

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до магістерської роботи
на освітньо-кваліфікаційний рівень магістр

на тему: **«РОЗРОБКА ТА НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ
РОЗПІЗНАВАННЯ КОМАНД КОРИСТУВАЧА НА БАЗІ СЕРВІСІВ
MICROSOFT»**

Виконав: студент 6 курсу, групи ПДМ-61 _____
спеціальності 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

Шевченко Д.О

(прізвище та ініціали)

Керівник _____

Щербина І.С.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ ТА
ІНФОРМАТИЗАЦІЇ

Факультет - Інформаційних технологій

Кафедра - Інженерії програмного забезпечення

Ступінь вищої освіти - «Магістер»

Спеціальність - 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

інженерії програмного забезпечення

_____ В.В. Онищенко

“ ____ ” _____ 2020 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Шевченко Дмитро Олександрович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка та навчання нейронної мережі для розпізнавання команд користувача на базі сервісів Microsoft»

Керівник роботи: Щербина Ірина Сергіївна к.т.н., доцент

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

затверджені наказом вищого навчального закладу від “ ____ ” __2019 року № _____

2. Строк подання студентом роботи _____

Бізнес-процес взаємодії із студентами центру дуальної _____ освіти, Microsoft
Dynamic CRM, мова програмування C#

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Визначення та функціональні можливості CRM систем.

2. Дослідження функціональних можливостей Microsoft Dynamnic CRM.

3. Розробка бізнес-процесу та архітектури CRM центру Дуальної Освіти навчального закладу.

4. Налаштування Microsoft Dynamics CRM та розробка власного Solution

5. Перелік графічного матеріалу

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	10.04-14.04.2019	
2	Визначення та функціональні можливості CRM систем	14.04-17.04.2019	
3	Дослідження функціональних можливостей Microsoft Dynamics CRM.	17.04-21.04.2019	
4	Розробка бізнес-процесу та архітектури CRM центру Дуальної Освіти навчального закладу.	21.04-23.04.2019	
5	Налаштування Microsoft Dynamics CRM та розробка власного Solution	23.04-10.05.2019	
6	Розробка обов'язкових демонстраційних матеріалів	19.05.2019	
7	Попередній захист роботи	13.05-17.05.2019	
8	Здача Здача роботи в деканат	24.05.2019	

Студент _____

(підпис)

(прізвище та ініціали)

Керівник роботи _____

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи 45 с., 27 рис., 30 джерел.

Об'єкт дослідження: Розпізнавання звукового сигналу (природної мови) за допомогою нейронних мереж.

Предмет дослідження: Методи і алгоритми нейронних мереж для розпізнавання природної.

Мета дослідження: Розробка програмного модуля AI для розпізнавання команд користувача.

Актуальність: Побудова системи голосового управління є в даний час актуальним завданням. Такі системи здатні істотно полегшити взаємодію користувача з комп'ютерною системою. Особливо ця ідея розвинена в концепції так званих «Розумних будинків».

Завдання дослідження, які дозволяють досягти поставленої мети:

1. Аналіз архітектур нейронних мереж.
2. Розгляд структури нейронної мережі.
3. Аналіз методів навчання нейронних мереж.
4. Аналіз алгоритму розпізнавання природної мови.
5. Аналіз сервісів Microsoft Azure AI.
6. Розробка архітектури програмного модуля.
7. Розробка модуля.

Стислий опис результатів дослідження: реалізовано програмний модуль AI для розпізнавання команд користувача на базі сервісів Microsoft Azure AI, що включає в себе такі функціональні можливості: розпізнавання мови, виділення смислових об'єктів, виконання команд, синтез мови.

Ключові слова: Архітектура, Нейронна мережа, Microsoft Azure AI, Модуль, Блок, ANN, AI.

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	8
ВСТУП	9
РОЗДІЛ 1. НЕЙРОННІ МЕРЕЖІ ТА ОБРАЗИ	10
1.1 Різновидності архітектур нейронних мереж	10
1.2 Структура нейронної мережі	15
1.3 Різновиди навчання нейронної мережі	18
1.4 Розпізнавання образів як наукова задача	19
1.5 Застосування нейронної мережі в розпізнаванні	25
РОЗДІЛ 2. ТЕОРІЯ РОЗПІЗНАВАННЯ ОБРАЗІВ	27
2.1 Мовний сигнал і його опис.	27
2.2 Штучний інтелект і природна мова	33
РОЗДІЛ 3. ПЛАТФОРМА MICROSOFT AZURE AI ДЛЯ РОЗРОБКИ НЕЙРОМЕРЕЖ	38
3.1 Когнітивні послуги	39
3.2 Сервіси Azure AI	40
РОЗДІЛ 4. РОЗРОБКА ТА НАВЧАННЯ ПРОГРАМНОГО МОДУЛЯ AI ДЛЯ РОЗПІЗНАВАННЯ КОМАНД КОРИСТУВАЧА	45
4.1 Постановка задачі	45
4.2 Архітектура	45
4.3 Розробка блоків Програмного модулю AI	46
ВИСНОВОК	55
ПЕРЕЛІК ПОСИЛАНЬ	56

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ANN – (artificial neural networks) штучна нейронна мережа;

AI – (Artificial Intelligence) Штучний інтелект;

Deep Learning – (Глибинне навчання; англ. Deep learning) - сукупність методів машинного навчання (з учителем, з частковим залученням вчителя, без учителя, з підкріпленням);

Microsoft Azure AI – Платформа компанії Microsoft для побудови програмних додатків з використанням нейронних мереж.

ВСТУП

З розвитком комп'ютерних систем стає все більш очевидним, що використання систем розпізнавання мови розшириться, якщо стане можливим використання людської мови при роботі безпосередньо з комп'ютером, і зокрема стане можливим управління машиною звичайним голосом в реальному часі, а також введення і виведення інформації у вигляді звичайної людської мови.

Одним з перспективних способів організації людино-машинного взаємодії є передача комп'ютерній системі інструкцій користувача в форматі мовних команд.

Голосовий інтерфейс є необхідним компонентом, коли мова йде про створення комфортних умов життя для людей з порушеннями опорно-рухового апарату. Такі системи з часом увійдуть в повсякденний побут в процесі реалізації концепції так званих «розумних будинків».

Об'єкт дослідження: Розпізнавання звукового сигналу (природної мови) за допомогою нейронних мереж.

Предмет дослідження: Методи і алгоритми нейронних мереж для розпізнавання природної.

Мета дослідження: Розробка програмного модуля AI для розпізнавання команд користувача.

Актуальність: Побудова системи голосового управління є в даний час актуальним завданням. Такі системи здатні істотно полегшити взаємодію користувача з комп'ютерною системою. Особливо ця ідея розвинена в концепції так званих «Розумних будинків».

РОЗДІЛ 1. НЕЙРОННІ МЕРЕЖІ ТА ОБРАЗИ

ANN (artificial neural networks) - це математична модель функціонування традиційних для живих організмів нейромереж, які представляють собою мережі нервових клітин. Як і в біологічному аналозі, в штучних мережах основним елементом виступають нейрони, з'єднані між собою і утворюють шари, число яких може бути різним у залежності від складності нейромережі і її призначення.

1.1 Різновидності архітектур нейронних мереж

Нейронна мережа - це математична модель у вигляді програмного і апаратного втілення, що будується на біологічних принципах функціонування нейронів мозку. Виділяються кілька різних архітектур штучних нейронних систем.

1.1.1 Повнозв'язна нейронна мережа

Повнозв'язна нейронна мережа (Fully connected neural network, FCNN) повністю складається з ряду повністю зв'язаних шарів, які з'єднують кожен нейрон в одному шарі з кожним нейроном в іншому шарі. Кожен нейрон передає свій вихідний сигнал решті нейронів, включаючи самого себе. Вихідними сигналами мережі можуть бути всі вихідні сигнали нейронів. Елементи шаруватих і повнозв'язаних мереж можуть вибиратися по-різному. Існує, втім, стандартний вибір: нейрон з адаптивним неоднорідним лінійним суматором на вході, схема повнозв'язної нейронної мережі продемонстрована (рис 1.1).

Основною перевагою повністю зв'язаних мереж є те, що вони є “агностичними за структурою”, тобто немає особливих припущень щодо вхідних даних.

Для повнозв'язаної мережі вхідний суматор нейрона фактично розпадається на два: перший обчислює лінійну функцію від вхідних сигналів мережі, другий — лінійну функцію від вихідних сигналів інших нейронів, отриманих на попередньому кроці. Функція активації нейронів (характеристична функція) — це нелінійний перетворювач вихідного сигналу суматора. Якщо функція одна для всіх нейронів мережі, то мережу називають однорідною (гомогенною). Якщо ж

характеристична функція залежить ще від одного або декількох параметрів, значення яких змінюються від нейрона до нейрона, то мережу називають неоднорідною [1].

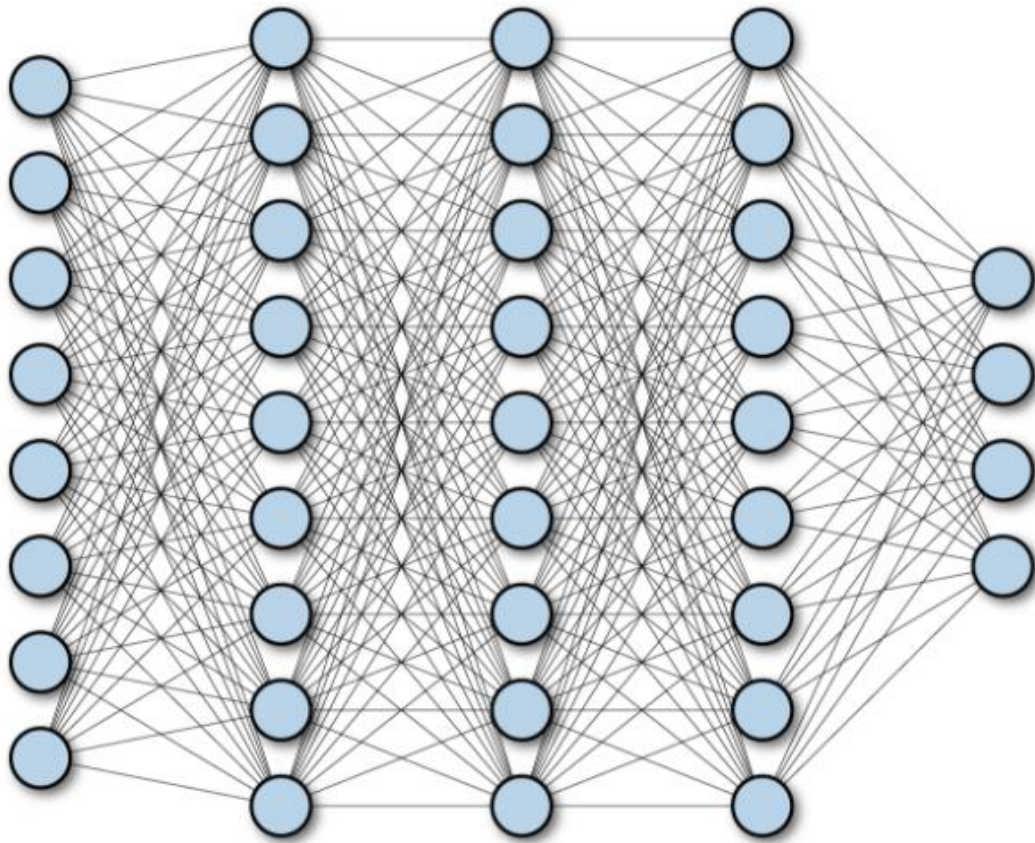


Рисунок 1.1 – Схема повнозв'язної нейронної мережі

Нейронні мережі можуть мати декілька прихованих шарів. Складати мережу з нейронів стандартного вигляду необов'язково. Шарувата або повнозв'язана архітектура не накладає істотних обмежень на елементи, що беруть участь в них. Єдина жорстка вимога, яка висувається архітектурою до елементів мережі, це відповідність розмірності вектора вхідних сигналів елемента (вона визначається архітектурою) числу його входів.

Незважаючи на те, що агностична структура робить повністю підключені мережі дуже широко застосовними, такі мережі, як правило, мають слабку ефективність, ніж мережі спеціального призначення, налаштовані на структуру проблемного простору.

1.1.2 Згорткова нейронна мережа

Згорткова нейронна мережа (Convolutional neural network, CNN) має спеціальну архітектуру, яка дозволяє їй максимально ефективно розпізнавати образи. Ідея CNN чітко висловлюють припущення, що вхідні дані - це зображення, що дозволяє кодувати певні властивості в архітектурі моделі (рис 1.2).

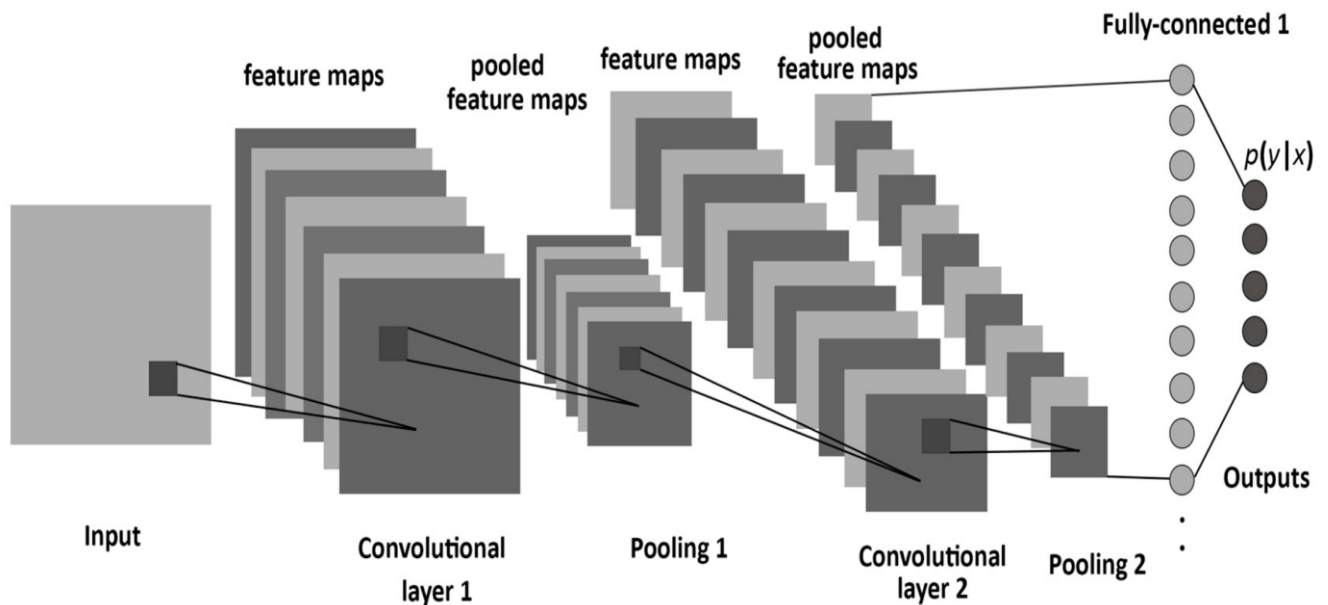


Рисунок 1.2 - Схема згорткової нейромережі

Простий CNN - це послідовність шарів, і кожен шар CNN перетворює один обсяг активацій в інший через диференційовану функцію. Для побудови архітектури CNN використовуються три основні типи шарів: згортковий шар, шар об'єднання та повністю з'єднаний шар.

Модель CNN ґрунтується на чергуванні згорткових і субдискретизуючих шарів (pooling), а структура є односпрямованою. CNN отримала свою назву від операції згортки, яка передбачає, що кожен фрагмент зображення буде помножений на ядро згортки поелементно, при цьому отриманий результат повинен додаватися і записатися в схожу позицію вихідного зображення.

