

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАУКОВО-НАВЧАЛЬНИЙ
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра математичного моделювання і аналізу даних

До захисту допущено:

Завідувач кафедри

Куссуль Наталія Миколаївна

«__» _____ 2022 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Математичні методи
моделювання, розпізнавання образів та безпеки даних»
спеціальності: 113 «Прикладна математика»

на тему: «Модель розвитку простих організмів з використанням
генетичних алгоритмів та глибинного навчання»

Виконав: здобувач вищої освіти IV курсу, групи ФІ-82
Татенко Вадим Сергійович

Керівник: Кандидат фізико-математичних наук
Олександр Арсенійович Орехов _____

Рецензент: Кандидат технічних наук
Яковлев Сергій Володимирович _____

Засвідчую, що у цій дипломній
роботі немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАУКОВО-НАВЧАЛЬНИЙ
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра математичного моделювання і аналізу даних

Рівень вищої освіти — перший (бакалаврський)
Спеціальність (освітня програма) — 113 Прикладна математика,
ОПП «Математичні методи моделювання, розпізнавання образів та
безпеки даних»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Куссуль Наталія Миколаївна

«__» _____ 2022 р.

ЗАВДАННЯ
на дипломну роботу

Студент: Татенко Вадим Сергійович

1. Тема роботи: *«Модель розвитку простих організмів з використанням генетичних алгоритмів та глибинного навчання»*,

керівник:

Кандидат фізико-математичних наук Олександр Арсенійович Орехов,

затверджені наказом по університету №__ від «__» _____ 2022р.

2. Термін подання студентом роботи: «__» _____ 2022р.

3. Вихідні дані до роботи: Набір нейронів для евристичного заповнення популяції.

4. Зміст роботи: Огляд існуючих методів побудови генетичного алгоритму; Розгляд основних положень та визначень стосовно нейронних мереж; Побудова моделі розвитку простих організмів з використанням генетичного алгоритму та глибинного навчання.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): Презентація доповіді

6. Дата видачі завдання: 10 вересня 2021 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	20-30 жовтня 2021 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Листопад-грудень 2021 р.	Виконано
3	Написання 1-го розділу	Січень 2022 р.	Виконано
4	Написання 2-го розділу	Лютий 2022 р.	Виконано
5	Написання 3-го розділу	Березень 2022 р.	Виконано
6	Створення вступів та висновку, написання анотацій	Травень 2022 р.	Виконано
7	Здача готової роботи на кафедрі		
8	Попередній захист на кафедрі		
9	Подання завершеної роботи до Державної екзаменаційної комісії		

Студент _____ Татенко В.С.

Керівник _____ Орехов О.А.

РЕФЕРАТ

Кваліфікаційна робота містить: 63 стор., 36 рисунки, 5 таблиць, 9 джерел.

Об'єктом дослідження є модель розвитку простих організмів.

Предметом є дослідження поведінки простих організмів при зміні початкових параметрів.

Метою дослідження є побудова та дослідження моделі розвитку простих організмів з використанням генетичного алгоритму та нейронної мережі.

На основі проаналізованої літератури було побудовано модель розвитку простих організмів використовуючи генетичний алгоритм та нейронну мережу. Зроблено аналіз параметрів моделі та підбрано оптимальні для кращої роботи моделі.

ЕВОЛЮЦІЙНІ АЛГОРИТМИ, ГЕНЕТИЧНІ АЛГОРИТМИ,
НЕЙРОННІ МЕРЕЖІ, ПОПУЛЯЦІЯ, ОРГАНІЗМ, РОЗВИТОК,
СЕЛЕКЦІЯ, СХРЕЩУВАННЯ, МУТАЦІЯ

ABSTRACT

Qualification work contains: 63 pages, 36 drawings, 5 tables, 9 sources.

The object of study is a model of development of simple organisms.

The subject is the study of the behavior of simple organisms when changing the initial parameters.

The aim of the study is to build and study a model of development of simple organisms using genetic algorithms and neural networks.

Based on the analyzed literature, a model of development of simple organisms was built using a genetic algorithm and a neural network. The analysis of model parameters is made and the optimal ones for the best work are selected.

EVOLUTIONARY ALGORITHMS, GENETIC ALGORITHMS,
NEURAL NETWORKS, POPULATION, ORGANISM, DEVELOPMENT,
SELECTION, CROSSING, MUTATION

ЗМІСТ

Перелік умовних позначень, скорочень і термінів	8
Вступ.....	9
1 Еволюційні та генетичні алгоритми.....	11
1.1 Еволюційні алгоритми.....	11
1.2 Генетичні алгоритми.....	12
1.3 Генетичне уявлення.....	13
1.4 Пристосованість.....	16
1.5 Структура алгоритму	18
1.5.1 Популяція	18
1.5.2 Представлення популяцій.....	20
1.5.3 Селекція	22
1.5.4 Схрещування.....	25
1.5.5 Мутація	28
1.5.6 Політика відбору	31
1.6 Застосування генетичного алгоритму	32
Висновки до розділу 1.....	36
2 Нейронні мережі	37
2.1 Структура нейронних мереж	38
2.1.1 Функція втрат	40
2.1.2 Функція активації	41
Висновки до розділу 2.....	47
3 Аналіз та реалізація моделі.....	48
3.1 Реалізація моделі	48
3.1.1 Поле.....	48
3.1.2 Організми.....	49
3.1.3 Генетичний алгоритм та нейронна мережа	49
3.2 Аналітична частина	50
3.2.1 Кількість нод у прихованому шарі	51

3.2.2	Методи схрещування	7 53
3.2.3	Методи мутації.....	54
3.2.4	Частота мутації	55
	Висновки до розділу 3.....	56
	Висновки	58
	Перелік посилань	59
	Додаток А Таблиці результатів симуляцій	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

EA — еволюційний алгоритм,

ГА — генетичний алгоритм,

AVG — середнє значення пристосованості.

ВСТУП

Еволюція — дуже цікавий та корисний процес у будь-якій сфері або житті будь-якого організму. Вона надає неймовірні бонуси та покращення, якщо проходить успішно. Однією з багатьох особливостей еволюції є те, що вона має на увазі процес мутації та рекомбінації. Це надає випадкову різноманітність, яка з правильним використанням приведе до успішного виживання та покращення організму, або знаходження кращого результату.

Навчання всюди супроводжує еволюцію, без нього нічого не відбудеться. Від його успішності багато залежить. Тому коректність навчання відіграє вагомий роль в процесі еволюції.

Модель розвитку простих організмів буде реалізовуватись за допомогою генетичного алгоритму та нейронної мережі. Робота торкнеться таких понять еволюції, як розвиток, селекція, схрещування, мутація.

Генетичний алгоритм використовують для розв'язання оптимізаційних задач. Тож у роботі будуть підібрані оптимальні параметри для кращого розвитку організмів. Буде проаналізовано особливість поведінки моделі залежно від кількості прихованих нейронів у нейронній мережі, від методу схрещування, від методу мутації, від частоти мутації.

Актуальність дослідження

На даний момент нейронні мережі використовуються для вирішення великої кількості задач. Генетичний алгоритм використовується для оптимізації. Поєднавши ГА та нейронні мережі, отримаємо непоганий виграш в часі для вирішення задач нейронною мережею. Зокрема дослідження в цьому напрямку надасть можливість отримати нові методи вирішення задач.

Мета дослідження

Метою дослідження є побудова та аналіз моделі розвитку простих організмів з використанням генетичного алгоритму та нейронної мережі.

Задача дослідження

- Розглянути необхідні положення, визначення про генетичний алгоритм та методи його реалізації
- Розібрати необхідну інформацію стосовно нейронних мереж
- Реалізувати модель розвитку популяції простих організмів
- Виконати аналіз поведінки простих організмів в залежності від зміни початкових параметрів

Об'єкт дослідження

Об'єктом дослідження є модель розвитку простих організмів.

Предмет дослідження

Предметом є дослідження поведінки простих організмів при зміні початкових параметрів.

Методи дослідження

При вирішенні поставлених задач використовувались такі методи дослідження: моделювання, спостереження, порівняння, аналіз.

Наукова новизна

Наукова новизна отриманих результатів полягає в побудові моделі розвитку простих організмів за допомогою комбінації генетичного алгоритму та нейронної мережі, та оптимізації отриманої моделі.

Практичне значення

Запропонована модель може використовуватись при моделювання простих організмів та у реальних проектах.

1 ЕВОЛЮЦІЙНІ ТА ГЕНЕТИЧНІ АЛГОРИТМИ

Еволюційні алгоритми — обчислювальні методи пошуку на основі природнього відбору.

До традиційних варіацій ЕА можна віднести такі: еволюційна стратегія, генетичний алгоритм, генетичне програмування, генетичне поліпшення, граматична еволюція, лінійне генетичне програмування, декартове генетичне програмування, дифереціальна еволюція, програмування експресії генів. Також треба відмітити, що це далеко не весь перелік, який постійно змінюється і має різного виду варіації в залежності від потреб.

Також треба виділити важливу особливість кожного ЕА — вони засновані на еволюційному процесі, який має визначені кроки з їх варіаціями.

1.1 Еволюційні алгоритми

Еволюційний алгоритм тісно пов'язаний з механізмами зміни форм живих організмів, їх пристосованістю до виживання, та подальшого розвитку. Ті, які прагнуть розвитку, під час еволюції надають найкорисніші характеристики для нащадків. Це робить більш ймовірним успішне виживання організму.

Процес вирішення обчислювальної задачі еволюційним алгоритмом протікає через розвиток набору екземплярів (населення), які несуть в собі рішення. Кожен з цього набору зростає по своєму, після чого завдяки методам селекції обираються кращі екземпляри, як батьки наступного покоління[1].

Також в алгоритмі присутній оператор мутації екземпляра, який надає більше простору для огляду рішень задачі, яку запропонує модель.

Треба зазначити, що це надає захист від конвергенції, що призводить до поганих наслідків при вирішенні поставленої задачі.

Для того, щоб виявити кращий розв'язок, модель передбачає наявність функції пристосованості.

Еволюційний алгоритм можна вважати, як «метод останньої надії», через його здатність до вирішення задач, де класичні методи не є ефективними. Мова йде про задачі, які є складними за своєю об'ємністю, розміром проблемної області, а кількість цілей робить вирішення задачі важким для дослідження. За цих обставин рішення більш схоже на «достатньо хороше», ніж рішення з високою точністю (але це не виключає того, що точність є ціллю). ЕА виступає поганим кандидатом для вирішення простих задач, де легше та коректніше використати звичні алгоритми, які добре справляються з вирішенням таких завдань.

Зважаючи на те, що еволюційний алгоритм застосовується в основному до великих та складних задач, очевидним є те, що цей факт передбачає наявність великої кількості параметрів.

Задання значень параметрів відіграє значну роль ефективності моделі, яку ми отримуємо в результаті. Зокрема фіксованість параметрів негативно впливає на результат, через що рішення може бути неоптимальним. Тому для запобігання цього моменту реалізується адаптивність алгоритму, яка корегує параметри з плином його роботи. І звичайно існують різні види адаптацій, які можна використовувати в еволюційних алгоритмах для забезпечення зазначеної особливості роботи.

1.2 Генетичні алгоритми

Генетичні алгоритми — комп'ютерні програми, які моделюють природну еволюцію, все частіше застосовуються в багатьох дисциплінах. Вони були використані для вирішення різноманітних задач оптимізації від пошуку архітектури нейронної мережі до стратегічних ігор, а також для моделювання феноменів адаптації та навчання.

Генетичні алгоритми здатні досліджувати великі та складні простори можливих рішень, швидко знаходити перспективні елементи та надавати адекватний інструмент моделювання для опису еволюційних систем, від ігор до економіки. Однак вони страждають від високих витрат на обчислення та складної конфігурації параметрів.

Останні розробки, такі як обчислення на графічному процесорі, паралельні та квантові обчислення, концепція потужних методів керування параметрами та нові підходи до стратегій представлення, можуть стати ключем до подолання цих обмежень.

Генетичний алгоритм є членом сімейства еволюційних алгоритмів, які є методами обчислювального пошуку, натхненними природним відбором. Вони моделюють дарвінівську еволюцію на окремих сутностях, зібраних у популяції. Таким чином жирафи еволюціонували до довшої шиї або люди з плином часу стали більш пристосованими до життя в середовищі, яке їх оточує. Еволюція, синтезована в комп'ютерній програмі, прагне покращити ці сутності. Вони можуть бути будь-якими: цифровими (тобто штучними) організмами, бітовими рядками, комп'ютерними програмами, фінансовими стратегіями, тощо [2].

Оскільки еволюція впливає на сутності, наступні версії популяції, які ще називають поколіннями, розвиваються, щоб максимізувати свою придатність, тобто здатність виживати в середовищі або виконувати задане завдання, або максимізувати показник продуктивності. Подібним чином одноклітинні організми еволюціонували до *homo sapiens*, ЕА перетворюють невитончені тіла на складні, адаптивні сутності. Як комп'ютерна програма, весь цей процес еволюції починається зі способу представлення цих сутностей: генетичного уявлення.

1.3 Генетичне уявлення

ГА утворюють окрему підкатегорію ЕА, явно використовуючи генетичне представлення цих сутностей. Тобто, замість математичних

функцій чи абстрактних об'єктів, сутності формалізовані в комп'ютерній програмі способом, близьким до генетики, до послідовності ДНК. Представлення сутностей - це послідовність символів, що кодують характеристики. Те, як характеристики або поведінка сутностей формується в комп'ютерній програмі, позначається представленням (або кодуванням).

У ГА ці сутності представлені генотипом і фенотипом, поняттями з генетики. Представлення з використанням генотипу можна розглядати як кодування генетичної інформації про їхні характеристики у спрощеній формі ДНК. Хоча деякі ГА не роблять цієї відмінності та змішують генетичну інформацію та характеристики сутності.

Генотип сутності полягає в її генетичній інформації. Вона може бути виражена різними способами. Популярною технікою представлення в ГА є бітова послідовність з двійковим кодуванням, тобто складається з 0 і 1. Подібно до ДНК людини, генетична інформація також може зберігатися за допомогою алфавітного кодування, напр. А,С,Т,Г як у ДНК. Сфера генетичного програмування навіть розглядає генотипи, що складаються з послідовностей або дерев, або комп'ютерних інструкцій, або математичних операторів.

Незважаючи на таку різноманітність уявлень, генотипи в основному кодуються як рядок або послідовність. Ці рядки часто називають хромосомами, послідовністю елементів, що позначаються генами, за аналогією з тим, як генетична інформація зберігається в живих організмах. J. Holland, 1992 використовував термін схема або будівельний блок для опису деяких конкретних груп генів, які є будівельними блоками генотипу сутності. Розташування генів у хромосомі (рис. 1.1) позначається їхньою алелею, і зазвичай припускають, що кожен ген має свою власну алелю на хромосомі: у послідовності [1, 2, 3] кожен елемент є унікальним геном, присутнім у своїй окремій алелі. Різні значення, які може набувати кожен ген, позначаються алелями. Наприклад, у бінарному кодуванні кожен ген може мати або алель 0, або 1. Бінарна

хромосома стає бітовим рядком, тобто послідовністю нулів і одиниць. У наведеному вище алфавіті «ДНК» кодування кожен ген може мати алелі А С Т або G.

