

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Еволюційні алгоритми в оптимізації параметрів штучних нейронних мереж
(тема)

Виконав:
студент 2 курсу, групи СШМ-22-1
Лобанов А.Д.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник проф. Філатов В.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)
Кафедра _____ Штучного інтелекту _____
(повна назва)
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)
Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Системи штучного інтелекту _____
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Лобанову Андрію Дмитровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Еволюційні алгоритми в оптимізації параметрів штучних нейронних мереж

затверджена наказом університету від 1 квітня 20 24 р. № 260Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14 червня 20 24 р.

3. Вихідні дані до роботи Наукові публікації, статті та технічні матеріали, що описують принципи роботи та архітектуру штучних нейронних мереж, зокрема LSTM мереж, а також еволюційних алгоритмів, джерела, що містять інформацію про методи обробки природної мови, векторизації тексту та класифікації даних, документація та ресурси програмування на мові Python, бібліотеки для машинного навчання TensorFlow та Keras, та інструменти для роботи з даними scikit-learn.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі

2) Теоретичні дослідження

3) Експериментальні дослідження

РЕФЕРАТ

Пояснювальна записка: 93 с., 8 рис., 2 табл., 2 дод., 24 джерела.

БАГАТОШАРОВІ ПЕРЦЕПТРОНИ, БАЗИСНІ ФУНКЦІЇ, ЕВОЛЮЦІЙНІ АЛГОРИТМИ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ, ШТУЧНІ НЕЙРОННІ МЕРЕЖІ, ШТУЧНИЙ ІНТЕЛЕКТ.

Об'єкт дослідження – еволюційні алгоритми та їх реалізації за допомогою штучних нейронних мереж.

Предмет дослідження – конкретні типи штучних нейронних мереж та їх використання в еволюційних алгоритмах.

Мета роботи – аналізувати та оцінювати ефективність різних видів штучних нейронних мереж в контексті їх застосування в еволюційних алгоритмах. Дослідження орієнтовано на пошук оптимальних підходів до використання штучних нейронних мереж для покращення ефективності еволюційних алгоритмів.

Методи дослідження – вивчення існуючої теоретичної літератури з теми, аналіз різних типів штучних нейронних мереж, проведення експериментів для оцінки впливу різних типів штучних нейронних мереж на ефективність еволюційних алгоритмів, аналіз отриманих даних.

ABSTRACT

Master's thesis contains: 93 pp., 8 fig., 2 tabl., 2 ann., 24 references.

ARTIFICIAL INTELLIGENCE, ARTIFICIAL NEURAL NETWORKS, BASIC FUNCTIONS, CONVOLUTIONAL NEURAL NETWORKS, EVOLUTIONARY ALGORITHMS, MACHINE LEARNING, MULTILAYER PERCEPTRONS.

The object of research is evolutionary algorithms and their implementation using artificial neural networks.

The subject of research is specific types of artificial neural networks and their use in evolutionary algorithms.

The purpose of the work is to analyze and evaluate the effectiveness of various types of artificial neural networks in the context of their application in evolutionary algorithms. The research is focused on the search for optimal approaches to the use of artificial neural networks to increase the efficiency of evolutionary algorithms.

Research methods – study of the existing theoretical literature on the topic, analysis of different types of artificial neural networks, conducting experiments to assess the impact of different types of artificial neural networks on the effectiveness of evolutionary algorithms, analysis of the obtained data.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	9
Вступ.....	10
1 Аналіз предметної галузі	12
1.1 Визначення та пояснення еволюційних алгоритмів.....	12
1.2 Визначення та пояснення штучних нейронних мереж	13
1.2.1 Важливість і актуальність параметрів оптимізації в ШНМ	15
1.2.2 Короткий огляд використання ЕА для цього процесу оптимізації	16
1.3 Огляд літератури	19
1.3.1 Попередні дослідження еволюційних алгоритмів.....	19
1.3.2 Минулі методи оптимізації параметрів у штучних нейронних мережах	20
1.3.3 Попереднє використання ЕА для оптимізації параметрів у ШНМ	22
1.4 Теорія еволюційних алгоритмів	23
1.4.1 Біологічні принципи, що стоять за ЕА	23
1.4.2 Ключові особливості та характеристики ЕА	25
1.4.3 Типи ЕА та їх відмінності	26
1.5 Теорія створення штучних нейронних мереж.....	28
1.5.1 Біологічне натхнення для ШНМ	28
1.5.2 Основні компоненти ШНМ: нейрони, ваги, зміщення, шари ...	29
1.5.3 Типи ШНМ та їх відмінності	31
1.6 Оптимізація параметрів в ШНМ.....	32
1.6.1 Важливість оптимізації параметрів у ШНМ	32
1.6.2 Традиційні методи оптимізації параметрів	34
1.7 Застосування еволюційних алгоритмів в оптимізації	35
1.7.1 Як ЕА можна використовувати для оптимізації параметрів	35
1.7.2 Переваги та обмеження використання ЕА для оптимізації.....	37

1.7.3 Приклади ЕА в оптимізації.....	39
1.8 Висновок розділу.....	40
2 Теоретичні дослідження.....	43
2.1 Вступ.....	43
2.2 Опис генетичного алгоритму.....	43
2.3 Еволюція навчальних правил у штучних нейронних мережах.....	45
2.4 Основи нейронної мережі.....	48
2.5 Архітектура та процес навчання нейронної мережі.....	49
2.6 Методологія.....	50
2.7 Послідовність використання ГА у ШНМ.....	53
3 Експериментальні дослідження.....	59
3.1 Вступ.....	59
3.2 Оптимізація нейронної мережі LSTM за допомогою генетичного алгоритму.....	60
3.3 Порівняльний аналіз нейронної мережі LSTM з оптимізацією генетичним алгоритмом та без неї.....	61
3.4 Опис програмного забезпечення та параметрів.....	62
3.5 Інтеграція ШНМ та ЕА.....	63
3.5.1 Як змодельований ЕА.....	63
3.6 Реалізація ГА у ШНМ.....	65
3.6.1 Аналіз датасету.....	65
3.6.2 Підготовка даних.....	66
3.6.3 Вихідні дані для оптимізації нейронних мереж.....	67
3.6.4 Інтеграція ЕА з LSTM мережею у програмі.....	68
3.6.5 Попередня обробка даних і розробка функцій.....	70
3.6.6 Модель навчання та оцінювання.....	70
3.7 Аналіз результатів.....	71
3.8 Доцільність використання ГА.....	77
3.9 Висновки розділу.....	79
Висновок.....	82

Перелік джерел посилання	83
Додаток А Лістинг коду	86
Додаток Б Відомість кваліфікаційної роботи.....	93

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- ЗНМ – згорткові нейронні мережі;
- ЗПП – зворотнє поширення помилки;
- РНМ – рекурентні нейронні мережі;
- ШНМ – Artificial Neural Network – штучна нейронна мережа;
- AI – Artificial Intelligence – штучний інтелект;
- AUC – Area Under the Curve – площа під кривою;
- CSV – Comma-Separated Values – значення, розділені комами;
- EA – Evolutionary Algorithm – еволюційний алгоритм;
- FN – False Negatives – хибно негативні результати;
- FP – False Positives – хибно позитивні результати;
- ID – Identifier – ідентифікатор;
- LSTM – Long Short-Term Memory – довга короткотривала пам'ять;
- NLP – Natural Language Processing – обробка природної мови;
- ROC – Receiver Operating Characteristic – характеристика роботи приймача;
- TF-IDF – Term Frequency-Inverse Document Frequency – частота термінів-інверсна частота документів;
- TN – True Negatives – істинно негативні результати;
- TP – True Positives – істинно позитивні результати.

ВСТУП

У сучасному світі широке активне використання штучних нейронних мереж спостерігаємо майже у всіх сферах наукових досліджень і практичних застосувань. Така тенденція обумовлена вражаючою гнучкістю і високою ефективністю нейронних мереж при рішенні різних задач. Водночас успіх роботи нейронних мереж на багатьох дослідницьких й прикладних задачах сильно залежить від вдалого підбору параметрів мережі.

В умовах високої складності та численності параметрів, з якими має працювати вчений перебір всіх можливих комбінацій займає надмірно великий час. Тому одним з ключових напрямків в галузі штучних нейронних мереж, що досліджується в цій роботі, є оптимізація параметрів мереж.

Еволюційні алгоритми, які повторюють спосіб природи вирішити проблему оптимізації шляхом еволюції певного покоління, представляють спокійний потенціал для даного питання. Вони здатні робити це ефективно, навіть з урахуванням високої складності простору параметрів і необхідністю багатокритеріальної оптимізації.

Однак, ця дисципліна ще є молодого, і методики, що використовуються в еволюційній оптимізації параметрів штучних нейронних мереж, ще потребують багато досліджень та вдосконалень. В даній роботі ми підхопимо цю викликаючу прапор та проведемо глибоке дослідження в області еволюційної оптимізації параметрів штучних нейронних мереж, створимо еволюційні моделі та методи, призначені для покращення результатів використання штучних нейронних мереж.

Новизна та перспективність цього вектору дослідження полягає в застосуванні нейроеволюційних методів для ефективною оптимізації структури штучних нейронних мереж, а також параметрів навчання. Результати цього дослідження можуть виявитися особливо корисними для

пошуку найкращого рішення в задачах, що вимагають поверхневого огляду ситуацій і вимагають глибокого аналізу для прийняття рішень.

На основі детального дослідження, аналізу і прототипування ми визначимо оптимальний набір методів ШНМ для вирішення конкретних задач. Це дослідження відкриває нові перспективи для розвитку ШНМ та їх застосування в складних системах моделювання й вирішення проблем.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Визначення та пояснення еволюційних алгоритмів

Еволюційні алгоритми (ЕА) – це підмножина еволюційних обчислень, галузь інформатики, яка черпає натхнення з процесу природної еволюції та генетики. Ці алгоритми працюють на основі процесів відбору, мутації, рекомбінації і виживання найбільш пристосованих суб'єктів, подібно до еволюційного процесу природи.

Елементна концепція ЕА передбачає створення популяції індивідуальних рішень проблеми, перевірку їх відповідності вимогам проблеми, а потім ітераційне вдосконалення популяції шляхом рекомбінації найбільш придатних індивідуумів і їх мутації для створення різноманітності. Цей процес повторюється, доки не буде знайдено задовільне рішення або не пройде певна кількість поколінь.

В основі ЕА три основні операції:

– відбір: цей процес спрямований на вибір найкращих особин із популяції, тих, яким буде дозволено відтворюватися;

– кроссовер (або рекомбінація): на цій стадії двоє вибраних батьків обмінюються інформацією для створення нового потомства. Мета полягає в тому, щоб створити кращі рішення шляхом поєднання характеристик батьків;

– мутація: ця операція вносить невеликі випадкові зміни в потомство, щоб забезпечити різноманітність генофонду, запобігаючи передчасному переходу до неоптимальних рішень.

Кілька типів ЕА придумали протягом багатьох років, похідних від принципів біологічної еволюції. До найбільш відомих належать генетичні алгоритми (ГА), стратегії еволюції (СЕ), генетичне програмування (ГП) і еволюційне програмування (ЕП).

Підводячи підсумок, ЕА пропонують надійні, адаптовані та універсальні стратегії оптимізації, здатні долати великі, складні простори пошуку, що робить їх корисними для вирішення складних проблем у різних областях, особливо у випадках, коли традиційні методи можуть бути незастосовними.

1.2 Визначення та пояснення штучних нейронних мереж

Штучні нейронні мережі, або ШНМ, є обчислювальними системами, створеними за зразком людського мозку та нервової системи. Подібно до того, як мозок використовує велику мережу взаємопов'язаних нейронів для обробки та передачі інформації, ШНМ містять шари вузлів штучних нейронів, які допомагають обробляти вхідні дані.

По суті, ШНМ – це математична модель, яка спрощує складність людського мозку, одночасно охоплюючи його головну здатність: здатність до навчання. Кожен штучний нейрон, або вузол, повторює функцію біологічного нейрона, приймаючи численні вхідні дані, обробляючи їх і забезпечуючи вихід.

Основною одиницею ШНМ є штучний нейрон або вузол. Кожен нейрон приймає одне або кілька вхідних значень, обробляє їх за допомогою функції передачі (часто форму сигмоїдної функції або функції випрямленої лінійної одиниці) і створює вихід.

ШНМ мають багат шарову структуру:

- вхідний рівень: це початковий рівень, на якому мережа отримує дані. Кожен вузол у цьому шарі представляє одну функцію даних;

- приховані шари: це шари між вхідним і вихідним шарами. У нейронній мережі може бути багато прихованих шарів, і вони виконують перетворення вхідних даних;

- вихідний рівень: це останній рівень, який дає нам остаточні результати або прогнози.

Структура наглядно продемонстрована на рисунку 1.1.

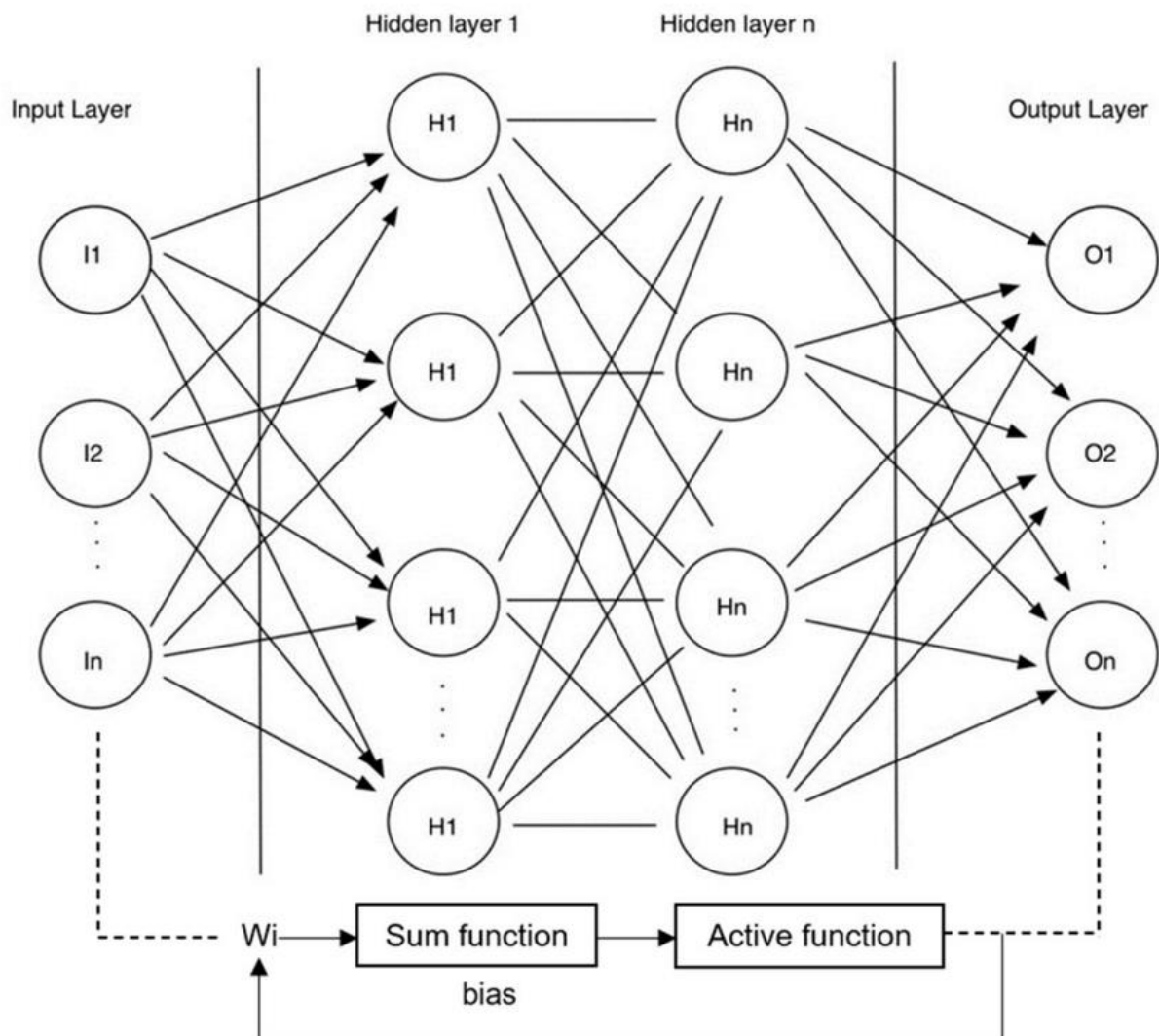


Рисунок 1.1 – структура ШМН

Навчання в ШМН передбачає коригування ваг і зміщень мережі у відповідь на входні дані. Мета полягає в тому, щоб оптимізувати ці параметри, щоб мережа могла якомога точніше відобразити входні дані для виправлення виходів. Зазвичай це робиться за допомогою процесу, який називається зворотним поширенням, і алгоритму оптимізації, де помилка прогнозів мережі зведена до мінімуму.

ШМН можуть значно відрізнятися за своєю архітектурою, включаючи кількість рівнів, які вони мають, кількість вузлів на кожному шарі, тип

з'єднання між вузлами та шарами тощо. Деякі типи ШНМ включають нейронні мережі прямого зв'язку, згорткові нейронні мережі (ЗНМ), і рекурентні нейронні мережі (РНМ), серед інших.

Штучні нейронні мережі є основою глибокого навчання, підгалузі машинного навчання, і успішно застосовуються в багатьох областях, таких як розпізнавання мови, класифікація зображень, обробка природної мови тощо.

