

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра комп'ютерних наук

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему «Класифікація звуків пострілів на основі спайкінгової нейронної мережі»

Виконав: студент 2 курсу,
групи 1КН-18 м
спеціальності 122 «Комп'ютерні науки»
Кюльян І.Г.
Керівник: к.т.н., доц. Колесницький О.К.
Рецензент: к.т.н, доц. Рейда О.М.

Вінниця - 2019 року

ЗАТВЕРДЖУЮ
Директор ТОВ «ІНКОРСОФТ»
Сивинюк О. В.
(наук. ст., вч. зв., ініц. та прізви.) (підпис)
" ____ " _____ 2019 р.

ЗАТВЕРДЖУЮ
Завідувач кафедри __ КН
д. т. н., проф. Яровий А.А.
(наук. ст., вч. зв., ініц. та прізви.) (підпис)
" ____ " _____ 2019 р.

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра зі спеціальності: 122 – «Комп'ютерні науки»

08-22.МКР. 008.18.107.ПЗ

Магістранта групи 1КН-18м Кюльяна Ігоря Геннадійовича

Тема магістерської кваліфікаційної роботи: «Класифікація звуків пострілів на основі спайкінгової нейронної мережі»

Вхідні дані: тривалість звукового сигналу не менше 2 с, кількість класів не менше 4, достовірність розпізнавання не менше 90%, мова програмування об'єктно орієнтована, операційна система Linux/Ubuntu.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: Схема роботи алгоритму програмного засобу класифікації звуків пострілів на основі спайкінгової нейронної мережі, структура спайкінгової нейронної мережі, структура програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі, UML-діаграма класів програмного засобу класифікації звуків пострілів на основі спайкінгової нейронної мережі, візуалізація частотних смуг, результати тестування роботи програмного засобу класифікації звуків пострілів на основі спайкінгової нейронної мережі.

2. Текстова (пояснювальна записка): вступ, аналіз предметної області класифікації звуків пострілів на основі спайкінгової нейронної мережі, розробка методу класифікації звуків пострілів на основі спайкінгової нейронної мережі, програмна реалізація методу класифікації звуків пострілів на основі спайкінгової нейронної мережі, економічна частина, висновки, перелік використаних джерел, додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

| № етапу | Назва етапу | Термін виконання | | Очікувані результати |
|---------|---|------------------|--------|--|
| | | початок | кінець | |
| 1 | Аналіз предметної області класифікації звуків пострілів на основі спайкінгової нейронної мережі. Постановка задач дослідження | | | Аналітичний огляд літературних джерел, задач досліджень, розділ 1 ПЗ |
| 2 | Розробка методу класифікації звуків пострілів на основі спайкінгової нейронної мережі. Розробка структури нейронної мережі. Удосконалення алгоритму навчання. | | | Метод класифікації, структура мережі, алгоритм навчання, розділ 2 |
| 3 | Програмна реалізація розробленого методу класифікації звуків пострілів на основі спайкінгової нейронної мережі, тестування та оцінка параметрів | | | Програмне забезпечення, розділ 3 |
| 4 | Підготовка економічної частини | | | Розділ 4 |
| 5 | Апробація та/або впровадження результатів дослідження | | | Тези доповідей/акт впровадження |
| 6 | Оформлення пояснювальної записки, графічного матеріалу та презентації | | | Пояснювальна записка, графічний матеріал, презентація |

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____ канд. техн. наук, доц., доц. кафедри КН
 (підпис) _____ наук. ступінь, вчене звання (посада)
 “ ____ ” _____ 20__ р. _____ О. К. Колесницький
 ініціали та прізвище

2. Економічна частина _____ канд. екон. наук, доцент кафедри ЕПВМ
 (підпис) _____ наук. ступінь, вчене звання (посада)
 “ ____ ” _____ 20__ р. _____ М. В. Бальзан
 ініціали та прізвище

Дата попереднього захисту роботи “ ____ ” _____ 20__ р.

Рецензент _____ канд. техн. наук, доц., доц. кафедри ПЗ
 (підпис) _____ наук. ступінь, вчене звання (посада)
 _____ О. М. Рейда
 ініціали та прізвище

Завдання видав науковий керівник _____ канд. техн. наук, доц., доц. кафедри КН
 (підпис) _____ наук. ступінь, вчене звання (посада)
 _____ О. К. Колесницький
 ініціали та прізвище

“ ____ ” _____ 20__ р.

Завдання отримав магістрант _____ І.Г. Кюльян
 (підпис) _____ ініціали та прізвище

“ ____ ” _____ 20__ р.

РЕФЕРАТ

Магістерська кваліфікаційна робота присвячена розробці методу класифікації звуків пострілів на основі спайкінгової нейронної мережі. Даний метод забезпечує автоматичну класифікації звуків пострілів та здатен класифікувати навіть сильно-корельовані звукові образи.

В ході роботи проведено аналіз предметної області класифікації звуків пострілів. Розглянуто відомі методи класифікації звуків та засоби аналогів для класифікації звуків. Удосконалено метод навчання нейронної мережі таким чином, щоб зменшити загальний час навчання. Розроблено метод класифікації звуків на основі спайкінгової нейронної мережі та реалізовано на мові програмування Kotlin програмний засіб для класифікації звуків пострілів на основі спайкінгової нейронної мережі.

ABSTRACT

The master's qualification work is dedicated to the development of the gunshots sounds classification method based on spiking neural network. This method provides automatic gunshots sounds classification and is able to classify even highly correlated sound images.

During the work was analyzed of the subject area of the gunshots classification. Known methods of the gunshots classification and existing analogues for gunshots classification were considered. Neural network training method was improved to reduce overall training time. A method of classifying gunshots sounds based on spiking neural network was developed and a software tool for classifying gunshots sounds based on spiking neural network was implemented using Kotlin programming language.

ЗМІСТ

| | |
|--|----|
| ВСТУП | 7 |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ КЛАСИФІКАЦІЇ ЗВУКІВ ПОСТРІЛІВ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОННОЇ МЕРЕЖІ | 12 |
| 1.1 Постановка задачі класифікації звуків пострілів..... | 12 |
| 1.2 Аналіз відомих методів класифікації звуків | 14 |
| 1.3 Аналітичний огляд існуючих програмних засобів класифікації звуків | 16 |
| 1.4 Висновок | 19 |
| 2 РОЗРОБКА МЕТОДУ КЛАСИФІКАЦІЇ ЗВУКІВ ПОСТРІЛІВ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОННОЇ МЕРЕЖІ | 20 |
| 2.1 Методи класифікації звуків..... | 20 |
| 2.1.1 Характеристики звуку..... | 20 |
| 2.1.2 Класифікація звуків за допомогою часового-частотного аналізу..... | 21 |
| 2.1.3 Класифікація звуків за допомогою методів машинного навчання. | 23 |
| 2.2 Обґрунтування вибору моделі нейронного елемента для спайкінгової нейронної мережі | 25 |
| 2.2.1 LIF модель..... | 27 |
| 2.2.2 Модель Ходжкіна-Хакслі. | 28 |
| 2.2.3 SRM модель. | 30 |
| 2.3 Структура та гіперпараметри спайкінгової нейронної мережі | 31 |
| 2.4 Методи навчання спайкінгової нейронної мережі | 33 |
| 2.4.1 Метод STDP – модифіковане правило Хебба. | 33 |
| 2.4.2 Навчання зі зворотнім поширенням помилки..... | 35 |
| 2.4.3 Керований метод навчання Хебба..... | 37 |
| 2.4.4 Метод ReSuMe..... | 38 |
| 2.4.4 Запропоноване удосконалення методу STDP. | 39 |
| 2.5 Попередня обробка аудіо сигналу та форматування вхідних даних для спайкінгової нейронної мережі..... | 41 |

| | |
|--|--|
| 2.6 Розробка методу класифікації звуків на основі спайкінгової нейронної мережі | 43 |
| 2.7 Метрики оцінювання якості класифікації | 43 |
| 2.8 Висновок | 46 |
| 3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ КЛАСИФІКАЦІЇ ЗВУКІВ ПОСТРІЛІВ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОННОЇ МЕРЕЖІ..... | 48 |
| 3.1 Обґрунтування вибору мови | 48 |
| 3.2 Обґрунтування вибору середовища програмування | 50 |
| 3.3 Розробка алгоритму функціонування програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі | 51 |
| 3.4 Реалізація програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі..... | 54 |
| 3.6 Тестування та аналіз результатів роботи програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі | 57 |
| 3.7 Висновок | 61 |
| 4 ЕКОНОМІЧНА ЧАСТИНА | 62 |
| 4.1 Оцінювання комерційного потенціалу розробки..... | 62 |
| 4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько-технологічної роботи..... | 63 |
| 4.3 Прогнозування комерційних ефектів від реалізації результатів розробки | 67 |
| 4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності.. | 68 |
| ВИСНОВКИ..... | 72 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 73 |
| Додаток А Інструкція користувача | Ошибка! Закладка не определена. |
| Додаток Б Лістинг програми..... | Ошибка! Закладка не определена. |
| Додаток В Графічна частина..... | Ошибка! Закладка не определена. |
| Додаток Г Довідка про впровадження | Ошибка! Закладка не определена. |

ВСТУП

Актуальність теми дослідження. У час інформаційних технологій штучний інтелект (ШІ) почав проникати майже у всі області людської діяльності. В теорії першочерговою метою досліджень ШІ є створення моделі мозку та моделювання процесу мислення. З практичної точки зору ШІ – це комп'ютерна система, що має певні ознаки інтелекту, а саме:

- здатна сприймати інформацію;
- здатна шукати спосіб досягнення певних результатів;
- спроможна самостійно приймати рішення;
- може навчатися та самовдосконалюватися.

Завдяки швидкому розвитку мікроконтролерів, появи дешевих датчиків та стрімкому розповсюдженню смарт-фонів кількість наявної аудіо інформації швидко зростає. Як результат, зростає необхідність в розвитку систем, спрямованих на обробку цієї інформації в автоматизованому режимі. Однією з важливих задач є задача класифікації звукових сигналів, яка важлива для сенсорних систем, які мають обробляти інформацію, що поступає з датчиків в реальному часі. Наразі існує велика кількість методів, що направлені на вирішення цієї задачі, проте не завжди вдається успішно проводити розпізнавання образів за допомогою обчислювальних засобів, заснованих на алгоритмічному функціонуванні. Тут на допомогу або навіть заміну традиційним методам приходять нейро-мережеві, основна відмінність яких полягає в тому, що з їх допомогою можна вирішувати неформалізовані завдання, для яких через ті або інші причини не існує детермінованих алгоритмів розв'язання.

Штучна нейронна мережа (ШНМ) – це математична модель, а також її програмна або апаратна реалізація, побудовані для імітації функціонування біологічних нейронних мереж – мозку. Це поняття з'явилося при вивченні процесів, які відбуваються в мозку, та при спробах моделювання цих процесів.

Дослідження моделювання ШНМ починалось за допомогою апаратних засобів. Проте стрімкий розвиток технологій та збільшення обчислювальних

можливостей сучасних комп'ютерних систем зробили можливим комп'ютерне моделювання ШНМ та відкрило нові шляхи для підвищення ефективності класифікації звукових сигналів із застосуванням нових модифікованих нейромережових алгоритмів.

Останнє десятиліття ознаменувалося бурхливим розвитком третього покоління штучних нейронних мереж – спайкінгових нейронних мереж. Це було викликано появою можливості розробляти і виробляти великі апаратні нейронні мережі, реалізовані у вигляді чипів, що дозволило говорити про можливість заміни у багатьох областях застосування традиційної архітектури комп'ютерів фон Неймана на нейро-мережову, засновану на спайкінгових нейронних мережах з низькою енергозатратністю.

Спайкінгова нейронна мережа – третє покоління ШНМ, яке відрізняється від бінарних (1 покоління) і аналогових (2 покоління) ШНМ тим, що в ньому нейронні елементи обмінюються короткими імпульсами однакової амплітуди. Вона є найбільш реалістичною, з точки зору фізіології, моделлю ШНМ [1].

Основним недоліком існуючих моделей спайкінгових нейронних мереж є відсутність досконалого алгоритму навчання. Це відкриває можливості до підвищення швидкості та точності існуючих алгоритмів. Тому дослідження теми досі залишається актуальним.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження магістерської кваліфікаційної роботи є зменшення часу навчання за методом STDP (Spike Timing Dependent Plasticity) спайкінгової нейронної мережі для класифікації звуків пострілів.

Для досягнення поставленої мети необхідно розв'язати такі наступні завдання:

- здійснити аналіз проблеми розв'язання задачі класифікації звуків пострілів;
- розглянути існуючі методи розв'язання задачі класифікації звуків пострілів та обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;
- розробити метод класифікації звуків пострілів на основі спайкінгової нейронної мережі;
- розробити структуру алгоритм роботи програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі;
- обґрунтувати вибір програмного інструментарію для розробки програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі;
- виконати програмну реалізацію запропонованого програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі;
- провести тестування програмного засобу та виконати аналіз отриманих результатів.

Об'єкт дослідження – це процес класифікації звуків пострілів на основі спайкінгової нейронної мережі.

Предмет дослідження – це методи та програмні засоби класифікації звуків пострілів.

Методи дослідження. У даній роботі було використано такі методи наукових досліджень: теорія штучних нейронних мереж, теорія розпізнавання образів, теорія моделювання систем, теорія машинного навчання, теорія паралельних обчислень.

Наукова новизна одержаних результатів полягає в наступному:

- удосконалено метод класифікації звуків пострілів за рахунок використання спайкінгової нейронної мережі, що дозволить проводити класифікацію з більшою точністю та у реальному часі;

- удосконалено метод навчання спайкінгової нейронної мережі оснований на модифікованому правилі Хебба STDP, за рахунок удосконалення розподілу вихідних збуджуючих нейронів по класах, тобто присвоєння нейронам міток класів, що дасть змогу уникнути перенавчання на початкових епохах, що в свою чергу дозволить суттєво скоротити швидкість навчання нейронної мережі;

Практичне значення одержаних результатів полягає у наступному:

1. Розроблено алгоритм класифікації звукових сигналів пострілів на основі спайкінгової нейронної мережі.

2. Розроблено структура спайкінгової нейронної мережі для класифікації звукових сигналів пострілів з модифікованою кількістю синаптичних зв'язків між шарами та в межах одного шару, а також модифікованим відношенням збуджувальних та гальмівних нейронів у кожному шарі

3. Розроблено алгоритм навчання спайкінгової нейронної мережі для підвищення швидкості навчання.

4. Розроблено програмний засіб для вирішення задачі класифікації звукових сигналів пострілів на основі спайкінгової нейронної мережі.

Розроблений програмний засіб був протестований та плануються до впровадження на ТОВ «ІНКОРСОФТ УКРАЇНА», про що є відповідна довідка – див. додаток Г.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки задач, коректністю застосуванням математичних методів під час дослідження, збіжністю результатів математичного моделювання з результатами тестування розробленої програмної реалізації та порівнянням результатів з відомими аналогами.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, здобуті самостійно. У роботах, опублікованих у співавторстві, магістранту належать такі результати: [1] – дослідження використання нейро-мережевих методів розпізнавання динамічних образів; [2] – експериментальні дослідження використання спайкінгової нейронної мережі для вирішення задачі розпізнавання образів; [3] – розробка моделі спайкінгової

нейронної мережі для вирішення задачі класифікації динамічних образів та модифікація алгоритму навчання STDP; [4] – програмна реалізація системи моделювання спайкінгової нейронної мережі для класифікації динамічних образів.

Апробація результатів роботи. Результати роботи були апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2019)», XLVIII науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії у березні 2019 м. Вінниці, XLVII науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії у березні 2018 м. Вінниці.

Публікації. За результатами досліджень отримано свідоцтво про реєстрацію авторського права на твір – комп'ютерна програма «Моделювання спайкінгової нейронної мережі» [4], опубліковано троє тез доповідей науково-технічних конференцій [1-3].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ КЛАСИФІКАЦІЇ ЗВУКІВ ПОСТРІЛІВ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОННОЇ МЕРЕЖІ

1.1 Постановка задачі класифікації звуків пострілів

Автоматична класифікація звуку – це сучасна область досліджень з численними програмними реалізаціями у реальному світі. Незважаючи на те, що існує велика кількість досліджень у суміжних сферах пов'язаних з аудіо, таких як розпізнавання мови та класифікація музики, робіт над класифікацією звуків навколишнього середовища порівняно мало [5].

