ЗМІСТ

ДОДАТКИ

Д.1 Копії графічних матеріалів

ІАЛЦ.045490.005 Д1 Система розпізнавання ключових слів. Підготовка вхідних даних. Схема алгоритму.

ІАЛЦ.045490.006 Д2 Система розпізнавання ключових слів. Створена комп’ютерна система. Схема структурна.

ІАЛЦ.045490.007 Д3 Система розпізнавання ключових слів. Схема алгоритму.

ІАЛЦ.045490.008 Д4 Система розпізнавання ключових слів. Рекурентна нейронна мережа. Схема структурна

Д.2 Лістинг програми

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

КМ – комп’ютерна мережа

КС – комп'ютерна система

НМ – нейрона мережа

ОС – операційна система

ПЗ – програмне забезпечення

ПСІ – прикладний програмний інтерфейс

СШН – схований шар нейронів

CLI – інтерфейс командної строки (Command Line Interface)

DCT – Дискретне косінусне перетворення

FFT – Швидке перетворення Фур’є (Fast Fourier Transform)

MFCC – Мел-частотні кепстральні коефіцієнти (Mel-frequency cepstral coefficients).

WAV – формат аудіофайлу (waveform audio format)

ВСТУП

В наш час обчислювальна техніка використовується в багатьох областях людської діяльності, будучи зручним і багатофункціональним інструментом для вирішення широкого кола завдань. Однак, користувачі ЕОМ змушені використовувати способи взаємодії, що слабо адаптовані до можливостей людського спілкування і обмежують можливості людини до обміну інформацією. Основна мета удосконалення та розвитку інтерфейсу людина-комп'ютер полягає в організації обміну інформацією з ЕОМ таким чином, щоб:

- знизити час освоєння програмних і апаратних засобів;
- знизити рівень помилок при передачі інформації;
- робити роботу з ЕОМ можливої для людей, які не мають можливості користуватися традиційними засобами інтерфейсу;
- Знизити стомлюваність, збільшити суб'єктивне задоволення користувача від роботи.

Для досягнення поставлених цілей необхідно застосування засобів взаємодії, що більш повно використовують комунікативні здібності людини. Людина наділена великою кількістю можливостей сприймати і передавати інформацію: зір, слух (в тому числі усне мовлення), жести і рухи, міміка, дотик і іншими. У взаємодії людини і комп'ютера існують два інформаційних потоки:

- керуючі команди і дані, що передаються комп'ютеру для обробки;
- результати обчислень і інша інформація, яка надається комп'ютером користувачеві.

Поширений в даний час людино-машинний інтерфейс використовує зір, як основний канал подання інформації користувачу, відображаючи дані у вигляді умовних знаків на екрані комп'ютера. Сприймати інформацію природними для людини способами (розпізнавати мову, жести, міміку і так далі) сучасні засоби інтерфейсу практично не в змозі.

Особа людини є важливим джерелом інформації при спілкуванні між людьми. Вираз обличчя, міміка, артикуляція при розмові, рухи головою є зручним, природним і, що важливо, необтяжливим способом передачі інформації. Нездатність комп'ютера з одного боку сприйняти, а з іншого боку відтворити настільки природні для людини способи спілкування ускладнює передачу і сприйняття інформації при роботі з ЕОМ.

Для забезпечення ефективного усного діалогу між користувачем і ЕОМ необхідні стійкі системи розпізнавання мови.

Основною метою дипломної роботи є розробка системи розпізнавання голосових сигналів для подальшого голосового управління електромеханічними пристроями.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1 Загальний опис проблеми розпізнавання голосових сигналів

Проблема створення усного діалогу людини з машинами є однією з найбільш актуальних проблем кібернетики, інформатики і обчислювальної техніки. Оснащення ЕОМ засобами розпізнавання та синтезу мови має та ще в більшій степені буде мати велике економічне та соціальне значення. Це забезпечить доступність ЕОМ всьому населенню, можливість програмування і рішення задач на природній мові, безпаперову технологію управління, скорочення термінів навчання користувачів ЕОМ, підвищення продуктивності праці в сферах виробництва, розподілення і в побуті, підвищення ефективності використання техніки, створення сприятливих умов праці.

Таким чином, мова йде про створення й використання інтелектуальних ЕОМ і людино-машинного інтерфейсу на природній мові в комп'ютерних системах.

Історія науки і техники налічує чимало спроб створення "слухаючих" та "говорящих" машин починаючи ще з XVIII століття. Цьому в значній степені сприяли становлення та розвиток електроніки та електрозв'язку. Але найбільший інтерес до проблеми та її розвиток починаються одночасно з появою ЕОМ і їх широким розповсюдженням, з автоматизацією різних областей діяльності людини.

Усний діалог людини з ЕОМ в найбільш зручній та звичній для людини формі – голосом – став техніко-економічною і соціальною необхідністю. Чисто в науковому плані кінцевою метою досліджень є створення засобів усного діалогу людини і ЕОМ на природніх мовах,

наприклад автоматичною машинки, шо друкує та редагує тексти під диктовку, або машин-перекладачів з голосу.

Процес розпізнавання мови являє собою перетворення акустичного сигналу, отриманого від мікрофона, в послідовність слів. Отриманий набір гіпотез ланцюжків слів далі використовується для розуміння мови. При цьому виникає ряд проблем. По-перше, людина зазвичай не робить паузи між словами, а при злитому проголошенні до задачі розпізнавання додається ще й завдання виділення слів з потоку мови, що свідомо більш складно. Виникає необхідність виділяти односкладові слова - саме з ними і пов'язано максимальне число помилок реально існуючих систем. Можна вимагати, щоб людина вимовляв слова по одному, роблячи досить тривалі паузи або щоб кожне наступне слово вимовлялося після звукового сигналу. Але даний підхід не зручний і може бути застосований лише для подачі простих команд.

Наступна проблема - різниця голосів, діалектів, дикція, вікових відмінностей, емоційний і фізичний стан диктора. Значний вплив вносить акустичний аспект, тобто зміна мікрофона, розташування мікрофона щодо рота, акустична обстановка в приміщенні.

Саме через ці та багатьох інших проблем до повного вирішення задачі розпізнавання мови і раніше досить далеко. Існує два істотно розрізняються режими роботи: з налаштуванням на голос певного диктора і без такого налаштування. Розміри словника при роботі з налаштуванням на диктора (speaker-dependent) в даний час можуть досягати декількох (і навіть багатьох) тисяч слів при злитому проголошенні. Процедура настройки на диктора виглядає наступним чином: диктор читає якийсь спеціальним чином складений текст, комп'ютер розпізнає слова і видає варіант розпізнавання. Диктор позначає помилки і читає текст знову. Після кількох таких ітерацій процес сходиться, і комп'ютер виявляється в стані

розпізнавати мовлення.

Нарешті, останній, найбільш складний для реалізації, але водночас і найбільш перспективний режим роботи - розпізнавання без настройки на диктора. При цьому гарантується, що система розпізнає будь-яке включене в словник слово, ким би воно не було вимовлено. Тут, як правило, словники налічують невелику кількість слів (зазвичай не більше двох десятків) і існують для відносно невеликого числа мов (приблизно тридцяти). Українська мова в це число хоча і входить, проте кількість розпізнаваних українських слів невелика.

Створення словника для розпізнавання мови без настройки на голос вимагає великих витрат. Для вирішення цього завдання розробникам доводиться опитувати велику кількість (кілька сотень або тисяч) носіїв мови, виділяти якісь загальні елементи мови, усереднювати їх - і все це для того, щоб забезпечити розпізнавання десяти-двадцяти слів. Найчастіше словник без настройки на голос користувача вимагає роздільного проголошення слів. Для цілого ряду додатків цього, однак, виявляється цілком достатньо.

Розпізнавання мовлення часто називають терміном "розпізнавання мови". Це не зовсім правильно, оскільки існує окрема задача розпізнавання мови, що передбачає відповідь на запитання, якою мовою розмовляє користувач, якого ми називатимемо терміном "диктор". Інколи вживається термін "розпізнавання голосу". Це може означати і введення тексту голосом, і ідентифікацію людини за голосом, і виділення голосових сегментів у звуковому сигналі.

Загалом, метою розпізнавання мовлення є отримання різного роду інформації на основі вхідного мовленнєвого (голосового) сигналу: про що говориться, хто говорить, якою мовою, в якому фізичному стані перебуває диктор тощо.

Ось доволі вичерпний перелік проблем, які вирішуються в ділянці розпізнавання та розуміння мовлення:

- автоматичне перетворення мовленнєвого сигналу на текст;
- введення інформації голосом, диктувальна машина;
- пошук ключових слів і фраз у потоці мовлення;
- смислова інтерпретація голосових повідомлень;
- ідентифікація та верифікація диктора;
- адаптація до голосу диктора та акустичного каналу;
- розпізнавання мови, якою говорить диктор, його акценту;
- усний переклад з однієї мови на іншу;
- розпізнавання емоційного та фізичного стану мовця.

Завдяки розпізнаванню мовлення вивільняються руки користувача при керуванні комп'ютерними системами, введенні текстової інформації, транскрибуванні (стенографуванні) фонограм тощо. Вже тепер починають з'являтися системи, що допомагають в оволодіванні розмовною іноземною мовою на основі технології розпізнавання мовлення. Велике майбутн

-

ля

захисту персональної інформації.

Якщо поруч із звуковим нам доступний зоровий канал, то його можна використовувати як додаткову інформацію при вирішенні наведених задач. В такому разі йдеться про технології мультимодального розпізнавання та розуміння мовлення. А при поєднанні технологій розуміння мовлення та синтезу мовлення за текстом виникає система усного діалогу.

1.2 Особливості розпізнавання голосових сигналів в системах управління електронними обчислювальними машинами

Ідея створення системи керування електронним пристроєм, що базується не тільки на тактильній взаємодії людина- машина, але і на голосовому керуванні не нова. Комп'ютерні системи розпізнавання мови поступово знаходять застосування не тільки в науковій сфері, але й у побутовій. Прикладом тому можуть служити офісні пакети й інше ПЗ з убудованим розпізнаванням мови для голосового введення текстової інформації. Що ж стосується портативних пристроїв, то в них лише зараз починають упроваджуватися технології розпізнавання мови. На сьогоднішній день в умовах глобального розвитку інформатизації, конвергенція технологій голосового керування з мультимедійними функціями портативної і побутової техніки обумовлює науково-технічний прогрес у створенні нової функціональності передової техніки. Однією з проблем упровадження голосових технологій у портативній техніці був низький обчислювальний ресурс мікропроцесорів і недостатній обсяг оперативної пам'яті. Крім цього алгоритми з достатньою надійністю розпізнавання мови в умовах складної шумової обстановки навколишнього середовища були занадто ресурсоємні для портативного застосування. 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 того щоб машина навчилась розуміти людську мову, відповідати

на питання потрібно затратити багато сил і часу, забиваючи її гігантською інформацією тільки для того, щоб розпізнати окремі звуки. У кожного звуку складна структура, яка включає в себе різні частоти і коливання, до того ж, те саме слово різні люди вимовляють по-різному: різний тембр голосу, різні інтонації, різна чистота вимови. Скільки людей, стільки й голосів. Голос – індивідуальна ознака особистості. Щоб навчити машину впізнавати мову, її потрібно заставити прослуховувати слова, сказані як однією людиною, так і різними людьми. Задача машини – прослухавши всі дані, взяти середні значення особливостей вимови, повністю виключити індивідуальність, щоб потім, почувши слово, не зробити помилку. Найбільші проблеми виникають в умовах:

- довільний користувач;
- спонтанна мова, яка супроводжується мовленнєвим «сміттям»
- наявність акустичних завад і скривлень;
- наявність мовленнєвих завад.

Для спрощення процесу розпізнавання мови доцільно було б використовувати шаблони окремих звуків єдині для всіх дикторів. На даний час таких шаблонів не існує через те, що не виявлено інформативних ознак звуків, які не залежать від характерних особливостей голосу. Тому для реалізації ефективних систем автоматизованого розпізнавання мови, що не залежать від диктора, необхідно виділити інформативні ознаки звуків мови, розробити математичні методи їх опрацювання з метою створення єдиних для всіх дикторів шаблонів. У такому випадку система розпізнавання не буде потребувати навчання (створення набору шаблонів окремо для кожного диктора), її швидкодія збільшиться, оскільки відпаде необхідність створення набору шаблонів слів і з’явиться можливість розпізнавати мову незалежно від характерних особливостей голосу диктора.

Використання сучасних, але високоінтелектуальних інформаційних комп'ютерних технологій у сфері людської діяльності вимагає кардинальної зміни в управлінні автоматизованими системами для більш зручного та раціонального їх використання. Необхідність в мовленнєвому спілкуванні з комп'ютером є абсолютно природною. Найбільшою мірою її стимулює не стільки бажання створити більші зручності користувачу, скільки існування специфічних областей комп'ютеризації, де голосові команди є найбільш придатними чи навіть єдиним можливим рішенням. До них можна віднести голосовий доступ до автоматичних довідкових систем, керування віддаленим комп'ютером чи портативним пристроєм, що відбувається під час руху. Створення повноцінних мовленнєвих інтерфейсів, які підтримують діалог «користувач – комп'ютер» є дуже перспективним, але надзвичайно складним напрямом розвитку сучасних комп'ютерних систем, що стикаються з велетенською кількістю проблем на шляху їх вирішення. На сьогодні, під поняттям «розпізнавання голосу» прихована ціла сфера наукової та інженерної діяльності. В цілому, завдання розпізнавання голосу зводиться до того, щоб виділити, класифікувати та відповідним чином відреагувати на людський голос з вхідного звукового потоку. Це може бути виконання певної дії на команду людини чи виокремлення певного слова-маркера з великого масиву телефонних розмов, чи система для голосового вводу тексту. Також всім відомі програми голосової ідентифікації користувачів, що реалізовані в деяких системах безпеки. Потенційно, сфера використання голосового розпізнання надзвичайно широка, але, на жаль, на даний момент не може бути реалізована внаслідок слабкої стійкості самих систем розпізнання мови до різних факторів

Кожна система розпізнання мови має певні задачі, які вона створена вирішувати, та комплекс методів котрий використовується для рішення

цих задач. Класифікація систем розпізнання мови буде проводитися згідно нового стандарту прийнятого в галузі програмування таких систем — Microsoft Speech API. Згідно з цим стандартом системи розпізнання мови розрізняються за певними ознаками.

