МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики
(повне найменування назва факультету)

Дискретного аналізу та інтелектуальних систем
(повна назва кафедри)

Магістерська робота

Проектування систем розпізнавання аудіоінформації за допомогою згорткових нейронних мереж

Виконав: студент групи ПМІМ-23
спеціальності
122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис) Андрус О.І.
(прізвище та ініціали)

Керівник (підпис) доц. Олійник Р.М.
(прізвище та ініціали)

Рецензент (підпис) Білецький В.М.
(прізвище та ініціали)

Львів – 2022

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет Прикладної математики та інформатики

Кафедра Дискретного аналізу та інтелектуальних систем

Спеціальність 122 — "Комп`ютерні науки"

(шифр і назва)

«ЗАТВЕРДЖУЮ»

Завідувач кафедри **Притула М.М.**

"__31__" ___серпня____ __2022__ року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Андрусу Олегу Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Проектування систем розпізнавання аудіоінформації за допомогою згорткових нейронних мереж

керівник роботи: доц. Олійник Роман Миколайович,

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

затверджені Вченою радою факультету від "13" __вересня__ 2022 року № 15

2. Строк подання студентом роботи 12.12.2022

3. Вихідні дані до роботи Офіційна документація Keras та Tensorflow. Python. платформа для машинного навчання Google Colab. відкриті бази даних Kaggle

4. Зміст магістерської роботи (перелік питань, які потрібно розробити)

1. Огляд предметної області;
2. Проектування системи розпізнавання музичних жанрів;
3. Проектування системи розпізнавання емоцій.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Архітектури нейронних мереж. Графіки отриманих результатів, зображення інтерфейсу користувача.

6. Консультанти розділів роботи

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

7. Дата видачі завдання 31 серпня 2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Огляд наявних досліджень у галузі штучного інтелекту	01.04.2022-01.05.2022	Виконано
2	Огляд публікацій про множинну класифікацію аудіосигналу	02.05.2022-20.05.2022	Виконано
3	Вивчення наявних на ринку рішень	21.05.2022-01.06.2022	Виконано
4	Формулювання вимог до системи розпізнавання жанрів	02.06.2022-14.06.2022	Виконано
5	Проектування архітектури та створення моделі	15.06.2022-01.07.2022	Виконано
6	Навчання нейромережі та оцінка результатів	02.07.2022-20.07.2022	Виконано
7	Побудова веб-застосунку, створення дизайну	20.07.2022-15.08.2022	Виконано
8	Тестування та виправлення помилок	15.08.2022-01.09.2022	Виконано
9	Формулювання вимог до системи розпізнавання емоцій	02.09.2022-14.09.2022	Виконано
10	Проектування архітектури та створення моделі	14.09.2022-01.10.2022	Виконано
11	Навчання нейромережі та оцінка результатів	02.10.2022-15.10.2022	Виконано
12	Побудова веб-застосунку, створення дизайну	16.10.2022-30.10.2022	Виконано
13	Тестування та виправлення помилок	30.10.2022-05.11.2022	Виконано
14	Перевірка правильності виконаної роботи	06.11.2022-08.11.2022	Виконано
15	Оформлення дипломної роботи	08.11.2022-01.12.2022	Виконано
16	Внесення правок	02.12.2022-05.12.2022	Виконано

Студент _____ **Андрус О.І.**
(підпис) (прізвище та ініціали)

Керівник роботи _____ **Олійник Р.М.**
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Актуальність теми: сучасні комп'ютери та мобільні пристрої володіють все більшою кількістю пам'яті та обчислювальних ресурсів аніж це було десятиліття тому. У зв'язку з цим доводиться обробляти, автоматизовувати та ефективно зберігати все більше інформації, зокрема у аудіо форматі. Таким чином однією з важливих задач є класифікації звукових сигналів на основі певних критеріїв чи характеристик. На даному етапі наукового прогресу це питання широко досліджується як за допомогою класичних методів так і з застосуванням нейронних мереж, проте дослідження теми залишається актуальним.

Мета курсової роботи: вдосконалення процесу класифікації аудіоінформації, а саме емоцій та музичних жанрів, у різних сферах застосування за допомогою методів глибинного навчання.

Об'єкт дослідження: програмні засоби для класифікації аудіосигналів за певними характеристиками (музичний жанр, емоція) із використанням згорткових нейронних мереж.

Методи дослідження: наукові досягнення в дослідженні хвильових спектрів аудіосигналу та в областях машинного навчання.

Область застосування: проект та висновки даної роботи можуть бути застосовані як у мобільних, так і веб застосунках для формування бібліотеки аудіо файлів, а також у кол-центрах для автоматичного розпізнавання рівня задоволеності клієнтів.

Задачі дослідження:

1. Аналіз наявних алгоритмів класифікації звуку;
2. Формування та попередня обробка даних;
3. Розробка алгоритмів на базі згорткових нейронних мереж;
4. Розробка серверних додатків із нейронною мережею та зручним інтерфейсом користувача.

Ключові слова: штучні нейронні мережі, спектрограма, згорткові нейронні мережі, глибинне навчання, аудіо дані, категоризація, множинна класифікація, tensorflow, python, keras, класифікація аудіофайлів, розпізнавання емоцій.

ABSTRACT

Topicality of the study: modern computers and mobile devices have more and more memory and computing resources than a decade ago. In this regard, it is necessary to process, automate and efficiently store more information, in particular in audio format. Thus, one of the important tasks is sound signals classification based on certain criteria or characteristics. At this stage of scientific progress, this issue is widely investigated both with the help of classical methods and with the use of neural networks, but the research of the topic remains relevant.

The aim of the diploma's paper: is to improve the audio information classification process, namely in the categorization of emotions and musical genres, in various fields of application using deep learning methods.

The object of the research: is software tools for the classification of audio signals according to certain characteristics (musical genre, emotion) using convolutional neural networks.

Research methods: scientific advances in the study of audio signal wave spectra and in the fields of machine learning.

Field of application: the project and conclusions of this research can be applied both in mobile and web applications to form a library of audio files, as well as in call centres to automatically recognise the level of customer satisfaction.

Research objectives:

1. Analysis of existing sound classification algorithms;
2. Formation and preliminary processing of data;
3. Development of an algorithms based on convolutional neural networks;
4. Development of a server applications with a neural network and convenient user interface.

Keywords: artificial neural networks, spectrogram, convolutional neural networks, deep learning, audio data, categorisation, multiple classification, tensor flow, python, keras, audio file classification, emotion recognition.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ШНМ — Штучна нейронна мережа

ЗНМ — Згорткова нейронна мережа

MFC — Mel-frequency cepstral (мел-частотний кепстр)

MFCCs — Mel-frequency cepstral coefficients (мел-частотні коефіцієнти)

API — Applied program interface (інтерфейс прикладного рівня)

MGR — Music genre recognition (розпізнавання музичних жанрів)

MVP — Minimum valuable product (мінімально життєзданий продукт)

TESS — Toronto emotional speech set

SAVEE — Surrey Audio-Visual Expressed Emotion

RAVDESS — Ryerson Audio-Visual Database of Emotional Speech and Song

ВСТУП

Відповідно до закону Мура, ми знаходимося на тому етапі обчислювальних потужностей, коли стало можливим обробляти терабайти інформації на пристрої, що поміщається у долоню. Кількість та різноманіття інформації, зокрема аудіоінформації (дані з сенсорів, музика, записи телефонних розмов тощо), невпинно зростає з кожною хвилиною.

Навіть більше, обсяг даних має навіть не лінійний, а скоріш експоненціальний приріст. Як наслідок, зростає необхідність у створенні таких систем та технологій, здатних обробляти, класифікувати, оцінювати аудіосигнали автоматизовано.

Проектування та впровадження в активне використання продуктів для розпізнавання мови, класифікації музики за жанровою приналежністю чи емоційною складовою набувають все більшого розповсюдження. Перспективним напрямком для апробації таких підходів став інтернет речей, що може бути яскраво представлений “розумними” будинками чи автопілотами, де команди потрібно віддавати за допомогою голосу.

У даній роботі буде розглянуто програмні засоби для класифікації аудіоінформації за певними характеристиками із використанням штучного інтелекту. Безпосереднім об’єктом дослідження буде класифікація за двома параметрами — жанрова приналежність та емоція.

Характеристики розпізнавання обрано з огляду на численність сфер застосування. Необхідність класифікувати об’єкти за жанровою приналежністю виникла у зв’язку з потребою у створенні музичних активів та пришвидшення індексування громіздких колекцій.

Класифікація емоцій на основі вхідного аудіосигналу широко застосовується у логістичних компаніях та кол-центрах для оцінювання якості наданих послуг та рівня задоволеності клієнтів.

На даному етапі наукового прогресу подібні задачі досліджуються як за допомогою класичних методів так і з застосуванням нейронних мереж. Ручні методи класифікації не здатні відповідати сучасним викликам та вимогам.

Класифікація аудіосигналу — це проблема розпізнавання образів, яка включає в себе функції вилучення і створення класифікатора[1]. ШНМ активно використовується для розпізнавання образів завдяки високій здатності до узагальнення.

Оскільки ми працюватимемо із згортковими нейронними мережами, котрі здебільшого використовують для роботи із зображеннями, буде розглянуто поняття спектрограми. Також буде висвітлено роль даних у машинному навчанні та методи регуляризації для оптимізації нейромереж.

Практичне значення отриманих результатів полягатиме у розробці серверного додатку із зготковою нейронною мережею та зручним інтерфейсом користувача, котрий дасть можливість отримати жанр та емоцію завантаженого користувачем аудіофайлу.

1 ОСНОВНІ ХАРАКТЕРИСТИКИ ЗВУКУ

1.1 Цифрове відображення аудіосигналу

У фізичному плані аудіосигнал — це складне за формою коливання, котре можна описати залежністю амплітуди звукової хвилі від часу. Цифрове відображення аудіосигналу є результатом перетворення аналогового сигналу (неперервного в часі) звукового діапазону в цифровий формат звуку. Процес перетворення називають аналогово-цифровим перетворенням.

Процес дискретизації за часом — це процес отримання точкових значень аналогового сигналу з певним кроком у часі, так званим кроком дискретизації.

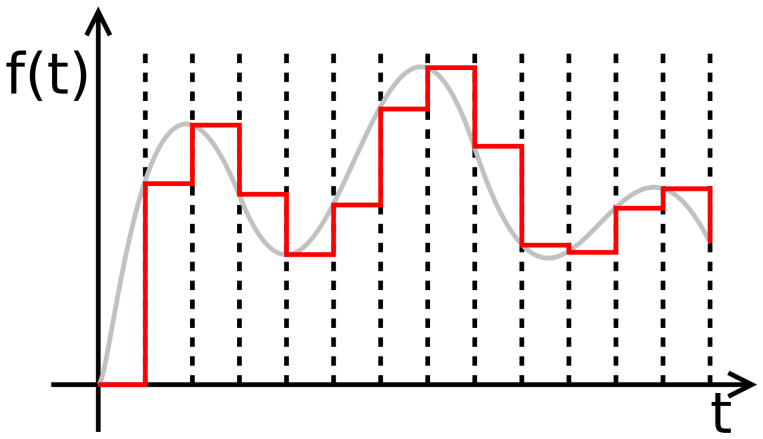


Рисунок 1.1.1 — Дискретизація сигналу

Кількість здійснюваних в одну секунду замірів величини сигналу називають частотою дискретизації або частотою вибірки. Чим менший крок дискретизації сигналу, тим більшою буде частота і відповідно більш точне відображення сигналу.

Існує теорема відліків Найквіста-Шеннона, котра говорить, що для того, щоб відновити сигнал за його відліками без втрат, необхідно, щоб частота дискретизації була хоча б вдвічі більшою за максимальну частоту первинного неперервного сигналу[10].

$$F_{\text{д}} \geq 2Fmax \qquad (1.1.1)$$

Наприклад слух людини здатний сприймати сигнал у діапазоні від 20 Гц до 40 КГц. У цьому випадку оптимальна частота дискретизації становитиме 40 КГц. За стандарт взято значення 44.1 КГц.

Варто зазначити, що хоч оцифрування аудіо сигналу у вигляді чисельних рядів можливе, записати виміряні значення сигналу із ідеальною точністю неможливо. Звідси виникли формати файлів для зберігання аудіоінформації, найбільш популярним серед яких є MP3.

У ньому використовується алгоритм стиснення з втратами, що призводить до втрати певної частини інформації. Таке кодування призводить до того, що оцифрований сигнал при відтворенні звучить подібно до оригінального, але його теоретично не можна назвати ідентичним. MP3 формат балансує між істотним зменшенням розміру даних та якістю вдтворення звуку.

1.2 Класифікація ключових ознак

Наступним етапом у класифікації оцифрованого аудіосигналу є виділення ключових ознак. Ключовими ознаками у машинному навчанні називають вимірювані властивості, характеристики чи параметри спостережуваного об’єкта.

Цілком природно, що спосіб їх представлення здебільшого описується можливостями математичного апарату, а саме n-вимірним вектором числових ознак. Наочним прикладом може слугувати описова змінна у лінійній регресії.

У задачах розпізнавання мовлення до ознак можна віднести рівень шуму, тривалість звуків, потужність сигналу, збіг із фільтрами, динаміка зміни голосу, розподіл частот тощо. Характеристики звуку об’єднано у три групи, а саме: спектрально-часові, амплітудно-частотні, кепстральні.

Спектрально-часові ознаки описують аудіосигнал з фізико-математичної перспективи, а саме як складне за формою коливання. У цьому випадку розглядаються періодичні, або ж тональні, ділянки звукової хвилі та неперіодичні, або ж шумові, що не містять голосових пауз.

Дані характеристики дозволяють побачити своєрідність форми для спектра і тимчасового ряду імпульсів голосу. До спектрально-часових ознак відносять наступні:

- варіація огинаючих спектрів мовлення;
- нормалізований час перебування аудіосигналу у конкретній ділянці спектру;
- медіанне значення спектра голосу;
- відносна потужність спектра мови;
- коефіцієнти кореляції спектральних огинаючих між ділянками спектра;
- коефіцієнт форми сегмента;
- висота семента;
- тривалість сегмента.

