TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN ĐIỆN TỬ - VIỄN THÔNG

BÁO CÁO THỰC TẬP CUỐI KHÓA

Đề tài:

NGHIÊN CỨU THIẾT KẾ CÁC PHƯƠNG PHÁP PHÁT HIỆN ÂM THANH BẤT THƯỜNG

Nơi thực tập: SPARC Lab – Phòng 618 thư viện Tạ Quang Bửu

Nhóm thực hiện:

Sinh viên	**MSSV**	**Lớp**
Nguyễn Duy Quang	20152956	Điện tử 06 – K60
Nguyễn Minh Hiếu	20151336	Điện tử 03 – K60
Nguyễn Đình Quốc	20153060	Điện tử 06 – K60

Giảng viên hướng dẫn: TS. Hàn Huy Dũng

TS. Nguyễn Thị Kim Thoa

Hà Nội, 5/2020

LỜI NÓI ĐẦU

Trong đợt thực tập cuối khóa của sinh viên K60 Viện Điện Tử - Viễn Thông, Đại học Bách Khoa Hà Nội, em đã có cơ hội thực tập tại Phòng thí nghiệm Xử lí tín hiệu và Truyền thông vô tuyến (Signal Processing and Radio Communications Laboratory – SPARC Lab) dưới sự hướng dẫn của TS. Hàn Huy Dũng. Trong khoảng thời gian này em có cơ hội được tiếp xúc với môi trường làm việc thực tế, chuyên nghiệp, được vận dụng những kiến thức và kỹ năng đã học ra ngoài thực tế. Đồng thời em còn học hỏi được những kinh nghiệm, kiến thức quý báu trong công việc và cuộc sống từ những người đi trước để có thể chuẩn bị những hành trang tốt nhất cho tương lai.

Thời gian thực tập tại SPARC Lab thực sự rất thuận lợi và mang lại nhiều kiến thức bổ ích và kinh nghiệm làm việc cho em. Về điều kiện khách quan thầy Hàn Huy Dũng có chuyên môn cao, rất tận tình với sinh viên, luôn sẵn sàng giải thích các vấn đề cho sinh viên hiểu rõ, tạo điều kiện tốt cho sinh viên đạt hiệu quả làm việc tốt nhất. Bên cạnh đó, các anh chị trên lab SPARC của thầy rất thân thiện, nhiệt tình và luôn sẵn sàng giải đáp những thắc mắc. Điều này đã giúp em có cơ hội được có sát nhiều với thực tế. Đồng thời sau mỗi công việc các anh chị luôn dành thời gian để sửa những lỗi sai, hướng dẫn cách làm đúng đắn và nhanh nhất, ngoài ra còn gợi ý cho em tìm hiểu thêm các vấn đề liên quan đến Điện tử - Viễn thông, qua đó em có thể rút ra cho mình những bài học và vận dụng cho những trường hợp sau này. Hơn thế nữa, trong thời gian thực tập em cũng học hỏi thêm một số kỹ năng khác như kỹ năng trình bày vấn đề, kỹ năng tìm sự kiện và lọc thông tin nhanh … Ngoài ra với việc được tiếp xúc với các mọi người trong Lab, em học hỏi được thêm nhiều điều trong cuộc sống ngoài công việc, điều này cũng là một trang bị tốt cho hành trang khởi nghiệp sau này.

Bên cạnh những thuận lợi thì em cũng phải đối mặt với những khó khăn nhất định. Đầu tiên là do đây là lần đầu tiếp xúc với công nghệ cao như Deep Learning, yêu cầu nhiều thuật toán phức tạp nên ban đầu em đã cảm thấy rất khó khăn để thích nghi và tốc độ tiếp thu và làm việc vẫn còn chậm. Đây là lần đầu tiên em phải vận dụng những kiến thức đã học vào công việc thực tế nên còn rất nhiều bỡ ngỡ bởi giữa lý thuyết và thực tế khác nhau rất nhiều. Tuy nhiên mỗi khi gặp khó khăn thầy và các anh chị trong lab cũng luôn sẵn sang giúp đỡ và hướng dẫn giải thích tận tình nên em đã dần bắt

nhịp được với công việc. Ngoài ra, em cũng nhận thấy một số hạn chế của bản thân như: kinh nghiệm thực tế chưa có, kỹ năng đọc tài liệu tiếng anh còn bị hạn chế rất nhiều.

