HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

TRẦN THỊ NHI AN

NHẬN DẠNG NGƯỜI NÓI THEO TIẾP CẬN MÁY HỌC HIỆN ĐẠI

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

TP. HỒ CHÍ MINH – NĂM 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

TRẦN THỊ NHI AN

NHẬN DẠNG NGƯỜI NÓI THEO TIẾP CẬN MÁY HỌC HIỆN ĐẠI

Chuyên ngành: HỆ THỐNG THÔNG TIN
Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC:

PGS.TS VŨ HẢI QUÂN

TP. HỒ CHÍ MINH – NĂM 2022

LỜI CAM ĐOAN

Tôi cam đoan rằng luận văn: ***"Nhận dạng người nói theo tiếp cận máy học hiện đại"*** là công trình nghiên cứu của chính tôi dưới sự hướng dẫn của **PGS.TS Vũ Hải Quân**.

Tôi cam đoan các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Không có sản phẩm/nghiên cứu nào của người khác được sử dụng trong luận văn này mà không được trích dẫn theo đúng quy định.

TP. Hồ Chí Minh, ngày 04 tháng 05 năm 2022

Học viên thực hiện luận văn

Trần Thị Nhi An

LỜI CẢM ƠN

Trước hết, em xin bày tỏ lòng biết ơn chân thành và sâu sắc tới Thầy **PGS.TS Vũ Hải Quân** người Thầy kính yêu đã hết lòng giúp đỡ, hướng dẫn, động viên, tạo điều kiện cho tôi trong suốt quá trình thực hiện và hoàn thành luận văn.

Tôi xin chân thành cảm ơn Ban Giám Đốc, Phòng đào tạo sau đại học và quý Thầy Cô của Học viện Công Nghệ Bưu Chính Viễn thông cơ sở tại TP.HCM đã giảng dạy và tạo điều kiện học tập thuận lợi trong suốt khóa học.

Tôi xin chân thành cảm ơn gia đình, bạn bè, đồng nghiệp trong cơ quan đã động viên, hỗ trợ tôi trong lúc khó khăn để tôi có thể học tập và hoàn thành luận văn.

Mặc dù đã có nhiều cố gắng, nỗ lực, nhưng do thời gian và kinh nghiệm nghiên cứu khoa học còn hạn chế nên không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý của quý Thầy Cô cùng bạn bè đồng nghiệp để kiến thức của tôi ngày một hoàn thiện hơn.

Một lần nữa tôi xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 04 tháng 05 năm 2022

Học viên thực hiện luận văn

Trần Thị Nhi An

DANH SÁCH HÌNH VẼ

Hình 1.1: Các đặc tính sinh trắc 1
Hình 2.1: Phân biệt xác minh và định danh 14
Hình 2.2: Trí tuệ nhân tạo – AI 18
Hình 2.3: Lấy mẫu và số hóa tín hiệu analog, sau đó tái tạo lại tín hiệu này 19
Hình 2.4: Cấu trúc của hệ thống nhận dạng người nói 20
Hình 2.5: Các bước trích xuất MFCC từ tín hiệu âm thanh.....24
Hình 2.6: Các lĩnh vực ứng dụng của Machine Learning 27
Hình 2.7: Ba mô hình học tập cho các thuật toán 28
Hình 3.1: Các lớp của một mạng nơ-ron điển hình.....30
Hình 3.2: Mối liên hệ giữa AI, ML và DL.....32
Hình 3.3: Perceptron 33
Hình 3.4: Feed Forward Neural Networks.....33
Hình 3.5: Multilayer Perceptron 34
Hình 3.6: Convolutional Neural Network.....35
Hình 3.7: Radial Basis Function Neural Networks.....35
Hình 3.8: Recurrent Neural Networks 36
Hình 3.9: Long Short-Term Memory.....37
Hình 3.10: Modular Neural Network.....38
Hình 3.11: Ví dụ về dự đoán thời tiết 39
Hình 3.12: Một mô hình Markov ẩn 41
Hình 3.13: Các giai đoạn xử lý trong HTK.....42
Hình 3.14: Huấn luyện từ phụ trong HMM 45
Hình 3.15: Mạng truyền thẳng một lớp ẩn.....49
Hình 3.16: Cấu trúc mạng feedforward-DNN 54
Hình 4.1: Biểu đồ hiển thị tỉ lệ giới tính trong bộ dữ liệu.....56
Hình 4.2: Biểu đồ hiển thị tỉ lệ vùng miền trong bộ dữ liệu 57
Hình 4.3: Biểu đồ thống kê độ tuổi của bộ dữ liệu 57

Hình 4.4: Kết quả thống kê trên tập huấn luyện 64
Hình 4.5: Kết quả thống kê trên tập kiểm thử.....64
Hình 4.6: Biến thiên độ chính xác theo số lần chạy mô hình 66
Hình 4.7: Giao diện chương trình demo 67
Hình 4.8: Chọn file âm thanh để tiến hành nhận dạng.....68
Hình 4.9: Trường hợp nhận dạng với HMM.....68
Hình 4.10: Trường hợp nhận dạng với Feedforward-DNN 68

DANH SÁCH BẢNG

Bảng 4.1: Thông tin người tham gia ghi âm 55
Bảng 4.2: Thông tin chi tiết của một bản ghi âm 58
Bảng 4.4: Độ chính xác của mô hình qua số lần chạy training.....65

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Nghĩa Tiếng Anh	Nghĩa Tiếng Việt
SVM	Support vector machine	
HTK	Hidden Markov Model Toolkit	Bộ công cụ mô hình Markov ẩn
HMM	Hidden Markov Model	Mô hình Markov ẩn
CNN	Convolutional neural network	Mô hình tích hợp
DNN	Deep Neural Network	Mô hình học sâu
WER	Word Error Rate	Tỉ lệ lỗi từ
LPCC	Linear Predictive Cepstral Coefficients	
PLPC	Perceptual Linear Prediction Coefficients	
MFCC	Mel-Frequency Cepstral Coefficients	
ADC	Analog-to-Digital Converter	Bộ chuyển đổi analog sang kỹ thuật số
DAC	Digital-to-Analog Converter	Bộ chuyển đổi tín hiệu digital thành analog

MỤC LỤC

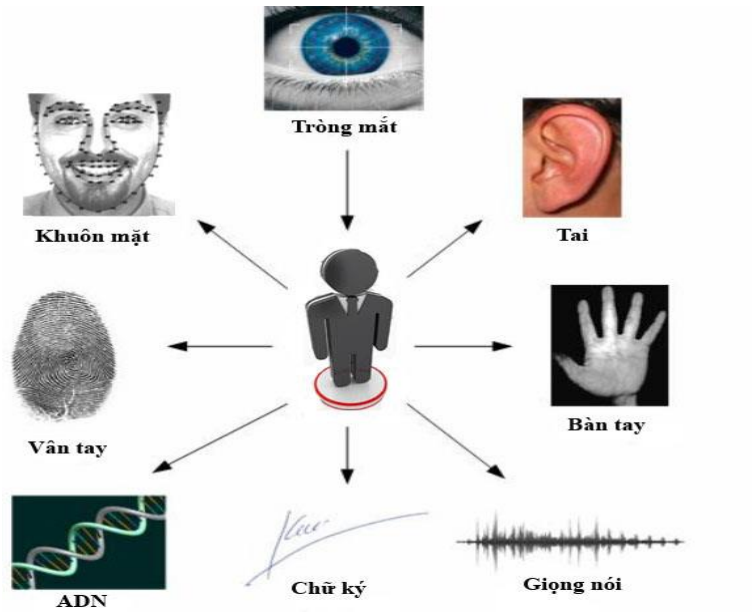
LỜI CAM ĐOAN i
LỜI CẢM ƠN ii
DANH MỤC HÌNH ẢNH iii
DANH MỤC BẢNG v
DANH MỤC CHỮ VIẾT TẮT vi
MỤC LỤC vii
CHƯƠNG 1: PHẦN MỞ ĐẦU 1
1.1. Lĩnh vực đề tài 1
1.2. Tình hình nghiên cứu liên quan đến đề tài 2
1.2.1. Các công trình nghiên cứu trong nước 2
1.2.2. Các công trình nghiên cứu trên thế giới 4
1.3. Mục tiêu, ý nghĩa khoa học và thực tiễn 9
1.4. Đối tượng và phạm vi nghiên cứu 10
1.4.1. Đối tượng nghiên cứu 10
1.4.2. Phạm vi nghiên cứu 10
1.5. Phương pháp nghiên cứu 10
1.5.1. Phương pháp nghiên cứu lý thuyết 10
1.5.2. Phương pháp nghiên cứu thực nghiệm 11
1.6. Bố cục luận văn 11
CHƯƠNG 2: TỔNG QUAN ĐỀ TÀI 12
2.1. Giới thiệu chung 12
2.1.1. Nhận dạng người nói là gì? 12
2.1.2. Ứng dụng công nghệ nhận dạng người nói vào đời sống 15
2.1.3. Tổng quan về trí tuệ nhân tạo (AI) 18
2.2. Tín hiệu giọng nói 19
2.3. Các thành phần chính của hệ thống nhận dạng người nói 20
2.4. Rút trích đặc trưng 21

2.4.1. Rút trích đặc trưng là gì 21
2.4.2. Các đặc trưng âm thanh phổ biến cho việc thiết lập mô hình 22
2.5. Mô hình máy học 25
2.5.1. Khái niệm về máy học 25
2.5.2. Các loại mô hình máy học 28
CHƯƠNG 3: NHẬN DẠNG NGƯỜI NÓI VỚI DEEP LEARNING 30
3.1. Mạng nơ-ron và deep learning 30
3.1.1. Mạng nơ-ron 30
3.1.2. Deep learning 31
3.2. Phân loại / các dạng mạng neural nhân tạo 32
3.3. Nhận dạng người nói 38
3.3.1. Nhận dạng người nói với HMM 38
3.3.2. Nhận dạng người nói với Feedforward-DNN 48
CHƯƠNG 4: THỰC NGHIỆM 55
4.1. Dữ liệu thực nghiệm 55
4.2. Kịch bản thực nghiệm 58
4.2.1. Chuẩn bị môi trường 58
4.2.2. Chuẩn bị dữ liệu 59
4.2.3. Xây dựng mô hình và huấn luyện 60
4.3. Thực nghiệm và đánh giá 62
4.3.1. Độ đo đánh giá 62
4.3.2. Thực nghiệm và so sánh 64
4.3.3. Phân tích và đánh giá 66
4.4. Chương trình demo 66
CHƯƠNG 5: KẾT LUẬN 70
5.1. Các đóng góp của luận văn 70
5.2. Kết luận và hướng phát triển 70
DANH MỤC TÀI LIỆU THAM KHẢO 72

CHƯƠNG 1: PHẦN MỞ ĐẦU

1.1. Lĩnh vực đề tài

Đề tài thuộc lĩnh vực Sinh trắc học (Biometrics). Sinh trắc học là khoa học nghiên cứu các phương pháp phân tích và thống kê trên các dữ liệu sinh học. Cụm từ "biometrics" xuất phát từ chữ "bio" (life) và "metrics" (measure) trong tiếng Hy Lạp. Trong lĩnh vực công nghệ thông tin, sinh trắc học được áp dụng trong việc nhận dạng người dựa trên những đặc điểm sinh lý học (physiological) và các mẫu hành vi (behavioral). Các hệ thống sinh trắc đã và đang được phát triển trong các ứng dụng thực tế như: các hoạt động của chính phủ, các công ty, tổ chức thương mại - tài chính, bao gồm việc quản lý nhân công, quản lý khách hàng, quản lý kiểm soát vào ra, đến quản lý xuất nhập cảnh, quản lý tội phạm, hệ thống bầu cử, v.v... Nhận dạng sinh trắc hiện đại đang nhận được nhiều sự quan tâm trong các lĩnh vực cần mức độ bảo mật và an toàn cao, cũng như do tính thuận tiện và năng động mà nó mang lại. Từ đó nó đã ngày càng chứng minh được tiềm năng ứng dụng rộng rãi so với các phương pháp nhận dạng truyền thống.



Hình 1.1: Các đặc tính sinh trắc *(nguồn [2])*

Dựa trên những đặc trưng sinh trắc (Hình 1.1), ta có thể phân chia thành hai nhóm chính là sinh trắc thể (physiological) và sinh trắc hành vi (behavioral):

- Sinh trắc thể (Physiological): là những đặc trưng liên quan đến hình dạng, cấu tạo của cơ thể bao gồm các đặc trưng sinh học như khuôn mặt (face), DNA, vân tay (fingerprint), hình dạng bàn tay (hand geometry), tròng mắt (iris), giọng nói (voice),... Trong đó, vân tay là đặc trưng đã được nghiên cứu và sử dụng tương đối rộng rãi trong các hệ thống nhận dạng như hệ thống đăng nhập hệ điều hành máy tính, hệ thống khóa cửa vân tay, v.v...
- Sinh trắc hành vi (Behavioral): là các đặc điểm về hành vi của con người như thói quen gõ phím (keystroke), chữ ký (signature), giọng nói (voice)...

Ở thời điểm bùng nổ về CNTT-TT, IoT và CMCN 4.0 thì vai trò của Sinh trắc học càng được nhấn mạnh hơn trong nhiều lĩnh vực xã hội và đời sống. Ngày càng có nhiều công trình trên thế giới khai thác các đặc tính sinh trắc để làm cầu nối giữa ứng dụng thực tiễn và xác thực chủ thể. Tuy nhiên, nghiên cứu trong nước về lĩnh vực này lại chưa nhiều, chưa có những giải pháp thực sự thuyết phục được cộng đồng và doanh nghiệp. Do đó, tôi chọn đề tài "**Nhận dạng người nói theo tiếp cận máy học hiện đại**", với mong muốn góp một phần nhỏ vào khảo sát học thuật mà cụ thể là đặc tính sinh trắc về giọng nói, nhằm làm tăng tính khả thi hơn cho ứng dụng trong nước.

1.2. Tình hình nghiên cứu liên quan đến đề tài

Hiện nay, các công trình liên quan đến đề tài nhận dạng giọng nói ngày càng phát triển và đa dạng, nghiên cứu các công trình này sẽ góp phần giúp củng cố hơn phần cơ sở lý thuyết và định hướng nghiên cứu, phát triển cho đề tài của luận văn.

1.2.1. Các công trình nghiên cứu trong nước

Tác giả Cao Truong Tran và cộng sự đã công bố bài nghiên cứu "*Deep Representation Learning for Vietnamese Speaker Recognition*" [1]. Bài báo này đã đề xuất một phương pháp học tập chuyển giao sâu tích hợp cả học tập chuyển giao và học tập sâu để xây dựng mô hình nhận dạng người nói tiếng Việt. Họ đã tạo cấu

hình theo mô hình nhận dạng người nói của SOTA. Đường cơ sở này đã được cố định cho tất cả các thử nghiệm đào tạo. Cấu hình đường cơ sở này sử dụng một đoạn thời gian 2 giây có độ dài cố định được trích xuất ngẫu nhiên từ mỗi câu nói. Nhấn mạnh trước được áp dụng cho tín hiệu đầu vào sử dụng hệ số 0,97. Hơn nữa, các biểu đồ quang phổ được trích xuất từ một cửa sổ hamming có chiều rộng 25ms và bước là 10ms và kích thước FFT là 512. Mel-filterbanks 64 chiều được sử dụng làm đầu vào cho mạng. Sau đó, kết hợp tổn thất Nguyên mẫu với tổn thất softmax để chứng minh sự cải thiện liên tục trong việc sử dụng từng hàm tổn thất. Hơn nữa, đề cập đến mô hình được tối ưu hóa hiệu suất, Attentive Statistics Pooling (ASP) được sử dụng để tổng hợp các khung thời gian, trong đó độ lệch chuẩn có trọng số theo kênh được tính toán ngoài giá trị trung bình có trọng số. Các tác giả đã đào tạo tất cả các phương pháp với 500 epoch và một cấu hình tương tự. Ngoài ra, tốc độ lấy mẫu cho tất cả các bài huấn luyện có độ nhất quán là 16000 mẫu / giây (16kHz), hoàn toàn đủ cho hầu hết các mô hình nhận dạng người nói cơ bản. Các tác giả đã sử dụng các biến thể khác nhau của mạng Residual bao gồm ResNetSE34V2, ResnetSEHalf, ResNetSE34L, VGG-Vox theo kiến trúc lưu trữ mô hình cơ bản trong học sâu. Kết quả thử nghiệm chỉ ra rằng phương pháp được đề xuất có thể xây dựng các mô hình chính xác để nhận dạng người nói tiếng Việt.

Tác giả Diep Dao Thi Thu, Quang Nguyen Hong và cộng sự đã công bố bài nghiên cứu "*Text-dependent Speaker Recognition for Vietnamese*" [2]. Bài báo này trình bày một phương pháp mới để nhận dạng người nói phụ thuộc vào văn bản tiếng Việt. Hệ thống được lập mô hình cho từng người nói sử dụng mô hình hỗn hợp Gaussian GMM (Gaussian Mixture Model). Các âm vị trong các từ khóa được biểu diễn bằng các mô hình Markov ẩn HMM. Xác suất trước và sau cho từ khóa và người nói đã được kết hợp với nhau để xác định người nói. Kết quả cho thấy trong trường hợp người nói không nói một cụm từ đủ dài, cách tiếp cận này đã tăng hiệu suất nhận dạng người nói.

Xác thực người nói là nhận dạng người dùng từ sinh trắc học giọng nói và có nhiều ứng dụng như bảo mật ngân hàng, tương tác với máy tính của con người và xác

thực môi trường xung quanh. Trong công trình "*Vietnamese Speaker Authentication Using Deep Models"* [3], nhóm tác giả khảo sát tính hiệu quả của các tính năng âm thanh như hệ số âm tần Mel (MFCC), hệ số âm tần Gammatone (GFCC) và Mã dự đoán tuyến tính (LPC) được trích xuất từ các luồng âm thanh để xây dựng hình ảnh phổ đặc trưng. Ngoài ra, chúng tôi đề xuất sử dụng các mô hình mạng Residual sâu để xác minh người dùng từ các hình ảnh phổ đặc trưng. Phương pháp đề xuất được đánh giá theo hai cài đặt trên bộ dữ liệu được thu thập từ 20 người nói tiếng Việt. Kết quả, với tỷ lệ Equal Error là khoảng 4%, đã chứng minh rằng tính khả thi của xác thực người nói tiếng Việt bằng cách sử dụng các mô hình mạng Residual sâu được đào tạo với hình ảnh tính năng phổ GFCC.

Bài báo "*Speaker Diarization in Vietnamese Voice*" của Nguyen Duc Nam và Hieu Trung Huynh [4].Phân cực người nói là quá trình phân chia luồng âm thanh đầu vào thành các phân đoạn đồng nhất theo các loa khác nhau. Đây là một quá trình quan trọng để hỗ trợ hệ thống nhận dạng người nói và xác định người nói trong chương trình phát sóng, bản ghi cuộc họp và thư thoại. Đặc biệt nó là bước cơ bản của hệ thống đánh giá đọc checklist tự động trong phòng mổ. Trong nghiên cứu này, nhóm tác giả giới thiệu một cách tiếp cận phân cực người nói trong giọng nói tiếng Việt. Phương pháp được đề xuất bao gồm vectơ hóa giọng nói dựa trên vectơ x và sau đó phân nhóm bằng các kỹ thuật phân cấp trung bình, k-means và tổng hợp để xác định người nói trong âm thanh. Phương pháp này đạt độ chính xác 89,29% đối với cuộc đối thoại giả 2 người được tạo từ bộ thử nghiệm của bộ dữ liệu VIVOS Corpus.

1.2.2. Các công trình nghiên cứu trên thế giới

Trong bài báo [5] của tác giả Rashid Jahangir và các cộng sự vào năm 2020 đã chỉ ra rằng hầu hết các nghiên cứu về nhận dạng người nói đã sử dụng các tính năng thời gian ngắn, chẳng hạn như hệ số dự đoán tuyến tính cảm nhận (PLP) và hệ số tần số Mel (MFCC), do khả năng nắm bắt tính chất lặp lại và hiệu quả của tín hiệu. Nhiều nghiên cứu khác nhau đã chỉ ra hiệu quả của các tính năng MFCC trong việc xác định chính xác người nói. Tuy nhiên, hiệu suất của các tính năng này bị suy giảm trên các tập dữ liệu giọng nói phức tạp, và do đó, các tính năng này không xác định

được chính xác các đặc điểm của người nói. Để giải quyết vấn đề này, nghiên cứu này đề xuất một sự kết hợp mới giữa MFCC và các tính năng dựa trên thời gian (MFCCT), kết hợp hiệu quả của MFCC và các tính năng miền thời gian để cải thiện độ chính xác của hệ thống nhận dạng người nói không phụ thuộc vào văn bản (SI). Các tính năng MFCCT trích xuất được đưa vào làm đầu vào cho mạng nơ-ron sâu (DNN) để xây dựng mô hình nhận dạng người nói. Kết quả cho thấy rằng các tính năng MFCCT được đề xuất cùng với DNN hoạt động tốt hơn các tính năng MFCC và miền thời gian cơ sở hiện có trên tập dữ liệu LibriSpeech. Ngoài ra, DNN thu được kết quả phân loại tốt hơn so với năm thuật toán học máy đã được sử dụng gần đây trong nhận dạng người nói. Hơn nữa, nghiên cứu này đã đánh giá hiệu quả của phương pháp phân loại một cấp và hai cấp để xác định người nói. Kết quả thực nghiệm cho thấy phân loại hai cấp cho kết quả tốt hơn phân loại một cấp. Các tính năng được đề xuất và mô hình phân loại để xác định một người nói có thể được áp dụng rộng rãi cho các loại tập dữ liệu về người nói khác nhau.

Năm 2019, Yanbing và cộng sự đã công bố bài nghiên cứu "*Deep CNNs With Self-Attention for Speaker Identification*" [6]. Hầu hết các công trình hiện tại về nhận dạng người nói đều dựa trên phương pháp i-vector; tuy nhiên, có một sự thay đổi rõ rệt từ phương pháp i-vector truyền thống sang phương pháp học sâu, đặc biệt là ở dạng mạng CNN. Thay vì thiết kế các tính năng và mô hình phân loại riêng lẻ tiếp theo, nhóm tác giả giải quyết vấn đề bằng cách tìm hiểu các tính năng và hệ thống nhận dạng bằng cách sử dụng mạng nơ-ron sâu. Dựa trên CNN, bài báo này trình bày một phương pháp xác định ra định danh của người nói độc lập với văn bản mới để phân tách người nói. Cụ thể, bài báo này dựa trên hai mạng CNN tiêu biểu, được gọi là mạng nhóm hình học trực quan visual geometry group (VGG) và mạng nơ-ron dư (nets and residual neural networks – ResNets). Không giống như các phương pháp nhận dạng người nói dựa trên mạng nơ-ron sâu trước đây thường dựa trên tổng số trung bình hoặc tối đa tạm thời trên tất cả các bước thời gian để ánh xạ các phát biểu có độ dài thay đổi với một vectơ có chiều cố định, bài báo này trang bị cho hai CNN này một cơ chế tự chú ý có cấu trúc để tìm hiểu mức trung bình có trọng số qua tất cả

các bước thời gian. Sử dụng lớp tự chú ý có cấu trúc với nhiều bước chú ý, mạng CNN được đề xuất không chỉ có khả năng xử lý các phân đoạn có độ dài thay đổi mà còn có thể tìm hiểu các đặc điểm của người nói từ các khía cạnh khác nhau của chuỗi đầu vào. Kết quả thử nghiệm trên cơ sở dữ liệu điểm chuẩn nhận dạng người nói, VoxCeleb chứng minh tính ưu việt của phương pháp được đề xuất so với các phương pháp dựa trên i-vector truyền thống và các đường cơ sở khác của CNN.

Bài báo nghiên cứu [7] được đăng trên tạp chí *International Journal of Machine Learning and Computing* đã áp dụng mô hình học sâu cụ thể là sử dụng mạng CNN để xác định người nói được đề xuất. Đầu vào bằng giọng nói cho phương thức không bị hạn chế về những từ mà người nói nói. Điều đó có nghĩa là nó ở dạng độc lập với văn bản khó hơn hệ thống phụ thuộc vào văn bản. Theo phương pháp này, mỗi 2 giây giọng nói của người nói được chuyển đổi thành hình ảnh quang phổ và đầu vào cho quá trình đào tạo mô hình CNN được tạo từ đầu. Phương pháp dựa trên CNN được đề xuất được so sánh với phương pháp chiết xuất đặc trưng dựa trên hệ số MFCC cổ điển được phân loại bằng SVM. Cho đến nay, MFCC là phương pháp trích xuất tính năng phổ biến nhất cho tín hiệu âm thanh và giọng nói. Hình ảnh quang phổ được sử dụng làm đầu vào cũng được so sánh với trường hợp khi hình ảnh của sóng tín hiệu thô được sử dụng cho mô hình CNN. Thử nghiệm được thực hiện trên bài phát biểu của năm người nói bằng tiếng Thái, trong đó các giọng nói được trích xuất từ YouTube. Nó cho thấy phương pháp được CNN đề xuất đào tạo dựa trên hình ảnh quang phổ của giọng nói là tốt nhất so với hai phương pháp còn lại. Kết quả xếp loại trung bình của bài kiểm tra theo phương pháp đề xuất là 95,83%. Đối với phương pháp dựa trên MFCC là 91,26% và đối với mô hình CNN được đào tạo trên hình ảnh của sóng tín hiệu thô chỉ là 49,77%. Phương pháp được đề xuất rất hiệu quả khi chỉ sử dụng giọng nói ngắn gọn để làm đầu vào.

