

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра прикладної математики
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Класифікація медичних часових рядів за допомогою методів
машинного навчання
(тема)

Виконав:

студент 2 курсу, групи ПМм-19-1
Гайова А.Ю.
(прізвище, ініціали)

Спеціальність 113 Прикладна математика
(код і повна назва спеціальності)

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

Освітня програма Прикладна математика
(повна назва освітньої програми)

Керівник проф. Кіріченко Л.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ

Тевяшев А.Д.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту

Кафедра прикладної математики

Рівень вищої освіти другий (магістерський)

Спеціальність 113 Прикладна математика

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Прикладна математика

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____

(підпис)

“ ____ ” _____ 2020 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Гайовій Анастасії Юріївні

(прізвище, ім'я, по батькові)

1. Тема роботи Класифікація медичних часових рядів за допомогою методів машинного навчання

затверджена наказом по університету від 23 жовтня 2020 р. № 1422 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 грудня 2020 р.

3. Вихідні дані до роботи результати електрокардіограм пацієнтів

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Системний аналіз проблеми класифікації медичних часових рядів

2. Вибір і обґрунтування методу розв'язання

3. Програмна реалізація

4. Результати обчислювального експерименту

5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій _____

1. Актуальність теми роботи _____

2. Постановка задачі _____

3. Системний аналіз проблеми _____

4. Метод чисельного аналізу _____

5. Результати обчислювального експерименту _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	вересень 2020 р.	Виконано
2	Вибір та обґрунтування методу	жовтень – листопад 2020 р.	Виконано
3	Розробка алгоритму і програми	листопад – грудень 2020 р.	Виконано
4	Проведення аналітичних досліджень та розрахунків	листопад – грудень 2020 р.	Виконано
5	Робота над текстом пояснювальної записки	грудень 2020 р.	Виконано
6	Представлення роботи на рецензію в ЕК	грудень 2020 р.	Виконано

Дата видачі завдання 1 вересня 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Кіріченко Л.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 67 с., 8 табл., 30 рис., 1 дод., 11 джерел.

ГЛИБИННЕ НАВЧАННЯ, ЕЛЕКТРОКАРДІОГРАМА (ЕКГ), ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, КЕРАС, КЛАСИФІКАЦІЯ СЕРЦЕВОГО РИТМУ.

Об'єкт дослідження – зміни електрокардіограм.

Мета роботи – класифікувати медичні часові ряди на прикладі електрокардіограм.

Метод дослідження – згорткові нейронні мережі.

У атестаційній роботі представлено використання згорткових нейронних мереж. Описано три архітектури які використовують різну кількість шарів у мережі. Відома база даних аритмії Інституту Массачусетського технологічного інституту (MIT-BIH) використовувалася для підготовки та тестування наших класифікаційних моделей, що, в свою чергу, дозволяє порівняти наші результати з сучасними методами.

ABSTRACT

Introductory note: 67 pages, 8 tables, 30 figures, 1 appendixes, 11 sources.

CLASSIFICATION OF HEART RATE, CONVOLUTION NEURAL NETWORKS, DEEP LEARNING, ELECTROCARDIOGRAPH (ECG), KERAS.

Object of research – changes in electrocardiograms.

Purpose of work – to classify medical time series on the example of an electrocardiogram.

Methods of research – convolutional neural networks.

The use of convolutional neural networks is presented in the attestation work. It describes three architectures that use different numbers of layers in a network. The well-known arrhythmia database of the Massachusetts Institute of Technology (MIT-BIH) was used to prepare and test our classification models, which in turn allows us to compare our results with participatory methods.

ЗМІСТ

	С.
Вступ	8
1 Системний аналіз проблеми класифікації часових рядів та постановка задач дослідження	9
1.1 Системний аналіз проблеми класифікації медичних часових рядів.....	9
1.1.1 Вербальна модель системи	9
1.1.2 Морфологічний опис системи	9
1.1.3 Функціональна модель системи	12
1.1.4 Інформаційна модель системи	14
1.2 Аналіз сценаріїв вирішення проблеми вибору методу класифікації	14
1.2.1 Модель аналізу проблеми	14
1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу ієрархій	15
1.3 Змістовна та формальна постановка задачі	19
1.3.1 Змістовна постановка задачі	19
1.3.2 Формальна постановка задачі	23
2 Вибір та обґрунтування методу розв'язання	25
2.1 Основні архітектури нейронних мереж	25
2.2 Технології машинного навчання для прогнозування на ЕКГ	28
2.3 Згорткові нейронні мережі для класифікації ЕКГ-сигналів	28
2.4 Методи, спеціально розроблені для CNN.....	36
2.5 Запропоновані моделі	40
3 Програмна реалізація	42
3.1 Python як мова машинного навчання	42
3.2 Опис програми.....	43
4 Результати обчислювального експерименту	45
5 Аналіз можливих застосувань	52
Висновки	53

Перелік джерел посилання	54
Додаток А Програма	55

ВСТУП

Електрокардіограми (ЕКГ) широко використовуються для відображення захворювань серця, обстеження їхніх схем виявилось корисним для полегшення діагностики серцевих захворювань. На відміну від інших біометричних способів, як відбитки пальців, обличчя, райдужки та підпис, ЕКГ є безперервними часовими рядами, чий аналіз може бути корисним, якщо доступні записи ЕКГ протягом достатнього часу. Фактично, використання комп'ютерних методів автоматичної обробки сигналів ЕКГ може принести користь кардіологам у діагностиці. За останнє десятиліття багато методів було запропоновано для виявлення аритмії. Деякі з цих підходів засновані на методах цифрової обробки сигналів та алгоритмах класифікації, такі як метод опорних векторів. Одна із найуспішніших архітектур глибокого навчання – це згорткові нейронні мережі. У даній атестаційній роботі досліджується потенціал ЗНМ у виявленні та класифікації аритмій за допомогою сигналів ЕКГ.

1 СИСТЕМНИЙ АНАЛІЗ ПРОБЛЕМИ КЛАСИФІКАЦІЇ СЕРЦЕВИХ РИТМІВ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Системний аналіз проблеми класифікації серцевих ритмів

1.1.1 Вербальна модель системи

Об'єкт аналізу – «Часові ряди на прикладі електрокардіограми».

Предмет аналізу – «Виявлення захворювань за допомогою ЕКГ».

Точка зору: дослідник.

Ціль: класифікація серцевих ритмів.

Головний вихід системи – перелік властивостей часового ряду який був поданий на вхід.

В дипломній роботі розглядаються часові ряди на прикладі електрокардіограм. Метою є виявлення основних властивостей і відмінностей таких процесів. Для рутинного аналізу електричної активності серця, як правило, використовують ЕКГ, зареєстровану з 12 окремих провідників. Кожен удар серця містить серію відхилень від базової лінії на ЕКГ, або хвилі, що відображають еволюцію часу електричної активності в серці.

1.1.2 Морфологічний опис системи

Досліджувана система – задача класифікації серцевих ритмів за допомогою електрокардіограм.

Призначенням моделі є її подальше використання задля полегшення визначення стану серцево-судинної системи людини та також для виконання подальших наукових розробок у цій сфері.

Для реалізації мети системи використовуються різні ресурси: інформаційні, технологічні.

Морфологічний опис системи почнемо розглядати з опису зовнішнього середовища, який наведено на рисунку 1.1.

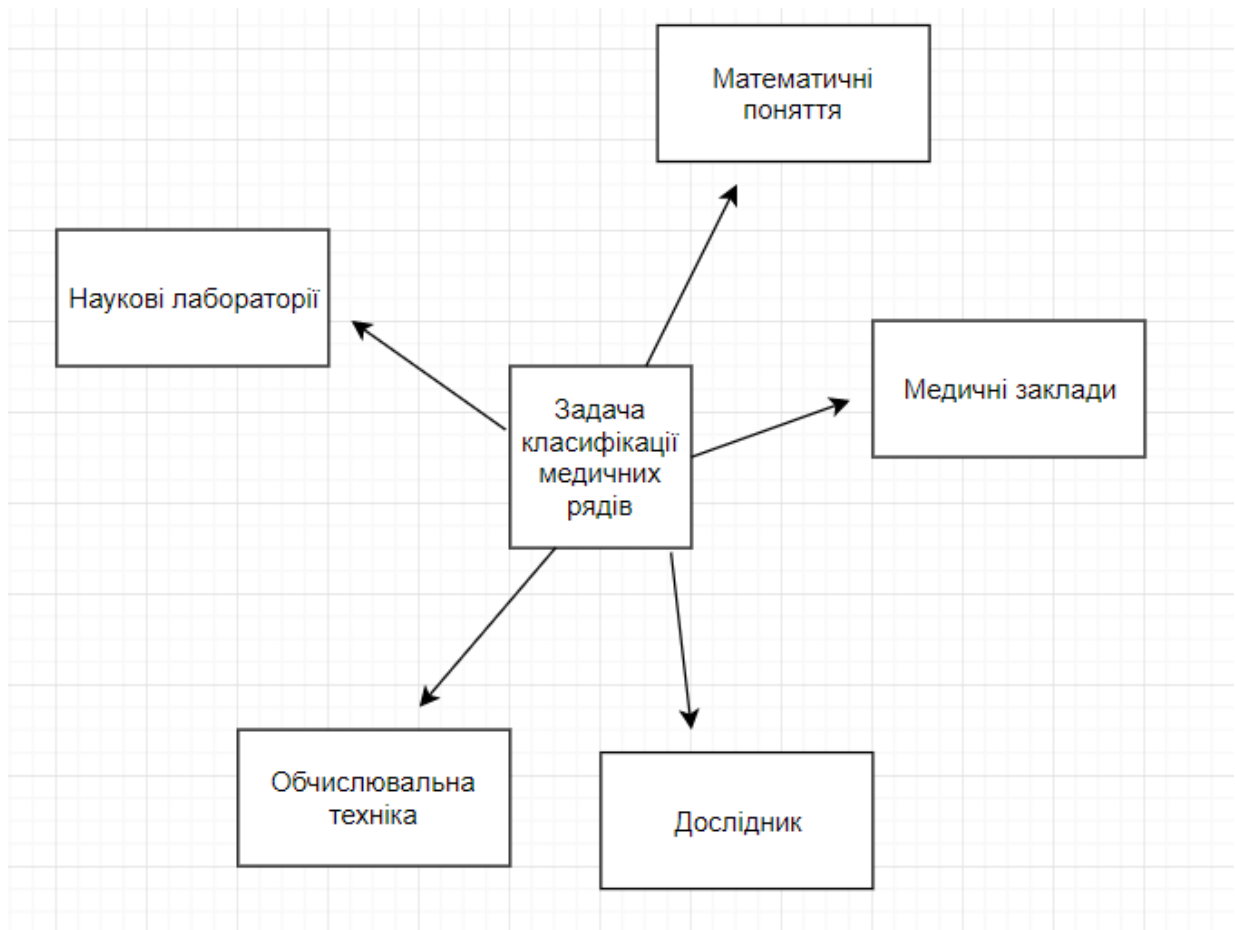


Рисунок 1.1 – Зовнішнє середовище

Зовнішнє середовище – сукупність всіх об'єктів поза межі системи, зміна властивостей яких впливають на систему, а також тих об'єктів, чії властивості змінюються в результаті поведінки системи.

Об'єкти зовнішнього середовища:

- а) наукові лабораторії забезпечують умови для проведення досліджень;
- б) обчислювальна техніка є засобом для отримання практичних результатів;
- в) дослідник безпосередньо проводить експерименти, обирає методи і способи класифікації часових рядів;
- г) на основі різних математичних понять таких як вейвлети, статистики

вищого порядку та інших, створення математичної моделі;

д) медичні заклади використовують результати досліджень для покращення своїх розробок.

Модель типу «чорний ящик» представлена на рисунку 1.2.

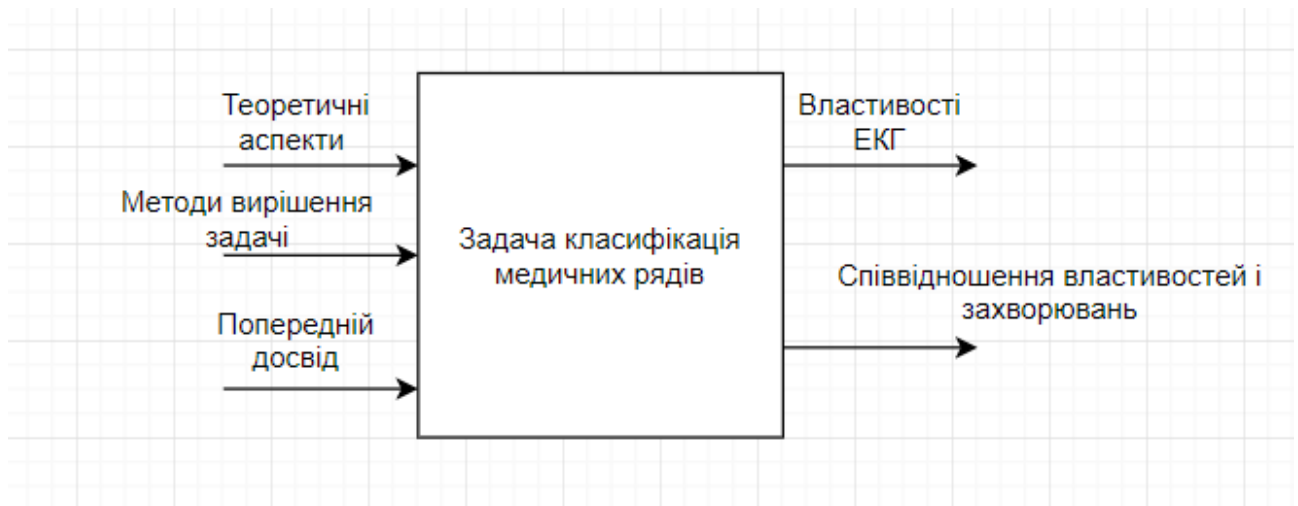


Рисунок 1.2 – Модель «чорний ящик»

Така модель є вихідною при побудові моделі складної системи, акцентує увагу дослідника на взаємодії системи з зовнішнім середовищем. Тут виходами системи є її цільові продукти. А входи – це вплив середовища на систему. Зміст «чорного ящика» не розкривається, тому що увага звертається тільки на кордон системи. Кордон, в свою чергу, підкреслює цілісність системи, відокремленість її від зовнішнього середовища і взаємодія системи і середовища.

