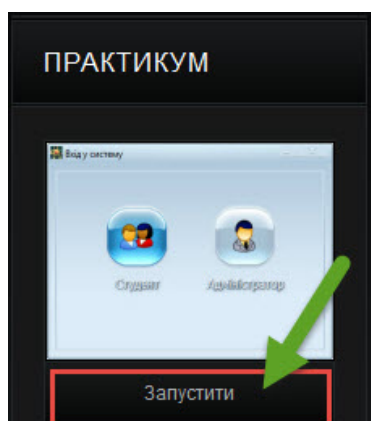
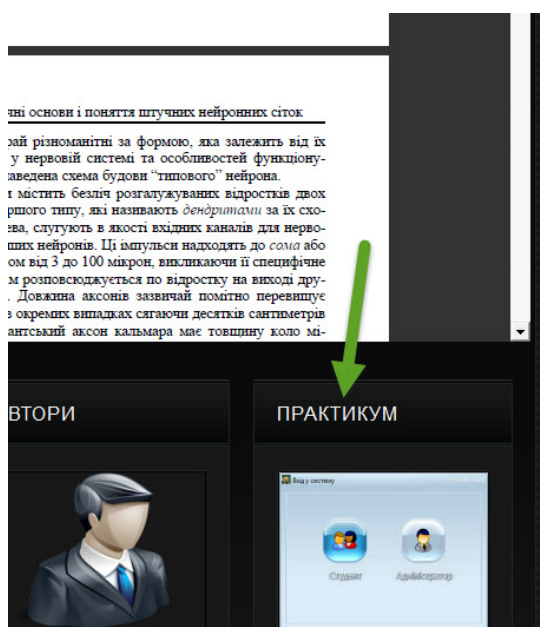
3. Узагальнено-регресійна нейронна мережа (GRNN - generalized regression neural network)



Практичні заняття

Ймовірнісні нейронні сітки

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 12

Динамічні рекурсивні нейронні сітки

Краще прийти і почекати, а ніж бігти і не встигнути
Житейська мудрість

Касир пішла на обід. Буде, коли повернеться
Оголошення

12.1. Узагальнення структури динамічних рекурсивних сіток

Загальні зауваження. На практиці зазвичай нейросітки застосовують для опрацювання деяких послідовностей: звукових хвиль, відеопослідовностей, даних з вимірювальних пристроїв тощо. Для пред'явлення нейросітці дані розбиваються на блоки і подаються до ШНС. Така модель часто працює, але володіє недоліком – звичайний багатошаровий перцептрон розглядає кожний блок незалежно від іншого, не враховуючи можливий зв'язок між ними. Для опрацювання послідовностей це може виявитись суттєвим недоліком.

Відсутність зворотного зв'язку гарантує безперечну стійкість сіток. Вони не можуть увійти в режим, коли вихід безперервно блукає від стану до стану і є непридатним до використання. Проте ця дуже бажана властивість досягається “не безкоштовно”, НС без зворотних зв'язків мають більш обмежені можливості у порівнянні з сітками зі зворотними зв'язками.

Через те, що ШНС зі зворотними зв'язками мають шляхи, які передають сигнали від виходів до входів, *відгук таких сіток є динамічним*, тобто після прикладання нового входу обчислюється вихід і, передаючись по ланцюгу зворотного зв'язку, модифікує вхід. Саме ця особливість функціонування ШНС зі зворотними зв'язками обумовила і назву цього класу сіток як *динамічні рекурсивні нейронні сітки* (ДРНС). Далі вихід знов обчислюється, і процес повторюється знову і знову. Для стійкої ДРНС послідовні ітерації приводять до все менших змін виходу, поки врешті решт вихід не стає постійним. Для багатьох ШНС процес ніколи не закінчується, і такі сітки називають *нестійкими*. Нестійкі ШНС володіють цікавими властивостями і вивчалися як приклад хаотичних систем.

Означення 12.1. Багатошарові рекурсивні сітки (БРС) являють собою розвиток односпрямованих сіток перцептронного типу за рахунок додавання в них відповідних зворотних зв'язків (рис. 12.1, рис. 12.2).

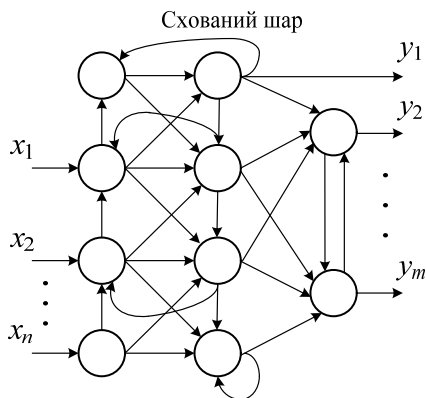


Рис. 12.1. Приклад структури ДРНС

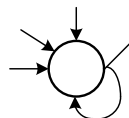


Рис. 12.2. Локальні зв'язки нейрона

Відмінність ДРНС від сіток Хопфілда полягає в наступному:

- рекурсивна сітка може мати декілька шарів (рис. 12.1);
- нейрони (рис. 12.2) можуть мати власні зворотні зв'язки ($w_{ij} \neq 0$);
- матриця ваг може бути несиметричною;
- може здійснюватися контрольоване навчання шляхом використання алгоритму зворотного поширення.

Перевага забезпечується наявністю зворотних зв'язків між нейронами різних шарів, в тому числі, і нейронами шару на виході. Крім того, в даній сітці нейронами на виході можуть бути будь-які зі складу НС (наприклад, на рис. 1 один з нейронів схованого шару є одночасно і нейроном на виході – вихід y_1).

Наявність зворотних зв'язків між нейронами різних шарів, в тому числі і шару на виході, забезпечує ДРНС додаткові позитивні властивості, які неможливо досягти у статичних багат шарових сітках прямого поширення. З точки зору програмування в сітках зі зворотними зв'язками з'являється аналог циклічного виконання, а з точки зору систем – така сітка еквівалентна скінченному автомату. Такі особливості потенційно надають безліч можливостей для моделювання біологічних НС.

До суттєвих властивостей ДРНС належить, наприклад, можливість роботи з образами, параметри яких змінюються у часі. Дійсно, сигнали, які надходять на входи нейронів вхідного шару в деякий момент часу $t > 0$, перетворюються останніми за допомогою відповідних активаційних функцій $f_a(x, w)$ і в наступний момент часу передаються по зв'язкам, які є у сітці, іншим нейронам, які їх також перетворюють. Далі перетворений сигнал по прямим і зворотним зв'язкам надходить на входи нейронів, і процес повторюється.

Наявність зворотних зв'язків приводить до того, що в залежності від значень вхідного сигналу та вагових параметрів сітка може:

- досягати деякого стійкого стану;
- осцилювати, тобто періодично повторювати значення вихідних сигналів;
- хаотично змінювати свій стан.

Таким чином, динаміка ДРНС аналогічна динаміці сітки Хопфілда (див. розд. 6.7). Наявність зворотних зв'язків в даній сітці не дозволяє використовувати для її опису такі прості співвідношення, які застосовувалися у розглянутих раніше сітках. Через це їх динаміка, як і динаміка сіток Хопфілда, які є *поодиноким випадком динамічної ДРНС*, описується нелінійними диференційними (у неперервному випадку) або різницевими (у дискретному випадку) рівняннями першого порядку.

Модифікація двошарового персептрона. Найбільш розповсюдженим є такий підхід: за основу береться звичайний двошаровий персептрон (схематично поданий на рис. 12.3), для якого виходи схованого шару передають сигнал на шар на виході і одночасно через затримку T з певним ваговим коефіцієнтом W знов подають на входи схованого шару, змішуючи їх з сигналом a зовнішнього світу.

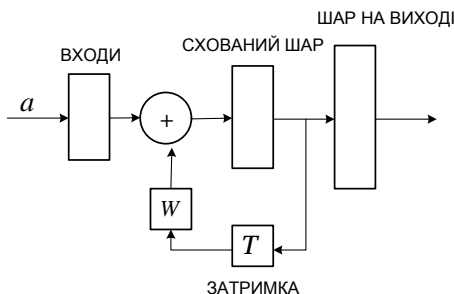


Рис. 12.3. Схематичне подання модифікованого під ДРНС двошарового персептрона

Розглянемо проходження сигналів через модифікований під рекурсивну нейросітку двошаровий персептрон. Сигнал a надходить на вхід ДРНС і проходить на схований шар. Перетворений схованим шаром сигнал далі подається на шар на виході і виходить з ДРНС, а його копія – на затримку T . Далі на вхід ДРНС надходить другий сигнал, який проходить на схований шар, підсумовуючись з сигналом, який після помноження на деякий коефіцієнт W надходить із затримки на попередньому кроці. Після цього підсумований сигнал з виходу схованого шару піде на шар на виході, а його копія знов збережеться в елементі T затримки. І так далі. Зазвичай, перед підсумовуванням

“+” сигнал із затримки T помножується на деякий коефіцієнт W , менший за одиницю. Чим більшим є цей коефіцієнт, тим більший вплив попередніх сигналів на вихід ДРНС. Іноді між сигналом у затримці і сигналом на виході ДРНС береться середньоарифметичне.

З наведеної структури зрозуміло, що кількості нейронів схованого шару і входів ДРНС (розмірність вхідного образу) мають бути рівними. Такий варіант ДРНС реалізований, наприклад, у Neural Networks Toolbox в системі MatLab.

Можна запропонувати безліч варіантів і комбінацій побудовання ДРНС як модифікацій багатозарового персептрона, але суть залишиться незмінною: результат роботи одного з шарів сітки в тому чи іншому вигляді подається на входи при опрацюванні наступного сигналу/образу.

Можна зробити декілька загальних для усіх рекурсивних НС зауважень:

- одиницею опрацювання є не один образ, а їх послідовність;
- нейросітка “пам’ятає” початок опрацьовуваної послідовності. Пам’ять про кожний з образів, а відтак і його вплив, згасає експоненціально;
- сітка зберігає всі властивості багатозарового персептрона;
- для першого елемента опрацьовуваної послідовності всі рекурсивні елементи припускаються неактивними або приймають заздалегідь напередвизначений стан.

Недоліки ДРНС. Зауважимо, що кожного разу при оптимізації ваг ми працюємо з нейросіткою як зі звичайним багатозаровим персептроном. Слід зазначити, що у випадку рекурсивних НС використовується не сам градієнт помилки, а лише його наближення через те, що при підрахуванні помилки не враховується похибка, внесена через дані на затримки. Спроба урахування цієї складової напряму може привести до занадто глибокої рекурсії. Через це навчання збігається повільніше, і для якісного функціонування ДРНС необхідно, в загальному випадку, більша кількість нейронів, аніж для звичайної нейронної сітки.

Переваги ДРНС. Проте, наведені недоліки ДРНС перебиваються багатьма їх перевагами, а саме:

- спроможністю опрацьовувати послідовності образів, враховуючи зв’язок між елементами послідовності;
- спроможністю виробляти послідовності у відповідь як на послідовності, так і на поодинокі образи (досить доповнити образ до послідовності нульовими сигналами);
- високою спроможністю до узагальнення, яка є вищою навіть ніж у звичайного багатозарового персептрона.

Застосування ДРНС. Сітка застосовується для моделювання динамічних процесів в режимі «On-Line», задач кластеризації. Типовим прикладом її застосування може служити імітація нелінійних динамічних об'єктів, для яких сітка виступає в ролі моделі, а алгоритм уточнення – процедура ідентифікації параметрів моделі. По цій причині сітка найбільш популярна для імітації систем управління машинами, пристроями і динамічними процесами.

12.2. Частково-рекурсивні дискретні нейронні сітки

Рекурсивні сітки зручні для розв'язання задач розпізнавання, класифікації образів та прогнозування часових рядів. Іноді замість одиничних образів на вхід сітки подається одночасно послідовність S часткових образів у вигляді деякого вікна, яке зсувається назад при надходженні кожного нового образу. І хоча таке ковзне вікно може бути реалізованим за допомогою ШНС прямого поширення, більш ефективним є розв'язання подібних задач за допомогою *частково-рекурсивних нейронних сіток* (ЧРНС). Такі сітки займають проміжне положення між “чистими” сітками прямого поширення і “чистими” рекурсивними сітками.

Найпоширеніші модифікації ЧРНС. В перших ДРНС ідея полягала у наданні сітці можливості бачити свій образ на виході на попередньому кроці. У такої сітки тільки частка рецепторів приймає сигнали з навколишнього світу, а на інші рецептори надходить образ на виході з попереднього моменту часу.

Модель нейрона така ж, як і у багатошаровому персептрона.

Структура зв'язків: будь-яка ДРНС будується на основі двошарового персептрона.

Розглянемо більш детально проходження послідовності вхідних векторів через найпоширеніші модифікації ДРНС.

Структура ДРНС, наведена на рис. 12.4, передбачає наявність зворотних зв'язків з шару на виході на деяку кількість рецепторів на вході сітки. Вектор сигналу/образу надходить на групу рецепторів a , з'єднаних із зовнішнім світом, і проходить на схований шар (будемо вважати, що початковому етапі інші рецептори b неактивовані).

Перетворений схованим шаром сигнал надходить на шар на виході, а далі виходить з нейросітки, тоді як його копія подається на елемент T затримки. Далі на вході рецептори a ДРНС, сприймаючи зовнішні сигнали, надходить вектор другого сигналу/образу, а на другу, неактивовану при сприйнятті першого образу групу рецепторів b – образ на виході з попереднього кроку з ваговими коефіцієнтами $w_{bi}, i = \overline{1, N}$ (N – кількість вхідних рецепторів b) після елемента T затримки. Тепер вже з усіх рецепторів сигнал піде на схований шар,

для якого природа сигналу з другої групи рецепторів b є байдужою, і через це він буде функціонувати, як у звичайному багатошаровому персептроні. Сигнал зі схованого шару надійде на шар на виході. Опрацьований шаром на виході, сигнал вийде з ДРНС, а його копія знов попадає на елемент T затримки і далі з ваговими коефіцієнтами $w_{bi}, i = \overline{1, N}$ на входи рецепторів b . І так далі.

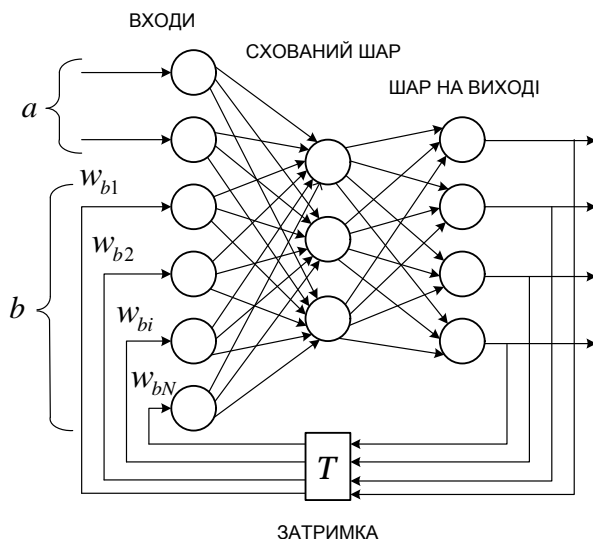


Рис. 12.4. Структура динамічної рекурсивної нейронної сітки

Наступним, модифікуючим ДРНС кроком є використання того ж самого принципу, але щодо збереження не образу на виході, а сигналу зі схованого шару. Як бачимо зі схеми на рис. 12.5, у модифікованій ДРНС два входи a для сигналів ззовні, три нейрона у схованому шарі і три нейрона b у вхідному шарі, на які з ваговими коефіцієнтами $w_{bi}, i = \overline{1, 3}$ після деякої затримки T надходить копія результату роботи схованого шару.

Динамічні нейронні сітки розподіляються на:

- неперервні;
- дискретні.

До різновидів дискретних ДРНС відносяться:

- повнозв'язні;
- частково-рекурсивні сітки;
- локально-рекурсивні сітки прямого поширення;
- двоспрямовані ДРНС (розглядаються у розд. 6.9)

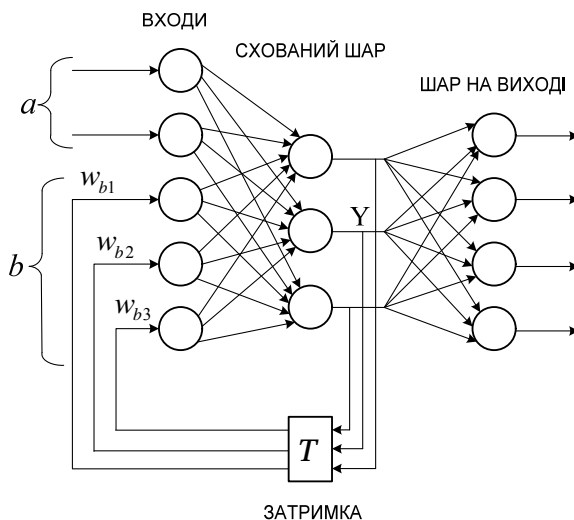


Рис. 12.5. Структура модифікованої рекурсивної нейросітки

12.3. Динаміка неперервної рекурсивної нейросітки

Динаміка i -го нейрона неперервної ДРНС описується рівняннями виду:

$$\tau_i \frac{dx_i}{dt} = -x_i(t) + f_i \left(\sum_{j=1}^M w_{ij} x_j(t) + v_i \right), \quad i = \overline{1, M} \quad (12.1)$$

де τ_i – постійна часу i -го нейрону; $x_i(t)$ – стан i -го нейрону в момент часу t ; $f_i(*)$ – нелінійна функція активації; v_i – зовнішній вхідний сигнал i -го нейрону; M – кількість нейронів у сітці.

Елементи w_{ij} вагової матриці \mathbf{W} визначаються шляхом розв'язання рівнянь:

$$\frac{dx(t)}{dt} = 0. \quad (12.2)$$

В залежності від виду матриці \mathbf{W} розрізняють три типи сіток:

- з симетричною матрицею ваг з нульовими діагональними елементами – ДРНС типу сітки Хопфілда;
- з трикутною матрицею ваг – ДРНС прямого поширення без зворотних зв'язків;
- з ваговою довільною матрицею – ДРНС загального виду.

В останньому випадку сітка в залежності від значень її параметрів може або досягати деякого стійкого стану, або осцилювати, або хаотично змінювати свій стан. Зокрема, для досягнення стійкого стану вагові коефіцієнти w_{ij} повинні задовольняти наступній умові:

$$\sum_i \sum_j w_{ij}^2 < \left(\max_i |f'_i| \right)^2, \quad (12.3)$$

де $f'_i = \frac{df}{dx}$ – похідна функції активації.

12.4. Динаміка дискретної рекурсивної нейросітки

Елементи вагової матриці \mathbf{W} визначаються шляхом розв'язання системи рівнянь:

$$x_i(k+1) = f \left(\sum_{j=1}^M w_{ij} x_j(k) + v_i \right), \quad (12.4)$$

де k – кількість передуючих тактів ($k=1, 2, 3, \dots$); v_i – зовнішній вхідний сигнал i -го нейрона.

При $\Delta t \rightarrow 0$ поведінки дискретної та неперервної ДРНС ідентичні.

В загальному випадку поведінка і дискретної, і неперервної сіток навіть при однакових значеннях вагової матриці відмінна – наприклад, якщо одна досягає стійкого стану, інша може осцилювати.

Рівняння, що описують динаміку дискретної ДРНС, отримуються з (12.1) шляхом переходу до скінченних різниць, а елементи вагової матриці є розв'язком системи рівнянь (12.4).

Від залежностей типу (12.4) можна перейти до подання у просторі станів виду:

$$\begin{aligned} y(k) &= \chi(q(k), x(k)); \\ q(k+1) &= f(q(k), x(k)), \end{aligned} \quad (12.5)$$

де $q(k)$ – стан сітки в момент часу, відповідаючому k -му такту; χ^* , f^* – функції, відповідно визначаючи вихід та стан сітки на наступному такті (в теорії цифрових автоматів їх називають функціями виходів і переходів відповідно).

12.5. Повнозв'язні динамічні рекурсивні нейросітки

Повнозв'язні ДРНС (ПДРНС) розглядалися в роботах [82; 180; 181; 188].

Архітектура ПДРНС (рис. 12.6) була первісно запропонована для розв'язання задач, пов'язаних з аналізом та обробкою послідовностей сигналів, що надходять на вхід сітки, але надалі була використана також для ідентифікації нелінійних динамічних об'єктів. Проте даній сітці притаманний серйозний недолік – повільна збіжність (суттєва тривалість процесу навчання) і виникаючі при цьому проблеми стійкості [98].

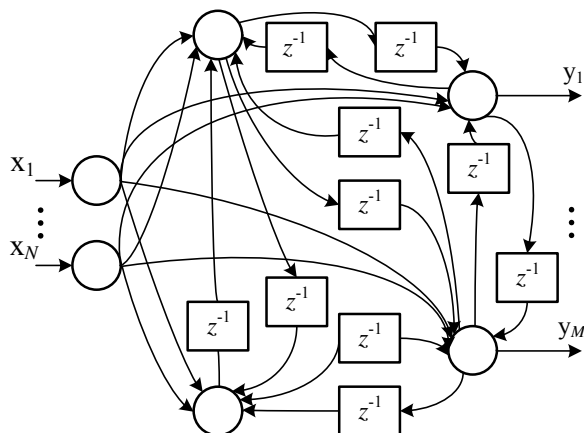


Рис. 12.6. Архітектура повнозв'язної дискретної динамічної рекурсивної нейросітки (z^{-1} – елементи затримки)

12.6. Сітка Джордана

Загальні відомості. Новий етап у розвитку нейронних сіток зі зворотними зв'язками – ДРНС розпочався з нейронної сітки Джордана, статтю про яку він публікує в 1986 році [148; 149].

Означення 12.2. Сітка Джордана (рис. 12.7) - вид НС, який здобувається з багат шарового персептрона, якщо на його вхід подати, окрім вхідного вектора, вихідний із затримкою на один або декілька тактів.

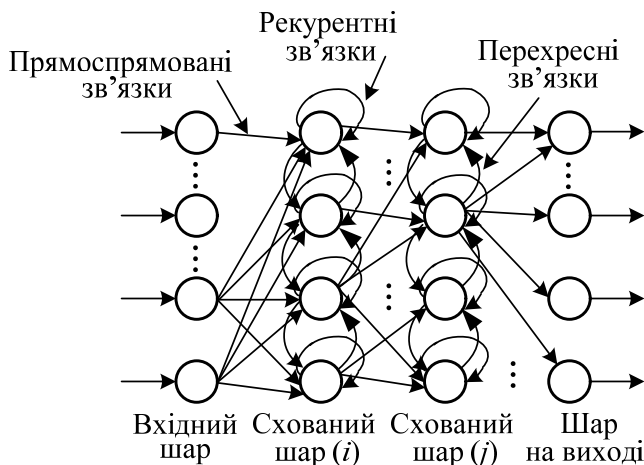


Рис. 12.7. Багат шарова рекурсивна сітка Джордана

Іноді сітку Джордана називають *багатошаровим перцептроном з контекстним шаром*, або *рекурсивним багатошаровим перцептроном Джордана* (РБПД). У загальному вигляді структура багатошарової ДРНС Джордана, на відміну від багатошарового перцептрона, має окрім прямоспрямованих зв'язків також і рекурентні та перехресні зв'язки.

Взагалі у такої сітки тільки частина рецепторів приймає сигнали з навколишнього середовища, на інші ж рецептори надходить образ з виходу сітки з попереднього моменту часу.

Розглянемо проходження послідовності сигналів через сітку.

Отже, на відміну від ПДРНС запропонована *М. Джорданом* частково-рекурсивна НС уявляє собою багатошаровий перцептрон, доповнений так званим *контекстним шаром* (рис. 12.8), нейрони якого реалізують пам'ять ЧРНС і кількісно співпадають з кількістю виходів ШНС.

Вхідні сигнали сітки разом з сигналами контекстного шару надходять на входи нейронів схованого шару, вихідні сигнали яких передаються на вихідний шар. Виходи нейронів цього шару є виходами ШНС. Крім того, вихідні сигнали у вигляді сигналів зворотного зв'язку передаються з вагами $w^{(зв)}$ на контекстний шар. Нейрони контекстного шару мають власні зворотні зв'язки з вагою γ , яка зазвичай не змінюється.

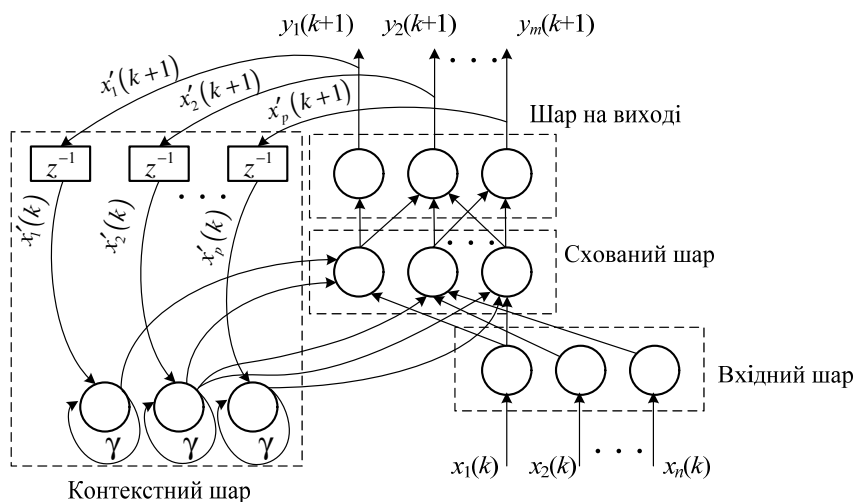


Рис.12.8. Структура динамічної рекурсивної нейросітки Джордана

Зворотний зв'язок може виходити або з шару нейронів на виході, або зі схованого шару. У кожному контурі такого зв'язку присутній елемент одиничної затримки, який передає значення на так званий *контекстний нейрон*, дякуючи якому потік сигналів може вважатися односпрямованим (сигнал з виходу сітки попереднього часового циклу розглядається як апіорі заданий, який просто збільшує розмірність вхідного вектора x сітки, додаючи контекстні нейрони). Подана у такий спосіб ДРНС з урахуванням способу формування сигналу на виході функціонує як односпрямована персептронна сітка. Проте алгоритм навчання такої сітки, адаптуючий значення синаптичних ваг, є більш складним через залежності сигналів у момент часу t від їх значень у попередні моменти і відповідно через більш громіздкі формули для розрахунку вектора градієнта.

Сигнал на виході сітки в кожний момент часу залежить від значення вхідного сигналу і стану, запам'ятованого у контекстному шарі. Тому ШНС описується рівняннями (12.5).

Послідовна зміна станів нейронів при деякому початковому стані z_0 подається формулою:

$$z(k) = \begin{cases} z_0, & \text{якщо } k = 1; \\ \gamma z(k-1) + w^{(3b)} y(k-1), & \text{якщо } k > 1, \end{cases} \quad (12.6)$$

тобто

$$z(k) = \gamma^{k-1} z_0 + w^{(3b)} \sum_{n=1}^{k-1} \gamma^{n-1} y(k-n), \quad (12.7)$$

де $w^{(3b)} \in [0, 1]$ – ваговий параметр зворотного зв'язку (зазвичай $w^{(3b)} = 1$).

Якщо $z_0 = 0$ та $w^{(3b)} = 1$, то

$$z(t) = \sum_{n=1}^{t-1} \gamma^{n-1} y(t-n), \quad (12.8)$$

тобто стан уявляє собою експоненційно зважену суму усіх сигналів на виході, які до цього часу мали місце.

При малих значеннях ваг γ елементів контекстного шару вплив попередніх станів замалий, а при $\gamma \rightarrow 1$ великий. При $\gamma = 1$ всі стани мають однакову вагу. Таким чином, використання параметра $\gamma < 1$ з одного боку зменшує вплив раніше надходжуваних образів на поточний, а з іншого – дозволяє враховувати цей вплив.

Під час навчання сітки ваги γ не змінюються, хоча принципово змінювання можливе. Проте це лише приводить до затягання процесу навчання без суттєвого покращення роботи ШНС.

Узагальнена модель РБПД. Це динамічна сітка, яка характеризується запізнюванням вхідних і на виході сигналів, об'єднаних у вхідний вектор сітки, узагальнена структура якої подана на рис. 12.9.

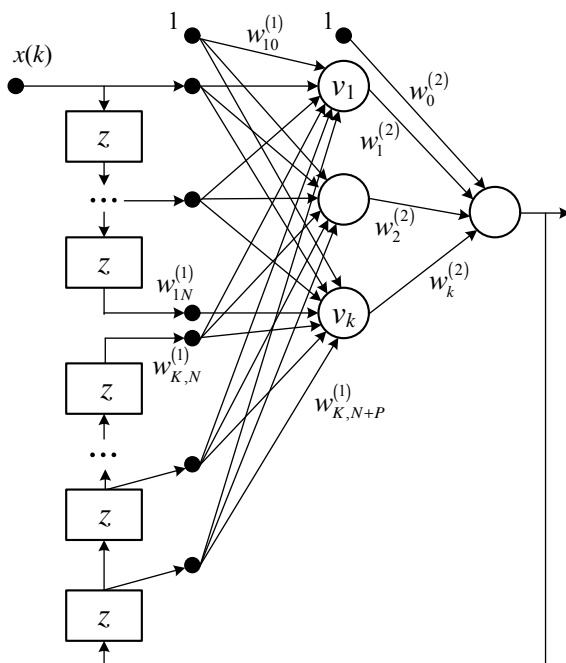


Рис.12. 9. Структура рекурсивного багатошарового персептрона Джордана

Міркування будуть торкатися тільки одного вхідного вузла $x(k)$ та одного нейрона на виході, а також одного схованого шару. Така система реалізує відображення:

$$\begin{aligned} y(k+1) &= f(x(k), x(k-1), \dots, x(k-(N-1))), \\ y(k-1), y(k-2), \dots, y(k-P)), \end{aligned} \quad (12.9)$$

де $N-1$ – кількість затримок вхідного сигналу, а P – кількість затримок сигналу на виході. Позначимо через K кількість нейронів у схованому шарі. В цьому випадку сітку РБПД можна схарактеризувати трійкою чисел (N, P, K) .

Надходячий на вхід сітки вектор \mathbf{x} має вид:

$$\mathbf{x}(k) = \begin{bmatrix} 1, x(k), x(k-1), \dots, x(k-(N-1)), \\ y(k-P), y(k-P+1), \dots, y(k-1) \end{bmatrix}^T. \quad (12.10)$$

Припустимо, що всі нейрони мають сигмоїдальну функцію активації. Позначимо через u_i зважену суму сигналів i -го нейрона схованого шару, а через g – зважену суму сигналів нейрона на виході. При уведених позначеннях сигнали на виході конкретних нейронів описуються залежностями:

$$\begin{aligned} u_i &= \sum_{j=0}^{N+P} w_{ij}^{(1)} x_j; \\ v_i &= f(u_i); \\ g &= \sum_{i=0}^K w_i^{(2)} f(u_i); \\ y &= f(g). \end{aligned} \tag{12.11}$$

Сітка Джордана спроможна асоціювати всілякі вхідні образи з різними послідовностями на виході.

12.7. Багатoshаровий персептрон Дж. Елмана

В літературі ДРНС часто називають *сітками Елмана*, проте, Елман є автором лише однієї з моделей. Ідея полягає у тому, щоб пристосувати двошаровий персептрон до опрацювання послідовностей. Ці сітки можуть застосовуватися в системах *керування рухомими об'єктами* через те, що їх головною особливістю є запам'ятовування послідовностей.

Сітка Дж. Елмана [102] є модифікацією сітки М. Джордана. Найпростіша сітка Елмана складається з одного схованого та одного контекстного шару нейронів (рис. 12.10), у якій сигнали зворотних зв'язків надходять не з вихідного шару, а з виходів нейронів схованого шару, тому кількість нейронів контекстного та схованого шарів співпадає. Крім того, тут нема власних зворотних зв'язків нейронів контекстного шару. Дані нейрони виконують роль *функції активації*. Надходження на вхід сітки першого образу активує нейрони усіх шарів: схованого, вихідного і контекстного. Через те, що нейрони контекстного шару відіграють роль, аналогічну активаційній функції, вони перейдуть у новий стан, відповідаючий стану нейронів схованого шару, копіюючи тим самим (запам'ятовуючи) інформацію. З виходів нейронів схованого шару сигнал передається на входи нейронів вихідного шару, які і формують вихідний сигнал сітки. При надходженні наступного образу стан нейронів контекстного шару відповідає попередньому образу.

Основною задачею нейронів схованого шару є продукування бажаного вихідного сигналу на підставі порівняння знову надійшовшого образу та образу, який запам'ятався у контекстному шарі

Стан нейронів описується наступним виразом:

$$z(t) = \sum_{n=1}^{t-1} \gamma^{n-1} y(t-n) \quad (12.12)$$

При малих значеннях γ вплив попередніх станів малий, а при суттєвих – значний. При цьому усі стани мають однакову вагу (використання параметра γ з одного боку зменшує вплив раніше надійшовших образів на поточний, а з іншого – дозволяє врахувати цей вплив).

Під час навчання сітки ваги γ не змінюються, хоча принципово зміна γ можлива; проте, як показують дослідження, це лише приводить до уповільнення процесу навчання без істотного покращення роботи ШНС. Проте питання вибору γ , малі значення якого дозволяють краще реагувати на образ, який знов надходить, а великі – забезпечувати запам'ятовування усіх образів, що надійшли раніше, залишається відкритим.

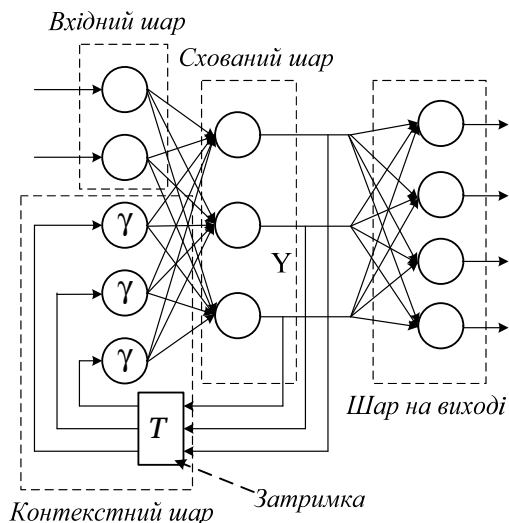


Рис. 12.10. Структура динамічної рекурсивної нейросітки Елмана

Отже, нейронна сітка Елмана характеризується частковою рекурентністю у формі зворотного зв'язку між схованим шаром та шаром на виході, який реалізується за допомогою одиничних елементів запізнювання z^{-1} (запізнювання на один часовий крок). В узагальненій структурі цієї сітки (рис. 12.11) кожен схований нейрон має свій аналог у контекстному шарі. Шар на виході складається з односпрямованих нейронів, зв'язаних тільки з нейронами схованого шару (подібно до сітки РБПД).

Позначимо внутрішній вектор збудження сітки через \mathbf{x} (до його складу входить також одиничний сигнал поляризації), стан схованих нейронів як $\mathbf{v} \in R^K$, а сигнали на виході сітки як $\mathbf{y} \in R^M$. При таких позначеннях вхідний вектор сітки в момент t має форму

$$\mathbf{x}(k) = [x_0(k), x_1(k), \dots, x_N(k), v_1(k-1), v_2(k-1), \dots, v_K(k-1)]. \quad (12.13)$$

Ваги синаптичних зв'язків першого (схованого) шару позначимо як $w_{ij}^{(1)}$, а другого (на виході) шару як $w_{ij}^{(2)}$. Якщо зважену суму i -го нейрона схованого шару позначити через u_i , а його сигнал на виході через v_i , то

$$u_i(k) = \sum_{j=0}^{N+P} w_{ij}^{(1)} x_j(k); \quad (12.14)$$

$$v_i(k) = f_1(u_i(k)).$$

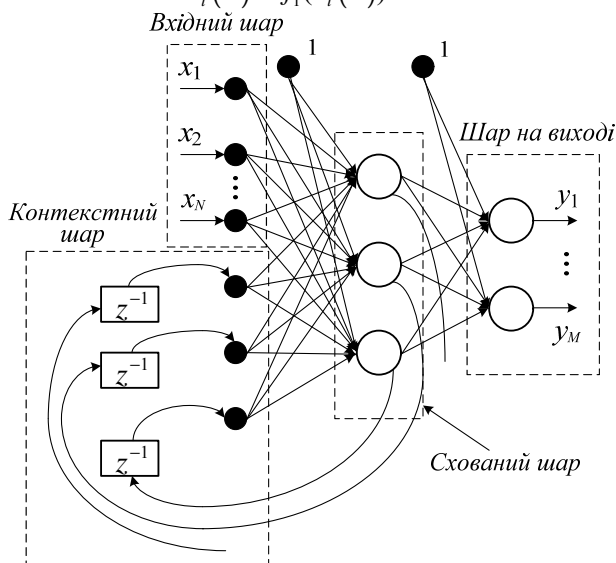


Рис. 12.11. Узагальнена структура сітки Елмана

Ваги $w_{ij}^{(1)}$ утворюють матрицю \mathbf{W}^1 синаптичних зв'язків схованого шару, а $f_1(u_i)$ – функція активації i -го нейрона цього шару. Аналогічно можна позначити зважену суму i -го нейрона шару на виході через g_i , а відповідний до нього сигнал на виході сітки як y_i . Ці сигнали описуються залежностями

$$g_i(k) = \sum_{j=0}^K w_{ij}^{(2)} v_j(k); \quad (12.15)$$

$$y_i(k) = f_2(g_i(k)).$$

В свою чергу, ваги $w_{ij}^{(2)}$ утворюють матрицю \mathbf{W}^2 , якою описуються синаптичні зв'язки нейронів шару на виході, а $f_2(g_i)$ – функція активації i -го нейрона шару на виході.

Ієрархічна частково-рекурсивна НС. Для розв'язання більш складних задач використовують ієрархічну частково-рекурсивну нейронну сітку (ІЧРНС), в якій кожний схований шар має свій контекстний шар (рис. 12.12). Можливості сітки різко зростають внаслідок наявності вибору різних значень γ .

Якщо видалити сигнали зворотних зв'язків γ , що надходять на контекстний шар, то отримаємо чисту сітку прямого поширення, у якій контекстний шар є доповнюючим входним. При цьому розширений вхідний вектор складається безпосередньо з вхідного вектора X та вектора Z станів нейронів контекстного шару, який визначається на кожному тактові функцією переходів.

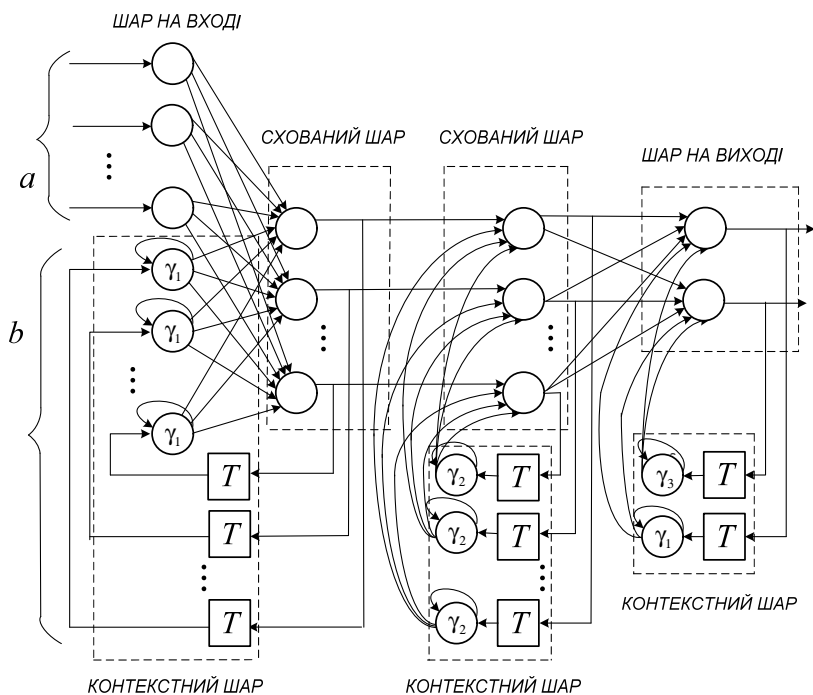


Рис. 12.12. Ієрархічна частково-рекурсивна НС

На основі сітки Елмана будується нейронна сітка класу рекурсивної автоасоційованої пам'яті (РААП), яка за своєю структурою повторює елманівську і уявляє собою подвійну сітку Елмана виду $2N-N-2N$, яку використовують для стиснення і шифрування інформації. На вхід РААП подається бітовий сигнал з $2N$ біт. Зазвичай

сітка має розмір 20-10-20, де перші 10 біт називають “лівими”, а другі 10 – “правими”. З самого початку на ліву матрицю подається нульовий вектор біт (0000000000), а на праву – код символу або речення (наприклад, 0010000000=“А”). Те ж саме подається на матриці виходу. Методом зворотного поширення похибки сітка РААП навчається. Далі 10 біт зі схованого шару передаються на ліву вхідну матрицю, а на праву надходить черговий символ. В процесі такого рекурсивного навчання інформація стискається і шифрується, що було застосовано на практиці французьким професором Томасом Вєгтліном у 2002 році в роботі [200].

12.8. Локально-рекурсивні сітки прямого поширення

Локально-рекурсивні нейронні сітки (ЛРНС) не використовують ані зворотний зв'язок між нейронами суміжних шарів, ані laterальні зв'язки між нейронами одного шару. Рекурсивність в них завжди обмежується одним нейроном. Отримувані при цьому структури є лінійними, а уводжувані зворотні зв'язки інтерпретуються як фільтри зі скінченою або нескінченою імпульсною характеристикою (СІХ або НІХ). Існує три різних способи одержання локальної рекурсивності або, іншими словами, локального введення у сітку динаміки [48]:

- *динаміка синапсів*, яка використовує локальні зворотні зв'язки;
- *“активізаційна динаміка”*, яка є поодиноким випадком динаміки синапсів і дозволяє у випадку однакових передатних функцій всіх синапсів одного нейрона суттєво спростити структуру сітки; в цьому випадку фільтри, використовувані у синапсах, можуть бути замінені одним фільтром, який має ті самі нулі і полюси і розташований після операції підсумовування [174];
- *динаміка зворотного зв'язку*, яка реалізується шляхом введення лінійного зворотного зв'язку з виходу нейрона на його вхід.

Хоча основою ЛРНС є зазвичай багатозаровий перцептрон, ці сітки можуть бути реалізованими і на основі сіток радіально-базисних функцій (див. розд. 10).

12.9. Навчання рекурсивної нейросітки

Не дивлячись на всю складність структури зв'язків у ДРНС, навчання виконується майже за тією ж схемою, що і для багатозарового перцептрона. Методи навчання ДРНС:

- навчання з підкріпленням;
- зворотного поширення похибки.

Алгоритм є однаковим для усіх варіантів рекурсивної архітектури. Основною відмінністю є те, що навчаюча вибірка тепер складається не з окремих образів, а з їх послідовностей. Коротко наведемо цей алгоритм для багат шарової ДРНС (рис. 12. 13).

Алгоритм 12.1. Нехай u нас є навчаюча вибірка $\{X^\alpha, Y^\alpha\}$; $\alpha=1, 2, \dots, p$, де p – кількість елементів у послідовності і може бути фіксованим для усіх послідовностей у навчаючій вибірці, а може залежати від α . Реалізуємо кроки:

1. Взяти деяку послідовність $\{X_n^\alpha, Y_n^\alpha\}$ з навчаючої вибірки.
2. Виконати ініціалізацію випадковим чином ваг нейронів схованого шару і шару на виході.
3. Для пари образів $\{X_t^\alpha, Y_t^\alpha\}$ провести корекцію ваг за допомогою алгоритму зворотного поширення похибки, вважаючи затримку T порожньою (нульовий сигнал), заповнивши усі елементи затримки відповідно до конкретної архітектури зв'язків – в кожний момент часу t при заданому збудженні у виді вектора X розрахувати ваги всіх нейронів сітки відповідно до формул:

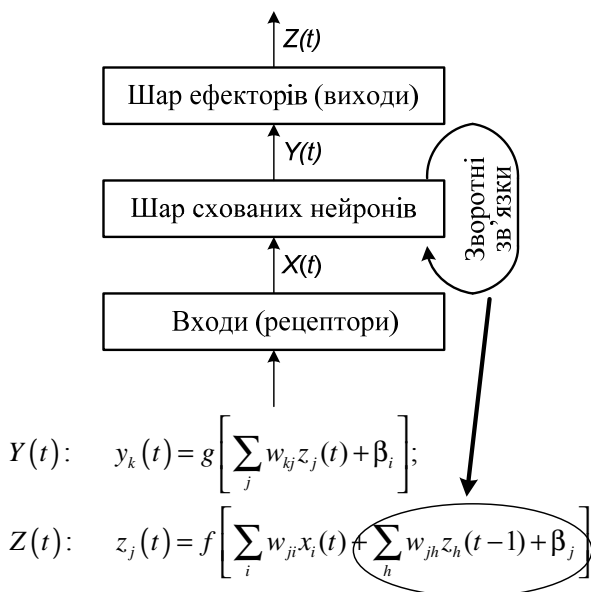


Рис. 12.13. Багат шарова рекурсивна нейронна сітка

$$g = \sum_{i=0}^K w_i^{(2)} f(u_i); \quad y = f(g). \quad (12.16)$$

$$u_i = \sum_{j=0}^{N+P} w_{ij}^{(1)} x_j; \quad v_i = f(u_i); \quad (12.17)$$

4. За допомогою залежностей

$$\frac{dy(k)}{dw_{\alpha}^{(2)}} = \frac{df(g(k))}{dg(k)} \times \left[v_{\alpha}(k) + \sum_{i=0}^k w_i^{(2)} \frac{df(u_i(k))}{du_i^{(k)}} \sum_{j=1}^P w_{i,j+N}^{(1)} \frac{dy(k-P-1+j)}{dw_{\alpha}^{(2)}} \right]; \quad (12.18)$$

$$\frac{dy(k)}{dw_{\alpha,\beta}^{(1)}} = \frac{df(g(k))}{dg(k)} \sum_{i=1}^K w_i^{(2)} \frac{df(u_i(k))}{du_i^{(k)}} \times \left[\sum_{j=0}^P w_{i,j+N}^{(1)} \frac{dy(k-P-1+j)}{dw_{\alpha}^{(2)}} + \delta_{i\alpha} x_{\beta} \right] \quad (12.19)$$

визначити значення похідних для всіх значень α і β , що відповідають вагам сітки з початково вибраною структурою.

5. Актуалізувати ваги відповідно до формул:

$$\Delta w_{\alpha}^{(2)} = -\eta [y(k) - d(k)] \frac{dy(k)}{dw_{\alpha}^{(2)}}; \quad (12.19)$$

$$\Delta w_{\alpha,\beta}^{(1)} = -\eta [y(k) - d(k)] \frac{dy(k)}{dw_{\alpha,\beta}^{(1)}}.$$

6. Спорожнивши затримку, повернутись до п. 2 і повторити кроки 1-3 для послідовностей у навчаючій вибірці, поки сумарна похибка не стане достатньо малою.

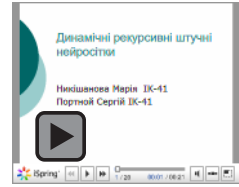
Контрольні завдання

1. Розкрити динамічний аспект функціонування нейронних сіток зі зворотними зв'язками.
2. Охарактеризувати багатoshарові рекурсивні сітки як розвиток одношарованих сіток персептронного типу.
3. Проаналізувати функціонування модифікованого під динамічну рекурсивну нейронну сітку двошарового персептрона.
4. Прокоментувати структуру динамічної рекурсивної нейронної сітки і можливості її модифікації.

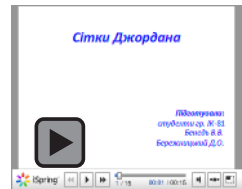
5. Охарактеризувати динаміку неперервної динамічної рекурсивної сітки.
6. Охарактеризувати динаміку дискретної динамічної рекурсивної сітки.
7. Розкрити особливості архітектури повнозв'язної дискретної динамічної рекурсивної нейросітки.
8. Навести означення динамічної рекурсивної нейронної сітки Джордана і розкрити роль контекстного шару в сітці.
9. Прокоментувати особливості побудовання структури і функціонування рекурсивного багатошарового персептрона Джордана.
10. Пояснити, в чому полягає модифікація Дж. Елманом динамічної рекурсивної нейронної сітки.
11. Охарактеризувати часткову рекурентність в узагальненій структурі сітки Елмана.
12. Прокоментувати структуру і особливості функціонування ієрархічної частково-рекурсивної нейронної сітки.
13. Охарактеризувати локально-рекурсивні нейронні сітки прямого поширення.
14. Розкрити узагальнений алгоритм навчання для усіх варіантів нейронних сіток рекурсивної структури.

Презентації

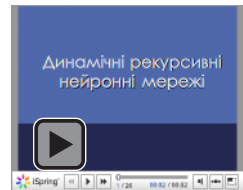
1. Динамічні рекурсивні штучні нейросітки



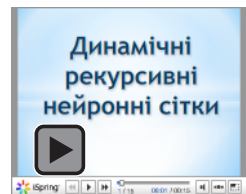
2. Сітки Джордана



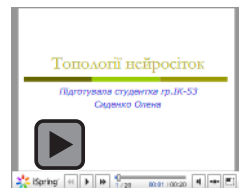
3. Динамічні рекурсивні нейронні мережі



4. Динамічні рекурсивні нейронні сітки



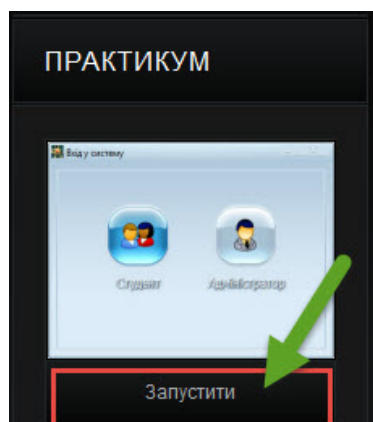
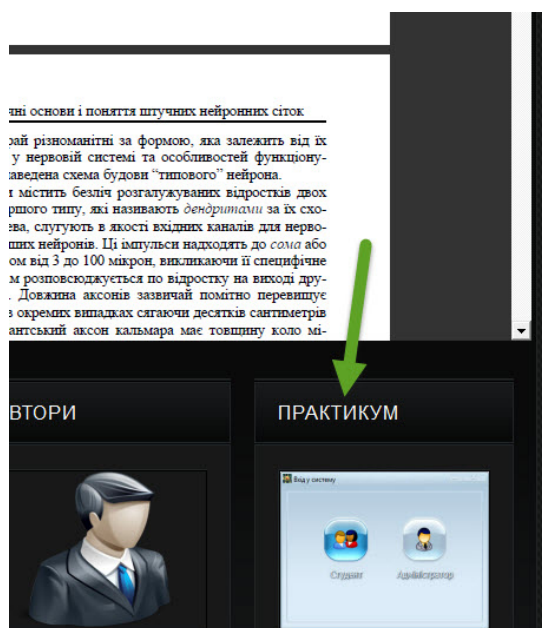
5. Топології нейросіток



Практичні заняття

Динамічні рекурсивні нейронні сітки

Для запуску практичних занять на
вкладці "Практикум" натисніть
"Запустити"



Розділ 13

Когнітрон та неокогнітрон

*Особливо різко точку зору змінює удар в око
З численних спостережень*

*Мета виправдовує засоби
Нікколо Макіавеллі
... Але прибуток — перевищує все*

13.1 Проблеми штучного відтворення процесу розпізнавання образів

Люди розв'язують складні задачі розпізнавання образів з надзвичайною легкістю, розрізняючи без ускладнень велику кількість облич та інших об'єктів з навколишнього середовища не зважаючи на зміну віддалі, перспективи, освітлення та кутів зору. Це наводить на думку, що вивчення відповідних людських здібностей зробить простою задачу створення комп'ютера, відтворюючого ці природжені властивості людини до розпізнавання, проте, ніщо не може бути більш далеким від істини. Схожість і відмінність образів є настільки ж очевидними для людини, як і поки що “незрозумілими” для найскладніших комп'ютерних систем розпізнавання.

Дослідник, конструюючий типову систему розпізнавання образів, зазвичай починає з розпізнавання друкованого тексту. Ці методи часто є неадекватними задачі, і прагнення розробників швидко зводяться до створення алгоритмів, вузько специфічних для даної задачі. В інших випадках метою створення системи розпізнавання образів є оптимізація її функціонування над вибраним набором образів, причому, розробник частіше за все завершує цю задачу виявленням нового, приблизно схожого образу, що призводить до невдалого завершення алгоритмів. Такий процес може продовжуватися невідомо довго, ніколи не приводячи до усталеного розв'язку, достатнього для відтворення процесу сприйняття людини, яка оцінює якість функціонування системи.

Очевидно, причина полягає в тому, що біологічні моделі відтворити важко: по-перше, надвисока складність людського мозку утруднює розуміння принципів його функціонування; по-друге, виникають проблеми з проведенням експериментів на мікроскопічному рівні для дослідження внутрішньої електрохімії вузлів при одночасному контролюванні великої кількості вузлів із спостереженням їх

взаємодії; нарешті, по-третє, етичні міркування забороняють проведення багатьох експериментів, які можуть виконати тільки на людях (тварини ж не володіють здібностями людини описувати свої враження).

На мікроскопічному рівні було виявлено, що нейрони володіють як *збуджуючими*, так і *гальмуючими* синапсами. Перші прагнуть до збудження нейрона, тоді як останні – стримувати його збудження. Це наводить на думку, що мозок адаптується або зміною впливу цих синапсів, або утворенням чи руйнуванням синапсів в результаті дії навколишнього середовища. Дане міркування залишається поки гіпотезою з обмеженим фізіологічним підтвердженням. Проте дослідження, проведені в рамках цієї гіпотези, привели до створення цифрових моделей, і деякі з них володіють спроможностями до адаптивного розпізнавання образів.

13.2. Когнітрон

Структура когнітрона. Когнітрон був запропонований Куніхіко Фукушимою у 1975р і конструюється у вигляді шарів нейронів, сполучених синапсами [114; 115]. Як показано на рис. 13.1, *передсинаптичний* нейрон в одному шарі пов'язаний *постсинаптичним* нейроном в наступному шарі.

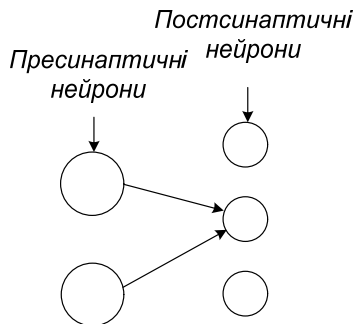


Рис. 13.1. Пресинаптичні та постсинаптичні нейрони

Є два типи нейронів: збуджуючі вузли, які прагнуть викликати збудження постсинаптичного вузла; гальмівні вузли, які гальмують це збудження. Збудження нейрона визначається зваженою сумою його збуджуючих і гальмівних входів, проте насправді механізм є складнішим, ніж просте підсумовування.

На рис. 13.2 показано, що кожний нейрон зв'язаний лише з нейронами в сусідній області, що зветься областю зв'язку. Це обмеження області зв'язку узгоджується з анатомією зорової кори, в якій

рідко поєднуються між собою нейрони, розташовані один від одного на відстані більш ніж один міліметр.

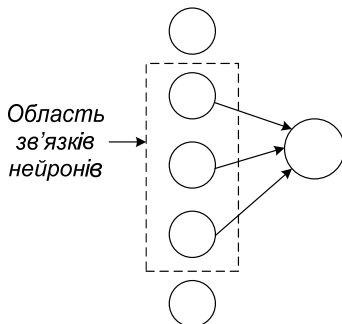


Рис. 13.2. Область зв'язків нейрона

Навчання. Оскільки когнітрон реалізований у вигляді багат шарової сітки, виникають складні проблеми навчання, пов'язані з обраною структурою. Фукушима К. відкинув кероване навчання як біологічно неправдоподібне, використовуючи замість цього навчання без вчителя. Отримуючи навчальний набір вхідних образів, сітка самоорганізовується за допомогою зміни сили синаптичних зв'язків. При цьому відсутні заздалегідь визначені вихідні образи, що представляють необхідну реакцію сітки, проте сітка самоналаштовується з метою розпізнавання вхідних образів з високою точністю.

Алгоритм навчання когнітрона є концептуально відмінним. У заданій області шару навчається лише найбільш сильно збуджений нейрон. Автор порівнює це з “елітним навчанням”, коли навчаються лише “розумні” елементи. Ті нейрони, які вже добре навчені, що виражається силою їх збудження, отримають приріст ваги своїх синапсів з метою подальшого посилення свого збудження.

На рис. 13.3 показано, що області зв'язку сусідніх вузлів значно перекриваються.

Це неефективне дублювання функцій виправдовується взаємною конкуренцією між найближчими вузлами. Навіть якщо вузли в початковий момент мають абсолютно ідентичний вихід, невеликі відхилення завжди мають місце – один з вузлів завжди матиме сильнішу реакцію на вхідний образ, чим сусідні. Його сильне збудження буде стримувати збудження сусідніх вузлів, і лише його синапси посилюватимуться; синапси сусідніх вузлів залишаться незмінними.

Збуджуючий нейрон. Грубо кажучи, вихід збуджуючого нейрона в когнітроні визначається відношенням його збуджуючих входів до гальмівних входів. Ця незвичайна функція має важливі перева-

ги, як практичні, так і теоретичні. Сумарний збуджуючий вхід в нейрон є зваженою сумою входів від збуджуючих в попередньому шарі.

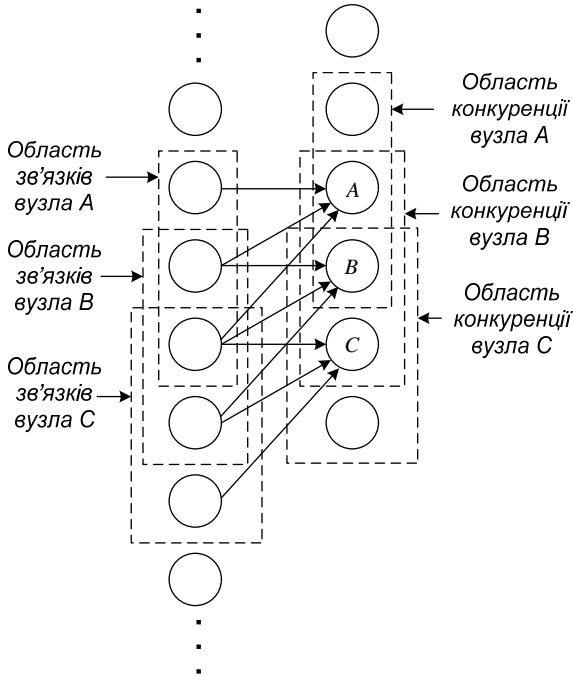


Рис. 13.3. Області зв'язку і конкуренції

Аналогічно сумарний гальмуючий вхід є зваженою сумою входів від всіх гальмівних нейронів. У символічному вигляді

$$E = \sum_i a_i u_i, I = \sum_j b_j u_j, \quad (13.1)$$

де a_i – вага i -го збуджуючого синапсу, u_i – вихід i -го збуджуючого нейрона, b_j – вага j -го гальмуючого синапсу, v_j – вихід j -го гальмуючого нейрона.

Зазначимо, що ваги мають лише позитивні значення. Вихід нейрона далі обчислюється таким чином:

$$\begin{aligned} \text{NET} &= \frac{1+E}{1+I} - 1, \\ \text{OUT} &= \text{NET} \text{ при } \text{NET} \geq 0, \\ \text{OUT} &= 0 \text{ при } \text{NET} < 0. \end{aligned} \quad (13.2)$$

Уявляючи, що NET має позитивне значення, це можна записати таким чином:

$$\text{OUT} = \frac{E - I}{1 + I}. \quad (13.3)$$

Коли гальмівний вхід малий ($I \ll 1$), OUT може бути апроксимоване як $\text{OUT} = E - I$, що відповідає виразу для звичайного лінійного порогового елементу (з нульовим порогом).

Алгоритм навчання когнітрона дозволяє вагам синапсів зростати без обмежень. Завдяки відсутності механізму зменшення ваг вони просто зростають в процесі вчення. У звичайних лінійних порогових елементах це привело б до довільно великого значення виходу елементу. У когнітроні великі збуджуючі та гальмівні входи підсумовуються в обмежуючій формулі вигляду:

$$\text{OUT} = \frac{E}{I} - 1, \text{ якщо } E \gg 1 \text{ та } I \gg 1. \quad (13.4)$$

В даному випадку OUT (вихід) визначається відношенням збуджуючих входів до гальмівних входів, а не їх різницею. Таким чином, величина OUT обмежується, якщо обидва входи зростають в одному і тому ж діапазоні X . Припустивши, що це так, E та I можна записати наступним чином: $E = pX$, $I = qX$; p, q – константи, і після деяких перетворень

$$\text{OUT} = \frac{p - q}{2q} \left[1 + \text{th} \left(\frac{\log(pq)}{2} \right) \right]. \quad (13.5)$$

Ця функція зростає згідно із законом Вебера-Фехнера, який часто використовується в нейрофізіології для апроксимації нелінійних співвідношень входу/виходу сенсорних нейронів. При використанні співвідношення (13.2) нейрон когнітрона в точності емулює реакцію біологічних нейронів. Це робить його як потужним обчислювальним елементом, так і точною моделлю для фізіологічного моделювання.

Гальмівні нейрони. У когнітроні шар складається із збуджуючих і гальмівних вузлів. Як показано на рис. 13.4, нейрон шару 2 має область зв'язку (наприклад, область конкуренції вузла j), для якого у нього існують синаптичні з'єднання з набором виходів нейронів у шарі 1.

Аналогічно в шарі 1 існує гальмівний нейрон, що має ту ж область зв'язку. Синаптичні ваги гальмівних вузлів не змінюються в процесі навчання; їх ваги заздалегідь встановлені таким чином, що сума ваг в будь-якому з гальмівних нейронів дорівнює одиниці. Відповідно до цих обмежень, вихід гальмівного вузла INHIB (інгібітор)

є зваженою сумою його входів, які в даному випадку є середнім арифметичним виходів збуджуючих нейронів, до яких він приєднаний. Таким чином,

$$\text{INHIB} = \sum_i w_i \text{OUT}_i, \quad \sum_i w_i = 1, \quad (13.6)$$

де w_i – збуджуюча вага i -го нейрона.

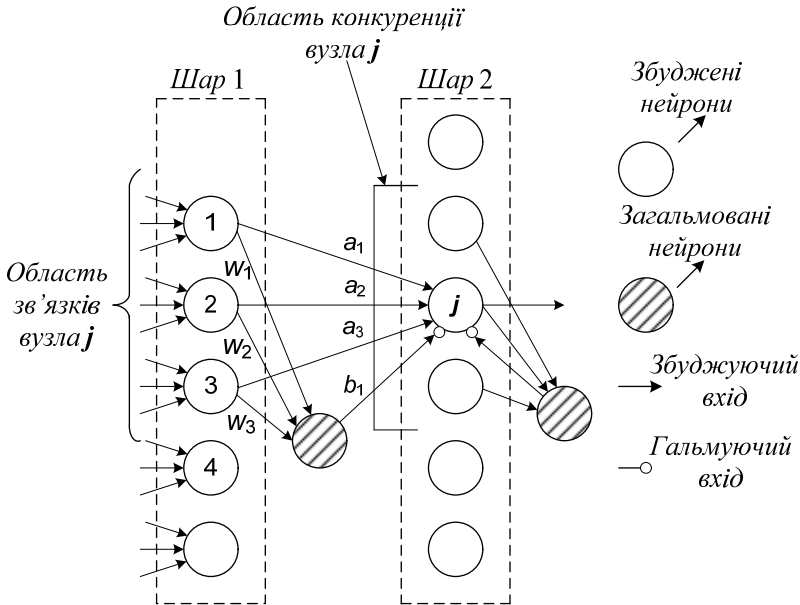


Рис. 13.4. Шари когнітрона

Процедура навчання. Як пояснювалося раніше, ваги збуджуючих нейронів змінюються лише тоді, коли нейрон збуджений сильніше, ніж будь-який з вузлів в області конкуренції. Якщо це так, зміна в процесі навчання будь-якої з його ваг визначається як:

$$\Delta a_i = q w_j u_j, \quad (13.7)$$

де w_j – гальмуюча вага зв'язку нейрона j у шарі 1 з гальмівним нейроном i ; u_j – вихід нейрона j у шарі 1; a_i – збуджуюча вага i ; q – нормуючий коефіцієнт навчання.

Зміна гальмівних ваг нейрона i в шарі 2 пропорційна відношенню зваженої суми збуджуючих входів до подвоєного гальмівного входу. Обчислення проводяться за формулою

$$\Delta b_i = \frac{q \sum_j a_j u_j}{2 \cdot \text{INGIB}_i}. \quad (13.8)$$

Коли збуджених нейронів в області конкуренції немає, для зміни ваг використовуються інші вирази. Це необхідно, оскільки процес навчання починається з нульовими значеннями ваг; тому спочатку немає збуджених нейронів в жодній з областей конкуренції, і навчання виконуватися не може. У всіх випадках, коли переможець в області конкуренції нейронів немає, зміна ваг нейронів обчислюється таким чином:

$$\Delta a_i = q' w_{ij} u_j, \quad \Delta b_i = q' \text{INGIB}, \quad (13.9)$$

де q' – додатний навчальний коефіцієнт менший, ніж q .

Наведена стратегія налаштування гарантує, що вузли з великою реакцією примушують збуджуючі синапси, якими вони управляють, збільшуватися сильніше, ніж гальмівні синапси. І навпаки, вузли, що мають малу реакцію, викликають мале зростання збуджуючих синапсів, але більше зростання гальмівних синапсів. Таким чином, якщо вузол 1 у шарі 1 має більший вихід, синапс a_1 зросте більше, ніж синапс b_1 . І навпаки, вузли, що мають малий вихід, забезпечать малу величину для приросту a_i . Проте інші вузли в області зв'язку збуджуватимуться, тим самим збільшуючи сигнал INHIB та значення b_i .

В процесі навчання ваги кожного вузла у шарі 2 налаштовуються таким чином, що разом вони складають шаблон, що відповідає образам, які часто пред'являються в процесі навчання. При пред'явленні схожого образу, коли шаблон відповідає йому, вузол виробляє великий вихідний сигнал. Образ, що сильно відрізняється, виробляє малий вихід і зазвичай пригнічується конкуренцією.

Латеральне гальмування. На рис. 13.4 показано, що кожен нейрон шару 2 отримує латеральне гальмування від нейронів, розташованих в його області конкуренції. Гальмівний нейрон підсумовує входи від усіх нейронів в області конкуренції і виробляє сигнал, спрямований на гальмування цільового нейрона. Цей метод є ефективним, але з обчислювальної точки зору повільним. Він охоплює велику систему із зворотним зв'язком, що включає кожен у шарі, і для його стабілізації може знадобитися велика кількість обчислювальних ітерацій.

Для прискорення обчислень використовується *метод прискореного латерального гальмування* (рис. 13.5). Тут додатковий вузол латерального гальмування обробляє вихід кожного збуджуючого вузла для моделювання необхідного латерального гальмування (LAT_INGIB). Спочатку він визначає сигнал, що дорівнює сумарному гальмівному впливу в області конкуренції:

$$\text{LAT_INGIB} = \sum_i w_i \text{OUT}_i, \quad (13.10)$$

де OUT_i – вихід i -го нейрона в області конкуренції; w_i – вага зв'язку від цього нейрона до латерально-гальмівного нейрона; w_i вибирається таким чином, що $\sum_i w_i = 1$.

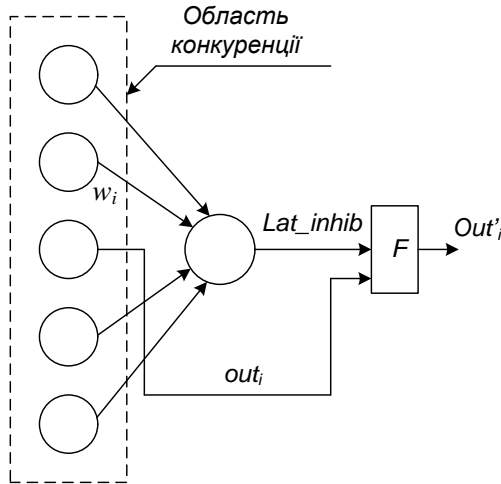


Рис. 13.5. Прискорене гальмування

Вихід гальмівного нейрона OUT' потім обчислюється наступним чином:

$$\text{OUT}' = \frac{1 + \text{OUT}_i}{1 + \text{LAT_INHIB}} - 1. \quad (13.11)$$

Завдяки тому, що всі обчислення, пов'язані з таким типом латерального гальмування, є нерекурсивними, їх можна провести за один прохід для шару, тим самим визначаючи зростання ефективності прискореного гальмування через економію в обчисленнях.

Цей метод латерального гальмування вирішує і іншу складну проблему. Припустимо, що вузол в шарі 2 збуджується сильно, але збудження сусідніх вузлів зменшується поступово із збільшенням відстані. При використанні звичайного латерального гальмування навчатиметься лише центральний вузол. Інші вузли визначають, що центральний вузол в їх області конкуренції має вищий вихід. Із запропонованою системою латерального гальмування такої ситуації статися не може. Безліч вузлів можуть навчатися одночасно, і процес навчання є достовірним.

Рецепторна область. Аналіз, що проводився до цього моменту, був спрощений розглядом лише одновимірних шарів. Насправді когнітрон конструювався як каскад двовимірних шарів, причому в даному шарі кожний нейрон отримує входи від набору нейронів на частині двовимірного плану, що складає його область зв'язку в попередньому шарі.

З цієї точки зору когнітрон організований подібно до зорової кори людини, що є тривимірною структурою, яка складається з декількох різних шарів. Виявляється, що кожний шар мозку реалізує різні рівні узагальнення: вхідний шар чутливий до простих образів, таких, як лінії, і їх орієнтації в певних областях візуальної області, тоді як реакція інших шарів є складнішою, абстрактнішою і незалежною від позиції образу [147].

Аналогічні функції реалізовані в когнітроні шляхом моделювання організації зорової кори. На рис. 13.6 показано, що нейрони когнітрона у шарі 2 реагують на певну невелику область вхідного шару 1. Нейрон у шарі 3 пов'язаний з набором нейронів шару 2, тим самим реагуючи побічно на більш широкий набір нейронів шару 1. Так само нейрони в наступних шарах чутливі до більш широких областей вхідного образу доти, поки у вихідному шарі кожний нейрон не стане реагувати на все вхідне поле.

Якщо область зв'язку нейронів має постійний розмір у всіх шарах, потрібна велика кількість шарів для перекриття всього вхідного поля вихідними нейронами. Кількість шарів може бути зменшена шляхом розширення області зв'язку в подальших шарах. На жаль, результатом цього може стати настільки велике перекриття областей зв'язку, що нейрони вихідного шару матимуть однакову реакцію. Для вирішення цієї проблеми може бути використане розширення області конкуренції. Оскільки в даній області конкуренції може збудитися лише один вузол, вплив малої різниці в реакціях нейронів вихідного шару посилюється.

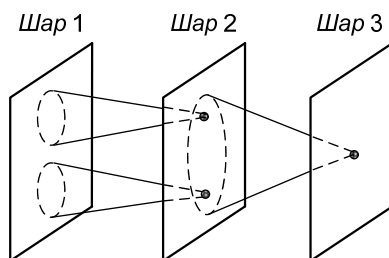


Рис. 13.6. Области зв'язків когнітрона

В альтернативному варіанті зв'язки з попереднім шаром можуть бути розподілені імовірно з більшістю синаптичних зв'язків в обмеженій області та з довгими з'єднаннями, що зустрічаються набагато рідше. Це відображає імовірнісний розподіл нейронів в мозку, що дозволяє у когнітроні кожному нейрону вихідного шару реагувати на повне вхідне поле за наявності обмеженої кількості шарів.

Недоліком когнітрона є його нездатність (або дуже велика утрудненість) у розпізнаванні образів незалежно від перетворень (обертань, спотворень тощо) останніх.

13.3. Неокогнітрон

У спробах покращити когнітрон була розроблена потужна парадигма, названа *неокогнітрон* [112; 113; 116]. Тоді як когнітрон і неокогнітрон мають певну схожість, між ними також існують фундаментальні відмінності, пов'язані з еволюцією досліджень авторів. Обидва зразки є багаторівневими ієрархічними сітками, організованими аналогічно зоровій корі. В той же час неокогнітрон більш відповідає моделі зорової системи. В результаті неокогнітрон є набагато потужнішою парадигмою з точки зору здатності розпізнавати образи незалежно від їх перетворень, обертань, спотворень і змін масштабу. Як і когнітрон, неокогнітрон використовує самоорганізацію в процесі навчання, хоча була описана версія [117], в якій замість цього використовувалося кероване навчання.

Неокогнітрон запропонований Фукушимом у 1980 р. і орієнтований на моделювання зорової системи людини. Він отримує на вході двовимірні образи, аналогічні зображенням на сітчастій оболонці ока, і обробляє їх в подальших шарах аналогічно тому, як це було виявлено в зоровій корі людини. Звичайно, в неокогнітроні немає нічого, що обмежує його використання лише для обробки візуальних даних, він досить універсальний і може використовуватись як узагальнена система розпізнавання образів.

У зоровій корі були виявлені вузли, що реагують на такі елементи, як лінії і кути певної орієнтації. На вищих рівнях вузли реагують на складніші і абстрактніші образи, такі, як кола, трикутники і прямокутники. На ще вищих рівнях міра абстракції зростає доти, поки не визначаться вузли, що реагують на обличчя і складні форми. У загальному випадку вузли на вищих рівнях отримують вхід від групи низькорівневих вузлів і тому реагують на ширшу область візуального поля. Реакції вузлів більш високого рівня менш залежать від позиції і стійкіші до спотворень.

Найсуттєвіша відмінність неокогнітрона від повнозв'язного багатошарового перцептрона – це *значно менша кількість використовуваних ваг* при тієї ж кількості нейронів. Так виходить за рахунок “трюку”, який дозволяє неокогнітрона визначати образи незалежно від їх положення. Площина простих клітин – це по суті один нейрон, ваги якого визначають ядро згортки. Це ядро застосовується до попереднього шару, “пробігаючи” його в усіх можливих позиціях. Власне нейрони кожної площини задають своїми зв'язками координати цих позицій. Це приводить до того, що всі нейрони шару простих клітин слідкують за тим, чи не з'явиться в їх рецептивному полі образ, відповідаючий ядру. Тобто, якщо такий образ зустрінеться де-небудь у вхідному для цього шару сигналі, це буде виявлено хоча б одним простим нейроном і викличе активність відповідного складного нейрона. Така хитрість дозволяє знайти характерний образ у будь-якому місці, де б він не з'явився. Проте, слід пам'ятати, що це саме хитрість, і вона не дуже відповідає роботі реальної кори.

Структура. Неокогнітрон має ієрархічну структуру, орієнтовану на моделювання зорової системи людини. Він складається з послідовності оброблювальних шарів, організованих в ієрархічну структуру (рис. 13.7). Вхідний образ подається на перший шар і передається через *площини*, відповідні подальшим шарам доти, поки не досягне вихідного шару, в якому ідентифікується образ, який розпізнається.

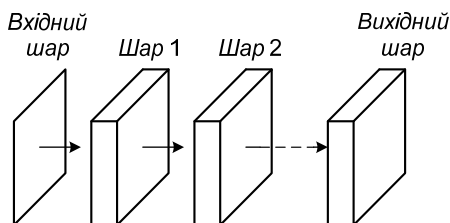


Рис. 13.7. Структура шарів неокогнітрона

Структура неокогнітрона складна для представлення у вигляді діаграми, але концептуально проста. Щоб підкреслити його багаторівневість (для спрощення графічного подання), використовується аналіз верхнього рівня. Неокогнітрон показаний багатошаровим, *шари складаються з набору площин*, і *площі складаються з вузлів*.

Характеристика шарів. Кожний шар неокогнітрона складається з двох масивів площин (рис. 13.8). Масив площин, що містить прості вузли, отримує виходи попереднього шару, виділяє певні образи і потім передає їх в масив площин, що містить комплексні вузли,

де вони обробляються так, щоб зробити виділені образи менш позиційно залежними.

Площини вузлів. Всередині шару площини простих і комплексних вузлів існують парами, тобто для площини простих вузлів існує одна площина комплексних вузлів, що обробляє її виходи. Кожна площина може бути візуально представлена як двовимірний масив вузлів.

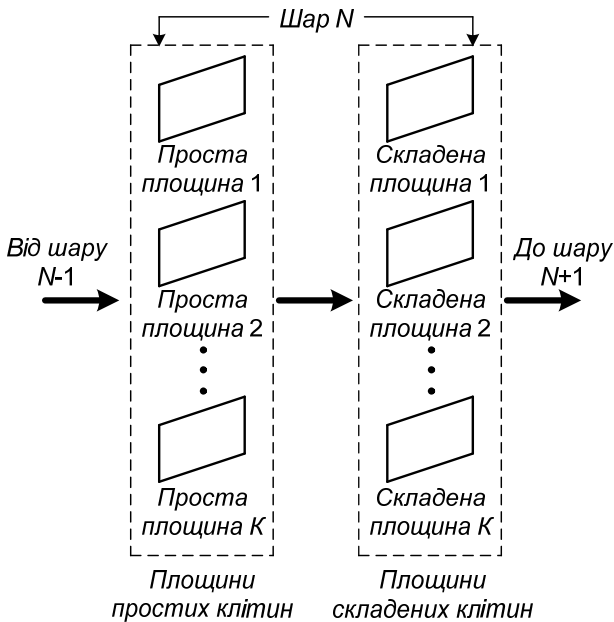


Рис. 13.8. Структура площин неокогнітрона

Прості вузли. Всі вузли в даній площині простих вузлів реагують на один і той самий образ. Як показано на рис. 13.9, площина простих вузлів являє собою масив вузлів, кожен з яких “налаштовується” на один специфічний вхідний образ. Кожний простий вузол чутливий до обмеженої області вхідного образу, що зветься його *рецепторною областю*. Наприклад, всі вузли у верхній площині простих вузлів на рис. 13.9 реагують на “С”. Вузол реагує, якщо “С” зустрічається у вхідному образі і якщо “С” виявлене в його рецепторній області.

На рис. 13.9 показано, що інші площини простих вузлів в цьому шарі можуть реагувати на обертання “С” на 90° , інші на обертання на 180° і так далі. Якщо мають бути виділені інші літери (та їх спотворені версії), додаткові площини необхідні для кожної з них.

Рецепторні області вузлів в кожній площині простих вузлів перекриваються з метою покриття всього вхідного образу цього шару. Кожний вузол отримує входи від відповідних областей всіх площин комплексних вузлів в попередньому шарі. Отже, простий вузол реагує на появу свого образу в будь-якій складній площині попереднього шару, якщо він виявиться усередині його рецепторної області.

Комплексні вузли. Завданням комплексних вузлів є зменшення залежності реакції системи від позиції образів у вхідному полі. Для досягнення цього кожний комплексний вузол отримує в якості вхідного образу виходи набору простих вузлів з відповідної площини того ж шару. Ці прості вузли покривають неперервну область простої площини, що зветься *рецепторною областю комплексного вузла*. Збудження будь-якого простого вузла в цій області є достатнім для збудження даного комплексного вузла. Таким чином, комплексний вузол реагує на той самий образ, що і прості вузли у відповідній йому площині, але він менш чутливий до позиції образу, ніж будь-який з них.

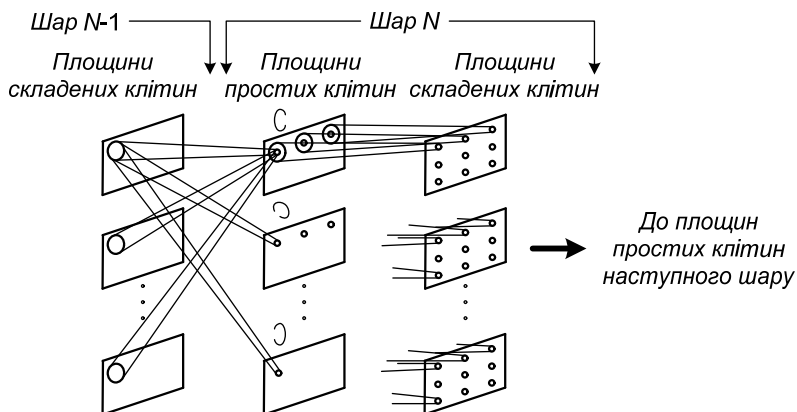


Рис. 13.9. Система неоконітрона

Таким чином, кожний шар комплексних вузлів реагує на більш широку область вхідного образу, ніж це робилося в попередніх шарах. Ця прогресія зростає лінійно від шару до шару, приводячи до необхідного зменшення позиційної чутливості системи в цілому.

Узагальнення. Кожний нейрон в шарі, близькому до вхідного, реагує на певні образи у визначеному місці, такі як кут з певною орієнтацією – в заданій позиції. Кожний шар в результаті цього має абстрактнішу, менш специфічну реакцію у порівнянні з попереднім; вихідний шар реагує на повні образи, показуючи високу міру незалеж-

ності від їх положення, розміру і орієнтації у вхідному полі. При використанні в якості класифікатора комплексний вузол вихідного шару з найбільшою реакцією реалізує виділення відповідного образу у вхідному полі. В ідеальному випадку це виділення нечутливе до позиції, орієнтації, розмірам або іншим спотворенням.

Обчислення. Прості вузли в неокогнітроні мають такі самі характеристики, що і описані для когнітрона, і використовують ті ж формули для визначення їх виходу. Тому тут вони повторно не наводяться.

Гальмівний вузол виробляє вихід, пропорційний квадратному кореню із зваженої суми квадратів його входів. Зазначимо, що входи в гальмівний вузол ідентичні входам відповідного простого вузла, і область включає область відповіді в усіх комплексних площинах. У символічному вигляді (рис. 13.10):

$$u = \sqrt{\sum_i (w_i u_i)^2}, \quad (13.12)$$

де u – вихід гальмівного вузла; i – область над усіма комплексними вузлами, з якими зв'язаний гальмівний вузол; w_i – вага i -го синаптичного зв'язку від комплексного вузла до гальмівного вузла; u_i – вихід i -го комплексного вузла.

Ваги w_i вибираються такими, що монотонно зменшуються із збільшенням відстані від центру області реакції, при цьому сума їх значень має дорівнювати одиниці.

Навчання неокогнітрона. Тільки прості вузли мають ваги, що налагоджуються. Це ваги зв'язків, що сполучають вузол з комплексними вузлами в попередньому шарі і мають змінну силу синапсів, яка налагоджується так, щоб виробити максимальну реакцію на певні стимулюючі властивості. Деякі з цих синапсів є збуджуючими і прагнуть збільшити вихід вузлів, тоді як інші є гальмівними і зменшують вихід вузла.

На рис. 13.10 показана повна структура синаптичних зв'язків між простим вузлом і комплексними вузлами в попередньому шарі. Кожний простий вузол реагує лише на набір комплексних вузлів усередині своєї рецепторної області. Крім того, існує гальмівний вузол, що реагує на ті ж самі комплексні вузли. Ваги синапсів гальмівного вузла не навчаються і вибираються так, щоб вузол реагував на середнє значення виходів усіх вузлів, до яких він підключений. Єдиний гальмівний синапс від гальмівного вузла до простого вузла навчається, як і інші синапси.

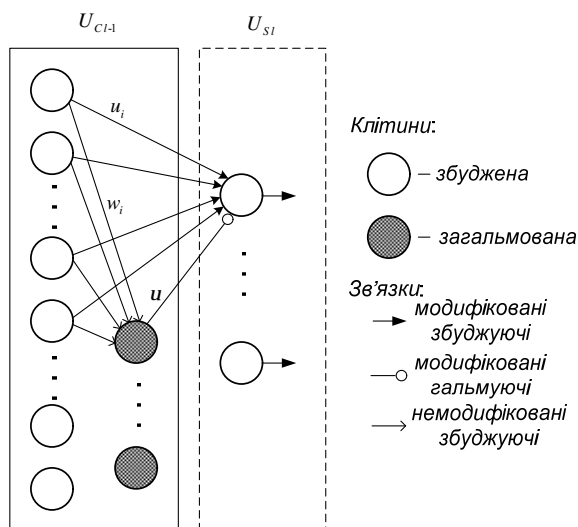


Рис. 13.10. Зв'язки від складних кліток одного рівня до простих кліток наступного рівня

Навчання без вчителя. Для навчання неокогнітрона на вхід мережі подається образ, який необхідно розпізнати, і ваги синапсів налаштовуються шар за шаром, починаючи з набору простих вузлів, найближчих до входу. Величина синаптичного зв'язку від кожного комплексного вузла до даного простого вузла збільшується тоді і лише тоді, коли задовольняються наступні дві умови:

- комплексний вузол реагує;
- простий вузол реагує сильніше, ніж будь-який з його сусідніх (всередині його області конкуренції).

Таким чином, простий вузол навчається реагувати сильніше на образи, що з'являються найчастіше в його рецепторній області. Якщо розпізнаваний образ відсутній на вході, гальмівний вузол запобігає випадковому збудженню.

Математичний опис процесу навчання та метод реалізації латерального гальмування аналогічні описанам для когнітрона, тому тут вони не повторюються. Необхідно відзначити, що виходи простих і комплексних вузлів є аналоговими, безперервними і лінійними і що алгоритм навчання передбачає їх додатність.

Коли вибирається простий вузол, ваги синапсів якого мають бути збільшені, він розглядається як представник всіх вузлів в площині, викликаючи збільшення їх синаптичних зв'язків на тому ж самому образі. Отже, всі вузли в площині навчаються розпізнавати одні й ті ж властивості, і після навчання робитимуть це незалежно від позиції образу в полі комплексних вузлів в попередньому шарі.

Ця система має важливу здібність до самовідновлення. Якщо даний вузол вийде з ладу, буде знайдений інший вузол, що реагує сильніше, і цей вузол буде навчений розпізнаванню вхідного образу, тим самим перекриваючи дії свого “товариша”, що відмовив.

Навчання з вчителем. В роботах [116; 117] описано самоорганізоване некероване навчання. Поряд з цими вражаючими результатами були опубліковані звіти про інші експерименти, в яких використовувалося навчання з вчителем [118]. Тут необхідна реакція кожного шару заздалегідь визначається експериментатором. Потім ваги налаштовуються з використанням звичайних методів для вироблення необхідної реакції. Наприклад, вхідний шар налаштовується для розпізнавання відрізків в різних орієнтаціях, що багато в чому аналогічно першому шару обробки зорової кори. Подальші шари навчаються реагувати на складніші і абстрактніші властивості доти, поки у вихідному шарі необхідний образ не буде виділений.

Реалізація навчання. У звичайних конфігураціях рецепторне поле кожного нейрона зростає при переході до наступного шару. Проте кількість нейронів в шарі зменшуватиметься при переході від вхідних до вихідних шарів. Нарешті, вихідний шар має лише один нейрон в площині складних вузлів. Кожний такий нейрон представляє певний вхідний образ, якому сітка була навчена. В процесі класифікації вхідний образ подається на вхід неокогнітрона, і обчислюються виходи шар за шаром, починаючи з вхідного шару. Оскільки лише невелика частина вхідного образу подається на вхід кожного простого вузла вхідного шару, деякі прості вузли реєструють наявність характеристик, яким вони навчені, і збуджуються. У наступному шарі виділяються складніші характеристики як певні комбінації виходів комплексних вузлів. Шар за шаром властивості комбінуються у все зростаючому діапазоні; виділяються загальніші характеристики і зменшується позиційна чутливість.

В ідеальному випадку лише один нейрон вихідного шару повинен збудитися. Насправді зазвичай збуджуватиметься декілька нейронів з різною силою, і вхідний образ має бути визначений з урахуванням співвідношення їх виходів. Якщо використовується сила латерального гальмування, збуджуватися буде лише нейрон з максимальним виходом. Проте це часто не є кращим варіантом. На практиці проста функція від невеликої групи найбільш сильно збуджених нейронів часто покращуватиме точність класифікації.

13.4. Висновки

Як когнітрон, так і неокогнітрон справляють велике враження з точки зору точності, з якою вони моделюють біологічну нервову систему. Той факт, що ці системи показують результати, що імітують

деякі аспекти здібностей людини до навчання та пізнання, наводять на думку, що наше розуміння функцій мозку наближається до рівня, здатного принести практичну користь.

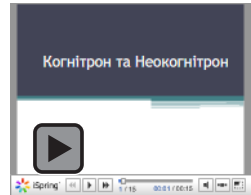
Неокогнітрон є складною системою і потребує суттєвих обчислювальних ресурсів. Саме з цих міркувань здається малоімовірним, що такі системи реалізують оптимальний інженерний розв'язок сьогоденних проблем розпізнавання образів. Тому навіть не дивлячись на те, що багато з яких підходів, які уявлялися нереалізовуваними декілька років тому, є загальновизнаними сьогодні і можуть виявитися тривіальними через декілька років, реалізація моделей неокогнітрона на універсальних комп'ютерах є безперспективною. Необхідно досягти тисячократних покращень вартості та продуктивності комп'ютерів за рахунок спеціалізації архітектури і впровадження технології СБІС, щоб зробити неокогнітрон практичною системою для розв'язання складних проблем розпізнавання образів, проте ані та, ані будь-яка інша модель штучних нейронних мереж не повинні відкидатися тільки на підставі їх високих обчислювальних вимог.

Контрольні завдання

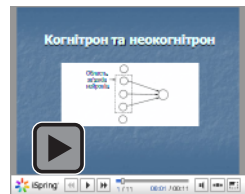
1. Сформулювати проблеми, якими супроводжується процес розпізнавання образів.
2. Пояснити роль пресинаптичних та простинаптичних нейронів, збуджуючих та гальмівних вузлів в моделі когнітрона Куніхіко Фукушими, а також розкрити поняття “області зв'язків в нейроні” та “конкуренція”.
3. Прокоментувати особливості навчання когнітрону, виокремивши роль збуджуючих та гальмівних нейронів при цьому.
4. Розтлумачити поняття “латеральне гальмування” і розкрити метод прискореного латерального гальмування.
5. Прокоментувати відмінність рецепторної області зору з можливістю сприймати багатовимірні об'єкти від спрощеного варіанту розгляду лише одновимірними шарами в когнітроні.
6. Навести структуру і дати характеристику шарів в неокогнітроні.
7. Охарактеризувати неокогнітрон і розкрити його найсуттєвішу відмінність від повнозв'язного багат шарового перцептрона.
8. Навести характеристику шарів неокогнітрону, прокоментувавши властивості площин вузлів і розкривши роль простих і комплексних вузлів в структурі.
9. Навести особливості навчання неокогнітрону.

Презентації

1. Когнітрон та Неокогнітрон



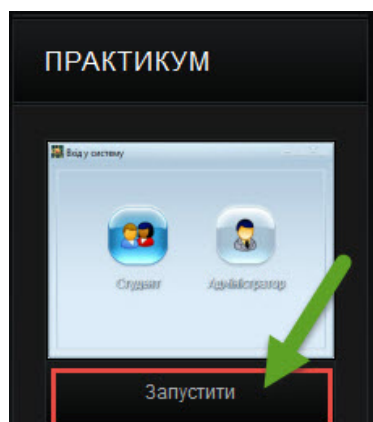
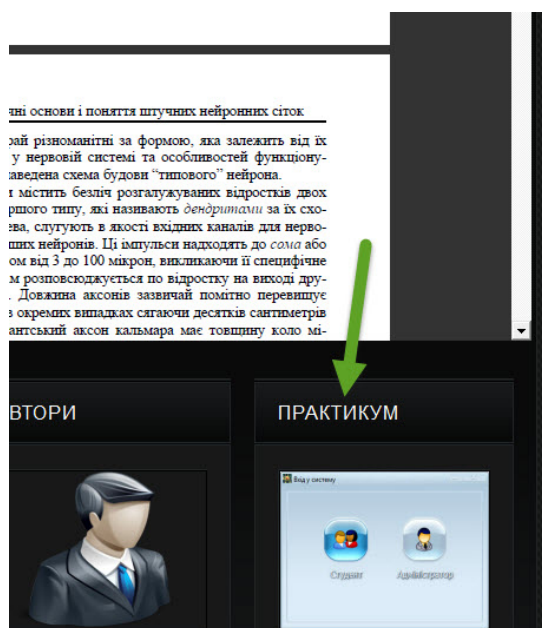
2. Когнітрон та неокогнітрон



Практичні заняття

Когнітрон та неокогнітрон

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 14

Згортаючі нейронні сітки

Я милого впізнаю по ходу

Гарик Сукачев

Крапля каміння довбить

Овідій Назон Публій

14.1. Властивості згортаючої нейронної сітки

Загальні зауваження. Як правило, під “звичайними” НС розуміють повнозв’язні нейронні сітки прямого поширення зі зворотним поширенням похибки. Як вже зазначалось (див. розд. 1.4), такі НС дістали назву прямоспрямованих (ПСНС).

При виборі методу пред’явлення даних на вхід НС найпростішим і майже безальтернативним для ПСНС є виразити двовимірну матрицю зображення у вигляді одновимірного вектора. Це означає, що, наприклад, для зображення рукописної цифри розмірністю 28×28 необхідні 784 входи, що вже немало. Далі треба вибрати архітектуру НС, але досі не існує методів, дозволяючих однозначно визначити топологію сітки виходячи з опису розв’язуваної задачі (зазначимо тут, що в розд. 25 підручника наводяться рекомендації щодо обґрунтування такого вибору). Для важкоформалізованих задач такий метод навряд чи буде колись створений. Крім того, існує безліч різних методик редукції сітки (наприклад, OBD [211]), а також всілякі евристики та емпіричні правила, одне з яких наголошує, що кількість нейронів у схованому шарі повинна бути *хоча б на порядок більшою* за кількість входів. Якщо прийняти до уваги, що перетворення із зображення в індикатор класу досить складне і суттєво нелінійне, одного схованого шару для означеної вище задачі виявиться замало, і тоді кількість нейронів наблизиться до 15000 (10000 у другому шарі та 5000 у третьому). За таких конфігураційних умов кількість настроюваних і навчуваних зв’язків в сітці перевищить 10 млн. між входами і першим схованим шаром, а також 50 млн. між першим і другим схованими шарами, та ще й 50 тис. між другим шаром і шаром на виході. Всього виходить приблизно *60 000 000 настроюваних зв’язків*, для кожного з яких треба обчислювати градієнт похибки.

Крім того, при перетворенні зображень в лінійний ланцюжок обов’язково дещо безповоротно втрачається, причому з кожним ша-

ром ця втрата тільки збільшується, а саме – втрачається топологія зображення (взаємозв'язок між окремими його складовими). При цьому задача розпізнавання має на увазі “вміння” нейросітки бути сталою до невеликих зсувів, поворотів і змін масштабу зображення, тобто вона повинна витягати з даних якісь інваріанти, які не залежать від особливостей його креслення. Через усе це й виникають вимоги до НС щодо обмеженості обчислювальної складності і, в той же час, інваріантності до всіляких спотворень зображень.

Розв'язок цієї проблеми був запропонований американським вченим французького походження Яном ЛеКуном з урахуванням праць нобелівських лауреатів в галузі медицини Торстеном Візелем (Torsten Nils Wiesel) та Девідом Х'юбелем (David H. Hubel), які дослідили зорову кору головного мозку кішки і виявили, що існують так звані прості клітини, які вкрай сильно реагують на прямі лінії під різними кутами, і складні клітини, які реагують на рух ліній в одному напрямку. Як результат, Ян ЛеКун запропонував використовувати так звані *згортаючі нейронні сітки* (ЗНС) (англ. *convolutional neural network, CNN*) [163], які багато в чому базуються на ідеях неокогнітивістської парадигми Фукушими (див. розд. 13.2) і входять до складу технологій глибокого навчання (англ. *deep learning*).

В наведених вище роботах Ян ЛеКун показав, як можна використати метод зворотного поширення похибки для навчання НС, архітектура яких, як і у неокогнітрона, віддалено нагадує побудову кори мозку. Основні уявлення, які були використані – це *локальність зон сприйняття* та *ділення нейронів за функціями* всередині одного шару.

Твердження 14.1. *Локальність сприйняття* означає, що нейрон, отримуючий інформацію, слідує не за всім вхідним простором сигналів, а тільки за його частиною; така область спостереження називається *рецептивним полем нейрона*.

Означення 14.1. *Рецептивне поле нейрона* – це простір *рецепторів*, який впливає на роботу нейрона.

Твердження 14.2. *Рецепторами* є нейрони, безпосередньо сприймаючі зовнішні сигнали.

Уявимо нейронну сітку, яка складається з двох шарів, де перший шар – шар рецепторів, а другий шар – нейрони, з'єднані з рецепторами. Для кожного нейрона другого шару ті рецептори, що мають з ним контакт – це й є його рецептивне поле.