Така архітектура забезпечує інваріантність розпізнавання щодо зсуву об'єкта, поступово збільшуючи «вікно», на яке «дивиться» згортка, виявляючи все більш і більш великі структури і патерни в зображенні[2]

1.1.3 Рекурсивна нейронна мережа

Рекурсивні нейронні мережі (Recursive neural network, RvNN) - вид нейронних мереж, що працюють з даними змінної довжини. Моделі рекурсивних мереж використовують ієрархічні структури зразків при навчанні. Наприклад, зображення, складені з сцен, які об'єднують підсцени, що включають багато об'єктів. Виявлення структури сцени і її деконструкція- нетривіальне завдання. При цьому необхідно як ідентифікувати окремі об'єкти, так і всю структуру сцени (рис 1.3).

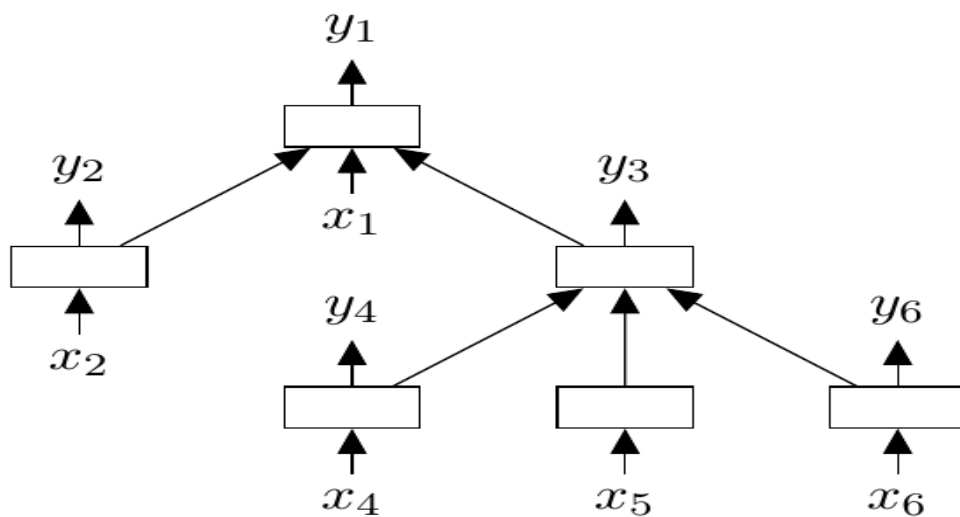


Рисунок 1.3 - Схема рекурсивної нейронної мережі

У рекурсивних мережах нейрони з однаковими вагами активуються рекурсивно відповідно до структури мережі. В процесі роботи рекурсивної мережі виробляється модель для передбачення для структур змінної розмірності, так і скалярних структур через активацію структури відповідно до топології. Мережі RvNNs успішно застосовуються при навчанні послідовних структур і дерев в задачах обробки природної мови, при цьому фрази і пропозиції моделюються через векторне подання слів. RvNNs спочатку з'явилися для розподіленого представлення структур, використовуючи предикати математичної логіки[3].

МДКП. Мережа довгої короткостроковій пам'яті - вид рекурсивної НС, що дозволяє максимально точно моделювати тимчасові послідовності, а також характерні для них залежно в довгостроковій перспективі.

1.1.4 Генеративно-змагальна мережа

Генеративно-змагальна мережа (**Generative Adversarial Networks, GAN**) - це алгоритм машинного навчання без учителя, побудований на комбінації з двох нейронних мереж, одна з яких (мережа G) генерує зразки (Генеративна модель), а інша (мережа D) намагається відрізнити правильні («справжні») зразки від неправильних [4] (рис 1.4).

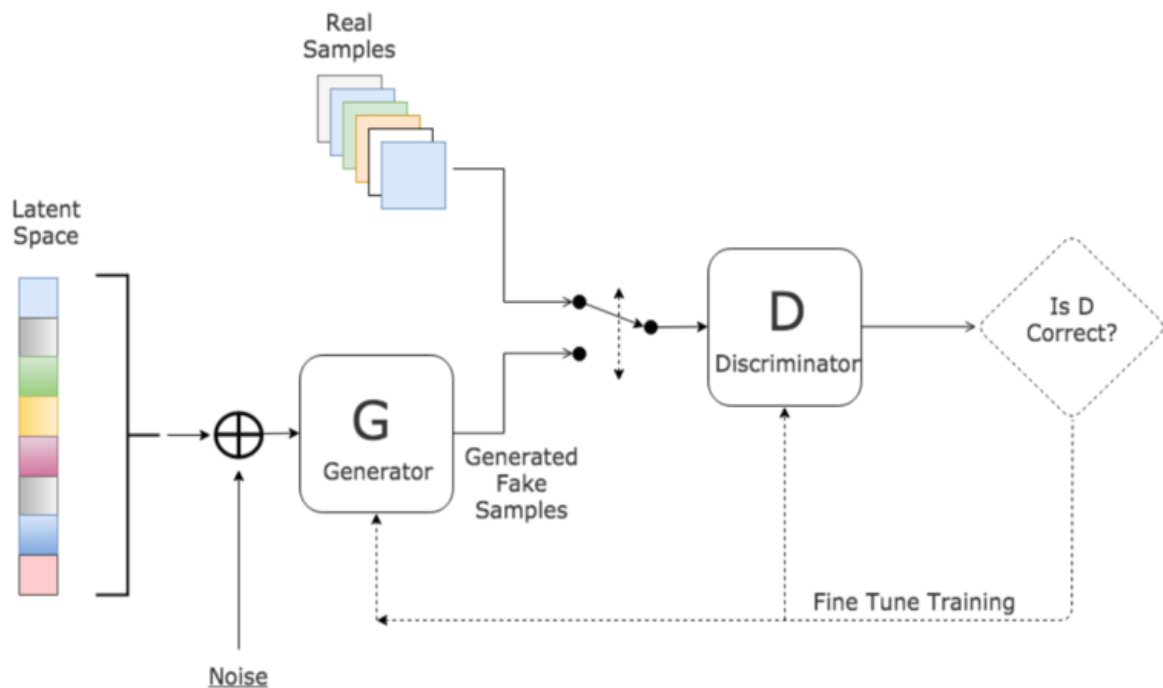


Рисунок 1.4 - Схема генеративно-змагальна мережі

Базовий GAN складається з наступних елементів:

1. Генеративна модель (тобто генератор) генерує об'єкт. Генератор нічого не знає про реальних об'єктах і вчиться, взаємодіючи з дискримінатором. Наприклад, генератор може генерувати зображення.
2. Дискримінаційна модель (тобто Дискримінатор) визначає, чи є об'єкт реальним (зазвичай представлений значенням, близьким до 1) або підробленим (представлений значенням, близьким до 0).
3. Дискримінатор подає генератору сигнал змагальної втрати (або помилки) таким чином, що він дозволяє генератору генерувати об'єкти, максимально наближені до реальних об'єктів.

1.2 Структура нейронної мережі

Штучний нейрон, також відомий, є основною одиницею нейронної мережі (рис 1.5). Простіше кажучи, це математична функція, заснована на моделі біологічних нейронів. Це також можна розглядати як простий логічний затвор з двійковими виходами.

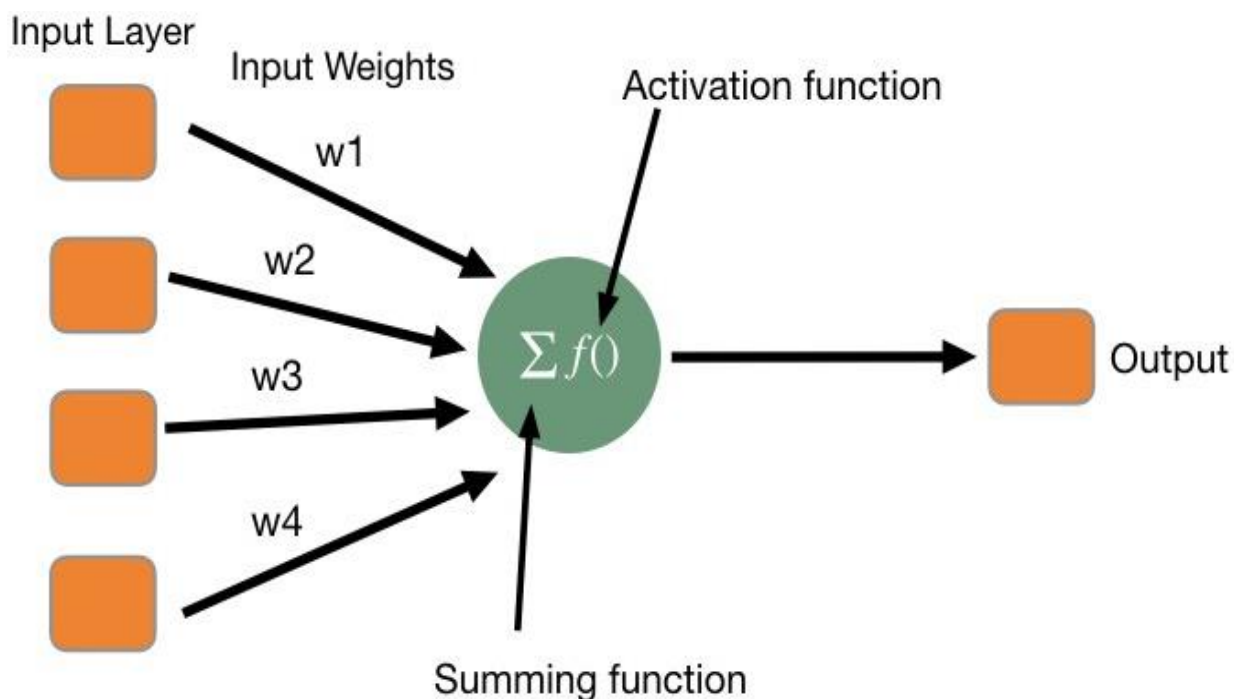


Рисунок 1.5 - Діаграма штучного нейрону

Кожен штучний нейрон виконує такі основні функції:

1. Бере вхідні дані з рівня введення
2. Зважує їх окремо і підсумовує
3. Передайте цю суму через нелінійну функцію, щоб отримати результат.

Штучний нейрон складається з 4 частин:

1. Вхідні значення або Один вхідний шар
2. Ваги та зміщення
3. Функція активації
4. Вихідний рівень

Вхідні значення нейрону, використовуючи вхідний шар. Це може бути щось таке просте, як колекція значень масиву. Він схожий на дендрит в біологічних нейронах.

Функція активації - функції активації використовуються для введення нелінійності в нейронні мережі. Він стискає значення в меншому діапазоні, а саме. функція активації сигмоїдів стискає значення в діапазоні від 0 до 1 [5]. Є багато функцій активації, що використовуються в галузі глибокого навчання. У якості активаційної Функції зазвичай використовують сигмоїдну функцію (1.1):

$$f(z) = \frac{1}{1 + \exp(-z)} \quad (1.1)$$

де z - змінна, яка вказує клас об'єкта.

Вхідний рівень - це перший рівень у нейронній мережі. Він приймає вхідні сигнали (значення) і передає їх наступному шару. Він не застосовує жодних операцій з вхідними сигналами (значеннями) і не має пов'язаних значень ваги та зміщення.

Вхідна форма - це форма вхідної матриці, яку ми передаємо вхідному шару. Вхідний рівень нашої мережі має 4 нейрони, і він очікує 4 значення 1 вибірки.

Вузол - ієрархічно з'єднані в мережами, де вихід одних нейронів є входом для інших нейронів. Такі нейрони можна представити у виде з'єднаних шарів у вузли. Кожен вузол приймає зваження на вхід, активує активаційну функцію для суми входів, та генерує вихід [6]. Іншими словами, зважений вхід у вузол можна відобразити у вигляді формули (1.2):

$$x_1w_1 + x_2w_2 + x_3w_3 + b \quad (1.2)$$

де $w(1,2,3)$ - числові значення ваги,

b - вагою елемента зміщення.

З'єднання - Він з'єднує один нейрон в одному шарі з іншим нейроном в іншому шарі або тому ж шарі. З'єднання завжди має значення ваги. Метою тренінгу є оновлення цього значення ваги для зменшення втрат (помилки).

Ваги (параметри) - Вага представляє міцність зв'язку між одиницями. Якщо вага від вузла 1 до вузла 2 має більшу величину, це означає, що нейрон 1 має більший вплив на нейрон 2. Вага знижує важливість вхідного значення. Вага біля нуля означає, що зміна цього входу не призведе до зміни вихідного сигналу. Негативні ваги означають, що збільшення цього вкладу зменшить вихід. Вага вирішує, який вплив вхід матиме на вихід.

Ваги є значеннями, які будуть змінюватись протягом процесу навчання, зміщення на +1, включення ваги b робить вузол гнучкішим [6]. Вивід активаційної функції представлений через " h ". За вагу беруться числа (не бінарні), які потім множаться на вході і сумуються у вузлі (рис 1.6).

Упередження (зміщення) - це додатковий вхід до нейронів, який завжди дорівнює 1, і має власну вагу з'єднання. Це гарантує, що навіть коли всі входи відсутні (всі 0), у нейроні відбудеться активація.

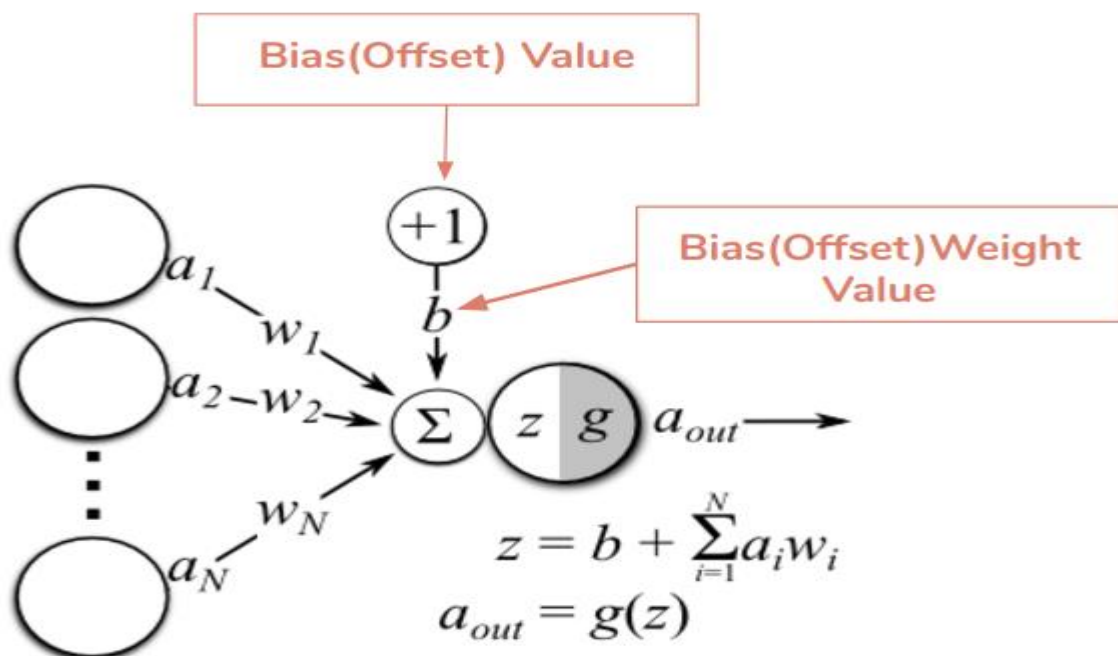


Рисунок 2.6 – Зміщення та вага штучного нейрону

Приховані шари - приховані шари мають нейрони (вузли), які застосовують різні перетворення до вхідних даних. Один прихований шар - це сукупність нейронів, розташованих вертикально (подання) [1].

1.3 Різновиди навчання нейронної мережі

Розпізнавання образів за допомогою нейронних мереж можливо тільки за допомогою спеціального навчання, що представляє собою процес, спрямований на настройку параметрів нейронної мережі. Способи навчання нейронних мереж приведені нижче.

