

**ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ,
ПРАВА ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ КАФЕДРА
ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

Пояснювальна записка

до кваліфікаційної роботи на здобуття ступеня вищої освіти магістр

на тему: «Технології класифікації аудіо контенту на основі нейронної мережі»

Виконав: здобувач вищої освіти за
освітньо-професійною програмою
Інформаційні управляючі системи та технології
спеціальності 126 Інформаційні системи та
технології
освітнього ступеня магістр
групи 126ІСТ_мд_21
Красюк А.О.
Керівник: Протас Н.М.
Рецензент: Біловод О.І.

Полтава – 2023 року

ВСТУП

Актуальність теми. У сучасному інформаційному суспільстві велика увага приділяється розробці та вдосконаленню технологій обробки та аналізу аудіо контенту. З поширенням стрімінгових платформ, аудіо книг, подкастів та інших форм розважального аудіо контенту, виникає потреба в автоматизованих методах класифікації та розпізнавання аудіо файлів.

Одним із перспективних напрямків у цьому контексті є використання нейронних мереж для класифікації аудіо контенту. Нейронні мережі, зокрема глибокі нейронні мережі, здатні автоматично виділяти складні патерни в даних, що робить їх ефективними інструментами для аналізу аудіо інформації. Комбінування цих технологій дозволяє розробляти системи, які спроможні класифікувати аудіо контент за різноманітними критеріями, такими як жанр, настрій, мова тощо.

Метою кваліфікаційної роботи є дослідження та реалізація технологій розпізнавання мовлення на основі нейронних мереж. Дослідження буде спрямоване на визначення ефективних архітектур нейронних мереж для завдань розпізнавання мовлення, а також на вивчення впливу різних параметрів та підходів до побудови моделей. Результати дослідження дозволять розробити та оптимізувати систему класифікації аудіо, яка може знайти застосування в різних сферах, включаючи медіа, рекламу, та інші галузі, пов'язані із звуковим контентом.

Завданнями кваліфікаційної роботи є:

- аналіз розвитку систем та моделей нейронних мереж,
- визначення особливостей апаратних та програмних систем,
- створення нейронної мережі призначеної для розпізнавання мовлення.

Об'єктом дослідження є архітектури нейронних мереж що призначені для класифікації аудіо даних.

Предметом дослідження є ІС, робота якої базується на нейронній мережі що призначена для аналізу мовлення.

Методи дослідження – проведені в роботі дослідження базуються на методах системного аналізу, нейронних мереж, які були ключовими в створенні ІС для розпізнавання мовлення.

Інформаційна база кваліфікаційної роботи складається з наукових статей, міжнародних аналітичних видань і звітів, матеріалів наукових конференцій інтернет-ресурсів, що містять інформацію про архітектуру сучасних нейронних мереж, а також даних, отриманих від провідних ІТ-компаній у сфері створення ІС.

Практична значущість роботи полягає в можливості повторного застосування та модифікації розробленої ІС, та подальшої модифікації під певні потреби.

Апробація результатів дослідження відбувалася шляхом оприлюднення доповідей на наукових конференціях, семінарах.

Публікації. За результатами проведеного дослідження опубліковано тези: 1. «Технології класифікації аудіо контенту на основі нейронної мережі». Матеріали науково-практичної конференції за підсумками проходження виробничих практик здобувачів вищої освіти спеціальності 126 Інформаційні системи та технології, кафедра інформаційних систем та технологій Полтавського державного аграрного університету, 17 вересня 2023 р. Вип. VII (частина I). Полтава: ПДАУ.

Структура та обсяг кваліфікаційної роботи логічно пов'язані з задачами досліджень. Робота містить перелік умовних позначень, вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг текстової частини дипломної роботи складає 63 сторінок формату А4. Вона містить 30 рисунків та 1 таблиця. В роботі використано 39 науково-технічних джерела.

РОЗДІЛ 1

АКТУАЛЬНІСТЬ ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ КЛАСИФІКАЦІЇ АУДІО КОНТЕНТУ

1.1 Використання нейронних мереж для класифікації аудіо контенту

Актуальність задач класифікації аудіо контенту за допомогою нейронних мереж в сучасному світі обумовлена стрімким розвитком цифрових технологій та зростанням об'ємів аудіо контенту в мережі. З великим обсягом аудіо даних, таких як музика, звукові файли у відео-контенті, подкасти, виникає необхідність в автоматизованих методах їх класифікації для ефективного пошуку, рекомендацій, контролю вмісту та забезпечення кращої користувацької взаємодії з мультимедійними платформами. Крім того, такі системи можуть мати важливе значення в сферах безпеки, відстеження аудіо вимог індустрії, а також в розробці інноваційних сервісів у медіа та розвагах.

Розвиток технологій нейронних мереж надає неймовірні можливості для розробки систем класифікації аудіо контенту. Використання цих методів виявляється особливо важливим в контексті росту обсягу аудіозаписів, що потребують автоматизованого аналізу. Класифікація аудіо контенту дозволяє визначати жанри музики, розпізнавати мови у мовних аудіозаписах, а також визначати контекст і емоційний тон аудіо [1].

Актуальність використання нейронних мереж для класифікації аудіо контенту можна розділити за наступними пунктами:

1. Зростання обсягу аудіо даних в мережі інтернет. Нейронні мережі використовуються для визначення та аналізу зростання обсягу даних в мережі інтернет, що спричинено великим зростом популярності використання стрімінгових платформ, музичних сервісів та інших джерел.

2. Роль використання нейронних мереж в сучасних мультимедійних платформах. Використовуються для аналізу впливу класифікації контенту на якість досвіду користувача на відео та музичних стрімінгових платформах, що в

подальшому використовується для створення персоналізованих рекомендацій та поліпшення функцій пошуку контенту через ефективну систему класифікації.

3. Безпека та контроль вмісту. Даний метод використання нейронних мереж для класифікації аудіо контенту використовується для виявлення небажаного або небезпечного вмісту в аудіо даних та впровадження систем контролю вмісту з урахуванням різноманітності культурних та соціальних контекстів.

4. Інновації у відстеженні трендів та аналізі аудіо вимог. Використання можливостей аналіз даних для відстеження змін в попиті та прогнозування трендів аудіо контенту, а також використання мереж для створення контенту відповідно потреб користувачів.

5. Потенціал для розвитку нових сервісів та інновацій. Методологія яка за допомогою інтеграції системи розпізнавання аудіо даних може служити для покращення наявних сервісів в яких використовуються аудіо данні та створення інноваційних систем які будуть базуватися на нейронних мережах [3].

Необхідність правильної підготовки даних перед тренуванням нейронної мережі є важливим етапом. Конвертація аудіо даних у числовий формат, такий як спектрограма, дозволяє нейронним мережам ефективно опрацьовувати надані дані. Розподіл даних на тренувальний та тестовий набори, разом із налаштуванням параметрів моделі та оптимізацією, сприяє покращенню загальної точності та взаємодії моделі з реальним аудіо контентом (рис. 1.1).



Рисунок 1.1 – Схема етапів розпізнавання мовлення

В сучасному світі, з великим обсягом аудіо інформації, розробка та вдосконалення систем класифікації аудіо контенту потребують точності на кожному з етапів, а саме:

1. Збір та підготовка даних. На цьому етапі обирається направлення в якому буде розроблятися нейронна мережа, та вибір типу аудіо даних які будуть класифікуватися. Напрямки в яких може тренуватися нейронна мережа та для класифікації якого типу даних вона призначена наступні:

- мови (українська, англійська, іспанська);
- музичні жанри (рок, поп, хіп-хоп, джаз);
- звуки природи (дощ, звуки океану, пісня птахів);
- типи молення (розповідь, опис, роздум);
- акустичні (транспортні засоби, сміх, звук дзвінка);
- звукові події в середовищі (кухонні звуки, звуки друку);
- емоційні стани (радість, смуток, агресія).

Найважливішим на даному етапі є визначення конкретного завдання класифікації.

2. Попередня обробка даних. Піддавання аудіо даних попередній обробці, що необхідно для подальшого перетворення сигналів в спектрограму, та інші види обробки даних, які будуть використанні як вхідні дані для нейронної мережі.

3. Вибір архітектури моделі. В залежності від специфіки поставленої задачі використовують:

- згорткові нейронні мережі (розроблені для того щоб використовувати дані з мінімальним обсягом попередньої обробки);
- рекурентні нейронні мережі (можуть використовувати внутрішню пам'ять для обробки послідовностей).

4. Навчання нейронної мережі. Використання зібраних даних які вже були попередньо оброблені. Під час цього процесу відбувається навчання яке в подальшому сприяє розділенню аудіо контенту на різні класи. Надалі від результатів навчання залежать результати роботи нейронної мережі.

5. Налаштування. Модель перевіряється за допомогою валідаційного набору даних, та в результаті перевірок відбувається пере-налаштування або ж зміна деяких параметрів для покращення її продуктивності.

6. Тестування та оцінка продуктивності. Модель навчається на підготовленому тестовому наборі даних, результатом якого є оцінка продуктивності за допомогою параметрів, таких як, точність, чутливість, специфічність, тощо.

7. Впровадження. Після всіх тестів та змін в налаштуваннях моделі, вона може бути впроваджена в реальні умови, та залежно від вимог може бути повторне навчання та оптимізація.

Для досягнення високої точності класифікації, використання різних архітектур нейронних мереж є критичним. Рекурентні нейронні мережі (RNN) і конволюційні нейронні мережі (CNN) дозволяють ефективно враховувати часові та просторові залежності у аудіо даних.

Крім того, комбінація цих архітектур, наприклад, CRNN (комбінація RNN і CNN), може покращити загальну ефективність моделі.

1.2 Огляд існуючих методів розпізнавання мовлення на базі нейронної мережі

Розпізнавання мовлення за допомогою нейронних мереж - це актуальний напрямок досліджень в галузі штучного інтелекту, який знаходить масштабне застосування в різних сферах життя.

За допомогою глибокого навчання з'явилась можливість будувати потужні та ефективні моделі, спроможні відтворювати складні зв'язки в аудіо сигналах та конвертувати їх у змістовний текст. Це має значущі практичні застосування у сучасних технологіях [2].

Однією з ключових областей використання розпізнавання мовлення є системи голосового управління. Вони дозволяють користувачам взаємодіяти з

пристроями та програмами, використовуючи лише свій голос. Наприклад, голосові асистенти, такі як Siri від Apple, Google Assistant та Amazon Alexa, стали невід'ємною частиною повсякденного життя людини, надаючи можливість швидкого доступу до інформації та виконання різних завдань.

Ще однією важливою областю є автоматизована транскрибація аудіозаписів. У бізнесі, освіті та наукових дослідженнях велику увагу приділяють точному текстовому представленню аудіо даних.

Системи розпізнавання мовлення дозволяють ефективно трансформувати голосовий контент у письмовий вигляд, що спрощує обробку та аналіз отриманої інформації.

Крім того, розпізнавання мовлення грає важливу роль у розвитку інтерфейсів людина-комп'ютер та технології розумного дому. Можливість взаємодіяти з пристроями та системами за допомогою голосових команд робить використання технологій більш доступним та зручним для користувачів.

На основі цих прикладів, що вказують на великий потенціал розпізнавання мовлення в різних сферах життя, можна стверджувати що за допомогою поєднання передових методів глибокого навчання та постійного розвитку обчислювальної потужності, ця технологія продовжує розширювати свої можливості та стає важливим інструментом в сучасному цифровому світі. Враховуючи тенденції розвитку, можна очікувати, що розпізнавання мовлення залишиться в центрі інновацій та забезпечить нові можливості для спілкування та взаємодії людей з технологією в майбутньому [5].

Тож у сучасному світі де такий стрімкий прогрес технологій розпізнавання мовлення на основі нейронних мереж є необхідним напрямком досліджень та здійснюється наступними підходами:

1. Розпізнавання фонем. В даному підході мовлення розглядається як послідовність найменших звукових одиниць мовлення – фонем. За допомогою цього методу модель навчається розпізнавати фонем з використанням нейронних мереж RNN або LSTM. Використання цього підходу часто використовується для низько рівневого аналізу аудіо сигналів (рис. 1.2).

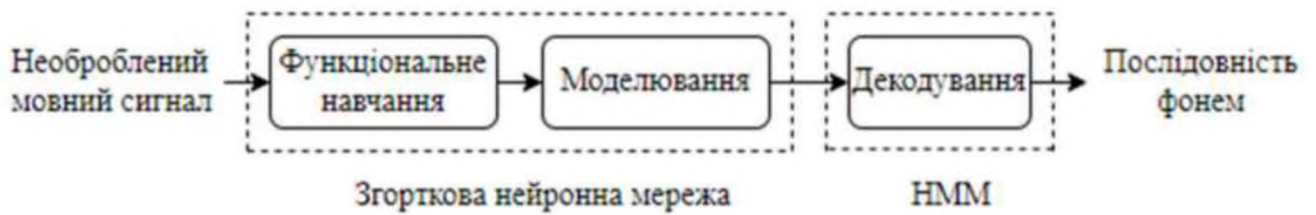


Рисунок 1.2 – Схематичне зображення розпізнавання фонем

2. Розпізнавання слів. Цим підходом модель на основі нейронної мережі навчається розпізнавати слова або короткі фрази, з використанням таких архітектур як, DNN, CNN або комбінації різних типів мереж (рис. 1.3).