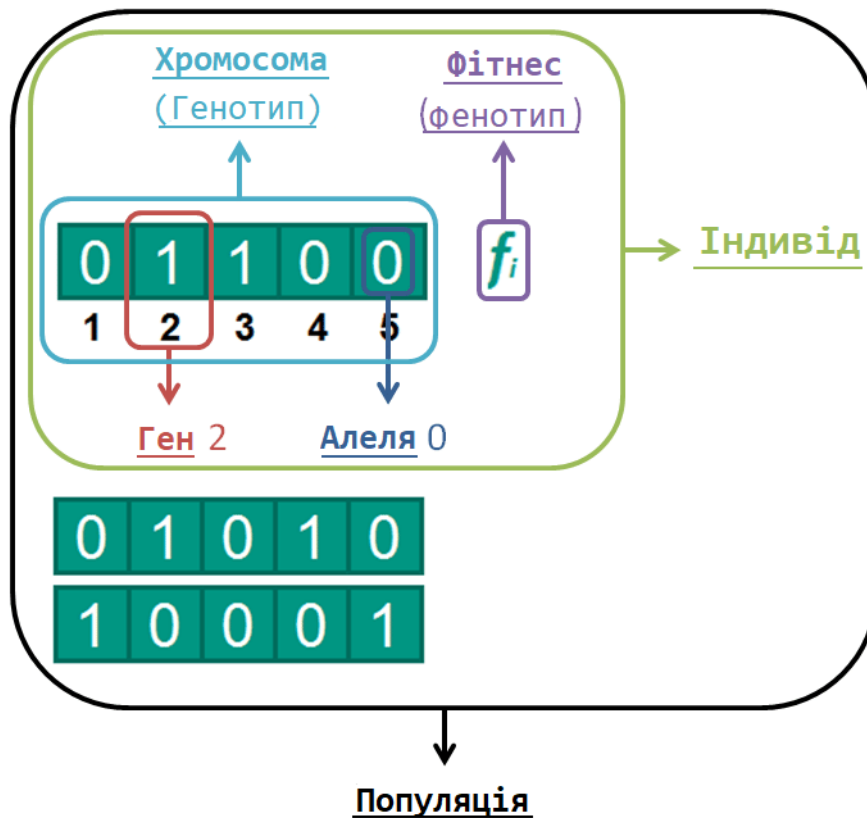


Рисунок 1.1 – Загальне представлення моделі

Незважаючи на те, що більшість живих організмів мають кілька хромосом, присутніх у кількох копіях, зазначимо, що ГА часто обмежуються сутностями: гаплоїдними та моноплоїдними. Гаплоїдні: їх генотип характеризується тільки однією хромосомою. Моноплоїдні: ці сутності мають лише одну копію кожної хромосоми.

Фенотип відповідає характеристикам сутності: формі, розміру, кольору, здібностям, тощо, які закодовані в генетичній інформації (генотипі).

Генотип і фенотип відрізняються: якщо фенотип можна спостерігати, генотип — ні. Але наразі важливо додати роль навколишнього середовища. Зазвичай вважається, що генотип і

середовище сутності разом визначають його фенотип, а не лише генетична інформація. Цей аспект часто відсутній у минулих та поточних ГА. Зазвичай вони вважають відповідність між генотипом і фенотипом фіксованою незалежно від контексту навколишнього середовища.

Тепер ми показали спосіб представлення сутностей у нашій програмі розвитку. Як і в природі, рушійною силою їх еволюції є пристосованість.

1.4 Пристосованість

У природних системах еволюція надає перевагу істотам, які є більш здатними до розмноження через певні переваги: вони можуть бути сильнішими, вони можуть бути більш привабливими, вони можуть бути більш пристосованими до свого середовища. У ГА пристосованість вимірює результативність суб'єктів у популяції в адаптації до даного середовища або виконання даного завдання. У природній еволюції критерії придатності є неявними. Сутності борються за відтворення та виживання всіма доступними засобами, які можна описати за допомогою симуляції.

Однак у більшості ГА критерій припристосованості стає явним шляхом введення функції пристосованості, що дозволяє суворо пов'язувати фенотипи з придатністю. Це тісно пов'язано з подібними концепціями винагород у навчанні, функції виплат/втрат у загальному машинному навчанні або функції цінності в оптимізації. Ці фітнес-функції можуть бути простими, наприклад підрахувати кількість одиниць у бітовому рядку. Інші можуть бути набагато складнішими, наприклад вирішення багатоцільової оптимізаційної проблеми. Ці функції фітнесу неявно або явно визначають ландшафт фітнесу.

Фітнес-ландшафти – це фізичне уявлення про взаємозв'язок між фенотипами та фітнесом. Деякі фенотипи можуть мати дуже високу пристосованість, що відповідає вершинам або горам у фітнес-ландшафті. Деякі форми поведінки можуть мати низьку придатність і

відобразатимуться в ландшафті як долини або печери. Одна важлива відмінність між ГА та їхніми двоюрідними братами з машинного навчання полягає в тому, що, розвиваючи фенотипи сутностей у популяції, ГА підтримує популяцію фенотипів, досліджуючи фітнес-ландшафт, керуючись еволюційними операторами, а не унікальною індивідуальністю, що низхідна (що рухається в напрямку найбільшого поліпшення, що вимірюється похідними функції придатності).

Еволюція популяції прагне ідентифікувати фенотипи з хорошою придатністю: втеча з долин і підйом в гори. Таким чином, ГА досліджує простір пошуку: простір можливих рішень або сутностей. Наскільки великим, гладким або жорстким є цей ландшафт фітнесу та наскільки важким може бути цей процес дослідження, залежатиме від конкретної проблеми, яка розглядається, та генетичного уявлення, яке використовується. Під гладким ми будемо мати на увазі фітнес-ландшафт, який є однопіковим або уні-модальним. В іншому випадку він може мати кілька гір (мультимодальний), або «ховати» гору в низькій долині, або представляти деякі неправильні структури: це характеризує жорсткий фітнес-ландшафт.

Можна було розглядати ГА як популяцію туристів, які намагаються знайти найвищу гору країни. Спочатку вони випадковим чином розміщуються у фітнес-ландшафті і бачать лише дуже малу територію, на якій вони знаходяться. Кожен має свою власну стратегію для походу: одні підуть на північ, інші підуть на південь, деякі вирішують рухатися випадковим чином, інші хочуть бігти. Ці мандрівники спілкуються один з одним, поєднують інформацію про свої позиції та пересуваються ландшафтом, намагаючись знайти вищу гору, щоб піднятися. Незважаючи на те, що кожен розвивається у своїх власних інтересах, вони розвивають колективний інтелект, використовуючи інформацію та стратегії інших туристів, щоб покращити власні. Якщо агент знаходить величезну гору, що йде на північ, деякі інші агенти більш схильні

включати північний напрямок у свій рух. Завдяки цьому ця децентралізована популяція мандрівників досягає сильної здатності досліджувати великі, нерівні ландшафти та швидко визначати хороші регіони, тобто хороші місця в цьому ландшафті. Це дослідження інтуїтивно важче, коли є кілька гір різного розміру та долин між ними. Але колективний характер пошуку робить ГА все ж здатним досягти найвищої гори.

1.5 Структура алгоритму

Генетичні алгоритми, по суті, виконують п'ять основних кроків (рис. 1.2), які присутні у всіх ГА, хоча з різними реалізаціями. Спочатку створюється початкова сукупність сутностей: перше покоління. Далі описані кроки виконуються протягом вказаної кількості ітерацій. Суб'єкти поточного покоління оцінюються з точки зору пристосованості. По закінченню виконання ітерацій обираються батьки з найбільшою пристосованістю для подальшого розмноження. Розмножуючись, вони створюють потомство, яке переймає частину їх генетичної інформації. На отримані генотипи дітей впливають мутації, які збільшують генетичну різноманітність популяції. Повторення цієї процедури до тих пір, поки ми не отримаємо повністю нове покоління, є кроком ГА. Після створення нового покоління ми повторюємо кроки програми, поки ця еволюція не досягне певної мети або поки не буде досягнуто максимальної кількості вказаних ітерацій [3].

1.5.1 Популяція

Чисельність популяції — важливий параметр, бо вона є нашим набором рішень, який повинен надати найкращу відповідь на задачу, яка розглядається. Через це треба ретельно підійти до визначення цього

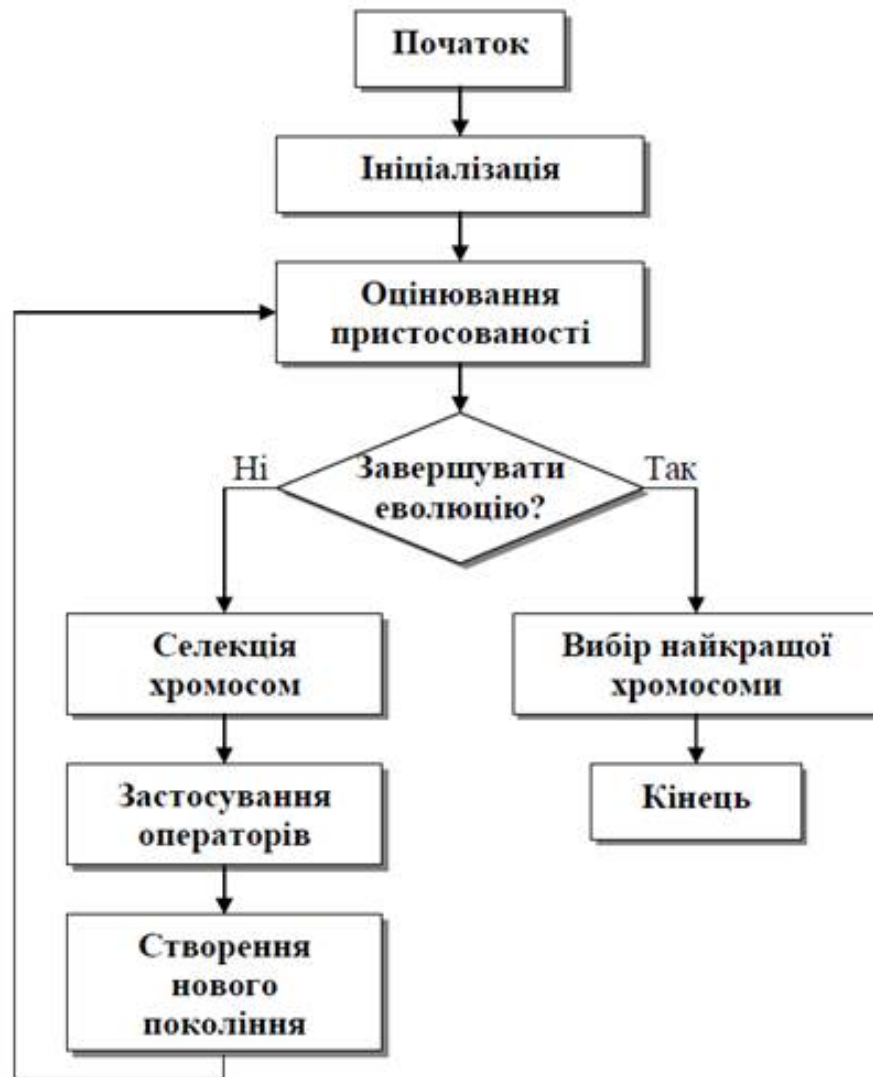


Рисунок 1.2 – Алгоритм ГА

параметра. Велика чисельність надає переваги у різноманітності рішень і точності знаходження коректного результату. Проте, в данному випадку ми зіштовхуємося з проблемою, що модель буде потребувати великої кількості ресурсів на еволюцію поколінь. В свою чергу, занадто мале значення чисельності популяції не дасть нам того оптимального рішення, якого ми прагнемо.

Через це можна назвати завдання цього параметру «грою в угадайку» під керівництвом спроб та помилок. Хоча зазвичай, у більшості ГА чисельність популяції фіксована й обмежується кількома десятками чи сотнями особин. Проте в живих системах чисельність популяцій дуже велика, а їх розміри є продуктом складних екологічних взаємодій.

Визначитись з величиною популяції — це одна справа. Є ще такий момент, як ініціалізація популяції. І тут в нас є два варіанти, як це можна зробити. Перше, і доволі очевидне, — випадкова ініціалізація початкових екземплярів. Другий варіант — евристична ініціалізація, але навідміну від першого варіанту, евристика не дасть нам такої різноманітності, як рандом.

Якщо казати про долю популяції між ітераціями, то існують дві моделі населення. Модель покоління: з популяції розміру n , генерується n нащадків, які замінюють n екземплярів у поточному поколінні. Стаціонарна модель: генерується декілька нащадків, які замінюють такуж кількість екземплярів у поточному поколінні.

1.5.2 Представлення популяцій

Одним з основних аспектів популяції є метод її представлення. Від того, яким чином будуть "зображені" хромосоми організмів, залежить успішність роботи генетичного алгоритму. Тож подивімось на можливі варіанти представлення.

Бітове представлення є бітовою стрічкою (рис. 1.3) довжини n , яка складається з нулів та одиниць. Дуже зручно використовувати у випадках, коли алея кожного гену відповідає на питання "так" чи "ні".

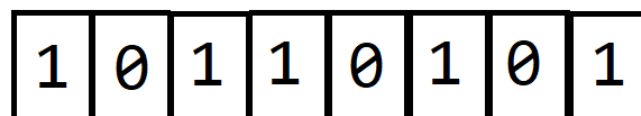


Рисунок 1.3 – Бітове представлення

Реальне представлення. Використовується для задач, в яких ми хочемо представити гени з найбільшою точністю значень (рис. 1.4), бо в

такому випадку вона буде обмежена лише здатностями комп'ютера до розрахунку.

0.3	2.1	0.5	0.7	0.2	0.8	1.3	0.4
-----	-----	-----	-----	-----	-----	-----	-----

Рисунок 1.4 – Реальне представлення

Цілочислене представлення. Навідміну від бітового представлення, яке несе в собі лише 2 варіанти в просторі рішень, даний метод (рис. 1.5) представлення розрахований на більшу кількість класів, які можна закодувати під цифру.

1	2	4	3	1	2	4	3
---	---	---	---	---	---	---	---

Рисунок 1.5 – Цілочислене представлення

Представлення перестановок. Даний метод (рис. 1.6) представлення є дуже корисним для випадків, коли задача заключається в пошуку оптимальної послідовності дій. Також можна враховувати умову відсутності повторень дій, які необхідно виконати.

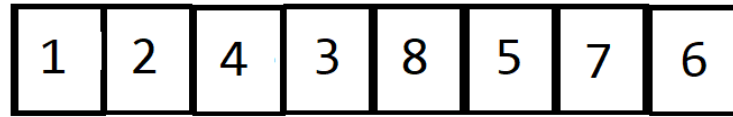


Рисунок 1.6 – Представлення перестановок

1.5.3 Селекція

Селекція — це метод, під час якого особини в поточній популяції обираються як початкові батьки для наступних поколінь. У цифровій еволюції батьки не обмежуються двома, можна вибрати будь-яке число. Необхідно приділити достатню увагу цьому моменту, через те, що від якості цього етапу залежить подальший розвиток покоління, а значить точність і достовірність отриманого результату.

