Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра Програмної інженерії

АТЕСТАЦІЙНА РОБОТА МАГІСТРА

пояснювальна записка

Дослідження методів штучного інтелекту для реалізації системи розумного дому

Магістрант гр. ІПЗм-16-1 Самоцький В.О.

Керівник роботи: доц. Лещинський В.О.

Рецензент: доц. Вечур О.В.

Рецензент: д.т.н, проф. Шостак І.В.

Допускається до захисту
Зав. кафедри, проф. _____ Дудар З.В.

2018 р.

Харківський національний університет радіоелектроніки

Факультет *комп'ютерних наук*

Кафедра *програмної інженерії*

Спеціальність *121- Інженерія програмного забезпечення*

Освітня програма *Програмне забезпечення систем*

ЗАТВЕРДЖУЮ:

«____»_____2018 р _____

Зав. кафедри проф. З.В.Дудар

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ МАГІСТРА

магістрантові

Самоцькому Владиславу Олександровичу

1. Тема роботи *«Дослідження методів штучного інтелекту для реалізації системи розумного дому»* затверджена наказом університету №450 Ст від «16» квітня 2018 р

2. Термін здачі студентом закінченої роботи «12» червня 2018 р.

3. Вихідні дані до проекту (роботи): *Дослідити методи штучного інтелекту, які вже використовуються в системах розумного дому. Визначити основні недоліки методів та їх особливості використання. Знайти методи покращення даних методів та існуючих систем розумного дому. Використовувати будь-які методи та засоби програмної інженерії.*

4. Зміст пояснювальної записки *мета роботи, аналіз вимог і підходів до автоматичного розпізнавання мови, методи побудови інформаційних ознак і акустичних моделей на основі глибоких нейронних мереж, проектування та розробка програмного забезпечення*

5. Перелік графічного матеріалу *базові підходи до автоматичного розпізнавання мови, приклад глибокої нейронної мережі, обмежена машина Больцмана, дескримінативне пренавчання нейронних мереж, схема адаптації з використанням і-векторів, глибока нейронна мережа як, розподіл велечин ваг в*

типічній DNN, відсоток неактивних нейронів на кожному шарі dnn, середнє і забезпечити максимальне значення, на кожному шарі для DNN, схема навчання GMM-HMM акустичної моделі з використанням ознак, схема витягвання bottleneck-ознак другого рівня, основні етапи навчання глибокої нейронної мережі з вузьким горлом, схема алгоритму побудови ознак за допомогою адаптованої з використанням і-векторів глибокої нейронної мережі з вузьким горлом, діаграма компонентів, діаграма кооперації, отримання мовного повідомлення користувача, передача мовної команди, передача файлу до сервісу розпізнання.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Об'єктний аналіз поставленої задачі	25-02-2018	виконано
2	Аналіз предметної галузі	12-03-2018	виконано
3	Аналіз існуючих рішень	15-04-2018	виконано
4	Дослідження існуючих методів	28-05-2018	виконано
5	Пошук та вирішення недоліків алгоритмів та систем	15-05-2018	виконано
6	Підготовка пояснювальної записки.	20-06-2018	виконано
7	Підготовка презентації та доповіді	01-06-2018	виконано
8	Попередній захист	04-06-2018	виконано
9	Нормоконтроль, рецензування	07-06-2018	виконано
10	Занесення диплома в електронний архів	07-06-2018	виконано
11	Допуск до захисту у зав. кафедри	08-06-2018	виконано

Дата видачі завдання «22» січня 2018 р.

Керівник доц. _____ Лещинський В.О.

Завдання прийняв до виконання _____ Самоцький В.О.

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до атестаційної роботи: 77 с., 17 рис., 6 табл., 3 додатки, 26 джерел.

МЕТОДИ, МОВЛЕННЯ, РОЗПІЗНАВАННЯ, СИСТЕМА, ШТУЧНИЙ ІНТЕЛЕКТ, ARDUINO, JAVASCRIPT.

Об’єктом дослідження є методи штучного інтелекту, які використовуються в системах розумного дому.

Метою роботи є дослідити, проаналізувати та модернізувати або винайти механізм покращення роботи систем розумного дому, які використовують методи штучного інтелекту в своїй роботі.

Методи розробки базуються на технологіях, які використовуюсь системи розумного дому для роботи з методами штучного інтелекту, а саме: C++, JavaScript, NodeJS та інші.

У результаті роботи здійснене дослідження методів штучного інтелекту та розроблена схема модернізацію методів, які використовуються в даний час в системах розумного дому.

METHODS, ARTIFICIAL INTELLIGENCE, RECONITION, VOICE, SYSTEM, JAVASCRIPT, ARDUINO.

The object of the study is the methods of artificial intelligence that are used or used earlier and have been upgraded or excluded, in systems of intelligent home.

The purpose of the work is to investigate, analyze, and modernize or invent a mechanism for improving the functioning of systems of a smart home that use artificial intelligence methods in their work.

As a result of the work, the research of methods of artificial intelligence was carried out and the scheme of modernization of methods, which are currently used in systems of a smart home, was developed.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ASR – Automatic Speech Recognition

AM – Акустична модель

ЯМ – Язикова модель

WER – Word Error Rate

MFCC – Mel-Frequency Cepstral Coefficients

FBANK – Mel-frequency filterbank log energies

PLP – Perceptual Linear Prediction

LDA – Linear Discriminant Analysis

CMN – Cepstral Mean Normalization

CMVN – Cepstral Mean and Variance Normalization

VTLN – Vocal Tract Length Normalization

HMM –Hidden Markov Model

GMM – Gaussian Mixture Model

ML – Maximum Likelihood

EM – Expectation-Maximization

MLLT – Maximum Likelihood Linear Transformation

MLLR – Maximum Likelihood Linear Regression

fMLLR – feature-domain Maximum Likelihood Linear Regression

MAP-LR – Maximum a Posteriori Linear Regression

DNN – Deep Neural Network

ANN – Artificial Neural Network

MSE – Mean Square Error

CE – Cross-Entropy

NLL – Negative Log-Likelihood

BP – Error Backpropagation

SGD – Stochastic Gradient Descent

NAG – Nesterov Accelerated Gradient

RBM – Restricted Boltzmann Machine

DBN – Deep Belief Network

DPT – discriminative pretraining

LBP – layer-wise error backpropagation

CD-DNN-HMM – Context-Dependent Deep Neural Network – Hidden Markov Model

FER – Frame Error Rate

ST – Sequence-discriminative Training

MMI – Maximum Mutual Information

BMMI – Boosted Maximum Mutual Information

MPE –Minimum Phone Error

MBR – Minimum Bayes Risk

sMBR – state Minimum Bayes Risk

F-Smoothing – Frame Smoothing

fDLR – feature Discriminant Linear Regression

JFA – Joint Factor Analysis

VTS – Vector Taylor Series

UBM – Universal Background Model

PPL – Perplexity

RNNLM – Recurrent Neural Network Language Model

FLM – Factored Language Model

MaxEnt – Maximum Entropy

PCA – Principal Component Analysis

HLDA – Heteroscedactic Linear Discriminant Analysis

SVD – Singular Values Decomposition

GPGPU – General-purpose computing for graphics processing units

VAD – Voice Activity Detector

RTF – Real-Time Factor

ВСТУП

Актуальність теми. У наш час персональні комп’ютери відіграють важливу роль майже у всіх сферах діяльності людини. Обчислювальна техніка значно полегшує працю людини. Сьогодні у зв'язку з розвитком інформаційних технологій, появою їх у всіх сферах життєдіяльності людини, збільшенням обсягів і потоків інформації, що зберігається в основному на електронних носіях, проводиться автоматизація майже усіх сфер діяльності. Навіть сфери діяльності, які, здавалось би, ніяк не можуть обійтись без участі людини, автоматизуються.

Питання людино-машинного взаємодії є одними з найважливіших при створенні нових комп'ютерів. Найбільш ефективними засобами взаємодії людини з машиною були б ті, які є природними для нього: через візуальні образи і мову. Створення мовних інтерфейсів могло б знайти застосування в системах самого різного призначення: голосове управління для людей з обмеженими можливостями, надійне управління бойовими машинами, «розуміючими» тільки голос командира, автовідповідачі, обробні в автоматичному режимі сотні тисяч дзвінків на добу (наприклад, в системі продажу авіаквитків) і т.д. При цьому, мовний інтерфейс повинен включати в себе два компоненти: систему автоматичного розпізнавання мови для прийому мовного сигналу і перетворення його в текст або команду, і систему синтезу мовлення, що виконує протилежну функцію – конвертацію повідомлення від машини в мову.

Однак, не дивлячись на стрімко зростаючі обчислювальні потужності, створення систем розпізнавання мови залишається надзвичайно складною проблемою. Це обумовлюється як її міждисциплінарним характером (необхідно володіти знаннями в філології, лінгвістиці, цифровій обробці сигналів, акустиці, зі статистикою, розпізнаванні образів і т.д.), так і високою обчислювальною складністю розроблених алгоритмів. Останнє накладає суттєві обмеження на системи автоматичного розпізнавання мови – на обсяг оброблюваного словника, швидкість отримання відповіді і його точність. Не можна також не згадати про те,

що можливості подальшого збільшення швидкодії ЕОМ за рахунок вдосконалення інтегральної технології рано чи пізно будуть вичерпані, а все зростаюча різниця між швидкодією пам'яті і процесора тільки посилює проблему.

Існують області застосування систем автоматичного розпізнавання мови, де описані проблеми проявляються особливо гостро через жорстко обмежених обчислювальних ресурсів, наприклад, на мобільних пристроях. Виробники мобільних телефонів і планшетів знайшли вихід в перенесенні ресурсномістких обчислень з пристроїв користувачів на сервери в хмарі, де, власне, і виконується розпізнавання. Користувацький додаток тільки відправляє туди мовні запити і приймає відповіді, використовуючи підключення до мережі Інтернет. За цією схемою успішно працюють системи Siri від Apple і Google Voice Search від Google. Однак, для такої реалізація необхідні певні умови, наприклад, безперервний доступ до мережі Інтернет, які в ряді випадків недосяжні, і потрібно створити компактний і надійний самостійний пристрій, що використовує тільки доступні «на місці» обчислювальні потужності. Описані труднощі виникають при створенні інтелектуальних пристроїв як у військовій сфері, так і в цивільній.

Зв'язок роботи з програмами наукових досліджень кафедри ПІ. Кафедра ПІ Харківського національного університету займається широким спектром досліджень у сфері інформаційних технологій, програмного забезпечення та програмної інженерії. Одним із напрямків дослідження є дослідження методів штучного інтелекту у різноманітних системах.

Мета і задачі дослідження.

– Аналіз існуючих моделей, методів і алгоритмів розпізнавання мовлення з метою виявлення ступеня їх відповідності сучасним вимогам і вибору прототипів для власних досліджень.

– Розробка моделей, методів, і алгоритмів розпізнавання мови, що забезпечують досягнення наступних показників розпізнавання голосових команд:

1) швидкість роботи, достатня для використання в режимі реального часу;

2) висока якість розпізнавання (95% правильно розпізнаних мовних команд в умовах відсутності шумовий складової - співвідношення сигнал / шум 25дБ);

3) легкість модифікації словника команд: можливість додавання нових слів і команд без перепрограмування системи.

Об'єктами дослідження є методи, алгоритми та системи розпізнавання мовлення.

Предмет дослідження є методи штучного інтелекту для реалізації системи розумного дому.

В якості *методів дослідження* використовуються методи цифрової обробки сигналів, теорії ймовірностей і математичної статистики, машинного навчання, прикладної лінгвістики, а також методи розробки програмного забезпечення.

Наукова новизна одержаних результатів. Запропоновано метод побудови інформативних ознак, що витягають з глибокої нейронної мережі з вузьким горлом, що відрізняється застосуванням адаптації до диктора і акустичних умов і дозволяє поліпшити якість акустичних моделей для спонтанного мовлення.

Практичне значення одержаних результатів. Застосування запропонованого методу розпізнавання в порівнянні з іншими підходами розпізнавання дозволяє системам адаптуватися до диктора і акустичних умов, що дозволяє покращити ступінь розпізнавання голосових команд.

Публікації по матеріалам роботи були розміщені в журналі «Наука онлайн» видавництва «Інтернаука» (статті присвоєно ідентифікатор DOI: 10.25313/2524-2695-2018-4-doslidzhennya-metodiv-shtuchnogo-intelektu-dlya-realizatsiyi-sistemi-rozumnogo-domu), також по матеріалам роботи були опубліковані та обговорені тези на XXII міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» (Додаток Б).

1 АНАЛІЗ ВИМОГ І ПІДХОДІВ ДО АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ МОВИ

Процес автоматичного розпізнавання мови являє собою перетворення акустичного сигналу, отриманого від мікрофона, в послідовність слів, яка потім може використовуватися для розуміння сенсу мовного висловлювання.

У першому розділі наведено аналіз основних вимог, які пред'являються до систем розпізнавання мови, представлена базова архітектура системи автоматичного розпізнавання мови, яка спирається на стохастичні моделі, а також здійснено огляд існуючих моделей розпізнавання мови.

1.1 Основні вимоги до сучасних систем розпізнавання мови

Завдання розпізнавання мови характеризується багатьма параметрами, в першу чергу, це властивості каналу передачі мови, розмір словника, варіативність мови, рівень навколишнього шуму, тип введення мови (ізольована / злита) [1].

Для розпізнавання ізольованих слів необхідно, щоб диктор робив короткі паузи між словами, що уповільнює введення і погіршує природність, в той час як при введенні злитої промови цього не потрібно. На відміну від друкованого тексту або від штучних сигналів природна мова не допускає простого і однозначного членування на елементи (фонеми, слова, фрази), оскільки ці елементи не мають явних фізичних кордонів. Вони вичленяються в свідомості слухача – носія цієї мови в результаті складного багаторівневого процесу розпізнавання і розуміння мови [2]. Якщо попросити слухача записати у вигляді фонем незнайому іноземну мову, то він зробить безліч помилок членування слів і фраз, тобто навіть людина не може членувати промову без використання знань лексики, граматики, сенсу. Межі слів можуть бути визначені лише в процесі

розпізнавання, за допомогою підбору оптимальної послідовності слів, найкращим чином узгоджується з вхідним потоком мови за акустичними і лінгвістичними критеріями.

Складність проблеми розпізнавання мови, головним чином, пов'язана з варіативністю її основних параметрів, на які впливає безліч чинників. Перш за все, це випадкова компонента процесу рече творення, яка призводить до різноманіття описів одного і того ж слова, сказаного одним і тим же диктором. Більш суттєва варіативність пов'язана з індивідуальними відмінностями мовних апаратів різних дикторів. Тут потрібно також відзначити вплив статі диктора, вікових відмінностей, діалектів, емоційного і фізичного стану диктора. Крім того, значний вплив вносить акустичний аспект, тобто зміна мікрофона, розташування його щодо рота, акустична обстановка в приміщенні.

Точність розпізнавання суттєво погіршується зі збільшенням словника, так як при цьому, з'являються групи акустично подібних слів, що призводить до акустичної неоднозначності, причому вона експоненціально посилюється з зростанням словника. Існує кілька можливих класифікацій розміру розпізнається словника. Малим словником вважається словник, що містить одиниці і десятки слів. Завдань і додатків, де використовується малий словник розпізнавання, дуже багато: розпізнавання послідовностей цифр (номерів телефонів); системи мовного командного управління рухомими технічними об'єктами (автомобілем, літаком, і т.д.), системи дистанційного керування роботами, системи управління обладнанням (наприклад, медичним) і т.д. Середній словник містить сотні слів. Такого словника досить для більшості діалогових або запитання-відповідь систем. Великий словник начиняється від тисяч слів, такі системи розпізнавання можуть використовуватися в автоматизованих довідкових системах або системах диктування в обмеженою предметної області. Словник розміром понад сотні тисяч слів вважається надвеликих розмірів і він дозволяє реалізовувати системи стенографії практично будь-якого тексту (для аналітичних мов) [3].

При роботі з реальною діалогової системою або при введенні тексту голосом користувач хоче отримати відповідь від системи негайно, він не готовий

чекати навіть кілька секунд, тому система, що розпізнає мову повинна працювати в режимі реального часу без істотних затримок в відповіді. Звичайно, існують задачі розпізнавання, де час реакції не відіграє суттєвої ролі, наприклад перетворення в текст архівних звукових записів, але число таких додатків дуже невелика.

Таким чином, найбільш важливими вимогами, яким повинні прагнути задовольнити сучасні системи автоматичного розпізнавання мови, є: злитий введення мови, дикторонезалежність, здатність розпізнавати велику кількість слів і високу швидкодію системи.

Вкрай важливим завданням є багатокритеріальне оцінювання таких складних інтелектуальних систем, як системи розпізнавання мови, і обґрунтований вибір оптимальних моделей і їх параметрів. Для оцінки ефективності розроблюваних систем автоматичного розпізнавання мови застосовують цілий ряд критеріїв на кожному з рівнів обробки мови, серед них два критерії є інтегральними: точність розпізнавання і час реакції (відповіді) системи. Ідеальна автоматична система повинна миттєво видавати безпомилковий результат.

1.2 Базові підходи до автоматичного розпізнавання мови

Завдання розпізнавання мови складається в підборі оптимальної послідовності моделей слів, яка найбільш імовірна (правдоподібна) оброблюваному мовному сигналу. Аналіз оглядових статей провідних світових вчених показав, що в даний час практично всі системи автоматичного розпізнавання мови будуються на основі декількох базових підходів (див. рис. 1.1):

- приховані марковські моделі;
- штучні нейронні мережі;

– динамічне програмування.

Рисунок 1.1 – Базові підходи до автоматичного розпізнавання мови

Довгий час підхід на основі динамічного програмування був домінуючим. Він дозволяє проводити порівняння мовного фрагмента зі створеним заздалегідь еталоном слова. Для того щоб порівняти слово з еталоном, треба шляхом деформації осі часу поєднати ділянки, відповідні одним і тим же звукам, виміряти залишкові відмінності між ними і підсумувати ці приватні відстані, взяті з деякими ваговими коефіцієнтами. Завдання динамічного програмування зводиться до пошуку оптимального нелінійного узгодження двох відрізків мовлення. Для цього широко використовувалися алгоритми динамічного програмування, що базуються на фундаментальних роботах Р. Беллмана [4]. Одна з перших публікацій по застосуванню динамічного програмування в розпізнаванні мови належить українському вченому Т.К. Вінцюку. Існує кілька підходів до розпізнавання злитого мовлення методами динамічного програмування:

– дворівневий алгоритм динамічного програмування;
– метод побудови рівнів (level-building);
– однопрохідний (one-pass) метод.

Алгоритми використовують однакові базові принципи і відрізняються обчислювальної складністю, обсягом пам'яті і складністю реалізації. Також був запропонований метод розпізнавання злитого мовлення на основі динамічного

програмування із застосуванням аналізу мови в ковзному вікні і теорії розмитих множин [5].

Основним недоліком підходів, заснованих на динамічному програмуванні, є їх дікторозалежність. Крім того, кожен новий користувач системи, перед тим як її використовувати, повинен створити свої еталони, тобто наговорити всі слова, які присутні в словнику. Для підвищення надійності розпізнавання під час запису еталонів користувачеві доводиться повторювати всі слова по кілька разів. З цієї причини такий підхід зараз використовується лише для додатків з малим словником, наприклад, виклик певного абонента в мобільних телефонах або персоніфіковане голосове управління офісними програмами.

Штучні нейронні мережі також використовуються при розпізнаванні мови. Вони являють собою спробу використання процесів, що відбуваються в нервових системах біологічних організмів. При правильно обраної структурі мережа, натренована на певному наборі навчальних вибірок, видаватиме правильні результати при подачі на її вхід даних, що відносяться до тої ж множини, але безпосередньо не беруть участі в процесі навчання. На практиці використовуються нейронні мережі, що мають один або кілька прихованих шарів нейронів між входом і виходом мережі. У цьому випадку складність мережі визначається кількістю нейронів в прихованому шарі, так як кількість нейронів у вхідному і вихідному шарах фіксоване і залежить від умов завдання. Поширеним є підхід, коли на входи нейронної мережі подаються вектора ознак мовного сигналу, а виходи мережі пов'язані з розпізнаваним словником (кількість виходів дорівнює кількості слів в словнику). Нейронні мережі здатні навчатися на голосах кількох дикторів, дозволяючи створювати дикторонезалежні системи розпізнавання, проте їх застосування для злитого мовлення важко, так як при злитому введенні невідома заздалегідь тривалість мовного сигналу, а відповідно і кількість векторів ознак, а також кількість і порядок виголошених слів, що значно ускладнює створення і навчання мережі. Однак нейронні мережі іноді застосовують в комбінованих з прихованими Марківськими моделями системах розпізнавання мови. В цьому випадку нейронні мережі задіюються або на рівні

попередньої обробки векторів ознак мови, або на рівні постобробки текстів гіпотез розпізнавання [6].