У моделі типу «білий ящик» описаний «склад системи», вказані взаємозв'язки між елементами системи. Модель представлена на рисунку 1.3.

Дана система А складається з двох підсистем: А1 і А2:

1) А1 – теоретична база, що складається з:

- а) А11 дослідника та його знань;
- б) А12 результати попередніх досліджень ;

2) А2 – обчислювальна база, що складається з:

- а) А21 методи виконання завдання;
- б) А22 обслуговування: ЕОМ.

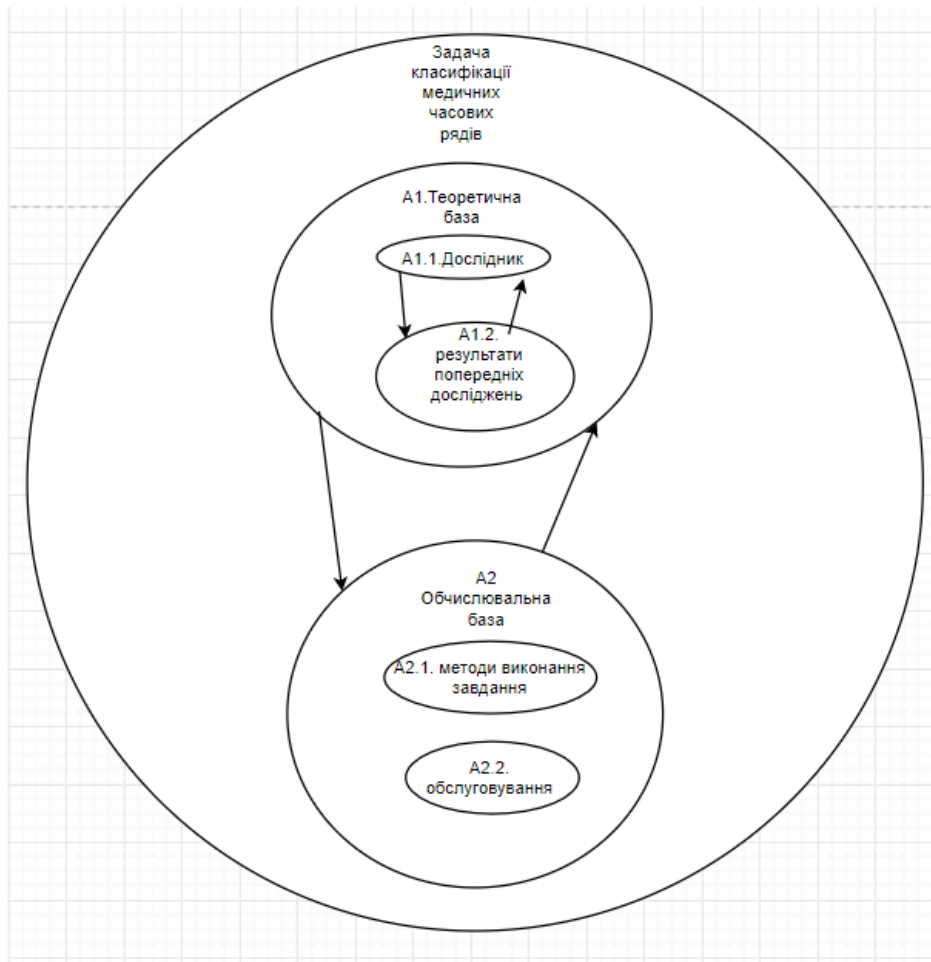


Рисунок 1.3 – Модель «білий ящик»

1.1.3 Функціональна модель системи

Для нових систем застосування IDEF0 має за ціль визначення вимог та функцій для подальшої розробки системи, що відповідає вимогам та реалізує виділені функції. Стосовно існуючих систем IDEF0 може бути застосована для аналізу функцій, які виконує система, та відображення механізмів, за допомогою яких ці функції виконуються. В рамках методології IDEF0 процес представляється у вигляді набору елементів, які взаємодіють між собою, а також показуються ресурси, що споживаються кожною роботою.

Перша діаграма в ієрархії діаграм IDEF0 являє собою саме загальний опис системи та її взаємодії із зовнішнім середовищем. Такі діаграми називають контекстними. Контекстна діаграма зображує функціонування системи в

цілому. Після опису системи в цілому проводиться декомпозицією її на великі фрагменти. Після декомпозиції контекстної діаграми проводиться декомпозиція кожного великого фрагмента системи на більш дрібні і так далі, до досягнення потрібного рівня деталізації опису. Діаграма IDEF0 представлена на рис. 1.4.

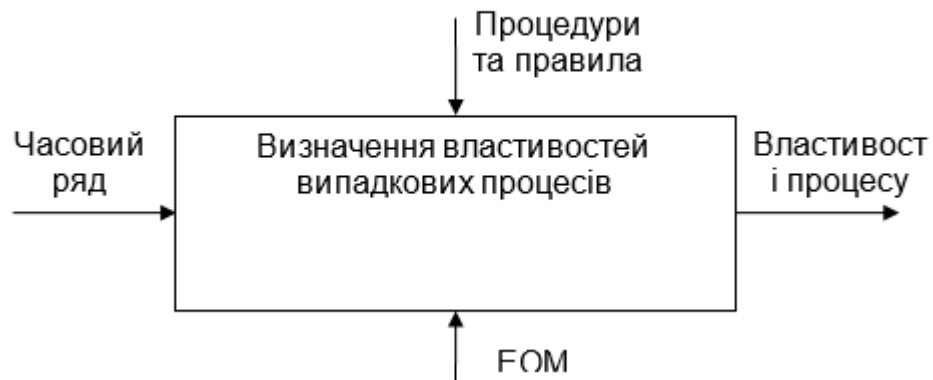


Рисунок 1.4 – Діаграма IDEF0

IDEF3 – це метод, який має основною метою дати можливість аналітикам описати ситуацію, коли процеси виконуються в певній послідовності, а також описати об'єкти, які беруть участь спільно в одному процесі. IDEF3 широко застосовується при розробці інформаційних систем. При цьому використовується інструмент візуального моделювання бізнес-процесів. Діаграма IDEF3 представлена на рис. 1.5.

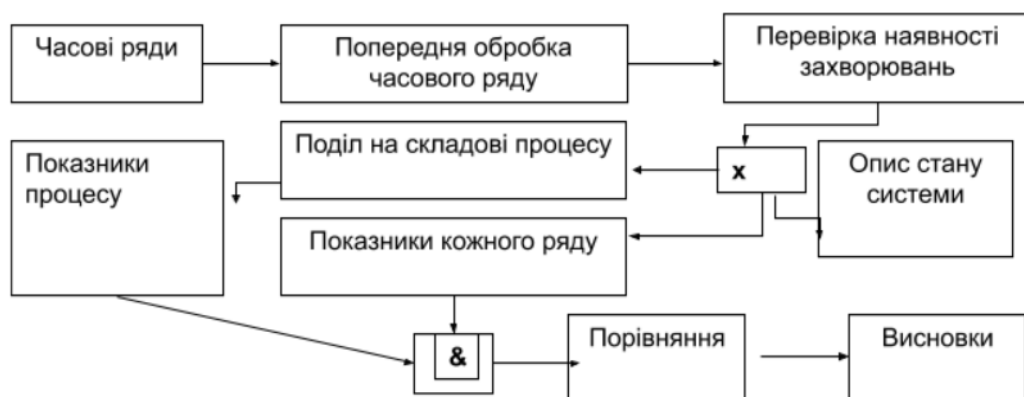


Рисунок 1.5 – Діаграма IDEF3

1.1.4 Інформаційна модель

Інформаційна модель системи відображає зв'язки між елементами системи у вигляді структур даних, акцентуючи увагу дослідника на склад та взаємозв'язках потоків даних. Діаграми потоків даних (Data Flow Diagramming, DFD) використовуються для опису документообігу та обробки інформації. Подібно до IDEF0, DFD є модельною системою. Діаграма DFD представлена на рис. 1.6.

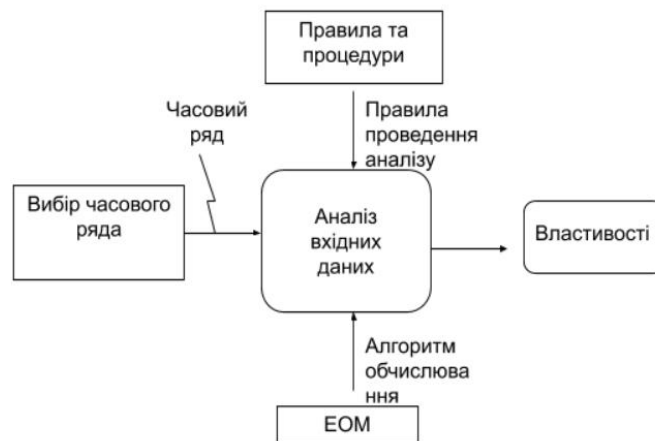


Рисунок 1.6 – Діаграма DFD

1.2 Аналіз сценаріїв вирішення проблеми класифікації медичних рядів

1.2.1 Модель аналізу проблеми

Об'єктом дослідження є проблемомістка система (ПМ система) на основі моделі «Розв'язання задачі класифікації медичних часових рядів».

Виділимо критерії, які на наш погляд здійснюють найбільший вплив на очікуваний результат. Такими критеріями є:

- точність отриманого результату (K1);
- зручність використання відповідно до вхідних даних (K2);

– складність програмування (K3).

Розглянемо вищевказані критерії більш детально та конкретизуємо що саме буде порівнюватися у запропонованих далі альтернатив. В свою чергу, альтернативами будуть методи, які частіше всього використовувались раніше у досліді для вирішення таких проблем.

Точність отриманого результату є ключовим критерієм серед розглядаємих, оскільки від точності може залежати визначення діагнозу, тому ми повинні орієнтуватися на цей показник і намагатися удосконалювати його.

Зручність використання методу відповідно до вхідних даних. Від того як виглядають вхідні дані залежить яким чином треба з ним працювати і які дії можливо зробити. Тому вибираючи метод потрібно врахувати як гарно цей метод може працювати з медичними часовими рядами, спираючись на його властивості та попередній досвід.

Складність програмування є найменш значущим критерієм, але про нього не можна забувати, бо складний код може призвести до занадто довгого виконання класифікації. В масштабах лікування це недопустимо.

1.2.2 Оцінювання вектора пріоритетів незадоволеностей методом аналізу ієрархій

Розглянемо декілька методів класифікації часових рядів:

- а) наївний байесовський класифікатор (Naive Bayes) (альтернатива 1);
- б) AdaBoost-SAMME (альтернатива 2);
- в) CNN (альтернатива 3);

Задамо 3 основні критерії для вибору методу:

- а) точність знаходження показника (критерій 1);
- б) зручність використання відповідно до вхідних даних (критерій 2);
- в) складність програмування (критерій 3).

Ієрархічна структура проблеми вибору представлена на рисунку 1.7.

