

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ НАВЧАННЯ
ГОЛОСОВОЇ МОДЕЛІ ЗА ДОПОМОГОЮ ШТУЧНОГО ІНТЕЛЕКТУ
RESEARCH AND SOFTWARE IMPLEMENTATION OF TEACHING
A VOICE MODEL USING AI

Виконав(ла): студент(ка) 2 курсу, групи КНТ-222м
Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Системи штучного інтелекту

П'ЯТЕРНЯ П.О.

(ПРІЗВИЩЕ та ініціали)

Керівник ГЛАДКОВА О.М.

(ПРІЗВИЩЕ та ініціали)

Рецензент ЗЕЛЕНЬОВА І.Я.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

П'ЯТЕРНІ Пилипа Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація навчання голосової моделі за допомогою штучного інтелекту

Research and Software Implementation of Teaching a Voice Model Using AI

керівник проєкту (роботи) к.т.н., ГЛАДКОВА Ольга Миколаївна

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 14 ” листопада 2023 року № 437

2. Строк подання студентом проєкту (роботи) 01 грудня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне

завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Дослідження проблеми навчання голосової моделі.

2. Глибокі неймережі. 3. Налаштування середовища для навчання. 4. Запуск процесу навчання.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ГЛАДКОВА О.М., доцент		
Нормоконтролер	БЄЛОВА А.В., асистент		

7. Дата видачі завдання “08” вересня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2-3 тижні	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів	4-5 тижні	Розділ 2
	вирішення задачі.		
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7-8 тижні	Розділ 4
6	Тестування та експериментальне дослідження	9 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів до неї.	10-11 тижні	Додатки
8	Нормоконтроль та рецензування.	12 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

_____ Пилип П'ЯТЕРНЯ
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

_____ Ольга ГЛАДКОВА
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
76 с., 1 табл., 30 рис., 3 дод., 19 джерел.

DATASET, COLAB, CONFIDENCE, DEEP LEARNING, DEEP NEURAL NETWORK, PYTHON, SVC.

Об'єкт дослідження – сучасні підходи та засоби для навчання голосової моделі.

Предмет дослідження – дослідження та навчання голосової моделі з можливістю перетворення вхідного голосу на голос моделі.

Мета роботи – розробка програмного забезпечення, яка навчає голосову модель за допомогою штучного інтелекту для можливості перетворення голосу.

Матеріали, методи та технічні засоби: штучний інтелект, глибоке навчання, датасет, віддалена лабораторія Google Colab.

Результати. Розроблена комп'ютерна система, яка перетворює голос людини в голос навченої моделі.

Висновки. Дана система є оптимальною з точки зору апаратних витрат, оскільки вона обробляє аудіо з датасету на платформі Colab у хмарному середовищі та не потребує додаткової інфраструктури для своєї роботи.

Галузь використання – індустрія розваг, аудіо-контент, системи автоматизованого виклику, синтез голосу для віртуальних асистентів та в інших областях, де необхідно відтворення голосового контенту з високою якістю та без значних витрат.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 76 pages, 1 tables, 30 figures, 3 appendixes, 19 sources.

DATASET, COLAB, CONFIDENCE, DEEP LEARNING, DEEP NEURAL NETWORK, PYTHON, SVC.

The object of the study is modern approaches and tools for teaching the voice model.

The subject of the research is the research and training of a voice model with the ability to transform the input voice into the model's voice.

The purpose of the study is to develop software that trains a voice model using artificial intelligence to be able to transform the voice.

Materials, methods and techniques: artificial intelligence, deep learning, dataset, Google Colab remote laboratory.

Results. A computer system that converts a human voice into the voice of a trained model has been developed.

Conclusions. This system is optimal in terms of hardware costs, since it processes audio from the dataset on the Colab platform in the cloud environment and does not require additional infrastructure for its operation.

Applications include the entertainment industry, audio content, automated calling systems, voice synthesis for virtual assistants, and other areas where voice content needs to be played back with high quality and at low cost.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Дослідження проблеми навчання голосової моделі	12
1.1 Порівняння аналогів	12
1.2 Огляд проблеми навчання голосової моделі	13
1.2.1 Проблеми навчання голосових моделей.....	13
1.2.2 Можливі напрями вирішення проблем	14
1.3 Складності у клонуванні голосу	14
1.4 Загальний огляд розвитку штучного інтелекту	15
1.5 Підходи і напрямки до розуміння штучного інтелекту	17
1.6 Загальні поняття нейронних мереж.....	21
1.7 Класифікація аудіофайлів	22
1.8 Технології штучного інтелекту навчання голосової моделі	23
1.9 Висновки за розділом 1	24
2 Глибокі нейромережі	25
2.1 Огляд структури глибоких нейронних мереж.....	25
2.2 Зворотне поширення помилки	25
2.3 Нейронна мережа прямого поширення.....	27
2.4 Мережа радіальних базисних функцій	28
2.5 Генеративна змагальна мережа.....	29
2.6 Згорткові нейронні мережі	31
2.7 Структура згорткової нейронної мережі	34
2.8 Згортковий шар	35
2.9 Рекурсивна нейронна мережа	36
2.10 Оптимізатори	37
2.11 Рекурентна нейронна мережа	37
2.12 Проблеми навчання нейронних мереж	38

2.13 TensorFlow.....	39
2.14 Бібліотека NumPy.....	39
2.15 Google Colab	40
2.16 WORLD vocoder	40
2.17 Librosa.....	40
2.18 Onnx	41
2.19 Torchaudio	42
2.20 Розрахунок споживання пам'яті мережею	42
2.21 Висновки за розділом 2	43
3 Налаштування середовища для навчання.....	44
3.1 Реалізація навчання.....	46
3.2 Підключення необхідних бібліотек для роботи.....	46
3.3 Загальний опис датасету.....	47
3.4 Розподіл даних для навчання та перевірки	49
3.5 Розробка глибокої згорткової нейронної мережі.....	49
3.6 Висновки за розділом 3	52
4 Запуск процесу навчання.....	53
4.1 Обчислювальний експеримент	53
4.2 Результати роботи нейронної мережі	55
4.3 Висновки за розділом 4	56
Висновки	57
Перелік джерел посилань	58
Додаток А Технічне завдання	60
Додаток Б Тести навчання голосової моделі.....	63
Додаток В Слайди презентації.....	69

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

AI	– (англ. Artificial Intelligence) – штучний інтелект;
AN	– (англ. Artificial neuron) – штучний нейрон;
CNN	– (англ. Convolutional Neural Network) – згорткова нейронна мережа;
GAN	– (англ. Generative adversarial network) – генеративні змагальні мережі;
GPU	– (англ. Graphics Processing Unit) – графічний процесор;
SVC	– (англ. Singing voice conversion) – перетворення співочого голосу;
TPU	– (англ. Tensor Processing Unit) – тензорний процесор.

ВСТУП

Штучний інтелект - це атрибут інтелектуальних систем, які виконують інноваційні функції, що традиційно вважалися прерогативою людини, а також наука і технологія створення інтелектуальних машин, особливо інтелектуальних комп'ютерних програм. Штучний інтелект пов'язаний з проблемою використання комп'ютерів для розуміння людського інтелекту, але не обов'язково обмежується біологічними методами. Сфера застосування сучасних інтелектуальних систем дуже вузька. Штучний інтелект - це здатність автоматизованих систем формалізувати та ідентифікувати якості, пов'язані з людською поведінкою. Розвиток штучного інтелекту пов'язаний з такими науками, як психологія, нейрофізіологія, математика та інформаційні технології. Штучний інтелект - дуже молодий науковий напрям, заснований американськими нейрофізіологами. Автори розробили перші нейронні моделі на основі теорій діяльності людського мозку. Сьогодні штучний інтелект залишається одним з найбільш перспективних і недосліджених напрямків у розвитку інформаційних систем і технологій управління. Поняття штучного інтелекту включає нейронні мережі, нечітку логіку, експертні системи, комп'ютери п'ятого покоління та системи моделювання мислення [1].

У сучасному середовищі випереджаючого технологічного розвитку з'являються нові наукові дисципліни в усіх галузях і сферах людської діяльності. Комп'ютерні науки в останні роки переживають стрімкий та інтенсивний підйом, виходячи з класу фундаментальних теоретичних дисциплін і значно розширюючи свої практичні застосування.

Розвиток штучних нейронних мереж тісно пов'язаний з біологією. Штучні нейрони за визначенням є спрощеними моделями біологічних нейронів. Математично це нелінійна функція з аргументом, який є лінійною комбінацією вхідних сигналів. Синапси - це зв'язки між окремими нейронами, подібні до зв'язків між природними нейронами.

Нейронні мережі створюються шляхом з'єднання штучних нейронів. Нейронні мережі мають важливі властивості, які демонструють значний потенціал і широкий спектр можливостей. Це дозволяє значно прискорити обробку інформації. Ще однією важливою особливістю нейронних мереж є здатність до навчання та узагальнення інформації. Таким чином досягається схожість з людським мозком [1].

Такі поняття, як штучні нейрони та штучні нейронні мережі є відносно новими: У 1943 році в роботі нейрофізіолога В. Маккалоха та математика Вольтера Піттса вперше було описано функціонування штучних нейронів та представлення нейромережових моделей.

В останні роки спостерігається зростання інтересу до використання нейронних мереж для вирішення різних завдань і застосування їх у різних сферах людського життя.

Використання нейронних мереж відкрило можливості для обробки інформації в областях, які раніше були сферою лише людського інтелекту. З'явилися різні можливості для створення систем, дуже близьких до інтелектуальних здібностей людини до навчання, запам'ятовування та аналізу інформації [1].

Типові завдання, які можна вирішити за допомогою нейронних мереж і нейрокомп'ютерів, включають класифікацію, автоматизацію прогнозування, автоматизацію прийняття рішень, управління, кодування і декодування інформації, розпізнавання образів.

Нейронні мережі можуть використовуватися практично у всіх галузях і сферах людської діяльності, включаючи економіку, медицину, телекомунікації, системи безпеки та обробку інформації.

У сфері систем безпеки нейронні мережі потрібні для розпізнавання облич, розпізнавання мови, розпізнавання натовпу, розпізнавання автомобільних номерів, аналізу аерокосмічних зображень, моніторингу інформаційних потоків і виявлення шахрайства [1].

Обробка інформації нейронні мережі можуть бути використані для обробки широкого спектру зображень, включаючи квитанції, підписи, відбитки пальців і розпізнавання голосу.

Нейронні мережі можна навчити класифікувати хвороби та симптоми, наприклад, у МРТ-діагностиці [2].

В аграрному секторі методи аналізу та розпізнавання зображень були розроблені та застосовані для отримання даних з відкритих супутників, таких як LSAT, і використовуються для прогнозування майбутньої врожайності на певній ділянці землі.

Першою задачею, яку навчили вирішувати нейронні мережі, була класифікація зображень.

Системи розпізнавання облич широко використовуються в різних сферах бізнесу, таких як віддалена аутентифікація клієнтів у фінансових установах і банках, онлайн-страхування та безконтактна ідентифікація в логістиці [2].

Отже, у сучасному світі майбутнє за нейронними мережами досить близьке. Дослідження нейроінформатики та нейронних мереж у різних галузях проводяться вченими в різних країнах світу. Штучні нейронні мережі можуть обробляти, аналізувати та узагальнювати інформацію подібно до того, як працює людський мозок. Нейронні мережі використовуються в різних галузях, таких як економіка, медицина, телекомунікації, системи безпеки та охорони, введення та обробка інформації. Звичайно, цей список ще не повний, але він дає уявлення про природу нейромережових технологій.

Отже, мова йде про навчання мовних моделей і перетворення вхідного мовлення на навчену мовну модель.

1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ НАВЧАННЯ ГОЛОСОВОЇ МОДЕЛІ

1.1 Порівняння аналогів

Було зроблено порівняння бібліотек для навчання голосової моделі: SVC-SO-VITS, Whisper та Vox2Vox. Ми розглянемо їхні можливості, сильні та слабкі сторони у табл. 1.1.

Таблиця 1.1 – Порівняння бібліотек

Критерій	SVC-SO-VITS	Whisper	Vox2Vox
Може генерувати різні тембри	Так	Ні	Так
Адаптація до різних мов	Так	Так	Ні
Висока точність до оригіналу	Так	Ні	Так
Низький відсоток помилок	Так	Ні	Так
Швидкий час навчання	Ні	Ні	Так

Згідно з результатами даної таблиці я вирішив обрати для використання бібліотеку SVC-SO-VITS для навчання голосової моделі.

1.2 Огляд проблеми навчання голосової моделі

Голосові моделі стали важливою складовою сучасних технологій, зокрема в областях розпізнавання мови, синтезу голосу та інтерфейсів на основі голосових команд. Проте, навчання голосових моделей є завданням, яке включає в себе численні труднощі та виклики [2].

1.2.1 Проблеми навчання голосових моделей

Нестача аудіо-даних. Однією з ключових проблем є обмежена доступність якісних та різноманітних аудіо-даних для навчання. Відсутність репрезентативних даних може призвести до неправильного навчання моделі та погіршення її продуктивності на реальних завданнях.

Великий обсяг даних. Навчання голосових моделей, особливо глибоких нейронних мереж, вимагає великої кількості даних. Збір та обробка великих обсягів аудіоданих може бути витратними і часозатратними завданнями [2].

Узагальнення та адаптація. Голосова модель, навчена на одному наборі даних чи для конкретного голосу, має обмежену здатність узагальнювати і працювати з іншими мовами чи голосами. Важливо розробляти методи узагальнення та адаптації моделей для різних умов використання.

Вартість обчислень. Величезна обчислювальна потужність, необхідна для тренування глибоких голосових моделей, може бути вартісною для багатьох дослідницьких та комерційних проектів. Ефективність алгоритмів та оптимізація ресурсів стають ключовими аспектами розв'язання цього питання.

Проблема реального часу. Деякі застосування голосових моделей вимагають миттєвої реакції, такої як голосові асистенти або системи управління [2]. Забезпечення роботи в реальному часі є складною задачею через великий обсяг обчислень та обробки аудіо-сигналів.

1.2.2 Можливі напрями вирішення проблем

Аугментація даних. Збільшення кількості доступних даних може бути досягнуто шляхом аугментації аудіо-даних, що дозволяє створювати нові приклади шляхом випадкових змін параметрів аудіосигналу [2]-[4].

Адаптивні моделі. Розробка моделей, які можуть адаптуватися до нових даних та умов, є важливою для покращення узагальнення голосових систем.

Оптимізація архітектур. Постійне вдосконалення архітектур голосових моделей, використання легших та більш ефективних моделей, може допомогти зменшити обчислювальні витрати.

Технології передового обчислення. Використання передових технологій обчислення, таких як GPU та TPU, може значно збільшити швидкість тренування та ефективність голосових моделей [2]-[4].

Вимоги до голосової моделі:

- згенерований голос має бути максимально точним до оригіналу;
- голос моделі повинен яскраво передавати емоції спікера;
- голос моделі не повинен містити сторонніх звуків.