- Інтервал між окремими словами. Якщо система розпізнає злиту мову, користувач може вимовляти фрази в природному вигляді, не роблячи проміжків між словами. Неперервне розпізнання має перевагу, але його реалізація більш складна та вимагає більших апаратних можливостей комп'ютерів, результатом чого є мала кількість таких систем. В системах, що працюють з дискретною мовою диктор має робити паузи між окремими словами, як правило не менше 1/4 секунди. Третім різновидом є системи, які виділяють одне слово – маркер, в певному мовленнєвому інтервалі. Навіть, якщо маркер знаходиться всередині фрази вимовленої здільно.
- Залежність від диктора. За визначенням система залежна від диктора призначена для використання одним користувачем, в той час, як альтернативні системи призначені для роботи з будь-яким диктором. Незалежність від диктора – складна задача оскільки під час навчання системи вона налаштовується на параметри голосу диктора, на прикладі якого вона навчається. Кількість помилок в таких системах, як правило в 4-5 разів більша, ніж в системах залежних від диктора. Системи, що володіють відносною незалежністю від диктора, дозволяють працювати з ними без попереднього налаштування, навчання системи, однак результати все таки є кращими, за умови навчання системи. Незалежність від диктора, як правило, досягається за рахунок збереження звукових еталонів для всіх найбільш типових голосових носіїв даного типу, що в результаті ставить більші апаратні вимоги до таких систем. Процес навчання, налаштування під диктора, як правило, займає від 30 хв. до кількох годин. Саме цей факт є головною незручністю для користувачів. Третім різновидом за даною

ознакою є системи, що автоматично налаштовуються на голос диктора в процесі їх експлуатації. У систем такого типу є дві особливості: їм необхідно знати чи зробив користувач помилку, вимовляючи те чи інше слово (інакше навчання буде не вірним); після налаштування на конкретного диктора, ці системи стають менш надійними при роботі з іншим диктором.

- Ступінь деталізації при задаванні еталонів. Розрізняють алгоритми, в яких за еталони приймають цілі слова та алгоритми, що використовують в якості еталонів частини слів. Порівняння цілих слів дає більшу точність, швидкість, але при цьому вимагає більшого обсягу пам'яті. Алгоритми порівняння елементів слів (фонем, складів і т.д.) доводиться використовувати у випадку великих словників, оскільки об'єм необхідної пам'яті пропорційний кількості цих еталонних слів та не залежить від об'єму словника.
- Розмір словника. Системи розпізнання можуть використовувати як великі, так і маленькі словники. Системи, що працюють з маленькими словниками (близько 50 слів), дозволяють користувачу давати комп'ютеру прості команди. Для диктування текстів необхідний великий словник (десятки тисяч слів). Очевидно, що чим більший розмір словника, котрий закладено в систему розпізнання, тим більша частота помилок під час роботи системи. Наприклад, словник із 20 слів може бути розпізнано майже без помилок, тоді як частота помилок при роботі зі словником в 1000 слів може досягати 45%. З іншого боку, навіть розпізнання невеликого словника може дати велику кількість помилок, якщо слова в даному словнику дуже схожі одне на одне.

Не дивлячись на те, що в теорії можлива будь- яка комбінація даних характеристик, на практиці найбільш популярними є системи голосового управління комп'ютером та систем дискретного диктування тексту. У

процесі створення системи розпізнання голосу потрібно обрати рівень абстракції адекватний поставленій задачі. Параметри звукової хвилі мають використовуватися для розпізнання та методів розпізнання цих параметрів. Можна виокремити таку основну різницю в структурі і процесі роботи різноманітних систем розпізнання голосу:

- За типом структурної одиниці. У процесі аналізу голосу, як базові одиниці можуть бути обрані окремі слова чи частини вимовлених слів: фонеми, дичи, трифони, аллофони. Залежно від того, яка структурна частина обрана, змінюється структура, універсальність та складність словника елементів, що розпізнається.
- За виділенням ознак. Сама послідовність відрізків тиску звукової хвилі – надмірно збиткова для систем розпізнавання звуків та містить багато зайвої інформації, яка для розпізнання не потрібна чи навіть шкідлива. Таким чином, для представлення голосового сигналу з нього слід виокремити усі параметри, що адекватно представляють даний сигнал для розпізнання.
- За механізмом функціонування. В сучасних системах широко використовуються різноманітні підходи до механізму функціонування розпізнавальних систем. Імовірнісно-мережевий підхід полягає в тому, що голосовий сигнал розбивається на певні частини (кадри або за фонетичною ознакою), після чого імовірнісна оцінка того, до якого саме елементу словника, що розпізнається має відношення дана частина і (чи) весь вхідний сигнал. Підхід, оснований на рішенні зворотної задачі синтезу звука, полягає в тому, що за вхідним сигналом визначається характер руху артикулярів мовного каналу та за спеціальним словником відбувається визначення вимовлених фонем. Для кращого розуміння особливостей задач розпізнання мови слід відмітити, що основна маса систем працюють практично однаково, використовуючи переважно одні й ті ж методи та

алгоритми. Різниця полягає в манері диктування голосу, розмірі словника, ступені фільтрації вхідного сигналу, обумовлена лише специфікою задачі та наявними технічними можливостями. Якщо спробувати представити спрощено процес розпізнання голосу, то він може бути описаний в послідовності таких кроків:

фільтрація шуму та виокремлення необхідного сигналу;

- перетворення вхідного голосового сигналу в набір акустичних параметрів;
- приведення акустичної форми сигналу до внутрішнього алфавіту еталонних елементів;
- розпізнання послідовності фонем та перетворення їх на слова.

1.3 Аналіз існуючих рішень розпізнавання голосових сигналів

Обробка голосового сигналу починається з його оцифровки. Для цього необхідно заздалегідь записати його в оперативну пам'ять комп'ютера або на машинний носій. Як було сказано вище, більшість персональних комп'ютерів вже оснащені обладнанням, необхідним для введення і виведення звуку. Це мікрофон і звукова плата. У загальному вигляді процес введення мовних повідомлень показаний на рис. 1.1.

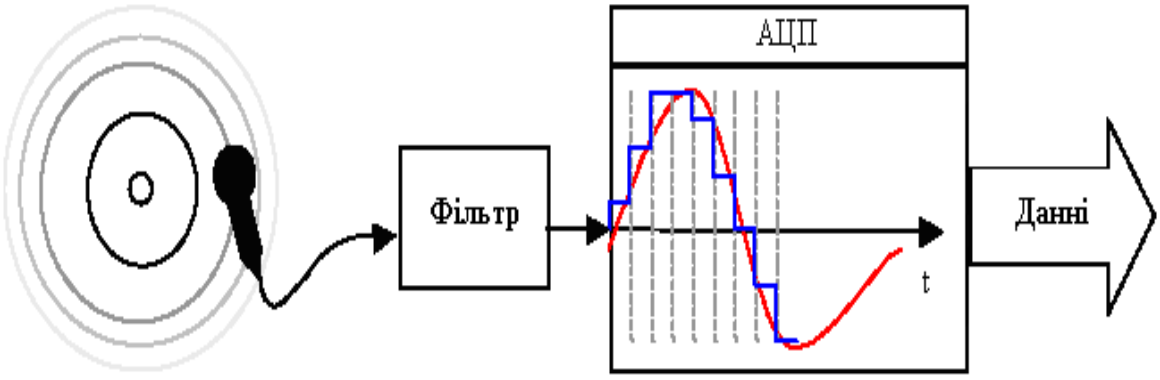

Рисунок 1.1 – Схема вводу голосових повідомлень в персональний комп’ютер

Голосовий сигнал формується і передається в просторі у вигляді звукових хвиль. Джерелом голосового сигналу є мовоутворюючий тракт людини. Приймачем сигналу є датчик звукових коливань, мікрофон – пристрій перетворення звукових коливань в електричні. Існує велика кількість типів мікрофонів (вугільні, електродинамічні, електростатичні, п'єзоелектричні та ін. Деякі з цих мікрофонів для своєї роботи вимагають зовнішнього джерела струму (наприклад, вугільні та конденсаторні), інші під впливом звукових коливань здатні самостійно виробляти змінна електрична напруга (це електродинамічні і електретні мікрофони). Є також мікрофони, призначені спеціально для комп'ютерів. Такі мікрофони зазвичай кріпляться на підставці, що стоїть на поверхні столу. Комп'ютерні мікрофони можуть комбінуватися з головними телефонами.

Як же вибрати з усього різноманіття мікрофонів той, що найкраще підходить для систем розпізнавання мовлення? В принципі, можна експериментувати з будь-яким наявним мікрофоном, якщо тільки його можна підключити до звукового адаптера комп'ютера. Однак розробники систем розпізнавання мовлення рекомендують придбати такий мікрофон, який при роботі буде перебувати на постійному відстані від рота диктора.

Якщо відстань між мікрофоном і ротом не змінюється, то середній рівень електричного сигналу, що надходить від мікрофона, також буде мінятися не занадто сильно. Це матиме позитивний вплив на якість роботи сучасних систем розпізнавання мови. Якщо мікрофон стоїть на столі, то при повороті голови або зміні положення тіла відстань між ротом і мікрофоном буде змінюватися. Це призведе до зміни рівня вихідного сигналу мікрофона, що, в свою чергу, погіршить надійність розпізнавання мовлення. Тому при роботі з системами розпізнавання мовлення найкращі результати будуть досягнуті, якщо використовувати мікрофон,

прикріплений до головних телефонів. При використанні такого мікрофона відстань між ротом і мікрофоном буде постійним.

Чутливим елементом мікрофона будь-якого типу є пружна мембрана, яка залучається до коливальний процес під впливом звукових хвиль. Мембрана пов'язана з перетворюючим елементом, який перетворює коливання мембрани в електричний сигнал.

З виходу мікрофону сигнал подається на вхід звукової карти персонального комп'ютера. Величина вхідного сигналу, що надходить від мікрофона, змінюється періодично і приймає як позитивні, так і негативні значення. При записі звукова карта представляє собою аналого-цифровий перетворювач з можливостями налаштування параметрів оцифровки.

Основними параметрами є частота дискретизації і розрядність кодування. Дані параметри визначають якість і розмір одержуваної вибірки в результаті запису. Вибір частоти дискретизації безпосередньо залежить від узагальненої спектральної щільності потужності мовного сигналу (рис. 1.2). Узагальнена спектральна щільність потужності має максимум в діапазоні 250-500 Гц і затухає зі швидкістю, що дорівнює 8-10 дБ на октаву (при подвоєнні частоти). Це призводить до того, що на частотах вище 4000 Гц спектральна щільність падає до рівня 60 дБ, що відповідає послабленню потужності в порівнянні з максимумом (-25 ... -30 дБ) в 20 і більше разів. Це дозволяє вважати, що смуга пропускання для каналів передачі звукових повідомлень може бути обмежена частотою 4-5 кГц. Відповідно теореми Котельника, частота дискретизації цього сигналу повинна становити 8-10 кГц. Зазначимо, що частота дискретизації 8 кГц являється стандартною для телефонних апаратів.

В звукових картах персональних комп’ютерів ця частота є мінімальною, при цьому передбачена можливість суттєво підвищити частоту дискретизації. Таким чином, можна вважати, що звукова карта

персонального комп’ютера дозволяє з достатньою якістю дискретизувати голосовий сигнал.

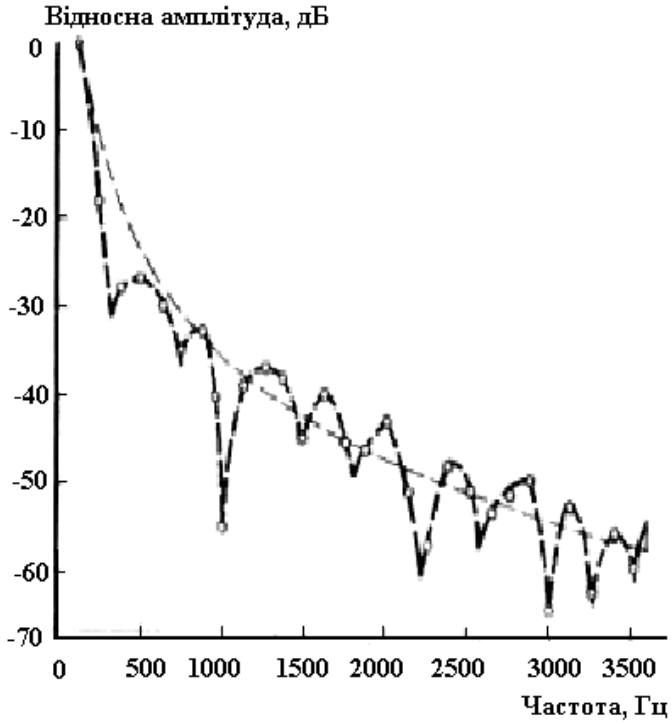

Рисунок 1.2 – Графік амплітудного спектру потужності голосового сигналу

В сучасних звукових картах використовується імпульсно-кодова модуляція, при якій кожен дискретний відлік мовного повідомлення кодується відповідно з деякими правилами. Розраховано, що для забезпечення співвідношення сигнал/шум квантування, рівного 36 дБ, потрібно не менше семи двійкових розрядів і що для отримання високоякісного цифрового кодування сигналу мовлення необхідно 11 розрядів. На практиці кількість розрядів визначається розрядністю персонального комп’ютера. Як правило, вона дорівнює або кратна восьми розрядам. При використанні найбільш поширеного програмного забезпечення кількість розрядів дорівнює 16, 32 або 64. Першочергова фільтрація шумів в дискретизованому сигналі полягає в накладенні на цей сигнал вікон різного типу – Кайзера, Хемінга та інших. В доступній

літературі не знайдено критеріїв, по яким вибирається тип вікна.

