МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп’ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

МЕЛЯНЧУК Артем Вікторович

Метод розпізнавання голосових команд з допомогою штучних нейронних мереж / Method of voice command recognition using artificial neural networks

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп’ютерні науки

Кваліфікаційна робота

Виконав студент групи
КНм-21
А.В. Мелянчук
_____

Науковий керівник:
д.т.н., доцент В.С. Коваль
_____

Кваліфікаційну роботу
допущено до захисту:
«___» _____ 20___ р.
Завідувач кафедри
_____ М.П. Комар

ТЕРНОПІЛЬ - 2021

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 3
ВСТУП 4
1 АНАЛІЗ ІСНУЮЧИХ РІШЕННЬ РОЗПІЗНАВАННЯ ЗВУКОВИХ ДАНИХ 7
1.1 Актуальність задачі розпізнавання звукових даних 7
1.2 Огляд існуючих підходів рішення задачі 11
1.3 Постановка задачі дослідження 25
2 РОЗРОБЛЕННЯ МЕТОДУ РОЗПІЗНАВАННЯ ГОЛОСОВИХ КОМАНД З ДОПОМОГОЮ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ 26
2.1 Запропонований спосіб розпізнавання голосових команд 26
2.2 Архітектура штучних нейронних мереж для розпізнавання голосових команд 30
2.3 Алгоритм методу розпізнавання голосових команд 45
3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ 47
3.1 Структурна схема системи розпізнавання голосових даних 47
3.2 Опис засобів розробки 50
3.3 Експериментальне дослідження методу розпізнавання голосових команд 65
ВИСНОВКИ 67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 68
ДОДАТКИ ..... **Помилка! Закладку не визначено.**
Додаток А. Публікація №1 V “Науково-практичної конференції "Інтелектуальні комп’ютерні системи та мережі”**Помилка! Закладку не визначено.**
Додаток Б. Публікація №2 V Науково-практичної конференції "Інтелектуальні комп’ютерні системи та мережі”**Помилка! Закладку не визначено.**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ANN — Штучні нейронні мережі.

API — Application Program Interface.

ASR — Система, або системи автоматичного розпізнавання мовлення.

BPTT — Зворотне поширення через час.

CNN — Згорткові нейронні мережі.

CTC — Конекціоністська тимчасова класифікація.

DNN — Глибокі нейронні мережі.

DTW — Генеративна модель розпізнавання образів.

GMM — Модель Суміші Гауса.

gRPC — Google Remote Procedure Call.

GSR API — Google Speech Recognition API.

HMM — Прихована Марківська Модель.

IDE — Інтегроване середовище розробки.

LTSM — Мережі довгої короткочасної пам’яті.

MLP — Багатошаровий перцептрон.

RFB — Мережа радіально базисних функцій.

RTRL — Пряме поширення через час.

SVM — Метод опорних векторів.

VC — Розмірність Вапника-Червоненкіса.

ПК — Персональний комп’ютер.

ВСТУП

Актуальність теми дослідження. Протягом усього свого життя люди спілкувалися переважно за допомогою голосу, оскільки в ранні часи вони освоювали відповідні навички та продовжували покладатися на усне спілкування та розвивати його. Тож таким чином, спілкування за допомогою усної мови набагато ефективніше, аніж за допомогою клавіатури та миші. Розпізнавання голосу та усного мовлення надає методи, за допомогою яких комп'ютери можуть сприймати введення людським голосом замість введення з клавіатури. Це надзвичайно корисно для дітей, людей похилого віку та людей з обмеженими можливостями. Що ж до молоді та дорослих людей — голосовий ввід значно прискорює введення тексту.

Голосові команди як і голосовий ввід тісно зв'язані. У сьогоденні вони є вельми популярними функціями, їх використовують майже у всіх сферах людської діяльності де задіяні комп'ютерні та електронні пристрої. Через їх тісний взаємозв'язок переважну більшість електроніки можна цілком або частково використовувати за допомогою голосових команд.

Сучасні алгоритми та методи розпізнавання голосового вводу та голосових команд знаходяться на досить просунутому рівні, проте навіть так вони не є досконалими, та переважно всі стикаються з декількома загальними проблемами.

Попереднім методом обробки усного мовлення є процес вивчення звукових сигналів і способи обробки цих сигналів. Звичайний метод розпізнавання усного мовлення наполягає на представленні кожного слова його вектором ознак і узгодженням із статистично доступними векторами за допомогою нейронних мереж. На відміну від старомодного методу Прихованої Марківської Моделі, нейронні мережі не вимагають попередніх знань про мовний процес і не потребують статистики мовних даних [1].

На вимову так, а отже і на голосове розпізнавання значною мірою впливає велика кількість факторів серед яких: стать, грубість голосу, його висота, тональність, гучність, акцент, фонові шуми, відлуння, якість мікрофону та інші.

Тому з практичної точки зору розробка нових та покращення існуючих методів розпізнавання команд та голосового вводу за допомогою нейронних мереж

є актуальною та буду залишатися такою до розробки нейронних мереж, або штучного інтелекту, що будуть досконало розуміти усне мовлення не залежно від вищеперерахованих факторів.

З огляду на вище написане, зроблено висновок, що тема кваліфікаційної роботи «Метод розпізнавання голосових команд з допомогою штучних нейронних мереж», є актуальною на момент написання даної роботи.

Мета роботи. В даній кваліфікаційній роботі взято за мету дослідження існуючих систем та методів розпізнавання голосового вводу та команд за допомогою штучних нейронних мереж та створення нового чи покращення функціональних можливостей існуючих методів розпізнавання голосових команд на базі програмно-апаратного підходу, що дозволить проводити розпізнавання голосових команд більш чітко навіть при досить гучних фонових шумах на невисокоякісних мікрофонах. Для досягнення мети кваліфікаційної роботи необхідно:

- розглянути та проаналізувати існуючі системи та їхні методи розпізнавання звукових даних;
- на основі проведеного аналізу виявити проблеми і недоліки розглянутих систем та методів;
- змоделювати можливе рішення знайдених проблем в системах розпізнавання звукових даних;
- зробити висновки та запропонувати метод, що дозволить вирішити, або зменшити негативний вплив на системи знайдених проблем на основі проведених досліджень.

Об’єкт дослідження. Об’єктом дослідження даної кваліфікаційної роботи є системи, методи та процеси розпізнавання звукових даних.

Предмет дослідження. Предметом дослідження є використанням методу шумоусунення на основі прийомів штучних нейронних мереж за допомогою програмно-апаратної складової.

Методи дослідження. Поставлені задачі кваліфікаційної роботи вирішуються шляхом експериментальних та теоретичних досліджень. При проведенні аналізів існуючих рішень розпізнавання голосового вводу та

підвищення коректності їх функціонування було використано модулі програмування та звуко аналізу на мові Python серед яких: PyAudio, librosa, а також системи штучних нейронних мереж для розпізнавання голосового введення та модуль шумоусунення RTX на основі штучних нейронних мереж.

Наукова новизна одержаних результатів. Під час виконання даної кваліфікаційної роботи будо удосконалено коректність розпізнавання людського мовлення за допомогою штучних мереж при наявності гучних фонових шумів.

Практичне значення одержаних результатів. Після розробки методу розпізнавання голосових команд за допомогою штучних нейронних мереж його можна застосовувати для розпізнавання голосового вводу на таких місцях, як наприклад: металургічні заводи, фабрики, будівництва, різноманітні громадські місця та інші сильно зашумлені місця для впровадження голосового керування, де немає можливості встановити шумоізоляцію чи шумопоглинаючі матеріали.

Апробація. Результати роботи знайшли своє відображення у роботі V Науково-практичної конференції інтелектуальних комп’ютерних систем та мереж з участю «МЕТОД РОЗПІЗНАВАННЯ ГОЛОСОВИХ КОМАНД З ДОПОМОГОЮ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ» (Тернопіль, 02 грудня 2021 року).

1 АНАЛІЗ ІСНУЮЧИХ РІШЕННЬ РОЗПІЗНАВАННЯ ЗВУКОВИХ ДАНИХ

1.1 Актуальність задачі розпізнавання звукових даних

Звук — це коливальний рух частинок в середовищі, що поширюються у хвилеподібному вигляді у таких середовищах, як: рідини, тверді тіла чи гази та сприймаються слуховим апаратом. Також під звуком мають на увазі коливання, котрі сприймаються сенсорно-слуховою системою людей та тварин [2]. У такому випадки йдеться про збурення, що з певною частотою поширюються в одному із вищеперерахованих середовищ. Людський слуховий апарат може сприймати ці коливання у відносно невеликому діапазоні частот. Слуховий апарат переважної більшості тварин же – може сприймати звукові коливання в значно обширнішому частотному діапазоні. Також загалом цей термін визначає процес поширення вібрацій у середовищах з різноманітними фізичними властивостями, у котрих силою, що намагається відновити початкове положення частинок, які було збуджено в початковий стан спокою, являється сила пружності. Хвильові збудження, що характеризуються, під звуком є об'єктивно реальними та існують без будь-якої залежності від їхнього сприймання будь-якими живими істотами. Вивчення закономірностей сприйняття, генерації та поширення звукових коливань у різноманітних середовищах належить до галузі наукових знань під назвою акустика.

Майже всі природні явища супроводжуються певними звуками, що розпізнаються та сприймаються слуховими апаратами тварин та людей і слугують для спілкування та орієнтування у просторі [3]. Особливість у сприйнятті звукових коливальних рухів слухом людини поділяє різноманіття звуків на гармонійні (приємні) звуки, такі, як співи пташок, звуки ліри та інші музичні гармонічні звуки, й звуки що мають специфічне наповнення у певному спектрі, зазвичай дратівливі чи небажані, що зазвичай означують, як шум.

Шум або акустичний шум — це коливання частинок довкілля, що сприймається органами слуху людини як небажані сигнали. З точки зору акустики: шум — це нестійкі або випадкові акустичні коливання, що характеризуються випадковою зміною амплітуди і частоти [4].

Шуми класифікують відповідно до джерела їх походження, а саме [5]:

- шуми, що виникають у газовому середовищі називають аеродинамічними;
- шуми, що виникають у рідинному середовищі називають гідродинамічними;
- шуми, котрі виникають в результаті впливу змінних магнітних сил, що викликають небажані збудження електромеханічних елементів у пристроях називають електромагнітними;
- шуми, які виникають в результаті ударів у стикованих місцях деталей, вібрацій поверхні обладнання чи машин, або конструкцій загалом називають механічними.

Також шуми класифікують за діапазоном частот:

- низькочастотний до 400 Герц;
- средньочастотний від 400 до 1000 Герц;
- високочастотний понад 1000 Герц.

У вузькоспеціалізованих галузях таких, як електроніка та акустика існують поняття про кольори шумів, за яким певний колір присвоюється шумовому сигналу в залежності від його статистично-спектральних властивостей. Однією з є спектральна густина, що характеризує розподіл потужності за частотами. В акустиці прийнято загалом поділяти спектри шумів за наступними кольорами: білий шум, рожевий шум, червоний (коричневий) шум та сірий шум. Інколи виділяють й інші різновиди [5].

Білий шум — постійний шум, спектральні складові якого рівномірно розподілені по всьому діапазону частот (рисунок 1.1). проте білим шумом можна вважати будь-який шум, спектральна щільність якого однакова (або майже однакова) у даному діапазоні частот. Назву одержав від білого світла, яке включає електромагнітні хвилі частот усього видимого діапазону електромагнітного випромінювання [6].

Рожевий шум — шум, спектральна густина якого змінюється з частотою f за законом 1/f (рисунок 1.2). Іноді рожевим шумом називають будь-який шум, спектральна густина якого зменшується зі зменшенням частоти [7].

Червоний шум (броунівський шум) — шум також відомий як броунівський шум, тому що його зміна звукового сигналу від однієї миті до іншої є випадковою. Спектральна густина червоного шуму пропорційна 1/f2, де f — частота (рисунок 1.3) [8].

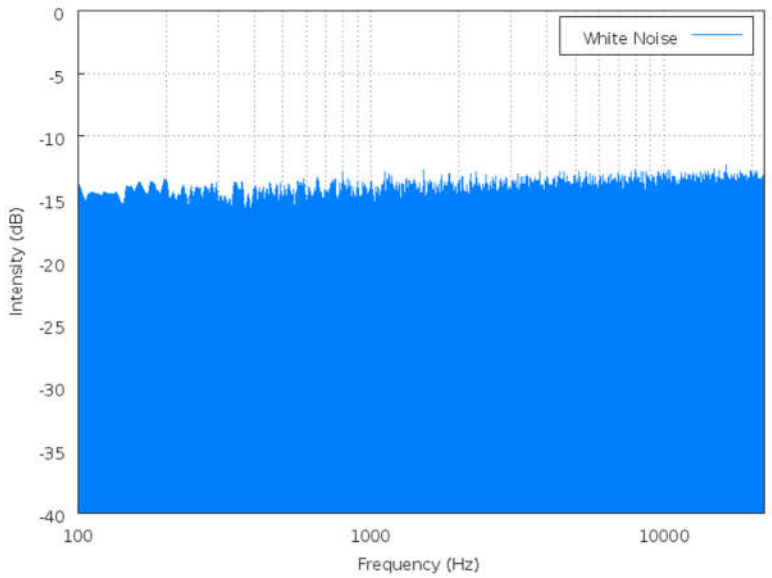

Рисунок 1.1 — Спектр білого шуму [6]

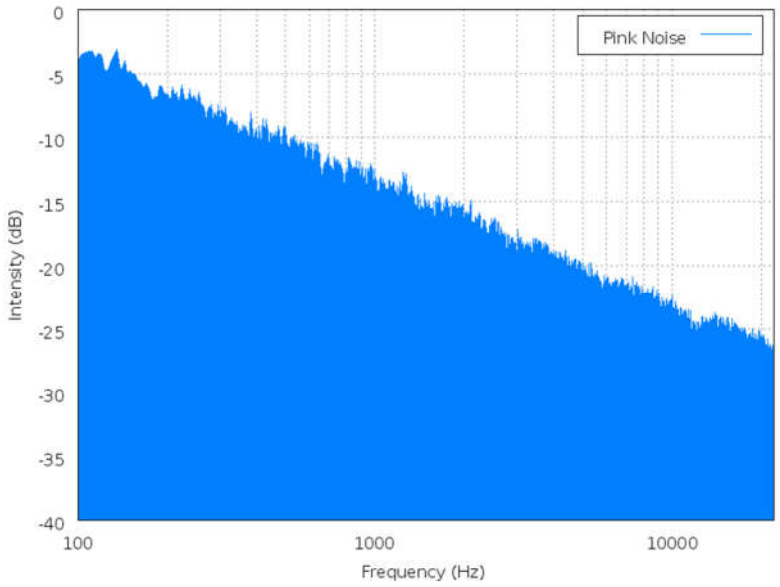

Рисунок 1.2 — Спектр рожевого шуму

Сірий шум — шумовий сигнал, відповідний акустичній кривій сталої гучності на всіх частотах, тобто для людського вуха він має однакову гучність на всіх частотах (рисунок 1.4) [9].

Також на розпізнавання звукових даних впливають такі явища, як: ревербація (утворюється зазвичай у великих приміщення та характеризується багаторазовим відбиттям звуків від стін), дифракція (зумовлюється накладанням звукових хвиль одна на одну при обминанні перешкод), рефракція та іншими явищами, котрі менш впливають на розпізнавання звуків в області розроблюваного методу за рахунок умов та середовищ у яких вони виникають.

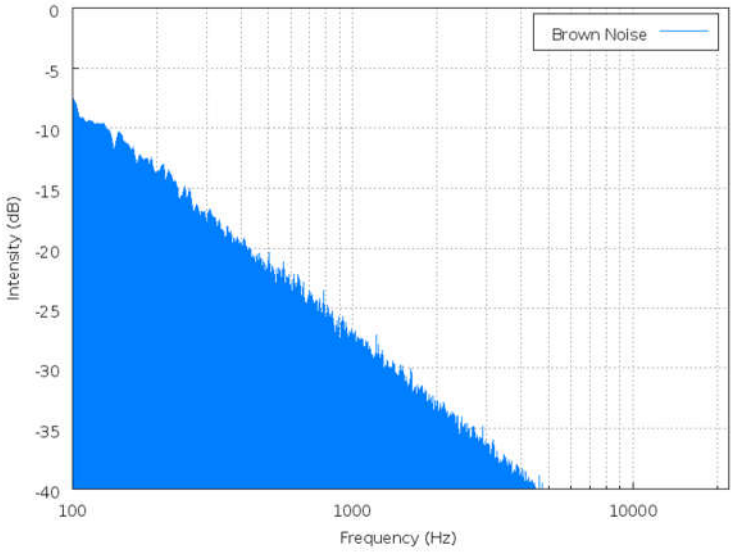

Рисунок 1.3 — Спектр червоного шуму

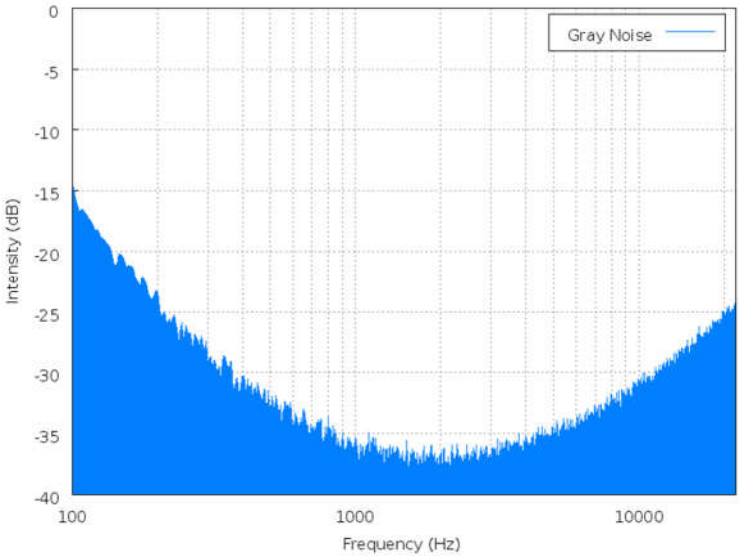

Рисунок 1.4 — Спектр сірого шуму

Попри те, що розпізнавання звуків та людського мовлення все більше інтегрується в нашу повсякденність, та все більше розвиваються, що значно

спрощує: спілкування введенням тексту голосом, пошук інформації за звуковим зразком, управління величезною кількістю різних пристроїв та навіть просто задля розваг. На жаль, коректність розпізнавання людського мовлення мало того, що є вельми ресурсномістким завданням воно також вимагає достатньо чіткого та чистого людського голосу, що являється досить складно виконуваною умовою зважаючи на всі шуми, що постійно оточують нас. Попри це, кінцеві користувачі систем розпізнавання звукових даних та голосового вводу очікують високої точності у розпізнаванні команд та відповідної реакції системи на віддані команди.

Але сукупність раніше згаданих явищ та шумів, навіть кожен окремо значною мірою ускладнюють завдання з розпізнавання мовлення людини. Виходячи з цього можна зробити висновок, що: «Розробка методу розпізнавання голосових команд за допомогою штучних нейронних мереж з усуненням шумів» є актуальною на даний момент.

1.2 Огляд існуючих підходів рішення задачі

Методи розпізнавання мовлення: які можуть використовувати ASR можна розділити на такі категорії:

- Виокремлене слово - вимагає, щоб кожне висловлювання було чітким та з обох боків відрізка зразка була тиша. Одночасно приймаються лише окремі слова та окремі висловлювання.
- Безперервні слова - засоби безперервного розпізнавання мовлення надають користувачам можливість безперервно говорити та майже природно, паралельно з цим комп'ютер визначає зміст мовлення. ASR, які передають засоби безперервного мовлення, досить важко створити, оскільки вони вимагають деяких особливих і своєрідних методів для визначення меж висловлювань.
- Сполучені слова - дуже схожі на виокремлені слова, але вони дозволяють розпізнавати окремі висловлювання з «мінімальними паузами» між ними.

– Спонтанна мова - на елементарному рівні спонтанне мовлення можна розглядати як природне мовлення, а не спеціально підготоване для дослідження. Автоматичний засіб розпізнавання мовлення повинен мати можливість обробляти широкий спектр мовленнєвих функцій, як-от слова, що вимовляються разом.