З біологічної сторони зрозуміло, що у краще пристосованих батьків будуть кращі нащадки. Але у випадку ГА є момент, який слід зазначити. Якщо еволюція поколінь буде часто залежити від одного правильного розв'язку, який обирається впродовж декількох поколінь поспіль, то наше рішення буде дуже наближеним до цього одного правильного розв'язку, що є поганою практикою. Модель зіштовхнеться з проблемою відсутності різноманітності. А розмаїття надає нам доступ до найбільш точних рішень і більший простір для дослідження.

Різноманітність є важливим поняттям, коли йдеться про здорове населення. Здорове населення має важливе значення для пошуку правильних рішень. Вона безпосередньо пов'язана з кількістю варіацій, що застосовуються до сутностей. Можна стверджувати, що локальні схеми відбору (стаціонарні), природньо, з більшою ймовірністю збережуть різноманітність порівняно з глобальною схемою відбору. Місцевий відбір

означає, що еволюція потенційно відбувається з різними темпами в популяції.

Приклад локальної схеми відбору називається **турнірним відбором** (рис. 1.7). Як видно з назви, вибір вимагає проведення турніру між випадково обраною групою осіб. Таких турнірів буде проведено необхідна кількість, поки всі індивіди не пройдуть турнір. Потім обирається «переможець» кожного турніру для подальшої еволюції. Таким чином більша ймовірність того, що еволюція буде проходити з більшою різноманітністю.



Рисунок 1.7 – Турнірний відбір

Наступний метод має назву **вибір по рейтингу** (таблиця 1.1). Він може працювати з від'ємними значеннями пристосованості. Зазвичай використовується у випадках, коли екземпляри мають доволі близькі значення пристосованості, що може відбуватися наприкінці ітерацій алгоритму, коли популяція в цілому виглядає доволі пристосовано і не має індивидів, що явно випадають з загальної картини.

Переходимо до **методу рулетки** (рис. 1.8). Він заключається в тому, що ймовірність проходження у індивіда пропорційна його пристосованості. В результаті маємо те, що більш пристосовані організми мають більше шансів

Таблиця 1.1 – Вибір по рейтингу

Хромосома	Значення пристосованості	Ранг
A	0.1	6
B	-2	7
C	2	5
D	5	4
E	5.5	2
F	5.1	3
G	5.9	1

бути відібрані для продовження роду.

Проте цей метод не виключає того, що екземпляри з малою пристосованістю все ще мають можливість пройти відбір. Це через те, що процес відбору полягає у тому, що організми займають сектор кола пропорційно своїй пристосованості. Обирається точка, біля якої обертається коло і той організм, який зупиниться навпроти цієї точки стає одним із батьків. Слід зазначити, що особливості цього методу (а саме можливість слабо пристосованих організмів пройти) надає ту саму різноманітність, яка надає користь ГА, а саме простір дослідження і таким чином більш оптимальне рішення.

Стохастична вибірка. Сенс данного методу дуже схожа на попередньо зазначеного методу рулетки але з деякими корективами (рис. 1.9). Він передбачає наявність не однієї фіксованої точки, а декількох, що надає більшу ймовірність для найбільш пристосованих організмів пройти відбр. Також треба зазначити, що всі батьки обираються за одну прокрутку колеса а не за декілька, як було в методі рулетки. Також особливістю подібних цьому методів є те, що вони не працюють з від'ємними показниками.

Завершує наш перелік методів селекції — **випадковий відбір** (рис. 1.10). Назва цього методу каже сама за себе, а саме: випадковим чином з усього покоління обираються організми для продовження роду.

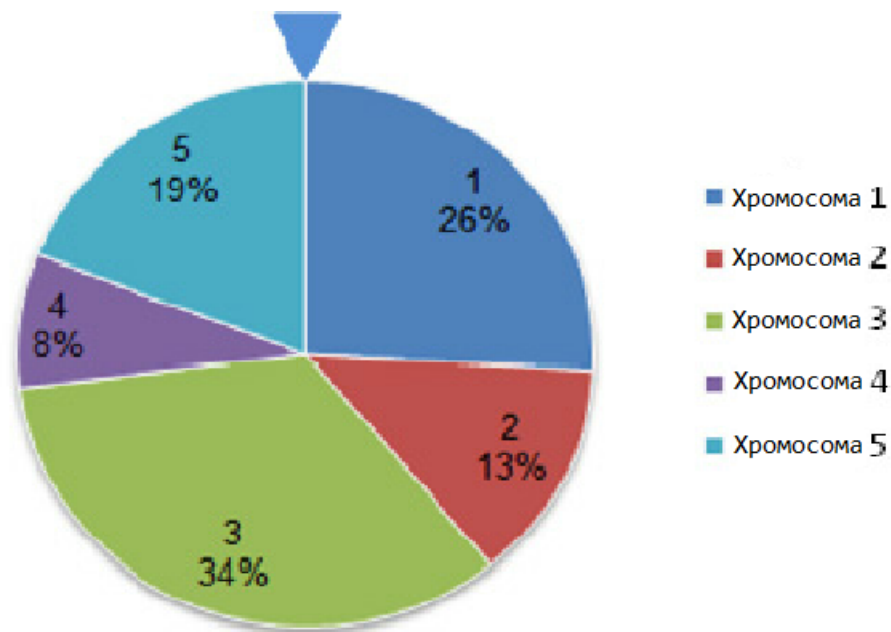


Рисунок 1.8 – Метод рулетки

Особливість методу теж є доволі очевидною: він не має жодної фільтрації у виборі організму, тому є велика ймовірність втрати великої кількості найбільш пристосованих екземплярів. Що в свою чергу відкриває можливості для індивідів з малим показником пристосованості. Хоч і зазначалось, що різноманітність є дуже важливою, але такі обставини передбачають відсутність, або уповільнений розвиток поколінь. Через такі особливості цей метод не є пріоритетним при створенні моделі на практиці.

1.5.4 Схрещування

Після вибору перспективних батьків на основі придатності та/або удачі, крок рекомбінації об'єднує їхній генетичний матеріал для створення двох нових сутностей у популяції. Це важливий крок ГА, на якому самовідтворювані сутності передають свій генотип наступному поколінню, в ідеалі зберігаючи хороші характеристики та відкидаючи погані характеристики. Ці нащадки мають спільну комбінацію генетичної

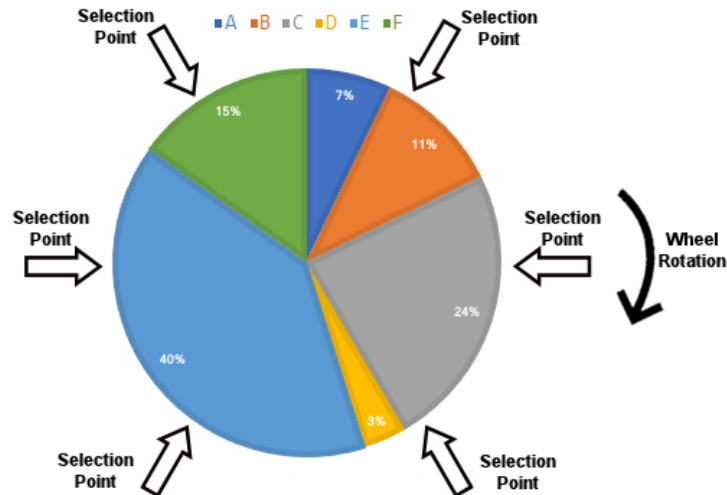


Рисунок 1.9 – Стохастична вибірка

інформації їхніх двох батьків, тобто нові фенотипи, використовуючи переваги попередніх. З ідеєю, що було відібрано двох продуктивних батьків, рекомбінація дозволяє змішувати їхні генотипи разом, щоб розвинути потенційно більш продуктивну дитину. Таким чином є шанс на те, що розвиток нашої популяції піде в гору[5].

Одноточкове схрещування. Обирається точка схрещування, якою будуть розділяться хромосоми батьків; одна половина у батьків залишається на місці, а інша - міняється місцями між батьками (рис. 1.11).

Багатоточкове хрещування. Обираються точки схрещування, які будуть розділяти хромосоми батьків на частини, після чого йде наступне: перша частина залишається, наступна міняється місцями, залишається і так далі (рис. 1.12).

Однорідне схрещування. Суть даного методу в тому, що ми не розділяємо хромосоми на частини, а проходимося по кожному гену окремо і «підкиданням монетки» вирішуємо, чи буде відбуватися обмін геном, який стоїть на позиції, яку ми розглядаємо (рис. 1.13). При цьому, якщо монетка дає нам приблизно рівні шанси на виграш одного чи іншого рішення про заміну, то ми можемо трохи змінити ймовірність на користь того чи іншого

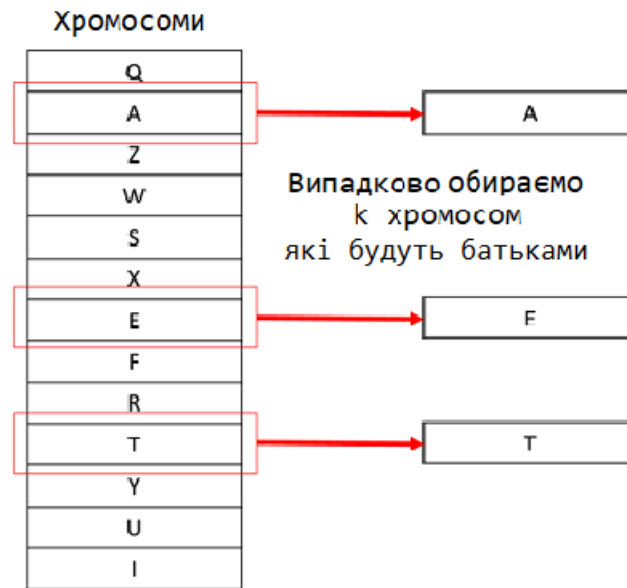


Рисунок 1.10 – Випадковий відбір

батька для того, щоб отримати від нього більше генетичного матеріалу.

Повна арифметична рекомбінація. Данний метод схрещування проходить за формулою.

$$C_i = P_i^1 \cdot \alpha + P_i^2 \cdot (1 - \alpha)$$

Де P – алеля батька

На рисунку 1.14 надано приклад для випадку, коли $\alpha = 0.5$.

Схрещування Девіса (OX1). Використовується для схрещування на основі перестановки з метою передачі інформації про відносне впорядкування нащадкам. Цей метод полягає в тому, що ми обираємо випадковим чином дві точки, якими розділимо кожного з батьків [5, 4]. Далі ми беремо середню частину від першого батька і вставляємо на ту ж саму позицію у першого нащадка. Далі, починаючи з другої розділяючої точки, у другого пращура послідовно по колу починаємо переносити гени до першого нащадка таким чином, щоб заповнити всі пусті гени (рис. 1.15). Аналогічно повторюємо таку ж послідовність дій для другого нащадка.

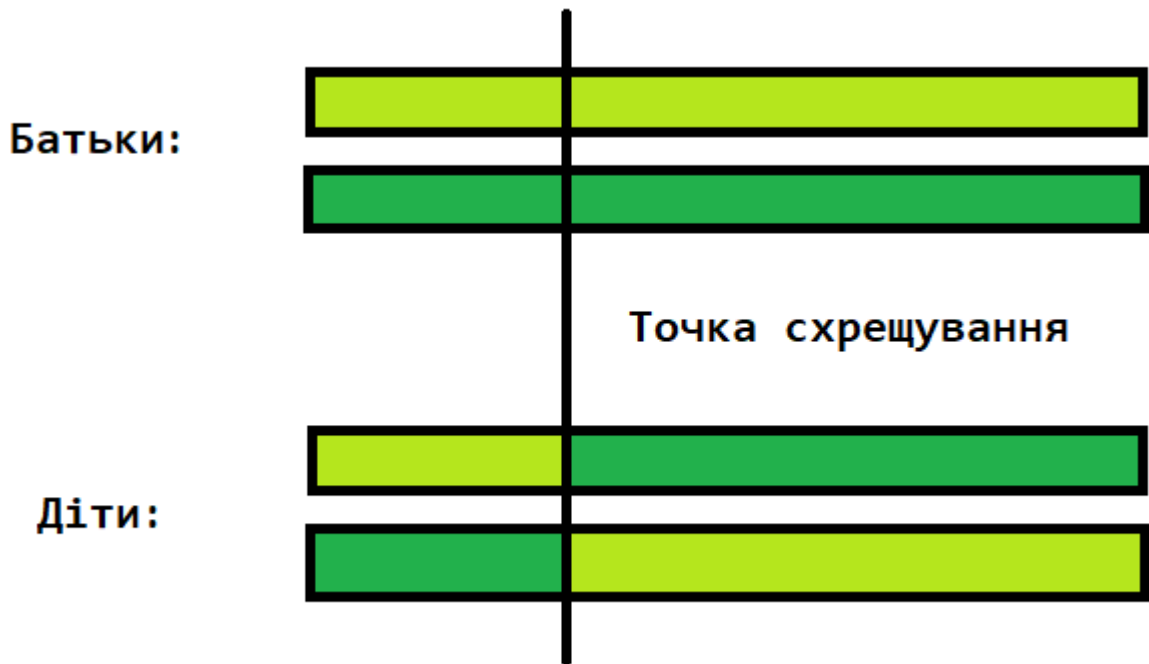


Рисунок 1.11 – Одноточкове схрещування

1.5.5 Мутація

Додатковим способом підтримки різноманітності є етап мутації. Підтримка різноманітності протягом поколінь є важливою для продуктивності ГА. ГА допускають деякі випадкові збурення генотипів, що призводять до сприятливих, нейтральних або шкідливих варіацій фенотипу. Вони вважаються важливим джерелом генетичного різноманіття і можуть передаватися наступному нащадку мutowаної сутності [5].

Особа з корисною мутацією користується перевагою придатності, що робить її більш імовірною для розмноження та передачі цієї корисної ознаки наступним поколінням. І навпаки, сутність, яка отримує шкідливу мутацію, як правило, відкидається, і ця ознака може зникнути в наступних поколіннях. Мутації також забезпечують страхування від розвитку однорідних популяцій, нездатних до подальшої еволюції.

Мутації відповідають збуренням у послідовності генотипу. Одним із поширених прикладів у ГА є мутація перевертання, або точкова мутація.