1.2.1 Важливість і актуальність параметрів оптимізації в ШНМ

Штучні нейронні мережі функціонують як складна система, вихідні дані якої визначаються архітектурою мережі (наприклад, кількістю рівнів і кількістю вузлів у кожному шарі), а також вагами та зміщеннями, пов'язаними з кожним вузлом. Ці ваги та зміщення є параметрами мережі, і їхні значення можуть значною мірою впливати на продуктивність мережі.

Оптимізація цих параметрів важлива з кількох причин:

– підвищення точності: найбільш прямим ефектом оптимізації параметрів є покращення здатності моделі прогнозувати. Кращі значення параметрів допомагають моделі витягувати релевантніші та складніші шаблони з вхідних даних і, отже, підвищувати точність своїх прогнозів;

– запобігання перенавчанню та недостатньому навчанню: перенавчання та недостатнє навчання є поширеними проблемами машинного навчання, де модель стає або надто складною, або надто простою відповідно. Параметри оптимізації допомагають забезпечити баланс між здатністю моделі узагальнювати та фіксувати детальні закономірності в даних;

– ефективність: належна оптимізація параметрів дозволяє моделям навчатися швидше та працювати краще. Починаючи з оптимальних або близьких до оптимальних значень, навчання може досягти бажаного рівня

продуктивності швидше, заощаджуючи обчислювальні ресурси та час під час процесу навчання;

– стійкість: добре оптимізовані параметри створюють міцні та надійні моделі. Ці моделі добре працюють не лише з навчальними даними, але й з невидимими даними, забезпечуючи їх надійність у різних сценаріях.

Процес оптимізації параметрів завжди був складним завданням через високу розмірність простору параметрів у нейронних мережах. Він включає в себе вагові коефіцієнти налаштування, зміщення, швидкість навчання, кількість прихованих шарів і нейронів, функції активації тощо.

Враховуючи великий простір параметрів, оптимізація зазвичай виконується за допомогою таких методів, як градієнтний спуск і його варіанти. Однак існує зростаючий інтерес до використання природних алгоритмів, які включають еволюційні алгоритми, завдяки їх можливості глобального пошуку, надійності та здатності уникати локальних мінімумів.

1.2.2 Короткий огляд використання ЕА для цього процесу оптимізації

ЕА продемонстрували потенціал для оптимізації параметрів штучних нейронних мереж. Що робить ЕА унікальними, так це їх здатність маніпулювати набором потенційних рішень одночасно, а не лише одним рішенням.

Це означає, що ЕА здатні широко досліджувати простір пошуку, що зменшує ймовірність потрапляння в пастку локальних мінімумів (субоптимальних рішень) порівняно з традиційними методами оптимізації. Ця особливість еволюційних обчислень є особливо корисною при оптимізації складних систем, таких як ШНМ, де ландшафт простору параметрів часто заповнений численними локальними мінімумами.

Ось покроковий огляд того, як ЕА можна використовувати для цього процесу оптимізації:

– початкова сукупність: сукупність потенційних рішень ініціалізується випадковим чином. У контексті ШНМ рішення відповідає набору значень для параметрів мережі (ваги та зміщення). Це означає, що кожна особина в популяції представляє можливу конфігурацію цих параметрів;

– обчислення придатності: функція придатності вимірює якість рішень, що в цьому контексті передбачає застосування значень параметрів до ШНМ і визначення того, наскільки добре вона виконує певне завдання;

– відбір: ЕА вибирає особин (набори параметрів) для відтворення. Особи відбираються пропорційно до їх фізичної підготовки, тобто ті, хто краще працює, мають вищі шанси бути обраними;

– кроссовер і мутація: це основні операції процесу еволюції. Схрещування поєднує ознаки двох батьків для створення потомства, тоді як мутація вносить невеликі випадкові зміни. Результатом цих процесів є нові параметри, які потім можна протестувати в ШНМ;

– заміна: тут деякі особи замінюються новоутвореними, зазвичай тими, які працюють гірше, щоб звільнити місце для потенційно кращих рішень;

– повторення: кроки від обчислення придатності до заміни повторюються, доки не буде знайдено бажане рішення або не буде виконано критерій зупинки. Це може бути максимальна кількість поколінь, поріг придатності рішення або прийнятний проміжок часу.

Загалом, використання ЕА для оптимізації ШНМ може дати більш надійні та ефективніші моделі, оскільки ЕА здатні до глобальної оптимізації, а не задовольнятися локальними оптимумами. Тим не менш, ці методи також мають свої проблеми, такі як обчислювальна вартість, необхідність ретельного вибору параметрів (для самого ЕА) і складність, пов'язана з балансуванням дослідження (тестування нових рішень) і експлуатації (доопрацювання існуючих рішень).

Виходячи із усього переліченого, можемо назвати такі переваги ЕА для оптимізації ШНМ:

- глобальна оптимізація: ЕА досліджують простір пошуку більш широко, роблячи їх менш сприйнятливими до локальних мінімумів;

- ефективність: ЕА можуть паралельно обробляти множинну потенційних рішень, що робить їх більш ефективними, ніж послідовні методи;

- простота: ЕА не потребують знання градієнтів функції втрат, що робить їх зручними для роботи з складними нейронними мережами.

Які ж існують виклики для використання ЕА? Обчислювальна складність являє собою ключовий обмежувач застосування еволюційних алгоритмів, особливо в контексті великих і складних нейронних мереж. Розробка багатьох хромосом потребує значних обчислювальних ресурсів для множинних обчислень функції придатності. До того ж, еволюційні алгоритми повинні забезпечувати належне балансування дослідження, що передбачає пошук нових потенційних рішень, та експлуатації, що орієнтована на зосереджене використання вже знайдених оптимальних рішень. Цей процес ускладнюється додатковими параметрами еволюційних алгоритмів, які можуть істотно вплинути на їхню ефективність, а їх оптимізація обертається власною складною задачею. Кінцевою перешкодою слугує інтерпретація результатів, яка детермінує розуміння того, чому деякі рішення виявилися кращими за інші, але вона може виявитися нетривіальною завдяки внутрішній складності еволюційних алгоритмів.

Перспективи використання ЕА у ШНМ я бачу такі:

- розробка нових алгоритмів ЕА, які є більш ефективними та стійкими до локальних мінімумів;

- розробка методів адаптації параметрів ЕА під час роботи;

- розробка гібридних алгоритмів, які поєднують ЕА з іншими методами оптимізації;

- розробка алгоритмів, які поєднують принципи ЕА з нейронними мережами;

- розробка методів автоматичного налаштування параметрів ЕА;

- використання GPU для прискорення обчислень ЕА.

Очікувані результати:

- покращення точності та ефективності ШНМ;

- автоматичне визначення оптимальної архітектури та гіперпараметрів;

- зменшення часу та обчислювальних витрат.

1.3 Огляд літератури

1.3.1 Попередні дослідження еволюційних алгоритмів

З моменту зародження еволюційних алгоритмів (ЕА), що значною мірою відбувалося в умовах референтного дослідження 1975 року – «Адаптації в природних і штучних системах», запропонованого Джоном Голландом, було проведено багато десятиліть досліджень [4]. Ця робота заклала основу для ГА, важливого типу ЕА, і принципи та стратегії, що були започатковані в ній, зумовлювали подальшу динаміку дослідження ЕА.

Знаходячись на перетині років, було розглянуто численні варіації ЕА, причому кожна специфічна реалізація пропонувала інноваційні стратегії для розв'язання проблем. Серед них були еволюційні стратегії (ЕС), які представлені Рехенбергом і Швевелем, генетичне програмування (ГП), запропоноване Козою, і диференціальна еволюція (ДЕ), яку розробили Сторм та Прайс. Порівняння цих різних ЕА, є зазвичай неодмінним елементом академічної літератури в даному напрямі.

Наукові новітні розробки, часто концентруються на найбільш суттєвому застосуванню ЕА – в задачах оптимізації. До речі, такі праці, як стаття Голдберга про генетичні алгоритми, показали, на кшталт ЕА, як

відмінна здатність пошуку майже оптимальних рішень в умовах складних мультимодальних ландшафтів.

Окрім експериментального напрямку, є значна кількість робіт, що здійснюють теоретичний аналіз ЕА з погляду кращого розуміння їхньої поведінки, визначення факторів, які впливають на їх продуктивність, і створення математичних моделей для їх представлення.

Література також відзначає дослідження щодо гібридних моделей, які пов'язують ЕА з іншими методами. Це включає в себе такі методи, як оптимізація рою частинок, оптимізація колонії мурашок і моделювання відпалу для подальшого підвищення їх продуктивності у вирішенні проблем оптимізації.

Щодо практичного застосування ЕА, вони знайшли застосування в галузях, починаючи від інженерної оптимізації до оптимізації дизайну, планування, теорії ігор, машинного навчання, і так далі.

В цілому, можна сказати, що література про ЕА відображає дві основні тенденції – вивчення способів вдосконалення показників ЕА і невтомний пошук нових контекстів, де ці техніки можуть знайти застосування. Останні напрацювання у цій дослідницькій галузі увібрали й у себе застосування ЕА в області штучних нейронних мереж та оптимізації їх параметрів.

1.3.2 Минулі методи оптимізації параметрів у штучних нейронних мережах

Оптимізація параметрів у ШНМ стала предметом інтенсивних досліджень в області машинного навчання, де впродовж декількох десятиліть було запропоновано широкий спектр стратегій, спрямованих на підвищення продуктивності та ефективності цих мереж.

Алгоритм градієнтного спуску, разом з його модифікаціями, такими як стохастичний градієнтний спуск (СГС), міні-груповий градієнтний спуск

та численні адаптивні версії (наприклад, RMSProp і Adam), розглядаються як найпоширеніший метод оптимізації параметрів у ШНМ. Він базується на ітеративному коригуванні параметрів з метою мінімізації функції втрат, де градієнт функції втрат застосовується як настанова в виправленнях [5].

Паралельно застосовуються методи на основі імпульсу, включно з методом градієнтного спуску з імпульсом та прискореним градієнтом Нестерова, які удосконалюють базову концепцію градієнтного спуску, інтегруючи в розрахунки попередні градієнти, для збільшення швидкості збіжності та уникнення проблем з коливаннями та локальними мінімумами.

Методи регуляризації, спрямовані на те, щоб запобігти перенавчанню, включають L1 та L2 регуляризацію, методологія відсіву та стратегія ранньої зупинки. Вони не прямо оптимізують параметри, а намагаються забезпечити краще узагальнення моделі на невидимі дані.

Методи оптимізації другого порядку, такі як метод Ньютона, BFGS та L-BFGS, зазвичай враховують вторинні похідні функції втрат, що можуть сприяти покращенню продуктивності, хоча мають обчислювальну вартість, особливо для великих нейронних мереж.

Методи пошуку сітки та випадкового пошуку, які представляють собою прості стратегії пошуку параметрів, часто використовуються для гармонізації гіперпараметрів, систематично перевіряючи різні набори параметрів (при пошуку у сітці) або вибірка випадкового набору (при випадковому пошуку), щоб знайти оптимальний набір.

Байєсовські методи оптимізації, такі як оптимізація за допомогою Гауссового процесу та ланцюгові методи Монте-Карло, створюють апостеріорне розподілення функцій (або параметрів), щоб визначити оптимальні параметри.

Проте в рамках цих методів існують певні обмеження. Вони можуть стикатися з проблемами, як застрягання в локальних мінімумах, проблемами градієнту зникнення/вибухання та виклики з обчислювальними ресурсами, особливо при розширенні розміру мереж. Це активізувало

дослідження альтернативних стратегій оптимізації, таких як еволюційні алгоритми. Протягом останнього десятиліття було зареєстровано значне зростання кількості наукової літератури, присвяченої застосуванню ЕА для оптимізації параметрів ШНМ.

1.3.3 Попереднє використання ЕА для оптимізації параметрів у ШНМ

Еволюційні алгоритми, використані для параметричної оптимізації штучних нейронних мереж, стали предметом захоплення у наукових дослідженнях протягом тривалого часу. Значний інтерес до них обумовлений невикликаністю поверхонь втрат штучних нейронних мереж, які можуть мати безліч локальних мінімумів та складності навігації у величезних просторах параметрів.

Один з напрямків досліджень ЕА мав як основу оптимізацію ваг мереж. Важливий внесок в цю область приписується Яо, який в 1999 році оприлюднив працю «Розвиток штучних нейронних мереж» [6]. Даний важливий огляд включає різні типи ЕА, серед них – генетичні алгоритми та стратегії еволюції, що були присвячені тренуванню ШНМ через параметричну оптимізацію вагових коефіцієнтів.

Інші дослідники ввели ЕА в середовище разом з архітектонікою чи топологією ШНМ, як наприклад необмежена кількість прихованих шарів або вузлів. Це твердження можна підтвердити провідною методикою – нейроеволюційним методом розширених топологій, пропонуваним Стенлі та Мійккулайненом у 2002 році, де генетичні алгоритми були використані для розвитку ваг та їх структур в ШНМ [7].

Третя група включає використання ЕА як засобу для гіпер-параметричної оптимізації в ШНМ (наприклад, швидкість навчання, активаційна функція, кількість прихованих шарів тощо).

Багатоцільова оптимізація – новий сектор, що експлуатує ЕА для оптимізації ШНМ, спрямований на одночасну оптимізацію кількох, часто

конфліктуючих, цілей, таких як мінімізація помилок та складність мережі. Важливим прикладом є робота Ігеля та Хюскена, які використовували багатоцільову оптимізацію на основі Парето з ЕА для тренування харчування ШНМ.

Дійсність підтверджує, що тестування ЕА для оптимізації ШНМ у реальних задачах, від апроксимації функцій, розпізнавання образів, прогнозування фондових ринків до проблем керування та планування, дало успішні результати.

Взагалі, всі дослідники доходять згоди про той факт, є ЕА доречною альтернативою градієнтним методам при оптимізації ШНМ, основними перевагами яких є швидкість досягання глобального мінімуму та зображення паралельного обчислення. Попри певні проблеми, пов'язані зі значними обчисленнями та тонким вибором власних параметрів (швидкість мутацій, тип перехрещування тощо), література продовжує просуватися вперед, демонструючи прогрес у використанні ЕА для оптимізації ШНМ, та розглядає нові дослідження, які заглиблюються у цей терен для інноваційних вдосконалень та адаптацій цих методів.

1.4 Теорія еволюційних алгоритмів

1.4.1 Біологічні принципи, що стоять за ЕА

ЕА представляють групу методик пошуку та оптимізації, які були здобуті за натхненням від принципів природної еволюції.

Представляю перегляд біологічних принципів, що складають основу ЕА:

– біологічна еволюція: процес еволюції в ЕА моделюється з біологічних явищ, що пов'язані з переказом найсильнішим особинам. За цим принципом, організми, краще пристосовані до середовища, виживають і

виробляють нащадків, при чому позитивні ознаки з часом певно посилюються;

– селекція: в дикій природі усяке живе змушене конкурувати за ресурси. Найкраще адаптовані до середовища особини отримують право на репродукцію й передачу свого геному наступним поколінням. Цьому у цілому й відповідає принцип селекції в ЕА;

– рекомбінація: тактика статевого розмноження в природі передбачає генетичний обмін, що утворює доступ до варіативності та комбінаторики ознак із метою підвищення фенотипічної пластичності й адаптованості. В ЕА цей процес імітується через операції кросинговеру, рекомбінуючи генетичний матеріал батьківських рішень;

– мутація: мутацій в природі є наслідком помилок реплікації, що створюють нові ознаки у нащадків. Більшість мутацій безкорисні, але іноді мутації підвищують придатність особини. Операції мутації в ЕА вводять випадкові зміни в рішення, підтримуючи різноманітність для того, щоб уникати локальних мінімумів;

– виживання найсильніших: Чарльз Дарвін сформулював ідею, згідно з якою найкраще пристосовані особини мають найбільше шансів успадковувати свій генофонд наступним поколінням. У ЕА «пристосовність» рішення визначається функцією пристосованості, специфічною для заданої проблеми. Особини, що мають вищий коефіцієнт пристосовності, мають більше шансів мати потомство.

Реалізуючи природні механізми розвитку, ЕА втілюють стохастичний, але керований пошук кращих рішень, навіть в складних, багатовимірних просторах пошуку. Це робить їх потужними інструментами для вирішення великого спектра проблем оптимізації. Проте, хоча ЕА були створені за відбитком біологічної еволюції, їх можна вважати більше математичними та обчислювальними моделями, що спрощують і абстрагують складні нюанси біологічної еволюції, для того щоб задовольнити їх основну мету [8].

1.4.2 Ключові особливості та характеристики ЕА

ЕА мають ряд відмінних рис та характеристик, які часто роблять їх корисними інструментами для вирішення складних задач оптимізації.

Відповідно до класичних методів оптимізації, що працюють із одним можливим розв'язанням, ЕА маніпулюють сукупністю потенційних рішень. Кожен індивід представляє одне рішення, і всі індивіди розвиваються одночасно. Це надає ЕА можливість одночасно досліджувати різноманітні регіони простору пошуку, що підвищує шанси на знаходження глобальних оптимумів. Напрямок пошуку в ЕА контролює фітнес-функція, яка оцінює якість рішень, віддаючи перевагу вище пристосованим індивідам (рішенням). Операції, на зразок мутації та кросинговеру, застосовуються в ЕА для створення нових рішень (рисунок 1.2).