– Класифікація звуків мовлення зазвичай здійснюється на основі двох процесів, заснованих на тому, як розглядається процес класифікації:

– На основі процесу обструкції (звуків з урахуванням звуків перешкод та без них). Процес класифікації звуків по відношенню до процесу обструкції ґрунтується на уявленні про рух звуку у повітрі. Утворюючи звуки повітря виходячи з людини, може бути у двох станах; воно відбивається в роті чи горлі або ні. Відповідно, звуки, які виробляються в результаті обструкції, не є однаковими. Наприклад, усі голосні є звуками мовлення без перешкод, а всі приголосні — звуки мовлення, що виникають у результаті перешкод.

– На основі процесу голосних та глухих звуків. Голосні звуки утворюються, коли голосові зв'язки вібрують та виробляється звук. Тоді, як в глухих звуках вібрація голосових зв’язок не виникає. Усі голосні дзвінкі, тоді, як деякі приголосні на стільки ж дзвінкі, як і беззвучні.

– Даний підрозділ ілюструє різні методи розпізнавання мовлення, а саме кілька основних підходів до розпізнавання мовлення:

– акустично-фонетичний підхід;

– підхід розпізнавання образів;

– підхід штучного інтелекту;

– підхід нейронних мереж;

– підхід на основі Прихованої Марківської Моделі (HMM) та підхід на основі Моделі Гаусової Суміші (GMM).

1.2.1 Акустично-фонетичний підхід

Акустично-фонетичний підхід (рисунок 1.5), або акустичне моделювання використовується дослідниками для розуміння фонетичних правил і використання цих правил у ASR та є життєво важливими аспектами сучасних статистично заснованих ASR. Окрім розробки нових наборів ознак та методів моделювання,

акустично-фонетичний підхід зосереджується на розумінні фонетичних одиниць та їх взаємозв’язків у різних контекстах їх використання. З його допомогою дослідники прагнуть зрозуміти фонетичні правила для розробки систем розпізнавання мовлення у відповідних областях, таких, як: класифікація акцентів, виявлення мовленнєвої активності та виділення вокальної мелодії. Сама акустична фонетика займається передачею звуків від мовця до слухача. Такий підхід дає можливість вивчити природу мовного сигналу для різних звуків незалежно від ознак, що представляють ці звуки. Цей спосіб дає можливість аналізувати та розуміти природу різних звуків, таких як голосні, напівголосні, дифтонги тощо. З використанням цього підходу зазвичай дослідники з усього світу розробляють системи розпізнавання для різних мов. Сучасні методи моделювання, що використовуються в області розпізнавання мовлення, добре підходять для мов з хорошими ресурсами. Швидкість розпізнавання і точність цих систем залежать від обсягу даних, які використовуються для навчання систем. Ці системи не підходять для мов з обмеженими ресурсами, де мовленнєвий матеріал недоступний, і для тональних мов, де важливий високий акцент. Для розробки систем розпізнавання мов різних регіонів світу не можна використовувати загальні рамки або правила, необхідно розуміти лінгвістичні особливості певної мови під яку проводиться розробка. Акустично-фонетичний підхід використовується дослідниками для вирішення цих питань та вирішення різних проблем області розпізнавання мовлення. Найчастіше цей підхід використовують у багатомовному розпізнаванню мовлення та класифікації наголосу.

Цей підхід допомагає дослідникам з усього світу проводити аналіз фонетичних одиниць, таких, як: алофони та морфеми, що характерні для конкретної мови, щоб отримати певні підказки, котрі допоможуть підвищити точність систем ASR. У деяких мовах на вимову фонем сильно впливає їхній контекст. У цих ситуаціях лише акустичне моделювання не може розрізняти різні звуки через відсутність різниці у вимові в усному мовленні.

Рисунок 1.5— Схема акустично-фонетичного методу

1.2.2 Підхід розпізнавання образів

Існує два основних етапи підходу до розпізнавання шаблонів: навчання шаблонів і порівняння шаблонів. Використання добре сформульованої математичної основи та ініціювання послідовного представлення мовних шаблонів для надійного порівняння шаблонів. Набір позначених навчальних вибірок за допомогою формального алгоритму навчання є важливою ознакою цього підходу. При цьому існує два методи: підхід на основі шаблонів і стохастичний підхід. Це більш підходящий підхід до розпізнавання мовлення, оскільки він використовує ймовірнісні моделі для роботи з невизначеною або неповною інформацією. У цьому підході існує багато методів, таких як HMM, SVM, DTW, VQ тощо, серед цих прихованих моделей Маркова є найпопулярнішим стохастичним підходом сьогодні. Типова схема функціонування такого підходу зображена нижче на рисунку 1.6.

VQ (Векторне квантування): - Цей метод попередньо заснований на принципі блочного кодування. Цей метод дозволяє моделювати функції щільності ймовірності шляхом циркуляції векторів-прототипів. Раніше він використовувався для стиснення даних. Він виконує відображення вектора з великого векторного простору на скінченну кількість областей у цьому просторі. Кожен регіон відомий як кластер і може бути зображений його центром, який називається кодовим словом. Векторне квантування використовується для стиснення даних із втратами, корекції даних із втратами, а також для кластеризації та розпізнавання образів.

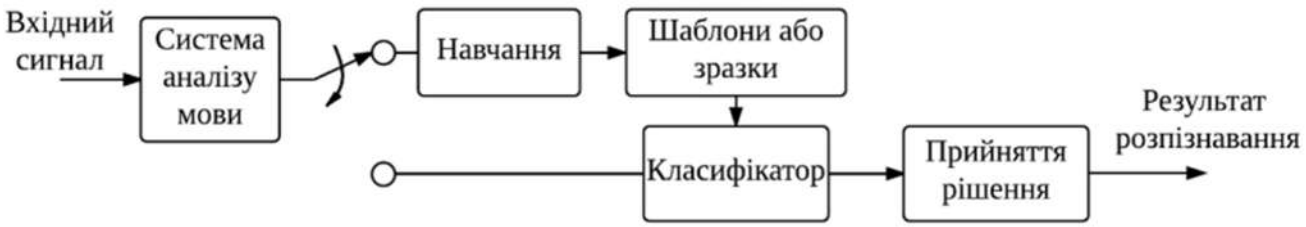

Рисунок 1.6 — Схема підходу розпізнаванням образів

Метод опорних векторів (SVM). Є одним із найпотужніших інструментів для розпізнавання образів, котрий використовує дискримінаційний підхід. Він використовує лінійні та нелінійні розділові гіперплощини для класифікації даних. Незважаючи на це, SVM можуть класифікувати лише вектори даних фіксованої довжини. Цю процедуру може бути нелегко застосувати до завдань, що включають класифікацію даних змінної довжини. Ці дані мають бути перетворені у вектори фіксованої довжини, перш ніж SVM можна буде використовувати в системі автоматичного розпізнавання мовлення в реальному часі. Це узагальнений лінійний класифікатор з функціями підгонки максимальною маржею. Ця функція дає регуляризацію, яка допомагає краще узагальнювати класифікатор. Традиційні статистичні та нейромережні методи контролюють складність моделі за допомогою невеликої кількості функцій. SVM обмежує складність моделі, керуючи розмірами VC своєї моделі. Цей метод не залежить від розмірності і може використовувати простори дуже великих розмірів, що забезпечує відповідний вибір функцій під час навчання.

Метод динамічного викривлення часу (DTW) — це один з алгоритмів для вимірювання подібності двох тимчасових послідовностей, які можуть відрізнятися за швидкістю. За допомогою DTW можна виявити схожість у ходьбі, навіть якщо одна людина йшла швидше за іншу, або якщо під час спостереження спостерігалися прискорення та уповільнення. DTW був застосований до тимчасових послідовностей відео, аудіо та графічних даних — насправді, будь-які дані, які можна перетворити на лінійну послідовність, можна проаналізувати за допомогою DTW. Добре відомим додатком було автоматичне розпізнавання мовлення, щоб впоратися з різними швидкостями мовлення. Інші програми розроблені за

допомогою цього алгоритму включають розпізнавання мовця та онлайн-розпізнавання підпису. Його також можна використовувати в програмах часткової відповідності форми [10].

Загалом, DTW — це метод, який обчислює оптимальну відповідність між двома заданими послідовностями (наприклад, часовими рядами) з певними обмеженнями та правилами [10]:

- Кожному індексу з першої послідовності має відповідати один або більше індексів з іншої послідовності, і навпаки;
- Перший індекс з першої послідовності повинен відповідати першому індексу з іншої послідовності (але це не повинно бути єдиним збігом);
- Останній індекс з першої послідовності має відповідати останнім індексам з іншої послідовності (але це не обов'язково має бути єдиним збігом);
- Відображення індексів з першої послідовності в індекси з іншої послідовності має бути монотонно зростаючим, і навпаки, тобто якщо $j > i$ є індексами з першої послідовності, то в іншій послідовності не повинно бути двох індексів $l > k$, так що індекс i збігається з індексом l, а індекс j збігається з індексом k, і навпаки

1.2.2.1 Підхід на основі шаблону

У підходах зіставлення на основі шаблонів невідоме мовлення порівнюється з набором попередньо записаних слів, щоб знайти найкращу відповідність, що є досить продуктивним підходом для пошуку точних моделей слів. Проте він також має значний недолік, що попередньо записані шаблони фіксуються. Через це варіації в мовленні можна моделювати лише за допомогою великої кількості шаблонів для кожного слова, що з часом стає непрактичним. У цьому підході шаблони зазвичай складаються з репрезентативних послідовностей векторів ознак для відповідних слів. Основна мета тут — вирівняти висловлювання з кожним із шаблонних слів, а потім вибрати слово або послідовність слів, які містять ідеальну відповідність. Для кожного висловлювання відповідність між шаблоном і векторами спостережуваних ознак обчислюється за допомогою різних вимірювань відстані, і ці локальні відстані компілюються вздовж кожного можливого шляху

вирівнювання. Тоді шлях із найнижчим балом визначає оптимальне вирівнювання для слова, а шаблон слова, який отримує найнижчий загальний бал, інтерпретує розпізнане слово або послідовність слів.

1.2.2.2 Статистичний підхід

У статистичних підходах відхилення в мовленні моделюються статистично. В ньому використовується автоматична процедура статистичного навчання, як правило, HMM. Цей підхід відображає сучасний стан ANN. Основним недоліком статистичних моделей є те, що вони повинні приймати попередні припущення моделювання, які можуть ввести в оману, погіршуючи продуктивність системи. В останні роки з'явилася нова техніка, що кидає виклик проблемі розпізнавання розмовної мови. Очікується, що це подолає деякі фундаментальні обмеження традиційної Прихованої Моделі Маркова (HMM). Цей новий підхід є радикальним відходом від поточних підходів статистичного моделювання на основі HMM. Замість використання великої кількості неструктурованих компонентів Гаусової Суміші (GMM) для врахування величезної різниці в спостережуваних акустичних даних висококоартикулованого природного мовлення. Нова модель мовлення, яка була розроблена, щоб забезпечити багату структуру для частково спостережуваної динаміки в області резонансів вокального тракту.

1.2.3 Підхід штучного інтелекту (підхід на основі знань)

Це поєднання акустичного фонетичного підходу та підходу до розпізнавання образів. Підхід штучного інтелекту намагається механізувати процедуру розпізнавання відповідно до того, як людина застосовує свій інтелект у візуалізації, аналізі та, нарешті, прийнятті рішення щодо виміряних акустичних характеристик. У цьому підході широко використовуються системи високого рівня майстерності. У підходах, заснованих на знаннях: знання експертів про варіації мовлення вручну закодовані в систему. Це корисно для явного моделювання варіацій у мовленні; але, на жаль, такі експертні знання важко отримати та успішно використати. Таким чином, цей підхід було визнано непрактичним, і замість нього було використано автоматичну процедуру навчання.

1.2.4 Підхід ANN

Підхід штучного інтелекту намагався механізувати процедуру розпізнавання відповідно до того, як людина застосовує свій інтелект у візуалізації, аналізі та, нарешті, прийнятті рішення щодо виміряних акустичних характеристик. Серед прийомів, що використовувалися в цьому класі методів, було використання експертної системи, яка об’єднує:

Лексичні, фонематичні, семантичні, синтаксичні та прагматичні знання для сегментації та маркування. Вона використовує такі інструменти, як ANN для вивчення взаємозв’язків між фонетичними подіями. В основному цей підхід зосереджений на репрезентації знань та інтеграції джерел знань [11].

Традиційно нейронні мережі називають нейронами або ланцюгами. В даний час термін нейронні мережі називають штучною нейронною мережею, що складається зі штучних нейронів або вузлів. Це мережа елементарних елементів, відомих як штучні нейрони, яка отримує вхід, змінює стан відповідно до цього входу та генерує вихід. Взаємопов'язана група природних або штучних нейронів використовує математичну модель для обробки інформації на основі коннекціоністського підходу до спілкування. Нейтральні мережі можна розглядати як прості математичні моделі, що визначають функцію *f: X→Y*, або розподіл за *X*, або обидва *X і Y*, але у багатьох випадках моделі тісно пов’язані з певним правилом навчання [12].

Термін «штучна нейронна мережа» відноситься до взаємозв’язків між нейронами в різних шарах кожної системи. Математично функція мережі нейрона *f(x)* визначається як композиція інших функцій $g_i(x)$, які далі можна визначити як композицію інших функцій. Найбільш часто використовуваним типом композиції є нелінійна зважена сума, де $f(x) = K\,(\sum_i w_i g_i(x))$, де *K* — попередньо визначена функція. Посилання на колекцію функцій g_i, як вектор g просто було б більш вигідним. Перший вигляд є функціональним видом; даний x потім змінюється на тривимірний вектор h, який потім остаточно перетворюється на *f*. Такий погляд часто зустрічається в контексті оптимізації. Імовірнісний погляд — це другий погляд, і довільна змінна $G = g\,(H)$ залежить від $H = h\,(x)$, яка, у свою чергу, залежить від *X* (випадкова величина). У ракурсі графічних моделей такий погляд

зустрічається найчастіше. Мережі, описані вище, часто називають прямим зв'язком, оскільки його графік є DAG (спрямований ациклічний графік). Мережі, у яких є цикли, називаються рекурентними нейронними мережами.

Мовне моделювання та акустичне моделювання є життєво важливими аспектами сучасних статистичних систем розпізнавання мовлення. Глибокі нейронні мережі (приховані моделі Маркова (HMM)) — це генеративна модель, в якій передбачається, що спостережувані акустичні характеристики генеруються з прихованого процесу Маркова, який переходить між станами *S = {s1, s2 ...., sk}*.

1.2.5 Підхід на основі HMM

HMM — це набір кінцевих станів, де вона вивчає приховані або неспостережувані стани і дає ймовірність спостережуваних станів. Поточний стан завжди залежить від безпосереднього попереднього стану. У прихованій моделі Маркова стан не видно спостерігачеві (приховані стани), тоді як стани спостереження, які залежать від прихованих станів, є видимими. HMM пояснює ймовірність спостережуваного стану або змінної, вивчаючи приховані або неспостережувані стани. Розпізнавання мовлення в основному використовує акустичну модель, яка є моделлю HMM. Це традиційний метод розпізнавання мови та виведення тексту за допомогою фонем. У розпізнаванні мовлення приховані стани — це фонеми, тоді як спостережувані стани — мова або звуковий сигнал. HMM широко використовуються в областях, де приховані змінні керують спостережуваними змінними. Розпізнавання мовлення, розпізнавання зображень, розпізнавання жестів, розпізнавання рукописного тексту, тегування частин мови, аналіз часових рядів є деякими з програм HMM.

HMM був найбільш широко використовуваним методом для розпізнавання мовлення з великим словником (LSVR) протягом щонайменше двох десятиліть. Вирішальними параметрами в HMM є початковий розподіл ймовірності стану $f = \{p\ (q_t = s_i)\}$, де q_t– стан в момент часу *t*; ймовірності переходу $a_{ij} = p\left(q_t = s_j \mid q_{t1} = s_i\right)$; і модель для оцінки ймовірностей спостереження $p\ (x_t\ |s_i)$. Звичайна модель HMM, яка використовується в процесі автоматичного розпізнавання мовлення (ASR), мала ймовірність спостереження, змодельовану за

допомогою моделі Гаусової суміші (GMM). Навіть GMM мав величезну кількість переваг, проблема полягала в тому, що вони були статистично неспроможними для моделювання даних, які лежать на нелінійному різноманітному просторі або поблизу нього. Наприклад, моделювання точок, що знаходяться дуже близько до поверхні сфери, навряд чи вимагає якогось параметра з використанням відповідного класу моделі, але вимагає великої кількості діагональних Гауссів або досить величезної кількості повноковаріаційних Гауссів. Через це інші типи моделей можуть працювати краще, ніж GMM, з використанням інформації, вбудованої у велике вікно фреймів. З іншого боку, ANN (штучна нейронна мережа) може більш ефективно обробляти дані, що знаходяться на поблизу нелінійнійної моделі або на ній, а також вивчати в рази кращі моделі даних. З іншого боку, ANN (штучна нейронна мережа) може більш ефективно обробляти дані, що знаходяться на нелінійній моделі або поблизу неї, і вивчати набагато кращі моделі даних. За останні роки було досягнуто значних успіхів, як в алгоритмах машинного навчання, так і в комп'ютерному обладнанні, що в кінцевому підсумку призвело до більш ефективних методів навчання, котрі мають багато шарів і великий вихідний шар. Вихідний шар повинен містити велику кількість станів HMM, які виникають на кожній фонемі, що моделюється різною кількістю трифонів. Використовуючи різні нові методи навчання, велика кількість дослідницьких груп показала, що глибока нейронна мережа (DNN) має кращу продуктивність, ніж GMM, при акустичному моделюванні для розпізнавання мовлення, яке включає величезний словниковий запас і гігантський набір даних. Штучна нейронна мережа має більше одного шару прихованих одиниць між входами та виходами, тоді як глибока нейронна мережа є мережею прямого зв'язку. Кожна прихована одиниця в DNN, j, використовує логістичну функцію, щоб відобразити всі свої вхідні дані з нижнього шару, x_j, у скалярний стан, y_j, який він надсилає до верхнього лежачого шару.

$$Y_j = \log(x_j) = \frac{1}{(1 + e^{-x_j})}$$

$$X_j = b_j + \sum_i y_j w_{ij}$$

Де b_j – зміщення одиниці j, i – індекс одиниць у нижньому шарі, а w_{ij} – вага з'єднання з блоком j, від блоку i на шарі нижче. Для багатокласової класифікації вихідний блок j перетворює свій загальний вхід, x_j , у ймовірність класу p_j, використовуючи використовуючи м'яку максимальну нелінійність:

$$p_j = \exp(x_j) \ / \ \sum_k \exp(x_k)$$

де k — індекс для всіх класів.

DNN глибокої нейронної мережі можна навчити шляхом зворотного поширення похідних функції вартості, яка вимірює розбіжність між цільовими і фактичними результатами. При роботі з функцією м'якої максимальності натуральну вартість C можна розрахувати, як:

$$\mathrm{C} = -\sum_j d_j \log p_j$$

де p - вихід м'якого максимуму, а d представляє цільові ймовірності.

DNN, що мають багато прихованих шарів, важко оптимізувати. Оптимальний спосіб - не вибирати градієнтний спуск з довільної початкової точки поблизу початку координат, щоб знайти гарний набір ваг, і якщо початкові масштаби ваги не будуть ретельно вибрані, в різних шарах будуть різні величини спинки. поширені градієнти. На додаток до цих проблем DNN може погано виконувати узагальнення тестових даних. Шари DNN є досить гнучкими і кожен має велику кількість параметрів.

В результаті — це робить DNN здатними моделювати дуже складні та нелінійні зв'язки між входами та виходами. Існує ймовірність надмірної підгонки, цю проблему можна усунути шляхом дострокової зупинки або зменшення ваг, але

це можливо лише шляхом значного зменшення потужності. Велика кількість набору даних може одночасно зменшити ймовірність надмірної підгонки і зберегти потужність моделювання, але це збільшує обчислювальний час. Тому існує явна потреба використання інформації в процесі навчального набору для побудови різних шарів нелінійних детекторів ознак.