Bài báo "*Novel cascaded Gaussian mixture model-deep neural network classifier for speaker identification in emotional talking environments*" [8] vào năm 2018. Nghiên cứu này nhằm trình bày một cách tiếp cận hiệu quả để nâng cao hiệu suất nhận dạng người nói không phụ thuộc vào văn bản trong môi trường nói chuyện

cảm xúc dựa trên bộ phân loại mới có tên là Mô hình hỗn hợp xếp tầng Gaussian và mạng nơ-ron sâu (Cascaded Gaussian Mixture Model - Deep Neural Net) viết tắt là GMM-DNN. Kết quả chỉ ra rằng bộ phân loại được đề xuất cải thiện hiệu suất nhận dạng người nói ở các cảm xúc khác nhau bằng cách sử dụng hai cơ sở dữ liệu giọng nói riêng biệt: Cơ sở dữ liệu giọng nói của Emirati (Bộ dữ liệu tiếng Ả Rập Các Tiểu vương quốc Ả Rập Thống nhất) và bộ dữ liệu tiếng Anh "giọng nói dưới căng thẳng thực tế và mô phỏng". Bộ phân loại được đề xuất hoạt động tốt hơn các bộ phân loại cổ điển như perceptron nhiều lớp và SVM trong mỗi tập dữ liệu. Hiệu suất nhận dạng người nói đạt được dựa trên GMM-DNN xếp tầng tương tự như hiệu suất nhận được từ đánh giá chủ quan của người nghe.

Bài báo "*Towards directly modeling raw speech signal for speaker verification using cnns*" [9] lấy cảm hứng từ sự thành công của các phương pháp tiếp cận dựa trên mạng nơ-ron để tạo mô hình trực tiếp tín hiệu giọng nói thô cho các ứng dụng như nhận dạng giọng nói, nhận dạng cảm xúc và chống giả mạo, tác giả đề xuất một phương pháp xác minh người nói trong đó thông tin phân biệt người nói được học trực tiếp từ tín hiệu giọng nói bằng cách: (a) đầu tiên đào tạo hệ thống nhận dạng người nói dựa trên CNN để lấy tín hiệu giọng nói thô đầu vào và học cách phân loại trên người nói (hệ thống xác minh người nói chưa biết); và sau đó (b) xây dựng bộ dò cho mỗi người nói trong hệ thống xác minh người nói bằng cách thay thế lớp đầu ra của hệ thống nhận dạng người nói bằng hai đầu ra (người thật, người giả mạo) và điều chỉnh hệ thống theo cách phân biệt với dữ liệu lời nói của người thật và của kẻ mạo danh. Các cuộc điều tra trên cơ sở dữ liệu Voxforge cho thấy rằng cách tiếp cận này có thể mang lại cho các hệ thống khả năng cạnh tranh với các hệ thống hiện đại. Phân tích các bộ lọc trong lớp tích chập đầu tiên cho thấy rằng các bộ lọc nhấn mạnh đến thông tin ở các vùng tần số thấp (dưới 1000 Hz) và ngầm hiểu để mô hình hóa thông tin tần số cơ bản trong tín hiệu giọng nói để phân biệt người nói.

Bài báo "*An MFCC-based text-independent speaker identification system for access control*" [10] của Jung-Chun Liu và cộng sự vào năm 2017. Trong bài báo này, để bảo vệ các đối tượng trong thế giới thực, chẳng hạn như các tòa nhà, nhóm

tác giả phát triển một hệ thống nhận dạng người nói có tên là hệ số nhận dạng người nói dựa trên tần số mel (MFCC) để kiểm soát truy cập (viết tắt là MSIAC), xác định người nói bằng cách thu thập tín hiệu giọng nói của họ và chuyển đổi tín hiệu sang miền tần số. Mô hình lọc thính giác của con người dựa trên MFCC được sử dụng để điều chỉnh mức năng lượng của các tần số khác nhau. Tiếp theo, một mô hình hỗn hợp Gaussian được sử dụng để biểu diễn sự phân bố của các đặc trưng logarit dưới dạng mô hình âm học cụ thể. Ví dụ như khi một người muốn truy cập vào một đối tượng trong thế giới thực được bảo vệ bởi MSIAC, mô hình âm thanh của người đó sẽ được so sánh với mô hình âm thanh đã có. Dựa trên kết quả nhận dạng, MSIAC sẽ xác định quyền truy cập sẽ được chấp nhận hay bị từ chối. Kết quả kiểm tra hệ thống có thể thấy rằng độ chính xác nhận dạng của hệ thống được đề xuất cao hơn khi nội dung giọng nói đào tạo của hệ thống bao gồm nội dung giọng nói kiểm tra.

Năm 2017, Sarthak Yadav và công sự đã công bố bài nghiên cứu "*Learning Discriminative Features for Speaker Identification and Verification*" [11]. Trong bài báo này, nhóm tác giả đề xuất mạng CNN dựa trên CNNs Very Deep VGG [12], với những sửa đổi quan trọng để phù hợp với đầu vào quang phổ có độ dài thay đổi, giảm yêu cầu về dung lượng đĩa của mô hình và giảm số lượng thông số, dẫn đến giảm đáng kể thời gian đào tạo. Tác giả cũng đề xuất một hệ thống thống nhất cho cả nhận dạng người nói độc lập với văn bản và xác minh người nói, bằng cách đào tạo mạng được đề xuất dưới sự giám sát chung về hàm tính tổn thất Softmax (Softmax loss) và tổn thất Trung tâm (Center loss) để có được các tính năng có tính phân biệt cao phù hợp cho cả nhiệm vụ xác minh và nhận dạng người nói. Bài báo sử dụng tập dữ liệu VoxCeleb mới phát hành gần đây [13], chứa hàng trăm nghìn câu nói trong thế giới thực của hơn 1200 người nổi tiếng thuộc nhiều sắc tộc khác nhau, để đánh dấu phương pháp tiếp cận của tác giả. Mô hình CNN tốt nhất được đề xuất đạt độ chính xác là 84,6%, cải thiện tuyệt đối 4% so với phương pháp của Vox Celeb, trong khi đào tạo kết hợp với Center Loss đã cải thiện độ chính xác lên 89,5%, chứng minh tuyệt đối 9% so với cách tiếp cận của Voxceleb.

Bài báo "*Speaker identification and clustering using convolutional neural networks*" [14] của tác giả Yanick Lukic và các cộng sự. Đối với việc phân cụm người nói, người ta vẫn thường sử dụng các chuỗi xử lý thủ công như các tính năng của MFCC và các mô hình dựa trên GMM đã lỗi thời. Trong bài báo này, chúng tôi sử dụng các gam quang phổ đơn giản làm đầu vào cho CNN và nghiên cứu thiết kế tối ưu của các mạng đó để nhận dạng và phân cụm người nói. Hơn nữa, tác giả giải thích thêm về câu hỏi làm thế nào để chuyển một mạng, được đào tạo để nhận dạng người nói, sang phân cụm người nói. Nhóm tác giả đã chứng minh cách tiếp cận của mình trên tập dữ liệu TIMIT nổi tiếng, đạt được kết quả có thể so sánh với hiện đại khi sử dụng đầu ra của các lớp dày đặc mức độ cao (speaker embedding) thay vì lớp softmax (cohort modeling).

1.3. Mục tiêu, ý nghĩa khoa học và thực tiễn

Mục tiêu của đề tài là khảo sát tính khả thi của việc áp dụng các mô hình máy học hiện đại cho lĩnh vực nhận dạng người nói, kỳ vọng sẽ mang lại hiệu năng/độ chính xác cao hơn các phương pháp truyền thống. Khi mà nền tảng công nghệ được cải tiến hơn, các ứng dụng sinh trắc sẽ hấp dẫn hơn với thị trường và doanh nghiệp.

Xuất phát từ những mục tiêu chính trên, luận văn hướng tới những kết quả sau:

- Tìm hiểu tổng quan về nhận dạng người nói.

- Tìm hiểu các thuật toán trong việc nhận dạng người nói.

- Tìm hiểu và xây dựng bộ dữ liệu người nói dùng để làm đầu vào cho mô hình.

- Cài đặt thực nghiệm mạng Feedforward DNN cho nhận dạng người nói tiếng Việt.

- Trực tiếp đánh giá so sánh kết quả đạt được với mô hình truyền thống HMM trên cùng tập dữ liệu.

- Xây dựng chương trình demo.

1.4. Đối tượng và phạm vi nghiên cứu

1.4.1. Đối tượng nghiên cứu

Mô hình nhận dạng người nói tiếng Việt trong máy học, cụ thể là Deep Learning với mô hình HMM và Feedforward-DNN. Từ đối tượng nghiên cứu này, ta có các khách thể nghiên cứu khác như nhận dạng người nói, tầm quan trọng và ứng dụng của nhận dạng người nói.

1.4.2. Phạm vi nghiên cứu

Nhận dạng người nói gồm nhiều nhánh nghiên cứu khác nhau. Trong phạm vi của một luận văn cao học, đề tài tập trung vào mảng định danh người nói tiếng Việt độc lập văn bản và dữ liệu thực nghiệm là trên 40 người nói khác nhau. Cụ thể hơn, các gói công việc (WP – work package) sẽ gồm:

- WP1. Khảo sát học thuật và kiến thức nền tảng
- WP2. Thu thập dữ liệu thực nghiệm
- WP3. Xây dựng mô hình máy học
- WP4. Thực nghiệm đánh giá
- WP5. Xây dựng chương trình demo
- WP6. Viết báo cáo luận văn

1.5. Phương pháp nghiên cứu

1.5.1. Phương pháp nghiên cứu lý thuyết

- Nghiên cứu về lĩnh vực nhận dạng người nói.
- Nghiên cứu về mô hình nhận dạng người nói trong máy học.
- Tổng hợp các tài liệu liên quan đến lĩnh vực nghiên cứu: nhận dạng người nói, mạng HMM và Feedforward-DNN trong DL.
- Phân tích, thiết kế hệ thống theo quy trình sao cho dễ sử dụng, hiệu quả, dễ nâng cấp, sửa chữa bổ sung.

1.5.2. Phương pháp nghiên cứu thực nghiệm

- Nghiên cứu về bộ dữ liệu và cách xây dựng bộ dữ liệu cho đề tài nhận dạng người nói;
- Nghiên cứu cách xử lý bộ dữ liệu, áp dụng bộ dữ liệu vào mô hình dự đoán;
- Cài đặt, huấn luyện, thử nghiệm mô hình HMM và Feedforward DNN cho nhận dạng người nói tiếng Việt;
- Đánh giá, so sánh hiệu năng giữa phương pháp máy học hiện đại (DNN) so với phương pháp truyền thống (HMM).

1.6. Bố cục luận văn

Chương 1: Phần mở đầu

Chương 2: Tổng quan đề tài

Chương 3: Nhận dạng người nói với Deep Learning

Chương 4: Thực nghiệm

Chương 5: Kết luận

CHƯƠNG 2: TỔNG QUAN ĐỀ TÀI

2.1. Giới thiệu chung

2.1.1. Nhận dạng người nói là gì?

Mức độ quan tâm và nhu cầu sử dụng các phần mềm nhận dạng giọng nói tăng lên chóng mặt trong những năm gần đây. Không chỉ vậy, công nghệ giọng nói liên tục được cải tiến và ngày càng trở nên tinh vi hơn. Do đó các doanh nghiệp nên đầu tư nhiều hơn vào việc triển khai và tích hợp công nghệ này để tăng phạm vi tiếp cận cũng như cải thiện doanh số của mình.

Nhận dạng giọng nói gồm 2 thuật ngữ: Voice recognition và Speech recognition.

- **Speech recognition** chỉ nhận dạng các từ ngữ và tập trung vào việc dịch từ ngữ đó thành văn bản.
- **Voice recognition** có khả năng nhận dạng và định danh giọng nói của từng người dùng (hay còn gọi là **nhận dạng người nói**). Cụ thể là xác thực người nói bằng cách phân tích các mẫu và trình tự giọng nói của một người từ đó có thể định danh người nói chính xác.

Với những tiềm năng ứng dụng rộng rãi của nhận dạng sinh trắc học nói chung và nhận dạng người nói nói riêng, lĩnh vực nhận dạng người nói đã được đi sâu nghiên cứu trong nhiều thập kỷ qua và cũng đã đạt được nhiều thành tựu đáng kể. Về cơ bản, hệ thống nhận dạng người qua giọng nói cũng tuân thủ các bước của một hệ thống nhận dạng dựa trên sinh trắc học. Tuy nhiên, cũng cần có những nét đặc trưng chuyên sâu nhằm tăng cường kết quả nhận dạng. Chẳng hạn ở giai đoạn rút trích đặc trưng, tùy ứng dụng mà chúng ta nên xem xét những đặc trưng nào cần được rút trích sao cho phù hợp và đạt hiệu quả cao.

Nhận dạng người qua giọng nói là một trong những nhánh được nghiên cứu phát triển mạnh trong sinh trắc học, bởi lẽ như ta đã biết trong các đặc tính sinh học trên cơ thể người, tiếng nói là một đặc điểm mang tính phổ thông, dễ phát sinh và không cần đến các thiết bị thu phức tạp. Nhiều công trình đã được nghiên cứu trên

tiếng nói nhằm khai thác các thông tin từ lĩnh vực này. Cụ thể hơn, nhận dạng người nói (speaker recognition) [2] bao gồm 2 loại là: nhận dạng độc lập văn bản (text-independent) và nhận dạng phụ thuộc văn bản (text-dependent).

Text-dependent with fixed passwords

Phương pháp dựa trên văn bản yêu cầu người nói cung cấp các từ hoặc câu chính để sử dụng cho cả quá trình đào tạo và nhận dạng. Các phương pháp này thường dựa trên kỹ thuật đối sánh mẫu/mô hình-trình tự trong đó chiều thời gian của mẫu giọng nói đầu vào và các mẫu tham chiếu được căn chỉnh và sự tương đồng giữa chúng sẽ được hệ thống tích lũy trên từng câu phát âm để làm căn cứ ra quyết định. Vì có thể khai thác đặc tính biến đổi ít của tiếng nói theo từng âm vị hoặc âm tiết, nên phương pháp dựa trên văn bản thường đạt được độ chính xác nhận dạng cao hơn so với phương pháp không phụ thuộc văn bản.

Một số kỹ thuật phổ biến truyền thống của phương pháp text-dependent là DTW (Dynamic Time Warping – quy hoạch thời gian động) hoặc sử dụng mô hình HMM (Hidden Markov Model – mô hình Markov ẩn).

Text-independent with no specific passwords

Phương pháp text-independent không dựa vào một văn bản được định nghĩa trước cụ thể nào. Do đó, ưu điểm của phương pháp này là nó có thể nhận ra người nói độc lập với nội dung của câu phát âm. Vì rất khó có thể mô hình hóa hoặc so khớp các mẫu tiếng nói ở cấp độ từ hoặc câu, nên đối với phương pháp không dựa trên văn bản, các kỹ thuật mô hình hóa toàn câu phát âm sử dụng phương pháp thống kê thường được sử dụng.

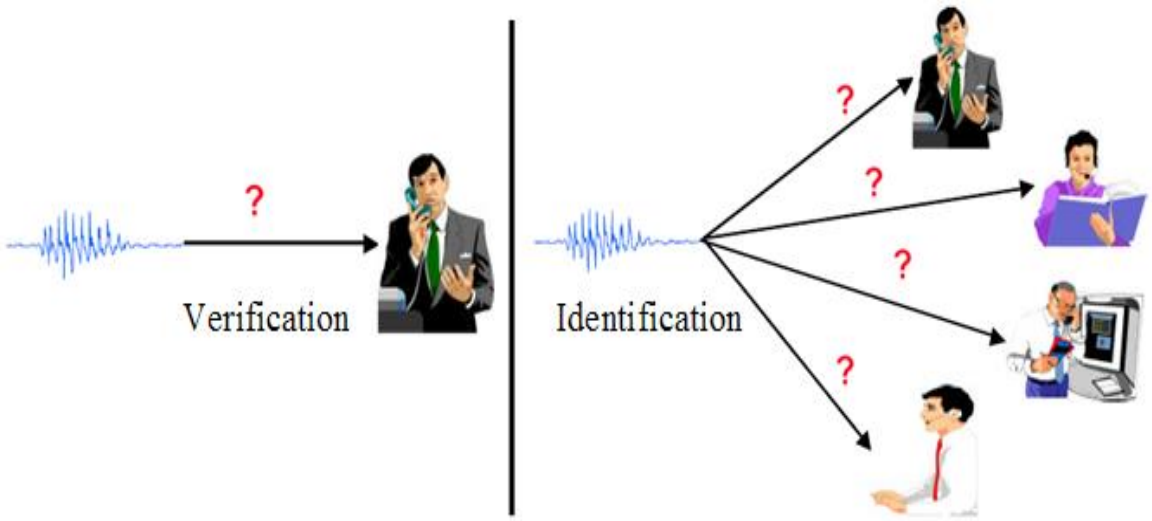
Text-Prompted Speaker Recognition

Tuy nhiên, cả hai phương pháp text-dependent và text-independent kể trên đều lộ rõ điểm yếu, đặc biệt trong bối cảnh các công nghệ ghi âm, tái tạo giọng nói phát triển mạnh. Do đó, để khắc phục các hạn chế của những phương pháp truyền thống, các nhà khoa học đề xuất một cách tiếp cận mới có tên Text-Prompted Speaker Recognition. Đối với phương pháp này, các câu nói mang tính "chìa khóa" được thay đổi liên tục trong mỗi lần truy cập. Hệ thống chỉ chấp nhận lời nói đầu vào khi xác

định rằng chính người được cấp quyền đang thực hiện câu lệnh do máy cung cấp. Phương pháp này không chỉ nhận dạng chính xác người nói mà còn giúp phòng tránh trường hợp một giọng nói được ghi âm và phát lại.

Phương pháp Text-Prompted Speaker Recognition sử dụng các mô hình âm vị tương ứng với từng người nói làm đơn vị âm thanh cơ bản. Các mô hình âm vị có thể được biểu diễn bằng các mô hình thống kê như Gaussian-mixture, tied-mixture HMMs, hoặc Deep learning models. Trong giai đoạn nhận dạng nếu sử dụng phương pháp tied-mixture HMMs, hệ thống sẽ ghép các mô hình âm vị của từng người nói đã đăng ký để tạo ra một chuỗi các HMM dựa theo văn bản được nhắc. Sau đó, điểm của chuỗi này sẽ được hệ thống tính toán và sử dụng để xác minh người nói.

Theo thể thức (Hình 2.1) thì nhận dạng người nói gồm dạng xác minh (verification) và dạng định danh (identification).



Hình 2.1: Phân biệt xác minh và định danh *(nguồn [2])*

Trong bài toán xác minh người nói (verification), chương trình sẽ đối sánh xem mẫu lời nói cần kiểm tra có so khớp với một người nói đã cho trước hay không (so sánh 1:1) và kết quả trả về là kết luận mẫu lời nói kiểm tra đúng hoặc sai (không thuộc) so với người nói đã biết trước. Còn đối với bài toán định danh người nói (identification), chương trình sẽ phải đối sánh mẫu lời nói với tất cả dữ liệu lưu trữ (so sánh 1:n) và kết quả trả về là kết luận mẫu kiểm tra là của một trong số những

người đã lưu trữ hay là một người giả mạo từ bên ngoài. Có một sự khác nhau lớn trong độ phức tạp tính toán giữa hai loại trên. Trong xác minh người nói, hệ thống chỉ cần kiểm tra mô hình người nói (speaker) quan tâm (mô hình người nói cho trước) mà không đòi hỏi kiểm tra những mô hình người nói khác. Tuy nhiên, trong những ứng dụng thực tế, chỉ đơn giản xem xét một mô hình sẽ không có đủ thông tin nhận xét để đưa ra quyết định tốt. Bởi vì hệ thống sẽ không có sự đối sánh rõ ràng nếu chỉ xem xét chỉ riêng một mô hình.

Trên thế giới có thể kể ra một số hệ thống nhận dạng người nói điển hình được các công ty phát triển như:

- Hammacher Schlemmer (Mỹ) - giới thiệu đến người tiêu dùng mẫu USB đầu tiên trên thế giới có công nghệ bảo mật bằng giọng nói.
- VoiceVault cung cấp các giải pháp sinh trắc học qua giọng nói trên thiết bị và trên ứng dụng điện thoại đi động.
- Nuance Communications Gatekeeper cũng là một ví dụ cho công nghệ nhận diện giọng nói. Ứng dụng này có khả năng xác định giọng nói của người lớn tuổi. Từ đó, nhân viên nhận cuộc gọi sẽ đưa họ vào danh sách ưu tiên trong đại dịch Covid-19.
- Siri, Alexa hay Google Assistant cũng ứng dụng công nghệ nhận diện giọng nói. Đây còn là giải pháp rất hiệu quả để phân khúc người tiêu dùng trong kinh doanh.
- Hãng điện thoại đi động OPPO cũng đã tích hợp khả năng khoá máy bằng giọng nói lên hệ điều hành dựa trên nền tảng Android ColorOS 2.0 của mình.

2.1.2. Ứng dụng công nghệ nhận dạng người nói vào đời sống

Các công nghệ nhận dạng người nói được sử dụng trong các lĩnh vực ứng dụng rộng rãi [15]. Các lĩnh vực mà các kỹ thuật nhận dạng người nói có thể được sử dụng này là xác thực, giám sát và nhận dạng trong pháp y. Tùy thuộc vào những điều này, lĩnh vực nhận dạng người nói một lần nữa được chia thành ba loại cụ thể: nhận dạng,

phát hiện / xác minh và phân đoạn và phân cụm. Một số ứng dụng ví dụ của công nghệ nhận dạng người nói như

Xác thực

Nhận dạng người nói để xác thực cho phép người dùng xác định người sử dụng giọng nói của họ. Một người có thể được xác định bằng nhiều đặc điểm khác nhau như chữ ký, dấu vân tay, giọng nói, đặc điểm khuôn mặt, v.v… Loại phương pháp xác thực này được gọi là xác thực người sinh trắc học. Trong trường hợp này, khả năng bị lạm dụng các loại vấn đề nhận dạng này ít hơn so với việc chìa khóa hoặc thẻ tín dụng có thể bị đánh cắp hoặc bị mất, do đó, số PIN hoặc mật khẩu có thể dễ dàng bị sử dụng sai hoặc quên. Mỗi người có đặc điểm giải phẫu, sinh lý và những thói quen học được mà những người thân quen sử dụng trong cuộc sống hàng ngày để nhận ra người đó. Điều này có thể thuận tiện hơn nhiều so với các phương tiện xác thực truyền thống yêu cầu mang theo chìa khóa bên mình hoặc ghi nhớ mã PIN.

Giám sát

Cơ quan an ninh có một số phương tiện thu thập thông tin. Một trong số đó là nghe lén điện tử các cuộc trò chuyện qua điện thoại và radio. Vì những điều này dẫn đến số lượng lớn dữ liệu, cơ chế lọc phải được áp dụng để tìm thông tin liên quan. Một trong những bộ lọc này có thể là nhận dạng những người nói có quan tâm đến dịch vụ.

Pháp y

Đây là một ứng dụng quan trọng của nhận dạng người nói. Nếu có một mẫu lời nói đã được ghi lại trong quá trình phạm tội. Giọng nói của nghi phạm có thể được so sánh với mẫu giọng nói để đưa ra dấu hiệu về sự giống nhau của hai giọng nói. Chứng minh danh tính của giọng nói được ghi âm có thể giúp kết tội tội phạm hoặc xử một người vô tội trước tòa. Mặc dù tác vụ này có thể không được thực hiện bởi hệ thống nhận dạng người nói tự động, tuy nhiên, các kỹ thuật xử lý tín hiệu có thể được sử dụng trong lĩnh vực này.

Bảo mật

Nó là ứng dụng rõ ràng nhất của bất kỳ kỹ thuật xác thực sinh trắc học nào. Nhận dạng người nói có thể được sử dụng trong các giao dịch thẻ tín dụng như một phương pháp xác thực kết hợp với một số phương thức khác như nhận dạng khuôn mặt. Công nghệ nhận dạng người nói có thể cung cấp cơ sở xác thực giao dịch hoặc kiểm soát truy cập máy tính, giám sát, xác thực giọng nói qua điện thoại để gọi đường dài hoặc truy cập ngân hàng, v.v…

Nhận dạng giọng nói

Nhận dạng giọng nói và người nói là lĩnh vực nghiên cứu kép theo nghĩa rằng khả năng thay đổi của người nói là một trong những vấn đề chính trong nhận dạng giọng nói, trong khi nhận dạng người nói đó là một lợi thế. Công nghệ nhận dạng người nói có thể được sử dụng để giảm sự biến đổi của người nói trong hệ thống nhận dạng giọng nói bằng sự thích nghi của người nói. Ví dụ, hệ thống nhận dạng giọng nói có thể có một bộ phận định vị người nói nhận biết ai đang nói. Sau đó, hệ thống có thể điều chỉnh các thông số của trình nhận dạng giọng nói để phù hợp hơn với người nói hiện tại hoặc để chọn trình nhận dạng giọng nói phụ thuộc vào người nói từ cơ sở dữ liệu của nó.