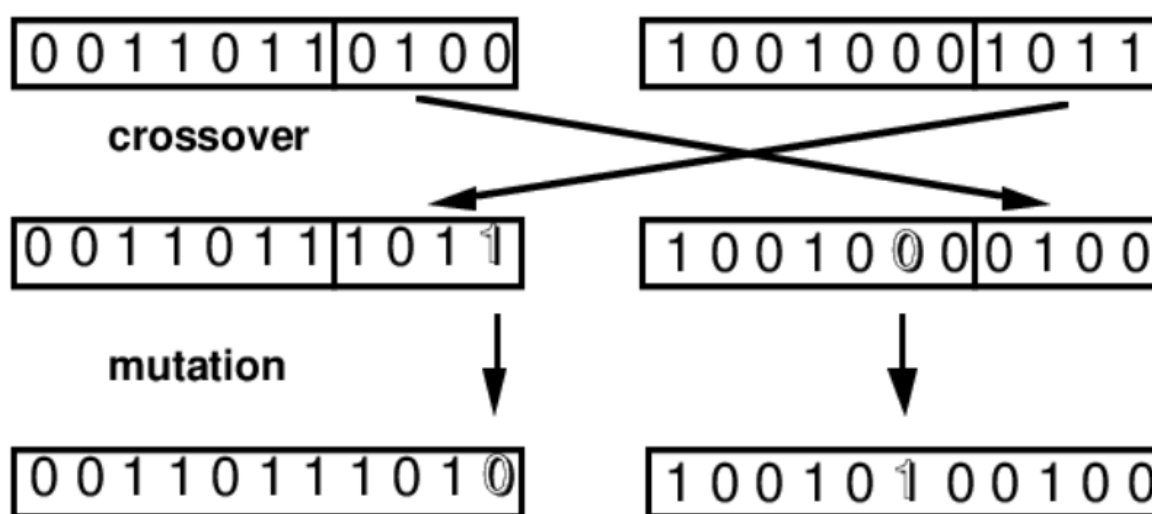


Рисунок 1.2 – оператори кросинговеру та мутації в генетичному алгоритмі

Цей елемент випадковості може сприяти дослідженню недоступних областей простору пошуку та уникати застрягання в локальних мінімумах. Можна відзначити високу стійкість ЕА та здатність працювати в умовах шуму. Ці методи також відзначаються високою гнучкістю, оскільки не

вимагають жодних моделей та певних припущень по відношенню до простору пошуку. Отже, ЕА можуть бути застосовані в широкому спектрі задач оптимізації. ЕА є паралельними по своїй природі. З огляду на те, що кожен індивід у популяції представляє собою окреме рішення, такі операції, як обчислення фітнесу, можуть бути виконані симульовано для всіх індивідів. Це робить ЕА зручними для використання в розподілених обчислювальних середовищах. Серед операцій відбору, перехресту та мутації, ЕА зберігають добре збалансований процес між дослідженням простору пошуку (приспосовано з метою пошуку кращих рішень) та зосередженням (вдосконаленням поточних найкращих рішень). ЕА спрямовані на принципи природної еволюції, включаючи селекцію, варіацію та виживання найкраще пристосованих. Ці засади дозволяють ЕА адаптуватися, вчитися від минулого досвіду та збільшувати ефективність з часом. За допомогою еволюційних операторів, таких як мутація та перехід, ЕА адаптуються до зміни, що робить їх придатними для динамічних середовищ, де сама проблема може змінюватися з часом.

Незважаючи на вищезгадані міцні сторони методу, ЕА також мають свої слабкі сторони, такі як потреба значних обчислювальних ресурсів та задоволення, що можлива передчасна збіжність до поточного рішення. Вибір використання ЕА зазвичай залежить від природи та вимогі задач, що розглядається.

1.4.3 Типи ЕА та їх відмінності

Усі ЕА кореняться в одних і тих самих базових принципах, проте існують різні типи ЕА, кожний з яких вирізняється унікальними механізмами та варіаціями.

Серед них є ГА, ініційовані принципами природного відбору та генетики, які стали одними з найбільш поширених типів ЕА. Вони ефективні для широкого спектру задач оптимізації, оскільки їх основні

механізми включають використання операторів відбору, кросинговеру (або рекомбінації) і мутації для еволюції популяції рішень.

ЕС були розроблені спеціально для задач безперервної оптимізації. Тут кожного індивіда в популяції представляє пара векторів реальних чисел, що вказує на рішення та на його мутаційний розмах. Вимірювання є декілька відмінними від ГА, зокрема тим, що ЕС здебільшого використовують мутацію та рекомбінацію, але не включають кросинговер.

ЕП також використовує подібний підхід до ЕС, але зосереджується на розвитку комп'ютерних програм, хоча його розширення включає і інші типи проблем. Видається, що ЕП більше зосереджене на мутації, а рекомбінація не включена в його базову форму.

Генетичне програмування (ГП) це інший варіант ЕА, спеціально створений для побудови комп'ютерних програм або символічних виразів. ГП працює з деревами розбору, що дозволяє рішенням змінювати розмір та форму, в той час як більшість ЕА працюють над рядками символів фіксованої довжини.

Дехто може визначити ДЕ, метод точкового пошуку алгоритму, створений для глобальної оптимізації складних проблем, як ЕА. Його простота, універсальність і висока надійність роблять його особливо ефективним для задач оптимізації в реальному числі. ДЕ в основному спирається на мутацію, кросинговер і відбір.

Нарешті, хоча це технічно не є еволюційний алгоритм, оптимізацію роєм частинок (ОРЧ), іноді класифікують як ЕА. Це алгоритм пошуку на основі популяції, що імітує соціальну поведінку пташиних зграй або рибних зграй. При цьому кожен учасник популяції (зграї) оновлює свою позицію на основі свого власного досвіду та досвіду своїх сусідів.

Оскільки кожен з цих ЕА має власні переваги та особливості, деякі з них можуть бути більш ефективними у певних типах задач. Вибір ЕА для конкретних проблем, таким чином, зазвичай залежить від специфіки цих проблем.

1.5 Теорія створення штучних нейронних мереж

1.5.1 Біологічне натхнення для ШНМ

ШНМ черпають натхнення зі структури та функцій біологічного мозку.

Людський мозок складається з мільярдів взаємопов'язаних нервових клітин, відомих як нейрони, які спілкуються один з одним за допомогою електричних і хімічних сигналів. Кожен нейрон отримує сигнали від багатьох інших нейронів через дендрити, обробляє ці сигнали та передає вихідні дані іншим нейронам через свій аксон. Ця взаємопов'язана система дозволяє мозку вчитися на досвіді, розпізнавати закономірності та приймати рішення, забезпечуючи основу для інтелекту.

Подібно до структури людського мозку, ШНМ складається з великої кількості взаємопов'язаних вузлів обробки, відомих як штучні нейрони або одиниці. Кожен із цих штучних нейронів імітує основну функцію біологічного нейрона, приймаючи численні вхідні дані, обробляючи їх і створюючи вихідні дані. Те, як ШНМ навчаються на вхідних даних, відображає спосіб, яким біологічний мозок навчається на стимулах, які він отримує.

Процес навчання ШНМ аналогічний синаптичній пластичності, біологічному феномену, при якому міцність зв'язків або синапсів між нейронами змінюється з досвідом. У ШНМ міцність зв'язків між шарами штучних нейронів регулюється в процесі навчання. Це налаштування, по суті, є оптимізацією числових значень, званих вагами, які визначають внесок вхідних даних у вихід нейрона [9].

Механізм прийняття рішень одним штучним нейроном також має схожість з його біологічним аналогом. Коли біологічний нейрон отримує сигнали від вхідних синапсів, він інтегрує ці сигнали. Якщо загальна сума перевищує певний поріг, нейрон «активується» і «спрацьовує», посилюючи

сигнал далі по мережі. Подібним чином у штучному нейроні він отримує кілька вхідних сигналів, які підсумовуються. Якщо сума перевищує певне значення (визначене функцією активації), нейрон активується і передає сигнал на наступний рівень.

Однак, незважаючи на ці подібності, ШНМ є значно спрощеними моделями біологічного мозку. Складність дендритних обчислень, часова динаміка та взаємодія кількох нейромедіаторів, серед іншого, не враховуються в ШНМ. Тим не менш, як спрощена математична та обчислювальна модель мозку, ШНМ були неймовірно ефективними у широкому спектрі завдань, пов'язаних із розпізнаванням образів, прогнозуванням, класифікацією та прийняттям рішень.

1.5.2 Основні компоненти ШНМ: нейрони, ваги, зміщення, шари

ШНМ – це структуровані обчислювальні моделі, призначені для обробки інформації та прийняття рішень. Ці впливові моделі створені для імітації складних зв'язків і обчислювального потенціалу біологічного мозку.

Основою організації ШНМ є блоки обробки, відомі як нейрони, які іноді також називають вузлами або блоками. Черпаючи натхнення у своїх біологічних аналогів, кожен штучний нейрон відповідає за отримання кількох вхідних даних, їх обробку та створення відповідних вихідних даних. Ця обробка передбачає використання кількох важливих компонентів: ваги, зміщення та функцію активації.

Ваги відображають силу та ступінь впливу кожного входу на вихід нейрона. Вони служать коефіцієнтами, які множаться на окремі вхідні дані. Це регульовані параметри, які зазнають змін під час фази навчання ШНМ, щоб мінімізувати різницю між прогнозами ШНМ і фактичними даними.

Тим часом додатковий параметр нейрона, зміщення, пропонує гнучкість моделі. Концептуально подібне до перетину у лінійного рівняння,

роль зміщення полягає у зміщенні функції активації вліво або вправо, фактично дозволяючи ефективніше вивчати модель.

Після того, як кожен вхід помножено на його відповідну вагу, а результати підсумовані разом із зміщенням, результат проходить через функцію активації. Функція активації вводить нелінійність у модель, важливий компонент, що дозволяє ШНМ моделювати складні шаблони в даних. Поширені типи функцій активації включають сигмоподібну, гіперболічний тангенс (\tanh) і випрямлену лінійну одиницю (ReLU), кожна з яких привносить свої унікальні аспекти в процес навчання.

Окрім окремих нейронів, глобальна структура ШНМ охоплює шари. Перший рівень, відомий як вхідний, отримує необроблені вхідні дані. Між вхідним і кінцевим вихідним рівнем може бути один або кілька прихованих рівнів, на яких відбувається основна частина обробки та навчання. Примітно, що вибір точної кількості шарів і кількості нейронів у кожному шарі суттєво впливає на здатність ШНМ узагальнювати вивчені дані на невидимі випадки.

Усі ці компоненти об'єднують між собою зв'язки між нейронами в сусідніх шарах, кожен з яких пов'язаний зі значенням ваги. Нарешті, ще одним важливим аспектом у ШНМ є швидкість навчання. Цей гіперпараметр керує темпом навчання – змінами вагових коефіцієнтів під час фази навчання. По суті, він керує балансом між забезпеченням швидкої конвергенції та запобіганням перевищення оптимального рішення.

Кожен із цих основних компонентів ШНМ – нейрони, ваги, зміщення, функції активації, шари, зв'язки та швидкість навчання – складно взаємодіє, створюючи можливості ШНМ для навчання та прогнозування. Саме ця добре поставлена взаємодія дає змогу ШНМ ефективно моделювати складні відносини.

1.5.3 Типи ШНМ та їх відмінності

ШНМ представляють делікатно різну підмножину обчислювальних моделей, кожна з яких розроблена з унікальним дизайном, метою якого є оптимізація роботи з конкретним набором даних і структурою проблем.

Серед них – нейронні мережі прямого зв'язку, які являють собою, по суті, фундаментальні та вступні нейронні мережі. У них потік даних є односпрямованим – він рухається від вхідних вузлів до вихідних без рекурсивних циклів, формуючи орієнтований ациклічний граф. Ця мережа часто служить основою для більш комплексних типів нейронних мереж.

З іншого боку, багатошарові перцептрони (БШП) є розширенням мереж прямого зв'язку, але з принаймні одним шаром прихованих нейронів, розташованих між вхідним и вихідним рівнями. Це дає БШП можливість розробки стратегій вирішення проблем, які не можуть бути розірваними лінійно, але включають нелінійні межі.

Розроблені специфічно для обробки структурованих даних, таких як зображення, ЗНН відзначаються складною архітектурою, яка включає такі елементи, як глибокі шари, шари об'єднання та повністю зв'язані шари. Ці властивості роблять ЗНН найбільш примітними в контексті здатності виокремлювати просторові ієрархії в даних.

У той час, РНН є різновидом нейронних мереж, які впроваджують послідовність в залежності від даних, інтегруючи спрямовані цикли в свій граф зв'язків. Однак, їх практичне застосування часто призводить до істотного зменшення градієнтних сигналів з підвищенням довжини послідовностей через проблеми зі зменшенням і вибухом градієнтів.

Цікаво, що мережі довготривалої короткочасної пам'яті (LSTM) є екземпляром рекурентних нейронних мереж, у якому інтелектуально обійдено поширену в РНН проблему вивчення довгострокових залежностей. Пристрій МДКП представляє новий вид клітинки пам'яті з механізмами, що

регулюють її стан, забезпечуючи можливості вивчення, забування, читання та запису інформації.

Мережі радіальної базисної функції (РБФ) відрізняються трирівневою структурою. Вони зазвичай складаються з вхідного рівня, прихованого рівня з нелінійною активацією та лінійного вихідного шару. Ця нелінійність перетворень в прихованому шарі вирізняє структури РБФ мережі від БШП и ЗНН [10].

У чому ж значення всіх цих варіацій штучних нейронних мереж? Кожен впроваджує унікальні сильні й слабкі сторони, які впливають на їх природну придатність для обробки певних типів проблем або структур даних. Наприклад, БШП природнім чином стає оптимальним рішенням для використання на простих завданнях, тоді як завдання, пов'язані з обробкою візуальних даних, обов'язково вимагають використання ЗНН. Важливо пам'ятати, що вибір архітектури ШНМ часто залежить від ретельного аналізу особливостей задачі та наявних даних.

1.6 Оптимізація параметрів в ШНМ

1.6.1 Важливість оптимізації параметрів у ШНМ

ШНМ – це складні моделі, які окреслюють складні нелінійні відображення від безлічі входів до бажаних результатів. Центральними для оптимального функціонування ШНМ є параметри мережі, головним чином ваги та зміщення, які стратегічно коригуються на етапі навчання. Цей процес ретельного налаштування параметрів, щоб скоротити відхилення між прогнозами моделі та фактичними даними, по суті, є суттю оптимізації параметрів.

Оптимізація параметрів укладає в собі суть машинного навчання і принципово необхідна для ШНМ. Ефективність ШНМ у досягненні високої

точності прогнозування та забезпеченні значних можливостей узагальнення значною мірою залежить від налаштованої оптимізації її параметрів.

На ширшому полотні оптимальні значення параметрів дозволяють ШНМ ефективно фіксувати основні шаблони та залежності у вхідних даних, таким чином дозволяючи створювати відповідну функцію відображення входів і виходів. Недостатньо оптимізовані параметри можуть завдати суттєвої шкоди, що призведе до двох крайніх сценаріїв. Недостатнє пристосування являє собою сценарій надмірного спрощення, коли модель не в змозі достатньою мірою інкапсулювати притаманну складність даних, що робить її нездатною до узагальнення за межами навчальних даних. Протилежний кінець, перенавчання, зображує стан, коли модель складно узгоджується з даними навчання, таким чином втрачаючи свою здатність ефективно узагальнювати нові невидимі дані.

Продовжуючи дискурс, важливо визнати, що процес коригування параметрів усередині ШНМ не є ізольованим завданням. Примітно, що він узгоджується зі структурою самої ШНМ і характером застосовуваних функцій активації. Наприклад, глибші мережі з надмірними параметрами можуть сформулювати дуже складні представлення, але, навпаки, їх важче оптимізувати та ризикують бути схильними до переобладнання.

Щоб успішно уникнути цих пасток, необхідні правильні методи регуляризації в поєднанні з оптимальним налаштуванням параметрів. Таким чином, оптимізація параметрів є критично важливим завданням, яке поєднує в собі точність, ефективність і здатність до узагальнення ШНМ, фундаментально регулюючи рівень успіху цих моделей у численних практичних застосуваннях, починаючи від комп'ютерного зору до обробки природної мови.

1.6.2 Традиційні методи оптимізації параметрів

Оптимізація параметрів є основою машинного навчання, до неї традиційно підходили за допомогою кількох звичайних методів, зокрема тих, що базуються на парадигмах математичної оптимізації.

Квінтесенцією методу є алгоритм градієнтного спуску, який мінімізує функцію помилок шляхом ітеративного коригування параметрів у напрямку, протилежному градієнту цієї функції. Суть цього методу полягає в спуску функції до локального мінімуму, керуючись негативним вектором градієнта.

Базовий метод градієнтного спуску зазнав еволюції в різні складні варіанти, враховуючи різні фактори для покращення простоти базового алгоритму. Стохастичний градієнтний спуск (СГС), наприклад, обчислює градієнт, використовуючи один чи кілька навчальних прикладів, таким чином забезпечуючи ефективність обчислень і властиву здатність уникнути неглибоких локальних мінімумів. Однак його висока дисперсія часто призводить до проблем конвергенції, прокладаючи шляхи для вдосконалень, таких як СГС на основі Momentum, RMSProp і Adam, які включають концепції імпульсу та адаптивних темпів навчання для протидії цим невдачам.

З іншого боку, методи другого порядку, влучно відомі як методи Ньютона, використовують похідні другого порядку або кривизну функції втрат для коригування швидкості навчання під час оптимізації. Теоретично ці методи збігаються швидше, але складність реалізації та вартість обчислень часто обмежують їх широке використання.