Амплітудно-частотні ознаки[24] найчастіше використовують для отримання даних, які можуть змінюватися у залежності від зміщення фрейму вибірки або ж параметрів дискретного перетворення Фур'я, а саме форми та ширини вікна.

Аудіосигнал як коливання описується частотою, тобто кількість коливань у секунду, інтенсивністю, або ж амплітудою коливання, і тривалістю. До таких коливань відносять:

- енергія;
- модуляція за допомогою амплітуди — шиммер (англ. "shimmer");
- тремтіння частотної модуляції основного тону— джіттер (англ. "jitter");
- інтенсивність та амплітуда;
- частота основного тону.

Кепстральні ознаки отримали назву від мел-частотного кепстру (MFC). MFC — це представлення короткочасного спектру потужності звуку, яке використовує лінійне косинусне перетворення на нелінійній мел-шкалі частот.

Мел-шкала емпірична, нелінійна, частотна і використовується для формування мел-частотних коефіцієнтів (MFCC) при тренуванні нейронних мереж, призначених для розпізнавання мови [4].

Алгоритм отримання MFCC можна описати наступним чином:

1. відобразити аудіосигнал за допомогою перетворення Фур’є у мел-шкалі;
2. прологарифмувати у кожній шкалі Мела;
3. виконати дискретне косинусне перетворення.

Таким чином можна отримати амплітуду сигналу у кожному спектрі. Крім MFCC до кепстральних ознак також відносять коефіцієнти:

- для позначення потужності частоти фіксувань;
- кепстра лінійного передбачення;
- для спектра лінійного прогнозування.

Варто зазначити, що кепстральний аналіз найчастіше використовують саме у машинному навчанні. Широке застосування мел-частотних коефіцієнтів зумовлене достатньо високою роздільністю між звуками. Отримані коефіцієнти використовують для побудови спектограм.

1.3 Побудова спектрограми

У попередньому розділі було розглянуто аналогово-цифрове перетворення аудіосигналу. Для збереження інформації у більшості випадків достатньо MP3 формату. Проте варто зазначити, що масив значень амплітуд у часі (так звані фрейми), що зберігається в MP3, не зручний для рекурентних нейронних мереж. Для класифікації аудіо файлів використовують спектрограми.

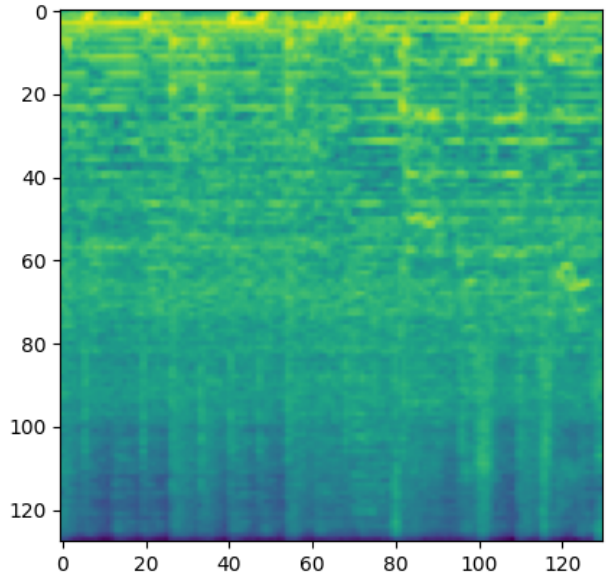


Рисунок.1.3.1 — Відрізок спектрограми аудіо файлу

Спектрограма — це візуальне зображення спектру частот сигналу в часі. При застосуванні до звукового сигналу спектрограми іноді називають сонографами, голосовими відбитками або голосограмами. На отриманому зображенні представлена залежність спектральної щільності потужності сигналу від часу.

Простіше кажучи, це двовимірний графік із часом на горизонтальній осі та частотою на вертикальній осі. Колір кожної точки зображення представляє значення амплітуди на певній частоті в певний момент.

Для побудови спектрограми використовують два способи:

1. апроксимація як набір фільтрів;
2. віконне перетворення Фур’є.

Заззначимо, що до появи сучасних цифрових методів обробки сигналів можливим було лише використання фільтрів. Наведені вище способи для побудови спектрограм насправді створюють різні квадратичні частотно-часові розподілу, але за певних умов можуть бути еквівалентними.

На практиці використовують віконне перетворення Фур’є. Для формування спектрограми сигнал розбивають на частини і потім проводять перетворення Фур’є з метою отримати значення частотного спектра для кожної ділянки.

Варто зазначити, що при великому об’ємі вхідних даних варто використовувати mel спектрограми. MEL - це кількісна одиниця виміру висоти звуку, яка описує сприйняття звукових хвиль людиною і залежить від частоти:

$$m = 1127.01048\, ln(1 + \frac{f}{700}) \qquad (1.3.1)$$

де - мел одиниця виміру висоти звуку,

f - частота звуку.

Таке представлення дозволяє виділити ті частоти, які є найбільш значимими для класифікації, та зменшити кількість параметрів на відміну від звичайної спектрограми та часового представлення сигналу. Можна сказати, що mel спектограма відображає частоту частот.

1.4 Віконне перетворення Фур’є

Віконне перетворення Фур’є — це перетворення Фур’є, яке використовується для визначення синусоїдальної частоти та фазового вмісту локальних частин сигналу зі змінними в часі властивостями. Простіше кажучи, це операція, яка відображає одну функцію реальних змінних на іншу функцію.

Математичне подання такого перетворення виглядає наступним чином:

$$F(t,w) = \int_{-\infty}^{+\infty} f(i) * W(i - t) * e^{-iwt} dx \qquad (1.4.1)$$

де $W(i - t)$ - деяка віконна фунція.

Функція описує коефіцієнти (“амплітуди”) при розкладанні вихідної функції на елементарні складові - гармонійні коливання з різними частотами.

На практиці перетворення Фур’є у більшості випадків замінюється на дискретне перетворення Фур’є, а як вікно беруть одну з функцій-перетворень:

1. синус в кубі;
2. прямокутне;
3. Гаусса;
4. Хеммніга;

5. Ханна.

Візьмемо, наприклад, вікно Ханна. Зафіксуємо ширину M вікна Ханна:

$$g_M(j) = \frac{1}{2} * (1 - cos\frac{2\pi j}{M}) \quad (1.4.2)$$

Ділянка сигналу, локалізована поблизу моменту часу k, визначається формулою:

$$x_M(j,k) = x_{k+j} g_m(j) \quad (1.4.3)$$

Дискретне перетворення Фур'є можна знайти наступним чином:

$$X_M(n,k) = \sum_{j=0}^{M} x_M(j,k) W^{-jn}{}_M, \quad (1.4.4)$$

де $W_n = exp[i(\frac{2\pi}{N})]$

2 ШТУЧНІ НЕЙРОННІ МЕРЕЖІ. ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1. Історія досліджень у галузі штучного інтелекту

Дослідження у галузі штучного інтелекту можна звести до пошуку відповіді на запитання: «Чи може машина (пізніше комп'ютерна програма) самонавчатись, приймати рішення, подавати ознаки розумної структури на подобу людського розуму?»

До вирішення поставленої задачі були залучені спеціалісти різних галузей (машинне навчання виникло на перетині таких наук як когнітивна психологія, математика, біологія, хімія, програмування та інші) — Воррен Маккалох, Дональд Гебб, Френк Розенблат та інші.

Перші спроби симуляції процесу мислення людини стали можливими завдяки новим апаратним можливостям в інформатиці та електроніці. У 1943 році інноваційний підхід було висвітлено у публікації нейрофізіолога Уоррена Маккалоха та математика Уолтера Піттса про штучні нейрони та їх представлення за допомогою електричних схем [28].

Водночас із прогресом в нейроанатомії і нейрофізіології психологами було створено моделі людського навчання. У 1950-ті —1960-ті роки група дослідників на чолі з Натаніелем Рочестером об'єднали наявні методики та підходи у публікацію "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence" [17].

Мінські, Розенблатт, Відроу та інші вчені розробили мережі, що складаються з одного шару штучних нейронів, які вони назвали перцептронами[14]. Ці мережі у порівнянні із сучасними хоч і не були надзвичайно ефективними, проте уже тоді могли використовуватися у задачах прогнозування (наприклад, погоди).

Здавалося б, що ключ до інтелекту знайдено, а симуляція роботи людського мозку є лише питанням часу та конструювання більш складних систем. Проте

тогочасні мережі мали низьку здатність до узагальнення і могли бути застосовні лише у вузькому колі задачі.

Інтерес до штучних нейронних мереж стрімко зріс з 1980-х років. Фахівці з таких галузей, як інженерія, філософія, фізіологія та психологія, заінтриговані можливостями, які пропонує ця технологія, і активно прагнуть застосувати її у своїх дисциплінах. У 2007 році Джеффрі Хінтон з університету м. Торонто створив алгоритми глибокого навчання для багатошарових нейронних мереж[12].

Це поклало початок активному використанню ШНМ у різних сферах людської діяльності та стало початком розвитку технологій штучного інтелекту. Дослідження тривають і досі, тому увесь потенціал цієї галузі остаточно ще не розкритий.

2.2 Нейромережі як імітація людського мозку

Як було відзначено раніше, ШНМ — це математична модель роботи людського мозку, реалізована на основі програмно-апаратних засобів для створення штучного інтелекту. Подібно до мозку, що складається із густої сітки нейронних клітин (приблизно 10,000,000,000 одиниць), ШНМ складається із математичних моделей біологічних нейронів.

Основне завдання такої складової — це приймати сигнали з вхідних шарів мережі, опрацьовувати та передавати їх у наступні штучні нейрони[3]. Біологічні нейрони складаються із розгалуженої структури:

- дендритів (приймає сигнали);
- аксонів (передає сигнал далі);
- ядер (опрацьовує сигнал).

Синапс з’єднує аксони. У синапсах відбувається посилення чи послаблення електрохімічного сигналу. Зв’язки між штучними нейронами називають за аналогією синаптичними, або ж просто синапсами. Синапс приймає один параметр — ваговий коефіцієнт, який відповідає за коригування зміни інформації при передачі між нейронами.

Саме завдяки цьому вхідні дані опрацьовуються і на виході перетворюються у результат, а навчання полягає у тому, щоб експериментально підібрати оптимальні значення вагового коефіцієнта для кожного синапса, що рано чи пізно призведе до отримання бажаного результату.

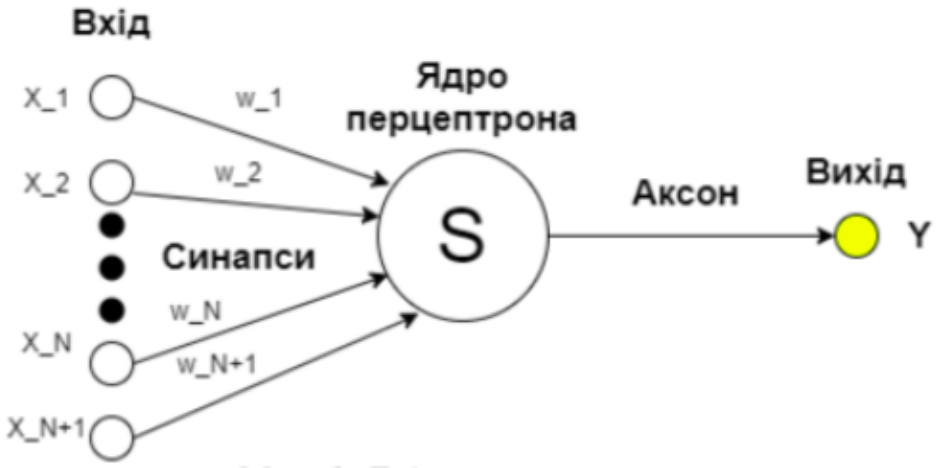
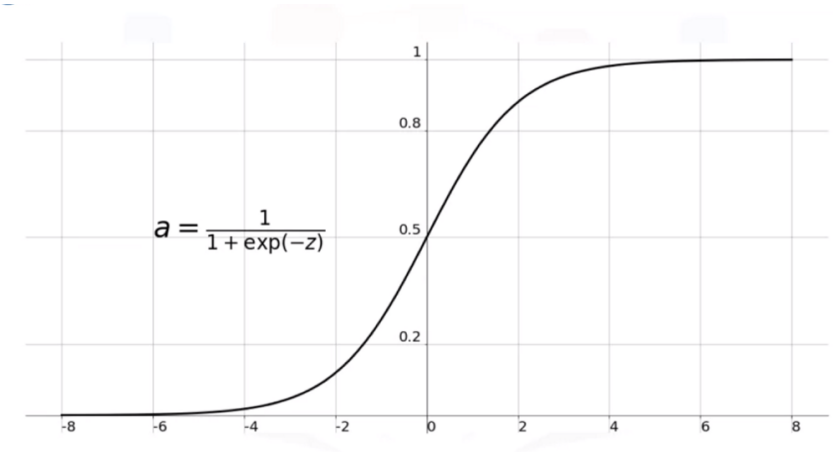


Рисунок 2.2.1 — Будова перцептрона

Наочним прикладом може слугувати математична модель сигмоідального перцептрона, що побудована на перцептроні Френка Розенблата. На цьому прикладі видно, що перцептрон складаєтсья з N входів. Розенблат запропонував обраховувати значення на виході за простим правилом, а саме: позначити ваги $w1, w2, \ldots, wN$, дійсні числа, що відображають значимість синапсу для вихідного значення та зміщення b.

У звичайному перцептроні Розенблата на вихідному шарі можливі два варіанти відповіді: 0 або 1 залежно від критичного значення. Звідси вводиться нова змінна $\sigma(w * x + b)$, де $\sigma(x)$ — це сигмоідальна функція, визначена наступним чином

$$\sigma(x) = \frac{1}{1+ exp(-\sum_j w_j x_j - b)} = \frac{1}{1+e^{-x}} \qquad (2.2.1)$$



Рискунок 2.2.2 — Графік сигмоїди

Для узагальнення описаних понять можна взяти за приклад 3-шарову нейронну мережу. Нейрони вхідного шару отримують дані ззовні (наприклад, від камери відеоспостереження чи будь-якої сенсорної системи) і після опрацювання отриманих даних передають сигнали через синапси до наступних шарів нейронів[3].