Theo dõi nhiều người nói

Trong phần này, một số người nói được bao gồm trong bản ghi âm. Ngoài ra, người ta cũng muốn biết ai đang phát biểu trong hội nghị từ xa, đặc biệt là khi có nhiều người tham dự trong cuộc họp qua điện thoại và những người tham dự không quen thuộc với nhau. Ba loại của tác vụ đa người nói khác nhau được nhận dạng - phát hiện người nói, theo dõi người nói và phân đoạn người nói. Nhiệm vụ phát hiện bao gồm việc quyết định xem một người nói đã biết có xuất hiện trong bản ghi nhiều người nói hay không. Trong nhiệm vụ theo dõi, khoảng thời gian nói của một người nói nhất định được đặt trong bản ghi âm. Nhiệm vụ phân đoạn bao gồm việc xác định các khoảng giọng nói của mỗi người nói khác nhau. Trong trường hợp chung nhất, có thể không có kiến thức trước về người nói hoặc số lượng của họ. Các ứng dụng

của phân đoạn người nói đã được đề xuất để phân đoạn các chương trình phát sóng tin tức.

Giao diện người dùng được cá nhân hóa

Chẳng hạn như hộp thư thoại ngày càng trở nên phổ biến hơn do sự phát triển của công nghệ giọng nói nói chung. Bằng cách nhận dạng người nói, hệ thống có thể thích ứng với nhu cầu và sở thích của họ. Các ứng dụng trên yêu cầu các kỹ thuật nhận dạng người nói mạnh mẽ, ví dụ: trong các dịch vụ hỗ trợ qua điện thoại, người dùng có thể gọi đến trong các loại điều kiện âm thanh khác nhau như trong văn phòng, trên đường phố, v.v. và sử dụng các mạng điện thoại khác nhau như điện thoại cố định hoặc di động. Trong các tình huống cuộc họp, những người tham gia có thể nói chuyện trong khi di chuyển xung quanh đối mặt với micrô theo các hướng khác nhau và khoảng cách khác nhau. Các điều kiện không khớp có thể gặp phải bất cứ lúc nào trong những trường hợp này. Do đó, tính mạnh mẽ là một trong những yếu tố quan trọng quyết định sự thành công của tính năng nhận dạng người nói trong các ứng dụng này

2.1.3. Tổng quan về trí tuệ nhân tạo (AI)

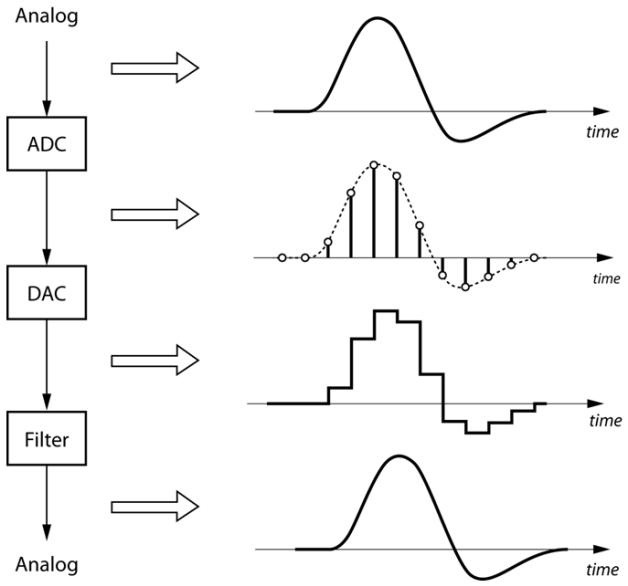
Trí tuệ nhân tạo (AI) [16] đã trở nên rất phổ biến trong thế giới ngày nay. Trí tuệ nhân tạo là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người (hình 2.2). Trí tuệ nhân tạo khác với việc lập trình logic trong các ngôn ngữ lập trình là ở việc ứng dụng các hệ thống học máy để mô phỏng trí tuệ của con người trong các xử lý mà con người làm tốt hơn máy tính.

Hình 2.2: Trí tuệ nhân tạo – AI *(nguồn [16])*

Cùng với sự phát triển xã hội hiện đại như ngày nay, AI đã không còn là một từ ngữ xa lạ. Công nghệ ngày càng lớn mạnh, AI càng đi sâu vào nhiều lĩnh vực của đời sống con người chẳng hạn như sức khỏe, kinh doanh, giáo dục, sản xuất,... Amazon Go là một minh chứng rõ ràng nhất của AI, khi khách hàng không cần phải xếp hàng mà chỉ cần đăng nhập vào tài khoản Amazon Go, lấy các sản phẩm cần thiết và rời đi. Các thao tác thanh toán, gửi hóa đơn được thực hiện một cách chính xác, tự động và không cần phải nhờ đến sự trợ giúp của con người. Với tốc độ phát triển cao như vậy thì trong tương lai AI chắc chắn sẽ đạt được nhiều thành tựu hơn nữa.

2.2. Tín hiệu giọng nói

Tín hiệu âm thanh là một đại diện của âm thanh. Nó mã hóa tất cả các thông tin cần thiết cần thiết để tái tạo âm thanh. Tín hiệu âm thanh có hai loại cơ bản: analog và digital. Hình 2.3 mô tả việc xử lý tín hiệu analog.



Hình 2.3: Lấy mẫu và số hóa tín hiệu analog, sau đó tái tạo lại tín hiệu này *[17]*

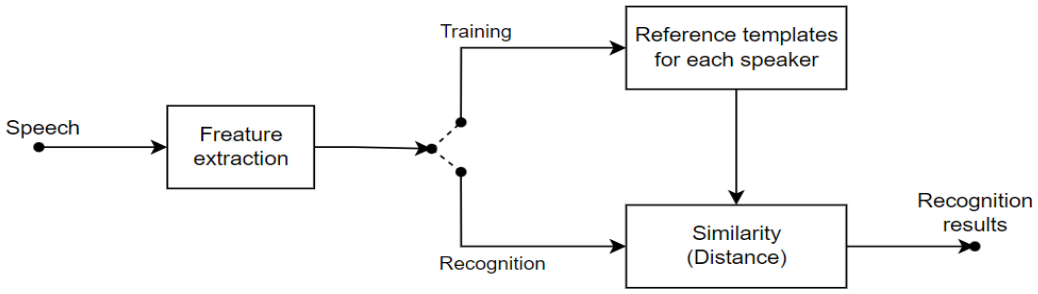
Analog là âm thanh được ghi lại bằng cách sử dụng các phương pháp tái tạo sóng âm thanh gốc. Ví dụ bao gồm các bản ghi vinyl và băng cassette. Âm thanh digital được ghi lại bằng cách lấy các mẫu của sóng âm thanh gốc ở một tốc độ xác

định, được gọi là tốc độ lấy mẫu (sampling rate). Đĩa CD và tệp MP3 là những ví dụ về các định dạng kỹ thuật số.

Trong thế giới thực, việc chuyển đổi giữa các dạng sóng digital và analog là phổ biến và cần thiết. ADC và DAC là một phần của quá trình xử lý tín hiệu âm thanh và chúng đạt được những chuyển đổi này.

2.3. Các thành phần chính của hệ thống nhận dạng người nói

Hệ thống nhận dạng người nói thường bao gồm ba đơn vị chính [18] như hình dưới đây. Đầu vào cho giai đoạn đầu tiên hoặc cho hệ thống xử lý đầu cuối là tín hiệu giọng nói. Tại đây giọng nói được số hóa và sau đó việc rút trích đặc trưng sẽ diễn ra. Không có tính năng độc quyền nào truyền tải danh tính của người nói trong tín hiệu giọng nói, tuy nhiên, theo lý thuyết bộ lọc nguồn của quá trình tạo ra giọng nói, hình dạng phổ giọng nói được mã hóa trong đó thông tin về hình dạng đường âm của người nói thông qua công thức và nguồn âm thanh qua sóng hài cao độ. Do đó, một số dạng hoặc dạng khác của các đặc trưng dựa trên quang phổ được sử dụng trong hầu hết các hệ thống nhận dạng người nói. Quá trình cuối cùng trong giai đoạn xử lý đầu cuối là một số hình thức bù kênh. Các thiết bị đầu vào khác nhau (ví dụ: các thiết bị cầm tay điện thoại khác nhau) áp đặt các đặc điểm phổ khác nhau lên tín hiệu giọng nói, chẳng hạn như giới hạn và định hình băng tần. Do đó bù kênh được thực hiện để loại bỏ các tác dụng không mong muốn này. Thông thường nhất, một số dạng bù kênh tuyến tính, chẳng hạn như phép trừ trung bình cộng dài hạn và ngắn hạn được áp dụng cho các tính năng. Cơ bản của phép trừ quang phổ là năng lượng quang phổ của tín hiệu lời nói bị nhiễu bởi tiếng ồn bằng tổng năng lượng quang phổ của tín hiệu và nhiễu.



Hình 2.4: Cấu trúc của hệ thống nhận dạng người nói *(nguồn [19])*

Quá trình nhận dạng người nói bao gồm giai đoạn đào tạo và giai đoạn nhận dạng (hình 2.4). Trong giai đoạn đào tạo, các đặc điểm trong tín hiệu lời nói của người nói được lưu trữ dưới dạng các đặc trưng tham chiếu. Các vectơ đặc trưng của lời nói được sử dụng để tạo mô hình người nói. Số lượng mẫu tham chiếu cần thiết để nhận dạng người nói hiệu quả tùy thuộc vào loại tính năng hoặc kỹ thuật mà hệ thống sử dụng để nhận dạng người nói. Trong giai đoạn nhận dạng, các đặc trưng tương tự như các đặc trưng được sử dụng trong mẫu tham chiếu được trích xuất từ câu nói đầu vào của người nói với danh tính được yêu cầu phải được xác định. Quyết định nhận dạng phụ thuộc vào khoảng cách được tính toán giữa mẫu tham chiếu và mẫu được tạo ra từ lời nói đầu vào. Trong nhận dạng người nói, khoảng cách giữa lời nói đầu vào và tất cả các mẫu tham chiếu có sẵn đều được tính toán. Mẫu của người dùng đã đăng ký, có khoảng cách với mẫu câu đầu vào là nhỏ nhất, cuối cùng được chọn làm người phát biểu của câu đầu vào. Trong trường hợp xác minh người nói, khoảng cách chỉ được tính giữa lời nói đầu vào và mẫu tham chiếu của người nói được xác nhận quyền sở hữu. Nếu khoảng cách nhỏ hơn ngưỡng xác định trước, người nói được chấp nhận, người nói khác bị từ chối với tư cách là kẻ mạo danh.

2.4. Rút trích đặc trưng

2.4.1. Rút trích đặc trưng là gì

Trích xuất đặc trưng âm thanh [18], [20] là một bước cần thiết trong xử lý tín hiệu âm thanh, là một trường con của quá trình xử lý tín hiệu. Nó liên quan đến việc xử lý hoặc thao tác các tín hiệu âm thanh, loại bỏ tiếng ồn không mong muốn và cân bằng các dải tần số thời gian bằng cách chuyển đổi tín hiệu kỹ thuật số và tín hiệu analog. Nó tập trung vào các phương pháp tính toán để thay đổi âm thanh, biến đổi tín hiệu âm thanh thô thành một biểu diễn nhỏ gọn. Một chuỗi các vectơ đặc trưng biểu thị tín hiệu giọng nói nhỏ gọn sẽ được tính toán bằng phương pháp rút trích. Các vectơ đặc trưng được trích xuất từ tín hiệu thô, trong mô-đun trích xuất đặc trưng nhấn mạnh các thuộc tính cụ thể của người nói và loại bỏ dư thừa trong thống kê. Sử dụng vectơ đặc trưng của người nói cụ thể để đào tạo mô hình người nói, khai thác

tính năng liên quan đến việc đơn giản hóa lượng tài nguyên cần thiết để mô tả chính xác một tập hợp lớn dữ liệu. Khi thực hiện phân tích dữ liệu phức tạp, một trong những vấn đề lớn bắt nguồn từ số lượng các biến liên quan. Trích xuất đặc trưng là một thuật ngữ chung để chỉ các phương pháp xây dựng tổ hợp các biến để giải quyết các vấn đề này trong khi vẫn mô tả dữ liệu với độ chính xác đầy đủ.

Trong nhận dạng người nói, độ chính xác và tỷ lệ nhận dạng suy giảm do các khía cạnh khác nhau như: sự thay đổi từ người nói; lời nói do người nói cung cấp (có thể thay đổi bất cứ lúc nào vì cảm xúc và bệnh tật). Ngoài ra, sự thay đổi từ môi trường, tiếng ồn trong tín hiệu giọng nói (do kênh truyền), tiếng ồn nền và độ vang làm hỏng tín hiệu giọng nói đầu vào ở chế độ thử nghiệm. Đặc trưng về thời gian không hiệu quả vì nó thay đổi đáng kể khi cùng một người nói cùng một giọng. Các đặc trưng sẽ cung cấp thông tin chính xác và chống nhiễu tốt nên được tính toán trong những trường hợp như vậy.

2.4.2. Các đặc trưng âm thanh phổ biến cho việc thiết lập mô hình

Đặc trưng của âm thanh là mô tả về âm thanh hoặc tín hiệu âm thanh về cơ bản có thể được đưa vào các mô hình thống kê hoặc ML để xây dựng hệ thống âm thanh thông minh. Các ứng dụng âm thanh sử dụng các đặc trưng này bao gồm phân loại âm thanh, nhận dạng giọng nói, gắn thẻ nhạc tự động, phân đoạn âm thanh và tách nguồn, lấy dấu vân tay âm thanh, làm giảm âm thanh, truy xuất thông tin âm nhạc, v.v…

Các đặc trưng khác nhau sẽ bắt lấy các khía cạnh khác nhau của âm thanh. Nói chung, các đặc trưng âm thanh được phân loại theo các khía cạnh sau:

a. Mức độ trừu tượng

Các danh mục thuộc phân loại này chủ yếu bao gồm các tín hiệu âm nhạc hơn là âm thanh nói chung

- **High-level**: Đây là những đặc điểm trừu tượng được con người hiểu và tận hưởng. Chúng bao gồm nhạc cụ, phím, hợp âm, giai điệu, hòa âm, nhịp điệu, thể loại, tâm trạng, v.v...

- **Mid-level**: Đây là những đặc điểm mà chúng ta có thể cảm nhận được. Chúng bao gồm cao độ, bộ mô tả liên quan đến nhịp, khởi đầu nốt, mẫu dao động, MFCC, v.v… Có thể nói rằng đây là tổng hợp các đặc trưng cấp thấp.
- **Low-level**: Đây là các đặc trưng thống kê được trích xuất từ âm thanh. Những điều này có ý nghĩa đối với máy móc, nhưng không có ý nghĩa đối với con người. Các ví dụ bao gồm đường bao biên độ, năng lượng, tâm quang phổ, thông lượng quang phổ, tốc độ xuyên 0, v.v...

b. Phạm vi tạm thời

Loại của phân loại này áp dụng cho âm thanh nói chung, nghĩa là cả âm nhạc và không âm nhạc

- **Tức thời (Instantaneous)**: Như tên cho thấy, các đặc trưng này cung cấp cho chúng ta thông tin tức thời về tín hiệu âm thanh. Chúng coi là những phần nhỏ của tín hiệu âm thanh, trong phạm vi mili giây. Độ phân giải thời gian tối thiểu mà con người có thể đánh giá cao là khoảng 10ms.
- **Cấp độ phân đoạn (Segment-level):** Các đặc trưng này có thể được tính toán từ các phân đoạn của tín hiệu âm thanh trong phạm vi giây.
- **Toàn cầu (global):** Đây là các đặc trưng tổng hợp cung cấp thông tin và mô tả toàn bộ âm thanh.

c. Khía cạnh âm nhạc: Thuộc tính âm thanh bao gồm nhịp, nhịp điệu, âm sắc (màu sắc của âm thanh), cao độ, hòa âm, giai điệu, v.v…

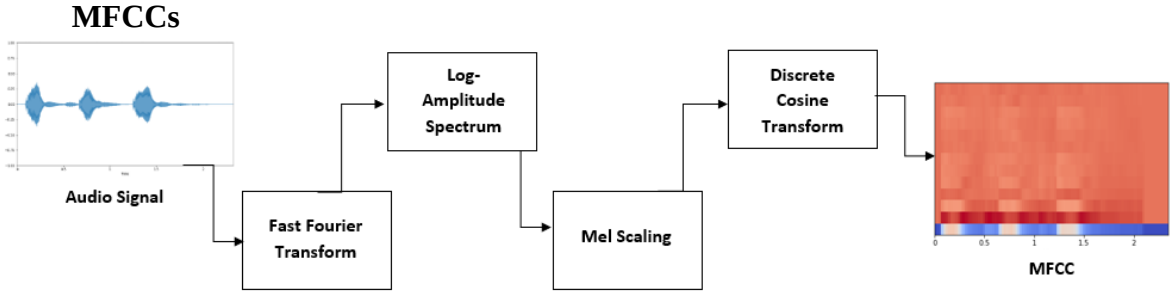
d. Miền tín hiệu (Signal domain)

- **Miền thời gian (time-domain)**: Chúng được trích xuất từ các dạng sóng của âm thanh thô. Các ví dụ về tỷ lệ giao nhau bằng không, đường bao biên độ và năng lượng RMS.
- **Miền tần số (frequency-domain)**: Chúng tập trung vào các thành phần tần số của tín hiệu âm thanh. Các tín hiệu thường được chuyển đổi từ miền thời gian sang miền tần số bằng cách sử dụng Fourier Transform [21]. Tỉ lệ năng lượng dải, tâm quang phổ, và thông lượng quang phổ là những ví dụ.

- **Biểu diễn thời gian-tần số (time-frequency representation)**: Các đặc trưng này kết hợp cả thành phần thời gian và tần số của tín hiệu âm thanh. Biểu diễn thời gian-tần số thu được bằng cách áp dụng Short-Time Fourier Transform (STFT) trên dạng sóng miền thời gian. Quang phổ (Spectrogram), quang phổ mel (mel-spectrogram) và hằng số biến đổi-Q (constant-Q transform) là những ví dụ.

e. Phương pháp tiếp cận ML

- **Học máy truyền thống**: coi tất cả hoặc hầu hết các đặc trưng từ cả miền thời gian và tần số là đầu vào của mô hình. Các đặc trưng cần được chọn lọc thủ công dựa trên ảnh hưởng của nó đối với hiệu suất của mô hình. Một số đặc trưng được sử dụng rộng rãi bao gồm Amplitude Envelope, Zero-Crossing Rate (ZCR), Năng lượng Root Mean Square (RMS), Trung tâm phổ, Tỷ lệ năng lượng băng tần và Băng thông phổ.
- **Học sâu**: xem xét các biểu diễn âm thanh không có cấu trúc như quang phổ hoặc MFCC. Nó tự trích xuất các mẫu. Vào cuối những năm 2010, đây đã trở thành cách tiếp cận được ưa thích vì việc trích xuất tính năng là tự động. Nó cũng được hỗ trợ bởi lượng dữ liệu dồi dào và sức mạnh tính toán.



Hình 2.5: Các bước trích xuất MFCC từ tín hiệu âm thanh *(nguồn [18])*

Từ hình 2.5 đã trình bày thông tin về tốc độ thay đổi trong các dải phổ của một tín hiệu được đưa ra bởi cepstrum của nó. Cepstrum về cơ bản là một phổ của nhật ký về phổ của tín hiệu thời gian. Phổ kết quả không nằm trong miền tần số cũng không nằm trong miền thời gian do đó nó được đặt tên là miền quefrency (đảo chữ cái của từ tần số). Cepstrum truyền tải các giá trị khác nhau cấu tạo nên các chất tạo

thành (một thành phần đặc trưng của chất lượng âm thanh lời nói) và âm sắc của âm thanh. Vì vậy, MFCC rất hữu ích cho các mô hình học sâu.

2.5. Mô hình máy học

2.5.1. Khái niệm về máy học

Học máy (Machine Learning) [22] là một tập hợp con của AI cung cấp cho máy tính khả năng tự động học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể. Do đó, học máy là một phương pháp giúp máy móc giải quyết vấn đề bằng cách đạt được khả năng suy nghĩ.

ML được áp dụng vào nhiều phương diện khác theo trong các lĩnh vực của đời sống (hình 2.6). Dưới đây trình bày ứng dụng phổ biến nhất trong thế giới thực của ML [23]:

a. Nhận dạng hình ảnh

Nhận dạng hình ảnh là một trong những ứng dụng phổ biến nhất của ML. Nó được sử dụng để xác định đối tượng, người, địa điểm, hình ảnh kỹ thuật số, v.v… Trường hợp sử dụng phổ biến của nhận dạng hình ảnh và nhận diện khuôn mặt chẳng hạn như là gợi ý gắn thẻ bạn bè tự động của Facebook. Facebook cung cấp tính năng gợi ý tự động gắn thẻ bạn bè. Bất cứ khi nào tải lên một bức ảnh với bạn bè trên Facebook, sẽ tự động nhận được đề xuất gắn thẻ với tên và công nghệ đằng sau điều này là thuật toán nhận dạng và phát hiện khuôn mặt của ML.

b. Nhận dạng giọng nói

Trong khi sử dụng Google, ta nhận được tùy chọn "Tìm kiếm bằng giọng nói", tùy chọn này có tính năng nhận dạng giọng nói và là một ứng dụng phổ biến của ML. Nhận dạng giọng nói là một quá trình chuyển đổi hướng dẫn bằng giọng nói thành văn bản và nó còn được gọi là "Lời nói thành văn bản" hoặc "Nhận dạng giọng nói của máy tính". Hiện tại, các thuật toán học máy được sử dụng rộng rãi bởi các ứng dụng khác nhau của nhận dạng giọng nói. Trợ lý Google, Siri, Cortana và Alexa đang sử dụng công nghệ nhận dạng giọng nói để làm theo hướng dẫn bằng giọng nói.

c. Dự đoán giao thông

Ví dụ điển hình của dự đoán giao thông là Google Maps giúp định vị đường đi chính xác với tuyến đường ngắn nhất và dự đoán tình trạng giao thông. Nó dự đoán các điều kiện giao thông như liệu giao thông đã thông thoáng, di chuyển chậm hay tắc nghẽn nặng với sự trợ giúp của hai cách: Vị trí thời gian thực của hình thức phương tiện ứng dụng Google Map và các cảm biến; Thời gian trung bình đã diễn ra vào những ngày qua cùng một lúc. Tất cả những người đang sử dụng Google Map đều giúp ứng dụng này trở nên tốt hơn. Nó lấy thông tin từ người dùng và gửi trở lại cơ sở dữ liệu của nó để cải thiện hiệu suất.

d. Đề xuất sản phẩm

ML được sử dụng rộng rãi bởi các công ty giải trí và thương mại điện tử khác nhau như Amazon, Netflix, v.v… để giới thiệu sản phẩm cho người dùng. Bất cứ khi nào ta tìm kiếm một sản phẩm nào đó trên Amazon, thì quảng cáo nhận được cho cùng một sản phẩm trong khi lướt Internet trên cùng một trình duyệt và điều này là do ML. Google hiểu sở thích của người dùng bằng cách sử dụng các thuật toán học máy khác nhau và đề xuất sản phẩm theo sở thích của khách hàng. Tương tự như vậy, khi sử dụng Netflix, tìm thấy một số đề xuất cho loạt phim giải trí, phim, v.v. và điều này cũng được thực hiện với sự trợ giúp của ML.

e. Ô-tô tự động lái

Một trong những ứng dụng thú vị nhất của ML là ô tô tự lái. ML đóng một vai trò quan trọng trong ô tô tự lái. Tesla, công ty sản xuất xe hơi phổ biến nhất đang nghiên cứu về xe tự lái. Nó đang sử dụng phương pháp học không giám sát để đào tạo các mô hình ô tô để phát hiện người và đồ vật trong khi lái xe.

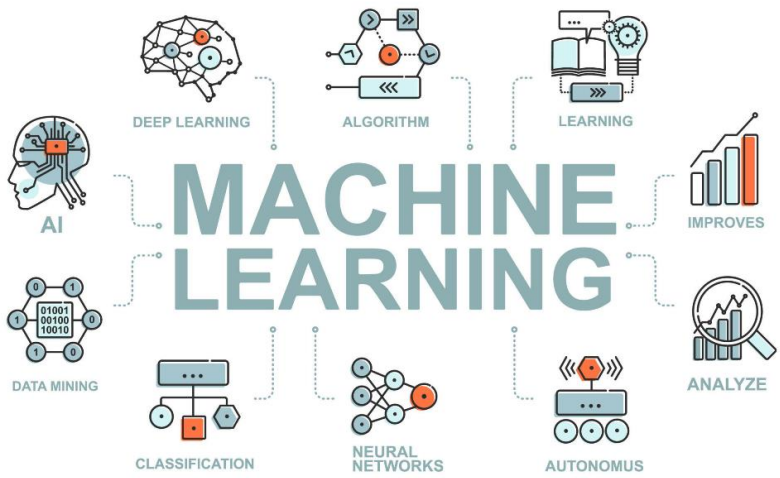
f. Spam email và lọc các phần mềm độc hại

Bất cứ khi nào nhận được một email mới, email đó sẽ được tự động lọc thành email quan trọng, bình thường và thư rác. Ta có thể nhận được một thư quan trọng trong hộp thư đến của mình với biểu tượng quan trọng và các email spam trong hộp thư rác và công nghệ đằng sau điều này là Máy học. Dưới đây là một số bộ lọc thư rác được Gmail sử dụng: Bộ lọc nội dung; Bộ lọc tiêu đề; Bộ lọc danh sách đen chung; Bộ lọc dựa trên quy tắc; Bộ lọc phân quyền. Một số thuật toán học máy như Multi-

Layer Perceptron, Cây quyết định và trình phân loại Naïve Bayes được sử dụng để lọc thư rác email và phát hiện phần mềm độc hại.

g. Trợ lý ảo

Hiện nay có nhiều trợ lý ảo cá nhân khác nhau như trợ lý Google, Alexa, Cortana, Siri. Như tên cho thấy, chúng giúp tìm kiếm thông tin bằng cách sử dụng hướng dẫn bằng giọng nói. Những trợ lý này có thể giúp ta theo nhiều cách khác nhau chỉ bằng hướng dẫn bằng giọng nói, chẳng hạn như phát nhạc, gọi cho ai đó, mở email, lên lịch cuộc hẹn, v.v… Các trợ lý ảo này sử dụng các thuật toán ML như một phần quan trọng. Các trợ lý này ghi lại các hướng dẫn bằng giọng nói, gửi nó qua máy chủ trên một đám mây và giải mã nó bằng các thuật toán ML và hành động theo đó.