Після фільтрації шумів для виділення із звукового потоку окремих слів застосовується аналіз енергії сигналу на протязі кожних 10-20 мс. Крім того, визначити початок/кінець слова можна по всплеску/затуханню величини сигналу. Обробка вхідного оцифрованого сигналу з метою зменшення обсягу вхідних даних полягає у застосуванні різних методів спектрального аналізу даних. Спектральний аналіз даних реалізується або за допомогою віконного дискретного перетворення Фур'є або за допомогою дискретних вейвлет-перетворень. Зазначимо, що діапазон частот, які чує людина, знаходиться в межах від 16 Гц до 20000 Гц. Однак в більшості систем, виходячи в тому числі і із можливостей комп'ютерної звукової апаратури, застосовується діапазон частот від 50 Гц до 16000 Гц. На сьогодні загальноприйнято проводити стиснення спектру за допомогою процедури визначення мел-кепстральних коефіцієнтів. Її результатом являється 16-24 коефіцієнтів, які достатньо повно характеризують весь діапазон звукових частот, які відчуває людина. Ще один підхід стиснення спектру базується на методі визначення формант голосового сигналу [7, 8].

Для порівняння еталонного та піддослідного сигналів в теперішній час в основному використовуються сховані марківські моделі, методи динамічного програмування та нейронні мережі.

Використання схованих марківських моделей базується на постулаті, що голосовий сигнал може бути розділений на стаціонарні фрагменти, які відповідають окремим станам ланцюга Маркова

$$O = \{o_1, o_2, ..., o_\tau\}. \quad (1.1)$$

Перехід між станами відбувається миттєво, а ймовірність відображення породженого моделлю фрагменту залежить тільки від поточного стану моделі та не залежить під попередніх станів.

Власне схованою марківською моделлю називається модель, що складається із N станів, в кожному із яких деяка система може приймати одне з M значень деякого параметра. Як правило, модель задається виразом

$$\lambda = \{A, B, \pi\}, \tag{1.2}$$

де A – матриця ймовірностей переходів по станам, B – вектор ймовірностей випадіння кожного із M значень параметру в кожному із N станів, π – вектор розподілу початкових ймовірностей. Перший етап використання моделі полягає у її навчанні на прикладах, що відповідають еталону ключового слова.

Результатом навчання являється розрахунок параметрів виразу (1.2). Після цього на вхід моделі можна подавати послідовність, яка відповідає невідомому голосовому сигналу. Вирішення задачі знаходження ймовірності появи цієї послідовності у кожній із попередньо навчених моделей дозволить визначити ту модель, яка найбільш достовірно відповідає голосовому сигналу, а значить, і розпізнати ключове слово. До основних недоліків схованих марківських моделей відносять велику обчислювальну складність та складність формування бази даних ключових слів. Застосування методів динамічного програмування зводиться до розрахунку ключового слова найбільш схожого на невідоме.

Критеріями схожості слів виступають відстань Евкліда, відстань Хемінга та інші. В доступній літературі методики вибору критерію схожості не знайдено.

Використання нейронних мереж базується на їх здатності класифікувати голосові сигнали, задані за допомогою коефіцієнтів, які відповідають спектральним характеристикам [10]. Не зважаючи на перспективність даного напрямку, застосуванню нейронних мереж перешкоджає відсутність методики оптимізації типу та параметрів мережі.

В підсумку можна зазначити, що основною невирішеною задачею в області пошуку ключових слів є задача порівняння еталонних та невідомих голосових фрагментів.

На сьогодні перераховані технології розпізнавання мовних сигналів реалізовані в наступних програмних комплексах:

- програми голосового управління комп'ютером VoiceNavigator, Truffaldino;
- бібліотека розпізнавання голосових команд VoiceCom;
- систему голосового розмежування доступу, розроблену компанією «Центр мовних технологій»;
- програми документування усних виступів - комп'ютерний транскрайбер, системи Нестор і Алегро;

Програма VoiceNavigator – є типовим представником програм голосового управління комп'ютером. Вона дозволяє користувачеві запускати додатки голосом, не торкаючись клавіатури, і виконувати довільно задані команди. Перед використанням програми VoiceNavigator її необхідно навчити, вимовивши в мікрофон слова команд. Так як програма VoiceNavigator розпізнає команди за зразками, то команди можна вимовляти будь-якою мовою і будь-яким голосом.

Щоб програма почала розпізнавати голосові команди, її необхідно "розбудити", вимовивши ключове слово. Після цього програма буде реагувати тільки на ваші команди, ігноруючи інші звуки. У програмі є функція голосової відповіді-підтвердження команд. Ця функція дозволяє переконатися, що ваша команда розпізнана системою і готова для виконання. Програма VoiceNavigator невимоглива до ресурсів комп'ютера. Ви можете використовувати її в комп'ютері, обладнаному процесором з тактовою частотою 200 МГц або вище, причому для введення звукових команд підійде будь-який звуковий адаптер, наприклад, Creative Sound

Blaster.

Бібліотека розпізнавання голосових команд VoiceCom. – становить ядро описаних вище програм VoiceNavigator і Truffaldino. З її допомогою розробники можуть додати голосове управління в створювані ними програми. Скориставшись готової бібліотекою VoiceCom, розробники можуть легко додати у додатки наступні функціональні можливості: • управління обладнанням за допомогою голосу; виконання мовних запитів до баз даних через мікрофон або навіть по телефону; пошук за ключовими словами в звукових WAV-файлах. Слід зазначити, що бібліотека VoiceCom дозволяє вбудовувати голосові функції не тільки в звичайні програми для персональних комп'ютерів, але і в автономні пристрої, обладнані цифровими сигнальними процесорами DSP. Алгоритми, реалізовані в бібліотеці розпізнавання голосових команд VoiceCom, мають високу швидкодію, невибагливі до обсягу оперативної пам'яті і здатні адаптуватися до шумів. Бібліотека VoiceCom забезпечує розпізнавання команд, виголошених будь-яким голосом і будь-якою мовою. При цьому є можливість структурування для практично необмеженого словника. При цьому алгоритми дозволяють розпізнавати 100-200 команд з попереднім навчанням для кожного диктора, і 30-50 команд для будь-якого диктора (в режимі, не залежному від диктора). Якщо команди вимовляються по телефону, то алгоритми бібліотеки VoiceCom дозволяють розпізнати 10-20 слів, вимовлених яким диктором. Ну і, звичайно, у бібліотеці реалізована можливість активації розпізнавання команд за ключовим словом, що виключає несподівані реакції системи на сторонні звуки.

Голосове розмежування доступу за допомогою бібліотеки VoiceKey Kit. Компанія "Центр мовних технологій" створила бібліотеку розмежування доступу по голосу VoiceKey Kit, яку можна легко вмонтувати в будь-які додатки. Це можуть бути офісні додатки,

комп'ютерні ігри, системи "батьківського контролю" та ін. Ця бібліотека дозволяє розпізнавати парольні фрази (типу "Сезам, відкрийся!"), або особливості голосу тієї чи іншої людини. В якості її недоліків вказують низьку надійність розпізнавання.

Системи Нестор і Алегро. Система Нестор призначена для багатоканальної цифрової звукозапису та оперативного текстового розшифрування кількох усних виступів і фонограм мови за принципом розподіленої обробки (стенографування). Комплекс Нестор забезпечує синхронну обробку до 24 акустичних каналів (виступаючих і/або фонограм мови). В цю систему входить комп'ютер станції звукозапису, обладнаний 4-каналоним звуковим адаптером і спеціалізованим програмним забезпеченням і звуковий сервер для архівування звукових записів на диски CD-RW. Комплекс Нестор може комплектуватися педаллю для управління відтворенням звукового сигналу. В системі передбачені робочі місця адміністратора, керівника групи та операторів. В залежності від варіантів поставки в комплексі може бути від 3 до 50 робочих місць операторів, від 1 до 8 робочих місць керівників груп і 1-2 робочих місця адміністратора. Таким чином, система Нестор придатна для автоматизованого документування досить великих нарад і форумів. Поступаючий на її вхід мовної сигнал записується на жорсткий диск комп'ютера. Потім він розбивається на фрагменти і розподіляється між операторами-стенографістами, які виконують його прослуховування і розшифрування. Отримані таким чином ділянки тексту автоматично з’єднуються в єдиний документ, який після перевірки може бути збережений і роздрукований.

Програми для диктування тексту. Сьогодні існують потужні програми, здатні не тільки розпізнавати і виконувати команди, а й розпізнавати мову в режимі диктування. Як правило, такі програми або

забезпечуються власним редактором тексту, або здатні працювати з будь-якими редакторами тексту і таблиць, наприклад, такими, як Microsoft Word і Microsoft Excel. Крім того, система для диктування тексту входить до складу браузеру Chrome та сучасних версій операційної системи Android. Основним недоліком вказаних систем є високі обчислювальна ресурсоємність.

1.4 Постановка задач дослідження

В результаті проведеного дослідження можна сформулювати такі вимоги до комп'ютерної системи розпізнавання ключових слів:

- використання швидкого перетворення Фур'є для спектрального аналізу звукового сигналу;
- використання мел-кепстрального згладжування спектру голосового сигналу з метою зменшення обсягу вхідних параметрів системи порівняння еталонного та піддослідного голосових сигналів;
- використання методів та моделей теорії нейронних мереж для порівняння еталонного та піддослідного голосових сигналів;
- програмний додаток повинен відповідати сучасним вимогам до надійності, швидкості та ресурсоємності;
- програмний додаток повинен бути орієнтований на використання в комп'ютерних системах, що функціонують під управлінням операційних систем сімейства Windows.

Крім того проведений аналіз задачі розробки інформаційної системи розпізнавання голосових сигналів дозволяє виділити такі етапи її вирішення:

1. Побудова архітектури системи розпізнавання.
2. Розробка математичного забезпечення інформаційної системи.

3. Розробка інформаційного забезпечення.

4. Розробка програмного забезпечення.

5. Експериментальне підтвердження отриманих результатів.

2. МЕТОД ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ГОЛОСОВИХ СИГНАЛІВ

2.1 Опис інструментарію

Операційною системою для розробки було обрано Ubuntu GNU/Linux.

Linux - сімейство Unix-подібних операційних систем на базі ядра Linux, що включають той чи інший набір утиліт і програм проекту GNU, і, можливо, інші компоненти. Як і ядро Linux, системи на його основі як правило створюються і поширюються відповідно до моделі розробки вільного та відкритого програмного забезпечення. Linux-системи поширюються в основному безкоштовно у вигляді різних дистрибутивів - у формі, готової для установки і зручною для супроводу і оновлень, - і мають свій набір системних і прикладних компонентів, як вільних, так можливо і власницьких.

Ubuntu – один з таких дистрибутивів. Ubuntu орієнтована на зручність і простоту використання. Вона включає широко поширене використання утиліти sudo, яка дозволяє користувачам виконувати адміністраторські завдання, що не запускаючи потенційно небезпечну сесію суперкористувача.

Ubuntu, крім того, має розвинену інтернаціоналізацію, що забезпечує максимальну доступність для представників різних мовних груп.

Ubuntu для роботи рекомендується від 512 мегабайт RAM і, при установці на жорсткий диск, від п'яти гігабайт вільного простору, а гранично мінімальні вимоги набагато нижче.

Як вже було зазначено, для розробки системи використано мову програмування Python.

Python - високорівнева мова програмування загального призначення,

орієнтована на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує кілька парадигм програмування, в тому числі структурне, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване програмування. Основні архітектурні риси - динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень і зручні високорівневі структури даних. Код в Python організовується у функції та класи, які можуть об'єднуватися в модулі (вони в свою чергу можуть бути об'єднані в пакети).

Еталонної реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється під вільною ліцензією Python Software Foundation License, що дозволяє використовувати його без обмежень в будь-яких додатках, включаючи пропрієтарні. Є реалізації інтерпретаторів для JVM (з можливістю компіляції), MSIL (з можливістю компіляції), LLVM та інших. Проект PyPy пропонує реалізацію Python з використанням JIT-компіляції, яка значно збільшує швидкість виконання Python-програм.

Python – мова програмування, що активно розвивається, нові версії (з додаванням / зміною мовних властивостей) виходять приблизно раз в два з половиною роки. Внаслідок цього і деяких інших причин на Python відсутні стандарт ANSI, ISO або інші офіційні стандарти, їх роль виконує CPython.