За типами розділяють такі моделі Маркова:

- Залежні від спікера;
- Незалежні від спікера;
- Розпізнавачі одного слова;
- Безперервні розпізнавання слів.

Котрі в свою чергу поділяють на:

- Вилучення ознак;
- Узгодження ознак;
- Парування слово-фонема;

1.2.5.1 Вилучення ознак

Вхід - це мовний або аудіосигнал в аналоговій формі, де система не може зрозуміти аналоговий сигнал. Працює тільки в цифровому форматі. Тому аудіосигнал необхідно конвертувати в цифровий формат. Цей процес відомий як вилучення ознак.

У функції вилучення сигналу в першу чергу сигнал поділяється на невеликі періоди, наприклад, 10 мс. Кожна частина мовного сигналу називається кадром. Кожен кадр перетворюється з часової області в частотну за допомогою перетворення Фур’є. Результатом вилучення ознак є вектор ознак. Він передає амплітуду всього звукового сигналу у векторний формат, який є цифровим.

Для вилучення ознак використовуються три наступні методи:

- Мелчастотні кепстральні коефіцієнти (MFCC);
- Лінійно-передбачуване кодування (LPC);
- Короткочасне перетворення Фур'є (STFT).

Мелчастотні кепстральні коефіцієнти (MFCC): Перетворення Фур'є заданого сигналу відображаються в шкалі Мел (нелінійна частотна шкала). Він приймає

логарифми в кожній шкалі Мела. Дискретне косинусне перетворення (DCT) виконується після взяття логарифмів. Це являє собою амплітуду мовного сигналу в кожному спектрі. Широко використовуваним методом вилучення ознак є MFCC.

Лінійне передбачуване кодування (LPC): Цей процес ділить весь звуковий сигнал на секунди. Кожна секунда розділена на 30-50 кадрів. І витягує з нього вектор ознак. Він фактично обчислює потужність спектру.

Короткочасне перетворення Фур'є (STFT): ділить мовний сигнал на менші частини і обчислює перетворення Фур’є для кожної, що дає спектр Фур’є кожного сегмента мовного сигналу. Таким чином, він витягує вектор характеристик із заданого звукового сигналу.

1.2.5.2 Узгодження ознак

Узгодження ознак — це процес навчання акустичної моделі з витягнутим вектором ознак. Він дає зв’язок між вектором ознак і фонемами. Фонеми - це окремі звукові одиниці, за якими можна відрізнити одне слово від іншого. Модель Гаусової суміші (GMM) використовується в основному для ущгодження ознак.

GMM використовується, як класифікатор для порівняння об’єктів, витягнутих із вектора ознак із збереженими шаблонами. Модель Гаусової суміші (GMM) — це параметрична функція щільності ймовірності, представлена як зважена сума густин Гаусових компонентів. GMM зазвичай використовуються як параметрична модель розподілу ймовірностей безперервних вимірювань або ознак у біометричній системі, наприклад спектральних характеристик, пов’язаних з голосовим трактом, у системі розпізнавання мовця. Параметри GMM оцінюються на основі даних навчання за допомогою ітераційного алгоритму максимізації очікування (EM).

1.2.5.3 Сполучення слово-фонема

Модель Гаусової суміші (GMM) класифікує фонеми за вектором ознак під час навчання моделі, тоді як під час тестування вона дотримується зворотної процедури. Він узгоджує вектор ознак з попередньо навченими фонемами, розпізнає послідовність фонем і видає розпізнане слово у текстовому форматі.

Три основні проблеми HMM:

1. Першою проблемою являється проблема оцінки, а саме: обчислення ймовірності того, що спостережувана послідовність була створена за моделлю. Ми повинні вибрати той, з максимальною ймовірністю дасть кращий результат. Алгоритм Вітербі використовується для цієї задачі оцінки, але також використовується прямий алгоритм.

2. Другою проблемою являється: визначення прихованого стану (декодування), а саме вибір відповідної послідовності станів є цілком оптимальним. Немає правильного рішення для виявлення прихованої частини проблеми. Використання оптимального критерію є найкращим можливим рішенням. Рішенням задачі декодування є також алгоритм Вітербі.

3. Третьою ж — є навчання: налаштування параметрів моделі для максимізації ймовірності, щоб описати, як виходить дана послідовність спостереження. Послідовність спостереження використовується для навчання HMM. Навчання дозволяє нам налаштувати параметр моделі, щоб створити найкращу модель для заданої послідовності навчання. Цю проблему можна виправити за допомогою алгоритму «Вперед-Назад».

В заключенні про Модель Маркова можна сказати, що вона є важливим статистичним інструментом для моделювання даних з послідовними кореляціями в сусідніх вибірках, таких як дані часових рядів. Та являється одним з найуспішніших методів у обробці природної мови (NLP).

1.2.5.4Алгоритм Вітербі — це алгоритм динамічного програмування для отримання максимальної апостеріорної оцінки ймовірності найімовірнішої послідовності прихованих станів, що називається шляхом Вітербі, що призводить до послідовності спостережуваних подій, особливо в контексті джерел інформації Маркова та HMM. Алгоритм знайшов універсальне застосування при декодуванні згорткових кодів, що використовуються як у цифрових стільникових CDMA, так і в GSM, модемах комутованого зв'язку, супутниковому зв'язку, зв'язку далекого космосу та бездротових локальних мережах 802.11. Зараз він також широко використовується в розпізнаванні мовлення, синтезі мовлення, діаризації,

визначення ключових слів, комп'ютерній лінгвістиці та біоінформатиці. Наприклад, у мові в текст (розпізнавання мовлення) акустичний сигнал розглядається як спостережувана послідовність подій, а рядок тексту вважається «прихованою причиною» акустичного сигналу. Алгоритм Вітербі знаходить найбільш вірогідний рядок тексту за акустичним сигналом [13].

1.3 Постановка задачі дослідження

Результат проведених дослідженнь дає можливість стверджувати, що завдання розробки методів з розпізнавання голосових команд за допомогою штучних нейронних мереж, котрий покращив би існуючі системи розпізнавання голосового вводу в цілому, на даний момент є актуальним. Одним із способів вирішення поставленого завдання являється реалізація методу розпізнавання голосових команд за допомогою використання прийомів усунення шумів на основі штучних нейронних мереж, що збільшить коректність розпізнавання голосових команд існуючими системами.

Отже, в результаті проведення оцінки актуальності розробки методів та порівняння існуючих аналогів може виникнути питання, чого не вистачає в існуючих підходах та системах, та що можливо зробити, щоб підвищити ефективність розпізнавання команд?

Оскільки неможливо втрутитися в алгоритми роботи існуючих ASR, а розробка своєї є практично не доцільною через необхідність великих об'ємів даних для навчання та таких же потужностей для забезпечення навчання систем. Задачею даного дослідження буде розробка методу з використанням існуючих ASR рішень для покращення коректності їхньої роботи.

2 РОЗРОБЛЕННЯ МЕТОДУ РОЗПІЗНАВАННЯ ГОЛОСОВИХ КОМАНД З ДОПОМОГОЮ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

2.1 Запропонований спосіб розпізнавання голосових команд

Розпізнавання мовлення — справді важкий і виснажливий процес. В основному його розділяють на 5 наступних кроків (рисунок 2.1):

1. Мовлення;
2. Попередня обробка мовлення;
3. Вилучення ознак;
4. Класифікація мовлення;
5. Розпізнавання

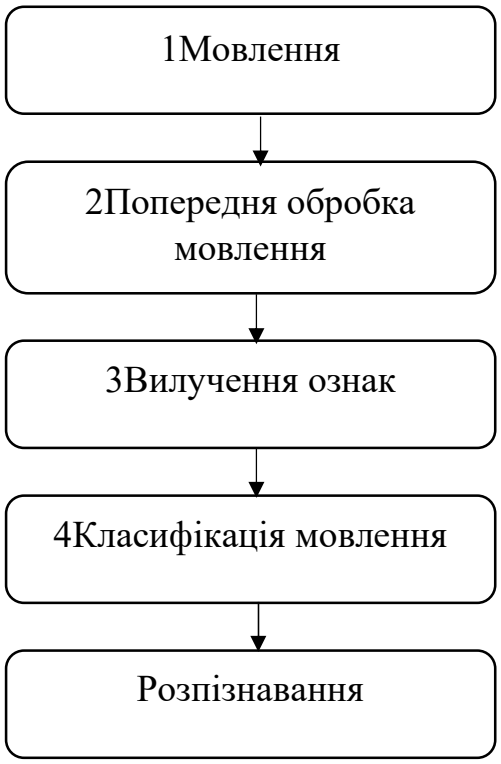

Рисунок 2.1 — Процес розпізнавання мовлення

2.1.1 Мовлення — визначається як здатність висловлювати свої думки та почуття за допомогою артикулюючих звуків. Спочатку мова людини сприймається у формі хвилі. Також є численні інструменти та програмне забезпечення, які записують мовлення людей. Звукове середовище та використовуване обладнання

мають значний вплив на створювану мову. Існує ймовірність поєднання фону або реверберації кімнати з промовою, але це абсолютно небажано.

2.1.2 Попередня обробка мовлення. Рішенням описаної вище проблеми є «Попередня обробка мовлення». Він відіграє важливу роль у скасуванні тривіальних джерел варіацій. Попередня обробка мовлення зазвичай включає придушення реверберації, відлуння, згладжування висоти голосу тобто його нормалізацію (зведення в одну тональність), що значно покращує точність розпізнавання мовлення.

2.1.3. Вилучення ознак. У кожної людини різна мова та різна інтонація. Це пов’язано з різними характеристиками, закладеними у їхньому висловлюванні. Повинна бути ймовірність ідентифікації мови з теоретичної форми сигналу, принаймні теоретично. У результаті величезних варіацій у мовленні виникає неминуча потреба зменшити варіації, виконавши виділення деяких ознак. Далі буде описано деякі технології вилучення ознак, що надзвичайно часто використовуються в наш час.

LPC (лінійно-передбачуване кодування): - це надзвичайно корисна техніка аналізу мовлення для кодування якісного мовлення з низькою швидкістю передачі даних і є одним із найпотужніших методів. Ключова ідея цього методу полягає в тому, що конкретний зразок мовлення в поточний момент можна апроксимувати як лінійну комбінацію попередніх зразків мовлення. У цьому методі цифровий сигнал стискається для грамотного зберігання та передачі. Принцип використання LPC полягає в тому, щоб зменшити суму квадратів відстані між вихідним мовленням і оціненим мовленням протягом кінцевої тривалості. Його можна також використовувати для надання унікального набору коефіцієнтів предиктора.

MFCC (Мел Частотні Кепстральні Коефіцієнти): - це стандартний метод вилучення ознак. За Вікіпедією: «Шкала Мел (від слова мелодія) — це шкала сприйняття звуків, які слухачі оцінюють, як рівні один від одного відстані. Орієнтовна точка між цією шкалою і нормальним вимірюванням частоти визначається шляхом присвоєння тону сприйняття в 1000 мелів, тону 1000 Гц

(рисунок 2.2), що на 40 дБ вище порогу слухача. При частоті вище приблизно 500 Гц слухачі вважають, що все більші інтервали, створюють рівні кроки висоти» [14].

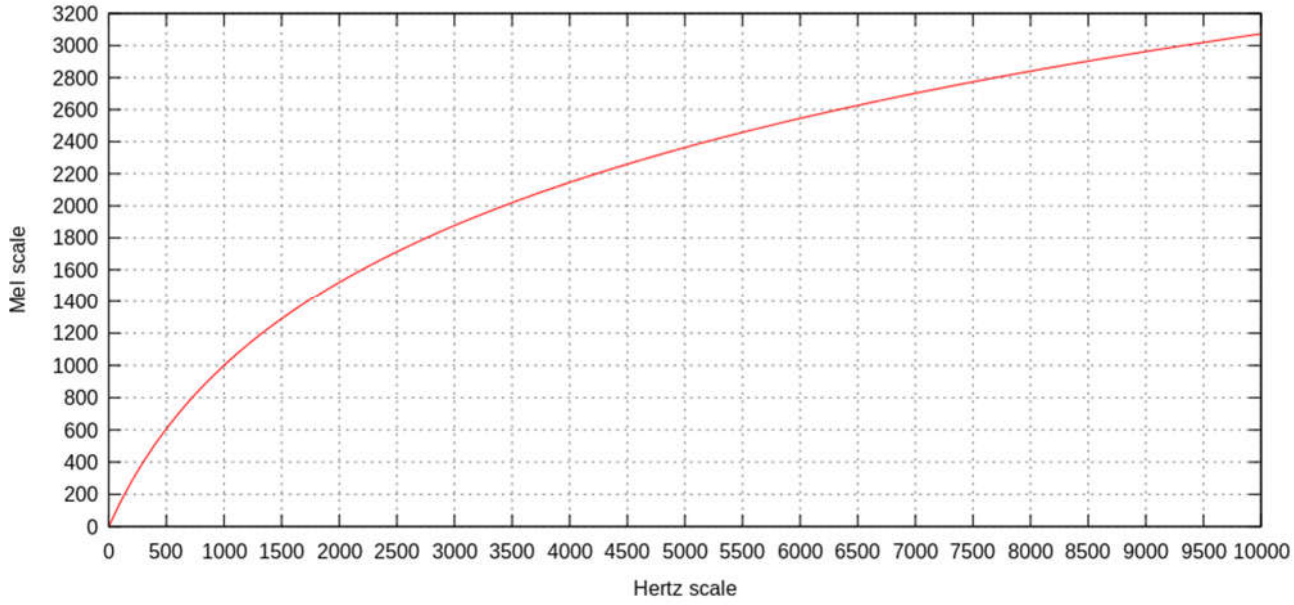

Рисунок 2.2 — Графік співвідношення Мел шкали проти шкали Герца

Він попередньо заснований на частотній області, що базується на Мел шкалі котра базується на шкалі людського вуха. Вони точніші, ніж характеристики часової області, оскільки вони належать до категорії характеристик частотної області. Найбільш помітною перешкодою є його чутливість до шуму, оскільки він сильно залежить від спектральної форми. Для подолання цього недоліку можна використовувати прийоми, що використовують періодичність мовних сигналів, хоча мова також включає аперіодичний вміст.

2.1.4 Класифікація мовлення

Ці системи використовуються для вилучення прихованої інформації з вхідних обробних сигналів і містять складні математичні функції. Далі буде коротко описано деякі часто використовувані методи класифікації мовлення.

DTW (Dynamic Time Warping): — У аналізі часових рядів DTW — це свого роду алгоритм, який вимірює подібність або спорідненість між двома тимчасовими послідовностями, які відрізняються за швидкістю чи часом. Він співвідносить слова мовлення з довідковими словами. Цей метод змінює часовий вимір

невиявлених слів, доки вони не збігаються з опорним словом. Добре відомим застосуванням DTW є автоматичне розпізнавання мови, щоб впоратися з різними швидкостями мовлення. Іншими добре відомі додатки – це розпізнавання підпису в режимі онлайн та розпізнавання мовця.

HMM (Прихована Марківська Модель): - це найбільш часто використовуваний метод для того, щоб розпізнати шаблон у мовленні. Він є безпечнішим і має надійну математичну основу в порівнянні з підходом на основі шаблонів і знань. У цьому методі система, що моделюється, вважається Марковським процесом, що має приховані стани. Промова розподіляється на менші звучні сутності, кожне з яких представляє державу. У простішій моделі Маркова стани чітко видно користувачеві, і, таким чином, ймовірності переходу стану є лише параметрами. З іншого боку, у прихованій моделі Маркова стан безпосередньо не видно, але результат, який залежить від стану, очевидний. HMM особливо відомі своїм застосуванням у навчанні з підкріпленням і розпізнаванні образів, таких, як мова та почерк.

VQ (Векторне квантування): - Даний метод попередньо заснований на принципі кодування блоками. Він дозволяє шляхом циркуляції векторів-прототипів моделювати функціональні щільності ймовірності. Використовувався раніше для стискання даних. Цей метод відображає вектор з великого простору векторів на певні області у цього ж простору. Кодовим словом називаються регіони, що визначаються, як кластери та можуть бути відображеними його центром. Його застосовують для стискання даних з втратами їхньої корекції, а також для розпізнавання та кластеризації образів.

2.1.5 Розпізнавання

Після чотирьох вищезазначених етапів запису мовлення, попередньої обробки мовлення, виділення ознак і класифікації мовлення останнім кроком, який залишився, є розпізнавання мовлення. Після успішного виконання всіх вищезазначених кроків розпізнавання мови можна здійснити кількома підходами, що були описані раніше: акустично-фонетичним підходом; підходом розпізнавання образів; підходом штучного інтелекту та іншими.

2.2 Архітектура штучних нейронних мереж для розпізнавання голосових команд

Штучні нейронні мережі (ANN) (рисунок 2.3)— це обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок тварин. Такі системи навчаються задач (поступально покращують свою продуктивність на них), розглядаючи приклади, загалом без спеціального програмування під задачу. Наприклад, у розпізнаванні зображень вони можуть навчатися ідентифікувати зображення, які містять котів, аналізуючи приклади зображень, позначені, як «кіт» і «не кіт», і використовуючи результати для ідентифікування котів в інших зображеннях. Вони роблять це без жодного апріорного знання про котів, наприклад, що вони мають хутро, хвости, вуса та котоподібні писки. Натомість, вони розвивають свій власний набір доречних характеристик з навчального матеріалу, який вони оброблюють. ANN ґрунтуються на сукупності з'єднаних вузлів, що називають штучними нейронами (аналогічно до біологічних нейронів у головному мозку тварин). Кожне з'єднання (аналогічне синапсові) між штучними нейронами може передавати сигнал від одного до іншого. Штучний нейрон, що отримує сигнал, може обробляти його, й потім сигналізувати штучним нейронам, приєднаним до нього. В поширених реалізаціях ANN сигнал на з'єднанні між штучними нейронами є дійсним числом, а вихід кожного штучного нейрону обчислюється нелінійною функцією суми його входів. Штучні нейрони та з'єднання зазвичай мають вагові коефіцієнти, які підлаштовуються в перебігу навчання. Вони збільшують або зменшують силу сигналу на з'єднанні. Штучні нейрони можуть мати такий поріг, що сигнал надсилається лише якщо сукупний сигнал перетинає цей поріг. Штучні нейрони зазвичай організовано в шари (рис 2.4). Різні шари можуть виконувати різні види перетворень своїх входів. Сигнали проходять від першого (входового) до останнього (вихідного) шару, можливо, після проходження шарами декілька разів. Первинною метою підходу ANN було розв'язання задач таким же способом, як це робив би людський мозок. З часом увага зосередилася на відповідності певним розумовим здібностям, ведучи до відхилень від біології. ANN використовували в ряді різноманітних задач, включно з комп'ютерним баченням, розпізнаванням мовлення, машинним перекладом,

соціально-мережевим фільтруванням, грою в настільні та відеоігри, та медичним діагностуванням [15].

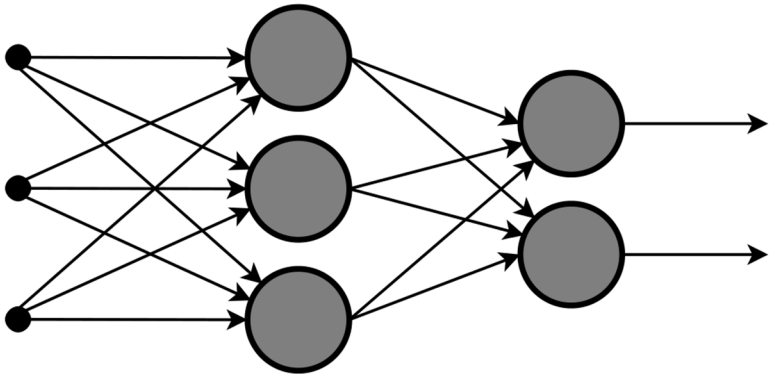

Рисунок 2.3 — Загальна архітектура ANN

Підхід штучних нейронних мереж для ASR можна розділити на дві основні категорії: звичайні нейронні мережі та періодичні нейронні мережі. Головним конкурентом багатошарового перцептрона є RBF, який стає все більш популярною нейронною мережею з різноманітними додатками. Традиційні методи класифікації статистичних моделей стали натхненням для мереж RBF. У RFB вхідний вектор x використовується, для входу усіх базисних радіальних функцій, у кожній з яких містяться різні параметри. Вихід же зумовлює лінійну комбінацію виходів з базисних радіальних функцій (рисунок 2.5).