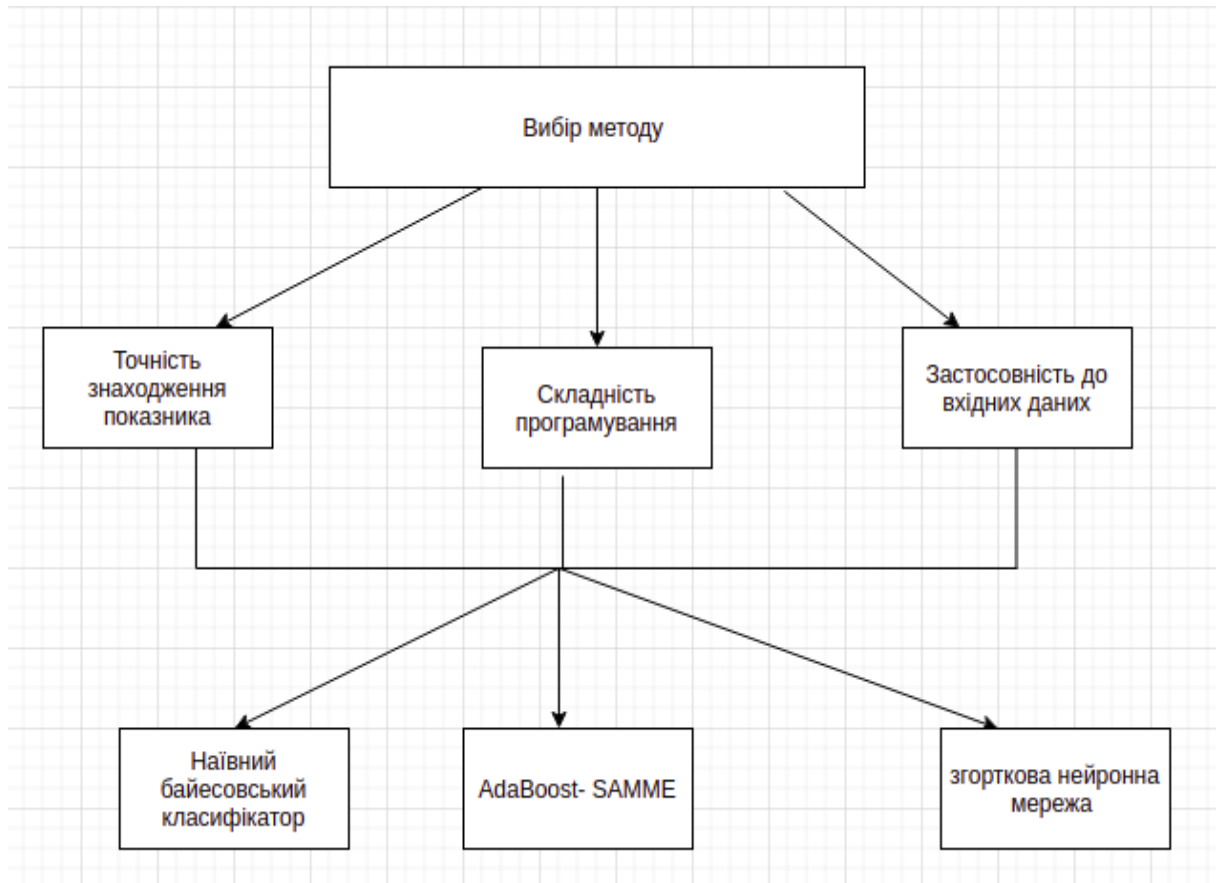


Рисунок 1.7 – Ієрархічна структура проблеми вибору методу

Спочатку сформуємо матрицю попарних порівнянь для елементів першого рівня ієрархії, тобто матрицю попарних порівнянь важливості критеріїв. Отримаємо наступні порівняння:

– точність знаходження показника значно важливіше, ніж складність програмування і очевидно важливіше, ніж зручність використання відповідно до вхідних даних;

– зручність використання слабо значима, ніж складність програмування.

Таким чином, маємо таку матрицю попарних порівнянь:

$$\begin{pmatrix} 1 & 5 & 7 \\ \frac{1}{5} & 1 & 3 \\ \frac{1}{7} & \frac{1}{3} & 1 \end{pmatrix}.$$

Розрахуємо для цієї матриці вектор локальних пріоритетів і величину значущості. Розрахунки зведені в таблицю 1.1.

Таблиця 1.1 – Розрахунки вектора локальних пріоритетів

Критерій	Оцінка компонентів	Вектор пріоритетів	Величина значущості
Критерій 1	3,271	0,731	0,9812
Критерій 2	0,843	0,188	1,1932
Критерій 3	0,362	0,081	0,8906
Разом	4,477	–	3,0649

Оскільки матриця попарних порівнянь критеріїв – це матриця третього порядку, то випадкова узгодженість дорівнює 0,58. Тоді отримаємо індекс узгодженості (IC) і відносною узгодженість (OC):

$$IC = \frac{3,0649 - 3}{3 - 1} = 0,0324, \quad (1.1)$$

$$OC = \frac{0,0324}{0,58} = 0,0559. \quad (1.2)$$

Результати порівнянь за шкалою Т.Сааті зведені в таблиці 1.2–1.4.

Таблиця 1.2 – Матриця попарних порівнянь альтернатив за K_1

K_1	A_1	A_2	A_3
A_1	1	2	$\frac{1}{3}$
A_2	$\frac{1}{2}$	1	$\frac{1}{3}$
A_3	3	3	1

Таблиця 1.3 – Матриця попарних порівнянь альтернатив за K_2

K_2	A_1	A_2	A_3
A_1	1	$\frac{1}{2}$	$\frac{1}{4}$
A_2	2	1	$\frac{1}{3}$
A_3	4	3	1

Таблиця 1.4 – Матриця попарних порівнянь альтернатив за K_3

K_3	A_1	A_2	A_3
A_1	1	3	5
A_2	$\frac{1}{3}$	1	2
A_3	$\frac{1}{5}$	$\frac{1}{2}$	1

Всі індекси відносної узгодженості в межах норми. Індекс загальної узгодженості буде дорівнює:

$$OC = \frac{IC}{CC} = \frac{0,0539058}{1,16} = 0,046 = 4,6\%, \quad (1.3)$$

де $IC = IC^K + (X^K, \overrightarrow{IC}^A)$ – індекс узгодженості;

$CC = CC^K + CC^A$ – випадкова узгодженість.

Таблиця 1.5 – Співвідношення альтернатив і критеріїв

Критерій/ Альтернатива	Критерій 1	Критерій 2	Критерій 2	Узагальнені Пріоритети
Альтернатива 1	0,249	0,136	0,648	0,260
Альтернатива 2	0,157	0,238	0,230	0,178
Альтернатива 3	0,178	0,625	0,122	0,561

Співвідношення альтернатив і критеріїв зведені в таблицю 1.5.

Виходячи з результатів проведеного аналізу, для збільшення точності класифікації будемо використовувати згорткові нейронні мережі.

1.3 Змістовна та формальна постановка задачі

1.3.1 Змістовна постановка

Електрокардіограма (ЕКГ) є найбільш широко використовуваною неінвазивною методикою при діагностиці захворювань серця. Оскільки вона відображає електричну активність всередині серця під час скорочення, то час і, його форма надають багато інформації про стан серця.

У XIX сторіччі виявили завдяки дослідженням у патології, що серце має електричний потенціал та протягом своєї роботи виробляє певну кількість електричної енергії. Перші ЕКГ 1888 року описав французький фізик, лауреат Нобелівської премії з фізики 1908 року Габріель Ліппман з використанням ртутного капілярного електрометра. Але ця методика не знайшла практичного використання. Нідерландський фізіолог Віллем Ейнтховен сконструював 1903 року прилад (струнний гальванометр), що дозволив йому реєструвати справжню ЕКГ. Ейндховен став засновником електрокардіографії, вперше 1906 року застосувавши її для медичної діагностики. Він дав назву зубцям ЕКГ, описав певні порушення в роботі серця. 1924 року йому було присуджено Нобелівську премію з фізіології та медицини з формулюванням «За відкриття техніки електрокардіограми» [1].

ЕКГ часто використовується для виявлення порушень ритму серця, також відомих як аритмії, які можна визначити в два способи: (і) як унікальний нерегулярний серцевий удар або (ii) як набір нерегулярних ударів. Аритмії можуть бути рідкісними і нешкідливими, але можуть також викликати серйозні серцеві проблеми. Деякі групи аритмій є небезпечними для життя і вимагають негайно-

го догляду і часто втручання з дефібрилятором. Інші типи аритмій, ті, які не потребують миттєвої уваги, можуть вимагати лікування для запобігання подальших проблем і їх слід також виявити. Багато аритмій проявляються як послідовності серцевих скорочень з незвичайними термінами або морфологією ЕКГ. Важливим кроком у визначенні аритмії є класифікація серцевих скорочень. Ритм сигналу ЕКГ потім можна визначити, знаючи класифікацію послідовних серцевих скорочень в сигналах [2]. Класифікація серцевих скорочень може бути дуже трудомістким процесом і, отже, будь-якою автоматизованою обробкою ЕКГ, що допомагає цьому процесу, допоможе і є у центрі уваги цього дослідження.

Аритмії часто асоціюються з іншими формами серцевих захворювань. Згідно з Всесвітньою організацією охорони здоров'я (ВООЗ): «Серцево-судинні захворювання (ССЗ) є причиною смерті № 1 у всьому світі: більше людей щорічно гине від ССЗ, ніж з будь-якої іншої причини. За оцінками, у 2015 році від ССЗ загинуло 17,7 млн. осіб, що становить 31% від загальної кількості смертей у світі». Для рутинного аналізу електричної активності серця, як правило, використовують ЕКГ, зареєстровану з 12 окремих провідників. 12-свинцева ЕКГ складається з трьох біполярних відводів кінцівок (I, II і III), однополярних відведеннях кінцівок (AVR, AVL і AVF) і шести однополярних грудних відводів, які також називаються прекардіальними або V-виводами (V1, V2, V3, V4, V5 і V6). Кожен свинець є видом електричної активності серця з певного кута по всьому тілу [3]. Запис містить приблизно 2,5 секунди для кожного свинцю. Крім того, для точної оцінки ритму серця запис з одного свинцю використовується для забезпечення ритмічної смуги 10s. Свинець II є найбільш часто використовуваним для смуги ритму, оскільки зазвичай дає гарний огляд найбільш важливих хвиль: P, Q, R, S і T (див. рис. 1.8.). Кожен удар серця містить серію відхилень від базової лінії на ЕКГ, або хвилі, що відображають еволюцію часу електричної активності в серці. P-хвиля — це невелика дефекція, викликана деполяризацією передсердь, хвилі Q, R і S зазвичай розглядаються як окремі події, відомі як QRS-комплекс, який є найбільшою

амплітудною частиною ЕКГ, викликаній вентральною деполяризацією. Т-хвиля обумовлена вентральною деполяризацією. Нарешті, в деяких випадках додаткова U-хвиля може слідувати за Т-хвилею.

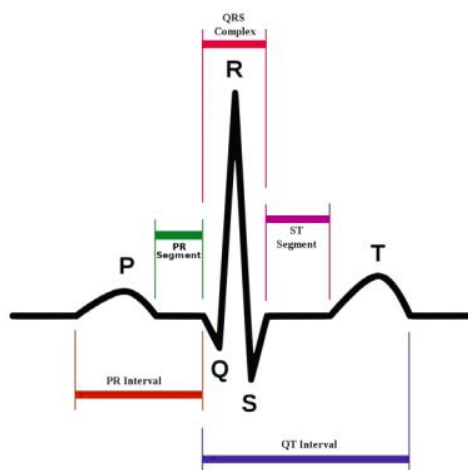


Рисунок 1.8 – Хвилі свинцевої ЕКГ II

На ЕКГ серцевий цикл розділений на зубці та інтервали, кожен з яких відповідає певній фазі розповсюдження хвилі збудження у міокарді. Зубець Р відповідає деполяризації передсердь (макс 0,12 сек). Інтервал Р–Q — поширення деполяризації до атріовентрикулярного вузла (проміжок часу від початку збудження передсердь до початку збудження шлуночків). Шлуночковий комплекс QRS (макс 0,10 сек, але у 21 % населення діагностується розширення комплексу до 0,12 сек, яке не вважається патологією) складається з трьох окремих зубців Q, R, S і відображає розповсюдження збудження тканиною шлуночків. Q — перше негативне відхилення від ізоелектричної лінії, R — перше позитивне відхилення, S — негативне відхилення після R зубця. Сегмент S–T — повна деполяризація волокон міокарда шлуночків (тому різниця потенціалів не виявляється). Зубець Т — хвиля реполяризації шлуночків. U хвиля є непостійним компонентом і може з'являтися при електролітних порушеннях (наприклад, гіпокаліємії) [4].

Автоматизована класифікація серцевих скорочень була раніше повідомлена іншими дослідниками з використанням різних функцій для пред-

ставлення ЕКГ і ряду класифікації методів. Особливості включають морфологію ЕКГ, функцію інтервалу серцебиття, функції на основі частоти, вищий порядок кумулятивних ознак, експансія Кархунена – Лоева, морфологію ЕКГ і поліномили герміту. Застосовувані класифікатори включають лінійні дискримінанти, нейронні мережі зворотного розповсюдження, самоорганізовані карти з вивченням векторного квантування та самоорганізовані мережі.

Розробка повністю автоматичної системи, яка здатна класифікувати серцебиття ЕКГ, була предметом досліджень, що викликало високий інтерес протягом останніх десятиліть. На рис. 1.9 представлена схема загальної автоматичної системи класифікації аритмій. Спочатку сигнали, які були захоплені через пристрій, попередньо обробляються. Цей етап зазвичай включає видалення базової лінії і очищення високочастотного шуму. Далі застосовується алгоритм сегментації серцевих скорочень, щоб розділити сигнал на рівні ритму. Зазвичай це робиться виявленням комплексу QRS. Потім до кожного збитків застосовуються кілька дескрипторів, щоб витягти функції для подачі класифікатора, який, нарешті, визначає тип серцебиття.

