МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

НА ТЕМУ:

Нейромережевий застосунок синтезу голосу

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Аналітика даних»**

Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи АнД- 41

Лаврій Руслан Романович
(прізвище та ініціали)

Керівник Іларіонов О. Є.
(прізвище та ініціали)

к.т.н., доцент
(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту рішенням кафедри *інтелектуальних технологій*
Протокол № 13 від 05.06.2023 р.
зав. кафедри_____ доц. Іларіонов О.Є.

Київ – 2023

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій
Кафедра інтелектуальних технологій
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
інтелектуальних технологій
Іларіонов О.Є.
_____

"___"_____2023 р.

ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ
_____Лаврію Руслану Романовичу _____
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи): «Нейромережевий застосунок синтезу голосу» затверджена протоколом засідання кафедри від «11» листопада 2022 р. № 4
2. Термін здачі студентом закінченого проекту (роботи) _____ травня 2023 року
3. Вихідні дані до проекту (роботи)
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
Дослідження та аналіз предметної області, що пов'язана із методами синтезу голосу на основі тексту, програмна реалізація застосунку, тестування розробленого застосунку
5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)
Мета дослідження дипломного проекту (1 слайд), актуальність завдання (1 слайд), постановка задачі (1 слайд), функціональний аналіз (1 слайд), опис навчального набору даних (1 слайд), алгоритм вирішення задачі (1 слайд), схема архітектури програмного модулю та опис інструментів програмної реалізації (1 слайд), аналіз результатів роботи застосунку (2 слайди), висновки (1 слайд).

6. Дата видачі завдання 15 лютого 2023 року

Керівник _____ / Іларіонов О.Є. /
(підпис) (ПІБ)

Завдання прийняв до виконання _____ / Лаврій Р.Р. /
(підпис) (ПІБ)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1.	Обговорення з керівником постановки завдання, підбір літератури, огляд існуючих рішень	15.02.2023 – 01.03.2023	
2.	Аналіз постановки задачі, формалізація задачі, аналіз літературних джерел, написання розділу 1	02.03.2023 – 20.03.2023	
3.	Проектно-технологічна реалізація нейромережного застосунку синтезу голосу.	21.03.2023 – 11.04.2023	
4.	Розробка та тестування нейромережного застосунку синтезу голосу.	12.04.2023 – 15.05.2023	
5.	Робота над оформленням пояснювальної записки та презентаційних матеріалів	16.05.2023 – 29.05.2023	

Студент-дипломник _____ / Лаврій Р.Р. /
(підпис) (ПІБ)

Керівник випускної
кваліфікаційної роботи _____ / Іларіонов О.Є. /
(підпис) (ПІБ)

РЕФЕРАТ

Кваліфікаційна робота складається зі вступу, 3 розділів, 17 підрозділів, висновків, списку використаних джерел (29 джерел) та 4 додатків. Загальний обсяг роботи 62 сторінок. Робота містить 2 таблиць та 26 рисунків.

Актуальність теми. В останні роки дослідження в області синтезу речі (TTS) з клонуванням голосу набувають все більшої популярності. Клонування голосу дозволяє створювати синтезовану мову, яка максимально наближена до оригінального голосу мовця. Це має широкий спектр застосувань, включаючи аудіокниги, розмовні асистенти, озвучення фільмів та інші додатки, де натуральність мовлення є критичними факторами.

Метою кваліфікаційної роботи є розробка нейромережевого застосунку для синтезу голосу з використанням TTS-системи Tacotron 2. Застосунок буде забезпечувати здатність створювати синтезовану мову, що точно відтворює особливості оригінального голосу говорячої особи.

Об’єктом дослідження є технології синтезування голосу з використанням TTS-системи Tacotron 2.

Предметом дослідження є розробка нейромережевого застосунку для синтезу голосу з використанням TTS-системи Tacotron 2.

Результати роботи. Нейромережевий застосунок синтезу голосу на основі алгоритму TTS – Tacotron.

Ключові слова: клонування голосу, синтез речі, TTS, Tacotron 2, програмний модуль, нейромережевий застосунок.

ABSTRACT

The qualification work consists of an introduction, 3 chapters, 17 subsections, conclusions, a list of used sources (29 sources) and 4 appendix. The total volume of work 62 pages. The work contains 2 tables and 26 figures.

Actuality of theme. In recent years, speech-to-speech (TTS) research with voice cloning has become increasingly popular. Voice cloning allows you to create a synthesized speech that is as close as possible to the speaker's original voice. This has a wide range of applications, including audiobooks, conversational assistants, film dubbing, and other applications where naturalness of speech is critical.

The purpose of the qualification work is to develop a neural network application for voice synthesis using the Tacotron 2 TTS system. The application will provide the ability to create synthesized speech that accurately reproduces the features of the speaker's original voice.

The object of research is voice synthesis technology using the Tacotron 2 TTS system.

The subject of research is the development of a neural network application for voice synthesis using the Tacotron 2 TTS system.

Work results. Neural network application of voice synthesis based on the TTS algorithm - Tacotron.

Keywords: voice cloning, thing synthesis, TTS, Tacotron 2, software module, neural network application.

Зміст

Скорочення та умовні позначення

TTS – Текст в мовлення

SPSS - Статистичний параметричний синтез мови

HMM – Прихована марківська модель

LSTM – Довга короткочасна пам’ять

CNN – згорткова нейронна мережа

DNN – розгортаюча нейронна мережа

RNN – рекурентна нейронна мережа

ВСТУП

У сучасному світі синтез речі (TTS) з клонуванням голосу набуває все більшої популярності. Здатність створювати синтезовану мову, що максимально наближена до оригінального голосу мовця, має широкий спектр застосувань, включаючи аудіокниги, розмовні асистенти, озвучення фільмів та інші додатки, де натуральність мовлення є критичним фактором. В цьому контексті розробка програмного застосунку для клонування голосу стає актуальною задачею.

Метою даного дослідження є розробка програмного застосунку для клонування голосу з використанням TTS-системи Tacotron 2. Цей модуль має на меті створити синтезовану мову, що точно відтворює особливості оригінального голосу мовця. Шляхом аналізу технологій клонування голосу та використання Tacotron 2, я маю за мету розробити ефективний та натуральний програмний модуль, який забезпечить високу якість синтезованої речі.

Об'єктом дослідження є технології клонування голосу з використанням TTS-системи Tacotron 2. Розробка програмного застосунку є предметом дослідження, який передбачає аналіз, проектування та реалізацію програмного рішення для синтезу голосу з використанням Tacotron 2.

Результати даного дослідження можуть мати велике значення в розвитку галузей, що використовують синтез речі. В Україні це може сприяти розвитку аудіокниг, розмовних асистентів та інших застосувань, де висока якість та натуральність голосу є ключовими факторами. Крім того, впровадження такого програмного модуля може сприяти українській літературі та культурі, надаючи можливість озвучення текстів у натуральному голосі говорячої особи. На основі проведеного критичного аналізу та порівняння з відомими вирішеннями проблеми я переконаний в актуальності та доцільності розробки програмного модуля клонування голосу з використанням TTS-системи Tacotron 2. Його впровадження може привести до покращення якості синтезованої речі та

забезпечити користувачам більш реалістичний та натуральний досвід аудіовізуальних застосувань.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД НЕЙРОМЕРЕЖЕВОГО ЗАСТОСУНКУ

1.1 Аналіз існуючих методів синтезу мовлення

Статистичний параметричний синтез мови

Статистичний параметричний синтез мови (SPSS) відноситься до групи керованих даними методів TTS, які з'явилися наприкінці 90-х. У SPSS зв'язок між характеристиками, обчисленими на основі вхідного тексту, і вихідними акустичними характеристиками вивчається за допомогою статистичної генеративної моделі (акустичною моделлю). Таким чином, повна структура SPSS (рис. 1.1) також включає конвеєр для вилучення функцій із тексту для синтезу, а також систему, здатну реконструювати форму аудіосигналу з акустичних характеристик, створених акустичною моделлю (така система називається вокодером). На відміну від акустичної моделі, ці дві частини фреймворку можуть бути повністю розроблені без використання статистичних методів.

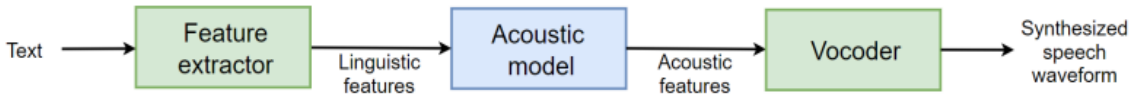


Рисунок 1.1 Загальна структура SPSS

Роль засобу виділення ознак полягає в тому, щоб надати дані, які більш вказують на те, як звучить мова, виробленої моделі. Мовлення є складним процесом, і пряме введення символів у слабку акустичну модель є неефективним. Надання додаткових функцій від техніки обробки природної мови (NLP) може значно зменшити обсяг завдання, яке має вивчати акустична модель. Однак це може призвести до проблем, особливо коли йдеться про натуральність для рідкісних або невідомих слів. Дійсно, створені вручну евристики не повністю характеризують усі тонкощі розмовної мови. З цієї причини виділення ознак

також можна виконувати за допомогою навчених моделей. Межа між екстрактором ознак і акустичною моделлю може стати розмитою, особливо для глибоких моделей. Фактично, тенденція, яка є загальною для всіх областей, де глибокі моделі випередили традиційні методи машинного навчання, полягає в тому, що вилучення ознак складається з меншої кількості евристик, оскільки дуже нелінійні моделі стають здатними працювати на вищих рівнях абстракції.

Загальною технікою виділення ознак є створення фреймів, які інтегрують навколишній контекст ієрархічно. Наприклад, фрейм на складовому рівні може включати слово, яке входить до його складу, його положення в слові, сусідні склади, фонеми, що утворюють склад, ... Лексичний наголос і акцент окремих складів можна передбачити за допомогою статистичної модель, такої як дерево рішень. Для кодування просодії можна використовувати набори правил, таких як ToBI (Beckman and Elam, 1997).

Причина, чому акустична модель не передбачає безпосередньо форму звукового сигналу, полягає в тому, що аудіо важко моделювати: і звукові сигнали, як правило, дуже нелінійні. більш зрозумілим способом є частотно-часова область. Спектрограми є більш гладкими та набагато менш щільними, ніж їхні хвилеподібні аналоги. Вони також мають перевагу двовимірності, що дозволяє моделям краще використовувати просторові зв’язки. На жаль, спектрограма — це представлення форми хвилі з втратами. Немає унікальної функції зворотного перетворення, і отримання такої, яка дає результати природного звучання. Коли мова йде про мову, ця генеративна функція називається вокодером.

Прихована Марківська модель

Новішою версією є алгоритм заснований на прихованій Марківській моделі (рис. 1.2). Цей підхід, полягає в кластеризації лінгвістичних особливостей, витягнутих із вхідного тексту, за допомогою дерева рішень і навчання для кожного кластера. Моделі Маркова мають завдання створити

розподіл коефіцієнтів спектрограми, їх похідну, другу похідну та двійковий прапор, який вказує, які частини згенерованого аудіо мають містити голос. За допомогою алгоритму генерації параметрів максимальної правдоподібності коефіцієнти спектрограми вибираються з цього розподілу та в кінцевому підсумку подаються до вокодера. Можна модифікувати створений голос шляхом кондиціонування моделі Маркова на динаміку або налаштування згенерованих параметрів мови за допомогою методів адаптації чи інтерполяції.