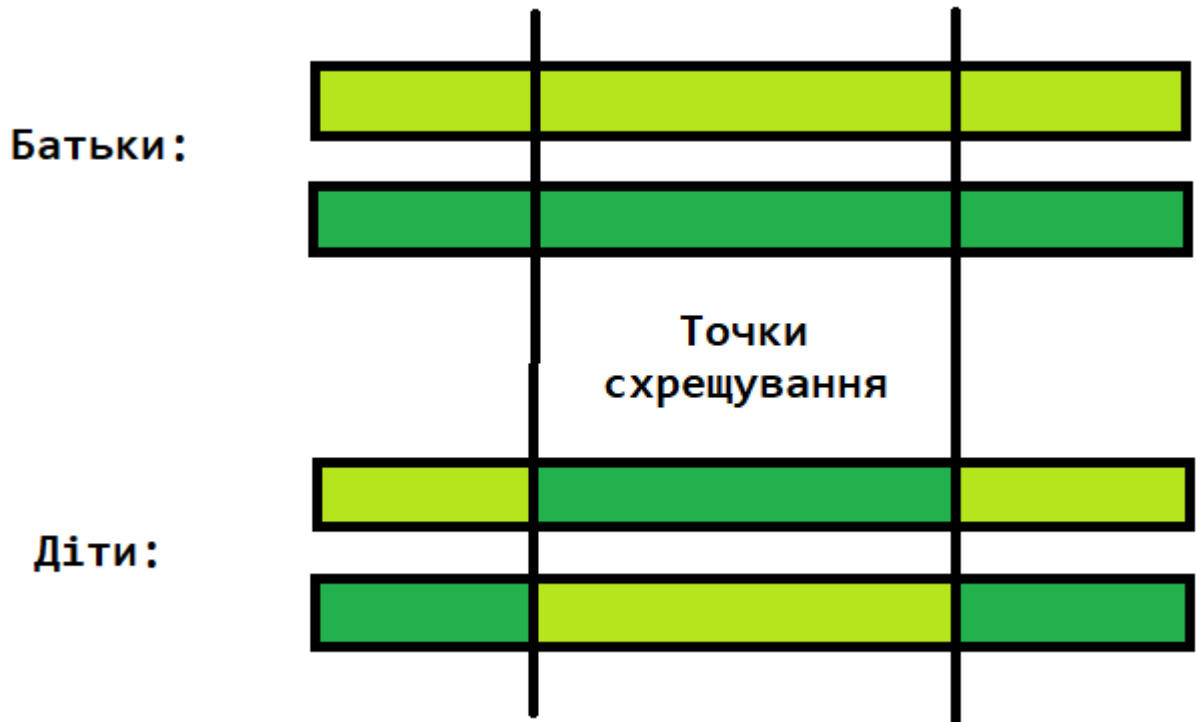


Рисунок 1.12 – Багатоточкове хрещування(випадок для двох точок)

У ДНК людини А може перетворитися на С, змінивши закодовану основу, активацію білка, що, можливо, вплине на фенотип. У двійковому рядку нуль може змінитися на одиницю або від одиниці до нуля вздовж послідовності генотипу.

Мутації відбуваються незалежно для кожного гена, незалежно від положення чи попередніх мутацій. Будь-яка алеля має незалежну та ідентичну ймовірність мутації, яка зазвичай відбувається з однаковою ймовірністю: ймовірністю мутації. Потенційно можуть постраждати всі алелі всіх сутностей у популяції, оскільки всі мутації зазвичай незалежні між алелями та між індивідами.

Визначення швидкості мутації є важливим, оскільки нульова ймовірність мутації шкодить пошуку, зменшуючи різноманітність, але занадто великий рівень мутації може порушити розвиток хороших генотипів.

У ГА існує велика різноманітність реалізацій механізмів мутації, але не так багато, як у природі. Ці оператори мутацій відрізняються залежно

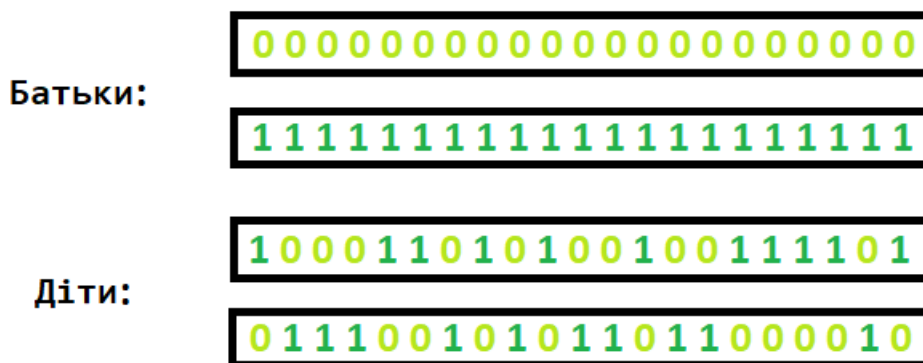


Рисунок 1.13 – Однорідне схрещування



Рисунок 1.14 – Повна арифметична рекомбінація

від проблеми, яку вирішують. Ліндгрєн, 1992 ввів мутацію дуплікації, яка збільшує довжину генотипу. Гени можуть бути перемішані або інвертовані. Деякі розділи можуть бути видалені. Деякі гени можуть піддаватися транслокації та замінюватися в іншому алелі шляхом заміни. Тож подивимось на деякі можливі методи мутацій поближче.

Біт-фліп в обраній хромосомі перевертається обрана випадковим чином кількість генів. Даний метод працює з двійковим представленням хромосом(рис. 1.16).

Випадкове скидання — це теж саме, що і біт-фліп, але використовується для цілочисельного представлення (рис. 1.17). Тому треба зазначити відмінність цих методів. Вона полягає в тому, що ми таким же чином обираємо ген або гени, які будуть замінюватись на рандомне значення з допустимих для гена.

Мутація обміну в данному методі випадковим чином обираються дві алелі, які будуть поміняні місцями (рис. 1.18).

Скрамбл мутація Випадковим чином обирається підмножина генів, які перемішуються між собою таким же випадковим чином (рис. 1.19).

Інверсивна мутація. В данній мутації обирається випадкова

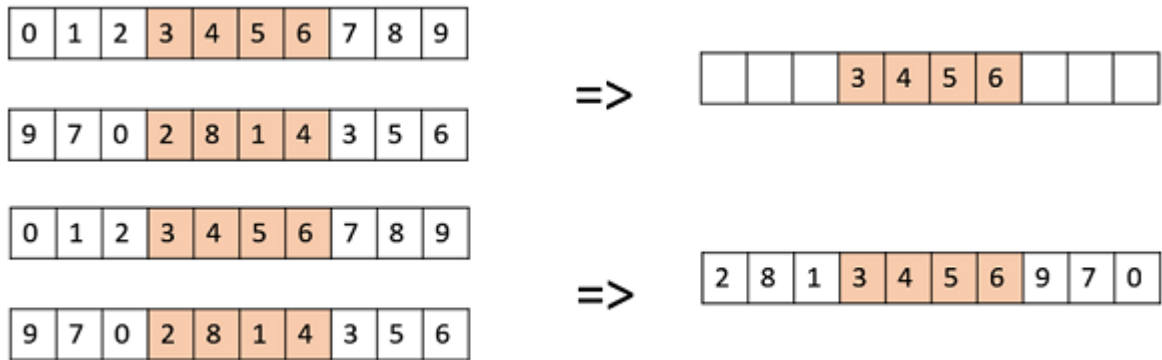


Рисунок 1.15 – Схрещування Девіса (OX1)

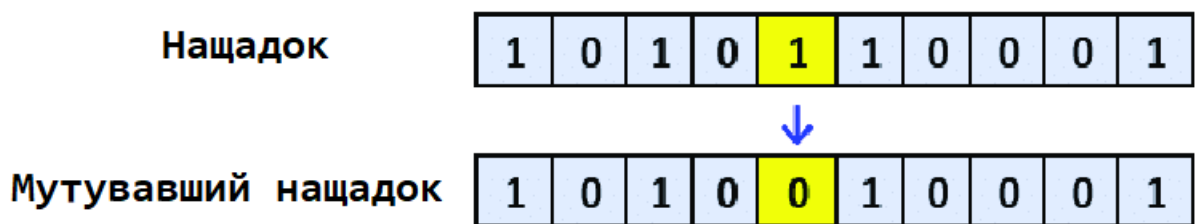


Рисунок 1.16 – Біт-фліп мутація

послідовна підмножина генів хромосоми, яка інвертується (рис. 1.20).

1.5.6 Політика відбору

Дана політика визначає те, які особи вимруть, а які продовжать жити. Вона гарантує те, що успішні індивиди продовжать існування, а також збережеться різноманітність покоління.

Елітизм гарантує те, що в наступне покоління перейдуть особи, які мають найбільшу пристосованість. Вони гарантують успішний розвиток моделі в майбутньому.

Є декілька варіантів суті цих політик. Однією з них є ситуація, коли **помирають випадкові особи** при переході на наступне покоління (рис. 1.21). Але даний метод не передбачає гарантію того, що найбільш розвинені індивіди виживуть та перейдуть далі.

Політика вибору **залежно від фітнесу** полягає в тому, що



Рисунок 1.17 – Випадкове скидання

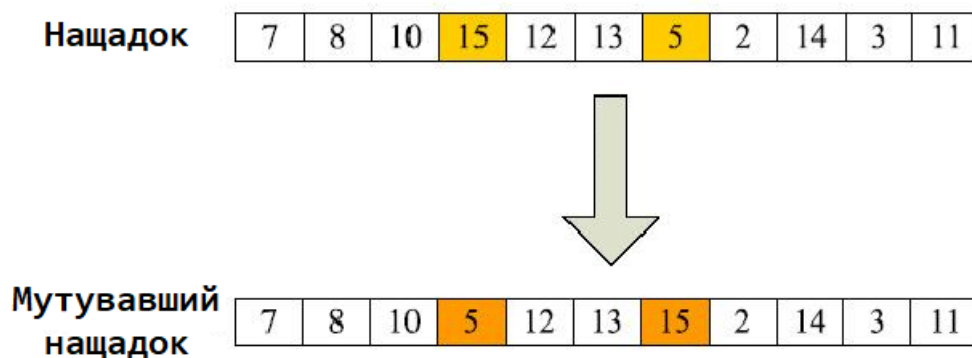


Рисунок 1.18 – Мутація обміну

обираються найменш пристосовані особини (рис. 1.22).

Наступною політикою є **вибір за віком**. Він полягає в тому, що кожній особині дозволено жити кінцеву кількість поколінь. Після того, як цей ліміт на життя вичерпано, від таких особин позбуваються незалежно від їх пристосованості (рис. 1.23).

1.6 Застосування генетичного алгоритму

ГА має велику потужність у вирішенні не малої кількості задач, тому доречним було б навести деякі приклади того, в яких випадках доцільно використати Генетичний алгоритм.

TSP. Це одна з найпоширеніших задач комбінаторної оптимізації в реальному житті, яку можна вирішити за допомогою генетичної

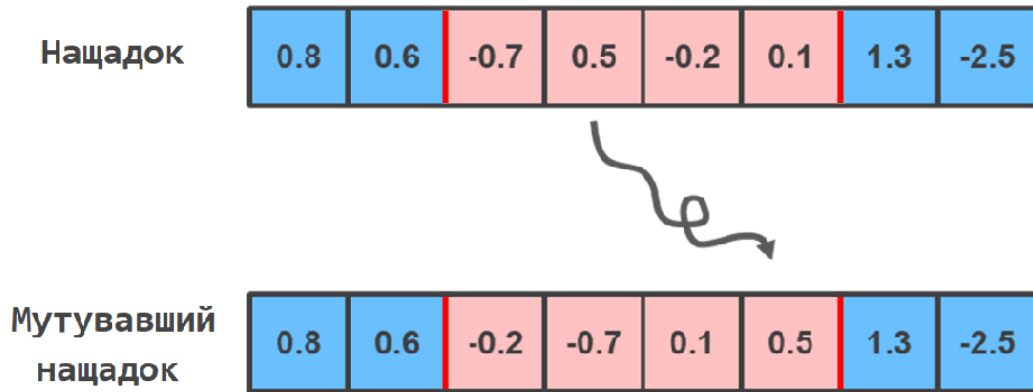


Рисунок 1.19 – Скрамбл мутація

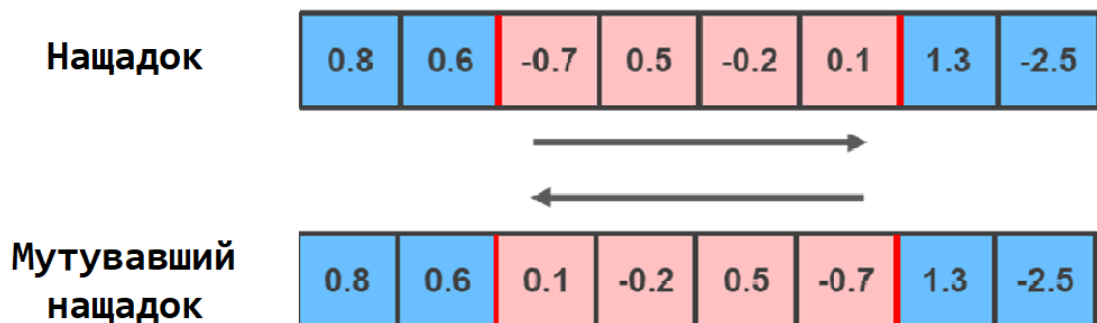


Рисунок 1.20 – Скрамбл мутація

оптимізації. Основним мотивом цієї проблеми є пошук оптимального шляху, який має пройти продавець на заданій карті з маршрутами та відстанню між двома точками. Якщо для пошуку найкращої структури маршруту використовуються генетичні алгоритми, ми не отримуємо рішення лише один раз. Після кожної ітерації ми можемо генерувати потомкові рішення, які можуть успадковувати якості батьківських рішень. TSP має безліч застосувань, таких як планування, логістика та виробництво.