Em xin chân thành gửi lời cảm ơn TS. Hàn Huy Dũng và các anh chị đã giúp đỡ, chỉ bảo và tạo điều kiện để em có thể hoàn thành những công việc được giao. Em cũng xin cảm ơn TS. Nguyễn Thị Kim Thoa, cô đã đưa ra những lời khuyên bổ ích và giúp đỡ về mặt kiến thức trong thời gian em thực tập ở lab.Trong quá trình thực tập mặc dù em đã cố gắng hết sức để hoàn thành những công việc được giao tuy nhiện đây là lần đầu tiên em được tiếp xúc với công việc thực tế, môi trường làm việc chuyên nghiệp và do những hạn chế về kinh nghiệm cũng như kĩ năng, nên em không thể tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp từ các thầy cô để em có thể hoàn thiện những kĩ năng của bản thân.

Em xin chân thành cảm ơn.

Hà Nội, tháng 5 năm 2020

Sinh viên thực hiện

Nguyễn Minh Hiếu

MỤC LỤC

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Từ viết tắt	Từ viết đầy đủ	Nghĩa tiếng Việt
SPARC	Signal Processing and Radio Communications	Xử lí tín hiệu và Truyền thông vô tuyến
AI	Artificial Intelligence	Trí thông minh nhân tạo
PAL	Perceptron Learning Algorithm	Thuật toán học Perceptron
MLP	Multi-layer Perceptron	Perceptron đa tầng
ML	Machine Learning	Học máy
DL	Deep Learning	Học sâu
ANN	Artificial Neural Network	Mạng neural nhân tạo
ASD	Abnormal Sound Detection	Phát hiện âm thanh bất thường
ASC	Acoustic Scene Classification	Phân loại ngữ cảnh âm thanh
AED	Acoustic Event Detection	Phát hiện sự kiện âm thanh
GMM	Gaussian Mixture Model	Mô hình hỗn hợp Gaussian
SVM	Support Vector Machine	Máy vector hỗ trợ
AE	Autoencoder	Bộ mã hóa tự động
FNN	Feedforward Neural Network	Mạng neural truyền thẳng
CNN	Convolutional Neural Network	Mạng neural tích chập
CRNN	Convolutional Recurrent Neural Network	Mạng neural tích chập truy hồi
VAE	Variational Autoencoder	Bộ mã hóa tự động biến thể
STFT	Short-time Fourier Transform	Biến đổi Fourier thời gian ngắn
PCA	Principal Component Analysis	Phân tích thành phần chính

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

TÓM TẮT BÁO CÁO

Báo cáo này trình bày nghiên cứu về các phương pháp phát hiện các sự kiện âm thanh bất thường sử dụng mạng Neural nhân tạo, bao gồm: Autoencoder và Variational Autoencoder. Bộ dữ liệu được sử dụng trong dự án này là các bản ghi âm cuộc họp bằng tiếng Nhật. Bộ dữ liệu được thực hiện và cung cấp bởi công ty đối tác của SPARC Laboratory, Tổng Công Ty SI Synergy Technology. Kết quả thực hiện của hai phương pháp này sẽ được so sánh với nhau, từ đó nhận xét về ưu điểm và nhược điểm của từng phương pháp.