1.3.1 Машинне навчання з вчителем

При навчанні нейронної мережі для розпізнавання образів з учителем є вибірка з істинними відповідями на питання, що зображено на картинці - мітками класів. Нейромережі подаються на вхід ці інформація, після чого обчислюється помилка, що порівнює вихідні значення з істинними мітками класів. Залежно від ступеня і характеру невідповідності передбачення нейронної мережі, її ваги коректуються, відповіді нейронної мережі підлаштовуються під достеменні відповіді, поки помилка не стане мінімальною[7].

1.3.2 Навчання без вчителя

В цьому випадку у навчальній вибірці немає міток класів, і перед нейронної мережі стоїть завдання знайти заздалегідь не відомі відповіді. Нейронна мережа намагається самостійно знайти закономірності в даних, витягуючи корисні ознаки і аналізуючи їх. Наприклад, кластеризація - найбільш поширена завдання для навчання без учителя. Алгоритм підбирає схожі дані, знаходячи спільні ознаки, і групують їх разом.

У навчанні без учителя складно обчислити точність алгоритму, тому що в цих відсутні «правильні відповіді» або мітки. Але розмічені дані буває складно або занадто дорого отримати. У таких випадках, надаючи моделі свободу дій для пошуку залежностей, можна отримати певний результат[7].

1.3.3 Навчання з частковим залученням вчителя

Навчальна вибірка містить як розмічені, так і розділеного дані. Цей метод особливо корисний, коли розмітити всі об'єкти - трудомістке завдання. Проте, нейронна мережа може отримати інформацію з невеликої частки розмічених даних

і поліпшити точність прогнозів в порівнянні з моделлю, яка навчається виключно на нерозмічену даних [7].

1.3.4 Навчання з підкріпленням

Навчання з підкріпленням (англ. Reinforcement learning) - один із способів машинного навчання, в ході якого випробувана система (агент) навчається, взаємодіючи з деякою середовищем. З точки зору кібернетики, є одним з видів кібернетичного експерименту. Відгуком середовища (а не спеціальної системи управління підкріпленням, як це відбувається в навчанні з учителем) на прийняті рішення є сигнали підкріплення, тому таке навчання є окремим випадком навчання з учителем, але учителем є середовище або її модель. Також потрібно мати на увазі, що деякі правила підкріплення базуються на неявних вчителів, наприклад, в разі штучної нейронної середовища, на одночасної активності формальних нейронів, через що їх можна віднести до навчання без учителя.

1.4 Розпізнавання образів як наукова задача

Здатність до розпізнавання є важливою особливістю усіх живих організмів. Людина усвідомлює структуру навколишнього світу, визначає форму та розміри кожного об'єкту. Розуміє природу звуків та відрізняє їх один від одного, будь то якась мова чи звук падаючого роялю. Дивлячись на відео запис, може легко підрахувати (і назвати) всіх людей у зображенні, здогадатися їх емоції за зовнішнім виглядом, розпізнати фразу та смисловий посил кожної персони на записі. Вже десятиліття вчені намагаються зрозуміти принципи функціонування візуальної система та моделюється достовірного голосового тракту, але ще остаточне рішення цієї загадки залишається незрозумілим.

Як відомо, в процесі набуття досвіду, прийняття рішень у правій півкулі головного мозку людини формуються когнітивні образи, які разом із методами їх розпізнавання утворюють базу знань. При цьому ліва півкуля виконує функції операційної системи. Таким чином, як доведено за результатами

експериментальних досліджень будь-яка діяльність біонічної системи здійснюється на основі образного мислення.

Розпізнавання образів є однією з актуальних задач теоретичної інформатики. Розпізнавання образів - це наукова дисципліна, метою якої є класифікація об'єктів по декількох категоріях або класами. Об'єкти називаються образами. В рамках цієї загальної задачі виділяють завдання розпізнавання зорових образів (зображень) і слухових образів (мови) і т. д.

1.4.1 Основні поняття задачі розпізнавання

Розпізнавання образів (об'єктів, сигналів, ситуацій, явищ або процесів) – це завдання ідентифікації об'єкта або визначення будь яких його властивостей або за його зображенням, аудіозаписом або іншими характеристиками. Одним із базових понять теорії є поняття множини. У комп'ютері множина представляється набором унікальних однотипних елементів. Унікальних означає, що якийсь елемент у множині або існує, або його там немає. Універсальна множина містить усі можливі для розв'язуваної проблеми елементи, порожня множина не містить жодного.

Образ – це класифікаційне угруповання у системі, яка виділяє певну групу об'єктів за конкретною характеристикою. Образи мають характерну ознаку, що виявляється у процесі ознайомлення із скінченним числом явищ однієї фіксованої множини [8]. Це надає можливість отримувати інформацію про яку завгодно кількість її представників.

Кожне відображення будь якого об'єкта на сприймаючі органи, які їх розпізнають, незалежно від його локалізації відносно цих органів, називають зображенням об'єкта, а множини таких зображень, об'єднаних серією загальних властивостей, є образом [9].

Стан – це певної форми відображення вимірюваних поточних (або миттєвих) характеристик об'єкту спостереження. Сукупність станів визначає *ситуацію*.

Ситуацією прийнято називати деяку сукупність станів складного об'єкта, кожна з яких характеризується одними і тими ж або схожими характеристиками об'єкта. Наприклад, якщо об'єктом спостереження розглядають деякий об'єкт

керування, то ситуація об'єднує такі стани цього об'єкта, у яких слід застосовувати одні і ті ж керуючі впливи.

Методика віднесення елемента до визначеного образу називається *вирішальним правилом*.

Метрика - спосіб визначення відстані між елементами універсальної множини. Чим меншою є відстань, тим більш подібними є між собою об'єкти (символи, сигнали та ін.), що ми розпізнаємо. Звичайно елементи задаються у вигляді набору чисел, а метрика – у вигляді функції [10].

Від виду представлення образів і способу реалізації метрики залежить ефективність програми, один і той же алгоритм розпізнавання але з різними метриками буде помилятися з різною частотою. Навчанням називатимемо процес вироблення в деякій скінченній системі реакції на групи зовнішніх однакових сигналів шляхом багатократного впливу на систему зовнішнього коригування. Таке зовнішнє коригування у процесі навчання прийнято називати "*покараннями*". Механізм генерації коригування є повністю визначальним для алгоритму навчання.

Самонавчання є частинним випадком навчання, але відрізняється від останнього тим, що тут додаткова інформація про правильність реакції у системі не повідомляється.

Адаптація – це процес зміни структури і параметрів системи, а можливо – і керуючих впливів, на основі інформації, що поступає з метою досягнення необхідного стану системи при початковій невизначеності і мінливих умовах роботи [11].

Процес розпізнавання образів, можна розділити на три етапи: формування вихідного опису, знаходження системи ознак і побудова вирішального правила. Існують два формулювання завдання розпізнавання: у вузькому і широкому сенсах [12]. У вузькому сенсі завдання розпізнавання зводиться до побудови класифікатора, в широкому сенсі - до розпізнавання в умовах невизначеності (в даному випадку не відомі безліч ознак і безліч класів).

1.4.2 Характеристика методів розпізнавання

Задачі розпізнавання мають характерні риси. Процес виконання складається з двох етапів: – перетворення вхідних даних до вигляду, зручного для розпізнавання; власне розрізнення (числовий параметр, як вказівка про належність об'єкту до певному класу).

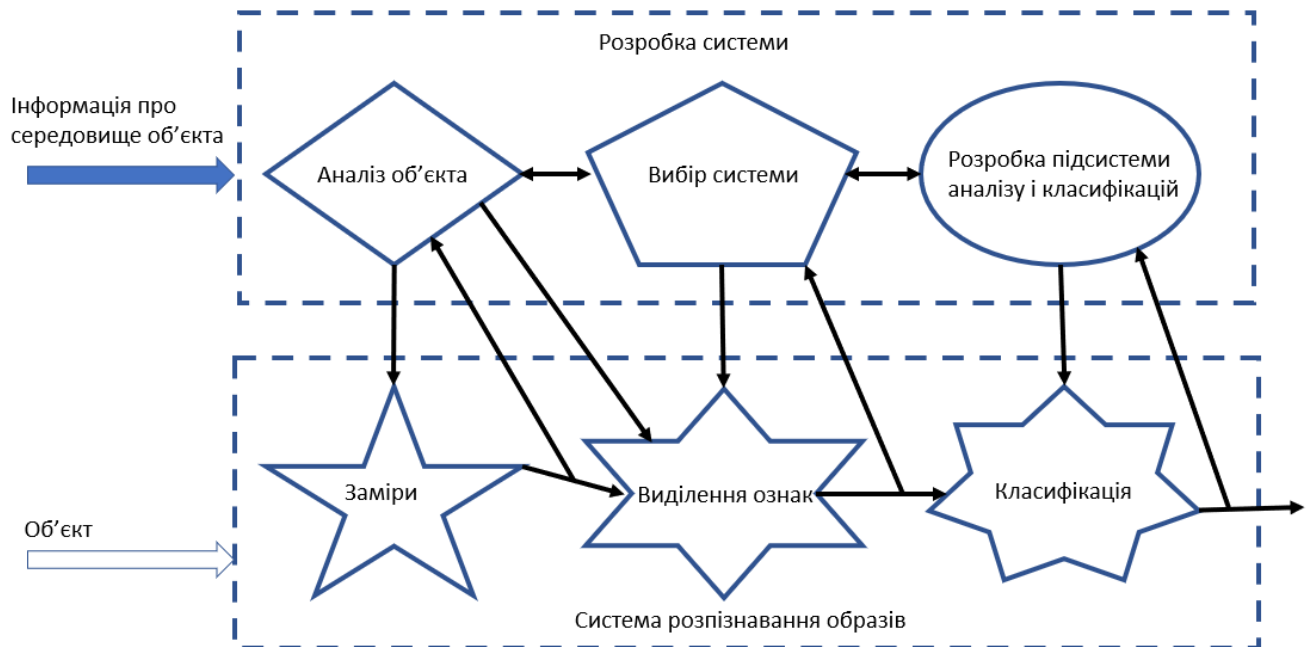


Рисунок 1.7 - Структура системи розпізнавання

У таких задачах можна вводити поняття аналогії або подібності об'єктів і формулювати правила, на підставі яких об'єкт відносять в один і той же клас або в різні класи (рис 1.7). У таких проблемах можна оперувати набором прецедентів (еталонів), класифікація яких встановлена і які у вигляді формалізованих описів представлені алгоритму розпізнавання для регулювання завдання [13]. Для таких завдань нелегко будувати формалізовані теорії і застосовувати традиційні математичні методи. Часто недоступною є інформація для точної математичної моделі, або вигреш від використання такої моделі і математичних методів несумірний з витратами. Виділяють кілька типів задач розпізнавання:

1. Завдання розрізнення – віднесення представленого об'єкту по його опису до деякого із заданих класів.

2. Автоматична класифікація – розбиття безлічі об’єктів, ситуацій, явищ по їх описах на систему класів, що не перетинаються, наприклад: таксономія, кластерний аналіз, самонавчання.
3. Завдання вибору інформативної множини ознак при розпізнаванні.
4. Завдання приведення вхідних даних до вигляду, зручного для розрізнення.
5. Динамічне розпізнавання та динамічна класифікація.
6. Завдання прогнозування – відноситься до динамічного типу, у якому рішення повинне відноситися до деякого моменту у майбутньому.

Для сприйняття образу необхідно виділити множину релевантних інформативних ознак та, вимірявши їх значення, сформувати вектор-реалізацію образу. При цьому на практиці, як правило, виникає необхідність використання 14 додаткових процедур попередньої обробки, фільтрації та відновлення образів.

Важливу роль під час розв’язання задачі інформаційного синтезу відіграє апріорна інформація про об’єкти розпізнавання, необхідна для формування алфавіту класів розпізнавання, словника ознак і навчальної матриці. Крім того, для побудови високодостовірних вирішальних правил на етапі формування вхідного математичного опису системи розпізнавання необхідно здійснювати так званий розвідувальний аналіз із метою забезпечення статистичної стійкості й однорідності реалізацій образу [14].

Методи розпізнавання поділяють на дві основні групи:

1. Методи, які для формування вирішальних правил використовують класифіковану навчальну матрицю (методи навчання з учителем).
2. Методи автоматичної класифікації, здатні обробляти некласифіковані дані (методи навчання без учителя). В середині кожної з цих основних груп виконується подальший поділ методів на підгрупи залежно від апріорної інформації, необхідної для їх застосування. Розглянемо в рамках геометричного підходу узагальнену постановку задачі інформаційного синтезу системи розпізнавання образів, здатної навчатися [15].

1.4.3 Проблеми існуючих методів розпізнавання образів

1. Проблема формування вихідного опису, пов'язана з тим, що існуючі моделі і методи розпізнавання адаптовані до конкретного класу прикладних задач і вимагають апріорного знання властивостей аналізованих сигналів [16];
2. Проблема формування системи ознак, пов'язана з вибором кінцевого безлічі ознак, що забезпечують однозначність рішення завдання класифікації на етапі розпізнавання і відповідає вимогам необхідності і достатності. Етап вибору системи ознак необхідний для скорочення розмірності вхідного опису. З огляду на, що завдання скорочення розмірності - оптимізаційна задача, то для її рішення необхідно використання критерію інформативності. Відсутність моделі апріорної невизначеності і моделі її розкриття породило велику кількість методів у виборі критерію інформативності, що, в свою чергу, породило велику кількість можливих варіантів ознак [16];
3. Проблема прийняття рішень в умовах апріорної невизначеності. Етап прийняття рішення полягає в порівнянні з наявним еталоном признакового опису аналізованого сигналу. Передбачається, що еталону відповідає компактне безліч точок в системі ознак. Однак перешкоди, структурні зміни одного і того ж представника класу призводять до перекриття класів. Тому проблема прийняття рішення замикається на проблеми формування системи ознак, що дозволяє сформувати еталон, який має компактне уявлення [16].

1.4.4 Механізм рішення завдання нейронної мережі при розпізнаванні образів

Як образи для розпізнавання можуть виступати найрізноманітніші об'єкти, включаючи зображення, рукописний або друкований текст, звуки і багато іншого. При навчанні мережі їй пропонуються різні зразки з міткою того, до якого саме типу їх можна віднести. Як зразок застосовується вектор значень ознак, а сукупність ознак в цих умовах повинна дозволити однозначно визначити, з яким класом образів має справу НС.

Важливо при навчанні навчити мережу визначати не тільки достатню кількість і значення ознак, щоб видавати хорошу точність, але і не перенавчитися, тобто, надмірно не «підлаштуватися» під навчальну вибірку. Після завершення правильного навчання НС повинна вміти визначати образи (тих же класів), з якими вона не мала справи в процесі навчання.

Важливо враховувати, що вихідні дані для нейронної мережі повинні бути однозначні і несуперечливі, щоб не виникали ситуації, коли НС буде видавати високі ймовірності приналежності одного об'єкта до декількох класів. В цілому створення нейронної мережі для розпізнавання зображень включає в себе:

- збір та підготовку даних;
- вибір топології;
- підбір характеристик;
- підбір параметрів навчання;
- навчання;
- перевірку якості навчання;
- коригування;
- вербалізацію.

1.5 Застосування нейронної мережі в розпізнаванні

Робота з зображеннями, рукописним або друкований текстом, звуками - важлива сфера застосування технологій Deep Learning. Глобально всі дані світу створюють бібліотеку неструктурованих даних. Задіявши нейронної мережі, машинне навчання і штучний інтелект, ці дані структурують і використовують для виконання різних завдань: побутових, соціальних, професійних і державних, зокрема, забезпечення безпеки.

1.5.1 Теорія

Основою всіх архітектур є аналіз, першою фазою якого буде розпізнавання об'єкта. Потім штучний інтелект за допомогою машинного навчання розпізнає дії і класифікує його.

