Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Інформаційна технологія інтелектуального синтезу мовлення на основі текстових даних
(тема)

Виконав:
здобувач четвертого року навчання,
групи ІТШІ-22-2
Володимир АГАФОНОВ
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Освітня програма Штучний інтелект
(повна назва спеціалізації)

Керівник ас. Микола ЧЕРНЕНКО
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри _____ Лариса ЧАЛА
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Освітня програма Штучний інтелект
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Агафонову Володимиру Дмитровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна технологія інтелектуального синтезу мовлення на основі текстових даних

затверджена наказом університету від 5 червня 20 26 р. № 488Ст
2. Термін подання студентом роботи до екзаменаційної комісії 14 липня 20 26 р.
3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел, відомі наукові-дослідження та проекти, документація мови програмування Python, документація фреймворку PyTorch, датасети для тренування та тестування нейронних мереж

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз існуючих рішень синтезу мовлення
2) Використання дифузії через Flow Matching для синтезу мовлення
3) Розробка програмної реалізації моделі з використанням Flow Matching
4) Проведення експериментального дослідження та порівняльного аналізу з аналогами

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.06.2026	виконано
2	Аналіз предметної галузі	10.06.2026	виконано
3	Огляд існуючих метрик оцінювання мовлення	12.06.2026	виконано
4	Аналіз підходів до синтезу мовлення	14.06.2026	виконано
5	Експериментальне моделювання та навчання	20.06.2026	виконано
	моделі		
6	Написання пояснювальної записки	01.07.2026	виконано
7	Перевірка на академічний плагіат	02.07.2026	виконано
8	Нормоконтроль	04.07.2026	виконано
9	Підготовка презентації та доповіді	08.07.2026	виконано
10	Попередній захист	09.07.2026	виконано
11	Рецензування	10.07.2026	виконано
12	Захист перед ЕК	14.07.2026	

Дата видачі завдання 8 червня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ ас. Микола ЧЕРНЕНКО
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 71 с., 26 рис., 1 дод., 21 джерело.

АКУСТИЧНА МОДЕЛЬ, ВОКОДЕР, ГЕНЕРАЦІЙНА МОДЕЛЬ, МЕЛ-СПЕКТОГРАМИ, ПРОСОДІЯ, СИНТЕЗ МОВЛЕННЯ, ФОНЕМИ.

Об’єкт дослідження – синтез мовлення з текстових даних.

Предмет дослідження – використання Flow Matching для швидкого та якісного синтезу мовлення.

Мета роботи – розробка моделі для синтезу мовлення, шляхом створення архітектури моделі, її програмної реалізації та навчання моделі і подальше використання її для синтезу мовлення.

Методи дослідження – теоретичний збір необхідної інформації, а саме збір даних про функціонування мовлення, фонетики, аналіз існуючих рішень та їхніх переваг і недоліків. Експериментальне дослідження, шляхом розробки архітектури та її програмної реалізації використовуючи мову програмування Python на базі фреймворку PyTorch та супутніх бібліотек оброки даних.

У результаті роботи було проведено теоретичний аналіз існуючих архітектур, з врахуванням їхніх недоліків було створено власну архітектуру з використанням Flow Matching. Модель було навчено на датасеті LJSpeech та паралельно навчено ще одну копію моделі на українському датасеті filatov-24000. Отримані навчені моделі змогли показати кращий результат ніж проаналізовані минулі архітектури, які були навчені на аналогічних наборах даних.

ABSTRACT

Bachelor’s thesis contains: 71 pp., 26 fig., 1 ann., 21 references.

ACOUSTIC MODEL, GENERATIONAL MODEL, MEL-SPECTOGRAMS, PHONEMES, PROSODY, SPEECH SYNTHESIS, VOCODER.

The object of research is speech synthesis from text data.

The subject of the research is the use of Flow Matching for fast and high-quality speech synthesis.

The purpose of the work is to develop a model for speech synthesis by creating a model architecture, its software implementation, and training the model and its further use for speech synthesis.

Research methods – theoretical collection of necessary information, namely, collection of data on the functioning of speech, phonetics, analysis of existing solutions and their advantages and disadvantages. Experimental research, by developing an architecture and its software implementation using the Python programming language based on the PyTorch framework and related data mining libraries.

As a result of the work, a theoretical analysis of existing architectures was conducted, taking into account their shortcomings, an own architecture was created using Flow Matching. The model was trained on the LJSpeech dataset and in parallel another copy of the model was trained on the dataset with ukrainian language filatov-24000. The obtained trained models were able to show a better result than the analyzed past architectures, which were trained on similar datasets.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 8
Вступ 9
1 Теоретичні та методологічні основи теми 10
1.1 Основи формування мовлення завдяки мовному апарату 10
1.2 Структура та частини людського мовлення 13
1.3 Етичні та правові виклики синтезу мовлення 17
1.4 Методи збереження та обробки аудіо-даних 18
2 Аналіз існуючих рішень синтезу мовлення 27
2.1 Синтез мовлення методом формантів 27
2.2 Альтернативні традиційні методи синтезу мовлення 29
2.3 Використання вокодерів на базі нейромережей 32
2.3.1 WaveNet 32
2.3.2 HiFi-GAN 35
2.4 Моделі синтезу мовлення на основі нейромережевих технологій 37
2.4.1 Моделі Tacotron 38
2.4.2 Моделі FastSpeech 41
3 Використання дифузії для синтезу мовлення 45
3.1 Механізм генерації завдяки дифузії 45
3.2 Conditional flow matching для швидкої генерації даних 47
3.3 Flow Matching для синтезу мовлення 48
3.4 Генерування усередненої мел-спектрограми 49
3.5 Вокодер iSTFTNet 51
3.6 Запропонована архітектура синтезу мовлення 52
4 Програмна реалізація 54
4.1 Використанні модулі та бібліотеки 54
4.2 Реалізація взаємодії з моделлю 55
4.3 Обробка тексту та перетворення у фонеми 55
4.4 Реалізація запропонованої архітектури 56

4.4.1 Реалізація енкодера та ембедінга з текстом 57
4.4.2 Реалізація декодера з Flow Matching 60
4.4.3 Реалізація вокодера 62
4.5 Результати навчання моделі та синтезу 63
4.6 Недоліки та можливі доповнення 65
Висновки 67
Перелік джерел посилання 68
Додаток А Відомість кваліфікаційної роботи 71

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Фонема – найменша одиниця мовлення у фонетиці;

ШІ – штучний інтелект;

API – Application programming interface – прикладний програмний інтерфейс;

CLI – Command line interface – інтерфейс командного рядка;

FFN – Feed Forward Net – Мережа прямого поширення;

MAS – Monotonic Alignment Search – пошук монотонного вирівнювання;

RoPE – Rotary Positional Embedding – ротаційне позиційне кодування;

TTS – Text to speech – текст у мовлення.

ВСТУП

Задача синтезу мовлення є актуальною проблемою, оскільки мовлення є одним із основних способів комунікації та передачі інформації між людьми. Синтез мовлення може допомагати людям з обмеженими можливостям, як і для людей з слабким зором, у якості озвучення текстової інформації, так і для людей у яких відсутня можливість самим вербально розмовляти.

Перші рішення у сфері синтезу мовлення мали доволі ефективні результати на свій час, що давало можливість вирішити проблему комунікації для німих людей, завдяки пристроям, які могли синтезувати мовлення. Але класичні рішення синтезу мовлення, мали низьку якість синтезованого мовлення і не могли якісно симулювати звучання голосу з усіма особливостями просодії мовлення, що заважало сприймати інформацію з синтезованого мовлення так як і інформацію з натурального мовлення.

З 2010-х років сталася революція у сфері синтезу мовлення, завдяки використанню методів глибокого навчання, через нейронні мережі. З того часу почався бурхливий розвиток різних моделей та архітектур, які намагалися вирішити дві головні проблеми, які все ще має синтез мовлення. Це підвищення якості синтезованого мовлення та зменшення часу синтезу та навчання для моделі.

Ціль цієї роботи також у вирішенні проблеми якості синтезу та швидкості синтезу мовлення. Рішення цих проблем буде досягнуто завдяки використанню більш ефективних методів дифузії, що дають можливість швидко та якісно генерувати дані.

1 ТЕОРЕТИЧНІ ТА МЕТОДОЛОГІЧНІ ОСНОВИ ТЕМИ

1.1 Основи формування мовлення завдяки мовному апарату

Перед початком розробки інтелектуальної моделі синтезу мовлення необхідно розібрати, як саме людський організм формує мовлення та з чого складається людське мовлення. Це допоможе визначити за якої архітектурою необхідно розробити модель та які проблеми можуть бути у задачі синтезу мови.

Формування мовлення у людини це багатофакторний процес, який складається з багатьох етапів, кожен з яких формує нюанси людського мовлення для того щоб у кінці отримати індивідуальний звук мовлення для кожної людини. Сформуємо загальну картину, як саме формується мовлення у людини і розглянемо кожен етап формування мовлення.

Для того щоб людині сказати будь-яку інформацію використовуючи мовлення, ці думки необхідно сформувати у головному мозку, як деяку думку тексту, яку людина хоче донести через мовлення. За цей процес у людини відповідають дві ділянки мозку це ділянка Брока та ділянка Верніке, розглянемо ці дві ділянки мозку більше детально та розберемо за що вони відповідають. Почнемо з ділянки Верніке, ця ділянка відповідальна за інтерпретацію як і вербальних так і невербальних даних людського спілкування, отже ця ділянка також відповідальна за інтерпретацію і людського мовлення. Завдяки медицині, а саме феномену афазії Верніке, відомо, що ця ділянка окрім інтерпретації також і відповідальна за формування думки того що саме хоче сказати людина. Люди з цією хворобою можуть формувати слова але не можуть формувати логічного мовлення і їх розмова має вигляд набору випадкових слів. Ця ділянка мозку може формувати думку мовлення використовуючи думки з усього іншого мозку, тобто людина може вільно висловити будь-що формуючи це зі своїх думок або отриманої інформації з органів відчуття. Потім сформована

думка що саме хоче висловити людина перенаправляється у ділянку Брока, через дугоборазний пучок, яка відповідальна за керування людським мовним апаратом та власне формування звуку людської мови з думки, шлях взаємодії зон формування мови продемонстрована на рисунку 1.1. Цю ділянку також як і ділянку Верніке виявили завдяки хворобі афазії Брока, при якому людина може розуміти мову та формувати думку мовлення, але механічно не може сформувати звук мовлення або може вимовити лише деякі звуки або слова.

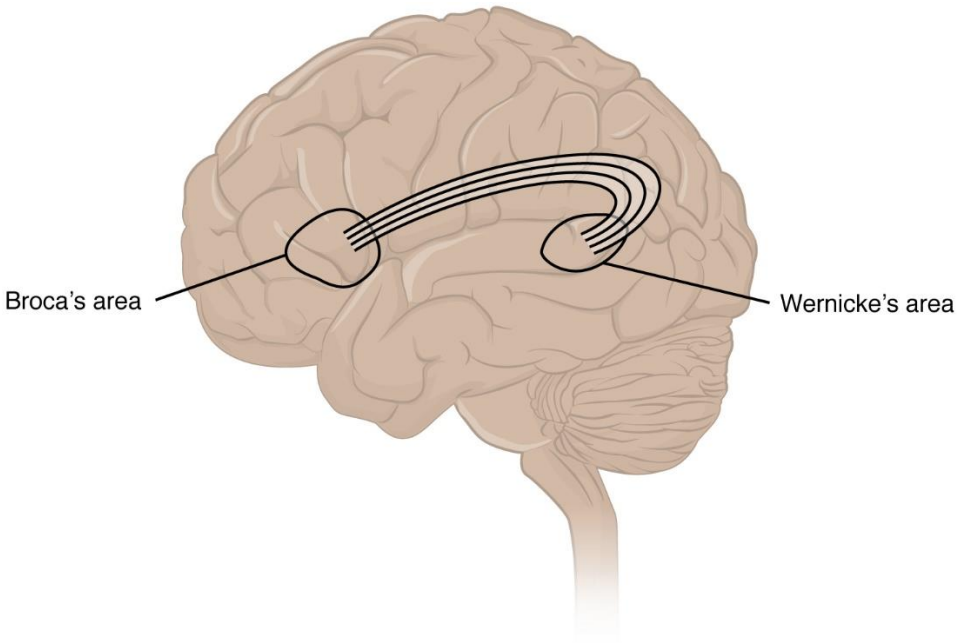


Рисунок 1.1 – Схема взаємодії зони Вернеке з зоною Брока через дугоборазний пучок

Маємо загальну картину, що у людини є два окремих відділи мозку, які відповідальні за мовлення, відділ Верніке відповідальний за обробку думок мовлення, а відділ Брока за механіку мовлення. З цих даних необхідно висновок, для архітектури синтезу мовлення, що модель синтезу мовлення, не буде симулювати ділянку Верніке, хоча це і важлива ділянка для людського мовлення, а необхідно лише симулювати ділянку Брока, яка відповідальна за механічне формування мови, так як для моделі завжди буде

подаватися вже сформована думка у вигляді тексту, для якого необхідно синтезувати мовлення.

Визначивши як саме формується мова у людському мозку далі необхідно визначити чим саме керує ділянка Брока та як формується звук людського мовлення. Мовний апарат людини це великий комплекс різних органів людини, які зв’язані між собою та формують людське мовлення, умовно людський мовний апарат ділять на 3 частини, це джерело енергії, фонетичний апарат та артикуляційний апарат. Найпростіше це джерело енергії для мовлення, це лише видих повітря з легень, і тому весь мовний апарат, знаходиться у дихальних шляхах.

Розберемо фонетичний апарат, його задача це створення звуку фонем. Фонетичний апарат складається з язика, губ, щелепи, а його найголовніша частина це голосові зв’язки. Голосові зв’язки виступають резонатором базової хвилі, з яка утворюється фонеми. Голосові зв’язки змінюючи натяг та просвіт між зв’язками, можуть змінювати те як спілкується людина, наприклад розмова пошепки, це зменшення амплітуди всіх фонем, яке відбувається завдяки зменшенню енергії видиху з легень та зміни положення натягу зв’язок. Голосові зв’язки самі можуть формувати всі голосні фонеми, а приголосні фонеми зазвичай формуються з використанням вже органів у ротовій порожнині, наприклад звук фонеми «к», утворюється завдяки тому, що язик швидко закриває дихальний шлях, і потім швидко його відкриває.

Кожна людина має однаковий принцип формування звуку фонем завдяки фонетичному апарату, але при цьому кожна людина має свій унікальний голос, за формування унікальної мелодії звуку мовлення людини відповідальний артикуляційний апарат. Артикуляційний апарат у медицині включає у себе і фонетичний апарат, але ми його розглянули окремо для того щоб зрозуміти як формуються базові фонеми. Артикуляційний апарат це комплексна система, яка включає у себе багато різних органів, які продемонстровані на рисунку 1.2. Ці органи мають

невеличкі відмінності у кожній людині, які змінюють індивідуально під кожну людини голос, наприклад, товщина голосових зв’язок, яка відповідає за басистість голосу, форма порожнини носоглотки які формують резонанс голосу, функція праці язика, яка відповідає за більшість приголосних фонем.



Рисунок 1.2 – Загальна схема артикуляційного апарату людини

Отже крім того що для моделі штучного синтезу голосу, необхідно симулювати керування мовним апаратом, йому також необхідно симулювати і сам мовний апарат, який є дуже комплексною системою, де між собою взаємодіють велика кількість факторів та органів для формування унікальної мелодії голосу.

1.2 Структура та частини людського мовлення

Мовлення, як і мовний апарат, теж є комплексною системою. Мовлення складається з поєднань таких фонетичних явищ, як, вимовляння фонеми та їх коартикуляції, наголос на слова, тембр та гучність мовлення, швидкість мовлення, інтонація та тон мовлення, ці фактори разом складають просодію мовлення. Завдяки просодії мовлення, саме мовлення і відчувається живим, через нього окрім текстової інформації та її нюансів,

наприклад поставлення питання, можливо зрозуміти контекст тексту або, що важливіше, передати емоції людини, її настрій людини або як вона до вас відноситься, і так просодія мовлення формує унікальну рису ідентичності людини. Тому для якісного відтворення синтетичного мовлення, кожну з цих частин необхідно розібрати та проаналізувати, як вони утворюються та як загально впливають на звучання мовлення.

Базою будь-якої форми вербального мовлення людини, є фонеми, найпростіші фонетичні звуки, вони формуються завдяки голосовим зв'язкам. Людина може вимовити обмежену кількість фонем, а саме 150 фонем. Кожна мова завжди використовує фонеми з міжнародного фонетичного алфавіту, який продемонстрований на рисунку 1.3.