1.3 Складності у клонуванні голосу

Штучний інтелект, зокрема технології глибокого навчання, в останні роки зазнав значного прогресу, впливаючи на різні сфери життя, включаючи обробку мови та голосові технології [2]-[4]. Однак разом з розвитком цих технологій виникає новий клас проблем, серед яких особливе місце посідає клонування голосу за допомогою штучного інтелекту.

Технологічні відділи. Однією з головних проблем є те, що доступні технології дозволяють відтворювати голос особи без її відома та згоди. Це викликає серйозні технологічні відділи, оскільки клонований голос може бути

використаний для різних маніпуляцій, включаючи шахрайство, створення фейкових аудіозаписів та інші види маніпуляцій [2]-[4].

Етичні аспекти. Клонування голосу породжує численні етичні питання. Використання скопійованого голосу для створення фальшивих аудіозаписів може призвести до розповсюдження дезінформації та впливу на громадську думку. Це може призвести до погіршення відносин між людьми та навіть впливати на політичну атмосферу.

Безпека особистості. Виробництво автентичних голосових ідентифікаторів для осіб може порушити приватність та безпеку особистості. Зловмисники можуть використовувати ці технології для створення фальшивих голосових повідомлень або викликів, шахрайської активності та інших злочинів [5].

Можливість масового використання. З покращенням технологій клонування голосу з'являється можливість масового використання цих технік. Це може призвести до поширення фальшивих голосових повідомлень, що важко виявити, що може призвести до загрози безпеці та надійності голосових комунікацій.

Засоби боротьби та регулювання. На сьогоднішній день існують обмежені засоби для виявлення та запобігання клонуванню голосу. Потрібні додаткові зусилля в розробці технологій аутентифікації, які можуть виявляти фальшиві голосові дані. Також необхідне регулювання, яке визначатиме правила та обмеження використання подібних технологій [5].

1.4 Загальний огляд розвитку штучного інтелекту

Розробка сучасних систем штучного інтелекту почалася в середині 20-го століття. Програма, яка його започаткувала, була призначена для доведення теорем про числа. Автори проекту назвали її експертною системою. Ця робота поклала початок дослідженням штучного інтелекту, які тісно пов'язані з розробкою програм, що розв'язують задачі за допомогою різних евристик. Цей

етап розвитку призвів до появи і широкого використання терміну "штучний інтелект". Фахівці зі штучного інтелекту завжди намагалися розробляти програми, які розв'язують проблеми таким чином, який вважається розумним для вирішення проблем людиною. Якщо алгоритм розв'язання задачі не відомий заздалегідь, ця задача вважається інтелектуальною. На початку розвитку штучного інтелекту були спроби змодельовати процеси мислення людини, але вони виявилися невдалими. Стало зрозуміло, що розробка універсальної програми - це гарантія. Через складність гарантування універсальності програм фокус розробки змістився на загальні методи і прийоми для конкретних програм [5]. Починаючи з 1970-х років, вчені зосередилися на розробці нових методів вираження, тобто методів вираження проблеми. Розвиток методів представлення, тобто методів формулювання проблеми таким чином, щоб її можна було легко розв'язати; розвиток методів пошуку, тобто методів, придатних для контролю за ходом розв'язання проблеми, щоб проблему можна було розв'язати в реальному часі і за допомогою реальних інструментів; на початку 80-х років "ефективність програми розв'язання проблеми залежить більше від формул, які вона використовує, і схеми міркувань, ніж від знань, якими вона володіє". Сьогодні розвиток фундаментальних досліджень у галузі штучного інтелекту включає вирішення таких завдань, як автоматична розробка програмного забезпечення, автоматичний переклад, пошук інформації, генерація документів, організація природної взаємодії користувача і комп'ютера, обробка і сприйняття природної мови і тексту, системи розпізнавання образів і візуальних образів, створення баз знань, створення експертних систем [5].

Штучний інтелект (ШІ) - це наука і технологія створення інтелектуальних машин (машин, керованих програмним забезпеченням), здатних керувати певними функціями розумової діяльності людини (наприклад, вибирати і реалізовувати оптимальні рішення на основі раціонального аналізу минулого досвіду і зовнішніх впливів). Інше визначення штучного інтелекту - це наукова дисципліна, в якій ставиться і вирішується

завдання апаратного або програмного моделювання. Штучний інтелект дозволяє комп'ютерам вчитися на власному досвіді. Тобто вони можуть обробляти великі обсяги даних, знаходити певні закономірності, адаптуватися до заданих параметрів і виконувати багато завдань, які раніше були під силу лише людині [6].

1.5 Підходи і напрямки до розуміння штучного інтелекту

Штучний інтелект - це мистецтво створення інтелектуальних машин, особливо інтелектуальних комп'ютерних програм. Штучний інтелект пов'язаний зі схожим завданням використання комп'ютерів для розуміння людського інтелекту, але обов'язково обмежується біологічно правдоподібними методами. Точного визначення цієї науки не існує, оскільки філософія не вирішила питання про природу і статус людського інтелекту. Не існує також точних критеріїв набуття комп'ютером "інтелекту", хоча було висунуто низку гіпотез щодо штучного інтелекту, таких як тест Тюрінга та гіпотеза Ньюелла-Саймона. Сьогодні існує багато підходів до розуміння завдань штучного інтелекту та побудови різних інтелектуальних систем [6].

Інтелект або розум - це якість психіки, що складається зі здатності адаптуватися до нових ситуацій, вчитися і запам'ятовувати досвід, розуміти і застосовувати абстрактні поняття, а також використовувати свої знання для контролю над навколишнім середовищем. Інтелект - це загальна здатність вчитися і вирішувати проблеми, що поєднує в собі всі когнітивні здібності людини, такі як відчуття, сприйняття, пам'ять, уява, мислення і мрії.

На початку 1980-х років вчені-комп'ютерники Барр і Фейгенбаум запропонували наступне визначення штучного інтелекту: "Штучний інтелект - це галузь інформатики, що займається розробкою інтелектуальних комп'ютерних систем, які можуть розуміти мову, навчатися, міркувати і вирішувати проблеми, які ми традиційно асоціюємо з людським розумом. Він

займається розробкою систем з можливостями, які ми традиційно асоціюємо з людським розумом.

Така класифікація розрізняє два підходи до розвитку штучного інтелекту [6]:

- низхідний, семіотичний - створення символічних систем, які моделюють високорівневі психічні процеси (мислення, судження, мову, емоції, творчість тощо);

- висхідний, біологічний – вивчення нейронних мереж та еволюційних обчислень для моделювання інтелектуальної поведінки на основі більш менших "не інтелектуальних" елементів.

Ця наука пов'язана з психологією, нейрофізіологією та трансгуманізмом. Вона використовує математичні інструменти, подібні до комп'ютерних наук. Особливо важливими є філософія та робототехніка. Штучний інтелект - дуже молода галузь досліджень, заснована в 1956 році. Його історична траєкторія нагадує синусоїду. Сьогодні його розвиток мінімальний, оскільки поступається інтеграції досягнень в інших сферах науки, промисловості, бізнесу і навіть повсякденного життя [6].

Існують різні підходи до розробки систем штучного інтелекту. Наразі можна виділити чотири підходи: логічний, структурний, еволюційний та імітаційний.

Логічний підхід базується на булевій алгебрі. Вона відома всім, хто цікавиться програмуванням. Наука булева алгебра розвинулася у вигляді числення предикатів, яке розширюється за рахунок введення таких символів об'єктів та відношень між ними. Кожна така машина також має блок формування мети, і система міркувань намагається довести цю мету у вигляді теореми. Після того, як мета досягнута, набір використаних правил дає ланцюжок дій, необхідних для реалізації мети (такі системи також називають експертними). Потужність такої системи залежить від можливостей

генератора цілей і можливості довести теорему . Нечітка логіка - це новий підхід для досягнення кращої виразності. Основна відмінність полягає в тому, що істинність або хибність твердження може приймати проміжні значення, такі як "не знаю", а також "так/ні". Цей підхід ближчий до людського мислення, оскільки на нього рідко відповідають "так" або "ні" [6].

Структурні підходи стосуються спроб створення штучного інтелекту шляхом моделювання структури людського мозку. Однією з перших таких спроб був перцептрон Френка Розенблата. Основною моделюючою структурною одиницею перцептрону є нейрон. Пізніше з'явилися інші моделі, більш відомі як нейронні мережі та їхні реалізації - нейрокомп'ютери. Ці моделі відрізняються структурою окремих нейронів, топологією зв'язків між ними та алгоритмами навчання. Наразі найбільш відомими є нейронні мережі зі зворотним поширенням, Кохонні мережі, мережі Хопфілда та стохастичні нейронні мережі [7]. Відмінності між логічним і структурним підходами не такі принципові, як може здатися. Алгоритми, які спрощують і вербалізують нейронні мережі, перетворюють моделі структурного підходу в явні логічні моделі. З іншого боку, у 1943 році Маккалох і Піттс показали, що нейронні мережі можуть реалізувати будь-яку функцію логічної алгебри.

При побудові систем штучного інтелекту за допомогою цього методу основна увага приділяється побудові початкової моделі та правилам, необхідним для того, щоб ця модель змінювалася (еволюціонувала). Також моделі можуть бути побудовані по-різному, це можуть бути НМ, набір логічних правил або інші моделі. Потім, при включенні комп'ютера, на основі перевірки моделі вибирається найбільш підходяща модель і на її основі створюється нова модель за різними правилами. З еволюційних алгоритмів класичним є генетичний алгоритм.

Імітаційний підхід є класичним у кібернетиці, і одним з його основних понять є чорний ящик. Чорним ящиком називається об'єкт, поведінка якого моделюється. Неважливо, які моделі в ньому знаходяться і як вони функціонують, але важливо, щоб наші моделі поводитися однаково в схожих

ситуаціях. Це здатність копіювати те, що зробили інші, не розбиваючи це на базові операції або формальні визначення поведінки. Ця особливість часто економить багато часу, особливо в перші дні існування об'єкта [7].

Гібридні інтелектуальні системи намагаються поєднати ці дисципліни. Правила експертного висновку генеруються нейронними мережами, а згенеровані правила отримуються шляхом статистичного навчання. Новий підхід, також відомий як підсилення інтелекту, розглядає досягнення штучного інтелекту в процесі еволюційного розвитку як постійні ефекти доповнення людського інтелекту за допомогою технологій.

Основна мета досліджень ІІІ - розкрити таємниці та секрети мислення і побудувати моделі людського мозку. Фундаментальний потенціал для моделювання інтелектуальних процесів впливає з фундаментальних епістемологічних наслідків кібернетики. Це означає, що будь-яка функція мозку, будь-яка розумова діяльність, описана мовою з точним значенням за допомогою кінцевого числа слів, в принципі може бути перенесена на електронний цифровий комп'ютер. Сучасне наукове розуміння природи мозку, принаймні в суто інформаційних аспектах, дозволяє припустити, що найважливіші закономірності роботи мозку визначаються кінцевою (можливо, дуже великою) системою правил [8].

Проблема штучного інтелекту досить тісно пов'язана з потребами застосування, і хоча не існує єдиної загальної прикладної проблеми, яка чітко визначає розвиток теорії, є багато часткових і вузько визначених проблем. Таким чином, проблема ІІІ - це комплекс завдань, що характеризуються різним ступенем загальності, абстрактності, складності та розробленості, кожна з яких має свої фундаментальні та практичні проблеми. До них відносяться розпізнавання образів, навчання і самонавчання, евристичне програмування, побудова загальної теорії систем, що самоорганізуються, і побудова фізичних моделей нейронів. У всіх цих сферах досягнуто важливих практичних і теоретичних результатів, і інтенсивні дослідження все ще тривають [8].

1.6 Загальні поняття нейронних мереж

Штучні нейронні мережі (ШНМ) є програмною реалізацією нейронної структури мозку [8]. Складна біологічна структура мозку не описана, але достатньо знати, що мозок містить нейрони - своєрідні органічні перемикачі. Нейрони можуть змінювати тип сигналу, що передається електричними або хімічними сигналами. Нейронна мережа людського мозку - це величезна взаємопов'язана система нейронів, в якій сигнали, що передаються від одного нейрона, передаються тисячам інших нейронів. Навчання відбувається, коли кілька нейронних зв'язків активуються багаторазово. Це збільшує ймовірність того, що відповідна вхідна інформація (сигнал) буде використана для отримання бажаного результату. Для цього типу навчання використовується зворотний зв'язок, і коли отримано правильний результат, нейронні зв'язки, які дають цей результат, стають щільнішими.

Штучні нейронні мережі імітують поведінку мозку в більш простій формі. Нейронні мережі можна навчати контрольованим і неконтрольованим способами. У контрольованих ШНМ мережа навчається шляхом передачі зразків відповідної вхідної та вихідної інформації. Наприклад, у випадку фільтра спаму у поштової скриньці вхідною інформацією є список слів, які часто зустрічаються у спам-повідомленнях, а вихідною - класифікація повідомлення (спам чи ні). Цей тип навчання додає ваги до зв'язків ШНМ, про які йтиметься нижче [8].

Некероване навчання в ШНМ намагається "змусити", щоб ШНМ "зрозуміла" структуру переданої вхідної інформації "самостійно".

Розвиток і глобалізація Інтернету призвели до появи великої кількості інформації, яка фізично не може бути оброблена людиною. Нейронні мережі використовуються в наступних ситуаціях:

- класифікація та аналіз даних за заданими параметрами;
- створення аналітичних прогнозів, на основі вхідної інформацією;
- порівняння та розпізнавання однакових даних.

Синапсоїдні функції краще підходять для опису механіки системного аналізу, оскільки вони більш чіткі в обмеженому діапазоні вхідної інформації.

Обчислювальні алгоритми:

- дані надсилаються на нейрон;
- перевіряється їх вага;
- результат обчислень надсилаються до наступного нейрона;
- процес відбувається знову.

Кількість обчислень визначається заданням кількості шарів. Сучасні нейронні мережі мають десятки, іноді сотні обчислювальних шарів. У книгах з програмування наводяться приклади коду на Java, які показують, що технологія розвивається не тільки в області десктопних додатків, але і для мобільних платформ. Це свідчить про ефективність нейронних мереж [8].

1.7 Класифікація аудіофайлів

Класифікація аудіоданих передбачає аналіз вхідного набору аудіофайлів, який вже був класифікований за певними категоріями. У сфері машинного навчання, це завдання передбачає розпізнавання певних акустичних ознак у зв'язку та визначення, до якої категорії вони відносяться. Цей процес може бути використаний для класифікації музичних жанрів, розпізнавання мови, аналізу звукових ефектів тощо.

Для розв'язання завдань класифікації аудіоданих застосовують схожий підхід. Спочатку формується тренувальний набір, що складається з аудіофайлів, які вже мають визначені класи [8]. Потім використовується алгоритм навчання, який аналізує акустичні особливості цих класів на основі тренувального набору.

Оцінка якості класифікатора зазвичай проводиться за допомогою тестового набору аудіофайлів, де визначається, наскільки точно алгоритм класифікує нові аудіозаписи за їхніми категоріями.

