УДК 004.8

Л.С. Ямпольський, О.І. Лісовиченко, К.Б. Остапченко

ОБГРУНТУВАННЯ ВИБОРУ І ПІДГОТОВКА НЕЙРОННИХ СІТОК ДО МОДЕЛЮВАННЯ ПРИКЛАДНИХ ЗАДАЧ

Анотація: Розглядаються основні етапи обгрунтування вибору і підготовки ШНС до розв'язання прикладних задач. Запропонована модель поетапного синтезу топологій ШНС, яка враховує ітераційні процедури експериментального підбирання параметрів і власне навчання.

Ключові слова: штучні нейронні сітки, перевірка адекватності навчання, вербалізація ШНС, вибір топології ШНС.

1. Постановка задачі

Розширення інструментальної бази штучних нейронних сіток (ШНС), яким супроводжується останнє десятиріччя, дозволяє більш "тонко" і професійно підходити до вибору їх моделей, адекватних умовам розв'язуваної задачі.

Проте в більшості випадків недостатній рівень професійної підготовленості користувачів приводить до суттєвих додаткових матеріальних (апаратних, обчислювальних) і/або часових ресурсів, які, як правило, усуваються експериментально, що в свою чергу знижують ефективність застосування ШНС.

Іншою причиною неефективного використання навіть вірно вибраної топології ШНС слугує надмірна (для конкретної задачі) потужність сітки (а разом з тим, витратна складова її застосування), яка без особливих утруднень може бути спрощена (шляхом редукування чи регуляризації).

В решті решт, розв'язанню прикладних задач передують етапи підготовки ШНС щодо попереднього опрацювання даних і формування навчаючих вибірок.

Основними етапами обгрунтування вибору і підготовки ШНС до розв'язання прикладних задач можна вважати: *збирання даних для навчання*; *підготовка і попереднє опрацювання даних*; *вибір топології ШНС*; *експериментальне підбирання характеристик ШНС*; *експериментальне підбирання параметрів навчання*; *власне навчання*; *перевірка адекватності навчання*; *коректування параметрів, остаточне навчання*; *вербалізація ШНС з метою подальшого використання.* Розглянемо детальніше деякі основні етапи.

2. Обгрунтування вибору топологій нейросіток

Як правило, ШНС використовуються в двох варіантах: будується нейросітка, яка розв'язує певний клас задач; під кожний екземпляр задачі створюється деяка ШНС, яка відшукує квазіоптимальний розв'язок цієї задачі. Проте в обох випадках прийняття рішень лягає на користувача.

Покращення ситуації нам уявляється в утворенні умов для автоматизованого синтезу адекватних до прикладної задачі ШНС і полягає у: формуванні *набору вирішальних класифікаційних ознак* (НРКП) і створенні класифікатора ШНС на його основі; побудуванні чіткої *логічної моделі поетапного синтезу* ШНС; створенні строгої *узагальненої моделі вибору типових топологій* ШНС для конкретних прикладних задач, яка базується на нормалізованих системах подання знань [1] з використанням НРКП та *агентно-орієнтованого підходу* [2].

***Означення* 1.** *Набір вирішальних класифікаційних ознак ШНС* – така їх мінімально допустима сукупність, яка є *необхідною* для формалізації процесу подання основних властивостей та вибору задовольняючих топологій ШНС і *достатньою* для адекватного обслужування вимог (критеріїв оцінки) з боку прикладної розв'язуваної/моделюваної задачі.

***Означення* 2.** *Модель поетапного синтезу ШНС* – така послідовність їх перебирання в просторі НРКП, яка, будучи виконуваною користувачем і/або *мультиагентною підсистемою автоматизованого вибору* (МАПАВ), відтворює принципи агентно-орієнтованого підходу і дозволяє автономно виокремлювати топологію/топології ШНС, спроможну/спроможні задовольняти критерії обслуговування властивостей розв'язуваної/ моделюваної задачі.

