Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра інформатики

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: «**ГЕНЕРАЦІЯ МОВИ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ**»

Виконала: студентка 4-го року навчання,

Освітньої програми «Комп'ютерні науки», 122

Пархоменко Анастасія Олександрівна

Керівник Бучко О.А.,_____

кандидат фіз.-мат. наук, доцент

Рецензент _____

(прізвище та ініціали)

Кваліфікаційна робота захищена

з оцінкою _____

Секретар ЕК _____

«____» _____ 20____ р.

Київ – 2024

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Завідувач кафедри інформатики,

Кандидат фізико-математичних наук

Гороховський С.С.

_____

«____» _____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на кваліфікаційну роботу

студентки 4 курсу факультету інформатики

Пархоменко Анастасії Олександрівни

Тема: Генерація мови з використанням нейронних мереж
Зміст текстової частини кваліфікаційної роботи:

- Вступ
- Розділ 1. Теоретичні основи синтезу мовлення
- Розділ 2. Дослідження наявних моделей з використанням нейронних мереж
- Розділ 3. Адаптація моделі Glow TTS та впровадження допоміжних інструментів для навчання моделі на україномовному датасеті
- Висновки
- Список використаних джерел

Дата видачі «____» _____ 2024 р.
Керівник _____

Завдання отримано _____

ГРАФІК ПІДГОТОВКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

№ з/п	ПЕРЕЛІК РОБІТ	Термін виконання	Дата ознайомлення наукового керівника	Підпис наукового керівника	Примітки
1.	Вибір теми, затвердження її на засіданні кафедри та закріплення наукового керівника Узгодження календарного графіка підготовки кваліфікаційної роботи. Ознайомлення студента з критеріями оцінювання кваліфікаційної роботи (п. 8.5).	Жовтень			
2.	Вивчення джерел літератури, матеріалів архівів, періодичних видань, збір та узагальнення фактів, даних	жовтень – листопад			
3.	Складання плану каліф. роботи та узгодження з науковим керівником	Листопад			
4.	Написання розділів роботи *або* Постановка експерименту, аналіз отриманих результатів наукового дослідження	листопад – березень			
5.	Проміжний контроль виконання роботи	Лютий			
6.	Написання кваліфікаційної роботи в цілому, ознайомлення з її першим варіантом наукового керівника	січень – березень			
	Розділ 1 (постановка проблеми, теоретичні основи, огляд літературних джерел)				
	Розділ 2 (аналітично-дослідницька частина)				
	Розділ 3 (проектно-рекомендаційна частина)				
7.	Повне завершення написання кваліфікаційної роботи, оформлення її згідно з вимогами й подання на відгук науковому керівнику	квітень – початок травня			
8.	Подання кваліфікаційної роботи для перевірки письмових робіт студентів НаУКМА на відповідність вимогам академічної доброчесності,	середина травня			
9.	Подання на зовнішню рецензію	середина травня			
10.	Підготовка до захисту кваліфікаційної роботи на засіданні кафедри: написання доповіді та виготовлення ілюстративного матеріалу	**до 15 травня**			
11.	Попередній захист кваліфікаційної роботи на засіданні кафедри	**до 15 травня**			
12.	Подання кваліфікаційної роботи на кафедру з усіма супроводжувальними документами	**до 23 травня**			
13.	Публічний захист кваліфікаційної роботи перед екзаменаційною комісією	згідно з розкладом роботи ЕК			

Графік узгоджено «___» _____ 2024 р.
Науковий керівник _____ (ПІБ)

Виконавець кваліфікаційної роботи Пархоменко Анастасія Олександрівна

ЗМІСТ

Стор.

ВСТУП

У сучасному світі штучний інтелект стає невід'ємною частиною багатьох аспектів нашого життя, пропонуючи рішення, які значно підвищують ефективність та доступність технологій. Однією з найбільш перспективних галузей, де AI демонструє вражаючі успіхи, є генерація мовлення. Ця технологія відкриває широкі можливості для розробки інтерфейсів нового покоління, що можуть спілкуватися з користувачем максимально природньо та ефективно. Системи синтезу мовлення знайшли широке застосування в різноманітних сферах, включаючи системи асистування, реабілітаційні програми для людей із порушеннями зору, інтерактивні ігри і навіть в державних ініціативах.

Дипломна робота присвячена аналізу та дослідженню основних етапів розвитку технологій генерації мовлення, включно з переходом від традиційних архітектур до сучасних нейронних мереж, які значно покращили якість і натуральність синтезованого мовлення. В роботі розглядаються різноманітні моделі, включаючи такі, як Tacotron 2, WaveNet, FastSpeech та Glow TTS, аналізуються їхні сильні та слабкі сторони.

Актуальність дослідження визначається широким спектром застосування технологій генерації мовлення: від систем автоматизованого обслуговування клієнтів до асистентів у смартфонах і домашніх розумних пристроях, що вимагає підвищення якості та природності мовлення. Незважаючи на значний прогрес у розробці уніфікованих моделей, вже зазначених вище, актуальність досліджень у цій області висока через потребу покращення природності, інтонаційної різноманітності та ефективності TTS систем, особливо для рідкісних та регіональних мов.

Мета даної роботи – підбір та тренування нейронної моделі для генерації мовлення українською мовою, а також удосконалення звучання за допомогою фонетичного аналізу задля досягнення розбірливого і природнього звучання.

Основні завдання, які ставляться в роботі, включають: дослідження існуючих моделей генерації мовлення, адаптацію однієї з них для роботи з українською мовою та оцінку ефективності обраної моделі.

Предметом дослідження є процеси та механізми синтезу мовлення з тексту, зокрема розробка та оптимізація наскрізних нейронних мереж для синтезу мовлення на українській мові..

Об’єктом дослідження є нейронні мережі, які використовуються для генерації мовлення. Для досягнення поставлених завдань використовуються методи машинного навчання, зокрема глибоке навчання, а також методи обробки природної мови.

Таким чином, дипломна робота спрямована на дослідження та застосування ефективних методів синтезу українського мовлення.

Методи дослідження:

1) Аналіз літератури для з'ясування сучасних досягнень у технологіях синтезу мовлення.
2) Фонетичний аналіз для адаптації фонемного набору до особливостей української мови.
3) Машинне навчання і глибоке навчання для тренування та оптимізації моделей TTS.
4) Експериментальне тестування для оцінки якості для удосконалення згенерованого мовлення.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ СИНТЕЗУ МОВЛЕННЯ

1.1. Вступ до генерації мовлення

Синтез мови (Text-to-Speech, TTS) — це технологія, яка перетворює текстові дані на мовлення. Цей процес дозволяє комп'ютерам генерувати мовлення, яке імітує людський голос і інтонацію. Завданням системи TTS є створення зрозумілого і природнього звучання мови з текстового вводу. Математично процес можна представити як функцію, що відображає послідовність символів C у послідовність звукових сигналів S: $S = f(C)$, де f – це функція синтезу мови, яка може включати різні компоненти обробки.

Ця технологія відіграє важливу роль у різних сферах застосування - від засобів доступності для людей з вадами зору до інтерактивних систем голосового управління, віртуальних асистентів і мультимедійних розваг. Розвиток систем TTS значно просунувся за останні десятиліття, еволюціонуючи від простих роботизованих голосів до дуже природної та динамічної генерації мовлення.

Початкові зусилля в технології TTS були механічними і включали системи, які намагалися імітувати людську мову, використовуючи фізичні аналоги людських голосових органів. У 1950-х роках стартувала ера синтезу мовлення з розробки формантних синтезаторів. Перші цифрові системи покладалися процес генерації та комбінації набору основних тонів і формантів, що відповідають різним резонансним частотам голосового тракту. Одним з найвідоміших прикладів є синтезатор VODER, демонстрований на Всесвітній виставці в Нью-Йорку у 1939 році[1].

На наступному етапі з'явився метод лінійного передбачення (LPC), який став основою для синтезу мовлення в 1960-х і 1970-х роках. LPC аналізує звуковий сигнал, розбиваючи його на зразки, за допомогою яких може бути відтворена акустична

структура мовлення. Цей метод покращив якість синтезованого мовлення, роблячи його більш зрозумілим.

Конкатенативний синтез мовлення, що зазнав розквіту в кінці 1980-х, базується на об'єднанні коротких записів або звукових сегментів мовлення в логічній цілісній послідовності, яка наближалась до природньої. Ця техніка використовує величезні бази даних записаних розмовних уривків. Успіхи в технологіях зберігання даних дозволили значно розширити обсяги цих баз, що зробило конкатенативний синтез популярним методом.

З розвитком машинного навчання та нейронних мереж, новий клас синтезу мовлення — параметричний та нейронний — став можливим. На відміну від конкатенативного, параметричний синтез використовує набір правил, а точніше, математичні моделі, для створення мови з фонетичної та просодичної інформації, закодованої в тексті. Ці моделі визначають параметри звукового сигналу, такі як основний тон, інтенсивність і форманти, які потім синтезуються для створення мовленнєвого виводу. Незважаючи на свою ефективність, ці методи часто створюють мову, яка, хоча і є зрозумілою, але позбавлена природної плавності та емоційності людського мовлення. Параметричний синтез може використовувати різноманітні методи, включаючи традиційні статистичні підходи, такі як приховані моделі Маркова (HMM), або більш сучасні методи, засновані на глибокому навчанні, як нейронні мережі. Власне, інтеграція машинного навчання, особливо глибокого навчання, зробила революцію в системах TTS. Нейронні мережі дозволили розробити моделі, які можуть навчатися на великому обсязі розмовної мови, щоб генерувати мовлення, яке точно імітує людські інтонації та ритми. Варто відзначити два значних досягнення:

1) застосування архітектури глибокого навчання: такі моделі, як Tacotron і WaveNet, змістили парадигму від синтезу на основі правил до моделей, які навчаються і генерують мову безпосередньо з текстових і аудіо даних. Ці моделі використовують

складні архітектури, які обробляють різні аспекти мовлення, зокрема тон, наголос і потік, що значно підвищує природність синтезованого голосу.

2) неавторегресійні моделі та моделі на основі потоку: нещодавні інновації, такі як FastSpeech і Glow TTS, зосереджені на підвищенні швидкості та ефективності генерації мовлення, усуваючи один із суттєвих недоліків попередніх нейронних моделей TTS. Ці моделі зменшують затримку у виведенні мовлення, що робить їх придатними для додатків у реальному часі без втрати якості.

Незважаючи на величезний прогрес, у технології TTS залишається кілька викликів. Досягнення емоційної виразності та адаптація до різних мов, особливо з обмеженим обсягом даних, є постійними напрямками досліджень. Крім того, прагнення підвищити ефективність моделей без втрати якості синтезу продовжує стимулювати інновації в цій галузі.

1.2. Основні компоненти аналізу тексту для синтезу мовлення

Першим кроком в системі TTS є аналіз тексту. Цей процес включає нормалізацію тексту, токенізацію і розбір речень на складові частини.

Нормалізація тексту – це процес перетворення вхідного «сирого» тексту, що містить спеціальні позначки, числа, символи, абревіатури на повні слова: $\mathrm{T}_{\mathrm{norm}} = \mathrm{f}_{\mathrm{norm}}(\mathrm{T})$, де T містить елементи, як-от "3D", а T_{norm} містить елементи "три D".

Токенізація може бути представлена як функція f_{token} , яка розділяє нормалізований текст T_{norm} на токени W (слова або фрази): $W = f_{token}(T_{norm})$, що є основою для фонетичної транскрипції і синтаксичного аналізу. Тут W є вектором токенів $w_1, w_2, \dots, w_n$, наприклад, ["три", "D"] для вхідного "три D".

Сегментація тексту в системах TTS дозволяє перетворювати великі текстові блоки на зручніші для обробки одиниці, такі як речення та слова. Технології машинного навчання, такі як умовні випадкові поля (CRF), дозволяють вдосконалити

цей процес шляхом використання більш складних моделей для ідентифікації меж речень, що значно підвищує точність в складних мовних структурах: $P(y|x) = \frac{1}{Z(x)} \exp(\sum_{t=1}^{T} \sum_k \lambda_k f_k(y_{t-1}, y_t, x_t))$, де x — вхідний текст, y — послідовність міток, що визначають межі речення, f_k — функції ознак, що оцінюють стан і переходи між мітками, λ_k — вагові коефіцієнти, $Z(x)$ — нормалізаційний фактор, що забезпечує, щоб сума ймовірностей всіх можливих послідовностей міток дорівнювала 1[2].

