TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xác minh người nói trong tiếng Việt với học sâu

VI THÀNH ĐẠT
dat.vt164803@sis.hust.edu.vn

Ngành: Khoa học Máy tính

Giảng viên hướng dẫn: TS. Nguyễn Thị Thu Trang _____
ThS. Đỗ Tuấn Anh _____

Bộ môn: Khoa học máy tính và Công nghệ phần mềm
Viện: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 06/2021

Phiếu giao nhiệm vụ đồ án tốt nghiệp

Thông tin về sinh viên

- **Họ tên:** Vi Thành Đạt
- **Email:** dat.vt164803@sis.hust.edu.vn
- **Điện thoại liên lạc:** 0336863831
- **Lớp:** CNTT2.02-K61
- **Hệ đào tạo:** Đại học chính quy
- **Đồ án tốt nghiệp được thực hiện tại:** viện Công nghệ thông tin và truyền thông.
- **Thời gian làm đồ án tốt nghiệp:** Từ ngày 10/2/2021 đến 15/06/2021.

Mục đích nội dung của đồ án tốt nghiệp

Tìm hiểu, thử nghiệm các mô hình xác minh người nói, xây dựng bộ dữ liệu xác minh người nói tiếng Việt. Phát triển và cải tiến mô hình xác minh người nói nhằm nâng cao độ chính xác với dữ liệu nhỏ.

Các nhiệm vụ cụ thể của đồ án tốt nghiệp

1. Tìm hiểu bài toán xác minh người nói.
2. Tìm hiểu các phương pháp tốt nhất hiện nay cho bài toán xác minh người nói.
3. Đề xuất các mô hình mới hiệu quả hơn với ít dữ liệu huấn luyện.
4. Xây dựng bộ dữ liệu xác minh người nói tiếng Việt.
5. Thực nghiệm và đánh giá kết quả các mô hình đề xuất.

Lời cam đoan của sinh viên

Tôi - *Vi Thành Đạt* - cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Nguyễn Thị Thu Trang* và *ThS. Đỗ Tuấn Anh*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN - bao gồm hình ảnh, bảng biểu, số liệu và các câu từ trích dẫn - đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế nhà trường.

. *Hà Nội*, ngày 15 tháng 06 năm 2021
Sinh viên

Vi Thành Đạt

Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành và cho phép bảo vệ:

. . . . . .
. . . . . .
. . . . . .

. *Hà Nội*, ngày 15 tháng 06 năm 2021
Giảng viên hướng dẫn

TS. Nguyễn Thị Thu Trang

Lời cảm ơn

Lời đầu tiên tôi xin gửi lời cảm ơn chân thành nhất tới các thầy giáo, cô giáo trường Đại học Bách khoa Hà Nội, viện Công nghệ thông tin và Truyền thông, bộ môn Khoa học máy tính đã có môi trường học tập tốt nhất để giúp tôi có kiến thức bổ ích và những kinh nghiệm quý báu trong suốt quá trình học tập và rèn luyện.

Đặc biệt tôi xin gửi lời cảm ơn sâu sắc nhất tới cô TS. Nguyễn Thị Thu Trang – Giảng viên bộ môn Công nghệ phần mềm và thầy ThS. Đỗ Tuấn Anh - Giảng viên bộ môn Khoa học máy tính, viện Công nghệ thông tin và Truyền thông, trường Đại học Bách khoa Hà Nội đã tận tình hướng dẫn tôi trong quá trình làm đồ án tốt nghiệp. Thầy cô dẫn dắt tôi không chỉ bằng kiến thức mà còn cả sự kiên trì và niềm đam mê. Tôi cảm thấy thật sự may mắn khi được học và phát triển dưới sự hướng dẫn của thầy cô.

Tôi xin gửi lời cảm ơn đến các đồng nghiệp của tôi tại VNG, họ đã đóng góp một phần thời gian, nguồn lực và kinh nghiệm của mình để giúp tôi hoàn thành đồ án này.

Cuối cùng, tôi xin chân thành cảm ơn gia đình, bạn bè, các bạn làm cùng nhóm đồ án tốt nghiệp đã luôn động viên, hỗ trợ và tạo những điều kiện tốt nhất để tôi có thể hoàn thành đồ án tốt nghiệp này.

Tuy nhiên, do thời gian và kiến thức của tôi cũng còn nhiều hạn chế nên chắc chắn không tránh khỏi những thiếu sót vì vậy tôi rất mong nhận được sự đóng góp ý kiến của các thầy giáo, cô giáo và toàn thể các bạn đọc.

Tôi xin chân thành cảm ơn!

Tóm tắt

Xác minh người nói (Speaker Verification) là quá trình tự động xác nhận danh tính một người bằng việc sử dụng thông tin độc nhất của người nói có trong tín hiệu giọng nói. Hiện nay, xác định danh tính một cá nhân cần kết hợp nhiều yếu tố như mật khẩu, mống mắt, khuôn mặt, ... nhằm tăng cường tính bảo mật. Trên thế giới, phương pháp học sâu đang được ưa chuộng cho xác minh người nói với hiệu suất cao. Trong tiếng Việt, so với các ngôn ngữ khác, lượng dữ liệu huấn luyện còn rất hạn chế chưa đủ để xây dựng mô hình chất tượng tốt cho bài toán xác minh người nói. Vì vậy, đồ án tập trung vào xây dựng mô hình học sâu và bộ dữ liệu cho bài toán xác minh người nói tiếng Việt.

Đồ án nghiên cứu và sử dụng phương pháp học chuyển tiếp (Transfer Learning) nhằm sử dụng kiến thức học được của mô hình tiếng Anh cho tiếng Việt. Đồ án cũng sử dụng phương thức tối ưu SGD với tính khái quát hoá tốt thay cho Adam trong mô hình cơ sở. Cùng với đó, đồ án cải tiến hàm mất mát nguyên mẫu góc (Angular Prototypical - AP) bằng cách thêm hệ số phạt biên nhằm tăng tính phân tách người nói của mô hình. Kết quả đánh giá cho thấy mô hình đề xuất đạt tỉ lệ lỗi bằng nhau (Equal Error Rate - EER) 3.115% vượt trội so với 7.602% của mô hình cơ sở cho bài toán xác minh người nói tiếng Việt. Để xây dựng dữ liệu cho bài toán, đồ án kết hợp sử dụng các bộ dữ liệu ZaloAI, VIVOS, VLSP và CommonVoice. Lỗi trong các bộ dữ liệu được loại bỏ bằng cách phân tích ma trận tương đồng của biểu diễn các câu nói.

Kết quả của đồ án đã được tổng hợp và nộp tại Hội nghị Châu Á Thái Bình Dương về Ngôn ngữ, Thông tin và Tính toán (PACLIC) với tiêu đề "Speaker Verification Model with Angular Margin Prototypical Loss for Low-Resource Languages and Vietnamese Datasets".

Trong tương lai, đồ án sẽ thu thập thêm dữ liệu người nói từ nguồn Youtube và triển khai ứng dụng hoặc API xác minh người nói. Ngoài ra, đồ án sẽ đưa nhóm người có giọng nói tương tự vào cùng một mini-batch để huấn luyện mô hình có tính phân tách cao hơn.

Mục lục

Danh sách hình vẽ

Danh sách bảng

Chương 1

Bài toán xác minh người nói

1.1 Giới thiệu bài toán xác minh người nói

Xác minh người nói (Speaker Verfication) là một trong hai ứng dụng lớn của nhận dạng người nói (Speaker Recognition). Trong phần này, đồ án sẽ giới thiệu về bài toán nhận dạng người nói và bài toán xác minh người nói.

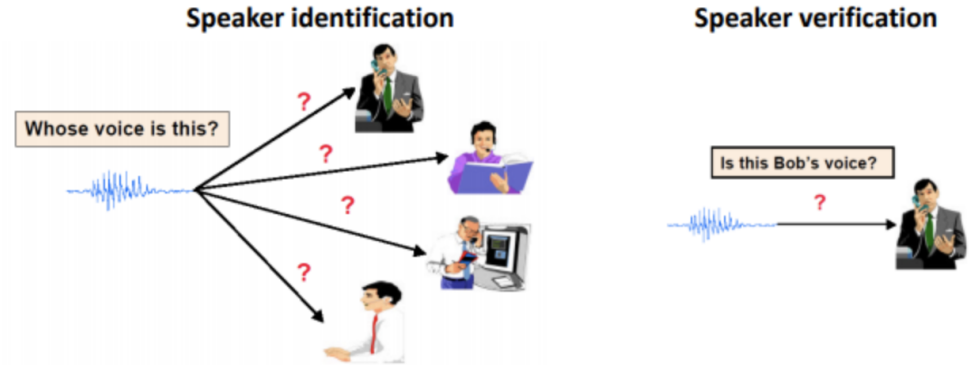
1.1.1 Nhận dạng người nói

Nhận dạng người nói là quá trình tự động nhận dạng người đang nói bằng cách sử dụng thông tin độc nhất của người nói đó có trong tín hiệu giọng nói. Nhận dạng người nói được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau, tuy nhiên cũng gặp không ít khó khăn khi triển khai trong thực tế. Do vậy, nghiên cứu bài toán nhận dạng người nói rất được quan tâm bởi nhiều nhà khoa học trên thế giới. Một số ứng dụng của bài toán có thể kể đến như: (i) Bảo mật cho các hệ thống tài chính, ngân hàng: người dùng dùng giọng nói kết hợp với các lớp bảo mật khác cho xác thực để tăng tính bảo mật khi giao dịch. (ii) Tăng trải nghiệm khách hàng trong tổng đài chăm sóc khách hàng. (iii) Xác định danh tính tội phạm trong an ninh khi thu được dữ liệu giọng nói. (iv) Kết hợp với các hệ thống nhận dạng tiếng nói để xây dựng ứng dụng gỡ băng cuộc họp.

1.1.2 Xác minh người nói

Dựa vào ứng dụng, nhận dạng người được phân loại thành xác định người nói (Speaker Identitfication) và xác minh người nói (Hình 1.1). Trong xác định người nói, một đoạn tiếng nói từ một người không xác định được phân tích và so sánh với biểu diễn giọng nói của những người đã biết. Người này được xác định là

người có biểu diễn giọng nói phù hợp nhất với câu nói đầu vào. Trong xác minh người nói, một người lạ xác nhận một danh tính đã biết; đoạn tiếng nói của người này được so sánh với biểu diễn giọng nói của danh tính đang được xác nhận. Nếu điểm tương đồng đủ tốt, nghĩa là trên một ngưỡng nào đó, danh tính của người lạ được chấp nhận. Ngưỡng cao khiến những kẻ mạo danh khó được chấp nhận bởi hệ thống, nhưng có nguy từ chối nhầm người dùng hợp lệ. Ngược lại, ngưỡng thấp cho phép chấp nhận người dùng hợp lệ một cách nhất quán, nhưng có nguy cơ chấp nhận những người giả mạo. Xác minh người nói được ứng dụng phổ biến hơn xác định người nói do nhu cầu và dễ thực hiện hơn về mặt tính toán (một phép so sánh so với N phép so sánh trong xác định người nói). Thông thường, cải tiến trong một trong hai bài toán có thể được áp dụng sang bài toán còn lại.



Hình 1.1: Xác định người nói và xác minh người nói [1].

Xác minh người nói có tiềm năng ứng dụng trong xác thực cá nhân, bao gồm xác minh thẻ tín dụng, truy cập bảo mật qua điện thoại (di động, Internet) trong các trung tâm cuộc gọi. Ngoài ra, xác minh người nói còn được ứng dụng trong an ninh có thể kể đến như nhận dạng nghi phạm qua giọng nói, truy cập vào toà nhà hay các biện pháp an ninh quốc gia chống khủng bố bằng cách sử dụng giọng nói làm bảo mật cho các ứng dụng quan trọng. Với việc bảo mật thông tin cá nhân trở thành một vấn đề nóng hổi trong xã hội, các công ty Internet cũng có thể sử dụng xác minh người nói để ngăn chặn khả năng gian lận danh tính.

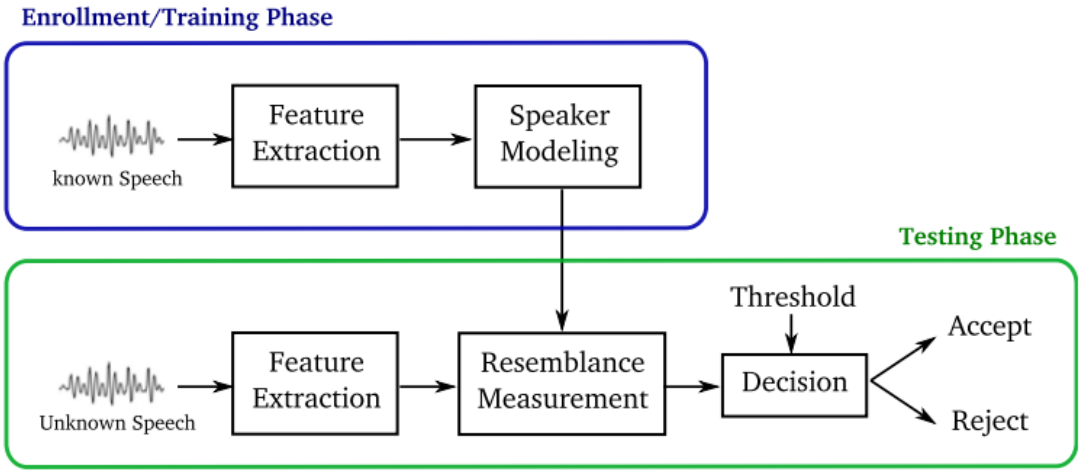
Thông thường, hệ thống xác minh người nói được chia thành 3 pha (Hình 1.2):

- Pha 1: Phát triển (Development). Trong pha này, mô hình có khả năng biểu diễn đặc trưng người nói được huấn luyện và tối ưu trên một cơ sở dữ liệu lớn các đoạn tiếng nói từ một nhóm nhiều người nói khác nhau.
- Pha 2: Ghi danh (Enrollment). Trong pha ghi danh, biểu diễn của người

[1] https://wiki.aalto.fi/display/ITSP/Speaker+Recognition+and+Verification

dùng mới được trích xuất bằng mô hình phát triển ở pha 1 và lưu trữ trong cơ sở dữ liệu để phục vụ cho pha 3.

- Pha 3: Kiểm tra (Testing). Trong pha này, được cung cấp danh tính đầu vào, hệ thống chỉ so sánh đoạn tiếng nói đầu vào và đoạn của danh tính trong hệ thống để đưa ra quyết định.



Hình 1.2: Tổng quan hệ thống xác minh người nói [2].

Trong thực tế, hiệu năng của hệ thống xác minh người nói bị suy giảm do sự khác biệt của các kênh và phiên giữa tín hiệu giọng nói trong pha ghi danh và pha kiểm tra. Các yếu tố làm ảnh hưởng tới tín hiệu giọng nói bao gồm:

- Sử dụng các loại micrô khác nhau khi thu tín hiệu đăng ký và kiểm tra.
- Điều kiện tiếng ồn và độ vang của môi trường.
- Sự khác biệt trong giọng nói của người nói ở các giai đoạn khác nhau của độ tuổi, sức khoẻ, phong cách nói và trạng thái cảm xúc.
- Các kênh truyền như các loại điện thoại di động khác nhau, micrô, giao thức truyền tín hiệu giọng nói qua Internet có thể làm thay đổi giọng.

Dựa vào sự tương đồng của các câu nói đầu vào, phương pháp giải quyết bài toán xác minh người nói có thể được chia thành hai loại: phụ thuộc văn bản (Text-Dependent Speaker Verification - TDSV) và không phụ thuộc văn bản (Text-Independent Speaker Verification - TISV). Các hệ thống TDSV yêu cầu người nói cung cấp các đoạn tiếng nói có nội dung thuộc một tập các từ hoặc câu được định sẵn; nội dung các câu nói phải được giữ nhất quán trong cả quá trình huấn luyện và xác minh. Ngược lại, TISV không yêu cầu người dùng phải

[2]https://wiki.aalto.fi/display/ITSP/Speaker+Recognition+and+Verification

thu theo bất cứ một văn bản nào. Với các đoạn tiếng nói ngắn, các hệ thống TDSV đã có thể đạt được hiệu suất nhận diện cao, trong khi TISV yêu cầu các câu nói dài hơn và một lượng lớn dữ liệu để huấn luyện các mô hình đáng tin cậy và đạt được hiệu suất tốt. Do sự tiện lợi của TISV so với TDSV, các hệ thống sử dụng trong thương mại hiện nay phần lớn là TISV.

1.2 Nghiên cứu xác minh người nói trên thế giới

Nghiên cứu về xác minh người nói hay nhận dạng người nói trên thế giới đã bắt đầu từ những năm 1980; các mô hình cho xác minh người nói có thể được phân vào 2 nhóm: các mô hình thống kê và các mô hình học sâu.

1.2.1 Mô hình thống kê

Mô hình xác minh người nói và nhận dạng người nói tự động đầu tiên được đề xuất năm 1995 bởi Reynolds và cộng sự [11]. Phương pháp sử dụng trong nghiên cứu là mô hình Gaussian hỗn hợp (Gaussian Mixture Model - GMM). GMM là tổng hợp của nhiều hàm mật độ xác suất Gaussian, thường được dùng để mô hình dữ liệu đa biến. Sử dụng GMM để mô hình hoá đặc trưng của một người nói thu được hàm mật độ xác suất phụ thuộc vào người nói. Xác suất tương đồng giữa GMM của một người nói và một câu nói bất kì có thể được tính nhờ hàm mật độ xác suất này. Với một ứng dụng xác minh người nói đơn giản, trong pha ghi danh, một GMM được tính toán cho mỗi người trong cơ sở dữ liệu. Trong pha kiểm tra, câu nói được cung cấp được so sánh với GMM của danh tính mà người nói cung cấp. Nếu điểm tương đồng vượt qua một ngưỡng nhất định, danh tính này được chấp nhận.

Một thời gian sau đó, nhận thấy phương pháp GMM yêu cầu quá nhiều dữ liệu cho mỗi người nói, phương pháp GMM-UBM ra đời. Về cơ bản, UBM (mô hình nền phổ quát - Universal Background Model) là một GMM lớn mô tả phân phối không phụ thuộc vào người nói từ đặc trưng tiếng nói của tất cả người nói trong cơ sở dữ liệu. Tại pha ghi danh, UBM được thích ứng cho mỗi người nói sử dụng phương pháp thích ứng Baysian. Phương pháp GMM-UBM cho kết quả vượt trội so với việc huấn luyện GMM độc lập cho mỗi người nói từ đầu với một lượng dữ liệu ít hơn.

Một trong những vấn đề với xác minh người nói là dữ liệu giọng nói huấn luyện và kiểm tra có thể có thời lượng khác nhau. Điều này đòi hỏi sự so sánh của

hai câu nói có độ dài khác nhau. Do đó, một trong những nỗ lực hướng tới việc xác minh người nói hiệu quả là có được biểu diễn với chiều cố định cho một câu nói duy nhất. Việc có được biểu diễn có chiều cố định cực kì hữu ích vì nhiều phương pháp phân loại khác nhau trong học máy có thể sử dụng các biểu diễn này để phân loại.

Một giải pháp hiệu quả để có được vectơ chiều cố định từ câu nói có thời lượng thay đổi là GMM siêu vec-tơ (Supervector), về cơ bản là một vectơ lớn thu được bằng cách ghép các tham số trong mô hình GMM. Phương pháp ứng dụng siêu vec-tơ trong xác minh người nói được đề xuất năm 2003 bởi Kenny và cộng sự [12], thúc đẩy nhiều phương pháp thích ứng mô hình mới. Cộng đồng nhận ra các siêu vec-tơ với chiều lớn là một nền tảng tốt để thiết kế các phương thức loại bỏ thông tin về kênh và phiên như đã nói trong mục trước để có được vec-tơ biểu diễn người nói hiệu quả hơn. Các phương pháp thống trị hoạt động trên không gian siêu vec-tơ dựa trên phân tích nhân tố (Factor Analysis) và máy vec-tơ hỗ trợ (Support Vector Machine). Các phương pháp sử dụng GMM-UBM và siêu vec-tơ vẫn được coi là phương pháp hiện đại nhất cho bài toán xác minh người nói cho đến nửa cuối của những năm 2010 với sự phát triển mạnh mẽ của học sâu.