Тепер візьмемо складну багат шарову сітку. Чим далі ми будемо відділятися від входу, тим складніше буде указати, які рецептори та як впливають на активність нейронів, які знаходяться в глибині. З певного моменту може статися, що для якого-небудь нейрона всі існуючі рецептори можуть бути названі його рецептивним полем. В такій ситуації рецептивним полем нейрона хочеться назвати тільки ті нейрони, з якими він має безпосередній синаптичний контакт. Щоб розвести ці поняття, прийнято називати простір вхідних рецепторів – початковим рецептивним полем. А той простір нейронів, що взаємодіє з нейроном безпосередньо – локальним рецептивним полем або просто рецептивним полем, без додаткового уточнення.

Твердження 14.3. Ділення нейронів за функціями пов’язане з виявленням в первинній зоровій корі двох основних типів нейронів: *простих* (simple) *нейронів*, реагуючих на розташований у визначеному місці їх початкового рецептивного поля стимул; *складних* (complex) *нейронів*, виявляючих активність на стимул, незалежно від його положення.

Ідея надточних НС полягає в чергуванні згортаючих (англ. *convolution layers* – *C-layers*), субдискретизуючих (англ. *subsampling layers*, або підвибірки – *S-layers*) шарів та наявності повнозв’язних (*F-layers*) шарів на виході. Структура сітки – прямокутна (без зворотних зв’язків), принципово багат шарова. Для навчання використовуються стандартні методи, частіше за все – метод зворотного поширення похибки. Функція активації нейронів (передаткова функція) – будь-яка, за вибором досліджувача.

Парадигми згортаючої НС. Назву архітектура сітки одержала через наявність операції згортки, суть якої розкривається нижче.

Така архітектура містить в собі 3 основних парадигми:

- локальне сприйняття;
- поділяючі ваги;
- субдискретизація.

Локальне сприйняття припускає, що на вхід одного нейрона подається не все зображення (або виходи попереднього шару), а лише деяка його область. Такий підхід дозволяє зберігати топологію зображення від шару до шару.

Концепція *поділяючих ваг* припускає, що для значної кількості зв’язків використовується дуже обмежений набір ваг. Іншими словами, якщо на вході є зображення розміром 32x32 пікселя, то кожен

з нейронів наступного шару приймає на вхід тільки незначну ділянку цього зображення, наприклад, 5×5 , причому кожний з фрагментів буде опрацьований одним і тим же набором. При цьому самих наборів ваг може бути багато, але кожний з них буде застосований до *всього* зображення. Такі набори часто називають ядрами (kernels). Легко підрахувати, що навіть для 10 ядер розміром 5×5 для вхідного зображення розмірами 32×32 кількість зв'язків приблизно дорівнюватиме 256000 (порівняємо з 10 млн.), а кількість настроюваних параметрів *всього* 250.

До того ж якість розпізнавання покращиться через те, що таке штучне уведення обмеження на ваги покращить узагальнюючі властивості (generalization) НС, що в решті решт позитивно відбивається на спроможності сітки знаходити інваріанти в зображенні і реагувати головним чином на них, не звертаючи уваги на інший шум. Тут можна нагадати аналогію при розв'язанні подібних задач (розпізнавання зображень) із застосуванням двовимірних фільтрів. Фільтр уявляє собою матрицю коефіцієнтів, зазвичай задану вручну. Ця матриця застосовується до зображення за допомогою математичної операції, яка дістала назву *згортки* і суть якої полягає в тому, що кожний фрагмент зображення помножується на матрицю (ядро) згортки поелементно, і результат підсумовується та записується в аналогічну позицію зображення на виході. Основна властивість таких фільтрів полягає в тому, що значення їх виходу є тим більшим, чим більше фрагмент зображення схожий на сам фільтр. Таким чином, зображення, згорнуте з деяким ядром, дає інше зображення, кожний піксель якого буде означати ступінь схожості фрагмента зображення на фільтр. Іншими словами, це буде *мапа ознак* (англ. *feature map*).

Саме так відбувається процес розповсюдження сигналу в С-шарі, наведений на рис. 14.1.

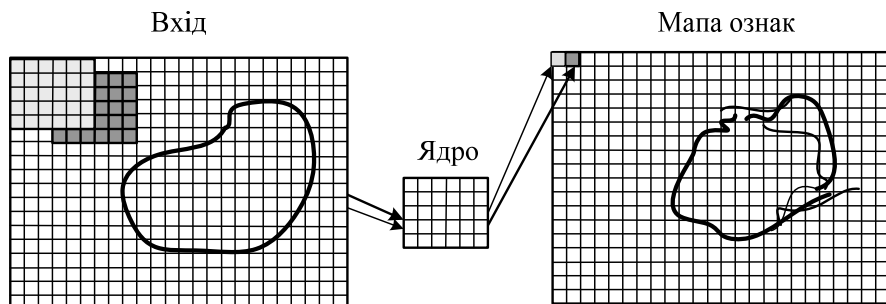


Рис. 14.1. Процес розповсюдження сигналу в С-шарі

Отже, кожний фрагмент зображення поелементно помножується на невелику матрицю ваг (ядро), результат підсумовується. Ця сума є пікселем зображення на виході, яке називається мапою ознак. Зважена таким чином сума входів ще пропускається через активаційну функцію (як і у будь-якій іншій НС), що може відбуватися, наприклад, і в S -шарі. В ідеальному випадку не різні фрагменти проходять послідовно через ядро, а паралельно усе зображення проходить через ідентичні ядра. Крім того, кількість ядер (наборів ваг) визначається користувачем і залежить від кількості ознак, яку необхідно виокремити. Інша особливість згортаючого шару полягає в тому, що він трохи зменшує зображення за рахунок крайових ефектів.

Суть *субдискретизації* та S -шарів полягає в зменшенні просторової розмірності зображення, коли вхідне зображення грубо (усередненням) зменшується в задану кількість разів (частіше за все у 2 рази, хоча може бути і нерівномірне змінення, наприклад, 2 по вертикалі та 3 по горизонталі). Субдискретизація необхідна для забезпечення інваріантності до масштабу.

Чергування шарів дозволяє складати мапи ознак з мап ознак, що на практиці означає спроможність розпізнавання складних ієрархій ознак.

Зазвичай після проходження декількох шарів карта ознак вироджується у вектор або навіть скаляр, проте таких карт ознак стає сотні. В такому виді вони подаються на один-два шари повнозв'язної сітки. В шарі на виході застосовуються різні активаційні функції, в тому числі тангенціальні (для найпростіших випадків) або радіально-базисні.

14.2. Принципи організації та функціонування згортаючої нейросітки

Порядок організації згортаючої НС. В звичайному перцептроні, який уявляє собою повнозв'язну НС, кожний нейрон зв'язаний зі всіма нейронами попереднього шару, і кожний зв'язок має свій персональний ваговий коефіцієнт. В згортаючій нейронній сітці в операції згортки використовується лише обмежена матриця ваг незначного розміру, яку "пересувають" по всьому опрацьовуваному шару (в самому початку – безпосередньо по вхідному зображенню), формуючи після кожного зсуву сигнал активації для нейрона наступного шару з аналогічною позицією. Матриця ваг, яка також є набором ваг (*ядро згортки*), побудована так, що графічно кодує яку-небудь одну ознаку, наприклад, наявність похилої лінії під визначеним кутом.

Тоді наступний шар, отриманий в результаті операції згортки такою матрицею ваг, показує наявність даної похилої лінії в опрацьовуваному шарі та її координати, формуючи карту ознак. Звісно, в згортаючій НС набір ваг не один, а ціла гама, кодуюча всілякі лінії та дуги під різними кутами. Проходження кожним набором ваг формує свій власний екземпляр мапи ознак, перетворюючи нейронну сітку у багатовимірну (багато незалежних мап ознак на одному шарі). Також слід зауважити, що при “перебиранні” шару матрицею ваг її пересувають звичайно не на повний крок (розмір цієї матриці), а на невелику віддаль. Так, наприклад, при розмірності матриці ваг 5×5 її зсувають на один або два нейрона (пікселя) замість п'яти, щоб не “переступити” шукану ознаку.

Операція субдискретизації виконує зменшення розмірності сформованих мап ознак. В даній архітектурі сітки вважається, що інформація щодо факту наявності шуканої ознаки важливіша за точне знання її координат, через що з декількох сусідніх нейронів мапи ознак вибирається максимальний і приймається за один нейрон мапи ознак зменшеної розмірності. Також іноді застосовують операцію визначення середнього між сусідніми нейронами. За рахунок даної операції, крім прискорення подальших обчислень, сітка стає більш інваріантною до масштабу вхідного зображення.

Таким чином, повторюючи один за одним декілька шарів згортки і субдискретизації будується згортаюча НС. На виході сітки часто додатково встановлюють декілька шарів повнозв'язної НС (перцептрон), на вхід якої подаються кінцеві мапи ознак.

Якщо в першому шарі ядро згортки проходить тільки по одному початковому зображенню, то на внутрішніх шарах одне і те ж ядро проходить паралельно по всіх мапах ознак цього шару, а результат згортки підсумовується, формуючи (після проходження функції активації) одну мапу ознак наступного шару, відповідну цьому ядру згортки.

14.3. Архітектура згортаючої нейронної сітки

Нейрологічне обґрунтування побудови ЗНС. За кожним шаром згортки слідує обчислювальний шар, реалізуючий *локальне усереднення і підвибірку*. Завдяки цього досягається зменшення розділення для мап ознак. Ця операція приводить до зменшення чутливості вихідного сигналу відображення ознак до зміщення та іншим фор-

мам деформації. Таке побудування сіток згортки має нейробіологічне обґрунтування, описане в працях Ян ЛеКуна, присвячених локально чутливим та вибірково-орієнтаційним нейронам зорового апарату.

Приклад архітектурної побудови ЗНС. На рисунку 14.2 показана архітектурна схема згортаючої сітки, що складається з одного вхідного, чотирьох схованих і одного нейронного шару на виході і призначена для розпізнавання рукописних цифр.

Вхідний шар складається з матриці 28×28 сенсорних вузлів і отримує зображення різних символів, які попередньо зміщені до центру і нормалізовані за розміром. Після цього обчислювальні шари по чергово реалізують операції згортки і підвибірки, як описується далі.

Перший схований шар виконує згортку. Він складається з чотирьох мап ознак, кожна з яких представляє матрицю з 24×24 нейронів. Кожному нейрону відповідає поле чутливості розміром 5×5 .

Другий схований шар виконує підвибірку і локальне усереднення. У нього теж чотири мапи ознак, але тепер вони містять матриці розміром 12×12 нейронів. Кожному нейрону відповідають рецепторне поле розміром 2×2 , налагоджуваний коефіцієнт, налагоджуваний поріг і сигмоїдальна функція. Налагоджуваний коефіцієнт і поріг визначають робочу область нейрона.

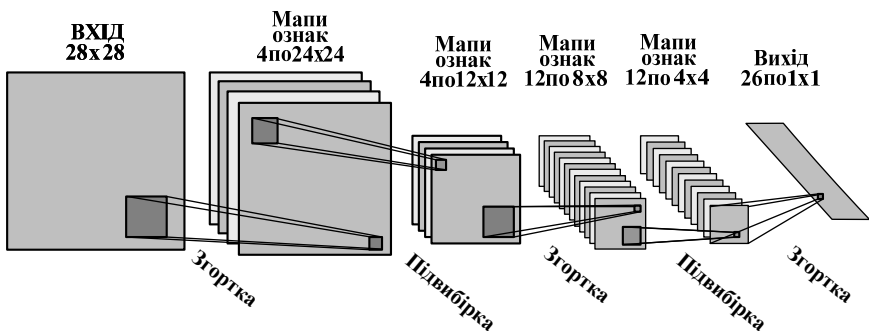


Рис. 14.2. Архітектура згортаючої нейронної сітки для розпізнавання рукописного тексту

Третій схований шар виконує повторну згортку. Він складається з 12 мап ознак, кожна з яких представляє собою матрицю 8×8 нейронів. Кожний нейрон цього шару може мати синаптичні зв'язки з різними мапами ознак попереднього схованого шару. В іншому випадку його робота була б аналогічною першому шару згортки.

Четвертий схований шар здійснює другу підвибірку і повторне локальне усереднення. Він складається з 12 мап ознак, однак на цей раз кожна мапа містить матрицю з 4x4 нейронів. В іншому випадку його робота була б аналогічною першому шару підвибірки.

Шар на виході здійснює останній етап згортки. Він складається з 26 нейронів, кожному з яких відповідає одна з 26 літер латинського алфавіту. Як і раніше, кожному нейрону відповідає поле розміром 4x4.

При послідовному проходженні шарів згортки і підвибірки був отриманий *безпірамідальний ефект*, тобто в кожному шарі згортки або підвибірки кількість мап збільшується у порівнянні з попереднім при одночасному зменшенні просторового розділення. Ідея чергування згортки і підвибірки була викликана чергуванням «простих» і «складних» клітин, описаним при дослідженні неокогнітрону.

14.4. Особливості навчання згортаючих нейронних сіток

Формальний підхід. Навчання розв'язуванню складних задач ЗНС реалізується з вчителем, при цьому використовуються сітки, архітектура яких задовольняє наступним обмеженням:

- *здобування ознак.* Кожний нейрон отримує вхідний сигнал від локального рецептивного поля (*receptive field*) попереднього шару, здобуваючи таким чином його локальні ознаки. Як тільки ознаку здобуто, її точне місцезрештування не має значення, оскільки встановлено її розташування відносно інших ознак;
- *відображення ознак (feature mapping).* Кожний обчислювальний шар сітки складається з множини мап ознак (*feature map*), кожна з яких має форму площини, на якій всі нейрони повинні спільно використовувати одну й ту ж множину синаптичних ваг.

Ця форма архітектурних обмежень має наступні переваги:

- *інваріативність до зміщення*, реалізовану за допомогою мап ознак з використанням згортки з ядром невеликого розміру, виконуючим функцію “сплющування”;
- *скорочення кількості вільних параметрів*, реалізоване за допомогою спільного використання синаптичних ваг.

Для того, щоб можна було розпочати навчання сітки, необхідно визначитись з тим, як вимірювати якість розпізнавання. Для

цього можна скористатись розповсюдженою в теорії НС функцією середньоквадратичної похибки (СКО, MSE) [162]:

$$\delta^{(p)} = 0,5 \left(O_6^{(p)} - O^{(c)}(I^{(p)}, W) \right)^2, \quad (14.1)$$

де $\delta^{(p)}$ – похибка розпізнавання для p -ї навчаючої пари; $O_6^{(p)}$ – бажаний вихід сітки; $O^{(c)}(I^{(p)}, W)$ – вихід сітки, який залежить від p -го входу $I^{(p)}$ та вагових коефіцієнтів W , куди входять ядра згортки, зсуви, вагові коефіцієнти S - та F - шарів.

Задача навчання полягає в такій налагоді ваг W , за якою вони для будь-якої навчаючої пари $(I^{(p)}, W)$ дають мінімальну похибку $\delta^{(p)}$.

Щоб обчислити похибку для всієї навчаючої просто береться середнє арифметичне за похибками для всіх навчаючих пар. Таку усереднену похибку позначимо як δ .

Для мінімізації функції похибки $\delta^{(p)}$ найефективнішими є градієнтні методи. Розглянемо сутність градієнтних методів на прикладі одновимірного випадку (тобто коли має місце всього одна вага). Якщо розкласти в ряд Тейлора функцію похибки $\delta^{(p)}$, то отримаємо наступне співвідношення:

$$\begin{aligned} \delta(W) &= \delta(W_n) + (W - W_n) \frac{d\delta(W_n)}{dW} + \\ &+ 0,5(W - W_n)^2 \frac{d^2\delta(W_n)}{dW^2} + \dots, \end{aligned} \quad (14.2)$$

де δ – та сама функція похибки; W_n – деяке початкове значення ваги.

Тепер для визначення екстремуму функції похибки (14.2) беремо її похідну за вагами, відкинувши члени вищі 2-го порядку:

$$\frac{d\delta(W)}{dW} = \frac{d\delta(W_n)}{dW} + (W - W_n) \frac{d^2\delta(W_n)}{dW^2}. \quad (14.3)$$

З виразу (14.3) випливає, що вагу, за якої значення функції похибки δ буде мінімальним, можна визначити з наступної залежності:

$$W_{\min} = W_n - \frac{d\delta(W_n)}{dW} \left(\frac{d^2\delta(W_n)}{dW^2} \right)^{-1}, \quad (14.4)$$

тобто оптимальна вага обчислюється як поточна мінус похідна функції похибки по вазі, поділена на другу похідну функції похибки. Для багатовимірного випадку (тобто для матриці ваг) все відбувається так само, окрім того, що перша похідна перетворюється в градієнт

(вектор частинних похідних), а друга похідна перетворюється в гессіан (матрицю других частинних похідних). При цьому можливі два варіанти:

1. При опущенні другої похідної має місце алгоритм найшвидшого градієнтного спуску.
2. При урахуванні другої похідної необхідна велика кількість обчислювальних ресурсів для підрахування повного гессіану та обертання його потім. Тому зазвичай гессіан замінюють більш простим методом, наприклад, алгоритмом Левенберга-Марквардта (АЛМ) (його апроксимацією за допомогою квадратного Якобіана).

Означення 14.2. *Алгоритм Левенберга-Марквардта* – це метод оптимізації, спрямований на розв'язання задач про найменші квадрати. Є альтернативою методу Ньютона. Може розглядатися як комбінація останнього з методом градієнтного спуску або як метод надійних інтервалів.

Алгоритм АЛМ призначений для оптимізації параметрів нелінійних регресійних моделей. Припускається, що в якості критерія оптимізації використовується середньоквадратична похибка моделі на навчаючій вибірці. Алгоритм полягає у послідовному наближенні заданих початкових значень параметрів до шуканого локального мінімуму.

Алгоритм відрізняється від методу спряжених градієнтів тим, що використовує матрицю Якобі-моделі, а не градієнт вектора параметрів. Від алгоритму Гаусса-Ньютона цей алгоритм відрізняється тим, що використовує *параметр регуляризації*.

При цьому необхідно вважати на те, що АЛМ вимагає опрацювання всієї навчаючої вибірки, тоді як алгоритм градієнтного спуску може опрацьовувати кожен окремо взяту навчаючу вибірку. В останньому випадку алгоритм називають *стохастичним градієнтом*. Ураховуючи розмірність навчаючої вибірки (яка може сягати понад десятків тисяч навчаючих зразків), перевагу слід віддавати стохастичному градієнту. Ще однією перевагою стохастичного градієнта є його менша схильність до попадання в локальний мінімум у порівнянні з АЛМ.

Наведені залежності (14.1 – 14.4) дозволяють легко обчислювати похідну похибки за вагами шару на виході. Обчислити похибку у схованих шарах дозволяє широко розповсюджений в нейротехнології метод зворотного поширення похибки (див. *розд. 4.5*).

Висновки з процедури навчання ЗНС. Узагальнення процедури навчання ЗНС полягає в наступному:

- *по-перше*, багат шаровий перцептрон керованого розміру здатний навчитися складному багатовимірному і нелінійному відображенню за допомогою обмеження його архітектури шляхом залучення до неї апріорних знань щодо поставленої задачі.
- *по-друге*, синаптичні ваги і рівні порогів можуть бути навчені за допомогою циклової роботи простого алгоритму зворотного поширення на множині прикладів навчання.

14.5. Переваги і недоліки згортаючих нейросіток

До переваг згортаючих нейронних сіток необхідно віднести наступне:

- один з кращих алгоритмів щодо розпізнавання і класифікації зображень;
- у порівнянні з повнозв'язною НС (типу перцептрона) – набагато менша кількість налагоджуваних ваг, бо одне ядро ваг використовується повністю для усього зображення, замість того, щоб робити для кожного пікселя вхідного зображення свої персональні вагові коефіцієнти. Це “підштовхує” НС при навчанні до узагальнення демонстрованої інформації, а не попиксельному запам'ятовуванню кожної показаної картинки в міриадах вагових коефіцієнтів, як це робить перцептрон;
- зручне розпаралелювання обчислень, і, відтак, можливість реалізації алгоритмів роботи і навчання НС на графічних процесорах;
- відносна сталість до повороту та зсуву розпізнаваних зображень;
- навчання класичним методом зворотного поширення похибки. Серед недоліків відзначимо наступні:
- архітектура згортаючої НС в більшій мірі придатна тільки для розпізнавання зображень;
- занадто багато варіюваних параметрів НС, і через це залишається незрозумілим, які необхідні налагодження для певних задач та обчислювальних потужностей. Так, до варіюваних параметрів можна віднести: кількість шарів, розмірність ядра згортки для кожного з шарів, кількість ядер для кожного з шарів, крок зсуву ядра при опрацюванні шару, необхідність у

шарах субдискретизації, ступінь зменшення ними розмірності, функція щодо зменшення розмірності (вибір максимуму, середнього тощо), передаткова функція нейронів, наявність та параметри повнозв'язної НС на виході згортаючої. Всі ці параметри суттєво впливають на результат, проте вибираються досліджувачами емпірично;

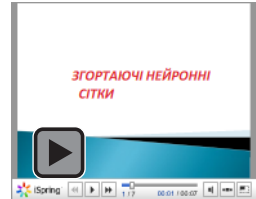
- існує декілька перевірених і добре функціонуючих конфігурацій сіток, проте немає правил, за якими необхідно робити вибір для нової задачі.

Контрольні завдання

1. Навести область застосування згортаючих нейронних сіток і обґрунтувати доцільність їх створення.
2. Розкрити ідею створення надточних нейронних сіток через чергування згортаючих та субдискретизуючих шарів та наявність повнозв'язних шарів на виході.
3. Розкрити концепції утворення трьох парадигм згортаючих нейронних сіток: локального сприйняття, подільності ваг і субдискретизації.
4. Прокоментувати порядок організації згортаючої нейронної сітки.
5. Дати нейроподібне обґрунтування структури згортаючої нейросітки.
6. Навести архітектуру згортаючої нейронної сітки і розтлумачити функціональну належність схованих шарів в сітці.
7. Охарактеризувати процедуру навчання згортаючої нейронної сітки.
8. Розкрити оптимізаційні особливості алгоритму Левенберга-Марквардта при навчанні згортаючих нейросіток.
9. Проаналізувати недоліки і переваги згортаючих нейронних сіток.

Презентації

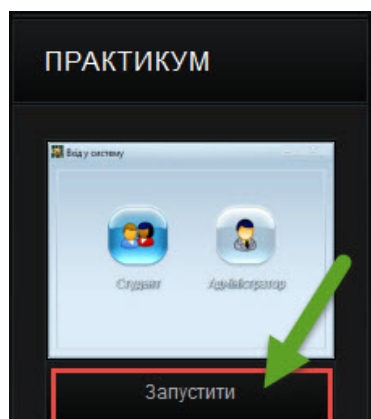
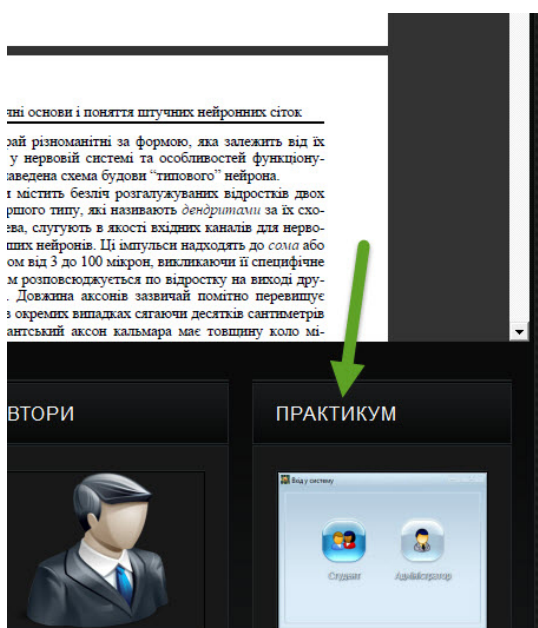
1. Згортаючі нейронні сітки



Практичні заняття

Згортаючи нейронні сітки

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 15

Генетичні алгоритми

Так связан, съединён от века
Союзом кровного родства
Разумный гений человека –
С творящей силой естества ...
Ф. И. Тютчев

Де початок того кінця, яким закінчується початок
Афоризм

Зустрічаються двоє крутих українців. Один до одного:
- Вован, я чув, ти вчишся грати на гітарі?
- Так, але вже кинув.
- А чому?
- Гітара має 6 струн, а в мене тільки 5 пальців.
- О це ж які мутанти грають на фортепіано?!
Анекдот

15.1. Механізм генетичної спадковості

Відповідно до *еволюційної теорії* кожний біологічний вид цілеспрямовано розвивається і змінюється для того, щоб найкращим чином пристосуватися до навколишнього середовища. *Еволюція* в цьому сенсі уявляє собою процес оптимізації усіх живих організмів. Природа розв'язує цю задачу шляхом *природного відбору*, суть якого полягає в тому, що більш пристосовані особі мають більше можливостей для виживання і розмножування і, як наслідок, приносять більше потомства, аніж погано пристосовуванні особі.

Означення 15.1. *Еволюція* – процес розвитку, який складається з поступових якісних змін без стрибків, тобто *процес неперервної оптимізації біологічних видів*.

Означення 15.2. *Природний відбір* – процес, за допомогою якого природа відбирає ті хромосоми, які кодують кращі (в сенсі деякого критерію) характеристики, з метою відтворення цих характеристик у нащадків.

При цьому при передаванні генетичної інформації (*генетичного успадкування*) нащадки успадковують від батьків основні їх якості. Отже, нащадки сильних особень також будуть відносно добре пристосованими, а їх частка у загальній масі особень буде зростати. Після зміни декількох десятків або сотень поколінь середня пристосова-

ність особень даного виду помітно зросте. І хоча механізми, стимулюючи процес розвитку, до кінця ще не зрозумілі, можливості еволюції вивчені досить повно.

Дамо коротеньку довідку про те, як улаштовані механізми генетичної спадковості. У кожній клітині живого організму міститься уся генетична інформація даної особи. Ця інформація записана у вигляді набору молекул ДНК (наприклад, людська ДНК містить близько 80000 генів), кожна з яких уявляє собою ланцюжок з нуклеотидів чотирьох типів – аденін, тимін, цитозин, гуанін, відповідно позначуваних *A*, *T*, *C* і *G*. Власне інформацію несе порядок прямування нуклеотидів у ДНК. Таким чином, генетичний код особи – це довгий рядок в ланцюжку символів, представлених в дво-, три- або чотирилітерному алфавіті. У тваринній клітині кожна молекула ДНК оточена оболонкою, і таке утворення називається *хромосомою*.

Означення 15.3. *Хромосоми* – стандартні елементи ядра клітини, які містять дезоксирабінуклеїнову кислоту (ДНК), у якій зберігається спадкова інформація організму. Сполучення трьох поруч розташованих нуклеотидів у ланцюгу ДНК утворюють *генетичний код*.

Означення 15.4. *Генетичний код* – властива живим організмам єдина система “запису” спадкової інформації у молекулах нуклеїнових кислот у вигляді послідовності нуклеотидів.

Кожна уроджена якість особі кодується певною частиною хромосоми, яка називається *геном* цієї властивості.

Означення 15.5. *Ген* – одиниця спадкового матеріалу, відповідальна за формування якої-небудь елементарної ознаки.

Унікальність властивостей генів, які знаходяться у складній взаємодії з іншими генами, полягає у поєднанні високої сталості (незмінності в ряді поколінь) із спроможністю до наступних змін – *мутаціям*. У вищих організмів ген входить у склад до хромосом. Відмінні значення гена називаються його *алелями*.

Означення 15.6. *Геном* – сукупність генів, яка міститься у гаплоїдному (одинарному) наборі хромосом даного організму.

Означення 15.7. *Алелі* — відмінні форми одного й того ж гена, розташовані в однакових ділянках гомологічних хромосом і визначаючи альтернативні варіанти розвитку однієї і тієї ж ознаки.

При розмноженні особень відбувається злиття двох батьківських статевих клітин, і їх ДНК взаємодіють, утворюючи ДНК нащадка. Основний спосіб взаємодії – *кросовер* або *кросинговер* (*схрещення*). При кросовері ДНК предків діляться на дві частини, а потім обмінюються своїми половинками.

Означення 15.8. *Кросинговер* (в літературі з генетичних алгоритмів також вживається термін *кросовер* або *схрещення*) – складний процес обміну блоками генів між хромосомами, в результаті чого народжується одна або декілька нових хромосом.

Хромосома розділяється на ділянки у визначених точках, одних і тих же для одного виду, що може бути визначенням виду на генетичному рівні; місце розташування цих точок задається єдиним геном.

При наслідуванні можливі *мутації*, в результаті яких можуть змінитися деякі гени в статевих клітинах одного з батьків.

Означення 15.9. *Мутація* – стійке (тобто таке, яке може бути успадковане нащадками даної клітини або організму) перетворення *генотипу*, яке відбувається випадково під впливом зовнішнього або внутрішнього середовища. Найбільш розповсюджений вид мутацій – випадкове змінення тільки одного з генів хромосоми.

Вважається, що саме мутації є причиною появи нових біологічних видів, а кросинговер визначає вже мінливість всередині виду (наприклад, генетичні розбіжності між людьми).

Означення 15.10. *Генотип* – сукупність усіх генів організму, яка складає його генетичну конструкцію.

Змінені гени передаються нащадку і надають йому нові властивості. Якщо ці нові властивості корисні, вони, скоріше за все, збережуться у даному виді. При цьому відбудеться стрибкоподібне зростання пристосованості виду.

15.2. Генетичний алгоритм

Реалізація наближеної оптимізації. Перші роботи з проблем симуляції еволюції були проведені у 1954 році Нільсом Баричеллі на комп'ютері, встановленому в Інституті Просунутих Досліджень Принстонського університету [84]. У 1966 р. Л.Дж. Фогель, А.Дж. Оуенс, М.Дж. Уолш запропонували і дослідили еволюцію простих автоматів, завбачаючих символи у цифрових послідовностях [62]. Правильно стверджувати, що *генетичні алгоритми* (ГА) вперше були запропоновані та досліджені відомим українським вченим академіком НАН України О. Г. Івахненко [24-26] у шестидесятих роках минулого століття і засновані на використанні розробленого ним експериментального методу самоорганізації моделей – *методу групового урахування аргументів* (МГУА) [24]. Приблизно десятьма роками пізніше Дж. Холландом в роботі [142] було сформоване форма-

льне подання процесів, якими супроводжується процедури кросоверу або кросинговеру з подальшою мутацією і наближеною оптимізацією. Проте практичне застосування ГА розпочалося майже два десятиріччя по тому. Ці роботи склали основу головних напрямків розробки еволюційних алгоритмів. Простий ГА на основі праць Дж. Холланда був вперше описаний Д. Гольдбергом [198]. Первісна мета полягала у побудуванні методів оптимізації з використанням механізмів природного відбору і генетики, зокрема, елементів випадковості і паралелізму, що супроводжується порівняно простими обчисленнями. При цьому приймалося за основу існування деякої первісної популяції, яка володіє деякою генетичною структурою шляхом визначених генетичних операцій передачі спадковості (селекції, розмноження і мутації) і утворює гілки, декотрі з яких в процесі боротьби за виживання (виживає найпотужніший) укріплялися і утворювали генетичну структуру наступної популяції.

Означення 15.11. *Генетичний алгоритм* – це евристичний алгоритм пошуку, який використовується для розв'язання задач оптимізації і моделювання шляхом послідовності керуючих дій і операцій відтворення еволюційних процесів на основі аналогів механізмів генетичної спадковості та природного відбору.

Взагалі ГА уявляють собою скоріше підхід, аніж єдині алгоритми. Вони вимагають змістовного наповнення для розв'язання кожної конкретної задачі. Необхідно також зазначити, що *ГА реалізує наближену оптимізацію*. Він не гарантує відшукування найкращого розв'язку як *цільову функцію*, його функціонування виявляється у наближенні до останнього.

Означення 15.12. *Цільова функція в ШНС* – це деяка опукла функція похибки, мінімізовувана в процесі навчання на всьому наборі навчання.

В ГА як цільова функція розглядається “*придатність*” або “*пристосованість*”, яка кількісно визначає якість хромосоми, тобто дозволяє формувати висновки щодо порівняння даної хромосоми з іншими. Пристосованість характеризується функцією пристосованості, яка зазвичай має додатні значення і вибирається в залежності від конкретної задачі. Максимальне значення цієї функції і відповідає оптимальному розв'язку, отримуваному за допомогою ГА, і підтверджує той факт, що дана хромосома має більшу ймовірність її вибору для створення наступного покоління. Цільові функції пристосовано-

сті можуть мати велику розмірність, бути недиференційованими, багатоекстремальними і мати розриви.

Як приклад, нехай задана деяка складна цільова функція, яка залежить від декількох змінних, і необхідно розв'язати задачу оптимізації, тобто знайти такі значення змінних, за яких значення функції мінімальне або максимальне.

Цю задачу можна розв'язати, застосовуючи розглянуті вище біологічні еволюційні підходи до оптимізації. Будемо як і раніше розглядати кожний варіант (набір значень змінних) як особь, а значення цільової функції для цього варіанта – як пристосованість даної особі. Тоді в процесі еволюції пристосованість особин буде зростати, отже, будуть з'являтися все більш оптимальні варіанти. Зупинивши еволюцію у деякий момент і вибравши кращий варіант, можна отримати прийнятний розв'язок задачі.

На рис. 15.1 наведений один з варіантів структури ГА. Спочатку генерується випадкова популяція – декілька особин з випадковим набором хромосом (числових векторів). Генетичний алгоритм імітує еволюцію цієї популяції як циклічний процес схрещення особин, мутації і зміни поколінь (відбору).

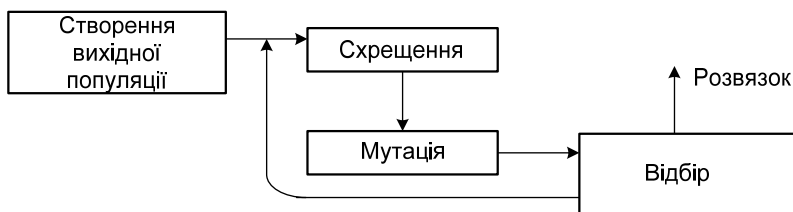


Рис. 15.1. Варіант структури генетичного алгоритму

Схема роботи ГА наведена на рисунках 15.2 і 15.3. Оскільки ГА уявляє собою штучну еволюцію, яка містить визначені етапи (рис. 15.2), для кожного з яких існують свої *генетичні оператори* (ГО).

При цьому хромосоми вихідної популяції в процесі еволюції замінюються іншими, більш пристосованими хромосомами (рис. 15.3). Процес відбувається за наступною схемою.

За період життєвого циклу популяції в результаті декількох випадкових схрещувань (завдяки кросовера) і мутацій до неї додається якась кількість нових варіантів. Далі відбувається відбір, в результаті якого зі старої популяції формується нова, після чого стара популяція відмирає. Після відбору до нової популяції знов застосовуються операції кросовера і мутації, далі знов відбувається відбір, і так далі.

Відбір в ГА тісно пов'язаний з принципами природного відбору наступним чином:

- пристосованість особі відповідає значенню цільової функції на заданому варіанті;
- виживання найбільш пристосованих особин відповідає тому, що популяція наступного покоління варіантів формується з урахуванням цільової функції: чим пристосованіша особина, тим більша ймовірність її участі у кросовері, тобто розмноженні.

Таким чином, модель відбору визначає, як необхідно будувати популяцію наступного покоління. Як правило, ймовірність участі особі у схрещенні береться пропорційною її пристосованості.

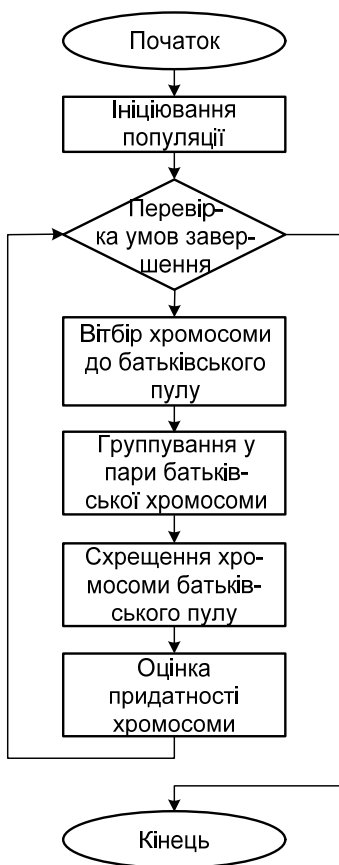


Рис. 15.2. Блок-схема ГА

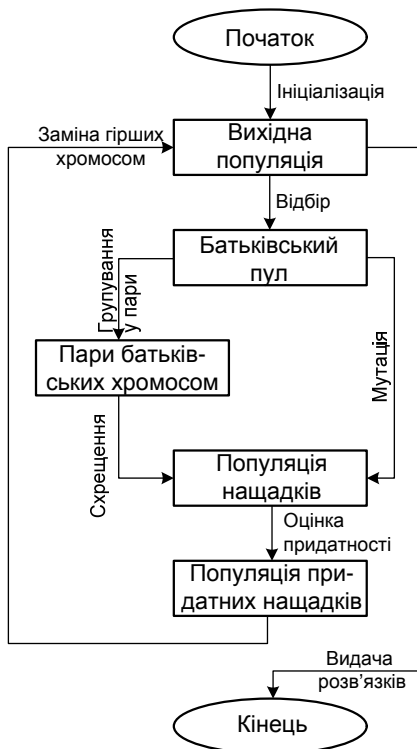


Рис. 15.3. Схема еволюції розв'язків в ГА

Схема функціонування генетичного алгоритму. Тепер, з урахуванням наведеної вище термінології, перейдемо до опису функціонування ГА за його класичним варіантом:

1. Ініціювати вихідний момент часу $t = 0$ (іноді ще називають епохою). За випадковою стратегією сформувати *вихідну популяцію* з k особин: $H_b = \{L_1, L_2, \dots, L_k\}$.

2. Обчислити *пристосованість* $\Pi_{L_i} = \text{fit}(L_i)$, $i = 1 \dots k$ кожної особі і популяції $\Pi_t = \text{fit}(H_t)$ у цілому (іноді використовується термін *фітнес*). Значення цієї функції визначає придатність особі, описаної хромосомою, для розв'язання задачі.

3. Вибрати особь L_c з популяції $L_c = \text{Get}(H_t)$.

4. З визначеною ймовірністю (ймовірність кросовера P_{cr}) вибрати другу особь з популяції $L_{c1} = \text{Get}(H_t)$ і виконати *оператор кросовера* $L_{cr} = \text{Crossing}(L_c, L_{c1})$.

5. З визначеною ймовірністю (ймовірність мутації P_m) виконати *оператор мутації* $L_c = \text{mutation}(L_c)$.

6. З визначеною ймовірністю (ймовірність інверсії P_{inv}) виконати *оператор інверсії* $L_c = \text{inversion}(L_c)$.

7. Помістити $\text{insert}(H_{t+1}, L_c)$. отриману хромосому у нову популяцію

8. Виконати операції, починаючи з кроку 3, k разів.

9. Збільшити номер поточної епохи $t = t + 1$.

10. Якщо виконався *критерій* (умова) *зупинення*, то завершити роботу, інакше перехід на крок 2.

Критерій зупинення функціонування ГА:

- відшукання глобального або субоптимального розв'язку;
- вихід на "плато";
- вичерпання кількості поколінь, відпущених на еволюцію;
- вичерпання часу, відпущеного на еволюцію;
- вичерпання заданої кількості звертань до цільової функції.

Найбільшу роль в успішному функціонуванні алгоритму відіграє етап відбору батьківських хромосом на кроках 3 і 4. При цьому можливі різні варіанти. Найчастіше використовується метод відбору, названий *рулеткою*. При використанні такого методу ймовірність вибору хромосоми визначається її пристосованістю, тобто $P_{\text{Get}(L_i)} : \text{Fit}(L_i) / \text{Fit}(H_t)$. Використання цього методу приводить до

зростання ймовірності передавання ознак більш пристосованими особинами нащадкам.

Інший часто використовуваний метод – *турнірний відбір*. Він полягає в тому, що випадково вибирається декілька особин з популяції (зазвичай дві) і переможницею вважається особь з найбільшою пристосованістю.

Часто застосовується так звана *стратегія елітизму*, за якою декілька кращих особин переходять до наступного покоління без змін, не приймаючи участі у кросовері і відборі. У будь якому разі кожне наступне покоління буде в середньому кращим за попереднє. Коли пристосованість особин перестає помітно збільшуватися, процес зупиняють, і в якості розв'язку задачі оптимізації беруть найкращий з знайдених варіантів.

Означення 15.13. *Елітизм* – алгоритм математичної моделі, яка описує розвиток популяцій, за яким найбільш пристосовані особи дають приплід з більшою ймовірністю.

Використання елітизму зазвичай дозволяє прискорити збіжність ГА, проте недоліком використання стратегії елітизму є підвищення ймовірності попадання алгоритму у локальний мінімум.

Кодування генетичної інформації. Як вже вище наголошувалося, природний хромосомний матеріал уявляє собою лінійну послідовність всіляких комбінацій чотирьох нуклеотидів у ДНК, і тому в ГА використовується запис інформації у вигляді послідовностей, які складаються з ланцюжка символів, поданих у дво-, три- або чотирилітерному алфавіті. Враховуючи на те, що більш наочним і доступним для розуміння є двійкове кодування, подання хромосомної інформації у вигляді бітових рядків (нулів та одиниць) вважається найбільш уживаним. Проте суттєвим недоліком звичайного двійкового коду є його непостійна *віддаль Хеммінга* між поданням двох крокуючих одна за одною цифр.

Означення 15.14. *Віддаль Хеммінга* – кількість позицій, в яких відповідні символи двох слів однакової довжини різні. У більш загальному випадку віддаль Хеммінга застосовується для рядків однакової довжини будь-яких q -їчних алфавітів і слугує метрикою відмінності (функцією, яка визначає віддаль у метричному просторі) об'єктів однакової розмірності.

В цьому сенсі ефективнішим є *двійковий код Грея* [48]. Як впливає з таблиці 15.1, в коді Грея віддаль Хеммінга завжди дорівнює одиниці, тобто як у простому двійковому коді вона може дорівнювати чотирьом (дійсно, при зміні десятичного числа 7 на 8 всі чотири біта двійкового коду змінюються). Більша віддаль Хеммінга

веде до зниження стійкості ГА через те, що при малих віддальх Хеммінга зміна одиночного біта через мутацію викличе меншу наступну зміну.

Як приклад можна розглянути три поруч розташовані рядка з табл. 15.1, які кодують десяткові цифри 10, 11 і 12. Нехай кодуючі число 3 фрагменти хромосоми належать до деякого оптимального вектора. Якщо краща особь з поточної популяції містить фрагмент хромосоми, якою кодується число 2, то для отримання вірного розв'язку досить замінити біти у молодшому розряді. Якщо ж краща особь містить фрагмент хромосоми, якою кодується число 4, то для отримання розв'язку необхідно проінвертувати біти вже у трьох розрядах. Аналогічно можна дослідити і ситуацію зі змінами у кодуючих числах 10, 11 і 12. В обох випадках для коду Грея обидві ситуації аналогічні і зводяться до корекції тільки одного біта.

Отже, використання коду Грея сприяє суттєвому покращенню процесу пошуку екстремуму деяких функцій і забезпечує отримання розв'язків в усіх випадках не гірше, аніж двійкове кодування.

Таблиця 15.1

Залежність віддалі Хеммінга від виду кодування

Десяткове число	Двійковий код	Віддаль Хеммінга	Код Грея	Віддаль Хеммінга
0	0000		0000	
1	0001	1	0001	1
2	0010	2	0011	1
3	0011	1	0010	1
4	0100	3	0110	1
5	0101	1	0111	1
6	0110	2	0101	1
7	0111	1	0100	1
8	1000	4	1100	1
9	1001	1	1101	1
10	1010	2	1111	1
11	1011	1	1110	1
12	1100	3	1010	1
13	1101	1	1011	1
14	1110	2	1001	1
15	1111	1	1000	1
16	11111	5	11000	1

Математична інтерпретація кодування генетичної інформації. Математично викладене можна подати наступним чином. Нехай є векторна цільова функція від багатьох змінних

$$f(x_1, x_2, \dots, x_n),$$

у якої треба знайти глобальний максимум або мінімум.

Уявимо незалежні змінні $x_i, i = \overline{1, n}$ у вигляді хромосом. Для цього виконаємо кодування незалежних змінних або у двійковому форматі, або у форматі з рухомою комою.

В разі двійкового кодування використовується n біт для кожного параметра, причому, n може бути різним. Якщо параметр може змінюватися між мінімальним (\min) і максимальним (\max) значеннями, можна скористатися наступними залежностями для перетворення:

$$w = \frac{d(\max - \min)}{2^n - 1} + \min, \quad d = \frac{w - \min}{(\max - \min)(2^n - 1)},$$

де d – значення параметра у двійковому форматі; w – значення параметра у форматі з рухомою комою.

Хромосоми у форматі з рухомою комою задаються шляхом послідовного розміщення закодованих параметрів один за одним.

Як вже було доведено (табл. 15.1), найбільш прийнятні результати дає варіант представлення хромосом у двійковому форматі з використанням коду Грея. Проте в цьому випадку необхідно постійно здійснювати кодування/декодування параметрів генів.

Генетичні оператори. Розглянемо роботу ГА детальніше. Заздалегідь підбирається деяке уявлення для розглядуваного розв'язку, розмір і структура популяції. У першому поколінні випадковим чином генерується популяція хромосом. Зазвичай, розмір популяції постійний. Визначається “корисність” хромосом. Після цього ГА може розпочинати генерування нової популяції.

Далі еволюція відбувається через процес репродукування (відтворення), який включає *кросовер* (схрещення), *мутацію* та *інверсію* (рекомбінацію), які реалізуються при моделюванні генетичних структур відповідно *операторами кросовера, мутації та інверсії*, порядок застосування яких не є важливим. При цьому зберігається наведена вище біологічна термінологія в спрощеному вигляді, але прив'язана до математичного моделювання:

Означення 15.15. *Хромосома* – вектор (послідовність) з нулів та одиниць, кожна позиція (біт) якого уявляється *геном*.

Означення 15.16. *Кросовер* – оператор, формуючий хромосому нащадка, використовуючи для цього фрагменти батьківських

хромосом. Іншими словами, це операція, за якою дві хромосоми обмінюються своїми частками.

Означення 15.17. *Мутація* – процес випадкової зміни однієї або декількох позицій (біт) в хромосомі.

З трьох генетичних операторів **кросовер** є найважливішим. Кожна соматична клітина людини містить 46 хромосом, кількість яких є постійною в усіх наступних поколіннях. Існує регуляторний механізм – *мейоз*, який дозволяє зберігати постійність об'єму генетичної інформації по поколіннях організмів і складається з двох клітинних поділів при одному синтезі ДНК, яка безпосередньо приймає участь у шерензі генетичної інформації.

Складний процес хромосомного обміну блоками генів і визначає сутність кросовера. Існує декілька варіантів кросовера. Найбільш простим є *одноточковий*, за яким дві хромосоми “перерізаються” у випадково вибраній точці, і хромосома нащадка отримується з початку однієї і кінця іншої батьківських хромосом:

<i>Батьківські хромосоми</i>	<i>Нові хромосоми</i>	<i>Вибір</i>
Генотип1 100101110010	11000	
	→ 100101110010	11100
Генотип2 101101101000	11100	→ 101101101000
		11000

Якщо одна з нових хромосом отримала 60% характеристик від першого батька і 40% від другого, то друга нова хромосома отримає 40% характеристик від першого батька і 60% від другого. Саме завдяки операції, реалізовуваній кросовером особі популяції обмінюються між собою генетичною інформацією, різко збільшуючи тим самим комбінаторну спадкову змінність організмів. Обидві результуючі хромосоми входять до *пул-генів* (див. *рис.* 15.3), де вони замінюють “погані” або “гарні” хромосоми або відкидаються. Але якщо хромосома відкидається, то унікальні її можливості (як “погані”, так і “гарні”) втрачаються назавжди.

Тільки шляхом **мутації** така хромосома може бути створена знову. Мутація уявляє собою випадкову зміну хромосоми (зазвичай просту зміну стану одного з бітів на протилежний). Даний оператор дозволяє, по-перше, більш швидко знаходити локальні екстремуми і, по-друге, “перескочити” в інший локальний мінімум:

Хромосома до мутації	Після мутації
00110010111001 0 11000	→ 00110010111001 1 1000.

↑
місце прикладання оператора мутації

В залежності від того, в якому компоненті хромосоми відбулася зміна, визначається величина віддалення нащадка від батька.

Мутація викликає різку зміну характеристик хромосоми і впливає на всі її наступні покоління, які містять цей видозмінений компонент. Якщо мутація приводить до зниження якості хромосоми, то дана хромосома відкидається і усувається з пулу-генів.

Інверсія змінює порядок бітів у хромосомі шляхом циклічного переустановлення (випадкову кількість разів).

Хромосоми до інверсії Після інверсії

0011001011100101**11000** → **11000**001100101110010 .

↑
місце прикладання оператора інверсії

Для багатьох модифікацій ГА даний оператор не є обов'язковим.

15.3. Генетичні алгоритми і штучні нейросітки

Отже, ГА прийнято розглядати як алгоритми оптимізації з відшукуванням розв'язків на основі ідеї адаптації, тобто визначення елементів популяції, які надають екстремум вибраному функціоналу (функції пристосованості). Можливість знаходження за допомогою ГА глобального екстремуму робить ці алгоритми особливо привабливими.

В свою чергу, алгоритми навчання ШНС (див. розд. 4), підналагоджуючи синаптичні ваги, мінімізують кількість невірних реакцій ШНС, тобто також розв'язуючи деяку оптимізаційну задачу – як ГА, так і ШНС, копіюючи процеси живої природи, прагнуть забезпечити виживаність досліджуваного об'єкта. Схожість в розв'язуванні задач дозволяє об'єднувати обидва підходи, наприклад, шляхом конструювання ШНС за допомогою ГА.

Найвимоганішими областями застосування ГА в ШНС є:

- пошук оптимальних значень синаптичних ваг в задачах навчання ШНС, в тому числі і нечітких;
- визначення оптимальної структури ШНС для розв'язання конкретних прикладних задач.

Навчання нечітких ШНС на основі ГА. Перевага ГА перед іншими методами оптимізації, особливо градієнтними, які використовуються для налагоджування вагових коефіцієнтів ШНС, полягає в наступному:

- ГА здатні досягати глобального екстремуму, не застрягаючи у локальних;
- використовуваний функціонал може бути недиференційованим;

- реалізація ГА можлива з використанням паралельних обчислень, забезпечуючих можливість пошуку розв'язку одночасно на всьому просторі параметрів.

Зазвичай ГА кодують параметри ШНС рядком (бітовим) або списком його реквізитів. Хоча для ефективної роботи ГА необхідна велика кількість рядків (хромосом) або списків з великою кількістю наборів параметрів даної сітки, реалізація алгоритму не викликає утруднень через використання паралельних обчислень. Закодовані рядки об'єднуються із залученням операторів схрещення (кросовера) і мутації для формування наступної популяції, задовольняючої вибрану функцію пристосованості, яка часто вибирається у вигляді промасштабованої інверсії сіткової похибки.

Використання ГА в ШНС для визначення вагових коефіцієнтів забезпечує більш швидке попадання в окіл оптимального розв'язку, аніж пошук на основі АЗПП, проте у самому околі оптимуму властивості алгоритмів змінюються на протилежні. Це пояснюється тим, що збіжність ГА у точку оптимуму визначається оператором мутації.

В роботі [203] ГА використовувалися для керування алгоритмом зворотного поширення у нейросіток невеликої розмірності, які в свою чергу були модулями ШНС великої розмірності. Такий підхід забезпечив більш високу швидкість навчання результуючої ШНС у порівнянні з навчанням вихідної сітки.

Загальними етапами такого навчання є наступні:

Крок 1. Виокремлення керуючих параметрів задачі навчання.

Крок 2. Отримання розв'язку при фіксованих значеннях параметрів.

Крок 3. Визначення розузгодженості отриманого і необхідного розв'язків.

Крок 4. Вибір нових значень параметрів на основі роботи генетичного алгоритму.

Крок 5. Зупинення у випадку отримання задовольняючої розузгодженості розв'язку, інакше – перехід до кроку 2.

В якості керуючих параметрів навчання ШНС, впливаючих на якість розв'язання, можуть вибиратися параметри функцій приналежності, а також різна формалізація логічних правил.

Оптимізація топології ШНС. Вибір топології ШНС на сьогодні є емпіричним і базується на досвіді і кваліфікації проектувальника (або користувача).

Формулювання задачі оптимальної топології ШНС для розв'язування конкретного класу прикладних проблем як задачі дискретної оптимізації уможливорює і робить доцільним застосування ГА для її розв'язання через велику апіорну невизначеність щодо топології ШНС, якій повинна відповідати велика і первісно різноманітна популяція, відбиваюча відповідно велику розмірність і різномірність вихідного простору.

Використовуючи двійкове кодування хромосом (бітові рядки), С. Харп і Т. Самад [129] розширили поняття хромосом і бітових рядків до поняття об'єкта деревуваної структури, кожний ген якої характеризується адресою, кількістю нейронів, ваговими параметрами тощо. Для управління ГА і набором з'єднаних нейронів використовувався схема “синьки” (*blueprint*), в якій сіткові характеристики визначені як структури даних. *Blueprint* визначає вид реальної ШНС, а остання навчається за алгоритмом, параметри якого уточнюються у *blueprint*. Навчання ШНС потім тестується для оцінки її стійкості при виключенні з неї деяких модулів або при зміні значень отримуваних вагових коефіцієнтів. Далі визначається її пристосованість, задана довільною функцією, наприклад, точністю і швидкістю навчання, кількістю вузлів, максимальною кількістю виходів з вузла тощо.

Після цього формується наступна популяція сітки у *blueprint* з використанням генетичних операторів схрещування, мутації та інверсії, яка полягає у створенні нового *blueprint* на основі двох батьківських *blueprint*. Отже, генетичний підхід тут використовується на двох рівнях: на макрорівні – до бітових рядків; на макрорівні – до *blueprint*. Такий підхід дозволяє розробнику (або користувачу) суттєво розширити область можливих варіантів розв'язань.

15.4. Особливості генетичних алгоритмів

Порівняємо стандартні підходи з генетичними алгоритмами на прикладі задачі комівояжера (*Travelling Salesman Problem*), суть якої полягає у відшуванні найкоротшого замкнутого шляху замкнутого обходу міст, заданих своїми координатами.

Щоб використати ГА для розв'язання практичних задач, необхідно розглядати більш складні варіанти введених вище понять. Пояснимо це на прикладі задачі комівояжера для 30 міст. В якості індивідуумів будемо розглядати маршрути обходу. Інформацію щодо маршруту можна записати у вигляді одної хромосоми – вектора довжиною 30, де в першій позиції стоїть номер першого міста на шляху

прямування, потім – номер другого міста і т. д. Перше ускладнення виникає, коли спробувати визначити мутації для таких хромосом – стандартна операція, змінююча тільки одну позицію вектора, неприпустима, бо приводить до некоректного маршруту. Але можна визначити мутацію як переставлення значень двох випадково вибраних генів. При такому перетворенні шлях прямування змінюється тільки в двох містах. За умовою задачі в розглядуваних хромосомах кожний ген (номер міста) повинен зустрічатися тільки один раз. Такий різновид хромосом має назву “*перелічувані хромосоми з унікальними генами*” і часто використовується в комбінаторних задачах. Стандартна операція схрещування для цього типу хромосом знов же некоректна, тому використовується більш складна схема двоточкового схрещування, яка викладається в спеціальній літературі.

Отже, для 30 міст пошук оптимального шляху уявляє собою складну задачу, сприяючи розвитку нових методів розв’язання задачі (в тому числі НС і ГА).

Кожний варіант розв’язку (для 30 міст) – це числовий рядок, де на j -ій позиції стоїть номер i -го за порядком обходу міста. Таким чином, у цій задачі 30 параметрів, причому, не всі комбінації значень припустимі.

Генетичні алгоритми – не єдиний спосіб розв’язання задач оптимізації. Крім нього існують ще принаймні два основних підходи для розв’язання подібних задач – *перебірний* і *локально-градієнтний*, кожний з яких має свої переваги і недоліки.

Перебірний метод найпростіший у програмуванні. Для пошуку оптимального розв’язання (максимуму цільової функції) необхідно послідовно обчислити значення цільової функції в усіх можливих точках, запам’ятовуючи максимальне з них. Недоліком методу є висока обчислювальна складність: треба обрахувати довжини більш 10^{30} варіантів шляхів, що абсолютно нереально. Проте, якщо перебирання усіх варіантів за припустимий час можливе, то знайдений розв’язок є оптимальним.

Локально-градієнтний метод оснований на парадигмі градієнтного спускання. Ідея цього методу полягає в тому, що оптимальний розв’язок отримується ітераціями. Береться випадкова вихідна точка, а далі у циклі відбувається зсув цієї точки на малий крок, причому крок робиться у тому напрямку, в якому цільова функція зростає швидше за все. При досягненні локального максимуму такий метод зупиняється, тому для пошуку глобального оптимуму необхідні

додаткові заходи. Отже у випадку, якщо цільова функція досить гладка і має тільки один локальний максимум (*унімодальна*), то оптимальний розв'язок можна отримати методом градієнтного спускання.

Градієнтні методи працюють швидко, але не гарантують оптимальності отриманого розв'язку. Вони ідеальні для використання у так званих *унімодальних* задачах, де цільова функція має єдиний локальний максимум (він же – глобальний). Проте задача комівояжера не є такою.

Практичні задачі, як правило, *мультимодальні* і *багатовимірні* (містять багато параметрів). Для них не існує універсальних методів, дозволяючих досить швидко знайти абсолютно точні розв'язки. Комбінуючи перебірний і градієнтний методи, можна отримати наближені розв'язки, точність яких буде зростати зі зростанням часу розрахунку.

Генетичний алгоритм уявляє собою саме такий комбінований метод. Механізми схрещування і мутації в якомусь сенсі реалізують перебірну частину методу, а відбирання кращих розв'язків – градієнтне спускання. На рис. 15.4 показано, що таке сполучення забезпечує стійко задовільну ефективність генетичного пошуку для будь-яких типів оптимізаційних задач.

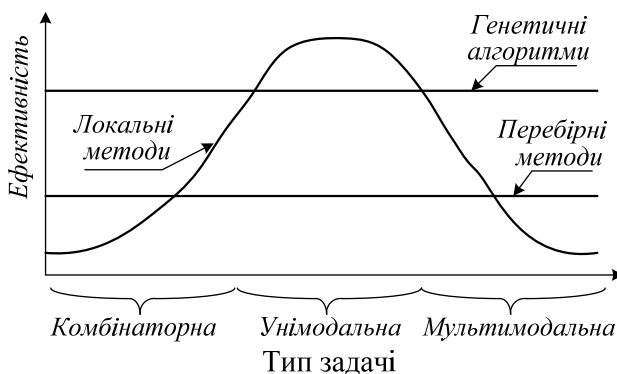


Рис. 15.4. Ефективність генетичних алгоритмів

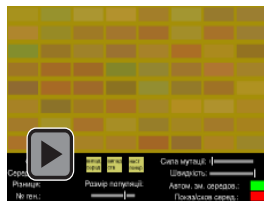
Таким чином, якщо на деякій множині задана складна функція від декількох змінних, то ГА за прийнятний час знаходить значення функції, досить наближене до оптимального. Задаючи час розрахунку, можна мати один з кращих розв'язків, які реально отримати за цей час.

Контрольні завдання

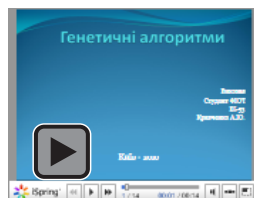
1. Навести і розтлумачити основні терміни еволюційної теорії: еволюція, природний відбір, хромосоми, ген, геном, генетичний код тощо.
2. Дати визначення генетичного алгоритму і прояснити, чому він реалізує наближену оптимізацію в штучних нейронних сітках.
3. Прокоментувати схему еволюції розв'язків в генетичному алгоритмі.
4. Навести спрощену структуру та прокоментувати процеси популяції блок-схемою генетичного алгоритму.
5. Розкрити схему функціонування генетичного алгоритму і обґрунтувати критерії зупинки функціонування.
6. Прокоментувати використання коду Грея як механізму кодування генетичної інформації (на відміну від віддалі Хеммінга).
7. Дати тлумачення стратегії елітизму в роботі генетичного алгоритму.
8. Дати математичну інтерпретацію кодування генетичної інформації.
9. Пояснити, в чому полягає збіжність та відмінність біологічної термінології (хромосоми, кросовера, мутації, інверсії), прив'язаної до математичного моделювання.
10. Розкрити роль, яку відіграють генетичні алгоритми в інтерпретації штучних нейронних сіток.
11. Порівняти стандартні підходи з генетичними алгоритмами на прикладі задачі комівояжера, суть якої полягає у відшуванні найкоротшого замкнутого шляху замкнутого обходу міст, заданих своїми координатами.

Презентації

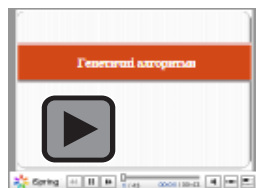
1. Генетичні алгоритми



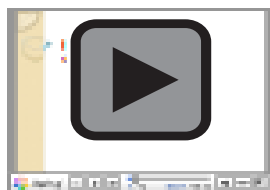
2. Генетичні алгоритми



3. Генетичні алгоритми



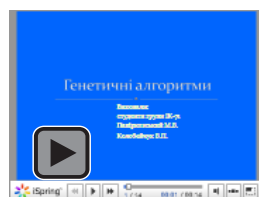
4. Генетичний алгоритм
Основні принципи.
Демонстрація.



5. Генетичні алгоритми



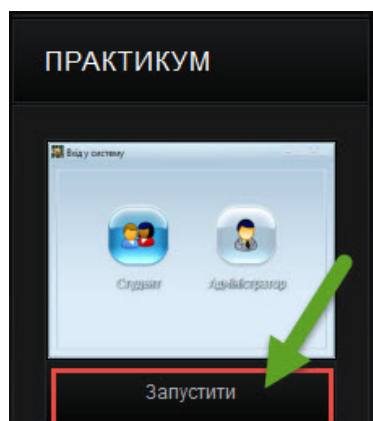
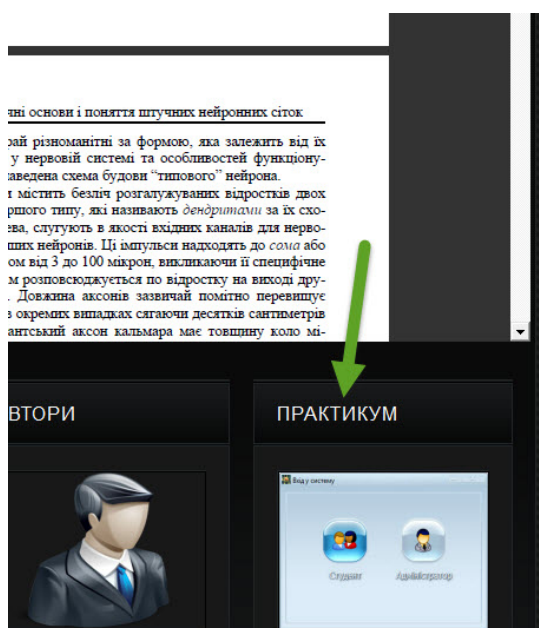
6. Генетичні алгоритми



Практичні заняття

Генетичні алгоритми

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 16

Метод групового урахування аргументів в нейромоделюванні

Все вірне – не нове, все нове – невірне
Афоризм

Нічого неможна сказати такого, що не було б висловлене раніше
Теренцій Публій

16.1. Метод групового урахування аргументів як предтеча ідеї нейросіткового еволюційного моделювання

За своїми принципами інваріантного відображення середовища багаторядні алгоритми *методу групового урахування аргументів* (МГУА) [24] надзвичайно близькі до ідей нейросіткового моделювання, зокрема, багат шаровому персептрону Ф. Розенблатта.

В математичній кібернетиці розрізняють два види ітеративних процесів розвитку систем [26]:

- *адаптація*, за якою екстремум (мета руху системи) залишається сталим;
- *еволюція*, за якою рух супроводжується зміною розташування екстремуму.

Якщо самоорганізація пов'язана тільки з *адаптаційними механізмами підстроювання* реакцій системи (наприклад, змінюванням значень вагових коефіцієнтів), то поняття еволюції пов'язане з *можливістю ефектора змінювати свою власну структуру*, тобто кількість елементів, спрямованість та інтенсивність зв'язків, настраюючи їх оптимальним чином відносно поставлених задач у кожний конкретний момент часу. В процесі еволюції в умовах складного та мінливого середовища ефектор здатний придбавати принципово нові якості, виходити на наступний ступінь розвитку. Наприклад, в процесі біологічної еволюції виникли надзвичайно складні і разом з тим напрочуд продуктивно функціонуючі живі організми.

Означення 16.1. *Еволюційне моделювання* – суттєво універсальний спосіб побудування прогнозів макростанів системи в умовах, коли повністю відсутня апостеріорна інформація, а апіорні дані задають лише передісторію цих станів.

Загальна схема алгоритму еволюції має наступний вид:

- задається початкова організація системи (в еволюційному

модельованні у цій якості може фігурувати, наприклад, скінчений детермінований автомат Мілі;

- проводять випадкові “мутації”, тобто змінюють випадковим чином поточний скінчений автомат;
- відбирають для подальшого “розвитку” ту організацію (той автомат), яка є “кращою” в сенсі деякого критерія, наприклад, максимальної точності передбачення послідовності значень макростанів досліджуваної системи.

Означення 16.2. *Автомат Мілі* – скінченний автомат, чиї символи на виході визначаються його станом та символами на вході (на відміну від автомату Мура, символи на виході якого визначаються тільки його станом). На ребрах в діаграмі станів позначають символи на вході та на виході (а в автоматі Мура символи на виході позначають на вершинах).

Критерій якості моделі в цьому випадку мало чим відрізняється, наприклад, від мінімуму *середньоквадратичної похибки* (СКП) на навчаючій послідовності *методу найменших квадратів* (МНК) (зі всіма впливаючими з цього недоліками). Проте, на відміну від адаптації, в еволюційному програмуванні структура розв’язувача мало змінюється при переході від однієї мутації до іншої, тобто відбувається перерозподілення ймовірностей, які б закріплювали мутації, що привели до успіху на попередньому кроці. Пошук оптимальної структури відбувається в більшій мірі випадковим і нецілеспрямованим, що зволікає процес пошуку, але забезпечує найкращу пристосовуваність до конкретних змінюваним умовам.

В останні 30-35 років спостерігається підвищена зацікавленість до найбільш “біологізованих” моделей еволюції з використанням генетичного алгоритму, який можна вважати “інтелектуальною” формою “методу спроб і похибок”. *Генетичний алгоритм*, запозичений у природних аналогів, є найбільш елегантним представником еволюційних методів і уявляє собою потужний пошуковий засіб, ефективний в усіляких проблемних областях і заснований на трьох компонентах:

- *генетичний пам’яті*, сконцентрованої у “хромосомах”;
- відтворення, здійснюваного за допомогою операторів *кросинговера* і *мутації*;
- *селекції* продуктивних розв’язків методами оптимізації багатокритеріальних функцій.

Нарешті, оскільки як моделі, побудовані на засадах МГУА, так і власне генетичні алгоритми без сумніву відносяться до пробле-

матики штучного інтелекту, ще одним широко використовуваним при побудові систем ІІІ слугує *імітаційний підхід*. Даний підхід є класичним для кібернетики з одним з її базових понять – “чорною скринькою” – пристроєм, інформація про внутрішню структуру і зміст якої відсутні повністю, але мають місце специфікації сигналів на вході та виході. Об’єкт, поведінка якого імітується, саме й уявляє собою таку “чорну скриньку”, причому не важливо, що у неї та у моделі всередині і як він функціонує, головне, щоб наша модель в аналогічних ситуаціях поводи́ла себе аналогічно. Таким чином, тут після навчання і самоорганізації моделюється ще одна властивість людини – здібність копіювати те, що роблять інші, не удаючись в деталі, навіть це необхідно. Основним недоліком імітаційного підходу є низька інформаційна здатність більшості моделей, побудованих за його допомогою.

16.2. Деякі підходи до моделювання в комбінованих інтелектуальних системах

До основних задач в комбінованих інтелектуальних системах моделювання можна віднести:

- системний аналіз взаємодії змінних у складних об’єктах з метою визначення множини характеристичних змінних і розподілу її на підмножини вхідних та вихідних (незалежних і залежних) змінних;
- структурну й параметричну ідентифікації об’єктів, що моделюються, з уже відомими моделями взаємозмінних, які їх характеризують;
- якісне та кількісне прогнозування процесів з різною попередністю;
- планування, управління та прийняття рішень.

При цьому існує дещо принципова відмінність двох підходів до моделювання: а) базування тільки на досвіді спостережень за діями об’єкта, поданого у вигляді уявлень користувача, що склалися в нього про об’єкт; б) порівняння ряду розрахунків на основі дослідних даних як результатів проведення активного чи пасивного експерименту. *Перший* супроводжується використанням принципів *імітаційного моделювання*, а *другий* – *експериментального методу самоорганізації моделей* [26].

Уразливим місцем імітаційних методів є те, що користувач загальної моделі змушений зазначити закономірності для всіх елементів досліджуваної системи, в тому числі й для тих з них, які він знає гірше від інших або які взагалі найменшою мірою піддаються іміта-

ції. Тому найкраще застосовувати *комбінований метод*, у якому використовуються обидва підходи і який забезпечує найвищу ефективність дослідження складних об'єктів. При цьому моделі погано вивчених елементів формуються за допомогою експериментальних методів самоорганізації моделей, тоді як відомі елементи задаються безпосередньо користувачем моделі.

У цьому разі побудова динамічної моделі процесу зводиться до виконання таких етапів:

- виявлення змінних, які характеризують процес, і планування експериментів;
- реєстрування даних під час спостережень за об'єктом при проведенні експериментів (зміни параметрів слід проводити в однакові інтервали часу та кроків у просторі вимірювань; максимально припустимий крок таблиці треба визначати за теоремою відліків Котельникова – Шеннона);
- вибір фізичного закону, який доцільно покласти в основу моделі (елемент імітаційного моделювання). Якщо в основу моделювання той чи інший фізичний закон покласти важко, застосовується *індуктивний підхід*, згідно з яким *структура моделі оптимальної складності підбирається випробуванням кількох моделей-претендентів за критеріями селекції*. При цьому в просторі критеріїв кожній моделі відповідає певна характеристична точка. Відповідно до принципу самоорганізації систем вибирається деяка кількість точок (отже, й моделей), найближче розташованих до початку координат. Процедура довизначення моделі полягає в тому, що з відібраних кращих моделей береться одна за головним критерієм:
- заміна безперервного математичного опису його дискретним аналогом (різницеvim рівнянням);
- оцінювання й адаптування (облік за всіма даними) коефіцієнтів різницевого рівняння (елемент самоорганізації моделей).

Самоорганізація (тобто перебирання) моделей, так само як регресійний аналіз, належить до експериментальних методів моделювання, оскільки вона ґрунтується на обробці *пасивного* (спостереження за нормальною роботою об'єкта, як правило, при ручному керуванні ним) або *активного* (зміна зовнішніх дій здійснюється за спеціальною програмою) експериментів. Але якщо в регресійному аналізі структура моделі задається, то самоорганізація моделей передбачає визначення *моделі оптимальної складності*, коли при поступовому ускладненні структури моделі (наприклад, при збільшенні

кількості членів і степені поліноміальної моделі) значення зовнішніх критеріїв спочатку зменшуються, а потім збільшуються, тобто є їх мінімум, який визначає модель оптимальної складності.

Загалом процес самоорганізації подібний до роботи *генетичних алгоритмів*, які починають свою роботу з деякого набору даних і процедур і намагається комбінувати їх різним способом (відбувається *схрещування*). Далі з цих комбінацій відбираються найкращі за певним критерієм (відбувається *селекція*), і саме вони беруть участь у подальших схрещуваннях. Інколи до даних і процедур, які залучені до еволюційного відбору, вносяться певні випадкові зміни (відбуваються *мутації*). Генетичні алгоритми добре зарекомендували себе для розв'язання ряду погано формалізованих та важко-розв'язуваних задач.

Оскільки аналітична залежність того чи іншого критерію від складності моделі невідома, задача пошуку його мінімуму розв'язується *методом індукції*.

Твердження 16.1. Метод індукції супроводжується *повним* або *неповним* перебиранням багатьох моделей-претендентів.

У програмах, що реалізують самоорганізацію моделі, передбачаються блок генерування множини моделей-претендентів, блок обчислення значень певного критерію та вихідний блок вибору моделі оптимальної складності за мінімумом цього критерію.

16.3. Особливості використання методу групового урахування аргументів

Алгоритми самоорганізації моделей можуть бути *однорядними* або *багаторядними*, причому як перші, так і другі реалізують *метод групового урахування аргументів* (МГУА) [24].

Твердження 16.2. Алгоритми самоорганізації моделей належать до однорядних, якщо кількість членів повного опису $t < 20$.

Твердження 16.3. Алгоритми самоорганізації моделей належать до багаторядних, якщо кількість членів повного опису $20 < t < 1000$.

Означення 16.3 Метод повної математичної індукції – такий метод повного перебирання всіх варіантів моделі, які можна одержати із заданого повного опису, що реалізується в *однорядних* (комбінаторних) *алгоритмах* МГУА і застосовується для розв'язання визначених і перевизначених задач моделювання з кількістю членів повного опису t (найчастіше полінома), меншою від кількості точок у таблиці дослідних даних M (тобто $t < M$).

Означення 15.4. Метод неповної математичної індукції – такий метод раціонального перебирання варіантів моделі, за яким кількість випробуваних моделей значно зменшується з дуже малим ризиком втрати найоптимальнішої моделі (тут кількість моделей, що перебираються ЕОМ, збільшується не в степеневій, а в поліноміальній залежності від кількості змінних). Таким чином, для розв’язання *недовизначених задач* ($M < m$) створено багаторядні алгоритми МГУА, що й є засобом раціонального перебирання моделей [24].

При переході від однорядних алгоритмів до багаторядних можлива як втрата найоптимальнішої моделі, так і невелика “похибка багаторядності” (аналогічна статичній похибці систем управління), яку для поліномних алгоритмів можна визначити як різницю прогнозувань (рішень), одержаних за одно- та багаторядним алгоритмами. Багаторядний алгоритм, так само як система стабілізування, збігається до стану рівноваги – “точки згущення”.

Для багаторядних алгоритмів селекції моделі в алгоритмах самоорганізації дуже корисним є *принцип неостаточних рішень* [51], за яким з ряду в ряд селекції передається не одна, а кілька кращих за критеріями моделей, тобто на кожному з рядів зберігається “свобода вибору”, і тільки на останньому ряді вибирається єдина (для кожного ансамблю критеріїв) модель оптимальної складності.

Проте вибір критеріїв з ансамблю все ще є великою майстерністю, й тому метод самоорганізації моделей дістав назву *евристичного*. Для забезпечення самоорганізації моделі можна, наприклад, включити до складу ансамблю *критерій мінімуму зсуву* (КМЗ)

$$m_{\text{зс}}^2 = \sum_1^M (\overset{!}{g}_A - \overset{!}{g}_B)^2 / \sum_1^M g_{\phi}^2 \rightarrow \min, \quad (16.1)$$

де $\overset{!}{g}_A, \overset{!}{g}_B$ – виходи моделей однакової структури, коефіцієнти яких обчислено на частинах A та B вибірки; g_{ϕ} – фактичні (табличні) дані; $M = M_A + M_B$ – множина точок початкових даних, поділена на дві частини: M_A ; 70% і M_B ; 30%.

Можна також використати *критерій балансу змінних* (КБЗ)

$$H = \sum_1^M h_i^2 \rightarrow \min, \quad h_i = f_1(g_i) - f_2(g_i), \quad (16.2)$$

де h_i – непогодженість допустимого зв’язку функцій f_1 і f_2 .

Критерій КМЗ рекомендується застосовувати тоді, коли необхідно визначити фізичну модель, а критерій КБЗ – при розв’язуванні задачі довизначення моделі на довготерміновому прогнозуванні.

16.3.1. Однорядні (комбінаторні) алгоритми

Застосування МГУА знайшов у різних галузях знань, де використовуються структурна, параметрична ідентифікація та прогнозування. Раніше вважалося, що точність моделі можна підвищити виключно за рахунок урахування більшої кількості факторів та їх композицій. Проте такий підхід вимагав все більшої та більшої ретроспективи (періоду розгляду статистичних даних), що частіше за все було неможливим. Та ще й кількість структурних елементів моделі була обмеженою, що внаслідок теореми Геделя про неповноту свідчило про існування такої залежності, заданої таблично, яка не могла бути апроксимованою композицією даного набору елементів.

Теорема 16.1 (формулювання Геделя). Для будь-якої системи існує теорема, яка не може бути доведеною за допомогою аксіом цієї системи.

Автор МГУА запропонував використати *принцип зовнішнього доповнення*, базуючись на наступній теоремі Карла Вейєрштрасса:

Теорема 16.2 (1-ша теорема Вейєрштрасса). Якщо функція f неперервна на відрізку $[a, b]$, то для будь-якого $\varepsilon < 0$ існує багаточлен P з властивостями $\max_{x \in [a, b]} |f(x) - P(x)| < \varepsilon$.

Нехай матрицю $C = (\overline{X}_1^T, \overline{X}_2^T, \dots, \overline{X}_n^T, G^T)$ вибрано як початкові дані, де $\overline{X}_i^T, i = \overline{1, n}$ та G^T – вектори-стовпці вимірністю m , причому \overline{X}_i^T – фактори на вході, а G^T – характеристика на виході. Задача полягає в ідентифікації залежності

$$G = F(x_1, x_2, \dots, x_n) \quad (16.3)$$

поліномом Колмогорова-Габора

$$G = a_0 + \sum_{i=1}^n a_i x_i + \sum_{i < j} a_{ij} x_i x_j + \sum_{i < j < k} a_{ijk} x_i x_j x_k + \dots \quad (16.4)$$

Відомо, що при зростанні степеня цього полінома точність наближення їм функції $F(x)$ зростає, а потім спадає. В момент, коли точність максимальна, процес ускладнення полінома закінчується. Кількість точок експериментів може бути значно меншою за кількість членів полінома.

На першому етапі вибирається опорна функція. Частіше використовуються залежності виду:

$$1) g = a_0 + a_1 x_i x_j; 2) g = a_0 + a_1 x_i + a_2 x_j; 3) g = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j;$$

$$4) g = a_0 + a_1 x_i + a_2 x_j + a_3 x_i^2 + a_4 x_j^2 + a_5 x_i x_j.$$

Для першої функції необхідні дані хоча б трьох експериментів, для другої – чотирьох, для третьої – п'яти, для четвертої – семи. Це викликано тим, що для визначення коефіцієнтів буде використаний метод МНК. Позначимо $g_k = f(x_i, x_j)$, де f – одна з наведених залежностей, або, можливо, подібна до них.

Наступним кроком буде визначення МНК коефіцієнтів рівнянь другої опорної функції:

$$g_1 = f(x_1, x_2); g_2 = f(x_1, x_3); g_{n-1} = f(x_1, x_n); g_n = (x_2, x_3); \dots;$$

$$g_p = (x_{n-1}, x_n), \text{ де } p = C_n^2,$$

і т.д. для визначення МНК коефіцієнтів усіх вищенаведених рівнянь.

Отже, комбінаторних алгоритмах виконується перебирання всіляких окремих моделей із заданого базису (поліномного чи гармонійного) з вибором найкращої за заданим критерієм селекції. При перебиранні складність окремих моделей (тобто кількість аргументів або частот) поступово нарощується від 1 до m (m – кількість аргументів базисного набору функцій).

Таким чином, загальна схема комбінаторного алгоритму зводиться до виконання наступних операцій: за методом МНК визначаються коефіцієнти всіх окремих моделей при складності $K = \overline{1, m}$; для кожної з них обчислюється значення зовнішнього індивідуального чи комбінованого критерію селекції; єдина модель оптимальної складності вибирається за найменшим значенням вибраного критерію.

Таким чином, комбінаторний алгоритм МГУА ґрунтується на методі повної математичної індукції, оскільки при цьому не упускається жоден з можливих варіантів моделі, закладених у початковому повному базисі.

Приклад 16.1. Розглянемо кілька прикладів. Нехай задано повний квадратичний поліном для випадку трьох змінних

$$g = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_1^2 + b_5x_2^2 + b_6x_3^2 + b_7x_1x_2 + b_8x_1x_3 + b_9x_2x_3. \quad (16.5)$$

Маємо 10 членів полінома ($m=10$). У будь-якому разі повний поліном є лінійним за коефіцієнтами, для визначення яких використовується метод найменших квадратів. При цьому степені та коваріації початкових змінних розглядаються як еквівалентні перепозначені лінійні аргументи. Тоді відповідно до наведеної вище схеми алгоритм працює так. Спочатку визначаються всі моделі при $K=1$ (тобто такі, що складаються лише з одного аргументу): $g_1=b_0$; $g_2=b_1x_1$; $g_3=b_2x_2$; ...; $g_{10}=b_9x_2x_3$ (отже, таких моделей $C_{10}^1=10$).

Далі розглядаються всілякі моделі при $K=2$ (що складаються з двох аргументів):

$$g_{11} = b_0 + b_1 x_1; \quad g_{12} = b_0 + b_2 x_2; \quad \mathbf{K}; \quad g_i = b_0 + b_9 x_2 x_3; \quad g_j = b_1 x_1 + b_2 x_2; \quad \mathbf{K};$$

$g_k = b_1 x_1 + b_9 x_2 x_3; \quad \mathbf{K}; \quad g_{45} = b_8 x_1 x_3 + b_9 x_2 x_3$, яких уже буде $C_{10}^2 = 45$. Аналогічно одержуємо C_{10}^3 (для $K=3$), C_{10}^4 (для $K=4$) і так далі до $C_{10}^{10} = 1$ (для $K=m$), тобто до повного перебирання варіантів моделі.

Загальна кількість N_m окремих моделей, побудованих з m аргументів (доданків) повного полінома методом повного комбінаторного перебирання, обчислюється за формулою

$$N_m = \sum_{K=1}^m C_m^K = 2^m - 1. \quad (16.6)$$

Зі збільшення m швидко збільшується N_m , і тому можливості комбінаторного перебирання обмежені, а його максимальна складність K_{\max} задається виходячи з можливостей ЕОМ (як правило, $m \leq 18$).

Якщо йдеться про гармонічний аналіз полінома (16.5) з дискретним набором гармонік, тобто

$$x_i = E_i + L_i \sin \omega_i t + R_i \cos \omega_i t, \quad i = \overline{1, m}, \quad (16.7)$$

то коефіцієнти E_i , L_i , R_i визначаються за методом найменших квадратів на всіх M точках початкових даних. При цьому, починаючи з $K=2$, обчислюються коефіцієнти в моделях відносно вже визначених гармонічних аргументів $g_i = b_0 + b_1 x_i + b_2 x_j$. Обсяг перебирання становить $N_m = 2^m - 1$, і на такий алгоритм накладаються ті самі обмеження, що й на поліномний.

Гармонічні алгоритми з аналітичним визначенням або з перебиранням частот процесу, що визначені аналітично, теж мають ті самі обмеження за швидкодією, але вони принципово відмінні від звичайного гармонічного алгоритму, оскільки тут частоти не призначаються, а визначаються аналітично, і тому вони максимально наближені до істинних гармонік коливального процесу.

16.3.2. Багаторядні алгоритми

Такі алгоритми можуть не передбачати обчислення залишків (рис. 16.1, а) або передбачати їх (рис. 16.1, б) [78].

Спочатку розглянемо перший випадок, вважаючи, що початковий склад аргументів, з якого починається процедура багаторядної селекції моделі процесу, будується на нульовому ряді алгоритму, який по-різному організовується в поліномних і гармонічних алгоритмах. Наприклад, у класі алгебраїчних функцій найбільш загальною моделлю є поліном Колмогорова-Габора, що містить v змінних:

$$g = b_0 + \sum_{i=1}^v b_i \alpha_i + \sum_{i=1}^v \sum_{j=1}^v b_{ij} \alpha_i \alpha_j + \sum_{i=1}^v \sum_{j=1}^v \sum_{l=1}^v b_{ijl} \alpha_i \alpha_j \alpha_l + \mathbf{K} \quad (16.8)$$

і є сумою лінійних, квадратичних, кубічних та інших членів.

Після перетворення всіх його складових дістанемо лінійний поліном $g = b_0 + b_1x_1 + b_2x_2 + \mathbf{K} + b_mx_m$, члени якого і становлять початковий набір аргументів, побудований на нульовому ряді багаторядного алгоритму.

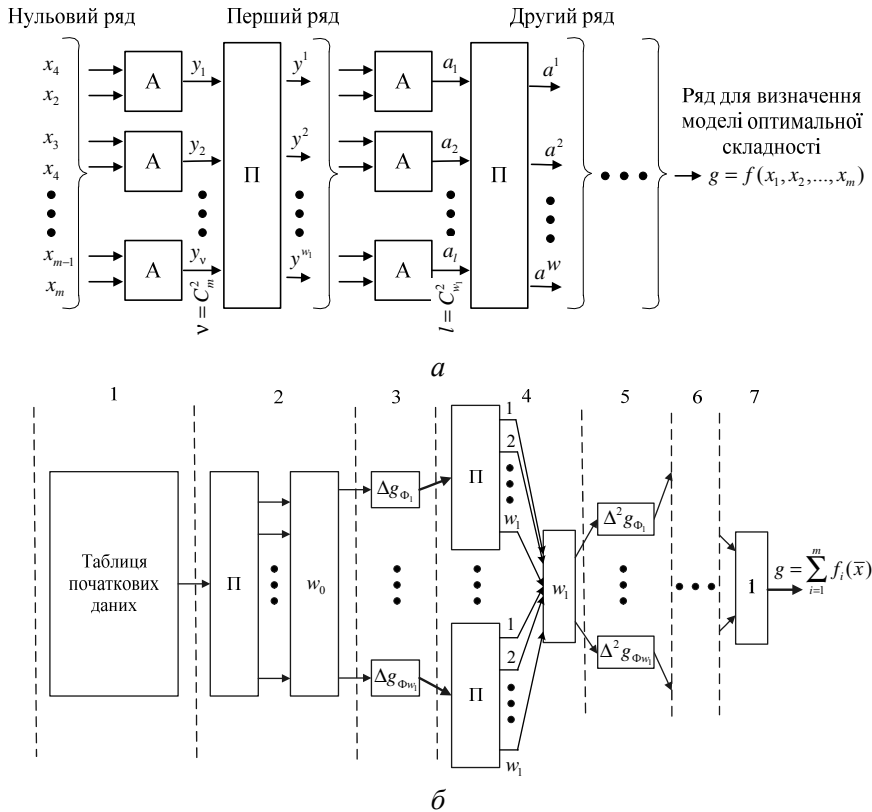


Рис. 16.1. Багаторядні алгоритми МГУА (А – алгоритм попарної обробки аргументів; П – пороговий відбір w кращих моделей)

У випадку різницевої моделі до аналогічного вигляду зводиться модель, яку можна одержати перепозначенням усіх початкових змінних, їх запізнювань і заданих нелінійних функцій.

Для багаторядних гармонічних алгоритмів (як з обчисленням залишків, так і без їх обчислення) початковий склад аргументів повинен будуватися на нульовому ряді так само, як і в комбінаторному

гармонічному алгоритмі з перебиранням частот, визначених аналітично: спочатку одержуємо l моделей у вигляді суми однієї, двох, трьох і так далі до m гармонік з некрatними частотами, а потім всі наявні в цих моделях $m=0,5(l+1)$ гармонічних складових перепозначимо як нові аргументи:

$$x_i = L_i \sin \omega_i t + R_i \cos \omega_i t, \quad i = \overline{1, m}.$$

Після таких перепозначень матимемо однотипний базовий набір початкових аргументів як для поліномних, так і для гармонічних алгоритмів. Тому охарактеризуємо багаторядні схеми перебирання загального вигляду.

Багаторядні алгоритми, як правило, працюють за такою загальною схемою:

- *перший ряд* – на основі даних таблиці спостережень будуються окремі описи всіх попарних комбінацій початкових (перепозначених) аргументів з розташуванням за методом найменших квадратів вихідної змінної y :

$$y_1 = f_1(x_1, x_2); \quad y_2 = f_2(x_1, x_3); \quad \mathbf{K}; \quad y_n = f_n(x_{m-1}, x_m),$$

і з цих $v = C_m^2$ моделей відокремлюється деяка кількість w_1 найкращих за зовнішнім критерієм;

- *другий ряд* – одержанні змінні беруться як аргументи другого ряду і знову будуються всі окремі моделі двох аргументів:

$$a_1 = \xi_1(y_1, y_2); \quad a_2 = \xi_2(y_1, y_3); \dots; \quad a_l = \xi_l(y^{w_1-1}, y^{w_1}), \quad (16.9)$$

де $l = C_{w_1}^2$.

З них за зовнішнім критерієм відокремлюються w_2 кращих моделей як змінні для наступного ряду і т. д. Ряди нарощуються доти, поки знижується значення зовнішнього критерію.

Кожний такий опис може бути лінійною $f = b_0 + b_1 x_i + b_2 x_v$ чи нелінійною $f = b_0 + b_1 x_i + b_2 x_v + b_3 x_i x_v$ функцією двох змінних (або $f = b_0 + b_1 x_i + b_2 x_v + b_3 x_i x_v + b_4 x_i^2 + b_5 x_v^2$), коефіцієнти яких можна визначити за методом МНК, маючи відповідно три, чотири (або шість) точок спостережень у навчальній послідовності. Виключивши проміжні змінні після призупинення алгоритму, дістанемо модель, кількість коефіцієнтів якої значно перевищує кількість точок спостережень. Таким чином, у багаторядних алгоритмах МГУА відтворюється *схема масової селекції*.

Структура багаторядних алгоритмів МГУА з обчисленням залишків (рис. 16.1, б) містить: 1 – таблицю початкових даних; 2 –

нульовий ряд, на якому за комбінаторним алгоритмом МГУА здійснюється вибір w_0 кращих моделей; 3 – обчислення перших різниць (залишків) $\Delta g_\phi - g^*$ (тут g_ϕ – фактичні дані таблиці спостережень; g^* – вихід моделі), які утворюють w_0 таблиць з даними перших залишків; 4 – перший ряд селекції, на якому за комбінаторним алгоритмом МГУА визначаються $w_0 N_m$ моделей, з яких вибираються w_1 кращих; 5 – обчислення других різниць (залишків) $\Delta^2 g_\phi = \Delta g_\phi - \Delta g^*$ (тут $\Delta^2 g_\phi$ – фактичне значення першого залишку; Δg^* – вихід моделі); 6 – наступні ряди селекції й обчислення залишків; 7 – останній ряд, на якому здійснюється вибір моделі оптимальної складності.

При нарощуванні рядів селекції такого алгоритму комбінований критерій селекції $\rho^2 = m_{3c}^2 + \Delta^2(D)$ монотонно зменшується і селекція зупиняється зовсім, щойно системний критерій починає збільшуватись. Тому сигналом для призупинення лічби є досить мале зменшення системного критерію, коли об'єктивний характер вибору моделі втрачається.

Справді, поблизу мінімуму кривої комбінаторного критерію селекції розміщується послідовність характеристичних точок, які відповідають кращим моделям. Таким чином, не обов'язково вибирати модель, що відповідає найнижчій точці на цій кривій. Порівнявши за вторинним критерієм (наприклад, точністю прогнозування, балансом змінних тощо) кілька кращих моделей, можна вибрати оптимальну модель за двома критеріями.

Якщо будуються моделі до довгострокового прогнозу, то замість пари критеріїв $m_{3c}^2, \Delta^2(D)$ потрібно узяти пару критеріїв $m_{3c}^2, J_2(M)$. Останнє стосується тільки поліномних різницевих моделей, оскільки аналог критерію $J_2(M)$ для алгебраїчних і гармонічних моделей поки, що добре ще не розроблено.

16.3.3. Структура комбінованого алгоритму

Як уже згадувалось, у структурі кожного з алгоритмів МГУА можна виділити три основні блоки [24]:

- перетворення початкових даних відповідно до вибраної системи опорних (базисних) функцій, в якій відшукується модель (базис може бути поліномним, тригонометричним, у вигляді різницевих рівнянь тощо);

- генерування (перебирання) повної або неповної множини ускладнених окремих моделей у вибраному базисі;
- обчислення значень деякого критерію селекції з властивостями зовнішнього доповнення та послідовного відбирання кращих за цим критерієм моделей.

Відмінність усіх алгоритмів МГУА полягає в структурі “генератора моделей”, тобто схемі обчислень, яка реалізується в другому блоці.

За цією ознакою алгоритми МГУА поділяються на *багаторядні, комбінаторні та мішані*, для яких характерними є специфічні схеми нарощування складності окремих описів. Нижче як приклад детальніше розглядаються обчислювальні особливості комбінаторного алгоритму МГУА для кожного з його основних блоків.

Блок формування базису. Розглянемо випадок поліномних опорних функцій. Якщо задано значення деяких вхідних змінних a_1, \dots, a_e об'єкта моделювання (ОМ) і максимального степеня полінома, то кількість доданків m у повному поліномі степеня v_{\max} при e змінних визначається однозначно:

$$m = \prod_{j=1}^e \frac{v_{\max} + j}{j} = \frac{(v_{\max} + e)! - (v_{\max})!}{v!}. \quad (16.10)$$

Повний поліном при цьому записується у вигляді

$$y = \sum_{i=1}^m b_i \prod_{j=1}^e a_j^{v_{ij}} = \sum_{i=1}^m b_i x_i, \quad (16.11)$$

де кожний узагальнений лінійний аргумент x_i є нелінійною функцією початкових змінних a_j :

$$x_i = \prod_{j=1}^e a_j^{v_{ij}}. \quad (16.12)$$

Степені v_{ij} аргументів можуть набувати цілі значення $0, 1, \dots, v_{\max}$, такі, що сума їх не перевищує значення v_{\max} , тобто

$\sum_{j=1}^e v_{ij} \leq v_{\max}$. Для формування степенів аргументів зручно організувати

процедуру одержання послідовності r -розрядних чисел $v_i = (v_{li}, \mathbf{K}, v_{ri})$ з основою $1, 2, \dots, v_{\max} + 1$ і відбирати з них ті, що задовольняють вказане обмеження. Наприклад, при $r=2$ (два аргументи) і $v_{\max} = 2$ маємо послідовність $m=6$ чисел $v_i, i=\overline{1, 6}$, а саме: 00,

10, 01, 20, 11, 02, що відповідає повному поліному

$$y = b_1 + b_2 a_1 + b_3 a_2 + b_4 a_1^2 + b_5 a_1 a_2 + b_6 a_2^2.$$

Отже, члени x_i повного полінома є базисним набором опорних функцій для комбінаторного алгоритму МГУА. Наведена процедура дає змогу сформувати матрицю вимірювань узагальнених аргументів $X[M \times m]$, де M – кількість точок вимірювань. Нехай ці точки розподілено (підряд) на три послідовності: навчальну A завдовжки M_A , перевірну B завдовжки M_B та екзаменаційну D завдовжки M_D , причому $M_A + M_B + M_C + M_D = M$.

Значимо, що для переходу до другого базису, наприклад різнцевого чи тригонометричного, досить обчислити відповідно аргументи a згідно з (16.9) і потім скористатися процедурою формування матриці X .

Блок генерування (перебирання) окремих моделей. До основних операцій, що виконуються цим блоком, належать: формування структури чергової окремої моделі; формування відповідної нормальної системи рівнянь; розв'язування одержаної системи рівнянь (оцінювання коефіцієнтів моделі).

Формування структур окремих моделей формалізується за допомогою двійкового структурного вектора $u = (u_1, u_2, \dots, u_m)$. Якщо елемент u_i цього вектора набуває одиничного значення, то відповідний i -й аргумент включається в окрему модель, а якщо нульове – не включається ($i = \overline{1, m}$).

Найпростішим способом організації зміни стану вектора u є такий, за яким спочатку одержують усі можливі варіанти розміщення у векторі u однієї одиниці (загалом $C_m^1 = m$ варіантів), потім двох одиниць (загалом $C_m^2 = \frac{1}{2}m(m+1)$ варіантів) і так далі аж до m одиниць ($C_m^m = 1$ варіант). Загальна кількість варіантів згідно з формулою (16.6) становитиме $N_m = 2^m - 1$ різних структур (повне перебирання). Проте програмна реалізація цього способу є далеко не швидкодіючою, і таке перебирання структур доцільно застосовувати тільки в алгоритмах їх неповного перебирання.

Більш компактною й універсальнішою є схема зміни двійкового вектора за принципом роботи двійкового лічильника, в останній розряд якого додається одиниця. Цей спосіб алгоритмічно є дуже

простим і зручним саме при повному перебиранні, незважаючи на те, що кількість і склад аргументів в окремих моделях увесь час змінюються (наприклад, при $m=3$ послідовно формуються структури 001, 010, 011, 100, 110, 111). До того ж він дає змогу навіть організувати рекурентну процедуру перебирання моделей.