Морфологічний аналіз визначає структуру слова, включаючи його корінь, афікси та частину мови, що є важливим для правильної інтерпретації і генерації мовлення. Алгоритми двонаправленої довгострокової короткочасної пам'яті (BiLSTM) використовуються для досягнення цієї мети: $\overrightarrow{h_t} = \text{LSTM}(\overrightarrow{\text{h}_{\text{t}-1}}, \text{x}_\text{t})$- це рівняння описує прямий прохід LSTM, де $\overrightarrow{h_t}$ є прихованим станом на кроці t, залежним від попереднього стану $\overrightarrow{\text{h}_{\text{t}-1}}$ та поточного входу x_t. $(\leftarrow)\overline{\text{h}_\text{t}} = \text{LSTM}(\overleftarrow{h_{t+1}}, x_t)$ - це рівняння описує зворотній прохід LSTM, де $\overline{\text{h}_\text{t}}$ є прихованим станом на кроці t, який обраховується в зворотному напрямку(від майбутнього до теперішнього) та залежить від наступного стану $\overleftarrow{h_{t+1}}$ та поточного входу x_t . $y_t = \text{softmax}(W[\overrightarrow{h_t}; \overline{h_t}] + b)$ - це кінцеве рівняння, яке визначає вихід y_t як софтмакс функцію від лінійної комбінації об’єднаних прямого та зворотнього прихованих станів, зважених вагами W зі зміщенням b (навчальні параметри моделі)[3].

Наступним кроком у текстовій обробці є синтаксичний аналіз, який полягає в побудові моделей для ідентифікації граматичних структур в реченнях та встановленні залежностей між словами. Найпоширеніші парсери залежностей, такі як Stanford Parser і spaCy, використовують графові структури для представлення синтаксичних зв'язків між словами, що дозволяє системі TTS точно відтворювати синтаксичні особливості мовлення. У графовій структурі для синтаксичного аналізу важливою є матриця суміжності, яка вказує, які слова пов'язані між собою. Кожен елемент цієї матриці може відображати наявність (або відсутність) зв'язку між двома словами. Наприклад,

якщо між словами "серце" і "б'ється" існує синтаксична залежність, то відповідний елемент у матриці суміжності буде ненульовим. $H^{(l+1)} = \sigma\left(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$, де A — матриця суміжності, яка представляє залежності між вузлами (словами), D — діагональна матриця ступенів вузлів, $H^{(l)}$ — вхідні характеристики вузлів на l-тому шарі, $W^{(l)}$ — вагові параметри на l-тому шарі, σ — нелінійна функція активації. Ця формула є характеристичною для моделей на основі графових згорткових мереж. Вона представляє оновлення вузлів у шарі графової нейронної мережі[4].

1.3. Фонеми та їх роль в моделях TTS

Для підвищення якості генерованого мовлення також використовують фонетичну транскрипцію, де текстові дані перетворюються на послідовності фонемних репрезентативів - найменших структурних звукових одиниць мови. Математично, це може бути представлено через відображення G: $P_i = G(w_i)$, де $w_i\$$ — і-те слово у W, а P_i — відповідні фонеми.

У більшості мов письмовий текст не завжди точно відповідає вимові, що вимагає використання певного символічного представлення для опису правильної вимови. Кожна мова має свій фонетичний алфавіт, який складається з різного набору можливих фонем та їхніх комбінацій, причому загальна кількість фонетичних символів у кожній мові може варіюватися від 20 до 60. Зокрема, в англійській мові існує близько 40 фонем, але через складність та різноманітність фонематичних систем, точне визначення кількості фонем в англійській чи будь-якій іншій мові є непростим завданням.

Фонетичний алфавіт поділяється на дві основні категорії: голосні та приголосні звуки. Голосні завжди є дзвінкими звуками, які вимовляються завдяки вібрації голосових зв'язок, тоді як приголосні можуть бути як дзвінкими, так і глухими. Звуки голосних мають більшу амплітуду і стабільність, що полегшує їх акустичний аналіз та

синтез. Натомість приголосні, які швидко змінюються, є складнішими для аналізу та синтезу в системах генерації мовлення.

З огляду на складність фонетичних систем, у минулому були спроби створити універсальні фонетичні алфавіти, такі як Міжнародний фонетичний алфавіт (IPA), який охоплює великий набір символів для позначення фонем, тонів, контурів словесного наголосу та діакритичних знаків. IPA сприяє стандартизації запису фонем і уніфікації процесів транскрипції на глобальному рівні, оскільки включає символи для приблизно 107 звуків, які охоплюють широкий спектр мовних систем.

Фонетична транскрипція є ключовим елементом систем TTS, адже вона дозволяє перетворити написаний текст на аудіоформат, що максимально наближений до натуральної людської вимови. Для цього процесу використовується як вищезазначений Міжнародний фонетичний алфавіт (IPA), так і різноманітні алгоритми автоматичного фонетичного кодування (G2P).

Алгоритми G2P можуть використовувати класичні лінгвістичні правила або машинне навчання для адаптації до особливостей конкретної мови. Зокрема, глибокі нейронні мережі дозволяють системам TTS навчатися безпосередньо з великих обсягів мовних даних, значно підвищуючи точність фонетичної транскрипції. Такі алгоритми також можуть ефективно обробляти діалектні варіації та винятки, роблячи транскрипцію більш гнучкою та адаптивною. Окрім того, фонетичні словники, як-от CMU Pronouncing Dictionary, що використовує ARPAbet для англійської мови, забезпечують важливі ресурси для TTS-систем. ARPAbet, розроблений дослідниками з Carnegie Mellon, включає символи для фонем англійської мови, які відображають голосні, приголосні, а також інші аспекти вимови, такі як акцент та тон.

1.3.1. Використання графем та фонем при обробці тексту

Сучасні реалізації систем TTS надають можливість використати два підходи обробки тексту до синтезу мовлення: графеми (письмове представлення слів) та

фонеми (відокремлені одиниці звуку) в поєднанні з аудіодоріжками. Порівнюючи ці методи, можна виявити кілька факторів, які впливають на вибір залежно від конкретних потреб та контекстів. З одного боку, використання графем спрощує процес навчання, оскільки воно безпосередньо відображає письмовий текст на усну мову без потреби в проміжному фонетичному транскрибуванні. Це може бути особливо вигідним у мовах з послідовною фонетичною структурою. Сучасні моделі з використанням seq2seq алгоритмів, такі як Tacotron та її альтернативи, можуть навчатися безпосередньо від даних графем, уникаючи потреби в явних фонетичних анотаціях, що може знизити навантаження, пов'язане зі створенням та підтримкою фонетичних словників. Проте, графеми не завжди представляють фонетичні нюанси та варіації, особливо в мовах з нерегулярною ортографією. Це може призвести до неточностей у вимові, де правила є не прямолінійними.

З іншого ж боку, фонеми надають більш точний контроль над вимовою, оскільки вони призначені для представлення кожного окремого звуку. Це особливо корисно для мов зі складною фонетичною структурою або нерегулярними правилами написання. Фонеми дозволяють ефективно керувати різними акцентами, діалектами та варіаціями мовлення, що може бути важливим для додатків, які вимагають високих рівнів зрозумілості та природності мовлення. Використання фонем зазвичай вимагає попереднього етапу обробки для перетворення графем на фонеми, що часто включає фонетичний словник або модель перетворення графем у фонеми. Це додає складності системі та вимагає додаткових ресурсів для розробки точних фонетичних транскрипцій або надійних моделей G2P, з чим ми і зіштовхнемося у подальших розділах.

1.4. Просодії та методи їх обчислень

Просодія – це набір знань, завдяки яким ми можемо окреслити ритм, інтонацію, темп та інші аспекти мовлення, які перевищують структуру окремих звуків або фраз.

Модулі просодії, якщо такі наявні в моделях, визначають висоту тону, тривалість звучання фонем та їх інтенсивність, які адаптуються згідно з контекстом у реченні, які генератор мовлення перетворює ці просодично анотовані фонеми в аудіосигнал.

При хорошому контролі над просодичними характеристиками, можна добре моделювати стать, вік, емоції та інші особливості мовлення. Однак, здається, майже все впливає на просодичні особливості природного мовлення, що робить точне моделювання дуже складним. Просодичні особливості можна розділити на кілька рівнів, таких як склад, слово або фраза. Наприклад, на рівні слова голосні звуки більш інтенсивні, ніж приголосні. На рівні фрази правильну просодію створити складніше, ніж на рівні слова. Контур висоти тону або основна частота в реченні (інтонація) в природному мовленні - це поєднання багатьох факторів. Контур висоти тону залежить від змісту речення. Наприклад, у звичайній мові висота тону трохи знижується до кінця речення, а коли речення має форму питання, то контур висоти тону підвищується до кінця речення. В кінці речення може також спостерігатися продовження підйому, що вказує на те, що мова буде продовжуватися. Підвищення або зниження основної частоти також може вказувати на наголошений склад. Нарешті, на контур висоти тону також впливає стать, фізичний та емоційний стан і ставлення мовця.

Тривалість або часові характеристики також можна досліджувати на кількох рівнях: від тривалості фонем до тривалості речень, темпу та ритму мовлення. Сегментна тривалість визначається набором правил для визначення правильного хронометражу. Зазвичай деяка властива тривалість фонеми змінюється за правилами між максимальною та мінімальною тривалістю. Загалом, тривалість фонеми відрізняється через сусідні фонеми. На рівні речення важливими є швидкість мовлення, ритм і правильне розміщення пауз для правильного розмежування фраз[5].

Патерн інтенсивності сприймається як гучність мови в часі. На рівні складів голосні зазвичай більш інтенсивні, ніж приголосні, а на рівні фраз склади в кінці

висловлювання можуть ставати слабшими за інтенсивністю. Інтенсивність у мовленні тісно пов'язана з основною частотою.

Підкреслити висоту тону та інтонацію можливо за допомогою спеціальних символів IPA або ARPAbet, наприклад, IPA має деякі символи для позначення тону, як-от, знаки зниження та піднесення (˩, ˥). У деяких системах TTS можуть бути використані додаткові параметри для контролю висоти тону і акценту. Ці параметри можуть бути передані в модель синтезу мови разом з фонемізованим текстом і використовуватися для генерації мови з відповідними інтонаційними властивостями. Однією з комплексних технологій, завдяки яким це можливо, є спектральний аналіз.

Основні методи спектрального аналізу включають швидке перетворення Фур'є (FFT) і лінійне прогнозування (LPC). Ці методи дозволяють ефективно аналізувати частотні складові звукового сигналу та ідентифікувати форманти, які є критичними для визначення тембру мовлення.

Швидке перетворення Фур'є — це алгоритм, що дозволяє ефективно виконувати перетворення Фур'є, переводячи часовий сигнал у частотний спектр. Це важливо для аналізу мовлення в системах TTS, оскільки дозволяє виділити основні частотні компоненти звуку, які визначають його характеристики. Для дискретного сигналу $x[n]$, де n є індексом часу, FFT перетворює цей сигнал у набір комплексних чисел, що представляють амплітуду та фазу для кожної частоти: $\mathrm{X[k]}\sum_{\mathrm{n=0}}^{\mathrm{N-1}}\mathrm{x\,[n]}\cdot \mathrm{e}^{-\mathrm{i}2\pi n/N}$, де: X[k] — спектральні компоненти сигналу, N — кількість точок у дискретному сигналі, k — індекс частоти. FFT широко використовується для аналізу мовлення, дозволяючи виявляти форманти та інші важливі акустичні характеристики, що впливають на тембр та ясність мовлення.

Лінійне прогнозування — це метод аналізу, який використовується для оцінювання спектральних характеристик голосового сигналу. LPC моделює сигнал як комбінацію його попередніх значень, пропонуючи спосіб представлення формантної структури мовлення за допомогою невеликої кількості параметрів: $\hat{x}[n] = -\Sigma_{k=1}^{p} a_k \cdot$

$x[n-k]$, де: $\hat{x}[n]$ — прогнозоване значення сигналу в часовий момент n, $x[n-k]$ — фактичні значення сигналу в попередні моменти часу, a_k — коефіцієнти LPC, які мінімізують помилку прогнозування, p — порядок моделі LPC. Коефіцієнти LPC визначаються таким чином, щоб мінімізувати сумарну квадратичну помилку між фактичними значеннями сигналу та їх прогнозами. Ці коефіцієнти використовуються для визначення формантів, які є ключовими для забезпечення природності мовлення[6].