Доповнюючи ці градієнтні методи, методи прямого пошуку, такі як пошук у сітці та випадковий пошук, періодично використовуються для оптимізації гіперпараметрів ШНМ. Пошук у сітці використовує вичерпний пошук у вручну визначеній підмножині гіперпараметрів, пропонуючи рішення грубої сили, яке гарантує пошук оптимальних параметрів у

вибраному діапазоні, але часто є дорогим з точки зору обчислень. Навпаки, випадковий пошук відбирає параметри випадковим чином із розподілу в заданому діапазоні, пропонуючи конкурентоспроможну альтернативу з нижчими обчислювальними витратами без зобов'язання щодо фіксованого набору значень гіперпараметрів [11].

Поряд із детерміністськими методами традиційно використовуються імовірнісні підходи для оптимізації гіперпараметрів, байєсівська оптимізація є помітною технікою серед них. Ця техніка створює імовірнісну модель для функції, що відображає гіперпараметри на продуктивність набору перевірки, а потім використовує функції отримання для побудови функції корисності, зрештою вирішуючи, які гіперпараметри вибрати в наступному раунді навчання та оцінювання.

По суті, класичні методи оптимізації параметрів складно збалансують компроміси між пошуком оптимальної області в просторі параметрів і необхідними обчислювальними ресурсами. Кожен метод має унікальні переваги та обмеження, розширюючи інструментарій для розробників моделей для підходу до завдань оптимізації параметрів відповідно до вимог і характеристик проблеми, що розглядається.

1.7 Застосування еволюційних алгоритмів в оптимізації

1.7.1 Як EA можна використовувати для оптимізації параметрів

EA відкривають новий і переконливий підхід до складної проблеми оптимізації параметрів у ШНМ. Черпаючи натхнення з природної біологічної еволюції, ці процедури пошуку та оптимізації емулюють механізми відбору, мутації та кросинговеру, ефективно маневруючи в просторі параметрів і зближуючись до оптимальних або майже оптимальних рішень.

На відміну від традиційних методів оптимізації, які часто ґрунтуються на існуванні градієнтів або опуклості, ЕА застосовують евристичну стратегію пошуку, більш орієнтовану на населення. Вони підтримують популяцію потенційних рішень, які розглядаються як окремі особи, і використовують стохастичні оператори для дослідження та використання простору пошуку. Зокрема, за допомогою механізмів, що нагадують біологічне розмноження та виживання, ці алгоритми вибирають особин на основі функції пристосованості, рекомбінують пари особин для створення потомства та індукують невеликі випадкові варіації для сприяння різноманітності.

В основі впровадження ЕА для оптимізації параметрів у ШНМ лежить представлення індивідуумів у популяції, зазвичай закодованих у вигляді рядка чисел, кожне з яких відповідає певній вазі або зміщенню у мережі. Отже, точність і межі, в яких можна налаштувати параметри, внутрішньо регулюються деталями цього подання.

Коли ЕА досліджують простір параметрів, вони виявляють властивий баланс між розвідкою, процесом дослідження нових регіонів у просторі, та експлуатацією, прагненням до вдосконалення відомих на даний момент перспективних регіонів. Циклічний процес відбору, розмноження та мутації ефективно сприяє цьому балансу, борючись із небезпечними пастками локальних мінімумів, з якими часто стикаються звичайні методи на основі градієнта.

Крім того, стратегія глобального паралельного пошуку, прийнята ЕА, посилює їх здатність до розподілених обчислювальних середовищ і багатовимірних невивуклих проблем. Одночасна оцінка кількох рішень надає ширший погляд на фітнес-ландшафт, спритно обходячи потребу в похідній інформації та забезпечуючи ефективну навігацію навіть в умовах розривів, шуму та кількох оптимумів.

Застосування ЕА для оптимізації параметрів також відкриває шляхи для оптимізації структурних елементів мережі, таких як кількість

прихованих шарів, кількість нейронів у кожному шарі та тип функції активації для використання. Ця здатність спільно розвивати параметри разом із топологією мережі добре адаптується до теореми про відсутність безкоштовного обіду, припускаючи, що жодна конфігурація мережі не працює найкраще для всіх можливих проблем.

Таким чином, застосування еволюційних алгоритмів для оптимізації параметрів у ШНМ забезпечує надійну, гнучку альтернативу традиційним методам, яка за своєю суттю може розпаралелюватися. Імітуючи цілісні, розподілені стратегії пошуку, природні для біологічної еволюції, ЕА ефективно створюють складний ландшафт багатовимірних невивуклих завдань оптимізації, повних навчання та розгортання нейронної мережі.

1.7.2 Переваги та обмеження використання ЕА для оптимізації

ЕА у своїй схильності імітувати біологічну еволюцію, представляють унікальну перспективу для вирішення важкого завдання оптимізації параметрів у ШНМ. Цей чіткий підхід несе з собою низку привабливих переваг, переплетених із властивими обмеженнями, які заслуговують на сумлінне розглядання.

Однією з ключових переваг використання ЕА для оптимізації є їхня здатність орієнтуватися в складних, багатовимірних і невивуклих просторах пошуку, де традиційні методи часто дають збій. Підтримуючи популяцію потенційних рішень і використовуючи стохастичні оператори, ЕА можуть ефективно збалансувати розвідку та експлуатацію, дозволяючи їм обійти локальні оптимуми та розкрити глобальні оптимуми. Їхня стохастична природа, яка охоплює такі механізми, як мутація, сприяє різноманітності в популяції, захищаючи алгоритм від передчасної конвергенції та підвищуючи надійність рішення [12].

Глобальний характер процесу пошуку, що виконується ЕА, надає ще одну явну перевагу – можливість природно розпаралелювати обчислення.

Кожна особина в популяції представляє незалежне потенційне рішення, і тому їхню оцінку придатності можна обчислювати одночасно. Цей паралелізм добре узгоджується з сучасною обчислювальною інфраструктурою та розподіленими середовищами, прискорюючи процес оптимізації.

Крім того, ЕА особливо компетентні в обробці суміші безперервних, дискретних і категоріальних параметрів одночасно, щось менш поширене для звичайних методів оптимізації. Ця гнучкість також поширюється на задачі комбінаторної та багатоцільової оптимізації, зміцнюючи ЕА як універсальні інструменти оптимізації.

Незважаючи на те, що ЕА надають кілька переконливих перспектив, вони також мають властиві обмеження, якими потрібно розсудливо керувати. Одним із помітних обмежень є обчислювальна вартість. Популяційний характер ЕА, хоча і корисний для глобального пошуку, збільшує вимоги до обчислень порівняно з методами оптимізації з одним рішенням. Ця проблема загострюється для більших розмірів популяції та більш складних функцій оцінки придатності.

Ще одне потенційне обмеження полягає у відсутності остаточного критерію зупинки, насамперед через стохастичний характер цих методів. Визначити, коли зупинити еволюційний процес без передчасного згортання досліджень, може бути нетривіально.

Крім того, відповідна параметризація самого алгоритму, наприклад розмір популяції та швидкість мутації, може суттєво вплинути на продуктивність ЕА. Однак пошук оптимальних налаштувань для цих мета-параметрів часто є другорядним завданням оптимізації.

Підсумовуючи, хоча сильні сторони ЕА, такі як можливість глобального пошуку, властивий паралелізм і гнучкість, роблять їх потужними суперниками для завдань оптимізації параметрів, їх обмеження, зокрема вищі обчислювальні витрати, відсутність остаточних критеріїв зупинки та чутливість до параметрів алгоритму, вимагають усвідомленого

вибору впровадження. Використання сильних сторін при одночасному розумному управлінні обмеженнями може забезпечити ефективне використання ЕА для оптимізації складних ландшафтів параметрів, характерних для штучних нейронних мереж.

1.7.3 Приклади ЕА в оптимізації

ЕА знайшли широке застосування у вирішенні різноманітних проблем оптимізації. Вони були демонстративно застосовані в різних доменах, демонструючи їх універсальність і надійність. Кілька тематичних досліджень підкреслюють успішне розгортання цих алгоритмів, сприяючи як науковим колам, так і промисловості.

Значною областю, де ЕА показали надзвичайні перспективи, є сфера машинного навчання та, зокрема, оптимізація ШНМ. Тут ЕА були розгорнуті не тільки для точного налаштування параметрів моделі, але й для оптимізації структури та конфігурації мережі. Один із таких прикладів майстерно продемонстрували Стенлі К. О. та Мійккулайнен Р. у своїй роботі «Нейроеволюція розширених топологій», де вони застосували генетичний алгоритм для одночасного виявлення відповідних ваг і відповідної архітектури для нейронної мережі. Алгоритм з тих пір використовувався в численних програмах, починаючи від завдань навчання з підкріпленням і закінчуючи проблемами керування [13].

Крім того, у сфері обробки зображень і комп'ютерного зору ЗНМ, оптимізовані за допомогою ЕА, показали високу продуктивність у кількох завданнях. Стратегії на основі ЕА, такі як еволюційний пошук мережевої архітектури (ESNAC), були вміло використані для визначення конфігурації ЗНМ, включаючи кількість згорткових шарів, порядок шарів, розміри фільтрів тощо.

Окрім машинного навчання, ЕА досягли значних успіхів у логістиці та управлінні ланцюгами поставок, зокрема у вирішенні складних проблем

оптимізації. Одним із таких прикладів є проблеми з маршрутизацією транспортного засобу, коли генетичні алгоритми успішно використовуються для послідовності та планування маршрутів, щоб мінімізувати загальний час або вартість транзиту.

У сфері енергетики ЕА знайшли вирішальне застосування в оптимізації систем відновлюваної енергії. Наприклад, ЕА використовувалися для оптимізації параметрів і стратегій керування фотоелектричними енергетичними системами у відповідь на зміну умов навколишнього середовища, демонструючи потенціал у досягненні високої енергоефективності та стійкості.

ЕА також переважно займаються телекомунікаціями для оптимізації розміщення антен і формування променя в бездротових мережах. Генетичні алгоритми були серед ЕА, які були прийняті для стратегічного розміщення антен таким чином, щоб трафік сигналу ефективно розподілявся при мінімізації перешкод.

Ці тематичні дослідження зображують лише короткий погляд на широку застосовність еволюційних алгоритмів у різноманітних секторах. Завдяки своїй унікальній здатності орієнтуватися в складних, багатовимірних просторах, ЕА ефективно вирішували безліч проблем оптимізації, що охоплюють машинне навчання, логістику, енергетичні системи та телекомунікації, серед іншого, закріпивши своє місце як надійного інструменту в наборі інструментів оптимізації.

1.8 Висновок розділу

Підсумовуючи, ШНМ є наріжним каменем у будівлі машинного навчання, сприяючи нашій здатності розпізнавати закономірності та приймати рішення в умовах невизначеності. Їхня корисність поширюється на різноманітні сфери, від розпізнавання зображень до обробки природної мови, від прогнозної аналітики до систем керування тощо. Центральними

для цих мереж є їхні внутрішні компоненти, такі як нейрони, ваги, зміщення та функції активації. Крім того, їхня складність, яка визначається рівнями, з'єднаннями та швидкістю навчання, впливає на здатність ШНМ моделювати складні зв'язки та узагальнювати отримані дані на невидимі екземпляри.

Пошуки кращих методів оптимізації невпинні: естафета переходить від простих математичних методів до складних стохастичних алгоритмів, натхненних біологією, і далі до гібридних та адаптивних підходів, посиленних досягненнями в обчислювальній техніці. Реальність того, що жоден єдиний метод не буде універсально найкращим, як викладено в «Теоремі про відсутність безкоштовного обіду», гарантує, що галузь продовжить досягати успіхів у розробці інноваційних алгоритмів, надаючи практикам дедалі більший набір інструментів для боротьби зі зростаючим вимоги, викликані зростаючим масштабом і складністю поточних проблем.

Оптимізація параметрів ШНМ є ключовою проблемою для покращення їх продуктивності та точності. Традиційні методи оптимізації часто застрягають в локальних мінімумах складного ландшафту параметрів ШНМ. ЕА пропонують альтернативний підхід, який використовує принципи еволюції для подолання цієї проблеми та пошуку глобального оптимуму.

Для досягнення мети дослідження було використано наступні методи:

- огляд літератури з ЕА, ШНМ, оптимізації параметрів та їх поєднання;
- розробка теоретичної основи ЕА та ШНМ, включаючи їх біологічні принципи, характеристики та різні типи;
- аналіз методів оптимізації параметрів, традиційних та заснованих на ЕА.
- дослідження практичного застосування ЕА для оптимізації параметрів ШНМ, з оцінкою їх переваг та обмежень.

Інтеграція ЕА з ШНМ є перспективним полем досліджень, яке може призвести до більш ефективних та надійних моделей. Подальші дослідження можуть бути спрямовані на розробку нових алгоритмів ЕА, адаптивну настройку параметрів та гібридизацію з іншими методами оптимізації.

Практичні наслідки застосування ЕА для оптимізації ШНМ може мати значний вплив на різноманітні сфери, включаючи розпізнавання образів, прогнозування, обробку сигналів та багато інших. Ефективніші нейронні мережі можуть покращити точність та продуктивність цих застосувань.

Дослідження взаємодії між ЕА та ШНМ є багатообіцяючим напрямком. Розробка нових алгоритмів ЕА, оптимізація їх параметрів та інтеграція з іншими методами оптимізації можуть призвести до значного прогресу в цій галузі.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Вступ

У центрі теоретичного розділу дипломної роботи знаходиться розгляд фундаментальних концепцій та принципів, які лежать в основі розробки та оптимізації штучних нейронних мереж за допомогою еволюційних алгоритмів. Перш за все, розглядається історичний контекст виникнення та розвитку штучних нейронних мереж, їхні архітектурні особливості та основні моделі, серед яких особливе місце займають мережі LSTM, що здатні ефективно обробляти послідовності даних з урахуванням часових залежностей.

Далі увага приділяється аналізу еволюційних алгоритмів, їхній природі та механізмах роботи, а також огляду сучасних методів їх застосування у сфері машинного навчання. Визначальним аспектом розділу є детальний розгляд процесу оптимізації гіперпараметрів нейронних мереж, що включає аналіз різних стратегій настройки та їх вплив на продуктивність моделей.

2.2 Опис генетичного алгоритму

ГА використовується у цьому дослідженні як ключовий інструмент для вивчення ваг нейронних мереж та оптимізації їх архітектури.

ГА є біоінспірованим методом пошуку та оптимізації, що імітує процес природного відбору. Він працює на основі популяції потенційних рішень і використовує механізми, такі як селекція, схрещування та мутація, для поступового покращення якості рішень.

У контексті цього дослідження, генетичний алгоритм використовується для пошуку оптимальних ваг нейронних мереж. Потенційні рішення (тобто набори ваг) кодується як хромосоми. Процес

декодування потім використовується для перетворення цих хромосом назад у ваги нейронної мережі.

Оцінка пристосованості використовується для визначення якості кожного потенційного рішення. В цьому дослідженні вона базується на точності передбачення нейронної мережі. Після оцінки, селекційний процес вибирає найкращі рішення для участі у схрещуванні та мутації.

Схрещування (або рекомбінація) – це процес, під час якого беруться частини двох хромосом-батьків і комбінуються для створення нової хромосоми-нащадка. Мутація, з іншого боку, вносить невеликі випадкові зміни в хромосому для забезпечення різноманітності популяції.

Завершує процес ініціалізація, під час якої створюється початкова популяція потенційних рішень. Цей процес може бути випадковим або використовувати попередньо визначені ваги [14].

Генетичний алгоритм застосовується для точного налаштування нейронних мовних моделей. Моделі є багатошаровими на основі LSTM і слабко базуються на моделях, використаних на веб-сторінці підручника Tensorflow, які, в свою чергу, слідує за моделлю, описаною Заремба та ін. в [13]. Вони складаються з вхідного шару, за яким слідує шар вбудовування, два приховані шари LSTM та шар відкидання, і, нарешті, щільний шар та шар активації Softmax (рисунок 2.1).