Для формування нормальної системи рівнянь, що відповідає черговому структурному вектору, можна піти формальним шляхом: із стовпців повної матриці X , зазначених одиничними елементами u_i , складається окрема матриця X_i , а потім обчислюються елементи $X_i^T, X_i, X_i^T u$ нормальної матриці. Проте при повному перебиранні цей формальний шлях є найгіршим, оскільки призводить до багаторазового обчислення тих самих скалярних добутків. Наприклад, для структур 010, 011, 110 тричі обчислюється та сама величина

$$x_2^T x_2 = \sum_{j=1}^M x_{2j}^2. \text{ Тому в комбінаторному алгоритмі МГУА досить тільки}$$

один раз обчислити матриці повної нормальної системи, яка містить елементи всіх окремих нормальних систем:

$$X^T X = \begin{pmatrix} x_1^T x_1 & x_1^T x_2 & \mathbf{K} & x_1^T x_m \\ x_2^T x_1 & x_2^T x_2 & \mathbf{K} & x_2^T x_m \\ \mathbf{K} & \mathbf{K} & \mathbf{K} & \mathbf{K} \\ x_m^T x_1 & x_m^T x_2 & \mathbf{K} & x_m^T x_m \end{pmatrix}, \quad X^T Y = \begin{pmatrix} x_1^T y \\ x_2^T y \\ \mathbf{K} \\ x_m^T y \end{pmatrix}. \quad (16.13)$$

Блок відбирання окремих моделей за зовнішнім критерієм.

Обчислення в цьому блоці, як і в попередньому, організовується з урахуванням застосовуваних зовнішніх критеріїв. Якщо критерій є несиметричним (типу регулярності $\Delta^2(B)$), то оцінювання коефіцієнтів b_l виконуються на одній послідовності, а значення критерію відшукується на іншій (що й забезпечує його зовнішній характер):

$$\Delta_l^2(B) = \left(y_B - X_{lB} \hat{b}_l(A) \right)^T \left(y_B - X_{lB} \hat{b}_l(A) \right). \quad (16.14)$$

Якщо критерій є симетричним, то для тієї самої структури u_l коефіцієнти b_l обчислюються на обох послідовностях (A та B), а потім визначається, наприклад, критерій мінімуму зсуву

$$m_{zc}^2 = \left(X_l \hat{b}_l(B) - X_l \hat{b}_l(A) \right)^T \left(X_l \hat{b}_l(B) - X_l \hat{b}_l(A) \right). \quad (16.15)$$

Селекція (відбирання) кращих моделей за заданим критерієм J зазвичай виконується не наприкінці їх перебирання (коли сформовано всі окремі моделі), а в його процесі. З цією метою заповнюються значення критерію J для заданої кількості w перших моделей, після чого значення критерію J_l для кожної наступної моделі порівнюються з найгіршим J_{\max} із w значень. Якщо $J_l < J_{\max}$, то нова модель (структура, оцінки коефіцієнтів і критерій) запам'ятовується замість найгіршої; якщо ж $J_l \geq J_{\max}$, то ця модель (тобто J_l) з подальшого аналізу виключається. Після закінчення перебирання моделей ті w з них, які залишилися, і є найкращими з усіх N_m за заданим критерієм.

В алгоритмах самоорганізації моделей застосовується також відбирання їх за двома й більше критеріями. Проте, наприклад, використання *комбінованих критеріїв* $d^2 = \beta m_{sc}^2 + (1 - \beta) \Delta^2(B)$, що складаються із суми двох наведених критеріїв, утруднюється складністю вибору вагового коефіцієнта β . Тому зазвичай кілька критеріїв використовуються послідовно – за ступенем важливості тієї чи іншої властивості моделей, що оцінюються за певним критерієм. Наприклад, за критерієм m_{sc}^2 відбираються w_1 найменше зсунених моделей, потім з них за критерієм $\Delta^2(B)$ здійснюється вибір $w < w_1$ найточніших на перевірній послідовності.

Для підвищення вірогідності та завадостійкості відбирання моделей застосовується інформація третьої частини D вибірки. При цьому спочатку виконується етап адаптації коефіцієнтів – перелік оцінювань w_1 моделей, що були відібрані на попередньому етапі, за точками двох перших вибірок AUB . Потім з w_1 моделей відбираються $w < w_1$ за новим критерієм регулярності $\Delta^2(D)$. Наприклад, двоетапне відбирання $[m_{sc}^2 \rightarrow \Delta^2(D)]$ моделей набагато підвищує завадостійкість моделювання за комбінаторним алгоритмом МГУА.

В алгоритмах МГУА зазвичай виконується ще один етап обчислень – *оцінювання якості відібраних кращих моделей*. При цьому обчислюються, наприклад, квадратична похибка апроксимації $\delta^2(A+B)$ (точність оцінювання на AUB) і похибка екстраполяції (прогнозування) $\Delta^2(D)$.

16.4. Штучні нейронні сітки, засновані на МГУА

Алгоритми МГУА можуть ототожнюватися з Мадаліною, елементами якої є Н-Адаліни Уїдроз-Хоффа (див. розд. 1). Оскільки

в алгоритмах МГУА аргументи та проміжні змінні об'єднані попарно в групи, які складаються з двох змінних, точніше щодо МГУА говорити як про *метод попарного угруповання аргументів*, і тому ШНС, яка заснована на МГУА і складається з t шарів, може реалізувати поліном у степені 2^m [48, 182] (рис. 16.2).

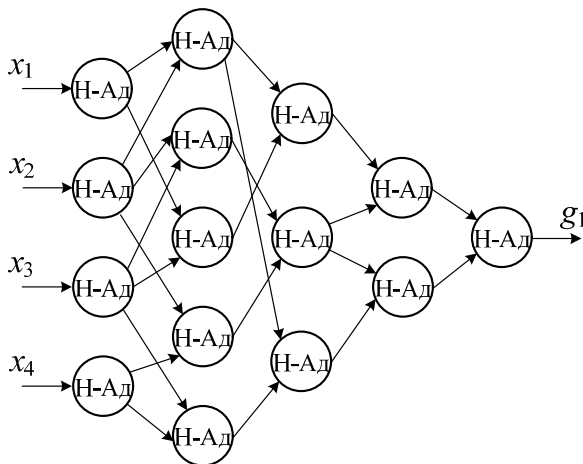


Рис. 16.2. Штучна нейронна сітка, заснована на МГУА

Як вже наголошувалося вище, критерієм оптимізації в МГУА є мінімум СКП при перебиранні варіантів з навчальними та перевірковими послідовностями даних і виборі кращих варіантів за допомогою *порогового принципу*. Як і в МГУА, цей критерій використовується в ШНС, заснованій на МГУА, принаймні на чотирьох етапах її функціонування [48]:

1. при обчисленні вагових коефіцієнтів кожної з Н-Адалін;
2. при виборі кращих комбінацій пар аргументів, тобто входів Н-Адалін;
3. при визначенні величини порогів, на підставі яких відбувається відбирання кращих варіантів. При цьому кількість ураховуваних пар змінних в кожному шарі багатозарової нейронної сітки саме й визначаються порогоми, які обчислюються за мінімумом СКП;
4. при визначенні степені нелінійності використовуваного опису і відбирання утворюючих його функцій.

Оскільки дана ШНС призначена для *моделювання і прогнозування складних процесів, сигнали на вході і на виході яких є випадковими*, перед початком навчання сітки здійснюється нормування (центрування і нормалізація змінних за формулами (див. розд. 3.3):

$$\hat{x}_i = (x_i - \bar{x}_i) / \sigma_{x_i}; \hat{g}_i = (g_i - \bar{g}_i) / \sigma_{g_i}, \quad (16.16)$$

де $x_i, g_i, \bar{x}_i, \bar{g}_i$ – змінні і вході та на виході та їх середні значення, відповідно; $\sigma_{x_i}, \sigma_{g_i}$ – середньоквадратичні відхилення x_i та g_i .

Тільки після цього формуються всілякі пари вхідних сигналів Н-Адалін першого шару і здійснюється корекція їх ваг за правилом Уідроу-Хоффа (див. *розд.* 4.2).

З досягненням мінімального значення сумарної СКП першого шару корекція ваг цього шару закінчується і в подальшому їх значення не змінюються. Процедура навчання (селекція сигналів, корекція тощо) повторюється для другого шару и т.д. до досягнення всією сіткою бажаного або припустимого значення СКП.

Контрольні завдання

1. Дати оцінку видам ітеративних процесів розвитку систем.
2. Розкрити вміст поняття “самоорганізація”.
3. Прокоментувати визначення еволюційного моделювання.
4. Пояснити відмінність автоматів Мілі та Мура.
5. Дати математичне визначення критерію мінімуму середньоквадратичної похибки.
6. Розкрити вміст генетичного алгоритму як еволюційного методу досліджень явищ.
7. Охарактеризувати МГУА як базис генетичного алгоритму.
8. Пояснити етапи формування моделей оптимальної складності в МГУА.
9. Дати визначення методу індукції з повним і неповним перебиранням моделей-претендентів в МГУА.
10. Навести умови використання однорядних і багаторядних алгоритмів перебирання моделей-претендентів МГУА.
11. Прокоментувати теореми Геделя та Вейерштрасса (1-ша теорема) щодо факторів та їх комбінацій на точність моделі.
12. Навести покрокове функціонування багаторядного алгоритму МГУА.
13. Розкрити структуру комбінованого алгоритму МГУА.
14. Охарактеризувати топологію штучної нейронної сітки, заснованої на МГУА.
15. Прокоментувати чотири етапи функціонування ШНС, заснованої на МГУА.

Презентації

1. Нейронні сітки на генетичних алгоритмах та МГУА



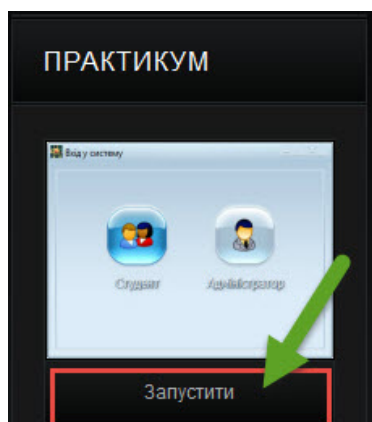
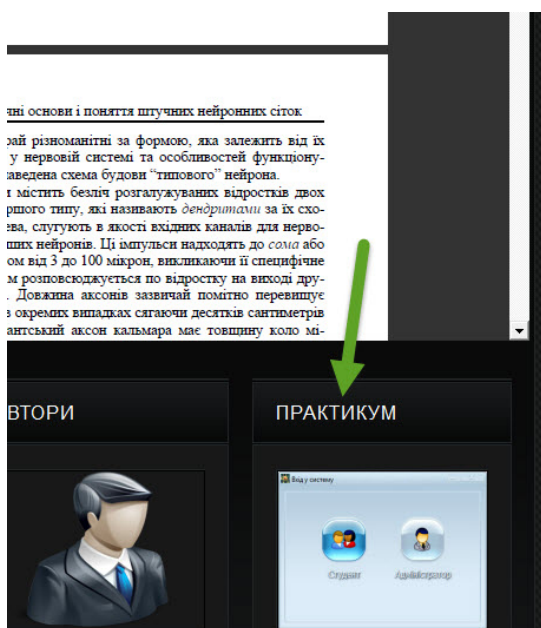
2. Метод групового урахування аргументів в нейромоделюванні



Практичні заняття

Метод групового урахування аргументів в нейромоделюванні

Для запуску практичних занять на
вкладці "Практикум" натисніть
"Запустити"



Розділ 17

Каскадно-кореляційні нейронні сітки

*Краще досконало виконати незначну частку справи,
аніж зробити погано у десять разів більше*
Аристотель

З ким поведешся, від того й наберешся
Луцій Анней Сенека молодший

17.1. Методи нарощування нейронної сітки

В алгоритмах редукції (див. розд. 23.3) як відправна використовується надлишкова архітектура НС, яка в процесі навчання або після його завершення спрощується шляхом виключення найменш значущих ваг і нейронів.

Протилежний підхід полягає у первісному включенні до складу сітки невеликої кількості схованих нейронів (часто вони взагалі відсутні), і з розвитком процесу навчання їх кількість поступово збільшується. Серед багатьох існуючих методів розширення НС можна виокремити алгоритми: Мезарда-Надаля [134], Мерчанда [134] та Лі – Тафтса [165], в яких всі навчаючі вибірки проектуються в одновимірний простір з наступним вибором такої гіперплощини, яка б відокремила дані бажаного класу від інших. Багаторазово повторюючи цю процедуру на залишившийся множині вибірок, в решті решт можна забезпечити повний поділ даних. Мінімізація кількості гіперплощин (схованих нейронів) в методі [165] досягається застосуванням булівої алгебри, зокрема, мапи Карно.

Перелічені методи мають відносно низьку ефективність при великій розмірності вхідного вектора.

Одним з найбільш відомих методів розширення НС вважається алгоритм каскадної кореляції (АКК) С. Фалмана [107; 108], який розглядається далі.

17.2. Каскадна кореляція

Загальні відомості. Каскадно-кореляційні нейронні сітки (ККНС) були введені С. Фалманом в 1990 році та відносяться до ШНС прямого поширення зі змінюваною (зростаючою) архітектурою. Як і інші ШНС, ККНС містять вхідні, приховані шари та шари на виході, причому приховані шари з'являються тільки в процесі на-

вчання сітки. Відноситься до ШНС з алгоритмом навчання з вчителем, причому може використовуватись будь-який алгоритм навчання одношарових НС (С. Фалман використовував QuickProp).

Означення 17.1. *Каскадно-кореляційна НС* – це багатошарова сітка спеціальної (каскадної) архітектури, яка дозволяє навчати її конструктивним способом: при припиненні збіжності до НС поступово додаються нові нейрони і при подальшому навчанні модифікуються зв'язки тільки доданих нейронів.

Такий підхід дозволяє визначити розмір сітки, адекватний розв'язуваній задачі, і суттєво скоротити обчислювальні витрати на навчання, причому більшість параметрів навчання (такі як архітектура, розмір НС або параметри швидкості навчання) визначаються самим процесом навчання.

Означення 17.2. *Каскадна кореляція (КК)* – це архітектура навчання з «учителем», яка будує топологію з найменшою кількістю вузлів багатошарової НС прямого поширення.

У КК існує дві переваги: досліджувачу нема необхідності пильнуватися щодо топології НС; вона суттєво швидша за звичайні алгоритми навчання.

17.3. Навчання каскадно-кореляційних нейросіток

Алгоритм каскадної кореляції. Каскадна кореляція об'єднує дві ідеї: перша – *каскадна топологія*, в якій сховані шари додаються один раз і залишаються незмінними; друга – *навчаючий алгоритм*, який утворює і додає нові сховані вузли. При цьому АКК побудований таким чином, що навчається, фактично, тільки один шар навчуваних зв'язків, що дозволяє обмежитись спрощеною (та більш обчислювально простою) реалізацією алгоритму зворотного поширення похибки. Для кожного нового схованого вузла АКК прагне максимізувати значення кореляції між виходом нового вузла і похибкою НС.

Алгоритм КК реалізується наведеним нижче способом:

1. *Крок 1.* Каскадна кореляція починається з мінімальної НС, яка складається тільки з вхідного шару і шару на виході. Обидва шари – повнозв'язні, тобто існують всі можливі зв'язки між усіма вузлами при врахуванні обмежень, які накладаються на НС прямого поширення;
2. *Крок 2.* Навчаються звичайним алгоритмом всі зв'язки, що ведуть до шару на виході, доти, доки помилка не переставатиме зменшуватись;

3. *Крок 3.* Утворюються так звані *вузли-кандидати*. Кожний вузол-кандидат зв'язаний з усіма вхідними шарами і зі всіма існуючими схованими вузлами. Між вузлами-кандидатами і вузлами на виході зв'язок відсутній;
4. *Крок 4.* Виконується спроба максимізувати кореляцію між задіянням вузлів-кандидатів і залишковою похибкою НС шляхом навчання зв'язків, які ведуть до вузлів-кандидатів. Навчання виконується за допомогою звичайних алгоритмів і припиняється, коли показники кореляції перестають покращуватись;
5. *Крок 5.* Вибирається вузол-кандидат з максимальною кореляцією, фіксуються ваги його вхідних зв'язків, і він додається у сітку. Для перетворення вузла-кандидата у схований вузол створюються зв'язки між ним і вузлами на виході.

Цей процес називається “заволодінням”. Після заволодіння кандидат стає повноцінним новим вузлом НС. Далі повертаємося до пункту 2 і заново навчаємо всі коефіцієнти на зв'язках на виході, в тому числі і для нового вузла. Цей процес додавання нового вузла і перенавчання шару на виході повторюється, доки помилка не стає незначною у межах припустимої. Оскільки нові вузли отримують вхідні з'єднання від старих, кожний новий вузол ефективно додає новий шар до сітки.

Функціональна інтерпретація алгоритму навчання ККНС.

Таким чином, навчання НС базується на 2-х принципах:

- покрокове інкрементне введення нейронів у приховані шари;
- навчання знову введених нейронів при збереженні ваг нейронів, що вже перебувають у сітці.

На рис. 17.1 зображено зміну архітектури в процесі навчання.

На самому початку процесу навчання ШНС (*Крок 1*) складається тільки з двох шарів: вхідного та вихідного. Кількість входів та виходів залежить від специфіки задачі та не змінюється. Нейрони на виході ШНС можуть мати лінійні або нелінійні активаційні функції (або використовувати їхню комбінацію). Початкові значення ваг – випадкові.

На вході ШНС подаються навчальні пари (*Крок 2*) й за допомогою обраного алгоритму здійснюється налаштування ваг сітки до досягнення мінімального значення вихідної помилки. Використовуються будь-які методи навчання. С. Фалман в своїй роботі використовував алгоритм *QuickProp*, який характеризується швидкою збіжністю до точки розв'язку і полягає в тому, що *крок градієнтного спу-*

ску по поверхні похибки визначається параболічним наближенням за попередніми точками. Якщо вихідна помилка є припустимою з точки зору очікуваного значення, то процес навчання закінчується, якщо ні – перехід до Кroku 3.

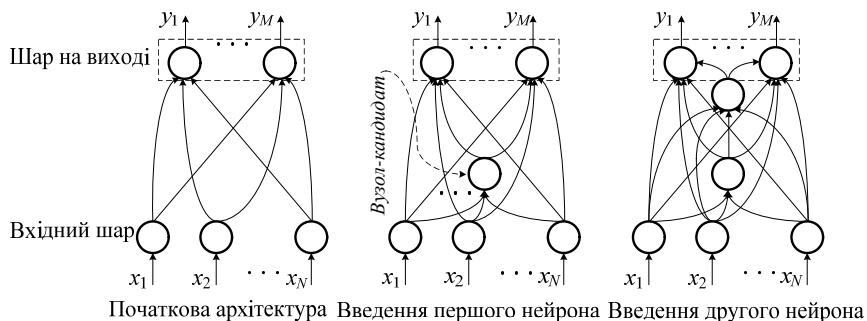


Рис. 17.1. Покрокові зміни в архітектурі КНС

В цьому випадку в прихований шар вводиться нейрон прихованого шару – вузол-кандидат, на вхід якого подаються сигнали з виходів усіх нейронів вхідного та прихованого (якщо вони є) шарів. Знову введений нейрон не зв'язується поки з нейронами шару на виході. В момент підключення нейрона до структури фіксуються ваги його вхідних зв'язків (в майбутньому не змінюються). З нейронами на виході поки що зв'язок не встановлюється, а в майбутньому вага цих зв'язків буде постійно уточнюватись.

Кожний додаваний нейрон – вузол-кандидат утворює новий шар.

Ваги зв'язків вузла-кандидата піддаються навчанню (Крок 4), але сигнал на його виході на даному етапі нікуди не подається. В такому стані нейрон – вузол-кандидат навчається з використанням множини навчаючих вибірок. Мета навчання полягає в тому, щоб підібрати значення вагових коефіцієнтів введеного нейрона, що максимізують кореляцію S_i між активністю цього нейрона (сигналом на виході) та значенням похибки на виході ШНС.

Дана залежність визначається коефіцієнтом кореляції S :

$$S_i = \sum_k \left| \sum_p (z_{pi} - \bar{z}_i)(e_{pk} - \bar{e}_k) \right|, \quad (17.1)$$

де k - кількість вихідних нейронів; p - кількість образів (навчальних вибірок); z_{pi} – вихід знову введеного i -го нейрона для p -образу; e_{pk} – залишкова помилка шару на виході, що обчислюється як різниця не-

обхідних і реально обчислених сигналів на виході; \bar{z}_i та \bar{e}_k – середні значення z_{pi} та e_{pk} .

Для максимізації значення функції S_i необхідно визначити її похідну відносно ваг w_{ij} всіх вхідних зв'язків нейрона-кандидата:

$$\frac{\partial S_i}{\partial w_{ij}} = \sum_k \sum_p \sigma_k f' \alpha(x_{pi}) z_{pi} (e_{pk} - \bar{e}_k), \quad (17.2)$$

де σ_k – значення кореляції між j -тим нейроном на виході та помилкою на виході k -го нейрона для p -образу; $f' \alpha(*)$ – перша похідна використовуваної функції (*похідна функція активації, розрахована для p -образу навчальної вибірки відносно зваженої суми його вихідних сигналів*); x_{pi} та z_{pi} – i -та компонента вхідного сигналу сітки та вихід j -го нейрона вхідного шару для p -го образу (*імпульс, отриманий від i -го збуджуючого сигналу*).

Як вже наголошувалося, виконується максимізація S на основі будь-якого алгоритму оптимізації (QuickProp у С. Фалмана), який використовується для підбору ваг нейронів на виході.

Після досягнення максимуму кореляції значення ваг вхідних сигналів введеного нейрона фіксуються, і його виходи з'єднуються з усіма нейронами шару на виході. Відповідні вагові коефіцієнти обчислюються відповідно до обраного алгоритму навчання.

Для досягнення найкращих результатів кореляційного навчання навчається декілька нейронів-кандидатів одночасно.

Беручи до уваги, що навчання розпочинається з випадкових значень, то кожний нейрон-кандидат отримує різні кінцеві значення коефіцієнту кореляції. Серед них в ШНС вводиться той, що має найбільше значення S (Крок 5).

Виконуване для прискорення процедури паралельне кореляційне навчання декількох нейронів-кандидатів зменшує ймовірність потрапляння в точку локального мінімуму та введення в сітку нейрона з погано підібраними вагами, які на наступних етапах не можна буде відкоригувати.

Кожний нейрон-кандидат, що претендує на включення в нейросіткову структуру, може мати свою функцію активації (сигмоїдальну, гаусівську, радіальну, тощо). Перемагає той нейрон, який краще адаптується до умов, створених масивом навчальних вибірок. Результат такого підходу – ККНС може об'єднувати нейрони з різними функціями активації, які підібрані алгоритмом з урахуванням ролі, яку вони відіграють в структурі сітки.

Зображені на рис. 17.1 структури ШНС з 1-м та з 2-ма нейронами в прихованому шарі саме й ілюструють процедуру покрокового додавання нейронів – вузлів-кандидатів у приховані шари саме й

ілюструють процедуру покрокового додавання нейронів – вузлів-кандидатів у приховані шари ККНС.

Якщо необхідне значення залишкової помилки не досягнуто, у ШНС вводиться наступний нейрон, для якого спочатку визначаються ваги зв'язків з нейронами вхідного та прихованого шарів. Навчання другого нейрона подібне до навчання першого. Кроки 3 та 4 повторюються доти, поки для всіх пар образів не буде досягнута необхідна точність (величина припустимої помилки).

Каскадна кореляція навчається значно шоріше, ніж сітка зі зворотним поширенням похибки, завдяки декількох факторів:

- по-перше, спочатку тільки один шар вагових коефіцієнтів навчається у будь-який даний момент часу. Ніколи не виникає необхідності поширювати інформацію про помилки обернено через зв'язки, що позбавляє драматичного уповільнення, яке є типовим при навчанні багат шарових ШНС зі зворотним поширенням похибки.

- по-друге, це “жадібний” алгоритм: кожна нова одиниця (вузол-кандидат) захоплює настільки багато “залишеної” похибки, наскільки тільки може. В стандартній сітці зі зворотним поширенням похибки всі сховані вузли міняються одночасно, конкуруючи за різні задачі, які мають бути досягнуті – це занадто повільний і іноді ненадійний процес.

Модифікації алгоритму навчання ККНС. Алгоритм допускає інші реалізації. Зокрема, максимізацію кореляції можна здійснювати, вводячи не один новий нейрон прихованого шару, а декілька, використовуючи як початкові значення вагових коефіцієнтів різні випадкові значення. Оскільки ці нейрони між собою не пов'язані і їхнє навчання здійснюється на тих самих образах, то можлива одночасна (паралельна) корекція їхніх вагових коефіцієнтів, що значно прискорює процес навчання.

Крім того, замість критерію максимуму кореляції для налаштування ваг можна використати критерій мінімуму середньоквадратичної помилки. Тоді алгоритм має елементи, що попереджують зациклювання в точці локального мінімуму, який виникає в результаті роботи нейрона на фазі насичення сигмоїдальною кривою, де через близькість до нуля похідної функції активації процес навчання практично зупиняється.

Вага w_{ij} на k -му кроці алгоритму змінюється за правилом:

$$\Delta w_{ij}(k) = -\eta_k \left[\frac{\partial E(w(k))}{\partial w_{ij}} + \gamma w_{ij}(k-1) \right] + \alpha_{ij}^k \Delta w_{ij}(k-1). \quad (17.3)$$

Відмінність від класичного градієнтного методу полягає у наявності трьох доданків у виразі (17.3): перший $\left(\eta_k \frac{\partial E(w(k))}{\partial w_{ij}} \right)$ – відповідає оригінальному алгоритму найшвидшого сходження; другий $(\eta_k \gamma w_{ij}(k-1))$ – призначений для мінімізації абсолютних значень ваг; третій $(\alpha_{ij}^k \Delta w_{ij}(k-1))$ – фактору моменту. Коефіцієнт γ (зазвичай має малу величину 10^{-4}) – це фактор, що призводить до зменшення ваг аж до можливого розриву відповідних зважених зв'язків. Константа η_k – це коефіцієнт навчання, який в даному алгоритмі може мати ненульове значення η_0 (як правило від 0,01 до 0,6) на початку навчання, коли (якщо) $\Delta w_{ij}(k-1) = 0$ або $\left[\frac{\partial E(w(k))}{\partial w_{ij}} + \gamma w_{ij}(k) \right] \Delta w_{ij} > 0$, або нульове значення в іншому випадку.

Важливу роль в алгоритмі QuickProp відіграє фактор моменту $(\alpha_{ij}^k \Delta w_{ij}(k-1))$, який адаптується до поточних результатів процесу навчання. Відповідно до алгоритму С. Фалмана *коефіцієнт моменту* α_{ij} визначається індивідуально до кожної ваги за правилом:

$$\alpha_{ij}(k) = \begin{cases} \alpha_{\max}, & \text{якщо } \beta_{ij}(k) > \alpha_{\max}; \\ \beta_{ij}(k), & \text{якщо } S_{ij}(k) \Delta \omega_{ij}(k-1) \beta_{ij}(k) < 0, \end{cases} \quad (17.4)$$

$$S_{ij}(k) = \frac{\partial E(\omega(k))}{\partial \omega_{ij}} + \gamma \omega_{ij}(k); \quad \beta_{ij}(k) = \frac{S_{ij}(k)}{S_{ij}(k-1) - S_{ij}(k)},$$

де константа α_{\max} – максимальне значення коефіцієнта моменту. Фалманом рекомендується $\alpha_{\max} = 1,75$.

Також відома *спрощена версія алгоритму QuickProp*, в якій значення ваг змінюються в відповідності до правила:

$$\Delta \omega_{ij}(k) = \begin{cases} \alpha_{ij} \Delta \omega_{ij}(k-1), & \text{якщо } \Delta \omega_{ij}(k-1) \neq 0 \\ \eta_0 \frac{\partial E}{\partial \omega_{ij}}, & \text{якщо } \Delta \omega_{ij}(k-1) = 0; \end{cases} \quad (17.5)$$

$$\alpha_{ij}(k) = \min \left\{ \frac{S_{ij}(k)}{S_{ij}(k-1) - S_{ij}(k)}, \alpha_{\max} \right\}; \quad S_{ij}(k) = \frac{\partial E(\omega(k))}{\partial \omega_{ij}}.$$

В ній зменшена кількість керуючих параметрів та спрощена формула уточнення значень ваг. Проте, згідно досліджених С. Фалманом результатів, ефективність модифікованого алгоритму порівняно однакова з оригінальним.

17.4. Переваги й недоліки каскадно-кореляційних нейросіток

Основні *переваги* ККНС над іншими ШНС:

- *відсутня необхідність апріорного вибору* архітектури сітки; починаючи з найпростішої, що складається тільки із вхідного шару та шару на виході шляхом поступового ускладнення, сітка сама визначає свою архітектуру, що відповідає обраному критерію;
- *швидкість процесу навчання*, в порівнянні з іншими нейросітками, зростає внаслідок того, що нейрони одного шару не пов'язані, то ж між ними відсутня конкуренція;
- *відсутня необхідність перенавчання* всієї системи при отриманні нових образів шляхом корекції лише вагових коефіцієнтів зв'язків нейронів прихованого та на виході шарів;
- *процес навчання сітки значно простіший* для обчислення, оскільки під час навчання корегуються ваги тільки одного шару, і сигнали в ШНС прямують тільки в одному напрямку;
- *сітка зберігає засвоєну раніше інформацію* при повторному навчанні, що забезпечує її значну стійкість.

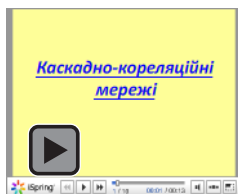
До *недоліків* слід віднести той факт, що, оскільки система фактично сама себе будує, топологія побудованої за методом каскадної кореляції ШНС *майже завжди неоптимальна*.

Контрольні завдання

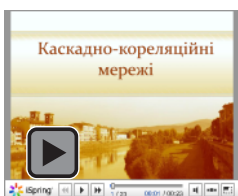
1. Пояснити, в чому полягають два підходи щодо редукції нейронних сіток.
2. Охарактеризувати каскадну кореляцію як процес і каскадно-кореляційну нейронну сітку як багатошарову спеціальної архітектури.
3. Прокоментувати алгоритм каскадної кореляції як процедуру “заволодіння”.
4. Розкрити принцип швидкої збіжності з кроками градієнтного спуску по поверхні похибки як параболічне наближення за попередніми точками.
5. Дати тлумачення модифікованому алгоритму навчання каскадно-кореляційної нейронної сітки – максимізаційній корекції.
6. Розкрити переваги й недоліки каскадно-кореляційних НС порівняно з іншими штучними нейронними сітками.

Презентації

1. Каскадно-кореляційні мережі



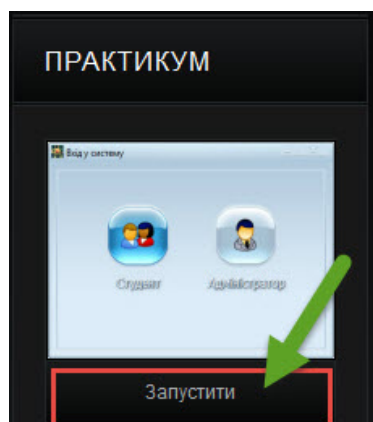
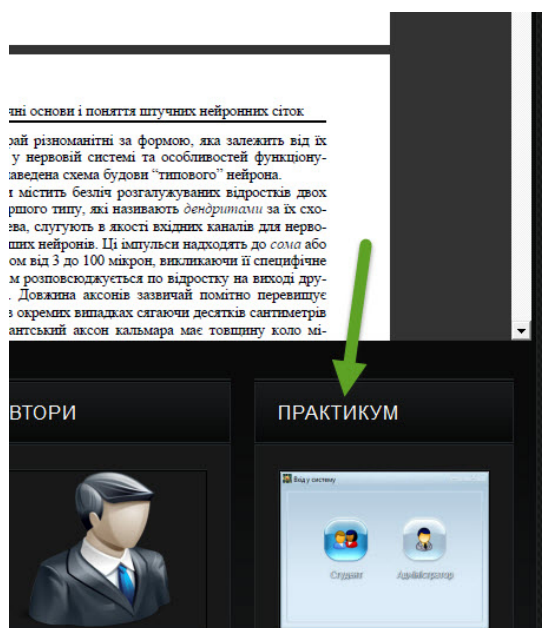
2. Каскадно-кореляційні мережі



Практичні заняття

Каскадно-кореляційні нейронні сітки

Для запуску практичних занять на
вкладці "Практикум" натисніть
"Запустити"



Розділ 18

Нейронні сітки адаптивної резонансної теорії

*Запозичення з одного джерела – плагіат, з двох – компіляція,
з трьох – творчість, або аналітичне дослідження*

Порада починаючим науковцям

Шляхи, які нікуди не ведуть, заводять далі всіх інших
Жорж Вольфром

18.1. Адаптивна резонансна теорія

Дилема стабільності-пластичності сприйняття. Проблема стабільності-пластичності є однією з самих складних і важко розв'язуваних задач при побудованні штучних систем, моделюючих сприйняття. Характер сприйняття зовнішнього світу живими організмами (і, перш за все, людиною) постійно пов'язаний з розв'язанням дилеми, чи є деякий образ “ноюю” інформацією, а отже реакція на нього повинна бути пошуково-пізнавальною, зі збереженням цього образу в пам'яті, або цей образ є варіантом “старої”, вже знайомої картинки, і в цьому випадку реакція організму повинна відповідати раніше накопиченому досвіду. Спеціальне запам'ятовування цього образу в останньому випадку не вимагається. Таким чином, сприйняття одночасно *пластичне*, адаптовано до нової інформації, і при цьому воно *стабільне*, тобто не руйнує пам'ять про старі образи.

Розглянуті в попередніх розділах підручника нейронні системи не пристосовані до розв'язання цієї задачі. Так, наприклад, багаторазовий персептрон, який навчається за методом зворотного поширення, запам'ятовує весь пакет навчаючої інформації, при цьому образи навчаючої вибірки пред'являються в процесі навчання багаторазово. Спроби потім навчити персептрон новому образу приведуть до модифікації синаптичних зв'язків з неконтрольованим, взагалі кажучи, руйнуванням структури пам'яті про попередні образи – знадобиться процедура повного перенавчання. Таким чином, персептрон не спроможний до запам'ятовування нової інформації без необхідного повного перенавчання НС.

Аналогічна ситуація має місце і в сітках Кохонена [153] та Ліппмана-Хеммінга [166], які навчаються на основі самоорганізації. Ці сітки завжди видають позитивний результат при класифікації. Тим

самим ці НС не в змозі виокремити нові образи від спотворених або зашумлених версій старих образів.

Суть проблеми або, скоріше, *дилеми стабільності-пластичності пам'яті* полягає в тому, що НС при сприйнятті нової інформації повинна не просто додати її в пам'ять, але й співвіднести нову інформацію з вже запам'ятованою і, якщо є яка-небудь схожість між новою та старою інформацією, скорегувати запам'ятовану інформацію, тобто “прийняти до уваги” нову інформацію.

Отже, неможливість за допомогою вже відомих НС розв'язати дилему стабільності-пластичності з'явилась однією з основних причин розробки принципово нових конфігурацій нейросіток. Дослідження з проблеми стабільності-пластичності, виконані під керівництвом Стефана Гроссберга, привели до формування адаптивної резонансної теорії (adaptive resonance theory) (АРТ) і створення нейросіткових архітектур нового типу на її основі. Загальні положення АРТ, висунутих С. Гроссбергом у 1976 році, детально викладені в основоположних роботах Стефана Гроссберга і Гейла Карпентера (S. Grossberg, G. Carpenter) [122; 92].

Принцип адаптивного резонансу. В АРТ уведено декілька суттєвих елементів, поданих нижче через відповідні твердження.

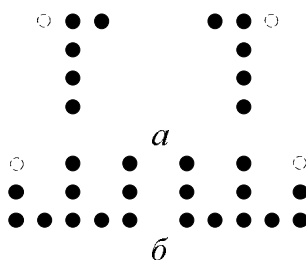
Твердження 18.1. (Наявність внутрішнього “детектора новизни”). Якщо критерій новизни задовольняється, то нейросітка класифікує пред'явлений образ як новий, якщо критерій не задовольняється, то виникає *резонанс на моді*, відповідній одному з раніше запам'ятованих образів, при цьому сітка змінює запам'ятований раніше образ – *адаптивний резонанс*. Це твердження є носієм *першої суттєвої умови АРТ*.

Привабливою особливістю НС з адаптивним резонансом є те, що вони зберігають пластичність при запам'ятовуванні образів і, в той же час, запобігають модифікацію старої пам'яті. Нейросітка має внутрішній детектор новини – тест на порівняння пред'явленого образу із вмістом пам'яті. При вдалому пошуку в пред'явлений образ класифікується з одночасно уточнюючою модифікацією синаптичних нейрона, виконавшого класифікацію. Про таку ситуацію говорять, як про виникнення адаптивного резонансу в ШНС у відповідь на пред'явлення образу. Якщо резонанс не виникає у межах деякого заданого порогового рівня ρ , то успішним вважається тест новизни, і образ сприймається сіткою як новий. Модифікації ваг нейронів, не зазнавших резонансу, при цьому не відбувається.

Іншим важливим поняттям в АРТ є так званий *шаблон критичних рис* (critical feature pattern) інформації.

Твердження 18.2. Не всі риси (деталі), подані в деякому образі, є суттєвими для системи сприйняття – *результат розпізнавання визначається наявністю специфічних “критичних рис”* (особливостей) в образі. Це твердження уявляє собою *другу суттєву умову АРТ*. В ідеалі, виокремлення критичної частини інформації повинно відбуватися автоматично в процесі роботи та навчання сітки, на основі її індивідуального досвіду.

Розглянемо ілюстрацію до поняття “критична риса” в образі на прикладі, наведеному на рис. 18.1.



**Рис. 18.1. Графічна інтерпретація поняття
“критична риса” образу**

Обидва малюнки на рис. 18.1 мають загальну властивість: в кожній з пар чорна точка в правому верхньому куті на рис. 18.1, *а* замінена на білу (пунктирна точка), тоді як така ж зміна має місце на лівому верхньому куті. Така зміна для нижньої пари малюнків (на рис. 18.1, *б*) ця пара уявляє літеру “Ш”), очевидно, є не більше аніж шумом, і обидва образи є спотвореними версіями того ж самого зображення – літери “Ш”. Отже, змінені точки не є для цього образу критичними.

Зовсім інша ситуація має місце для верхньої пари малюнків (рис. 18.1, *б*). Тут така ж зміна точок виявляється занадто суттєвою для образу (літера “Т”), то ж правий та лівий малюнки є відмінними образами. Таким чином, одна і та ж риса образу може бути не суттєвою в одному випадку і критичною в іншому. Задачею ШНС буде формування вірної реакції в обох випадках: “пластичне” рішення щодо виникнення нового образу для пари “Т” і “стабільне” рішення щодо співпадання для пари “Ш”. При цьому виокремлення критичної частки інформації повинно отримуватися автоматично в процесі роботи та навчання ШНС на підставі її “індивідуального досвіду”.

Зазначимо, що у загальному випадку одного лише переліку рис (навіть якщо він заздалегідь виконаний людиною з урахуванням визначених умов подальшої роботи ШНС) може виявитись недостат-

ньо для успішного функціонування штучної нейронної системи – критичними можуть виявитись специфічні зв'язки між декількома окремими рисами.

Третьою значною умовою АРТ є необхідність самоадаптації алгоритму пошуку образів в пам'яті. Нейронна сітка працює в постійно змінюваних умовах, через що напередвизначена схема пошуку, яка відповідає деякій структурі інформації, може в подальшому виявитись неефективною при зміні цієї структури. В АРТ це досягається введенням спеціалізованої *орієнтуючої системи*, яка самоузгоджено припиняє подальший пошук резонансу в пам'яті і приймає рішення щодо новизни інформації. Орієнтуюча система також навчається в процесі роботи.

Твердження 18.3. *Орієнтуюча система* сама приймає рішення щодо припинення пошуку в пам'яті та переході до навчання новому вектору. Орієнтуюча система забезпечує самоадаптацію алгоритму пошуку образів у пам'яті, оскільки НС працює в постійно змінюваних умовах.

У випадку наявності резонансу АРТ припускає можливість прямого доступу до образу пам'яті, який відгукнувся на резонанс. В цьому випадку “шаблон критичних рис” виступає ключем-прототипом для прямого доступу.

Твердження 18.4. (Правило 2/3, або “два з трьох”). Кожний елемент (нейрон або “вентиль”) шару порівняння має три входи; сигнал на виході формується за наявності сигналів на двох входах. Ця *четверта умова АРТ* формує активацію нейрона в шарі порівняння.

Твердження 18.5. *Зустрічне очікування* – процедура відшукування сіткою у пам'яті раніше запам'ятованого вектора, який “на її думку” найбільш відповідає вхідному, витягає його з пам'яті і порівнює з вхідним, щоб перевірити за заданим критерієм схожості дійсну відповідність вхідному вектору. Ця *п'ята умова АРТ* формує механізм прийняття рішень НС.

Ці та інші особливості адаптивної резонансної теорії знайшли своє відображення в нейросіткових архітектурах, які й дістали таку ж назву – АРТ.

Загальні зауваження. Є декілька різновидів сіток АРТ. Історично першою з'явилась НС, яка отримала назву АРТ-1 [94]. Ця сітка орієнтована на опрацювання образів, які містять двійкову інформацію. Подальший крок – архітектура АРТ-2 [90] – орієнтована на роботу як з двійковими, так і з аналоговими образами. У повідомленні

про систему АРТ-3 [91] йдеться про розповсюдження адаптивної резонансної теорії Стефана Гроссберга та Гейла Карпентера на багатшарові нейроархітектури.

Парадигма сіток АРТ використовує функцію неконтрольованого навчання і аналізує значні вхідні дані, виявляє можливі ознаки та класифікує образи у вхідному векторі.

В загальному випадку сітка АРТ складається з двох взаємопов'язаних шарів нейронів (рис. 18.2), розташованих між вхідним шаром і шаром на виході. Кожний вхідний образ нижчого шару резонансу стимулює очікуваний образ на вищому шарі, який пересилається знов до нижчого шару, щоб впливати на наступний вхід. Це створює ефект “резонансу” між нижчим та вищим шарами для полегшення сіткової адаптації образів.

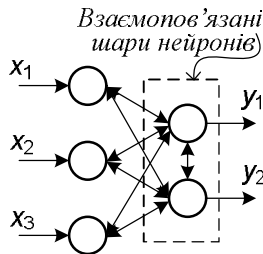


Рис. 18.2. Узагальнена архітектура сітки АРТ

Сітка АРТ переважно використовується у біологічному моделюванні, проте існують і деякі технічні застосування. Головним обмеженням сіткової архітектури є її шумова чутливість: навіть мала частка шуму на вхідному векторі знижує узагальнюючі можливості навченої сітки.

Повний опис структури сітки адаптивного резонансу і теорії її роботи, подані в наведеній вище публікації С. Гроссберга і Г. Карпентера, досить громіздкий, через що далі використовується більш спрощена “калька” з книги Ф. Уоссермена [59], доповнена загальним описом особливостей АРТ-2 та нової архітектури АРТ-3.

Теорема АРТ. Основні властивості сіток АРТ сформульовані в роботі [94] у вигляді доведених теорем. Виокремимо наступні:

1. *Можлива вірна класифікація без фази пошуку* (на основі прямого доступу) для навчаючих векторів за умови досягнення стабільного стану навчання.

2. *Процес пошуку стійкий.* Після визначення нейрона-переможця в сітці не буде збуджень інших нейронів в результаті змі-

ни векторів виходу шару порівняння; тільки сигнал скидання може викликати такі зміни.

3. *Процес навчання стійкий*. Навчання ваг нейрона-переможця не приведе в подальшому до перемикавання на інший нейрон.

4. *Процес навчання скінченний*. Навчений стан для заданого набору образів буде досягнутий за скінченну кількість ітерацій, при цьому подальше пред'явлення цих образів не викличе циклічних змін значень ваг.

18.2. Нейронна сітка АРТ-1

Особливості функціонування сітки АРТ-1. Зупинимось детальніше спочатку на класичній сітці АРТ-1.

Нейросистема АРТ-1 є класифікатором вхідних двійкових образів за декількома сформованими сіткою категоріями, тобто *вхідні сигнали в цій моделі бінарні*. Сітка АРТ-1 *навчається без учителя* і реалізує алгоритм кластеризації, дуже схожий на алгоритм “послідовного лідера”. Відповідно до цього алгоритму перший вхідний сигнал вважається зразком першого кластера. Кажуть, що вхідний сигнал “прямує за лідером” і належить першому кластеру, якщо відстань до зразка першого кластера менша за поріг. Інакше кажучи, рішення приймається у вигляді збудження одного з нейронів шару розпізнавання в залежності від міри схожості образу на “шаблон критичних рис” даної категорії. Якщо ця міра схожості є незначною, тобто образ не відповідає жодній з наявних категорій, то для нього формується новий клас, і другий вхідний сигнал стає зразком другого кластера, який в подальшому буде модифікуватися і уточнюватися іншими образами, формуючи свій шаблон критичних ознак. Для подання нової категорії відводиться новий, раніше не задіяний нейрон в шарі розпізнавання. Таким чином, кількість кластерів зростає з часом і залежить як від значення порога, так і від метрики відстані, яка використовується для порівняння вхідних сигналів і зразків класів.

Основна частина сітки АРТ-1 схожа з сіткою Хеммінга (див. розд. 6.3) і відрізняється зворотними зв'язками від нейронів на виході до вхідних; крім того є можливість виключати нейрон на виході з максимальним значенням відповідності і проводити тестування відповідності вхідного сигналу і зразків кластерів, як того вимагає алгоритм “послідовного лідера” (наводиться нижче). Сітки АРТ-1 складаються з п'яти функціональних модулів (рис. 18.3): двох шарів нейронів - *шару порівняння* та *шару розпізнавання*, а також трьох керуючих спеціалізованих нейронів – *скидання*, *керування 1* (K_1) та *керування 2* (K_2).

Твердження 18.6. Розмірність входу і виходу сітки обмежені при програмній реалізації тільки можливостями обчислювальної системи, на якій моделюється НС, а при апаратній реалізації – технологічними можливостями.

Твердження 18.7. Ємність НС збігається з кількістю нейронів другого шару і може збільшуватись в процесі її функціонування.

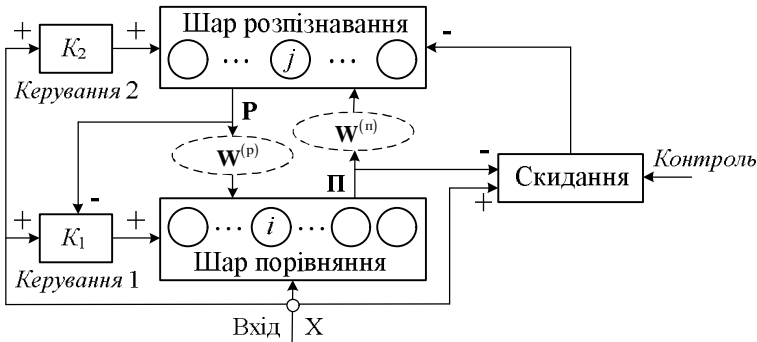


Рис. 18.3. Загальна схема нейронної сітки АРТ-1

Шар порівняння. Шар порівняння порівнює сигнал на виході шару розпізнавання з сигналом X , який надходить до сітки. Початкове значення нейрона керування 1 приймається дорівнюючим одиниці: $K_1=1$. Вхідний двійковий вектор X надходить на шар порівняння, який спочатку пропускає його без змін, при цьому вектор на виході шару порівняння $\Pi=X$. Це досягається застосуванням так званого “правила 2/3” для нейронів шару порівняння (це правило ще інтерпретується як “два з трьох”). Кожний з нейронів цього шару має три двійкових входи – сигнал від відповідної компоненти вектора X , сигнал від нейрона K_1 керування 1 і сигнал зворотного зв’язку з шару розпізнавання P (який в початковий момент дорівнює нулю). Для активації нейрона в шарі порівняння необхідно, щоб хоча б два з трьох сигналів дорівнювали одиниці (отже, “правило 2/3”), що й досягається у початковий момент входом від керування 1 та активними компонентами вектора X . Отже, щоб i -й компонент вектора Π мав одиничне значення (тобто вихід i -го нейрона дорівнював одиниці), необхідно, щоб як мінімум дві з трьох компонент K_1 , x_i , $w_j^{(p)}$ дорівнювали одиниці, тобто:

$$\pi_i = \begin{cases} 1, & \text{якщо } x_i w_j^{(p)} \cup x_i K_1 \cup w_j^{(p)} K_1 = 1; \\ 0, & \text{у противному випадку.} \end{cases} \quad (18.1)$$

Приклад 18.1. Для наочності розглянемо приклад. Нехай

$$\mathbf{X} = 110101101101$$

$$\mathbf{W}^{(p)} = 101100100100,$$

тоді:

$$\text{при } K_1 = 0 \quad \mathbf{\Pi} = 100100100100;$$

$$\text{при } K_1 = 1 \quad \mathbf{\Pi} = 111101101101.$$

Отже, умова “правила 2/3” в векторі $\mathbf{\Pi}$ при $K_1 = 0$ виконується для першого, четвертого, сьомого та десятого нейронів, а при $K_1 = 1$ активуються ще й другий, третій, шостий, дев’ятий та дванадцятий нейрони вектора $\mathbf{\Pi}$.

Динаміка (активність) i -го нейрона шару порівняння є досить складною і для спрощення подання процесу далі описується в узагальненій формі наступним диференціальним рівнянням:

$$\xi \frac{d\pi_i}{dt} = -\pi_i + (1 - \lambda_1 \pi_i) \cdot I_i^+ - (\chi_1 + \psi_1 \pi_i) \cdot I_i^-, \quad (18.2)$$

де $I_i^+ = x_i + w_j^{(p)}$; $I_i^- = \sum_h f(w_h)$ – відповідно узагальнені підсилюючі і гальмуючі сигнали, які надходять на i -й нейрон; ξ , λ_1 , χ_1 , ψ_1 – деякі невід’ємні параметри.

Шар розпізнавання. Шар розпізнавання класифікує вхідні вектори. Сформований шаром порівняння сигнал $\mathbf{\Pi}$ надходить на входи нейронів шару розпізнавання, кожний з яких має вектор ваг $w_j^{(p)}$ – дійсних чисел; при цьому збуджується тільки один нейрон цього шару, вектор ваг якого найближчий до $\mathbf{\Pi}$. Це досягається, зокрема, за рахунок механізму латерального гальмування типу “Переможець забирає все” (див. розд. 7). Вихід нейрона-переможця встановлюється дорівнюваним одиниці, решта нейронів повністю загальмовані. Сигнал зворотного зв’язку від нейрона-переможця знов надходить в шар порівняння через синаптичні ваги $w_j^{(p)}$. Вектор ваг $\mathbf{W}^{(p)}$, по суті, є носієм критичних рис категорії, визначуваної вигравшим нейроном.

Динаміка j -го нейрона шару розпізнавання описується рівнянням, аналогічним до (18.2):

$$\xi \frac{dp_j}{dt} = -p_j + (1 - \lambda_2 p_j) \cdot I_j^+ - (\chi_2 + \psi_2 p_j) \cdot I_j^-, \quad (18.3)$$

де $I_j^+ = \phi(w_j^{(p)}) + \mathbf{W}_j^{(p)}$ – сигнал властивого додатного зворотного зв’язку j -го нейрона; $I_j^- = \sum_{h \neq j} \phi(w_h)$ – сума сигналів від’ємних зворотних зв’язків, які надходять з усіх інших нейронів шару розпізнавання; ξ , λ_2 , χ_2 , ψ_2 – деякі невід’ємні параметри, які вибираються таким чином, щоб активувався тільки один нейрон, на вхід якого надходить найбільший сигнал $w_j^{(n)}$ (j -й нейрон є переможцем).

Вихід нейрона K_1 керування 1. Реалізує механізм співставлення вхідного сигналу з сигналом, який надходить з шару розпізнавання, формуючи сигнал K_1 за правилом $K_1 = x_1 \overline{W}$. Значення істинності сигналу K_1 наведене у табл. 18.1:

Таблиця 18.1

Значення істинності сигналу K_1

Компоненти		K_1
x_1	W	
0	0	0
0	1	0
1	0	1
1	1	0

Таким чином, вихід нейрона K_1 дорівнює одиниці, тільки коли вхідний образ **X** має ненульові компоненти, тобто цей нейрон виконує функцію детекції факту надходження образу на вхід. Проте, коли виникає ненульовий відгук **P** нейронів шару розпізнавання, значення керування 1 занулюється, тобто $K_1=0$.

Сигнал нейрона K_2 керування 2. Реалізує логічну функцію “або” булевих компонент $K_2 = x_1 \cup x_2 \cup \dots \cup x_N$ вхідного вектора **X**. Він також встановлюється в одиницю при ненульовому векторі **X**. Задачею цього нейрона є погашення активності на шарі розпізнавання, якщо в сітку не надійшло ніякої інформації.

Отже, при генеруванні відгуку **P** шару розпізнавання вихід $K_1=0$, і тепер нейрони шару порівняння активуються сигналами образу **X** та відгуку **P**. Правило “двох третин” приводить до активації тільки тих нейронів шару порівняння, для яких **X**, і **P** є одиничними. Таким чином, вихід **П** шару порівняння тепер вже не дорівнює в точності **X**, а містить лише ті компоненти **X**, які відповідають критичним рисам перемігшої категорії. Цей механізм в теорії АРТ дістав назву *адаптивної фільтрації образу X*.

Тепер задачею системи є встановити, чи достатнім є набір цих критичних рис для остаточного віднесення образу **X** до категорії нейрона-переможця. Цю функцію здійснює нейрон скидання, який вимірює схожість між векторами **X** і **П**. Вихід нейрона скидання визначається відношенням кількості одиничних компонент у векторі **П** до кількості одиничних компонент образу **X** на вході. Якщо це відношення нижче деякого визначеного рівня схожості, нейрон видає сигнал скидання, означаючий, що рівень “резонансу” образу **X** з рисами пропонованої категорії не достатній для позитивного висновку щодо завершення класифікації. Умова (критерій *F*) виникнення сигналу скидання:

$$F = |\mathbf{P}|/|\mathbf{X}| < \rho, \quad (18.4)$$

де $\rho < 1$ – параметр схожості. Цей параметр показує, наскільки повинен вхідний сигнал збігатися з одним із запам'ятованих зразків, щоб вони вважалися схожими. Близьке до одиниці значення параметра схожості (або – порогового рівня) вимагає майже повного збігу.

Сигнал скидання виконує повне гальмування нейрона-переможця-“невдахи”, який не приймає в подальшому участі в роботі НС.

Алгоритм функціонування сітки (алгоритм “*послідовного лідера*”). Функціонування сітки відбувається за наступними кроками:

Крок 1. Ініціалізація сітки:

$$w_{ij}^{(n-p)}(0) = 1; w_{ij}^{(p-n)}(0) = 1/(1+N), 0 < i < (N-1); \quad (18.5)$$

$$0 < j < (M-1); 0 < \rho < 1,$$

де $w_{ij}^{(n-p)}$ – синаптична вага зв'язку від i -го нейрона першого шару (порівняння) до j -го нейрона другого шару (розпізнавання) на момент часу t ; $w_{ij}^{(p-n)}$ – синаптична вага зв'язку від i -го нейрона другого шару (розпізнавання) до j -го нейрона першого шару (порівняння) на момент часу t ; ρ – значення порогу (параметра схожості).

Ваги $w_{ij}^{(n-p)}$ та $w_{ij}^{(p-n)}$ визначають зразок, відповідний нейрону j .

Крок 2. Вхідні сигнали пред'являються шару нейронів на виході аналогічно тому, як це робиться в сітці Хеммінга (див. розд. 6.3).

Крок 3. Обчислення значень відповідності:

$$y_j = \sum_{i=1}^N w_{ij}(t) x_i. \quad (18.6)$$

Значення відповідності обчислюються паралельно для всіх зразків, запам'ятованих у сітці, аналогічно сітці Хеммінга.

Крок 4. Вибір зразка з найбільшою відповідністю:

$$y_j = \max(y_j). \quad (18.7)$$

Ця операція виконується за допомогою латерального гальмування.

Крок 5. Порівняння з порогом:

$$\text{якщо } \frac{\|\mathbf{W} \cdot \mathbf{X}\|}{\|\mathbf{X}\|} > \rho - \text{перехід до кроку 7, інакше - до кроку 6.} \quad (18.8)$$

В умові (18.8)

$$\|\mathbf{X}\| = \sum_{j=1}^N x_j; \|\mathbf{W} \cdot \mathbf{X}\| = \sum_{j=1}^N w_{ij}^{(n)} \times x_j.$$

На цьому кроці обчислюється відношення скалярного добутку вхідного сигналу і зразка з найбільшим значенням відповідності

до кількості одиничних біт вихідного сигналу. Значення відношення порівнюється з порогом, введеним на першому кроці. Якщо значення відношення більше порогового, то вхідний сигнал вважається схожим на зразок з найбільшим значенням відповідності. У цьому випадку зразок модифікується шляхом виконання операції AND (логічне "І"), тобто новий зразок має значення зразка на попередньому кроці + значення вхідного сигналу.

Якщо ж значення відношення менше порогового, то вважається, що вхідний сигнал відрізняється від всіх зразків, і він розглядається як новий зразок. У цьому випадку у сітку вводиться нейрон, який відповідає новому зразку, і обчислюються значення синаптичних ваг за його участю.

Крок 6. Усування прикладу з найбільшим значенням відповідності.

Вихід нейрона з найбільшим значенням відповідності тимчасово встановлюється рівним нулю і не приймає участі у кроці 4.

Крок 7. Адаптація прикладу з найбільшим значенням відповідності:

$$w_{ij}^{(n-p)}(t+1) = w_{ij}^{(n-p)}(t)x_j; \quad w_{ij}^{(p-n)}(t+1) = \frac{w_{ij}^{(n-p)}(t)x_j}{0,5 + \sum_{j=1}^N w_{ij}^{(n-p)}(t)x_j}. \quad (18.9)$$

Крок 8. Включення всіх усунутих на кроці 6 зразків. Повернення до кроку 2.

Основні фази функціонування сітки. Отже умовно в процесі функціонування сітки можна виокремити три основні фази: початкового стану НС; порівняння; пошуку.

Початковий стан сітки. Нульові значення компонент вхідного вектора X встановлюють сигнал нейрона K_2 керування 2 в нуль ($K_2 = 0$), одночасно встановлюючи в нуль виходи нейронів шару розпізнавання. При виникненні ненульових значень X обидва сигнали керування (K_1 і K_2) встановлюються дорівнюваними одиниці. При цьому за правилом "двох третин" виходи Π нейронів шару порівняння в точності дорівнюють компонентам X .

Вектор Π надходить на входи нейронів шару розпізнавання, які в конкурентній боротьбі визначають нейрон-переможець, описуючий гаданий результат класифікації. Врешті решт вектор P на виході шару розпізнавання містить рівно одну одиничну компоненту, решта значень дорівнюють нулю. Ненульовий вихід нейрона-переможця встановлює в нуль сигнал нейрона K_1 керування 1: $K_1 = 0$. Через зворотний зв'язок нейрон-переможець надсилає сигнали в шар порівняння, і розпочинається фаза порівняння.

Фаза порівняння. У шарі порівняння “віяло” сигналів відгуку шару розпізнавання порівнюється з компонентами вектора \mathbf{X} . Вихід Π шару порівняння тепер містить одиничні компоненти тільки в тих позиціях, в яких одиниці є й у вхідного вектора \mathbf{X} , й у вектора \mathbf{P} зворотного зв’язку. Якщо в результаті порівняння векторів Π та \mathbf{X} не буде виявлено значних відмінностей, то нейрон скидання залишається неактивованим. Вектор Π знову викличе збудження того ж самого нейрона-переможця в шарі розпізнавання, що й успішно завершить процес класифікації. В протилежному разі випрацьовується сигнал скидання, який загальмує нейрон-переможець в шарі розпізнавання, і розпочнеться *фаза пошуку*.

Фаза пошуку. В результаті дії гальмуючого сигналу скидання всі нейрони шару розпізнавання отримають нульові виходи, і, отже, нейрон *керування* 1 прийме одиничне значення активності. Знов сигнал Π на виході шару порівняння встановиться дорівнюваним в точності \mathbf{X} , як це було на початку роботи сітки. Проте тепер в конкурентній боротьбі в шарі розпізнавання попередній нейрон-переможець не приймає участі, і буде виявлена нова категорія – кандидат. Після чого знов повторюється фаза порівняння.

Ітераційний процес пошуку завершується двома можливими способами:

1. Буде знайдена запам’ятована категорія, схожість якої з вхідним вектором \mathbf{X} виявиться достатньою для успішної класифікації. Після цього відбувається навчаючий цикл, в якому модифікуються ваги $w_i^{(p)}$ та $w_i^{(n)}$ векторів $\mathbf{W}^{(p)}$ та $\mathbf{W}^{(n)}$ збудженого нейрона, здійснившого класифікацію.

2. В процесі пошуку всі запам’ятовані категорії виявляться перевіреними, але жодна з них не дала бажаної схожості. В цьому випадку образ \mathbf{X} оголошується новим для нейросітки, і йому виокремлюється новий нейрон в шарі розпізнавання. Вагові вектори цього нейрона $\mathbf{W}^{(p)}$ та $\mathbf{W}^{(n)}$ встановлюються дорівнюваними вектору \mathbf{X} .

Як бачимо, остаточний результат класифікації не виникає з першої спроби, і через це додається фаза пошуку. Адже навчання та функціонування сітки АРТ відбувається одночасно. Нейрон-переможець визначає в просторі вхідних векторів найближчий до заданого вхідного образу вектор пам’яті, і якби всі риси вхідного вектора були критичними, це й було б вірною класифікацією. Проте численні критичні риси стабілізуються лише після відносно тривалого навчання. На даній фазі навчання лише деякі компоненти вхідного вектора належать актуальній множині критичних рис, через це може знайтись інший нейрон-класифікатор, який *на множині критичних рис* виявиться ближче до вихідного образу. Він і визначається в результаті пошуку.

Необхідність пошуку в сітці АРТ визначається тим, що використовуються два критерії “схожості” векторів. Перший – максимум скалярного добутку $\max_j(W_j^{(p)}, X)$ при виборі “переможця” в шарі розпізнавання. Другий – критерій (18.4) схожості в блоці скидання.

Твердження 18.8. Задача класифікації в сітці АРТ полягає в забезпеченні необхідної та достатньої умов: знайти ядро с максимальним скалярним добутком $(W_j^{(p)}, X)$ (*необхідна умова*) і щоб при цьому виконувався критерій схожості (18.4) (*достатня умова*). Ці два критерії (умови) не є еквівалентними, тому і фаза пошуку, і фаза розпізнавання є необхідними і не можуть бути випущені.

Проте, зазначимо, що після відносної стабілізації процесу навчання класифікація виконується без фази пошуку. В цьому випадку кажуть, що *формується прямий доступ до пам'яті*. Виникнення в процесі навчання прямого доступу доводиться в теорії АРТ.

18.3. Навчання сітки АРТ

Хоча сітки АРТ не висувають вимог до порядку появи вхідних образів і не вимагають з'явлення всіх образів з однаковою частотою, при їх навчанні також використовується поняття епохи (послідовного пред'явлення кожного з навчаючих образів). Процес навчання сіток АРТ може займати багато епох.

Нейронні сітки АРТ – динамічні об'єкти, які описуються системами звичайних диференціальних рівнянь, тому їх навчання у загальному випадку є досить трудомістким. Проте моделі сіток АРТ можна спростити, якщо уявити, що змінювання сигналів на виходах нейронів відбувається набагато швидше, ніж змінення вагових векторів їх зв'язків. Тому в нейросітках теорії адаптивного резонансу можна уважати, що після виокремлення для навчання прийнятного елемента (настанні резонансу між пред'явленим і збережуваним в пам'яті образами), сигнали на виходах всіх нейронів залишаються незмінними на протязі тривалого періоду часу, за який відбуваються зміни ваг зв'язків.

В АРТ розрізняють два типи навчання: швидке і повільне. Вони відрізняються як теоретичними передпосиланнями, так і своїми характеристиками. В *методі швидкого навчання* ваги в період часу тривалого резонансу досягають рівноважного стану при кожному пред'явленні зображень. Ця форма навчання є типовою для двійкових нейросіток АРТ-1. В *методі повільного навчання* має місце суттєво більш повільне змінення ваг під час резонансу, і вони не досягають рівноважних значень при кожному пред'явленні вхідних обра-

зів. Цей метод більш властивий нейросіткам АРТ-2, хоча теоретично може застосовуватись (проте, практично не застосовується) для навчання і дискретних сіток АРТ-1. Він потребує більшої кількості пред'явлень вхідних образів, проте відносно невеликого обсягу обчислень при кожному пред'явленні.

Особливості процесу навчання сіток АРТ. На початку функціонування всі ваги $\mathbf{W}^{(p)}$ та $\mathbf{W}^{(n)}$ нейронів, а також параметр схожості ρ отримують початкові значення.

Ваги матриці $\mathbf{W}^{(n)}$. Відповідно до теорії АРТ ці значення повинні задовольняти умові:

$$w_{ij}^{(n)} = (S \pi_i) / \left(S - 1 + \sum_k \pi_k \right), \quad (18.10)$$

де k – кількість компонент вхідного вектора \mathbf{X} ; π_i – i -й компонент вектора $\mathbf{\Pi}$ на виході шару порівняння; значення $S > 1$ (наприклад, $S=2$). Такий вибір ваг буде приводити до стійкого навчання. Сума в знаменнику уявляє собою кількість одиниць на виході шару порівняння, тобто “розмір” цього вектора. У такій інтерпретації “великі” вектори $\mathbf{\Pi}$ продукують менші величини ваг $w_{ij}^{(n)}$, ніж “маленькі” вектори $\mathbf{\Pi}$. Рівень схожості ρ вибирається на основі вимог розв’язуваної задачі. При високих значеннях цього параметра буде сформована велика кількість категорій, до кожної з яких будуть відноситись тільки дуже схожі вектори. При низькому рівні ρ сітка сформує невелику кількість категорій з високим ступенем узагальнення.

Процес навчання, як вже зазначалося вище, відбувається без учителя на основі самоорганізації. Навчання виконується для ваг нейрона-переможця у випадку як успішної, так і неуспішної класифікації. Ця властивість масштабування робить можливим розділення двох векторів у випадку, коли один вектор є піднабором іншого, тобто коли набір одиничних компонент одного вектора складає підмножину одиничних компонент іншого.

Приклад 18.2. Для розуміння важливості властивості самомасштабування розглянемо приклад класифікації раніш відомих сітці векторів $\mathbf{X}_1=10000$ та $\mathbf{X}_2=11100$.

Нехай параметр $S=1$. Тоді ваги матриці зв’язків визначаються наступним чином:

$$\left\{ \begin{array}{l} w_1^{(n)1} = \frac{S \pi_1}{S - 1 + \sum_k \pi_k} = \frac{1 \cdot 1}{1 - 1 + 1} = 1 \\ w_1^{(n)2} = w_1^{(n)3} = w_1^{(n)4} = w_1^{(n)5} = \frac{S \pi_1}{S - 1 + \sum_k \pi_k} = \frac{1 \cdot 0}{1 - 1 + 1} = 0 \end{array} \right. \Rightarrow \mathbf{W}_1^{(n)} = [1 \ 0 \ 0 \ 0 \ 0],$$

$$\left\{ \begin{array}{l} w_2^{(n)1} = w_2^{(n)2} = w_2^{(n)3} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{1 \cdot 1}{1 - 1 + 3} = \frac{1}{3} \\ w_2^{(n)4} = w_2^{(n)5} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{1 \cdot 0}{1 - 1 + 3} = 0 \end{array} \right. \Rightarrow W_2^{(n)} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \end{bmatrix}.$$

Неважко бачити, що якщо масштабування нема, то при пред'явленні сітці вектора $X_2=11100$ обидва нейрони в шарі розпізнавання будуть мати однаковий рівень активації, дорівнюваний 1.

Якщо $S=2$, то

$$\left\{ \begin{array}{l} w_1^{(n)1} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{2 \cdot 1}{2 - 1 + 1} = 1 \\ w_1^{(n)2} = w_1^{(n)3} = w_1^{(n)4} = w_1^{(n)5} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{2 \cdot 0}{2 - 1 + 1} = 0 \end{array} \right. \Rightarrow W_1^{(n)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\left\{ \begin{array}{l} w_2^{(n)1} = w_2^{(n)2} = w_2^{(n)3} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{2 \cdot 1}{2 - 1 + 3} = \frac{1}{2} \\ w_2^{(n)4} = w_2^{(n)5} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{2 \cdot 0}{2 - 1 + 3} = 0 \end{array} \right. \Rightarrow W_2^{(n)} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{bmatrix},$$

тобто $W_1^{(n)}=10000$ та $W_2^{(n)} = \frac{1}{2} \frac{1}{2} \frac{1}{2} 0 0$. Тоді, подаючи на вхід сітки вектор X_1 , отримаємо збудження 1, 0 для нейрона 1 в шарі розпізнавання та 1/2 для нейрона 2; таким чином, нейрон 1 (вірний) стане переможцем. Аналогічно, пред'явлення сітці вектора X_2 збудить нейрон 1 до рівня 1, і нейрон 2 до рівня 3/2, тим самим знову чітко визначивши переможця – нейрон 2.

Тепер покажемо, що вибір значення параметра S дозволяє змінювати логіку роботи шару розпізнавання. Розглянемо випадок сітки, навченої векторами $X_1=10000$ та $X_2=11100$. Нехай на вхід сітки надходить вектор $X_3=111000$.

Якщо $S = 1$, то збудження нейрона в шарі розпізнавання P буде 1,0, а нейрона 2 становитиме 2/3. Нейрон 1 перемагає. Сітка “вважає”, що вектор $X_3=111000$ більш схожий на вектор $X_1=10000$, ніж на вектор $X_2=11100$.

Якщо $S = 1,5$, то

$$\left\{ \begin{array}{l} w_1^{(n)1} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{1,5 \cdot 1}{1,5 - 1 + 1} = 1 \\ w_1^{(n)2} = w_1^{(n)3} = w_1^{(n)4} = w_1^{(n)5} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{1,5 \cdot 0}{1,5 - 1 + 1} = 0 \end{array} \right. \Rightarrow W_1^{(n)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\left\{ \begin{array}{l} w_2^{(n)1} = w_2^{(n)2} = w_2^{(n)3} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{1,5 \cdot 1}{1,5 - 1 + 3} = \frac{3}{7} \\ w_2^{(n)4} = w_2^{(n)5} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{1,5 \cdot 0}{1,5 - 1 + 3} = 0 \end{array} \right. \Rightarrow W_2^{(n)} = \left[\frac{3}{7} \frac{3}{7} \frac{3}{7} 0 0 \right],$$

і при пред'явленні вектора $\mathbf{X}_3=11000$ збудження нейрона 1 в шарі розпізнавання \mathbf{P} буде 1,0, а нейрона 2 становитиме 6/7. Нейрон 1 перемагає. Сітка, як і раніше, “вважає”, що вектор $\mathbf{X}_3=11000$ більш схожий на вектор $\mathbf{X}_1=10000$, ніж на вектор $\mathbf{X}_2=11100$.

Якщо $S = 2$, то збудження нейрона в шарі розпізнавання \mathbf{P} буде 1,0, а нейрона 2 становитиме також 1. Сітка “вважає”, що вектор $\mathbf{X}_3=11000$ однаково схожий як на вектор $\mathbf{X}_1=10000$, так і на вектор $\mathbf{X}_2=11100$. Який нейрон переможе тепер – справа випадку, і на рішення можуть вплинути навіть незначні флуктуації. Якщо ж уявити, що будь-які флуктуації відсутні, то сітка буде знаходитись в стані “Буриданова віслюка”.

Тепер розглянемо випадок, коли $S=3$; в цьому випадку матриця ваг визначається співвідношеннями:

$$\left\{ \begin{array}{l} w_1^{(n)1} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{p \cdot 1}{3 - 1 + 1} = 1 \\ w_1^{(n)2} = w_1^{(n)3} = w_1^{(n)4} = w_1^{(n)5} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{3 \cdot 0}{3 - 1 + 1} = 0 \end{array} \right. \Rightarrow W_1^{(n)} = [1 \ 0 \ 0 \ 0 \ 0],$$

$$\left\{ \begin{array}{l} w_2^{(n)1} = w_2^{(n)2} = w_2^{(n)3} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{3 \cdot 1}{3 - 1 + 3} = \frac{3}{5} \\ w_2^{(n)4} = w_2^{(n)5} = \frac{S \pi_i}{S - 1 + \sum_k \pi_k} = \frac{3 \cdot 0}{3 - 1 + 3} = 0 \end{array} \right. \Rightarrow W_2^{(n)} = \left[\frac{3}{5} \frac{3}{5} \frac{3}{5} 0 0 \right],$$

і збудження нейрона 1 в шарі розпізнавання буде 1,0, тоді як для нейрона 2 становитиме 6/5. Нейрон 2 перемагає. Навчена у такий спосіб НС “вважає”, що вектор $\mathbf{X}_3=11000$ більш схожий на вектор $\mathbf{X}_2=11100$, ніж на вектор $\mathbf{X}_1=10000$.

Отже, на прикладі 18.2 показано, що, варіюючи значення параметра S , можна змінювати логіку роботи НС – її дискримінаційну спроможність, тобто спроможність розрізняти (або не розрізняти) близькі вектори.

Додамо також, що для правильного функціонування сітки необхідна ініціалізація ваг матриці $\mathbf{W}^{(n)}$ малими значеннями, бо при великих значеннях ініціалізації можливий варіант, коли раніше запам'ятований вхідний вектор буде активувати не “свій” нейрон, а новий і сітка буде позбавлена спроможності “згадувати”, нескінченно навчаючись і нічого не пригадуючи з минулого.

Ваги матриці $W_j^{(p)}$. Ваги матриці $W_j^{(p)}$, пов'язаної з запам'ятованим вектором, змінюються так, щоб вони дорівнювали відповідним двійковим величинам вектора порівняння Π :

$$w_{ij}^{(p)} = \pi_i \text{ для всіх } i, \quad (18.11)$$

де $w_{ij}^{(p)}$ – вага зв'язку між нейроном-переможцем j в шарі розпізнавання та нейроном i в шарі порівняння.

Нерозв'язані проблеми і недоліки АРТ-1. Нейронні сітки АРТ, при всіх їх чудових властивостях, мають ряд недоліків. Одним з них є велика кількість синаптичних зв'язків в сітці в розрахунку на одиницю запам'ятовуваної інформації. При цьому багато з ваг цих зв'язків виявляються після навчання нульовими. Цю особливість слід урахувати при апаратних реалізаціях.

Сітка АРТ-1 пристосована до роботи тільки з бітовими векторами. Ця незручність долається в сітках АРТ-2 та АРТ-3. Проте в цих архітектурах, як і в АРТ-1, зберігається основний недолік АРТ – *локалізованість пам'яті*. Пам'ять нейросітки АРТ не є розподіленою: деякій заданій категорії відповідає цілком конкретний нейрон шару розпізнавання. При його руйнуванні втрачається пам'ять про всю категорію. Ця особливість, на жаль, не дозволяє говорити про сітки АРТ як про безпосередні моделі біологічних НС. Пам'ять останніх є розподіленою.

18.4. Розширення базової моделі АРТ – нейросітки АРТ-2 та АРТ-3

Нейросітка АРТ-2. Основною відмінною рисою нейронної сітки АРТ-2 є можливість роботи з аналоговими векторами і сигналами: власне, сітка АРТ-2 призначена для класифікації як двійкових, так і неперервнозначущих векторів [90]. Хоча деталі архітектури і алгоритмів роботи для АРТ-1 та АРТ-2 відмінні, проте вони мають загальну базову архітектуру. У порівнянні з АРТ-1 в архітектурі сітки зроблені деякі зміни, які дозволяють окремим підсистемам функціонувати асинхронно, що є принциповим для апаратних реалізацій.

Важливою відмінністю аналогових сигналів від бітових є принципова можливість аналогових векторів бути скільки завгодно близькими один до одного (в той час, як простір бітових векторів дискретний). Це накладає додаткові вимоги на функціонування нейронів шару порівняння – вимагається більш “тонкий” і чутливий механізм для виокремлення областей резонансу. Загальним розв'язком тут є перехід до багат шарової архітектури з все більш точною налагодкою при переході від шару до шару, що й застосовано в АРТ-2. Функціонування шару розпізнавання принципово не змінюється.

Сітки АРТ-2 застосовувались для розпізнавання рухомих зображень. Оскільки нейросистеми АРТ не містять механізму інваріантного розпізнавання (на відміну від *неокогнітрона*, див. розд. 13.3), то в сполученні з ними використовуються спеціалізовані (часто не нейросіткові) системи інваріантного подання образів, наприклад, двовимірне перетворення Фур'є або більш складні алгоритми. Більш детальний розгляд особливостей і застосувань АРТ-2 вимагає професійного вивчення і не входить в задачі курсу.

Нейросітка АРТ-3. Наступним кроком у розвитку АРТ виявилась сітка АРТ-3 [93]. Особливості навчання нейронів сіток АРТ-1 та АРТ-2 не дозволяють використовувати ці сітки як *елементи більш великих ієрархічних нейросистем*, зокрема, компонувати з них багатопшарові сітки. Це утруднює уявлення в АРТ ієрархічно організованої інформації, що є характерним для систем сприйняття людини та тварин.

Ці проблеми розв'язані у сітці АРТ-3, яка виступає як багатопшарова архітектура. При переході від шару до шару відбувається контрастування входних образів і запам'ятовування їх у вигляді все більш загальних категорій. При цьому основною задачею кожного окремого шару є стискання входної інформації.

Образ входить в адаптуючий резонанс між деякою парою шарів, в подальшому цей резонанс розповсюджується на наступні шари ієрархії. В АРТ-1 та АРТ-2 недостатній рівень резонансу приводив до генерації сигналу скидання, що викликало повне гальмування шару розпізнавання. У випадку багатопшарової сітки АРТ-3 це є неприпустимим, бо розриває потік інформації. Через це в АРТ-3 уведений спеціальний механізм залежності активності синапсів зворотних зв'язків від часу, аналогічний рефрактерному гальмуванню біологічного нейрона після передавання збудження. Тому замість повного скидання сигналу відбувається гальмування синаптичних сигналів зворотного зв'язку, і шар порівняння отримує початковий стан збудження для виконання фази пошуку нового резонансу.

Цікавою пропозицією є також використання у багатопшаровій ієрархії шарів, які не є шарами АРТ, а належать деякій іншій архітектурі. В такому випадку *система стає гібридною*, що може привести до виникнення нових корисних властивостей.

18.5. Нейронна сітка АРТ МАП

Група моделей АРТ-сіток з суфіксом “МАП” (ARTMAP та ін.) класифікує і входні вектори, і вектори на виході, а також будує зв'язки між ними, дозволяючи формувати відображення аналогічно до сітки зустрічного поширення (див. розд. 8). Дана НС уявляє собою комбінацію двох сіток АРТ, якими можуть бути АРТ-1 або АРТ-2. Схематичне уявлення АРТ МАП наведено на рис. 18.4.

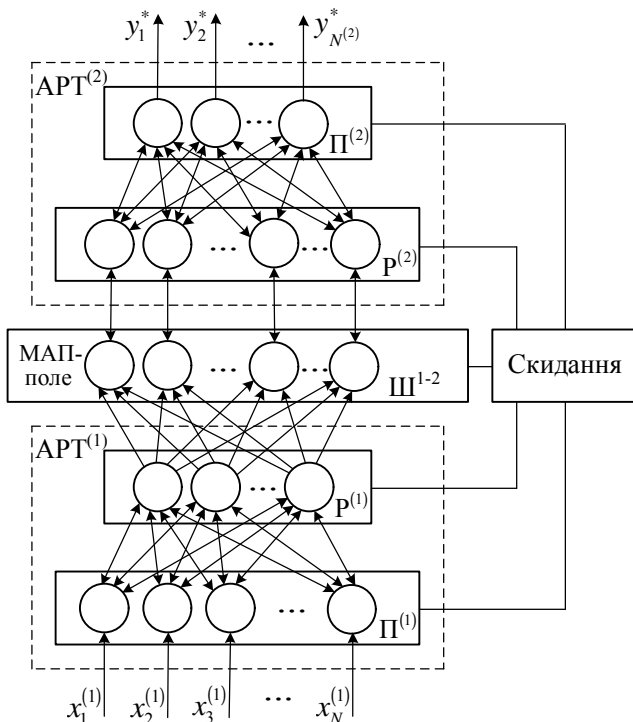


Рис. 18.4. Архітектура сітки АРТ МАП

Архітектурно сітка складається з двох АРТ – АРТ⁽¹⁾ і АРТ⁽²⁾, об'єднаних шаром Ш¹⁻² – МАП-полем (асоціативною пам'яттю). Як і будь-яка сітка АРТ, кожна з сіток АРТ⁽¹⁾ і АРТ⁽²⁾ має відповідні шари розпізнавання P⁽¹⁾ і P⁽²⁾ та порівняння Π⁽¹⁾ і Π⁽²⁾. Вектор характеризуючих z-й образ вхідних сигналів $\mathbf{X}_z^{(1)} = (x_{1z}^{(1)}, x_{2z}^{(1)}, \dots, x_{Nz}^{(1)})^T$ розмірності $N^{(1)} \times 1$ надходить на АРТ⁽¹⁾, а на вхід сітки АРТ⁽²⁾ подається асоційований з $\mathbf{X}_z^{(1)}$ сигнал виходу $\mathbf{X}_z^{(2)} = (x_{1z}^{(2)}, x_{2z}^{(2)}, \dots, x_{Nz}^{(2)})^T$. Шари P і Π обох сіток зв'язані між собою, як звичайно, за допомогою ваг $w_{ij}^{(1n)}$ та $w_{ij}^{(1p)}$, $w_{ij}^{(2n)}$ та $w_{ij}^{(2p)}$. Зв'язки ж з МАП-полем організовані по-різному. Якщо всі нейрони шару P⁽¹⁾ зв'язані з усіма нейронами МАП-поля, і сигнали розповсюджуються тільки в одному напрямку – від нейронів шару P⁽¹⁾ до нейронів МАП-поля, то кожний нейрон шару P⁽²⁾ зв'язаний тільки з одним нейроном МАП-поля, і сигнали можуть передаватися в обох напрямках. Тому кількість нейронів шару P⁽²⁾ та МАП-поля співпадає.

Класифікація пред'явлених образів здійснюється в МАП-полі. Навчання з вчителем і застосування спеціальних методів кодування вхідних векторів підвищує ефективність розв'язання задач класифікації.

18.6. Суттєві недоліки сітки АРТ

У сітки АРТ є декілька суттєвих недоліків:

1. *Чутливість до порядку пред'явлення векторів.* Більшість різновидів АРТ досить чутливі до порядку *пред'явлення вхідних векторів* X . Вигляд ядер класів, сформованих сіткою, принципово змінюються при різних видах упорядкування.

2. *Неможливість класифікації зашумлених векторів.* Нехай вхідні вектори містять шум, якщо компонента незашумленого вхідного вектора дорівнює x_n , то пред'явлені сітці значення будуть визначатися ймовірнісним законом:

$$\begin{cases} p(X_n = x_n) = 1 - \epsilon \\ p(X_n = \neg x_n) = \epsilon, \end{cases} \quad (18.12)$$

де ϵ — мале додатне число, яке характеризує рівень шуму.

Якщо такі дані будуть пред'явлені АРТ, то буде спостерігатися деградація і розмноження класів. Якщо сіткою сформоване правильне ядро для класу, до якого відноситься вектор X , то як тільки компонента X_n прийме нульове значення за рахунок шуму (якщо вектори пред'являються неодноразово), відповідна компонента ядра також буде встановлена на нульову позначку. Через те, що випадкове нульове значення може прийняти будь-яка компонента X , то з часом всі компоненти ядра будуть встановлені на нульову позначку, запам'ятована інформація про цей клас — втрачена. Якщо після цього пред'явити незашумлений варіант вектора X , то для нього буде виокремлений новий нейрон, тобто сформований новий клас. Це явище називається розмноженням класів. Через деякий час в сітці буде множина нейронів з нульовими вагами, і всі нейрони будуть розподілені. Робота сітки припиниться. Це явище визначається вихідною асиметрією алгоритмів АРТ відносно значень 0 і 1.

Існують методи для усунення асиметрії та запобігання розмноження класів, наприклад, комплементарне кодування.

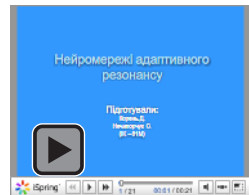
Контрольні завдання

1. Охарактеризувати дилему стабільності-пластичності пам'яті в штучних нейронних сітках.

2. Розкрити принцип адаптивного резонансу – внутрішнього “детектора новизни” як першої важливої умови адаптивної резонансної теорії.
3. Розтлумачити другу суттєву умову адаптивної резонансної теорії – “специфічності критичних рис”.
4. Розкрити характер третьої значної умови адаптивної резонансної теорії – необхідності самоадаптації алгоритму пошуку образів в пам’яті.
5. Прокоментувати четверту умову адаптивної резонансної теорії – правило “два з трьох”.
6. Показати, як виконується п’ята умова адаптивної резонансної теорії – “зустрічне очікування” щодо реалізації механізму прийняття рішень.
7. Пояснити, в чому полягає парадигма нейронних сіток адаптивної резонансної теорії у використанні функції неконтрольованого навчання.
8. Охарактеризувати узагальнену архітектуру нейросіток адаптивної резонансної теорії.
9. Прокоментувати основні теореми адаптивної резонансної теорії про правильність класифікації, а також стійкість і скінченність процесів навчання та пошуку.
10. Розкрити особливості функціонування нейросіток адаптивної резонансної теорії: ємність сітки; функції шарів порівняння та розпізнавання; формування сигналів.
11. Прокоментувати алгоритм функціонування нейросіток адаптивної резонансної теорії.
12. Розкрити фази функціонування нейронних сіток адаптивної резонансної теорії: початкового стану, порівняння, пошуку.
13. Проаналізувати необхідну і достатню умови задачі класифікації в сітках адаптивної резонансної теорії.
14. Охарактеризувати особливості процесу навчання нейросіток адаптивної резонансної теорії.
15. Навести і охарактеризувати основні розширення базової нейронної сітки адаптивної резонансної теорії – нейронні сітки ART2 та ART3.
16. Охарактеризувати групу моделей сіток АРТ МАП: навести архітектуру і розкрити принцип функціонування.
17. Прокоментувати суттєві недоліки нейронних сіток адаптивної резонансної теорії.

Презентації

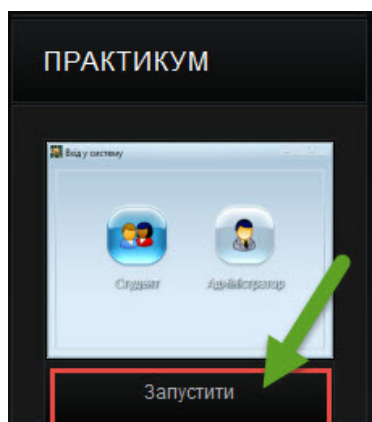
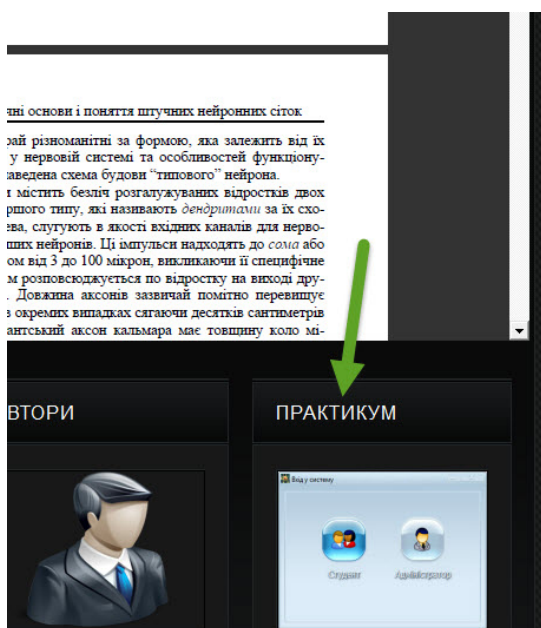
1. Нейромережі адаптивного резонансу



Практичні заняття

Динамічні рекурсивні нейронні сітки

Для запуску практичних занять на
вкладці "Практикум" натисніть
"Запустити"



Розділ 19

Осциляторні нейронні сітки

*Я ніколи не стою, якщо можна сидіти,
і я ніколи не сиджу, якщо можна лежати*
Уїнстон Черчилль

*Сила слабкої статі в слабкості сильної статі
перед слабкою статтю*
Афоризм

19.1. Нейрофізіологічна інтерпретація осциляції

На відміну від інших парадигм, при вивченні *осциляторних нейронних сіток* (ОНС) основний інтерес зосереджений на динамічних, коливальних аспектах функціонування НС.

Означення 19.1. *Осциляторні нейронні сітки* (ОНС) – нейронні сітки, основними структурними одиницями яких є осцилятори. Функціонують ОНС за рахунок коливань окремих елементів або груп елементів та їх взаємодії. Осциляторні НС уявляють науковий інтерес через те, що суттєву роль у розумових процесах людини відіграють коливання.

Відповідно до цього вибирається така конструкція окремого елемента і така архітектура сітки, при яких спостерігаються регулярні, квазіперіодичні або стохастичні коливання. При цьому представляють інтерес умови виникнення та синхронізації коливань.

Вивчення ОНС стимулюється результатами нейрофізіологічних експериментів, які вказують на істотну, а можливо, і центральну роль коливальних процесів в роботі нервової системи. Одна з центральних гіпотез полягає в тому, що процес обробки інформації в нервовій системі може описуватися в термінах синхронізації активності різних нейронних структур. Ця гіпотеза була сформульована в роботах відомих російських нейрофізіологів: Ухтомського А.А. [60; 61] і Ліванова М.Н. [33].

В останні роки інтерес до осциляційної нейронної активності знову посилюється завдяки нейрофізіологічним дослідженням У. Фрімена, В. Зінгера, Р. Екхорна та інших авторів [101; 111; 120]. Як показано в цих роботах, в нюховій та зоровій зонах кори можуть виникати високочастотні коливання (40-60 герц) у відповідь на зовнішню стимуляцію, причому за певних умов спостерігається синхронізація коливань в далеких одна від одної ділянках кори.

Вважається, що за допомогою синхронної коливальної активності в мозку підсвідомо на ранніх стадіях обробки інформації відбувається виділення об'єкта з фону (під час так званої обробки “перед увагою”).

У дослідження ОНС як моделей нервової системи в даний час залучені кілька наукових груп у Росії, США, ФРН, Великобританії, Італії та Японії. Робота ведеться в двох взаємопов'язаних напрямках:

- математичне дослідження явищ синхронізації в ОНС (аналітичні і комп'ютерні методи);
- застосування ОНС для побудови моделей нюхової, зорової кори і рухової системи, а також пам'яті та уваги (імітаційне моделювання).

Функціональною одиницею ОНС, як правило, є осцилятор. залежно від цілей дослідження вибирається не тільки певна архітектура зв'язків сітки, але і конструкція окремого осцилятора. У найбільш складних і максимально наближених до реальності моделях використовуються осцилятори, які складаються з елементів з інтегративно-пороговими властивостями різного ступеня складності. Це – аналоги біологічних нейронів. Осцилятори включають нейрони як з гальмівними, так і з збуджуючими зв'язками. Це призводить до того, що за певних умов сумарна активність нейронів набуває коливальний характер. *Такі ОНС досліджуються зазвичай методами імітаційного моделювання.*

Інший підхід полягає в описі динаміки усередненої активності нейронних ансамблів, які в результаті взаємодії утворюють осцилятор. У ряді робіт вихідні припущення призводять до ОНС з осциляторів Ван дер Поля. В інших осцилятори формуються в результаті взаємодії невеликої кількості нейронних ансамблів, з'єднаних збуджуючими і гальмівними зв'язками. Виникаючі на цьому шляху багатовимірні динамічні системи все ще, як правило, занадто складні для математичного аналізу, тому їх досліджують чисельно або редукують до ще більш простих осциляторів, поведінка яких описується однією змінною - фазою коливання. Іноді сітки з таких максимально спрощених осциляторів виводяться аксіоматично. Такі ОНС придатні для надто наближеного, якісного опису реальних процесів. Їх головна перевага – можливість аналітичного дослідження. Надалі використовуватиметься термін “*нейронний осцилятор*” в тому випадку, коли осцилятор утворюється в результаті взаємодії елементів за допомогою збуджуючих і гальмівних зв'язків. Осцилятори, описувані однією змінною фазою, ми будемо називати *фазовими осциляторами*.

Більш точне визначення нейронного і фазового осциляторів буде дано в наступному розділі.

19.2. Математичне дослідження динаміки осциляторних нейросіток

З точки зору математичного моделювання зручно представляти ОНС у вигляді окремих, взаємодіючих між собою осциляторів.

Означення 19.2. *Осцилятори* – це безліч спільно функціонуючих елементів (нейронів або нейронних ансамблів), здатних працювати в коливальному режимі.

Означення 19.3. *Нейронний осцилятор* – такий осцилятор, в структурі якого є збуджуючі і гальмівні нейрони, що відрізняються характером впливу на інші елементи нейронної сітки: збуджуючі нейрони збільшують, а гальмівні зменшують їх.

Осцилятори описується системою диференціальних (або різницевих) рівнянь, іноді з випадковим шумом. Таких рівнянь може бути багато (кілька десятків або сотень) у разі детального урахування специфіки біологічних нейронів [207]. Якщо ж вивчення проводиться на рівні нейронних популяцій, то розглядаються зазвичай два – п'ять рівнянь, що описують усереднену за ансамблем динаміку кожної популяції [110; 208]. В випадку фазового осцилятора розглядається лише одна змінна – фаза коливань [105].

Залежно від архітектури зв'язків між осциляторами далі розглядатимуться ОНС двох типів.

1. Повнозв'язні осциляторні НС. У цьому випадку кожен з осциляторів пов'язаний з усіма іншими осциляторами.

2. Осциляторні НС з локальними зв'язками. У цьому випадку кожний осцилятор пов'язаний тільки з осциляторами зі свого околу фіксованого радіуса. Іноді враховуються тимчасові затримки в зв'язках.

У більшості розглянутих робіт передбачається, що величина взаємодії (сила зв'язку) осциляторів мала. Тоді можна використовувати *асимптотичні методи*: той чи інший варіант теорії усереднення або ж перехід до безперервної апроксимації. Інша можливість – розгляд нейронних мереж, що складаються з дуже великого числа осциляторів. Тут також використовуються *асимптотичні методи*, а також *метод перенормування* (укрупнення осциляторів). Якщо не вдається отримати аналітичні результати, то дослідження проводиться за допомогою імітаційного моделювання.

Динаміку осцилятора зручно представляти як рух уздовж траєкторії у фазовому просторі. Тоді регулярним коливанням буде від-

повідати граничний цикл, квазіперіодичним - тор, стохастичним – дивний (стохастичний) атрактор. При зміні параметрів мережі можуть відбуватися біфуркації (фазові переходи), в результаті яких з'являються і зникають атрактори системи. Перш ніж перейти до вивчення динаміки ОНС, покажемо, що ОНС загального виду зі слабкими зв'язками редукується до сітки з фазових осциляторів.

Нехай осцилятор описується системою Т-автономних диференціальних рівнянь:

$$\frac{dx}{dt} = F(x), x \in \mathbb{R}^m. \quad (19.1)$$

Припустимо, що у фазовому просторі осцилятора існує асимптотно стійкий граничний цикл з періодом $2\pi/\omega$. Тоді в малому околі циклу можна вибрати нові координати (θ, y) , де θ – фаза руху по циклу та $y \in \mathbb{R}^{m-1}$ – координати нормального перетину (на граничному циклі $y=0$). У нових координатах рівняння (19.1) приймають вид:

$$\frac{d\theta}{dt} = \omega; \frac{dy}{dt} = \alpha(\theta)y + O(\|y\|^2). \quad (19.2)$$

Припустимо тепер, що осцилятори в ОНС з'єднані слабкими зв'язками порядку ε :

$$\frac{dx_k}{dt} = F_k(x_k) + \varepsilon D_k(x_1, x_2, \dots, x_N), x_k \in \mathbb{R}^m, k=1, 2, \dots, N. \quad (19.3)$$

У разі парних і адитивних зв'язків між осциляторами рівняння (19.3) приймають вид:

$$\frac{dx_k}{dt} = F_k(x_k) + \varepsilon \sum_j D_{kj}(x_k, x_j), x_k \in \mathbb{R}^m, k=1, 2, \dots, N. \quad (19.4)$$

При $\varepsilon=0$ пряме утворення граничних циклів дає інваріантний N -вимірний тор (початкова фаза θ_0 на кожному граничному циклі може вибиратися довільно). Можна показати, що цей інваріантний тор збережеться і при малих $\varepsilon > 0$ (чим сильніше тяжіння до граничних циклів окремих осциляторів, тим більші величини зв'язку є припустимими). У цьому випадку рівняння для фази k -го осцилятора ОНС на інваріантному торі можуть бути записані у вигляді:

$$\frac{d\theta_k}{dt} = \omega_k \varepsilon \sum_j D_{kj}(\theta_k, \theta_j), k=1, 2, \dots, N, \quad (19.5)$$

де 2π – періодичні функції D_{kj} виходять в результаті застосування методу усереднення і залежать від типу осцилятора (тобто вектор-функції $F_k(x)$), а також від архітектури і типу зв'язків (тобто вектор-функції $D_{kj}(x)$). Такий висновок рівнянь (19.5) міститься, наприклад,

в роботі [105]. В ряді випадків (наприклад, для ланцюжка осциляторів) функції D_{kj} залежать лише від різниці фаз $\phi_k = \theta_j - \theta_k$, і тоді замість рівнянь для фаз (19.5) можна отримати рівняння для різниць фаз ϕ .