1.3 Приховані марковські моделі та моделі гаусівських домішок

Більшість сучасних систем автоматичного розпізнавання мови використовують приховані марковські моделі для обліку тимчасової варіативності мовного сигналу. Прихована марковська модель [7] задається:

– числом N станів в моделі і множиною станів $S = \{S_1, S_2, \ldots, S_N\}$. Стани моделі в проміжок часу t позначається q_t;

– множиною спостігаємих значень, які можуть породжуватися моделлю, спостерігаємі в момент часу t позначаються O_t;

– розподілом ймовірностей переходів між станами $A = \{a_{ij}\}$, $i,j = 1,2,\ldots,N$, де

$$a_{ij} = \mathsf{P}(q_t = S_j | q_{t-1} = S_i), \quad i,j = 1,2,\ldots,N. \tag{1.1}$$

– розподілом ймовірностей спостережень в стані S_j

$$\mathsf{P}(\mathbf{o}_t | S_j), \quad j = 1,2,\ldots,N. \tag{1.2}$$

– початковим розподілом ймовірностей станів $\pi = \{\pi_1;\pi_2;\ldots;\pi_N\}$, де

$$\pi_i = \mathsf{P}(q_1 = S_i), \quad i = 1,2,\ldots,N. \tag{1.3}$$

У задачах розпізнавання мови стану прихованих марковських мереж найчастіше моделюють фонеми (зазвичай використовується 3 стани на фонему), в якості спостереження розглядається вектор ознак, а для визначення того,

наскільки добре певний стан певної марковської моделі описує поточний кадр мовного сигналу, застосовуються моделі гауссових сумішей (Gaussian Mixture Models). В цьому випадку, щільність розподілу ймовірностей емісії задається сумішшю гауссових розподілів

$$b_i(\mathbf{o_t}) = \sum_{m=1}^{M} \frac{c_{i,m}}{(2\pi)^{D/2}|\mathbf{\Sigma}_{i,m}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{o}_t - \boldsymbol{\mu}_{i,m})^T \mathbf{\Sigma}_{i,m}^{-1}(\mathbf{o}_t - \boldsymbol{\mu}_{i,m})\right], \quad (1.4)$$

де набір параметрів A_i включає в себе ваги суміші $c_{i,m}$, вектори математичних очікувань гауссіан $\mu_{i,m}$ і коваріаційні матриці гауссіан $\Sigma_{i,m}$.

Нехай $q_1^T = (q_1, q_2, \ldots, q_T)$ – послідовність станів прихованих харківських мереж, $O_1^T = (O_1, O_2, \ldots, O_T)$ – послідовність спостережень. Імовірність породження мережею послідовності спостережень O_1^T для послідовності станів q_1^T визначається виразом

$$\begin{aligned} \mathsf{P}(\mathbf{o}_1^T|\mathbf{q}_1^T) &= \prod_{t=1}^{T} b_{q_t}(\mathbf{o}_t) = \\ &= \prod_{t=1}^{T}\sum_{m=1}^{M} \frac{c_{q_t,m}}{(2\pi)^{D/2}|\mathbf{\Sigma}_{q_t,m}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{o}_t - \boldsymbol{\mu}_{q_t,m})^T \mathbf{\Sigma}_{q_t,m}^{-1}(\mathbf{o}_t - \boldsymbol{\mu}_{q_t,m})\right] \end{aligned} \quad (1.5)$$

З іншого боку, ймовірність появи послідовності станів q_1^T являє собою множину ймовірностей переходів між станами мережі, тобто

$$\mathsf{P}(\mathbf{q}_1^T) = \pi_{q_1} \prod_{t=1}^{T-1} a_{q_t q_{t+1}} \quad (1.6)$$

Тоді спільна ймовірність появи послідовності спостережень O_1^T і послідовності станів O_1^T моделі є не що інше, як множення ймовірностей 1.5 і 1.6.

$$\mathsf{P}(\mathbf{o}_1^T, \mathbf{q}_1^T) = \mathsf{P}(\mathbf{o}_1^T|\mathbf{q}_1^T)\,\mathsf{P}(\mathbf{q}_1^T). \quad (1.7)$$

Повна ймовірність появи послідовності спостережень O_1^T для даної моделі визначається виразом

$$\mathsf{P}(\mathbf{o}_1^T) = \sum_{\mathbf{q}_1^T} \mathsf{P}(\mathbf{o}_1^T, \mathbf{q}_1^T) \tag{1.8}$$

і може бути обчислена за допомогою алгоритму прямого-зворотного ходу за час, пропорційне *T*.

Навчанням прихованих марковських мереж за критерієм максимальної правдоподібності (Maximum Likelihood) називається підстроювання параметрів моделі по заданій послідовності спостережень таким чином, щоб для модифікованої моделі збільшити ймовірність появи цієї послідовності спостережень. Таке навчання може виконуватися за допомогою ЕМ-алгоритму (алгоритму математичного очікування – модифікації) [8]. Маючи достатню кількість параметрів, моделі гауссових сумішей можуть описати розподіл ймовірностей з необхідною точністю. Точність розпізнавання мови за допомогою систем заснованих на базі моделі прихованих марковських систем може бути додатково підвищена за допомогою наступних технік:

а) Лінійні перетворення ознак, максимізує середнє правдоподібність, такі як Maximum Likelihood Linear Transformation і Semi-Tied Covariance [9].

б) Адаптація – корекція параметрів акустичної моделі для поліпшення якості її роботи в умовах, відмінних від умов навчання, або аналогічне перетворення простору ознак. Методи адаптації можна розділити на два сімейства: адаптація з учителем і адаптація без вчителя. При адаптації з учителем заздалегідь відомий розпізнається текст, за яким здійснюється настройка моделі, при адаптації без вчителя в якості еталонного тексту використовується результат розпізнавання. Широко використовуються такі техніки адаптації моделей, як лінійна регресія максимальної правдоподібності (Maximum Likelihood Linear Regression), лінійна регресія максимальної правдоподібності в просторі ознак (Feature-domain Maximum Likelihood Linear Regression), лінійна регресія максимальної апостеріорної ймовірності (Maximuma Posteriori Linear Regression)].

Застосування цих методів адаптації дозволяє скоротити помилку розпізнавання на 5-30% [10].

в) Діскрімінатівність навчання [11]. Після стандартного навчання система додатково навчається таким чином, щоб збільшити правдоподібність істинної гіпотези (пропозиції) щодо альтернативних гіпотез.

г) Доповнення вектора ознак ознаками, отриманими за допомогою нейронних мереж.

д) Використання контекстно-залежних фонем. У сучасних системах розпізнавання мови моделюють вони не ізольовані фонеми, а фонеми, вимовлені в контексті інших фонем. Як правило, використовуються трифони, тобто контекст з одного звуку зліва і праворуч від моделюємої фонеми. Очевидно, що кількість можливих трифонів дуже велике, і багато хто з них можуть не зустрітися в навчальній вибірці. Для вирішення цієї проблеми замість станів трифонів використовують так названі зв'язані стани, або сенони (senones) – стани трифонів об'єднуються в групи (наприклад, за допомогою дерева рішень), кожна з яких отримує загальний набір параметрів гауссових сумішей.

Незважаючи на широку поширеність в системах розпізнавання мови, акустичні моделі на основі прихованих марковських мереж та моделі гаусовських домішок мають ряд суттєвих недоліків [12]:

а) Вони статистично неефективні для моделювання даних, що лежать близько до кордонів або на кордонах нелінійних різноманіть. Так, наприклад, для моделювання даних, що лежать на кордоні сфери, буде потрібно величезна кількість діагональних і велика кількість полноковаріаційних гауссових сумішей. У той час як процес речотворення може бути описаний відносно невеликою кількістю параметрів.

б) У сучасних системах через вимоги до швидкості розпізнавання і навчання моделей застосовуються переважно суміші з діагональною матрицею коваріації, що тягне за собою необхідність використання некорельованих ознак. Це не дозволяє ефективно враховувати інформацію від суміжних кадрів –

необхідно застосування декореляцію (як правило, зі зменшенням розмірності і, отже, втратою інформації).

1.4 Акустичні моделі на основі глибоких нейронних мереж

1.4.1 Глибокі нейронні мережі

Альтернативним способом обчислення ймовірностей емісій є використання глибоких нейронних мереж (Deep Neural Network). Глибокої нейронною мережею прийнято називати штучну нейронну мережу (Artificial Neural Network) з двома або більше прихованими шарами. Глибока нейронна мережа з вхідним шаром, трьома прихованими шарами і вихідним шаром продемонстрована на рисунку 1.2.

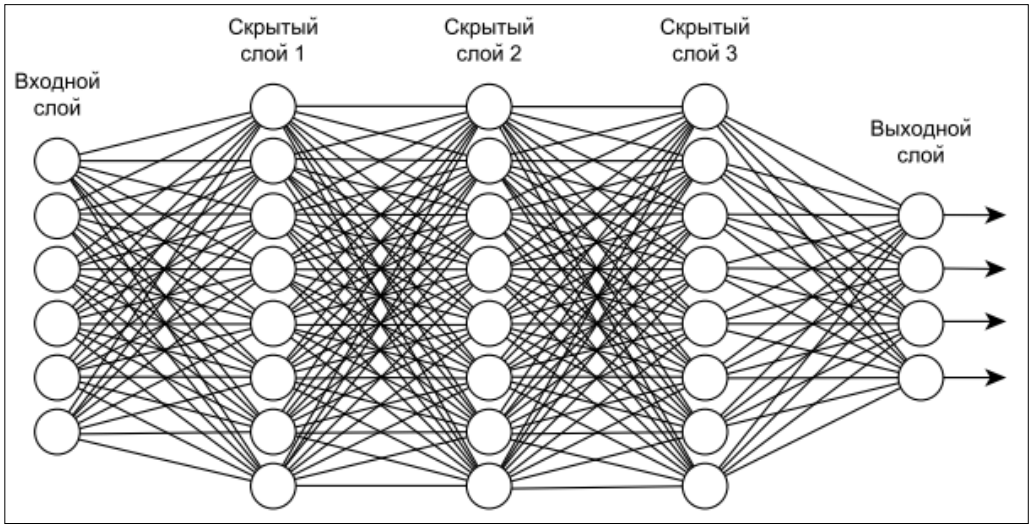


Рисунок 1.2 – Приклад глибокої нейронної мережі з вхідним шаром, трьома прихованими шарами і вихідним шаром

У нейронної мережі з $L + 1$ шарами будемо позначати вхідний шар як шар 0,

вихідний шар як шар L. Для вхідного шару і прихованих шарів виконується

$$\mathbf{v}^l = f(\mathbf{z}^l) = f(\mathbf{W}^l\mathbf{v}^{l-1} + \mathbf{b}^l), \quad 0 < l < L \tag{1.9}$$

де $z^l = W^l v^{l-1} + b^l \in R^N{}_l$, $v^l \in R^{Nl}$, $W^l \in R^{Nl \times Nl-1}$, $b^l \in R^{Nl}$ та $N^l \in R$ – відповідно, вектор індукованого локального поля, вектор активації, матриця ваг, вектор зміщення і кількість нейронів для шару l; $v^0 = o$ – вектор спостереження, або вектор ознак, $N_0 = D$ – розмірність вектора ознак;

$f(\cdot)$: $R^{Nl} \rightarrow R^{Nl}$ – функція активації, що застосовується поелементно до вектору індукованого локального поля. Найчастіше в якості функції активації використовується сигмо їда

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{1.10}$$

або гіперболічний тангенс

$$\text{th}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{1.11}$$

Функція активації для вихідного шару вибирається залежно від завдання. Для задач регресії використовується лінійний вихідний шар

$$\mathbf{v}^L = \mathbf{z}^L = \mathbf{W}^L\mathbf{v}^{L-1} + \mathbf{b}^L. \tag{1.12}$$

Для задач класифікації кожен вихідний нейрон відповідає за клас $i \in \{1, 2, \ldots, C\}$, де $C = N_L$ – число класів. У цих завданнях значення вихідного i-го нейрона зазвичай обчислюється за формулою

$$\mathbf{v}_i^L = \mathsf{P}_{dnn}(i|\mathbf{o}) = \text{softmax}_i(\mathbf{z}^L) = \frac{e^{z_i^L}}{\sum_{j=1}^{C} e^{z_j^L}} \tag{1.13}$$

і інтерпретується як імовірність того, що спостереження O належить класу i.

Маючи вектор спостережень O, вихід глибокої нейронної мережі, яка визначається набором параметрів $\Theta = \{W, b\} = \{W^l, b^l \mid 0 < l \leqslant L\}$, може бути обчислений за допомогою послідовного обчислення векторів активації відповідно до рівняння 1.9, починаючи з шару 1 і закінчуючи шаром L-1, і далі за допомогою рівняння 1.12 для задач регресії або рівняння 1.13 для задач класифікації. Цей процес називають прямим проходом (forward pass) [12].

1.4.2 Навчання глибоких нейронних мереж

Навчанням глибоких нейронних мереж називається настройка параметрів $\Theta = \{W, b\}$ по наявних навчальних прикладів $S = \{(o^m, y^m) \mid 0 \leqslant m < M\}$, де M – кількість прикладів, o^m і y^m – вектори спостережень і бажаний вихідний вектор для m-го прикладу. Процес навчання характеризується критерієм навчання і навчальним алгоритмом.

Критерій навчання повинен сильно корелювати з кінцевою метою завдання, щоб поліпшення навчального критерію приводило до поліпшення підсумкового результату.

У задачах класифікації y є розподіл ймовірностей акустичних класів, і часто використовується критерій мінімізації взаємної ентропії (Cross-Entropy)

$$J_{CE}(\mathbf{W},\mathbf{b};\mathbb{S}) = \frac{1}{M}\sum_{m=1}^{M} J_{CE}(\mathbf{W},\mathbf{b};\mathbf{o}^m,\mathbf{y}^m) \tag{1.14}$$

де

$$J_{CE}(\mathbf{W},\mathbf{b};\mathbf{o},\mathbf{y}) = -\sum_{i=1}^{C} y_i \log v_i^L, \tag{1.15}$$

$y_i = P_{emp}(i|o)$ є емпірична, тобто спостерігається по навчальних даних, ймовірність того, що спостереження o належить класу i, а v^L_i – та ж ймовірність, обчислена за

допомогою глибокої нейронної мережі. У більшості випадків використовуються жорсткі мітки класів, тобто

$$y_i = \begin{cases} 1, & c = i, \\ 0, & c \neq i, \end{cases} \tag{1.16}$$

де c – мітка класу в навчальних даних для спостереження O. В цьому випадку, критерій мінімізації взаємної ентропії, який визначається рівнянням 1.15, перетворюється в негативний логарифм правдоподібності (Negative Log-Likelihood)

$$J_{CE}(\mathbf{W},\mathbf{b};\mathbf{o},\mathbf{y}) = -\log v_c^L. \tag{1.17}$$

При наявному навчальному критерії параметри моделі $\{W,b\}$ можуть бути навчені за допомогою широко відомого алгоритму зворотного поширення помилки (Error Backpropagation), що полягає у використанні правила диференціювання складної функції для обчислення градієнта. У найпростішому вигляді, параметри моделі оновлюються відповідно до формул

$$\mathbf{W}_{t+1}^l = \mathbf{W}_t^l - \varepsilon \Delta \mathbf{W}_t^l, \tag{1.18}$$

$$\mathbf{b}_{t+1}^l = \mathbf{b}_t^l - \varepsilon \Delta \mathbf{b}_t^l, \tag{1.19}$$

де W^l_t і b^l_t є матрицю ваг і вектор зміщення для шару l після t-го поновлення,

$$\Delta \mathbf{W}_t^l = \frac{1}{M_b} \sum_{m=1}^{M_b} \nabla_{\mathbf{W}_t^l} J(\mathbf{W}_t,\mathbf{b}_t;\mathbf{o}^m,\mathbf{y}^m), \tag{1.20}$$

$$\Delta \mathbf{b}_t^l = \frac{1}{M_b} \sum_{m=1}^{M_b} \nabla_{\mathbf{b}_t^l} J(\mathbf{W}_t,\mathbf{b}_t;\mathbf{o}^m,\mathbf{y}^m), \tag{1.21}$$

є, відповідно, середній градієнт матриці ваг і середній градієнт вектора зміщення на ітерації t, обчислені за навчальною порції (batch), що містить M_b прикладів, $\mathcal{E}$– швидкість навчання, а $\nabla_x J$ – градієнт функції J по відношенню до x. Незважаючи на простоту алгоритму зворотного поширення помилки, для ефективного навчання мережі необхідно приділяти увагу практичним питанням, найбільш значимі з яких перераховані нижче.

а) Нормалізація вхідних ознак до нульового середнього і одиничною дисперсії. Здійснюється з метою приведення вхідних даних до близького діапазону чисельних значень, що дозволяє використовувати одну і ту ж швидкість навчання для всіх ваг.

б) Ініціалізація навчання моделі – існує велика кількість евристик. Згідно з однією з них, важливо ініціювати параметри випадковим чином, оскільки в іншому випадку різні нейрони будуть визначати одні і ті ж шаблони ознак на нижніх шарах. Для нейронних мереж з прихованими шарами розміру 1000–2000, які зазвичай використовуються в розпізнаванні мови, ефективно працює ініціалізація матриць ваг гаусовим розподілом з нульовим середнім і дисперсією 0,05, або рівномірним розподілом в діапазоні [-0.05,0.05]. Вектори зсувів можна форматувати нулями.

в) Додавання регулярізуючого доданка $R(W)$ до критерію навчання

$$\ddot{J}(\mathbf{W},\mathbf{b};\mathbb{S}) = J(\mathbf{W},\mathbf{b};\mathbb{S}) + \lambda R(\mathbf{W}), \tag{1.22}$$

де λ називають вагою регуляризації. Одним з найбільш часто використовуваних варіантів регулярізуючого доданка є по елементна p-норма матриці ваг (зазвичай $p = 1$ або $p = 2$), яка визначається згідно з формулою

$$\|\mathbf{W}\|_p = \left(\sum_{i,j} |\mathbf{W}_{ij}|^p \right)^{1/p} \tag{1.23}$$

Регуляризація застосовується для уникнення перенавчання (overfitting) – явища, при якому побудована модель добре пояснює приклади з навчальної вибірки, але

погано працює на прикладах, які не брали участі в навчанні. Це особливо актуально при маленьких розмірах навчальної вибірки.

г) Вибір розміру навчальної порції впливає і на швидкість збіжності, і на якість навчання. Найпростіший спосіб – брати в якості навчальної порції все навчальні дані (full-batch training), в цьому випадку обчислюється точний градієнт по навчальних даних. Недоліками цього способу, що проявляються на великих навчальних вибірках, є, по-перше, низька швидкість навчання, і, по-друге, схильність до попаданню в поганій локальний мінімум. Альтернативою є метод стохастичного градієнтного спуску (Stochastic Gradient Descent), при якому оновлення параметрів моделі відбувається після кожного навчального прикладу. Неточна оцінка градієнта в цьому випадку є перевагою, а не недоліком, оскільки дозволяє уникнути поганих локальних мінімумів і перенавчання. До недоліків цього методу можна віднести труднощі в розпаралелюванні і неможливість досягнення повної збіжності. Компромісом між fullbatch training і Stochastic Gradient Descent є оцінка градієнта і оновлення параметрів моделі по малій порції даних, випадковим чином обраної з навчальних прикладів (minibatch training). Розмір порції, використовуваний в задачах розпізнавання мови, зазвичай становить 128 – 1024 прикладів.