Найпоширенішим методом для класифікації аудіоданих є використання рекурентних нейронних мереж (RNN) або довгих короткочасних нейронних мереж (LSTM), які здатні аналізувати послідовності звукових даних та враховувати їхні контексти.

Так само, як і в класифікації зображень, процес навчання включає в себе оптимізацію параметрів моделі для досягнення оптимальної точності класифікації [8].

1.8 Технології штучного інтелекту навчання голосової моделі

Штучний інтелект (ШІ) є однією з найбільш перспективних галузей науки, яка значно розширює можливості автоматизації та інтелектуалізації систем. Одним з інноваційних напрямків в галузі ШІ є розвиток технологій навчання голосових моделей. Голосові моделі використовуються для розпізнавання та синтезу мови, а їх навчання за допомогою штучного інтелекту відкриває нові можливості для покращення комунікації між людьми та комп'ютерами [8].

Основні принципи голосових моделей. Голосові моделі базуються на принципах обробки мовлення та звуку. Однією з ключових складових є мовленнєва акустика, яка дозволяє аналізувати звукові хвилі та виділяти характеристики, що відповідають конкретним фонемам та словам. Технології глибокого навчання, такі як нейронні мережі, використовуються для створення голосових моделей, які здатні адаптуватися до різних голосових варіацій та мовних особливостей.

Використання нейронних мереж у навчанні голосових моделей. Нейронні мережі є ключовим інструментом у розвитку голосових технологій. Вони дозволяють автоматизовано визначати та вивчати складні патерни у звукових даних. Рекурентні та сверхтонкі нейронні мережі здатні враховувати контекст та структуру мови, що робить їх ефективними у вирішенні завдань голосового розпізнавання та синтезу мови.

Адаптація до індивідуальних особливостей голосу. Ще однією важливою характеристикою технологій навчання голосових моделей є їх здатність адаптуватися до індивідуальних особливостей голосу користувачів. Це включає в себе навчання моделей розпізнавання голосу певного користувача та адаптацію синтезу мови до його особливостей [8].

Виклики та перспективи. Незважаючи на значні досягнення в цій галузі, існують виклики, такі як покращення точності розпізнавання в різних мовних середовищах та робота з шумами. Зокрема, важливо розробляти технології, які забезпечать надійність та точність голосового інтерфейсу у реальних умовах.

1.9 Висновки за розділом 1

Перший розділ направлений на ознайомлення з проблеми навчання голосової моделі за допомогою штучного інтелекту, а також були освітлені етичні моменти та моменти безпеки щодо клонування голосу за допомогою штучного інтелекту є серйозною проблемою, яка вимагає уваги від науковців, розробників технологій та регуляторів. Захист приватності, безпека особистості та боротьба зі зловживанням цією технологією стають ключовими завданнями для подальшого розвитку та впровадження голосових технологій.

Технології навчання голосових моделей відкривають нові горизонти для розвитку голосових інтерфейсів та систем. Застосування штучного інтелекту у цій галузі сприяє покращенню якості голосового взаємодії та розширенню можливостей автоматизації наших щоденних завдань. З урахуванням постійного розвитку цих технологій можна очікувати подальших інновацій та вдосконалень у сфері голосового інтерфейсу.

2 ГЛИБОКІ НЕЙРОМЕРЕЖІ

Глибинні нейронні мережі набули великої популярності через свою здатність вирішувати складні завдання. Існує сім основних видів глибинних нейронних мереж: нейронна мережа прямого поширення, мережа радіальних базисних функцій, самоорганізаційна карта Кохонена, рекурсивна нейронна мережа, рекурентна нейронна мережа, згорткова нейронна мережа, модулярна нейронна мережа.

Як і у випадку зі штучними нейронними мережами, навчання глибоких нейронних мереж у примітивний і наївний спосіб може викликати труднощі. Дві основні проблеми - це перенавчання та тривалий час обчислень [9].

2.1 Огляд структури глибоких нейронних мереж

Глибокі нейронні мережі - це штучні нейронні мережі з декількома прихованими шарами між вхідним і вихідним шарами. Як і звичайні нейронні мережі, глибинні мережі можуть моделювати складні нелінійні зв'язки. Архітектури глибоких нейронних мереж створюють складові моделі, де об'єкти представлені у вигляді багаторівневої конфігурації примітивів зображень, наприклад, для виявлення та аналізу об'єктів. Глибинні нейронні мережі зазвичай розробляються як мережі прямого поширення, але рекурентні нейронні мережі також дуже успішно використовуються тут [9]. Згорткові нейронні мережі використовуються в комп'ютерному зорі. Згорткові нейронні мережі також застосовуються в акустичному моделюванні для автоматичного розпізнавання мови з успіхом у порівнянні з традиційними моделями.

2.2 Зворотне поширення помилки

Глибока нейронна мережа навчається звичайним алгоритмом backpropagation.

Backpropagation – (Метод зворотного поширення помилки) - це ітеративний алгоритм, що є модифікацією методу градієнтного спуску, та використовується з метою мінімізації помилки роботи багатошарової нейронної мережі [9].

Згідно з різними джерелами, основи безперервного розповсюдження були отримані в 1960 році Генрі Келлі (Henry J. Kelly) і в 1961 році Артуром Брайсоном (Arthur E. Bryson) в контексті теорії управління із застосуванням принципів динамічного програмування; в 1962 році було опубліковано більш просте міркування, засноване виключно на ланцюгових законах. У 1970 році Сеппо Ліннайрмаа представив загальний метод автоматичного диференціювання роз'єднаних мереж з використанням вкладеного диференціювання. Цей метод відповідає сучасній версії методу зворотного поширення і є ефективним навіть при слабкому зв'язку мережі; у 1973 році Стюарт Дрейфус використав алгоритм зворотного поширення для адаптації параметрів регулятора пропорційно градієнту помилки; у 1974 році Пол Вербос згадав про можливість застосування цього принципу до штучних нейронних мереж, а у 1982 році метод вкладеного диференціювання Ліннайрмаа був застосований до нейронних мереж, які широко використовуються сьогодні; у 1986 році Девід Е. Румелхарт, Джеффри Ел. Румельхарт, Джеффри Е. Хінтон, Рональд Дж. Вільямс показали за допомогою комп'ютерних експериментів, що метод може створювати корисне внутрішнє представлення вхідних даних у прихованому шарі нейронної мережі. 1993 року Ерік А. Ванг виграв міжнародний конкурс з розпізнавання образів за допомогою зворотного розповсюдження. Він став першим переможцем [9].

Домінуючим методом навчання глибоких нейронних мереж є залежність до помилок навчання (наприклад, зворотне поширення з градієнтним спуском), оскільки він простий у реалізації і має тенденцію збігатися до локального оптимуму краще, ніж інші методи навчання. Однак ці методи можуть бути обчислювально дорогими (тобто вимагати більше часу на навчання і пам'яті), особливо для глибоких нейронних мереж. При навчанні

глибоких нейронних мереж необхідно враховувати ряд параметрів навчання, таких як розмір (кількість шарів, кількість одиниць на шар), швидкість навчання і початкові ваги. Покриття всього простору параметрів для пошуку оптимальних параметрів може бути неможливим через брак часу та обчислювальних ресурсів. Різні підходи, такі як використання міні-партій, виявилися ефективними для прискорення обчислень; велика пропускну здатність графічних процесорів забезпечила значні переваги для навчання через векторно-матричну природу обчислень, які вони вимагають і добре виконують.

2.3 Нейронна мережа прямого поширення

Пряма нейронна мережа - це тип нейронної мережі, в якій сигнал поширюється в одному напрямку, нейрони починаються з вхідного шару і поширюються від прихованого шару до вихідного, а вихідні нейрони отримують результати обробки сигналу. У цьому типі мережі немає зворотного зв'язку. Протилежним типом нейронної мережі зі зворотним зв'язком є рекурентна нейронна мережа. Прикладом нейронної мережі зі зворотним зв'язком є персептрон Розенблата, який стоїть біля витоків нейронних мереж зі зворотним зв'язком [9]. Основною математичною задачею, з якою можуть впоратися персептрони, є лінійне розділення випадкових нелінійних множин, що називається лінійною сепарабельністю. Терміни "персептрон", "багатошаровий персептрон" і "нейронна мережа прямого поширення" часто використовуються в літературі як синоніми. Всі персептрони є нейронними мережами прямого поширення і відрізняються головним чином кількістю шарів, функціями активації та методами навчання.

2.4 Мережа радіальних базисних функцій

Для математичного моделювання використовуються мережі радіальних базисних функцій - це штучна нейронна мережа з радіальними базисними функціями як функціями активації. Вихід мережі є лінійною комбінацією вхідних радіальних базисних функцій і параметрів нейронів. Мережі з радіально-базисними функціями мають багато застосувань, включаючи апроксимацію функцій, прогнозування часових рядів, класифікацію та управління системами [9].

Мережі з радіальними базисними функціями мають ряд переваг над багатошаровими мережами прямого поширення. По-перше, довільна нелінійна функція моделюється лише одним проміжним шаром, тому розробнику не потрібно визначати кількість шарів. По-друге, параметри лінійної комбінації вихідного шару можуть бути повністю оптимізовані за допомогою відомих методів лінійної оптимізації, які є швидкими і не страждають від локальних мінімумів, які є проблематичними для навчання зі зворотним поширенням. Тому мережі з радіально-базисними функціями можна навчати дуже швидко, набагато швидше, ніж за допомогою алгоритмів зворотного поширення.

Недоліки радіально-базисних мереж: ці мережі не мають екстраполяційних властивостей і стають досить громіздкими, коли вхідний вектор великий.

Найпростіша форма нейронної мережі радіально-базисної функції містить три шари: звичайний вхідний шар, який розподіляє дані вибірки для першого вагового шару, шар прихованих нейронів з радіально-симетричними функціями активації, вихідний шар, призначений для зберігання окремого еталонного вектора у вигляді вагового вектора $w_j(h)$ для кожного j -го [9].

2.5 Генеративна змагальна мережа

Генеративні змагальні мережі (Generative Adversarial Networks, GAN) - це тип архітектури нейронної мережі, яка може генерувати нові дані, що відповідають вивченим шаблонам. GAN можна використовувати для генерації зображень, таких як мова і обличчя, а також для перетворення тексту в зображення. GANs знаходяться в авангарді багатьох передових систем штучного інтелекту, додатків і досліджень, оскільки вони можуть генерувати абсолютно нові дані.

GAN є прикладом генеративних моделей. Більшість моделей штучного інтелекту можна розділити на дві категорії: керовані та некеровані моделі. Моделі керованого навчання часто використовують для розрізнення різних категорій вхідних даних для класифікації. Моделі керованого навчання вивчають розподіл набору даних, щоб з цього вивченого розподілу можна було взяти зразки для генерації нових даних [10]. GAN - це фактично комбінація двох різних мереж: GAN складається з генеративної моделі та дискримінативної моделі, які також називають генератором і дискримінатором.

Архітектура GAN Генеративні змагальні мережі утворюються шляхом поєднання моделі генератора і моделі дискримінатора. Модель-генератор генерує нові вибірки даних на основі закономірностей, які модель вивчила з навчальних даних. Завданням моделі-дискримінатора є аналіз зображення (за умови, що вона була навчена на зображенні) і визначення того, чи є зображення створеним/підробленим або справжнім, рисунок 2.1.

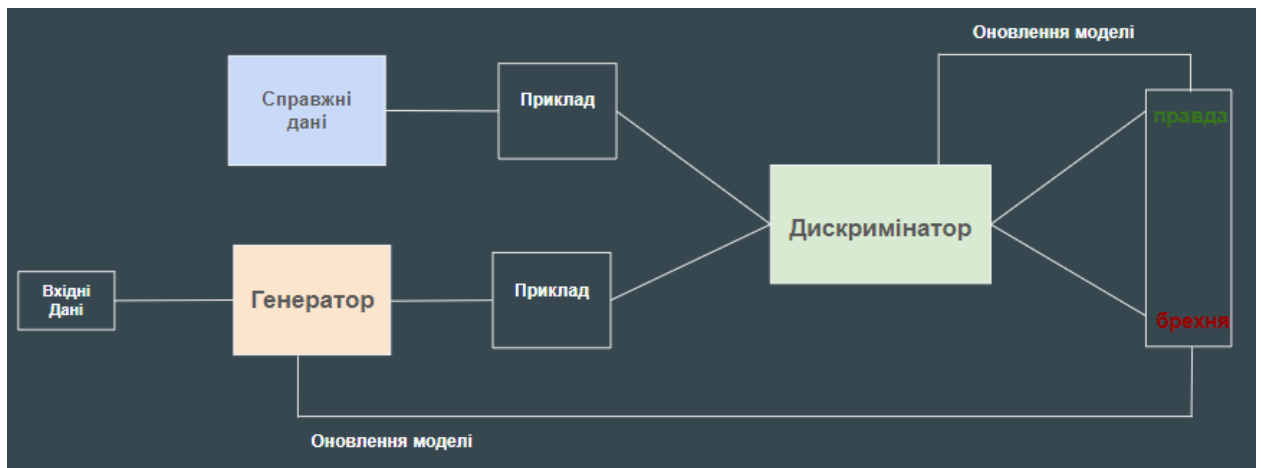


Рисунок 2.1 – Алгоритм нейронної мережі GAN

У теорії ігор дві навчені моделі перебувають у конфлікті. Мета моделі-генератора - створити зображення, яке введе в оману її опонента, модель-дискримінатор. Завдання дискримінативної моделі - перемогти свого опонента, модель-генератор, і зафіксувати хибні образи, створені генератором. Протистояння моделей призводить до гонки озброєнь, в якій обидві моделі вдосконалюються. Дискримінатор отримує зворотний зв'язок про те, які зображення є справжніми, а які створені генератором, в той час як генератор отримує інформацію про те, які зображення дискримінатор позначає як фальшиві. Обидві моделі вдосконалюються під час навчання для того, щоб навчити генеративну модель, яка може виробляти неправдиві дані, які майже не відрізняються від реальних даних [10].

Дискримінантні моделі живляться вибірками з усієї навчальної області, що складається як з реальних, так і з генерованих зразків даних. Справжні дані містяться в навчальному наборі даних, тоді як хибні дані генеруються генеративною моделлю. Процес навчання дискримінантної моделі точно такий самий, як і базове навчання моделі бінарної класифікації.

Генеративна та дискримінантна моделі грають між собою в гру з нульовою сумою. Гра з нульовою сумою - це гра, в якій одна сторона виграє за рахунок іншої (сума дій обох сторін дорівнює нулю). Якщо дискримінативна модель успішно розрізняє справжні та підроблені приклади,

параметри ідентифікації залишаються незмінними. Однак, якщо модель не може розрізнити реальні та фальшиві приклади, параметри моделі суттєво змінюються. Протилежне стосується генеративної моделі, яка карається, якщо їй не вдасться обдурити дискримінатор (параметри оновлюються), але в іншому випадку її параметри не змінюються (або заохочуються).