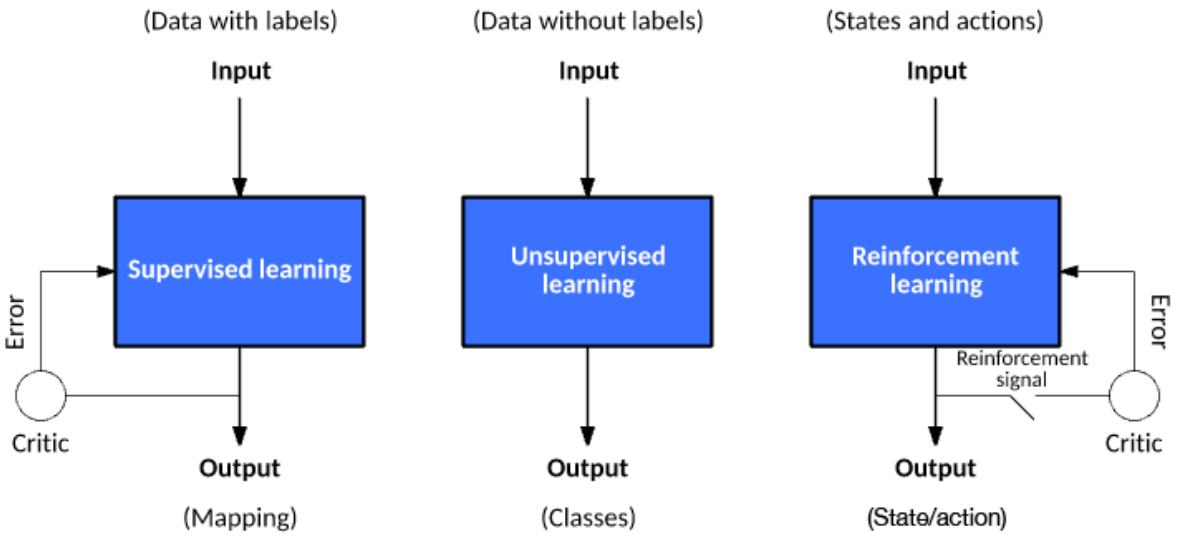
Hình 2.6: Các lĩnh vực ứng dụng của Machine Learning *(nguồn [24])*

Các thuật toán được sử dụng trong học máy được chia thành ba loại: học có giám sát (supervised learning), không giám sát (unsupervised learning) và học tăng cường (reinforcement learning). Học tập có giám sát bao gồm phản hồi để chỉ ra khi nào một dự đoán là đúng hay sai, trong khi học không giám sát không có phản hồi: thuật toán chỉ đơn giản là cố gắng phân loại dữ liệu dựa trên cấu trúc ẩn của nó. Học

tập củng cố tương tự như học tập có giám sát ở chỗ nó nhận được phản hồi, nhưng không nhất thiết đối với từng đầu vào hoặc từng trạng thái.

2.5.2. Các loại mô hình máy học

Các thuật toán học máy không ngừng lớn mạnh và phát triển. Tuy nhiên, trong hầu hết các trường hợp, các thuật toán có xu hướng chuyển thành một trong ba mô hình cho việc học tập (hình 2.7). Các mô hình tồn tại để tự động điều chỉnh theo một cách nào đó nhằm cải thiện hoạt động hoặc hành vi của chúng.



Hình 2.7: Ba mô hình học tập cho các thuật toán *(nguồn [25])*

Trong học có giám sát, tập dữ liệu bao gồm các đầu ra (hoặc nhãn) mong muốn của nó để một hàm có thể tính toán lỗi cho một dự đoán nhất định. Việc giám sát được thực hiện khi một dự đoán được đưa ra và một lỗi được tạo ra (thực tế so với mong muốn) để thay đổi chức năng và tìm hiểu ánh xạ.

Trong học tập không có giám sát, tập dữ liệu không bao gồm đầu ra mong muốn; do đó, không có cách nào để giám sát chức năng. Thay vào đó, hàm cố gắng phân đoạn tập dữ liệu thành các "lớp" để mỗi lớp chứa một phần của tập dữ liệu với các tính năng chung.

Cuối cùng, trong học tập củng cố, thuật toán cố gắng học các hành động cho một tập hợp các trạng thái nhất định dẫn đến trạng thái mục tiêu. Lỗi không được đưa ra sau mỗi ví dụ (như trường hợp học có giám sát) mà thay vào đó là khi nhận được

tín hiệu củng cố (chẳng hạn như đạt được trạng thái mục tiêu). Hành vi này tương tự như hoạt động học tập của con người, trong đó phản hồi không nhất thiết phải được cung cấp cho tất cả các hành động nhưng khi phần thưởng được đảm bảo.

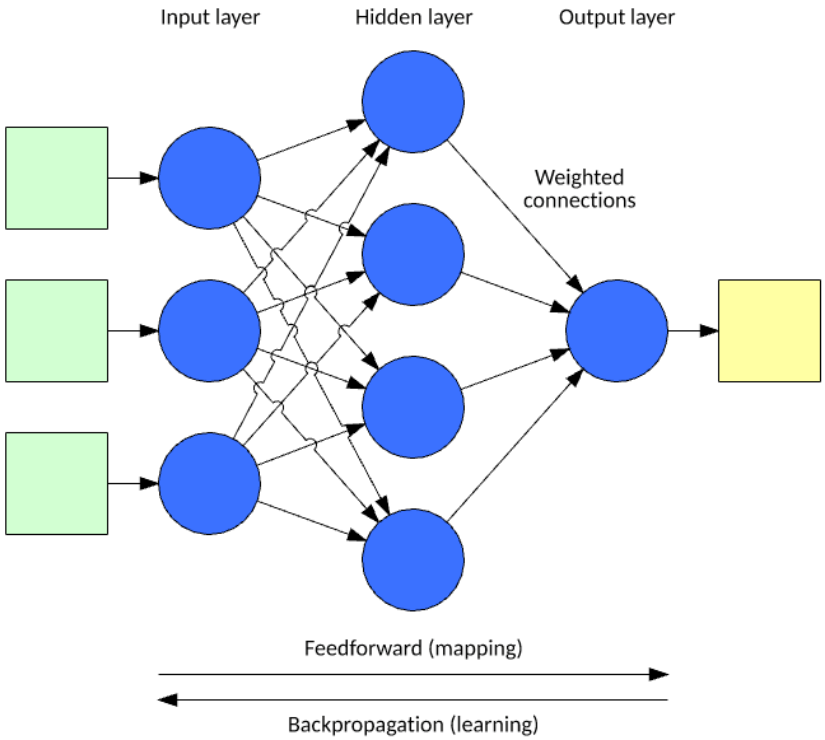
Tùy vào dữ liệu đầu vào và cách tiếp cận vấn đề mà ta có thể sử dụng các thuật toán khác nhau hoặc kết hợp chúng để tạo nên mô hình phù hợp với bài toán.

CHƯƠNG 3: NHẬN DẠNG NGƯỜI NÓI VỚI DEEP LEARNING

3.1. Mạng nơ-ron và deep learning

3.1.1. Mạng nơ-ron

Neural network [26] hay còn gọi là mạng nơ-ron được xây dựng dựa trên mạng nơ-ron sinh học. Nó là một mạng lưới gồm các nút được kết nối với nhau – gọi là nơ-ron và các cạnh nối chúng lại với nhau. Một mạng nơ-ron xử lý một vectơ đầu vào thành một vectơ đầu ra kết quả thông qua một mô hình lấy cảm hứng từ các nơ-ron và khả năng kết nối của chúng trong não. Mô hình bao gồm các lớp tế bào thần kinh được kết nối với nhau thông qua các trọng số làm thay đổi tầm quan trọng của một số đầu vào nhất định so với những đầu vào khác. Mạng nơ-ron là một mạng có cấu trúc và nhiều lớp (layer). Một mạng nơ-ron có 3 lớp chính là: input, hidden và output. (hình 3.1)



Hình 3.1: Các lớp của một mạng nơ-ron điển hình *(nguồn [26])*

Mỗi nơ-ron bao gồm một hàm kích hoạt xác định đầu ra của nơ-ron (như một hàm của vectơ đầu vào nhân với vectơ trọng số của nó). Đầu ra được tính toán bằng cách áp dụng vectơ đầu vào cho lớp đầu vào của mạng, sau đó tính toán đầu ra của mỗi nơ-ron thông qua mạng (theo kiểu chuyển tiếp). Trọng số làm tăng hoặc giảm cường độ của tín hiệu tại một cạnh. Các nơ-ron có thể có ngưỡng sao cho tín hiệu được gửi đi chỉ khi tín hiệu tổng hợp vượt qua ngưỡng đó, khi đó các output của lớp này sẽ là input của lớp phía sau. Thông qua việc lặp lại các bước trên, mạng nơron học thông qua nhiều lớp và các nơ-ron rồi sau đó kết hợp lại ở lớp cuối cùng để cho ra một dự đoán.

Một số hàm kích hoạt được sử dụng phổ biến:

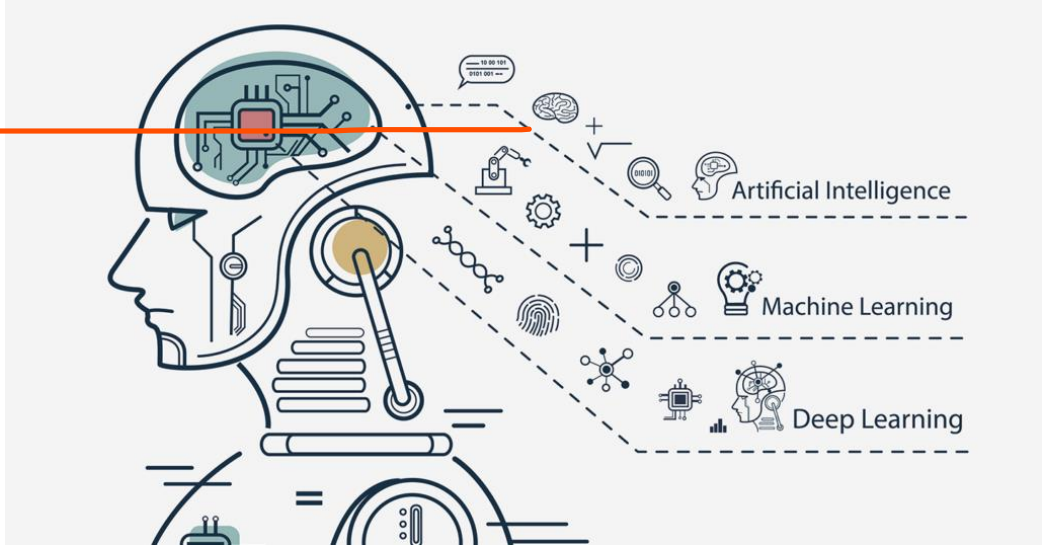
- Binary step: $f(x) = 1, x \geq 0$
- Linear: $f(x) = ax$
- Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$
- Tanh: $tanh(x) = \frac{2}{1+e^{-2x}} - 1$
- ReLU: $f(x) = \max(0, x)$
- Leaky ReLU: $f(x) = \begin{cases} x \; với \; i \; x > 0 \\ ax \; ngược \; lại \end{cases}$
- Softmax: $a(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \; với \; j = 1, \ldots, k$

3.1.2. Deep learning

Khi mà khả năng tính toán của các máy tính ngày càng được nâng lên một tầm cao mới và lượng dữ liệu ngày càng khổng lồ, Machine Learning đã tiến thêm một bước dài và một lĩnh vực mới được ra đời gọi là Deep Learning (hình 3.2 mô tả mối quan hệ giữa AI, ML và DL).

Là một phạm trù nhỏ của ML, DL tập trung giải quyết các vấn đề liên quan đến mạng thần kinh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, tầm nhìn máy tính và xử lý ngôn ngữ tự nhiên. Chỉ trong thời gian ngắn, Deep Learning đã giúp máy tính làm được rất nhiều công việc phức tạp như: chỉnh màu cho ảnh đen trắng, thêm âm thanh vào phim câm, dịch máy tự động, phân loại các đối

tượng trong ảnh, tạo chữ viết tay tự động, tạo phụ đề cho hình ảnh,... Ngoài ra còn rất nhiều lĩnh vực khác đã được đơn giản hóa và nâng cao hiệu quả hoạt động với sự trợ giúp của DL.

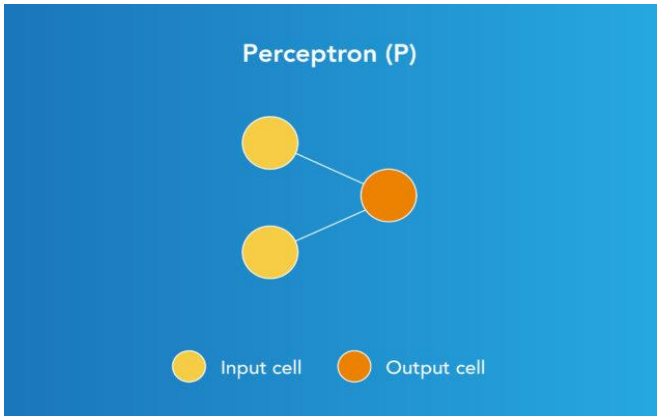


Hình 3.2: Mối liên hệ giữa AI, ML và DL *(nguồn [27])*

3.2. Phân loại / các dạng mạng neural nhân tạo

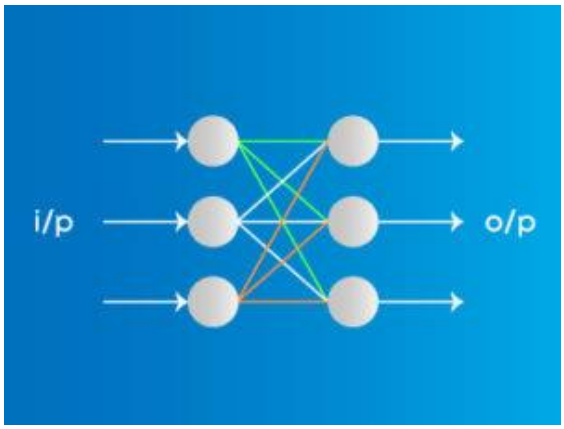
Có nhiều loại mạng nơ-ron có sẵn hoặc có thể đang trong giai đoạn phát triển [28]. Chúng có thể được phân loại tùy thuộc vào: cấu trúc; luồng dữ liệu; tế bào thần kinh được sử dụng và mật độ của chúng; lớp và bộ lọc kích hoạt độ sâu của chúng, v.v…

- **Perceptron**: Perceptron là một thuật toán học có giám sát phân loại dữ liệu thành hai loại, do đó nó là một bộ phân loại nhị phân. Perceptron có thể triển khai các Cổng logic như AND, OR hoặc NAND, tuy nhiên mạng này có nhược điểm là chỉ có thể học các bài toán phân tách tuyến tính như bài toán boolean AND. Đối với các bài toán phi tuyến tính như bài toán boolean XOR, nó không hoạt động. Hình dưới đây mô tả cấu trúc của một perceptron.



Hình 3.3: Perceptron *(nguồn [28])*

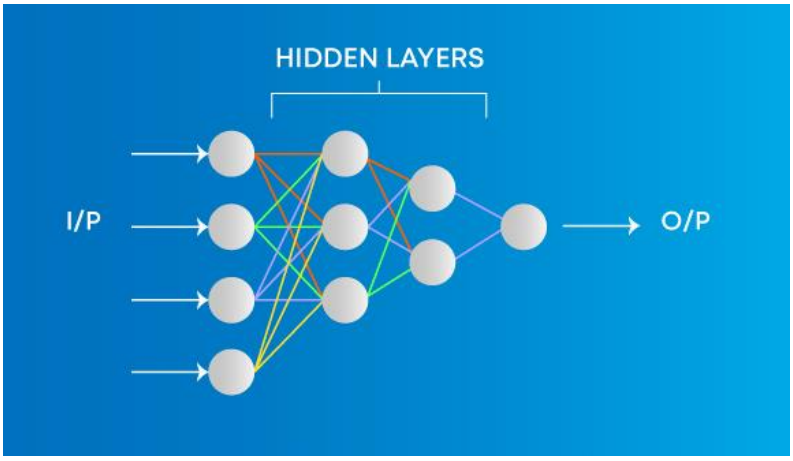
- **Feed Forward Neural Network**: Dạng mạng nơ-ron đơn giản nhất trong đó dữ liệu đầu vào chỉ truyền theo một hướng, đi qua các nút thần kinh nhân tạo và thoát ra qua các nút đầu ra. Mạng này được ứng dụng trong dạng bài phân loại đơn giản, nhận dạng khuôn mặt, thị giác máy tính và nhận dạng giọng nói. Một số đặc điểm của mạng là ít phức tạp, dễ thiết kế và bảo trì; nhanh chóng và tốc độ cao; đáp ứng cao với dữ liệu nhiễu. Tuy nhiên, không sử dụng được cho học sâu. Cấu trúc của một Feed Forward Neural Network được mô tả qua hình dưới đây.



Hình 3.4: Feed Forward Neural Networks *(nguồn [28])*

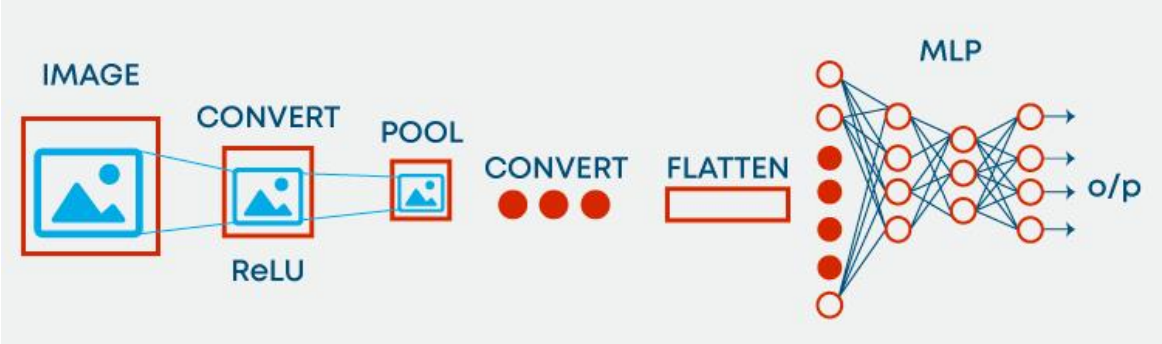
- **Multilayer Perceptron**: Một điểm vào hướng tới mạng thần kinh phức tạp, nơi dữ liệu đầu vào truyền qua các lớp tế bào thần kinh nhân tạo khác nhau. Mỗi nút đơn được kết nối với tất cả các nơ-ron trong lớp tiếp theo, điều này làm cho nó trở thành một mạng nơ-ron được kết nối hoàn chỉnh. Các lớp đầu

vào và đầu ra hiện có nhiều lớp ẩn, tức là tổng số ít nhất ba lớp trở lên (hình 3.5). Nó có sự lan truyền hai chiều, tức là sự lan truyền tiến và lan truyền ngược. Mạng được ứng dụng trong dạng bài phân loại phức tạp, dịch máy và nhận dạng giọng nói. Mạng được sử dụng cho học sâu vì thế nó tương đối phức tạp để thiết kế và bảo trì.



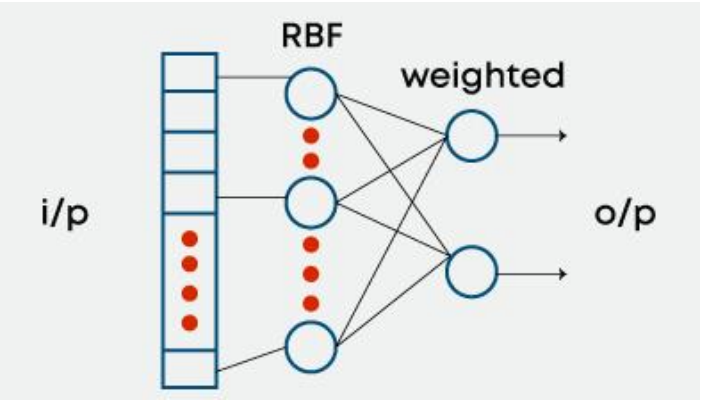
Hình 3.5: Multilayer Perceptron *(nguồn [28])*

- **Convolutional Neural Network**: Mạng CNN (hình 3.6) chứa một sự sắp xếp ba chiều của các nơ-ron, thay vì mảng hai chiều tiêu chuẩn. Lớp đầu tiên được gọi là lớp chập. Mỗi nơ-ron trong lớp chập chỉ xử lý thông tin từ một phần nhỏ của trường thị giác. Các tính năng đầu vào được thực hiện theo lô giống như một bộ lọc. CNN hiểu các hình ảnh theo từng phần và có thể tính toán các hoạt động này nhiều lần để hoàn thành quá trình xử lý hình ảnh đầy đủ. Quá trình xử lý bao gồm việc chuyển đổi hình ảnh từ thang RGB hoặc HSI sang thang xám. Việc chia nhỏ các thay đổi trong giá trị pixel sẽ giúp phát hiện các cạnh và hình ảnh có thể được phân loại thành các loại khác nhau. CNN được ứng dụng trong dạng bài xử lý hình ảnh, thị giác máy tính và nhận dạng giọng nói và dịch máy. Một số đặc điểm của CNN là sử dụng để học sâu với ít tham số, cần ít tham số để học tập hơn so với lớp được kết nối đầy đủ. Bên cạnh đó, CNN cũng có các nhược điểm tương tự Multilayer Perceptron.



Hình 3.6: Convolutional Neural Network *(nguồn [28])*

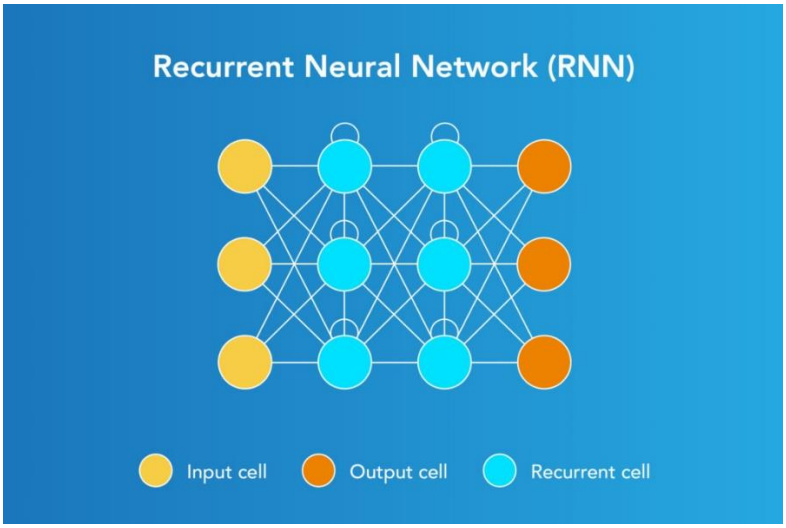
- **Radial Basis Functional Neural Network**: mạng bao gồm một vectơ đầu vào theo sau là một lớp nơ-ron RBF và một lớp đầu ra với một nút cho mỗi danh mục (hình 3.7). Việc phân loại được thực hiện bằng cách đo mức độ tương tự của đầu vào với các điểm dữ liệu từ tập huấn luyện trong đó mỗi nơ-ron lưu trữ một nguyên mẫu. Đây sẽ là một trong những ví dụ từ tập huấn luyện. Khi một vectơ đầu vào mới cần được phân loại, mỗi nơ-ron sẽ tính toán khoảng cách Euclide giữa đầu vào và nguyên mẫu của nó. Ví dụ: nếu chúng ta có hai lớp, tức là lớp A và lớp B, thì đầu vào mới được phân loại gần với nguyên mẫu lớp A hơn là nguyên mẫu lớp B. Do đó, nó có thể được gắn thẻ hoặc phân loại là loại A.



Hình 3.7: Radial Basis Function Neural Networks *(nguồn [28])*

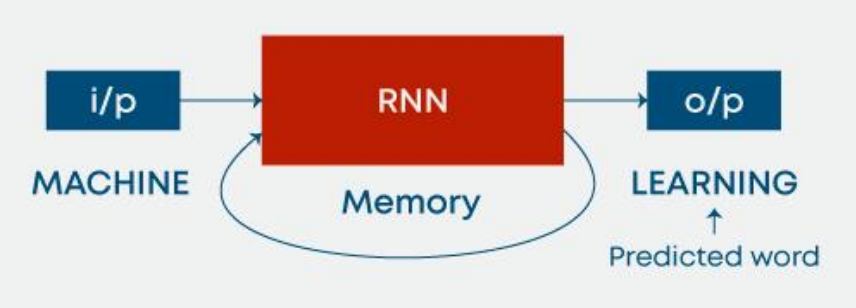
- **Recurrent Neural Network**: Được thiết kế để lưu đầu ra của một lớp, RNN được đưa trở lại đầu vào để giúp dự đoán kết quả của lớp. Lớp đầu tiên thường là một mạng nơ-ron feed forward, tiếp theo là lớp RNN nơi một số thông tin

mà nó có trong bước thời gian trước đó được ghi nhớ bởi một chức năng bộ nhớ (hình 3.8). Quá trình truyền chuyển tiếp được thực hiện trong trường hợp này. Nó lưu trữ thông tin cần thiết để sử dụng trong tương lai. Nếu dự đoán sai, tỷ lệ học tập được sử dụng để thực hiện những thay đổi nhỏ. Do đó, làm cho nó tăng dần theo hướng đưa ra dự đoán đúng trong quá trình lan truyền ngược. Một số ứng dụng của RNN: Xử lý văn bản như tự động đề xuất, kiểm tra ngữ pháp, v.v.; Xử lý văn bản thành giọng nói; Trình gắn thẻ hình ảnh; Phân tích cảm xúc; Dịch.