Для того щоб розпізнати об'єкт, нейронна мережа повинна бути перш за навчена на даних. Це дуже схоже на нейронні зв'язки в людському мозку - ми володіємо певними знаннями, бачимо об'єкт, аналізуємо його і ідентифікуємо.

Нейронної мережі вимогливі до розміру і якості датасета, на якому вона буде навчатися.

1.5.2 Практика

На практиці означає, що до певної межі чим більше прихованих шарів в нейронної мережі, тим точніше буде розпізнано зображення. Як це реалізується?

Об'єкт розбивається на маленькі ділянки, кожна з яких буде вхідним нейроном. За допомогою синапсів сигнали передаються від одного шару до іншого. Під час цього процесу сотні тисяч нейронів з мільйонами параметрів порівнюють отримані сигнали з уже обробленими даними.

Простіше кажучи, якщо ми просимо машину розпізнати фотографію кішки, ми розіб'ємо фото на маленькі шматочки і будемо порівнювати ці шари з мільйонами вже наявних зображень кішок, значення ознак яких мережа вивчила. У якийсь момент збільшення числа шарів призводить до просто запам'ятовування вибірки, а не навчання.

РОЗДІЛ 2. ТЕОРІЯ РОЗПІЗНАВАННЯ ОБРАЗІВ

2.1 Мовний сигнал і його опис.

Мова є носієм інформації, що використовується людиною для передачі повідомлень - сигналом. За фізичну природу це акустичний сигнал, безперервно змінюється в часі.

Більшість сигналів (мовних в тому числі) мають аналогову природу, тому для обробки їх на цифрових комп'ютерах вони перетворюються в дискретні сигнали за допомогою аналого-цифрового перетворення (АЦП). За допомогою цієї процедури отримують набір відліків $s[n]$ - знятих в моменти $\Delta t \cdot n$ миттєвих значень безперервного сигналу, які вже позбавлені фізичної природи, а їх максимальне і мінімальне значення задається розрядністю АЦП.

Для опису та перетворення дискретних сигналів застосовні засоби цифрової обробки сигналів (ЦОС). Найважливішою процедурою ЦГЗ є дискретне перетворення Фур'є (ДПФ). ДПФ дозволяє перейти з тимчасової області в частотну, тобто розкласти $s[n]$ на набір гармонік і знайти залежність амплітуди (енергії) гармоніки від її частоти (рис. 2.1).

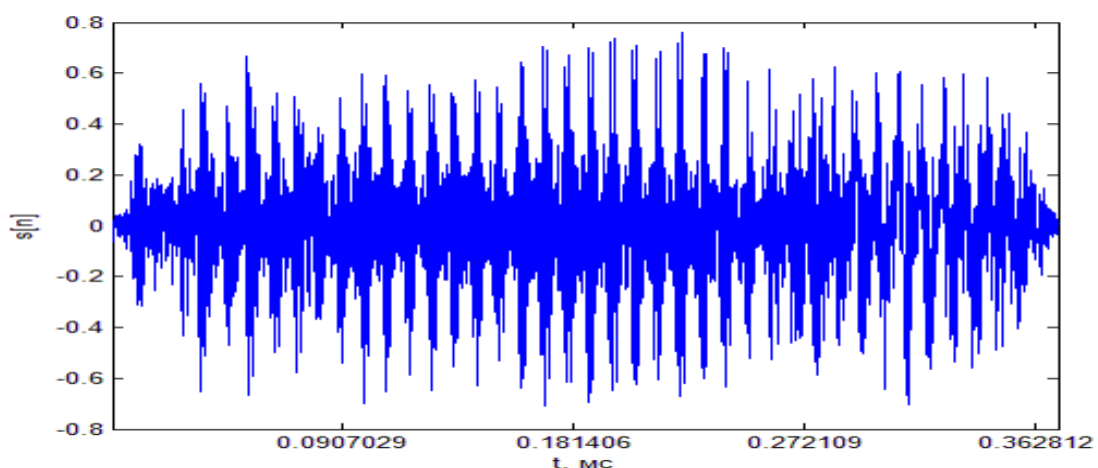


Рисунок 2.1 - Ділянка мови з голосним звуком «а»

Мова є нестационарним сигналом, тобто його характеристики змінюються в часі. Можна наочно зобразити ці зміни, побудувавши графіки модулів ДПФ для йдуть підряд фрагментів (фреймів) мовного сигналу.

2.1.1 Загальна структура системи автоматичного розпізнавання мови

У роботі системи автоматичного розпізнавання мови виділяють три етапи: виділення ознак, навчання і розпізнавання (рис. 2.2).

На першому етапі з вихідного сигналу отримують вектор ознак - стислий опису мовного сигналу, в якому присутня тільки значуща для розпізнавання інформація. Для цього використовуються методи, що працюють як в частотній області (крейда-кепстральних коефіцієнти, коефіцієнти лінійного передбачення), так і в тимчасовій (наприклад, на короткочасному значенні енергії) [17].

Послідовність векторів ознак довжиною T називають акустичної або спостерігається послідовністю $O = (o_1, o_2, \dots, o_T)$. З допомогою цієї послідовності людина передає ланцюжок слів $W = (w_1, w_2, \dots, w_N)$. Сама задача розпізнавання мови ставиться таким чином: необхідно відшукати ланцюжок слів W , яка відповідає акустичної послідовності O [18].

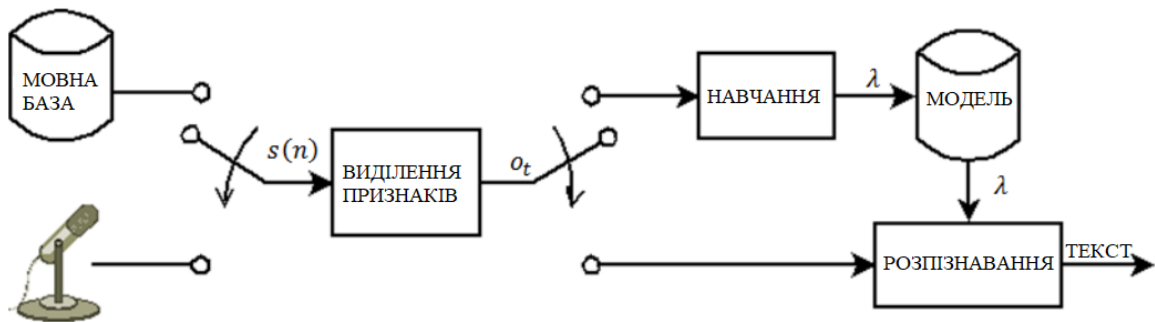


Рисунок 2.2 - Загальна схема системи автоматичного розпізнавання мови.

Для вирішення цього завдання на етапі навчання складається модель λ , яка здатна породжувати всі можливі послідовності O для всіх можливих W . Нехай функція $h(W, \lambda)$ повертає всі можливі O тільки для заданої W . Тоді розпізнаванням буде знаходження такого ланцюжка слів W^* , яка, відповідно до моделі λ , породить акустичну послідовність, найбільш близьку до даної O (2.1):

$$W^* = \text{ArgMin}_{W \in W} d(h(W, \lambda), O) \quad (2.1)$$

де $d(O', O)$ – дистанція між O' та O .

Таким чином, потрібно перевірити всі ланцюжки слів W , що, природно, недосяжно на практиці. Для полегшення цього завдання вводять різні обмеження за допомогою граматики мови або вирішується більш вузька завдання, наприклад, розпізнавання тільки ізольованих слів.

2.1.2 Блок виділення ознак

Завдання блоку виділення ознак - скласти ланцюжок векторів ознак $O = (o_1, o_2, \dots, o_T)$ вихідного сигналу. Блок виділення ознак сканує вхідний сигнал, в межах якого і складається один вектор ознак (рис. 2.3).

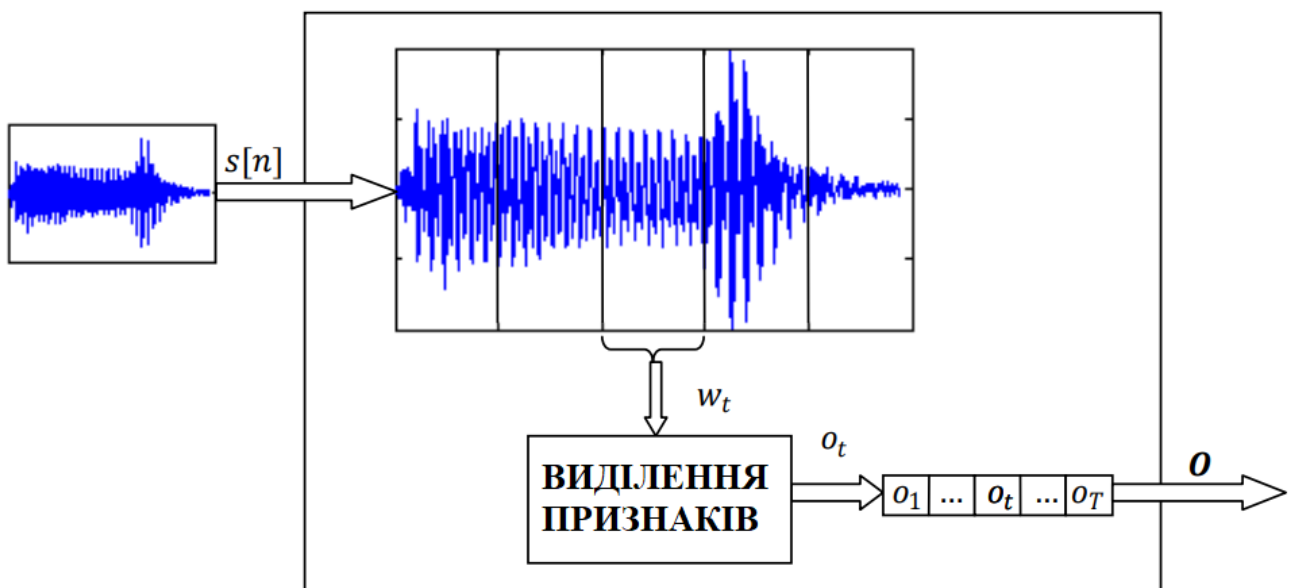


Рисунок 2.3 - Виділення ознак. Розбиття вхідного сигналу на ланцюжок

Найкращим чином мова представляється ознаками, до таких ознак відносяться (Linear Predictive Codes – LPC), (Perceptual Linear Prediction - PLP), (Mel-Frequency Cepstral Coefficients - MFCC) [19] Ці ознаки ґрунтуються на акустичній моделі відтворення речі, згідно з якою мовний сигнал можна представити у вигляді сигналу на виході лінійної системи з змінними в часі параметрами.

Оцінка параметрів цієї лінійної системи і є завдання методів знаходження ознак в частотній області. Основною ідеєю методів лінійного передбачення є

можливість апроксимації поточного відліку мовного сигналу за допомогою лінійної комбінації попередніх відліків (2.2) [20]:

$$s(n) = \sum_{k=1}^{\rho} \alpha_k s(n-k) \quad (2.2)$$

де $\{\alpha_k\}$ - коефіцієнти лінійного передбачення.

Передавальна функція лінійної системи, в яку входять збудження і мовний сигнал, описується наступним чином:

$$H(z) = \frac{G}{1 - \sum_{k=1}^{\rho} \alpha_k z^{-k}} \quad (2.3)$$

де G - коефіцієнт посилення збудження.

Таким чином, визначення параметрів α_k дозволить оцінити спектральні властивості мовного сигналу.

2.1.3 Блок розпізнавання

На етапі розпізнавання робота ведеться з послідовністю векторів ознак довжиною T $O = (o_1, o_2, \dots, o_T)$, з допомогою якої передається ланцюжок слів $W = (w_1, w_2, \dots, w_N)$. Використовуючи загальну для розпізнавання образів термінологію, ланцюжок O називається чином - областю в просторі ознак [21].

Для розпізнавання розглянутої O' Необхідно за допомогою моделі мови λ , в якій встановлено зв'язок між усіма можливими O і всіма можливими W , знайти такий ланцюжок слів W^* , Для якої λ породить послідовність ознак, найбільш близьку до O' (2.1).

Головна проблема розпізнавання мови полягає в тому, як скласти модель λ . Виділяють дві процедури роботи САРР:

1. Навчання, коли налаштовуються параметри моделі на навчальній вибірці, що репрезентує собою безліч пар O', W' . чим більша вибірка, тим адекватніше вийде себе модель.
2. Розпізнавання, коли перевіряються всі ланцюжки слів W і вибирається та, чия акустична послідовність W , λ найближче до розглянутої.

2.1.4 Метод порівняння з еталоном

Для кожного слова складається модель-еталон проголошення O' , щоб на етапі розпізнавання вибрати ту модель, еталон якої найближче до розглянутої акустичної послідовності O . Головна проблема методів цієї групи полягає в тому, що мовні образи сильно варіюються по тривалості, отже необхідний спосіб порівнювати образи різної довжини.

Єдиний представник групи – метод динамічного вирівнювання часу (Dynamic Time Warping - DTW) [22]. У ньому проблема різниці довжини еталона і розглянутого способу вирішується таким шляхом: складається матриця C розміром $M \times N$ (2.4).

$$\begin{aligned}
 C_{1,1} &= D_{1,1} \\
 C_{i,1} &= D_{i,1} + C_{i-1,1}, i = 2, \dots, M \\
 C_{1,j} &= D_{1,j} + C_{1,j-1}, j = 2, \dots, N \\
 C_{i,j} &= D_{i,j} + \min(C_{i-1,j}, C_{i-1,j-1}, C_{i,j-1}), i = 2, \dots, M; j = 2, \dots, N
 \end{aligned}
 \tag{2.4}$$

де N - довжина зразка,

M - довжина даної послідовності.

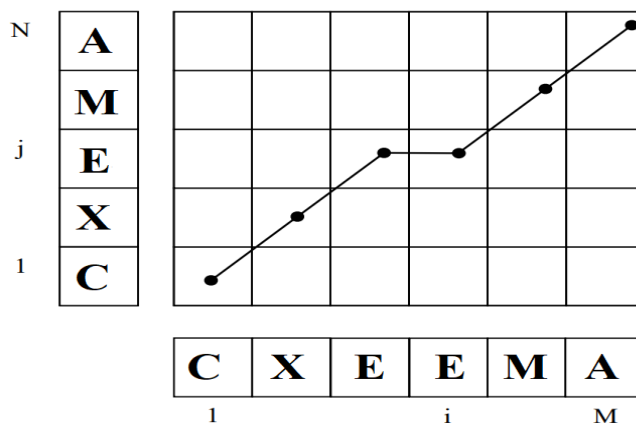


Рисунок 2.4 - Приклад порівняння образу з еталоном методом динамічного вирівнювання часу.

Недоліком методу динамічного вирівнювання часу є труднощі, що виникають при складанні еталона, викликані сильною варіативністю мови. Крім цього за допомогою DTW складно організувати розпізнавання злитого мовлення.

2.1.5 Знаходження значень короткочасної енергії і частоти проходів через нуль

Існуючі методи виділення мови з сигналу $s(n)$ працюють у тимчасовій області. До них відносяться, знаходження короткочасної енергії сигналу $E_s(m)$, короткочасної потужності сигналу $P_s(m)$ і частоти проходів через нуль $Z_s(m)$ [20]:

$$Z_s(m) = \frac{1}{L} \sum_{n=m-L+1}^m \frac{|\operatorname{sgn}(s(n)) - \operatorname{sgn}(s(n-1))|}{2} \quad (2.5)$$

Ці характеристики обчислюються для m -го ділянки (блоку) довжини L сигналу $s(n)$. Видно, що потужність, фактично, еквівалентна енергії сигналу, віднесеної до довжини блоку. Вирішальне правило VAD (Voice Activation Detection) - це порогова функція, де враховують всі вищеописані характеристики (2.6):

$$\begin{aligned} VAD(m) &= \begin{cases} 1, & W_s(m) \geq t_w \\ 0, & W_s(m) < t_w \end{cases} \\ W_s(m) &= P_s(m) \cdot (1 - Z_s(m)) \cdot S_c \\ t_w &= \mu_w + \alpha \delta_w \\ \alpha &= 0.2 \cdot \delta_w^{-0.8} \end{aligned} \quad (2.6)$$

де t_w виступає як поріг який складається з μ_w ,

μ_w - середнього значення шумових відліків,

δ_w – відхилення значень шумових відліків від середнього.