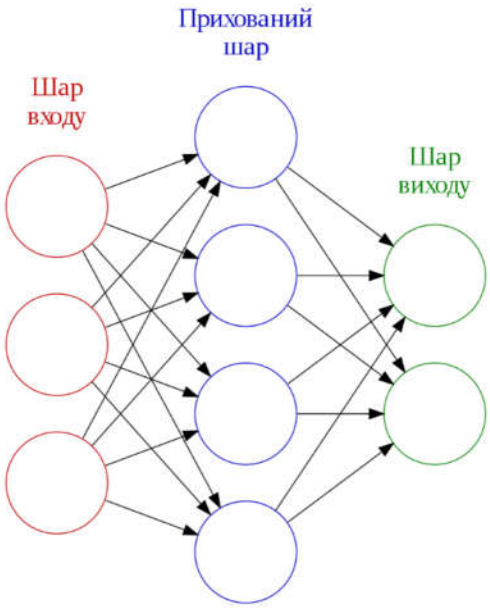

Рисунок 2.4 — ANN з кількома шарами

Мережі радіально базисних функцій (RBF) (рис 2.6)зазвичай мають три шари: вхідний шар, прихований шар з нелінійною RBF функцією активації та лінійний вихідний рівень. Вхід можна моделювати як вектор дійсних чисел $x \in R^n$. Вихід мережі тоді, є скалярною функцією вхідного вектора, $\varphi : R^n \to R$, і має вигляд [16]:

$$\varphi(x) = \sum_{i=1}^{N} a_i \, p(||x - c_i||)$$

Де N — кількість нейронів у прихованому шарі, c_i є центральним вектором для нейрона i, та a_i — це вага нейрона i в лінійному виході нейронів. Функції, які залежать лише від відстані від центру вектора, є радіально симетричними щодо цього вектора, отже, називаються радіальною базисною функцією. У базовій формі всі входи пов'язані з кожним прихованим нейроном. За норму, як правило, обирається Евклідова відстань (хоча відстань Махаланобіса, загалом, більш пасує), та радіальна базисна функція зазвичай вважається розподілом Ґауса [16]:

$$p\big(||x - c_i||\big) = \exp[-\beta||x - c_i||^2]$$

Гаусові базисні функції близькі до центрального вектора в тому сенсі, що зміна параметрів одного нейрона має лише невеликий ефект для вхідних значень, що знаходяться далеко від центру цього нейрона [16].

$$\lim_{||x||\to\infty} p\big(||x - c_i||\big) = 0$$

Завдяки гнучким умовам на форму функції активації, RBF мережі є універсальними апроксиматорами на компактному просторі R^n. Це означає, що мережа RBF з достатньою кількістю прихованих нейронів може апроксимувати будь-яку неперервну функцію на замкненій обмеженій множині з довільною

точністю. Параметри a_i, c_i та β_i визначаються так, щоб оптимізують відповідність між φ і даними [16].

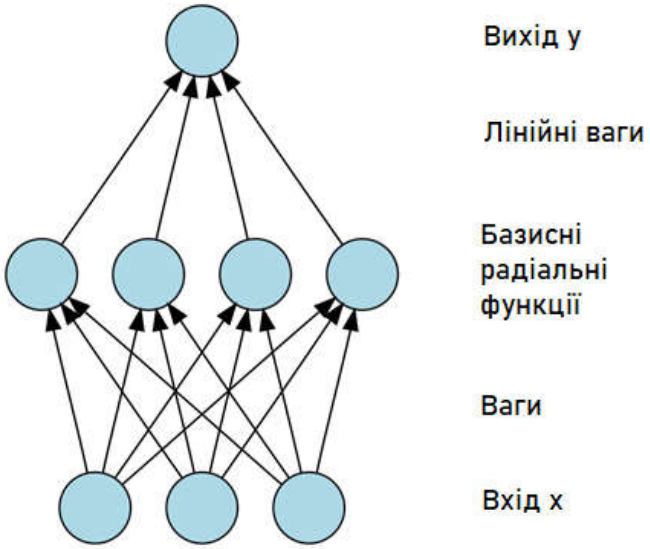

Рисунок 2.5 — Архітектура мережі RBF.

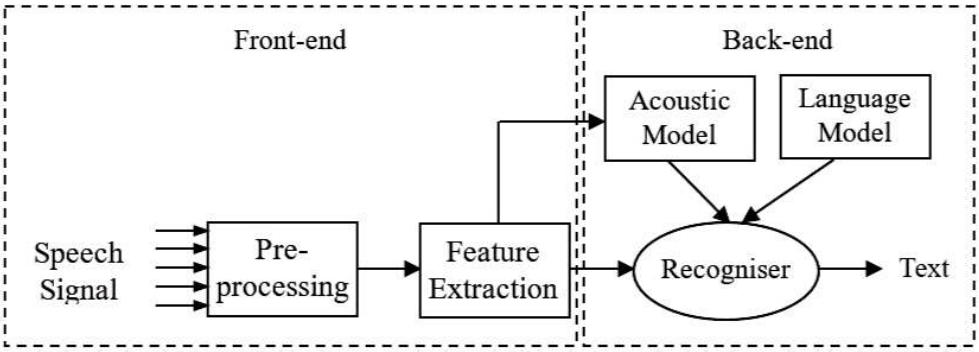

Рисунок 2.6— Загальна архітектура ASR [17]

2.2.1 Рекурентні нейронні мережі (RNN). Є різновидом штучних нейронних мереж, котрі представлені у вигляді спрямованого циклу, де кожен вузол з'єднаний з іншими вузлами. Два блоки стають динамічними, як тільки між ними відбувається зв'язок. Оскільки RNN використовує внутрішню пам’ять так само, як мережі прямої подачі, для обробки послідовності довільного введення, це робить їх ідеальним вибором для розпізнавання мовлення. Ключовою особливістю RNN є те, що активації протікають по циклу, оскільки мережа містить принаймні одне зворотне з’єднання, що дозволяє мережам виконувати тимчасову обробку та вивчати послідовності.

Рекурентні архітектури нейронних мереж бувають різних типів, як-от стандартний багатошаровий перцептрон (MLP) з доданими циклами. Завдяки потужним можливостям нелінійного відображення, MLP має трохи більше пам’яті. Деякі мають більш однорідну структуру і мають астохастичні функції активації.

Архітектуру рекурентних DNN зображено на рисунку 2.7.

Основною структурою є DNN прямого поширення. Вони мають вхідний і вихідний шари і певний прихований шар з повними повторюваними з'єднаннями.

Навчання в RNN можна досягти навіть для простих архітектур. У цих простих архітектурах можна використовувати детерміновані функції активації та подібні процедури градієнтного спуску. Алгоритм зворотного поширення для мереж прямого зв'язку забезпечить оптимальні результати.

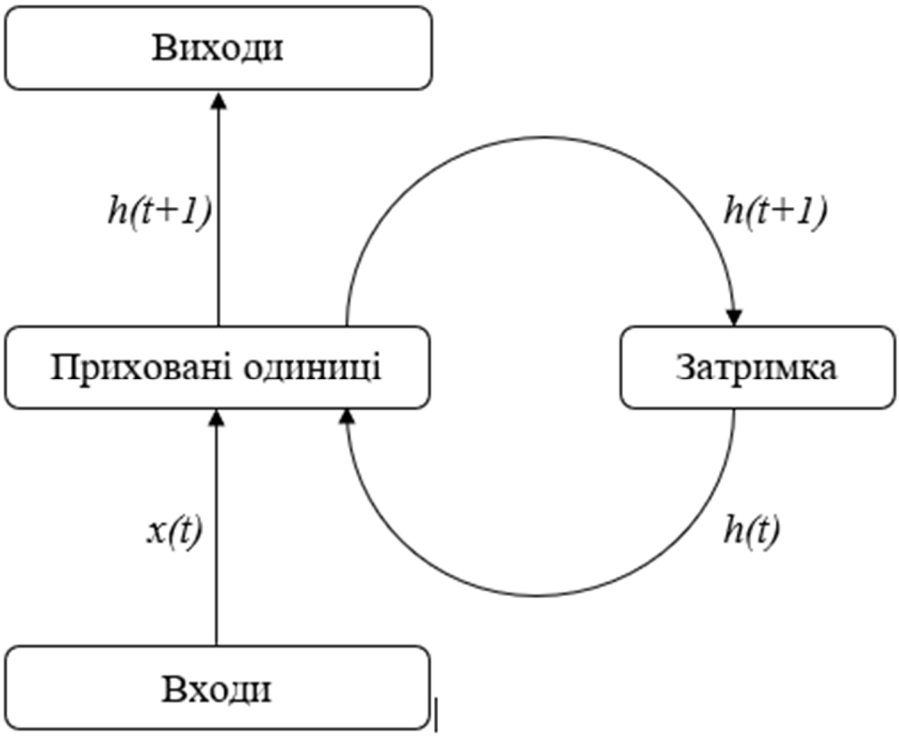

Рисунок 2.7 — Архітектура RNN [18]

На рисунку 2.7 використано найпростішу форму повністю рекурентної нейронної мережі, тобто MLP з попереднім набором прихованих одиничних активацій *h(t)*, які повертаються в мережу разом із входом *h(t+1)*. Тут час *t* потрібно дискретизувати, оновлюючи активації на кожному кроці часу. Шкала часу може

відповідати роботі реальних нейронів, а для штучних систем можна використовувати будь-який розмір кроку часу, відповідний для даної задачі. Необхідно ввести одиницю затримки, щоб затримувати активації, поки вони не будуть оброблені на наступному етапі часу.

2.2.2 Мережі довгої короткотривалої пам’яті (LTSM).

LSTM — це архітектура RNN, яка містить блоки LSTM замість або на додаток до звичайних мережевих одиниць. Архітектура мережі LSTM з одним блоком пам'яті, де зелені лінії є з'єднаннями з затримкою в часі зображено на рисуснку 2.8. Блок LSTM можна описати як «розумний» мережевий блок, який може запам’ятовувати значення протягом довільного періоду часу. Блок LSTM містить лінії поділу (також їх називають вентилями, шлюзами чи воротами), які визначають, коли вхід є достатньо значущим, щоб запам'ятати, коли він повинен продовжувати запам'ятовувати або забути значення, і коли він повинен вивести значення. LSTM були успішно використані для багатьох завданнях позначення послідовностей та їх передбачення. В архітектурі LSTM повторюваний прихований шар складається з набору періодично підключених підмереж, відомих як «блоки пам’яті». Кожен блок пам’яті містить одну або кілька самоз’єднаних осередків пам’яті та три мультиплікаційні лінії розподілу для керування потоком інформації. У кожному блоці LSTM потік інформації в комірку та з неї охороняється навченими вхідними та вихідними лініями розподілу. Пізніше, щоб забезпечити спосіб очищення блоків пам’яті, додається шар забуття [18].

Загальноприйняті LSTM можна визначити наступним чином:

Враховуючи вхідну послідовність $x = (x_1, x_2 \dots x_T)$, звичайна RNN обчислює приховану векторну послідовність $h = (h_1, h_2 \dots h_T)$ і вихідну векторну послідовність $y = (y_1, y_2 \dots y_T)$ від t = 1 до T наступним чином:

$$h_t = H\,(W_{xh}x_t + W_{hh}h_t - 1 + b_h)$$

$$y = W_{hy}h_t + b_y$$

Де, W позначає вагові матриці, b позначає вектори зміщення, а $H(\cdot)$ є рекурентною функцією прихованого шару.

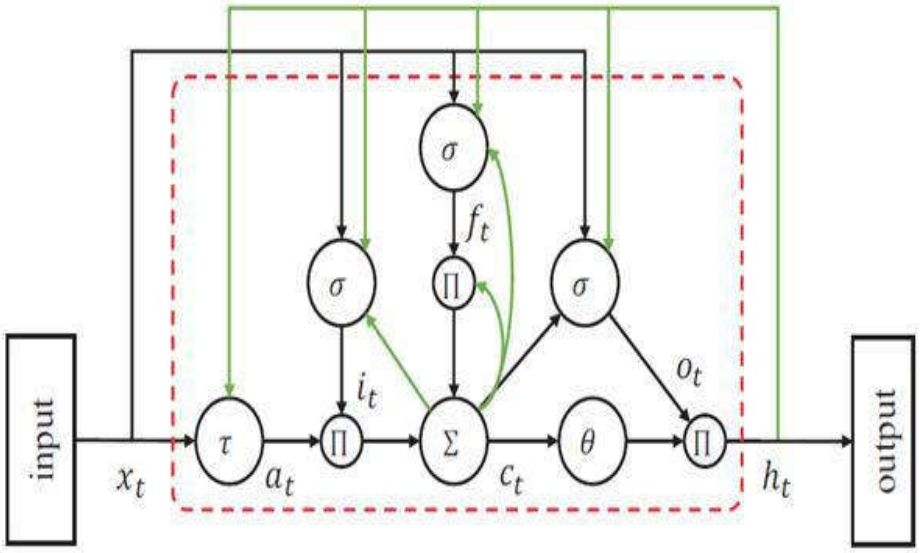

Рисунок 2.8 — Мережа LTSM з одним блоком пам’яті [18]

Переваги та недоліки LSTM будуть наведені нижче:

а. Недоліки LTSM:

– Зворотне поширення LSTM зберігає лише один із вхідних даних, який не зменшить похибку, що, у свою чергу, не зменшить поступово помилку навіть при розв’язанні підцілі. Це можна зменшити за допомогою використання повного градієнта.

– Але повний градієнт не рекомендується, тому, що: 1) Він більш складний; 2) Потік помилок можна побачити, лише коли використовується усічений LSTM; 3) Повний BPTT не буде перевершувати усічений BPTT.

– Оскільки один прихований шар замінюється на три шари, ваговий коефіцієнт збільшується до 32. Таким чином, кожен блок пам'яті потребує двох додаткових блоків.

– LSTM має ті ж проблеми, що й у мережах прямого поширення, оскільки вона веде себе, як нейронна мережа прямого поширення, навчена нейронною мережею зворотного поширення, щоб побачити повний вхідний рядок.

– Підрахунок часових кроків є практичною проблемою усіх градієнтно базованих підходів; Навіть у LSTM завдання простіше, коли сигнал виник 5 кроків тому, і був проблемою, якщо він виник, скажімо, 100 кроків тому [18].

б. Переваги LTSM:

– Постійне зворотне поширення помилок у блоках пам'яті призводить до здатності LSTM подолати дуже тривалі часові затримки у випадку проблем, подібних до тих, що розглядалися вище. Для проблем із довгою затримкою, подібних до розглянутих раніше, LSTM може обробляти шуми, такі, як розподілені уявлення та безперервні значення. На відміну від повної автоматизації, або прихованих Марківських моделей LSTM не вимагає попереднього вибору кінцевої кількості станів. В принципі, він може мати справу з необмеженою кількістю станів.

– Для проблем, які обговорювалися раніше, LSTM добре узагальнює, навіть якщо позиції широко відокремлених релевантних вхідних даних у вхідній послідовності не мають значення. На відміну від попередніх підходів, цей швидко вчиться розрізняти два або більше широко відокремлених входження певного елемента у вхідній послідовності, не залежно від відповідних прикладів навчання з короткою затримкою.

– LSTM добре працює в широкому діапазоні параметрів, таких як швидкість навчання, зміщення вхідних ліній поділу та зміщення вихідних ліній поділу. Наприклад, швидкість навчання може здатися досить невеликою. Однак така швидкість навчання зумовлена очищенням вихідних ліній розподілу, таким чином автоматично усуваючи власні негативні ефекти.

– Складність оновлення LSTM за кроком ваги та часу, по суті, є складністю BPTT. Це чудово в порівнянні з іншими підходами, такими як RTRL. Однак, на відміну від повного BPTT, LSTM є локальним як у просторі, так і в часі.

2.2.3 Згорткові нейронні мережі (CNN)

З іншої сторони, наскрізні системи обробляють акустичні кадри до фонем лише за один крок. Наскрізне навчання означає, що всі модулі навчаються одночасно. Передові методи глибокого навчання полегшують навчання системи

наскрізним способом. Вони також мають можливість тренувати систему безпосередньо за допомогою необроблених сигналів, тобто без функцій, створених вручну. Таким чином, парадигма ASR змінюється від кепстральних ознак, таких як MFCC, PLP , до дискримінаційних ознак, засвоєних безпосередньо з необробленого мовлення. Наскрізна модель може приймати необроблений мовний сигнал як вхідні дані і генерує умовні ймовірності класу фонем як вихідні дані. Три основних типи наскрізних архітектур для ASR – це метод, заснований на увазі, конекціоністської тимчасова класифікації (CTC) і моделі прямого мовлення на основі CNN (рисунок 2.9).

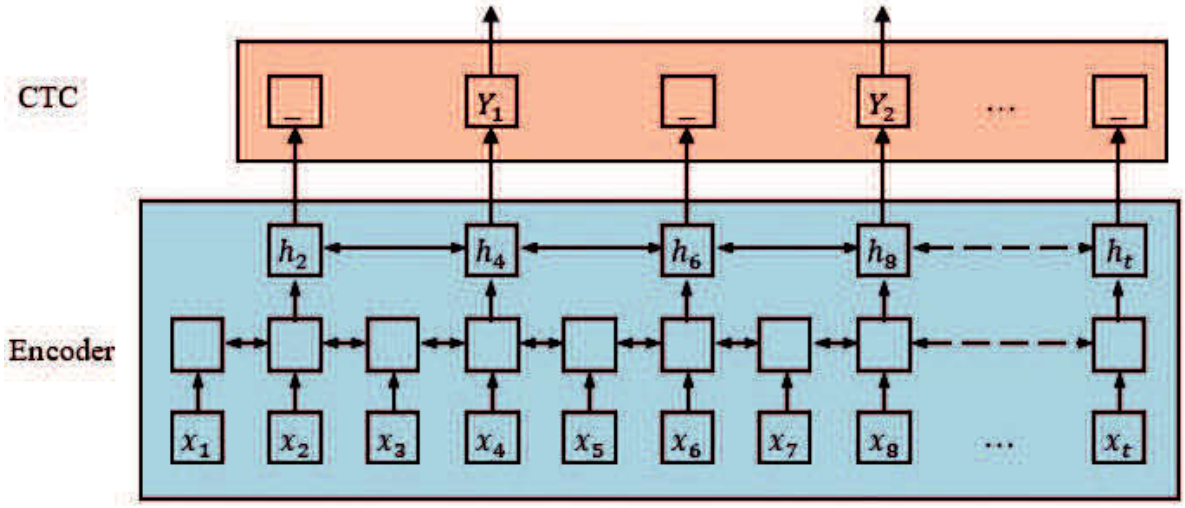

Рисунок 2.9 — Модель CTC для розпізнавання мовлення [19]

Конекціонізм — підхід штучного інтелекту до пізнання, в якому численні зв’язки між вузлами (еквівалентні клітинам мозку) утворюють масивну інтерактивну мережу, в якій багато процесів відбуваються одночасно. Певні процеси в цій мережі, що працюють паралельно, об’єднані в ієрархії, які приносять такі результати, як думки чи дії.

Моделі на основі уваги (рисунок 2.10) безпосередньо перетворюють мовлення в фонеми. Декодер, орієнтований на увагу, використовує RNN для виконання відображення послідовності в послідовність без будь-якого попередньо визначеного вирівнювання. У цій моделі вхідна послідовність спочатку перетворюється у представлення вектора фіксованої довжини, а потім декодер відображає цей вектор фіксованої довжини у вихідну послідовність. Декодер, орієнтований на увагу, значною мірою здатний вивчати зіставлення між вхідними

та вихідними послідовностями змінної довжини. Чоровскі і Джейтлі запропонували незалежну від мовця модель послідовності до послідовності і досягли 10,6% WER без окремих мовних моделей і 6,7% WER з триграмною мовною моделлю для набору даних Wall Street Journal [20].