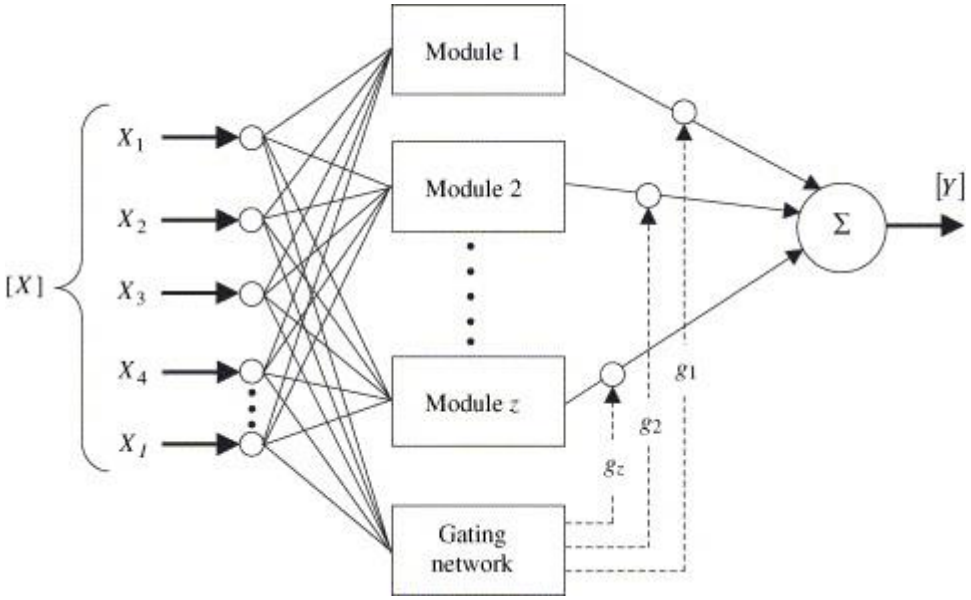
Hình 3.8: Recurrent Neural Networks *(nguồn [28])*

- **LSTM – Long Short-Term Memory**: Mạng LSTM là một loại RNN sử dụng các đơn vị đặc biệt ngoài các đơn vị tiêu chuẩn. Các đơn vị LSTM bao gồm một 'ô nhớ' có thể duy trì thông tin trong bộ nhớ trong thời gian dài. Một tập hợp các cổng được sử dụng để kiểm soát thời điểm thông tin đi vào bộ nhớ khi nó xuất ra và khi nào nó bị quên. Có ba loại cổng, cổng vào, cổng ra và cổng quên. Cổng đầu vào quyết định có bao nhiêu thông tin từ mẫu cuối cùng sẽ được lưu trong bộ nhớ; cổng đầu ra điều chỉnh lượng dữ liệu được truyền đến lớp tiếp theo, và cổng quên kiểm soát tốc độ xé của bộ nhớ được lưu trữ. Kiến trúc này cho phép họ học các phụ thuộc lâu dài hơn. Cấu trúc của mạng LSTM được mô tả qua hình dưới đây



Hình 3.9: Long Short-Term Memory *(nguồn [28])*

- **Sequence to Sequence Models (seq2seq)**: Mô hình seq2seq bao gồm hai mạng RNN. Ở đây, tồn tại một bộ mã hóa xử lý đầu vào và một bộ giải mã xử lý đầu ra. Bộ mã hóa và bộ giải mã hoạt động đồng thời - sử dụng cùng một thông số hoặc các thông số khác nhau. Mô hình này, trái ngược với RNN thực tế, đặc biệt áp dụng trong những trường hợp độ dài của dữ liệu đầu vào bằng độ dài của dữ liệu đầu ra. Mặc dù chúng có những lợi ích và hạn chế tương tự như RNN, nhưng những mô hình này thường được áp dụng chủ yếu trong chatbots, máy dịch và hệ thống trả lời câu hỏi.
- **Modular Neural Network**: là mạng có một số mạng khác nhau hoạt động độc lập và thực hiện các nhiệm vụ con (hình 3.10). Các mạng khác nhau không thực sự tương tác hoặc báo hiệu cho nhau trong quá trình tính toán. Chúng làm việc độc lập để đạt được kết quả đầu ra. Kết quả là, một quá trình tính toán lớn và phức tạp được thực hiện nhanh hơn đáng kể bằng cách chia nhỏ nó thành các thành phần độc lập. Tốc độ tính toán tăng do các mạng không tương tác hoặc thậm chí không kết nối với nhau. Ứng dụng của mạng này là Hệ thống dự đoán thị trường chứng khoán, Adaptive MNN để nhận dạng ký tự, nén dữ liệu đầu vào mức cao.



Hình 3.10: Modular Neural Network *(nguồn [28])*

3.3. Nhận dạng người nói

Với mục tiêu nhận dạng người nói, luận văn sẽ xây dựng các mô hình bao gồm HMM và Feedforward-DNN để đánh giá mức độ hiệu quả của máy học hiện đại so với máy học truyền thống.

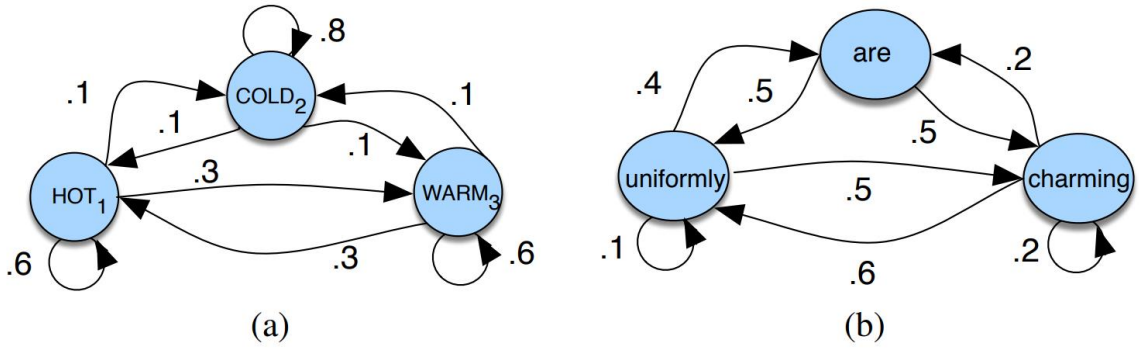
3.3.1. Nhận dạng người nói với HMM

Mô hình Markov ẩn (HMM) [29] là một mô hình thống kê cũng được sử dụng trong học máy. Nó có thể được sử dụng để mô tả diễn biến của các sự kiện có thể quan sát được phụ thuộc vào các yếu tố bên trong mà không thể quan sát trực tiếp được. Đây là một loại mô hình đồ họa xác suất cho phép chúng ta dự đoán một chuỗi các biến chưa biết từ một tập hợp các biến quan sát.

Markov Chains

HMM dựa trên việc tăng cường chuỗi Markov. Chuỗi Markov là một mô hình cho biết điều gì đó về xác suất của chuỗi các biến ngẫu nhiên, các trạng thái, mỗi biến ngẫu nhiên có thể nhận các giá trị từ một số tập hợp. Những tập hợp này có thể là các từ, hoặc thẻ, hoặc biểu tượng đại diện cho bất kỳ thứ gì, như thời tiết. Chuỗi Markov đưa ra một giả định rất mạnh mẽ rằng nếu muốn dự đoán tương lai trong chuỗi, tất cả những gì quan trọng là trạng thái hiện tại. Các trạng thái trước trạng thái hiện tại

không có tác động đến tương lai ngoại trừ thông qua trạng thái hiện tại. Như thể để dự đoán thời tiết ngày mai, có thể kiểm tra thời tiết của ngày hôm nay nhưng không được phép xem thời tiết của ngày hôm qua. (Ví dụ trong hình 3.11).



Hình 3.11: Ví dụ về dự đoán thời tiết *(nguồn [29])*

Hình 3.11 Với một chuỗi Markov cho thời tiết (a) và cho các từ (b), hiển thị các trạng thái và quá trình chuyển đổi. Một phân phối bắt đầu π là bắt buộc; đặt π = [0,1, 0,7, 0,2] cho (a) có nghĩa là xác suất 0,7 bắt đầu ở trạng thái 2 (cold), xác suất 0,1 bắt đầu ở trạng thái 1 (hot), v.v… Cụ thể hơn, hãy xem xét một chuỗi các biến trạng thái q1, q2, ..., qi. Một mô hình Markov thể hiện giả định Markov về các xác suất của chuỗi này: rằng khi dự đoán tương lai, quá khứ không quan trọng, chỉ hiện tại.

Giả thuyết Markov: $P(q_i = a|q_1 \dots q_{i-1}) = P(q_i = a|q_{i-1})$

Hình 3.11a cho thấy một chuỗi Markov để gán xác suất cho một chuỗi các sự kiện thời tiết, trong đó từ vựng bao gồm HOT, COLD và WARM, như là các cạnh. Chuyển tiếp là xác suất: giá trị của các cung rời khỏi trạng thái cho trước phải có tổng bằng 1. Hình 3.11b cho thấy một chuỗi Markov để gán xác suất cho một chuỗi các từ $w_1 \dots w_n$. Trên thực tế Markov chuỗi này đại diện cho một mô hình ngôn ngữ bigram, với mỗi cạnh biểu thị xác suất p (w_i | w_j)! Với hai mô hình trong Hình 3.11, có thể gán một xác suất cho bất kỳ chuỗi nào từ vốn từ vựng. Thông thường, một chuỗi Markov được chỉ định bởi các thành phần sau:

$Q = q_1 q_2 \dots q_N$ một tập hợp của N các trạng thái

$A = a_{11}a_{12} \ldots a_{n1} \ldots a_{nn}$	một ma trận xác suất chuyển đổi A, mỗi a_{ij} biểu diễn xác suất chuyển từ trạng thái i sang trạng thái j, $\sum_{j=1}^{n} a_{ij} = 1 \; \forall i$
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	một phân phối xác suất ban đầu trên các trạng thái. π_i là xác suất để chuỗi Markov bắt đầu ở trạng thái i. Một số trạng thái j có thể có $\pi_j = 0$, nghĩa là chúng không thể là trạng thái ban đầu. Ngoài ra, $\sum_{i=1}^{N} \pi_i = 1$

HMM

Chuỗi Markov rất hữu ích khi cần tính xác suất cho một chuỗi các sự kiện có thể quan sát được. Tuy nhiên, trong nhiều trường hợp, các sự kiện quan tâm bị ẩn đi. Ví dụ: ta thường không quan sát việc gán nhãn từ loại trong một văn bản. Thay vào đó, ta nhìn thấy các từ và phải suy ra các nhãn từ chuỗi từ. Ta gọi các nhãn là ẩn bởi vì chúng không được quan sát thấy.

HMM cho phép nói về cả các sự kiện được quan sát (như các từ nhìn thấy trong đầu vào) và các sự kiện ẩn (như gán nhãn từ loại) mà ta coi là các yếu tố nhân quả trong mô hình xác suất. HMM được chỉ định bởi các thành phần sau:

$Q = q_1 q_2 \ldots q_N$	một tập hợp của N các trạng thái
$A = a_{11}a_{12} \ldots a_{n1} \ldots a_{nn}$	một ma trận xác suất chuyển đổi A, mỗi a_{ij} biểu diễn xác suất chuyển từ trạng thái i sang trạng thái j, $\sum_{j=1}^{n} a_{ij} = 1 \; \forall i$
$O = o_1 o_2 \ldots o_T$	một chuỗi T quan sát, mỗi quan sát được rút ra từ một từ vựng $V = v_1, v_2, \ldots, v_V$
$B = b_i(o_t)$	một chuỗi các khả năng quan sát, còn được gọi là xác suất phát xạ, mỗi thể hiện xác suất của một quan sát o_t được tạo ra từ trạng thái i
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	một phân phối xác suất ban đầu trên các trạng thái. π_i là xác suất để chuỗi Markov bắt đầu ở trạng thái i. Một số

trạng thái j có thể có $\pi_j = 0$, nghĩa là chúng không thể là trạng thái ban đầu. Ngoài ra, $\sum_{i=1}^{N} \pi_i = 1$

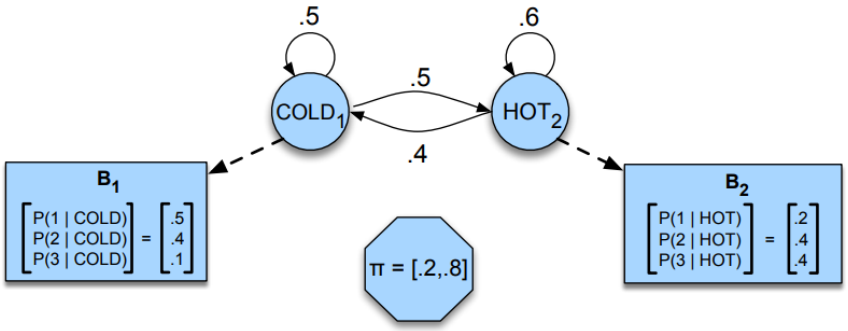
Mô hình Markov ẩn bậc nhất tạo ra hai giả thiết đơn giản hóa. Đầu tiên, như với chuỗi Markov bậc nhất, xác suất của một trạng thái cụ thể chỉ phụ thuộc vào trạng thái trước đó:

Giả thuyết Markov: $P(q_i = a|q_1 \dots q_{i-1}) = P(q_i = a|q_{i-1})$

Thứ hai, xác suất của một quan sát đầu ra o_i chỉ phụ thuộc vào trạng thái tạo ra q_i quan sát chứ không phụ thuộc vào bất kỳ trạng thái nào khác hoặc bất kỳ quan sát nào khác:

Đầu ra: $P(o_i|q_1 \dots q_i, \dots, q_T, o_1, \dots, o_i, \dots, o_T) = P(o_i|q_i)$

Để minh họa những mô hình này, sử dụng một nhiệm vụ do Jason Eisner phát minh. Mục tiêu là sử dụng những quan sát từ nhật ký của Jason Eisner, liệt kê số lượng kem Jason đã ăn để ước tính nhiệt độ mỗi ngày. Đơn giản hóa nhiệm vụ thời tiết này bằng cách giả sử chỉ có hai loại ngày: lạnh (C) và nóng (H). Vì vậy, nhiệm vụ của Eisner như sau: Cho một chuỗi quan sát O (mỗi số nguyên đại diện cho số lượng kem đã ăn trong một ngày nhất định) tìm chuỗi 'ẩn' Q của trạng thái thời tiết (H hoặc C) khiến Jason ăn kem. Hình 3.12 cho thấy một HMM mẫu cho nhiệm vụ ăn kem. Hai trạng thái ẩn (H và C) tương ứng với thời tiết nóng và lạnh, và các quan sát tương ứng với số lượng kem mà Jason đã ăn trong một ngày nhất định.



Hình 3.12: Một mô hình Markov ẩn *(nguồn [29])*

Với mỗi mô hình Markov ẩn có ba vấn đề chính cần được xem xét:

Vấn đề 1: Tính toán độ tương đồng (Computing likelihood) : cho mô hình λ(A,B,π) và chuỗi quan sát được O xác định độ tương đồng (likelihood) P(O|λ). Ví

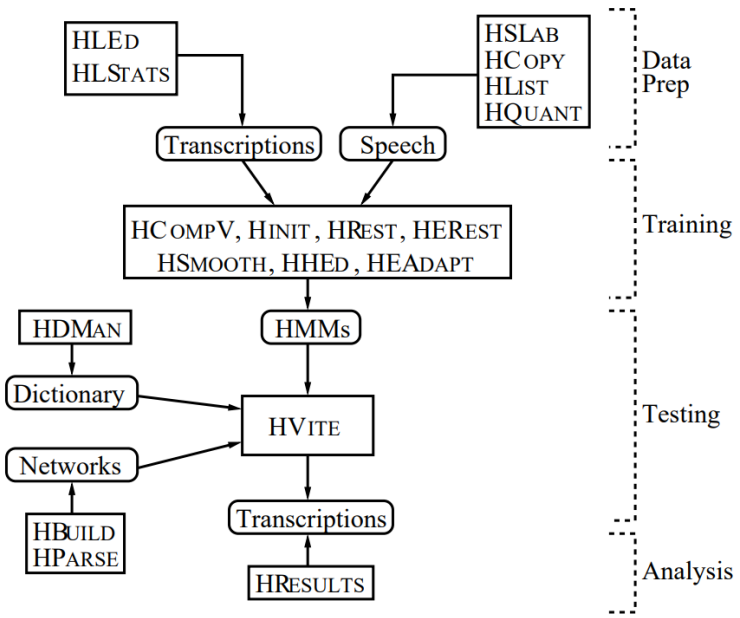
dụ trong nhận dạng tiếng nói, ta có quan sát O là tín hiệu tiếng nói và λ là mô hình, vậy bài toán cần giải là tính độ tương đồng P để mô hình λ quan sát được O.

Vấn đề 2 (decoding): Cho một chuỗi quan sát O và mô hình HMM λ(A,B,π), tìm ra chuỗi Q tối ưu nhất đã phát sinh ra O. Trong nhận dạng tiếng nói thì đây chính là bài toán nhận dạng, khi quan sát O là tín hiệu tiếng nói thì bài toán là tìm chuỗi âm vị Q tương ứng với tín hiệu này.

Vấn đề 3 (learning): cho một chuỗi quan sát O và tập các trạng thái của HMM, điều chỉnh các tham số λ = {A, B, π} của HMM để P(O| λ) lớn nhất. (Đây chính là bài toán huấn luyện mô hình. Bài toán này đem lại khả năng rất quan trọng của HMM đó là mô hình hóa đối tượng cụ thể trong thực tế với dữ liệu liên tục).

HTK

HTK là một bộ công cụ để xây dựng Mô hình Markov ẩn (HMM). HMM có thể được sử dụng để mô hình hóa bất kỳ chuỗi thời gian nào và cốt lõi của HTK cũng có mục đích chung tương tự. Tuy nhiên, HTK chủ yếu được thiết kế để xây dựng các công cụ xử lý giọng nói dựa trên HMM, cụ thể là các trình biên dịch. Do đó, phần lớn sự hỗ trợ về cơ sở hạ tầng trong HTK được dành riêng cho nhiệm vụ này. Các module của HTK được thể hiện qua Hình 3.13 dưới đây



Hình 3.13: Các giai đoạn xử lý trong HTK *(nguồn [30])*

HTK bao gồm một tập hợp các mô-đun thư viện và các công cụ có sẵn ở dạng nguồn C. Có 4 giai đoạn chính: chuẩn bị, đào tạo, kiểm tra và phân tích dữ liệu.

Các công cụ chuẩn bị dữ liệu

Để tạo một tập hợp các HMM, cần có một tập hợp các tệp dữ liệu giọng nói và các phiên âm liên quan của chúng. Thông thường, dữ liệu lời nói sẽ được lấy từ các kho lưu trữ cơ sở dữ liệu, thường là trên CD-ROM. Trước khi có thể được sử dụng trong đào tạo, nó phải được chuyển đổi thành dạng tham số thích hợp và bất kỳ phiên âm nào liên quan phải được chuyển đổi để có định dạng chính xác và sử dụng nhãn điện thoại hoặc từ bắt buộc. Nếu bài phát biểu cần được ghi lại, thì công cụ HSLab có thể được sử dụng để ghi lại bài phát biểu và chú thích thủ công bài phát biểu đó với bất kỳ phiên âm bắt buộc nào.

Mặc dù tất cả các công cụ HTK đều có thể tham số hóa dạng sóng ngay lập tức, nhưng trong thực tế, tốt hơn hết là chỉ nên tham số hóa dữ liệu một lần. Công cụ HCopy được sử dụng cho việc này. Như tên cho thấy, HCopy được sử dụng để sao chép một hoặc nhiều tệp nguồn sang tệp đầu ra. Thông thường, HCopy sao chép toàn bộ tệp, nhưng nhiều cơ chế được cung cấp để trích xuất các phân đoạn tệp và nối tệp. Bằng cách đặt các biến cấu hình thích hợp, tất cả các tệp đầu vào có thể được chuyển đổi sang dạng tham số khi chúng được đọc trong. Do đó, chỉ cần sao chép từng tệp theo cách này sẽ thực hiện mã hóa cần thiết. Công cụ HList có thể được sử dụng để kiểm tra nội dung của bất kỳ tệp giọng nói nào và vì nó cũng có thể chuyển đổi đầu vào một cách nhanh chóng, nó có thể được sử dụng để kiểm tra kết quả của bất kỳ chuyển đổi nào trước khi xử lý số lượng lớn dữ liệu. Bản ghi âm cũng sẽ cần chuẩn bị. Thông thường, các nhãn được sử dụng trong các bản ghi nguồn gốc sẽ không chính xác như yêu cầu, chẳng hạn như do sự khác biệt về bộ điện thoại được sử dụng. Ngoài ra, đào tạo HMM có thể yêu cầu các nhãn phụ thuộc vào ngữ cảnh. Công cụ HLEd là một trình chỉnh sửa nhãn theo hướng tập lệnh được thiết kế để thực hiện các chuyển đổi cần thiết cho các tệp nhãn. HLEd cũng có thể xuất tệp ra một tệp MLF của nhãn chính duy nhất, điều này thường thuận tiện hơn cho quá trình xử lý tiếp theo. Cuối cùng về chuẩn bị dữ liệu, HLStats có thể thu thập và hiển thị số liệu thống kê trên các

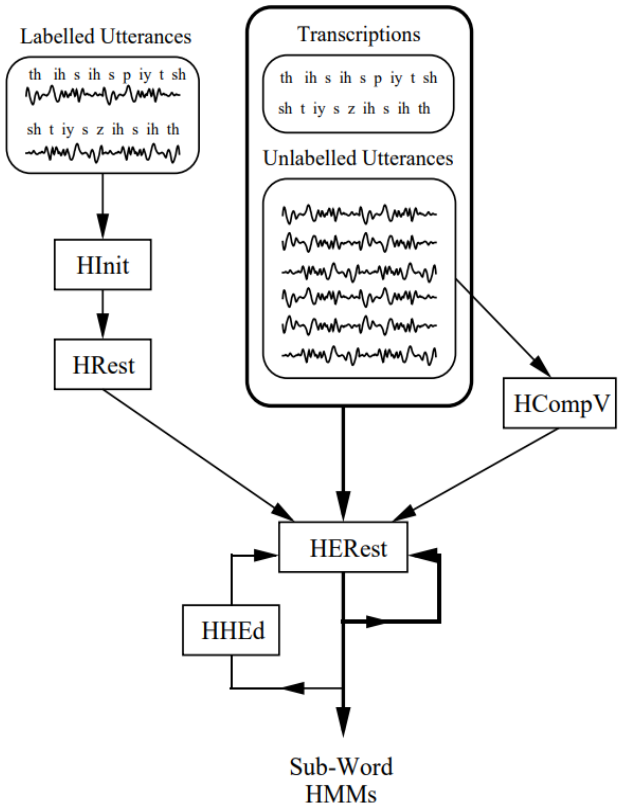
tệp nhãn và nếu cần, HQuant có thể được sử dụng để xây dựng sổ mã VQ nhằm chuẩn bị cho việc xây dựng hệ thống HMM xác suất rời rạc.

Các công cụ huấn luyện

Bước thứ hai của xây dựng hệ thống là xác định cấu trúc liên kết cần thiết cho mỗi HMM bằng cách viết một định nghĩa nguyên mẫu. HTK cho phép các HMM được xây dựng với bất kỳ cấu trúc liên kết mong muốn nào. Các định nghĩa HMM có thể được lưu trữ bên ngoài dưới dạng các tệp văn bản đơn giản và do đó có thể chỉnh sửa chúng bằng bất kỳ trình soạn thảo văn bản thuận tiện nào. Ngoài ra, phân phối HTK tiêu chuẩn bao gồm một số nguyên mẫu HMM mẫu và một tập lệnh để tự động tạo ra các cấu trúc liên kết phổ biến nhất. Ngoại trừ xác suất chuyển đổi, tất cả các tham số HMM được đưa ra trong định nghĩa nguyên mẫu đều bị bỏ qua. Mục đích của định nghĩa nguyên mẫu chỉ là xác định các đặc điểm tổng thể và cấu trúc liên kết của HMM. Các thông số thực tế sẽ được tính toán sau bởi các công cụ đào tạo. Các giá trị hợp lý cho các xác suất chuyển đổi phải được đưa ra nhưng quá trình đào tạo rất thiếu nhạy cảm với các giá trị này. Một chiến lược đơn giản và có thể chấp nhận được để lựa chọn các xác suất này là làm cho tất cả các chuyển đổi ra khỏi bất kỳ trạng thái nào đều có khả năng xảy ra như nhau.