Недоліком цього методу є припущення про те, що мова звучить голосніше, ніж шум, що, звичайно, не завжди так. Якщо у вхідному сигналі зустрінуться ділянки, скажімо, з досить гучним шумом гортання сторінок або іншим елестінням, значення $P_s(m)$ буде порівнянно з тим, яке воно приймає для ділянок, що містять мова. також недоліком можна вважати необхідність ретельного експериментального підбору масштабуючих коефіцієнтів. Для усунення зазначених недоліків був розроблений підхід на основі аналізу розподілу локальних екстремумів.

2.2 Штучний інтелект і природна мова

Однією із основних ліній робіт з нейромережами є завданням оптимізації спільної роботи "штучного інтелекту" і власне інтелектуальних можливостей людини. Це завдання тісно пов'язане з лінгвістикою, психологією.

Забезпечення взаємодії з ЕОМ природною мовою (ПМ) є найважливішим завданням досліджень у штучному інтелекті. У той час як багато фундаментальних проблем в області обробки ПМ ще не вирішені, прикладні системи можуть оснащуватися інтерфейсом, що розуміє ПМ при певних обмеженнях.

Існують два види і, отже, дві концепції обробки природної мови:

1. Для окремих речень.
2. Для ведення інтерактивного діалогу.

2.2.1 Основна проблема обробки природної мови

Основною проблемою обробки ПМ є мовна неоднозначність. Існують найрізноманітніші види неоднозначності: синтаксична (структурна), значеннєва неоднозначність, відмінкова неоднозначність і т.д.

Центральна проблема, як для загальної, так і для прикладної обробки ПМ – розв'язання такого роду неоднозначностей - вирішується за допомогою перекладу зовнішнього подання ПМ в якусь внутрішню структуру [23]. Для загальної обробки ПМ таке перетворення вимагає набору знань про реальний світ.

Прикладні системи обробки ПМ мають переваги перед загальними, тому що працюють у вузьких предметних областях. Проте, створення систем, що мають можливість спілкування на ПМ в широких областях, можливо, хоча поки результати далекі від задовільних.

На основі аналізу досліджень в області людино-машинної комунікації можна виділити наступні ключові проблеми:

1) Головна проблема полягає в створенні методів і алгоритмів сталого (robust) розуміння мови. Тут хорошим орієнтиром можуть служити механізми сприйняття людини, які ігнорують багато видів мовних неточностей і виробляють «точний сенс з неточних слів».

2) Створення моделей інтеграції всіх основних видів інформації в ході процесу розуміння мови.

3) Проблема розумній мірі повноти вхідної інформації. Традиційні методи семантичного розкладу зазвичай спираються на ідею «розуміння є здатність відповісти на всі питання, пов'язані з вхідним висловлюванням». Цей принцип зазвичай зводиться до необхідності скрупульозного заповнення всіх слотів семантичного фрейму, що веде до ускладнення модуля управління діалогом, до перепитав і до необгрунтованого уповільнення ходу діалогу.

4) Удосконалення модулів розпізнавання слів в частині більш швидкої адаптації до голосу, акустичної обстановці і конкретної прикладної задачі.

2.2.2 Розпізнавання природної мови

У міру розвитку комп'ютерних систем стає все більше очевидним, що використання цих систем набагато розшириться, якщо стане можливим використання людської мови при роботі безпосередньо з комп'ютером, і зокрема стане можливим керування машиною звичайним голосом у реальному часі, а також уведення й висновок інформації у вигляді звичайної людської мови.

Існуючі технології розпізнавання мови не мають поки достатніх можливостей для їхнього широкого використання, але на даному етапі досліджень проводиться інтенсивний пошук можливостей уживання коротких багатозначних слів (процедур) для полегшення розуміння.

Розпізнавання мови в цей час знайшло реальне застосування в житті, мабуть, тільки в тих випадках, коли використовувався словник скорочений до 10 знаків, наприклад при обробці номерів кредитних карт і інших кодів доступу до базованих на комп'ютерах систем, що обробляють передані по телефону дані. Так що насущне завдання - розпізнавання, принаймні, 20 тисяч слів природної мови - залишається поки недосяжним. Ці можливості поки недоступні для широкого комерційного використання. Однак ряд компаній самотужки намагається використати вже існуючі в даній галузі науки знання.

Для успішного розпізнавання мови варто вирішити наступні завдання [23]:

- Обробку словника (фонемний склад);

- обробку синтаксису;
- скорочення мови (включаючи можливе використання твердих сценаріїв);
- вибір диктора (включаючи вік, стать, рідну мову й діалект), тренування дикторів;
- вибір особливого виду мікрофона;
- умови роботи системи й одержання результату із вказівкою помилок.

Існуючі сьогодні системи розпізнавання мови ґрунтуються на зборі всієї доступної (часом навіть надлишкової) інформації, необхідної для розпізнавання слів. Дослідники вважають, що в такий спосіб завдання розпізнавання зразка мови, засноване на якості сигналу, підданого змінам, буде достатнім для розпізнавання, але, проте, у цей час навіть при розпізнаванні невеликих повідомлень нормальної мови, поки неможливо після одержання різноманітних реальних сигналів здійснити пряму трансформацію в лінгвістичні символи, що є бажаним результатом.

Для будь-яких методів розпізнавання мовних сигналів основним є встановлення міри подібності виділеного (на основі попередньої сегментації [1]) фрагмента сигналу з деяким сигналом, прийнятим за еталон, або з сукупністю еталонних сигналів та віднесення на основі цієї міри сигналу до відповідного класу.

Нижче розглянуто деякі міри подібності, які використовуються в методах та алгоритмах розпізнавання мовних сигналів, запропоновано алгоритм порівняння та проведено порівняння окремих мір щодо їх ефективності для задач розпізнавання мовних сигналів. Крім того, ефективність використання міри подібності визначається також обсягом обчислювальних затрат, оскільки переважно оброблення та розпізнавання мовних сигналів повинні здійснюватися в реальному часі.

2.2.3 Існуючі міри подібності мовних сигналів та їх порівняння

Розглядаються два фрагменти мовних сигналів, один з яких $x[i]$, $i = \overline{1, n}$ отриманий за попередньої сегментації мовного потоку, а інший $y[i]$ є деяким еталоном, з яким порівнюється виділений сигнал. Для забезпечення

можливості порівняння сигнали попередньо нормуються в такий спосіб, щоб їх питома енергія була однаковою і дорівнювала 1, (2.7).

$$\sum_{i=1}^n x^2[i] = \sum_{i=1}^n y^2[i] = 1 \quad (2.7)$$

Як міри подібності можуть бути використані як універсальні міри, придатні для сигналів різних типів, так і міри, які переважно використовуються під час оброблення та порівнянні мовних сигналів завдяки певним характерним особливостям цих сигналів [4]. Розглянемо деякі з них:

Сума квадратів відхилень - ця міра визначається співвідношенням (2.8) і знаходиться в межах: (2.9). Внаслідок коливального характеру мовних сигналів ця міра є ефективною лише у тому разі, якщо частоти основного тону порівнюваних сигналів збігаються і проводиться попередня обробка “вирівнювання фаз” сигналів.

$$\Delta_1 = \sum_{i=1}^n (x[i] - y[i])^2 \quad (2.8)$$

$$0 \leq \Delta_1 \leq 2 \quad (2.9)$$

Коефіцієнт кореляції Визначається співвідношенням (2.10). Аналогічно до попереднього випадку ефективна лише за таких самих умов.

$$\Delta_2 = \sum_{i=1}^n x[i] \cdot y[i] \quad (2.10)$$

Спектральна міра будується на основі коефіцієнта кореляції між модулями спектрів сигналів з попереднім використанням віконної функції, наприклад, функції Хеммінга. Використання цієї міри пов’язане з моделями формування мовних сигналів, насамперед сигналів, що відповідають голосним звукам, спектри яких мають формантну структуру. При цьому фазові співвідношення сигналів не відіграють істотної ролі.

Спектрально-часова невизначеність Визначається як добуток тривалості сигналу на ефективну ширину його спектра: $\Delta t \cdot \Delta f$. Ця міра є ефективною лише для фрагментів малої тривалості, насамперед таким, що відповідають голосним

звукам (один квазіперіод сигналу), і в деяких випадках є доповненням до спектральної міри.

Міра на основі кепстральних коефіцієнтів Ця міра визначається гомоморфною обробкою сигналів, є специфічною для мовних сигналів внаслідок врахування моделей їх формування і широко використовується в системах розпізнавання мовних сигналів на основі нейромережових технологій [24].

Міра Махаланобіса Дає змогу визначити міру подібності фрагмента мовного сигналу до сукупності еталонних сигналів, зокрема, з врахуванням можливості побудови множин різних еталонів для однакових елементів мовних сигналів [25]. Нехай вектори μ_i та μ_j визначають середні значення векторів X_i та X_j відповідно, тобто $\mu_i = E[x_i]$, де E – оператор математичного сподівання. Для вимірювання відстані між двома множинами можна використати відстань Махаланобіса, квадрат якої визначається співвідношенням (2.11).

$$d_{ij}^2 = (x_i - \mu_i)^T \Sigma^{-1} (x_j - \mu_j) \quad (2.11)$$

де Σ^{-1} – матриця, обернена до матриці коваріації.

Припускається, що матриця коваріації для обох множин одна й та сама (2.12).

$$\Sigma = E[(x_i - \mu_i)(x_i - \mu_i)^T] = E[(x_j - \mu_j)(x_j - \mu_j)^T] \quad (2.12)$$

У частковому випадку, коли $x_i = x_j, \mu_i = \mu_j = \mu$, де I – одинична матриця, відстань Махаланобіса вироджується в Евклідову відстань між вектором X_i і вектором математичного сподівання μ .

РОЗДІЛ 3. ПЛАТФОРМА MICROSOFT AZURE AI ДЛЯ РОЗРОБКИ НЕЙРОМЕРЕЖ

Платформа Microsoft AI працює на Microsoft Azure середовищі хмарних обчислень, яке забезпечує обчислення як послугу, плата стягується за те, що використовується на даний час. Платформа дозволяє розробникам створювати рішення для штучного інтелекту ефективно та економічно.

AI Platform - це гнучкий, відкритий, набір послуг, інструментів з інфраструктурою корпоративного рівня, яка дозволяє розробникам максимально продуктивно розробляти рішення AI (рис. 3.1). Розробка рішення для глибокого навчання вимагає великої кількості експериментів, великих обчислювальних потужностей часто з використанням передових апаратних засобів, таких як GPUs і FPGAs.

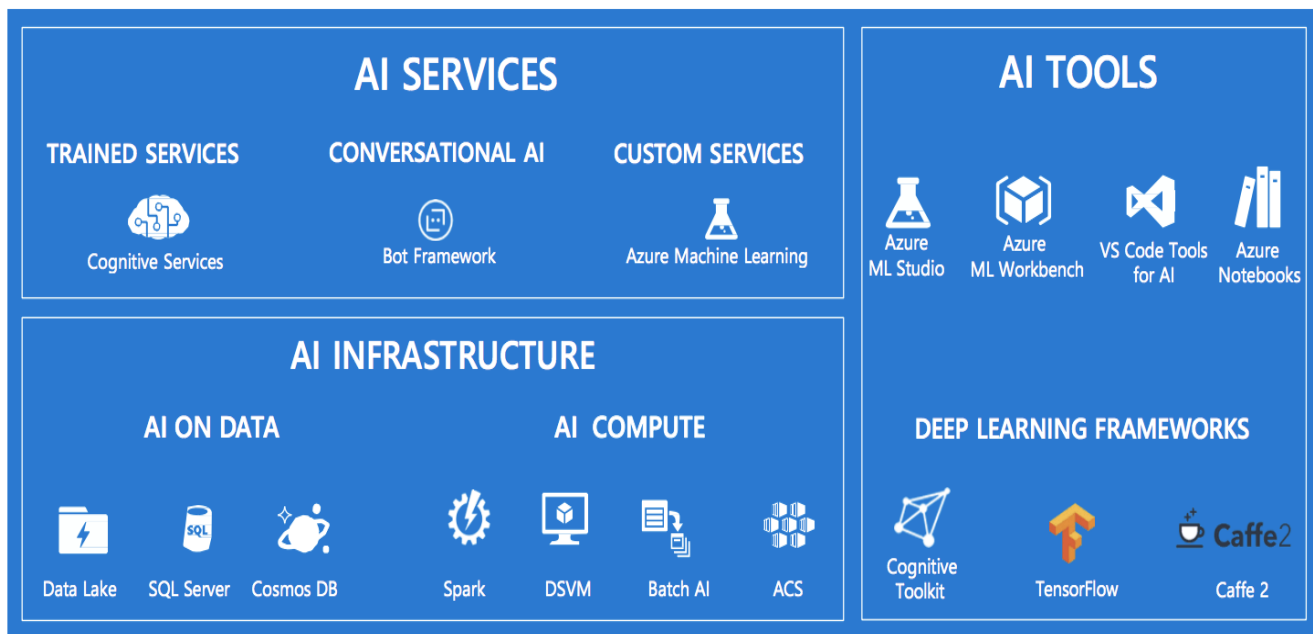


Рисунок 3.1 – Інфраструктура Microsoft Azure AI

Звичайно, не всі ці сервіси необхідні для одного конкретного рішення, але, скоріше, взяті разом, забезпечують платформу, на якій може бути побудовано будь-яке інтелектуальне додаток, що використовує кращі технології з відкритим вихідним кодом, інфраструктура та інструменти.

3.1 Когнітивні послуги

Когнітивні сервіси-це набір сервісів, доступних розробникам і дослідникам даних для створення AI рішень, з можливостями, пов'язаними із зором, мовою, мовою, знаннями та пошуком (таблиця 3.1). Ці сервіси популярні, так як їх легко додавати в додатки. Когнітивні послуги бувають декількох основних типів:

1. Попередньо навчені моделі, доступні у вигляді REST API, готові до використання в додатках кінцевих користувачів без будь-якого налаштування.
2. Кастомні сервіси які потребують тільки данні користувача (Bring-your-own-data services). Наприклад Vision Service, який дозволяє розробнику створити користувацьку модель класифікації зображень просто завантаживши зображення різних класів і натиснувши кнопку для навчання моделі.
3. Сервіси які як модулі можуть використовуватися в кастомному додатку користувача. Додаток також може знаходитися на платформі Azure та користуватися всіма її бенефітами як: *Knowledge exploration*, *Web language model*, *Video Indexer* і тд. Повний список продемонстровано в таблиці 3.1.

Таблиця 3.1 - Приклади пізнавальних служб, доступних у Microsoft AI

Vision	Language	Speech	Search	Knowledge
Computer vision	Text analytics	Speaker recognition	Web search	Academic knowledge
Face recognition	Spell check	Speech	Image search	Entity linking service
Emotion	Web language model	Speech Service	Video search	Knowledge exploration
Content Moderator	Linguistic analysis		News search	Recommendations
Video Indexer	Translator		Autosuggest	QnA maker
Vision Servicea	Language Understanding		Search	Decision Service

3.2 Сервіси Azure AI

Платформа Microsoft AI складається з цілої низки сервісів, від повністю кероване програмне забезпечення до служб для створення спеціальних додатків AI. Залежно від сценарію та необхідної гнучкості, може бути застосовними різні рішення. Послуги розбиті на три основні сфери:

1. Попередньо побудований AI. Ці інструменти використовують заздалегідь побудовані моделі додаток за допомогою вже наявних алгоритмів побудований для того, щоб бачити, чути, говорити та розуміти за допомогою Когнітивні послуги [26].
2. Розмовний AI. Будують взаємодію природного характеру через додаток через Bot Framework, який має доступ до загальних каналів, таких як Facebook Messenger, Slack, Skype та Bing [26].
3. Спеціальні послуги AI. Вони адаптуються до сценарію з гнучкість служб машинного навчання Azure, Пакетна послуга AI або те й інше [26].