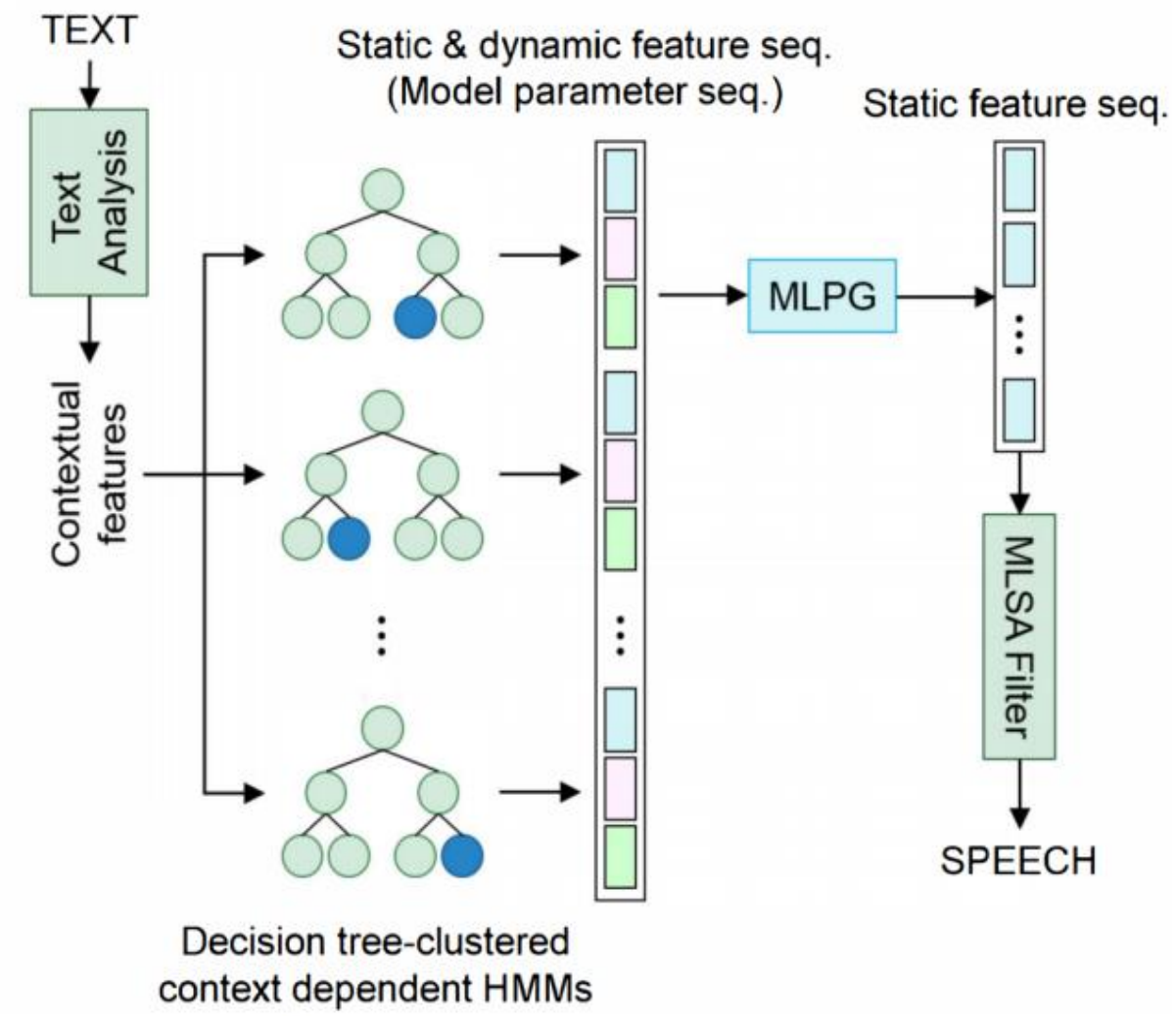


Рисунок 1.2 Підхід на основі алгоритму прихованій Марковській моделі

Удосконалення цієї структури пізніше були внесені глибокими нейронними мережами прямого зв’язку у результаті прогресу апаратного та програмного забезпечення. Пропонується повністю замінити HMM з кластером

дерева рішень на користь DNN. Дана структура має кращу ефективність даних, оскільки навчальний набір більше не фрагментований у різних кластерах контекстів. Вона демонструє покращення якості мовлення.

WaveNet

Поява WaveNet зробила суттєвий прорив у TTS. WaveNet — це глибока згорточна нейронна мережа, яка для необробленої аудіосигналу моделює розподіл одного зразка умовно до попередніх. Таким чином, можна безпосередньо генерувати аудіо, прогнозуючи семпли по одному за допомогою авторегресії. WaveNet використовує стеки одновимірних розширених звивин із коефіцієнтом розширення, що експоненціально зростає з глибиною шару, що забезпечує дуже велике сприйнятливе поле та сильну нелінійність, необхідну для моделювання необробленого аудіо. Для виконання TTS необхідна адаптація моделі до лінгвістичних особливостей. Таким чином, WaveNet діє і як акустична модель, і як вокодер.

Було проведено дослідження, в якому порівнюють WaveNet зі старішим параметричним підходом і з конкатенативним підходом (рис. 1.3). Параметричний підхід є системою на основі LSTM, тоді як інший є конкатенативною системою вибору одиниць, що керується HMM.

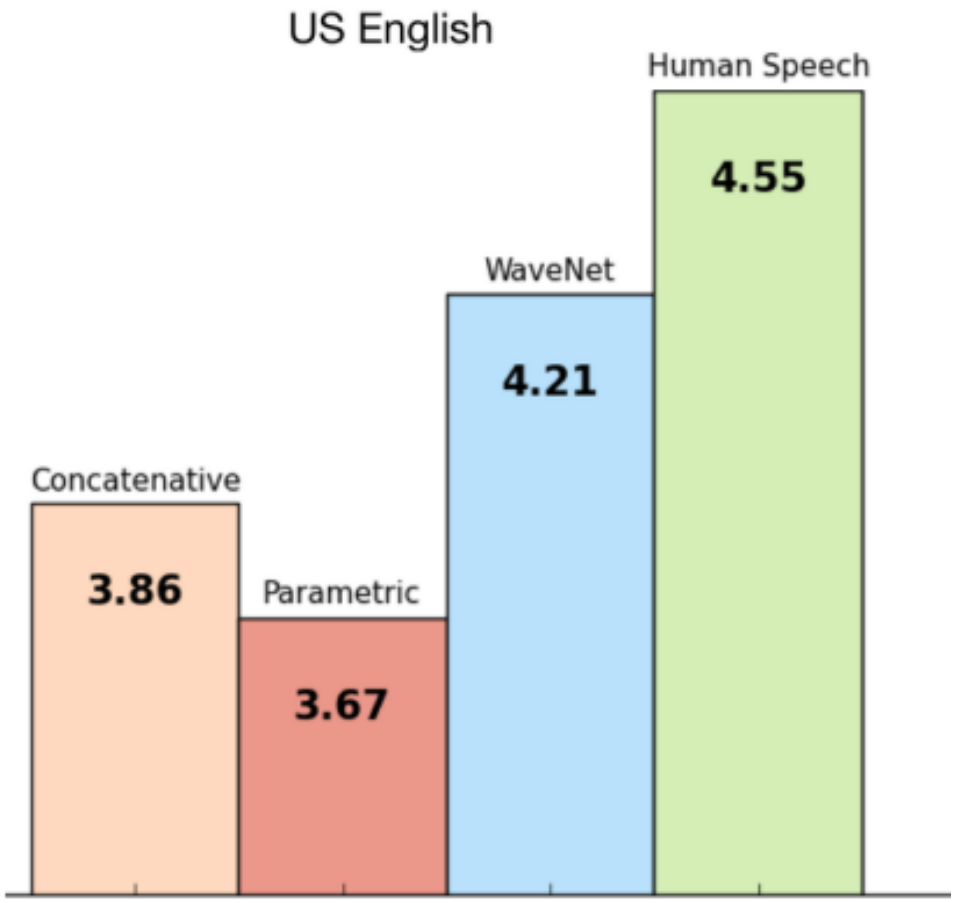


Рисунок 1.3 Порівняння продуктивносты Wavenet в порівнянні з параметричним методом, конкатенивним і людською мовою

Tacotron

Згодом вийшла Tacotron, а після неї і Tacotron 2. Архітектура другої версії залишається кодера-декодера з увагою, хоча внесено кілька змін до типу шарів. Основною відмінністю є використання WaveNet як вокодера. На тому ж наборі даних результат був набагато кращий. Автори провели опитування з 7 відповідями для порівняння з людською мовою. Результати показали що людська мова зовсім трохи краща ніж створена на Tacotron2 (рис 1.4).

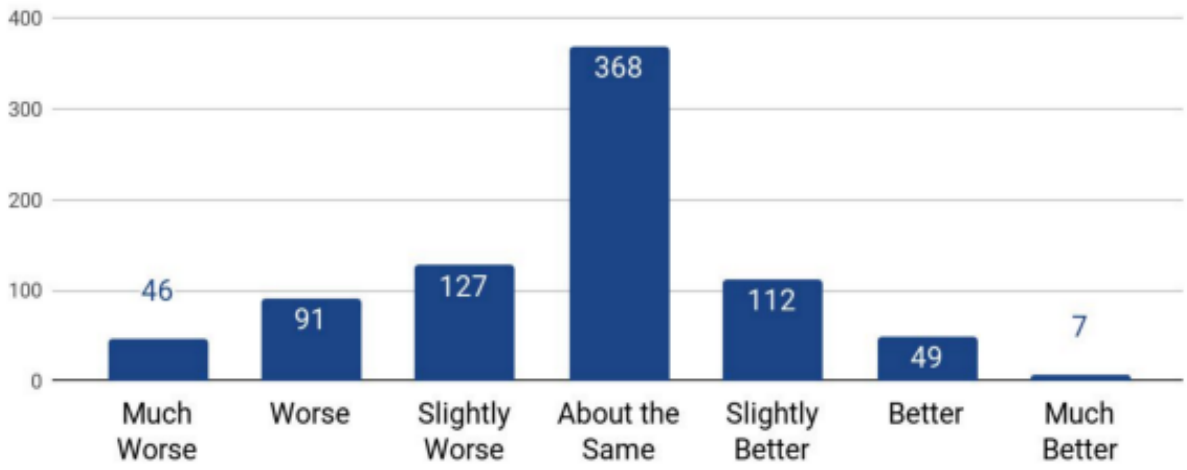


Рисунок 1.4 Порівняння Tacotron2 із оригінальними записами.

1.2 Аналіз актуальності розробки

Актуальність розробки програмного модулю клонування голосу з використанням TTS-системи Tacotron 2 обумовлена кількома чинниками:

1. Зростання популярності синтезу речі: У сучасному цифровому світі збільшується попит на якісні та натуральні голосові додатки та сервіси. Синтез речі на основі клонування голосу є одним з найефективніших способів створення такої синтезованої речі, яка максимально наближена до оригінального голосу.
2. Застосування в різних галузях: Технології клонування голосу мають широкий спектр застосувань, включаючи аудіокниги, розмовні асистенти, озвучення фільмів та інші мультимедійні проекти. Розробка програмного модулю клонування голосу стане цінним ресурсом для розширення можливостей цих галузей та забезпечення високої якості синтезованої речі.
3. Поліпшення користувацького досвіду: Забезпечення натурального та виразного голосового взаємодії з комп'ютерними системами та пристроями є однією з ключових цілей. Розробка програмного модулю клонування голосу дозволить покращити користувацький досвід, забезпечивши гладку та натуральну голосову взаємодію з програмними продуктами.

4. Потенційна економія ресурсів: За допомогою програмного модулю клонування голосу можна забезпечити автоматизований процес створення синтезованої речі з використанням наявного оригінального голосу. Це може зменшити затрати на процес запису та озвучення речі, що є особливо важливим для великих проектів та компаній.

5. Дослідницький потенціал: Розробка програмного модулю клонування голосу є важливим кроком у вивченні та розвитку області синтезу речі та голосових технологій. Цей модуль може послужити основою для подальших наукових досліджень та розробок в цій сфері, що призведе до появи нових інновацій та покращення існуючих методів клонування голосу.

Отже, розробка програмного модулю клонування голосу є актуальною та перспективною задачею, яка відповідає сучасним потребам ринку і може принести значну користь у багатьох галузях, сприяючи поліпшенню якості синтезованої речі та забезпечуючи натуральну та виразну голосову взаємодію з користувачами.

1.3 Аналіз області застосування

Область застосування технологій клонування голосу є широкою і має значний потенціал у різних сферах. Деякі з них включають:

- Аудіокниги та озвучення текстів: Застосування клонування голосу дозволяє створювати синтезовані аудіокниги та озвучувати текстові матеріали. Це сприяє поліпшенню доступності для людей з вадами зору та забезпечує більш реалістичне та переконливе аудіовізуальне досвіду.
- Розмовні асистенти та віртуальні помічники: Клонування голосу може бути використане для створення персоналізованих голосових інтерфейсів

для розмовних асистентів та віртуальних помічників. Це дозволяє створити більш людяний та інтерактивний досвід взаємодії з такими системами.

- Комп’ютерні ігри та віртуальна реальність: Клонування голосу може забезпечити більш реалістичне аудіовізуальне середовище в комп’ютерних іграх та віртуальній реальності. Це дозволяє створювати персонажів з унікальними голосами та доповнює іммерсивний геймплей.
- Мовленнєва терапія та реабілітація: Застосування клонування голосу може бути корисним у мовленнєвій терапії та реабілітації людей з порушеннями мовлення. Синтезований голос, який точно відтворює їхні особливості, може бути використаний для вправ та тренувань з поліпшення мовленнєвих навичок.
- Озвучення фільмів та мультимедійних продуктів: Клонування голосу може бути використане для озвучення фільмів, мультфільмів та інших мультимедійних продуктів. Це дозволяє дотримуватись оригінального голосу персонажів або створювати нові голосові характеристики для різних ролей.
- Медична сфера: Технології клонування голосу можуть бути корисними у медичній сфері. Наприклад, вони можуть допомогти людям з вадами голосу відновити свої голосові можливості за допомогою синтезованого голосу, що максимально наближений до їхнього оригінального голосу. Крім того, ці технології можуть знайти застосування в дослідженнях голосових хвороб та розумінні механізмів розладів голосу.
- Мультимедійні платформи та соціальні мережі: Використання клонування голосу може зробити мультимедійні платформи та соціальні мережі більш цікавими та привабливими для користувачів. Вони можуть надати можливість створювати персоналізовані голосові повідомлення, фільтри та ефекти голосу, що допоможуть користувачам виразніше комунікувати та викладати свої ідеї.
- Електронна комерція та маркетинг: Застосування технологій клонування голосу можуть бути цінними в електронній комерції та маркетингу. Синтезований голос, що наближений до реального, може бути використаний для створення рекламних аудіо- та відеоматеріалів, розповіді про товари та послуги,

що сприяє покращенню взаємодії з клієнтами та збільшенню їхньої зацікавленості.