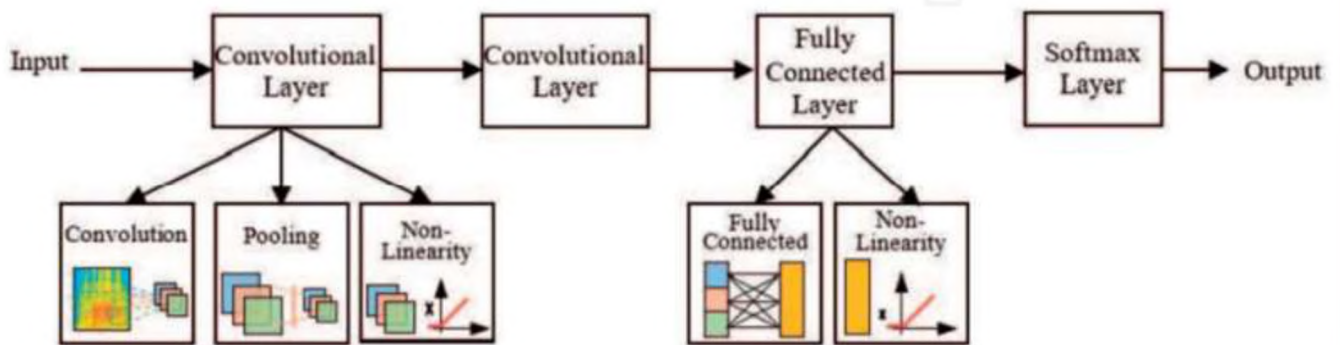


Рисунок 1.3 – Схематичне зображення розпізнавання слів

3. Розпізнавання фраз. В основу цього підходу закладено ціль в розпізнаванні фраз або повних речень, замість слів чи фонем. Виконується навчання та класифікація на основі більш складних архітектур, таких як RNN або трансформери, для врахування контексту.

4. Розпізнавання мови в реальному часі. Системи що створені та спрямованні на розпізнавання мовлення в режимі реального часу, де важливу роль відіграє швидкість обробки аудіо даних, та ґрунтуються на використанні оптимізованих архітектур нейронних мереж.

5. Мультиmodalне розпізнавання мовлення. В цьому підході об'єднується аудіо дані з візуальними, що дає змогу їх аналізу та використанню для кращого розуміння контексту отриманих даних.

6. Передача навчання (Transfer Learning). Використання попередньо навчених моделей для подальшого навчання на обмеженій кількості даних може стати ефективним підходом в разі недостатнього обсягу даних для повного мовного тренування моделі [4].

7. Синтез мовлення. Виконується підхід на основі архітектур LSTM, WaveNet, або Tacotron, для створення реалістичного звучання мовлення.

1.3 Використання нейронних мереж в сучасній освіті

У сучасному світі, де розвиток технологій стрімко зростає, використання штучного інтелекту та нейронних мереж набуває великої актуальності в різноманітних сферах життя, у тому числі в освіті.

Сучасний етап розвитку технологій суттєво впливає на освіту, зокрема використання нейронних мереж стає ключовим фактором для покращення методів навчання та аналізу звукових матеріалів та надає нові можливості, які відкриваються завдяки застосуванню нейронних мереж в цій сфері.

Інтеграція інноваційних технологій у навчальний процес є ключовим аспектом сучасної освіти, основним завдання якої є створення ефективних, персоналізованих методів навчання, що відповідають потребам кожного учня. Однією із перспективних областей впровадження нейронних мереж в освіті є їхнє використання для класифікації аудіо контенту.

Аудіо аналітика в освіті відкриває широкі можливості для розвитку інноваційних підходів, зокрема у вдосконаленні процесів вивчення мов, розробці індивідуальних навчальних програм, та сприянню полегшення взаємодії студентів із різними освітніми потребами.

Можливості використання нейронних мереж що спеціалізуються на обробці аудіо даних в наступні:

1. Розпізнавання мовлення. В освіті даний метод використання нейронних мереж використовується для надання допомоги в поліпшенні навичок вимови та

інтонації, а саме за допомогою проведення аналізу отриманих аудіо даних, в результаті чого надаються індивідуальні рекомендації стосовно навиків вимови та інтонації. Також можлива адаптація під графічні зображення (рис. 1.4).

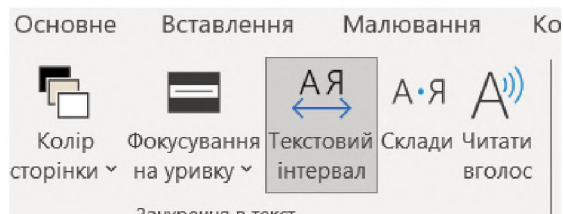


Рисунок 1.4 – Функція читання в Word

2. Аналіз емоцій. В ході навчального процесу можуть бути зібрані дані для аналізу емоцій студентів за їхнім голосом, які в подальшому будуть використані для створення більш сприятливого для навчання середовища, та виявлення особливих потреб студентів.

3. Стимулювання самостійної роботи. Створення мереж що будуть інтегровані з графічними зображеннями, в вигляді різноманітних вправ, в результаті проходження яких надаватимуться індивідуальні відгуки, що можуть сприяти ефективному тренуванню вимови та стимулюватимуть самостійну роботу над матеріалом [6].

4. Електронні підручники з аудіо супроводом. Моделі нейронних мереж можуть аналізувати аудіо матеріали в електронних підручниках, надаючи студентам додаткові звукові пояснення до навчального матеріалу (рис. 1.5).

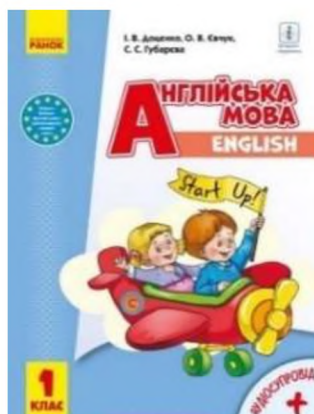


Рисунок 1.5 – Підручник з аудіо супроводом

5. Сприяння умов освіти для людей з особливими потребами. Використання нейронних мереж може використовуватись для створення індивідуальних спеціалізованих завдань що будуть базуватися на потребах студента, та сприятиме успіху студентам з різними особливими потребами [7].

Висновки до розділу 1

Узагальнюючи, класифікація аудіо контенту полегшує організацію, пошук та споживання контенту, а також може служити основою для різноманітних аналітичних інструментів і додатків. Вона є важливою для розвитку нових технологій та інновацій в аудіо індустрії. Наприклад, вона може використовуватися для створення ефективних алгоритмів обробки звуку, покращення акустичних систем чи розробки нових аудіо технологій, може бути використана в освітніх та дослідницьких програмах для аналізу тенденцій у музиці, звуковому дизайні, аудіо архітектурі та інших сферах. Це сприяє подальшому розвитку сучасного звукового мистецтва та технологій та відіграє ключову роль у багатьох аспектах сучасного цифрового життя, починаючи від покращення користувацького досвіду, використанням для інновацій у сфері звуку та аудіо технологій а також в освіті.

А також можна стверджувати що за допомогою нейронних мереж досягається індивідуалізація навчання. Моделі можуть адаптуватися до потреб кожного учня, створюючи персоналізовані програми, що враховують стилі та темпи навчання. Це стимулює активніше засвоєння матеріалу та підвищує загальну ефективність навчання, автоматизована оцінка та фідбек забезпечують швидку інформацію про успішність учнів. Використання нейронних мереж для аналізу спрощує процес оцінювання, а також дозволяє швидко отримувати конструктивну відповідь. Покращення методів навчання та аналіз звукових матеріалів стає ключовим елементом ефективного, індивідуалізованого та захопливого навчання.

РОЗДІЛ 2

АРХІТЕКТУРИ ТА МОДЕЛІ НЕЙРОННИХ МЕРЕЖ

2.1 Архітектура штучних нейронних мереж

Штучні нейронні мережі (ANN) представляють собою обчислювальні системи, інспіровані біологічними нейронними мережами, які утворюють мозок тварин (рис. 2.1). Ці системи здатні навчатися вирішувати завдання, покращуючи свою продуктивність на основі прикладів, і це робиться без потреби спеціального програмування для конкретної задачі [8]. Наприклад, у випадку розпізнавання зображень ANN можуть навчатися ідентифікувати зображення, що містять котів, аналізуючи приклади зображень, які були позначені як "собака" або "не собака".

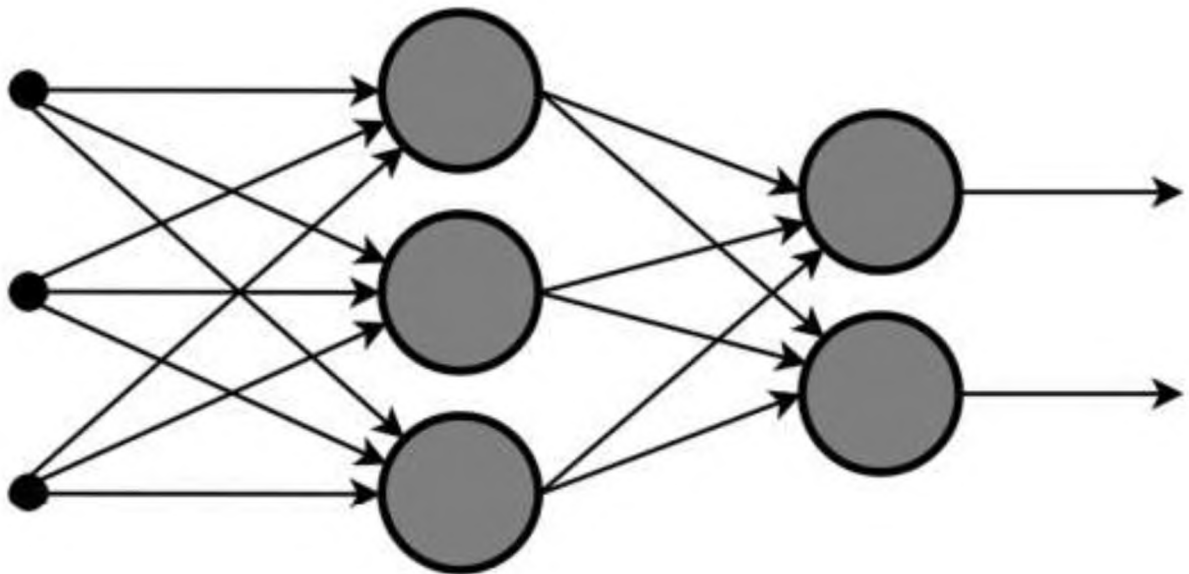


Рисунок 2.1 – Схема штучних нейронних мереж

Ці системи розвивають свій власний набір характеристик під час обробки навчального матеріалу, не володіючи апіорним знанням про конкретні ознаки котів, такі як хутро, хвіст, вуса, тощо. Архітектурно ANN базується на з'єднаних вузлах, які називають штучними нейронами, а ці вузли є аналогами біологічних нейронів у головному мозку тварин. Кожне з'єднання між штучними нейронами,

подібно до синапсів у біологічних нейронах, може передавати сигнали від одного нейрона до іншого.

Штучні нейронні мережі можуть мати різні архітектурні конфігурації, що визначаються кількістю шарів, нейронів у кожному шарі, та типами з'єднань між нейронами [9] (рис. 2.2). Два основних типи архітектур це, перцептрони та згорткові нейронні мережі, а також можливе використання:

- довільні архітектури глибокого навчання;
- архітектури Transformer;
- глибокі згорткові рекурентні нейронних мереж.

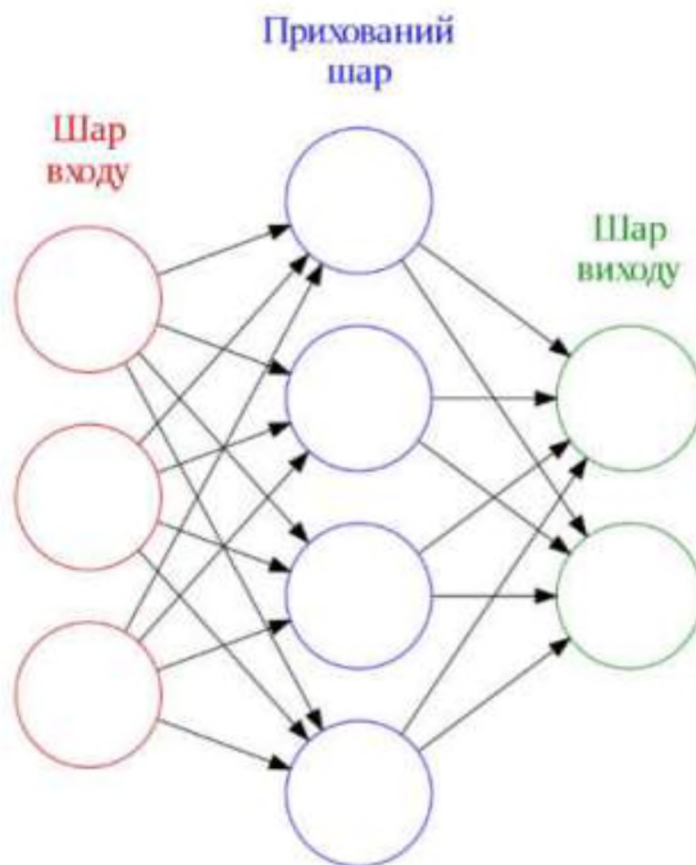


Рисунок 2.2 – Одна з архітектурних конфігурацій штучної нейронної мережі

Ці різноманітні архітектури надають можливість вибору моделі в залежності від конкретного завдання класифікації аудіо даних та властивостей вхідних сигналів.