У цій роботі зосереджуємось на двох останніх кроках, вилученні ознак і класифікації. Було запропоновано багато ознак для опису серцевих скорочень ЕКГ, що виділяють використання вейвлетів, статистики вищого порядку і інтервалів серцевих скорочень, широко відомих як інтервали RR. Для побудови моделі класифікації, численні попередні роботи повідомляли про доцільність алгоритмів машинного навчання для цього завдання; включаючи такі методи, як лінійний дискримінант, AdaBoost, багатошаровий перцептрон, генетична алгоритм зворотного розповсюдження нейронної мережі, і, особливо, згорткові нейронні мережі.

Мета роботи: класифікувати медичні часові ряди за допомогою методів машинного навчання

Задачі, які були виконані для досягнення мети:

- а) вивчення технічної та медичної літератури;
- б) вивчення методів вирішення даної проблеми класифікації;

- в) вибір методу;
- г) розробка програми , реалізація методу;
- г) побудова класифікатора;
- д) опис проведеної роботи та результатів експерименту.

1.3.2. Формальна постановка задачі

В даній роботі вирішується задача класифікації на M класів.

Нехай задана множина об'єктів X , множина допустимих відповідей Y і існує цільова функція (target function)

$$y^* : X \rightarrow Y, \quad (1.4)$$

значення якої $y_i = y^*(x_i)$ відомі тільки на кінцевій підмножині об'єктів $\{x_1, \dots, x_i\} \subset X$. Пари «об'єкт-відповідь» (x_i, y_i) називаються прецедентами. Сукупність пар $X^l = (x_i, y_i)_{i=1}^l$ називається навчальною вибіркою (training sample). Завдання навчання по прецедентах полягає в тому, щоб за вибіркою X^l відновити залежність y^* , тобто побудувати вирішальну функцію (decision function) $a : X \rightarrow Y$, яка наближала б цільову функцію $y^*(x)$, причому не тільки на об'єктах навчальної вибірки , а й на всій множині X . Вирішальна функція a повинна допускати ефективну комп'ютерну реалізацію; з цієї причини будемо називати її алгоритмом.

Якщо $Y = \{1, \dots, M\}$, то це завдання класифікації (classification) на M непересічних класів. У цьому випадку вся множина об'єктів X розбивається на класи $K_y = \{x \in X : y^*(x) = y\}$, і алгоритм $a(x)$ повинен давати відповідь на питання «якого класу належить x ?»

Метод навчання (learning algorithm) — це відображення $\mu : (X \times Y)^l \rightarrow A$,

яке довільній кінцевій вибірці $X^l = (x_i, y_i)_{i=1}^l$ ставить у відповідність деякий алгоритм $a \in A$. Кажуть також, що метод μ будує алгоритм a по вибірці X^l . Метод навчання повинен допускати ефективну програмну реалізацію. Отже, в задачах навчання за прецедентами чітко розрізняються два етапи. На етапі навчання метод μ за вибіркою X^l будує алгоритм $a = \mu(X^l)$. На етапі застосування алгоритм a для нових об'єктів x видає відповіді $y = a(x)$. Етап навчання найбільш складний. Як правило, він зводиться до пошуку параметрів моделі, що доставляють оптимальне значення заданому функціоналу якості [9].

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

2.1 Основні архітектури нейронних мереж

Нейронні мережі, а точніше штучні нейронні мережі — це мережі для моделювання процесу навчання в мозку людини. Нервова система людини складається з нейронів, які з'єднані між собою за допомогою аксонів, які використовуються для передачі сигналів, і дендритів, які приймають сигнали від інших нейронів. Штучні нейронні мережі імітують цей біологічний механізм. Для зручності, штучні нейронні мережі називатимемо "нейронними мережами". Найпростіша архітектура нейронної мережі показана на рисунку 2.1. , де нейрон поєднує входи з регульованими вагами $\{w_i\}$ і подає комбінацію до функції активації для створення вихідних даних. Так як змінюються сили синаптичних зв'язків у відповідь на зовнішні подразники, значення цих ваг модифікуються під час тренувального процесу з використанням позначених навчальних даних. Згодом, мережа імітує функцію нейронно-біологічних організмів.

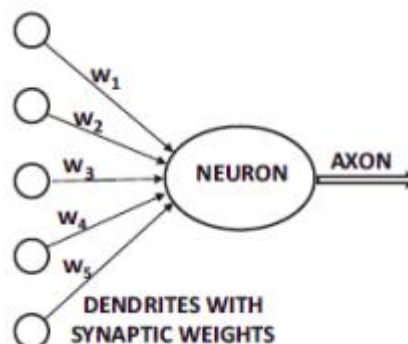


Рисунок 2.1 — Найпростіша архітектура нейронної мережі

Вищезазначену нейронну мережу часто називають сприйняттям. На рисунку 2.2 показано сприйняття без упередженість. У цій архітектурі вхідний рівень містить d входів $x = [x_1, \dots, x_d]$, які передаються до одиниці обчислення з

вагами $w = [w_1, \dots, w_d]$ і лінійною функцією

$$w \cdot x = \sum_{i=1}^d w_i x_i, \quad (2.1)$$

використовується для об'єднання зважених входів. Після цього функція знаку

$$f(z) = \{1, z \geq 0; -1, z < 0\}, \quad (2.2)$$

застосовується як активація, щоб отримати вихід y , а саме,

$$y = \text{sign}\left(\sum_{i=1}^d w_i x_i\right). \quad (2.3)$$

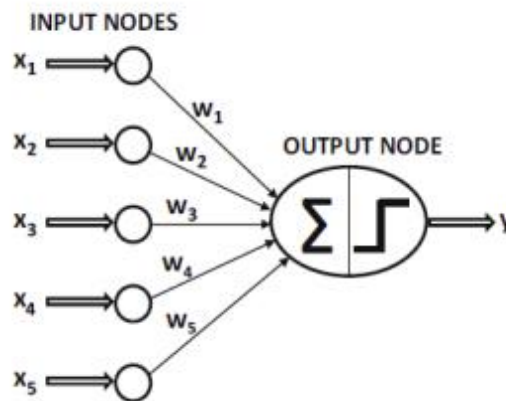


Рисунок 2.2 — Персептрон без біосу

У випадку, коли набір даних є дуже незбалансованим, прогнозування ефективності вищезазначеної архітектури може погіршитися. Наприклад, правильне середнє значення незбалансованого двійкового набору даних з мітками класів -1 або 1 є ненульовим, і в цьому випадку зазначена мережа потребує інваріантного зміщення b для вирішення проблеми див. рисунок 2.3.

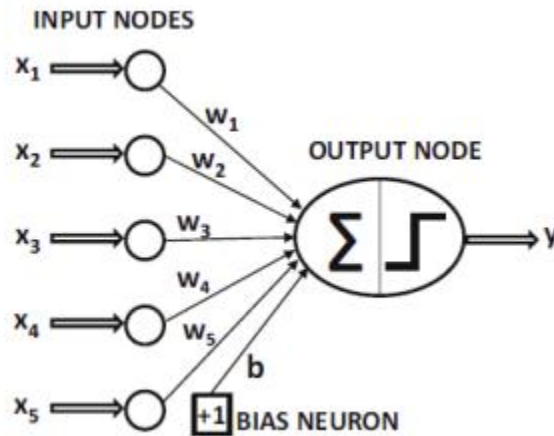


Рисунок 2.3 — Персептрон з біасом

Співвідношення вхід-вихід мережі на рис. 2.3 задано формулою

$$y = \text{sign}\left(\sum_{i=1}^d w_i x_i\right) + b. \quad (2.4)$$

Поєднуючи загальне сприйняття S , можна побудувати одношарову нейронну мережу, див. рис. 2.4.

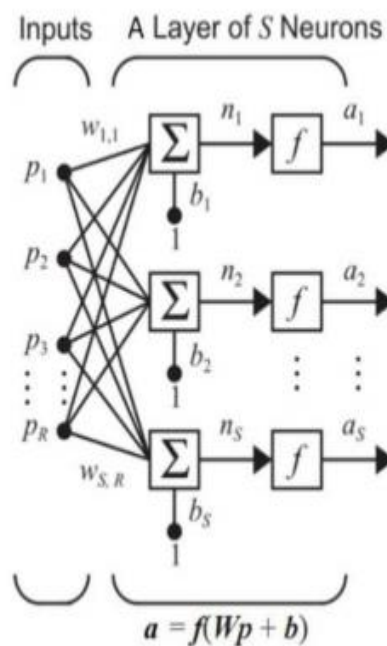


Рисунок 2.4 — Одношарова нейронна мережа

2.2 Технології машинного навчання для прогнозування на ЕКГ

Протягом багатьох років було запропоновано багато методів на основі ЕКГ для виявлення аритмії [1-8]. В загалом, метод у цьому класі складається з трьох основних етапів, а саме попередньої обробки, вилучення ознак, та класифікація. Ці методи відрізняються один від одного способом вилучення ознак та проведення класифікації. Загалом, методи вилучення ознак включають морфологію [2], тимчасову інформацію [3], статистику високого порядку [4], основні функції Ерміта [5], приховане Марківське моделювання (НММ) [6] та згорткові нейронні мережі (ЗНМ) [7]. Робота [7] описує метод досягнення точності 99,3% у діагностичній базі даних ЕКГ, де використовується ЗНМ для вилучення ознак та для підтримки класифікації використовується машина векторного забезпечення (МОП).

Робота [8] повідомляє про найсучасніші результати архітектури глибокого навчання на базі ЗНМ для ЕКГ прогнозування серцевих аномалій, де його показники були порівнянні з результатами шести кардіологів щоб продемонструвати, що модель перевершує в середньому показники роботи кардіологів, так і точність роботи [8] продемонструвала, що поглиблення вивчення, яке базується на ЗНМ, показує виявлення аритмій за сигналами ЕКГ.

2.3 Згорткові нейронні мережі для класифікації ЕКГ-сигналів

Згорткова нейронна мережа (ЗНМ) — одна з перших успішних архітектур глибокого навчання, особливо для класифікації зображень, відео, текстів та мови. У цьому розділі ми зупинимось на застосуванні ЗНМ для класифікації сигналів ЕКГ.

Стани в кожному шарі ЗНМ розташовані відповідно до просторової структури сітки. Значення з кожного виходу вузла виводиться у невелику локальну просторову область в попередньому шарі, і ці просторові відносини ус-

падковуються від одного шару до наступного [11]. Для класифікації зображень кожен шар у ЗНМ має три виміри — це висота, вага та глибина. Для сигналів ЕКГ, однак, вхід є одновимірним (1D), кожен шар ЗНМ має двовимірну архітектуру, а саме висота і глибина. Висота шару в ЗНМ відноситься до довжини вхідних сигналів, тоді як термін "глибина" відноситься до кількості каналів у кожному шарі.

У ЗНМ є три типи шарів, а саме згорткові шари, об'єднання шарів та активаційні шари. З цими рівнями пов'язані додаткові шляхи передачі сигналу та процедури для підвищення продуктивності, до них належать ярлик, пакетна нормалізація та відсіювання.

Операція згортки є визначальною характеристикою ЗНМ. Математична сутність згортки — це внутрішній добуток двох векторів (або матриць, або багатовимірних масивів) однакових розмірів, де один із векторів включає набір вагових коефіцієнтів, який зазвичай називають ядром, та інший вектор — це набір локальних вибірок із вхідних даних, що називається місцевим рецептивним полем. Внутрішній продукт завжди виконується з фіксованим ядром, тоді як місцеве сприйнятливим поле зміщується для того, щоб охоплюють весь вхід [6], див. рис. 2.5 та 2.6 для ілюстрації випадку з одним кроком.

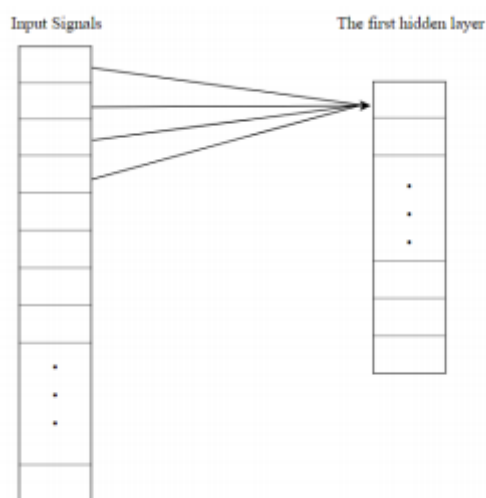


Рисунок 2.5 — Локальне рецептивне поле 1

Зазвичай розмір ядра є невеликим відносно довжини вводу для того, щоб гарантувати, що локальній функції з вводу витягуються. Якщо ми дозволимо введенням бути $x = \{x_i, i = 0, 1, \dots, n - 1\}$ і ядро бути довжиною L і позначається $w = \{w_l, l = 0, 1, \dots, L - 1\}$, одновимірна згортка $y = w \otimes x$ визначена від

$$y_i = \sum_{l=0}^{L-1} w_l x_{i+l}, \quad (2.5)$$

де індекс i для повномасштабної одномоментної згортки варіюється від 0 до $n - L$. Для повномасштабного S -кроку згортки, приріст індексу i дорівнює S , тому i буде змінюватися від 0 до $((n - L) / S)$, де нижній рівень (p) позначає найбільше ціле число не більше p .