д) Використання накопиченого градієнта дозволяє домогтися прискорення збіжності на пологих ділянках. Однією з таких технік є «момент» (momentum), при використанні якого параметри моделі θ оновлюються відповідно до формул

$$\boldsymbol{\Theta}_t = \boldsymbol{\Theta}_{t-1} + \boldsymbol{\nu}_t, \tag{1.24}$$

$$\boldsymbol{\nu}_t = \mu_{t-1}\boldsymbol{\nu}_{t-1} - \varepsilon_{t-1}\frac{1}{M_b}\sum_{m=1}^{M_b}\nabla J(\boldsymbol{\Theta}_{t-1}; \mathbf{o}^m, \mathbf{y}^m), \tag{1.25}$$

де $\mathcal{E}$ і μ_t – відповідно швидкість навчання і коефіцієнт моменту на ітерації t. У завданнях розпізнавання мови часто використовується постійне значення коефіцієнта μ в діапазоні від 0,5 до 0,9. Більш складна техніка, призначена для

поліпшення стійкості і збіжності градієнтного спуску – прискорений градієнт Нестерова (Nesterov Accelerated Gradient). Як і для «моменту», оновлення параметрів моделі здійснюється за формулою 1.24, але замість формули 1.25 використовується наступна:

$$\boldsymbol{\nu}_t = \mu_{t-1}\boldsymbol{\nu}_{t-1} - \varepsilon_{t-1}\frac{1}{M_b}\sum_{m=1}^{M_b}\nabla J(\boldsymbol{\Theta}_{t-1} + \mu_{t-1}\boldsymbol{\nu}_{t-1}; \mathbf{o}^m, \mathbf{y}^m) \qquad (1.26)$$

Проста реалізація прискореного градієнту Нестерова використовує для оновлення параметрів моделі формулу

$$\boldsymbol{\Theta}_t = \boldsymbol{\Theta}_{t-1} - \mu_{t-1}\boldsymbol{\nu}_{t-1} + \mu_t\boldsymbol{\nu}_t + \boldsymbol{\nu}_t, \qquad (1.27)$$

де v_t визначається згідно з формулою 1.25.

е) Вибір розкладу зміни швидкості навчання має суттєвий вплив на якість навчання нейронної мережі. Існує велика кількість методик. У завданнях розпізнавання мови популярний алгоритм «newbob», який полягає в здійсненні кількох повних проходів навчання за всіма даними (так названих епох навчання) з постійною швидкістю. Як тільки абсолютне зменшення помилки класифікації кадрів на кроссвалідаційній вибірці (порція даних, обрана випадковим чином і не бере участі в навчанні) виявиться менше певного порогу (наприклад, 0,5%), швидкість для кожної наступної епохи зменшується в кілька разів (наприклад, в 2 рази). Навчання зупиняється, як тільки абсолютне зменшення помилки класифікації кадрів виявиться досить малим (наприклад, менше 0,1%). Інша проста і ефективна техніка полягає в зменшенні швидкості навчання для наступної епохи в кілька разів (наприклад, в два рази), якщо відносне поліпшення критерію навчання на кроссвалідаційній вибірці після поточної епохи виявилося менш певного порогового значення (наприклад, 0,01).

ж) Вибір архітектури нейронних мереж сильно впливає на ефективність її роботи. У задачах розпізнавання мови зазвичай застосовуються глибокі

нейронні мережі, що мають 5–7 прихованих шарів по 1000–2000 нейронів в кожному [12].

1.4.3 Більш досконалі техніки ініціалізації навчання глибоких нейронних мереж

До недавнього часу глибокі нейронні мережі не мали широкого поширення через відсутність високопродуктивного апаратного забезпечення, необхідного для їх якісного навчання. Ще однією причиною було те, що без акуратною ініціалізації початкових параметрів алгоритм зворотного поширення помилки погано працює для багатошарових мереж через проблеми з експоненціальним загасанням або зростанням градієнта (gradient vanishing and exploding problem), які призводять до розбіжності алгоритму або знаходження поганого локального екстремуму. Цим проблемам присвячена багато робіт [13]. У 2006 році в області машинного навчання стався прорив – Geoffrey Hinton запропонував алгоритм навчання багатошарових нейронних, що складається з двох етапів:

– жадібне пошарове пренавчання – використовується для ініціалізації параметрів глибокої нейронної мережі;

– тонка настройка (fine-tuning) – корекція ваг за допомогою алгоритму зворотного поширення помилки.

В оригінальній роботі для пренавчання використовувалися обмежені машини Больцмана (Restricted Boltzmann Machine).

Обмежена машина Больцмана представляє собою енергетичну модель, в якій кожної конфігурації N_v видимих змінних v і N_h прихованих змінних h ставиться у відповідність енергія $E(v, h)$. Для обмеженої машини Больцмана Бернуллі – Бернуллі, у якій $v \in \{0,1\}^{Nv}$ і $h \in \{0,1\}^{Nh}$, енергія визначається виразом

$$E(\mathbf{v},\mathbf{h}) = -\mathbf{a}^T\mathbf{v} - \mathbf{b}^T\mathbf{h} - \mathbf{h}^T\mathbf{W}\mathbf{v}, \tag{1.28}$$

де W є $R^{Nv \times Nh}$ – матриця ваг, a є R^{Nv} – вектор зсувів спостережуваних змінних, b є R^{Nh} – вектор зсувів прихованих змінних. У випадку обмеженої машини Больцмана Гаусса-Бернуллі спостерігаються змінні приймають дійсні значення, а функція енергії визначається виразом

$$E(\mathbf{v},\mathbf{h}) = \frac{1}{2}(\mathbf{v}-\mathbf{a})^T(\mathbf{v}-\mathbf{a}) - \mathbf{b}^T\mathbf{h} - \mathbf{h}^T\mathbf{W}\mathbf{v}. \quad (1.29)$$

Обмежену машину Больцмана можна представити у вигляді графічної ймовірнісної моделі, в якій вузли прихованих і спостережуваних змінних об'єднані в двочастковий граф з двонаправленими зв'язками від прихованих змінних до спостережуваних і назад, але без зв'язків між різними прихованими або різними спостерігаються змінними (див. рис. 1.3).

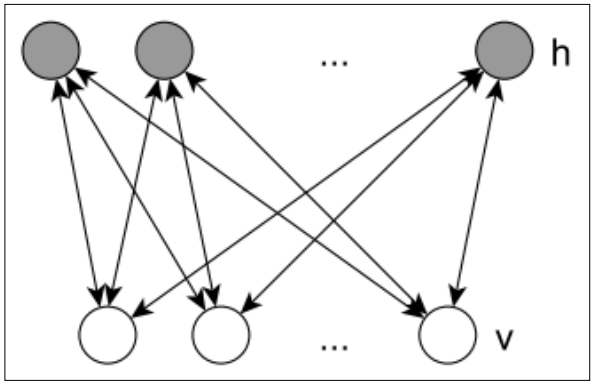


Рисунок 1.3 – Обмежена машина Больцмана

Також кожної конфігурації змінних ставиться у відповідність імовірність

$$\mathrm{P}(\mathbf{v},\mathbf{h}) = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}. \quad (1.30)$$

Можна показати, що як для обмеженої машини Больцмана Бернуллі-Бернуллі, так і для обмеженої машини Больцмана Гаусса-Бернуллі

$$\mathrm{P}(\mathbf{h}=\mathbf{1}|\mathbf{v}) = \sigma(\mathbf{W}\mathbf{v}+\mathbf{b}), \quad (1.31)$$

де під $\sigma(\cdot)$ розуміється сигмоїда 1.10, застосована до кожної компоненті вектора.

Оскільки для прихованого шару нейронної мережі з функцією активації – сигмоідою вектор активації визначається як 1.31, можна використовувати комбінацію з N обмежених машин Больцмана для ініціалізації навчання глибокої нейронної мережі з N прихованими шарами. Для цього спочатку навчається обмежена машина Больцмана Гаусса-Бернуллі, в якій в якості спостережуваних змінних виступають вектори ознак. Потім послідовно навчаються N-1 обмежених машин Больцмана Бернуллі-Бернуллі, в яких в якості спостережуваних змінних використовуються значення прихованих змінних попередньої обмеженої машини Больцмана. Така комбінація обмежених машин Больцмана називається глибокої мережею довіри (Deep Belief Network). Далі кожен прихований шар глибокої мережі довіри ініціалізується значеннями W і b відповідної обмеженої машини Больцмана. Нарешті, додається ініціалізований випадковим чином вихідний softmax-шар.

У більш пізніх роботах було показано, що для пошарового пренавчання можна також використовувати автоенкодери – нейронні мережі з одним прихованим шаром, що навчаються відтворювати свої вхідні дані. Використання автоенкодерів засноване на розумінні, що прихований шар буде захоплювати основні закономірності у вхідних даних. Щоб обійти потенційну проблему автоенкодерів – навчання на одиничну функцію, не захоплюючу ніяких закономірностей з вхідних даних, застосовують кілька технік. Одна з них, що застосовується в шумоподавляючих автоенкодерах (denoising autoencoders), полягає в додаванні випадкового шуму до вхідних даних. Найпростішим способом зашумлення вхідних даних є завдання випадкового підмножини вхідного вектора нулями. Таким чином, шумоподавляючий автоенкодер не тільки витягує закономірності з вхідних даних, але і бореться з ефектом випадкового ушкоджуваного процесу, що дозволяє додати більш стійкі ознаки [14].

Процедури пренавчання за допомогою автоенкодерів та обмежених машин Больцмана володіють близькими властивостями: вони є генеративних, або породжуючими (тобто побудовані в результаті пренавчання модель може бути

використана для породження нових даних) і не вимагають розмічених даних. Це дозволяє починати тонке налаштування параметрів нейронних мереж з відносно хорошою початкової точки і надає неявний регулярізуючій ефект.

Альтернативою генеративному пренавчанню є дескрімінатівне пренавчання (discriminative pretraining), що здійснюється так само пошарово, але з використанням розмічених навчальних даних. На відміну від породжуючих алгоритмів, модель, побудована в результаті дескрімінатівного пренавчання, моделює умовний розподіл ймовірностей неспостерігаємих змінних (сенонов) до спостерігаємих (ознаки). Одним з варіантів дескрімінатівного пренавчання є пошарове зворотне поширення помилки (layer-wise error backpropagation) (див. рис. 1.4).

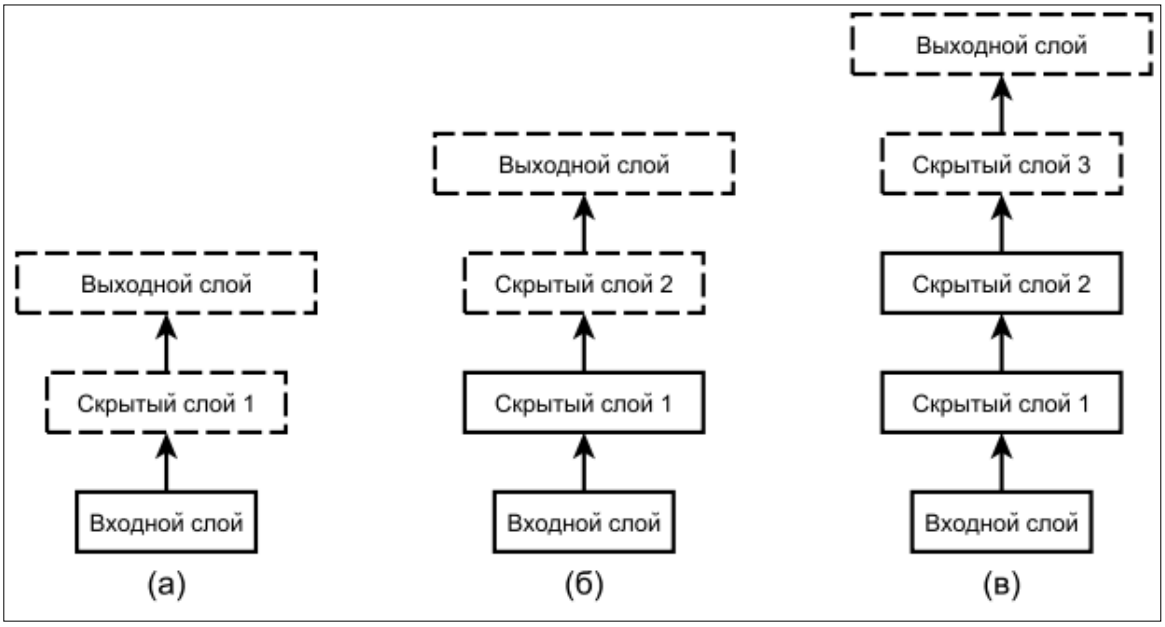


Рисунок 1.4 – Дескримінативне пренавчання нейронних мереж

При цьому спочатку навчається нейронна мережа з одним прихованим шаром $v1$ з використанням міток (див. рис. 1.4 (а)). Потім вихідний шар видаляється, додаються другий прихований шар $v2$ і новий вихідний шар, ваги яких започатковано випадковим чином (див. рис. 1.4 (б)), і знову відбувається навчання. Цей процес повторюється, поки не буде досягнуто бажане число шарів.

Інший варіант дескрімінативного пренавчання, званий жадібним пошаровим навчанням (Greedy layer-wise training), відрізняється від вищеописаного тим, що оновлюються тільки параметри нових доданих шарів. Щоб уникнути попадання нейронів в діапазон насичення рекомендується не проводити навчання до повної збіжності, а замість цього виконувати лише кілька ітерацій навчання на кожному кроці. Метою дескрімінативного пренавчання є приведення ваг моделі до хорошого локальному екстремуму. При цьому регулярізуючий ефект генеративного пренавчання відсутній, тому дескрімінативне пренавчання найкраще працює на великих обсягах навчальних даних.

1.4.4 Навчання глибоких нейронних мереж з використанням критеріїв поділу послідовностей

Критерій мінімізації взаємної ентропії, описаний у попередніх підрозділах, розглядає кожен кадр вхідних даних незалежно. Однак розпізнавання мови є завданням класифікації послідовностей. Навчання акустичних моделей на основі глибоких нейронних мереж з використанням критеріїв поділу послідовностей (Sequence-Discriminative Training) враховує цю особливість і завдяки цьому дозволяє досягти на 3-17% зменшення помилок розпізнавання в порівнянні з нейронною мережею, навченими за критерієм мінімізації взаємної ентропії.

Позначимо через $o^m = o^m_1,\ldots, o^m_t,\ldots, o^m_{Tm}$, $w^m = w^m_1,\ldots, w^m_t,\ldots, w^m_{Nm}$ та $s^m = s^m_1,\ldots,s^m_t,\ldots,s^m_{Tm}$ відповідно послідовність спостережень, еталонну текстову розшифровку і послідовність станів марковької мережі, відповідних еталонної текстової розшифровки для пропозиції m з навчальної вибірки, де T_m – число кадрів в реченні m, N_m – число слів в текстовій розшифровці цього речення.

Критерій максимуму взаємної інформації (Maximum Mutual Information) [15] націлений на максимізацію взаємної інформації між розподілами послідовності спостережень і послідовності слів. За навчальною вибіркою

$S = \{(o^m, w^m, s^m) \mid 0 \leqslant m < M\}$, де M є повне число пропозицій в навчальній вибірці, яке визначається за формулою

$$J_{MMI}(\Theta; \mathbb{S}) = \sum_{m=1}^{M} J_{MMI}(\Theta; \mathbf{o}^m, \mathbf{w}^m, \mathbf{s}^m) = \\ = \sum_{m=1}^{M} \log \mathsf{P}(\mathbf{w}^m \mid \mathbf{o}^m : \Theta) = \sum_{m=1}^{M} \log \frac{\mathsf{P}(\mathbf{o}^m \mid \mathbf{s}^m : \Theta)^k \mathsf{P}(\mathbf{w}^m)}{\sum_{\mathbf{w}} \mathsf{P}(\mathbf{o}^m \mid \mathbf{s}^{\mathbf{w}} : \Theta)^k \mathsf{P}(\mathbf{w})}, \quad (1.32)$$

де k – масштабуючий коефіцієнт, а під s^w розуміється послідовність станів прихованої марковської моделі, відповідна послідовності слів w. Теоретично, суму в знаменнику слід вважати за всіма можливими послідовностям слів, проте на практиці для зниження обчислювальної складності її зазвичай вважають за списком гіпотез, отриманих в результаті розпізнавання пропозиції m.

Аналогічно до критерію максимуму взаємної інформації, критерій посиленого максимуму взаємної інформації (Boosted Maximum Mutual Information) визначається за формулою

$$J_{BMMI}(\Theta; \mathbb{S}) = \sum_{m=1}^{M} J_{BMMI}(\Theta; \mathbf{o}^m, \mathbf{w}^m, \mathbf{s}^m) = \\ = \sum_{m=1}^{M} \log \frac{\mathsf{P}(\mathbf{w}^m \mid \mathbf{o}^m : \Theta)}{\sum_{\mathbf{w}} \mathsf{P}(\mathbf{w} \mid \mathbf{o}^m : \Theta) \mathrm{e}^{-bA(\mathbf{w}, \mathbf{w}^m)}} = \quad (1.33) \\ = \sum_{m=1}^{M} \log \frac{\mathsf{P}(\mathbf{o}^m \mid \mathbf{s}^m : \Theta)^k \mathsf{P}(\mathbf{w}^m)}{\sum_{\mathbf{w}} \mathsf{P}(\mathbf{o}^m \mid \mathbf{s}^{\mathbf{w}} : \Theta)^k \mathsf{P}(\mathbf{w}) \mathrm{e}^{-bA(\mathbf{w}, \mathbf{w}^m)}},$$

де b – коефіцієнт посилення, а функція $A(w, w^m)$ визначає точність відповідності між послідовностями слів w і w^m і може обчислюватися на рівні слів, фонем або станів прихованих марковських систем.

Критерії сімейства мінімального Байєсова ризику (Minimum Bayes Risk) націлені на мінімізацію очікуваної помилки на рівні фонем (Minimum Phone Error)

або станів прихованої морковської моделі (state Minimum Bayes Risk) і визначаються за формулою

$$J_{MBR}(\Theta;\mathbb{S}) = \sum_{m=1}^{M} J_{MBR}(\Theta;\mathbf{o}^m,\mathbf{w}^m,\mathbf{s}^m) = \\ = \sum_{m=1}^{M}\sum_{\mathbf{w}} \mathsf{P}(\mathbf{w}\,|\,\mathbf{o}^m;\,\Theta)A(\mathbf{w},\mathbf{w}^m) = \\ = \sum_{m=1}^{M} \frac{\sum_{\mathbf{w}} \mathsf{P}(\mathbf{o}^m\,|\,\mathbf{s}^{\mathbf{w}};\,\Theta)^k\,\mathsf{P}(\mathbf{w})A(\mathbf{w},\mathbf{w}^m)}{\sum_{\mathbf{w}'} \mathsf{P}(\mathbf{o}^m\,|\,\mathbf{s}^{\mathbf{w}'};\,\Theta)^k\,\mathsf{P}(\mathbf{w}')}, \tag{1.34}$$

де в якості опції $A(w,w^m)$, що визначає точність відповідності між послідовностями слів w і w^m, береться число фонем, що збіглися, для критерію очікуваних помилок на рівні фонем і число станів, що збіглися, прихованої марківскої моделі для критерію очікуваних помилок станів прихованої морковської моделі.

В якості навчальної порції при навчанні з використанням критеріїв поділу послідовностей використовується ціле речення. Для ініціалізації навчання зазвичай використовується глибока нейронна мережа, навчена по критеріям мінімізації взаємної ентропії. Ця ж глибока нейронна мережа найчастіше (хоча і не завжди) використовується для розмітки навчальних пропозицій на стану прихованої марковської системи і для генерації гіпотез розпізнавання навчальних пропозицій [16].

Порівняння різних критеріїв поділу послідовностей в завданні розпізнавання англійської мови, показує незначну перевагу state MBR критерію над іншими. Результати цього порівняння наведено в таблиці 1.1.

Навчання DNN з використанням критеріїв поділу послідовностей володіє схильністю до перенавчання: нерідко виникає ситуація, коли послідовний критерій поліпшується, але при цьому значно погіршується точність класифікації кадрів. Для боротьби з цим запропонована техніка, названа кадровим згладжуванням (Frame Smoothing, F-Smoothing): замість мінімізації одного лише

послідовного критерію здійснюється мінімізація зваженої суми послідовного критерію і взаємної ентропії

$$J_{FS}(\mathbf{W}, \mathbf{b}; \mathbb{S}) = (1 - H) J_{CE}(\mathbf{W}, \mathbf{b}; \mathbb{S}) + H J_{SEQ}(\mathbf{W}, \mathbf{b}; \mathbb{S}). \quad (1.35)$$

Таблиця 1.1 – Порівняння різних критеріїв поділу послідовностей в завданні розпізнавання англійської спонтанної мови на підвибірках Switchboard з тестових вибірок HUB5 Eval 2000 і HUB5 Eval 2001

Акустична модель	WER HUB5 2000 (sw)	WER HUB5 2001 (sw)
GMM bMMI	18,6	18,9
DNN CE	14,2	14,5
DNN MMI	12,9	13,3
DNN bMMI	12,9	13,2
DNN MPE	12,9	13,2
DNN sMBR	12,6	13,0

У задачі розпізнавання англійської спонтанної мови ця техніка дозволяє отримати до 4% відносного зменшення помилки розпізнавання.