Наступні групи нейронів, окрім вихідної, називають прихованими, оскільки вони безпосередньо не пов’язані ні з вхідним, ні з вихідним шарами ШНМ. Приховані нейрони обробляють отримані сигнали і передають їх на вихід системи. Кожен процесор вхідного рівня пов’язаний з одним або більше процесорами прихованого рівня, кожен з яких, у свою чергу, пов’язаний з одним або ж декількома процесорами вихідного рівня.

Такого типу структура ШНМ піддається навчанню (переоцінка ваг за рахунок зміщення b) і придатна для виділення простих зв’язків у наборі даних. Кількість зв’язків та прихованих шарів не обмежена, що дає простір для експериментів та досліджень.

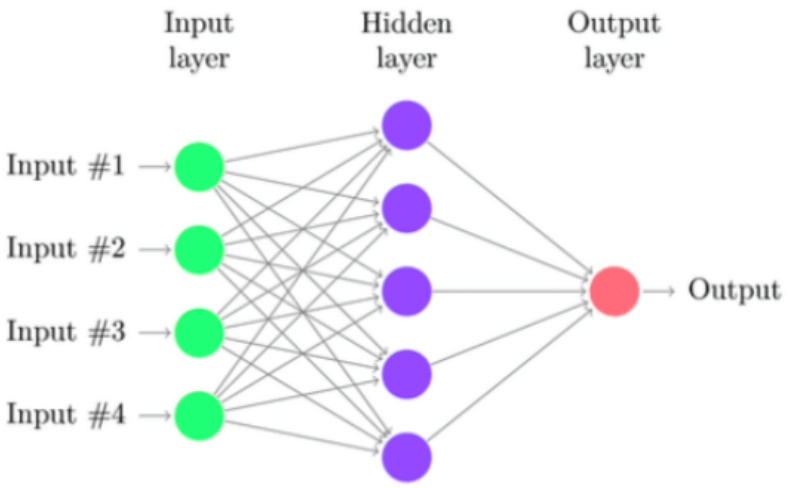


Рисунок 2.2.3 — Базова архітектура ШНМ

2.3 Процес навчання ШНМ

Особливість та унікальність нейромережі як спроби імітувати біологічний розум полягають у здатності до навчання на основі вхідної вибірки. У залежності від кількості та якості інформації про задану предметну область ШНМ у результаті навчання може самостійно підвищувати власну продуктивність відповідно до певних правил. Такий процес відбувається за умови двох факторів, а саме: визначення архітектури нейромережі та ітеративної адаптації вагових коефіцієнтів при нейронах.

У навчанні ШНМ можна виділити наступні кроки:

1. сигнал (інформація) надходить із зовнішнього середовища;
2. вагові коефіцієнти при нейронах змінюються залежно від ітерації та функцій активації;
3. після зміни внутрішньої структури ШНМ відповідає на вхідні сигнали вже іншим чином.

Зазначимо, що правила, які задають зміну вагових коефіцієнтів, називають алгоритмом навчання. Природно припустити, що універсального правила не існує у зв’язку із різноманіттям архітектур нейромереж. Найбільш поширеним правилом незмінно залишається метод спроб та помилок.

Алгоритми навчання можна узагальнено описати на два класи, а саме: детермновані та стохастичні. Для перших притаманна наявність чітких меж та жорстких правил, а для других контрольована рандомізація[2].

Також концептуально виділяють три парадигми:

- навчання із вчителем, або ж контрольоване навчання;
- навчання без вчителя, або самонавчання;
- навчання з підкріпленням.

Здебільшого ШНМ використовують навчання із вчителем, де вихід, що змінюється, постійно порівнюється із бажаним входом. При першій ітерації вагові коефіцієнти при нейронах ініціалізовані випадковим чином, проте із кожною новою епохою змінюються, допоки різниця між бажаним та поточним результатом не дорівнюватиме заданому пороговому значенню.

Очевидно, що для безпосереднього використання нейромережа має пройти процес навчання, щоб знайти оптимальні ваги для вирішення поставленої задачі. При досягнені бажаного результата коефіцієнти фіксуються та зберігаються для подальшого використання.

Недоліком такого підходу є детермінованість навчання, тобто під час безпосереднього використання якість ШНМ не змінюється. Таким чином при навчанні із вчителем точність дуже залежить від якості та повноти тренувальних даних.

У класичному розумінні навчання без вчителя краще описує спосіб опрацювання інформації біологічним мозком, адже відбувається у недетермінованому режимі і у реальному часі. Простими словами, штучна нейронна мережа розвивається неперервно у ході безпосереднього використання, а не лише під час тренування.

На відміну від підходу з вчителем немає відомих вхідних та вихідних значень для побудови моделі між заданим входом та очікуваним виходом. ШНМ незалежні від зовнішніх факторів, а самі виділяють патерни та правила для коректування ваг, виділяючи загальні тенденції, та роблять адаптацію у залежності

від навчальної функції. Незважаючи на відсутність правильних відповідей, нейромежа повинна мати інформацію про власну організацію, що визначається функціями активації, топологією та архітектурою.

Алгоритми у цій навчальній парадигмі намагаються кластеризувати нейрони, які працюють разом. Якщо вхідний сигнал активує будь-який вузол у групі, сумарна дія в загальному збільшується. Справедливе і обернене твердження.

В основі процесу навчання лежить конкуренція між нейронами. Таким чином коректуються ваги лише для нейрона-переможця.

До найбільш вживаних алгоритимів відносять:

- кластеризація методом k-середніх;
- метод найближчих сусідів;
- метод виявлення аномалій;
- ієрархічна кластеризація.

Навчання з підкріпленням, або ж змішане навчання, є поєднанням двох описаних раніше підходів. У цьому випадку частина ваг при нейронах визначається за допомогою вчителя, а іншу ШНМ обраховує самостійно.

2.4 Методи підвищення точності ШНМ

При імплементації парадигм навчання ШНМ, розглянутих у попередньому розділі, на практиці доволі часто виникає проблема перенавчання (анг. “overfitting”). Суть цього явища у регресійному аналізі полягає у тому, що статистична модель описує випадкову похибку замість безпосереднього взаємозв’язку між параметрами.

В області штучного інтелекту перенавчання проявляється у надмірно близькій підгонці навчальної вибірки до тестової. Це призводить до того, що нейромережа втрачає здатність до узагальнення. Ідентифікатором перенавчання є суттєва різниця у точності моделі на навчальній вибірці та нових даних.

Замість того, щоб вчитися класифікувати нові приклади, модель адаптувалася до уже відомих прикладів.

Причинами виникнення перенавчання можуть бути:

1. кількість ітерацій;
2. недостатнья кількість прикладів у тренувальній вибірці;
3. громіздка структура мережі.

Одне із найпростіших рішень проблеми полягає у поділі вхідних даних на дві окремі вибірки: навчальну та тестову. Перша множина використовується у алгоритмі навчання, а інша для перевірки адекватності побудованої моделі. Варто зазначити, що обидві вибірки не повинні перетинатися.

Таким чином при використанні окремих множин можна побачити зміну похибки прогнозу на тестовій множині парельно зі спостереженнями на навчальній. До певної ітерації похибка зменшується для обох вибірок, допоки не почне зростати на тестовій, а на навчальній - спадати.

У цей момент ШНМ аналізує “шум” замість виділення нових ознак, що і є початком перенавчання. Тоді навчання потрібно завершити для отримання найбільшої точності прогнозу.

2.4.1 Регуляризація

Одним із способів оптимізації ШНМ є використання регуляризації. Термін регуляризація у задачах статистики та штучного інтелекту позначає модифікацію даних, параметрів, архітектури, функції витрат тощо для зменшення кількості помилок на тестовій множині[23]. Іншими словами, регуляризація - це будь-яка техніка, мета якої покращити здатність нейромережі до узагальнення.

Найчастіше говорять про регуляризацію функції витрат, метою якої є спростити модель шляхом зменшення вагових коефіцієнтів при найбільш активних нейронах. Таким чином у цільовій функції вводиться поняття “штрафу” для параметрів.

Тобто значення тих нейронів, що виділяють шум замість ключових ознак, будуть штрафуватися або навіть прирівнюватися до нуля. Варто зазначити, що варто уникати надмірної лінійності моделі, щоб уникнути недонавчання (англ. “underfitting”).

Найбільш відомими є L1 (англ. “Lasso regression”) та L2 (Тихонова) регуляризації [9]. Обидва підходи оновлюють функцію витрат шляхом додавання регуляторного параметра. Загальний випадок для L1 та L2 виглядає наступним чином:

$$Cost\ function\ =\ Loss\ +\ Regularization \qquad (2.4.1.1)$$

Тому для L1 отримаємо:

$$Cost\ function\ =\ \frac{1}{2}\sum_{i=1}^{N}(f(x_i w)\ -\ y_i)^2\ +\ \lambda\ *\ \sum_{j=0}^{M}\left|w_j\right|, \qquad (2.4.1.2)$$

де w - вектор ваг поліному;

λ - коефіцієнт регуляризації.

Регуляризація Тихонова:

$$Cost\ function\ =\ \frac{1}{2}\sum_{i=1}^{N}(f(x_i w)\ -\ y_i)^2\ +\ \frac{\lambda}{2}\ *\ \sum_{j=0}^{M}\left|w_j\right|^2, \qquad (2.4.1.3)$$

де $\left|w_j\right|^2$ - квадратична норма вектору ваг.

Обидва методи виглядають подібно і їхня ефективність залежить від конкретної задачі. Наприклад, L1 проводить селекцію ознак, тобто значення нейронів, що не додають до загальної точності моделі, можуть обнулятися, чого не відбувається при L2. L1 ефективний при однорідності та простій структурі даних, а L2 при більш складних даних[16].

Не менш використовуваним методом регуляризації є виключення (англ. “dropout”)[6]. Основа відмінність dropout від попередніх підходів полягає у тому, що L1 та L2 модифікують навчальну множину та коефіцієнти ваг, а dropout змінює структуру мережі. Це є прикладом регуляризації на архітектурному рівні.

Алгоритм можна схематично описати наступним чином:

1. у процесі навчання випадковим чином виділяють підмережу;
2. на кожній ітерації відкидаються нейрони на основі коефіцієнта виключення;
3. як наслідок конкуренції, нейрони-переможці отримують більшу вагу.

У результаті кожна підмережа навчалася по-різному, що зменшує ймовірність перенавчання моделі вцілому.

2.4.2 Аугментація

У випадку регуляризаціїї на основі даних відбувається:

1. балансування - кількість екземплярів рівномірно розподіляєтсья між усіма класами;
2. перетворення - покращення якості екземплярів за допомогою методів обробки зображень (наприклад, використання операторів дискретного диференціювання для виділення контурів - Собеля, Превіта тощо);
3. поділ - у випадку з класифікацією аудіоінформації звукову доріжку можна розділити на менші частини, що збільшить загальну кількість елементів навчальної множини;
4. аугментування - застосування різноманітних операцій для генерування нових екземплярів з уже наявних.

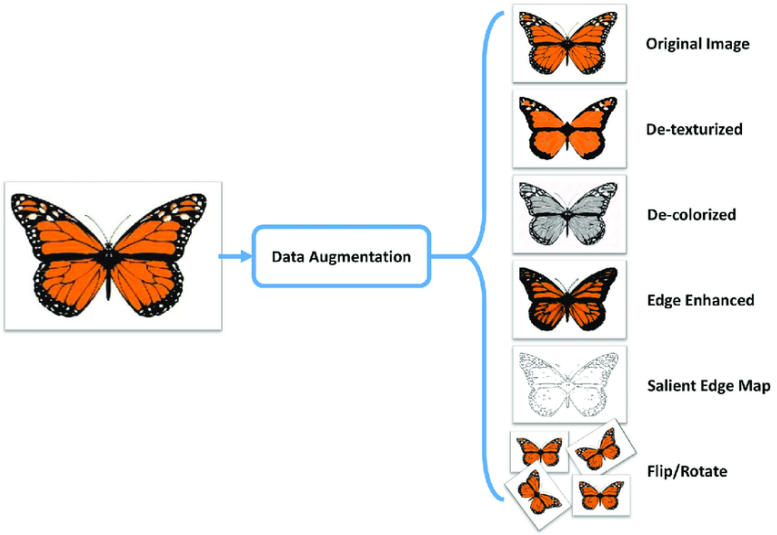


Рисунок 2.4.2.1 — Аугментація даних

Закономірно вважати, що якість та кількість даних безпосередньо впливають на точність прогнозування. Наукові публікації [31][29] наочно демонструють застосування прийому аугментації у медицині, де здебільшого бази даних невеликі, що зумовлено чутливістю інформації та ціною її отримання.

Найбільш використовуваними методами аугментації є [21]:

1. Повороти та симетрія. Незважаючи на те, що для людського ока такі перетворення суттєво змінюють сприйняття зображженя, проте для машинного навчання такі маніпуляції вважають інваріантними для алгоритму;
2. Застосування Гаусового шуму. Із допомогою розподілу Гауса випадковим чином генеруємо маску, яка накладаєтсья на зображення;
3. Зміна контрасту та яскравості зображення;
4. Гамма-корекція - нелінійна операція, що використовується для кодування та декодування яскравості [29];
5. Афінна трансформація - геометричні перетворення, що зберігають паралельність та пропорційність прямих і векторів[20].

За рахунок збільшення та урізноманітнення датасету модель менш схильна до перенавчання та зменшується ймовірність виділення шуму замість ключових ознак. Таким чином можна покращити здібність ШНМ до узагальнення.

2.5 Згорткова нейронна мережа

Згорткові нейронні мережі — це клас глибинних штучних нейронних мереж прямого поширення, який успішно застосовується для аналізу візуальних зображень[8]. Така модель є одним із найвідповідніших інструментів для класифікації (як бінарної так і мультимножинної) об’єктів, облич на фотографіях, розпізнавання мови тощо.

Свою назву мережа отримала від операції, що називається згортка і часто використовується для оброблення зображень. Шар згортки перемножує значення

фільтра на вихідні значення пікселів зображення (поелементне множення), після чого всі добутки сумуються, тобто цей шар виконує згортку зображення [5].