У роботах, де осцилятор описується рівнянням динаміки фази типу (19.5), режим синхронізації ОНС визначається як режим, при якому кожний з осциляторів стабілізує свою частоту при великих t :

$$\frac{d\theta_k}{dt} \xrightarrow{t \rightarrow \infty} \Psi_k. \quad (19.6)$$

Це визначення синхронізації використовується в наступних трьох модифікаціях.

Осциляторна синхронізація ОС-1. Всі осцилятори працюють на одній частоті, $\Psi_k = \Psi$. У цьому випадку іноді буває зручно розглядати не самі фази, а їх різниці, наприклад $\phi_k = \theta_k - \theta_0$ (у випадку особливого становища в ОНС деякого осцилятора з фазою θ_0) або $\phi_k = \theta_{k+1} - \theta_k$ (для ланцюжка осциляторів). Тоді умова синхронізації приймає вид:

$$\frac{d\phi_k}{dt} \xrightarrow{t \rightarrow \infty} 0. \quad (19.7)$$

Осциляторна синхронізація ОС-2. Осцилятори розділяються на групи з рівними або близькими частотами, утворюючи тим самим кластери синхронно-коливальних осциляторів. Умова (19.6) виконується лише наближено при малих зв'язках і великій кількості осциляторів в ОНС. Частина осциляторів взагалі може не входити в синхронізацію, для них (19.6) виконуватися не буде.

Осциляторна синхронізація ОС-3. Для частот Ψ_k справедливий відношення

$$\Psi_1 : \Psi_2 : \dots : \Psi_N = n_1 : n_2 : \dots : n_N, \quad (19.8)$$

де $n_1 : n_2 : \dots : n_N$ — цілі числа.

Основне завдання полягає в тому, щоб визначити критичні значення параметрів, при яких той чи інший тип синхронізації має місце. У цьому випадку прийнято говорити про *фазовий перехід* і про виникнення *колективної поведінки* в ОНС. При синхронізації другого типу становить інтерес задача визначення кількісних характеристик кластерів (середнього і дисперсії розміру кластера, швидкості його переміщення по сітці тощо). Також значний інтерес представляє функція розподілу фаз коливань при різних режимах роботи ОНС.

З подальшого викладу буде ясно, яке з визначень синхронізації (ОС-1 – ОС-3) використовується в тій або іншій ОНС.

19.3. Повнозв'язні осциляторні нейронні сітки

Розглянемо спочатку ОНС, що складаються з осциляторів Ван дер Поля [176]. Передбачається, що ці осцилятори слабо нелінійні (майже гармонічні), їх динаміка описується рівняннями:

$$\begin{aligned}\frac{dx_i}{dt} &= y_i, \\ \frac{dy_i}{dt} &= -\omega_i^2 x_i + \varepsilon(1 - \beta x_i^2) + \varepsilon \frac{D}{N} \sum_{j=1}^N (y_j - y_i) + \lambda_i,\end{aligned}\tag{19.9}$$

де λ – білий шум.

У роботі Е. Ямагучі і Х. Шімізу [210] була розроблена теорія самосинхронізації для системи N осциляторів Ван дер Поля (19.9) з розподіленими частотами і зовнішніми шумами. Питання про синхронізацію розглядалося в сенсі ОС-2 у термінах макрохарактеристик (параметр порядку) системи

$$X = \frac{1}{N} \sum_{i=1}^N x_i.$$

З її використанням були знайдені умови, за яких в системі відбувається синхронізація коливань з частотою, що дорівнює усередненій. Для цього виведено наближене рівняння динаміки параметра порядку:

$$\frac{dX}{dt} = e_1 X - e_2 X^3.$$

Якщо e_1 – негативна величина, то стійким є стаціонарний стан $X=0$, і в системі не з'являється синхронної колективної поведінки. Якщо $e_1 > 0$, то стійкими стаціонарними станами будуть $X_{ст} = \pm(e_1/e_2)^{1/2}$.

Ці умови детально проаналізовані в термінах параметрів осциляторів для різних розподілів власних частот. Виявилось, що в цих термінах існують два типи біфуркацій: *механічна біфуркація*, залежна від ширини розподілу власних частот осциляторів в присутності зовнішніх шумів; *теплова біфуркація*, що залежить від дисперсії шуму при рівності власних частот. Чисельні експерименти показують, що має місце *біфуркація типу Ландау*, що об'єднує як механічну, так і теплову біфуркації.

В роботі [160] досліджувалася динаміка сіток фазових осциляторів з синусоїдальною взаємодією:

$$\frac{d\theta_i}{dt} = \omega_i + \frac{L}{N} \sum_{j=1}^N \sin(\theta_j - \theta_i), \quad i = 1, 2, \dots, N, \quad (19.10)$$

де θ_i – поточна фаза i -го осцилятора; ω_i – його власна частота; L – константа зв'язку ($L > 0$).

Якщо всі частоти в рівняннях (19.10) однакові ($\omega_i = \omega_j$), то ця система відома у фізиці як класична ХВ - спінова модель з нульовою температурою.

Нас цікавитиме випадок, коли.

Якщо власні частоти ω_x різні, і їх розкид обмежений і досить малий, то має місце синхронізація всіх осциляторів мережі в сенсі ОС-1. Зручна форма опису поведінки фаз виходить при розташуванні осциляторів на відрізку $[0, 1]$ і перехід до межі при $N \rightarrow \infty$. У цьому випадку (19.10) приймає вид:

$$\frac{d\theta(x, t)}{dt} = \omega(x) + \alpha \int_0^1 \sin(\theta(x', t) - \theta(x, t)) dx'. \quad (19.11)$$

Як показано в роботі [103], після встановлення режиму синхронізації фази приймають вид:

$$\theta(x, t) = \bar{\omega}t + \varphi(x), \quad (19.12)$$

що відповідає синхронізації на середній частоті $\bar{\omega}$. Співвідношення фаз різних осциляторів ОНС задається функцією $\varphi(x)$.

Тепер розглянемо сітку фазових осциляторів типу (19.5) у більш загальному виді, не припускаючи парності взаємодії:

$$\frac{d\theta_k}{dt} = \omega_k + \varepsilon D_k(\theta_1, \theta_2, \dots, \theta_N). \quad (19.13)$$

Виберемо ортогональний базис в просторі $R^N : \omega, S_2, \dots, S_N$ і введемо нові змінні

$$\begin{aligned} v, u_2, \dots, u_N : \\ v = \omega\theta, u_j = S_j\theta, j = 2, \dots, N, \\ (S_j\omega = 0, S_jS_k = 0 \text{ для } j \neq k). \end{aligned} \quad (19.14)$$

Рівняння для фаз θ_k в нових змінних будуть мати вигляд:

$$\begin{aligned} \frac{dv_k}{dt} &= \omega^2 + \varepsilon \omega D(v, u_2, \dots, u_N), \\ \frac{du_k}{dt} &= \varepsilon S_j D(v, u_2, \dots, u_N), \quad j = 2, \dots, N. \end{aligned} \quad (19.15)$$

Таким чином, малий параметр ε дозволяє перейти до однієї швидкої змінної $v(v : \omega^2 t)$ і повільної змінної u , що необхідно для

застосування методу усереднення та подальшого аналізу системи. Після усереднення за v маємо:

$$D^*(u) = \lim_{Y \rightarrow \infty} (1/Y) \int_0^Y D(v, u_2, \dots, u_N) dv \quad (19.16)$$

рівняння (19.16) перетворюються до виду:

$$\frac{du_j}{dt} = \epsilon S_j D_j^*(u_2, \dots, u_N), \quad j = 2, \dots, N. \quad (19.17)$$

Доведено, що якщо існує асимптотично стійкий стаціонарний стан усередненої системи ($u \rightarrow u^*$ при $t \rightarrow \infty$), то початкова система буде синхронізуватися в сенсі ОС-3.

19.4. Осциляторні нейросітки з локальними зв'язками

При розгляді ОНС з локальними зв'язками передбачається, що осцилятори розташовані у вузлах d -вимірної цілочисельної решітки ($d = 1, 2, 3, \dots$), і кожний осцилятор має зв'язки тільки з осциляторами зі свого околу фіксованого радіуса. Зазвичай це зв'язки з $2d$ найближчими сусідами. При $d = 1$ мережа називається ланцюжком осциляторів. Іноді розглядається замикання ланцюжка в кільце.

Роботи, присвячені ОНС з локальними зв'язками, можна розділити на дві групи. У першій з них вивчаються *сітки зі слабкими зв'язками*, а осцилятори утворюють ланцюжок або кільце. В інших роботах вивчаються *сітки з сильними зв'язками*, а розмірність сітки може бути довільною. І в тому, і в іншому випадках дослідження зводиться до сіток фазових осциляторів.

Фазові осцилятори зі слабкими зв'язками. В роботах [155; 156] розглядаються ланцюжки фазових осциляторів зі слабкими зв'язками. При малих коефіцієнтах зв'язку ϵ фазові осцилятори входять з класу осциляторів загального виду

$$\frac{dx_k}{dt} = F(x_k) \epsilon [D^+(x_{k+1}, x_k) + D^-(x_{k-1}, x_k)]; \quad D^+(x, x) = 0 = D^-(x, x)$$

за допомогою техніки, описаної на початку цього розділу.

Рівняння на інваріантному торі будуть мати вигляд:

$$\theta_1 = \omega_1 + \epsilon H^+(\phi_1) + o(\epsilon^2);$$

$$\frac{d\phi_k}{dt} = \epsilon [\Delta_k + H^+(\phi_{k+1}) + H^+(-\phi_k) - H^-(\phi_k) - H^-(-\phi_{k-1})] + \quad (19.18)$$

$$+ o(\epsilon^2); \quad H^-(-\phi_1) = H^+(\phi_N),$$

де $\phi_k = \theta_{k+1} - \theta_k$; $\varepsilon \Delta_k = \omega_{k+1} - \omega_k$ (різниці власних частот різних осциляторів передбачаються малими); H^+ (відповідно H)- 2π – періодичні функції, що залежать від F і D^+ (відповідно D) і обчислюються за допомогою техніки усереднення.

Система (19.18) вважається синхронізованою, якщо всі осцилятори мають однакові частоти і фіксовані в часі різниці фаз ϕ_k :

$$\begin{aligned}\frac{d\theta_k}{dt} &= \Omega, \\ \theta_k &= \Omega t + \theta_k^0, \\ \phi_k &= \theta_{k+1} - \theta_k = \theta_{k+1}^0 - \theta_k^0.\end{aligned}\tag{19.19}$$

Тому питання про синхронізацію зводиться до дослідження стаціонарних рішень рівняння (19.18):

$$\Delta_k + H^+(\phi_{k+1}) + H^+(-\phi_k) - H^-(\phi_k) - H^-(-\phi_{k-1}) = 0. \tag{19.20}$$

При лінійному загасання власної частоти осциляторів уздовж ланцюжка при невеликому градієнті частоти існує розв'язок рівнянь (19.20), тобто можлива синхронізація системи [104]. Якщо ж градієнт стає досить великим, то синхронізація неможлива. При цьому в системі (19.18) з'являється граничний цикл великої амплітуди (порядку 1), який відповідає наявності в системі двох “частотних осередків”, іншими словами, ланцюжок розпадається на дві підланки, такі, що осцилятори в лівій підланці мають одну і ту ж частоту і осцилятори в правому підланцюгу також мають одну і ту ж, але вже іншу частоту.

Фазові осцилятори з сильними локальними зв'язками. У разі синусоїдальної взаємодії рівняння динаміки мають вид:

$$\frac{d\phi_i}{dt} = \omega_i + L \sum_{j \in N_i}^N \sin(\phi_j - \phi_i), \quad i = 1, 2, \dots, N, \tag{19.21}$$

де N_i – безліч сусідів i -го осцилятора з його околиці кінцевого розміру. Нехай $q(\omega)$ – щільність розподілу власних частот. Основна розв'язувана задача – з'ясувати, чи існує таке L , що кількість осциляторів, функціонуючих синхронно в сенсі ОС-1, буде мати порядок N при $N \rightarrow \infty$. Відповідь на це питання залежить від розмірності d решітки, на якій розташована сітка, і “хвостів” функції q . Чим більше d і чим вужчі “хвости” q , тим скоріше можна очікувати, що досить велика кількість осциляторів в мережі буде працювати синхронно.

У загальному випадку, коли взаємодія осциляторів регулюється досить довільною функцією, а розподіл q не обов'язково нормальний, отримати нижню оцінку для критичної розмірності істотно

важче і можливо при використанні методу перенормування. Це зроблено в роботі [100], в якій використовується гіперкуб, що містить нейросітку, який розбивається на підкуби. Кожному підкубу ставиться у відповідність новий осцилятор, фаза якого дорівнює середній фазі осциляторів, розташованих у цьому підкубі. Після деякого перенормування часу і фаз нових осциляторів рівняння для цих фаз стають майже такими ж, як для вихідних осциляторів. Важлива відмінність полягає в тому, що тепер параметр взаємодії дорівнюватиме:

$$J_{\alpha} = N^{\delta}, \quad (19.22)$$

де δ залежить від розмірності d і параметра α , що характеризує швидкість спадання хвоста щільності q ($\alpha = 2$ для нормального розподілу і $0 < \alpha < 2$, якщо дисперсія власних частот осциляторів нескінченна):

$$\delta = 1 - 1/\alpha - 1/d. \quad (19.23)$$

Нескладний аналіз показує, що синхронізація нових осциляторів може мати місце лише за $\delta \geq 0$. Звідси випливає, що

$$d \geq \alpha/(\alpha - 1), \text{ де } 1 < \alpha \leq 2, \quad (19.24)$$

а при $\alpha \leq 1$ синхронізація неможлива ні за якої розмірності d . Зауважимо, що при $\alpha = 2$ оцінка $d \geq 2$ все ж гірша, ніж $d > 2$. Інформація про будову кластерів синхронізованих осциляторів міститься в роботі [194]. Виявляється, що якщо власні частоти осциляторів розподілені нормально, то ці кластери мають дуже пухку, губчасту будову: ймовірність того, що такий кластер містить куб розміру порядку N , при $N \rightarrow \infty$ експоненціально убуває до 0.

19.5. Осциляторні нейросітки з затримками у зв'язках

Введення тимчасової затримки в передання сигналів по зв'язках вносить суттєві ускладнення в поведінку локально пов'язаних осциляторів. Розглянемо найпростішу систему двох фазових осциляторів із затримками:

$$\begin{aligned} \frac{d\theta_1(t)}{dt} &= \omega_1 + L \sin(\theta_1(t) - \theta_2(t - \tau)), \\ \frac{d\theta_2(t)}{dt} &= \omega_2 - L \sin(\theta_2(t) - \theta_1(t - \tau)). \end{aligned} \quad (19.25)$$

Нехай Ω – частота, на якій синхронізується осцилятори ($d\theta_1(t)/dt = d\theta_2(t)/dt = \Omega$), а $\phi = \theta_1(t) - \theta_2(t)$ – відповідна різниця фаз. Як показано в роботі [191], введення затримки τ призводить до

наступних важливих результатів: для даної пари (L, τ) існує кілька розв'язків $\Omega_i(L, \tau)$, $\phi_i(L, \tau)$ з різними розмірами областей тяжіння і різною стійкістю (в той же час, очевидно, що за відсутності затримки існує єдиний розв'язок); на площині (L, τ) існують критичні лінії, при перетині яких стійкі розв'язки Ω, ϕ різко змінюють свої значення.

У роботі [175] на двовимірній решітці досліджувалася синхронізація в сенсі ОС-1 великої мережі фазових осциляторів, пов'язаних з найближчими сусідами. Система рівнянь для такої мережі є узагальненням (19.21), включаючи затримку τ в переданні сигналів від сусідів і білий шум $\lambda_i(t)$ з дисперсією T^2 (T – аналог температури). Розглядався випадок $\omega_i = \omega_0$.

Розрахунки показали, що із зростанням величини затримки τ середня частота

$$\Omega = \frac{1}{N} \sum_i \left\langle \frac{d\theta_i}{dt} \right\rangle$$

зменшується ($\langle \bullet \rangle$ – середнє за імовірнісним простором). Крім того, динаміка системи визначається значеннями температури T . Для високих температур зростає просторова неоднорідність у розподілі фаз, і система досить швидко досягає мінімальної з можливих частот синхронізації. Для низьких температур просторова неоднорідність малоімовірна, і система може довго перебувати в метастабільних станах перш, ніж перейде в стан з мінімальною частотою.

19.6. Висновки

Математичне дослідження динаміки ОНС. Математична теорія ОНС розроблена в основному для систем, в яких є малий параметр: слабкий зв'язок між осциляторами, мала величина $1/W$ при необмеженому зростанні кількості осциляторів N , мала амплітуда майже гармонійних коливань поблизу біфуркації тощо. У цих випадках асимптотичні методи дозволяють схарактеризувати динамічну поведінку ОНС, зокрема, сформулювати умови синхронізації коливань. Сітки з двох або трьох пов'язаних осциляторів досить добре досліджуються методами теорії біфуркацій. Труднощі виникають, коли кількість осциляторів становить сотні або тисячі, і зв'язки між ними є суттєвими. Зазвичай такі системи вивчаються методами імітаційного моделювання на ЕОМ.

Подальший розвиток теорії ОНС пов'язують з розв'язанням наступних питань:

1. Дослідженням динаміки і синхронізації ОНС для осциляторів з локальними зв'язками на площині і в тривимірному просторі у разі багат шарових архітектур, в ієрархічних НС і з центральним осцилятором, при наявності тимчасових затримок у зв'язках тощо.

2. Вивченням режимів, в яких можливе утворення динамічних кластерів синхронно працюючих осциляторів і дослідженням властивостей цих кластерів.

3. Дослідженням динаміки і синхронізації в ОНС із змінними параметрами навчання НС, постановкою і розв'язанням оптимізаційних задач.

Осциляційні моделі нюхової системи. У більшості математичних моделей нюхової системи розглядається процес розпізнавання запахів і їх запам'ятовування. Моделі включають популяції збуджуючих і гальмівних елементів, характерними динамічними режимами таких мереж є регулярні та стохастичні коливання. Режим стохастичних коливань інтерпретується як готовність системи до швидкого переходу в новий стан.

Експериментальні дослідження та моделювання нюхової системи дозволили У. Фрімену та ін. [212] розробити нові алгоритми класифікації, застосування яких для розв'язання однієї з прикладних задач розпізнавання зображень показало їх високу ефективність у порівнянні з традиційними методами.

Моделі зорової системи. Моделі зорової системи присвячені в більшості поясненню експериментальних даних про виникнення високочастотних коливань в первинних областях зорової кори у відповідь на подання стимулу. Важливий експериментальний факт, що знайшов відображення в моделях, полягає в тому, що за певних умов стимуляції виникаючі коливання синхронізуються з нульовою різницею фаз навіть у тому випадку, коли коливання спостерігаються в далеких зонах кори. Для пояснення цього факту в моделях розглядається можливість синхронізації під впливом загального джерела або синхронізації в результаті взаємодії осциляторів з тимчасовими затримками у зв'язках. Щоб з'ясувати, яка з цих можливостей реалізована в нервовій системі, потрібні додаткові дослідження.

Моделювання локомоторної активності. Основний інтерес з точки зору ОНС в роботах з моделювання локомоторної активності

представляють моделі центрального генератора патернів. Такий генератор, керований зовнішнім сигналом, здатний виробляти різні набори коливальних патернів, що відповідають певним типам ритмічних рухів. У досліджуваних моделях центральний генератор патернів складається або з невеликої кількості осциляторів (два – чотири), або являє собою ланцюжок локально з'єднаних осциляторів. Випадок великої кількості осциляторів, а також більш складна геометрія зв'язків практично не досліджені. Недостатньо вивчені і питання щодо самоорганізації центрального генератора патернів в онтогенезі, “записи” набору програм, управління переходами від однієї програми до іншої, коригування патернів активності сенсорними входами тощо.

Моделі пам'яті та уваги. Завдання моделювання пам'яті давно привертає увагу дослідників. Велика частина сучасних наукових праць заснована на ідеї формування в нейронній сітці безлічі стійких стаціонарних станів, які відповідають запам'ятовуваним образам. На жаль, відомі моделі мають малу інформаційну ємність і працюють недостатньо ефективно. Проведені дослідження в цьому напрямку показують принципову можливість побудови пам'яті на осциляторних елементах, проте переваги такого підходу ще не виглядають досить переконливо. Можливо, більший успіх буде досягнутий при переході до системних моделей мозку, які будуть враховувати взаємодію різних його структур і дозволять розглядати пам'ять у тісному зв'язку з іншими сторонами психічної активності, такими, наприклад, як інтеграція ознак і уваги.

Запропоновані моделі уваги дозволяють пояснити деякі результати психологічних експериментів, однак цілісна концепція всього процесу обробки інформації на основі регулярної періодичної і стохастичної динаміки ще не склалася. Цікавою є роль високочастотних (кора) і низькочастотних (гіпокамп) коливань і їх взаємодія на різних етапах обробки інформації.

Одним із стимулів розвитку ОНС є бажання сформулювати нові принципи обробки інформації, подібні до тих, які лежать в основі функціонування мозку, і використовувати ці принципи для розробки високоефективних технічних пристроїв. Робота в цьому напрямку тільки розпочалася. У більшості робіт з моделями ОНС містяться лише результати імітації і пояснення експериментальних даних.

Дослідження в області моделювання ОНС є спробами пояснити результати низки нейрофізіологічних і психологічних експери-

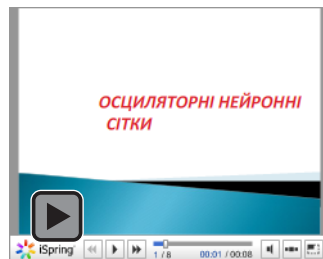
ментів, які не знаходять пояснення в рамках традиційних неколивальних моделей НС. Виникаючі тут труднощі очевидні: в одних випадках не ясні шляхи коректної формалізації даних, в інших – існують різні підходи до моделювання, але експериментальний контроль модельних передбачень обмежений. Проте, продовження досліджень в даному напрямку вкрай важливе.

Контрольні завдання

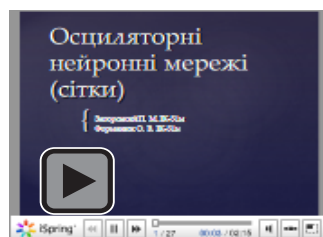
1. Розкрити визначення осциляторної нейронної сітки.
2. Охарактеризувати два типи осциляторних нейронних сіток.
3. Пояснити термін “динаміка осциляторів” та розкрити його значення в динамічних осциляторних нейронних сітках.
4. Прокоментувати сутність осциляторної синхронізації і розкрити її роль в осциляторних нейронних сітках.
5. Проаналізувати можливість використання осциляторі Ван дер Поля з розподіленими частотами та зовнішніми шумами. Охарактеризувати механічну і теплову біфуркації та біфуркацію типу Ландау.
6. Навести особливості використання в осциляторних нейронних сітках фазових осциляторів зі слабкими зв'язками.
7. Охарактеризувати особливості використання в осциляторних нейронних сітках фазових осциляторів з сильними зв'язками.
8. Пояснити, яку роль відіграють введення тимчасових затримок в переданні сигналів в осциляторних нейронних сітках з затримками у зв'язках.
9. Навести і прокоментувати приклади використання динамічних осциляторних нейронних сіток.

Презентації

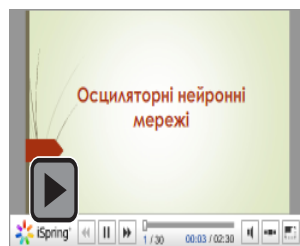
1. Осциляторні нейронні сітки



2. Осциляторні нейронні мережі (сітки)



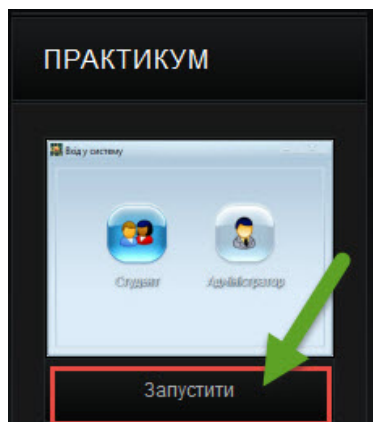
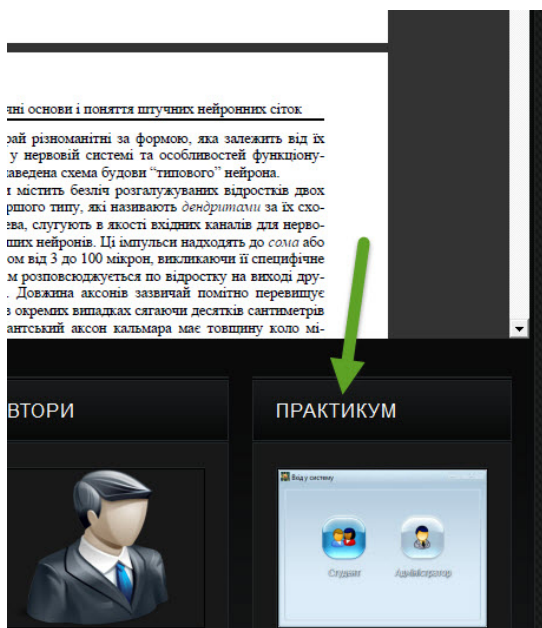
3. Осциляторні нейронні мережі



Практичні заняття

Осциляторні нейронні сітки

Для запуску практичних занять на
вкладці "Практикум" натисніть
"Запустити"



Розділ 20

Синергетичний комп'ютер

Одни противоположности порождают иные
Марк Тулій Цицерон

*Одни говорят слишком много, чтобы ничего не сказать,
другие тоже говорят слишком много, потому что не знают, что
сказать.*

*Одни говорят, чтобы скрыть, что они думают,
другие — чтобы скрыть, что они ничего не думают*
Василий Ключевский

*Дар уявлення та фантазії значать більше,
аніж будь-які здібності до абстрактного мислення*
Альберт Ейнштейн

20.1. Означення й поняття в синергетиці

Синергетика перекладається як “енергія сукупної дії”. Засновником синергетики як наукового поняття вважається проф. Штутгартського університету Герман Хакен [127], який не тільки визначив місце і роль цього напряму наукових досліджень в теорії та практиці вивчення природних явищ і відтворення процесів на основі самоорганізації систем, але й запропонував нову топологію інтелектуальних нейронних сіток (ІНС) – синергетичний комп'ютер (СК). Перш, ніж ознайомитись з принципами організації і функціонування СК, розглянемо основні поняття синергетики.

Означення 20.1. *Синергетика* – міждисциплінарний напрямок наукових досліджень, завданням якого є вивчення природних явищ і процесів на основі принципів самоорганізації систем, що складаються з підсистем.

Іншими словами, синергетика – наука, що займається вивченням процесів самоорганізації виникнення, підтримки, стійкості і розпаду структур різної природи.

Цілі синергетики. Багато об'єктів, що визначаються в науках про людину, можуть, так чи інакше, розглядатися як системи. Такі об'єкти/системи складаються з багатьох частин/підсистем, що взаємодіють одна з одною більш-менш складним чином, і синергетика допомагає виявляти, яким чином взаємодія таких підсистем приводить до виникнення просторових, часових або просторово-часових структур в макроскопічному масштабі.

Прикладами можуть служити суспільство, яке складають люди, організм з його клітинами, мозок з його нейронами або, наприклад, у фізиці – рідина, що складається з молекул. Через кооперацію окремих частин у системи з'являються нові якості, в яких можна виявити ефекти самоорганізації.

В буквальному сенсі синергетика, *по-перше*, визначає явища, які виникають від *сукупної дії декількох різних факторів*, тоді як кожний фактор відокремлено до цієї дії не приводить.

По-друге, синергетику часто визначають як науку про *самоорганізацію*, тобто мимовільне ускладнення форми або структури системи при повільному і плавному змінненні параметрів (прояви так званої дисипативності).

І, *по-третє*, синергетика – це наука про *неочікувані* (непередбачені) *явища*. Аналіз, який розкриває причину неочікуваного явища, і складає *предмет* синергетики.

Метод синергетики (математичний апарат) – теорія динамічних систем.

Однією з *задач* синергетики є виявлення законів побудування організації і виникнення упорядкованості як явищ, їх розвитку і самовдосконалення.

Початковою точкою для всіх досліджень в області синергетики є адекватне подання *стану системи* на різних рівнях.

Означення 20.2. *Стан системи* — характеристика системи на даний момент її функціонування. Оскільки система описується визначеним комплексом суттєвих змінних і параметрів, то для того, щоб виразити стан системи, треба визначити значення, які приймаються ними в розглядуваний момент.

З позицій синергетики будь-який об'єкт можна подавати багаторівневою системою. На *мікроскопічному рівні* подається поведінка об'єктів, наприклад, автотранспорту на вулиці – можна прослідкувати рух кожного окремого автомобіля і потім відобразити рух цих об'єктів на деякій діаграмі. На *мезоскопічному рівні* виокремлюють деяку область на вулиці, в якій все ще будуть якісь автомобілі, але ця область є вже меншою у порівнянні з усією вулицею. На цьому рівні можна визначити середню кількість та середню швидкість автомобілів в такій області.

В решті решт, на *макроскопічному рівні* можна дослідити, чи існують якісь властивості великомасштабної поведінки транспорту, наприклад, коли на одному боці вулиці рух відбувається в одному

напрямку, а на другому боці – в протилежному напрямку. Зрозуміло, що подання таких станів системи на різних рівнях може відноситись до цілковито різних кількостей об'єктів, а також до абстрактних понять, наприклад, до думки або поведінки людей або цілих соціальних груп. Подання поведінки системи на різних рівнях може бути виконаним за допомогою так званого *вектора станів* – першого важливого для синергетики поняття.

Означення 20.3. *Вектор станів* – вектор, який містить повну інформацію про стан системи.

Наступне поняття, яке використовується в синергетиці – *керуючий параметр*, який може бути представлений як поодиноким, так і декількома керуючими параметрами.

Означення 20.4. *Керуючі параметри* (або управління) – поняття математичної теорії оптимальних процесів, динамічного програмування: змінні величини (функції часу), визначаючи напрямки і швидкість руху керованої системи у фазовому просторі. Керуючі параметри характеризують рішення, які треба здійснювати в кожний даний момент часу з інтервалу між початковим і кінцевим станами системи.

Твердження 20.1. Кількість керуючих параметрів фіксована і накладається на систему ззовні.

Твердження 20.2. Керуючі параметри не змінюються із зміненнями системи.

Синергетика фокусує свою увагу на тих ситуаціях, в яких поведінка системи змінюється якісно при змінненні керуючих параметрів.

Твердження 20.3. Структура системи називається стійкою, а система – структурно стійкою, якщо вона зберігається при змінненні умов середовища (тобто керуючих параметрів); але якщо структура змінюється, то говорять про відносну нестійкість.

Як показано математично в синергетиці, в багатьох випадках поведінка системи, близька до таких точок нестійкості, може залежати від поведінки досить небагатьох змінних, можна навіть сказати, що поведінка окремих компонент системи просто визначатиметься цими небагатьма факторами. Ці фактори називають *параметрами порядку*, які можуть, в тому числі, представляти неупорядковані, хаотичні стани або керувати ними.

Означення 20.5. *Параметр порядку* – величина, що характеризує даний порядок в середовищі, який виникає в результаті спонтанного порушення симетрії при фазовому переході.

Параметри порядку відіграють суттєву роль в концепції синергетики. Вони “підкоряють” окремі компоненти, тобто визначають поведінку цих компонент. Зв'язок між параметрами порядку та окремими компонентами системи називається *принципом підпорядкованості*. З визначенням параметрів порядку практично описується поведінка системи. В такому випадку замість того, щоб описувати поведінку системи через опис окремих її частин, треба мати справу або описувати поведінку тільки параметрів порядку. Відбувається, так би мовити, величезне інформаційне стиснення.

В умовах нового керуючого параметра система сама створює специфічні структури, що й називається *самоорганізацією*.

Означення 20.6. Самоорганізація — процес упорядкування елементів одного рівня в системі за рахунок внутрішніх факторів, без зовнішнього специфічного впливу. Як результат — виникнення одиниці наступного якісного рівня.

Твердження 20.4. Рівноважний параметр порядку дорівнює нулю у неупорядкованій фазі (рис. 20.1, а) і відмінний від нуля — у впорядкованій (рис. 20.1, б-д).

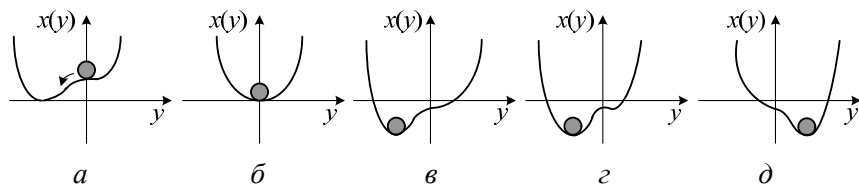


Рис. 20.1. Символічна форма подання стану параметра порядку

Стан одного параметра порядку традиційно відображає в символічній формі позицією кульки на горбистій поверхні. Коли кулька знаходиться “в глобальному мінімумі”, є тільки один стан параметра порядку. Цей стан *стійкий*. Після кожного збурення кулька повертається в *позицію рівноваги*.

При зміні керуючого параметра поверхня може спотворюватися (рис. 20.1, в), наближаючи частину поглиблення до мінімального (локального або глобального) значення. Якщо вважати, що параметр порядку все ще підпорядкований маленьким випадковим подіям (так званими коливаннями), це означає, що кулька буде гойдатися навколо однієї з позицій рівноваги.

Коливання досить великої амплітуди розгойдування можуть стати “критичними”.

Поведінка параметрів порядку. Коли значення параметра перевершить критичний рівень, може з'явитися новий ландшафт – з двома мінімумами (ямками) (рис. 20.1, *з*). Вище зазначена (рис. 20.1, *а*) позиція кульки стане нестабільною, кулька тепер може скотитися в будь-який з двох мінімумів (тобто має місце порушення симетрії) – наприклад, як це наведено на рис. 20.1, *з*.

У цій ситуації маленьке коливання вирішить, в якому напрямку буде рухатися кулька, причому, система може спуститися тільки в лівий або правий мінімум, що повинно порушити симетрію.

Якщо подивитися на рис. 20.1, *в*, кулька, що представляє стан параметра порядку, розташовується в найнижчій ямці. На рис. 20.1, *з*, демонструється процес зміни значення керуючого параметра ландшафту, але кулька все ще розташовується у попередній лівій позиції.

Нарешті, у міру подальшого збільшення параметра управління, ландшафт набуває “праву” форму (рис. 20.1, *д*), і тепер кулька зайняла нову позицію, яка вказує, що параметр порядку i , таким чином, система змінили свій стан.

Тепер змінимо керуючий параметр в протилежному напрямку і простежимо відповідні зміни в позиціях кульки (рис. 20.2, *а-в*). У середній частині (рис. 20.2, *б*) кулька залишається в різній позиції по відношенню до відповідного положення на рис. 20.1, *з* при одному і тому ж значенні керуючого параметра. Послідовність, в якій були виконані зміни в положеннях кульки, зіграла роль в стані системи при проміжному значенні керуючого параметра.

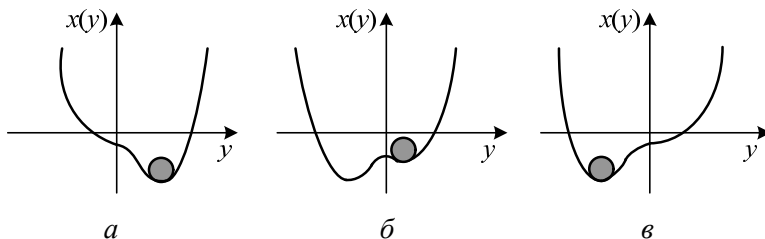


Рис. 20.2. Розташування кульки при зміні керуючого параметру відносно наведеного на рис. 20.1, *в-д*

Таким чином, стан системи залежить від її історії – з яких попередніх станів прийшла система до цього стану. Отже, можна сформулювати наступні висновки:

- для одного і того ж значення керуючого параметра існують два можливі стани (що видно з попереднього зіставлення положень кульки на рисунках 20.1, *з* та 20.2, *б*).

- така подвійність перспектив у розвитку системи і затримка, інерційність системи в переході до якісно іншого стану називається *гістерезисом* і з'являється саме тому, що існують два, а не один параметр порядку.
- іншим прикладом двох параметрів порядку є коливання системи в межах деяких значень – це так звані *обмежені цикли*.

Гістерезис. Явище гістерезису можна спостерігати, якщо розглядати послідовність зображень (еволюційний їх перегляд) на рис. 20.3, починаючи з лівого верхнього до правого нижнього. При цьому можна бачити, як чоловіче обличчя поступово перетворюється на фігуру дівчини (станеться це приблизно в середині нижнього ряду).

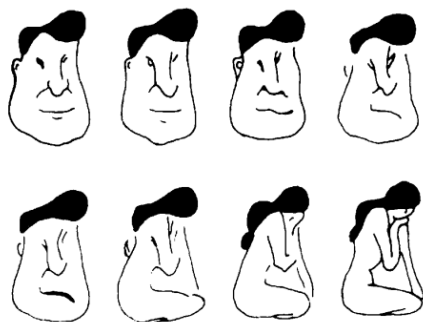


Рис. 20.3. Інтерпретація гістерезисного явища в еволюційному перегляді зображень

Роздивляючись зображення в зворотному порядку, виявляється, що фігура дівчини помітна ще й у верхньому ряду, і лише під кінець вона знову стає чоловічим обличчям.

Таким чином, можна співставити таку невідповідність в уявлених зображеннях при “прямому” і “зворотному” послідовних перегляданніях як наявність гістерезисного циклу з притаманною йому “петлею гістерезису”.

20.2. Синергетичний комп'ютер

Як вже наголошувалося, інтелектуальна нейронна сітка (ІНС) “синергетичний комп'ютер” (рис. 20.4) була запропонована засновником синергетики Г. Хакеном, функціонування якої може бути пояснено з синергетичної точки зору [48].

Цей комп'ютер відноситься до *асоціаторів*, а збережені в ньому образи можуть бути інтерпретовані як локальні енергетичні мінімуми.

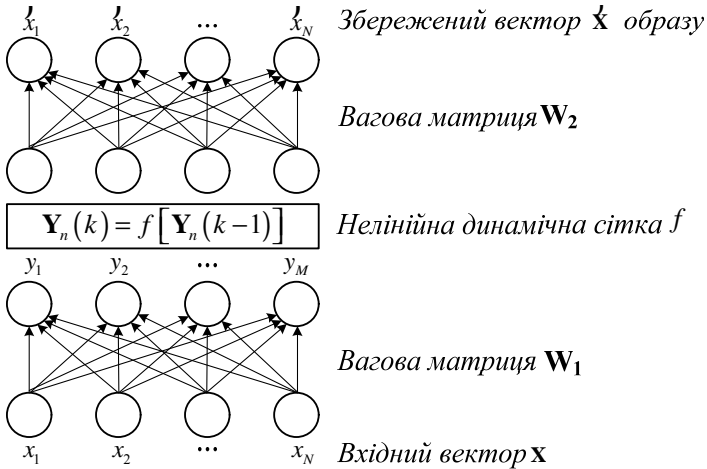


Рис. 20.4. Структура інтелектуальної нейронної сітки – синергетичного комп'ютера

Ця сітка має багато спільного з сіткою Хопфілда (див. розд. 6.7) і, крім того, може розглядатися як один з її варіантів, який працює з аналоговими сигналами.

Відмінність його від мережі Хопфілда полягає в тому, що його нейрони не містять нелінійностей через те, що нелінійне перетворення відбувається спеціальним модулем.

Принцип побудови синергетичного комп'ютера. Інтелектуалізована НС складається з вхідного шару, на який надходить зашумлений або зіпсований N -мірний образ, який відповідає одному з M збережених образів :

$$\bar{x}_n, n = \overline{1, M}. \quad (20.1)$$

Вхідний вектор X перетворюється нейронним шаром з ваговою матрицею W_1 . Вихідним сигналом шару є M -вимірний вектор Y , кількість компонентів якого співпадає з кількістю збережених образів.

Вектор Y подається на нелінійний модуль, що описується системою нелінійних диференціальних рівнянь, який після деякої кількості ітерацій обчислює нові значення компонентів вектора Y .

Далі вектор Y з новими обчисленими компонентами подається на наступний шар через матрицю W_2 . Сигнал на виході останнього шару \bar{X} є вектором, який відповідає образу X_m .

Таким чином, асоціативний виклик збереженого образу відбувається подачею на вхід вектора X .

Вибір значень елементів матриць W_1 та W_2 визначається безпосередньо векторами збережуваних образів за формулою:

$$\mathbf{W}_1^T = \mathbf{W}_2 = [x_1, x_2, \dots, x_l], \quad l = \overline{1, M}. \quad (20.2)$$

Вихідний сигнал першого шару

$$y = \mathbf{W}_1 x. \quad (20.3)$$

Початкові значення компонентів вектора y :

$$y_n(k) = y_n(k-1) \left[\alpha_k - \beta \sum_{i=1; i \neq n}^M y_i^2(k-1) + (\beta - 1) y_n^2(k-1) \right]. \quad (20.4)$$

Рівняння можна розглядати як рівняння руху тіла в потенціальному полі, на яке діє сила, що задається похідною потенціальної функції по координатах тіла.

Вигляд потенціальної функції

$$E(y_n) = -\frac{1}{2} \sum_{i=1}^M \alpha_i y_i^2 + \frac{\beta}{4} \sum_{i=1; i \neq n}^M y_n^2 y_i^2 + \frac{1}{4} \sum_{i=1}^M y_i^4. \quad (20.5)$$

Рівняння руху

$$y_n(k) = -\frac{dE}{dy_n}. \quad (20.6)$$

Після рекурсивних розрахунків тільки один компонент вектора на виході буде дорівнювати 1, а всі інші — нулю. Цей вектор передається з ваговою матрицею W_2 на шар на виході

$$\mathbf{x} = W_2 y. \quad (20.7)$$

На відміну від сітки Хопфілда СК не має нестійких станів. Його динамічний модуль завжди збігається до одного з базисних векторів. Кількість збережених образів M не може бути більше розмірності образу N .

Можна сказати, що дану сітку назвали “синергетичним комп'ютером” тому, що *основним принципом синергетики є аналіз стаціонарних станів динамічної системи, залежних від початкових умов та областей притягання її стаціонарних точок.*

Отже, система рівнянь

$$y_n(k) = f[y_n(k-1)] \quad (20.8)$$

описує синергетичний модуль ІНС.

Кожному пікселю сприйнятого зображення ставиться у відповідність “модельний” нейрон (рис. 20.5), що становить, в сукупності з іншими нейронами, деякий плоский шар.

Кожний окремий нейрон моделі характеризується власними

рівнями збудження та активності, що змінюються залежно від зміни відтінку сірого кольору (аналогічним чином світло різної інтенсивності впливає на зорові клітини сітківки).

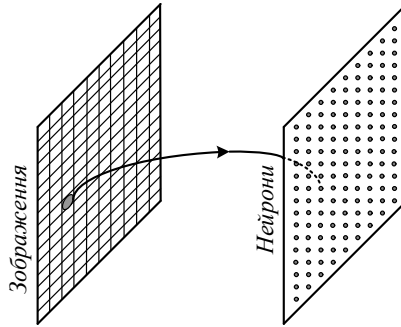


Рис. 20.5. Модель сприйняття зображення синергетичним модулем ІНС

Мережева архітектура синергетичного комп'ютера. Нейрони мозку з'єднані один з одним синапсами; в моделі мережевої архітектури (рис. 20.6) нейрони також з'єднані один з одним, причому кожний модельний нейрон безпосередньо пов'язаний з усіма іншими нейронами і здатний посилати їм сигнали, сила яких відповідає його власному рівню збудження.

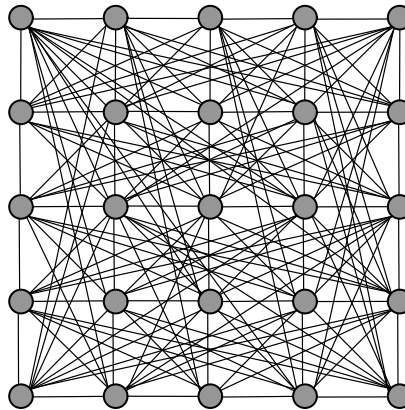


Рис. 20.6. Мережева архітектура синергетичного комп'ютера

20.3. Приклади використання синергетичного підходу

Розглянемо декілька прикладів використання синергетичного підходу до задач розпізнавання образів, виокремивши деякі основні властивості синергетичного підходу при цьому.

Задання прототипів. На рис. 20.7 наведені приклади використовуваних облич як прототипів осіб. Літери служать для ідентифікації образів і в той же час виступають в ролі кодового позначення зображень.



Рис. 20.7. Зображення образів з кодовими позначеннями

Розпізнавання. Наведені на рис. 20.8 зображення обличчя ілюструють хід процесу розпізнавання образу синергетичним комп'ютером. Якщо СК пред'явити фрагмент зображення обличчя (крайня ліва картинка), то комп'ютер зможе відтворити обличчя цілком і встановити його кодове позначення.



Рис. 20.8. Послідовність процесу відновлення зображення образу і його кодового позначення

Якщо СК пред'явлена тільки кодова літера (наприклад, D), що служить позначенням одного із збережених образів, то комп'ютеру в процесі розпізнавання вдається відтворити через стадію “розмитості” відповідну особу (рис. 20.9).



Рис. 20.9. Процедура відтворення синергетичним комп'ютером розпізнаваного образу за кодовим позначенням

Насичення уваги. Проте в деяких (але нечисленних) випадках комп'ютер видає невірні відповіді. Але ж і людські здібності до

розпізнавання не безмежні. Переконатися в цьому допоможе рис. 20.10, *а*, на якому, на перший погляд, наведене зображення Ейнштейна. Вдивляючись в обличчя пильніше, в деякий момент часу можна розгледіти на рисунку пейзаж з трьома купальницями. Наша власна здібність до розпізнавання (а втім, і аналогічна “здібність” комп'ютера) керується якимись “очікуваннями”, а точніше – певними параметрами уваги. Таким чином задіється ефект так званих *амбівалентних*, або “подвійних”, зображень.

Наступне зображення на рис. 20.10, *б* відоме ще з римських часів. Безумовно в цій статуетці можна розгледіти і стару жінку, обгорнуту шаллю, і закохану пару.

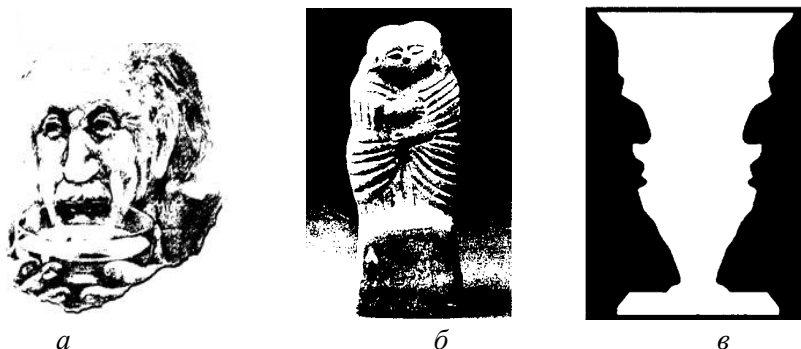


Рис. 20.10. Приклади синергетичного підходу щодо сприйняття зображень з урахуванням “насичення уваги”

Досліджуючи сприйняття всіх цих подвійних картинок, можна прийти до одного дуже важливого висновку, а саме – неможливо утримувати увагу на будь-якому з “варіантів” інтерпретації такої картинки довше декількох секунд.

Наприклад, розглядаючи протягом деякого часу білу вазу (рис. 20.10, *в*), раптом усвідомлюєш, що вази більше немає, а замість неї на малюнку – два чорних профілі; потім вони зникають, повернувши на місце білу вазу, але незабаром знову повертаються і т. д. причому відбувається все це зовсім мимоволі.

Отже, що на рис. 20.10, *а-в* відповідно наведено: портрет Ейнштейна чи пейзаж з трьома купальницями, стареньку жінку чи двох закоханих, білу вазу чи два чорних обличчя? Якщо сконцентрувати увагу на одній з версій, то через деякий час перед нами мимоволі з’являється уявлення другої версії. Цей цікавий психологічний фе-

номен дуже легко пояснюється, якщо уявити, що як тільки образ (або його частина) за першою версією стає розпізнаним, відповідний параметр уваги за цією версією перетворюється в нуль. Далі до справи “підключається” параметр уваги, віднесений вже до другої версії зображення, і він володіє ситуацією доти, поки і цей образ другої версії не буде розпізнаний, після чого увага знов переключається на першу версію зображення.

Підсумовуючи ці ідеї, можна стверджувати: *описані механізми розпізнавання образу можуть транслюватися у подібні ж механізми прийняття рішень.*

Бістабільність сприйняття. І нарешті можна виокремити ще одну властивість синергетичного трактування процесів сприйняття розпізнаваних образів – бістабільність.

Бістабільність сприйняття одного і того ж зображення можна описати за допомогою руху м'яча по горбистій поверхні (рис. 20.11). Стійкі положення м'яча відповідають інтерпретації зображень “портрет Ейнштейна” – “три купальниці”, “жінка похилого віку” – “закохана пара” та “ваза” – “особи”, наведених на рис. 20.10, а-в, відповідно.

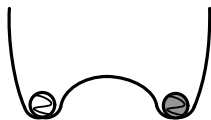


Рис. 20.11. Бістабільність сприйняття розпізнаваних образів

Початкова інформація визначає положення м'яча праворуч або ліворуч від “вершини”, після чого, в міру розглядання картинки, м'яч скочується в одну з “локальних долин” – сприйняття успішно завершено.

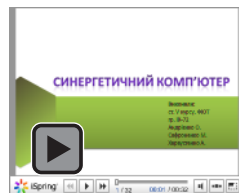
Контрольні завдання

1. Охарактеризувати синергетику як новий міждисциплінарний напрямок наукових досліджень.
2. Окреслити і прокоментувати цілі синергетики.
3. Дати визначення предмета, метода та основних задач синергетики.
4. Дати визначення стану і вектора станів системи охарактеризувати багаторівневість подання об'єктів с позицій синергетики.

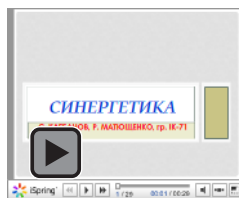
5. Пов'язати поняття “керуючих параметри” зі структурною стійкістю системи.
6. Навести роль параметрів порядку в концепції синергетики і дати тлумачення принципу підпорядкованості компонент в системі з позицій синергетичного аналізу.
7. Навести умови, за яких шляхом самоорганізації створюються специфічні структури.
8. Розкрити зміст понять “рівноважний параметр порядку” та “критичні коливання” системи при зміні керуючого параметра.
9. Дати тлумачення явищу гістерезису в еволюційному перетворенні зображень в синергетичному комп'ютері.
10. Охарактеризувати структуру інтелектуальної нейронної сітки Г. Хакена і прокоментувати принципи її побудови та функціонування.
11. Аргументувати схожість та відмінності синергетичного комп'ютера відносно сітки Хопфілда.
12. Розкрити основний принцип синергетики – аналіз стаціонарних станів динамічної системи як такий, що “спровокував” Г. Хакена назвати запропоновану ним інтелектуальну НС синергетичним комп'ютером.
13. Прокоментувати мережеву структуру синергетичного комп'ютера.
14. Навести приклади використання синергетичного комп'ютера та розтлумачити основні властивості синергетичного підходу в задачах розпізнавання образів.

Презентації

1. Синергетичний комп'ютер



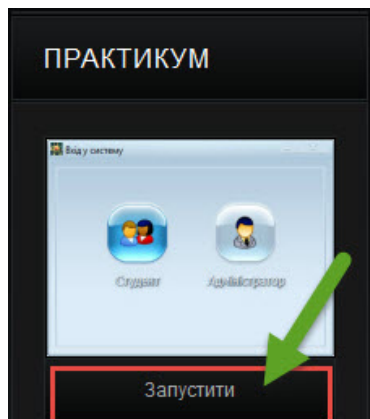
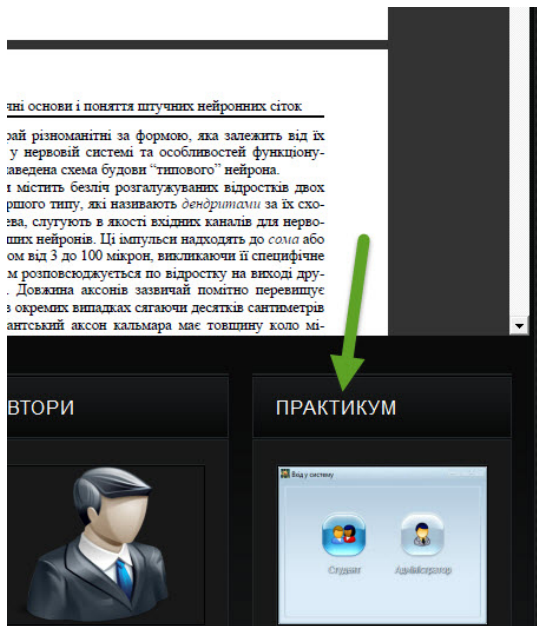
2. Синергетика



Практичні заняття

Синергетичний комп'ютер

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 21

Недостовірні та нечіткі знання

*Теорії, в які ми віримо, ми називаємо фактами,
а факти, в які не віримо – теоріями*
Фелікс Коен

*Математик вже децю може, проте, певна річ, не те,
що від нього жадають отримати на даний момент*
Альберт Ейнштейн

*Коли слухаючий не розуміє промовляючого,
а промовляючий не знає, що він має на меті, – це філософія*
Марі Франсуа Аруе

21.1. Загальні положення

Найбільш вражаючою властивістю людського інтелекту є спроможність приймати правильні рішення в умовах неповної та нечіткої інформації. Побудова моделей наближених міркувань людини та використання їх в інтелектуальних комп'ютеризованих системах є нині одним з найперспективніших напрямів розвитку сучасної обчислювальної техніки.

Якщо для класичного поняття множини характеристична функція в інтервалі $(0, 1)$ набуває значення 0 або 1, то, поширивши поняття на *нечіткі множини*, говорять, що *функція належності* (ФН) в цьому випадку може набувати будь-яких значень в межах цього інтервалу [214]. В цій праці американський вчений Лотфі Заде визначив також ряд операцій над нечіткими множинами і запропонував узагальнення відомих методів логічного виведення «modus ponens» і «modus tollens».

Запроваджуючи поняття *лінгвістичної змінної* і припустивши, що її значеннями (термами) виступають нечіткі множини, Л. Заде запропонував апарат для подання процесів інтелектуальної діяльності, включаючи нечіткість і невизначеність висловлювань. Це дало змогу створити фундамент теорії нечітких множин і нечіткої логіки, а також передумову для впровадження методів нечіткого управління в інженерну практику. У свою чергу, це зумовило постановку цілого ряду нових проблем, зокрема таких, як створення нових архітектур комп'ютерів для нечітких обчислень, елементної бази нечітких комп'ютерів і контролерів, методів і інструментальних засобів розробки нечітких систем управління тощо.

Математична теорія нечітких множин дає змогу подавати нечіткі поняття і знання, оперувати ними та робити нечіткі виведення. Нечітке управління є особливо корисним, коли досліджувані процеси є занадто складними для аналізу за допомогою загальноприйнятих методів або коли доступні джерела інформації інтерпретуються неякісно, неточно або невизначено. Нечітка ж логіка, на якій ґрунтується нечітке управління, надаючи ефективні засоби відображення невизначеностей і неточностей реального світу, ближче до людського мислення та істотних мов, аніж традиційні логічні системи.

Характерним для фаззи-логіки і фаззи-управління є безпосереднє застосування експертних знань, які якісно формулюються для генерування керуючих впливів. Знання про взаємодію з процесом за цією методикою подається у формі *правил продукції* такого виду:

ЯКЩО (вихідна ситуація), ТО (відповідна реакція). Такі ЯКЩО-ТО- правила відповідають найпростішій формі людських взаємодій. На відміну від розглянутого в роботі [73] (див. *книгу 2, розд. 2.2*), у фаззи-логіці “вихідна ситуація” і “відповідна реакція” являють собою нечіткі висловлення, наприклад:

ЯКЩО (швидкість пересування ланки маніпулятора *є надмірною*),

ТО (напругу на відповідний привідний двигун *зменшити*).

Або таке логічне висловлення:

ЯКЩО (при захоплюванні деталі спостерігається її *незначне* проковзування в захватному пристрої ЗП промислового робота ПР) І (*є достатній* запас міцності деталі для збільшення зусилля захоплення), І (діапазон зусилля захоплення ЗП уможливорює *деяке* його збільшення), ТО (необхідно *повністю* використати можливості ЗП).

Форму опису, що дає змогу операціоналізувати сформульовані вище знання, фаззи-теорія розглядає як розширення класичної теорії множин [214].

Отже, апарат нечітких множин і нечіткої логіки вже давно і з успіхом застосовується для розв’язування задач, в яких вихідні дані є ненадійними та слабоформалізованими.

Водночас для нечітких систем характерні й деякі недоліки, зокрема:

- вихідний набір постульованих нечітких правил формулюється експертом-людиною і може виявитися неповним або суперечливим;
- вид і параметри ФН, які описують параметри системи на вході та на виході, вибираються суб’єктивно і можуть виявитися не зовсім такими, що відображають реальну дійсність.

Для усунення, принаймні частково, наведених недоліків намагаються створювати нечіткі системи адаптивними, коригуючи, у міру їх функціонування, правила та параметри ФН. Найбільш вдалими прикладами таких систем є, зокрема, нечіткі нейронні сітки, які формально за структурою ідентичні з багат шаровими нейронними сітками з навчанням, наприклад, за алгоритмом зворотного поширення похибки (див. *розд. 4.5*), проте сховані шари в ній відповідають етапам функціонування нечіткої системи:

- перший шар нейронів виконує функцію введення нечіткості (fuzzification) на основі заданих ФН входів;
- другий шар відображує сукупність нечітких правил;
- третій шар виконує функцію зведення до чіткості (defuzzification).

Кожний з цих шарів характеризується набором параметрів (ФН, нечіткими розв'язувальними правилами, активаційними функціями, вагами зв'язків), настроювання яких виконується по суті так само, як і для звичайних нейросіток.

21.2. Поняття про недостовірні та нечіткі знання

Неповні та недостовірні знання. В *розділах 1 та 2* в роботі [78] розглядалися проблеми, пов'язані з використанням знань в експертних системах (ЕС) та системах подання знань (СПЗ). Розглянемо інші аспекти цієї проблеми.

Неповні знання. Ця категорія знань свідчить про те, що для доведення або спростування певного твердження може не вистачати необхідної інформації. У багатьох системах логічного виведення прийнято *постулат замкненості* світу [13], що означає, що на запит про істинність деякого твердження відповіддю буде “так” тоді і тільки тоді, коли його можна довести; якщо ж довести це твердження неможливо, то відповіддю стане “ні”. Водночас “*неможливо довести через брак інформації*” (тобто неповноту знань) і “*доведено, що ні*” (тобто підтверджена неможливість використання знань через їх неповноту або хибність) — це зовсім не одне й те саме. Тому бажано, щоб в ЕС, якій доводиться мати справу з такими невизначеностями, формувався запит до користувача про факти, яких не вистачає.

Недостовірність знань. На результат виконання деякого правила в продукційній СПЗ можуть впливати як випадкові чинники (*об'єктивна невизначеність*), так і такі, які залежать від невпевненості експерта в якомусь факті чи правилі виведення (*суб'єктивна невизначеність*).

Таким чином, ненадійність знань і недостовірність наявних фактів можуть значною мірою вплинути на результати логічного виведення, які формуються інтелектуалізованою системою, і цей фактор обов'язково повинен враховуватися в процесі логічних побудов. Якщо ж просто відкидати факти і правила виведення, які викликають сумнів, то доведеться відмовитися від цінної інформації, яка була неточно подана або неправильно інтерпретована системою. Часто буває так, що експерт не зовсім упевнений в тому чи іншому факті, але, незважаючи на це, інформація залишається цінною і повинна бути включена в базу знань. Тому необхідно розвивати процедури, які дають змогу здійснювати логічні побудови при недостовірних даних, використовуючи ці процедури в ЕС.

Нечіткість знань. На практиці часто доводиться мати справу з неточно визначеними або нечітко сформульованими поняттями — лінгвістичними *невизначеностями*, такими, як “багато”, “мало”, “сильно”, “слабо” тощо.

Враховуючи викладене, наведемо деякі означення.

Означення 21.1. *Висловлювання називається неточним*, якщо його істинність або хибність не можуть бути встановлені однозначно, тобто твердження не є ані абсолютно достовірним, ані абсолютно хибним [27].

Означення 21.2. *Неточним виведенням називається* логічне виведення в умовах неточності (недостовірності) знань.

Модальне твердження. Неточне виведення традиційно розглядається як самостійний напрям, хоча неточне твердження можна було б інтерпретувати як частковий випадок *модального твердження*. Суть останнього полягає в тому, що воно оперує твердженнями, які дають оцінку іншим твердженням, або, інакше кажучи, є *опосередкованими* через інші твердження.

Означення 21.3. *Модальними логіками* є логічні системи, які оперують модальними твердженнями.

Як приклад наведемо такі модальні твердження, які можна занести в БЗ експертної системи: “ПР може захопити неправильно зорієнтовану деталь” (у таких випадках говорять про *логіку можливого*, або *алетичну логіку*) або “Система інформаційного забезпечення електророботарки визначає, що останній пройшов контрольну (реперну) точку відліку його місцерозташування в просторі” (це приклад так званих *епістемічних логік*).

Неточне виведення слід відрізняти від роботи з нечіткими знаннями, хоча останні також формально підпадають під сформульоване вище визначення.

21.3. Чіткі та нечіткі множини

Властивості чітких множин. Нехай M – універсальна множина, x – елемент M , а G – деяка властивість. Звичайна (чітка) підмножина A універсальної множини M , елементи якої задовольняють властивості G , визначається як множина упорядкованих пар:

$$A = \{\mu_A(x) / x\},$$

де $\mu_A(x)$ – *характеристична функція* (або ФН), для якої:

$$\mu_M(x) = \begin{cases} 1, & \text{якщо } x \text{ задовольняє властивості } G; \\ 0 - & \text{в протилежному разі.} \end{cases}$$

Види характеристичних (активаційних) функцій розглянуто в розд. 1.2.

Властивості нечітких множин. Великого значення для практичного застосування в задачах, які розв'язуються інтелектуальними роботами, набуває теорія нечітких множин, яка дає змогу будувати керування складними об'єктами в умовах лінгвістичної невизначеності. Наприклад, можна сказати: об'єкти роботизації, вага яких визначається грамами, *зовсім неважкі*, об'єкти вагою декілька кілограм *більш менш важкі*, а вагою десятків кілограм — *досить важкі* і т. ін. При цьому так і залишається невизначеним поріг, з якого об'єкти роботизації стають важкими. Для формалізації таких умов використовують апарат *нечітких множин*.

Останнім часом у дослідженнях нечітких множин велика увага приділяється розробці *правил нечіткого умовного логічного висновку*, так званого Fuzzy Conditional Inference Rules, або скорочено FCIR. Ця обставина пов'язана з тим, що в семантиці звичайної мови існує певна кількість нечітких концепцій, і робляться логічні висновки, в яких передумови і наслідки включають такі концепції.

У 1965 р. професором Л. Заде [22] було введено поняття *функції нечіткої належності* (ФНН) $\mu_M(x)$ елемента до множини, дозволивши цій функції набувати будь-якого значення від 0 до 1. Чим більшим є значення ФН, тим більшою мірою елемент належить до множини, причому, якщо $\mu_A(x)=1$, елемент x чітко належить множині A , якщо ж $\mu_A(x)=0$, то елемент x чітко не належить цій множині. При $0 < \mu_A(x) < 1$ елемент належить множині нечітко. Чим вагомішим є значення ФН, тим більшою мірою елемент належить до множини. Іншими словами, нечітка множина відрізняється від звичайної тим, що для елементів x з M немає однозначної відповіді

«так-ні» щодо властивості G . Тоді нечітка підмножина A універсальної множини M визначається як множина упорядкованих пар

$$A = \{\mu_A(x)/x\},$$

де $\mu_A(x)$ – характеристична ФН, яка приймає значення в деякій цілком упорядкованій множині E (наприклад, $E = [0, 1]$).

Означення 21.4. Нечіткою підмножиною A довільної непорожньої множини M називається множина пар

$$A = \{(x, \mu_A(x))\}, \text{ де } x \in M, \mu_A(x) \in [0, 1].$$

Означення 21.5. Функцією належності нечіткої підмножини A є функція $\mu_A : E \rightarrow [0, 1]$, базовою множиною (або базовою шкалою) якої є E .

Означення 21.6. Ступенем належності елемента x нечіткій множині M називається значення $\mu_A(x)$, визначене для кожного $x \in M$.

Означення 21.7. Носієм нечіткої підмножини A називається чітка підмножина A^* множини M , що містить ті елементи з M , для яких $\mu_A(x) = 0$.

Отже, нечітка підмножина характеризується своєю ФН, яка визначає, якою мірою елемент базової множини належить до нечіткої підмножини.

Відтак, ФНН вказує ступінь/рівень належності елемента x підмножині A . Тоді E є множиною належностей, і якщо $E = \{0, 1\}$, то нечітку підмножину A можна розглядати як звичайну чітку множину.

Твердження 21.1. Носієм нечіткої підмножини завжди є чітка звичайна підмножина базової множини.

Приклад 21.1. Розглянемо як приклад формальне подання поняття “високий коефіцієнт завантаження обробного ресурсу”. Для цього насамперед визначимо ФН до відповідної нечіткої множини.

Очевидно, базовою множиною M є множина невід’ємних дійсних чисел. Функцію ФН можна задати аналітичною формулою (рис. 21.1):

$$\mu_A(x) = \begin{cases} 0, & x < 0.5; \\ 1x - 0.5 & 0.5 \leq x \leq 0.9; \\ 1, & x > 0.9. \end{cases}$$

Тоді відповіддю на запитання «Чи є високим коефіцієнт завантаження обробного ресурсу 0,82?» може бути відповідь «Ступінь належності 0,8», що можна проінтерпретувати як «досить висока».

Але цю саму нечітку множину можна задати і по-іншому, якщо явно перелічити деякі ступені належності, наприклад:

$$A = \{(0.5, 0), (0.6, 0.25), (0.7, 0.5), (0.8, 0.75), (0.9, 1)\}.$$

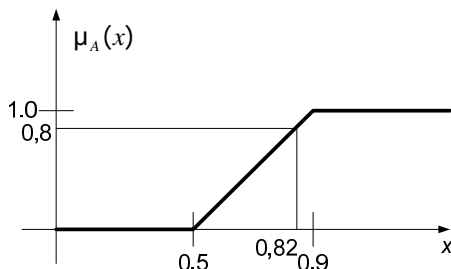


Рис. 21.1. Функція належності поняття “високий коефіцієнт завантаження обробного ресурсу”

Необхідно звернути увагу на такі важливі моменти:

- *функція ФН не може бути визначена однозначно, оскільки вона є дуже суб'єктивною; кожна людина може запропонувати свої значення цієї функції;*

- *самі ступені належності можна розглядати як розмиті. Уявимо собі, що дехто оцінив коефіцієнт завантаження обробного ресурсу (КЗОР) 0,85 см як високий зі ступенем належності 0.8. Якщо у нього спитати, чому 0.8, а не 0.85 чи 0.82, він не зможе чітко відповісти на це запитання. Тому замість конкретних числових значень ступенів належності можна говорити про *інтервали оцінок ступенів належності*. Якщо ступінь належності кожного елемента оцінюється інтервалом, говорять про *нечіткі множини другого роду* [46] з ФН $\mu_A : E \rightarrow [0, 1]^{[0, 1]}$. Такі міркування можна продовжити і далі.*

Втім, на практиці найчастіше користуються описаними раніше нечіткими множинами першого роду (з конкретними числовими значеннями ступенів належності);

- *незважаючи на те, що ФН є суб'єктивною оцінкою, її не можна вибирати довільно (так, для приклада з вагою об'єктів роботизації ФН повинна бути неспадною). Наприклад, у нашому випадку ФН, яка описує поняття «високий зріст», повинна бути неспадною: неможливо, щоб хтось оцінював КЗОР 0,65 більшим ступенем належності, аніж для КЗОР 0,85.*

Основні характеристики нечітких множин. Нехай $E=[0, 1]$ та A – нечітка підмножина з елементами універсальної множини M та множиною належностей E .

Означення 21.8. Величина $\sup_{x \in U} \mu_A(x)$ називається *висотою* нечіткої множини A .

Зазначимо таке:

- якщо $\sup_{x \in U} \mu_A(x) = \begin{cases} 1, & \text{то нечітка множина є нормальною;} \\ 0, & \text{то нечітка множина є субнормальною;} \end{cases}$
- нечітка множина є *порожньою*, якщо $\forall x \in M$ і спостерігається $\mu_A(x) = 0$;
- *непорожню субнормальну* множину можна нормувати за формулою:

$$\mu_A(x) := \mu_A(x) / \sup_{x \in U} \mu_A(x);$$

- нечітка множина *унімодальна*, тобто $\mu_A(x) = 1$ тільки на одному x з M ;
- *носієм* нечіткої множини A є звичайна підмножина із властивістю $\mu_A(x) > 0$, тобто *носії* $A = \{x / \mu_A(x) > 0\}$, $\forall x \in M$;
- елементи $x \in M$, для яких $\mu_A(x) = 0,5$, називаються *точками переходу* множини A .

Методи побудови функції належності нечітких множин.

Вирізняють *прямі* та *опосередковані* методи побудови ФН.

Як правило, *прямі методи* використовуються для вимірних понять (наприклад, швидкість, час, віддаль, тиск тощо) або якщо виокремлюються полярні значення. Тоді експерт задає для кожного $x \in M$ значення $\mu_A(x)$. Так, виділивши при характеристиці об'єкта набір його ознак і для кожної з них визначивши полярні значення, які відповідають значенням функції належності 0 або 1, і виходячи з отриманої шкали задають $\mu_A(x) \in [0, 1]$, формуючи векторну ФН $\{\mu_A(x_1), \mu_A(x_2), \dots, \mu_A(x_n)\}$. Зокрема, різновидом прямих методів побудови функції ФН є *прямі групові методи*, коли, наприклад, групі експертів показують конкретний об'єкт, і кожний з експертів повинен дати одну з двох відповідей: належить чи ні цей об'єкт до заданої множини. Кількість ствердних відповідей, поділена на загальну кількість експертів, і визначає значення функції ФН об'єкта до даної нечіткої множини.

Опосередковані (непрямі) методи визначення значень функцій ФН використовуються у тих випадках, коли немає вимірюваних елементарних властивостей, через які визначається нечітка множина.

Якби значення функцій ФН було відомим (наприклад, $\mu_A(x_i) \in w_i, i=1, 2, \dots, n$), то попарні порівняння можна подати матрицею відношень $\mathbf{A} = \{\alpha_{ij}\}$, де $\alpha_{ij} = w_i / w_j$ (операція ділення).

На практиці експерт сам формує матрицю \mathbf{A} , при цьому припускається, що діагональні елементи дорівнюють 1, а для елементів, симетричних відносно головної діагоналі, співвідношення мають вигляд $\alpha_{ij} = 1/\alpha_{ji}$, тобто якщо один елемент оцінюється в α разів більш значущим, ніж другий, то цей останній має бути в $1/\alpha$ разів більш значущим, ніж перший. У загальному випадку задача зводиться до пошуку вектора \mathbf{w} , що задовольняє рівнянню виду $\mathbf{A}\mathbf{w} = \xi_{\max} \mathbf{w}$, де ξ_{\max} – найбільше власне значення матриці \mathbf{A} . Оскільки матриця \mathbf{A} додатна за побудовою, розв'язок цієї задачі існує і є додатнім.

Нарешті, використовуються також *методи, які базуються на типових формах кривих* для ФН (зокрема, формі L - R -типу [31]) з уточненням їх параметрів згідно з експериментальними даними.

Операції над нечіткими множинами та нечіткі логічні виведення. Наведемо означення деяких операцій над нечіткими множинами A і B на універсальній множині M .

Означення 21.9. Рівними між собою називаються дві нечіткі підмножини A та B (позначається як $A = B$) базової множини M , якщо для будь-якого $x \in M$ виконується $\mu_A(x) = \mu_B(x)$.

Означення 21.10. Операція включення відбувається тоді, коли підмножиною іншої множини B є деяка нечітка множина A (позначається $A \subseteq B$), якщо для будь-якого $x \in M$ виконується $\mu_A(x) \leq \mu_B(x)$.

Означення 21.11. Перетином двох нечітких множин A і B (позначається $A \cap B$) базової множини M називається нечітка множина $D = A \cap B$ з функцією належності $\mu_D(x) = \min(\mu_A(x), \mu_B(x))$.

Означення 21.12. Об'єднанням двох нечітких множин A і B (позначається $A \cup B$) називається нечітка множина $D = A \cup B$ з функцією належності $\mu_D(x) = \max(\mu_A(x), \mu_B(x))$.

Означення 21.13. Доповненням нечіткої множини A називатимемо нечітку множину B , якщо $\mu_B(x) = 1 - \mu_A(x)$. Очевидно, що $(\overline{\overline{A}}) = A$ (тут доповнення визначено для $E = [0, 1]$, проте також очевидно, що його можна поширити на будь-яке упорядковане E).

Означення 21.14. Різницею двох нечітких множин A і B базової множини M є нечітка множина $C = A - B = A \cap \bar{B}$ з функцією належності $\mu_C(x) = \mu_{A \cap \bar{B}}(x) = \min(\mu_A(x), 1 - \mu_B(x))$.

Означення 21.15. Диз'юнктивною сумою нечітких множин A і B є нечітка множина

$$H = A \oplus B = (A - B) \cup (B - A) = (A \cap \bar{B}) \cup (\bar{A} \cap B),$$

для якої існує ФН

$$\mu_H(x) = \max \left\{ \left[\min(\mu_A(x), 1 - \mu_B(x)) \right]; \left[\min(1 - \mu_A(x), \mu_B(x)) \right] \right\}.$$

Для нечітких множин виконуються всі основні властивості звичайних чітких множин, крім двох: об'єднання $A \cup \bar{A}$ не обов'язково дорівнює базовій множині; перетин $A \cap \bar{A}$ не обов'язково дорівнює порожній множині.

Як бачимо, подання основних логічних операцій за допомогою характеристичних функцій застосовується відома з булівої алгебри двійкова логіка, що є основою всіх ЕОМ. Для графічної інтерпретації переведення основних логічних операцій у нечіткі множини наведемо спочатку значення двох множин A і B у формі характеристичних функцій 1 і 2 чітких множин і ФН 3 і 4 нечітких множин (рис. 21.2).

Пояснимо коротко окремі логічні операції:

- множина логічного об'єднання $D = A \cup B$ (АБО) обох множин A і B визначається за всіма тими елементами x , що належать A або B або обом множинам. Характеристичну функцію множини D , як це випливає з означення 21.12, можна в такий спосіб отримати формуванням максимуму (АБО) $\mu_M(x) = \max[\mu_A(x), \mu_B(x)]$ (крива 5 на рис. 21.2);

- множина логічного перетину $D = A \cap B$ (І) обох множин A і B визначається за всіма тими елементами x , що належать як A , так і B . Характеристичну функцію D можна за означенням 21.11 отримати формуванням мінімуму (І) $\mu_M(x) = \min[\mu_A(x), \mu_B(x)]$ (крива 6 на рис. 21.2);

- множина логічного доповнення $D = \bar{A}$ (НІ) до множини A визначається всіма тими елементами x , що не належать до множини A . Характеристичну функцію чіткої множини логічного доповнення D , згідно з означенням 21.13, можна подати у вигляді $\mu_B(x) = 1 - \mu_A(x)$ (крива 7 на рис. 21.2).

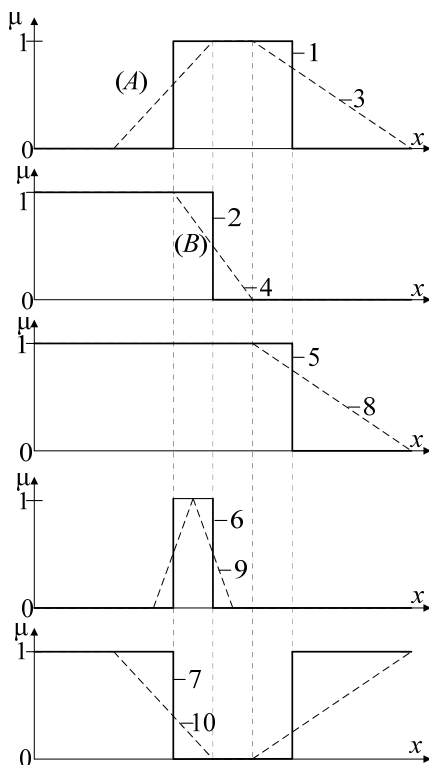


Рис. 21.2. Характеристичні функції множин A і B при двійковій логіці: m_A, m_B — графіки характеристичних функцій множин A і B при двійковій логіці (суцільні лінії 1 і 2) і відповідних ФН при фаззі-логіці (пунктирні лінії 3 і 4); графіки подання основних операцій при двійковій логіці (суцільні лінії 5, 6, 7) і відповідні ФН при фаззі-логіці (пунктирні лінії 8, 9, 10)

Оскільки характеристичну функцію чіткої множини можна розглядати як окремий випадок ФН нечіткої множини, наведені вище форми подання основних операцій з чіткими множинами можуть бути безпосередньо застосовані і для нечітких множин:

- для $D = A \cup B$ (АБО) маємо $\mu_{A \cup B}[\mu_A(x), \mu_B(x)]$ (крива 8 на рис. 21.2);
- для $D = A \cap B$ (І) маємо $\mu_{A \cap B}[\mu_A(x), \mu_B(x)]$ (крива 9 на рис. 21.2);

- для $D = \bar{A}$ (НІ) маємо $\mu_{\bar{A}}(x) = \mu_B(x) = 1 - \mu_A(x)$ (крива 10 на рис. 21.2).

Попарне порівняння кривих 5 і 8, 6 і 9, а також 7 і 10 наочно ілюструють процеси переходу від чіткої до фаззі-логіки.

Відзначимо також, що крім основних операцій для нечітких множин зберігаються в силі й інші відомі обчислювальні закони булівої алгебри, а саме:

- комутативність

$$A \cup (B \cap C) = (A \cup B) \cap C \text{ або } A \cap (B \cap C) = (A \cap B) \cap C;$$
- асоціативність

$$A \cup B = B \cup A \text{ або } A \cap B = B \cap A;$$
- дистрибутивність

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \text{ або } A \cap (B \cup C) = (A \cap B) \cup (A \cap C);$$
- $$\left. \begin{aligned} A \cup \emptyset &= A \\ A \cap \emptyset &= \emptyset \end{aligned} \right\}, \text{ де } \emptyset - \text{ порожня множина, тобто } \mu_{\emptyset}(x) = 0 \text{ для } \forall x \in M;$$
- $$\left. \begin{aligned} A \cap M &= A \\ A \cup M &= M \end{aligned} \right\}, \text{ де } M - \text{ універсальна множина;}$$
- $$\left. \begin{aligned} \overline{A \cap B} &= \bar{A} \cup \bar{B} \\ \overline{A \cup B} &= \bar{A} \cap \bar{B} \end{aligned} \right\} - \text{ формули де Моргана}$$

та ін. (див. розд. 2.1 [78]), крім неприйнятності співвідношень $A \cup \bar{A} = 1$, $A \cap \bar{A} = 0$.

Зауваження 21.1. Зазначимо, що введені вище операції над нечіткими множинами базуються на використанні операцій **max** і **min**. В теорії нечітких множин розглядаються питання побудови узагальнених, параметризованих операторів перетину, об'єднання та доповнення, які дають змогу враховувати різні змістовні відтінки відповідних їм операторів «І», «АБО», «НІ».

Один з підходів до узагальнення операторів перетину та об'єднання полягає у їх визначенні у класі трикутних норм і конорм, які випливають з наведених на рис. 1.5 (розд. 1.2) нормованих активіційних функцій (або ще функцій належності):

Означення 21.16. Трикутною нормою (*T-нормою*) є двомісна дійсна функція $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$, що задовольняє такі умови:

$$T(0, 0) = 0; T(\mu_A, 1) = \mu_A; T(1, \mu_A) = \mu_A - \text{обмеженість}; \quad (21.1)$$

$$T(\mu_A, \mu_B) \leq T(\mu_C, \mu_D), \text{ якщо } \mu_A \leq \mu_C, \mu_B \leq \mu_D - \text{монотонність}; \quad (21.2)$$

$$T(\mu_A, \mu_B) = T(\mu_B, \mu_A) - \text{комутативність}; \quad (21.3)$$

$$T(\mu_A, T(\mu_B, \mu_C)) = T(T(\mu_A, \mu_B), \mu_C) - \text{асоціативність}. \quad (21.4)$$

Прикладами T -норм можуть слугувати: $\min(\mu_A, \mu_B)$; \rightarrow добуток $\mu_A \cdot \mu_B$; $\max(0, \mu_A + \mu_B - 1)$;

Означення 21.17. Трикутною конормою (T -конормою) є двомісна функція $S : [0, 1] \times [0, 1] \rightarrow [0, 1]$ з наступними властивостями:

$$S(1, 1) = 1; S(\mu_A, 0) = \mu_A; S(0, \mu_A) = \mu_A - \text{обмеженість}; \quad (21.5)$$

$$S(\mu_A, \mu_B) \geq S(\mu_C, \mu_D), \text{ якщо } \mu_A \geq \mu_C, \mu_B \geq \mu_D - \text{монотонність}; \quad (21.6)$$

$$S(\mu_A, \mu_B) = S(\mu_B, \mu_A) - \text{комутативність}; \quad (21.7)$$

$$S(\mu_A, S(\mu_B, \mu_C)) = S(S(\mu_A, \mu_B), \mu_C) - \text{асоціативність}. \quad (21.8)$$

Прикладами T -конорм є: $\max(\mu_A, \mu_B)$; $\min(1, \mu_A + \mu_B)$.

Нечіткі та лінгвістичні змінні. Поняття нечіткої та лінгвістичної змінних застосовується при поданні об'єктів та явищ за допомогою нечітких множин.

Означення 21.18. Нечітка змінна характеризується тройкою параметрів $\langle a, X, A \rangle$, де a – ім'я змінної; X – універсальна множина (область визначення a); A – нечітка множина на X , що описує обмеження (тобто $\mu_A(x)$) на значення нечіткої змінної a .

Означення 21.19. Лінгвістична змінна (ЛЗ) характеризується набором параметрів $\langle b, T, X, S, G \rangle$, де b – ім'я лінгвістичної змінної (символ b може використовуватися як для назви самої змінної, так і для позначення усіх її значень. Наприклад, терм «незначне», який є значенням лінгвістичної змінної b = «зусилля захватного пристрою», одночасно може бути і нечіткою множиною S («незначне», що може впливати з контексту подання лінгвістичної змінної); T – множина значень лінгвістичної змінної (терм-множина), що є назвою нечітких змінних, областю визначення кожної з яких є множина S . Множина T називається базовою терм-множиною ЛЗ; S – семантична процедура, яка дає змогу перетворити кожне нове значення лінгвістичної змінної, утворене процедурою G , в нечітку змінну, тобто сформувати відповідну нечітку множину; G – синтаксична процедура, яка дає змогу оперувати елементами терм-множини T , зокрема генерувати нові терми (значення).

Означення 21.20. Розширеною терм-множиною ЛЗ називається множина $T \cup G(T)$, де $G(T)$ – множина згенерованих термів.

Приклад 21.2. Розглянемо приклад, за яким експерт визначає зусилля, яке необхідне захватному пристрою (ЗП), щоб утримувати крихкий об'єкт. При цьому він оперує поняттями «мале зусилля», коли об'єкт проковзує в ЗП, «нормальне зусилля», коли об'єкт утримується ЗП без пошкоджень поверхневого шару і руйнації об'єкта, і «велике зусилля», коли поверхневий шар об'єкта руйнується від тиску губок ЗП на поверхню цього об'єкта. Нехай також мінімальне зусилля ЗП визначається значенням 10 г/мм^2 , а максимальне — 100 г/мм^2 (див. рис. 21.3).

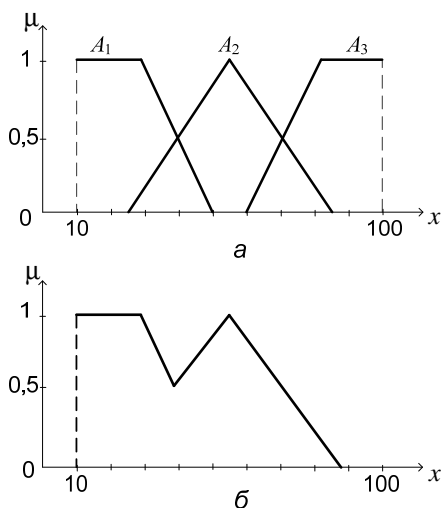


Рис. 21.3. Функції належності нечітких множин: A_1 =<мале зусилля>, A_2 =<нормальне зусилля>, A_3 =<велике зусилля> (а) та $A_1 \cup A_2$ =<мале або нормальне зусилля> (б)

Формалізацію такого подання можна виконати за допомогою такої лінгвістичної змінної $\langle b, T, X, S, G \rangle$, де b — зусилля ЗП; T — {«мале зусилля», «нормальне зусилля», «велике зусилля»}; X — $[10, 100]$; G — процедура утворення нових термів за допомогою зв'язок «і», «або» та модифікаторів типу «дуже», «ні», «слабо». Наприклад: «мале або нормальне зусилля», «дуже мале зусилля» тощо; S — процедура задавання на X — $[10, 100]$ нечітких підмножин A_1 =<мале зусилля>, A_2 =<нормальне зусилля>, A_3 =<велике зусилля>, а також нечітких множин для термів з $G(T)$ згідно з правилами трансляції нечітких зв'язок і модифікаторів [31] «і», «або», «дуже», «ні», «слабо» та інших над нечіткими множинами виду: $A \cap B$, $A \cup B$, \bar{A} , $\text{CON } A = A^2$, $\text{DIL } A = A^{0.5}$ тощо.

Треба зазначити, що поряд з наведеними вище базовими значеннями ЛЗ «зусилля»

$$(T = \{ \langle \text{мале зусилля} \rangle, \langle \text{нормальне зусилля} \rangle, \langle \text{велике зусилля} \rangle \})$$

можливі значення, які залежать від області визначення X . Нехай у розглянутому випадку значення ЛЗ «зусилля ЗП» можуть бути визначені як «близько 30 г/мм²», «близько 60 г/мм²», «близько 80 г/мм²», тобто у вигляді нечітких чисел.

Терм-множину та розширену терм-множину в умовах наведеного прикладу можна схарактеризувати функціями належності, поданими на рис. 21.3, а та 21.3, б відповідно.

21.4. Нечіткі числа

Означення 21.21. Нечіткі числа — це нечіткі змінні, визначені на числовій осі, тобто нечітке число визначається як нечітка множина A на множині дійсних чисел R з ФН $\mu_A(x) \in [0, 1]$, де $x \in R$.

Означення 21.22. Нечітке число A є нормальним, якщо $\max_x \mu_A(x) = 1$, та є опуклим, якщо для будь-яких $x \leq y \leq z$ виконується $\mu_A(x) \geq \mu_A(y) \wedge \mu_A(z)$.

Означення 21.23. Множина α -рівня нечіткого числа A визначається як $A_\alpha = \{x / \mu_A(x) \geq \alpha\}$.

Означення 21.24. Носієм нечіткого числа A називається підмножина $C_A \subset R$, якщо $C_A = \{x / \mu_A(x) > 0\}$.

Означення 21.25. Нечітке число A є унімодальним, якщо умова $\max_x \mu_A(x) = 1$ справедлива тільки для однієї точки дійсної осі.

Означення 21.26. Нечітким нулем є опукле нечітке число A , якщо справедливе $\mu_A(0) = \sup_x (\mu_A(x))$.

Означення 21.27. Нечітке число A є унімодальним, якщо $\forall x \in C_A, x > 0$, та є від'ємним, якщо $\forall x \in C_A, x < 0$.

Операції над нечіткими числами розглянуто у [31].

21.5. Нечітке логічне виведення

Спрощена постановка задачі. Найтиповіша задача логічного виведення за умов нечіткості формулюється так :

«дано продукційне правило: “Якщо A , То B ; спостерігається A^* (A в певній мірі)”.

Яким повинно бути B ?»

Формальніше, задані носії C і D та їх нечіткі підмножини A і B . Задано або елемент $c \in C$, або множина $A' \subseteq C$. Потрібно визначити елемент $d \in D$, що являє собою виведення системи, яке визначає результат застосування нечіткого продукційного правила.

Приклад 21.3. Дано нечітке продукційне правило:

“Якщо промисловий робот ПР інтенсивно задіяний у виробничому процесі, то в нього високий показник споживання електроенергії (ПСЕ)”.

Як множину C ми розглядаємо множину чисел, що визначають кількість годин на тиждень, які ПР може бути задіяним у виробничому процесі. Візьмемо C як діапазон чисел від 0 до 40 і для простоти подальших міркувань обмежимося невеликою кількістю можливих значень:

$$C = \{0, 6, 12, 18, 24, 30, 35, 40\}.$$

Аналогічно, якщо за максимальну оцінку D показника ПСЕ для ПР прийняти 10, а діапазон його значень взяти від 0.3 до 1.0, то знову обмежимося невеликим набором можливих значень:

$$D = \{3.0; 5.0; 6.5; 8.0; 9.0; 10.0\}.$$

Задамо ФН для нечітких множин A (*“інтенсивно задіяний у виробничому процесі”*) та B (*“високий ПСЕ”*) таким чином:

$$A = \{(6, 0); (12, 0.2); (18, 0.4); (24, 0.6); (30, 0.8); (35, 0.9); (40, 1.0)\};$$

$$B = \{(3.0, 0.1); (5.0, 0.2); (6.5, 0.4); (8.0, 0.7); (9.0, 0.85); (10.0, 1.0)\}.$$

Нехай явним чином задана кількість годин функціонування ПР у виробничому процесі, або ступінь належності, що визначає, чи інтенсивно він задіяний у виробничому процесі (що еквівалентно, бо, знаючи конкретну кількість годин, ми можемо визначити відповідний ступінь належності). Нехай ступінь належності дорівнює α . Тоді для виведення можна застосувати *метод простої підстановки нечіткого значення*, відповідно до якого d вибирається з умови:

$$\mu_B(d) = \alpha.$$

Нехай у нашому прикладі дано, що *ПР задіяний у виробничому процесі 29 годин*. При цьому ступінь належності дорівнює 0.77, і система повинна дійти висновку, що за такої інтенсивності використання ПР у виробничому процесі його ПСЕ дорівнюватиме 8.5 (*“досить великий”*). Оскільки в нашій спрощеній множині D не виявилося значення точно з таким ступенем належності, величину ПСЕ визначено шляхом інтерполяції (або вибором наближеного значення).

Узагальнена постановка задачі нечіткого виведення. Тепер розглянемо більш детально процедуру нечіткого виведення. Механізм нечітких виведень, який використовується в різного роду експертних і керуючих системах, формується спеціалістами предметної галузі у вигляді сукупності нечітких предикатних правил виду:

$$П_1: \text{ЯКЩО } x \in A_1, \text{ ТО } y \in B_1;$$

$$П_2: \text{ЯКЩО } x \in A_2, \text{ ТО } y \in B_2;$$

...

$$П_n: \text{ЯКЩО } x \in A_n, \text{ ТО } y \in B_n,$$

(21.9)

де x – вхідна змінна (ім'я відомих значень даних); y – змінна виходу (ім'я значення даних, яке буде обчислене); A і B – функції належності, визначені відповідно на x і y .

Отже, знання експерта $A \rightarrow B$ відображує нечітке причинне відношення передумови і висновку, тому його називають *нечітким відношенням* і визначають як $R = A \rightarrow B$, де « \rightarrow » – нечітка імплікація.

Відношення R можна розглядати як нечітку підмножину прямого добутку x та y повної множини передумов X і виведень Y . Тоді процес отримання нечіткого результату виведення B' із застосуванням цього спостереження A' та знання $A \rightarrow B$ можна подати у вигляді композиційного правила *нечіткий «modus ponens»*:

$$B' = A' \bullet R = A' \bullet (A \rightarrow B), \quad (21.10)$$

де « \bullet » — операція згортання (композиції).

Операції композиції та імплікації в алгебрі нечітких множин можна реалізувати по-різному (з різними результатами), проте у будь-якому разі загальне логічне виведення здійснюється за такі чотири етапи [4; 31]:

1. Введення нечіткості (фаззифікація) — для визначення ступеня істинності кожної передумови кожного правила застосування функцій належності, визначених на вхідних змінних, до їх фактичних значень.

2. Логічне виведення — застосування обчисленого значення істинності для передумов кожного правила до виведень кожного правила. Це приводить до однієї нечіткої множини, яка буде призначена кожною змінною виведення для кожного правила. Як правила логічного виведення зазвичай використовуються тільки операції **min** (МІНІМУМ) або **prod** (ДОБУТОК). У логічному виведенні **min** функція належності виведення «відтинається» за висотою, відповідною обчисленому ступеню істинності передумови правила (так звана нечітка логіка «I»). В логічному виведенні **prod** функція належності виведення масштабується за допомогою обчисленого ступеня істинності передумови правила.

3. Композиція — об'єднання разом усіх нечітких підмножин, призначених до кожної змінної виведення (в усіх правилах), для формування однієї нечіткої підмножини для всіх змінних виведення. При такому об'єднуванні зазвичай використовуються операції **max** (МАКСИМУМ) або **sum** (СУМА). При композиції **max** комбіноване виведення нечіткої підмножини конструюється як покроковий максимум за всіма нечіткими підмножинами (так звана нечітка логіка

«АБО»). При композиції **sum** комбіноване виведення нечіткої підмножини формується як покрокова сума за всіма нечіткими підмножинами, призначеними змінною виведення правилами логічного виведення.

Властивості max-min-композиції. Операція **max-min**-композиції $R_3 \bullet (R_2 \bullet R_1) = (R_3 \bullet R_2) \bullet R_1$ асоціативна, тобто дистрибутивна відносно об'єднання, але недистрибутивна відносно перетину:

$$\begin{aligned} R_3 \bullet (R_2 \cup R_1) &= (R_3 \bullet R_2) \cup (R_3 \bullet R_1); \\ R_3 \bullet (R_2 \cap R_1) &\neq (R_3 \bullet R_2) \cap (R_3 \bullet R_1). \end{aligned} \quad (21.11)$$

Крім того, для **max-min**-композиції виконується така важлива властивість:

$$\text{ЯКЩО } R_1 \subset R_2, \text{ ТО } R_3 \bullet R_1 \subset R_3 \bullet R_2. \quad (21.12)$$

Властивості max--композиції.* Для **max-min**-композиції відношень R_1 і R_2 у виразі $\mu_{(R_1 \circ R_2)}(x, z) = \bigvee_y [\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z)]$ операцію « \wedge » можна замінити будь-якою іншою, для якої виконуються ті ж самі обмеження (наприклад, асоціативність, монотонність), що й для « \wedge ». Тоді маємо:

$$\mu_{(R_1 * R_2)}(x, z) = \bigvee_y [\mu_{R_1}(x, y) * \mu_{R_2}(y, z)]. \quad (21.13)$$

Зокрема, операція « \wedge » може бути замінена алгебраїчним добутком, і тоді кажуть про **max-prod**-композицію.

4. Зведення до чіткості (дефаззифікація) — перетворення (при необхідності) нечіткого набору виведень до чіткої величини (числа). Існує багато методів зведення до чіткості, деякі з алгоритмів яких розглянуто далі. Але спочатку розглянемо приклад деякої системи, яка подається нечіткими правилами.

Приклад 21.4. Нехай деяка система задається нечіткими правилами вигляду (21.9):

$$\begin{aligned} \Pi_1: \text{ЯКЩО } x \in A, \text{ ТО } w \in D; \\ \Pi_2: \text{ЯКЩО } y \in B, \text{ ТО } w \in E; \\ \Pi_3: \text{ЯКЩО } z \in C, \text{ ТО } w \in F, \end{aligned} \quad (21.14)$$

де x, y, z — імена вхідних змінних; w — ім'я змінної виведення; A, B, C, D, E, F — задані функції належності (наприклад, трикутної форми, див. розд. 1.2, рис. 1.5, ∂).