Сервіси Azure корисні для створення користувальницьких рішень AI і допомагають прискорити розробку інтелектуальних додатків та дозволяють:

1. Розробку, розгортання та управління моделями в масштабі.
2. Розробку за допомогою інструментів і фреймворків, популярних в співтоваристві з відкритим кодом.

3.2.1 Інтегроване середовище розробки

За допомогою Microsoft Azure для створення додатків штучного інтелекту можна використовувати будь-яке інтегроване середовище розробки (IDE) або редактор. У деяких популярних IDE є модулі або розширення, які роблять розробку ще простіше, наприклад публікація безпосередньо в Azure [26].

Ці IDE мають хороші можливості для прискорення розробки, але, звичайно, інші популярні IDE, такі як PyCharm і RStudio, можуть бути використані для розробки коду, який буде працювати на платформі Microsoft AI, і з часом стане доступно більше розширень.

3.2.2 Сервіс: Bot Framework

Bot Framework включає в себе інструменти і сервіси, що дозволяють розробникам створювати ботів, які спілкуються з користувачами. Наприклад, розробник може легко розробити бота, який взаємодіє з користувачами на веб-сайті, щоб направляти їх через покупку продукту або послуги, а не переміщатися по веб-сторінці. За допомогою цієї структури можна один раз розробити, а потім виставити бот через безліч каналів, які включені в черевик Фреймворки, такі як Skype, Facebook та Web. Боти можуть бути побудовані за допомогою комплекту розробки програмного забезпечення Boat Builder Software Development Kit (SDK) з використанням C# або Node.js або за допомогою Azure Bot Service [27].

Боти можуть бути створені для природного спілкування, особливо з використанням розширених можливостей з інтеграцією когнітивних сервісів, таких як мова Розуміння служби розвідки (LUIS) та інтеграції з іншими когнітивними службами (рис 3.2).

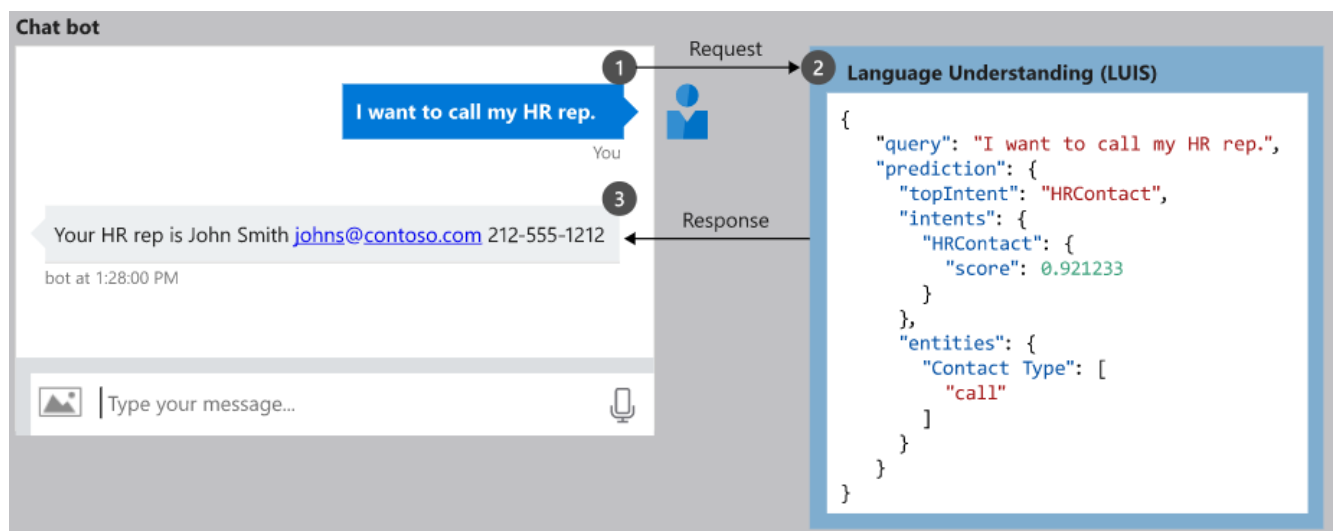


Рисунок 3.2 – Приклад бота на основі Microsoft LUIS

Як керована служба в Azure, вона масштабована, і витрати виникають тільки для використовуваних ресурсів. Після публікації програми Azure LUIS клієнтська програма надсилає висловлювання (текст) до API кінцевої точки обробки природного мови LUIS і отримує результати як відповіді JSON. Поширеним клієнтським додатком для LUIS є чат-бот.

3.2.3 Віртуальна машина Data Science або DSVM

DSVM-це попередньо налаштоване середовище в хмарі для моделювання, розробки та розгортання даних і штучного інтелекту. В якості операційних систем використовуються Windows Server та Linux спеціалізована версія для глибокого навчання, відома як “*DLVM*” [28], яка працює на графічному процесорі (рис 3.3).

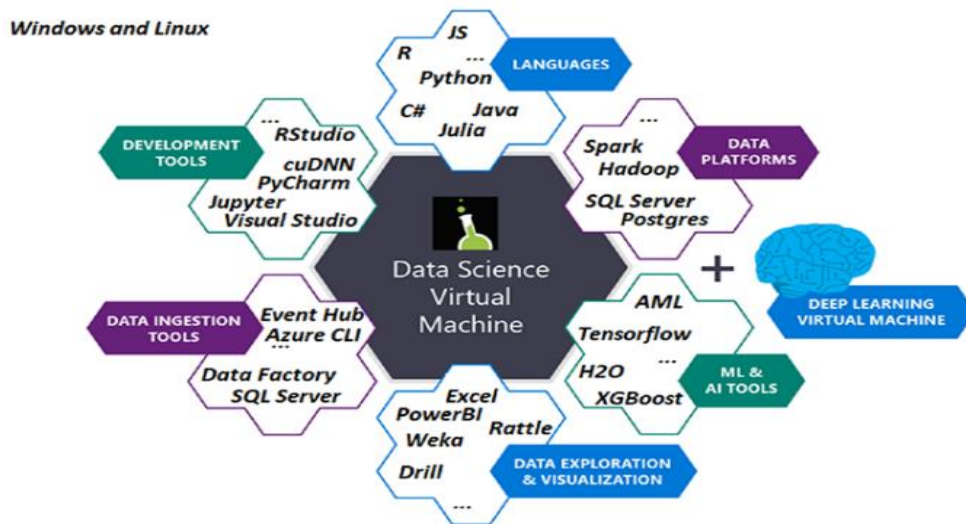


Рисунок 3.3 – Інфраструктура DSVM

Популярні мови для data science розробки, такі як Python, R і Julia, C#, готові до негайного використання, а дані підключаються з багатьох сховищ даних, таких як SQL Data Warehouse, Azure Data Lake, Azure Storage та інші.

Dsvm надзвичайно популярні серед фахівців з обробки даних з наступних причин:

1. Вони надають аналітичний робочий стіл в хмарі з простим налаштуванням і можливістю більш легкої передачі проектів між колегами.
2. Вони мають еластичну ємність на вимогу, здатність повертатися вимикається і вмикатися.
3. Є приклади і шаблони, вбудовані для початку роботи з Data Science і Deep Learning.
4. Існує можливість підключення до інших служб, таким як використання DSVM як обчислювальної мети в рамках проекту, керованого через служби машинного навчання Azure, або в якості служби обчислень для Batch AI.

3.2.4 Deep Learning Frameworks

Платформа Microsoft AI-це відкрита платформа, заснована на кращих технологіях з відкритим вихідним кодом. Фреймворки глибокого навчання, такі як Microsoft Cognitive Toolkit(CNTK), Tensorflow, Caffe і PyTorch, які є проектами з відкритим вихідним кодом, підтримуються в багатьох вже згаданих інструментах, службах та інфраструктурі [29].

LLVM поставляється попередньо налаштованим з багатьма популярними фреймворками, і ці фреймворки можуть бути використані для розробки рішень штучного інтелекту і розгортання на них. Наприклад, Azure, on Azure IoT Edge або Windows Machine Learning.

3.2.5 Служба машинного навчання Azure

Служби машинного навчання Azure надають платформу для управління проектом Data science. За допомогою цих сервісів можна створити обчислювальну середовище, найбільш підходящу для навчання їх моделей штучного інтелекту, наприклад:

1. Наука Про Дані Віртуальної Мащини.
2. Spark на Databricks або HDInsight.
3. Azure Batch AI.

Служба експериментів “*Experimentation service*” допомагає керувати залежностями проектів, масштабувати навчальні завдання та забезпечувати спільне використання проектів у галузі науки про дані [26].

Доступні бази даних Azure Cosmos DB. Багато інструментів ml та AI попередньо встановлені, наприклад, багато популярних фреймворки глибокого навчання. Потім фахівці з обробки даних і розробники можуть налаштувати віртуальну машину так, як це необхідно для їх використання.

Служба управління моделями використовує розгортання на основі контейнерів *docker*, щоб допомогти розробникам розгортати рішення на одному вузлі (в хмарі або локально), а також масштабувати кластерні розгортання, такі як

Контейнерні служби Azure, а також прикордонне розгортання через Azure IoT Edge. На даний момент служби машинного навчання Azure працюють також з Python.

Docker - програмне забезпечення для автоматизації розгортання і управління додатками в середовищах з підтримкою контейнеризації. Дозволяє «упакувати» додаток з усім його оточенням і залежностями в контейнер, який може бути перенесений на будь-яку Linux-систему з підтримкою cgroups в ядрі, а також надає середовище з управління контейнерами.

2.2.6 Контейнерний хостинг (Azure Container Services)

Служба Azure Kubernetes Service (AKS) - це повністю керована служба оркестровки контейнерів Kubernetes [30]. AKS являє собою надзвичайно гнучку універсально обчислювальну платформу. Це сервіс часто використовується для розміщення масштабованих моделей AI для оцінки в реальному часі, хоча AKS також може використовуватися для масштабованого навчання AI. Служби машинного навчання включають в себе службу управління моделями, яка полегшує розгортання моделей штучного інтелекту в якості REST API в контейнері (рис 3.4).

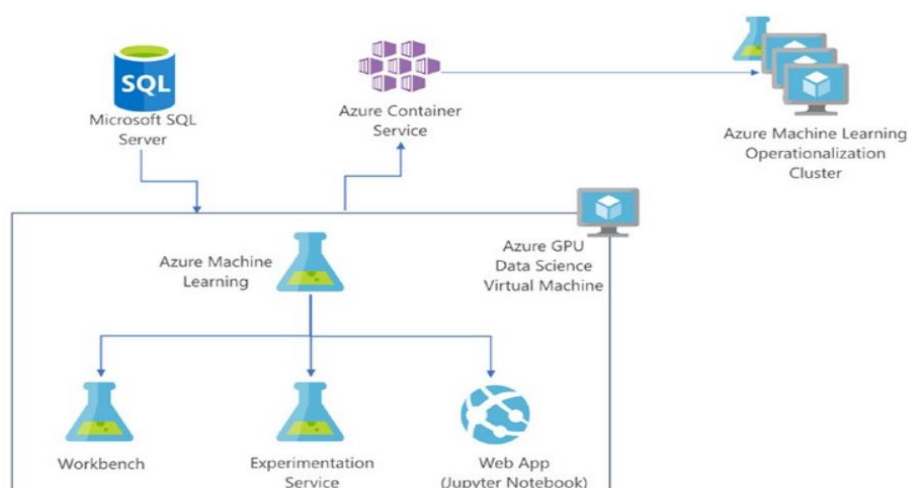


Рисунок 3.4 - Архітектура рішення глибокого навчання Azure

Розробники також можуть розміщувати контейнери за допомогою екземплярів контейнерів Azure, де контейнер може розміщуватися без оркестратора контейнерів, це особливо корисно для тестування або розміщення простого додатка та не вимагає масштабування.

РОЗДІЛ 4. РОЗРОБКА ТА НАВЧАННЯ ПРОГРАМНОГО МОДУЛЯ АІ ДЛЯ РОЗПІЗНАВАННЯ КОМАНД КОРИСТУВАЧА

4.1 Постановка задачі

Технічним завданням даної роботи є розробка програмного модуля АІ для розпізнавання команд користувача. На основі аналізу існуючих рішень та проведення дослідження було виявлено, що програмний модуль АІ повинен містити наступні функціональні можливості:

- Розпізнавання мови (Automatic Speech Recognition, ASR);
- Виділення смислових об'єктів (Natural Language Understanding, NLU);
- Виконання команд;
- Синтез мови (Text-to-Speech, TTS);

4.2 Архітектура

Кожна функціональна можливість являє собою мікросервіс. З цього випливає, що при необхідності функції можуть використовуватися окремо одна від одної. Вихід з ладу однієї з функцій не вплине на роботу інших.



Рисунок 4.1 – Функціональні програмного модуля АІ

Кожна функція є окремим блоком, які складаються воедино, що вже являє собою кінцевий результат. Блоки комунікують між собою по схемі (Chain of Responsibility) ланцюга обов'язків, тобто кожен модуль відповідає за свою роботу (переклад мовного сигналу в текст, виділення смислових образів, виконання

команд і т.д.). Кінцева точка роботи модуля (результат його роботи) є вихідною точкою для наступного. Так мова користувача перетворюється в текст і відправляється на обробку алгоритмам машинного навчання для визначення намірів користувача. Залежно від цього наміру активується потрібний клас в модулі виконання команд, який виконує вимогу користувача. По завершенні операції модуль виконання команд передає інформацію про статус виконання команди модулю синтезу мови, який, в свою чергу, оповіщає користувача.

4.3 Розробка блоків програмного модулю AI

4.3.1 Розпізнавання мови

Першим етапом розпізнавання мови є обробка голосового сигналу і витяг ознак. Найпростішим поданням звукового сигналу може служити осцилограма. Вона відображає кількість енергії в кожен момент часу. Однак для визначення сказаного звуку цієї інформації недостатньо. Важливо знати, яка кількість енергії міститься в різних частотних діапазонах (див. підрозділ 2.1.5). Для цього проводиться перехід від осцилограми до спектру (рис. 4.2).

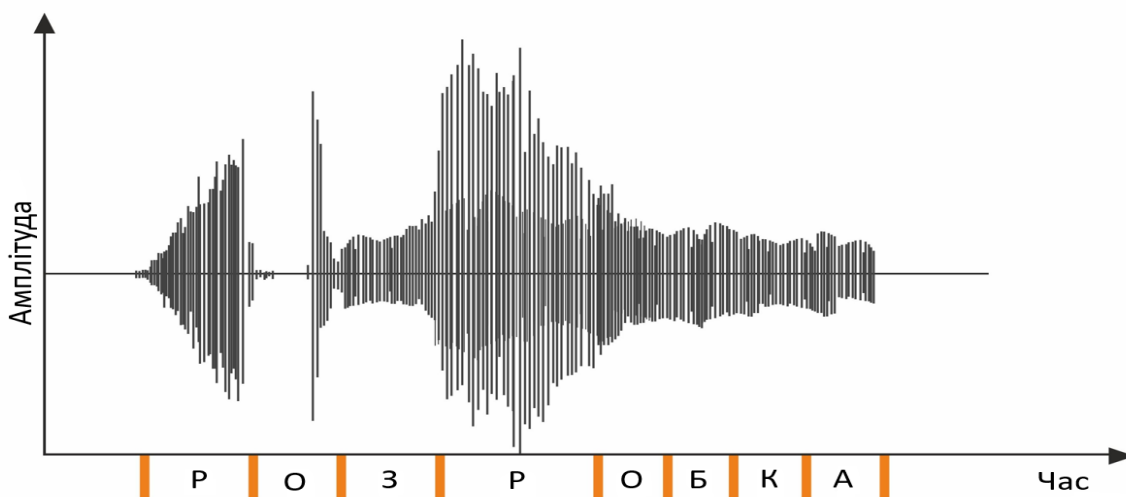


Рисунок 4.2 – Осцилограма слова

Для класифікації слова важлива лише інформація про конфігурацію фільтра. Виділити таку інформацію дозволяє перехід від спектра до кепстра (cepstrum), що виконується за допомогою зворотного перетворення Фур'є від логарифма спектра.