Оскільки в ряді наших попередніх публікацій [3, 4] детально розглянутий етап обгрунтування вибору топології ШНС, далі зупинимось на двох інших визначальних етапах – *вербалізації ШНС з метою подальшого їх використання,* а також *підготовки і попереднього опрацювання даних.*

3. 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* 4.** *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 ШНС. Вербалізація звичайно здійснюється засобами спеціалізованих пакетів.

Мета проріджування нейросіток. Перед вербалізацією сітки, як правило за допомогою продукційних правил, для деяких видів ШНС було запропоновано спрощувати структуру сіток – *проріджувати*.

***Твердження* 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.

Треба мати на увазі, що *твердження* 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 ШНС можна віднести:

1. Спрощення архітектури нейронної сітки.
2. Зменшення кількості вхідних сигналів.
3. Зведення параметрів ШНС 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.

3.1. Основні напрями спрощення нейронних сіток

Класифікація методів спрощення нейронних сіток. Існує два напрями спрощення НС за рахунок (рис. 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 ШНС*) [5], або ті елементи, наявність яких в найменшому ступені впливають на зменшення похиб-

Рис. 1 – Класифікація методів спрощення ШНС

ки сітки. Найбільш розповсюдженими представниками цих методів є "*метод мозкового руйнування*", або *OBD*-метод (*Optimal Brain Damage*) [6], а також "*метод мозкового відтворення*", або *OBS*-метод (*Optimal Brain Surgeon*) [7].

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 ШНС [8], а заснований на аналогічному до weight-decay-підході з використанням вартісної функції метод придатний для видалення нейронів схованого шару [9].

3.2. Методи відсікання вагової компоненти нейросіток

3.2.1 Редукція нейронних сіток з урахуванням чутливості *Загальні особливості регуляризації ШНС.* 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 ШНС. Тому найкращим критерієм слід визнати урахування чутливості ШНС до варіацій ваг [10]. Без суттєвих наслідків для ШНС 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:

$$\Delta I = \sum_i g_i \Delta w_i + 0,5 \left[\sum_i h_{ii} \left[\Delta w_{ii} \right]^2 + \sum_{i \neq j} h_{ij} \Delta w_i \Delta w_j \right] + U \left(||\Delta w||^2 \right), \tag{1}$$

де Δw_i – варіація i-тої 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 5. *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 ШНС 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 *регуляризацію* ШНС) 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 ШНС за рахунок відсікання вагової компоненти (weight decay) так чи інакше базуються на загальній ідеї регуляризації [5], коли в звичайний мінімізаційний функціонал вводиться додатковий штрафний член

$$I^{(d)} = 0.5 \sum_p \sum_j \left(y^*_{pj} - y_{pj} \right)^2 + \frac{\chi}{2} \sum_i \sum_j w^2_{ij}, \tag{2}$$

де $\chi > 0$ – коефіцієнт штрафу; $p = \overline{1,\ P}$ – кількість подаваних образів навчання.

Мінімізація функціоналу (2) приводить до наступного алгоритму корекції ваг [11]:

$$\Delta_p w_{ij}\,(k+1) = -\beta \left(y^*_{pj}\,(k) - y_{pj}\,(k) \right) y'_{pj}\,(k) + \chi w_{nij}\,(k), \tag{3}$$

де $z'_{pj}\,(k) = \frac{\partial y_{pj}(k)}{\partial w_{ij}(k)}$; 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 [6]. 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 [6] пропонується виходити з того, що внаслідок додатної визначеності гессіану матриця $\mathbf{H}$ є діагонально домінуючою. Саме тому можна враховувати тільки діагональні елементи

$$h_{qq} = \frac{\partial^2 I}{\partial w_{ij}^2}, \tag{4}$$

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:

$$S_{ij} = 0.5 h_{qq} w_{ij}^2. \tag{5}$$

Видалення ваг з найменшими значеннями показника S_{ij} не викликає суттєвих змін в процесі функціонування ШНС, і цю процедуру редукції останньої можна подати послідовністю:

Крок 1. Повне попереднє навчання ШНС вибраної структури з використанням будь-якого алгоритму.