Процедура отримання логічного виведення ілюструється рис. 21.4. Припускається, що задано конкретні (чіткі) значення вхідних змінних: x_0, y_0, z_0 .

На *першому етапі* (фаззифікація) на основі значень i , виходячи з фун-

кцій належності A, B, C , визначаються ступені істинності $\alpha(x_0), \alpha(y_0), \alpha(z_0)$ для передумов кожного з трьох наведених правил (21.14).

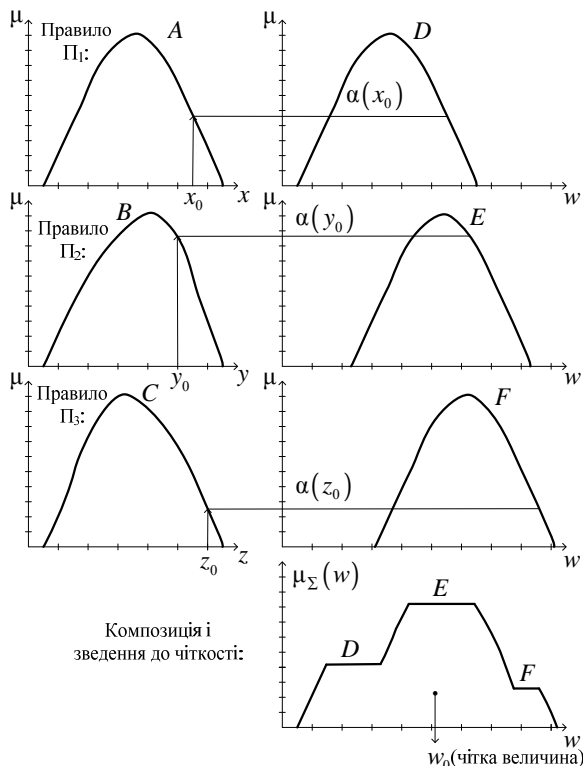


Рис. 21.4. Ілюстрація процедури логічного виведення

На *другому етапі* (логічне виведення) відбувається «відтинання» функцій належності виведень D, E, F правил (21.14) на рівнях $\alpha(x_0), \alpha(y_0), \alpha(z_0)$.

На *третьому етапі* (композиція) розглядаються функції належності, зрізані на попередньому етапі, і виконується їх об'єднання з використанням операції **max**, в результаті чого одержуємо комбіновану нечітку підмножину, яка подається функцією належності $\mu_\Sigma(w)$ і відповідає логічному виведенню для вихідної змінної w .

На *четвертому етапі* (дефаззифікація) визначається, за необхідності, чітке значення вихідної змінної, що може реалізовуватися із застосуванням, наприклад, *центроїдного методу*: чітке значення вихідної змінної визначається як центр ваги для кривої $\mu_\Sigma(w)$:

$$w_0 = \frac{\int_{\Omega} w \mu_\Sigma(w) dw}{\int_{\Omega} \mu_\Sigma(w) dw}. \quad (21.15)$$

Якщо проінтерпретувати наведені в *прикладі 21.4* результати *прикладом 21.2* з визначенням з виразу (21.15) чіткого значення w_0 для нормального утримання об'єкта в захватному пристрої, то це значення лежить у площині функції належності E (рис. 21.4).

21.5.1. Модифіковані алгоритми нечіткого виведення

Для розгляду найбільш уживаних модифікацій алгоритмів нечіткого виведення спростимо задачу, вважаючи, що базу знань утворюють тільки два нечітких правила виду:

$$\begin{aligned} \text{П}_1: & \text{ЯКЩО } x \in A_1 \text{ і } y \in B_1, \text{ ТО } z \in C_1; \\ \text{П}_2: & \text{ЯКЩО } x \in A_2 \text{ і } y \in B_2, \text{ ТО } z \in C_2, \end{aligned} \quad (21.16)$$

де x і y – імена вхідних змінних; z – ім'я змінної виведення; $A_1, A_2, B_1, B_2, C_1, C_2$ — деякі задані функції належності.

При цьому чітке значення z_0 необхідно визначити на основі наведеної інформації і чітких знань x_0 і y_0 .

Алгоритм Mamdani. Цей алгоритм відповідає вже розглянутому *прикладу 21.4* і проілюстрований рис. 21.4. Математично для розглянутої ситуації його було подано чотирма етапами:

- *введення нечіткості* (фаззифікація) з визначенням ступеня істинності для передумов кожного правила: $A_1(x_0), A_2(x_0), B_1(y_0), B_2(y_0)$;
- *логічне виведення* із знаходженням рівнів «відтинання» для передумов кожного з правил (із застосуванням операції **min**):

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0); \quad \alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

де через « \wedge » позначено операцію логічного мінімуму **min**. Далі визначаються «відтяти» функції належності:

$$C'_1(z) = (\alpha_1 \wedge C_1(z)); \quad C'_2(z) = (\alpha_2 \wedge C_2(z));$$

- *композиція* з об'єднанням знайдених відтятих функцій із застосуванням операції **max** (яку надалі позначатимемо як $<\vee>$), що дозволяє отримати підсумкову нечітку підмножину для змінної виходу з функцією належності:

$$\mu_{\Sigma}(z) = C(z) = C'_1(z) \vee C'_2(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z));$$

- *зведення до чіткості* з визначенням чіткого значення z_0 , наприклад, центроїдним методом.

Алгоритм Tsukamoto. Для цього алгоритму вихідні посилання такі самі, що й для алгоритму Mamdani, але тут припускається, що функції $C_1(z)$ і $C_2(z)$ є монотонними. Тоді алгоритм Tsukamoto визначається такою послідовністю:

- *введення нечіткості* (як і в алгоритмі Mamdani);
- *нечітке виведення* відбувається в два етапи. Спочатку визначаються рівні «відтинання» α_1 і α_2 (як в алгоритмі Mamdani), а потім, розв'язуючи рівняння

$$\alpha_1 = C_1(z) \text{ і } \alpha_2 = C_2(z),$$

знаходяться чіткі значення z_1 і z_2 для кожного вихідного правила;

- *зведення до чіткості* значень змінної виведення (як зважене середнє z_1 і z_2):

$$z_0 = (\alpha_1 z_1 + \alpha_2 z_2) / (\alpha_1 + \alpha_2), \quad (21.17)$$

або для загального випадку (дискретний варіант центроїдного методу):

$$z_0 = \left(\sum_{i=1}^n \alpha_i z_i \right) / \left(\sum_{i=1}^n \alpha_i \right). \quad (21.18)$$

Приклад 21.5. Нехай задано (рис. 21.5):

ступені істинності для передумов кожного правила (21.16)

$$A_1(x_0) = 0,8; \quad A_2(x_0) = 0,7; \quad B_1(y_0) = 0,4; \quad B_2(y_0) = 0,9,$$

відповідні рівні «відтинання»:

$$\alpha_1 = \min\{A_1(x_0); B_1(y_0)\} = \min\{0,8; 0,4\} = 0,4,$$

$$\alpha_2 = \min\{A_2(x_0); B_2(y_0)\} = \min\{0,7; 0,9\} = 0,7,$$

а також значення $z_1=4,5$ і $z_2=2,5$, які знайдено в результаті розв'язання рівнянь $C_1(z_1)=0,4$ і $C_2(z_2)=0,7$ (див. рис. 21.5).

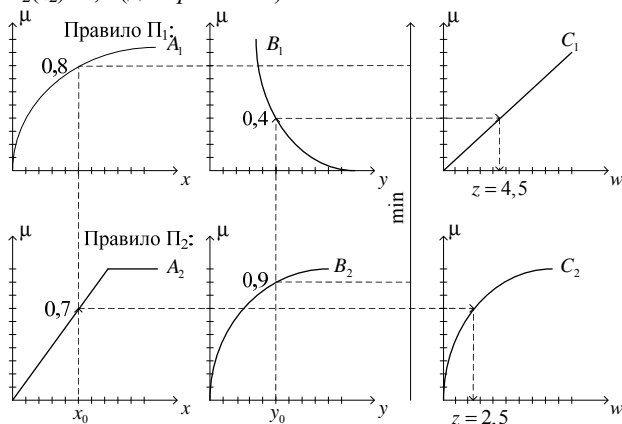


Рис. 21.5. Ілюстрація до алгоритму Tsukamoto

Тоді в результаті дефазифікації чітке значення змінної виведення, обчислене за виразом (21.17), має вигляд:

$$z_0 = (\alpha_1 z_1 + \alpha_2 z_2) / (\alpha_1 + \alpha_2) = (0,4 \cdot 4,5 + 0,7 \cdot 2,5) / (0,4 + 0,7) \cong 3,23.$$

Алгоритм Sugeno. Цей алгоритм містить введення нечіткості (як в алгоритмі Mamdani) та нечітке виведення із знаходженням:

- рівнів «відтинання»:

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0) \text{ і } \alpha_2 = A_2(x_0) \wedge B_2(y_0);$$

- індивідуальних виходів правил:

$$z_1^* = a_1 x_0 + b_1 y_0; \quad z_2^* = a_2 x_0 + b_2 y_0;$$

- чіткого значення змінної виходу:

$$z_0 = (a_1 z_1^* + a_2 z_2^*) / (a_1 + a_2),$$

що ілюструється рис. 21.6.

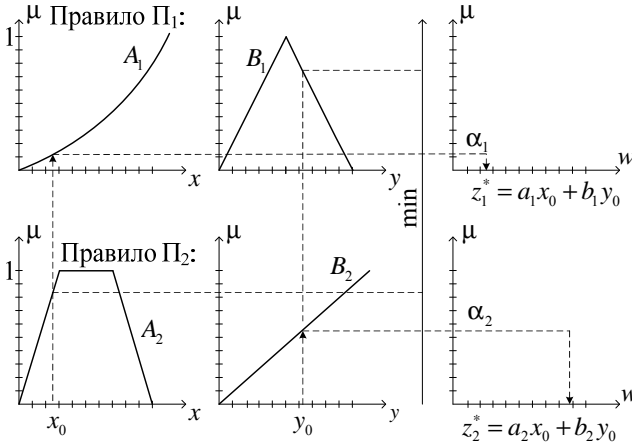


Рис. 21.6. Ілюстрація до спрощеного алгоритму Sugeno нечіткого логічного виведення

21.6. Методи зведення до чіткості

Вище розглядався *центроїдний метод*, який визначається залежностями (21.17) для загального випадку і (21.18) — для дискретного варіанту. Тепер наведемо ще деякі часто вживані методи.

Метод першого максимуму. Чітка величина виведення визначається як найменше значення, за яким досягається максимум підсумкової нечіткої множини

$$z_0 = \min \left\{ z \mid C(z) = \max_u C(u) \right\}. \quad (21.19)$$

Метод першого максимуму ілюструється рис. 21.7, а.

Метод середнього максимуму. Чітке значення за цим методом, зокрема для дискретного варіанта (C є дискретним) знаходять за формулою (рис. 21.7, б):

$$z_0 = \frac{1}{n} \sum_{j=1}^n z_j. \quad (21.20)$$

Метод критеріального максимуму. За цим методом, на відміну від (21.19), чітке значення вибирається довільно серед множини елементів, для яких C сягає максимуму:

$$z_0 \in \left\{ z \mid C(z) = \max_u C(u) \right\}. \quad (21.21)$$

Метод висотної дефазифікації. Для цього методу характерним є те, що елементи області визначення Ψ , для яких значення функції належності менше, ніж деякий рівень «відтинання» α , не враховуються, і чітке значення розраховується за виразом:

$$z_0 = \left(\int_{C_\alpha} z \cdot C(z) dz \right) / \left(\int_{C_\alpha} C(z) dz \right), \quad (21.22)$$

де C_α – нечітка множина α -рівня.

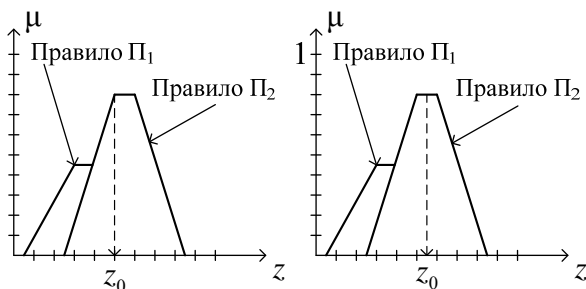


Рис. 21.7. Ілюстрація до метода зведення до чіткості: *а* — перший максимум; *б* — середній максимум

21.7. Ненадійні знання та виведення

На практиці в задачах, які розв'язуються інтелектуальними роботами, здебільшого рішення приймаються в умовах *невизначеності*. Остання може мати різну природу, тому застосовуються й різні способи роботи з невизначеними даними та значеннями: імовірнісна Байєсівська логіка, нечітка логіка, багатозначна логіка, логіка умовчання, робастні методи оцінювання. Розглянемо детальніше реалізацію міркувань в умовах двох основних типів невизначеності: *випадковості* та *нечіткості* [38; 69], а також ознайомимось з посиланнями *логіки умовчання* та *теорії можливостей* (ця теорія має на меті розглянути з єдиних позицій модальні, недостовірні та нечіткі знання).

Базові посилання імовірнісної логіки. В усіх випадках недостовірність логічного виведення тісно пов'язано з апаратом теорії ймовірностей. Тому для розуміння цього розділу потрібно навести базові поняття з теорії ймовірностей.

Означення 21.28. *Ймовірність події* A (позначається $P(A)$) є мірою достовірності деякої події, за якої завжди виконується властивість $0 \leq P(A) \leq 1$. При цьому *неможлива подія* (подія, яка ніколи не

може відбутися) має ймовірність 0, а ймовірність *достовірної події* дорівнюватиме 1.

Дамо більш формалізовані визначення.

Розглянемо *множину елементарних наслідків* Ω .

Означення 21.29. *Елементарні наслідки* — це можливі наслідки деякого експерименту, при цьому ці наслідки вважаються *рівно-можливими* та *взаємовиключними*.

Тоді будь-яка подія A розглядається як підмножина множини елементарних наслідків.

Твердження 21.2. *Подія A відбувається, якщо існує будь-який елементарний наслідок, що входить до множини A .*

Якщо елементарний наслідок входить до множини A , кажуть, що він є *сприятливим* для події A .

Твердження 21.3. *Ймовірність події A визначається як відношення кількості сприятливих наслідків до загальної кількості елементарних наслідків.*

На основі цього твердження в ряді випадків можна безпосередньо розрахувати ймовірності.

Подією, *протилежною* до події A (позначається \bar{A}), називається та, яка полягає в тому, що подія A не відбувається. На основі класичного визначення ймовірностей легко переконатися в тому, що

$$P(\bar{A}) = 1 - P(A).$$

Частотне визначення ймовірності. Нехай проводиться серія однотипних експериментів і нехай $K_A(n)$ — кількість випадків, в яких відбулася подія A , якщо було проведено n експериментів. Тоді

$$P(A) = \lim_{n \rightarrow \infty} [K_A(n) / n].$$

Наприклад, якщо проводиться 1000 експериментів, а ймовірність події A дорівнює 0.3, то подія A настане приблизно в 300 випадках. При цьому зі зростанням кількості експериментів частина випадків, в яких відбувається ця подія, щораз більше наближається до ймовірності.

Означення 21.30. *Ймовірність події* — це частота появи цієї події.

Ймовірнісна Байєсівська логіка. Випадкова невизначеність підпорядковується законам Байєса. Так, при функціонуванні виробничих дільниць не обійтись без відмов (виходу з ладу окремих елементів), можливих перебоїв у постачанні деталей або комплектуючих виробів. Природно, на стадії проектування та розроблення системи управління такими дільницями апріорі невідомо, який елемент і коли вийде з ладу, яких деталей і коли не вистачатиме.

Ймовірність прийняття рішення визначається за заданою сукупністю поточних даних, виходячи з апіорної ймовірності рішення (чи факту) й умовних ймовірностей, що пов'язують евристичні знання та спостереження з цими фактами.

Теорема Байєса записується у вигляді

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}, \quad (21.23)$$

де $P(A/B)$, $P(B/A)$ – умовна (або апостеріорна) ймовірність події (A/B) за наявності події (B/A); $P(A)$, $P(B)$ – безумовна (апіорна) ймовірність події (A/B) за відсутності супровідних умов.

Методи, які розроблено на основі байєсівського підходу, ґрунтуються на припущенні, що для будь-якої події існує (нехай дуже незначна) апіорна ймовірність того, що вона є істинною. Це дає змогу виконувати обчислення.

Якщо апіорно приймається ймовірність справедливості деякої гіпотези, то повинна існувати деяка сукупність даних, що підтверджують це припущення. Якби таких даних не було, то ця ймовірність завжди залишалася б незмінною. Проте за наявністю пов'язаних із подією даних і відомостей апіорну ймовірність можна модифікувати для одержання вже апостеріорної ймовірності тієї самої гіпотези з урахуванням нових поточних даних.

Формула (21.23) називається *формулою Байєса* і має дуже велике значення, оскільки дає змогу обчислювати *апостеріорні* ймовірності $P(B/A)$ “гіпотез” B за умови, що подія A відбулася через апіорні ймовірності $P(A/B)$. Більшість методик неточного логічного виведення так чи інакше пов'язані з формулою повної ймовірності та формулою Байєса.

21.8. Нечітка логіка в задачах робототехніки

Невизначеність другого типу пов'язана з поняттям нечіткості [39]. Розроблення й широке використання апарату теорії нечітких множин пов'язані з прагненням формально описувати лінгвістичні поняття, якими оперує людина в процесі прийняття рішень, а також імітувати міркування на підставі тих категорій і правил, на які вона спиралася. Багато понять і правил нечіткої логіки є узагальненням або розвитком логіки предикатів.

Як вже наголошувалося, створення апарату теорії нечітких множин стало потужним інструментом при вирішенні широкого кола проблем, в яких важливе місце займають суб'єктивні знання, що важко формалізуються.

Нехай, наприклад, треба синтезувати систему керування транспортним роботом – електророботом (ЕРК), який повинен зупинятися при зустрічі з перепоною недалеко від неї. Або нехай потрібно розробити алгоритм керування роботом, що відбраковує заготовки циліндричної форми, довжина яких є значно більшою (чи меншою), ніж задана. В наведених прикладах невизначеність зумовлена нечіткими поняттями «недалеко» і «значно більшою», «значно меншою». Такого роду *невизначеність називається лінгвістичною*.

Лінгвістична змінна може бути подана набором (X, T, U, G, M) , де X – назва змінної; T – терм-множина змінної X , тобто множина всіх назв лінгвістичних значень змінної X (до того ж кожне з таких значень є нечіткою змінною X із значеннями з універсальної множини U при базовій змінній u); G – синтаксичне правило, що зумовлює назву значення змінної X ; M – семантичне правило, яке ставить у відповідність кожній нечіткій змінній X зміст $M(X)$. Конкретна назва X , яка зумовлена синтаксичним правилом G , називається *термом*.

У свою чергу, нечітка множина M визначається як деяка множина значень X з функцією належності $\mu_M(X)$, що набуває значень в інтервалі $[0,1]$:

$$M = \int_{x \in X} \mu_M(X) / X.$$

Для кожного терму з терм-множин, які визначено для лінгвістичних змінних, будується відповідна нечітка множина M із своїм носієм, що визначається після вивчення експертних оцінювань (ЕО). При цьому під *носієм нечіткої множини* розуміють множину X' таку, що

$$X' = \{x / \mu_M(x) > 0, x \in X'\}.$$

На наступному етапі будується набір правил, які описують, наприклад, технологічний процес. Ці правила мають вигляд

$$\text{ЯКЩО } A=N, \text{ ТО } B=M, \quad (21.24)$$

де A, B – лінгвістичні змінні; N, M – їх терми.

На основі набору правил будується матриця нечітких відношень. Нечітке бінарне відношення R є підмножиною декартового добутку $X \times Y$, тобто

$$R = \int \mu_R(x, y) / (x, y), (x, y) \in X \times Y. \quad (21.25)$$

Далі побудовані матриці (21.25) об'єднуються в матрицю

$$\mu_R = \max(\mu_{R_1}, \dots, \mu_{R_n}). \quad (21.26)$$

Для організації нечіткого умовного логічного виведення застосовуються композиційні правила, які охоплюють в основному три види умовних пропозицій:

$P1 = \text{Якщо } x \in A, \text{ то } y \in B;$

$P2 = \text{Якщо } x \in A, \text{ то } y \in B, \text{ інакше } C;$

$P3 = \text{Якщо } x_1 \in A_1 \text{ і } x_2 \in A_2 \text{ і } \dots, \text{ і } x_n \in A_n, \text{ то } y \in B.$

Концептуальною основою формалізації правил нечіткого умовного виведення є так зване правило «відділення» («Modus Ponens», див. (2.11) розд. 2.1 [78]), що формально свідчить: $\langle\langle \text{якщо } (a \rightarrow b) \text{ істинне і } a \text{ істинне, то } b \text{ істинне} \rangle\rangle$. Інакше це правило називається *першою формою гіпотетичного силісму* (або *правилом виведення*).

Означення 21.31. Під силісмом розуміють дедуктивне виведення, в якому одна думка є необхідним наслідком двох інших.

У свою чергу, методологічною основою такої формалізації є запропоноване Л. Заде [22] так зване композиційне правило виведення, в якому логічними передумовами і наслідками є умовні вирази, що включають нечіткі концепції. Дещо пізніше Мамдані також сформулював своє правило виведення, яке, як і правило Л. Заде, було розроблене для умовного виразу вигляду $P1$. Іншими словами, розглядається нечітке умовне виведення такої форми:

Передумова 1: Якщо $x \in A$, то $y \in B$;

Передумова 2: $x \in A'$;

Наслідок: $y \in B'$.

Тут A і A' – нечіткі концепції, подані як нечіткі множини в універсумі U ; B – нечітка концепція або нечітка множина в універсумі V . Звідси B' є наслідком, поданим як нечітка множина у V . Використовуючи правило $P1$ для отримання логічного висновку B' , передумови 1 і 2 приводять до бінарного нечіткого відношення вигляду $R(A_1(x), A_2(y))$ і унарного нечіткого відношення вигляду $R(A_1(x))$. Множини $A_1(x)$ і $A_2(y)$ визначаються атрибутами x і y , які набувають значення з універсумів U і V відповідно. При цьому

$$R(A_1(x)) = A, \quad (21.27)$$

а $R(A_1(x), A_2(y))$, згідно з правилами виведення Заде–Мамдані, визначаються так:

$P1$. Максимінне правило умовного виразу

$$R_m(A_1(x), A_2(y)) = (A \times B) \cup (\bar{A} \times V), \quad (21.28)$$

де \times, \cup, \bar{A} – декартовий добуток, об'єднання та інверсія відповідно.

P2. Арифметичне правило умовної пропозиції

$$R_c(A_1(x), A_2(y)) = (\bar{A} \times V) \oplus (U \times B), \quad (21.29)$$

де \oplus – операція «обмежена сума».

P3. Міні-функціональне правило умовної пропозиції

$$R_c(A_1(x), A_2(y)) = (A \times B). \quad (21.30)$$

Логічний наслідок $R(A_2(y))$, який подається як B' в (21.26), можна отримати з урахуванням (21.27) як

$$R(A_2(y)) = A' \circ (A \times B) \cup (\bar{A} \times V);$$

$$R(A_2(y)) = A' \circ (\bar{A} \times V) \oplus (\bar{U} \times B),$$

Або

$$R(A_2(y)) = A' \circ (A \times B),$$

де \circ — операція максимінної композиції нечітких множин. Зазначимо, що з використанням цих же правил можна запропонувати правила для умовної пропозиції вигляду P2 [2], тобто

$$\begin{aligned} P4. R_4(A_1(x), A_2(x)) &= \\ &= \left[(\bar{A} \times V) \oplus (U \times B) \right] \cap \left[(A \times V) \oplus (U \times C) \right]; \end{aligned} \quad (21.31)$$

$$\begin{aligned} P5. R_5(A_1(x), A_2(x)) &= \\ &= \left[(\bar{A} \times V) \cup (U \times B) \right] \cap \left[(A \times V) \cup (U \times C) \right]; \end{aligned} \quad (21.32)$$

$$P6. R_6(A_1(x), A_2(x)) = (A \times B) \cup (\bar{A} \times C), \quad (21.33)$$

або правила для умовної пропозиції вигляду P3:

$$P7. R_7(A_1(x), A_2(x)) = \left[\bigcap_{i=1, n} (\bar{A}_i \times V) \right] \oplus (U \times B); \quad (21.34)$$

$$P8. R_8(A_1(x), A_2(x)) = \left[\bigcap_{i=1, n} (\bar{A}_i \times V) \right] \cup (U \times B); \quad (21.35)$$

$$P9. R_9(A_1(x), A_2(x)) = \left[\bigcap_{i=1, n} \bar{A}_i \right] \times B. \quad (21.36)$$

Для з'ясування питання про те, наскільки запропоновані правила можуть служити для моделювання поведінки робота, можна використовувати деякі критерії для нечіткого умовного виведення, викладені в [22]:

Критерій Кр 1.	$\left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in A, \\ \text{Передумова 2: } x \in A'; \\ \text{Наслідок: } y \in B'. \end{array} \right.$
Критерій Кр 2.	$\left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in A, \text{ то } y \in B, \\ \text{Передумова 2: } x \in \text{ дуже } A; \\ \text{Наслідок: } y \in \text{ дуже } B. \end{array} \right.$
Критерій Кр 3.	$\left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in A, \text{ то } y \in B, \\ \text{Передумова 2: } x \in \text{ не дуже } A; \\ \text{Наслідок: } y \in \text{ не дуже } B. \end{array} \right.$
Критерій Кр 4.	$\left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in A, \text{ то } y \in B, \\ \text{Передумова 2: } x \in \text{ більш менш } A; \\ \text{Наслідок: } y \in \text{ більш менш } B. \end{array} \right.$
Критерій Кр 4-1.	$\left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in A, \text{ то } y \in B, \\ \text{Передумова 2: } x \in \bar{A}; \\ \text{Наслідок: } y \in \text{ невідомо.} \end{array} \right.$
Критерій Кр 4-2.	$\left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in A, \text{ то } y \in B, \\ \text{Передумова 2: } x \in \bar{A}; \\ \text{Наслідок: } y \in \bar{B}. \end{array} \right.$

У [22] було також показано, що правила (21.28)–(21.30) не завжди задовольняють ці критерії.

У праці [2] було описано умови незадоволення критеріїв максимінного правила $P1$.

Умовою ж для вибору логіки при розробленні правил нечіткого умовного виведення є такі очевидні міркування.

Нехай нечіткі множини A з U і B з V задано у вигляді

$$A = \int_U \mu_A(u) / u; \quad B = \int_V \mu_B(v) / v.$$

Тоді, як було сказано вище, умовна пропозиція $P1$ може бути переведена у бінарне нечітке відношення $R(A_1(x), A_2(y))$ адаптацією операції імплікації в багатозначній логіці, тобто

$$R(A_1(x), A_2(x)) = A \times B \rightarrow U \times V = \int_{U \times V} (\mu_A(u) \rightarrow \mu_B(v)) / (u, v), \quad (21.37)$$

де значення $\mu(u) \rightarrow \mu_B(v)$ визначається залежно від вибраної логіки.

Далі, поклавши $R(A_1(x))=A$, можна отримати логічний висновок $R(A_2(y))$ використовуючи критерій Кр1 для $R(A_1(x))$ і $R(A_1(x), A_2(y))$. Звідси очевидно випливає (з урахуванням (21.34)):

$$\begin{aligned} R(A_2(y)) &= A \circ R(A_1(x), A_2(y)) = \\ &= \int_U \mu_A(u) / u \circ \int_{U \times V} (\mu_A(u) \rightarrow \mu_B(v)) / (u, v) = \\ &= \int_V \bigvee_{u \in U} (\mu_A(u) \wedge (\mu_A(u) \rightarrow \mu_B(v))) / (u, v). \end{aligned} \quad (21.38)$$

Для того, щоб у виразі (21.38) критерій Кр 1 задовольнявся, необхідно, щоб виконувалася рівність $R(A_2(y))=B$ або

$$\bigvee_{u \in U} (\mu_A(u) \wedge (\mu_A(u) \rightarrow \mu_B(v))) = \mu_B(v),$$

або задовольнялася нерівності

$$\mu_A(u) \wedge (\mu_A(u) \rightarrow \mu_B(v)) \leq \mu_B(v), \quad (21.39)$$

яка спостерігається для довільних $u \in U$ і $v \in V$ або в термінах величин істинності $v(P \wedge (P \rightarrow Q)) \leq v(Q)$.

Резюмуючи сказане, сформулюємо основні дві умови, виконання яких необхідне для формалізації правил нечіткого умовного виведення, які повинні:

а) відповідати критеріям Кр1–Кр4; б) задовольняти нерівності (21.39).

В зв'язку з цим представляється, що формалізація удосконалених правил нечіткого умовного виведення повинна зводитися до пошуку такої багатозначної (нечіткої) логіки, яка відповідає пп. 1) і 2). Це спостерігається тільки за умов застосування S - і G -логік, при використуванні яких з'являється принципова можливість формалізувати модифіковані правила нечіткого умовного виведення:

S -логіка (*Standart sequence*)

$$a \xrightarrow{s} b = \begin{cases} 1, & \text{якщо } a \leq b; \\ 0 & \text{в протилежному випадку,} \end{cases}$$

G -логіка (*Gudelian sequence*)

$$a \xrightarrow{s} b = \begin{cases} 1, & \text{якщо } a \leq b; \\ b & \text{в протилежному випадку.} \end{cases}$$

Такі модифіковані правила в S - і G -логіках були запропоновано і досліджено в [22]. Зазначимо, що в цій праці було сформульовано правила тільки для умовної пропозиції вигляду $P1$. Ці правила мають такий вигляд:

$$1) R(A_2(y)) = R(A_1(x)) \bullet R_s(A_1(x), A_2(y)).$$

Тут

$$R_s(A_1(x), A_2(y)) = A \times B \xrightarrow{s} U \times B = \\ = \int_{U \times V} \mu_A(u)/(u, v) \xrightarrow{s} \int_{U \times B} \mu_B(v)/(u, v) = \int_{U \times V} (\mu_A(u) \xrightarrow{s} \mu_B(v))/(u, v)$$

де

$$\mu_A(u) \xrightarrow{s} \mu_B(v) = \begin{cases} 1, & \text{якщо } \mu_A(u) \leq \mu_B(v); \\ 0, & \text{в протилежному випадку.} \end{cases}$$

Це визначення ґрунтується на операції імплікації в S -логіці, тобто

$$v(P \rightarrow Q) = \begin{cases} 1, & \text{якщо } v(P) \leq v(Q); \\ 0, & \text{якщо } v(P) > v(Q). \end{cases}$$

$$2) R(A_2(y)) = R(A_1(x)) \bullet R_g(A_1(x), A_2(y)),$$

$$\text{де } R_g(A_1(x), A_2(y)) = A \times V \xrightarrow{g} U \times B = \int_{U \times V} (\mu_A(u) \xrightarrow{g} v)/(u, v),$$

що відповідає

$$\mu_A(u) \xrightarrow{g} v = \begin{cases} 1, & \text{якщо } v(P) \leq v(Q); \\ v(Q), & \text{якщо } v(P) > v(Q). \end{cases}$$

Наведені вище результати робіт у сфері формалізації правил нечіткого умовного виведення вказують на ряд істотних недоліків:

1) правила нечіткого умовного виведення, які ґрунтуються на ідеях Заде–Мамдані, не задовольняють нерівність (21.39) і критеріїв Кр1–Кр4;

2) модифіковані правила нечіткого умовного виведення в S - і G -логіках розроблено тільки для умовної логічної пропозиції вигляду $P1$;

3) у багатозначній (нечіткій) S -логіці значення імплікації визначається за принципом класичної булівої логіки 1 або 0, що істотно обмежує можливості її застосування при побудові моделі світу роботів елементами штучного інтелекту, а також при плануванні їх поведінки, оскільки не враховує нечіткий характер вхідних і вихідних посилань. При цьому для G -логіки при визначенні імплікації не враховується значення вхідної логічної посилки.

21.9. Логіка умовчання

Класична логіка виходить з того, що набір визначених в ній аксіоматичних знань є повним, і правильний висновок не зміниться

навіть після додання нової аксіоми. Таке знання називається *одноманітністю*.

Проте при доданні нової аксіоми іноді можливе заперечення виведення, яке вважалося правильним у деякій системі аксіом (БЗ). Як засіб формального оброблення неповних знань, при якому необхідно здійснювати немонотонні виведення, запропоновано методи *немонотонної логіки*, що активно досліджуються. Немонотонна логіка ще повністю не вивчена, але функції значень за умовчанням уже знаходять практичне застосування, наприклад, при поданні фреймів у робототехніці.

Якщо в монотонній логіці атомарні формули зі знаком M можна записувати в довільному місці ППФ, то в логіці умовчання їх використовують тільки в посилках правил виведення.

Розглянемо формулу $\frac{\alpha(x) : M\beta(x)}{\beta(x)}$, що означає

$\alpha(x) \wedge M\beta(x) \supset \beta(x)$, тобто $x \in \alpha$; $x \in \beta$, якщо це не суперечить іншим значенням. Передумову правила записано над ризикою й розділено знаком «:» на дві частини, де є формула без знака M і формула зі знаком M . Таким чином, вводиться обмеження на розташування формул із знаком M . При цьому можливі кілька розв'язків і виникають ситуації, коли розв'язки не можна визначити однозначно. Відповідний розв'язок розглядається як одна з можливих областей і називається розширенням (extension) із системи аксіом. Перехід від одного вибраного розв'язку до іншого вважається переходом між можливими областями. Правила виведення зі знаком M коротко називаються *правилами умовчання*.

Запис, при якому формула після знака M у передумові збігається з формулою у виведенні, називається *нормальним умовчанням*. Здебільшого значення за умовчанням можна подати в такій формі. Нормальне умовчання, в якому $\alpha(x)$; $\beta(y)$ не містять вільних змінних, називається *закритим нормальним умовчанням*. Якщо всі умовчання саме такі, то можна показати, що ця система знань має принаймні одне розширення (можливу область).

21.10. Теорія можливостей

У рамках теорії можливостей з кожним твердженням S пов'язуються дві міри: *міра можливості* $Z(S)$ та *міра необхідності* $U(S)$. Ці міри, з одного боку, мають модальний характер, а з

іншого — є узагальненням мір упевненості, які беруться до розгляду при неточному логічному виведенні.

Основною аксіомою теорії можливостей є *аксіома монотонності*, яка формулюється таким чином:

Аксіома 21.1. Якщо $S \Rightarrow M$, то $Z(M) \geq Z(S); U(M) \geq U(S)$.

Інакше кажучи, міра можливості або необхідності висновку не може бути меншою, ніж відповідна міра передумови. У [20] наводяться такі основні властивості мір можливості та необхідності:

$$Z(S \vee M) = \max((Z(S), Z(M)));$$

$$U(S \wedge M) = \min(U(S), U(M));$$

$$Z(S) = 1 - U(\neg S).$$

$Z(S) > U(S)$ (якщо твердження є *необхідно істинним*, воно повинно бути *можливо істинним*);

Якщо $U(S) > 0$, то $Z(S) = 1$;

Якщо $U(S) < 1$, то $Z(S) = 0$.

21.11. Формалізація знань інтелектуалізованих роботів

Логічні зв'язки в нечіткій логіці Z_L . У багатьох технічних системах останнім часом використовуються моделі управління, в яких закладені механізми прийняття рішень в умовах нечіткості та невизначеності. До таких технічних об'єктів безперечно належать *інтелектуалізовані роботи*, функціонування яких у складних виробничих умовах пов'язане з моделюванням їх рефлекторної діяльності.

У розд. 21.3 було проаналізовано існуючі підходи до формалізації таких наближених міркувань і показано необхідність використання для цієї мети правил нечіткого умовного виведення, які ґрунтуються на реалізації операції імплікації в різних нечітких логіках. При цьому існуючі нечіткі логіки, за винятком S - і G -логік, не задовольняють класичного правила «модус-поненс», а запропоновані Л. Заде і Мамдані правила нечіткого умовного виведення, побудовані на їх основі, не задовольняють очевидних критеріїв [22]. Водночас значення істинності операції імплікації в нечіткій G -логіці рівні або 1, або 0, а при визначенні операції імплікації в G -логіці використовуються тільки значення істинності логічного висновку. Таким чином, знижується ступінь «розмитості» імплікації, що є істотним недоліком.

В роботах [2, 73, 78] розглядаються нечіткі логіки, в яких ступінь істинності операцій імплікації визначається з урахуванням ступеня істинності передумови і висновку і які дають змогу формалізу-

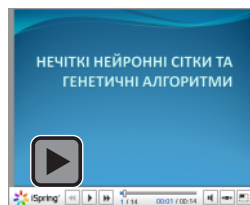
вати правила нечіткого умовного виведення, що задовольняють вищезазначені критерії, при одночасному виконанні правила «модус-поненс» (2.11 [78]). Водночас, в наведених роботах розглядаються властивості нечіткої логіки і логічні зв'язки в нечітких логіках Z_1 – Z_4 .

Контрольні завдання

1. Розкрити зміст понять неповних, недостовірних та нечітких знань.
2. Охарактеризувати властивості і навести основні ознаки нечітких множин.
3. Розкрити зміст операцій над нечіткими множинами і пояснити особливості нечіткого логічного виведення.
4. Прокоментувати особливості використання трикутної норми і конорми.
5. Дати означення нечіткого числа та розкрити зміст понять носія та унімодальності нечіткого числа.
6. Навести спрощену та узагальнену постановки задачі нечіткого виведення.
7. Розкрити чотири етапи нечіткого логічного виведення: фаззифікацію, власне логічне виведення, композицію та дефаззифікацію (зведення до чіткості).
8. Охарактеризувати модифіковані алгоритми нечіткого виведення: Mamdani, Tsukamoto.
9. Розкрити основні методи зведення до чіткості: центроїдного, першого середнього та критеріального максимумів, висотної дефаззифікації.
10. Навести означення базових посилок імовірнісної логіки.
11. Пояснити зміст байєсівської імовірнісної логіки та розтлумачити теорему (формулу) Байєса.
12. Дати пояснення терміна лінгвістичної невизначеності.
13. Проілюструвати значення положень нечіткої логіки для задач робототехніки.
14. Охарактеризувати загальні положення теорії можливостей та розкрити зміст аксіоми монотонності.
15. Навести збіжності та відмінності в нечітких логіках Z_1, Z_2, Z_3, Z_4 і прокоментувати їх переваги над правилами нечіткої логіки Л. Заде та Мамдані.

Презентації

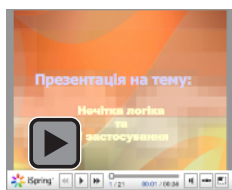
1. Нечіткі нейронні сітки
та генетичні алгоритми



2. Недостовірні та нечіткі
знання



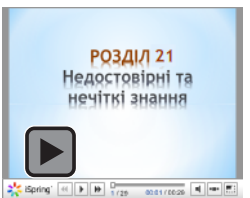
3. Нечітка логіка та її
застосування



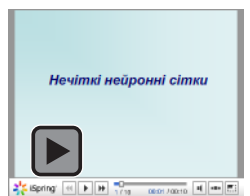
4. Нейро-фаззі технології в
системах із штучним
інтелектом



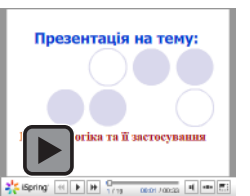
5. Недостовірні та нечіткі
знання



6. Нечіткі нейронні сітки



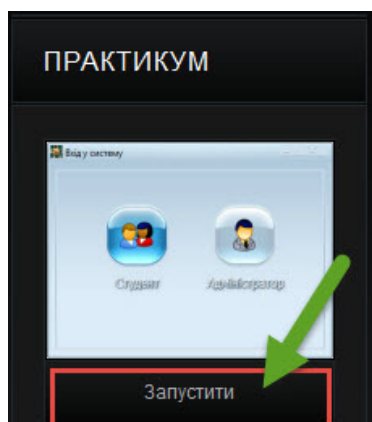
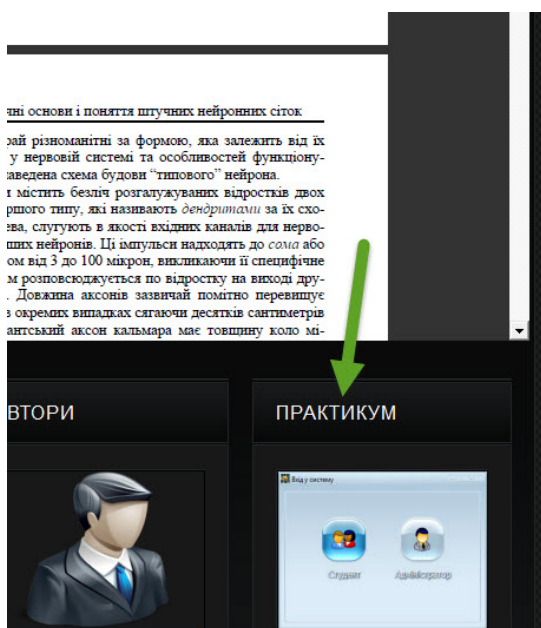
7. Нечітка логіка та її
застосування



Практичні заняття

Недостовірні та нечіткі знання

Для запуску практичних занять на
вкладці "Практикум" натисніть
"Запустити"



Розділ 22

Нейро-фаззі-технології

*Треба прагнути не до того,
щоб нас будь-хто зрозумів, а до того,
щоб нас неможливо було не зрозуміти*
Марон Публій Вергілій

*Насамперед необхідні факти,
А вже потім їх можна перекручувати*
Марк Твен

22.1. Вихідні міркування

Різні типи інтелектуальних систем мають свої особливості, наприклад, за спроможністю до навчання, узагальнення та продукування результатів, що робить їх більш придатними до розв'язання одних класів задач і менш придатними — до інших. Зокрема, *системи з ШІ, в основі яких як СПЗ використовуються нейронні сітки*, застосовуються для розпізнавання образів, проте вони дуже незручні для пояснення того, *як вони таке розпізнавання здійснюють*. Нейро-сітки можуть автоматично здобувати знання, але процес їх навчання часто відбувається досить повільно, а аналіз навченої сітки дуже складний. При цьому яку-небудь апріорну інформацію (знання експерта) для прискорення процесу її навчання в нейросітку ввести неможливо.

В *системах з нечіткою логікою*, навпаки, добре інтерпретуються отримані з їх допомогою виведення, проте *вони не можуть автоматично здобувати знання для використання їх у механізмах виведень*. Необхідність розбиття універсальних множин на окремі області, як правило, обмежує кількість вхідних змінних у таких системах несуттєвим значенням.

При створенні математичних моделей технологічних процесів доводиться стикатися з елементами невизначеності у даних. Не всі неточності опису можна віднести на рахунок похибок від вимірювань або експерименту, дуже часто невизначеність не можна усунути, вона є невід'ємною часткою процесу або контрольно-вимірювальної апаратури. Нейронні ж сітки як універсальні апроксиматори є потужним засобом математичного моделювання, і їх сполучення з нечіткою логікою надають додаткові гнучкість і функціональність створюваним моделям.

Відсутність апріорної інформації про причинно-наслідкові зв'язки у вигляді рівнянь того чи іншого типу або логіч-

них/лінгвістичних правил підштовхує розробників на використання моделей “чорної скриньки” типу системи регресійних рівнянь або ШНС. Підвищенню ролі значущості підходу з ШНС може сприяти використання в них апріорних знань щодо змінних на вході та виході сітки як про нечіткі лінгвістичні змінні.

До тепер розроблено доволі багато типів інтелектуалізованих систем з використанням гібридного архітектурного нейро-фаззі підходу (в тому числі, з використанням нейро-фаззі контролерів). Проте вони мають істотні обмеження: повільна швидкість збіжності (обумовлена використанням процедур навчання, які базуються на алгоритмі зворотного поширення похибки); неможливість функціонування в умовах істотного перетинання класів даних; відсутність апріорної інформації (зокрема, попередньо невідомій кількості кластерів).

В зв'язку з цим актуальною для класифікації і кластеризації стає задача розробки нових гібридних ШНС та ефективних методів їх навчання та самонавчання, які можуть функціонувати в умовах апріорної та поточної невизначеностей відносно характеру розподілення та перетину класів.

Теоретично системи з нечіткою логікою та штучні нейросітки подібні одна з одною, однак, згідно з викладеним вище, на практиці мають свої переваги і недоліки. Саме це і було покладено в основу створення апарату *нечітких нейронних сіток*, в яких виведення робляться на основі інструментаріїв нечіткої логіки, але відповідні функції належності піднастроюються з використанням алгоритмів навчання НС, наприклад, алгоритму зворотного поширення (розд. 4.5). У таких системах не тільки використовується апріорна інформація, а й існують механізми для здобуття нових знань, залишаючись при цьому логічно прозорими.

Нейро-нечіткі ШНС. Сітки даного типу отримали свою назву через те, що для апроксимації залежності сигналу на виході від вхідного вектора вони використовують вирази, запозичені з теорії нечітких систем, зокрема, Мамдані-Заде, Такагі-Сугена-Канга. Теоретично доведено, що ці вирази дозволяють з довільною точністю апроксимувати будь-яку неперервну нелінійну функцію багатьох змінних сумою функцій (які називають нечіткими) однією змінною.

22.2. Принципи побудови і структура нейро-фаззі-систем

Як правило, складні об'єкти являють собою багатопараметричні системи, які характеризуються суттєво нелінійними і у ряді випадків змінними у часі процесами. Просте ПІД-регулювання на

базі лінійної теорії у таких системах не завжди дає задовільні результати. Для застосування більш складних методів управління часто не вистачає інформації про процес і надійних математичних моделей, які адекватно його описують. Цим зокрема пояснюється той факт, що як і раніше деякі складні процеси керуються вручну досвідченими операторами. Знання про хід процесу, на які спирається досвідчений оператор, підсвідомо реалізуються ним у формі правил «ЯКЩО – ТО», що мають нечіткий інформаційний зміст. Цей же принцип використовується при автоматизації управління процесами на основі так званих *фаззі-контролерів* (ФК), які, по суті, реалізують наведені в розд. 22.4 чотири етапи загального логічного виведення.

Принципово узагальнену структуру фаззі-управління можна подати рис. 22.1, де зображено функціональні компоненти ФК, які й виконують процедури чотириетапного логічного виведення: *введення нечіткості–логічне виведення–композиція–зведення до чіткості*.

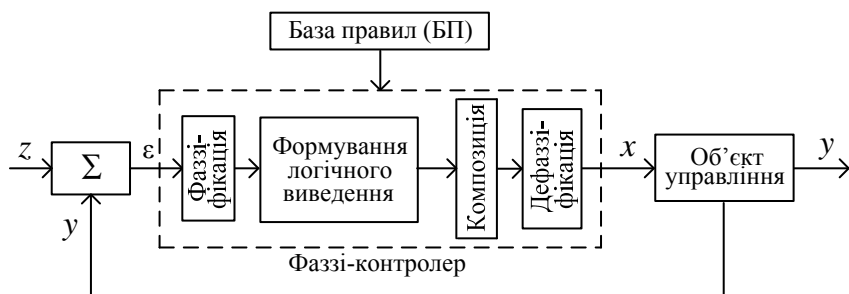


Рис. 22.1. Спрощена узагальнена структуру управління з фаззі-контролером: Z — дія, яка задається; x — керуюча дія; \hat{a} — обробка і перетворення вхідних і вимірювальних величин; y — величина, яка регулюється; БП — сукупність емпіричних знань про процес у формі правил «ЯКЩО—ТО»

1. *Процедура фаззіфікації параметрів процесу* полягає в переведенні поточних значень вхідних змінних ФК в лінгвістичні величини істинності. Для цього можливий чисельний діапазон розглянутих параметрів процесу якісно оцінюють лінгвістичними величинами, зокрема, “мало”, “середньо”, “багато”. Кожна лінгвістична величина інтерпретується як фаззі-множина й описується функціями належності. Цим якісне висловлення переводиться в кількісну величину в тому розумінні, що вона для кожного поточного числового значення змінної величини процесу відображає ступінь належності до

тієї нечіткої підмножини, яка символізує конкретну лінгвістичну величину. Оскільки *функції належності* (ФН), як правило, перекривають одна одну (див. *рисунки* 21.3 – 21.6), то для тієї самої змінної процесу декілька ФН можуть відображати різні величини істинності, які відрізняються від нуля.

2. *Формування логічного розв'язку* реалізується за допомогою лінгвістичних правил вигляду «ЯКЩО *A*, ТО *B*», що дістали назву *робочих правил*, що описують поведінку ФК відповідно до зазначених підстратегій: частина «ЯКЩО» (передумова) може при цьому означати поєднання логічних операцій будь-якої складності (див. *розд.* 21.3). Частина «ТО» (розв'язок, висновок) являє собою просте посилення на лінгвістичну величину для вихідної дії ФК. Відповідним формулюванням правил досягається результат, при якому для будь-якої лінгвістичної величини *x* керуючої дії як мінімум одне з правил виявляється прийнятним. Математично як об'єднання фаззі-множини можуть бути прийняті ті правила, які визначені в процесі фаззіфікації. А це означає, що як результат обробки деякого правила з поточним змінням величини процесу ФН частини «ТО» (на виході) буде обмеженою величиною істинності, яка відповідає результуючій величині вхідної частини «ЯКЩО» цього правила.

3. *Процедура композиції* полягає в тому, що визначення величини самої частини «ЯКЩО» виконується за допомогою формування “максимуму” величини істинності сполучення «АБО» або побудовою “мінімуму” величини істинності сполучення «І». Залежно від вибору методу побудови логічного розв'язку кожне робоче правило у будь-який момент процедури оцінки результату видає обмежену або зв'язану функцію ФН частини «ТО». Сукупність усіх сформульованих правил (база правил) є ядром ФК. Перетворення правил в єдину стратегію здійснюється об'єднанням (накладанням) окремих правил з метою одержання обмеженої (мінімізованої) кількості ФН за рівняннями (21.11) – (21.13) шляхом формування “максимуму” (див. *приклад* 21.4 з *рис.* 21.4). В результаті отримують лише стільки ФН, скільки лінгвістичних значень має вихідна змінна ФК. Придатна результуюча ФН визначає собою поточний вплив бази правил. Процедура обробки бази правил з формуванням результуючої ФН являє собою логічний розв'язок для розрахунку вихідної величини *x* ФК. У найпоширенішому методі логічного розв'язку – “мінімаксному” – за рівнянням (21.13) на першому кроці (**min**-крок) ФН частини «ТО» кожного з правил об'єднується з величиною істинності “ЯКЩО” (**min**-операція), а на другому кроці (**max**-крок) сполучені ФН окремих правил взаємно накладаються відповідно вимогам **min-max**-операції.

4. Дефаззифікація відтворює процедуру генерування керуючої дії x на виході ФК як нечіткої множини у формі ФН.

Деякі спеціальні методи дефаззифікації описані в розд. 21.5.

22.3. Особливості формування нейро-фаззі систем

Основні поняття і визначення нечітких нейросіток. Деякі фаззі-правила легко піддаються інтерпретації як локальні задачі нейронної сітки. Ряд фаззі-правил системи взаємно незалежні і можуть оброблятися паралельно. Стосовно лінгвістичних змінних, які застосовуються у правилах, а також їх структурних взаємозв'язків у нейро-фаззі-системі задається структура задачі і тим самим виконується вимога до чітко структурованої локальної НС.

Для пояснення сутності нечітких нейросіток нагадаємо принцип формування і функціонування простої НС, що складається з одного нейрона з двома входами (див. розд. 1.4). Вхідні сигнали x “взаємодіють” із синаптичними вагами w_i :

$$p_i = w_i x_i, \quad i = 1, 2.$$

Ці частинні добутки підсумовуються, утворюючи значення net нейрона:

$$net = p_1 + p_2 = w_1 x_1 + w_2 x_2.$$

Вихід нейрона утворюється в результаті перетворення значень net деякою активаційною ФН f :

$$y = f(net) = f(w_1 x_1 + w_2 x_2).$$

Наведена однеї нейронна сітка, в якій використовуються операції добутку, підсумовування та сигмоїдна функція активації, уявляє собою стандартну НС.

Означення 22.1. Нечіткою нейронною сіткою називають НС з чіткими сигналами, вагами та активаційною функцією, але з об'єднанням x_i і w_i , p_1 і p_2 із застосуванням операцій *T-норми* або *T-конорми*, які задовольняють властивості (21.1) – (21.8), або деяких інших неперервних операцій. Виходи, входи та ваги нечіткої НС — дійсні числа, що належать відрізку $[0, 1]$.

Нейронні сітки з фаззі-структурою. Основною метою при формуванні нейро-фаззі-систем є перетворення фаззі-системи в еквівалентну, так звану *інтелектуалізовану НС* (ІНС) таким чином, щоб ця сітка могла ініціюватися значеннями фаззі-системи. Після навчання результат з такої НС переводиться назад у фаззі-правила, які потім аналізуються експертом щодо їхньої прийнятності. Завдяки цій процедурі ІНС може доповнювати базу фаззі-правил.

Характерною для більшості прямих перетворень нейро-фаззі-структур є заміна всіх підфункцій фаззі-системи елементами, подіб-

ними до звичайних і спеціальних (нечітких) нейронів. Загалом, все це дає змогу оптимізувати параметри, використовуючи нейронні алгоритми навчання.

Фаззіфікуючий нейрон. Вже на етапі фаззіфікації, завданням якої є переведення вхідних змінних за допомогою ФН в лінгвістичні значення істинності, повинна враховуватися роздільність ділянок нелінійності характеристик (див., наприклад, рис. 1.5, в, д, розд. 1.2).

Приклад 22.1. Найпростіша можливість перетворення (фаззіфікація) відтворюється наведеним на рис. 22.2 прикладом, коли фаззіфікуючий нейрон (рис. 22.2, в) формується шляхом апроксимації ФН трикутної форми (рис. 22.2, а)

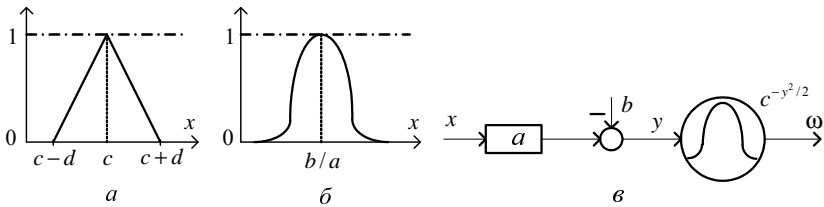


Рис. 22.2. Процес апроксимації ФН трикутної форми (а) у відповідну функцію Гауса (б) фаззіфікуючим нейроном (в)

$$y(x) = \max(0; 1 - |x - c|/d), \quad (22.1)$$

з площею апроксимованої поверхні $F_{\Delta} = d$ у рівну за площею функцію Гауса (рис. 22.2, б), що може бути подано так:

$$f(x) = e^{-(ax-b)^2},$$

де $a = \pi/d$; $b = c\sqrt{\pi}/d$, а площа функції Гауса $F_r = \sqrt{\pi}/a$.

Отже, з рис. 22.2 легко пересвідчитись, що фаззіфікуючий нейрон (рис. 22.2, в) на основі ФН (рис. 22.2, а) може бути безпосередньо параметризований.

Розглянемо ще декілька прикладів елементарних нечітких НС.

Приклад 22.2. Нечіткий нейрон «І». Сигнали x_i та ваги w_i у цьому випадку об'єднуються за допомогою T -конорми (див. означення 21.16 – 21.17):

$$p_i = S(w_i, x_i), \quad i = 1, 2,$$

а вихід утворюється із застосуванням T -норми (рис. 22.3, а)

$$y = \text{AND}(p_1, p_2) = T(p_1, p_2) = T(S(w_1, x_1), S(w_2, x_2)).$$

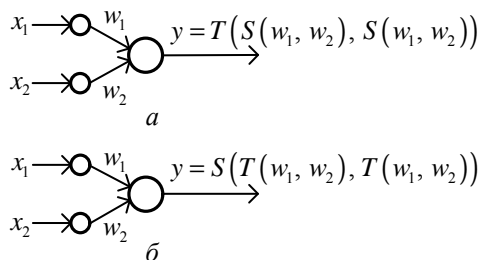


Рис. 22.3. Структури нечітких нейронів: а — «І»; б — «АБО»

Якщо прийняти $T = \max$, $S = \min$, то нечіткий нейрон «І» реалізує композицію **min-max**:

$$y = \min\{(w_1 \vee x_1), (w_2 \vee x_2)\}.$$

Приклад 22.3. Нечіткий нейрон «АБО». Сигнали x та ваги w в цьому нейроні об'єднуються за допомогою T -норми:

$$p_i = T(w_i, x_i), \quad i = 1, 2,$$

а вихід утворюється за допомогою T -конорми (рис. 22.3, б):

$$y = \text{OR}(p_1, p_2) = S(p_1, p_2) = S(T(w_1, x_1), T(w_2, x_2)).$$

Якщо прийняти $T = \min$, $S = \max$, то нечіткий нейрон «АБО» реалізує композицію **max-min**:

$$y = \max\{(w_1 \wedge x_1), (w_2 \wedge x_2)\}.$$

22.4. Відображення системи фаззі-управління в нейросітках

Для ФН вхідних сигналів, як ми переконалися, прийнятні тільки симетричні функції трикутної форми, тоді як для вихідних сигналів — монотонні. Для логічних виведень застосовується максимумний метод, а для фаззіфікації — спрощений метод центра ваги за залежностями (21.17) – (21.18).

Подання фаззіфікації. Слід зазначити, що трикутне подання ФН наближено відображає поведінку рецепторів у природі, що відповідає рівнянню

$$z_i(x) = \max(0; 1 - y_i | x - x_i), \quad (22.4)$$

де x_i відповідає номінальній точці, у якій ФН набуває значення 1; y_i — нахил обох гілок цієї функції. При цьому x_i набуває весь заданий діапазон, а y_i — лише значення > 0 . Для подання усього заданого діапа-

зону вхідної величини необхідно застосування n таких функцій. Як свідчить досвід роботи з фаззі-системами, як правило, достатньо семи ФН на одну вхідну величину.

В роботі [215] розглянуто спеціальну структуру фаззі-нейрона, де ФН включено у ваги вхідних сигналів і вхідні величини є чіткими. При цьому рівняння (22.4) в цьому випадку можна розглядати безпосередньо як вагову функцію для чіткої вхідної величини x . Недоліком тут є те, що сама вага також являє собою функцію. Тому на рис. 22.4, а наведено більш прийнятний для оптимізації підхід, який макс-функцію за рівнянням (22.4) переводить у відображаючу функцію $f(x) = \max(0; 1 - |\Sigma|)$; $\Sigma = w_2x - w_1$, а також реалізує розташування і форму ФН у вагах $w_1 = ux$ і $w_2 = y$.

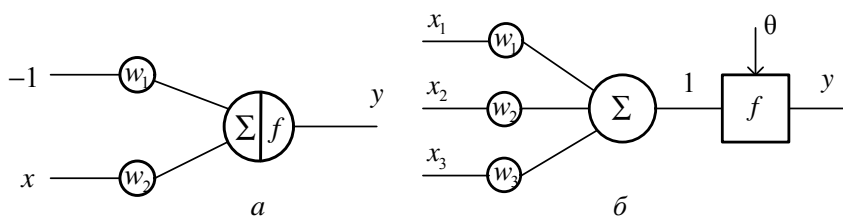


Рис. 22.4. Фаззі-нейрон для відображення фаззіфікації (а) і нейрон для формування max-, min-операцій (б)

Логічне розв'язання в нейро-фаззі-системі. Фаззі-правило і його відповідне розв'язання, тобто зв'язок між нечіткими вхідною і вихідною множинами, є також нечіткими множинами, що уявляють собою сформований з цих двох множин висновок. Нехай маємо стандартне правило (див. розд. 21.5) «ЯКЩО A , ТО B » з елементами передумови A і постумови (висновку) B і відповідними ФН $\mu_A(x)$ і $\mu_B(y)$. Тоді висновком буде основна множина $A \times B$, і його можна подати у формі

$$R = \{(x, y), \mu_R(x, y); x \in A, y \in B\}. \quad (22.5)$$

Нагадаємо (див. розд. 21.5), що для визначення ФН $\mu_R(x, y)$ існує декілька методів, з яких частіше застосовується **min-операція**, що відповідає логічному сполученню «І» нечітких множин:

$$\mu_R(x, y) = \min(\mu_A(x), \mu_B(y)). \quad (22.6)$$

Функція $\mu_A(x)$ в рівнянні (22.6) задає для кожної величини $x_i \in A$ максимум-обмеження величини ФН $\mu_B(y)$. Декілька правил звичайно сполучаються за допомогою оператора «АБО» для того,

щоб отримати діючу композицію усіх правил. Тому ФН композиції правил являє собою **max-утворення** ФН окремих правил (див. також *приклад* 21.4 та 21.5):

$$\mu_E(y) = \max(\mu_{R1}(x, y), \mu_{R2}(x, y), \dots). \quad (22.7)$$

Тоді для результуючого логічного розв'язку (виведення) за сумою правил (22.6) і (22.7) маємо:

$$\mu_E(y) = \max \min(\mu_{Ai}(x), \mu_{Bi}(y)), \quad i = \overline{1, n}. \quad (22.8)$$

Логічне розв'язання. Логічне розв'язання повністю складається саме з таких аналогових **max-**, **min-**утворень. Для їх відображення на нейросітці спочатку визначаються її спеціальні тип і топологія, які здатні подати простою побудовою цю нелінійну функцію. Така схема відповідає лінійному типу НС (рис. 22.4, б), яка піддається спеціальному граничному випробуванню з вхідними величинами x_i , вагами w_i до цих вхідних величин, граничною величиною θ і нелінійною функцією відображення

$$f(J) = 0,5(\text{sign}(J - \theta) + 1)J = \begin{cases} J, & \text{якщо } J > \theta \\ 0,5J, & \text{якщо } J = \theta \\ 0, & \text{якщо } J < \theta \end{cases} \quad (22.9)$$

За допомогою цього типу сітки з вагами $w_1=1$, $w_2=1$ і f_2 як такої, що лінійно відображає функції формування максимуму

$$y = y_1 + y_2 = 0,5(\text{sign}(x_1 + x_2) + 1)x_1 + 0,5(\text{sign}(x_2 - x_1) + 1)x_2 = \begin{pmatrix} x_1 & x_1 > x_2 \\ 0,5x_1 & x_1 = x_2 \\ 0 & x_1 < x_2 \end{pmatrix} + \begin{pmatrix} 0 & x_1 > x_2 \\ 0,5x_2 & x_1 = x_2 \\ x_2 & x_1 < x_2 \end{pmatrix} = \max(x_1, x_2) \quad (22.10)$$

можна подати схемою (рис. 22.5, а).

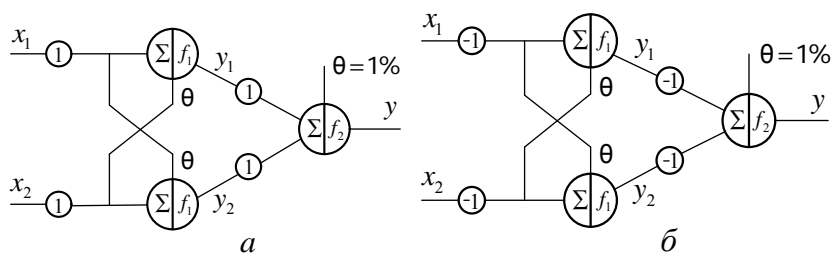


Рис. 22.5. Формування максимуму (а) і мінімуму (б) для двох функцій y_1 та y_2

Формування мінімуму виконується за тією ж схемою сітки, але зі зміненими вагами $w_1 = -1$; $w_2 = -1$ (рис. 22.5, б):

$$\begin{aligned}
 y &= -y_1 - y_2 = -0,5(\text{sign}(x_2 - x_1) + 1)(-x_1) - \\
 &- 0,5(\text{sign}(x_1 - x_2) + 1)(-x_2) = \\
 &= \begin{cases} 0 & x_1 > x_2 \\ 0,5x_1 & x_1 = x_2 \\ x_1 & x_1 < x_2 \end{cases} + \begin{cases} x_2 & x_1 > x_2 \\ 0,5x_2 & x_1 = x_2 \\ 0 & x_1 < x_2 \end{cases} = \min(x_1, x_2). \quad (22.10)
 \end{aligned}$$

Для придушення малих, близьких до нуля величин на вході доцільно для підсумовуючих нейронів на виході встановлювати певне граничне значення, наприклад, як це показано на рис. 22.5, б на рівні $\theta = 1\%$. Це дасть змогу значення ФН, менші від 0,001, відображати точно в нуль. Це вкрай необхідно в разі зашумленості вхідних сигналів, що утруднює їх розшифрування і застосування градієнтного методу для оптимізації.

Відповідні структури для більш як двох входів можна побудувати послідовним з'єднанням (рис. 22.6, а). Зокрема, логічне заперечення деякої величини з урахуванням визначень фаззі-множин означає віднімання одиниці $\bar{b} = 1 - b$ і формування такого доповнення може виконуватися одним нейроном з одним входом і однією опорною величиною (рис. 22.6, б).

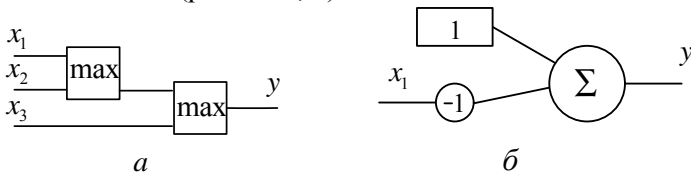


Рис. 22.6. Формування: а — максимуму при трьох вхідних сигналах; б — доповнення фаззі-параметра

Логічний розв'язок, тобто імплікація частини «ЯКЩО» до частини «ТО», легко відображається за умови, якщо для величини на виході застосувати так званий *фаззі-монотонний сигнал*. У цьому випадку кожна величина на виході фаззі-системи складатиметься з ряду дискретних значень, які при розв'язанні визначаються їх вагами. Нехай g є таким монотонним сигналом на виході з ФН $\mu_g(y)$, а $\mu_\omega(x_1)$ результуючий ступінь належності частини «ЯКЩО» правила; тоді для розв'язання буде справедливим таке правило:

$$\mu_R(x, y) = (\mu_\omega(x_1), \mu_g(y)). \quad (22.11)$$

Отже, завдяки застосуванню фаззі-монотонного сигналу в нейронному поданні формування мінімуму вироджується в лінійну передатну функцію з підсиленням, яке дорівнює одиниці. Монотонні величини в межах наступної дефаззіфікації однозначно розподіляються, що забезпечує можливість визначення ФН $\mu_g(y)$.

Відображення дефаззіфікації. Чітка величина сигналу z_i на виході при застосуванні фаззі-монотонного сигналу визначається залежностями (21.15), (21.19) – (21.22), і ця процедура детально розкривається в розд. 21.6. Інтерпретацію цієї процедури нейро-фаззі-компонентами, наприклад, для найпоширенішого методу розрахунку за законом центра ваги, можна подати такими міркуваннями. Скористуємося для цього видозміною залежності (21.15):

$$z_i = \sum_{j=1}^n y_{ij} x_{ij} / \sum_{j=1}^n y_{ij}, \quad (22.12)$$

де x_{ij} – керуюча дія, яка відповідає кожному *фаззі-монотонному сигналу*; y_{ij} – ваги, які визначаються логічним розв'язком фаззі-системи. Отже, якщо розглядати y_{ij} як ваги, то лічильник може легко відтворювати рівняння (22.12) як лінійний нейрон з входами x_{ij} . Аналогічно знаменник у виразі (22.12) можна відтворювати лінійним нейроном з вхідними вагами 1.

Для генерування сигналу Y на виході застосовується тип нейрона (рис. 22.7, а), який припускає в активаційній функції керувану вагою активність. В такому нейроні поряд зі змінюваною кількістю входів тільки величини ваг y_{ij} можуть змінюватися. У результаті залежність (22.12) однозначно трансформується в нейронне уявлення.

На рис. 22.7, б наведено схему НС з комплексним нейро-фаззі-перетворенням, що включає фаззіфікацію, логічне розв'язання, **max-min**-композицію та дефаззіфікацію. Вхід сітки утворюється трьома ФН, на виході використовуються дві ФН. Поряд з вхідним і останнім шарами присутні також три сховані шари h_1, h_2, h_3 .

Перший схований шар h_1 з його вагами виконує *фаззіфікацію*, другий шар h_2 формує частину «ЯКЩО» правил, використовуючи при цьому виключно **min**-співвідношення, у той час, як **max**-співвідношення між двома фаззі-вхідними величинами перекриваються застосуванням двох правил. Третій схований шар h_3 формує *композиційний логічний розв'язок* за застосованими правилами. Між схованими шарами h_1 і h_2 вбудований настроювальний механізм для навчання сітки у вигляді регульованих ваг w_i , гранична величина θ відіграє роль фільтруючого елемента, а збурювання у вигляді Q - Z діє на вході *дефаззіфікуючого* нейрона.

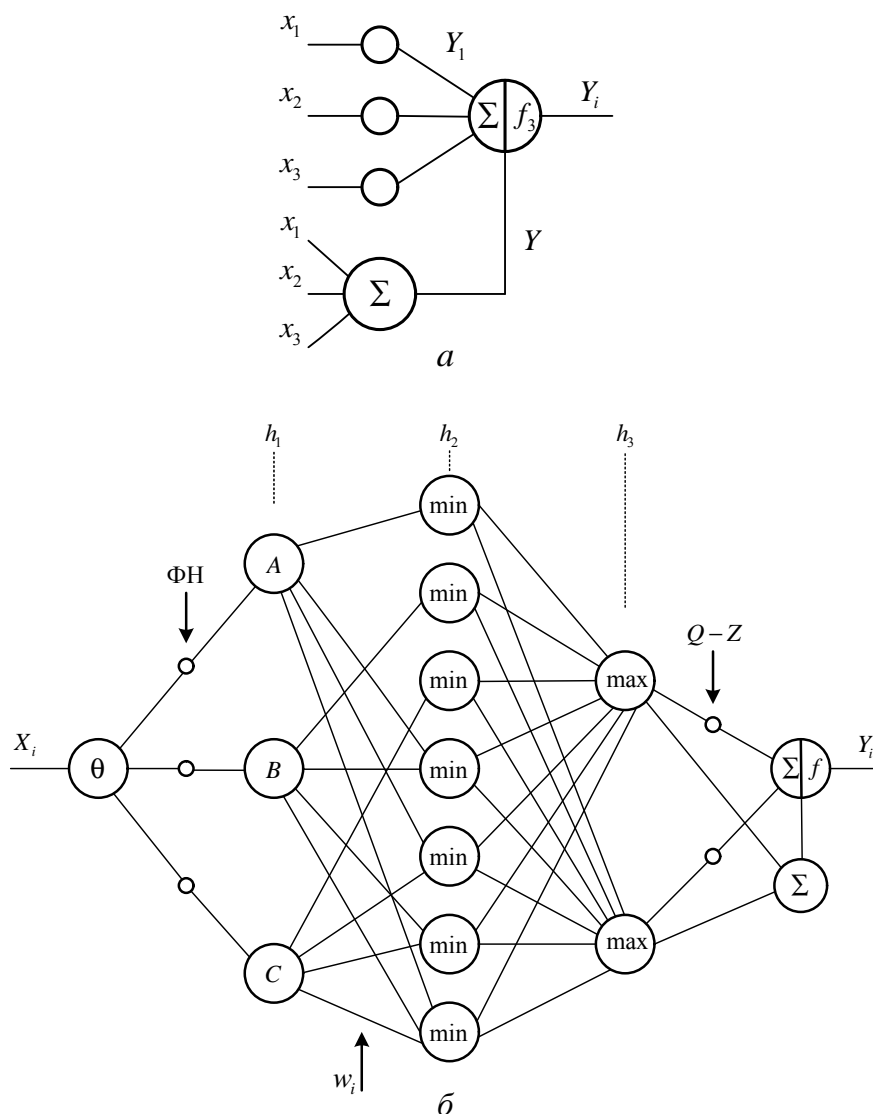


Рис. 22.7. Тип дефаззифікуючого нейрона (а)
і нейросітка з комплексним нейро-фаззі-перетворенням (б)

Кожна з ваг схованого шару h_1 подає пари взаємозалежних ві-
льних настроювань (w_1, w_2), які визначають параметри кожної ФН. Ва-
ги шару h_2 можуть бути двійковими (0/1), і в цьому випадку вони щора-
зу вводять або виключають імплікацію (операцію «І»). Можуть бути

також застосовані аналогові ваги між 0 і 1, які уявляють собою *надійні коефіцієнти імплікації*.

Ваги Q - Z вихідного шару характеризують положення фаззі-монотонності у певному діапазоні сигналу на виході.

Якщо прийняти ваги шару h_1 (ФН-входу) аналоговими, ваги шарів h_2 і h_3 двійковими, а ваги Q - Z останнього шару (фаззі-монотонності) теж аналоговими, то перетворення стає однозначним, тобто відображення буде на 100 відсотків достовірним. Це дозволяє скільки завгодно переходити від фаззі-зображення до нейро-зображення і навпаки.

Для оптимізації відомими методами навчання (див. *розділи 4.5 та 21.6*), наприклад, за допомогою алгоритму зворотного поширення, рекомендується використовувати аналогові ваги також і для схованих шарів h_2 і h_3 . Це супроводжується, однак, одним недоліком, який полягає в тому, що при зворотному перетворенні ці аналогові ваги повинні знову переводитися у двійкові.

22.5. Алгоритми тренування нейро-фаззі-систем

Особливості узгодження областей керування в інтегрований нейро-фаззі-системі. Фаззі-логіка і техніка НС ілюструє їх позитивні особливості в різних сферах застосувань. Перша дає змогу безпосередньо формулювати стратегії розв'язання задач при побудові систем управління процесами, тоді як НС мають бути відтреновані за введеними даними. Фаззі-системи самостійно не навчаються, і всі знання повинні вводитися в них ззовні, а спроможності НС, навпаки, за введеними даними мають бути відтреновані. Нарешті, фаззі-логіка успішно функціонує, якщо є знання у формі стратегій, необхідних для розв'язання поставленої мети, наприклад, у задачах моделювання, проектування та управління процесами в об'єктах автоматизації. Водночас, за допомогою НС можуть успішно оброблятися передусім складні задачі розпізнавання та аналізу даних. Як бачимо, слабкі та сильні сторони обох цих підходів діаметрально протилежні. Тому зрозуміле мотивування до розробки *інтегрованих нейро-фаззі-систем* (ІНФС). Спряження НС з фаззі-системами може виконуватися в декількох варіантах, які визначають характер їх взаємодії. На рис. 22.8 наведено структуру ІНФС з паралельною обробкою сигналів у процесі її тренування. Ця схема відповідає випадкам, коли для фази навчання НС є дані тільки частини простору вхід/вихід (випадок часткового навчання), тому за межами гарантованої робочої області НС фаззі-система (або фаззі-контролер) має забезпечити заданий режим

450

роботи. Як це досягається, наведено на рис. 22.8, б для векторів вхідної $\bar{X}_{(i)}$ та вихідної $\bar{Y}_{(i)}$ змінних.

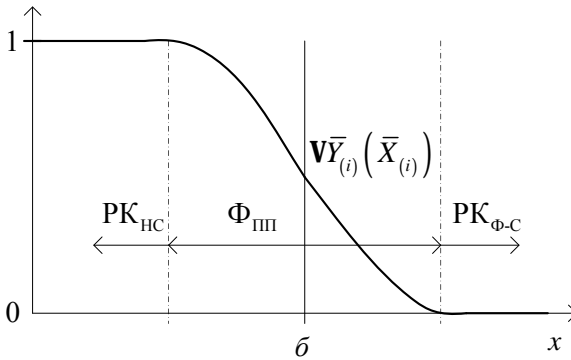
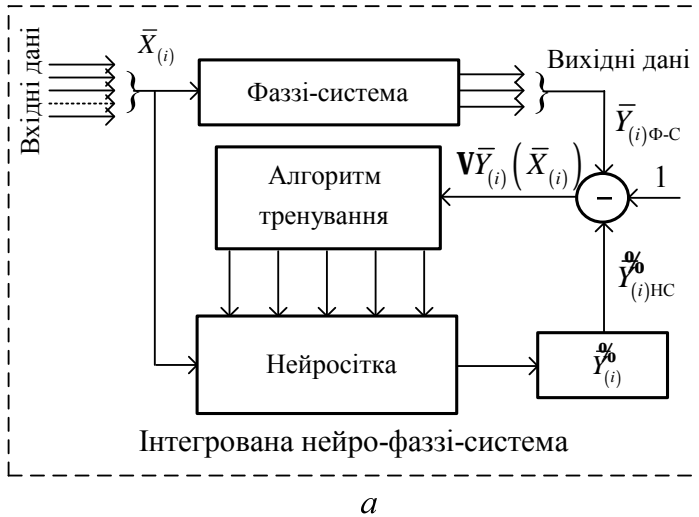


Рис. 22.8. Структура інтегрованої нейро-фаззі-системи з паралельною обробкою сигналів (а) і функція узгодження між фаззі-системою та НС областей керування (б)

Так, для плавного переходу на межу ділянку між вихідними змінними фаззі-системи і НС, як це впливає з рис. 22.8, б, з НС необхідно вивести сигнал $V\bar{Y}_{(i)}(\bar{X}_{(i)})$ достовірності, який вказує, де саме НС повинна діяти самостійно, а де достовірність її дії є ненадійною, а також де дії НС повинні бути зовсім виключені. Справді, як можна спостерігати з рис. 22.8, б, функція $\Phi_{ПП}$ плавного переходу

від області режиму РК_{НС} самостійної дії НС до області режиму РК_{Ф-С} керування від фаззі-системи базується на значеннях сигналу $V\bar{Y}_{(i)}(\bar{X}_{(i)})$ достовірності керуючої дії. Нарешті, вихідний сигнал $\bar{Y}_{(i)НС}^{\%}$ і вихідна величина $\bar{Y}_{(i)Ф-С}$ фаззі-системи можуть далі підсумовуватися за принципом вагового формування середнього (див. розд. 21.6) для отримання ефективного значення результуючого сигналу:

$$\bar{Y}_{(i)} = \Delta \bar{Y}_{(i)}(\bar{X}_{(i)})_{\bar{Y}_{(i)НС}} + |1 - \Delta \bar{Y}_{(i)}(\bar{X}_{(i)})|_{\bar{Y}_{(i)Ф-С}}. \quad (22.13)$$

Означення 22.2. *Інтегрована нейро-фаззі-система* — це програмно-алгоритмічна і технічна реалізація, яка характеризується за своєю суттю і функціональною спроможністю до навчання (що є характерною ознакою НС) і наочністю подання знань з елементами лінгвістичної невизначеності (характерною особливістю фаззі-логіки).

Концепція, яка базується на використанні фаззі-систем із структурованими знаннями у вигляді СПЗ з можливістю подальшого навчання останніх, впливає з того, що технологічні процеси в об'єктах керування при зростанні їх складності неможливо точно математично описати, а вважається, що людина за інструкцією або з досвіду цілком спроможна складні взаємодії у ході протікання процесу як мінімум якісно оцінити та цілеспрямовано на них вплинути.

Алгоритм тренування нечіткої НС. Розглянемо типовий підхід до побудови алгоритмів навчання НС. Уявимо, що нечіткою НС повинно бути реалізоване невідоме відображення:

$$y^{(g)} = f(x^{(g)}) = f(x_1^{(g)}, x_2^{(g)}, \dots, x_m^{(g)}); \quad g = 1, \dots, M$$

за наявності навчаючої множини $\{(x^{(1)}, y^{(1)}), \dots, (x^{(M)}, y^{(M)})\}$.

Для моделювання невідомого відображення f використаємо спрощений алгоритм нечіткого виведення (див. розд. 21.4), застосовуючи таку форму запису предикатних правил:

$$P_i: \text{ЯКЩО } x_1 \in A_{j1} \text{ І } x_1 \in A_{j1} \text{ І } \dots \text{ І } x_m \in A_{jm}, \text{ ТО } y = z_j; \quad j = 1, \dots, k,$$

де A_{ij} — нечіткі числа трикутної форми; z — дійсні числа.

Ступінь істинності i -го правила визначається за допомогою операції добутку:

$$\alpha_j = \prod_{j=1}^m A_{ij}(x_j^{(g)}),$$

причому можна використовувати й будь-які інші подання для моделювання логічного оператора «І».

Вихід нечіткої системи визначається згідно з (21.18) для дискретного варіанта центроїдного методу:

$$z_0^{(g)} = \left(\sum_{j=1}^n \alpha_j z_j \right) / \sum_{j=1}^n \alpha_j.$$

Метою тренування є мінімізація похибки, що стосується кожного комплексу зразків (образів) так, щоб прийнятий комплект зразків був достатнім для виведення закону зміни ваги. Введення функції похибки для g -го поданого зразка виду

$$E_{(g)} = 0,5 \left(z_0^{(g)} - y^{(g)} \right)^2$$

дає змогу в подальшому, як і у звичайних НС, використовувати градієнтний метод для піднастроювання параметрів заданих предикатних правил. Тоді величини z_j можна корегувати за співвідношенням:

$$z_j := z_j - \xi \frac{\partial E_{(g)}}{\partial z_j} = z_j - \xi \left(z_0^{(g)} - y^{(g)} \right) \frac{\alpha_j}{\alpha_1 + \alpha_2 + \dots + \alpha_n}; \quad j = 1, \dots, n,$$

де ξ – константа, що характеризує швидкість навчання.

Процес тренування. На першому кроці реалізується грубий підхід на базі фаззі-системи. Цю первинну структуру загалом можна досить швидко знайти і часто можна безпосередньо застосовувати для поєднання з процесом, що моделюється, сприяючи одержанню попередніх результатів, які, однак, ще далекі від оптимальних. Сформована таким шляхом базисна структура вже визначає всі сигнали на вході і виході, які мають застосовуватися у фаззі-структурі, і являє собою *стартовий комплект параметрів для наступної оптимізації із застосуванням НС.*

На другому кроці у фазі навчання НС отримує від фаззі-системи всі сформовані нею прийнятні сигнали $\bar{Y}_{(i)}$ процесу і керуючі сигнали $\bar{X}_{(i)}$, які надходять, наприклад, від людини-оператора (див. рис. 22.8, а). Вони рекурсивно спрягаються. На основі отриманої таким шляхом моделі виконується розрахунок похибки за критерієм якості (наприклад, формуванням середнього значення), з якого визначають, чи достатньо добре збігаються модель і дійсність. Якщо отримано задовільний збіг, подальше тренування припиняється, і НС або трансформована фаззі-структура готові до роботи в активному режимі з керованим процесом.

У тих ситуаціях, які не відтреновані НС, відбувається керування процесом оператором у ручному режимі, ІНФС перестає брати участь в роботі і переходить в режим нової фази тренувань.

Приклад 22.4. Як приклад розглянемо більш складну нечітку систему, що має наведену базу знань і до якої застосовується алгоритм Tsukamoto:

П1: ЯКЩО $x_1 \in A_1$ І $x_2 \in A_2$ І $x_3 \in A_3$, ТО $y \in D$;

П2: ЯКЩО $x_1 \in B_1$ І $x_2 \in B_2$ І $x_3 \in A_3$, ТО $y \in E$;

П3: ЯКЩО $x_1 \in B_1$ І $x_2 \in B_2$ І $x_3 \in B_3$, ТО $y \in F$,

де x_1, x_2, x_3 – вхідні змінні; y – вихід системи; $A_1, A_2, A_3, B_1, B_2, B_3, D, E, F$ – деякі нечіткі множини з функціями належності сигмоїдного типу:

$$B_j(x_i) = 1 / \left[1 + \exp(-b_j(x_i - c_j)) \right], \quad A_j(x_i) = 1 / \left[1 + \exp(b_j(x_i - c_j)) \right], \quad j = 1, 2, 3;$$

$$D(z_1) = 1 / \left[1 + \exp(-b_4(z_1 - c_4 + c_5)) \right], \quad E(z_2) = 1 / \left[1 + \exp(-b_4(z_2 - c_4)) \right],$$

$$F(z_3) = 1 / \left[1 + \exp(b_4(z_3 - c_4)) \right].$$

Для визначення вихідної змінної скористуємося алгоритмом Tsukamoto (ілюстративно наведено на рис. 22.9), у відповідності за яким:

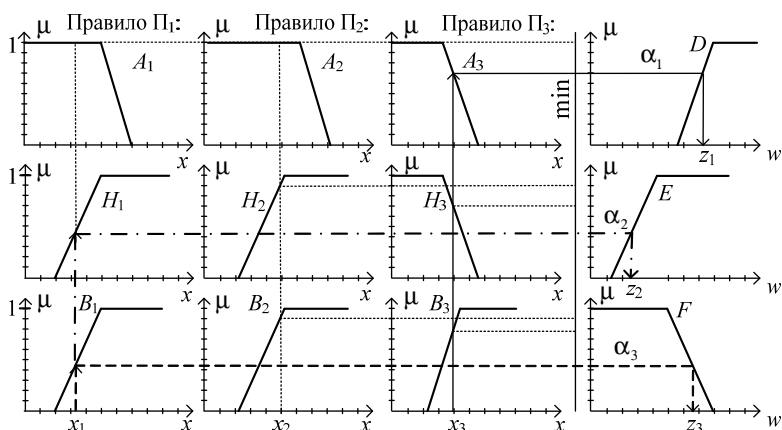


Рис. 22.9. Ілюстративне подання прикладу 22.4 нечіткої системи

- підраховуються значення істинності передумов для кожного правила:

$$\alpha_1 = A_1(x_1) \wedge A_2(x_2) \wedge A_3(x_3); \alpha_2 = H_1(x_1) \wedge H_2(x_2) \wedge H_3(x_3);$$

$$\alpha_3 = B_1(x_1) \wedge B_2(x_2) \wedge B_3(x_3),$$

де x_1, x_2, x_3 – поточні значення входів системи;

- для кожного правила визначаються частинні виходи:

$$z_1 = D^{-1}(\alpha_1) = c_4 + c_5 + (1/b_4) \ln[(1 - \alpha_1)/\alpha_1];$$

$$z_2 = E^{-1}(\alpha_2) = c_4 + (1/b_4) \ln[(1 - \alpha_2)/\alpha_2];$$

$$z_3 = F^{-1}(\alpha_3) = c_4 + (1/b_4) \ln[(1 - \alpha_3)/\alpha_3]$$

визначається загальний вихід системи:

$$z_0 = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3) / (\alpha_1 + \alpha_2 + \alpha_3).$$

22.6. Узагальнена структура адаптивної нейро-фаззі системи

Нечітку НС, яка реалізує наведений у цьому прикладі механізм тренування і виведення, подано на рис. 22.10. Зауважимо, що сітка з такою архітектурою дістала також назву *адаптивної системи з нейро-фаззі-виведеннями* (Adaptive Neuro-Fuzzi Inference System).

Такі сітки можна описати так:

Шар 1. Виходи нейронів цього шару уявляють собою значення функцій належності при конкретних (заданих) значеннях входів.

Шар 2. Виходами нейронів цього шару є ступені істинності передумов кожного правила бази знань системи, які обчислюються за формулами:

$$\alpha_1 = A_1(x_1) \wedge A_2(x_2) \wedge A_3(x_3),$$

$$\alpha_2 = B_1(x_1) \wedge B_2(x_2) \wedge A_3(x_3),$$

$$\alpha_3 = B_1(x_1) \wedge B_2(x_2) \wedge B_3(x_3).$$

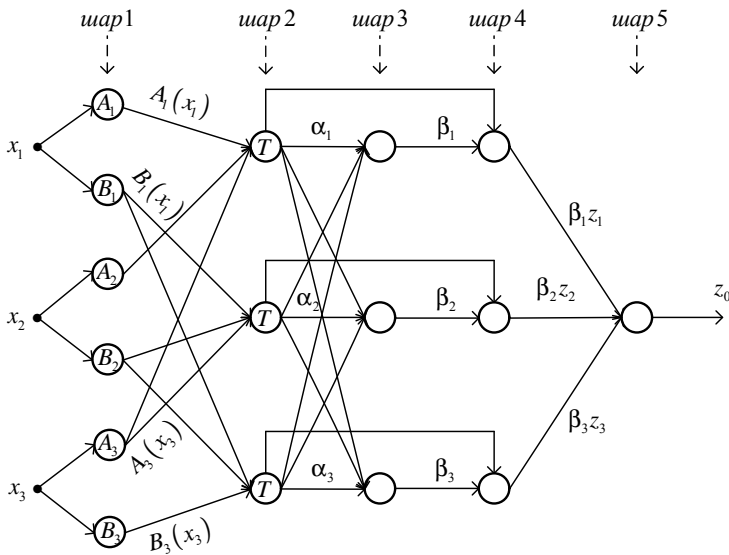


Рис. 22.10. Структура адаптивної системи з нейро-фаззі виведеннями

Усі нейрони цього шару позначені символом T , і це означає, що вони можуть реалізовувати довільну T -норму (див. розд. 21.3) для моделювання операції «І».

Шар 3. Нейрони цього шару обчислюють величини (операція нормування):

$$\beta_1 = \alpha_1 / (\alpha_1 + \alpha_2 + \alpha_3); \beta_2 = \alpha_2 / (\alpha_1 + \alpha_2 + \alpha_3); \beta_3 = \alpha_3 / (\alpha_1 + \alpha_2 + \alpha_3).$$

Шар 4. Нейрони цього шару виконують операції:

$$\beta_1 z_1 = \beta_1 D^{-1}(\alpha_1); \beta_2 z_2 = \beta_2 E^{-1}(\alpha_2); \beta_3 z_3 = \beta_3 F^{-1}(\alpha_3).$$

Шар 5. Єдиний нейрон цього шару обчислює вихід сітки:

$$z_0 = \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_3.$$

Коригування параметрів системи для функцій належності D , E , F (див. рис. 22.9, приклад 22.4) виконується відповідно до раніше розглянутих підходів. Так, наприклад, настроювання коефіцієнтів b_4 , c_4 , c_5 для функцій належності здійснюється за формулами:

$$b_4 := b_4 - \xi \frac{\partial E_{(g)}}{\partial b_4} = b_4 - \frac{\xi}{b_4^2} \delta_{(g)} \frac{\alpha_1 + \alpha_2 - \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3};$$

$$c_4 := c_4 - \xi \frac{\partial E_{(g)}}{\partial c_4} = c_4 + \xi \delta_{(g)} \frac{\alpha_1 + \alpha_2 + \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3} = c_4 + \xi \delta_{(g)};$$

$$c_5 := c_5 - \xi \frac{\partial E_{(g)}}{\partial c_5} = c_5 + \xi \delta_{(g)} \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3},$$

де $\delta_{(g)} = y^{(g)} - z_0^{(g)}$.

Відповідні вирази можна отримати і для решти коефіцієнтів b_j і c_j , $j = 1, 2, 3$.

Визначають два підходи до модифікації топології нечіткої НС на етапах навчання та використання. *Перший традиційний* базується на введенні додаткових продукційних правил у базу знань системи з урахуванням виконання вимоги несуперечливості її поповнення.

Другий підхід, запропонований в [31], передбачає генерацію нових продукційних правил (що не суперечать правилам з бази знань системи), виходячи з аналізу експериментальних даних про об'єкт.

22.7. Архітектура нечітких нейросіток

Сітка Такагі-Сугено-Канга. Нечіткі моделі нейросітки Такагі-Сугено-Канга (скорочено TSK) – це нелінійні системи, здатні апроксимувати широкий клас складних або нелінійних систем за допомогою “ЯКЩО-ТО” правил продукції [195], які описують локально-лінійну динаміку всієї системи.

В сітці TSK сигнал на виході розраховується за допомогою виразу:

$$z(X) = \text{sum}[i = 1 : M](\alpha_i * z_i(X)) / \text{sum}[i = 1 : M](\alpha_i), \quad (22.14)$$

де $z_i(X) = p_{i0} + \text{sum}[j = 1 : N](p_{ij} * x_j)$ – i -й поліноміальний компонент апроксимації.

Ваги α_i компонентів обчислюються за наступною формулою (з використанням раціональної форми функції Гаусса):

$$\begin{aligned} \alpha_i &= \text{prod}[j = 1 : N](\alpha_{ij}(x_j)) = \\ &= \text{prod}[j = 1 : N] \left(\frac{1}{1 + ((x_j - c_{ij}) / s_{ij})^{2 * q_{ij}}} \right). \end{aligned} \quad (22.15)$$

Наведеним виразам (22.14) та (22.15) відповідає п'ятишарова нейронна сітка, структурна схема якої представлена на рис. 22.11.

Перший шар містить $N \times M$ вузлів, кожний з яких реалізує розрахунок функції Гаусса (в експоненціальному або раціональному виді) з параметрами центра c_{ij} , ширини s_{ij} та форми q_{ij} . Параметри A_{NM} , B_{NM} , G_{NM} – деякі нечіткі множини з функціями належності гауссового типу. З точки зору нечітких систем це шар *фаззіфікації* вхідних змінних (див. розд. 22.6). Шар називається *параметричним*, оскільки в процесі навчання сітки підбиранню належать параметри саме цього шару.

Другий шар параметрів не містить. З точки зору нечітких систем це шар агрегування лівих часток продукцій. Усі нейрони цього шару позначені символом T , і це означає, що вони можуть реалізовувати довільну T -норму (див. розд. 21.3) для моделювання операції «І».

Третій шар – генератор (поліноміальних) функцій TSK $z_i(X)$ та їх помножувач на ваговий коефіцієнт α_i . Це *параметричний* шар, в якому коефіцієнти $p_{ij}, i = 1, 2, \dots, M, j = 0, 1, \dots, N$ в процесі навчання сітки піддаються адаптації. Загальна кількість p_{ij} в сітці дорівнює $M * (N + 1)$.

Четвертий шар складають два нейрона-суматора. Перший розраховує зважену суму сигналів $z_i(X)$, а другий – суму ваг $\alpha_i, i = 1, 2, \dots, M$. Це *непараметричний* шар.

Останній, *п'ятий шар* здійснює нормалізацію ваг. Це також *непараметричний* шар.

З опису сітки TSK випливає, що вона містить два параметричних шари (перший та третій), параметри яких підлягають підбиранню в процесі навчання. Параметри першого шару називають *неліній-*

ними через те, що вони відносяться до нелінійної функції, а параметри третього шару – лінійними.

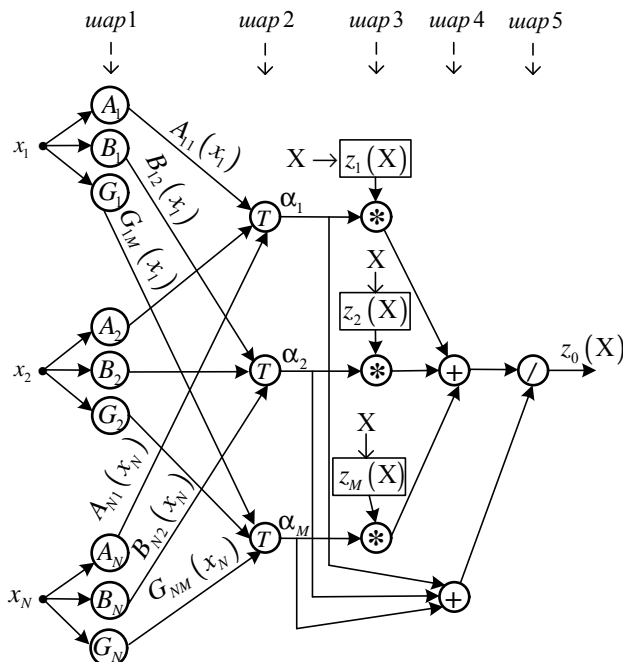


Рис. 22.11. Структура сітки Такагі-Сугено-Канга

Загальна кількість параметрів (лінійних і нелінійних) сітки TSK дорівнює $M * 3 * N + M * (N + 1) = M * (4N + 1)$.

В багатьох практичних застосуваннях це надмірна величина, тому часто для входніх змінних x_j використовують обмежений набір функцій $mu(x_j)$, що зменшує кількість нелінійних параметрів.

Щодо питання відносно стійкості TSK-сіток, то слід зазначити, що одним з достатньо розроблених методів дослідження стійкості нечітких систем є метод лінійних матричних нерівностей (LMI-метод), який вперше був використаний для аналізу стійкості однорідних неперервних [197] і дискретних [196] нечітких систем. Для систем диференціальних рівнянь з імпульсним діянням цей метод розвинутий в роботі [52]. Зручність LMI-методу обумовлена його чисельною реалізацією в інтегрованому середовищі MATLAB.

В роботі [19] отримано результати, які дозволяють зводити задачу про стійкість нечіткої імпульсної системи до питання про сумісність деякої системи лінійних матричних нерівностей і як резуль-

тат доведено, що стабілізація в імпульсних системах Такагі-Сугено можлива навіть у випадку нестійкості її неперервної та дискретної компонент.