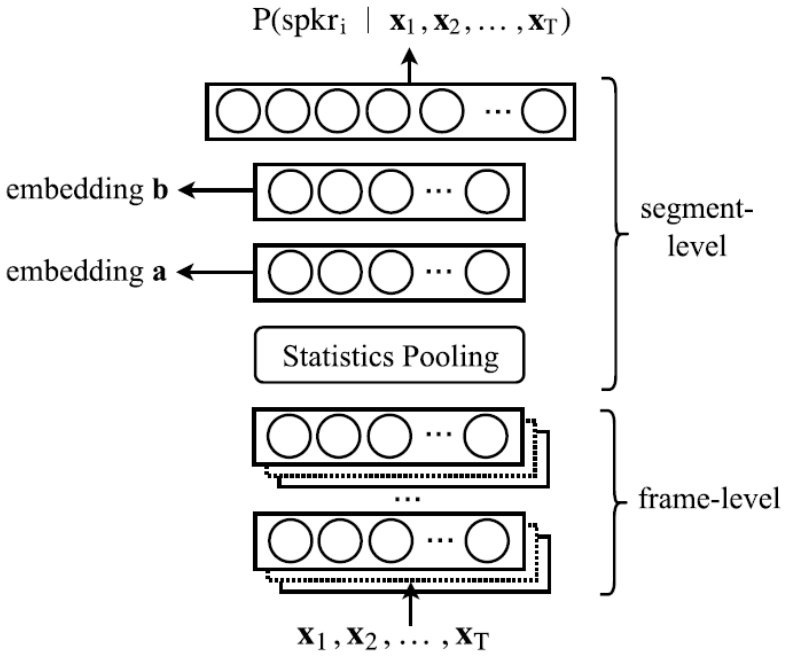
1.2.2 Mô hình học sâu

Các mô hình học sâu hiện đại giải quyết bài toán xác minh người nói thường gồm ba phần chính (Hình 1.3):

- Một mạng nơ-ron làm công cụ trích xuất biểu diễn người nói cho một khung đặc trưng tiếng nói.
- Lớp tổng hợp dữ liệu: lớp này sử dụng biểu diễn của các khung giọng nói từ mạng nơ-ron để tổng hợp ra một vec-tơ duy nhất đại diện cho đoạn tiếng nói đầu vào.
- Một hàm mất mát để tối ưu toàn bộ mô hình.

Trong pha kiểm tra, mạng nơ-ron và lớp tổng hợp dữ liệu được sử dụng để tạo ra vec-tơ biểu diễn của đoạn tiếng nói kiểm tra. Hàm mất mát chỉ được sử dụng trong pha phát triển, và bị loại bỏ trong pha kiểm tra và pha ghi danh.

Phần lớn các nghiên cứu học sâu cho xác minh người nói hướng tới cải thiện hệ thống qua nâng cao hiệu quả của lớp tổng hợp dữ liệu và hàm mất mát. Về



Hình 1.3: Kiến trúc mô hình xác minh người nói với học sâu [1].

mạng nơ-ron trích xuất đặc trưng, các nghiên cứu chủ yếu sử dụng các mạng xương sống thành công trong các bài toán khác như phân loại hình ảnh (Ví dụ: mạng VGG [13] và mạng ResNet [14]) và nhận dạng tiếng nói (mạng TDNN [15] và mạng LSTM [16]).

Sự thành công và hiệu quả của học sâu cho bài toán xác minh người nói trong tiếng Anh đến từ hai yếu tố. Thứ nhất, các kiến trúc mô hình, lớp tổng hợp dữ liệu và hàm mất mát được nghiên cứu kỹ lưỡng để phù hợp hơn với dạng dữ liệu và bài toán từ đó cải thiện hiệu năng. Thứ hai, các bộ dữ liệu người nói cực lớn như VoxCeleb [17] và SITW [18] được phát triển và mở công khai cho cộng đồng nghiên cứu sử dụng. Các nghiên cứu gần nhất về học sâu cho xác minh người nói không phụ thuộc văn bản đạt kết quả rất tốt xấp xỉ 1% EER trên tập VoxCeleb1 [19, 20].

1.3 Xác minh người nói trong tiếng Việt và vấn đề đặt ra

Ở Việt Nam trong những năm vừa qua, nhờ vào sự phát triển của ngành công nghệ thông tin cũng như sự phát triển mạnh mẽ của nền kinh tế đã tạo điều kiện nghiên cứu, phát triển và triển khai các ứng dụng công nghệ mang lại nhiều lợi ích cho xã hội. Các ứng dụng trí tuệ nhân tạo như nhận dạng khuôn mặt, xác minh người nói cũng không nằm ngoài xu thế. Tuy nhiên, theo hiểu biết của tác giả, các nghiên cứu về xác minh người nói tiếng Việt hay dữ liệu công khai

cho bài toán còn rất hạn chế.

Năm 2010, luận án tiến sĩ của TS. Ngô Minh Dũng [3] nghiên cứu giải quyết bài toán xác minh người nói tiếng Việt phụ thuộc văn bản. Nghiên cứu xây dựng cơ sở dữ liệu với 150 người nói với 17 âm tiết khác nhau bao gồm 10 âm tiết số và 7 âm tiết khác để thử nghiệm. Luận án sử dụng mô hình Gaussian hỗn hợp nhằm mô tả phân phân bố tần số cộng hưởng của tuyến phát âm để mô tả người nói. Nghiên cứu [21, 22] bởi các trường Đại học Sư phạm Huế, Đại học Sư phạm Kỹ thuật Hưng Yên và Đại học Bách khoa Hà Nội cùng giải quyết bài toán xác minh người nói phụ thuộc văn bản sử dụng mô hình Gaussian hỗn hợp. Nghiên cứu đầu tiên áp dụng học sâu cho bài toán xác minh người nói tiếng Việt phụ thuộc văn bản đạt 3.87% EER được công bố tại hội nghị SoICT lần thứ 9 vào năm 2018 [23]. Hiện tại chưa có nghiên cứu hay hệ thống thương mại nào cho xác minh người nói không phụ thuộc văn bản trong tiếng Việt.

Các mô hình xác minh người nói không phụ thuộc văn bản tuy thuận tiện khi sử dụng nhưng để xây dựng mô hình chất lượng tốt cần một lượng dữ liệu khổng lồ. Bộ dữ liệu VoxCeleb thường được sử để xây dựng các mô hình tiếng Anh VoxCeleb có hơn 7,000 danh tính khác nhau, hơn 1,000,000 đoạn giọng nói với tổng độ dài hơn 2,000 giờ. Bộ dữ liệu công khai duy nhất trong tiếng Việt phục vụ cho bài toán không phụ thuộc văn bản được xây dựng cho cuộc thi ZaloAI challenge [24] bao gồm 400 danh tính, 10,000 đoạn tiếng nói với tổng độ dài 8.7 giờ, ít hơn rất nhiều so với bộ dữ liệu tiếng Anh. Với lượng dữ liệu nhỏ, việc xây dựng mô hình chất lượng tốt cho tiếng Việt trở nên rất khó khăn.

Như đã đề cập trong mục này và các mục trước, xác minh người nói có tính ứng dụng cao và nhận được nhiều sự quan tâm từ cộng đồng nghiên cứu. Đối với xác minh người nói không phụ thuộc văn bản trong tiếng Việt, bộ dữ liệu phục vụ cho bài toán còn rất nhỏ và chưa có nghiên cứu về học sâu cho bài toán trong tiếng Việt. Do đó, mục tiêu của đồ án là xây dựng bộ dữ liệu và đề xuất mô hình cho bài toán TISV trong tiếng Việt.

1.4 Định hướng giải pháp

Để giải quyết việc thiếu hụt dữ liệu phục vụ cho huấn luyện mô hình, tác tác giả bổ sung dữ liệu danh tính bằng các bộ dữ liệu nhận dạng tiếng nói là VIVOS, VLSP và CommonVoice. Ngoài ra nhận thấy bộ dữ liệu còn nhiều lỗi, tác giả cũng xây dựng một bộ làm sạch dữ liệu dựa trên biểu diễn người nói từ mô hình

[3] http://luanan.nlv.gov.vn/luanan?a=d&d=TTcFabDIJkxi2010.1.5#

giúp loại bỏ các đoạn tiếng nói không hợp lệ, loại bỏ hay hợp nhất danh tính dễ dàng hơn từ đó tăng chất lượng bộ dữ liệu.

Trong khuôn khổ đồ án, tác giả tập trung vào thử nghiệm các kĩ thuật huấn luyện mô hình học sâu cho xác minh người nói không phụ thuộc văn bản trong tiếng Việt.

1.5 Bố cục đồ án

Trong chương 1, tác giả đã giới thiệu tổng quan về bài toán xác minh người nói, thảo luận về tình hình phát triển xác minh người nói trong tiếng Việt. Phần còn lại của đồ án được tổ chức như sau.

Chương 2 trình bày về cơ sở lý thuyết về học máy và học sâu, và phương pháp trích xuất đặc trưng âm học từ tín hiệu giọng nói.

Chương 3 trình bày về mô hình cơ sở sử dụng trong đồ án, các giải pháp đề xuất để huấn luyện mô hình một cách hiệu quả cùng với các nghiên cứu liên quan.

Trong chương 4, đồ án mô tả chi tiết phương pháp xây dựng bộ dữ liệu, phương pháp thực nghiệm và đánh giá kết quả thu được.

Chương 5 tổng kết các kết quả đạt được và định hướng phát triển trong tương lai.

Chương 2

Cơ sở lý thuyết

Chương 2 trình bày cơ sở lý thuyết về học máy, học sâu, và phương pháp trích xuất đặc trưng âm học từ tín hiệu giọng nói. Đây là nền tảng để xây dựng mô hình trong chương 3.

2.1 Học máy

2.1.1 Tổng quan về học máy

Học máy (Machine learning - ML) là một nhánh con của trí tuệ nhân tạo. Nghiên cứu học máy tập trung phát triển và xây dựng các thuật toán và kỹ thuật để giúp chương trình máy tính có thể "học" thực thi một tác vụ nào đó từ kinh nghiệm. Trong [25], nhà tiên phong về học máy Tom M. Mitchell định nghĩa như sau: Một chương trình máy tính được nói là học từ kinh nghiệm E để thực thi một tác vụ T nếu khả năng của chương trình được đo bằng độ đo P tiến bộ với kinh nghiệm E. Rõ ràng hơn, học máy là những chương trình máy tính có thể tự học được dữ liệu mà không cần được lập trình một cách cụ thể.

Với các cách định nghĩa T, P, và E khác nhau, các thuật toán học máy có thể được phân vào các nhóm: học có giám sát, học không giám sát, học bán giám sát, và học tăng cường.

Học có giám sát là lớp các thuật toán sử dụng dữ liệu được gán nhãn từ trước để tìm mối liên hệ giữa đầu vào và đầu ra. Mục tiêu của học có giám sát là tạo ra mô hình có thể dự đoán được đầu ra cho các đầu vào mới mà mô hình chưa gặp bao giờ. Tập dữ liệu gán nhãn trên được gọi là dữ liệu huấn luyện. Cho tập dữ liệu huấn luyện gồm N ví dụ $D = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N)\}$ trong đó $\boldsymbol{x}_i$ là vec-tơ đặc trưng của ví dụ thứ i và y_i là nhãn tương ứng. Thuật toán học có

giám sát tìm hàm ánh xạ $f : X \longrightarrow Y$ trong đó X là không gian vec-tơ đầu vào và Y là tập các nhãn.

Các thuật toán học có giám sát thường được sử dụng để giải quyết hai loại bài toán: phân loại (classification) và hồi quy (regression). Bài toán phân loại có nhãn rời rạc thuộc một tập cho sẵn. Ví dụ: phân loại đồ vật, phân loại phương tiện giao thông, phân loại giới tính dựa trên giọng nói. Khác với bài toán phân loại, bài toán hồi quy lại có nhãn nằm trên một miền liên tục. Ví dụ: dự đoán giá nhà đất, dự đoán thị trường chứng khoán.

Trong **học không giám sát**, khác với học có giám sát, ta chỉ có dữ liệu đầu vào mà không có dữ liệu đầu ra. Mục tiêu chính của các thuật toán là tìm ra cấu trúc ẩn trong dữ liệu hoặc trích xuất đặc trưng chung của tập dữ liệu.

Học bán giám sát nằm giữa học có giám sát và không có giám sát. Học bán giám sát thường được sử dụng khi có một lượng lớn dữ liệu không có nhãn và một số ít dữ liệu có nhãn. Phương pháp học bán giám sát được sử dụng rộng rãi nhất - tự huấn luyện (self-training) sử dụng mô hình huấn luyện trên tập có nhãn để sinh nhãn giả từ tập không nhãn nhằm tăng cường khả năng học của mô hình.

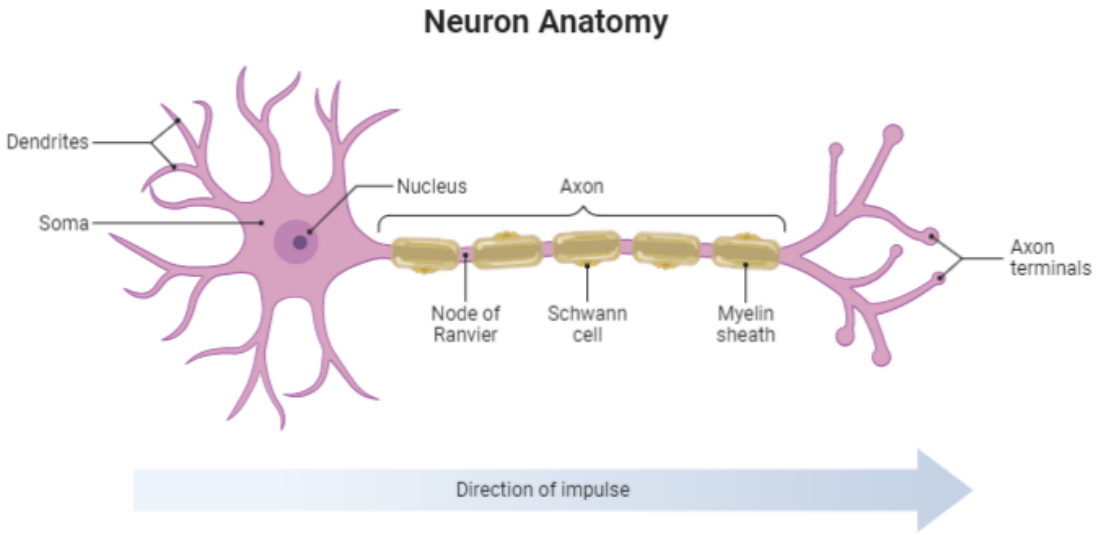
Hệ thống **học củng cố** được xem như là một tác tử trong một môi trường vô định và phức tạp. Với mỗi hành động, tác tử này nhận điểm thưởng hoặc phạt từ môi trường. Mục tiêu của học củng cố là các tác tử thông minh để tối đa điểm thưởng và thực hiện tác vụ của nó. Học củng cố hiện đang là lĩnh vực nghiên cứu tập trung nhiều nguồn lực và công sức với nhiều ứng dụng thực tế liên quan tới điều khiển robot hay vận hành nhà máy một cách tự động.

2.1.2 Mạng nơ-ron nhân tạo

Mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN) là một mô hình tính toán mô phỏng cách nơ-ron trong não người phân tích và xử lý thông tin. Học sâu với nền tảng là ANN cho phép xử lý dữ liệu và thông tin với độ chính xác vượt xa các mô hình xác suất cổ điển và tốc độ vượt trội so với con người.

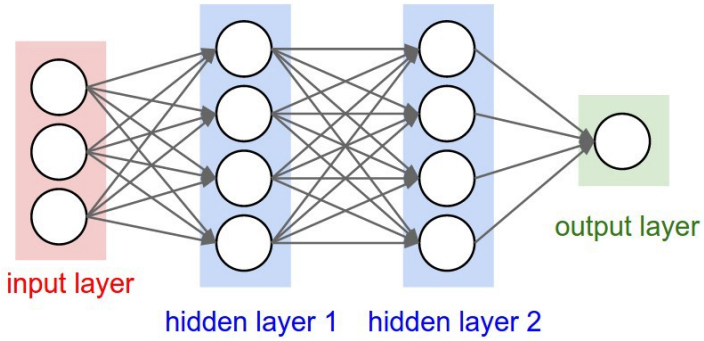
Nơ-ron sinh học gồm 3 phần chính: dendrite, thân tế bào (soma) và axon (Hình 2.2). Đầu tiên, tín hiệu đầu vào được thu thập bởi các khớp thần kinh của dendrite. Vai trò của soma là xử lý đầu vào và tổng hợp thông tin dựa vào độ quan trọng của tín hiệu. Sau đó, axon sẽ truyền thông tin đã được xử lý tới đầu ra. Thành phần cấu thành nên ANN là perceptron cũng có cách thức hoạt động

tương tự. Perceptron bao gồm lớp đầu vào, tập trọng số cùng hàm kích hoạt để tổng hợp thông tin và lớp đầu ra, tương tự như dendrite, soma và axon.



Hình 2.1: Cấu tạo nơ-ron sinh học [1].

Mạng nơ-ron là mô hình gồm nhiều lớp chồng lên nhau, mỗi lớp bao gồm nhiều perceptron (hay nơ-ron) tổng hợp thông tin từ lớp trước, mỗi lớp có một tập các nơ-ron độc lập. Lớp đầu tiên trong mạng được gọi là lớp đầu vào (input layer), lớp cuối cùng được gọi là lớp đầu ra (output layer), các lớp ở giữa được gọi là lớp ẩn (hidden layer) (Hình 2.2).



Hình 2.2: Cấu trúc của mạng nơ-ron nhân tạo[2].

Cho một mạng nơ-ron nhân tạo gồm L lớp, với $\boldsymbol{a}^i$ là tập các nơ-ron trong lớp thứ i, hai lớp nơ-ron liên tiếp có chỉ số i và $i+1$ được kết nối với nhau bằng một ma trận trọng số $\boldsymbol{W}^i$ và một vec-tơ bias $\boldsymbol{b}^i$. Ta có thể mô tả quá trình tính toán các nơ-ron từ lớp đầu vào và đầu ra dưới dạng toán học như trong Công

[1] https://app.biorender.com/biorender-templates/t-5f5b7e6139954000b2bde860-neuron-anatomy
[2] https://towardsdatascience.com/vanilla-neural-networks-in-r-43b028f415

thức 2.1.

$$\boldsymbol{a}^{(i)} = \sigma(\boldsymbol{W}^i \boldsymbol{a}^{(i-1)} + \boldsymbol{b}^i),\ 0 \leq i \leq L \tag{2.1}$$

Trong đó, σ được gọi là hàm kích hoạt (activation function). Hàm kích hoạt là một thành phần quan trọng không thể thiếu trong mạng nơ-ron nhân tạo. Để mạng nơ-ron nhân tạo có thể xấp xỉ hay học được những biến đổi phức tạp, σ phải là một hàm phi tuyến. Nếu không có hàm kích hoạt hay hàm kích hoạt là tuyến tính thì mạng nơ-ron dù có nhiều lớp đến đâu cũng có thể quy về một hàm tuyến tính và không có khả năng học được nhiều thông tin từ dữ liệu.