Цей підхід дозволяє зменшити обсяг інформації, тому ШНМ може краще обробляти зображення з вищою роздільною здатністю та виділяти ключові характеристики зображення, такі як краї, силуети чи контури обличчя. На наступному рівні обробки з цих країв і граней можна ідентифікувати повторювані фрагменти текстури, які потім об'єднують у фрагменти зображення.

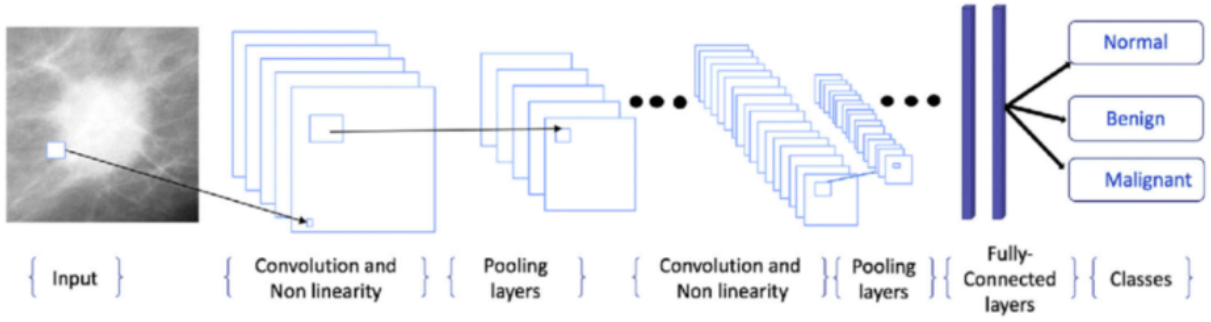


Рисунок 2.5.1 —Загальна структура згоркової нейронної мережі

Згорткова нейронна мережа (ЗНМ) – тип багатошарової нейронної мережі, що може бути описана наступною формулою[5]:

$$(f \times g)[m.n] = \sum_{k,l} f(m - k, n - l] * g[k, l] \qquad (2.5.1)$$

де f – матриця зображення на вході,

g – матриця для операції згортки.

Простими словами, цю операцію можна описати наступним чином: вікном розміром g проходимо з заданим кроком все зображення f, перемножуючи поелементно на кожному кроці вміст вікна на ядро g та сумуючи отримані результати.

Розмір ядра може бути довільним, проте здебільшого використовують квадрат у межах від $3x3$ до $7x7$. Варто зауважити, що від розміру ядра залежить якість виділених ознак. Наприклад, якщо взяти маленьку сітку, то неможливо буде якісно виділити ознаки, а у випадку великого ядра кількість зв'язків між

нейронами може зрости кратно, що вплине на точність прогнозу[5]. Під час самого навчання ШНМ визначає потрібні ядра згортки на основі даних на вході, тому здебільшого усе відбувається автоматично.

Структура мережі – односпрямована (без зворотних зв’язків), багатошарова[19]. Згорткові нейромережі здебільшого складаються з:

- згорткові (англ. “convolutional”);
- субдискретизуючі (англ. “subsampling”, підвибірка);
- прошарки «звичайної» нейронної мережі – персептрона.

Архітектура згорткових нейронних мереж реалізує три ідеї, які забезпечують інваріантність мережі до невеликих зрушень, змін масштабу і спотворень:

1. кожен нейрон отримує вхідний сигнал від локального рецептивного поля попереднього шару. Це гарантує локальну двовимірну зв’язність нейронів;
2. кожен прихований шар мережі складається з безлічі карт ознак, на яких всі нейрони мають загальні ваги, що забезпечує інваріантність до зміщення і суттєве зменшення кількості вагових коефіцієнтів мережі;
3. за кожним шаром згортки слідує обчислювальний шар, який здійснює локальне усереднення та підвибірку. Це дає ефект сповільнення розширення для карт ознак.

Для повноцінної роботи згорткові ШНМ використовують карти ознак та фільтри.

Фільтром називають матрицю, що представляє характеристики вихідного зображення. Існує верхній та нижній фільтри. Перший необхідний для того, щоб визначити частини вихідного зображення, які мають вертикальні лінії, а другий, щоб визначити частини зображення, які мають горизонтальні лінії.

Процес виявлення безпосередньо базується на операції згортання фільтра вихідного зображення. Результат згортки визначає положення вихідних елементів зображення, яке називається картою ознак.

Метою згорткового процесу є зменшення розмірності карти ознак до такої міри, щоб повний набір функцій міг бути оброблений мережею прямого розповсюдження (у більшості випадків багатошаровим перцептроном).

Згортковий рівень реалізує ідею локального рецептивного поля, тобто кожен вихідний нейрон підключений лише до певної (невеликої) області вхідної матриці, тим самим імітуючи деякі характеристики людського зору.

Для підвищення ефективності роботи ЗНМ необхідно знайти оптимальне значення наступних параметрів:

- початкова ініціалізація ваг;
- розмір вікна;
- кількість карт ознак;
- щільність зв'язків між картами ознак.

3 ОГЛЯД ФРЕЙМВОРКІВ ТА ПРОГРАМНИХ КОМПОНЕНТ

Для вирішення поставленої задачі потрібно вибрати відповідні програмні засоби для роботи з аудіофайлами, побудови нейронних мереж та математичних операцій. Для програмної реалізації було використано наступний стек технологій, а саме: Python3, Opencv, Dllib, Keras, Tensorflow та NumPy (див. мал. 1).

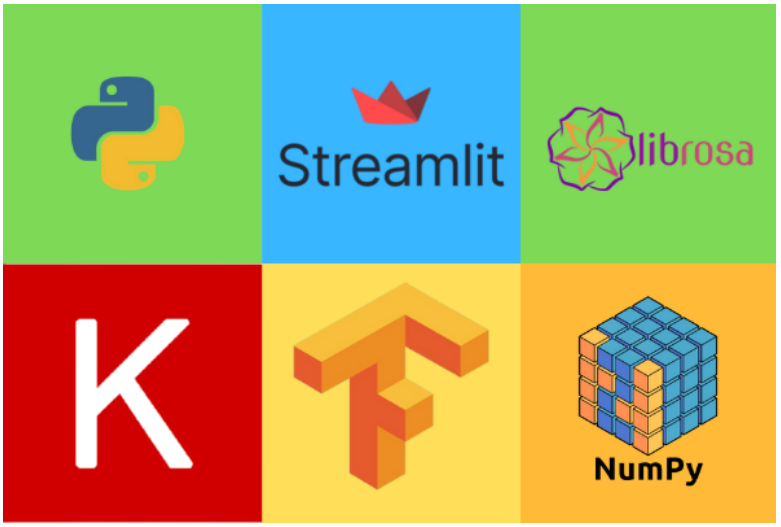


Рисунок 3.1. — Стек використаних технологій

Серед наведеного:

Librosa та pydub — пакети для обробки та аналізу аудіофайлів. Зокрема, завдяки librosa можна сформувати спектрограми та отримати ключові характеристики аудіосигналу. Pydub, у свою чергу, можна використати для обрізання, об’єднання файлів.

Streamlit — бібліотека Python із відкритим кодом призначена для створення веб-застосунків із мінімальними конфігураціями. Даний модуль набув популярності серед експертів машинного навчання для побудови зручного інтерфейсу користувача. Більшість компонент доступні “out of the box”.

Tensorflow[26] — це відкрита програмна бібліотека для машинного навчання, розроблена компанією Google для задоволення її потреб у системах, здатних будувати та тренувати нейронні мережі для виявлення та розшифрування

образів та кореляцій, аналогічно до навчання й розуміння, які застосовують люди. Платформа була розроблена командою Google Brain і використовується в сервісах Google для розпізнавання мови, виділення облич на фотографіях, визначення схожості зображень, відсіювання спаму на Gmail, підбору новин і змістовного перекладу тексту. Бібліотека написана на мовах Python та C++, тому цілком природно, що Tensorflow забезпечує програмний прикладний інтерфейс для Python, C++, Haskell, Java, Go, що спрощує процес розробки.

Keras[18] — одна із найбільш вживаних бібліотек Tensorflow, чий API використовують виключно для побудови нейронних мереж. Основні будівельні блоки для побудови нейронної мережі (шари, цільові та передавальні функції) братимемо саме звідси.

Numpy — розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій із цими масивами.

Pillow — бібліотека Python, призначена для роботи з растровою графікою. Модуль надає зручний інтерфейс для маніпуляцій із зображеннями: I/O операції, конвертування форматів тощо.

Kaggle — платформа для змагань з аналітики та передбачувального моделювання, у рамках якого інженери даних конкурують у створенні найкращих моделей для прогнозування та опису даних, запропонованих компаніями або користувачами. Kaggle надає зручний доступ до баз даних, котрі використовують для навчання нейромереж.

4 ПРОЕКТУВАННЯ СИСТЕМИ РОЗПІЗНАВАННЯ МУЗИЧНИХ ЖАНРІВ

4.1 Постановка задачі

Основною задачею даного розділу є проектування системи розпізнавання музичних жанрів, що складатиметься з наступних кроків:

1. Аналіз наявних алгоритмів класифікації звуку;
2. Формування та попередня обробка даних;
    a. збільшення кількості зразків за допомогою поділу файлів на відрізки;
    b. формування спектрограм;
    c. створення вибірок для тренування та валідації.
3. Розробка алгоритму на базі згорткових нейронних мереж;
    a. створення архітектури моделі;
    b. процес навчавння з вчителем;
    c. збереження ваг із найкращими результатами.
4. Аналіз отриманих результатів;
5. Розробка серверного додатку із зручним інтерфейсом користувача.

4.2 Обробка та аналіз даних

Для досягнення вимог поставленої задачі необхідно сформувати базу даних для навчання згорткової нейронної мережі. У цій роботі як джерело даних використано платформу Kaggle, яка агрегує набори датасетів для різноманітних задач.

Для дослідження було використано базу даних GTZAN [15]. Датасет структуровано наступним чином:

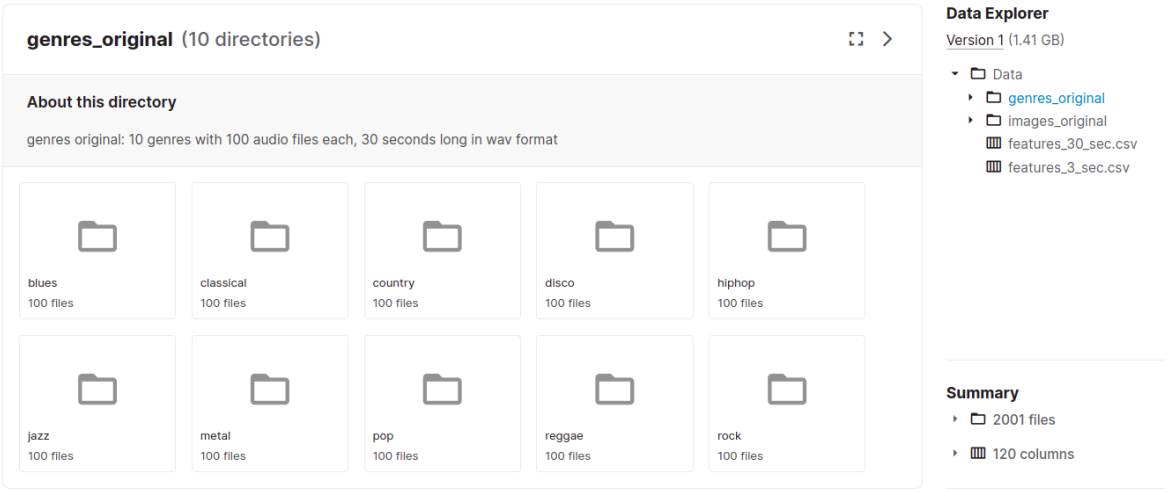


Рисунок 4.2.1. — Структура GTZAN датасету

GTZAN містить 1000 аудіофайлів (див. **genres_original**) у форматі WAV (Waveform Audio Format), поділених на 10 директорій, де назва відповідає музичному жанру (блюз, джаз, класика тощо). Для кожного елемента датасету побудовано спектрограму (див. **images_original**), де назва спектрограми відповідає назві аудіофайлу. Таблиці **features_30sec** та **features_3_sec** містять ключові ознаки файлів, для яких обчислено середнє значення вибірки та дисперсія). Об'єм необхідної пам'яті становить 1.41 Gb.

У нейромережах точність отриманих результатів залежить від кількості та якості вхідних даних. У зв'язку з цим було поділено кожен із аудіофайлів на сегмент тривалістю 3 секунди кожен. Для цього використано модуль **pydub**.

```
newAudio = AudioSegment.from_wav(song)
new = newAudio[t1:t2] # from 0 to 3, 3-6
```

Рисунок 4.2.2. — Сегментація аудіофайлу

У результаті сформовано вибірку **audio3sec**, що у 10 разів більша ніж **genres_original.** Для кожного отриманого елемента бази даних необхідно сформувати відповідні спектрограми. Використовуючи функціонал бібліотеки librosa, можна доволі швидко побудувати 10000 спектрограм:

```
y, sr = librosa.load(song_path, duration=3)
mels = librosa.feature.melspectrogram(y=y, sr=sr)
```

Рисунок 4.2.3. — Генерація спектрограми

Отримані результати збережено у директорії **spectrograms3sec** відповідно до назв категорій та аудіофайлів. Далі сформуємо вибірки для навчання. Для цього поділемо спектрограми на вибірки для тренування та валідації у відношенні 9:1 відповідно.

Попередньо рекомендовано випадковим чином перемішати елементи вибірки, щоб запобігти монотонності даних. Такий підхід збільшує ймовірність того, що у навчальній вибірці аудіофайли усіх жанрів будуть розподілені неодноманітно.

За допомогою keras сформуємо генератори даних для отриманих вибірок із створених **spectrograms3sec/train**, **spectrograms3sec/test** директорій. Keras автоматично співставляє спектрограми із категоріями на основі назв папок, у яких вони знаходяться. Оскільки класифікація відбувається на 10 категорій, то кожна із них маркується натуральним числом від 0 до 10.