Оскільки Python — інтерпретована мова, математичні алгоритми, часто працюють в ньому набагато повільніше ніж у компільованих мовах, таких як C або навіть Java. NumPy намагається вирішити цю проблему для великої кількості обчислювальних алгоритмів забезпечуючи підтримку

багатовимірних масивів і безліч функцій і операторів для роботи з ними. Таким чином будь-який алгоритм який може бути виражений в основному як послідовність операцій над масивами і матрицями працює також швидко як еквівалентний код написаний на C.

NumPy можна розглядати як гарну вільну альтернативу MATLAB, оскільки мова програмування MATLAB зовні нагадує NumPy: обидві вони інтерпретовані, і обидві дозволяють користувачам писати швидкі програми поки більшість операцій проводяться над масивами або матрицями, а не над скалярами. Перевага MATLAB у великій кількості доступних додаткових тулбоксів, включаючи такі як пакет Simulink. Основні пакети, що доповнюють NumPy, це: SciPy — бібліотека, що додає більше MATLAB-подібної функціональності; Matplotlib — пакет для створення графіки в стилі MATLAB. Внутрішньо як MATLAB, так і NumPy базується на бібліотеці LAPACK, призначеної для вирішення основних задач лінійної алгебри.

TensorFlow — відкрита програмна бібліотека для машинного навчання цілій низці задач, розроблена компанією Google для задоволення її потреб у системах, здатних будувати та тренувати нейронні мережі для виявлення та розшифровування образів та кореляцій, аналогічно до навчання й розуміння, які застосовують люди. Її наразі застосовують як для досліджень, так і для розробки продуктів Google, часто замінюючи на його ролі її закритого попередника, DistBelief. TensorFlow було початково розроблено командою Google Brain для внутрішнього використання в Google, поки її не було випущено під відкритою ліцензією Apache 2.0 9 листопада 2015 року.

TensorFlow забезпечує ППІ для Python, а також для C++, Haskell, Java та Go.

Серед застосувань, для яких TensorFlow є основою, є програмне

забезпечення автоматизованого опису зображень, таке як DeepDream. 26 жовтня 2015 року Google офіційно реалізувала RankBrain, який підтримує TensorFlow. RankBrain тепер обробляє суттєве число пошукових записів, замінюючи та доповнюючи традиційні статичні алгоритми на основі результатів пошуку.

2.2 Визначення перспективних нейромережевих архітектур

Проведений аналіз сучасного стану найромережевих технологій дозволяє сформулювати висновок про те, що доцільність застосування конкретного типу НМ слід визначати на основі співставлення характеристик мережі з умовами прикладної задачі. До вказаних характеристик та умов відносяться:

- параметри навчальних даних,
- загальні обмеження процесу навчання,
- вимоги до обчислювальних потужностей,
- вимоги до вихідної інформації,
- обмеження технічної реалізації НМ,
- сфера застосування.

Розглянемо вказані характеристики в ракурсі комп'ютерної системи розпізнавання голосових сигналів.

До основних параметрів навчальних даних відносяться:

- Кількість параметрів, що характеризують навчальний приклад.
- Вид параметрів, дискретний (символьний) чи безперервний (числовий).
- Загальна кількість навчальних прикладів.
- Наявність помилок (шуму) в навчальних прикладах.
- Наявність кореляції навчальних прикладів.

- Можливість та необхідність попередньої обробки вхідних даних з метою їх нормалізації та видалення шуму.
- Повнота вибірки, тобто можливість відображення в ній всіх аспектів процесу, що моделюється.
- Пропорційність навчальних прикладів, що відповідають різним аспектам процесу, що моделюється.

Загальні обмеження процесу навчання обумовлюються:

- Максимальним терміном навчання.
- Необхідністю представлення в навчальних даних очікуваного вихідного сигналу НМ. Цим визначається тип навчання – з вчителем або без вчителя.
- Можливістю автоматизації процесу навчання, яка визначається кількістю та важливістю емпіричних параметрів. Вказана можливість багато в чому визначає умови застосування НМ. Мережі в яких процес навчання не автоматизовано можуть використовуватись тільки в лабораторних умовах.
- Можливістю донавчання в процесі експлуатації.
- Вимогами до якості навчання, яке звичайно оцінюють по величині максимальної та середньої помилки розпізнавання навчальних та тестових даних. При цьому тестові дані повинні не значно відрізнятись від навчальних.
- Можливістю навчання НМ в лабораторних умовах. Доцільність навчання в лабораторних умовах пояснюється потребами оптимального механізму створення та оновлення бази знань НМ.

На практиці вимоги до обчислювальних потужностей визначаються

максимальною кількістю прикладів (обсяг пам'яті), яку може запам'ятати мережа для досягнення необхідної достовірності прийняття рішення. В свою чергу достовірність прийняття рішення характеризується допустимими величинами максимальної та середньої помилки мережі на реальних даних які в загальному випадку можуть виходити за межі множини навчальних даних. Відповідно виникає задача екстраполяції результатів навчання НМ за межі навчальних прикладів. Відзначимо, що обчислювальна потужність мережі залежить від її типу та алгоритму навчання. Ще однією вимогою може бути незмінність виходу мережі для різних прикладів з однаковими параметрами.

Вимоги до вихідної інформації НМ вказують на те в якому вигляді має бути представлена ця інформація. Наприклад, при розпізнавані слів може виникнути необхідність не тільки визначення ситуації “слово А присутнє”, але й розрахунку ймовірності появи цієї ситуації. Також вимогою може бути необхідність визначення вербальних залежностей між вхідною та вихідною інформацією.

Обмеження технічної реалізації НМ стосуються: швидкості прийняття рішення, інтеграції в існуючі ЗЗІ, обсягу та складності програмної реалізації. Для зменшення обсягу можливо розділити програмний код для навчання мережі від коду, що відповідає за її функціонування.

Сфера застосування визначає ЗЗІ в яких буде використовуватись НМ. На сьогодні достатньо дослідженим є використання НМ для розпізнавання образів та при проведенні оптимізаційних розрахунків. Відзначимо, що системи розпізнавання образів принципово відрізняються від систем аналізу тексту тим, що в них кількість вихідних та кількість комбінацій вхідних параметрів принципово обмежена. В системах аналізу тексту ця кількість принципово необмежена. В перспективі доцільно

застосувати НМ з метою реалізації паралельних розрахунків в КС, що дозволить значно підвищити їх стійкість від багатьох типів атак з метою відмови в обслуговуванні.

Крім того сфера застосування визначається пристосованістю мережі до автономного функціонування. Для цього в архітектурі НМ повинно бути передбачено можливість повної автоматизації процесу донавчання на експлуатації.

Якісні оцінки відповідності основних характеристик НМ умовам задач захисту ПЗ для описаних в п. 1.1-1. 8 типів мереж наведені в табл. 2.1. В табл. 2.1 відсутні характеристики, які хоча і застосовуються при побудові мережі, але не впливають на вибір типу НМ. Оцінки відповідності виставлені в числовому вигляді по трьохбальній системі (-1 – мінімальна, 0 – середня, 1 – максимальна). Відсутність оцінки означає, що для її визначення потрібні додаткові дослідження.

Таблиця 2.1

Якісні оцінки відповідності НМ умовам задач захисту

Умова	БШП	РБФ	SOM	АРТ	СНМ	PNN/ GRNN	Асоціа-тивні
Навчальні дані							
Допустимість шуму	1	0	1	-1	1	0	-1
Допустимість кореляції	1	1	1	1	1	1	-1
Повнота виборки	-1	1	1	-1	-1	1	0
Пропорційність прикладів	1	-1	-1	-1	-1	-1	0
Загальні обмеження процесу навчання							
Короткий термін навчання	-1	0	1	1	0	1	1
Представлення в навчальних прикладах очікуваного	1	1	-1	-1	-1	1	1

Умова	БШП	РБФ	SOM	АРТ	СНМ	PNN/ GRNN	Асоціа-тивні
виходу							
Автоматизація навчання	1	-1	0	1	1	1	0
Можливість донавчання	0	1	1	1	1	1	0
Якість навчання	1	0	0	1	1	1	1
Обчислювальні потужності							
Обсяг пам'яті	1	-1	-1	-1		-1	0
Екстраполяції результатів навчання	1	-1	-1	-1		-1	1
Незмінність результатів	1	1	0	1	1	1	0
Вихідна інформація							
Інтерпретації виходу у вигляді ймовірності	0	0	-1	-1	-1	1	0
Інтерпретації виходу у графічному вигляді	-1	-1	1	-1	-1	-1	-1
Можливість вербалізації	1	0	-1	-1	-1	0	-1
Обмеження технічної реалізації НМ							
Швидкості прийняття рішення	1	1	1	1	0	1	-1
Обсяг програмної реалізації	-1	1	-1	0	-1	-1	0
Сфера застосування							
Системи розпізнавання образів	1	1	1	1	0	1	1
Системи аналізу тексту	-1	-1	1	0	1	0	-1
Системи управління	-1	-1	1	-1	-1	-1	1
Автономність функціонування	-1	-1	-1	1	1	-1	-1

Використання даних табл. 2.1 дозволяє визначити принципову доцільність застосування того чи іншого типу НМ для вирішення задачі. Остаточне рішення про використання конкретного типу НМ із декількох можливих повинно бути прийняте після проведення порівняльних експериментів.

Відповідно матеріалів даної роботи та результатів [2, 7] можна зробити висновок про те, що основними напрямками застосування НМ в галузі комп'ютерного забезпечення технічних та економічних систем є розпізнавання образів, визначення оптимальних управляючих рішень та створення асоціативної пам'яті. До першого напрямку віднесемо задачі класифікації образів, кластеризації образів та апроксимації функцій. Зазначимо, що до групи задач апроксимації функції слід віднести розрахунок параметрів процесів, що відбуваються в технічних системах.

Адже по своїй суті оцінка регресивних або прогнозованих значень параметрів деякого процесу є апроксимацією функції, що описує цей процес. До другого напрямку віднесемо власне задачі оптимального управління та задачі управління з еталонною моделлю.

До третього напрямку входять задачі створення інформаційно-обчислювальних систем з пам'яттю, що адресується за змістом. Теоретична постановка задач всіх трьох напрямків наведена в п.1.1.

Для вирішення поставленого завдання найдоцільніше застосовувати рекурентні нейронні мережі, які в процесі роботи можуть зберігати інформацію про своїх попередніх станах. Далі ми розглянемо принципи роботи таких мереж на прикладі рекуррентной мережі Елмана.

2.3 Використання рекурентних нейронних мереж

Штучна нейронна мережа Елмана, відома так само як Simple Recurrent Neural Network, складається з трьох шарів — вхідного

(розподільного) шару, прихованого і вихідного (обробних) шарів. При цьому прихований шар має зворотний зв'язок сам на себе. На рис. 2.1 представлена схема нейронної мережі Елмана.

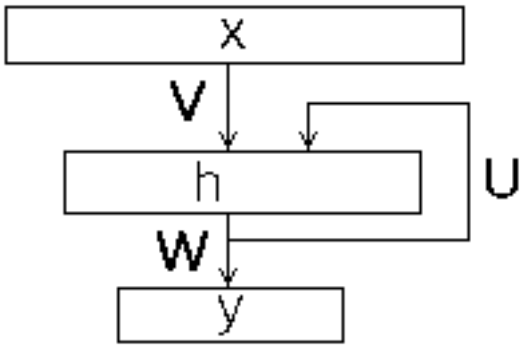

Рисунок 2.1 – Схема нейронної мережі Елмана

На відміну від звичайної мережі прямого поширення, вхідний образ рекурентної мережі — це не один вектор, але послідовність векторів. Вектори вхідного образу в заданому порядку подаються на вхід, при цьому новий стан прихованого шару залежить від його попередніх станів. Мережа Елмана можна описати наступними співвідношеннями:

$$h(t) = f(V^*x(t) + U * h(t-1) + b_h) \quad (2.1)$$

$$y(t) = g(W * h(t) + b_y) \quad (2.2)$$

де

- $x(t)$ – вхідний вектор номер t;
- $h(t)$ – стан прихованого слою для входу $x(t)$ $(h(0) = 0)$;
- $y(t)$ – вихід мережі для входу $x(t)$;
- U – вагова матриця розподільного слою;
- W – вагова (квадратна) матриця зворотніх зв’язків прихованого слою;

- b_h – вектор зсувів прихованого слою;
- V – вагова матриця вихідного слою;
- b_y – вектор зсувів вихідного слою;
- f – функція активації прихованого слою;
- g – функція активації вихідного слою.

При цьому можливі різні схеми роботи мережі. Залежно від того як сформувати вхід і вихід рекурентної мережі, можна різними способами задати схему її роботи. Розглянемо це питання докладніше, для цього розгорнемо схему рекурентної мережі в часі (рис. 2.2).