В ідеалі, генератор може покращити свою роботу до такої міри, що дискримінатор не зможе відрізнити фальшиві зображення від справжніх. Іншими словами, згенероване зображення стає не відрізнити від справжнього. На практиці GAN зазвичай не досягають цієї точки. Однак, навіть якщо генеративна модель не створює ідеально схожих зображень, GAN корисні для багатьох завдань, в яких вони використовуються [10].

Такий підхід використовується при навчанні голосової моделі SVC-SO-VITS.

2.6 Згорткові нейронні мережі

Convolutional Neural Network – (Згорткова нейронна мережа) – це нейронна мережа прямого поширення спеціального виду, що працює на основі фільтрів, які займаються розпізнаванням характеристик певних класів об'єктів на зображенні: прямих ліній, кривих певного типу тощо [11].

Згорткова нейронна мережа – основний інструмент для навчання голосової моделі, на основі існуючого датасету. Dataset – це оброблена та структурована певним чином інформація, необхідна для навчання нейронної мережі. Існує велика кількість варіантів застосування штучних згорткових нейронних мереж, такі як глибокі мережі, Region-згорткові мережі, Fully Convolutional Neural Networks та Mask R-згорткові нейронні мережі та інші.

Згорткові нейромережі потребують досить мало попередньої обробки. Створена мережа навчається використовуючи датасет із аудіо файлів. Згорткові нейронні мережі мають шари входу та виходу, а також певну кількість прихованих шарів. Приховані шари згорткової нейронної мережі за

допомогою спеціального процесу, який називається згортка, можна одночасно зменшити кількість інформації, що зберігається в інформаційній пам'яті [11].

Кодер вмісту SVC-SO-VITS являє собою згорткову нейронну мережу (CNN), яка навчається на наборі даних з вихідних аудіофайлів та їх текстових транскрипцій. Згорткові нейронні мережі є одними з найпоширеніших нейронних мереж для обробки зображень та звуку. Вони ефективні для виявлення локальних шаблонів у даних, таких як частоти та візерунки в спектрі звуку рисунку 2.2.

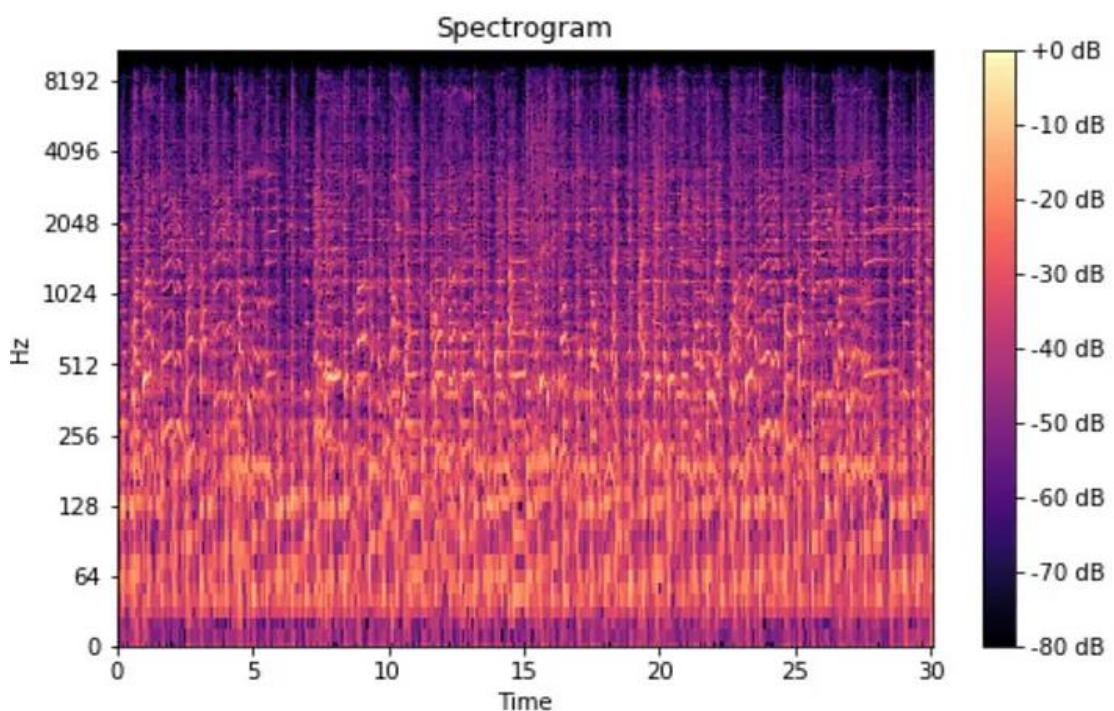


Рисунок 2.2 – Mel спектрограма голосу [11]

SVC-SO-VITS використовує згорткове навчання для вилучення характеристик голосу з вихідного аудіофайлу. Ці характеристики потім використовуються для створення нового аудіофайлу зі зміненим голосом.

Згорткове навчання є важливим компонентом SVC-SO-VITS, який дозволяє їй забезпечувати високу якість перетворення голосу.

Сьогодні глибинне навчання лежить в основі сервісів багатьох компаній: Facebook використовує нейронні мережі для алгоритму

автоматичного тегування, Google - для пошуку фотографій користувачів, Amazon - для генерування рекомендацій щодо товарів, Pinterest - для персоналізації домашніх сторінок користувачів, а Instagram - для пошукової інфраструктури. Однак класичним і, мабуть, найпоширенішим застосуванням мереж є обробка мовлення [11].

Характерною особливістю згорткових нейронних мереж є те, що нейрони в першому шарі організовані в певну структуру. Тобто в першому шарі нейрони розбиті на зображення (іноді їх називають картами) певного розміру, а різні карти в межах одного шару відповідають різним типам нейронів, що реагують на різні особливості мовного запису. Існує два основних типи обчислення активації в наступному шарі згорткової нейронної мережі. У першому типі активація наступного шару обчислюється як лінійна комбінація активації попереднього шару, і ваги цих лінійних активацій залежать тільки від взаємного розташування нейронів, типу нейрона, а не від положення нейрона на карті; у другому типі обчислення активація наступного рівня нейрона просто повторює активацію нейрона на попередньому рівні [11]. Ця додаткова структура робить згорткові нейронні мережі дуже придатними для обробки аудіо.

SVC-SO-VITS може виконувати такі завдання:

- перетворення голосу однієї людини в голос іншої людини – це може бути використано для створення реалістичних вокальних дубляжів, створення нових музичних творів, а також для розваг;
- зміна висоти тону голосу – це може бути використано для створення спеціальних ефектів, таких як голос робота або голос дитини;
- зміна тембру голосу – це може бути використано для створення більш драматичного або емоційного звучання [11].

SVC-SO-VITS є потужним інструментом, який може бути використаний для різних цілей. Вона забезпечує високу якість перетворення голосу, зберігаючи при цьому природність голосу.

Ось деякі конкретні приклади того, як SVC-SO-VITS може бути використана.

Студія звукозапису може її використовувати для створення вокальних дубляжів для фільмів і телешоу. Це може заощадити час і гроші, оскільки не потрібно наймати професійних акторів озвучування.

Композитор може її використовувати для створення нових музичних творів з голосами різних співаків. Це може допомогти створити більш різноманітні та цікаві музичні твори.

YouTube може її використовувати для створення відео, в яких вони співають голосами своїх улюблених співаків. Це може бути цікавим способом залучити глядачів і створити більш захоплюючий контент.

SVC-SO-VITS - це ще нова технологія, але вона має великий потенціал для застосування в різних сферах [11].

2.7 Структура згорткової нейронної мережі

Згорткові нейронні мережі широко використовуються в різних галузях. У згортковій нейронній мережі виходи проміжних шарів утворюють матрицю або набір матриць. Основними типами шарів у згортковій нейронній мережі є згортковий шар, об'єднуючий шар і повністю пов'язаний шар.

У порівнянні з нейронною мережею прямого поширення згорткова нейронна мережа використовує значно меншу кількість параметрів. На рисунку 2.3 зображена її архітектура [12].

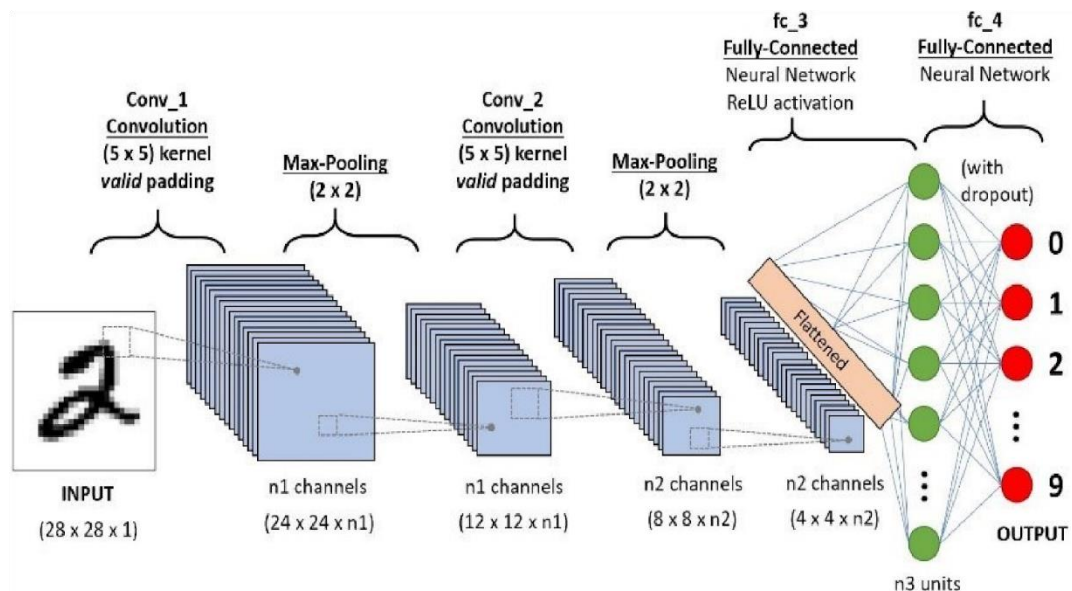


Рисунок 2.3 – Приклад архітектури згорткової нейронної мережі [12]

2.8 Згортковий шар

Згортковий шар є основним структурним елементом ШНМ. Параметри цього шару складаються з набору навчальних фільтрів (або ядер), які мають невелике сприйнятливим поле, але охоплюють всю глибину вхідного об'єму. Під час прямого проходу кожен фільтр згортається по ширині і висоті вхідного об'єму, і генерується 2D карта активації цього фільтра шляхом обчислення скалярного добутку фільтра і вхідних даних. В результаті мережа вчиться, які фільтри активувати, коли вона бачить певний тип об'єкта в певному просторовому розташуванні у вхідних даних.

Сума карт активації всіх фільтрів уздовж напрямку глибини формує загальний вихідний об'єм згорткового шару. Таким чином, кожен вхід у вихідному об'ємі можна інтерпретувати як вихід нейрона, який дивиться на невелику область входу і має спільні параметри з нейроном на тій самій карті активації [12].

Перший шар згорткової нейронної мережі – згортковий (convolution layer). Згорткова нейронна мережа працює на основі фільтрів, які займаються розпізнаванням ознак певних спектрів у аудіофайлі: прямих ліній, кривих певного типу тощо. Фільтр – це матриця чисел, які навчаються виявляти

ознаки певних об'єктів у аудіофайлі. Фільтр переміщується уздовж усього аудіозапису та виявляє, чи присутня деяка ознака у його частині. Для отримання такої відповіді здійснюється операція згортки (convolution operation), яка математично визначається як сума добутків елементів фільтра та частини вхідного аудіодоріжки. Назва згорткової нейронної мережі походить саме від назви цієї операції. На рисунку 2.4 зображена операція згортки.

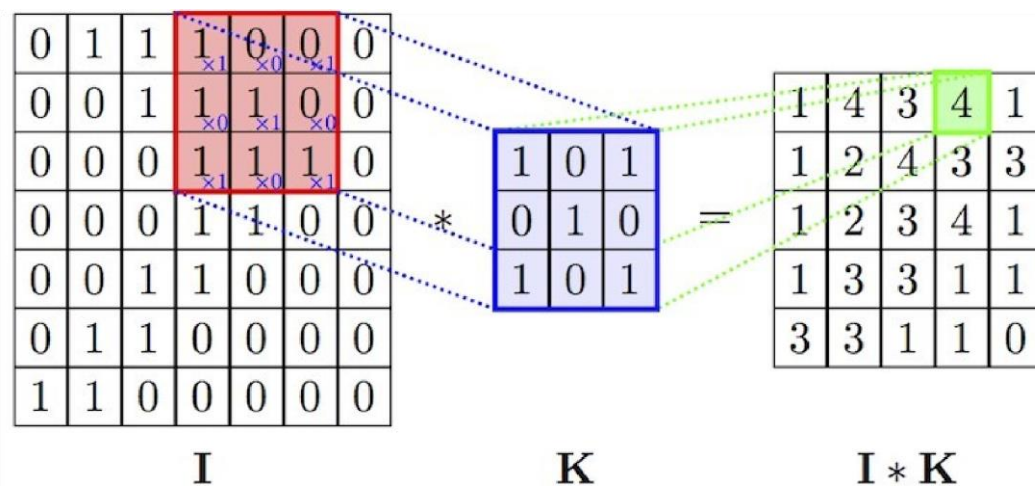


Рисунок 2.4 – Операція згортки [12]

2.9 Рекурсивна нейронна мережа

Рекурентні нейронні мережі - це тип глибоких нейронних мереж, побудованих шляхом рекурсивного застосування одного і того ж набору ваг до надбудови та активації структур у топологічному порядку, роблячи скалярні або структурні прогнози на вхідних структурах змінного розміру рисунок 2.5 [12]. У простих архітектурах для з'єднання вузлів з батьківськими об'єктами використовуються нелінійні характеристики, такі як тангенціальні функції активації або вагові матриці.

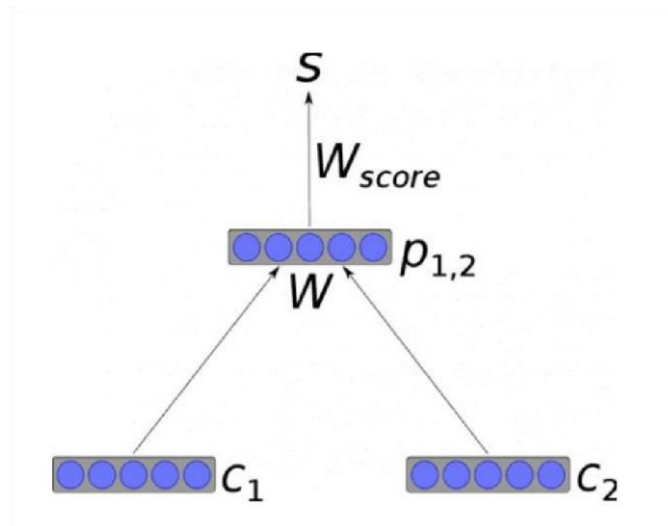


Рисунок 2.5 – Загальний вигляд рекурсивної нейронної мережі [12]

2.10 Оптимізатори

Оптимізатор виконує оптимізацію моделі, регулюючи прямий і зворотний градієнти сітки та генеруючи оновлення параметрів, які намагаються покращити функцію втрат. Мережа використовується для генерації втрат і градієнта, тоді як оптимізатор використовується для управління оптимізацією і генерації оновлень параметрів.