1.4.5 Методи адаптації акустичних моделей на основі глибоких нейронних мереж

Однією з головних причин помилок, що виникають при роботі систем автоматичного розпізнавання мови, є невідповідність умови навчання і експлуатації. Для поліпшення точності розпізнавання в умовах, відмінних від умов навчання, розробляються алгоритми адаптації акустичних моделей. На відміну від методів нормалізації, методи адаптації нерозривно пов'язані з типом використовуваної акустичної моделі і призначені для підстроювання моделі під конкретні умови експлуатації. Для CD-DNN-HMM розроблено велику кількість методів адаптації.

Одним з напрямків є виділення і налаштування підмножини параметрів нейронної мережі. До них відносяться:

– адаптація лінійного вхідного шару. Суть методу полягає в тому, що параметри всіх шарів дикторонезалежної нейронної мережі, крім першого, фіксуються і на даних певного диктора алгоритмом зворотного поширення помилки здійснюється настройка параметрів першого шару;

– адаптація лінійного прихованого шару;

– адаптація лінійного вихідного шару;

– дескрімінативна лінійна регресія в просторі ознак (Feature Discriminant Linear Regression, fDLR). Є різновидом алгоритму адаптації лінійного вхідного шару, в якій для кожного кадру застосовується однакове перетворення, тобто відповідні параметри матриці перетворення є спільними. У порівнянні з методом адаптації лінійного вхідного шару, даний метод має меншу кількість параметрів, що налаштовуются, тому він менш схильний до перенавчання і демонструє кращу якість роботи при наявності невеликої кількості адаптаційних даних;

– використання дікторозалежного шару. Ідея методу полягає в тому, що найбільш чутливі до міждикторскої варіативності параметри нейронної мережі локалізовані в певному шарі багатошарової нейронної мережі. Найбільшою чутливістю володіють параметри другого шару. Запропонована схема адаптації складається з трьох етапів. На першому етапі навчається дикторонезалежна нейронна мережа. На другому етапі також виконується навчання нейронної мережі алгоритмом зворотного поширення помилки, але для кожного диктора використовується індивідуальний набір параметрів дікторозалежного шару (другого). На третьому етапі фіксуються параметри всіх дикторонезалежних шарів, отримані на другому етапі, і за даними цільового диктора налаштовуються параметри дікторозалежного шару;

– факторизація параметрів нейронної мережі і подальше виділення дікторозалежного фактору. Відомо, що багатошарові нейронні мережі мають велику надмірністю, зокрема, велика частина параметрів близька до нуля. Це дозволяє представити параметри нейронної мережі в більш компактному вигляді

без втрати якості. Одним із способів для скорочення кількості параметрів є сингулярне перетворення. За допомогою сингулярного перетворення матриці ваг представляються у вигляді добутку двох матриць, які мають істотно меншу розмірність в порівнянні з вихідною – формується так назване вузьке горло (bottleneck). Отримана після факторизації мережа заново навчається. Подібний спосіб навчання часто дозволяє не тільки не погіршити точність розпізнавання, але і трохи поліпшити. Пропонується два методи адаптації з використанням факторизации параметрів нейронної мережі. У першому методі в факторізовану за допомогою сингулярного перетворення нейронну мережу між відповідними лівими і правими матрицями вставляються квадратні поодинокі матриці, які потім налаштовуються у процесі адаптації, інші параметри при цьому залишаються фіксованими. У другому методі передбачається, що матриці різниці параметрів дикторозалежної і дикторонезжної мереж матиме низький ранг. Ці різниці піддаються сингулярному перетворенню. Решта кроків ідентичні першому методу [17].

Інший напрямок – налаштування всіх параметрів DNN з використанням в цільової функції додаткового регулярізуючого доданка, що не дозволяє налаштованим параметрам занадто сильно відхилитися від вихідної моделі. Як регуляризатора застосовують:

– L2-штраф на зміну параметрів моделі;

– дивергенцію Кульбака-Лейблера вихідного розподілу сенонів.

Надання нейронної мережі додаткової інформації про фонограму або її ділянки також є одним із шляхів до адаптації DNN-HMM. В даній групі можна виділити:

а) Використання дикторських кодів для швидкої адаптації до диктора. Ідея методу полягає в тому, щоб в просторі ознак навчити додаткову вхідну мережу, на вхід кожного шару якої подаються не тільки виходи попереднього шару або, для першого шару, акустичні ознаки, а й спеціально навчаюваний дикторський код, який представляє собою малорозмірний вектор дикторських характеристик. При цьому адаптаційна мережа вчиться по всім навчальним даним

і не змінюється в залежності від диктора, а дикторські коди навчаються для кожного диктора тільки за його даними.

Навчання акустичної моделі виконується в два етапи. На першому етапі стандартним способом навчається дикторонезалежна нейронна мережа. На другому етапі навчаються адаптаційна нейронна мережа та коди дикторів; параметри дикторонезалежної нейронної мережі при цьому залишаються незмінними, параметри адаптаційної мережі налаштовуються по всіх навчальних даних, а коди кожного диктора налаштовуються тільки за його даними.

Адаптація за допомогою даного методу застосовується в режимі роботи з учителем, тобто передбачається наявність точних текстових розшифровок і розмітки адаптаційної вибірки на дикторів. В процесі адаптації параметри обох нейронних мереж не змінюються, а налаштовуються тільки дикторський коди, які потім подаються на вхід адаптаційної нейронної мережі.

б) Адаптація за допомогою *i*-векторів. У задачі ідентифікації диктора вектор акустичних ознак x_t розглядається як згенерований з моделі гауссових сумішей з діагональними ковариаційними матрицями, званої також універсальної фонової моделлю (Universal Background Model, UBM), яка навчається за великим обсягом фонограм.

$$\mathbf{x}_t \sim \sum_{k=1}^{K} c_k \mathcal{N}(\cdot; \boldsymbol{\mu}_k(0); \boldsymbol{\Sigma}_k). \tag{1.36}$$

При цьому вектор акустичних ознак $x_t(s)$, що належить дикторові s, вважається згенерованим з адаптованої до цього диктора моделі Гауссових сумішей.

$$\mathbf{x}_t(s) \sim \sum_{k=1}^{K} c_k \mathcal{N}(\cdot; \boldsymbol{\mu}_k(s); \boldsymbol{\Sigma}_k). \tag{1.37}$$

Ідея методу *i*-векторів полягає в припущенні, що існує лінійна залежність між дикторозалежними математичними очікуваннями $\mu k(s)$ і дикторонезалежними математичними очікуваннями $\mu_k(0)$, яка визначається виразом 1.38

$$\boldsymbol{\mu}_k(s) = \boldsymbol{\mu}_k(0) + \mathbf{T}_k\mathbf{w}(s), \tag{1.38}$$

де $w(s)$ – вектор, що характеризує диктора s, або i-вектор. Таким чином, i-вектор являє собою малорозмірний вектор, кодуючий відмінність щільності розподілу ймовірностей акустичних ознак, оціненої по фонограмі, від еталонної. I-вектор містить канальну і дикторську інформацію. I-вектори широко застосовуються в задачі ідентифікації диктора. Метод адаптації DNN за допомогою i-векторів полягає в додаванні до вектору акустичних ознак і вектора, обчисленого за фрагментом фонограми, яка відповідає певному дикторові (див. рис 1.5). Таким чином, здійснюється адаптація як до диктора, так і до акустичної обстановки.

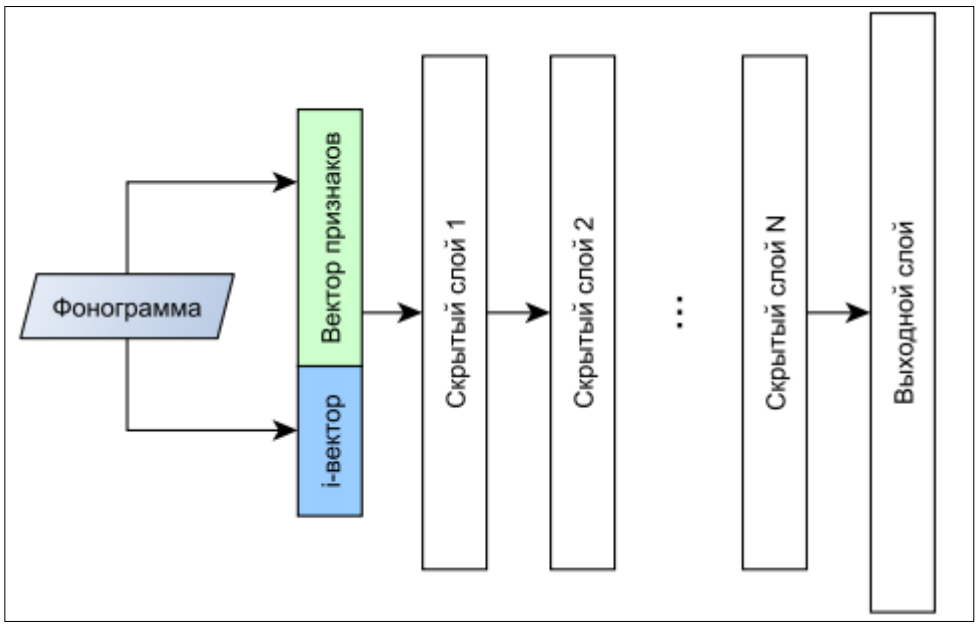


Рисунок 1.5 – Схема адаптації з використанням i-векторів

Для надійної оцінки i-вектора необхідна наявність достатньої кількості даних, що припадають в середньому на одного диктора (десятки секунд). У цьому випадку навчання можна проводити стандартним чином. Якщо даних недостатньо, навчання можна проводити в два етапи. На першому етапі виконується стандартне навчання дикторонезалежної нейронної мережі. На другому етапі вхідний прошарок нейронної мережі розширюється до розмірності

вектора ознак, доповненого *i*-векторами. Відповідні параметри ініціалізуються нулями, і виконується донавчання нейронної мережі по дикторозалежним ознакам зі штрафом на відхилення параметрів від параметрів дикторонезалежної мережі і меншою початковою швидкістю навчання.

в) Використання акустичних факторів. Метод застосовується для адаптації до каналу. Суть методу полягає у виділенні з мовного сигналу факторів, що характеризують акустичну обстановку і додаванні цих факторів на вхід вихідного шару нейронної мережі. Кожному фактору встановлюється відповідно індивідуальна матриця параметрів, яка налаштовується в процесі адаптації за допомогою алгоритму зворотного поширення помилки. Виділення акустичних чинників може здійснюватися за допомогою спільного факторного аналізу (Joint Factor Analysis, JFA) і розкладання сигналу в векторний ряд Тейлора (Vector Taylor Series, VTS).

Нарешті, ще одним напрямком адаптації є використання ознак, адаптованих за допомогою GMM-HMM моделей. Подібні ознаки, зокрема, адаптовані за допомогою fMLLR, можуть бути успішно використані в гібридних акустичних моделях.

Істотним недоліком алгоритмів з першої і другої груп та методу з використанням дикторських кодів є те, що вони демонструють гарну якість роботи тільки в умовах адаптації з учителем, тобто за наявності еталонного тексту. В реальних задачах ця вимога часто не виконується, і застосовується адаптація без вчителя. Для використання ознак, адаптованих за допомогою GMM-HMM моделей, необхідно виконати попередній прохід розпізнавання, що призводить до значного зниження швидкості роботи системи. Адаптація за допомогою *i*-векторів працює без учителя і не робить істотного впливу на швидкодію, тому можна зробити висновок про перспективність цього підходу для розробки системи розпізнавання російської телефонної мови.

Варто відзначити, що кращі на даний момент гібридні CD-DNN-HMM системи розпізнавання англійської спонтанної мови працюють з fMLLR адаптированими ознаками, доповненими *i*-векторами [18].

1.4.6 Аналіз ефективної методики навчання системи розпізнавання англійської мови

Існуючі системи розпізнавання англійської мови забезпечують дуже високу точність розпізнавання (80-90%). У зв'язку з цим представляється вкрай важливим проведення аналізу основних технологій і методів, що використовуються в цих системах.

На даний момент дослідникам доступна велика кількість інструментів для побудови систем розпізнавання мови, найбільш відомими з яких є HTK Toolkit, Kaldi ASR і CMU Sphinx. Kaldi ASR є найбільш популярним інструментом серед дослідників по деяким причинам:

– підтримує велику частину сучасних методів і алгоритмів розпізнавання мови;

– дає досліднику можливість реалізовувати власні методи і алгоритми;

– забезпечує більш високу точність розпізнавання мови, в порівнянні з іншими інструментами;

– включає в себе готові рецепти для побудови ефективних систем розпізнавання для різних завдань, в тому числі і для розпізнавання мовлення англійською мовою.

До складу інструменту Kaldi ASR входить методика навчання системи розпізнавання мови для бази Switchboard, що демонструє одні з кращих на сьогоднішній день результатів в задачі розпізнавання англійської мови. Цю методику надалі будемо називати рецептом swbd (s5c).

Для мовного моделювання в ній використовується триграмна модель зі словником близько 30000 слів, навчена по текстовим розшифровок записів з корпусу Switchboard і містить близько 750000 n-грам. Процедуру навчання акустичних моделей можна розбити на два етапи. Перший етап – навчання GMM-HMM, складається з наступних основних стадій:

а) Навчання монофонної моделі (mono) з 1000 Гауссіан по 30000 речень. Використовуються 13-мірні MFCC ознаки з CMN, доповнені першими і другими похідними.

б) Навчання першої трифонної моделі (tri1) з 30000 Гауссіан і 3200 пов'язаних станів по 100000 речень. Використовуються 13-мірні MFCC ознаки з CMN, доповнені першими і другими похідними.

в) Навчання другої трифонної моделі (tri2) з 70000 Гауссіан і 4000 пов'язаних станів по 100000 речень. Використовуються 13-мірні MFCC ознаки з CMN, доповнені першими і другими похідними.

г) Навчання третьої трифонної моделі (tri3) з 140000 Гауссіан і 6000 пов'язаних станів за всіма даними. В якості ознак використовуються взяті для 9 сусідніх кадрів (центральний кадр і по 4 кадри зліва і справа) 13-мірні MFCC ознаки з CMN, до яких застосовано LDAMLLT перетворення з пониженням розмірності до 40.

д) Навчання четвертої трифонної моделі (tri4) з 200000 Гауссіан і 11500 пов'язаних станів за всіма даними. В якості ознак використовуються взяті для 9 сусідніх кадрів (центральний кадр і по 4 кадри зліва і справа) 13-мірні MFCC ознаки з CMN, до яких застосовано LDA-MLLT перетворення з пониженням розмірності до 40, адаптовані до диктора за допомогою fMLLR перетворення.

е) Діскрімінативність донавчання четвертої трифонної моделі з використанням bMMI критерію (tri4_mmi_b0,1) за всіма даними.

При цьому для навчання кожної наступної GMM-HMM моделі використовується вирівнювання, отримане за допомогою попередньої. Другий етап – навчання DNN-HMM з 6 прихованими шарами по 2048 нейронів з сигмоїд як функцій активації і вихідним шаром з 8768 нейронів, що відповідають пов'язаним станам моделі tri4. Включає в себе наступні стадії:

а) Перенавчання DNN за допомогою обмежених машин Больцмана.

б) Навчання DNN за критерієм мінімізації взаємної ентропії (dnn5b) з використанням LDA-MLLT-fMLLR ознак від моделі tri4, взятих для 11 сусідніх кадрів (центральний кадр і по 5 кадрів зліва і справа). Використовується розклад

зміни швидкості навчання, аналогічне «newbob», при цьому в якості крос-валідаційної вибірки беруться 10% пропозицій, обраних з навчальних даних випадковим чином.

в) Одна ітерація донавчання моделі dnn5b за критерієм sMBR (dnn5b_smbr).

г) Чотири ітерації донавчання моделі dnn5b_smbr за критерієм sMBR (dnn5b_smbr_i1lats).

У таблиці 1.2 наведені результати, показані акустичними моделями, навченими за цією методикою на повній тестовій базі HUB5 Eval 2000 (другий стовпець) і на підвибірки Switchboard тестової бази HUB5 Eval 2000 (третій стовпець).

Таблиця 1.2 – Результати, показані моделями, навченими за рецептом swbd (s5c) з інструменту Kaldi ASR для бази Switchboard, на повній тестовій базі HUB5 Eval 2000 (другий стовпець) і на підвибірки Switchboard тестової бази HUB5 Eval 2000 (третій стовпець)

Акустична модель	WER, % (FULL)	WER, % (SWBD)
tri1	44,0	36,1
tri2	40,6	32,3
tri3	34,2	26,2
tri4	28,6	21,3
tri4_mmi_b0,1	26,4	19,5
dnn5b	20,4	14,6
dnn5b_smbr	19,3	13,3
dnn5b_smbr_i1lats	18,8	12,9

За цими результатами можна зробити наступні основні висновки:

а) Методи нормалізації (LDA-MLLT) і адаптації (fMLLR) істотно поліпшують точність розпізнавання англійської мови при використанні GMM-HMM акустичних моделей.

б) DNN-HMM акустичні моделі демонструють помітну перевагу в порівнянні з GMM-HMM в задачі розпізнавання англійської мови. [19]

2 МЕТОДИ ПОБУДОВИ ІНФОРМАЦІЙНИХ ОЗНАК І АКУСТИЧНИХ МОДЕЛЕЙ НА ОСНОВІ ГЛИБОКИХ НЕЙРОННИХ МЕРЕЖ

2.1 Інтерпретація глибокої нейронної мережі як каскаду нелінійних перетворень ознак

Результати великої кількості досліджень, проведених в останні роки, говорять про те, що глибокі нейронні мережі демонструють велику перевагу над моделями гауссових сумішей в задачах розпізнавання злитого мовлення, і, зокрема, в задачі розпізнавання англійської мови.

Дослідники з Microsoft Research вважають цю перевагу результатом здатності глибокої нейронної мережі витягати з ознак внутрішні уявлення, стійкі до багатьох джерел варіативності мовного сигналу і володіють високою дискрімінативної здатністю (тобто здатністю добре розділяти акустичні класи). Відповідно до цієї точки зору, глибоку нейронну мережу можна інтерпретувати як складову модель, яка поєднувала каскад нелінійних перетворень вхідних ознак і логлінійний класифікатор (див. рис. 2.1) [20].

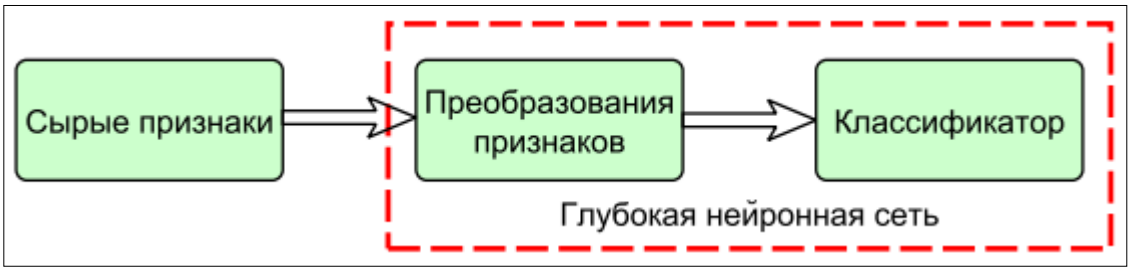


Рисунок 2.1 – Глибока нейронна мережа як складова модель, що складається з перетворень ознак і класифікатора

Комбінація прихованих шарів глибокої нейронної мережі може розглядатися як вчений модуль вилучення ознак. Незважаючи на те, що кожен прихований шар зазвичай реалізує просте нелінійне перетворення, композиція таких простих перетворень може описувати дуже складні закономірності.

