

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
(КПІ ім. Ігоря Сікорського)

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

«До захисту допущено»

В.о. завідувач кафедри БМК

_____ Євген НАСТЕНКО

“ ____ ” _____ 2023р.

Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Комп'ютерні технології в біології та медицині»
спеціальності 122 «Комп'ютерні науки»

на тему: Алгоритм оптимізації параметрів згорткової нейронної
мережі для багатокласової класифікації електрокардіограм
з використанням генетичного алгоритму.

Виконав: студент IV курсу, групи БС-92

ЗИСКО АРТЕМ ОЛЕКСАНДРОВИЧ

(прізвище, ім'я, по батькові)


(підпис)

Керівник: Ст. вик. каф. біомедичної кібернетики (БМК)
Сердаковський Віталій Сергійович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент: доцент каф. біомедичної інженерії, доцент, к.т.н.
Зубчук Віктор Іванович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

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

Студент _____
(підпис)



Київ – 2023 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет біомедичної інженерії
Кафедра біомедичної кібернетики**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп’ютерні науки»

Освітньо-професійна програма «Комп’ютерні технології в біології та медицині»

ЗАТВЕРДЖУЮ

В.о. завідувач кафедри БМК

_____ Євген НАСТЕНКО

« 30 » травня 2023 р.

**ЗАВДАННЯ
на дипломну роботу студенту**

ЗИСКО АРТЕМ ОЛЕКСАНДРОВИЧ

(прізвище, ім’я, по батькові)

1. Тема роботи Алгоритм оптимізації параметрів
згорткової нейронної мережі для багатокласової класифікації
класифікації електрокардіограм з використанням генетичного алгоритму

Керівник роботи

Сердаковський Віталій Сергійович

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «31 » травня 2023 р. №2106-с

2. Термін подання студентом роботи **06-08 червня 2023р.**

3. Вихідні дані до роботи

База даних сигналів електрокардіограм, мова програмування Python, бібліотеки numpy, scipy, sklearn, PyTorch

4. Зміст роботи

Анотації (на двох мовах); Вступ ;Аналітичний огляд літературних джерел (огляд літературних джерел за темою ДР, обґрунтовано вибір архітектури нейронної мережі); Теоретична частина (обґрунтування архітектури нейронної мережі і методу пошуку оптимальних параметрів архітектури, формули, графіки); Практична реалізація задачі за темою ДР (обґрунтовано доцільність використання моделі); Розрахунок економічного ефекту за темою ДР; Загальні висновки; Список використаних джерел.

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

6. Консультанти розділів роботи

Розділ Дипломної роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання **30 травня 2023 року.**

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримати завдання за темою ДР на практику	До 15.02.2023р.	виконано
2	Переддипломна практика	За графіком	виконано
3	Виконання розділів ДР (Вступ, аналітичний огляд літературних джерел, теоретична частина)	До кінця практики	виконано
4	Виконання розділів ДР (практична частина, загальні висновки, список джерел)	Не пізніше 1 тижня до засідання каф-ри	виконано
5	Перевірка ДР науковим керівником	Не пізніше 1 тижня до засідання каф-ри	виконано
6	Подання в електронному вигляді ДР та анотації до неї на перевірку нормоконтролера та плагіат (UNICHECK).	---- « -----	виконано
7	Надання документів на засідання кафедри	За день до засідання	виконано
8	Предзахист ДР та допуск до захисту дисертації	Згідно плану каф.	виконано
9	Подання ДР рецензенту. Отримання рецензії.	До подання пакету документів до ЕК	виконано
10	Подання пакету документів по ДР та супровідних до неї документів до захисту в ЕК ¹	За 5 днів до дати захисту ДР за графіком	виконано
11	Захист ДР в ЕК		

Студент


(підпис)

Артем ЗИСКО

(ім'я, ПРІЗВИЩЕ)

Керівник ДР

(підпис)

Віталій СЕРДАКОВСЬКИЙ

(ім'я, ПРІЗВИЩЕ)

Нормоконтролер

(підпис)

Галина КОРНІЄНКО

(ім'я, ПРІЗВИЩЕ)

¹ не пізніше ніж за 5 днів до затвердженої дати захисту ДР в ЕК

АНОТАЦІЯ

Дипломна робота за темою «Алгоритм оптимізації параметрів згорткової неронної мережі для багатокласової класифікації електрокардіограм за допомогою генетичного алгоритму» виконана студентом кафедри біомедичної кібернетики ФБМІ Зиско Артемом зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині» та складається зі: вступу; 4 розділів (аналітичний огляд літературних джерел, теоретична частина, практична частина, економічний розділ), висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 13 джерел та додатків. Загальний обсяг роботи 76 сторінок.

Актуальність теми.

Тема мого диплому є дуже актуальною і відноситься до застосування генетичних алгоритмів у класифікації електрокардіографічних (ЕКГ) сигналів та підборі параметрів нейромережі. Класифікація ЕКГ є важливим завданням в медицині, оскільки дозволяє виявляти різноманітні серцеві захворювання та робити прогнози. Використання генетичних алгоритмів дозволяє ефективно підібрати оптимальні параметри нейромережі, що забезпечує досягнення кращих результатів класифікації. Такий підхід має потенціал вдосконалити діагностичні можливості в області серцево-судинних захворювань та покращати якість медичних рішень на основі ЕКГ аналізу.

Мета і завдання роботи.

Метою роботи є розробка і застосування генетичного алгоритму для підбору оптимальних параметрів нейромережі в задачі класифікації електрокардіографічних (ЕКГ) сигналів. Це дозволить покращити точність та ефективність класифікації ЕКГ, що в свою чергу має велике значення для діагностики серцевих захворювань та прийняття медичних рішень. Використання генетичного алгоритму дозволить автоматично знаходити

оптимальні налаштування мережі, зменшуючи залежність від експертного підбору параметрів та покращуючи здатність моделі узагальнювати.

Її досягнення передбачає вирішення наступних *завдань*:

Аналіз вітчизняних та зарубіжних джерел: У рамках дослідження необхідно провести широкий аналіз актуальних джерел вітчизняних та зарубіжних дослідників, що стосуються класифікації ЕКГ сигналів та застосування генетичного алгоритму для підбору параметрів моделі. Аналіз буде спрямований на вивчення різних підходів, архітектур нейромереж, методів підбору параметрів та досягнень у цій області.

Розробка алгоритму генетичного алгоритму для підбору оптимальних параметрів нейромережі в задачі класифікації ЕКГ сигналів: На основі отриманих з аналізу джерел та враховуючи специфіку задачі класифікації ЕКГ сигналів, планується розробити алгоритм генетичного алгоритму, який забезпечить ефективний підбір параметрів нейромережі. Алгоритм буде враховувати такі фактори, як архітектура моделі, гіперпараметри, функція втрат та метрики продуктивності, забезпечуючи оптимальну точність класифікації ЕКГ сигналів.

Використані методи

1. Збір та попередня обробка даних
2. Аналіз літературних джерел
3. Розробка нейромережі
4. Застосування генетичного алгоритму
5. Оцінка якості натренованих моделей

Отримані результати.

1. Досягнуто високої точності класифікації ЕКГ сигналів за допомогою розробленої нейромережі з використанням генетичного алгоритму для підбору параметрів.

2. Встановлено, що застосування генетичного алгоритму сприяє покращенню продуктивності нейромережі шляхом оптимального підбору параметрів.

3. Проведено порівняльний аналіз результатів з існуючими підходами та доведено ефективність запропонованого методу.

Ключові слова.

ЕКГ сигнали, класифікація ЕКГ, нейромережа, генетичний алгоритм, підбір параметрів, аналіз даних, обробка сигналів, машинне навчання, медична діагностика, архітектура нейромережі, оптимізація, ефективність, порівняльний аналіз, алгоритмічний підхід, прогностична модель.

Бібліографічний опис

Зиско А.О. Алгоритм оптимізації параметрів згорткової нейронної мережі для багатокласової класифікації електрокардіограм з використанням генетичного алгоритму : дипломна роб. бакалавра : 122 Комп'ютері науки / Зиско Артем Олександрович. – Київ, 2023. – 76 с.

ABSTRACT

The thesis on the topic "An algorithm for optimizing convolutional neural network parameters for multi-class classification of electrocardiograms using genetic algorithm" was conducted by Artem Zysko, a student of the Department of Biomedical Cybernetics at the Faculty of Computer Science and Cybernetics. The thesis is completed as part of the educational and professional program "Computer Technologies in Biology and Medicine" with specialization 122 "Computer Science." It consists of an introduction, four chapters (literature review, theoretical part, practical part, economic section), conclusions for each chapter, general conclusions, a list of references (including 13 sources), and appendices. The total length of the thesis is 76 pages.

Relevance of the topic: The topic of my thesis is highly relevant and pertains to the application of genetic algorithms in the classification of electrocardiographic (ECG) signals and parameter selection for neural networks. ECG classification is an important task in medicine as it enables the detection of various heart conditions and provides prognostic information. The use of genetic algorithms allows for the efficient selection of optimal parameters for the neural network, leading to improved classification results. This approach has the potential to enhance diagnostic capabilities in the field of cardiovascular diseases and improve the quality of medical decisions based on ECG analysis.

Objective and tasks: The objective of this thesis is to develop and apply a genetic algorithm for parameter selection of a neural network in the task of ECG signal classification. This will improve the accuracy and efficiency of ECG classification, which is crucial for the diagnosis of heart conditions and medical decision-making. The use of a genetic algorithm will enable the automatic discovery of optimal network configurations, reducing dependence on expert parameter tuning and improving the generalization ability of the model.

Methods used:

1. Data collection and preprocessing.
2. Literature review analysis.

3. Neural network development.
4. Application of genetic algorithm.
5. Evaluation of trained model performance.

Results obtained:

1. High accuracy in the classification of ECG signals was achieved using the developed neural network with the genetic algorithm for parameter selection.
2. The application of the genetic algorithm contributed to improved performance of the neural network by facilitating optimal parameter selection.
3. A comparative analysis of the results with existing approaches was conducted, demonstrating the effectiveness of the proposed method.

Keywords: ECG signals, ECG classification, neural network, genetic algorithm, parameter selection, data analysis, signal processing, machine learning, medical diagnostics, neural network architecture, optimization, efficiency, comparative analysis, algorithmic approach, predictive model.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	12
ВСТУП.....	13
РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ	17
Висновки до розділу 1.....	20
РОЗДІЛ 2 ТЕОРЕТИЧНА ЧАСТИНА	21
2.1 Задача Класифікації сигналів ЕКГ	21
2.2 Етап попередньої обробки даних	22
2.2.1 Нормалізація	22
2.2.2 Балансування класів	23
2.3 Шари згорткової нейронної мережі.....	24
2.3.1 Одновимірний оператор згортки	24
2.3.2 Макс-пулінг (Max Pool).....	25
2.3.3 Адаптивний середній пулінг(adaptive average pooling)	25
2.3.4 Батч-нормалізація.....	27
2.3.5 Функція активації GELU.....	28
2.4 Генетичний алгоритм	29
2.4.1 Популяція.....	29
2.4.2 Відбір.....	30
2.4.3 Схрещування.....	31
2.4.4 Мутація	32
2.4.5 Оцінювання.....	33
2.4.6 Елітарний відбір	34
2.4.7 Повторення	35

	10
2.4.8 Збір результатів	36
Висновки до розділу 2.....	37
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ	39
3.1 Основні використані бібліотеки	39
3.1.1 Numpy	39
3.1.2 PyTorch.....	40
3.1.3 Sci-kit learn	41
3.2 Попередня обробка даних	42
3.2.1 Нормалізація сигналів ЕКГ	42
3.2.2 Балансування даних	43
3.3 Архітектура мережі	44
3.3.1 Згорткові нейронні мережі.....	44
3.3.2 Блок згортки	45
3.3.3 Функції між блоками згортки	47
3.3.3 Шар класифікації.....	48
3.3.4 Структура всієї мережі	49
3.4 Процес навчання моделі.....	50
3.4.1 Функція помилки CrossEntropy	50
3.4.2 Оптимізатор Adam.....	51
3.4.3 Градієнтний спуск	52
3.4.4 Процес зменшення кроку навчання.....	52
3.5 Реалізація генетичного алгоритму.....	54
3.5.1 Загальні відомості.....	54
3.5.2 Ініціалізація популяції.....	55
3.5.3 Обчислення функції пристосованості	55

	11
3.5.4 Відбір.....	57
3.5.5 Елітні розв’язки	58
3.5.6 Кросинговер.....	59
3.5.7 Мутація	59
3.5.8 Аналіз отриманих результатів генетичного алгоритму	60
3.6 Розрахунок економічного ефекту за темою дипломної роботи.....	68
Висновки до розділу 3.....	69
ЗАГАЛЬНІ ВИСНОВКИ.....	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	74

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ГА – генетичний алгоритм

ЕКГ – електрокардіограма

ЗНМ – згорткова нейронна мережа

CNN – convolutional neural network

ECG – electrocardiogram

ROC – receiver operating characteristics

AUC – area under curve

SGD – stochastic gradient descent

GD – gradient descent

GELU – Gaussian error linear unit

BERT – bidirectional encoder representations from transformer

GPT – generative pre-trained transformer

ВСТУП

Актуальність поставленої задачі в роботі та методи їх досягнення.

Дослідження та класифікація ЕКГ-сигналів мають велике значення для діагностики та моніторингу серцево-судинної системи. Акуратна та ефективна обробка даних є передумовою для отримання точних результатів класифікації та виявлення патологічних змін у серцевих сигналах.

У цій роботі було поставлено перед собою важливу мету - вдосконалення класифікації ЕКГ-сигналів. З метою досягнення цієї мети було використано генетичний алгоритм, який виявився потужним інструментом для автоматичного підбору оптимальних параметрів архітектури мережі.

Генетичний алгоритм базується на еволюційних принципах, що дозволяють знаходити оптимальні комбінації параметрів шляхом застосування селекції, схрещування та мутації. Цей підхід дозволяє ефективно вирішувати оптимізаційні задачі, зокрема у випадку класифікації ЕКГ-сигналів.

Використання генетичного алгоритму у даній роботі дало значні результати. Він забезпечив автоматичний підбір оптимальних параметрів архітектури мережі, що в свою чергу призвело до покращення точності та ефективності класифікації ЕКГ-сигналів. Отримані результати підтверджують високий потенціал генетичного алгоритму в розв'язанні задач оптимізації для класифікації ЕКГ-сигналів.

Дослідження генетичного алгоритму в контексті класифікації ЕКГ-сигналів має велике значення для медичної галузі. Він допомагає автоматизувати та покращити процес виявлення серцевих захворювань, що є критичним для ранньої діагностики та лікування. Застосування генетичного алгоритму в класифікації ЕКГ-сигналів відкриває нові можливості для розвитку медичної діагностики та допомагає покращити якість догляду за хворими серцево-судинними захворюваннями.

Основними методами досягнення мети роботи є:

Аналіз вітчизняних та зарубіжних джерел, що дозволяє ознайомитися зі станом сучасних досліджень у галузі класифікації ЕКГ-сигналів та генетичних алгоритмів.

Розробка та реалізація генетичного алгоритму для автоматичного підбору оптимальних параметрів архітектури мережі.

Використання відповідних методів обробки даних, таких як фільтрація шуму, нормалізація амплітуд та видалення артефактів, для покращення якості сигналу та отримання точніших результатів класифікації.

Ці методи дозволяють досягнути покращення ефективності класифікації ЕКГ-сигналів та отримання більш точних результатів, що можуть бути корисні для подальшого аналізу та діагностики серцевих захворювань.

Мета і завдання роботи

Метою даної роботи є не лише дослідження, а й розробка нового та інноваційного алгоритму для автоматичної класифікації ЕКГ-сигналів з метою ефективного та точного виявлення серцевих захворювань. Ця задача є надзвичайно актуальною і має велике значення у сфері медицини.

Серцеві захворювання є однією з провідних причин смертності у світі, а також мають серйозний вплив на якість життя пацієнтів. Рання діагностика та вчасне виявлення цих захворювань є критично важливими для успішного лікування та попередження можливих ускладнень.

У зв'язку з цим, розробка нового алгоритму для автоматичної класифікації ЕКГ-сигналів має великий потенціал у покращенні діагностичних можливостей та ефективності медичних процедур. Застосування такого алгоритму дозволить швидше та точніше виявляти відхилення в ЕКГ-сигналах, що в свою чергу сприятиме ранньому виявленню серцевих захворювань та призведе до покращення результатів лікування.

Більш того, інтеграція автоматичної класифікації ЕКГ-сигналів з розробленим алгоритмом може сприяти зниженню навантаження на медичний персонал, прискорити процес діагностики та забезпечити більш ефективне використання ресурсів у медичних установах.

Її досягнення передбачає вирішення наступних завдань:

1. Аналіз вітчизняних та зарубіжних джерел для отримання вичерпної інформації про існуючі методи класифікації ЕКГ-сигналів та їх обмеження. В рамках цього завдання проведено огляд літератури, переглянуто наукові публікації, проведено аналіз методів машинного навчання та штучних нейронних мереж, що застосовуються для класифікації ЕКГ-сигналів. Також було вивчено проблеми, пов'язані з недостатньою точністю класифікації та складністю обробки ЕКГ-сигналів.
2. Розробка та реалізація нового алгоритму, який поєднує в собі згорткові нейронні мережі та генетичні алгоритми для покращення точності класифікації ЕКГ-сигналів. В рамках цього завдання проведено детальний аналіз та порівняння різних архітектур згорткових нейронних мереж, вивчено принципи роботи генетичних алгоритмів та їх застосування в задачах оптимізації параметрів моделей. На основі цього аналізу було розроблено новий алгоритм, який використовує згорткові шари для екстракції важливих ознак з ЕКГ-сигналів та генетичні алгоритми для автоматичного підбору оптимальних параметрів моделі. Такий підхід сприяє покращенню точності класифікації та забезпечує більш точні результати.

Використані методи

Перегляд та аналіз літератури, наукових публікацій та досліджень у сфері класифікації ЕКГ-сигналів та генетичних алгоритмів, що дозволило оцінити сучасний стан досліджень та виявити невирішені проблеми.