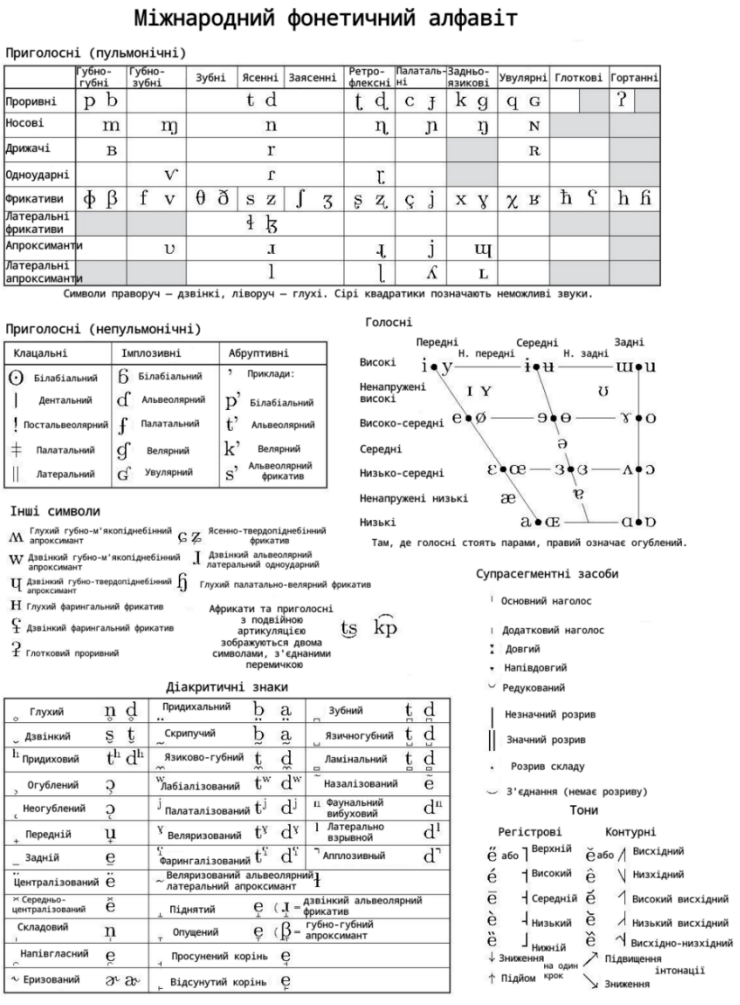


Рисунок 1.3 – Міжнародний фонетичний алфавіт

Отже це обмежена кількість фонем дає можливість їх відтворювати методом синтезу, так одні з перших моделей синтезу використовували, заздалегідь записані окремі фонеми і потім їх складала згідно з фонетичної транскрипції слова, але результат синтезу був дуже нерозбірливим та звучав дуже схоже на звук робота з фільмів. Це відбувалося як раз через відсутність інших частин просодії мовлення, але саме нерозбірливість мовлення відбувалася через відсутність коартикуляцій фонем. Коартикуляція фонем – це явище при якому звук однієї фонеми впливає на звучання наступної фонеми у слові. Наприклад якщо вимовити слово «дуб», то звук фонеми д, буде м’яким, а якщо вимовити слово «двері», то звук д, буде більш твердим. Тобто лише явище коартикуляцій, вже робить неможливим вирішення проблеми синтезу мовлення лише через просте складання звуку фонем у мовлення, бо звучання стає нерозбірливим та неприємним на слух. Також важливим фактором є наголос у слові, бо з його зміною окрім натурального звучання мовлення також залежить сенс слова наприклад це слово, за́мок та замо́к, один значить замок, як захисну споруду, а інша має значення замка яким хочуть зачинити якийсь простір. Тому модель ще повинна буде зрозуміти у контексті розмови, про який саме предмет іде мова і правильно поставити наголос, щоб не загубити сенс вкладений у текст.

Далі розберемо, явища, які не дають голосу людини звучати монотонно та допомагають передати емоції у мовлені. Почнемо розбір з гучності мовлення, це найпростіший аспект який може змінити монотонне мовлення. Так через гучність мовлення можливо передавати емоції, найпростіший приклад, це емоція гніву, коли людина гнівається вона зазвичай підвищує гучність свого голосу ніж у спокійному стані. Крім гучності є ще базовий метод, щоб висловити свій стан, це швидкість мовлення. Швидкість мовлення також може показувати стан людини, наприклад люди у збудженому стані будуть більш швидше розмовляти, і це працює також навпаки, коли людина знаходиться у подавленому стані. Швидкість та гучність голосу також можливо змінювати у межах мовлення

одного речення. Швидкість мовлення можливо змінювати для того щоб промовити паузу між реченнями або для придання деяким частинам речення емоційного відтінку. У кожної людини також присутній свій ритм мовлення, який придає її унікальні риси голосу, але і ритм також може змінюватися у залежності від стану людини. А при поєднані зміни швидкості та гучності можливо висловити інтонацію. Інтонація виконує функцію передачі емоційного забарвлення тексту який вимовляється, також інтонація використовується для того щоб поставити питання, що не завжди можливо висловити використовуючи лише монотонне мовлення, без використання додаткових слів, які можуть вибиватися з загального ритму мовлення та звучати ненатурально.

І останній розділ який формує просодію мовлення, це унікальні риси голосу, це тон та тембр. Тон та тембр не впливають на лексику та синтаксис речення яке промовляється, їх задача це передати емоційне забарвлення та передати унікальний для людини характер розповідача через ці аспекти мовлення. Тон мовлення – це основна частота основної звукової хвилі, яку формує голосові зв’язки. Частота тону мовлення дуже змінюється у залежності від статі людини, так для жінок, діапазон частоти тону варіюється від 80 до 120 Гц, а для чоловіків від 150 до 320 Гц, тобто за тоном мовлення можливо відрізнити стать людини за мовою. Якщо за тоном голосу можливо відрізнити стать розповідача, то персону людини можливо відрізнити за тембром голосу. Тембр голосу фонетика описує як унікальний колір голосу кожної людини, саме тембр голосу є найвагомішою частиною просодії мовлення і завдяки цьому дає можливість мозку відрізняти людей один від одного за голосом. Тембр формується з тонів та обертонів, і саме велика палітра обертонів, які формуються усіма органами артикуляційного апарату і дають унікальне звучання завдяки тембру кожній людині. Завдяки тембру передається характер розповідача, якщо тембр голосу буде доволі жорстким, то він і буде у людини на підсвідомості асоціюватися з доволі сильною та жорсткою людиною. Тембр завжди на підсвідомості буде

асоціюватися з тієї чи іншою рисою характеру людини або повністю описувати її характер, навіть якщо насправді ця людина має інший характер. Отже якщо тембр асоціюється з рисами характерів то й палітра тембрів які мозок асоціює з ними є теж широким. Загалом для кожної статі виділяють по 3 типу тембру голосу, які відображають за собою загальну висоту звучання, але кожна людина має свій тембр у діапазоні цих 3 тембрів.

Розібравши основні фонетичні складові мовлення та з чого формується просодія мовлення формується висновок, що синтез природнього людського мовлення є дуже тяжкою задачею, яка вимагає симуляції всіх особливостей артикуляційного апарату та симуляцію складного контролю над артикуляційним апаратом, для того щоб слухач міг повірити у природність синтезованої мови та зміг сформувати образ людини яка стоїть за мовленням, як зі справжньою людиною. Кожна частина артикуляційного апарату вносить хоч і малий внесок у загальне звучання, але зміни будь-якого органу артикуляційного апарату до загального звучання мовлення будуть завжди відчутні для слуху людини, через унікальну та складну природу людського мовлення.

1.3 Етичні та правові виклики синтезу мовлення

Перед тим як почати розбирати технічні аспекти функціонування синтезу мовлення, необхідно розглянути етичну та правову сторону використання цієї технології. Ці проблеми з’являються через перспективу подальшого розвитку синтезу мовлення до рівня невідрізного від людського голосу та розвитком технологій клонування голосу та більш гнучких налаштувань синтезованого голосу. Уся етична проблема іде від одного джерела, це хто саме є власником голосу, як і синтезованого так і скопійованого і як цим голосом користуються. Так дослідження [1] показує, що мала кількість людей здатна відрізнити синтезований голос від натурального, тож постає питання, а як саме відрізнити синтезований голос

від звичайних, бо сучасні моделі, мають мало маркерів [2], за якими можливо було відрізнити підробку від оригіналу. Так вже є факт [3] того що моделі синтезу мовлення допомагають шахраям обманювати людей, за допомогою скопійованих голосів, нехай то може бути і родичі так і просто відомі люди, через що люди стають більш вразливі до шахрайських дій. Вже розробляються закони для захисту права людини на голос, так наприклад вже є прецедент з Elvis Act у штаті Теннессі. І для виконання цих законів, також окрім стягнення штрафу або відповідальності для правопорушника необхідно також реалізувати механізм того, щоб більше неможливо було використовувати захищений голос. І тут стає проблема, оскільки наразі [4] не існує реальних механізмів, щоб було можливо забути вже навчений голос, окрім як повного скиду даних всіх вагів у моделі, тобто просто повне видалення моделі.

1.4 Методи збереження та обробки аудіо-даних

Перед розглядом того, які саме є вже існуючи методи синтезу мовлення, розбір їх функціоналу та впливу на загальні методи синтезу, необхідно розібрати фундамент з яким працюють ці моделі, а саме, як програмно обробляється аудіо-дані. У цьому розділі буде розібрано, як комп’ютер представляє аудіо-дані та які є методи для аналізу даних та здобуття необхідних ознак аудіо-даних.

Звукова хвиля є послідовністю коливань повітря, які людський слуховий апарат може сприймати та перетворювати у корисні дані, як дані отримані з розмови, або дані про оточуюче середовище. Так як звукова хвиля має особливість приймати лише одну частоту на дуже маленькому відрізку даних, то цієї властивістю користуються для запису аудіо-даних. Так як аналогові звукові хвилі мають безперервну природу, це стає проблемою для комп’ютера, бо для збереження будь-яких даних, як правило, комп’ютеру необхідна дискретизація даних, у випадку з аудіо-

даними також відбується дискретизація і визначається частота семплів. Так від частоти семплів залежить скільки змін частоти коливання хвилі може бути записано у секунду, стандартом частоти семплів є 44100 семплів, хоча нейромережі можуть використовувати і іншу частоту семплів такі як 16000 так і 32000 для економії ресурсів під час навчання та генерації.

Хоча традиційні методи синтезу мовлення мали великий спектр використовуваних аналітичних маркерів та показників аудіо-даних, у цій роботі буде розібрано лише ті які використовуються у сучасних методах інтелектуального синтезу даних. Основним перетворенням, яким користуються для синтезу мовлення, є перетворення з частоти у герцах у значення мела. Мел представляє собою показник того, як сприймає людський слух різницю між частотами коливань звукової хвилі. Факт того, що людина не сприймає лінійно різницю між частотами коливань давно відомий у сфері музики, це проявляється як раз у різниці частоті між нотами. Якщо між нотами буде лінійна різниця частот, то ноти з все більшою частотою будуть менш відрізними на слух між собою, тому у музиці є логарифмічна гармоніка між нотами, це все зумовлено особливостями сприйняття звуку людським слухом.

Формула перетворення значень частоти звука у герцах у мели має такий вигляд:

$$m = 1127 * \ln(1 + \frac{f}{700}), \tag{1.1}$$

де f – частота звуку, Гц.

Для параметричних моделей, постає важливе питання, як саме представляти аудіо-дані для проведення навчання, на що саме спиратися моделі на відтворення. Надефективним методом, було передача моделі повністю сирий файл аудіо-даних з усіма семплами, попередньо перетворених у мели для покращення натуральності звучання симуляції мовлення, але тоді постає проблема ефективності, через високу кількість

семплів необхідних для якісного зберігання аудіо-даних. Рішенням є дискретизація аудіо-даних, та перетворення їх у спектрограму, приклад звичайної спектрограми запису промови на рисунку 1.4.

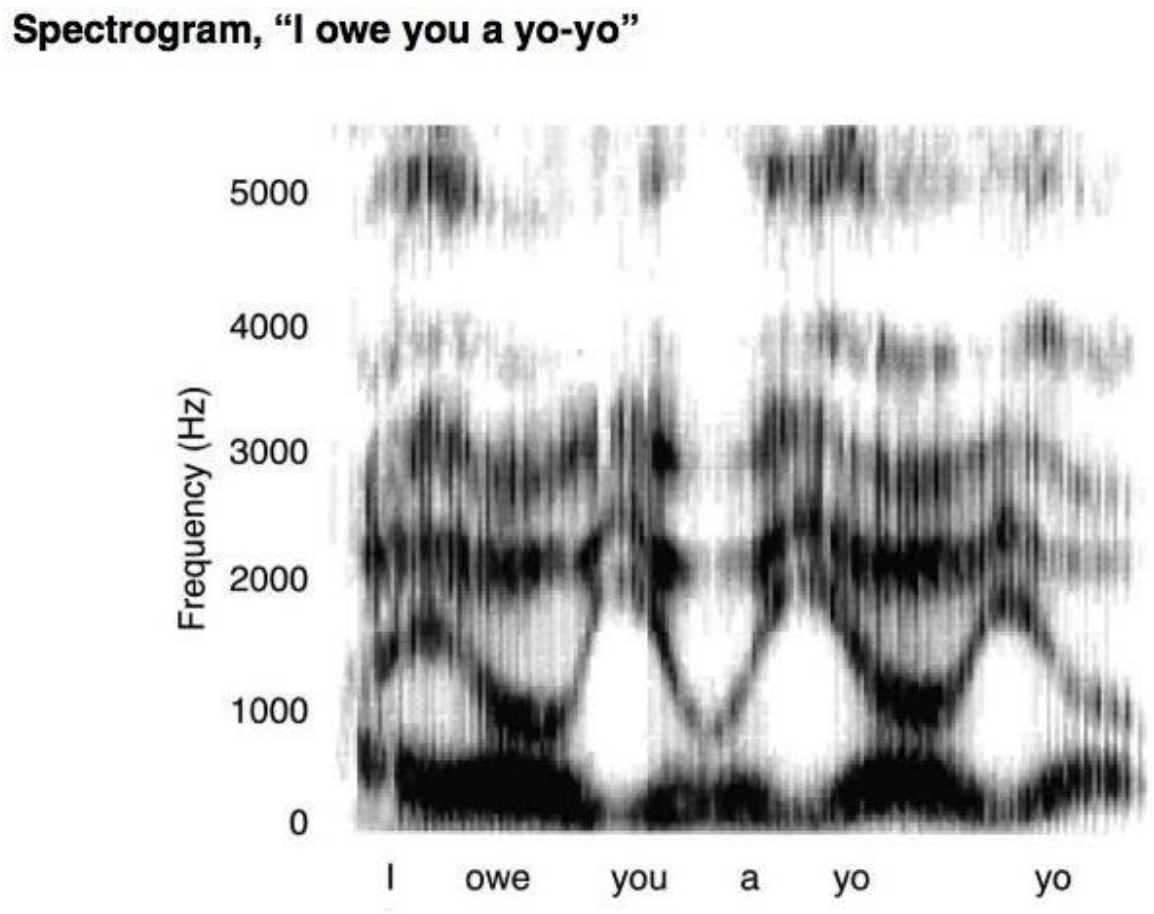


Рисунок 1.4 – Спектрограма промови речення «I owe you a yo-yo»

Спектрограма дозволяє отримати візуальне представлення аудіо, що надає можливості для додаткового аналізу аудіо-даних. Також спектрограму можливо перетворити на шкалу мела і відобразити спектрограму мела, для збереження того, як людина сприймає на слух аудіо-дані. Так як же є добре відомі практики перетворення зображення у формат, який може обробляти нейронна мережа, то перетворення аудіо-даних спектрограму дозволяє використовувати схожі методи передачі даних у нейромережі, зі схожими техніками як з даними зображення.

Розглянемо процес формування спектрограми мела з аудіо-даних. Процес формування спектрограми починається з створення фреймів аудіо-даних, тобто розділення аудіо-даних на рівні між собою проміжки. Розділення аудіо-даних на фрейми має вимоги від перетворення Фур'є, яке буде розглянуте далі, ці вимоги потребують того, що фрейми між собою повинні мати відстань у половину фрейма, і пересікатися між собою. Такі вимоги зумовлені ефектом спектрального витоку та його рішенням, що зображено на рисунку 1.5.