Оскільки процес фонетичної транскрипції і просодії і нині залишається важким комплексним питанням, багато моделей залишають можливість послуговуватись графемами - що є найменшими вже писемними одиницями мови, тобто буквами або комбінаціями букв. Але варто враховувати, що моделі, які тренуються для мов зі значними графем-фонемними відмінностями, як-от англійська чи французька, матимуть проблеми з коректністю вимови, її природністю і зрозумілістю.

Комбінуючи вищезгадані елементи, загальну модель синтезу мовлення в системі TTS можна виразити як: $S = G(\pi(P)$. Ця формула показує, що аудіосигнал S генерується з послідовності фонем P через модулі просодії π і генерації мовлення G.

1.5. Висновки до розділу 1

У першому розділі ми розглянули широкий спектр аспектів, що стосуються генерації мовлення, зосереджуючись на основних етапах процесу, від фонетичної транскрипції до генерації просодії та кінцевого мовленнєвого сигналу. Особливу увагу було приділено розгляду історичного розвитку технологій TTS, переходу від механічних систем до сучасних цифрових методів, таких як конкатенативний та параметричний синтез, і до впровадження глибокого навчання, яке суттєво підняло якість синтезованого мовлення. Було виявлено, що використання нейронних мереж в системах TTS дозволяє не тільки поліпшити природність мовлення, але й розширює можливості їх застосування завдяки здатності моделювати складні мовні патерни.

Велику увагу було приділено також фонетичній теорії, яка є основою для розуміння процесів, що лежать в основі TTS. Ми розглянули, як сучасні системи обробляють і перетворюють текст у фонеми, що є нелегким завданням через плавучу природу фонемних одиниць і їх контекстно-залежні варіації, які становлять труднощі для формалізації та стандартизації у глобальних масштабах.

Окрім технічних аспектів, було висвітлено і виклики, з якими стикаються розробники TTS систем, особливо у контексті досягнення високої емоційної виразності та адаптації до мовних особливостей різних мов. Виклики, пов'язані з адаптацією систем TTS до менш ресурсомістких мов, як українська, вимагають особливої уваги та інноваційних підходів для подолання мовних бар'єрів.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ НАЯВНИХ МОДЕЛЕЙ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

2.1. Огляд традиційних архітектур для синтезу мовлення

Синтез мовлення — це останній етап у системі TTS, де фонемний репрезентатив перетворюється на аудіосигнал, який імітує людське мовлення. Цей процес може бути реалізований за допомогою різних технологічних підходів, зокрема, формантного, конкатенативного та параметричного синтезу.

Формантний синтез є методом, що використовує моделювання звукового тракту людини для генерації мовлення. Він базується на синтезі звуків шляхом керування формантами — піковими резонансами в звуковому спектрі, які характерні для людського голосу. У технічному виконанні формантний синтезатор складається із декількох ключових компонентів: генератора джерела звуку, формантних фільтрів та системи керування артикуляцією. Генератор джерела створює основний звуковий сигнал, який є аналогом вібрації голосових зв'язок. Далі, цей сигнал проходить через набір формантних фільтрів, кожен з яких моделює певний резонансний пік у голосовому тракті. Система керування артикуляцією дозволяє змінювати параметри фільтрів в реальному часі для імітації різних мовних звуків та емоційних відтінків.

Конкатенативне перетворення тексту в мовлення - одна з найбільш традиційних форм синтезу мовлення, яка відома тим, що дозволяє створювати мовлення, яке звучить дуже природно. Цей підхід конструює мову шляхом об'єднання попередньо записаних аудіосегментів (часто їх називають блоками), які ретельно відбираються і з'єднуються разом на основі вхідного тексту. Конкатенативні TTS покладаються на велику базу даних записаного мовлення, з якої вони виділяють і об'єднують короткі сегменти, щоб сформувати цілісні висловлювання. Ці сегменти можуть бути фонемами, складами, словами або навіть довшими фразами. Мовленнєва база даних – це ретельно

підготовлена база даних, яка містить безліч екземплярів мовленнєвих одиниць, записаних у контрольованому середовищі. Ці одиниці анотовані фонетичною, просодичною та контекстуальною інформацією для полегшення точного пошуку та відбору. Алгоритм виділення одиниць є ядром конкатенативної системи TTS, яка вибирає найбільш відповідні сегменти, що відповідають вхідному тексту. Цей процес можна математично змоделювати як оптимізаційну задачу. Процес виділення одиниць можна змоделювати як пошук оптимальної послідовності одиниць $u_1, u_2, \dots, u_n$ з бази даних, що мінімізує функцію витрат C, враховуючи цільові та конкатенаційні витрати: $C = \sum_{i=1}^{n-1} C_{target}(u_i, t_i) + C_{concat}(u_i, u_{i+1})$, де: $C_{target}(u_i, t_i)$- цільове значення, що відображає різницю в акустичних і просодичних характеристиках між обраним пристроєм u_i і очікувана ціль t_i, $C_{target}(u_i, t_i)$ - є конкатенативним значенням, що представляє акустичну дисперсію на межі між двома послідовними блоками u_i і $u_{\{i+1\}}$. Цільове значення зазвичай включає кілька вимірів, таких як фонетична точність, просодія та контекстуальна доречність. Вона може бути виражена як: $C_{target}(u_i, u_{i+1}) = w_p \cdot d_p\big(p(u_i), p(t_i)\big) + w_f \cdot d_f\big(f(u_i), f(t_i)\big) + \cdots$, де $p(u_i)$ і $p(t_i)$ представляють фонетичні особливості одиниці та цілі відповідно, d_p - це метрика відстані (наприклад, евклідова) для фонетичних ознак. w_p, w_f це ваги, присвоєні різним ознакам (фонетичним, просодичним тощо). Значення конкатенації оцінюється на основі акустичних властивостей в точці з'єднання між блоками: $C_{concat}\big(u_i, u_{\{i+1\}}\big) = d_a\Big(a(u_i), a\big(u_{\{i+1\}}\big)\Big)$, де $a(u_i)$ і $a\big(u_{\{i+1\}}\big)$ це акустичні особливості в кінці u_i і на початку $u_{\{i+1\}}$, а d_a є придатною метрикою для вимірювання акустичної безперервності, наприклад, спектрального розриву або RMSE (середньоквадратичної похибки). Задача оптимізації полягає в переборі можливих комбінацій одиниць для знаходження послідовності, яка мінімізує остаточне значення C. Враховуючи складність проблеми, для ефективного пошуку наближеного до оптимального рішення часто використовують евристичні алгоритми, такі як динамічне програмування або A^*-

перебір. Після вибору найкращих блоків до меж між блоками можна застосувати деякі методи згладжування або обробки сигналу, такі як лінійне предиктивне кодування (LPC) або пітч-синхронне перекриття-добавлення (PSOLA), щоб підвищити природність мовлення на виході[7]. Параметричний TTS синтезує мовлення шляхом використання математичних моделей для генерації мовлення з параметричної інформації, такої як основний тон, інтонація та довжина звуків, які перетворюються в чутний звук за допомогою модуля синтезу. На відміну від конкатенативного TTS, який безпосередньо використовує записані мовленнєві сегменти, параметричний TTS моделює генеративний процес виробництва мовлення за допомогою статистичних методів і включає два основні етапи: по-перше, модель генерації параметрів прогнозує параметри мовлення з тексту; по-друге, модель синтезу перетворює ці параметри у форму сигналу. Параметричний синтез може включати генерування основного тона $F_0(t)$ і формування звукового сигналу на основі цього тона: $s(t) = A(t) \cdot sin(2\pi J_0^t F_0(\tau)d\tau + \Phi)$, де $A(t)$ — амплітуда сигналу, що може змінюватися в часі, $F_0(t)$ — основний тон у часі, а Φ — початкова фаза. У параметричній системі TTS акустичне моделювання може бути виражене як функція f , що відображає набір лінгвістичних ознак L у набір акустичних параметрів A : $A = f(L; \theta)$, де L - лінгвістичні ознаки, отримані з вхідного тексту, A представляє акустичні параметри, такі як спектральні криві, контур висоти тону та тривалість, а θ позначає параметри моделі, які вивчаються під час навчання. Модель f зазвичай навчається за допомогою набору даних записів мовлення та їхніх відповідних текстових транскрипцій. Метою є мінімізація функції втрат $\mathcal{L}$, яка кількісно визначає різницю між передбаченими акустичними параметрами $\hat{A}$ та їх істинними значеннями A:$\min_\theta \mathcal{L}(\hat{A}, A)$. Ця функція втрат може бути середньоквадратичною похибкою (MSE) для неперервних параметрів, таких як F_0, або втратою перехресної ентропії для категоріальних параметрів, таких як тотожність фонем. Методи оптимізації, такі як стохастичний градієнтний спуск (SGD), використовуються для оновлення параметрів θ з метою мінімізації втрат $\mathcal{L}$. Цей процес

вимагає значної кількості мовних даних, щоб охопити різні аспекти людського мовлення і гарантувати, що модель може добре узагальнювати різні вхідні дані. Етап синтезу передбачає перетворення передбачених акустичних параметрів у цифрову форму сигналу. Зазвичай це досягається за допомогою методів обробки сигналів, таких як $s(t) = Vocoder(A)$, де $s(t)$ - синтезований мовний сигнал як функція часу t , а $Vocoder(A)$ - функція, реалізована вокодером для перетворення акустичних параметрів у звук[8].

2.2. Наскрізні системи TTS

Наскрізні системи TTS представляють собою значний перехід від традиційних багатоступеневих архітектур TTS до уніфікованої моделі, яка безпосередньо перетворює текст у мовлення. Цей підхід використовує глибоке навчання для спрощення конвеєра TTS, зменшуючи потребу в обширній лінгвістичній та акустичній інженерії. За допомогою глибоких нейронних мереж створюється відображення послідовностей символів або фонем безпосередньо в аудіосигнали або кадри спектрограми. Ці системи інтегрують весь процес синтезу мови в єдину модель, від аналізу тексту до генерації хвильових форм, оптимізуючи традиційні етапи обробки тексту, акустичного моделювання та синтезу звуку. Першим етапом вхідний текст перетворюється на послідовність токенів, які можуть бути символами, підсловами або фонемами. Ця послідовність слугує вхідними даними для нейронної мережі. Далі глибока нейронна мережа, часто заснована на таких архітектурах, як моделі «sequence-to-sequence»(seq2seq) з механізмами уваги, трансформерні або згорткові нейронні мережі, обробляє вхідні маркери для прогнозування аудіо вихідних даних. Завершальним етапом відбувається перетворення спектрограми у форму сигналу: якщо мережа виводить спектрограму, вокодер або нейронний генератор сигналів перетворює її на остаточну форму звукового сигналу. Для цього зазвичай використовуються такі

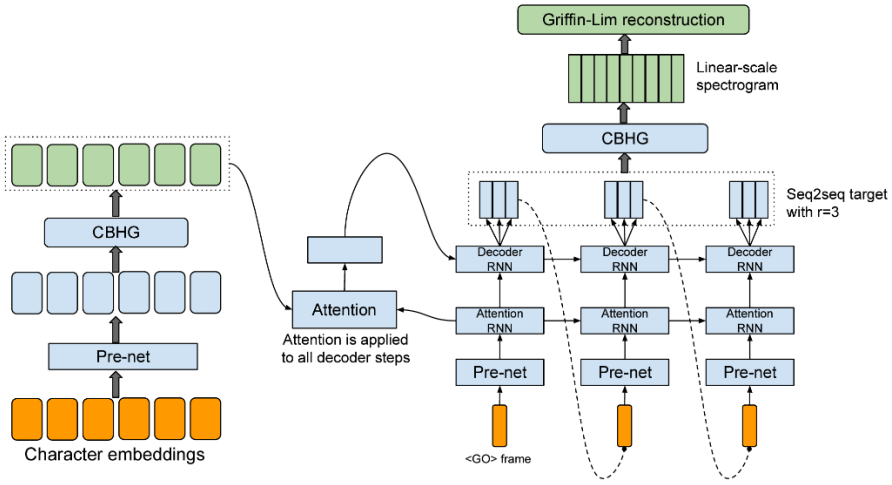
моделі, як WaveNet, Griffin-Lim або WaveGlow. Ядром наскрізної системи TTS є модель seq2seq з архітектурою кодера-декодера. Багато наскрізних моделей TTS включають механізм уваги, який динамічно фокусується на різних частинах текстового вводу під час генерації аудіовиводу. Наскрізні моделі TTS зазвичай навчаються на великому корпусі пар <текст, аудіо>. Метою навчання є мінімізація функції втрат, яка вимірює розбіжність між прогнозованим і реальним звуком.