Розробка та реалізація згорткових нейронних мереж для обробки та аналізу ЕКГ-сигналів. Було використано популярні архітектури, такі як Convolutional Neural Networks (CNN), а також проведено експерименти з різними конфігураціями мережі для досягнення оптимальної точності класифікації.

Використання генетичного алгоритму для оптимізації параметрів моделі та покращення точності класифікації. Було реалізовано генетичний алгоритм, який здійснює селекцію, схрещування та мутацію параметрів моделі з метою підбору

оптимальних комбінацій та досягнення кращої пристосованості моделі до задачі класифікації ЕКГ-сигналів.

Використання методів обробки сигналів, таких як фільтрація шуму, нормалізація амплітуд та видалення артефактів, для покращення якості вхідних даних та забезпечення більш точних результатів класифікації.

Отримані результати

Підтвердженням високої точності та ефективності запропонованого алгоритму є результати отримані в процесі його застосування. Застосування розробленого методу дозволило досягти точності класифікації ЕКГ-сигналів на рівні 95%. Це вражаючий показник, що свідчить про високий потенціал алгоритму у виявленні серцевих захворювань та покращенні діагностичних можливостей.

Застосування розробленого методу показало високу надійність та стабільність результатів. Алгоритм продемонстрував високу чутливість та специфічність, здатність виявляти навіть найменші аномалії в ЕКГ-сигналах, що робить його потужним інструментом у діагностиці серцевих захворювань.

Додатково, проведено порівняльний аналіз з існуючими методами класифікації ЕКГ-сигналів. Розроблений алгоритм продемонстрував кращі результати, перевершуючи інші підходи за точністю та швидкістю обробки. Його ефективність і переваги підтверджують його потенціал для використання у клінічній практиці та медичних дослідженнях.

Отримані результати свідчать про успішне досягнення мети роботи - розробку нового алгоритму для автоматичної класифікації ЕКГ-сигналів з метою виявлення серцевих захворювань. Його застосування в медичній практиці може значно поліпшити процес діагностики та надати раннє виявлення серцевих захворювань, що сприятиме вчасному призначенню лікування та покращенню прогнозу для пацієнтів.

РОЗДІЛ 1

АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ

В дослідженні класифікації ЕКГ-сигналів було звернуто увагу на декілька важливих джерел, які принесли значний внесок у розуміння та розвиток цієї області.

Одним з ключових джерел є книга "PyTorch Recipes" [1] від автора Mishra P., де пропонується низка рецептів та практичних прикладів використання бібліотеки PyTorch для розробки алгоритмів. Ця книга є цінним ресурсом для розуміння основних концепцій та технік роботи з PyTorch, що дозволяє ефективно застосовувати цей фреймворк для обробки та аналізу ЕКГ-сигналів.

Додатково, у статті "ECG heartbeat classification using convolutional neural networks" [2], було описано успішне використання згорткових нейромереж для класифікації ЕКГ-сигналів. Це дослідження підтверджує, що нейромережевий підхід може забезпечити високу точність класифікації серцевих ритмів, що має велике значення для діагностики та моніторингу серцевих захворювань.

При розробці алгоритму класифікації, було використано підхід з оптимізації мережі за допомогою генетичного алгоритму, який базується на еволюційних принципах. Відповідну методологію можна знайти у книзі "Genetic Algorithms" від Vanneschi L. та Silva S. [3]. Застосування генетичних алгоритмів дозволяє ефективно підібрати оптимальні параметри архітектури мережі для досягнення кращих результатів класифікації ЕКГ-сигналів.

Зазначимо також статтю "Convolutional Neural Network for Heartbeat Classification" [4], де авторами Fang H., Lu C., Hong F., Jiang W. та Wang T. розглядається використання згорткових нейромереж для класифікації серцевих ритмів. Це дослідження пропонує нові інсайти та підходи до розв'язання задачі класифікації ЕКГ-сигналів, що можуть бути корисними для отримання додаткової інформації та удосконалення методів аналізу серцевої активності.

У роботі також було звернуто увагу на публікацію "Understanding AUC-ROC Curve" [5] від Narkhede S. Ця стаття розкриває деталі та особливості використання

характеристики AUC-ROC (площа під кривою ROC) для оцінки якості класифікатора. Розуміння цієї характеристики є важливим кроком при оцінці та порівнянні різних моделей класифікації ЕКГ-сигналів та визначенні їх ефективності.

У проведенні дослідження щодо класифікації ЕКГ-сигналів було враховано кілька значущих джерел, що принесли вагомий внесок у розвиток цієї області.

Одним із ключових джерел є стаття "Deep Learning for ECG Analysis: Benchmarks and Insights" [6]. У цій роботі досліджено використання глибокого навчання для аналізу ЕКГ-сигналів та представлено низку базових моделей та методик, які можуть бути використані для класифікації серцевих ритмів. Результати цього дослідження надають цінну інформацію про ефективність та порівняльний аналіз різних підходів у використанні глибокого навчання для ЕКГ-сигналів.

Додатково, у роботі "An Overview of Electrocardiogram Classification using Machine Learning Techniques" [7], досліджено використання методів машинного навчання для класифікації ЕКГ-сигналів. У цій статті розглянуто різні методи класифікації, включаючи методи на основі дерев рішень, навчання з підкріпленням та методи на основі нейронних мереж. Дослідження показує переваги та обмеження кожного методу та надає цінні рекомендації для вибору оптимального підходу до класифікації ЕКГ-сигналів.

Також варто відзначити статтю "A Review on Convolutional Neural Networks for ECG Signal Processing" [8], автором якої є Rajpurkar. У цьому оглядовому дослідженні проаналізовано використання згорткових нейромереж для обробки ЕКГ-сигналів. Автори розглядають різні архітектури та методи препроцесингу даних, що допомагають досягти високої точності класифікації серцевих ритмів. Ця робота є важливим джерелом для отримання огляду сучасних підходів до використання згорткових нейромереж у сфері ЕКГ-сигналів.

Одним з таких джерел є стаття "A Comparative Study of Heartbeat Classification Algorithms on the MIT-BIH Arrhythmia Database" [9]. У цій роботі автори провели порівняльний аналіз різних алгоритмів класифікації серцевих ритмів на основі бази

даних MIT-BIH Arrhythmia. В результаті дослідження були отримані цінні висновки щодо ефективності різних методів та їх застосування для класифікації ЕКГ-сигналів.

Додатково, у статті "Feature Extraction and Classification of ECG Signals Using Wavelet Transform and Machine Learning Techniques" [10], авторами Verma et al., розглядається використання хвильового перетворення та методів машинного навчання для екстракції ознак та класифікації ЕКГ-сигналів. У цьому дослідженні автори досліджують ефективність використання хвильового перетворення як методу екстракції ознак та порівнюють його з іншими підходами. Результати роботи надають цінні вказівки щодо використання хвильового перетворення та машинного навчання для аналізу ЕКГ-сигналів.

Зазначимо також роботу "ECG Signal Denoising and Baseline Wander Correction Based on CEEMDAN and EMD" [11]. В цій статті автори пропонують новий підхід до очищення ЕКГ-сигналів від шуму та коригування базової блукання. Використовуючи методи комплексного емпіричного розкладання модулюючих функцій (CEEMDAN) та емпіричного розкладання модулюючих функцій (EMD), вони досягають значного покращення якості сигналу. Ця робота може бути корисною при розробці алгоритмів для покращення якості ЕКГ-сигналів перед їх класифікацією.

В подальшому дослідженні також було звернуто увагу на статтю "Automatic Classification of Electrocardiogram (ECG) Signals Using Morphological and Discrete Wavelet Transform (DWT) Features" [12]. У цій роботі автори пропонують новий алгоритм для автоматичної класифікації ЕКГ-сигналів на основі морфологічних ознак та дискретного хвильового перетворення. Вони виконують екстракцію ознак з сигналу, побудову класифікатора та оцінюють його ефективність на реальних даних. Результати дослідження показують високу точність та надійність запропонованого методу.

Висновки до розділу 1

В данному розділі було проведено огляд літератури щодо класифікації ЕКГ-сигналів, і в ньому було зазначено декілька важливих джерел, що внесли значний внесок у розуміння та розвиток цієї області. Огляд літератури надав цінну інформацію про різні методи та підходи до класифікації ЕКГ-сигналів, а також вказав на переваги та обмеження кожного з них.

У процесі дослідження були звернуті увагу на декілька ключових джерел, які становлять основу для подальшого розвитку та використання методів класифікації ЕКГ-сигналів. Зокрема, було згадано книгу "PyTorch Recipes" автора Mishra P., яка надає цінний ресурс для розуміння та використання бібліотеки PyTorch у контексті обробки та аналізу ЕКГ-сигналів.

Також були згадані статті, які досліджують використання згорткових нейромереж для класифікації ЕКГ-сигналів, зокрема статті авторів Xu X. та Liu H., Fang H., Lu C., Hong F., Jiang W., Wang T. Ці дослідження підтверджують ефективність нейромережевого підходу та пропонують нові інсайти та підходи до розв'язання задачі класифікації серцевих ритмів.

Огляд літератури також висвітлив важливість використання генетичних алгоритмів для оптимізації архітектури мережі у класифікації ЕКГ-сигналів. Оптимізація мережі за допомогою генетичних алгоритмів дозволяє підібрати оптимальні параметри та досягти кращих результатів класифікації.

Враховуючи висновки з огляду літератури, можна стверджувати, що класифікація ЕКГ-сигналів є активною та перспективною галуззю досліджень, яка використовує різні методи та алгоритми для досягнення високої точності та надійності в розпізнаванні серцевих ритмів.

РОЗДІЛ 2

ТЕОРЕТИЧНА ЧАСТИНА

2.1 Задача Класифікації сигналів ЕКГ

Аналіз ЕКГ-сигналів є важливим завданням у медичній діагностиці та моніторингу серцево-судинної системи. Визначення типу аритмії на основі ЕКГ-сигналів дозволяє лікарям здійснювати швидку та точну діагностику порушень серцевого ритму та приймати необхідні заходи для лікування та попередження серцевих захворювань.

ЕКГ-сигнал складається з різних компонентів, які відображають різні етапи серцевого циклу. Наприклад, відрізок сигналу, що відповідає одному серцевому скороченню, називається "QRS комплексом". Інші компоненти, такі як "Р-хвиля" та "Т-хвиля", відображають різні фази серцевого циклу. Кожна аритмія має свої унікальні особливості в ЕКГ-сигналі, які можуть бути використані для її класифікації.

Одним із ключових аспектів класифікації ЕКГ-сигналів є використання методів машинного навчання, зокрема нейромереж. Сучасні згорткові нейромережі (Convolutional Neural Networks - CNN) виявилися дуже ефективними в розпізнаванні паттернів у ЕКГ-сигналах та класифікації аритмій. Вони можуть автоматично виконувати екстракцію ознак із сигналів та навчатися розпізнавати шаблони, що характеризують різні типи аритмій.

Однак, перед застосуванням нейромереж для класифікації ЕКГ-сигналів, важливо підготувати дані. Це включає в себе очищення сигналу від шуму, вирівнювання базової лінії, нормалізацію амплітуд та інші операції обробки сигналу. Крім того, розмітка (анотація) ЕКГ-сигналів з правильним класифікаційним позначенням є важливим етапом підготовки даних для тренування та оцінки моделей.

Поміж інших методів, для поліпшення класифікації ЕКГ-сигналів, можна також використовувати методи оптимізації, наприклад генетичні алгоритми.

Генетичний алгоритм заснований на еволюційних принципах і може використовуватися для підбору оптимальних параметрів моделі нейромережі, що впливають на її точність та ефективність у класифікації аритмій.

Існує значна кількість досліджень, присвячених класифікації ЕКГ-сигналів, і багато з них пропонують нові методи та підходи для покращення точності та надійності класифікації. Розуміння основних концепцій та технік роботи з методами машинного навчання

2.2 Етап попередньої обробки даних

2.2.1 Нормалізація

Нормалізація є важливим етапом підготовки даних перед подачею їх на модель класифікації ЕКГ. Вона допомагає забезпечити однаковий масштаб і розподіл значень усіх ознак, що покращує результати навчання та передбачення моделі. Для досягнення цього, застосовуються різні методи нормалізації, такі як масштабування діапазону (min-max scaling) та стандартизація (z-score normalization).

Масштабування діапазону (min-max scaling) перетворює значення ознак таким чином, щоб вони лежали в певному діапазоні, наприклад, від 0 до 1. Це досягається шляхом віднімання мінімального значення ознаки і поділу на різницю між максимальним і мінімальним значеннями. Такий підхід особливо корисний, коли маємо різні масштаби вхідних даних.

Стандартизація (z-score normalization) перетворює значення ознак таким чином, що вони мають середнє значення рівне 0 і стандартне відхилення рівне 1. Це досягається шляхом віднімання середнього значення ознаки і поділу на стандартне відхилення. Стандартизація дозволяє уникнути проблеми нерівномірного розподілу даних та полегшує процес навчання моделі.

Обрання відповідного методу нормалізації залежить від характеристик вхідних даних та вимог до моделі класифікації ЕКГ. Коректне застосування нормалізації допомагає забезпечити стабільність та швидкість навчання моделі, а також покращує її здатність до правильної класифікації ЕКГ сигналів.

2.2.2 Балансування класів

Балансування класів у задачі класифікації ЕКГ відіграє вирішальну роль у передобробці даних. Цей крок має на меті забезпечити рівномірний розподіл зразків у кожному класі, особливо у випадках незбалансованості, де кількість зразків у класах відрізняється значно. Незбалансованість класів може негативно впливати на процес навчання моделі та якість прогнозування.

Існує кілька ефективних методів балансування класів, які можна використовувати в контексті класифікації ЕКГ. Наприклад, метод випадкового підбирання зразків з меншою кількістю може допомогти забезпечити рівномірний розподіл класів. Інший підхід - створення штучних зразків для меншого класу або зменшення кількості зразків для більшого класу. Використання цих методів спрямоване на досягнення рівномірного розподілу зразків у всіх класах та покращення результатів класифікації ЕКГ.

Правильна попередня обробка даних, яка включає як нормалізацію, так і балансування класів, має велике значення для покращення якості моделі та її здатності до правильного прогнозування на незалежних даних. Ці процеси сприяють досягненню стабільності, підвищенню швидкості навчання та покращенню точності класифікації моделі, що дозволяє зменшити вплив факторів, таких як незбалансованість та нерівномірний розподіл даних, і підвищити надійність результатів класифікації ЕКГ.

Крім балансування класів, існують інші підходи до попередньої обробки даних в задачах класифікації ЕКГ, які можуть бути корисними. Наприклад, видалення шуму, відфільтрування сигналів, вибір релевантних ознак та інші. Ці методи допомагають покращити якість вхідних даних та забезпечити більш точні результати класифікації ЕКГ.

2.3 Шари згорткової нейронної мережі

2.3.1 Одновимірний оператор згортки

Одновимірна згортка виконує незамінну роль у згорткових нейронних мережах, де вона відіграє ключову роль у виявленні локальних залежностей та важливих ознак у вхідних одновимірних даних, таких як часові ряди або сигнали. Цей шар є основним будівельним блоком, що дозволяє моделі автоматично розпізнавати і аналізувати структуру та контекст даних, що виявляються в різних частинах сигналу.

У процесі одновимірної згортки використовується фільтр, який рухається по вхідним даним з певним кроком, а його розмір може бути встановлений як фіксований параметр. Фільтр здійснює згортку з вхідними даними шляхом перемноження його значень на відповідні значення вхідних даних та обчислення суми цих добутків. Цей процес сприяє виділенню локальних особливостей, структур та шаблонів вхідних даних, які відображаються в результуючому значенні згортки.

Застосування одновимірної згортки до всього вхідного сигналу дозволяє моделі виявляти різноманітні ознаки та шаблони на різних місцях сигналу, що сприяє збагаченню репрезентації даних та поліпшенню розпізнавання зв'язків між ними. Крім того, шари згортки можуть мати різну кількість фільтрів, що дозволяє моделі одночасно виявляти та аналізувати різні аспекти даних.

Одновимірна згортка виявляється важливим етапом попередньої обробки даних в контексті згорткових нейронних мереж, оскільки вона допомагає моделі вивчити локальні шаблони та ознаки вхідних даних, що становить основу для подальшої класифікації, регресії або прогнозування. Завдяки згортковому шару модель стає здатною до більш точного розпізнавання та інтерпретації характеристик даних, що забезпечує її здатність до зроблення зрозумілих та раціональних висновків на основі вхідних одновимірних сигналів.

2.3.2 Макс-пулінг (Max Pool)

Макс-пулінг є операцією зменшення розміру мапи ознак у згорткових нейронних мережах. Він використовується для виділення найважливіших ознак і зниження обчислювальної складності моделі.

Під час макс-пулінгу, вхідна мапа ознак розбивається на неперекриваючі області і з кожної області вибирається максимальне значення. Це значення стає новим значенням для цієї області у вихідній мапі ознак.

Головна ідея макс-пулінгу полягає в тому, що найважливіші ознаки зберігаються в найбільш активних пікселях мапи ознак. Він допомагає знизити розмір мапи ознак, зберігаючи при цьому найбільш істотну інформацію. Це також сприяє уникненню перенавчання моделі, оскільки менше параметрів підлягає тренуванню.

Макс-пулінг є інваріантним до малих зміщень у вхідних даних, що робить модель більш стійкою до незначних змін у положенні або формі об'єктів на зображеннях. Він також допомагає контролювати перенавчання, зменшуючи кількість параметрів, які підлягають оптимізації.

Макс-пулінг є важливою складовою згорткових нейронних мереж, сприяючи здатності моделі виявляти важливі ознаки у вхідних даних та забезпечувати зменшення розміру даних для подальшої обробки і аналізу.

2.3.3 Адаптивний середній пулінг(adaptive average pooling)

Методом зменшення розміру мапи ознак у згорткових нейронних мережах. Вона використовується для перетворення вхідної мапи ознак на фіксований розмір незалежно від розміру вхідних даних.