Для проведення відмінності між тимчасовими областями кепстра і вихідного звукового сигналу по осі x відкладається час.

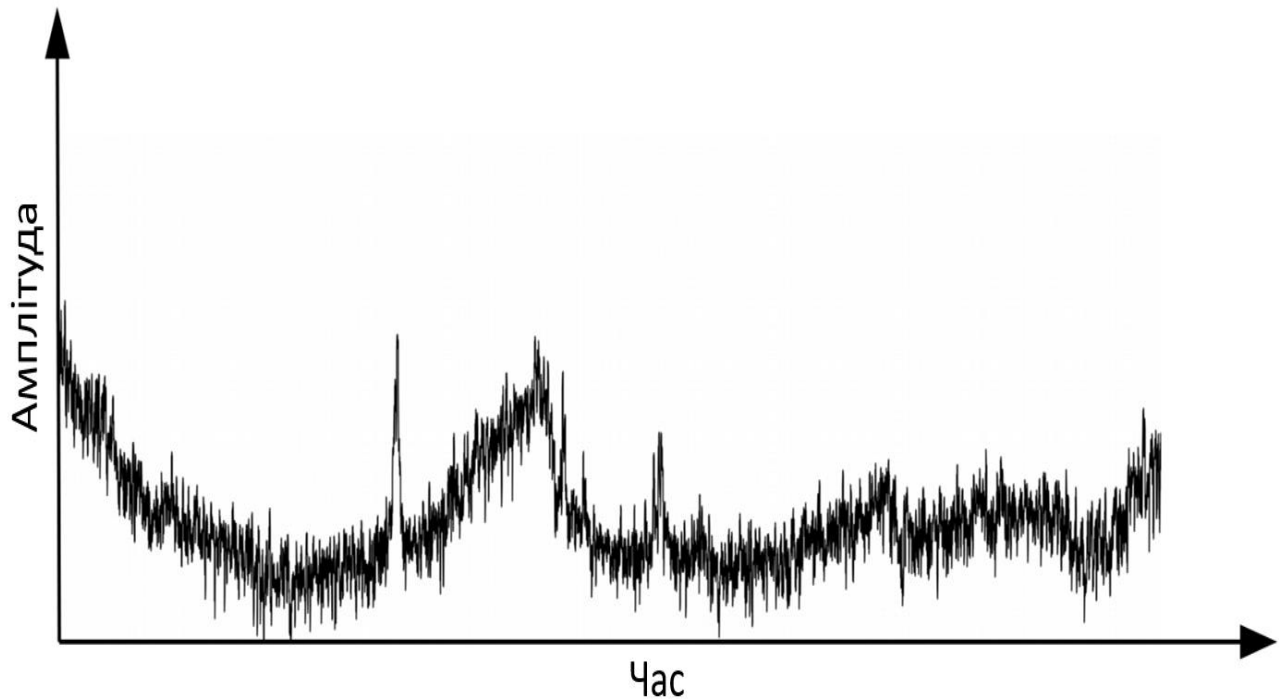


Рисунок 4.3 – Кепстра або “спектр логарифма спектра”

Інформація про стан голосового тракту знаходиться в 12 перших коефіцієнтах кепстра. Ці 12 кепстральних коефіцієнтів доповнюються динамічними ознаками, що описують зміни звукового сигналу[22]. Отриманий вектор значень зветься MFCC вектор (Mel-frequency cepstral coefficients) і є найбільш поширеною акустичною ознакою, використаною в розпізнаванні мови.

Ознаки використовуються в якості вхідних даних для акустичної моделі. Вона показує, яка лінгвістична одиниця створила подібний MFCC вектор. У різних системах такими лінгвістичними одиницями можуть слугувати частини фонем або навіть слова. Таким чином, акустична модель дозволяє перетворити послідовність MFCC векторів в послідовність найбільш ймовірних фонем. Для послідовності фонем необхідно підібрати відповідну послідовність слів. Тут в справу вступає словник мови, що містить транскрипцію всіх розпізнавальних системою слів.

Для виконання цього завдання використовується Azure Cognitive Speech Service. Цей сервіс який формує мовну модель, з його допомогою можна голосовий запис перетворити в текст для подальшої обробки (рис. 4.4).

```
public async Task<string> GetTextFromAudioAsync(Stream audiostream)
{
    var requestUri = @"https://speech.platform.bing.com/recognize?scenarios=smd&appid=D4052672-91D7-4C74-8AD8-4281D98141A5&locale=en";
    using (var client = new HttpClient())
    {
        var token = Authentication.Instance.GetAccessToken();
        client.DefaultRequestHeaders.Add("Authorization", "Bearer " + token);

        using (var binaryContent = new ByteArrayContent(StreamToBytes(audiostream)))
        {
            binaryContent.Headers.TryAddWithoutValidation("content-type", "audio/wav; codec=audio/pcm; samplerate=16000");

            var response = await client.PostAsync(requestUri, binaryContent);
            if (response.StatusCode != System.Net.HttpStatusCode.OK)
                throw new HttpException((int)response.StatusCode, $"{response.StatusCode} {response.ReasonPhrase}");

            var responseString = await response.Content.ReadAsStringAsync();
            try
            {
                dynamic data = JsonConvert.DeserializeObject(responseString);
                return data.header.name;
            }
            catch (JsonReaderException ex)
            {
                throw new Exception(responseString, ex);
            }
        }
    }
}
```

Рисунок 4.4 – Ділянка коду яка відповідає за виділення слів із звукового сигналу

На наступному етапі мовна модель визначає апріорну ймовірність пропозиції в мові. Іншими словами, модель дає оцінку наскільки поява такої пропозиції в мові правдоподібна (рис 4.5).

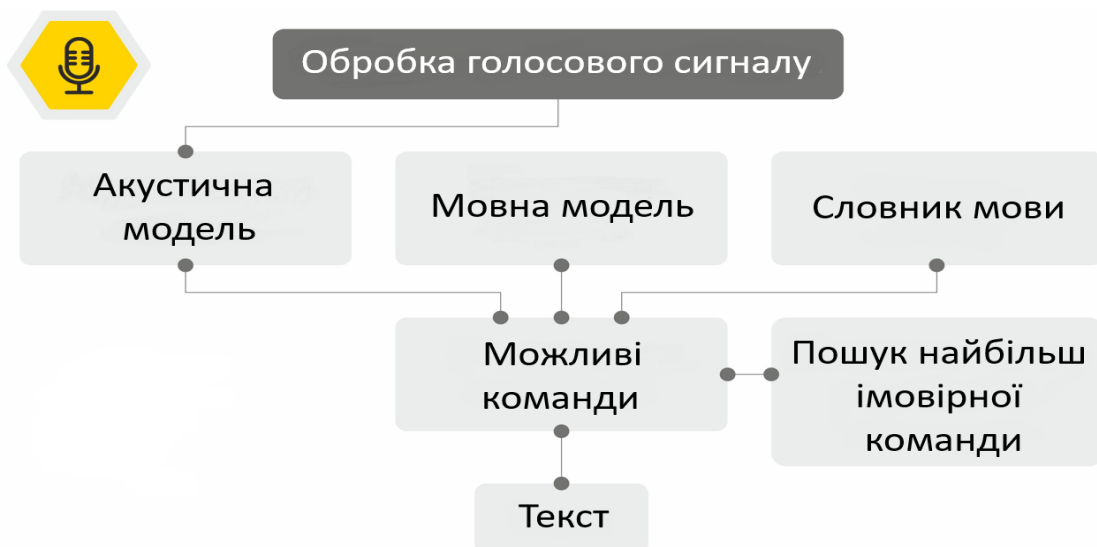


Рисунок 4.5 – Схема роботи системи розпізнавання мови

Комбінація акустичної моделі, мовної моделі і словника вимови створює всі можливі комбінації гіпотез послідовності слів, з яких за допомогою алгоритму

динамічного програмування можна знайти найбільш ймовірну, її система запропонує в якості розпізнаного тексту. Для перетворення послідовності фонем в словах використовується послуга Azure Web Language Model.

Azure Web Language Model - це хмарний сервіс, заснований на REST, що надає найсучасніші інструменти для обробки природної мови. Використовуючи цей сервіс, додаток може використовувати потужність *big data* за допомогою мовних моделей зібраних компанією Bing.

4.3.3 Виділення смислових об'єктів

При розробці цього блоку було використано Azure Luis, приклад використання в кінцевому коді продемонстровано на (рис. 4.6). Щоб виділити смисл з тексту розпізнаної фрази, необхідно вирішити два завдання машинного навчання:

1. Класифікація команди користувача (Intent Classification).
2. Виділення іменованих сутностей (Named Entity Recognition).

```
private static async Task<StockLUIS> GetEntityFromLUIS(string Query)
{
    Query = Uri.EscapeDataString(Query);
    StockLUIS Data = new StockLUIS();
    using (HttpClient client = new HttpClient())
    {
        string RequestURI = "https://api.projectoxford.ai/luis/v1/application?id:
        HttpResponseMessage msg = await client.GetAsync(RequestURI);

        if (msg.IsSuccessStatusCode)
        {
            var JsonDataResponse = await msg.Content.ReadAsStringAsync();
            Data = JsonConvert.DeserializeObject<StockLUIS>(JsonDataResponse);
        }
    }
    return Data;
}
```

Рисунок 4.6 - Ділянка коду яка відповідає за виділення смислових об'єктів

Для класифікації команд користувача Azure Luis представляє текст у вигляді числового вектора, який може бути оброблений машиною. Для подібного

перетворення використовується нейронна модель StarSpace, що дозволяє «вкласти» текст запиту і клас запиту в загальний простір.

Під час навчання така нейронна мережа вчиться порівнювати сутності, так щоб мінімізувати відстань між вектором запиту і вектором вірного класу і максимізувати відстань до векторів відмінних класів.

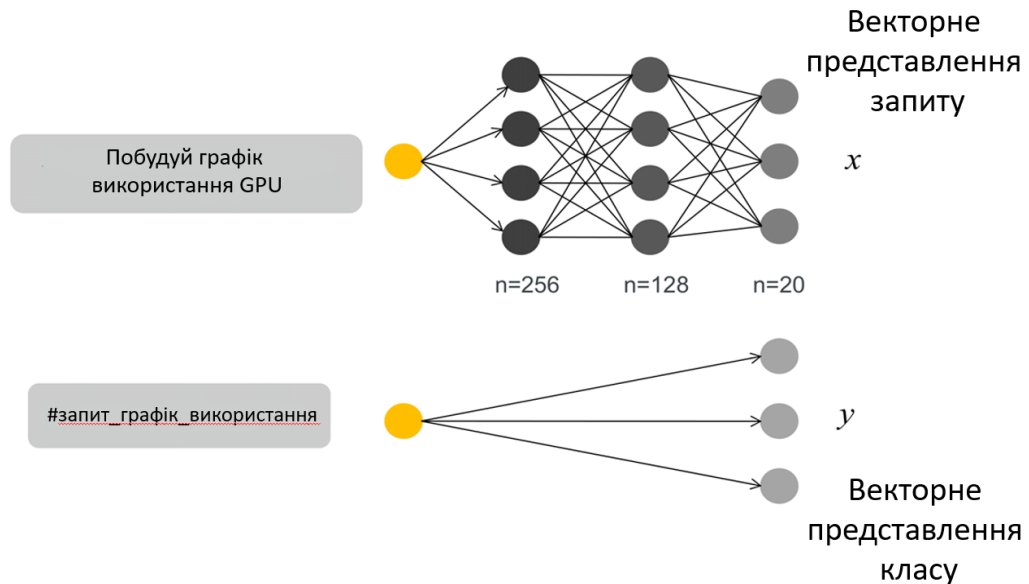


Рисунок 4.7 – Нейронна модель StarSpace

Під час тестування для запиту x вибирається клас y , так щоб (4.1):

$$\max_y \text{similarity}(x, y) \quad (4.1)$$

В якості запобіжної схожості векторів використовується косинусна відстань:

$$\text{similarity} = \cos(\theta) = \frac{x \cdot y}{\|x\| \cdot \|y\|} \quad (4.2)$$

де (4.1 – 4.2) x - запит користувача,

y - категорія запиту.

4.3.4 Навчання класифікатора намірів

Для навчання класифікатора намірів користувача було розміщено 100 запитів, всього вийшло 6 класів. Вибірка була розділена на навчальну і тестову у співвідношенні 60/40 за допомогою методу стратифікації за цільовою змінною. Стратифікація дозволила зберегти вихідний розподіл класів в трейн і тесті.

Якість навченої моделі оцінюється відразу за кількома критеріями:

1. Повнота (Recall) - частка вірно класифікованих запитів щодо всіх запитів даного класу.
2. Частка вірно класифікованих запитів (Accuracy).
3. Точність (Precision) - частка вірно класифікованих запитів щодо всіх запитів, які система віднесла до даного класу.
4. Міра F1 - гармонійне середнє між точністю і повнотою.

Також для оцінки якості моделі класифікації використовується матриця помилок системи. По осі Y проставлений істинний клас висловлювання, по осі X - клас, передбачений алгоритмом. Метрики моделі на тестовому датасеті: Accuracy - 92%, F1 - 90%.

Друге завдання - виділення іменованих сутностей, полягає в ідентифікації слів і словосполучень, що позначають конкретний предмет. Такими сутностями можуть бути, наприклад, назва девайсу або операції яку необхідно провести на девайсі.

Цільова змінна

NA

NA

Мапа

Токен

Відкрити мапу девайсів

Рисунок 4.8 – Приклад з навчальної вибірки для задачі Named Entity Recognition.

Для вирішення завдання використовується алгоритм умовних ймовірнісних полів (Conditional Random Fields), що представляють собою різновид Марковських полів. Щоб виконувати прохання користувачів, необхідно виділяти типи іменованих сутностей. Для навчання моделі був підготовлений датасет і приведена анотація, до кожного слова в вибірці присвоєно відповідний клас.

Обсяг всього датасета склав 200 запитів, які розділені на трейн і тест в співвідношенні 60/40. Якість моделі оцінюється за допомогою метрики F1 і складає 98% при тестуванні на контрольній вибірці. На контрольній вибірці модель показала результати продемонстровані в (додатку А).

4.3.5 Виконання команд

Залежно від класу запиту користувача, визначеного на етапі класифікації, система активує відповідний клас. Кожен клас складається як мінімум двома методами: методом, який безпосередньо виконує запит, і методом генерації відповіді для користувача.

Наприклад, при віднесенні команди до класу "запит_графік_використання" створюється об'єкт класу RequestDeviceKillProcess, який виносить інформацію з бази даних.

```
[DisplayName("Kill Process")]
[Category("Action")]
[JobStep(ImageKey = "excluded")]
1 reference | - changes | -authors, -changes
public class KillProcessStep : JobStepBase<KillProcessStepSettings>
{
    0 references | - changes | -authors, -changes
    public KillProcessStep()
    {
        this.PossibleResults = new string[] { JobResult.ProcessKilled, JobResult.NoneResult };
    }
    - references | - changes | -authors, -changes
    public override Task<IJobStepResult> StartOrResume()
    {
        var processKilled = false;

        var processName = Path.GetFileNameWithoutExtension(ReplaceVariables(SpecificSettings.ProcessName));

        log.Info($"Attempting to find and kill process:{processName}");

        var processes = Process.GetProcessesByName(processName);
        foreach(var process in processes)
        {
            try
            {
                log.Info($"Attempting to kill process:{processName} Id:{process.Id}");
                process.Kill();
                processKilled = true;
            }
            catch { }
        }

        log.Info($"Process was killed:{processName} {processKilled}");

        this.OutputResult = processKilled ? JobResult.ProcessKilled : JobResult.NoneResult;

        return JobStepResult.OkTask();
    }
}
```

Рисунок 4.9 – Ділянка коду яка відповідає за виконання команд

4.3.6 Синтез мови

Згенерований на етапі виконання команди одним із методів генерації відповіді текст оповіщення користувача виводиться на екран, а також використовується в якості входу для модуля синтезу усного мовлення. Генерація мови здійснюється з використанням бібліотеки Microsoft Cognitive Services Speech.