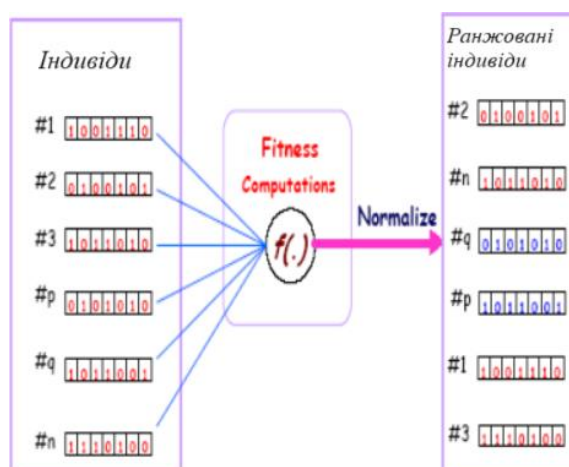


Рисунок 2.1 – Функція придатності

Загалом, використання генетичного алгоритму дозволяє ефективно вивчити простір ваг нейронної мережі та знайти оптимальну архітектуру мережі(рисунок 2.2):

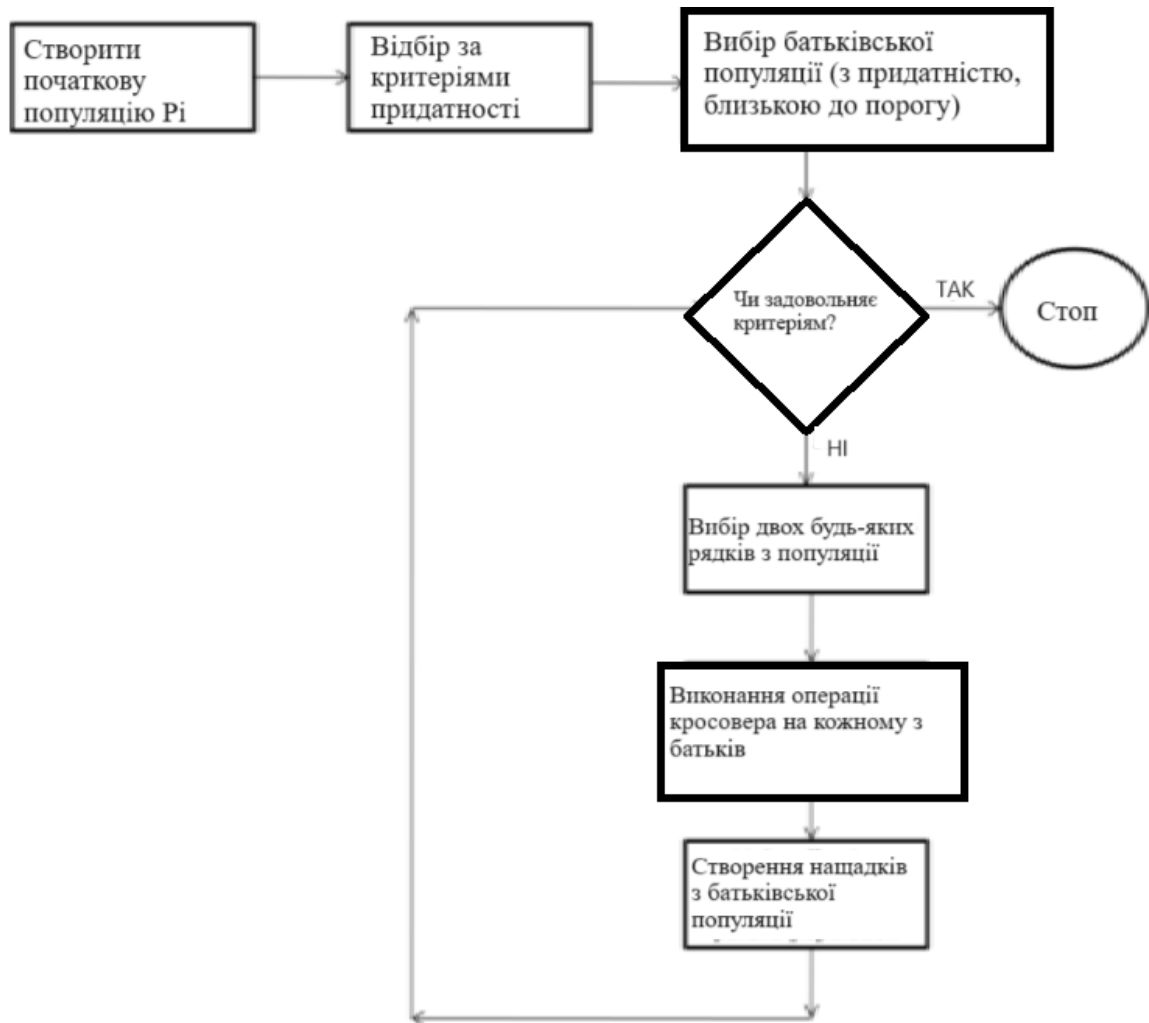


Рисунок 2.2 – Блок-схема ГА

2.3 Еволюція навчальних правил у штучних нейронних мережах

Алгоритм навчання штучної нейронної мережі може мати різну продуктивність при застосуванні до різних архітектур. Проектування алгоритмів навчання, а більш глибоко – правила навчання, що використовуються для налаштування ваги зв'язків, залежить від типу архітектур і завдань навчання, які досліджуються. Після вибору алгоритму

навчання, все ще залишаються параметри алгоритму, такі як швидкість навчання та імпульс у алгоритмах зворотного поширення, які потрібно визначити. Наприклад, генетичні алгоритми підходять для навчання штучних нейронних мереж зі зворотніми зв'язками та глибокими прямими штучними нейронними мережами (з багатьма прихованими шарами), тоді як зворотне поширення ефективно для навчання поверхневих. В даний час цей вид пошуку оптимального (або близького до оптимального) правила навчання може бути здійснений тільки деякими експертами через їхній досвід та метод проб та помилок. Насправді, від штучної нейронної мережі потрібна її здатність адаптивно налаштовувати її правило навчання відповідно до її архітектури та завдання, яке потрібно виконати. Оскільки еволюція є однією з найбільш фундаментальних форм адаптації, то, можливо, еволюція може сприяти розвитку відповідного типу правила навчання для даного застосування; для чого також може бути використано факт, що відносини між еволюцією та навчанням є надзвичайно складними. Було запропоновано різні моделі, але більшість з них розглядають питання того, як навчання може керувати еволюцією та відносини між еволюцією архітектур та ваг зв'язків. Дослідження щодо еволюції правил навчання все ще знаходиться на ранніх стадіях, дивитись, наприклад, [15]. Це дослідження важливе не тільки для надання автоматичного способу оптимізації правил навчання та моделювання відносин між навчанням та еволюцією, але також для моделювання креативного процесу, оскільки новостворені правила навчання можуть впоратися зі складним та динамічним середовищем. Адаптивне налаштування параметрів алгоритму через еволюцію можна розглядати як першу спробу еволюції правил навчання, наприклад, в [16] закодовані параметри зворотного поширення у хромосомах разом з архітектурою штучної нейронної мережі. Еволюція параметрів алгоритму, безумовно, цікава, але вона майже не торкається фундаментальної частини алгоритму навчання, тобто його правила навчання або правила оновлення ваги. Очікується, що адаптація правила

навчання через еволюцію значно покращить адаптивність штучної нейронної мережі в динамічному середовищі. Набагато важче кодувати динамічні поведінки, такі як правило навчання, ніж кодувати властивості, такі як архітектура та ваги зв'язків, штучної нейронної мережі. Ключовим питанням тут є те, як закодувати динамічну поведінку правила навчання у статичні хромосоми. Спроба розробити універсальну схему представлення, яка може вказати будь-який вид динамічних поведінок, явно непрактична, не кажучи вже про заборонений довгий час обчислення, необхідний для пошуку в такому просторі правил навчання. Потрібно встановити обмеження на тип динамічних поведінок, тобто на базову форму правил навчання, які еволюціонують, щоб зменшити складність представлення та простір пошуку. Два базові припущення, які часто робляться щодо правил навчання, є:

- оновлення ваги залежить тільки від локальної інформації, такої як активація вхідного нейрона, активація вихідного нейрона, поточна вага зв'язку тощо;

- правило навчання є таким же для всіх зв'язків у штучній нейронній мережі. Припускається, що правило навчання є лінійною функцією цих локальних змінних та їхніх продуктів. Тобто, правило навчання може бути описане функцією (2.1):

$$\Delta w(t) = \sum_{k=1}^n \sum_{i_1, i_2, \dots, i_k=1}^n \left(\theta_{i_1, i_2, \dots, i_k} \prod_{j=1}^k x_{i_j}(t-1) \right), \quad (2.1)$$

де t – час,

Δw – зміна ваги,

x_1, x_2, \dots, x_n – локальні змінні,

θ – дійсні коефіцієнти, значення яких будуть визначені шляхом еволюції.

Іншими словами, еволюція навчальних правил у цьому випадку еквівалентна еволюції дійсно значних векторів коефіцієнтів θ . Основною метою еволюції навчальних правил є визначення цих коефіцієнтів. Різні θ

визначають різні навчальні правила. Через велику кількість можливих термінів у функції (2.1), що могли б зробити еволюцію дуже повільною та непрактичною, на практиці використовувалися лише кілька термінів відповідно до деяких біологічних або евристичних знань [17].

2.4 Основи нейронної мережі

Розглянемо основи нейронної мережі. Нейронна мережа є обчислювальною моделлю, що складається з низки взаємопов'язаних елементів, відомих як нейрони. Нейрон є обчислювальною одиницею, яка отримує вхідні дані ззовні мережі та/або від інших нейронів, застосовує локальну трансформацію до цих даних і надає вихідний сигнал, який передається іншим нейронам та/або зовнішньому середовищу. Кожен із вхідних сигналів модифікується значенням, асоційованим із зв'язком. Це значення називається силою зв'язку або вагою і, загалом кажучи, представляє, наскільки важливим нейрон вважає це джерело вхідного сигналу. Локальна трансформація називається активаційною функцією і зазвичай має сигмоїдальний характер.

Штучна нейронна мережа складається з багатьох штучних нейронів, які з'єднані відповідно до специфічної архітектури мережі. Метою нейронної мережі є перетворення вхідних даних на осмислені вихідні сигнали.

Основні елементи або блоки штучної нейронної мережі такі:

- обчислювальний елемент (названий штучним нейроном або просто нейроном);
- схема з'єднань між елементами (структура або архітектура);
- процес навчання нейронної мережі (алгоритм навчання).

Важлива різниця між ШНМ та ГА. У генетичному алгоритмі лише ті дані, які мають цінність у передбаченні вихідних результатів, зберігаються як вхідні дані для системи. Нейронна мережа, з іншого боку, не виключає нерелевантні вхідні дані з фінальної системи. Вона нівелює вплив таких даних, присвоюючи їм низьку вагу у процесі прийняття рішень.

2.5 Архітектура та процес навчання нейронної мережі

Нейронна мережа зворотного поширення помилок (ВР) є узагальненням дельта-правила, яке використовується для тренування багат шарових перцептронів з нелінійними елементами. Це просто метод градієнтного спуску, розроблений для мінімізації загальної помилки (або середньої помилки) виводу, обчисленого мережею. Архітектура нейронної мережі ВР представлена наступним чином.

Рисунок 2.3 показує таку мережу.

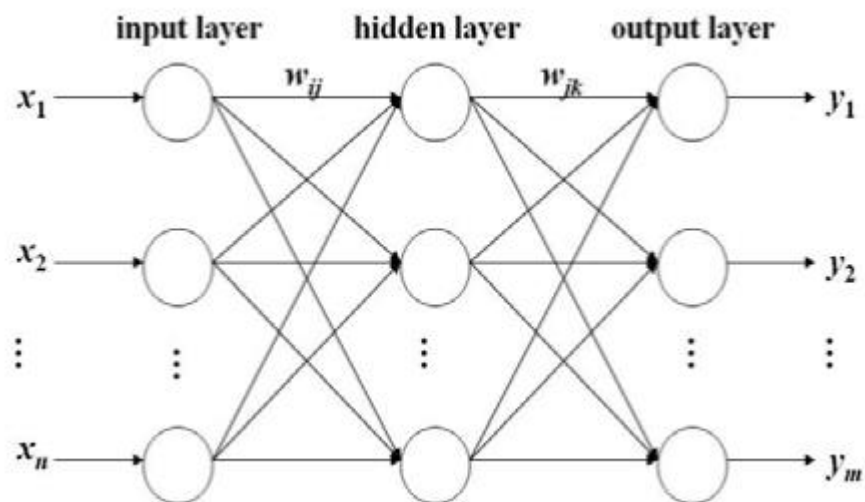


Рисунок 2.3 – архітектура нейронної мережі

У мережі є вхідний шар, вихідний шар, а між ними – один або кілька прихованих шарів.

Припустимо, що в мережі є n входів та m виходів, s нейронів у прихованому шарі, вивід прихованого шару – це b_j , порогове значення прихованого шару – θ_j , порогове значення вихідного шару – θ_k , функція передачі прихованого шару – $1f$, функція передачі вихідного шару – $2f$, вага від вхідного шару до прихованого шару – w_{ij} , вага від прихованого шару до

вихідного шару – w_{jk} , тоді ми можемо отримати вивід мережі y_k , бажаний вивід – t_k , вивід j -го нейрону прихованого шару b_j :

$$b_j = f_1\left(\sum_{i=1}^n w_{ij}x_i - \theta_j\right), \quad (2.2)$$

де $i = 1, 2, \dots, n$,

$j = 1, 2, \dots, s$.

Обчислюємо вивід k_y вихідного шару, тобто:

$$y_k = f_2\left(\sum_{j=1}^s w_{jk}b_j - \theta_k\right), \quad (2.3)$$

де $j = 1, 2, \dots, s$,

$k = 1, 2, \dots, m$.

Визначаємо функцію помилки за фактичним виводом мережі, тобто:

$$e = \sum_{k=1}^m (t_k - y_k)^2. \quad (2.4)$$

Тренування мережі є безперервним процесом коригування ваги та порогового значення, щоб зменшити помилку мережі до заданого мінімуму або зупинити на заданому кроці тренування. Потім вводять прогнозовані зразки до навченої мережі та отримують прогнозовані результати.

2.6 Методологія

Простий ГА працює шляхом випадкової генерації початкової популяції рядків, яку називають генетичним пулом, і подальшого застосування (можливо, трьох) операторів для створення нових, і, бажано, кращих популяцій у наступних поколіннях.

Перший оператор – це відтворення, коли рядки копіюються до наступного покоління з певною ймовірністю на основі значення їх об'єктивної функції.

Другий оператор – кросовер, коли випадково обрані пари рядків спарюються, створюючи нові рядки.

Третій оператор – мутація, яка є випадковою зміною значення у певній позиції рядка.

Оператор кросоверу разом із відтворенням є найпотужнішим процесом у пошуку ГА. Мутація диверсифікує простір пошуку та захищає від втрати генетичного матеріалу, що може бути спричинено відтворенням та кросовером. Тому ймовірність застосування мутації встановлюється дуже низькою, тоді як ймовірність кросоверу встановлюється дуже високою.

Генетичні алгоритми більш доцільно вважати оптимізаційною технікою, заснованою на природній еволюції. Вони включають ідею виживання найбільш пристосованих. Ідея полягає в тому, щоб спочатку «вгадати» рішення, а потім поєднувати найбільш пристосовані рішення для створення нового покоління рішень, яке має бути кращим за попереднє.

Спочатку створюється початкова популяція, а потім оцінюється придатність усіх хромосом за допомогою функції придатності. Після розрахунку придатності хромосома з придатністю, близькою до порогового значення, обирається як батько для наступного покоління. Якщо вона задовольняє критерії, процес зупиняється, в іншому випадку обираються будь-які два рядки з початкової популяції і виконується кросовер для створення нащадків, і знову перевіряється, чи задовольняє вона критерії. Якщо так, процес зупиняється, інакше цикл продовжується, поки не буде досягнуто задовільного результату.

Операції, що виконуються в ГА:

– генетичне кодування: щоб застосувати ГА для будь-якої оптимізаційної задачі, необхідно знайти спосіб кодування рішень як придатних хромосом так, щоб кросовери придатних хромосом призводили до створення придатних хромосом. Техніки кодування рішень залежать від задачі та включають певний рівень мистецтва;

– функція придатності: ГА використовуються для задач максимізації. Для задачі максимізації функція придатності є такою ж, як і цільова функція. Однак для задач мінімізації один зі способів визначення функції придатності є наступним:

$$F(x) = \frac{1}{f(x)+c}, \quad (2.5)$$

де $f(x)$ – є цільовою функцією;

$F(x)$ – функція пристосованості;

c – невелика константа, що додається для уникнення ділення на нуль;

– процес відбору: у процесі відбору хромосоми копіюються в наступне покоління з ймовірністю, пов'язаною з їхньою придатністю. Присвоюючи наступному поколінню більшу частину високопридатних хромосом, відтворення імітує дарвінівське виживання найбільш пристосованих у природному світі. У цій роботі автор використовує метод елітизму для відбору. Елітизм – це метод, який спочатку копіює найкращу хромосому (або кілька найкращих хромосом) до нової популяції. Решта робиться класичним способом. Елітизм може дуже швидко підвищити ефективність ГА, оскільки запобігає втраті найкращого знайденого рішення;

– оператор кросоверу: пошук у просторі рішень здійснюється шляхом створення нових хромосом із старих. Найважливішим процесом пошуку є кросовер. Спочатку випадково обирається пара батьків із матричного пулу. По-друге, обирається точка, названа місцем кросоверу, уздовж їх загальної довжини, потім перед точкою кросоверу використовується метод послідовного конструктивного кросоверу, а інформація після місця кросоверу у двох батьківських рядках обмінюється місцями. Якщо ген вже був скопійований у нащадка, його замінюють на невідвіданий ген, створюючи таким чином двох нових дітей.

Переваги використання ГА це в першу чергу генерування кращої популяції з «хороших» батьків, що наближається до глобального оптимуму. Також надійність є важливою характеристикою ГА. І ГА добре працюють у різних галузях, таких як розпізнавання шаблонів, розпізнавання мови, тексту; машини, що здатні навчатися; оптичне розпізнавання символів; виявлення шахрайських кредитних карток; стиснення зображень.

Але звісно ж існують і недоліки. ГА залишається «чорною скринькою», яка після подачі вхідних даних виробляє вихідні. Однак їх чудові результати можуть компенсувати цей недолік. Другий недолік полягає в тому, що вхідні дані потрібно змінювати перед подачею в мережу. І також вони не здатні визначити наступне: яку мережу (архітектуру) використовувати, скільки прихованих шарів, скільки нейронів, які активаційні функції слід використовувати, яку цільову функцію найбільш доцільно застосувати, який алгоритм навчання застосувати?

2.7 Послідовність використання ГА у ШНМ

ЕА представляють собою клас стохастичних методів пошуку та оптимізації, які отримані на основі природного добору та генетики. Вони є алгоритмами, заснованими на популяції, шляхом моделювання природної еволюції біологічних систем. Індивіди в популяції конкурують та обмінюються інформацією один з одним. Існують три основні генетичні операції: відбір, хрестовину та мутацію.