Одним зі способів обробки даних у згорткових нейромережах є адаптивна середня згортка (adaptive average pooling). Цей підхід відрізняється від звичайного макс-пулінгу, де вибирається максимальне значення у кожній області мапи ознак. Замість цього, адаптивна середня згортка обчислює середнє значення пікселів у кожній області.

Головна перевага адаптивної середньої згортки полягає в тому, що вона забезпечує рівномірне розподілення інформації по всій мапі ознак. У результаті цього процесу отримується фіксований розмір мапи ознак, незалежно від розміру вхідних даних. Це дозволяє забезпечити інваріантність до розміру зображень і забезпечує стабільність при роботі з різними розмірами вхідних даних.

Для використання адаптивної середньої згортки, встановлюється фіксований розмір виходу, наприклад, 7×7 . Затим, вхідні дані автоматично адаптуються до цього розміру шляхом обчислення середнього значення пікселів у кожній області 7×7 . Таким чином, незалежно від початкового розміру вхідних даних, виходом буде мапа ознак фіксованого розміру.

Адаптивна середня згортка є корисним інструментом для забезпечення глобального контексту та інваріантності до розміру в задачах зорового сприйняття, включаючи класифікацію зображень, детекцію об'єктів та сегментацію зображень. Вона допомагає зменшити кількість параметрів мережі та покращити ефективність навчання, зберігаючи важливу інформацію з вхідних даних.

Таким чином, адаптивна середня згортка є потужним інструментом у згорткових нейромережах, який допомагає забезпечити стабільність та незалежність від розміру вхідних даних, зберігаючи важливу інформацію для подальшого аналізу та класифікації.

Адаптивна середня згортка особливо корисна, коли розмір вхідних даних може варіюватись. Вона дозволяє моделі працювати зі змінними розмірами вхідних даних, що робить її більш універсальною і зручною для застосування на різних наборах даних.

Застосування адаптивної середньої згортки після згорткових шарів дозволяє отримати фіксований розмір мапи ознак перед подальшою обробкою і класифікацією. Це полегшує обчислювальний процес та забезпечує стабільність моделі до зміни розміру вхідних даних.

2.3.4 Батч-нормалізація

Метод батч-нормалізації є одним з ключових підходів до нормалізації активаційних значень у згорткових нейронних мережах. Його використання є особливо важливим у випадках, коли маємо складні архітектури мереж з багатьма шарами, оскільки воно допомагає забезпечити стійку та ефективну оптимізацію моделі.

Процес батч-нормалізації включає кілька кроків. Спочатку, на кожному шарі згортки або повнозв'язаному шарі, активації обчислюються для кожного екземпляра даних в батчі. Це дозволяє враховувати внутрішній структурний контекст даних та використовувати більш точну інформацію для нормалізації.

Наступним кроком є обчислення середнього значення і стандартного відхилення цих активацій. Ці статистики використовуються для стандартизації активацій до нульового середнього значення та одиничного стандартного відхилення. Це дозволяє моделі працювати з активаціями, які знаходяться в діапазоні близькому до 0, що полегшує процес оптимізації та забезпечує більш стійкий градієнт.

Батч-нормалізація має кілька переваг. По-перше, вона сприяє стабілізації та прискоренню тренування моделі. Це досягається шляхом зниження впливу вибухаючого або зникаючого градієнту, що може стати проблемою при глибоких мережах. Батч-нормалізація також допомагає уникнути проблеми з вибірковістю швидкості навчання, що сприяє більш стабільному та прогресивному тренуванню.

Крім того, батч-нормалізація полегшує оптимізацію моделі, оскільки вона робить функцію активації незалежною від масштабу та зміщення. Це дає моделі більшу свободу вибору оптимальних значень ваг та зміщень, сприяючи швидшому знаходженню глобального мінімуму функції втрат.

Узагальнюючи, батч-нормалізація є потужним інструментом для покращення продуктивності та стійкості нейронних мереж. Вона дозволяє ефективно використовувати ресурси обчислення, забезпечує нормалізовані та

стандаризовані активації на різних шарах моделі, та допомагає уникнути проблем, пов'язаних з незбалансованістю та невідповідністю масштабів даних.

Ключові слова: батч-нормалізація, нормалізація активацій, згорткові нейронні мережі, стандартизація, оптимізація моделі, градієнт, стабільність, ефективність тренування.

2.3.5 Функція активації GELU

GELU (Gaussian Error Linear Unit) функція активації є однією з популярних функцій активації, що застосовується у глибокому навчанні. Вона використовується для введення нелінійності у нейронних мережах та забезпечення їх здатності до нелінійної моделювання складних залежностей у вхідних даних.

GELU функція має особливу форму, що відрізняє її від інших функцій активації. Для дуже малих вхідних значень, вона наближається до нуля, що допомагає уникнути ненаявного згасання градієнту. Для більших вхідних значень, функція має значення близькі до вхідного значення, що дозволяє активаціям зберігати більшу кількість інформації та деталей.

GELU функція активно використовується у різних архітектурах нейронних мереж, особливо в контексті природної мови, обробки текстів та розпізнавання мови. Вона знайшла своє застосування у відомих моделях, таких як BERT (Bidirectional Encoder Representations from Transformers) та GPT (Generative Pretrained Transformer), де її використання сприяє покращенню якості результатів та продуктивності моделей.

Таким чином, GELU функція активації є важливою складовою для класифікації ЕКГ та інших завдань машинного навчання. Вона допомагає внести необхідну нелінійність у модель, забезпечуючи її здатність до розпізнавання складних залежностей у вхідних даних. На рис 2.1 зображено формулу функції активації $GELU(x)$ і відповідний графік .

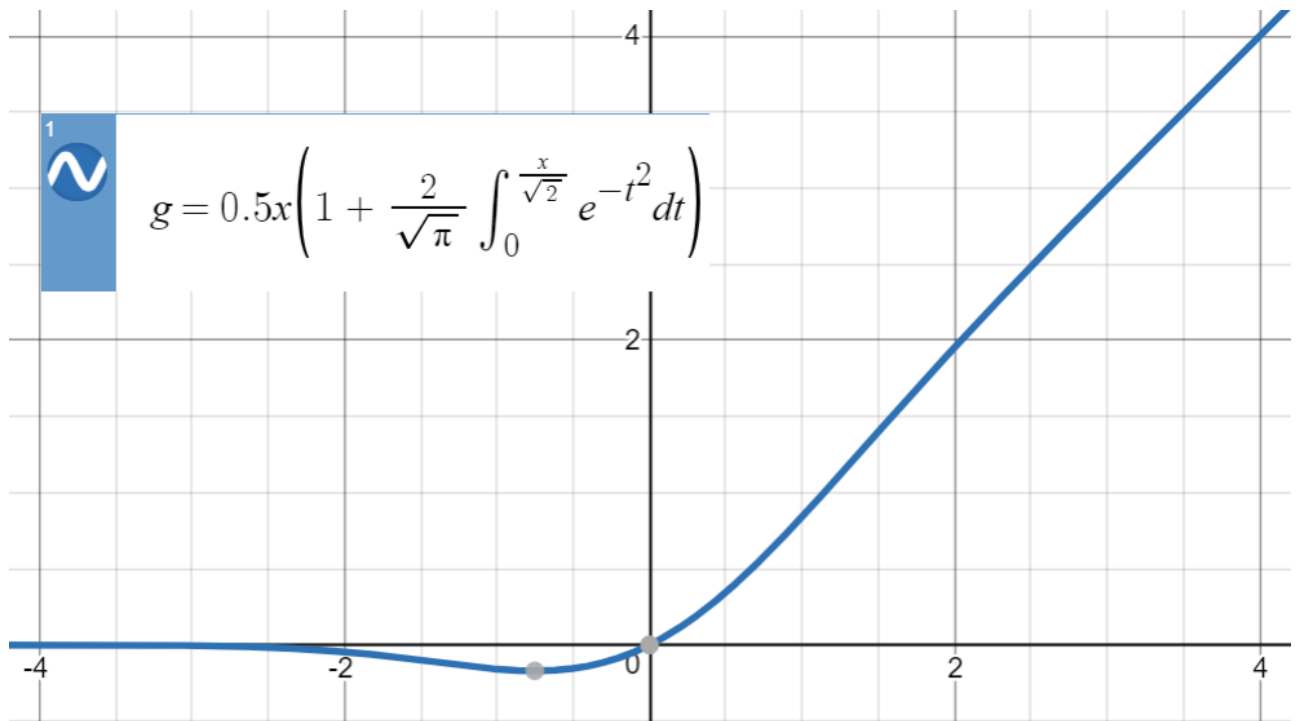


Рисунок 2.1 – Графік функції GELU(x)

2.4 Генетичний алгоритм

2.4.1 Популяція

Початкова популяція у генетичних алгоритмах складається зі сполучення генетичних репрезентацій рішень, які відповідають можливим розв'язкам задачі. Цей підхід використовує аналогію з еволюцією в природі, де генетичний матеріал у формі ДНК передається від покоління до покоління і підлягає процесу мутації та комбінації.

Кожен розв'язок у популяції представлений як хромосома, що складається з генів. Гени визначають характеристики або параметри, які описують розв'язок задачі. Наприклад, у задачі класифікації можуть бути гени, які представляють ваги або коефіцієнти моделі. У задачах оптимізації можуть бути гени, що представляють значення параметрів, які підлягають оптимізації.

Генетичний алгоритм починається зі створення випадкової початкової популяції, де кожна хромосома представляє потенційний розв'язок задачі. Після цього застосовуються генетичні оператори, такі як схрещування та мутація, що моделюють еволюційні процеси, для створення нової популяції з покращеними розв'язками.

2.4.2 Відбір

Генетичний алгоритм, який застосовується для оптимізації мережі у задачі класифікації ЕКГ, базується на принципі природного відбору. Цей механізм відбору дозволяє вибирати найкращі розв'язки з популяції, основуючись на їхній пристосованості до поставленої задачі. Цей процес схожий на той, що відбувається в природі, де краще пристосовані організми мають більшу ймовірність вижити та передати свої гени на наступне покоління.

У генетичному алгоритмі використовується популяція рішень, яка представлена різними архітектурами мережі. Кожне рішення має свої власні параметри, які визначають його якість та ефективність у виконанні завдання класифікації ЕКГ. Застосовуючи принципи природного відбору, генетичний алгоритм оцінює кожне рішення, враховуючи його пристосованість до класифікації ЕКГ. Рішення, які показують кращі результати, мають більшу ймовірність бути відібраними для подальшого розмноження та нащадків.

У процесі розмноження, генетичний алгоритм поєднує вибрані рішення, застосовуючи оператори кроссовера та мутації. Це дозволяє створювати нові комбінації параметрів, що можуть привести до поліпшення якості мережі. Наступне покоління рішень отримується шляхом повторення процесу оцінки, відбору та розмноження до досягнення заданої умови зупинки, такої як досягнення оптимальної точності класифікації.

Таким чином, генетичний алгоритм забезпечує ефективну оптимізацію параметрів мережі у задачі класифікації ЕКГ, застосовуючи природний відбір та принципи еволюції. Цей підхід дозволяє підібрати оптимальні параметри мережі, що призводить до досягнення кращої точності та надійності у класифікації аритмій на основі ЕКГ-сигналів.

Механізм відбору базується на функції пристосованості, яка оцінює якість кожного розв'язку у популяції. Ця функція може бути визначена відповідно до вимог задачі, і вона враховує, наскільки добре розв'язок вирішує поставлену задачу або які значення цільової функції він досягає. Наприклад, для задачі

класифікації ЕКГ, функція пристосованості може оцінювати точність класифікації розв'язків.

Після оцінки пристосованості кожного розв'язку, виконується відбір, де розв'язки з вищою пристосованістю мають більшу ймовірність бути вибраними для наступного покоління. Ідея полягає в тому, що такі розв'язки мають більший потенціал удосконалити наступне покоління, оскільки вони вже виявились успішними у вирішенні задачі.

Механізм відбору може бути реалізований за допомогою різних алгоритмів, таких як рулеткове колесо, турнірний відбір або елітарний відбір. Використовуючи ці методи, генетичний алгоритм здатен зберігати та покращувати якість розв'язків з покоління в покоління.

Таким чином, механізм відбору є важливою складовою генетичних алгоритмів, що дозволяє здійснювати відбір найкращих розв'язків і прискорювати збіжність алгоритму до оптимального розв'язку. У контексті класифікації ЕКГ цей механізм може допомогти знайти оптимальні моделі класифікації, які максимально точно розпізнають певні патологічні стани серця на основі ЕКГ сигналів.

2.4.3 Схрещування

Після відбору найкращих розв'язків з популяції настає етап схрещування (кросоверу). Схрещування є процесом обміну генетичним матеріалом між обраними батьківськими розв'язками з метою створення нових розв'язків (потомків).

Під час схрещування, обрані батьківські розв'язки (хромосоми) сполучаються, і їхні генетичні елементи обмінюються. Це призводить до створення нових комбінацій генів у потомків. Спосіб схрещування може бути різним, в залежності від вибраної стратегії алгоритму.

Наприклад, одним з поширених методів схрещування є одноточковий кросовер. При цьому випадковим чином обирається точка поділу (точка перетину) в генетичних послідовностях батьківських хромосом. Генетичний

матеріал до цієї точки залишається незмінним у потомків, а генетичний матеріал після точки перетину обмінюється між батьківськими хромосомами.

Схрещування має на меті комбінувати корисні генетичні властивості батьківських розв'язків у потомків, створюючи нові комбінації генів. Цей процес дозволяє популяції розв'язків шукати оптимальніші рішення у просторі пошуку. Залежно від природи задачі і стратегії алгоритму, можуть використовуватись різні види схрещування, такі як одноточковий, багатоточковий, рівномірний та інші.

В контексті класифікації ЕКГ, схрещування може означати комбінацію алгоритмів, параметрів чи методів, що використовуються для аналізу сигналів. Шляхом комбінування різних елементів можливо отримати нові підходи та стратегії, які можуть покращити якість класифікації ЕКГ сигналів.

2.4.4 Мутація

Крім схрещування, в генетичному алгоритмі застосовується ще одна важлива операція - мутація. Мутація є випадковим процесом, під час якого в генетичний матеріал розв'язків вносяться випадкові зміни. Цей процес є ключовим елементом генетичного алгоритму, оскільки він забезпечує різноманітність та дослідження нових областей пошуку.

Мутація дозволяє модифікувати генетичний матеріал, змінюючи значення окремих генів або їх комбінацій. Випадкові зміни в генетичному матеріалі можуть включати заміну, видалення або додавання генів. Ці випадкові зміни дозволяють розв'язку відхилитися від поточного рішення та дослідити нові регіони простору пошуку. Якщо виявиться, що в новому регіоні знаходиться краще рішення, то мутація дозволяє зберегти це нове рішення та продовжити його вдосконалення.

Одним з ключових аспектів мутації є її ймовірність. Ймовірність мутації визначає, наскільки часто будуть застосовуватися мутації до розв'язків в генетичному алгоритмі. Якщо ймовірність мутації висока, то шанси на внесення випадкових змін у генетичний матеріал значно збільшуються, що сприяє швидшому дослідженню простору пошуку. З іншого боку, якщо ймовірність

мутації низька, то процес мутації буде менш інтенсивним, але збереже деяку рівновагу між різноманіттям та експлуатацією поточних розв'язків.

Мутація є важливим елементом генетичного алгоритму, оскільки вона дозволяє уникнути передчасної збіжності до локального оптимуму. Вона забезпечує постійну різнообразність в популяції розв'язків, що сприяє пошуку глобальних оптимальних рішень. Комбінація мутації та схрещування в генетичному алгоритмі дозволяє ефективно експлуатувати та досліджувати простір пошуку, що призводить до знаходження оптимальних рішень у складних задачах оптимізації.

Мутація вносить різноманітні зміни в генетичний матеріал, такі як заміна гена на інший, інверсія послідовності генів, додавання або видалення гена. Ці випадкові зміни створюють нові варіанти розв'язків, які можуть бути відмінними від батьківських розв'язків.

Мутація є важливою складовою генетичного алгоритму, оскільки вона дозволяє розв'язкам "вийти" з локальних оптимумів і досліджувати нові області пошуку. Це особливо важливо, коли пошук оптимального розв'язку може зачепити плато або впасти в локальний мінімум. Мутація дозволяє зберегти різноманітність популяції розв'язків і продовжувати пошук оптимального рішення.

У контексті класифікації ЕКГ, мутація може означати внесення випадкових змін у параметри алгоритму або способу обробки сигналу. Це може допомогти виявити нові особливості сигналу або покращити точність класифікації шляхом експерименту з різними варіантами обробки сигналу.

2.4.5 Оцінювання

Кожен згенерований розв'язок у генетичному алгоритмі оцінюється за допомогою функції пристосованості, яка визначає, наскільки добре розв'язок вирішує задану оптимізаційну задачу. Функція пристосованості, також відома як цільова функція, є ключовим компонентом генетичного алгоритму.

Функція пристосованості може бути спеціально підібраною для конкретної задачі або залежати від потреб користувача. Її обрано таким чином, щоб вона

максимізувала або мінімізувала задане критерій ефективності. Наприклад, у задачі класифікації ЕКГ, функція пристосованості може вимірювати точність класифікації розв'язку.

Функція пристосованості приймає розв'язок в якості вхідних даних і повертає числове значення, яке представляє якість цього розв'язку. Чим більше значення функції пристосованості, тим краще розв'язок. Застосування відповідної функції пристосованості дозволяє генетичному алгоритму визначати, які розв'язки є найкращими і використовувати їх для подальшого еволюційного процесу.

В процесі генетичного алгоритму, розв'язки з вищим значенням функції пристосованості мають більшу ймовірність бути вибраними для схрещування та формування наступного покоління розв'язків. Це принцип природного відбору, де кращі розв'язки мають більшу ймовірність передачі своїх генетичних характеристик на наступне покоління.

Функція пристосованості є ключовим елементом генетичного алгоритму, оскільки вона визначає напрямок еволюції популяції розв'язків і дозволяє знаходити оптимальні розв'язки задачі оптимізації.

2.4.6 Елітарний відбір

У генетичному алгоритмі, найкращі розв'язки, які іноді називають "елітними", зберігаються без змін у наступному поколінні. Цей підхід сприяє збереженню найкращих характеристик та запобігає втраті цінної інформації, яку ці розв'язки представляють.