З іншої сторони, акустична модель на основі CNN, запропонована Палазом та іншими [17, 19, 21], котра обробляє вихідне мовлення безпосередньо як вхідні дані. Ця модель складається з двох етапів: етап вивчення ознак, тобто кілька згорткових шарів, і етап класифікатора, тобто повністю пов’язані шари. Обидва етапи вивчаються спільно шляхом мінімізації вартості на основі функції відносної ентропії. У цій моделі інформація витягується фільтрами на першому згортковому шарі та моделюється між першим і другим згортковим шаром. На стадії класифікатора вивчені функції класифікуються за повністю підключеними шарами та шаром м’яких максимальних значень. Цей підхід вимагає порівнянну або кращу продуктивність, ніж традиційна система, заснована на кепстральних (MFCC) ознаках, з подальшим навчанням ANN для розпізнавання фонем у наборі даних TIMIT.

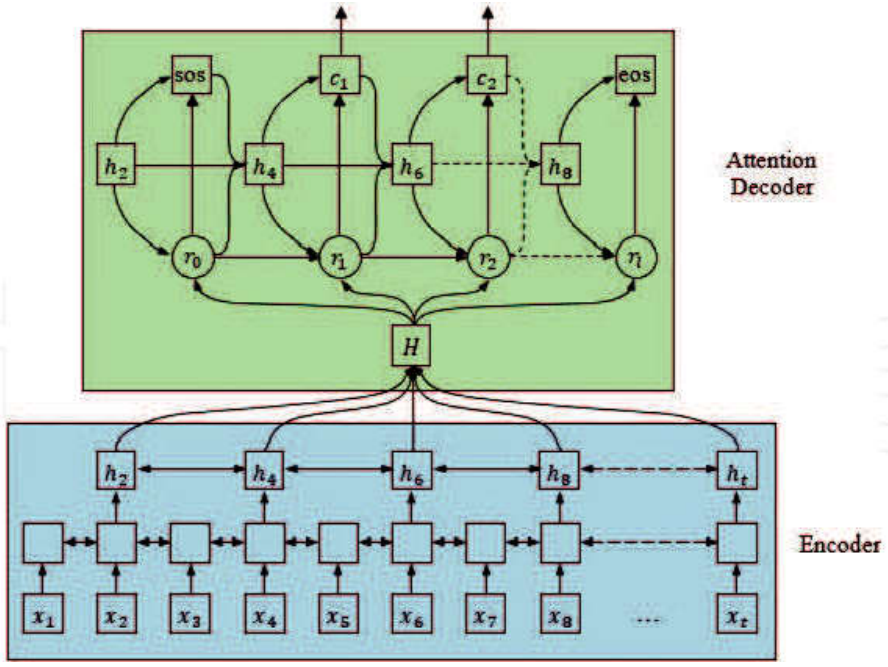

Рисунок 2.10 — Модель ASR на основі уваги [19]

Набір даних TIMIT — це корпус фонематично та лексично транскрибованого мовлення носіїв американської англійської мови різної статі та діалектів. Кожен транскрибований елемент був окреслений в часі [22].

Коннекціоністська тимчасова класифікація (CTC) також базується на теорії рішень Байєса. Слід зазначити, що послідовність букв L-довжини,

$$C' = \{< b >, c_1, < b >, c_2, < b >, \dots, c_L, < b >\} \approx$$

$$\approx c'_l \in U \cup \{< b >\} | l = 1, \dots, 2L + 1$$

У C', c'_l завжди є « < b > » і коли l є непарним і парним числом відповідно. Подібно до моделі DNN/HMM (рисунок 2.11), кадрова послідовність букв з додатковим порожнім символом також представлено.

$$Z = \{ z_t \in U \cup \{< b >\} \mid t = 1, \dots, T\}$$

Апостеріорний розподіл, $p(C|X)$, можна розкласти, як:

$$p(C|X) = \sum_z p(C|Z, X) p(Z|X) \approx$$

$$\approx \sum_z p(C|Z) p(Z|X)$$

CTC також використовує припущення Маркова, тобто $p(C|Z, X) \approx p(C|Z)$, щоб спростити залежність акустичної моделі CTC, $p(Z|X)$, та літерної моделі CTC, $p(C|Z)$.

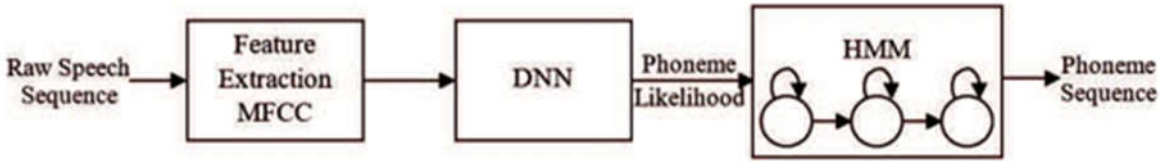

Рисунок 2.11 — Гібридне розпізнавання фонем DNN/HMM [23].

Акустична модель CTC, так само, як акустична модель DNN/HMM, $p(Z|X)$ можна додатково розкласти на множники, використовуючи правило ймовірнісного ланцюга та припущення Маркова наступним чином:

$$p(Z|X) = \prod_{t=1}^{T}(z_t|z_1, \dots, z_{t-1}, X) \approx$$

$$\approx \prod_{t=1}^{T}(z_t|X)$$

Покадровий апостеріорний розподіл, $p(Z|X)$, обчислюється з усіх вхідних даних, *X*, і безпосередньо моделюється за допомогою двонаправленого LSTM [24]:

Буквенна модель CTC. Застосовуючи правило ймовірнісного ланцюга теорії рішень Байєса та припущення Маркова, $p(Z|X)$, можна записати, як:

$$p(C/Z) = \frac{p(C/Z)p(C)}{p(Z)} =$$

$$= \prod_{t=1}^{T} p(z_t|z_1, \dots, z_{t-1}, C)\frac{p(C)}{p(Z)} \approx$$

$$\approx \prod_{t=1}^{T} p(z_t|z_{t-1}, C)\frac{p(C)}{p(Z)}$$

де $p(z_t|z_{t-1}, C)$ представляє ймовірність переходу стану. $p(C)$ представляє модель мови на основі букв, а $p(Z)$ представляє попередню ймовірність стану. Архітектура CTC включає в себе модель мови на основі літер. Архітектура CTC також може включати модель мови на основі слів, використовуючи перетворювач

кінцевого стану від букви до слова під час декодування [25]. CTC має властивість монотонного вирівнювання, тобто коли $z_{t-1} = c'_m$, тоді $z_t = c'_l$, де $l \geq m$.

Властивість монотонного вирівнювання є важливим обмеженням для розпізнавання мовлення, тому відображення ASR послідовності в послідовність має відповідати монотонному вирівнюванню. Цю властивість також задовольняє HMM/DNN.

Формулювання CTC таке ж, як і HMM/DNN. Невелика різниця полягає в тому, що правило Байєса застосовується до $p(C|Z)$ замість $p(W|X)$. Він також має три компоненти розподілу, такі як HMM/DNN, тобто покадровий задній розподіл, $p(z_t|X)$; ймовірність переходу, $p(z_t|z_{t-1}, C)$; і буквенну модель, $p(C)$. Він також використовує припущення Маркова. Він не повністю використовує переваги наскрізного ASR, але його подання виведення символів все ще має наскрізні переваги.

CNN є популярними варіантами глибокого навчання, які широко використовуються в системах ASR. CNN мають багато привабливих удосконалень, наприклад, розподіл ваги, згорткові фільтри та об'єднання. Таким чином, CNN досягли вражаючої продуктивності в ASR. CNN складаються з кількох згорткових шарів. На рисунку 2.12 показана блок-схема CNN. Зазвичай описують три стани згорткового шару, а саме: згортання, об'єднання та нелінійність.

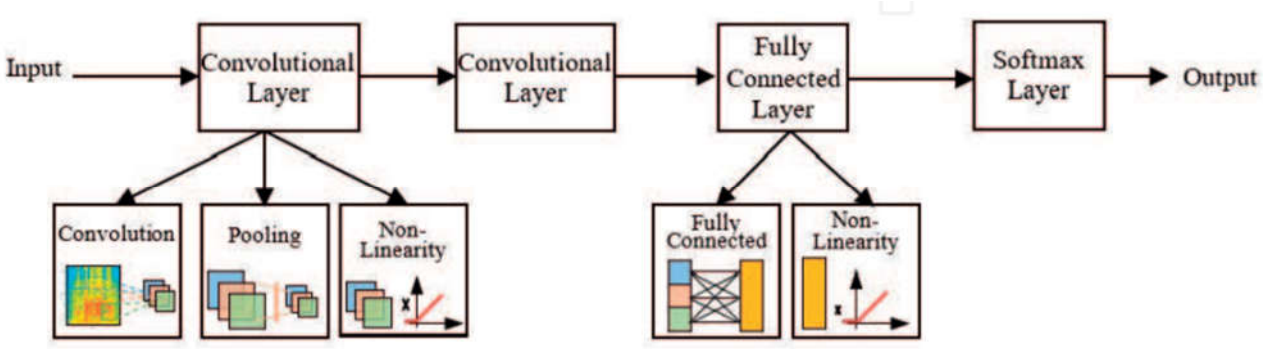

Рисунок 2.12 — Блок-схема CNN [23]

Глибокі CNN встановили нову віху, досягнувши приблизної продуктивності людського рівня за допомогою передових архітектур та оптимізованого навчання [23]. CNN використовують нелінійну функцію для безпосередньої обробки даних

низького рівня. CNN здатні вивчати функції високого рівня з високою складністю та абстракцією. Об’єднання – це серце CNN, яке зменшує розмірність карти об’єктів.

Об’єднання є важливою концепцією, яка перетворює спільне представлення ознак у цінну інформацію, зберігаючи корисну інформацію та усуваючи незначну інформацію. Невеликі частотні зсуви, які є поширеними в мовному сигналі, ефективно обробляються за допомогою об’єднання. Об’єднання також допомагає зменшити спектральну дисперсію, присутню у вхідній мові. Воно відображає вхід із сусідніх шарів у вихідний, застосовуючи спеціальну функцію.

Після поелементних нелінійностей об’єкти передаються через шар об’єднання. Цей шар виконує зниження дискретизації на картах об’єктів, що надходять з попереднього шару, і створює нові карти об’єктів зі згорнутою роздільною здатністю.

Цей шар різко зменшує просторовий розмір вхідних даних. Він служить двом основним цілям: перша полягає в тому, що кількість параметрів або ваги зменшуються на 65%, що зменшує витрати на обчислення. По-друге, він контролює надмірну підгонку. Цей термін відноситься до того, коли модель занадто налаштована на навчальні приклади.

2.2.4 Наскрізний підхід на основі CNN

Нова акустична модель, заснована на CNN, що було запропоновано Палазом, Колобертом та Досс [21], що показано на рисунку 2.13. При цьому необроблений мовний сигнал сегментується на вхідний мовний сигнал $s_t^c = \{s_{t-c}, \dots, s_t, \dots, s_{t+c}\}$, у контексті кадрів 2c, що мають вікно w_{in} мілісекунд. Перший згортковий рівень вивчає корисні функції з необробленого мовного сигналу, а інші згорткові шари далі обробляють ці функції в корисну інформацію.

Після обробки мовного сигналу CNN оцінює клас умовної ймовірності, тобто $P(i/s_t^c)$, який використовується для розрахунку імовірності масштабу емісій $P(s_t^c/i)$. Перед етапом класифікації в мережі є кілька етапів фільтрації. Стадії фільтрації — це комбінація згорткового шару, шару об’єднання та нелінійності.

Спільне навчання етапу ознак і стадії класифікатора виконується за допомогою алгоритму зворотного поширення.

Наскрізний підхід передбачає розуміння наступного:

1. Мовні сигнали мають нестаціонарний характер. Тому вони обробляються в короткочасній манері. Традиційні методи вилучення ознак зазвичай використовують розширюваний розмір фрейму в 20–40 мс. Хоча при наскрізному підході короткочасна обробка сигналу являється необхідністю. Тому розмір короткочасного фрейму береться за гіперпараметр, який автоматично визначається під час навчання.

2. Вилучення ознак є операцією фільтрації, оскільки такі компоненти, як перетворення Фур'є, дискретне косинусне перетворення тощо, є операціями фільтрації. У традиційних системах фільтрація застосовується як за частотою, так і за часом. Отже, цей фактор також враховується при побудові згорткового шару в наскрізній системі. Тому кількість шарів фільтрації та їх параметри приймаються, як гіперпараметри, які автоматично визначаються під час навчання.

3. Короткочасна обробка мовного сигналу поширює інформацію в часі. У традиційних системах ця поширена інформація моделюється шляхом обчислення тимчасових похідних і контекстної інформації. Таким чином, проміжне представлення надходить у класифікатор і розраховується за допомогою тривалого інтервалу часу вхідного мовного сигналу. Тому за гіперпараметр береться w_{in}, розмір вхідного вікна, який оцінюється під час навчання.

Наскрізна модель оцінює $P(i/s_t^c)$ шляхом обробки мовного сигналу з мінімальними припущеннями або попередніми знаннями.

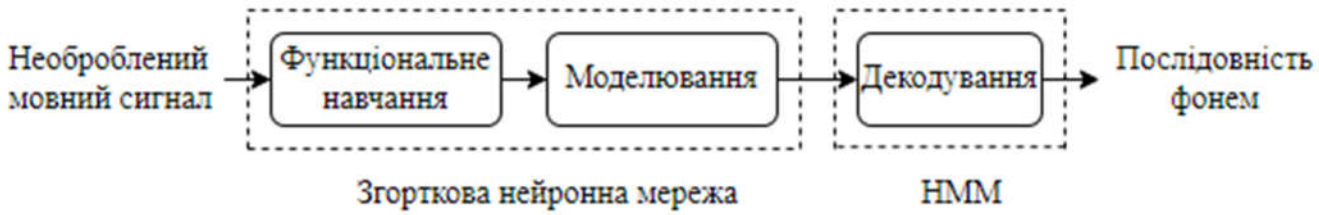

Рисунок 2.13 — Система розпізнавання фонем необробленого мовлення на основі CNN [21]

2.3 Алгоритм методу розпізнавання голосових команд

На рисунку 2.14 зображено блок-схему алгоритму розроблюваного методу розпізнавання голосових команд.

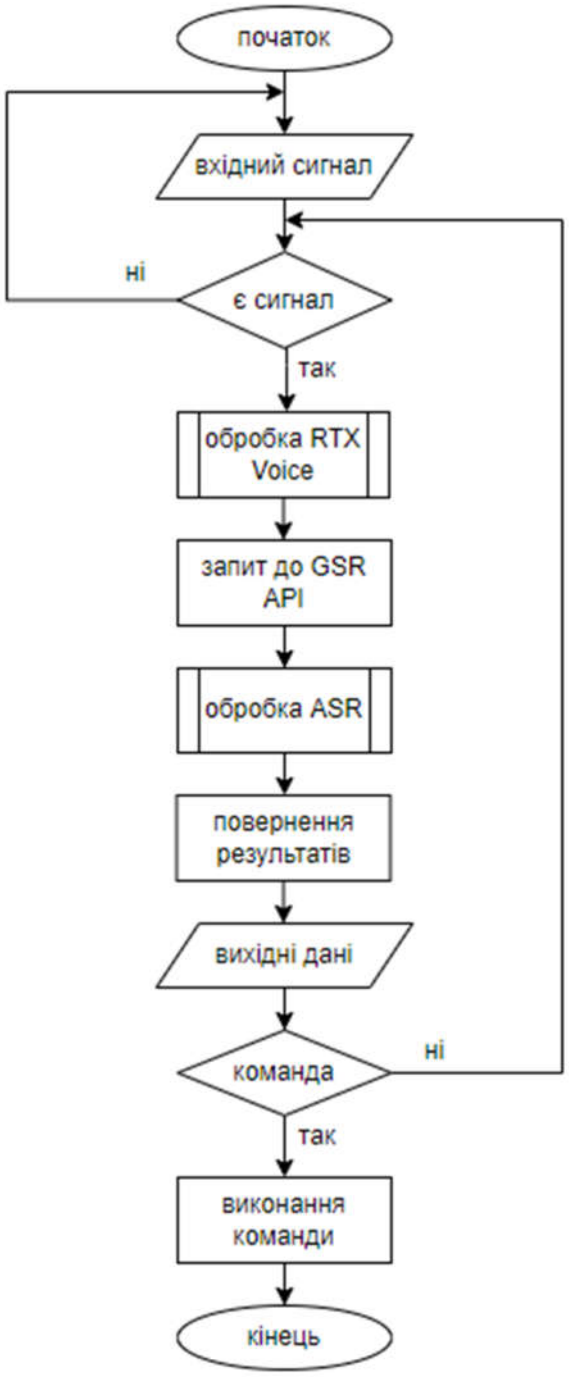

Рисунок 2.14 — Блок-схема розроблюваного методу

Опис алгоритму:

- Запуск скрипта.
- Подача вхідного сигналу на мікрофон.
- Далі слідує перевірка на наявність сигналу, якщо його нема переходимо до попереднього пункту, якщо є — йдемо далі.

- Відбувається шумоусунення за допомогою RTX Voice.
- Надсилається запит до SGR API.
- Відбувається обробка даних ASR
- Повертаються оброблені вихідні дані.
- Перевірка вихідних даних на наявність команд, якщо команд не було виявлено повертаємося на початок, якщо були — йдемо далі.
- Виконуємо розпізнану команду.
- Завершаємо роботу скрипта

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ

3.1 Структурна схема системи розпізнавання голосових даних

Основною метою розробки методу є зниження помилок при розпізнаванні, за умов зашумлення та планування відносно зручного інтерфейсу для його демонстрації чи подальшого використання (у вигляді Python скрипта).

Розроблюваний скрипт за допомогою методу повинен:

- Покращити точність розпізнавання ASR;
- Забезпечити коректне виконання команд.

Вихідні дані до роботи:

- Google ASR – методи та функції ANN, котрі забезпечують розпізнавання мовлення.
- GSR API – методи та функції, котрі «зв’язують» ASR розміщену на сервері з Python скриптом та надають йому доступ до спеціальних функцій Google ASR.
- Python Script – функції, класи та методи, котрі забезпечують отримання команд за допомогою апробації ASR через GSR API та забезпечують їх виконання середовищі OS.
- RTX Voice – підхід на основі ANN для усунення шумів із вхідної мовленнєвої послідовності.
- OS Windows – середовище виконання отриманих команд.

Загальну структуру системи зображено на рисунку 3.1.

Інтеграція між скриптом та GSR API, полягає в тому, що у скрипті можна викликати методи, класи чи функції Google ASR, для автоматичного розпізнавання послідовності звуків, що надходить із системи з подальшою обробкою за допомогою RTX Voice. Після отримання розшифрованого тексту, скрипт оброблює вхідний текст та за наявності команди виконує її шляхом виклику відповідногї функції бібліотеки класу.

На рисунку 3.2 зображена проектована черга виконання запитів (процесів) між компонентами розроблюваного методу та його вихідними даними згідно з порядком обробки обраного шаблону розробки.

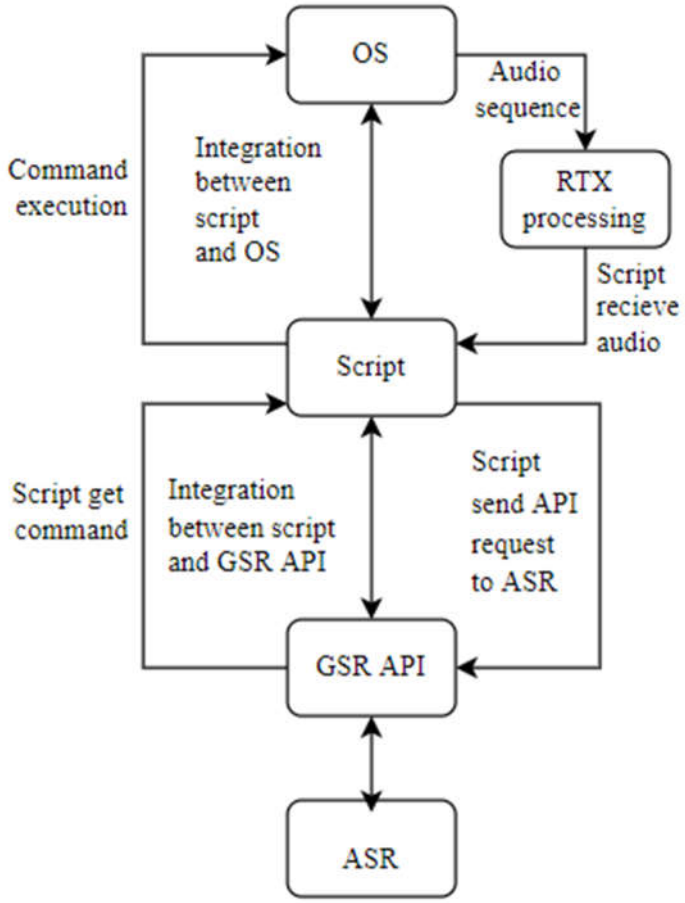

Рисунок 3.1 — Загальна структура системи

А саме: користувач з системи вводу ОС надсилає голосовий запит до скрипту (з котрого попередньо було усунуто посторонні шуми), той у свою чергу зв’язаний з GSR API, що запитують обробку отриманого голосового повідомлення у ASR котра розміщена на серверах Google. Котра розшифрує отримані дані та відбудеться один з наступних варіантів в залежності від повідомлення: якщо користувач не надіслав голосових даних – ніяких дій виконано не буде, у разі отримання будь-якого голосового повідомлення, воно буде декодоване та пройде зворотній шлях (від частини ASR розміщеної на серверах Google до його скриптової частини) та буде оброблено цією частиною відповідно до розроблюваного модуля логіки методу обробки команд виконає команду в середовищі ОС, щоб видати відповідний результат.