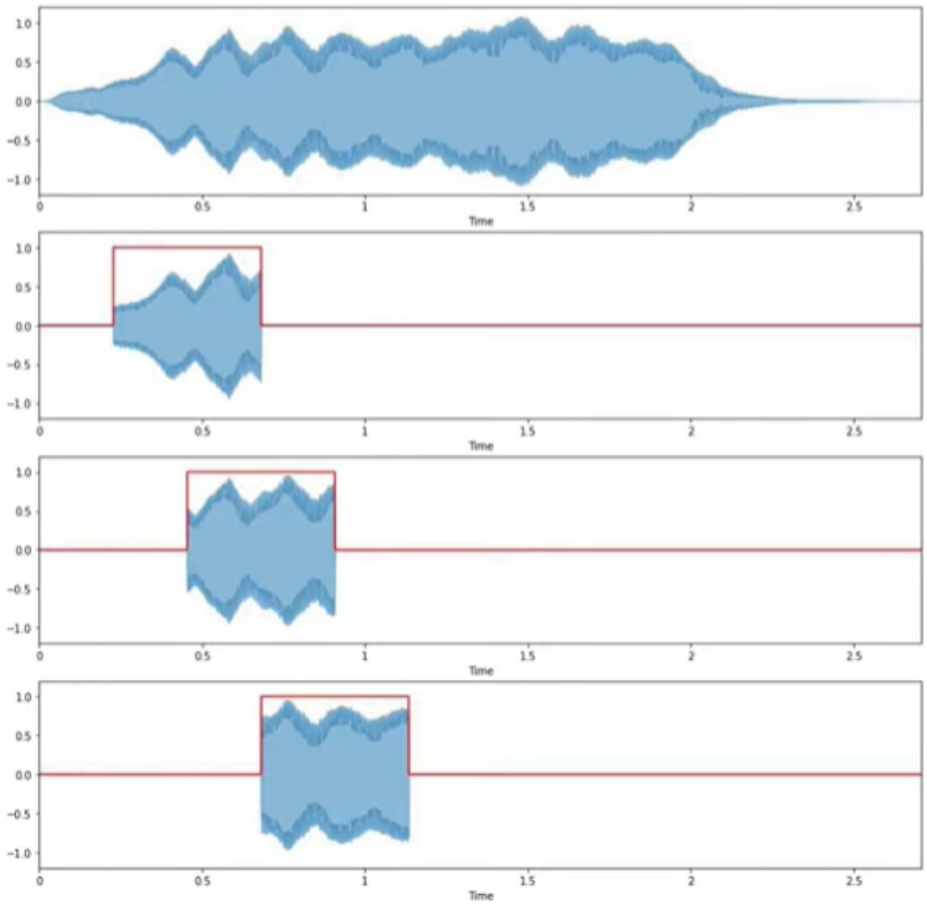


Рисунок 1.5 – Поділення фреймів та їх перетин

Спектральний виток це особливість обробки алгоритмом дискретного перетворення Фур'є фреймів функції. Якщо взяти фрейм, який має у собі дані про функцію синусоїди трохи більше ніж один періоду самої функції

синусу, та передати його у функцію дискретного перетворення Фур'є, то дискретне перетворення Фур'є буде невірно інтерпретувати періодичність функції, як це показано у рисунку 1.6. Внаслідок невірної інтерпретації синусоїдної функції та виконання дискретного перетворення Фур'є буде отримано спотворений результат з розкладання функції на складові частини.

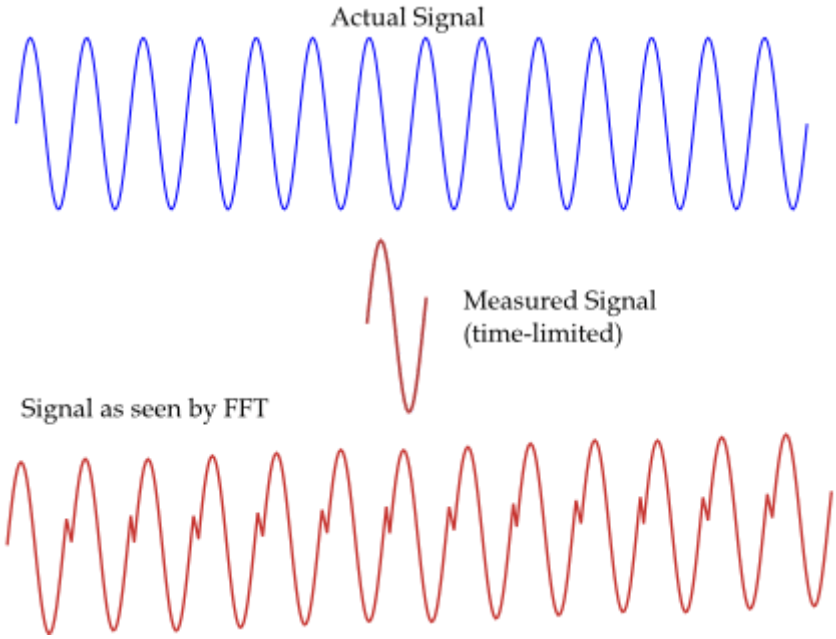


Рисунок 1.6 – Графічна демонстрація спектрального витоку

Рішенням спектрального витоку є перетворення даних у фреймі завдяки функції пом'якшення даних, а саме функції віконного перетворення Ханна. Формула віконної функції Ханна має такий вигляд:

$$w(k) = 0.5 * (1 - \cos(\frac{2\pi n}{N})), 0 < n < N, \quad (1.2)$$

де n – це позиція відліку дискретної частини вікна;

N – це дискретна довжина вікна.

Ця функція перетворює дані так, що вони набувають лінійного виду ближче до початку та кінця фрейму, що робить функцію, завжди гладкою

без скачків на краях фрейму, зміни фрейму після використання віконної функції наведені у рисунку 1.7.

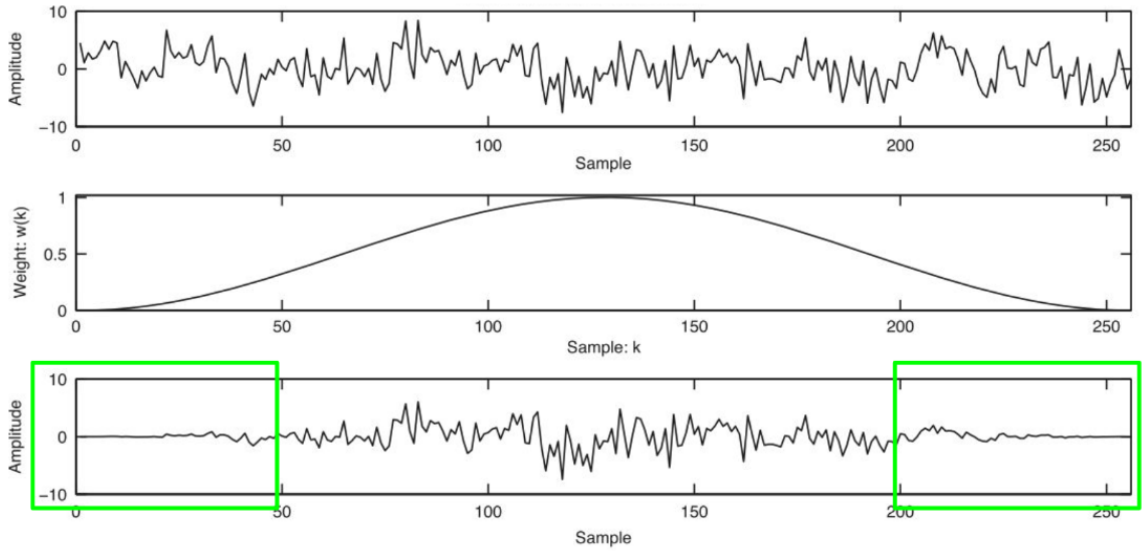


Рисунок 1.7 – Графіки фрейму, віконної функції та злагодженого фрейму після використання віконної функції

Використання віконної функції Ханна прибирає артефакти, які виникають під час перетворення Фур'є використовуючи фрейми. Через використання віконної функції Ханна, на краях фреймів через злагодження втрачалися корисні аудіо-дані, і для компенсації цієї втрати, фрейми між собою пересікаються, і так зберігають дані для дискретного перетворення Фур'є прибираючи артефакти від поділення даних на фрейми.

Після розбиття даних на фрейми, ці фрейми передаються у функцію дискретного перетворення Фур'є і отримується спектр частот та їх потужності у фреймі. Ці фрейми потім складаються згідно з частотою фреймів у секунду і утворюють спектрограму частот аудіо-даних.

Після утворення частотної спектрограми необхідно створити фільтри мела, для дискретного перетворення частоти у спектрограмі у шкалу мела. Необхідність створення фільтрів мела, також як і існування одиниці мела, зумовлена особливостями сприйняття звуку людським слухом. Людський

слух відрізняє чітко між собою лише деякі частоти, наприклад такі частоти як 14000Гц та 14100Гц, на слух будуть звучати майже однаково, а от різниця між 100Гц та 200Гц буде відчутна на слух. Через цю різницю, фільтр мела має трикутний графік функції, який показує як відчутна різниця на слух деякої визначеної фільтром частоти між суміжними частотами. Кількість фільтрів зазвичай обирають від 30 до 120, їх відносне положення між собою є логарифмічним і визначається за формулою перетворення з частоти у герцах мели, графічний вигляд фільтрів мела наведений у рисунку 1.8.

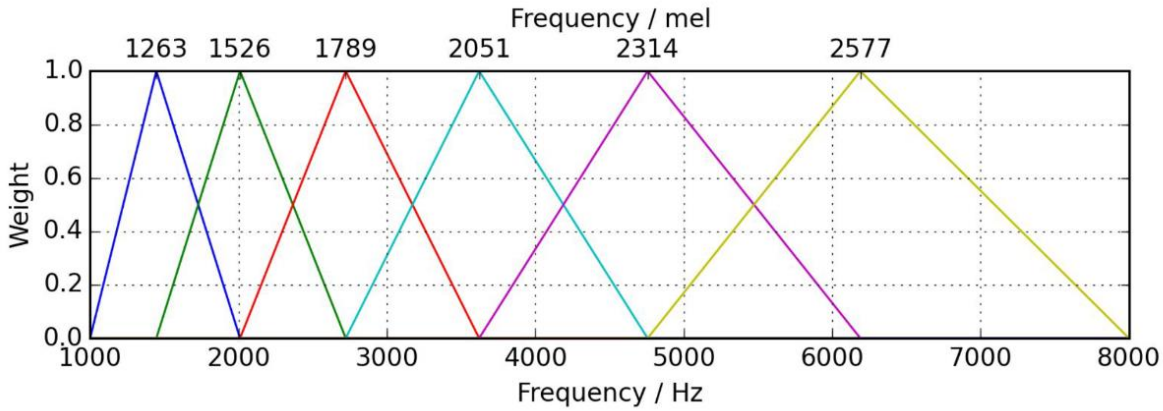


Рисунок 1.8 – Графік трикутної функції 6 мел фільтрів

Після перетворення даних через фільтри мела, усі дані перетворюються з герців у мели, за вище зазначеною формулою. Після всіх перетворень формується матриця потужності у децибелах до мелів за визначеним проміжком часу, яка називається спектрограмою мела. Цей формат даних відображає усі необхідні дані, як людський слух сприймає аудіо-дані у часовому проміжку програвання цих аудіо-даних.

1.5 Попередня обробка вхідних даних до моделі

Перед тим, як розбирати вже існуючі методи, розберемо загально, як у загальних випадках відбувається процес синтезу мовлення.

Передбачається, що у деяку програму буде введено текстові дані, а програма повинна синтезувати мовлення на основі введеного тексту. Але як було вже розглянуто, звичайний текст не завжди повністю відображає для програми всі фонетичні дані тексту, якщо дати програмі набір сирих текстових символів, то буде втрачено багато фонетичних особливостей, які будуть істотно впливати на кінцевий результат. Тому завжди у кожній традиційній програмі, є окремий компонент нормалізації текстової інформації та фонетичний транскриптор нормалізованого тексту, який сформує зрозумілу для програми фонетичну транскрипцію. Текстова нормалізація це компонент, який перетворює символьні описи у тексті, як речення, які складаються лише з літер абетки, без втрати сенсу тексту, наприклад при нормалізації тексту, таке речення, як «С-нт. Цариценко» перетворюється у «Сержант Цариценко». Нормалізація тексту також має свої труднощі, які пов’язані з контекстом у реченні, наприклад програмі необхідно правильно зрозуміти, як нормалізувати такий текст, як «Необхідно закупити 5100 банок» у речення «Необхідно закупити п’ять тисяч сто банок», а не як «Необхідно закупити п’ять тисяч соту банок». Тому рішення з використання словника використовувалося» лише на початку вивчення синтезу мовлення, а більш пізні програми для нормалізації, вже використовували контекстуальний аналіз тексту. Найскладнішою частиною обробки тексту, є складення фонетичної транскрипції, оскільки немає єдиного універсального рішення для кожної мови, бо кожна мова має свої фонетичні особливості. Наприклад у українській мові, є явище пом’якшення приголосних, тобто виходячи з контексту слова, необхідно у фонетичній транскрипції буде пом’якшувати визначену приголосну, приклад легко, то його транскрипція буде, легко. Але якщо з фонетичною транскрипцією української мови більш легше, бо правила доволі транскрипції доволі не змінюються від контексту речення, то у англійській мові, як раз існує така особливість. Наведемо приклад зміни вимовляння слова у залежності від контексту, так маємо слово bow, яке змінює значення у залежності від контексту речення і змінює

і фонетичну транскрипцію, так у реченнях bow tie (краватка-метелик), з транскрипцією / boʊ taɪ / і інша інтерпретація bow down (вклонитися) з транскрипцією /baʊ dɔːn/, слово bow матимемо різну транскрипцію, отже і повинен мати інше синтезоване вимовляння. Проблема фонетичної транскрипції вирішується зазвичай двома шляхами, або використовується словник, у якому вже записані всі фонетичні транскрипції слова, або використовується навчена модель фонетичної транскрипції, яка представляє собою модель машинного навчання яка навчена на формування фонетичних транскрипцій, зазвичай використовують комбінований алгоритм, який спочатку шукає фонетичну транскрипцію у великому словнику, а якщо такого слова немає, то відбувається фонетична транскрипція використовуючи модель машинного навчання.

Маючи біологічні та фонетичні основи функціонування мови, дає розуміння що є можливість синтезу мовлення програмою. Та для цього необхідно розуміння того, як саме необхідно обробляти текст для того, щоб програма могла з ним коректно та ефективно працювати з текстовими даними. Також необхідно мати механізми, того, як саме необхідно працювати з аудіо-даними та перетворювати їх у значення, які відображають, як людина сприймає на слух звук. Ці дані дають необхідні основи для функціонування будь-якої системи синтезу мовлення. Жодна система мовлення не може обійтися без вище описаних основ і це довела перша система синтезу мовлення, а саме синтезу формантами, бо якщо видалити, хоча б одну з вище описаних основ з цієї системи, то її функціонування стане технічно неможливим.

2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ СИНТЕЗУ МОВЛЕННЯ

2.1 Синтез мовлення методом формантів

Синтез методом формантів є першою у світі архітектурою для синтезу мовлення. Саме цей традиційний метод синтезу мовлення, дав розуміння як використовувати обробку тексту, та як необхідно вибудовувати взаємодію між компонентами синтезу мовлення, спираючись на біологію мовного апарату, та як взагалі синтезувати аудіо-дані мовлення, використовуючи знання про обробку аудіо-даних та як працює фонетика. Синтез формантів, як слідкує з назви, використовував для синтезу, форманти. Конкретно для розбору синтезу формантів, для аналізу використовувався DECTalk [5]. Для синтезу формантів та будь-яких маніпуляцій з даними, необхідно мати пристрій або програму для зміни та оперуванням аудіо-даними, такий пристрій звуть вокодером і з його створенням і став можливий синтез мовлення і вокодер і досі використовується для синтезу мовлення у передових моделях синтезу мовлення. Тоді використовувався вокодер, який за принципом дії був доволі схожим з синтезатором, вокодер оперував деякої базовою звуковою хвилею, яка мала гармоніку та заздалегідь визначену частоту, вокодер обробляв цю частоту за допомогою набору математичних фільтрів які змінювали характеристики хвилі і його звучання. І правильне налаштування фільтрів вокодера дозволяло змінювати частоту формантів, що і стало основним принципом який дозволив створити систему синтезу мовлення. Вокодер для синтезу мовлення оперував архітектурою «Джерело-фільтр», це значить, що вокодер оперував вже синтезованою базовою аудіо-хвилю з необхідними гармоніками та деякою зафіксованою частотою, для того щоб симулювати тон голосу, створення базової хвилі симулювало принцип роботи голосових зв’язок, і ця базова частота потім проходить через один або комплекс фільтрів вокодера для симуляції звучання фонем, проходження через фільтри є симуляції того, як

базова звука хвиля проходить через артикуляційний апарат людини і формує мовлення. Тоді для налаштування того, як вокодер обробить базову звукову хвилю для синтезу фонеми, розробникам необхідно було дослідити кожну симульовану фонему та визначити на якій частоті знаходяться форманти при вимовлені фонеми. З отриманих даних про частоти формантів формувалася система з правилами, яка описувала, які фільтри повинен використовувати вокодер та як їх налаштовувати для відтворення частоти формантів і синтезу звуку фонеми.

Цей метод синтезу мовлення був актуальним весь період 20-сторіччя, і використовується навіть у сьогоденні, як дуже простий та дешевий спосіб синтезу мовлення для людей які були позбавленні можливості розмовляти або пошкодили голосові зв’язки, так наприклад саме завдяки цьому методу синтезу мовлення і міг спілкуватися вчений Стівен Гокінг. Хоча синтез формантів і дозволяв створити мовлення, яке могла зрозуміти людина, але він мав свій спектр недоліків. Так для налаштування голосу, та тону, необхідно було мати експертні знання з того, як працюють форманти, вокодер та синтез базової аудіо-хвилі. Також сама архітектура вокодера, з тим що це набір фільтрів звукової хвилі, які використовують математичні функції, не дає можливості добре симулювати артикуляційний апарат, оскільки окремі органи артикуляційний апарат описуються математичними формулами дуже тяжко і також вокодер обмежений лише зміною обмеженої кількості формантів, їх зазвичай було 5, і з цього виходить головний недолік синтезу мовлення формантами. Головний недолік синтезу формантів це те що просодія мови була дуже недовільної якості.