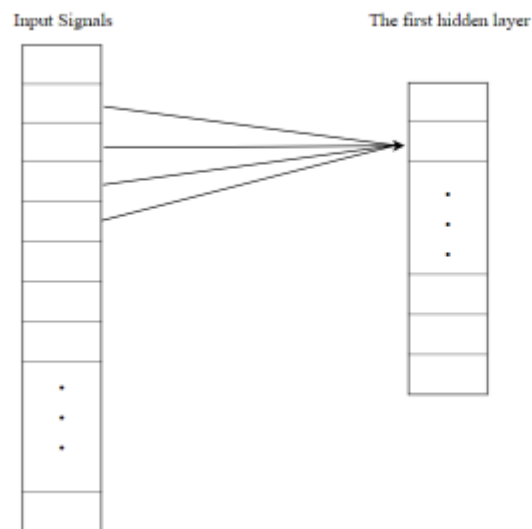


Рисунок 2.6 — Локальне рецептивне поле 2

В результаті односмугова згортка дає вектор із $L - 1$ компонентів менше у порівнянні з введенням. Наприклад, з $L = 9$ та сигналом ЕКГ довжиною 512 із набору даних MIT-BIH, згорткова операція дає вектор розміром 504 на 1. Потім вихід згортки подається у функцію активації f для отримання

$$a_i = f\left(\sum_{l=0}^{L-1} w_l x_{i+l} + b\right). \quad (2.6)$$

З виходу активації функція містить локальні особливості вхідного сигналу, вектор $\{a_i\}$ називається картою об'єктів. Щоб витягувати функції з вхідних даних послідовно, важливо, щоб усі локальні операції згортки використовували однаковий вектор ваги w та зміщення b . Ця стратегія позначається як спільні ваги та спільне упередження.

Очевидно, що w та b повинні бути обрані правильно, щоб ЗНМ працювала належним чином. Це виконується шляхом навчання $\{w, b\}$ з використанням позначених вхідних даних. На цьому етапі наголошується на тому, що згортка с пара тренуваних $\{w, b\}$ витягує лише одну карту об'єктів, а також для ЗНМ для класифікації ЕКГ для ефективної роботи необхідно використовувати кілька ядер, $\{W_i\}$ і b для $i = 1, 2, \dots, J$, в кожному даному прихованому шару ЗНМ див. рис. 2.7 для ілюстрації, де знаходяться три карти функцій виробляється трьома різними звивистими ядрами.

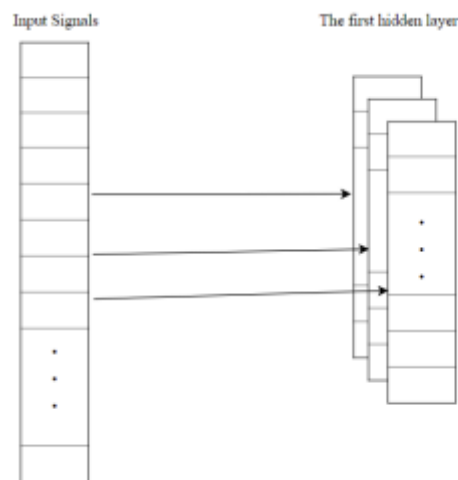


Рисунок 2.7 — Згорнутий прихований шар із трьома картами об'єктів

У порівнянні з повністю підключеними нейронними мережами (НМ) кількість параметрів у ЗНМ різко зменшилась. Для ілюстрації цього зменшення

складності розглянемо введення розміру 512 на 1, який подається у прихований шар з $J = 18$ та $L = 9$, щоб сформувати 18 карт функцій. У цьому випадку загальні $(9 + 1) \times 18 = 180$ параметрів беруть участь у шарі, тоді як повністю підключена НМ з 30 нейрони використовує $(512 + 1) \times 30 = 15390$ параметрів, у 84 рази більше ніж у ЗНМ.

Зменшення розміру не є бажаним у деяких реалізаціях, але проблему можна вирішити шляхом заповнення нулів на обох кінцях вхідної послідовності перед виконанням згортки. Кількість нулів, задіяних на кожному кінці в цій дії, називається розміром заповнення.

Функцію знаку введемо як функцію активації. Доступні різні інші функції активації, такі як сигмовидна функція, гіперболічний тангенс ($\tanh$), визначений

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (2.7)$$

і випрямлена лінійна одиниця (ReLU), яка дає $\max(0, x)$ у відповідь на вхід x . Ці і деякі інші популярні функції активації зображені на рисунку 2.8.

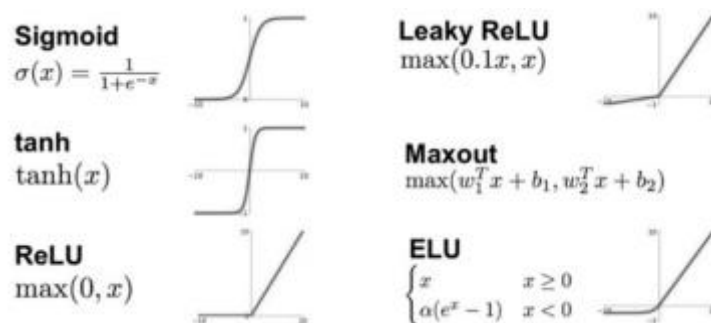


Рисунок 2.8 — Функції активації

Згорнутий шар у ЗНМ завжди супроводжується шаром об'єднання, що має на меті подальше зменшення складності та покращення стійкості мережі, що означає зменшення чутливості до просторових змін вхідних даних. Операція об'єднання спочатку розділяє ознаку відображення на субрегіони не-

великого розміру $P_q \times 1$ і вибирає одне репрезентативне значення з кожного регіону. Таким чином, дія об'єднання дає відповідну карту особливостей значно зменшеного розміру.

Середнє та максимальне об'єднання — це дві найпоширеніші операції об'єднання. На рисунку 2.9 показано простий приклад максимального об'єднання.



Рисунок 2.9 — Максимальне об'єднання

Якщо використовується крок 1, максимальне об'єднання з q -го згорткового шару дає новий шар розміру $(L_q - P_q + 1) \times d_q$, де L_p передає висоту вхідного сигналу, P_q дає розмір об'єднання і d_q передає кількість каналів цього нового шару. Якщо довжина кроку $S_q > 1$, розмір нового шар дорівнює

$$\left(\frac{L_q - P_q}{S_q} + 1 \right) \times d_q. \quad (2.8)$$

Середнє сподівання широко використовується в ранніх роботах ЗНМ, таких як LeNet-5, і називається це субдискретизація. Середнє об'єднання вибирає середнє значення по кожному субрегіону, просте приклад середнього об'єднання показаний на рисунку 2.10.

Рисунок 2.11 ілюструє зміни розміру 1D-сигналу, які індексуються за допомогою згортки і пулінг-операції, де три ядра 9×1 використовуються в згортковому шарі для генерації трьох карт функцій розміром 504, за якими слідує шар об'єднання розміром 2×1 і крок 2 для створення трьох нових функціональних карт із розміром, зменшеним удвічі.

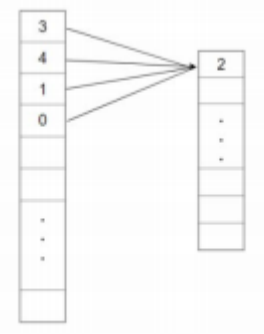


Рисунок 2.10 — Середнє об'єднання

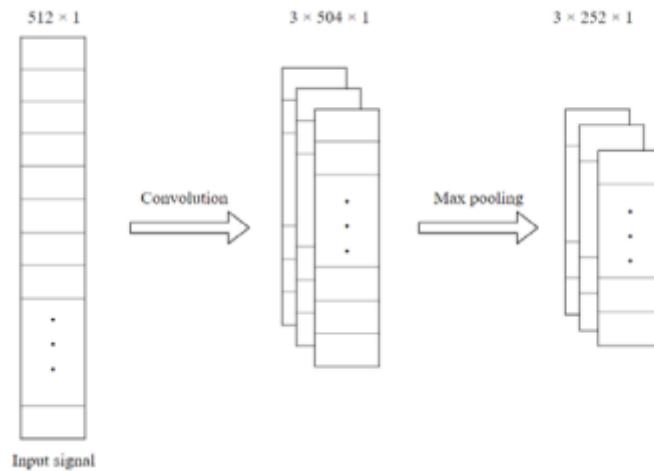


Рисунок 2.11 — Зміна розміру при поєднанні згорткового шару з шаром об'єднання

Повністю підключений шар — це звичайний шар у нейронній мережі, до якого підключений кожен вузол, всі ознаки якого, створені з попереднього шару, і ці ознаки поєднуються з вагами та зміщенням, чітко присвоєними цьому вузлу. На рис. 2.12 показана мережева архітектура з повністю з'єднаним шаром, який має п'ять нейронів. У цьому прикладі результат повністю пов'язаного ша-

ру є вектор розміром 5×1 . У цій архітектурі зв'язки між останньою картою об'єктів і повністю з'єднаними шарами мають $3 \times 252 \times 5 = 3780$ ваг.

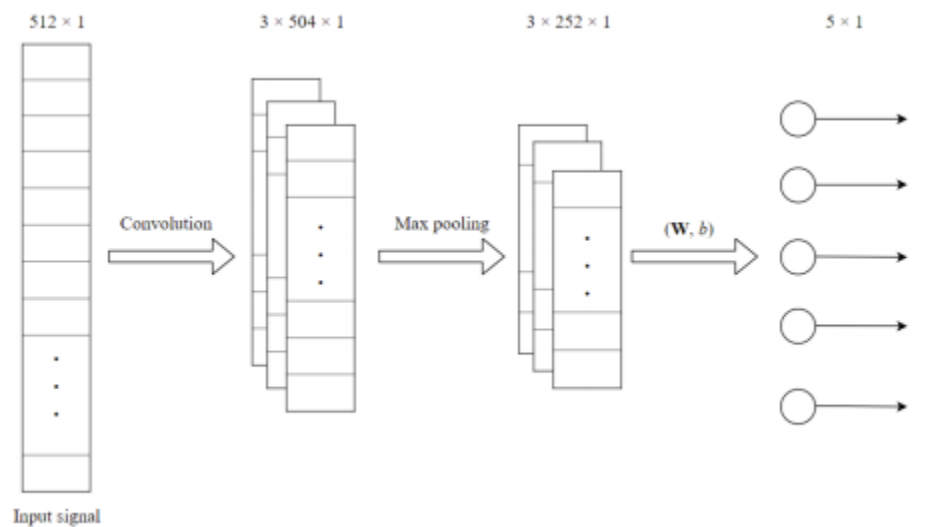


Рисунок 2.12 — ЗНМ з повністю зв'язаним шаром

Виходи останнього повністю підключеного шару подаються у вихідний шар для отримання вектора s , який передбачає ймовірність класу вхідних даних. Як правило, softmax використовується у вихідному рівні, де j -та компонента передбачає умовну ймовірність введення x належить до класу j і передбачає форму

$$s_j = \frac{e^{a_j}}{\sum_{k=1}^T e^{a_k}}, \quad (2.9)$$

де $\{a_i, i = 1, 2, \dots, T\}$ позначають виходи з останнього повністю підключеного шару.

2.4 Методи, спеціально розроблені для ЗНМ

Протягом багатьох років були розроблені різноманітні методики для підвищення продуктивності ЗНМ. Нижче описуються три з них, а саме нормалізація, ярлик і відсіювання, які були використані в моделі.

Навчання мережі складне, у цілій мережа використовується алгоритм, який називається зворотнім розповсюдженням для оцінки градієнта наперед заданої функції втрат.

Запропоновано пакетну нормалізацію для вирішення деяких із цих проблем. Рівень нормалізації пакетів завжди розміщується поруч із згортковим шаром, куди надходять дані від попереднього шару, нормалізується, масштабується та переміщується міні-партія за міні-партією. Рисунок 2.13 узагальнює етапи пакетної нормалізації.

Input: a mini-batch of m data samples $B = \{x_1, x_2, \dots, x_m\}$; parameters to be leaned: γ, β

Output: normalized, scaled, and shifted m data samples, $\{y_i = \text{BN}_{\gamma, \beta}(x_i), i = 1, 2, \dots, m\}$.

Step 1. Compute mean of B : $\mu_B = \frac{1}{m} \sum_{i=1}^m x_i$

Step 2. Compute variance of B : $\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$

Step 3. Normalize B : $\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$ for $i = 1, 2, \dots, m$

Step 4. Scaling and shifting: $y_i = \gamma \hat{x}_i + \beta \triangleq \text{BN}_{\gamma, \beta}(x_i)$ for $i = 1, 2, \dots, m$

Рисунок 2.13 — Етапи нормалізації партії

З перших трьох кроків на рисунку ми бачимо, що кожна міні-партія зразків даних нормується. Параметри γ, β , на кроці 4, слід навчити зсуву та масштабуванню. Робота [11] показує, що ці два параметри корисні стабілізованій роботі мережі під час навчального процесу. За останні роки глибина ЗНМ значно зросла, порівняно з відомим LeNet-5 з 5 шарами до деяких нещодавно за-

пропонованих архітектур ЗНМ з більш ніж 150 шарами [24]. Як очікується, глибші нейронні мережі важче навчити, оскільки збільшення мереж часто призводить до насиченої точності, навіть погіршення продуктивності. На малюнку 2.16 показано типовий приклад цього типу деградації, де випробовували 20-шарову та 56-шарову рівнинні мережі.