Có rất nhiều hàm kích hoạt khác nhau, ví dụ như hàm ReLU, hàm sigmoid, hàm Tanh, hàm softplus,... Trong các mạng nơ-ron hiện đại, hàm ReLU [26] (Công thức 2.2) được sử dụng phổ biến nhất do có nhiều lợi ích cho quá trình huấn luyện và tốc độ tính toán nhanh.

$$ReLU(x) = \max(0, x) \tag{2.2}$$

Quá trình lần lượt tính toán mạng nơ-ron từ lớp đầu vào, tới lớp ẩn và cuối cùng là lớp đầu ra được gọi là quá trình lan truyền tiến (feedforward).

Mạng nơ-ron học từ quá trình đối nghịch với lan truyền tiến gọi là lan truyền ngược (backpropagation). Với vec-tơ đặc trưng đầu vào $\boldsymbol{x}$ với nhãn tương ứng y, đầu tiên ta tính giá trị của nơ-ron lớp đầu ra $\boldsymbol{a}^{L-1}$ bằng quá trình lan truyền tiến. Sau đó, ta "lan truyền" sai khác giữa $\boldsymbol{a}^{L-1}$ và y tới toàn mạng để điều chỉnh các bộ trọng số $\boldsymbol{W}^i$ và $\boldsymbol{b}^i$ với mục tiêu để giảm sai khác này.

Để đo đạc sự sai khác trong đầu ra của mạng và nhãn đầu vào, ta sử dụng một hàm mất mát. Dựa vào tác vụ khác nhau của bài toán ta cần sử dụng các hàm mất mát khác nhau. Thông thường, bài toán hồi quy sử dụng hàm trung bình bình phương sai số (Mean Square Error - MSE), còn bài toán phân loại sử dụng hàm entropy chéo (Cross Entropy - CE). Với tập dữ liệu $D = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N)\}$, giả sử lớp cuối cùng của mạng có một nơ-ron, gọi a_i^{L-1} là giá trị của nơ-ron đầu ra tương ứng với dữ liệu đầu vào $\boldsymbol{x}_i$. Hàm mất mát MSE được tính trong Công thức 2.3.

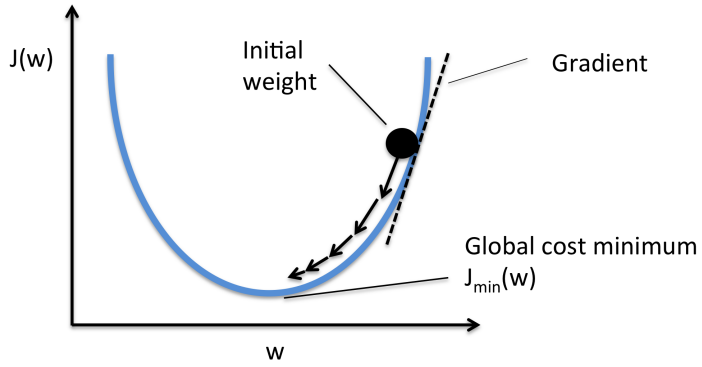
$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - a_i^{L-1})^2 \tag{2.3}$$

Hàm CE được tính theo Công thức 2.4.

$$CE = -\frac{1}{N}\sum_{i=1}^{N} a_i^{L-1} log(y_i) \tag{2.4}$$

Để điều chỉnh các trọng số, ta cần phải biết giá trị cập nhật cho mỗi trọng số. Dựa trên giá trị mất mát, các tín hiệu mất mát của nơ-ron trong lớp thứ i được tính toán dựa trên lỗi mà lớp thứ $i+1$ gây ra. Tín hiệu mất mát này được lan truyền từ lớp đầu ra về tới lớp đầu vào, sau đó thực hiện cập nhật các trọng số $\boldsymbol{W}, \boldsymbol{b}$ nhằm giảm các giá trị lỗi. Các giá trị lỗi của các bộ trọng số được gọi là gradient.

Sau khi có bộ gradient, ta có thể cập nhật mạng bằng cách đi ngược lại với hướng của gradient. Giá trị mất mát trên bộ dữ liệu sẽ giảm nếu gradient được cập nhật với bước đủ nhỏ. Bằng cách lặp đi lặp lại quá trình lan truyền tiến, lan truyền ngược và cập nhật trọng số bằng gradient, bộ trọng số có thể hội tụ tại một điểm cực tiểu của hàm mất mát. Phương pháp tối ưu lặp này được gọi là gradient descent (Hình 2.3).



Hình 2.3: Phương pháp gradient descent [3].

Hệ số học (learning rate) được điều chỉnh để kiểm soát bước cập nhật trong gradient descent. Với hệ số học lớn, bước đi của gradient descent trở nên lớn hơn và có thể gặp khó khăn hội tụ khi gần điểm cực tiểu. Hệ số học nhỏ hơn giúp mô hình dễ hội tụ hơn tuy nhiên mất nhiều vòng lặp hơn.

Phương pháp gradient descent truyền thống sử dụng gradient cho cả bộ dữ liệu. Tuy vậy, với các bộ dữ liệu cực lớn, điều này là không thể. Do vậy, ta phải xấp xỉ gradient của cả bộ dữ liệu bằng cách chia nhỏ bộ dữ liệu gốc thành các lô nhỏ (mini-batch) và thực hiện việc cập nhật trọng số trên các lô này. Phương pháp xấp xỉ này được gọi là stochastic gradient descent - SGD. Gần đây, nhiều

[3]https://machinelearningnotepad.wordpress.com/2018/04/15/gradient-descent

phương pháp biến thể của SGD được phát triển có thể kể đến như RMSprop, Adadelta [27] và Adam [28] giúp tăng tốc độ hội tụ cũng như tăng cường hiệu năng khi huấn luyện mô hình.

2.2 Học sâu

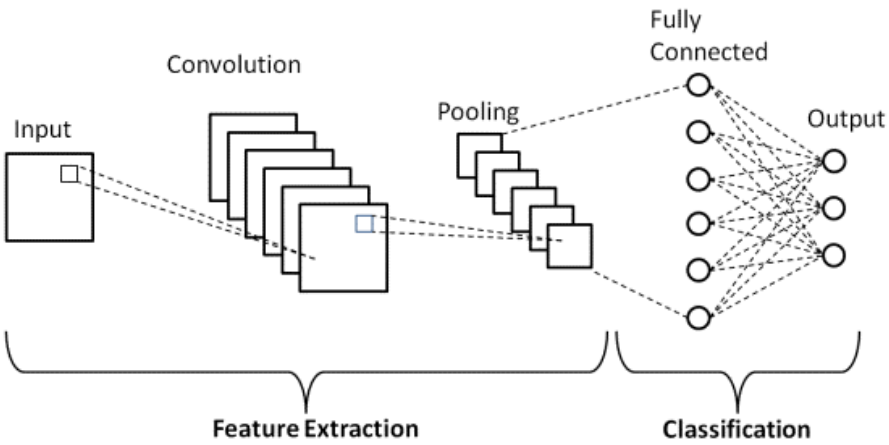
Học sâu là một lớp các mô hình học máy được xây dựng dựa trên mạng nơ-ron nhân tạo. Nếu mạng nơ-ron nhân tạo chỉ gồm vài lớp nơ-ron, các mạng học sâu được thiết kế để mở rộng ra hàng chục, trăm, thậm chí hàng nghìn lớp nơ-ron. Điều này giúp các mô hình học sâu giải quyết được nhiều bài toán phức tạp với độ chính xác ngang bằng con người. Huấn luyện mô hình học sâu yêu cầu một lượng dữ liệu và tài nguyên tính toán cực lớn. Trong thập kỉ vừa qua với sự bùng nổ của dữ liệu và tài nguyên tính toán, nghiên cứu học sâu trở thành tâm điểm của lĩnh vực trí tuệ nhân tạo.

2.2.1 Mạng nơ-ron tích chập

Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN), hay mạng tích chập, được đề xuất lần đầu vào năm 1989 bởi Yann LeCun [29] để giải quyết bài toán nhận dạng chữ viết tay. Mạng tích chập được thiết kế với mục tiêu xử lý dữ liệu dạng bảng - lưới, ví dụ như dữ liệu chuỗi thời gian có thể biểu diễn dưới dạng bảng 1D hay ảnh là dữ liệu 2D của các điểm ảnh. Năm 2012, Alex Krizhevsky xây dựng một mạng tích chập (AlexNet [30]) và tăng tốc quá trình huấn luyện sử dụng GPU. Mô hình đề xuất của Krizhevsky đứng đầu trong bảng xếp hạng trong cuộc thi phân loại ảnh ImageNet [31] với độ chính xác vượt hơn 10% các đội tham gia. Hiện nay, mạng tích chập được áp dụng xử lý nhiều bài toán phức tạp trong xử lý ảnh, xử lý ngôn ngữ tự nhiên, xử lý tín hiệu âm thanh, ...

Thời điểm hiện tại đã có rất nhiều kiến trúc mạng nơ-ron tích chập khác nhau được xây dựng, tuy nhiên chúng đều được cấu thành từ các lớp thành phần (Hình 2.4), bao gồm: lớp tích chập (convolutional layer), lớp tổng hợp (pooling layer) và lớp kết nối đầy đủ (fully connected layer).

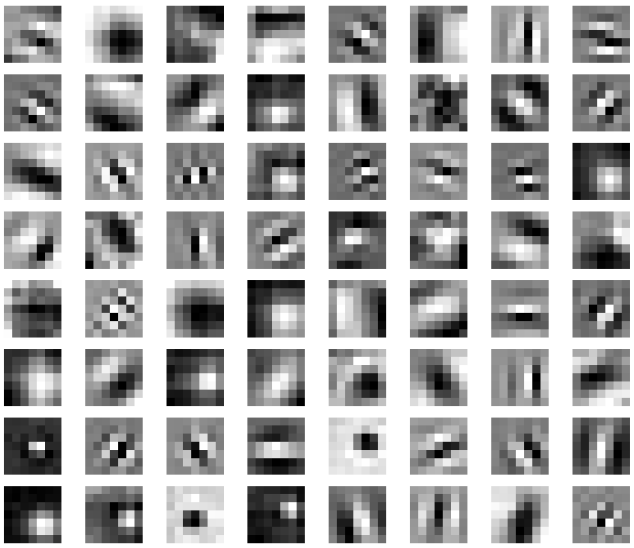
[4]https://www.upgrad.com/blog/basic-cnn-architecture



Hình 2.4: Các thành phần cơ bản của mạng tích chập [4].

Lớp tích chập

Lớp tích chập là lớp được sử dụng nhiều nhất trong mạng nơ-ron tích chập, mục tiêu của lớp tích chập là học các đặc trưng cục bộ từ dữ liệu đầu vào (Hình 2.5). Cái tên lớp tích chập xuất phát từ việc lớp sử dụng phép biến đổi toán học tuyến tính gọi là tích chập (convolution).



Hình 2.5: Các đặc trưng học được trong lớp tích chập [5].

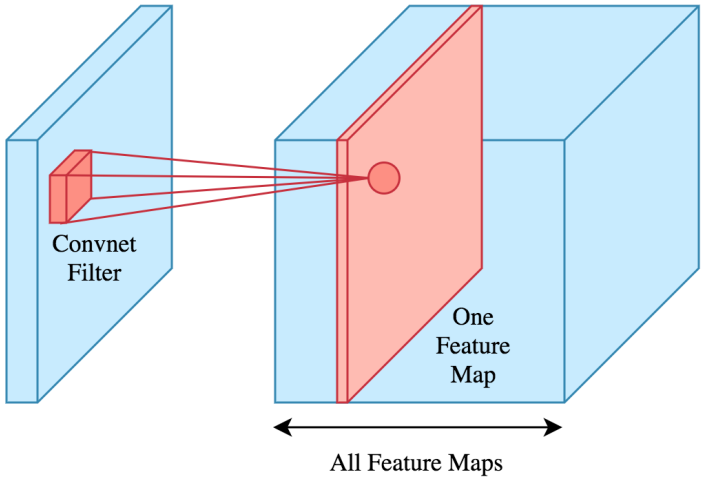
Ban đầu, phép tích chập được được sử dụng phổ biến trong xử lý tín hiệu. Về sau nguyên lý biến đổi trong tích chập mới được áp dụng trong lĩnh vực xử lý hình ảnh. Công thức tích chập cho 2 hàm x và w theo chiều thời gian t được

[5]https://debuggercafe.com/visualizing-filters-and-feature-maps-in-convolutional-neural-networks-using-pytorch

tính trong Công thức 2.5.

$$(x * w)(t) = \sum x(\alpha)w(t - \alpha) \tag{2.5}$$

trong đó $(x * w)$ là ký hiệu tích chập. Trong xử lý ảnh, phép tích chập được mở rộng ra 2D hoặc 3D. Trong lớp tích chập, x được gọi là đầu vào của lớp còn w được gọi là bộ lọc (filter) (Hình 2.6). Phép toán tích chập dựa trên nơ-ron của lớp trước để tính toán ra giá trị nơ-ron của lớp ngay sau.



Hình 2.6: Một ví dụ của lớp tích chập [6].

Bộ lọc là một bộ trọng số học được có kích thước nhỏ (ví dụ 5x5x3, dài rộng 5, chiều sâu 3 của khối đầu vào phía trước). Trong quá trình lan truyền tiến, bộ lọc được dịch chuyển trên cả chiều dài và chiều rộng của khối đầu vào để tổng hợp thông tin thành một khối đặc trưng 2D. Cùng với kích thước bộ lọc, lớp tích chập còn có 2 tham số bước nhảy (stride) và đệm (padding). Tham số bước nhảy quy định độ dịch chuyển của bộ lọc, còn tham số đệm là số lượng giá trị 0 được thêm vào xung quanh khối đầu vào để kiểm soát kích thước đầu ra của lớp tích chập.

Có nhiều bộ lọc trong một lớp tích chập để học nhiều dạng thông tin đặc trưng khác nhau như cạnh thẳng, cạnh chéo, đốm màu hay các đặc trưng phức tạp hơn ở các lớp sâu hơn. Các khối đặc trưng 2D của nhiều bộ lọc được xếp chồng lên nhau thành một khối đặc trưng 3D (Hình 2.6).

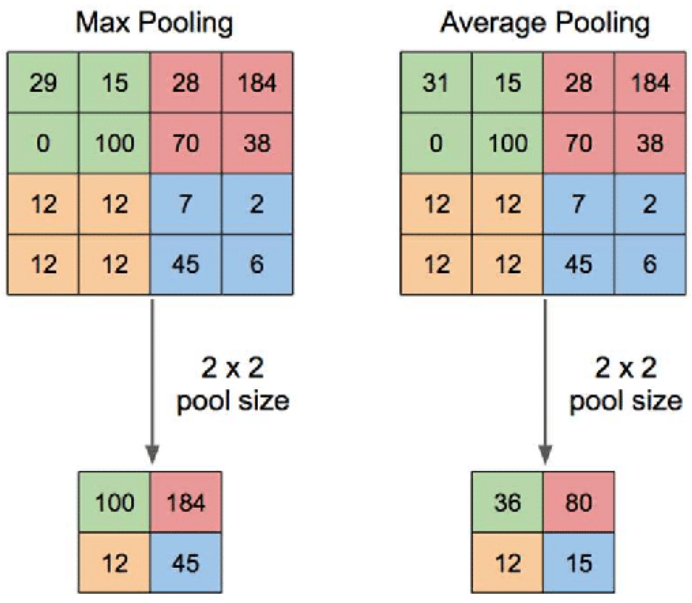
Điểm khác biệt chính giữa lớp tích chập và lớp ẩn thông thường là tính cục bộ. Mỗi nơ-ron trong lớp tích chập chỉ được kết nối tới một số điểm nhất định trong lớp phía trước trong khi một nơ-ron trong lớp ẩn thông thường được kết nối tới tất cả nơ-ron ở lớp trước.

[6] https://towardsdatascience.com/convolution-neural-networks-a-beginners-guide-implementing-a-mnist-hand-written-digit-8aa60330d022

Mạng tích chập giả định đặc trưng học được ở một vùng ảnh cũng có thể phát hiện đặc trưng tương tự tại một vùng khác trong ảnh. Trong quá trình tính toán tích chập, bộ trọng số của bộ lọc được sử dụng đi sử dụng lại tại các vùng khác nhau của ảnh. Điều này giúp cho tổng số trọng số của lớp tích chập ít hơn nhiều so với một lớp ẩn trong mạng nơ-ron thông thường mà vẫn học được nhiều đặc trưng ý nghĩa. Trong quá trình lan truyền ngược, gradient của bộ lọc được tổng hợp từ nhiều vùng khác nhau.

Lớp tổng hợp

Lớp tổng hợp (pooling layer) có chức năng tổng hợp thông tin từ lớp trước, giảm số nơ-ron và trọng số trong mạng từ đó giảm chi phí tính toán. Ngoài ra lớp tổng hợp còn giúp biểu diễn qua các lớp nhất quán hơn với sự thay đổi nhỏ trong đầu vào từ đó giúp mô hình hiệu quả hơn với dữ liệu mới. Cách thức hoạt động của lớp tổng hợp tương đối đơn giản, nó dùng một cửa sổ (thường có kích thước 2x2) để trượt trên đầu vào. Với mỗi điểm trượt, lớp tổng hợp chọn giá trị lớn nhất trong vùng (tổng hợp cực đại - max pooling) hoặc lấy trung bình của các giá trị (tổng hợp trung bình - average pooling) làm giá trị cho ma trận đầu ra (Hình 2.7).



Hình 2.7: Một ví dụ của tổng hợp cực đại và tổng hợp trung bình [2].

Lớp kết nối đầy đủ

Trong mạng nơ-ron nhân tạo thông thường, mọi lớp trừ lớp đầu vào là lớp kết nối đầy đủ, nghĩa là nơ-ron ở lớp sau được tính toán dựa trên tất cả nơ-ron của

lớp trước. Thông thường trong mạng nơ-ron tích chập, lớp kết nối đầy đủ nằm ở cuối mạng, đóng vai trò là bộ phân loại của mạng. Đầu ra của lớp tích chập hay lớp tổng hợp thường là ma trận 2D hoặc 3D, do vậy cần phải được duỗi thẳng thành một vec-tơ để làm đầu vào cho lớp kết nối đầy đủ.

2.2.2 Hàm mất mát

Như đã trình bày trong phần 2.1.2, trong học máy, hàm mất mát đo đạc sự sai khác của đầu ra trong nhãn đầu vào. Điều tương tự cũng áp dụng cho học sâu. Trong học sâu, hàm mất mát có thể chia thành hai nhóm: hàm mất mát phân loại (classification loss) và hàm mất mát phân biệt (discriminative loss hay metric loss). Hai hàm mất mát đơn giản và được sử dụng rộng rãi nhất cho 2 nhóm là hàm softmax và hàm triplet.