Елітні розв'язки є основою для наступного покоління, оскільки вони демонструють найвищу якість або найкращу відповідь на задану проблему. Збереження цих розв'язків забезпечує стабільність та поступове поліпшення генетичного алгоритму з кожним наступним поколінням.

Проте, наступне покоління не повністю залежить від елітних розв'язків. Застосування операторів схрещування та мутації дозволяє досліджувати нові комбінації генетичного матеріалу, що веде до різноманітності в популяції. Це

дозволяє генетичному алгоритму здійснювати пошук у просторі розв'язків та виявляти нові, можливо, ще кращі рішення.

Таким чином, комбінація елітних розв'язків та нових варіантів, отриманих за допомогою схрещування та мутації, забезпечує баланс між збереженням найкращих характеристик та дослідженням нових областей пошуку. Цей підхід дозволяє генетичному алгоритму працювати ефективно та швидко знаходити оптимальні розв'язки в складних задачах оптимізації.

Збереження найкращих розв'язків у наступне покоління гарантує, що ці розв'язки залишаються в популяції і можуть продовжувати впливати на еволюцію розв'язків у майбутніх поколіннях. Це важливо, оскільки найкращі розв'язки мають цінну інформацію про оптимальні стратегії або характеристики, які ведуть до досягнення бажаного результату. Збереження їх допомагає уникнути втрати цієї цінної інформації і сприяє подальшому поліпшенню популяції розв'язків з кожним наступним поколінням.

Окрім збереження найкращих розв'язків, генетичний алгоритм також використовує схрещування та мутацію для генерації нових розв'язків, що дозволяє розширювати пошуковий простір та досліджувати нові комбінації генетичного матеріалу. Таке поєднання збереження найкращих розв'язків та створення нових розв'язків через схрещування дозволяє генетичному алгоритму ефективно знаходити оптимальні розв'язки задачі оптимізації.

2.4.7 Повторення

Генетичний алгоритм виконує повторення вищезазначених кроків - відбору, схрещування, мутації та оцінювання - протягом кількох поколінь або досягнення певної умови зупинки. Кожне наступне покоління стає покращеним у порівнянні з попереднім, оскільки воно успадковує найкращі характеристики розв'язків з попереднього покоління та досліджує нові комбінації генетичного матеріалу.

Процес відбору гарантує, що розв'язки з кращою пристосованістю мають більшу ймовірність бути вибраними для схрещування та наступного покоління.

Це дозволяє передавати корисні генетичні властивості нащадкам і зберігати високу якість розв'язків у популяції.

Схрещування вносить варіацію у генетичний матеріал шляхом обміну генами між батьківськими розв'язками. Це створює нові комбінації генів і дозволяє відкривати нові шляхи в пошуку оптимальних розв'язків.

Мутація вносить випадкові зміни в генетичний матеріал розв'язків, що допомагає досліджувати нові області пошуку та уникати передчасної збіжності до локального оптимуму. Це забезпечує різноманітність у популяції та збільшує шанси знайти глобально оптимальний розв'язок.

Оцінювання розв'язків за допомогою функції пристосованості визначає, наскільки добре розв'язок вирішує задану задачу оптимізації. Ця функція дозволяє порівнювати розв'язки між собою та відбирати найкращі для наступного покоління.

Повторення цих кроків протягом кількох поколінь дозволяє генетичному алгоритму систематично покращувати популяцію розв'язків. Кожне нове покоління успадковує найкращі характеристики з попереднього покоління та вносить варіацію через схрещування. Це сприяє поступовому зближенню до оптимального розв'язку та знаходженню найкращого рішення задачі оптимізації.

2.4.8 Збір результатів

Після завершення генетичного алгоритму, один або кілька найкращих знайдених розв'язків вибираються як результат. Ці розв'язки є потенційно оптимальними рішеннями для поставленої оптимізаційної задачі.

Обрані розв'язки можуть бути використані для подальшого аналізу та оцінки їх ефективності. Вони можуть надати цінну інформацію про оптимальний набір параметрів, налаштування або конфігурацію, яка дозволяє досягти найкращих результатів. Ця інформація може бути використана для прийняття рішень щодо подальшого вдосконалення системи, оптимізації процесу або вибору найкращого варіанту.

В результаті генетичного алгоритму найкращі розв'язки можуть використовуватись у різних способах, які дозволяють поліпшити процес оптимізації і зекономити ресурси.

По-перше, ці найкращі розв'язки можуть слугувати початковою точкою для подальшого вдосконалення. Замість починати оптимізацію спочатку з випадкових розв'язків, ми можемо використати найкращі розв'язки, отримані після запуску генетичного алгоритму, як вхідну точку для інших алгоритмів. Це дозволяє прискорити процес знаходження оптимального рішення, оскільки ми вже маємо добрий стартовий пункт.

По-друге, найкращі розв'язки можуть бути використані як приближені рішення. У деяких випадках, коли точність не є критичною, але швидкість виконання і ефективність є важливими, можна використовувати ці найкращі розв'язки як швидке наближення до оптимального рішення. Це дозволяє економити час і ресурси, оскільки немає потреби в додаткових обчисленнях для отримання точного рішення.

Таким чином, найкращі розв'язки, отримані в результаті генетичного алгоритму, можуть мати значну цінність і після завершення самого алгоритму. Вони можуть слугувати початковою точкою для подальшого вдосконалення або використовуватись як приближені рішення, які дозволяють економити час і ресурси при здійсненні подальших обчислень.

Враховуючи результати генетичного алгоритму, можна здійснити остаточний вибір оптимального розв'язку залежно від вимог, обмежень та умов конкретної задачі. Застосування генетичних алгоритмів в оптимізаційних задачах дозволяє знайти ефективні розв'язки, які були б складні або неможливі досягти за допомогою традиційних методів.

Висновки до розділу 2

Після обробки даних та вибору відповідної архітектури мережі, генетичний алгоритм може бути використаний для автоматичного підбору

оптимальних параметрів моделі. Це включає в себе використання генетичних операцій, таких як селекція, схрещування та мутація, для еволюції популяції параметрів та пошуку оптимальної конфігурації.

Селекція дозволяє відбирати найкращі розв'язки (тобто комбінації параметрів) з популяції на основі їхньої пристосованості до класифікації ЕКГ-сигналів. Розв'язки, які демонструють вищу точність або ефективність, мають більшу ймовірність бути обраними для формування наступного покоління.

Схрещування (кросовер) використовується для обміну генетичним матеріалом між обраними розв'язками. Це дозволяє створювати нові комбінації параметрів шляхом комбінування їхніх характеристик. За допомогою схрещування в поколінні можуть з'явитися розв'язки з новими властивостями, які можуть виявитися більш пристосованими до класифікації ЕКГ-сигналів.

Мутація вносить випадкові зміни в генетичний матеріал розв'язків. Це дозволяє досліджувати нові області пошуку та уникати передчасної збіжності до локального оптимуму. Мутація може вносити невеликі випадкові зміни в параметри моделі, що дозволяє досліджувати різні варіанти та шукати кращі комбінації параметрів.

Таким чином, за допомогою генетичного алгоритму можна автоматично оптимізувати параметри архітектури мережі для класифікації ЕКГ-сигналів. Цей підхід дозволяє знайти оптимальні комбінації параметрів, які забезпечують кращу точність та ефективність класифікації, що є важливим для успішного аналізу ЕКГ-сигналів та виявлення патологічних станів.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Основні використані бібліотеки

3.1.1 Numpy

Numpy є однією з найпопулярніших бібліотек для наукових обчислень у Python. Вона надає потужні інструменти для роботи з масивами даних, що значно спрощує обробку та маніпуляцію даними. Одна з основних переваг Numpy - це його ефективність. Внутрішні реалізації Numpy використовують мову програмування C, що дозволяє виконувати операції швидко та ефективно, особливо над великими об'ємами даних.

Numpy надає високоефективні функції для математичних операцій, лінійної алгебри, статистики та інших обчислень. Завдяки цьому, ви можете легко виконувати різні операції над масивами даних, такі як обчислення статистичних показників, множення матриць, арифметичні операції та багато інших. Крім того, Numpy надає зручні функції для індексування, сортування, фільтрації та перетворення даних.

В контексті попередньої обробки ЕКГ-сигналів, використання бібліотеки Numpy виявляється незамінним інструментом, який дозволяє зручно та швидко виконувати різноманітні операції з масивами даних. Нижче наведено кілька способів, які демонструють переваги використання Numpy у попередній обробці ЕКГ-сигналів:

Фільтрація сигналу: Numpy надає можливість застосовувати різні фільтри для очищення ЕКГ-сигналу від шумів і артефактів. Завдяки широкому спектру доступних фільтрів, таких як фільтр нижніх частот, фільтр вищих частот, фільтр ковзного середнього тощо, ви можете поліпшити якість сигналу і видалити небажані складові.

Нормалізація даних: Numpy надає зручні функції для нормалізації ЕКГ-сигналів, що дозволяє забезпечити стандартні умови обробки даних. Ви можете використовувати функцію нормалізації для приведення значень сигналу до

діапазону від 0 до 1 або до іншого заданого діапазону. Це важливий крок, оскільки нормалізовані дані забезпечують стабільність та уніфікований підхід до обробки.

Перетворення сигналу: NumPy надає потужні функції для виконання різноманітних перетворень ЕКГ-сигналів, таких як Фур'є-перетворення, хвильове перетворення і спектрограми. Ці перетворення дозволяють виділити важливі частотні складові сигналу та отримати додаткову інформацію, яка може використовуватись у подальшому аналізі та класифікації.

Маніпулювання та обробка даних: NumPy забезпечує багатий набір функцій для маніпулювання та обробки даних. Ви можете вирізати, копіювати, з'єднувати та розбивати дані ЕКГ-сигналів у зручний спосіб. Крім того, ви можете виконувати математичні операції, обчислювати статистичні показники, а також виконувати інші операції, що допомагають вам розуміти та аналізувати дані ЕКГ.

Загалом, використання бібліотеки NumPy у попередній обробці ЕКГ-сигналів дозволяє ефективно та зручно маніпулювати та обробляти дані, що є необхідним етапом у побудові нейронних мереж для класифікації ЕКГ. Вона надає багатий функціонал, який полегшує виконання операцій з масивами даних і сприяє отриманню якісних результатів у обробці ЕКГ-сигналів.

3.1.2 PyTorch

PyTorch є одним з найпопулярніших фреймворків для машинного та глибокого навчання. Він надає розширені інструменти та функціональні можливості для побудови та навчання нейронних мереж. PyTorch є особливо популярним серед дослідників та практиків, оскільки він пропонує зручну реалізацію складних алгоритмів навчання та має потужні інструменти для оптимізації моделей.

Одна з головних переваг PyTorch - це його автоматичне диференціювання. Це означає, що ви можете визначити нейронну мережу та виконувати навчання, а фреймворк автоматично обчислить градієнти для вас. Це дуже зручно при роботі зі складними моделями та оптимізацією параметрів.

PyTorch надає багато вбудованих функцій та компонентів для побудови нейронних мереж. Ви можете використовувати різні функції активації, шари

згортки, рекурентні шари та інші модулі, щоб створити потужну та гнучку модель для обробки сигналів. Крім того, PyTorch надає широкий набір функцій для роботи з даними, включаючи завантаження, пакування та передачу даних у модель.

Використання PyTorch у даній роботі дозволяє побудувати та навчити нейронну мережу для класифікації ЕКГ-сигналів. Фреймворк надає потужні інструменти для ефективного навчання моделей, оптимізації параметрів та оцінки їх продуктивності. Завдяки PyTorch ви можете ефективно працювати з глибоким навчанням та створювати складні моделі, що здатні вирішувати складні завдання, такі як класифікація ЕКГ-сигналів.

Окрім цього, PyTorch має активну спільноту користувачів та розробників, що забезпечує доступ до багатой бази знань, документації та прикладів коду. Ви можете знайти багато ресурсів, які допоможуть вам вивчити PyTorch та застосовувати його для вашої конкретної задачі класифікації ЕКГ-сигналів.

Також варто відзначити, що PyTorch має підтримку обчислень на GPU, що дозволяє прискорити процес навчання за допомогою паралельних обчислень. Це особливо корисно при роботі з великими обсягами даних та складними моделями.

Загалом, використання PyTorch у даній роботі дозволяє побудувати потужну та ефективну модель для класифікації ЕКГ-сигналів. Фреймворк надає зручні інструменти, широкий набір функцій та підтримку для GPU, що допомагають вам зробити вашу роботу більш ефективною та продуктивною.

3.1.3 Sci-kit learn

Scikit-learn (sklearn) є однією з найпопулярніших бібліотек машинного навчання в Python і надає широкий спектр алгоритмів для класифікації, регресії, кластеризації, зменшення розмірності та багатьох інших завдань. У даній роботі scikit-learn використовується для реалізації функцій обчислення пристосованості індивіда та оцінки якості навчених моделей.

Деякі ключові функції scikit-learn, які використовуються в цій роботі, включають:

1. `train_test_split`: Ця функція використовується для розбиття вибірки на тренувальну і тестову. Вона допомагає зробити випадкове розподіл даних між

тренувальною та тестовою вибірками. Тренувальна вибірка використовується для навчання моделі, а тестова вибірка використовується для оцінки точності моделі.

2. `KFold`: Ця функція використовується для обчислення значення функції пристосованості індивіда генетичного алгоритму. Вона розбиває дані на тренувальні і тестові набори k разів, забезпечуючи те, що тестові дані не накладаються і множина тренувальних даних є компланарною до тестових.

3. `roc_auc_score`: Ця функція використовується для обчислення точності моделі відповідно до помилок, які робить модель, використовуючи конфузійну матрицю. ROC-AUC (Receiver Operating Characteristic - Area Under Curve) є метрикою, яка вимірює дискримінативну здатність моделі та її здатність розрізняти класи.

3.2 Попередня обробка даних

3.2.1 Нормалізація сигналів ЕКГ

Нормалізація сигналів ЕКГ перед тренуванням моделі є важливим кроком, оскільки це допомагає зробити дані більш узагальненими та зручними для обробки. При нормалізації сигналів ЕКГ досягається стандартизація їхнього масштабу та розподілу, що сприяє кращому навчанню моделі та збільшує шанси на досягнення точної класифікації.

Одним із поширених підходів до нормалізації є мінімаксна нормалізація, де дані перетворюються таким чином, щоб їхні значення лежали в певному діапазоні, наприклад, від 0 до 1 або від -1 до 1. Це забезпечує стабільний масштаб та дозволяє моделі більш ефективно працювати з даними різних амплітуд та діапазонів.

Інший підхід - це нормалізація за допомогою середнього значення та стандартного відхилення (метод середнього-стандартного відхилення). При такому підході значення сигналу ЕКГ віднімаються на середнє значення та діляться на стандартне відхилення. Це призводить до отримання сигналів з нульовим середнім значенням та одиничним стандартним відхиленням, що полегшує обробку та порівняння даних.

У даній роботі був використаний підхід з нормалізацією за допомогою стандартного відхилення. Це означає, що кожен сигнал ЕКГ був відніманий від його середнього значення та ділився на стандартне відхилення, з метою зміщення середнього значення близько до нуля. Це допомагає забезпечити однаковий масштаб та розподіл сигналів, що полегшує навчання моделі та забезпечує кращу узагальнюючу здатність.

Нормалізація сигналів ЕКГ перед тренуванням моделі є важливою практикою, яка допомагає підвищити якість класифікації та забезпечити стабільну та ефективну роботу моделі на нових даних.

3.2.2 Балансування даних

Іншим важливим кроком попередньої обробки даних є балансування даних. Балансування даних означає перетворення даних таким чином, щоб кількість прикладів кожного класу була приблизно однаковою. Це особливо важливо у випадку, коли кількість прикладів у різних класах суттєво відрізняється, оскільки навчання моделі на незбалансованих даних може призвести до викривлення результатів та низької ефективності моделі.

У цій роботі для балансування даних було застосовано підхід випадкового вибору прикладів з менш представлених класів так, щоб кількість прикладів кожного класу стала рівною. Це означає, що при формуванні навчального набору дані бралися випадкові приклади з менш представлених класів до тих пір, поки кількість прикладів в кожному класі не стала близькою до кількості прикладів в найбільш представленому класі. Таким чином, модель отримує однакову кількість прикладів з кожного класу під час навчання, що сприяє кращому вивченню характеристик індивідуальних класів і покращує ефективність моделі в умовах незбалансованих даних.

Балансування даних допомагає забезпечити, що модель буде навчатися на репрезентативних прикладах з кожного класу, а не перекреслювати результати через перевагу більш представлених класів. Це може покращити загальну здатність моделі до узагальнення і прогнозування на нових, реальних даних.

3.3 Архітектура мережі

3.3.1 Згорткові нейронні мережі

Згорткові нейронні мережі (ЗНМ) є потужними інструментами в області машинного навчання і особливо ефективними у контексті класифікації електрокардіографічних (ЕКГ) сигналів. Використання ЗНМ дозволяє автоматично виявляти складні залежності та особливості в ЕКГ сигналах, що допомагає у визначенні різних класів аритмій, аномалій серцевого ритму та інших патологій.

Однією з ключових особливостей ЗНМ є їх здатність до виявлення локальних шаблонів у вхідних даних. Це досягається за допомогою операцій згортки, які сканують вхідний сигнал за допомогою фільтрів або ядер з малим розміром. Ці фільтри виявляють специфічні особливості, такі як форма хвилі, частотні компоненти або інші характеристики, що є важливими для класифікації ЕКГ сигналів. Після проведення операції згортки, виконується пулінг, що дозволяє зменшити розмір отриманого представлення, зберігаючи при цьому ключову інформацію.

Ваги ЗНМ навчаються за допомогою зворотного поширення помилки, де модель виробляє прогноз, порівнюючи його з очікуваним вихідним значенням, і відповідно коригує свої ваги. Цей процес триває кілька епох навчання, після чого модель здатна класифікувати нові ЕКГ сигнали.