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

Chương 1 giới thiệu và nêu ra tính cần thiết của đề tài, cũng như trình bày sơ lược về trí thông minh nhân tạo, học máy, học sâu, mạng neural nhân tạo và xem xét các phương pháp hiện có trong việc phát hiện âm thanh bất thường

1.1 Giới thiệu đề tài

Để đáp ứng các nhu cầu về an toàn công cộng đang ngày càng gia tăng trên thế giới hiện nay, các hệ thống giám sát dựa trên video đã được phát triển để có thể tự động phát hiện người hoặc vật thể khả nghi, cũng như các hệ thống an ninh dựa trên âm thanh có khả năng phát hiện những sự kiện âm thanh bất thường. Các hệ thống giám sát dựa trên video hiện hoạt động rất hiệu quả, tuy nhiên bị hạn chế về vị trí lắp đặt camera nên khó có thể giám sát toàn bộ khu vực. Ở chiều ngược lại, giám sát bằng âm thanh không có hạn chế về điểm mù, và chi phí lắp đặt micro thu âm rẻ hơn nhiều so với camera. Do đó, các hệ thống an ninh dựa trên âm thanh có thể bù đắp nhược điểm của hệ thống camera, và kết hợp hệ thống giám sát bằng âm thanh và video với nhau có thể tạo nên những hệ thống thông minh hơn.

Hiện nay các công nghệ về trí thông minh nhân tạo, hay còn gọi là AI (Artificial Intelligence), đã được nghiên cứu phát triển và ứng dụng ở nhiều lĩnh vực trong cuộc sống. Tuy nhiên nghiên cứu các hệ thống phát hiện bất thường dựa trên âm thanh là một đề tài còn mới, chưa thực sự phổ biến như các hệ thống giám sát bằng camera. Đứng trên góc nhìn là sinh viên ngành Điện tử - Viễn thông và được sự gợi ý đề tài từ thầy Hàn Huy Dũng, em đã tìm hiểu, nghiên cứu về lĩnh vực phát hiện âm thanh bất thường. Vì vậy, em quyết định thực hiện đề tài *Nghiên cứu thiết kế các phương pháp phát hiện âm thanh bất thường* trong thời gian thực tập ở SPARC Lab.

1.2 Tổng quan về trí thông minh nhân tạo

1.2.1 Giới thiệu

Trong cuộc sống hiện đại ngày nay, chúng ta hẳn đã nghe nói về trí tuệ nhân tạo (Artificial Intelligence – AI). AI đã được ứng dụng trong hầu hết các lĩnh vực, ví dụ như nhận diện khuôn mặt, hệ thống khuyến nghị, trò chơi máy tính, chẩn đoán bệnh,

v.v. Sau thi thế giới trải qua ba cuộc cách mạng công nghiệp, lần thứ nhất với sự phát minh ra động cơ hơi nước vào cuối thế kỉ XVIII, lần thứ hai là sự ra đời của năng lượng điện vào cuối thế kỉ XIX, và lần thứ ba khi công nghệ thông tin bùng nổ vào những năm 1980, thì hiện tại thế giới đang bước vào cuộc cách mạng công nghiệp lần thứ tư khi AI len lỏi vào đời sống chúng ta.

Ý tưởng về AI đã có từ đầu thế kỉ XX, nhưng phải đến năm 1956, tại hội nghị Dartmouth được tổ chức bởi John McCarthy tại Hanover, New Hampshire, Hoa Kỳ, thuật ngữ AI mới lần đầu được nhắc đến [1], từ đó mới được sử dụng rộng rãi. Một trong nền tảng của AI là Thuật toán học Perceptron (Perceptron Learning Algorithm – PAL) giúp giải quyết các bài toán phân loại nhị phân. Tuy nhiên, vào năm 1969, Marvin Minsky và Seymour Papert trong cuốn sách *Perceptrons* [2] đã chứng minh rằng không thể sử dụng PLA nếu dữ liệu không tách biệt tuyến tính. Phát hiện này khiến các nghiên cứu về AI bị gián đoạn gần 20 năm. Mãi đến năm 1986, Geoffrey Hinton mới chứng minh một mạng Perceptron đa tầng (Multi-layer Perceptron – MLP) có thể được huấn luyện dựa trên thuật toán lan truyền ngược (Backpropagation) [3]. Tuy nhiên, việc nghiên cứu AI vẫn còn nhiều khó khăn vì khả năng tính toán của máy tính thời ấy còn hạn chế. Từ đó đến khoảng cuối thập niên 2000, AI vẫn phát triển nhưng chưa thu hút được nhiều sự chú ý cũng như ứng dụng nhiều trong đời sống. Phải đến khoảng thời gian 10 năm trở lại đây, khi phần cứng máy tính có sự đột phá về khả năng xử lí và lưu trữ dữ liệu, thì AI mới thực sự bùng nổ và có mặt tại mọi khía cạnh đời sống.