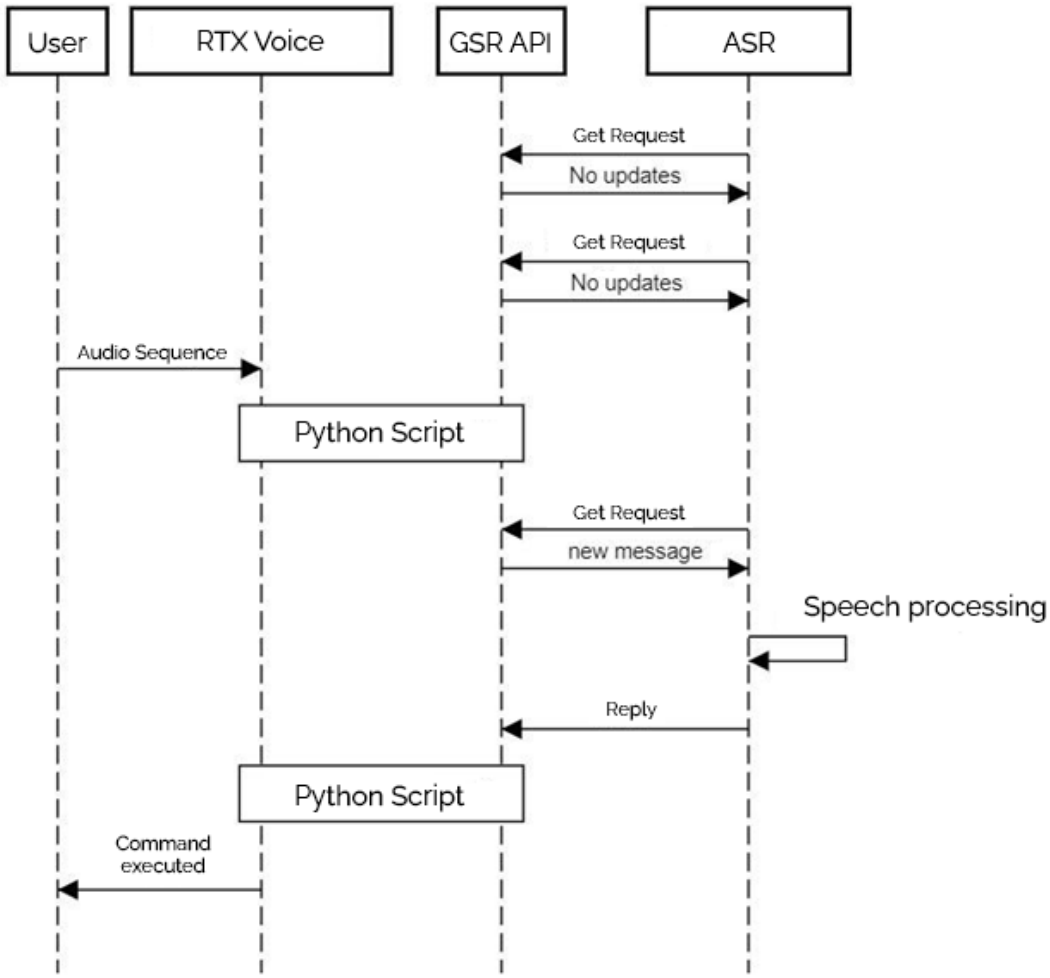

Рисунок 3.2 – Діаграма послідовності розроблюваної системи

Обробка запиту, що зображена на рисунку 3.3: потребує вхідних даних з боку користувача у вигляді голосової команди, що являє собою wav-файл, який надалі буде оброблятися ASR. Що у свою чергу вимагає обов'язкової участі GSR API, програмного забезпечення у вигляді: запущеного Python скрипту, RTX Voice у вигляді програмно-апаратного забезпечення на базі відеокарт NVIDIA RTX котрі в свою чергу використовують нейронніі мережі та запитуваних даних у вигляді бази даних, або переліку команд. Процес обробки даних коригується щодо якості звукового сигналу та методів розробки ASR. На виході користувачу надсилається результат виконання залежно від введених раніше даних: при коректних запитах користувачу буде виконана необхідна команда, або ж при відсутності команд чи от, якщо запит введено некоректно – буде відправлено прохання повторити спробу, або змінити свій вибір на один із запропонованих.

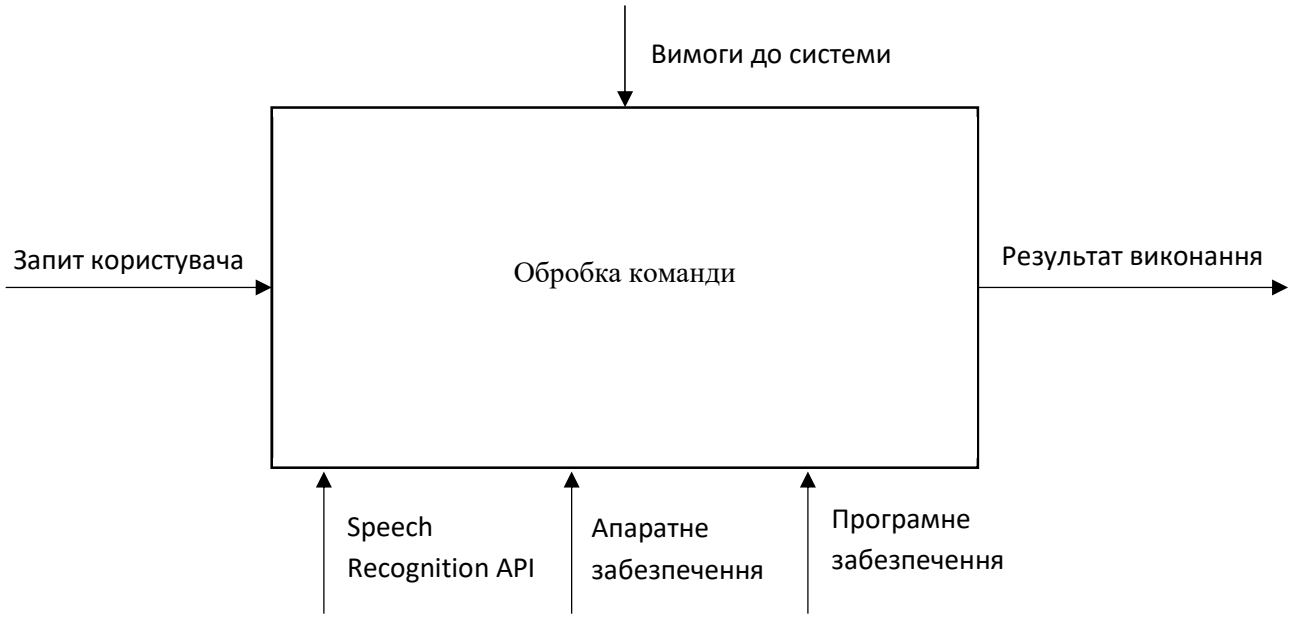

Рисунок 3.3 – DFD діаграма обробка команд

3.2 Опис засобів розробки

3.2.1 Python. Для розробки взаємодії частин системи та її об’єктів, таких, як програмне забезпечення, нейромережі у хмарі та апаратне забезпечення з використанням нейронних мереж для виявлення різноманітних зашумлених команд та їхнього розпізнавання було обрано скриптову мову програмування Python. Оскільки база бібліотек та різноманітних мікросервісів написаних із використанням Python’ну вельми значна та містить необхідні для зчитування з мікрофону та різноманітних звукових карт, файлів чи інших пристроїв голосового вводу звукових даних для подальшої роботи з ними, а також містить значну кількість бібліотек та підтримуваних API, окрім обраних раніше, для можливості подальшого покращення обраного методу розпізнавання мови, включно з бібліотеками PyAudio, Google Speech API, апаратними засобами розробки та іншими. А також через високу потужність за допомогою використання його засобів та методів і низький поріг входу у Python, він являється найкращим кандидатом для

демонстративної реалізації методу, проведення експериментів на його основі та подальшого розвитку розроблюваного методу.

Python – це скриптова мова програмування. Своєю універсальністю, пасує для вирішення величезної кількості різноманітних задач практично на усіх платформах, починаючи з серверних ОС та Android і до, проте не обмежуючись системами керування мобільними пристроями. Він використовується у розробці веб-додатків, сайтів, плагінів та інших різноманітних веб-продуктів, створенні додатків для персональних комп'ютерів, смартфонів, як згадувалося раніше різноманітних мобільних пристроїв та систем керування ними, розробці ігор, програмуванні чат-ботів, а також на даний час є однією з найпотужніших мов в машинному навчанні, Big Data та Data Science.

Python являє собою інтерпретовану мову програмування, тобто він не компілюється, а саме до ініціалізації та запуску скрипту інтерпретатором написаний скрипт сам пособі буде текстовим файлом з іншим розширенням, а саме «.py». Займатися розробкою програмного забезпечення з його допомогою можна практично на будь-яких пристроях та платформах, та як мова логічна, добре зрозуміла завдяки лаконічності та хорошій проектованості мови.

Розробка програмного забезпечення на Python'ні проходить в кілька разів швидше, за рахунок відсутності boilerplate коду тому, що доводиться писати значно менше коду, аніж на C, Java чи інших мовах.

Поняття відноситься до розділів коду, які повинні бути написані в багатьох місцях з мінімальними змінами. Часто використовується по відношенню до мов, на яких програміст повинен написати багато коду для виконання мінімального завдання [26]

3.2.2 PyCharm IDE.

PyCharm - одна з найкращих, якщо не найкраща, повнофункціональна, спеціалізована та універсальна IDE для розробки на Python. Він має масу можливостей, які економлять час, допомагаючи із рутинними завданнями.

PyCharm являється найпопулярнішим і широко використовуваним спеціалізованим IDE для розробки та програмування додатків Python.

PyCharm — це мультиплатформенний IDE, розроблений JetBrains спеціалізованим для Python, на відміну від VisualStudio. PyCharm як основну IDE для розробці на Python використовують у таких організаціях, як, наприклад: Twitter, Facebook, Amazon і Pinterest.

PyCharm IDE (рисунок 3.4) складається з редактора та компілятора, котрі використовуються для написання й компіляції програм. Він поєднує в собі певну кількість функцій, необхідних для комфортної розробки програмного забезпечення.

Наявність IDE значно полегшує процес розробки та програмування. Він інтерпретує написаний код, і пропонує релевантні ключові слова для вставки. Зручно розрізняти класи та методи, оскільки PyCharm IDE виділяє їх різними кольорами. IDE також надає різний колір шрифту чи підкреслення для правильних і неправильних ключових слів. При написанні неправильних ключових слів, IDE намагається передбачити ключове слово, яке повинно було бути написано, і, або автоматично заміняє його, або пропонує заміну.

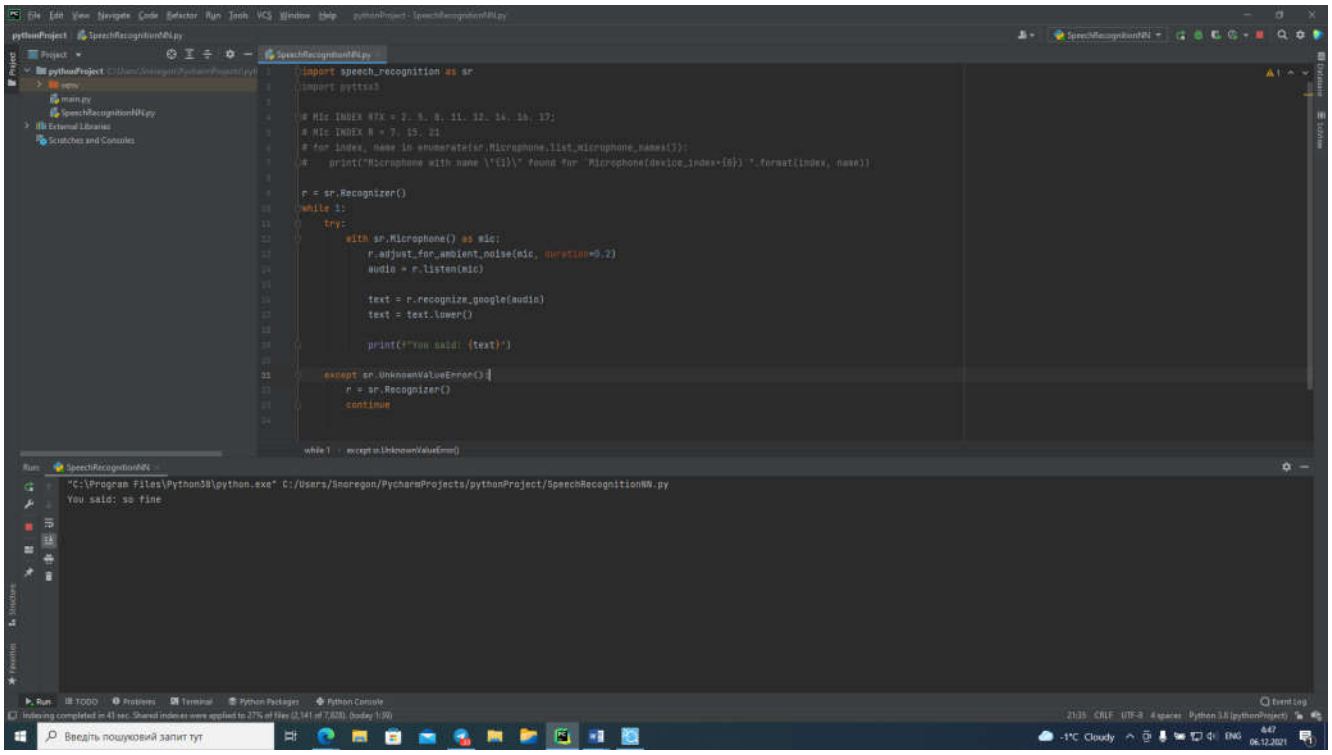

Рисунок 3.4 — PyCharm IDE

Нижче наведено основні причини використання PyCharm IDE для розробки:

– IDE складається з вікна текстового редактора, де можна писати код програми, з підтримками функцій автозаповнення, завершення та внесення правок до коду, що описано раніше, це в свою чергу допомагає лаконічному оформленню коду, підвищує його читабельність та допомагає легко виявляти помилки.

– Навігація по коду: PyCharm підтримує перехід до файлу, класу, або ж функції за допомогою спеціальних комбінацій клавіш, що в рази полегшує правки коду в різних місцях лістингу програми, особливо полегшує редагування відносно «далеких» елементів коду програми та пошуку їх входження.

– Рефакторинг: Вище перелічені функції редактору значно полегшують задачі з внесення змін до проекту, рефакторинг коду під нові вимоги чи роботу над Legacy проектами.

– Підтримка синтаксису багатьох мов. Не зважаючи на те, що PyCharm розроблявся спеціалізованим під Python, через специфіку модульного використання він підтримує синтаксис досить великої кількісті мов програмування таких, наприклад, як: JavaScript, CoffeeScript, TypeScript, Cython, SQL, HTML / CSS, мови шаблонів, AngularJS, Node.js та інші, якщо ж підтримки потрібної мови не має за замовчуванням — допоможе встановлення відповідного плагіну.

– Також в ньому присутнє вікно редактора проекту, де можна бачити структуру проекту цілком з усіма вкладеними файлами розроблюваного програмного проекту.

– За допомогою вбудованого компілятора можна надавати різні вхідні дані та перевіряти працездатність та ефективність написаного коду безпосередньо в IDE, перевіряючи вихідні дані, що відображаються у вікні виводу компілятора.

– При виникненні помилок чи некоректному оформленні коду, IDE показує попередження та пропозиції щодо їхнього усунення чи виправлення в вікні виводу.

– Віддалена розробка. Одним із найпоширеніших джерел помилок у багатьох додатках є те, що середовища розробки та експлуатації не співпадають. Хоча, як правило, для розробки неможливо надати точну копію середовища експлуатації, за допомогою PyCharm можна налагоджувати свої програми,

використовуючи інтерпретатор з іншого комп'ютера, наприклад, з віртуальної машині Linux. В результаті можена використовувати той самий інтерпретатор, що й в робочому середовищі. Це дозволяє виправляти та уникати велику кількість помилок.

Всі вище перелічені пункти значно допомагають значно підвищити ефективність створення програмного забезпечення.

Короткий перелік переваги та недоліків PyCharm нижче:

а. Переваги:

- Легке встановлення.
- Простота у використанні.
- Наявність великої кількості корисних плагінів і та простота їх установки, можна навіть вивчити іншу мову програмування, або ж розробити свій плагін безпосередньо у PyCharm.
- Підтримка контролю версій Git за замовчуванням.
- Спільнота розробників, що використовують PyCharm надзвичайно велика, тому можна легко вирішувати запитання чи проблеми з використання IDE чи Python.
- Безкоштовна професійна версія для студентів за певних умов.
- Безкоштовний пробний період професійної версії для будь-кого, з можливістю його продовження за умови виконання завдань академії JetBrains.

б. Недоліки

- Урізаний (проте достатній для розробки лише на Python) функціонал безкоштовної версії.
- Досить дорога професійна версія.
- Часті проблеми з використанням та налаштуванням venv у новачків.

Як було сказано раніше PyCharm – є однією з найкращих, якщо не найкращою, спеціалізованою, повнофункціональною та універсальною IDE для програмування на Python. З величезною масою можливостей, котрі значно зекономлять час, полегшуючи рутинну роботу.

3.2.3 Google Speech API

GSR API – це технологія, широко поширена в різних галузях досліджень і для різних мов. Це продукт компанії Google. Що дозволяє використовувати голосовий пошук на основі теорії розпізнавання мовлення.

Ця технологія інтегрована в смартфони та комп'ютери з можливістю розпізнавання мовлення. Влітку 2011 року Google інтегрувала мовленнєву технологію у свою пошукову систему (Google Search). На ПК ця технологія підтримується тільки браузером Google Chrome. Його також підтримують смартфони під управлінням операційної системи Android та iOS.

Спочатку Google Voice Search підтримував лише короткий запит довжиною до 35-40 слів. Для відправки запиту необхідно було вмикати та вимикати мікрофон. Ця функція все ще знаходиться в рядку пошуку Google; вам просто потрібно натиснути мікрофон, щоб почати. Але в лютому 2013 року в Chrome була додана можливість безперервного розпізнавання мовлення, таким чином голосовий пошук Google перетворився на мовленнєве введення.

З травня 2014 року став можливим доступ до API. Голосовий пошук Google представлено багатьма популярними сервісами: Google, YouTube, Yahoo, DuckDuckGo, Bing, Wolfram|Alpha, Wikipedia тощо. Є можливість додавати власні пошукові системи. На даний момент навіть є розширення, яке додає кнопку введення голосової інформації для сайтів, які використовують пошукові форми HTML5. Для роботи з додатками необхідно використовувати мікрофон.