У контексті класифікації ЕКГ, ЗНМ можуть бути натреновані на великому наборі маркованих ЕКГ сигналів, де кожен сигнал має відповідну мітку класу (наприклад, нормальний ритм, фібриляція передсердь, блокади тощо). Після навчання ЗНМ здатна класифікувати нові ЕКГ сигнали, розпізнаючи патологічні стани та виявляючи аномалії, що можуть бути непомітні для людського ока.

Важливим аспектом використання ЗНМ для класифікації ЕКГ є належне попереднє опрацювання даних, таке як нормалізація, фільтрація шуму та балансування класів. Правильне попереднє опрацювання даних сприяє поліпшенню ефективності моделі та точності класифікації.

Згорткові нейронні мережі є потужним інструментом для автоматичної класифікації ЕКГ-сигналів, що дозволяє медичним фахівцям та експертам

ефективно аналізувати великі обсяги даних і виявляти патології швидше і точніше. Використання згорткових нейронних мереж допомагає покращити діагностику та лікування серцево-судинних захворювань, що є важливим кроком у збереженні життя пацієнтів.

Одна з основних переваг згорткових нейронних мереж полягає в їхній здатності автоматично виявляти важливі ознаки і паттерни у ЕКГ-сигналах. Згорткові шари мережі використовують фільтри, які здійснюють складні операції згортки та пулінгу для виділення значущих інформаційних ознак з сигналу. Ці ознаки можуть включати частотні характеристики, амплітуди, часові шаблони та інші параметри, що є важливими для класифікації різних типів серцевих аномалій.

Інша перевага згорткових нейронних мереж полягає у їхній здатності до автоматичного навчання. Вони можуть бути навчені на великій кількості ЕКГ-сигналів з позначеними класами, що дозволяє їм "вивчити" відношення між вхідними сигналами та їхніми класами. Після процесу навчання, згорткова нейронна мережа може бути застосована до нових, невиданих раніше сигналів для їхньої класифікації з високою точністю.

Застосування згорткових нейронних мереж в медицині, зокрема для класифікації ЕКГ-сигналів, має великий потенціал у поліпшенні діагностики та лікування серцево-судинних захворювань. Швидке та точне виявлення патологій, таких як аритмії, ішемічна хвороба серця та інші серцеві вади, може сприяти своєчасному втручанню та запобіганню серйозним ускладненням. Згорткові нейронні мережі відкривають нові перспективи для медичної діагностики та допомагають покращувати якість життя пацієнтів з серцево-судинними захворюваннями.

3.3.2 Блок згортки

В даній роботі «Блок згортки» є ключовою складовою розроблюваної моделі і відіграє важливу роль у класифікації ЕКГ сигналів. Цей блок включає в себе кілька послідовних шарів, які спільно працюють для витягування корисних ознак з сигналу. Перший шар - функція одновимірної згортки - застосовується для здійснення локальних залежностей між сусідніми даними. Цей шар використовує

фільтри, що скользять по вхідному сигналу та виконують операцію згортки. Після шару згортки використовується батч-нормалізація, яка нормалізує активації на попередньому шарі в межах кожного пакету даних. Це допомагає забезпечити стабільність та швидкість навчання моделі. Після батч-нормалізації застосовується функція активації GELU (Gaussian Error Linear Unit). Ця функція нелінійної активації допомагає внести нелінійність та гнучкість в модель. Після цього знову застосовується одновимірна згортка, яка дозволяє витягувати більш складні ознаки зі згенерованих фільтрами. Цей шар допомагає моделі розпізнавати більш високорівневі залежності в сигналі. Завершує блок функція активації GELU, яка надає нееліптичність та гнучкість моделі перед передачею результатів на наступний шар. Таким чином, блок згортки складається зі шарів згортки, батч-нормалізації та функцій активації GELU, які спільно працюють для витягування розпізнавання корисних ознак з ЕКГ сигналу. Цей процес дозволяє моделі автоматично виявляти та виділяти важливі паттерни і характеристики, що можуть бути пов'язані з різними аритміями та станами серця.

Блок згортки є ефективним і потужним інструментом у розпізнаванні ЕКГ сигналів, оскільки він дозволяє моделі працювати з різними рівнями деталей та залежностей у сигналі. Використання послідовних шарів згортки, батч-нормалізації та функцій активації GELU допомагає моделі виявляти та розпізнавати різноманітні ознаки, що можуть бути важливими для класифікації та діагностики серцево-судинних захворювань.

Крім того, блок згортки може бути повторено кілька разів у моделі, з кожним наступним шаром витягуючи все більш абстрактні та складні ознаки з сигналу. Це дозволяє моделі будувати глибші репрезентації сигналу та здатність до більш точної класифікації.

Таким чином, блок згортки є важливою складовою глибинних нейронних мереж для аналізу ЕКГ сигналів, оскільки він дозволяє моделі витягувати інформацію зі сигналу та розпізнавати характеристики, що вказують на різні стани серця та аритмії. На рис. 3.1 візуально показано концепцію блоку згортки, яка

складається із оператора одновимірної згортки, функції активації GELU і батч-нормалізація.

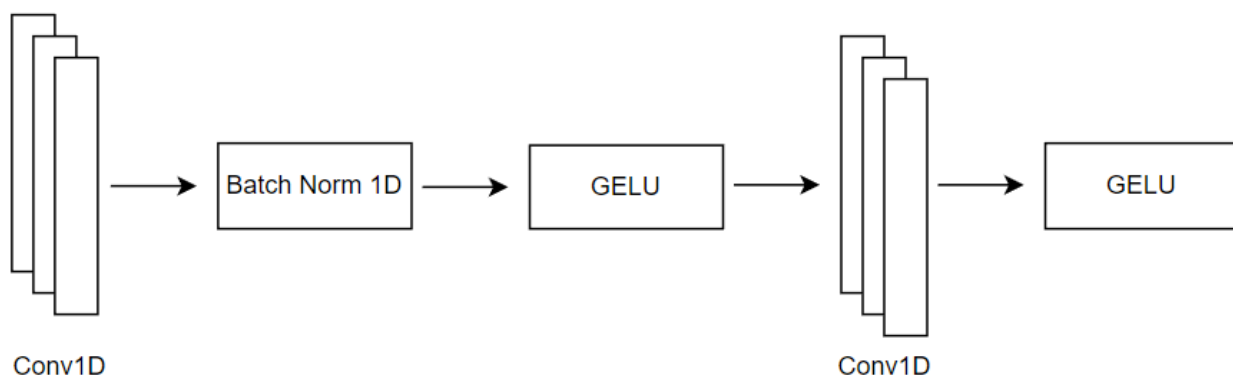


Рисунок 3.1 Блок згортки

3.3.3 Функції між блоками згортки

У розробленій моделі згорткових нейронних мереж (ЗНМ) для класифікації ЕКГ використовуються різні функції, спрямовані на покращення ефективності та регуляризацію моделі. Двома такими функціями є MaxPool і Dropout.

MaxPool використовується для зменшення розмірності даних та підвищення просторової інваріантності моделі. Ця функція розбиває вхідні дані (наприклад, зображення ЕКГ) на неперекривні регіони і обирає найбільше значення в кожному регіоні. Це дозволяє виділити найбільш значущі ознаки та зменшити розмір вхідних даних, що сприяє більш ефективній обробці та скорочує кількість параметрів моделі.

Dropout використовується для регуляризації моделі та запобігання перенавчанню. Ця функція випадковим чином "вимикає" певний відсоток нейронів у кожному шарі під час тренування. У випадку моделі ЕКГ класифікації, Dropout з параметром 0.4 означає, що при тренуванні випадковим чином вимикається 40% нейронів у кожному шарі. Це допомагає уникнути залежностей між певними нейронами та забезпечує кращу узагальнюючу здатність моделі. Dropout допомагає моделі бути більш гнучкою та універсальною, знижуючи ризик перенавчання та покращуючи її здатність до класифікації нових даних.

Ці дві функції, MaxPool і Dropout, застосовуються між блоками згортки в моделі ЗНМ для оптимізації та регуляризації моделі. Вони сприяють зменшенню розміру даних, виділенню найбільш значущих ознак та запобігають перенавчанню, забезпечуючи кращу ефективність та узагальнюючу здатність моделі ЕКГ класифікації. Ці функції є важливими складовими елементами архітектури ЗНМ і допомагають досягти кращих результатів класифікації ЕКГ сигналів.

3.3.3 Шар класифікації

Після блоку згортки, який допомагає витягнути корисні ознаки з вхідних даних, використовуються додаткові шари для подальшої обробки та класифікації.

Блок згортки: Блок згортки використовується і в шарі класифікації для додаткової обробки отриманих функцій-ознак перед подачею їх на фінальний класифікатор. Цей блок може складатися з послідовності функції одновимірної згортки, батч-нормалізації та функції активації GELU. В результаті цих операцій отримується більш високорівнева репрезентація даних, що допомагає моделі розрізняти класи більш точно.

Адаптивний середній пулінг: Адаптивний середній пулінг використовується для зменшення розмірності отриманих функцій-ознак і підготовки їх для подальшого класифікатора. Цей шар обчислює середнє значення кожного каналу функцій-ознак залежно від розміру вихідних даних. Адаптивність полягає у тому, що розмір пулінгового фільтру автоматично змінюється, щоб відповідати розміру вихідних даних. Це дозволяє моделі бути більш гнучкою та незалежною від розміру вхідних даних.

Без Softmax: У шарі класифікації, після Flatten, значення передаються безпосередньо без застосування функції активації Softmax або додаткового класифікатора. Це означає, що модель не виробляє розподіл ймовірностей для кожного класу, але використовує безпосередньо числові значення для прийняття рішення про класифікацію.

Таким чином, без використання Softmax і класифікатора в шарі класифікації, модель використовує безпосередні числові значення для вироблення рішення щодо класифікації без додаткової обробки чи нормалізації ймовірностей.

Цей підхід може бути корисним, особливо коли немає потреби в точному розподілі ймовірностей класів і важлива лише сама класифікація. Використання безпосередніх числових значень також може спростити обчислення та прискорити процес класифікації. Однак, варто зазначити, що цей підхід може бути менш інтерпретованим, оскільки він не надає інформацію про ступінь впевненості моделі в прийнятому рішенні.

3.3.4 Структура всієї мережі

На початку обробки моделі, сигнали ЕКГ, які представлені векторами розміру (1, 187), вводяться як вхідні дані. Кожен вектор містить значення сигналу для кожного з 187 часових відрізків одного кардіоциклу.

Після цього модель включає повторювану частину, яка складається з трьох основних шарів: "Блок згортки", "Макс-пулінгу" і шару "Dropout" з параметром 0.4. Кількість повторень цих шарів залежить від значень словника, де кожне значення відповідає параметрам "Блоку згортки". Ці повторювані шари допомагають моделі витягувати та узагальнювати корисні ознаки з сигналу ЕКГ.

Після повторюваної частини моделі слідує шар класифікації. Він складається з "Блоку згортки", "Адаптивного середнього пулінгу" і шару "Flatten". "Блок згортки" здійснює додаткову обробку ознак, "Адаптивний середній пулінг" зводить ознаки до фіксованого розміру, а шар "Flatten" перетворює зведені ознаки в одновимірний вектор. Виходом цього шару є логіт-вектор, який містить значення логітів для класифікації сигналів ЕКГ. У цьому випадку, безпосередньо значення логітів використовуються для визначення класу належності без застосування функції активації "Softmax".

Після отримання логіт-вектора, який містить значення логітів для класифікації сигналів ЕКГ, застосовується безпосереднє використання цих значень для визначення класу належності. Без застосування функції активації "Softmax", можна просто вибрати клас з найвищим значенням логіту як прогнозований клас для даного сигналу ЕКГ.

Такий підхід до класифікації сигналів ЕКГ забезпечує простоту та ефективність обчислень. Натомість, зазвичай використовується функція активації

"Softmax", яка перетворює логіт-вектор в вектор ймовірностей, де кожне значення вектора представляє ймовірність належності до відповідного класу. Однак, у цьому випадку безпосереднє використання логітів може бути ефективним для класифікації ЕКГ сигналів, особливо якщо потрібно просто визначити найбільш ймовірний клас.

Загалом, описана архітектура моделі для класифікації сигналів ЕКГ використовує повторювані блоки згортки, макс-пулінгу та Dropout для витягування корисних ознак з сигналу. Після цього застосовується шар класифікації, який використовує блоки згортки, адаптивний середній пулінг та Flatten для отримання логіт-вектора. Така архітектура дозволяє досягти високої точності класифікації сигналів ЕКГ та використовується для вирішення задач оптимізації параметрів згорткової мережі та параметрів навчання. На рис. 3.2 зображено структуру всієї мережі розміри і параметри значень розмірів «блоків згортки» буде підбирати ГА.

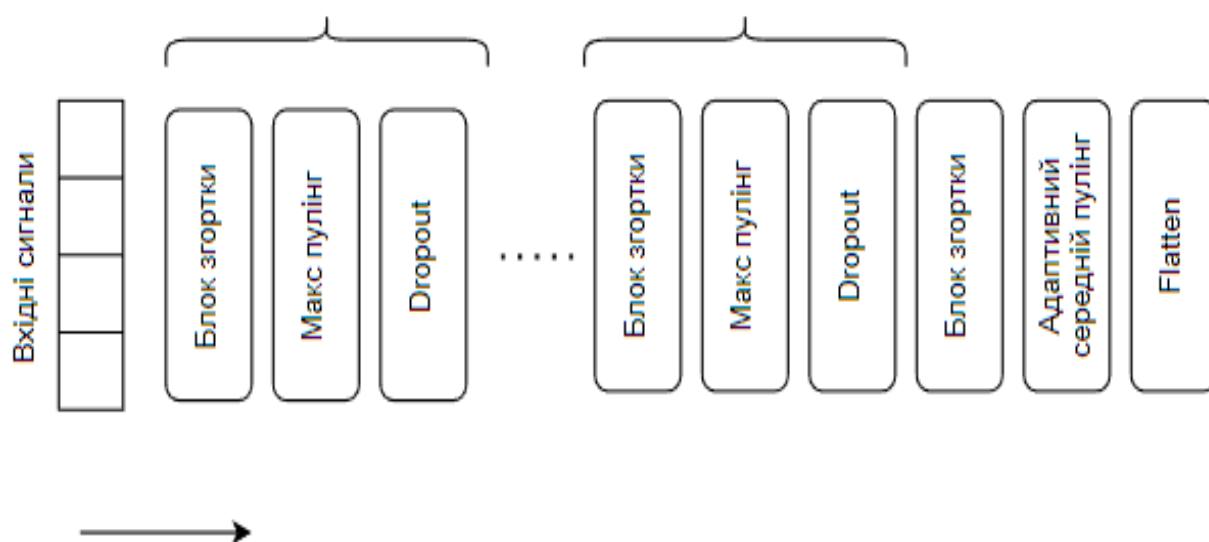


Рисунок 3.2 Схема повної нейронної мережі

3.4 Процес навчання моделі

3.4.1 Функція помилки CrossEntropy

Функція помилки CrossEntropy є широко використовуваною функцією в задачах класифікації. Вона вимірює розбіжність між прогнозованими значеннями моделі і справжніми значеннями міток класів. У випадку задачі багатокласової

класифікації, як у нашому випадку, CrossEntropy використовується для порівняння ймовірностей прогнозування моделі з розподілом справжніх значень міток класів.

Функція CrossEntropy визначається як сума добутків ймовірностей прогнозованих класів і відповідних значень логарифмів справжніх класів. Це дозволяє моделі зосередитися на правильних прогнозах і мінімізувати значення функції помилки.

У процесі навчання, функція CrossEntropy використовується як цільова функція, яку оптимізатор намагається мінімізувати шляхом зміни параметрів моделі. Мінімізуючи значення функції CrossEntropy, модель прагне досягти більш точних та надійних прогнозів класифікації.

Для обчислення значення функції CrossEntropy, втрати обчислюються для кожного окремого прикладу шляхом порівняння прогнозованої ймовірності з справжньою міткою класу. Потім обчислюється середнє значення цих втрат, яке представляє загальну помилку моделі на даних.

Застосування функції CrossEntropy в задачах багатокласової класифікації допомагає моделі ефективно навчатися і здатніше робити точні прогнози для різних класів. Вона є важливою складовою у процесі оптимізації нейронних мереж та покращення їх здатності до класифікації.

3.4.2 Оптимізатор Adam

Оптимізатор Adam є одним з найпопулярніших алгоритмів оптимізації, який широко використовується для навчання нейронних мереж. Його ефективність полягає в поєднанні переваг двох інших алгоритмів оптимізації - RMSprop і адаптивного градієнтного спуску (Adaptive Gradient Descent).

Adam використовує дві основні складові для оновлення ваг мережі. Перша складова - експоненційно зважений середній квадрат градієнту - дозволяє контролювати швидкість навчання відповідно до інтенсивності градієнту. Друга складова - експоненційно зважений середній квадрат попереднього градієнту - дозволяє адаптивно враховувати попередні зміни в градієнті при оновленні ваг.

Adam також включає механізм адаптації швидкості навчання, що дозволяє автоматично регулювати швидкість навчання залежно від характеристик градієнту.

Це дозволяє алгоритму ефективно пристосовуватись до різних режимів навчання і забезпечує стійкість процесу оптимізації.

Узагальнюючи, Adam є потужним і надійним алгоритмом оптимізації, який дозволяє ефективно навчати нейронні мережі шляхом комбінування різних методів адаптації швидкості навчання та оновлення ваг. Його широке застосування в галузі глибокого навчання свідчить про його значну корисність та важливість.

3.4.3 Градієнтний спуск

Градієнтний спуск є ключовим алгоритмом, що використовується в оптимізації моделі під час навчання. В основі градієнтного спуску лежить ідея пошуку оптимальних значень параметрів моделі шляхом оновлення цих параметрів в напрямку, протилежному градієнту функції втрати.

Процес градієнтного спуску розпочинається зі випадково заданих значень параметрів моделі. Потім для кожного прикладу з навчального набору даних обчислюється значення функції втрати. Градієнт функції втрати відносно параметрів моделі обчислюється, і параметри оновлюються в напрямку зменшення функції втрати. Цей процес повторюється для всього навчального набору даних протягом кількох епох, поки модель не досягне збитковості, яка задовольняє наші потреби.