2.1.1 Використання персептронів для класифікації аудіо даних

У контексті сучасного розвитку нейронних мереж та обробки аудіо даних виникає питання про використання персептронів для класифікації аудіо сигналів. Персептрони, як базові блоки в галузі нейронних мереж, можуть бути потенційно використані для цієї задачі, але важливо враховувати особливості аудіо даних [10]. Аудіо сигнали зазвичай є послідовностями амплітуд, що ускладнює їхню обробку за допомогою стандартних персептронів, призначених для роботи з векторними входами. Однак існують підходи, які можуть допомогти вирішити цю проблему (рис. 2.3).

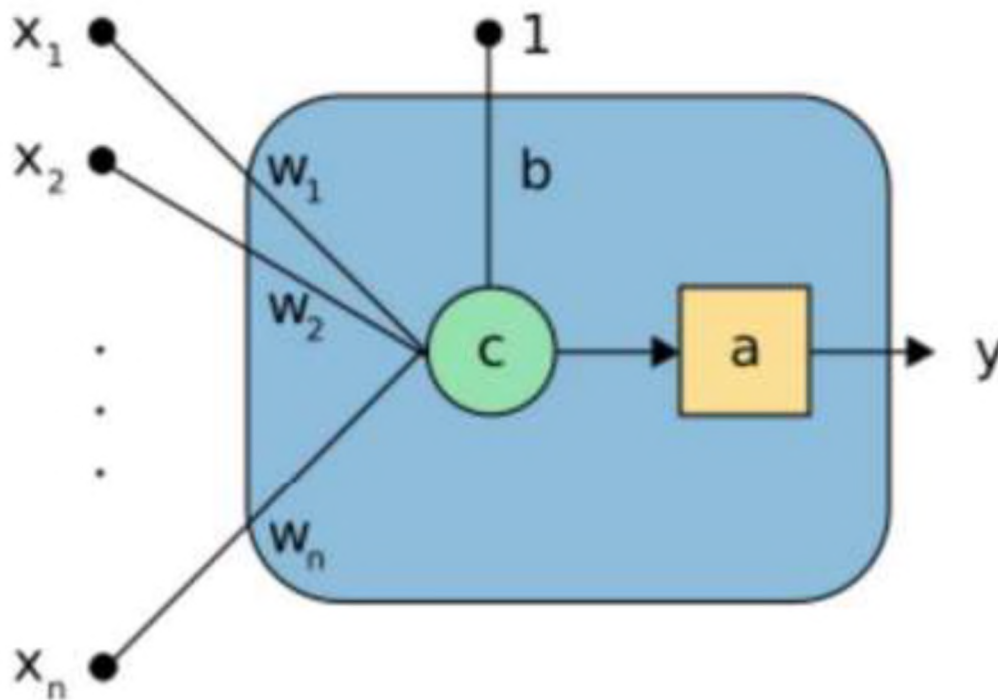


Рисунок 2.3 – Графічне представлення Персептрона

Перший підхід полягає в використанні акустичних ознак аудіо сигналів, таких як спектральні характеристики чи частотні коефіцієнти амплітуд. Ці ознаки можуть бути отримані з аудіо сигналу та подані у вигляді вектора для подальшої обробки персептронами.

Другий підхід включає поділ аудіо сигналу на невеликі часові фрейми для локалізації характеристик в часі. Кожен фрейм може бути оброблений окремо, а

отримані результати можуть бути комбіновані для отримання кінцевого класифікаційного результату.

До третього підходу входять рекурентні перцептрони, здатні моделювати динаміку в часі аудіо сигналу. Вони можуть враховувати послідовність аудіо фреймів та встановлювати зв'язок між ними, що робить їх ефективними для обробки послідовностей. Крім того, використання глибоких перцептронів варто розглядати для автоматичного вивчення вкладених представлень аудіо сигналів на різних рівнях абстракції. Важливо зауважити, що традиційні перцептрони можуть бути обмежені в розпізнаванні динамічних залежностей в аудіо сигналах. Тому для більш складних завдань, особливо в обробці послідовностей, бажано розглядати інші архітектури, такі як рекурентні нейронні мережі, які здатні ефективно працювати з послідовностями.

Загалом, використання перцептронів для класифікації аудіо даних може бути успішним, з врахуванням адекватної обробки аудіо сигналів та розуміння особливостей їхнього характеру.

2.1.2 Згорткові нейронні мережі

Згорткові нейронні мережі представляють собою інструмент для класифікації аудіо даних, де ключову роль у розпізнаванні аудіо сигналів відіграють просторові та часові залежності (рис. 2.4). Основною перевагою цих штучних нейронних мереж є їхня здатність автоматизовано вивчати корисні характеристики отриманих сигналів без необхідності їхнього ручного вибору або проектування якихось ознак [11].

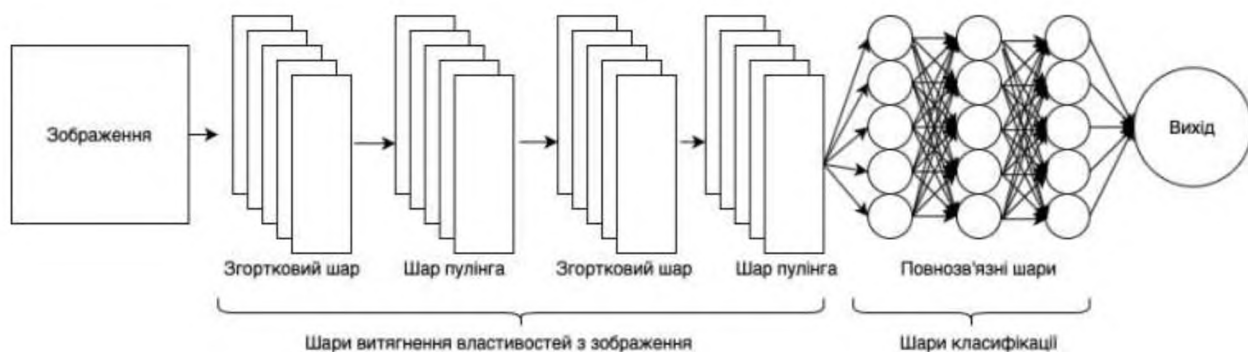


Рисунок 2.4 – Типова архітектура згорткової нейронної мережі

Архітектура згорткових нейронних мереж включає в себе кілька ключових етапів:

1. Вхідні аудіо дані. Це сигнал що перетворюється у спектрограму, яка візуально відображає час та частоту [12].

2. Згорткові шари. Використовуються для виявлення локальних патернів у спектрограмі, здатних виділити важливі аспекти аудіо сигналу.

3. Пулінгові шари. Допомагають відзначити ключові особливості та зменшити обсяг обчислень.

4. Завершує процесу. Повністю з'єднаний шар, що класифікує оброблений вектор ознак на конкретні класи аудіо сигналів [13].

Використання функції втрат та оптимізації дозволяє оцінити загальну різницю між прогнозованими та фактичними класами аудіо даних, а також оптимізувати параметри мережі для досягнення мінімальних втрат.

Застосування згорткових нейронних мереж у класифікації аудіо даних є доволі широким у використанні, надаючи змогу охоплювати такі завдання як:

- розпізнавання мови;
- класифікація жанрів музики;
- визначення аудіо файлів з різними характеристиками тощо.

Вони виявляються ефективним інструментом для розв'язання різноманітних завдань у сфері обробки аудіо даних, сприяючи подальшому розвитку та удосконаленню технологій машинного сприйняття та аналізу аудіо інформації.

2.1.3 Рекурентні нейронні мережі

Рекурентні нейронні мережі (RNNs) являють собою інструмент для класифікації аудіо даних, де важлива роль відводиться аналізу динаміки та послідовностей в часі (рис. 2.5).

Здатність RNNs моделювати довгострокові залежності робить їх ефективними в завданнях, де важлива можливість зберігання інформації про попередні часові кроки та використання цієї інформації при обробці нових даних. Це робить RNNs ефективними для вирішення завдань, пов'язаних із залежністю в часі та динамікою даних.

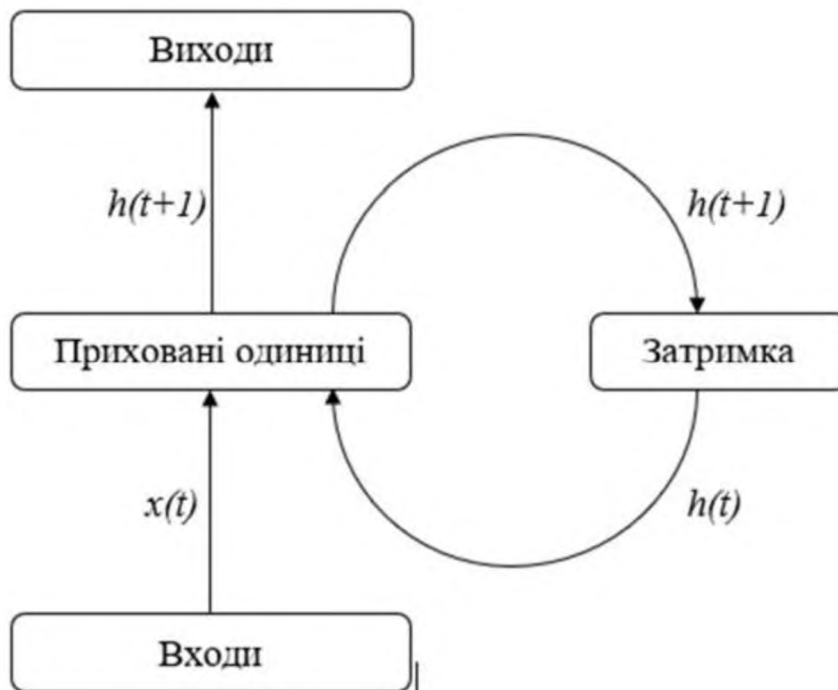


Рисунок 2.5 – Типова архітектура рекурентної нейронної мережі

При використанні RNNs для класифікації аудіо даних першим кроком є підготовка вхідних даних. Аудіо сигнали можуть бути конвертовані у послідовності акустичних ознак, таких як спектральні коефіцієнти або MFCC, які представляють собою числові вектори, що характеризують властивості звуку у різних часових і частотних областях [12].

Подальший аналіз аудіосигналів виконується за допомогою архітектури RNN. Це може включати в себе різні типи RNNs, такі як Long Short-Term Memory (LSTM) або Gated Recurrent Unit (GRU), які розроблені для подолання проблем зниклих градієнтів та здатні ефективно працювати з довгостроковими залежностями в даних.

Останній шар RNNs визначає класифікацію аудіо сигналу за допомогою функції активації, такої як softmax. Функція втрати, така як категоріальна кросс-ентропія, використовується для оцінки різниці між прогнозованими та фактичними класами

Важливою перевагою використання RNNs є їхні можливості адаптації до різних типів аудіо даних, включаючи розпізнавання мови, класифікацію музичних

жанрів, аналіз аудіо файлів та інші завдання [14]. Вони дозволяють ефективно моделювати динаміку різних звукових характеристик та забезпечують глибокий аналіз послідовностей даних. Таким чином, використання рекурентних нейронних мереж для класифікації аудіо даних є перспективним напрямком досліджень, що дозволяє підвищити точність та ефективність аналізу аудіо інформації в різноманітних застосуваннях [15].

2.1.4 Архітектури Transformer

Трансформери, спочатку призначені для обробки послідовностей у машинному перекладі, отримали широке застосування в різних областях обробки природної мови, включаючи аудіо аналітику. Архітектури Transformer для аудіо включають в себе інноваційні підходи до обробки акустичних сигналів та взаємодії між різними частинами моделі. Однією з ключових архітектур для аудіо є SpecAugment. SpecAugment використовує трансформації в часовому та частотному просторі для допомоги у тренуванні моделей на маленьких наборах даних та поліпшенні їхньої загальної здатності до узагальнення [19].