Розв'язання задачі класифікація звуків є досить важливим для значного полегшення життя людей, наприклад за рахунок зменшення часу реакції служб у випадку небезпечних ситуацій, або впровадження певних систем «розумних» домашніх помічників, що реагують на певні типи звуків, наприклад, на дзвінок у двері, або дзвін розбитого вікна, або крик у під'їзді. Не менш важливою сферою застосування такої технології може бути збереження природи та захист тварин від бракон'єрів —так як система, що використовує аудіо інформацію, зібрану датчиками, що розташовані на певній природній області, що потребує охорони, може внести неоцінний вклад через вчасне виявлення і класифікацію таких звуків, як звуки пострілів або вибухів. Також системи класифікації звукових сигналів можуть знайти своє застосування в охоронних системах, які теж зможуть працювати більш надійно, завдяки обробці інформацію з мікрофонів, розташованих на території об'єкту, що охороняється. Інших можливих способів застосувати подібні технології безліч:

- Індексція та пошук мультимедії на основі контенту;
- Надання допомоги глухим особам у їхній щоденній діяльності;
- Промислові використання, такі як прогнозне обслуговування.

Спайкінгові нейронні мережі якнайкраще підходять для вирішення проблеми класифікації динамічних образів та мають ряд важливих переваг над нейронними мережами минулого покоління:

- є динамічними, а значить відмінно підходять для роботи з динамічними процесами (розпізнавання мови і динамічних зображень);
- підтримують багатозадачність, адже вхідні дані обробляються в нейронній мережі з зворотними зв'язками, а різні групи зчитуючих нейронів можуть бути навчені на рішення різних завдань;
- здатні здійснювати розпізнавання з передбаченням (тобто не обов'язково володіти повною інформацією про об'єкт);
- просто навчати, так як досить навчити тільки вихідні зчитувальні нейрони;
- мають підвищену продуктивність обробки інформації і стійкість перед перешкодами, так як використовують тимчасове надання інформації;
- вимагає меншого числа нейронів, так як спайкінговий нейрон замінює два нейрона (збуджувальний і гальмівний) класичної ШНМ;
- мають високу швидкість роботи і великий потенціал розпаралелювання, так як для передачі імпульсу необхідно відправити лише 1 біт;
- можуть навчатися в процесі роботи.

Спайкінгові нейронні мережі дають змогу визначити до якого екземпляру класифікації відноситься розпізнаваний образ ще до завершення його повної передачі. Це може значно економити час і обчислювальні потужності, без втрати точності. По даному принципу працює мозок людини. Люди можуть розпізнавати об'єкти по пам'яті, коли їм показують лише частину даного об'єкту. Так, наприклад, можна розпізнати пісню, тільки почувши стартову мелодію. Ще одним прикладом є розпізнавання малюнку людиною. Знаючи лише початкові штрихи, можна вгадати, як буде виглядати кінцевий малюнок, по досвіду спостерігання.

Існує два напрямки дослідження спайкінгових нейронних мереж:

1) створення комп'ютерних моделей, які точно повторюють моделі функціонування нейронів реального мозку, завдяки чому стане можливим як

пояснення механізмів його роботи, так і діагностика / лікування захворювань і травм центральної нервової системи;

2) створення комп'ютерних моделей, які абстрактно повторюють моделі функціонування нейронів реального мозку, що дозволить використовувати всі переваги реального мозку, таких як завадостійкість і енергоефективність, при аналізі великих обсягів даних.

В даній роботі будуть досліджуватися методи та засоби моделювання класифікації звуків пострілів на основі спайкінгової нейронної мережі, яка абстрактно повторює модель функціонування нейронів реального мозку. Буде розроблено програмний засіб для вирішення задачі класифікації звуків пострілів на основі спайкінгової нейронної мережі. Таке програмне забезпечення може стати корисним помічником як у повсякденному житті так і у професійній діяльності.

Користувачем даної програми є людина, котрій необхідно класифікувати звук пострілу. Програмний засіб для класифікації звуків пострілів на основі спайкінгової нейронної мережі повинен бути розроблений як додаток для системи linux_x64 та win_x64. Для класифікації звуку необхідно завантажити аудіофайл на носій та вказати шлях до нього класифікатору.

Вхідні дані можуть бути як аудіофайл розширення mp3 так і потік аудіо сигналів зі зчитуючого пристрою.

Вихідні дані – мітка класу, до якого класифікатор відніс вхідний аудіо запис.

1.2 Аналіз відомих методів класифікації звуків

Існуючі методи класифікації звуків можна розділити на такі види:

- з використанням перетворення Фур'є (ПФ);
- з використанням вейвлет-перетворення (ПВ);
- з використанням прихованих марковських моделей (ПММ);
- з використанням штучних нейронних мереж (ШНМ).

ПФ використовується в багатьох сферах науки, в тому числі і в технологіях обробки звуків. В області класифікації звуків ПФ розглядається як метод перетворення сигналу з часової області в частотну і розкладанням звуку на частотні складові. Проте даний метод не дає змогу детально аналізувати короткотривалі локальні особливості звуків, що є серйозним недоліком.

Останнім часом досить часто ВП використовують для обробки звукових сигналів. Вейвлетом називається деяка функція, яка добре локалізована як в часовій так і в частотній області. До даної функції застосовується дві операції: зсув (переміщення області локалізації в часі) та масштабування (переміщення області локалізації по частоті). Найбільш часто застосовувані вейвлети: вейвлет Хаара та вейвлет «Мексиканська шляпа». ВП має ряд переваг порівняно з ПФ, тому що дає змогу аналізувати короткотривалі локальні особливості звуків.

Одним зі статистичних методів є метод ПММ – модель, що імітує роботу процесу, схожого на марківський процес з невідомими параметрами. Головною метою ПММ є визначення невідомих параметрів на основі тих, що спостерігаються. Класифікація звуків базується на наступних припущеннях:

- сигнал може бути сегментований на фрагменти (стани), всередині яких сигнал може розглядатися як стаціонарний. Перехід між станами миттєвий;
- ймовірність появи сигналу породженого моделлю залежить тільки від поточного стану моделі.

Проте більшість статистичних методів програють в ефективності нейромережевим методам. ШНМ складається з набору штучних нейронів поєднаних один з одним синаптичними зв'язками (гальмуючими та збуджувальними), що характеризуються вагами w_{ij} , що з фізичної точки зору еквівалентний електричній провідності. Нейрон працює як суматор, тобто його стан визначається сумою усіх входів. На виході нейрон має функцію активації. Для використання ШНМ в задачах класифікації звуків необхідно побудувати структуру мережі, навчити її множині звуків, підібрати вагові коефіцієнти для мінімізації кількості помилок.

1.3 Аналітичний огляд існуючих програмних засобів класифікації звуків

Існують рішення, що базуються на методі трансформації аудіо сигналів в спектограми та подальшій обробці візуальних зображень за допомогою штучної нейронної мережі (багатошаровий перспетрон, згорткові нейронні мережі). Одним з таких є рішення запропоноване Майклом Смолсом [6].

Він розробив архітектуру на основі багатошарового персептрона використовуючи мову програмування Python та фреймворк для машинного навчання Tensorflow, а для побудови моделі використав фреймворк Keras (рис 1.1).

| Layer (type) | Output Shape | Param # |
|---------------------------|--------------|---------|
| dense_1 (Dense) | (None, 256) | 10496 |
| activation_1 (Activation) | (None, 256) | 0 |
| dropout_1 (Dropout) | (None, 256) | 0 |
| dense_2 (Dense) | (None, 256) | 65792 |
| activation_2 (Activation) | (None, 256) | 0 |
| dropout_2 (Dropout) | (None, 256) | 0 |
| dense_3 (Dense) | (None, 10) | 2570 |
| activation_3 (Activation) | (None, 10) | 0 |
| Total params: 78,858 | | |

18

Рисунок 1.1 – Структура нейронної мережі Udacity-ML-Capstone

Приклад класифікації даною програмою зображено на рисунку 1.2. На вхід нейронної мережі подається зображення частотної мел-спектрограми розмірністю 256 на 256 пікселів. В даному програмному засобі можливо виконати класифікацію за такими класами:

- шум дітей;

- гавкіт собаки;
- вулична музика;
- відбійний молоток;
- звук двигуна автомобіля;
- звук кондиціонеру;
- звук дрилі;
- звук сирени;
- автосигнал;
- звук пострілу.

```
# Class: Air Conditioner

filename = '../UrbanSound Dataset sample/audio/100852-0-0-0.wav'
print_prediction(filename)

The predicted class is: air_conditioner

# Class: Drilling

filename = '../UrbanSound Dataset sample/audio/103199-4-0-0.wav'
print_prediction(filename)

The predicted class is: drilling

# Class: Street music

filename = '../UrbanSound Dataset sample/audio/101848-9-0-0.wav'
print_prediction(filename)

The predicted class is: street_music

# Class: Car Horn

filename = '../UrbanSound Dataset sample/audio/100648-1-0-0.wav'
print_prediction(filename)

The predicted class is: car_horn
```

Рисунок 1.2 – Скріншот програми Udacity-ML-Capstone

Як видно, це рішення має широкі можливості щодо класифікації різних типів звуків. Можливо відрізнити звук сирену від звуку пострілу. Проте недоліком даної програми є те, що вона не здатна класифікувати сильно-корельовані образи, такі як звуки пострілу мало каліберної зброї та багато каліберної зброї, або постріл з міномету та постріл з танку.

Ще один програмний засіб, розробником якого є Тушар Сінгх. Він пропонує автоматизовану класифікацію звуків пострілів, таких як [7]:

- звук гвинтівки;
- звук рушниці;
- звук пістолету.

Дане рішення базується на схожому принципі з минулим. Написано на мові програмування Python з використанням Tensorflow та Keras. Розроблена штучна нейронна мережа отримує на вхід зображення отримане трансформацією звукового сигналу у спектограму. Приклад класифікації даною програмою зображено на рисунку 1.3.

```
filename = '../gun_shots/dataset/audio/handgun-001.wav'  
  
result = classify(filename)  
  
Result:  
handgun - 88 %  
shotgun - 10 %  
rifle - 2 %
```

Рисунок 1.3 – Скріншот роботи програми Gunshot-sound-classification

Програма пропонує швидкий та зручний спосіб класифікувати звук, проте достовірність класифікації не велика мала (менше 90%) .

1.4 Висновок

Проаналізувавши існуючі програмні засоби для класифікації звуків, можна сказати, що не існує досконалого засобу, який дозволить з достатньою точністю (близько 95%) та швидкістю класифікувати сильно-корельовані звуки пострілів.

Аналіз сучасних методів класифікації звуків показав, що одним з найефективніших методів є класифікацію з використанням штучних нейронних мереж та машинного навчання.

Спайкінгові нейронні мережі показали значну перевагу над ШНМ минулого покоління в плані точності і швидкодії, що відкрило можливості до удосконалення існуючих алгоритмів класифікації звуків. Найкраще даний тип ШНМ підходить для задач, де важливо виконувати задачі в обмеженому часі. Тобто, для задач в яких важливий не кінцевий результат, з яким зв'язаний образ розпізнавання, а зміна вектору розпізнавання, в ту сторону, в яку змінюються зв'язані з ним характеристики в часі. Таким чином, не знаючи завершеного образу, а знаючи тільки зміну вектору його параметрів, можна визначити, до якого образу відноситься екземпляр.

Тому було вирішено розробити програмний засіб для класифікації звуків пострілів саме на основі спайкінгової нейронної мережі.

2 РОЗРОБКА МЕТОДУ КЛАСИФІКАЦІЇ ЗВУКІВ ПОСТРІЛІВ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОННОЇ МЕРЕЖІ

2.1 Методи класифікації звуків

2.1.1 Характеристики звуку.

Існує досить велика кількість виділених характеристик звуку, що можуть використовуватись для роботи з із звуковими сигналами: [8]

1) Спектральні ознаки:

- середнє значення спектра сигналу;
- нормалізоване середнє значення спектра;
- медіанне значення спектра сигналу в смугах;
- відносний час перебування сигналу в смугах;
- нормалізований час перебування сигналу в смугах;
- відносна потужність спектра сигналу в смугах;

2) Часові ознаки:

- тривалість аудіо сегменту;
- висота аудіо сегменту;

3) Амплітудно-частотні ознаки:

- енергія;
- інтенсивність;
- амплітуда;
- частота основного тону;
- джиттер (небажані фазові та/або частотні випадкові спотворення під

час передачі сигналу);

- радіальна базисна функція.

4) Нелінійні ознаки:

- відображення Пуанкаре;
- рекурентний графік;
- максимальний характеристичний показник Ляпунова;

- фазовий портрет;

Спектральні і часові ознаки характеризують сигнал в його фізико-математичної суті за допомогою аналізу трьох видів ділянок:

- 1) Періодичних ділянок звукової хвилі;
- 2) Неперіодичних ділянок звукової хвилі – шумів;
- 3) Ділянок без пауз.

Амплітудно-частотні ознаки дають змогу отримувати значення, що можуть змінюватися залежно від параметрів перетворення Фур'є.

Нелінійні ознаки розглядають мовний сигнал як скалярну величини, яка спостерігається в процесі мовлення людини.

Більшість сучасних автоматичних систем розпізнавання мови зосереджують зусилля на отриманні частотної характеристики мовного тракту людини, відкидаючи при цьому характеристики сигналу збудження. Це пояснено тим, що коефіцієнти першої моделі забезпечують кращу роздільність звуків.

2.1.2 Класифікація звуків за допомогою часового-частотного аналізу.

Перетворення Фур'є – це наразі найвідоміший метод обробки цифрових сигналів. Однак цей метод має ряд недоліків, що спонукає до розробки цілком нових або удосконалених відомих цифрового аналізу нестационарних сигналів. Найбільшим недоліком перетворення Фур'є можна назвати усереднення характерних рис по всій тривалості аудіо сигналу, що робить неможливим застосування цього методу при аналізі змін сигналу в часі. Для аналізу швидкоплинних нестационарних сигналів доцільно використовувати сучасний математичний засіб – вейвлет-перетворення (ВП) [9].

ВП сьогодні широко застосовується для аналізу аудіо сигналів та інших динамічних даних. Цей метод дозволяє отримати частотно-часове відображення сигналу. Безперервне ВП – це розклад сигналу в базисі деякої материнської вейвлетної функції. Базис вейвлет-перетворення (ВП) будується шляхом масштабних перетворень та переносів материнського вейвлету з неперервними

значеннями базисних параметрів – масштабного коефіцієнту a та параметра зсуву b :

$$\Psi_{a,b}(t) = |a|^{-1/2} \Psi\left(\frac{t-b}{a}\right) \quad (2.1)$$

де:

$\Psi_{a,b}(t)$ – дочірній вейвлет базису;

t – параметр часу;

$a, b \in \mathbb{R}$.

В інтегральній формі ВП безперервного сигналу $s(t)$ може бути подано як:

$$W(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} s(t) \Psi_{a,b}^*(t) dt \quad (2.2)$$

де $*$ – оператор комплексного спряження. Використовуючи рівняння Парсеваля, що описує зв'язок між функціями та їхніми образами Фур'є, вираз може бути записаний в альтернативній формі:

$$W(a, b) = \sqrt{a} F^{-1}\{S(f) \Psi^*(af)\} \quad (2.3)$$

де:

F^{-1} – оператор зворотного перетворення Фур'є;

$\Psi^*(af)$ – образ Фур'є дочірнього вейвлету;

$S(f)$ – образ Фур'є сигналу;

f – частота сигналу.

Дане подання дає змогу зменшити час виконання вейвлет-коефіцієнтів за рахунок використання швидкого перетворення Фур'є. Формула 3 показує, що ВП може бути розглянуте як спеціальна операція фільтрації з вейвлетом в якості фільтра. Параметри обраного вейвлету та його налаштування можуть суттєво

впливати на результати перетворення й можливість їх правильної інтерпретації [10].

Серед існуючих вейвлет-функцій, що використовуються в безперервному вейвлет-аналізу, вейвлети сімейства Гауса, а саме вейвлет Морле та вейвлет Мексиканський капелюх (рис. 2.1), більш за все подібні за формою імпульсним складовим нестационарних сигналів. Цей факт робить Гаусове сімейство вейвлетів найбільш відповідним методом аналізу таких сигналів.