Quá trình đào tạo thực tế diễn ra theo từng giai đoạn và nó được minh họa chi tiết hơn trong Hình 3.14. Đầu tiên, một bộ mô hình ban đầu phải được tạo. Nếu có sẵn một số dữ liệu giọng nói mà vị trí của ranh giới từ phụ (tức là điện thoại) đã được đánh dấu, thì dữ liệu này có thể được sử dụng làm dữ liệu bootstrap. Trong trường hợp này, các công cụ HInit và HRest cung cấp đào tạo kiểu từ riêng biệt bằng cách sử dụng dữ liệu bootstrap được gắn nhãn đầy đủ. Mỗi HMM bắt buộc được tạo riêng lẻ. HInit đọc tất cả dữ liệu đào tạo bootstrap và cắt bỏ tất cả các ví dụ về điện thoại được yêu cầu. Sau đó, nó tính toán lặp đi lặp lại một bộ giá trị tham số ban đầu bằng cách sử dụng thủ tục k-mean phân đoạn. Trong chu kỳ đầu tiên, dữ liệu huấn luyện được phân đoạn đồng nhất, mỗi trạng thái mô hình được đối sánh với các phân đoạn dữ liệu tương ứng và sau đó giá trị và phương sai được ước tính. Nếu các mô hình Gaussian hỗn hợp đang được đào tạo, thì một dạng phân cụm k-mean đã được sửa

đổi sẽ được sử dụng. Vào chu kỳ thứ hai và chu kỳ kế tiếp, sự phân đoạn đồng đều được thay thế bằng sự liên kết Viterbi. Các giá trị tham số ban đầu do HInit tính toán sau đó được HRest ước tính lại. Một lần nữa, dữ liệu bootstrap được gắn nhãn đầy đủ được sử dụng nhưng lần này thủ tục k-mean phân đoạn được thay thế bằng thủ tục ước lượng lại Baum-Welch được mô tả trong chương trước. Khi không có sẵn dữ liệu bootstrap, có thể sử dụng cái gọi là khởi động phẳng. Trong trường hợp này, tất cả các kiểu điện thoại được khởi tạo để giống hệt nhau và có phương tiện trạng thái và phương sai bằng phương sai và phương sai trung bình của giọng nói toàn cầu. Công cụ HCompV có thể được sử dụng cho việc này.



Hình 3.14: Huấn luyện từ phụ trong HMM *(nguồn [30])*

Sau khi tập hợp mô hình ban đầu đã được tạo, công cụ HERest được sử dụng để thực hiện đào tạo nhúng bằng cách sử dụng toàn bộ nhóm đào tạo. HERest thực hiện một ước tính lại Baum-Welch duy nhất của toàn bộ tập hợp các mẫu điện thoại HMM đồng thời. Đối với mỗi câu nói đào tạo, các mô hình điện thoại tương ứng được nối với nhau và sau đó thuật toán chuyển tiếp lùi được sử dụng để tích lũy số liệu

thống kê về nghề nghiệp của trạng thái, phương tiện, phương sai, v.v…, cho mỗi HMM trong trình tự. Khi tất cả dữ liệu đào tạo đã được xử lý, thống kê tích lũy được sử dụng để tính toán ước tính lại các tham số HMM. HERest là công cụ đào tạo HTK cốt lõi. Nó được thiết kế để xử lý cơ sở dữ liệu lớn, nó có các phương tiện để cắt tỉa nhằm giảm bớt tính toán và nó có thể chạy song song trên một mạng máy móc. Triết lý xây dựng hệ thống trong HTK là các HMM nên được tinh chỉnh từng bước. Do đó, một tiến trình điển hình là bắt đầu với một tập hợp đơn giản các mô hình điện thoại độc lập ngữ cảnh Gaussian và sau đó tinh chỉnh lặp đi lặp lại bằng cách mở rộng chúng để bao gồm phụ thuộc ngữ cảnh và sử dụng nhiều phân phối Gaussian thành phần hỗn hợp. Công cụ HHEd là một trình biên tập định nghĩa HMM sẽ sao chép các mô hình thành các bộ phụ thuộc vào ngữ cảnh, áp dụng nhiều loại tham số và tăng số lượng các thành phần hỗn hợp trong các bản phân phối được chỉ định. Quy trình thông thường là sửa đổi một tập hợp các HMM trong các giai đoạn bằng cách sử dụng HHEd và sau đó ước tính lại các tham số của tập đã sửa đổi bằng HERest sau mỗi giai đoạn. Để cải thiện hiệu suất cho các diễn giả cụ thể, các công cụ HEAdapt và HVite có thể được sử dụng để điều chỉnh HMM nhằm mô hình hóa tốt hơn các đặc điểm của các diễn giả cụ thể bằng cách sử dụng một lượng nhỏ dữ liệu đào tạo hoặc điều chỉnh. Kết quả cuối cùng của nó là một hệ thống điều chỉnh người nói.

Vấn đề lớn nhất duy nhất trong việc xây dựng hệ thống HMM phụ thuộc vào ngữ cảnh luôn là thiếu dữ liệu. Bộ mô hình càng phức tạp thì càng cần nhiều dữ liệu để đưa ra các ước tính mạnh mẽ về các tham số của nó và vì dữ liệu thường bị giới hạn, nên phải cân bằng giữa độ phức tạp và dữ liệu có sẵn. Đối với hệ thống mật độ liên tục, sự cân bằng này đạt được bằng cách gắn các thông số lại với nhau như đã đề cập ở trên. Việc ràng buộc tham số cho phép dữ liệu được gộp chung để các tham số được chia sẻ có thể được ước tính một cách chắc chắn. Ngoài các hệ thống mật độ liên tục, HTK cũng hỗ trợ các hệ thống hỗn hợp được ràng buộc hoàn toàn và các hệ thống xác suất rời rạc. Trong những trường hợp này, vấn đề thiếu dữ liệu thường được giải quyết bằng cách làm mịn các bản phân phối và công cụ HSmooth được sử dụng cho việc này.

Các công cụ nhận dạng

HTK cung cấp một công cụ nhận dạng duy nhất được gọi là HVite sử dụng thuật toán chuyển mã thông báo được mô tả trong chương trước để thực hiện nhận dạng giọng nói dựa trên Viterbi. HVite lấy đầu vào là một mạng lưới mô tả các chuỗi từ được phép, một từ điển xác định cách mỗi từ được tạo danh từ chuyên nghiệp và một tập hợp các HMM. Nó hoạt động bằng cách chuyển đổi mạng từ thành mạng điện thoại và sau đó gắn định nghĩa HMM thích hợp vào từng phiên bản điện thoại. Sau đó, nhận dạng có thể được thực hiện trên danh sách các tệp giọng nói được lưu trữ hoặc trên đầu vào âm thanh trực tiếp. Như đã lưu ý ở cuối chương trước, HVite có thể hỗ trợ các bộ ba chữ chéo và nó có thể chạy với nhiều mã thông báo để tạo ra các mạng chứa nhiều giả thuyết. Nó cũng có thể được cấu hình để phân chia lại các mạng và thực hiện căn chỉnh bắt buộc. Các mạng từ cần thiết để thúc đẩy HVite thường là các vòng lặp từ đơn giản trong đó bất kỳ từ nào có thể theo sau bất kỳ từ nào khác hoặc chúng là các biểu đồ có hướng biểu thị ngữ pháp nhiệm vụ trạng thái hữu hạn. Trong trường hợp trước đây, xác suất bigram thường được gắn với các chuyển đổi từ.

Mạng từ ngữ được lưu trữ bằng định dạng mạng tiêu chuẩn HTK. Đây là một định dạng dựa trên văn bản và do đó, các mạng từ có thể được tạo trực tiếp bằng trình soạn thảo văn bản. Tuy nhiên, điều này khá tẻ nhạt và do đó HTK cung cấp hai công cụ để hỗ trợ tạo mạng từ. Thứ nhất, HBuild cho phép các mạng con được tạo và sử dụng trong các mạng cấp cao hơn. Do đó, mặc dù cùng một ký hiệu cấp thấp được sử dụng, nhưng sẽ tránh được nhiều sự trùng lặp. Ngoài ra, HBuild có thể được sử dụng để tạo các vòng lặp từ và nó cũng có thể đọc trong mô hình ngôn ngữ bigram đã được hỗ trợ và sửa đổi các chuyển đổi vòng lặp từ để kết hợp các xác suất bigram. Lưu ý rằng công cụ thống kê nhãn mà HLStats đã đề cập trước đó có thể được sử dụng để tạo mô hình ngôn ngữ bigram dự phòng. Để thay thế cho việc chỉ định trực tiếp một mạng từ, một ký hiệu ngữ pháp cấp cao hơn có thể được sử dụng. Ký hiệu này dựa trên Extended Backus Naur Form (EBNF) được sử dụng trong đặc tả của trình biên dịch và nó tương thích với ngôn ngữ đặc tả ngữ pháp được sử dụng trong các phiên bản trước đó của HTK. Công cụ HParse được cung cấp để chuyển đổi ký hiệu này

thành mạng từ tương đương. Cho dù phương pháp nào được chọn để tạo mạng từ, sẽ rất hữu ích khi có thể xem các ví dụ về ngôn ngữ mà nó định nghĩa. Công cụ HSGen được cung cấp để thực hiện việc này. Nó lấy đầu vào là một mạng và sau đó duyệt ngẫu nhiên các chuỗi từ xuất ra của mạng. Sau đó, các chuỗi này có thể được kiểm tra để đảm bảo rằng chúng tương ứng với những gì được yêu cầu. HSGen cũng có thể tính toán mức độ phức tạp theo kinh nghiệm của nhiệm vụ. Cuối cùng, việc xây dựng các từ điển lớn có thể liên quan đến việc hợp nhất một số nguồn và thực hiện nhiều phép biến đổi trên mỗi nguồn. Công cụ quản lý từ điển HDMan được cung cấp để hỗ trợ quá trình này.

Các công cụ thống kê

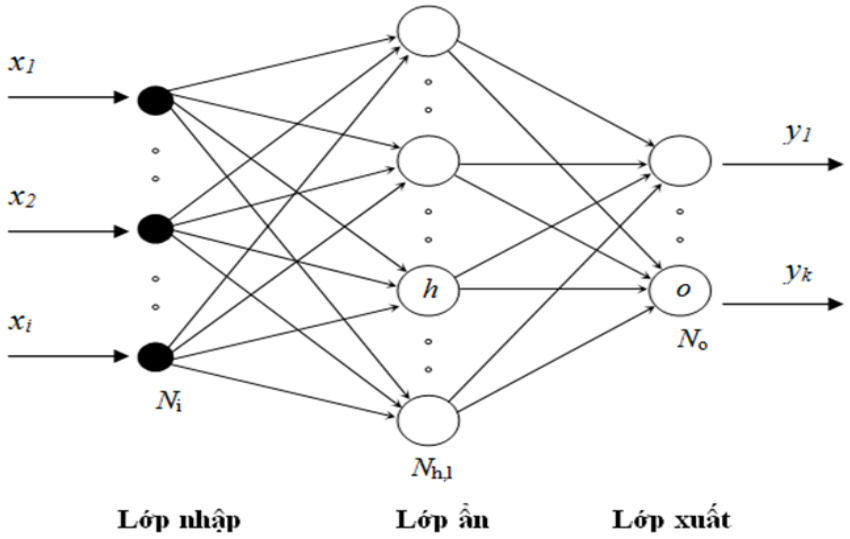
Sau khi trình hoàn thiện dựa trên HMM đã được xây dựng, cần phải đánh giá hiệu suất của nó. Điều này thường được thực hiện bằng cách sử dụng nó để phiên âm một số câu kiểm tra được ghi trước và khớp đầu ra của trình biên dịch với các phiên âm tham chiếu chính xác. Việc so sánh này được thực hiện bởi một công cụ có tên là HResults sử dụng lập trình động để căn chỉnh hai phiên âm và sau đó đếm các lỗi phụ, xóa và chèn. Các tùy chọn được cung cấp để đảm bảo rằng các thuật toán và định dạng đầu ra được HResults sử dụng tương thích với các định dạng được sử dụng bởi Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (NIST). Cũng như các thước đo hiệu suất toàn cầu, HResults cũng có thể cung cấp bảng phân tích từng người nói, ma trận nhầm lẫn và bản chép lời phù hợp với thời gian. Đối với các ứng dụng dò chữ, nó cũng có thể tính toán điểm số Figure of Merit (FOM) và Receiver Operating Curve (ROC).

3.3.2. Nhận dạng người nói với Feedforward-DNN

Mạng nơ-ron với một số mức độ phức tạp, thường có ít nhất hai lớp, đủ điều kiện là mạng nơ-ron sâu (DNN), hay gọi tắt là deep net. DNN xử lý dữ liệu theo những cách phức tạp bằng cách sử dụng mô hình toán học phức tạp. Có nhiều loại mô hình DNN nhưng ở phạm vi luận văn này sẽ áp dụng với feedforward-DNN.

Đầu tiên ta sẽ tìm hiểu về mạng feedforward truyền thống. Mạng truyền thẳng truyền thống thường có ba lớp: lớp nhập, một lớp ẩn, và lớp xuất . Mỗi lớp gồm một

hay nhiều neural. Số neural trong lớp nhập và số neural trong lớp xuất được xác định tương ứng với số biến đầu vào và số biến đầu ra của bài toán. Số neural của lớp ẩn do người thiết kế mạng quyết định dựa vào độ phức tạp của hàm ánh xạ do mạng thực hiện. Mỗi neural của lớp thứ i (0<i<n) liên kết với mọi neural của lớp thứ i+1, và các neural trong cùng lớp không liên kết với nhau. Kiểu kết nối giữa các lớp như thế này gọi là kết nối đầy đủ. Một mạng truyền thẳng là một mạng kết nối đầy đủ. Mỗi kết nối trong mạng được gắn một trọng số w ∈ R. Trọng số thể hiện mức độ quan trọng (hay độ mạnh) của dữ liệu đầu vào đối với quá trình xử lý thông tin tại mỗi neural.



Hình 3.15: Mạng truyền thẳng một lớp ẩn

Trong hình 2.4, mô tả một mạng truyền thẳng truyền thống. Giá trị của các biến đầu vào được chuyển cho các neural nhập của mạng. Các neural nhập không xử lý gì cả. Mỗi neural trong lớp nhập chuyển giá trị đầu vào cho tất cả các neural ẩn. Tại mỗi neural ẩn, quá trình tính giá trị kết xuất của nó được thực hiện bằng áp dụng một hàm kích hoạt trên giá trị tích hợp tất cả các giá trị đầu vào chuyển đến cho neural. Giá trị tích hợp tất cả các giá trị đầu vào chuyển đến cho neural gọi là tổng trọng, được tính như sau:

$$u_j = a_{oj} + \sum_{i=1}^{n} w_{ij} x_i \qquad (2.5)$$

Với n là số nút nhập, J là số nút ẩn, wij là trọng số tương ứng của nút nhập i với nút ẩn j, a_{0j} là một ngưỡng hay bias của nút ẩn j.

Một hàm kích hoạt được áp dụng trên tổng trọng này để cho ra giá trị thực của nút ẩn: $y_j = g(u_j), j = 1,\dots J.$ (2.6)

Hàm kích hoạt được dùng để giới hạn phạm vi đầu ra của mỗi neural. Thông thường, đầu ra của mỗi neural được giới hạn trong đoạn [0, 1] hoặc [-1, 1]. Các hàm kích hoạt rất đa dạng, có thể là các hàm tuyến tính hoặc phi tuyến.

Sau khi xác định được giá trị của các nút ẩn, những giá trị này trở thành giá trị đầu vào của các nút xuất. Tại mỗi nút xuất, quá trình tính giá trị của nút xuất thực hiện tương tự như tại nút ẩn. Giá trị của mỗi nút xuất là một đầu ra của mạng. Kết xuất của nút xuất thứ k là:

$$z_k = g(v_k), k = 1,\dots, K \quad (2.7)$$

Trong đó g(vk) là hàm kích hoạt với tham số là tổng trọng vk

$$v_k = b_{ok} + \sum_{j=1}^{J} b_{jk} y_j \quad (2.8)$$

Với yj là kết xuất của J nút ẩn, b_{jk} là các trọng số tương ứng của nút ẩn j với nút xuất k, b0k là một ngưỡng hay bias của nút xuất thứ k.

Quá trình tính các kết xuất của mạng như trên gọi là quá trình ánh xạ hay lan truyền tiến. Dòng dữ liệu di chuyển theo một chiều bắt đầu từ lớp nhập, đến lớp ẩn, và sau cùng đến lớp xuất. Mạng không có các kết nối feedback tham chiếu ngược trở lại kết xuất đầu ra trong những lần ánh xạ tiếp theo.

Các kết xuất của mạng thu được sau quá trình ánh xạ thường không chính xác so với kết xuất mong muốn, tạo ra các lỗi sai lệch do tập trọng số và bias xác định ban đầu là khởi tạo ngẫu nhiên và thường không đúng. Mạng sẽ trải qua một quá trình gọi là huấn luyện để điều chỉnh tập trọng số và bias sao cho mạng xấp xỉ tốt nhất với hàm cần tìm. Một tập dữ liệu X được dùng làm đầu vào cho huấn luyện mạng được gọi là tập huấn luyện (training set). Các phần tử $x \in X$ được gọi là các mẫu huấn luyện (training examples). Qua quá trình huấn luyện trên tập X, tập trọng số và bias của mạng sẽ hội tụ dần tới các giá trị sao cho với mỗi vector đầu vào $x \in X$, mạng sẽ cho ra vector đầu ra y như mong muốn.

Một giá trị lỗi E được sử dụng để đo độ sai lệch giữa ánh xạ cần xây dựng và hàm đích cho trước qua tập mẫu X. Trong các mạng truyền thẳng truyền thống, giá trị lỗi E được tính theo phương pháp trung bình bình phương lỗi (mean squared error):

$$E = \frac{\frac{1}{2}\sum_1^K (y_k - y_k^*)^2}{K} \qquad (2.9)$$

Với K mẫu huấn luyện, yk là kết xuất mong muốn của mẫu thứ k, y*k là kết xuất của mạng ứng với mẫu nhập k.

Ứng với tập huấn luyện X, ta có thể xem E là một hàm lỗi được xác định bởi tập trọng số wi và bias bi của mạng. Ta cần tìm mô hình mạng với tập wi và bi sao cho kết xuất của mạng xấp xỉ hàm đích hay tối thiểu hàm lỗi E(w,b). Từ đây, ký hiệu wi, bi đại diện cho trọng số và bias của các neural, không phân biệt neural ẩn hay neural xuất.

Trong các mạng truyền thẳng truyền thống, phương pháp xác định tập trọng số và bias để tối thiểu hàm lỗi E là phương pháp giảm gradient. Giảm gradient là phương pháp tối ưu hóa xuống đồi theo hướng vecto đạo hàm. Theo đó trọng số và bias của mạng được điều chỉnh theo quy tắc:

$$W_k = W_{k-1} - \eta \sum_{n=1}^{N} \left(\frac{\partial E}{\partial w}\right)_n , \qquad (2.10a)$$

Với η là hệ số học, N là số mẫu huấn luyện, Wk là vector trọng số tại bước huấn luyện thứ k, $\frac{\partial E}{\partial w}$ là đạo hàm của hàm lỗi E theo từng trọng số w.

Ta có công thức điều chỉnh tương tự với bias bk:

$$b_k = b_{k-1} - \eta \sum_{n=1}^{N} \left(\frac{\partial E}{\partial b}\right)_n \qquad (2.10b)$$

Trong hai công thức trên, dấu trừ diễn tả hướng ngược lại với vecto tổng. Một hằng số dương, có giá trị nhỏ và cố định gọi là hệ số học η. Chọn hệ số học có ảnh hưởng tới hiệu quả của thuật toán huấn luyện. Nếu chọn η quá nhỏ dẫn tới thay đổi của trọng số quá ít và làm cho lỗi E giảm chậm và kết quả thuật toán học rất chậm. Nếu chọn η khá lớn, biến thiên của trọng số lớn có thể dẫn tới lỗi E lúc nào đó sẽ tăng chứ không giảm.

Thuật toán thực hiện huấn luyện mạng truyền thẳng truyền thống là thuật toán lan truyền ngược (BackPropagation). Trong thuật toán này, trọng số của mạng được cập nhật theo chiều từ lớp xuất đến lớp nhập dựa trên giảm gradient. Thuật toán mô tả như sau:

Cho một tập n mẫu huấn luyện (x_j, t_j) j:1,..,n

Bước 1: Khởi tạo là giá trị ngẫu nhiên w_{ij}, a_{oj} , b_{ok} , b_{jk}

Bước 2: Tính giá trị đầu ra cho các neural của lớp ẩn theo công thức (2.6)

Bước 3: Tính giá trị đầu ra cho các neural của lớp xuất theo công thức (2.7)

Bước 4: Tính gradient và điều chỉnh các trọng số và bias theo công thức (2.10a), (2.10b)

Bước 5: Lặp lại từ bước 2 cho đến khi đạt ngưỡng của E hoặc số lần lặp

Thuật toán huấn luyện lan truyền ngược dựa trên giảm gradient có tốc độ hội tụ chậm. Vì vậy, một số thuật toán đưa ra cải thiện về tốc độ hội tụ hay tốc độ học của mạng như thuật toán Levenberg-Marquardt, thêm momentum, khởi động trọng số (Widrow - Nguyen), …

- Mặc dù có nhiều cải tiến, nhưng hầu hết các thuật toán lan truyền ngược dựa trên giảm gradient có các nhược điểm sau:
- Chọn hệ số học, khi hệ số học η quá nhỏ thì thuật toán hội tụ rất chậm. Tuy nhiên η quá lớn thì giá trị cực trị có thể bỏ qua.
- Một vấn đề khác là thuật toán này có thể bị tối ưu cục bộ thay vì tối ưu toàn cục.
- Vấn đề quá khớp xảy ra khi mạng được luyện quá sát với dữ liệu huấn luyện (kể cả nhiễu), nên nó sẽ trả lời chính xác những gì đã được học, còn những gì không được học thì nó không quan tâm. Như vậy mạng sẽ không có được khả năng tổng quát hóa.
- Phương pháp học dựa trên giảm gradient là mất nhiều thời gian trong hầu hết các ứng dụng.

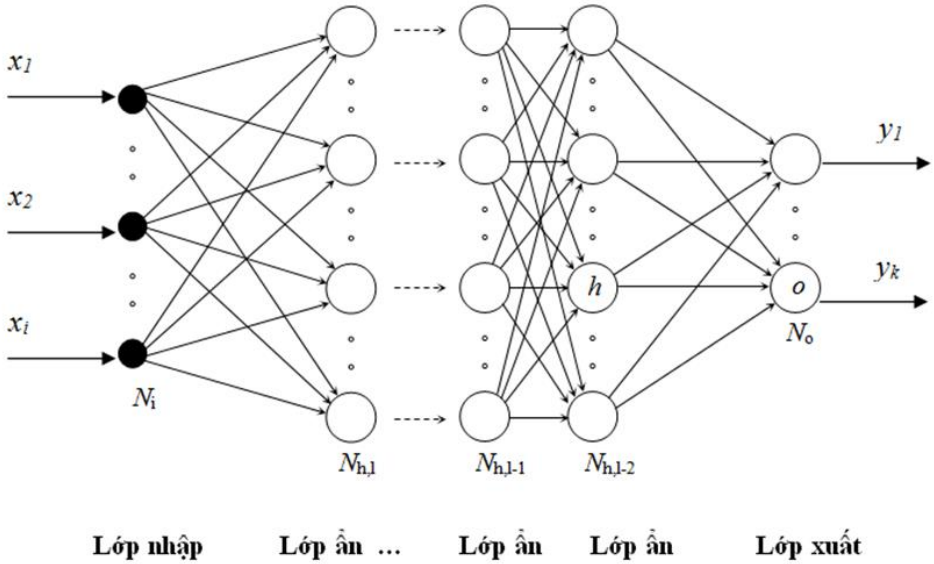
Mạng neural truyền thống hay mạng feedforward với nhiều lớp ẩn cho phép ta hiện thực “deep learning”. Mạng feedforward nhiều lớp là một hàm toán học, ánh xạ

một tập đầu vào thành các giá trị đầu ra. Hàm được hợp thành từ nhiều hàm đơn giản. Ví dụ, ta có ba hàm f(1), f(2), f(3). Chúng được kết nối thành một chuỗi, có dạng f(x) = f(3)(f(2)(f(1)(x)))). Trong trường hợp này, f(1) chính là lớp thứ nhất, f(2) là lớp thứ hai, và f(3) là lớp thứ ba của mạng. Toàn bộ chiều dài của chuỗi tạo nên độ sâu của mạng. Dữ liệu đầu vào, được biến đổi khi đi qua lần lượt các lớp. Mức độ trừu tượng của dữ liệu tăng dần khi qua các lớp của mạng.

Mạng feedforward sử dụng thuật toán lan truyền ngược để huấn luyện mạng. Thuật toán lan truyền ngược được phát triển vào những năm 1960s – 1970s và được áp dụng vào mạng neural vào năm 1981. Vào cuối 1980s, phần lớn các ứng dụng chỉ thành công khi dùng feedforward với một lớp ẩn. Các lớp ẩn bổ sung thêm thường không mang lại lợi ích. Điều này được khẳng định bởi định lý xấp xỉ tổng quát (the universal approximation theorem) từ Hornik et al., 1989; Cybenko, 1989 cho rằng : "một mạng feedforward với một lớp xuất tuyến tính, và ít nhất một lớp ẩn dùng một hàm kích hoạt nào đó (như logistic sigmoid) có thể xấp xỉ bất kỳ một hàm liên tục nào trên một tập con đóng và bị chặn Rn từ một không gian hữu hạn chiều tới không gian khác với số lỗi mong muốn, với điều kiện là mạng có đủ số nút ẩn". Tuy nhiên định lý này không nói về vấn đề thuật toán huấn luyện có khả năng học một hàm như vậy hay không. Và như vậy, một mạng feedforward có khả năng biểu diễn hàm, thì việc học vẫn có thể không thành công vì hai lý do:

- Thuật toán tối ưu được dùng để huấn luyện có thể không có khả năng tìm giá trị của các tham số mà đúng với hàm mong muốn.
- Thuật toán huấn luyện có thể chọn sai hàm bởi vì overfitting.