Вихідний softmax-шар є простим логлінійним класифікатором, іноді званим моделлю максимальної ентропії (Maximum Entropy, MaxEnt). Тоді для глибокої нейронної мережі з L-1 прихованими шарами обчислення апостеріорної ймовірності $P(s|x)$ може трактуватися як двоетапний процес. На першому етапі вхідний вектор ознак x трансформується в вектор v^{L-1} за допомогою $L-1$ нелінійних перетворень, здійснюваних на прихованих шарах глибокої нейронної мережі. На другому етапі відбувається обчислення апостеріорної ймовірності $P(s|v^{L-1})$ за допомогою логлінійної моделі. Таким чином, приховані шари глибокої нейронної мережі витягують з «сирих» вхідних ознак внутрішні уявлення, які ефективно класифікуються за допомогою логлінійної моделі на вихідному шарі. При цьому навчання класифікатора і перетворень ознак відбувається одночасно.

На прихованих шарах глибокої нейронної мережі, близьких до вхідного шару, витягуються низькорівневі ознаки. Низькорівневі ознаки зазвичай визначають локальні шаблони, вельми чутливі до незначних змін вхідних ознак. З іншого боку, високорівневі ознаки, які добуваються на прихованих шарах глибокої нейронної мережі, близьких до вихідного прошарку, і, по суті, побудовані на низькорівневих ознаках, є більш абстрактними і інваріантними до малих змін вхідних ознак. Щоб переконатися в цьому, розглянемо глибоку нейронну мережу з $L-1$ прихованими шарами і сигмоїдами як функцій активації.

Припустимо, що до вхідного вектору ознак $x = v^0$ додалося мале обурення δ^0. Тоді значення активації $v^l = \sigma(W^l v^{l-1} + b^l)$ для l-го прихованого шару ($l = 1,2,\ldots,L-1$) зміниться на величину

$$\delta^l = \sigma(\mathbf{W}^l(\mathbf{v}^{l-1} + \delta^{l-1}) + \mathbf{b}^l) - \sigma(\mathbf{W}^l\mathbf{v}^{l-1} + \mathbf{b}^l) \approx \approx \mathrm{diag}(\sigma'(\mathbf{W}^l\mathbf{v}^{l-1} + \mathbf{b}^l))(\mathbf{W}^l)^T\delta^{l-1}, \quad (2.1)$$

де під $\sigma(z)$ розуміється вектор, кожна компонента якого є сигмоїда (див. формулу 1.10) від відповідної компоненти вектора z, під diag(z) – матриця, у якій на діагоналі стоять компоненти вектора z, а всі інші елементи дорівнюють нулю. Неважко бачити, що

$$\sigma'(\mathbf{W}^l\mathbf{v}^{l-1}+\mathbf{b}^l))=\sigma(\mathbf{W}^l\mathbf{v}^{l-1}+\mathbf{b}^l)\circ(1-\sigma(\mathbf{W}^l\mathbf{v}^{l-1}+\mathbf{b}^l))=\mathbf{v}^l\circ(1-\mathbf{v}^l), \quad (2.2)$$

де символом ◦ позначено поелементне перемноження двох векторів. Тоді зміни $\delta^l(l = 1,2,\ldots,L\text{-}1)$ можна оцінити так

$$\begin{aligned}\|\delta^l\| \approx \|\mathrm{diag}(\sigma'(\mathbf{W}^l\mathbf{v}^{l-1}+\mathbf{b}^l))(\mathbf{W}^l)^T\delta^{l-1}\| \leqslant \\ \leqslant \|\mathrm{diag}(\sigma'(\mathbf{W}^l\mathbf{v}^{l-1}+\mathbf{b}^l))(\mathbf{W}^l)^T\|\|\delta^{l-1}\| = \\ = \|\mathrm{diag}(\mathbf{v}^l\circ(1-\mathbf{v}^l))(\mathbf{W}^l)^T\|\|\delta^{l-1}\|\end{aligned} \quad (2.3)$$

У глибоких нейронних мережах з досить великими розмірами прихованих шарів величини більшості елементів матриць ваг зазвичай малі (див. рис. 2.2).

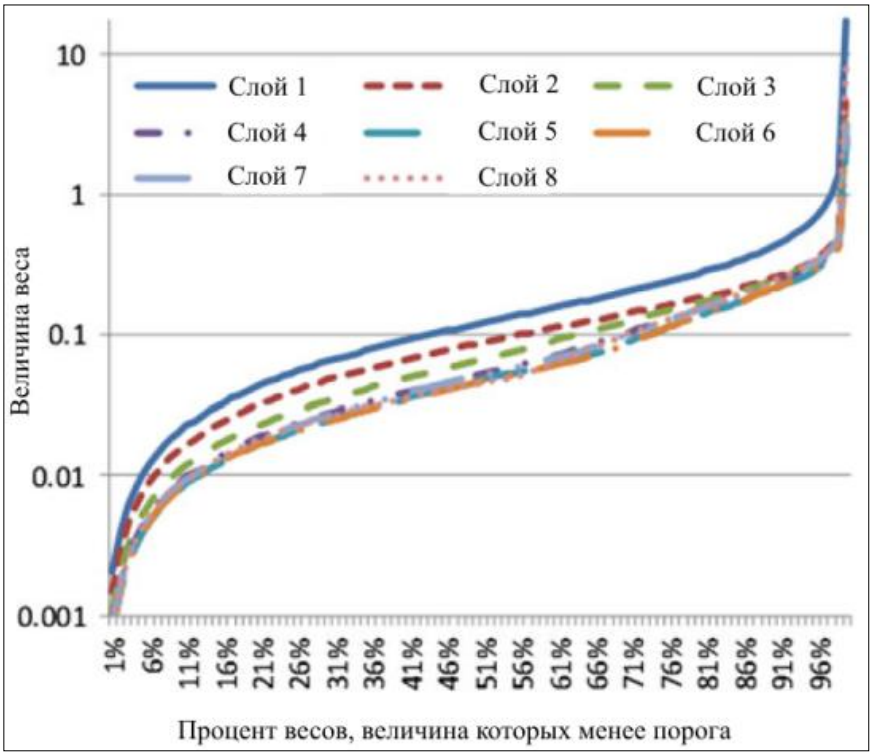


Рисунок 2.2 – Розподіл велечин ваг в типічній DNN

Наприклад, в DNN з 6 шарами по 2000 нейронів, навченої по 30 годинам англійської мови з корпусу Switchboard, величини 98% ваг у всіх шарах, за винятком вхідного, виявилися менш ніж 0,5. Величина кожного компонента

вектора $v^l \circ (1- v^l)$ не може перевищувати 0,25. В реальності це значення набагато нижче, оскільки великий відсоток нейронів є неактивними (тобто значення їх активації близькі до нуля або одиниці) (див. рис. 2.3).

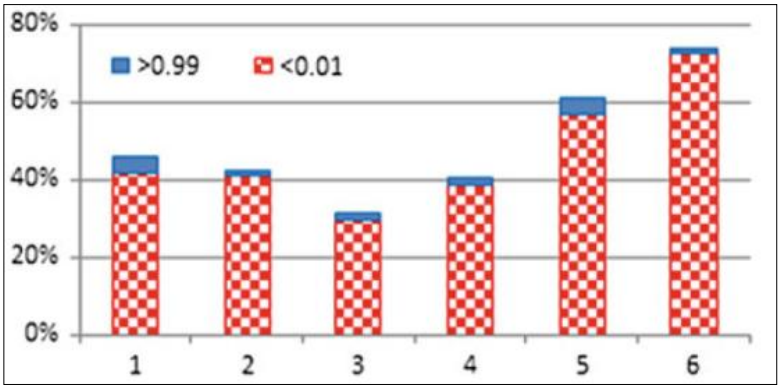


Рисунок 2.3 – Відсоток неактивних нейронів на кожному шарі DNN

За рахунок цього середнє значення, яке приймається нормою $\|\text{diag}\,(v^l \circ (1- v^l))\,(W^l)^T\|$, виявляється меншим, ніж одиниця (див. рис. 2.4).

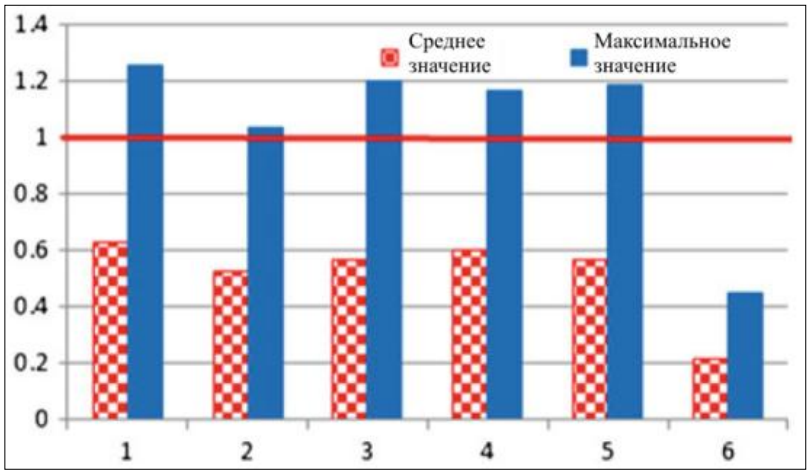


Рисунок 2.4 – Середнє і забезпечити максимальне значення, на кожному шарі для DNN з 6 шарами по 2000 нейронів

Таким чином, мала похибка у вхідних даних буде зменшуватися з кожним прихованим шаром. За рахунок цього внутрішні уявлення, які добуваються

прихованими шарами глибокої нейронної мережі з вхідних ознак, стають менш чутливими до малих похибок вхідного сигналу з ростом числа прихованих шарів. Однак це працює тільки для малих похибок, тому для ефективного навчання DNN необхідно, щоб навчальні дані були в достатній мірі близькі до реальних даних, на яких ця DNN буде експлуатуватися.

Після проходження каскаду нелінійних перетворень в прихованих шарах глибокої нейронної мережі ознаки стають більш стійкими по відношенню до міждикторської варіативності, канальної варіативності, варіацій темпу мови і акустичної обстановки. Це дозволяє системам розпізнавання мови, заснованим на DNN, навіть без використання алгоритмів нормалізації ознак і адаптації моделі перевершувати GMM-HMM системи [21].

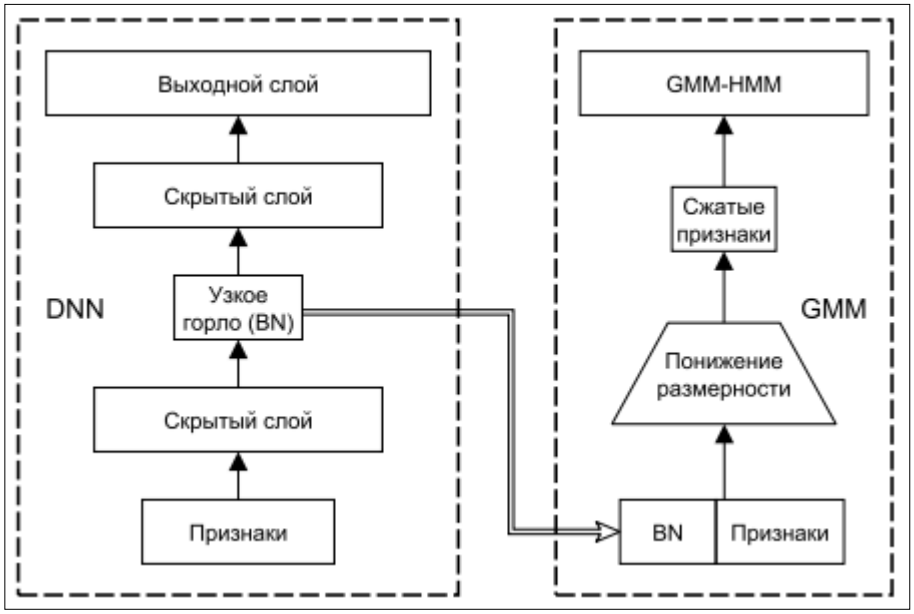
2.1.1 Ознаки, які добувають із нейронної мережі з вузьким горлом

З огляду на сказане в розділі 2.1, очевидним способом отримання стійких по відношенню до акустичної варіативності мовного сигналу, які володіють високою дискрімінативною здатністю, ознак для навчання акустичних моделей є використання в якості ознак вектори активацій одного з прихованих або вихідного шарів. Вперше ця ідея, названа тандемним підходом (Tandem approach), була запропонована в роботі Хермaнського. Перетворені різними способами ймовірності фонем, які генеруються на вихідному шарі нейронної мережі з одним прихованим шаром, використовувалися як вектор ознак для навчання GMM-HMM акустичної моделі.

При використанні в якості вектора ознак вектора активацій одного з останніх прихованих шарів, або вихідного шару глибокої нейронної мережі, навченої класифікувати зв'язані стани трифонів, виникають проблеми через занадто великої розмірності отриманого таким чином вектора. Альтернативною технікою, запропонованої в роботах дослідників з технічного університету Брно

(Чехія), є отримання компактних ознак з так названого вузького горла (bottleneck) – малорозмірного прихованого шару зазвичай з лінійною функцією активації, розташованого в середині або ближче до останніх прихованим верствам глибокої нейронної мережі. Ознаки, які добувають із глибокої нейронної мережі з вузьким горлом, також називають bottleneck-ознаками.

У більшості робіт ознаки, витягнуті з нейронної мережі з вузьким горлом, об'єднуються з простими ознаками (наприклад, MFCC) і після перетворення, яке здійснює зниження розмірності і декореляції, використовуються надалі для навчання GMM-HMM акустичних моделей. Як перетворення зазвичай використовують метод головних компонент (Principal Component Analysis, PCA) або гетероскедастичний лінійний дискримінантний аналіз (Heteroscedastic Linear Discriminant Analysis, HLDA). Схема такого підходу зображена на рисунку 2.5.



Рисунко 2.5 – Схема навчання GMM-HMM акустичної моделі з використанням ознак, які витягнуті з нейронної мережі з вузьким горлом

Це дозволяє використовувати численні техніки, розроблені для поліпшення GMM-HMM моделей, такі як адаптація і дискрімінативність навчання, і таким чином досягти високої якості роботи системи розпізнавання. Іноді для навчання

GMM-HMM моделі використовуються bottleneck-ознаки другого рівня: на ознаках, витягнутих з глибокої нейронної мережі з вузьким горлом і взятих з деяким контекстом, навчається ще одна глибока нейронна мережа з вузьким горлом (див. рис. 2.6).

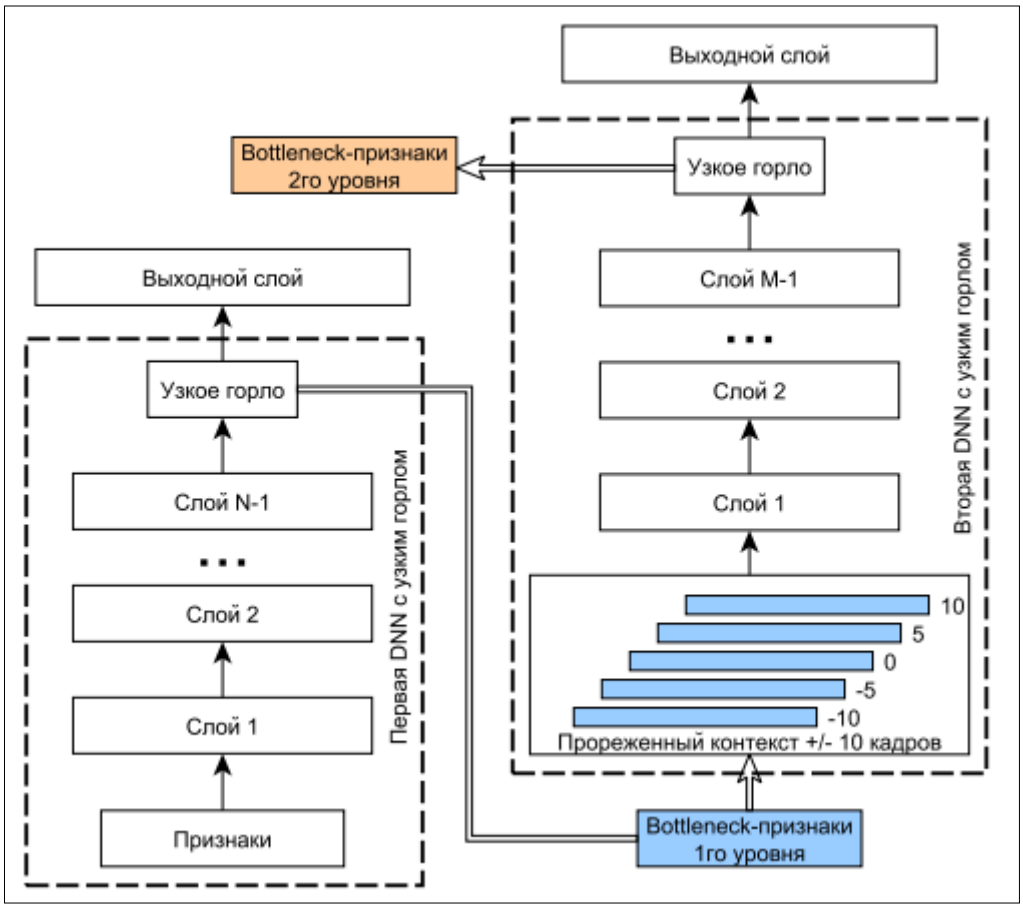


Рисунок 2.6 – Схема витягвання bottleneck-ознак другого рівня

Також дуже популярно використання ознак, що отримуються з глибокої нейронної мережі, навченої за великим обсягом даних однієї мови, для створення системи розпізнавання будь-якого іншого мови, для якого немає великих навчальних баз. Використання таких ознак дозволяє значно підвищити якість роботи системи розпізнавання. Так, в роботі зроблено спробу використовувати такі ознаки, нейронна мережа для отримання яких була навчена на базі англійської мови, для побудови системи розпізнавання російської мови. Успішність цього підходу свідчить про те, що bottleneck-ознаки забезпечують

стійкість системи розпізнавання мови до акустичної варіативності мовного сигналу.

За результатами аналізу вищенаведених досліджень, присвячених використанню в розпізнаванні мови ознак, які з глибокої нейронної мережі з вузьким горлом, були зроблені наступні висновки:

– в якості міток для навчання глибокої нейронної мережі з вузьким горлом краще використовувати связані стани трифонів, ніж моно фони;

– вузьке горло краще розміщувати ближче до вихідного шару, ніж до вхідного;

– у більшості досліджень використовується від 30 до 100 нейронів у вузькому шарі;

– якість роботи моделей, навчених з використанням bottleneck ознак, залежить від точності розпізнавання, яка забезпечується глибокої нейронною мережею з вузьким горлом, з якої ці ознаки були витягнуті;

– bottleneck-ознаки, витягнуті з нейронної мережі, навченої для однієї мови, забезпечують високі результати і на інших мовах [22].

2.2 Метод побудови інформаційних ознак, які з адаптованої до диктора і акустичних умов глибокої нейронної мережі з вузьким горлом

Ідея запропонованого методу полягає у використанні для вилучення ознак адаптованої глибокої нейронної мережі. Основою для цієї ідеї послужив наступний висновок: чим краще точність розпізнавання, яка забезпечується глибокої нейронною мережею з вузьким горлом, тим кращу точність розпізнавання буде забезпечувати система, побудована на основі ознак, витягнутих з цієї нейронної мережі.

Аналіз алгоритмів адаптації глибоких нейронних мереж показав, що адаптація глибоких нейронних мереж з використанням *i*-векторів, запропонована

дослідниками з IBM, значно підвищує точність розпізнавання за рахунок надання глибокої нейронної мережі додаткової інформації про фонограму. Таким чином, в основі запропонованого методу лежить припущення, що ознаки, які добувають із глибокої нейронної мережі з вузьким горлом, адаптованої за допомогою *i*-векторів, будуть володіти більшою стійкістю по відношенню до акустичної варіативності і кращої дискрімінативною здатністю, ніж аналогічні ознаки, витягнуті з адаптованою нейронної мережі.