Вибір функції втрат може варіюватися, але вона часто включає такі компоненти, як втрати за мел-спектрограмою або втрати на основі форми сигналу, якщо в модель інтегровано генерацію форми сигналу. Синтез мовлення відбувається в дві фази: декодер генерує спектрограму, яка представляє частотні компоненти мови в часі, а вокодер або нейронна модель синтезу перетворює спектрограму в часову форму сигналу.

Tacotron, розроблена командою Google, представляє собою наскрізну систему синтезу мовлення, яка безпосередньо перетворює текст на мовлення. Система використовує seq2seq модель з увагою, яка дозволяє ефективно моделювати мовні карти з тексту. Tacotron включає дві основні компоненти: кодер, який перетворює текст у контекстуалізовані вектори, і декодер, що генерує спектрограму з цих векторів. Завдяки впровадженню механізму уваги, Tacotron здатен виробляти зв'язне та природне мовлення. Модель використовує фреймовий підхід для генерації мовлення, що забезпечує значну перевагу у швидкості над методами авторегресії на рівні вибірки та зменшує залежність від лінгвістичних даних експертів.

Модель "sequence-to-sequence" – це тип архітектури нейронної мережі, що використовується переважно для завдань, які передбачають перетворення однієї послідовності в іншу, наприклад, переклад тексту з однієї мови на іншу або перетворення тексту в мовлення. Він складається з двох основних компонентів: кодера і декодера. Кодер обробляє вхідну послідовність і перетворює інформацію у вектор контексту - представлення вхідної послідовності фіксованої довжини. Потім декодер

бере цей вектор і крок за кроком генерує вихідну послідовність. Ця модель часто включає механізми уваги, які дозволяють декодеру зосереджуватися на різних частинах вихідних даних кодера, покращуючи його здатність обробляти довгі вхідні послідовності та підвищуючи точність вихідних даних[9].



1. Character Embeddings: Тут вхідний текст перетворюється у векторні представлення, відомі як символьні вбудовування (embeddings), які кодують кожен символ у багатовимірному просторі.
2. Pre-net: Цей шар обробляє вбудовування символів, зазвичай використовуючи повнозв'язні шари (fully connected layers) для отримання більш складних представлень, що можуть включати контекстуальну інформацію.
3. Encoder CBHG: CBHG (convolutional bank + highway network + GRU (Gated Recurrent Unit)) – це модуль, який обробляє проміжні представлення від Pre-net. Він складається з набору згорткових шарів (convolutional layers), які застосовуються паралельно для захоплення місцевих залежностей, магістральної мережі (highway network) для адаптації потоку інформації, і рекурентної мережі (GRU) для захоплення послідовностей, що частково вирішує проблему зникання або вибуху градієнтів.
4. Attention Mechanism: Механізм уваги зв'язує енкодер з декодером, дозволяючи моделі фокусуватися на різних частинах вхідних даних під час генерації виходу.

Це ключовий елемент для забезпечення, що вихід правильно відображає вхідний текст.

5. Decoder RNN: Рекурентний декодер генерує цільову послідовність звуків (аудіо фреймів) один за одним. В цьому процесі використовується механізм уваги для отримання контекстуальної інформації з енкодера.
6. CBHG after Decoder: Цей модуль слід за декодером і поліпшує якість спектрограми перед кінцевим синтезом звуку.
7. Griffin-Lim Reconstruction: Останній крок полягає у перетворенні спектрограми у часовий сигнал. Гріффін-Лім алгоритм використовує ітеративний процес для перетворення спектрограми назад у звуковий хвильовий сигнал.

2.3. Трансформерні системи TTS

Системи TTS на основі трансформерів являють собою складну інтеграцію архітектури трансформерів у сферу синтезу мовлення і також є прикладом наскрізних моделей. Стає в нагоді їхня здатність обробляти складні залежності та розпаралелювати обчислення, щоб підвищити ефективність та якість генерації мовлення.

Трансформерні TTS-системи використовують архітектуру трансформерів, спочатку розроблену для задач обробки природної мови, для перетворення послідовностей тексту в послідовності аудіо або кадрів спектрограми. На відміну від традиційних рекурентних моделей, трансформери використовують механізми самоуваги для зважування важливості різних частин вхідних даних на кожному кроці конвеєра обробки, що дозволяє їм більш ефективно фіксувати складні лінгвістичні патерни. Спочатку вхідний текст перетворюється на послідовність токенів, зазвичай символів або фонем, які потім кодуються у вставки. Наступним чином ряд шарів самоуваги та нейронних мереж прямого поширення обробляють ці ембеддінги, щоб

передбачити відповідні звукові характеристики. Вихідні дані, як правило, мел-спектрограми, перетворюються в звукову форму сигналу за допомогою вокодера або методу прямого синтезу хвиль. Механізм self-attention в трансформерах обчислює представлення послідовності, розглядаючи всі частини послідовності одночасно. Функція уваги для single head може бути описана математично так: $Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$, де: Q,K,V - матриці запиту, ключа та значення, отримані з вхідних введень, d_k - розмірність ключових векторів, яка використовується для масштабування, а операція softmax гарантує, що ваги уваги дорівнюють одиниці. Щоб розширити можливості моделі фокусуватися на різних позиціях, трансформер використовує multi-head увагу: $MultiHead(Q,K,V) = Concat(head_1,\dots,head_h)$, де $head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$. Кожна head відображає різні аспекти вхідної інформації, а W^O, W_i^Q, W_i^K, W_i^V є матрицями параметрів, які вивчаються під час навчання[10].

Трансформерні TTS моделі навчаються на великих наборах даних <текст,аудіо> пар. Метою навчання зазвичай є мінімізація різниці між прогнозованою та фактичною спектрограмами, використовуючи функції втрат. Найпоширеніші варіанти функції втрат включають втрати L1 або L2 між прогнозованою та фактичною спектрограмами. Функція втрат L1 (абсолютне середнє відхилення) вимірює середнє абсолютне відхилення між кожним елементом прогнозованої спектрограми та відповідним елементом фактичної спектрограми. Це означає, що кожна різниця між прогнозованими та фактичними значеннями обчислюється без урахування знаку (тобто від'ємні та додатні відмінності обробляються однаково). Функція втрат L2 (квадратичне середнє відхилення) вимірює середнє квадратичне відхилення між кожним елементом прогнозованої та фактичної спектрограми. Квадратичне значення збільшує вплив великих відмінностей, що робить L2 більш чутливою до великих помилок у прогнозах. В процесі синтезу мови трансформер виводить мел-спектрограму

з високою роздільною здатністю як своє передбачення, фіксуючи нюанси спектральних деталей мови. Наступним кроком нейронний вокодер або традиційний метод на основі DSP(digital signal processing), такий як Гріффін-Лім, перетворює спектрограму в кінцеву форму сигналу. Трансформерні моделі, такі як BERT або GPT, стали відправною точкою для революції в області обробки природної мови завдяки їх здатності здійснювати паралельну обробку даних і велику масштабованість. Однак, хоча авторегресійні трансформери показали чудові результати у багатьох задачах, вони стикаються з обмеженнями, пов'язаними зі швидкістю та ефективністю обчислень при генерації тексту. Це призвело до зростання інтересу до неавторегресивних моделей, таких як FastSpeech.

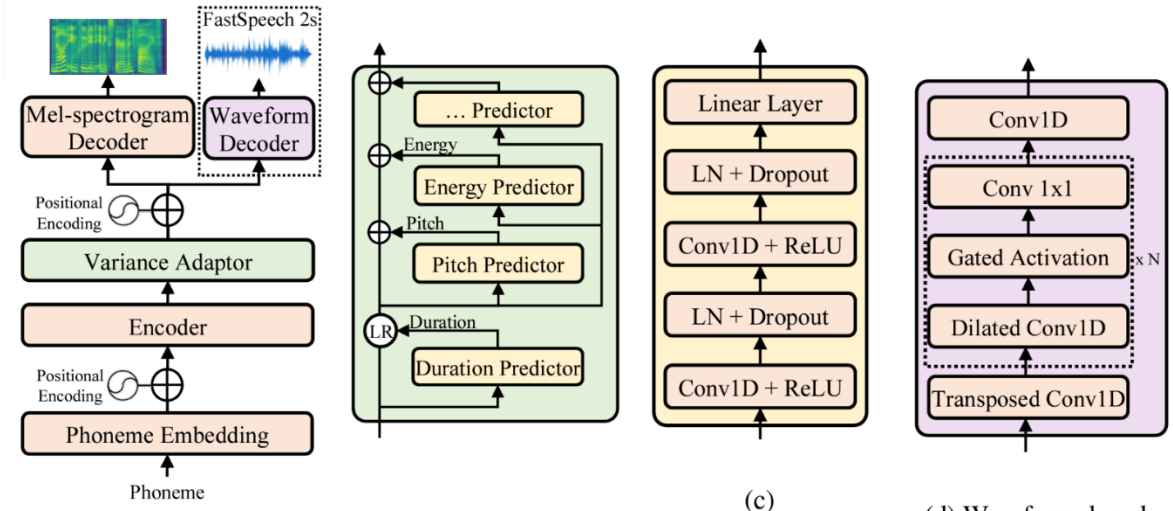
Неавторегресивні моделі вирішують ці виклики, пропонуючи швидшу генерацію тексту, оскільки вони генерують весь вихід одночасно, а не послідовно. FastSpeech, зокрема, оптимізує процес синтезу мовлення, зменшуючи залежність від довжини вхідних даних, що значно підвищує швидкість обробки. Це робить їх ідеальними для застосувань у реальному часі, де відгук системи є критичним.

Неавторегресійні моделі TTS були розроблені для усунення деяких обмежень, притаманних авторегресійним моделям, головним чином послідовної обробки, яка може призвести до уповільнення часу виведення. Неавторегресійні моделі генерують мову швидше, оскільки вони не залежать від попередніх результатів на кожному кроці генерації. Це дає змогу виконувати паралельну обробку, що значно прискорює синтез мови. На відміну від авторегресійних моделей, де прогноз кожного кроку лінійно залежить від попередніх прогнозів, неавторегресійні моделі намагаються передбачити всі частини вихідних даних одночасно. Незалежність від попередніх кроків усуває послідовну залежність, що дозволяє проводити паралельні обчислення під час виведення.

Важливим аспектом неавторегресійних моделей є узгодження між вхідним текстом і вихідним мовленням. Таке вирівнювання зазвичай передбачає безпосереднє

прогнозування тривалості фонем або сегментів, які потім розширюються відповідно до довжини вихідної послідовності. Моделі можуть використовувати додаткові мережі або механізми (наприклад, регулятори довжини) для прогнозування цих тривалостей. Неавторегресійні моделі особливо ефективні в додатках, де низька затримка є критично важливою і прийнятним є розумний компроміс між якістю і швидкістю. Це робить їх дуже придатними для синтезу мови в реальному часі.

FastSpeech, розроблена командою Microsoft, є моделлю, яка покращує Tacotron, зосереджуючись на швидкості генерації мовлення та його надійності – використовує трансформери для забезпечення ефективності та точності в генерації мовлення, вдаючись до паралельної обробки і усунення залежностей від послідовного введення вихідних даних. Це досягається завдяки використанню feed-forward мережі на основі трансформерів для одночасного прогнозування мел-спектрограм з даних тексту без необхідності чекати на попередні результати, як це було в Tacotron. Вона включає в себе як кодер для обробки вхідного тексту, так і декодер для генерації вихідної мел-спектрограми. Модель вводить адаптери дисперсії для висоти тону, тривалості та енергії, які забезпечують більш виразний і природний синтез мовлення. Ця архітектура дозволяє уникнути ітеративного процесу генерації, типового для авторегресійних моделей, що значно скорочує час, необхідний для синтезу мови.



(a) FastSpeech 2 (b) Variance adaptor (c) Duration/pitch/energy predictor (d) Waveform decoder

Це діаграма моделі FastSpeech 2, оновленої версії FastSpeech, яка є нейронною мережею для синтезу мовлення. Основні компоненти цієї моделі:

(a) FastSpeech 2: Схема загальної структури FastSpeech 2, яка включає фонемні вбудовування, енкодер, адаптер варіативності, декодер мел-спектрограми та декодер хвильових форм.

1) Phoneme Embedding: Початкове вбудовування фонем перетворює текстові фонеми на вектори.