Крок 2. Визначення діагональних елементів гессіану (4), відповідних кожній вазі, та розрахунок значень параметру S_{ij} з (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 з них, 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 ШНС 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 [7] 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:

$$S_i = 0.5 \frac{w_i^2}{[\mathbf{H}^{-1}]_{ii}}, \tag{6}$$

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 [7]:

$$\Delta w = \frac{w_i}{[\mathbf{H}^{-1}]_{ii}} \mathbf{H}^{-1} \mathbf{\Pi}_i, \tag{7}$$

де $\mathbf{\Pi}_i$ – поодинокий вектор з одиницею в i-й позиції, тобто $\mathbf{\Pi}_i = [0, \ldots, 0, 1, \ldots, 0]^T$. Корекція виконується після видалення кожної чергової ваги і замінює повторне навчання ШНС, необхідне в разі використання OBD-методу. Процедуру OBS-методу регуляризації можна подати наступними кроками [7]:

Крок 1. Навчання ШНС попередньо вибраної структури ШНС аж до відшукання локального мінімуму цільової функції.

Крок 2. Обрахування оберненої гессіану матриці $^{-1}$ та вибір ваги w_i з найменшим значенням показника (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 (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.

3.2.2. 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:

$$I(w) = I^{(\mathrm{B})}(w) + \chi \sum_{ij} w_{ij}^2, \tag{8}$$

де $I^{(\mathrm{B})}(w)$ – *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 $I^{(\mathrm{B})}(w)$ стандартним методом зворотного поширення; корекції значень ваг, обумовленої модифікуючим фактором. Якщо через $w_{ij}^{(\mathrm{B})}$ позначити значення ваги w_{ij} після першого етапу навчання, то в результаті корекції ця вага буде модифікована за градієнтним методом найшвидшого спуску за формулою:

$$w_{ij} = w_{ij}^{(\mathrm{B})} \left(1 - \beta\chi\right), \tag{9}$$

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 видалятись, має підбиратися на основі численних експериментів для відшукання порогу, при якому процес навчання ШНС піддається найменшим збуренням.

Більш придатні результати, які не викликають зменшення значень усіх ваг, можна отримати модифікацією подання цільової функції у формі:

$$I(w) = I^{(\mathrm{B})}(w) + 0.5\chi \sum_{ij} \left(w_{ij}^2 / \left(1 + \sum\nolimits_q w_q^2 \right) \right). \tag{10}$$

Мінімізація цієї функції викликає не тільки редукцію міжнейронних 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:

$$w_{ij} = w_{ij}^{(\mathrm{B})} \left[1 - \beta\chi \frac{1 + 2 \sum\limits_{q \neq j} \left(w_{ij}^{(\mathrm{B})} \right)^2}{1 + 2 \sum\limits_{q.} \left(w_{ij}^{(\mathrm{B})} \right)^2} \right] \tag{11}$$

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 ШНС. При великих значеннях ваг, які ведуть до i-го нейрона, їх корекційна складова зникаюче замала і дуже мало впливає на процес редукції сітки.

3.2.3. Редукція нейросіток з використанням вартісної функції В роботі [12] запропоновано в якості штрафної функції, аналогічної до (2), застосовувати *квадратичну вартісну функцію* сигналів на виході нейронів схованого шару

$$B = \sum_j B\left(y_j\right)^2, \tag{12}$$

де $B\left(y_j\right)$ – монотонна функція виходу j-го нейрона.

Тоді мінімізований функціонал приймає вид:

$$I^{(\mathrm{B})} = \varsigma^{(I)} I + \varsigma^{(\mathrm{B})} \mathrm{B}, \tag{13}$$

де $\varsigma^{(I)}$; $\varsigma(\mathrm{B})$ – вагові коефіцієнти частинних критеріїв.