4.3 Архітектура нейромережі

Одним із найважливіших і у той же час найменш передбачуваних моментів побудови ШНМ є вибір оптимальної відносно ресурсів та вимог архітектури нейронної мережі. У цьому випадку домінує часто застосовний в експериментальних дослідженнях метод спроб і невдач.

Емпіричний досвід розробників свідчить, що для побудови ШНМ ефективно використовувати послідовне (англ. “Sequential”) задання елементів, де вихідні дані одного шару стають вхідними даними наступного, що полегшує процес відлагодження помилок та оптимізації.

У даному дослідженні мережа побудована наступним чином:

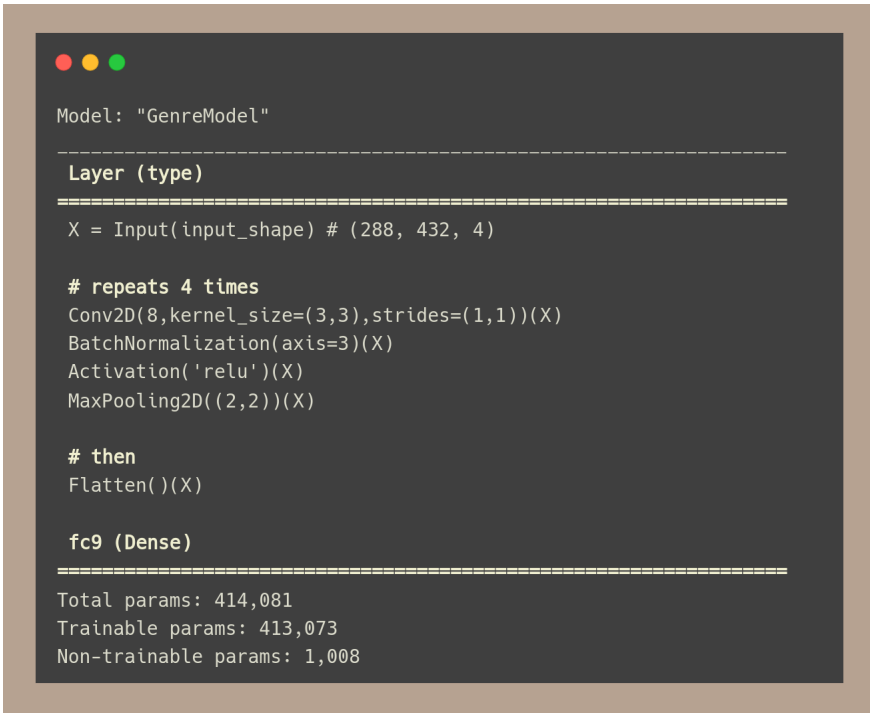


Рисунок 4.3.1. — Архітектура мережі

Архітектура нейронної мережі здебільшого складається із згорткових шарів (із відносно невеликим розміром ядра), шарів агрегації та звичайних щільних шарів. Короткий опис деяких з шарів подано нижче:

Convolutional later (Conv2D) — двовимірний шар згортки (наприклад, просторова згортка зображень). Операція згортки у нашому випадку відбувається над спектрограмами розмірністю 128x128 px із застосуванням функції активації «relu».

Relu — напівлінійна функція активація, яка повертає значення x, якщо x додатне, у протилежному випадку отримуємо 0.

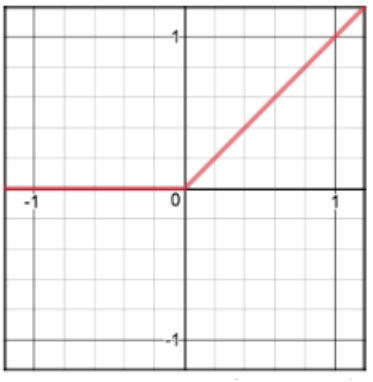


Рисунок 4.3.2. — Графік напівлінійної функції активації

MaxPooling2d layer — агрегувальний шар, або ж шар субдискретизації. Головне його завдання полягає у зменшенні розмірності даних з одночасним збереженням найважливіших характеристик шляхом формування залежності між кількома елементами (нейронами) з попереднього шару з єдиним елементом поточного шару.

Dense layer — звичайний щільний шар перцептронів із заданою функцією активації. Останнім шаром рекомендовано обирати Dense із одним нейроном (відповідає за значення отриманої ймовірності) та сигмоїдальною функцією активації.

Flatten layer — змінює розмірність даних.

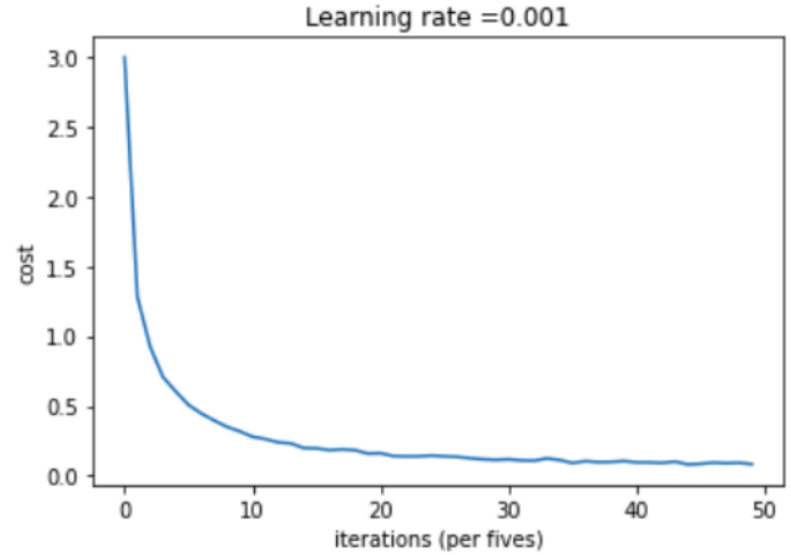
4.4 Навчання нейронної мережі та аналіз отриманих результатів

Оскільки генератори даних та архітектура згорткової нейронної мережі готові, можна перейти до етапу навчання. Навчання є ітеративним, кожну ітерацію прийнято називати епохами. Епоха у такому контексті відповідає за одну ітерацію у процесі навчання.

Під час однієї епохи увесь датасет проходить через нейронну мережу у прямому і зворотному (стохастичний спуск) напрямку один раз. З огляду на те, що одночасно тримати у пам’яті увесь датасет під час проходження однієї ітерації не зовсім доцільно, то датасет ділять на підвибірки — батчі.

У нашому випадку кількість епох дорівнює 50, а розмір батчу - 128 спектрограм. Варто зазначити, що час навчання залежить від обчислювальних можливостей комп’ютера.

Після декількох годин навчання отримали наступні результати на тренувальній та тестувальній вибірках:




```
Parameters have been trained!
Train Accuracy: 0.98724747
Test Accuracy: 0.8848485
```

Рисунок 4.4.1. — Графік втрат та точність

На основі даного графіка втрат можна сказати, що після десятої епохи результати точність моделі не збільшувалася суттєво. На тестовій вибірці точність склала 98%, а на тестовій - 88%.

4.5 Побудова веб-застосунку

Оскільки процес навчання потребує багато обчислювальних ресурсів, то збережемо ваги натренованої моделі. Це дозволить нам при побудові веб-застосунку завантажувати ваги у модель замість того, щоб тренувати її знову, адже часто доводиться перезапускати проект або випускати оновлення.

Для побудови мінімально життєздатного продукту (далі MVP) використаємо фреймворк streamlit. Варто зазначити, що streamlit часто використовують для

швидкої розробки застосунків, котрі використовують алгоритми глибинного навчання. Більшість компонент доступні “out-of-the-box”: кнопки, меню, аудіопрогравач тощо.

MVP повинен задовольняти наступні вимоги:

1. надати можливість завантажувати аудіофайли з власного пристрою;
2. мати чіткий та інтуїтивно зрозумілий інтерфейс користувача;
3. містити графіки із результатами, а саме генерувати спектрограму та графік подібності;
4. містити приклади для короткого демо;
5. мати можливість прослухати завантажений аудіофайл;

При проектуванні варто взяти до уваги той факт, що мережа тренувалася на аудіофайлах довжиною у три секунди формату .**wav.** З цих міркувань додано бізнес-логіку для конвертації та вибору оптимального сегмента (зазвичай достатньо взяти середину).

Протестуємо модель на прикладі композицій “Nirvana-Smells Like Teen Spirit”, жанром якої є рок, “Bob Marley - War” - реггі та “Tailor Swift - Love Story” - кантрі. Отримаємо наступні результати:

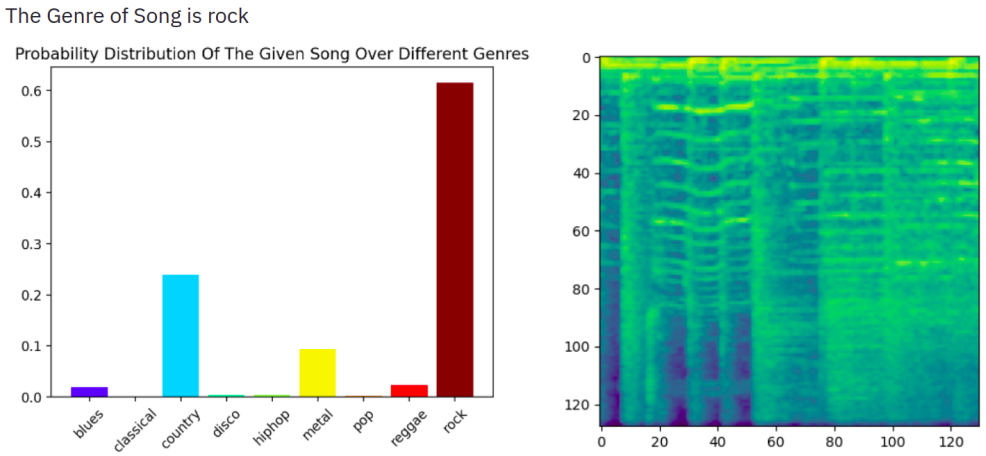


Рисунок 4.5.1 — Графік подібності та спектрограма для рок-пісні

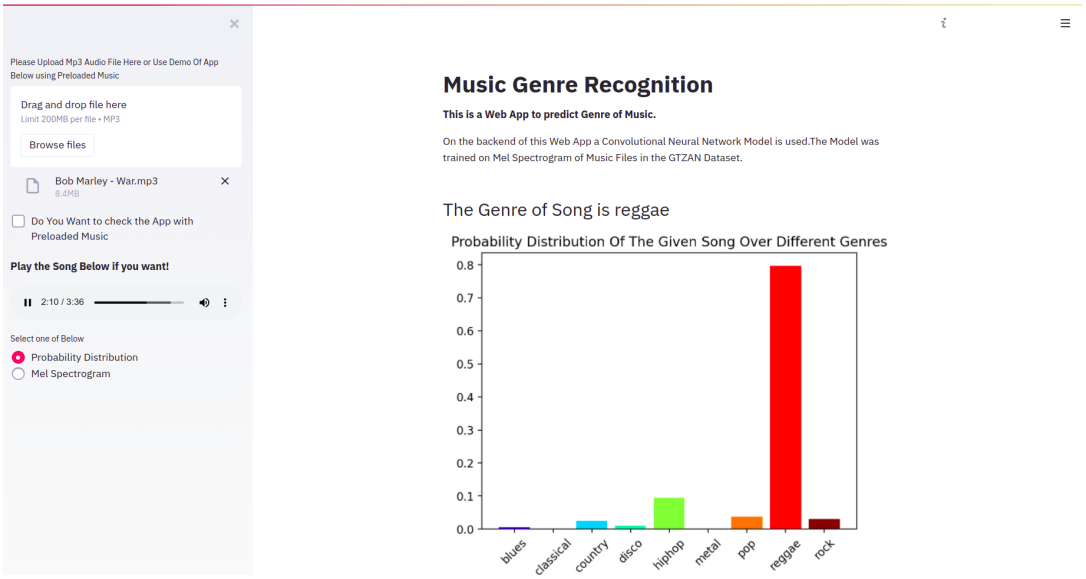


Рисунок 4.5.2. — Інтерфейс для завантаженого файлу “Bob Marley - War”

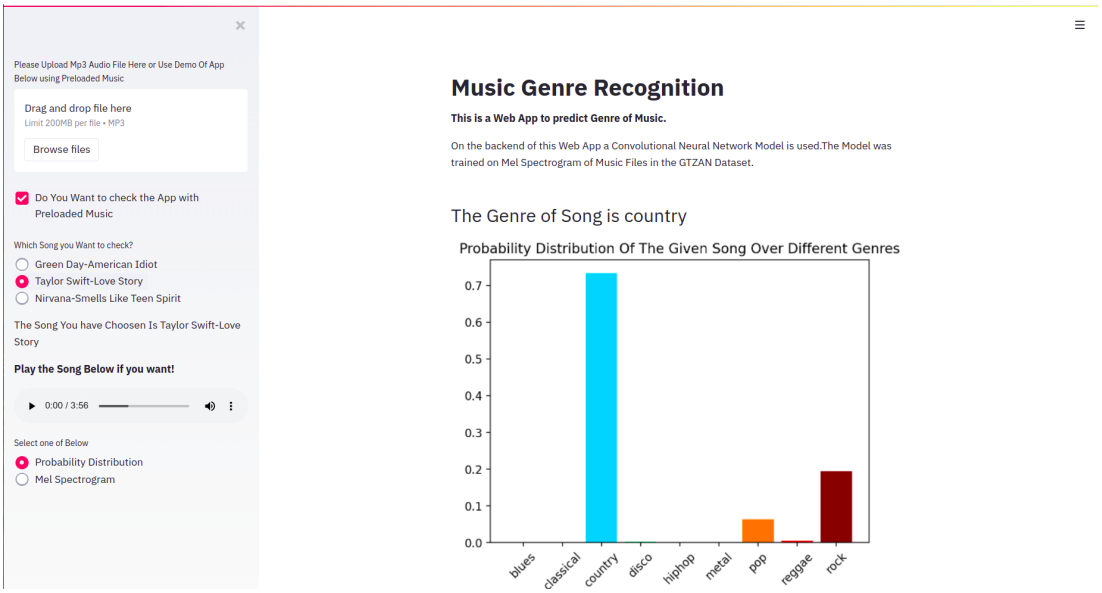


Рисунок 4.5.3. — Інтерфейс для завантаженого файлу “Taylor Swift-Love Story”

Варто зазначити, що інколи аудіофайл може володіти характеристиками декількох жанрів одночасно, що цілком природно. Попри 88% точність, система робить помилки на піснях, котрі суттєво відрізняються від тренувальної вибірки.