Hàm mất mát softmax

Về cơ bản, hàm mất mát softmax là một hàm mất mát phân loại đa lớp gồm hàm kích hoạt softmax kết hợp hàm entropy chéo như mô tả trong Công thức 2.4. Thông thường trong một mạng học sâu phân loại, lớp đầu ra có số nơ-ron tương ứng với số lớp cần được phân loại (ví dụ mô hình phân loại 6 loại xe có 6 nơ-ron ở lớp đầu ra). Giá trị đại diện cho mỗi lớp ở tầng đầu ra là cái nhận được khi sử dụng mạng dự đoán cho một ví dụ. Hàm softmax đưa vec-tơ đầu ra về khoảng $(0, 1)$ và tổng của chúng đúng bằng 1. Bởi vì giá trị của một nơ-ron đại diện cho một lớp, có thể xem đó là xác suất dự đoán của lớp đó. Hàm mất mát softmax cho một ví dụ có vec-tơ biểu diễn $\boldsymbol{z}$ thuộc lớp c được tính bằng Công thức 2.6.

$$\mathcal{L}_{\text{softmax}}(\boldsymbol{z}, c) = -log(\frac{e^{\boldsymbol{z}_c}}{\sum_j e^{\boldsymbol{z}_j}}) \tag{2.6}$$

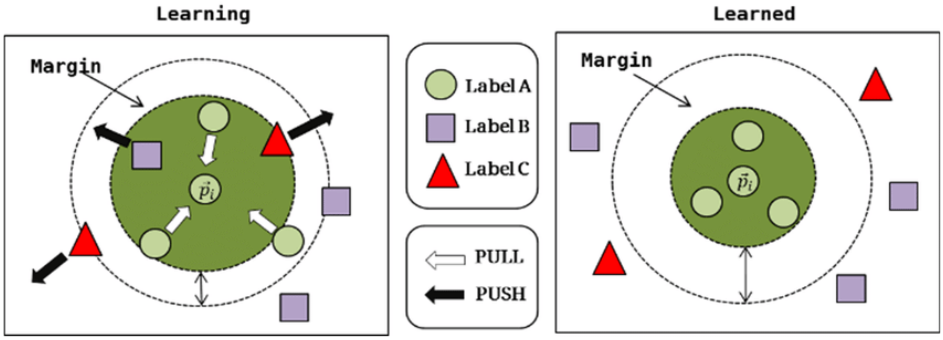
Hàm mất mát triplet

Khác với hàm mất mát phân loại, mục tiêu của hàm mất mát phân biệt là học điểm khác biệt giữa những vật thể hay ví dụ khác nhau. Các hàm phân biệt như triplet giúp mạng học không gian biểu diễn mà trong đó các ví dụ giống nhau sẽ gần nhau và những ví dụ khác nhau nằm xa nhau.

Đầu vào của hàm triplet bao gồm 3 điểm trong không gian biểu diễn. Trong

mỗi ba điểm, có một điểm được gọi là neo (anchor) kí hiệu A, một điểm dương (positive) kí hiệu P có cùng nhãn với A, và một điểm âm (negative) kí hiệu N khác nhãn với A. Mục tiêu của hàm triplet là kéo điểm P gần hơn tới A trong không gian biểu diễn và đẩy P ra khỏi A sao cho khoảng cách A - P nhỏ hơn khoảng cách A - N một khoảng m gọi là biên (margin) (Hình 2.8). Với m là hệ số biên, $\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_k$ lần vượt là vec-tơ A, P, N, giá trị hàm triplet được tính như trong Công thức 2.7.

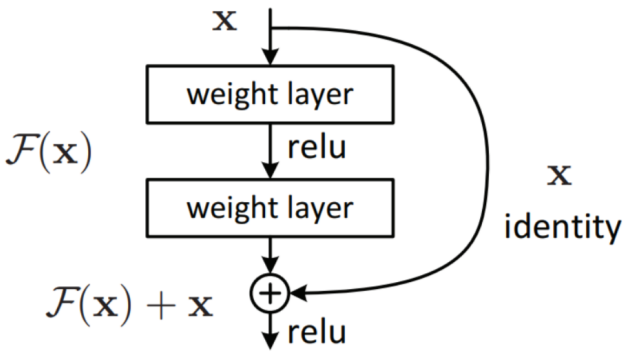
$$\mathcal{L}_{\text{triplet}}\ (\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_k) = \max\left(0, \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2 - \|\boldsymbol{z}_i - \boldsymbol{z}_k\|_2^2 + m\right) \tag{2.7}$$



Hình 2.8: Ví dụ hàm mất mát triplet [3].

2.2.3 Mạng nơ-ron kết nối tắt

Các mạng học sâu khi mở rộng càng nhiều lớp càng có khả năng xấp xỉ các hàm phức tạp và giải quyết bài toán phức tạp hơn. Tuy nhiên, mạng rất sâu lại gặp phải vấn đề vanishing/exploding gradients, hiện tượng mà gradient tiến tới 0 hoặc vô hạn trong quá trình lan truyền ngược trong các lớp đầu khiến cho mô hình không được cải thiện.



Hình 2.9: Skip connection trong ResNet [4].

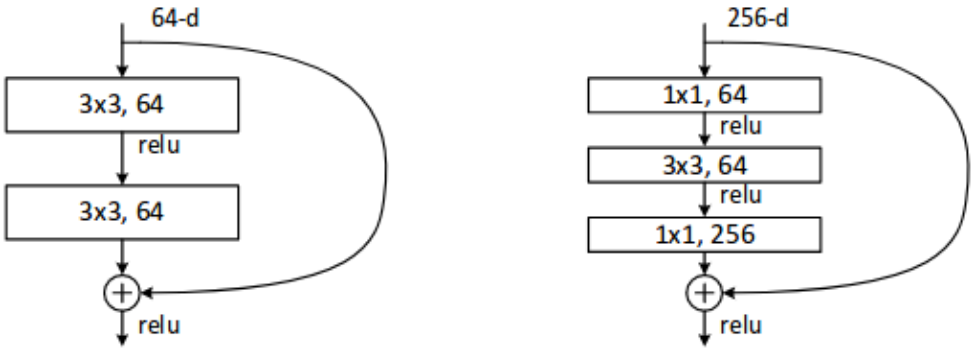
Mạng nơ-ron kết nối tắt (Residual Neural Network - ResNet) [4] được ra đời để

giải quyết vấn đề vanishing/exploding gradients. Để luồng thông tin tới được các lớp đầu của mạng trong quá trình lan truyền ngược, ResNet sử dụng một cấu trúc đặc biệt gọi là kết nối tắt (skip connection) (Hình 2.9). Kết nối tắt khác với kết nối thông thường ở chỗ nó không kết nối 2 lớp liên tiếp mà kết nối 2 lớp không liền kề nhau. Khi lan truyền ngược, các kết nối này đưa gradient trực tiếp về các lớp phía đầu của mạng và tránh khỏi hiện tượng vanishing/exploding gradients.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Hình 2.10: Các kiến trúc khác nhau của ResNet [4].

ResNet có nhiều kiến trúc với nhiều độ sâu khác nhau, hình 2.10 mô tả các kiến trúc này. Với ResNet-18 và ResNet-34, các khối residual được xây dựng từ các lớp tích chập với bộ lọc kích thước 3x3. Các mô hình nhiều lớp hơn như ResNet-50, ResNet-101, ResNet-152 sử dụng khối nút thắt cổ chai (bottleneck) để giảm chi chí tính toán. Hình 2.11 mô tả kiến trúc hai khối này.



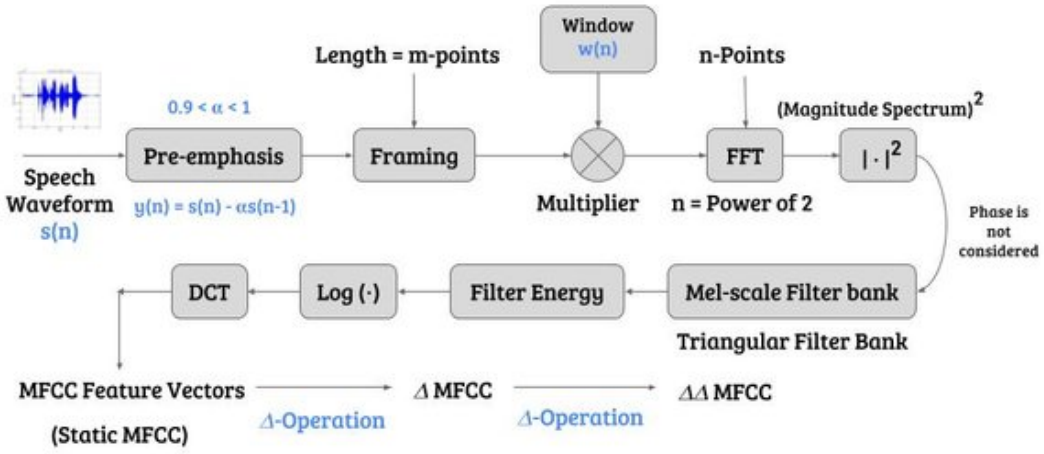
Hình 2.11: Kiến trúc hai khối residual trong các kiến trúc mạng ResNet [4].

2.3 Trích xuất đặc trưng âm học

Việc trích xuất đặt trưng được thực hiện bằng cách thay đổi tín hiệu giọng nói thành dạng biểu diễn tham số với tốc độ dữ liệu (data rate) tương đối thấp để dễ dàng xử lý và phân tích ở các bước sau. Trích xuất đặc trưng biến đổi tín hiệu ban đầu thành dạng ngắn gọn nhưng logic, có tính phân biệt cao và đáng tin cậy hơn tín hiệu thực. Do trích xuất đặc trưng là thành phần đầu tiên trong hệ thống, chất lượng các phần tiếp theo bị ảnh hưởng đáng kể bởi trích xuất đặc trưng.

Trích xuất đặc trưng đóng vai trò quan trọng trong bất kì hệ thống tiếng nói nào, từ nhận dạng tiếng nói tới xác minh người nói. Hai trong những đặc trưng âm thanh phổ biến nhất được sử dụng là filter banks và Mel-Frequency Cepstral Coefficients (MFCCs).

MFCCs ban đầu được đề xuất để xác định các từ đơn âm trong các câu nói liên tục nhưng không dùng để nhận dạng người nói, xác minh người nói. Cách tính MFCCs mô phỏng hệ thống thính giác của con người nhằm tái lập nguyên lý hoạt động của tai với giả định rằng tai người là thiết bị nhận dạng người nói đáng tin cậy. MFCCs giống tai người với các bộ lọc tần số tuyến tính ở tần số thấp và logarit ở tần số cao đã được sử dụng để giữ lại các đặc tính quan trọng về mặt ngữ âm của tín hiệu giọng nói.



Hình 2.12: Thuật toán trich xuất MFCCs [5].

Trích xuất filter banks và MFCCs tuân theo quy trình khá tương tự nhau, trong đó cả hai trường hợp filter banks đều được tính toán và với một vài bước bổ sung

có thể thu được MFCCs. Một cách ngắn gọn, một tín hiệu đi qua một bộ lọc nhấn mạnh (pre-emphasis) đầu tiên; sau đó được cắt thành các khung (frame) (chồng lên nhau) và được biến đổi bằng một loại cửa sổ (Hanning, Hamming, ...); sau đó, mỗi khung được biến đổi bằng phép biến đổi Fourier (Fourier transform) để tính toán âm phổ (spectrum) và filter banks. Để có được MFCCs, phép biến đổi cô-sin rời rạc (Disrete Cosine Transform) được áp dụng lên filter banks giữ lại một vài hệ số trong khi phần còn lại bị loại bỏ. Bước cuối cùng trong cả 2 trường hợp là chuẩn hoá trung bình (Hình 2.12).

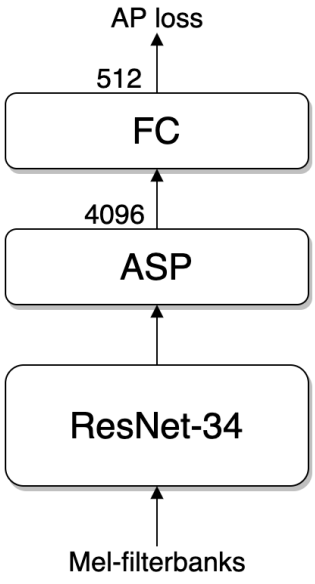
MFCCs tương đối chính xác cho các bài toán nhận diện mẫu liên quan tới giọng nói. Tuy nhiên, hiệu năng và tính khái quát hoá của MFCCs suy giảm trong trường hợp âm thanh trong thế giới thực chứa nhiều tạp âm. Ngoài ra, MFCCs còn bỏ đi thông tin cao độ của giọng nói - thông tin quan trọng để phân biệt giọng nói. Vậy nên, MFCCs không được ưa chuộng bằng filter banks trong các hệ thống nhận dạng tiếng nói hay xác minh người nói hiện đại.

Chương 3

Mô hình xác minh người nói tiếng Việt

Trong chương này, tác giả trình bày mô hình học sâu cơ sở giải quyết bài toán xác minh người nói, đề xuất phương pháp để cải tiến mô hình cho tiếng Việt và trình bày các nghiên cứu liên quan.

3.1 Mô hình cơ sở



Hình 3.1: Tổng quan mô hình cơ sở sử dụng trong đồ án.

Mô hình cơ sở [19] là mô hình tốt nhất cho xác minh người nói tại thời điểm hiện tại được sử dụng trong đồ án tuân theo hệ thống ba pha được mô tả như trong phần 1.2 (Hình 3.1). Mô hình nhận đầu vào là một mini-batch các đặc trưng âm học của nhiều câu nói khác nhau. Mạng ResNet-34 trích xuất biểu

diễn của các đặc trưng đầu vào; do mỗi câu nói có nhiều khung, mô hình sử dụng một lớp tổng hợp thống kê tập trung (Attentive Statistic Pooling - ASP) để giúp biểu diễn câu nói luôn có chiều cố định. Cuối cùng, giá trị mất mát của một mini-batch được tính toán bằng hàm mất mát nguyên mẫu góc (Angular Prototypical - AP). Toàn bộ mô hình được cập nhật cho mỗi mini-batch sử dụng lan truyền ngược và phương pháp tối ưu Adam.

Trong các mục tiếp theo, đồ án sẽ trình bày chi tiết các thành phần chính của mô hình.

3.1.1 Biểu diễn khung giọng nói bằng mạng ResNet

Mạng ResNet sử dụng trong đồ án là biến thể của mạng ResNet-34 như mô tả trong phần 2.2. Tại mỗi lớp, mạng sử dụng một nửa số bộ lọc so với mạng ResNet-34 gốc trong các khối ResNet và chứa tổng cộng 8.0 triệu trọng số. Trong lớp tích chập đầu tiên, tham số bước nhảy được chỉnh thành 1 so với 2 trong mạng ResNet-34 gốc khiến đầu vào của các lớp đằng sau lớn hơn từ đó tăng khối lượng tính toán. Chi tiết cấu trúc của mạng được mô tả trong Bảng 3.1.

Bảng 3.1: Kiến trúc mạng ResNet sử dụng trong đồ án

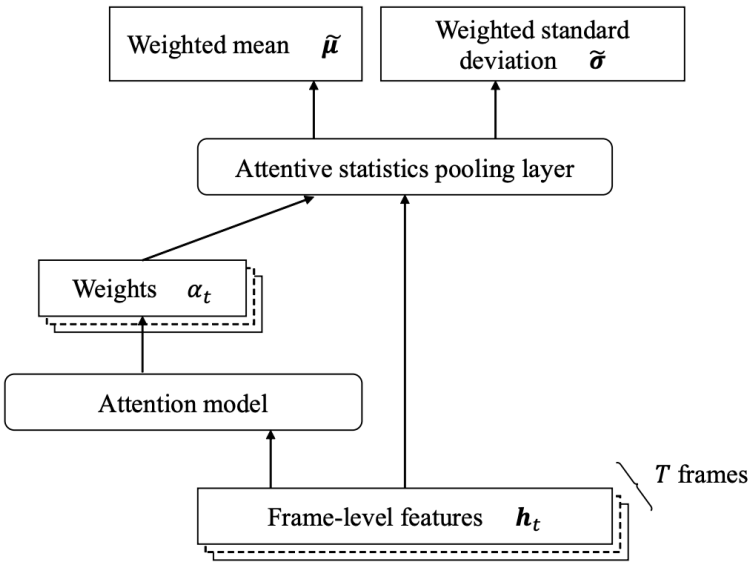
Layer	**Kernel size**	**Stride**	**Output shape**
Conv1	$3 \times 3 \times 32$	1×1	$L \times 64 \times 32$
ResBlock1	$3 \times 3 \times 32$	1×1	$L \times 64 \times 32$
ResBlock2	$3 \times 3 \times 64$	2×2	$L/2 \times 32 \times 64$
ResBlock3	$3 \times 3 \times 128$	2×2	$L/4 \times 16 \times 128$
ResBlock4	$3 \times 3 \times 256$	2×2	$L/8 \times 8 \times 256$
Flatten	-	-	$L/8 \times 2048$

3.1.2 Tổng hợp thống kê tập trung

Trong thực tế, không phải khung âm thanh nào cũng chứa nhiều thông tin của người nói do độ dài một khung rất ngắn thông thường chỉ 25 mili giây. Ví dụ, một khung có thể chứa nhiều tiếng ồn, hoặc không hề chứa giọng nói. Do vậy, hệ thống sử dụng tổng hợp thống kê tập trung để đánh trọng số cho biểu diễn của các khung với mong muốn tăng thông tin của những khung nhiều ý nghĩa và giảm thông tin của những khung ít ý nghĩa. Từ đó có thể phân biệt giọng nói một cách hiệu quả hơn.

ASP [6] nhận đầu vào là tập các vec-tơ biểu diễn khung $\boldsymbol{h}_t$ $(t = 1, ..., T)$. Vec-tơ biểu diễn của toàn đoạn âm thanh được tính qua 2 bước: đánh trọng số cho từng khung bằng cơ chế tập trung và tổng hợp thông tin thống kê dựa trên trọng số

tính được (Hình 3.2).



Hình 3.2: Tổng hợp thống kê tập trung [6].

Cơ chế tập trung

Bằng cơ chế tập trung, trọng số của từng khung có thể được tính theo hai Công thức 3.1 và 3.2.

$$e_t = \boldsymbol{v}^T f(\boldsymbol{W}\boldsymbol{h}_t + b) + k \tag{3.1}$$

$$\alpha_t = \frac{\alpha_t}{\sum_{\rho}^{T} \alpha_\rho} \tag{3.2}$$

Trong Công thức 3.1, với $\boldsymbol{v}, \boldsymbol{W}$ là các ma trận trọng số có thể học được, f là hàm kích hoạt phi tuyến như tanh hoặc ReLu, ta tính được điểm cho mỗi khung e_t. Sau đó, điểm của mỗi khung được chuẩn hoá trên tất cả các khung để thu được trọng số tập trung bằng hàm softmax như trong Công thức 3.2.

Tổng hợp thống kê

Sau khi có được trọng số của các khung, ta tính vec-tơ trung bình có trọng số với Công thức 3.3.

$$\boldsymbol{\mu} = \sum_{t}^{T} \alpha_t \, \boldsymbol{h}_t \tag{3.3}$$

Bằng cách tính này, vec-tơ biểu diễn của đoạn âm thanh tập trung hơn vào những khung tiếng nói có ý nghĩa cao. Ngoài ra, các trọng số tập trung còn được sử dụng để tính độ lệch chuẩn có trọng số theo Công thức 3.4.

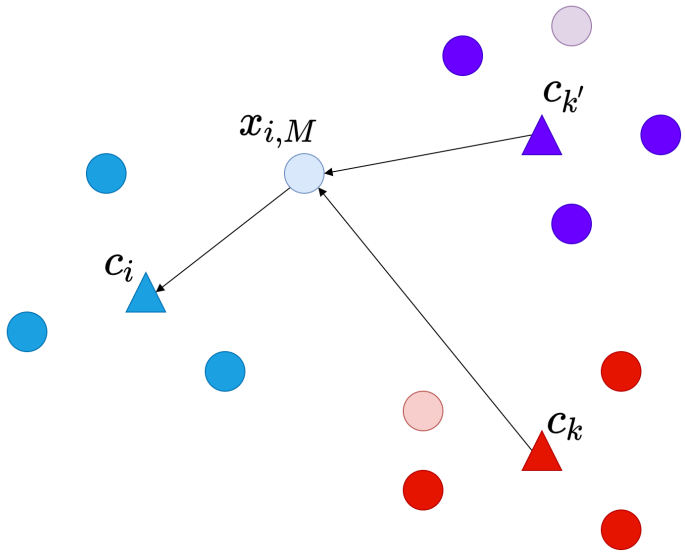
$$\boldsymbol{\sigma} = \sqrt{\sum_{t}^{T} \alpha_t \boldsymbol{h}_t \odot \boldsymbol{h}_t - \boldsymbol{\mu} \odot \boldsymbol{\mu}} \tag{3.4}$$

Với $\odot$ là phép nhân Hadamard, $\boldsymbol{\mu}$ là vec-tơ trung bình có trọng số tính trong Công thức 3.3. Sau khi hoàn tất quá trình tính toán vec-tơ trung bình và độ lệch chuẩn có trọng số, $\boldsymbol{\mu}$ và $\boldsymbol{\sigma}$ được ghép lại để biểu diễn cho một đoạn tiếng nói. Bằng cách này, mọi đoạn âm thanh dài ngắn đều có vec-tơ biểu diễn với số chiều như nhau, được tổng hợp từ những khung âm thanh có ý nghĩa nhất trong câu.