Ці області застосування є лише деякими прикладами використання технологій клонування голосу. З розвитком цих технологій можуть з’явитись нові можливості та перспективи в інших галузях, що вимагають високоякісного синтезу речі та голосової інтеракції.

1.4 Аналіз невирішених питань синтезу мовлення

Під час аналізу технологій клонування голосу та вивчення TTS-системи Tacotron 2 можуть виявитися наступні проблемні моменти:

1. Якість синтезованої речі: Один з основних викликів полягає у досягненні максимально натурального та реалістичного звучання синтезованого голосу. Необхідно вирішити проблему артифактів, неприродних модуляцій та інших аспектів, що можуть впливати на якість звучання.
2. Персоналізація голосу: Існуючі системи клонування голосу часто мають обмежені можливості персоналізації, тобто важко досягти точного відтворення унікальних особливостей голосу конкретної особи. Вирішення цієї проблеми потребує розвитку методів, що забезпечують більш точне клонування голосу з урахуванням індивідуальних характеристик.
3. Обмеження в навчальних даних: Для ефективного клонування голосу необхідні великі обсяги якісних навчальних даних. Однак, отримання достатньої кількості високоякісних голосових зразків може бути проблематичним завданням. Вирішення цієї проблеми передбачає пошук та розробку методів для ефективного використання обмежених навчальних даних.
4. Розмір моделей та обчислювальна складність: Технології клонування голосу, особливо з використанням глибокого навчання, можуть потребувати великого обсягу пам'яті та високої обчислювальної потужності. Це може

створити обмеження для розгортання модулю на обмежених платформах або пристроях з обмеженими ресурсами.

5. Відтворення емоцій та інтонацій: Вірогідність клонування голосу також полягає у здатності передавати емоції та інтонації, які є невід'ємною частиною мовлення. Забезпечення точного відтворення інтонаційних нюансів та емоцій вимагає додаткових досліджень та розробки.

6. Адаптація до різних голосових умов: Залежно від умов запису та акценту говорячої особи, які можуть відрізнятися в різних аудіозаписах, можуть виникати проблеми з точністю та якістю клонування голосу. Розробка алгоритмів та технік, що забезпечують адаптацію до різних голосових умов, є важливим аспектом дослідження.

Ці проблеми підкреслюють необхідність досліджень і розробки програмного модулю клонування голосу з метою вирішення цих викликів та поліпшення якості та натуральності синтезованої речі.

1.5 Аналіз вхідної і вихідної інформації

Вхідна інформація:

1. Голосові дані мовця: Для клонування голосу потрібні голосові дані оригінальної мовця. Ці дані можуть бути записи голосу з різних джерел, таких як аудіозаписи, звукові файли або спеціально створені голосові набори.

2. Мовленнєві зразки: Для досягнення максимальної точності та натуральності синтезованого голосу, можуть використовуватися мовленнєві зразки, що містять текстові фрази або речення, які будуть синтезовані голосом. Ці зразки допомагають моделі Tacotron 2 вивчати зв'язок між текстом та звуком.

Вихідна інформація:

1. Синтезована мова: Основним результатом дослідження є синтезована мова, яка є відтворенням оригінального голосу мовця. Ця вихідна інформація представлена у вигляді аудіозапису або звукового файла, що містить мовленнєві фрази або речення, які були синтезовані з використанням об'єкта дослідження.

Вхідна і вихідна інформація дозволяють створювати зв'язок між оригінальним голосом мовця та синтезованою мовою, що дозволяє використовувати клонований голос у різних застосуваннях та сценаріях.

1.6 Постановка задачі

Постановка задачі для розробки програмного модулю клонування голосу включає наступні етапи:

1. Зібрати набір тренувальних даних: Збір достатнього обсягу голосових записів оригінального мовця, які будуть використовуватись для навчання моделі клонування голосу. Дані повинні бути репрезентативними та включати різні фрази та акценти.
2. Передобробка даних: Провести передобробку голосових записів для видалення шуму, нормалізації амплітуди та вирівнювання голосових функцій. Цей етап допоможе покращити якість тренувальних даних та забезпечити більш точне клонування голосу.
3. Вибір моделі клонування голосу: Розглянути різні моделі та алгоритми, що використовуються для клонування голосу, зокрема TTS-систему Tacotron 2. Визначити найбільш підходящу модель для розробки програмного модулю.

4. Тренування моделі: Застосувати обрану модель для тренування на підготовлених тренувальних даних. Виконати ітеративний процес навчання моделі з метою покращення її здатності клонування голосу та відтворення особливостей оригінального голосу.

5. Оцінка якості синтезованої речі: Провести оцінку якості та натуральності синтезованої речі, порівняючи її з оригінальним голосом. Використовувати як кількісні, так і якісні метрики для оцінки результатів.

6. Аналіз моделі: Провести аналіз результатів та ідентифікувати проблеми або недоліки моделі клонування голосу.

7. Документування та звітність: Зібрати всі необхідні деталі та документацію про розроблений програмний модуль, включаючи опис алгоритмів, вхідну та вихідну інформацію, технічні вимоги та інструкції щодо використання. Підготувати звіт, який відображатиме усі кроки дослідження та розробки.

Системні вимоги:

1. Підтримка операційних систем: Система повинна бути сумісною з різними операційними системами, такими як Windows, macOS, Linux і т.д.

2. Апаратні вимоги: Система повинна працювати на різних апаратних платформах з достатнім рівнем продуктивності. Це можуть бути персональні комп'ютери, сервери або мобільні пристрої з достатнім обсягом оперативної пам'яті та процесорною потужністю.

Функціональні вимоги:

1. Здатність завантажувати аудіофайли з оригінальним голосом говорячої особи в якості вхідної інформації.

2. Можливість обробки вхідного аудіофайлу для отримання репрезентації голосу, яка буде використовуватись для клонування.

3. Реалізація нейронної мережі для клонування голосу на основі отриманих репрезентацій.

4. Забезпечення здатності синтезувати голосові дані, що відповідають оригінальному голосу говорячої особи.

5. Можливість зберігання синтезованого голосу у вигляді аудіофайлу.

Нефункціональні вимоги:

1. Якість голосу: Синтезована речь повинна мати високу якість і бути натуральною, що дозволить створювати реалістичний клон голосу.

2. Швидкодія: Система повинна працювати ефективно та максимально швидко, забезпечуючи миттєвий відгук на запити користувача.

3. Масштабованість: Система повинна бути масштабованою, що дозволить обробляти великі обсяги даних.

Вимоги до окремих видів забезпечення:

1. Вимоги до апаратних компонентів: Система повинна працювати на різних апаратних платформах та використовувати доступну апаратну потужність для ефективної обробки голосових даних.

2. Вимоги до програмного забезпечення: Система повинна використовувати певне програмне забезпечення, таке як TTS-система Tacotron 2, для розробки моделі клонування голосу.

Ці вимоги визначають основні вимоги до системи клонування голосу та його компонентів, що допоможуть забезпечити ефективну та надійну роботу модуля у різних сценаріях застосування.

Висновок до першого розділу

У розділі 1 проведений аналітичний огляд нейромережевого застосунку синтезу голосу дозволив отримати важливу інформацію щодо історії синтезу мовлення, актуальності розробки, області застосування, проблемних моментів, а також вхідної і вихідної інформації та постановки задачі.

Актуальність розробки підтверджується розширенням сфери застосування, яка охоплює аудіокниги, розмовні асистенти, озвучення фільмів та багато інших додатків, де якість та натуральність голосу мають вирішальне значення.

Аналіз області застосування показав, що модуль клонування голосу може бути успішно використаний у таких галузях, як медіа, розваги, освіта, телекомунікації та інші, де потреба в точному відтворенні голосу говорячої особи є важливою.

Виявлені проблемні моменти включають складність аналізу та синтезу голосу, досягнення високої якості і натуральності речі, а також оптимізацію продуктивності системи.

Аналіз вхідної і вихідної інформації показав необхідність розробки програмного модулю, який забезпечує аналіз оригінального голосу та створення синтезованої речі, що максимально наближена до особливостей оригінального голосу.

Поставлення задачі визначило основні завдання розробки програмного модулю, які включають створення ефективного і точного алгоритму

РОЗДІЛ 2. ПРОЕКТУВАННЯ НЕЙРОМЕРЕЖЕВОГО ЗАСТОСУНКУ СИНТЕЗУ МОВЛЕННЯ

2.1 Опис застосованих нейронних мереж

Нашою задачею є написання застосунку, який буде перетворювати текст у мовлення. Текст може буде різним, може мати зміну всього лиш у одному слові, знаку пунктуації. Тобто нам не підійде зразок прикладів, нам потрібна модель. Модель – це відтворення об’єкту, що імітує певні принципи внутрішньої організації, ознаки, властивості, характеристики об’єкта дослідження. Нам потрібно виокремити головні властивості, щоб ми могли застосувати їх у подальшому.

Для створення такої моделі нам необхідна нейронна мережа Головна її задача – це створення, і навчення моделі на основі готових прикладів, або ж методом проб з порівнянням з результатами. Вони роблять це без апріорних знань. Якщо розглядати в моїй предметній області, то воно розбирає мовлення не так як людина. Людина розглядає букви, словосполучення, наголоси. А нейронна мережа у моєму прикладі повинна розглядати який набір частот робить нашу букву саме тою буквою, що робить її подовженою, протягненою, голосною в децибельному вигляді і тд.

Штучні нейронні мережі побудовані на біологічній моделі нейронних мереж. Вони містять шари штучних нейронів, які пов’язані між собою і впливають один на одного. Кожне з’єднання та нейрон мають вагу, вони в процесі навчання змінюються і утворюють модель яка подібна до нашого об’єкта.

Кожна архітектура має свою побудова, свій алгоритм роботи і проходу, і очевидно що кожна підійде під конкретну задачу. Я розгляну 3 типи нейронних мереж які використовуються для досягнення цілі.

2.1.1 Згорткові нейронні мережі(CNN)

Даний клас глибинних штучних нейронних мереж використовується зазвичай для обробки зображень, аудіо(не завжди).

CNN взяли за основу схему з’єднання нейронів зорової кори тварин. Окремі нейрони реагують на властивості в обмеженій області зорового поля, яке ще називається рецептивне поле. Якщо транслювати це на мережу, то замість того щоб будувати всю картину за раз і переносити її за раз, мережа використовує «сканер» (рис. 2.1). Він будує матриці однакового розміру, які зрушуються на одну умовну одиницю, і бере нову матрицю. Таким чином ми не обробляємо ціле «зображення» шукаючи властивості, а обробляємо багато маленьких «зображень». Даний метод дозволяє точніше будувати нашу модель, і обробляти набагато швидше.

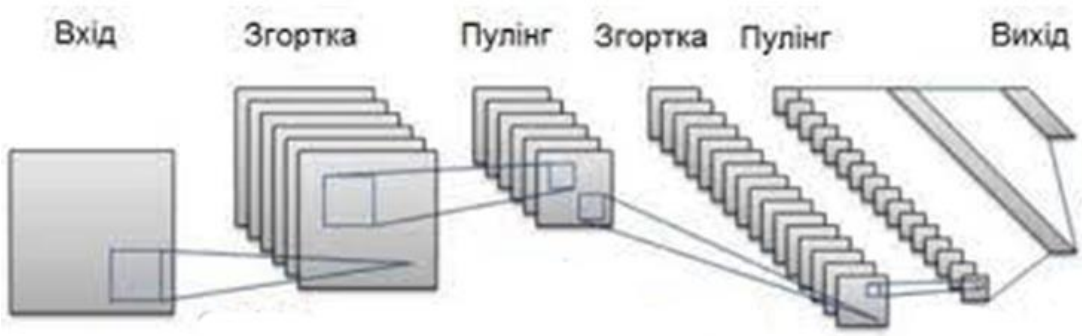


Рисунок 2.1 Схематичне зображення роботи «сканера»

«Сканер» не розбиває вхідні дані, а рухається по ньому. Ці дані подаються на згортаючі шари. У них вузли не зв’язані між всіма, а лише між сусідами. Зазвичай кількість зменшується із глибиною проходу, і як зазвичай удвічі.

Після згорткового шару йде агрегувальний шар. Даний спосіб має за мету зменшити розмірність даних. Одними з прикладами є максимізаційне агрегування, яке використовує максимальне значення з кожного з кластерів нейрону попереднього шару, або ж усереднювальне агрегування, яке використовує середнє значення кожного кластера.

Після повторень згорткових і пулінгових шарів йде повноз’єднаний шар. Він з’єднує кожен нейрон одного шару з нейроном наступного шару.

Важливою перевагою є те що для CNN потрібно відносно мало попередньої обробки в порівнянні з іншими алгоритмами. Тобто мережа навчається з фільтрами, які в інших алгоритмах потрібно було би забирати або сортувати.