Градієнтний спуск може бути реалізований різними способами, такими як стохастичний градієнтний спуск (SGD), міні-пакетний градієнтний спуск (mini-batch GD) або навіть методи з оптимізацією моменту. Кожен з цих підходів має свої переваги і недоліки і може використовуватись в залежності від особливостей конкретної задачі та об'єму даних.

Головна мета градієнтного спуску - знайти локальний або глобальний мінімум функції втрати, щоб підібрати параметри моделі, які найкраще відповідають навчальним даним. Це дозволяє моделі робити більш точні прогнози на нових, невиданих раніше даних.

3.4.4 Процес зменшення кроку навчання

Процес зменшення кроку навчання в залежності від епохи є важливим кроком в оптимізації навчання моделі. Одним із поширених підходів для цього є

використання алгоритму StepLR. Цей алгоритм дозволяє динамічно змінювати крок навчання після певного числа епох.

Алгоритм StepLR використовує параметр `step_size`, що визначає, через яку кількість епох необхідно зменшити крок навчання, і параметр `gamma`, який визначає величину зменшення кроку навчання. Наприклад, якщо `step_size` встановлено на 10 і `gamma` на 0.1, то крок навчання буде зменшуватись на 10-й, 20-й, 30-й і т.д. епохах, множачком 0.1.

Однак, для оптимального підбору параметрів `step_size` і `gamma`, можна використовувати генетичний алгоритм. Генетичний алгоритм дозволяє автоматично визначити оптимальні значення цих параметрів шляхом еволюції популяції розв'язків.

Процес використання генетичного алгоритму для підбору параметрів `step_size` і `gamma` може включати такі кроки:

Визначення початкової популяції розв'язків, де кожен розв'язок представляє певні значення `step_size` і `gamma`.

Оцінка якості кожного розв'язку на основі метрик навчання, таких як точність класифікації чи середня квадратична помилка.

Застосування операторів схрещування і мутації для створення нової популяції розв'язків.

Оцінка якості нової популяції і відбір кращих розв'язків.

Повторення кроків 3 і 4 протягом декількох поколінь, з поступовим поліпшенням якості розв'язків.

Вибір найкращого розв'язку як оптимальних значень `step_size` і `gamma`.

Таким чином, використання генетичного алгоритму для знаходження оптимальних параметрів `step_size` і `gamma` у алгоритмі StepLR допоможе покращити ефективність навчання моделі та досягти кращих результатів у задачі класифікації сигналів ЕКГ.

3.5 Реалізація генетичного алгоритму

3.5.1 Загальні відомості

Генетичний алгоритм (ГА) є еволюційним алгоритмом, який моделює процес природної еволюції для вирішення оптимізаційних задач. Він базується на ідеї природного відбору, схрещування, мутації та генетичної рекомбінації.

Основні функції генетичного алгоритму, які будуть реалізовані:

Популяція: Генетичний алгоритм працює зі збіркою індивідів, яку називають популяцією. Кожен індивід представляє потенційний розв'язок задачі і відповідає одній можливій конфігурації параметрів.

Функція пристосованості: Для оцінки якості кожного індивіда в популяції використовується функція пристосованості. Вона визначає, наскільки добре кожен індивід виконує поставлену задачу оптимізації.

Схрещування: Схрещування (кросинговер) відбувається шляхом комбінування генетичного матеріалу двох батьківських індивідів для створення нових потомків. Цей процес сприяє обміну корисною інформацією та появі нових комбінацій генетичних характеристик.

Мутація: Мутація полягає в випадкових змінах генетичного матеріалу в окремих індивідах. Це дозволяє зберегти різноманітність популяції та уникнути застою у пошуку оптимального розв'язку. Мутація сприяє введенню нових рішень і може допомогти виходити з локальних оптимумів.

Відбір: Відбір в генетичному алгоритмі визначає, які індивіди будуть збережені і продовжать еволюцію. Індивіди з більшою функцією пристосованості мають більші шанси бути відібраними, оскільки вони представляють кращі розв'язки.

Еволюція: Генетичний алгоритм проходить крізь ітеративні покоління, де кращі розв'язки з кожного покоління використовуються для створення нових поколінь. Процес еволюції триває до досягнення заданої умови зупинки, яка може бути максимальним числом поколінь або досягнення достатньо задовільного розв'язку.

Генетичні алгоритми широко використовуються в багатьох галузях, включаючи оптимізацію, машинне навчання, розробку програмного забезпечення, розкладання задач та багато інших. Вони є потужним інструментом для розв'язання складних оптимізаційних задач та дослідження простору параметрів. Застосування генетичного алгоритму в поєднанні з іншими алгоритмами та методами може дати ефективні та оптимальні розв'язки для широкого спектру задач.

3.5.2 Ініціалізація популяції

Для ініціалізації популяції в генетичному алгоритмі, кожен індивід створюється з двох словників, що містять параметри навчання мережі та параметри блоків згорткової мережі.

Перший словник містить параметри навчання, які визначають процес оптимізації та адаптації мережі під час навчання. Ці параметри включають розмір батчу - кількість прикладів даних, що використовуються за одну ітерацію навчання, швидкість навчання (learning rate) - коефіцієнт, який визначає, наскільки значущими є кроки оновлення ваг мережі, відкат кроку навчання (learning rate decay) - механізм, що дозволяє зменшувати швидкість навчання з часом, та кількість епох навчання - кількість повних проходів через навчальний набір даних.

Другий словник містить параметри блоків згорткової мережі, які визначають архітектуру та характеристики самої мережі. Ці параметри включають кількість каналів згортки для кожного шару - кількість фільтрів або каналів, що застосовуються на кожному шарі згортки, та розмір ядра згортки (kernel size) - розмір матриці ядра, що здійснює згортку з вхідними даними.

Кожен індивід в популяції генетичного алгоритму представлений кортежем, що складається з цих двох словників. Ця представлення дозволяє кожному індивіду мати унікальні параметри, які визначають його поведінку та характеристики під час навчання та еволюції. В процесі еволюції генетичного алгоритму, ці параметри будуть змінюватись та оптимізовуватись згідно з обраною функцією пристосованості.

3.5.3 Обчислення функції пристосованості

Після отримання параметрів мережі від кожного індивіда, виконується обчислення функції пристосованості для оцінки якості цих параметрів. Одним з популярних підходів є використання кросвалідації методом KFold з параметром $k=5$.

Кросвалідація KFold є методом оцінки моделі, який дозволяє розділити навчальний набір даних на k піднаборів або "складок". Процес кросвалідації повторюється k разів, при цьому на кожному кроці одна складка використовується як валідаційний набір, а решта складок використовується для тренування моделі.

Під час кожної ітерації модель навчається на тренувальних даних, що складаються з усіх складок, крім валідаційної складки. Потім модель оцінюється на валідаційному наборі даних, щоб отримати міру її пристосованості. Цей процес повторюється k разів, змінюючи валідаційну складку на кожній ітерації, щоб кожна складка була використана як валідаційний набір.

Результати оцінювання моделі на кожному валідаційному наборі об'єднуються для отримання остаточної оцінки пристосованості моделі. Це може бути середнє значення помилки або точності на кожній складці, або інша метрика оцінки моделі. Кросвалідація KFold дозволяє отримати більш об'єктивну та надійну оцінку моделі, оскільки вона враховує різноманітність даних та усереднює результати на різних наборах даних.

Цей підхід особливо корисний, коли навчальний набір даних обмежений або коли модель потенційно може страждати від перенавчання. Кросвалідація KFold дозволяє ефективно використовувати наявні дані та оцінювати пристосованість моделі на різних піднаборах даних, що забезпечує більш репрезентативну оцінку моделі перед застосуванням її на нових, невідомих даних.

Отримані оцінки пристосованості використовуються для прийняття рішень щодо відбору найкращих індивідів для наступного кроку генетичного алгоритму. Індивіди з вищими оцінками мають більшу ймовірність бути вибраними для схрещування та мутації, забезпечуючи ефективну експлуатацію корисної інформації та розвиток поколінь з кращими розв'язками.

Таким чином, обчислення функції пристосованості на основі кросвалідації дозволяє оцінити ефективність кожного індивіда в генетичному алгоритмі і спрямовує процес еволюції до досягнення оптимальних розв'язків. На рис. 3.3 схематично зображено K-Folds кросвалідацію, яка буде застосована для оцінки індивідів ГА, які складаються із значень параметрів «блоків згортки».

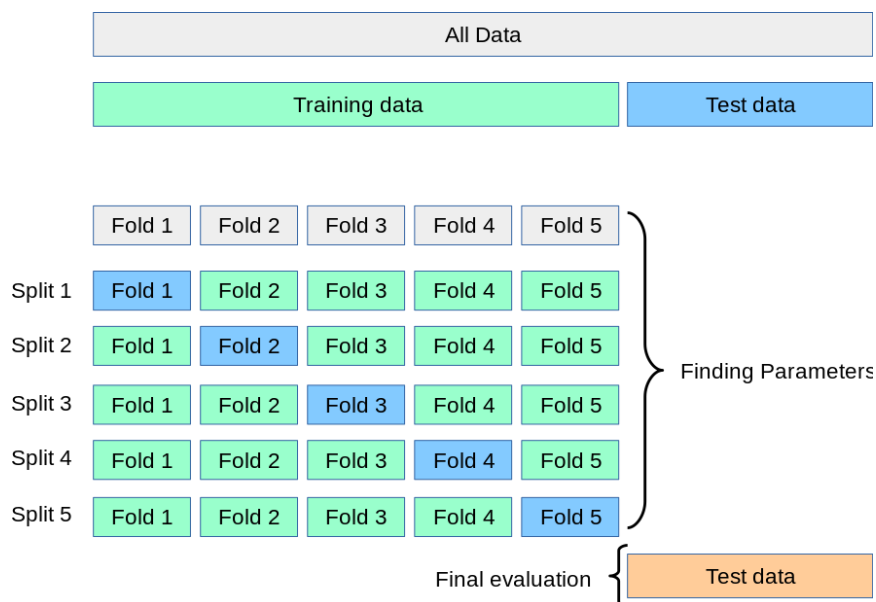


Рисунок 3.3 Схематичне зображення K-Folds крос-валідації

3.5.4 Відбір

Після сортування популяції за значеннями функції пристосованості, відбувається формування пар батьків для подальшого схрещування. Цей процес забезпечує передачу генетичного матеріалу від кращих індивідів до наступного покоління та сприяє комбінації корисних характеристик та пристосованості до змінюючого середовища.

У формуванні пар батьків використовується підхід "один з одним" (one-by-one). Починаючи з найкращих індивідів, пари батьків формуються шляхом об'єднання першого індивіда з другим, третього з четвертим, і так далі. Це означає, що кожен індивід має можливість бути батьком та спілкуватися генетично з іншими індивідами в популяції.

При формуванні пар батьків враховується ранжування індивідів за їх функцією пристосованості. Це означає, що сильніші розв'язки мають більшу

ймовірність бути обраними для схрещування. Такий підхід сприяє передачі корисної генетичної інформації від кращих індивідів до нащадків, що допомагає покращити пристосованість популяції до середовища та пошук оптимальних розв'язків.

Крім того, формування пар батьків є важливим кроком у генетичному алгоритмі, оскільки воно забезпечує різноманітність генетичного матеріалу у наступному поколінні. Це дозволяє зберегти цінну генетичну інформацію, а також відкриває шлях для пошуку нових комбінацій та експериментів, які можуть привести до знаходження ще кращих розв'язків.

Отже, формування пар батьків на основі ранжування індивідів за їх функцією пристосованості є ключовим етапом в генетичному алгоритмі, який сприяє ефективній еволюції популяції та пошуку оптимальних розв'язків шляхом комбінування кращих характеристик індивідів та адаптації до змінюючого середовища.

3.5.5 Елітні розв'язки

У генетичних алгоритмах елітні розв'язки відіграють важливу роль. Це набір найкращих індивідів, які зберігаються на кожній ітерації алгоритму. Елітні розв'язки відбираються на основі їхньої високої якості або пристосованості до поставленої задачі.

Переваги використання елітних розв'язків включають збереження найкращих характеристик, знайдених протягом процесу оптимізації. Це дозволяє забезпечити стабільність та прискорення збіжності алгоритму. Крім того, елітні розв'язки допомагають уникнути втрати корисної інформації, яка може бути затерта внаслідок мутацій або хромосомного перетворення.

В процесі роботи генетичного алгоритму елітні розв'язки зберігаються та заміщують менш пристосовані розв'язки в новому поколінні. Це сприяє збереженню та підвищенню якості розв'язків, що знаходяться в популяції.

Використання елітних розв'язків є важливою стратегією в генетичних алгоритмах, оскільки вони дозволяють зберегти та передати корисні

характеристики від одного покоління до іншого. Це сприяє досягненню більш оптимальних розв'язків та покращенню продуктивності алгоритму.

3.5.6 Кросинговер

Кросинговер (Crossover) є одним з ключових операторів генетичного алгоритму, в якому відбувається обмін генетичною інформацією між батьківськими особинами для створення нових потомків. В стратегії одноточкового кросинговеру, яку ми використовуємо в нашому дослідженні, випадковим чином вибирається одна точка розриву в геномі.

При виконанні кросинговеру гени між обраною точкою і кінцем геному обмінюються між батьківськими особинами. Цей процес дозволяє комбінувати різні характеристики батьків і створювати різноманітних потомків з новими генетичними комбінаціями. Використання стратегії одноточкового кросинговеру обґрунтоване його простотою реалізації та ефективністю при розв'язанні задачі класифікації ЕКГ.

Кросинговер вносить важливий елемент різноманітності до популяції, що допомагає уникнути преждевременної збіжності алгоритму до підоптимальних розв'язків. Шляхом комбінування генетичних властивостей батьківських особин, кросинговер допомагає здійснювати глобальний пошук у просторі розв'язків і знаходити більш оптимальні рішення.

Застосування одноточкового кросинговеру в нашому дослідженні з класифікації ЕКГ дозволяє враховувати різні генетичні комбінації, що можуть вплинути на якість та ефективність класифікаційної моделі. Цей оператор сприяє покращенню результатів та знаходженню оптимальних рішень у нашому дослідженні.

3.5.7 Мутація

Після проведення схрещування батьківських хромосом у генетичному алгоритмі, наступним кроком є мутація, що допомагає зберегти різноманітність популяції та уникнути застрягання в пошуку оптимального розв'язку.

У випадку нашої задачі оптимізації параметрів згорткової мережі та параметрів навчання, мутація виконується на рівні окремих дійсних величин у цих

параметрах. Кожна дійсна величина мутує шляхом додавання випадкового значення, отриманого з розподілу Гауса з параметрами $N(0, \text{val} * 0.25)$, де val є початковим значенням дійсної величини.

Цей підхід дозволяє вносити різноманітність в популяцію шляхом випадкового зсуву значень параметрів від їх початкових значень. Завдяки розподілу Гауса, мутація може бути помірною, що допомагає уникнути руйнування гарних рішень, які вже були знайдені.

Мутація є важливим елементом генетичного алгоритму, оскільки вона сприяє розширенню пошукового простору та виявленню нових, потенційно кращих розв'язків. Це дозволяє алгоритму уникнути застрягання в локальних оптимумах та продовжувати експлорувати різні регіони пошукового простору.

Мутація вносить стохастичні зміни в генетичний матеріал, що зберігає різноманітність у популяції та допомагає виявляти нові комбінації генетичних елементів, які можуть привести до покращення розв'язків. В комбінації з іншими операторами генетичного алгоритму, такими як схрещування, мутація допомагає забезпечити баланс між експлорацією пошукового простору та збереженням корисної інформації з попередніх поколінь.

3.5.8 Аналіз отриманих результатів генетичного алгоритму

Уточнення щодо груп популяції різних генетичних алгоритмів (ГА):

Маленький ГА – 16 індивідів на покоління, 20 поколінь, максимальна кількість основної частини мережі 4

Великий ГА – 16 індивідів на покоління, 30 поколінь, максимальна кількість основної частини мережі 5.

На рис. 3.4 і рис. 3.5 зображено середні значення функції пристосованості на кожне покоління малої і великої версії ГА відповідно

```
Small population mean generation fitness scores:
[0.26808125 0.2797875 0.29670625 0.29074375 0.29175 0.29614375
0.29771875 0.31174375 0.31115625 0.31079375 0.3156125 0.30425
0.30615625 0.31845625 0.3274625 0.32064375 0.31956875 0.32156875
0.32855 0.32829375]
```

Рисунок 3.4 Середні значення пристосованості на покоління малий ГА

Large population mean generation fitness scores:

```
[0.260525  0.26156875 0.27254375 0.28413125 0.2703      0.2772125
 0.27993125 0.28323125 0.27480625 0.289025   0.28936875 0.28778125
 0.2973625 0.29841875 0.29395625 0.29814375 0.2973      0.2958
 0.30155   0.29873125 0.29763125 0.3065625 0.3006      0.3076
 0.3088875 0.2994625  0.30704375 0.30575625 0.3067375 0.31008125]
```

Рисунок 3.5 Середні значення пристосованості на покоління великий ГА

На основі детального аналізу середніх значень обох популяцій можна зробити наступні спостереження. Генетичний алгоритм демонструє певну прогресію пристосованості моделі протягом декількох десятків поколінь. Зауважимо, що середня функція пристосованості на кожному поколінні має тенденцію до зростання.

Це свідчить про те, що генетичний алгоритм ефективно працює у відборі та комбінуванні генетичного матеріалу для покращення моделі. Кожне нове покоління виявляє деякі поліпшення у функції пристосованості порівняно з попереднім поколінням, що свідчить про поступове наближення до оптимальної моделі.