Для використання технології Google Voice Search необхідно виконати POST-запит на адресу з аудіоданими у форматі «.flac» або «.spx». Тоді WAVE-файли повинні розпізнаватися будь-якою програмою. GSR API здебільшого схожий на Dragon Mobile SDK від Nuance, але не має обмежень щодо кількості запитів на день. Розробники Google використовують глибоку нейронну мережу для розпізнавання ключової фрази «Окей, Google». ASR Linux також базується на GSR API.

Перетворення мовлення в текст має три основні методи розпізнавання мовлення. Котрі наведено нижче:

Синхронне розпізнавання (REST і gRPC) надсилає аудіодані до GSR API, виконує розпізнавання цих даних і повертає результати після обробки всього аудіо.

Синхронні запити на розпізнавання обмежені аудіоданими тривалістю 1 хвилину або менше.

Асинхронне розпізнавання (REST і gRPC) надсилає аудіодані до GSR API та ініціює тривалу операцію. Використовуючи цю операцію, ви можете періодично опитувати результати розпізнавання. Використовуючи асинхронні запити можна розпізнавати аудіодані будь-якої тривалості до 480 хвилин.

Розпізнавання у реальному часі (тільки gRPC) виконує розпізнавання аудіоданих, що надаються в рамках двонаправленого потоку gRPC. Потокові запити розроблені для цілей розпізнавання в реальному часі, наприклад, для запису живого звуку з мікрофона. Розпізнавання потокового потоку забезпечує проміжні результати під час запису звуку, дозволяючи відображати результат, наприклад, коли користувач все ще говорить.

Механізм GSR API підтримує різноманітні мови та діалекти. Щоб обрати мову (і національний або регіональний діалект) аудіо необхідно ввести відповідний ідентифікатор в полі коду мови конфігурації запиту.

Google Speech API зручний. Він досить зручний та швидкий завдяки величезних обчислювальних потужностях, також плюсом є відсутність обмежень на кількість запитів на день.

Основна перевага GSR API у порівнянні з іншими ASR — висока точність розпізнавання (завдяки величезним бібліотекам баз даних та обчислювальним потужностям) і швидкість розпізнавання мовлення.

3.2.3 RTX Voice

NVIDIA RTX Voice (рисунок 3.5) — це новий плагін розробки NVIDIA, який використовує графічні процесори NVIDIA RTX та їх можливості штучного інтелекту для видалення відволікаючих фонових шумів від ваших трансляцій, голосових чатів та віддалених відеоконференцій. Це дозволяє користувачам «виходити в ефір» або приєднуватися до зустрічі, не турбуючись про небажані звуки, як-от голосний набір тексту на клавіатурі або інші навколишні шуми в шумному навколишньому середовищі. RTX Voice також усуває фоновий шум від

програвачів у гучному оточенні, що робить вхідний звук чистішим, а отже і легшим для розуміння.

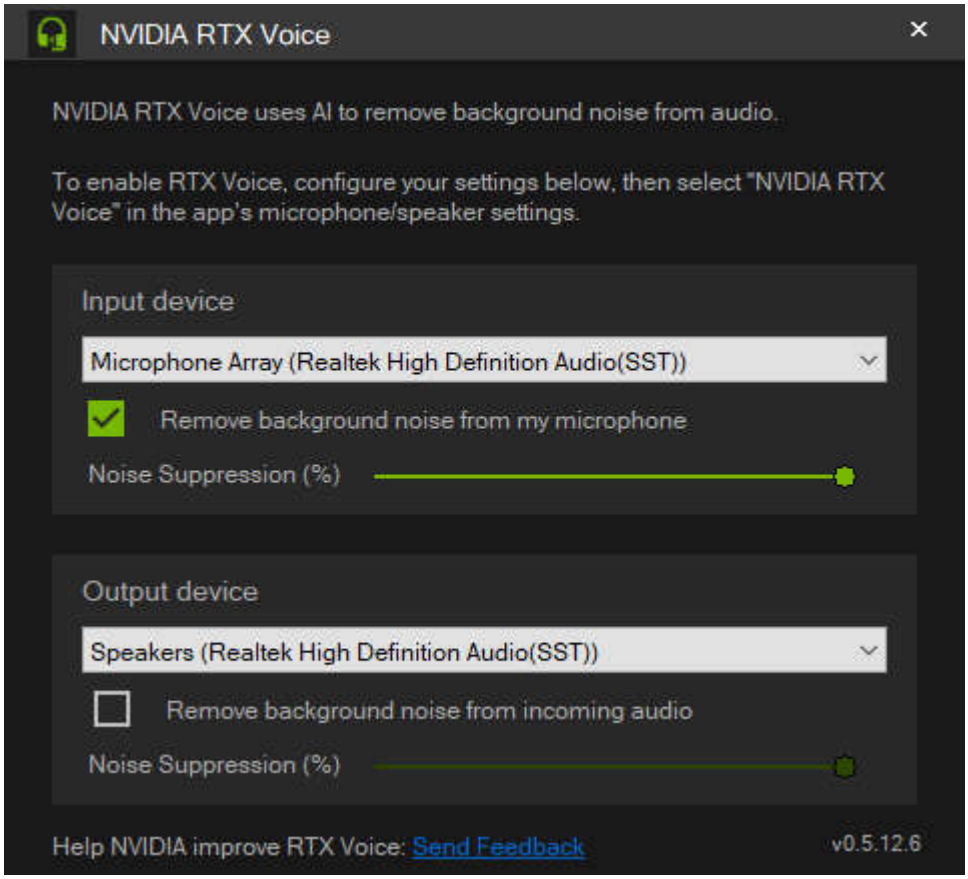

Рисунок 3.5 — NVIDIA RTX Voice

RTX Voice на даний час знаходиться на стадії бета тестування. На жаль через закритість коду не має можливості провести аналіз застосованих підходів ANN.

RTX Voice додає віртуальний мікрофон і динаміки у систему. Надалі їх можна вибрати в найбільш популярних додатках для прямих трансляцій, ігор чи відеоконференцій, включаючи, але не обмежуючись [27]:

- Battle.net Chat
- Discord
- Google Chrome
- OBS Studio
- Skype
- Slack
- Steam Chat
- Streamlabs

- Teams
- Telegram
- Twitch Studio
- WebEx
- XSplit Broadcaster
- XSplit Gamecaster
- Zoom

Хоча розробники в основному і позиціонують програму, як націлену на використання в розважальних програмах чи для простого спілкування її можливо впровадити для використання у системах ASR, для усунення шумів і покращення якості та коректності розпізнавання у місцях з сильним зашумленням.

3.2.4 Audacity

Audacity (рисунок 3.6) можна використовувати для виконання низки потреб у редагуванні аудіо. Вона відмінно підходить для покращення підкастів, запису музики, створення звукових доріжок для відео або перетворення аудіофайлів з одного формату в інший.

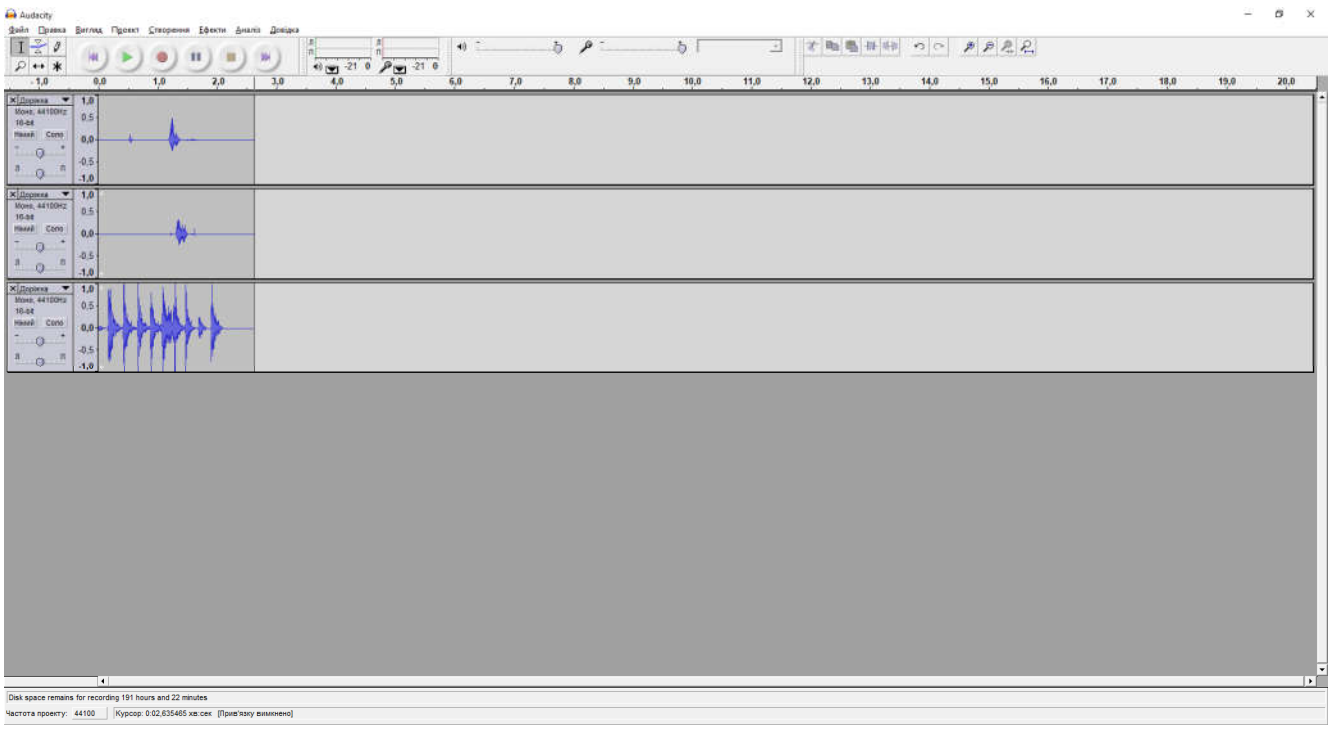

Рисунок 3.6 — Audacity

Audacity — це програмне забезпечення для редагування аудіо з відкритим вихідним кодом, яке можна безкоштовно завантажити для Windows, Mac і Linux. У мене немає сильного досвіду в редагуванні аудіо, але я зміг швидко зрозуміти основні концепції роботи з програмою та зробити необхідні записи та порівняння для подальшого тестування їх системою. Щодо всього, що я не зміг зрозуміти, наприклад, видалення відлуння чи білого шуму, в документації добре описано послідовність необхідних дій, що для цього потрібно зробити.

Програма підтримує чималу кількість аудіоформатів, також з кількох доріжок можна об’єднати все в один файл, користь цього полягає в тому, що можна штучно додати до голосу будь-які шуми та протестувати працездатність розроблюваного методу. Є можливість редагувати всі звукові доріжки одночасно, або кожну окремо.

Багатоколійна здатність Audacity: Існує багато різних ефектів, котрі можна додати, для покращення звуку, на мою думку «Зниження шуму» є найбільш корисним, проте в плані розробки даного методу найкориснішим є комбінування доріжок для детальнішого тестування методу.

Ще можна налаштувати рівні звуку, низькі та високі частоти, а також налаштувати гучність для усієї звукової доріжки, або окремих її частин.

Audacity не просто редагує, а й записує аудіо, це універсальний інструмент. Записані за допомогою мікрофону чи відтворення ПК можна конвертувати майже до будь-якого аудіо формату. Даний засіб необхідний для запису аудіо в файли з віртуального пристрою RTX, це зумовлено складністю розпізнавання потокового мовлення в librosa, та проведення експериментів з порівнянням амплітудних хвиль та спектрограм.

3.2.5 Librosa

Librosa спочатку була розроблена для обробки відносно коротких фрагментів записаного аудіо, як правило, тривалістю не більше кількох хвилин. Хоча це й описує більшість популярної музики (початковою ціллю була дана область застосування), проте це вельми погане рішення для багатьох інших форм аудіо даних, особливо тих, що зустрічаються в біоакустиці та акустиці навколишнього

середовища. У цих умовах аудіосигнали зазвичай тривають кілька годин, якщо не днів чи тижнів. Цей викликає запитання: Як і чи взагалі можна обробляти довгі аудіофайли за допомогою librosa? Далі буде описано інтерфейс потоку, прийнятий у librosa починаючи з версії 0.7, включаючи деякі відомості про загальний дизайн бібліотеки та конкретне рішення даної проблеми.

Перш ніж переходити до деталей того, як обробляються великі файли, потрібно зрозуміти модель даних librosa більш загально. Від початку створення librosa розробниками було прийнято рішення покладатися лише на типи даних *numpy*, а не розробляти більш структуровану модель об'єктів. Це рішення було мотивовано кількома факторами, серед, але не обмежуючись якими були:

- простота виконання,
- простота використання,
- простота взаємодії з іншими бібліотеками та синтаксична подібність до попередніх реалізацій на основі MATLAB, а також теоретичні (математичні) визначення.

Це означає, що замість об'єктно-орієнтованого коду (рисунок 3.7) використовується більш процедурний стиль (рисунок 3.8).

```
x = not_librosa.Audio(some parameters)  # an object of type not_librosa.Audio
melspec = not_librosa.feature.MelSpectrogram(x)  # an object of type not_librosa.feature.MelSpectrogram
```

Рисунок 3.7 — Об'єктно-орієнтований код

```
y, sr = librosa.load(some parameters)  # a numpy array
melspec = librosa.feature.melspectrogram(y, sr)  # another numpy array
```

Рисунок 3.8 — Процедурно стилізований код

В результаті дані з librosa досить легко перемістити в інші застосунки на Python. З цього виходить, що наявність власного об'єктного інтерфейсу для аудіо та функцій заважала б, навіть якщо б це спростило деякі дизайнерські рішення.

Проблема великих аудіофайлів криється у обмеженнях numpy, а саме типу numpy.ndarray: описані вище підходи (об'єктний і процедурний) мають свої плюси і мінуси. Переваги процедурного підходу перераховані вище, але одним із недоліків, котрий успадковано від numpy, являється те, що всі вхідні дані повинні

бути створені, перш ніж можна буде створити melspec. Зрештою, це обмеження типу numpy.ndarray, який розроблений контейнером для суміжних областей пам’яті фіксованої довжини з узгодженими основними типами даних (наприклад, int або float). Для коротких записів котрі являються найчастішим випадком у librosa – це добре: записи зазвичай поміщаються в пам'ять та відомі заздалегідь. Однак, коли аудіо, для аналізу, триває годину чи більше, або є потоковим з пристрою вводу, ndarray не являється підходящим типом контейнера. Розробникам було про це відомо, проте вони вирішили оптимізувати роботу для частого використання та розібратися з наслідками пізніше. Якби для Librosa був обраний об’єктивний інтерфейс, цю проблему можна було б вирішиити різними способами. Наприклад, абстрагувати логіку індексації за часом, щоб дані завантажувалися лише тоді, коли їх запитують, наприклад *Audio.get_buffer(time=SOME_NUMBER, duration=SOME_NUMBER).* Або ж використати внутрішній стан об’єкта для підтримки буфера в пам’яті, не завантажуючи весь запис із пам’яті, та надати інтерфейс для пошуку певної позиції часу в сигналі. Різноманітні бібліотеки реалізують такі рішення та чудово працюють, проте мають певну додаткову складність інтерфейсу і можуть обмежити взаємодію [28].

Рішення, до якого підійшли у версії 0.7, — це використання Python генераторів. Замість завантаження усього аудіо сигналу цілком, використовується звуковий файл для створення послідовних фрагментів сигналу, які потім передаються назад користувачеві.

Функції генератора в Python дозволяють оголосити функцію, яка веде себе як ітератор, тобто її можна використовувати в циклі for.

Тож обробка на основі блоків дозволяє легко застосувати деякі, але не всі функції librosa до великих аудіофайлів. Хоча, в принципі, це можна застосувати до потокового запису в прямому ефірі, у librosa немає стабільної реалізації, на яку можна покладатися.

Librosa підтримує наступні функції:

– Декодування Вітербі: накладає тимчасове згладжування на кадрові передбачення стану (рисунок 3.9).

– Пресети налаштувань: дозволяє використовувати пакет пресетів для зміни параметрів за замовчуванням для librosa, таких, як частота вибірки, кількість вибірок між кадрами, кількість вибірок на кадр у STFT-подібних аналізах. Вони допомагають, коли виникає потреба змінити ці значення відповідно до програми, але виконувати це послідовно під час кожного виклику функції дещо незручно. Попередні налаштування дозволяють легко зробити все це відразу, охоплюючи усе відразу, а саме: модуль і всі виклики функцій перевизначаючи аргументи за замовчуванням.

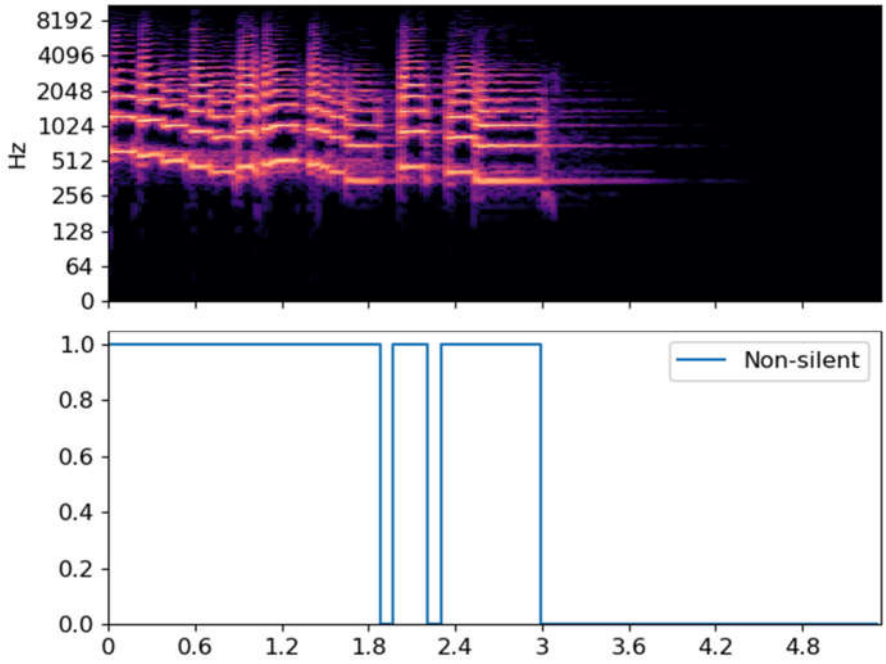

Рисунок 3.9 — Декодування Вітербі [29]

– Початки надпотоку: цей алгоритм покращує точність виявлення початку за наявності вібрато (рисунок 3.10).

– Синхронізація музики з динамічним викривленням часу (DTW): дана функція використовує DTW для синхронізації музики, за допомогою librosa. Наприклад з двох записів перших тактів знаменитої брас-секції у виконанні Стіві Уандера «Sir Duke». Через різницю в темпі перший запис триває приблизно 7 секунд, а другий запис приблизно 5 секунд. Функція може знайти узгодження між цими двома записами за допомогою DTW(рисунок 3.11).