Повний алгоритм побудови ознак, згідно із запропонованим методом, складається з наступних кроків:

– побудова кепстральних ознак (наприклад, MFCC) для навчання GMM-HMM моделі;

– навчання трифонів GMM-HMM моделі;

– формування розмітки навчальних даних на зв'язані стани трифонів за допомогою GMM-HMM моделі;

– побудова ознак для навчання глибокої нейронної мережі (ці ознаки можуть відрізнятися від використовуваних при навчання GMM-HMM моделі);

– приведення вхідних даних для навчання глибокої нейронної мережі до нульового середнього та одиничної дисперсії;

– ініціалізація навчання глибокої нейронної мережі з L прихованими шарами одним із способів;

– навчання глибокої нейронної мережі за критерієм мінімізації взаємної ентропії (див. рис. 2.7а);

– побудова *i*-векторів для навчальної бази;

– приведення побудованих *i*-векторів до нульового середнього та одиничної дисперсії, або нормалізація будь-яким іншим способом;

– розширення вхідного шару навченої глибокої нейронної мережі з ініціалізацією відповідних коефіцієнтів матриці ваг нульовими значеннями;

– донавчання глибокої нейронної мережі з розширеним вхідним шаром по ознаками, до яких на кожному кадрі доданий *i*-вектор, який відповідає цьому ділянці фонограми (див. рис. 2.7б). При цьому використовується менша

швидкість навчання, а до цільової функції додано доданок $R(W)$, штраф відхилення ваг W^l навченою моделі від значень ваг $\tilde{W}^l$ вихідної моделі, що визначається за формулою

$$R(\mathbf{W}) = \lambda \sum_{l=1}^{L+1} \|\mathrm{vec}(\mathbf{W}^l - \tilde{\mathbf{W}}^l)\|_2 = \lambda \sum_{l=1}^{L+1} \sum_{i=1}^{N_l} \sum_{j=1}^{N_{l-1}} (\mathbf{W}_{ij}^l - \tilde{\mathbf{W}}_{ij}^l)^2, \qquad (2.4)$$

де під vec(W) розуміється вектор, отриманий в результаті об'єднання всіх стовпців матриці W, а λ – величина штрафу, зазвичай обирається в діапазоні між 10^{-8} і 10^{-6};

– розбиття шару l глибокої нейронної мережі (наприклад, останнього прихованого шару) на два шару наступним чином:

$$\mathbf{v}^l = f(\mathbf{W}^l \mathbf{v}^{l-1} + \mathbf{b}^l) \approx f(\mathbf{W}_{out}^l (\mathbf{W}_{bn}^l \mathbf{v}^{l-1} + \mathbf{0}) + \mathbf{b}^l). \qquad (2.5)$$

Тут перший шар – малорозмірний шар з лінійною функцією активації, матрицею ваг W^l_{bn} і нульовим вектором зсувів; другий шар – нелінійний шар з матрицею ваг W^l_{out} і вектором зсувів b^l, що має розмірність вихідного разбиваемого шару. Розбиття здійснюється за допомогою сингулярного розкладання (Singular Values Decomposition, SVD) матриці ваг W^l

$$\mathbf{W}^l = USV^T \approx \tilde{U}_{bn} \tilde{V}_{bn}^T = \mathbf{W}_{out}^l \mathbf{W}_{bn}^l, \qquad (2.6)$$

де нижній індекс $_{bn}$ означає знижену розмірність. Таким чином, вихідна глибока нейронна мережа з L прихованими шарами перетворюється у глибоку нейронну мережу з (L + 1) прихованими шарами з лінійним тонким шаром l. Додавання вузького прошарку перед вихідним шаром DNN, дозволило зменшити в кілька разів число параметрів акустичної моделі без погіршення якості її роботи;

– донавчання отриманої глибокої нейронної мережі з вузьким горлом (див. рис. 2.7в) з меншою швидкістю і штрафом на відхилення ваг від ваг вихідної моделі;

– відкидання шарів глибокої нейронної мережі, що настають за вузьким горлом;

– використання отриманої нейронної мережі з вузьким горлом для побудови високорівневих ознак.

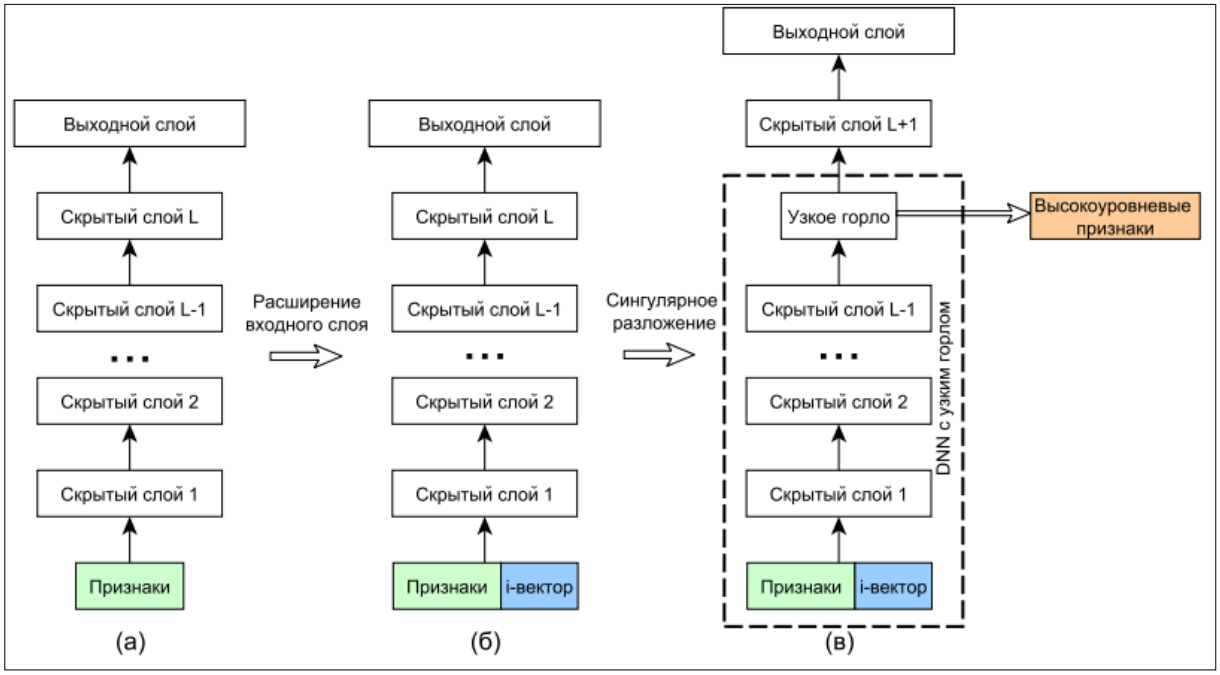


Рисунок 2.7 – Основні етапи навчання глибокої нейронної мережі з вузьким горлом, адаптованої за допомогою *i*-векторів: навчання не адаптованою глибокої нейронної мережі (а), навчання адаптованої глибокої нейронної мережі (б), навчання адаптованої глибокої нейронної мережі з вузьким горлом (в)

Оскільки якість розмітки, що генерується GMM-HMM моделлю, виявляє помітний вплив на навчання DNN, для досягнення кращих результатів має сенс повторити кроки алгоритму, починаючи з другого, використовуючи для навчання GMM-HMM ознаки, побудовані за допомогою глибокої нейронної мережі з вузьким горлом. Схема запропонованого алгоритму побудови ознак представлена на рисунку 2.8.

Ознаки, побудовані за цим алгоритмом, в подальшому можуть використовуватися як для навчання GMM-HMM моделей, так і для навчання DNN-HMM моделей.

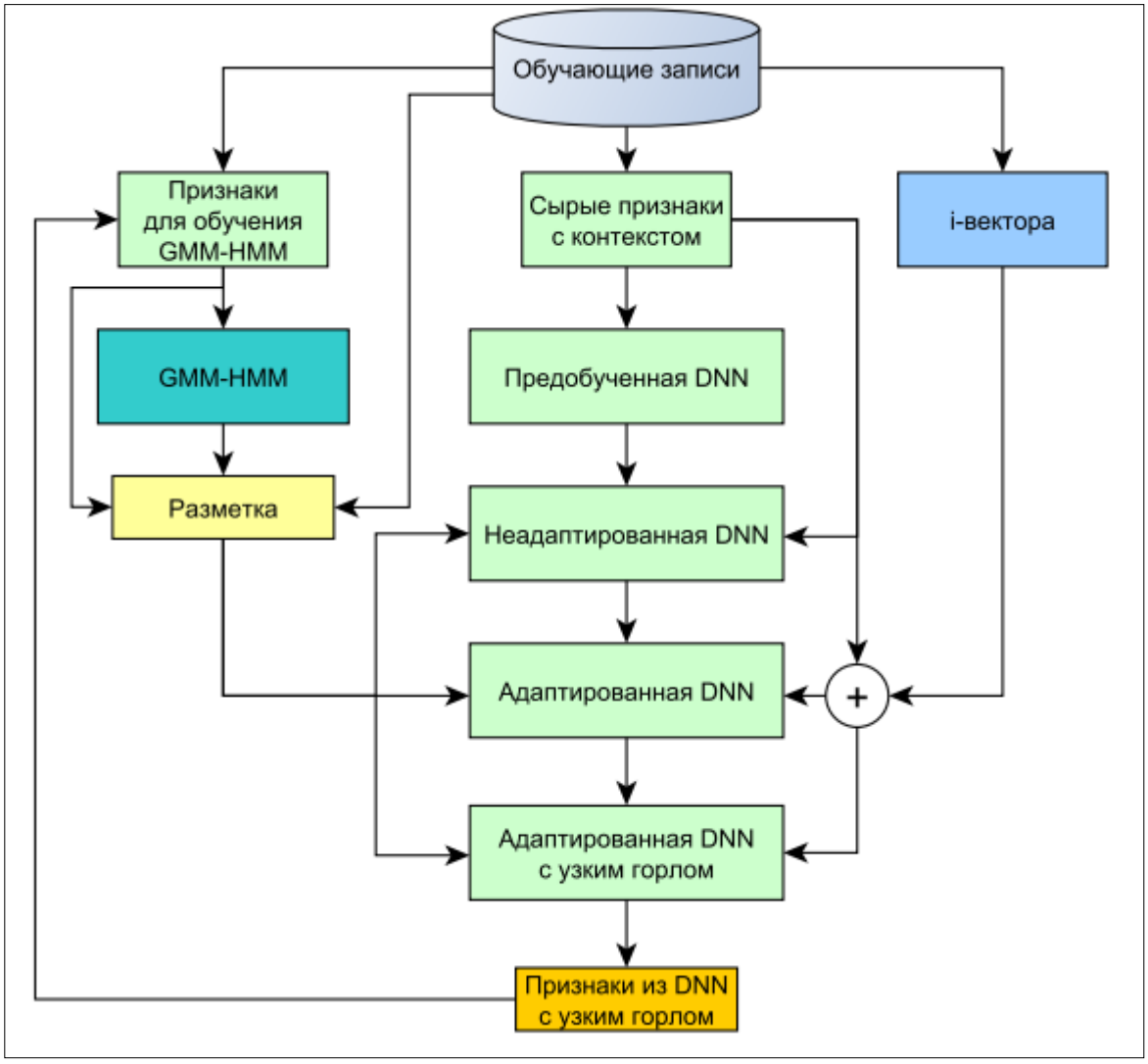


Рисунок 2.8 – Схема алгоритму побудови ознак за допомогою адаптованої з використанням *i*-векторів глибокої нейронної мережі з вузьким горлом

Пропонується наступний алгоритм навчання акустичних моделей на основі ознак, побудованих відповідно до запропонованого методу:

– навчання трифонів GMM-HMM моделі на основі побудованих ознак;

– розмітка навчальних даних на зв'язані стани трифонів за допомогою побудованої трифонів GMM-HMM моделі;

– навчання DNN-HMM моделі з використанням побудованих ознак, взятих з широким тимчасовим контекстом (наприклад, в 31 кадр). Одним з переваг такого навчання глибоких нейронних є можливість ефективно враховувати більш широкий тимчасовий контекст, в порівнянні з глибокими нейронними мережами, навченими на «сирих» ознаках [23].

2.2.1 Експерименти по оцінці ефективності запропонованого методу побудови ознак в задачі розпізнавання англійської мови

У цьому розділі описані експерименти, проведені для оцінки ефективності роботи розробленого методу побудови ознак в задачі розпізнавання англійської мови. Для навчання акустичних моделей використовувалася база Switchboard, для оцінки результатів – підвибірка Switchboard тестової бази HUB5 Eval 2000. В якості основи для експериментів був використаний рецепт swbd (s5c) з інструменту Kaldi ASR [24].

Для побудови *i*-векторів використовувалася система на основі UBM (Universal Background Model) з 512 гауссіанами, навчена на 13-мірних MFCC ознак, доповнених першими і другими похідними. З її допомогою витягувалися *i*-вектори розмірності 100 для навчальних і тестових записів.

2.2.1.1 Навчання на fMLLR-адаптованих ознаках

У цій серії експериментів в якості базової моделі була взята глибока нейронна мережа dnn5b з 6 прихованими шарами по 2048 нейрона і сигмоїдами як функцій активації з рецепта swbd (s5c), навчена на fMLLR-адаптованих за допомогою трифонів GMM-HMM моделі tri4 40-мірних ознаках, взятих з

тимчасовим контекстом в 11 кадрів (центральний кадр і по 5 кадрів зліва і справа).

Адаптована модель dnn5b_iv була навчена на вхідних ознаках базової моделі, доповнених *i*-вектором розмірності 100. Навчання ініціалізувалося базовою моделлю з розширеним вхідним шаром, при цьому використовувалася швидкість навчання 0,002 і штраф $4 \cdot 10^{-8}$ на відхилення ваг від значень базової моделі.

У адаптовану модель за допомогою сингулярного розкладання матриці ваг 6-го прихованого шару був доданий лінійний шар розмірності 80. Отримана таким чином нейронна мережа з вузьким горлом використовувалася для ініціалізації навчання моделі dnn5b_iv_bn6-80, при цьому навчання проводилося з швидкістю 0,002 і штрафом $4 \cdot 10^{-8}$ на відхилення ваг від значень ініціалізуючьої нейронної мережі. Ця модель після видалення останнього прихованого шару і вихідного шару використовувалася для побудови 80-мірних ознак (SDBN).

На SDBN ознаках була навчена трифонна GMM-HMM модель tri_sdbn з тими ж числом Гауссіан (200000) та пов'язаних станів (11500), що і базова трифонна модель tri4. Також SDBN ознаки, взяті з контекстом в 31 кадр, зрідженим за часом через 5 кадрів (тобто [-15 -10 -5 0 5 10 15]), були використані для навчання DNN-HMM моделі dnn_sdbn з 4 прихованими шарами по 2048 нейронів з сигмоїдами, з ініціалізацією навчання за допомогою обмежених машин Больцмана. Для навчання всіх перерахованих вище моделей використовувалася розмітка на зв'язані стани трифонів, зроблена за допомогою базової GMM-HMM моделі tri4.

DNN-HMM модель dnn_sdbn_smbr_i1lats була навчена по sMBR-критерієм поділу послідовностей, за схемою навчання, аналогічної моделі dnn5b_smbr_i1lats з рецепта swbd (s5c).

Нарешті, DNN-HMM модель dnn_sdbn_sdbn-ali_smbr_i1lats була навчена аналогічним чином з використанням розмітки на зв'язані стани трифонів, зробленої за допомогою GMM-HMM моделі tri_sdbn. Результати експериментів, наведені в таблиці 2.1, говорять про наступне:

а) DNN-HMM модель dnn5b_iv, адаптована до диктора і акустичної обстановки за допомогою *i*-векторів, продемонструвала 0,5% абсолютне і 3,4% відносне зменшення помилки розпізнавання, в порівнянні з базовою DNN-HMM моделлю.

б) Адаптована за допомогою *i*-векторів DNN-HMM модель з вузьким горлом dnn5b_iv_bn6-80 дала очікуване погіршення в порівнянні з моделлю dnn5b_iv, але тим не менше виявилася кращою базовою DNN-HMM моделі на 0,3% абсолютних і 2,1% відносних.

в) Трифонна GMM-HMM модель на SDBN ознаках tri_sdbn показала 5,7% абсолютне і 26,8% відносне зменшення помилки розпізнавання, в порівнянні з базовою трифонною GMM-HMM моделлю.

Таблиця 2.1 – Результати, які демонструються моделями, навченими на ознаках,побудованих за допомогою запропонованого методу, на підвибірки Switchboard тестової бази HUB5 Eval 2000.

Акустична модель	WER, %	ΔWER, %	WERR, %
tri4	21,3	–	–
tri_sdbn	15,6	5,7	26,8
dnn5b	14,6	–	–
dnn5b_iv	14,1	0,5	3,4
dnn5b_iv_bn6-80	14,3	0,3	2,1
dnn_sdbn	13,6	1,0	6,8
dnn5b_smbr_i1lats	12,9	–	–
dnn_sdbn_smbr_i1lats	12,4	0,5	3,9
dnn_sdbn_sdbn-ali_smbr_i1lats	12,1	0,5	6,2

г) DNN-HMM модель dnn_sdbn продемонструвала 1,0% абсолютне і 6,8% відносне зменшення помилки розпізнавання, в порівнянні з базовою DNN-HMM моделлю, а також 0,5% абсолютне і 3,5% відносне зменшення помилки розпізнавання, в порівнянні з адаптованої DNN-HMM моделлю dnn5b_iv.

д) DNN-HMM модель dnn_sdbn_smbr_i1lats, навчена з використанням критерію sMBR, виявилася на 0,5% абсолютних і 3,9% відносних краще, ніж базова модель dnn5b_smbr_i1lats, навчена за тим же критерієм.

е) DNN-HMM модель dnn_sdbn_sdbn-ali_smbr_i1lats, навчена з використанням критерію sMBR і розмітки від GMM-HMM моделі tri_sdbn, виявилася на 0,9% абсолютних і 6,2% відносних краще, ніж базова модель dnn5b_smbr_i1lats, навчена за тим же критерієм.

Результати дозволяють зробити висновок про високу ефективність розробленого методу в задачі розпізнавання англійської спонтанної мови.

2.2.1.2 Навчання на сирих ознаках без використання fMLLR-адаптації

У наведених вище експериментах навчання проводилося на вже адаптованих до диктора за допомогою fMLLR-перетворення ознаках. Однак використання *i*-векторів здійснює адаптацію до диктора, як і fMLLR адаптація. Використання fMLLR-адаптованих ознак зменшує приріст, який забезпечувався б за рахунок застосування адаптації за допомогою *i*-векторів. Отже, є підстави очікувати, що без використання fMLLR-адаптації розроблений метод побудови ознак продемонструє ще більшу ефективність [25].

Для оцінки роботи запропонованого методу побудови ознак в умовах відсутності fMLLR-адаптації була проведена ще одна серія експериментів. При цьому для навчання моделей використовувалася та ж розмітка на зв'язані стани трифонів, зроблена за допомогою моделі tri4, а в якості ознак для навчання глибоких нейронних мереж були використані сирі спектральні ознаки – логарифми енергій сигналу в 23-х трикутних Мел-частотних фільтрах (FBANK), доповнені першими і другими похідними і взяті з тимчасовим контекстом в 11 кадрів (центральний кадр і по 5 кадрів зліва і справа). Аналогічним чином, що і в експериментах на fMLLR-ознаках, були навчені наступні моделі:

– базова DNN-HMM модель dnn-fbank з 6 прихованими шарами по 2048 нейронів з сигмоїдами;

– DNN-HMM модель dnn-fbank_iv, адаптована за допомогою *i*-векторів (використовувалися ті ж 100-мірні *i*-вектори, що і в експериментах на fMLLR-ознаках);

– адаптована DNN-HMM модель dnn-fbank_iv_bn6-80 з лінійним тонким шаром розмірності 80, розташованим перед останнім прихованим шаром. Далі глибока нейронна мережа з вузьким горлом dnn-fbank_iv_bn6-80 була використана для побудови 80-мірних ознак (SDBN-FBANK), на яких були навчені трифонна GMM-HMM модель tri_sdbn-fbank і DNN-HMM модель dnn_sdbn-fbank з 4 прихованими шарами по 2048 нейронів з сигмоїдами як функцій активації (як і в попередньому експерименті, ознаки для цієї DNN-HMM моделі бралися з тимчасовим контекстом в 31 кадр, зрідженим через 5 кадрів). У таблиці 2.2 представлені результати, отримані в ході цих експериментів.

Таблиця 2.2 – Результати, які демонструються моделями, навченими на ознаках, побудованих за допомогою запропонованого методу без використання fMLLR-адаптації, на підвибірки Switchboard тестової бази HUB5 Eval 2000.