Виробнича система. Одним з основних застосувань генетичної оптимізації є мінімізація функції вартості за допомогою оптимізованого набору параметрів. У виробництві ми можемо побачити різні приклади функції вартості, і знайти оптимальний набір параметрів для цієї функції можна, дотримуючись генетичної оптимізації. У багатьох випадках ми

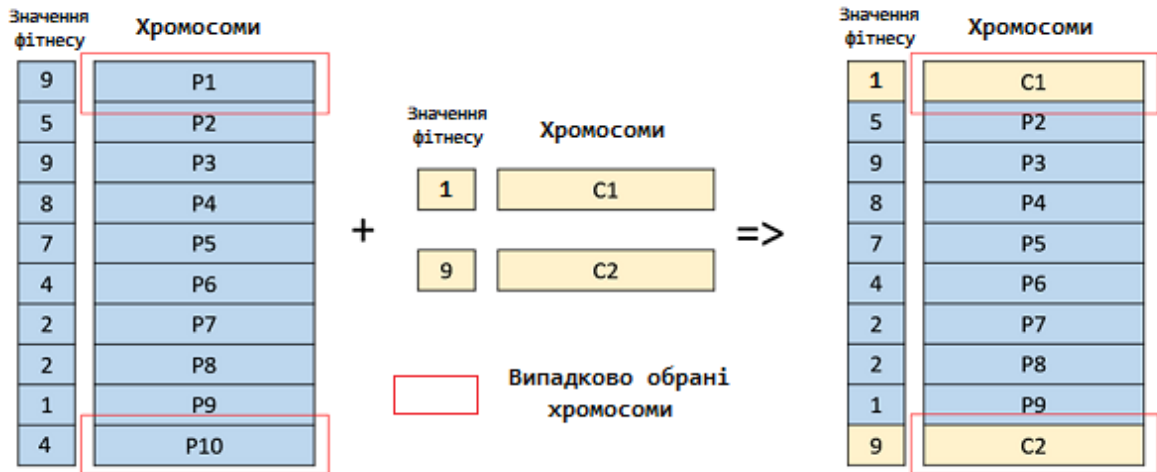


Рисунок 1.21 – Політика випадкової особи

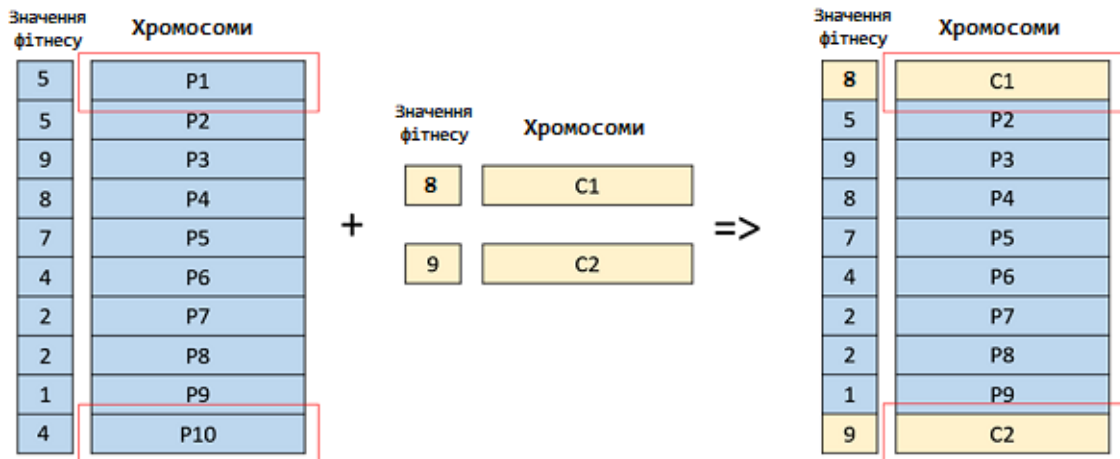


Рисунок 1.22 – Політика залежно від фітнесу

можемо знайти застосування генетичної оптимізації у виробництві продукції (варіювання виробничих параметрів або порівняння різного комплектування обладнання). Основним мотивом застосування генетичної оптимізації є досягнення оптимального плану виробництва з урахуванням динамічних умов, таких як запаси, потужність або якість матеріалу.

Обробка зображення. Існують різноманітні роботи та дослідження, які показують випадки використання генетичної оптимізації в різних задачах обробки зображень. Одним із основних завдань, пов'язаних з генетичним підходом до обробки зображень, є сегментація зображення. Хоча ці генетичні оптимізації можна використовувати в

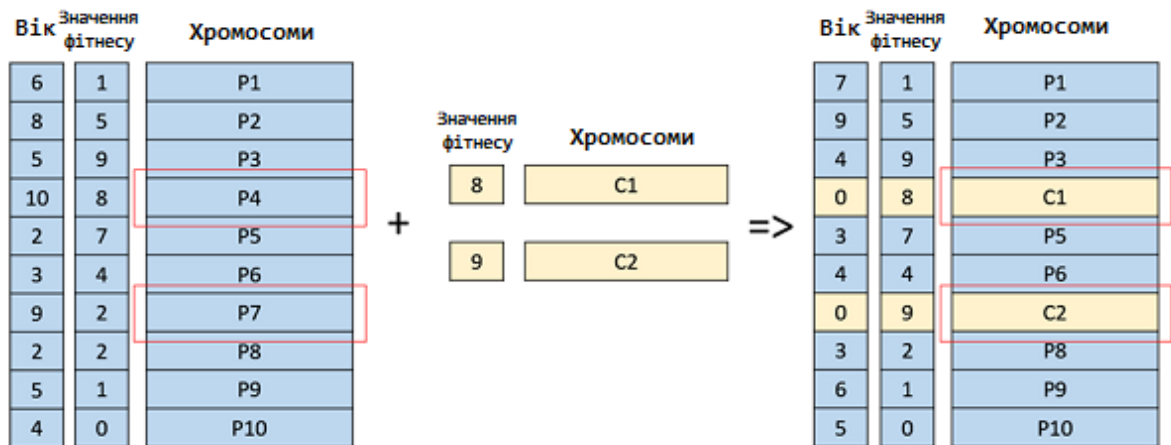


Рисунок 1.23 – Політика вибору за віком

різних областях аналізу зображень для вирішення складних завдань оптимізації. Використання генетичної оптимізації в комплексі з методами сегментації зображень може зробити всю процедуру проблемою оптимізації.

Нейронні мережі. Нейронні мережі в машинному навчанні є однією з найбільших областей, де для оптимізації використовувалися генетичні алгоритми. Одним з найпростіших прикладів використання генетичної оптимізації в нейронних мережах є пошук оптимального набору параметрів для нейронної мережі. Замість них ми можемо знайти використання генетичних алгоритмів в оптимізації конвеєрів нейронних мереж, успадкування якостей нейронів тощо.

Медицина У медичній науці ми можемо знайти багато прикладів використання генетичної оптимізації. Генетичний алгоритм може бути застосовано при генерації препарату для діагностики будь-якого захворювання в організмі. У різних прикладах ми бачимо використання генетичної оптимізації в прогнозованому аналізі, як-от передбачення структури РНК, прогнозування оперонів, прогнозування білка тощо. Також є деякі випадки використання генетичної оптимізації у вирівнюванні процесів, наприклад, вирівнювання множинних послідовностей біоінформатики, профілювання експресії генів. аналіз, згортання білка тощо.

Висновки до розділу 1

Даний розділ продемонстрував нам особливості роботи та структури генетичного та еволюційного алгоритмів. Стало зрозуміло, що вони шукали натхнення в еволюційній теорії Дарвіна, яка передбачала той факт, що організми прагнуть досягти кращої пристосованості залежно від середовища, в якому вони знаходяться. Це відкриває можливості для кращої виживаємості та ще більшого розвитку наступних поколінь.

Всі основні процеси існування реального покоління аналогічно, але більш стисло, що пов'язане з кількістю параметрів в реальному середовищі, продемонстровані і в комп'ютерній моделюванні. Починаючи з можливих методів відбору кращих продовжувачів роду, до мутацій деяких індивідів для зберігання різноманітності покоління, що в подальшому може призвести до кращої адаптації до середовища. «Сильніші та пристосованіші виживають і відкривають двері для подальшого розвитку».

2 НЕЙРОННІ МЕРЕЖІ

Нейронні мережі — обчислювальні системи, натхненні біологічними нейронними мережами. Вони імітують роботу та поведінку мозку для вирішення складних задач, які залежать від даних. Вхідні дані обробляються за допомогою різних шарів штучних нейронів, об'єднаних разом для отримання бажаного результату.

Вже існують численні методи обчислювального інтелекту, що брали натхнення з біологічних нервових систем, природного відбору, поведінки комах, тощо.

Незважаючи на обмеження, пов'язані з кожним із цих методів, вони є надійними і застосовувалися у вирішенні реальних життєвих проблем у сферах науки, техніки, бізнесу та торгівлі. Гібридизація двох або більше з цих методів усуває обмеження, які можуть виникнути, і веде до кращого рішення. В результаті гібридизації в наш час розробляється багато ефективних і корисних систем для вирішення великої кількості задач[6].

Останні дослідження, що гібридизували методи обчислювального інтелекту для пошуку оптимальних або близьких до оптимальних рішень, включають: генетичний алгоритм, оптимізацію роя частинок та оптимізацію гібридизації колоній мурах; інтеграція нечіткої логіки та експертної системи; комбінація нейронних мереж і нечіткої логіки; гібридизація генетичного алгоритму та оптимізації роя частинок; комбінація механізму нечіткого висновку, онтології та нечіткої мови розмітки; гібридизація машини опорних векторів і оптимізація роя частинок. Звичайно, треба відмітити, що такі дослідження не обмежуються вищеназваними темами. Проте нейронні мережі і генетичний алгоритм вважаються найбільш надійними та перспективними методами обчислювального інтелекту.

2.1 Структура нейронних мереж

Нейронна мережа — деяка математична модель, яка працює, імітуючи людський мозок. Оброблюючи вхідні в нейрони дані вони подають їх далі через один або декілька виходів. А далі дані потрапляють до прихованого шару, де зустрічаються з іншими даними, які передані з інших виходів для остаточного прийняття рішення та відання результату.

Нейрони — вузол штучної нейронної мережі, що є спрощеною моделлю природного нейрона (рис. 2.1).

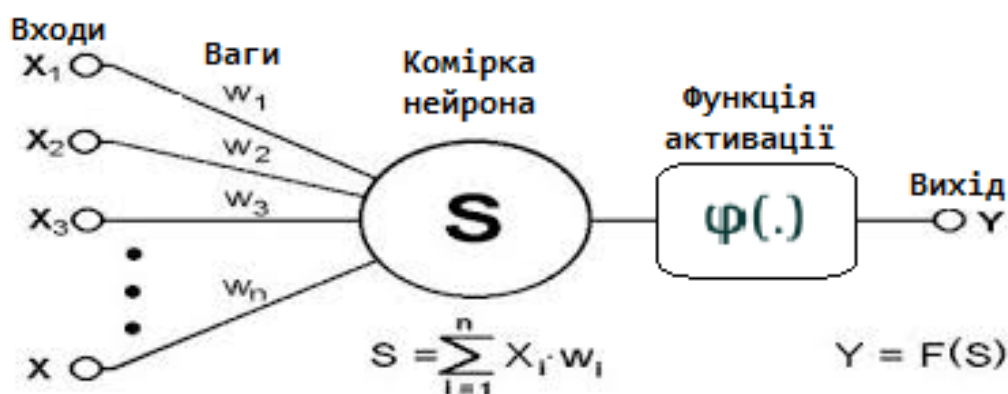


Рисунок 2.1 – Нейрон

Біас(зміщення) — роль полягає в зміщенні значення, створеного функцією активації. Його роль подібна до ролі константи в лінійній функції.

Функції активації — вносить нелінійність у роботу нейронів, щоб враховувати різну лінійність з входами. Без цього вихід був би просто лінійною комбінацією вхідних значень і не міг би внести нелінійність у мережу.

Ваги зв'язків — їх головна функція полягає в тому, щоб надати значення тим характеристикам, які більше сприяють навчанню. Це робиться шляхом введення скалярного множення між вхідним значенням і ваговою матрицею. Наприклад, негативне слово вплине на рішення

моделі аналізу настроїв більше, ніж пара нейтральних слів.

Вхід — це набір ознак, які вводяться в модель для процесу навчання. Наприклад, вхідними параметрами для виявлення об'єктів може бути масив значень пікселів, що належать до зображення.

Коли кілька нейронів об'єднані в один ряд, вони утворюють шар. А кілька шарів, зібраних один біля одного, називаються багатошаровою нейронною мережею.

Вхідний шар — дані, які ми надаємо в модель, завантажуються на вхідний рівень із зовнішніх джерел. Це єдиний видимий шар, який передає повну інформацію із зовнішнього світу.

Приховані шари — це те, що робить глибоке навчання таким, яким воно є сьогодні. Вони є проміжними шарами, які виконують усі обчислення та витягують особливості з даних.

Може існувати кілька взаємопов'язаних прихованих шарів, які враховують пошук різних прихованих об'єктів у даних. Наприклад, при обробці зображень перші приховані шари відповідають за елементи вищого рівня, такі як краї, форми або межі. З іншого боку, пізніші приховані шари виконують більш складні завдання, такі як ідентифікація цілісних об'єктів (автомобіль, будівля, людина).

Вихідні шари — вихідний рівень приймає оброблені дані від попередніх прихованих шарів і надходить до остаточного прогнозу на основі знань моделі. Це найважливіший шар, на якому ми отримуємо кінцевий результат.

У випадку моделей класифікації/регресії вихідний шар зазвичай має один вузол. Однак це повністю залежить від конкретної проблеми і залежить від того, як була побудована модель.

Дизайн мережі є доволі гнучким, тож його можна підлаштувати будь-яким необхідним чином. В залежності від задачі, яку необхідно вирішити, кількість нейронів у кожному з вищеперерахованих шарів може змінюватись.

Зокрема треба приділити достатню увагу швидкості навчання. Адже

від цього моменту залежить чутливість моделі, а значить і результат. При дуже швидкому навчанні модель буде різко реагувати на нові параметри і кардинально змінювати результат, що є не добре. З іншої сторони при занадто повільному навчанні, модель погано реагує на параметри, в ітозі рішення не сильно краще.

2.1.1 Функція втрат

Модель машинного навчання, така як нейронна мережа, намагається вивчити розподіл ймовірностей, що лежить в основі даних спостережень. У машинному навчанні зазвичай використовують статистичну систему оцінки максимальної правдоподібності як основу для побудови моделі. Це означає, що ми намагаємося знайти набір параметрів і попередній розподіл ймовірностей, наприклад, нормальний розподіл, щоб побудувати модель, яка представляє розподіл за нашими даними[8].

У контексті алгоритму оптимізації функція, яка використовується для оцінки рішення(тобто набору ваг), називається цільовою функцією.

Ми можемо прагнути максимізувати або мінімізувати цільову функцію, що означає, що ми шукаємо рішення, яке має найвищий або найнижчий бал відповідно.

Як правило, з нейронними мережами ми прагнемо мінімізувати помилку. Таким чином, цільову функцію часто називають функцією витрат або функцією втрат, а значення, обчислене функцією втрат, називають просто «втратою».

Функція втрат виконує важливу роботу, оскільки вона повинна точно зводити всі аспекти моделі в єдине число таким чином, щоб покращення цього числа були ознакою кращої моделі.

При розрахунку похибки моделі в процесі оптимізації необхідно вибрати функцію втрат.

Це може бути складною проблемою, оскільки функція повинна охоплювати властивості проблеми та бути мотивована проблемами, які

важливі для проекту та зацікавлених сторін.

Важливо, що вибір функції втрат безпосередньо пов'язаний з функцією активації, яка використовується у вихідному шарі вашої нейронної мережі. Ці два елементи конструкції пов'язані між собою.

Функція втрат у нейронній мережі кількісно визначає різницю між очікуваним результатом і результатом, отриманим моделлю машинного навчання. З функції втрат ми можемо отримати градієнти, які використовуються для оновлення ваг. Середнє значення за всіма втратами становить вартість. Щоб зрозуміти, як градієнти обчислюються та використовуються для оновлення ваг, треба зазначити таке поняття, як зворотне поширення з градієнтним спуском.

Зворотне поширення є сутністю навчання нейронної мережі. Це метод точного налаштування ваг нейронної мережі на основі частоти помилок, отриманої в попередню епоху. Правильне налаштування ваг дозволяє знизити частоту помилок і зробити модель надійною за рахунок збільшення її узагальнення.