2.1.2 Розгортаючі нейронні мережі(DNN)

Розгортаючі нейронні мережі, це нейронна мережа зворотня до згорткової. Задачою цієї мережі є побудова. Припустимо що ми навчили модель на згортковій мережі, і ми видобули важливі нам властивості, але у нас ще є ціль побудувати назад наш файл. Відповідно до наших даних в моделі ми навчаємо модель генерувати файли, порівнюючи її з кінцевим результатом. Всі шари є подібними до згорткової мережі, але виконують зворотню функцію. Наприклад якщо ми поєднювали нейрони і будували зв’язки, нам потрібно їх роз’єднати, потім розширити, і збільшити, і так до отримання бажаного результату.

2.1.3 Рекурентні нейронні мережі(RNN)

Рекурентні мережі – це клас мереж які орієнтовані у часі. Вони будуть зв’язок між вузлами у вигляді графа орієнтованого у часі. Нейрони отримують інформацію не лише від попередніх шарів, але й інформацію від себе з попереднього проходу. Це створює внутрішній стан мережі, що дозволяє проявляти динамічну поведінку у часі. В даній мережі дуже важливий порядок подання інформації в мережу, адже при зміні інформації будуть різні результати.

В даній мережі є одна проблема. Це зникання інформації. Інформація з часом використання функції буде зникати, а точніше ставати недостатньо інформативною.

Областю застосування рекурентних мереж застосовується також і з даними які не пов’язані з часом, а й зданими які представляються у вигляді послідовностей. Якщо наприклад дані це текст, то ваги будуть застосовуватися для попередніх елементів послідовностей, навідмінну від аудіо, які застосовують ваги для інформації яка була деякий час перед цим. Прослідкував еволюцію методів синтезу мовлення TTS. Визначив і розібрав нейронні мережі які застосовуватимуться у програмному модулі, і визначив переваги і недоліки, якщо вони будуть присутні у моїй темі. А також визначив яку попередню обробку потрібно зробити для моїх вхідних даних і реалізував програмно.

2.2 Розробка TTS архітектури

За основу було взято TTS алгоритм tacotron (рис. 2.2). В його роботі можна визначити основні три основні частини. Це кодер, синтезатор, декодер.

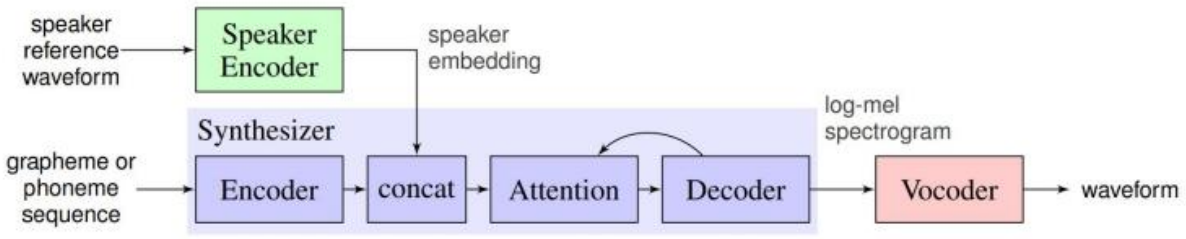


Рисунок 2.2 Структура TTS архітектури

На вхід, в кодер, який на рисунку зображений зеленим блоком, подається короткий запис мовця для клонування. Завданням кодувальника полягає щоб взяти деякий вхідний звук певного динаміка та вивести вбудування, яке фіксує «як динамік звучить».

Сині блоки представляють як високорівневий вигляд архітектури Tacotron2. Для кращого звучання було добавлено також Spectrogram Super-resolution Network (SSRN).

Даний синтезатор (рис. 2.3) складається з двох незалежних нейронних мереж: Text2Mel і SSRN. На вхід синтезатора подається текст, який ми хочемо озвучити, він проходить кодування. Також попередньо оброблений закодований запис. Text2Mel синтезатор за допомогою механізму уваги відгадує мел спектр з тексту. Результатом його роботи є відновлення розрідженого мел-спектру який був розріджений в механізмі уваги. Мережа SSRN відновлює з мел-спектру повноцінний амплітудний спектр, враховуючи пропуски кадрів і відновлюючи частоту дискретизації.

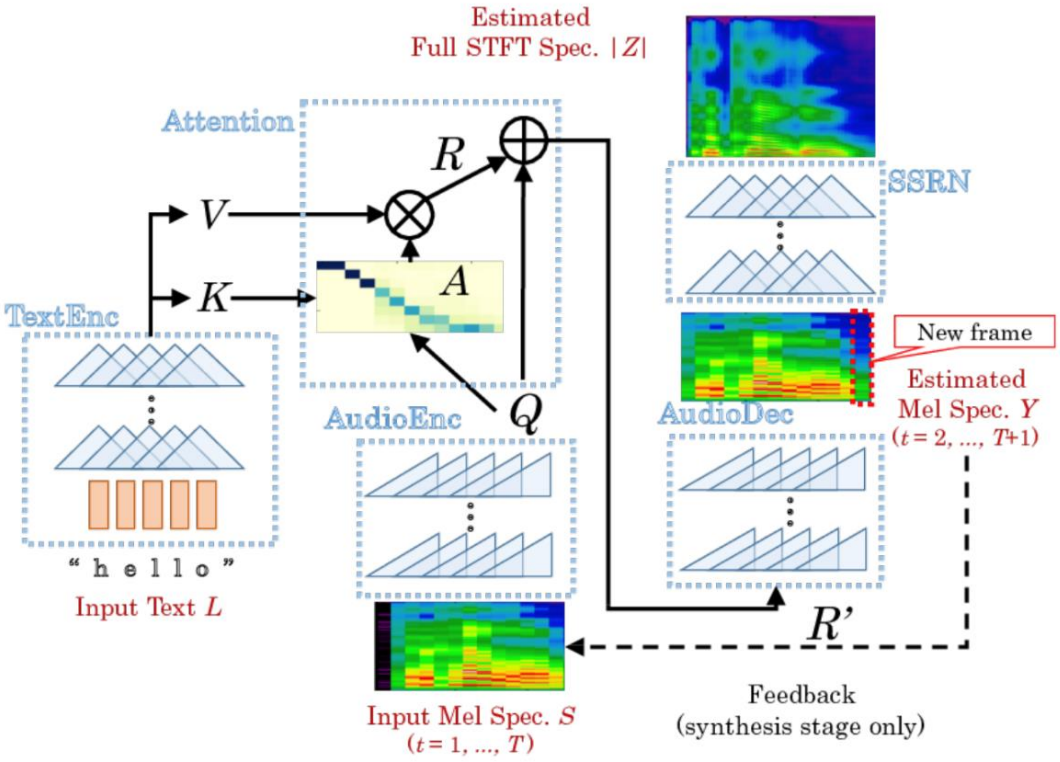


Рисунок 2.3 Схематичне зображення синтезатора

Червоним блоком алгоритму TTS є вокодер. Він призначений для синтезу мови з моделі. На його вхід подається текст який ми хочемо озвучити і також наша модель, яка згенерувалася в наслідок синтезування.

В результаті роботи програми ми отримуємо .wav файл з озвучуванням нашого диктора.

2.3 Використання Griffin-Lim алгоритму в якості вокодеру

Griffin-Lim алгоритм – це метод фазової реконструкції, який заснований на надлишковості перетворення Фур'є.

Алгоритм відтворює спектрограму комплексних значень, використовуючи формулу:

$$X^{[m+1]} = P_C\left(P_A\left(X^{[m]}\right)\right), \tag{2.1}$$

де X – спектрограма з комплексними значеннями, яка оновлюється ітеративно;

P_A – проекція на множину A;

m – індекс поточної ітерації;

A − множина спектрограм.

P_C розраховується за формулою:

$$P_C(X) = G * G^* * X, \tag{2.2}$$

де G – значення короткочасного перетворення Фур'є;

G^* - це псевдо обернене значення короткочасного перетворення Фур'є

X – спектрограма з комплексними значеннями

2.4 Перетворення Фур’є

2.4.1 Математичне представлення перетворення Фур’є

Перетворення Фур’є має вигляд:

$$f(\xi) = \int_{-\infty}^{\infty} f(x) * e^{-i*2*\pi*\xi*x}\, dx, \tag{2.3}$$

де ξ – значення частоти.

Інтегрування для всіх значень ξ дає можливим створення функції частотного спектру.

У випадках, коли ми маємо діло зі скінченною кількістю x, перетворення Фур’є можна представити у вигляді суми:

$$f(\xi) = \sum_{0}^{N-1} f(N) * e^{-i*2*\pi*\xi*x}, \tag{2.3}$$

де N – кількість елементів вектора x.

2.4.2 Математичне представлення короткочасного перетворення Фур'є

В якості основного алгоритма перетворення звукової хвилі на спектр частот в рамках роботи використовується короткочасне перетворення Фур'є. Даний алгоритм виступає модифікацією звичайного алгоритму Фур'є.

Короткочасне перетворення Фур'є має вигляд:

$$F(\tau, \xi) = \int_{-\infty}^{\infty} f(x) * w(x - \tau) * e^{-i*2*\pi*\xi*x}\, dx \quad (2.4)$$

де $F(\tau, \xi)$ представлення сигналу у часово-частотному просторі

τ – індекс часового вікна від початку сигналу;

ξ – індекс частоти в дискретному спектрі;

$f(x)$ – представлення вхідного сигналу;

$w(x - \tau)$ – віконна функція, яка використовується для зменшення спектральних витоків;

$e^{-i*2*\pi*\xi*x}$ – комплексний експоненційний сигнал, який представляє фазову інформацію сигналу.

2.5 Архітектура програмного модулю клонування голосу

Вхідні дані включають:

- Записи голосу або аудіофайли, що містять голосову інформацію, як вхід для процесу клонування;
- Транксрипції до записів, що містять вирази які вимовив мовець;

- Текстові дані, які використовуються для створення синтезованої мови.

Вихідні дані включають:

- Синтезовані голосові файли або аудіофайли.

Чинники, що впливають на модуль клонування голосу:

- Налаштування параметрів синтезу голосу;
- Доступність достатньої обчислювальної потужності для ефективної роботи модуля.

Обмеження, що існують, можуть включати:

- Часові обмеження для вхідних і вихідних файлів;
- Обмеження розміру вхідних аудіофайлів і текстових даних;
- Обмеження на якість синтезованого голосу, які можуть бути обумовлені обраними алгоритмами.

Програмний модуль описаний у вигляді IDEF0 (рис. 2.4).

Рисунок 2.4 Опис системи в нотації IDEF0

Система клонування голосу включає в себе дві основні функції:

- Побудова моделі;
- Синтезування голосу.

Функція побудови моделі включає у себе наступні підфункції:

- Попередня обробка даних;
- Мережа text2mel;
- Мережа SSRN.

Функція синтезування голосу включає у себе наступні підфункції:

- Загрузка моделі;
- Відновлення mel спектрограми з тексту;
- Перетворення в аудіофайл.

Із описаного вище функціонального аналізу побудуємо дерево функцій (рис 2.5).

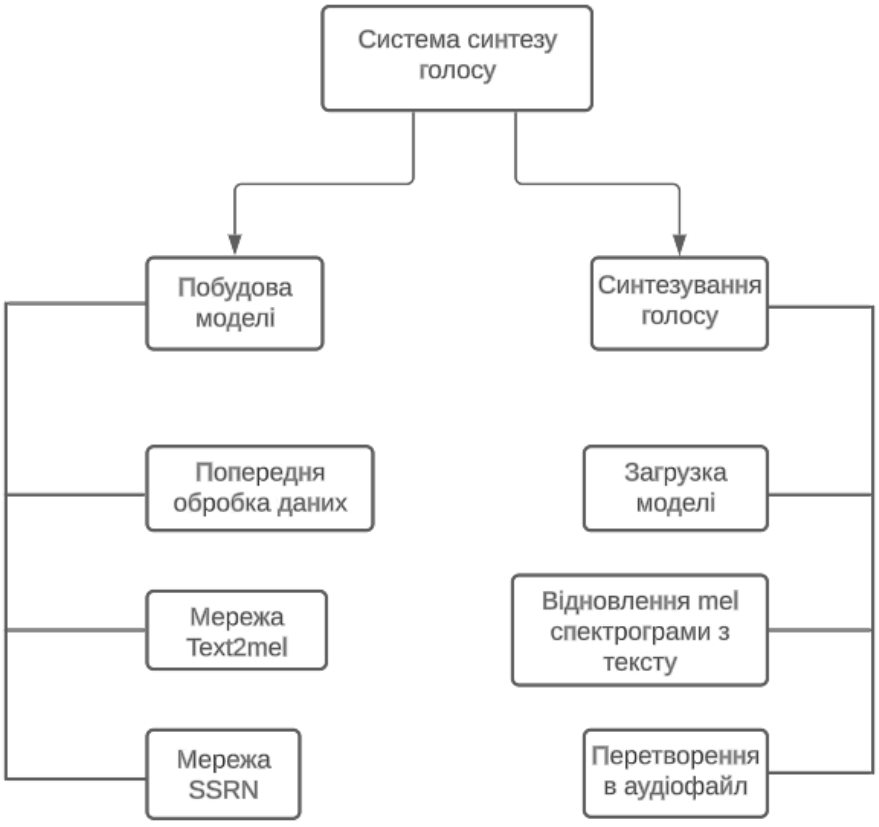


Рисунок 2.5 Дерево функцій системи

Застосунок буде мати подійно орієнтовану архітектуру (рис. 2.6). Вона має 3 вузли, кожен з яких відповідає окремій події.