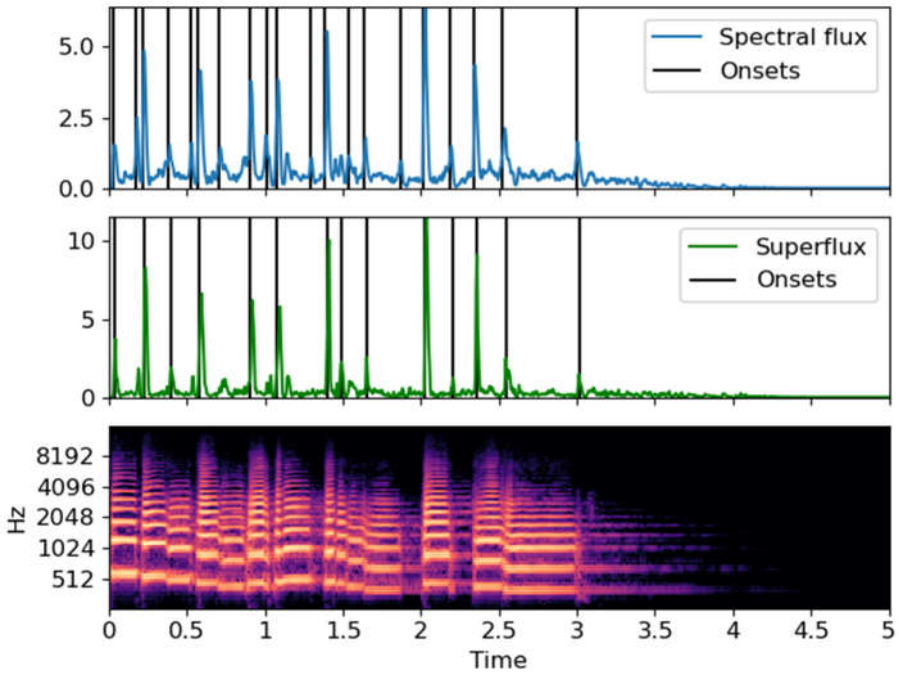

Рисунок 3.10 — Початки надпотоку [29]

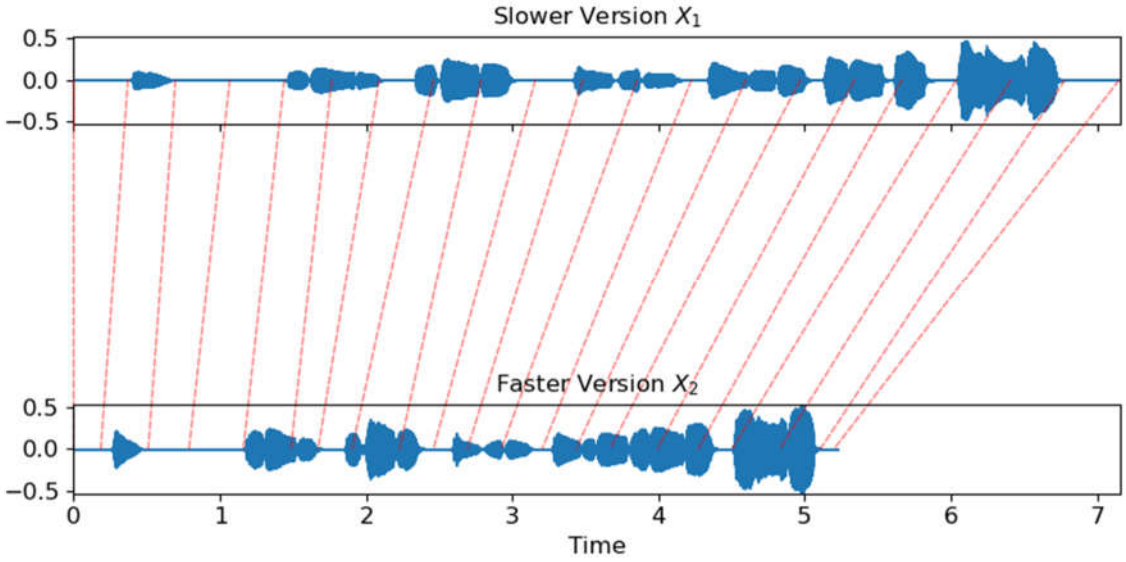

Рисунок 3.11 — Синхронізація музики з використанням DTW [29]

– Вокальне розділення (рисунок 3.12): функція відповідає за просту техніку відокремлення вокалу від супутніх інструментів. Функція заснована на методі «REPET-SIM» від Rafii та Pardo [30], але нелокальна фільтрація перетворюється на м’яку маску за допомогою фільтрації Вінера. Вона подібна до методу м’якого маскування, використаного Fitzgerald [31], але на практиці є дещо більш стабільною.

– Гармоніко-ударне розділення джерела (рисунок 3.13): ця функція розділює аудіо сигнал на його гармонічні та ударні компоненти. Функція реалізована за допомогою оригінального підходу Fitzgerald [32] на основі медіанної фільтрації, та його розширенням на основі маржі завдяки Dreidger, Mueller and Disch [33].

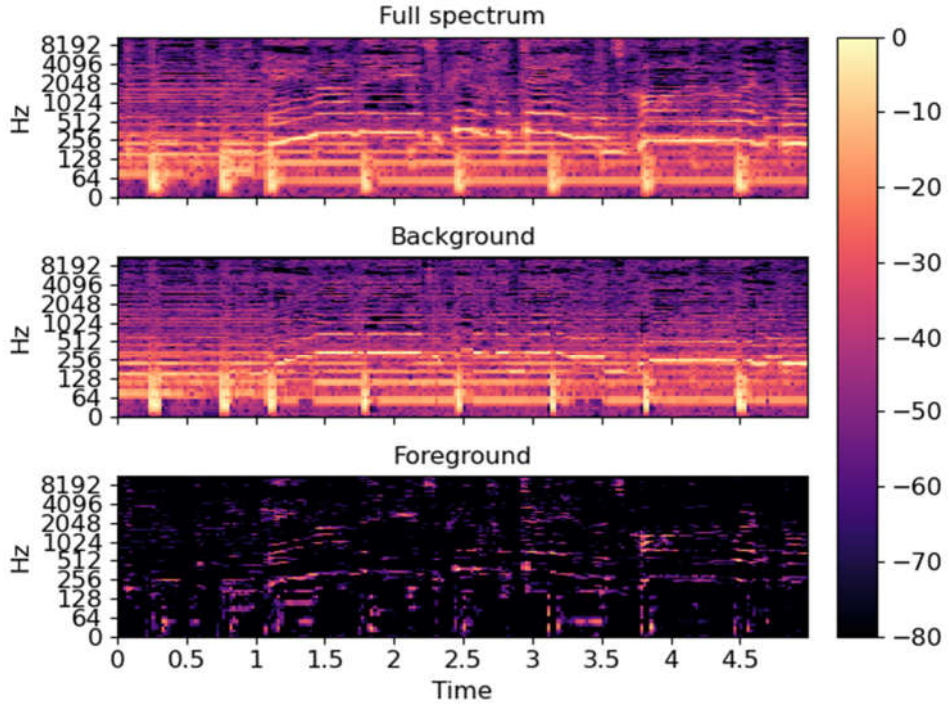

Рисунок 3.12 — Виділення вокалу [29]

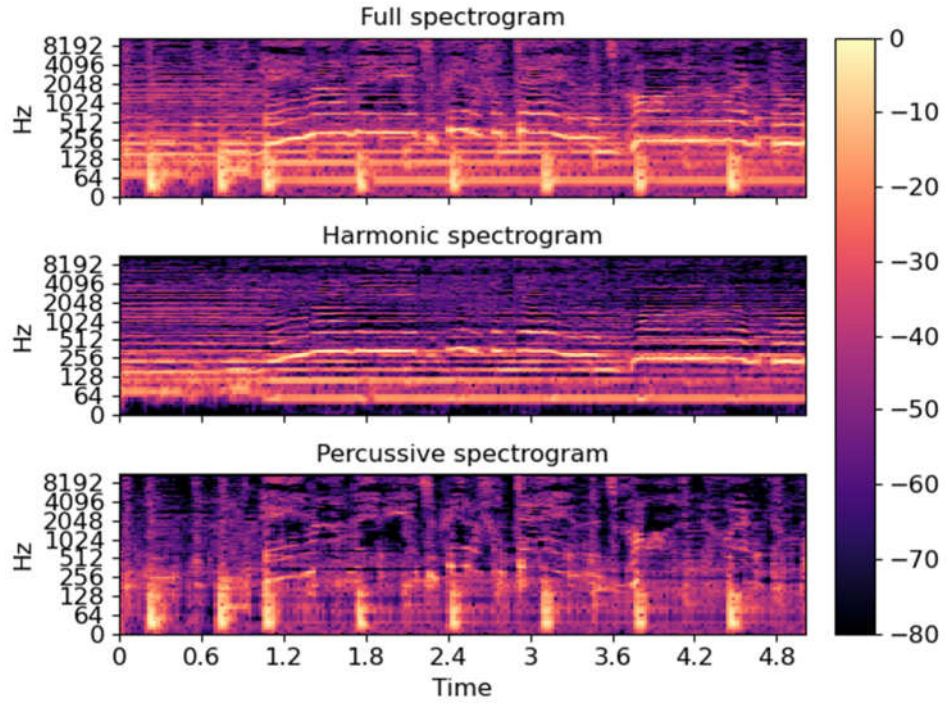

Рисунок 3.13 — Гармоніко-ударне розділення джерела [29]

3.3 Експериментальне дослідження методу розпізнавання голосових команд

З метою дослідження ефективності даного методу, було проведено експериментальне дослідження на основі існуючої ASR без використання розробленого методу, та з його допомогою, методом статистичного експерименту, порівняння котрий здійснюється на підставі статистичного порівняння графіків даних звукових хвиль.

На рисунку 3.14 зображено графік звукових хвиль при вводі слова «word» з шумом пилососа в фоні на другій доріжці.

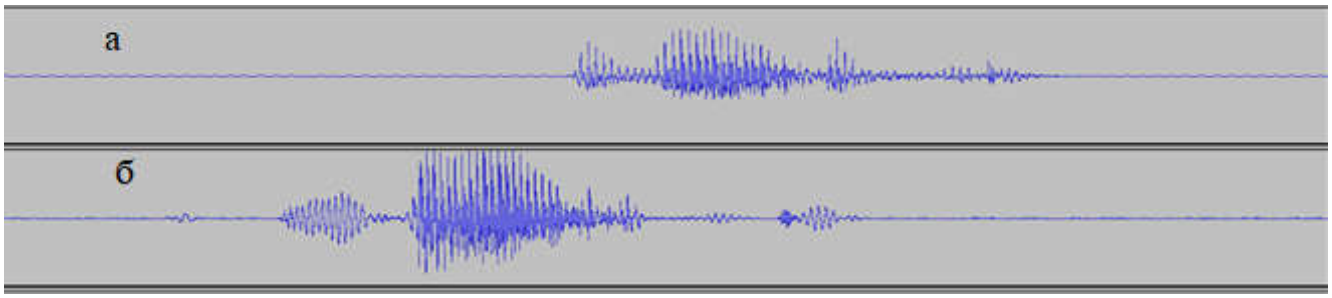

Рисунок 3.14 — Звукові хвилі без попередньої обробки

а) На перший погляд різниця між цими доріжками відсутня, проте на звуковій доріжці відсутній шум пилососа, тоді, як на наступній він присутній, що помітно за невеличкими коливаннями

б) він значно помітніший, що ускладнює обробку мови для ASR.

Тоді, як на рисунку 3.15 зображено графік звукових хвиль при вводі слова «word» з шумом пилососа в фоні на другій доріжці проте з попередньою обробкою RTX Voice.

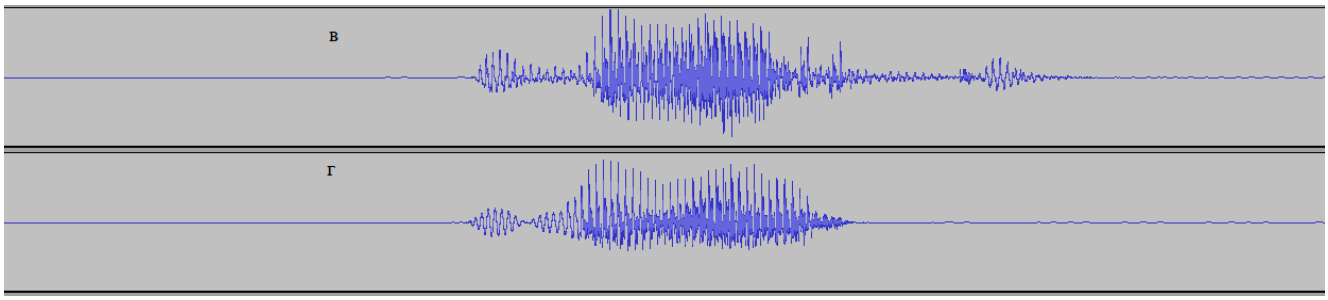

Рисунок 3.15 — Звукові хвилі з попереднью обробкою

в) Тут відображено той же ввід тільки з попередньою обробкою, при відсутності шуму пилососа у даному варіанті звук голосу звучить дуже чисто, що значно покращує результати розпізнавання мовлення.

г) Проте при увімкненні пилососа активувалися нейронні мережі для фільтрації зайвих шумів, навіть трохи занадто, тому, що голос став трохи гіршим через надмірну фільтрацію ніж в умовах звичайної тиші, проте все одно є значно кращим аніж результат доріжки б.

На рисунку 3.16 зображено безпосередньо ввід слова «word» з розташованим на відстані 40 см телевізором.

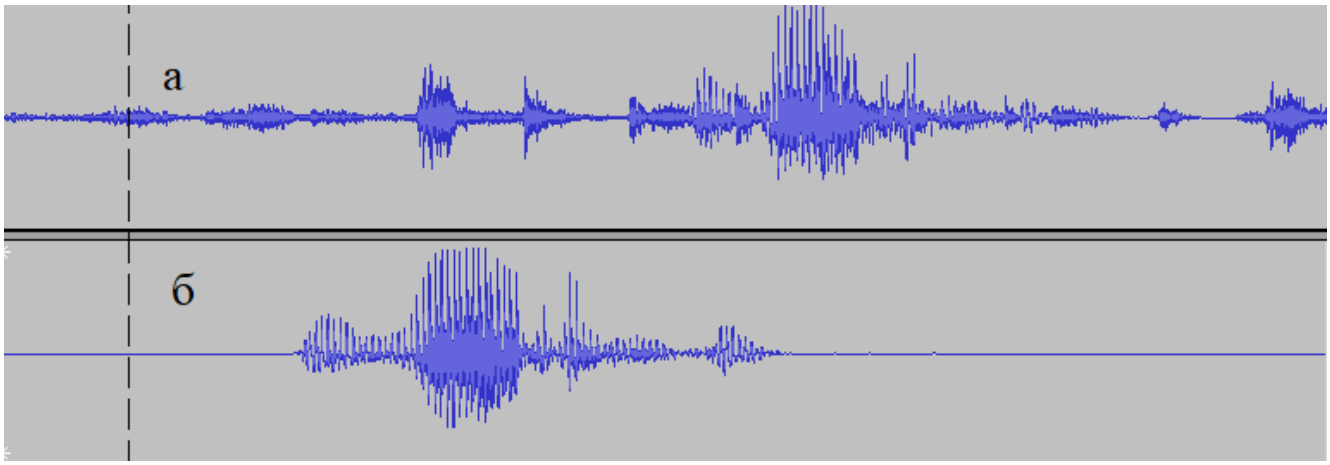

Рисунок 3.15 — Звукові хвилі без шумоусунення а, та з ним б

а) В даному випадку навіть неозброєним оком видно явну відсутність шумоусунення, через те що грала музика з речитативом ASR так і не змогла розпізнати цей фрагмент.

б) Що не можна сказати про б випадок, де нейронні мережі справилися з усуненням речитативу на фоні дуже добре та розпізнавання пройшло абсолютно без проблем.

ВИСНОВКИ

В даній кваліфікаційній роботі проведено дослідження існуючих систем і методів розпізнавання голосових команд та вводу за допомогою штучних нейронних мереж, а також розроблено новий метод розпізнавання голосових команд з шумоусуненням на базі програмно-апаратного підходу, за допомогою якого покращено функціональні можливості існуючої системи розпізнавання голосових команд а саме Google ASR, що дозволяє проводити розпізнавання голосових команд більш чітко навіть при фонових шумах певної гучності, таких, як: шум пилососа, телевізора з вихідною потужністю динаміків 40Вт.

Для досягнення мети кваліфікаційної роботи отримано наступні результати:

- розглянуто та проаналізувано існуючі системи та їх методи розпізнавання звукових даних;
- змодельовано рішення задачі розпізнавання голосових команд в системах розпізнавання звукових даних;
- запропонувано метод, що дозволяє зменшити негативний вплив шумів на системи та підвищити коректність розпізнавання даних у зашумленому середовищі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zou J., Han Y., So S.S. (2008) Overview of Artificial Neural Networks. In: Livingstone D.J. (Eds) Artificial Neural Networks. Methods in Molecular Biology™, vol 458. pp. 14-22
2. И. П. Голямина. Звук. Физическая энциклопедия. URL: www.femto.com.ua/articles/part_1/1222.html
3. GetAClass - Физика в опытах и экспериментах. Источники звука. URL: https://youtu.be/nFcmTTT9yiE
4. Шум. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Шум
5. ДСТУ 2325-93 Шум. Терміни та визначення.
6. Білий шум. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Білий_шум
7. Рожевий шум. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Рожевий_шум
8. What Is Brown Noise? Live Science. URL: https://www.livescience.com/38547-what-is-brown-noise.html
9. What is Gray Noise? Techopedia. URL: https://www.techopedia.com/definition/27898/gray-noise
10. Dynamic Time Warping. HandWiki. URL: https://handwiki.org/wiki/Dynamic_time_warping
11. Suryo Wijoyo, "Speech Recognition Using Linear Predictive Coding and Artificial Neural Network for Controlling Movement of Mobile Robot", International Conference on Information and Electronics Engineering IPCSIT vol.6, IACSIT Press, pp.179-183, Singapore, 2011.
12. Neural network. Wikipedia. URL: https://en.wikipedia.org/w/index.php?title=Neural_network
13. Viterbi algorithm. Wikipedia. URL: https://en.wikipedia.org/wiki/Viterbi_algorithm
14. Mel scale. Wikipedia. URL: https://en.wikipedia.org/wiki/Mel_scale
15. Штучна нейронна мережа. Вікіпедія. URL: https://uk.wikipedia.org/wiki/%D0%A8%D1%82%D1%83%D1%87%D0%BD%D0%

B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0

16. Мережа радіальних базисних функцій. Матеріал з Вікіпедії. URL: https://uk.wikipedia.org/wiki/Мережа_радіальних_базисних_функцій

17. Palaz D., Collobert R. Analysis of CNN-Based Speech Recognition System Using Raw Speech as Input. In Proceeding of Interspeech 2015

18. G. Gnaneswari, S. R. VijayaRaghava, A. K. Thushar, Dr.S.Balaji, Recent Trends in Application of Neural Networks to Speech Recognition. International Journal on Recent and Innovation Trends in Computing and Communication, Volume: 4. Issue 1, pp 18 - 25

19. Palaz, D., Doss, M., & Collobert, R. (2015). Convolutional Neural Networks-based continuous speech recognition using raw speech signal. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4295-4299.

20. Chorowski J., Jaitly N. Towards better decoding and language model integration in sequence to sequence models. 2016. arXiv:1612.02695 [cs.NE]

21. Palaz D., Collobert R., Doss M. M. End-to-end phoneme sequence recognition using con-volutional neural networks. 2013. arXiv:1312.2137 [cs.LG]

22. TIMIT — Wikipedia. URL: https://en.wikipedia.org/wiki/TIMIT

23. Abdel-Hamid O., Abdel-Rahman M., Hui J., Li D., Penn G., Yu D. IEEE/ACM Transactions on Audio, Speech and Language Processing. Volume 22. Issue 10: Convolutional neural networks for speech recognition. October 2014. pp. 1533–1545

24. Graves A, Jaitly N, Mohamed AR. Hybrid speech recognition with deep bidirectional LSTM. In: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE; 2013

25. Miao Y, Gowayyed M, Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In: Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on. IEEE; 2015

26. Что такое Boilerplate code? Stackoverflow. URL: https://ru.stackoverflow.com/a/583346

27. NVIDIA RTX Voice: Setup Guide GeForce News. URL: https://www.nvidia.com/en-us/geforce/guides/nvidia-rtx-voice-setup-guide/

28. McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. “librosa: Audio and music signal analysis in python.” In Proceedings of the 14th python in science conference, pp. 18-25. 2015.

29. Advanced examples — librosa 0.8.1 documentation. URL: https://librosa.org/doc/latest/advanced.html

30. Z. Rafi, B. Padro, MUSIC/VOICE SEPARATION USING THE SIMILARITY MATRIX. In Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Pages 583-588

31. Fitzgerald, D. (2012) Vocal separation using nearest neighbours and median filtering. 23rd IET Irish Signals and Systems Conference, Maynooth. 28-29th. June 2012.

32. Fitzgerald, D. "Harmonic/percussive separation using median filtering." In 13th International Conference on Digital Audio Effects (DAFX10). 2010.

33. Driedger. J. & Müller. M. & Disch, S. (2014). Extending Harmonic-Percussive Separation of Audio Signals.