2) Encoder: Кодує вбудовування фонем з позиційним кодуванням для підготовки до модуляції.

3) Variance Adaptor: Вносить варіативність у такі параметри, як тривалість, висота тону та енергія, щоб виходи були експресивнішими.

4) Mel-spectrogram Decoder: Перетворює вихід адаптера варіативності у мел-спектрограму.

5) Waveform Decoder: Використовується для генерації хвильової форми з мел-спектрограми.

(b) Variance Adaptor: Цей модуль використовується для внесення варіацій у динаміку мовлення, таких як висота тону, енергія та тривалість звучання фонем, що є важливим для природності мовлення.

1) Duration Predictor: Прогнозує, як довго кожна фонема має звучати.

2) Pitch Predictor: Прогнозує висоту тону для кожної фонеми.

3) Energy Predictor: Прогнозує енергію мовлення.

(c) Duration/Pitch/Energy Predictor: Модулі для прогнозування тривалості, висоти тону та енергії використовують шари згортання 1D, функцію активації ReLU, шари нормалізації (LN) та dropout для регулювання параметрів мовлення.

(d) Waveform Decoder: Модуль для декодування спектрограм у вигляд хвильової форми включає розширені згорткові шари, шлюзовану функцію активації та шари згортання 1x1[11].

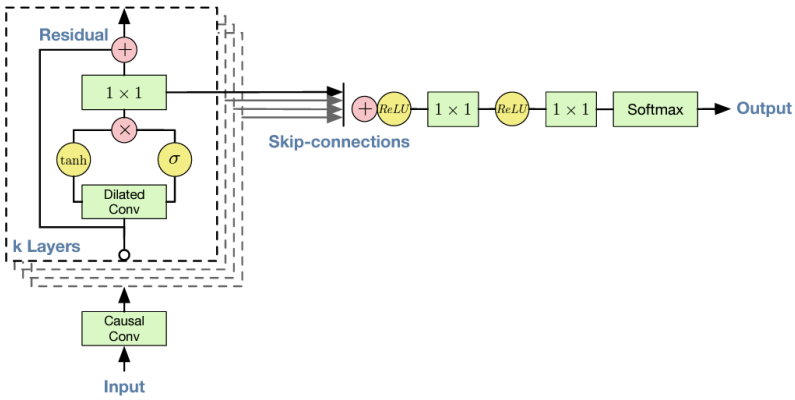
2.4. Вокодери – системи генерації звукових хвиль з спектрограм

Вокодери є важливими компонентами в процесі синтезу мовлення, вони відповідають за перетворення зображень, таких як мел-спектрограми, у звукові форми хвиль. Еволюція технології вокодерів відіграла вирішальну роль у покращенні природності та розбірливості синтезованого мовлення.

Вокодери працюють, генеруючи кінцевий аудіовихід з проміжних акустичних характеристик, які зазвичай створюються за допомогою TTS-моделі. Перші вокодери, що використовувалися для синтезу мовлення, базувалися на методах обробки сигналів, які передбачали модуляцію шумових і тональних компонентів. Сучасні вокодери, особливо ті, що використовуються з системами TTS на основі нейронних мереж, використовують передові методи машинного навчання для створення високоякісного мовлення.

Нехай $S(f,t)$ - спектрограма на частоті f і часу t, і нехай $x(t)$ - синтезований мовний сигнал. Завданням вокодера є обчислення $x(t)$ таким чином, щоб його спектрограма $S{\sim}(f,t)$ близько збігалась з $S(f,t)$.

Розглянемо WaveNet для прикладу: розроблена DeepMind, є глибинною нейронною мережею, яка генерує звук безпосередньо з аудіо даних. Ця модель є повністю згортковою та здатна генерувати реалістичне мовлення, використовуючи механізм авторегресії, де кожен вихідний зразок аудіосигналу залежить від попередніх зразків. WaveNet досягає значно вищої якості звучання, ніж традиційні методи TTS, завдяки своїй здатності моделювати складні аудіопатерни та виразність мовлення. $P(x_t \mid x_{1:t-1}, S) = \prod_{t=1}^{T} P(x_t \mid x_{1:t-1}, S)$\$, де x_t - це звуковий зразок у момент часу t, а S вхідний сигнал умовного сигналу (наприклад, мел-спектрограма).



Розглянемо архітектуру моделі:

1. Causal Convolution: Вхідні дані проходять через причинно-наслідкову згортку (causal convolution), яка забезпечує, що вихід в даному часовому кроці залежить лише від поточних і попередніх входів, а не від майбутніх. Це важливо для забезпечення послідовності під час генерації аудіо.

2. Dilated Convolution: Потім сигнал проходить через серію розширених згорткових шарів, які дозволяють мережі охоплювати більші часові відрізки без збільшення кількості параметрів або кількості обчислень. Це дає змогу моделі вловлювати довготривалі залежності в аудіо сигналах.

3. Gated Activation Units: Використання тангенсної (tanh) і сигмоїдної (σ) функцій активації з поелементним множенням дозволяє моделі регулювати інформаційний потік через мережу, що вдосконалює здатність моделі до вивчення складних шаблонів.

4. Residual and Skip Connections: Шари, які містять резидуальні (residual) та пропускні (skip) з'єднання, допомагають уникнути проблеми зникнення градієнту в глибоких мережах, а також поліпшують потік інформації під час навчання.

5. Post-Processing and Output: Після того як сигнал проходить через всі шари з розширеними згортками, він проходить додаткову обробку, яка включає кілька шарів ReLU та згорток 1x1, перш ніж пройти через softmax шар, який перетворює виходи мережі на кінцеве розподіл ймовірностей для генерації звуку[12].

2.5. Glow TTS – паралелізована модель на основі нормалізаційних потоків

Glow TTS – це клас генеративних моделей, які забезпечують гнучкий спосіб моделювання складних розподілів. Вони особливо корисні в TTS для генерації високоякісного мовлення шляхом вивчення точного представлення латентних змінних мовних даних. Потокові моделі працюють за принципом перетворення простого, відомого розподілу на складний за допомогою серії інверсійних перетворень (зокрема, з використанням нормалізуючих потоків), які дозволяють моделі відображати шум на розподіл даних (мел-спектрограму) у спосіб, який є простим для інверсії. Модель ефективно навчається "викривляти" цей простий розподіл крок за кроком до такого, що нагадує розподіл реальних мовних даних. Після навчання модель може безпосередньо генерувати мел-спектрограми з шуму, вибраного з базового розподілу. Цей процес ефективний і розпаралелюється, на відміну від авторегресійних моделей, які генерують результати послідовно.

Нормалізація потоків – це сімейство методів, які використовуються в машинному навчанні для моделювання складних розподілів ймовірностей. Як ми вже зазначили, вони працюють шляхом перетворення простого розподілу в більш складний за допомогою послідовності інверсійних функцій. Таке перетворення дозволяє як ефективно створювати вибірки (генерувати нові точки даних), так і оцінювати щільність (оцінювати, наскільки ймовірною є поява точки даних відповідно до модельованого розподілу). Кожна функція в послідовності є оберненою, що означає, що для кожної прямої операції, яка перетворює розподіл, існує відповідна зворотна операція, яка може повернутися до початкового розподілу. Формула зміни змінних в обчисленні вимагає обчислення визначника якобіанської матриці перетворення, щоб врахувати зміни об'єму (густини) при перетворенні. Властивості використаних функцій (зокрема, їхні якобіани, які легко обчислюються) роблять ці обчислення зручними. Загальне перетворення складається з декількох простіших обернених функцій. Така

композиція дозволяє моделювати дедалі складніші розподіли, коли до послідовності додається більше функцій.

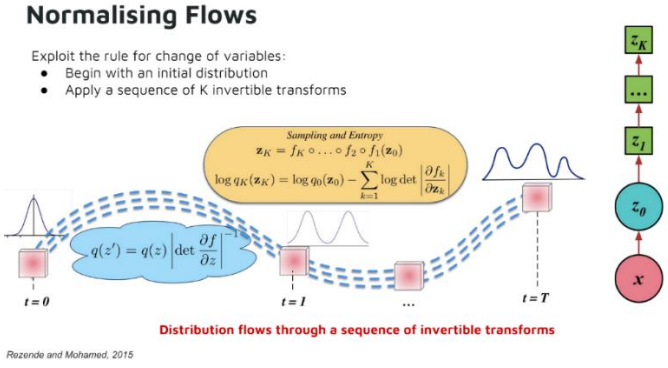
Розглянемо схему роботи нормалізаційних потоків:

1) Дано початковий розподіл z_0.

2) Послідовність перетворень $(f_1, f_2, \dots, f_K)$: застосовується послідовність інверсійних перетворень. Кожне перетворення коригує розподіл, збільшуючи його складність.

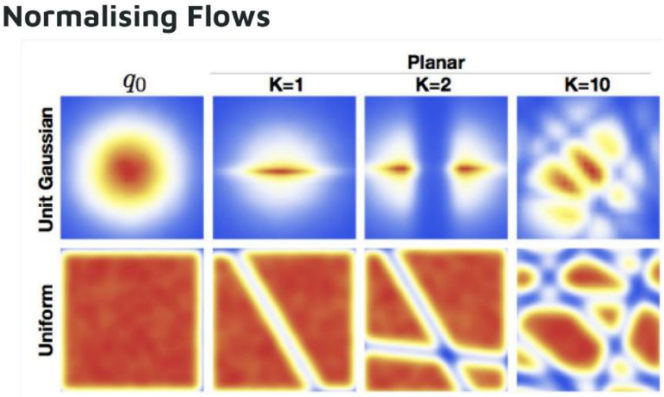
3) Final Distribution (z_K): Кінцевий результат після застосування всіх перетворень.

4) Визначник Якобіана $(\log \det \frac{\partial f_k}{\partial z_k})$: Ця частина пояснює важливість визначника Якобіана для обчислення щільності перетвореного розподілу. Він коригує щільність точок, щоб гарантувати, що перетворення відповідає правилам теорії ймовірностей.



[13]

5) Ентропія та вибірка: Показує, що при переході від початкового простого розподілу через кожне перетворення ентропія (невпорядкованість або непередбачуваність) може зростати, що дозволяє моделі генерувати більш різноманітні результати.



Нормалізовані потокові моделі можна розуміти як набір вкладених оборотних трансформацій, тобто $h_1 \cdot h_2 \cdot \ldots \cdot h_n$, де n позначає кількість шарів потоку в моделі. Щоб краще зрозуміти, що досягається цією складеною трансформацією, застосуємо логарифм до формули зміни змінних:

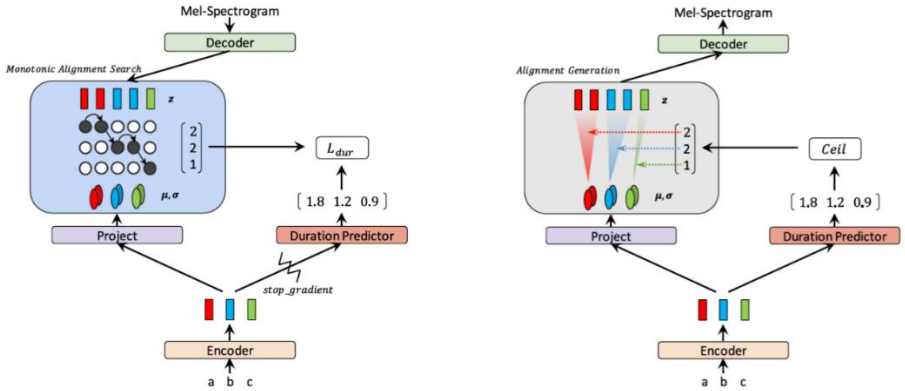
$$\log f_{Y(y)} = log\, f_X(x) - log \left| det \left(\frac{dh(x)}{dx} \right) \right|$$

Щоб спростити позначення, нехай π позначає представляє функцію густини ймовірностей (probability density function, PDF) і-ї випадкової змінної у складеній трансформації. Кожна функція h_i в моделі нормалізованих потоків приймає на вхід випадкову змінну з попереднього шару та перетворює її, змінюючи розподіл і форму її PDF. Тоді вкладена трансформація може бути виражена як

$$logf_n(x_n) = logf_{n-1}(x_{n-1}) - log \left| det \left(\frac{dh(x_{n-1})}{dx_{n-1}} \right) \right| = logf_{n-2}(x_{n-2}) -$$

$$log \left| det \left(\frac{dh(x_{n-1})}{dx_{n-1}} \right) \right| - log \left| det \left(\frac{dh(x_{n-2})}{dx_{n-2}} \right) \right| = logf_0(x_0) - \sum_{i=1}^{n} log \left| det \left(\frac{dh(x_i)}{dx_i} \right) \right|$$ [14].