1.2.2 Học máy

Học máy (Machine Learning – ML) là một cách tiếp cận để thực hiện AI. Theo định nghĩa của Arthur L. Samuel vào năm 1959, ML là một lĩnh vực con của khoa học máy tính, làm cho máy tính có khả năng học mà không cần lập trình cụ thể [4]. Có thể hiểu đơn giản rằng ML về cơ bản là sử dụng thuật toán để phân tích dữ liệu, học từ nó và sau đó đưa ra quyết định hoặc dự đoán, thay vì các phần mềm được lập trình với các lệnh cụ thể để hoàn thành một nhiệm vụ cụ thể.

Có một khái niệm nữa cần đề cập, đó là sự học của chương trình máy tính (Computer Program’s Learning). Theo định nghĩa của Tom M. Mitchell vào năm 1977, một máy tính được gọi là học từ kinh nghiệm E để giải quyết nhiệm vụ T với hiệu quả

được đo bằng phép đánh giá P, nếu hiệu quả thực hiện T, được đo bởi P, cải thiện theo kinh nghiệm E [5]. Các thuật ngữ này sẽ được giải thích sau đây.

- Nhiệm vụ T

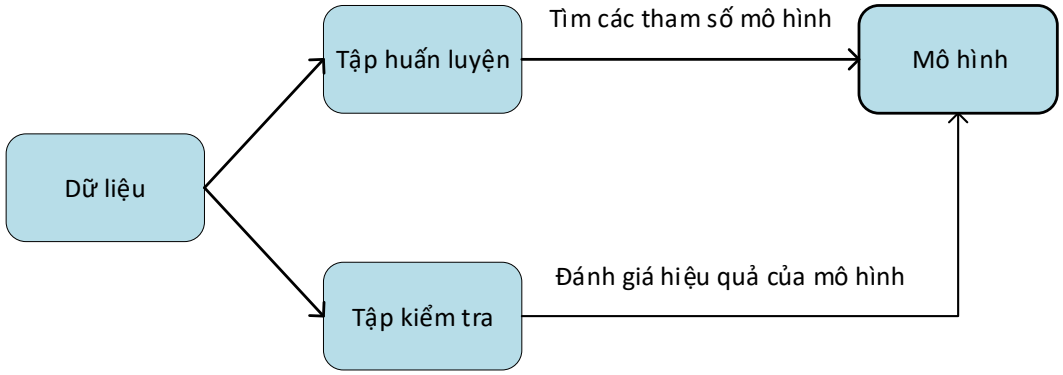
Các nhiệm vụ trong ML thường được mô tả như cách một hệ thống xử lý các điểm dữ liệu đầu vào như thế nào. Bảng 1.1 liệt kê một số bài toán phổ biến trong ML.

Bảng 1.1 Các bài toán thường gặp trong ML

Bài toán	Mục đích	Ví dụ
Phân loại (Classification)	Chỉ ra nhãn của một điểm dữ liệu. Nhãn được chia thành các lớp	Nhận dạng chữ viết tay
Hồi quy (Regression)	Bài toán trong đó nhãn là số thực	Ứng dụng dự đoán độ tuổi dựa trên ảnh chụp khuôn mặt
Phân cụm (Clustering)	Chia dữ liệu thành các nhóm dựa trên sự tương quan của chúng	Phân loại khách hàng dựa trên hành vi mua sắm
Hoàn thiện (Completion)	Hoàn thiện dữ liệu với các giá trị còn thiếu	Hệ thống khuyến nghị
Một số bài toán khác: xếp hạng (Ranking), giảm nhiễu (Denoising), khôi phục thông tin (Information retrieval), v.v		

- Phép đánh giá P

P được sử dụng để đánh giá độ hiệu quả của thuật toán ML. Hình 1.1 mô tả cách đánh giá hiệu quả mô hình.