Для аудіо аналізу використовують архітектури, які включають у себе Transformer-подібні блоки, такі як Time-domain or Frequency-domain Transformer (T-F Transformer) або Speech-Transformer. Вони адаптовані для ефективною обробки акустичних даних, зокрема для розпізнавання мови або генерації музики. Однією з ключових особливостей Transformer-подібних архітектур для аудіо є їхні здатності моделювати довгострокові залежності в аудіо сигналах, що може бути важливо в завданнях, таких як розпізнавання мови. Вони використовують абстрактні представлення, названі увагою, для виділення важливих частин аудіо сигналу та розуміння контексту [20].

Важливим напрямком є також робота з амбітнішими завданнями, такими як аудіо генерація або аудіо супровід для відео, використовуючи архітектури Transformer. Вони виявляються потужним інструментом для синтезу аудіо даних та вивчення складних залежностей між звуками. Взагалі, Transformer-подібні архітектури для аудіо виявляються ефективними в багатьох завданнях, де важлива обробка послідовностей або аудіо сигналів (рис. 2.6). Їхній успіх у великих

проектах підтверджує їхню здатність до узагальнення та вирішення різноманітних задач в області обробки аудіо даних [21].

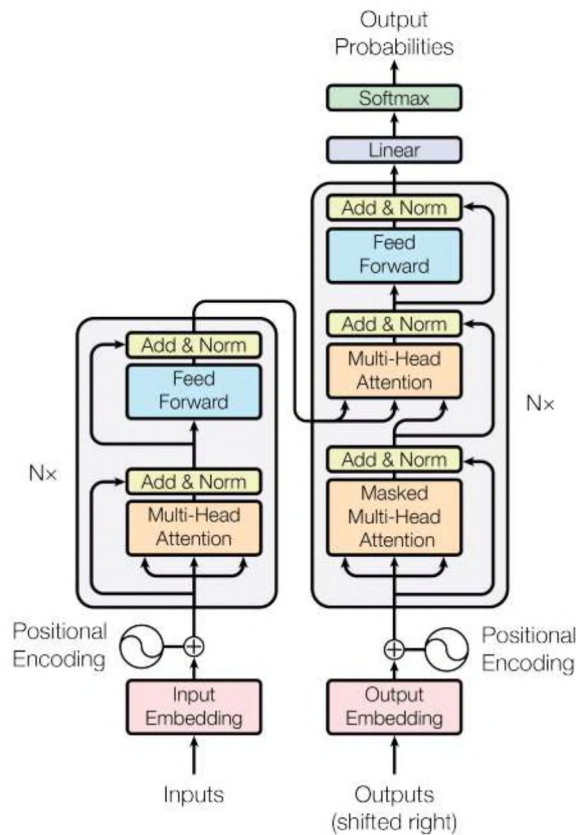


Рисунок 2.6 – Типова архітектура мережі Transformer

Помітний успіх Transformer-подібних архітектур для аудіо пояснюється їхньою здатністю працювати з послідовностями даних безпосередньо, враховуючи контекст та взаємозв'язки між елементами. Зокрема, T-F Transformer і Speech-Transformer використовують механізми само уваги для ефективної обробки акустичних сигналів.

Успіх SpecAugment вказує на важливість аргументації даних для досягнення кращих результатів у випадках, коли набір даних обмежений. Цей метод використовує випадкові трансформації в часовому та частотному просторі для створення різноманітних варіацій вхідних аудіо даних, поліпшуючи тим самим загальну здатність моделі до узагальнення на нові дані. Однією з головних переваг Transformer-подібних архітектур є їхня паралельна обробка даних, що дозволяє збільшити ефективність навчання на великих об'ємах даних. Це стає

особливо важливим у великих проектах, де моделі повинні пристосовуватися до різних типів аудіо даних та величезних обсягів інформації. Останні досягнення також вказують на те, що Transformer-подібні архітектури можуть бути успішно використані у сферах аудіо генерації та синтезу звуку для різних застосувань. Їхня здатність до вивчення складних звукових патернів і створення реалістичних аудіо даних відкриває нові можливості у сфері мистецтва, розваг та виробництва мультимедійних контентів [22].

2.2 Технічне забезпечення що необхідне для роботи нейронної мережі

У зв'язку зі стрімким розвитком сучасних технологій, глибоке навчання та нейронні мережі перетворюють сучасний ландшафт обробки інформації та штучного інтелекту. Ці інноваційні підходи стають визначальними в різноманітних галузях, таких як машинне навчання, медицина, фінанси та інші, де вони знаходять широке практичне застосування.

Важливо відзначити, що ефективність функціонування нейронних мереж не залежить лише від якості алгоритмів, але й від належного технічного забезпечення. Технічна інфраструктура, що підтримує глибоке навчання, налічує ряд ключових компонентів, кожен з яких виконує важливу роль у забезпеченні високої ефективності нейронних мереж.

У цьому контексті, важливо ретельно розглянути ключові аспекти технічного забезпечення та визначити їхню роль та взаємодію в процесі тренування та інференції. Зі зростанням обсягів даних та складності поставлених завдань перед нейронними мережами, наростають вимоги до обчислювальної потужності, пам'яті, інфраструктури для зберігання даних та ефективного зв'язку між елементами системи.

2.2.1. Центральний процесор (CPU)

Нейронні мережі є ключовим елементом сучасних систем машинного навчання та штучного інтелекту. Вони спроектовані для моделювання та

вирішення складних завдань, навіть тих, які раніше були важкі для програмного забезпечення з визначеною логікою [23]. Ці мережі стали ефективними у вирішенні задач від розпізнавання об'єктів до глибокого навчання та генерації зображень (рис. 2.7).



Рисунок 2.7 – Зображення центрального процесора від компанії Intel

Центральний процесор (CPU) відіграє вирішальну роль у функціонуванні нейронних мереж, що вимагає як навчання, так і інференції. Основна функція CPU в цьому контексті — забезпечення обчислювальної потужності для різноманітних завдань, пов'язаних з нейронними мережами.

Розглянемо основні аспекти його взаємодії з нейронними мережами:

1. Обчислення параметрів. Процес навчання нейронних мереж передбачає оптимізацію параметрів моделі. Центральний процесор використовується для виконання численних обчислень, пов'язаних із зміною ваг та параметрів мережі під час навчання.

2. Пряме та зворотне поширення. У фазі інференції, CPU відповідає за обчислення та передачу інформації через різні шари нейронної мережі. Під час зворотного поширення, яке використовується для навчання та корекції параметрів, CPU відновлює обчислення градієнтів та виконує оновлення параметрів.

3. Оптимізація. Використовуючи різні оптимізаційні алгоритми, такі як стохастичний градієнтний спуск, CPU сприяє оптимізації параметрів мережі для покращення її продуктивності.

4. Інтерфейс з пам'яттю. CPU взаємодіє з пам'яттю, де зберігаються дані, необхідні для нейронних мереж, та де здійснюється читання та запис цих даних.

5. Управління завданнями. У складних системах, які включають багато завдань, CPU відповідає за ефективне управління виконанням різних операцій та задач.

Нарешті, важливо зазначити, що разом із CPU, для завдань, пов'язаних з нейронними мережами, широко використовуються графічні процесори (GPU) та тензорні процесори (TPU), які спеціалізуються на обробці паралельних обчислень і можуть значно прискорити роботу нейронних мереж.

2.2.2 Графічні процесори (GPU)

Графічні процесори GPU відіграють важливу роль в роботі нейронних мереж, забезпечуючи значну швидкість обчислень, необхідну для навчання та виконання інференції в глибоких нейронних мережах (рис. 2.8) [24].



Рисунок 2.8 – Зображення графічного процесора розробленого компанією Nvidia

Основні функції графічних процесорів у сфері нейронних мереж включають:

1. Паралельні обчислення. ГП призначені для обробки багатьох обчислень одночасно, що робить їх ідеальними для великих обчислювальних завдань, таких як навчання нейронних мереж, де часто виникає потреба в одночасній обробці багатьох даних або параметрів.

2. Масштабовані обчислення. ГП можуть легко масштабуватись для роботи з великими об'ємами даних, що робить їх ефективними для нейронних мереж, які вимагають великої кількості параметрів та обчислень.

3. Глибоке навчання (Deep Learning). Глибокі нейронні мережі, особливо з використанням складних архітектур, можуть вимагати значних обчислювальних ресурсів. ГП дозволяють реалізувати глибоке навчання, забезпечуючи високу швидкість обчислень для забезпечення швидкого навчання та інференції.

4. CUDA та інші програмні інтерфейси. Багато виробників графічних процесорів надають програмні інтерфейси, такі як CUDA (Compute Unified Device Architecture) від NVIDIA, які дозволяють розробникам легко використовувати графічний процесор для обчислень у мовах програмування, таких як Python чи C++.

5. Спеціалізовані графічні процесори для машинного навчання. Останнім часом виникли графічні процесори, спеціально розроблені для використання в машинному навчанні. Наприклад, NVIDIA випустила лінійку графічних процесорів NVIDIA Tesla та NVIDIA A100, які мають спеціалізовані ядра для прискорення обчислень у сфері машинного навчання.

2.2.3 Тензорний процесор (TPU)

Тензорні процесори відіграють важливу роль в роботі нейронних мереж, особливо у сфері глибокого навчання [25]. Вони спеціально розроблені для виконання операцій, які відбуваються під час навчання і інференсу нейронних мереж, зокрема операцій над тензорами (рис. 2.9).

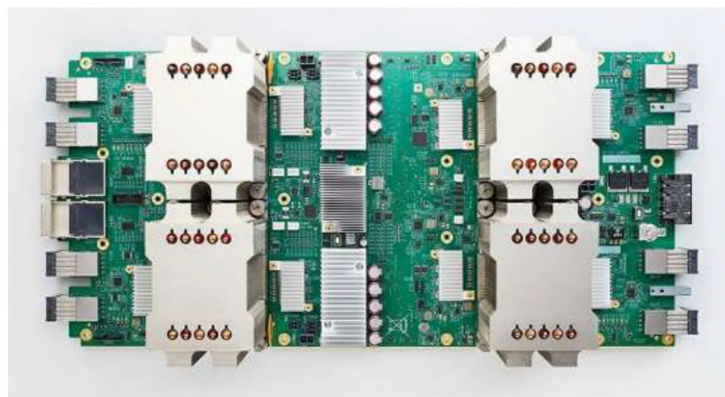


Рисунок 2.9 – Тензорний процесор Google для глибокого навчання

Основні функції тензорних процесорів у роботі нейронних мереж:

1. Виконання паралельних обчислень. Тензорні процесори призначені для швидкого виконання паралельних обчислень, які часто зустрічаються в операціях нейронних мереж. Це дозволяє прискорювати як навчання, так і інференс.

2. Оптимізація для тензорних операцій. Тензорні процесори оптимізовані саме для операцій над тензорами, які є основними об'єктами обчислень у глибокому навчанні. Це включає в себе операції, такі як множення матриць, згортки, та інші, що регулярно використовуються в нейронних мережах.

3. Ефективність енергоспоживання. Тензорні процесори спроектовані таким чином, щоб бути ефективними у використанні енергії, що робить їх ідеальними для вбудованих систем та інших пристроїв із обмеженими ресурсами.

4. Підтримка великих обсягів даних. Завдяки великій швидкості обчислень і високій продуктивності, тензорні процесори дозволяють ефективно працювати з великими обсягами даних, що є важливим для глибокого навчання.

5. Спеціалізований апаратний прискорювач. Тензорні процесори виконують операції, пов'язані з глибоким навчанням, значно швидше, ніж загально призначені процесори, такі як центральні процесори (CPU) чи графічні процесори (GPU), оскільки вони спеціалізовані саме на це.

В сукупності ці особливості дозволяють тензорним процесорам ефективно обробляти складні нейронні мережі, знижуючи час навчання і забезпечуючи високу швидкість інференсу, що робить їх важливим елементом у розвитку та застосуванні штучних нейронних мереж.

2.2.4 Оперативна пам'ять (RAM)

Оперативна пам'ять (RAM) є ключовим елементом у роботі нейронних мереж і виконує ряд критичних функцій, які дозволяють їм ефективно функціонувати (рис. 2.10).



Рис 2.10 – Модуль оперативної пам'яті від компанії Kingston

У своєму визначенні, оперативна пам'ять представляє собою тимчасовий сховище даних, до якого комп'ютер чи інший електронний пристрій має швидкий доступ для читання та запису [26].

У контексті нейронних мереж, де робота з великими обсягами даних та параметрів відбувається швидко та паралельно, роль оперативної пам'яті виявляється особливо важливою. Нижче наведено деякі ключові аспекти ролі оперативної пам'яті в нейронних мережах:

1. Зберігання параметрів моделі. Ваги і зсуви (biases) нейронної мережі є основними параметрами, які визначають її поведінку. Оперативна пам'ять використовується для зберігання цих параметрів під час навчання і використання моделі.