Синтез формантів дозволяв змінняти тон мовлення, завдяки зміни частоти базової звукової хвилі, що дозволяло вже тоді симулювати деякі тембри, наприклад басистий чоловічий голос або жіночий тон, але його звучання було дуже неприроднім. Мовлення синтезоване методом було монотонним та мало звучання із характерним електронним шумом і його загальне характерне неприродне звучання потім було названо

роботизованим звуком і у людей це характерне звучання через масову культуру закріпилося із машинами та роботами. Через машинне та неприродне звучання люди не сприймали синтезоване мовлення так же як і натуральне, через це задача усіх наступних моделей синтезу мовлення було досягнення натуральності звучання та відтворення просодії для синтезованого мовлення.

Хоча синтез формантів і вважається застарілим методом синтезу мовлення, цей метод має вплив на більшість наступних моделей для синтезу мовлення. Більшість моделей для синтезу використовують архітектуру синтезу формантів «Джерело-фільтр», бо вона і є анатомічно обґрунтованою, що є важливим критерієм, бо голос людини дуже сильно залежить від його анатомії мовного апарату, і ця архітектура довела свою спроможність генерувати чіткі для роздільні для слуху людини фонеми і виходячи з цього слова та мовлення. І з синтезу формантів також пізніше переосмислять та покращать ідею вокодерів для того, щоб покращити якість отриманої аудіо-хвилі синтезованого мовлення.

2.2 Альтернативні традиційні методи синтезу мовлення

Синтез формантів є домінуючим традиційним методом синтезу мовлення. Перевага синтезу формантів є у тому, що він є доволі простим методом синтезу, який дозволяє формувати зрозуміле мовлення, хоча поганої якості. Хоча і були спроби покращити якість синтезованого мовлення та прямування до натуральності просодії мовлення. Надалі і було багато різних направлень і способів покращення якості синтезованого мовлення. З цих напрямів можливо виділити два головних, це статистичний метод та конкатенаційний.

Статистичний метод, працює на базі моделі скритих марківських ланцюгів [6]. Цей метод буде формувати фонеми за схожим принципом з синтезом формантів, тобто буде формуватися базова хвиля і яка потім буде

передаватися у вокодер з деякими параметрами налаштування фільтрів і у кінці повинна сформуватися звукова хвиля фонеми. Відмінність статистичного методу від синтезу формантів у тому, що модель не потребує запису того, які форманти повинні бути застосовані для генерації якоїсь фонеми, модель сама навчається налаштовувати фільтри вокодера та генерувати необхідну базову хвилю. Статистичний метод складався з 2 моделей, 1 модель це передбачення того, скільки часу повинна звучати фонема, а 2 модель є акустичною, її задача було формування налаштувань вокодера базової хвилі та гучності аудіо-виходу. Так вокодер зміг краще відтворювати елементи просодії мовлення, результат синтезу голоса більше не звучав монотонно, хоча синтезоване аудіо все ще мало характерне звучання вокодера, яке було дуже відчутне для людського слуху. Також цей метод став одним з перших, який вже починав потребувати для своєї праці навчальних даних, які для моделі передавалися у вигляді пари, текст-спектрограма, це ускладнення вже давало тоді можливість моделі симулювати найпростіші елементи просодії мови, такі як базове звучання тонів та тембру голосу, що наприклад для синтезу формантів потребувало втручання та налаштування від висококваліфікованого експерту, який би налаштовував форманти окремо для кожної існуючої фонеми у мові. Цей метод синтезу тексту мав великий вплив для подальшого розвитку інтелектуальних методів синтезу тексту, так як це був взагалі перший метод, який показав можливість відтворювання акустичних даних через параметричні моделі, і показав, що можливо використовувати моделі машинного навчання для синтезу мовлення, також важливим внеском було результат того, що можливо використовувати модель Джерело-фільтр для синтезу мовлення використовуючи моделі машинного навчання, чим у подальшому також будуть користуватися сучасні моделі для синтезу мовлення.

Конкатенаційний метод базувався на методах динамічного програмування, та представляв собою програму, як підбирала найбільш

схожі заздалегідь записані фонеми у базі даних. База даних, для взяття фонем, мала розмір у 30–40 годин записаних фонем, у різних коартикуляційних зв’язках та інтонаціях. Цей метод, хоча і виглядає доволі простим у реалізації та мав доволі високу якість просодії, тому що оперував натуральними фонемами, мав великий недолік саме у своїй архітектурі. Так конкатенаційний метод не міг синтезувати мовлення, яке виходило за межі записаних фонем, оскільки, програмі немає з чого брати фонеми для синтезу мовлення. Також цей метод потребував доволі складної праці від акторів озвучення, оскільки необхідно було записати дуже багато різних фонем та у різних коартикуляціях та інтонаціях.

Усі розглянуті вище методи, або відмовлялися від концепції саме синтезу мовлення та просто вдало підбирали фонеми з існуючої бази даних, або намагалися синтезувати мовлення та були обмежені у якості синтезу, через складну природу мовлення та чіткості слуху людини на велику кількість дрібниць у звучанні мовлення. І через велику комплексність мовлення, та його представлення у аудіо-даних, просто не вистачало налаштувань вокодера. Так при синтезі формантів вокодер хоча і міг симулювати форманти, але на цьому його можливості і закінчувалися, він не міг симулювати дрібні зміни коартикуляційного апарату, через обмеженість налаштувань вокодера. Як раз статистичний метод це також підтверджував, бо він симулював більше параметрів, ніж просто форманти, але якість не була натуральною. Усі ці фактори наводили на дві проблеми, велика кількість параметрів необхідних для симулювання базової частоти та необхідна сильна комплексність вокодера, для більш якісної просодії мовлення. Методи глибокого навчання, як раз і розвивалися для вирішення задач, як потребувала великої кількості параметрів, які традиційними методами, було доволі довго та складно реалізовувати. Тому використання нейронних мережей для синтезу мовлення є ефективним рішенням цієї задачі.

2.3 Використання вокодерів на базі нейромережей

Класичний вокодер хоча і є комплексною та складною програмою, яка ефективно допомагає взаємодіяти та перетворювати аудіо інформацію, вона не може повністю удовольнити всі необхідні малі звукові перетворення які формує артикуляційний апарат і через які, як виявилося, людина може відрізняти натуральність голосу мовлення. Єдиною альтернативою використання вокодера, було альтернативний метод генерації мовлення, конкатенаційний метод, який використовував динамічне програмування для правильного підбору фонем при генерації мовлення, хоча результат синтезу мовлення, але цей метод де-факто лише займався підбором схожих фонем і через це був обмежений у можливостях синтезу лише своєю базою фонетичних даних та коартикуляцій. Вирішенням обмеженості вокодера, стало використання нейромереж для обробки аудіо-даних у якості вокодера, такою першою альтернативою стала нейромережа WaveNet [7].

2.3.1 WaveNet

WaveNet є першою ефективною нейромережею, яка могла обробляти та генерувати аудіо-дані. Архітектура WaveNet дозволяла перетворювати аудіо-дані без попереднього налаштування фільтрів як для класичного вокодера, що дозволяло нейромережі самій формувати необхідні перетворення та налаштування для внутрішніх симульованих фільтрів. WaveNet використовував згорткові шари, які дозволяли нейромережі обробляти великі об’єми даних, аудіо-дані які обробляли перші моделі WaveNet мали дискретизацію у 16000 семплів на секунду, що робило використання стандартних практик нейромереж для обробки таких великих даних неефективними як і у витратах часу так і у кінцевій якості передбачень. Розглянемо більш детальніше архітектуру WaveNet, яка зображена на рисунку 2.1, так головна відмінність від класичних згорткових

генеративних мереж, це те що ця мережа має лише згорткові шари, але не має шарів які підвищують роздільність вихідних даних, але при цьому кількість вхідних нейронів дорівнює кількості вихідних нейронів. Досягнути рівної кількості нейронів входу і виходу без підвищення роздільності, стало можливим завдяки використання підходу, розширеного підключення нейронів, тобто нейрони мають лише вертикальне та діагональне підключення між собою, це дає можливість враховувати дані минулих семплів аудіо, при цьому не витрачати ресурси на неефективне підключення усіх нейронів між собою, як у класичних нейромережах.

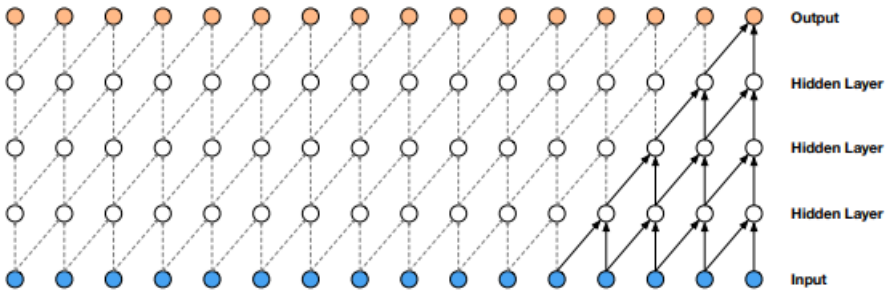


Рисунок 2.1 – Архітектура нейромережі WaveNet

Такий підхід дав і підвищення ефективності витрат часу на обрахування та тренування моделі та дав якісне підвищення результату передбачень. Так як WaveNet це модель, яка дозволяє загально працювати зі звуком, то з допомогою цієї моделі аналізували аудіо-дані, генерували прості музичні композиції і використовували як вокодер, ще під час розробки моделі. Використання WaveNet у якості вокодера [8] у пов’язані з акустичною моделлю з статистичного методу синтезу даних, дало підвищення якості просодії мовлення, краще ніж у конкатенаційному методу синтезу мовлення, що тоді вважалося найкращим за якістю методом, результати детальніше показані у рисунку 2.2. WaveNet зміг симулювати звуки, які раніше не міг синтезувати звичайний вокодер, такі звуки, як

наприклад рухи губ, клацання язика та вдихи та видих. Синтез цих звуків не було закладено заздалегідь у датасет для навчання вокодера, модель сама навчилася симулювати синтез цих звуків, які до цього вважалося неможливо з використання класичного вокодера і тим паче з високою якості просодії мовлення, яка дуже схожа на натуральне вимовляння людиною.

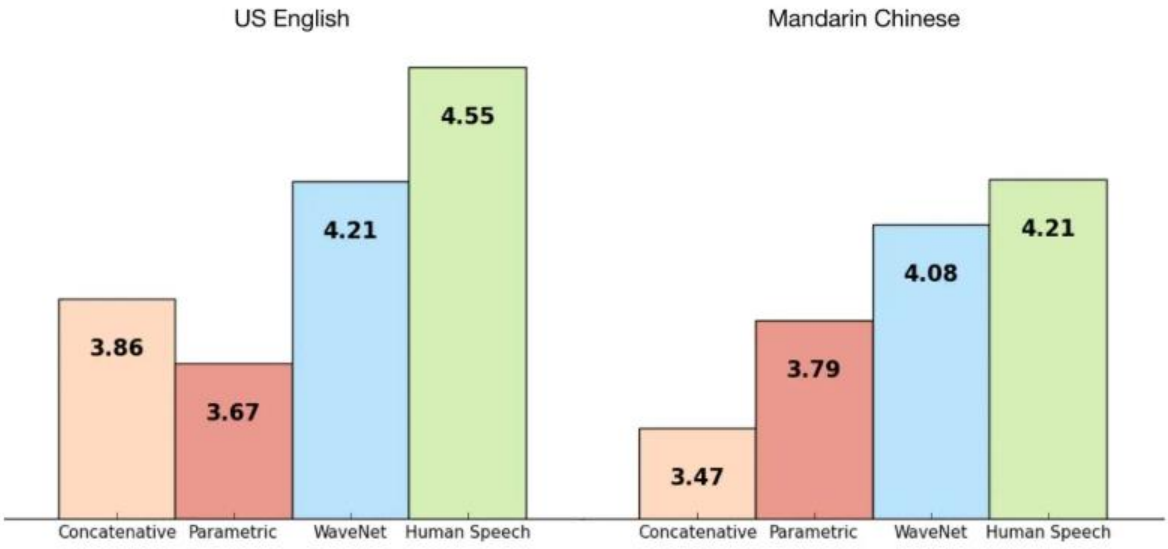


Рисунок 2.2 – Результати якості синтезованого мовлення використовуючи WaveNet у якості вокодера для статистичних методів синтезу мовлення

Подальші дослідження WaveNet показали результат, що модель здатна синтезувати відразу багато голосів, за допомогою модифікації моделі з додаванням додаткового параметру, який визначає тип голосу. Так навчання на добірці з різними просодіями голосу підвищували навіть якість генерації якогось окремо визначеного голосу, що було протилежним результатом на відмінну від традиційних методів синтезу мовлення. Для статистичної модель, навчання на добірці з різними голосами модель навчалася синтезувати мовлення звучанням середньої просодії між чоловічими та жіночими голосами, а WaveNet такий підхід тільки покращив

кінцевий результат. Це стало свідченням, того що модель може виділяти для себе саме закономірності формулювання артикуляційних особливостей формування просодії мовлення, а не просто наближати синтезоване мовлення до середнього значення даних з навчальної добірки даних.

Хоча WaveNet і мав проривний результат у сфері синтезу мовлення у сфері синтезу натурального звучання просодії мовлення, ця модель також мала свої недоліки, які ще необхідно було покращувати. Найголовніший недолік моделі, це те що вона є авторегресійною, цей фактор і зробив її ефективною у генеруванні мовлення, але через це генерування завдяки WaveNet є дуже повільним, генерування одного речення, могло зайняти часу до декількох десятків хвилин, що виключало можливість використання для синтезу мовлення відразу, як наприклад озвучування відповіді від чат-боту. Хоча і було відзначено підвищення якості синтезованого мовлення, у порівняні з традиційними методами, усе ще було відчутно легку роботизованість мовлення у коартикуляційних вимовляннях, вимовлені протяжних фонем та у переходах між словами. Також просодія мовлення все ще має легке характерне вокодерне звучання, але воно менше відчутне вуху, що є значущим результатом враховуючи комплексність мовлення та високі вимоги людського розпізнання мови. Наступні моделі вокодерів намагалися вирішити проблему високого часу генерації через авторегресію.

2.3.2 HiFi-GAN

Найефективнішим не авторегресійним методом для вокодера, є використання генеративної-змагальної архітектури. Однією з найкращих моделей, на цій архітектурі, є HiFiGAN [9]. Її суть у тому, що існує дві моделі, одна модель є генератором даних, а інша є дискримінатором, її задача відрізняти реальні дані від синтезованих. Необхідність такої архітектури виражається у тому, що цей підхід відноситься до синтезування даних, як спробу симулювати ймовірнісне простір, у якому знаходяться

дані. Отже у генераторі знаходиться представлення ймовірнісного простору, яке може змінюватися згідно з оцінкою дискримінатора, саме він дає оцінку позибки якості генерації. Дискримінатор також має свою функцію похибки, згідно з тим де знаходяться реальні та синтезовані дані. Таким чином модель генератора та дискримінатора вступають у змагальні відносини, де одна модель намагається перевершити іншу або у оцінці даних, або у їх підробці. Головний проблема виражається у знаходження оптимального підходу до архітектури генератора та дискримінатора, щоб одна модель не змога, значно перевершити іншу. Синтез даних виконує генератор.

HiFi-GAN використовував у генераторі паралельні згорткові шари, для аналізу мел-спектрограми та відновлення синусоїдної періодичності [9], для покращення якості звука для людського сприйняття. Як виявилося у результаті дослідження, саме завдяки фокусу генератора на обробці періодичності звука покращувалася якість висоти звука отримуваного з мел-спектрограми.

Для реалізації фокусу генератора саме на обробці періодичності звуку у мел-спектрограмі, у архітектуру було додано не один, а два дискримінатори, у яких різні задачі, повна архітектура взаємодії двух дискримінаторів з генератором показана на рисунку 2.3.

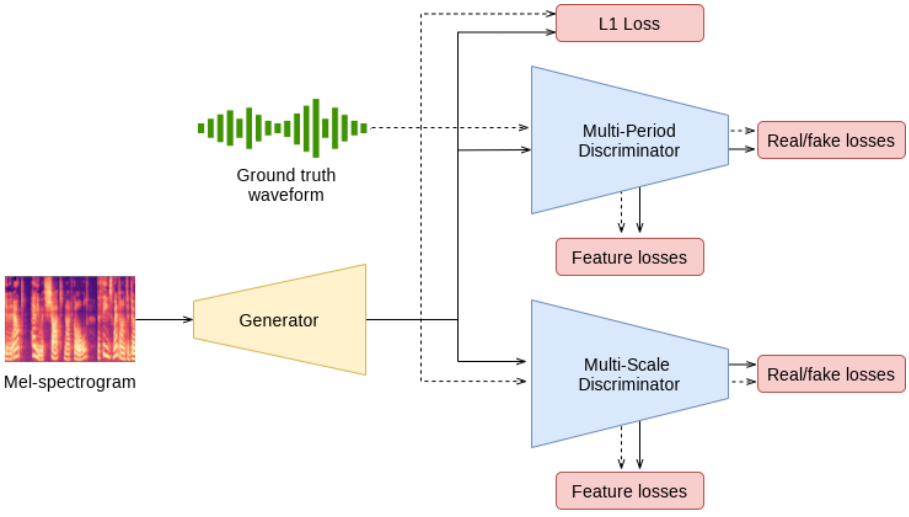


Рисунок 2.3 – Діаграма архітектури HiFi-GAN

Перший дискримінатор називається мульти-періодичним, його задача, це оцінка реальності синтезованих періодичностей у звуці. Другий мульти-скалярний дискримінатор, зайнятий оцінкою реальності сили та масштабів промовляння фонем у синтезованих даних. Усі дискримінатори є розбитим на деяку кількість під-дискримінаторів, для паралелізації задачі та пришвидшення аналізу даних при навчанні.