Набір даних CIFAR-10 та більш глибока мережа забезпечили вищу підготовку, а також помилки тестування. Цікаво, що погіршення стану не було спричинене переобладнанням, але помилка в навчанні зростає, коли до начебто адекватної моделі додається більше шарів.

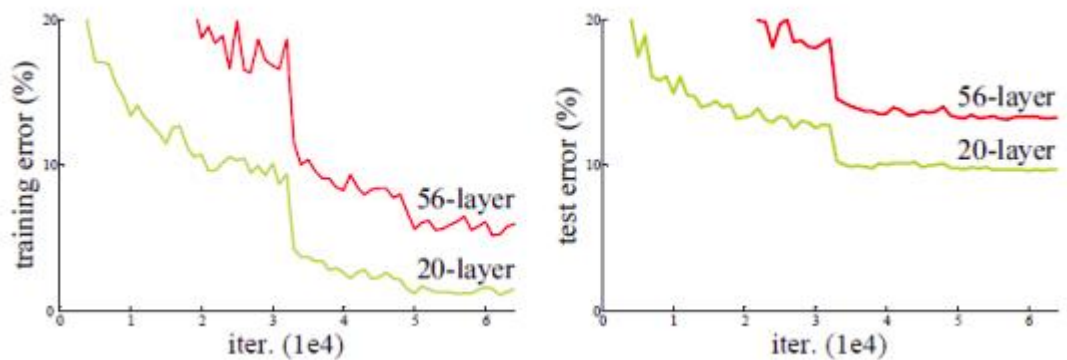


Рисунок 2.14 — Деградація більш глибоких мереж

Для вирішення проблеми в [10] запроваджено основи глибокого залишкового навчання. Для ілюстрації залишковий блок показаний на малюнку 2.15, де замість безпосереднього подавання вихідного шару на наступному шарі додається шлях ярлика для передачі даних із попереднього шару (рівнів) у поточний рівень активації.

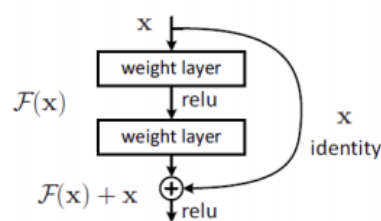


Рисунок 2.15 — Залишкове навчання: будівельний блок

Цей тип архітектури є корисним для зниження продуктивності деградація в більш глибоких мережах.

Випадання — це техніка вирішення проблеми переобладнання, яка часто є серйозною проблемою у глибокій нейронній мережі з великою кількістю параметрів, особливо коли навчальні дані обмежені. Також ефективним способом вирішення проблем із переобладнанням є припинення навчання, як тільки продуктивність набору перевірок починає погіршуватися. Ще одна корисна техніка — введення ваг штрафних санкцій різних видів, такі як регуляризація L_1 або L_2 та м'який розподіл ваг. Однак ці методи стають менш ефективними у великомасштабних мережах. Рисунок 2.16 показує приклад використання відсіву.

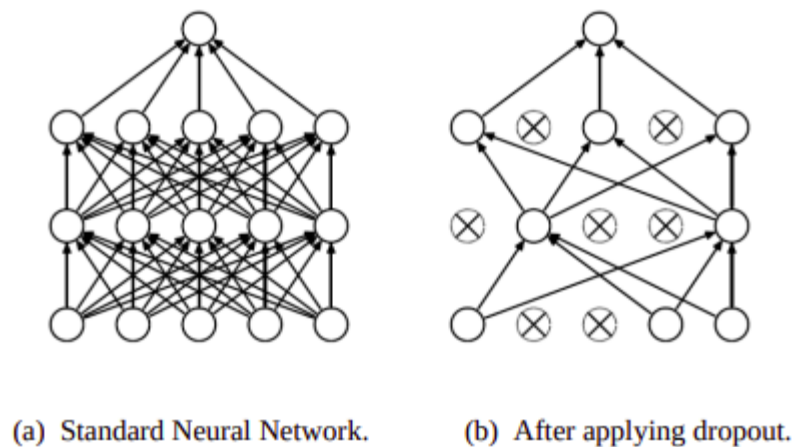


Рисунок 2.16 — Виключення нейронної мережі

У процесі навчання для кожної ітерації, прихованого шару та навчальної вибірки техніка тимчасово видаляє з мережі випадково вибрані блоки та їх вхідні та вихідні з'єднання з незалежно встановленою ймовірністю p створює «розріджену» мережу. І пряме, і зворотне розповсюдження здійснюється лише в цій розрідженій мережі. У тестовому екземплярі, як показано на рисунку 2.17, одиниці в мережі завжди присутні, а ваги помножені на ймовірність p .

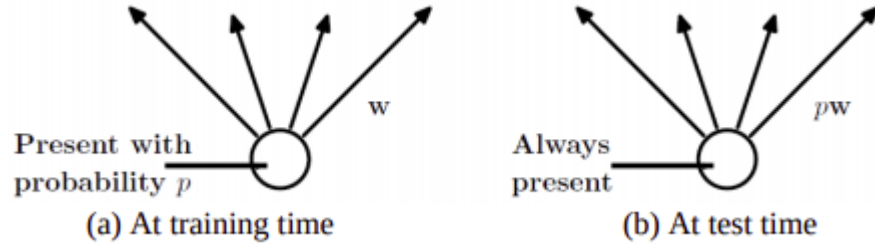


Рисунок 2.17 — Одиниці в виключених нейронних мережах

Щоб краще проілюструвати процедуру відсіву, розглянемо нейронну мережу з L прихованими шарами де вхід l -го шару подається $z^{(l)}$ для $l = 1, \dots, L$, вихід l -го шару дорівнює подаровані $y^{(l)}$, та $\{w^{(l)}, b^{(l)}\}$ — це ваги та ухил шару l . Як показано на малюнку 2.17 (а), в стандартні нейронні мережі, операція прямого пересилання для будь-якого прихованого блоку i можу описати як

$$\begin{aligned} z_i^{(l+1)} &= w_i^{(l+1)} y^{(l)} + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}), \end{aligned} \quad (2.10)$$

де f — функція активації.

Випадкова величина Бернуллі, яка має можливість p бути 1, $\tilde{y}^{(l)}$, зменшується і використовується як введення шару гнізда. Ця операція застосовується до кожного шару і створює набір розріджених підмереж.

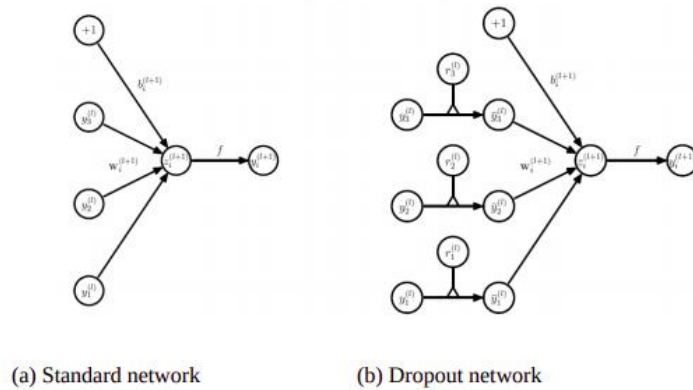


Рисунок 2.18 — Порівняння основних операцій стандартної мережі

з мережею з виключенням

2.5 Запропоновані моделі

Ці три моделі є ЗНМ з 35 шарами, 19 шарами та 9 шарами відповідно. Високий рівень архітектурної ілюстрації моделей показана на рисунку 2.20. Загальним моментом цих моделей є те, що кожна модель складається з трьох блоків: першого блоку, циклічного блоку і заключного блоку.

Перший блок включає три згорткові шари, блок циклу включає два згорткових шари, тоді як кінцевий блок має згортковий шар, повністю зв'язаний шар, і активацію softmax, яка передбачає ймовірність п'яти класів серцевих захворювань.

У 35-шаровій мережі, кількість циклів була встановлена на 15, тоді як це число було встановлено на 7 і 2 в 19-шаровій та 9-шаровій моделях відповідно. Нормалізація партії та активація ReLU застосовуються до кожного згорткового шару. Щоб запобігти переобладнанню, шари, що випадають, наносяться між згортковими шарами.

Крім того, у наших моделях використовуються ярликові з'єднання. Ці ярличні з'єднання покращуються навчальний процес, дозволяючи згортковому шару отримувати інформацію безпосередньо з попередніх шарів.

У наших моделях є $n + 1$ залишкових блоків, де n — кількість петлів. Операція максимального об'єднання застосовується для передискретизації входів ярликових з'єднань.

Якщо індекс блоків парний, розмір максимального об'єднання встановлюється як a , інакше він встановлюється як b , де $a = 2$ в 35-шаровій мережі, $a = 4$ у 19-шаровій мережі та $a = 16$ у 9-шаровій мережі, тоді як $b = 1$ у 35-шаровій та 19-шаровій мережі та $b = 8$ у 9-шаровій мережі.

Ця операція зменшує розмір виведення ярликових з'єднань на відповідне число, щоб переконатися, що його можна додати на вхідний шар, яким він подається.

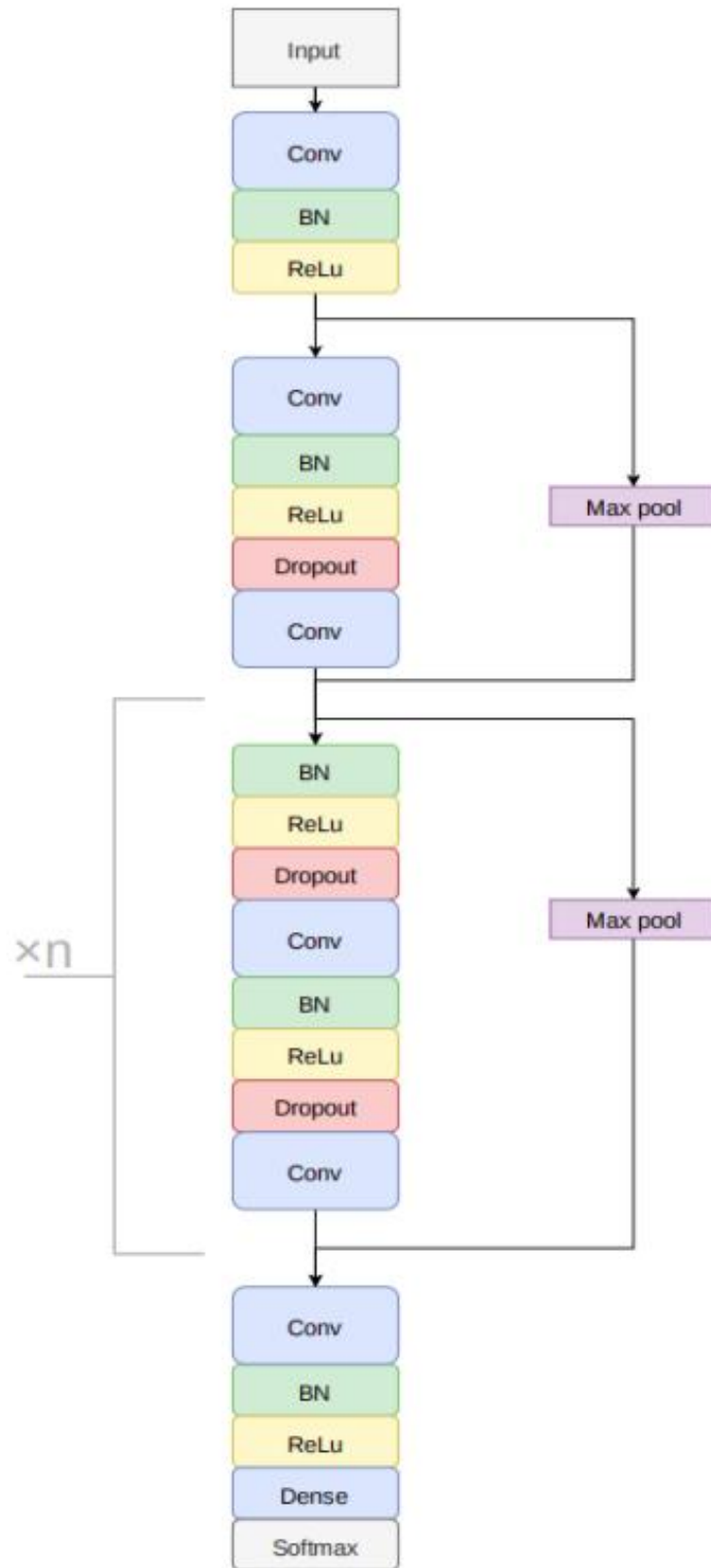


Рисунок 2.19 — Архітектурна ілюстрація 3-ох ЗНМ

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Python як мова машинного навчання

Ще в 1991 році, коли Гвідо ван Россум випустив Python як свій побічний проект, він не очікував, що це буде найшвидша зростаюча комп'ютерна мова у світі. Якщо ми будемо стежити за тенденціями, Python це мова для швидкого прототипування.