Цього можна було б уникнути, маючи більшу базу даних та/або використовуючи, наприклад, навчання без вчителя, що є цікавим для подальшого вивчення. Проте на даному етапі умови поставленої задачі виконано.

4.6 Висновки до розділу

У розділі “4 ПРОЕКТУВАННЯ СИСТЕМИ РОЗПІЗНАВАННЯ МУЗИЧНИХ ЖАНРІВ” розглянуто теоретичні та практичні засади побудови системи класифікації музичних жанрів. Дотримано усіх вимог, окреслених у постановці задачі, а саме:

- розглянуто методику оцифровування аудіосигналів та алгоритм формування спектрограм;
- проаналізовано якісні та кількісні показники бази даних, продемонстровано способи збільшення датасету за допомогою сегментації і використання генераторів на основі директорій;
- розроблено архітектуру згорткової нейронної мережі, ваги якої збережено для повторного використання;
- описано особливості та результати навчання нейромережі;
- описано мінімально життєздатний продукт у вигляді веб-застосунку із зручним користувацьким інтерфейсом.

Отримані знання та підходи можна також застосовувати в інших областях глибинного навчання, а саме: розпізнавання мови чи емоцій. Логічним продовженням дослідження може бути пошук аудіофайлу не за жанровою приналежністю, а конкретно за назвою.

Прототипом може слугувати сервіс “Shazam”, котрий дозволяє у режимі реального часу за допомогою мікрофона телефона визначити назву пісні. Така розробка вимагатиме у рази більшої кількості даних та ймовірно інших підходів глибинного навчання, наприклад, навчання без вчителя.

У зв’язку з тим, що обчислювальні потужності комп’ютерів дозволяють опрацьовувати все більше даних, дослідження продовжуються, а нейронні мережі пропонують нове рішення для класифікації музичних аудіо файлів. Саме тому ця тема залишається актуальною і надалі.

5 ПРОЕКТУВАННЯ СИСТЕМИ КЛАСИФІКАЦІЇ ЕМОЦІЙ

5.1 Постановка задачі

У даному розділі буде розглянуто процес проектування системи для розпізнавання емоцій, що включатиме наступне:

1. Формування та обробка даних для навчання;
2. Опис основних понять про емоції та способи їх представлення;
3. Побудова архітектур згорткових нейромереж, а саме:
    a. для класифікації шести основних емоцій;
    b. для визначеннння статі мовця;
    c. для визначення рівня задоволеності.
4. Аналіз отриманих результатів;
5. Збереження вагових коефіцієнтів для подальшого використання;
6. Побудова веб-застосунку із зручним інферфейсом користувача.

5.2 Про емоції

Емоції є невід’ємною частиною життя кожної людини. Навіть більше, при правильному їх тлумаченні ми можемо розуміти одне одного, співпереживати та правильно комунікувати. Емоцію можна описати як переживання людиною себе, навколишнього світу.

Саме тому для штучного інтелекту та науки вцілому розпізнавання емоцій мало неабиякий інтерес. Результатом десятирічних досліджень стало застосування до даної задачі таких методологій, як Баєсова мережа [30], прихована мережа Маркова [7], нейронні мережі тощо.

Аристотель виділив шість базових емоцій [13], а саме:

1. Гнів;
2. Страх;
3. Огида;
4. Щастя;

5. Сум;
6. Здивування.

Даний список не є вичерпним, адже на перетині вище наведених емоцій виділяють й інші.

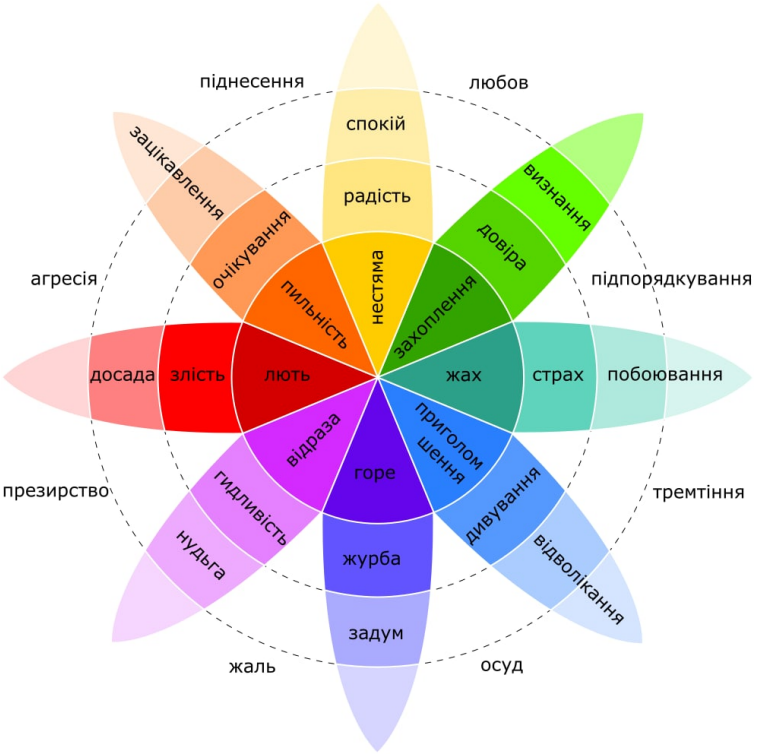


Рисунок 5.2.1. — Колесо емоцій Плутчика

Варто зазначити, що у поставленій задачі буде також розглянуто класифікацію на основі:

- рівня задоволеності
  - позитивний (об’єднання щастя та здивування);
  - нейтральний;
  - негативний (об’єднання гніву, страху, огиди та суму).
- статі
  - чоловік;
  - жінка.

5.3 Огляд даних

Як було уже зазначено раніше, для розв’язування поставленої задачі необхідно мати достньо вичерпну, збалансовану та різноманітну колекцію екземплярів для навчання. Оскільки якісних та підготовлених датасетів у відкритому доступі обмежена кількість, то вибірка сформована як комбінація трьох датасетів, а саме: TESS [27], SAVEE [25], RAVDESS [22]. Варто зауважити, що завантажити бази даних та ознайомитися із основними характеристиками можна безкоштовно на платформі Kaggle.

Toronto emotional speech set (TESS) можна вважати основним за кількістю екземплярів - 2800. Датасет містить 1-3 секундні аудіофайли із записами голосів двох акторок, котрі умовно позначені як OAF та YAF. Відповідно дані посортовані по 14 директоріям (7 емоцій * 2 актриси) , 200 файлів кожна. Загальний об’єм даних становить 281 Мб. Особливістю цієї бази даних є рівномірний розподіл між усіма класами.

Surrey Audio-Visual Expressed Emotion (SAVEE) складаєтсья із записів чотирьох чоловічих голосів (DC, JE, JK, KL відповідно). Для кожної з семи емоцій по 15 записів, що у результатів дає 420 екземплярів. Загальний об’єм даних дорівнює 152 Мб.

Ryerson Audio-Visual Database of Emotional Speech/Song (RAVDESS) містить 1400 аудіфайлів. Участь у проекті брало 24 актори, серед яких 12 чоловіків та жінок. Кількість емоцій та сама, що й у попередніх вибірках. Цікавим є той факт що актори промовлють лише дві фрази, виражаючи відповідну емоцію. Загальний об’єм даних становить 590 Мб.

Варто зазначити, що окрім шести класів для емоцій, виділено також додаткову нейтральну категорію, яка об’єднює файли без чітко вираженої емоції. Об’єднавши наведеніі вище датасети, отримаємо збалансовану вибірку, зображену на рисунку 5.3.1.

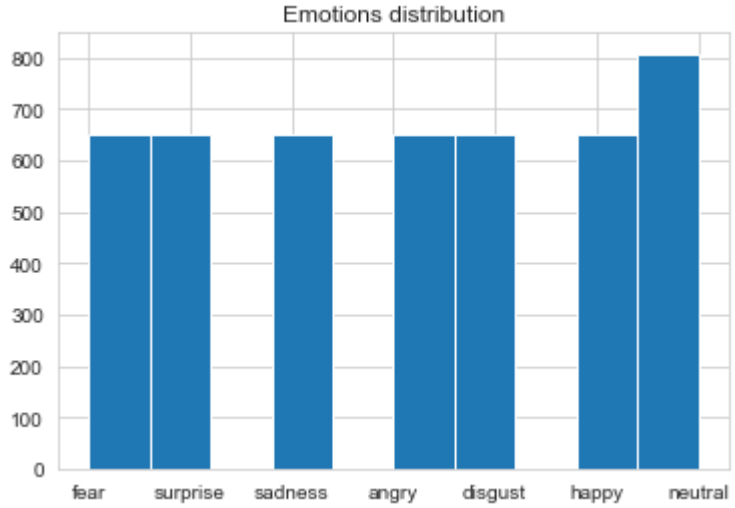


Рисунок 5.3.1. — Загальна вибірка

Довільним чином обравши тестовий файл, наведемо графік розподілу частот, графік кепстральних коефіцієнтів (отримати коефіцієнти можна за допомогою функції librosa.feature.mfcc) та мел-спектрограму:

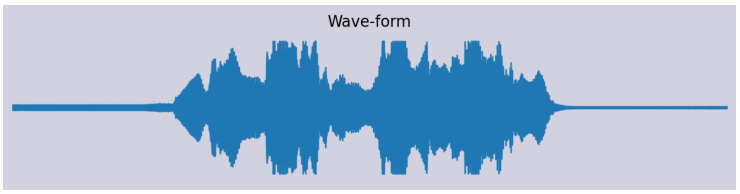


Рисунок 5.3.2. — Графік розподілу частот

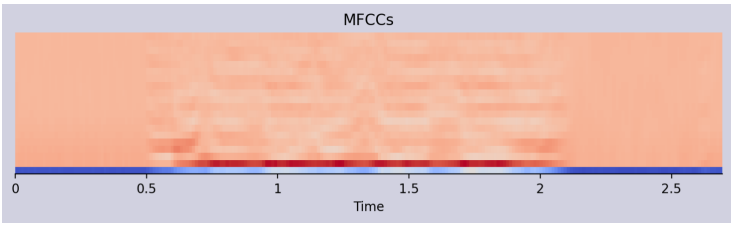


Рисунок 5.3.3. — Графік кепстральних коефіцієнтів

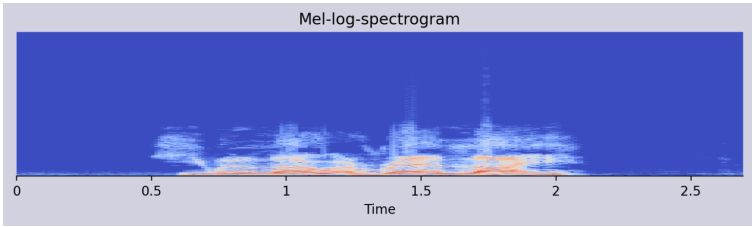


Рисунок 5.3.4. — Спектрограма

5.4 Архітектура ШНМ

Оскільки система класифікації емоцій повинна виділяти одночасно рівень задоволеності та сім базових емоцій, то потрібно побудувати дві різні моделі. Позначимо їх model_positivity та model_seven_emotions відповідно.