Для зменшення вартості схованих шарів, розташованих ближче до шару на виході, обчислюють похідні

$$\frac{\partial}{\partial w_{ij}} = \frac{\partial}{\partial z_j} \frac{\partial z_j}{\partial w_{ij}} = \upsilon_j^{(\mathrm{B})} \frac{\partial z_j}{\partial w_{ij}} = \upsilon_j^{(\mathrm{B})} y_i, \tag{14}$$

де

$$\upsilon_j^{(\mathrm{B})} = \frac{\partial\left(y_j^2\right)}{\partial z_j} = \frac{\partial\left(y_j^2\right)}{\partial y_j^2} \frac{\left(y_j^2\right)}{\partial y_j} \frac{y_j}{\partial z_j} = 2\mathrm{B}' y_j f_j'\left(z_j\right), \tag{15}$$

де $\mathrm{B}' = \frac{\partial\left(y_j\right)^2}{\partial y_j}$; $f_j'\left(z_j\right)$ – похідна активаційної (логістичної) функції входів НС.

Для ваг усіх шарів, розташованих після розглядуваного, вартісна функція дорівнює нулю. Якщо ж аналізованому шару передує схований, то для цього передуючого шару

$$\upsilon_j^{(\mathrm{B})} = f_{\mathrm{j}}'\left(nz_j\right)\sum_l \upsilon_l^{(\mathrm{B})} w_{jl}, \tag{16}$$

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:

$$\Delta w_{\imath j} = -\beta\varsigma^{(\mathrm{I})} y_i \upsilon_j^{(I)} - \beta\varsigma^{(\mathrm{B})} y_i \upsilon_j^{(\mathrm{B})} = -\beta y_i \upsilon_j^{(I,\mathrm{B})}, \tag{17}$$

де $\upsilon_j^{(I,\mathrm{B})}$ – 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 (2) наступним чином:

$$\tilde{I} = \varsigma^{(\mathrm{I})}\mathrm{I} + \varsigma^{(\mathrm{B})}\mathrm{B} + \varsigma^{(\mathrm{w})}\mathrm{W}. \tag{18}$$

3.3. Спрощення нейросіток видаленням нейронів

Видалення нейронів так чи інакше оцінюється наслідками, які відбиваються на значеннях величини помилковій реакції нейросітки.

3.3.1. Відсікання нейронів з урахуванням їх важливості ***Метод скелетонізації.*** Метод *скелетонізації* [8] 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 (приймемо до уваги, що для кожного нейрона необхідні P образів), то ця ситуація вимагала б $H(nP)$ пред'явлень навчаючих образів. Тому для скорочення кількості обчислень при визначенні ϑ використовується наближена формула, для виведення якої введений *параметр впливу* (attention strength) α_i $\left(i = \overline{1,n}\right)$ кожного з нейронів [11]

$$y_j = f\left(\sum_i w_{ij}\alpha_i y_i\right). \tag{19}$$

Якщо $\alpha_i = 0$, то нейрон не впливає на сітку, проте при $\alpha_i = 1$ він відповідає звичайному нейрону (див. рис. 2).

Подавши значущість нейронів як

$$\vartheta_i = I_{\alpha_i=0} - I_{\alpha_i=1}, \tag{20}$$

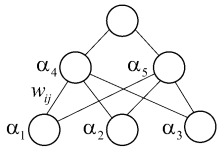

Рис. 2 – Нейросітка з коефіцієнтами впливу

Можна апроксимувати коефіцієнт ϑ_i похідною $\partial I/\partial\alpha_i$ в точці $\alpha_{i=1}$

$$\lim_{\beta\to 1}\frac{I_{\alpha_i=0}-I_{\alpha_i=1}}{\beta-1}=\left.\frac{\partial I}{\partial\alpha_i}\right|_{\alpha_i=1}. \tag{21}$$