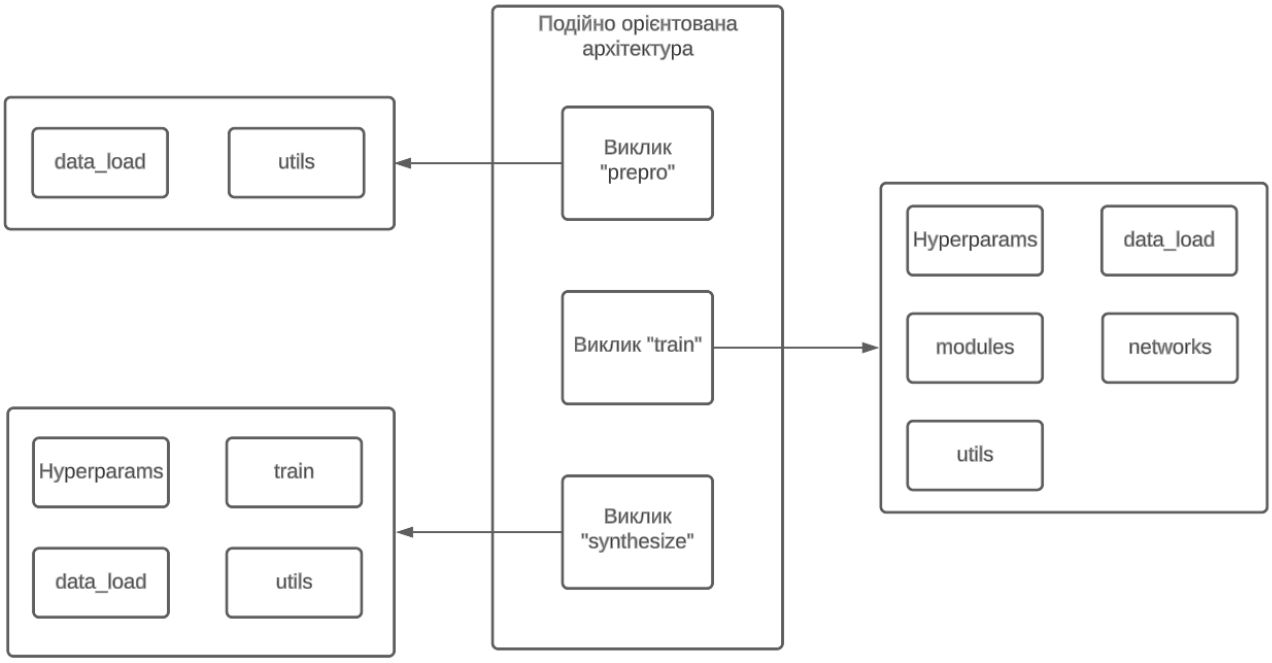


Рисунок 2.6 Компонентна архітектура застосунку

Подія «Виклик «prepro»» відповідає за виклик компонента який виконує попередню обробку даних, і складається з наступних компонентів:

- data_load: відповідає за загрузку і перетворення даних до необхідних типів;
- utils: здійснює перетворення звуку в спектрограми.

Подія «Виклик «train»» відповідає за виклик компонента який виконує навчання моделі, і складається з наступних компонентів:

- Hyperparams: компонент який містить параметри;
- data_load: перетворює дані до необхідних типів і загружає їх;
- modules: містить функції необхідні для навчання моделі;
- networks: містить нейронні мережі за допомогою яких здійснюється навчання моделі;
- utils: загружає функції обробки звуку.

Подія «Виклик «synthesize»» відповідає за виклик компонента який синтезує аудіо, і складається з наступних компонентів:

- Hyperparams: компонент який містить параметри;
- train: загружає навчальну модель;
- data_load: перетворює дані до необхідних типів і загружає їх;
- utils: містить функцію перетворення спектрограми в аудіозапис.

Для діаграми прецедентів (рис. 2.7) було визначено одного актора – «Користувач». Для нього буде доступно:

- Завантажити дані;
- Попередня підготовка даних;
- Застосування синтезатора;
- Навчання моделі;
- Декодер.

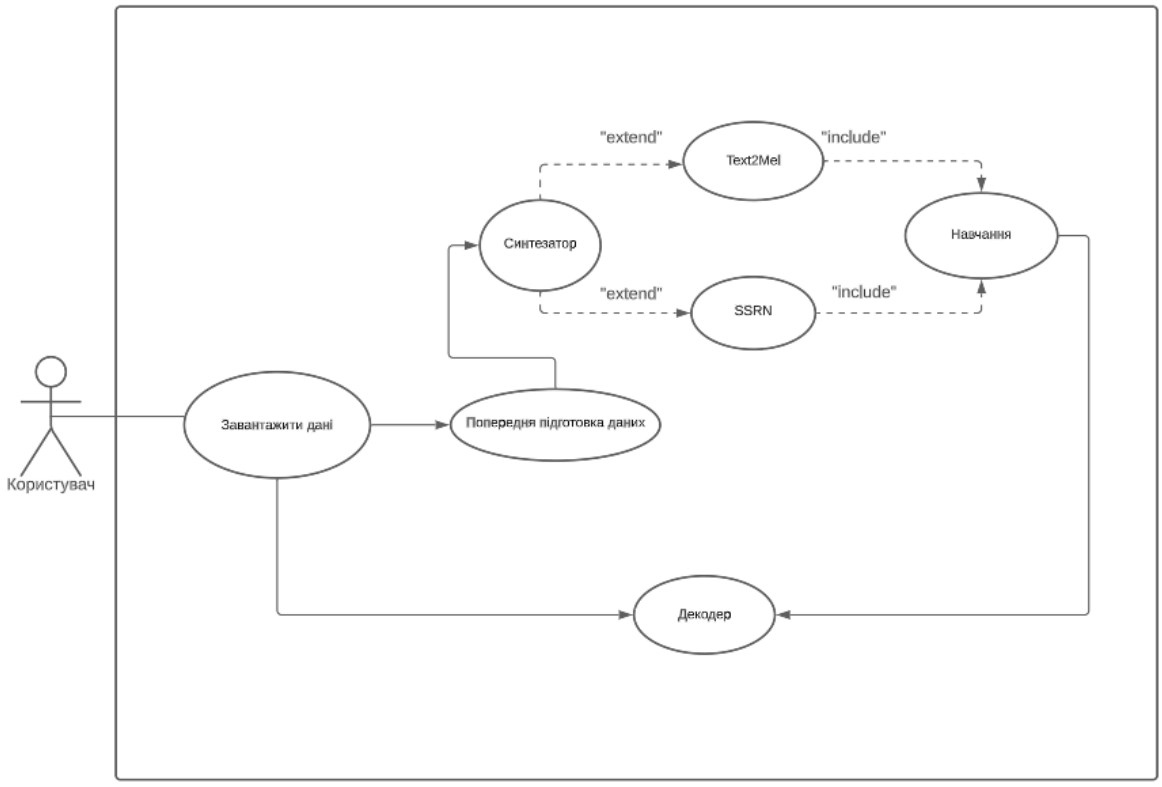


Рисунок 2.7 Діаграма прецедентів

Також для усвідомлення послідовності процесів була розроблена діаграма послідовності (рис. 2.8). Із наведеної діаграми можна зрозуміти що процес починається з попередньої обробки. Після виконання дані направляються в синтезатор, він на основі цих даних навчає модель. Декодер після виконання забирає модель і генерує аудіо і повертає його користувачу.

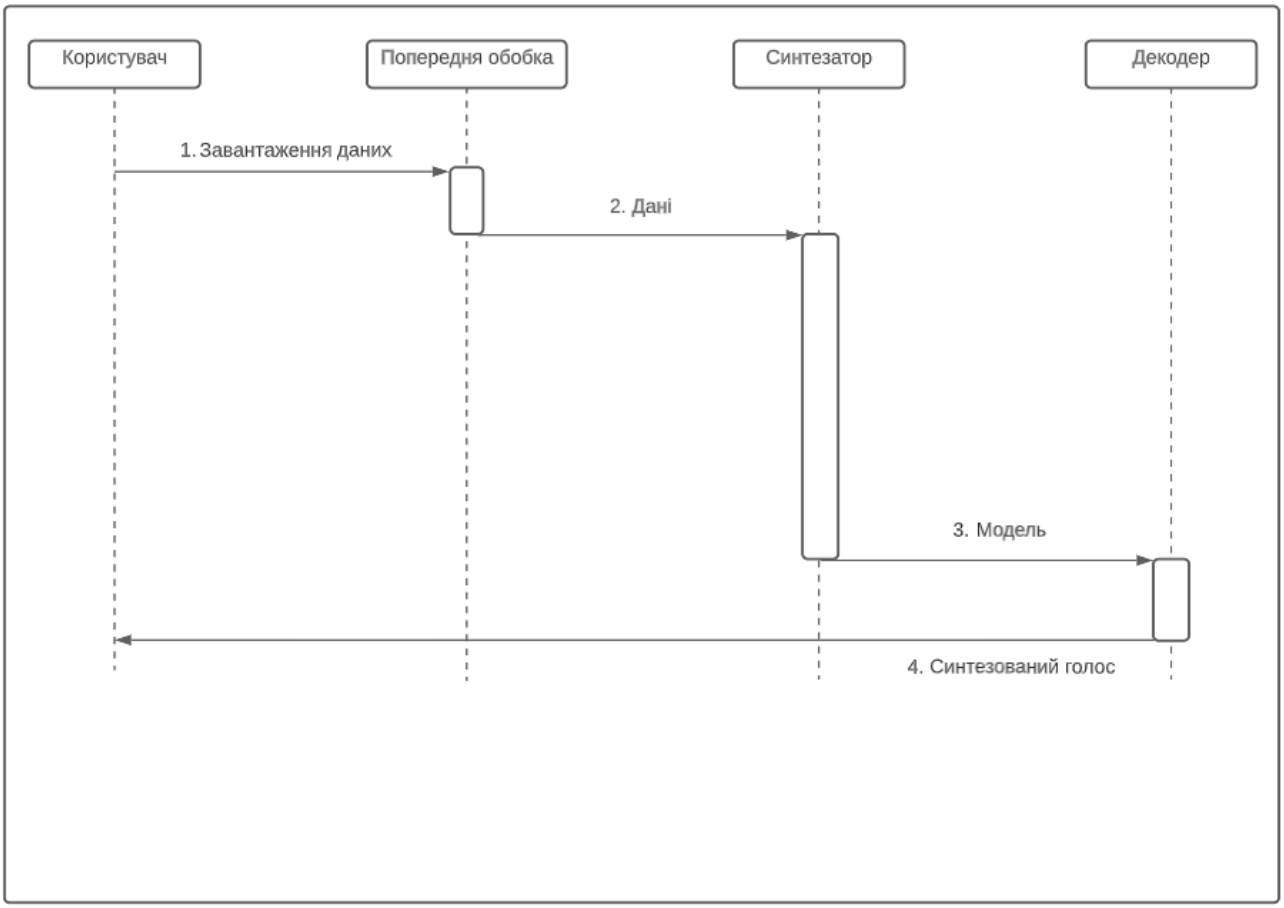


Рисунок 2.8 Діаграма послідовності

Висновок до другого розділу

У розділі 2 "Проектування нейромережевого застосунку синтезу голосу" було детально розглянуто та описано різні аспекти, необхідні для успішної реалізації засосунку синтезу голосу.

Було представлено детальний опис застосованих нейронних мереж, зокрема згорткових, розгортаючих та рекурентних мереж, які використовуються

для аналізу та синтезу голосових сигналів. Пояснено їх роль у процесі клонування голосу.

Окрему увагу було приділено розробці TTS (Text-to-Speech) алгоритму, який дозволяє перетворювати вхідний текст на відповідне мовлення.

Також було представлено Griffin-Lim алгоритм який застосовуватиметься як вокодер для генерації аудіо з синтезованого мовлення.

В рамках розділу було також представлено математичне представлення перетворення Фур'є та короткочасного перетворення Фур'є, що є важливими елементами для аналізу та обробки звукових сигналів.

Крім того, була розроблена архітектура нейромережевого застосунку синтезу голосу, яка була представлена у вигляді нотації IDEF0, дерева функцій системи, компонентної архітектури, діаграми прецедентів і послідовностей.

РОЗДІЛ 3. АЛГОРИТМІЧНИЙ АНАЛІЗ ПРОГРАМНОГО ЗАСТОСУНКУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Опис інструментів для реалізації програмної частини

TensorFlow

Бібліотека з відкритим кодом для створення моделей глибинного навчання (DeepLerning), машинного навчання (MachineLearning) та інших предиктивних аналітичних обчислень. Створена для оптимізації розробки, зменшення ресурсоємності та покращення якості аналітичних прогнозувань.