2.1.2 Функція активації

Функція активації вирішує, чи слід активувати нейрон, чи ні. Це означає, що вона вирішуватиме, чи є вхід нейрона в мережу важливим в процесі передбачення, чи ні, за допомогою простіших математичних операцій.

Функції активації перетворює підсумований зважений вхід з вузла у вихідне значення, яке буде передано до наступного прихованого шару або виведено як результат.

Функції активації вводять додатковий крок на кожному рівні під час прямого поширення, але його обчислення того варте. Ось чому:

Припустимо, у нас нейронна мережа працює без функцій активації. У цьому випадку кожен нейрон буде виконувати лише лінійне перетворення на входах, використовуючи вагові коефіцієнти та зміщення. Це тому, що

не має значення, скільки прихованих шарів ми приєднаємо до нейронної мережі. Всі шари будуть вести себе однаково, оскільки композиція двох лінійних функцій сама є лінійною функцією.

Хоча нейронна мережа стає простішою, вивчення будь-якої складної задачі неможливо, і наша модель була б просто моделлю лінійної регресії.

Звичайно, функція активації не є єдиною для всіх випадків і задач, з якими ми можемо зіткнутися, тому подивимось на деякі їх типи та особливості.

Функція бінарного кроку залежить від порогового значення, яке визначає, чи слід активувати нейрон чи ні.

Вхід, що надходить на функцію активації, порівнюється з певним порогом; якщо вхід більше, нейрон активується, в іншому випадку він деактивується, тобто його вихід не передається на наступний прихований шар (рис. 2.2).

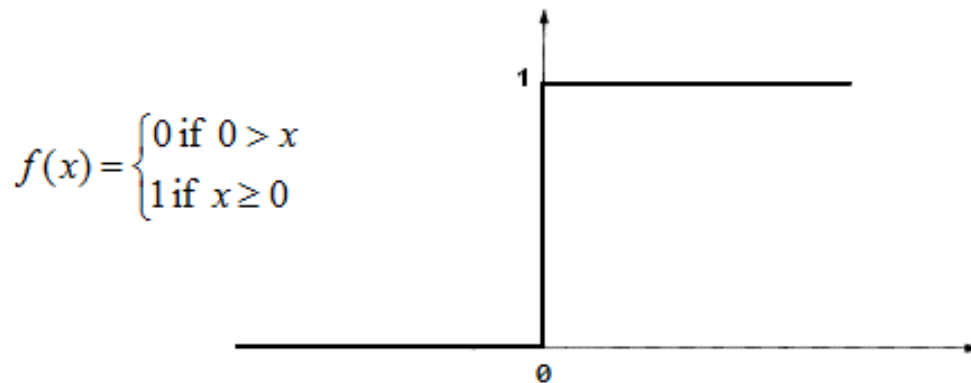


Рисунок 2.2 – Функція бінарного кроку

Ця функція має деякі обмеження:

Вона не може надавати багатозначні результати — наприклад, її не можна використовувати для задач класифікації з кількома класами.

Гradient ступінчастої функції дорівнює нулю, що спричиняє перешкоду в процесі зворотного поширення.

Лінійна функція активації, також відома як «без активації» або

«функція ідентичності»: активація, пропорційна введеним інформації (рис. 2.3).

Функція не робить нічого зі зваженою сумою введених даних, вона просто видає значення, яке їй було надано.

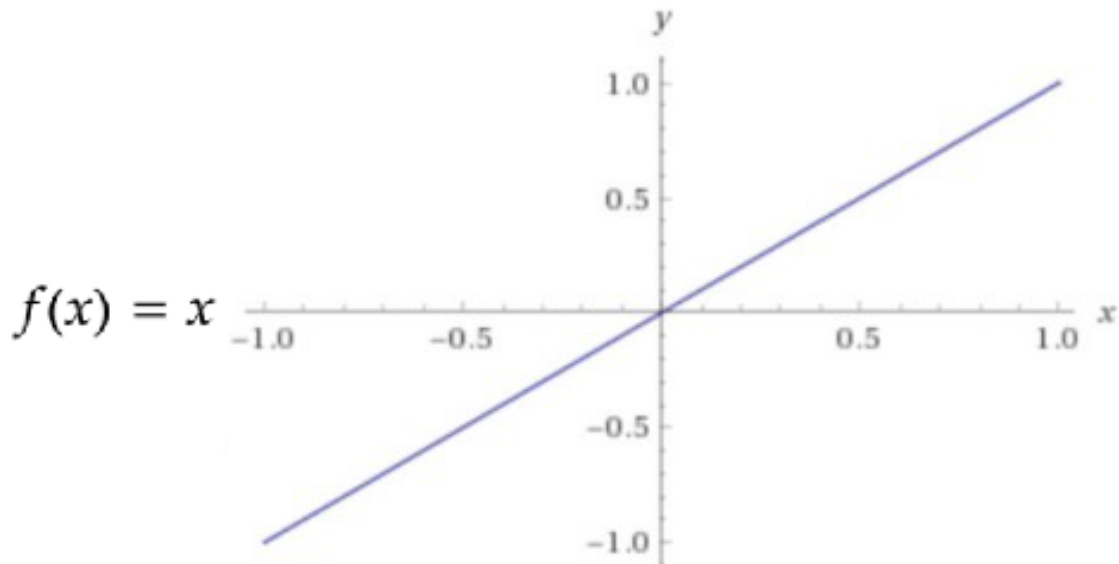


Рисунок 2.3 – Лінійна функція активації

Обмеження: Не можна використовувати зворотнє поширення, оскільки похідна функції є константою і не має відношення до входу.

Усі шари нейронної мережі згорнуться в один, якщо буде використана функція лінійної активації. Незалежно від кількості шарів у нейронній мережі, останній шар все одно буде лінійною функцією першого шару. Таким чином, по суті, функція лінійної активації перетворює нейронну мережу лише в один шар.

Через обмежену потужність це не дозволяє моделі створювати складні відображення між входами та виходами мережі.

Нелінійні функції активації вирішують такі обмеження лінійних функцій активації:

1) Вони дозволяють зворотнє поширення, тому що тепер похідна функції буде пов'язана з входом, і можна повернутися назад і зрозуміти,

які ваги у вхідних нейронах можуть забезпечити кращий прогноз.

2) Вони дозволяють об'єднувати кілька шарів нейронів, оскільки на виході тепер буде нелінійна комбінація вхідних даних, пропущених через кілька шарів. Будь-який вихід може бути представлений у вигляді функціонального обчислення в нейронній мережі [9].

Вони дозволяють об'єднувати кілька шарів нейронів, оскільки на виході тепер буде нелінійна комбінація вхідних даних, пропущених через кілька шарів. Будь-який вихід може бути представлений у вигляді функціонального обчислення в нейронній мережі.

Тепер подивимось на деякі нелінійні функції активації:

Сигмоїда або Логістична функція активації. Ця функція приймає будь-яке реальне значення як вхідні дані та виводить значення в діапазоні від 0 до 1 (рис. 2.4).

Чим більше вхід (більш позитивний), тим ближче вихідне значення буде до 1,0, тоді як чим менше вхід (більш негативний), тим ближче вихід буде до 0,0.

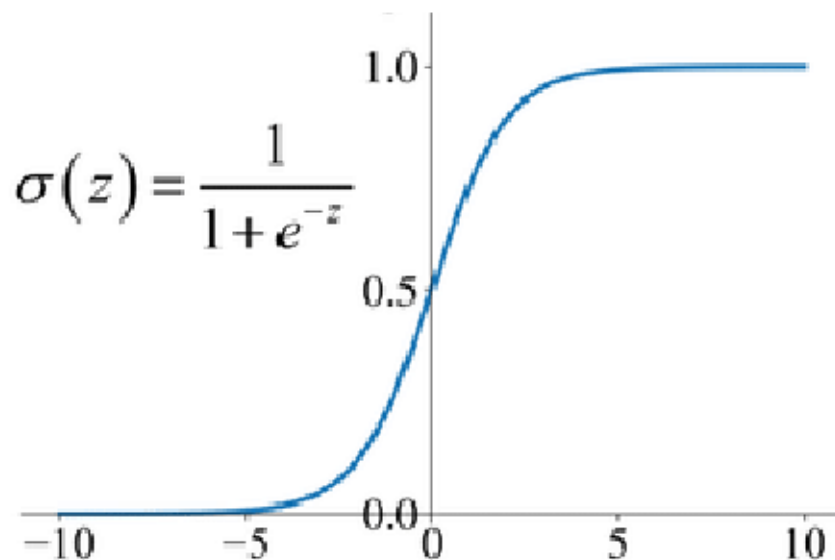


Рисунок 2.4 – Сигмоїда

Переваги використання:

1) Вона зазвичай використовується для моделей, де ми повинні передбачити ймовірність як результат. Оскільки ймовірність будь-чого існує лише між діапазоном від 0 до 1, сигмоїда є правильним вибором через її діапазон.

2) Функція диференційована і забезпечує плавний градієнт, тобто запобігає стрибкам вихідних значень. Це представлено S-подібною формою сигмовидної функції активації.

Функція Tanh дуже схожа на функцію сигмоїдної/логістичної активації і навіть має ту саму S-подібну форму з різницею в діапазоні вихідних даних від -1 до 1. У Tanh, чим більше вхід (більш позитивний), тим ближче вихідне значення буде до 1,0, тоді як чим менше вхід (більш негативний), тим ближче вихід буде до -1,0 (рис. 2.5).

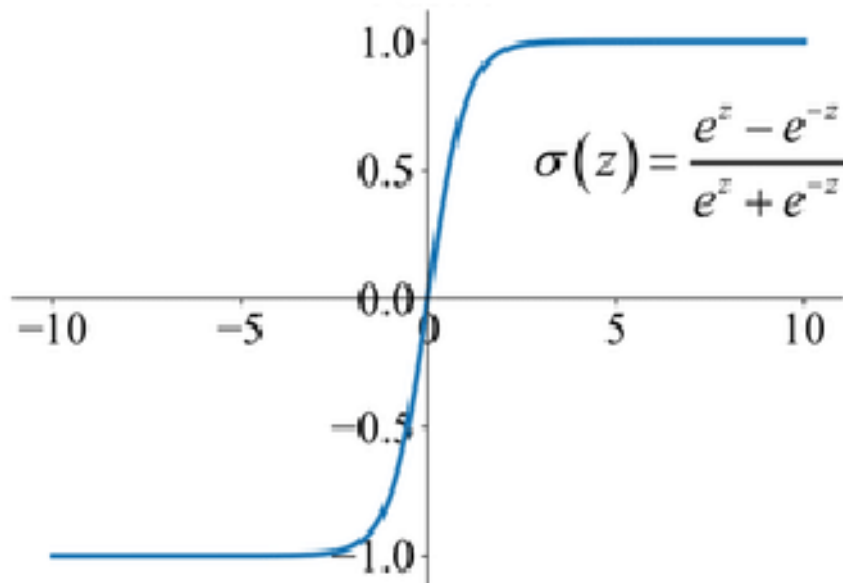


Рисунок 2.5 – Функція Tanh

Як перевагу використання цієї функції активації можна відмітити, що вихід Tanh має нульовий центр. Отже, ми можемо легко відобразити вихідні значення як сильно негативні, нейтральні або сильно позитивні.

Зазвичай використовується в прихованих шарах нейронної мережі, оскільки його значення лежать від -1 до 1; тому середнє значення для

прихованого шару виявляється рівним 0 або дуже близьким до нього. Це допомагає центрувати дані та значно полегшує навчання для наступного рівня.

ReLU(Rectified Linear Unit).

ReLU має похідну функцію і дозволяє здійснювати зворотне поширення, водночас роблячи її обчислювально ефективною (рис. 2.6).

Головна проблема тут полягає в тому, що функція ReLU не активує всі нейрони одночасно.

Нейрони будуть дезактивовані, лише якщо вихідні дані лінійного перетворення менше 0.

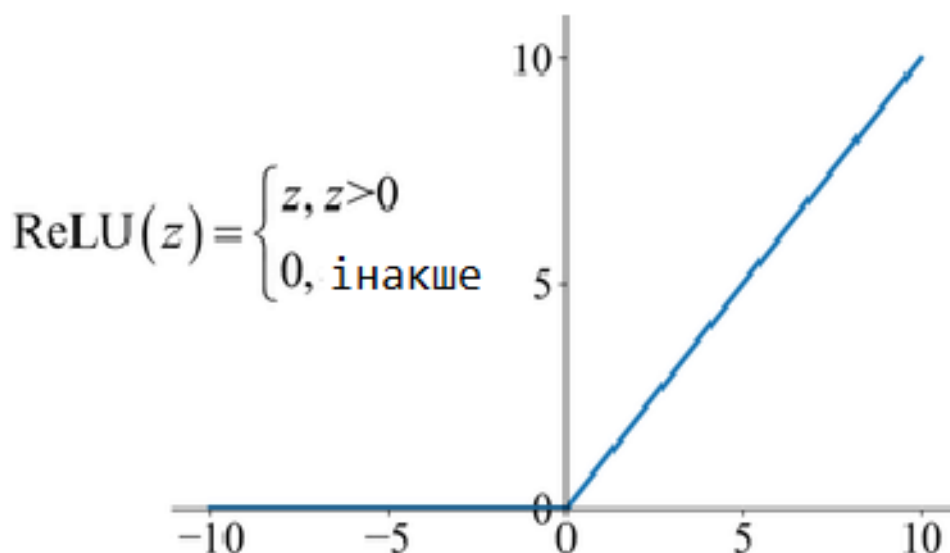


Рисунок 2.6 – Функція ReLU

Переваги використання ReLU як функції активації полягають у наступному:

Оскільки активується лише певна кількість нейронів, функція ReLU є набагато ефективнішою в обчислювальному відношенні порівняно з функціями сигмоїди і Tanh.

ReLU прискорює зближення градієнтного спуску до глобального мінімуму функції втрат завдяки своїй лінійній властивості, що не має

насичення.

Висновки до розділу 2

Нейронні мережі — набір алгоритмів, який створено на основі людського мозку для того, щоб знаходити схожість. Вони обробляють дані завдяки їх особистому сприйняттю того, що їм подається на вхід. Навчаються завдяки початковим даним та роблять висновки.

Вони демонструють доволі гарні результати при обробці великої кількості інформації. Це відбувається завдяки їх можливості пристосовуватись до кожної ситуації окремо.

3 АНАЛІЗ ТА РЕАЛІЗАЦІЯ МОДЕЛІ

В моделі буде реалізовано розвиток простих організмів. За для цього використовуємо генетичний алгоритм. Також застосуємо нейронні мережі для реалізації "мізків" наших організмів[7].

Для того, щоб обрати кращий сценарій розвитку організмів, розглянемо поведінку системи в залежності від змін, які ми вносимо стосовно обраних для спостереження параметрів.

3.1 Реалізація моделі

Модель, яка буде розглядатись має, деякі особливості, які треба зазначити. Тож, ми маємо поле, на якому будуть розвиватись організми. Також на цьому полі буде їжа для організмів.