3.1.3 Hàm mất mát nguyên mẫu góc AP

Trong thực tế, ta cần tổng hợp từ một số câu nói nhất định để tạo vec-tơ biểu diễn người nói. Do vậy, Chung và cộng sự [32] đề xuất hàm mất mát AP tối ưu không gian biểu diễn dựa trên nguyên mẫu (prototype) của người nói. Mỗi người nói có một nguyên mẫu và một câu truy vấn, mục tiêu của AP là đẩy xa truy vấn của một người ra xa nguyên mẫu của những người khác và kéo nó lại gần nguyên mẫu của người đó (Hình 3.3).

Xét một mini-batch gồm M đoạn tiếng nói từ mỗi N người nói, gọi $\boldsymbol{x}_{i,j}$ là vec-tơ biểu diễn của đoạn tiếng nói thứ j của người thứ i, $1 \leq i \leq N, 1 \leq j \leq M$. Giả sử truy vấn của một người là câu cuối cùng của người đó $\boldsymbol{x}_{i,M}$, nguyên mẫu của



Hình 3.3: Hàm mất mát Angular Prototypical.

một người nói được tính toán như Công thức 3.5.

$$\boldsymbol{c}_i = \frac{1}{M-1}\sum_{m=1}^{M-1} \boldsymbol{x}_{i,m} \tag{3.5}$$

Trong AP, độ tương đồng cô-sin được sử dụng để làm độ đo. Độ tương đồng được tính theo Công thức 3.6 với hệ số scale w và bias b. Hai hệ số này giúp mô hình hội tụ ổn định hơn và khái quát hoá tốt hơn với thay đổi trong đặc trưng đầu vào [33].

$$\boldsymbol{S}_{i,k} = w \cdot cos(\boldsymbol{x}_{i,M}, \boldsymbol{c}_k) + b \tag{3.6}$$

Trong quá trình huấn luyện, câu truy vấn của mỗi người được phân loại dựa trên độ tương đồng đối với N nguyên mẫu trong mini-batch:

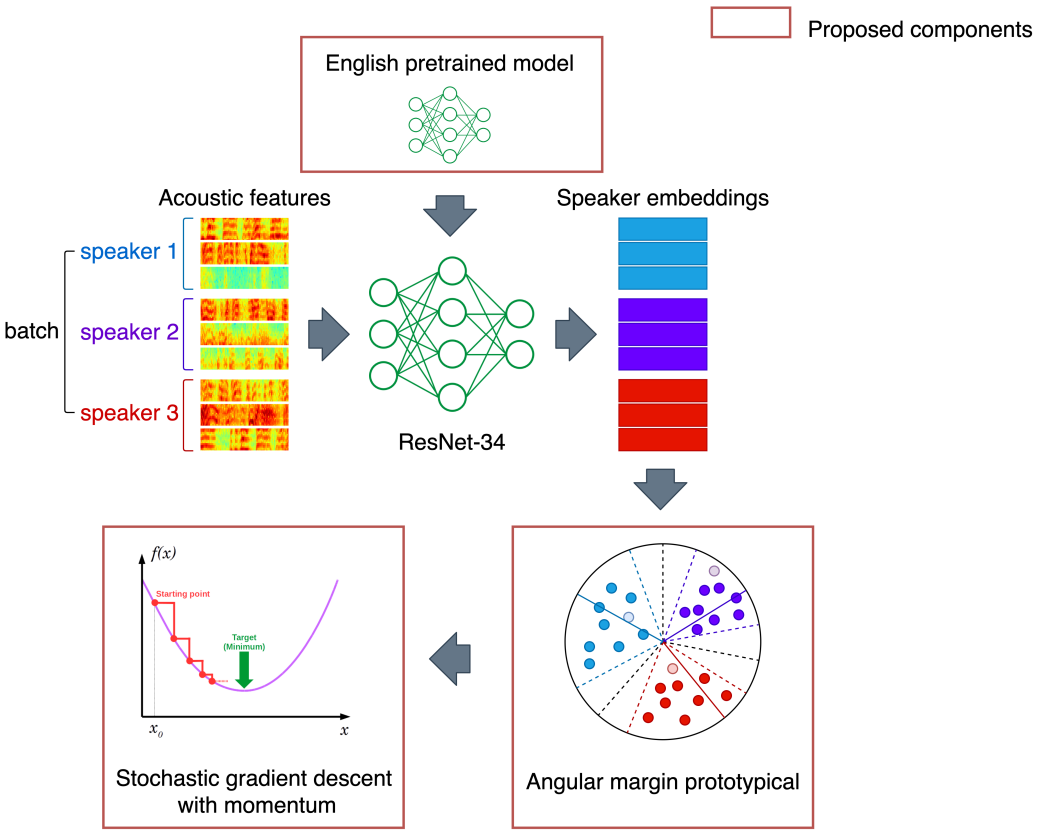
$$L_{AP} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{\boldsymbol{S}_{i,i}}}{\sum_{k=1}^{N} e^{\boldsymbol{S}_{i,k}}} \tag{3.7}$$

Trong Công thức 3.7, $\boldsymbol{S}_{i,i}$ là độ tương đồng của truy vấn người i và nguyên mẫu của chính người đó. Bằng việc sử dụng hàm softmax, $\boldsymbol{S}_{i,i}$ được đẩy gần hơn tới 1 và mô hình bị "phạt" nặng hơn nếu độ tương đồng của truy vấn người i tới nguyên mẫu của người khác lớn.

3.2 Đề xuất mô hình cho tiếng Việt

3.2.1 Mô hình tổng quan

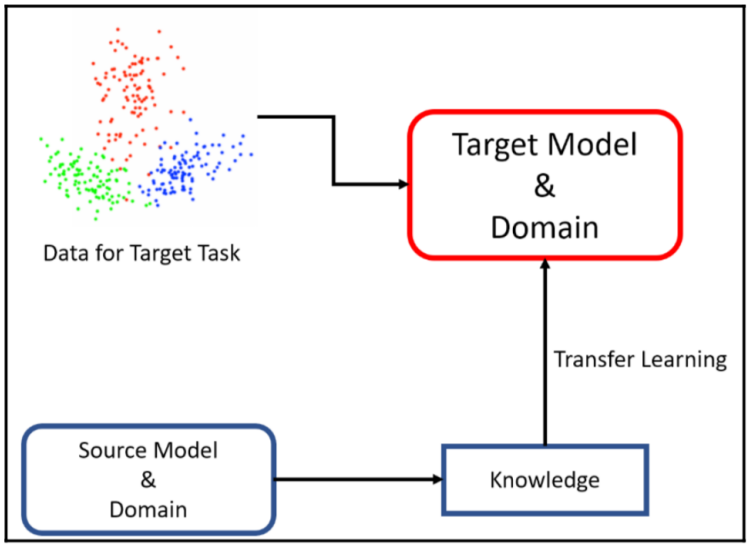
Như đã trình bày trong phần 1.3, dữ liệu tiếng Việt còn rất hạn chế, vì vậy đồ án đề xuất mô hình dựa trên học chuyển tiếp nhằm tận dụng kiến thức từ lượng dữ liệu khổng lồ có trong tiếng Anh. Hình 3.4 mô tả tổng quan mô hình đề xuất. Tương tự mô hình cơ sở, mô hình nhận đầu vào là một mini-batch đặc trưng âm học của các câu nói từ nhiều người khác nhau. Mạng ResNet-34 được khởi tạo với trọng số từ mô hình tiếng Anh huấn luyện trên VoxCeleb để học chuyển tiếp. ASP cũng được sử dụng bởi mô hình nhằm tổng hợp biểu diễn khung. Tiếp theo, giá trị mất mát được tính toán với hàm mất mát nguyên mẫu góc với hệ số phạt biên (Angular Margin Prototypical - AMP). Bộ trọng số của mô hình được cập nhật với hàm tối ưu SGD thay cho Adam. Trong các mục tiếp theo trong chương này, đồ án trình bày chi tiết về các thay đổi của mô hình đề xuất so với mô hình đề xuất: học chuyển tiếp, hàm mất mát và phương pháp tối ưu.



Hình 3.4: Tổng quan phương pháp đề xuất cho xác minh người nói tiếng Việt.

3.2.2 Học chuyển tiếp sử dụng kiến thức trên tiếng Anh

Học chuyển tiếp là một kỹ thuật trong học máy khi mà một mô hình được huấn luyện cho một tác vụ nhất định được sử dụng làm điểm bắt đầu cho một tác vụ khác. Học chuyển tiếp cho phép rút ngắn quá trình huấn luyện và gia tăng hiệu năng cho quá trình huấn luyện mô hình trên tác vụ mới với ít dữ liệu hơn đáng kể.

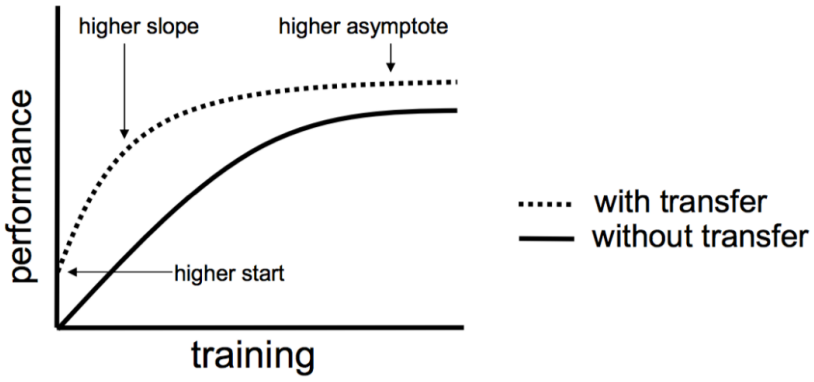


Hình 3.5: Sơ đồ mô tả học chuyển tiếp sử dụng kiến thức hiện có cho các tác vụ mới [7].

Cụ thể hơn, học chuyển tiếp hỗ trợ việc huấn luyện tác vụ mục tiêu theo những cách sau:

- Hiệu suất cơ sở tốt hơn (higher start): khi ta tăng cường kiến thức của mô hình mới với kiến thức từ mô hình gốc, hiệu suất cơ sở có thể cải thiện nhờ việc chuyển giao kiến thức.
- Thời gian huấn luyện ngắn hơn (higher slope): tốc độ hội tụ của mô hình mới có thể nhanh hơn dẫn tới thời gian huấn luyện ngắn hơn.
- Kết quả cuối cùng tốt hơn (higher asymptote): hiệu suất cuối cùng cao hơn có thể đạt được bằng việc sử dụng học chuyển tiếp.

Huấn luyện mô hình học sâu cần một lượng lớn tài nguyên tính toán và dữ liệu, do đó học chuyển tiếp được sử dụng rộng rãi trong cộng đồng nghiên cứu học sâu cho bài toán thị giác máy tính hay xử lý ngôn ngữ tự nhiên. Các thư viện học sâu nổi tiếng như Pytorch hay Tensorflow đều công bố các mô hình huấn luyện sẵn trên hàng triệu hay chục triệu ảnh giúp cộng đồng phát triển các mô hình học sâu dễ dàng và hiệu quả hơn với ít dữ liệu. Gần đây, cộng đồng nghiên



Hình 3.6: Lợi ích của học chuyển tiếp đối với việc huấn luyện mô hình [8].

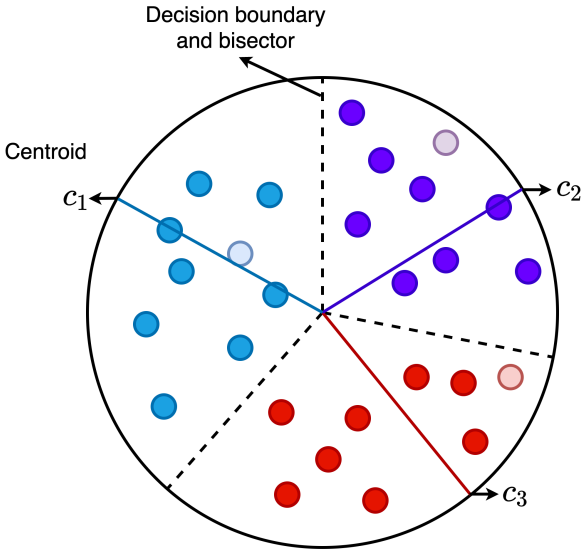
cứu xử lý ngôn ngữ tự nhiên cũng có thể lợi dụng các mô hình huấn luyện sẵn cực lớn với hàng trăm tỉ trọng số như BERT [34], GPT-2 [35] hay GPT-3 [36] để cải tiến các tác vụ hạ lưu như phân loại nhận dạng thực thể, phân tích cú pháp phụ thuộc, tóm tắt văn bản, ...

Trong đồ án, phương pháp học chuyển tiếp từ mô hình huấn luyện trên người nói tiếng Anh được áp dụng để cải thiện mô hình giọng nói tiếng Việt. Mặc dù âm điệu hay đặc điểm của người nói tiếng Anh khác nhiều so với tiếng Việt, kiến thức về âm thanh huấn luyện trên hàng nghìn giờ dữ liệu tiếng Anh khả năng cao sẽ làm tăng cường kết quả trong tiếng Việt.

3.2.3 Hàm mất mát AMP tăng tính phân tách biểu diễn người nói

Như đã mô tả trong phần 3.1.3, hàm mất mát AP khuyến khích biểu diễn câu nói của một người gần hơn tới nguyên mẫu $\boldsymbol{c}_i$ của người đó hơn là các nguyên mẫu khác. Tuy vậy, khoảng cách giữa các câu nói của một người còn khá lớn và các câu khác người nói ở gần biên quyết định hàm softmax có khoảng cách nhỏ. Đây cũng là điểm yếu của hàm softmax mà nhiều nghiên cứu trước đây cũng đã chỉ ra. Biên quyết định yếu có thể gây ra tỉ lệ nhầm lẫn cao khi triển khai mô hình trong thực tế.

Nhận thấy sự hiệu quả của việc dùng hệ số phạt biên trong các hàm mất mát phân loại như AM-Softmax [37] hay [38], tác giả đề xuất hàm mất mát Angular Margin Prototypical (AMP) thêm hệ số phạt biên vào hàm AP. Hàm AMP có thể được chia thành 2 loại AMP-cos hoặc AMP-arc phụ thuộc vào cách thêm hệ số phạt vào điểm tương đồng cô-sin hoặc góc giữa điểm biểu diễn và nguyên mẫu.



Hình 3.7: Mô tả biểu diễn người nói học bởi hàm softmax trong không gian góc. Đường kẻ chấm đen là đường phân giác giữa 2 tâm.

AMP-cos

Trong hàm AMP-cos, hệ số phạt biên được thêm trực tiếp vào điểm tương đồng của 2 câu. Công thức tính điểm tương đồng 3.6 được viết trong Công thức 3.8.

$$\boldsymbol{S}_{i,k} = \begin{cases} w \cdot (\cos\left(\theta_{\boldsymbol{x}_{i,M},\boldsymbol{c}_k}\right) - m) + b, & \text{nếu } i = k \\ w \cdot \cos\left(\theta_{\boldsymbol{x}_{i,M},\boldsymbol{c}_k}\right) + b, & \text{ngược lại} \end{cases} \tag{3.8}$$

trong đó $\theta_{\boldsymbol{x}_{i,M},\boldsymbol{c}_k}$ là góc giữa truy vấn $\boldsymbol{x}_{i,M}$ của người i và nguyên mẫu $\boldsymbol{c}_k$ của người k, w và b là các trọng số học được, và m là hệ số phạt biên. Thay vào Công thức 3.7, ta được hàm mất mát AMP-cos như trong 3.9.

$$L_{AMP-cos} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{w\cdot(\cos\left(\theta_{\boldsymbol{x}_{i,M},\boldsymbol{c}_i}\right)-m)+b}}{e^{w\cdot(\cos\left(\theta_{\boldsymbol{x}_{i,M},\boldsymbol{c}_i}\right)-m)+b} + \sum_{k=1,k\neq i}^{N} e^{w\cdot\cos\left(\theta_{\boldsymbol{x}_{i,M},\boldsymbol{c}_k}\right)+b}} \tag{3.9}$$

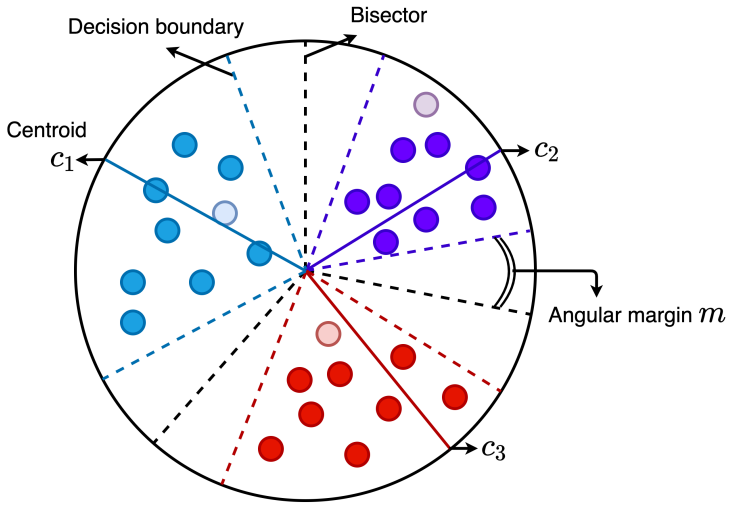
AMP-arc

Hệ số phạt biên được thêm vào góc giữa 2 câu trong hàm AMP-arc. Công thức tính điểm tương đồng 3.6 được thay đổi như trong Công thức 3.10.

$$\boldsymbol{S}_{i,k} = \begin{cases} w \cdot \cos\left(\theta_{\boldsymbol{x}_{i,M},\boldsymbol{c}_k} + m\right) + b, & \text{nếu } i = k \\ w \cdot \cos\left(\theta_{\boldsymbol{x}_{i,M},\boldsymbol{c}_k}\right) + b, & \text{ngược lại} \end{cases} \tag{3.10}$$

Thay công thức trên vào 3.7, ta được hàm mất mát AMP-arc như Công thức 3.11.

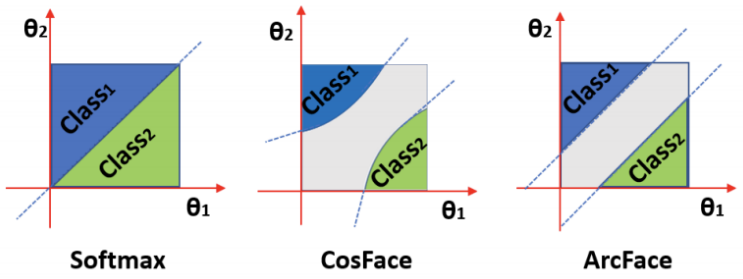
$$L_{AMP-arc} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{w\cdot\cos\left(\theta_{\boldsymbol{x}_{i,M},\boldsymbol{c}_i}+m\right)+b}}{e^{w\cdot\cos\left(\theta_{\boldsymbol{x}_{i,M},\boldsymbol{c}_i}+m\right)+b} + \sum_{k=1,k\neq i}^{N} e^{w\cdot\cos\left(\theta_{\boldsymbol{x}_{i,M},\boldsymbol{c}_k}\right)+b}} \tag{3.11}$$



Hình 3.8: Mô tả biểu diễn người nói học bởi hàm AMP-arc trong không gian góc.