2. Виконання обчислень. Під час навчання та використання моделі, величезна кількість обчислень відбувається в нейронній мережі. Оперативна пам'ять служить для тимчасового зберігання проміжних результатів обчислень та активів нейронів під час прямого та зворотного поширення (forward and backward propagation) градієнтів.

3. Міні-батчі та оптимізації. У навчанні нейронних мереж зазвичай використовуються міні-батчі даних для ефективного оновлення параметрів. Оперативна пам'ять використовується для зберігання цих міні-батчів та проміжних результатів, що дозволяє використовувати оптимізаційні алгоритми, такі як стохастичний градієнтний спуск.

4. Кешування даних. Зберігання часткових результатів обчислень та проміжних даних в оперативній пам'яті може покращити швидкість виконання операцій в мережі, особливо при великих об'ємах даних.

5. Робота з великими моделями. Для великих та складних моделей, таких як глибокі нейронні мережі, велика кількість параметрів може займати значну частину оперативної пам'яті.

6. Управління ресурсами. Оперативна пам'ять також використовується для управління ресурсами, такими як пам'ять для буферів обчислень та тимчасових даних, що може впливати на ефективність роботи мережі.

Загалом, оперативна пам'ять є критичною для ефективного функціонування нейронних мереж, та впливати на можливість працювати з різними обсягами даних, що також впливає на продуктивність моделі нейронної мережі.

2.2.5 Система охолодження

Система охолодження є надзвичайно важливою для правильної роботи комп'ютера. Компоненти комп'ютера, такі як центральний процесор, графічний процесор, та інші елементи, можуть створювати значний обсяг вироблення тепла під час роботи [27]. Якщо це тепло не видаляється ефективно, може виникнути перегрів, що може призвести до різноманітних проблем в системах технічного забезпечення яке використовується для роботи моделі нейронної мережі. Для досягнення ефективного охолодження використовуються вентилятори, теплові труби, теплові пластини та інші технології. Також важливо правильно розташовувати комп'ютер та дотримуватися рекомендацій виробників з вентиляції приміщення. Основними аспектами використання системи охолодження ЕОМ є:

1. Запобігання перегріву. Комп'ютерні компоненти, такі як процесори та графічні картки, оптимально працюють при певних температурах. Якщо ці температури перевищуються через недостатнє охолодження, компоненти можуть знизити свою продуктивність або автоматично зменшити тактову частоту для уникнення перегріву. Перегрів може призвести до пошкодження компонентів. Наприклад, довготривале використання при високих температурах може спричинити втомленість матеріалів і погіршити надійність елементів.

2. Підтримання стабільної роботи: Система охолодження допомагає забезпечити стабільну температуру пристроїв, що може бути важливим для забезпечення надійності та довговічності комп'ютера.

3. Збільшення продуктивності: При належному охолодженні компоненти можуть працювати на вищих тактових частотах, що призводить до збільшення продуктивності. Багато сучасних комп'ютерних компонентів також використовують автоматичне регулювання тактової частоти в залежності від температури, щоб уникнути перегріву. Існує кілька методів охолодження

комп'ютерів, призначених для забезпечення ефективного видалення тепла, що виробляється під час роботи компонентів. Нижче наведено детальний їх огляд.

1. Вентилятори. Найпоширенішим методом охолодження є використання вентиляторів. Ці пристрої створюють потік повітря, виводять гаряче повітря з корпусу та подають свіже для охолодження компонентів. Вентилятори використовуються як для охолодження корпусу, так і для конкретних компонентів, наприклад, процесорів та відеокарт.

2. Теплові труби. Теплові труби є ефективним методом транспортування тепла від гарячих компонентів до більш віддалених областей охолодження. Вони використовують внутрішні теплові транспорти, що допомагає ефективно переносити тепло і забезпечувати оптимальне охолодження.

3. Теплові пластини. Теплові пластини або радіатори використовуються для збільшення площі, доступної для розсіювання тепла. Вони розташовуються на поверхні гарячих компонентів, таких як процесори, і використовуються для оптимізації процесу охолодження.

4. Рідинне охолодження. Системи рідинного охолодження використовують рідину для передачі тепла від компонентів до радіатора, де відбувається охолодження. Цей метод дозволяє досягти високої ефективності охолодження, особливо в високопродуктивних системах.

5. Фазове охолодження. Використовуючи пару для видалення тепла, системи фазового охолодження переносять рідинний холодоагент через фазові переходи, забираючи тепло від компонентів та виводячи його на радіаторі.

2.3 Перспективи розробки нових архітектур і моделей нейронних мереж

В сучасному світі інформаційних технологій та штучного інтелекту виникає все більше можливостей для розвитку та вдосконалення систем розпізнавання мовлення. Однією з найперспективніших галузей є створення інформаційних систем, які здатні не лише розпізнавати слова, але і аналізувати контекст та

взаємодіяти з користувачем на більш глибокому рівні. У цьому контексті важливим етапом є розробка нових архітектур та моделей для аналізу мовлення на основі нейронних мереж.

Однією з ключових перспектив є покращення точності розпізнавання мовлення. Розробники активно працюють над створенням більш складних та глибоких нейронних мереж, які забезпечують високу точність навіть в умовах шуму чи різноманітних акцентів.

Розширення мовного покриття також виступає важливим завданням. Здатність розпізнавати різні мови та діалекти дозволяє створювати універсальні системи, які користуються популярністю в різних країнах та серед різних груп користувачів.

Адаптація до специфічних галузей, таких як медицина або технічна сфера, також відіграє важливу роль. Високотехнологічні галузі вимагають спеціалізованих моделей для ефективного розпізнавання термінів та виразів, характерних саме для цих областей.

Робота з різноманітними типами аудіо даних, такими як розмови, подкасти чи аудіо книги, також є актуальною задачею. Створення моделей, які ефективно працюють з різними форматами та стилями мовлення, розширює сферу застосування систем розпізнавання мовлення.

Емоційне мовлення стає все важливішим аспектом аналізу. Розробка моделей, які здатні розпізнавати та аналізувати емоції в мовленні, відкриває нові можливості для розуміння контексту та індивідуальних потреб користувача.

Забезпечення конфіденційності та безпеки особистої інформації є однією з головних турбот при розробці систем розпізнавання мовлення. Розробники шукають нові методи та архітектури для збереження конфіденційності при обробці чутливої інформації користувачів.

Інтеграція із суміжними технологіями, такими як природні мови, машинне навчання та обробка сигналів, сприяє створенню комплексних систем, які можуть надавати більше функцій та бути універсальними у використанні, та в подальшому надає можливість повної автоматизації всіх процесів.

Оптимізація для різних пристроїв, включаючи мобільні телефони, вбудовані системи та хмарні платформи, відкриває можливості для ширшого розповсюдження та доступності систем розпізнавання мовлення.

У світлі цих перспектив розробка нових архітектур та моделей для аналізу мовлення залишається активною та перспективною галуззю, яка визначає подальший розвиток сучасних технологій.

Висновки до розділу 2

У цьому розділі детально розглядається широкий спектр архітектур нейронних мереж, включаючи згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN), та використання персиптронів для класифікації аудіо даних. Також розглядається важливість технічного забезпечення для ефективної роботи нейронної мережі, особливо в контексті аналізу даних.

1. Згорткові нейронні мережі (CNN). CNN виявляються потужним інструментом для обробки візуальних даних, таких як зображення. Вони використовують фільтри для виявлення просторових шаблонів, що дозволяє їм ефективно вирішувати задачі розпізнавання об'єктів та зображень.

2. Рекурентні нейронні мережі (RNN). RNN стають необхідним інструментом для обробки послідовних даних, які мають важливий контекст часу, наприклад, текст чи часові ряди. Вони можуть моделювати динамічні залежності та використовуються в задачах, де порядок даних має значення.

3. Використання персиптронів для аудіо класифікації. Персиптрони виявляються ефективними для аудіо класифікації, оскільки вони можуть взяти на себе завдання аналізу акустичних ознак і класифікації звукових сигналів. Їхній внутрішній шар може розпізнавати патерни у звукових вхідних даних.

4. Важливість технічного забезпечення. Підкреслюється, що технічне забезпечення має велике значення для нейронних мереж. Обчислювальний ресурс, вірно налаштовані гіперпараметри та відповідне програмне забезпечення є ключовими факторами для успішного розвитку та навчання моделей.

Цей аналітичний огляд архітектур та технічних аспектів нейронних мереж надає читачам глибоке розуміння їхнього функціоналу та визначає ключові аспекти, які слід враховувати при їхньому використанні для аналізу та обробки різноманітних видів даних.

У цілому, Transformer-подібні архітектури виявляються перспективними та динамічно розвиваються у величезному спектрі аудіовізуальних застосувань, від розпізнавання мови та аудіозаписів до створення музичних творів та інших аудіо даних. При розгляді Transformer-подібних архітектур для аудіо важливо відзначити інтеграцію таких моделей у сферу розпізнавання мови. Transformer-підхід дозволяє здійснювати більш ефективну обробку аудіо сигналів, що виявляється корисним в завданнях автоматичного перекладу, транскрипції аудіо та інших задачах, пов'язаних з обробкою мовленнєвих відомостей. Деякі розширені Transformer-подібні архітектури для аудіо також використовують механізми взаємодії між різними модальностями, такими як текст і звук. Це відкриває нові можливості для завдань, таких як автоматичний аудіо опис для відео чи створення текстових описів аудіо файлів.

Застосування Transformer-подібних архітектур у музичних дослідженнях також відкриває нові перспективи у галузі генерації музики, аналізу аудіо сигналів та взаємодії з музичним контентом. Однак важливо відзначити, що успішне використання Transformer-подібних архітектур для аудіо часто вимагає великих обсягів обчислювальних ресурсів та даних для навчання. Такі моделі можуть бути важкими для реалізації в реальному часі або на ресурсоменшених пристроях, таких як мобільні телефони.

У майбутньому можна очікувати подальший розвиток Transformer-подібних архітектур для аудіо та їхнє використання у різноманітних застосуваннях, що зробить їх ще більш потужними та універсальними інструментами в області обробки аудіо даних.

Узагальнюючи, графічні процесори грають ключову роль у розвитку та застосуванні нейронних мереж, забезпечуючи ефективність та швидкість обчислень, що є важливими для успішного навчання та використання нейронних

мереж у практичних застосування, система охолодження є критичним елементом для забезпечення стабільної та ефективної роботи комп'ютера, а також для збереження його функціональності та тривалого терміну служби а вибір методу охолодження залежить від потреб користувача, його бюджету, а також конкретних вимог до ефективності та надійності системи, хоча на даний момент багато сучасних комп'ютерів використовують комбінації цих методів для забезпечення оптимального охолодження.

РОЗДІЛ 3

СТВОРЕННЯ ІС ДЛЯ АНАЛІЗУ МОВЛЕННЯ НА ОСНОВІ БІБЛІОТЕКИ GOOGLE SPEECH TO TEXT

3.1 Використання бібліотеки Google Cloud Speech to Text в ІС

Google Cloud Speech to Text – це потужний хмарний інструмент для розпізнавання мовлення, що за допомогою передових технологій може достовірно й швидко перетворювати аудіо дані в текст понад 120 мовами, що робить його ідеальним інструментом для впровадження в ІС систему, що призначена для розпізнавання мовлення [28][29]. Це потужне рішення від Google, яке дозволяє розробникам використовувати автовідповідачі в колл-центрах, взаємодіяти з користувачами за допомогою IoT-пристроїв та конвертувати текстові повідомлення у голосовий формат.

Служба Speech-to-Text, раніше відома як Cloud Speech API, була представлена у 2016 році. За інформацією від Google, протягом перших років її існування використання API збільшувалося вдвічі кожні шість місяців. Це рішення використовує передові алгоритми глибокого навчання від Google для автоматичного розпізнавання мови (ASR) [30].

Також є можливість швидко впровадити ASR у хмарі, використовуючи API, або навіть локально, використовуючи локальне перетворення мовлення на текст, інтегруючи технології розпізнавання мовлення Google у своє локальне рішення. Дотримуючись вимог щодо розміщення даних та відповідаючи вимогам безпеки, є змога контролювати свою інфраструктуру, отримуючи при цьому переваги технології розпізнавання мовлення з високо-захищеними мовними даними.

Для того щоб використовувати цю бібліотеку користувачу необхідно спочатку створити обліковий запис Google Cloud Platform і налаштувати проект [31]. Наступним кроком в інтеграції бібліотеки даних Google Cloud Speech to text в ІС є ввімкнення API перетворення мовлення аудіо в текст в Console Cloud, та в подальшому згенерувати ключ API, файл якого буде використовуватись в

програмному коді нашої ІС. Нижче буде наведено поетапне ввімкнення API Speech to Text та створення ключа служби [32].

1. Створення акаунту в Google Cloud (рис 3.1).