Numpy

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

Librosa

Python для аналізу музики та аудіо. Він забезпечує будівельні блоки, необхідні для створення систем пошуку музичної інформації. Ця бібліотека є назвичайно важливою в даному проекті, оскільки нам потрібно працювати в основному зі звуками

TQDM

З арабської прогрес, дана бібліотека потрібна для візуалізації прогресу обробки нашої моделі.

SciPy

Дана бібліотека містить модулі для оптимізації, інтегрування, спеціальних функцій, обробки сигналів, обробки зображень, генетичних алгоритмів,

розв'язування звичайних диференціальних рівнянь та інших задач, які розв'язуються в науці і при інженерній розробці.

(Вимоги щодо версій і необхідних встановлений бібліотек буде наведено в додатку А)

3.2 Опис даних

За основу вхідних даних для нашої моделі було взято записи Лінди Джонсон з LibriVox, іменовані як LJ Speech Dataset (таблиця 3.1). Цей набір даних знаходиться у вільному доступі і обмежень щодо його використання немає. Як зазначають автори ця робота є суспільним надбанням і можна використовувати без посилання на джерела. Ці записи було зроблені на основі 7 робіт. (додаток Б).

Таблиця 3.1. Статистика аудіозаписів бази даних

Всього кліпів	13 100
Всього слів	225 715
Загальна кількість символів	1 308 678
Загальна тривалість	23:55:17
Середня тривалість кліпу	6.57 сек
Мінімальна тривалість кліпу	1.11 сек
Максимальна тривалість кліпу	10.10 сек
Середня кількість слів у кліпі	17.23
Унікальні слова	13 821

Ці записи останні часом вважають еталонним набором даних у завдання text-to-speech. Цей набір даних має текстові записи мовця, що є невід'ємливою

складовою для нашої задачі. Ці записи були сегментовані на основі тиші в аудіозаписах. Завдяки великої кількості годин, наша модель краще навчається.

Як вхідні дані для утворення синтезованого аудіозапису було використано «Гарвардські вирази». Вони вважаються як «рекомендовані вирази щодо вимірювання якості англійського мовлення». (Додаток В). Для правильної роботи модулю я виділив наступні вимоги для запису виразів у текстовий файл:

- Перший рядок програма пропускає і починає з другого, його краще залишити пустим, або вказати якусь інформацію для себе.
- Алфавіт допустимих символів включає в себе латинський алфавіт і символи “ ‘ ” та “ . ” .
- Кожен файл це один рядок. Кількість вихідних файлів дорівнює кількість рядків не включаючи перший.
- Якщо ви бажаєте створити аудіозапис з декількох речень, розділяйте їх символом “ . ”. Тоді програма створить запис де буде розподіл між реченнями за інтонацією.
- Результуючий аудіозапис створюється до 12 секунд запису. Якщо кількість слів не вміститься, програма або включатиме в себе слова які вмістилися в цей час, або спотворить останні секунди файлу.
- Якщо залишити пустий рядок (не включаючи перший), програма створить файл без голосу, тому для уникнення таких ситуацій необхідно видалити всі пусті рядки, і перевірити чи після всіх виразів немає пустих рядків.

3.3 Підготовка даних для моделі

Одним із важливих етапів є підготовка даних до алгоритму. Всі дії ми зробимо програмно. Оскільки наша система оперує не самим сигналом а спектрами, ми повинні тут реалізувати перетворення в амплітудний і мел-спектр.

Етапи:

- Отримання сигналу фіксованої чатоти дискретизації.

- Обрізання тишини по краях.

- Фільтр попереднього виділення (діапазон вхідних частот, найбільш сприятливий до шуму, підвищується).

- Віконне перетворення Фур'є (застосовується для визначення синусоїдної частоти).

- Визначення амплітудного спектру.

- Визначення мел-спектру.

- Переведення цих даних в децибели.

- Їхня нормалізація.

- Транспонування і приведення до потрібних типів.

- Добивання нулями до правильних розмірностей.

- Пониження частоти дискретизації для мел-спектру.

Тепер наші дані готові для входу в кодер.

3.4 Опис алгоритму

3.4.1 Кодер

Модель являє собою 3-шарову LSTM із 768 прихованими вузлами, за якими слідує проекційний шар із 256 одиниць. Проекційний рівень - це повністю зв’язаний шар із 256 виходами на LSTM, який неодноразово застосовується до кожного виходу LSTM.

Вхідними даними для моделі є 40-канальні спектрограми log-mel із шириною вікна 25 мс і кроком 10 мс. Результатом є L2-нормалізований прихований стан останнього шару, який є вектором із 256 елементів.

Кодер динаміка зосереджений не на словах яких говорить мовець, а все що його цікавить, це голос оратора, наприклад високий чи низький голос, акцент, тон, тощо.

Усі ці характеристики об’єднуються в низьковимірний вектор, або як його називають d-вектор.

Фіксована тривалість висловловлювань у навчальній партії становить 1,6 секунд. Це часткові вислослювання відібрані з довших повних висловлювань у наборі даних. Висловлювання сегментують за довжиною по 1,6 секунд, які перекриваються на 50% наступним сегментом. Отримані результати усереднюються і потім нормалізуються для створення висловлювання.(рис 3.2)

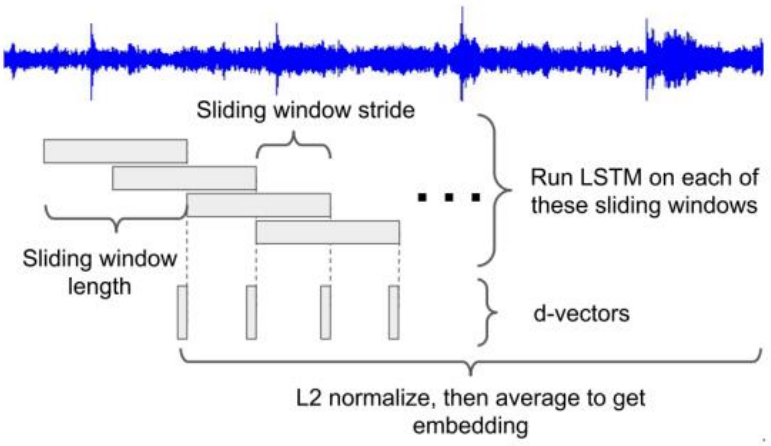


Рисунок 3.2 Схема навчання кодера

3.4.2 Синтезатор

Першою частиною синтезатора є Text2mel (рис 3.3). Це рекуретна модель послідовності до послідовності, яка прогнозує мел-спектрограму з тексту.

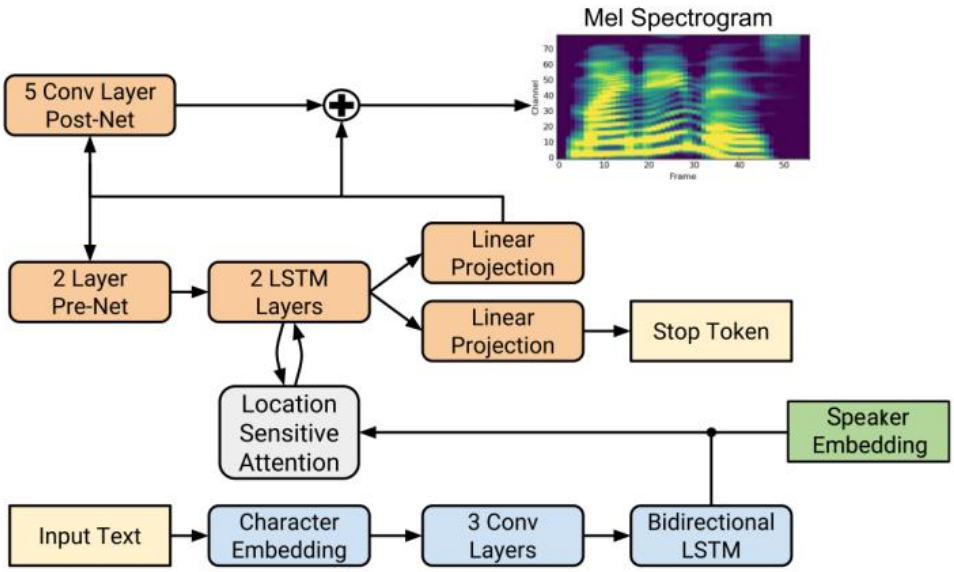


Рисунок 3.3. Архітектура Tacotron синтезатора

Він має структуру кодера-декодера. Окремі символи з текстової послідовності спочатку вбудовуються як вектори. Далі слідують згорткові шари, щоб збільшити діапазон одного кадру кодера. Ці кадри пропускаються через двонаправлений LSTM для створення вихідних кадрів кодера. Вбудований динамік об’єднується з кожним кадром, який виводить кодер Tacotron. Механізм уваги звертає увагу на вихідні кадри кодера для генерації вхідних кадрів декодера. Кожен вхідний кадр декодера об’єднується з вихідним кадром попереднього декодера, що проходить через попередню мережу, що робить модель авторегресійною. Цей конкатенований вектор проходить через два односпрямовані шари LSTM перед проектуванням на один кадр спектрограми Mel. Інша проекція того самого вектора на скаляр дозволяє мережі самостійно передбачити, що вона повинна припинити генерувати кадри, випромінюючи значення вище певного порогу. Уся послідовність кадрів пропускається через залишкову пост-мережу, перш ніж вона стане спектрограмою Mel. Інша проекція того самого вектора на скаляр дозволяє мережі самостійно передбачити, що вона повинна припинити генерувати кадри, випромінюючи значення вище певного порогу. Уся послідовність кадрів пропускається через залишкову пост-мережу, перш ніж вона стане спектрограмою Mel.

Цільові мел-спектрограми для синтезатора мають більше можливостей, ніж ті, які використовуються для кодера динаміка. Вони обчислюються з вікна 50 мс з кроком 12,5 мс і мають 80 каналів. У моїй реалізації вхідні тексти не обробляються для вимови, і символи подаються як вони є. Існує кілька процедур очищення: заміна абревіатур і цифр їхньою повною текстовою формою, примусове переведення всіх символів у ASCII, нормалізація пробілів і перетворення всіх символів у нижній регістр. Можна використовувати і пунктуацію, але її немає в моєму набору даних.

Тренування синетзатора проходить на 200 тисяч кроків.

3.4.3 Декодер

Декодер програми складається без штучного інтелекту WaveNet. В результаті синтезу ми отримуємо модель яка може генерувати звук, але ми його прослухати не можемо. Нам потрібно відновити спектрограму. Під цю задачу чудово підходить алгоритм Гріфіна-Ліма.

У даному алгоритмі можна виділити такі етапи:

- Створення складної матриці, вставляння спектограми як реальний шар і створення уявного шару рівномірного шуму.
- Виконання на шарі зворотного короткочасного перетворення Фур’є.
- Обчислення короткочасного перетворення Фур’є отриманого часового ряду.
- У цьому перетворенні, якою є складна матриця, замінюється реальний шар на оригінальну спектрограму (в результаті ми отримуємо комплексну спектрограму з незмінною інформацією про амплітуду).
- Повертаємося до виконання на шарі зворотного короткочасного перетворення Фур’є.
- Повторюємо доки не отримаємо бажаний результат.

Метод є ітеративним, тому ми на практиці перевіряємо результат.

На мою думку 50-100 ітерацій достатньо, особисто я використав 100, результат між 50 і 100 ітерацій майже не помітний.

В результаті ми отримуємо сигнал, який ми потім перетворюємо в wav сигнал.

Загальний алгоритм вокодера має вигляд:

- Транспонування спектрограми
- Денормалізація
- Повернення від децибел до амплітуд
- Запуск алгоритму Гріфіна-Ліма.
- Перетворення в wav запис.
- Обрізання.

Найкращим способом для вокодера все одно залишається WaveNet. Оскільки він використовується штучний інтелект і якість набагато краща. Можна в цьому переконатися взявши будь-який вхідний запис, перетворити його на спектрограму, і потім прогнавши його через вокодер.

3.5 Аналіз результатів

Для тексту який ми озвучуємо, я взяв гарвардські вирази. Вони знаходяться у блоках по 10 виразів. Для переконливості я замість останнього виразу вставив свій.

- «Hello world my name’s mr. Ruslan»

Я проаналізував результат. На мою думку, досить замітно що це штучний голос. Також є великі проблеми з шумами в доріжці. В гарвардських реченнях проблем з вимовою і наголосами немає. Але в моєму речені досить замітна

проблема з вимовою імені. Мережа досить незрозуміло поставила наголос в звучання. Наголос ніби рівномірний, але трохи більше на перший склад, хоча має бути на другий. Також є проблема і з звучанням, друга буква вимовляється ніби це було слово, а не ім’я.

Кількість ітерацій синтезатора 200 тисяч. На мою думку для покращення результату ми можемо зробити такі речі:

- Зробити кодер динаміка з використання WaveNet.
- Збільшити кількість ітерацій синтезатора.
- Зробити вокодер з застосуванням WaveNet.
- Для кращої вимови і наголосу можна також використовувати словники.