2.4 Моделі синтезу мовлення на основі нейромережевих технологій

З розвитком використання глибокого навчання для вокодерів, стало зрозуміло, те що є можливим створення систем, які повністю базуються на глибокому навчанні. Перші рішення були створенні на базі рекурентних нейронних мереж, вони могли навіть успішно генерувати мел-спектрограми, або акустичні параметри, як це робили параметричні методи. Рекурентний підхід мав великий недолік, що вони не можуть генерувати довільний результат для запиту, який більше ніж пару слів. Це пов’язано з обмеженістю латентного простору у середині мережі, через що мережа втрачала важливі дані при запитах, які були більш ніж пару слів. Рішенням цієї проблеми став перегляд підходу до синтезу мовлення, а саме бачення генерування мовлення, як задачі з декодування даних, де текст є дуже стислими даними, які необхідно перетворити у нестислі дані, якими є мовлення. Тому майже всі існуючі актуальні рішення з синтезу даних, будуть мати елементи енкодера та декодера, під час того, коли енкодер кодує дані, вони переходять у латентний абстрактний стан, де нейромережа зможе сама абстрагувати та керувати латентними признаками.

Лише зміна підходу не давала довільних результатів, бо залишалася та сама проблема як і з рекурентними мережами, обмеженість латентного стану. Цю проблему можливо вирішити лише з застосуванням механізму уваги. Модель з увагою зможе аналізувати латентний простір для кожного

окремого токену тексту і приймати рішення, яку саме фонему синтезувати та з якою коартикуляцією та просодією.

2.4.1 Моделі Tacotron

Модель Tacotron [10] була першою моделлю синтезу мовлення, яка повністю базувалася на методах глибокого навчання. Tacotron є базою для всіх інтелектуальних акустичних моделей, оскільки саме він зміг вирішити проблему синтезу саме мовлення, а не просто вокодування вже існуючої частоти. Саме Tacotron першим почав використовувати підхід до задачі синтезу мовлення як до задачі декодування, це досягалося завдяки архітектурі seq2seq, а саме взаємодії декодер-енкодер. Архітектура seq2seq, також мала проблему обмеженості латентного стану, через саму суть цієї архітектури і саме для цієї архітектури і було вперше відкрито та застосовано механізм уваги. Механізму уваги [11], який було запропоноване ще у іншій сфері, а саме машинному перекладі, який також базувався на seq2seq, став рішенням проблеми латентного стану. Суть уваги у тому що модель не намагається за один раз обробити все у латентному стані, а бере багато латентних окремих станів і аналізує у ході навчання їхню важливість у залежності від контексту вже минулих латентних станів інших компонентів, за допомогою матриці уваги. Цей підхід дозволяв розширити обмеження з природи латентного стану і дав можливість переводити у абстрактний простір великі дані. Після додавання механізму уваги, стало можлива генерація мел-спектрограм, їхня якість була вище ніж у спектрограм які генерувалися статистичним традиційним методом, результати оцінки якості наведені у рисунку 2.4, що і стало ключовим моментом, коли традиційні методи перестали використовуватися саме для якісної генерації мовлення.

Table 2: 5-scale mean opinion score evaluation.

	mean opinion score
Tacotron	3.82 ± 0.085
Parametric	3.69 ± 0.109
Concatenative	4.09 ± 0.119

Рисунок 2.4 – Результати оцінки синтезованого мовлення різними методами

Хоча Tacotron і мав високу якість синтезування мовлення, але він теж мав вагомі недоліки, які вимагали рішення. Ці проблеми були через недосконаліть механізму уваги, оскільки тоді це була ще доволі нова техніка, яку використовували лише у вузькій сфері роботи з текстом. Першим недоліком, були вади синтезування мовлення, модель могла просто пропускати або повторювати слова, вже тоді було зрозуміло, що це є проблемою саме механізму уваги, бо увага була відповідальна за декодування тексту у аудіо, а отже увага бувало перескакувала з одного латентного стану на минулий у якому вже була багато разів і внаслідок цього виникали артефакти та збої у генерації. Другою вагомою проблемою стало те що модель також як і WaveNet була доволі сильно зав’язана на механізмі авторегресії через що також мала малу швидкість генерації мел-спектрограм.

Рішенням цих проблем стала наступна ітерація Tacotron, а саме Tacotron 2 [12]. Її головна особливість була у використанні іншого механізму уваги, архітектуру використання уваги зображено на рисунку 2.5, а саме локаційна-чутлива увага, вже з назви зрозуміло, що цей механізм уваги також враховував не лише латентні дані та їх положення відносно авторегресійної ітерації генерації мовлення, а також і враховував і позицію самого латентного стану у моделі.

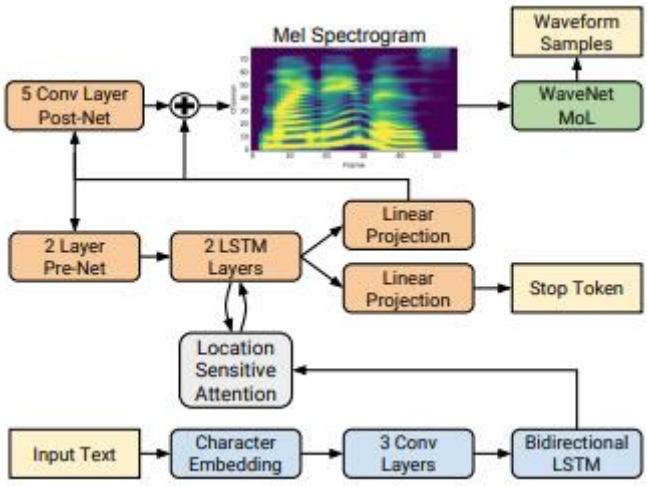


Рисунок 2.5 – Архітектура використання механізму локаційно-чутливої уваги у Tacotron 2

Так було вирішено проблему галюцинацій моделі, повторення або пропуск слів, що були у тексті, але відсутні у аудіо-файлі, бо механізм уваги більше не блукав колами, а мав знання про своє положення. Також разом з цим і підвищилася загальна якість синтезованого мовлення, так ця модель вже обходила, тодішню найкращий метод, конкатенаційний, результат оцінки синтезованого мовлення надано у рисунку 2.6.

System	MOS
Parametric	3.492 ± 0.096
Tacotron (Griffin-Lim)	4.001 ± 0.087
Concatenative	4.166 ± 0.091
WaveNet (Linguistic)	4.341 ± 0.051
Ground truth	4.582 ± 0.053
Tacotron 2 (this paper)	$\mathbf{4.526 \pm 0.066}$

Рисунок 2.6 – Таблиця оцінки мовлення різних моделей

Також це був один з перших випадків, коли люди не могли відрізнити синтезоване мовлення від оригінального, як це продемонстровано на

рисунку 2.7, у вигляді графіку розкиду оцінок якості синтезованого мовлення.

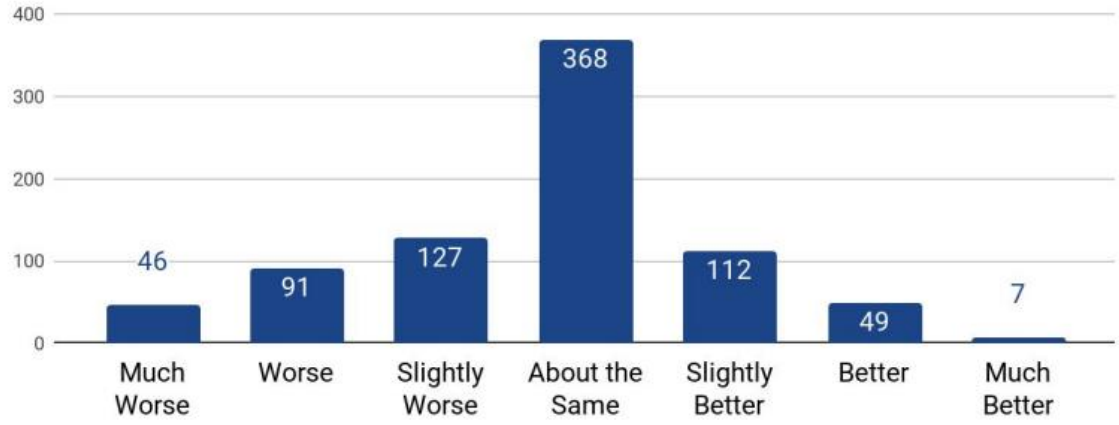


Рисунок 2.7 – Графік розкиду оцінок синтезованого мовлення Tacotron 2

Маючи високі показники якості синтезованого мовлення, Tacotron 2 все ще був зав’язаний на механізмі авторегресії. Генерування синтезованого мовлення мало квадратичну залежність через авторегресію, тому генерування також як і першій моделі займало доволі багато часу.

Усі наступні моделі, після успіху якості синтезованого мовлення Tacotron 2, почали спрямовувати сили на пришвидшення синтезування мовлення, без втрати якості синтезованих даних, для того щоб мати перевагу над Tacotron 2.

2.4.2 Моделі FastSpeech

Моделі FastSpeech, є першими успішними моделями, які змогли реалізувати синтезування мовлення, спочатку з частковим залученням авторегресії, а потім зовсім без використання механізму авторегресії, і маючи при цьому якість яка була вища ніж у Tacotron 2.

FastSpeech [13] використовувала такий же принцип, як і Tacotron, а саме використання енкодеру та декодеру, але ці компоненти були реалізовані використовуючи Multi-head Attention [14], що є стабільною та передбачуваною імплементацією уваги, це виключало можливість галюцинацій або повторення слів, як було у моделях Tacotron. Також архітектура FastSpeech, для свого навчання використовувала іншу нейромережу, яка мала авторегресійну природу і з неї брала дані сходження токенів та довжини звучання фонем та їхньої просодії і навчалася на цих даних. Таким чином FastSpeech навчався вже готовим закономірностям які на той час було можливо вивести лише завдяки авторегресійним моделям, цей процес показано на рисунку 2.8, який описує архітектуру FastSpeech.

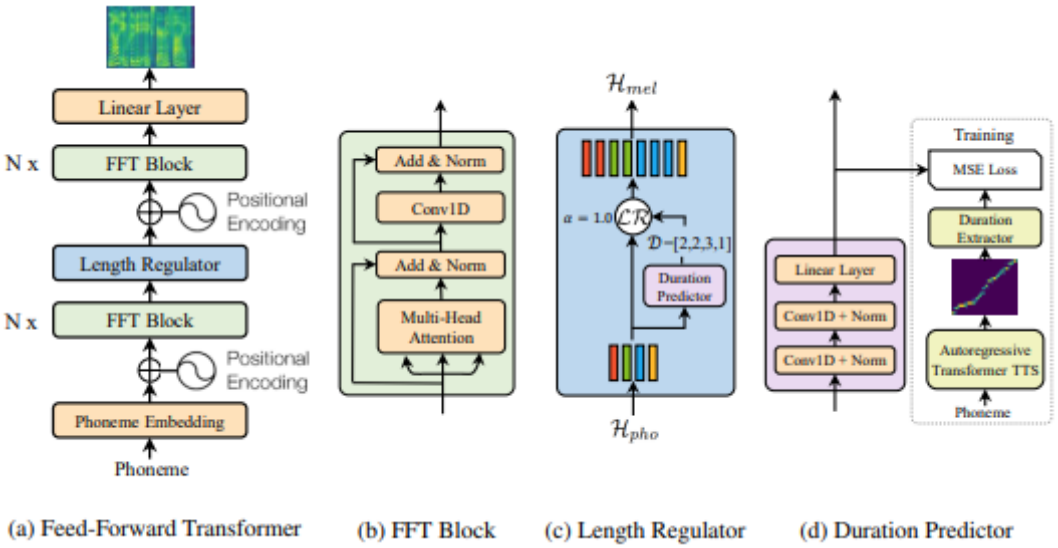


Рисунок 2.8 – Архітектура FastSpeech

Завдяки такому процесу тренування вдалося обійти проблему авторегресії під час синтезу. Синтез FastSpeech потребував лише одного проходження через мережу, що робило процес генерації мовлення лінійно залежним від тексту, без втрати якості, оскільки фонемні закономірності були вивчені авторегресійними моделями заздалегідь, а FastSpeech завдяки Multi-head Attention зміг їх узагальнити у лінійну залежність. Завдяки

паралелізації синтезуванню швидкість синтезування мовлення підвищилася у 260 разів, що зробило генерацію майже моментальною. Але FastSpeech мав ваду через залежність від авторегресії, як і у швидкості тренування так і у тому, що якщо авторегресійна мала вади галюцинацій, то і FastSpeech також навчиться галюцинаціям.

Наступна ітерація FastSpeech, а саме FastSpeech 2 [15], вирішувала проблему залежності навчання від вже навчених авторегресійних моделей. Рішенням стала відмова від залежності які були вивчені авторегресійними моделями та створення власних блоків, для того, щоб модель могла інтерпретувати латентні дані мовлення. Цим блоком став Variance adaptor. Variance adaptor складається з декількох предикторів, це продемонстровано у рисунку 2.9.

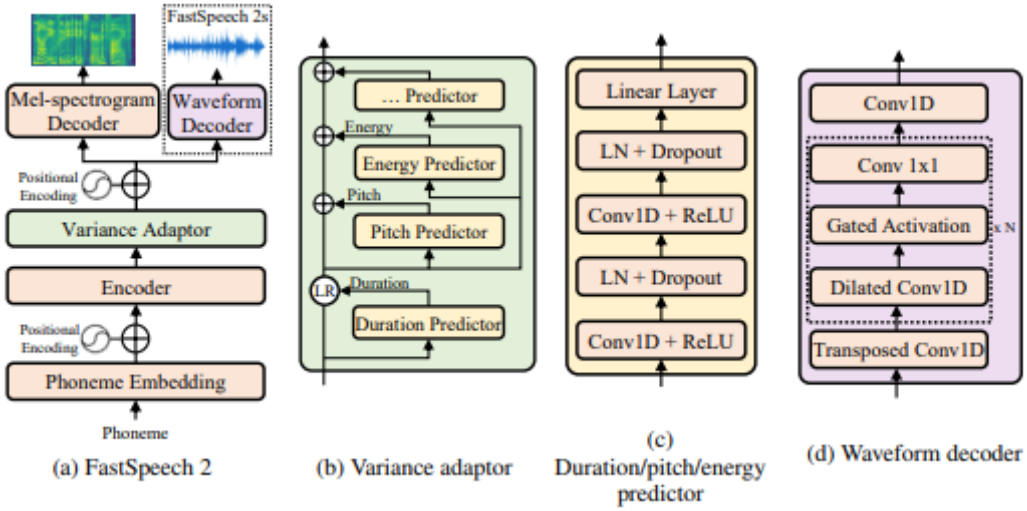


Рисунок 2.9 – Архітектура FastSpeech 2

У оригінальній праці предикторів 3, це Duration predictor, який відповідає за довжину промовляння фонеми у спектрограмі, pitch predictor, за передбачення мел значень у спектрограмі, energy predictor, який відповідальний за енергійність синтезованого мовлення. Також можливо додавання інших предикторів, у дослідженні [15] вказано, що при додаванні додаткових предикторів, знижувався час навчання та підвищувалася якість

та додавалися додаткові функції контролю синтезу, але FastSpeech 2 зупинився на 3 предикторах, як найефективніше рішення.

Так FastSpeech 2 зміг досягнути високої якості синтезу, що перевершувала якість Tacotron, швидкості синтезу та модульності та контролю системи, що давало простір для модифікацій. Завдяки цим факторам FastSpeech 2 вважається класичною архітектурою, рішення якої використовуються і у сучасних моделях, а саме використання у енкодерах та декодерах Multi-head Attention, через їх ефективність і стабільність. Проте FastSpeech 2 мав недолік попередньої обробки даних для кожного нового предіктору та ще присутні недоліки у просодії мовлення.

Досвід Tacotron та FastSpeech показали, що задачу синтезу мовлення можливо вирішити швидко та якісно завдяки використанню лише методів машинного навчання. Але ці моделі мали і свої істотні недоліки, такі як великі затрати по часу генерації, для авторегресійного підходу, або додаткової обробки даних чи залежність від інших готових рішень з синтезу мовлення, як це було з FastSpeech. Існує якісне рішення для генерації даних, яке можливо використовувати і для синтезу мовлення, оскільки оригінальна її ціль була генерація зображень, а мовлення також можливо представити, як картинку мел-спектрограми. Цим рішенням є використання дифузії для синтезу мовлення. Сучасні підходи до дифузії дозволяють відмовитися від авторегресії, додавання необхідних додаткових даних до датасету та, що головне спрямовані на підвищення швидкості генерації зі збереженням високої якості синтезованих даних.