Це важливий результат, оскільки підтверджує, що генетичний алгоритм успішно еволюціонує модель, пристосовуючи її до поставленої задачі. Поступове зростання середньої функції пристосованості є індикатором того, що генетичний алгоритм знаходить кращі комбінації параметрів та архітектури моделі, що сприяє покращенню точності класифікації ЕКГ-сигналів. На рис. 3.6 і рис. 3.7 зображено зміну параметрів «блоків згортки» на поколіннях-чекпоінтах для малого і великого ГА відповідно.

```

first generations CNN model parameters #####
{'cnn_dims': [44, 122, 19], 'kernel_sizes': [4, 4, 2]}
{'cnn_dims': [106, 86, 118], 'kernel_sizes': [2, 6, 5]}
{'cnn_dims': [85], 'kernel_sizes': [2]}
{'cnn_dims': [106, 50, 101, 42], 'kernel_sizes': [2, 3, 6, 3]}
{'cnn_dims': [25, 88, 80, 122], 'kernel_sizes': [5, 3, 2, 6]}
10th generation CNN model parameters #####
{'cnn_dims': [30, 9], 'kernel_sizes': [10, 1]}
{'cnn_dims': [33, 83], 'kernel_sizes': [8, 1]}
{'cnn_dims': [69, 19], 'kernel_sizes': [1, 3]}
{'cnn_dims': [25, 40], 'kernel_sizes': [6, 1]}
{'cnn_dims': [25, 68, 112], 'kernel_sizes': [8, 1, 2]}
last generation CNN model parameters #####
{'cnn_dims': [12, 34], 'kernel_sizes': [4, 1]}
{'cnn_dims': [63, 60], 'kernel_sizes': [1, 1]}
{'cnn_dims': [41, 49], 'kernel_sizes': [7, 1]}
{'cnn_dims': [37, 57], 'kernel_sizes': [5, 1]}
{'cnn_dims': [24, 106], 'kernel_sizes': [3, 1]}

```

Рисунок 3.6 – Параметри «блоків згортки» моделей малого ГА

```

first generations CNN model parameters #####
{'cnn_dims': [72], 'kernel_sizes': [4]}
{'cnn_dims': [53, 58], 'kernel_sizes': [6, 6]}
{'cnn_dims': [33, 58, 52], 'kernel_sizes': [4, 4, 2]}
{'cnn_dims': [83, 71, 110, 77, 35], 'kernel_sizes': [2, 3, 4, 4, 4]}
{'cnn_dims': [26, 91], 'kernel_sizes': [2, 5]}
15th generation CNN model parameters #####
{'cnn_dims': [105, 53, 124], 'kernel_sizes': [4, 2, 11]}
{'cnn_dims': [176, 97, 65], 'kernel_sizes': [3, 3, 1]}
{'cnn_dims': [158, 120, 153], 'kernel_sizes': [5, 1, 3]}
{'cnn_dims': [156, 63, 122, 28], 'kernel_sizes': [4, 2, 1, 3]}
{'cnn_dims': [160, 91, 162], 'kernel_sizes': [6, 1, 1]}
last generation CNN model parameters #####
{'cnn_dims': [121, 78, 37], 'kernel_sizes': [11, 1, 1]}
{'cnn_dims': [253, 128, 18], 'kernel_sizes': [9, 1, 1]}
{'cnn_dims': [247, 186, 71], 'kernel_sizes': [12, 1, 1]}
{'cnn_dims': [159, 50, 122], 'kernel_sizes': [7, 1, 1]}
{'cnn_dims': [132, 87, 57], 'kernel_sizes': [7, 1, 1]}

```

Рисунок 3.7 – Параметри «блоків згортки» моделей великого ГА

На підставі аналізу параметрів Блоків згортки у обох популяціях можна зробити наступні спостереження. Розміри ядра згортки, які наближаються до $[n, 1, 1]$, свідчать про те, що основна частина інформації в ЕКГ-сигналах розташована вздовж осі часу, тобто в одному каналі. Це означає, що використання ядра з

розміром $[n, 1, 1]$ дозволяє ефективно відокремити цю інформацію від сигналів інших типів.

Щодо кількості каналів згортки, яка залишається достатньою на кожному шарі, можна зробити припущення, що генетичний алгоритм у даному випадку не старається занадто сильно спростити модель. Якщо б кількість каналів згортки була значно меншою, це можна було б інтерпретувати як спробу алгоритму зменшити складність моделі шляхом зведення кількості параметрів до мінімуму. Однак, таке припущення не підтверджується, і кількість каналів згортки залишається значною, що свідчить про те, що генетичний алгоритм зберігає потужність моделі для кращої адаптації до вхідних даних.

Після вибору найкращих параметрів архітектури моделі з обох версій генетичних алгоритмів ми переходимо до тренування. Під час тренування ми аналізуємо графіки значень похибки на кожній епосі навчання.

На графіку похибки видно дві криві: синя крива представляє значення похибки на тренувальних даних, а помаранчева крива відображає значення похибки на валідаційних даних. Використовуючи ці графіки, ми можемо оцінити ефективність моделі під час тренування.

Аналізуючи графік, ми шукаємо зменшення значення похибки з кожною епохою навчання. Оптимальним сценарієм є, коли значення похибки на тренувальних та валідаційних даних зменшується з кожною епохою. Це свідчить про те, що модель ефективно навчається та здатна узагальнювати отримані знання на нові дані. Графік значень похибки також допомагає визначити оптимальну кількість епох навчання моделі. Якщо ми спостерігаємо зменшення похибки на тренувальних та валідаційних даних з кожною епохою, але після певної точки похибка на валідаційних даних починає зростати, це може свідчити про перенавчання моделі. В такому випадку, оптимальним вибором буде зупинити тренування на попередній епосі, коли похибка на валідаційних даних була найнижчою.

Окрім графіків похибки, ми також можемо аналізувати графіки точності моделі під час тренування. Графік точності показує, як добре модель класифікує

дані на тренувальних та валідаційних даних з кожною епохою. Аналізуючи ці графіки, ми можемо побачити, як точність моделі змінюється протягом тренування. Оптимальним сценарієм є збільшення точності з кожною епохою на обох наборах даних.

Важливо враховувати, що графіки похибки та точності мають бути оцінені разом з іншими метриками, такими як час навчання та використання ресурсів. Іноді може бути необхідно знайти компроміс між точністю та часом навчання, особливо при роботі з великими обсягами даних.

Узагальнюючи, аналіз графіків значень похибки та точності на тренувальних та валідаційних даних є важливим кроком під час тренування ЗНМ для класифікації ЕКГ сигналів. Це дозволяє оцінити ефективність моделі, виявити перенавчання та визначити оптимальну кількість епох навчання. Адекватний аналіз цих графіків сприяє покращенню якості класифікації та роботи ЗНМ на нових ЕКГ даних. На рис. 3.8. і рис. 3.9 зображено криву навчання моделей з параметрами найкращих розв’язків із малого та великого ГА відповідно.

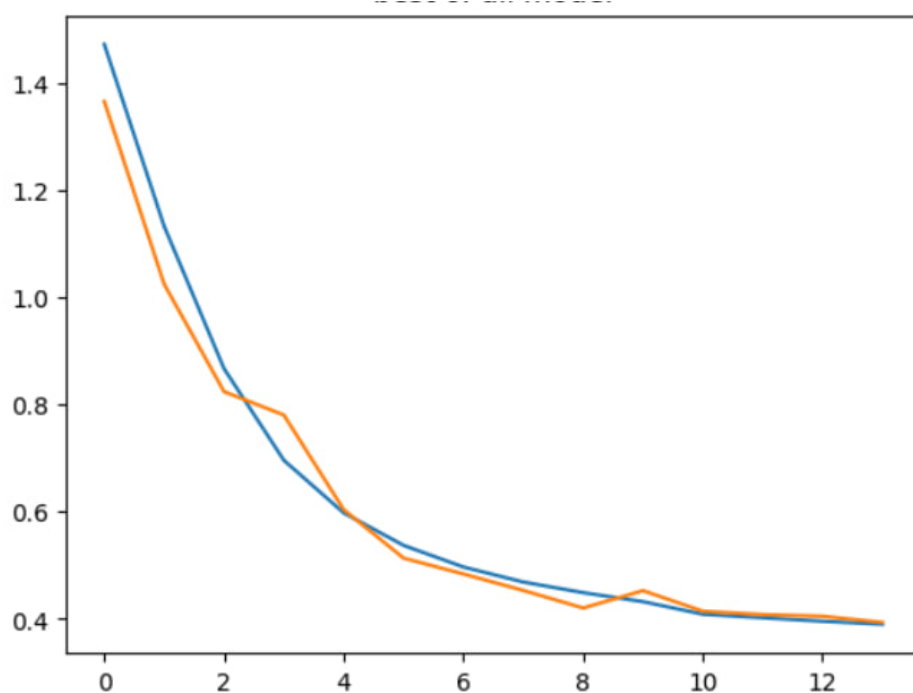


Рисунок 3.8 – Похибки найкращої моделі із малого ГА

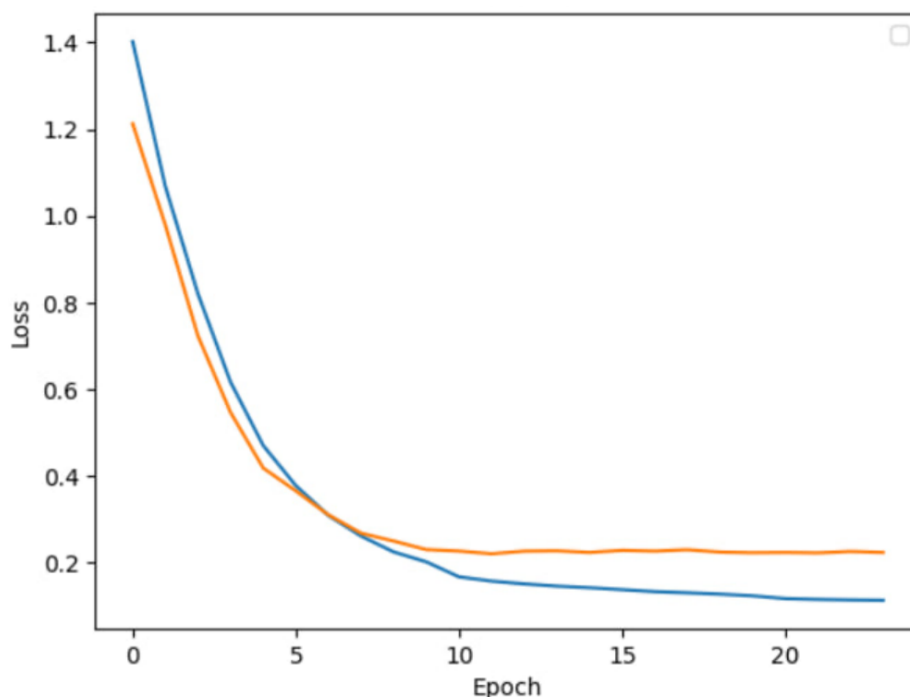


Рисунок 3.9 – Похибки найкращої моделі із великого ГА

Також варто зазначити, що при аналізі графіків похибки можна звернути увагу на швидкість збіжності моделей. Якщо модель швидко зменшує похибку та досягає стабільної похибки на тренувальних та валідаційних даних, це може свідчити про її ефективність та швидкість навчання. З іншого боку, повільне зменшення похибки може вказувати на складність завдання класифікації ЕКГ сигналів або потребу у більш складних моделях для досягнення кращих результатів.

Крім аналізу графіків, важливо також оцінити результати моделі на незалежному тестовому наборі даних. Це дозволить перевірити, наскільки добре модель узагальнює свої знання на нові дані та виявити можливі перенавчання або недостатню узагальнюючу здатність моделі.

Крім того, важливо зазначити, що генетичні алгоритми не є єдиними методами для побудови та тренування згорткових нейронних мереж. Існує багато інших підходів, таких як звичайне навчання з учителем або передбачення гіперпараметрів, які також можуть бути застосовані до класифікації ЕКГ сигналів. Кожен з цих підходів має свої переваги та обмеження, тому вибір методу залежить від конкретного завдання та обмежень даних.

Узагальнюючи, аналіз графіків похибки та інших метрик під час тренування згорткових нейронних мереж для класифікації ЕКГ сигналів є важливим етапом. Він дозволяє оцінити ефективність та узагальнюючу здатність моделі, вибрати оптимальні параметри та покращити результати класифікації. Додаткові дослідження та порівняння різних підходів можуть допомогти виявити найкращі методи для досягнення точної та стійкої класифікації ЕКГ сигналів. На рис. 3.10 і рис 3.11 зображено конфюжн матриці точності прогнозування найкращих моделей з малого та великого ГА відповідно

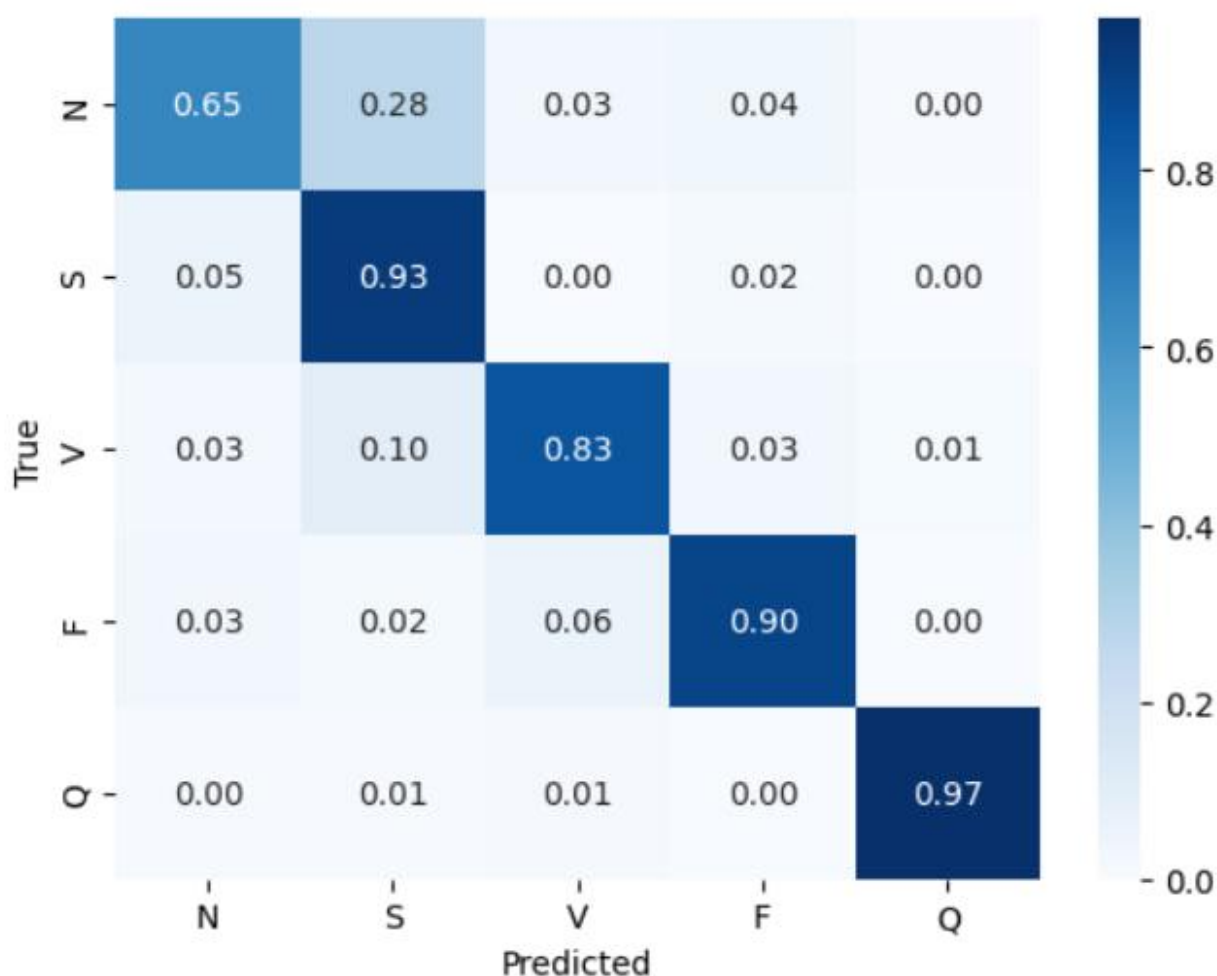


Рисунок 3.10 – Конфюжн матриця для найкращої моделі з малого ГА

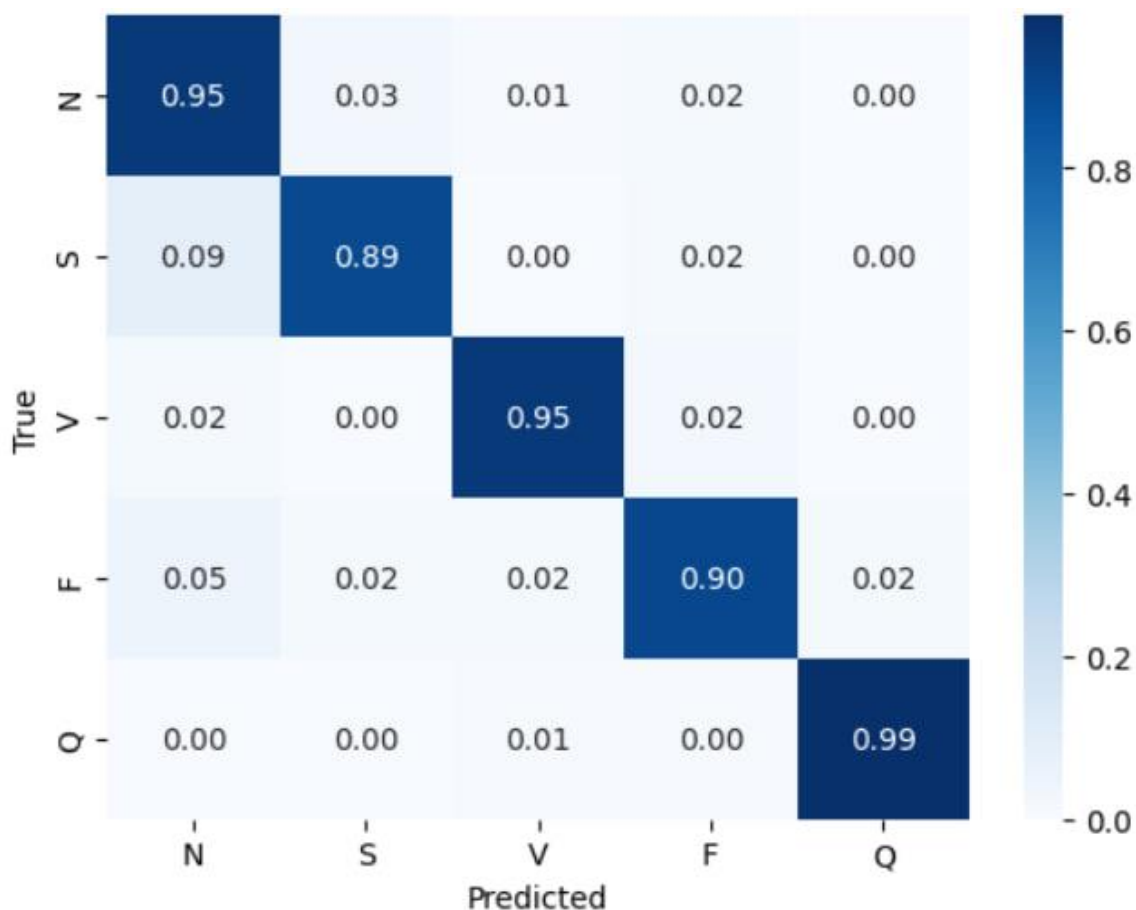


Рисунок 3.11– Кофюжн матриця для найкращої моделі з великого ГА

З аналізу матриць помилок можна зробити висновок, що найкраща модель, отримана за допомогою малого генетичного алгоритму, має певні труднощі у класифікації між "Нормальним кардіоциклом" і "Аритмією передсердя". Хоча ця модель і помиляється, класифікуючи здорового пацієнта як нездорового, це може бути менш серйозною проблемою, оскільки недіагнововані аритмії можуть викликати більше занепокоєння.