Sự co cụm của các câu người nói (intra-class compactness) và khoảng cách với các người nói khác (inter-class separability) là 2 yếu tố chính đóng góp cho khả năng phân biệt biểu diễn người nói trong không gian vec-tơ. Việc thêm hệ số phạt biên trực tiếp làm tăng khoảng cách giữa các người nói và gián tiếp làm co cụm vùng biểu diễn của một người (Hình 3.8).

Theo các tác giả của hàm ArcFace [38], thiết kế hệ số phạt theo các cách khác nhau có ảnh hưởng rất lớn trong quá trình huấn luyện mô hình. Hàm AMP-arc lấy cảm hứng từ hàm ArcFace có thuộc tính hình học tốt hơn hàm AMP-cos do nó có sự tương ứng chính xác với khoảng cách trong không gian góc. Hình 3.9



Hình 3.9: Biên quyết định của các hàm khác nhau trong phân loại nhị phân [9].

cho thấy trong trường hợp phân loại nhị phân, hàm arc biên quyết định tuyến tính trong toàn không gian trong khi hàm cos có biên quyết định phi tuyến tính.

3.2.4 SGD khái quát hoá tốt hơn Adam

Adam [28] là một thuật toán tối ưu thích nghi hệ số học. Được công bố vào năm 2014, Adam được trình bày tại một hội nghị rất uy tín cho cộng đồng học sâu - ICLR 2015. Bài báo phát triển một thuật toán rất hứa hẹn, cho thấy sự hội tụ vượt trội so với các thuật toán hiện hành, dẫn đến tăng tốc trong quá trình huấn luyện.

Adam thích nghi hệ số học cho một trọng số của mạng sử dụng giá trị trung bình trượt của gradient và gradient bình phương của trọng số đó qua các mini-batch. Cho các trọng số $w^{(t)}$ và hàm mất mát $L^{(t)}$, trong đó t là chỉ số vòng lặp trong quá trình huấn luyện, việc cập nhật trọng số trong Adam như Công thức 3.12.

$$\begin{aligned} m_w^{(t+1)} &\leftarrow \beta_1 m_w^{(t)} + (1-\beta_1)\,\nabla_w L^{(t)} \\ v_w^{(t+1)} &\leftarrow \beta_2 v_w^{(t)} + (1-\beta_2)\left(\nabla_w L^{(t)}\right)^2 \\ \hat{m}_w &= \frac{m_w^{(t+1)}}{1-\beta_1^{t+1}} \\ \hat{v}_w &= \frac{v_w^{(t+1)}}{1-\beta_2^{t+1}} \\ w^{(t+1)} &\leftarrow w^{(t)} - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w} + \epsilon} \end{aligned} \tag{3.12}$$

trong đó ϵ là một hệ số cực bé (ví dụ 10^{-6}) để tránh phép chia cho 0, β_1 và β_2 là hệ số quên cho giá trị trung bình trượt bậc 1 và bậc 2 tương ứng. Phương pháp tối ưu đã được sử dụng trong nhiều ứng dụng do hiệu suất cạnh tranh và

khả năng hoạt động tốt mà không cần điều chỉnh các hệ số. Công thức 3.12 cho thấy kích thước bước nhảy của quy tắc cập nhật trong Adam bất biến với độ lớn của gradient.

Tuy nhiên sau một thời gian, cộng đồng bắt đầu nhận thấy trong một số trường hợp, Adam huấn luyện mô hình tệ hơn so với phương pháp tối ưu truyền thống SGD. Bằng thực nghiệm, nghiên cứu [39] cho thấy Adam dù trong những vòng lặp đầu vượt trội so với SGD nhưng nhanh chóng trì trệ trong những vòng lặp sau. Wilson cùng cộng sự trong [40] chỉ ra rằng các phương pháp thích ứng như Adam hay Adadelta không khái quát hoá SGD sau khi thử nghiệm trên một loạt các tác vụ, không khuyến khích cộng đồng sử dụng các thuật toán thích ứng. Nhìn chung, các nghiên cứu cho thấy rằng tính khái quát hoá của Adam tệ hơn SGD.

Trong bài toán xác minh người nói, người nói trong pha kiểm tra thường không có trong tập huấn luyện, tính khái quát hoá của mô hình là đặc biệt quan trọng. Lý do chính là bởi vì môi trường trong pha kiểm thử khác nhau rất nhiều so với môi trường trong dữ liệu huấn luyện và đăng ký. Do vậy mô hình có tính khái quát hoá cao, nghĩa là ít nhạy cảm với điều kiện của môi trường cho kết quả tốt hơn.

Vì các lý do kể trên, đồ án đề xuất thay thế sử dụng SGD thay cho Adam trong mô hình cơ sở.

3.3 Nghiên cứu liên quan

Như đã trình bày trong phần 3.2, mô hình đề xuất sử dụng học chuyển tiếp, hàm tối ưu SGD và thêm hệ số phạt góc cho hàm mất mát AP. Sau đây, đồ án trình bày một số nghiên cứu liên quan về các phần trong mô hình đề xuất.

Học chuyển tiếp. Phương pháp học chuyển tiếp được ứng dụng tương đối rộng rãi cho bài toán xác minh người nói dưới nhiều dạng khác nhau, chủ yếu để giải quyết vấn đề giữ liệu trong tác vụ mới. Trong các nghiên cứu [41, 42], các tác giả sử dụng phương pháp học đối kháng với mục đích thích ứng mô hình xác minh người nói trên dữ liệu thu bằng micrô sang dữ liệu điện thoại di động, mô hình huấn luyện trên dữ liệu chứa tiếng vang sang dữ liệu sạch và mô hình trên tiếng Anh sang tiếng Trung Quốc. Điểm chung của các tập dữ liệu mục tiêu là thời lượng ít hơn rất nhiều so với tập dữ liệu gốc và không có nhãn người nói. Trong lời giải đứng nhất [10] trong cuộc thi VOXSRC-20, tác giả Thienpondt sử

dụng hàm mất mát HPM để thích ứng mô hình xác minh người nói tiếng Anh sang tiếng Ba Tư với bộ dữ liệu mục tiêu gồm 588 người nói cho kết quả 1.83% EER rất tốt đối với ngôn ngữ nghèo dữ liệu.

Hàm mất mát với hệ số phạt biên. Việc nghiên cứu cách thêm hệ số phạt biên vào hàm mất mát phân loại nhằm tăng tính phân biệt tương đối phổ biến trong bài toán nhận dạng - xác minh mặt người. Để giải quyết điểm yếu biên quyết định yếu của softmax, các công trình [37, 38, 43] đề xuất các hàm A-softmax, CosFace và ArcFace bằng cách thêm hệ số phạt biên theo các cách khác nhau. Hai hàm CosFace và ArcFace trở nên phổ biến cho bài toán xác minh người nói do cài đặt dễ dàng và hiệu năng cao [32, 44].

Các hàm mất mát phân biệt dựa trên cặp hay bộ ba như RLL [45], LS [46] hay hàm triplet đều được thiết kế với hệ số phạt biên. Ngược lại, các hàm mất mát phân biệt vận hành trên một mini-batch nhiều câu nói như GE2E [47], prototypical [48] hay AP [32] đều không được thiết kế với hệ số phạt biên. Gần đây, tác giả Wei cùng cộng sự nghiên cứu đưa hệ số phạt biên vào hàm GE2E cho kết quả đáng mong đợi [49].

Phương thức tối ưu SGD. Hiện nay, theo hiểu biết của tác giả thì chưa có nghiên cứu nào so sánh sự hiệu quả của việc sử dụng SGD thay cho Adam trong xác minh người nói. Hơn nữa các nghiên cứu thường ít bàn luận lý do tại sao lại chọn SGD hay Adam thay vì phương thước tối ưu còn lại. Tuy nhiên, một điểm chung có thể thấy là SGD thường được sử dụng để đạt kết quả hiện đại nhất (state of the art) cho các mạng như ResNet [14], DenseNet [50], ResNeXt [51], SENet [52], ... Trong khi Adam thường được sử dụng cho các mạng lớn hay hệ thống phức tạp như BERT [53], GPT-3 [36], GANs [54] nhờ tính ổn định của nó.

Chương 4

Xây dựng dữ liệu và thực nghiệm

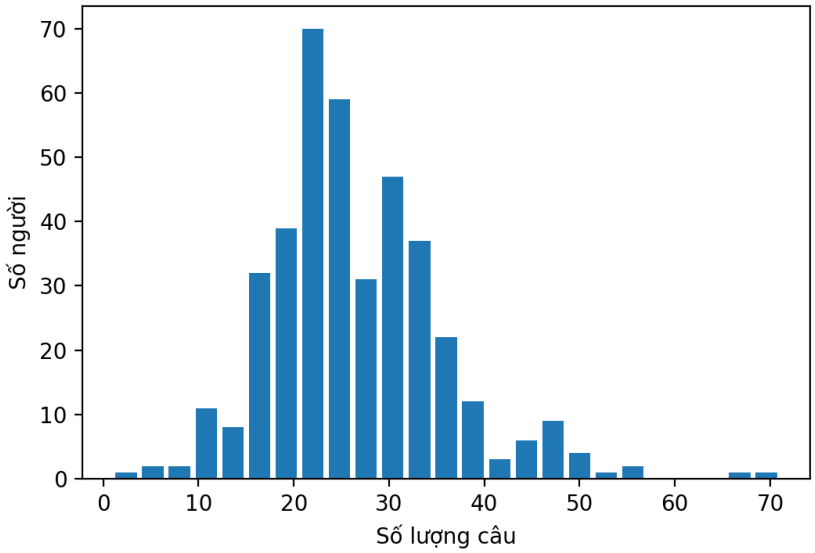
Trong chương 4, đồ án trình bày quy trình xây dựng bộ dữ liệu cho bài toán xác minh người nói tiếng Việt, thông tin chuẩn bị thực nghiệm và kết quả thực nghiệm.

4.1 Xây dựng dữ liệu

4.1.1 Hiện trạng

Bắt đầu từ năm 2018, ZaloAI challenge [24] là một cuộc thi thường niên tập trung vào trí tuệ nhân tạo do Zalo Group, VNG tổ chức. Năm 2020, ZaloAI challenge thử thách các nhà phát triển và kỹ sư học máy Việt Nam với ba bài toán: tóm tắt tin tức, phát hiện biển báo giao thông, và xác thực người nói. Bộ dữ liệu huấn luyện công khai của bài toán xác thực giọng nói bao gồm 400 danh tính với tổng cộng 8.7 giờ âm thanh thu thập từ chương trình truyền hình Bạn muốn hẹn hò. Mỗi danh tính có trung bình 26.4 câu nói với phân phối mô tả trong Hình 4.1. Bộ dữ liệu tuy đa dạng về mặt độ tuổi giới tính tuy nhiên vẫn còn vấn đề như: trùng lặp danh tính, nhiều tạp âm như nhạc nền, người nói phía sau, câu nói của danh tính này lẫn vào danh tính kia, ...

Với 8.7 giờ và 400 danh tính, bộ dữ liệu ZaloAI so với hơn 2,000 giờ và 7,000 danh tính trong bộ dữ liệu VoxCeleb là quá nhỏ bé. Để bổ sung thêm dữ liệu cho bài toán, tác giả sử dụng nguồn dữ liệu công khai nhận dạng tiếng nói tiếng Việt. Các bộ dữ liệu nhận dạng tiếng nói tiếng Việt bao gồm VLSP, INFORE, VIVOS và FPT. Trong đó, chỉ bộ dữ liệu VLSP và VIVOS có nhãn người nói. Đồ án tiến hành khảo sát các bộ dữ liệu VIVOS và VLSP để bổ sung dữ liệu cho bài toán xác minh người nói tiếng Việt.



Hình 4.1: Biểu đồ phân phối số danh tính theo số câu nói của bộ dữ liệu ZaloAI.

VIVOS là tập giọng nói tiếng Việt phục vụ cho bài toán nhận dạng tiếng nói thu thập bởi phòng thí nghiệm khoa học máy tính AILAB từ trường Đại học Khoa học Tự nhiên - Đại học Quốc Gia TP.HCM [55]. Tuy chủ đích của bộ dữ liệu là dành cho nhận dạng tiếng nói nhưng lại có nhãn danh tính cụ thể nên có thể sử dụng cho bài toán nhận dạng người nói. Tập huấn luyện VIVOS bao gồm 40 danh tính với trung bình 253.5 câu nói mỗi người. Tập kiểm thử có 19 danh tính không trùng với tập huấn luyện với trung bình 40 câu nói mỗi người. Chất lượng âm thanh của VIVOS rất tốt do điều kiện thu âm được kiểm soát nên không yêu cầu xử lý gì thêm.

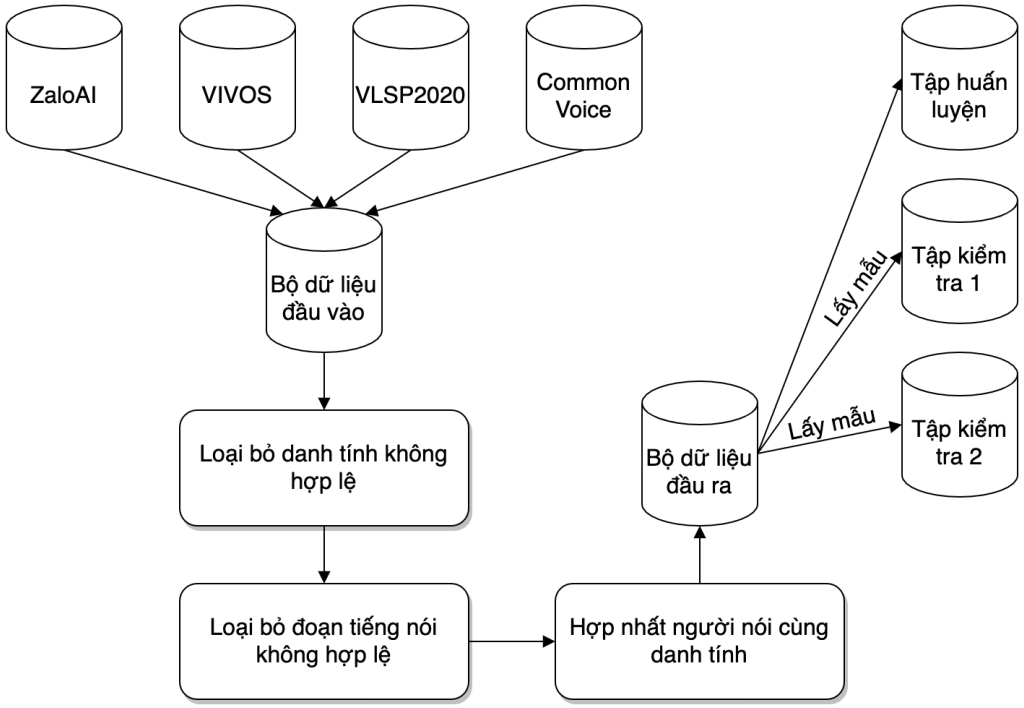
Bộ dữ liệu VLSP [56] nằm trong chiến dịch đánh giá năm 2020 của Hiệp hội xử lý Ngôn ngữ và Tiếng nói tiếng Việt. Giống như VIVOS, VLSP được thu thập và thiết kế cho bài toán nhận diện giọng nói nhưng có nhãn danh tính cho các câu nói. Tổng số danh tính trong VLSP là 567 người với trung bình 22.3 câu mỗi người. Tuy nhiên, dữ liệu danh tính của bộ dữ liệu lại không được chuẩn xác và có nhiều vấn đề tương tự như bộ ZaloAI. Những vấn đề này được giải quyết bằng phương pháp mô tả trong phần 4.1.3.

Ngoài các bộ dữ liệu tương đối lớn được mô tả bên trên, đồ án còn thu thập thêm dữ liệu từ CommonVoice [57]. CommonVoice là dự án nguồn cung cứng cộng đồng bắt nguồn từ Mozilla để tạo các cơ sở dữ liệu miễn phí nhằm phát triển nhận dạng tiếng nói. Bộ dữ liệu tiếng Việt được xác thực đến từ CommonVoice gọi là CommonVoice-vi chỉ bao gồm 23 người và 253 câu nói. Với lượng dữ liệu ít ỏi và tính chất nhìn chung khác với các bộ dữ liệu còn lại. Trong hai tập kiểm

tra, CommonVoice đóng vai trò làm bộ kiểm thử ngoài miền huấn luyện.

4.1.2 Tổng quan quy trình

Bộ dữ liệu phục vụ cho bài toán được xây dựng theo quy trình (Hình 4.2) bắt đầu với việc gộp dữ liệu người nói từ nhiều nguồn khác nhau thành một bộ dữ liệu lớn để làm sạch. Quá trình làm sạch bao gồm ba bước: loại bỏ danh tính không hợp lệ, loại bỏ đoạn tiếng nói không hợp lệ và hợp nhất có cùng danh tính. Bộ dữ liệu đầu ra sau đó được tách thành tập huấn luyện và 2 tập kiểm thử. Chi tiết về làm sạch dữ liệu và bộ dữ liệu được trình bày trong các phần sau.



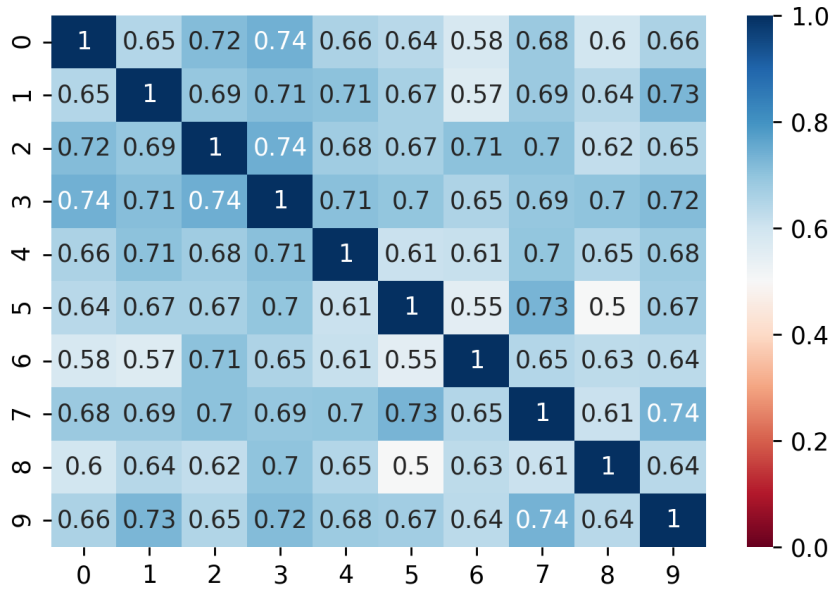
Hình 4.2: Quy trình xây dựng bộ dữ liệu.

4.1.3 Làm sạch dữ liệu

Các bộ ZaloAI, VIVOS, VLSP cùng CommonVoice-vi có tổng cộng gần 1 nghìn danh tính và hơn 20 nghìn câu. Do vậy, việc kiểm tra dữ liệu rất khó khăn và tốn thời gian. Việc này còn trở nên khó khăn hơn khi đánh giá bằng tai người, ví dụ để phân biệt giọng của 2 người cùng là nam, giọng trầm miền bắc thì cần sự tập trung cao độ để tìm điểm khác biệt. Vì thế, đồ án phân tích ma trận tương đồng của các câu nói để tìm ra sự không nhất quán từ đó thu hẹp phạm vi kiểm tra.