Định lý xấp xỉ tổng quát cũng không chỉ ra mạng cần bao nhiêu nút ẩn là là đủ, nên nếu số nút ẩn không đủ lớn, mạng có thể không học tốt và không đạt độ chính xác ta muốn. Bởi vậy, có thể dùng mô hình nhiều lớp ẩn để thay thế.



Hình 3.16: Cấu trúc mạng feedforward-DNN

Gần đây, khi khái niệm "deep learning" ra đời, mạng feedforward có kiến trúc lớn hơn nhiều so với các mạng truyền thống. Chúng có nhiều lớp ẩn hơn, mỗi lớp ẩn có nhiều nút ẩn hơn (Hình 3.16). Thiết kế này bám sát theo tư duy mô hình hóa thông tin trừu tượng. Cứ qua mỗi lớp ẩn, đặc trưng hay thông tin có ở lớp trước lại được biến đổi sang tầng biểu diễn cao hơn. Càng về sau, mức độ trừu tượng của thông tin càng cao. Khi đó một câu hỏi được đặt ra là bao nhiêu lớp ẩn và bao nhiêu nút ẩn cho mỗi lớp thì đủ. Câu trả lời vẫn luôn phụ thuộc vào qui mô, tính chất và kích thước của dữ liệu học. Người ta chọn các thông số này qua thực nghiệm.

Có 2 vấn đề phát sinh khi đưa từ mô hình đơn lớp sang đa lớp ẩn:

- Một là tính khả thi của mô hình. Mạng có kích thước càng lớn, chi phí tính toán càng cao. Với cấu trúc nhiều lớp ẩn, chi phí tính toán có thể bùng nổ rất lớn, vượt quá khả năng cho phép của phần cứng hiện tại. Do đó, cần có các hiệu chỉnh về thuật toán cho thích nghi với vấn đề này.
- Hai là khả năng biểu diễn thông tin. Mục tiêu chính của DNN là chuyển hóa được thông tin trừu tượng, trong khi ANN quan tâm đến xấp xỉ hàm phi tuyến cho phân lớp. Chính vì vậy, cần có thay đổi hợp lý trong thuật toán cho DNN.

CHƯƠNG 4: THỰC NGHIỆM

4.1. Dữ liệu thực nghiệm

Bộ dữ liệu thực nghiệm của bài toán được tác giả thu thập tại nơi sinh sống thuộc địa bàn tỉnh Tây Ninh. Bộ dữ liệu được thu thập từ 42 người giới tính nam nữ với độ tuổi từ 12-52, chủ yếu đến từ miền Nam của Việt Nam. Nội dung của bộ dữ liệu là các file ghi âm giọng nói của các đối tượng tham gia khảo sát.

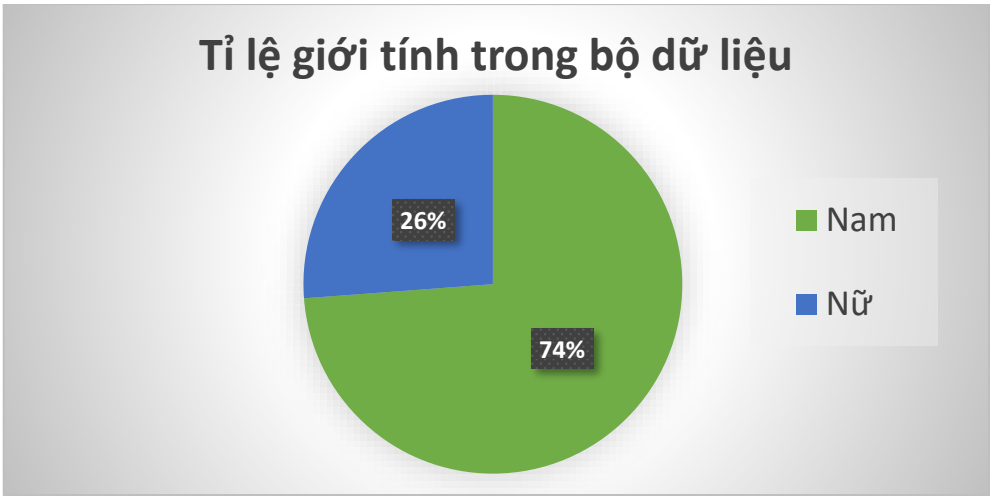
Môi trường thu âm yên tĩnh, micro gần, với mỗi người tham gia ghi âm, tác giả đã chuẩn bị một nội dung văn bản tự do thu thập từ nguồn báo online để thỏa ngữ cảnh "độc lập văn bản" với thời lượng thu âm mỗi người trên dưới 10 phút và yêu cầu họ đọc đoạn nội dung một cách tự nhiên nhất. Kết quả thu được bộ dữ liệu thô bao gồm đoạn ghi âm và thông tin nhận dạng của người tham dự với các tiêu chí bao gồm: họ tên, giới tính, độ tuổi, vùng miền được liệt kê trong bảng 4.1.

Bảng 4.1: Thông tin người tham gia ghi âm

BẢN THU	Tên	Tuổi	Giới tính	Vùng miền
N2	Mỹ Dung	36	Nữ	Nam
N4	Chon	35	Nam	Nam
N5	Bé Ba	30	Nữ	Nam
N6	Huy	18	Nam	Bắc
N7	Hoàng	14	Nam	Bắc
N8	Nhàn	42	Nữ	Bắc
N9	Khang	16	Nam	Nam
N10	An	42	Nữ	Nam
N11	Ngân	30	Nữ	Bắc
N12	Bo	23	Nam	Nam
N13	Quế	52	Nữ	Nam
N14	Thư	27	Nữ	Nam
N15	Duyên	42	Nữ	Bắc
N16	Anh Ngọc	47	Nam	Bắc
N17	Khánh	12	Nữ	Bắc
N18	Gia Huy	15	Nam	Nam
N19	Minh Hoàng	27	Nam	Nam
N20	Quý	48	Nam	Nam
N21	Luyến	32	Nam	Nam
N22	Trí	45	Nam	Nam

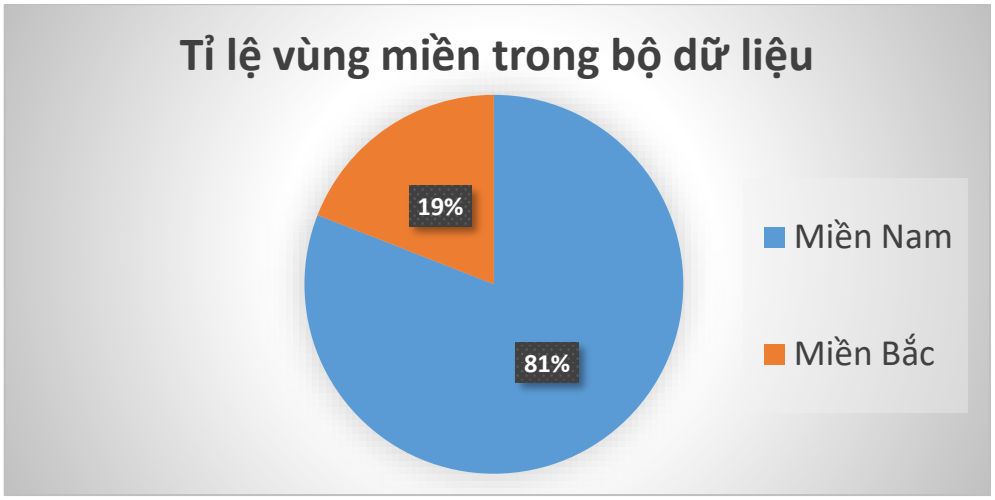
N23	Duyên	48	Nam	Nam
N24	Trực	45	Nam	Nam
N25	Nhựt	48	Nam	Nam
N26	Cường	43	Nam	Nam
N27	Vũ	45	Nam	Nam
N28	Thái	45	Nam	Nam
N29	Vinh	45	Nam	Nam
N30	Lâm Hải	40	Nam	Nam
N31	Anh Thành	49	Nam	Nam
N32	Trọng Hải	38	Nam	Nam
N33	Phương	36	Nam	Nam
N34	Phát	34	Nam	Nam
N35	Nguyên	24	Nam	Nam
N36	Đạt	45	Nam	Nam
N37	Hoàng Minh	45	Nam	Nam
N38	Lộc	45	Nam	Nam
N39	Tuấn	34	Nam	Nam
N40	Hưởng	47	Nam	Bắc
N41	Thùy Duyên	39	Nữ	Nam
N42	Châu	45	Nam	Nam
N43	Hùng Phương	26	Nam	Nam
N44	Quân	50	Nam	Nam

Chi tiết cụ thể thống kê bộ dữ liệu được thể hiện trong các sơ đồ dưới đây (hình 4.1):



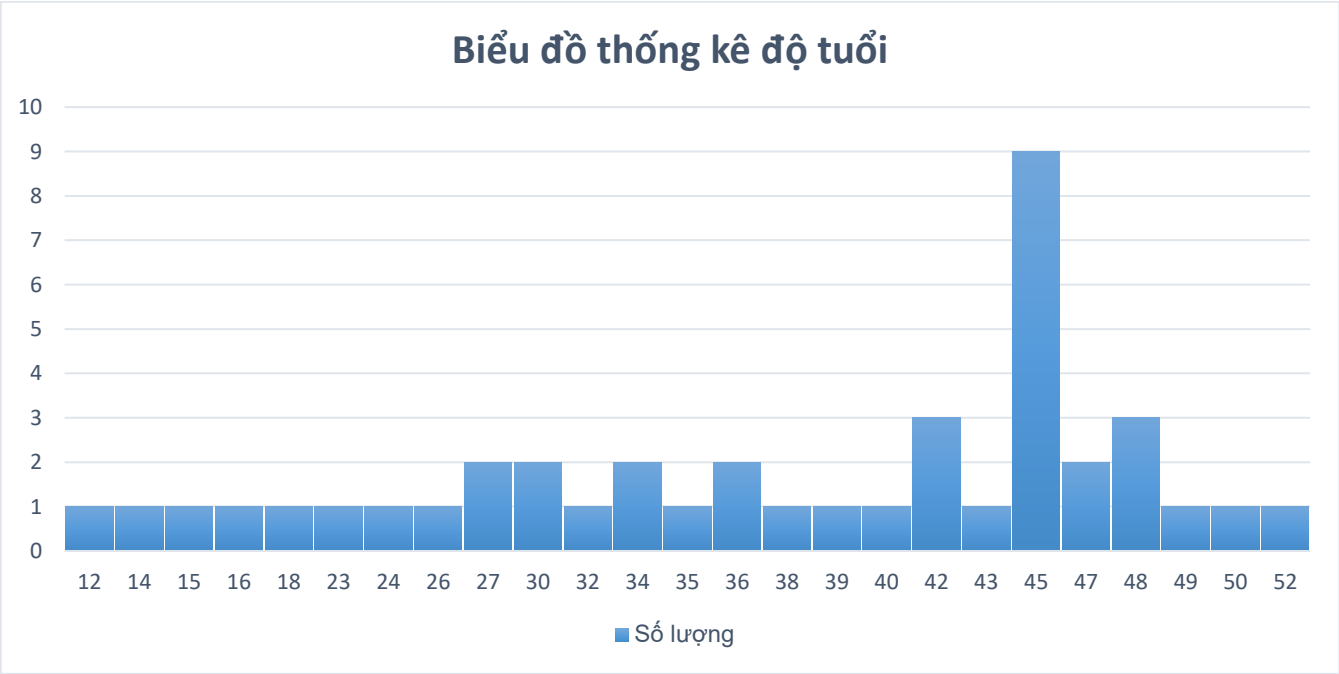
Hình 4.1: Biểu đồ hiển thị tỉ lệ giới tính trong bộ dữ liệu

Bộ dữ liệu có tổng cộng 31 người thuộc giới tính nam chiếm 74% và 11 người giới tính nữ chiếm 26%. Nhận thấy có sự chênh lệnh lớn về yếu tố giới tính trong bộ dữ liệu này.



Hình 4.2: Biểu đồ hiển thị tỉ lệ vùng miền trong bộ dữ liệu

Xét về yếu tố vùng miền (hình 4.2), đa số người đến từ đến từ miền Nam chiếm tỉ lệ 81% với tổng cộng là 34 người và 8 người đến từ miền Bắc chiếm chỉ 19%. Mặc dù Việt Nam là một quốc gia đa dân tộc, tuy nhiên, ta thấy bộ dữ liệu chưa có sự cân bằng và đa dạng về giọng nói ở nhiều địa phương, vùng miền khác nhau.



Hình 4.3: Biểu đồ thống kê độ tuổi của bộ dữ liệu

Từ biểu đồ ở Hình 4.3 ta thấy, với số lượng mẫu là 42 thì độ tuổi dao động trong khoảng từ 12-52 tuổi, trong đó số mẫu mỗi độ tuổi là khá nhỏ. Bên cạnh đó Hình 4.3 cũng cho ta thấy tỉ lệ phân bố về độ tuổi của người tham dự tập trung nhiều ở độ tuổi từ 42-48.

Nhận xét thấy bộ dữ liệu khá ít và có sự phân bố chưa đồng đều về các yếu tố giới tính, vùng miền, độ tuổi,… Sự phong phú về bộ dữ liệu là hạn chế, vì thế điều này có thể ảnh hưởng ít nhiều đến kết quả thực nghiệm của mô hình.

Về chất lượng các bản ghi âm, các bản thu được thực hiện trong môi trường khá tương đồng, giọng nói của các cá thể tham gia được ghi lại rõ ràng, rành mạch. Đây có thể là một yếu tố giúp mô hình có thể học tập dễ dàng hơn vì dữ liệu "sạch". Bảng 4.2 dưới đây mô tả thông tin chi tiết của một bản ghi âm điển hình.

Bảng 4.2: Thông tin chi tiết của một bản ghi âm

Format	Bit rate	Channel(s)	Sampling rate	Bit depth
Wave	768 kbps	1 channel	48 KHz	16 bits

4.2. Kịch bản thực nghiệm

Để kiểm chứng bộ dữ liệu với những góc nhìn khác nhau, luận văn đặt kịch bản thực nghiệm 2 mô hình đã đề cập trong tình huống như sau. Xây dựng mô hình HMM bằng cách sử dụng bộ công cụ HTK và Feedforward-DNN bằng Tensorflow, sử dụng đặc trưng MFCC được trích xuất dựa trên bộ dữ liệu. So sánh, đánh giá độ chính xác, phù hợp của mô hình trên bộ dữ liệu tự xây dựng.

4.2.1. Chuẩn bị môi trường

Mã nguồn của luận văn với mô hình Feedforward-DNN được viết bằng ngôn ngữ Python kết hợp với framework Tensorflow. Bên cạnh đó mô hình HMM được xây dựng bởi bộ công cụ của HTK. HTK được cung cấp công khai và có thể được tải về để chỉnh sửa và sử dụng tùy ý. Cấu hình máy:

- CPU: Core i5, UBUNTU 20
- SSD: 80GB
- RAM: 8GB

- Software: Tensorflow (Training), HTK (GET MFCC)
- Language code: C#, Python, Perl

4.2.2. Chuẩn bị dữ liệu

Tiến hành xây dựng cấu trúc các thư mục, tệp tin như hướng dẫn của HTK, tiếp theo bắt đầu quá trình tiền xử lý dữ liệu đầu vào

1. **Tạo file listwavmfc từ folder wav**

```
perl pl/listwavmfc.pl wav txt/listwavmfc
```

2. **Lấy MFCC với module HCopy**

Rút trích đặc trưng sẽ chuyển đổi âm thanh ở dạng sóng sang định dạng MFCC được thực hiện bởi HCopy, nhằm chuyển đổi các tệp âm thành các tệp có phần mở rộng là .mfc.

```
bin/HCopy -C cfg/HCopy.cfg -S txt/listwavmfc
```

3. **Tạo file train.scp**

File train này sẽ bao gồm tất cả dữ liệu âm thanh của bài toán. Từ file gốc này tách ra thành 2 phần với tỉ lệ 7:3 để phục vụ cho việc huấn luyện và kiểm thử.

```
perl pl/mkTrainFile.pl wav txt/train.scp
```

4. **Xây dựng file gram.txt**

Để sử dụng các mô hình mà HTK cung cấp, ta phải định nghĩa nên kiến trúc cơ bản của trình nhận dạng (hay còn gọi là task grammar). Trong HTK, task grammar được viết trong tệp văn bản (thường đặt tên là gram.txt). File gram này sẽ chứa tên để định danh của người nói. Bấy nhiêu người nói là bấy nhiêu định danh. Sau đó thực hiện lệnh HParse để biên dịch task grammar (được mô tả trong gram.txt) thành task network lưu vào file wdnet.txt.

```
bin/HParse txt/gram.txt txt/wdnet.txt
```

5. **Xây dựng file promts.txt**

Hệ thống cần phải biết rằng HMM nào tương ứng với từng biến của grammar, vì thế ở đây chúng ta cần xây dựng thêm task dictionary

```
perl pl/mkPromt.pl wav txt/prompts.txt
```

6. Tạo label cho mô hình

```
perl pl/prompts2mlf.pl mlf/phones0.mlf txt/prompts.txt
perl pl/prompts2wlist.pl txt/prompts.txt txt/wlist.txt
bin/HDMan -m -w txt/wlist.txt -n ph/monophones0 txt/dict txt/dict.dct
```

4.2.3. Xây dựng mô hình và huấn luyện

Bước đầu tiên trong đào tạo **HMM** là xác định một mô hình nguyên mẫu. Các tham số của mô hình này không quan trọng, mục đích của nó là xác định cấu trúc liên kết của mô hình. Một trong các cấu trúc liên kết tốt để sử dụng là 3-state left-right, ta tiến hành xây dựng lên nó trong file proto như sau:

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"
<BeginHMM>
 <NumStates> 5
 <State> 2 <NumMixes> 8
      <Mixture> 1 0.1
            <Mean> 39
              0.0 0.0 0.0 …
            <Variance> 39
              1.0 1.0 1.0 …
      <State> 4 <NumMixes> 8
      <Mixture> 1 0.1
            <Mean> 39
              0.0 0.0 0.0 …
            <Variance> 39
              1.0 1.0 1.0 …
      …
      <Mixture> 8 0.2
            <Mean> 39
              0.0 0.0 0.0 …
            <Variance> 39
              1.0 1.0 1.0 …
 <TransP> 5
  0.0 1.0 0.0 0.0 0.0
  0.0 0.6 0.4 0.0 0.0
  0.0 0.0 0.6 0.4 0.0
  0.0 0.0 0.0 0.7 0.3
  0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Trong đó mỗi vec-tơ có độ dài là 39. Con số này được tính bởi độ dài của vectơ tĩnh được tham số hóa (MFCC_0 = 13) cộng với hệ số delta (+13) cộng với hệ số gia

tốc (+13). Tiếp theo, sử dụng HCompV để quét qua tập dữ liệu, tính toán giá trị trung bình và phương sai toàn cục và đặt tất cả Gaussian trong một HMM nhất định để có cùng phương sai và giá trị trung bình. Với danh sách tất cả các tệp đào tạo được lưu trữ trong file train_70.scp, thực thi câu lệnh sau:

```
bin/HCompV  -C cfg/HCompV.cfg -f 0.01 -m -S txt/train_70.scp -M hmm0
proto
```

Model được lưu trữ trong thư mục hmm0 được ước tính lại bằng cách HERest

```
bin/HERest -C cfg/HERest.cfg -I mlf/phones0.mlf -t 250.0 150.0 1000.0
-S txt/train_70.scp -H hmm0/macros -H hmm0/hmmdefs -M hmm1 ph/monophones0
```

Tác dụng của việc này là tải tất cả các mô hình trong hmm0 được liệt kê trong danh sách các mô hình ở monophones0. Sau đó, chúng được ước tính lại bằng cách sử dụng dữ liệu được liệt kê trong train_70.scp và tập hợp mô hình mới được lưu trữ trong thư mục hmm1. Tiếp tục huấn luyện thêm cho mô hình.

```
bin/HERest -C cfg/HERest.cfg -I mlf/phones0.mlf -t 250.0 150.0 1000.0
-S txt/train_70.scp -H hmm1/macros -H hmm1/hmmdefs -M hmm2
ph/monophones0

bin/HERest -C cfg/HERest.cfg -I mlf/phones0.mlf -t 250.0 150.0 1000.0
-S txt/train_70.scp -H hmm2/macros -H hmm2/hmmdefs -M hmm3
ph/monophones0
```

Đến đây mô hình cơ bản đã được huấn luyện xong. Tiếp theo sẽ huấn luyện với **Feedforward-DNN**. Với đầu vào tương tự như HMM, thực hiện qua các bước sau:

1. **Chuyển file mfc ở dạng nhị phân sang dạng text**

```
perl pl/mkListMFCC.pl wav txt/log_mfcc.sh
sh ./txt/log_mfcc.sh
```

2. **Chuyển đổi mfc sang vec-tơ mean + vec-vơ varian**

```
perl ./pl/createMeanAndVar.pl wav dnn_mean
```

3. **Tạo file log chứa dữ liệu từ tất cả các file đặc trưng ở folder dnn_mean**

```
perl ./pl/createFeature.pl dnn_mean dnn/log.dnn
```

4. **Chia tập dữ liệu thành train và test với tỉ lệ 8:2**
5. **Tiến hành training**

Ở đây ta xây dựng bộ phân loại qua 3 lớp ẩn rồi đưa mô hình vào huấn luyện

```
# Define classifer
classifier = tf.estimator.DNNClassifier(
        feature_columns=my_feature_columns,
        hidden_units=[1024,512,256],
        n_classes=45,
        model_dir='mode_dnn/')

# Train the Model
classifier.train(
        input_fn=lambda:train_input_fn(train_x, train_y,
        args.batch_size),
        steps=args.train_steps)
```

Tiến hành gọi lệnh training với batch size là 100 và số lần chạy là tăng dần từ 1000-50000 để quan sát và có được kết quả tốt, phù hợp nhất.

```
python3 dnn.py --batch_size 100 --train_steps 30000 >
100_30000_log.txt
```

4.3. Thực nghiệm và đánh giá

4.3.1. Độ đo đánh giá

Với HMM, HTK sử dụng HResults đọc một tập hợp các tệp chứa nhãn và so sánh chúng với các tệp phiên âm tham chiếu tương ứng. Để phân tích đầu ra nhận dạng giọng nói, việc so sánh dựa trên quy trình căn chỉnh chuỗi Dynamic Programming-based (DP). Khi được sử dụng để tính toán độ chính xác của câu bằng cách sử dụng DP, đầu ra cơ bản là thống kê nhận dạng cho toàn bộ tệp ở định dạng như bên dưới.

Dòng đầu tiên cung cấp độ chính xác ở cấp độ câu dựa trên tổng số tệp chứa nhãn giống với tệp phiên âm. Dòng thứ hai là độ chính xác của từ dựa trên sự trùng khớp DP giữa các tệp nhãn và phiên âm. Ở dòng thứ hai, H là số nhãn đúng, D là số lần xóa, S là số lần thay thế, I là số lần chèn và N là tổng số nhãn trong các tệp phiên âm đã xác định. Số phần trăm nhãn được nhận dạng chính xác và độ chính xác của nó được tính toán bởi công thức sau:

$$\%Correct = \frac{H}{N} \times 100\% \,;\, Accuracy = \frac{H - I}{N} \times 100\%$$

Với Feedforward-DNN, độ chính xác (hay còn gọi là accuracy) sẽ được sử dụng trong trường hợp này. Độ chính xác là một thước đo để đánh giá các mô hình phân loại. Nói chính xác hơn thì độ chính xác là một phần nhỏ của các dự đoán mà mô hình đã đúng. Về mặt hình thức, độ chính xác được định nghĩa là bằng tỉ lệ giữa số lượng dự đoán chính xác và tổng tất cả các dự đoán, công thức như sau:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Đối với phân loại nhị phân, độ chính xác cũng có thể được tính theo mặt tích cực (Positive) và tiêu cực (Negative) với công thức như sau:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Cụ thể:

- TP (True Positive): đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Positive (dự đoán đúng)
- TN (True Negative): đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Negative (dự đoán đúng)
- FP (False Positive): đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Positive (dự đoán sai)
- FN (False Negative): đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Negative (dự đoán sai)

True/False thể hiện cho tính chính xác khi phân loại của mô hình, True đồng nghĩa với việc mô hình phân loại đúng và ngược lại. Positive/Negative là lớp mà đối tượng được mô hình xếp vào, Positive có nghĩa rằng mô hình phân đối tượng vào lớp Positive và ngược lại.

Trong khuôn khổ luận văn với hai mô trình HMM và Feedforward-DNN, ta đều cùng quan sát độ đo là accuracy để xem xét mô hình được xây dựng để nhận dạng giọng nói tiếng Việt có chính xác và hiệu quả hay không.