Негайним наслідком цього викладу є те, що повторне застосування формули зміни змінних забезпечує прямий спосіб обчислення правдоподібності спостереження з деякого складного розподілу реальних даних f_n, виходячи з апріорного f_0 і набору оборотних трансформацій $h_1, h_2, \ldots, h_n$. Цей висновок ілюструє силу нормалізованих потоків: він пропонує прямий спосіб вимірювання правдоподібності складних, високовимірних даних.



(a) An abstract diagram of training procedure. (b) An abstract diagram of inference procedure.

Figure 1. Training and inference procedures of Glow-TTS.

Введемо поняття латентного простору - це абстрактний простір високої розмірності, в якому дані трансформуються таким чином, щоб зробити певні особливості даних більш очевидними або такими, що легко піддаються маніпулюванню за допомогою алгоритмів. В контексті генеративних моделей, латентний простір виконує кілька функцій:

1) Зменшення розмірності: Дані можуть бути стиснуті в простір меншої розмірності, який фіксує найбільш релевантні ознаки, необхідні для таких завдань, як реконструкція або прогнозування.

2) Навчання особливостей: Модель вивчає представлення в латентному просторі, які є корисними для поставленої задачі, часто виявляючи основні закономірності в даних, які не одразу помітні у вихідному просторі.

3) Розв'язка: В ідеальному випадку різні виміри в латентному просторі можуть представляти різні незалежні характеристики даних, що робить поведінку моделі більш інтерпретованою, а її вихідні дані - більш керованими.

У системах потокових, таких як Glow TTS, латентний простір може представляти різні аспекти мови, такі як тон, стиль, інтонацію або навіть емоції, які не присутні явно у вихідних фонетичних даних, але мають вирішальне значення для реалістичного синтезу мовлення.

Отже, модель перетворює вхідну мел-спектрограму x у латентну змінну z за допомогою декодера на базі потоків $f_{dec}: x \rightarrow z$ без будь-якої текстової інформації, а латентна змінна z слідує деякому ізотропному гауссівському розподілу P_Z. Потім, текстовий енкодер f_{enc}, перетворює текстову умову c у високорівневе представлення тексту h, і проектує h у статистику, μ та σ, гауссівського розподілу. Таким чином, кожен токен текстової послідовності має свій відповідний розподіл, і кожен кадр латентної змінної z_j слідує одному з цих розподілів, передбачених текстовим енкодером.

Ця відповідність між латентною змінною і розподілом є вирівнюванням A. Таким чином, якщо латентна змінна z_j слідує передбаченому розподілу і-го текстового токена $N(z_j;\ \mu_i, \sigma_i)$, то ми визначаємо $A(j) = i$. Це вирівнювання можна інтерпретувати як жорстку увагу в моделюванні seq2seq. Таким чином, маючи вирівнювання A, ми можемо обчислити точну логарифмічну правдоподібність даних наступним чином:

$\log P_X(x|c, \theta, A) = \log P_Z(z;\ c, \theta, A) + \log det\left(\frac{df_{dec}}{dx}\right)$ (1),

$\log P_Z(z;\ c, \theta, A) = \sum_{j=1}^{T_{mel}} logN(z_j;\ \mu_{A(j)}, \sigma_{A(j)})$ (2)[15].

Для вирішення проблеми пошуку вирівнювання, був введений новий алгоритм пошуку вирівнювання, монотонний алгоритм пошуку (MAS).

Algorithm 1 Monotonic Alignment Search

Input: latent representation z, the statistics of prior distribution μ , σ, the mel-spectrogram length T_{mel}, the text length T_{text}
Output: monotonic alignment A^*
Initialize $Q_{\cdot,\cdot} \leftarrow -\infty$, a cache to store the maximum log-likelihood calculations
Compute the first raw $Q_{1,j} \leftarrow \sum_{k=1}^{j} \log \mathcal{N}(z_k; \mu_1, \sigma_1)$
for $j = 2$ **to** T_{mel} **do**
  for $i = 2$ **to** $\min(j, T_{text})$ **do**
    $Q_{i,j} \leftarrow \max(Q_{i-1,j-1}, Q_{i,j-1}) + \log \mathcal{N}(z_j; \mu_i, \sigma_i)$
  end for
end for
Initialize $A^*(T_{mel}) \leftarrow T_{text}$
for $j = T_{mel} - 1$ **to** 1 **do**
  $A^*(j) \leftarrow \arg\max_{i \in \{A^*(j+1)-1, A^*(j+1)\}} Q_{i,j}$
end for

Монотонний пошук вирівнювання – це метод, який використовується для вирівнювання вхідних послідовностей з вихідними послідовностями (наприклад, аудіокадрами) у спосіб, що зберігає природний порядок вхідних даних. Ця техніка гарантує, що вирівнювання між входами та виходами не буде зворотним, тобто кожен елемент вхідної послідовності вирівнюється з одним або кількома елементами вихідної послідовності в прямому напрямку. На відміну від традиційних механізмів уваги, які динамічно зважують вхідні дані на кожному кроці декодування, монотонний пошук вирівнювання визначає фіксоване вирівнювання на основі прогнозованої тривалості, яку він обчислює під час навчання. Крім максимізації логарифмічної правдоподібності, також натренований передбачувач тривалості f_{dec}, який передбачає, скільки кадрів

мел-спектрограми вирівнюються до кожного текстового токена. Для навчання передбачувача тривалості потрібна мітка тривалості для кожного текстового токена. Ця мітка витягується з найімовірнішого вирівнювання A^*, результату MAS, хоча MAS може забезпечити погане вирівнювання на початку навчання.

З вирівнювання A^* ми можемо підрахувати, скільки мовних кадрів вирівнюються до кожного тексту за Рівнянням 3 і використовувати кількість кадрів d_j як мітку тривалості для j-го вхідного токена. Виходячи з високорівневого представлення тексту h, наш передбачувач тривалості f_{dur} навчається з втратою середньоквадратичної помилки (MSE) як у Рівнянні 4. Як і у FastSpeech, ми навчаємо f_{dur} з тривалістю d_j у логарифмічному домені. Ми також застосовуємо оператор зупинки градієнту $sg[.]$, який видаляє градієнт вхідних даних у зворотному проході, до вхідних даних передбачувача тривалості, щоб уникнути впливу на максимальну логарифмічну правдоподібність. Тому наша остаточна цільова функція є Рівняння 5. $d_j = \Sigma_{i=1}^{T_{mel}} 1_{\mathrm{A(i)=j}}, j = 1, \dots, T_{text}$ (3), $L_{dur}(\theta_{dur}) = MSE\big(f_{dur}(sg[h]), d_j;\ \theta_{dur}\big)$ (4), $\max L_{total}(\theta, \theta_{dur}) = L(\theta, A^*) + L_{dur}(\theta_{dur})$ (5).

Під час інференції, як показано на Рисунку 1b, Glow TTS передбачає статистику попередніх розподілів та тривалість кожного текстового токена з текстовим кодером f_{enc} та предиктором тривалості f_{dur}. Ми округляємо ці передбачені тривалості до цілого числа і дублюємо кожен розподіл на відповідну тривалість. Цей розширений розподіл є попереднім для Glow TTS під час виведення. Потім модель вибирає латентну змінну z з цієї попередньої вибірки і паралельно синтезує мел-спектрограму, застосовуючи зворотне перетворення декодера f_{dec}^{-1} до латентної змінної z.

2.6. Висновки до розділу 2

Розділ 2 детально розглядає різні архітектури систем синтезу мовлення, демонструючи еволюцію від традиційних методів до передових наскрізних рішень на

основі глибокого навчання. Ключові моменти розділу включають аналіз традиційних технік, таких як формантний, конкатенативний, і параметричний синтез, а також розгляд революційних наскрізних систем, які оптимізують процеси TTS за допомогою нейронних мереж.

В цьому розділі ми проробили велику роботу для розуміння ролі наскрізних нейронних архітектур в TTS. Наскрізні системи, такі як Tacotron та WaveNet, використовують глибоке навчання для обробки тексту і трансформацію його в звук, обходячи традиційні багатоетапні підходи. Це значно знижує складність системи та підвищує ефективність, роблячи технологію доступною для ширшого спектру застосунків, включаючи реальний час. Також було зазначено важливість неавторегресійних моделей, які сприяють прискоренню процесу синтезу без втрати якості звучання. Моделі, такі як FastSpeech чи Glow TTS, демонструють, що можна досягти високої ефективності при збереженні достатнього рівня природності мовлення. У цьому розділі також було висвітлено деякий порівняльний аналіз різних моделей TTS, який демонструє їхні переваги та недоліки в різних сценаріях застосування.

РОЗДІЛ 3: Адаптація моделі Glow TTS та впровадження допоміжних інструментів для навчання моделі на україномовному датасеті

3.1. Порівняльний аналіз розглянутих моделей

Feature	Tacotron 2	FastSpeech	WaveNet	Glow TTS
Model Type	Авторегресійна seq2seq	Неавторегресійна seq2seq	Авторегресійний вокодер	Flow-based неавторегресійна
Quality	Висока якість, близька до природньої	Висока, але гірша за Tacotron 2	Дуже висока (State of the art)	Висока якість, на рівні з Tacotron 2
Real-Time Factor	Низький (~1-2 RTF)	Дуже високий (~0.02 RTF)	Середній (~0.5 RTF)	Високий (~0.05 RTF)
Dataset requirements	Великий (години мовних даних)	Помірний (може використовувати дистильовані дані з моделей авторегресії)	Великий (години високоякісних мовних даних)	Помірний (подібний до FastSpeech)
Adaptiveness	Помірна (вимагає перенавчання або доопрацювання під нові голоси)	Висока (легше адаптується до нових голосів завдяки надійному вирівнюванню)	Помірна (залежить від умовних шарів)	Висока (гнучкість та адаптивність завдяки потоковому характеру)
Pros	-Природне звучання -Хороша просодія та інтонація	-Швидкий час виведення -Стійкість до різних довжин речень	-Надзвичайно природний звук - Може моделювати будь-який голос за наявності достатньої кількості даних	-Швидкий час виведення -Розпаралелювана генерація -Ефективне навчання

Cons	-Повільний час виведення -Схильний до помилок у вимові та проблем зі стабільністю	-Якість дещо поступається найкращим авторегресійним моделям -Може потребувати додаткового налаштування	-Дуже дорогі обчислення для навчання та висновків -Потребує більше даних для стабільної якості	-Потребує ретельного налаштування потокової архітектури -Менш зріла технологія, менше прикладів у виробництві
Typical Use Case	Аудіокниги, високоякісні голосові помічники	Додатки в реальному часі, Мобільні пристрої	Високоякісний синтез мови на виробництві, де достатньо обчислювальних ресурсів	Додатки, що вимагають швидкого синтезу та адаптації, наприклад, інтерактивні системи голосового зв'язку

На основі зібраних і проаналізованих даних, що висвітлені в аблиці, робимо висновок, що WaveNet забезпечує, мабуть, найкращу якість звуку серед вокодерів, створюючи дуже природну мову. Tacotron 2 генерує дуже реалістичну та виразну мову, але може бути нестабільним. FastSpeech пропонує баланс з дещо зниженою природністю для значного виграшу в швидкості. Glow TTS пропонує високоякісний синтез мови, конкурентоспроможний з провідними авторегресійними моделями, такими як Tacotron 2, але з додатковою перевагою неавторегресійної генерації.

RTF, в свою чергу, вимірює, у скільки разів повільніший або швидший синтез порівняно з відтворенням у реальному часі. FastSpeech є найшвидшим, що робить його придатним для додатків у реальному часі. Tacotron 2, як правило, повільніший, і хоча WaveNet швидший за Tacotron 2, він все одно не є оптимальним для використання в реальному часі без значного апаратного забезпечення. Завдяки своїй неавторегресійній природі та ефективному дизайну моделі на основі потоків, Glow-TTS перевершує за швидкістю, полегшуючи синтез швидше, ніж в реальному часі, що має вирішальне значення для інтерактивних додатків і додатків в реальному часі.