3 ВИКОРИСТАННЯ ДИФУЗІЇ ДЛЯ СИНТЕЗУ МОВЛЕННЯ

3.1 Механізм генерації завдяки дифузії

Дифузія спершу була запропонована у 2015 році [16], як можливий механізм генерації даних, але практичний механізм її функціонування та навчання був розроблений лише у 2020 році, як ефективний метод для генерації зображень [17]. Хоча метод дифузії і був запропонований як рішення для генерації зображень, цей метод також підходе для синтезу мовлення, бо модель завжди генерувала мел-спектрограму, а мел-спектрограму можливо представити як одно канальне зображення, де один канал, це мел-значення.

Сам механізм дифузії представляє собою навчання нейромережі для покрокового очищення випадкової послідовності гаусівського шуму у необхідне зображення, яке була запитано через текстові токени. Процес навчання дифузійної моделі представляє зворотній від генерації процес. Для навчання модель поступово додає шуму до оригінального зображення поки воно не досягне близьких значень випадковості шуму з гаусівським розподілом ймовірностей. Функцію втрати для тренування береться з квадратної різниці між отриманим зашумленням від нейромережі на деякому кроці зашумлення та зашумленною версією зображення через гаусівський розподіл на цьому ж кроці.

Необхідність створення шумів на тренувальному зображені для генерації деякого нового та унікального зображення полягає у тому, що через процес додавання шумів модель вчиться оперувати у деякому багатовимірному ймовірнісному просторі, який репрезентує положення зображень на яких вчиться модель. Навчаючись на різниця прогнозованого додавання шуму та реального гаусівського шуму модель починає формувати латентний простір, який описує, як можливо рухатися у деякому ймовірнісному просторі для досягнення найбільшої ймовірності, тобто

генерування деякого зображення, без повного знання того який має вид реальний ймовірнісний простір зображень. Створення при генерації зображення випадкового гаусівського шуму та процес прибирання шуму, є процесом пересувань зображення яке необхідно згенерувати у ймовірнісному просторі зображень, від випадкового місця, де через гаусівський шум, вірогідність зображення завжди буде 0, до необхідного зображення, яке у ймовірнісному просторі буде мати високу ймовірність. Під час створення зображення з шуму, шум повинен пройти велику кількість кроків з прибирання шуму, оскільки при проходженні лише одного кроку, та збільшенні цього кроку, для переміщення у ймовірнісному просторі, вийде ефект того, що генероване зображення може або не дійти або перейти далі глобального ймовірнісного максимума, або взагалі мати не той напрям який необхідний, через вплив інших ймовірнісних максимумів. Також з використанням великої кількості кроків з прибирання шуму, точка генерованого зображення у ймовірнісному просторі, може обходити локальні ймовірнісні максимуми та трохи відхилятися від екстремума глобального максимума ймовірностей і таким чином трохи змінювати дані на генерованому зображені у відмінності від тих на яких був навчений, що дає ефект креативності генерації.

У класичної дифузії є недолік, через який її використання для синтезу мовлення було непрактичним у порівнянні з іншими, а саме час генерації мел-спектрограм. Так як для синтез даних першочерговим даним згенерованим гаусівським розподілом необхідно було пройти від сотні кроків, а іноді і тисячі, що підвищувало час генерації спектрограм. Також час підвищувався у залежності з розміром тексту, через процес конкатенації різних мел-спектрограм для окремих блоків тексту, що також потребувало повторної дифузії. Рішенням проблеми довгої генерації мовлення є використання більш просунутих технік для переміщення та представлення ймовірнісного простору мел-спектрограм, у цій праці це буде використання технології flow matching.

3.2 Conditional flow matching для швидкої генерації даних

Головною проблемою при пришвидшенні генерації даних дифузії, як було вказано, є те що точка генерованого зображення не в змозі знайти таким чином глобальний ймовірнісний максимум для свого запиту. Flow matching [18] пропонує рішення, яке полягає у переосмислені процесу переходу простору ймовірностей гаусівського шуму у простір ймовірностей необхідних даних завдяки векторним полям. Векторні поля будуть спрямовувати семпл з простору ймовірностей гаусівського шуму у простір ймовірностей необхідних даних. З введенням поля швидкості, також буде введена змінна часу, яка необхідна для дискретного рішення задачі векторного поля швидкості. Процес вивчення векторних полів виглядає таким чином, що береться випадковий семпл з параметром часу, який описує його положення на шляху від гаусівського шуму до даних, для цього семплу мережа генерує передбачення швидкості переходу до даних і обраховує свою помилку згідно з реальною швидкістю переходу між просторами. Проводячи навчання на багатьох семплах, модель вчиться симулювати векторне поле швидкості, яке змінюється, разом зі змінною часу, це необхідно для дискретного обчислення шляху даних по векторному полю. Окрім зміни скалярного значення векторів, зі змінною параметру часу, також, , векторне поле має властивість змінювати і напрями векторів у полі, внаслідок навчання на багатьох семплах, цю властивість називають маргінальною. Саме завдяки вивченим маргінальним полям і можливий синтез необхідних даних без повного рішення векторного поля швидкостей, а використовуючи лише передбачення нейромережі.

Завдяки маргінальним векторним полям та дискретизації проходження по векторному полю швидкості, тепер для генерації якісних даних необхідно до 10 кроків, замість сотень кроків, які потребувала оригінальна імплементація дифузії.

Єдина проблема яку мав Flow matching, це була проблема, якщо дані були дуже різноманітними, до зростала кількість необхідних кроків проходження векторного поля, для стабільної генерації, через викривлення маргінального векторного поля. Рішення для цієї проблеми запропонували у роботі про Rectified flow [19]. У праці Rectified flow, було досліджено аналогічний процес з Flow matching, але з додатком того, що було додано фінальний крок до навчання, reflow. Reflow – це процес навчання моделі векторним полям, але з врахуванням маргінального векторного простору, який вже може симулювати модель. Завдяки цій техніці, модель навчається випрямляти вектори швидкості, завдяки чому симулювання проходження по векторному полю може займати менше кроків.

Flow matching, наразі є ведучою технологією для генерації даних, бо вона має дуже високу швидкість генерації даних та велику якість генерованих даних.

3.3 Flow Matching для синтезу мовлення

Як вже було зазначено, методи дифузії та Flow Matching, спрямовані на швидку та якісну генерацію даних з деякого простору ймовірностей, що є ключовим вирішенням проблем синтезу мовлення, для швидкості генерації та високої якості мовлення. Більшість досліджень та відомих проектів, які використовують Flow Matching спрямовані на генерацію зображень, завдяки вводу текстових даних. Вже зазначалося, що всі моделі на базі глибокого навчання генерують саме мел-спектрограму для подальшого синтезу мовлення. Мел-спектрограма представляє з себе те ж саме зображення, але лише з одним каналом, мел-значеннями, а не з 3 каналами, як це у стандартних зображеннях. Таким чином можливо використовувати дифузію для генерування мел-спектрограм і подальшому використовувати їх для синтезу мовлення.

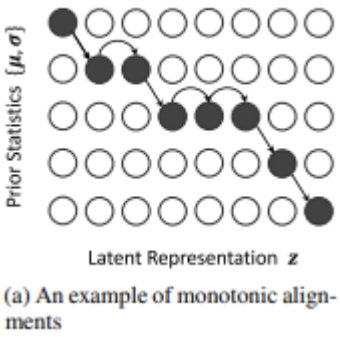
Ключовими проблемами у такому підході, є навігація у ймовірнісному просторі, згідно з текстових даних, та проблема розмірності мел-спектрограми. Класична імплементація Flow Matching, не дасть можливості якісно генерувати мел-спектрограми завдяки текстовим даним, оскільки модель під час синтезу буде втрачати деталі просодії та мати велику кількість незначних, але відчутних на слух, різницю у довжині промовляння фонем. І як раз з цієї різниці промовляння фонем та різної довжини тексту буде виникати друга проблема, з розміром генерованої мел-спектрограми, бо чим довше вона буде, тим гірше загальна якість буде. Наївним рішенням проблеми розмірності буде навчання моделі на Flow Matching на більш довгих відрізках мел-спектрограм, що є неефективним рішенням.

Тому для ефективної генерації на Flow Matching, буде використано інший підхід до синтезу, а саме синтез на основі вже заздалегідь передбаченої усередненої мел-спектрограми. Таким чином Flow Matching не повністю з шуму та тексту генерує, а лише доповнює усереднену мел-спектрограму деталями. Так вирішуються раніше описані проблеми для синтезу мовлення з використанням Flow Matching. Відразу вирішується проблема розмірності мел-спектрограми, оскільки цю задачу на себе бере усереднена мел-спектрограма, де вже повинні бути визначенні довжина звучання фонеми, як вони зв’язані між собою по інтонації, гучності, а Flow Matching буде лише покращувати якість просодії. Також використання усередненої мел-спектрограми дає можливість Flow Matching краще орієнтуватися, оскільки більшість частин кроків для генерації з гаусівського шуму була пройдена і Flow Matching сконцентрована завжди на покращенні якості просодії мовлення та загальної якості мел-спектрограми.

3.4 Генерування усередненої мел-спектрограми

Як вже було визначено, для ефективного синтезу мовлення з використанням Flow Matching, необхідно мати усереднену мел-

спектрограму. Для генерації усередненої мел-спектрограми було взяте за основу рішення запропоноване у Glow-TTS [20] з використанням трансформерів та алгоритму MAS. Задача моделі, яка буде генерувати усереднену мел-спектрограму, це симулювання роботи MAS у створенні залежності, яка довжина фрейму повинна бути для кожної окремої фонеми, з врахуванням минулих фонем, та передбачення усереднених значень мел-спектрограми для кожної фонеми з латентних даних трансформерів та передбачених значень MAS. Симулювання роботи алгоритму MAS та виведення цього процесу у нейронні мережі, пов’язано з доволі великим часом виконання цього алгоритму. Сам алгоритм MAS незважаючи на великі часові затрати для свого виконання, використовується, бо він гарантує свою монотонність у передбачені довжини фонеми у фреймі мел-спектрограми, приклад результату MAS, показаний у рисунку 3.1.



(a) An example of monotonic alignments

Рисунок 3.1 – Результат роботи MAS у вигляді монотонної послідовності для фонем у відношенні фреймів

Це ключовий пункт, оскільки він гарантує відсутність проблеми галюцинацій, як у Tacotron, і не вимагає додаткової обробки даних.

Рішення про використання цієї імплементації моделі для генерування усередненої мел-спектрограми прийнято, бо Glow-TTS, виконує схожу задачу, яка необхідна для функціонування Flow Matching, лише з тією відмінністю, що у Glow-TTS використовується лише класична дифузія.

3.5 Вокодер iSTFTNet

Маючи рішення про генерацію мел-спектрограм, залишається необхідність у перетворенні спектрограми у аудіо-дані. HiFi-GAN наразі є найпопулярнішим рішенням для вокодерів, завдяки фокусу генератора на періодичності звучання, але існують і більш якісніші рішення на його основі. iSTFTNet [21] – є вокодером, який працює на основі передбачень необхідних параметрів для виконання інверсійного перетворення Фур'є, використовуючи механізми навчання генеративної-змагальних мереж від HiFiGAN, для збереження фокусу навчання на періодичності висоти звуку. Для виконання інверсійного перетворення Фур'є та отримання аудіо-даних, окрім спектрограми також потребує параметрів розміра вікна, фрейму та інших параметрів.

Ключова відмінність iSTFTNet від HiFi-GAN формується у генераторі. Оскільки для генератора iSTFTNet необхідно менше послідовних згорткових шарів, через спрощення самої задачі генерації, через підхід до використання інверсійного алгоритму перетворення Фур'є, більш детально архітектура генератора показана на рисунку 3.2.

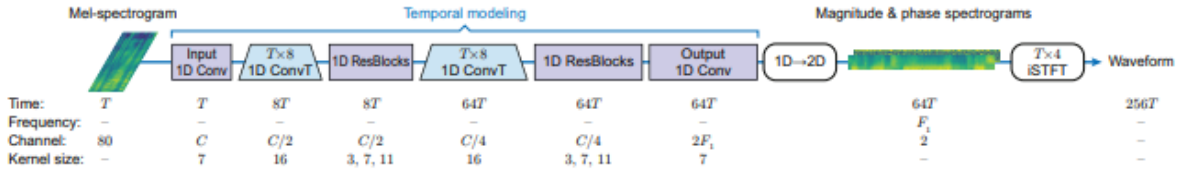


Рисунок 3.2 – Архітектура генератора iSTFTNet

Так послідовні згорткові шари навчаються симулювати спектральні маски, щоб мати змогу дізнатися параметри для інверсійного перетворення Фур'є та взяти ці параметри з мел-спектрограми. Згідно статистики, що представлена на рисунку 3.3, цей метод є більш швидким ніж перетворення

вокодера даних відразу у звук, завдяки використанню лише згорткових шарів у генераторі.

Model	MOS↑	cFW2VD↓	RTF↓	# Param↓
Ground truth	4.71 ±0.07	–	–	–
HiFi-GAN V2	4.20 ±0.10	0.046	0.053 (100)	0.93M (100)
iSTFTNet-C8C8I4	4.12 ±0.10	0.042	0.029 (55)	0.89M (96)
iSTFTNet-C8C1I32	3.71 ±0.13	0.071	0.018 (34)	1.30M (140)

Рисунок 3.3 – Таблиця якості мовлення синтезованого вокодерами

Завдяки більш простій та ефективнішій архітектурі, вокодер на базі iSTFTNet, буде використовуватися для синтезу мовлення у цій праці.

3.6 Запропонована архітектура синтезу мовлення

Маючі запропоновані рішення для покращення швидкості та якості синтезу мовлення у вигляді використання Flow Matching, усереднених мел-спектрограм та використання вокодера iSTFTNet, перейдемо до створення архітектури взаємодії цих компонентів для функціонування системи синтезу мовлення.

Система буде мати класичну модель взаємодії «Джерело-фільтр», де модель буде генерувати мел-спектрограму і передавати її у вокодер для синтезу аудіо-даних. У якості вокодера буде використано iSTFTNet, але буде можливість змінювати вокодери на інші моделі, якщо вони були попередньо створені та навчені.

У якості моделі для синтезу мел-спектрограм буде використано архітектуру взаємодії двох моделей енкодера та декодера, як це було представлено у минулих проаналізованих рішення таких як Tacotron та FastSpeech. Енкодер та декодер будуть представляти собою такі елементи, як для енкодера це генерування усередненої мел-спектрограми з текстових

даних, а для декодера, це модель симулювання Flow Matching, яка буде покращувати якість усередненої мел-спектрограми та віддавати синтезовану мел-спектрограму.

Перед передачею тексту до енкодера текст буде перетворюватися у фонетичну транскрипцію згідно міжнародного фонетичного алфавіту, щоб надати можливість моделі оперувати будь-якою існуючою мовою світу для синтезу мовлення. Уся запропонована архітектура продемонстрована на загальній діаграмі на рисунку 3.4.

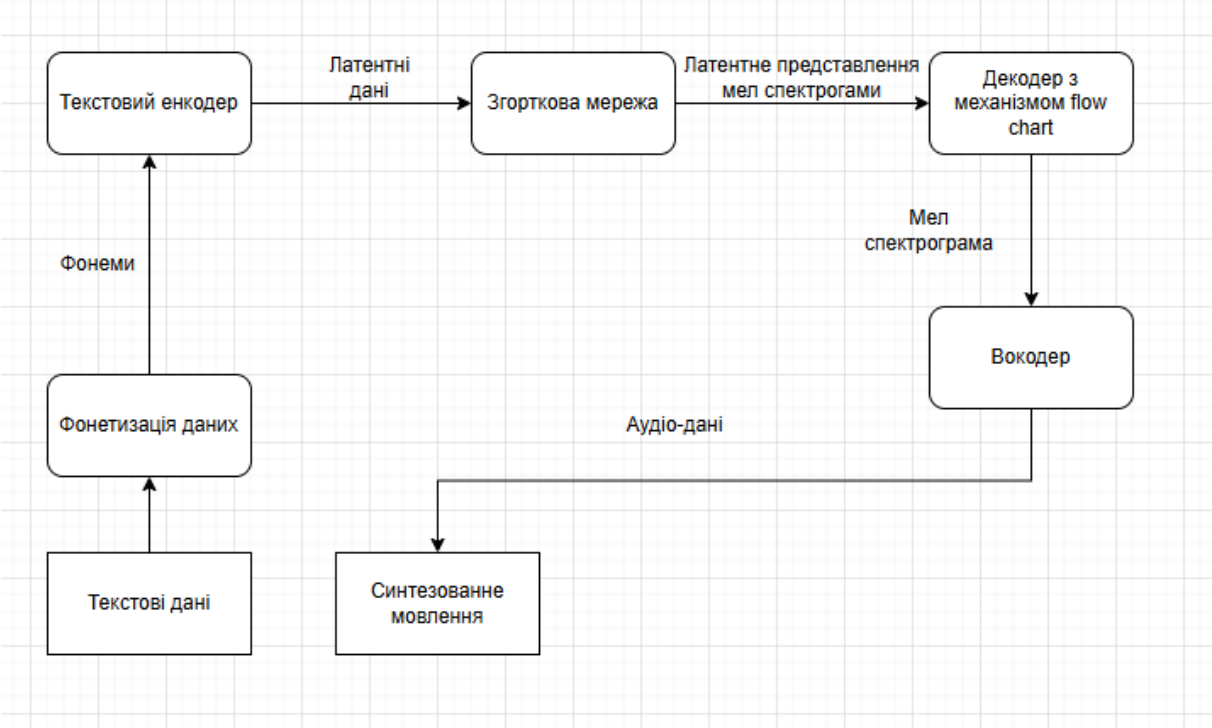


Рисунок 3.4 – Діаграма архітектури моделі синтезу мовлення з використанням технології Flow Matching

Маючи запропоновану архітектуру, яка зможе синтезувати мовлення використовуючи технологію Flow Matching, далі перейдемо до програмної реалізації архітектури, її навчанню та аналізу результатів навчання та синтезу мовлення.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Використанні модулі та бібліотеки

Для реалізації інтелектуальної системи синтезу мовлення, необхідно створювати доволі великий комплекс різних рішень, як і для задач глибокого навчання, так і для різних оптимізаційних та програмних рішень для полегшення коду. У цьому підрозділі буде наведений перечень бібліотек, які використовуються у проекті, та причини їхнього використання.