Thực nghiệm và so sánh

❖ **HMM**

Kiểm tra trên tập train: thực thi đoạn lệnh dưới đây ta sẽ thu được kết quả như như hình bên dưới.

```
bin/HVite -C cfg/HVite.cfg -H hmm3/macros -H hmm3/hmmdefs -S
txt/train_70.scp -i recout_train.mlf -w txt/wdnet.txt txt/dict.dct
txt/wlist.txt
bin/HResults -f -t -I mlf/phones0.mlf txt/wlist.txt recout_train.mlf
> result_train.mlf
```

```
---------------------- Overall Results -------------------------
SENT: %Correct=95.76 [H=3504, S=155, N=3659]
WORD: %Corr=95.76, Acc=95.76 [H=3504, D=0, S=155, I=0, N=3659]
==================================================================
```

Hình 4.4: Kết quả thống kê trên tập huấn luyện

Kết quả từ hình 4.4 cho thấy, dòng bắt đầu bằng SENT: cho biết rằng trong số 3659 câu nói huấn luyện, có 3504 câu được nhận dạng chính xác chiếm tỉ lệ 95,76% và 155 câu bị nhận dạng nhầm qua người khác. Vì HMM ở trường hợp này được tiếp cận theo hướng nhận dạng ra người nói (không theo khuynh hướng nhận dạng văn bản), nên sẽ bỏ qua việc thống kê trên từ vựng.

Kiểm tra trên tập test

```
bin/HVite -C cfg/HVite.cfg -H hmm3/macros -H hmm3/hmmdefs -S
txt/test_30.scp -i recout_test.mlf -w txt/wdnet.txt txt/dict.dct
txt/wlist.txt
bin/HResults -f -t -I mlf/phones0.mlf txt/wlist.txt recout_test.mlf >
result_test.mlf
```

Sau khi chạy thống kê thu được kết quả như sau:

```
---------------------- Overall Results -------------------------
SENT: %Correct=93.04 [H=321, S=24, N=345]
WORD: %Corr=93.04, Acc=93.04 [H=321, D=0, S=24, I=0, N=345]
==================================================================
```

Hình 4.5: Kết quả thống kê trên tập kiểm thử

Trong số 345 câu nói kiểm thử, có 321 câu được nhận dạng chính xác chiếm tỉ lệ 93,04% và có 24 câu bị nhận dạng sai (hình 4.5). Kết quả từ tập kiểm thử có chênh lệch một ít so với kết quả thống kê từ dữ liệu huấn luyện, nhận thấy rằng đây

chính là kết quả chính xác cuối cùng của mô hình. Tỉ lệ chính xác cao, cho thấy mô hình xây dựng chạy nhận dạng tốt, phù hợp với bộ dữ liệu xây dựng.

❖ **Feedforward-DNN**

Với Feedforward-DNN ta cũng dựa trên độ chính xác (accuracy) để đánh giá mô hình. Tiến hành chạy thống kê mô hình trên bộ dữ liệu kiểm thử với câu lệnh sau:

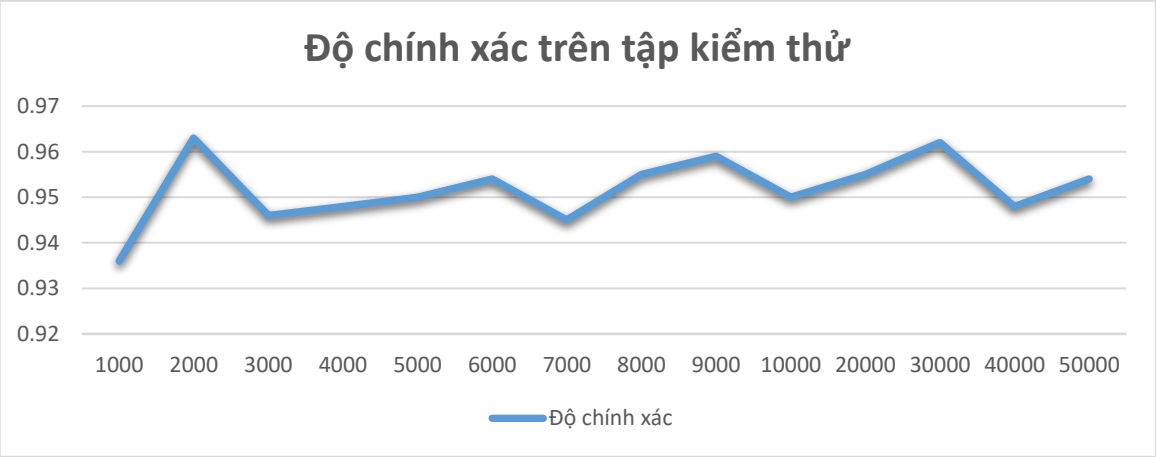
```
eval_result = classifier.evaluate(
        input_fn=lambda:eval_input_fn(test_x, test_y,
                                                args.batch_size))

print('\nTest set accuracy: {accuracy:0.3f}\n'.format(**eval_result))
```

Tương ứng với các lần tăng dần chạy ta có được kết quả trong bảng 4.4 như sau:

Bảng 4.3: Độ chính xác của mô hình qua số lần chạy training

Lần chạy	Độ chính xác
1000	0.936
2000	0.963
3000	0.946
4000	0.948
5000	0.950
6000	0.954
7000	0.945
8000	0.955
9000	0.959
10000	0.950
20000	0.955
30000	0.962
40000	0.948
50000	0.954



Hình 4.6: Biến thiên độ chính xác theo số lần chạy mô hình

Quan sát biểu đồ ở hình 4.6 ở các lần chạy khác nhau, ta thấy mô hình ban đầu cho kết quả độ chính xác khá cao 93,6%. Đồng thời khi tăng số lần chạy mô hình lên dần độ chính xác cũng thay đổi nhưng cũng không đáng kể. Qua 14 lần chạy training, nhận thấy độ chính xác trung bình của mô hình khoảng 95%. So với độ chính xác từ mô hình HMM, Feedforward-DNN có lớn hơn đôi chút.

4.3.2. Phân tích và đánh giá

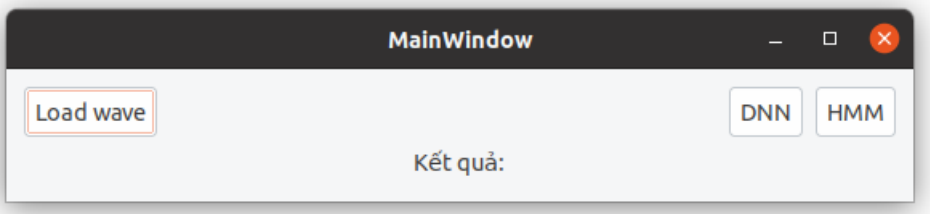
Kết quả từ việc chạy huấn luyện và kiểm thử mô hình cho thấy với HMM và Feedforward-DNN kết quả đạt khá cao 93.04%, 95% trên cùng bộ dữ liệu. Bộ dữ liệu xây dựng cũng chứng minh được độ tin cậy của mình khi hoạt động tốt với hai mô hình nhận dạng. Từ đó, nhận thấy với phương pháp học máy hiện đại, việc xây dựng mô hình nhanh chóng và dễ dàng, ít cấu hình hơn nhiều so với sử dụng bộ công cụ HTK. Bên cạnh đó, cách xây dựng phát triển mô hình cũng trực quan dễ hiểu, có thể chỉnh sửa để phát huy hơn nữa mà kết quả mang lại cũng khả thi hơn khi xem xét trên cùng tập dữ liệu đào tạo.

4.4. Chương trình demo

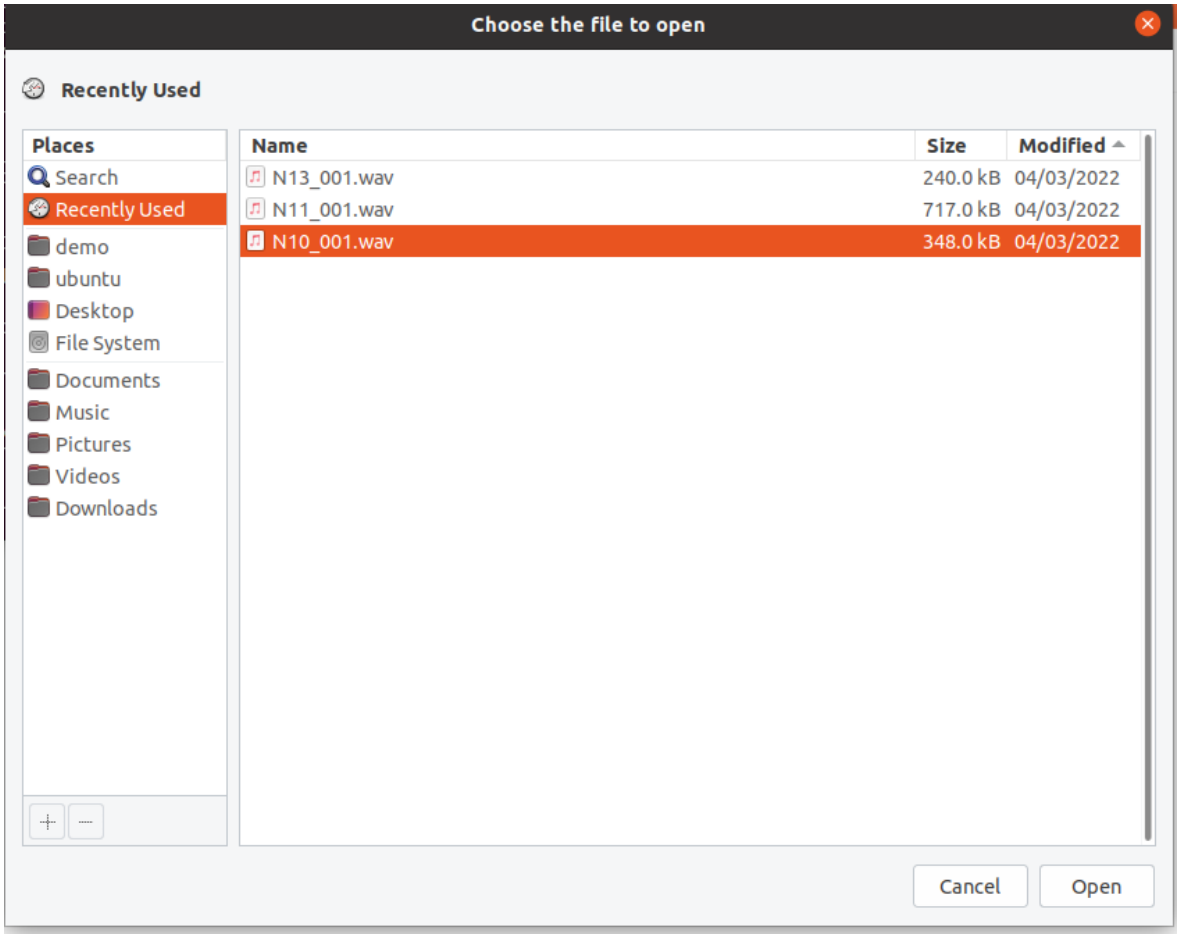
Nhận dạng người nói là một mảng nghiên cứu lớn, do thời gian hạn hẹp nên chỉ tập trung vào chứng minh khái niệm cho lĩnh vực này. Xây dựng chương trình demo chỉ mang tính minh họa và xác thực cho kết quả thực nghiệm, làm tiền đề cho xây dựng ứng dụng về sau. Cụ thể với hai mục tiêu:

1) Kiểm tra độ chính xác của mô hình: đã nhận dạng được đúng người với giọng nói hay chưa.
2) So sánh độ chính xác của hai mô hình HMM và Feedforward-DNN.

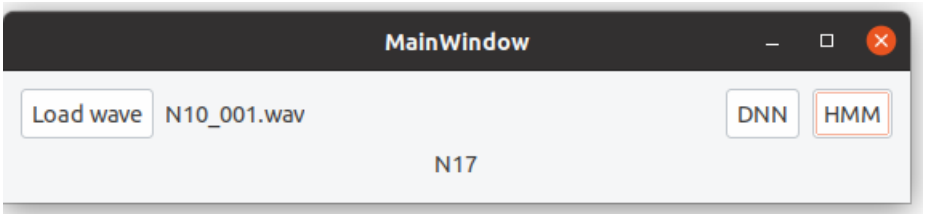
Đầu tiên khởi động chương trình ta được giao diện như Hình 4.7. Giao diện đơn giản bao gồm nút Load wave bên trái dùng để mở giao diện chọn file âm thanh cần nhận dạng, nút DNN và HMM ở bên phải dùng để chọn mô hình dự đoán. Và cuối cùng kết quả mà mô hình dự đoán được sẽ được hiển thị ở phần giữa của chương trình. Hình ảnh minh họa chương trình Demo như bên dưới (Hình 4.7 - 4.10).



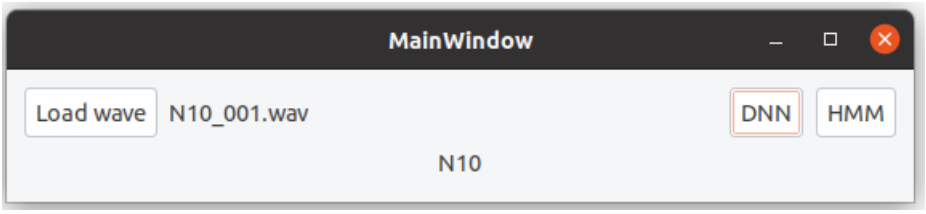
Hình 4.7: Giao diện chương trình demo



Hình 4.8: Chọn file âm thanh để tiến hành nhận dạng



Hình 4.9: Trường hợp nhận dạng với HMM



Hình 4.10: Trường hợp nhận dạng với Feedforward-DNN

Kết quả từ chương trình chạy demo cho thấy, với HMM (hình 4.9) mô hình đã dự đoán chưa chính xác người nói. Cụ thể nhãn chính xác của file ghi âm là N10

nhưng mô hình dự đoán ra N17. Ngược lại, DNN đã dự đoán chính xác nhãn phân loại, chứng tỏ rằng mặc dù độ chính xác của hai mô hình có sự chênh lệnh khá nhỏ nhưng thực tế DNN lại hoạt động hiệu quả hơn.

Phần demo luận văn thực hiện vẫn còn khá đơn giản khi chỉ mới diễn đạt được tính đúng sai, chưa hiển thị các thống kê, đánh giá khác,... Tuy nhiên đã đạt được mục tiêu nhận dạng người nói của đề tài.

CHƯƠNG 5: KẾT LUẬN

5.1. Các đóng góp của luận văn

Luận văn "**Nhận dạng người nói theo tiếp cận máy học hiện đại**" nghiên cứu các thuật toán sử dụng trong việc nhận dạng người nói, bộ dữ liệu sử dụng trong quá trình huấn luyện và kiểm thử mô hình, đồng thời là độ đo và cách phương pháp đánh giá độ chính xác, sai số của mô hình. Quá trình nghiên cứu đã đạt được nhiều mục tiêu đề ra như sau:

- Nghiên cứu mô hình nhận dạng người nói sử dụng mô hình HMM và Feedforward-DNN.
- Xây dựng được bộ dữ liệu giọng nói tiếng Việt ở khu vực Tây Ninh phục vụ cho việc huấn luyện và kiểm thử mô hình.
- Sử dụng bộ dữ liệu kết hợp với mô hình xây dựng kiểm chứng độ chính xác, phù hợp trong các điều kiện, tình huống đề xuất.
- Độ chính xác của mô hình xây dựng với HMM, Feedforward-DNN lần lượt là 93.04% và 95%.
- Áp dụng mô hình chạy dự đoán người nói với kết quả khả quan.
- Thực nghiệm với bộ dữ liệu đạt kết quả cao chứng tỏ độ tin cậy, phù hợp để sử dụng cho việc thực nghiệm đánh giá các mô hình khác.
- Có thể ứng dụng mô hình để nhận diện giọng nói, sinh trắc học; kết hợp với các phương pháp sinh trắc học khác như mống mắt, vân tay,... để định danh người chính xác, xác thực nhanh chóng không cần chứng thực. Mức độ sai sót của hệ thống có thể thay đổi nhưng không đáng kể.

5.2. Kết luận và hướng phát triển

Hạn chế luận văn

- Vì thời điểm thu thập dữ liệu này trùng với dịch Covid diễn ra phức tạp nên bộ dữ liệu chưa được đa dạng. Bộ dữ liệu vẫn còn nhỏ và mức độ phân bố rộng

khắp về mặt độ tuổi, vùng miền chưa cao. Bên cạnh đó về chất lượng giọng nói của người tham dự chưa phong phú về biểu cảm, cảm xúc khi nói.

- Phần ứng dụng demo còn đơn giản do quỹ thời gian hạn hẹp.
- Đây mới chỉ là bài nghiên cứu, chưa áp dụng vào thực tế.

Vấn đề kiến nghị và hướng đi tiếp theo của nghiên cứu:

- Đưa kết quả nghiên cứu vào ứng dụng thực tế.
- Tìm thêm các cách xử lý tối ưu dữ liệu, tìm hoặc tự xây dựng hoặc tối ưu hóa mô hình, hiệu chỉnh hơn nữa độ chính xác của mô hình. Nghiên cứu tìm kiếm các phương pháp tiếp cận khác.
- Làm phong phú hơn cho bộ dữ liệu như: thu thập giọng nói ở nhiều vùng, tỉnh thành, mở rộng phạm vi độ tuổi, độ cao thấp của giọng nói, ảnh hưởng của môi trường (yên lặng, ồn ào,...).

DANH MỤC TÀI LIỆU THAM KHẢO

[1] C. T. Tran, D. T. Nguyen and H. T. Hoang, "Deep Representation Learning for Vietnamese Speaker Recognition," in *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*, 2021.

[2] D. D. Thi Thu, L. T. Van, Q. N. Hong and H. P. Ngoc, "Text-dependent speaker recognition for vietnamese," in *2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR)*, 2013.

[3] Son T. Nguyen, Viet D. Lai, Quyen Dam-Ba, Anh Nguyen-Xuan, and Cuong Pham, "ietnamese Speaker Authentication Using Deep Models," in *Proceedings of the Ninth International Symposium on Information and Communication Technology (SoICT 2018)*, 2018.

[4] Nguyen Duc Nam and Hieu Trung Huynh, "Speaker Diarization in Vietnamese Voice," in *Communications in Computer and Information Science book series*, 2021.

[5] R. Jahangir et al, "Text-Independent Speaker Identification Through Feature Fusion and Deep Neural Network," *IEEE Access*, vol. 8, pp. 32187-32202, 2020.

[6] N. N. An, N. Q. Thanh and Y. Liu, "Deep CNNs With Self-Attention for Speaker Identification," *IEEE Access*, vol. 7, pp. 85327-85337, 2019.

[7] Bunrit, Supaporn & Inkian, Thuttaphol & Kerdprasop, Nittaya & Kerdprasop, Kittisak, "Text-Independent Speaker Identification Using Deep Learning Model of Convolution Neural Network," *International Journal of Machine Learning and Computing*, vol. 9, pp. 143-148, 2019.

[8] Shahin, Ismail & Nassif, Ali & Hamsa, Shibani, "Novel Cascaded Gaussian Mixture Model-Deep Neural Network Classifier for Speaker Identification in Emotional Talking Environments," *Neural Computing and Applications*, 2020.

[9] H. Muckenhirn, M. Magimai.-Doss and S. Marcell, "Towards Directly Modeling Raw Speech Signal for Speaker Verification Using CNNS," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[10] Liu, Jung-Chun & Leu, Fang-Yie & Lin, Guan-Liang & Susanto, Heru, "An MFCC-based text-independent speaker identification system for access control," *Concurrency and Computation: Practice and Experience,* 2017.

[11] Yadav, Sarthak & Rai, Atul, "Learning Discriminative Features for Speaker Identification and Verification," *Interspeech 2018,* pp. 2237-2241, 2018.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional," *arXiv preprint,* 2014.

[13] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech 2017,* 2017.

[14] Lukic, Yanick and Vogt, Carlo and Dürr, Oliver and Stadelmann, Thilo, "Speaker identification and clustering using convolutional neural networks," in *2016 IEEE Iinternational Workshop On Machine Learning For Signal Processing,* 2016.

[15] Nilu Singh, R.A. Khan, and Raj Shree, "Applications Of Speaker Recognition," in *International Conference on Modelling, Optimisation and Computing (ICMOC 2012),* 20120.

[16] B. Copeland, "Artificial intelligence," Encyclopedia Britannica, 11 August 2020. [Online]. Available: https://www.britannica.com/technology/artificial-intelligence. [Accessed 15 April 2021].

[17] M. H. Buur, "Is the quality of a DAC related to software implementation?," Sound Design, StackExchange, September 14, 2016.

[18] Saranga-K-Mahanta-google, arvindpdmn, "Audio Feature Extraction," Devopedia, [Online]. Available: https://devopedia.org/audio-feature-extraction. [Accessed 23 May 2021].

[19] S. K. Singh, "Features And Techniques For Speaker Recognition," M. Tech. Credit Seminar Report, Electronic Systems Group, EE Dept, IIT Bombay.

[20] S.B.Dhonde, S.M.Jagade, "Feature Extraction Techniques in Speaker Recognition: A Review," *International Journal on Recent Technologies in Mechanical and Electrical Engineering (IJRMEE),* vol. 2, no. 5, pp. 104-106, 2015.

[21] W. contributors, "Fourier transform," [Online]. Available: https://en.wikipedia.org/wiki/Fourier_transform. [Accessed 31 March 2022].

[22] V. H. Tiệp, "Machine Learning cơ bản," 26 December 2016. [Online]. Available: https://machinelearningcoban.com/2016/12/26/introduce/.

[23] Javatpoint, "Applications of Machine learning," [Online]. Available: https://www.javatpoint.com/applications-of-machine-learning.

[24] K. Krzyk, “Coding Deep Learning For Beginners,” Towards Data Science, 25 July 2018. [Trực tuyến]. Available: https://towardsdatascience.com/coding-deep-learning-for-beginners-types-of-machine-learning-b9e651e1ed9d.

[25] M. T. Jones, "Models for machine learning," IBM Developer, 5 December 2017. [Online]. Available: https://developer.ibm.com/articles/cc-models-machine-learning/.

[26] Wikipedia contributors, "Artificial neural network," Wikipedia, The Free Encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Accessed 24 May 2021].

[27] E. Kavlakoglu, “AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What’s the Difference?,” IBM, [Trực tuyến]. Available: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks.

[28] G. L. Team, "Types of Neural Networks and Definition of Neural Network," Great Learning, 25 September 2021. [Online]. Available: https://www.mygreatlearning.com/blog/types-of-neural-networks/.

[29] Daniel Jurafsky & James H. Martin, "Hidden Markov Models," in *Speech and Language Processing*, 2021.

[30] "HTK," [Online]. Available: https://htk.eng.cam.ac.uk/.

[31] Thanh T Nguyen, Binh A Nguyen, Manh Hoang, Tung V Nguyen, Giao N Pham, "A method for speech Vietnamese recognition based on deep learning," *International Journal of Multidisciplinary Research and Growth Evaluation*, vol. 2, no. 4, pp. 152-156, 2021.

[32] Quang H. Nguyen and Tuan-Dung Cao, "A Novel Method for Recognizing Vietnamese Voice Commands on Smartphones with Support Vector Machine and Convolutional Neural Networks," *Hindawi Wireless Communications and Mobile Computing*, 2020.

[33] Quoc Truong Do, Pham Ngoc Phuong, Hoang Tung Tran, Chi Mai Luong, "Development Of High-Performance And Large-Scale VietNamese Automatic Speech Recognition Systems," *Journal of Computer Science and Cybernetics*, vol. 34, pp. 335-348, 2018.

[34] "The Association for Vietnamese Language and Speech Processing," [Online]. Available: https://vlsp.org.vn/vlsp2018/eval.

[35] Phan Duy Hung, Truong Minh Giang, Le Hoang Nam, Phan Minh Duong, "Vietnamese Speech Command Recognition using Recurrent Neural Networks," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, 2019.

BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung luận văn/luận án qua phần mềm Kiểm tra tài liệu (https://kiemtratailieu.vn) một cách trung thực và đạt kết quả mức độ tương đồng **6%** toàn bộ nội dung luận văn/luận án. Bản luận văn/luận án kiểm tra qua phần mềm là bản cứng luận văn/luận án đã nộp bảo vệ trước hội đồng. Nếu sai sót tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

TP. Hồ Chí Minh, ngày 04 tháng 05 năm 2022

Học viên thực hiện luận văn

Trần Thị Nhi An

KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

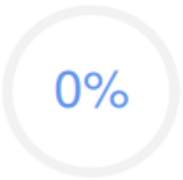
Thông tin tài liệu

Tên tài liệu:	Nhận dạng người nói theo tiếp cận máy học hiện đại
Tác giả:	Trần Thị Nhi An
Điểm trùng lặp:	6
Thời gian tải lên:	21:45 13/05/2022
Thời gian sinh báo cáo:	21:47 13/05/2022
Các trang kiểm tra:	86/86 trang

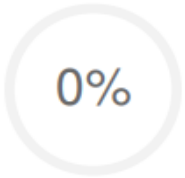
Kết quả kiểm tra trùng lặp

Có **6%** nội dung trùng lặp

Có **94%** nội dung không trùng lặp



Có **0%** nội dung người dùng loại trừ



Có **0%** nội dung hệ thống bỏ qua

Nguồn trùng lặp tiêu biểu

123docz.net tailieu.vn luanvan.moet.gov.vn

Học viên

Trần Thị Nhi An

Người hướng dẫn khoa học

PGS.TS Vũ Hải Quân