Акустична модель	WER, %	ΔWER, %	WERR, %
tri4	21,3	–	–
tri_sdbn-fbank	16,3	5,0	23,5
dnn-fbank	16,4	–	–
dnn-fbank_iv	14,9	1,5	9,1
dnn-fbank_iv_bn6-80	14,9	1,5	9,1
dnn_sdbn-fbank	14,2	0,7	13,4

Результати свідчать про наступне:

а) DNN-HMM модель dnn-fbank_iv, адаптована до диктора і акустичної обстановці за допомогою і-векторів, продемонструвала 1,5% абсолютне і 9,1% відносне зменшення помилки розпізнавання, у порівнянні з базовою DNN-HMM моделлю dnn-f bank.

б) Адаптована за допомогою *i*-векторів DNN-HMM модель з вузьким горлом dnn-fbank_iv_bn6-80 продемонструвала такі ж результати, як і адаптована DNN-HMM модель dnn-fbank_iv.

в) Трифонна GMM-HMM модель tri_sdbn-fbank на SDBN-FBANK ознаках показала 5,0% абсолютне і 23,5% відносне зменшення помилки розпізнавання, в порівнянні з базовою трифонною GMM-HMM моделлю.

г) DNN-HMM модель dnn_sdbn-fbank продемонструвала 2,2% абсолютне і 13,4% відносне зменшення помилки розпізнавання, в порівнянні з базовою DNN-HMM моделлю, а також 0,7% абсолютне і 4,7% відносне зменшення помилки розпізнавання, в порівнянні з адаптованої DNN-HMM моделлю dnn-fbank_iv.

2.3 Двоетапний алгоритм ініціалізації навчання акустичних моделей на основі глибоких нейронних мереж

Для експериментальної оцінки ефективності запропонованого двоетапного алгоритму ініціалізації навчання з його допомогою були навчені глибокі нейронні мережі в наступних конфігураціях:

а) fMLLR-адаптовані за допомогою трифонів GMM-HMM моделі tri4 40-мірних ознаки, взяті з тимчасовим контекстом в 11 кадрів (центральний кадр і по 5 кадрів зліва і справа); 6 прихованих шарів по 2048 нейронів з сигмоїдами.

б) Логарифми енергій сигналу в 23-х трикутних Мел-частотних фільтрах (FBANK), доповнені першими і другими похідними і взяті з тимчасовим контекстом в 11 кадрів (центральний кадр і по 5 кадрів зліва і справа); 6 прихованих шарів по 2048 нейронів з сигмоїдами.

в) Побудовані в розділі 2.2.1 80-мірні ознаки SDBN, отримані з адаптованої за допомогою *i*-векторів глибокої нейронної мережі і взяті з контекстом в 31 кадр, зрідженим за часом через 5 кадрів.

Як і раніше, використовувалася розмітка на зв'язані стани від трифонної GMM-HMM моделі tri4. На першому етапі виконувалося пренавчання з використанням обмежених машин Больцмана, на другому етапі – навчання за критерієм мінімізації взаємної ентропії зі швидкістю 0,008 за навчальною вибіркою, з якої було випадковим чином викинуто 98% прикладів, відповідних паузі. Отримані таким чином глибокі нейронні мережі для fMLLR, FBANK і SDBN-конфігурацій були використані для ініціалізації донавчання по повній навчальній вибірці. При цьому швидкість навчання була зменшена до 0,0004, використовувався прискорений градієнт Нестерова з показником 0,7, а також використовувався штраф $4 \cdot 10^{-8}$ на відхилення ваг від значень ініціалізуючої глибокої нейронної мережі.

В результаті були отримані DNN-HMM моделі dnn5b_2step, dnnfbank_2step і dnn_sdbn_2step для fMLLR, FBANK і SDBN-конфігурацій відповідно. Результати, які демонструються цими моделями, наведені в таблиці 2.3.

Таблиця 2.3 – Результати, які демонструються моделями, навченими з використанням двоетапного алгоритму ініціалізації, на підвибірки Switchboard тестової бази HUB5 Eval 2000.

Акустична модель	WER, %	ΔWER, %	WERR, %
dnn5b	14,6	–	–
dnn5b_2step	14,5	0,1	0,7
dnn-fbank	16,4	–	–
dnn-fbank_2step	15,9	0,5	3,0
dnn_sdbn	13,6	–	–
dnn_sdbn_2step	13,5	0,1	0,7

За цими даними можна зробити висновок про перевагу запропонованого двоетапного алгоритму ініціалізації навчання над алгоритмом пренавчання за допомогою обмежених машин Больцмана на 0,1-0,5% абсолютних і 0,7-3,0% відносних, в залежності від використовуваної конфігурації. Варто відзначити, що ефективність алгоритму висока для «сирих» неадаптованих ознак (FBANK) і

знижується при переході до більш складним адаптованим ознаками (fMLLR або SDBN).

Метою наступного експерименту було з'ясування того, який із трьох запропонованих в розділі 2.3 способів навчання адаптованої за допомогою *i*-векторів глибокої нейронної мережі з використанням двоетапного алгоритму пренавчання демонструє кращі результати. Експеримент проводився на FBANK-конфігурації. В якості базової моделі була обрана адаптована за допомогою i-векторів DNN-HMM модель dnn-fbank_iv, навчена без застосування запропонованого двоетапного алгоритму ініціалізації.

Таблиця 2.4 – Порівняння трьох способів навчання адаптованих за допомогою *i*-векторів глибоких нейронних мереж з використанням двоетапного алгоритму ініціалізації на підвибірки Switchboard тестової бази HUB5 Eval 2000.

Акустична модель	WER, %	ΔWER, %	WERR, %
dnn-fbank_iv	14,9	–	–
dnn-fbank_sil2_iv	14,7	0,2	1,3
dnn-fbank_sil2_sil100_iv	14,7	0,2	1,3
dnn-fbank_sil2_iv-sil2_sil100	14,7	0,2	1,3

Три адаптовані за допомогою *i*-векторів DNN-HMM моделі були навчені з використанням запропонованого двоетапного алгоритму ініціалізації навчання:

а) Модель dnn-fbank_sil2_iv була навчена за даними з непроріженою паузою. Як ініціалізації використовувалася неадаптована глибока нейронна мережа, навчена з проріджуванням паузи до 2%.

б) Модель dnn-fbank_sil2_sil100_iv була навчена за даними з непроріженою паузою. Для ініціалізації навчання використовувалася неадаптована глибока нейронна мережа dnn-fbank_2step, навчена за даними з непроріженою паузою з використанням двоетапного алгоритму.

в) Модель dnn-fbank_sil2_iv-sil2_sil100 була навчена за даними з непроріженою паузою. Для ініціалізації навчання використовувалася адаптована за допомогою і-векторів глибока нейронна мережа, навчена з проріджуванням паузи до 2%, навчання якої ініціалізувалося адаптованою глибокої нейронною мережею, навченої з проріджуванням паузи до 2%.

Результати, представлені в таблиці 2.4, свідчать про те, що три адаптовані за допомогою *і*-векторів моделі, навчені з використанням двоетапного алгоритму ініціалізації, продемонстрували однакові результати, перевершивши базову модель dnn-fbank_iv на 0,2% абсолютних і 1,3% відносних.

Таким чином було проведено експериментальні дослідження, що підтверджують ефективність запропонованих методу та алгоритму в задачі розпізнавання англійської мови [26].

3 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ

3.1 Архітектура системи

Система має багаторівневу архітектуру, яка складається з таких частин:

– клієнтський додаток, яки зчитує мовні сигнали;

– сервер обробки мовних сигналів;

– сервер розпізнавання команди;

– модуль передачі сигналів на пристрої;

– пристрої, під’єднанні до системи.

Більш детально структуру системи можна побачити на діаграмі компонентів (див. рис. 3.1).

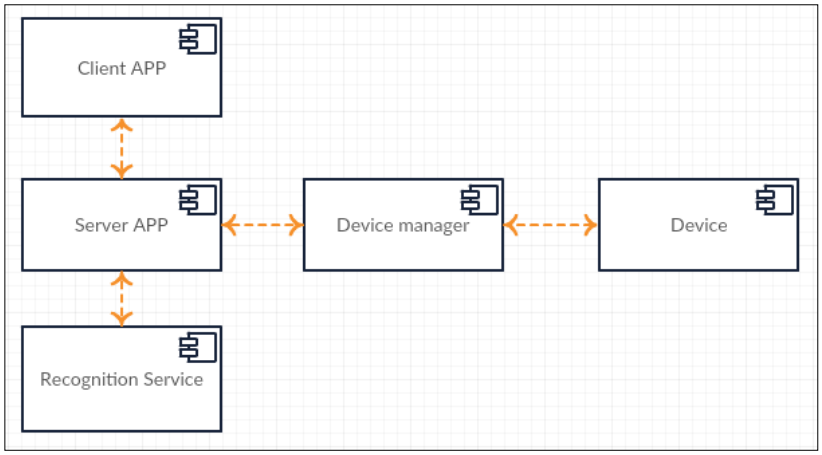


Рисунок 3.1 – Діаграма компонентів

Діаграма компонентів відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись. Модуль програмного забезпечення може бути представлено в якості компоненти. За допомогою діаграми компонентів представляються інкапсульовані класи разом з їх інтерфейсними оболонками, портами і внутрішніми структурами (які теж можуть

складатися з компонентів і конекторів). Компоненти зв'язуються через залежності, коли з'єднується необхідний інтерфейс одного компонента з наявними інтерфейсом іншого компонента. Таким чином ілюструються відносини клієнт-джерело між двома компонентами. Залежність показує, що один компонент надає сервіс, необхідний іншому компоненту. Залежність зображується стрілкою від інтерфейсу або порту клієнта до імпортованого інтерфейсу. Коли діаграма компонентів використовується, щоб показати внутрішню структуру компонентів, що надається і необхідний інтерфейси складеного компонента можуть делегуватися до відповідних інтерфейси внутрішніх компонентів.

Діаграма кооперації показує взаємодію між частинами системи (див. рис. 3.2). Користувач, відправляє мовний запит у клієнтському додатку. Клієнтський додаток відправляє мовний запит на сервер логіки, який в свою чергу відправляє запит до сервісу розпізнавання. Отримавши розпізнаваний сигнал, сервер логіки відправляє команду до менеджеру пристроїв, який в свою чергу дає команду до пристрою, якому відповідає ця команда.

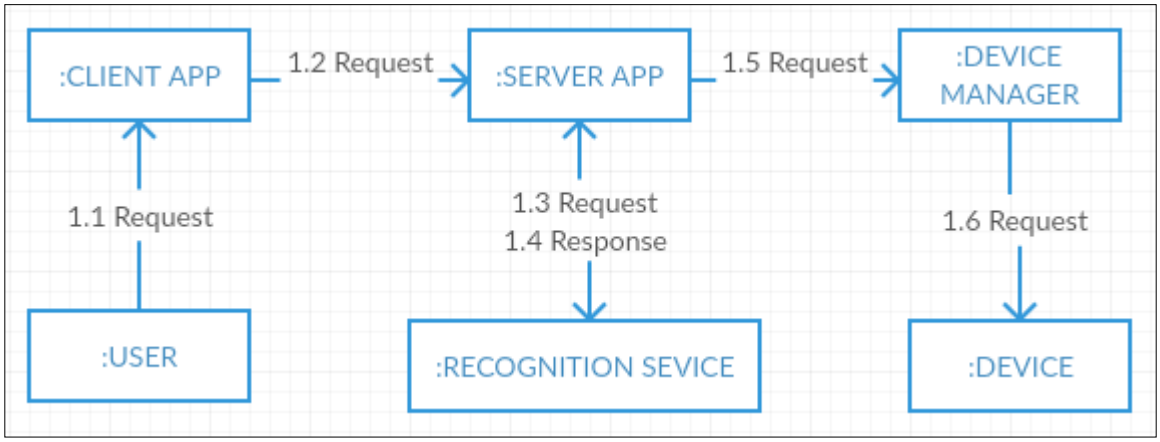


Рисунок 3.2 – Діаграма кооперації

Діаграма комунікації моделює взаємодії між об'єктами або частинами в термінах впорядкованих повідомлень. Комунікаційні діаграми представляють комбінацію інформації, взятої з діаграм класів, послідовності і варіантів використання, описуючи відразу і статичну структуру і динамічну поведінку

системи. Комунікаційні діаграми мають вільний формат упорядкування об'єктів і зв'язків як в діаграмі об'єктів. Щоб підтримувати порядок повідомлень при такому вільному форматі, їх хронологічно нумерують. Читання діаграми комунікації починається з повідомлення 1.0 і триває по напрямку пересилання повідомлень від об'єкта до об'єкта. Діаграма комунікації показує багато в чому ту ж інформацію, що і діаграма послідовності, але через іншого способу подання інформації якісь речі на одній діаграмі бачити простіше, ніж на інший. Діаграма комунікацій наочніше показує, з якими елементами взаємодіє кожен елемент, а діаграма послідовності ясніше показує в якому порядку відбуваються взаємодії.

За допомогою даних діаграм можна спостерігати основну структуру системи та етапи взаємодії чистин системи між собою.

3.2 Структура обміну даними

На сьогоднішній день прийнято використовувати REST – метод взаємодії компонентів розподіленого додатка в мережі Інтернет, при якому виклик віддаленої процедури являє собою звичайний HTTP-запит, а необхідні дані передаються як параметри запиту.

У свою чергу HTTP – протокол передачі даних, що використовуєтьсяв комп'ютерних мережах. Назва скорочена від Hyper Text Transfer Protocol, протокол передачі гіпер-текстових документів.

HTTP – протокол прикладного рівня, схожими на нього є FTP і SMTP.Обмін повідомленнями йде за звичайною схемою «запит– відповідь». Для ідентифікації ресурсів HTTP використовує глобальні URI. На відміну відбагатьох інших протоколів, HTTP не зберігає свого стану. Це означає відсутність збереження проміжного стану між парами «запит – відповідь».Компоненти, що використовують HTTP, можуть самостійно здійснювати збереження інформації про стан, пов'язаний з останніми запитами та відповідями. Браузер, котрий

посилає запити, може відстежувати затримки відповідей. Сервер може зберігати IP-адреси та заголовки запитів останніх клієнтів. Проте, згідно з протоколом, клієнт та сервер не мають бути обізнаними з попередніми запитами та відповідями, у протоколі не передбачена внутрішня підтримка стану й він не ставить таких вимог до клієнта та сервера.

Кожен запит/відповідь складається з трьох частин:

– стартовий рядок;

– заголовки;

– тіло повідомлення, що містить дані запиту, запитаний ресурс або опис проблеми, якщо запит не виконано.

Для функціонування методу REST потрібно наступне:

– клієнт-серверна архітектура;

– сервер не зобов'язаний зберігати інформацію про стан клієнта.

– у кожному запиті клієнта повинно явно міститися вказівка про можливість кешування відповіді і отримання відповіді з існуючого кеша;

– клієнт може взаємодіяти не безпосередньо з сервером, а з довільною кількістю проміжних вузлів. При цьому клієнт може не знати про існування проміжних вузлів, за винятком випадків передачі конфіденційної інформації;

– уніфікований програмний інтерфейс сервера.

Завдяки використанню REST наш продукт набув таких якостей:

– надійність (за рахунок відсутності необхідності зберігати інформацію про стан клієнта, яка може бути загублена);

– продуктивність (за рахунок використання кеша);

– масштабованість;

– прозорість системи взаємодії, особливо необхідна для додатків обслуговування мережі;

– простота інтерфейсів;

– портативність компонентів;

– легкість внесення змін;

– здатність еволюціонувати, пристосовуючись до нових вимог (на прикладі Всесвітньої павутини).

У якості контейнера даних використовується JSON – це текстовий формат обміну даними між комп'ютерами. Формат дозволяє описувати об'єкти та інші структури даних.

Для передачі команд від серверу логіки до менеджеру пристроїв використовується UART – тип асинхронного приймача-передавача, компонентів комп'ютерів та периферійних пристроїв, що передає дані між паралельною та послідовною формами. UART звичайно використовується спільно з іншими комунікаційними стандартами, такими як EIA RS-232.

UART це зазвичай окрема мікросхема чи частина мікросхеми, що використовується для з'єднання через комп'ютерний чи периферійний послідовний порт. UART нині загалом включені в мікроконтролери. Здвоєний UART (Dual UART або DUART) об'єднує двоє UART в одній мікросхемі. Багато сучасних мікросхем сьогодні випускаються з можливістю комунікації в синхронному режимі, такі прилади називають USART.

Біти даних передаються з одного місця в інше через дроти або інші носії. Якщо мова йде про великі відстані, вартість дротів стає великою. Щоб зменшити вартість довгих комунікацій, що переносять кілька біт паралельно, біти даних передають послідовно один за одним, , і використовують UART для перетворення паралельної форми на послідовну на кожному кінці лінії зв'язку. Кожен UART має зсувний регістр, який є фундаментальним методом для перетворення між паралельними та послідовними формами.

Зазвичай UART не отримує і не генерує зовнішні сигнали, які подорожують між різними частинами обладнання. Як правило, для перетворення логічного рівня UART в та з зовнішнього рівня сигналів використовується окремий інтерфейсний блок.

Зовнішній сигнал може мати багато різних форм. Прикладами стандартизованих напруг сигналу можуть служити RS-232, RS-422 чи RS-485 від EIA. Історично присутність або відсутність струму (в електричному колі)

використовувалася в телеграфних схемах. Деякі ж сигнальні схеми не використовують електричних дротів. Як приклад можна навести оптоволокно, інфрачервоний зв'язок чи Bluetooth в своєму Serial Port Profile (SPP). Прикладами модуляції є аудіо сигнал телефонних модемів, РЧ модуляція даних, або DC-LIN для комунікацій по силових дротах. Зв'язок може бути «дуплексним» (можливість одночасного прийому та передачі) або «напівдуплексним» (пристрої переключаються між режимами прийому та передачі).

UART широко використовується в інтерфейсі RS-232 для вбудованих систем комунікацій. Він використовується для зв'язку між мікроконтролерами і комп'ютером. Багато чипів забезпечують функціональність UART, та існують дешеві мікросхеми для конвертації логічного рівня сигналу (типу TTL) в сигнал рівня RS-232.

Під час асинхроної передачі UART телетайпного типу посилає стартовий біт, потім від п'яти до восьми бітів даних, перший – найменш значимий, потім опціональний біт парності, і потім один, півтора чи два стопових біти. Стартовий біт надсилається в зворотній полярності до звичайного незайнятого стану ліній зв'язку. Стоповий біт відповідає незайнятому стану лінії і забезпечує паузу перед наступною порцією даних. Це зветься асинхронною старт-стоповою передачею. В механічних телетайпах стоповий біт часто був розтягнутим вдвічі, щоб дати можливість механізму надрукувати символ. Розтягнутий стоповий біт також допомагав при ресинхронізації. Біт парності перевіряє кількість одиниць між стартовим і стоповим бітами або парним та непарним, або ж цей біт може бути відсутнім. Непарна перевірка надійніша, бо вона може засвідчити, що принаймні одна одиниця передалася, а це дозволяє багатьом UART пересинхронізуватися. В синхронній передачі частота тактового генератора відновлюється окремо з потоку даних і старт-стопові біти не використовуються. Це покращує ефективність каналу зв'язку для надійних ліній, також надсилається більше корисних даних. Асинхронна передача не посилає нічого, коли нема що передавати. Натомість синхронний інтерфейс має завжди посилати якісь дані, щоб підтримувати синхронізацію між передавачем і приймачем.

3.3 Розробка додатку

Для розробки додатку використовувалась мова розробки JavaScript. JavaScript – динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

```
var session = {
  audio: true,
  video: false
};
var recordRTC = null;
navigator.getUserMedia(session, function (mediaStream) {
  recordRTC = RecordRTC(MediaStream);
  recordRTC.startRecording();
}, onError);
```

Рисунок 3.3 – Отримання мовного повідомлення користувача

Клієнтський додаток не має користувацького інтерфейсу, а лише розміщається в фоновому режимі на пристрої користувача. Це дає можливість масштабування нашої системи. Додаток може бути поміщено в контейнер для

використання на сторінці в браузері або на мобільному додатку. Запити та передача мовних команд виконується за рахунок протоколу HTTP, який більш детально описано в розділі 3.2.