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 $\widehat{\vartheta}_i$

$$\widehat{\vartheta}_i=\left.\frac{\partial I}{\partial\alpha_i}\right|_{\alpha_i=1}. \tag{22}$$

Похідна (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 $\widehat{\vartheta}_i$ зменшується експоненційно. В [8] запропоновано проводити корекцію $\widehat{\vartheta}_i$ у відповідності з залежністю:

$$\widehat{\vartheta}_i(k+1)=0.8\widehat{\vartheta}_i(k)+0.2\frac{\partial I}{\partial\alpha_i}. \tag{23}$$

Такий підхід дозволяє, на відміну від обрахування *квадратичної похибки*, яке використовується при настроюванні ваг, для визначення значущості застосовувати *лінійний функціонал*:

$$I_{\text{linear}}=\sum_n\sum_j\left|\left(y^*_{nj}-y_{nj}\right)\right|. \tag{24}$$

Як показано в [8], 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 черговістю кроків:

Крок 1. Навчання сітки доти, доки всі виходи усіх нейронів не стануть достатньо близькими до бажаних.

Крок 2. За залежностями (22) або (23) вираховується значущість $\widehat{\vartheta}_i$ 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 [12] запропоновано наступну модифікацію цільової функції:

$$I(w) = I^{(\mathrm{B})}(w) + \nu \sum_{i=1}^{N} \sum_{j=1}^{P} e\left(\Delta_{ij}^2\right), \tag{25}$$

де $I^{(\mathrm{B})}(w)$ – стандартно визначена цільова функція; ν – коефіцієнт ступеня відносного впливу коректуючого фактора на значення цільової функції; Δ_{ij} – змінення 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 $j(j = 1, 2, \ldots, P)$ 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:

$$e' = \frac{\partial e\left(\Delta_i^2\right)}{\partial \Delta_i^2} = \frac{1}{\left(1 + \Delta_i^2\right)^z}, \tag{26}$$

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.

3.3.2. Видалення нейронів з урахуванням вартісної функції Для видалення нейронів схованого шару С. Хенсоном і А. Праттом [9] використаний аналогічний weight-decay-методу (2) підхід з доповненням критерію I вартісним показником B

$$I^{(\mathrm{B})} = I + \mathrm{B}. \tag{27}$$

Корекція ваг здійснюється з умови $\min\limits_{w} I^{(\mathrm{B})}$.

Вид алгоритму корекції залежить від виду вартісної функції . Зокрема, якщо – квадратична функція відносно w_{ij}, тобто $\mathrm{B} = \sum_i \sum_j w_{ij}^2$, то алгоритм корекції ваг визначається видом [11]:

$$\Delta w_{ij}(k+1) = \beta\left(-\frac{\partial I}{\partial w_{ij}} - 2w_{ij}(k)\right), \tag{28}$$

Звідки

$$w_{ij}(k) = \beta\sum_{m=1}^{k}\left[(1-2\eta)^{k-m}\left(-\frac{\partial I(m)}{\partial w_{ij}}\right)\right] + (1-2\beta)^k w_{ij}(0). \tag{29}$$

Як вартісну Д. Руммельхарт запропонував наступну функцію:

$$\mathrm{B} = \sum_i \sum_j \frac{w_{ij}^2}{1+w_{ij}^2}, \tag{30}$$

Використання якої в алгоритмі похідних

$$\frac{\partial \mathrm{B}}{\partial w_{ij}} = \frac{2w_{ij}}{\left(1+w_{ij}^2\right)^2} \tag{31}$$

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:

$$w_j = \sum_i |w_{ij}|. \tag{32}$$

Поряд з видозміненою формою функції (3.30)

$$\mathrm{B} = \sum_j \frac{w_j^2}{1+w_j^2} \tag{33}$$

з похідною

$$\frac{\partial \mathrm{B}}{\partial w_{ij}} = \frac{2w_j sgn(w_{ij})}{\left(1+w_j^2\right)^2} \tag{34}$$