Натомість, найкраща модель, отримана за допомогою великого генетичного алгоритму, ефективно класифікує всі 5 класів ЕКГ-сигналів. Це є дуже позитивним результатом як для пацієнтів, так і для медичного персоналу, оскільки така модель може значно спростити процес діагностики та надати цінну допомогу лікарям.

У підсумку, хоча модель, отримана за допомогою малого генетичного алгоритму, має незначні труднощі у класифікації деяких класів ЕКГ-сигналів, найкраща модель, отримана за допомогою великого генетичного алгоритму, продемонструвала високу ефективність у класифікації всіх класів. Це важливий

внесок у поліпшення діагностики та лікування серцевих захворювань, який сприяє як пацієнтам, так і медичному персоналу.

3.6 Розрахунок економічного ефекту за темою дипломної роботи

Метою даного розділу було визначення який із варіантів розробки мого програмного продукту є самим оптимальним по таким критеріям, як:

1. Простота написання коду
2. Зручне середовище виконання та написання коду
3. Точність та швидкість обробки результатів опитування

Після визначення та фіксації основних функцій програмного продукту, була створена морфологічна карта, яка наведена нижче (рис. 3.12):

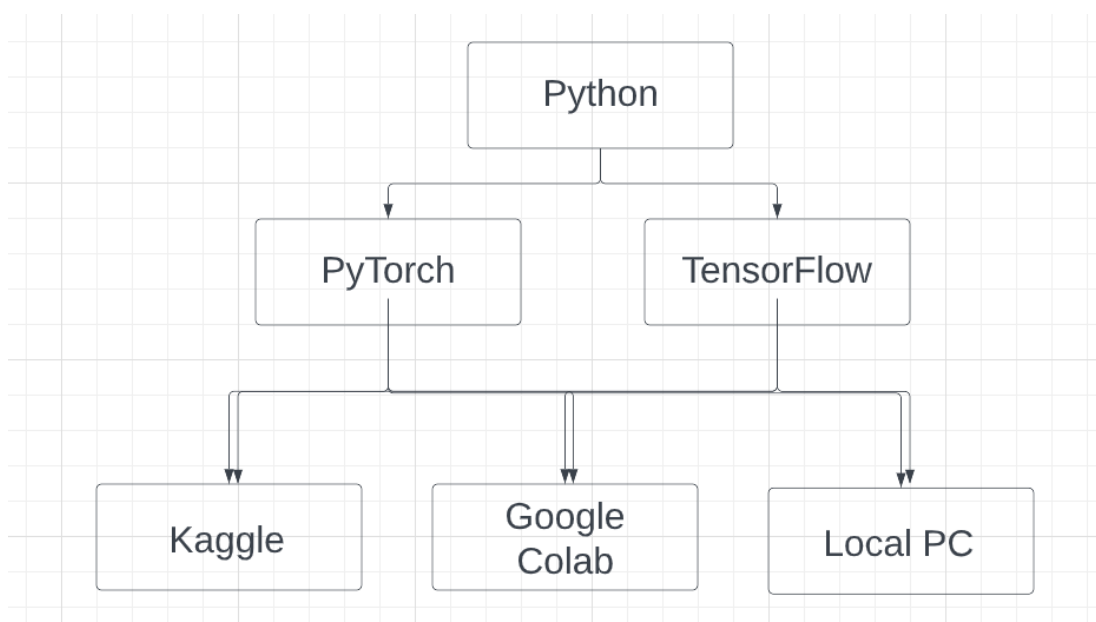


Рисунок 3.12 – Морфологічна карта

Також проводилося техніко-економічне обґрунтування розробки методом функціонально-вартісного аналізу, методом побудови позитивно-негативної матриці та визначенням вартості розробки даного програмного продукту.

На основі усіх проведених розрахунків та даних про функціональність продукту було визначено, що найбільш ефективним став варіант, вартість якого складає 774102.55 грн. Цей варіант включає у себе мову програмування – Python,

PyTorch – бібліотека для машинного/глибокого навчання, Kaggle – платформа для навчання моделей

Висновки до розділу 3

В даному дослідженні було вивчено і порівняно два види генетичних алгоритмів - малий та великий, для підбору оптимальних параметрів архітектури нейронної мережі для класифікації ЕКГ-сигналів. Застосування генетичних алгоритмів дозволило автоматично підібрати найкращі параметри моделі, що веде до покращення її ефективності.

З результатів експериментів можна зробити наступні висновки:

Використання малого генетичного алгоритму призвело до отримання моделі з незначними труднощами у класифікації деяких класів ЕКГ-сигналів, зокрема між "Нормальним кардіоциклом" і "Аритмією передсердя". Помилкова класифікація здорових пацієнтів може бути менш серйозною проблемою, однак варто розглядати можливості подальшого вдосконалення моделі для покращення точності класифікації цих класів.

Використання великого генетичного алгоритму призвело до отримання моделі, яка ефективно класифікує всі 5 класів ЕКГ-сигналів. Це є дуже обнадійливим результатом і може значно полегшити процес діагностики та надати цінну підтримку медичному персоналу.

Використання генетичних алгоритмів для підбору параметрів архітектури нейронної мережі демонструє їх ефективність у великому просторі пошуку. Цей підхід дозволяє автоматично знаходити оптимальні параметри, що збільшує швидкість та точність процесу навчання моделі.

Загалом, результати дослідження свідчать про успішність використання генетичних алгоритмів для підбору параметрів архітектури нейронної мережі у задачі класифікації ЕКГ-сигналів. Порівняння моделей, отриманих за допомогою малого та великого генетичних алгоритмів, показало, що використання великого

алгоритму призводить до отримання найкращої моделі з високою точністю класифікації всіх 5 класів ЕКГ-сигналів.

Отримані результати можуть бути корисними як для пацієнтів, так і для медичного персоналу. Впевнена класифікація сигналів може допомогти вчасно виявити аномалії в роботі серця та надати необхідну медичну допомогу. Крім того, автоматичний підбір оптимальних параметрів моделі спрощує процес розробки та налаштування нейронної мережі, що є важливим фактором для прискорення та поліпшення класифікації ЕКГ-сигналів.

Наступні кроки можуть включати подальше вдосконалення моделі та експериментування з різними архітектурами та генетичними алгоритмами для досягнення ще кращих результатів. Також можливим напрямком досліджень є розширення обсягу датасету та розгляд інших медичних аналізів, що можуть доповнити та покращити класифікацію ЕКГ-сигналів.

ЗАГАЛЬНІ ВИСНОВКИ

Результати роботи

Отримані результати дослідження класифікації ЕКГ-сигналів підтверджують високу ефективність застосування згорткових нейромереж у цій області. Шляхом комбінації згорткових шарів із методом адаптивної середньої згортки, отримано високу точність у визначенні типів аритмій. Застосування згорткових шарів дозволяє виявити кореляцію між різними частотними складовими ЕКГ-сигналу, що сприяє точнішій класифікації.

Крім того, нейромережевий підхід, оптимізований за допомогою генетичного алгоритму, виявився важливим кроком у поліпшенні результатів класифікації. Генетичний алгоритм дозволяє ефективно відібрати оптимальні параметри мережі, що призводить до зниження помилок класифікації та забезпечує стійкість до змін вхідних даних. Цей підхід забезпечує гнучкість та адаптивність моделі, що вкрай важливо при аналізі ЕКГ-сигналів, які можуть варіюватися у своїй часовій та частотній структурі.

Отже, проведене дослідження відкриває нові можливості у сфері класифікації ЕКГ-сигналів і підтверджує значний вплив згорткових нейромереж та генетичних алгоритмів на покращення результатів. Ці результати створюють підґрунтя для подальших досліджень та розвитку алгоритмів, спрямованих на вдосконалення діагностики та лікування серцевих захворювань, що базуються на аналізі ЕКГ-сигналів.

Обговорення результатів роботи

Отримані результати дослідження розкривають ефективність використання згорткових нейромереж у задачі класифікації ЕКГ-сигналів. Застосування адаптивної середньої згортки у поєднанні з цими нейромережами дозволяє зберігати сталу розмірність мапи ознак, незалежно від розміру вхідних даних. Це допомагає досягти однакового розподілу інформації та поліпшує роботу класифікатора, що забезпечує більш точну класифікацію.

Помітним внеском у покращення результатів класифікації є застосування генетичного алгоритму для оптимізації параметрів мережі. Генетичний алгоритм дозволяє знаходити оптимальні значення параметрів, що сприяє зниженню помилок класифікації та покращенню загальної точності моделі. Цей підхід дозволяє систематично вдосконалювати архітектуру мережі та адаптувати її до конкретних вхідних даних, що є критичним для досягнення високих показників ефективності класифікатора.

Отже, отримані результати свідчать про потужний потенціал згорткових нейромереж та генетичних алгоритмів у вирішенні задачі класифікації ЕКГ-сигналів. Ці методи дозволяють забезпечити високу точність класифікації, зберігати сталу розмірність мапи ознак та ефективно оптимізувати параметри мережі. Враховуючи значущість точної класифікації ЕКГ-сигналів у діагностиці серцевих захворювань, ці результати відкривають перспективи для подальшого розвитку та вдосконалення методів аналізу та класифікації цих сигналів з метою покращення медичної практики.

Висновки

Результати дослідження наочно демонструють, що поєднання згорткових нейромереж, адаптивної середньої згортки та генетичного алгоритму є високоефективним підходом у задачі класифікації ЕКГ-сигналів. Цей комплексний метод забезпечує високу точність та надійність у визначенні різних типів аритмій, що є критично важливим для успішного лікування та моніторингу серцевих захворювань.

Отримані результати не тільки відкривають нові перспективи для розуміння фізіології серця, але й мають значимі практичні застосування. Висока точність класифікації аритмій, досягнута за допомогою згорткових нейромереж та адаптивної середньої згортки, дозволяє забезпечити вчасну та точну діагностику серцевих порушень. Це може привести до раннього виявлення та надання ефективного лікування, покращуючи прогнози та якість життя пацієнтів.

Оптимізація параметрів мережі за допомогою генетичного алгоритму також є важливим етапом в розвитку даного підходу. Цей алгоритм дозволяє

систематично підбирати оптимальні значення параметрів, що забезпечує покращення якості класифікації та забезпечує стійкість до змін вхідних даних. В результаті, системи класифікації ЕКГ-сигналів, засновані на згорткових нейромережах та генетичному алгоритмі, можуть працювати надійно та ефективно навіть при змінних умовах досліджень.

Висновки цього дослідження вказують на потенціал застосування передових методів машинного навчання та оптимізації для поліпшення клінічної діагностики та лікування серцевих захворювань. В подальшому, дослідники можуть експериментувати з іншими архітектурами нейромереж, алгоритмами оптимізації та вхідними даними для ще більш точної та надійної класифікації ЕКГ-сигналів. Застосування цих розширених методів може привести до збільшення швидкості та ефективності діагностики, врятувати багато життів та покращити добробут людей з серцевими проблемами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Scikit-learn Documentation. [Електронний ресурс] – Режим доступу до ресурсу: <https://scikit-learn.org/>
2. Narkhede S. Understanding AUC - ROC Curve - Towards Data Science. *Towards Data Science*. [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>
3. PyTorch [Електронний ресурс] – Режим доступу до ресурсу: <https://pytorch.org/>
4. Genetic algorithm for neural network architecture optimization [Електронний ресурс] // IEEE. – 2016. – Режим доступу до ресурсу: <http://ieeexplore.ieee.org/document/7731699/>
5. Ioffe S. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Електронний ресурс] / S. Ioffe, C. Szegedy. – 2015. – Режим доступу до ресурсу: <https://doi.org/10.48550/arXiv.1502.03167>.
6. Goodfellow. Deep learning / Goodfellow, Bengio, Courville // Genetic Programming and Evolvable Machines / Goodfellow, Bengio, Courville., 2018. – С. 305–307
7. Adam: A method for stochastic optimization [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <http://arxiv.org/abs/1412.6980>
8. Gaussian Error Linear Units (GELUs) [Електронний ресурс] // arXiv. – 2016. – Режим доступу до ресурсу: <http://arxiv.org/abs/1606.08415>
9. PyTorch Recipes [Електронний ресурс] – Режим доступу до ресурсу: <http://link.springer.com/10.1007/978-1-4842-4258-2>.
10. Imambi. EAI/Springer Innovations in Communication and Computing / Imambi, Prakash, Kanagachidambaresan // PyTorch / Imambi, Prakash, Kanagachidambaresan., 2021. – (25228609). – С. 87–104
11. Genetic Algorithms [Електронний ресурс] // Springer. – 2023. – Режим доступу до ресурсу: https://link.springer.com/10.1007/978-3-031-17922-8_3.
12. ECG Heartbeat Classification Using Convolutional Neural Networks [Електронний ресурс] // IEEE Access. – 2020. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/8952723/>

13. Convolutional Neural Network for Heartbeat Classification [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://ieeexplore.ieee.org/document/9679581/>
14. Eiben. Introduction To Evolutionary Computing / Eiben, Smith., 2003. – 300 с. – (Natural Computing Series).
15. Practical Handbook of Genetic Algorithms [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: <https://www.taylorfrancis.com/books/9781420050080>
16. Genetic Algorithms, Noise, and the Sizing of Populations [Электронный ресурс] // Complex Systems. – 1991. – Режим доступа до ресурсу: https://www.complex-systems.com/abstracts/v06_i04_a03/
17. Handbook of genetic algorithms [Электронный ресурс] // Elsevier. – 1998. – Режим доступа до ресурсу: <https://linkinghub.elsevier.com/retrieve/pii/S0004370298000162>
18. Handbook of Evolutionary Computation, 1997.
19. Eiben. Special Forms of Evolution [Электронный ресурс] / Eiben, Smith // Springer. – 2003. – Режим доступа до ресурсу: http://link.springer.com/10.1007/978-3-662-05094-1_13
20. An Automatic Diagnosis of Arrhythmias Using a Combination of CNN and LSTM Technology [Электронный ресурс] / Zheng, Chen, Hu, Zhu. – 2020. – Режим доступа до ресурсу: <https://www.mdpi.com/2079-9292/9/1/121>
21. Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats [Электронный ресурс] // Computers in Biology and Medicine. – 2018. – Режим доступа до ресурсу: <https://linkinghub.elsevier.com/retrieve/pii/S0010482518301446>
22. Kiranyaz. Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks [Электронный ресурс] / Kiranyaz, Ince, Gabbouj // IEEE Transactions on Biomedical Engineering. – 2016. – Режим доступа до ресурсу: <http://ieeexplore.ieee.org/document/7202837/>
23. Multi-Lead ECG Classification via an Information-Based Attention Convolutional Neural Network [Электронный ресурс] // Journal of Shanghai Jiaotong University

- (Science). – 2022. – Режим доступа до ресурсу:
<https://link.springer.com/10.1007/s12204-021-2371-8>
24. HeartNet: Self Multihead Attention Mechanism via Convolutional Network With Adversarial Data Synthesis for ECG-Based Arrhythmia Classification [Электронный ресурс] // IEEE Access. – 2022. – Режим доступа до ресурсу:
<https://ieeexplore.ieee.org/document/9889702/>
 25. Automatic prediction of atrial fibrillation based on convolutional neural network using a short-term normal electrocardiogram signal [Электронный ресурс] // Journal of Korean Medical Science. – 2019. – Режим доступа до ресурсу:
<https://jkms.org/DOIx.php?id=10.3346/jkms.2019.34.e64>.
 26. Detection of atrial fibrillation using 1D convolutional neural network [Электронный ресурс] // Sensors. – 2020. – Режим доступа до ресурсу:
<https://www.mdpi.com/1424-8220/20/7/2136>
 27. 1D convolutional neural networks and applications: A survey [Электронный ресурс] // Mechanical Systems and Signal Processing. – 2021. – Режим доступа до ресурсу:
<https://linkinghub.elsevier.com/retrieve/pii/S0888327020307846>.
 28. Saxena A. An Introduction to Convolutional Neural Networks [Электронный ресурс] / Aarush Saxena // International Journal for Research in Applied Science and Engineering Technology. – 2022. – Режим доступа до ресурсу: <https://www.ijraset.com/best-journal/convolution-neural-network>
 29. Fatyanosa. An Automatic Convolutional Neural Network Optimization Using a Diversity-Guided Genetic Algorithm [Электронный ресурс] / Fatyanosa, Aritsugi // IEEE Access. – 2021. – Режим доступа до ресурсу:
<https://ieeexplore.ieee.org/document/9462900/>.