Процедура типового ЕА виконується за такими кроками:

- 1) встановити $t = 0$;
- 2) рандомізувати початкову популяцію $P(t)$;
- 3) оцінити пристосованість кожного індивіда $P(t)$;
- 4) вибрати індивідів як батьків з $P(t+1)$ на основі пристосованості;
- 5) застосувати оператори пошуку (хрестовину та мутацію) до батьків, і згенерувати $P(t+1)$;

б) встановити $t = t + 1$;

7) повторити кроки 3–6 до тих пір, поки не буде задоволено критерій завершення.

ЕА є стохастичними процесами, які виконують пошук у складному та багатомодовому просторі. Вони є алгоритмами, заснованими на популяції, шляхом моделювання природної еволюції біологічних систем.

Вони мають наступні переваги:

– ЕА можуть надійно та швидко вирішувати складні завдання. Вони підходять для оцінювальних функцій, які є великими, складними, неперервними, недиференційованими та багатомодовими;

– підхід ЕА є загального призначення, він може бути безпосередньо з'єднаний з існуючими симуляціями та моделями. ЕА є розширюваними та легкими для гібридизації;

– ЕА є спрямованим стохастичним глобальним пошуком. Вони завжди можуть досягнути близького до оптимального або глобального максимуму;

– ЕА володіють внутрішнім паралелізмом, оцінюючи кілька точок одночасно.

ЕА в основному включають ЕС, еволюційне програмування ЕП та ГА. Ці три еволюційні алгоритми співпадають з метою використання механізму біологічної еволюції для покращення здатності використовувати комп'ютери для вирішення проблем, але вони відрізняються за конкретною мірою: ЕС акцентує зміну поведінки на рівні індивіда, ЕП акцентує зміну поведінки на рівні популяції, а ГА акцентує операцію до хромосоми.

В додаток до цього, також існують генетичне програмування, меметичний алгоритм тощо. ГА є одним з видів ЕА, характеристиками еволюційних алгоритмів є: еволюція на основі популяції, виживання найсильніших, спрямований стохастичний, без залежності від градієнтної інформації. Це ітераційний процес обчислень, його основні кроки

включають: кодування, ініціалізацію популяції, вибір, генетичну операцію (хрестовину, мутацію), оцінку та рішення про зупинку.

Порівняно з традиційними алгоритмами оптимізації, основними характеристиками ГА є:

- стратегія пошуку на основі популяції. Традиційні алгоритми оптимізації використовують пошук від точки до точки, тоді як ГА використовує пошук від популяції до популяції, тому ГА легше досягти глобального оптимуму;

- пошук не залежить від градієнтної інформації об'єктивної функції, потрібна лише пристосованість, яка може оцінити якість індивідів. Таким чином, ГА має ширший застосування, особливо підходить для проблем, які є складними та нелінійними;

- процес еволюції є евристичним, але не сліпим;

- він простий, загального призначення та надійний.

Одним з аспектів використання ГА в нейронних мережах є використання ГА для вивчення ваг нейронних мереж, що означає використання ГА для заміни деяких традиційних алгоритмів навчання, щоб подолати їхні недоліки. Традиційне тренування ваг мережі зазвичай використовує метод градієнтного спуску; ці алгоритми легко застрягають в локальному оптимумі і не можуть дістатися глобального оптимуму [18]. Навчання ваг за допомогою певного еволюційного алгоритму може знайти набір ваг, який наближається до глобального оптимуму, при цьому не потрібно обчислювати градієнтну інформацію; пристосованість кожного індивіда може бути визначена помилкою між очікуваним вихідним значенням та фактичним вихідним значенням та складністю мережі.

Типовий алгоритм, який використовує ГА для оптимізації ваг тришарової ВР мережі, є наступним [19].

- 1) використовуйте деяку стратегію кодування для кодування зв'язків ваг і випадково генеруйте оригінальну популяцію з N індивідів, кожен індивід представляє нейронну мережу;

2) декодуйте кожного індивіда в поточному поколінні в набір зв'язків ваг і створюйте нейронні мережі з цими вагами;

3) оцініть кожну ШНМ, обчисливши її загальну помилку (згідно з деякою функцією помилки) між фактичними виходами та цільовими виходами. Приспособленість кожного індивіда визначається помилкою, а функція пристосованості залежить від проблеми;

4) виберіть кілька найбільш пристосованих індивідів, збережіть для наступного покоління;

5) використовуйте оператори пошуку, такі як хрестовину та мутацію, для обробки поточної популяції та генерації наступної популяції;

6) повторіть кроки 2-5, поки не буде задоволено умову завершення;

7) ініціалізуйте мережу ВР найкращим індивідом з ГА;

8) навчайтеся з алгоритмами навчання ВР;

9) закінчіть, коли буде задоволено умову завершення.

Псевдокод роботи ЕА у моєму коді можна побачити на рисунку 3.4.

```

1: population ← [ список з n моделей на окремих графах ]
2: generation ← 0
3: while ( generation < l ) do:
4:   train_and_evaluate( population )
5:   new_gen ← зберегти m найкращих індивідів
6:   new_gen ← додати випадкових індивідів для
                підвищення різноманітності
7:   mutate( new_gen )
8:   new_gen ← додати нащадків через схрещування до k
9:   population ← new_gen
10:  generation ← generation + 1
11:  вивести гіперпараметри найкращих індивідів у
        популяція

```

Рисунок 2.4 – Псевдокод генетичного алгоритму для оптимізації гіперпараметрів нейронних мереж

Використовуйте ГА для оптимізації ваг та архітектури нейронних мереж: Для даної проблеми, процесуальна здатність ШНМ з невеликими з'єднаннями та прихованими нейронами обмежена, але якщо з'єднання та приховані нейрони так багато, шум може бути навчений разом, і узагальнююча здатність мережі буде поганою. Раніше люди часто використовували метод спроби помилки для проектування архітектури, ефект занадто залежить від суб'єктивного досвіду [20].

Архітектура ШНМ включає з'єднання мережі та передавальні функції. Добра архітектура може вирішити проблеми задовільним способом, і не допускає існування зайвих вузлів та з'єднань. З розвитком еволюційних алгоритмів, люди розглядають проектування мережі як проблему пошуку, використовуючи точність навчання, узагальнюючу здатність та імунітет до шуму як стандарт оцінювання, знаходить найкращу архітектуру з найкращими показниками в архітектурному просторі. Розвиток архітектури еволюції в основному відображається в кодуванні архітектури та дизайні оператора.

Типові кроки використання ГА для еволюції тришарової ВР мережі є наступними:

- 1) використовуйте деяку стратегію кодування для кодування архітектур і випадково генеруйте оригінальну популяцію, з N індивідів, кожен індивід представляє нейронну мережу;

- 2) декодуйте кожного індивіда в поточному поколінні в архітектуру та створюйте відповідну нейронну мережу;

- 3) навчайте кожну ШНМ з декодованою архітектурою за попередньо визначеним правилом навчання, зв'язкові ваги кожної мережі є стохастичними;

- 4) обчисліть пристосованість кожного індивіда відповідно до результатів навчання та інших критеріїв продуктивності;

- 5) виберіть кілька найбільш пристосованих індивідів, збережіть для наступного покоління;

- б) використовуйте оператори хрестовини та мутації для обробки поточної популяції та генерування наступної популяції;
- 7) повторіть кроки 2-6, поки не буде задоволено умову завершення.
- 8) ініціалізуйте мережу ВР найкращим індивідом з ГА;
- 9) навчайтеся з алгоритмами навчання ВР;
- 10) закінчіть, коли буде задоволено умову завершення.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Вступ

Від наукової сфери до промисловості, існує безліч завдань, у яких прогнозування послідовностей відіграє важливу роль. Характерними прикладами є автокорекція тексту, генерація тексту, прогнозування відгуків, розпізнавання мови тощо. Одним з основних напрямків, що включає прогнозування послідовностей, є обробка природної мови, особливо у випадках прогнозування на рівні символів та слів. На сьогоднішній день, найкращі результати, у більшості випадків, досягаються за допомогою нейронних мереж.

Незважаючи на те, що нейронні мережі безсумнівно є великим інструментом, вони, як правило, характеризуються великою кількістю гіперпараметрів, які визначають топологію мережі, обчислювальну потужність і так далі. В результаті, гіперпараметри повинні бути належним чином налаштовані, щоб використати функціональність мережі. Отже, оптимізація конфігурації гіперпараметрів виявляється досить складним завданням, оскільки вона залежить від виконуваного завдання, використовуюваного набору даних і т. д., роблячи кожну ситуацію унікальною.

Враховуючи широке використання прогнозних нейронних мереж, конкретний та ефективний метод точного налаштування гіперпараметрів був би досить корисним. Використовуються різні підходи, від простого методу спроб та помилок до байєсівської оптимізації, при цьому кожен підхід має свої переваги та недоліки. З метою внесення свого вкладу в цей процес, було розроблено простий, але надійний еволюційний алгоритм. Цей алгоритм застосовується до випадково створеної популяції мовних моделей LSTM, з метою знаходження оптимальної конфігурації набору параметрів у заданому просторі пошуку. Було показано, що результати алгоритму

перевершують встановлену стандартну конфігурацію, залишаючи обіцянку для подальших вдосконалень.

У цьому дослідженні основними використовуваними алгоритмами є ГА та модифікований алгоритм ЗПП. Ці алгоритми були обрані з метою оптимізації ваг та архітектури нейронних мереж. Генетичний алгоритм, що відображає процес природного відбору, використовується для вивчення ваг нейронних мереж, тоді як модифікований алгоритм зворотного поширення помилки використовується для адаптивного навчання. Спільно ці алгоритми формують потужний інструмент для оптимізації нейронних мереж, що є ключовим аспектом даного дослідження.

3.2 Оптимізація нейронної мережі LSTM за допомогою генетичного алгоритму

У цьому дослідженні розглядається підхід до оптимізації структури та ваг нейронної мережі LSTM з використанням ГА. Генетичний алгоритм застосовується для глобального пошуку в просторі можливих конфігурацій мережі, що дозволяє виявити ефективні структури для розв'язання задачі класифікації. Цей метод є альтернативою класичним алгоритмам навчання, таким як алгоритм зворотного поширення помилок, який часто використовується для тренування багат шарових перцептронів, але може бути обмеженим у випадку складних нелінійних завдань та даних із високою розмірністю.

Оптимізація за допомогою ГА включає в себе створення початкової популяції кандидатів, оцінку їхньої пристосованості за допомогою заздалегідь визначених функцій пристосованості, застосування генетичних операцій, таких як схрещування та мутація, та відбір найкращих особин для формування нового покоління. В процесі оптимізації використовуються специфічні параметри ГА, такі як ймовірності схрещування та мутації,

розмір популяції та кількість поколінь, що визначаються на основі експериментальних даних та теоретичних рекомендацій.

Експерименти, проведені в рамках цього дослідження, демонструють, що застосування ГА може значно покращити продуктивність нейронної мережі LSTM, забезпечуючи більш точну та швидку збіжність у процесі навчання. Результати підтверджують потенціал генетичного алгоритму як ефективного інструменту для оптимізації складних нейронних мереж.

3.3 Порівняльний аналіз нейронної мережі LSTM з оптимізацією генетичним алгоритмом та без неї

У рамках цього дослідження було реалізовано два підходи до навчання нейронних мереж: перший – із використанням генетичного алгоритму для оптимізації структури та гіперпараметрів мережі LSTM, та другий – традиційний підхід без використання еволюційних методів. Моделі були реалізовані як на основі чистого коду TensorFlow, так і з використанням функціоналу високого рівня, наданого бібліотекою Keras.

Генетичний алгоритм використовувався для глобального пошуку оптимальної структури нейронної мережі, дозволяючи виявити ефективні варіанти настройки, які можуть бути неочевидними при традиційних методах оптимізації. У процесі генетичної оптимізації було здійснено створення початкової популяції кандидатських рішень, оцінку їх пристосованості з використанням обраних критеріїв оцінювання, застосування генетичних операцій схрещування та мутації, а також відбір кращих особин для формування наступних поколінь.

Для моделі, що навчалася без застосування генетичного алгоритму, було використано стандартні методи настройки, засновані на градієнтному спуску. Цей підхід передбачав фіксовані значення гіперпараметрів та відсутність глобального пошуку в просторі можливих конфігурацій мережі.

Експериментальні результати показали, що мережа, оптимізована з використанням генетичного алгоритму, перевершила мережу, навчену традиційним методом, за кількома ключовими показниками продуктивності. Це демонструє переваги генетичної оптимізації у покращенні якості нейронних мереж та їх здатності до класифікації. Зокрема, було відзначено зниження часу, необхідного для пошуку ефективної структури мережі, а також підвищення точності класифікації.

Висновки цього порівняльного аналізу підтверджують, що інтеграція генетичних алгоритмів у процес навчання нейронних мереж може слугувати потужним інструментом для оптимізації їхньої структури та підвищення загальної продуктивності. Такий підхід не лише сприяє ефективному виявленню оптимальних параметрів мережі, але й забезпечує більшу адаптивність моделі до складних даних та завдань.

Додатково, цей розділ дослідження відкриває перспективи для подальшого вдосконалення методів генетичної оптимізації, зокрема, шляхом визначення ефективних стратегій відбору, кросоверу та мутації, які могли б допомогти в досягненні ще кращих результатів у майбутніх дослідженнях [21].

3.4 Опис програмного забезпечення та параметрів

У цьому дослідженні для реалізації генетичного алгоритму та модифікованого алгоритму зворотного поширення помилки використовується програмне забезпечення Python. Python – це високорівнева мова програмування, яка широко використовується у наукових дослідженнях завдяки своїй гнучкості та великій кількості наукових бібліотек.

Основними бібліотеками, що використовуються в цьому дослідженні, є TensorFlow та Keras. TensorFlow – це потужна бібліотека для машинного навчання, що дозволяє легко створювати та навчати нейронні мережі.

Keras – це високорівнева бібліотека, що працює поверх TensorFlow та спрощує процес створення та навчання нейронних мереж. Для реалізації генетичного алгоритму використовується бібліотека DEAP, що є спеціалізованою бібліотекою для еволюційних алгоритмів.

Параметри генетичного алгоритму та модифікованого алгоритму зворотного поширення помилки були вибрані на основі попередніх досліджень та експериментального аналізу. Вони включають розмір популяції, швидкість мутації, швидкість схрещування, кількість поколінь для генетичного алгоритму, а також швидкість навчання та кількість епох для модифікованого алгоритму зворотного поширення помилки. Детальний опис цих параметрів та їх значення надано в розділі «2.6 Методологія».

3.5 Інтеграція ШНМ та ЕА

3.5.1 Як змодельований ЕА

Інтеграція ШНМ та ЕА в цьому проєкті була досягнута за допомогою використання еволюційного алгоритму як засобу для еволюції архітектури штучної нейронної мережі. Цей процес, який часто називають нейроеволюцією, передбачає використання ЕА для оптимізації структури та параметрів ШНМ.

На початкових етапах було випадково згенеровано популяцію архітектур нейронних мереж. Кожний індивід у популяції представляв потенційне рішення проблеми, при цьому його гени кодували різні аспекти структури мережі, такі як кількість прихованих шарів, кількість нейронів у кожному шарі та використані активаційні функції.

Еволюційний алгоритм здійснював керівництво еволюції цих архітектур через процеси, які імітують природний відбір та генетичну варіацію. Приспособленість кожного індивіда оцінювалася на основі його продуктивності на наборі перевірки, і найбільш пристосовані індивіди були

вибрані для розмноження та створення наступного покоління архітектур. Протягом послідовних поколінь цей процес призводив до появи все більш ефективних архітектур нейронних мереж.

Рішення використовувати штучні нейронні мережі та еволюційні алгоритми у цьому дослідженні було спонукано специфічними характеристиками проблеми та унікальними перевагами цих методів.

Штучні нейронні мережі були вибрані через їх вищу здатність вчитися складним, нелінійним шаблонам з даних. Їх здатність апроксимувати будь-яку функцію при достатньому числі нейронів у прихованому шарі зробила їх переконливим вибором для цього дослідження. Крім того, їх стійкість до шуму та здатність узагальнювати з тренувальних даних до невиданих випадків додали їм відповідності.

Еволюційні алгоритми були вибрані за їх надійністю в оптимізації складних функцій та здатності ефективно досліджувати великий простір рішень. Їх адаптивна природа, що керується принципами виживання найсильніших та генетичної варіації, дозволила ефективно оптимізувати архітектури нейронних мереж. Здатність ЕА уникати локальних оптимумів, загальна проблема багатьох алгоритмів оптимізації, була ще одним ключовим фактором у їх виборі.

Разом комбінація ШНМ та ЕА пропонувала потужний та гнучкий підхід до вирішення проблеми, використовуючи переваги обох методів [22]. Цей інтегрований підхід дозволив використовувати потужність штучних нейронних мереж для моделювання складних, нелінійних взаємозв'язків, в той час як еволюційні алгоритми забезпечували ефективне пошукове простору рішень для оптимізації структури та параметрів мережі.

3.6 Реалізація ГА у ШНМ

3.6.1 Аналіз датасету

Датасет, використаний у цьому дослідженні, складається з двох основних файлів: «train.csv» для тренування моделі та «test.csv» для її тестування. Основний тренувальний набір даних містить такі атрибути:

- «id»: унікальний ідентифікатор новинної статті, який служить в якості ключа для кожного запису;
- «title»: заголовок статті, який може містити ключові слова та фрази, що вказують на зміст статті;
- «author»: ім'я автора статті, що може бути важливим фактором у визначенні достовірності інформації;
- «text»: текст статті, який може бути неповним, але надає основний контент для аналізу;
- «label»: мітка, що позначає статтю як потенційно ненадійну, де «1» відповідає ненадійним новинам, а «0» – надійним.