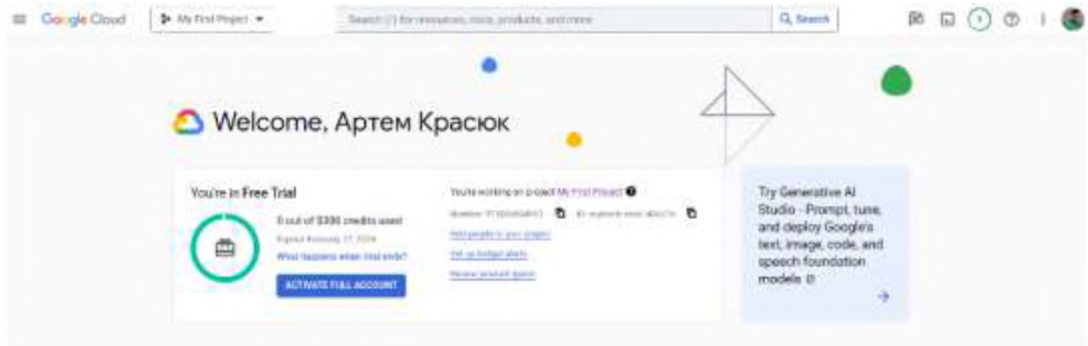


Рисунок 3.1 – Створення акаунту в сервісі

2. Створення нового проекту. Для цього необхідно натиснути Select a project та увімкнення Speech to Text API (рис 3.2 – 3.3).

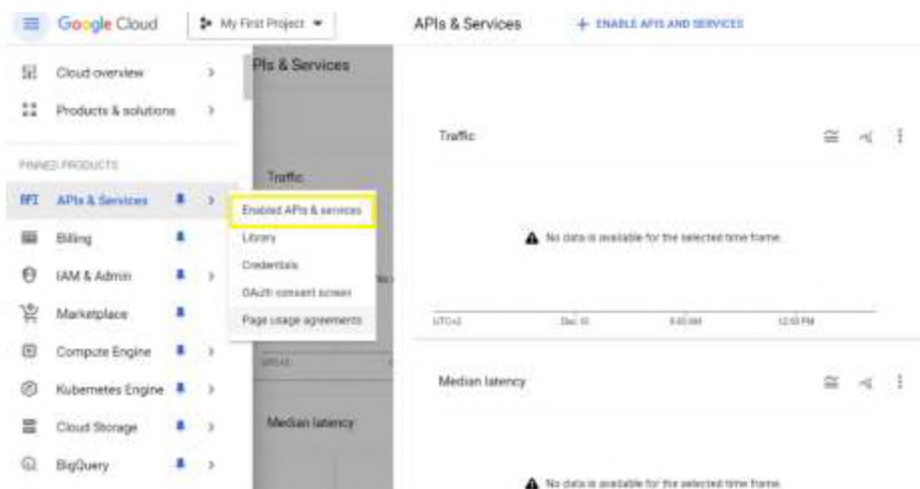


Рисунок 3.2 – Створення нового проекту

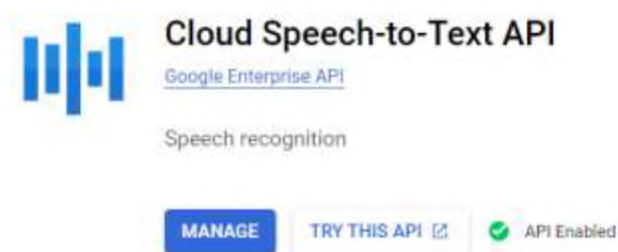


Рисунок 3.3 – Ввімкнення функції Speech to Text

3. Створення ключа служби. Для створення ключа служби необхідно перейти до «IAM & адміністрування», обрати «Ключі служби» та натиснути «Створити ключ». Для того щоб інтегрувати його в нашу ІС необхідно обрати формат файлу Json (рис 3.4).

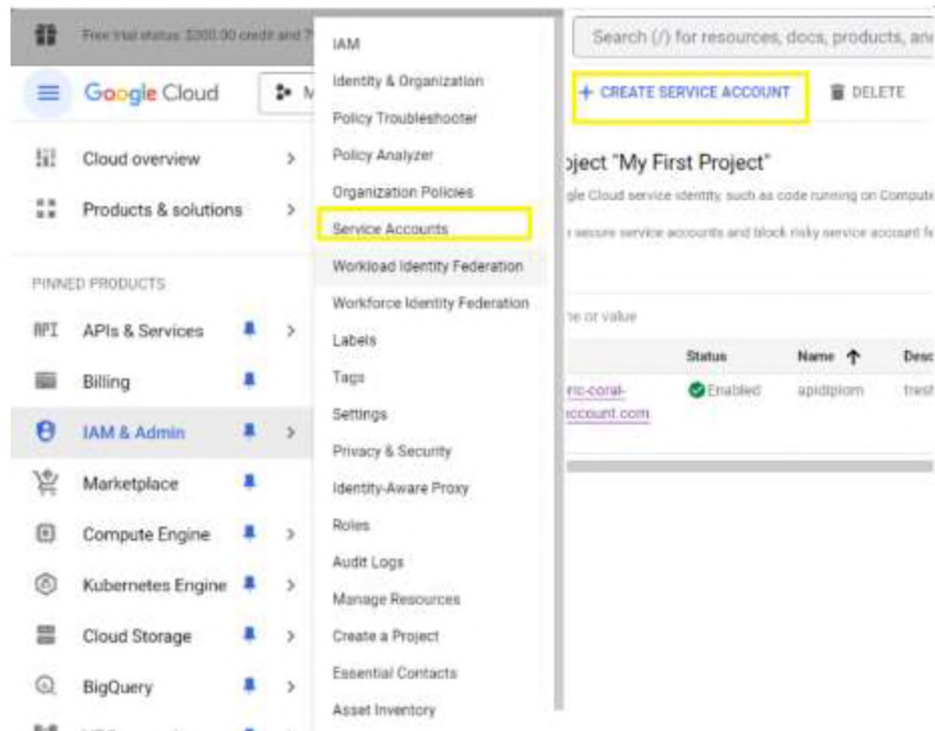


Рисунок 3.4 – Створення ключа служби

Узагальнюючи, використання сервісу Google Cloud Speech to text при інтеграції в систему розпізнавання мовлення є надзвичайно корисним, так як, надає можливість, не використовувати бази даних, для навчання нейронної мережі.

3.2 Области застосування ІС на основі бібліотеки Google Speech to Text

Зростання попиту на автоматизовані рішення для обробки аудіо даних визначає актуальність застосування систем розпізнавання мовлення.

Однією з сучасних технологій у цій області є Інформаційна Система (ІС) для розпізнавання мовлення на основі бібліотеки Google Speech to Text.

Ця система має великий потенціал та розкриває широкі перспективи в різних сферах діяльності. ІС, яка базується на бібліотеці Google Speech to Text, створена для відповіді на цей виклик, пропонуючи надійний та ефективний інструмент для перетворення мовлення у текстовий формат.

Прогрес технологій розпізнавання мовлення відкриває нові можливості для оптимізації робочих процесів та полегшення взаємодії з інформаційними системами. Розгляд різноманітних галузей використання ІС для розпізнавання мовлення розкриває потенціал цієї технології та підкреслює її важливість у сучасному цифровому світі, а саме:

1. Транскрибація інтерв'ю та лекцій. Інформаційна система для розпізнавання мовлення на основі Google Speech to Text забезпечує автоматизовану транскрибацію аудіозаписів інтерв'ю, лекцій та навчальних матеріалів. Це спрощує обробку та зберігання інформації, полегшуючи доступ до текстової версії мовлення.

2. Розробка голосових інтерфейсів. ІС дозволяє розробляти голосові інтерфейси для мобільних додатків та веб-сайтів, покращуючи зручність взаємодії користувачів з технічними системами через голосове управління.

3. Медична сфера. Система допомагає у медичній сфері, автоматизуючи створення текстових звітів лікарів на основі аудіозаписів консультацій та медичних виступів, що полегшує ведення медичної документації.

4. Навчання та освіта. Використання ІС для транскрибації лекцій, створення субтитрів та надання підтримки студентам із втратою слуху, що сприяє ефективній освітній взаємодії.

5. Редагування відео та аудіоматеріалів. Система допомагає в редагуванні відеоматеріалів шляхом створення автоматичних субтитрів або перетворення аудіозаписів у текстовий формат.

6. Комунікація та трансляції. Використання для автоматичного перекладу мовлення на текст у комунікаційних платформах та під час трансляцій для полегшення розуміння великих аудиторій.

7. Системи відстеження та аналізу обговорень. Застосування для виявлення ключових слів та тем в аудіозаписах для аналізу обговорень.

8. Технічна підтримка та обробка звернень. ІС використовується для створення голосових інтерфейсів для систем технічної підтримки та автоматизованої обробки телефонних звернень.

9. Розваги та ігрова індустрія. Впровадження голосового управління та розпізнавання мовлення в відеоіграх та системах віртуальної реальності для покращення взаємодії гравців з віртуальним середовищем.

3.3 Використання мови програмування Python для створення ІС

В сучасному світі інформаційні технології відіграють ключову роль у різноманітних сферах життя. Однією з інноваційних тенденцій є розробка інформаційних систем для аналізу мовлення. В цьому розділі розглянемо процес створення такої системи, використовуючи мову програмування Python [33].

Python, зазнаючи визнання своєї простоти та гнучкості синтаксису, став широко використовуваною мовою програмування в інформаційних технологіях. В цьому контексті, аналіз мовлення визначається як ключовий аспект для численних завдань, таких як розпізнавання мовлення, переклад тексту, аналіз настрою та інші. Python стає популярним вибором для створення програм, що здійснюють аналіз мовлення [34].

Python, визнаний своєю читабельністю та простотою синтаксису, стає однією з найпопулярніших мов програмування. Його велика спільнота розробників забезпечує стабільний розвиток та багатий вибір бібліотек для різних задач [35].

У сфері розпізнавання мовлення ця мова програмування використовується в бібліотеці SpeechRecognition, що надає інтерфейс для роботи з різними сервісами розпізнавання мовлення. Простота використання цих інструментів дозволяє легко створювати програми для розпізнавання мовлення.

Python також є популярним вибором для розробки систем машинного навчання. Бібліотеки, такі як TensorFlow, PyTorch та scikit-learn, дозволяють

розробникам реалізувати складні алгоритми для аналізу мовлення. Застосування нейронних мереж у розпізнаванні мовлення, наприклад, у *deepspeech* від Mozilla, є показником розвитку цієї галузі [36].

Створення голосових асистентів, які взаємодіють з користувачами за допомогою мовлення, також відзначається використанням Python. Бібліотеки, такі як *SpeechRecognition*, інтегрують розпізнавання мовлення в голосові інтерфейси.

Однією з ключових переваг використання Python для аналізу мовлення є широкий вибір інструментів та бібліотек, що полегшують розробку. Простий синтаксис та велика активна спільнота роблять Python зручним вибором для розробників у цій галузі [37].

Python застосовується у різних галузях, включаючи бізнес, освіту, медицину. Системи аналізу мовлення знаходять застосування в автоматизації бізнес-процесів, навчанні та розвагах.

У світі, де взаємодія з технологією стає все більше голосовою, Python виявляється потужним інструментом для створення програм, які аналізують та розпізнають мовлення. Простота та гнучкість роблять його універсальним інструментом для розробки систем, що аналізують та розпізнають мовлення. Майбутні розвідки можуть включати в себе подальше вдосконалення алгоритмів розпізнавання, розширення функціоналу голосових асистентів та використання технологій штучного інтелекту для забезпечення ще більш точного та контекстно-чутливого аналізу мовлення. Загальні тенденції розвитку аналізу мовлення вказують на те, що в майбутньому ця галузь буде спрямована на досягнення вищого рівня точності та взаємодії. Відмінність та індивідуальні особливості голосових команд можуть бути враховані завдяки використанню технологій машинного навчання та нейронних мереж.

Однією з перспектив розвитку є розширення меж традиційного розпізнавання мовлення. Системи можуть вивчати не лише сам текст, а й емоційний стан користувача, його індивідуальні особливості та навіть спробувати передбачити інтенції за допомогою аналізу голосового тону та інтонації.

Технології штучного інтелекту також можуть сприяти розвитку інтерактивних систем. Голосові асистенти можуть стати більш ефективними та

адаптивними до потреб користувачів, забезпечуючи персоналізовану інтеракцію та забезпечуючи більший рівень задоволення від взаємодії.

Окрім цього, можливі напрями розвитку також включають розширення можливостей мульти-мовного розпізнавання та перекладу мовлення. Системи будуть більш адаптовані до різних мов, діалектів та акцентів, що поліпшить їхню глобальну доступність та застосовність. Нижче наведено програмний код створеної ІС з детальним описом.