Hình 1.1 Đo lường độ hiệu quả trong ML

Bộ dữ liệu được chia thành hai tập, tập huấn luyện (Training Set) và tập kiểm tra (Test Set). Tập huấn luyện được sử dụng để tìm các tham số để xây dựng mô hình và hiệu quả của mô hình sẽ được đánh giá bằng tập kiểm tra.

- Kinh nghiệm E

Các mô hình huấn luyện ML có thể được xem như là được cung cấp kinh nghiệm từ tập dữ liệu. Dựa trên các tính chất của tập dữ liệu, các thuật toán ML được chia thành hai nhóm: học có giám sát (Supervised Learning) và học không giám sát (Unsupervised Learning). Thuật toán học có giám sát dự đoán một hoặc nhiều điểm dữ liệu dựa trên các cặp (đầu vào, đầu ra) đã biết. Nhiệm vụ của ta là tìm một hàm xấp xỉ từng phần từ đầu vào với phần tử đầu ra tương ứng. Đối với thuật toán học không giám sát, chỉ có các dữ liệu đầu vào được sử dụng và hệ thống phải dựa vào cấu trúc dữ liệu để thực hiện nhiệm vụ. Trong thuật toán náy, ta không thể biết chính xác đầu ra của dữ liệu đầu vào là gì.

1.2.3 Học sâu

Học sâu (Deep Learning – DL) là một kĩ thuật để triển khai ML. Ban đầu, một phương pháp khác được sử dụng là mạng Neural nhân tạo (Artificial Neural Network – ANN). ANN được truyền cảm hứng từ bộ não con người mà ở đó có các kết nối giữa các tế bào thần kinh neutrons.

Năm 2012, Andrew Ng, một nhà khoa học máy tính và thống kê làm việc tại Google, đã có một bước đột phá trong công nghệ AI. Ông sử dụng ANN, tăng số lượng các lớp và tế bào thần kinh lên nhiều lần, sau đó chạy một lượng lớn dữ liệu thông qua hệ thống để huấn luyện nó, ở đây là hình ảnh từ 10 triệu video YouTube. Sau đó, ông gọi nó là học sâu (Deep learning) mô tả tất cả các lớp trong các ANN này. Từ đây, thuật ngữ DL đã ra đời. Lấy một ví dụ của học sâu, chương trình chơi cờ vây Google AlphaGo đã học cách chơi và được huấn luyện để ngày một chơi tốt hơn, nó điều chỉnh mạng lưới thần kinh của mình bằng cách chơi với chính nó một cách lặp đi lặp lại hàng triệu ván cờ [6].

DL đã tạo điều kiện cho nhiều ứng dụng thực tế của ML và từ đó mở rộng lĩnh vực tổng thể của AI. Xe không người lái, chăm sóc y tế dự phòng tốt hơn, thậm chí các đề xuất phim hay hơn là một số ứng dụng quan trọng của Học sâu trong cuộc sống.

1.3 Bài toán phát hiện âm thanh bất thường

Trong một vài năm trở lại đây, bài toán phát hiện âm thanh bất thường (Abnormal Sound Detection – ASD) đã nhận được nhiều sự chú ý, tuy nhiên vẫn chưa thực sự phổ biến. Việc phát hiện âm thanh bất thường có thể giúp phát hiện các mối nguy hiểm tiềm ẩn và từ đó có biện pháp xử lí kịp thời. Có thể kể ra một số ví dụ ứng dụng thực

tế của ASD như: giám sát trong chăn nuôi dựa vào âm thanh động vật [7], [8], giám sát hoạt động công nghiệp dựa vào âm thanh của máy móc trong nhà máy [9], [10], giám sát đường phố dựa vào âm thanh ngoài đường [11], [12], [13], [14].