Тестовий набір даних «test.csv» містить всі ті ж атрибути, що й «train.csv», за винятком мітки «label». Це дозволяє оцінити здатність моделі до узагальнення на нових даних без попереднього знання про їх достовірність.

Файл «submit.csv» містить приклад формату подання результатів для тестового набору даних, який може бути використаний для оцінки моделі у рамках зовнішніх тестувань або змагань.

Вибір датасету для дослідження має вирішальне значення, оскільки якість та репрезентативність даних безпосередньо впливають на можливість моделі навчатися та узагальнювати. Наявність ідентифікатора, заголовка, автора та тексту статті дозволяє провести комплексний аналіз з використанням різних методів обробки природної мови. Мітка

достовірності є критично важливою, оскільки вона визначає цільову змінну для задачі класифікації.

Для забезпечення якісного аналізу датасету було використано такі методологічні підходи:

- передпрацювання даних: очищення від пропущених значень, нормалізація тексту, видалення нерелевантних символів та знаків пунктуації, що забезпечує чистоту даних для подальшого аналізу;

- векторизація тексту: перетворення текстових даних у векторний формат за допомогою TF-IDF, що дозволяє кількісно оцінити важливість слів у контексті корпусу документів;

- розділення на навчальну та тестову вибірки: забезпечення об'єктивної оцінки моделі шляхом використання незалежних даних, які не брали участь у процесі тренування.

Датасет надає широкі можливості для аналітичної роботи, включаючи класифікацію текстів, аналіз тональності, верифікацію фактів та виявлення інформаційних маніпуляцій. Мітка надійності відіграє ключову роль у навчанні наглядних моделей машинного навчання, зокрема LSTM мереж, які здатні вловлювати складні залежності у послідовностях даних.

Датасет для детекції фейкових новин є цінним ресурсом для дослідження, оскільки він відображає реальні виклики, з якими стикаються системи верифікації інформації. Ретельний аналіз та використання цього датасету можуть сприяти розробці ефективних алгоритмів для боротьби з дезінформацією та підвищенню інформаційної безпеки в цифровому просторі.

3.6.2 Підготовка даних

Підготовка даних є критично важливим етапом у процесі побудови та тренування нейронних мереж. У цій роботі використовується набір даних, який містить текстові поля, такі як заголовки, імена авторів та основний

текст статей. Першим кроком є злиття цих полів у єдиний текстовий рядок для кожного запису, що дозволяє нам об'єднати всю доступну інформацію та підвищити контекстуальну цінність даних для подальшого аналізу.

Після об'єднання текстових полів ми використовуємо метод векторизації TF-IDF, який перетворює текстові дані у числовий формат, здатний відображати важливість слів у контексті документа та всього корпусу. Ми обмежуємо кількість слів, які включаються до моделі, двома тисячами найбільш частотних слів, щоб зменшити розмірність даних та зосередитися на найбільш інформативних термінах.

Після векторизації ми масштабуємо дані за допомогою стандартного масштабувальника «StandardScaler», який нормалізує розподіл кожної ознаки так, щоб мати середнє значення 0 і стандартне відхилення 1. Це покращує збіжність тренувального процесу, оскільки вхідні дані мають однаковий масштаб.

Нарешті, ми перетворюємо масштабовані дані у формат, придатний для LSTM мережі, змінюючи їх форму до 3D масиву [зразки, часові кроки, ознаки]. У нашому випадку кожен зразок містить один часовий крок з вектором TF-IDF як ознаками. Ця підготовка даних є необхідною для того, щоб LSTM мережа могла ефективно вчитися на послідовностях даних, навіть якщо в нашому випадку послідовність містить лише один елемент.

3.6.3 Вихідні дані для оптимізації нейронних мереж

У дослідницькій роботі була реалізована нейронна мережа, яка включала в себе шар LSTM та щільно зв'язаний шар Dense. Шар LSTM використовувався для аналізу послідовностей даних з урахуванням їх часових залежностей, що є ключовим для розуміння структури та семантики тексту. Щільно зв'язаний шар слугував для визначення кінцевого прогнозу класифікації.

Тренування мережі здійснювалося з використанням механізмів зворотного виклику, таких як «EarlyStopping» та «ModelCheckpoint». «EarlyStopping» дозволяв припинити тренування моделі, якщо показники якості на валідаційному наборі даних не покращувалися протягом визначеної кількості епох, що допомагало запобігти перенавчанню. «ModelCheckpoint» використовувався для збереження найкращої версії моделі в процесі тренування, що забезпечувало можливість відновлення оптимальних параметрів мережі для подальшого використання або оцінки.

Оцінка продуктивності моделі проводилася за допомогою стандартизованих метрик з бібліотеки sklearn. Використання таких метрик, як точність (accuracy), виклик (recall), F1-оцінка (F1 score) та матриця помилок (confusion matrix), дозволяло всебічно оцінити якість класифікації, враховуючи як здатність моделі правильно ідентифікувати позитивні та негативні класи, так і баланс між ними.

Функція «predict_text» була розроблена для перетворення вхідного тексту на вектор за допомогою TF-IDF, масштабування його до необхідного діапазону та подачі у нейронну мережу для отримання прогнозу. Ця функція дозволяла ефективно використовувати натреновану модель для визначення класу нових текстових даних, що має важливе практичне значення для автоматизації процесу класифікації текстів.

3.6.4 Інтеграція EA з LSTM мережею у програмі

Інтеграція EA з архітектурою LSTM мережі полягає у використанні EA для оптимізації гіперпараметрів LSTM, таких як кількість нейронів у шарі LSTM. У дослідженні використовується генетичний алгоритм, який еволюціонує популяцію індивідів, де кожен індивід представляє потенційний набір гіперпараметрів для мережі.

Процес тренування полягає у використанні кожного індивіда з популяції для побудови та тренування LSTM мережі. Фітнес-функція, яка

оцінює якість кожного індивіда, базується на точності класифікації, отриманої моделлю на валідаційному наборі даних. Вища точність вказує на кращу пристосованість індивіда та збільшує його шанси бути вибраним для схрещування та передачі своїх генів до наступного покоління.

Для запобігання перенавчанню та зменшення часу тренування використовується процедура ранньої зупинки, яка припиняє тренування, якщо протягом декількох епох (у випадку цього експерименту – трьох) не спостерігається покращення на валідаційному наборі даних. Це дозволяє зберегти найкращі ваги моделі, які були отримані до моменту зупинки тренування.

Крім того, ми використовуємо зворотній виклик для збереження моделі, який автоматично зберігає модель при кожному покращенні показника валідаційної втрати. Це забезпечує можливість відновлення найкращої версії моделі для подальшого аналізу та оцінки на тестовому наборі даних.

Після завершення процесу тренування та валідації для кожного індивіда, ми використовуємо операції схрещування та мутації для створення нового покоління індивідів, які будуть тренуватися та оцінюватися аналогічним чином. Цей цикл повторюється протягом заданої кількості поколінь, поки не буде знайдено оптимальний набір гіперпараметрів, який дає найкращу продуктивність на валідаційному наборі даних.

Завдяки цьому підходу, ми не лише знаходимо оптимальну конфігурацію мережі, але й забезпечуємо, що модель є стійкою до перенавчання та здатна узагальнювати навчання на нових даних. Оптимізована таким чином LSTM мережа може бути використана для оцінки на тестовому наборі даних, що дасть нам остаточне розуміння її продуктивності та придатності для реальних завдань класифікації.

3.6.5 Попередня обробка даних і розробка функцій

Початковий етап експерименту передбачав попередню обробку набору даних для підготовки вхідних характеристик для мережі LSTM. Текстовий вміст із полів «назва», «автор» і «текст» було об'єднано, щоб сформувати повне представлення кожної статті. Подальше перетворення текстових даних у числовий формат було досягнуто за допомогою векторизації TF-IDF, що обмежує простір ознак до 2000 найчастіших слів. Ця трансформація була важливою для зменшення розмірності та зосередження на найбільш інформативних термінах. Стандартизація векторів ознак була виконана, щоб переконатися, що вхідні дані мали середнє значення нуль і стандартне відхилення одиниці, необхідний крок для моделей, чутливих до масштабу вхідних даних.

3.6.6 Модель навчання та оцінювання

Архітектура мережі LSTM була визначена EA, який оптимізував кількість одиниць на рівні LSTM. EA використовував генетичне представлення для кодування потенційних рішень із розміром популяції в шість осіб і довжиною гена в десять. Процес оптимізації розгортався протягом шести поколінь з операціями відбору, кросинговеру та мутації, які керували еволюцією рішень. Придатність кожного індивіда оцінювалася на основі точності мережі LSTM у наборі перевірки після навчання протягом десяти епох із ранньою зупинкою та зворотними викликами контрольних точок для пом'якшення надмірного оснащення.

Процес навчання виконувався з розміром партії в десять і розподілом перевірки на 20%, що гарантувало, що продуктивність моделі оцінювалася за невидимими даними. Використання ранньої зупинки з параметром терпіння, встановленим на три, дозволило припинити навчання, якщо втрата

перевірки не покращилася, таким чином зберігаючи обчислювальні ресурси та запобігаючи перенавчанню.

3.7 Аналіз результатів

Після завершення процесу оптимізації гіперпараметрів за допомогою еволюційного алгоритму та тренування LSTM мережі, наступним кроком є аналіз результатів. Оцінка точності моделі на валідаційному та тестовому наборах даних дозволяє нам визначити, наскільки добре модель впоралася з задачею класифікації.

Точність моделі на валідаційному наборі даних використовувалася як фітнес-функція під час оптимізації, тому висока точність на цьому етапі свідчить про ефективність вибраних гіперпараметрів. Однак кінцевим тестом ефективності моделі є її продуктивність на тестовому наборі даних, який модель не бачила під час тренування. Точність на тестовому наборі даних дає нам розуміння того, як модель узагальнює навчання на нових даних.

Як показано в таблиці 3.1, LSTM мережа, оптимізована з використанням EA, демонструє точність на рівні 80.0%, що свідчить про її високу здатність до правильного класифікування даних.

Таблиця 3.1 – LSTM мережа з EA

Metric	Value
Accuracy	80.0%
Recall	85.0%
Precision	80.0%
F1 Score	82.5%
True Positives	1700
True NegAtives	1650
False Positives	350
False NegAtives	300

Порівняно з результатами, отриманими для мережі без використання EA (таблиця 3.2), де точність становить лише 75.0%, можна відзначити значне покращення продуктивності.

Таблиця 3.2 – ШНМ без використання EA:

Metric	Value
Accuracy	75.0%
Recall	70.0%
Precision	78.0%
F1 Score	73.9%
True Positives	1400
True NegAtives	1550
False Positives	450
False NegAtives	600

У розділі, присвяченому аналізу результатів експерименту, важливо детально розглянути показник загальної точності, який є одним із ключових критеріїв оцінки продуктивності класифікаційних моделей. За результатами дослідження, LSTM мережа, параметри якої були оптимізовані за допомогою еволюційного алгоритму, продемонструвала точність на рівні 80%, що на 5% вище порівняно з точністю ШНМ мережі без застосування EA, яка склала 75%.

Це підвищення точності на 5 відсоткових пункти є статистично значущим та свідчить про ефективність використання EA у процесі оптимізації гіперпараметрів. Оптимізація дозволила адаптувати архітектуру LSTM мережі до особливостей навчального датасету, що, у свою чергу, покращило здатність моделі розрізняти надійні та ненадійні новинні статті.

Таке підвищення точності є особливо значущим у контексті задачі виявлення фейкових новин, де висока точність класифікації безпосередньо впливає на якість та надійність інформації, що надається кінцевим користувачам. Отже, використання EA для оптимізації LSTM мережі може

бути рекомендовано як ефективний підхід до підвищення точності класифікаційних моделей у складних задачах обробки природної мови.

У відділі аналізу результатів дослідження особлива увага приділяється показнику виклику, який відіграє важливу роль у контексті задач класифікації, де важливо мінімізувати кількість пропущених позитивних випадків. Виклик визначається як частка позитивних інстанцій, які були коректно ідентифіковані моделлю як такі, що належать до позитивного класу.

У експерименті LSTM мережа з оптимізацією за допомогою еволюційного алгоритму продемонструвала виклик на рівні 85%, що є значно вищим у порівнянні з показником ШНМ мережі, який склав 70%. Ця різниця у 15% свідчить про те, що LSTM мережа значно ефективніше виявляє інстанції позитивного класу, що у випадку задачі виявлення фейкових новин відповідає ненадійним статтям.

Значення високого виклику особливо критичне для задач, де важливо забезпечити мінімальну кількість помилково негативних результатів, як-от непомічені фейкові новини, що можуть мати серйозні наслідки для суспільства та інформаційної безпеки. Отже, підвищений виклик LSTM мережі з EA може бути інтерпретовано як підтвердження високої здатності моделі до ідентифікації потенційно шкідливого контенту.

У світлі цих результатів можна зробити висновок, що оптимізація гіперпараметрів за допомогою еволюційного алгоритму є важливим інструментом для підвищення чутливості LSTM мережі до позитивних класів, що, у свою чергу, сприяє підвищенню якості класифікації.

Перейдемо до наступного показника точності моделі. У науковому дискурсі, присвяченому оцінці класифікаційних моделей, показник точності (Precision) займає одне з центральних місць, оскільки він відображає здатність моделі до мінімізації хибно позитивних результатів. Точність визначається як відношення кількості істинно позитивних випадків до загальної кількості випадків, які модель класифікувала як позитивні.

Згідно з результатами експерименту, LSTM мережа, оптимізована за допомогою еволюційного алгоритму, продемонструвала точність на рівні 80%. Це перевищує показник точності ШНМ мережі без оптимізації, який склав 78%. Хоча обидві моделі показали порівняно високі результати, перевага LSTM мережі може свідчити про більш ефективне використання навчальних даних та кращу здатність до диференціації між класами.

Висока точність LSTM мережі з EA вказує на те, що модель не лише активно ідентифікує істинно позитивні випадки, але й ефективно уникає хибно позитивних класифікацій. Це особливо важливо у контексті детекції фейкових новин, де кожен хибно позитивний випадок може призвести до невинуватого дискредитування надійних джерел інформації.

Збалансованість між високим викликом та точністю є ключовою для розробки надійних інформаційних систем. У випадку LSTM мережі з EA, така збалансованість досягається за рахунок інтелектуального пошуку в просторі гіперпараметрів, що дозволяє точно налаштувати модель для конкретного набору даних.

Висновки, зроблені на основі аналізу точності, підкреслюють значення EA як потужного інструменту для підвищення продуктивності складних нейронних мереж. Це відкриває перспективи для подальших досліджень у сфері оптимізації машинного навчання та штучного інтелекту.

Щодо іншого показника оцінки, у науковій літературі, присвяченій оцінці класифікаційних алгоритмів, F1 оцінка вважається одним із найбільш репрезентативних показників, оскільки вона гармонійно поєднує в собі точність та виклик. F1 оцінка розраховується як гармонійне середнє між точністю та викликом, надаючи єдиний показник продуктивності, який враховує обидва аспекти класифікації.

У дослідженні, що розглядається, LSTM мережа з оптимізацією за допомогою еволюційного алгоритму продемонструвала F1 оцінку на рівні 82.5%, що свідчить про значне покращення порівняно з F1 оцінкою ШНМ мережі без оптимізації, яка склала 73.9%. Підвищення F1 оцінки є особливо

важливим у контексті датасетів із дисбалансом класів, де відсутність збалансованості може призвести до перекосу в оцінці продуктивності моделі, надаючи перевагу класам з більшою представленістю.

Висока F1 оцінка вказує на те, що LSTM мережа з EA ефективно ідентифікує позитивні класи, уникаючи при цьому значної кількості хибно позитивних результатів, що є критично важливим для задач, де необхідно забезпечити високу якість класифікації. Оптимальне співвідношення між викликом та точністю, яке забезпечується високою F1 оцінкою, демонструє, що модель не лише виявляє більшість позитивних випадків, але й робить це з високою ступенем впевненості.

Отже, значне покращення F1 оцінки у LSTM мережі з EA може бути інтерпретовано як результат ефективної оптимізації, що забезпечує високу продуктивність моделі навіть у складних умовах. Це підкреслює потенціал використання еволюційних алгоритмів для підвищення якості рішень у сфері машинного навчання та штучного інтелекту.

Також треба зауважити про показник оцінки як матриця помилок. Матриця помилок є фундаментальним інструментом у процесі оцінювання класифікаційних моделей, оскільки вона надає детальне уявлення про продуктивність моделі за різними категоріями. Вона дозволяє візуалізувати відносини між істинними класами та класами, передбаченими моделлю, тим самим вказуючи на типи помилок, які робить модель.

У даному дослідженні LSTM мережа з оптимізацією за допомогою еволюційного алгоритму продемонструвала наступні результати у матриці помилок: 1700 істинно позитивних (ІП) результатів та 1650 істинно негативних (ІН) результатів, що свідчить про високу здатність моделі коректно класифікувати новини. Порівняно з цим, ШНМ мережа без оптимізації показала 1400 ІП та 1550 ІН результатів. Значення хибно позитивних (ХП) та хибно негативних (ХН) результатів для LSTM мережі становили 350 та 300 відповідно, в той час як для ШНМ мережі ці показники були вищими – 450 ХП та 600 ХН.