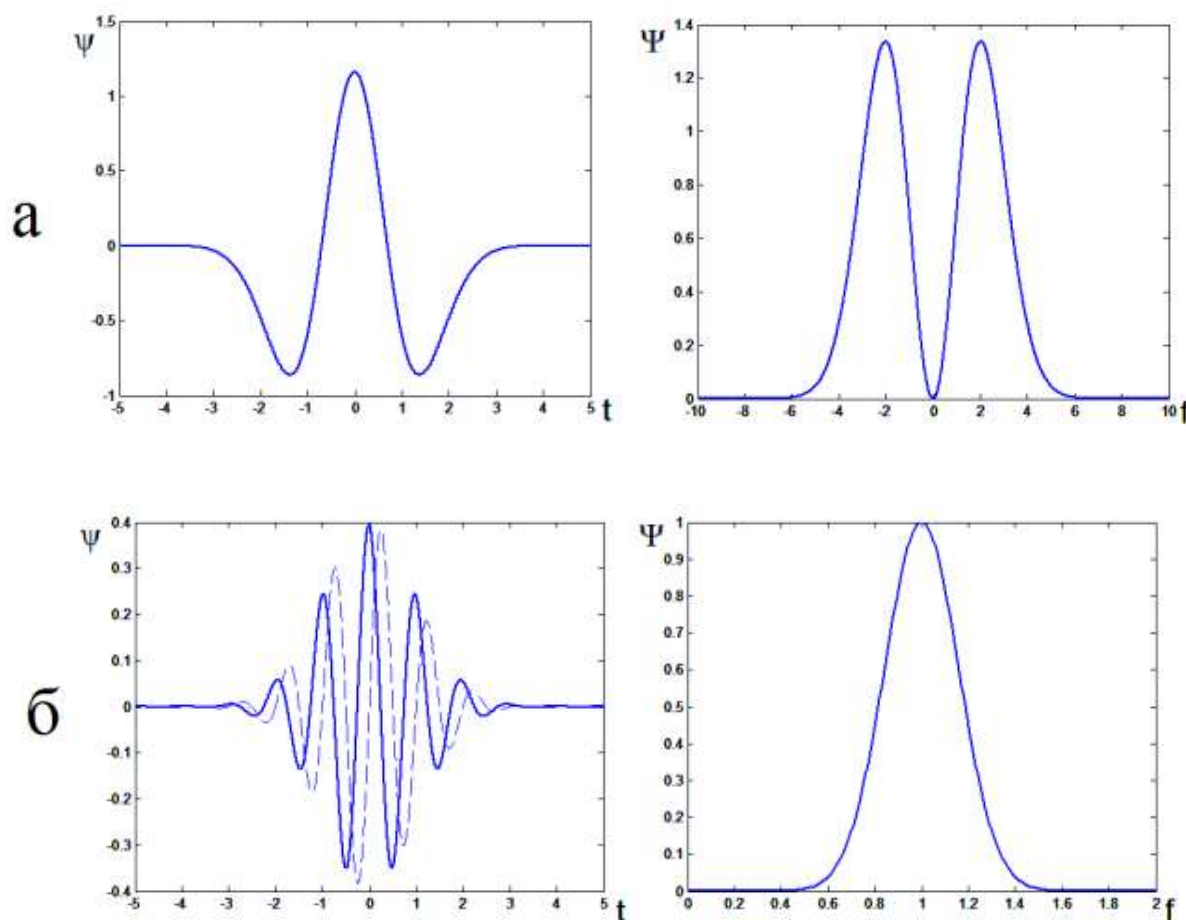


Рисунок 2.1 – Вейвлети Морле (а) та Мексиканський капелюх (б)

2.1.3 Класифікація звуків за допомогою методів машинного навчання.

Машинне навчання (МН) – великий підрозділ штучного інтелекту, що вивчає методи побудови алгоритмів, здатних навчатися. У найзагальнішому випадку розрізняють два типу машинного навчання: навчання по прецедентах, або індуктивне навчання, і дедуктивне навчання. Оскільки останнє прийнято

відносити до області експертних систем, то терміни «машинне навчання» і «навчання по прецедентах» можна вважати синонімами. Цей метод навчання зараз, як прийнято говорити, в тренді, а ось експертні системи переживають кризу. Бази знань, що лежать в їх основі, важко узгоджувати з реляційною моделлю даних, тому промислові СУБД неможливо ефективно використовувати для наповнення баз знань експертних систем.

Навчання по прецедентах, в свою чергу, поділяють на три основних типи: контрольоване навчання, або навчання з учителем (supervised learning), неконтрольоване навчання (unsupervised learning), або навчання без учителя, і навчання з підкріпленням (reinforcement learning). [11]

МН успішно застосовується в задачах пошуку, комп'ютерного зору, медицини, безпілотного керування роботів та автомобілів. Основою для багатьох з цих додатків є методи та засоби штучного інтелекту, такі як класифікація, локалізація і детектування на основі штучних нейронних мереж.

ШНМ є системою з'єднаних і взаємодіючих між собою штучних нейронів, виконаних на основі порівняно простих нейронних елементів, що імітують біологічний нейрон з певною достовірністю. Кожен елемент мережі періодично отримує сигнали від одних елементів і періодично посилає сигнали іншим. Всі разом ці прості нейронні елементи, з'єднані в мережу, здатні вирішувати досить складні завдання.

Найчастіше нейрони розташовуються в мережі за рівнями (їх ще називають шарами). Нейрони першого рівня називаються вхідними. Вони отримують дані ззовні і після їх обробки передають імпульси через синаптичні зв'язки нейронам на наступному рівні. Нейрони на другому рівні (його називають прихованим, оскільки він безпосередньо не пов'язаний ні з входом, ні з виходом ШНМ) обробляють отримані імпульси і передають їх нейронам на вихідному рівні. Оскільки мова йде про імітацію біологічних нейронів, то кожен нейронний елемент вхідного рівня пов'язаний з декількома елементами прихованого рівня, кожен з яких, в свою чергу, пов'язаний з декількома нейронами рівня вихідного.

Це є базовий принцип побудови архітектури ШНМ, що здатна до навчання і може знаходити прості закономірності в даних.

Класифікація – один з розділів машинного навчання, присвячений розв'язанню задачі розділення деяким чином множини об'єктів на класи. Класифікувати об'єкт означає віднести об'єкт до певного класу. Задана множина об'єктів, для яких відомо, до якого класу вони відносяться. Ця множина називається навчальною вибіркою. Класова приналежність інших об'єктів не відома. Вимагається побудувати алгоритм, здатний класифікувати будь-який об'єкт початкової множини [12].

Можна виділити два типи алгоритмів класифікації за типом вхідних даних:

1) За ознаками – кожен об'єкт має сукупність своїх характеристик, що називаються ознаками (features).

2) Безпараметричні – кожен об'єкт описується певною формулою відношення до інших об'єктів навчальної вибірки. Найвідоміші представники цього типу: метод найближчих сусідів, метод парензивного вікна, метод потенційних функцій.

2.2 Обґрунтування вибору моделі нейронного елемента для спайкінгової нейронної мережі

Загалом процес передачі імпульсного сигналу (рис. 2.2) можна описати так: потенціали дій рухаються вздовж вихідних аксонів через синапси до вхідних дендритів наступного елемента. Синапси випускають нейромедіатор, який швидко поширюється на пост-синаптичний нейрон. У пост-синаптичному нейроні ці нейротрансмітери впливають на потенціал мембрани нейрона.

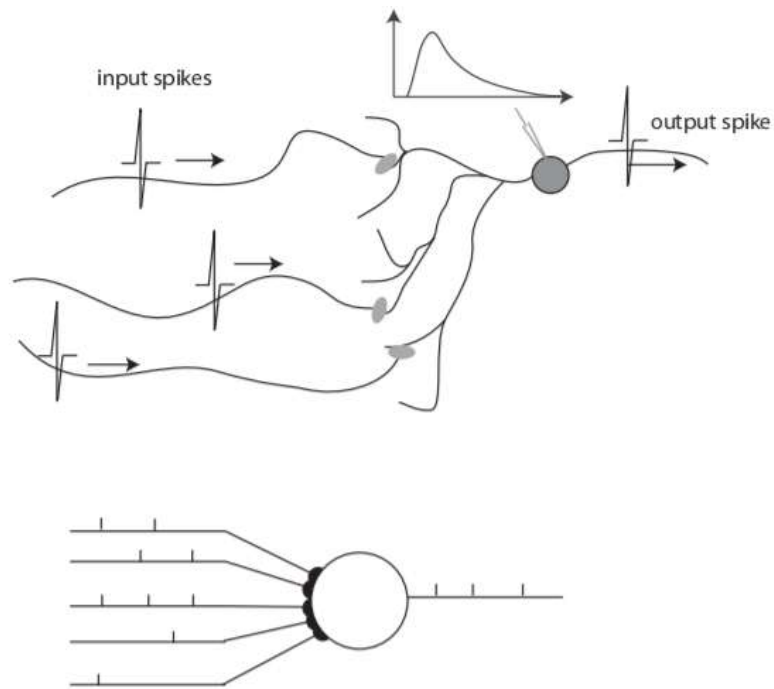


Рисунок 2.2 – Схематичне відображення як імпульси надходять з інших нейронів на синапси пост-синаптичного нейрона.

Схематичне відображення внутрішнього потенціалу нейрона у відповідь на входні імпульси зображене на рисунку 2.3. Коли досягається порогове значення θ , мембранний потенціал активується та скидається до нижнього значення, після чого генерується вихідний імпульс.

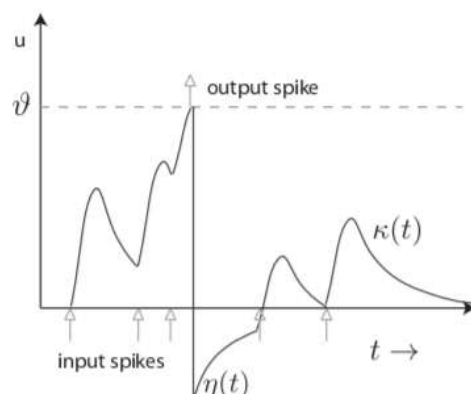


Рисунок 2.3 – Схематичне відображення внутрішнього стану нейрона.

Вищезгадане – це суттєве спрощення, і реальні нейрони відображають багато різних розбіжностей поведінки: деякі відповідають на вхід тільки після затримки, інші відповідають з цілою чергою імпульсів. Для того, щоб пояснити таку поведінку нейронів, були розроблені детальні формальні моделі. Відомі приклади моделей формальних нейронів – модель LIF (Leaky Integrate-and-Fire Model) та SRM (Spike Response Model). Однією з найвідоміших та найскладніших моделей вважається модель Ходжкіна-Хакслі [13].

Існують і більш складні моделі, проте загалом все зводиться до певного компромісу між нейронауковим реалізмом та потужністю обчислювальної техніки, яка необхідна на для моделювання. Нище наведено аналіз найбільш відомих формальних моделей нейронного елемента.

2.2.1 LIF модель.

LIF нейрон є, мабуть, найвідомішим прикладом формальної моделі імпульсного нейрона. Всі LIF нейрони можуть або стимулюватися зовнішнім імпульсом або синаптичним входом від пресинаптичних нейронів [14].

Кожен попередній синаптичний імпульс робить внесок, який описується функцією $\alpha(t)$, до постсинаптичного струму, а внески різних попередніх синаптичних імпульсів лінійно підсумовуються для отримання загального постсинаптичного струму. Таким чином, загальний постсинаптичний нейрон може бути описаний наступним чином:

$$I_i(t) = \sum_j w_{ij} \sum_f \alpha(t - t_j^{(f)}) \quad (2.4)$$

де:

$t_j^{(f)}$ – показує час f-го імпульсу j-го пресинаптичного нейрона;

w_{ij} – величина ваги синапсу між i-им та j-им нейроном.

На рисунку 2.4 відображено симуляцію LIF нейрона з 25 входами, де можна побачити графік постсинаптичного потенціалу, частоту пресинаптичних

імпульсів, та графік загального постсинаптичного струму. В даній симуляції ваги синапсів були обрані випадково з Гауссівського розподілу з середнім значенням 1.6мВ та відхиленням 0.5мВ. Поріг мембрани – 20мВ.

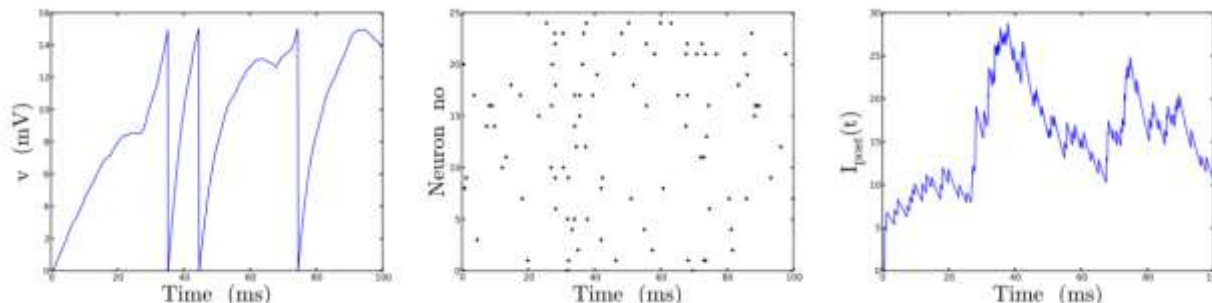


Рисунок 2.4 – Симуляція LIF нейрона.

2.2.2 Модель Ходжкіна-Хакслі.

Ходжкін і Хакслі запропонували досить складну систему рівнянь. Вона складається з чотирьох пов'язаних нелінійних диференціальних рівнянь, тому їх дуже важко математично аналізувати. Через це було запропоновано кілька спрощень рівнянь Ходжкіна-Хакслі. Найбільш відомий підхід спрощує набір з чотирьох диференціальних рівнянь до двовимірної задачі. В такому випадку можна застосувати дещо інший підхід, щоб зменшити чотири рівняння Ходжкіна-Хакслі до однієї змінної $u(t)$ – мембранного потенціалу нейронного елементу. Зокрема, ми встановимо апроксимаційне відображення між моделлю SRM та моделлю Ходжкіна-Хакслі.

Математичну модель Ходжкіна-Хакслі можна представити як окремий випадок SRM рівняння:

$$u_i(t) = \eta(t - \hat{t}) + \int_0^{t-\hat{t}} k(t - \hat{t}, s) I(t - s) ds + u_{rest} \quad (2.5)$$

Для побудови апроксимаційне відображення між SRM та рівняннями Ходжкіна-Хакслі, треба визначити три наступні терміни:

- 1) функцію η , яка описує відповідь на випуск імпульсу;

- 2) функцію k , яка описує відповідь на вхідний струм;
- 3) значення порогу ϑ .

Функція η дозволяє нам описати стандартну форму імпульсу та післяпотенційного імпульсу:

$$\eta(t - \hat{t}) = [u(t) - u_{rest}] \quad (2.6)$$

На рисунку 2.5 зображено графік потенціалу в моделі Ходжкіна-Хакслі з визначеною функцією η . Вхідний імпульс був зафіксований у момент часу $t = 0$.

Функція k характеризує лінійну відповідь нейрона на слабкий імпульс вхідного струму. Для вимірювання k використовується перший сильний імпульс для генерації імпульсу за часом $\hat{t} < 0$, а потім застосовується другий слабкий імпульс при $t = 0$.

Оскільки імпульс вхідного струму забезпечує заряд за дуже короткий проміжок часу, функція k миттєво підскакує в момент $t = 0$ до значення 1 мВ. Потім розтікається, повертаючись до нуля з невеликими коливаннями.

$$k(t - \hat{t}, t) = \frac{1}{c} [u(t) - \eta(t - \hat{t}) - u_{rest}] \quad (2.7)$$

Третій термін, який необхідно визначити – це порогове значення ϑ , яке необхідно обрати як фіксоване. Незважаючи на те, що існують припущення, що рівняння Ходжкіна-Хакслі мають певну форму порогової поведінки, поріг не є чітко визначеним, і взагалі досить складно оцінити поріг напруги безпосередньо з однієї серії симуляцій. Тому його можливо визначати як вільний параметр, який буде скорегований під час наступних моделювань.

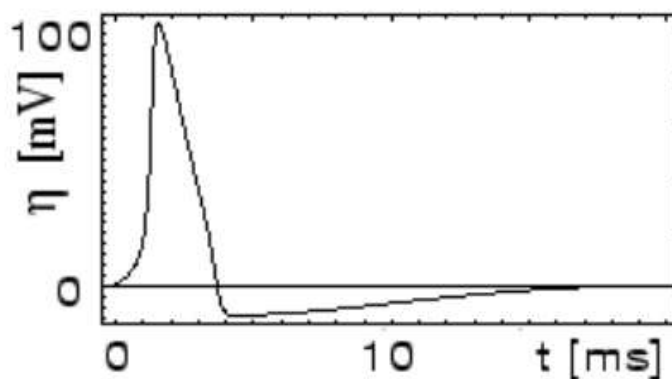


Рисунок 2.5 – Потенціал дії моделі Ходжкіна-Хакслі з функцією η .