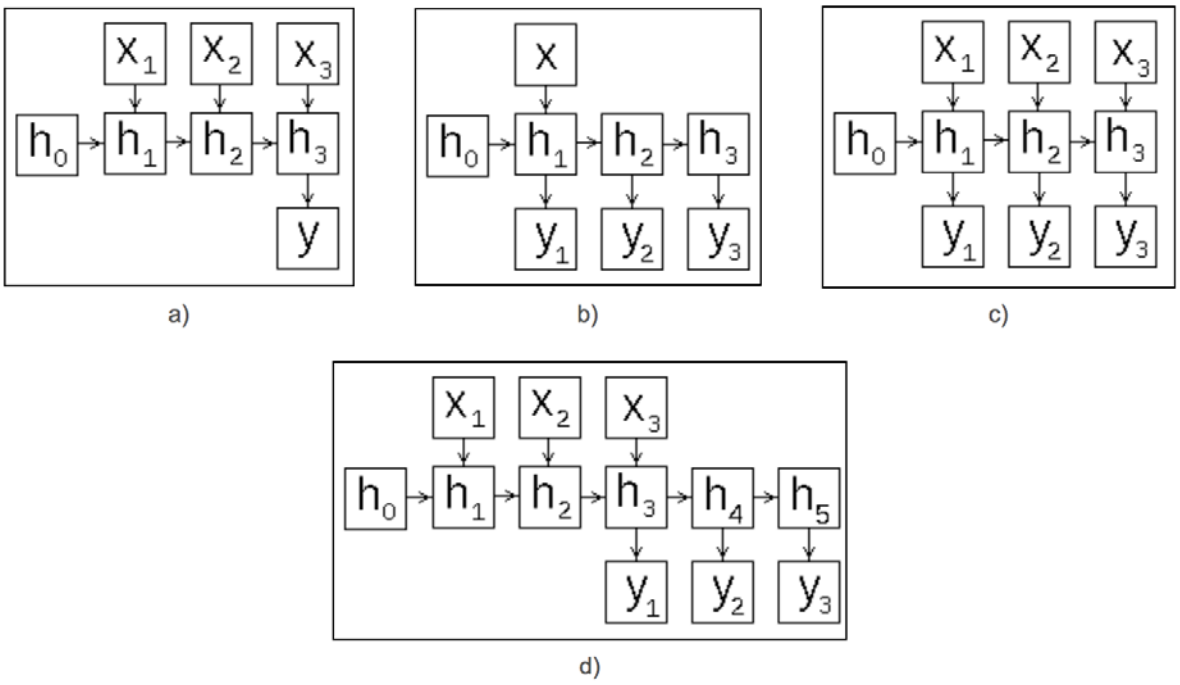

Рисунок 2.2 – Схема роботи рекурентної нейронної мережі

Існують декілька способів організації роботи рекурентної мережі:

- "Багато в один" (many-to-one) (рис. 2.2а) – прихований шар послідовно змінює свій стан, з його кінцевого стану обчислюється

вихід мережі, цю схему можна використовувати для класифікації текстів;

- "Один у багато" (one-to-many) (рис. 2.2b) – прихований шар ініціалізується одним входом, з ланцюжка його наступних станів генеруються виходи мережі, цю схему можна використовувати для анотування зображень;
- "Багато в багато" (many-to-many) (рис. 2.2c) – на кожен вхід мережу видає вихід, який залежить від попередніх входів, цю схему можна використовувати для класифікації відео;
- "Багато в багато" (many-to-many) (рис. 2.2d) – прихований шар послідовно змінює свій стан, його кінцевий стан служить ініціалізацією для видачі ланцюжка результатів, цю схему можна використовувати для створення систем машинного перекладу і чат-ботів.

Розглянемо метод навчання рекуррентной мережі Елмана за схемою many-to-one (рис. 2.2a), для реалізації класифікатора об'єктів, заданих послідовностями векторів. Для навчання мережі Елмана застосовуються ті ж градієнтні методи, що і для звичайних мереж прямого поширення, але з певними модифікаціями для коректного обчислення градієнта функції помилки. Він обчислюється за допомогою модифікованого методу зворотного поширення, який носить назву Backpropagation through time (метод зворотного поширення з розгортанням мережі в часі, BPTT). Ідея методу - розгорнути послідовність, перетворивши рекурентну мережу в "звичайну" (рис. 2.2a). Як і в методі зворотного поширення для мереж прямого поширення, процес обчислення градієнта (зміни ваг) відбувається в три наступних етапи.

- прямий прохід – обчислюємо стану шарів,

- зворотний прохід – обчислюємо помилку шарів,
- обчислення зміни ваг, на основі даних отриманих на першому і другому етапах.

3. ОПИС РОЗРОБЛЕНИХ АЛГОРИТМІВ

3.1 Алгоритм визначення вхідних параметрів нейронної мережі

Наша мова - це послідовність звуків. Звук в свою чергу - це суперпозиція (накладення) звукових коливань (хвиль) різних частот. Хвиля ж, як нам відомо з фізики, характеризуються двома атрибутами - амплітудою і частотою (рис 3.1).

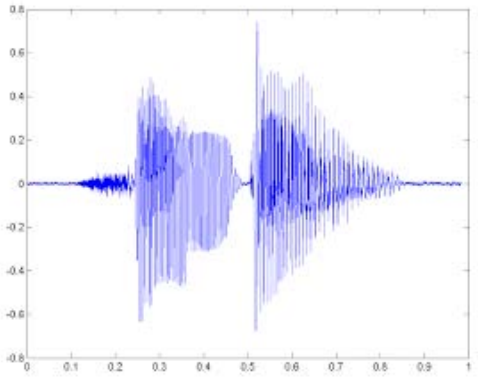

Рисунок 3.1 – Графік амплітудно-частотної характеристики звукової доріжки

Для того, щоб зберегти звуковий сигнал на цифровому носії, його необхідно розбити на безліч проміжків і взяти деякий «усереднене» значення на кожному з них (рис 3.2).

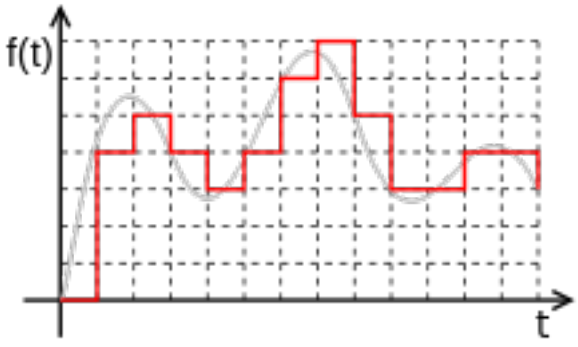

Рисунок 3.2 – Представлення звукової доріжки у вигляді цифрового

сигналу в залежності від часу

Таким чином механічні коливання перетворюються в набір чисел, придатний для обробки на сучасних ЕОМ. Звідси випливає, що завдання розпізнавання мови зводиться до «порівнянні» безлічі чисельних значень (цифрового сигналу) і слів з деякого словника (української мови, наприклад).

Припустимо у нас є деякий файл/потік з аудіоданими. Перш за все нам потрібно зрозуміти, як він влаштований і як його прочитати. Давайте розглянемо найпростіший варіант – WAV файл (рис 3.3, 3.4).

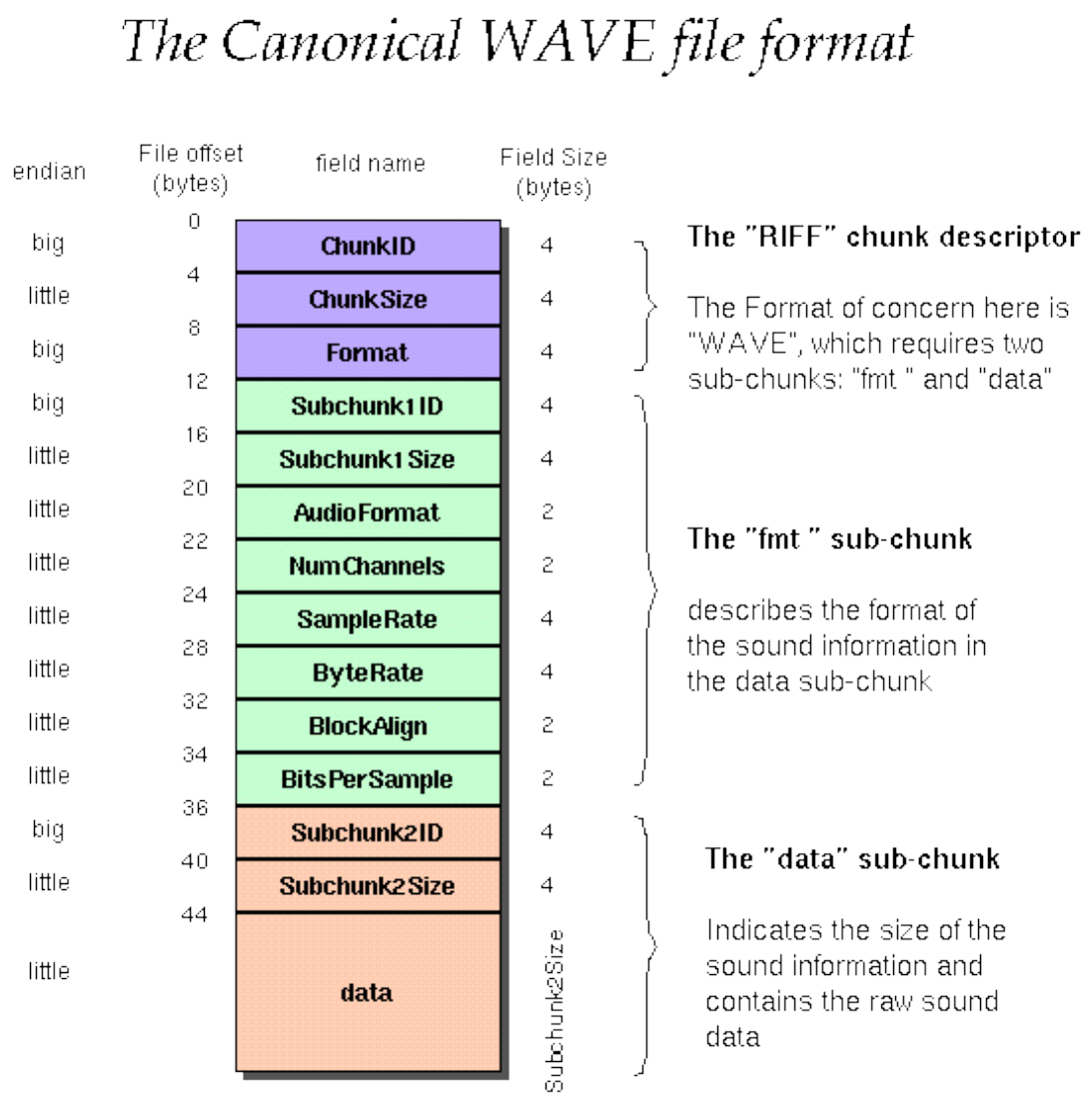

Рисунок 3.3 – Канонічна структура WAV-файлу

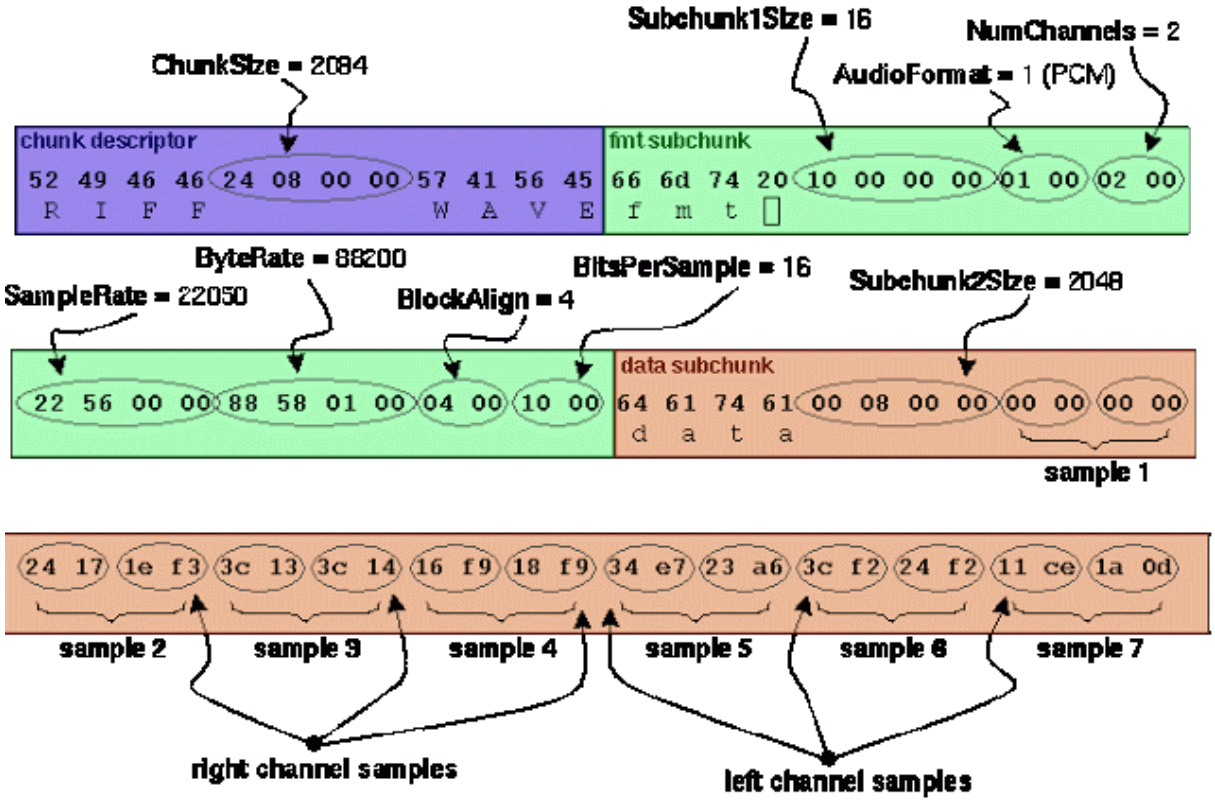

Рисунок 3.4 – Зображення канонічна структура WAV-файлу

Формат передбачає наявність у файлі двох блоків. Перший блок - це заголовок з інформацією про аудіопотоки: бітрейт, частота, кількість каналів, довжина файлу і т.д. Другий блок складається з «сирих» даних - того самого цифрового сигналу, набору значень амплітуд.