3.1.1 Поле

Поле складається з 2 різних об'єктів: їжі та організмів. Їжа нерухома і не має «мізків». Кожен шматочок їжі має статичну харчову цінність. Після того, як їжа була з'їдена, вона просто переміщується на нове місце на полі. Організми, з іншого боку, здатні приймати рішення про інших індивідів та їжу за допомогою нейронних мереж. Також вони повільно зменшуються в розмірі, коли не харчуються (або поїдаючи інші організми, або їжу).

Поле не має кордонів: коли об'єкт потрапляє на ліву стіну, він «телепортується» на праву стіну. Під час тестів з полем, в якого наявні кордони, вся популяція геномів мала тенденцію прилипнути до однієї зі стін, не перетворюючись на більш гнучку популяцію. Однак наявність стін без меж супроводжується проблемою, яку ще не вирішено: геноми, які знаходяться, наприклад, біля лівої стінки, не виявлять ворогів, які

знаходяться близько до правої стінки, навіть якщо відстань між організмами може бути дуже малою.

3.1.2 Організми

Геноми не можуть прискорюватися навмисно, вони відразу пристосовуються до швидкості залежно від їх маси, яку дає їх мозок. Вони можуть їсти інші організми, лише коли вони на 10% більші, і вони можуть вільно переміщатися крізь інші краплі, розміри яких менше ніж на 10%. Кожен геном бачитиме лише 3 найближчих організми і 3 найближчі кульки їжі в певному радіусі.

Вихідними даними "мозку" організмів є напрямок та швидкість
Деякі параметри конфігурації організмів:

- 1) Організм має бути на 10% більшим за іншого генома, щоб з'їсти його
- 2) Мінімальна площа організма становить 400 пікселів
- 3) Максимальна площа організма становить 10000 пікселів
- 4) Радіус виявлення потенційних ворогів та їжі становить 150 пікселів
- 5) Організм може бачити до 3 інших індивідів у своєму радіусі виявлення
- 6) Організм може побачити до 3 куль їжі в радіусі виявлення
- 7) Максимальна швидкість організма становить 3 пікселі/кадр
- 8) Мінімальна швидкість організма – 0,6 пікселів/кадр
- 9) Кожен кадр індивид втрачає 0,2% своєї маси

3.1.3 Генетичний алгоритм та нейронна мережа

Генетичний алгоритм є ядром штучного інтелекту. У першому кадрі певна кількість організмів евристично ініціалізується з нейронною мережею як мозком. Мозок представляє популяцію покоління. Потім ці

мізки розвиваються, поміщаючи все населення в єдине поле і дозволяючи їм змагатися один з одним. Найпридатніші мізки переміщуються в наступне покоління, менш пристосовані мізки мають високі шанси бути видалені.

Оцінити геноми досить легко: в залежності від їх розміру визначається значення пристосованості. При інших експериментах з системами підрахунку, багато з них стимулювали маленьких організмів самоликвідуватись, якщо їхній рахунок був занадто низьким протягом певного часу. Тобто селекція відбувається за допомогою відбору по рейтингу, а в нашому випадку — відбір по значенню пристосованості.

Стосовно схрещування, слід відмітити, що далі буде відбуватись аналіз, який буде спрямований на визначення кращого варіанту цього етапу, тож про це далі.

Методи мутації теж належать до одного з пунктів аналізу, тож більше про це на відповідному етапі роботи.

Деякі початкові конфігурації:

- 1) Оцінка виконується для 500 кадрів
- 2) Частота мутації становить 0,3
- 3) Елітизм становить 10%
- 4) Кожен геном починається з 0 прихованих вузлів

Нейронна мережа, яка буде використовуватись - рандомна, або її ще можна назвати "рідка".Ця мережа почнеться з заданого пулу вузлів, а потім створить випадкові з'єднання між ними.

Успішність розвитку нашого покоління буде визначатись завдяки середньому значенню пристосованості(фітнесу/маси) — AVG.

3.2 Аналітична частина

Тож тепер настав найцікавіший етап роботи. Необхідно визначитись з параметрами/методами, вплив на покоління яких, ми будемо аналізувати. Були обрані такі параметри: частота мутації, методи схрещування, методи

мутації, кількість прихованих нод.

3.2.1 Кількість нод у прихованому шарі

Прихований шар - група нод, що не є у вхідному та вихідному шарі. Він слугує для додаткових розрахунків, та покращення навчання. Але існують деякі нюанси: Якщо нод буде забагато, то відбудеться перенавчання. З іншої сторони монети: якщо нод замало, то відбувається нестача ресурсів для вирішення поставленої задачі, адже її складність може бути не такою уж і малою.

Розглянемо декілька випадків, коли кількість нод буде рівною 0, 5, 10, 15, 20, 25 відповідно.

При проведенні модуляцій з різними параметрами можна помітити, що організми з малою кількістю прихованих нейронів мають меншу кількість обчислень, значить їх навчання протікає швидше, вони мають

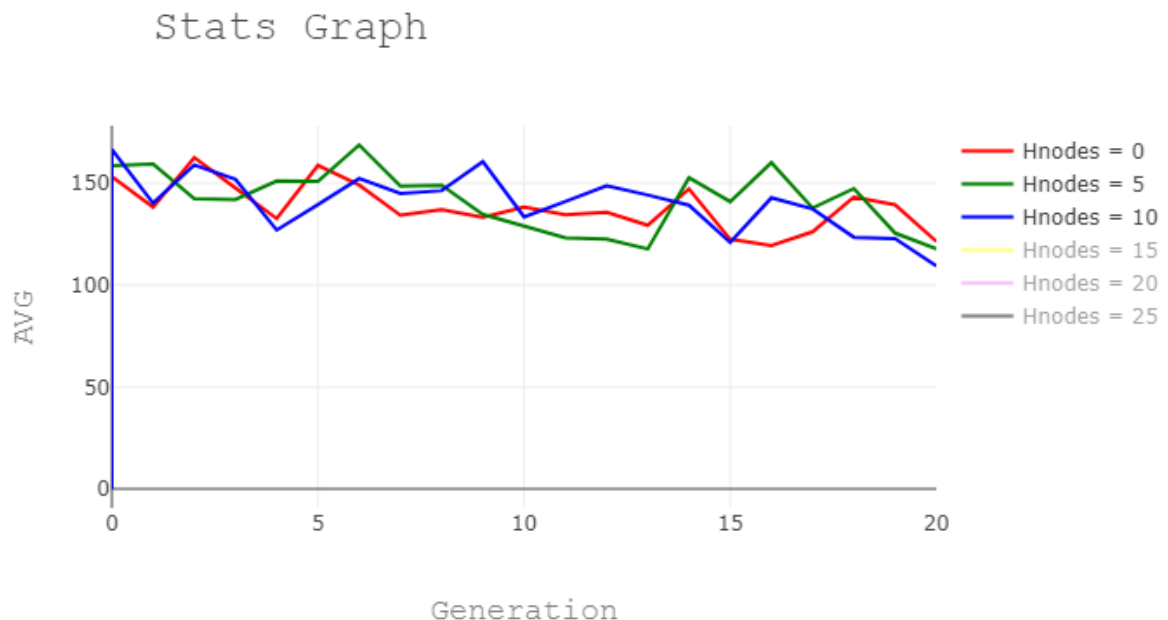


Рисунок 3.1 – Середнє значення пристосованості для скритих нейронів кількості 0-5-10

найменшу кількість ваг у нейронній мережі, але при цьому якість цього навчання не є високою і має багато просадок.

На графіку 3.1 представлено порівняння моделювань коли кількість нод рівна 0-5-10. Як можна помітити, найліпший результат показала модель з 5-ма нейронами у прихованому шарі.

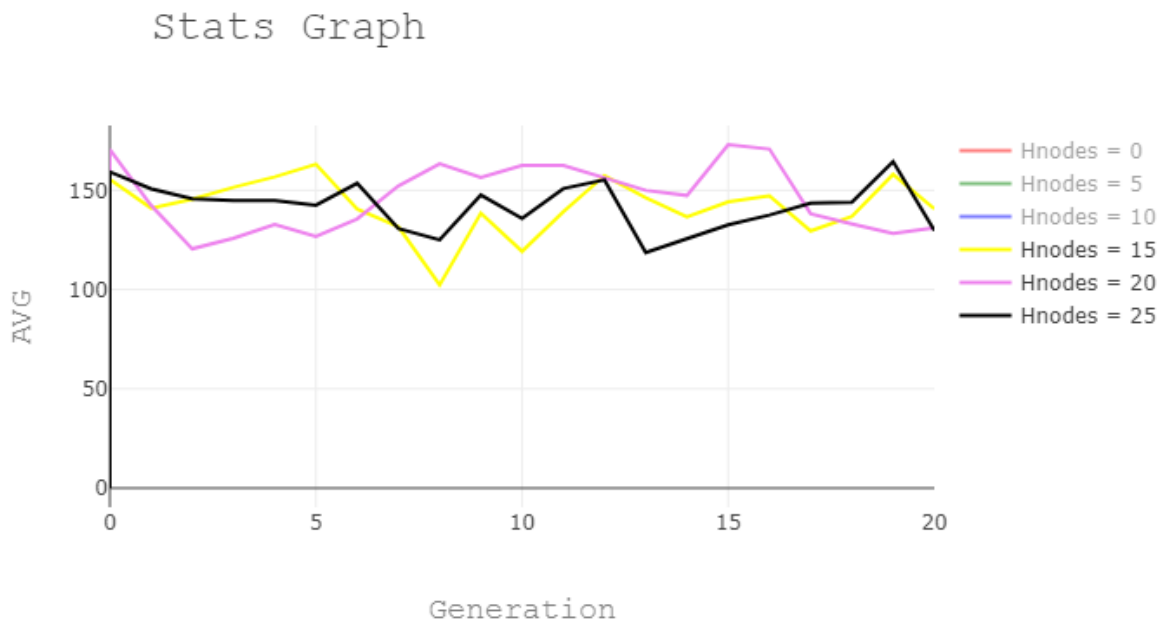


Рисунок 3.2 – Середнє значення пристосованості для скритих нейронів кількості 15-20-25

На графіку 3.2 порівнюються моделювання з 15-20-25 нод у прихованому шарі. Кращою виявилась симуляція з 20-ма нейронами.

Тепер порівняємо обидва результати. Як видно на графіку 3.3, оптимальною кількістю нейронів у прихованому шарі буде 20.

В таблиці А.1 можна побачити числові значення середньої пристосованості за поколіннями при кількості прихованих нейронів 0 - 5 - 15 - 20 - 25. За нею можна більш детально відстежити хід розвитку популяцій на кожному поколінні.

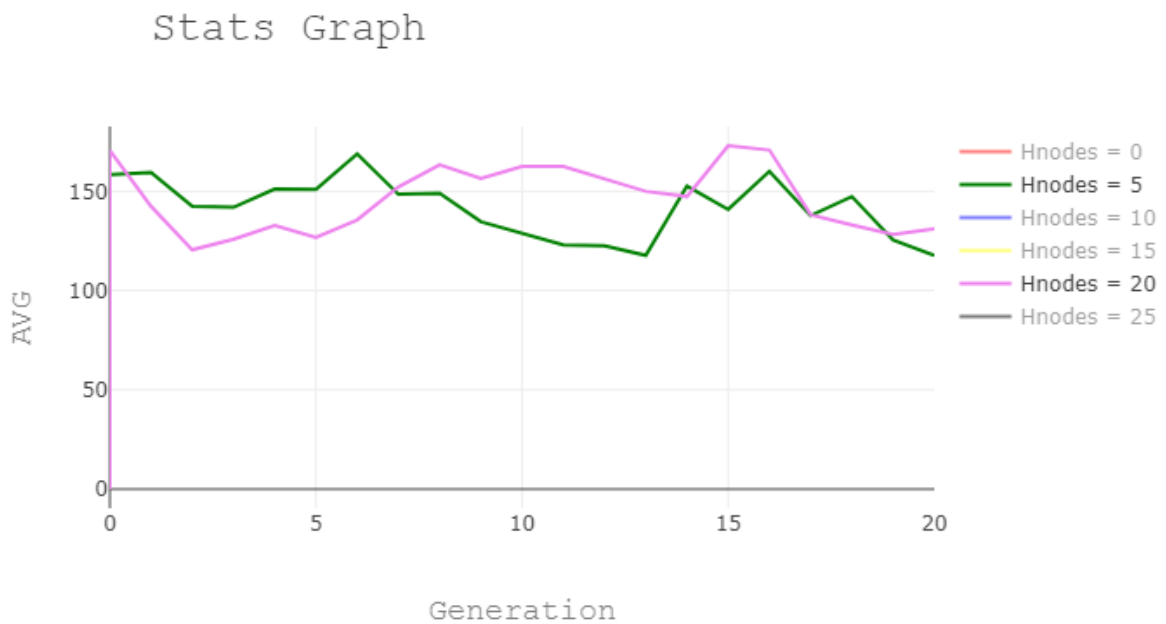


Рисунок 3.3 – Середнє значення пристосованості для скритих нейронів кількості 5-20

3.2.2 Методи схрещування

На даному етапі ми будемо оцінювати пристосованість популяції в залежності від вибору методу схрещування при створенні нового покоління. Проведемо 3 експерименти, в яких:

- 1) покоління отримуються лише завдяки одноточковому схрещуванню
- 2) покоління отримуються лише завдяки двоточковому схрещуванню
- 3) окоління отримуються лише завдяки випадковому вибору між одно-двоточковим схрещуванням

На графіку 3.4 представлено результат роботи моделі. Як можна помітити, кращий результат отримано при симуляції з двоточковим схрещуванням.

Такий результат можна пояснити тим, що при такому методі схрещування популяція досягає найбільшого різноманіття, через це в нього є більше можливостей розвитку. Також треба відмітити, що навіть

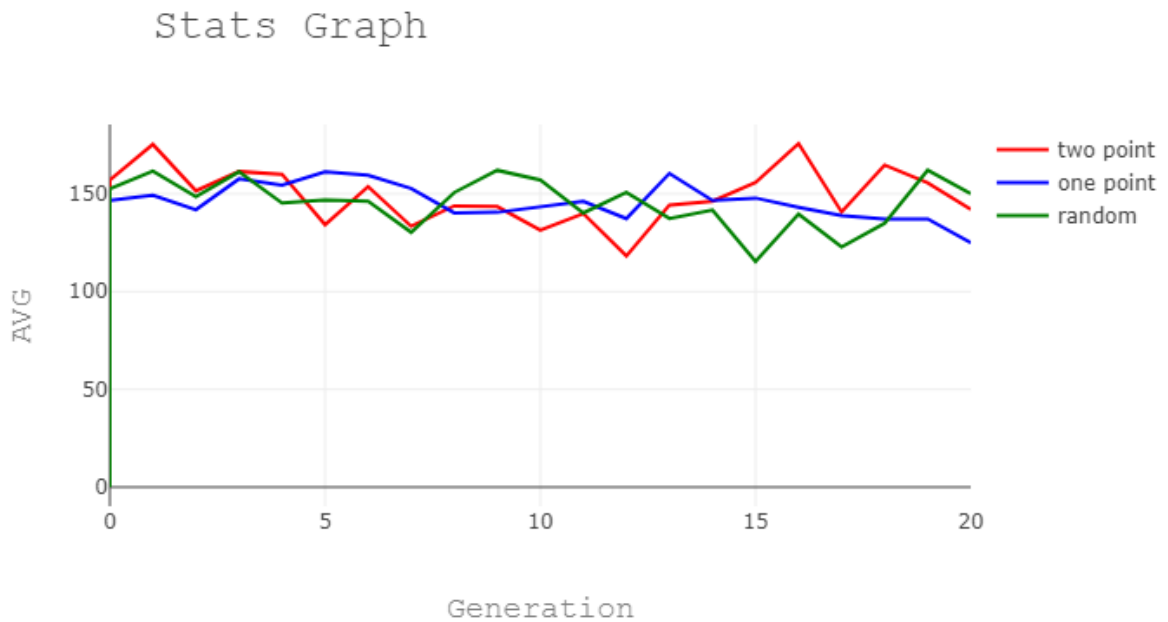


Рисунок 3.4 – Графік результатів симуляції для методів схрещування

випадкового схрещування недостатньо, враховуючи те, що в цьому випадку теж присутнє двоточкове схрещування.