2.2.3 SRM модель.

Модель Spike Response Model – є узагальненим випадком LIF моделі. Проте напрямком узагальнення є дещо іншим. У LIF моделі параметри є залежними від напруги, а в SRM вони залежать від часу минулого імпульсу. Також LIF та SRM відрізняються формулюванням рівнянь.

У рамках моделі SRM стан нейрона описується однією змінною u_i . За відсутності імпульсів, змінна u_i знаходиться в стані спокою $u_{rest} = 0$. Кожен вхідний імпульс збуджує нейрон і u_i зростає, після цього має пройти деякий час, перш ніж u_i повернеться до u_{rest} . Функція ϵ визначає часове зміщення у відповідь на вхідний імпульс. Якщо після підсумовування ефектів декількох вхідних імпульсів, u_i досягає порогу ϑ , генерується вихідний імпульс. Форма потенціалу дії та після потенціалу описується функцією η . Припустимо, якщо нейрон згенерував свій останній імпульс у час t_i , то після генерації імпульсу u_i можна описати як:

$$u_i(t) = \eta(t - t_i) + \sum_j w_{ij} \sum_f \epsilon_{ij}(t - t_j^{(f)}) \quad (2.8)$$

де:

$t_j^{(f)}$ – показує час f-го імпульсу j-го пре-синаптичного нейрона;

w_{ij} – величина ваги синапсу між i-им та j-им нейроном.

На рисунку 2.6 зображено схематичний приклад роботи SRM нейрона зі сталим порогом. Кожен вхідний імпульс викликає збуджуючий постсинаптичний потенціал. Усі вхідні потенціали сумуються. Якщо загальний потенціал досягає порогу, то відбувається активація нейрону, генерується вихідний імпульс та напруга скидається до стану спокою.

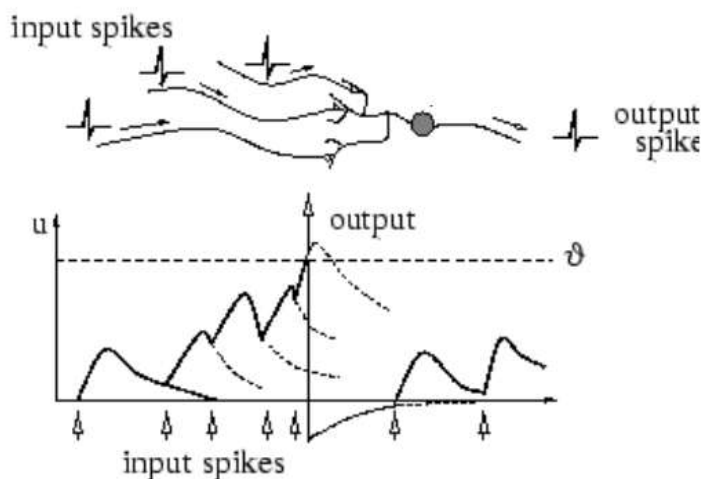


Рисунок 2.6 – Симуляція SRM нейрона.

2.3 Структура та гіперпараметри спайкінгової нейронної мережі

Існує загально прийнята структура спайкінгової нейронної мережі, що складається з трьох шарів: вхідний шар, прихований шар та вихідний шар. Структури відрізняються здебільшого кількістю нейронів в шарах та кількістю гальмівних та збуджувальних зв'язків. Спайкінгова нейронна мережа функціонує як просторово-часовий суматор. Кожний прихований нейрон накопичує вхідні імпульси до тих пір, поки не досягне порогового значення. Після цього йде спрацювання виходу нейрона і його імпульс передається далі по синаптичним зв'язкам [15].

На вхідний шар мережа отримує потокові дані, які в реальному часі передаються далі всередину мережі і збуджують конкретні нейрони в прихованому шарі, що в результаті призводить до спрацювання вихідних нейронів, коли вони накопичують порогове значення імпульсу – це є відповіддю

мережі. Залежно від того який вихідний нейрон частіше спрацьовував, до такого класу і буде віднесено вхідний образ.

Прийmemo таку кількість нейронів: у вхідному шарі – 5; у прихованому шарі $n = 150$; у вихідному шарі – 4. Кожен вхідний нейрон з'єднаний з 30% нейронів прихованого шару. У внутрішньому шарі кожен нейрон з'єднаний з 10% нейронами цього шару. А кожен нейрон вихідного шару з'єднаний з 20% нейронами прихованого шару. Кількість збуджувальних з'єднань та гальмівних 80% та 20% відповідно. Поріг нейрона – 1. Значення імпульсу підчас активації – 0.05. Максимальна можлива частота генерації імпульсів 300 за 1с, що є середнім значенням для біологічного нейрону. Структура спайкінгової нейронної мережі схематично представлена на рисунку 2.7.

Гіперпараметри навчання: η (швидкість навчання) – 0.001, k (кількість епох навчання) – 100.

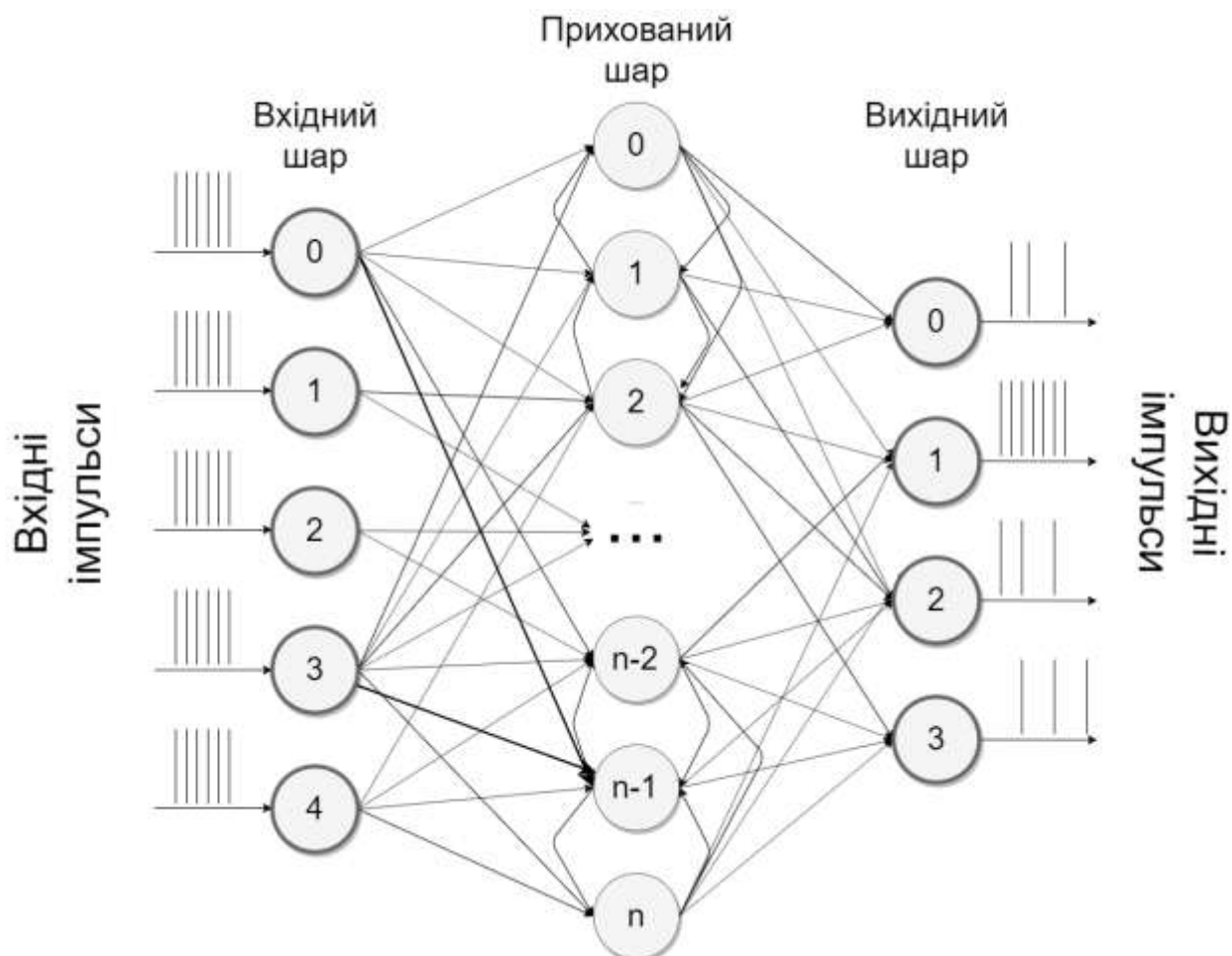


Рисунок 2.7 – Структура спайкінгової нейронної мережі

2.4 Методи навчання спайкінгової нейронної мережі

Можливі два підходи до вибору методів навчання спайкінгових нейронних мереж:

1) спробувати перенести відомі методи навчання класичних нейронних мереж на імпульсні нейронні мережі (наприклад, метод навчання зворотним поширенням помилки);

2) створювати нові методи і алгоритми з урахуванням особливостей спайкінгових нейронних мереж.

Можна виділити такі основні види методів навчання спайкінгових нейронних мереж:

- метод STDP (модифіковане правило Хебба);
- навчання зворотнім поширенням помилки;
- керований метод навчання Хебба (supervised Hebbian learning);
- метод ReSuMe;
- глибоке навчання (deep learning).

Проведемо аналіз методів навчання спайкінгових нейронних мереж для того, щоб обрати найбільш підходящий для створення нейронної мережі в даній роботі.

2.4.1 Метод STDP – модифіковане правило Хебба.

Як відомо, запропоноване Хеббом в 1940х роках правило навчання засноване на нейрофізіологічних даних і відповідає поведінці біологічних нейронних мереж, що зв'язки між нейронами зі схожими станами підсилюються, а між різними станами слабшають. У модифікації цього правила для спайкінгових нейронних мереж, в методі STDP (spike timing dependent plasticity) зміна ваг синапсів нейрона залежить від різниці в часі між пре-синаптичним і пост-синаптичним спайком, тобто від передбачуваної кореляції між імпульсом, який прийшов на синапс нейрона, та вихідним імпульсом згенерований цим же

нейроном. Залежність величини зміни ваги синапсу від різниці у часі між пост-синаптичних і пре-синаптичним спайком представлена на рисунку 2.8.

В [16] описано приклад застосування методу STDP для навчання спайкінгової нейронної мережі для розпізнавання чорно-білих (у відтінках сірого) зображень десяткових цифр з бази даних MNIST. Роздільна здатність кожного зображення 28 на 28 пікселів перетворюється у випадкову послідовність імпульсів, ймовірність появи яких в кожному такті пропорційна інтенсивності пікселя. Послідовність від кожного пікселя є вхідною інформацією для кожного з безлічі збуджувальних нейронів. Кожен зі збуджуючих нейронів пов'язаний по виходу з одним гальмуючим нейроном. Кожен з гальмівних нейронів пов'язаний з виходом з усіма збуджувальними нейронами крім того, який йому відповідає.

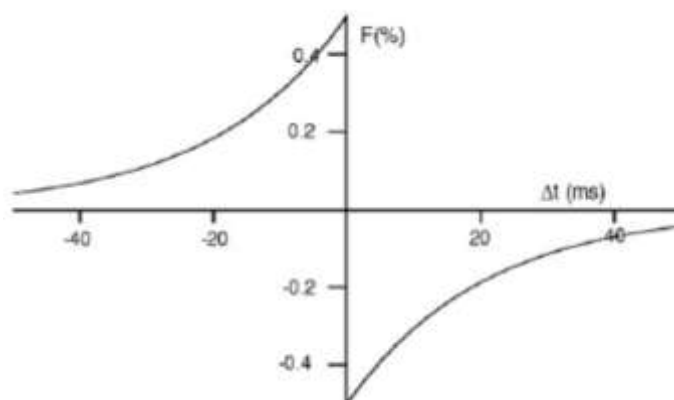


Рисунок 2.8 Залежність величини зміни ваги синапсу від різниці у часі між пост-синаптичних і пре-синаптичним спайком

Таким чином, гальмуючі нейрони реалізують латеральні зв'язки між збуджуючими нейронами. Класи, що розпізнаються, в процесі навчання не пред'являються мережі, тобто використовується навчання без учителя. Збуджуючі нейрони зв'язуються з класами після навчання, яке базується на виявленні максимально в середньому активного нейрона, що відповідає певному класу. Ніякі додаткові параметри не використовуються для передбачення класу, зокрема, будь-які лінійні класифікатори або подібні методи на виході мережі.

Крім синаптичної ваги кожен синапс зберігає параметр x_{pre} , який моделює пре-синаптичну історію приходу імпульсів. У кожному такті, якщо на синапс приходять імпульси, то x_{pre} збільшується на 1, а в іншому випадку експоненціально зменшується. Коли спрацьовує нейрон, зміна ваги його кожного синапсу Δw обчислюється на основі історії активності синапсу як:

$$\Delta w = \eta(x_{pre} - x_{tar})(w_{max} - w)^\mu \quad (2.9)$$

де:

η – коефіцієнт, що задає швидкість зміни ваги синапса;

w_{max} – максимальна величина ваги;

μ – задає залежність зміни ваги від попередньої ваги,

x_{tar} – цільове значення параметра x_{pre} при спрацьовуванні нейрона.

Формула (2.9) гарантує, що той синапс, який рідко призводить до спрацьовування нейрона, буде все більше і більше знижувати свою вагу. Розподіл збуджуючих нейронів по класах (присвоювання класам, що розпізнаються міток) проводиться після навчання шляхом виявлення найбільш активного нейрона при пред'явленні нейронній мережі типового представника відповідного класу.

Недоліком застосування модифікованого правила Хебба у вигляді методу STDP є те, що він забезпечує тільки навчання без учителя, хоча відомі спроби комбінування STDP з навчанням зі зворотнім поширенням помилки.

2.4.2 Навчання зі зворотнім поширенням помилки.

Метод зворотного поширення помилки (backpropagation) — це ітеративний градієнтний алгоритм, для вирішення задачі мінімізації помилки роботи штучної нейронної мережі. Основна ідея методу полягає в зворотному поширенні значень помилки від виходів мережі до її входів. Вимогою для використання методу

зворотного поширення помилки є бути диференційованість функція активації нейронів.

Навчання ШНМ можна подати як задачу оптимізації. Для того, щоб оцінити правильність роботи нейронної мережі, будується функція оцінки. Найпоширеніша функція оцінки — сума квадратів відстаней вихідних сигналів мережі від їх необхідних значень:

$$H = \frac{1}{2} \sum_{\tau \in \vartheta_{out}} (Z(\tau) - Z^*(\tau))^2 \quad (2.10)$$

де:

$Z^*(\tau)$ – необхідне значення вихідного сигналу.

Основною проблемою при застосуванні методу зворотного поширення помилки для спайкінгових нейронних мереж є бінарний характер входів та виходів спайкінгових нейронних елементів. Тому більшість градієнтних методів навчання не може бути використані безпосередньо для спайкінгових нейронних мереж.

Проте, на початку 2000 років був запропонований алгоритм навчання спайкінгової нейронної мережі SpikeProp, заснований на зворотному поширенні помилки. Автори використовували в якості безперервних змінних часи появи спайки на входах та виходах нейронів, а в якості помилки використовували різницю між очікуваним і реальним часом появи імпульсу [17].

Інший підхід, заснований на використанні алгоритму зворотного поширення помилки, був запропонований нещодавно для навчання нейрочипа TrueNorth компанії IBM. У цьому підході автори використовують оффлайн навчання на комп'ютері та пропонують для цього замінити вхідні та вихідні імпульси можливостями їх появи, що дозволяє застосувати градієнтний метод навчання [18].

Таким чином, навчання відповідно до методу зворотного поширення помилки виконується ітеративно у вигляді наступних кроків:

- 1) проходження вхідних сигналів через мережу від вхідного шару до вихідного шару;
- 2) порівняння виходів мережі з бажаними виходами і обчислення функції помилки;
- 3) зворотне поширення помилки з обчисленням для кожного синапсу проміжних шарів градієнта функції помилки;
- 4) використання отриманого градієнта для зміни параметрів синапсів.

Після навчання нейронна мережа з отриманими в процесі навчання значеннями ймовірностей синаптичних зв'язків в ядрі TrueNorth перетворюється в спайкінгову мережу для розміщення в чипі. При цьому для кожного класу, що розпізнається, створюється ансамбль з декількох нейронних мереж, в яких випадково формується матриця синаптичних зв'язки $c_{ij} = (0, 1)$ в ядрі так, щоб значення 0 і 1 в цій матриці по всій мережі відповідали отриманим в процесі навчання можливостям зв'язків. Ця матриця задає наявність зв'язків між i -м синапсом і j -м нейроном. Сумарний сигнал на j -му нейроні обчислюється за формулою:

$$I_j = \sum_i x_i c_{ij} s_{ij} + b_j \quad (2.11)$$

де:

x_i - вхідний сигнал на i -м синапсі,

s_{ij} - вага (сила) i -го синапсу стосовно j -му нейрона,

b_j - зміщення j -го нейрона.