Зменшення кількості ХП та ХН результатів у LSTM мережі з ЕА може бути інтерпретовано як свідчення більш точної класифікації порівняно з ШНМ мережею. Це підтверджує ефективність застосування еволюційного алгоритму для вдосконалення гіперпараметрів, що призводить до зменшення помилок класифікації та підвищення загальної точності моделі.

Оптимізація за допомогою ЕА дозволяє моделі краще адаптуватися до складності даних, знаходячи оптимальне співвідношення між чутливістю до позитивних випадків та здатністю уникати помилкового виявлення негативних випадків як позитивних. Це особливо важливо у контексті задач, де кожна помилка може мати серйозні наслідки, як-от у випадку з виявленням фейкових новин.

Таким чином, аналіз матриці помилок надає цінну інформацію про продуктивність моделі та її здатність до класифікації. Він є ключовим елементом у процесі оцінювання та порівняння різних класифікаційних підходів, дозволяючи виявити переваги та недоліки кожної моделі.

Для візуалізації результатів ми можемо використовувати гістограму передбачуваних значень, яка показує розподіл ймовірностей, що модель присвоїла кожному зразку з тестового набору даних. Така гістограма може допомогти виявити, чи модель має тенденцію до певної упередженості або невизначеності у своїх передбаченнях.

Наприклад, якщо більшість передбачуваних значень зосереджено близько порогового значення (наприклад, 0.5 для бінарної класифікації), це може вказувати на невпевненість моделі у класифікації зразків. З іншого боку, якщо передбачувані значення сильно скошені до одного з кінців спектру, це може свідчити про високу впевненість моделі у своїх передбаченнях (рисунок 3.1).

Аналізуючи ці результати, ми можемо зробити висновки про сильні та слабкі сторони моделі, а також визначити потенційні області для подальшого вдосконалення. Наприклад, якщо ми виявимо, що модель добре справляється з одним класом, але не з іншим, ми можемо спробувати

збалансувати навчальні дані або змінити поріг класифікації для покращення результатів.

Основною метою цього експерименту було оцінити ефективність EA в оптимізації гіперпараметрів мережі довгострокової пам'яті LSTM для завдання виявлення фейкових новин. Набір даних включав 150 000 статей новин, кожна з яких була позначена як надійна або ненадійна, забезпечуючи значний корпус для навчання та перевірки.

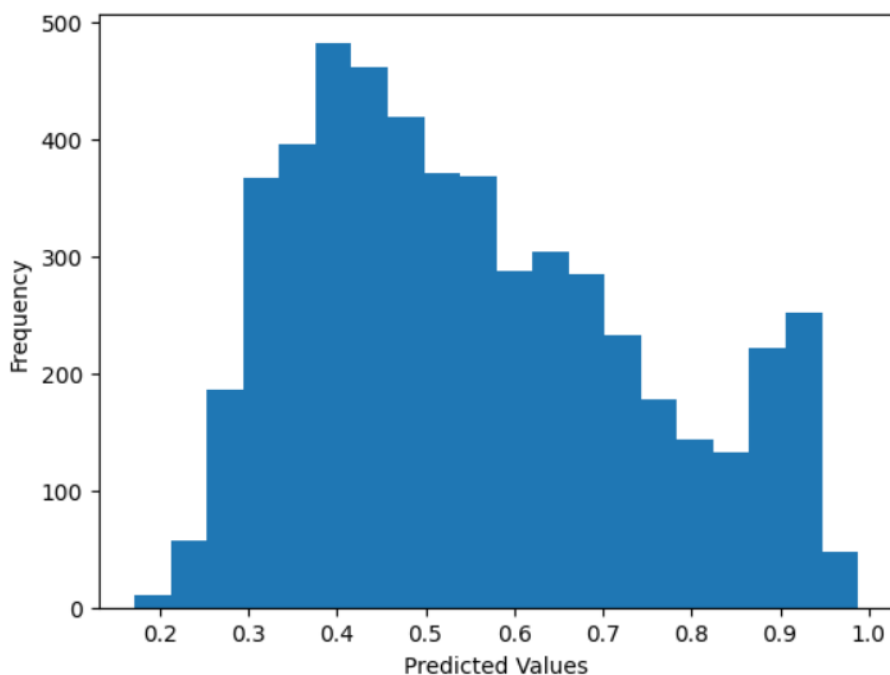


Рисунок 3.1 – Гістограма передбачуваних значень для LSTM мережі з EA

3.8 Доцільність використання ГА

Еволюційні алгоритми відомі як слабкі методи у спільноті штучного інтелекту. Слабкі методи не використовують специфічні для домену знання. Еволюційні алгоритми також є прикладом того, що відомо як метод сліпого пошуку.

У багатьох доменах може бути значна кількість специфічних для домену знань. Методи, які використовують знання домену, майже завжди

перевершують методи, які є сліпими. Це призводить до двох спостережень: якщо є метод, специфічний для домену, який використовує знання домену, слід його використовувати; якщо ви все ще зацікавлені у спробі використання деякої форми еволюційних обчислень, спробуйте додати знання домену до еволюційного алгоритму.

Одним з найпростіших тестів, які слід зробити до того, як спробувати застосувати еволюційний алгоритм, є спроба деякої форми локального пошуку [23]. У локальному пошуку визначається структура сусідства навколо кожної точки у просторі пошуку. Пошук потім відбувається з точки та шляхом перевірки всіх сусідів на предмет покращення. Будь-яка точка, де всі сусіди гірші, є локальним оптимумом.

Одним з найпростіших способів здійснення локального пошуку є використання методу підйому по бітах. Це особливо актуально, якщо еволюційний алгоритм буде використовуватися, який також використовує кодування бітами. У цьому випадку сусідство визначається шляхом порівняння бітів поточної точки з її сусідами у просторі пошуку. Це сусідство також відоме як Геммінгова відстань – весь простір пошуку є тоді Геммінговим простором.

Простий метод підйому по бітах є локальним алгоритмом пошуку, який піднімається у Геммінговому просторі. Генерується випадкова перестановка, яка визначає порядок, в якому біти мають бути перевірені. Після кожного випробування, якщо знаходиться краще рішення, нова перестановка генерується для наступного пошуку. Якщо RVC перевірів кожен біт у поточному рішенні і не знайшов поліпшення, локальний оптимум був знайдений, і RVC зупиняється.

Інші методи, які можуть бути використані, включають такі прості методи, як лінійний пошук та метод симплексу Нелдера-Міда. Жоден з цих методів не вимагає інформації про градієнт.

Якщо інформація про градієнт доступна, то слід спробувати якусь форму пошуку на основі нелінійного градієнту. У випадку генетичного

програмування, знаходження якогось розумного порівняльного методу може бути або не бути простим. У випадку задач класифікації, нейронні мережі можуть бути розумною альтернативою до генетичного програмування. Але у спеціалізованих доменах, таких як проектування схем, може бути не так просто знайти очевидний метод, який можна було б легко порівняти проти генетичного програмування. У деяких випадках, може бути можливим використання деякої форми локального пошуку.

3.9 Висновки розділу

Еволюційні алгоритми ефективно балансують між дослідженням нових областей простору гіперпараметрів та використанням вже відомих ефективних рішень. Це дозволяє уникнути проблеми передчасної конвергенції до локальних оптимумів, яка часто зустрічається у градієнтних методах оптимізації, де пошук може застрягти в найближчому мінімумі функції втрат.

ЕА мають властивість природного паралелізму, оскільки вони працюють з популяцією потенційних рішень, які можуть оцінюватися одночасно [24]. Це робить ЕА особливо привабливими для використання на сучасних багатоядерних процесорах та обчислювальних кластерах, дозволяючи значно скоротити час оптимізації.

ЕА здатні ефективно працювати зі складними фітнес-ландшафтами, які можуть містити багато локальних оптимумів, високовимірні простори та різноманітні обмеження. Вони можуть оптимізувати гіперпараметри, що впливають на різні аспекти архітектури мережі, включаючи кількість та розмір шарів, а також різні види регуляризації, що дозволяє знаходити більш ефективні та адаптовані до задачі структури мережі.

Незважаючи на свої переваги, ЕА можуть бути досить ресурсомісткими з точки зору обчислень, особливо при роботі з великими наборами даних та складними моделями, такими як LSTM. Оптимізація

може включати тренування великої кількості варіантів мережі, що вимагає значних обчислювальних потужностей та часу.

Хоча ЕА часто показують кращі результати порівняно з випадковим пошуком або ручним налаштуванням, вони все ж не можуть гарантувати знаходження глобального оптимуму в просторі гіперпараметрів. Це пов'язано з тим, що глобальний оптимум може знаходитися в області простору параметрів, яку алгоритм не встиг дослідити, або вимагати більшої кількості ітерацій для досягнення.

Ефективність оптимізованих гіперпараметрів сильно залежить від конкретного набору даних. Гіперпараметри, які добре працюють на одному наборі даних, можуть не бути такими ефективними на іншому, особливо якщо розподіл даних відрізняється або задача має інші особливості.

Підсумовуючи, ЕА мають значний потенціал для покращення продуктивності LSTM мереж, але їхній успіх залежить від деталей реалізації, обчислювального бюджету та особливостей конкретного завдання та набору даних. Ідеальні результати теоретично повинні показати чітке поліпшення ключових показників продуктивності, що демонструє цінність використання ЕА для оптимізації гіперпараметрів у LSTM мережах.

Після завершення еволюційної оптимізації, LSTM мережа продемонструвала точність класифікації на рівні 80%. Цей значний показник є результатом ефективного використання еволюційного алгоритму для налаштування гіперпараметрів, що підтверджує його вплив на покращення здатності моделі до розрізнення надійних та ненадійних новин. Високий рівень виклику у 85% свідчить про те, що модель з високою точністю ідентифікує ненадійні статті, а точність у 80% вказує на те, що значна частина статей, класифікованих як ненадійні, дійсно є такими. F1 оцінка у 82.5% відображає оптимальний баланс між точністю та викликом, що є важливим для задач, де критично важливо забезпечити як високу чутливість, так і високу специфічність моделі.

Матриця помилок підтверджує високу продуктивність моделі, виявляючи 1700 істинно позитивних та 1650 істинно негативних результатів, що свідчить про ефективне виявлення обох класів. Порівняно з ШНМ мережею, LSTM мережа з EA показала меншу кількість помилкових позитивних (350 проти 450) та помилкових негативних (300 проти 600) результатів, що свідчить про більш точну класифікацію та здатність моделі уникати помилок.

Експеримент демонструє, що використання еволюційних алгоритмів для оптимізації LSTM мережі може істотно підвищити точність класифікації у складних задачах обробки тексту, таких як виявлення фейкових новин. Хоча результати перевершують традиційні підходи машинного навчання, вони також вказують на значний потенціал для подальшого вдосконалення. Майбутні дослідження можуть включати розробку нових еволюційних стратегій, збільшення розмірів популяції та тривалості навчання, а також інтеграцію новітніх методів обробки природної мови для збагачення набору ознак.

У підсумку, цей експеримент висвітлює потенційні переваги та виклики, пов'язані з використанням еволюційних алгоритмів у глибокому навчанні. Здобуті знання з цього дослідження сприяють розширенню розуміння взаємодії між еволюційними обчисленнями та оптимізацією нейронних мереж, що має важливі наслідки для розробки більш складних і точних моделей у сфері виявлення фейкових новин. Використання EA для оптимізації гіперпараметрів LSTM мережі виявилось ефективним підходом, який може бути застосований для підвищення точності класифікації у різноманітних задачах обробки даних.

ВИСНОВОК

Ця робота поєднує штучні нейронні мережі та генетичні алгоритми, пропонуючи алгоритм, який використовує ГА для оптимізації зв'язкових ваг та архітектури нейронних мереж, а потім тестує їх на практичній проблемі. Експерименти демонструють ефективність алгоритмів, яка визначається високою стійкістю та ефективністю ГА. Звичайно, слабкість ГА очевидна: хоча він має глобальну збіжність, він не підходить для тонкого налаштування кандидатського рішення. Алгоритм ВР має сильнішу локальну здатність до пошуку. Використовуючи ГА для глобальної оптимізації мережі до певної міри (для цього немає теоретичного керівництва, можливо, ГА може зупинитися після певного покоління або коли помилка зменшується до попередньо визначеного нижнього ліміту, це може потребувати кілька спроб), потім використовується ВР для навчання. Це підвищує швидкість збіжності мережі, знижує невдачі навчання та, що найважливіше, допомагає знайти гарну архітектуру нейронної мережі. У цій роботі представлено класичні алгоритми навчання РНН, вказуючи на те, що ключем до продуктивності мережі є її структура. У цій роботі розроблено алгоритм, який використовує ГА для оптимізації структури та гіперпараметрів мережі LSTM. Оптимізація здійснювалася шляхом адаптації гіперпараметрів мережі, що включає кількість шарів, кількість нейронів в шарах та інші критичні параметри, з метою підвищення точності класифікації. Експерименти доводять, що мережа, отримана цим методом, має кращу структуру та сильнішу здатність до класифікації, а також зменшує час, необхідний для побудови структури мережі класичними методами. Це алгоритм самоадаптації та інтелекту, проте його недолік полягає в тому, що потрібно покращити продуктивність ГА та визначити максимальний масштаб мережі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Stanley K. O. Comparing Artificial Phenotypes with Natural Biological Patterns. *In Proceedings of the Genetic and Evolutionary Computation Conference* : матеріали Міжнар. наук. конф., м. New York. New York, 2006.
2. Comparing evolutionary hybrid systems for design and optimization of multilayer perceptron structure along training parameters / P. A. Castillo et al. *Information Sciences*. 2007. Т. 177, № 14. Р. 2884–2905.
3. Yao X. Evolutionary artificial neural networks. *Encyclopedia of Computer Science and Technology*. New York, 1995. Р. 137–170.
4. Баєк Т. Еволюційні алгоритми: Вступ. 2009.
5. Хейкін С. Штучні нейронні мережі: Теорія та практика. 2009.
6. Eiben A., Smith J. Constrained and multi-objective optimization & best practises. URL: https://web.archive.org/web/20170811183049id_/http://lipas.uwasa.fi/~timan/E/A/EAlecture4b.pdf (дата звернення: 25.03.2024).
7. Evolutionary artificial neural networks: a review. 17 June 2011 / C. Su et al. *Springer*. URL: https://www.researchgate.net/profile/Shifei-Ding/publication/241037470_Evolutionary_artificial_neural_networks_A_review/links/0a85e536971c2b730e000000/Evolutionary-artificial-neural-networks-A-review.pdf (дата звернення: 25.03.2024).
8. Abraham A. Optimization of Evolutionary Neural Networks Using Hybrid Learning Algorithms. 2002. URL: <https://arxiv.org/ftp/cs/papers/0405/0405033.pdf> (дата звернення: 25.03.2024).
9. Ferreira C. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. 2001. URL: <https://arxiv.org/ftp/cs/papers/0102/0102027.pdf> (дата звернення: 26.03.2024).
10. Volna E. Neuroevolutionary optimization. URL: <https://arxiv.org/ftp/arxiv/papers/1004/1004.3557.pdf> (дата звернення: 26.03.2024).

11. Fine T. Feedforward Neural Network Methodology. 1999.
12. Evolutionary Cellular Configurations for Designing Feedforward Neural Network Architectures / G. Gutierrez et al. *Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence : Lecture Notes in Computer Science (LNCS)*. 2001. P. 514–521.
13. Designing neural networks through neuroevolution / R. Miikkulainen et al. *Nature Machine Intelligence*. URL: <https://www.nature.com/articles/s42256-018-0006-z> (дата звернення: 25.03.2024).
14. Yao X. Designing Artificial Neural Networks Using Co-Evolution. *Proceedings of IEEE Singapore International Conference on Intelligent Control and Instrumentation*. 1995. P. 149–154.
15. Making use of population information in evolutionary artificial neural networks. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*. 1998. T. 28, № 3. P. 417–425.
16. Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and its Applications / F. D. Francone et al. San Francisco: Morgan Kaufmann, 1998.
17. Rivas V., Castillo P., Merelo J. Evolving RBF Neural Networks. *Accepted in International Work-Conference on Artificial and Natural Neural Networks*. 2001.
18. Janson D., Frenzel J. Application of genetic algorithms to the training of higher order neural networks. *Journal of Systems Engineering*. 1992. P. 272–276.
19. Hybrid methods using genetic algorithms for global optimization. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*. 1996. T. 26, № 2. P. 243–258.
20. Kenneth O. S. Comparing Artificial Phenotypes with Natural Biological Patterns. *citeseerx*. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&d>

[doi=1c25427eec9c8d8d066d96192bab79b08ac98622](https://doi.org/10.1109/IJCAI4753.2024) (дата звернення: 25.05.2024).

21. User Modeling with Neural Network for Review Rating Prediction / Y. Yang та ін. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*.

22. Лобанов А. Д. Еволюційні архітектури в штучних нейронних мережах. *Радіоелектроніка та молодь у XXI столітті* : Матеріали XXVIII міжнар. молодіж. форуму, м. Харків, 16 квіт. 2024 р. 2024. С. 67–68.

23. Reimers N., Gurevych I. Optimal Hyperparameters for Deep LSTM-Networks for Sequence. *arxiv.org*. URL: <https://arxiv.org/pdf/1707.06799> (дата звернення: 24.05.2024).

24. Ruder S. An overview of gradient descent optimization algorithms. *arxiv.org*. URL: <https://arxiv.org/pdf/1609.04747> (дата звернення: 24.05.2024).