Bài toán ASD có thể chia thành hai loại: học có giám sát và học không giám sát. Phương pháp học có giám sát sử dụng các nhãn dữ liệu được gán bằng tay, xác định âm thanh bất thường ở chỗ nào, bao gồm các bài toán phân loại ngữ cảnh âm thanh (Acoustic Scene Classification – ASC) [15], [16], và phát hiện sự kiện âm thanh (Acoustic Event Detection – AED) [17], [18], [19]. Nhiệm vụ của ASC là phân loại các đoạn âm thanh thành từng loại ngữ cảnh, trong khi AED xác định thời điểm bắt đầu và kết thúc của các sự kiện âm thanh. Các phương pháp này có khả năng nhận diện nhiều dạng âm thanh khác nhau, nhưng chúng đòi hỏi phải có nhãn dữ liệu. Ngược lại, các phương pháp học không giám sát không yêu cầu có nhãn trước, do đó được ưa chuộng hơn. Một phương pháp học không giám sát là phát hiện điểm thay đổi (Change Point Detection) [20], [21], [22], được thực hiện bằng cách so sánh mô hình tại thời điểm hiện tại với thời điểm trước đó để tính toán sự tương đồng giữa hai thời điểm, nếu xảy ra sự không tương đồng thì tại đó xác định điểm bất thường. Tuy nhiên, đối với không gian công cộng, các âm thanh có thể xảy ra với độ biến động cao và không có định, do đó các điểm thay đổi có thể không phải bất thường (ví dụ: tiếng tàu hỏa). Một phương pháp học không giám sát khác là phát hiện ngoại lệ (Outlier Detection) [23], [24], [25], thực hiện mô hình hóa các mẫu âm thanh bình thường của một môi trường, sau đó phát hiện các âm thanh không tương thích với mô hình và xác định đó là bất thường. Mô hình hỗn hợp Gaussian (Gaussian Mixture Model – GMM) [26], máy vector hỗ trợ (Support Vector Machine – SVM) [27] là các mô hình điển hình cho phương pháp này.

Với những tiến bộ gần đây của học sâu, các phương pháp dựa trên mạng neural đã được nghiên cứu để phát hiện âm thanh bất thường [28], [29]. Tư tưởng chính của những phương pháp này là huấn luyện một Bộ mã hóa tự động (Autoencoder – AE) chỉ sử dụng dữ liệu âm thanh bình thường. AE sẽ mã hóa dữ liệu đầu vào vào không gian ẩn, sau đó giải mã để khôi phục dữ liệu ban đầu. Sau đó mô hình AE đã huấn luyện được sử dụng để phát hiện bất thường bằng các tính toán Lỗi tái tạo (Reconstruction Error) giữa dữ liệu đầu vào và đầu ra của AE, nếu lỗi này cao thì tại đó được xác định là bất thường. Bên cạnh AE, cũng có thể sử dụng mạng Neural

truyền thẳng (Feedforward Neural Network – FNN), mạng Neural tích chập (Convolutional Neural Network – CNN), mạng Neural tích chập truy hồi (Convolutional Recurrent Neural Network – CRNN) để phát hiện bất thường, tuy nhiên những mô hình này có hạn chế là phải có nhãn các điểm bất thường để huấn luyện, do đó ít khi được sử dụng.

Trong báo cáo này, mô hình AE truyền thống và một cải tiến của nó, được gọi là AE biến thể (Variational Autoencoder – VAE) được sử dụng để giải quyết bài toán phát hiện âm thanh bất thường, sau đó so sánh kết quả thực hiện của chúng với nhau và rút ra nhận xét.