2.4.3 Керований метод навчання Хебба.

В [19] запропоновано керований метод Хебба (Supervised Hebbian Learning, SHL) як метод, що є найбільш подібний до нейро-біологічного методу навчання спайкінгових нейронних мереж. Метод SHL спрямований на те, щоб вихідний сигнал з'являвся в бажаний час. Для цього в ньому використовується так званий «повчальний» сигнал. Метод SHL має недолік, який полягає в тому, що навіть

після того, як бажаний патерн на виході досягнутий, алгоритм навчання продовжує змінювати ваги. Таким чином, повинні бути введені обмеження для забезпечення стабільності. Однак, проблема із встановленням обмежень вирішується не просто, тому що важко дізнатися, коли в процесі навчання вони повинні бути застосовані. Ваги змінюються після кожної епохи, що може призвести до нестабільності або появи при тестуванні неочікуваних та незадовільних результатів [20].

Процес навчання заснований на трьох спайках (два пре-синаптичних і один пост-синаптичний), згенерованих в кожному циклі навчання. Перший пре-синаптичний імпульс в момент часу t_1^{in} приймався як вхідний сигнал, другий пре-синаптичний імпульс в момент часу $t_2^{in} = t^d$ вказував на бажаний час спрацьовування пост-синаптичного нейрона. Формула, що описує правило навчання, має такий вигляд:

$$\Delta w = \eta(t^{out} - t^d) \quad (2.12)$$

де:

η – коефіцієнт швидкості навчання, повинен бути > 0 ;

t^{out} реальний час появи пост-синаптичного спайка.

2.4.4 Метод ReSuMe.

В [21, 22] було запропоновано метод навчання RuSuMe, оснований на, так званому, «віддаленому навчання» (remote supervision), що використовує множинні спайки. В ньому нейронна мережа вчиться генерувати задану послідовність імпульсів у відповідь на вхідну послідовність. Метод ReSuMe об'єднує переваги методу Хебба з концепцією віддаленого навчання. Поняття «віддалене навчання» означає, що бажані сигнали не направляються безпосередньо на нейрони, що навчаються (як це відбувається в методі SHL). Вони беруть участь у зміні синаптичних ваг зв'язків, що закінчуються на нейронах, що навчаються. А саме, синаптична вага w_{ki} , між деяким пре-

синаптичним нейроном $n_k^{in}(i)$ і деяким відповідним пост-синаптичним нейроном n_i^{out} , модифікується згідно з двома правилами. Перше правило залежить від кореляції між часом спрацьовування нейронів $n_k^{in}(i)$ і n_i^{out} . Друге правило визначається кореляцією між часом спрацьовування нейронів $n_k^{in}(i)$ і $n_j^d(i)$. Під $n_j^d(i)$ розуміють нейрон, що навчається, який виробляє бажаний сигнал для n_i^{out} . Для збуджуючих синапсів ці два правила відповідають правилам збільшення і зменшення ваг в методі STDP і описуються за допомогою вікон навчання (learning windows) $W^d(s^d)$ та $W^{out}(s^{out})$, що зображені на рисунку 2.9.

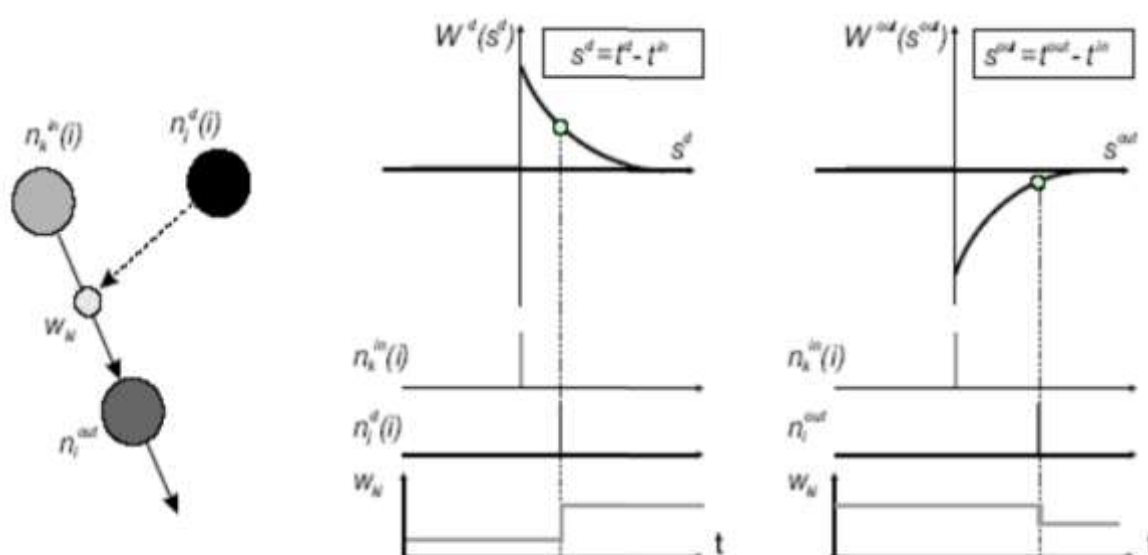


Рисунок 2.9 – Концепція віддаленого навчання і вікон навчання ReSuMe.

2.4.4 Запропоноване удосконалення методу STDP.

Збуджуючі нейрони зв'язуються з класами після навчання, яке базується на виявленні максимально в середньому активного нейрона, що відповідає певному класу.

При використанні методу STDP розподіл збуджуючих нейронів по класах проводиться після навчання шляхом виявлення найбільш активного нейрона при пред'явленні нейронній мережі образу відповідного класу.

В даній роботі пропонується удосконалити метод такими чином, щоб збуджуючі нейрони зв'язувались з класами не після всього процесу навчання, а вже після перших епох. Таким чином вихідним нейронам можна буде присвоїти мітки класів вже після декількох епох.

Також пропонується ввести «учителя» в роботу даного алгоритму. Завдяки тому, що після першої епохи, ми вже зв'язали вихідні нейрони з класами, ми можемо ввести певне правило, що буде давати «нагороду» тим нейронам, на яких правильно генеруються імпульси, тобто тим, з яких прийшов імпульс на вихідний нейрон з відповідною міткою класу, екземпляр якого був поданий на вхід мережі. Пропонується використати функцію оцінки — суму квадратів відстаней вихідних сигналів мережі від їх необхідних значень:

$$H = \frac{1}{2} \sum_{\tau \in \vartheta_{out}} (Z(\tau) - Z^*(\tau))^2 \quad (2.13)$$

Таким чином, навчання відповідно до удосконаленого методу STDP виконується у вигляді наступних кроків:

- 1) Випадкова ініціалізація синаптичних зв'язків між шарами та в межах прихованого шару;
- 2) Навчання спайкінгової нейронної мережі методом STDP n епох (де n — можливо конфігурувати залежно від датасету);
- 3) Шляхом виявлення найбільш активного нейрона при пред'явленні нейронній мережі певного представника відповідного класузначаються мітки класів до відповідних вихідних нейронів;
- 4) Подальше навчання вже з використанням функції оцінки вихідних сигналів відносно бажаних k епох або доки $H < \varepsilon$ (де k задане максимальне можливе значення епох, H — значення функції оцінки, ε — бажана точність, при якій можливо зупинити навчання)

2.5 Попередня обробка аудіо сигналу та форматування вхідних даних для спайкінгової нейронної мережі

Вхідні дані – одна з найважливіших речей під час моделювання спайкінгової нейронної мережі. Адже щоб мережа коректно працювала та могла навчатися необхідно правильно обробити їх. В нашому випадку вхідні дані – це аудіо сигнали, представлені чіткою послідовністю потенціалів вхідних імпульсів.

Необхідно правильно задати максимальне можливе значення вхідних імпульсів. Якщо воно буде надто малим, то вихідні нейрони будуть рідко генерувати імпульси, а якщо воно буде надто великим, то вихідні нейрони будуть надто часто генерувати імпульси. Це буде спотворювати процес навчання мережі. Щоб вирішити цю проблему можна нормалізувати вхідні імпульси так, щоб значення імпульсу лежало в межах від 0 до 1.

Щоб мережа могла розпізнавати схожі вхідні образи необхідно підібрати значення параметру дискретизації вхідних імпульсів. Для того, щоб вирішити проблему сильно корельованих образів задаємо параметр дискретизації так, щоб на вхідні нейрони імпульси приходили з частотою у проміжку від 100-200 за 1с.

Наступним кроком є отримання особливостей (features), які будуть подаватися на вхід спайкінгової нейронної мережі. Аудіо сигнал має характерну структуру в частотному діапазоні. У його складі є основний тон, як правило, з максимальною амплітудою, і супутні гармоніки, проте найбільш значущими є кілька перших гармонік.

Більшість систем класифікації звуків на основі штучних нейронних мереж використовують візуалізацію спектру частоти звуку (спектограму) для класифікації. Спектрограма – це зображення, що показує залежність спектральної щільності потужності сигналу від часу. Спектрограми застосовуються для ідентифікації мови, аналізу звуків тварин, в різних областях музики, радіо- і гідролокації, обробці мови, сейсмології і в інших областях. Найбільш відомий метод для отримання спектограми звуку MFCC. Основна

відмінність методу MFCC полягає в тому, що він використовує квазі-логарифмічну шкалу частоти, що більше схоже на те, як слухова система людини обробляє звуки [23].

Метод MFCC значно має декілька переваг порівняно з іншими методами:

- за рахунок неявного використання спектрової інформації використовується знання про хвильовий характер сигналу;
- за рахунок проектування спектру на мел-шкалу виділяються найбільш важливі для сприйняття частоти;
- завдяки можливості довільно обирати кількість мел-кепстральних характеристик сигналу, можливо стиснути звуковий фрейм, тобто можливо масштабувати обсяг оброблюваної інформації.

Проте метод візуалізації аудіо сигналу унеможливорює обробку звуків в живому часі. Тому вирішено використати особливості частотних смуг. Для цього розділимо вхідний аудіо сигнал на частотні смуги (рис 2.10). Зазвичай використовуються розділення на три смуги (низькі, середні і високі частоти), але для більшої точності класифікації можливе більш детальне розділення:

- дуже низькі частоти (ДНЧ);
- низькі частоти (НЧ);
- середні частоти (СЧ);
- високі частоти (ВЧ);
- дуже високі частоти (ДВЧ).

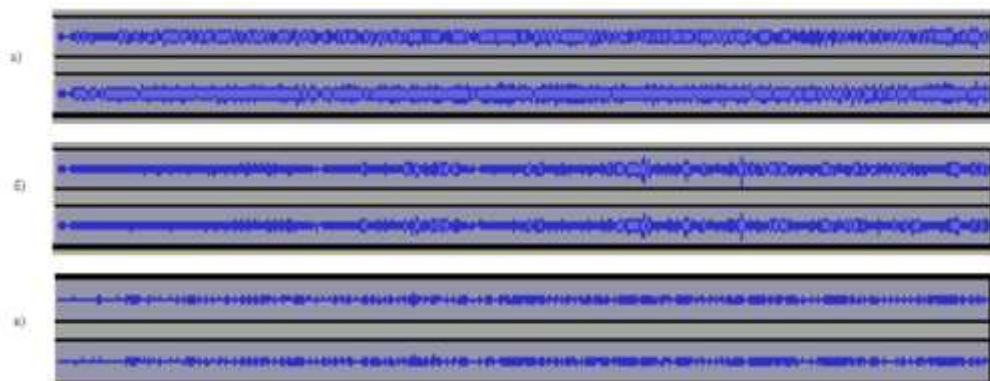


Рисунок 2.10 – Приклад отримання частотних смуг: а) – низькі частоти; б) – середні частоти; в) – високі частоти.

Прийнято використовувати такі значення діапазонів для смуг: ДНЧ – від 30 до 100 кГц, НЧ – від 100 до 400 кГц, СЧ – від 0.4 до 3 МГц, ВЧ – від 3 до 10 МГц, ДВЧ – від 10 до 30 МГц.

2.6 Розробка методу класифікації звуків на основі спайкінгової нейронної мережі

Запропонований метод класифікації звуків на основі спайкінгової нейронної мережі складається з таких кроків:

- 1) Початкова ініціалізація нейронної мережі;
- 2) Розподіл збуджувальних та гальмівних нейронів;
- 3) Визначення кількості вхідних, прихованих та вихідних нейронів;
- 4) Створення синаптичних зв'язків між шарами та в межах прихованого шару;
- 5) Первинна обробка звуку;
- 6) Формування частотних смуг аудіо сигналу;
- 7) Подання сигналів на вхідні нейрони;
- 8) Отримання результатів класифікації на вихідних нейронах.

2.7 Метрики оцінювання якості класифікації

При оцінюванні моделі ШНМ найважливіше визначити, яким чином оцінювати результат роботи алгоритму. Є велика кількість метрик і вибір правильної метрики є важливим пунктом під час навчання мережі. Зазвичай, якщо одна модель виконує задачу значно краще за іншу, то вона буде мати кращі показники по більшості метрик. Проте можливі ситуації, коли модель демонструє кращі результати за однією метрикою та гірші за іншою. Саме тому при вирішенні будь-якої задачі машинного навчання необхідним є визначення метрики, яка буде основною [24].

Також важливо обрати ціль, яку алгоритм повинен прагнути досягти. Наприклад, при проведенні наукової роботи часто є доречним використовувати таку саму метрику, яку використовували на таких же даних для іншого алгоритму, для того, щоб була можливість порівняти отримані результати за однаковими правилами. А при проведенні комерційної розробки алгоритму машинного навчання, можлива навіть розробка власної метрики, яка найкращим чином допоможе формалізувати поставлену проблему.

Отже, найчастіше використовуються наступні метрики: точність, «Точність та повнота», F-метрика.

Найпростіший випадок метрики – точність. Це частка результатів, за якими класифікатор прийняв правильне рішення.

$$Acc = \frac{P}{N} \quad (2.14)$$

де:

P – кількість результатів, за якими класифікатор прийняв правильне рішення;

N – розмірність усієї тестової вибірки.

За рахунок простоти цієї метрики, вона має одна неприємну особливість, яку необхідно враховувати при використанні. Вона надає всім результатам однакову вагу, що може призвести до того, що алгоритм має високу загальну точність, але при цьому в межах деяких класів класифікатор видає зовсім низькі показники точності.

«Точність та повнота» («precision and recall») це метрика, яка здебільшого використовуються в задачах розпізнавання образів, інформаційного пошуку та класифікації. Точність – це частка правильно класифікованих екземплярів серед усіх, а повнота – це частка правильно прогнозованих екземплярів відносно загальної кількості релевантних. Отже, як точність, так і повнота, ґрунтуються на розумінні і мірі релевантності.

$$Precision = \frac{TP}{TP+FP} \quad (2.15)$$

$$Recall = \frac{TP}{TP+FN} \quad (2.16)$$

де:

$TP(true\ positive)$ – істинно-позитивне рішення, класифікатор прийняв рішення про те, що образ відноситься до класу, і цей результат виявився правильним;

$TN(true\ negative)$ – істинно-негативне рішення, класифікатор прийняв рішення про те, що образ не відноситься до класу, і цей результат виявився правильним;

$FP(false\ positive)$ – хибно-позитивне рішення, класифікатор прийняв рішення про те, що образ відноситься до класу, але цей результат виявився неправильним;

$FN(false\ negative)$ – хибно-негативне рішення, класифікатор прийняв рішення про те, що образ не відноситься до класу, але цей результат виявився неправильним.

Ці значення легко розрахувати на підставі таблиці канонічної таблиці відповідей (таблиця 2.1). Вона будується окремо для кожного класу або на всіх класи разом.

Таблиця 2.1 – Приклад таблиці відповідей

| Клас-п | | Істинні значення | |
|--------------|------------------|------------------|------------------|
| | | Позитивна оцінка | Негативна оцінка |
| Класифікатор | Позитивна оцінка | TP | FP |
| | Негативна оцінка | FN | TN |

Очевидно, що чим вище точність і повнота, тим краще. Проте на реальних випадках максимальна точність та повнота недосяжні одночасно і доводиться шукати компроміс між одним та іншим. F-метрика є гармонійно об'єднує точність та повноту. Вона сходиться до нуля, якщо точність або повнота сходиться до нуля.

$$F = (\beta^2 + 1) \frac{Precision * Recall}{Precision + Recall}, \quad (2.17)$$

де β - приймає значення в діапазоні $0 < \beta < 1$ для підвищення ваги точності, при $\beta > 1$ пріоритет віддається повноті. При $\beta = 1$ створюється збалансована F-міра – F1 метрика.