Сітка Ванга-Менделя [31]. В сітці даного типу (рис. 22.12) сигнал на виході розраховується за допомогою співвідношення:

$$\begin{aligned} z(X) &= \text{sum}[i = 1 : M](c_i * \alpha_i) / \text{sum}[i = 1 : M](\alpha_i) = \\ &= \text{sum}[i = 1 : M](c_i * \text{prod}[j = 1 : N](\text{mu}_{ij}(x_j))) / \\ &\quad / \text{sum}[i = 1 : M](\text{prod}[j = 1 : N](\text{mu}_{ij}(x_j))), \end{aligned} \quad (22.16)$$

де c_i – ваговий коефіцієнт (з точки зору нечітких систем це центр функції належності правої частки продукції); $\text{mu}()$ – функція Гаусса (в експоненціальному або раціональному виді) з параметрами центра c_{ij} , ширини s_{ij} та форми q_{ij} (з точки зору нечітких систем $\text{mu}()$ – функція належності до нечіткої множини).

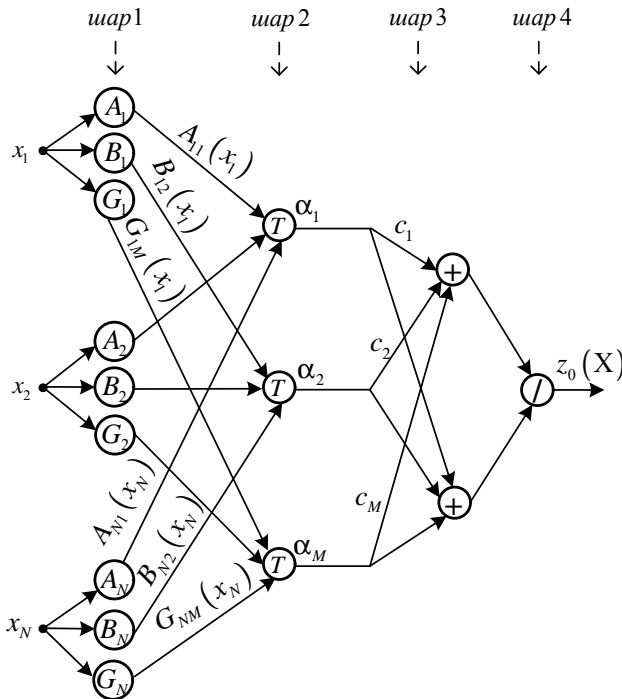


Рис. 22.12. Структура сітки Ванга-Менделя

Легко перекопати, що вираз для $z(X)$ в сітці Ванга-Менделя є частинним випадком аналогічного виразу (22.14) в сітці TSK, якщо

в останній прийняти $z_i(X)=c_i$. Тому сітка Ванга-Менделя простіша і має наступну чотиришарову структуру.

В даній сітці параметричними є перший і третій шари. Перший містить $N \times M \times 3$ нелінійних параметри функції Гаусса, а третій – M лінійних параметрів c_i .

Нечіткі нейронні сітки (як Ванга-Менделя, так і TSK) можуть бути узагальнені на випадок багатьох вихідних змінних. Їх навчання, як і класичних НС, може виконуватися як з вчителем, так і без нього.

Навчання з вчителем базується на мінімізації цільової функції, яка визначається з використанням евклідової норми

$$E = 0,5 \sum [k=1:p] \left(z(X^{(k)}) - d^{(k)} \right)^2. \quad (22.17)$$

Навчання без вчителя базується на самоорганізації сітки, забезпечуючої кластеризацію вхідних даних.

Гібридний алгоритм навчання нейросіток TSK та Ванга-Менделя. Даний алгоритм придатний до обох наведених вище структур, проте далі наведена послідовність відносно більш загальних сіток TSK. Гібридний алгоритм навчання нечітких НС можна вважати варіантом гібридного алгоритму навчання радіально-базисних НС.

Алгоритм реалізується чергуванням двох етапів:

1. При фіксованих значеннях нелінійних параметрів c_{ij} , s_{ij} та q_{ij} першого шару нейронів відшукуються значення лінійних параметрів p_{ij} третього шару сітки;
2. При фіксованих значеннях лінійних параметрів p_{ij} третього шару уточнюються нелінійні параметри c_{ij} , s_{ij} та q_{ij} першого шару сітки.

Етап 1. На даному етапі навчання нелінійні параметри зафіксовані. Сигнал на виході визначається як

$$z(X) = \sum [i=1:M] \left(\alpha' * \left(p_{i0} + \sum [j=1:N] (p_{ij} * x_j) \right) \right), \quad (22.18)$$

$$\alpha' = \chi_i = \text{prod} [j=1:N] (mu_{ij}(x_j)) / \sum [l=1:M] \text{prod} [j=1:N] (mu_{lj}(x_j)) = \text{const.}$$

Для K навчаючих вибірок $\langle X^{(k)}, d^{(k)} \rangle$, $k=1,2,...,K$ отримаємо систему K лінійних рівнянь

$$\mathbf{H} * \mathbf{P} = \mathbf{D}, \quad (22.19)$$

де $\mathbf{P} = [p_{10}, p_{11}, ..., p_{1N}, ..., p_{M0}, p_{M1}, ..., p_{MN}]^T$ – вектор ваг третього шару сітки, а $\mathbf{D} = [d^{(1)}, d^{(2)}, ..., d^{(k)}]^T$ – вектор очікуваних значень, складений

з усіх K навчаючих вибірок. Матриця \mathbf{H} наводиться нижче:

$$\begin{array}{cccccccc} \chi_1^{(1)} & \chi_1^{(1)} * x_1^{(1)} & \dots & \chi_1^{(1)} * x_N^{(1)} & \dots & \chi_M^{(1)} & \chi_M^{(1)} * x_1^{(1)} & \dots & \chi_M^{(1)} * x_N^{(1)} \\ \chi_1^{(2)} & \chi_1^{(2)} * x_1^{(2)} & \dots & \chi_1^{(2)} * x_N^{(2)} & \dots & \chi_M^{(2)} & \chi_M^{(2)} * x_1^{(2)} & \dots & \chi_M^{(2)} * x_N^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \chi_1^{(k)} & \chi_1^{(k)} * x_1^{(k)} & \dots & \chi_1^{(k)} * x_N^{(k)} & \dots & \chi_M^{(k)} & \chi_M^{(k)} * x_1^{(k)} & \dots & \chi_M^{(k)} * x_N^{(k)} \end{array}$$

Кількість рядків K матриці \mathbf{H} значно більша за кількість її стовпців $M * (N + 1)$. Розв'язок цієї системи лінійних алгебраїчних рівнянь може бути отриманий за один крок наступним чином:

$$\mathbf{P} = \mathbf{H}^+ * \mathbf{D}, \quad (22.20)$$

де \mathbf{H}^+ – псевдоінверсія матриці \mathbf{H} .

Етап 2. На цьому етапі фіксуються значення коефіцієнтів поліномів третього шару і здійснюється уточнення (як правило, неодноразове) коефіцієнтів функції Гаусса для першого шару сітки стандартним методом градієнта:

$$\begin{aligned} c_{ij}^{(k+1)} &= c_{ij}^{(k)} - tu_c * \partial E^{(k)} / \partial c_{ij}^{(k)}, \\ s_{ij}^{(k+1)} &= s_{ij}^{(k)} - tu_s * \partial E^{(k)} / \partial s_{ij}^{(k)}, \\ q_{ij}^{(k+1)} &= q_{ij}^{(k)} - tu_q * \partial E^{(k)} / \partial q_{ij}^{(k)}, \end{aligned} \quad (22.21)$$

де k – номер чергового циклу навчання (в режимі “онлайн” співпадає з номером навчаючої вибірки). З технічної точки зору отримання аналітичних виразів для похідних цільової функції за нелінійними параметрами проблем не уявляє.

Оскільки етап уточнення нелінійних параметрів функції Гаусса має багато меншу швидкість збіжності, в ході навчання реалізацію етапу 1, як правило, супроводжує реалізація декількох етапів 2.

22.8. Нечіткі нейросітки з самоорганізацією

Сітки даного типу на етапі навчання здійснюють групування вхідних векторів $X^{(k)}$, $k = 1, 2, \dots, p$ в M кластерів, кожний з яких визначається центром C_i , $i = 1, 2, \dots, M$. На етапі класифікації НС ото-тожнює черговий вхідний вектор даних X з одним із раніше визначених кластерів.

Нечітка сітка з самоорганізацією має просту двошарову структуру, наведену на рис. 22.13.

Нейрони *першого шару* реалізують узагальнену функцію Гаусса в раціональній формі:

$$mu_{ij}(x_j) = 1 / \left(1 + \left((x_j - c_{ij}) / s_{ij} \right)^{2 * q_{ij}} \right). \quad (22.22)$$

Кожний нейрон *другого шару* характеризується центром $C_i = [c_{1i}, c_{2i}, \dots, c_{Ni}]^T$.

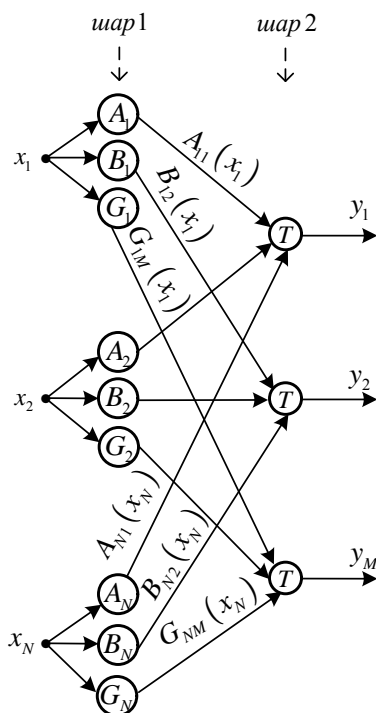


Рис. 22.13. Структура нечіткої сітки з самоорганізацією

Алгоритм нечіткої самоорганізації C-means. В даному алгоритмі подаваний на вхід черговий навчаючий вектор $X^{(k)}$ належить відмінним кластерам (представленим центрами $C_i, i = 1, 2, \dots, M$) в степені $u_i^{(k)}, 0 < u_i^{(k)} < 1$ при дотриманні умови

$$\sum [i = 1 : M] (u_i^{(k)}) = 1, \quad (22.23)$$

при цьому значення $u_i^{(k)}$ тим більше, чим ближче $X^{(k)}$ до C_i .

Похибка співвіднесення навчаючих векторів $X^{(k)}$ та центрів C_i для всіх p навчаючих векторів може бути подана наступним виразом:

$$E = \sum[i = 1 : M] \left(\sum[k = 1 : p] \left(\left(u_i^{(k)} \right)^m * |X^{(k)} - C_i|_2 \right) \right), \quad (22.24)$$

де m – показник, який вибирається з ряду 1, 2, 3, ... ; $|X^{(k)} - C_i|_2 = d_i^{(k)}$ – евклідова віддаль між $X^{(k)}$ та C_i .

Мета навчання – підбирання таких значень центрів C_i , які забезпечують мінімальне значення похибки E при одночасному дотриманні умови (22.23).

Розв’язання цієї задачі можна звести до мінімізації функції Лагранжа у виді:

$$LE = \sum[i = 1 : M] \left(\sum[k = 1 : p] \left(\left(u_i^{(k)} \right)^m * |X^{(k)} - C_i|_2 \right) \right) + \\ + \sum[k = 1 : p] \left(L^k * \left(\sum[i = 1 : M] \left(u_i^{(k)} \right) - 1 \right) \right), \quad (22.25)$$

де $L^{(k)}$, $k = 1, 2, \dots, p$ – множники Лагранжа.

Доведено, що розв’язання цієї задачі можна подати у виді:

$$C_i = \sum[k = 1 : p] \left(\left(u_i^{(k)} \right)^m * X^{(k)} \right) / \sum[k = 1 : p] \left(\left(u_i^{(k)} \right)^m \right), \\ u_i^{(k)} = 1 / \sum[l = 1 : M] \left(\left(\left(d_i^{(k)} \right)^2 / \left(d_l^{(k)} \right)^2 \right)^{1/(m-1)} \right), \quad (22.26)$$

де $d_i^{(k)} = |X^{(k)} - C_i|_2$ – евклідова віддаль між $X^{(k)}$ та C_i .

Алгоритм навчання, який реалізує наведений вище підхід та отримав назву *C-means*, є ітераційним і може бути описаний наступною послідовністю:

Крок 1. Виконати випадковий вибір коефіцієнтів $u_i^{(k)}$ з діапазону $[0, 1]$ при дотриманні умови (22.23).

Крок 2. Обчислити всі M центрів C_i за наведеною формулою (22.26).

Крок 3. Розрахувати значення похибки E за формулою (22.24). Якщо це значення менше встановленого порога або незначно змінилося відносно попередньої ітерації, то закінчити обчислення. Інакше перейти до кроку 4.

Крок 4. Розрахувати нові значення $u_i^{(k)}$ за формулою (22.26) та перейти до кроку 2.

Описаний вище ітераційний алгоритм веде до досягнення мінімуму похибки E , який необов’язково буде глобальним мінімумом.

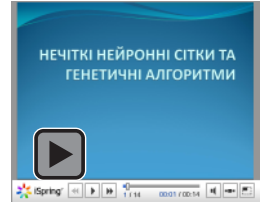
На ймовірність відшукування глобального мінімуму впливає вибір початкових значень $u_i^{(k)}$ та C_i . Спеціально для підбирання “задовольняючих” початкових значень центрів C_i розроблені процедури ініціалізації, зокрема, *алгоритм пікового групування* та *алгоритм різницевого групування* [9].

Контрольні завдання

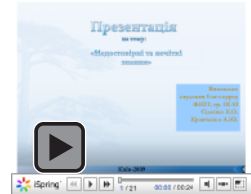
1. Розкрити функції складових в узагальненій структурі контуру управління фаззі-контролером при виконанні процедури фаззіфікації параметрів процесу.
2. Пояснити принципи утворення min-max-операцій у фаззі-контролері.
3. Навести і прокоментувати означення нечіткої НС.
4. Охарактеризувати НС з фаззі-структурою і пояснити принципи функціонування фаззіфікуючого нейрона при реалізації операцій “І” та “АБО”.
5. Розкрити зміст логічного розв’язання в нейро-фаззі-системі.
6. Дати інтерпретацію відображення дефаззіфікації у НС з комплексним нейро-фаззі-перетворенням.
7. Пояснити, що являє собою інтегрована нейро-фаззі-система.
8. Навести і прокоментувати алгоритм тренування нечіткої НС.
9. Прокоментувати структуру адаптивної системи з нейро-фаззі-виведеннями.
10. Розкрити особливості функціонування нечіткої моделі нейронної сітки Такагі-Сугено-Канга і охарактеризувати проблему стійкості TSK-сіток.
11. Навести особливості структури сітки Ванга-Менделя і визначити відмінність її від нейронної сітки Такагі-Сугено-Канга.
12. Охарактеризувати гібридний алгоритм навчання нейросіток TSK та Ванга-Менделя.
13. Навести відмінності структури і алгоритму навчання *C-means* нечіткої сітки з самоорганізацією.

Презентації

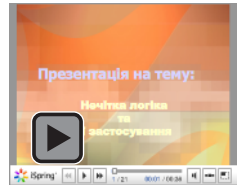
1. Нечіткі нейронні сітки
та генетичні алгоритми



2. Недостовірні та нечіткі
знання



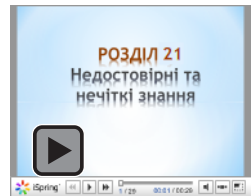
3. Нечітка логіка та її
застосування



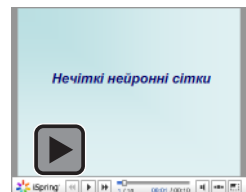
4. Нейро-фаззі технології в
системах із штучним
інтелектом



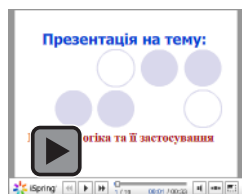
5. Недостовірні та нечіткі
знання



6. Нечіткі нейронні сітки



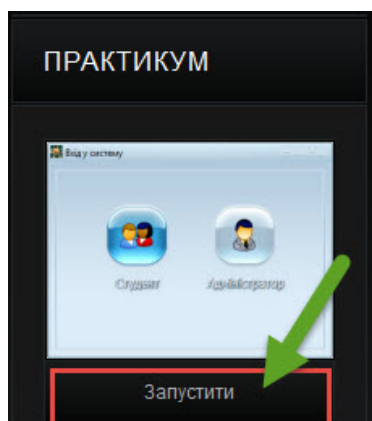
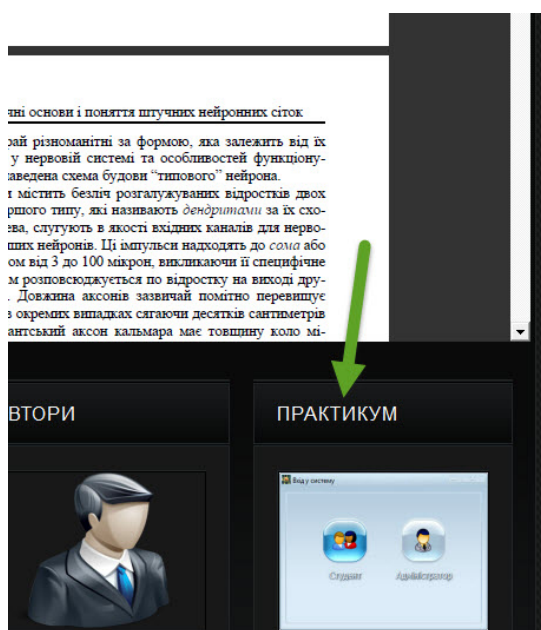
7. Нечітка логіка та її
застосування



Практичні заняття

Нейро-фаззі-технології

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Розділ 23

Методи спрощення нейронних сіток

*Не стоит выплескивать
из ванны с грязной водой и самого ребенка*
Мартин Лютер

*Жінки витрачають гроші з розумом.
В результаті – ані грошей, ані розуму*
Народне прислів'я

23.1. Мета спрощення нейронних сіток

Цілі вербалізації. Практичне застосування ШНС в значній мірі залежить від складності останніх, в зв'язку з чим як при проектуванні нових, так і при виборі існуючих топологій сіток під певну прикладну задачу прагнуть досягти їх максимального спрощення.

Крім того, одним з основних недоліків навчуваних ШНС з точки зору багатьох користувачів є те, що з навченої нейронної сітки важко витягти явний і зрозумілий користувачу алгоритм розв'язання задачі – сама ШНС є таким алгоритмом, і якщо структура сітки складна, то цей алгоритм незрозумілий. Проте, спеціальним чином побудована процедура *спрощення і вербалізації* часто дозволяє витягти явний метод розв'язання.

Означення 23.1. *Вербалізація* – це мінімізований опис роботи синтезованої і вже навченої нейронної сітки у виді декількох взаємозалежних алгебраїчних або логічних функцій.

Вербалізація здійснюється, зокрема, для підготовки навченої та спрощеної нейросітки до реалізації в програмному коді або у виді спеціалізованого електронного (оптоелектронного) пристрою, а також для використання результатів в вигляді явних знань. Під *симптомами* при цьому розуміють вхідні значення нейросітки, а під *синдромами* – значення на виходах нейронів. Кінцевий синдром – це значення на виході НС. Вербалізація звичайно здійснюється засобами спеціалізованих пакетів.

Мета проріджування нейросіток. Перед вербалізацією, як правило за допомогою продукційних правил, для деяких видів НС було запропоновано спрощувати структуру сіток – *проріджувати*.

Твердження 23.1. *Основна ідея проріджування* (англ. *pruning*) полягає в тому, що ті елементи моделі або ті нейрони сітки, які чинять малий вплив на похибку апроксимації, можна виключити з моделі без значного погіршення якості апроксимації.

Але при цьому треба мати на увазі, що *твердження 23.1* є справедливим тільки для розв'язуваної задачі. Якщо ж з'являться нові статистичні дані для навчання, то проріджена НС втратить спроможність до узагальнення, якою вона володіла б якщо б зв'язки не були втрачені (як мінімум зворотне не було доведено). Таким чином, мова йде про алгоритми з втратою якості, які можуть застосовуватися для поодиноких задач, але не можуть застосовуватися поза залежності від задачі.

До часто уживаних задач спрощення та вербалізації НС можна віднести:

1. Спрощення архітектури нейронної сітки.
2. Зменшення кількості вхідних сигналів.
3. Зведення параметрів НС до невеликої кількості виокремлених значень.
4. Зниження вимог до точності вхідних сигналів.
5. Формулювання явних знань у вигляді симптом-синдромної структури та явних формул формування синдромів з симптомів.

23.2. Основні напрями спрощення нейронних сіток

Класифікація методів спрощення нейронних сіток. Існує два напрями спрощення НС за рахунок (рис. 23.1): мінімізації кількості елементів вагової матриці сітки (*weight pruning*) – наведений суцільною лінією; зменшення кількості використовуваних нейронів (*unit pruning*) – наведений пунктирною лінією.

У першому випадку – мінімізації кількості елементів вагової матриці сітки (ще – видалення ваг) – прирівнюються нулю або елементи матриці ваг, які мають мінімальні значення (так звана ідея *редукції*, чи *регуляризації НС*) [57], або ті елементи, наявність яких в найменшому ступені впливають на зменшення похибки сітки. Найбільш розповсюдженими представниками цих методів є “метод мозкового руйнування”, або *OBD-метод* (*Optimal Brain Damage*) [164], а також “метод мозкового відтворення”, або *OBS-метод* (*Optimal Brain Surgeon*) [130].

Другий випадок – зменшення кількості використовуваних нейронів – менш розповсюджений. Застосовуваний для цього метод

скелетонізації (*skeletonization*) з видаленням нейронів з урахуванням їх важливості придатні для видалення нейронів як вхідного, так і схованих шарів НС [173], а заснований на аналогічному до *weight-decay*-підході з використанням вартісної функції метод придатний для видалення нейронів схованого шару [128].

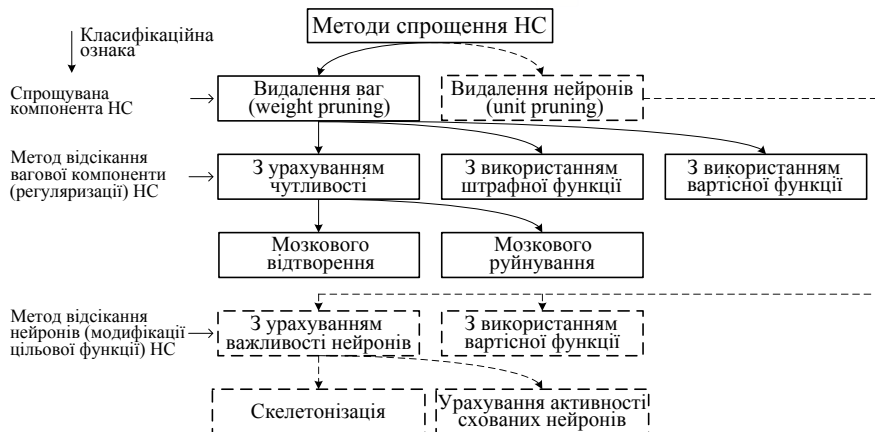


Рис. 23.1. Класифікація методів спрощення НС

23.3. Методи відсікання вагової компоненти нейросіток

23.3.1. Редукція нейронних сіток

з урахуванням чутливості

Загальні особливості регуляризації НС. Редукція НС виконується для зменшення кількості у схованих нейронів міжнейронних зв'язків. Оскільки кожний схований нейрон уявляє гіперплощину, поділяючу множину даних на кластери, редукція не спрощує такий поділ і підсилює здібність до узагальнення.

Найпростішим критерієм редукції вважається урахування величини ваг: ваги, які значно менші за середні, впливають незначною мірою на загальний рівень сигналу на виході зв'язаного з ним нейрона. Тому їх можна видаляти без суттєвої шкоди для його функціонування.

Проте, в деяких випадках невеликі значення ваг не обов'язково чинять найменший вплив на поведінку нейрона. В таких ситуаціях їх відсікання може привести до серйозних змін у роботі НС. Тому найкращим критерієм слід визнати урахування чутливості НС до варіацій ваг [43]. Без суттєвих наслідків для НС з неї можна виключити тільки ті ваги, чутливість до змін яких є мінімальною.

Підходом щодо проблеми видалення ваг є розкладання цільової функції в ряд Тейлора. У відповідності з ним вимірювання величини цільової функції, викликане варіацією ваг, можна подати залежністю:

$$\Delta I = \sum_i g_i \Delta w_i + 0,5 \left[\sum_i h_{ii} [\Delta w_{ii}]^2 + \sum_{i \neq j} h_{ij} \Delta w_i \Delta w_j \right] + U(\|\Delta w\|^2), \quad (23.1)$$

де Δw_i – варіація i -ї ваги; g_i – i -та складова вектора градієнта відносно

цієї ваги; $g_{ij} = \frac{\partial I}{\partial w_i}$; $h_{ij} = \frac{\partial^2 I}{\partial w_i \partial w_j}$ – елементи гессіану.

Означення 23.2. Гессіан функції – симетрична квадратична форма, яка описує поведінку функції у другому порядку. Гессіаном також часто називають і визначник матриці (a_{ij}) , тобто матрицю Гессе квадратичної форми, утворену другими частинними похідними функції.

Не рекомендується видаляти ваги в процесі навчання через те, що низька чутливість НС до конкретної ваги може бути пов'язана з її поточним значенням або з невдало обраною початковою точкою (наприклад, при “застряванні” нейрона в зоні глибокого насичення). Тому рекомендується видаляти ваги (тобто виконувати *регуляризацию* НС) тільки по завершенні процесу навчання, коли всі нейрони набувають своїх постійних характеристик. Це виключає застосування градієнта як показника чутливості, оскільки мінімум цільової функції характеризується нульовим значенням градієнта. Саме тому в якості показника важливості конкретних ваг доводиться використовувати другі похідні цільової функції – елементи гессіану.

Всі методи спрощення НС за рахунок відсікання вагової компоненти (weight decay) так чи інакше базуються на загальній ідеї регуляризації [57], коли в звичайний мінімізаційний функціонал вводиться додатковий штрафний член

$$I^{(d)} = 0,5 \sum_p \sum_j (y_{pj}^* - y_{pj})^2 + \frac{\chi}{2} \sum_i \sum_j w_{ij}^2, \quad (23.2)$$

де $\chi > 0$ – коефіцієнт штрафу; $p = \overline{1, P}$ – кількість подаваних образів навчання.

Мінімізація функціоналу (23.2) приводить до наступного алгоритму корекції ваг [48]:

$$\Delta_p w_{ij}(k+1) = -\beta (y_{pj}^*(k) - y_{pj}(k)) y'_{pj}(k) + \chi w_{nij}(k), \quad (23.3)$$

де $y'_{pj}(k) = \frac{\partial y_{pj}(k)}{\partial w_{ij}(k)}$; β – коефіцієнт швидкості навчання.

При цьому вага зі значенням в заданому ϵ -околі нуля видаляється.

Метод мозкового руйнування. Одним з кращих способів регуляризації НС вважається OBD-метод, запропонований Ле Куном [164]. Вихідна позиція цього методу – розкладання цільової функції в ряд Тейлора в околі поточного розв'язку.

Цей метод дозволяє на основі аналітичного передбачення впливу змінення вектора вагових параметрів на використовуваний функціонал похибки так скоригувати вектор ваг, щоб при відсіканні деяких з них значення функціоналу зростало у незначній мірі.

Для спрощення задачі в [164] пропонується виходити з того, що внаслідок додатної визначеності гессіану матриця **H** є діагонально домінуючою. Саме тому можна враховувати тільки діагональні елементи

$$h_{qq} = \frac{\partial^2 I}{\partial w_{ij}^2} \quad (23.4)$$

та ігнорувати всі інші.

Як міра значущості ваги w_{ij} в OBD-методі використовується показник S_{ij} , називаний *коефіцієнтом асиметрії* (англ.: *soliency*), який визначається залежністю:

$$S_{ij} = 0,5h_{qq}w_{ij}^2. \quad (23.5)$$

Видалення ваг з найменшими значеннями показника S_{ij} не викликає суттєвих змін в процесі функціонування НС, і цю процедуру редукції останньої можна подати послідовністю:

Крок 1. Повне попереднє навчання НС вибраної структури з використанням будь-якого алгоритму.

Крок 2. Визначення діагональних елементів гессіану (23.4), відповідних кожній вазі, та розрахунок значень параметру S_{ij} з (23.5), який характеризує залежність кожного синаптичного зв'язку для НС в цілому.

Крок 3. Сортування ваг в порядку спадання приписаних до них параметрів S_{ij} та видалення тих, які мають найменші значення.

Крок 4. Повертання до Кроку 1 для навчання сітки з редукованою структурою та повторення процесу відсікання аж до виключення усіх ваг з найменшим впливом на величину цільової функції.

Наведений OBD-метод вважається одним з найкращих способів редукції сітки серед методів урахування чутливості. Його застосування забезпечує досягнення сіткою високого рівня узагальнення і лише незначною мірою відрізняється від рівня похибки навчання. Особливо прийнятні результати дає повторне навчання НС після видалення найменш значущих ваг.

Проте є випадки, коли таке спрощення матриці Гессе не дозволяє отримати структуру мінімальної складності, а в деяких випадках, як наприклад, при розв'язанні проблеми “виключаючого АБО” взагалі можуть бути видалені не ті ваги.

Метод мозкового відтворення. Подальшим розвитком OBD-методу вважається OBS-метод, запропонований Б. Хассібі та Д. Шторком [130] трьома роками пізніше. Відправним посиланням цього методу (як і методу OBD) також є розкладання цільової функції в ряд Тейлора та ігнорування членів першого порядку. В цьому методі враховуються всі компоненти гессіану, а коефіцієнт асиметрії ваги визначається залежністю:

$$S_i = 0,5 \frac{w_i^2}{[\mathbf{H}^{-1}]_{ii}}, \quad (23.6)$$

де для позбавлення завантаженості індексацією вага w_{ki} позначається поодиноким індексом як w_i .

Як і раніше, видаленню підлягає вага з найменшим значенням S_i . Результатом такого підходу є нескладна формула корекції залишившихся ваг, яка дозволяє повернути сітку в стан з мінімальною цільовою функцією, не дивлячись на видалення ваги, а саме [130]:

$$\Delta w = \frac{w_i}{[\mathbf{H}^{-1}]_{ii}} \mathbf{H}^{-1} \mathbf{p}_i, \quad (23.7)$$

де \mathbf{p}_i – поодинокий вектор з одиницею в i -й позиції, тобто $\mathbf{p}_i = [0, \dots, 0, 1, \dots, 0]^T$. Корекція виконується після видалення кожної чергової ваги і замінює повторне навчання НС, необхідне в разі використання OBD-методу. Процедура OBS-методу регуляризації можна подати наступними кроками [130]:

Крок 1. Навчання НС попередньо вибраної структури НС аж до відшукування локального мінімуму цільової функції.

Крок 2. Обрахування оберненої гессіану матриці \mathbf{H}^{-1} та вибір ваги w_i з найменшим значенням показника (23.4). Якщо зміна величини цільової функції в результаті видалення цієї ваги набагато менша значення I , вага w_i видаляється і здійснюється перехід до Кроку 3, в противному разі видалення закінчується.

Крок 3. Корекція значень ваг, залишившихся у сітці після видалення i -тої ваги у відповідності до (23.5) з наступним поверненням до Кроку 2. Процес продовжується аж до видалення усіх малозначущих ваг.

Основні відмінності OBS-методу від OBD-методу, крім іншого визначення коефіцієнта асиметрії, полягають в наступному:

- корекції ваг після видалення найменш важливої ваги без повторного навчання сітки;
- можливості на кожному кроці видалення тільки однієї ваги (OBD-метод припускає можливість відсікання довільної кількості ваг);
- значно вищій обчислювальній складності – розрахунок діагональних елементів гессіану (у OBD-методі) тут замінений обчисленням повної матриці та оберненої до неї форми. На практиці цей етап можна значно скоротити при використанні апроксимованої форми матриці, оберненої гессіану, яка визначається, наприклад, методом змінної метрики. Проте таке спрощення викликає зниження точності розрахунків і дещо погіршує якість шуканого розв'язку.

Слід зазначити, що жодна вага не може видалятися, якщо після її видалення функціонал похибки суттєво зростає.

Даний метод забезпечує найкращі результати, якщо функціонал похибки в точці мінімуму є квадратичним.

23.3.2. Редукція нейронної сітки з урахуванням штрафної функції

Іншим методом регуляризації ваг є така організація процесу навчання, яка провокує самостійне зменшення значень ваг і дозволяє виключити ті з них, величина яких опуститься нижче встановленого порогу. На відміну від методів врахування чутливості в обговорюваних методах сама цільова функція модифікується таким чином, щоб в процесі навчання значення ваг мінімізувалось автоматично аж до досягнення визначеного порогу, при перетинанні якого значення відповідних ваг прирівнюються до нуля.

Найпростіший метод модифікації цільової функції передбачає додавання в неї доданку, штрафуючого за великі значення ваг:

$$I(w) = I^{(B)}(w) + \chi \sum_{ij} w_{ij}^2, \quad (23.8)$$

де $I^{(B)}(w)$ – стандартно визначена цільова функція (наприклад, у виді евклідової норми); χ – коефіцієнт штрафу за набуття вагами великих значень.

При цьому кожний з циклів навчання складається з двох етапів: мінімізації величини функції $I^{(B)}(w)$ стандартним методом зворотного поширення; корекції значень ваг, обумовленої модифікуючим фактором. Якщо через $w_{ij}^{(B)}$ позначити значення ваги w_{ij} після першого етапу навчання, то в результаті корекції ця вага буде модифікована за градієнтним методом найшвидшого спуску за формулою:

$$w_{ij} = w_{ij}^{(B)} (1 - \beta \chi), \quad (23.9)$$

де β – константа навчання.

Визначена у такий спосіб *штрафна функція* викликає зменшення значень ваг навіть тоді, коли з урахуванням специфіки розв'язуваної задачі окремі ваги повинні мати великі значення. Рівень значення, за яким вага може видалятися, має підбиратися на основі численних експериментів для відшукування порогу, при якому процес навчання НС піддається найменшим збуренням.

Більш придатні результати, які не викликають зменшення значень усіх ваг, можна отримати модифікацією подання цільової функції у формі:

$$I(w) = I^{(B)}(w) + 0,5\chi \sum_{ij} (w_{ij}^2 / (1 + \sum_q w_q^2)). \quad (23.10)$$

Мінімізація цієї функції викликає не тільки редукцію між-нейронних зв'язків, але й може привести до виключення нейронів, для яких величина $\sum_q |w_{iq}|$ близька до нуля. В цьому випадку правило корекції ваг:

$$w_{ij} = w_{ij}^{(B)} \left[1 - \beta \chi \frac{1 + 2 \sum_{q \neq j} (w_{iq}^{(B)})^2}{1 + 2 \sum_q (w_{iq}^{(B)})^2} \right] \quad (23.11)$$

При малих значеннях ваг w_{iq} , надходячих до i -го нейрона, відбувається подальше їх зменшення. Це веде до послаблення сигналу на виході до нуля і як наслідок – до виключення його з НС. При великих значеннях ваг, які ведуть до i -го нейрона, їх корекційна складова зникаюче замала і мало впливає на процес редукції сітки.

23.3.3. Редукція нейросіток з використанням вартісної функції

В роботі [96] запропоновано в якості штрафної функції, аналогічної до (23.2), застосовувати *квадратичну вартісну функцію* сигналів на виході нейронів схованого шару

$$B = \sum_j B(y_j)^2, \quad (23.12)$$

де $B(y_j)$ – монотонна функція виходу j -го нейрона.

Тоді мінімізований функціонал приймає вид:

$$I^{(B)} = \zeta^{(I)} I + \zeta^{(B)} B, \quad (23.13)$$

де $\zeta^{(I)}$, $\zeta^{(B)}$ – вагові коефіцієнти частинних критеріїв.

Для зменшення вартості схованих шарів, розташованих ближче до шару на виході, обчислюють похідні

$$\frac{\partial B}{\partial w_{ij}} = \frac{\partial B}{\partial z_j} \frac{\partial z_j}{\partial w_{ij}} = v_j^{(B)} \frac{\partial z_j}{\partial w_{ij}} = v_j^{(B)} y_i, \quad (23.14)$$

де

$$v_j^{(B)} = \frac{\partial B(y_j^2)}{\partial z_j} = \frac{\partial B(y_j^2)}{\partial y_j^2} \frac{\partial y_j^2}{\partial y_j} \frac{y_j}{\partial z_j} = 2B'(y_j) f'_j(z_j), \quad (23.15)$$

де $B' = \frac{\partial B(y_j^2)}{\partial y_j^2}$; $f'_j(z_j)$ – похідна активаційної (логістичної) функції

входів НС.

Для ваг усіх шарів, розташованих після розглядуваного, вартісна функція дорівнює нулю. Якщо ж аналізованому шару передуює схований, то для цього передуючого шару

$$v_j^{(B)} = f'_j(nz_j) \sum_l v_l^{(B)} w_{jl}, \quad (23.16)$$

що співпадає зі стандартним алгоритмом зворотного поширення помилки, який відрізняється від нього тільки зворотно поширюючим членом – сумарним сигналом похибки. Вираз для корекції ваг з врахуванням зворотно поширюваних похибок і вартості набуває виду:

$$\Delta w_{ij} = -\beta \zeta^{(I)} y_i v_j^{(I)} - \beta \zeta^{(B)} y_i v_j^{(B)} = -\beta y_i v_j^{(I,B)}, \quad (23.17)$$

де $v_j^{(I,B)}$ – сумарний сигнал похибки, який враховує як похибку сітки, так і вартість сигналу на виході j -го нейрона.

Метод може бути модифікований шляхом його комбінування з weight-decay-методом (23.2) наступним чином:

$$\bar{I} = \zeta^{(I)} I + \zeta^{(B)} B + \zeta^{(w)} W. \quad (23.18)$$

23.4. Спрощення нейросіток видаленням нейронів

Видалення нейронів так чи інакше оцінюється наслідками, які відбиваються на значеннях величини помилкової реакції нейросітки.

23.4.1. Відсікання нейронів з урахуванням їх важливості

Метод скелетонізації. Метод скелетонізації [173] з видаленням нейронів як вхідного, так і схованих шарів базується на використанні у функціоналі похибки додаткового члена – показника ϑ важливості нейрона, який визначається різницею між похибкою всієї НС та похибкою сітки, з якої видалений даний нейрон. Якщо важливість кожного нейрона НС, складеної з n нейронів, визначати у такий спосіб (прийmemo до уваги, що для кожного нейрона необхідні P образів), то ця ситуація вимагала б $H(nP)$ пред'явлень навчаючих образів. Для скорочення кількості обчислень при визначенні ϑ використовується наближена формула, для виведення якої введений параметр впливу (attention strength) $\alpha_i (i = \overline{1, n})$ кожного з нейронів [48]

$$y_j = f \left(\sum_i w_{ij} \alpha_i y_i \right). \quad (23.19)$$

Якщо $\alpha_i = 0$, то нейрон не впливає на сітку, проте при $\alpha_i = 1$ він відповідає звичайному нейрону (див. рис. 23.2).

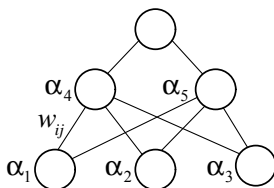


Рис. 23.2. Нейросітка з коефіцієнтами впливу

Подавши значущість нейронів як

$$\vartheta_i = I_{\alpha_i=0} - I_{\alpha_i=1}, \quad (23.20)$$

Можна апроксимувати коефіцієнт ϑ_i похідною $\partial I / \partial \alpha_i$ в точці $\alpha_{i=1}$

$$\lim_{\beta \rightarrow 1} \frac{I_{\alpha_i=0} - I_{\alpha_i=1}}{\beta - 1} = \left. \frac{\partial I}{\partial \alpha_i} \right|_{\alpha_i=1}. \quad (23.21)$$

Якщо припустити, що похідна в точці $\alpha_i = 1$ дає придатне наближення лівої частини рівняння та у випадку, коли коефіцієнт штрафу $\chi = 0$, можна замість ϑ_i використовувати її оцінку $\bar{\vartheta}_i$

$$\bar{\vartheta}_i = \left. \frac{\partial I}{\partial \alpha_i} \right|_{\alpha_i=1}. \quad (23.22)$$

Похідна (23.22) може бути обчисленою методом, аналогічним до методу зворотного поширення похибки.

На практиці похідна $\partial I / \partial \alpha_i$ швидко спадає з часом, через що навіть середнє значення $\bar{\vartheta}_i$ зменшується експоненційно. В [173] запропоновано проводити корекцію $\bar{\vartheta}_i$ у відповідності з залежністю:

$$\bar{\vartheta}_i(k+1) = 0,8\bar{\vartheta}_i(k) + 0,2 \frac{\partial I}{\partial \alpha_i}. \quad (23.23)$$

Такий підхід дозволяє, на відміну від обрахування *квадратичної похибки*, яке використовується при настроюванні ваг, для визначення значущості застосовувати *лінійний функціонал*:

$$I_{linear} = \sum_n \sum_j \left| (y_{nj}^* - y_{nj}) \right|. \quad (23.24)$$

Як показано в [173], звичайна квадратична функція від похибки забезпечує отримання поганих оцінок, коли значення реальних виходів близькі до бажаних. Тому в даній роботі для настроювання ваг використовувався квадратичний критерій, а для оцінювання значущості – лінійний.

Алгоритм скелетонізації подається наступною черговою кроків:

Крок 1. Навчання сітки доти, доки всі виходи усіх нейронів не стануть достатньо близькими до бажаних.

Крок 2. За залежностями (23.22) або (23.23) вираховується значущість $\bar{\vartheta}_i$ кожного нейрона.

Крок 3. Видаляється нейрон з найменшою значущістю.

Крок 4. Зупинка, якщо критерій зупинки задовольняється; в іншому випадку – перехід до *Кроку 1*.

Урахування активності схованих нейронів. Цей підхід базується на такій модифікації цільової функції, яка дозволяє виключити сховані нейрони, в найменшому ступені змінюючи свою активність в процесі навчання. При цьому враховується, що якщо сигнал на виході будь-якого нейрона при будь-яких навчаючих вибірках залишається незмінним (на його виходах постійно формується 1 або 0), то його присутність у нейросітці надлишкова. І навпаки, при високій активності нейрона вважається, що його функціонування надає важливу інформацію. В роботі [96] запропоновано наступну модифікацію цільової функції:

$$I(w) = I^{(B)}(w) + v \sum_{i=1}^N \sum_{j=1}^P e(\Delta_{ij}^2), \quad (23.25)$$

де $I^{(B)}(w)$ – стандартно визначена цільова функція; v – коефіцієнт ступеня відносного впливу коректуючого фактора на значення цільової функції; Δ_{ij} – зміна значення сигналу на виході i -го нейрона для j -ї навчаючої вибірки; Δ_{ij}^2 – коректуючий фактор цільової функції, який залежить від активності усіх N схованих нейронів для всіх j ($j = 1, 2, \dots, P$) навчаючих вибірок. Вид коректуючої функції підбирається так, щоб змінення цільової функції залежало від активності схованого нейрона, причому, при високій його активності (тобто частих змінах значень сигналу на виході) величина ΔI повинна бути малою, а при низькій активності – великою. Це досягається застосуванням функції e' , яка задовольняє відношення:

$$e' = \frac{\partial e(\Delta_i^2)}{\partial \Delta_i^2} = \frac{1}{(1 + \Delta_i^2)^z}, \quad (23.26)$$

де індекс z дозволяє керувати процесом штрафування за низьку активність. Мала активність нейронів карається сильніше, що в результаті може привести до повного виключення пасивних нейронів з сітки.

23.4.2. Видалення нейронів з урахуванням вартісної функції

Для видалення нейронів схованого шару С. Хенсоном і А. Праттом [128] використаний аналогічний weight-decay-методу (23.2) підхід з доповненням критерію I вартісним показником В

$$I^{(B)} = I + B. \quad (23.27)$$

Корекція ваг здійснюється з умови $\min_w I^{(B)}$.

Вид алгоритму корекції залежить від виду вартісної функції В. Так, якщо $B = \sum_i \sum_j w_{ij}^2$ – квадратична функція відносно w_{ij} , то *алгоритм корекції ваг* визначається видом [48]:

$$\Delta w_{ij}(k+1) = \beta \left(-\frac{\partial I}{\partial w_{ij}} - 2w_{ij}(k) \right), \quad (23.28)$$

звідки

$$w_{ij}(k) = \beta \sum_{m=1}^k \left[(1 - 2\eta)^{k-m} \left(-\frac{\partial I(m)}{\partial w_{ij}} \right) \right] + (1 - 2\beta)^k w_{ij}(0). \quad (23.29)$$

Як вартісну Д. Руммельхарт запропонував наступну функцію:

$$B = \sum_i \sum_j \frac{w_{ij}^2}{1 + w_{ij}^2}, \quad (23.30)$$

використання якої в алгоритмі похідних

$$\frac{\partial B}{\partial w_{ij}} = \frac{2w_{ij}}{(1 + w_{ij}^2)^2} \quad (23.31)$$

дозволяє швидко повертати в нуль малі ваги, а великі – змінювати незначно. Перехід від урахування вартості ваги до урахування вартості нейрона відбувається шляхом підсумовування значень ваг кожного зі схованих нейронів:

$$w_j = \sum_i |w_{ij}|. \quad (23.32)$$

Поряд з видозміненою формою функції (23.30)

$$B = \sum_j \frac{w_j^2}{1 + w_j^2} \quad (23.33)$$

з похідною

$$\frac{\partial B}{\partial w_{ij}} = \frac{2w_j \operatorname{sgn}(w_{ij})}{(1 + w_j^2)^2} \quad (23.34)$$

застосовуються також гіперболічна

$$B = \sum_j \frac{w_j}{1 + \lambda w_j} \quad (23.35)$$

з похідною

$$\frac{\partial B}{\partial w_{ij}} = \frac{\lambda \operatorname{sgn}(w_{ij})}{(1 + w_j)^2} \quad (23.36)$$

та спадна експоненційна функція

$$B = \sum_j (1 - e^{-\lambda w_j}) \quad (23.37)$$

з похідною

$$\frac{\partial B}{\partial w_{ij}} = \frac{\lambda \operatorname{sgn}(w_{ij})}{e^{\lambda w_j}}. \quad (23.38)$$

Слід приймати до уваги, що в деяких задачах експоненційна спадна функція може приводити до того, що процес навчання не буде збігатися. Скоріш за все це відбувається через використання градієнта функціоналу як для зменшення похибки сітки, так і для видалення схованого нейрона.

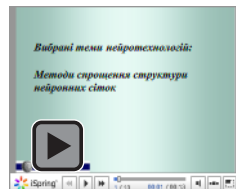
Контрольні завдання

1. Розкрити мету вербалізації та проріджування нейронних сіток та їх зв'язок з редукцією останніх.

2. Охарактеризувати класифікаційні ознаки щодо методів спрощення нейронних сіток.
3. Розкрити смутність двох методів спрощення нейронних сіток.
4. Проаналізувати загальні особливості регуляризації нейронних сіток.
5. Розкрити особливості методу мозкового руйнування (OBD-методу Ле-Куна) як способу регуляризації нейронних сіток.
6. Навести послідовність процесу видалення міжнейронних зв'язків з найменшими значеннями ваг у нейронних сітках.
7. Охарактеризувати метод мозкового відтворення (OBS-метод) як подальший розвиток методу мозкового руйнування.
8. Навести процедуру мозкового відтворення (OBS-методу) регуляризації нейронних сіток.
9. Прокоментувати метод регуляризації ваг нейронної сітки з урахуванням штрафної функції.
10. Пояснити можливість редукції нейронної сітки з використанням в якості штрафної – квадратичної вартісної функції.
11. Дати характеристику методу скелетонізації як механізму відсікання нейронів з урахуванням їх важливості в нейронній сітці.
12. Навести алгоритм скелетонізації за результатами вирахування значущості кожного з нейронів в нейронній сітці.
13. Розкрити метод редукції нейронних сіток з урахуванням активності схованих нейронів.
14. Охарактеризувати можливість спрощення нейронної сітки шляхом видалення нейронів її схованого шару з урахуванням вартісної функції.

Презентації

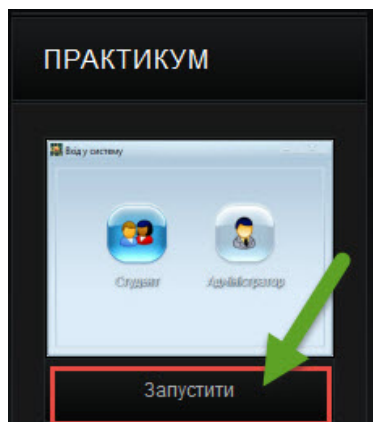
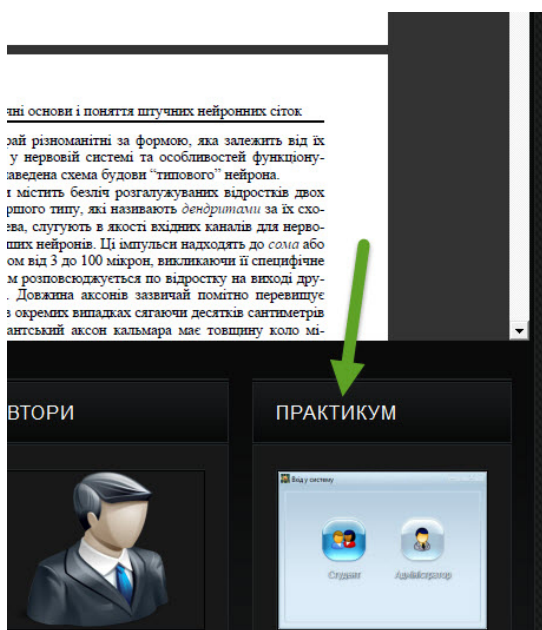
1. Методи спрощення структури нейронних сіток



Практичні заняття

Методи спрощення нейронних сіток

Для запуску практичних занять на
вкладці "Практикум" натисніть
"Запустити"



Розділ 24

Мультиагентні системи

*Необхідність є нещастям,
проте немає ніякої необхідності жити з необхідністю*
Епікур Самосський

*Агентом можешь ты не быть,
но интеллект иметь обязан...*
Напутствие неагенту

24.1. Мультиагентні технології

Задачі керування та розподіленої взаємодії в мережах динамічних систем привертають в останнє десятиріччя увагу все більшої кількості дослідників. В багатьох випадках це пояснюється широким застосуванням *багатоагентних*, або *мультиагентних систем* (МАС, англ. *Multi-agent system*) в різних галузях, включаючи автоматичне підналагодження параметрів нейронних сіток розпізнавання, розподілені сенсорні мережі, управління перевантаженням в мережах зв'язку, відносно вирівнювання груп супутників, керування рухом груп мобільних роботів, синхронізацію в енергосистемах тощо.

Означення 24.1. *Мультиагентна система* – це нова парадигма інформаційної технології, орієнтованої на сумісне використання наукових і технічних досягнень та переваг, яких надають ідеї та методи штучного інтелекту, сучасних локальних та глобальних комп'ютерних мереж, розподілених баз даних і розподілених обчислень для розв'язання складної задачі або проблеми з використанням систем, утворених з багатьох взаємодіючих інтелектуальних агентів. Іншими словами, МАС можуть використовуватися для розв'язання проблем, які складно або неможливо розв'язати за допомогою одного агента або монолітної системи.

Початок створення моделей і застосування штучних МАС на практиці було покладено в 1960-х роках. За основу були взяті досягнення таких галузей діяльності людини, як системи *штучного інтелекту* (*Artificial Intelligence*), *паралельні обчислення* (*Parallel Computing*), *розподілене розв'язання задач* (*Distributed Problem Solving*).

В класичній теорії штучного інтелекту розв'язання певної задачі зводиться до створення якоїсь однієї інтелектуальної системи – агента, яка, маючи в своєму розпорядженні всі необхідні знання, спроможності та обчислювальні ресурси, здатна розв'язати деяку глобальну проблему.

Сьогодні все частіше використовуються розподілені системи, виконуючі певні дії паралельно, для яких актуальною є задача розділення пакету завдань між декількома обчислювальними потоками (пристроями). Подібні задачі виникають не тільки в обчислювальних мережах, але також і у виробничій сфері, мережах обслуговування, транспортних, логістичних мережах тощо. Виявляється, що за природних обмежень на зв'язки децентралізовані стратегії спроможні ефективно розв'язувати такого типу задачі. Для розв'язання подібних проблем все частіше використовуються *мультиагентні технології*, в основі яких лежить *поняття “агента”*, яке останніми роками було адаптовано до багатьох областей як прикладного і системного програмування, так і до досліджень в областях штучного інтелекту та розподілених інтелектуалізованих систем. Причому в кожному конкретному випадку *поняттю “агента” надається децю різне значення*.

Означення 24.2. *Мультиагентні технології* – це принципово новий метод розв'язання задач керування та розподіленої взаємодії в мережах динамічних систем, коли на відміну від класичного способу пошуку деякого чітко визначеного (детермінованого) алгоритму реалізації найкращого розв'язання проблеми *розв'язок отримується автоматично в результаті взаємодії множини самостійних цілеспрямованих програмних модулів – так званих агентів*.

Твердження 24.1. *Мультиагентні технології охоплюють як технології створення та використання МАС, так і мультиагентне керування (МАК).*

І все ж таки, виникає питання: а що таке агент? Можна скористатись загальним формулюванням, наведеним у фундаментальній роботі [53]:

Твердження 24.2. *Агентом є все, що може розглядатися як сприймаюче своє середовище за допомогою давачів і діюче на це середовище за допомогою виконавчих механізмів.*

Проте ми далі будемо користуватись більш строгим визначенням агента:

Означення 24.3. *Агент/мультиагент* – це програмно-апаратний чи програмно-емуляційний автономний компонент мультиагентної системи, який функціонує за певним сценарієм/алгоритмом для досягнення поставлених користувачем або іншим агентом/мультиагентом перед МАС цілей.

Отже, будь-який агент уявляє собою відкриту систему, розміщену в деяке середовище, причому ця система володіє власною поведінкою, задовольняючою деяким екстремальним принципам.

Таким чином, агент вважається спроможним сприймати інформацію із зовнішнього середовища, опрацьовувати її на основі власних ресурсів, взаємодіяти з іншими агентами і впливати на середовище на протязі деякого часу, переслідуючи свої власні цілі.

Як приклад, у виробничій сфері діяльності людини можна зазначити, що більшість сучасних систем характеризуються відсутністю засобів ефективної ідентифікації нових потреб і можливостей в середовищі, які б дозволяли підприємству оперативно приймати раціональні рішення реконфігурації виробничих, кадрових, фінансових та інших ресурсів. Типовими прикладами подій, викликаючих необхідність заново ідентифікувати потреби і можливості, є наступні: поява нового вигідного замовлення, для виконання якого недостатньо власних ресурсів підприємства; вихід з ладу частини наявних ресурсів; зміна критеріїв прийняття рішень. Чим вища невизначеність, чим більш розподілений характер мають процеси прийняття рішень і чим частіше виникають незаплановані події, тим нижче ефективність існуючих систем, не здатних самостійно приймати рішення та автоматично переналагоджуватись під зміни у середовищі. Крім того, необхідність модифікації схеми прийняття рішень в традиційних системах видається складною та трудомісткою задачею, яка потребує високої кваліфікації виконавців. Це робить розробку та експлуатацію таких систем вкрай коштовними.

Саме для розв'язання подібних проблем останніми роками все частіше застосовуються мультиагентні технології, в основі яких лежить поняття “агента”, яке в теперішній час адаптоване до багатьох областей як прикладного і системного програмування, так і до досліджень в областях штучного інтелекту та розподілених інтелектуалізованих систем.

24.2. Особливості мультиагентного підходу

В основі мультиагентного підходу лежить поняття мобільного програмного агента, який реалізований і функціонує як самостійна спеціалізована комп'ютерна програма або елемент штучного інтелекту [190] (означення агента/мультиагента, алгоритму його дії, конкретної унікальної послідовності дій інтелектуальних агентів тощо для задачі створення на основі МАС автоматизованої системи вибору ШНС наводиться в *розділі 26*).

В теорії МАС за основу береться принцип колективної взаємодії, який полягає в наступному. Вважається, що один агент володіє лише частковим уявленням щодо глобальної проблеми, а відтак, він

може розв'язувати лише деяку частку загальної задачі. У зв'язку з цим для розв'язання складної задачі необхідно створити деяку множину агентів і організувати між ними ефективну взаємодію, що дозволить побудувати єдину МАС. В багатоагентних системах весь спектр задач по визначеним правилам розподіляється між всіма агентами, кожний з яких вважається "членом організації або групи". Розподілення завдань означає присвоєння кожному агенту деякої ролі, складність якої визначається, виходячи зі спроможностей та задач агента.

Для організації процесу розподілення задачі в МАС створюється або *система розподіленого розв'язання проблеми*, або *децентралізований штучний інтелект*. У першому варіанті процес декомпозиції глобальної задачі і зворотний процес композиції знайдених розв'язань відбувається під керуванням деякого єдиного «центра». При цьому МАС проектується строго зверху униз, виходячи з ролей, визначених для агентів, і результатів розбиття глобальної задачі на підзадачі. У випадку використання децентралізованого штучного інтелекту розподілення завдань відбувається в процесі взаємодії агентів і носить більш спонтанний характер. Нерідко це приводить до появи в МАС резонансних, синергетичних ефектів.

Отже, первісно, поки не існувало відповідних інформаційних технологій, "агент" був людиною, якій делегувалась частина повноважень – як у виконанні конкретних функцій, так і в прийнятті рішень. В перших (не комп'ютерних) МАС агенти уявляли собою працівників компаній, від імені і за дорученням яких вони взаємодіяли між собою при виконанні визначеної задачі. Такі *централізовані системи* успадковували деякі риси "бюрократичної" організації, в тому числі централізацію управління, статичну структуру та вузькоспеціалізовану агентну функціональність. Зокрема, базовий агент (резидент) отримував задачу, декомпозував її і розподіляв підзадачі між іншими агентами, після чого отримував результат і приймав рішення – при цьому, як правило, більшість агентів займалось виключно збиранням і постачанням інформації.

На заміну таким системам, копіюючим централізовану ієрархію, прийшли *розподілені системи*, в яких знання і ресурси розподілялися між достатньо "самостійними" агентами, проте зберігався загальний орган командного керування, приймаючий рішення в критичних або конфліктних ситуаціях.

Подальшим кроком в цьому напрямі стала парадигма повністю *децентралізованих систем*, в яких керування відбувається тільки

за рахунок локальних взаємодій між агентами. При цьому *вузька функціональна орієнтація* агента на розв'язання якоїсь однієї окремої частини “загальної” задачі поступово стала поступатися *універсальній цілісності* (автономності).

На рис. 24.1 наведені для порівняння дві схеми побудування програмного забезпечення: традиційна (рис. 24.1, а) і на основі МАС (рис. 24.1, б). В мультиагентній системі кожній сутності ставиться у відповідність *програмний агент*, який уявляє її інтереси.

Агенти дуже схожі на членів команди, які можуть змагатися один з одним або співпрацювати в процесі прийняття рішень. Це так званий *емерджентний інтелект*, основна особливість якого – динаміка і непередбачуваність процесу прийняття рішень. На практиці це означає, що рішення досягається за рахунок сотень або тисяч взаємодій, які майже неможливо відстежити. Проте це й не потрібно, оскільки агентам *задають мету*, якої вони повинні досягти, але *не зумовлюють сценарії виконання задач* щодо її досягнення. Ці сценарії формуються і виконуються агентами самостійно. На кожному кроці агенти переглядають входи системи і реагують на непередбачувані події (затримки, збої, зміни). Реакція може бути самостійною або здійснюватися у взаємодії з оператором. Таким чином, емерджентний інтелект – це не якийсь новий і спеціально сконструйований унікальний “блок”, який доповнює систему. Навпаки, це дещо (результат самоорганізації), що виникає спонтанно (за рахунок безлічі схованих або явних ситуативно складених умов) і в наперед не передбачуваний момент часу і так же випадково зникає, але в процесі свого існування визначаючим чином керує роботою всієї системи.

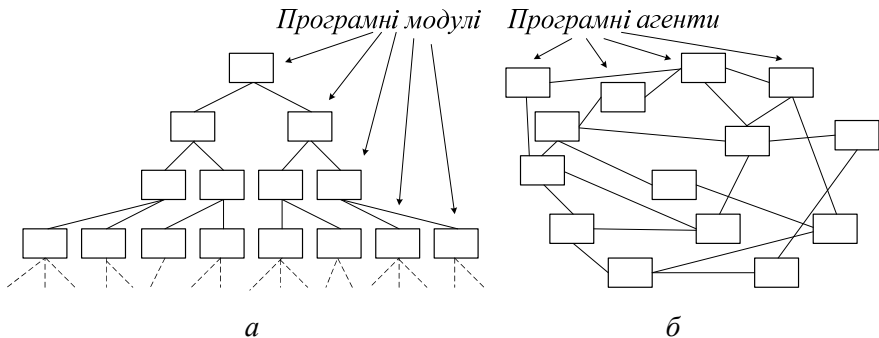


Рис. 24.1. Схеми побудови програмного забезпечення на основі: традиційних програмних модулів (а); програмних агентів МАС (б)

При цьому вважається, що інтелектуалізовані агенти мають можливості навчатись і перенавчатись. Серед спеціалістів з МАС як одна з перспективних моделей розглядається модель *самонавчаного агента*. Проте при цьому робляться посилання на результати в області витягання знань та машинного навчання, отримані раніше в штучному інтелекті стосовно до експертних систем. Очевидно, що по відношенню до МАС задача навчання має свою специфіку у порівнянні із задачами у загальній постановці, проте ця специфіка досі не вивчається і не проводяться дослідження щодо цієї проблеми. Вельми специфічна і задача навчання агентів колективній поведінці, адже кооперативне розв'язання задач має на меті сумісне використання знань декількох агентів. Це питання також поки ще залишається поза увагою спеціалістів з МАС.

24.3. Характеристика агентів мультиагентної системи

Відмінні ознаки агента. В загальному випадку агент сенсорно отримує вхідні впливи з *об'єкта сприйняття* – *навколишнього середовища* (яке може бути також агентом) та генерує дії на виході за посередництвом *виконавчих механізмів*, впливаючих на середовище.

Отже, відмінними ознаками агента можна вважати наступне:

1. Агент здатний сприймати своє *навколишнє середовище* (і чим складніша ця модель, тим більше інформації агент може використовувати в своїй діяльності).
2. У агента є поставлені користувачем або іншим агентом (або групою агентів) перед ним *цілі* – завдання, які необхідно вирішити, або результати, яких треба досягти (цілі можуть бути задані ззовні агента або породжуватися в результаті якоїсь його "розумової діяльності").
3. Для досягнення мети агент може здійснювати в середовищі певні дії (агент, звичайно, може намагатися здійснювати дії навмання. Але це неефективно. Тому агент *повинен володіти знаннями про набір своїх можливих дій, необхідні умови для їх здійснення і можливі наслідки*.)
4. Агентові дається тільки знання *про навколишній світ* і його можливості.
5. *Алгоритм дії* агента, або конкретна *унікальна послідовність його дій*, що приводить до мети, шукається агентом кожного разу з урахуванням поточних внутрішніх і зовнішніх умов і може мінятися і коректуватися по ходу виконання.

Властивості агентів. Можна виділити наступні властивості агентів, які дозволяють останнім виконувати покладені на них користувачем або іншими агентами задачі:

1. *Автономність*: агенти функціонують без прямого втручання людини чи кого-небудь іншого та наділені певною властивістю контролювати свої дії та внутрішній стан.
2. *Реактивність*: агенти можуть сприймати навколишнє середовище (тобто фізичний світ, користувача, який взаємодіє через графічний інтерфейс, колектив інших агентів тощо) та адекватно реагувати в певних часових рамках на зміни, які відбуваються.
3. *Комунікабельність і кооперативність*: агенти можуть обмінюватись інформацією з оточуючим їх середовищем та іншими агентами. Можливість комунікацій означає, що агент повинен отримувати інформацію про оточуюче середовище, що дає йому можливість будувати власну модель світу. Саме можливість комунікацій з іншими агентами є обов'язковою умовою сумісних дій для досягнення цілей.
4. *Володіння методами (способами) спілкування*: агенти взаємодіють з іншими агентами за допомогою комунікаційної мови.
5. *Активність*: агенти не просто реагують на зміни навколишнього середовища, а наділені цілеспрямованою поведінкою та властивістю проявляти ініціативу.
6. *Інтелектуальна поведінка*: поведінка агента має здібності до навчання, логічної дедукції чи конструювання моделі оточуючого середовища для того, щоб знаходити оптимальні способи поведінки.
7. *Індивідуальна картина світу*: кожний агент має власну модель оточуючого його світу_(середовища), яка описує те, як агент бачить світ. Агент будує свою модель світу на основі інформації, яку отримує з навколишнього середовища.

Формалізоване уявлення про агента. Спочатку дамо формалізоване уявлення про сприйняття агентом зовнішнього середовища – ця взаємодія ще дістала назву *зовнішньої архітектури агента*. Скористуємось для цього графічним зображенням взаємодії агента з оточуючим середовищем (рис. 24.2) і сформуємо її формальний опис у вигляді наборів:

$$\begin{aligned} C &= \{c_1, c_2, \dots, c_N\} \text{ стани навколишнього середовища;} \\ O_c &= \{o_{c1}, o_{c2}, \dots, o_{cK}\} \text{ – об'єкти в уявленні агента;} \\ D &= \{d_1, d_2, \dots, d_M\} \text{ – дії агента щодо виконання мети,} \end{aligned} \quad (24.1)$$

де N , K , M – відповідно, розмірності кількостей станів середовища,

його об'єктів спостереження та дій агента, які, в свою чергу, можна подати певними перетвореннями:

Агент (його реакція як результат сприйняття середовища):

$O_c \rightarrow D$;

Спостереження (стану середовища від дій агента): (24.2)

$S \times D \rightarrow O_c(C)$;

Сприйняття (його уявлення щодо змін в середовищі):

$C \rightarrow O_c$;

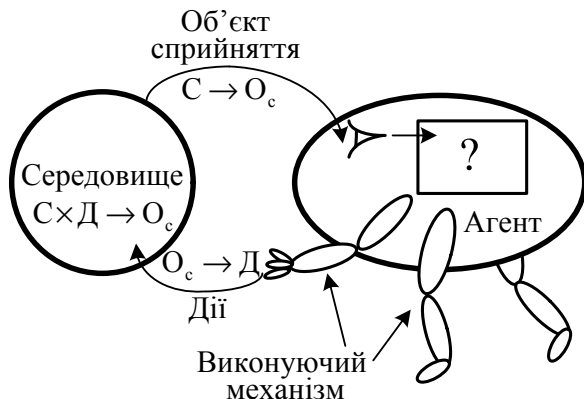


Рис. 24.2. Інтерпретація агентної взаємодії з навколишнім середовищем

Власне дії агента підпорядковуються його архітектурі реакції, яка характеризується наступними властивостями:

- відкидається використання логічного представлення;
- немає обмірковування або тлумачення;
- вхідні сприйняття відразу перетворюються в дії;
- необхідно розрізняти пріоритет між поведінками.

Тоді дія агента регламентується наступною послідовністю:

fired = all(O_c, D) with current O_c

for each *fired*

if (C, D) has max priority

return *D*

return *null*.

Графічна інтерпретація взаємодії агента зі станами з навколишнім середовищем наведена на рис. 24.3.

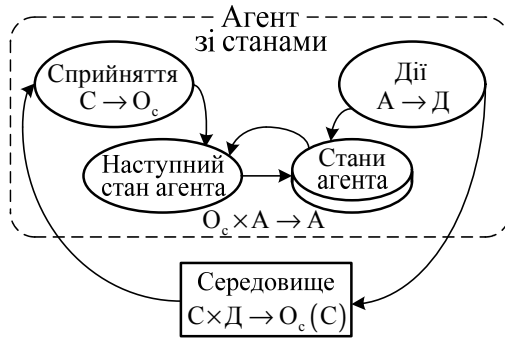


Рис. 24.3. Графічна інтерпретація агента зі станами

Формальний опис такого агента характеризується взаємовідносинами з оточуючим середовищем, поданими набором:

$$\begin{aligned}
 C &= \{c_1, c_2, \dots, c_N\} \text{ – стани навколишнього середовища;} \\
 O_c &= \{o_{c1}, o_{c2}, \dots, o_{cK}\} \text{ – об'єкти в уявленні агента;} \\
 D &= \{d_1, d_2, \dots, d_M\} \text{ – дії агента щодо виконання мети;} \\
 A &= \{a_1, a_2, \dots, a_L\},
 \end{aligned}
 \tag{24.3}$$

де L – розмірність кількостей станів агента.

Аналогічно наведеному у (24.2) запишемо через відповідні перетворення формалізоване подання взаємодії агента зі станами з навколишнім середовищем:

Агент (його реакція як результат сприйняття середовища): $A \rightarrow D$;

Спостереження (стани середовища від дій агента): $C \times D \rightarrow O_c(C)$;

Сприйняття (його уявлення щодо змін в середовищі): $C \rightarrow O_c$;

Наступний стан агента: $O_c \times A \rightarrow A$

Для агента зі станами архітектура логіки визначається наступними особливостями:

- використанням символного представлення (з традиційного штучного інтелекту);
- обмірковуванням, реалізованим через логічну дедукцію;
- набором дедуктивних правил R .

Тоді дія агента зі станами регламентується наступною послідовністю логічної дедукції:

for each possible D

if R applied on *and* prove do(D)

return D

```

for each possible Д
    if R applied on and do not prove not do(Д)
        return Д
return null.
    
```

Класифікація агентів. Фактично, використовуючи поняття “агент”, кожний автор або співтовариство наділяють свого агента конкретним набором властивостей в залежності від цілей розробки, розв’язуваних задач, техніки реалізації, критеріїв. Як наслідок, в межах даного напрямку з’явилася безліч типів агентів, наприклад: автономні, інтелектуальні, соціальні агенти, персональні асистенти тощо. А в залежності від ступеня можливості внутрішнього уявлення зовнішнього світу і способу поведінки агенти класифікуються як локальні, мережеві, мобільні, інтерфейсні, транслюючі, маршрутизаційні тощо.

Класифікація агентів з сукупністю класифікаційних ознак наведена на рис. 24.4, причому властивості агента/об’єкта описуються *початковою системою*, а правила поведінки – *породжуючою системою*. Стан агента/об’єкта визначається переліком його властивостей з поточними значеннями.

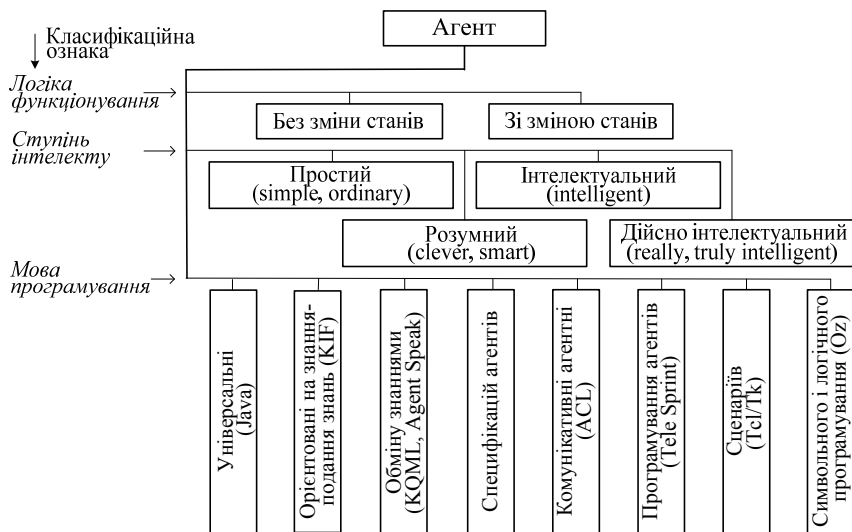


Рис. 24.4. Класифікація агентів МАС

В таблиці 24.1 [66; 67] наведений приклад характеристики властивостей агентів стосовно їх класифікації, поданої на рис. 24.4.

Характеристики властивостей типів агентів

Характеристики	Типи агентів			
	Прості	Розумні (smart)	Інтелектуальні (intelligent)	Дійсно інтелектуальні (truly intelligent)
Автономне виконання	+		+	+
Взаємодія з іншими агентами і/або користувачами	+	+	+	+
Спостереження за оточенням	+	+	+	+
Здібність використання абстракцій		+	+	+
Здібність використання предметних знань		+	+	
Можливість адаптивної поведінки у досягненні цілей			+	+
Навчання з оточення			+	+
Толерантність до похибок і/або невірних вхідних даних			+	
Real-time виконання			+	

За логікою функціонування або архітектурою логіки розподіл агентів розглянуто вище і упорядковується виразами (24.1) – (24.3). Так звані *агенти без зміни станів* (або ще – *реактивні агенти*) не мають ані скільки-небудь розвинутого уявлення щодо зовнішнього середовища, ані механізму багатокрокових міркувань, ані достатньої

кількості власних ресурсів. В силу цих недоліків реактивні (вони ж і *прості*) агенти володіють дуже обмеженим діапазоном передбачення і практично не здатні планувати свої дії, оскільки реактивність у чистому виді означає таку структуру зворотного зв'язку, яка не містить механізмів прогнозування. Як це впливає з самої назви, реактивні агенти функціонують в основному на рівні стимульно-реактивних зв'язків, володіючи дуже "бідною" індивідуальністю і сильною залежністю від зовнішнього середовища (спільноти агентів).

Агентам зі зміною станів притаманні в тій чи іншій мірі розумові (інтелектуальні) здатності. Такі агенти володіють добре розвиненою і поповнюваною символічною моделлю зовнішнього світу, що досягається завдяки наявності у них бази знань, механізмів розв'язання задач, аналізу дій і прийняття рішень. Близький термін "розсудливий (smart)" саме і зарезервований для позначення агента, який на основі символічної моделі зовнішнього середовища здатний проводити власні міркування, наприклад, використовуючи метод порівняння за зразком, і на їх основі приймати самостійні рішення або виконувати дії, впливаючи на середовище.

Інтелектуальні агенти значно автономніші за реактивні, мають яскравіше виражену індивідуальність і характеризуються доцільною поведінкою у співтоваристві агентів, а також намаганням використати ресурси інших агентів для досягнення власних цілей.

Означення 24.4. Під *інтелектуальним агентом* в інформатиці та штучному інтелекті розуміють будь-які фізичні або віртуальні одиниці, здібні, принаймні, підтримувати взаємодію з оточуючим середовищем, отримуючи від нього інформацію, і, реагуючи на неї своїми діями, виявляти власну ініціативу, посилати та отримувати повідомлення від інших агентів і вступати з ними у взаємодію, діяти без втручання ззовні, в тому числі і без втручання людини.

Отже, щодо *ступеня інтелектуальності* агенти або не можуть продукувати рішення і в процесі функціонування виконують директиви, які надходять від інших агентів, не маючи власного логічного уявлення і необхідних знань з розв'язуваної проблеми (так звані *прості агенти*), або наділені всіма арсеналами знань і спроможностей щодо самостійного прийняття рішень при виникненні будь-яких ситуацій в оточуваному середовищі (*дійсно інтелектуальні агенти*). Рівні обмеженості у прийнятті рішень притаманні розумним та інтелектуальним агентам.

Як для створення, так і для функціонування використовується широкий спектр *мов програмування* – від універсальних (Java) до спеціалізованих (Tale Sprint), а також орієнтованих на подання знань

(KIF – Knowledge Interchange Format (syntax)) і обмін знаннями (KQML – Knowledge Query and Manipulation Language, AgentSpeak, April) тощо, які як підтримують автономність функціонування агента в МАС, так і дозволяють створювачам останніх розробляти і “закладати” в систему всілякі сценарії поведінки агентів за змінюваних умов, які, яке правило, у складних ситуаціях неможливо передбачати заздалегідь.

Велику роль відіграють в МАС також комунікативні агентні мови (ACL – Agent Communication Language), які спираються на мовну взаємодію, тобто мовні дії при програмуванні відтворюються через стандартні ключові слова, що забезпечує одноманітність подання інформації, яка передається агентами.

Як впливає з наведеної таблиці 24.1, власне доцільна поведінка з’являється тільки на рівні інтелектуальних агентів. Для таких агентів необхідна не тільки наявність цілей функціонування, але й можливість використання досить складних знань про середовище, партнерів і про себе.

24.4. Особливості використання МАС

Ефективне використання МАС узагальнюються чотирма основними напрямками:

- розв’язанням однієї задачі декількома інтелектуальними підсистемами;
- розбиттям задачі на декілька підзадач, які розподіляються між агентами;
- забезпеченням взаємодії між агентами;
- створенням запитів одного агента до іншого агента на передачу деяких даних або на виконання певних дій.

Фактори використання МАС. Фактори використання МАС є різноманітними і залежать в значній мірі від сфери застосування і прикладної задачі. Далі наводяться лише ті фактори, які є інваріантними до застосовного аспекту використання МАС:

- деякі предметні області використовують МАС в тих випадках, *коли логічно кожного з учасників процесу представити у вигляді агента*. Наприклад: соціальні процеси, в яких кожний з учасників відіграє свою роль; виробничі процеси, де кожний з обробних ресурсів/агентів виконує визначену (в тому числі, й унікальну) операцію тощо;
- *паралельне виконання задач*, якщо предметне середовище легко представити в вигляді сукупності агентів, то незалежні

задачі можуть виконуватись різними агентами. Як приклад з виробничої сфери – виготовлення комплектуючих складальної одиниці сукупністю відповідних обробних ресурсів/агентів;

- *стійкість роботи системи*: коли контроль та відповідальність за дії, які виконуються, можна розподілити між декількома агентами. При відмові одного агента *система не перестає функціонувати*, бо функції агента, який вийшов з ладу, бере на себе інший агент. Як приклад технічної системи – резервування устаткування на випадок відмов і виходу з ладу основного обробного ресурсу/агента;
- *модульність МАС*, коли у випадках нестачі потужностей системи для розв'язання нової задачі або формуванні нових цілей при випадкових змінах ситуацій в оточуючому середовищі *необхідно нарощувати та видозмінювати систему*, тобто легше додати агента, чим змінити властивості єдиної програми.

Класифікація МАС. Основою для класифікації МАС слугують функції агентів в таких системах у розв'язанні локальних і глобальних задач/цілей:

1. *Кооперативні*: агенти *являються частинами єдиної системи* і вирішують підзадачі однієї загальної задачі, тому агент не може працювати поза системою та виконувати самостійні задачі;
2. *Конкуруючі*: агенти *є самостійними системами*, хоча для досягнення цілей вони можуть поєднувати свої зусилля, приймати цілі та команди від інших агентів, проте підтримка зв'язку з іншими агентами не обов'язкова;
3. *Змішані*: поєднання першого та другого варіантів.

Навчаючі аспекти в МАС. В основі навчання агентів МАС лежать викладені в розд. 4 методи, а також специфічні особливості індивідуального і суспільного навчання:

- *навчання під наглядом*: навчання *контролюється зовнішнім джерелом* (вчителем), агенту надається точна інформація про вхідні-вихідні пари даних;
- *навчання без нагляду*: навчання не контролюється зовнішньо, а *здійснюється агентом самостійно* (без вчителя, шляхом випробувань та помилок);

- *навчання з підкріпленням*: агент контролює своє власне навчання, обираючи дію для спроби та оцінюючи результати;
- *незалежне навчання*: агент навчається незалежно, без обміну інформації з іншими агентами;
- *суспільне навчання*: агент ділиться сприйняттям, розумінням середовища, відповідними лініями поведінки або розрахунками/винагородами з іншими агентами.

24.5. Мультиагентні робототехнічні системи

Принцип дії мультиагентних робототехнічних систем.

Розглянемо особливості МАС на прикладі робототехнічних систем як сучасної складної системи, в якій значною мірою проявляються проблеми штучного інтелекту, розподіленості та колективної взаємодії великої кількості обробних ресурсів. Зазвичай на практиці роботи інтегруються в *мультиагентні робототехнічні системи* (МАРТС) для сумісного досягнення загальних цілей і розв'язання складних задач. При цьому виникають нові проблеми групового керування і комунікації, пов'язані з організацією “колективної” поведінки роботів. Традиційно ці проблеми вирішувались на основі централізованого або децентралізованого керування.

Компромісний підхід полягає у мережевому керуванні та груповій навігації МАРТС. В цьому випадку роботи розглядаються як інтелектуалізовані мехатронні агенти МАРС з локальними БД і БЗ, які можуть оперативно обмінюватися інформацією.

Принцип дії МАРТС базується на декомпозиції загальної задачі на ряд локальних задач, доручуваних на агентів-роботів, розподіленні цих задач між ними, плануванні колективної поведінки агентів, координації взаємодії агентів на основі кооперації, реконфігурації, комунікації та розв'язування конфліктних ситуацій.

Задачі *стратегічного рівня* зазвичай покладаються на спеціального агента-координатора, а задачі *тактичного рівня* паралельно розв'язуються роботами як мехатронними агентами. В результаті мультиагентного керування та групової навігації значно підвищуються надійність, адаптивність і швидкодія МАРТС в змінюваному середовищі з перешкодами. В останні роки були розроблені основи теорії керування МАРТС та інформаційна технологія мультиагентної безконфліктної навігації “колективу” роботів, функціонуючих в змінюваному середовищі з перешкодами [54; 55; 56]. Зокрема, запропоновано методи групової навігації та керування колективним рухом роботів-агентів і розв'язання конфліктів (запобігання зіткнень).

Мультиагентне керування інформаційними потоками в МАРТС. Удосконалення МАРТС пов'язане з розвитком методології автоматизації, адаптації та інтелектуалізації систем мережевого управління інформаційними потоками на базі динамічних моделей *телекомунікаційних мереж* (ТКМ) складних об'єктів керування зі змінною структурою, методів оптимізації процесів маршрутизації інформаційних потоків і принципів адаптивного та інтелектуального керування трафіком з використанням мультиагентних технологій та протоколів нового покоління (IPv6 та ін.). На цьому новому шляху можливе як урахування реальної динаміки ТКМ, тобто фактичного стану або зміни структури і топології та параметрів (ваг каналів зв'язку) ТКМ в реальному часі, так і адаптація до факторів невизначеності на основі моніторингу і функціональної діагностики ТКМ [55].

Основні функції опрацювання інформації, самоорганізації та керування інформаційними потоками по запитам зовнішніх агентів розподіляються між внутрішніми агентами, роль яких виконують мережеві або *нейросіткові агенти*. Архітектура цих внутрішніх мережевих агентів аналогічна архітектурі ТКМ.

Нейросіткові агенти призначені перш за все для паралельної передачі та опрацювання складних мультимедійних сигналів і образів (2D- або 3D-зображень тощо). В результаті навчання за множиною прецедентів з навчаючої БД здійснюється налагодження архітектури (її топології) та параметрів (синаптичних ваг) нейронних агентів на розв'язувану задачу [199].

24.6. Проблеми при створенні мультиагентних систем

Одна з проблем, які супроводжують створення МАС *на принципах штучного інтелекту*, полягає в тому, що при проектуванні точної та повної моделі представлення світу, процесів і механізмів розмірковування в ньому *дуже важко створити адекватну і повну картину світу з великою кількістю параметрів*.

Всі інші проблеми пов'язані з тим, що технологія МАС хоча і нараховує вже понад десятирічну історію свого активного розвитку, ще й досі знаходиться в стадії становлення. Ведуться активні дослідження в області теоретичних основ формалізації основних понять і компонент систем, в особливості в області формалізації ментальних понять. Основні досягнення в цій сфері поки не дуже орієнтовані на аспекти практичної реалізації і далекі від практики. Зокрема, при

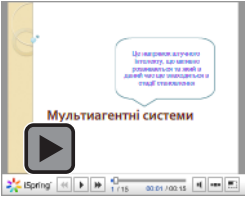
формалізації ментальних понять повністю ігноруються всі розроблені в штучному інтелекті підходи для роботи з погано структурованими поняттями, не досить визначеними поняттями, методи, які спираються на ймовірність і нечіткість. І це, безумовно, чимале, нове і чисте поле діяльності для відповідних спеціалістів.

Контрольні завдання

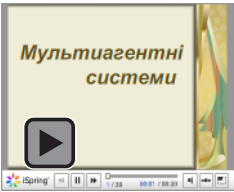
1. Розкрити визначення мультиагентної системи як нової парадигми інформаційної технології.
2. Обґрунтувати штучного інтелекту і розподіленого розв'язання задач в мультиагентних системах.
3. Прокоментувати мультиагентні технології як принципово новий метод розв'язання задач керування при розподіленні взаємодії їх складових в мережах динамічних систем.
4. Дати характеристику і розкрити властивості агента/мультиагента як автономного компоненту мультиагентної системи.
5. Охарактеризувати особливості колективної взаємодії агентів при розв'язанні задач великої розмірності мультиагентними системами.
6. Навести відмінні ознаки і розтлумачити властивості агентів в мультиагентних системах.
7. Дати формалізоване уявлення про агента і його взаємодію із зовнішнім середовищем.
8. Навести класифікаційні ознаки і прокоментувати класифікацію агентів за основними ознаками.
9. Розкрити поняття інтелектуального агента з позицій ступеня його інтелектуальності.
10. Розкрити особливості ефективного використання мультиагентних систем.
11. Проаналізувати навчальні аспекти мультиагентних систем.
12. Охарактеризувати принципи дії мультиагентних робототехнічних систем.
13. Розкрити проблеми, якими супроводжується створення мультиагентних систем.

Презентації

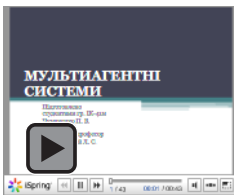
1. Мультиагентні системи



2. Мультиагентні системи



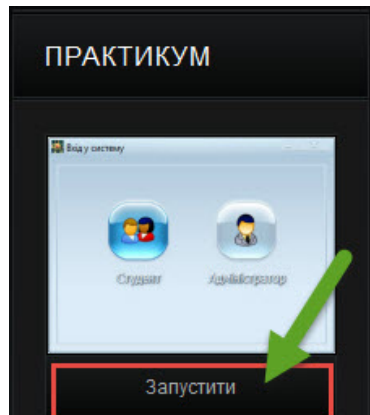
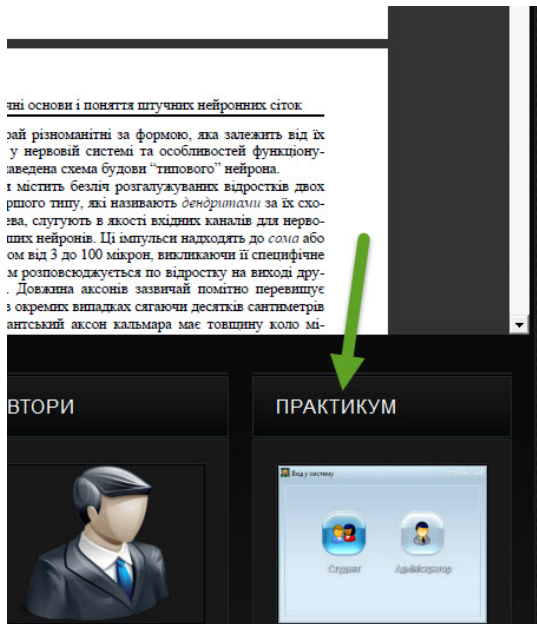
3. Мультиагентні системи



Практичні заняття

Мультиагентні системи

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



Концептуальна модель вибору штучних нейронних сіток

*Те, що ми знаємо, – обмежене,
а чого не знаємо – нескінчене*
Апулей з Мадаври

*І аж тоді підуть по цій дорозі люди,
як ми проб'єм її та вирівняєм всюди...*
Іван Франко

25.1. Вимоги щодо компетенції користувача

Вибір топології та настроювання ваг зв'язків штучної нейронної сітки є одними з важливіших етапів при використанні нейро-сіткових технологій для розв'язання практичних задач. Від цих етапів прямо залежить якість (адекватність) отриманої нейросіткової моделі, системи керування тощо. Побудування ШНС за традиційною методикою виконується, фактично, методом спроб і похибок. Досліджувач вибирає базову модель нейронів, задає кількість шарів (архітектуру), а також структуру зв'язків між ними (наявність/відсутність рекурентних зв'язків), а далі дивиться, що ж у нього створилося; врешті сітка навчається за допомогою якого-небудь методу, а далі тестується на тестовій вибірці. Якщо отримані результати роботи задовольняють заданим критеріям, то задача створення ШНС вважається виконаною успішно; в противному випадку – процес повторюється з іншими значеннями вихідних параметрів. Природно, що такий підхід до побудування ШНС як інструментарію дослідження складних об'єктів не може вважатися задовільним, що й зумовило пошуки інших підходів до розв'язання задачі пошуку оптимальних топологій і методів навчання, пристосованих до певних властивостей/вимог прикладних задач.

В даному випадку, на відміну від кінцевого користувача готового нейросіткового “розв'язувача”, під користувачем розуміється людина, яка безпосередньо створює нейросіткові моделі. Не дивлячись на численні заяви про те, що нейронні сітки доступні користувачам-неспеціалістам, насправді все виглядає інакше. *По-перше*, правильно формулювати задачу ніякий комп'ютер за користувача не в змозі (мається на увазі не тільки вірність змістовної постановки задачі, але й коректний вибір математичного методу розв'язання та його настроювань – а це у випадку нейронних сіток припускає вибір адек-

ватої структури НС, алгоритму навчання, критерія якості розв'язання задачі тощо). Звичайно, значна гнучкість і універсальність нейроалгоритмів припускають застосування наближених топологій НС, проте не завжди це приводить до найкращого результату. *По-друге*, існуючі автоматичні схеми підбирання оптимальних настроювань нейросіткових методів не можуть відшукати вірне розв'язання для більш-менш складних задач. Іншими словами, для багатьох класів задач прикладне застосування нейротехнологій вимагає професійних знань як методів нейроінформатики, так і інших методів аналізу даних (статистики, наприклад).

З іншого боку, користувач готового нейросіткового продукту стикається з труднощами “навпаки”. Він може правильно сформулювати задачу, проте набагато гірше усвідомлює принципи вибору структури НС, підготовки даних і алгоритму її навчання, перевірки якості функціонування НС і вербалізації тощо. І саме це спонукає і фахівців в області нейротехнологій, і користувачів-неспеціалістів в цій області інтенсифікувати дослідження для створення умов щодо автоматизації процесу поетапного синтезу штучних нейронних сіток.

25.2. Логічна модель поетапного синтезу штучних нейронних сіток

В *розділах 3.4-3.8* були окреслені загальні зауваження щодо особливостей традиційного підходу до вибору ШНС під конкретну розв'язувану задачу.

Як вже зазначалося у *розділі 2*, покращення ситуації з розв'язанням проблеми усунення суб'єктивного фактору при виборі топології ШНС, яка в найбільшій мірі задовольняє умови модельованої прикладної задачі, можна досягти при виконанні наступних дій:

- формуванні набору *вирішних класифікаційних ознак* (НВКО) і створенні класифікатора ШНС (ця процедура розглянута у *розділі 2*);
- побудуванні чіткої *логічної моделі поетапного синтезу* (ЛМПС) ШНС;
- створенні строгої *узагальненої моделі вибору типових топологій* (УМВТТ) ШНС для конкретних прикладних задач, що базується на створенні *гнучких інтелектуалізованих мультиагентних конфігурацій* (ГІМАК) *агентно-орієнтованих підсистем* (АОП) для конкретних прикладних задач.

Наведена на *рис. 25.1* схема містить відображення послідовності урахування НВКО при ітераційній процедурі синтезу ШНС на основі *конкретної унікальної послідовності дій* (КУПД) інтелектуальних агентів в процесі функціонування ГІМАК агентно-орієнтованих підсистем.

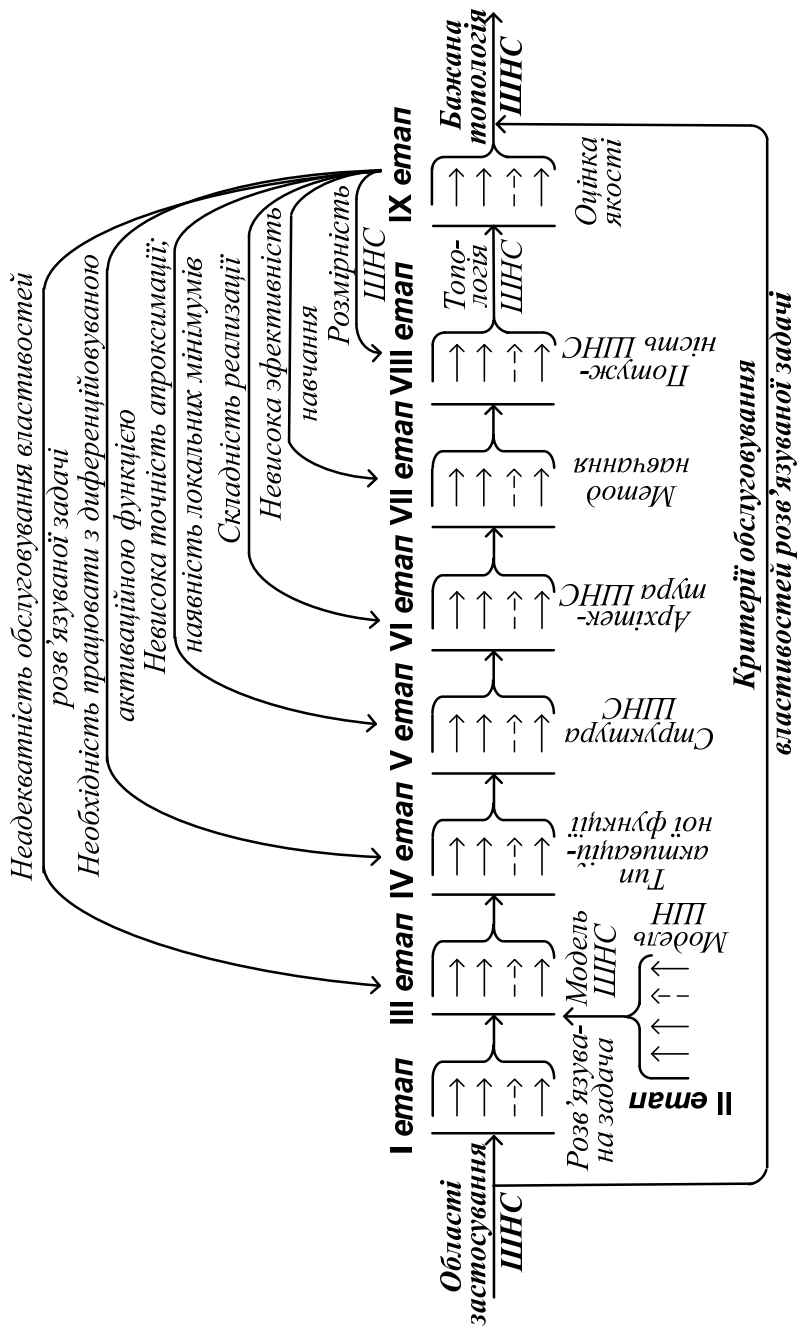


Рис. 25.1. Логічна схема поетапного синтезу ШНС

Переозначимо наведене у *розд. 24.1* *означення 24.3* у форму, яка більш точно характеризує розглядувану проблематику:

Означення 25.1. *Агент/мультіагент* – це програмно-апаратний чи програмно-емуляційний автономний компонент мультіагентної підсистеми, який функціонує за певним сценарієм/алгоритмом на основі КУПД в інтересах досягнення поставлених користувачем перед нею цілей.

Твердження 25.1. Алгоритм дії агента може мінятися і коректуватися по ходу виконання завдання заради досягнення мети.

Твердження 25.2. Конкретна унікальна послідовність дій, що приводить до мети, шукається агентом кожного разу з урахуванням відповідних *критеріїв обслуговування* властивостей даної розв’язуваної задачі.

Означення 25.2. *Критерій обслуговуваності* – показник задоволення топологією ШНС вимог з боку розв’язуваної задачі і професійної відповідності користувача.

Означення 25.3. *Логічна модель поетапного синтезу ШНС* – така послідовність їх перебирання в просторі НВКО, яка, будучи виконувана користувачем і/або ГІМАК АОП, відтворює принципи агентно-орієнтованого підходу та автономно дозволяє виокремити топологію/топології ШНС, здатну/здатні задовольнити критерії обслуговування властивостей задачі.

Передостання процедура поетапного синтезу ШНС полягає у перевірці виконання умов *достатності і ненадлишковості архітектури* синтезованої НС, яка в процесі навчання чи після його завершення або нарощується за алгоритмами Мезарда-Надаля, Мерчанда, Лі-Туфтса та каскадної кореляції (див. *розд. 17*), або спрощується шляхом видалення найменш значущих ваг і нейронів (див. *розд. 23*).

Заклучна процедура поетапного синтезу ШНС зводиться до перевірки реляційним перебиранням інтелектуальними агентами умов виконання вимог критеріїв прикладної задачі поточним вектором можливостей конкретної топології ШНС.