застосовуються також гіперболічна

$$\mathrm{B} = \sum_j \frac{w_j}{1+\lambda w_j} \tag{35}$$

з похідною

$$\frac{\partial \mathrm{B}}{\partial w_{ij}} = \frac{\lambda \mathrm{sgn}(w_{ij})}{(1+w_j)^2} \tag{36}$$

та спадна експоненційна функція

$$\mathrm{B} = \sum_j \left(1 - e^{-\lambda w_j}\right) \tag{37}$$

з похідною

$$\frac{\partial \mathrm{B}}{\partial w_{ij}} = \frac{\lambda \mathrm{sgn}\left(w_{ij}\right)}{e^{\lambda w_j}}. \tag{38}$$

Слід приймати до уваги, що в деяких задачах експоненційна спадна функція може приводити до того, що процес навчання не буде сходитися. Скоріш за все це відбувається через використання градієнта функціоналу як для зменшення похибки сітки, так і для видалення схованого нейрона.

4. Підготовка і попереднє опрацювання даних

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

- *репрезентативності* – дані повинні ілюструвати істинне положення речей у предметній галузі;
- *несуперечність* – 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 називається *навчаючею парою* або *навчаючим вектором*. Навчаючий вектор містить по одному значенню на кожний вхід сітки і, 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.

Якщо виникає необхідність використовувати нейросіткові методи для розв'язання конкретних задач, то перше, з чим стикається досліджувач після вибору і регуляризації ШНС – 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 досить трудомістким елементом нейросіткового аналізу. Успіх у навчанні ШНС 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 опрацьовування даних. Для побудування нейронечіткої 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) [13].

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 подані у числовій формі, і після відповідного нормування усі вхідні змінні і змінні на виході відображуються 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 $\tilde{x}_i$. Середня кількість інформації $\tilde{x}_i^{\alpha}$, яка надходить з кожним прикладом, дорівнює ентропії розподілення $H(\tilde{x}_i)$ 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 $\tilde{x}_i^{\alpha}$ рівномірно розподілені в одиничному інтервалі, інформація такої змінної максимальна.

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

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 опрацьовування*.

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 вхі-

дні 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* $\tilde{x}_i = (x - x_{i\,\min}) / (x_{i\,\max} - x_{i\,\min})$ в одиничний відрізок $\tilde{x}_i \in [0,1]$. Узагальнення для відображення даних в інтервал $[-1,1]$, який рекомендується для вхідних даних, тривіальне.

Твердження 2. *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 визначать, відповідно до попередньої формули, масштаб нормування. Це приведе до того, що основна маса значень нормованої змінної $\tilde{x}_i$ зосередиться поблизу нуля, тобто $[\tilde{x}_i] << 1$.

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

$$\tilde{x}_i = (x_i - \bar{x}_i)/\sigma_i, \tag{39}$$

де

$$\tilde{x}_i = P^{-1} \sum_{\alpha=1}^{P} x_i^{\alpha},\ \sigma_i^2 \equiv (P-1)^{-1} \sum_{\alpha=1}^{P} \left(x_i^{\alpha} - \bar{x}_i\right)^2. \tag{40}$$

В цьому випадку основна маса даних матиме одиничний масштаб, тобто типові значення усіх змінних будуть порівнянними (рис. 3)

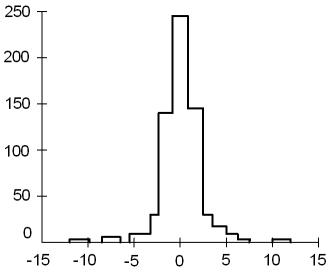

Рис. 3 – Гістограма значень змінної за наявності рідких, але значних за амплітудою відхилень від середнього

Проте, тепер нормовані величини не належать гарантовано одиничному інтервалу, більше того, розкид значень $\tilde{x}_i$ 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 можуть 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

$$\tilde{x}_i = f\left(\left(x_i - \bar{x}_i\right)/\sigma_i\right),\ f(\alpha) = \left(1 + e^{-\alpha}\right)^{-1} \tag{41}$$

нормує основну масу даних, одночасно гарантуючи, що $\tilde{x}_i \in [0,1]$ (рис. 4).