Cho một tập biểu diễn n đoạn âm thanh đầu vào $\boldsymbol{V} = \{\boldsymbol{v}_0, \boldsymbol{v}_1, ..., \boldsymbol{v}_{n-1}\}$, sử dụng độ tương đồng cô-sin, ma trận tương đồng cho các đoạn tiếng nói được tính theo Công thức 4.1. Ví dụ một ma trận tương đồng trong Hình 4.3, đường chéo chính có giá trị tương đồng là 1 do so sánh mỗi câu với chính câu đó.

$$\boldsymbol{S}_{i,j} = cos(\boldsymbol{v}_i, \boldsymbol{v}_j) = \frac{\boldsymbol{v}_i \cdot \boldsymbol{v}_j}{\|\boldsymbol{v}_i\| \|\boldsymbol{v}_j\|}, 0 \leq i, j \leq n \tag{4.1}$$



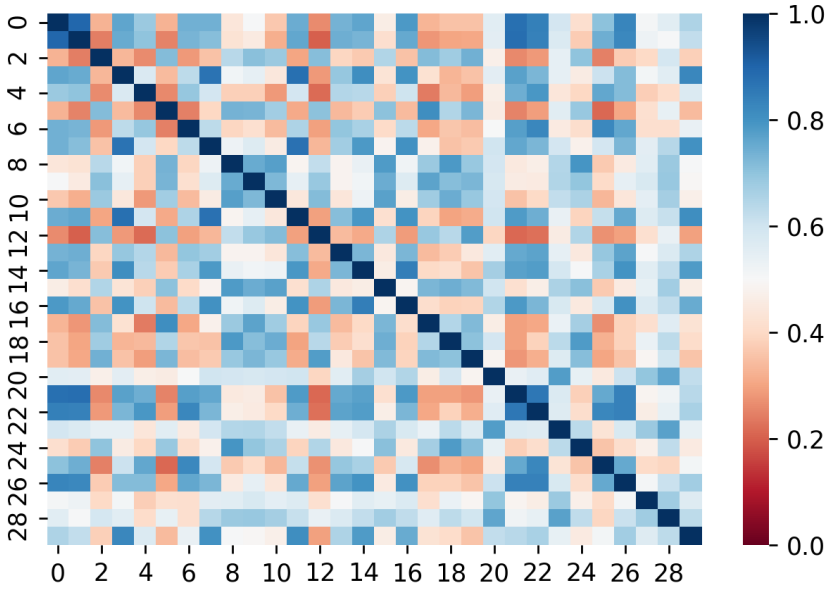
Hình 4.3: Ma trận tương đồng cho một tập 10 đoạn âm thanh của một người.

Ma trận tương đồng được áp dụng khá rộng rãi, trong đó, phổ biến nhất là phân tích âm nhạc dựa trên nội dung [58], phân tích văn bản [59] và tin sinh [60]. Các kĩ thuật được sử dụng chủ yếu là phân cụm và phân đoạn. Trong đồ án, tác giả sử dụng phân tích đơn thuần để tìm ra các người nói, câu nói có khả năng bị gán nhãn sai.

Loại bỏ người nói không hợp lệ

Việc loại bỏ một danh tính có thể do nhiều lý do: nhiều câu nói không thuộc về người đó, chất lượng âm thanh kém, môi trường xung quanh ồn ào, tệp âm thanh bị hư hại qua đường truyền hoặc thiết bị, ... Các nguyên nhân này dẫn đến việc chất giọng của danh tính không được đảm bảo gây bất lợi cho việc huấn luyện mô hình. Một số danh tính có số lượng câu có vấn đề lớn, làm sạch và loại bỏ từng câu bằng việc nghe rất tốn thời gian và công sức. Do vậy, việc loại bỏ hẳn những danh tính này là cần thiết. Khi nhìn vào ma trận tương đồng

của một danh tính, có thể thấy được và loại bỏ những danh tính không hợp lệ. Ma trận tương đồng của một người hợp lệ và bị loại bỏ có thể được thấy trong Hình 4.3 và Hình 4.4 tương ứng.



Hình 4.4: Ma trận tương đồng của một danh tính bị loại bỏ.

Loại bỏ đoạn tiếng nói không hợp lệ

Các đoạn tiếng nói không hợp lệ bao gồm sai nhãn danh tính, độ dài quá ngắn, tiếng ồn xung quanh quá lớn hay trong một đoạn có giọng của nhiều người khác nhau. Các đoạn này làm cho việc huấn luyện mô hình gặp khó khăn và giảm chất lượng của mô hình đầu ra. Lấy ma trận tương đồng của một người có câu nói không hợp lệ (Hình 4.5) làm ví dụ, để lọc ra đoạn có chỉ số 6 khá đơn giản bằng cách lấy một ngưỡng thấp (ví dụ 0.3). Những câu có độ tương đồng so với những câu khác của một danh tính mà dưới ngưỡng này ta sẽ xem là không hợp lệ. Tuy nhiên, cách này không hợp lý với những câu như câu chỉ số 2 trong Hình 4.5, có điểm nằm trong khoảng 0.4 - 0.6. Tuy có điểm tương đồng khá cao nhưng những câu này cũng cần được kiểm tra. Những câu này có thể được tìm thấy bằng cách phát hiện ngoại lệ sử dụng khoảng trong tứ phân vị (Interquartile range) [61].

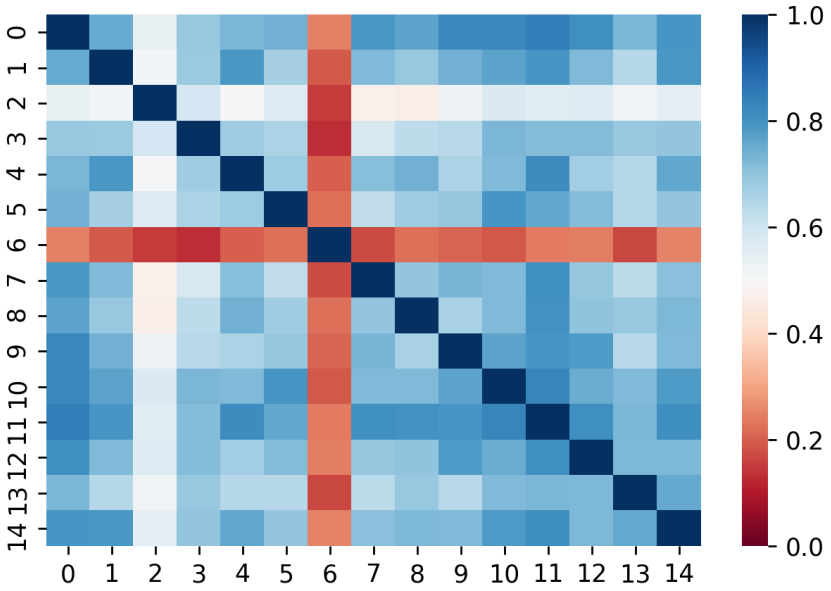
Với, Q1, Q3 lần lượt là tứ phân vị thứ nhất và thứ ba của tập điểm trung bình của các câu $a_i = \frac{1}{n}\sum_{j=0,j\neq i}^{n-1} \boldsymbol{S}_{i,j}$, dựa trên khoảng trong tứ phân vị, đoạn điểm

tương đồng hợp lệ cho tập điểm $\boldsymbol{a}$ được tính theo hai Công thức 4.2 và 4.3.

$$a_{min} = Q1 - 1.5 * IQR;\ a_{max} = Q3 + 1.5 * IQR \tag{4.2}$$

$$IQR = Q3 - Q1 \tag{4.3}$$

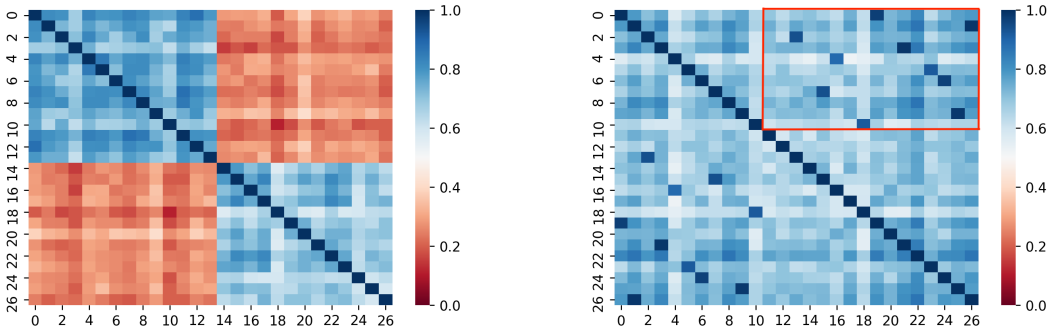
Các câu có điểm trung bình a_i nằm ngoài đoạn $[a_{min}, a_{max}]$ được đánh dấu và cần nghe lại để quyết định có loại bỏ hay không.



Hình 4.5: Ma trận tương đồng của một danh tính có đoạn âm thanh không hợp lệ.

Hợp nhất người nói có cùng danh tính

Do các bộ dữ liệu được thu thập một cách độc lập, có khả năng người nói trong bộ dữ liệu này trùng với bộ kia. Hơn nữa, một người đã tồn tại trong cơ sở dữ liệu cũng có khả năng được yêu cầu thu lại. Các cặp người nói trùng danh tính có thể được tìm dựa vào ma trận tương đồng chéo. Từ Hình a mô tả ma trận tương đồng của người nói 52-M-31 và 64-M-30, có thể thấy rõ đây là 2 người khác nhau do ma trận tương đồng chéo (phần màu đỏ) có điểm tương đồng rất thấp. Ngược lại, trong Hình 4.6, 2 người có nhãn khác nhau là 64-M-30 và 636-M-30 thực chất là cùng một người với ma trận tương đồng chéo nằm trong ô màu đỏ. Có nhiều cặp câu điểm tương đồng cao (ô xanh đậm) nhưng không đạt tới 1.0 như trên đường chéo chính do cơ bản có cùng nội dung nhưng khác



(a) Ma trận tương đồng của 52-M-31 và 64-M-30. (b) Ma trận tương đồng của 64-M-30 và 636-M-30.

Hình 4.6: Ma trận tương đồng cho các cặp người nói khác nhau.

biệt đến từ sự biến đổi nhất định trong quá trình xử lý. Các cặp người nói có giá trị trung bình của ma trận tương đồng chéo lớn hơn 0.7 yêu cầu được nghe lại và ra quyết định để hợp nhất.

4.1.4 Bộ dữ liệu VietSV

Sau khi loại bỏ 65 danh tính không phù hợp, hợp nhất 84 người nói vào người nói khác có cùng danh tính và loại bỏ 1,617 các câu có vấn đề, bộ dữ liệu đã có chất lượng tương đối tốt. Tổng số lượng người nói là 1113, được phân thành 3 tập: tập huấn luyện (training set) gồm 1031 người nói, tập kiểm tra 1 (test set 1) và tập kiểm tra 2 (test set 2). Bộ dữ liệu gồm 3 tập này được đặt tên là VietSV 4.1.

Bảng 4.1: Thông số sàng bộ dữ liệu VietSV

Tập dữ liệu	**Số người nói**	**Số cặp câu**
Tập huấn luyện	1031	-
Tập kiểm tra 1	59	48,148
Tập kiểm tra 2	23	12,192

Tập kiểm tra 1 bao gồm 48,148 cặp đoạn tiếng nói từ 59 người nói với tổng số 1,626 đoạn tiếng nói. Trong đó, 20 người được lấy ngẫu nhiên trong bộ ZaloAI với điều kiện cân bằng giới tính nam - nữ, 19 người còn lại trong tập kiểm tra là tập kiểm thử (development set) của bộ dữ liệu VIVOS. Để tăng độ khó cho các tập kiểm tra, các cặp tiếng nói âm tính được lấy mẫu ngẫu nhiên từ người nói có cùng giới tính và vùng miền. Tập kiểm tra 2 (test set 2) bao gồm 12,192 cặp đoạn tiếng nói từ 23 người nói trong CommonVoice với tổng số 253 câu nói. Cách lấy mẫu của tập kiểm tra 2 cũng tương tự như tập kiểm tra 1. Tuy số lượng câu nói trong tập kiểm tra tương đối nhỏ, độ khó của tập này đến từ việc dữ liệu trong tập này có tính chất khác với tập huấn luyện.

Trong đồ án này, tập huấn luyện được trích ra một nhóm 20 người cân bằng giới tính để sử dụng làm tập kiểm thử (validation set). Trong các thử nghiệm trong phần , đồ án sử dụng tập kiểm tra 1.

4.2 Chuẩn bị thực nghiệm

4.2.1 Môi trường thực nghiệm

Để thực hiện huấn luyện các mô hình, tác giả sử dụng Google Colaboratory: Hệ điều hành Ubuntu 18.04, 2vCPU Intel Xeon 2.2 Ghz, RAM 25GB, GPU Tesla T4 15GB.

Các thực nghiệm trong mục tiếp theo đều được chạy trên cùng bộ thông số như sau:

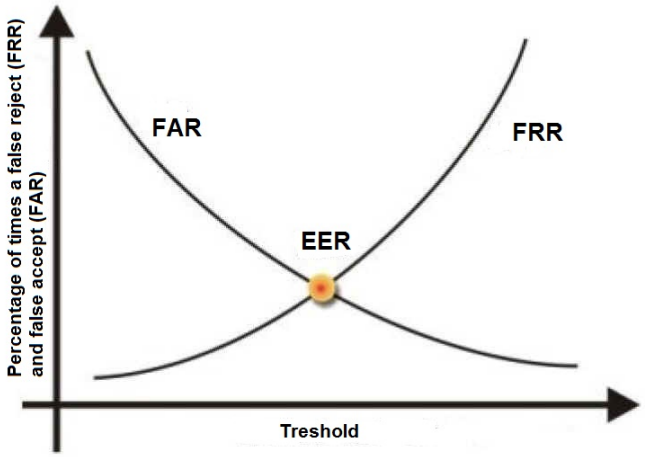
- Đặc trưng âm học: filter banks.
- Tăng cường dữ liệu: nhiễu MUSAN [62], tiếng vang [63].
- Mạng trích xuất đặc trưng: ResNet.
- Lớp tổng hợp: Tổng hợp thống kê tập trung.
- Batch size: 100.

4.2.2 Độ đo đánh giá

Trong phần thực nghiệm, tác giả sử dụng tỉ lệ lỗi bằng nhau (Equal Error Rate - EER) để đánh giá hiệu năng của các mô hình thực nghiệm. EER là điểm nằm cắt nhau giữa đường tỉ lệ chấp nhận giả (False Acceptance Rate - FAR, tỉ lệ mà người xâm nhập được coi là người dùng hợp lệ) và đường tỉ từ chối giả (False Rejection Rate - FRR, tỉ lệ mà người dùng hợp lệ bị từ chối) khi điều chỉnh ngưỡng (Hình 4.7). Mô hình nhận dạng người nói càng hiệu quả thì có EER càng nhỏ.

Hệ thống tính điểm cho một cặp câu bằng việc tính độ tương đồng cô-sin của biểu diễn của 2 đoạn tiếng nói. Hai câu nói của cùng một người nói được coi là hợp lệ; hai câu nói từ 2 người nói khác nhau được coi là câu nói không hợp lệ. Với một ngưỡng cho trước, nếu cặp điểm câu nói hợp lệ dưới ngưỡng này (độ tương đồng thấp), câu nói đó được tính vào tỉ lệ từ chối giả. Ngược lại, nếu một

[1]https://wentzwu.com/2019/05/05/which-is-more-important-accuracy-or-acceptability/



Hình 4.7: Mô tả EER [1].

cặp câu nói không hợp lệ có điểm nằm trên ngưỡng (tỉ lệ tương đồng cao), câu nói được tính vào tỉ lệ chấp nhận giả.

4.2.3 Cài đặt thực nghiệm

Để cài đặt thực nghiệm, tác giả sử dụng ngôn ngữ lập trình Python kết hợp với thư viện PyTorch phiên bản 1.7.1. PyTorch là thư mã nguồn mở của Facebook được xây dựng trên ngôn ngữ lập trình Lua. PyTorch cho phép người dùng xây dựng, tuỳ biến mô hình ở cả cấp cao và cấp thấp với thiết kế trực quan.

Trong đồ án, tác giả sử dụng mã nguồn mô hình trong [32] và cài đặt các phần liên quan tới huấn luyện và các phương pháp đề xuất. Ngoài ra, tác giả cũng sử dụng mã nguồn mô hình ECAPA [10] và cài đặt các phần huấn luyện để so sánh với phương án đề xuất.

4.3 Kết quả thực nghiệm và đánh giá

Tác giả thực hiện nhiều trường hợp thực nghiệm khác nhau với mục tiêu huấn luyện mô hình nhận dạng người nói một cách hiệu quả trên tiếng Việt. Thực nghiệm 1 đánh giá hiệu quả của việc làm sạch dữ liệu. Thực nghiệm 2 khảo sát học chuyển tiếp so với các cách huấn luyện khác nhau. Thực nghiệm 3 đánh giá hiệu quả trong việc khử nhiễu âm thanh tín hiệu giọng nói. Thực nghiệm 4 so sánh mô hình khi sử dụng phương thức tối ưu Adam và SGD. Thực nghiệm 5 kiểm tra tính hiệu quả của hàm mất mát AMP-cos và AMP-arc với các giá trị hệ số phạt khác nhau. Thực nghiệm 6 so sánh mô hình đề xuất với mô hình ECAPA sử dụng học chuyển tiếp trong [10].

4.3.1 Thực nghiệm 1: Làm sạch dữ liệu

Bảng 4.2 mô tả kết quả thực nghiệm với dữ liệu ban đầu và dữ liệu đã được làm sạch như đã trình bày trong phần 4.1.3. Như có thể thấy, kết quả huấn luyện trên tập đã sàng lọc cải thiện 0.93% EER so với dữ liệu gốc. Do vậy, các thực nghiệm về sau sẽ sử dụng bộ dữ liệu đã qua sàng lọc.

Bảng 4.2: EER trên tập kiểm tra với dữ liệu trước và sau khi cải thiện chất lượng

Dữ liệu	**EER trên tập kiểm tra**
Bộ dữ liệu ban đầu	8.532%
Bộ dữ liệu VietSV	**7.602%**

4.3.2 Thực nghiệm 2: Học chuyển tiếp

Trong thực nghiệm này, tác giả tiến hành khảo sát các phương thức huấn luyện mô hình. Bảng 4.3 mô tả kết quả với các trường hợp khác nhau bao gồm: mô hình huấn luyện sẵn [19], huấn luyện từ đầu trên dữ liệu tiếng Anh và tiếng Việt, huấn luyện từ đầu chỉ trên VietSV, học chuyển tiếp trên VietSV.

Do số người nói trong bộ dữ liệu VietSV ít hơn hẳn so với người nói tiếng Anh trong bộ dữ liệu VoxCeleb, huấn luyện không tập trung đủ để tìm ra các đặc trưng hữu ích để phân biệt người nói tiếng Việt, dẫn đến mô hình huấn luyện từ đầu kết hợp hai bộ dữ liệu đạt kết quả tệ hơn. Mô hình cơ sở (mô tả trong phần 3.1) huấn luyện từ đầu trên VietSV đạt kết quả 7.602% EER. Học chuyển tiếp bằng riêng dữ liệu tiếng Việt cho kết quả tốt hơn so với huấn luyện từ đầu bằng trên VietSV hoặc kết hợp VoxCeleb với EER 5.890%. Kết quả cho thấy các đặc trưng người nói trong tiếng Anh góp phần cải thiện mô hình tiếng Việt. Các thử nghiệm phía sau sử dụng phương pháp học chuyển tiếp.

Bảng 4.3: EER trên tập kiểm tra với các phương pháp huấn luyện và dữ liệu khác nhau

Phương pháp huấn luyện	**Dữ liệu huấn luyện**	**EER trên tập kiểm tra**
Mô hình cơ sở	VoxCeleb	14.954%
Mô hình cơ sở	VoxCeleb + VietSV	8.499%
Mô hình cơ sở	VietSV	7.602%
Học chuyển tiếp	**VietSV**	**5.890%**

4.3.3 Thực nghiệm 3: Khử tạp âm trong tín hiệu tiếng nói

Do chỉ có VIVOS là được thu thập từ phòng thu âm, tín hiệu tiếng nói trong tập dữ liệu còn chứa nhiều tạp âm, ví dụ nhạc nền, tiếng ồn nhỏ xung quanh, âm thanh đường phố xe cộ. Các nhiễu tạp âm có khả năng cản trở mô hình học

được giọng cần học từ dữ liệu. Do đó, tác giả sử dụng dữ liệu khử tạp âm dùng mô hình do bộ phận nghiên cứu trí tuệ nhân tạo tại Facebook phát triển [64] và đánh giá hiệu quả trong thực nghiệm này. Bảng 4.4 cho thấy việc khử tạp âm có hiệu quả cao với 0.444% EER cải thiện trên tập kiểm tra. Do vậy, trong các thực nghiệm tiếp theo, các mô hình được huấn luyện trên dữ liệu đã lọc tạp âm.