Логіка читання даних в цьому випадку досить проста. Прочитуємо заголовок, перевіряємо деякі обмеження (відсутність стиснення, наприклад), зберігаємо дані в спеціально виділений масив.

Теоретично, тепер ми можемо порівняти (поелементно) наявний у нас зразок з яким-небудь іншим, текст якого нам вже відомий. Тобто спробувати «розпізнати» голосові сигнали. Але краще цього не робити. Підхід повинен бути стійкий до зміни тембру голосу (людини, яка вимовляє слово), гучності і швидкості вимови. Поелементний порівнянням

двох аудіосигналів цього, природно, домогтися не можна. Тому ми підемо трохи іншим шляхом.

Насамперед розіб'ємо наші дані на невеличкі тимчасові проміжки - фрейми. Причому фрейми повинні йти не суворо один за одним, а "внахлест". Тобто кінець одного фрейму повинен перетинатися з початком іншого.

Фрейми є більш придатною одиницею аналізу даних, ніж конкретні значення сигналу, так як аналізувати хвилі набагато зручніше на деякому проміжку, ніж в конкретних точках. Розташування ж фреймів "внахлест" дозволяє згладити результати аналізу фреймів, перетворюючи ідею фреймів в якийсь "вікно", що рухається уздовж вихідної функції (значень сигналу).

Дослідним шляхом встановлено, що оптимальна довжина фрейму повинна відповідати проміжку в 10мс, «нахлест» – 50%. З урахуванням того, що середня довжина слова становить 500мс – такий крок дасть нам приблизно 500 / (10 * 0.5) = 100 фреймів на слово.

Перше завдання, яке доводиться вирішувати при розпізнаванні мови, є розбиття цієї самої мови на окремі слова. Для простоти припустимо, що в нашому випадку мова містить в собі деякі паузи (проміжки тиші), які можна вважати "роздільниками" слів.

У такому випадку нам потрібно знайти деяке значення, поріг – значення вище якого є словом, нижче – тишею. Варіантів тут може бути кілька:

- задати константою (спрацює, якщо вихідний сигнал завжди генерується при одних і тих же умовах, одним і тим же способом);
- кластеризувати значення сигналу, явно виділив безліч значень відповідних тиші (спрацює тільки якщо тиша займає значну частину

вихідного сигналу);

- проаналізувати ентропію;

Далі мова піде про останній пункт. Почнемо з того, що ентропія – це міра безладу, "міра невизначеності будь-якого досвіду". У нашому випадку ентропія означає те, як сильно "коливається" наш сигнал в рамках заданого фрейму.

Для того, що б підрахувати ентропію конкретного фрейму слід виконати такі дії:

- припустимо, що наш сигнал пронормований і всі його значення лежать в діапазоні [-1; 1];
- побудуємо гістограму (щільність розподілу) значень сигналу фрейму:
- підрахуємо ентропію за формулою

$$E = \sum_{i=0}^{N-1} P[i] * log_2(P[i]) \quad (3.1)$$

Таким чином, ми отримали значення ентропії. Але це всього лише ще одна характеристика фрейму, і для того, що б відокремити звук від тиші, нам як і раніше потрібно її з чимось порівнювати. У деяких статтях рекомендують брати поріг ентропії рівним середньому між її максимальним і мінімальним значеннями (серед всіх фреймів). Однак, в моєму випадку такий підхід не дав скільки-небудь хороших результатів.

На щастя, ентропія (на відміну від того ж середнього квадрата значень) - величина відносно самостійна, що дозволило мені підібрати значення її порога у вигляді константи (0.1).

Проте проблеми на цьому не закінчуються. Ентропія може просідати

по середині слова (на голосних), а може раптово схоплюватися через невеликого шуму. Для того, що б боротися з першою проблемою, доводиться вводити поняття "мінімально відстані між словами" і "склеювати" поблизу лежачі набори фреймів, розділені через просідання. Друга проблема вирішується використанням "мінімальної довжини слова" і відсіканням всіх кандидатів, які не пройшли відбір (і не використаних в першому пункті).

Якщо ж мова в принципі не є "членороздільної", можна спробувати розбити вихідний набір фреймів на певним чином підготовлені підпослідовності, кожна з яких буде піддана процедурі розпізнавання. Але це вже зовсім інша історія.

І так, ми у нас є набір фреймів, відповідних певному слову. Ми можемо піти по шляху найменшого опору і в якості чисельної характеристики фрейму використовувати середній квадрат всіх його значень (Root Mean Square). Однак, така метрика несе в собі вкрай мало придатною для подальшого аналізу інформації.

Ось тут в гру і вступають Мел-частотні кепстральні коефіцієнти (Mel-frequency cepstral coefficients). MFCC – це своєрідне уявлення енергії спектра сигналу. Плюси його використання полягають у наступному:

- Використовується спектр сигналу (тобто розкладання по базису ортогональних синусоїдальних функцій), що дозволяє враховувати хвильову "природу" сигналу при подальшому аналізі;
- Спектр проектується на спеціальну мел-шкалу, дозволяючи виділити найбільш значущі для сприйняття людиною частоти;
- Кількість обчислюваних коефіцієнтів може бути обмежена будь-яким значенням (наприклад, 12), що дозволяє "стиснути" фрейм і, як наслідок, кількість оброблюваної інформації.

Розглянемо процес обчислення MFCC коефіцієнтів для деякого фрейму. Уявімо наш фрейм у вигляді вектора $x[k], 0 \leq k < N$, де N - розмір кадру.

Насамперед розраховуємо спектр сигналу за допомогою дискретного перетворення Фур'є (бажано його "швидкої" FFT реалізацією).

$$X[k] = \sum_{n=0}^{N-1} x[n] * e^{\frac{-2*\pi*i*k*n}{N}}, 0 \leq k < N \quad (3.2)$$

Також до отриманими значеннями рекомендується застосувати віконну функцію Хеммінга, що б "згладити" значення на кордонах фреймів.

$$H[k] = 0.54 - 0.46 * \cos(\frac{2*\pi*k}{N-1}) \quad (3.3)$$

Тобто результатом буде вектор наступного вигляду:

$$X[k] = X[k] * H[k], 0 \leq k < N \quad (3.4)$$

Важливо розуміти, що після цього перетворення по осі X ми маємо частоту (Hz) сигналу, а по осі Y - магнітуду як спосіб піти від комплексних значень (рис 3.5).

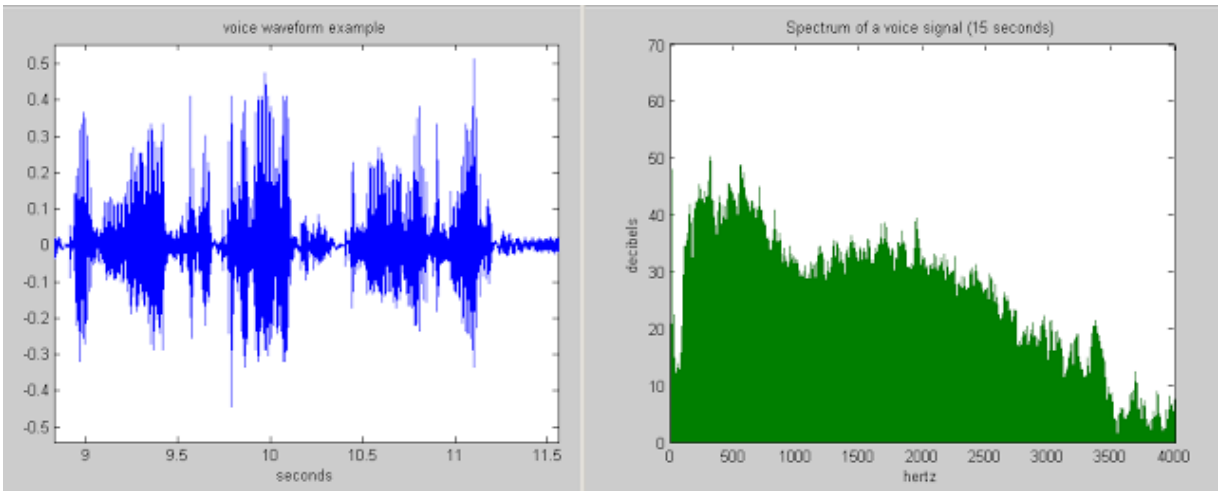

Рисунок 3.5 – Представлення спектру голосового сигналу у вигляді графіків

Наступним кроком буде підрахунок мел-фільтрів. Почнемо з того, що таке мел. Мел – це "психофізична одиниця висоти звуку", заснована на суб'єктивному сприйнятті середньостатистичними людьми. Залежить в першу чергу від частоти звуку (а так само від гучності і тембру). Іншими словами, ця величина, що показує, на скільки звук певної частоти "значущий" для нас.

Перетворити частоту в мел можна за такою формулою:

$$M = 1127 * \log\left(1 + \frac{F}{700}\right) \quad (3.5)$$

Зворотне перетворення виглядає так:

$$F = 700 * \left(e^{\frac{M}{1127}} - 1\right) \quad (3.6)$$

Намалюємо графік залежності мел від частоти (рис 3.6).

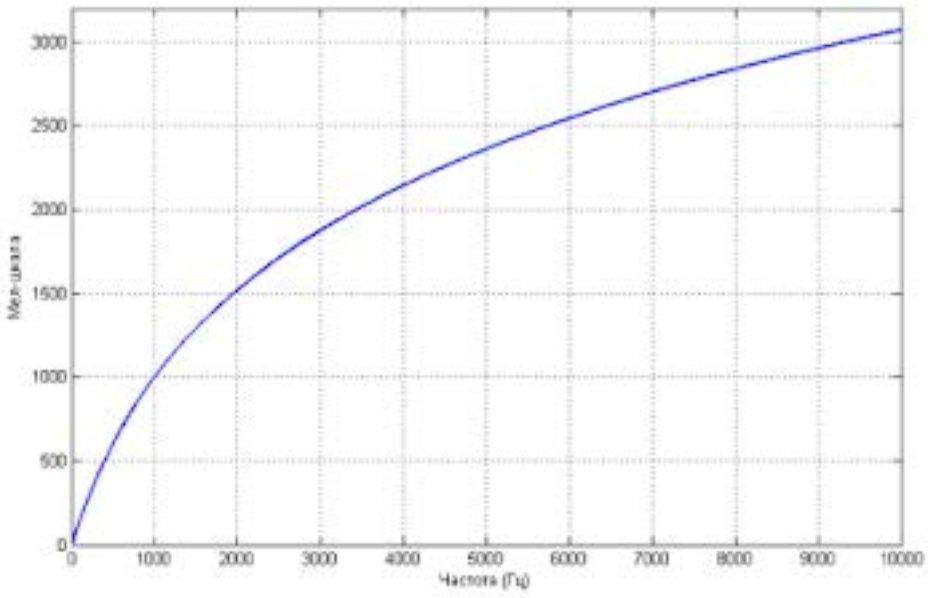

Рис 3.6. Графік залежності мел від частоти

Але повернемося до нашого завдання. Припустимо у нас є фрейм розміром 256 елементів. Ми знаємо (з даних про аудіоформат), що частота звуку в даній фреймі 16000hz. Припустимо, що людська мова лежить в

діапазоні від [300; 8000] Гц. Кількість шуканих мел-коефіцієнтів нехай буде M = 10 (рекомендований значення).

Для того, щоб розкласти отриманий вище спектр за мел-шкалою, нам буде потрібно створити "гребінку" фільтрів. По суті, кожен мел-фільтр це трикутна віконна функція, яка дозволяє підсумувати кількість енергії на певному діапазоні частот і тим самим отримати мел-коефіцієнт. Знаючи кількість мел-коефіцієнтів і аналізований діапазон частот ми можемо побудувати набір наступних фільтрів (рис 3.7).

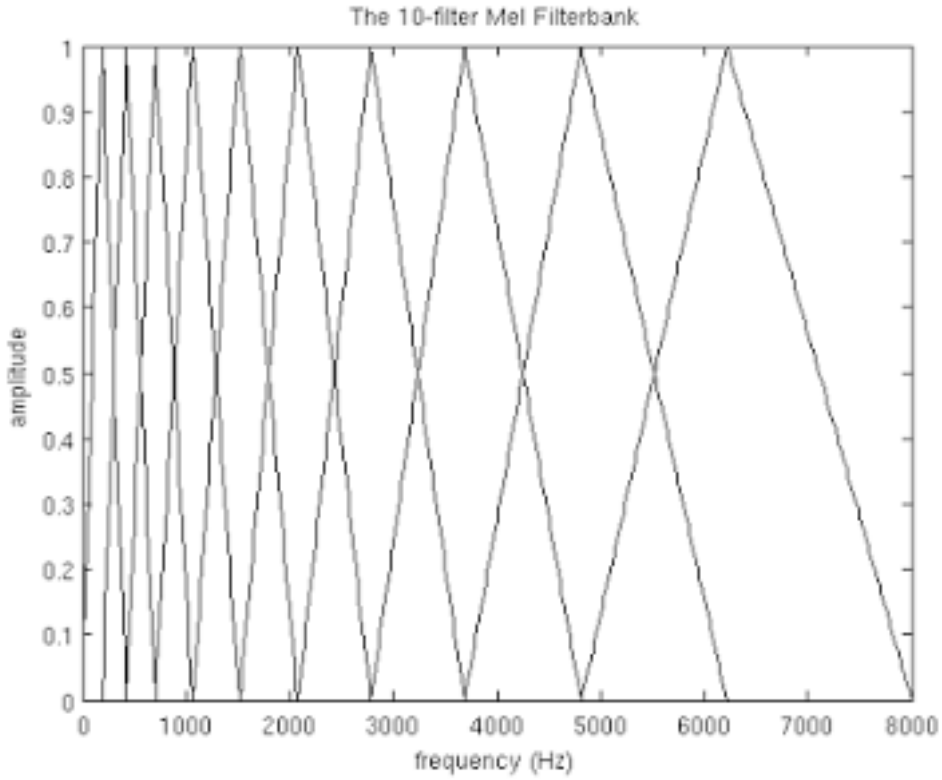

Рис 3.7 Графік залежності кількості мел-фільтрів від частоти

Зверніть увагу, що чим більше порядковий номер мел-коефіцієнта, тим ширше основа фільтра. Це пов'язано з тим, що розбиття цікавого нам діапазону частот на оброблювані фільтрами діапазони відбувається на шкалі мел-ов.