Головною бібліотекою для реалізації функціонування нейронних мереж та інших методів глибокого навчання є PyTorch та його сумісні бібліотеки, такі як torchvision, для роботи з зображеннями, у випадку цієї роботи, обробка мел-спектрограм та torchmetrics, для метрик нейромережей, їхньої швидкості синтезу, навчання. PyTorch, через свій інтерфейс взаємодії, вимагає описування процесу, взаємодії між собою вже розроблених модулів, на цих модулях буде будуватися велика частина системи. Для вирішення цієї вимоги, буде використовуватися бібліотека Lightning, яка сама динамічно аналізує взаємодію між собою користувацьких компонентів PyTorch та дозволяє навчати модель, без необхідності розписувати великі ланцюги взаємодії модулів.

Для аналізу результатів, були використані бібліотеки для наукового аналізу SciPy та бібліотеки для створення візуалізацій та графіків matplotlib та seaborn. Для взаємодії з аудіо-даними та перетворенню звуку у мел-спектрограми було використано бібліотеки librosa та torchaudio. Для роботи з таблицями даних, а саме мел-спектрограм, було використано pandas.

Також для пришвидшення виконання тяжкого та часто використовуваного алгоритму alignment search, було використано рішення для компіляції python-подібного коду, Cython.

Для вирішення задачі перетворення тексту у фонеми, було використано бібліотеку phonemize, який представляє собою програмний прошарок, для взаємодією з однієї з загальних програм для фонемізації. Для коректного функціонування фонемізації для багатьох мов, було використано espeak-ng.

4.2 Реалізація взаємодії з моделлю

Взаємодія з моделлю буде реалізована, за допомогою взаємодії з CLI. Таке рішення було прийнято для того, щоб з моделлю було простіше взаємодіяти, як і користувачу, так і розробникам, які можуть використовувати цю модель у подальшому. Для користувачів розроблений прості команди, які включають у себе або введення шляху до файлу з текстом, або його введення напряму у консольну строку. Через такий підхід, можлива взаємодія для розробників, що можуть взаємодіяти з цією моделлю, так для розробника не буде необхідності додавати багато інших модулів у свій проект, компілювати окремі файли оптимізації, та писати код який необхідно у майбутньому підтримувати, через можливий конфлікт між модулями у проекті. Взаємодія для розробника зменшується до встановлення моделі, завантаження чекпоінту або тренування з нуля і подальша взаємодія, через виклики у завдяки API командної строчці у програмі.

4.3 Обробка тексту та перетворення у фонеми

Перетворення у фонеми, завдяки використанню espeak-ng та phonemize, дозволить дати можливість працювати з багатьма мовами моделі, без втрати якості, через фактор іншої мови. Такий підхід дасть можливість навчити модель, не лише на англійському великому датасеті,

для перевірки загальної якості моделі, а також дасть можливість навчити модель синтезувати мовлення також українською мовою.

Попередня обробка тексту, буде включати у себе, як було вже обговорено, функціонал, перетворення деяких уніфікованих скорочень тексту, та перетворення символів, таких як символ «$» чи «₴» чи «@» у слова такі, як «долар», «гривня» та «равлик».

Також було реалізовано функціонал перетворення чисельних значень у текстові дані, як для «2000» перетворюється у «дві тисячі», це працює для більшості чисельних значень, які лежать у множині реальних чисел. Також перетворення дробних десяткових чисел, не будуть перетворюватися, у такому форматі, як «3.1415» у «три цілих чотирнадцять тисячних та п’ятнадцять сотих», а перетвориться як «три цілих та тисяча чотириста п’ятнадцять», це буде зумовлено через обмеженість можливостей бібліотеки про перетворення даних.

4.4 Реалізація запропонованої архітектури

Розглянувши необхідні загальні рішення та попередню обробку даних опишемо, як було реалізовано архітектуру синтезу мовлення. Загальна модель ділиться на 3 великі компонента. Перші два це енкодер та декодер, які безпосередньо відповідають за синтез мел-спектрограми та третій компонент вокодер, для отримання якісного аудіо-файлу. Енкодер та декодер буде компоновано у окремий клас «TTS-MelGen», який власне буде відповідати за генерацію мел-спектрограм, а вокодер буде окремим класом. Це зумовлено тим, що «TTS-MelGen», є ядром моделі, який робить синтез, а вокодер можливо доволі просто змінювати, якщо це необхідно, тому було прийнято такий розподіл.

Синтез між енкодером та декодером буде відбуватися, завдяки передачі один одному проміжних даних у загальній функції forward або synthesise. У цій функції, також окрім передачі, буде формуватися

відношення між даними тексту та вихідними даними енкодера, для можливості conditional генерації аудіо, що є базовою необхідною функцією.

Маючи знання про те як будуть взаємодіяти між собою головні компоненти моделі, перейдемо до практичної реалізації розглянутих компонентів, та почнемо з енкодера, як першого компонента у ланцюгу взаємодій

4.4.1 Реалізація енкодера та ембедінга з текстом

Енкодер, приймає у себе дані про перетворений текст у фонеми та перетворює їх у загальне передбачення мел-спектрограми та передбачає довжину промовляння кожної окремої фонеми.

Програмна реалізація енкодера здебільшого була спадкована з проекту синтезу мовлення GlowTTS [20], оскільки вона вирішує аналогічну проблему з кодуванням даних та створенням узагальненого передбачення мел-спектрограми для подальшого декодування у синтезовану мел-спектрограму. Хоча перша чергове GlowTTS вирішувала проблему за допомогою менш швидкого методу дифузії, але її рішення у кодуванні працюють і для flow matching.

Енкодер буде створювати латентні дані репрезентації фонемних даних, які називаються проекцією та передбачити довжину звучання фонем у мел-спектрограмі. Потім ці значення між собою перемножуються у компоненті ембедінгу тексту, завдяки механізму уваги, як раз у цьому компоненті у створиться узагальнена мел-спектрограма.

Розберемо більш детально механізм енкодера, та які рішення були використані. Енкодер буде будуватися на використанні згорткових шарів та механізмів уваги, які будуть реалізовані, як Multi-head Attention. Механізми уваги та згортки потребують репрезентації, як дані у 2-х вимірах, де одна вісь, це дані про частоту, а інша це часові дані, для кращої просодії, через часову залежність звучання фонем та їхню позицію у мовленні.

Використання 2-х вимірів збільшить розмір моделі у квадратній залежності, а з тим і час навчання і синтезу. Тому для збереження необхідних проміжних даних та більшої ефективності у енкодері буде використано RoPE. RoPE представляє собою особливе представлення латентних даних, завдяки якому можливо ефективно виражати відношення між токенами, у випадку цієї роботи, фонемами, без використання додаткової вісі даних. Головна ідея представлення даних, це обертання даних, завдяки матрицям оберту, у деякій н-мірній площині, для цієї моделі, це 2-мірна площина. Ці матриці оберту, дають можливість моделі вивчити оберт векторів, як вираження даних про позиційний зв’язок між фонемами. Загальна аріхтектура енкодера зображена на рисунку 4.1.

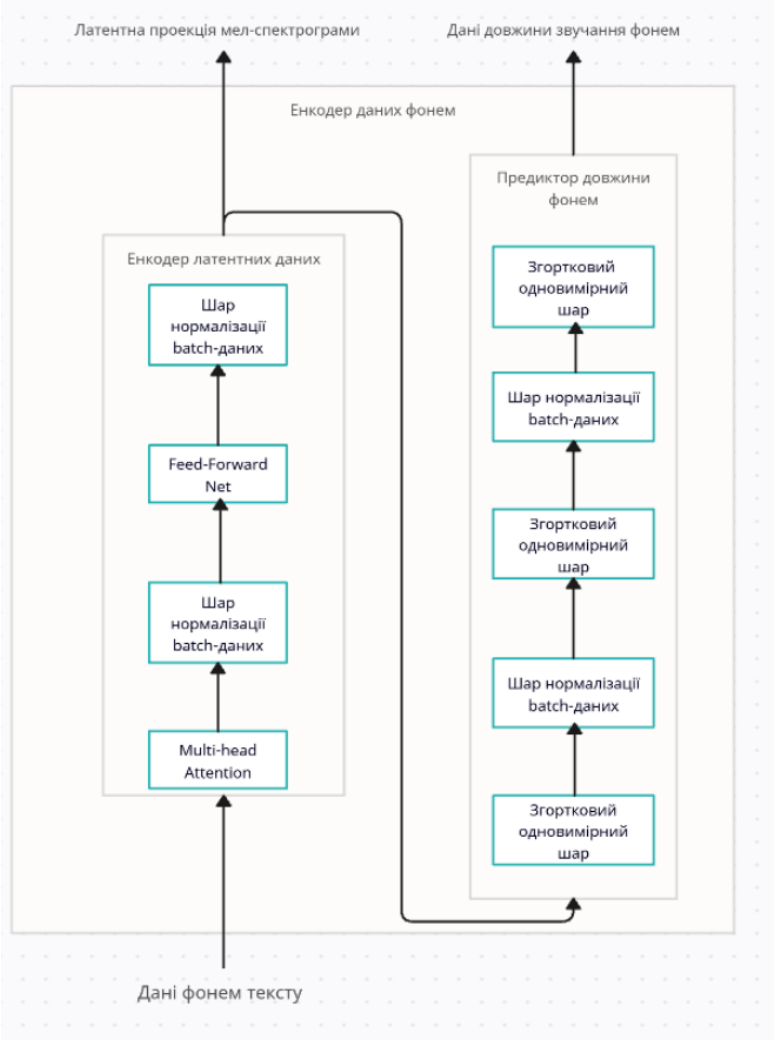


Рисунок 4.1 – Діаграма архітектури енкодера моделі

Було використано загальний підхід до кодування даних, який всебічно використовується як і у LLM так і у TTS-системах, а саме почергова обробка даних, спочатку у Attention блоках, а потім їх обробка у класичних FFN. Також було використано проріджения даних та вагів, між механізмами уваги та FFN. Предиктор довжини фонем, має просту архітектуру зі згортками, оскільки увага для нього не потрібна і тому він і був виділений у окремий компонент.

Так як обрахування обертів через матриці, потребує обрахування великої кількості функцій синусів та косинусів, було реалізовано функціонал кешування результатів синусів та косинусів у таблиці вже заздалегідь вирахуваних даних.

Просте навчання енкодера лише на сирих даних мел-спектрограм, не матиме задовільного результату та буде страждати від вже вказаних проблем Tacotron, такі як галюцинації та повторення фонем. Для виключення таких проблем і стала ще одна причина з якої було взято за основу енкодер Glow-TTS, бо вони вирішили цю проблему представив алгоритм monotonic aligment search. Monotonic aligment search це тяжкий алгоритм пошуку найоптимальнішого шляху у деякій матриці. З назви алгоритму зрозуміло, що його головна особливість, це монотонність створеного шляху, тобто він не може повернутися на якусь минулу фонему при генерації мел-спектрограми. Саме симуляції результату Monotonic aligment search і навчається предиктор довжини фонем. Через тяжкість обрахувань при тренуванні Monotonic aligment search, цей алгоритм був написаний на Cython.

Маючи дані з проекції та довжини звучання фонем, тепер необхідно перетворити їх у узагальнену мел-спектрограму. Це відбувається завдяки перемноженню матриці проекції, у якій записані представлення про фонемне звучання та їхню послідовність на перетворене значення передбачених довжин фонем, більш детально, це розглянуто на рисунку.

Таким чином виходить узагальнена матриця мел-спектрограм. Лише загальна матриця мел-спектрограм не підійде для результату синтезу, оскільки, якщо подивитися її відображення, як картинки і порівняти з результатом синтезу, то узагальнена мел-спектрограма буде мати розмитий вигляд, і якщо його вокодувати, то вийде аудіо дуже поганої якості. Узагальнена спектрограма буде використовуватися, як направляюча точка у алгоритмі flow matching і саме завдяки цій узагальненій картинці можливо отримати саме конкретний результат від flow matching, а не випадкове аудіо.

4.4.2 Реалізація декодера з Flow Matching

Як вже було розібрано, Flow Matching, сам по собі не може мати загального ефективного рішення, тому його необхідно симулювати завдяки нейромережам. Механізм симулювання Flow Matching буде реалізований, як декодер, за схожим принципом, як і у Glow-TTS був реалізований свій метод пошуку завдяки дифузії. Хоча загальний принцип і схожий, необхідно підмітити, які саме відмінності будуть у реалізації архітектури. Головне, це продовження, використання RoPE підходу, до блоків уваги моделі, це дозволить, як і зменшити розмірність даних так і загальне навантаження, як і на пам’ять так і на швидкість. Від зміни підходу, до уваги, також і зменшуються згорткові шари, оскільки немає необхідності обробляти багатовимірні дані, які до цього обраховувалися у декодері Glow-TTS. Також разом з цим і зменшиться розмірність значення даних часу для проходження шляху у Flow Matching.

Для симуляції поведінки векторного поля Flow Matching буде використано архітектуру U-Net, як і у Glow-TTS. Загальна архітектура моделі декодера зображено на діаграмі рисунку 4.2.

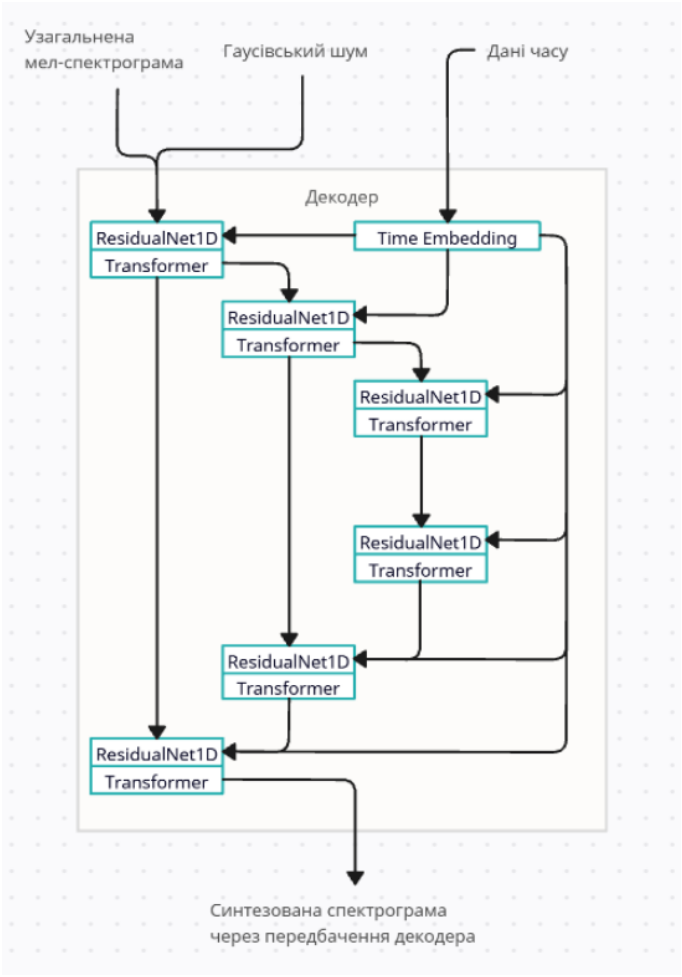


Рисунок 4.2 – Діаграма архітектури декодера моделі

Мережа буде складатися з послідовностей підвищення та зниження роздільності даних з послідовністю ResidualNet1D та Transformer. Таке пов’язане дозволить і реалізувати головну суть U-Net зі зміною роздільних здатностей разом зі згортковими перетвореннями, так і дозволити моделі завдяки Transformer-блокам аналізувати мел-спектрограму, та на її основі симулювати необхідні латентні дані для коректної симуляції переміщення по векторному полю швидкостей. Також важливим компонентом є додавання TimeStep Embedings до моделі, що означає, що кожен раз при виконанні forward функції Transformer буде приймати у свої параметри значення часу шляху, що є необхідним компонентом для симуляції Flow Matching.

Для навчання декодера на базі U-Net, аналогічно з енкодером, також необхідно реалізувати алгоритм, який воно буде симулювати, а саме Flow Matching. Навчатися декодер буде, як і було описано у теоретичному розборі Flow Matching, завдяки зашумленню істинної мел-спектрограми, у нашому випадку мел-спектрограми отриманої від енкодера, та обрахуванню різниці, у вигляді квадратичної помилки, між справжнім вектором швидкості, обрахованим алгоритмом та вектором швидкості, який передбачила модель.