Основними компонентами системи синтезу мови є лінгвістичний процесор, який обробляє подачу на вхід текст. Проводиться нормалізація тексту: цифри приводяться до письмового подання, абревіатури розшифровуються (рис. 4.10)

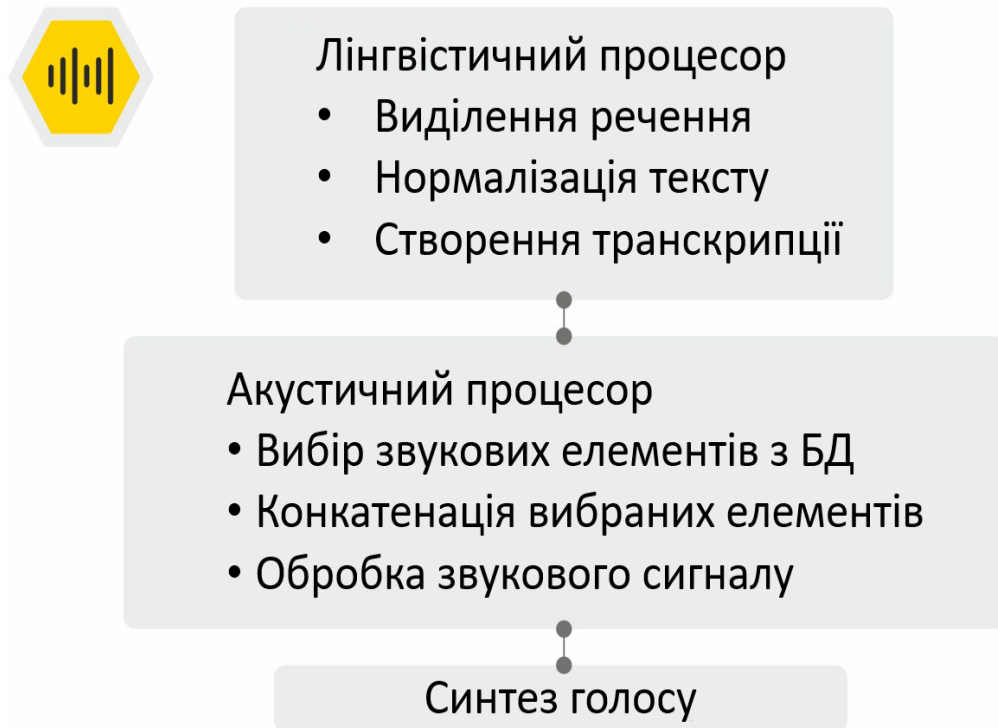


Рисунок 4.10 – Схема роботи синтезу мови

Далі за допомогою словника вимови відбувається створення транскрипції для тексту, що далі передається на вхід акустичного процесора (рис. 4.11).

```

private static PlayPrompt GetPromptForText(string text)
{
    var prompt = new Prompt { Value = text, Voice = VoiceGender.Male };
    return new PlayPrompt { OperationId = Guid.NewGuid().ToString(), Prompts = new List<Prompt> { prompt } };
}

public async Task SynthesizeAudioAsync(string text)
{
    var config = SpeechConfig.FromSubscription("788bec0c-3e46-11eb-b378-0242ac130002", "e37b21cc-3e46-11eb-b378-0242ac130002");
    using var audioConfig = AudioConfig(GetPromptForText(text))
    using var synthesizer = new SpeechSynthesizer(config, audioConfig);
    await synthesizer.SpeakTextAsync();
}
  
```

Рисунок 4.11 – Ділянка коду яка відповідає за синтез мови

Даний компонент відповідає за генерацію звукових елементів (синтетичної мови) та конкатенацію обраних елементів і обробку звукового сигналу.

4.3.7 Кінцевий результат

Отже, всі компоненти голосового помічника готові. Залишилося лише «зібрати» їх у правильній послідовності. Як вже згадували раніше, кожен модуль являє собою мікросервіс.

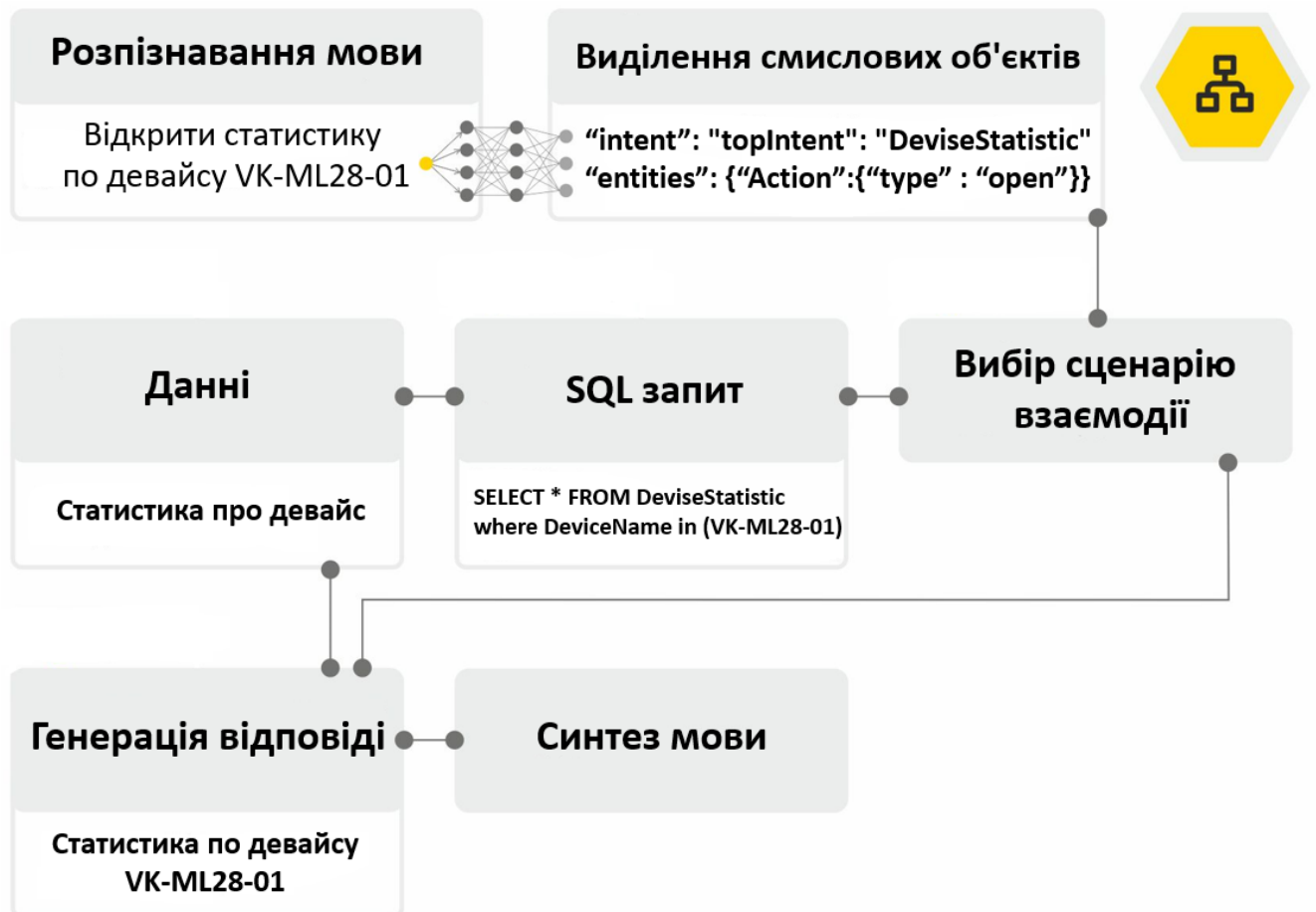


Рисунок 4.12 – Схема роботи програмного модулю AI

Для зв'язки всіх модулів використовується фреймворк RabbitMQ. Ілюстрація демонструє внутрішню роботу асистента на прикладі типового запиту користувача (рис. 4.12).

ВИСНОВОК

Нейронні мережі можуть знаходити саме різне застосування, причому не тільки для розпізнавання зображень і текстів, а й у багатьох інших сферах. НС здатні до навчання, завдяки чому їх можна оптимізувати і максимально збільшувати функціональність.

В дипломній роботі були розглянуті різні структури нейронних мереж, досліджено структуру штучного нейрону та проаналізовано різні стратегії навчання нейронних мереж. Розглянуто задачу розпізнавання образів як наукову задачу, проведено характеристику розпізнавання образів та виділені основні недоліки існуючих методів розпізнавання.

Після чого було розглянуто мовний сигнал та загальну структуру системи автоматичного розпізнавання мови та методів порівняння, досліджено теорію розпізнавання природньої мови за допомогою AI.

Архітектура програмного модуля повинна базуватися на сервісах Microsoft Azure AI, тож був проведений аналіз можливостей даної платформи, розібрані більш детально деякі із сервісів.

Після ознайомлення з Microsoft Azure AI відбулася постановка задачі, визначення необхідного функціоналу та формування вимог до нього. В результаті на основі аналізу необхідного функціоналу, була створена архітектура програмного модуля AI, яка був розподілений на блоки з унікальним функціоналом.

По результатам проведеної роботи, був реалізований програмний модуль AI для розпізнавання команд користувача на базі сервісів Microsoft Azure AI. Реалізовані блоки: Розпізнавання мови (Automatic Speech Recognition, ASR), Виділення смислових об'єктів (Natural Language Understanding, NLU), Виконання команд, Синтез мови (Text-to-Speech, TTS);

ПЕРЕЛІК ПОСИЛАНЬ

1. Helpiks [Електронний ресурс] : [Веб-сайт] – Режим доступу: <https://helpiks.org/5-72195.html>.
2. EverGreens [Електронний ресурс] : [Веб-сайт] – Режим доступу: <https://evergreens.com.ua/ua/articles/cnn.html>.
3. Бібліотека Університет ІТМО - [Електронний ресурс] : [Веб-сайт] – Режим доступу: <http://neerc.ifmo.ru/wiki/>.
4. EverGreens [Електронний ресурс] : [Веб-сайт] – <https://evergreens.com.ua/ru/articles/gan.html>.
5. Бібліотека [Електронний ресурс] : [Веб-сайт] – <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/16450/1/Neural%20systems1.pdf>.
6. Бібліотека [Електронний ресурс] : [Веб-сайт] – Режим доступу: <https://www.victoria.lviv.ua/library/nn/help/neuron.pdf>.
7. Neuronus [Електронний ресурс] : [Веб-сайт] – Режим доступу: <https://neuronus.com/theory/nn/238-obucheniya-nejronnoi-seti.html>.
8. Zacks Shon - The theory of statistical views. – 1975р. – 776 с.
9. David Forsyth - Computer Vision: A Modern Approach. – 2004. – 928 с.
10. Вапник В.Н. Теорія розпізнавання образів – 1979р. – 416 с.
11. Фомин Я.А. Розпізнавання образів: теорія і застосування 2-ге вид – 2012. – 429 с.
12. Утробин В. А. Елементи теорії активного сприйняття зображень – 2010р. – 61-69 с.
13. Чэн Ш.-К. Принципи проектування систем візуальної інформації –1994. – 408 с.
14. Hastie T. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. – 2nd ed, 2009р. – 746 с.
15. Краснополюсовський А. С. Інформаційний синтез інтелектуальних систем керування: підхід, що ґрунтується на методі функціонально-статистичних випробувань – 2004р. – 261 с.

16. O'Shaughnessy D. Acoustic Analysis for Automatic Speech Recognition. – 2013p – 1038-1053 с.
17. I.V. Ognev, A.I. Ognev, P.A. Paramonov, N.A. Sutula, The use of extrema distribution as a feature vector for speech patterns recognition – 2013p. – 114-117 с.
18. X. Huang, A. Acero. Spoken language processing: a guide to theory, algorithm, and system development, Prentice Hall. – 2001 p. – 1008 с.
19. Anusuya M.A., Katti S.K., Front end analysis of speech recognition: a review // International Journal of Speech Technology, – 2011p. – 99 с.
20. Л. Рабинер, Р. Шафер, Цифрова обробка мовних сигналів – 1981р. – 496 с.
21. Загоруйко Н. Г., Прикладні методи аналізу даних і знань – 1999р – 270 с.
22. О.С. Агашин, О.Н. Корелин, Методи цифрової обробки мовного сигналу в задачі розпізнавання ізольованих слів із застосуванням сигнальних процесорів – 2012р. – 32-44 с.
23. Setlab [Електронний ресурс] : [Веб-сайт] – Режим доступу: http://www.setlab.net/?view=Philosophy_Knowledge
24. Romanyshyn Y., Tkachenko V. Comparison of similarity measures of speech signals fragments – 2010p. – P.184..
25. Хайкин С. Нейронні мережі: повний курс; 2-вид. – 2006р. – 1104 с.
26. Mathew Salvaris. Deep Learning with Azure – 2018p. – 82-90с.
27. Microsoft Docs [Електронний ресурс] : [Веб-сайт] – Режим доступу: <https://docs.microsoft.com/ru-ru/composer/introduction>
28. Microsoft Docs [Електронний ресурс] : [Веб-сайт] – Режим доступу: <https://docs.microsoft.com/ru-ru/azure/machine-learning/data-science-virtual-machine/overview>
29. Microsoft Docs [Електронний ресурс] : [Веб-сайт] – Режим доступу: <https://docs.microsoft.com/en-us/azure/machine-learning/data-science-virtual-machine/dsvm-tools-deep-learning-frameworks>
30. Microsoft Docs [Електронний ресурс] : [Веб-сайт] – Режим доступу: <https://docs.microsoft.com/en-us/azure/aks/intro-kubernetes>

ДОДАТОК А

Запит на карантин девайсу	0.83	0.17	0.00	0.00	0.00	0.00	0.00	0.00
Запит оновлення девайсу	0.10	0.90	0.00	0.00	0.00	0.00	0.00	0.00
Запит сканування девайсу	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Запит показу карти підключень	0.00	0.29	0.00	0.71	0.00	0.00	0.00	0.00
Запит закриття процесу	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Запит статистики використання	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Запит запуску додатку	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
Запит використання	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
	Запит на карантин девайсу	Запит оновлення девайсу	Запит на сканування	Запит показу карти	Запит закриття процесу	Запит статистики	Запит запуску додатку	Запит використання

Таблиця – Модель результатів контрольної вибірки метрики моделі на тестовому датасеті: Ассурасу - 92%, F1 - 90%.

ДОДАТОК Б


Матеріали приклади коду, будуть додані перед друком пізніше

ДОДАТОК В


Презентація ще не дороблена, завтра буду консулюватися на цю тему с дипломник керівником. Матеріли будуть відображені як на слайді нижче

ДОДАТОК В

Слайд 1



**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**



Розробка інтелектуального модуля CRM на базі Microsoft Dynamics CRM для центру Дуальної Освіти

Виконав студент групи: ПД-41
Шевченко Дмитро Олександрович
Науковий керівник: доцент кафедри к.т.н.
Щербина Ірина Сергіївна

Київ 2019

Слайд 2

2

Об'єкт дослідження: Автоматизація робочих процесів центру Дуальної Освіти за допомогою CRM.

Предмет дослідження: Microsoft Dynamics CRM.

Мета дослідження: Автоматизація роботи з клієнтами та внутрішніх робочих процесів в центрі дуальної освіти.