Для нашого випадку діапазон цікавлять нас частот дорівнює [300,

8000]. Відповідно до формули 3.5 на мел-шкалі цей діапазон перетворюється в [401.25; 2834.99].

Далі, для того, що б побудувати 10 трикутних фільтрів нам буде потрібно 12 опорних точок: m[i] = [401.25, 622.50, 843.75, 1065.00, 1286.25, 1507.50, 1728.74, 1949.99, 2171.24, 2392.49, 2613.74, 2834.99].

Зверніть увагу, що на мел-шкалі точки розташовані рівномірно. Переведемо шкалу назад в Герци за допомогою формули 3.6: h [i] = [300, 517.33, 781.90, 1103.97, 1496.04, 1973.32, 2554.33, 3261.62, 4122.63, 5170.76, 6446.70, 8000].

Як ми можемо побачити тепер шкала стала поступово розтягуватися, вирівнюючи тим самим динаміку зростання "значущості" на низьких і високих частотах.

Тепер нам потрібно накласти отриману шкалу на спектр нашого фрейму. Як ми пам'ятаємо, по осі X у нас знаходиться частота. Довжина спектра 256 - елементів, при цьому в нього вміщається 16000hz. Вирішивши нехитру пропорцію можна отримати наступну формулу:

$$f(i) = floor((frameSize + 1) * h(i) / sampleRate) \quad (3.7)$$

що в нашому випадку еквівалентно

$$f(i) = 4, 8, 12, 17, 23, 31, 40, 52, 66, 82, 103, 128$$

Знаючи опорні точки на осі X нашого спектра, легко побудувати необхідні нам фільтри за такою формулою:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases} \quad (3.8)$$

Застосування фільтру полягає в попарному перемножуванні його

значень зі значеннями спектра. Результатом цієї операції є мел-коефіцієнт. Оскільки фільтрів у нас M, коефіцієнтів буде стільки ж.

$$S[m] = \log\left(\sum_{k=0}^{N-1} |X[k]|^2 * H_m[k]\right), 0 \le m < M \qquad (3.9)$$

Однак, потрібно застосувати мел-фільтри не до значень спектра, а до його енергії. Після чого логарифмувати отримані результати. Вважається, що таким чином знижується чутливість коефіцієнтів до шумів.

Дискретне косінусне перетворення (DCT) використовується для того, що б отримати ті самі "кепстральні" коефіцієнти. Сенс його в тому, що б "стиснути" отримані результати, підвищивши значимість перших коефіцієнтів і зменшивши значимість останніх.

$$C[l] = \sum_{m=0}^{M-1} S[m] * \cos\left(\pi * l * \frac{m+\frac{1}{2}}{M}\right), 0 \le l < M \qquad (3.10)$$

Тепер для кожного фрейму ми маємо набір з M mfcc-коефіцієнтів, які можуть бути використані для подальшого аналізу.

3.2 Тренування та використання рекурентної нейронної мережі

Навчальну та тестову вибірки було сформовано на основі мовленнєвих корпусів. Що ж таке мовленнєвий корпус? Нині в науковому словнику лінгвістів з'являються дуже близькі поняття: «електронні бібліотеки», «масив текстів», «колекція текстів», «електронний архів», «повнотекстова база даних». Серед них можна виділити лінгвістичні корпуси, або мовні корпуси. Корпус текстів — це вид корпусу даних, одиницями якого є тексти або їх достатньо значні фрагменти, що включають, наприклад, якісь повні фрагменти макроструктури текстів даної проблемної області.

Цінність корпусу вбачається в наступному:

- одного разу зроблений корпус може багато разів використовуватися;
- корпус показує мовні дані в їх реальному оточенні, що дозволяє досліджувати лексичну і граматичну структуру мови, а також безперервні процеси мовних змін, що відбуваються в мові впродовж певного відрізка часу;
- корпус характеризується показовістю, або збалансованим складом текстів, що дозволяє використовувати його для тестування пошукових машин, машинних морфологій, систем перекладу, а також використовувати його в різних лінгвістичних дослідженнях;
- корпус має важливе значення для викладання мови, оскільки за допомогою корпусу можна швидко і ефективно перевірити особливості вживання незнайомого слова або граматичної форми.

Перший комп'ютеризований лінгвістичний корпус був розроблений 1971 року Монреальським французьким проектом, що містить один мільйон слів.

Існування корпусів текстів дає можливість значно розширити й автоматизувати аналіз мовного матеріалу, який є найважливішою базою будь-якого лінгвістичного дослідження. Чим більше матеріалів аналізується, тим вища значущість висновків і рівень їх достовірності.

Сучасні комп'ютерні програми дозволяють знаходити потрібні приклади з корпусів текстів, які зберігаються в електронному вигляді на комп'ютері. Це економить значну кількість часу в порівнянні з традиційною технологією збору прикладів вручну.

Відзначимо, що саме анотація, або розмітка, — головна характеристика корпусу, яка і відрізняє його від електронних колекцій,

бібліотек, енциклопедій, широко представлених в сучасному Інтернеті. Розмітка тексту — це приписування тексту певної інформації для зручнішого аналізу.

Існують різні типи розмітки:

- метатекстова розмітка (автор, назва, дата створення, обсяг, тематика тексту і т. д.), яка характеризує текст в цілому;
- структурна розмітка є інформацією про структуру тексту, яка дозволяє відокремити одне слово від іншого, виділити межі словосполучення, речення, тексту;
- лінгвістична розмітка полягає в приписуванні одиницям тексту певної лінгвістичної інформації (заперечне речення або питальне, спонукальне або примикання і т. д.).

Як відомо, чим багатша і різноманітніша розмітка, тим вищою є наукова і навчальна цінність корпусу.

В Україні корпус текстів української мови розроблений співробітниками лабораторії комп'ютерної лінгвістики Інституту філології Київського національного університету імені Тараса Шевченка під керівництвом Н. П. Дарчук.

У корпусі зберігаються тексти, опрацьовані автоматичним лінгвістичним аналізатором. Це означає, що кожній одиниці тексту (морфемі, слову, словосполученню, реченню) приписана певна супровідна інформація: частиномовна належність, граматична форма, синтаксична функція, контекст тощо. Корпус надає інформацію двох типів:

Конкорданси, або ж контексти вживання шуканих одиниць (із вказівкою на джерела). За допомогою конкордансів можна вивчати особливості використання слів у текстах різних стилів, індивідуально-авторські вживання тих чи інших лексем, розвиток нових значень тощо. Конкорданси використовуються також для психолінгвістичних та

соціолінгвістичних досліджень (вивчення асоціативних зв'язків між словами), у літературознавчому аналізі (для розкриття авторського бачення певних концептів чи образів, особливостей мовної картини світу певного автора) тощо.

Кількісні характеристики вживання у текстах мовних одиниць. Частотна інформація розкриває закономірності лексичної та статистичної будови текстів, функціонування мови в мовленні, стилістичні особливості, формальні риси одиниць і граматичних категорій.

Таким чином для навчання нейронної мережі вибирається набір wav-файлів із вимовою ключових. Кожен файл має структурну розмітку.

Після отримання тренувальних прикладів, програма по черзі бере кожен приклад та, згідно алгоритму, що зображений на кресленні Д2 (див. Додаток 1), формує вхідні дані у числовому вигляді. Далі відбувається цикл тренування доки допустима похибка не буде в межах допустимої норми.

4. АНАЛІЗ РОЗРОБЛЕНОЇ СИСТЕМИ

4.1 Особливості реалізації системи

Дана комп’ютерна система була реалізована у вигляді CLI програми (рис 4.1).

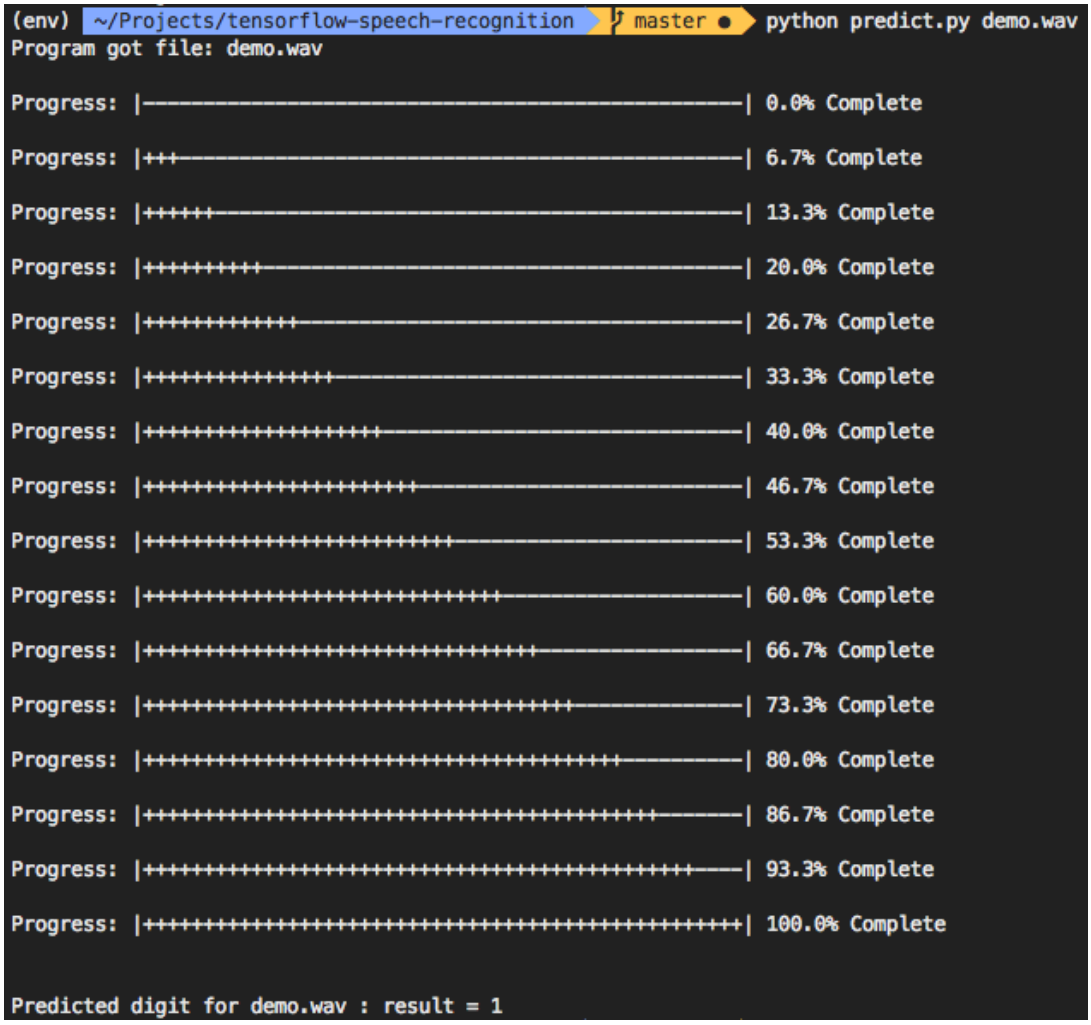

Рисунок 4.1 – Вивід даних про розпізнавання ключового слова у CLI додатку

Тренування відбувається наступним чином. Після запуску відповідної команди програма автоматично перевіряє у поточній директорії наявність директорії data/ що має містити тестові файли з мовленнєвими корпусами. Якщо вони є, то почнеться навчання системи. Якщо ні – програма спробує завантажити іх. Після старту навчання

програма починає виводити дані про поточний крок навчання з інформацію про похибку і затрачений час на певний крок, тощо (рис 4.2).