Як видно з рис. 4, розподілення значень після такого нелінійного перетворення значно ближче до рівномірного.

Усі вищеозначені 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 [14].

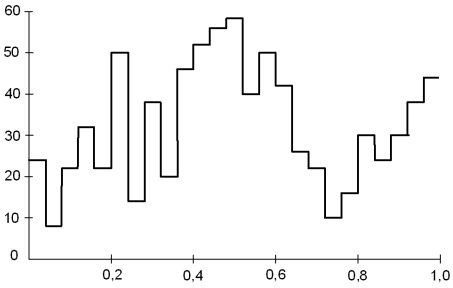

Рис. 4 – Нелінійне нормування з використанням логістичної функції $\mathbf{f}\,(\alpha) = \left(1 + e^{-\alpha}\right)^{-1}$

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 результатів 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 [15].

Традиційно застосовуваними інструментами скорочення розмірності вибірок даних є *методи відбирання інформативних ознак* (feature selection) [16] *та методи витягання ознак* (конструювання) (feature extraction) [17]. З іншого боку, розмірність даних можна скоротити шляхом формування вибірки меншого розміру з вихідної вибірки [18].

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* [19], 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*. 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 проектуються на простір меншої розмірності, породжений компонентами з найбільшою варіацією [14]. Один з недоліків класичного методу головних компонент полягає в тому, що це чисто лінійних метод, і він, відповідно, не може враховувати деякі важливі характеристики структури даних.

В теорії ШНС розроблені більш потужні алгоритми, здійснюючі *нелінійний аналіз головних компонент* [20]. 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, аніж у вхідному і вихідному шарах (рис. 5). Нейросітки подібного роду носять назву *автоасоціативних.*

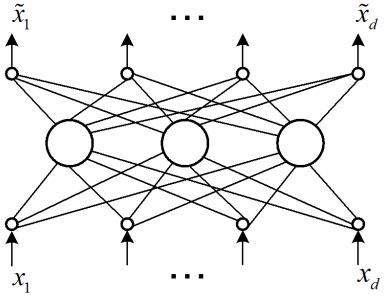

Рис. 5 – Автоасоціативні НС з "вузьким горлом" – аналог правила навчання Ойя

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:

$$\Delta w_{ij} = \eta y_i \left(x_j - \tilde{x}_j\right) = \eta y_i \left(x_j - \sum_{k=1}^{Z} y_k w_{kj}\right), \qquad (42)$$

де: $y_i = \sum\limits_{j=1}^{d} w_{ij} x_j \; (i = 1, \ldots, , Z)$; $\tilde{x}_j = \sum\limits_{k=1}^{Z} w_{kj} y_k \; (j = 1, \ldots, d)$; x_j – компонента вхідного вектора ($j = 1, 2, \ldots, d$); $\tilde{x}_j$ – виходи нейрона ($j = 1, 2, \ldots, d$); d – кількість нейронів у вхідному та вихідному шарах (розмірність вектора ознак); y_i – вихід з i-го нейрона прихованого шару, $i = 1, \ldots, Z$ – 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. 6) може бути збережений і використаний для пониження розмірності.

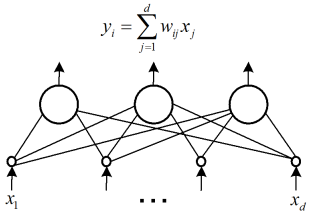

Рис. 6 – Шар лінійних нейронів

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

$$\Delta w_{ij} = \eta f(y_i) \left(x_j - \sum_{k=1}^{Z} f(y_k) w_{kj} \right) \tag{43}$$

веде до нової якості.