Щодо вимоги до набору даних, то як Tacotron 2, так і WaveNet вимагають великих і різноманітних наборів даних для хорошої роботи, особливо якщо потрібно працювати з кількома голосами і мовами. FastSpeech може використовувати знання, отримані з авторегресійної моделі, як Tacotron 2, що дозволяє йому добре працювати з меншою кількістю прямих навчальних даних. Подібно до FastSpeech, Glow TTS може ефективно працювати з наборами даних середнього розміру, особливо якщо їх оптимізувати за допомогою таких методів, як доповнення даних або перенесення навчання з інших моделей.

Архітектура FastSpeech забезпечує більш просту адаптацію до нових голосів або мов завдяки надійній обробці довжини речень і більш передбачуваній поведінці під час навчання. Потоковий підхід Glow TTS забезпечує більш просту адаптацію до нових мовних характеристик, таких як акценти або мови, що робить його універсальним у різних сценаріях застосування.

3.2. Перевірка тренованості моделі на англомовному датасеті

Основною метою цього етапу була перевірка базової здатності до навчання моделі Glow TTS від Coqui.ai[16], використовуючи добре відпрацьований англомовний набір даних. Цей крок був важливим для того, щоб переконатися, що базова архітектура здатна навчатися та генерувати мовлення ще до того, як її буде адаптовано до української мови – вся робота проводилась в Google Colab[17].

А) Методологія:

1) Вибір моделі: Обрали Glow TTS через його неавторегресійну природу та потенціал для швидкого та якісного синтезу мовлення.

2) Набір даних: Використано набір мовних даних LJ Speech, загальнодоступний набір мовних даних, що складається з 13 100 коротких аудіокліпів, на яких один диктор

читає уривки з книг, не захищених авторським правом, загальною тривалістю приблизно 24 години мовлення.

3) Налаштування навчання:

- Налаштування моделі для навчання з нуля.
- Встановили початкові гіперпараметри на основі стандартних налаштувань, рекомендованих в оригінальному дослідженні Glow TTS.
- Моніторинг показників втрат і якості звуку протягом усього процесу навчання.

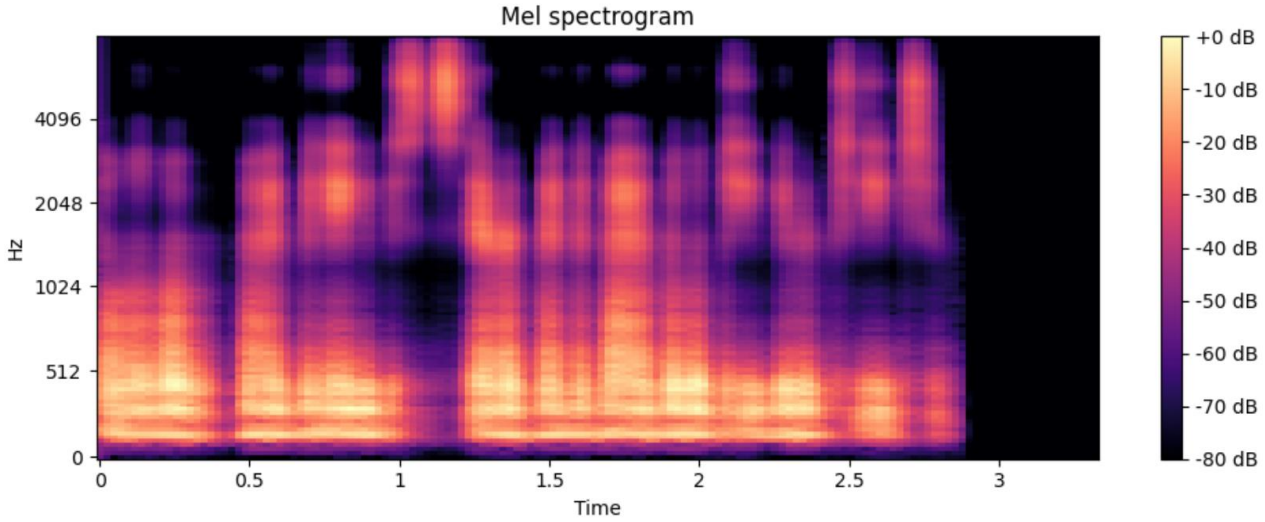
4) Навчальне середовище:

- Використовували середовище з GPU для прискорення тренувальних циклів.
- Навчання проводилося протягом 10 годин(~14k кроків), щоб зібрати попередні результати щодо поведінки моделі та якості вихідних даних.

Б) Результати:

1) Ефективність моделі:

- Після 10 годин навчання модель видавала результати, які були роботизованими і були розпізнані деякі відголоски вимови з англійським акцентом.
- Незважаючи на низьку якість результатів, модель продемонструвала базову функціональність у створенні мовлення на основі тексту, що підтверджує її здатність до навчання.



2) Набуті знання:

- Початкове навчання дало цінну інформацію про поведінку моделі за стандартних налаштувань параметрів і типової тривалості навчання.

- Виявлено потребу в налаштуванні параметрів, збільшенні тривалості навчання для покращення якості мовлення та виникло питання про додавання обробки вихідного сигналу додаткового шару вокодера

7) Виявлено проблеми:

- Роботизована якість вихідних даних підкреслила необхідність подальшого дослідження параметрів моделі та оптимізації функції втрат для підвищення чіткості та природності синтезу мовлення.

На цьому етапі було успішно встановлено базову здатність до навчання моделі Glow TTS з використанням набору даних LJ Speech. Хоча початкові результати не були оптимальними, вони підтвердили, що модель може навчатися на мовних даних, і забезпечили основу для подальших експериментів та оптимізацій, спеціально адаптованих до української мови. Цей початковий етап тестування мав вирішальне значення для визначення напряму подальших детальних досліджень більш складних конфігурацій моделі та стратегій навчання.

3.3. Підготовка набору даних

Метою цього етапу було створення повного та добре підготовленого набору даних, пристосованого для навчання моделей TTS на українській мові. Це передбачало збір та обробку значної кількості українських мовних даних, забезпечення їх правильного форматування відповідно до різних вимог TTS-моделей.

А) Методологія:

1) Збір даних:

- Використано 8 годин записаного мовлення від професійного українського актора озвучення не захищеного авторським правом та наданим у публічному доступі, що забезпечило широкий діапазон виразів та інтонацій.

- Отримано відповідні текстові транскрипції для кожного аудіозапису, щоб полегшити точне моделювання навчання.

2) Форматування набору даних:

- Підготували кілька версій набору даних для підтримки різних архітектур моделей. Це включало створення версій з фонемами та мел-спектрограмами, WAV-файлів з текстом та FLAC-файлів з фонемами.

3) Завдання попередньої обробки:

- Згенеровано мел-спектрограми з аудіофайлів, які слугуватимуть навчальними ознаками для моделей.

- Нормалізовано WAV-файли за допомогою ffmpeg для стандартизації рівнів і форматів звуку.

- Для перетворення текст у фонеми використано невелику бібліотеку, здатну конвертувати український текст у фонеми IPA.

Б) Результати:

1) Багатий набір даних:

- Успішно створено універсальний і багатий набір даних, який може задовольнити різні моделі TTS, що підвищує потенціал для всебічного навчання та оцінки.

2) Технічна готовність:

- Розроблено надійні конвеєри попередньої обробки, які забезпечили високу якість вхідних даних для навчання TTS, що має вирішальне значення для досягнення реалістичного та природнього синтезу мовлення.

3) Гнучкість у навчанні моделей:

- Наявність декількох форматів даних дозволила гнучко експериментувати з різними архітектурами TTS, що допомогло вибрати найефективнішу модель для української мови.

Ретельна підготовка різноманітного та добре обробленого набору даних полегшила безпосереднє застосування технологій TTS.

3.4. Використання Tacotron 2 та його недоліки

Метою цього етапу було дослідити можливість адаптації Tacotron 2 для синтезу українського мовлення. Дослідження було зосереджене на подоланні проблем сумісності програмного забезпечення та відсутності фонемайзера для української мови, що є важливим для фонетичного представлення, необхідного в системах TTS – вся робота проводилась в Google Colab [18].

А) Методологія:

1) Вибір моделі: Вибрали Tacotron 2 через його широке використання та високу продуктивність у генеруванні природної мови.

2) Сумісність з програмним забезпеченням:

- Виявили, що реалізація Tacotron 2 від NVIDIA не сумісна з TensorFlow 2, який підтримується Google Colab.
- Спроба модифікувати існуючу кодову базу, щоб зробити її сумісною з середовищем TensorFlow 2.

3) Труднощі з фонемізаторами:

- Дослідили існуючі інструменти фонематизації та виявили відсутність вбудованого фонематизатора eSpeak для української мови.
- Оцінили альтернативні методи та інструменти фонематизації, які можуть підтримувати українську мову.

4) Початкові адаптації:

- Спроба інтегрувати сторонню бібліотеку, здатну конвертувати український текст у фонеми ARPABET.

- Відрегулювати параметри моделі, щоб пристосувати її до особливостей української мови.

Б) Результати:

1) Технічні бар'єри:

- Зіткнулися зі значними труднощами в адаптації архітектури Tacotron 2 до TensorFlow 2, що перешкоджало прогресу в навчанні та тестуванні моделі.

- Відсутність простого фонемайзера для української мови ускладнювала підготовку навчальних даних, що впливало на здатність моделі запам'ятовувати точні вимови.

2) Точка прийняття рішення:

- Після кількох ітерацій вдосконалення моделі та вивчення варіантів фонетизації було вирішено, що Tacotron 2 не буде життєздатним без суттєвих модифікацій та додаткових ресурсів для розробки.

3) Набуті знання:

- Глибше зрозуміли залежності та складнощі, пов'язані з адаптацією складних моделей TTS до нових мов.

- Визначили потребу в більш гнучких або альтернативних фреймворках TTS, які могли б краще врахувати специфічні фонетичні та технологічні виклики.

- Визначили необхідність в пошуці або визначенні власного фонемайзера для української мови.

Цей етап виявив критичні проблеми в адаптації Tacotron 2 для української мови, насамперед через несумісність програмного забезпечення та проблеми фонетизації. Це дослідження стало цінним навчальним досвідом, підкресливши важливість гнучкості моделі та необхідність надійної мовної підтримки в технологіях TTS. Ці висновки спрямували подальший фокус на більш адаптовані та перевірені фреймворки TTS,

наприклад, повернення до Glow TTS, який запропонував більш перспективний шлях для досягнення високоякісного синтезу української мови.

3.5. Навчання моделі Glow TTS та удосконалення отриманих результатів

Метою цього етапу ми поставили натренувати модель, проаналізувати її можливості, перевірити здатність ефективно синтезувати українське мовлення, а також покращити її продуктивність за допомогою різних технічних налаштувань, включаючи розробку та інтеграцію фонетизації і використання вокодера – вся робота проводилась в Google Colab[19].

А) Методологія:

1) Оцінка моделі:

- Продовження випробувань з Glow TTS, спочатку з використанням графемного підходу для спрощення.
- Проведено тренування на наборі даних, що містить 6 годин чоловічого мовлення, в результаті якого було отримано роботизовану, але впізнавану мову з деякими невідповідностями через складні фонетичні комбінації.

2) Технічні вдосконалення:

- Внесено необхідні корективи до форматів наборів даних та конфігурацій навчання для кращого узгодження з вимогами Glow TTS.
- Інтегрували високоякісну бібліотеку генерації фонем, розроблену для української мови, у навчальний процес Glow TTS[20], і провели такий собі файн-тюнінг графемної моделі, що покращило здатність моделі справлятися з фонетичними складнощами української мови, хоча й не позбавило роботизованого звучання.