25.3. Узагальнена концептуальна модель процесу вибору топології штучних нейронних сіток

Трактування поняття вибору топології ШНС. Використання логічної послідовності синтезу ШНС під розв’язання (моделювання) конкретної прикладної задачі (ПЗ), який полягає у доборі базового нейрона (БН), виду активаційної функції (АФ), архітектури $A_{\text{ШНС}}$, структури $C_{\text{ШНС}}$, методу навчання ($MH_{\text{ШНС}}$) і методу спрощен-

ня ($МС_{ШНС}$) штучних нейросіток, є обов'язковою процедурою зі складовими формування топології ШНС. Інакше кажучи, склад і послідовність етапів синтезу повинні забезпечити весь набір ітераційних процедур, які й визначатимуть вибір такої топології ШНС, яка здатна адекватно відбивати властивості та обмеження модельовуваної (ВОЗ).

З цією метою будуватиметься так звана *віртуальна топологічна модель ШНС*, яка узагальнює в собі всі характеристики останньої, спроможні обслуговувати кожні з ВОЗ. За основу при цьому береться запропонований в роботі [72] підхід щодо синтезу/аналізу будь-якої складної організаційно-технічної системи, за яким на узагальненому **верхньому рівні** абстрагування концептуально *функція топології ШНС* ($\Phi_{ШНС}$ -функція) (як і будь-якої організаційно-технічної системи) як *об'єкта керування* (ОК) подається декартовим добутком множин БН, АФ, $A_{ШНС}$, $C_{ШНС}$, $MH_{ШНС}$, $MC_{ШНС}$:

$$\Phi_{ШНС}(ВОЗ) \subset БН \times АФ \times A_{ШНС} \times C_{ШНС} \times MH_{ШНС} \times MC_{ШНС}. \quad (25.1)$$

Таким чином, верхній рівень абстрагування визначає перебирання всіх варіантів складових топології ШНС (див. *означення 2.14 розд. 2.2*).

Як основа формалізації при описі функціональної бази ОК – топології ШНС та її підфункціональних компонентів – *елементарних складових* (ЕлС) на виділених ієрархічних рівнях використовуються скінчені графи, що відображають через НВКО властивості та обмеження задачі з ОК і його складовими ЕлО, а також *метод Ф-функцій* [73, 78], за якими встановлюється відповідність множин у виразі (25.1). Тоді метою етапів системного синтезу/аналізу топології ШНС є визначення складу та закономірностей взаємозалежності окремих компонентів в єдиній системі – ОК з урахуванням того, що *функція ОК визначає його адекватність модельовуваній задачі*.

При виборі та дослідженні ШНС вирізняють дві постановки задачі:

- *пряму* (задача аналізу) – відомі складові топології ШНС. Треба визначити ВОЗ – властивості та обмеження адекватно модельовуваної задачі (класу задач);
- *обернену* (задача синтезу) – відомі ВОЗ (властивості та обмеження прикладної задачі або класу задач). Треба визначити складові БН, АФ, $A_{ШНС}$, $C_{ШНС}$, $MH_{ШНС}$, $MC_{ШНС}$ топології ШНС, адекватно реалізуючої ВОЗ прикладної задачі.

Пряма задача виникає в процесі перевірки можливості використання певної топології ШНС для обслуговування властивостей конкретної задачі (класу задач), а обернена – при створенні нової або

суттєвій модифікації існуючої топології ШНС під певні ВОЗ.

В обох випадках узагальненою концептуальною моделлю може слугувати схема, наведена на рис. 25.2.

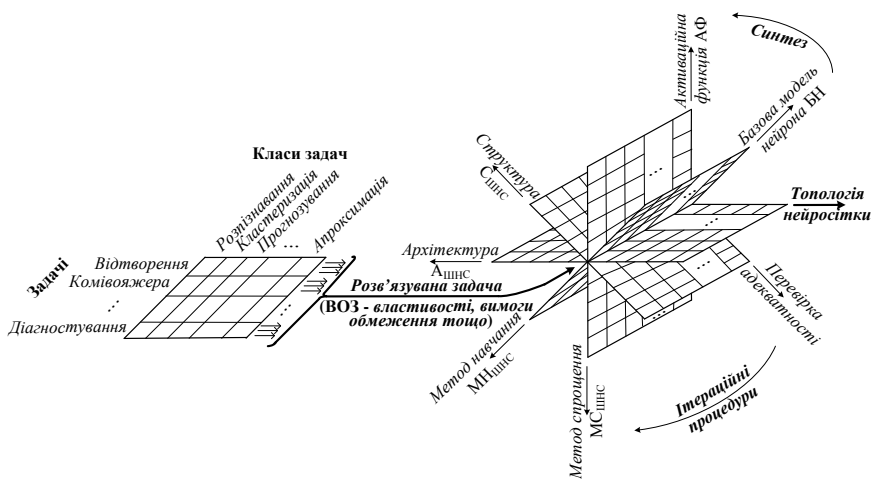


Рис. 25.2. Концептуальна модель процесу вибору топології ШНС

Отже, якщо загальна $\Phi_{\text{ШНС}}$ -функція як ОК може бути подана на вищому ієрархічному рівні абстрагування виразом (25.1), то кількість сполучень підфункцій складових процесів ($\Phi_{\text{склад-процеси}}$) перебирання на **другому рівні абстрагування** – $\Phi_{\text{БН}}$, $\Phi_{\text{АФ}}$, $\Phi_{\text{Ашнс}}$, $\Phi_{\text{Сшнс}}$, $\Phi_{\text{МНшнс}}$, $\Phi_{\text{МСшнс}}$, призначених для варіативного формування траєкторій tr руху в межах змінних кожної з наведених складових до кінцевої мети – оптимального шляху $tr_{\text{опт}}$, складатиме понад 3000 варіантів. І це тільки з урахуванням обмежено наведеної у відповідних розділах даної книги можливої різноманітності складових топології ШНС. Іншими словами, варіативний перетин всіх підфункцій в процесі реалізації ЛСПС ШНС уявляє собою багатоваріантну ієрархічну ітераційно-пошукову оптимізаційну задачу щодо послідовного перебирання складових топології ШНС, задовольняючих ВОЗ:

$$\begin{aligned}
 \Phi_{\text{БН}} &\subset \Phi \times \text{Ашнс} \times \text{Сшнс} \times \text{МНшнс} \times \text{МСшнс}; \\
 \Phi_{\text{АФ}} &\subset \text{БН} \times \text{Ашнс} \times \text{Сшнс} \times \text{МНшнс} \times \text{МСшнс}; \\
 \Phi_{\text{Ашнс}} &\subset \text{БН} \times \text{АФ} \times \text{Сшнс} \times \text{МНшнс} \times \text{МСшнс}; \\
 \Phi_{\text{Сшнс}} &\subset \text{БН} \times \text{АФ} \times \text{Ашнс} \times \text{Сшнс} \times \text{МНшнс}; \\
 \Phi_{\text{МНшнс}} &\subset \text{БН} \times \text{АФ} \times \text{Ашнс} \times \text{Сшнс} \times \text{МСшнс}; \\
 \Phi_{\text{МСшнс}} &\subset \text{БН} \times \text{АФ} \times \text{Ашнс} \times \text{Сшнс} \times \text{МНшнс}.
 \end{aligned}
 \tag{25.2}$$

Означення 25.4. Траєкторія tr руху – це слід у послідовності етапів синтезу топології ШНС, визначуваної перетином складових

топології ШНС з певними до ВОЗ показниками відповідності.

Таким чином, вирази (25.2) відображають собою проекції функціональних залежностей, що описують $\Phi_{\text{склад-процеси}}$, на відповідні координатні гіперплощини. Зокрема, серед виразів (25.2) є подання всіх складових, утворюючих топології ШНС і формуючих набір вирішних класифікаційних ознак (див. розд. 2.1) останніх.

Як і на наступних рівнях абстрагування, задача перебирання може, зокрема, виконуватися шляхом послідовного порівняння попарних сполучень фіксованих та змінних складових топології ШНС з відбиранням найкращих пар з умов задоволення ВОЗ. Процес добору оптимальних пар нагадує процедуру еволюційного добору генетичними алгоритмами (див. розд. 15) або алгоритмами МГУА (див. розд. 16). Відповідно, і реалізація процесу пошуку бажаної топології ШНС може виконуватися тими ж засобами.

Означення 25.5. *Траєкторія $tr_{\text{опт}} \text{ руху}$* – у послідовності етапів синтезу топології ШНС, визначуваної перетином складових топології ШНС з максимальними до ВОЗ показниками відповідності.

На наступних *третьому, четвертому, п'ятому та шостому рівнях абстрагування* відповідно до моделі рис. 25.2 і залежностей (25.3) відбувається ітеративне перебирання можливих варіацій топології ШНС при відповідних фіксованих (за кращими результатами попереднього ієрархічного рівня) сполучень складових з визначенням базового нейрона (БН), активаційної функції (АФ), архітектури ($A_{\text{ШНС}}$), структури ($C_{\text{ШНС}}$), метода навчання ($MH_{\text{ШНС}}$) і метода спрощення ($MC_{\text{ШНС}}$) нейронної сітки:

$$\begin{aligned}
 &\Phi(\text{БН}) \subset A\Phi \times A_{\text{ШНС}} \times C_{\text{ШНС}} \times MH_{\text{ШНС}} \times MC_{\text{ШНС}}; \\
 &\Phi(\text{БН}, A\Phi) \subset A_{\text{ШНС}} \times C_{\text{ШНС}} \times MH_{\text{ШНС}} \times MC_{\text{ШНС}}; \\
 &\Phi(\text{БН}, A_{\text{ШНС}}) \subset A\Phi \times C_{\text{ШНС}} \times MH_{\text{ШНС}} \times MC_{\text{ШНС}}; \\
 &\quad \dots; \\
 &\Phi(\text{БН}, A\Phi, A_{\text{ШНС}}) \subset C_{\text{ШНС}} \times MH_{\text{ШНС}} \times MC_{\text{ШНС}}; \\
 &\Phi(\text{БН}, A\Phi, C_{\text{ШНС}}) \subset A_{\text{ШНС}} \times MH_{\text{ШНС}} \times MC_{\text{ШНС}}; \\
 &\Phi(\text{БН}, A\Phi, MH_{\text{ШНС}}) \subset A_{\text{ШНС}} \times C_{\text{ШНС}} \times MC_{\text{ШНС}}; \\
 &\quad \dots; \\
 &\Phi(\text{БН}, A\Phi, A_{\text{ШНС}}, C_{\text{ШНС}}) \subset MH_{\text{ШНС}} \times MC_{\text{ШНС}}; \\
 &\Phi(\text{БН}, A\Phi, A_{\text{ШНС}}, MH_{\text{ШНС}}) \subset C_{\text{ШНС}} \times MC_{\text{ШНС}}; \\
 &\quad \dots; \\
 &\Phi(\text{БН}, A\Phi, A_{\text{ШНС}}, C_{\text{ШНС}}, MH_{\text{ШНС}}) \subset MC_{\text{ШНС}};
 \end{aligned} \tag{25.3}$$

З умов визначеності функціонального відображення (25.1) послідовність реалізацій $\Phi_{\text{ШНС}}$ можна подати орграфом^{*)} (рис. 25.3), який і уявляє собою *нижній ієрархічний рівень* ($\Phi_{\text{топ. шнс-процесу}}$) представлення функцій ОК.

^{*)}Орієнтованість орграфа відбиває послідовність реалізації ЛСПС ШНС

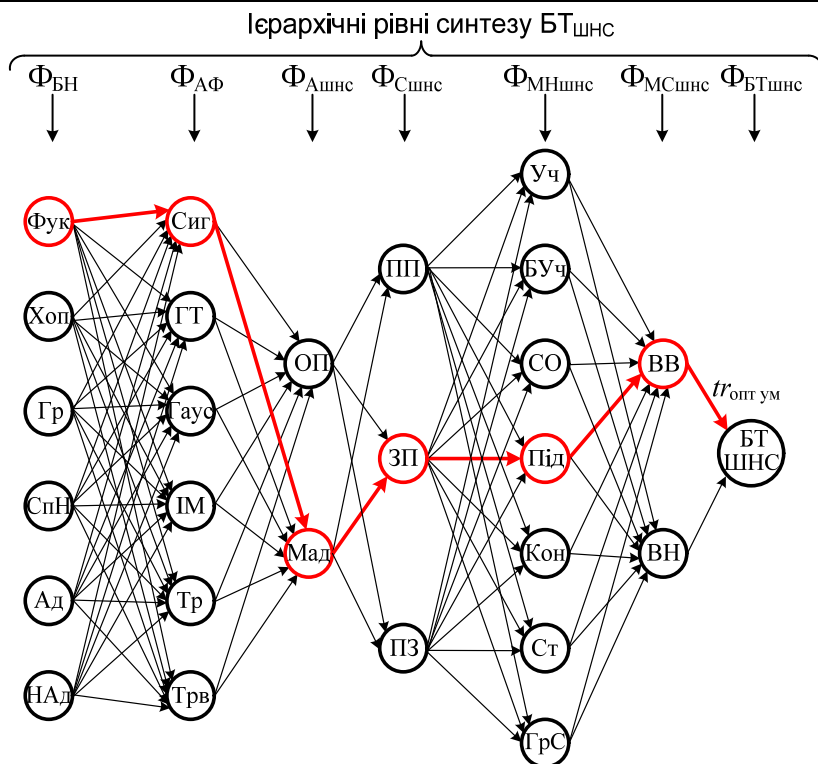


Рис. 25.3. Повний функціональний оргграф ОК – процесу синтезу ШНС с бажаною топологією

Виділення цього рівня ієрархії, що характеризує узагальнену процедурну частину, є надзвичайно важливим етапом системного аналізу/синтезу топології ШНС. Синтезовані на цьому етапі типи функціональних сполучень $\Phi_{\text{склад-процесів}}$ ($\Phi_{\text{БН}}$, $\Phi_{\text{АФ}}$, $\Phi_{\text{Ашнс}}$, $\Phi_{\text{Сшнс}}$, $\Phi_{\text{МНшнс}}$, $\Phi_{\text{МСшнс}}$, $\Phi_{\text{БТшнс}}$) є необхідними і достатніми для розв'язання прикладних задач побудови системи автоматизованого вибору топологій ШНС під певні ВОЗ. На рис. 25.3 використано, відповідно, наступні скорочення: Фук, Хоп, Гр, СпН, Ад, Над – базові нейрони Фукушими, Хопфілда, Гроссберга, Сигма-пі-нейрон, Адаліна та Н-Адаліна (див. розд. 1.3); Сиг, ГТ, Гаус, ІМ, Тр, Трв – активаційні функції сигмоїдна, гіперболічного тангенсу, гаусівська, інверсна мультиквадратична, трикутна, трапецієвидна (див. розд. 1.2); ОП, Мад – архітектури одношарового персептрона і Мадаліни (див. розд. 1.4 і 1.5); ПП, ПЗ, ЗП – структури прямого поширення, повнозв'язні, зворотного поширення (див. розд. 1.4); Уч, Буч, СО, Під, Кон, Ст, ГрС –

методи навчання з учителем, без учителя, шляхом самоорганізації, підкріплюване, конкурентне, стохастичне, градієнтні (див. розд. 4.2); ВВ, ВН – спрощення ШНС видаленням ваг, видаленням нейронів (див. розд. 23); БТ ШНС – бажана топологія ШНС, задовольняюча умовам ВОЗ.

Як вже вище наголошувалося, кількість траєкторій tr процедурного руху за оргграфом, наведеним на рис. 25.3 і відбиваючим реалізацію залежностей (25.3), складатиме понад 3000 варіантів, серед яких є й оптимальні $tr_{\text{опт}}$ за умов відповідності до певних ВОЗ. Зокрема, на рис. 25.3 жирною червоною лінією виділена умовно оптимальна траєкторія $tr_{\text{опт ум}}$, яка в результаті багатоітераційного перебирання визначає складові умовної бажаної топології ШНС:

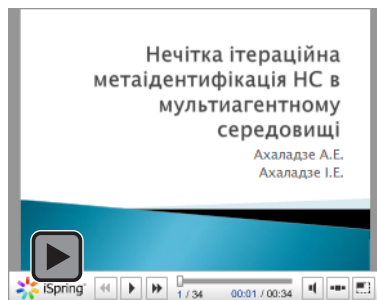
$$BT_{\text{ум}} \text{ ШНС} \rightarrow tr_{\text{опт ум}} \subset \text{Фук} \times \text{Сиг} \times \text{Мад} \times \text{ЗП} \times \text{Під} \times \text{ВВ}. \quad (25.4)$$

Контрольні завдання

1. Прокоментувати вимоги до користувача при виборі топології ШНС, адекватної властивостям розв'язуваної задачі.
2. Охарактеризувати послідовність ітеративного поетапного синтезу штучної нейронної сітки.
3. Розкрити зміст показника “Критерій обслуговуваності при виборі штучної нейронної сітки.
4. Розгумачити сутність конкретної унікальної послідовності дій інтелектуальних агентів в процесі функціонування гнучких інтелектуалізованих мультиагентних конфігурацій.
5. Дати визначення агента/мультиагента мультиагентної системи в задачах реалізації пошуку бажаних топологій ШНС.
6. Розкрити поняття віртуальної топологічної моделі ШНС.
7. Прокоментувати верхній рівень абстрагування концептуальної функції топології ШНС.
8. Навести і прокоментувати постановки прямої та оберненої задач вибору і дослідження ШНС.
9. Охарактеризувати узагальнену концептуальну модель вибору топології ШНС.
10. Розкрити зміст гіперплощин функціональних залежностей в системах нижніх ієрархічних рівней елементарних складових абстрагування віртуальної топологічної моделі ШНС.
11. Пояснити особливості формування і реалізації повного функціонального оргграфа процесу синтезу ШНС з бажаною топологією.
12. Прокоментувати особливості формування умовно оптимальної траєкторії $tr_{\text{опт ум}}$ процедурного руху за оргграфом процесу синтезу ШНС з бажаною топологією.

Презентації

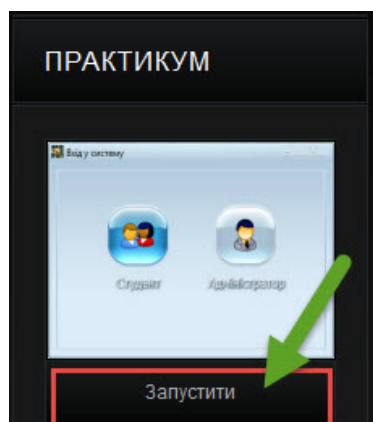
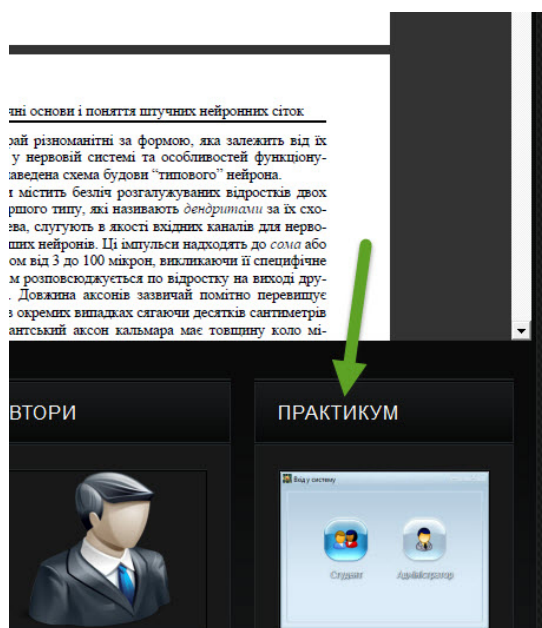
1. Нечітка ітераційна метаідентифікація НС в мультіагентному середовищі



Практичні заняття

Концептуальна модель вибору штучних нейронних сіток

Для запуску практичних занять на
вкладці "Практикум" натисніть
"Запустити"



Розділ 26

Агентно-орієнтована ідентифікація нейронних сіток

*“Видатним досягненням сучасної науки стала програма,
яка на будь-яке запитання дає будь-яку відповідь”*
З першоквітневого випуску «Наркому»

26.1. Модель агентно-орієнтованої ідентифікації топології нейросіток

Розглянемо можливість використання мультиагентної моделі із вбудованим реляційним механізмом для перебирання на основі ЛМПС *функціонально-спеціалізованими інтелектуалізованими агентами* (ФСІА) критеріїв обслуговуваності вимог з боку прикладних задач. Пропонується нижче підхід не є протиставленням більш загальним схемам розподіленого штучного інтелекту, які детально розглядаються в роботі [78]. В даному випадку ідея полягає у тому, щоб, розглядаючи нечіткі ідентифікуючі компоненти як складні *агентно-орієнтовані* (АОП) (або *мультиагентні*) *підсистеми* (МАП), використати переваги нечіткого підходу до керування процесом ідентифікації топологій ШНС самими МАП.

Означення 26.1. *Агентно-орієнтована* (або *мультиагентна*) *підсистема* – це складна система, в якій функціонують два чи більше інтелектуальних агента.

Твердження 26.1. *Завдання ідентифікації топології ШНС* полягає у динамічному побудуванні з наявних *функціонально-спеціалізованих інтелектуалізованих агентів* (ФСІА) таких *гнучких інтелектуалізованих мультиагентних конфігурацій* (ГІМАК) АОП, які в інформаційному просторі необхідного і достатнього НВКО ШНС найкращим чином виконують процедури ЛМПС щодо ітераційного наближення спроможностей поточної топології ШНС до обслуговування критеріїв з боку модельованої задачі.

Означення 26.2. *Метаідентифікація ШНС* – це ітераційна процедура поетапного синтезу топології нейросітки, адекватно задовольняючої умови прикладної задачі.

Твердження 26.2. *Метаідентифікація ШНС* може відбуватися, в тому числі, і за умов нечітких ідентифікуючих ознак як з боку прикладної задачі, так і властивостей поточної топології ШНС.

Означення 26.3. *Функціонально-спеціалізований інтелектуалізований агент* – це програмно-апаратний чи програмно-емуляцій-

ний автономний компонент АОП з функціями метаідентифікації в просторі НВКО ШНС, який функціонує за певним сценарієм/алгоритмом на основі КУПД в інтересах досягнення поставленої користувачем перед нею мети.

Твердження 26.3. *Алгоритм дії ФСІА може мінятися і коректуватися по ходу виконання завдання заради досягнення мети.*

Означення 26.4. *Агентно-орієнтована підсистема – це складна підсистема ГІМАК, в якій функціонують два або більше ФСІА, орієнтованих на розв'язання задач ідентифікації за певною вирішальною класифікаційною ознакою ШНС і утворюючих мультиагентне середовище (МАСе).*

Твердження 26.4. *Сукупність певним чином організованих ФСІА, забезпечуючих визначений вирішальною класифікаційною ознакою рівень ідентифікації ШНС, утворюють МАСе відповідної АОП.*

Твердження 26.5. *В АОП функції ФСІА можуть реалізуватися певними топологіями ШНС.*

26.2. Концепція нечіткої метаідентифікації в задачах вибору нейросіток

Реалізація вищезначеної ідеї можлива за допомогою побудування ГІМАК, особливостями якої є: використання об'єктно-орієнтованих ФСІА всіляких типів, реалізуючих складові класифікатора ШНС [74]; високий ступінь паралелізму; децентралізована структурна і параметрична метаідентифікація в межах АОП (рис. 26.1).

Для узагальнення підходу розглядається більш складний випадок нечіткої метаідентифікації класифікаційних ознак ШНС щодо вимог з боку прикладної задачі. В разі чіткої взаємозалежності “вимога – ознака” реалізація процесу метаідентифікації спрощується.

Означення 26.5. *Гнучка інтелектуалізована мультиагентна конфігурація – така мультиагентна конфігурація, яка: містить агенти $A^{(M)}$ з функціями метаідентифікації, що реалізують механізм розподіленого динамічного виявлення “ступеня важливості” інших агентів із всілякою природою; формує різні закони ідентифікації; забезпечує паралельність роботи агентів різнорідних “шарів”; реагує на зміни стану зовнішнього середовища (вихідних умов задачі) шляхом піднастроювання загального виходу у відповідності з ідентифікацією, задовольняючою поточний набір умов на вході.*

Означення 26.6. *Нечітка ідентифікація топології ШНС – ідентифікація в умовах наявності лінгвістичних невизначеностей на будь-якому етапі реалізації алгоритму синтезу ШНС в інформаційному просторі НВКО при формуванні виведень щодо адекватності поточної ШНС вимогам прикладної задачі.*

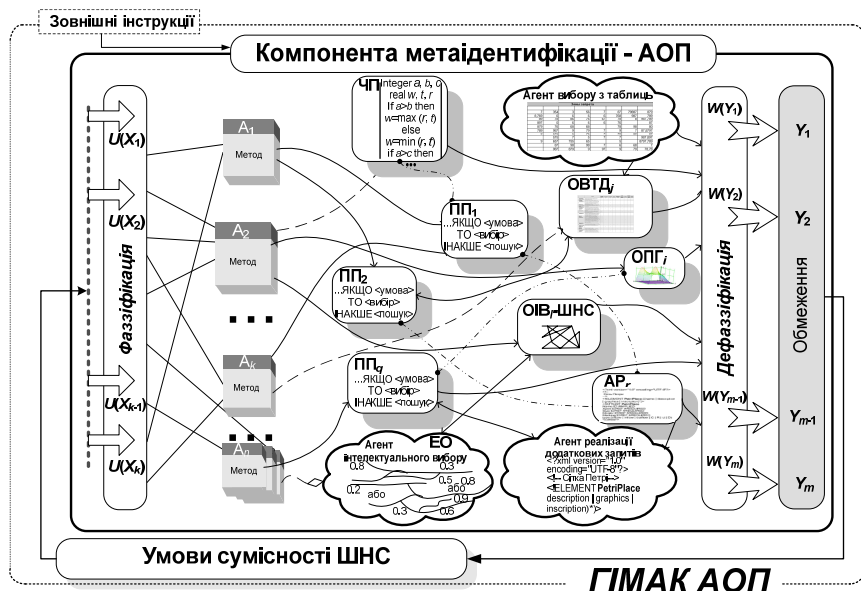


Рис. 26.1. Структура ГІМАК АОП

Означення 26.7. *Агенти з функціями метаідентифікації* – такі ФСІА, які спроможні приймати рішення відносно: активації інших агентів ГІМАК; формування виведень щодо задоволення поточного набору умов на вході АОП.

Мультиагентна структура ГІМАК АОП (рис. 26.1) формується з множини $A = \{A_1, \dots, A_n\}$ зв'язаних між собою ФСІА, утворюючих МАСе АОП. На вхід АОП від зовнішніх джерел (зокрема, і від користувача) надходить множина $U^{(x)} = \{U(X_1), \dots, U(X_k)\}$ значень вхідних змінних $X = \{X_1, \dots, X_k\}$, які відображають умови обслуговуваності властивостей (інформаційне поле) прикладної задачі. Етап фаззифікації ("чіткий \rightarrow нечіткий") трансформує U^x у множину факторів $F^{(x)} = \{F_1^{(x)}, \dots, F_l^{(x)}\}$ (нечітких множин, заданих на значеннях вхідних змінних) з відповідними факторами $C^{(x)} = \{C_1^{(x)}, \dots, C_l^{(x)}\}$ достовірності (ступенями належності за експертними оцінками [77]). Етап дефаззифікації ("нечіткий \rightarrow чіткий") трансформує множину вхідних факторів $F^{(y)} = \{F_1^{(y)}, \dots, F_p^{(y)}\}$ і відповідних факторів достовірності $C^{(y)} = \{C_1^{(y)}, \dots, C_p^{(y)}\}$ у множину $W^{(x)} = \{W(Y_1), \dots, W(Y_k)\}$ зна-

чень умов сумісності $Y = \{Y_1, \dots, Y_m\}$ поточної АОП (з відповідною до певної вирішальної класифікаційної ознаки складової топології ШНС) із заданим на вході набором властивостей прикладної задачі (процедури фаззі-перетворення наведено в [78]).

26.3. Компоненти ГІМАК АОП та їх взаємодія при функціонуванні

Функціонування ГІМАК АОП формується із загального “внеска” її компонент – ФСІА (див. рис. 26.1), які й утворюють відповідне мультиагентне середовище: продукційних правил (ПП), експертних оцінок (ЕО), об'єктів інтелектуального вибору ОІВ – штучних нейросіток (ШНС), чисельних процедур (ЧП), об'єктів вибору з табличних даних (ОВТД), об'єктів побудування графіків (ОПГ), агентів реалізації (АР) – якщо експерт або відповідний агент робить висновки щодо необхідності введення нових правил, обмежень тощо. Базуючись на вхідних даних і меті ідентифікації в залежності від етапу послідовності синтезу топології ШНС [76], ГІМАК АОП реалізує на виході відповідної АОП модель базового інтелектуального нейрона, структуру, архітектуру, метод навчання, потужність ШНС і в цілому як наслідок – її топологію, задовольняючи обслуговування властивостей розв'язуваної задачі, точності апроксимації, складності реалізації, розмірності та швидкодії, кваліфікації користувача.

Позначивши за E , Q відповідно набори (імена) змінних на вході і виході, а за N – набір з усіх змінних (вхідних і на виході включно), приймаючих участь у міжагентному інформаційному обміні $N \supset E, N \supset Q$, а також враховуючи, що структура ГІМАК АОП формується з множини $A = \{A_1, \dots, A_n\}$ взаємозв'язаних ФСІА, тоді, використавши множину компонент зі складових $V_i \subset V \cup A$ (вхідних відносно цього агента A_i) і дій $D_i \subset D \cup A$ (на виході відносно A_i), формальний опис i -го агента A_i можна подати у вигляді:

$$A_i = (T_i, K_i, U_i, W_i, V_i, D_i, H_i, L_i),$$

де T_i — тип агента (ЕС, НС, ЧП); K_i — тип умови для активації (наприклад: K_1 — зміною вхідних даних агента; K_2 — естафетною умовою, тобто завершенням поточного прогону визначених попередніх агентів; K_3 — незалежною активацією, в тому числі відповідно до часового закону; K_4 — у відповідь на запит від іншого агента, і так далі); U_i , W_i — набори вхідних і вихідних змінних агента A_i відповідно; H_i , L_i — набори розташованих вище відносно A_i (чий вказівки він виконує) і підлеглих (відносно A_i) агентів відповідно. При цьому:

$$\forall A_i \in A: (U_i \cap W_i = \emptyset) \wedge (V_i \neq \emptyset) \wedge (A_i \neq \emptyset) \wedge (U_i \neq \emptyset, U_i \subset E \cup (\cup W_j)) \wedge \\ \wedge (W_i \neq \emptyset, W_i \subset Q \cup (\cup U_j)),$$

а реалізація $|H_i| > 0$ вказівок вищерозташованого агента є припустимою. Для відображення факту можливості активізації A_i рішенням інших ФСІА вважаємо, що значення спеціальної “перемикаючої” змінної $\bar{\theta} \in U_i \setminus (E \cup Q)$ визначає його стан: “активний”, “очікування”.

Агенти МАСе відповідної АОП можуть характеризуватися: використовуваною моделлю – N_1 (тобто НС, ПП, ЧП тощо); типом подання – N_2 (наприклад, числовий, символьний або більш детально: фреймовий, семантичний, продукційний тощо. Ясно, що N_1 і N_2 тісно зв’язані); особливостями виконання – N_3 (як окремі фізичні (мікросхеми, комп’ютери) або віртуальні (в межах цієї ж програми) об’єкти і т. д.).

Тип N_2 передбачає необхідність проміжних перетворень даних, інтеграцію множин чисельних значень і формування символьних умов для логічного виведення, і навпаки — перетворення фактів (виведень) у чисельні значення (у випадку нечітких уявлень це відповідає перетворенням “чіткий \rightarrow нечіткий” і “нечіткий \rightarrow чіткий” відповідно). Ці перетворення можуть бути або “вбудованими” у загальне функціонування ФСІА, або виконуються спеціальними процедурами трансформацій. Сам ФСІА, у свою чергу, може також мати мультиагентну структуру. Так, для реалізації перетворень “нечіткий \rightarrow чіткий”, наприклад, за наявності невизначеності при ідентифікації етапів синтезу ШНС такий ФСІА може виконуватися у вигляді фаззі-контролера (див. розд. 4, книга 2 [73]). Кожний такий агент A_i формується з множини специфікацій $SP_i = \{R_{i1}, R_{i2}, \dots, R_{ik}\}$ якісних правил, наприклад, у формі: “ЯКЩО вхідній вимозі відповідає множина X значень змінних, ТО відгуковій реакції вектора ідентифікації у фазовому просторі відповідає множина Y ”. Для математичного опрацювання таких правил необхідні операції взаємодії між окремими нечіткими множинами, які належать до частини “ТО”. Тоді продукування рішень для фаззі-контролерів характеризується взаємодією задаваних експертом вхідних і вихідних функцій належності, причому, операції “І” и “АБО” використовуються як зв’язка в правилах у множинах, елементи яких належать до різних основних множин.

26.4. Узагальнена модель гнучкої інтелектуалізованої мультиагентної системи вибору нейросіток

Наведена на рис. 26.2 узагальнена модель *гнучкої інтелекту-*

алізованої мультиагентної системи (ГІМАС) об'єднує необхідну для формування алгоритму синтезу ШНС і достатню для задоволення вимог з боку розв'язуваних/модельованих прикладних задач сукупність цільових компонент метаідентифікації: АОП_{БМ ШНС}, АОП_А, АОП_{ТАФ}, АОП_С, АОП_{МН}, АОП_Т – базової моделі штучного нейрона, типу активаційної функції, архітектури, структури, метода навчання, топології ШНС та інших (за переліком НВКО [75]).

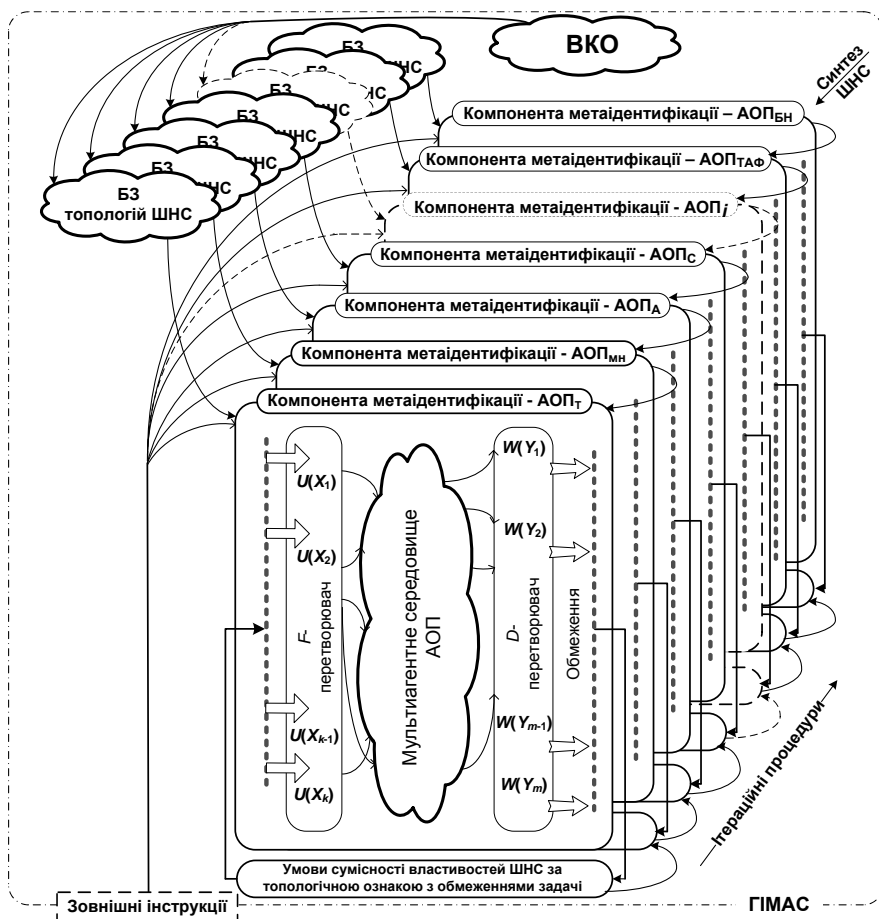


Рис. 26.2. Узагальнена модель ГІМАС вибору ШНС

Означення 26.8. Гнучка інтелектуалізована мультиагентна система – це сукупність ГІМАК АОП, в якій реалізується модель поетапного синтезу ШНС з такою послідовністю їх перебирання в просторі НВКО, яка, будучи виконувана користувачем і/або внутрі-

шнім ініціюючим джерелом, відтворює принципи агентно-орієнтованого підходу та автономно дозволяє виокремити топологію/топології ШНС, здатну/здатні задовольнити критерії обслуговування властивостей розв’язуваної задачі.

Заключна процедура поетапного синтезу ШНС в ГІМАС зводиться до реляційного перебирання агентами ФСІА умов виконання критеріїв обслуговуваності поточним вектором можливостей конкретної топології ШНС вимог з боку модельованої задачі.

Твердження 26.6. В ГІМАС при виконанні етапів синтезу ШНС формування умов переходу до наступної компоненти метаідентифікації може реалізовуватися певними топологіями ШНС з функціями ФСІА.

Твердження 26.7. Оскільки будь-яке застосування ШНС для розв’язання прикладних задач асоціюється з механізмом *прийняття рішень*, рівно як і прийняття рішень, в свою чергу, базується на перебиранні можливих варіантів і виборі задовольняючого заданий показник ефективності, в якості загального розв’язувача поставленої задачі – вибору топології ШНС, яка відповідає “образу” ефективного задоволення критеріїв обслуговуваності вимог прикладної задачі – можна використовувати певну топологію ШНС.

Саме перевірка при реалізації покрокового алгоритму синтезу ШНС виконання умов задоволення певною компонентою АОП відповідних ВОЗ виявляється причиною появи лінгвістичних невизначеностей, опрацювання яких потребує використання методів і засобів нейро-фаззи-технологій із залученням для розв’язання процедур фазифікації та дефазифікації (див. розділи 21.5 та 22.4).

Ітераційні процедури аналізу адекватності синтезованих топологій ШНС дозволяють більш “тонко” відфільтровувати в процесі ідентифікації такі остаточні їх рішення, які у визначеній обмеженні модельованої задачі мірі задовольняють критерій узгодженості за даною властивістю створюваної ШНС.

Необхідні для підтримки процесів ідентифікації компоненти на кожному з кроків алгоритму синтезу ШНС надходять з відповідних баз знань (БЗ), а після узгодження з умовами задачі нові реалізації поповнюють відповідні БЗ, тим самим розширюючи коло прикладних задач, для яких є вже готові розв’язки.

26.5. Інтерпретаційна модель вибору “розв’язувана прикладна задача – топологія штучної нейронної сітки”

На рис. 26.3 наведена узагальнена модель інтерпретації процесу вибору топології ШНС, заснована на реляційних відношеннях [70] між окремими компонентами НРКП, формування якого уявляє *першу складність* реалізації моделі.

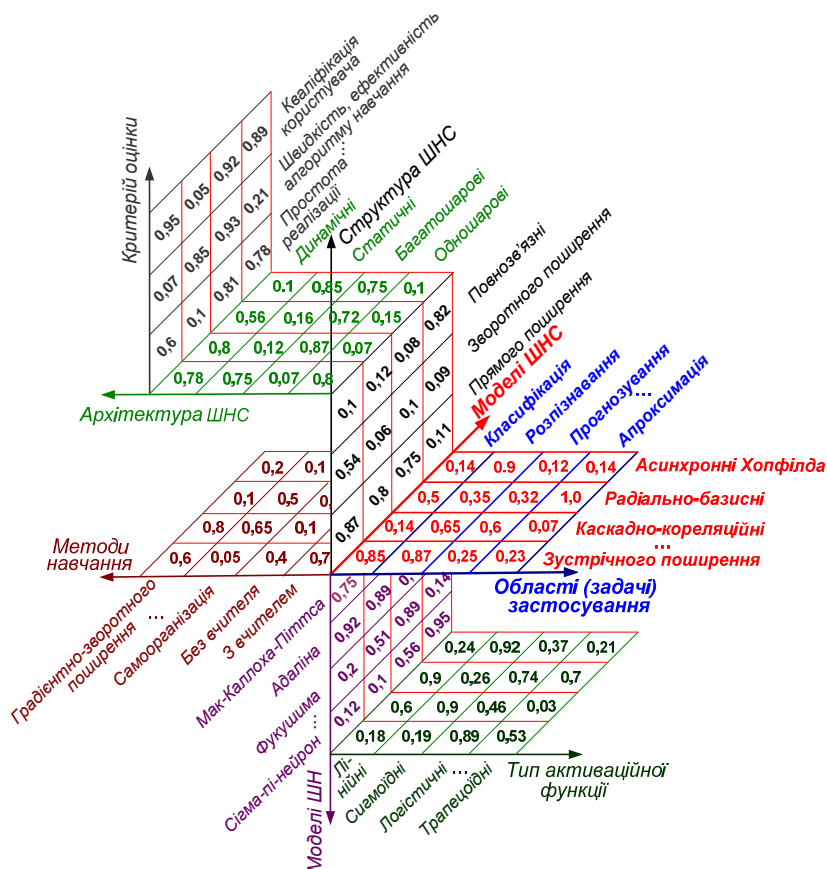


Рис. 26.3. Інтерпретаційна модель вибору топологій ШНС

Друга складність полягає у кількісному визначенні вагомості реляційних зв'язків між вирішними класифікаційними ознаками і реалізується експертним рейтинговим оцінюванням альтернативних варіантів із застосуванням методів ранжирування та попарних порівнянь [78]. Відповідні дослідження потребують глибоких і ретельних проробок щодо однозначності розв'язання цієї частини задачі адекватного вибору ШНС. На рис. 26.3 наведені умовні кількісні результати попередніх опрацювань джерельних даних, які, не претендуючи на строгість експертного оцінювання, можна розглядати як приклад загальної постановки при розв'язанні задачі адекватного вибору ШНС в інтерпретаційній моделі.

Згідно із Твердженням 26.5, в АОП функції ФСІА можна реалізувати певними топологіями ШНС. Отже, кожний з етапів ЛМПС

(рис. 25.1, розд. 25.2) концептуальної моделі процесу вибору топології ШНС (рис. 25.2, розд. 25.3) з послідовностями реалізацій у вигляді траєкторій $tr_{\text{опт}}$ руху (див означення 25.5, розд. 25.3), визначених повним функціональним орграфом процесу синтезу ШНС з бажаною топологією, можна відтворювати певними топологіями ШНС у вигляді *нейросіткової моделі реалізації такого графа* (рис. 26.4).

За таких умов процедура синтезу бажаної топології ШНС буде полягати у настроюванні ваг кожної зі складових $\Phi_{\text{БН}}$, $\Phi_{\text{АФ}}$, $\Phi_{\text{Ашнс}}$, $\Phi_{\text{Сшнс}}$, $\Phi_{\text{МНшнс}}$, $\Phi_{\text{БТшнс}}$ нейросіткової моделі на поточні значення експертних рейтингових оцінювань альтернативних варіантів з урахуванням вагомості реляційних зв'язків (див. рис. 26.3) згідно вимог ВОЗ для відповідного етапу синтезу. В решті решт, виконуючи ітераційні процедури настроювання, можна згідно із *Означенням 25.5* (розд. 25.3) отримати траєкторію $tr_{\text{опт}}$ руху, визначувану перетином складових топології ШНС з максимальними до ВОЗ показниками відповідності. Ті самі міркування в процесі синтезу бажаної топології ШНС стосуються і процедур керування етапами узагальненої моделі ГІМАС (рис. 26.2).

Як приклад, на рис. 26.4 червоною лінією виділена умовно оптимальна траєкторія $tr_{\text{опт ум}}$, яка в результаті багатоітераційного перебирання визначає складові умовної бажаної топології ШНС (функція (25.4), наведена у розд. 25.3):

$$\text{БТ}_{\text{ум}} \text{ ШНС} \rightarrow tr_{\text{опт ум}} \subset \text{Фук} \times \text{Сиг} \times \text{Мад} \times \text{ЗП} \times \text{Під} \times \text{ВВ}.$$

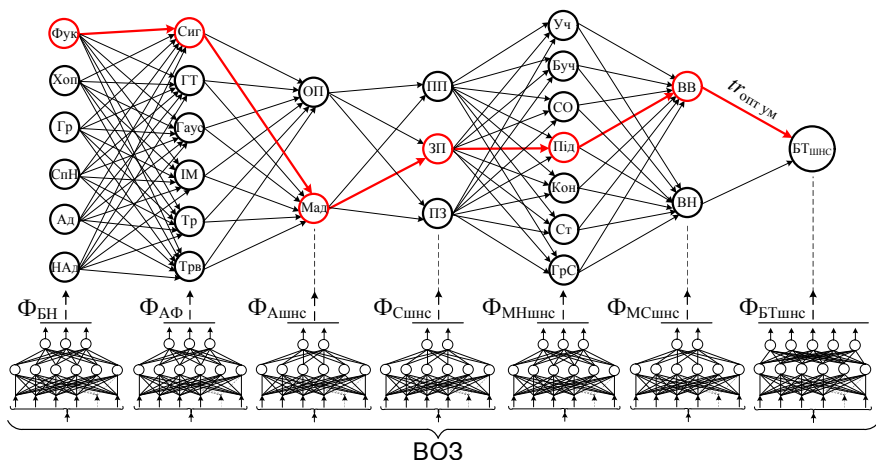


Рис. 26.4. Нейросіткова модель реалізації повного функціонального орграфа синтезу ШНС з бажаною топологією

Запропонований в розділах 2, 25, 26 підхід щодо автоматизованого синтезу ШНС з бажаною топологією не претендує на вичерпне розв'язання цієї архіважливої і вкрай трудомісткої задачі, але надає уявлення про шляхи подолання протиріччя між обмеженими можливостями користувача і різноманітністю прикладних задач.

Частіше за все використовують *програми-нейроімітатори* (емуляції) – вони працюють на звичайних комп'ютерах, а сучасні нейроалгоритми дозволяють швидко опрацьовувати значні обсяги інформації, тобто використовувати ШНС для розв'язання складних реальних задач.

26.6. Можливі способи застосування та реалізації вибраної (синтезованої) топології штучних нейронних сіток

Програми охоплюють як набори базових операцій по створенню, навчанню і маніпулюванню нейронними сітками, вхідними даними, властивостями НС і нейрорішень, так і автоматизовані процедури виконання найбільш рутинних ланцюжків дій, зокрема, щодо визначення оптимальних налаштувань сітки та алгоритму навчання.

Універсальні сучасні нейропрограми пропонують можливості згенерувати опис навченої НС на деякій мові програмування – для того, щоб отриманий програмний модуль можна було “вмонтувати” в програму користувача. Також програма користувача може використовувати засоби міжпрограмної комунікації (DDE, OLE, COM в середовищі Windows) для організації доступу до нейросіток в універсальних нейропрограмах (нейросітка залишається функціонувати в середовищі утворившої її нейросіткової програми, а програма користувача тільки формує необхідні команди-дані та отримує результати).

На *специфічні обчислювальні архітектури* (наприклад, сигнальні процесори) зазвичай переносять згенерований програмою-нейроімітатором опис навченої НС мовою програмування (наприклад, мовою C). Для ПЛИС/PLM/FPGA-кристалів існують і CAD-системи, в яких номенклатуру нейроелементів можна створити-описати у виді набору модулів і потім з цих модулів шляхом вибору і “перетягання” мишкою скласти необхідну топологію-архітектуру НС, а далі проект скомпілювати і “прошити” кристал. Сьогодні вже можливо створювати нейрокомп'ютери на базі відеомап (графічних процесорів).

Отже, існує широкий спектр досить універсальних способів організації інструментальних засобів і процесу застосування НС на різноманітній програмно-апаратній базі – все визначається властивостями задачі і вимогами до її розв'язання, а також компетенцією користувача або розробника. Але вибір саме топології НС, яка й відбиває властивості і задовольняє вимоги розв'язуваної задачі, поки ще віддається на розсуд користувача і не позбавлений суб'єктивізму.

26.7. Узагальнення та висновки

Можна констатувати, що з найпоширеніших сфер застосування ШНС є *розпізнавання образів*. Це тим важливіше враховувати через те, що у більшості інших застосувань задача розпізнавання, навіть не будучи метою використання у явному вигляді, все ж припускає цю процедуру при надходячих на вхід образах з наступною їх класифікацією, кластеризацією тощо. Під кластеризацією мають на увазі розбиття множини вхідних образів на класи, коли ані кількість, ані ознаки класів не відомі. Сітка може констатувати, що вхідний образ не відноситься до жодного з виділених класів – це свідчить про те, що з'явилися нові образи з відмінними ознаками від тих, що визначають належність до певного класу. Тобто сітка виявляє нові, невідомі раніше класи образів, виконуючи функції *кластеризатора*.

З іншого боку, будь-яке застосування ШНС для розв'язання прикладних задач асоціюється з механізмом *прийняття рішень при керуванні*, рівно як і прийняття рішень, в свою чергу, базується на перебиранні можливих варіантів і виборі задовольняючого заданий показник ефективності. А це й є визначенням такого стану ШНС, яке відповідає “образу” ефективного керування [68].

Прогнозування можна віднести до так званої “імпровізуючої” властивості ШНС у вигляді узагальнення і виокремлення схованих взаємозв'язків між даними на вході і виході. Інакше кажучи, якщо у вхідній виборці даних є сховані закономірності їх впливу на змінювання даних на виході, то можна використовувати *ШНС як прогнозуючий інструмент*. А вже далі, на підставі цієї прогнозуючої функції, ШНС може приймати рішення при керуванні.

В багатьох випадках буває необхідно підібрати для функції, заданої тільки таблично або графіком і відображаючої результати експериментальних даних, аналітичний вираз, відбиваючий ці залежності. Формула ж може виявитися досить складною і некоректною для вимаганої мети (наприклад, функція має бути проінтегрованою, а інтеграл від неї не виражається через елементарні функції). Тоді емпіричні формули для зображення необхідної функції $f(x)$ вибирають наближену функцію $g(x)$ як набір з функцій певного виду, вимагаючи, щоб функція $g(x)$ якомога ближче співпадала з $f(x)$ на деякому інтервалі ($a \leq x \leq b$). Нелінійна характеристика при цьому може бути довільною: від сигмоїдальної до будь-якого хвильового пакету, синуса або багаточлена. Від вибору нелінійної функції може залежати складність ШНС, але з будь-якою нелінійністю сітка залишається універсальним *апроксиматором* і при коректному виборі структури може

досить точно апроксимувати функціонування будь-якого неперервного автомата.

Теорема 26.1 (про повноту): *будь-яка неперервна функція на замкненій обмеженій множині може бути рівномірно наближена функціями, обчислюваними нейронами ШНС, якщо активаційна функція нейрона двічі неперервно диференціювана і неперервна.*

Отже, нейросітки є універсальними структурами, дозволяючими реалізувати будь-який обчислювальний алгоритм.

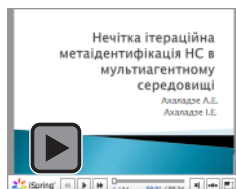
Зрештою, наведений в розділі 2 класифікатор ШНС, побудований на НВКО, а також запропонована вище інтерпретаційна модель вибору адекватної топології ШНС є основою для автоматизації цього процесу, що дозволить усунути суб'єктивний фактор некомпетентності користувача і реалізувати в автоматизованому виді логічну схему поетапного синтезу ШНС.

Контрольні завдання

1. Охарактеризувати завдання ідентифікації та метайдентифікації топології штучної нейронної сітки.
2. Розкрити зміст поняття “Функціонально-спеціалізований інтелектуалізований агент”.
3. Навести основні засади концепції нечіткої ідентифікації в задачах вибору штучних нейронних сіток.
4. Охарактеризувати загальну структуру гнучкої інтелектуалізованої мультиагентної конфігурації агентно-орієнтованих підсистем.
5. Розгумачити задачі нечіткої ідентифікації топології штучної нейронної сітки.
6. Прокоментувати взаємодію компонент гнучкої інтелектуалізованої мультиагентної конфігурації агентно-орієнтованих підсистем в умовах їх функціонування.
7. Розкрити структуру і прокоментувати ітераційні процедури відбирання адекватної штучної нейронної сітки узагальненою моделлю гнучкої інтелектуалізованої мультиагентної системи.
8. Розгумачити особливості побудови інтелектуалізованої моделі вибору топологій штучних нейронних сіток і організацію пошукових процедур з реалізацією механізму прийняття рішень.
9. Охарактеризувати способи застосування та реалізації вибраної топології штучної нейронної сітки.
10. Пояснити особливості побудування та функціонування нейросіткової моделі реалізації повного функціонального орграфа синтезу ШНС з бажаною топологією.

Презентації

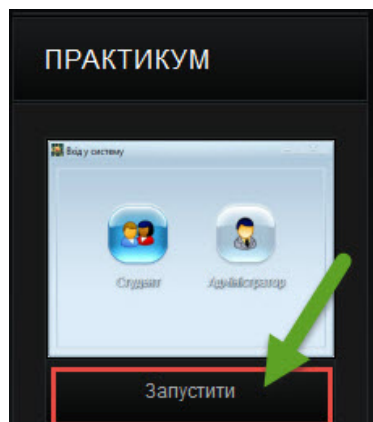
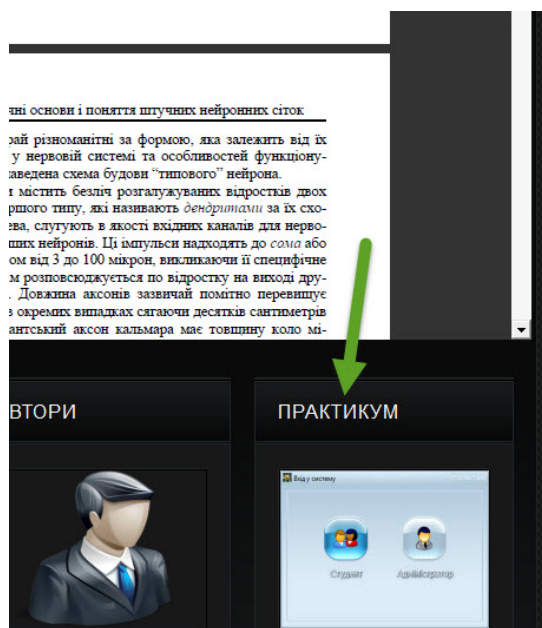
1. Нечітка ітераційна метаідентифікація НС в мультіагентному середовищі



Практичні заняття

Агентно-орієнтована ідентифікація нейронних сіток

Для запуску практичних занять на
вкладці "Практикум" натисніть
"Запустити"



Розділ 27

Нейрокомп'ютери

Чим більше нам дано, тим більшого ми жадаємо
Луцій Анней Сенека молодший

27.1. Загальні зауваження*)

Теорія нейросіток не внесла революційних новацій в алгоритми адаптації та оптимального управління. Самонавчувні системи відомі давно, теорія адаптивних регуляторів теж детально розроблена, все це широко застосовується у техніці. Теорія НС засвоює розроблені раніше методи і прагне їх пристосувати для створення все більш ефективних нейронних систем. Особливу важливість використання нейроструктур набуває з точки зору продуктивності ЕОМ. Згідно з гіпотезою Мінського реальна продуктивність типової паралельної обчислювальної системи з n процесорів зростає як $\log(n)$ (тобто продуктивність системи з 100 процесорів всього вдвічі вище, ніж продуктивність 10-процесорної системи – процесори довше чекають своєї черги, ніж обчислюють). Проте, якщо використати для розв'язання задачі ШНС, то паралелізм може бути використаний практично повністю, і продуктивність зростає “майже пропорційно” n .

В підручнику (зокрема, у передмові) неодноразово наводилися означення *нейрокомп'ютера* (НК). Наведемо ще одне, більш широко тлумачення цього терміну через те, що *нейрокомп'ютинг* як науковий напрям розробки обчислювальних систем шостого покоління (тобто саме НК) формується в окрему гілку знань лише наприкінці минулого сторіччя, і багато з яких термінів та означень нейрокомп'ютингу не сформувалися в строгу низку установлених визначень.

Означення 27.1. *Нейрокомп'ютинг* – це технологія створення систем опрацювання інформації (наприклад, НС), які спроможні автономно генерувати методи, правила та алгоритми опрацювання у виді адаптивної відповіді в умовах функціонування в конкретному інформаційному середовищі. Тут на місце програмування, по суті, приходять *навчання НК*.

Означення 27.2. *Нейрокомп'ютер* – це спеціалізований *нейрообчислювач* (НО) (від “*to compute*” – обчислювати), призначений для *обслуговування послідовності трьох стадій життєвого циклу ШНС*: створення (або вибору топологій сіток за ознакою їх обслуговування прикладної задачі), навчання і застосування (див. розд. 4.1).

*) Розділ 27 складений з залученням матеріалів огляду: *Шахнов В. А. Нейрокомп'ютери – архітектура і реалізація. Обзор / В. А. Шахнов, А. И. Влазов, А. С. Кузнецов, Ю. А. Поляков* – М.: МГТУ им. Н. Э. Баумана, 2000

Нейротехнології та нейрокомп'ютерні системи

В табл. 27.1 наведені відмінності в тлумаченні означення НК в залежності від наукового напрямку їх застосування.

Таблиця 27.1.

Порівняльні характеристики визначень нейрокомп'ютера

№	Науковий напрям	Визначення нейрообчислювальної системи
1	Математична статистика	<i>Нейрокомп'ютер</i> - це обчислювальна система, автоматично формуюча опис характеристик випадкових процесів або їх сукупності, які мають складні, часто апіорі невідомі функції розподілення
2	Математична логіка	<i>Нейрокомп'ютер</i> - обчислювальна система, алгоритм роботи якої представлений логічною сіткою елементів окремого виду - нейронів, з повною відмовою від булевих елементів типу І, АБО, НІ
3	Порогова логіка	<i>Нейрокомп'ютер</i> - обчислювальна система, алгоритм розв'язання задач в якій представлений у виді сітки порогових елементів з динамічно перенастроюваними коефіцієнтами та алгоритмами настроювань, незалежними від розмірності сітки порогових елементів та їх вхідного простору
4	Обчислювальна техніка	<i>Нейрокомп'ютер</i> - обчислювальна система з MSIMD архітектурою, в якій процесорний елемент однорідної структури спрощений до рівня нейрона, різко ускладнені зв'язки між елементами і програмування перенесено на змінення вагових коефіцієнтів зв'язків між обчислювальними елементами
5	Медицина (нейробіологічний підхід)	<i>Нейрокомп'ютер</i> - обчислювальна система, яка уявляє собою модель взаємодії клітинного ядра, аксонів і дендритів, пов'язаних синаптичними зв'язками (синапсами) (тобто модель біохімічних процесів, протікаючих в нервових тканинах)
6	Економіка та фінанси	Устояного визначення нема, але частіше за все під нейрообчислювачем розуміють систему, яка забезпечує паралельне виконання "бізнес-транзакцій" з елементами "бізнес-логіки"

Отже, НК опрацьовує інформацію на основі принципів роботи природних нейронних систем. Ці принципи були формалізовані, що дозволило говорити про теорію ШНС. Проблематика НК полягає в побудові реальних фізичних пристроїв, що дозволяє не просто мо-

делювати ШНС на звичайному комп'ютері, але так змінювати принципи роботи комп'ютера, що стає можливим говорити про те, що вони працюють у відповідності з теорією ШНС.

Нейрокомп'ютери – це ЕОМ нового покоління, яке якісно відрізняється від інших класів обчислювальних систем паралельного типу тим, що для розв'язання задач *вони використовують не заздалегідь розроблені алгоритми, а спеціальним чином підібрані приклади, на яких навчаються*. Їх поява обумовлена об'єктивними причинами: розвитком елементної бази, яка дозволяє на одній платі реалізувати персональний комп'ютер – повнофункціональний комп'ютер (модель нейрона), та необхідністю розв'язання важливих практичних задач, поставлених дійсністю.

В той же час, при прийнятті рішення щодо використання НК для розв'язання конкретних задач треба приймати до уваги як їх позитивні, так і негативні властивості.

Переваги НК у порівнянні зі звичайним комп'ютерами полягають, перш за все, у наступному:

- по-перше, висока швидкодія, що пов'язане з високим ступенем паралельності алгоритмів нейроінформатики;
- по-друге, нейросистеми стають надстійкими до завад і руйнувань;
- по-третє, стійкі та надійні нейросистеми можуть утворюватися з ненадійних елементів, які мають значний розкид параметрів.

Недоліки НК визначаються тим, що вони:

- створюються спеціально для розв'язання конкретних задач, пов'язаних з нелінійною логікою та теорією самоорганізації. Розв'язання таких задач на звичайних комп'ютерах можливе тільки чисельними методами;
- в силу своєї унікальності досить коштовні.

Додамо тільки, що *ефективне застосування НК є характерним для випадків, які потребують різкого скорочення часу опрацювання при розв'язанні просторових задач підвищеної розмірності*.

В подальшому викладі матеріалу даного розділу під нейрокомп'ютером будемо розуміти будь-яку обчислювальну систему з MSIMD архітектурою (означення №4, табл. 27.1).

27.2. Архітектура обчислювальних систем

Перш, ніж перейти до образу сучасних нейрообчислювачів та їх елементної бази, розглянемо класифікацію архітектур обчислювальних систем за Б. М. Каганом^{*)}:

^{*)} Каган Б. М. Электронные вычислительные машины и системы / Б. М. Каган. – М.: Энергоатомиздат, 1991. – 592 с.

- обчислювальна система з одним потоком команд і даних (однопроцесорна EOM – SISD) (рис. 27.1, а);
- обчислювальна система із загальним потоком команд (SIMD – одиничний потік команд та численний потік даних) (рис. 27.1, б);
- обчислювальна система із численним потоком команд та одиночним потоком даних (MISD – конвеєрна EOM) (рис. 27.1, в);
- обчислювальна система із численним потоком команд та даних MIMD (рис. 27.1, г).

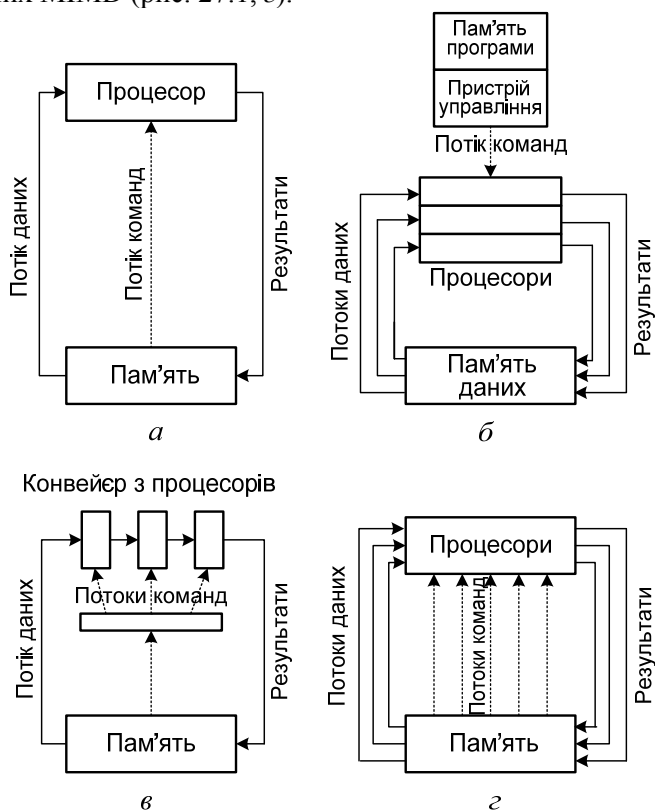


Рис. 27.1. Архітектури обчислювальних систем

Як вже вище наголошувалося, НК - це обчислювальна система з MSIMD архітектурою, тобто з паралельними потоками однакових команд та численним потоком даних. Розглянемо, яким чином співвідносяться нейрокомп'ютери з паралельними EOM. З аналізу даних, наведених в табл. 27.2, можна виділити три основні напрямки розвитку обчислювальних систем з масовим паралелізмом (ОСМП),

для кожного з яких сьогодні існують рішення, які реалізують ті чи інші нейросіткові парадигми.

Таблиця 27.2.

Основні напрямки розвитку ОСМП

№	Найменування напрямку	Опис
1	ОСМП на базі каскадного з'єднання універсальних SISD, SIMD, MISD мікропроцесорів	Елементна база – універсальні RISC або CISC процесори: Intel, AMD, Sparc, Alpha, Power PC, MIPS тощо
2	На базі процесорів з розпаралелюванням на апаратному рівні	Елементна база – DSP процесори: TMS, ADSP, Motorola
3	ОСМП на спеціалізованій елементній базі	Елементна база – від спеціалізованих однокітних процесорів до нейрочипів

Для спрощення аналізу в подальшому будемо вважати, що нейросіткові системи, реалізовані на апаратних платформах *першого напрямку* (нехай і мультипроцесорних) будемо відносити до **нейроемуляторів** – тобто системам, які реалізують типові нейрооперації (зважене підсумовування та нелінійне перетворення) на програмному рівні. Нейросіткові системи, які реалізовані на апаратних платформах *другого та третього напрямків* у виді плат розширення стандартних обчислювальних систем (1-го напрямку) дістали назву **нейроприско-рювачів** (зважене підсумовування, як правило, реалізується апаратно, наприклад, на основі трансверсальних фільтрів, а нелінійні перетворення – програмно). Нарешті системи, які реалізовані на апаратній платформі третього напрямку у виді функціонально закінчених обчислювальних пристроїв, відносять до **нейрокомп'ютерів** (всі операції виконуються у нейросітковому логічному базисі).

Враховуючи вищенаведений аналіз, сформулюємо більш узагальнюючі та точні, на наш погляд, означення та твердження:

Твердження 27.1. НК можна віднести до обчислювальних систем з високим паралелізмом (MSIMD архітектури), які реалізовані на основі спеціалізованої елементної бази, орієнтованої на виконання нейросіткових операцій в нейросітковому логічному базисі.

27.3. Елементна база нейрообчислювачів

Елементна база сучасних НК, наведена на рис. 27.2, містить як процесори загального призначення (GPP), так і вбудовані мікрокон-

тролери (mC), трансп'ютери, замовлювані кристали (ASIC), а також програмовувані логічні інтегральні схеми (ПЛИС), цифрові сигнальні процесори (DSP – Digital Signal Processor) та нейрочіпи. Використання наведених елементів дозволяє сьогодні реалізовувати НО, які функціонують у реальному масштабі часу.

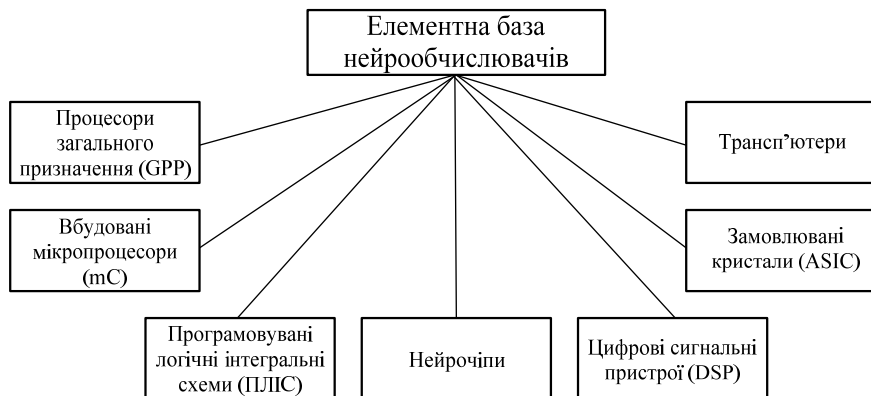


Рис. 27.2. Елементна база нейрокомп'ютерів

Далі більш детально розглядаються програмовувані схеми ПЛИС, сигнальні процесори DSP та нейрочіпи як основні базові елементи саме нейрообчислювачів.

27.4. Апаратна реалізація елементів нейрологіки

Одним з основних достоїнств НО є те, що його основу складають відносно прості, частіше за все – однотипові елементи, які імітують роботу нейронів мозку –“нейрони”, кожний з яких характеризується своїм поточним станом і за аналогією з нервовими клітинами головного мозку можуть бути збудженими або загальмованими.

Властивості нейронів та всіляких топологій нейронних сіток детально розглянуто в попередніх розділах, а розглядаючи наведену в *розділі 2* підручника класифікацію НС, можна простежити вплив на їх вибір та функціонування основних класифікаційних ознак.

Для того, щоб ШНС працювала, її треба навчати, і від якості навчання залежить спроможність сітки розв'язувати поставлені перед нею проблеми. Методи навчання ШНС викладено в *розділі 4*. Тут тільки зазначимо, що на етапі навчання крім параметра якості підбирання вагових коефіцієнтів важливу роль відіграє час навчання. Як правило, ці два параметра пов'язані оберненою залежністю і їх доводиться вибирати на основі компромісу.

Практично 80% реалізованих на сьогодні нейрочіпів, орієнтованих на задачі цифрового опрацювання сигналів, використовують при навчанні ШНС алгоритм зворотного поширення похибки, який, крім іншого, став певним еталоном для вимірювання продуктивності НО (наприклад, як БПФ на 1024 відліку для сигнальних процесорів).

Як вже наголошувалося вище в *розділах* 23-26, в загальному випадку проектування нейросистем уявляє собою складний та трудомісткий процес, в якому вибір конкретного алгоритму – це тільки один з декількох кроків процесу проектування.

Питання, які стоять перед розробниками нейросіткової елементної бази та НО, складні і вимагають проведення детальних досліджень: як доцільніше реалізувати нейрочіп – із вбудованими нелінійними перетвореннями (нехай і фіксованого набору, але реалізованих апаратно) або дозволити програмісту-розробнику ШНС самому формувати активаційну функцію програмно, розміщуючи відповідний код в *програмно-задаючому пристрої* (ПЗП). Чи варто гнатися за універсальністю нейрочіпу, яка б дозволила реалізовувати будь-які топології та парадигми, чи слід орієнтуватися на випуск спеціалізованих об'єктно-орієнтованих нейрочіпів, чи використовувати базові матричні кристали та програмовувані логічні інтегральні схеми (ПЛІС)? Всі ці питання ще чекають на відповідь.

27.4.1. Нейрообчислювачі на елементній базі DSP та ПЛІС

Елементною базою нейрообчислювальних систем другого та третього напрямків (див. *табл.* 27.2) є відповідно заказні кристали (ASIC), вмонтовувані мікроконтролери (mC), процесори загального призначення (GPP), програмовувана логіка (FPGA – ПЛІС), трансп'ютери, цифрові сигнальні процесори (DSP) та нейрочіпи. Причому використання як тих, так і інших дозволяє реалізовувати НО, які функціонують в реальному масштабі часу, проте найширше при реалізації НО використовуються ПЛІС, DSP та нейрочіпи.

Трансп'ютери (T414, T800, T9000) та, зокрема, трансп'ютероподібні елементи є важливими для побудування обчислювальних систем з масовим паралелізмом, а їх застосування поступово зсувається в бік комутаційних систем та мереж ЕОМ, хоча ще залишаються приклади реалізації на них шарів деяких ЕОМ з масовим паралелізмом у виді ґрат процесорних елементів.

Стрімкий перехід сучасних систем управління на цифрові стандарти привів до необхідності опрацьовувати з високою швидкістю досить великі обсяги інформації. Складне опрацювання та фільт-

рація сигналів, наприклад, розпаковка стиснутих аудіо- та відеоданих, маршрутизація інформаційних потоків тощо вимагає застосування досить продуктивних обчислювальних систем. Подібні системи можуть бути реалізовані на різній елементній базі, але найбільше розповсюдження отримали пристрої із застосуванням сигнальних процесорів та ПЛІС.

Процесор DSP, володіючи потужною обчислювальною структурою, дозволяє реалізовувати всілякі алгоритми опрацювання інформаційних потоків. Порівняно невисока ціна та розвинені засоби розробки програмного забезпечення дозволяють легко застосовувати їх при побудуванні обчислювальних систем з масовим паралелізмом.

Програмовувана логіка спроможна працювати на більш високих частотах, проте, оскільки управління реалізоване апаратно, то змінювання алгоритмів роботи потребує перепрограмування інформаційної системи. Низька тактова частота DSP поки що обмежує максимальну частоту опрацьовуваного аналогового сигналі до рівня в 10-20 МГц, але програмне управління дозволяє досить легко змінювати не тільки режими опрацювання, але й функції, виконувані DSP. Окрім опрацювання та фільтрації даних DSP можуть здійснювати маршрутизацію цифрових потоків, вироблення управляючих сигналів і навіть формування сигналів системних шин ISA, PCI тощо.

Особливістю використання DSP та ПЛІС в якості елементної бази НО є те, що орієнтація у виконанні нейросіткових операцій обумовлює, з одного боку, підвищення швидкостей обміну між пам'яттю та паралельними арифметичними пристроями, а з іншого – скорочення часу вагового підсумовування (множення та накопичення) за рахунок застосування фіксованого набору команд типу регістр-регістр.

Процесори DSP як елементна база нейрообчислювачів. Цифрові процесори DSP вже на протязі декількох десятиріч є елементною базою для побудування як нейроприскорювачів, так і контуру логіки загальносистемного управління НК.

При створенні нейрообчислювальних систем на базі цих процесорів слід пам'ятати, що вони володіють високим ступенем спеціалізації. В них широко використовуються методи скорочення тривалості командного циклу, характерні для універсальних RISC-процесорів, зокрема: конвеєризація на рівні окремих мікроінструкцій та інструкцій, розміщення операндів більшості команд в регістрах, використання тіньових регістрів для збереження стану обчислень при перемиканні контексту, розділення шин команд та даних. В той же час для сигнальних процесорів характерним є наявність апаратного помножувача, дозволяючого виконувати множення як мінімум двох

чисел за один командний такт. Іншою відмінністю процесорів DSP є введення в систему команд наступних операцій:

- множення з накопиченням MAC ($C=A \times B + C$ з вказаним в команді числом виконань в циклі та з правилом змінення індексів використовуваних елементів масивів A та B , тобто вже реалізовані прообрази базових нейрооперацій – зважене підсумовування з накопиченням);
- інверсія біт адреси;
- всілякі бітові операції.

В сигнальних процесорах реалізується апаратна підтримка програмних циклів, кільцевих буферів. Один або декілька операндів витягаються з пам'яті в циклі виконання команди.

Інтегральні схеми ПЛІС як елементна база нейрообчислювачів. Окремо розглянемо можливість створення паралельних обчислювачів (в тому числі й нейро) на базі ПЛІС, адже на ПЛІС можна реалізовувати системи як другого, так і третього типу (див. *табл. 27.2*).

Побудова НО на базі ПЛІС, з одного боку, дозволяє гнучко реалізовувати всілякі нейросіткові парадигми, а з іншого пов'язано з проблемами розведення всіх необхідних міжз'єднань. Виготовлювані в теперішній час ПЛІС мають різні функціональні можливості (з кількістю вентилів від 5 до 100 тисяч). НО на базі ПЛІС як правило позиціонуються як гнучкі нейрообчислювальні системи для науково-дослідних цілей та малосерійного виробництва. Для побудови більш продуктивних та ефективних НО як правило пріоритети віддаються сигнальним процесорам.

Також останнім часом широке розповсюдження отримали гібридні НО, коли блок опрацювання даних реалізується на DSP, а логіка управління – на ПЛІС. В теперішній час безліч фірм у світі займається розробкою та випуском всіляких ПЛІС, проте, лідерство ділять дві фірми: Xilinx та ALTERA. Виокремити продукцію якої-небудь однієї з них неможливо через те, що як за технічними характеристиками, так і за архітектурними композиціями вони дуже мало відрізняються.

27.4.2. Нейрочіпи як елементна база нейрообчислювачів

Елементною базою перспективних НО є нейрочіпи, виробництво яких налагоджується в багатьох країнах світу, причому більшість з вироблюваних нейрочіпів на сьогодні орієнтовані на закрите використання (тобто створювались для конкретних спеціалізованих управляючих систем). На рис. 27.3 наведена узагальнена класифікація нейрочіпів за їх основними класифікаційними ознаками:

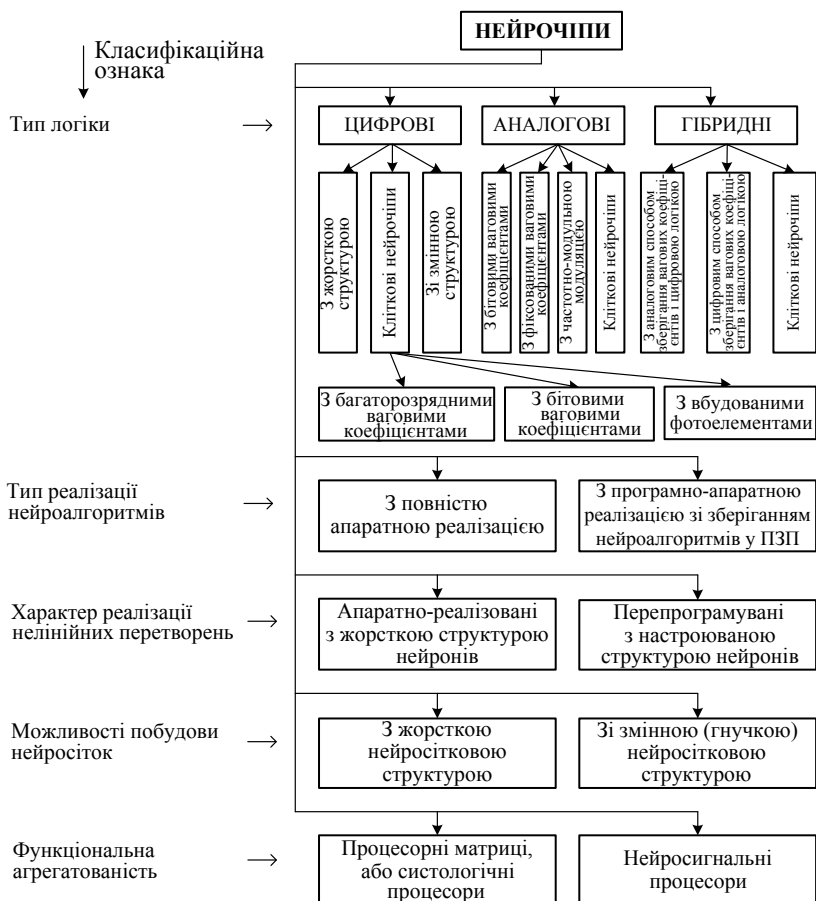


Рис. 27.3. Узагальнена класифікація нейрочіпів

- *тип логіки* (цифрові, аналогові, гібридні);
- *тип реалізації нейроалгоритмів* (з повністю апаратною реалізацією та з програмно-апаратною реалізацією зі зберіганням нейроалгоритмів у ПЗП);
- *характер реалізації нелінійних перетворень* (апаратно-реалізовані з жорсткою структурою нейронів, перепрограмовані з настроюваною структурою нейронів);
- *можливості побудови нейросіток* (з жорсткою та змінною (гнучкою) нейросітковою структурою);
- *функціональна агрегатованість* (процесорні матриці, або систологічні процесори – близькі до звичайних RISC процесорів чіпи, які об'єднують в своєму складі деяку кількість про-

цесорних елементів, тоді як решта логіки, як правило, реалізується на базі периферійних схем; нейросигнальні процесори – виокремлені в самостійний клас чіпи, ядро яких уявляє собою типовий сигнальний процесор, а реалізована на кристалі доповняльна логіка забезпечує виконання нейросіткових операцій – наприклад, доповняльний векторний процесор тощо).

На сьогодні виділяють *дві базові лінії розвитку ОСМП*:

- з модифікованими послідовними алгоритмами, характерними для однопроцесорних фоннейманівських алгоритмів;
- на основі принципово нових надпаралельних нейросіткових алгоритмів розв'язання всіляких задач (на базі нейроматематики).

Для оцінки продуктивності НО використовуються наступні показники:

- кількість змінених значень ваг за секунду (connections update per second CUPS) – *оцінює швидкість навчання*;
- кількість з'єднань (множень з накопиченням) за секунду (connections per second CPS) – *оцінює продуктивність*;
- кількість з'єднань примітивів за секунду (connections primitives per second CPPS): $CPPS = CPS * Bw * Bs$, де Bw, Bs – розрядність ваг та синапсів;
- кількість мільйонів множень з накопиченням за секунду (millions multiplies and accumulations per second MMAPC).

27.4.3. Нейроприскорювачі на базі каскадного з'єднання сигнальних процесорів

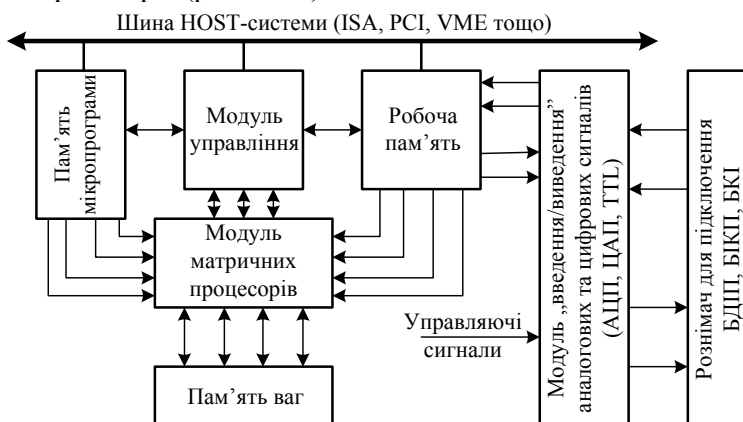
Такі НО уявляють собою мультипроцесорні системи з можливістю паралельного опрацювання, що дозволяє реалізовувати на їх основі нейрообчислювальні системи, в структурі яких можна виокремити дві основні частини:

- управляючу Host-EOM, реалізовану на основі звичайної обчислювальної системи з CISC або RISC мікропроцесорами;
- віртуальний (або зовнішній) апаратний пристрій, який підключається до Host-EOM завдяки внутрішніх (зовнішніх) системних інтерфейсів виконує основні обчислювальні операції.

Зупинимось на особливостях апаратної реалізації НО з можливістю паралельного опрацювання.

В основі побудови НО даного типу лежить використання сигнальних процесорів, об'єднаних між собою згідно визначеної архітектури, яка забезпечує паралельність виконання обчислювальних операцій. Як правило, такі НО будуються на основі гнучкої модуль-

ної архітектури, яка забезпечує простоту конфігурації системи та на-рощуваність обчислювальної потужності шляхом збільшення кілько-сті процесорних модулів або застосування більш продуктивних сиг-нальних процесорів (рис. 27.4.).



**Рис. 27.4. Узагальнена функціональна
схема віртуального нейрообчислювача**

Нейрообчислювачі даного типу реалізуються в основному на базі модулів стандартів ISA, PCI, VME. Основними їх функціональними елементами є модуль матричних сигнальних процесорів (МСП), робоча пам'ять, пам'ять програм, модуль забезпечення введення/виведення сигналів (містить АЦП, ЦАП та TTL лінії), а також модуль управління, який може бути реалізований на основі спеціалізованого управляючого сигнального процесора (УП), на основі ПЛІС або мати розподілену структуру, за якою функції загального управління розподілені між МСП.

Для побудови НО даного типу найбільш перспективним є використання сигнальних процесорів з рухомою точкою ADSP2106х, TMS320C4х, DSP96002 та ін.

Одним з найпотужніших сигнальних процесорів, із застосуванням якого можна значно підвищити продуктивність НО, вважається процесор TMS320C80 фірми Texas Instruments.

27.5. Приклади реалізації нейрообчислювачів

Нейрообчислювач NEURO TURBO. Прикладом реалізації НО на DSP фірми Моторолла є нейрообчислювач NEURO TURBO фірми Fujitsu. Він реалізований на основі 4-х зв'язаних кільцем 24-розрядних DSP с рухомою точкою MB86220 (основні параметри:

внутрішня точність 30 розрядів, машинний цикл 150 нс., пам'ять програм – 25К слів x2 (внутрішня), 64К слів x4 (зовнішня), технологія виготовлення КМОП 1,2 мкм). Активаційна функція нейронів обмежується в діапазоні від 0 до 1, а можливі значення входів не перевищують 16 розрядів, що обумовлює достатню точність при 24-х розрядній архітектурі. Побудування НК на основі кільцевої структури об'єднання DSP дозволяє знизити апаратні витрати на реалізацію підсистеми централізованого арбітражу міжпроцесорної взаємодії.

Нейрокомп'ютер NEURO TURBO складається з чотирьох DSP, зв'язаних один з одним через двопортову пам'ять. Кожний з DSP може звертатись до двох модулів такої пам'яті (ємність 2К слів кожна) і до робочої пам'яті (РП) (ємністю 64К слів x4 Банку) в своєму адресному просторі. Внаслідок того, що доступ до двопортової пам'яті здійснюється випадковим чином одним з сусідніх DSP, то передавання даних між ними відбувається в асинхронному режимі. Робоча пам'ять використовується для зберігання вагових коефіцієнтів, даних та допоміжної інформації. Для успішної роботи НС необхідне отримання згорток в усіх елементарних нейронних вузлах. Кільцева структура об'єднання DSP забезпечує конвейерну архітектуру згортки, причому передавання даних по конвейеру здійснюється через ДПП. Після того, як DSP завантажує дані з однієї ДПП, він записує результати своєї роботи в суміжну ДПП, отже, кільцева архітектура паралельного опрацювання забезпечує високу швидкість операції з використанням відносно простих апаратних рішень.

Для виконання функцій загального управління використовується Host-EOM на основі звичайної обчислювальної системи. Обмін даними між нейроплатою та Host-EOM відбувається через центральний модуль ДПП. Завантаження програм в DSP здійснюється завдяки пам'яті команд для кожного DSP. Отже, його архітектура повністю відповідає паралельній розподіленій архітектурі типу MIMD. Пікова продуктивність системи 24 MFLOPS.

Для реалізації моделі НС ієрархічного типу фірмою Fujitsu створена нейроплата на основі DSP MB86232 з власною пам'яттю до 4 Мб, що дозволяє здійснювати моделювання НС, яка містить понад 1000 нейронів. Структура НС включає в себе вхідний, проміжний та вихідний рівні (найбільша кількість схованих шарів – два (обмеження по пам'яті)). Для навчання НК використовуються оригінальні фірменні алгоритми: віртуального імпедансу, скоректованого та розширеного навчання.

Кожна з розглянутих структур реалізації НС може бути промодельована на основі різних варіантів побудови мультипроцесорних

НО. Так, для НО на основі TMS320C4x при реалізації якої-небудь з архітектур (кільце, ієрархічне дерево, гіперкуб тощо) досить тільки змінити призначення комунікаційних портів, що забезпечує гнучкість та масштабованість при дослідженні та розробках нейросіткових систем всілякої архітектури.

Високопаралельні обчислювачі та нейроприскорювачі виробництва АОЗТ “Инструментальные системы” (Росія). АОЗТ “Инструментальные системы” випускає широку номенклатуру всіляких багатofункціональних мультипроцесорних плат як у виді “віртуальних” пристроїв, так і у виду крейтових модулів та повнофункціональних обчислювачів. Елементною базою високопаралельних нейроприскорювачів даної фірми є DSP фірми Analog Devices (SISD Sharc, SIMD Sharc, Static Superscalar Sharc) та DSP фірми Texas Instrumens (в основному сімействі: TMS320C4x, TMS320C6xxx). Інтерфейси розроблюваних модулів різні: ISA, PCI, CompactPCI, VME, IEEE1394, RS232, RS482, RS485 тощо. Спектр заказних виробів практично необмежений.

Найбільш яскравим представником лінійки паралельних віртуальних обчислювачів є плата ADP160PCI, яка базується на процесорі ADSP-21160 та призначена для побудови систем радіозв'язку, радіолокації, гідролокації, вібродіагностики, опрацювання зображень. Вона побудована на базі 32-розрядного 600 Mflops процесора SHARC2 компанії Analog Devices. Сигнальний процесор ADSP-21160 працює на тактовій частоті 100 MHz, має два виконавчих пристрої для виконання SIMD операцій та внутрішню пам'ять 4 Мбіта; 64-х розрядна шина процесора забезпечує одночасну вибірку двох 32-х розрядних даних для їх паралельного опрацювання в процесорі. Процесор забезпечує обчислення 1024 точкового комплексного перетворення Фур'є з бітреверсними переставленнями за 90мкс.

Плата ADP160PCI встановлюється в PCI шину персонального комп'ютера і дозволяє виконувати програми у взаємодії з пристроями АЦП та ЦАП, розміщеними на дочірній розширювальній платі ADM. При цьому є можливість завантажувати програми ADSP-21160 та дані, здійснювати скидання процесора, переглядати пам'ять та ініціювати виконання програми. Для налагодження програмного забезпечення в середовищі VisualDSP до плати ADP160PC підключається внутрішньо-схемний емулятор EZ-ICE, який дозволяє завантажувати програми, розпочинати та зупиняти виконання програми, спостерігати та змінювати стан регістрів і пам'яті, виконувати інші операції налагодження.

Плата ADP160PCI містить процесор ADSP-21160 та одне місце для установлення дочірньої плати ADM. Об'єм синхронної динамічної пам'яті складає від 32 до 128 Мбайт. Мікросхема 32-х розряд- 530

ного моста “шина PCI-локальна шина” забезпечує доступ персонального комп'ютера до HOST-порту та внутрішньої пам'яті процесора ADSP-21160 і доступ процесора ADSP-21160 до пам'яті та портів персонального комп'ютера. Плата ADP160PCI забезпечує режим майстра шини у відповідності зі специфікацією PCI версії 2.2. Численні режими синхронізації та переривань, в тому числі, і за специфікацією I2O, забезпечуються за допомогою регістрів mailbox та doorbell. Шість лінків процесора ADSP-21160 забезпечують високошвидкісний (100 Мбайт/сек) безконфліктний обмін як з процесорами SHARC-2, так і з процесорами SHARC. Тим самим забезпечується можливість побудови масштабованих багатопроцесорних систем на процесорах сімейства ADSP-21160 та ADSP-21060/62. На платі ADP160PCI всі лінки процесора ADSP-21160 виведені на рознімачі, які забезпечують об'єднання з іншими процесорними платами для нарощування обчислювальних потужностей.

Звертання до процесору ADSP-21160 здійснюється по інтерфейсу шини PCI. Через те, що інтерфейс PCI безпосередньо зв'язаний з Host-портом процесора, персональний комп'ютер отримує доступ до його внутрішніх ресурсів. Перепускна спроможність моста при записі з персонального комп'ютера у внутрішню пам'ять процесора ADSP-21160 складає 15 Мбайт/сек, при зчитуванні персональним комп'ютером з внутрішньої пам'яті процесора ADSP-21160 складає 10 Мбайт/сек. Щоб розвантажити процесори від рутинної роботи по пересиланню масивів, в мікросхемі моста вбудовані два контролери прямого доступу до пам'яті, що забезпечує пересилання даних між пам'яттю персонального комп'ютера та внутрішньою пам'яттю процесора ADSP-21160 з швидкістю до 40 Мбайт/сек.

27.6. Висновки

Робота з нейроприскорювачами та нейрокомп'ютерами дає в руки сучасного інженера-дослідника та аналітика потужний засіб проведення обчислень в реальному масштабі часу. Проте, попри вражаючі їх можливості, НО не дуже розповсюджені на ринку через високу вартість (від одиниць до десятків тисяч USD для нейроприскорювачів і від десятків до сотень тис. USD для НК), а також через специфіку опанування – більшість з них реалізовані для спецзастосувань. Реалізація нейрообчислювальних систем і спеціалізованих обчислювачів з масовим паралелізмом на базі DSP та ПЛІС є ефективним при розв'язанні задач цифрового опрацювання сигналів, відео- та аудіоданих і побудовання технічних систем управління. При реалізації НО, як правило, пріоритетно використовується гібридна схема з *реалізацією блоку матричних обчислень* на базі кластерного з'єднання DSP

процесорів, а *логіки управління* – на основі ПЛІС. В якості елементної бази матричного кластера використовуються ADSP21060 та TMS320C44, а в найближчий час їм на заміну придуть ADSP2106x та TMS320C67xx. В подальшому матричне ядро буде реалізовуватись на базі нейрочіпів (про які далі буде йти мова в даному розділі), а сигнальні процесори та ПЛІС залишаться основою побудови логіки управління, що вже явно простежується на відомих сьогодні НО, зокрема, серії Synaps (створеної компанією Siemens Nixdorf Informationssysteme (SNI) – дочірнім підприємством Siemens у співробітництві з Маннгеймським університетом).

Найбільш яскравими прикладами НК є [12]: нейрокомп'ютер Synaps 1, 2, 3 (Siemens, Німеччина); нейрокомп'ютер “Сіліконовий мозок” (створений в США по програмі “Електронний мозок” та призначений для опрацювання аерокосмічних зображень, продуктивність 80 петафлоп (80x10¹⁵ операцій за секунду, об'єм дорівнює об'єму мозку людини, споживана потужність 20 Вт.); нейрокомп'ютер Ембріон (Росія).

Контрольні завдання

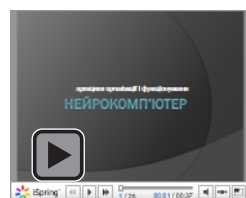
1. Порівняти характеристику продуктивності розв'язання задач в разі типової паралельної обчислювальної системи та ШНС.
2. Розкрити вміст наукового напрямку нейрокомп'ютингу.
3. Навести основні складові життєвого циклу ШНС та охарактеризувати роль нейрокомп'ютера в його обслуговуванні.
4. Проаналізувати порівняльні характеристики визначень нейрокомп'ютерів для різних наукових напрямів.
5. Дати характеристику недоліків та переваг нейрокомп'ютерів.
6. Розкрити основні архітектури обчислювальних систем.
7. Прокоментувати основні напрями розвитку обчислювальних систем з масовим паралелізмом.
8. Навести функціональні задачі нейроемуляторів, нейроприскорювачів та нейрокомп'ютерів.
9. Охарактеризувати елементну базу сучасних нейрокомп'ютерів.
10. Розкрити особливості сигнальних процесорів DSP як елементної бази нейрообчислювачів.
11. Розкрити особливості інтегральних схем ПЛІС як елементної бази нейрообчислювачів.
12. Розкрити особливості нейрочіпів як елементної бази нейрообчислювачів.
13. Прокоментувати узагальнену функціональну схему віртуального нейрообчислювача.
14. Навести приклади ефективної реалізації нейрообчислювачів.

Презентації

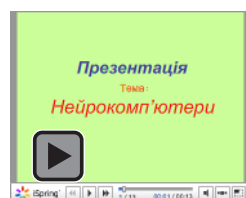
1. Нейронні сітки



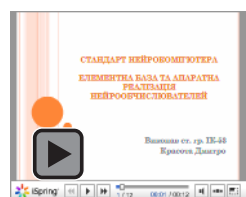
2. Нейрокомп'ютер



3. Нейрокомп'ютери



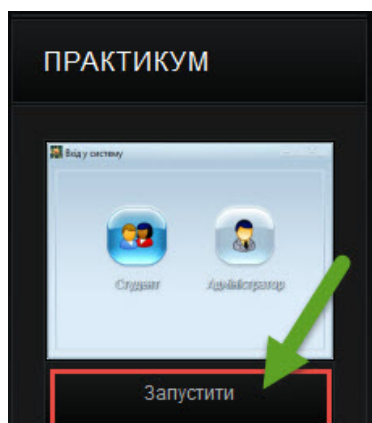
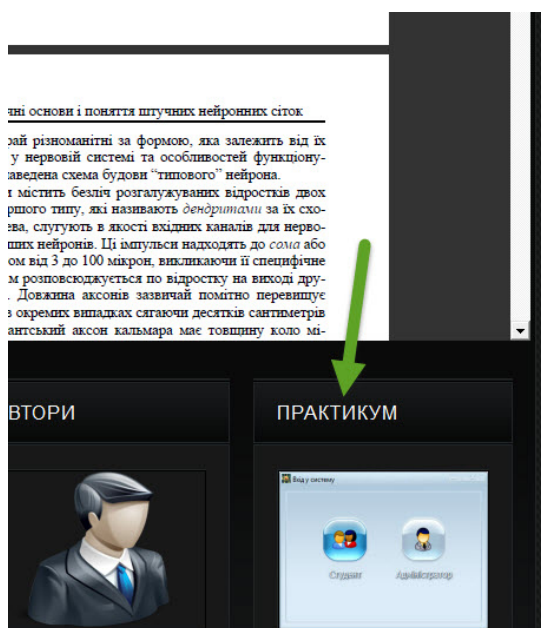
4. Стандарт нейрокомп'ютера: елементна база та апаратна реалізація нейрообчислювачів



Практичні заняття

Нейрокомп'ютери

Для запуску практичних занять на вкладці "Практикум" натисніть "Запустити"



ПІСЛЯМОВА

Приятны завершенные труды
Гомер

Вже все знаю — ще багато чого можу
журналіст-телеведучий Володимир Соловйов

*Ніколи не починай складних ходів,
якщо того ж можна досягти набагато простішими способами*
Еріх Марія Ремарк

Історично, штучні нейронні сітки за свою вже понад півсторичну історію переживали як періоди стрімких злетів і підвищеної уваги суспільства, так і змінюючі їх періоди скепсису і байдужості. В успішні часи вченим та інженерам здається, що врешті решт знайдена універсальна технологія, спроможна замінити людину у будь-яких когнітивних задачах. І тоді з'являються всілякі нові моделі НС, а між авторами — професійними вченими-математиками і спеціалістами з інформаційних технологій йдуть напружені сперечання щодо більшого або меншого ступеня біологічності запропонованих ними моделей. В той же час професіональні вчені-біологи спостерігають ці дискусії з нагадуванням, що такого в реальній природі не відбувається — і без особливого ефекту, бо нейросітковики-математики слухають біологів, як правило, тільки тоді, коли факти останніх узгоджуються з власними теоріями перших. Таким чином, з часом поступово накопичується пул задач, на яких ШНС працюють відверто погано, і ентузіазм людей на певний час згасає.