Таким чином, нейроалгоритми уявляють собою зручний інструмент нелінійного аналізу, дозволяючий відносно легко находити 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 апріорної 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.

Висновки

Розглянуто три етапи, які в основному і визначають особливості вибору і підготовки ШНС до розв'язання прикладних задач. Треба зазначити, що запропонована у [4] модель поетапного синтезу топологій ШНС враховує ітераційні процедури експериментального підбирання параметрів і власне навчання, а також перевірку адекватності останнього з коректуванням його параметрів, тобто практично всі заявлені в постановці задачі до статті етапи. А викладені в [3] підходи щодо мультиагентної реалізації топологій ШНС дозволяють ставити питання про автоматизацію процесу вибору бажаної топології ШНС під розв'язувану задачу в практичну площину.

Список використаних джерел

1. *Ямпольский Л.С.* Автоматизированные системы технологической подготовки робототехнического производства / Ямпольский Л.С., Калин О.М., Ткач М.М. – К.: Вища шк., 1987. – 271 с.
2. *Bellifemine F.L., Caire G. and Greenwood D.* Developing Multi-Agent Systems with JADE. – Wiley, 2007
3. *Ямпольський Л.С.* Мультиагентная реализация выбора топологии нейросетей для моделирования прикладных задач / Л.С. Ямпольский, Е.С. Пуховский, О.И. Лисовиченко // В научном зб.: Блъгарско списание за инженерно проектиране. – София: Технически университет, Брой №20, октомбри 2013. – С. 75–92
4. *Ямпольський Л.С.* Нечітка ітераційна метаідентифікація штучних нейросіток в мультиагентному середовищі // Вісник кіровоградського національного технічного університету — Кіровоград: КНТУ. – №26 – 2013. – С. 207–218
5. *Тихонов А.Н., Арсенин В.Я.* Методы решения некорректных задач. – М.: Наука, 1980. – 223 с.
6. *LeCun Y., Denker J.S. and Solla S.A.* Optimal Brain Damage / In Proceedings of the Neural Processing Systems 2. Touretsky D.S. (ed). – Morgan Kaufmann, 1990. – P. 598-605
7. *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
8. *Mozer M.C., Smolensky P.* Skeletonization: a Technique for Trimming the Fat from a Network Via Relevance Assesment // In Advanced in Neural Information Processing Systems 1. Tourezky D.S. (ed), 1989. – P. 107–105

9. *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

10. *Осовский С.* Нейронные сети для обработки информации. – М.: Финансы и статистика. – 2002. – 345 с.

11. *Руденко О. Г., Бодянский Е. В.* Основы теории искусственных нейроннных сетей. – Харьков: ТЕЛЕТЕХ, 2002. – 317 с.

12. *Chauvin Y.* A Back-Propagation Algorithm with Optimal Use of Hidden Units // Advanced in Neural Information Processing Systems 1. Tourezky D.S. (ed), 1989. – P. 519–526

13. *Михайлюк П.П.* Програмний комплекс синтезу нейронечітких моделей технологічних процесів / Дисертація на здобуття наукового ступеня канд. техн. наук. – С.-Петербург, 2007. – 199 с.

14. *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.

15. *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 СМИТ, 2014. – 284 с.

16. *Гуляев В.А.* Вычислительная диагностика. – К.: Наукова думка, 1992. – 232 с.

17. *Биргер И.А.* Техническая диагностика. – М.: Машиностроение, 1978. – 240 с.

18. *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

19. Bishop C.M. “Neural Networks and Pattern Recognition”. – Oxford Press. – 1995

20. *Горбань А.Н., Россиев Д.А.* Нейронные сети на персональном компьютере. – СП “Наука”. – РАН. – 1996

Отримано 24.04.2015 р.