Машинне навчання, в не професійному плані, полягає у використанні даних для того, щоб зробити машину розумним рішенням. Наприклад: ви можете створити алгоритм виявлення спаму, де правила можна дізнатися з даних або виявити аномалії рідкісних подій, переглянувши попередні дані або організувавши електронну пошту на основі тегів, які ви призначали, вивчаючи історію електронної пошти тощо.

Машинне навчання — це не що інше, як розпізнавання моделей у ваших даних.

Важливим завданням машинобудівного інженера в його робочому житті є вилучення, обробка, визначення, очищення, упорядкування і потім розуміння даних для розробки інтелектуальних алгоритмів. Тому для інженера з машинного навчання / інженера з комп'ютерного бачення чи інженера з вивчення даних / машинного навчання / інженера з алгоритму / глибокого навчання рекомендують Python, тому що його легко зрозуміти. Іноді поняття лінійної алгебри, числення є настільки складними, що вони потребують максимальної кількості зусиль. Швидка реалізація в Python допомагає інженеру ML перевіряти ідею.

Отже, все повністю залежить від типу завдання, в якому ви хочете застосувати машинне навчання. Уявіть, що все, що існує навколо вас, є даними. І вони сирі, неструктуровані, погані, неповні, великі. У цій мові дуже велика кількість пакетів. Це колекція коду різних сховищ з відкритим вихідним кодом, які розробляються людьми (все ще перебувають у процесі) для постійного вдо-

сконалення існуючих методів.

Найкраще використання цих пакетів полягає в тому, що вони мають нульову криву навчання. Як тільки у вас є базове розуміння Python, ви можете просто реалізувати його. Вони вільні для використання за ліцензією GNU. Просто імпортуйте пакет і використовуйте його.

Основною причиною або єдиною причиною, чому Python ніколи не буде використовуватися дуже широко, є накладні витрати, які він приносить. Невеликі процесори або обладнання з низькою пам'яттю сьогодні не будуть вміщувати базу кодів Python, але для таких випадків ми використовуємо C і C++ як наші засоби розробки. З усього вище написаного можна зробити висновок що Python повністю відповідає цілям даної роботи і є ідеальним варіантом для використання.

3.2 Опис програми

Запропоновані моделі будуються та навчаються з нуля в Python за допомогою популярної бібліотеки нейронних мереж Keras. Шари в мережах реалізовані наступним чином:

```
Convolutional layers: layer = Conv1D(filters = filter_length,
                                     kernel_size = config.kernel_size,
                                     padding = 'same',
                                     strides = 1,
                                     kernel_initializer = 'he_normal')(layer)

Activation layers:    layer = Activation('relu')(layer)

Batch Normalization: layer = BatchNormalization()(layer)

Dropout layers:       layer = Dropout(config.drop_rate)(layer)

Shortcut:              shortcut = MaxPooling1D(
                        pool_size=subsample_length)(layer)
```

Рисунок 3.1 — Реалізація шарів в мережі

Ваги в згорткових шарах ініціалізуються методом `he_normal` [8]. Клас моделі і функція компіляції в Keras використовувалася для навчальних мереж, Адам [9] — як оптимізатор, `categorical_crossentropy` втрата використовується як функція втрати, встановлено початкову швидкість навчання до 0,2, а потім зменшено в 10 разів відповідно до покращення втрат від перевірки:

```
model = Model(inputs=inputs, outputs=outputs)
adam = Adam(lr = 0.2, beta_1 = 0.9, beta_2 = 0.999, epsilon=None,
            decay = 0.0, amsgrad= False)
model.compile(optimizer= adam, loss='categorical_crossentropy',
              metrics=['accuracy'])
```

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАНОГО ЕКСПЕРИМЕНТУ

4.1 Вхідні дані

База даних MIT–BIH забезпечує 48 півгодинних двоканальних записів ЕКГ, які отримуються Лабораторією аритмії ВІН з набору 4000 24–годинних амбулаторних записів ЕКГ зібраних від пацієнтів у бостонській лікарні Бет-Ізраель між 1975 і 1979 рр. Використовувані масиви даних у цьому проекті були похідні з бази даних MIT–BІН та модифіковані для зручних для Python форм з використанням алгоритму [11]. Модифіковані набори даних містять чотири частини, описані в таблиці 4.1.

Таблиця 4.1 — Опис наборів даних

Name	Columns	Rows	Description
normalrr1.mat	33617	553	Normal beat
apcrr2.mat	2178	553	Atrial premature beat
pvcrr2.mat	5725	553	Premature ventricular contraction
lbbbrr1.mat	3992	553	Left bundle branch block beat
rbbbrr1.mat	4467	553	Right bundle branch block beat

Набори даних містять п'ять класів серцевих скорочень, а саме: нормальне серцебиття, передчасне серцебиття передсердь, передчасне скорочення шлуночків, биття гілки лівого пучка та блокування гілки правого пучка. Кожен такт представлений вектором розміром 553 на 1, отже, в ньому є 49979 ударів набори даних.

4.2 Результати обчислювального експерименту

Використовуючи функцію `fit` в класі `Model`, наші три моделі пройшли навчання та тестування за допомогою попередньо оброблених наборів даних, де кількість навчальних епох було встановлено на 50 і встановлено розмір партії до 128:

```
model.fit(xtr, ytr,
          validation_data=(xte, yte),
          epochs=config.epochs,
          batch_size=config.batch,
          callbacks=callbacks,
          initial_epoch=initial_epoch)
```

Оскільки база даних MIT–BIN не надає окремих даних тестування, ми випадково вибираємо частину даних для навчальних цілей та іншу частину даних для тестування. Слід зазначити, що незалежно від того, чи використовується конкретна форма серцебиття в якості тренувань чи тестування, вона цілком визначається початковим випадковим станом, що використовується для випадкового розбиття. В результаті числових висновків, що базуються на таких наборах даних, дуже ймовірно змінюватимуться, коли використовується набір даних, побудований за допомогою різних початкових випадкових станів. У нашому дослідженні загалом було досліджено 20 випадкових наборів даних, а результати статистично проаналізовані з точки зору класифікації точності, часу навчання та часу тестування. Точність класифікації визначається як відношення кількості зразків, які правильно прогнозуються до загальної кількості зразків:

$$accuracy = \frac{correctly\ predicted\ samples}{samples}. \quad (4.1)$$

Експерименти проводились на Google Collaboratory, і отримані результати зазначено в таблицях 4.2 та 4.3.

Таблиця 4.2 — Точність класифікації трьох моделей на базі CNN

Training set	Classification accuracy		
	9-layer network	19-layer network	35-layer network
1	0.9867	0.9882	0.9943
2	0.9876	0.9901	0.9886
3	0.9887	0.9923	0.9895
4	0.9877	0.9911	0.9888
5	0.9901	0.9931	0.9865
6	0.9845	0.9930	0.9903
7	0.9845	0.9884	0.9889
8	0.9876	0.9913	0.9882
9	0.9887	0.9896	0.9932
10	0.9823	0.9928	0.9901
11	0.9865	0.9921	0.9884
12	0.9812	0.9913	0.9895
13	0.9876	0.9987	0.9927
14	0.9888	0.9902	0.9927
15	0.9876	0.9923	0.9921
16	0.9834	0.9889	0.9897
17	0.9898	0.9923	0.9896
18	0.9902	0.9886	0.9890
19	0.9884	0.9885	0.9920
20	0.9854	0.9912	0.9883
Average	0.9869	0.9910	0.9900

З таблиці 4.2 видно, що 19-шарова мережа змогла досягти точності 99,1% всередній. Також зауважте, що 35-шарова та 19-шарова архітектури за-

безпечують майже однакові прогнози точність; але з таблиці 3.3 19-шарова модель значно швидша, ніж модель із 35 шарами, з точки зору як часу навчання, так і часу тестування. На рисунку 4.1 показані матриці сплутаності трьох моделей, які були усереднені за 20 набори даних. Зазначається, що розподіл помилок класифікацій видається дещо рівномірним, з більшою кількістю помилок класифікацій сталося для класів “n” та “a” щодо класів “l” та “r”.

	n	a	p	r	l
n	3318	13	9	2	1
a	10	186	0	1	0
p	13	7	578	1	1
r	0	0	1	453	0
l	3	1	2	0	398

	n	a	p	r	l
n	3328	7	10	1	0
a	9	197	1	0	0
p	7	2	577	1	3
r	0	1	1	455	1
l	0	0	1	0	396

	n	a	p	r	l
n	3326	8	3	1	0
a	10	195	2	1	0
p	6	1	576	1	3
r	2	3	8	454	0
l	0	0	1	0	397

Рисунок 4.1 — Матриці сплутаності трьох моделей

Вивчити стійкість нейронних мереж на базі CNN проти змін у відносних розмірах Дані навчання та тестування визначаємо коефіцієнт розбиття як відношення розміру тестування до розміру весь набір даних та оцініть середню ефективність трьох моделей щодо коефіцієнтів розбиття в діапазоні від 0,1 до 0,5. Чисельні результати наведені в таблиці 4.3.

Таблиця 4.3 — Середня ефективність трьох моделей

Splitting ratio	9-layer network	19-layer network	35-layer network
0.1	0.9869	0.9910	0.9900
0.2	0.9863	0.9902	0.9895
0.3	0.9856	0.9875	0.9876
0.4	0.9821	0.9868	0.9862
0.5	0.9813	0.9865	0.9858

З таблиці 4.3 видно, що три моделі досить нечутливі до змін розщеплення співвідношення.

Важливою характеристикою мереж CNN є те, що кожен згортковий шар (С-шар) генерує кілька функціональних карт, які переносяться на наступний С-шар за допомогою субдискретизації та об'єднання. Тому представляє інтерес візуалізувати окремі функції з різних каналів по-різному згорткові шари. Враховуючи вхідний сигнал ЕКГ, кожен рядок на малюнку 4.2 зображує 4 сигнали зеленого кольору з n -ї карти особливостей.

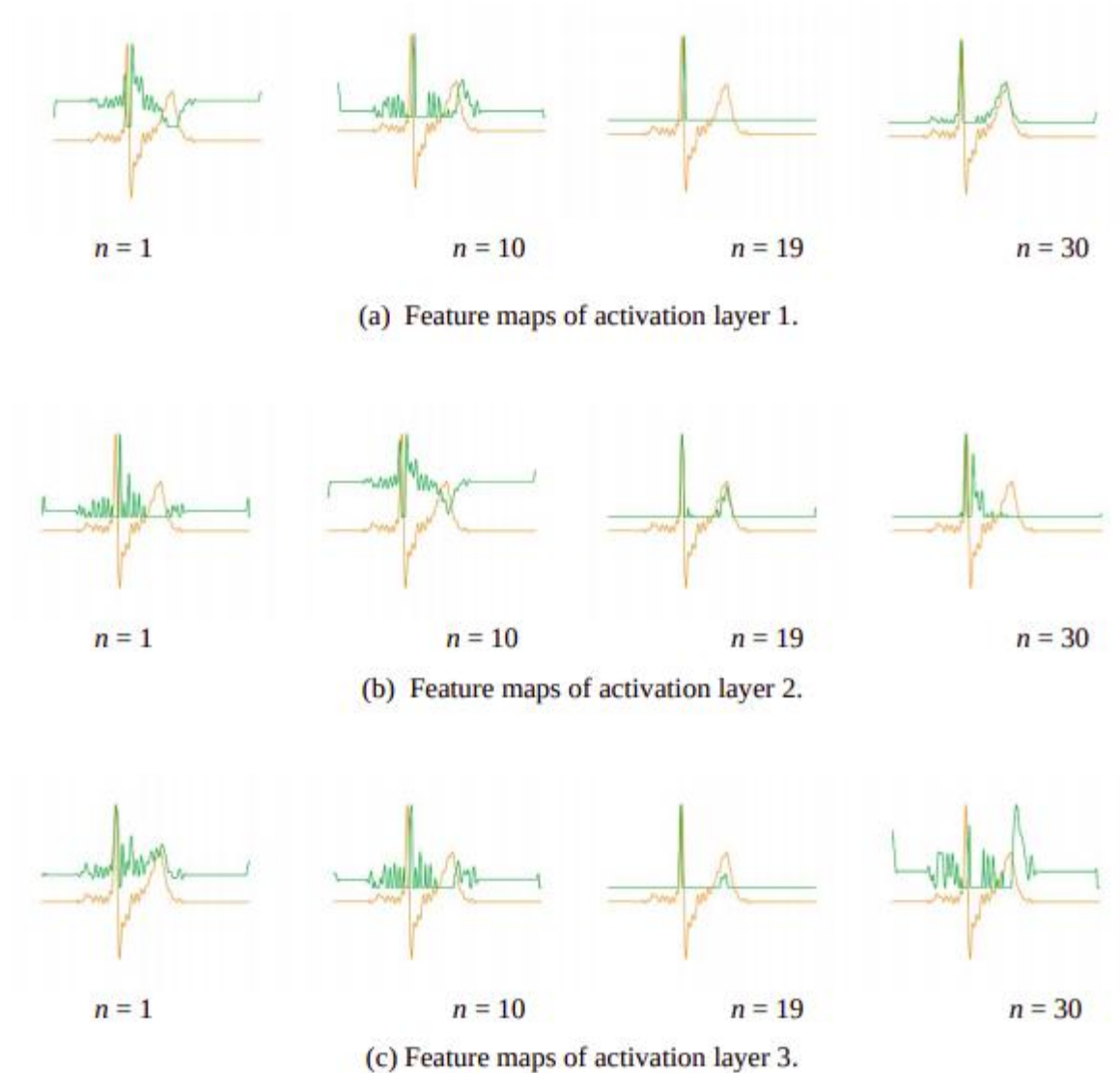
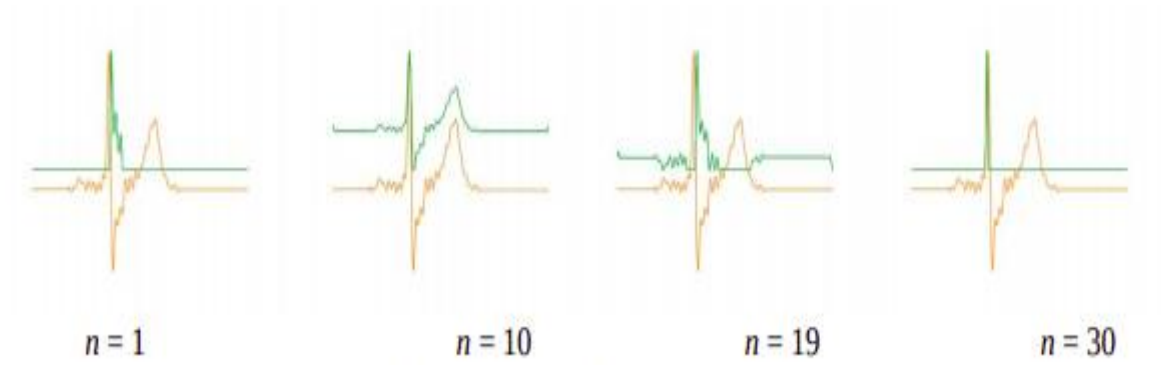
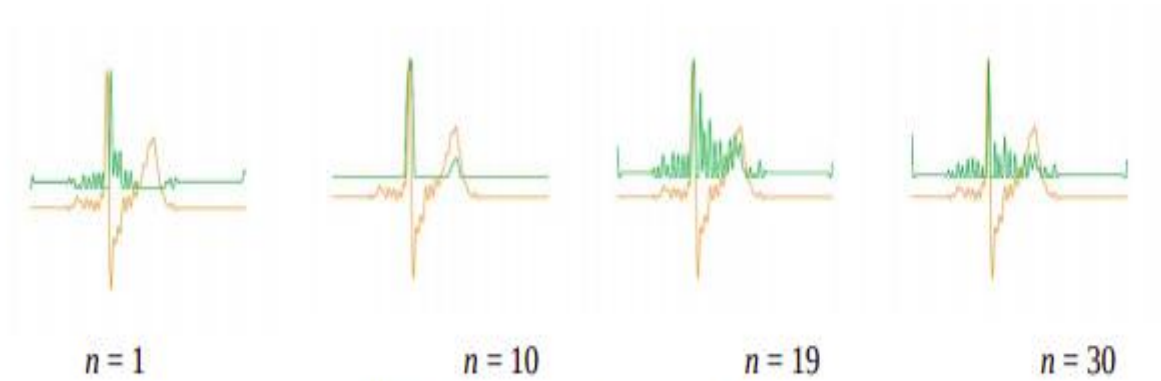