Таким чином отримавши мовну команду від користувача додаток відправляє її на сервер логіки. Приклад отримання мовного сигналу від користувача наведено на рисунку 3.3.

```
recordRTC.stopRecording(function(audioURL) {
  var formData = new FormData();
  formData.append('edition[audio]', recordRTC.getBlob())
  $.ajax({
    type: 'POST',
    url: 'some/path',
    data: formData,
    contentType: false,
    cache: false,
    processData: false,
  })
});
```

Рисунок 3.4 – Передача мовної команди

Коли команда записана, її необхідно передати до серверу бізнес логіки для подальшої обробки та виконання. Таким чином на рисунку 3.4 показано принцип відправки команди до серверу логіки, використовуючи бібліотеку jQuery AJAX. jQuery – популярна JavaScript-бібліотека з відкритим сирцевим кодом. Вона була представлена у січні 2006 року у BarCamp NYC Джоном Ресіґом (John Resig). Згідно з дослідженнями організації W3Techs, JQuery використовується понад половиною від мільйона найвідвідуваніших сайтів. jQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день. jQuery є вільним програмним забезпеченням під ліцензією MIT (до вересня 2012 було подвійне ліцензування під MIT та GNU General Public License другої версії).

Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і

розробки AJAX-застосунків. jQuery також надає можливості для розробників, для створення плагінів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня. Це сприяє створенню потужних і динамічних веб-сторінок.

Після отримання мовної команди сервером бізнес логіки ця команда повинна бути передана до сервісу розпізнання. Для цього запит зберігається у файлі формату WAV та відправляється до сервісу розпізнання (див. рис. 3.5).

```
  fetch('/api/speech-to-text/token').then(function(response) {
    return response.text();
  }).then(function (token) {

    stream = WatsonSpeech.SpeechToText.recognizeFile({
      token: token,
      file: document.querySelector('#audiofile').files[0],
      play: true, // play the audio out loud
      outputElement: '#output' // CSS selector or DOM Element (optional)
    });

    stream.on('error', function(err) {
        console.log(err);
    });

  }).catch(function(error) {
      console.log(error);
  });
};
```

Рисунок 3.5 – Передача файлу до сервісу розпізнання

WAV – формат аудіофайла розроблений компаніями Microsoft та IBM. WAVE базується на форматі RIFF, поширюючи його на інформацію про такі параметри аудіо, як застосований кодек, частота дискретизації та кількість каналів. WAV як і RIFF передбачався для комп'ютерів IBM PC, тому всі змінні записані у форматі little endian. Відповідником WAV для комп'ютерів PowerPC є AIFF. Хоча файли WAVE можуть бути записані за допомогою будь-яких кодеків аудіо, зазвичай використовується нестиснений PCM, який призводить до великих обсягів файлу (близько 172 кБ на секунду для CD-якості). Іншим недоліком файлу є обмеження

обсягу до 4 ГБ, через 32-бітну змінну. Формат WAV був частково витіснений стисненими форматами, проте, завдяки своїй простоті, надалі знаходить широке використання в процесі редагування звуку та на переносних аудіопристроях, як програвачі та цифрові диктофони.

Після того, як команда була розпізнана, до серверу логіки приходить відповідь з розпізнаною командою. Сервер логіки проводить аналіз на відповідність та належність розпізнаної команди у словнику команд, які доступні у системі. Для цього аналізу використовується дерево пошуку – динамічна нелінійна структура даних, кожен елемент якої містить власне інформацію (або посилання на те місце в пам'яті ЕОМ, де зберігається інформація) та посилання на кілька (не менше двох) інших таких же елементів. Дерева пошуку призначені для представлення словників як абстрактного типу даних. Вважається, що кожен елемент словника має ключ (вагу), що приймає значення з лінійно впорядкованої множини. Такою множиною може бути, наприклад, числова множина або множина слів в деякому алфавіті. В останньому випадку як лінійний можна розглядати лексикографічний порядок. Таким чином, дерево пошуку може бути використано і як словник, і як пріоритетна черга. Час виконання основних операцій пропорційний висоті дерева. Якщо кожен внутрішній вузол двійкового дерева має рівно двох нащадків, то його висота і час виконання основних операцій пропорційні логарифму числа вузлів. І навпаки, якщо дерево являє собою лінійний ланцюжок з n вузлів, цей час виростає до O(n). Відомо, що висота випадкового двійкового дерева є O(log n), так що в цьому випадку час виконання основних операцій є O(log n). Звичайно, що виникаючі на практиці двійкові дерева пошуку можуть бути далекі від випадкових. Однак, прийнявши спеціальні заходи по балансуванню дерев, можна гарантувати, що висота дерев з n вузлами буде O(log n).

Виконавши аналіз команди та отримавши вдалий результат, команда кодується у спеціальну послідовність символів, які потім передаються до менеджеру пристроїв. Отримавши команду від серверу бізнес логіки, менеджер пристроїв виконує необхідні дії над пристроєм до якого відноситься ця команда.

ВИСНОВКИ

У ході виконання атестаційної роботи була представлена структура сучасних систем розпізнавання мови і модулі, що входять до її складу, а саме модуль обробки сигналу і вилучення ознак, акустична модель, мовна модель, декодер. Було описано два підходи до побудови акустичних моделей – GMM-HMM і DNN-HMM. Було проведено огляд методів адаптації акустичних моделей на основі глибоких нейронних мереж, розроблених для компенсації невідповідності умов навчання і експлуатації і, таким чином, підвищують стійкість системи розпізнавання по відношення до акустичної варіативності мовного сигналу. Одним з найбільш перспективних методів адаптації слід визнати адаптацію з використанням і-векторів. Було розглянуто та проаналізовано ефективна методика навчання системи розпізнавання англійської мови. Зроблено висновки про перспективності алгоритмів нормалізації ознак і адаптації акустичних моделей, а також про перевагу DNN-HMM акустичних моделей над GMM-HMM в задачі розпізнавання мовлення.

Також була наведена інтерпретація глибокої нейронної мережі як складової моделі, що поєднує каскад нелінійних перетворень вхідних ознак і логлінейние класифікатор. Наведено результати досліджень, що показують, що нелінение перетворення ознак, що здійснюються на прихованих шарах глибокої нейронної мережі, забезпечують стійкість по відношенню до малих збурень вхідного сигналу. Дано опис глибоких нейронних мереж з вузьким горлом, службовців для вилучення ознак, що володіють стійкістю по відношенню до акустичної варіативності мовного сигналу.

Було розроблено метод побудови ознак, які з глибокої нейронної мережі з вузьким горлом, адаптованої до диктора і акустичної обстановці за допомогою і-векторів. Запропоновано алгоритм навчання акустичних моделей на основі глибоких нейронних мереж з використанням побудованих ознак. Розроблено двоетапний алгоритм ініціалізації навчання акустичних моделей на основі

глибоких нейронних мереж, призначений для зменшення впливу сегментів, які містять мова, на навчання акустичної моделі. Проведено експериментальні дослідження, що підтверджують ефективність запропонованих методу та алгоритму в задачі розпізнавання англійської мови.

У ході роботи була спроектована система розумного дому, яка використовує передові методи штучного інтелекту, а саме методи розпізнання людського мовлення.

Перспективними напрямками подальшої розробки теми можна виділити, по-перше, поліпшення методу побудови інформативних ознак, що витягають з адаптованої до диктора і акустичних умов глибокої нейронної мережі, за рахунок навчання глибокої нейронної мережі з вузьким горлом з використанням критеріїв поділу послідовностей. По-друге, підвищення точності розпізнавання команд за рахунок застосування акустичних моделей на основі загортальних і рекурентних нейронних мереж. По-третє, підвищення точності розпізнавання команд за допомогою застосування підходів до побудови мовних моделей, що дозволяють ефективно враховувати дальній смисловий контекст, а також морфологічну, синтаксичну та семантичну інформацію. Та на сам перед, підвищення швидкодії системи розпізнавання команд.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ронжин А.Л., Карпов А.А., Ли И.В. Речевой и многомодальный интерфейсы – М.: Наука, 2006. – 173 с.

2. Ушакова Т.Н. Проблема внутренней речи в психологии и психофизиологии. Психологические и психофизиологические исследования речи – М.: Наука, 1985. – С. 13–26.

3. Галунов В.И. Состояние исследований в области речевых технологий и задачи, выдвигаемые государственными заказчиками – М., 2002. – 3 с.

4. Беллман Р. Динамическое программирование – М.: ИЛ, 1960, 400 с.

5. Ронжин А. Метод распознавания слитной речи на основе анализа сигнала в скользящем окне и теории размытых множеств //Научно-теоретический журнал «Искусственный интеллект», №4. – Донецк, Украина, 2002, С. 256–263.

6. Геппенер В.В. Вейвлет-преобразование в задачах цифровой обработки сигналов: Учебное пособие – СПб.: Изд-во СПбГЭТУ, 2002. 78 с.

7. Baum L. Statistical inference for probabilistic functions of finite state Markov chains // Ann. Math. Statist. – 1966. – Vol. 37, no. 6. – P. 1554–1563.

8. Dempster A. Maximum-likelihood from incomplete data via the EM algorithm – 1977. – Vol. 39, no. 1. – P. 1–38

9. Gales M. Semi-tied Covariance Matrices for Hidden Markov Models – 1999. – Vol. 7. – P. 272–281.

10. Leggetter C. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models – 1995. – Vol. 9, no. 2. – P. 171–185.

11. Povey D. Discriminative Training for Large Vocabulary Speech Recognition – Cambridge : [s. n.], 2003.

12. Yu D. Automatic Speech Recognition: A Deep Learning Approach – London : Springer-Verlag, 2015.

13. Nielsen, M. Neural Networks and Deep Learning [Electronic resource]. – 2016. – URL: http://neuralnetworksanddeeplearning.com/ (online; accessed: 03.04.2018)

14. Extracting and composing robust features with denoising autoencoders // Proc. International Conference on Machine Learning (ICML). – 2008. – P. 1096–1103.

15. Povey D. Boosted MMI for model and feature-space discriminative training // Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2008. – P. 4057– 4060.

16. Veselý K. Sequence-discriminative training of deep neural networks // Proc. Annual Conference of International Speech Communication Association (INTERSPEECH). – 2013.

17. Xue J. Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network // Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2014. – P. 6359–6363.

18. Ortmanns, S. Look-ahead techniques for fast beam search // Computer Speech and Language. – 2000. – Vol. 14. – P. 15–32.

19. Gaida C. Comparing Open-Source Speech Recognition Toolkits – 2014. – URL: http://suendermann.com/su/pdf/oasis2014.pdf (online; accessed: 03.04.2018).

20. Ratnaparkhi A. A simple introduction to maximum entropy models for natural language processing // IRCS Technical Reports Series. – 1997.

21. Yu D. Feature Learning in Deep Neural Networks – studies on Speech Recognition Tasks // Proc. ICLR. – 2013.

22. Lamel L. Transcription of Russian conversational speech // Proc. SLTU. – 2012. – P. 156–161.

23. Zhang Y. Extracting deep neural network bottleneck features using low-rank matrix factorization // Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2014. – P. 185–189.

24. Kaldi ASR Toolkit [Electronic resource]. – 2016. – URL: http://kaldi-asr.org/ (online; accessed: 18.04.2018).

25. Saon G. Speaker adaptation of neural network acoustic models using i-vectors // Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). – 2013. – P. 55–59.

26. Меденников И. П. Двухэтапный алгоритм инициализации обучения акустических моделей на основе глубоких нейронных сетей // Научно-технический вестник информационных технологий, механики и оптики. – 2016.– Т. 16, № 2. – С. 379–381.

ДОДАТОК Б

Текст тез та наукових статей

Artificial intelligence, is a branch of computer science. This branch involves the development of software programs to complete specific tasks that would otherwise depend on human intelligence. The main goal of AI algorithms is to tackle logical reasoning, language understanding, problem–solving, perception, and learning. Robots that are artificially intelligent bridge the gap between AI and robotics. AI software and programs control these robots. Most robots today are not artificially intelligent. Until recently, industrially designed robots could only carry out a series of repetitive movements based on programming. As mentioned previously, repetitive motions do not require the use of artificial intelligence.

Искусственный интеллект сегодня одна из передовых областей исследований ученых. Причем рассматриваются как системы, созданные с его частичным использованием: например распознавание текстов, бытовые роботы, до возможности замены творческого труда человека искусственным. Сегодня в самых различных областях науки и техники требуется выполнение машинами тех задач, которые под силу были только человеку. На помощь тогда приходит искусственный интеллект, который может заменить человека в какой либо рутинной и скучной деятельности. Сегодня системы, как программные, так и аппаратные, созданные на основе искусственного интеллекта, находят все большее применение в технике. Голосовые помощники делают IoT привычным. В миллионах домов они сообщают время, погоду, включают музыку по запросу и помогают составить список покупок. Пользователь, привыкший к подобным функциям, с большей готовностью примет и другие технологии умного дома. Как показало исследование, приобретение Amazon Echo или Google Home, умных колонок с голосовым управлением, повышает вероятность покупки новых IoT–устройств [1].

Традиционные системы распознавания речи были основаны на математическом аппарате скрытых марковских моделей. Российский математик Андрей Марков, в честь которого названы модели, при исследовании задач обработки литературных текстов в начале XX века оценивал вероятность появления каждой буквы в тексте в зависимости от ее контекста. Для упрощения вычислений он предположил, что эти вероятности зависят только от одной предыдущей буквы – марковское свойство. Оказалось, что оценки вероятностей перехода от одной буквы к другой по разным фрагментам одного текста практически идентичны. В дальнейшем выяснилась уникальность параметров цепи Маркова для каждого автора, что позволило применить их в задачах определения авторства текста. Примерно до 2010 года на практике использовалась модель гауссовых смесей для задания распределения наблюдаемого сигнала в зависимости от фонемы. Для этого звуковой сигнал делится на небольшие участки (10–50 мс), для применения традиционной

обработки сигналов в частотной области для каждого участка сигнала выполняется быстрое преобразование Фурье. Дальше использовалось логарифмирование получаемого спектра в связи с известным логарифмическим восприятием человеческим ухом масштаба звука. Наконец, с помощью дискретного косинусного преобразования логарифма спектра получались практически независимые признаки – кепстральные коэффициенты, распределение которых и записывалось в виде смеси гауссовских случайных векторов с диагональными ковариационными матрицами. Потом в связи с революцией глубокого обучения вместо традиционного подхода к извлечению характерных признаков и их описания моделью гауссовых смесей для построения акустической модели речи стали использовать глубокие нейронные сети. В задаче распознавания речи применялись обычные сети прямого распространения с большим числом слоев, которые обучались в режиме без учителя последовательно от одного слоя к другому слою. Оказалось, что применение такого подхода совместно с аппаратом скрытых марковских моделей, включающих вероятности перехода от одной фонемы к другой, на десятки процентов повышают точность распознавания спонтанной речи. Именно этот подход в настоящее время реализован в большинстве современных программных библиотек распознавания речи. Наряду с появлением новой акустической модели речи вторым прорывным моментом стали новые языковые, лингвистические, модели. В них в самом простом случае требуется предсказать следующее слово по известным предыдущим словам – задача, типичная для обработки текстов. В традиционных системах применялись модели типа N-грамм, в которых на основе большого количества текстов оценивались распределения вероятности появления слова в зависимости от N предшествующих слов. Для получения надежных оценок распределений параметр N должен быть достаточно мал: одно, два или три слова – модели униграмм, биграмм или триграмм соответственно [2].

Развитие современных речевых технологий идет в сторону реализации полного цикла обучения систем распознавания спонтанной речи без выделения отдельных акустических и лингвистических моделей. Вместо предварительного отбора акустических признаков, таких как кепстральные коэффициенты, все участки речевого сигнала представляются своими спектрограммами, которые подаются на вход одной большой нейронной сети [3].

У наш час персональні комп’ютери відіграють важливу роль майже у всіх сферах діяльності людини. Обчислювальна техніка значно полегшує працю людини. Сьогодні у зв'язку з розвитком інформаційних технологій, появою їх у всіх сферах життєдіяльності людини, збільшенням обсягів і потоків інформації, що зберігається в основному на електронних носіях, проводиться автоматизація майже усіх сфер діяльності. Навіть сфери діяльності, які, здавалось би, ніяк не можуть обійтись без участі людини, автоматизуються.

Питання людино-машинного взаємодії є одними з найважливіших при створенні нових комп'ютерів. Найбільш ефективними засобами взаємодії людини з машиною були б ті, які є природними для нього: через візуальні образи і мову.

Створення мовних інтерфейсів могло б знайти застосування в системах самого різного призначення: голосове управління для людей з обмеженими можливостями, надійне управління бойовими машинами, «розуміючими» тільки голос командира, автовідповідачі, обробні в автоматичному режимі сотні тисяч дзвінків на добу (наприклад, в системі продажу авіаквитків) і т.д. При цьому, мовної інтерфейс повинен включати в себе два компоненти: систему автоматичного розпізнавання мови для прийому мовного сигналу і перетворення його в текст або команду, і систему синтезу мовлення, що виконує протилежну функцію – конвертацію повідомлення від машини в мову [1].

Однак, не дивлячись на стрімко зростаючі обчислювальні потужності, створення систем розпізнавання мови залишається надзвичайно складною проблемою. Це обумовлюється як її міждисциплінарним характером (необхідно володіти знаннями в філології, лінгвістиці, цифровій обробці сигналів, акустиці, зі статистикою, розпізнаванні образів і т.д.), так і високою обчислювальною складністю розроблених алгоритмів. Останнє накладає суттєві обмеження на системи автоматичного розпізнавання мови – на обсяг оброблюваного словника, швидкість отримання відповіді і його точність. Не можна також не згадати про те, що можливості подальшого збільшення швидкодії ЕОМ за рахунок вдосконалення інтегральної технології рано чи пізно будуть вичерпані, а все зростаюча різниця між швидкодією пам'яті і процесора тільки посилює проблему.

Можна виділити три головні проблеми концепції розпізнавання мови. Перша проблема – як буде виглядати модель розпізнавання мови, якщо не використовувати популярну, але малопродуктивну вірогідну модель? Тут багато що залежить від поставлених завдань і застосування системи. Друга проблема – на чому має ґрунтуватися вибір опису первинного мовного сигналу? Ця проблема тісно межує з можливостями розуміння людиною процесів формування мови. Тут великі надії подає квантова теорія мови, але поки що в більшості систем використовується стандартний статистичний аналіз акустичних параметрів мовлення. Також перспективним виглядає використання нейронних мереж. Третя проблема – проблема взаємодії первинних мовних ознак з більш високими рівнями (семантика, прагматика) при відмові від класичної лінійної моделі вхідного мовного сигналу. Тут основними завданнями є подолання проблем омонімії, «словесного сміття», перешкод різного типу, а також коректної побудови бази знань, системи попереднього навчання і аналізу [2].

Після проведення дослідження основних методів штучного інтелекту, а саме методів розпізнавання мовлення, було обрано стратегію агрегації для реалізації системи розумного дому. Дана стратегія дозволяє, за рахунок часу аналізу та кількості запитів, покращити ефективність розпізнавання та аналізу мовних команд для системи розумного дому. На сам перед, система повинна точно зчитувати мовну команду за рахунок якісної апаратури високої чіткості. Наступний етап – передача даних на різноманітні сервіси або бібліотеки оцінювання мовлення. Перший крок – передача даних для розпізнання. У разі невдалого розпізнання команди виконується наступний запит до іншої системи або бібліотеки розпізнання команд. Другий етап – пошук співпадіння команди зі списком, який предоставляє інтерфейс даної системи. Співпадіння вираховується

за рахунок логічної схожості команд системи та мовної команди користувача. Наприклад, команди «погасити світло» та «вимкнути світло» являються ідентичними с точки зору користувача, але різними з точки зору повного спів падіння. Даний етап дозволить розпізнавати команди без точного спів падіння. Наступний етап – оцінювання команди на наявність глузду. Даний етап виконується аналогічним чином: розпізнана команда передається до сервісу аналізу мовлення, у разі негативної відповіді, запит виконується повторно до наступної системи. Якщо запити до всіх систем були негативними, то система дає відмову на виконання даної команди, в іншому випадку, система дає команду системі, до якої відноситься дана команда, на виконання [3].

Таким чином дослідивши, оцінивши та обравши стратегію агрегації майбутня система наділяється такими показниками як: стерсостікість, надійність (за рахунок етапу оцінювання наявності глузду), незалежність від сторонніх ресурсів (велика кількість ресурсів дозволяє взаємозамінювати сервіси розпізнання мовлення).