На малюнку 23 наведено скорочений опис та послідовність шарів, а також кількість нейронів у кожному з них для model_seven_emotions. Більш вичерпну діаграму можна знайти у Б.1. Спосіб задання як для більшості згорткових нейронних мереж є послідовним.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 236, 256)          25856
_________________________________________________________________
activation (Activation)      (None, 236, 256)          0
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 29, 256)           0
_________________________________________________________________
dropout (Dropout)            (None, 29, 256)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 29, 128)           163968
_________________________________________________________________
activation_1 (Activation)    (None, 29, 128)           0
_________________________________________________________________
dropout_1 (Dropout)          (None, 29, 128)           0
_________________________________________________________________
flatten (Flatten)            (None, 3712)              0
_________________________________________________________________
dense (Dense)                (None, 6)                 22278
_________________________________________________________________
activation_2 (Activation)    (None, 6)                 0
=================================================================
Total params: 212,102
Trainable params: 212,102
Non-trainable params: 0
```

Рисунок 5.4.1. — Архітектура model_positivity

Більшість шарів було уже описано у розділі 4.3. Варто звернути увагу на виключення (Dropout). Шари виключення відповідають за регуляризацію (коефіцієнти відповідно дорівнюють 0.2 та 0.1 відповідно). Попарне використання шарів згортки та виключення допомагає зменшити ризики перенавчання.

Використано напівлінійну (“relu”) та нормовану експоненційну функції (“softmax”) для активації нейронів. Оскільки кількість класів дорівнює семи, то за

допомого softmax отримаємо список ймовірностей (значення від 0 до 1) такої ж розмірності.

Індекс найбільшого елемента є прогнозованим значенням. Визначити назву емоції можна за допомогою підстановки: 0 = страх, 1 = огида, 2 = нейтральний, 3 = щастя, 4 = сум, 5 = подив, 6 = злість.

Архітектура нейромережі model_positivity має подібну структуру. Єдина відмінність полягає у кількості класів для класифікації, коефіцієнтах регуляризації.

Класи сформовано наступним чином: 0 = negative_female, 1 = positive_female, 2 = neutral_female, 3 = neutral_male, 4 = positive_male, 5 = negative_male. Таким чином ШНМ може прогнозувати не лише рівень задоволеності, але й стать. Архітектуру для model_positivity наведено у додатку Б.1.

5.5 Навчання та аналіз отриманих результатів

Важливим етапом для навчання нейронної мережі є розподіл даних на тренувальну та тестову вибірки. Для моделі model_positivity екземпляри попередньо згруповані у директорії відповідно до класів за допомогою окремого модуля. Використавши функцію train_test_split утворимо вибірки у співвідношені 4:1, тобто для тестування виділено 20% від загального датасету.

За функцію витрат взято перехресну ентропію (Categorical Cross Entropy Loss Function). У задачах машинного навчання перехресна ентропія є мірою помилки у задачах багатокласової (кількість класів більша двох) класифікації.

Для оптимізації навчання використано алгоритм Adam [11], що у свою чергу є покращеною версією стохастичного градієнтного спуску. Точність визначається за допомогою метрики accuracy, що описує ефективність моделі для рівноважливих класів. Простими словами, це є співвідношення правильних прогнозів до їх загальної кількості.

Також використано два підходи для покращення якості навчання: зміна темпу навчання (англ. “learning rate”) та рання зупинка (англ. “early stop”). Для першого необхідно використати метод ReduceLROnPlateau із фреймворку Keras.

Суть цього методу полягає у зменшенні темпу навчання (множник factor) для заданої метрики (acc, loss, val_acc, val_loss) за умови, що на протязі певної кількості епох модель не покращує результат. Схематично це можна описати наступною формулою:

$$new\ learning\ rate\ =\ learning\ rate\ *\ factor \quad (5.5.1)$$

Для model positivity та model_seven_emotions ReduceLROnPlateau використовує такі параметри:

- кількість епох, при яких val_accuracy не змінюється: 4;
- factor: 0.5;
- мінімальне значення темпу навчання: 10^{-5}.

Метою використання ранньої зупинки є попередження перенавчання моделі. Метод відслідковує зміни функції витрат та зупиняє навчання тоді, коли при заданій кількості епох результат практично не змінюється. Таким чином поєднання ранньої зупинки та зміни темпу покращує загальну динаміку при стагнації нейромережі.

Для моделей model_positivity та model_seven_emotions рання зупинка відбуватиметься за умови, що протягом 45 ітерація функція втрат (val_loss) практично не змінюється.

До прикладу, рання зупинка відбулася на 201 ітерації для model_seven_emotions та 36 для model_positivity при загальній кількості епох - 500.

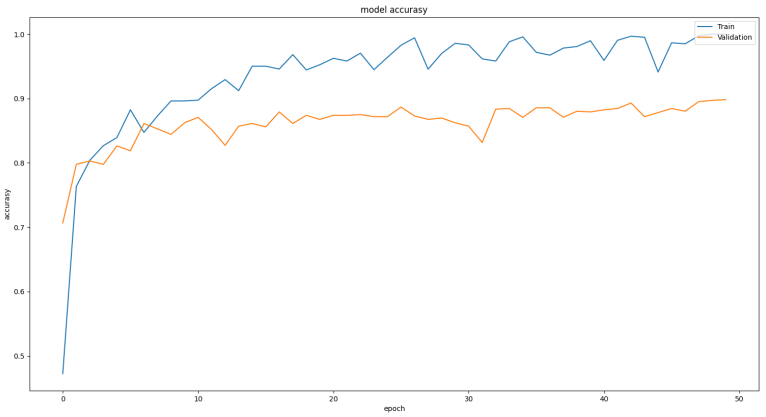


Рисунок 5.5.1. — Графік точності для model_positivity

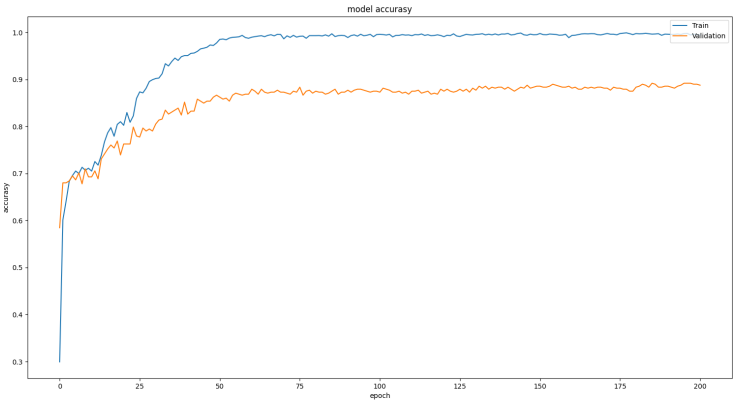


Рисунок 5.5.2. — Графік точності для model_seven_emotions

Із графіками функцій втрат для обох моделей можна ознайомитися у А.1 та А.2.

У результаті навчання було отримано наступні показники:

- model_positivity
  - тренувальна вибірка:
    - loss: 4.1290e-04
    - accuracy: 0.9989
  - тестова вибірка:
    - loss: 0.6335
    - accuracy: 0.8983
- model_seven_emotions:
  - тренувальна вибірка:

- loss: 0.0130
- accuracy: 0.9947

○ тестова вибірка:

- loss: 0.4682
- accuracy: 0.8877

Сформуємо звіти (англ. “classufucation reports”) про навчання та ефективність нейромереж для обох моделей за допомогою функції classification_report.

	precision	recall	f1-score
fear	0.83	0.83	0.83
disgust	0.85	0.89	0.87
neutral	0.82	0.91	0.86
happy	0.78	0.88	0.83
sadness	0.84	0.79	0.82
surprise	0.94	0.84	0.89
angry	0.92	0.81	0.86
accuracy			0.85
macro avg	0.85	0.85	0.85
weighted avg	0.85	0.85	0.85

	precision	recall	f1-score
negative_female	0.96	0.98	0.97
negative_male	0.73	0.74	0.74
neutral_female	0.99	0.94	0.97
neutral_male	0.80	0.77	0.79
positive_female	0.97	0.97	0.97
positive_male	0.59	0.58	0.58
accuracy			0.90
macro avg	0.84	0.83	0.83
weighted avg	0.90	0.90	0.90

Рисунок 5.5.3. — Classification report для model_seven_emotions та model_positivity

У даному прикладі наведені наступні метрики:

- влучність (англ. “precision”) - це прогностична значущість позитивного результату. Простими словами, це відношення правильних позитивних прогнозів до загальної кількості позитивних прогнозів;
- повнота (англ. “recall”) - це відношення правильних позитивних прогнозів до загальної кількості фактичних позитивних прогнозів;
- f-міра (англ. “f-score”) - це середнє зважене гармонійне значення точності та повноти. Формула має вигляд:

$$f\_score = 2 * (precision * recall) / (precision + recall) \quad (5.5.2)$$

5.6 Розробка веб-застосунку

За допомогою програмних компонент бібліотеки streamlit було створено мінімально життєздатний продукт із зручним інтерфейсом користувача (див. рисунки А.3 та А.4).

Функціонал веб-застосунку повинен містити:

- Можливість обрати ознаки розпізнавання (3 emotions, 7 emotions, gender), на основі яких відбуватиметься класифікація;
- Можливість завантажити будь-який аудіофайл;
- Можливість перевірити функціонал за допомогою тестового файлу;
- Графік завантаженого файлу у хвильовій формі;
- Графік для мел-кепстральних коефіцієнтів;
- Графік спектрограми;
- Вбудований аудіопрогравач;
- Результат класифікації із вказанням точності прогнозу;
- Діаграму для кожної обраної у меню ознаки;

Інтерфейс користувача має наступний вигляд:

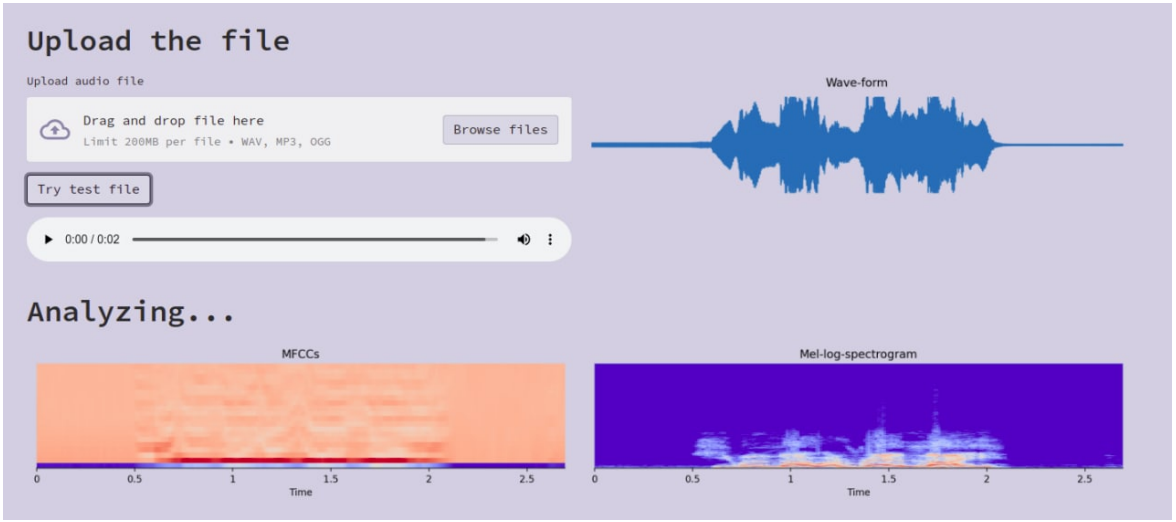


Рисунок 5.6.1. — Блок завантаження файлу

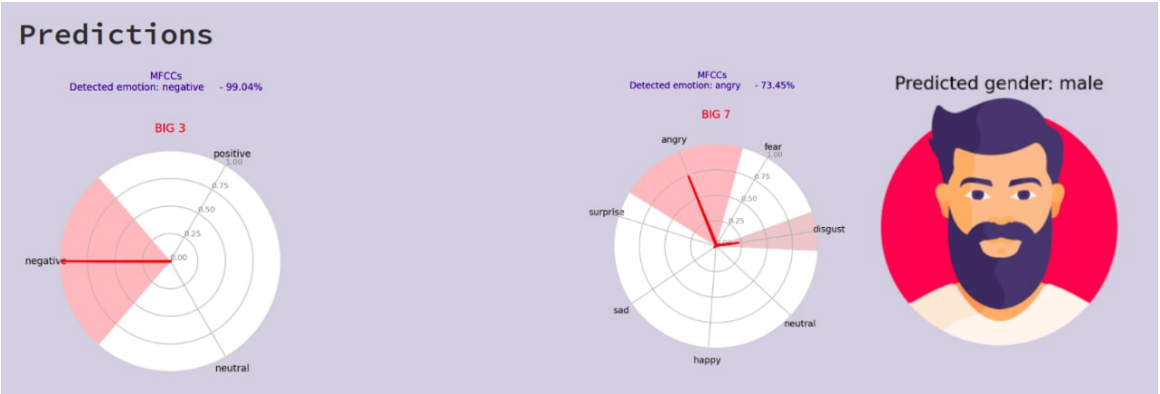


Рисунок 5.6.2. — Візуалізація отриманого результату