Стохастичний градієнтний спуск (часто скорочено SGD), який також називають інкрементним градієнтним спуском, є стохастичним підходом до методів оптимізації градієнтного спуску, який мінімізує цільову функцію, визначену як суму диференційованих членів. Іншими словами, SGD намагається знайти мінімальне або максимальне значення за допомогою ітерацій [12].

2.11 Рекурентна нейронна мережа

Рекурентні нейронні мережі - це тип штучних нейронних мереж, в яких зв'язки між вузлами утворюють орієнтований у часі граф. Це дозволяє формувати внутрішні стани мережі та демонструвати динамічну поведінку в часі. На відміну від нейронних мереж прямого поширення, RNN можуть

обробляти випадкові вхідні послідовності, використовуючи внутрішню пам'ять. Тому їх можна застосовувати для таких завдань, як розпізнавання мови.

Рекурентні нейронні мережі - це тип рекурентних ШНМ, в яких зв'язки між нейронами є спрямованими петлями, на відміну від прямих нейронних мереж [13]. Останнє означає, що вихідна інформація залежить не тільки від поточного входу, але й від стану нейронів на попередньому кроці. Така пам'ять дозволяє вирішувати задачі нейролінгвістичного програмування.

2.12 Проблеми навчання нейронних мереж

При навчанні нейронних мереж у будь-якому з підвидів навчання, можна зіткнутися з багатьма проблемами, які не дадуть досягти хороших результатів у вирішенні поставленої задачі. Тому є потреба в конфігурації самих нейронних мереж та процесу навчання. Розглянемо нижче основні проблеми при навчанні нейронних мереж [13].

Перенавчання нейронної мережі – це стан мережі, коли вона так привчає себе до певного набору даних, на якому вона навчається, що прогнозування на інших наборах даних стають повністю неправильними. Таким чином, коли ми навчаємо занадто довго нейронну мережу на одному і тому ж наборі даних, наша модель починає, замість вивчення узагальненої структури даних або їх закономірностей, вивчати шум даних або просто нерелевантну інформацію. Таку нейронну мережу називають перенавченою. Тоді нейронна мережа не в стані застосовувати вивчені закономірності до нових наборів даних, відповідно не в стані робити коректні прогнозування [13].

Недонавчання – це стан нейронної мережі, коли вона недостатньо вловила закономірності в даних, тому точність прогнозувань є дуже маленькою. Це протилежний стан перенавчання нейронної мережі.

В задачах регресії та класифікації, нескладно помітити, що недонавчена нейронна мережа не в стані ловити закономірності в даних на противагу перенавченій, яка починає вивчати шум в даних. Причини недонавченості нейронної мережі може бути недостатній час навчання, неправильна ініціалізація вагів, при якій модель не вчиться та мала складність моделі, при якій модель не встані взагалі вивчити хоч якісь закономірності в даних [13].

2.13 TensorFlow

TensorFlow - це бібліотека програмного забезпечення для машинного навчання з відкритим вихідним кодом, розроблена компанією Google для вирішення проблеми побудови та навчання нейронних мереж для клонування людського голосу. Використовується Google для досліджень та розробки власних продуктів. Основний API для використання бібліотеки реалізований для Python з реалізаціями на C Sharp, C++, Haskell, Java, Go та Swift; підтримує обчислення на CPU, GPU та спеціально розробленому Google TPU (tensor processing unit) [14].

2.14 Бібліотека NumPy

NumPy – це одна з основних бібліотек Python для обробки масивів даних. Ця бібліотека надає високопродуктивні об'єкти багатовимірних масивів та інструменти для роботи з цими масивами.

NumPy - це розширення Python, яке підтримує великі багатовимірні масиви та матриці і додає велику бібліотеку високорівневих математичних функцій для маніпулювання цими масивами. Попередник NumPy, Numeric, був створений Джимом Хугунінім. NumPy має відкритий вихідний код і має багато розробників.

Як інтерпретована мова, Python часто повільніше виконує математичні алгоритми, ніж скомпільовані мови, такі як C або Java; NumPy намагається

вирішити цю проблему в багатьох обчислювальних алгоритмах, надаючи підтримку багатовимірних масивів і різних функцій та операторів для маніпулювання ними. NumPy намагається вирішити цю проблему в багатьох обчислювальних алгоритмах, надаючи підтримку багатовимірних масивів і різних функцій та операторів для маніпулювання ними. Таким чином, практично будь-який алгоритм, який можна виразити як послідовність операцій над масивами або матрицями, може бути виконаний так само швидко, як еквівалентний код, написаний на C [14].

2.15 Google Colab

Google Colab – це безкоштовна платформа для виконання коду Python в хмарному середовищі, яка базується на інфраструктурі Google [15]. Вона надає можливість використовувати ресурси, такі як графічні процесори (GPU) та тензорні процесори (TPU), для тренування глибоких нейронних мереж, включаючи голосові моделі.

2.16 WORLD vocoder

WORLD vocoder - це безкоштовне програмне забезпечення для високоякісного аналізу, обробки та синтезу мовлення. Він може добре оцінювати основну частоту (F0), аперіодичність та спектральну оболонку та також генерувати мовлення, подібне до вхідного мовлення, лише з оцінених параметрів [15].

2.17 Librosa

Librosa - це бібліотека для обробки та аналізу аудіо-сигналів, яка написана мовою програмування Python. Ця бібліотека дозволяє розширювати можливості аналізу аудіоданих, зокрема для використання їх у завданнях

обробки сигналів, аудіоаналізу, машинного навчання та інших додаткових областей, ось приведені основні можливості [15]:

Завдання обробки сигналів:

- зчитування та запис аудіофайлів у різних форматах;
- виконання операцій преобразування сигналів, таких як Fourier-перетворення;
- визначення характеристик сигналу, таких як темп, тон, гучність тощо.

Аналіз аудіофайлів:

- виділення спектральних характеристик аудіо-сигналів;
- визначення ритму та тембру музичних композицій;
- обчислення мел-кепстральних коефіцієнтів (MFCC) для аудіоаналізу.

Робота з музичними функціями:

- визначення ключових та темпових характеристик музичних треків;
- екстракція ритмічних шаблонів та характеристик.

Підтримка машинного навчання:

- генерація векторів ознак для аудіофайлів, які можна використовувати в алгоритмах машинного навчання;
- підготовка даних для класифікації та розпізнавання звуку.

Візуалізація даних: створення графіків та візуалізація аудіо-сигналів та їх характеристик [15].

2.18 Onnx

ONNX - це відкритий формат для представлення моделей машинного навчання. Він був розроблений компаніями Microsoft, Facebook і Amazon Web Services. ONNX підтримує широкий спектр алгоритмів машинного навчання, включаючи навчання з підкріпленням, навчання за зразками та навчання без нагляду.

2.19 Torchaudio

Torchaudio - це бібліотека для обробки звуку з PyTorch. Torchaudio надає широкий спектр функцій для роботи зі звуком, включаючи [15]:

- читання та запис звуку: Torchaudio може читати та записувати звук у різних форматах, включаючи WAV, MP3, FLAC та інші;
- перетворення звуку: Torchaudio може перетворювати звук з одного формату в інший, наприклад, з моно в стерео або з низької якості в високу якість;
- аналіз звуку: Torchaudio може використовуватися для аналізу звуку, наприклад, для визначення частоти, тривалості та гучності звуку;
- синтез звуку: Torchaudio може використовуватися для синтезу звуку, наприклад, для створення мовлення або музики.

2.20 Розрахунок споживання пам'яті мережею

Ліміти пам'яті для сучасних графічних процесорів становлять 3, 4 або 6 ГБ, а найвищі GPU мають ліміт пам'яті 12 ГБ. Існує три основні джерела об'єктів, що споживають пам'ять. Це сира кількість активацій та їх градієнти в кожному шарі нейронної мережі. Як правило, більшість активацій розміщується в першому шарі згорткової нейронної мережі. Це слід зазначити, тому що ці активації необхідні для зворотного поширення. Однак у додатках, де нейронна мережа використовується лише під час навчання, кількість активацій можна зменшити, зберігаючи лише поточні активації в будь-якому шарі і відкидаючи всі попередні в нижніх шарах. Це числа, які включають параметри мережі, градієнти під час зворотного розповсюдження і часто кроки кешу. Тому пам'ять, що використовується для зберігання вектора параметрів, повинна бути приблизно втричі більшою. Згорткові нейронні мережі повинні підтримувати гетерогенну пам'ять, наприклад, пакети даних зображень, і, ймовірно, вимагатимуть складнішої реалізації. Після того, як приблизний

розмір загальної кількості значень (активація, градієнт і т.д.) відомий, результат потрібно конвертувати в ГБ. Кількість значень потрібно помножити на чотири, щоб отримати кількість байт, а потім розділити на 10243, щоб отримати пам'ять у гігабайтах [15].

2.21 Висновки за розділом 2

Розглянуто основні види глибинних нейронних мереж. На основі проведеного порівняльного аналізу архітектур нейронних мереж показано, що саме згорткові нейронні мережі знайшли широке застосування при обробці зображень. Описані основні елементи згорткової мережі та особливості вибору параметрів для навчання нейронних мереж.

Розглянуті бібліотеки необхідні для створення нейронної мережі для клонування голосу. Проаналізовані переваги та недоліки кожної бібліотеки. Також розглянуті та проаналізовані фреймворки, написані на різних мовах програмування, що необхідні для роботи нейронної мережі.

3 НАЛАШТУВАННЯ СЕРЕДОВИЩА ДЛЯ НАВЧАННЯ

Першим етапом, необхідним для навчання моделі розпізнавати емоції, є налаштування робочого середовища. Навчання було проведено на комп'ютері з наступними характеристиками:

- чотириядерний Intel Pentium Silver N5000 (1.1 - 2.7 ГГц);
- тип оперативної пам'яті DDR4-2133 МГц;
- 4 ГБ оперативної пам'яті;
- графічний адаптер, об'єм пам'яті: Intel UHD Graphics 605;
- операційна система Windows 10.

Наразі існує велика кількість мов програмування та програмних бібліотек для роботи зі штучними нейронними мережами та генетичними алгоритмами, реалізованих на різних мовах програмування, таких як C++, C# та Python. Python є мовою програмування, яку обирають для розв'язання задач та реалізації алгоритмів. Python, обрана як інструмент вибору, є високорівневою мовою програмування загального призначення, орієнтованою на підвищення продуктивності розробника та читабельності коду. Водночас стандартна бібліотека містить низку корисних функцій. Python підтримує структуроване, об'єктно-орієнтоване, функціональне, імперативне та вью-орієнтоване програмування. Ключові архітектурні особливості включають динамічну типізацію, автоматичне управління пам'яттю, повну інтроспекцію, обробку винятків, підтримку багатопотокових обчислень та високорівневі структури даних. Python підтримує розбиття програми на модулі та модулі, які можуть бути об'єднані в пакети [16]. Виходячи з цього, мовою програмування було обрано Python, а середовищем програмування - Microsoft Visual Studio Code. Існує низка бібліотек та систем, які підтримують інтерфейс мови Python для розв'язання задач машинного навчання. Деякі з них: Apache Singa, Caffee, Keras, Microsoft Cognitive Toolkit, MXNet, TensorFlow, Theano та PyTorch.

Для навчання голосової моделі була обрана SO-VITS-SVC.

Вона добре підходить для завдань синтезу мовлення та поєднує в собі переваги згорткових нейронних мереж (CNN) та трансформерів (Transformer). CNN забезпечують хороше розділення по частоті, а трансформери - хороше розділення по часу, також вона має високу ефективність навчання. Вона використовує паралельне навчання для прискорення процесу навчання та була успішно застосована в різних завданнях синтезу мовлення і показала хороші результати на наборах даних [16].

SO-VITS-SVC складається з двох основних компонентів: SVC (singing voice conversion) – це свернута нейронна мережа, яка використовується для вилучення частотних характеристик з вхідного сигналу, вона оптимізована для прямого перетворення висоти тону. Це означає, що ваш голос на виході співатиме ті самі ноти, що й на вході.

Це, по суті, поєднує можливості Tone Transfer з можливостями моделей глибокого перетворення тексту в мовлення, які ми бачили в минулі роки.

SO-ViTTS - це трансформер, який використовується для об'єднання частотних характеристик з часовими характеристиками.

SVC забезпечує хороше розділення по частоті, а SO-ViTTS – хороше розділення по часу. Це робить, що SO-VITS-SVC добре підходить архітектурою для завдань синтезу мовлення, які вимагають одночасно хорошого розділення по частоті та часу [16].

Також вона має високу ефективність навчання бо використовує паралельне навчання для прискорення процесу навчання. Паралельне навчання дозволяє навчати кілька моделей одночасно, що може значно скоротити час навчання.

На підставі наведених вище аргументів я вважаю, що це є хорошою архітектурою для навчання голосової моделі. Вона поєднує в собі переваги звернутих нейронних мереж та трансформерів, має високу ефективність навчання та була успішно застосована в різних завданнях синтезу мовлення.

3.1 Реалізація навчання

Для навчання було обране середовище Google Colab. Colab є безкоштовною платформою, яка надає можливість виконання коду Python у веб-середовищі з використанням віртуальних машин Google. Однією з актуальних задач у галузі штучного інтелекту є створення голосових моделей, які можуть розпізнавати та генерувати мову. Google Colab виявляється потужним інструментом для реалізації цього завдання завдяки своїм можливостям та ресурсам.

Можливості Google Colab для навчання голосових моделей [17]:

- GPU та TPU – Google Colab надає безкоштовний доступ до графічних та тензорних процесорів (GPU та TPU), що значно прискорює процес тренування глибоких нейронних мереж, включаючи голосові моделі;
- обмін інформацією та ресурсами – інтеграція з Google Drive дозволяє зберігати та здійснювати обмін даними та моделями між різними користувачами;
- доступність популярних бібліотек – Colab має попередньо встановлені бібліотеки, такі як TensorFlow та PyTorch, які часто використовуються для створення голосових моделей;
- робота з аудіо-даними – завдяки можливості завантаження та обробки аудіо-даних в Colab, користувач може легко використовувати різні бібліотеки для аналізу та обробки голосових сигналів;
- візуалізація результатів – Colab надає зручні інструменти для візуалізації результатів навчання, що дозволяє аналізувати ефективність голосових моделей.