Також одною з великих проблем є знаки пунктуації. Вирази розділенні за знаками пунктуації. Тому постається питання, як нам сказати вираз «Казнити неможна помилувати»?

Варто зазначити що програма може згенерувати речення якщо ми подали деякі скорочення на текст. В прикладі це «`s» та «mr.». (додаток Г)

Я вирішив проаналізувати час роботи програми, і визначити чи збільшується він кількості вхідних даних (таблиця 3.2).

Таблиця 3.2. Результат роботи програми в залежності від кількості виразів

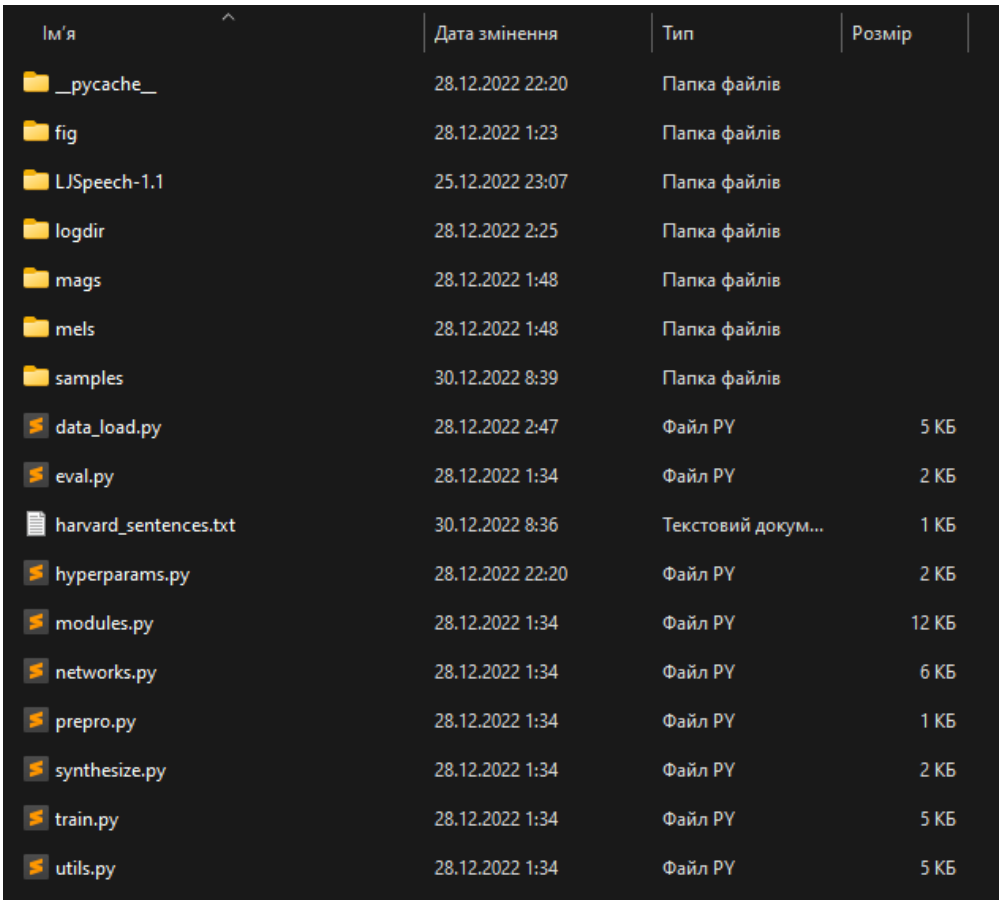
Кількість виразів	Час всіх записів	Загальний розмір	Час роботи програми	Залежність від часу	Залежність від розміру
1	2.08 с	0.242 мб	31 с	14.9 с/с	128.09 с/мб
10	25.3 с	2.09 мб	116 с	4.58 с/с	55.50 с/мб
20	49.0 с	4.06 мб	225 с	4.59 с/с	55.41 с/мб
50	119.8 с	9.91 мб	570 с	4.78 с/с	57.52 с/мб

Можна замітити, що залежність від розміру і від часу результуючих записів, час витрачений на роботу програми за 1 одиницю майже не відрізняється. Але коли ми лише запускаємо один вираз, час роботи сильно відрізняється, це можна пояснити тим, що в порівнянні з витраченим часом на синтез з багатьма виразами, час запуску компонентів і функцій на один вираз є відносно великим великим.

Якщо ж брати багато виразів, час витрачений на роботу програми на одну секунду запису дорівнюватиме ~4.64с, а на 1 мб ~56.11.

3.6 Інструктивний матеріал користувача для роботи з нейромережевим застосунком синтезу голосу

1. Для роботи з нейромережевим застосунком необхідно установити python відповідної версії, а також всі необхідні біблотеки.
2. Наступним кроком необхідно розархівувати папку з проектом (рис. 3.4).



Ім'я	Дата змінення	Тип	Розмір
__pycache__	28.12.2022 22:20	Папка файлів	
fig	28.12.2022 1:23	Папка файлів	
LJSpeech-1.1	25.12.2022 23:07	Папка файлів	
logdir	28.12.2022 2:25	Папка файлів	
mags	28.12.2022 1:48	Папка файлів	
mels	28.12.2022 1:48	Папка файлів	
samples	30.12.2022 8:39	Папка файлів	
data_load.py	28.12.2022 2:47	Файл PY	5 КБ
eval.py	28.12.2022 1:34	Файл PY	2 КБ
harvard_sentences.txt	30.12.2022 8:36	Текстовий докум...	1 КБ
hyperparams.py	28.12.2022 22:20	Файл PY	2 КБ
modules.py	28.12.2022 1:34	Файл PY	12 КБ
networks.py	28.12.2022 1:34	Файл PY	6 КБ
prepro.py	28.12.2022 1:34	Файл PY	1 КБ
synthesize.py	28.12.2022 1:34	Файл PY	2 КБ
train.py	28.12.2022 1:34	Файл PY	5 КБ
utils.py	28.12.2022 1:34	Файл PY	5 КБ

Рисунок 3.4 Архів проекту

3. Тепер необхідно налаштувати параметри.

3.1 У файлі hyperparams.py знайдіть змінні data(відповідає за навчальні дані, на яких вчиться модель) і test_data(відповідає за вирази які буде генерувати модель у вигляді аудіозапису). Змінна data повинна подаватися у вигляді папки, яка містить папку wavs де збережені файли з записом голосу у форматі .wav, і файл з текстом який є у записах (транскрипції) який повинен бути у форматі .csv. Змінна test_data повинна бути у форматі .txt. Змініть параметри змінних відповідно до своїх даних (рис. 3.5).

```
data = "LJSpeech-1.1"
test_data = 'harvard_sentences.txt'
```

Рисунок 3.5 Змінні відповідаючі за навчальні і тестові дані.

3.2 У файлі hyperparams.py знайдіть змінні logdir(шлях де знаходитиметься наша навчальна модель), sampledir (шлях де знаходитиметься

результат синтезування), та num_iterations(кількість ітерацій на яких навчатиметься модель). Перші дві змінні це папки, а третя це цілочислене число(чим більше число, тим довше навчатиметься наша модель, і тим кращі будуть наші результати, рекомендується від 50 000). Змініть параметри змінних відповідно до своїх побажань (рис. 3.6).

```
logdir = "logdir/01"
sampledir = 'samples'
num_iterations = 200000
```

Рисунок 3.6 Змінні відповідаючі за навчальні і тестові дані.

3.3 У файлі hyperparams.py знайдіть змінну prepro. Вона відповідає за попередню підготовку даних. Якщо ви бажаєте виконати підготовку встановіть значення True (рис. 3.7), інакше False.

```
prepro = True
```

Рисунок 3.7 Змінна відповідаюча за попередню підготовку даних.

3.4 У файлі data_load.py знайдіть змінну transcript. В цю змінну подаються наші транскрипції у форматі .csv. Файл повинен зберігатися у папці відповідно змінній data(пункт 3.1). Відповідно до назви свого файлу замініть значення на необхідне (рис. 3.8).

```
transcript = os.path.join(hp.data, 'metadata.csv')
```

Рисунок 3.8 Змінна відповідаюча за дані транскрипцій.

4. Запустіть командний рядок (Terminal або Windows PowerShell) з папки (рис. 3.9 та 3.10).

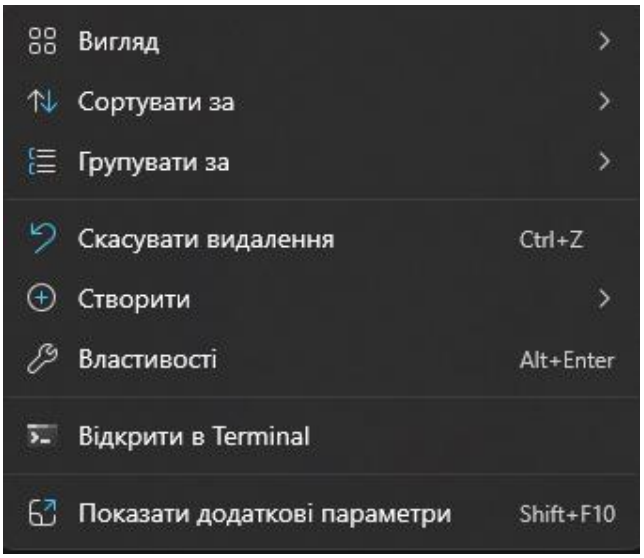


Рисунок 3.9 Вікно параметрів у файловому провіднику

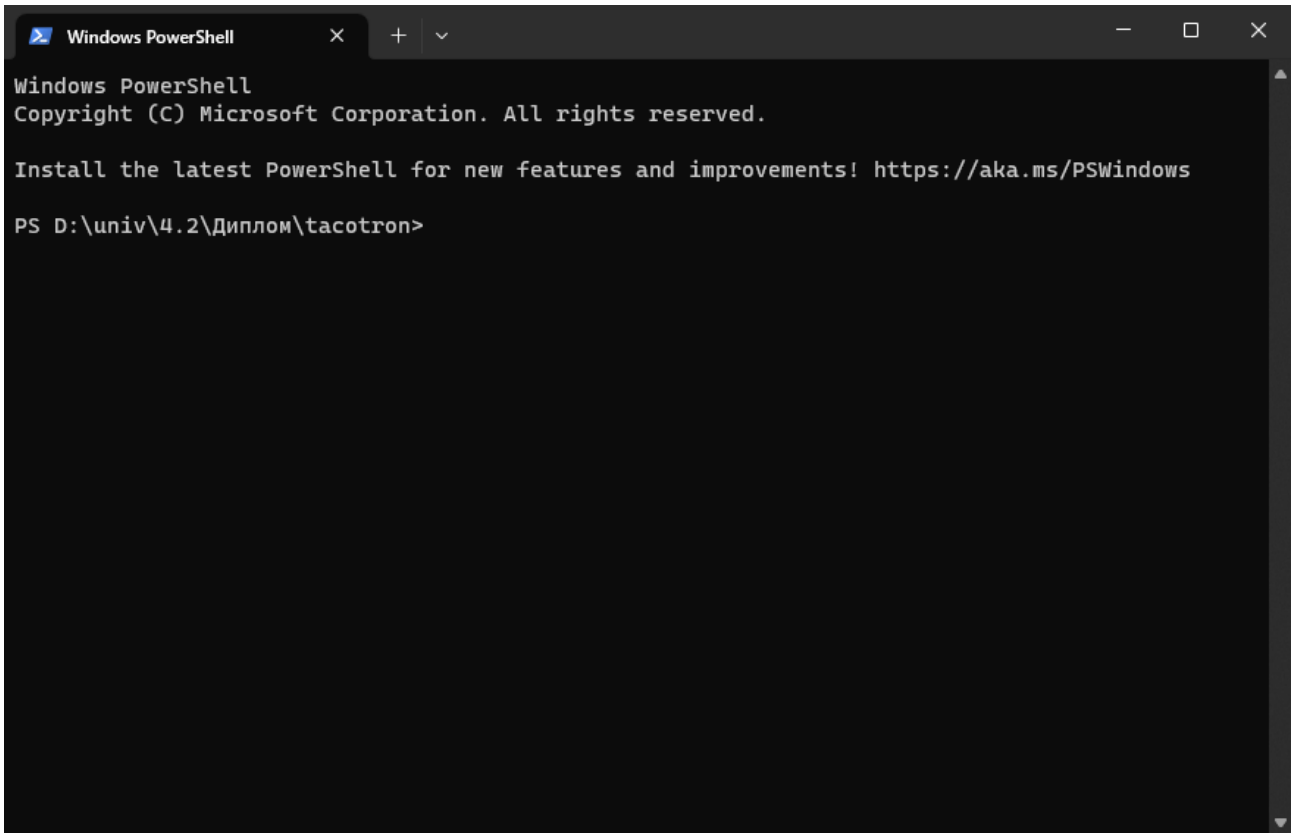


Рисунок 3.10 Вікно командного рядка.

5. Якщо ви бажаєте створити власну модель виконайте наступні етапи навчання мережі, інакше перейдіть до пункту 6.