В теперішній час НС знову на вершині слави дякуючи винаходу методу переднавчання “без учителя” на основі *Обмежених Больцманівських Машин* (Restricted Boltzmann Machines, RBM), що дозволяє навчати *глибокі нейронні сітки* (тобто з екстра-великою, порядку десятків тисяч, кількістю нейронів) та успіхам глибоких НС в практичних задачах розпізнавання усної мови [136] і зображень [99] — наприклад, розпізнавання мови в Android реалізовано саме на глибоких НС. Як довго продовжиться цей період “зачарування” глибокими НС і наскільки сильно вони виправдають покладені на них сподівання — невідомо.

Між тим, паралельно всім сперечанням і тенденціям, виразно виокремлюється спільнота користувачів ШНС — інженерів-програмістів-практиків, яких цікавить прикладний аспект НС, їх спроможність навчатися на зібраних даних і розв'язувати задачі розпізнавання. З багатьма практичними задачами чудово справляються добре

пропрацьовані і відносно невеликі моделі ШНС, розглянутих в цьому підручнику – перш за все, багат шарові персептрони і сітки радіально-базисних функцій, асоціативні та каскадно-кореляційні НС тощо.

Отже, вже сьогодні ШНС використовуються у багатьох областях, але перш ніж їх можна буде застосовувати там, де мова йде про людське життя або значні матеріальні ресурси, повинні вирішуватись питання щодо надійності їх функціонування і природи самої задачі.

У загальному випадку слід враховувати дві проблеми, які постають перед користувачем при вирішенні питання щодо застосування нейротехнологій у розв'язанні прикладних задач. *Перша*, пов'язана з аналізом надійності, виникає через припущення повної безпомилковості комп'ютерів, тоді як ШНС можуть бути некоректними навіть при їх правильному функціонуванні. Але, як і люди, комп'ютери також можуть помилятися: перші – через непрофесіоналізм або неуважність, втому, а останні – через різні технічні проблеми або помилки в програмах. Тому для особливо критичних задач необхідне дублювання, тобто при розв'язанні подібного роду задач треба, щоб НС виступали *не як єдиний засіб, а як доповнюючі, попереджаючі особливі ситуації або такі, що беруть на себе керування, коли проблема не розв'язується стандартно.*

Друга проблема використання ШНС полягає в неспроможності пояснити, яким чином вони розв'язують задачу. Внутрішнє уявлення результатів навчання часто настільки складне, що його неможливо проаналізувати, за виключенням декількох найпростіших випадків.

В останній час здійснюються активні спроби об'єднання ШНС та експертних систем. У такій системі ШНС може реагувати на більшість відносно простих випадків, а всі решта передаються для розгляду експертній системі. В результаті складні випадки приймаються на більш високому рівні, можливо, зі збиранням при цьому додаткових даних або навіть із залученням експертів.

Що стосується широкоуживаного підходу використання НС на практиці, він кардинально відрізняється від звичайного детермінованого підходу “запрограмував, працює – значить, працює завжди”. Нейронні сітки за своєю природою є ймовірнісними моделями, і підхід до них повинен бути зовсім інший. На жаль, багато хто з програмістів-новачків технологій машинного навчання, взагалі, і НС, зокрема, роблять системні помилки при роботі з ними, розчаруються і кидають цю справу. Перелік типових помилок користувачів НС наводиться в роботі Артема Черnodуба “Нейронные сети, “вредные советы””, 2014, з яких виокремимо дві найсуттєвіші поради:

- якщо є можливість не використовувати НС – не використовуйте їх. Вони утруднюють розв'язання задачі у випадку,

якщо, наприклад, даних багато, вони нелінійні, зашумлені і/або надмірної розмірності;

- *складність НС повинна бути адекватною складності задачі.* Сучасні персональні комп'ютери (наприклад, Core i5, 8 GB RAM) дозволяють за прийнятний час навчати НС на вибірках об'ємом у десятки тисяч прикладів і з розмірністю вхідних даних до сотні. Для глибоких НС з великими вибірками необхідні багатопроцесорні GPU.

Сьогодні ШНС є важливим розширенням поняття *обчислення*.

Вони вже дозволили впоратись з рядом непростих проблем і обіцяють створення нових програм і пристроїв, здатних розв'язувати задачі, які поки ще під силу тільки людині. Сучасні нейрокомп'ютери використовуються в основному в програмних продуктах і тому рідко задіюють свій потенціал "паралелізму". Епоха справжніх паралельних нейрообчислень розпочнеться з появи на ринку великої кількості апаратних реалізацій – спеціалізованих нейрочипів і плат розширень, призначених для опрацювання мови, відео, статичних зображень та інших типів образної інформації [30].

Більшість надій у відношенні ШНС сьогодні пов'язують з апаратними реалізаціями, проте час їх масового виходу на ринок ще не прийшов: вони або випускаються у складі спеціалізованих пристроїв, або досить дорогі. На їх розробку витрачається значний час, за який програмні реалізації на останніх моделях комп'ютерів виявляються лише на порядок менш продуктивними, що робить використання нейропроцесорів нерентабельним. Але все це тільки питання часу – ШНС чекає той самий шлях, по якому нещодавно розвивалися комп'ютери, збільшуючи свої можливості і продуктивність, захоплюючи все нові сфери застосування по мірі виникнення нових задач і розвитку технічної бази.

На сьогодні ШНС знайшли широке застосування для розв'язання задач медичної діагностики, а також при дослідженні та прогнозуванні всіляких економічних процесів та явищ, тобто там, де відсутні або недостатньо розвинені способи формального подання досліджуваних об'єктів і явищ. В свою чергу, при розв'язанні технічних задач, таких, наприклад, як ідентифікація об'єктів керування, синтез регуляторів, діагностування несправностей, застосування ШНС досі досить обмежене. Почасти це пояснюється наявністю розвиненого математичного апарату, яким описуються процеси та явища, що відбуваються в об'єктах, а також наявністю значної кількості методів розв'язання подібних задач. І все ж в останні роки спостерігається позитивна динаміка щодо використання ШНС для розв'я-

зання задач ідентифікації та діагностування в різних галузях технічного та економіко-організаційного спрямування.

Таким чином, сьогодні найперспективнішим є використання ШНС для розв'язання подібного роду задач, в першу чергу, в галузях, де відсутні способи формального подання досліджуваних об'єктів або явищ. Для розв'язання задач ідентифікації та діагностування технічних об'єктів за допомогою ШНС необхідна розробка формальних методів вибору вихідного опису об'єктів (наприклад, ШНС), обґрунтованого вибору алгоритму навчання ШНС і методів оптимізації в процедурі навчання. В термінах ШНС задача ідентифікації технічного об'єкта відноситься до класу задач апроксимації, а задача діагностування – до групи задач класифікації та розпізнавання образів. В останні роки спостерігається тенденція переходу від програмної до програмно-апаратної реалізації нейросіткових алгоритмів, на що вже наголошувалось вище. З боку сучасних програмно-технічних засобів відсутні обмеження на складність використовуваних алгоритмів, проте для реалізації значних потенційних можливостей, які мають системи управління на основі ШНС, необхідна розробка концептуально нових підходів до побудування таких систем.

Нарешті, ще однією з перспективних областей застосування ШНС є їх використання в спеціалізованих програмних агентах-роботах, призначених для опрацювання інформації. Інтелектуальні роботи полегшуватимуть користувачам роботу з інформацією та спілкування з комп'ютером. Їх відмінна риса – прагнення якомога краще зрозуміти, що від них вимагається, за рахунок спостерегання та аналізу поведінки свого господаря, прагнучи виявити в цій поведінці деякі закономірності та своєчасно запропонувати свої послуги для виконання визначених операцій. Саме тому НС, здатні узагальнювати дані та знаходити в них закономірності, є природним компонентом подібних програмних агентів.

Повинен змінитися й інтерфейс взаємодії користувача з сіткою, який буде базуватися на інтелектуальних агентах — новому виді програмного забезпечення, який одержав назву *agentware*. Агенти будуть взаємодіяти не тільки зі своїм користувачем, але й з іншими такими ж агентами зі спеціальними сервісами (див. *розділи 24, 25*). Внаслідок цього в мережі з'явиться свого роду соціум з самонавчуваними агентами, які будуть приймати рішення від імені користувача, і поки ще важко сказати, до чого це приведе.

Проте все це, природно, справа віддаленого майбутнього. Сьогодні ж нейронні сітки впевнено продовжують проникати в наше життя.

*Людина освічена – та, хто знає,
де знайти те, чого вона не знає*
Георг Зіммель

СПИСОК ЛІТЕРАТУРИ

1. *Айзерман М. Ф.* Метод потенциальных функций в теории обучения машин / М. Ф. Айзерман, Э. М. Браверман, Л. И. Розоноэр – М.: Наука, 1970
2. *Алиев Р. А.* Интеллектуальные роботы с нечеткими базами знаний. – М.: Радио и связь, 1994. – 177 с.
3. *Арсенин В. Я., Тихонов А. Н.* Методы решения некорректно поставленных задач. 2-е издание (переработанное). – М.: Наука. – 1979. – 335 с.
4. *Архангельский В. И.* Нейронные сети в системах автоматизации / В. И. Архангельский, И. Н. Богаенко, Г. Г. Грабовский, Н. А. Рюмшин – К.: Техніка, 1999. – 364 с.
5. *Барцев С. И.* Адаптивные сети обработки информации / Барцев С. И., Охонин В. А. – Красноярск: Ин-т физики СО АН СССР, 1986. Препринт № 59Б. – 20 с.
6. *Биргер И. А.* Техническая диагностика. – М.: Машиностроение, 1978. – 240 с.
7. *Блум Ф., Лейзерсон А., Хофстедтер Л.* Мозг, разум и проведение /пер. с англ. М.: Мир. – 1988. – 248 с.
8. *Бонгард М. М.* Проблема узнавания М.: Физматгиз, 1967
9. *Борисов В. В., Круглов В. В., Федулов А. С.* Нечеткие модели и сети. – М.: Горячая линия – Телеком, 2007. – 284 с.
10. *Винер Н.* Кибернетика или управление и связь в животном и машине. – М.: Сов. радио, 1968. – 326 с.
11. *Галушкин А. И.* Синтез многослойных систем распознавания образов. – М.: Энергия. – 1974. – 367 с.
12. *Галушкин А. И.* Теория нейронных сетей Кн. 1: Учебное пособие для вузов – М.: ИПРЖР, 2000. – 416 с.
13. *Глибовець М. М., Олецкий О. В.* Штучний інтелект. – К.: Вид. дім “КМ Академія”, 2002. – 366 с.
14. *Горбань А. Н.* Нейроинформатика: кто мы, куда мы идём, как путь наш измерить? // Вычислительные технологии. – М.: Машиностроение. – 2000. – № 4. – С. 10-14
15. *Горбань А. Н.* Нейрокомпьютер, или Аналоговый ренессанс / Мир ПК, 1994, № 10. – С. 126–130
16. *Горбань А. Н.* Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей // Сибирский журнал вычислительной математики / РАН. Сиб. Отд. – Новосибирск, 1998. – Т. 1, № 1. – С. 11-24

17. Горбань А. Н., Россиев Д. А. Нейронные сети на персональном компьютере. – Новосибирск: Наука, 1996. – 276 с.
18. Гуляев В. А. Вычислительная диагностика. – К.: Наукова думка, 1992. – 232 с.
19. Денисенко В. С. Стойкость нечётких импульсных систем Такаги-Сугено: метод линейных матричных неравенств // Доп. НАН України. – 2008. – №11. – С. 66-73
20. Дюк В. А. Компьютерная психодиагностика. – СПб.: Братство, 1994. – 364 с.
21. Ежов А. А., Шумский С. А. Нейрокомпьютинг и его применение в экономике и бизнесе: Учебное пособие / А. А. Ежов, С. А Шумский. – М.: МИФИ, 1998. – 224 с.
22. Заде Л. Понятие о лингвистической переменной и ее применении к принятию приближенных решений. – М.: Мир, 1976. – 167 с.
23. Зиновьев А. Ю. Визуализация многомерных данных. – Красноярск: Изд. Красноярского государственного технического университета, 2000. – 180 с.
24. Ивахненко А. Г. Метод группового учета аргументов – конкурент метода стохастической аппроксимации // Автоматика. – 1968. – № 3. – С. 58-72.
25. Ивахненко А. Г. Самообучающиеся системы распознавания и автоматического управления. – К.: «Техніка», 1969. – 392 с.
26. Ивахненко А. Г., Мюллер И. А. Самоорганизация прогнозирующих моделей. – К.: Техніка, 1985. – 221 с.
27. Исследование операций: В 2-х т. / Под ред. Маудера Дж., Элмаграби С. – М.: Мир, 1981. – 712 с., 692 с.
28. Итоги науки и техники. Сер. Физ. и мат. модели нейронных сетей. – М.: ВИНТИ, 1990. – Т. 1. – 384 с.
29. Итоги науки и техники. – Там же. – Т. 2. – 132 с.
30. Кальченко Д. Нейронные сети: на пороге будущего // Сетевой электронный журнал “КомпьютерПрогресс”, №1, 2005
31. Круглов В. В., Борисов В. В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия – Телеком, 2001. – 382 с.
32. Куссуль Э. М. Ассоциативные нейроподобные структуры. – К.: Наук. думка, 1992. – 144 с.
33. Ливанов М. Н. Нейронные механизмы памяти // Успехи физиологических наук, 1975, Т.6. – С. 66-89
34. Минский М., Пайперт С. Персептроны. – М.: Мир. – 1971. – 261 с.
35. Миркес Е. М. Нейроинформатика: Учеб. пособие для студентов. – Красноярск: ИПЦ КГТУ, 2002. – 347 с.
36. Миркес Е. М. Нейрокомпьютер. Проект стандарта. – Новосибирск: Наука, 1999. – 337 с.

37. *Михайлюк П. П.* Програмний комплекс синтезу нейро-нечітких моделей технологічних процесів / Дисертація на здобуття наукового ступеня канд. техн. наук. – С.-Петербург, 2007. – 199 с.
38. *Нейлор К.* Как построить свою экспертную систему: Пер. с англ. – М.: Энергоатомиздат, 1991. – 286 с.
39. *Нечеткие множества в моделях управления и искусственного интеллекта* / Под ред. Д. А. Поспелова. – М. : Наука, 1986. – 311 с.
40. *Олейник А. А.* Синтез диагностических и распознающих моделей на основе гибридных нейро-нечётких технологий вычислительного интеллекта / А. А. Олейник, С. А. Субботин, Т. А. Зайко; под ред. С. А. Субботина. – Х.: Компания СМІТ, 2014. – 284 с.
41. *Олешко Д. Н.* Построение качественной обучающей выборки для прогнозирующих нейросетевых моделей / Д. Н. Олешко, В. А. Крисилов, А. А. Блажко // Штучний інтелект. – 2004. – №3. – С. 567-573
42. *Омату С., Халид М., Юсоф Р.* Нейроуправление и его приложения / Пер. с англ. – М.: ИПРЖР, 2000. – 242 с.
43. *Осовский С.* Нейронные сети для обработки информации – пер. с польского. М.: Финансы и статистика, 2002. – 344 с.
44. *Петров А. П.* О возможностях персептрона // Известия АН СССР, Техническая кибернетика. – 1964. – №6
45. *Препарата Ф., Шеймос М.* Вычислительная геометрия: Введение. – М.: Мир, 1989. – 465 с.
46. *Проценко В. С, Чаленко П. И., Сорока Р. А.* Техника программирования. – К.: Наук. думка, 1990. – 183 с.
47. *Розенблатт Ф.* Принципы нейродинамики. Персептрон и теория механизмов мозга. – М.: Мир. – 1965. – 480 с.
48. *Руденко О. Г., Бодянский Е. В.* Основы теории искусственных нейронных сетей. – Харьков: ТЕЛЕТЕХ, 2002. – 317 с.
49. *Савельев А. В.* Нейрокомпьютеры в изобретениях // «Нейрокомпьютеры: разработка, применение». – М. Радиотехника. – 2004, № 2-3, с. 33-49
50. *Савельев А. В.* Нейроускоритель. Neural Network accelerator // Свид. о регистрации программы для ЭВМ № 2003610307, заявка 2002612174, 2003
51. *Себер Дж.* Линейный регрессионный анализ / Пер. с англ. – М.: Мир, 1980. – 456 с.
52. *Слынько В.И.* Линейные матричные неравенства и устойчивость движения импульсных систем // Доп. НАН України. – 2008. – № 4. – С. 68-71

53. *Стюарт Рассел, Питер Норвіг*. Искусственный интеллект: современный подход = Artificial Intelligence: A Modern Approach. – Второе. – М.: Вильямс. – 1407 с.
54. *Тимофеев А. В.* Адаптивные робототехнические комплексы. – Л.: Машиностроение, 1988. – 332 с.
55. *Тимофеев А. В.* Методы высококачественного управления, интеллектуализации и функциональной диагностики автоматических систем. Ч. I, Ч. II. Мехатроника, автоматизация, управление. – 2003. – № 5. – 2004. – № 2
56. *Тимофеев А. В.* Мульти-агентное и интеллектуальное управление сложными робототехническими системами. Юбилейный сборник “Теоретические основы и прикладные задачи интеллектуальных информационных технологий”, посвящённый 275-летию РАН и 20-летию СПИИ РАН. – СПб.: СПИИРАН, 1999. – С. 71-81
57. *Тихонов А. Н., Арсенин В. Я.* Методы решения некорректных задач. – М.: Наука, 1980. – 223 с.
58. *Український радянський енциклопедичний словник: У 3 т./* Редкол.: А. В. Кудрицький (відп. ред.) та ін. – 2-ге вид. – К.: Головна ред. УРЕ, 1986. – Т. 1. – 752 с.
59. *Уоссермен Ф.* Нейрокомпьютерная техника. – М.: Мир, 1992. – 184 с.
60. *Ухтомский А. А.* Доминанта. – М.: Наука, 1966. – 258 с.
61. *Ухтомский А. А.* Избранные труды. – Л.: Наука, 1978. – 360 с.
62. *Фогель Л., Оуэнс М., Уолш М.* Искусственный интеллект и эволюционное моделирование. М.: Мир, 1969. – 230 с.
63. *Фукунага К.* Введение в статистическую теорию распознавания образов / Пер. с англ. – М.: Наука, 1979. – 368 с.
64. *Хайкин С.* Нейронные сети: полный курс. Вильямс, 2006. – 1104 с.
65. *Хьюбел Д.* Глаз, мозг, зрение / Пер. с англ. – М.: Мир, 1990. – 239 с.
66. *Чеканов С. Г.* Интеллектуальные программные исполнительные устройства (агенты) в системах связи // Информационные технологии, № 4, 2001. – С. 6-11
67. *Чеканов Г. П., Чеканов С. Г.* Применение технологии многоагентных систем для решения интеллектуальной поддержки принятия решения (ИППР) // Сетевой электронный журнал “Системотехника”, №1, 2003
68. *Чернодуб А. Н.* Обзор методов нейроруправления / А. Н. Чернодуб, Д. А. Дзюба // Проблемы программирования. – 2011. – № 2. – С. 79-94
69. *Экспертные системы: Принцип работы и примеры:* Пер. с яп. / Под ред. Р. Форсайта. – М.: Радио и связь, 1987. – 185 с.

70. Ямпольский Л. С. Автоматизированные системы технологической подготовки робототехнического производства / Л. С. Ямпольский, О. М. Калинин, М. М. Ткач. – К.: Вища шк., 1987. – 271 с.
71. Ямпольський Л. С. Аналітичний підхід до вибору топології нейросіток при розв'язанні прикладних задач // В зб.: Адаптивні системи автоматичного управління. – Дніпропетровськ: Системні технології, № 20 (40). – 2012. – С. 159-179
72. Ямпольський Л. С. Гнучкі комп'ютеризовані системи: проектування, моделювання і управління: комплекс підруч. для студ. вищ. навч. закл. / Л. С. Ямпольський, П. П. Мельничук, Б. Б. Самотокін, М. М. Поліщук, М. М. Ткач, К. Б. Остапченко, О. І. Лісовиченко. – Житомир: ЖДТУ, 2005. – 680 с.
73. Ямпольський Л. С. Гнучкі комп'ютерно-інтегровані системи: планування, моделювання, верифікація, управління: комплекс підруч. в 2-х книгах з мультимедійними комп'ютерно-інтегрованими засобами ДВН для студ. вищ. навч. закл. / Л. С. Ямпольський, П. П. Мельничук, К. Б. Остапченко, О. І. Лісовиченко. – Житомир: ЖДТУ, 2010. – 786 с.
74. Ямпольский Л. С. Мультиагентная реализация выбора топологии нейросетей для моделирования прикладных задач // Л. С. Ямпольский, Е. С. Пуховский, О. И. Лисовиченко. – Софія: Технічески университет, Брой №20, октомври 2013. – Р. 72-95
75. Ямпольський Л. С. Мультиагентна реалізація ітераційного синтезу штучних нейросіток в умовах нечіткої метаідентифікації // Технологічні комплекси. – Луцьк: ЛНТУ.– № 2(8). – 2013. – С. 27-31
76. Ямпольський Л. С. Нечітка ітераційна метаідентифікація штучних нейросіток в мультиагентному середовищі // Вісник кіровоградського національного технічного університету — Кіровоград: КНТУ. – №26 – 2013. – С. 207-218
77. Ямпольський Л. С. Обґрунтування вибору топологій нейросіток в прикладних задачах // В зб.: Адаптивні системи автоматичного управління. – Дніпропетровськ: Системні технології, № 20 (40). – 2012. – С. 159-179
78. Ямпольський Л. С. Системи штучного інтелекту в плануванні, моделюванні та управлінні: підруч. для студ. вищ. навч. закл. / Л. С. Ямпольський, Б. П. Ткач, О. І. Лісовиченко. – К.: ДП «Вид. Дім «Персонал», 2011. – 544 с.
79. Aarts E., Korst J. Simulated Annealing and Boltzmann Machines. A stochastic Approach to Combinatorial Optimization and

- Neural Computing // Chichester: John Wiley and Sons. – 1989. – 272 p.
80. *Aarts F., Korst J.* Simulates Annealing and Boltzmann Machines // Chichester: John Wiley and Sons. – 1990. – 265 p.
 81. *Ackley, D. H., Hinton, G. E., and Sejnowski, T. J.* A learning algorithm for Boltzmann machines. // Cognitive Science, № 9. – 1985. – P. 147-169
 82. *Almeida L. B.* A Learning Rule for Asynchronous Perceptrons with Feedback in a Combinatorial Environment // Proc. of the First IEEE International Conference on Neural Networks, USA, SanDiego, 1987. – Vol. 2. – P. 609-618
 83. *Anderson J. A.* Two Models for Memory Organization // Mathematical Biosciences. – 1970. – 8. – P. 137-160
 84. *Barricelli N. A.* Esempi numerici di processi di evoluzione. Methodos: 1954. – P. 45-68
 85. *Bartsev S. I., Okhonin V. A.* The algorithm of dual functioning (back-propagation): general approach, versions and applications. Krasnoyarsk: Biophysics Institute SB AS USSR of, 1989. Pre-print, №107B. – 16 p.
 86. *Bellifemine F. L. / F. L. Bellifemine, G. Caire, and D. Greenwood.* – Developing Multi-Agent Systems with JADE. – Wiley, 2007
 87. *Beltratti A., Margarita S. and Terna P.* Neural Networks for Economic and Financial Modelling. – London: Int. Thomson Computer Press. – 1996
 88. *Bishop C. M.* Pattern Recognition and Machine Learning. Springer, 2006 – 738 p.
 89. *Broomhead D. S., Low D.* Multivariable Function Interpolation and Adaptive Networks // Complex System. – 1988. – 2. – P. 321-355
 90. *Carpenter G. A., Grossberg S.* ART2: Self-organization of Stable Category Recognition Codes for Analog Input Patterns // Applied Optics. – 1987. – 26. – P. 4919-4930
 91. *Carpenter G. A., Grossberg S.* ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures / Neural Networks, vol. 3, 1990. – P. 129-152
 92. *Carpenter G. A., Grossberg S.* Mammalian circadian rhythms: A neural network model // In G. A. Carpenter (Ed.): Some Mathematical Questions in Biology – Circadian Rhythms. – American Mathematical Society, Lectures on Mathematics in the Life Sciences, №19. – 1987. – P. 151-203

93. *Carpenter G. A., Grossberg S.* Pattern Recognition by Self-Organizing Neural Networks. – Cambridge: MA, MIT Press. – 1991
94. *Carpenter G. A., Grossberg S.* The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Networks // Computer. – 1988. – March. – P. 77-88
95. *Carpenter G. A., Grossberg S. and Rosen D. B.* Fuzzy ART: Fast Stable Learning and Categorization of Analog Input Patterns by an Adaptive Resonance System // Neural Networks. – 1991. – 4. – P. 759-771
96. *Chauvin Y.* A back propagation algorithm with optimal use of hidden units // In Advanced in Neural Information Processing Systems 2. Tourezky D.S. (ed), 1989. – P. 519-526
97. *Cibenko G.* Approximation by Superposition of a Sigmoidal Function // Math. Contr. Sign. Syst. – 1989. – 2. – P. 303-314
98. *Control and Dynamic Systems* / T. Leondes (ed). – London: Academic Press, 1998
99. *Ciresan D., Meier U., Masci J and Schmidhuber J.* Multi-column Deep Neural Network for Traffic Sign Classification. Neural Networks, Vol. 34, August 2012, pp. 333-338
100. *Daido H.* Lower critical dimension for populations of oscillators with randomly distributed frequencies: a renormalization-group analysis // Phys. Rev. Lett, 1988, V. 61, N2, P. 231-234
101. *Eckhorn R., Bauer R., Jordan W., Brosch M., Kruse W., Munk M. Reitboeck H.J.* Coherent oscillations: a mechanism of feature linking in the visual cortex? // Biol. Cybern., 1988, V.60., P. 121-130
102. *Elman J.L.* Finding Structure in Time // Cognitive Science. – 1990. – 14. – P. 179-211
103. *Ermentrout G. B.* Synchronization in a pool mutually coupled oscillators with random frequencies // J. Math. Biol., 1985, V. 22, P. 1-9
104. *Ermentrout G.B.* The behavior of rings of coupled oscillators // J. Math. Biology, 1985, V.23, P. 55-74
105. *Ermentrout G. B., Kopell N.* Frequency plateaus in a chain of weakly coupled oscillators. I. // SIAM J. Math. Anal., 1984, V. 15, N2, P. 215-237
106. *Ermentrout G. B., Pascal M., and Gutkin, B.* The effects of spike frequency adaptation and negative feedback on the synchronization of neuronal oscillators // Neural Comput., 2001, V.13, P. 1285-1310

107. *Fahlman S. E. and Lebiere C.* The Cascade-Correlation Learning Architecture / Carnegie Mellon Report. Nr. CMU-CS-88-162, 1990
108. *Fahlman S. E., and Lebiere C.* The Cascade-Correlation Learning Architecture / In Advances in Neural Information Processing Systems 2 // D. S. Touretzky (ed.), Morgan-Kaufmann, Los Altos CA, 1990
109. *Fahlman S. E., and Lebiere C.* The cascade-correlation learning architecture. / Tech. Rep. CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, August 1991
110. *Freeman W. J.* Simulation of chaotic EEG Patterns with a dynamic model of the olfactory system // Biol. Cybern., 1987, V.56, P. 139-150
111. *Freeman W. J., Skarda C. A.* Spatial EEG patterns, non-linear dynamics and perception: the neo-Sherringtonian view // Brain Res. Rev., 1985, V. 10, P. 147-175
112. *Fukushima K.* A hierarchical neural network model for associative memory // Biological Cybernetics 50: 1984. – P. 105-113
113. *Fukushima K.* A neural network model for selective attention in visual pattern recognition. Biological Cybernetics 55(1). – 1986. – P. 5-15
114. *Fukushima K.* Cognitron: A Self-organizing Multiplayered Neural Network // Biological Cybernetics. – 1975. – 20. – P. 121-136
115. *Fukushima K.* Cognitron: A self-organizing multilayer neural network model / NHK Technical Monograph No. 30, pp. 1–25. Available from Nippon Hoso Kyokai (Japanese Broadcasting Corp.), Technical Research Labs, Tokio, Japan. – 1981
116. *Fukushima K.* Neocognitron: A Self-organizing Neural Network for a Mechanism of Pattern Recognition Unaffected by Shift in Position // Biological Cybernetics. – 1980. – 36. – P. 193-202
117. *Fukushima K., Miyake S.* Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern recognition 15(6). – 1982. – P. 455–469
118. *Fukushima K., Miyake S., Takayuki I.* Neocognitron: A neural network model for a mechanism of visual pattern recognition. IEEE Transaction on Systems, Man and Cybernetics SMC–13(5). – 1983. – P. 826–834
119. *Gorban A. N., Kegl B., Wunsh D., Zinovyev A. Y.* (Eds). Principal Manifolds for Data Visualization and Dimension Reduction: Series: Lecture Notes in Computational Science and Engineering 58, Springer, Berlin – Heidelberg – New York, 2007, XXIV. – 340 p.

120. *Gray C., Konig P., Engel A.K., Singer W.* Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties // *Nature*, 1989, V.338, P. 334-337
121. *Grossberg S.* Classical and instrumental learning by neural networks / *Progress in theoretical biology*, vol. 3. – New York: Academic Press. – 1974
122. *Grossberg S.* Competitive Learning: From Interactive Activation do Adaptive Resonance // *Cognitive Science*. – 1987. – 11. – P. 23-63
123. *Grossberg S.* Embedding Fields: A Theory of Learning with Physiological Implications // *Cognitive Science*. – 1987. – 11. – P. 23-63
124. *Grossberg S.* Nonlinear Neural Networks: Principles, Mechanism and Architectures // *Neural Networks*, 1988. – V.1. – № 1. – P. 17-62
125. *Grossberg S.* Some networks that can learn, remember and reproduce any number of complicated space-time patterns / *Journal of Mathematics and Mechanics*, 19:53-91. – 1969
126. *Grossberg S.* Studies of mind and brain. – Boston: Reidel Press, 1982. – P. 65-156
127. *Haken H., Fuchs A., Banzhatt W.* Mustererkennung durch synergetische Computer. Teil 1. Und 2 // *Design and Elektr.*, 1989
128. *Hanson S. J., Pratt L. Y.* Comparing Biases for Minimal Network Construction with Back-Propagation // In *Advanced in Neural Information Processing Systems 1*. Tourezky D. S. (ed), 1989. – P. 177-185
129. *Harp S. and Samad T.* Genetic Optimization of Neural Networks Architectures for Electric Utility Applications / Final Report. Electric Power Research Institute, Research Project № 8016-04, Palo Alto, CA. March. – 1994
130. *Hassibi B., Stork D. G.* Second Order Derivatives for Network Pruning: Optimal Brain Surgeon / Hansen S.J., Cowen J.D. In *Proceedings of the Neural Processing Systems 5*. Giles C.L. (ed). – Morgan Kaufmann, 1993
131. *Hebb D. O.* The Organization of Behavior // A Neuro psychological Theory. – New York: Wiley Publications, 1949. – 430 p.
132. *Hecht-Nielsen R.* Counterpropagation networks / In *Proceedings of the IEEE First International Conference on Neural Networks*, eds. M. Caudill and C. Butler, vol. 2, pp. 19-32. San Diego, CA: SOS Printing. – 1987

133. *Hecht-Nielsen R.* Theory of the Backpropagation Neural Network / Proc. of Int. Joint. Conf. on Neural Networks. – Washington: D. C., 1989. – 1. – P. 593-606
134. *Herts J., Krogh A., Palmer R.* Wstep do teorii obliczen neuronowych. Wyd. II. – Warsaw: WNT, 1995
135. *Hinton G. E.* Connectionist Learning Procedures // Artificial Intelligence. – 1989. – 40. – P. 185-234
136. *Hinton G., Deng L., Yu D., Dahl G., Mohamed A., Jaitly N., Senior A., Vanhoucke V., Nguyen P., Sainath T. and Kingsbury B.* Deep Neural Networks for Acoustic Modelling in Speech Recognition, IEEE Signal Processing Magazine, Vol. 29, No. 6, 2012. – P. 82 – 97
137. *Hinton G. E., Rumelhart D. E., Williams R. J.* Learning Representations by Backpropagation Errors // Nature 323. – 1986. – P. 533-536
138. *Hinton G. E., Sejnowski T. J.* Analyzing Cooperative Computation / Proc. of the First Annual Conf. on the Cognitive Sci. Society. – Rochester, NY, 1983. – P. 488-553
139. *Hinton G. E., Sejnowski T. J.* Learning and relearning in Boltzmann machines / In Rumelhart D. E. and McClelland J. L., editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations, MIT Press, Cambridge, MA. – 1986
140. *Hinton G. E., Sejnowski T. J.* Optimal perceptual inference / Proc. of the IEEE Computer Society conference of Computer Vision and Pattern Recognition. – Washington DC. – 1983. – P. 448-453
141. *Hirai Y.* VLSI Neural Network System. – Phan. – 1995
142. *Holland J.L.* Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology // Control and Artificial Systems. – The University of Michigan Press, Ann Arbor. – 1975
143. *Hopfield J. J.* Neural Networks and Physical Systems with Emergent Collective Computational Abilities // Proc. of the National Academy of Science. – 1982. – 79. – P. 2554-2558
144. *Hopfield J. J.* Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons // Proc. of the National Academy of Science. – 1982. – 81. – P. 3088-3092
145. *Hopfield J. J. and Tank D. W.* Neural Computation of Decisions in Optimization Problems // Biol. Cybernetics. – 1995. – № 52 – p. 141-152

146. *Hornik K., Stinchcombe M., White H.* Multilayer Feedforward Networks are Universal Approximators // *Neural Networks*. – 1989. – 2. – P.359-366
147. *Hubel D. H., Wiesel. T. N.* Functional architecture of macaque monkey visual cortex // *Proceedings of the Royal Society, London. Ser. B* 198. – 1977. – P. 1-59
148. *Jordan M. I.* Attractor Dynamics and Parallelism in a Connectionist Sequential Machine // *Proc. of the Eight Annual Conference of the Cognitive Science Society*, Erlbaum, Hillsdale NJ, 1986. – P. 531-546
149. *Jordan M. I.* Serial order: A parallel distributed processing approach // *Institute for Cognitive Science Report 8604*. – University of California, San Diego. – 1986
150. *Kinnebrock W.* Neuronale Netze // *R. Oldenbourg Verlag München-Wien 2. Verbesserte Auflage*. – 1944. – 176 s.
151. *Kohonen T.* Associative Memory: A System Theoretic Approach. – Berlin: Springer, 1977. – 176 p.
152. *Kohonen T.* Improved versions of learning vector quantization / *IEEE International Joint Conference on Neural Networks*. – San Diego, 1990, vol. 1. – P. 545-550
153. *Kohonen T.* Self-organization and associative memory / *Series in Information Sciences*, vol. 8. Berlin: Springer Verlag. – 1984
154. *Kohonen T.* Self-Organized Formation of Topologically Correct Feature Maps // *Biological Cybernetics*. – 1982. – 43. – P. 59-69
155. *Kopell N., Ermentrout G.B.* Phase transitions and other phenomena in chains of coupled oscillators // *SIAM J. Appl. Math.*, 1990, V.50, N4, P. 1014-1052
156. *Kopell N., Zhang W., Ermentrout G. B.* Multiple coupling in chains of oscillators // *SIAM J. Math. Anal.*, 1990, V. 21, N4, P. 935-953
157. *Kosko B.* Adaptive Bidirectional Associative Memories // *Appl. Optics*. – 1987. – 26. – № 33. – P. 4947-4960
158. *Kosko B.* Bidirectional Associative Memories // *IEEE Trans. On Syst. Man and Cybern.* – 1988. – 18. – P. 49-60
159. *Koza J. P.* Genetic Programming: On the Programming of Computers by Means of Natural Selection. – Cambridge, MA, MIT Press, 1992
160. *Kuramoto Y.* Chemical oscillations, waves and turbulence. – Springer Series in Synergetic, Springer-Verlag, 1984, V. 19
161. *Lang K. J., Waibel A. H. and Hinton G. E.* A Time-Delay Neural Network Architecture for Isolated Word Recognition // *Neural Network*. – 1990. – 3. – № 1. – P. 23-43

162. *LeCun Y.* Efficient BackProp // *Y. LeCun, L. Bottou, G. Orr, K. Muller* // In Orr G. and Muller K. (Eds), *Neural Networks: Tricks of the trade*, Springer, 1998
163. *LeCun Y. and Bengio Y.* Convolutional Networks for Images, Speech, and Time-Series // In Arbib M. A. (Eds), *The Handbook of Brain Theory and Neural Networks*: MIT Press, 1995
164. *LeCun Y., Denker J. S. and Solla S. A.* Optimal Brain Damage / In *Proceedings of the Neural Processing Systems 2*. Touretzky D. S. (Ed). – Morgan Kaufmann, 1990. – P. 598-605
165. *Li Q., Tufts D.* Synthesizing neural networks by sequential addition of hidden modes // *IEEE Proc. ICNN, Orlando, 1994*. – P. 708-713
166. *Lippman R. P.* An Introduction to Computing with Neural Nets // *IEEE ASSP Magazine*. – 1987. – № 4. – P. 4-22
167. *Low D.* Adaptive Radial Basis Function Nonlinearities and the Problem of Generalization / *Proc. of IEE Int. Conf. of Artificial Neural Networks*, London, UK. – 1989. – P. 171-175
168. *Masters T.* *Neural, Novel and Hybrid*. Algorithms for Time Series Prediction. – N.Y.: John Willey and Sons, Inc. – 1995
169. *Mc Culloch W. S. and Pitts W.* A Logical Calculus of the Ideas Immanent in Nervous Activity // *Bulletin of Mathematical Biophysics*. – 1943. – 5. – P. 115-133
170. *Minai A. A. and Williams R. D.* Back-propagation heuristics: a study of the extended delta-bar-delta algorithm // *Proc. Int. Joint Conf. on Neural Networks*. Washington, DC – 1990. – Vol.1. – P. 676-679
171. *Minsky M. L., and Papert S. A.* *Perceptrons* // Cambridge (Mass): MIT press, 1969. – 321 p.
172. *Minsky M. L., Papert S. A.* *Perceptrons*. An Introduction to Computational Geometry. Expander Edition // Cambridge (Mass): MIT Press, 1988
173. *Mozer M. C., Smolensky P.* Skeletonization: a Technique for Trimming the Fat from a Network Via Relevance Assessment // In *Advanced in Neural Information Processing Systems 1*. Touretzky D. S. (Ed), 1989. – P. 107-105
174. *Nelles O., Ernst S. and Isermann R.* Neuronale Netze zur Identifikation nichtlinearer dynamischer Systeme: Ein Überblick // *Automatisierungstechnik*. – 1997. – 45. – № 6. – S. 251-262
175. *Niebur E., Kammen D. M., Koch C, Rudennan D., Shuster H. G.* Phase-coupling in two-dimensional networks of interacting oscillators // In: *Advances in Neural Network Information Processing Systems 3* (D. S. Touretzky, R. Lippman, eds), Morgan Kaufmann Publishers Inc., San Mateo, CA, 1991

176. *Omata S., Yamaguchi Y, Shimizu H.* Entrainment among coupled limit cycle oscillators with frustration // *Physical D*, 1988, V. 31, P. 397-408
177. *Parker, D.* Learning-logic. (Tech. Rep. TR-47). Cambridge, MA: MIT Sloan School of Management, 1985
178. *Parzen E.* On Estimation of a Probability Density Function and Mode // *Ann. Math. Stat.* – 1962. – 33. – P. 1065-1076
179. *Patterson D.* Artificial Neural Networks: Theory and Application. – Singapore: Prentice Hall Inc., 1996
180. *Pineda F. J.* Dynamic and Architectures for Neural Computation // *Journal of Complexity.* – 1988. – 4. – P 216-245
181. *Pineda F. J.* Recurrent Backpropagation and the Dynamic Approach to Adaptive Neural Computation // *Neural Computation.* – 1989. – 1. – P. 161-172
182. *Pham D. T. and Liu X.* Modelling and Prediction using GMDH Networks of Adalines with Nonlinear Preprocessors // *Intern. Journal System Science.* 1994. – 25. – № 11. – P. 1743-1759
183. *Pham D. T. and Liu X.* Neural Networks for Identification, Prediction and Control. – London: Springer Verlag. – 1997
184. *Powell V. J. D.* Radial Basis Functions for Multivariable Interpolation: A review / *Proc. of IMA Conf. on Algorithms for the Approximation of Functions and Data*, Shrivenham, UK. – 1985. – P. 143-167
185. *Robert A. Jacobs.* Increased rates of convergence through learning rate adaptation // *Neural Networks.* – 1988. – Vol. 1. – P. 295-307
186. *Rojas R.* Theorie der neuronalen Netze: eine systematische Einführung. – Berlin, Springer, 1996. – 446 s.
187. *Rosenblatt F.* The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain // *Psychological Review.* – 1958. – 65. – P. 386-408
188. *Rumelhart D. E., Hilton G. E. and Williams R. J.* Learning Internal Representations by Error Propagation / In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* D. E. Rumelhart, J. L. Mc Clelland (Eds). – Cambridge: MitPress, 1986. – Vol. 1. – Chapt. 8. – P. 318-364
189. *Sejnowski T. J. and Rosenberg Ch.* NETTalk: A Parallel Network that Learns to Read Aloud. John Hopkins University Electrical Engineering and Computer Science Technical Report JHU/EECS-86/01. – 1986. – 32 p.

190. *Shoham Yoav*. Multiagent systems: Algorithmic, Game-Theoretic, and Logical Foundations = Artificial Intelligence: A Modern Approach. – Cambridge University Press, 2009. – 504 c.
191. *Shuster H. G., Wagner P.* Mutual entrainment of two limit cycle oscillators with time delayed coupling // *Progr. Theor. Phys.*, 1989, V. 81, N 5, P. 939-945
192. *Specht D. F.* A General Regression Neural Network // *IEEE Trans. on Neural Networks*. – 1991. – 2. – №6. – P. 568-576
193. *Specht D. F.* Probabilistic Neural Networks for Classification, Mapping or Associative Memory / *Proc. of the IEEE Int. Conf. on Neural Networks, SanDiego*. – 1988. – Vol. 1 – P. 525-532
194. *Strogatz S. H., Mirollo R. E.* Collective synchronization in lattices of non-linear oscillators with randomness / / *J. Phys. A: Math. Gen.*, 1988, V. 21, P. 1699-1705
195. *Takagi T., Sugeno M.* Fuzzy identification of systems and its applications to modelling and control // *IEEE Trans. Syst. Man. Cybern.* – 1985. – No 15. – P. 116-132
196. *Tanaka K.* Advanced Fuzzy control. – Japan: Kyoritsu Pub., 1994. – 223 p.
197. *Tanaka K., Sugeno M.* Stability analysis and design of fuzzy control systems // *Fuzzy Sets and Systems*. – 1992. – No 45. – P. 135-156
198. *Thierens D., and Goldberg D. E.* Mixing in Genetic Algorithms // *Proceedings of the Fifth International Conference on Genetic Algorithms ICGA-93*, In Forrest S. (Ed) Morgan Kaufmann, San Mateo, California. – 1993. – P. 38-45
199. *Timofeev A. V., Azaletskiy P. S., Myshkov P. S., Kesheng Wang.* Neural Network System for Knowledge Discovery in Distributed Heterogeneous Data. – *Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management*. Vol. 207, 2006. – P. 144-151
200. *Voegtlin T.* Neural Networks and Self Reference. – 2002. – P. 979-991
201. *Waibel A., Hanazawa T., Hinton G., Shikano K. and Lang K. J.* Phoneme Recognition using Time-Delay Neural Network // *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1989. – 37. – № 3. – P. 328-339
202. *Werbos P. J.* Beyond regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph. D thesis. – Cambridge, MA, Harvard University, 1974
203. *Whitley D., Dominic S. and Das R.* Genetic Reinforcement Learning with Multilayer Neural Networks / In Belew and Booker. – 1991. – P. 562-570

204. *Widrow B. and Hoff M. E.* Adaptive Switching Circuits / IRE WESCON Convention Record – New York, IRE, 1960. – P. 96-104
205. *Widrow B., Lehr M.* 30 years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation // Proc. IEEE. – 1990. – 78. – № 9. – P. 1415-1442
206. *Widrow B., Winter R.* Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition // IEEE Computer. – 1988. – 21. – № 3. – P. 25-39
207. *Wilson M. A., Bower J. M.* A computer simulation of olfactory cortex with functional implications for storage and retrieval of olfactory information // In: Neural Informat. Proc. Syst. (D. Anderson, ed.), AIP Press, 1988. – P. 114-126
208. *Wilson H. R., Cowan J. D.* A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue // Kybernetik 13, 1973, P. 55- 80
209. *Woo M. A. C., Peppard L. E.* System-level modelling in VHDL // Microelectronics J. – 1992. – 23, №3. – P. 223 – 230
210. *Yamaguchi Y, Shimizu H.* The theory of self-synchronization in the presence of native frequency distribution and external noises // Physical D, 1984, P. 212-226
211. *Yann LeCun, Denker J. S., Solla S., Howard R. E. and Jackel L. D.* Optimal Brain Damage, in Touretzky, David (Eds), Advances in Neural Information Processing Systems 2 (NIPS*89), Morgan Kaufman, Denver, CO, 1990
212. *Yao Y., Freeman W. J., Burke B., Yang Q.* Pattern recognition by a distributed neural network: an industrial application // Neural Networks, 1991, V. 4, P. 103-121
213. *Yoh-han Pao.* Adaptive Pattern Recognition and Neural Networks.– Reading Massachusetts: Addison-Wesley.– 1989.– 309 p
214. *Zadeh L.* Fuzzy Sets // Information and Control, №8, 1965. – P. 338-353
215. *Zadeh L. et ol.* Fuzzy Logic for the management of uncertainty // John Willey & Sons, New York, 1992
216. *Zbikovsky R., Hant K. J.* Neural Adaptive Control- Berlin: Springer-Verlag. – 1996
217. *Zell A.* Simulation neuronaler Netze. – Munchen: R. Oldenburg Verlag. – 2000
218. *Zirilli J.* Financial Prediction Using Neural Network.- London: Int. Thomson Computer Press. – 1997

ПРЕДМЕТНИЙ ПОКАЖЧИК

*Не бійтесь заглядати у словник:
Це тишиний ряд, а не сумне провалля*
Максим Рильський

А

Автоасоціація 6.1, 6.3

Автомат Мілі 16.1

–“– неперервний В.1

Автономність агента 24.3

Агент 24.1, 26.2

–“– з функціями метаідентифікації 26.2

–“– зі зміною станів 24.3

–“–“– станами 24.3

–“– інтелектуалізований 24.3

–“– програмний 24.2

–“– простий 24.3

–“– реактивний 24.3

–“– самонавчаний 24.2

–“– функціонально-спеціалізований інтелектуалізований 26.1

Адаліна В.1, 1.1

Адаптація 16.1

Аксіома монотонності 21.10

Аксон 1.1

Активність агента 24.3

Алгоритм генетичний В.1, 15.2, 15.3, 16.1, 16.2

–“– градієнтний 4.1

–“– групування пікового 22.8

–“–“– різницевого 22.8

–“– дії агента 24.3

–“–“– функціонально-спеціалізованого агента 26.1

–“– “жадібний” 17.3

–“– зворотного поширення похибки В.1, 4.5

–“– каскадної кореляції 17.1, 17.3

–“– класифікації в машині Коші 9.6

–“– кластеризації ітеративний 10.6

–“– корекції ваг 23.4.2

–“– Левенберга-Марквардта 14.4

–“– МГУА 16.3, 16.4

–“– навчання асоціативної пам'яті 6.4

–“–“– когнітрона 13.2

–“–“– машини Коші 9.6

–“–“– нейрона В.1

–“–“– нечіткої нейросітки гібридний 22.7

–“–“– персептрона одношарового 4.4

–“–“– рекурсивної нейросітки 12.9

–“– нечіткого виведення

Mamdani 21.5

–“–“–“– модифікований 21.5

–“–“–“– Sugeno 21.5

–“–“–“– Tsukamoto 21.5

–“– нечіткої самоорганізації C-means 22.8

–“– оптимізації QuickProp 17.3

–“– “послідовного лідера” 18.2

–“– редукції 17.1

–“– самоорганізації моделей багаторядний 16.3

–“–“–“– однорядний 16.3

–“– скелетонізації 23.4.1

–“– стохастичний 4.1

–“– тренування нечіткої нейронної сітки 22.5

–“– функціонування сітки АРТ1 18.2

–“–“–“– Больтцмана 9.5

–“–“–“– Кохонена 7.4

–“–“–“– Хопфілда 6.7

Аналіз регресійний 11.4

Архітектура агента зовнішня 24.3

–“– логіки агентів зі станами 24.3

–“– нейронної сітки 1.1, 2.1

–“–“–“– АРТ МАП 18.5

–“–“–“– АРТ узагальнена 18.1

–“–“–“– динамічної рекурсивної повнозв’язної 12.5

–“–“–“– згортаючої 14.3

–“–“–“– радіально-базисних функцій 10.2

–“– нечіткої нейронної сітки 22.7

–“– обчислювальна специфічна 26.6

–“– реакції агента 24.3

–“– синергетичного комп’ютера мережева 20.2

Асоціативність 21.3, 21.5

Асоціатор 20.2

–“– лінійний 6.3

Асоціація 6.1

–“– безпосередня 6.1

–“– непряма 6.1

Атрактор динамічний 6.7

–“– статичний 6.7

Б

База Больтцман-машини 9.3

Бістабільність сприйняття 20.3

Біфуркація механічна 19.3

–“– теплова 19.3

–“– типу Ландау 19.3

Блок локальної пам’яті 1.2

–“– нелінійного перетворення 1.2

–“– підсумовування 1.2

В

Вектор вхідний 4.1

–“– вхідних образів 1.1

–“– вхідних сигналів 1.1

–“– навчальний 4.1

–“– навчаючий 3.2

–“– станів 20.1

–“– цільовий 4.1

Вербалізація *Передмова*, 23.1

–“– штучної нейронної сітки 3.9

Версія алгоритму QuickProp спрощена 17.3

Вибір початкових значень вагових векторів 8.3.1

–“– центрів радіусів в нейросітках радіально- та гіпербазисних функцій 10.6

Вибірка навчаюча 3.7, 4.1, 4.6

–“– перевірна 4.1, 4.6

–“– тестуюча 4.1, 4.6

Виведення неточне 21.2, 21.5

–“–“– логічне 21.5

–“– нечітке узагальнене 21.5

–“–“– умовне 21.8

Визначення ймовірності частотне 21.7

Висловлювання нечітке 21.2

Висота нечіткої множини 21.3

Відбір природний 15.1

Віддаль Хеммінга 6.3, 6.8, 15.2

Відношення нечітке 21.5

Вікно Парзена 11.2

Властивість автоасоціативної пам’яті 6.7

–“– агента 24.3

–“– Байєсівського класифікатора 11.2

–“– max-min композиції 21.5

–“– навченої сітки статистична
8.3.1

–“– статистичної нейронної сіт-
ки 11.1

–“– топології нейронної сітки
2.2

Вузол гальмівний 13.2

–“– збуджуючий 13.2

–“– комплексний 13.3

–“– простий 13.3

Г

Гальмування латеральне 13.2

Ген 15.1

Генетична пам'ять 16.1

Геном 15.1

Генотип 15.1

Гессіан функції 23.3.1

Гетероасоціація 6.1

Гіпотеза Хебба 4.3

Гістерезис 20.1

Гradient стохастичний 14.4

Д

Двоспрямована асоціативна
пам'ять Б. Коско 6.9

Дельта-правило Уідроу-Хоффа
4.3, 6.4

–“– –“– –“– розширене 4.3

Дендрит 1.1

Динаміка дискретної рекурсив-
ної нейросітки 12.4

–“– осцилятора 19.2

Дистрибутивність 21.3, 21.5

Діаграма Вороного 7.5, 7.6

Дилема стабільності-
пластичності сприйняття 18.1

–“– –“– –“– пам'яті 18.1

Дослідження динаміки осциля-
торних нейросіток 19.6

Доступ до пам'яті прямий 18.2

Е

Еволюція 15.1, 16.1

Евристика 1.6

Екстраполяція В.1

Елітизм 15.2

Енергія

–“– деформування мапи 7.6

–“– нейрона штучна 6.7

–“– пружної деформації 7.6

Епоха навчання 3.7, 4.4, 7.4

Ефект безпірамідальний 14.3

Є

Ємність нейронної сітки 18.2

З

Завдання ідентифікації тополю-
гії нейросітки 26.1, 26.2

Задача дихотомії 11.3

–“– загальновимірна 15.4

–“– класифікації в нейросітці

АРТ-1 18.2

–“– мультимодальна 15.4

–“– навчання 4.1

–“– нейросіткового моделюван-
ня 3.3

–“– комівояжера 6.7, 9.5, 15.4

–“– регресії 11.4

–“– синергетики 20.1

–“– унімодальна 15.4

–“– пряма (аналізу) 25.3

–“– обернена (синтезу) 25.3

Закон булівої алгебри обчислю-
вальний 21.3

Зациклованість персептрона 5.2

Збіжність персептрона 4.4, 5.2

Зведення до чіткості 21.5, 21.6

Змінна лінгвістична 21.1, 21.3

–“– нечітка 21.3
Знання 1.6
–“– достовірні 21.2
–“– другого роду (процедурні) 1.6
–“– ненадійні 21.7
–“– недостовірні 1.6, 21.2
–“– неповні 1.6, 21.2
–“– нечіткі 1.6
–“– першого роду (декларативні) 1.6
Зображення амбівалентне 20.3

I

Ідемпотентність 21.11
–“– часткова 21.11
Ідентифікація топології нейросітки нечітка 26.2
Імітація охолодження 9.3, 9.4
Інверсія 15.2
Ініціалізація нейронної сітки 7.4
Інтелект штучний *Передмова*, 24.1, 24.6
–“– –“– децентралізований 24.2
–“– емерджентний 24.2
Інтерпретація агента зі станами графічна 24.3
–“– агентної взаємодії з середовищем 24.3
–“– алгоритму навчання каскадно-кореляційної нейросітки 17.3
–“– геометричного розділення площин 7.5
–“– гістерезисного явища 20.1
–“– за допомогою центральних функцій 10.3, 10.5
–“– –“– –“– –“– –“– та поліномів 10.4
–“– кодування генетичної інформації 15.2

–“– осциляції нейрофізіологічна 19.1
–“– поняття “критична риса” 18.1
Інформатизація 1.6

Й

Ймовірність апостеріорна 21.7
–“– події 21.7

К

Картина світу агента 24.3
Квантування безперервних величин 4.1
–“– генетичної інформації 15.2
Керування мультиагентне 24.1
Класифікація агентів 24.3
–“– методів спрощення нейросітки 22.2
–“– мультиагентних систем 24.4
–“– штучних нейронних сіток 2.2
Когнітрон *B.1*, 13.2
Код генетичний 15.1, 15.2
–“– Грея двійковий 15.2
Коефіцієнт асиметрії 23.3.1
–“– імплікації надійний 22.4
–“– кореляції 11.4, 17.3
–“– моменту 17.3
–“– температурний 9.3, 9.6
–“– швидкості навчання 7.3, 7.4
–“– штрафу 23.3.2
Коливання “критичне” 20.1
Композиція 21.5
Комп’ютер синергетичний *B.1*, 20.2
Комунікабельність агента 24.3
Комутативність 21.3
Конвергенція нейросітки Хопфілда 6.7

Коннекціонізм *Передмова*
Конорма трикутна 21.3
Константа Больцмана 9.5
Конфігурація гнучка інтелекту-
алізована мультиагентна В.2,
25.2
Концепція поділяючих ваг 14.1
Кооперативність агента 24.3
Кореляція каскадна 17.2
Критерій балансу змінних 16.3
—“— комбінований 16.3
—“— мінімуму зсуву 16.3
—“— —“— середньоквадратичної
помилки 17.3
—“— обслуговуваності 25.2
Кросинговер 15.1, 15.2, 16.1
Кросовер В.1, 15.1, 15.2, 16.1

Л

Лінія регресії 11.4
Логіка алетична 21.2
—“— ймовірнісна Байєсівська
21.7
—“— немонотонна 21.9
—“— нечітка 21.11
—“— умовчання 21.9
Локальність сприйняття 14.1

М

Мапа Кохонена 7.1, 7.3, 7.5
—“— —“— самоорганізуюча В.1,
7.1
—“— ознак 14.1
—“— пружна 7.6
—“— топологічна 7.4
Мах-min-композиція 22.4
Мах-утворення 22.4
Машина Больцмана В.1
—“— Коші В.1, 9.6
Мета вербалізації 23.1

—“— навчання 2.2, 4.1
—“— —“— штучної нейронної сіт-
ки 4.5
—“— синергетики 20.1
Метазнання 1.6
Метаідентифікація нейросітки
26.1, 26.2
Метод акредитації 8.3.1
—“— асимптотичний 19.2
—“— векторного квантування 8.5
—“— висотної дефазіфікації 21.6
—“— групового урахування аргу-
ментів В.1, 16.1, 16.3, 16.4
—“— індукції 16.2
—“— інтерполяції 8.3.1
—“— k найближчих сусідів 10.6
—“— критеріального максимуму
21.6
—“— мозкового відтворення 23.2,
23.3.1
—“— —“— руйнування 23.2, 23.3.1
—“— навчання 2.1
—“— —“— градієнтний 4.2, 17.3
—“— —“— зворотного поширення
похибки 4.5
—“— —“— стохастичний 4.2
—“— найменших квадратів 11.4,
16.1
—“— неповної математичної ін-
дукції 16.3
—“— опуклого комбінування
8.3.1
—“— перенормування 20.2
—“— першого максимуму 21.6
—“— побудови функції належно-
сті опосередкований 21.3
—“— —“— —“— —“— прямий 21.3
—“— повільного навчання 18.2
—“— повної математичної індук-
ції 16.3

- “– прискороеного латерального гальмування 13.2
- “– пружних мап 7.6
- “– редукції 2.1
- “– розв’язання оптимізаційних задач 15.4
- “– –“– –“– –“– локально-градієнтний 15.4
- “– –“– –“– –“– перебірний 15.4
- “– розщеплення 7.6
- “– самоорганізації моделей експериментальний 16.2
- “– середнього максимуму 21.6
- “– синергетики 20.1
- “– скелетонізації 23.2, 23.3.1
- “– турнірного відбору 15.2
- “– Ф-функцій 25.3
- “– центроїдний 21.5
- “– швидкого навчання 18.2
- Метрика Евклідова 7.2, 10.2
- Механізм прийняття рішень 26.6, 26.7
- Мінімум енергетичної функції 9.2
- “– –“– –“– абсолютний 9.2
- “– –“– –“– глобальний 9.2, 9.3
- “– –“– –“– локальний 9.2
- Min-операція 22.4
- Міра можливого 21.10
- “– неможливого 21.10
- “– сусідства нейронів 7.2
- Множина нечітка 21.1, 21.3
- “– недостовірне 21.2
- “– неповне 21.2
- “– нормальна 21.3
- “– субнормальна 21.3
- “– унімодальна 21.3
- “– чітка 21.1, 21.3
- Модель асоціативної пам’яті 6.3
- “– базова штучного нейрона 2.2
- “– вибору типових топологій нейросіток концептуальна 25.3
- “– вибору типових топологій нейросіток узагальнена 25.2, 26.4
- “– –“– топологій нейросіток інтерпретаційна 26.5
- “– випадкова 9.1
- “– гнучкої інтелектуалізованої мультиагентної системи узагальнена 26.1
- “– детермінована 9.1
- “– досліджуваного об’єкта або явища математична 11.4
- “– зорової системи 19.6
- “– інтелектуального нейрона 2.1
- “– нейронної сітки Хопфілда 6.7
- “– логічна поетапного синтезу нейросіток В.2, 25.2
- “– оптимальної складності В.1
- “– пам’яті та уваги 19.6
- “– реалізації повного функціонального орграфа нейросіткова 26.5
- “– рекурсивного багатошарового персептрона Джордана 12.6
- “– сігма-пі-нейрона 1.3
- “– сприйняття зображень 20.2
- “– формального нейрона 1.1
- “– функціонування узагальнено-регресійної нейросітки 11.4
- “– штучного нейрона 1.1
- “– –“– –“– Гроссберга 1.3
- “– –“– –“– Мак-каллока і Пітгса 1.2
- “– –“– –“– Фукушими 1.3
- “– –“– –“– Хопфілда 1.3
- “– штучної нейронної сітки віртуальна топологічна 25.3

Моделювання локомоторної активності 19.6

–“– еволюційне 16.1

–“– імітаційне 16.2

–“– стохастичне 11.1

Модифікація двошарового персептрона 12.1

Мультиагент 24.1

Мутація *B.1, 15.1, 15.2, 16.1, 16.2*

Н

Набір вирішних класифікаційних ознак *B.2, 2.1, 25.2, 25.3*

Навчання агентів без нагляду 24.4

–“–“– з підкріпленням 24.4

–“–“– незалежне 24.4

–“–“– під надглядом 24.4

–“–“– суспільне 24.4

–“– нейросітки 4.1

–“–“– АРТ 18.2

–“–“– без вчителя 2.2, 4.1, 4.2

–“–“– Больцмана 9.5

–“–“– з вчителем 2.2, 4.1, 4.2, 8.3.2

–“–“– зустрічного поширення 8.3

–“–“– конкурентне 4.2, 7.2

–“–“– методом зворотного поширення похибки *B.1, 2.2*

–“–“–“– зустрічного поширення *B.1*

–“–“– підкріплюване 4.2

–“–“– радіально- та гіпербазових функцій ітеративне додаткове 10.6

–“–“– стохастичне 4.2

–“–“– шляхом самоорганізації 2.2, 4.1

–“– неокогнітрона 13.3

–“–“– без вчителя 13.3

–“–“– з вчителем 13.3

–“– нечітких штучних нейросіток на основі генетичних алгоритмів 15.3

–“– стохастичне 4.2

–“– шару Гроссберга 8.3.2

–“–“– Кохонена 8.3.1

Насичення уваги 20.3

Настроювання ваг 2.1

Невизначеність лінгвістична 20.8

–“– об'єктивна 21.2

–“– суб'єктивна 21.2

Недостовірність знань 21.2

Нейробіологія 1.1

Нейробіоніка *Передмова*

Нейроемулятор 27.2, 27.4

Нейроінформатика *Передмова*

Нейрокібернетика *Передмова*

Нейрокомп'ютер *Передмова*

Нейрокомп'ютинг 27.1, 27.2

Нейроморфна інженерія *Передмова*

Нейрон 1.1

–“– гальмівний 13.2

–“– збуджуючий 13.2

–“– нечіткий 22.3

–“– передсинаптичний 13.2

–“– постсинаптичний 13.2

–“– складний 14.1

–“– схованого шару нейросітки радіально-базисних функцій 10.2

–“– штучний *Передмова, B.1*

–“–“– базовий 1.1

–“– фаззифікуючий 22.3

Нейрообчислювач 27.2, 27.3, 27.4, 27.5

–“– віртуальний 27.4.3

Нейроприскорювач 27.2, 27.4
 Нейропроцесор *Передмова*,
 27.2, 27.3, 27.5
 Нейропрограма універсальна
 сучасна 26.6
 Нейротехнологія *Передмова*
 Нейрочіп 27.3, 27.4
 Некомутативність 21.11
 Ненадлишковість архітектури
 нейросітки 25.2
 Неокогнітрон *В.1, 13.3*
 Нечіткість 21.5
 –“–“– знань 21.2
 Норма трикутна 21.3
 Нормалізація вхідних векторів
 8.3.1
 Нормування даних 3.3, 4.1
 –“–“– з використанням логіс-
 тичної функції 3.3
 –“–“– індивідуальне 3.3
 –“–“– лінійне 3.3
 Носій нечіткого числа 21.4
 –“– нечіткої множини 21.3, 21.8

О

Область зв'язків нейронів 13.2
 –“– конкуренції нейронів 13.2
 –“– рецепторна 13.2, 13.3
 –“–“– комплексного вузла
 13.3
 Об'єкт керування 25.3
 –“– сприйняття 24.3
 Обчислення варіаційне для
 розв'язання проблеми апрокси-
 мації 10.5
 –“– паралельне 24.1
 Обчислювальна нейронаука *Пе-
 редмова*
 Одноманітність 21.9
 Ознака класифікаційна виріша-
 льна *Передмова*

Оператор генетичний 15.2
 Операція максимінної компози-
 ції нечітких множин 21.8
 –“– над нечіткими множинами
 21.3
 Оптимізація наближена 15.2
 –“– топології штучних нейросі-
 ток 15.3
 Орграф об'єкта керування по-
 вний функціональний 25.3
 Особина *В.1*
 Особливість навчання згортаю-
 чої нейросітки 14.4
 Осцилятор 19.2
 –“– нейронний 19.1, 19.2
 –“– фазовий 19.1, 19.3
 –“–“– зі слабкими зв'язками
 19.4
 –“–“– з сильними локальними
 зв'язками 19.4
 Очікування зустрічне 18.1

П

Пам'ять автоасоціативна 6.6
 –“– асоціативна *В.1, 6.1*
 –“–“– двоспрямована *В.1*
 –“–“– асоціативна 6.9
 –“– генетична 16.1
 –“– гетероасоціативна *В.1, 6.5*
 –“– з адресою за змістом 6.2
 Парадигма згортаючої нейросі-
 тки 14.1
 –“– навчання Delta Bar Delta 4.5
 –“–“– за алгоритмом розши-
 рення Delta Bar Delta 4.5
 –“–“– персептрона одношаро-
 вого 4.4
 –“–“–“– багатшарового 4.4
 –“–“–“– скерованим випадковим
 пошуком 4.5
 Параметр віддалі 7.3

- “– керуючий 20.1
- “– порядку 20.1
- “– –“– рівноважний 20.1
- “– регуляризації 14.4
- “– розсіяння 10.6
- Перетворення автомасштабне 3.3
- “– інтермальне масштабне 3.3
- Перехід фазовий 19.2
- Персептрон одношаровий 5.2
- “– багатшаровий 5.3
- “– –“– Елмана 12.7
- Підмножина нечітка 21.3
- Підсистема агентно-орієнтована В.2, 25.2
- “– мультиагентна 26.1
- Поведінка агента інтелектуальна 24.3
- Показник важливості нейрона 23.4.1
- Поле нейрона рецептивне 14.1
- Поняття модельованого відпаду 9.1
- Порядок організації згортаючої нейросітки 14.2
- Послідовність дій конкретна унікальна 25.2
- “– процесу видалення ваг нейросітки 23.3.1
- “– –“– регуляризації нейросітки 23.3.1
- Постулат замкненості 21.2
- Потужність нейронної сітки 2.1
- Похибка квадратична 23.4.1
- “– конкретної топології нейросітки 4.5
- “– середньоквадратична 17.3
- Правило виведення 1.6
- “– “два з трьох” 18.1
- “– “модус-поненс” 21.11
- “– навчання Ойя 3.3
- “– –“– пружної мапи 7.6
- “– –“– сітки векторного квантування 7.5
- “– –“– Хебба 4.3, 5.2, 6.4
- “– –“– –“– неконтрольоване 6.4
- “– продукції 21.1
- “– умовчання 21.9
- Предмет синергетики 20.1
- Приклад архітектурної побудови згортаючої нейросітки 14.3
- Принцип адаптивного резонансу 18.1
- “– лінійної суперпозиції 10.3
- “– підпорядкованості 20.1
- “– побудування синергетичного комп'ютера 20.2
- “– синергетики основний 20.2
- Пристосованість 15.2
- Проблема “Виключного АБО” 5.2
- Програми-нейроімітатори 26.6
- Програмування генетичне В.1
- Проріджування нейронної сітки 3.9
- Простір топологічний 2.2
- Процедура аналізу адекватності синтезованих топологій ШНС ітераційна 26.4
- “– композиції 22.2
- “– навчання когнітрона 13.2
- “– охолодження ітераційна покровока 9.3
- “– розрахунку стохастична 9.3
- “– функціонування узагальнено-регресійної нейросітки 10.4
- Процес ітерацій детерміністський 9.3
- “– навчання нейросіток АРТ 18.2
- “– тренування нечіткої нейросітки 22.5

Р

Рандомізація ваг 7.3
 Реактивність агента 24.3
 Регресія 11.4
 Регуляризація нейросітки 23.3.1
 Редукція нейросітки з викорис-
 танням вартісної функції 23.3.3
 –“–“–“–“– штрафної функ-
 ції 23.3.2
 Режим роботи асинхронний 6.7,
 6.9
 –“–“– синхронний 6.7, 6.9
 Резонанс адаптивний 18.1
 Рельєф простору станів 9.1
 Рецептор 14.1
 Решітка нейросітки Кохонена
 7.2
 Риса критична 18.1
 Рівень в синергетиці мікроско-
 пічний 20.1
 –“–“–“– макроскопічний
 19.1
 Рівняння регресії 11.4
 Роботи інтелектуалізовані 21.11
 Розв’язок логічний композицій-
 ний 22.2, 22.4
 Розв’язання задач розподілене
 24.1, 24.2
 –“– логічне 22.4
 Розв’язувач поставленої задачі
 узагальнений 26.4
 Розпізнавання Коші 9.6
 –“– Больцмана 9.5
 –“– образів В.1

С

Самоадаптація алгоритму по-
 шуку 18.1
 Самоорганізація 20.1

Селекція 15.2, 16.1 - 16.3
 Середовище навколишнє 24.3
 –“– мультиагентне 26.1
 Сигнал фаззі-монотонний 22.4
 Силогізм 21.8
 –“– гіпотетичний 21.8
 Символ Кронекера 6.7
 Синапс 1.1
 –“– гальмуючий 13.1
 –“– збуджуючий 13.1
 Синергетика 20.1
 Синхронізація осциляторна 19.2
 Система адаптивна з нейро-
 фаззі виведенням 22.6
 –“– багатоагентна 24.1
 –“– гнучка мультиагентна інте-
 лектуалізована 26.4
 –“– децентралізована 24.2
 –“– з нечіткою логікою 22.1
 –“–“– штучним інтелектом
 22.1
 –“– мультиагентна 24.1
 –“–“– робототехнічна 24.5
 –“– нейро-фаззі 22.4
 –“– нейро-фаззі інтегрована
 22.4
 –“– орієнтуюча 18.1
 –“– розподілена 24.2
 –“– структурностійка 20.1
 –“– централізована 24.2
 Сітка нейронна 1.1
 –“–“– автоасоціативна 3.3
 –“–“–“– прямого поширення
 6.3
 –“–“– адаптивної резонансної
 теорії В.1
 –“–“– АРТ-2 18.4
 –“–“– АРТ-3 18.4
 –“–“– АРТ МАП 18.5
 –“–“– багат шарова *Передмо-
 ва, В.1, 1.4*

–“–“– Б. Коско негомогенна 6.9
 –“–“– Ванга-Менделя 22.7
 –“–“– вищого порядку 1.5
 –“–“– Ворда 5.3
 –“–“– гіпербазисних функцій
 –“–“– Джордана 12.6
 –“–“– динамічна рекурсивна В.1, 12.1
 –“–“–“–“– багатoshарова 12.1
 –“–“–“–“– дискретна В.1
 –“–“–“–“– неперервна В.1
 –“–“–“–“– повнозв'язна 12.5
 –“–“– Елмана В.1
 –“–“– з динамічними зв'язками 2.2
 –“–“– згортаюча 14.1
 –“–“– з латеральними зв'язками 1.4
 –“–“– зустрічного поширення В.1, 8.1
 –“–“– з фіксованими зв'язками 2.2
 –“–“– із зустрічним розпізнаванням без зворотних зв'язків 8.1
 –“–“– ієрархічна 1.1
 –“–“– інтелектуалізована “Синергетичний комп'ютер” 20.2
 –“–“– ймовірнісна 11.1
 –“–“– каскадно-кореляційна В.1, 17.2
 –“–“– квантування навчального вектора 7.5
 –“–“– Кохонена 7.1
 –“–“– локальна 1.1
 –“–“– локально-зв'язна 1.4
 –“–“– нечітка 22.1, 22.3
 –“–“– з самоорганізацією 22.8

–“–“– одношарова *Передмова*, 1.4
 –“–“– осциляторна 19.1
 –“–“–“– з затримками у зв'язках 19.5
 –“–“–“–“– локальними зв'язками 19.4
 –“–“–“–“– повнозв'язна 19.3
 –“–“–“– прямого поширення локально-рекурсивна 12.8
 –“–“–“–“– повнозв'язна 1.4
 –“–“–“– спрямована 1.4
 –“–“–“– радіально-базисних функцій В.1, 10.1
 –“–“–“–“–“– розширена 10.5
 –“–“–“–“–“– узагальнена 10.4
 –“–“– рекурсивна неперервна 12.3
 –“–“– самоорганізуюча В.1
 –“–“– Такагі-Сугено-Канга 22.7
 –“–“–“– узагальненої регресії 11.4
 –“–“– Хеммінга 6.8
 –“–“– Хопфілда В.1
 –“–“–“– бінарна 6.7
 –“–“– частково-рекурсивна 12.2
 –“–“–“–“– ієрархічна 12.7
 –“–“–“– штучна *Передмова*, В.1
 –“–“– складова елементарна 25.3
 Сом 1.1
 Спайк 1.1
 Спосіб доступу асоціативний 6.1
 Сприйняття локальне 14.1
 Сталість двоскерованої асоціативної пам'яті 6.9

Стан системи 20.1
 –“– стійкий 20.1
 Статистика Больтцмана 9.4
 Стиснення даних 3.3, 8.5, 10.6
 Стійкість сітки з оберненими зв'язками 6.7
 Стратегія елітизму 15.2
 Структура двоспрямованої асоціативної пам'яті Б. Коско 6.9
 –“– генетичного алгоритму 15.2
 –“– динамічної рекурсивної нейросітки 12.2
 –“– –“– –“– Елмана 12.7
 –“– інтегрованої нейро-фаззі системи 22.5
 –“– когнітрона 13.2
 –“– модифікованої рекурсивної нейросітки 12.2
 –“– нейронної сітки 2.1, 2.2
 –“– –“– –“– Б. Коско 6.9
 –“– –“– –“– Больтцмана 9.5
 –“– –“– –“– Джордана 12.6
 –“– –“– –“– Елмана узагальнена 12.7
 –“– –“– –“– –“– –“– –“– радіально-базисних функцій 10.2
 –“– –“– –“– –“– –“– –“– узагальненої 10.5
 –“– –“– –“– стохастичної інтелектуалізованої 11.3
 –“– –“– –“– узагальненої регресії 11.4
 –“– –“– –“– Хеммінга 6.8
 –“– –“– –“– Хопфілда 6.7
 –“– неокогнітрона 13.3
 –“– площин неокогнітрона 13.3
 –“– стійкої системи 20.1
 Ступінь належності елемента нечіткій множині 21.3
 Субдискретизація 14.1

Схема еволюції розв'язків 15.2
 –“– логічна поетапного синтезу нейросіток *Передмова*, 25.1
 –“– масової селекції 16.3.2
 –“– функціонування генетичного алгоритму 15.2

Т

Твердження необхідно істинне 21.10
 –“– можливо істинне 21.10
 Теорема Байєса 21.7
 –“– де Моргана 21.11
 –“– про повноту 26.7
 Теорія еволюційна 15.1
 –“– можливостей 21.10
 Технологія мультиагентна 24.1
 Тип активаційної функції 2.1
 Топологія каскадна 17.3
 –“– нейронної сітки 1.1, 2.2, 25.1
 –“– –“– –“– Больтцмана 9.5
 –“– траєкторія руху 25.3
 Траєкторія руху 25.3
 –“– –“– оптимальна 25.3, 26.5
 –“– –“– умовно оптимальна 26.5

У

Умова безпомилкової роботи сітки Хопфілда 6.7
 Умовчання нормальне 21.9
 –“– –“– закрите 21.9

Ф

Фаза функціонування нейросітки АРТ-1 18.2
 –“– порівняння 18.2
 –“– пошуку 18.2
 Фаззі-контролер 22.2
 Фаззіфікація 22.2

Факти 1.6

Фактор використання мульти-
агентної системи 24.4

Фільтрація даних 4.1

—“— образу адаптивна 18.2

Форма гіпотетичного подання
силогізму перша 21.8

—“— подання нейронної сітки
аналітична 1.4

—“— —“— —“— векторна 1.4

—“— —“— —“— —“— таблична 1.4

Формальна система знань *Пе-
редмова*, 1.6

Формула Байєса 21.7

Функціонал лінійний 23.4.1

Функціонування двоспрямова-
ної асоціативної пам'яті 6.9

—“— сітки з квантуванням навча-
льного вектора 7.5

—“— —“— зустрічного поширення
нормальне 8.2, 8.4

Функція активаційна 1.1

—“— —“— гаусівська 1.1

—“— —“— гіперболічного тангенсу
1.1

—“— —“— інверсна мультиквадра-
тична 1.1

—“— —“— нейрона 1.1

—“— —“— сигмоїдна 1.1, 2.2

—“— —“— трапецієвидна 1.1

—“— базисна 10.2, 10.3

—“— вартісна квадратична 23.3.3,
23.4.2

—“— Гамільтонова 9.2

—“— енергетична 9.2

—“— лінійно нероздільна 5.2

—“— —“— трикутна 1.1

—“— належності 21.1, 22.2

—“— нечіткої належності 21.3

—“— оцінки роботи нейронної
сітки 4.7

—“— передатна однопараметрич-
на 4.5

—“— —“— багатопараметрична 4.5

—“— потенціальна 10.2

—“— радіально-базисна *B.1*, 10.3

—“— —“— симетрична *B.1*

—“— топології штучної нейрон-
ної сітки 25.3

—“— цільова 15.2

—“— штрафна 23.3.2

Х

Хромосома 15.1, 15.2

Ц

Центр базисних функцій 10.2

Ч

Число нечітке 21.4

—“— —“— унімодальне 21.4

Ш

Шаблон критичних рис 18.1

Шар Гроссберга 8.2

—“— згортаючий 14.1

—“— контекстний 12.6

—“— Кохонена 7.1, 8.2

—“— нейронної сітки вхідний 1.4

—“— —“— —“— на виході 1.4

—“— —“— —“— схований 1.4

—“— непараметричний 22.7

—“— параметричний 22.7

—“— порівняння 18.2

—“— розпізнавання 18.2

—“— субдискретизуючий 14.1

—“— фаззифікації 22.7

Швидкість навчання 2.2, 7.5

Я

Ядро згортки 14.1

ЗМІСТ

ПЕРЕДМОВА ЗАГАЛЬНА	3
СПИСОК СКОРОЧЕНЬ	14
ПЕРЕДМОВА	17
ВСТУП	29
В.1. Хронологія розвитку нейротехнології	29
В.2. Шляхи розв'язання дилеми обґрунтованого вибору нейросіток	41
Розділ 1. БІОЛОГІЧНІ ОСНОВИ І ПОНЯТТЯ ШТУЧНИХ НЕЙРОННИХ СІТОК	44
1.1. Загальні відомості про біологічні нейросітки	44
1.2. Модель формального кібернетичного нейрона	50
1.3. Базові моделі штучних нейронів	57
1.4. Математична інтерпретація штучних нейронних сіток ...	62
1.5. Нейросітки вищого порядку	71
1.6. Штучна нейронна сітка як модель подання знань	72
Контрольні завдання	77
Розділ 2. КЛАСИФІКАЦІЯ ШТУЧНИХ НЕЙРОННИХ СІТОК	79
2.1. Формування набору вирішних класифікаційних ознак ...	79
2.2. Класифікація штучних нейронних сіток	80
Контрольні завдання	87
Розділ 3. ПІДГОТОВКА ДАНИХ ПРИ РОБОТІ З ШТУЧНИМИ НЕЙРОННИМИ СІТКАМИ	88
3.1. Етапи розв'язання задач при підготовці штучних нейронних сіток	88
3.2. Збирання даних для навчання	88
3.3. Методи попереднього опрацювання даних	89
3.4. Вибір топології штучних нейронних сіток	96
3.5. Експериментальне підбирання характеристик сітки	96
3.6. Експериментальне підбирання параметрів навчання	96
3.7. Власне навчання сітки	97
3.8. Перевірка адекватності навчання	97
3.9. Вербалізація штучних нейронних сіток	98
Контрольні завдання	99
Розділ 4. НАВЧАННЯ НЕЙРОННИХ СІТОК	100
4.1. Механізм навчання штучних нейронних сіток	100
	565

4.2. Методи навчання штучних нейронних сіток	106
4.3. Правила навчання нейронних сіток	109
4.4. Парадигми навчання персептрона	112
4.5. Метод зворотного поширення похибки	118
4.6. Загальні зауваження щодо навчання нейронних сіток	127
4.7. Функція оцінки роботи сітки	128
Контрольні завдання	128
Розділ 5. ПЕРСЕПТРОН	130
5.1. Визначення персептрона	130
5.2. Навчання персептрона	131
5.3. Багатошаровий персептрон	135
5.4. Узагальнення	138
Контрольні завдання	138
Розділ 6. АСОЦІАТИВНІ НЕЙРОННІ СІТКИ	139
6.1. Асоціативна пам'ять нейронних сіток	139
6.2. Асоціативна пам'ять і програмування	143
6.3. Моделі асоціативної пам'яті	145
6.4. Алгоритми навчання асоціативної пам'яті	148
6.5. Гетероасоціативна пам'ять	152
6.6. Автоасоціативна пам'ять	154
6.7. Модель нейронної сітки Хопфілда з автоасоціативною пам'яттю	156
6.8. Нейронна сітка Хеммінга	168
6.9. Двоспрямована асоціативна пам'ять	172
Контрольні завдання	176
Розділ 7. НЕЙРОННІ СІТКИ КОХОНЕНА	178
7.1. Загальні відомості про нейронні сітки Кохонена	178
7.2. Навчання нейронної сітки Кохонена	181
7.3. Побудування мапи Кохонена	183
7.4. Алгоритм функціонування сітки Кохонена	185
7.5. Квантування навчального вектора (Learning VectorQuantization)	186
7.6. Пружні мапи	191
Контрольні завдання	194
Розділ 8. СІТКИ ЗУСТРІЧНОГО ПОШИРЕННЯ	195
8.1. Загальні відомості	195
8.2. Нормальне функціонування сітки зустрічного поширення	197
8.3. Навчання сітки зустрічного поширення	199
8.3.1. Навчання шару Кохонена	199
8.3.2. Навчання шару Гроссберга	205

8.4. Режим нормального функціонування сітки зустрічного поширення	206
8.5. Деякі практичні застосування	207
8.6. Загальні недоліки і переваги сіток зустрічного поширення	208
Контрольні завдання	209
Розділ 9. МАШИНА БОЛЬТЦМАНА	210
9.1. Загальні відомості	210
9.2. Енергетична функція	211
9.3. Імітація охолодження	212
9.4. Розподілення Больтцмана	215
9.5. Сітки Больтцмана	218
9.6. Машина Коші	224
Контрольні завдання	228
Розділ 10. НЕЙРОСІТКИ РАДІАЛЬНО-БАЗИСНИХ ФУНКЦІЙ	229
10.1. Загальні відомості	229
10.2. Архітектура РБФ-сіток	229
10.3. Математичні основи РБФ-сіток	233
10.4. Узагальнені РБФ-сітки	235
10.5. Розширення РБФ-сіток	236
10.6. Ітеративне додаткове навчання радіально- та гіпербазисних сіток	241
Контрольні завдання	245
Розділ 11. ЙМОВІРНІСНІ НЕЙРОННІ СІТКИ	247
11.1. Ймовірна (стохастична) нейронна сітка	247
11.2. Байєсівський класифікатор	252
11.3. Структура стохастичної сітки	254
11.4. Узагальнено-регресійна нейронна сітка	257
Контрольні завдання	263
Розділ 12. ДИНАМІЧНІ РЕКУРСИВНІ НЕЙРОННІ СІТКИ	264
12.1. Узагальнення структури динамічних рекурсивних сіток	264
12.2. Частково-рекурсивні дискретні нейросітки	268
12.3. Динаміка неперервної рекурсивної нейросітки	270
12.4. Динаміка дискретної рекурсивної нейросітки	271
12.5. Повнозв'язні динамічні рекурсивні нейросітки	271
12.6. Сітка Джордана	272
12.7. Багатошаровий персептрон Дж. Елмана	276
12.8. Локально-рекурсивні сітки прямого поширення	280
12.9. Навчання рекурсивної нейросітки	280
Контрольні завдання	282

Розділ 13. КОГНІТРОН ТА НЕОКОГНІТРОН	284
13.1. Проблеми штучного відтворення процесу розпізнавання образів	284
13.2. Когнітрон	285
13.3. Неокогнітрон	293
13.4. Висновки	299
Контрольні завдання	300
Розділ 14. ЗГОРТАЮЧІ НЕЙРОННІ СІТКИ	301
14.1. Властивості згортаючої нейронної сітки	301
14.2. Принципи організації та функціонування згортаючої нейросітки	305
14.3. Архітектура згортаючої нейронної сітки	306
14.4. Особливості навчання згортаючих нейронних сіток	308
14.5. Переваги і недоліки згортаючих нейросіток	311
Контрольні завдання	311
Розділ 15. ГЕНЕТИЧНІ АЛГОРИТМИ	313
15.1. Механізм генетичної спадковості	313
15.2. Генетичний алгоритм	315
15.3. Генетичні алгоритми і штучні нейросітки	324
15.4. Особливості генетичних алгоритмів	326
Контрольні завдання	329
Розділ 16. МЕТОД ГРУПОВОГО УРАХУВАННЯ АРГУМЕНТІВ В НЕЙРОМОДЕЛЮВАННІ	330
16.1. Метод групового урахування аргументів як предтеча ідеї нейросіткового еволюційного моделювання	330
16.2. Деякі підходи до моделювання в комбінованих інтелектуальних системах	332
16.3. Особливості використання методу групового урахування аргументів	334
16.3.1. Однорядні (комбінаторні) алгоритми	336
16.3.2. Багаторядні алгоритми	338
16.3.3. Структура комбінованого алгоритму	341
16.4. Штучні нейронні сітки, засновані на МГУА	345
Контрольні завдання	347
Розділ 17. КАСКАДНО-КОРЕЛЯЦІЙНІ НЕЙРОННІ СІТКИ	348
17.1. Методи нарощування нейронної сітки	348
17.2. Каскадна кореляція	348
17.3. Навчання каскадно-кореляційних нейросіток	349
17.4. Переваги й недоліки каскадно-кореляційних нейросіток	355
Контрольні завдання	355

Розділ 18. НЕЙРОННІ СІТКИ АДАПТИВНОЇ РЕЗОНАНСНОЇ ТЕОРІЇ	356
18.1. Адаптивна резонансна теорія	356
18.2. Нейронна сітка АРТ-1	361
18.3. Навчання сітки АРТ	368
18.4. Розширення базової моделі АРТ – нейросітки АРТ-2 та АРТ-3	372
18.5. Нейронна сітка АРТ МАП	373
18.6. Суттєві недоліки сітки АРТ	375
Контрольні завдання	375
Розділ 19. ОСЦИЛЯТОРНІ НЕЙРОННІ СІТКИ	377
19.1. Нейрофізіологічна інтерпретація осциляції	377
19.2. Математичне дослідження динаміки осциляторних нейросіток	379
19.3. Повнозв'язні осциляторні нейронні сітки	382
19.4. Осциляторні нейросітки з локальними зв'язками	384
19.5. Осциляторні нейросітки з затримками у зв'язках	386
19.6. Висновки	387
Контрольні завдання	390
Розділ 20. СИНЕРГЕТИЧНИЙ КОМП'ЮТЕР	391
20.1. Означення й поняття в синергетиці	391
20.2. Синергетичний комп'ютер	396
20.3. Приклади використання синергетичного підходу	399
Контрольні завдання	402
Розділ 21. НЕДОСТОВІРНІ ТА НЕЧІТКІ ЗНАННЯ	404
21.1. Загальні положення	404
21.2. Поняття про недостовірні та нечіткі знання	406
21.3. Чіткі та нечіткі множини	408
21.4. Нечіткі числа	418
21.5. Нечітке логічне виведення	418
21.5.1. Модифіковані алгоритми нечіткого виведення	423
21.6. Методи зведення до чіткості	425
20.7. Ненадійні знання та виведення	426
21.8. Нечітка логіка в задачах робототехніки	428
21.9. Логіка умовчання	434
21.10. Теорія можливостей	435
21.11. Формалізація знань інтелектуалізованих роботів	436
Контрольні завдання	437
Розділ 22. НЕЙРО-ФАЗЗИ-ТЕХНОЛОГІЇ	438
22.1. Вихідні міркування	438

22.2. Принципи побудови і структура нейро-фаззі-систем	439
22.3. Особливості формування нейро-фаззі систем	442
22.4. Відображення системи фаззі-управління в нейросітках	444
22.5. Алгоритми тренування нейро-фаззі-систем	450
22.6. Узагальнена структура адаптивної нейро-фаззі системи	455
22.7. Архітектура нечітких нейросіток	456
22.8. Нечіткі нейросітки з самоорганізацією	461
Контрольні завдання	464
Розділ 23. МЕТОДИ СПРОЩЕННЯ НЕЙРОННИХ СІТОК	465
23.1. Мета спрощення нейронних сіток	465
23.2. Основні напрями спрощення нейронних сіток	466
23.3. Методи відсікання вагової компоненти нейросіток	467
23.3.1. Редукція нейронних сіток з урахуванням чутливості	467
23.3.2. Редукція нейронної сітки з урахуванням штрафної функції	471
23.3.3. Редукція нейросіток з використанням вартісної функції	472
23.4. Спрощення нейросіток видаленням нейронів	473
23.4.1. Відсікання нейронів з урахуванням їх важливості	474
23.4.2. Видалення нейронів з урахуванням вартісної функції	476
Контрольні завдання	477
Розділ 24. МУЛЬТИАГЕНТНІ СИСТЕМИ	479
24.1. Мультиагентні технології	479
24.2. Особливості мультиагентного підходу	481
24.3. Характеристика агентів мультиагентної системи	484
24.4. Особливості використання мультиагентних систем	491
24.5. Мультиагентні робототехнічні системи	493
24.6. Проблеми при створенні мультиагентних систем	494
Контрольні завдання	495
Розділ 25. КОНЦЕПТУАЛЬНА МОДЕЛЬ ВИБОРУ ШТУЧНИХ НЕЙРОННИХ СІТОК	496
25.1. Вимоги щодо компетенції користувача	496
25.2. Логічна модель поетапного синтезу штучних нейронних сіток	497
25.3. Узагальнена концептуальна модель процесу вибору топології штучних нейронних сіток	499
Контрольні завдання	504

Розділ 26. АГЕНТНО-ОРІЄНТОВАНА ІДЕНТИФІКАЦІЯ НЕЙРОННИХ СІТОК	505
26.1. Модель агентно-орієнтованої ідентифікації топології нейронних сіток	505
26.2. Концепція нечіткої метаідентифікації в задачах вибору нейронних сіток	506
26.3. Компоненти ГІМАК АОП та їх взаємодія при функціонуванні	508
26.4. Узагальнена модель гнучкої інтелектуалізованої мультиагентної системи вибору нейронних сіток	509
26.5. Інтерпретаційна модель вибору “розв’язувана прикладна задача – топологія штучної нейронної сітки”	511
26.6. Можливі способи застосування та реалізації вибраної топології штучних нейронних сіток	514
26.7. Узагальнення та висновки	515
Контрольні завдання	516
Розділ 27. НЕЙРОКОМП’ЮТЕРИ	517
27.1. Загальні зауваження	517
27.2. Архітектура обчислювальних систем	519
27.3. Елементна база нейрообчислювачів	521
27.4. Апаратна реалізація елементів нейрології	522
27.4.1. Нейрообчислювачі на елементній базі DSP та ПЛІС	523
27.4.2. Нейрочіпи як елементна база нейрообчислювачів	525
27.4.3. Нейроприскорювачі на базі каскадного з’єднання сигнальних процесорів	527
27.5. Приклади реалізації нейрообчислювачів	528
27.6. Висновки	531
Контрольні завдання	532
ПІСЛЯМОВА	533
СПИСОК ЛІТЕРАТУРИ	537
ПРЕДМЕТНИЙ ПОКАЖЧИК	552

Автори:



Ямпольський Леонід Стефанович - кандидат технічних наук, професор кафедри Технічної кібернетики Національного технічного університету України "КПІ". Автор понад 450 наукових і науково-методичних праць, в т. ч. 43 авторських свідоцтв на винаходи, 15 монографій, 15 підручників, ряд з яких перекладено англійською, японською, чеською та польською мовами.

Лауреат першої премії Київського політехнічного інституту (1986), Державної премії України в галузі науки і техніки (1996).

Сфера наукових інтересів - новітні інтелектуалізовані технології проектування, моделювання і верифікації гнучких комп'ютерно-інтегрованих систем.



Лісовиченко Олег Іванович - кандидат технічних наук, доцент кафедри Технічної кібернетики Національного технічного університету України "КПІ". Автор понад 70 наукових і навчально-методичних праць, в т. ч. 6 підручників.

Сфера наукових інтересів - моделювання і програмне забезпечення складних систем.



Олійник Володимир Валентинович - кандидат технічних наук, ст. викладач кафедри Технічної кібернетики Національного технічного університету України "КПІ". Автор 11 публікацій в т. ч. 1 підручника.

Сфера наукових інтересів – інтелектуалізація процесів моделювання та керування в комп'ютеризованих інтегрованих системах.

Навчальне видання

*Ямпольський Леонід Стефанович
Лісовиченко Олег Іванович
Олійник Володимир Валентинович*

Нейротехнології та нейрокомп'ютерні системи

підручник

Технічний редактор *Л. С. Ямпольський*
Коректор *А. М. Дзінько*
Комп'ютерна верстка та макет *Л. С. Ямпольський*
Мультимедійна *О. І. Лісовиченко,*
версія на диску *В. В. Олійник, Р. І. Дзінько*
Оформлення обкладинки *Д.В. Лепешин*

Викладені основні засади нейротехнології як складової загальної науки нейроінформатики; досліджені принципи утворення, методи побудування та особливості функціонування штучних нейронних сіток (ШНС) як інструментарію формального подання знань в системах штучного інтелекту. Систематизовано термінологічний та понятійний апарати нейросистем. Наведено класифікацію за вирішальними ознаками та математичні моделі базових штучних нейронів і створюваних на їх основі топологій ШНС. Розглянуто методи та алгоритми навчання ШНС, а також представлено способи підготовки останніх до практичного застосування з опрацюванням даних, редукцією, вербалізацією тощо. Обґрунтована можливість створення автоматизованих систем вибору топологій ШНС під умови (властивості) прикладних (модельовуваних) задач і запропонована логічна схема поетапного ітераційного синтезу ШНС з використанням концепції нечіткої метаідентифікації. Супроводжується численними прикладами реалізації ШНС.

Для всіх вивчаючих і використовуючих методи і засоби нейроінформатики.

Підписано до друку 04.02.2016 р. Формат 60х84 1/16
Папір офсетний. Гарнітура Times. Ум. друк. арк. 36,0.
Наклад 500 пр. Зам. 16-004

Видавець та виготівник ТОВ «Дорадо-Друк»
09000, м. Сквир, вул. Щорса, 7
(044) 501-75-69
www.dorado-druk.com.ua

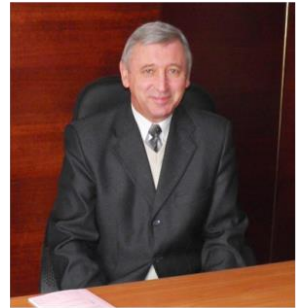
Свідцтво про внесення суб'єкта видавничої діяльності
у Державний реєстр ДК № 2600 від 01.09.2006 р.

Рецензенти:

Казак Василь Миколайович – д.т.н.,
професор, Директор науково-навчального
центру новітніх технологій Національного
авіаційного університету;



Осадчий Сергій Іванович – д.т.н.,
професор, завідувач кафедри автоматизації
виробничих процесів Кіровоградського
національного технічного університету;

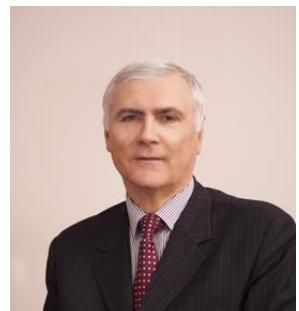


Бідюк Петро Іванович – д.т.н.,
професор, професор кафедри математичних
методів системного аналізу Інституту
прикладного системного аналізу НТУ України
“КПІ”



Відповідальний редактор:

Теленик Сергій Федорович, д.т.н.,
професор, зав. кафедри Автоматизації та
управління технічними системами НТУ
України “КПІ”



За спонсорської підтримки:

Діхтяр Захар Володимирович –
випускник 2006 року кафедри Технічної
кібернетики НТУ України “КПІ”,
ІТ-підприємець