3.2 Підключення необхідних бібліотек для роботи

```
import tensorflow as tf
from svc_so_vits import SVCModel
```

```

# Імпортуйте бібліотеки
import tensorflow as tf
from svc_so_vits import SVCModel
# Налаштуйте параметри навчання
batch_size = 64
epochs = 100
# Створіть набір даних
train_dataset = tf.data.Dataset.from_tensor_slices(
    (audio_files_train, target_voices_train)
)
train_dataset = train_dataset.shuffle(buffer_size=batch_size * 10)
train_dataset = train_dataset.batch(batch_size)
# Створіть модель
model = SVCModel()
# Навчіть модель
model.fit(train_dataset, epochs=epochs)
# Збережіть модель
model.save("path-to-model")

```

3.3 Загальний опис датасету

Дані складаються із пар аудіофайлів і цільових голосів. Аудіофайли повинні бути записані в тому ж форматі і мати однакову тривалість. Цільові голоси повинні бути різними, але близькими за тембром і висотою тону.

За основу набору датасету був обраний Петро Олексійович Порошенко, він є дуже популярною медійною особистістю і через це проблем з пошуком матеріалу для створення та підготовки датасету не було.

Шляхом проб та експериментів була обрана тривалість однієї аудіодоріжки від 8 до 10 секунд, попередньо були вирізані сторонні шуми

На спектрограмі рисунку 3.1 можна побачити що у аудіофайлі відсутні паузи, лише чисте мовлення.

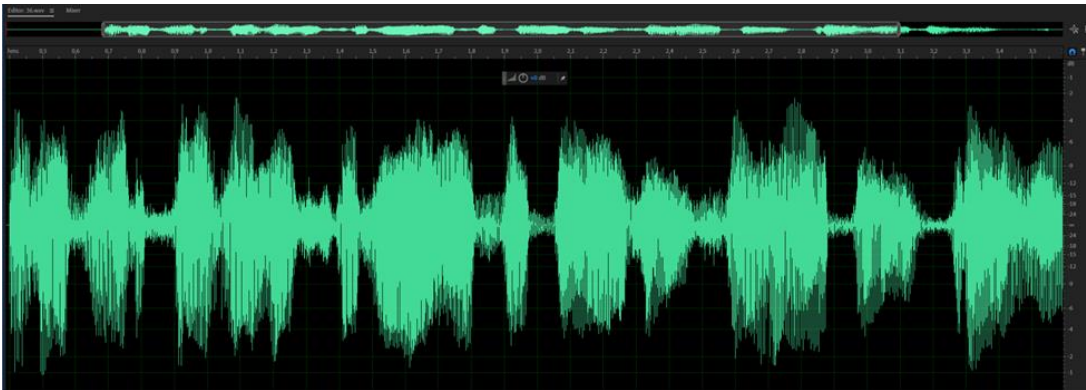


Рисунок 3.1 – Зображення спектрограми

Усього для навчання моделі було використано датасет на 186 файлів рисунку 3.2 та рисунку 3.3.

Mode	LastWriteTime		Length	Name
----	-----	-----	-----	----
-a----	29.10.2023	14:19	919704	1.wav
-a----	29.10.2023	14:23	387304	10.wav
-a----	29.10.2023	14:23	569328	11.wav
-a----	29.10.2023	14:23	619776	12.wav
-a----	29.10.2023	14:24	1092208	13.wav
-a----	29.10.2023	14:24	991976	14.wav
-a----	29.10.2023	14:25	468160	15.wav
-a----	29.10.2023	14:25	119864	16.wav
-a----	29.10.2023	14:26	1223184	17.wav
-a----	29.10.2023	14:26	681464	18.wav
-a----	29.10.2023	14:26	1146120	19.wav
-a----	29.10.2023	14:20	854240	2.wav
-a----	29.10.2023	14:27	519376	20.wav
-a----	29.10.2023	14:28	661608	21.wav
-a----	29.10.2023	14:28	279336	22.wav
-a----	29.10.2023	14:29	931672	23.wav
-a----	29.10.2023	14:30	1201928	24.wav
-a----	29.10.2023	14:31	295500	25.wav
-a----	29.10.2023	14:31	1149704	26.wav
-a----	29.10.2023	14:31	1020128	27.wav
-a----	29.10.2023	14:32	198536	28.wav
-a----	29.10.2023	14:32	195672	29.wav
-a----	29.10.2023	14:20	653768	3.wav
-a----	29.10.2023	14:33	558024	30.wav
-a----	29.10.2023	14:33	914096	31.wav
-a----	29.10.2023	14:33	694056	32.wav
-a----	29.10.2023	14:34	760960	33.wav
-a----	29.10.2023	14:34	604536	34.wav
-a----	29.10.2023	14:34	1230224	35.wav
-a----	29.10.2023	14:35	918704	36.wav
-a----	29.10.2023	14:35	1151960	37.wav
-a----	29.10.2023	14:35	213968	38.wav
-a----	29.10.2023	14:36	1251808	39.wav
-a----	29.10.2023	14:21	956680	4.wav
-a----	29.10.2023	14:36	783792	40.wav
-a----	29.10.2023	14:37	485728	41.wav
-a----	29.10.2023	14:37	1079224	42.wav

Рисунок 3.2 – Зображення датасету для навчання моделі

Mode	LastWriteTime		Length	Name
----	-----		-----	----
-a----	29.10.2023	15:14	823484	philipp_test_voice_1.wav
-a----	29.10.2023	15:14	823484	philipp_test_voice_2.wav
-a----	29.10.2023	15:14	823484	philipp_test_voice_3.wav

Рисунок 3.3 – Зображення датасету для тестів моделі

3.4 Розподіл даних для навчання та перевірки

У процесі розробки нейронної мережі відбувається поділ даних на набір для навчання та перевірки.

Розподіл тренувальних масивів.

```
{'train': {'log_interval': 200, 'eval_interval': 1000, 'seed': 1234, 'epochs': 10000, 'learning_rate': 0.0001, 'betas': [0.8, 0.99], 'eps': 1e-09, 'batch_size': 6, 'fp16_run': False, 'lr_decay': 0.999875, 'segment_size': 10240, 'init_lr_ratio': 1, 'warmup_epochs': 0, 'c_mel': 45, 'c_kl': 1.0, 'use_sr': True, 'max_speclen': 512, 'port': '8001', 'keep_ckpts': 3}, 'data': {'training_files': 'filelists/train.txt', 'validation_files': 'filelists/val.txt', 'max_wav_value': 32768.0, 'sampling_rate': 44100, 'filter_length': 2048, 'hop_length': 512, 'win_length': 2048, 'n_mel_channels': 80, 'mel_fmin': 0.0, 'mel_fmax': 22050, 'contentvec_final_proj': False}, 'model': {'inter_channels': 192, 'hidden_channels': 192, 'filter_channels': 768, 'n_heads': 2, 'n_layers': 6, 'kernel_size': 3, 'p_dropout': 0.1, 'resblock': '1', 'resblock_kernel_sizes': [3, 7, 11], 'resblock_dilation_sizes': [[1, 3, 5], [1, 3, 5], [1, 3, 5]], 'upsample_rates': [8, 8, 2, 2, 2], 'upsample_initial_channel': 512, 'upsample_kernel_sizes': [16, 16, 4, 4, 4], 'n_layers_q': 3, 'use_spectral_norm': False, 'gin_channels': 256, 'ssl_dim': 768, 'n_speakers': 200}, 'spk': {'Poroh': 0}, 'model_dir': './logs/44k', 'reset': False}
```

3.5 Розробка глибокої згорткової нейронної мережі

Згорткові нейронні мережі це чудовий спосіб структурування в мережі, побудови так званої інваріантності (виділення незмінних ознак).

Згорткові нейронні мережі досить щільно використовуються при роботі з аудіо та відео, включаючи медичні зображення. Також вони використовуються в мовних технологіях, де фахівці використовують глибоке навчання для розуміння та відтворення мовних конструкцій [18].

Двовимірний згортка – це згортка, де ядро має таку саму глибину, як і вхідні дані, тому, теоретично, вам не потрібно вказувати глибину ядра, якщо ви знаєте глибину вхідних даних [19].

Використовуючи TensorFlow, потрібно лише вказати ширину та висоту ядра в класі Conv2D, оскільки глибина автоматично обчислюється на основі глибини введення. Таким чином, походить від того, що ви ковзаєте між двома вимірами (тобто шириною та висотою).

Використання загальних синаптичних ваг у площині згорткової нейронної мережі зменшує кількість регульованих параметрів рисунок 3.4 та рисунок 3.5.

```
"train": {
  "log_interval": 200,
  "eval_interval": 1000,
  "seed": 1234,
  "epochs": 10000,
  "learning_rate": 0.0001,
  "betas": [
    0.8,
    0.99
  ],
  "eps": 1e-09,
  "batch_size": 6,
  "fp16_run": false,
  "lr_decay": 0.999875,
  "segment_size": 10240,
  "init_lr_ratio": 1,
  "warmup_epochs": 0,
  "c_mel": 45,
  "c_kl": 1.0,
  "use_sr": true,
  "max_speclen": 512,
  "port": "8001",
  "keep_ckpts": 3
},
"data": {
  "training_files": "filelists/train.txt",
  "validation_files": "filelists/val.txt",
  "max_wav_value": 32768.0,
  "sampling_rate": 44100,
  "filter_length": 2048,
  "hop_length": 512,
  "win_length": 2048,
  "n_mel_channels": 80,
  "mel_fmin": 0.0,
  "mel_fmax": 22050,
  "contentvec_final_proj": false
},
"model": {
  "inter_channels": 192,
  "hidden_channels": 192,
```

Рисунок 3.4 – Налаштування згорткової нейронної мережі

```

"model": {
  "inter_channels": 192,
  "hidden_channels": 192,
  "filter_channels": 768,
  "n_heads": 2,
  "n_layers": 6,
  "kernel_size": 3,
  "p_dropout": 0.1,
  "resblock": "1",
  "resblock_kernel_sizes": [
    3,
    7,
    11
  ],
  "resblock_dilation_sizes": [
    [
      1,
      3,
      5
    ],
    [
      1,
      3,
      5
    ],
    [
      1,
      3,
      5
    ]
  ],
  "upsample_rates": [
    8,
    8,
    2,
    2,
    2
  ],

```

Рисунок 3.5 – Налаштування згорткової нейронної мережі

Використання згорткових нейронних мереж має недоліки, які усуваються шляхом використання спільних синаптичних коефіцієнтів та загальних обчислювальних методів [19]. Локальні рецептивні поля в згортковому шарі частково перекриваються і мають спільні нейронні значення.

Тому не потрібно обчислювати значення всіх нейронів у локальних рецептивних полях, а використовувати значення, обчислені на попередніх ітераціях. Використовуючи цей метод, можна значно скоротити час обчислень і зменшити обчислювальну складність.

3.6 Висновки за розділом 3

Проведено обчислювальні експерименти з базовими архітектурами нейромереж. Проаналізовано вплив функцій активації та алгоритмів активації на штучні нейронні мережі. Проведені дослідження на вплив кількості нейронів на нейромережеве клонування голосу.

Розроблено нейронну мережу для клонування голосу.

Розроблено модель штучної згорткової нейронної мережі для клонування голосу.

4 ЗАПУСК ПРОЦЕСУ НАВЧАННЯ

У своїй промові Петро Олексійович має дуже великий спектр емоцій, що добре відзначилось на навчанні моделі.

Процес навчання нейромережі починається з тренування на датасеті `train.csv`. Даний датасет містить 186 файлів формату `.ogg`. Кожна аудіо доріжка була попередньо оброблена та нарізана по 10 секунд.

Після тренування розпочинається процес перевірки чи коректно працює модель, чи є якісь непотрібні звуки або дефекти мовлення.

4.1 Обчислювальний експеримент

Зазвичай дослідження проводяться з великими обсягами даних, які вимагають витрат ресурсів і часу. Щоб ефективно працювати з великими обсягами даних, потрібно використовувати параметри (епоха, розмір партії, ітерації), так як часто немає можливості завантажити відразу всі дані в обробку.

Для подолання цієї проблеми дані поділяються на частину меншого розміру, завантажують їх по черзі і оновлюють ваги нейромережі в кінці кожного кроку, підстроюючи їх під дані.

У термінології нейронних мереж епоха (`epoch`) – одна ітерація у процесі тренування (отримання вихідних значень та оновлення ваг всіх тренувальних екземплярів) [19]. Одна епоха (`epoch`) – весь датасет пройшов через нейронну мережу у прямому та зворотному напрямку лише один раз.

Для отримання певних результатів недостатньо одного повного проходу даних через нейронну мережу. Необхідно пустити повний набір даних по мережі кілька разів.

Також потрібно врахувати, що використовується обмежений набір даних, щоб оптимізувати навчання та створити криву інформації. Це робиться

за допомогою градієнтного спуску – ітеративного процесу. Тому оновлення ваг після одного проходження недостатньо.

Одна епоха призводить до недонавчання, а велика кількість епох – до перенавчання [19].

Зі збільшенням числа епох, ваги нейронної мережі змінюються дедалі більше разів. Крива щоразу краще підлаштовується під дані, переходячи послідовно з погано навченого стану в оптимальне. Якщо вчасно не зупинитися, то може відбутися перенавчання – коли крива дуже точно підлаштувалася, а узагальнююча здатність зникла.

В ході виконання роботи по навчанню нейронної мережі було проведено декілька обчислювальних експериментів з різною кількістю епох. В перший раз було проведено експеримент з 10000, потім з 30000 епохами. В третій раз був проведений обчислювальний експеримент з 50000 епохами.

Отримали наступні результати, які зображено на графіках рисунку 4.1, де синім кольором зображено тренування нейронної мережі, червоний відображає прогнозування нейронної мережі.

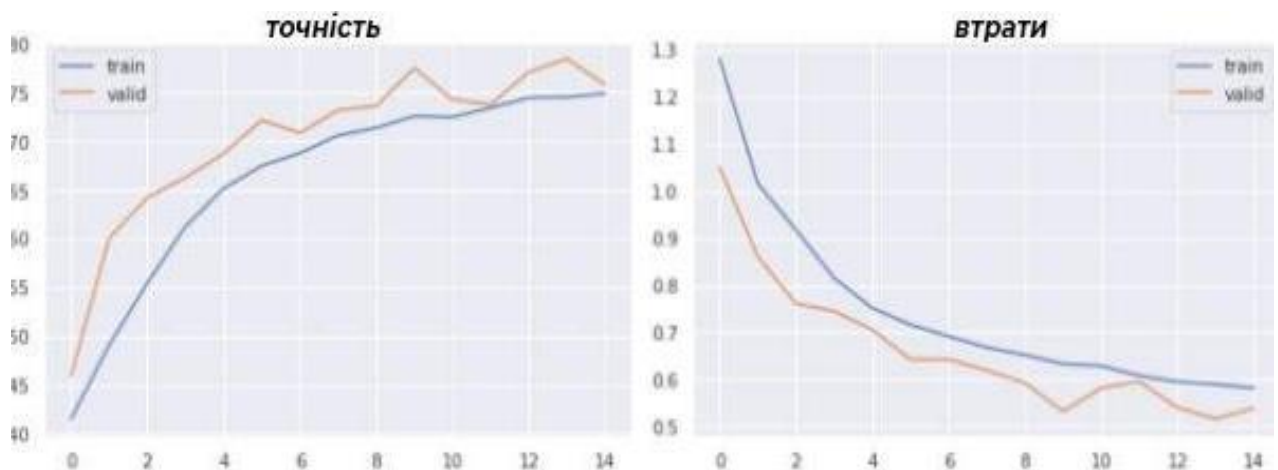


Рисунок 4.1 – Порівняння результатів роботи нейронної мережі на 50000 епохах

4.2 Результати роботи нейронної мережі

Кодувальник тексту у мел модель є нейронною мережею яка складається з кількох шарів згорткових нейронних мереж (CNN) і рекурентних нейронних мереж (RNN), які обробляють вхідний текст і генерують спектрограми Мела.