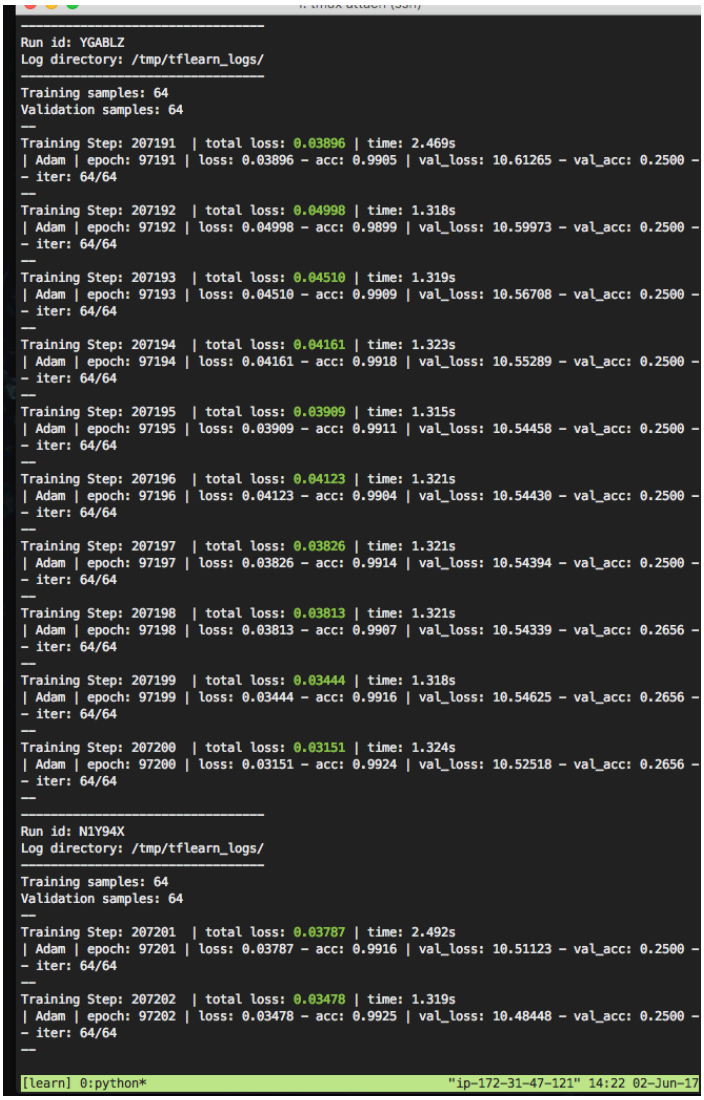

Рисунок 4.2 – Вивід програми про поточний стан навчання нейромережевої системи

Усі дані про навчання автоматично записуються в теку /tmp/rflearn_logs. Використовуючи ці дані, за допомогою CLI додатку Tensorboard можна проаналізувати, як працює система (рис. 4.3). Додаток обладнаний зручним та інтуітивно зрозумілим інтерфейсом.

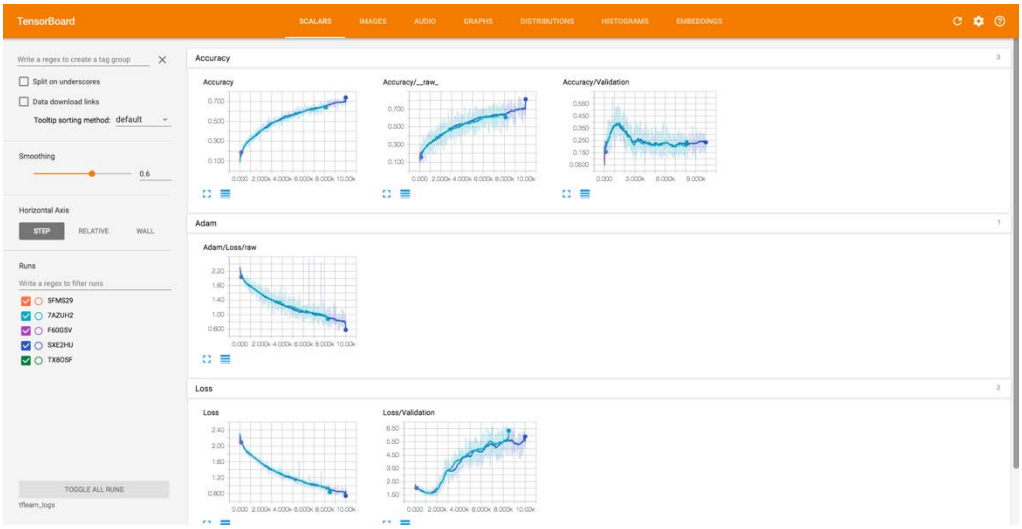

Рисунок 4.3 – Інформація про роботу нейромережевої системи.

Також можна детально роздивитися структуру нейромережевої системи, що була створена (рис 4.4). Структура зображена у вигляді інтерактивного графу із згрупованими вершинами.

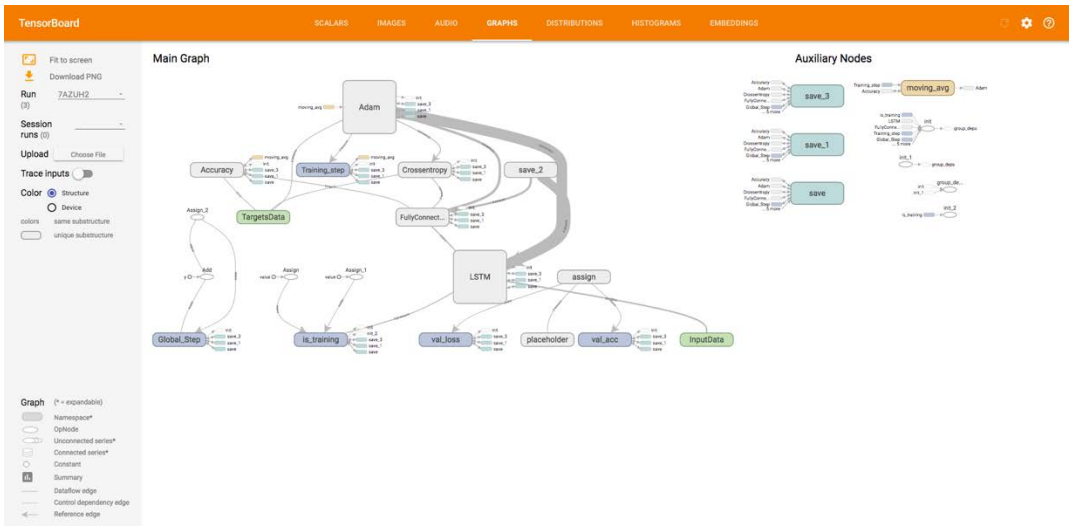

Рисунок 4.4 – Структура нейромережевої системи

Комп’ютерну систему розроблено на мові Python з використанням технології Tensorflow.

4.2 Тестування системи

Сформовано наступний алгоритм розробки КС, призначеної для розпізнавання голосових сигналів:

- Підготовка тестової та навчальної вибірки.
- Визначення вхідних параметрів нейронної мережі.
- Вибір виду та параметрів функцій активації для всіх типів нейронів.
- Проведення навчання.
- Проведення тестування.
- Якщо результати тестування не задовільні – зміна параметрів НМ.

В першому наближенні встановимо, що тестової вибіркою є структурно розмічені набори WAV файлів, що містять звуки вимови цифр від 0 до 10. Розмір вибірки 25 гігабайт. Також встановимо, що з кожного файлу на вхід нейронної мережі має подаватися 80 наборів по 20 мел-кепстральних коефіцієнтів. Встановимо коефіцієнт навчання 0.0001 та 100000 навчальних кроків для того, щоб зменшити похибку при розпізнаванні надалі. Структура НМ у загальному випадку приведена на кресленні Д4 (див. Додаток 1.)

Проведено її навчання. Термін навчання на 1100 прикладах склав близько 24 години, кількість навчальних ітерацій 10000, а середня відносна похибка навчання $8{,}2\times10^{-6}$. Для оптимізації використано метод градієнтного спуску.

Під час навчання було побудовано декілька графіків амплітудно-частотної характеристики та наборів мел-кепстральних коефіцієнтів (Рис 4.5) для наочності результатів підготовкі вхідних даних.

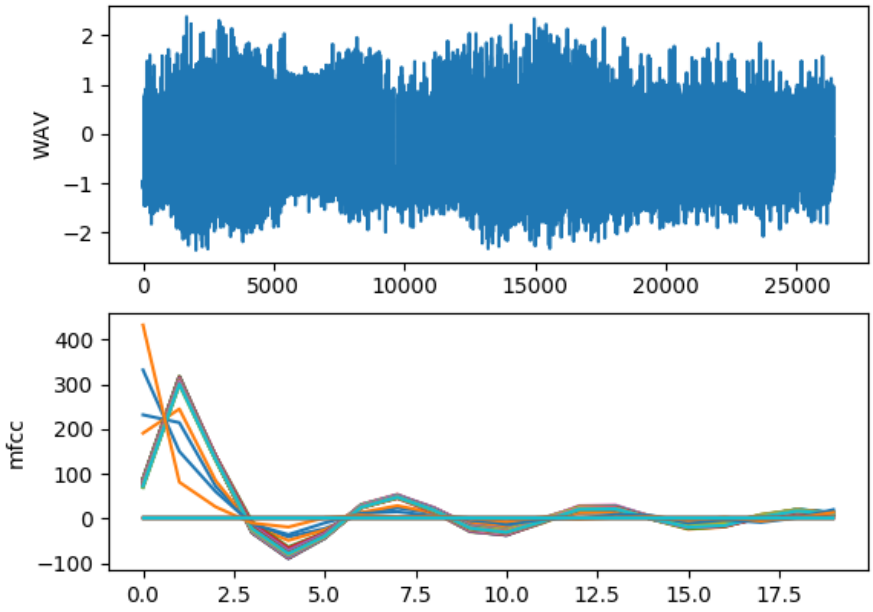

Рисунок 4.5 – Вивід програмою графіків амплітудно-частотних характеристик та наборів мел-кепстральних коефіцієнтів

Після навчання НМ були пред’явлені 40 тестових прикладів, що не ввійшли до складу навчальної вибірки. 20 тестових прикладів відповідали ключовим словам (в даному випадку, цифрам), інші 20 – іншим випадковим словам. У 3 із 40 випадках НМ вивела неправильні дані, що може бути зумовлено недостатньою кількістю виділеного часу на навчання. При подальшому навчанні цю похибку можна виправити.

ВИСНОВКИ

В дипломному проекті було розглянуто задачу розробки інформаційної системи розпізнавання ключових слів в голосових файлах. В процесі вирішення отримано наступні результати:

1. Показано, що перспективним шляхом забезпечення достатньої якості розпізнавання ключових слів являється вдосконалення математичного забезпечення інформаційної системи. Також показано, що для розпізнавання доцільно використовувати нейромережевий аналіз мел-кепстральних коефіцієнтів оцифрованого голосового сигналу.

2. Розроблена архітектура інформаційної системи розпізнавання, що адаптована до застосування запропонованого математичного забезпечення на основі використання мел-кепстральних коефіцієнтів та нейронних мереж.

3. Розроблене математичне та програмне забезпечення інформаційної системи, що базується на використанні апарату мел-кепстральних коефіцієнтів.

4. Проведені експериментальні дослідження підтвердили перспективність застосування розробленого математичного забезпечення для розпізнавання ключових слів.

5. Розроблену систему розпізнавання рекомендується впроваджувати в інформаційних системах загального призначення в яких є необхідність з помірною похибкою розпізнавати ключові слова.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Шалимов И.А., Бессонов М.А. Обзор методов автоматической идентификации языка аудиосообщения // Труды НИИР. 2011. № 3. С. 43-47.
2. Leonard R.G., Doddington G.R. Automatic language identification. Technical report RADC-TR-74-200, Air Force Rome Air Development Center, 1974.
3. Leonard R.G., Doddington G.R. Automatic language identification. Technical report RADC-TR-75-264, Air Force Rome Air Development Center, 1975.
4. Leonard R.G., Doddington G.R. Automatic language discrimination. Technical report RADC-TR-78-5, Air Force Rome Air Development Center, 1978.
5. Leonard R.G. Language recognition test and evaluation. Technical report RADC-TR-80-83, Air Force Rome Air Development Center, 1980.
6. House A.S., Neuberg E.P. Toward automatic identification of the language of an utterance. Preliminary methodological considerations. Jornal of the Acoustical Society of America, vol 62(3): 708-713, 1977.
7. Li K.P., Edwards T.J. Statistical models for automatic language identification. In Proceedings IEEE International conference on Acoustic, Speech and Signal Processing 80, Denver, CO, 1980.
8. Cimarusti D., Ives R.B. Development of an automatic identification system of spoken languages: Phase 1. In Proceedings IEEE International conference on acoustic, speech and signal processing, Paris, 1982.
9. Айвенс К. Компьютерные сети / Айвенс К. ; пер. с. англ. – СПб. : Питер, 2006. – 304 с.

10. Барский А. Б. Нейронные сети: распознавание, управление, принятие решений / А. Б. Барский. – М. : Финансы и статистика, 2004. – 176 с.
11. Вакуленко А. Биометрические методы идентификации личности: обоснованный выбор и внедрение / А. Вакуленко, А. Юхин. – М.: Наука, 2007. – 224 с..
12. Вилков А.С. Информационная безопасность персональных ЭВМ и мониторинг компьютерных сетей / А.С. Вилков. – М. : МИНИТ ФСБ России, 2005. – 210 с.
13. Галушкин А. И. Теория нейронных сетей / А. И. Галушкин. – М. : ИПРЖР, 2000. – 416 с.
14. Горбань А. Н. Обучение нейронных сетей / А. Н. Горбань. – М. : ParaGraph, 1990. – 160 с/
15. Задоров В.Б. Системний аналіз об’єктів і процесів: технологічні основи: Навчальний посібник. – К.: КНУБА - 2003. – 276 с.
16. Зиятдинов А.И. Принципы построения систем биометрической аутентификации / А.И. Зиятдинов. – М.: МФТИ, 2005. – 188 с..
17. Матвеев Ю.Н. Технологии биометрической идентификации личности по голосу и другим модальностям, Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение, 2012, № 3, Специальный выпуск Биометрические технологии, С. 46–61.
18. Зиновьев А.Ю. Визуализация многомерных данных / А. Ю. Зиновьев. М. : СК Пресс, 2005. – 180 с.
19. Терейковський І. Нейронні мережі в засобах захисту комп’ютерної інформації / І. Терейковський. К. : ПоліграфКонсалтинг. 2007. – 209 с.
20. Jeffrey L. Elman Finding Structure in Time // COGNITIVE SCIENCE 14, 179-211 (1990)