5.7 Висновки до розділу

У розділі “ПРОЕКТУВАННЯ СИСТЕМИ КЛАСИФІКАЦІЇ ЕМОЦІЙ” розглянуто теоретичні та практичні засади побудови системи класифікації аудіофайлів за емоційною ознакою. Дотримано усіх вимог, окреслених у постановці задачі, а саме:

- Розглянуто наявні у вільному доступі бази даних для розв’язку задачі;
- Описано основні поняття про емоції та їх види;
- Побудовано згорткові нейронні мережі, котрі дозволяють проводити класифікацію на основі:
    - шести основних емоцій;
    - статі мовця;
    - рівня задоволеності.
- Описано процес навчання та проаналізовано отримані результати;
- Створено веб-застосунок у якості мінімально життєздатного продукту.

Дослідження зроблені у цьому розділі мають цілком обгрунтоване практичне застосування. Перспективним напрямком використання можна вважати інтернет речей (англ. “internet of things”), а саме: смарт-будинки, отримання фідбеку та рівня задоволеності клієнтів у кол-центрах, у задачах нейролінгвістики.

Оскільки потенціал нейронних мереж у розпізнаванні мови ще не розкритий до кінця, то описані підходи та системи залишаються актуальними.

ВИСНОВКИ

У результаті виконання магістерської роботи застосовано згорткові нейронні мережі для розв’язування задач множинної класифікації аудіоінформації на основі декількох ознак - музичний жанр та емоція.

На практиці продемонстровано важливі аспекти проектування ШНМ, а саме: підвищення точності прогнозу на основі обробки бази даних, методи регуляризації та оптимізації, архітектурні особливості, навчальні парадигми, представлення та тлумачення результатів.

У ході розв’язування поставлених задач було продемонстровано, що предметна область застосування нейронних мереж є доволі широкою. Це зумовлено двома чинниками: ефективністю моделі в умовах невизначеності та співвідношенням ціна та кількість використаних ресурсів.

Як наслідок, застосування нейромереж, на противагу класичним методам розв’язування задач класифікації, є одним із найбільш перспективних, оскільки для отримання бажаного результату потрібно суттєво менше людського ресурсу і часу вцілому.

Поставлені задачі виконано у повному обсязі, а отримані знання та здобуті навики можуть бути використані в інших задачах множинної класифікації, а саме: опрацювання природної мови, системи відеоспостереження, конвеєрна стрічка на підприємстві, діагностика захворювань, отримання рівня задоволеності клієнтів у консалтингу, аудіотеки тощо.

Оскільки дослідження в області нейролінгвістики і машинного навчання вцілому тривають, то дипломна робота ще надовго залишатиметься актуальною.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Глибовець М.М. Штучний інтелект / М.М. Глибовець, О.В. Олецький. — К.: КМ Академія, 2002. - 282 с.
2. Комашинський В.І. Нейронні мережі і їх застосування у системах управління і зв'язку / В.І. Комашинський. - К.: КМ Академія, 2003. - 22 с.
3. Кутковецький В.Я. Розпізнавання образів: навчальний посібник / В.Я. Кутковецький. — Миколаїв: Вид-во МДГУ ім. П. Могили, 2003. — 196 с.
4. Пилипенко В. О. Варіант використання нейронної мережі в системі «Smart Home» / В. О. Пилипенко, І.І. Слюсар, В. І. Слюсар, В.М. Маруженко // Інтеграція інформаційних систем і інтелектуальних технологій в умовах трансформації інформаційного суспільства: Тези доп. Четвертої міжнародної наук. конф. (21-22 жовтня 2021р., м. Полтава). - Полтава, 2021. - 93-95 с.
5. Синєглазов В. Глибокі нейронні мережі для вирішення завдань розпізнавання і класифікації зображення [Електронний ресурс] / В. Синєглазов, О. Чумаченко. – 2017. 4 с. – Режим доступу до ресурсу: http://itcm.comp-sc.if.ua/2017/Sineglazov.pdf.
6. A. Labach. Survey of Dropout Methods for Deep Neural Networks / A. Labach, H. Salehinejad, S. Valaee. - 2019. - Available from: https://arxiv.org/abs/1904.13310.
7. B. Schuller, G. Rigoll and M. Lang, "Hidden Markov model-based speech emotion recognition," 2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698), 2003.
8. C. Bishop. Neural Networks for Pattern Recognition / Christopher Bishop., 1995 - p. 27.
9. C. Cortes. L2 Regularization for Learning Kernels / M. Mohri, A. Rostamizadeh. - 2012. - Available from: https://arxiv.org/abs/1205.2653.
10. C. E. Shannon. Communication in the presence of noise / C. E. Shannon. - Proc. Institute of Radio Engineers, 1949. - p. 21.

11. D. Kingma. Adam: A Method for Stochastic Optimization / D. Kingma, J. Lei Ba. - 2015.- Available from: https://arxiv.org/pdf/1412.6980v9.pdf.
12. E. Geoffrey Publications in Reverse Chronological Order.- Toronto: 2007. - Available from: http://www.cs.toronto.edu/~hinton/papers.html.
13. Ekman P. Basic emotions / P. Ekman // The handbook of cognition and emotion. New York.: John Wiley & Sons. - p. 45.
14. F. Rosenblatt. The Perceptron—a perceiving and recognizing automaton / F.Rosnblatt.- Cornell Aeronautical Laboratory, 1957.
15. GTZAN dataset - Music Genre Recognition. - Available from: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification.
16. I. Hrychaniuk. Analysis of data augmentation methods for retinal vessel segmentation problems / I. Hrychaniuk, O. Nosovets. - 2021. - Available from: https://www.molodyivchenyi.ua/index.php/journal/article/view/2369/2356.
17. J. McCarthy. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence / J. McCarthy, M. Minsky, N. Rochester, C. Shannon. - AI Magazine, 1955. - Available from: https://doi.org/10.1609/aimag.v27i4.1904.
18. Keras documentation. - Available from: https://keras.io/.
19. Khan S. A Guide to Convolutional Neural Networks for Computer Vision. / Khan S., Rahmani K., Shah S. - Deli: Morgan & Claypool, 2018. 207 p.
20. Linear Transform.- Available from: http://linear.ups.edu/html/section-LT.html.
21. Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules / D. Ho et al. 2019. Available from: https://arxiv.org/abs/1905.05393.
22. RAVDESS Emotional speech audio - Available from: https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio.
23. Regularization Techniques | Regularization In Deep Learning. Analytics Vidhya. Available from: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deeplearning-regularization-techniques/.

24. S. Steven. The Scientist and Engineer's Guide to Digital Signal Processing / S. Steven. - California Technical Pub, 1997. - pp. 177–180.
25. Surrey Audio-Visual Expressed Emotion (SAVEE) - Available from: https://www.kaggle.com/datasets/ejlok1/surrey-audiovisual-expressed-emotion-savee.
26. Tensorflow documentation. Available from: https://www.tensorflow.org/api_docs
27. Toronto emotional speech set (TESS). - Available from: https://www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess.
28. W. McCulloch. A logical calculus of the ideas immanent in nervous activity / W. McCulloch., W. Pitts. - Bull Math Biol., 1943.
29. X. Sun. Robust Retinal Vessel Segmentation from a Data Augmentation Perspective / X. Sun, H. Fang, D. Zhu. - 2007. p. 3. - Available from: https://arxiv.org/pdf/2007.15883.pdf.
30. Y. Miyakoshi and S. Kato, "Facial emotion detection considering partial occlusion of face using Bayesian network," 2011 IEEE Symposium on Computers & Informatics, 2011, pp. 96-101, doi: 10.1109/ISCI.2011.5958891. - p. 96-101.
31. Z. Wang. Data Augmentation is More Important Than Model Architectures for Retinal Vessel Segmentation / Z. Wang. - 2017. Available from: https://dl.acm.org/doi/10.1145/3348416.3348425.

ДОДАТОК А. ДІАГРАМИ

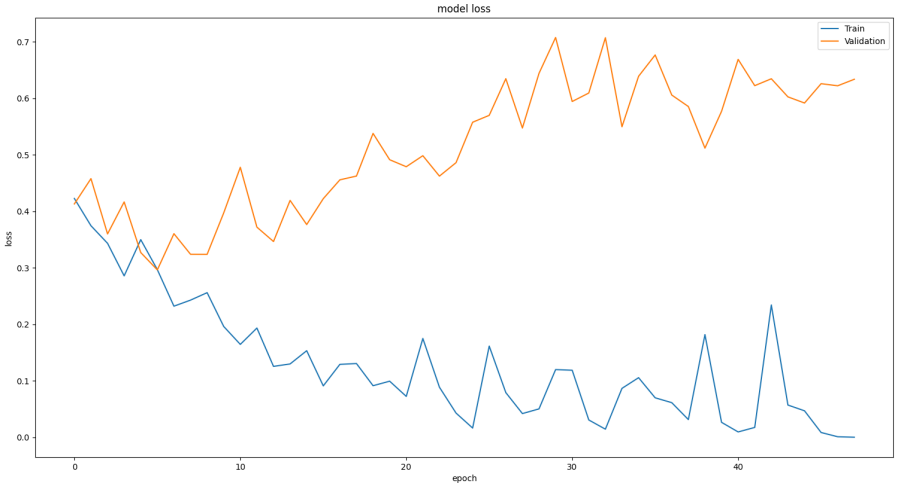


Рисунок А.1. — Графік втрат для model_positivity

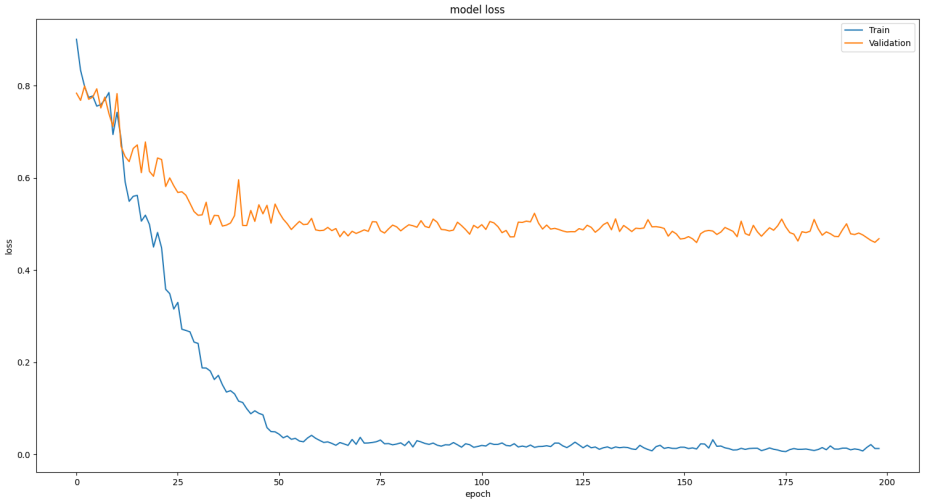


Рисунок А.2. — Графік втрат для model_seven_emotions

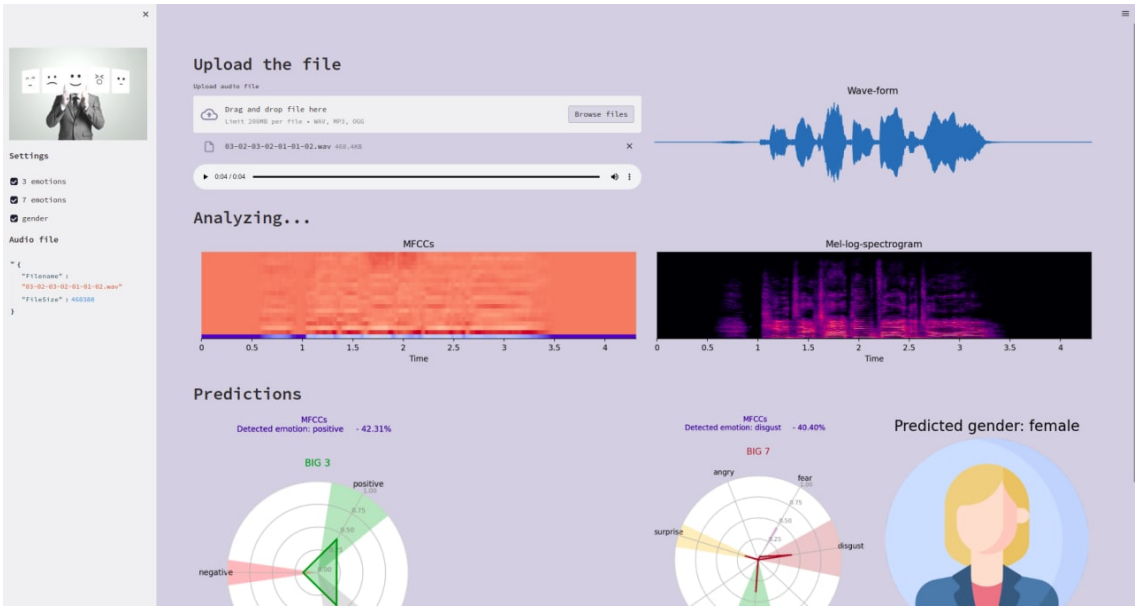


Рисунок А.3. — Інтерфейс користувача для емоцій

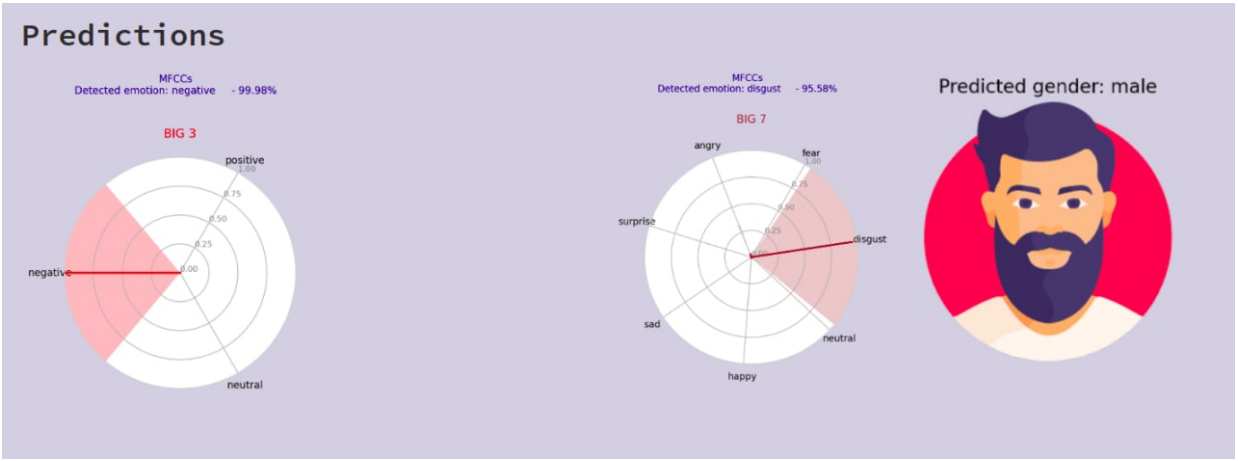


Рисунок А.4. — Діаграми для тестового файлу

ДОДАТОК Б. АРХІТЕКТУРИ НЕЙРОННИХ МЕРЕЖ

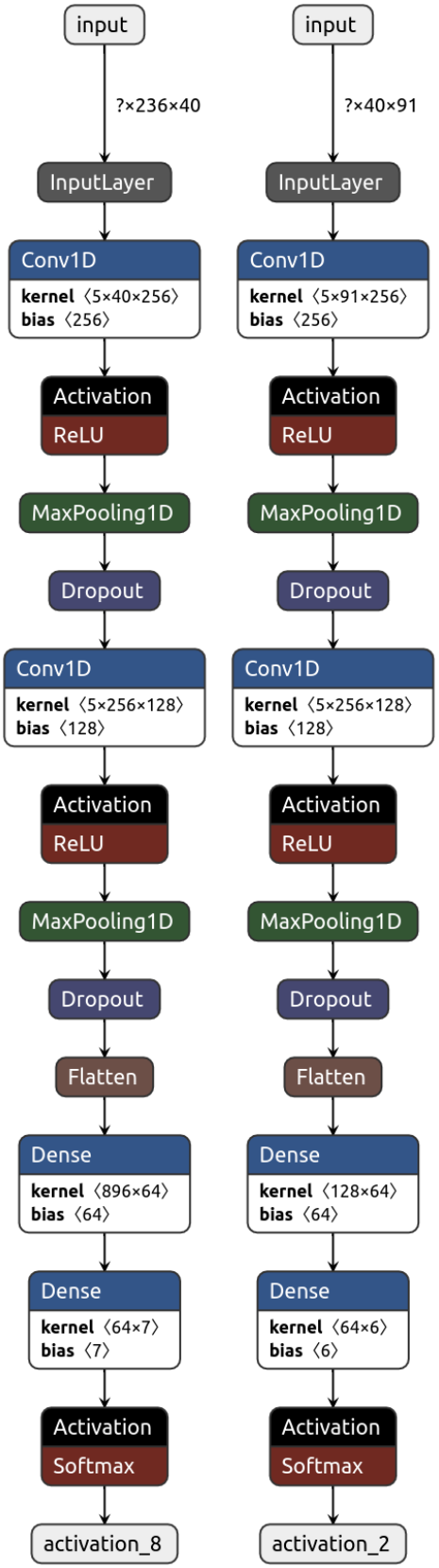


Рисунок Б.1. — Архітектури для model_seven_emotions, model_positivity