Також з таблиці А.2 можна наглядніше подивитись на шлях розвитку популяції в залежності від покоління.

3.2.3 Методи мутації

Для підбору кращого методу мутації, необхідно зробити чотири симуляції:

- 1) При мутації змінюється лише значення вагів зв'язків нейронної мережі
- 2) При мутації змінюється лише значення біасу(зміщення) нейрону з всіх шарів окрім вхідних
- 3) При мутації змінюється лише значення функції активації нейрону з всіх шарів окрім вхідних
- 4) Випадковим чином обирається одна з вищезазначених мутацій

Подивимось на результати вказаних симуляцій 3.5. Найкращий

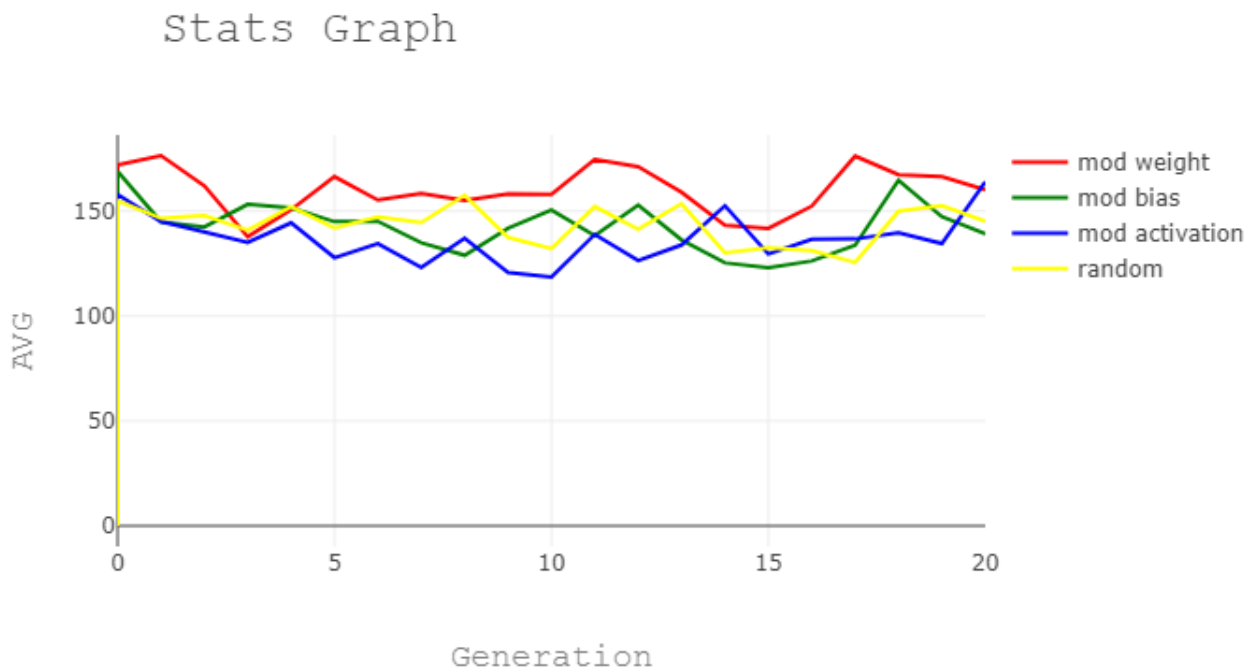


Рисунок 3.5 – Графік результатів симуляції для методів мутації

результат модель показала при мутації вагів, найгірший – мутація функції активації, що пояснюється необхідності приділити особливої уваги до підбору цієї функції для кожної моделі окремо.

Можна зробити висновок, що мутація вагів нейронної мережі найбільше впливає на пристосованість організму та подальшого успішного розвитку.

Також з таблиці А.3 можна наглядніше подивитись на шлях розвитку популяції в залежності від покоління.

3.2.4 Частота мутації

Частота мутації є доволі вагомим параметром через те, що від нього, буде залежати різноманітність покоління, яка, в свою чергу, може забезпечити кращий розвиток. Але слід відмітити, що надмірна кількість мутацій призведе до негативних наслідків, які будуть вповільнювати навчання.

Тож буде відбуватись 3 симуляції з різними параметрами частоти мутації, а саме: 0.3, 0.2 0.1 відповідно.

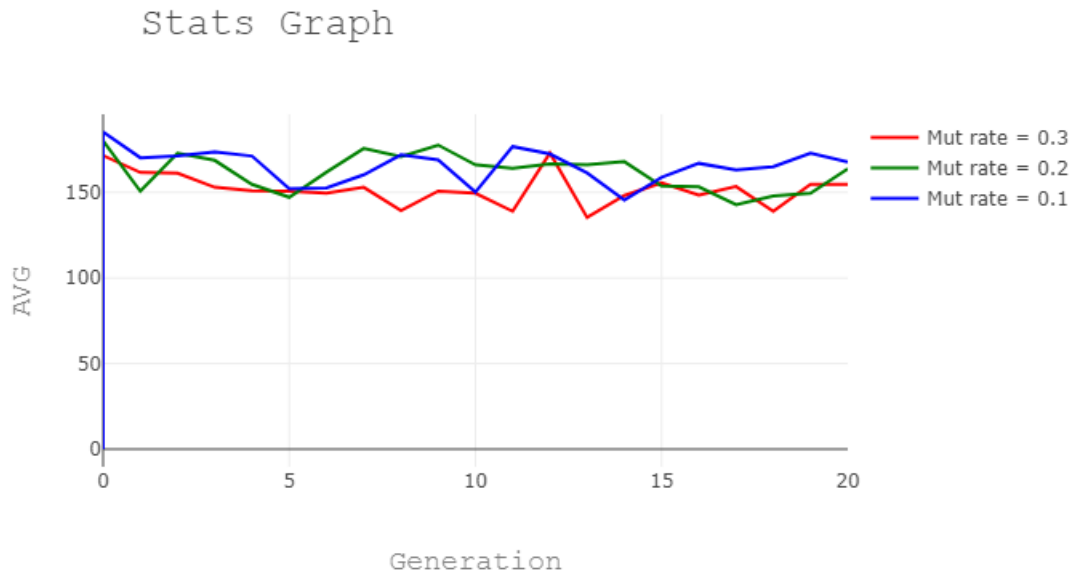


Рисунок 3.6 – Графік результатів симуляції для частоти мутації

На графіку 3.6 видно, що при 0.3 модель демонструє найгірші результати, що запевняє в тому, що це дуже великий показник для цього параметру. При цьому моделі з 0.1 та 0.2 демонструють доволі близькі результати, але судячи з порівняння середньої пристосованості цих ситуацій можна зробити висновок, що симуляція з 0.1 продемонструвала себе краще.

В таблиці А.4 можна побачити числові значення середньої пристосованості за поколіннями при різні частоті мутації для 0.3 - 0.2 - 0.1 відповідно. За нею можна більше детально відстежити хід розвитку популяцій на кожному поколінні.

Висновки до розділу 3

В цьому розділі було описано модель еволюційної поведінки організмів, які мають нейронну мережу, як мозок. Було описано

особливості даної моделі та вказані вразливості. Розвиток поколінь організмів відбувається завдяки генетичному алгоритму, а в якості хромосоми була використана нейронна мережа у кожного організму.

Після реалізацій моделі були надані оптимальні параметри та методи зокрема для: кількості прихованих нейронів у нейронній мережі мозку організму, методу схрещування організмів, методу мутації та показника частоти мутації.

ВИСНОВКИ

В дипломній роботі було проведено аналіз предметної області, розглянуто існуючі методи побудови моделей реалізації розвитку популяції простих організмів за допомогою генетичного алгоритму та нейронної мережі.

Отримана модель представляє собою популяцію організмів з нейронною мережею, яка представляє мозок і є хромосоною організму. За допомогою генетичного алгоритму відбувається оптимізація цієї структури.

Перший розділ роботи описує такі поняття, як генетичний та еволюційний алгоритми. Надає уявлення про методи реалізації алгоритму. Кожен крок, а саме: генерація популяції, розвиток, елітизм, селекція, схрещування та мутація, детально описано з наданням можливих варіантів їх реалізації та особливостей кожного. Також були представлені випадки використання генетичних алгоритмів.

Другий розділ присвячено нейронним мережам, їх основним поняттям та визначенням. Були розглянуті основні принципи роботи нейронних мереж. Також наведені ситуації використання нейронних мереж.

В практичній частині було побудовано модель розвитку простих організмів. Розвиток відбувається завдяки генетичному алгоритму. А розвивається нейронна мережа, яка виступає хромосоною та мозком індивиду. Описані особливості побудованої моделі.

Проаналізувавши отриману модель, було визначено оптимальні параметри та методи: для кількості нейронів у прихованому шарі — 20, метод схрещування — двоточкове, метод мутації — мутація вагів, частота мутації — 0,1.

ПЕРЕЛІК ПОСИЛАНЬ

1. Andrew N. Sloss and Steven Gustafson, Evolutionary Algorithms Review 2019.
2. Aymeric Viole, Alissa M. Kleinnijenhuis, Doayne J. Farmer, "Qualities, challenges and future of genetic algorithms: a literature review 2021, p.5-15.
3. An Introduction to Genetic Algorithms, Mitchell, M. 1998.
4. John Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan. 1975.
5. Genetic Algorithms + Data Structures = Evolution Programs, Michalewicz, Z., 2013.
6. Генетичні алгоритми оптимізації та машинне навчання, Голдберг Д. Е. с.84.
7. Бібліотека Neataptic.js [Електронний ресурс]. — Режим доступу: <https://wagenaartje.github.io/neataptic/>.
8. Rosasco, L.; De Vito, E. D.; Caponnetto, A.; Piana, M.; Verri, A. "Are Loss Functions All the Same?"2004.
9. Cybenko, G. (December 1989). "Approximation by superpositions of a sigmoidal function". Mathematics of Control, Signals, and Systems. 2 (4): 303–314.

ДОДАТОК А ТАБЛИЦІ РЕЗУЛЬТАТІВ СИМУЛЯЦІЙ

Таблиця А.1 – Матриця кореляції для кількості прихованих нод

Покоління	hnode = 0	hnode = 5	hnode = 15	hnode = 20	hnode = 25
0	153.26	158.54	155.61	170.62	159.29
1	138.24	159.51	141.06	142.44	150.83
2	162.64	142.45	145.56	120.55	145.75
3	147.65	141.95	151.57	125.87	144.96
4	132.63	151.13	156.81	132.87	144.79
5	158.76	151.01	163.18	126.77	142.51
6	149.09	168.82	140.54	135.57	153.56
7	134.23	148.52	131.68	152.24	130.71
8	137	149.05	102.33	163.43	125.04
9	133.23	134.68	138.55	156.54	147.61
10	138.28	128.86	119.23	162.59	135.77
11	134.51	123.14	139.25	162.56	150.93
12	135.72	122.52	157.47	156.18	155.25
13	129.33	117.72	146.21	150.01	118.63
14	147.2	152.65	136.68	147.43	125.72
15	122.52	140.95	144.19	173.07	132.63
16	119.37	160.07	147.23	170.82	137.57
17	126.2	137.83	129.59	138.24	143.57
18	143.29	147.36	136.91	133.01	143.97
19	139.48	125.52	158.02	128.2	164.49
20	121.44	117.8	140.73	131.18	129.86

Таблиця А.2 – Матриця кореляції для методів схрещування

Покоління	Двоточкове	Одноточкове	Випадкове
0	156.99	146.71	152.59
1	175.25	149.23	161.6
2	151.55	141.88	148.61
3	161.36	157.64	161.23
4	160.01	154.34	145.29
5	134.19	161.25	146.87
6	153.67	159.4	146.21
7	133.4	152.77	130.32
8	143.84	140.29	150.65
9	143.43	140.52	162
10	131.33	143.37	157.09
11	139.91	146.06	140.22
12	118.15	137.26	150.77
13	144.22	160.44	137.42
14	146.14	146.61	141.62
15	155.8	147.83	115.35
16	175.67	143.08	139.57
17	140.66	138.85	122.77
18	164.71	137.1	134.95
19	155.68	137.08	162.1
20	141.98	124.88	150.15

Таблиця А.3 – Матриця кореляції для методів мутації

Покоління	Ваги	Біас	Ф-я активації	Випадкова
0	172.07	169.06	157.99	154.96
1	176.49	144.91	144.99	146.55
2	162.07	142.45	139.97	147.95
3	137.74	153.26	135.12	140.86
4	150.7	151.47	144.43	152.05
5	166.41	144.98	127.82	141.94
6	155.31	145.15	134.54	147.31
7	158.43	134.95	123.11	144.56
8	155.15	128.98	137.1	157.57
9	158.19	141.88	120.81	137.51
10	157.9	150.58	118.49	132.19
11	174.72	138.35	138.93	152.19
12	171.22	152.92	126.43	141.4
13	159.02	136.35	133.92	153.57
14	143.3	125.35	152.54	130.16
15	141.78	122.99	129.62	132.49
16	152.4	126.25	136.58	131.33
17	176.28	133.67	136.72	125.51
18	167.4	164.75	139.7	149.98
19	166.48	147.35	134.58	152.55
20	160.17	139.13	163.97	145.15

Таблиця А.4 – Матриця кореляції для частоти мутації

Покоління	Частота = 0.3	Частота = 0.2	Частота = 0.1
0	171.57	180.24	185.37
1	161.7	150.7	170.26
2	161.29	172.87	171.3
3	152.97	168.71	173.75
4	151.08	154.73	171.25
5	150.69	147.19	152.14
6	149.65	161.88	152.65
7	152.97	175.66	160.31
8	139.33	170.81	172.01
9	150.82	177.69	168.93
10	149.49	166.17	150.25
11	138.96	163.99	176.79
12	173.02	166.61	172.5
13	135.39	166.23	161.4
14	148.21	167.95	145.54
15	155.64	153.78	158.87
16	148.5	153.23	166.97
17	153.41	142.81	163.08
18	138.97	147.83	165
19	154.7	149.52	172.99
20	154.6	163.84	167.7