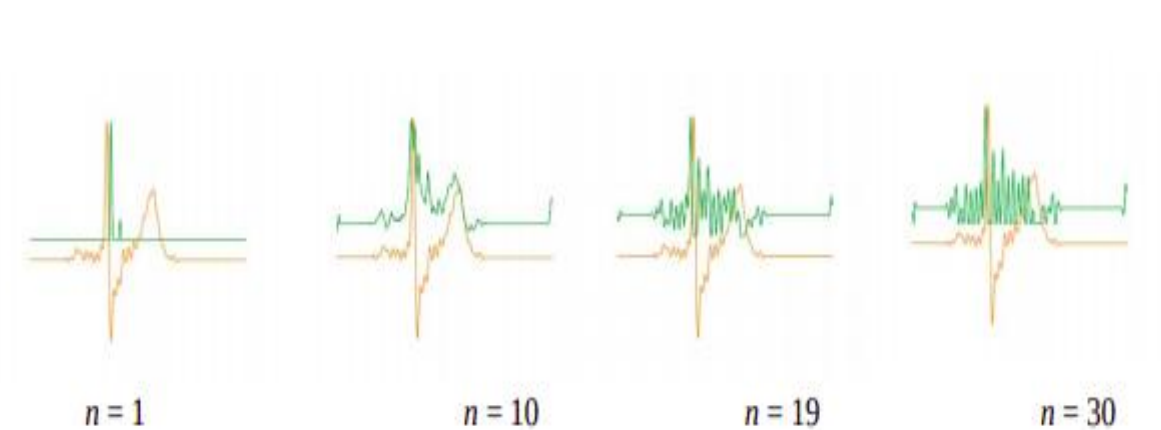
Рисунок 4.2 — Карти функцій для шарів 1-3 (1)



(a) Feature maps of activation layer 1.

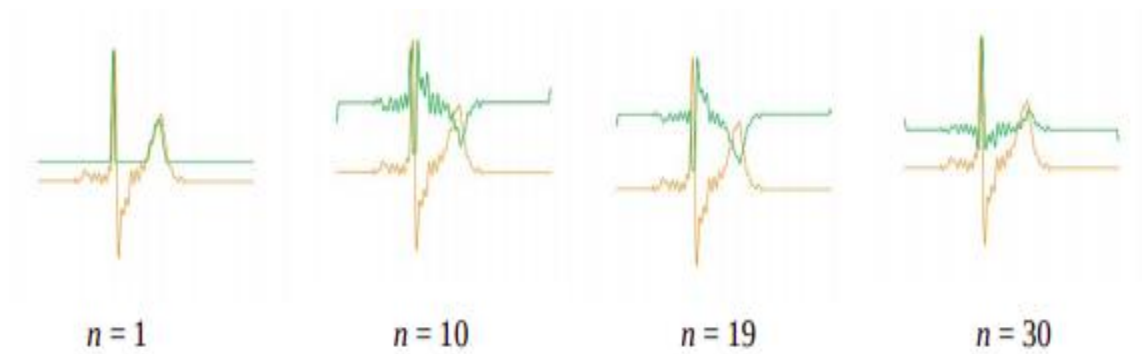


(b) Feature maps of activation layer 2.

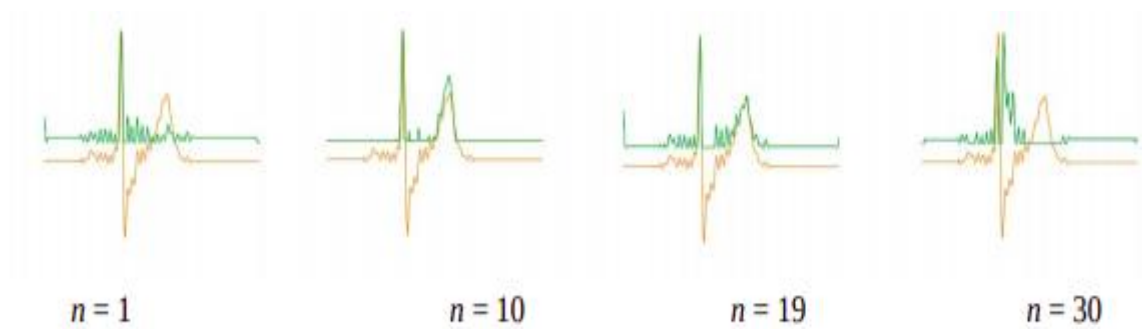


(c) Feature maps of activation layer 3.

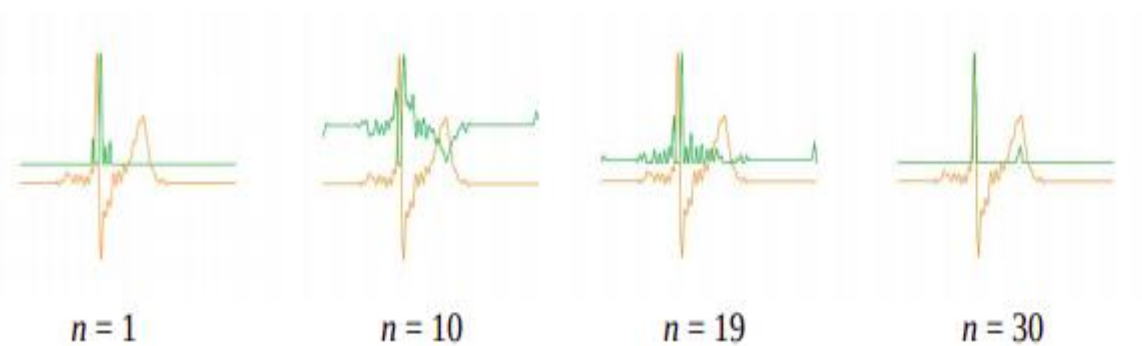
Рисунок 4.3 — Карти функцій для 1-3 (2)



(a) Feature maps of activation layer 1.



(b) Feature maps of activation layer 2.



(c) Feature maps of activation layer 3.

Рисунок 4.4 — Карти функцій для 1-3 (3)

5 АНАЛІЗ МОЖЛИВИХ ЗАСТОСУВАНЬ

Дана робота є одним з способів класифікації медичних часових рядів. Вона може бути використана як основа для подальших медичних досліджень у цій сфері.

Пряме застосування її у медицині на даному етапі не буде стовідсотково правильним.

Маючи обмежені дані, найкраще з дуже широкого різноманіття моделей машинного навчання зазвичай працюють краще, ніж один клас моделей (наприклад, нейронні мережі). Однак, з іншого боку, набори даних, використані у цій роботі, відносно невеликі.

Дана робота може стати порівняльною роботою для майбутніх її роглядань та для отримання найвищої точності результатів у інших роботах.

Крім того, наш підхід також показав кращі результати в порівнянні з попередніми методами машинного навчання сучасного рівня техніки.

ВИСНОВКИ

У роботі представлені три моделі на основі CNN для класифікації форм ЕКГ. Ці моделі були протестовані на наборах даних MIT-BIH. Ключовою перевагою нейронних мереж перед традиційним машинним навчанням є те що вони забезпечують абстракцію вищого рівня. Друга перевага полягає в тому, що нейронні мережі забезпечують простий спосіб регулювання складності моделі шляхом додавання або видалення нейронів з архітектури відповідно до наявності навчальних даних або обчислювальної потужності.

Ефективність традиційного машинного навчання в рази залишається кращою для менших наборів даних через більший вибір, більшу простоту інтерпретації моделі та схильність до ручної роботи інтерпретованих функцій. Маючи обмежені дані, найкраще з дуже широкого різноманіття моделей машинного навчання зазвичай працюють краще, ніж один клас моделей (наприклад, нейронні мережі). Однак, з іншого боку, набори даних, використані у цій роботі, відносно невеликі. Є 20 класів серцевих скорочень у наборах даних MIT-BIH, але в наших дослідженнях було використано лише п'ять з них.

Для вдосконалення майбутньої роботи слід враховувати інтеграцію таких методів, як LSTM, які продемонстрували вже вражаючі здібності для архітектур CNN.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Wikipedia. URL : <https://uk.wikipedia.org/wiki/Електрокардіографія> (дата звернення 24.05.2019).
2. de Chazal P. , O'Dwyer M. , Reilly R.B. Automatic classification of heartbeats using ECG morphology and heartbeat interval features // IEEE Trans. Biomed. Eng. 2004. 51 (7). P. 1196–1206.
3. Meek S. , Morris F. Introduction. I – Leads, rate, rhythm, and cardiac axis// BMJ. 2002. 324 (7334). P. 415–418.
4. Pan J. , Tompkins W.J. A real-time QRS detection algorithm // IEEE Trans. Biomed. Eng. 1985. BME-32 (3). P. 230–236.
5. Dietterich T.G. , Bakiri G. Solving multiclass learning problems via error-correcting output codes //J. Artif. Int. Res. 1995. 2 (1). P. 263–286.
6. Duin R.P.W. , Tax D.M.J. Experiments with classifier combining rules Multiple Classifier Systems // Berlin: Heidelberg. 2000. P. 16–29.
7. Schmidhuber J. Deep learning in neural networks: An overview // Neural Networks. 2015. Vol. 61. C. 85–117.
8. de Chazal P. , O'Dwyer M. , Reilly R.B. Automatic classification of heartbeats using ECG morphology and heartbeat interval features // IEEE Trans. Biomed. Eng. 2004. 51 (7) . P. 1196–1206.
9. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: исследование зависимостей. Москва : Финансы и статистика, 1985. 256 с.
10. Srivastava N., Hinton G. E, Krizhevsky A., Sutskever I., and Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting // Journal of Machine Learning Research. 2014. Vol. 15, no. 1. P. 1929–1958.
11. da S E.J. . Luz, Schwartz W.R. , Cmara-Chvez G. , Menotti D. ECG-based heartbeat classification for arrhythmia detection: a survey // Comput. Methods Prog. Biomed. 127 (Suppl. C). 2016. P. 144–164.