Створення інтерфейсу. Інтерфейс цієї ІС створюється за допомогою бібліотеки «tkinter». Основні частини коду відповідальні за створення інтерфейсу можна розділити за наступними групами:

1. Створення вікна та заголовку (рис 3.5).

```
root = tk.Tk()
root.title("Розпізнавання мовлення")
```

Рисунок 3.5 – Створення інтерфейсу ІС

2. Створення та розміщення візуальних елементів, таких як, мітка, текстові поля та кнопка. Створюються елементи інтерфейсу, такі як мітка (Label), текстові поля (Text) та кнопка (Button). Ці елементи розміщуються на вікні за допомогою методу pack (рис 3.6).

```
self.label = tk.Label(master, text="Натискайте кнопку та говоріть щось:")
self.label.pack(pady=10)

self.result_text = tk.Text(master, height=5, width=40)
self.result_text.pack(pady=10)

self.instant_text = tk.Text(master, height=5, width=40)
self.instant_text.pack(pady=10)

self.button = tk.Button(master, text="Розпізнати", command=self.start_r
self.button.pack(pady=20)
```

Рисунок 3.6 – Створення кнопок для управління ІС

3. Ініціалізація об'єктів Google Cloud та PyAudio. Тут ініціалізуються об'єкти для роботи з Google Cloud Speech API та PyAudio (рис 3.7).

```
gcp.b = vladqto.vladqto()
gcp.vladqto.vladqto = io
gcp.stt = gcp.vladqto.stt()
```

Рисунок 3.7 – Додавання до ІС бібліотек для роботи з мовленням

4. Головний цикл інтерфейсу. Цей рядок запускає головний інтерфейс, який очікує події та взаємодіє з користувачем (рис 3.8).

```
root.mainloop()
```

Рисунок 3.8 – Додавання функції що відповідає за запуск інтерфейсу

Усі ці частини коду разом створюють графічний інтерфейс програми для розпізнавання мовлення.

Наступним етапом в створенні цієї системи є імпорт необхідних бібліотек та створення їх класів. В першу чергу для імпорту цих бібліотек необхідно їх встановити в середовищі Visual Studio.

Бібліотеки які будуть імпортовані в середовище:

- tkinter. Для створення графічного інтерфейсу;
- speech та service_account з Google Cloud. Для роботи з Speech-to-Text API;
- io. Для роботи з байтами та потоками вводу/виводу;
- time. Для роботи зі споживанням часу в кодї;
- pyaudio. Для роботи з аудіо.

Після імпорту бібліотек відбувається створення їх класів та методів.

1. Метод `recognize_speech`. Цей метод викликається для розпізнавання аудіо, використовуючи Google Cloud Speech API. Результат розпізнавання виводиться в текстові поля (рис 3.9).

```
def recognize_speech(self, content):
    # Створення об'єкту для розпізнавання мовлення
    audio = speech.RecognitionAudio(content=content)
    config = speech.RecognitionConfig(
        encoding=speech.RecognitionConfig.AudioEncoding.LINEAR16,
        sample_rate_hertz=16000,
        language_code="uk-UA",
    )

    try:
        # Виклик API для розпізнавання
        response = self.client.recognize(config=config, audio=audio)
        recognized_text = ""
        for result in response.results:
            recognized_text += result.alternatives[0].transcript + "\n"

        # Виведення тексту в моментальне та основне текстові поля
        self.instant_text.delete(1.0, tk.END)
        self.instant_text.insert(tk.END, recognized_text)
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, recognized_text)
    except Exception as e:
        # Обробка помилок розпізнавання
        self.result_text.delete(1.0, tk.END)
        self.result_text.insert(tk.END, "Помилка розпізнавання мовлення.")
```

Рисунок 3.9 – Функція що викликає Google Cloud Speech API для розпізнавання мовлення

2. Метод `start_recognition`. Цей метод викликається при натисканні кнопки "Розпізнати". Він очищає текстові поля, виводить початковий текст у моментальне поле та запускає запис аудіо (рис 3.10).

```
def start_recognition(self):
    # Очищення текстових полів
    self.result_text.delete(1.0, tk.END)
    self.instant_text.delete(1.0, tk.END)

    # Виведення тексту перед розпізнаванням
    self.instant_text.insert(tk.END, "Початок розпізнавання...")

    # Запуск запису аудіо
    self.record_audio()
```

Рисунок 3.10 – Функція що призначена для початку запису аудіо

3. Створення класу `SpeechRecognitionApp`. У конструкторі класу ініціалізуються елементи графічного інтерфейсу та необхідні засоби для роботи з аудіо та Google Cloud Speech API (рис 3.11).

```
class SpeechRecognitionApp:
    def __init__(self, master):
        # Ініціалізація головного вікна
        self.master = master
        self.master.title("Розпізнавання мовлення")

        # Створення елементів інтерфейсу: мітка, текстові поля та кнопка
        self.label = tk.Label(master, text="Натискайте кнопку та говоріть")
        self.label.pack(pady=10)

        self.result_text = tk.Text(master, height=5, width=40)
        self.result_text.pack(pady=10)

        self.instant_text = tk.Text(master, height=5, width=40)
        self.instant_text.pack(pady=10)

        self.button = tk.Button(master, text="Розпізнати", command=self)
        self.button.pack(pady=20)

        # Ініціалізація об'єктів Google Cloud Speech API та PyAudio
        self.client = speech.SpeechClient()
        self.audio_stream = None
        self.p = pyaudio.PyAudio()
```

Рисунок 3.11 – Створення класу `SpeechRecognitionApp`

4. Метод `record_audio`. Цей метод відповідає за запис аудіо. Він використовує бібліотеку `PyAudio` для вивчення звукового сигналу та передачі його на розпізнавання через `Speech-to-Text API` (рис 3.12 – 3.13).

```
def record_audio(self):
    # Налаштування параметрів запису аудіо
    CHUNK = 1024
    FORMAT = pyaudio.paInt16
    CHANNELS = 1
    RATE = 16000

    # Відкриття потоку запису аудіо
    stream = self.p.open(format=FORMAT,
                        channels=CHANNELS,
                        rate=RATE,
                        input=True,
                        frames_per_buffer=CHUNK)

    print("Запис аудіо")

    frames = []

    try:
        # Запис аудіо
        while True:
            data = stream.read(CHUNK)
            frames.append(data)
```

Рисунок 3.12 – Метод що відповідає за запис та передачу даних

```

        # Якщо потрібна прервати запис, вийти з циклу
        if not self.audio_stream:
            break
    except Exception as e:
        print('Помилка запису аудіо: {}'.format(e))

    print('* * * Запис завершено')

    # Зупинка потоку запису
    stream.stop_stream()
    stream.close()

    # Об'єднання байтослів об'єкта в один
    content = b''.join(frames)

    # Виклик методу для розпізнавання аудіо
    self.recognize_speech(content)

def stop_recognition(self):
    # Зупинка запису аудіо
    self.audio_stream = None

```

Рисунок 3.13 – Метод що відповідає за запис та передачу даних

5. Головна частина програми. В головній частині програми створюється головне вікно і запускається головний цикл інтерфейсу (рис 3.14).

```

def main():
    # Створення головного вікна
    root = tk.Tk()

    # Встановлення шляху до ключа служби Google Cloud
    key_path = r"C:\D\0\1\euphoric-coral-406518-e8184404426c.json"
    # Встановлення змінної середовища GOOGLE_APPLICATION_CREDENTIALS
    import os
    os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = key_path

    # Створення об'єкта класу та виклик головного циклу інтерфейсу
    app = SpeechRecognitionApp(root)
    root.mainloop()

if __name__ == "__main__":
    main()

```

Рисунок 3.14 – Функція призначена для запуску усіх процесів

Таким чином, цей код реалізує інтерфейс для запису голосу користувача, передачі звукового сигналу на Google Cloud Speech-to-Text API для розпізнавання та відображення результату у вікні програми.

3.3 Програмне забезпечення що використано для створення ІС

В сучасному інформаційному суспільстві розробка та впровадження інтелектуальних систем обробки мовлення стає невід'ємною частиною технологічного прогресу. Однією з ключових областей в цьому контексті є створення інформаційних систем, здатних розпізнавати та аналізувати мовлення, а також забезпечувати його транскрибацію в текстовий вигляд. В даному контексті, високотехнологічне середовище розробки, таке як Visual Studio, надає важливі можливості для створення та вдосконалення таких систем. Створення інформаційних систем для обробки мовлення відкриває перед нами безліч нових можливостей та викликів. З одного боку, це надає можливість вдосконалення взаємодії між людьми та технологією, роблячи процес комунікації більш ефективним та зручним. З іншого боку, це вимагає від розробників глибокого розуміння сучасних методів обробки мовленнєвих даних та їх імплементації в інформаційних системах [38].

У даному контексті Visual Studio виступає як потужний інструментарій для розробки та впровадження інтелектуальних систем. Завдяки його інтегрованому середовищу розробки, розширеним можливостям відлагодження, підтримці різних мов програмування та інтеграції з хмарними сервісами, розробники мають змогу легко створювати різноманітні додатки для обробки мовлення. У цьому контексті наша увага зосереджена на створенні інформаційної системи, яка забезпечить не лише розпізнавання мовлення, але й його детальний аналіз та транскрибацію [39]. У процесі розгляду цього завдання ми використовуємо потужні можливості Visual Studio для розробки ефективних та інноваційних рішень.

Серед основних переваг використання програмного середовища є:

1. Мови програмування. Visual Studio підтримує різні мови програмування, такі як C#, Python, а також інші. Вибір мови залежить від вашого досвіду та вподобань. Наприклад, якщо ви обираєте Python, то Visual Studio підтримує розширення для роботи з Python, такі як IntelliSense, відладка та інші (рис. 3.15).

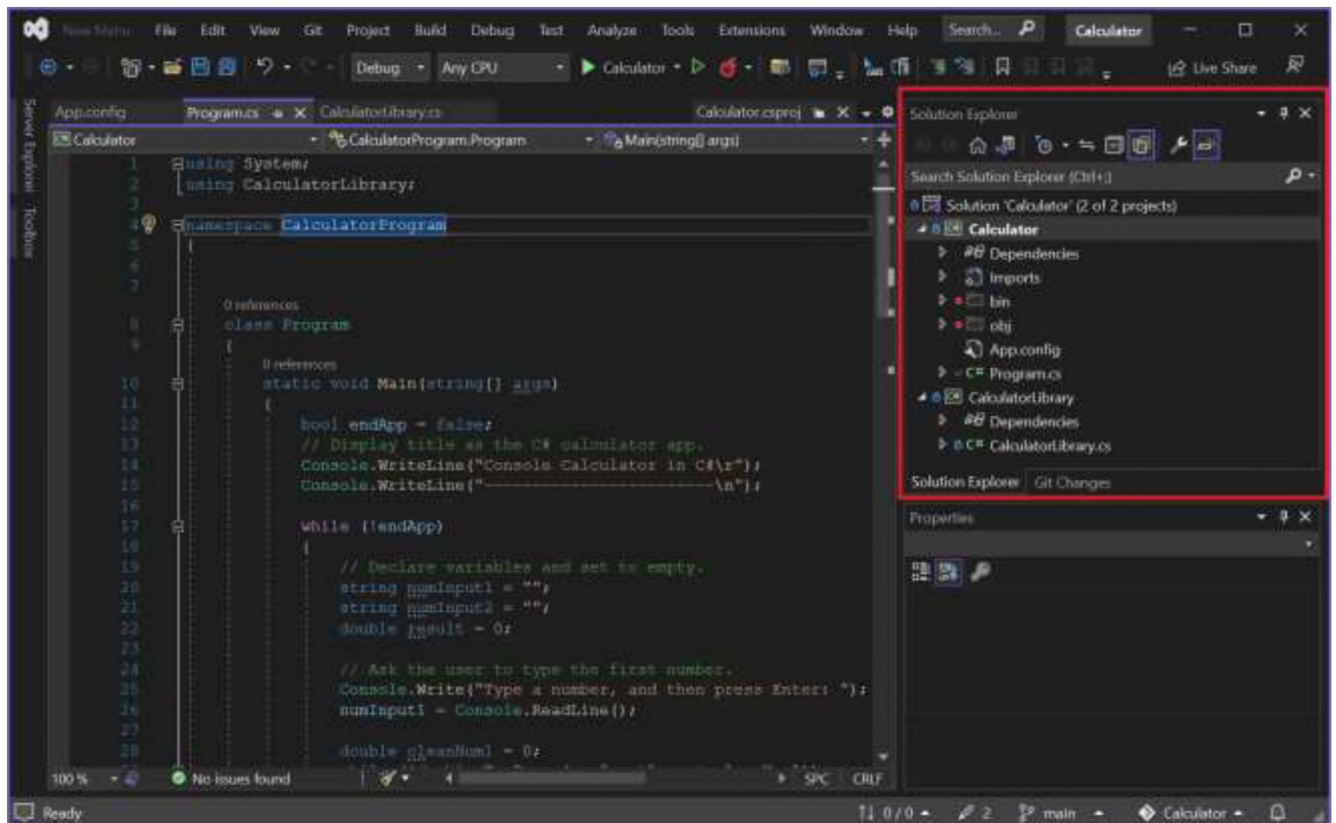


Рисунок 3.15 – Програмне вікно Visual Studio

2. Інтеграція з хмарою. Якщо в задачі поставлено використання хмарних служб для обробки аудіо чи інші послуги штучного інтелекту (AI), Visual Studio добре інтегрується з різними платформами.