2.8 Висновок

В результаті проведеного аналітичного аналізу існуючих математичних моделей імпульсних нейронів зрозуміло, що у цих моделей спайки цілком і повністю характеризуються часом генерації вихідного імпульсу $t(f)$, визначеним пороговим критерієм ϑ . Більшість моделей можливо спростити до вигляду моделі SRM, якщо знехтувати впливом непрямих ефектів, що пов'язані з попередніми вихідними спайками. Проте найбільш ефективною моделлю вважається LIF-модель за рахунок спрощення моделі нейрону.

Було розроблено структуру спайкінгової нейронної мережі для класифікації звуків пострілів, визначенні такі значення: кількість нейронів у кожному шарі, відсоток зв'язків між нейронами шарів, відношення збуджувальних до гальмівних зв'язків, швидкість навчання, кількість епох навчання.

Після проведеного детального аналізу методів навчання спайкінгових нейронних мереж видно, що найбільш використовуваними з них є методи основані на модифікованому правилі Хебба STDP і його модифікації.

Саме тому в даній роботі було вирішено використати LIF-модель спайкінгового нейронного елемента, а як метод навчання – модифіковане правило Хебба (STDP), та удосконалити його таким чином, щоб зменшити загальний час навчання.

Удосконалення полягає у тому, щоб провести розподіл вихідних-збуджуючих нейронів по класах (присвоювання класам міток) після декількох перших епох навчання та ввести додаткове правило для навчання з учителем. Вихідний нейрон, який найчастіше генерував імпульс на образах певного класу отримає мітку цього класу. Це дозволить уникнути перенавчання під час подальших епох.

Розглянуто методи попередньої обробки аудіо сигналів та обрано розподіл по частотним смугам як найефективний для вирішення задачі класифікації звуків пострілів на основі спайкінгової нейронної мережі.

На основі вище описаного було запропоновано метод класифікації звуків пострілів на основі спайкінгової нейронної мережі.

Розроблено загальну структурну схема функціонування програмного засобу та схема загального алгоритму функціонування програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі.

Було розглянуто найвідоміші метрики оцінки класифікації та запропоновано використати F-метрику для оцінювання точності і повноти класифікації звуків пострілів на основі спайкінгової нейронної мережі.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ КЛАСИФІКАЦІЇ ЗВУКІВ ПОСТРІЛІВ НА ОСНОВІ СПАЙКІНГОВОЇ НЕЙРОННОЇ МЕРЕЖІ

3.1 Обґрунтування вибору мови

Здійснимо вибір мови програмування для розробки програмного забезпечення для вирішення задачі класифікації звуків пострілів на основі спайкінгової нейронної мережі. Під час вибору будуть розглянуті сучасні мови програмування такі як Java, C#, Kotlin, Scala.

Kotlin – це мультипарадигмова мова програмування. Вона поєднує властивості об'єктно-орієнтованого та функційного програмування. Kotlin є статично типізована мова, що працює поверх Java Virtual Machine (JVM). Програми на Kotlin, так само як і на Java, компілюються в байт-код, який потім інтерпретується для кожної платформи індивідуально та виконується JVM, тому програми, будуть працювати всюди, де встановлена JVM [25].

Мова значно запозичила синтаксис із мов програмування Java та Scala. Зокрема, взято за основу об'єктну модель Java, проте її модифіковано, та функціональні особливості Scala.

Однією з переваг є те, що код на Kotlin повністю сумісний з Java, тому одну програму можна писати використовуючи дві мови програмування, а так як для неї існує величезна кількість відкритих бібліотек та вже готових рішень стандартних задач, більшість з яких знаходиться в публічному доступі, це відкриває безмежні можливості програмістам. Використовуючи готові та оптимізовані рішення, можна не витратити час на написання власних.

Також у Kotlin використовується «авто-прибирання сміття». Тобто відбувається автоматичне керування пам'яттю. Основний принцип, з яким працюють різні алгоритми збирання сміття, полягає в наступному:

- 1) Визначити об'єкти, які не потрібні для подальшої роботи програми;
- 2) Вивільнити ресурси, які були зайняті визначеними об'єктами.

А у Kotlin це працює так: об'єкт видаляється з пам'яті, якщо на нього не існує жодного посилання в межах програми.

Ще однією «новинкою» мови Kotlin є корутини (coroutines - співпрограми). У Kotlin підтримка корутин вбудовано в набір основних бібліотек. Корутини - це полегшені потоки. Полегшений потік означає, що він не прив'язаний до нативного потоку, тому він не вимагає перемикання контексту на процесор, тому він набагато швидше. Існує два види корутин: з використанням стеку та без використання стеку. Kotlin реалізую корутини без стеку – саме тому вони не привязані до нативного потоку. Подібно потокам, корутини можуть працювати паралельно, чекати один одного і спілкуватися. Найбільша відмінність полягає в тому, що корутини дуже дешеві, майже безкоштовні: ми можемо створювати їх тисячами і платити дуже мало з точки зору продуктивності. Потоки ж обходяться дорого. Тисяча потоків може стати серйозною проблемою навіть для сучасної машини.

Результати порівняння мов Kotlin і C# зображені в таблиці 3.1.

Таблиця 3.1 – Порівняння мов програмування Kotlin і C#

| Ознака порівняння | Kotlin | C# |
|--|--------|----------------------------|
| Підтримка методології ООП | Так | Так |
| Кроссплатформеність | Так | Ні, тільки з додатковим ПЗ |
| Наявність контейнерів (List, Set, Map) | Так | Так |
| Багато безкоштовних бібліотек | Так | Ні |
| Сумісність різних версій | Так | Ні |
| Відкритий доступ до вихідного коду | Так | Ні |

Продовження таблиці 3.1

| | | |
|--------------------------|-----|--|
| Безкоштовне використання | Так | Ні |
| Підтримка ко-рутин | Так | Тільки з використанням сторонніх бібліотек |

Враховуючи всі переваги і недоліки, була обрана мова Kotlin, бо вона дозволяє розроблювати додатки під будь-яку платформу і безкоштовна для використання, а також має безліч відкритий бібліотек та готових рішень.

3.2 Обґрунтування вибору середовища програмування

Intelijj IDEA — інтегроване середовище розробки (IDE) для мов програмування Java, Kotlin та Scala, представлене в січні 2001 року компанією JetBrains, написана на Java та згодом переписана на Kotlin [26].

Основні особливості, які доступні на даний момент користувачеві:

- Живі макети (layout): редагувальник інтерфейсу в реальному часі;
- Консоль розробника: підказки по оптимізації, допомога по перекладу, стеження за напрямком;
- Регулярні оновлення від розробників;
- Підтримка плагінів для збирання проектів (Maven, Gradle);
- Kotlin-орієнтований рефакторинг та швидкі виправлення;
- Шаблони для створення поширених дизайнів та компонентів;
- Багатий редактор макетів (layouts) що дозволяє користувачам перетягнути і покласти (drag-and-drop) компоненти користувацького інтерфейсу, як варіант, переглянути одночасно макети (layouts) на різних конфігураціях екранів;
- Легка інтеграція з базами даних;

Компанія JetBrains пропонує дві версії Intelijj IDEA:

- Community edition – повністю безкоштовна версія, що розповсюджується під ліцензією Apache 2.0. В ній реалізована повна підтримка Java SE, Kotlin, Groovy, Scala, а також інтеграція з найбільш популярними системами керування версій.

- Ultimate Edition – доступна під комерційною ліцензією. В ній додатково реалізована підтримка Java EE, UML-діаграм, підрахунок покриття коду, а також підтримка інших систем управління доступу, мов програмування та фреймворків.

Для студентів денної форми навчання компанія пропонує безкоштовні ліцензії для Ultimate Edition.

Враховуючи всі переваги і недоліки, було обрано середовище програмування IntelliJ IDEA, так як воно регулярно оновлюється, підтримується розробниками з JetBrains та має зручний графічний інтерфейс.

3.3 Розробка алгоритму функціонування програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі

Розробимо загальну структуру системи. Систему доцільно розбити на такі модулі:

1) Модуль обробки вхідних даних від користувача – інтерфейс. Він опрацьовує запити користувача та передає інформацію про дію далі в модуль первинної обробки аудіо сигналів. В цьому модулі оброблюються вхідні дані від користувача, такі як:

- Завантаження спайкінгової нейронної мережі;
- Вказання шляху до аудіо файлу з розширенням mp3 або вказання зчитуючого пристрою для налаштування форматування вхідних даних для мережі;
- Запуск процесу класифікації;
- Збереження результатів.

Також цей модуль виконує функцію виводу системних помилок та результатів роботи програми.

2) Модуль первинної обробки зображень відповідає за створення частотних смуг для аудіо файлу або розділення аудіо сигналу з зчитуючого пристрою на частоті смуги та формування вхідних даних для спайкінгової нейронної мережі у вигляді імпульсів заданої із необхідною частотою та напругою.

3) Модуль спайкінгової нейронної мережі відповідає за ініціалізацію мережі:

- створення трьох шарів нейронів: вхідного, прихованого, вихідного;
- створення статичних синаптичних зв'язків між вхідним та прихованим шаром;
- створення динамічних зв'язків між прихованим та вихідним шаром.

Після ініціалізації можливий прогін вхідних даних через мережу.

Розроблена загальна структурна схема функціонування програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі зображена на рисунку 3.1.

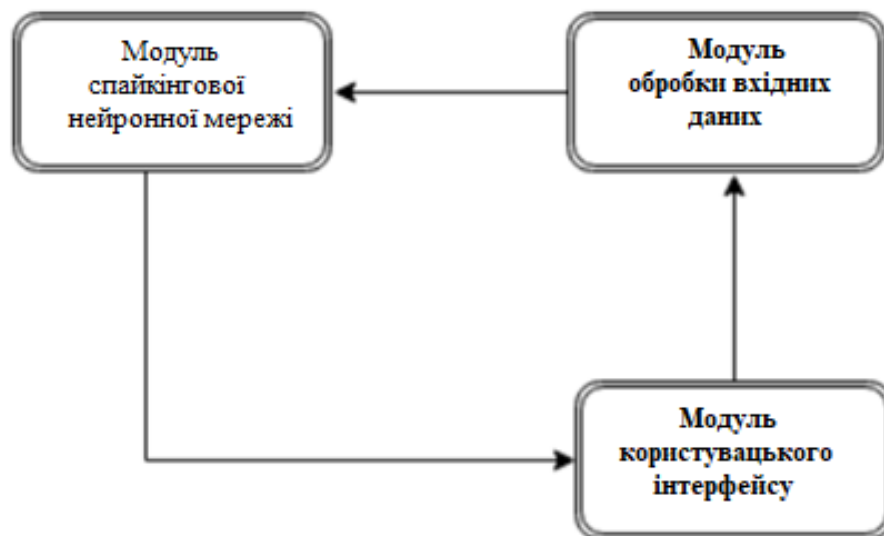


Рисунок 3.1 – Загальна структура програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі

Розроблена схема загального алгоритму функціонування програмного засобу для класифікації звуків на основі спайкінгової нейронної мережі. Запропонована схема зображена на рисунку 3.2.

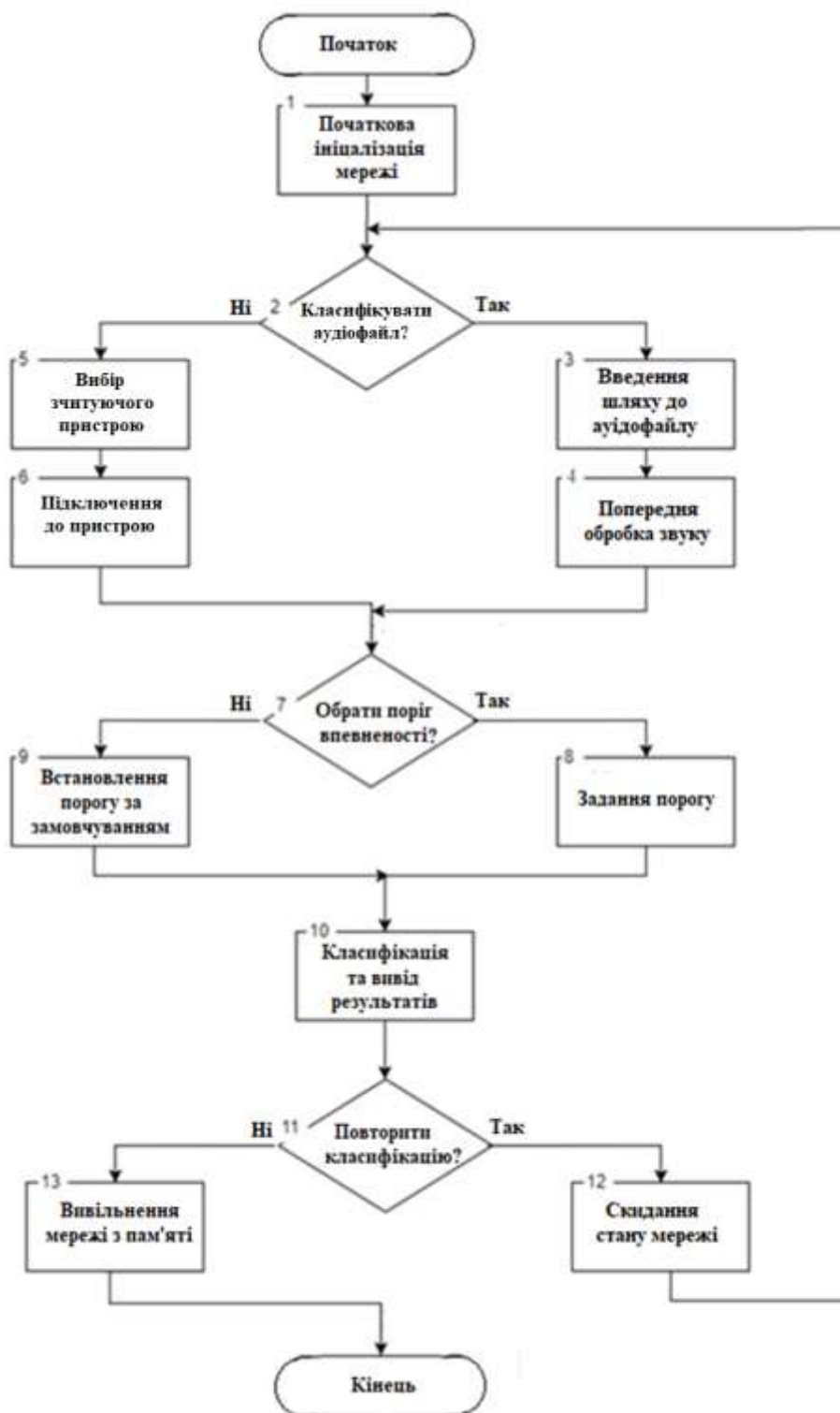


Рисунок 3.2 – Схема загального алгоритму роботи програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі

Даний алгоритм складається з таких кроків:

- 1) Початкова ініціалізація програми;
- 2) Користувачем обирається варіант класифікації нового аудіофайлу або раніше вже обробленого, якщо новий аудіо файл, то перехід на крок 3, інакше перехід на крок 5;
- 3) Введення шляху до нового аудіофайлу;
- 4) Відбувається попередня обробка аудіофайлу та форматування вхідних даних;
- 5) Введення шляху до попередньо оброблених даних;
- 6) Зчитування даних та форматування вхідних даних;
- 7) Користувачем обирається поріг впевненості класифікації, якщо обрано, то перехід на крок 8, інакше – на крок 9;
- 8) Задання користувацького порогу впевненості в класифікаторі;
- 9) Задання порогу впевненості за замовчуванням в класифікаторі;
- 10) Відбувається процес класифікації та вивід результатів;
- 11) Користувачем обирається чи виконати ще одну класифікації, якщо обрано так, то перехід на крок 12, інакше – на крок 13;
- 12) Скидання минулого стану мережі та перехід на крок 2;
- 13) Вивільнення мережі з пам'яті та завершення програми.

3.4 Реалізація програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі

Розробимо програму для класифікації звуків пострілів на основі спайкінгової нейронної мережі згідно з запропонованою структурою. Програму було розділено на три програмні модулі:

- 1) модуль вхідних/вихідних даних;

2) модуль для попередньої обробки даних та виділення особливостей з аудіо сигналів;

3) модуль класифікатора на основі спайкінгової нейронної мережі.