Bảng 4.4: EER trên tập kiểm tra của mô hình huấn luyện với dữ liệu còn nhiễu âm thanh và đã khử tạp âm

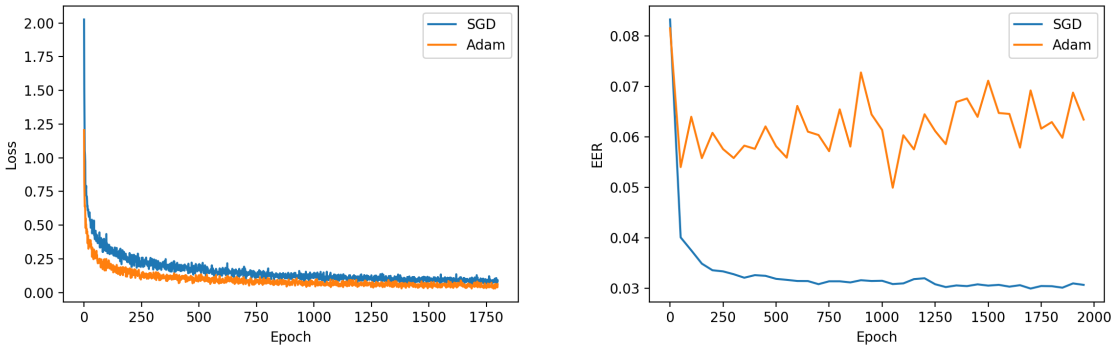
Dữ liệu huấn luyện	**EER trên tập kiểm tra**
Chứa tạp âm	5.890%
Khử tạp âm	**5.446%**

4.3.4 Thực nghiệm 4: Phương pháp tối ưu SGD

Bảng 4.5 cung cấp kết quả 2 mô hình huấn luyện bằng phương pháp tối ưu Adam và SGD. Kết quả huấn luyện với SGD cải thiện tới 1.907% so với mô hình huấn luyện với Adam. Thật vậy, khi quan sát giá trị của hàm mất mát trên tập huấn luyện trong hình 4.8a, sự hội tụ của Adam khá nhất quán, thậm chí có phần tốt hơn SGD. Tuy nhiên, trong phần 4.8b, kết quả EER trên tập kiểm thử của Adam rất không nhất quán ngay từ những vòng lặp đầu, trong khi SGD hội tụ rất tốt trên tập kiểm thử và không có dấu hiệu học quá khớp. Điều này chứng minh khả năng khái quát hoá vượt trội của SGD so với Adam cho bài toán.

Bảng 4.5: EER trên tập kiểm tra của mô hình huấn luyện với Adam và SGD

Phương pháp tối ưu	**EER trên tập kiểm tra**
Adam	5.446%
SGD	**3.539%**



(a) Giá trị hàm mất mát trên tập huấn luyện.

(b) EER trên tập kiểm thử.

Hình 4.8: So sánh giá trị mất mát và EER của mô hình huấn luyện với SGD và Adam qua các vòng lặp.

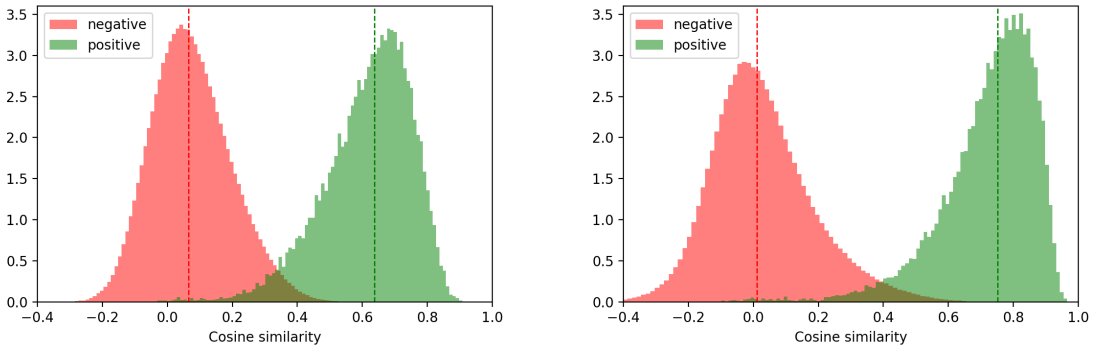
Các mô hình trong thực nghiệm 5 sử dụng phương pháp tối ưu SGD.

4.3.5 Thực nghiệm 5: Hàm mất mát AMP

Bảng 4.6 mô tả kết quả thử nghiệm khi sử dụng hai phiên bản phạt biên khác nhau AMP-cos, AMP-arc của hàm AP. Nhìn chung, việc sử dụng hàm AMP có hiệu quả tích cực khi huấn luyện mô hình. Hai hàm AMP-cos và AMP-arc với hệ số phạt lần lượt là $m = 0.4$ và $m = 0.2$ cho kết quả 3.211% và 3.115% tương ứng. Với các giá trị khác nhau của m, AMP luôn tốt hơn so với hàm AP thông thường. Trong AMP-arc, với m lớn hơn 0.2, kết quả có xu hướng tệ đi khi tăng giá trị của m do hiện tượng quá khớp.

Bảng 4.6: EER trên tập kiểm tra của mô hình huấn luyện với hàm mất mát AP, AMP-cos và AMP-arc với các giá trị phạt khác nhau

Hàm mất mát	m	**EER trên tập kiểm tra**
AP	0.0	3.539%
AMP-cos	0.1	3.319%
AMP-cos	0.2	3.269%
AMP-cos	0.3	3.232%
AMP-cos	0.4	3.211%
AMP-cos	0.5	3.331%
AMP-arc	0.1	3.240%
AMP-arc	**0.2**	**3.115%**
AMP-arc	0.3	3.194%
AMP-arc	0.4	3.298%
AMP-arc	0.5	3.352%



(a) Mô hình huấn luyện với AP

(b) Mô hình huấn luyện với AMP-arc (m=0.2)

Hình 4.9: Phân bố điểm tương đồng của các cặp câu dương tính và âm tính. Các đường nét đứt mô tả giá trị trung bình điểm trương đồng.

Để hiểu rõ hơn về tính phân loại của biểu diễn người nói, phân bố điểm tương đồng cho các cặp câu dương tính và âm tính được thể hiện trong Hình 4.9. Hình 4.9a thể hiện phân bố điểm đoán bởi mô hình huấn luyện với hàm AP, Hình 4.9b thể hiện phân bố điểm đoán bởi mô hình huấn luyện với hàm AMP-arc (m=0.2). Hiệu quả của tính nhận dạng được quyết định bởi phần chồng lên nhau ở giữa của hai phân bố dương tính và âm tính. Với hàm AMP-arc, phần đuôi chồng nhau của hai phân bố nhỏ hơn và phần thân của hai phân bố nghiêng về phía ngược lại đối với phần chồng nhau. Ngoài ra, khoảng cách của điểm trung bình

dương và âm tính được tăng lên gần 0.2 so với hàm AP, từ đó tăng tính phân loại.

4.3.6 Thực nghiệm 6: So sánh mô hình đề xuất và ECAPA

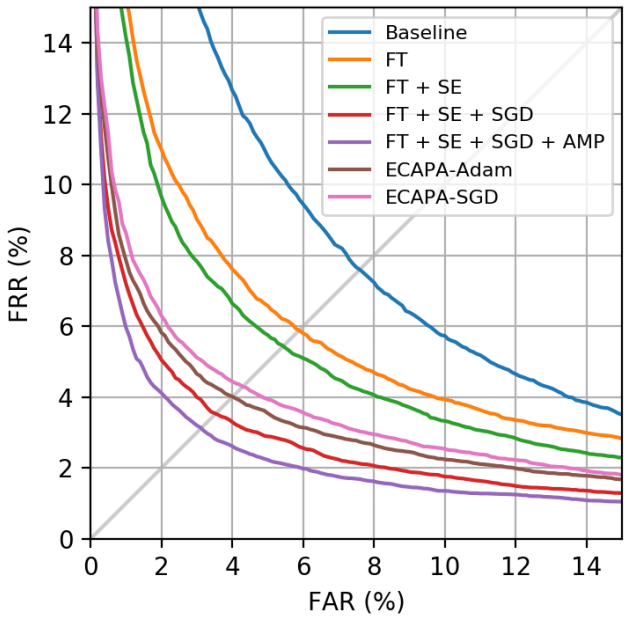
Bảng 4.7 mô tả kết quả giữa mô hình đề xuất và mô hình ECAPA đề xuất trong [10]. Mô hình ECAPA sử dụng học chuyển tiếp dùng hàm mất mát HPM có mục tiêu tương tự đồ án: phát triển mô hình xác minh người nói cho ngôn ngữ ít dữ liệu trong khuôn khổ cuộc thi SdSV 2020.

Bảng 4.7: EER của phương pháp đề xuất và phương pháp [10]

Phương pháp tối ưu	**EER trên tập kiểm tra**
AMP-arc	**3.115%**
ECAPA-Adam	4.017%
ECAPA-SGD	4.299%

4.3.7 Tổng kết kết quả thực nghiệm

Qua 6 phần thực nghiệm trình bày ở trên, có thể thấy rằng việc xử lý dữ liệu bao gồm làm sạch dữ liệu giúp cải thiện mô hình đầu ra. Phương pháp kết hợp sử dụng học chuyển tiếp, khử tạp âm tín hiệu giọng nói, SGD, và AMP-arc (m=0.2) đạt 3.115% EER cải thiện 4.487% so với mô hình cơ sở với 7.602% EER trên tập kiểm tra.



Hình 4.10: Đường cong DET với mô hình cơ sở và các cải tiến. Đường EER $x = y$ màu xám. FT: Học chuyển tiếp, SE: speech enhancement - khử tạp âm.

Đường cong đánh đánh đổi lỗi phát hiện (detection error tradeoff - DET) cho mô hình cơ sở và các mô hình cải tiến được mô tả trong Hình 4.10. Các điểm trên một đường đại diện cho một cặp FAR và FRR tại một ngưỡng nhất định. Có thể thấy rằng mô hình đề xuất tốt hơn mô hình cơ sở trên mọi điểm. Giả sử do nhu cầu mà một hệ thống muốn tỉ lệ chấp nhận nhầm kẻ xấu FAR là 2%, mô hình đề xuất cho tỉ lệ từ chối nhầm FRR tương đối thấp xấp xỉ 4% trong khi mô hình cơ sở cho kết quả cao tới khoảng 17-18% FRR.

Chương 5

Kết luận và hướng phát triển

5.1 Kết luận

Trong đồ án này, tác giả đã xây dựng bộ dữ liệu xác minh người nói tiếng Việt - VietSV bằng cách tổng hợp từ các nguồn ZaloAI, VIVOS, VLSP và CommonVoice. Dữ liệu tổng hợp được làm sạch bằng việc loại bỏ các người nói không hợp lệ, loại bỏ câu nói không hợp lệ và hợp nhất người nói có cùng danh tính. Bộ dữ liệu đầu ra được phân thành ba tập: tập huấn luyện, tập kiểm tra 1 và tập kiểm tra 2. Trong đó các tập kiểm tra, các cặp câu được lấy mẫu từ người nói có cùng giới tính và vùng miền nhằm tăng độ khó.

Đồ án đã đề xuất mô hình sử dụng phương pháp học chuyển tiếp với mong muốn tận dụng kiến thức của mô hình huấn luyện trên dữ liệu tiếng Anh để bổ trợ cho tiếng Việt. Nhận thấy dữ liệu còn nhiều tạp âm, đồ án thử nghiệm huấn luyện mô hình với dữ liệu khử tạp âm giúp cải thiện 0.444% EER so với mô hình huấn luyện trên bộ dữ liệu còn tạp âm. Thử nghiệm phương pháp tối ưu cho thấy SGD giúp mô hình khái quát hoá tốt hơn trên tập dữ liệu so với Adam. Vấn đề biên quyết định yếu của hàm softmax được khắc phục bằng cách đưa hệ số phạt biên vào hàm AP theo hai cách khác nhau gọi là AMP-cos và AMP-arc. Qua phân tích phân bố điểm tương đồng, AMP-arc cho hiệu quả rõ rệt so với hàm AP trong việc phân tách phổ điểm. Mô hình đề xuất với các cải tiến nói trên đạt kết quả 3.115% EER vượt trội so với mô hình cơ sở với kết quả 7.602% EER.

Kết quả của đồ án đã được tổng hợp và nộp tại Hội nghị Châu Á Thái Bình Dương về Ngôn ngữ, Thông tin và Tính toán (PACLIC) với tiêu đề "Speaker Verification Model with Angular Margin Prototypical Loss for Low-Resource Languages and Vietnamese Datasets".

5.2 Hướng phát triển

Trong tương lai, để cải thiện chất lượng mô hình đồ án sẽ thu thập thêm dữ liệu người nói từ nguồn Youtube. Ngoài ra, đồ án sẽ đưa nhóm người có giọng nói tương tự vào cùng một mini-batch để huấn luyện mô hình có tính phân tách cao hơn. Hơn nữa, điểm tương đồng sau khi dự đoán của mô hình cũng có thể được chuẩn hoá dựa trên biểu diễn của một nhóm người đa dạng vùng miền giới tính, từ đó tăng khả năng phân biệt. Hiện tại, miền dữ liệu cũng là một vấn đề với mô hình; trong tương lai, tác giả sẽ thử nghiệm phương pháp học đối kháng để giảm ảnh hưởng của miền dữ liệu (tạp âm, các thiết bị thu âm khác nhau, ...) đối với biểu diễn người nói.

Tài liệu tham khảo

[1] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE, 2018.

[2] Muhamad Yani et al. Application of transfer learning using convolutional neural network method for early detection of terry's nail. In *Journal of Physics: Conference Series*, volume 1201, page 012052. IOP Publishing, 2019.

[3] Jiazhi Ni, Jie Liu, Chenxin Zhang, Dan Ye, and Zhirou Ma. Fine-grained patient similarity measuring using deep metric learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1189–1198, 2017.

[4] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056. IEEE, 2014.

[5] Bidhan Barai, Debayan Das, Nibaran Das, Subhadip Basu, and Mita Nasipuri. An asr system using mfcc and vq/gmm with emphasis on environmental dependency. In *2017 IEEE Calcutta Conference (CALCON)*, pages 362–366. IEEE, 2017.

[6] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. Attentive statistics pooling for deep speaker embedding. *arXiv preprint arXiv:1803.10963*, 2018.

[7] Dipanjan Sarkar, Raghav Bali, and Tamoghna Ghosh. *Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras*. Packt Publishing Ltd, 2018.

[8] Yusuf Aytar and Andrew Zisserman. Tabula rasa: Model transfer for object category detection. In *2011 international conference on computer vision*, pages 2252–2259. IEEE, 2011.

[9] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[10] Jenthe Thienpondt, Brecht Desplanques, and Kris Demuynck. Cross-lingual speaker verification with domain-balanced hard prototype mining and language-dependent score normalization. *arXiv preprint arXiv:2007.07689*, 2020.

[11] Douglas A Reynolds and Richard C Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE transactions on speech and audio processing*, 3(1):72–83, 1995.

[12] Patrick Kenny, Mohamed Mihoubi, and Pierre Dumouchel. New map estimators for speaker recognition. In *Eighth European Conference on Speech Communication and Technology*, 2003.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing*, 37(3):328–339, 1989.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language*, 60:101027, 2020.

[18] Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson. The speakers in the wild (sitw) speaker recognition database. In *Interspeech*, pages 818–822, 2016.

[19] Hee Soo Heo, Bong-Jin Lee, Jaesung Huh, and Joon Son Chung. Clova baseline system for the voxceleb speaker recognition challenge 2020. *arXiv preprint arXiv:2009.14153*, 2020.

[20] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

[21] NGUYEN TU HA and NGO QUOC HUNG. A speaker recognition system using combination method between vector quantization and gaussian mixture model.

[22] Diep Dao Thi Thu, Loan Trinh Van, Quang Nguyen Hong, and Hung Pham Ngoc. Text-dependent speaker recognition for vietnamese. In *2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR)*, pages 196–200, 2013. doi: 10.1109/SOCPAR.2013.7054126.

[23] Son T Nguyen, Viet D Lai, Quyen Dam-Ba, Anh Nguyen-Xuan, and Cuong Pham. Vietnamese speaker authentication using deep models. In *Proceedings of the Ninth International Symposium on Information and Communication Technology*, pages 177–184, 2018.

[24] Zalo ai challenge. `https://challenge.zalo.ai/`. Truy cập vào: 10-05-2021.

[25] Tom M Mitchell et al. Machine learning. 1997.

[26] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[27] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[29] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[32] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han. In defence of metric learning for speaker recognition. *arXiv preprint arXiv:2003.11982*, 2020.

[33] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2593–2601, 2017.

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[35] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[36] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[37] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[38] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[39] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017.

[40] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *arXiv preprint arXiv:1705.08292*, 2017.

[41] Phani Sankar Nidadavolu, Saurabh Kataria, Jesús Villalba, and Najim Dehak. Low-resource domain adaptation for speaker recognition using cycle-gans. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 710–717. IEEE, 2019.

[42] Wei Xia, Jing Huang, and John HL Hansen. Cross-lingual text-independent speaker verification using unsupervised adversarial discriminative domain adaptation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5816–5820. IEEE, 2019.

[43] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[44] Xu Xiang, Shuai Wang, Houjun Huang, Yanmin Qian, and Kai Yu. Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1652–1656. IEEE, 2019.

[45] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. Ranked list loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5207–5216, 2019.

[46] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012, 2016.

[47] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[48] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[49] Yuheng Wei, Junzhao Du, and Hui Liu. Angular margin centroid loss for text-independent speaker recognition. *Proc. Interspeech 2020*, pages 3820–3824, 2020.

[50] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[51] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[52] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[53] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[54] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[55] Vivos corpus. `https://ailab.hcmus.edu.vn/vivos`. Truy cập vào: 10-05-2021.

[56] Automatic speech recognition for vietnamese. `https://vlsp.org.vn/vlsp2020/eval/asr`. Truy cập vào: 10-05-2021.

[57] Commonvoice. `https://commonvoice.mozilla.org/en`. Truy cập vào: 10-05-2021.

[58] Diego F Silva, Chin-Chia M Yeh, Yan Zhu, Gustavo EAPA Batista, and Eamonn Keogh. Fast similarity matrix profile for music analysis and exploration. *IEEE Transactions on Multimedia*, 21(1):29–38, 2018.

[59] Syed Fawad Hussain, Gilles Bisson, and Clément Grimal. An improved co-similarity measure for document clustering. In *2010 Ninth International Conference on Machine Learning and Applications*, pages 190–197. IEEE, 2010.

[60] A Bustamam, F Zubedi, and Titin Siswantining. Implementation χ–sim co-similarity and agglomerative hierarchical to cluster gene expression data of lymphoma by gene and condition. In *AIP Conference Proceedings*, volume 2023, page 020221. AIP Publishing LLC, 2018.

[61] Jiawei Yang, Susanto Rahardja, and Pasi Fränti. Outlier detection: how to threshold outlier scores? In *Proceedings of the international conference on artificial intelligence, information processing and cloud computing*, pages 1–6, 2019.

[62] David Snyder, Guoguo Chen, and Daniel Povey. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*, 2015.

[63] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5220–5224. IEEE, 2017.

[64] Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain. In *Interspeech*, 2020.