3. Інструменти для роботи з мовленням: Visual Studio дозволяє легко інтегрувати різні бібліотеки та інструменти для роботи з аудіо та мовленням. Наприклад, ви можете використовувати бібліотеку Speech SDK для роботи з послугами розпізнавання мовлення від Microsoft.

4. Відладка та тестування: Visual Studio надає розширені засоби відладки та тестування, що полегшує виявлення та усунення помилок у вашому коді.

5. Створення графічного інтерфейсу: Якщо ваша система передбачає наявність графічного інтерфейсу для користувача, Visual Studio має вбудовані інструменти для розробки Windows Forms або WPF додатків.

6. Спільна робота із командою: Visual Studio підтримує спільну роботу над проектом, що є важливим елементом при створенні складних систем.

3.5 Вибір та обґрунтування методики оцінки економічної ефективності

Аналіз основних показників порівняльної ефективності створення та функціонування інформаційних систем включає визначення наступних показників.

1. Річна економія поточних витрат, отримана від функціонування системи.
2. Додаткові капітальні вкладення (КВ), необхідні для створення системи.
3. Термін окупності додаткових КВ.
4. Розрахунковий коефіцієнт ефективності додаткових КВ.
5. Річний економічний ефект.
6. Річна економія трудових витрат на обробку даних у системі.

Оцінимо ці показники.

1. Річна економія поточних витрат складається з прямої економії та інших складових. Пряма економія виникає від автоматизації обробки інформації.

$$\Delta C_T = \Delta C_{\text{п}} + \Delta C_{\text{н}}, \quad (3.1)$$

де $\Delta C_{\text{п}}$ – пряма економія,

$\Delta C_{\text{н}}$ – непряма економія (у цій роботі не розраховується).

Пряма економія, яку отримують від автоматизації обробки інформації розраховується так:

$$\Delta C_{\text{п}} = \Delta C_{\text{б}} - \Delta C_{\text{пор}}, \quad (3.2)$$

де ΔC_b – базовий період, який береться до застосування системи,
 $\Delta C_{\text{пор}}$ – порівнюваний період, коли система працює у режимі автоматизації.
 Розрахунок показника за порівнюваний період містить:

$$\Delta C_{\text{пор}} = C_1 + C_2 + C_3 + C_4 + C_5 + C_{\text{спр}}, \quad (3.3)$$

де C_1 – витрати на оплату праці персоналу;
 C_2 – нарахування на фонд оплати праці (податок 22%);
 C_3 – витрати сировину, матеріали (картриджі, папір, комплектуючі);
 C_4 – амортизація обладнання, зазвичай розглядається як лінійна з терміном служби від 3 до 8 років;
 C_5 – інші витрати (витрати на відрядження, інформаційні витрати, плата за кредит, податки, представницькі витрати).

$C_{\text{спр}}$ – передвиробничі витрати, які потрібно визначити додатково.

Передвиробничі витрати – витрати, які можуть бути направлені на розробку (купівлю) програмних засобів, на навчання фахівців і т.д.

2. Додаткові капітальні вкладення охоплюють будівництво, оренду приміщення, ремонт, придбання мережевого обладнання тощо.

3. Термін окупності капітальних вкладень розраховується за формулою:

$$T = KД / \Delta C_{\text{т}}, \quad (3.4)$$

де $\Delta C_{\text{т}}$ – річна економія поточних витрат,

$KД$ – капітальні вкладення, наведені до 1 року

4. Розрахунковий коефіцієнт ефективності E_p є величина зворотна T ,
 $E_p = 1/T$. $E_n = 0,33$ – нормативний коефіцієнт ефективності.

Якщо розрахунковий коефіцієнт більший чи дорівнює E_n , то проект приймається до впровадження, тобто створення АІС ефективно.

5. Річний економічний ефект розраховується за формулою:

$$E = \Delta C_{\text{т}} - KД * E_n, E_n = 0,15 \quad (3.5)$$

6. Річна економія трудових витрат розраховується як різниця між базовим та порівняльним періодами до та після впровадження системи в режимі автоматизації:

$$\Delta T = \Delta T_{\text{б}} - \Delta T_{\text{пор}}, \quad (3.6)$$

де $\Delta T_{\text{б}}$ – період базовий до застосування системи,
 $\Delta T_{\text{пор}}$ – період, порівнюваний, тобто період роботи системи у режимі автоматизації.

Далі було виконано розрахунок основних економічних показників від впровадження інформаційної системи. Результати розрахунків представлені в табл. 3.1.

Таблиця 3.1. Результати розрахунку базових економічних показників

Затрати	Базовий період ($\Delta C_{\text{б}}$), грн.	Порівнюваний період ($\Delta C_{\text{пор}}$), грн.
C1	450000	250000
C2	185900	135800
C3	750	2250
C4	0	7500
C5	900	500
Спр		12000
Всього	637550	401300

$$\Delta C_{\text{п}} = 637550 - 401300 = 236250 \text{ грн.}$$

$$\Delta C_{\text{т}} = 236250 \text{ грн.}$$

1. Додаткові капітальні вкладення.

$$\text{Оренда приміщення (на рік)} = 4000 \text{ грн} \cdot 30 \text{ м}^2 = 120000 \text{ грн.}$$

Придбання комп'ютера (Комплектація: AMD Phenom II X6 1075, 8Гб, 2000Гб, GeForce GTX 560) = 15000 грн.

Придбання монітора (17» Acer V173Ab чорний 5ms) = 4500 грн.

Придбання принтера Canon LBP-7800 = 4700 грн.

Термін функціонування системи = 10 років.

Сума додаткових капітальних вкладень за рік

$$120000 + (15000 + 4500 + 4700) / 10 = 122420 \text{ грн.}$$

2. Термін окупності капітальних вкладень:

$$T = 122420 / 236250 = 0.51$$

3. Розрахунковий коефіцієнт ефективності E_p .

$$E_p = 1/0,51 = 1,96$$

Проект ефективний.

4. Річний економічний ефект

$$E = 236250 - 122420 \cdot 0,15 = 11383 \text{ грн.}$$

5. Річна економія трудових витрат:

$$\Delta T = 520000 - 240000 = 280000 \text{ грн.}$$

Розрахунок показав, що застосування системи принесе підприємству вигоду, ефект від використання системи становитиме 11383 грн на рік.

Висновки до розділу 3

У розділі, присвяченому створенню програми на базі нейронної мережі з використанням Python, ми розглянули кілька ключових аспектів, що впливають на процес розробки та ефективність моделі.

1. Вибір мови програмування. Зосереджуємось на Python як основній мові програмування. Обговорили його популярність у сфері машинного навчання та глибокого навчання.

2. Важливість бібліотек. Пояснили, як бібліотеки, такі як TensorFlow та PyTorch, допомагають спростити розробку нейронних мереж, надаючи потужні інструменти для створення, навчання та тестування моделей.

3. Архітектура нейронної мережі. Проаналізували вибір архітектури, наголошуючи на важливості вибору оптимального типу мережі (наприклад, конволюційна чи рекурентна) в залежності від конкретної задачі.

4. Підготовка даних. Підкреслили важливість коректної обробки даних перед їх використанням у навчанні. Описали методи чистки та підготовки даних для уникнення проблем під час тренування моделі.

5. Навчання та тестування. Розглянули етапи навчання моделі, включаючи оптимізацію гіперпараметрів. Пояснили, як ефективно тестувати та оцінювати модель для визначення її загальної ефективності.

Цей детальний аналіз надає можливість усвідомлення про ключові аспекти створення програми на базі нейронної мережі з використанням Python, а також підготовлює їх до подальших кроків у розвитку у цій області. Виконано аналіз основних показників порівняльної ефективності створення та функціонування інформаційних систем. В результаті проведених розрахунків економічної ефективності було встановлено, що застосування запропонованої розподіленої системи принесе вигоду, та ефект від використання системи становитиме 11383 грн. на рік.

ВИСНОВКИ

У роботі була поставлена і вирішена актуальна наукова задача розробки та тестування інформаційної системи на основі нейронної мережі призначеної для аналізу мовлення.

У результаті дослідження, проведеного в рамках даної роботи, було виявлено, що використання нейронних мереж для класифікації аудіо контенту є актуальним і перспективним напрямком в ряді сфер, зокрема в розпізнаванні мовлення та освітніх системах.

Розділ 1 присвячений вивченню актуальності цього питання та огляду існуючих методів розпізнавання мовлення на основі нейронних мереж. Аналізуючи сучасні тенденції в освіті, було виявлено, що використання нейронних мереж в даній області може значно поліпшити процеси навчання та оцінювання студентів. Основні висновки з різних аспектів цього розділу включають:

1. Вирізнення корисної інформації з цього обсягу даних вимагає високоточної системи класифікації. У цьому контексті застосування нейронних мереж надає можливість автоматизації цього процесу та покращення точності класифікації.

2. Використання нейронних мереж для розпізнавання мовлення є однією з важливих особливостей та дозволяє автоматизувати процеси, пов'язані з аналізом аудіо контенту. Нейронні мережі, зокрема згорткові та рекурентні мережі, демонструють вражаючі результати в завданнях класифікації голосу та розпізнавання мовлення.

3. Архітектури нейронних мереж для класифікації аудіоданих можуть включати різноманітні шари, такі як персиптрони для обробки аудіосигналів, згорткові шари для виявлення просторових особливостей та рекурентні шари для роботи з послідовністю аудіоданих. Також варто звернути увагу на сучасні архітектури, такі як Transformer, які успішно застосовуються у завданнях обробки послідовностей.

У розділі 2 були розглянуті різні архітектури та моделі нейронних мереж, такі як персиптрони, згорткові та рекурентні мережі, а також архітектури Transformer. Основними висновками з результату роботи є:

1. Досліджено технічне забезпечення, необхідне для роботи нейронної мережі, включаючи центральний процесор (CPU), графічні процесори (GPU), тензорний процесор (TPU), оперативну пам'ять (RAM) та систему охолодження.

2. Архітектури нейронних мереж, різноманітні структури та типи шарів, які взаємодіють для ефективного вирішення конкретних завдань. Зокрема, персиптрони — базові блоки нейронних мереж, що використовуються для обробки вхідних сигналів та класифікації даних. Згорткові нейронні мережі (CNN) спеціалізуються на роботі з великими вхідними образами, виявленні просторових шаблонів та підвищенні роботи зображень. Рекурентні нейронні мережі (RNN) використовуються для роботи з послідовністю даних, що робить їх ефективними у завданнях розпізнавання мовлення та обробки текстів. Також варто відзначити архітектури Transformer, які виявилися дуже ефективними в обробці послідовностей та отримали широке визнання у завданнях машинного перекладу та генерації текстів.

3. Вивчено перспективи розробки нових архітектур і моделей нейронних мереж, що можуть відкрити нові можливості в області класифікації аудіо контенту.

Загальносвітовий інтерес до архітектур та моделей нейронних мереж свідчить про їхню важливість у сучасному світі технологій. Вивчення та подальший розвиток цих аспектів сприяють покращенню роботи систем штучного інтелекту та відкривають нові можливості для застосування глибокого навчання в різних галузях від розпізнавання образів до роботи з природною мовою.

У розділі 3 роботи розглянута створення інформаційної системи (ІС) для аналізу мовлення на основі бібліотеки Google Speech to Text. Досліджено використання мови програмування Python для розробки ІС та визначено програмне забезпечення, використане для цієї мети. Обрано та обґрунтовано методику оцінки економічної ефективності створеної ІС. Серед основних аспектів розділу присвяченого розробці ІС можна виділити:

1. Google Speech to Text є потужним інструментом для розпізнавання мовлення, розробленим Google, який надає можливість перетворення аудіозаписів або розмов у текстовий вигляд. Застосування цієї бібліотеки стає дорогоцінним ресурсом для розробників та інженерів у сферах, де аналіз мовлення є ключовим елементом, таких як розробка голосових асистентів, транскрибація аудіо- та відеозаписів, системи контролю розмов, тощо. Створення ІС для аналізу мовлення на основі цієї бібліотеки включає в себе декілька ключових етапів. Перш за все, визначається область застосування системи, оскільки функціональні можливості можуть бути різними для різних сфер: від транскрибації розмов до створення інтерактивних голосових інтерфейсів. Після цього визначається технічне забезпечення системи, включаючи вибір мов програмування, розробку інтерфейсу користувача та інші технічні аспекти.

2. Мова програмування Python часто обирається для створення системи аналізу мовлення, оскільки вона володіє широким спектром бібліотек та модулів для обробки даних, а також має зручний інтерфейс для взаємодії з бібліотеками обробки мовленнєвих даних, зокрема Google Speech to Text.

3. Програмне забезпечення, використане для створення ІС, може включати в себе не лише код для взаємодії з бібліотекою розпізнавання мовлення, але й інші модулі для обробки та аналізу отриманих результатів. Наявність ефективної системи обробки та візуалізації результатів роботи ІС є ключовим аспектом забезпечення користувача зручністю та високою якістю отриманої інформації.