Зв'язки класів відобразимо на UML діаграмі класів програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі представлені на рисунку 3.3.

В першому модулі було розроблено клас `UserIOProcessor`, який слухає та оброблює вхідні запити від користувача, виводить проміжні помилки програми, та результат роботи класифікатора.

В модулі попередньої обробки аудіо сигналів було розроблено клас `AudioPreprocessor`, що відповідає за обробку аудіо сигналів та виділення особливостей зі звуку. Для розділення аудіо сигналу на частотні смуги було використано бібліотеку – `Librosa`.

В модулі нейронної мережі було розроблено такі класи: `NeuralNetwork`, `BaseNeuron`, `Neuron`, `Answer`, `AnswerListener`, `InputHelper`, `InputThread`, `Constants`.

`NeuralNetwork` – клас нейронної мережі, через який проводяться основні операції з нею, такі як створення, зчитування, запис у файл, навчання, тестування.

`BaseNeuron` – абстрактний клас, що містить в собі базові поля та методи необхідні для роботи нейронів.

`Neuron` – клас, що наслідує `BaseNeuron`, виконує функцію нейрона вхідного і прихованого шару та реалізує метод для відправлення імпульсу по своїм зв'язкам та метод для зміни ваг синапсів.

`Answer` – клас, що наслідує `BaseNeuron` та виконує функцію вихідного нейрона.

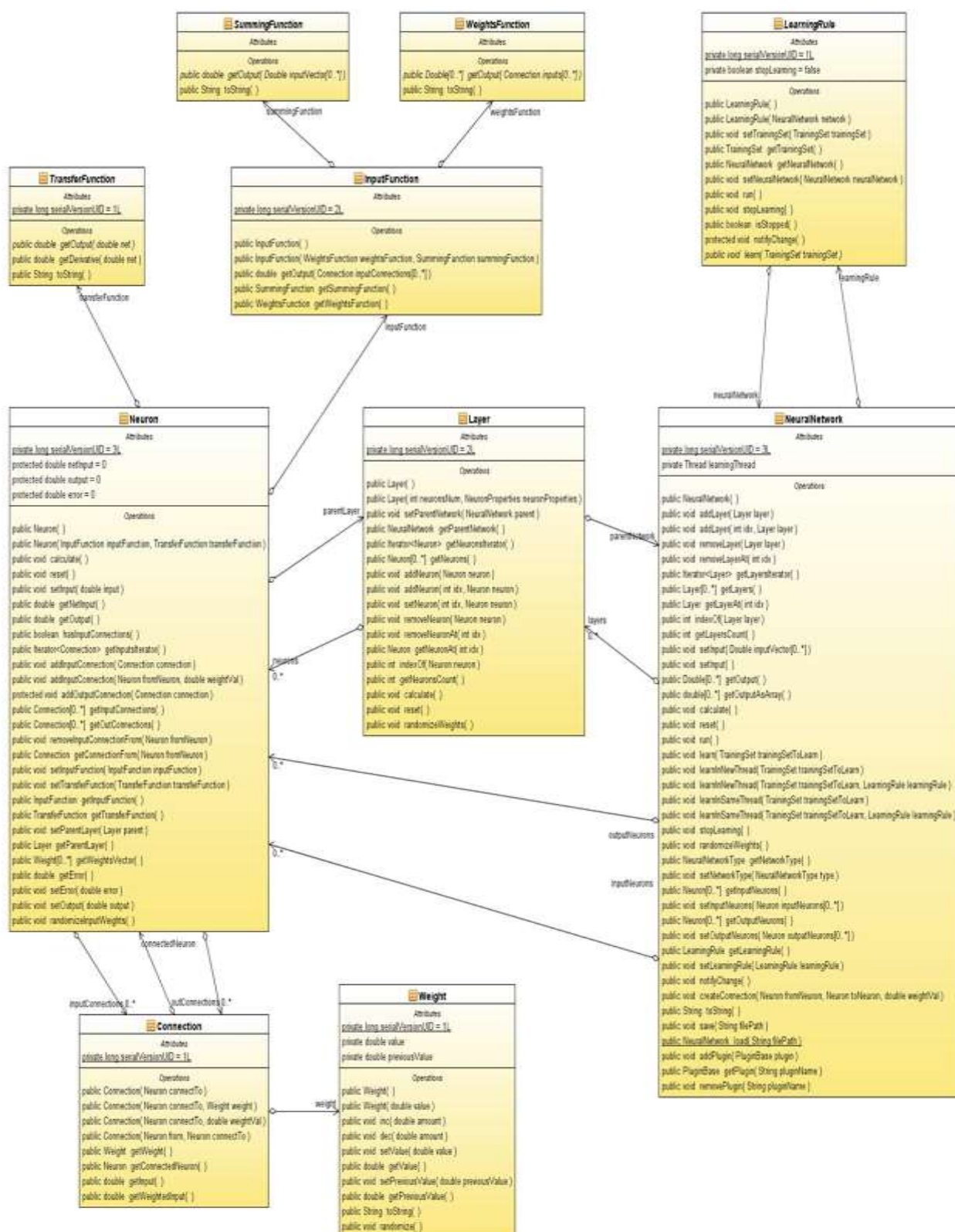


Рисунок 3.3 – UML діаграма класів програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі

AnswerListener – інтерфейс для передачі інформації з модуля нейронної мережі до модуля інтерфейсу.

`InputThread` – клас, що запускає окремий потік для подачі даних на вхідний нейрон.

`Constants` – клас, що містить у собі параметри налаштування нейронної мережі.

На діаграмі зображено усі класи та їх ієрархію зв'язків, що разом утворюють програмний засіб для класифікації звуків на основі спайкінгової нейронної мережі.

3.6 Тестування та аналіз результатів роботи програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі

Навчання спайкінгової нейронної мережі проводилось на наборі даних для чотирьох класів (Пістолет, Рушниця, Міномет та Танкова гармата), що складається з 80 аудіофайлів, по 20 на кожен клас. За для порівняння звичайного методу STDP та удосконаленого було проведено навчання на 350 епох.

На рисунку 3.4 зображено графік навчання спайкінгової нейронної мережі для класифікації звуків пострілів звичайним методом STDP. На рисунку 3.5 зображено навчання спайкінгової нейронної мережі для класифікації звуків пострілів методом STDP з удосконаленням запропонованим в даній роботі.

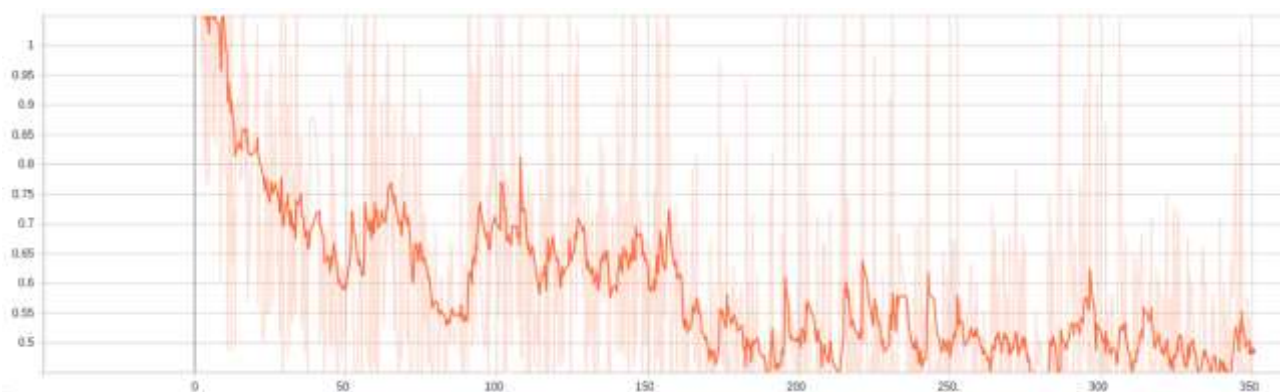


Рисунок 3.4 – Графік навчання спайкінгової нейронної мережі для класифікації звуків пострілів за методом STDP

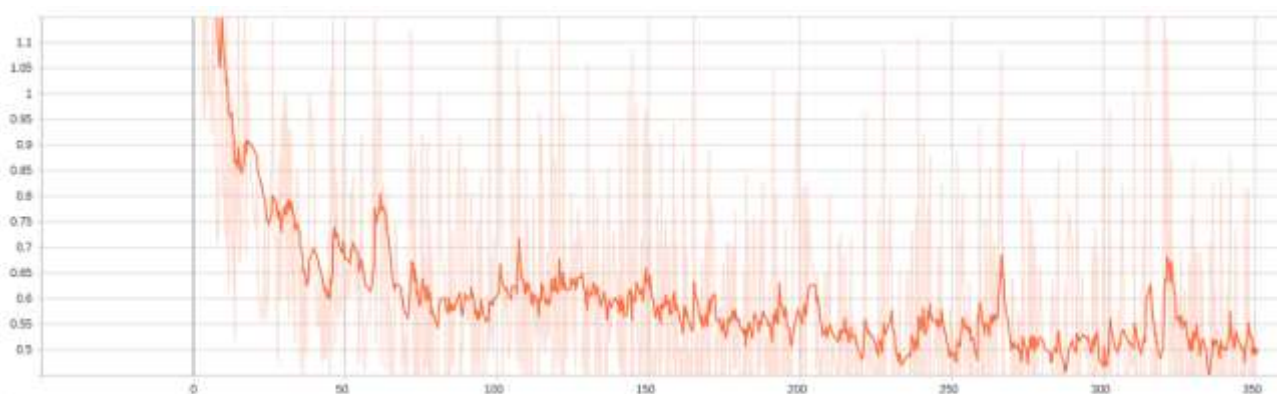


Рисунок 3.5 – Графік навчання спайкінгової нейронної мережі для класифікації звуків пострілів за удосконаленим методом STDP

Як видно з графіків, удосконалений метод навчання дозволяє нейронній мережі досягти такої самої точності та швидше зійтись. Також можна помітити, що стрибки від епохи до епохи стали значно меншими, адже тепер в мережі є додаткове правило, що керує навчанням нейронних зв'язків.

Порівняння метрик обох методів зображено на рисунку 3.6.

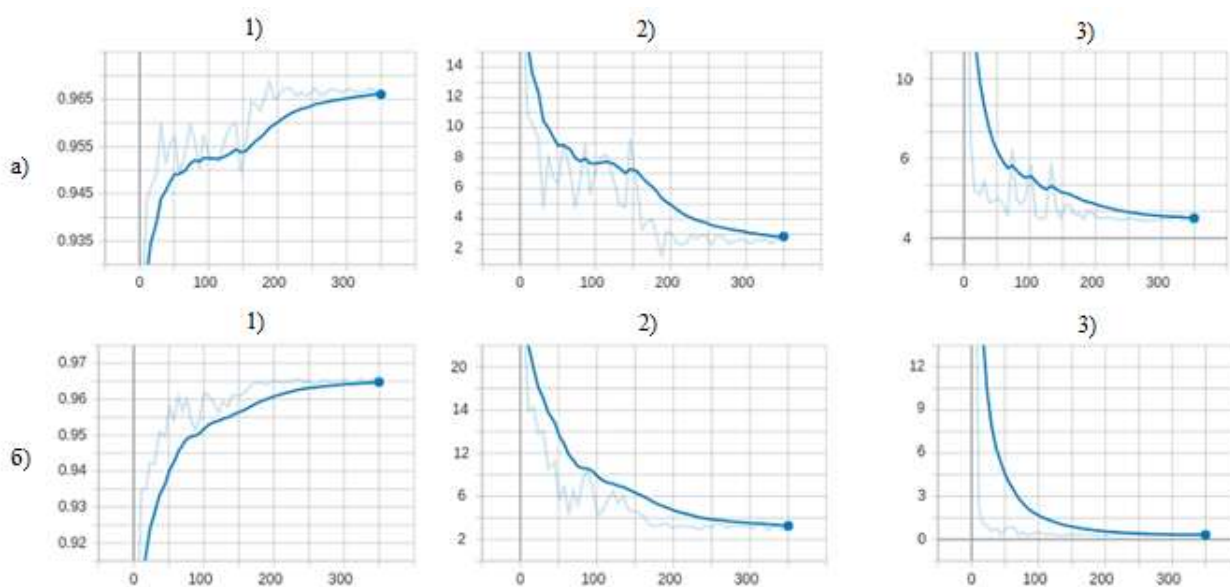


Рисунок 3.6 – Порівняння удосконаленого та звичайного методу STDP: а) звичайний метод STDP; б) удосконаленого метод STDP; 1) F1 метрика; 2) FN метрика; 3) FP метрика

Проаналізувавши отримані результати можна стверджувати, що для отримання однакової точності $F1=0.965$ нейронній мережі, яка навчається за удосконаленим методом STDP, необхідно 150 епох, порівняно зі звичайним методом, який потребує мінімум 200 епох навчання.

Провівши 20 подібних навчань спайкінгової нейронної мережі, 10 – з удосконаленням, 10 – без, було отримано усередненні значення часу навчання, які відображені в таблиці 3.2.

Таблиця 3.2 – Порівняння методів навчання

| № навчання \ Метод | STDP | Удосконалений STDP |
|-----------------------|-------------------------------|-----------------------|
| | Кількість епох ($F1=0.965$) | |
| 1 | 201 | 153 |
| 2 | 220 | 164 |
| 3 | 200 | 170 |
| 4 | 204 | 166 |
| 5 | 205 | 157 |
| 6 | 210 | 169 |
| 7 | 213 | 165 |
| 8 | 207 | 171 |
| 9 | 211 | 182 |
| 10 | 209 | 158 |
| Середнє значення | 204 | 165 |

Як видно з таблиці, застосування удосконаленого методу STDP зменшує кількість епох навчання в середньому в 1.2 рази, тобто зменшує час навчання на 20%.

На рисунку 3.7 зображено приклад роботи програмного засобу для класифікації мало каліберної зброї. Користувач вводить режим роботи – з файлу, та вводить шлях до файлу зі звуком пострілу пістолету. Після класифікації програма видає результат: Пістолет 93%. Час класифікації 53 мілісекунди.

```
root@max-System-Product-Name:~# ./classifier
Initializing network...
Select Mode(f - file, r- reader): f
Enter file path: dataset/handgun/audio/handgun_001.mp3
Reading file dataset/handgun/audio/handgun_001.mp3
Creating inputs...
Classifying..
Result: Handgun 93%, time spent 53 ms
root@max-System-Product-Name:~# █
```

Рисунок 3.7 – Приклад класифікації звуку мало каліберного пістолету на основі спайкінгової нейронної мережі

На рисунку 3.8 зображено приклад роботи програмного засобу для класифікації багато каліберної гармати. Користувач вводить режим роботи – з файлу, та вводить шлях до файлу зі звуком пострілу з танка. Після класифікації програма видає результат: Танкова гармата 97%. Час класифікації 47 мілісекунд.

```
root@max-System-Product-Name:~# ./classifier
Initializing network...
Select Mode(f - file, r- reader): f
Enter file path: dataset/tank/audio/tank_001.mp3
Reading file dataset/tank/audio/tank_001.mp3
Creating inputs...
Classifying..
Result: Tank 97%, time spent 47 ms
root@max-System-Product-Name:~# █
```

Рисунок 3.8 – Приклад класифікації звуку пострілу багато каліберної гармати на основі спайкінгової нейронної мережі

Тестування програмного засобу показало його працездатність та зменшення часу навчання спайкінгової нейронної мережі на 20%.

3.7 Висновок

В результаті порівняльного аналізу об'єктно-орієнтованих мов на основі об'єктивних переваг було обрано мову програмування Kotlin розроблену компанією JetBrains. Використання корутин якнайкраще підходить для комп'ютерного моделювання поведінки спайкінгових нейронних елементів.

Після аналізу існуючих середовищ програмування для обраної мови та платформи було обрано середовище програмування IntelliJ IDEA, яке підтримується розробниками мови Kotlin та має зручний графічний інтерфейс.

Було розроблено загальний алгоритм функціонування програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі.

На основі алгоритму функціонування програмного засобу було реалізовано програмний засіб для класифікації звуків пострілів на основі спайкінгової нейронної мережі.