Спектрограма в мел масштабі є двовимірною матрицею, що представляє частотний склад мовного сигналу з плином часу.

Вертикальна ось представляє частоту в герцах, а горизонтальна ось час в секундах. Кожна точка у матриці представляє амплітуду сигналу на певній частоті та часі.

Кодувальник голосу приймає спектрограму в масштабі Mel і перетворює у відповідну форму сигналу, він по суті перетворює спектрограму назад в звуковий сигнал.

Обумовивши кодувальник голосу вбудованим динаміком, ми можемо вловити унікальні характеристики голосу цільового мовця, як-от його інтонація, акцент і стиль мовлення. Це дає змогу синтезованій мові звучати так само, як цільовий мовець, навіть якщо вхідний текст вимовляється іншою людиною.

Кодер динаміка приймає короткий сегмент аудіо від цільового динаміка та кодує його у фіксований вектор довжини, який представляє унікальні характеристики голосу мовця. Цей вектор використовується для налаштування кодувальника голосу під час синтезу мовлення, дозволяючи моделі генерувати мовлення, яке звучить як цільовий мовець.

Загалом, комбінація цих компонентів дає змогу клонувати голос шляхом обробки вхідного тексту, генерації спектрограм у масштабі Mel, вилучення вбудованих динаміків, синтезування форми хвилі, яка звучить як цільова голосова модель рисунку 4.2.



Рисунок 4.2 – Схема роботи з навченою голосовою моделлю

Одним із ключових критеріїв ефективності моделі є її акустична точність, вимірювана з використанням метрик, таких як Mean Squared Error (MSE). Проаналізовано результати на валідаційному та тестовому наборах даних для визначення, наскільки добре модель відтворює оригінальний голос, інтонації та загальні схожості, щоб оцінити ступінь достовірності та реалістичності синтезованих голосових даних.

Було проведено ряд тестів на різному ступені навчання моделі, які можна переглянути в додатку Б.

4.3 Висновки за розділом 4

Було проаналізовано роботопридатність голосової моделі включно різних тестів кожні 10000 епох навчання, кожна наступна ревізія в тих чи інших тестах ставала краще, кінцева версія моделі показала гарний результат та задовольнила очікування.

ВИСНОВКИ

У роботі досліджено актуальну проблему клонування голосу людини. Розглянуто переваги та недоліки існуючих технологій та систем для її вирішення, сфери застосування таких систем та сучасні підходи до їх розв'язання.

Досліджено засоби для вирішення поставленої проблеми. Розглянуті моделі нейронних мереж придатні для навчання голосової моделі.

Було проведено порівняння сучасних технологій для навчання голосових моделей. У результаті порівняння між SVC-SO-VITS, Whisper і Vox2Vox було обрано SVC-SO-VITS, як архітектура, що найліпше підходить до вирішення поставленого завдання.

Проведено навчання моделі нейронної мережі для перетворення голосу людини на голос моделі: налаштоване середовище для навчання, підготовлені записи голосу людини з відповідного датасету, створені необхідні для тренування файли та успішно виконано навчання моделі.

Розроблена готова до впровадження комп'ютерна система, яка перетворює вхідний голос людини на голос навченої моделі Петра Порошенка.

Навчена нейронна мережа дозволяє клонувати голос людини. Абсолютна відсоткова похибка становить не більше 20%. При цьому виявлено, що точність розробленої системи залежить від якості навчання моделі нейронної мережі, яка у свою чергу залежить від якості датасету.

Наукова новизна роботи полягає у застосуванні та навчанні голосової моделі SVC-SO-VITS. А також у використанні інноваційного підходу до навчання, який дозволяє досягти значного успіху в області клонування голосу людини.

Поставлена проблема навчання голосової моделі за допомогою штучного інтелекту повністю вирішена з допустимою точністю.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Теорія та практика нейронних мереж [Електронний ресурс]. – Режим доступу:
https://ela.kpi.ua/bitstream/123456789/49841/1/Teoriia_praktyka_neironnykh_mer_ezh.pdf.
2. Stuart, R. Artificial Intelligence: A Modern Approach (AIMA) / R. Stuart, P. Norvig. — Pearson, 2009 – 1152 p.
3. Turing, A.M. Computing Machinery and Intelligence / A.M. Turing // Mind. – 1950. – vol. LIX, № 236. – P. 433 – 460.
4. Ai [Electronic resource]. – Access mode:
<https://en.wikipedia.org/wiki/Ai>.
5. Нейромережа: куди це все йде [Електронний ресурс]. – Режим доступу: <https://habr.com/ua/post/482794>.
6. Neural network types [Electronic resource]. – Access mode:
<https://www.mygreatlearning.com/blog/types-of-neural-networks/>.
7. Основи нейрокомп'ютінку [Електронний ресурс]. – Режим доступу:
https://blogs.kpi.kharkov.ua/v2/zakovorotny/wp-content/uploads/sites/8/2014/04/PZ_NK_ukr_2013.pdf.
8. Руденко, О.Г. Штучні нейронні мережі : навчальний посібник / О.Г. Руденко, Є.В. Бодянський. — Харків : ТОВ "Компанія СМІТ", 2006. — 404 с.
9. Backpropagation [Electronic resource]. – Access mode:
<https://en.wikipedia.org/wiki/Backpropagation>.
10. Generative adversarial networks [Electronic resource]. – Access mode:
<https://habr.com/ru/articles/352794/>.
11. SVC Voice Models [Electronic resource]. – Access mode:
<https://networkbuildz.com/svc-voice-changer/>.
12. Artificial neural networks [Electronic resource]. – Access mode:
<https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/>.

13. Основи штучних неймереж [Електронний ресурс]. – Режим доступу: https://learn.ztu.edu.ua/pluginfile.php/290224/mod_resource/content/1.
14. TensorFlow [Electronic resource]. – Access mode: <https://en.wikipedia.org/wiki/TensorFlow>.
15. Introduction to Colab Enterprise [Electronic resource]. – Access mode: <https://cloud.google.com/colab/docs/introduction>.
16. SVCVoiceCloning [Electronic resource]. – Access mode: <https://medium.com/analytics-vidhya/the-voice-real-time-voice-cloning-b7815afd6869>.
17. Апаратні засоби Libre [Електронний ресурс]. – Режим доступу: <https://www.hwlibre.com/uk/>.
18. Kernel (image processing) [Electronic resource]. – Access mode: [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing)).
19. Попов, С.В. Штучні нейронні мережі зі спеціалізованими архітектурами в задачах обробки часових рядів : автореф. дис. ... д-ра техн. наук : 05.13.23 «Системи та засоби штучного інтелекту» / С.В. Попов. – Х. : ХНУРЕ, 2009. – 36 с.

ДОДАТОК А
Технічне завдання

Вступ

Навчання голосової моделі за допомогою штучного інтелекту дозволяє використовувати такі моделі у різних сферах нашого життя, наприклад у сфері розваг або для бізнесу, так як Петро Олексійович є популярною та медійною особистістю пошук аудіозаписів з його голосом не спричинить складнощів, у своїх інтерв'ю зазвичай використовується високоякісне обладнання, що є плюсом і буде легше обробити датасет для навчання голосової моделі.

A.1 Підстави для розробки

Дипломна кваліфікаційна робота виконується на підставі завдання, виданого Національним університетом «Запорізька політехніка» згідно з наказом № 437 від 14 листопада 2023 р. за темою: «Дослідження та програмна реалізація навчання голосової моделі за допомогою штучного інтелекту».

A.2 Призначення розробки

Голосова модель має використовуватися для отримання вихідного аудіофайлу озвученого голосовою моделью Петра Олексійовича Порошенка.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Набір функціональних вимог, які має забезпечувати програма:

- чіткість мовлення;
- швидкість мовлення;
- коректна робота з довгими фразами;
- коректна робота з тишею;

- коректна робота з шумом;
- коректна робота з акцентом;
- передача емоційного забарвлення;
- коректна робота з різними голосами.

A.3.2 Вимоги до складу та параметрів технічних засобів

Для роботи програми: необхідні такі технічні характеристики:

- чотириядерний Intel Pentium Silver N5000 (1.1 - 2.7 ГГц);
- тип оперативної пам'яті DDR4-2133 МГц;
- 4 ГБ оперативної пам'яті;
- графічний адаптер, об'єм пам'яті: Intel UHD Graphics 605;
- операційна система Windows 10.

A.4 Порядок контролю та приймання

Дипломна кваліфікаційна робота мусить бути виконана у відповідності з визначеним календарним планом згідно встановлених завдань, їх строків і результатів виконання.

Захист роботи відбувається за оголошеним завчасно графіком захисту дипломних кваліфікаційних робіт і відповідно з графіком роботи екзаменаційної комісії.

ДОДАТОК Б
Тести навчання голосової моделі

Епоха 10000	
Епоха	Результат
Тест на чіткість мовлення	Модель виявляє помітне поліпшення у чіткості та точності вимови. Попередні дефекти значно зменшуються, і модель краще розпізнає та відтворює слова.
Тест на швидкість мовлення	Модель виявляє відчутне покращення у розпізнаванні швидкого темпу мовлення. Хоча ще є місце для вдосконалення, загальний рівень ефективності вже значно вищий.
Тест на довгі фрази	Модель вже здатна поліпшити обробку довгих фраз, але може втрачати деякий контекст і допускати невеликі неточності в визначенні довгих текстів.
Тест на тишу	Модель виявляє стабільність у розпізнаванні відсутності звуку та управлінні паузами, але може виявляти певні артефакти або шумові компоненти у відсутності аудіосигналу.
Тест на шум	Модель показує покращення в ізоляції голосу від фонового шуму, але ще може виявляти деякі труднощі при визначенні мовлення в умовах шуму.
Тест на акцент	Модель показує початкові зусилля в адаптації до різних акцентів. Є деякі випадки, коли розпізнавання може бути неоднорідним у присутності акцентованого мовлення.
Тест на емоційне забарвлення:	Модель починає виявляти базову здатність передавати емоційне забарвлення, проте існують випадки, де відтворене мовлення може бути менш виразним або недостатньо емоційним.
Тест на різні голоси	Модель демонструє початкові здатності вирізняння та адаптації до різних тембрів голосу. Є невеликі аспекти, де розпізнавання може бути менш точним або менш адаптивним.

Рисунок Б.1 – Епоха 10000

Епоха 20000	
Епоха	Результат
Тест на чіткість мовлення	Чіткість мовлення виявляє наближення до ідеального рівня. Модель відзначається виправленням більшості дефектів, і розпізнавання слів стає надзвичайно точним.
Тест на швидкість мовлення	Виявлення швидкого темпу мовлення стає більш вдосконалим. Модель ефективно справляється із завданням у порівнянні з попереднім етапом.
Тест на довгі фрази	Досягнуті помітні покращення в збереженні контексту при обробці довгих фраз. Модель ефективніше розпізнає і відтворює тривалі текстові вирази.
Тест на тишу	Певні поліпшення у роботі з відсутнім звуком. Модель майже ідеально справляється із збереженням тиші, але може залишатися деяка підвищена чутливість до непотрібних звукових елементів.
Тест на шум	Досягнуті помітні зміни у роботі з шумом. Модель краще ізолює голос від фонового шуму та ефективніше розпізнає мовлення.
Тест на акцент	Удосконалення у роботі з різноманітними акцентами. Модель ефективніше адаптується до вимови із застосуванням різних акцентів.
Тест на емоційне забарвлення:	Зафіксовано значний прогрес у передачі емоцій через мовлення. Модель стає більш виразною у відтворенні різних емоційних відтінків.
Тест на різні голоси	Помітні відчутні зміни у роботі з різноманітними голосами. Модель ефективніше розпізнає та адаптується до різних тембрів і інтонацій голосу.

Рисунок Б.2 – Епоха 20000

Епоха 30000	
Епоха	Результат
Тест на чіткість мовлення	Модель досягає високого ступеня чіткості у мовленні. Вимова стає практично бездоганною, і точність розпізнавання слів максимально висока.
Тест на швидкість мовлення	Модель демонструє високий рівень вирізняння швидкого темпу мовлення. Швидкість та точність суттєво зросли.
Тест на довгі фрази	Покращення стає більш виразним, і модель значно досконало обробляє тривалі фрази, підтримуючи високий рівень контексту.
Тест на тишу	Виявлено вражаючий прогрес у відсутності звуку. Модель більш ефективно розпізнає та обробляє аудіофайли, що складаються тільки з тиші.
Тест на шум	Модель виявляє прогрес у роботі з різноманітними видами шуму. Ізоляція голосу стає значно більш дієвою, знижуючи вплив шумових факторів.
Тест на акцент	Модель демонструє відчутні зміни у виявленні та адаптації до різних акцентів. Акцент майже не впливає на точність розпізнавання слів.
Тест на емоційне забарвлення	Моделью досягнуто істотні покращення в передачі емоцій. Виразність та розмаїття емоцій зростають і мовлення стає більш натуральним.
Тест на різні голоси	Різнороманітність голосів майже не впливає на точність розпізнавання.

Рисунок Б.3 – Епоха 30000

Епоха 40000	
Епоха	Результат
Тест на чіткість мовлення	Модель продовжує удосконалювати чіткість мовлення. Більшість дефектів, які залишилися, є мінімальними, і вимова стає вражаюче чистою та виразною.
Тест на швидкість мовлення	Розпізнавання швидкого темпу мовлення наближається до оптимального рівня. Модель більш точно та ефективно впоралася із завданням.
Тест на довгі фрази	Модель майже ідеально справляється з визначенням та відтворенням довгих текстів, зберігаючи повний контекст та уникаючи значних неточностей.
Тест на тишу	Модель практично бездоганно впорається з обробкою відсутності звуку, залишаючись стійкою до зовнішніх артефактів чи шуму у цих умовах.
Тест на шум	Майже неперевершена робота моделі з шумом. Голос ефективно ізольовано від фонового шуму і розпізнавання мовлення практично не порушується.
Тест на акцент	Чиста адаптація до різних акцентів. Модель ефективно і точно розпізнає та відтворює мовлення навіть з найбільш нестандартними акцентами.
Тест на емоційне забарвлення:	Майже ідеальне відтворення емоцій через мовлення. Модель точно передає різні емоційні відтінки з вражаючою натуральністю та виразністю.
Тест на різні голоси	Практично ідеальна адаптація до різних тембрів голосу. Модель точно розпізнає та адаптується до різноманітних характеристик голосу.