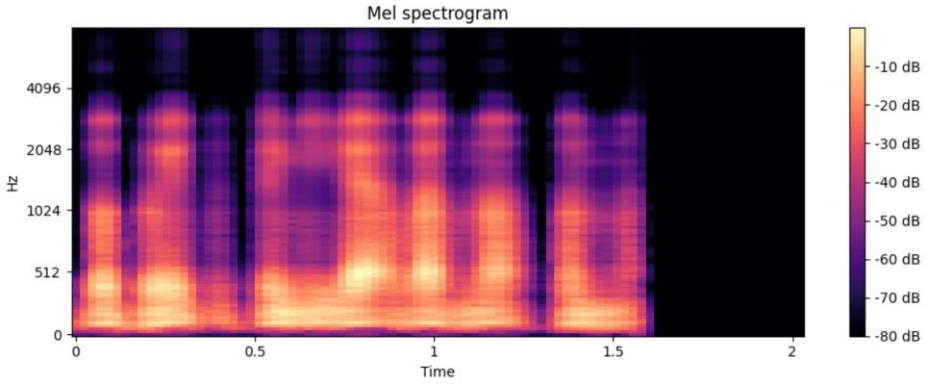
Б) Результати:

1) Вибір моделі:

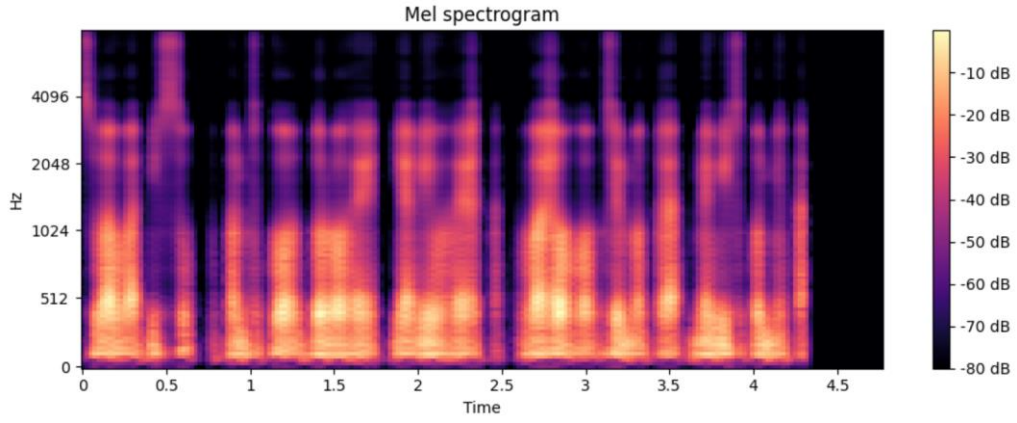
- Вирішено використовувати Glow TTS як основну модель після спостереження багатообіцяючих результатів перших експериментів з використанням графемних вхідних даних.

2) Покращена адаптація:

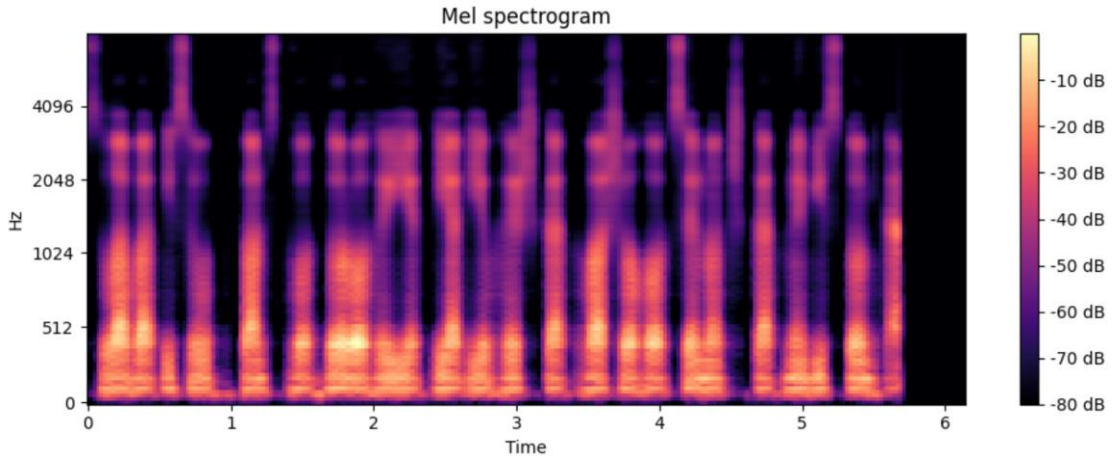
- Продемонстровано здатність Glow TTS генерувати мову з базовими звуками та словами, як видно на мел-спектрограмі, 5000 кроків з графемами:



- Продовження тренування моделі з подвоєним числом ітерацій покращило чіткість звуків, як це видно на другій мел-спектрограмі, де можна спостерігати зменшення шумів та покращення візуальної чистоти сигналу, 10000 кроків з графемами:



- Втілення фонемного підходу значно покращило якість мовлення. Модель перестала «ковтати» складні звуки, кінці слів, пом’якшення та підвищилась влучність пауз та наголосів. Третя спектрограма свідчить про покращення в розпізнаваності слів та звуків, відображаючи деталізацію та розширення частотного діапазону:



3) Надійність моделі:

- Модель продемонструвала підвищену стійкість і адаптивність до фонетичних тонкощів української мови, показавши багатообіцяючі результати для подальшого вдосконалення.

Цей етап значно просунув розробку української системи TTS. Завдяки інтеграції методів фонематизації та підбору певних налаштувань моделі, її здатність відтворювати природну та точну українську мову було значно покращено.

3.6. Висновки до розділу 3

На основі проведених досліджень та експериментів можна зробити висновок, що модель Glow TTS є зручною і перспективною для синтезу українського мовлення. Аналіз та тестування показали, що ця модель забезпечує високу якість синтезу, порівнюючи з іншими моделями, такими як WaveNet, Tacotron 2, та FastSpeech. Glow TTS поєднує переваги швидкого неавторегресійного синтезу з високою якістю звуку, що робить її ефективною для реального часу застосувань.

На етапі тренування на англомовному датасеті було підтверджено здатність Glow TTS до навчання та генерації мовлення, що забезпечило базову основу для подальшої адаптації моделі до української мови. Зібрані та підготовлені українські мовні дані створили міцний фундамент для навчання моделей TTS, що дозволило успішно

провести тренування та покращення моделі Glow TTS для синтезу українського мовлення.

Хоча адаптація Tacotron 2 виявилася проблематичною через технічні бар'єри та відсутність фонемайзерів, ці труднощі підкреслили необхідність використання більш гнучких фреймворків, таких як Glow TTS. Завдяки інтеграції спеціалізованих фонемайзерів для української мови та оптимізації параметрів навчання, вдалося значно підвищити якість синтезованого мовлення.

В результаті проведених досліджень, модель Glow TTS показала високу стійкість та адаптивність до фонетичних особливостей української мови, що робить її ефективним інструментом для синтезу природного українського мовлення.

ВИСНОВКИ

У цій роботі проведено всебічне дослідження сучасних методів синтезу мовлення та їх адаптації до української мови. Основна мета дослідження полягала у виборі та адаптації оптимальної моделі TTS (Text-to-Speech) для синтезу природного українського мовлення.

У першому розділі було розглянуто теоретичні основи синтезу мовлення, включаючи компоненти аналізу тексту та роль фонем у моделях TTS. Було обговорено методи обчислення просодії, що є критичним для створення природного мовлення.

У другому розділі було здійснено огляд сучасних моделей синтезу мовлення, включаючи традиційні архітектури, наскрізні системи та трансформерні системи TTS. Особлива увага була приділена системам генерації звукових хвиль (вокодерам) та моделі Glow TTS, яка використовує нормалізаційні потоки для паралельного синтезу мовлення. Було зроблено висновок, що Glow TTS має потенціал для швидкого та високоякісного синтезу мовлення, особливо для застосувань у реальному часі.

У практичній частині роботи було виконано кілька ключових завдань:

1) Було визначено, що WaveNet забезпечує найкращу якість звуку, Tacotron 2 генерує реалістичну мову, але є нестабільним, FastSpeech пропонує високу швидкість із дещо зниженою природністю, а Glow TTS забезпечує конкурентоспроможний синтез з перевагами неавторегресійної генерації.

2) Модель Glow TTS показала здатність до навчання та базову функціональність у створенні мовлення на основі англійських даних, що стало важливою основою для подальшої адаптації до української мови.

3) Було зібрано та підготовлено великий український мовний датасет, що включав різні формати даних (фонеми, мел-спектрограми), необхідні для різних архітектур TTS. Це створило міцну основу для навчання моделей.

4) Було виявлено значні технічні бар'єри у адаптації Tacotron 2 до української мови, що підкреслило необхідність більш гнучких фреймворків, таких як Glow TTS.
5) Було успішно натреновано модель Glow TTS з використанням українських мовних даних. Інтеграція фонематизації та оптимізація параметрів значно підвищили якість синтезованого мовлення, що підтвердило ефективність цього підходу.

За умов обмеженого часу та невеликих ресурсів для навчання, вибір моделі Glow TTS є найоптимальнішим рішенням. Вона здатна добре генералізувати ознаки, забезпечуючи високоякісний синтез без значних втрат у швидкості завдяки паралелізованій архітектурі. Також модель показала високий рівень адаптивності до фонетичних особливостей української мови. Результати цієї моделі не поступаються складним авторегресійним моделям. Ключовим елементом для досягнення точного та природного звучання мовлення є фонемний аналіз, який не варто оминати під час підготовки моделі. Інтеграція фонетизації значно підвищує якість синтезованого мовлення, зменшуючи кількість фонетичних помилок та покращуючи загальну природність синтезу.

Таким чином, результати дослідження демонструють, що модель Glow TTS є перспективним рішенням для синтезу українського мовлення, забезпечуючи високу якість звуку та ефективність роботи в реальному часі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. History of Information. (n.d.). Development of Text-to-Speech Synthesis. URL: https://historyofinformation.com/detail.php?id=620
2. Wang, R., Xu, Y., Liu, B., & Yu, Y. (2023). Leveraging Contextual Embeddings for Named Entity Recognition. // Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL). URL: https://aclanthology.org/2023.findings-eacl.65.pdf
3. Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. // arXiv. URL: https://arxiv.org/abs/1606.06871
4. Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., ... & Saurous, R. A. (2017). Tacotron: Towards End-to-End Speech Synthesis. // arXiv. URL: https://arxiv.org/abs/1609.02907
5. Lemmetty, S. (1999). Review of Speech Synthesis Technology. Master's Thesis. // Aalto University. URL: http://research.spa.aalto.fi/publications/theses/lemmetty_mst/chap5.html
6. Rabiner, L. (2012). Digital Speech Processing. Lecture Notes. URL: https://web.ece.ucsb.edu/Faculty/Rabiner/ece259/digital%20speech%20processing%20course/lectures_new/Lecture%2013_winter_2012.pdf
7. Khan, A. H. (2016). A Comparative Study of Hidden Markov Models and Deep Neural Networks in Speech Recognition. // International Journal of Computer Applications (IJCA). URL: https://www.ijcaonline.org/research/volume136/number3/khan-2016-ijca-907992.pdf
8. Popov, V., Vahdat, A., Yang, T., Gal, Y., & Bauer, C. (2021). Grad-TTS: A Diffusion Probabilistic Model for Text-to-Speech. // arXiv. URL: https://arxiv.org/pdf/2106.06863

9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All You Need. // arXiv. URL: https://arxiv.org/pdf/1703.10135v2.pdf
10. Ping, W., Peng, K., & Chen, J. (2019). ClariNet: Parallel Wave Generation in End-to-End Text-to-Speech. // arXiv. URL: https://arxiv.org/pdf/1809.08895
11. Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., & Liu, T. (2020). FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. // arXiv. URL: https://arxiv.org/pdf/2006.04558v8.pdf
12. Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., ... & Saurous, R. A. (2018). Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. // arXiv. URL: https://arxiv.org/pdf/1609.03499.pdf
13. Mohamed, S. (2020). Tutorial on Deep Generative Models. URL: https://www.shakirm.com/slides/DeepGenModelsTutorial.pdf
14. Jang, E. (2018). Normalizing Flows Tutorial. URL: https://blog.evjang.com/2018/01/nf1.html
15. Kim, J., Kong, J., & Son, J. (2020). Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search. // arXiv. URL: https://arxiv.org/pdf/2005.11129v1

Джерела для практичної частини та код на Google Colab:

16. Coqui. (n.d.). TTS: A Text-to-Speech Library. GitHub repository. URL: https://github.com/coqui-ai/TTS
17. Google Colab. (n.d.). Glow TTS for LJSpeech Training Notebook. URL: https://colab.research.google.com/drive/16W3zhWZpCYEMQ6E1HUJD9yPUMVw9v4ag?usp=sharing
18. Google Colab. (n.d.). Tacotron 2 Training Notebook. URL: https://colab.research.google.com/drive/1GYNnBSdrOqQqr1lSc7bCha7Pnbc7Za7E?usp=sharing

19. Google Colab. (n.d.). Glow-TTS Training Notebook. URL: https://colab.research.google.com/drive/1v9mbdUwgmly-WCgK4fd0hdD6sj1mns2T?usp=sharing
20. lang-uk. (n.d.). IPA Transcription for Ukrainian Language. GitHub repository. URL: https://github.com/lang-uk/ipa-uk