Було відібрано навчальну та тестові вибірку звуків пострілів, що дозволило в повній мірі продемонструвати працездатність програмного засобу для класифікації звуків пострілів на основі спайкінгової нейронної мережі. Порівняння швидкості навчання показало, що удосконалений метод навчання спайкінгової нейронної мережі зменшує час навчання в середньому на 20%.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть доцент кафедри КН Колесницький О.К та доцент кафедри КН Арсенюк І.Р.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.1.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розробки

| Критерії | Прізвище, ініціали, посада експерта | |
|---|--|----------------------|
| | Колесницький О.К | Арсенюк І.Р |
| | Бали, виставлені експертами: | |
| 1 | 4 | 4 |
| 2 | 4 | 3 |
| 3 | 5 | 4 |
| 4 | 4 | 4 |
| 5 | 5 | 4 |
| 6 | 5 | 4 |
| 7 | 4 | 3 |
| 8 | 5 | 4 |
| 9 | 4 | 4 |
| 10 | 4 | 3 |
| 11 | 4 | 4 |
| 12 | 3 | 4 |
| Сума балів | СБ ₁ = 55 | СБ ₂ = 45 |
| Середньоарифметична сума балів $\overline{СБ}$ | $\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{2} = 50$ | |

Отже, з отриманих даних таблиці 4.1 видно, що нова розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько-технологічної роботи.

Для розробки нового програмного продукту необхідні такі витрати. Основна заробітна плата для розробників визначається за формулою (4.1):

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M - місячний посадовий оклад конкретного розробника;

T_p - кількість робочих днів у місяці, $T_p = 21$ день;

t - число днів роботи розробника, $t = 40$ днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.2.

Таблиця 4.2 – Розрахунки основної заробітної плати

| Працівник | Оклад M , грн. | Оплата за робочий день, грн. | Число днів роботи, t | Витрати на оплату праці, грн. |
|--------------------|---------------------|------------------------------------|---------------------------|-------------------------------------|
| Науковий керівник | 7000 | 333,33 | 5 | 1666,65 |
| Інженер-програміст | 4500 | 214,28 | 40 | 8571,2 |
| Всього: | | | | 10237,85 |

Розрахуємо додаткову заробітну плату:

$$Z_{\text{дод}} = 0,1 \cdot 10237,85 = 1023,78 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5...40% від суми їхньої основної та додаткової заробітної плати:

$$H_{зп} = (З_о + З_р) \cdot \frac{\beta}{100}, \quad (4.2)$$

$$H_{зп} = (10237,85 + 1023,78) \cdot \frac{36,3}{100} = 4087,97 \text{ (грн.)}.$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12}, \quad (4.3)$$

де:

Ц – балансова вартість обладнання, грн;

H_a – річна норма амортизаційних відрахувань % (для програмних засобів 25%);

T – Термін використання (T=3 міс.).

Таблиця 4.3 – Розрахунок амортизаційних відрахувань

| Найменування програмного забезпечення | Балансова вартість, грн. | Норма амортизації, % | Термін використання, міс. | Величина амортизаційних відрахувань, грн |
|---------------------------------------|--------------------------|----------------------|---------------------------|--|
| Персональний комп'ютер | 10000 | 25 | 3 | 562,5 |
| Всього: | | | | 562,5 |

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n N_i \cdot C_i \cdot K_i, \quad (4.4)$$

де:

n – кількість комплектуючих;

N_i - кількість комплектуючих i -го виду;

C_i – покупна ціна комплектуючих i -го виду, грн;

K_i – коефіцієнт транспортних витрат (прийmemo $K_i = 1,1$).

Таблиця 4.4 - Витрати на комплектуючі, що були використані для розробки ПЗ.

| Найменування матеріалу | Одиниці виміру | Ціна, грн. | Витрачено | Вартість витрачених матеріалів, грн. |
|--|----------------|------------|-----------|--------------------------------------|
| Флешка | шт. | 150 | 1 | 150 |
| Пачка паперу | уп. | 120 | 1 | 120 |
| Ручка | шт. | 10 | 1 | 10 |
| Всього з урахуванням транспортних витрат | | | | 308 |

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi} \quad (4.5)$$

де V – вартість 1кВт-години електроенергії ($V=1,7$ грн/кВт);

P – установлена потужність комп'ютера ($P=0,6$ кВт);

Φ – фактична кількість годин роботи комп'ютера ($\Phi=180$ год.);

K_{Π} – коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,8$).

$$V_e = 1,7 \cdot 0,6 \cdot 180 \cdot 0,8 = 146,88 \text{ (грн.)}$$

Розрахуємо інші витрати $V_{ін}$.

Інші витрати I_B можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$V_{ін} = (1..3) \cdot (Z_o + Z_p) \quad (4.6)$$

Отже, розрахуємо інші витрати:

$$V_{ін} = 1 \cdot (10237,85 + 1023,78) = 11261,63 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$V = Z_o + Z_d + H_{зп} + A + K + V_e + I_B$$

$$V = 10237,85 + 1023,78 + 4087,97 + 562,5 + 146,88 + 308 + 11261,63 = 27682,61 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи $V_{заг}$ за формулою:

$$V_{заг} = \frac{V_{ін}}{\alpha} \quad (4.7)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$V_{заг} = \frac{27682,61}{1} = 27682,61$$

Прогнозування загальних витрат $ЗВ$ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{V_{заг}}{\beta} \quad (4.8)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{27682,61}{0,9} = 30698,45 \text{ (грн.)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \Delta N)_i \quad (4.9)$$

де:

$\Delta\Pi_{\text{я}}$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технології зменшаться на 25 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 25 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 300 користувачів, протягом другого року – на 200 користувачів, протягом третього року – 100 користувачів. Реалізація інформаційної технології до впровадження результатів наукової розробки складала 1500 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 200 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого прибутку $\Delta\Pi_1$ протягом першого року складатиме:

$$\Delta\Pi_1 = 25 \cdot 1500 + (200 + 25) \cdot 300 = 105000 \text{ грн.}$$

Протягом другого року:

$$\Delta\Pi_2 = 25 \cdot 1500 + (200 + 25) \cdot (300 + 200) = 150000 \text{ грн.}$$

Протягом третього року:

$$\Delta\Pi_3 = 25 \cdot 1500 + (200 + 25) \cdot (300 + 200 + 100) = 172500 \text{ грн.}$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{\text{абс}}$ вкладених інвестицій розраховується за формулою:

$$E_{\text{абс}} = (\text{ПП} - PV), \quad (4.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої наукової роботи, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3, 4.

Рух платежів (інвестицій та додаткових прибутків) представлений на рисунку 4.1.

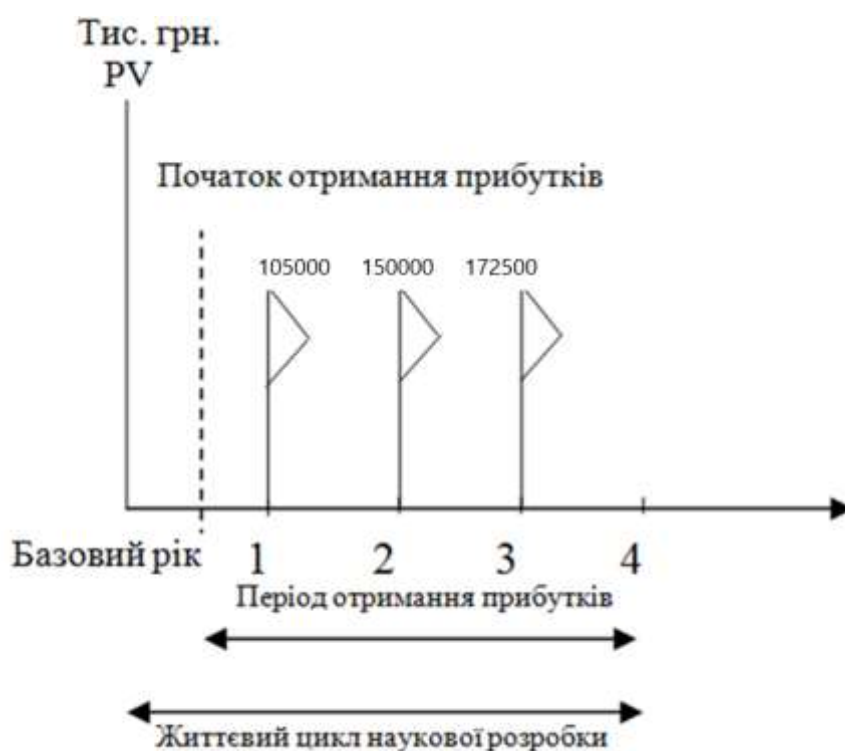


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів наукової роботи

Розрахуємо вартість чистих прибутків за формулою:

$$\text{ПП} = \sum_1^m \frac{\Delta\Pi_i}{(1+\tau)^t} \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої наукової роботи, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$\text{ПП} = \frac{30698,45}{(1+0,1)^0} + \frac{105000}{(1+0,1)^2} + \frac{150000}{(1+0,1)^3} + \frac{172500}{(1+0,1)^4} = 347992,34 \text{ (грн.)}$$

Тоді розрахуємо $E_{\text{абс}}$:

$$E_{\text{абс}} = 347992,34 - 30698,45 = 317293,89 \text{ грн.}$$

Оскільки $E_{\text{абс}} > 0$, то вкладання коштів на виконання та впровадження результатів наукової роботи буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_{\text{в}}$ за формулою:

$$E_{\text{в}} = \sqrt[T]{1 + \frac{E_{\text{абс}}}{\text{PV}}} - 1 \quad (4.12)$$

де $E_{\text{абс}}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій $\text{PV} = 3\text{В}$, грн;

$T_{\text{ж}}$ – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_B = \sqrt[3]{1 + \frac{317293,89}{30698,45}} - 1 = 1,24 \text{ або } 124 \%$$

Далі, розраховану величина E_B порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні $d = 0,2$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,1$.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки $E_B = 124\% > \tau_{\text{мін}} = 0,3 = 30\%$, то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{\text{ок}}$ розраховується за формулою:

$$T_{\text{ок}} = \frac{1}{E_B}$$

$$T_{\text{ок}} = \frac{1}{1,24} = 0,80 \text{ року}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

ВИСНОВКИ

В ході виконання магістерської кваліфікаційної роботи розроблено метод класифікації звуків пострілів на основі спайкінгової нейронної мережі. При аналізі предметної області було відзначено, що автоматична класифікація звуків можуть знайти своє застосування в багатьох сферах та можуть значно полегшити життя людям. Проте сьогодні не існує досконалого засобу, який дозволить з достатньою точністю (близько 95%) та швидкістю класифікувати сильно-корельовані звуки пострілів.

Визначено основні проблеми при аналізі характеристик звуків. Був проведений аналіз сучасних методів класифікації звуків пострілів, який показав, що класифікації звуків за допомогою з використанням штучних нейронних мереж та машинного навчання є найбільш точним та швидким методом.

Проаналізовано найвідоміші формальні моделі спайкінгових нейронних елементів, та обрано найбільш ефективний для задачі класифікації звуків пострілів. Розроблено структуру спайкінгової нейронної мережі, досліджено основні методи навчання спайкінгових нейронних мереж та обрано найбільш доцільний з власним удосконаленням його для зменшення часу навчання.

Було розроблено алгоритм роботи програмного засобу, на основі якого було реалізовано програмний засіб класифікації звуків пострілів на основі спайкінгової нейронної мережі за допомогою мови програмування Kotlin та середовища розробки IntelliJ IDEA.

Відібрано навчальну та тестову вибірку аудіофайлів, що дозволило в повній мірі продемонструвати можливості спайкінгової нейронної мережі класифікувати звуки пострілів. Тестування показало, що удосконалений метод навчання зменшує час навчання спайкінгової нейронної мережі в середньому на 20% порівняно зі звичайним методом STDP.

В результаті виконання даної кваліфікаційної роботи поставлені задачі були виконані в повній мірі. Отже, мета магістерської роботи досягнута.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кюльян І.Г. Використання імпульсних нейронних мереж у розпізнаванні динамічних образів. НТКП ВНТУ. Факультет інформаційних технологій та комп'ютерної інженерії. / Кюльян І.Г. [Електронний ресурс] – режим доступу : <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2018/paper/view/4113> (дата звернення 30.09.2019) – Назва з екрана.

2. Кюльян І.Г. Використання спайкінгових нейронних мереж у розпізнаванні образів. НТКП ВНТУ. Факультет інформаційних технологій та комп'ютерної інженерії. / Кюльян І.Г. [Електронний ресурс] – режим доступу : <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2019/paper/view/7285/> (дата звернення 30.09.2019) – Назва з екрана.

3. Кюльян І.Г. Dynamic image recognition using spiking neural networks. ВНТУ. Молодь в науці: дослідження, проблеми, перспективи (МН-2019). / Кюльян І.Г. [Електронний ресурс] – режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2019/paper/view/8157> (дата звернення 30.09.2019) – Назва з екрана.

4. Свідоцтво про реєстрацію авторського права на твір № 89126. Комп'ютерна програма «Моделювання спайкінгової нейронної мережі» / Колесницький О.К., Кюльян І.Г. Дата реєстрації Державною службою інтелектуальної власності України 30.05.2019.

5. Salamon, Justin & Jacoby, Christopher & Bello, Juan. A Dataset and Taxonomy for Urban Sound Research. Proceedings – 22nd ACM International Conference on Multimedia, 2014.

6. Sound Classification using Deep Learning [Електронний ресурс] – режим доступу: <https://github.com/mikesmales/Udacity-ML-Capstone> (дата звернення 30.09.2019) – Назва з екрана.

7. Gunshot sound classification using deep learning [Електронний ресурс] – режим доступу: <https://github.com/tusharsingh62/Gunshot-sound-classification-using-deep-learning> (дата звернення 30.09.2019) – Назва з екрана.

8. Сидоров К. В. Анализ признаков эмоционально окрашенной речи / К. В. Сидоров, Н. Н. Филатова. // Вестник ТвГТУ. – 2012. – №20. – С. 26–32.
9. Смоленцев Н. Основы теории вейвлетов. Вейвлеты в Matlab [Текст] / Н. Смоленцев. – М.: ДМК Пресс 2014. – 628 с.
10. Najmi A. The Continuous Wavelet Transform and Variable Resolution Time – Frequency Analysis [Text] / A. Najmi, J. Sadowsky // JOHNS HOPKINS APL TECHNICALDIGEST. – 1997. – Vol. 18, No. 1. – P. 134-140.
11. Вьюгин В.В. «Математические основы теории машинного обучения и прогнозирования» / В.В.Вьюгин М.: 2013. – 387 с.
12. Trevor H. The elements of statistical learning. Second edition / H. Trevor, R. Tibshirani, 2008.
13. Spiking Neural Networks: Principles and Challenges [Электронный ресурс] – режим доступа: <https://homepages.cwi.nl/~sbohte/publication/es201413Gruning.pdf>. (дата звернения 30.09.2019) – Назва з екрана.
14. W. Gerstner and W. Kistler. Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, 2002.
15. Импульсная нейронная сеть [Электронный ресурс] : [Веб-сайт]. – Режим доступа : http://ru.wikipedia.org/wiki/Импульсная_нейронная_сеть. (дата звернения 30.09.2019) – Назва з екрана.
16. Peter U. Diehl, Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity // Frontiers in Computational Neuroscience, August 2015, Volume 9, Article 99.
17. Sander M. Bohte; Joost N. Koko, Han La Poutr. Error-backpropagation in temporally encoded networks of spiking neurons // Neurocomputing, 48, 2002, pp. 17–37.
18. Esser S.K, Appuswamy R., Merolla P.A., Arthur J.V., Modha D.S. Backpropagation for Energy-Efficient Neuromorphic Computing // Proc. of Int. Conf. Neural Information Processing Systems 28, Montreal, Canada, 2015.
19. Ruf, B., Schmitt, M. Learning temporally encoded patterns in networks of spiking neurons // Neural Processing Letters, 5(1), 1997, pp. 9-18.

20. Kasinski A., Ponulak F. Comparison of Supervised Learning Methods for Spike Time Coding in Spiking Neural Networks // Int. J. Appl. Math. Comput. Sci., 2006, Vol. 16, No. 1, pp. 101–113.

21. Ponulak F., Kasiński A., Supervised learning in spiking neural networks with ReSuMe sequence learning, classification, and spike shifting // Neural Comput., 2010, vol. 22, pp. 467-510.

22. Ponulak F. Supervised learning in Spiking Neural Networks with ReSuMe Method. Doctoral Dissertation. Poznań University of Technology, 2006.

23. Mel frequency cepstral coefficients [Електронний ресурс] : [Веб-сайт]. – Режим доступу : https://en.wikipedia.org/wiki/Mel-frequency_cepstrum (дата звернення 30.09.2019) – Назва з екрана.

24. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville., 2016.

25. Брюс Екель. Філософія Java / Брюс Екель, 2015. – 1168с.

26. Середовище розробки «IntelliJ IDEA» [Електронний ресурс] : [Веб-сайт]. – режим доступу : https://en.wikipedia.org/wiki/IntelliJ_IDEA (дата звернення 30.09.2019) – Назва з екрана.