5.1 Якщо ви установили значення prepro – True, тоді запустіть файл prepro.py (рис. 3.11). Інакше пропустіть даний крок.

```
PS D:\univ\4.2\Диплом\tacotron> py -3.7 prepro.py
```

Рисунок 3.11 Запуск файлу попередньої підготовки даних.

5.2 Запустіть файл train.py з параметром 1 (рис. 3.12), це запустить навчання мережі Text2mel.

```
PS D:\univ\4.2\Диплом\tacotron> py -3.7 train.py 1
```

Рисунок 3.12 Запуск мережі Text2mel.

5.3 Запустіть файл train.py з параметром 2 (рис. 3.13), це запустить навчання мережі SSRN.

```
PS D:\univ\4.2\Диплом\tacotron> py -3.7 train.py 2
```

Рисунок 3.13 Запуск мережі Text2mel.

6. Запустіть файл synthesize.py (рис. 3.14), це запустить відновлення аудіозапису з тексту.

```
PS D:\univ\4.2\Диплом\tacotron> py -3.7 synthesize.py
```

Рисунок 3.14 Запуск синтезатора.

Наша модель буде зберігатися у папці яка вказана у змінній logdir(пункт 3.2), а відновлені аудіозаписи у папці яка вказана у змінній sampledir(пункт 3.2).

У системах може бути декілька версій python, але оскільки у нас є обмеження, то ми повинні вказувати при запуску яку версію ми хочемо запустити.

Висновок до третього розділу

У розділі 3 "Алгоритмічний аналіз нейромережевого застосунку та аналіз результатів" проведено детальний опис даних, їх підготовку для використання в моделі клонування голосу, а також описано алгоритм роботи модуля, який включає кодер, синтезатор та декодер.

Було проведено огляд інструментів, що використовуються для реалізації програмної частини модулю, зокрема, було розглянуто інструменти для роботи з нейронними мережами та обробки звукових сигналів.

У підрозділі 3.4 "Аналіз результатів" було проведено оцінку результатів роботи нейромережевого застосунку. Шляхом порівняння синтезованого голосу з оригінальним.

У розділі 3.5 "Інструктивний матеріал користувача для роботи з нейромежевим застосунком клонування голосу" було надано необхідну інформацію та пояснення користувачам щодо використання застосунку. Цей матеріал допомагає користувачам зрозуміти, як правильно використовувати застосунок, його можливості та обмеження.

Отже, розроблений нейромережевий застосунок синтезу голосу показав задовільні результати у відтворенні особливостей голосу говорячої особи. Результати аналізу підтверджують, що модуль може бути успішно використаний для створення синтезованої речі. Інструктивний матеріал користувача надає необхідну інформацію для ефективного використання нейромережевого застосунку синтезу голосу.

ВИСНОВОК

У даному дослідженні був розглянутий нейромережевий застосунок синтезу голосу, що базується на використанні нейронних мереж та алгоритмів обробки звукових сигналів.

Аналітичний огляд показав, що синтез мовлення є актуальною проблемою, яка має широкі можливості застосування в різних областях, таких як комунікація, розваги, навчання та інші. З'ясовано основні проблеми, пов'язані з якістю та натуральністю синтезованого голосу.

Архітектура програмного модулю була розроблена, включаючи компоненти кодера, синтезатора та декодера, які співпрацюють для досягнення бажаного результату. Застосовані нейронні мережі, зокрема згорткові, розгортаючі та рекурентні мережі, виявилися ефективними для аналізу та синтезу голосових сигналів.

Аналіз результатів показав, що програмний модуль здатний клонувати голос, але все життєво важливо провести подальші дослідження та вдосконалення для поліпшення якості і реалістичності синтезованого голосу.

Результатом роботи є нейромережевий застосунок синтезу голосу, який на основі вхідних даних за допомогою мереж будує навчальну модель, і на основі моделі здатний синтезувати аудіофайли.

Отже, розроблений програмний модуль клонування голосу має потенціал для використання в різних областях і може стати цінним інструментом для створення якісного синтезованого мовлення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Manfred R. Schroeder. A brief history of synthetic speech [Електронний ресурс] / Manfred R. Schroeder – Режим доступу до ресурсу: https://doi.org/10.1016/0167-6393(93)90074-U.

2. How Speech Technologies Can Help People with Disabilities [Електронний ресурс] / Vlado Delić, Milan Sečujski, Nataša Vujnović Sedlar та ін.] – Режим доступу до ресурсу: https://doi.org/10.1007/978-3-319-11581-8_30.

3. Applications of Synthetic Speech [Електронний ресурс] – Режим доступу до ресурсу: http://research.spa.aalto.fi/publications/theses/lemmetty_mst/chap6.html.

4. Bernd J. Kröger. Articulatory Synthesis of Speech and Singing: State of the Art and Suggestions for Future Research [Електронний ресурс] / Bernd J. Kröger, Peter Birkholz – Режим доступу до ресурсу: https://doi.org/10.1007/978-3-642-00525-1_31.

5. Text-to-Speech Engines Text Normalization [Електронний ресурс] Режим доступу до ресурсу: https://learn.microsoft.com/enus/windows/win32/lwef/text-to-speech-engines-textnormalization?redirectedfrom=MSDN.

6. Jonathan Harrington. Digital Formant Synthesis [Електронний ресурс] / Jonathan Harrington, Steve Cassidy – Режим доступу до ресурсу: https://doi.org/10.1007/978-94-011-4657-9_7.

7. A Review of Deep Learning Based Speech Synthesis [Електронний ресурс] / Yishuang Ning, Sheng He, Zhiyong Wu та ін.] – Режим доступу до ресурсу: https://doi.org/10.3390/app9194050.

8. Alan W Taylor. Automatically clustering similar units for unit selection in speech synthesis [Електронний ресурс] / Alan W Taylor, Paul A – Режим доступу до ресурсу: https://era.ed.ac.uk/handle/1842/1236.

9. A.J. Hunt. Unit selection in a concatenative speech synthesis system using a large speech database [Електронний ресурс] / A.J. Hunt, A.W. Black – Режим доступу до ресурсу: https://ieeexplore.ieee.org/abstract/document/541110.

10. Rolf Carlson Prof. Rule-Based Speech Synthesis [Електронний ресурс] / Rolf Carlson Prof., Björn Granström Prof. – Режим доступу до ресурсу: https://www.doi.org/10.1007/978-3-540-49127-9_20.

11. Yann LeCun. Deep learning [Електронний ресурс] / Yann LeCun, Yoshua Bengio, Geoffrey Hinton – Режим доступу до ресурсу: https://www.nature.com/articles/nature14539.

12. Matt Spencer. A Deep Learning Network Approach to ab initio Protein Secondary Structure Prediction [Електронний ресурс] / Matt Spencer, Jesse Eickholt, Jianlin Cheng – Режим доступу до ресурсу: https://ieeexplore.ieee.org/abstract/document/6872810.

13. Kyubyong Park. CSS10: A Collection of Single Speaker Speech Datasets for 10 Languages [Електронний ресурс] / Kyubyong Park, Thomas Mulc – Режим доступу до ресурсу: https://arxiv.org/abs/1903.11269.

14. Naihan Li. Neural Speech Synthesis with Transformer Network [Електронний ресурс] / Naihan Li, Shujie Liu, Yanqing Liu. – 2018. – Режим доступу до ресурсу: https://arxiv.org/abs/1809.08895.

15. M. Schuster. Bidirectional recurrent neural networks [Електронний ресурс] / M. Schuster, K.K. Paliwal – Режим доступу до ресурсу: https://ieeexplore.ieee.org/abstract/document/650093.

16. Fast Fourier Transformation FFT - Basics [Електронний ресурс] – Режим доступу до ресурсу: https://www.nti-audio.com/en/support/know-how/fastfourier-transform-fft.

17. Henri J. Nussbaumer . The Fast Fourier Transform [Електронний ресурс] / Henri J. Nussbaumer – Режим доступу до ресурсу: https://doi.org/10.1007/978-3-662-00551-4_4.

18. W.T. Cochran. What Is the Fast Fourier Transform [Електронний ресурс] / W.T. Cochran, J.W. Cooley, D.L. Favin – Режим доступу до ресурсу: https://doi.org/10.1109/PROC.1967.5957.

19. P.M. Bentley. Wavelet transforms: an introduction. Electronics & Communication Engineering Journal [Електронний ресурс] / P.M. Bentley, J.T.E. McDonnell – Режим доступу до ресурсу: https://doi.org/10.1049/ecej:19940401.

20. A. Arneodo. Wavelet Transform of Multifractals [Електронний ресурс] / A. Arneodo, G. Grasseau, M. Holschneider – Режим доступу до ресурсу: https://doi.org/10.1103/PhysRevLett.61.2281.

21. Dengsheng Zhang. Wavelet Transform [Електронний ресурс] / Dengsheng Zhang – Режим доступу до ресурсу: https://doi.org/10.1007/978-3-030-17989-2_3.

22. Nathanaël Perraudin. Stationary Signal Processing on Graphs [Електронний ресурс] / Nathanaël Perraudin, Pierre Vandergheynst – Режим доступу до ресурсу: https://ieeexplore.ieee.org/abstract/document/7891646.

23. Yuxuan Wang. Tacotron: Towards End-to-End Speech Synthesis [Електронний ресурс] / Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton – Режим доступу до ресурсу: https://arxiv.org/abs/1703.10135.

24. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation [Електронний ресурс] / Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre та ін.] – Режим доступу до ресурсу: https://arxiv.org/abs/1406.1078.

25. Yi Ren. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech [Електронний ресурс] / Yi Ren, Chenxu Hu, Xu Tan – Режим доступу до ресурсу: https://arxiv.org/abs/2006.04558.

26. Nathanaël Perraudin. A fast Griffin-Lim algorithm [Електронний ресурс] / Nathanaël Perraudin, Peter Balazs, Peter L. Søndergaard – Режим доступу до ресурсу: https://ieeexplore.ieee.org/abstract/document/6701851.

27. Yoshiki Masuyama. Deep Griffin–Lim Iteration [Електронний ресурс] / Yoshiki Masuyama, Kohei Yatabe, Yuma Koizumi – Режим доступу до ресурсу: https://ieeexplore.ieee.org/abstract/document/8682744.

28. Ilya Sutskever. Sequence to Sequence Learning with Neural Networks [Електронний ресурс] / Ilya Sutskever, Oriol Vinyals, Quoc V. Le – Режим доступу до ресурсу: https://arxiv.org/abs/1409.3215.

29. Martín Abadi. TensorFlow: Learning Functions at Scale [Електронний ресурс] / Martín Abadi – Режим доступу до ресурсу: https://doi.org/10.1145/2951913.2976746.

ДОДАТКИ

Додаток А.

Таблиця необхідних бібліотек і їх версій

Бібліотека	Вимоги
Python(середовище)	3<x<3.8
NumPy	x>=1.11.1
Tensorflow	1.15<=X<2
Librosa	Немає
TQDM	Немає
Matplolib	Немає
scipy	Немає

Додаток Б.

Уривки робіт які містяться в наборі даних

Автор	Робота
Morris, William, et al.	*Arts and Crafts Essays*. 1893.
Griffiths, Arthur.	*The Chronicles of Newgate, Vol. 2*. 1884.
Roosevelt, Franklin D.	*The Fireside Chats of Franklin Delano Roosevelt*. 1933-42.
Harland, Marion.	*Marion Harland's Cookery for Beginners*. 1893.
Rolt-Wheeler, Francis.	*The Science - History of the Universe, Vol. 5: Biology*. 1910.
Banks, Edgar J.	*The Seven Wonders of the Ancient World*. 1916.
President's Commission on the Assassination of President Kennedy.	*Report of the President's Commission on the Assassination of President Kennedy*. 1964.

Додаток В.

Таблиця вхідних даних для синтезування

Номер	Текст
1	The birch canoe slid on the smooth planks.
2	Glue the sheet to the dark blue background.
3	It's easy to tell the depth of a well.
4	These days a chicken leg is a rare dish.
5	Rice is often served in round bowls.
6	The juice of lemons makes fine punch.
7	The box was thrown beside the parked truck.
8	The hogs were fed chopped corn and garbage.
9	Four hours of steady work faced us.
10	Large size in stockings is hard to sell.
11	The boy was there when the sun rose.
12	A rod is used to catch pink salmon.
13	The source of the huge river is the clear spring.
14	Kick the ball straight and follow through.
15	Help the woman get back to her feet.
16	A pot of tea helps to pass the evening.
17	Smoky fires lack flame and heat.
18	The soft cushion broke the man's fall.
19	The salt breeze came across from the sea.
20	Hello world my name's mr. Ruslan.

Додаток Г.

Таблиця скорочень у тексті

Абревіатура	Розширення
Mr.	Mister
Mrs.	Misess
Dr.	Doctor
No.	Number
St.	Saint
Co.	Company
Jr.	Junior
Maj.	Major
Gen.	General
Drs.	Doctors
Rev.	Reverend
Lt.	Lieutenant
Hon.	Honorable
Stg.	Sergeant
Capt.	Captain
Esq.	Esquire
Ltd.	Limited
Col.	Colonel
Ft.	Fort