Рисунок Б.4 – Епоха 40000

Епоха 50000	
Епоха	Результат
Тест на чіпкість мовлення	Модель досягає очікуемого рівня у вимові та розпізнаванні тексту.
Тест на швидкість мовлення	Модель гарно працює з швидкими темпами мовлення.
Тест на довгі фрази	Модель ефективно обробляє найбільш об'ємні та складні фрази з високим ступенем точності та розуміння контексту.
Тест на тишу	Модель фактично ідеально реагує на відсутність звуку та виявляє мінімальні артефакти.
Тест на шум	Модель бездоганно адаптована до різних типів шумів та сторонніх звуків.
Тест на акцент	Модель чітко передає вимову слів, залишаючись високоточною.
Тест на емоційне забарвлення:	Модель досягає високого рівня у передачі емоцій через мовлення, виражаючи їх з великою точністю та емоційною виразністю.
Тест на різні голоси	Модель ідеально справляється з різними тембрами голосу, зберігаючи гарну дикцію.

Рисунок Б.5 – Епоха 50000

ДОДАТОК В
Слайди презентації

Національний університет “Запорізька політехніка”

Дипломна кваліфікаційна робота магістра
за темою:

«Дослідження та програмна реалізація навчання голосової
моделі за допомогою штучного інтелекту»

Виконав:
студент групи КНТ-222м

Пилип П’ЯТЕРНЯ

Керівник:

Ольга ГЛАДКОВА

Рисунок В.1 – Слайд 1

- ▶ Об’єкт дослідження – сучасні підходи та засоби для навчання голосової моделі.
- ▶ Предмет дослідження – дослідження та навчання голосової моделі з можливістю перетворення вхідного голосу на голос моделі.
- ▶ Мета роботи – розробка програмного забезпечення, яка навчає голосову модель за допомогою штучного інтелекту для можливості перетворення голосу.

2

Рисунок В.2 – Слайд 2

Завдання роботи

- ▶ проаналізувати існуючі аналоги голосових моделей;
- ▶ підготувати датасет та конфігурувати нейронну мережу для навчання голосової моделі;
- ▶ навчити голосову модель перетворювати вхідний голос на голос Порошенко П.О.

3

Рисунок В.3 – Слайд 3

Порівняння аналогів

Критерій	SVC-SO-VITS	Whisper	Vox2Vox
Може генерувати різні тембри	Так	Ні	Так
Адаптація до різних мов	Так	Так	Ні
Висока точність до оригіналу	Так	Ні	Так
Низький відсоток помилок	Так	Ні	Так
Швидкий час навчання	Ні	Ні	Так

4

Рисунок В.4 – Слайд 4

Вимоги до голосової моделі

- ▶ згенерований голос має бути максимально точним до оригіналу;
- ▶ голос моделі повинен яскраво передавати емоції спікера;
- ▶ голос моделі не повинен містити сторонніх звуків.

5

Рисунок В.5 – Слайд 5

Етапи навчання голосової моделі

- ▶ завантаження набору та розподіл даних;
- ▶ навчання голосової моделі;
- ▶ тести навченої моделі

6

Рисунок В.6 – Слайд 6

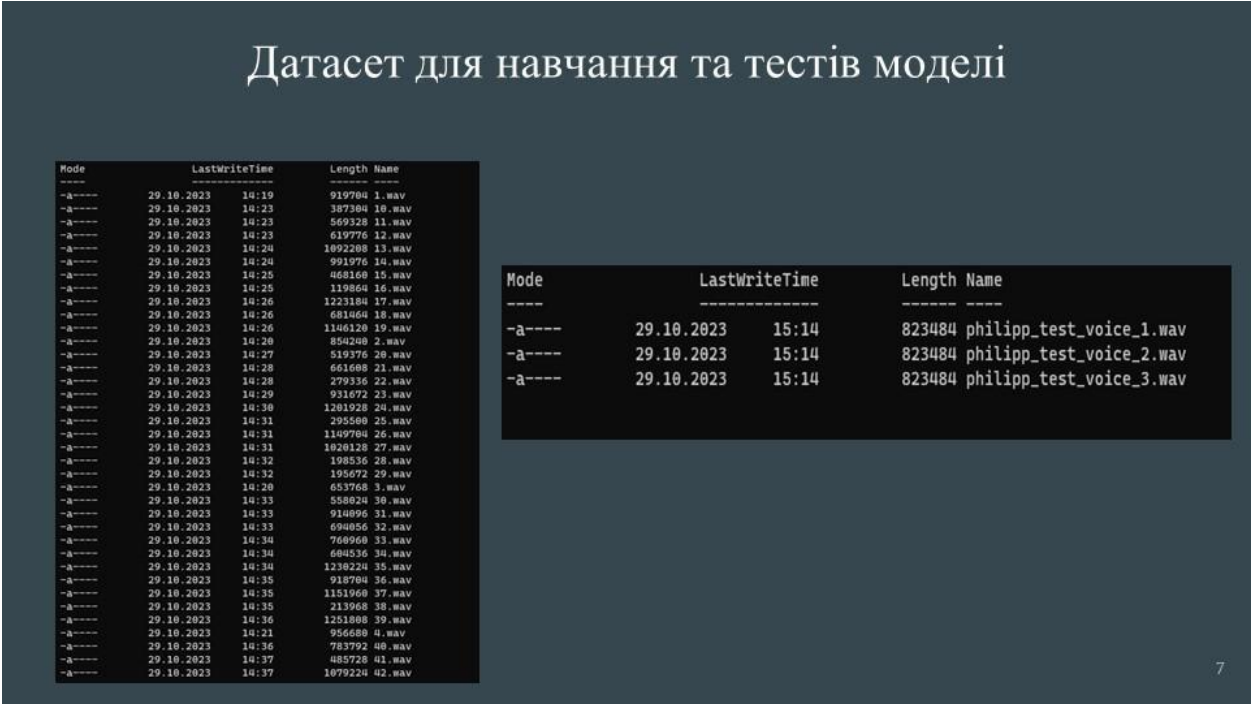


Рисунок В.7 – Слайд 7

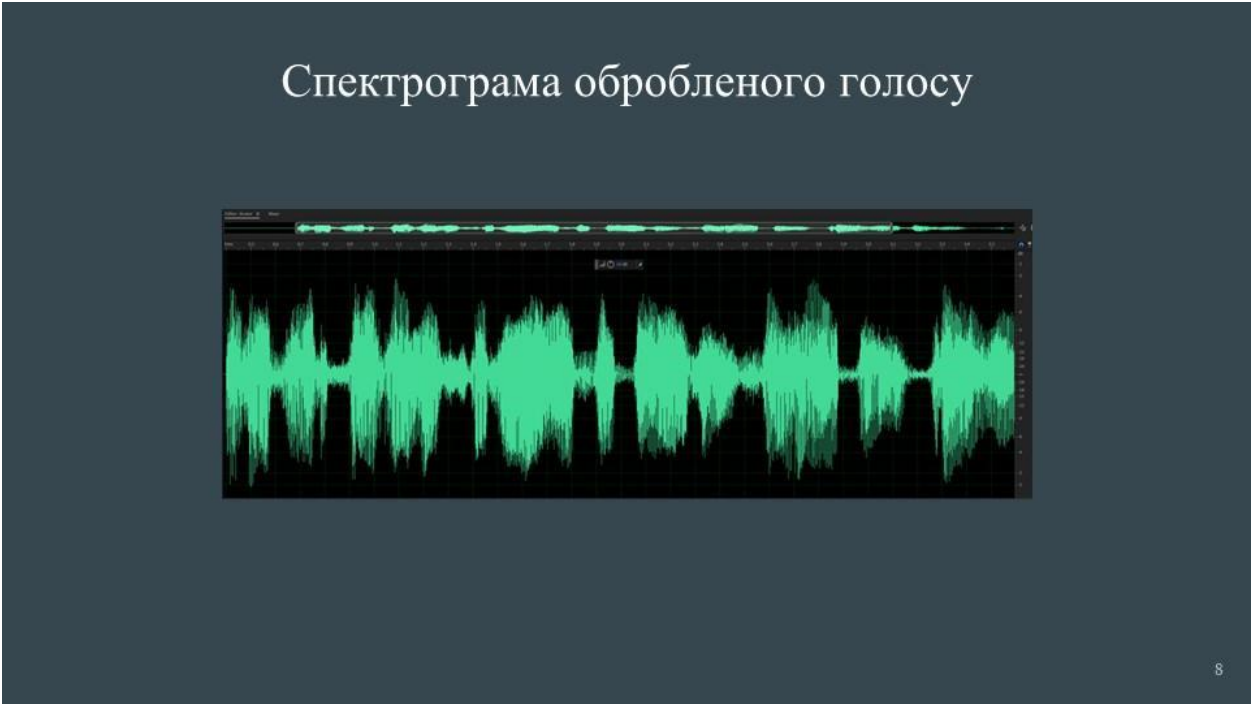


Рисунок В.8 – Слайд 8



Рисунок В.9 – Слайд 9

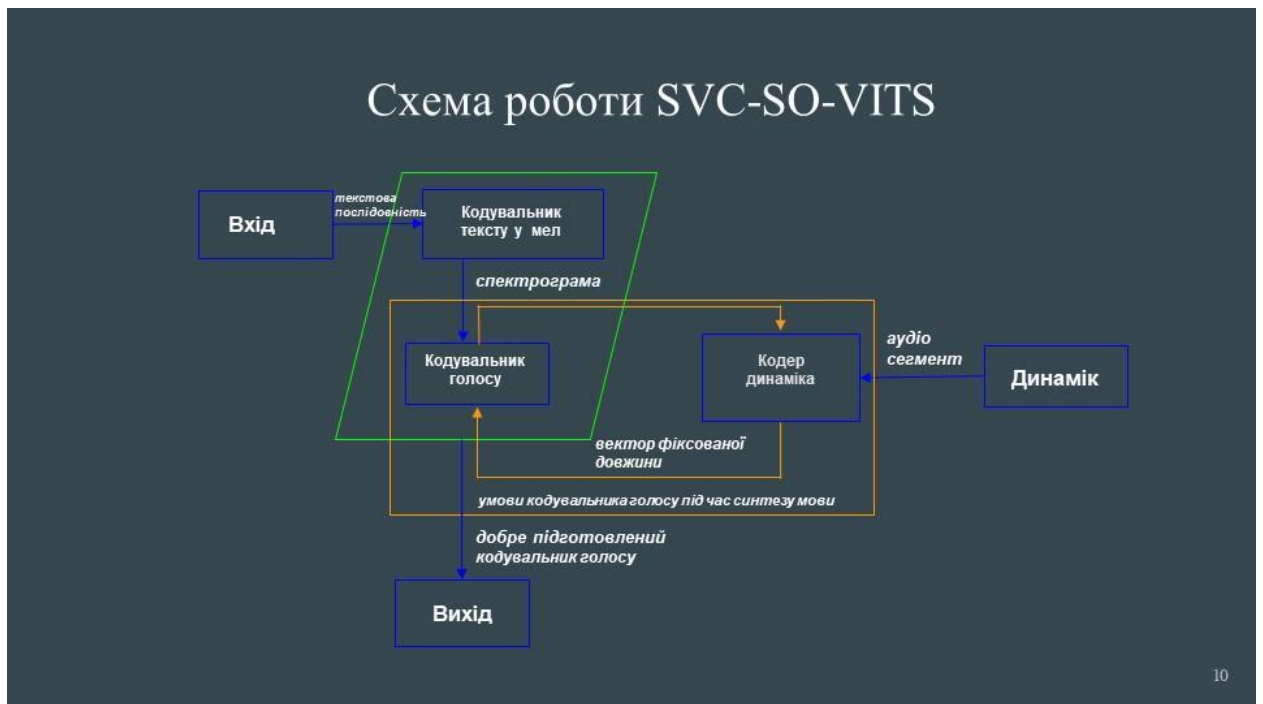


Рисунок В.10 – Слайд 10

Тести 10000 та 50000 епох	
Епоха 10000	
Епоха	Результат
Тест на чіткість мовлення	Модель виявляє помітне поліпшення у чіткості та точності вислову. Попередні дефекти значно зменшуються, і модель краще розпізнає та відтворює слова.
Тест на швидкість мовлення	Модель виявляє відсутнє порівняння у розпізнаванні швидкого темпу мовлення. Хоча ще є місце для вдосконалення, загальний рівень ефективності вже значно вищий.
Тест на довгі фрази	Модель вже здатна поліпшити обробку довгих фраз, але може втратити деякий контекст і допустити невеликі неточності в визначенні деяких слів.
Тест на тишу	Модель виявляє стабільність у розпізнаванні відсутності звуку із утриманим шумом, але може виявляти певні артефакти або шумові компоненти у відсутності аудіосигналу.
Тест на шум	Модель показує покращення в ізоляції голосу від фонових шумів, але ще може виявляти деякі труднощі при запізненні мовлення в умовах шуму.
Тест на акцент	Модель показує початкові зусилля в адаптації до різних акцентів. Є деякі випадки, коли розпізнавання може бути недовірливим у присутності акцентного мовлення.
Тест на емоційне забарвлення	Модель починає виявляти базову здатність передавати емоційне забарвлення, проте існують випадки, де відтворене мовлення може бути менш виразним або недостатньо емоційним.
Тест на різні голоси	Модель демонструє початкову здатність відрізняти та адаптуватися до різних тембрів голосу. Є невеликі аспекти, де розпізнавання може бути менш точним або менш адаптивним.
Епоха 50000	
Епоха	Результат
Тест на чіткість мовлення	Модель досягає очікуваного рівня у вислові та розпізнаванні тексту.
Тест на швидкість мовлення	Модель глибоко працює з швидкими темпами мовлення.
Тест на довгі фрази	Модель ефективно обробляє найбільш обсяжні та складні фрази з високим ступенем точності та розуміння контексту.
Тест на тишу	Модель фактично ідеально реагує на відсутність звуку та виявляє мінімальні артефакти.
Тест на шум	Модель бездоганно аудіювання до різних типів шумів та сторонніх звуків.
Тест на акцент	Модель чітко передає вимову слів, залишаючись високоточною.
Тест на емоційне забарвлення	Модель досягає високого рівня у передачі емоцій через мовлення, виражаючи їх з великою точністю та емоційною виразністю.
Тест на різні голоси	Модель ідеально справляється з різними тембрами голосу, зберігаючи гарну динаміку.

Рисунок В.11 – Слайд 11

Демонстрація навченої голосової моделі Порошенко П.О.

Вхідний аудіозапис



Оброблений моделью аудіозапис



Рисунок В.12 – Слайд 12

Висновки за роботою

- ▶ У роботі було проведено дослідження та програмна реалізація навчання голосової моделі за допомогою штучного інтелекту. Для цього було використано бібліотеку SVC-SO-VITS.
- ▶ В результаті отримуємо можливість клонування вхідного голосу на голос, яким була навчена модель, на високому рівні.
- ▶ Наукова новизна роботи полягає у застосуванні та навчанні голосової моделі SVC-SO-VITS. А також у використанні інноваційного підходу до навчання, який дозволяє досягти значного успіху в області клонування голосу людини. Цей напрям однозначно має потенціал для розвитку та подальших досліджень.

13

Рисунок В.13 – Слайд 13