4.4.3 Реалізація вокодера

Отримавши готову спектрограму від декодера моделі, її необхідно перетворити у готовий аудіо-файл, який можливо прослухати. У цьому проекту, буде реалізовано 3 види налаштування вокодування, це або за допомогою алгоритма Гріффіна-Ліма, або за допомогою іншого вокодера, або за замовчуванням вокодером iSTFTNet. Реалізація підключення інших вокодерів, буде потребувати налаштування моделі вокодера заздалегідь під необхідний формат даних, якщо вона вимагає додаткових параметрів окрім спектрограми, та її завантаження не буде реалізоване у цьому проекті. Алгоритм Гріффіна-Ліма у проекті буде реалізований для порівняння якості вокодування у порівнянні з вокодером iSTFTNet та оцінки різниці якості аудіо.

iSTFTNet, як де-факто окрема модель глибокого навчання, буде вже заздалегідь навчена на загальних аудіо-даних, а саме LJSpeech, та буде проведений fine-tuning на даних датасету, на якому буде навчатися основна модель «енкодер-декодер» зв’язки, відбуватися це буде паралельно під час навчання моделі на датасеті. Також буде додана можливість відключити функцію навчання, для LJSpeech, щоб запобігти перенавчанню та подальшому погіршенню якості синтезованих аудіо-даних.

4.5 Результати навчання моделі та синтезу

Після реалізації архітектури моделі, було розпочато її навчання. Навчання моделі, для замірів результатів відбувалося протягом 5 днів використовуючи один GPU GTX 1080. Навчання відбувалося, на двох датасетах окремо, LJSpeech, у якому аудіо-дані лише на англійській мові та на датасеті filatov-24000, який складається виключно з української мови.

Було досягнуто високої якості, у відношенні до витраченого часу та обрахувальних потужностей для навчання моделі. При навчанні та тестуванні було досягнуто результатів точності, для тренування 98%, а для тестування 90%. Просодія у порівнянні з класичними програмами для синтезу мовлення є дуже високої якості. При прослуховуванні синтезованого мовлення, чутно, як було сформовано унікальне звучання просодії мовлення людини. Розглянемо отримані спектрограми від моделі та порівняємо їх з істинним та з синтезованим мовленням від FastSpeech 2. Для перевірки було взято відрізок даних, з датасету LJSpeech, а саме фраза «It had established periodic regular review of the status of four hundred individuals», її спектрограма продемонстрована на рисунку 4.3.

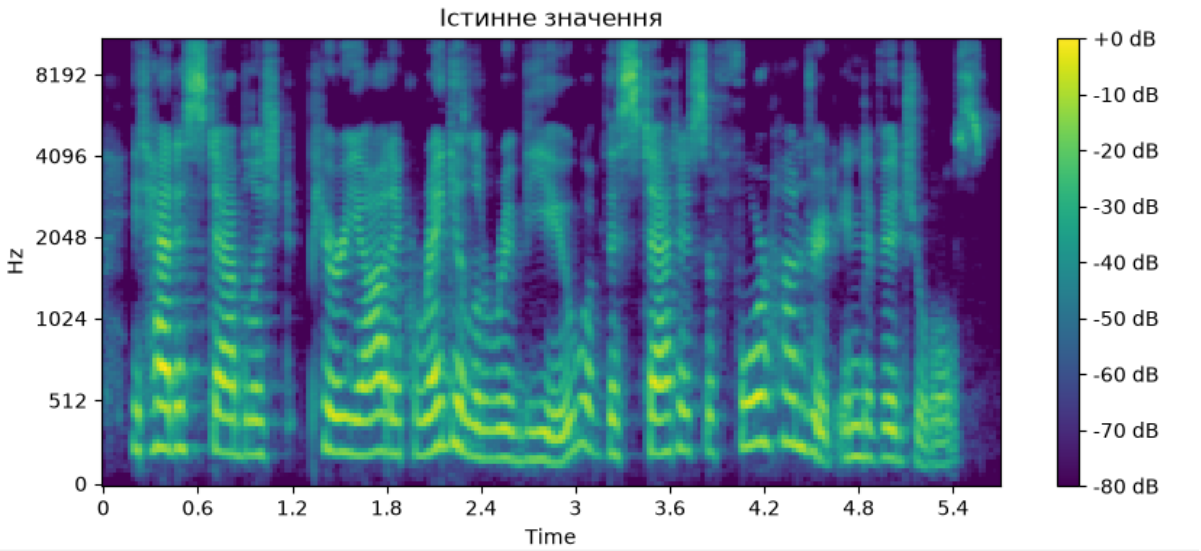


Рисунок 4.3 – Істинна мел-спектрограма мовлення

Це ж речення, було передано у розроблену модель, після чого було отримано синтезовану моделлю мел-спектрограму, яку продемонстровано на рисунку 4.4

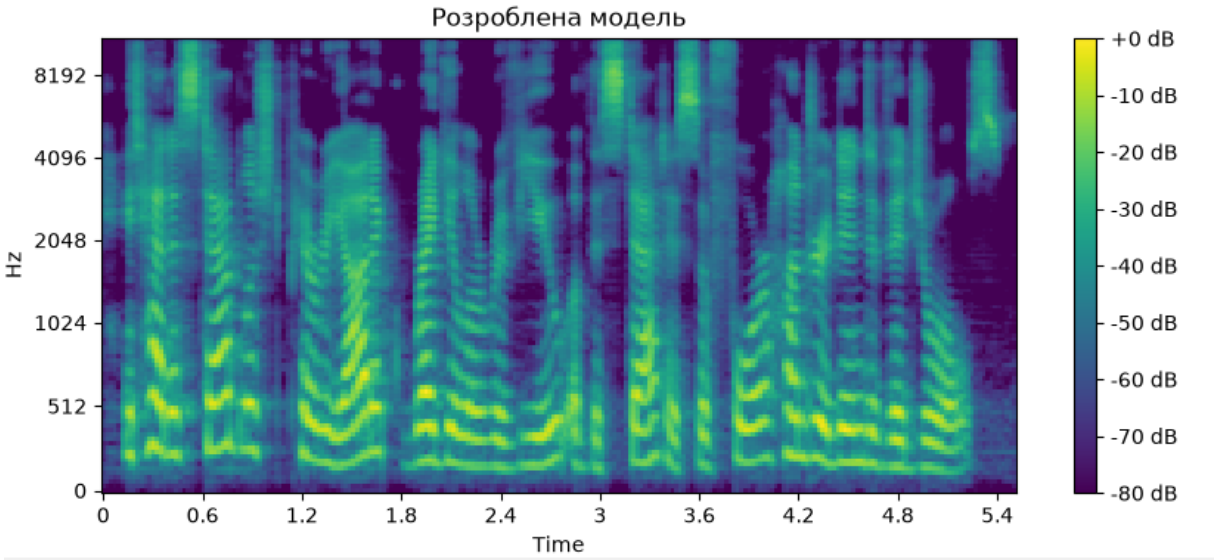


Рисунок 4.4 – Синтезована мел-спектрограма

У синтезованій мел-спектрограмі видно, як модель, навчилася синтезувати якісно форманти, які на спектрограмі представлені, як паралельні лінії з високою силою гучності, через що і підвищилась якість мовлення. Також у порівнянні можливо підмітити, як у більш високих значеннях частоти, у синтезованій спектрограмі втрачається чіткість значень і вони стають більш розмитими у порівнянні з оригінальними даними. У загальному виді видно, що модель навчилася оперувати головними фонетичними елементами, як правильно у часі вимовляти фонеми, їх коартикулювати між собою, правильно формувати форманти для фонем, та визначати головні елементи просодії. Також помітні загальні шуми на спектрограмі, які також відчутні на слух, через що якість, хоча і є високою, але не ідентичною до оригінального мовлення.

Порівняємо синтезоване мовлення розробленої моделі з мовлення FastSpeech 2, яке продемонстровано на рисунку 4.5

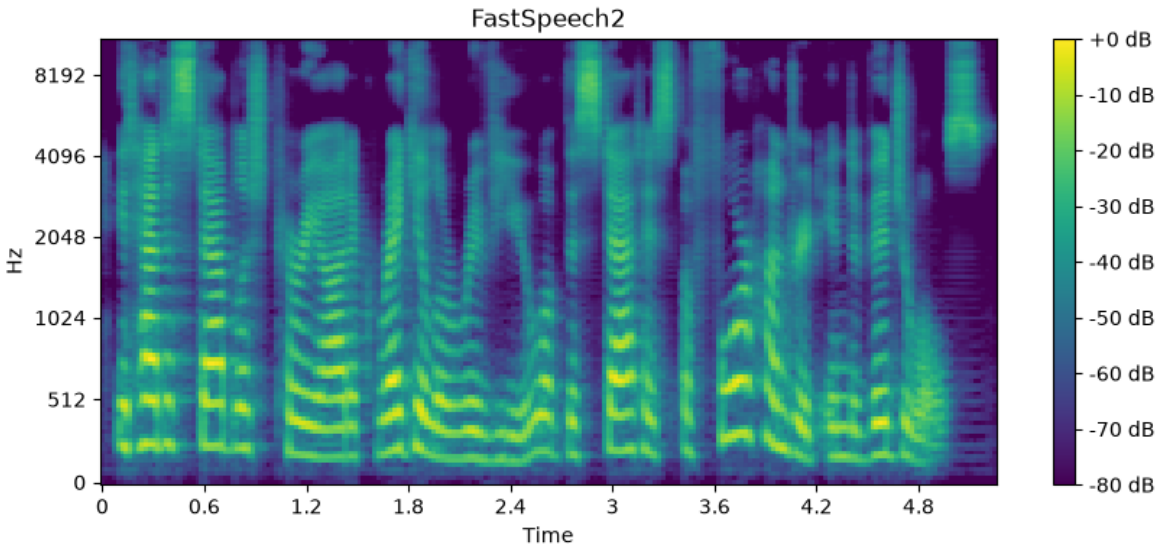


Рисунок 4.5 – Мел-спектрограма FastSpeech 2

Загально спектрограми дуже схожі, але у FastSpeech 2, менш чітка загальна картина частот, та присутні артефакти шуму, які видно у якості смуг на спектрограмі. Також у великих частотах видно дуже велику втрату якості у порівнянні з оригіналом. Усі ці фактор, дуже відчутні на слух, втрата чіткості частот, та артефакти, відчутні на слух, як оригінальне мовлення відтворюється з пошкоджених даних, у розробленої моделі, така проблема якості, відсутня.

Також модель здатна формувати довгі за звучанням аудіо-дані без втрати якості, або галюцинацій, завдяки прийнятим технологічним рішенням у енкодері. Синтез мовлення, як і було розібрано у теоретичній частині, є доволі швидким, та здатен формувати аудіо, з тексту у 1000 символів, від декількох секунд до десятка секунд.

4.6 Недоліки та можливі доповнення

Щодо якості генерованого аудіо, ще є простір для покращення звучання аудіо. Це можливо досягнути шляхом виділення більшої кількості необхідних ітерації reflow, для зменшення кількості кроків для генерації, а

відповідно до цього з’являється можливість просто залишити кількість кроків генерації та отримати кращий результат. При тому кращий результат не означає краща просодія, так навіть при запропонованих змінах, синтезоване мовлення все ще буде мати деякий відтінок роботизованості мовлення, при нестандартних промовляннях, або перепаду інтонацій між фонемами.

Також вокодер теж має свої особливості та вплив на просодію і майбутні рішення зможуть також покращити якість просодії, оскільки роботизованість на переходах фонем, також у більшій частині згладжується вокодером, це буде чути, якщо порівняти результат будь-якого вокодера та алгоритму Гріффіна-Ліма.

У синтезованому мовленні можливо почути унікальну просодію, та навіть розпізнати її звучання, як звучання окремої особистості, проте відсутній будь-який емоційний відтінок та забарвлення. Синтезоване звучання звучить доволі нейтрально, а від того і сприймається як монотонне, що шкодить загальному сприйняттю синтезованих даних. Можливий подальший розвиток для додавання регуляції емоцій, що буде потребувати модифікацій у енкодері та декодері та обробку датасету, для визначення ступеню емоцій. Як можлива пропозиція, то модифікація декодера з додаванням механізму, як у time embeddings, але з емоційним станом, щоб моделі симулювання було легше орієнтуватися у просторі з емоційними відтінками можливих мел-спектрограм.

Модель має можливість для подальшого розвитку та модифікацій. Оскільки створення даної моделі базувалося на модифікації або використанні готових рішень, для реалізації нового підходу для генерації аудіо завдяки дифузії, то є можливість, що при відкритті нового більш ефективного методу для генерації шляхом дифузії, можлива його швидка інтеграція у вже готову середу модулів, таких як енкодер та вже готовий вокодер.

ВИСНОВКИ

Під час дипломування було вивчено необхідні фундаментальні основи функціонування мовлення, фонетики, аудіо-даних. Було проаналізовано існуючі системи синтезу мовлення, їхні рішення для синтезу мовлення, та їхні переваги і недоліки. У результаті аналізу існуючих рішень та їхніх недоліків, було запропоновано використання механізму дифузії для генерації мел-спектрограм та їхнього перетворення у вокодері у аудіо-дані синтезованого мовлення. Для ефективного синтезу мовлення, було використано механізму Flow Matching, який дозволив вирішити, головний недолік дифузії, швидкість генерації.

Для програмної реалізації запропонованого рішення, були використані сучасні апробовані алгоритми та бібліотеки для взаємодії з нейронними мережами, обробки текстових даних та аудіо даних. Було реалізовано архітектуру «Джерело-Фільтр» у вигляді взаємодії енкодера та декодера. Також створено можливість для гнучкого вибору вокодерів для можливості у подальшому покращити результат синтезу завдяки зміні вокодера.

Розроблена модель, перевершує у швидкості та якості синтезованого мовлення, такі моделі, як розібрані у аналізі існуючих рішень, Tacotron та FastSpeech. Також модель перевершує за швидкістю синтезу мовлення, аналог у сфері синтезу мовлення завдяки дифузії, Glow TTS. Хоча за якістю синтезованого мовлення перевершують інші аналоги на архітектурі EnCodec, які більше спрямовані не на одночасне вирішення проблеми швидкості та якості синтезу, а лише на якість, серед таких моделей, кращу якість мають VibeVoice, QWEN.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Deep audio watermarks are shallow: limitations of post-hoc watermarking techniques for speech / P. O'Reilly et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2504.10782 (date of access: 20.06.2026).

2. Towards responsible evaluation for text-to-speech / Y. Yang et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2510.06927 (date of access: 20.06.2026).

3. Not my voice! A taxonomy of ethical and safety harms of speech Generators / W. Hutiri et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2402.01708 (date of access: 20.06.2026).

4. Do not mimic my voice: speaker identity unlearning for zero-shot text-to-speech / T. Kim et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2507.20140 (date of access: 21.06.2026).

5. Hallahan. W. DECtalk software: text-to-speech technology and implementation. *DECtalk Achive*. URL: https://dectalk.nu/Software%20and%20Manuals/Manuals/Miscellaneous/vol7num4art1.pdf (date of access: 21.06.2026).

6. Yu. S. Hidden semi-markov models. *ScienceDirect*. URL: https://doi.org/10.1016/j.artint.2009.11.011 (date of access: 21.06.2026).

7. Wavenet: a generative model for raw audio / A. Oord et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.1609.03499 (date of access: 21.06.2026).

8. Towards universal neural vocoding with a multi-band excited wavenet / A. Roebel et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2110.03329 (date of access: 22.06.2026).

9. HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis / J. Kong et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2010.05646 (date of access: 23.06.2026).

10. Tacotron: towards end-to-end speech synthesis / Y. Wang et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.1703.10135 (date of access: 24.06.2026).

11. Neural machine translation by jointly learning to align and translate / D. Bahdanau et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.1409.0473 (date of access: 25.06.2026).

12. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions / J. Shen et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.1712.05884 (date of access: 26.06.206).

13. FastSpeech: fast, robust and controllable text to speech / Y. Ren et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.1905.09263 (date of access: 26.06.2026)

14. Attention is all you need / A. Vaswani et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.1706.03762 (date of access: 26.06.2026).

15. Fastspeech 2: fast and high-quality end-to-end text to speech / Y. Ren et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2006.04558 (date of access: 27.06.2026).

16. Deep unsupervised learning using nonequilibrium thermodynamics / J. Sohl-Dickstein et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.1503.03585 (date of access: 28.06.2026).

17. Denoising diffusion probabilistic models / J. Ho et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2006.11239 (date of access: 28.06.2026).

18. Flow matching for generative modeling / Y. Lipman et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2210.02747 (date of access: 28.06.2026).

19. Flow straight and fast: learning to generate and transfer data with rectified flow / X. Liu et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2209.03003 (date of access: 29.06.206).

20. Glow-TTS: a generative flow for text-to-speech via monotonic alignment search / J. Kim et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2005.11129 (date of access: 30.06.2026).

21. ISTFTNet: fast and lightweight mel-spectrogram vocoder incorporating inverse short-time fourier transform / T. Kaneko et al. *ArXiv*. URL: https://doi.org/10.48550/arXiv.2203.02395 (date of access: 30.06.2026).