1.4 Kết luận chương

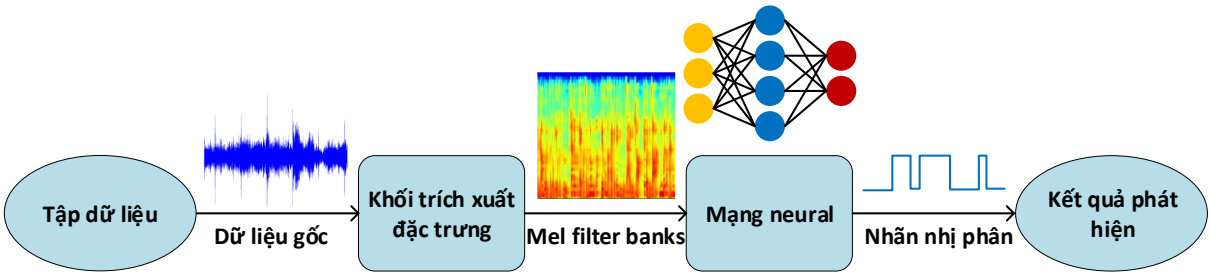
Chương 1 đã giới thiệu về đề tài phát hiện âm thanh bất thường, trình bày tổng quan nhất về AI, ML và DL, đây là những nền tảng cơ bản nhất để có thể phát triển và thực hiện đề tài lần này. Ngoài ra, chương này cũng đã trình bày sơ lược về các phương pháp phát hiện âm thanh bất thường và đặc điểm của từng phương pháp

CHƯƠNG 2. THIẾT KẾ MÔ HÌNH HỆ THỐNG

Chương 2 trình bày về mô hình hệ thống phát hiện âm thanh bất thường, lí thuyết về xử lí tín hiệu, cách trích xuất đặc trưng của dữ liệu âm thanh cũng như giới thiệu những kiến thức cơ bản về mạng Neural nhân tạo.

2.1 Tổng quan hệ thống

Hình 2.1 thể hiện tổng quan về hệ thống sử dụng để phát hiện âm thanh bất thường.



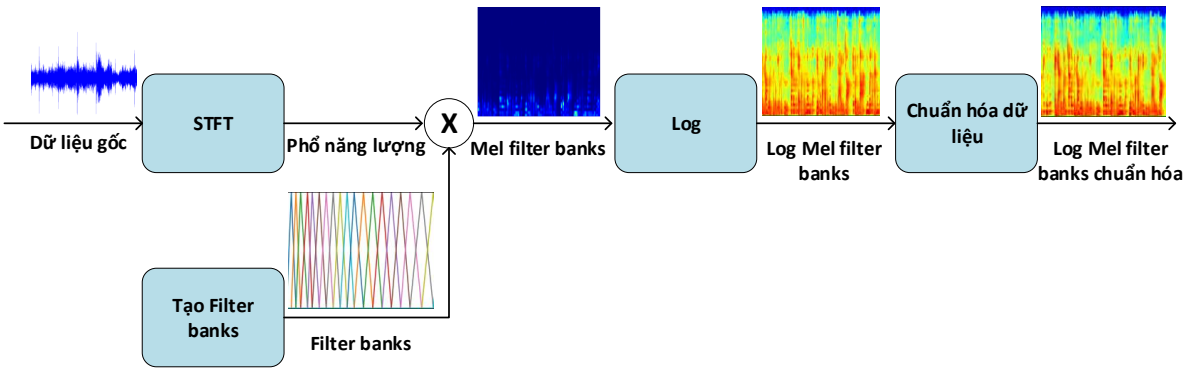
Hình 2.1 Tổng quan hệ thống

Ban đầu, dữ liệu âm thanh được lấy ra từ tập dữ liệu và đưa vào khối trích xuất đặc trưng. Sau đó, đặc trưng được trích xuất của dữ liệu âm thanh, ở đây là Mel filter banks, được đưa vào một mạng neural để huấn luyện. Cuối cùng, kết quả phát hiện được thể hiện dưới dạng mã nhị phân, với giá trị 1 thể hiện tại đó là âm thanh bất thường và ngược lại, 0 là âm thanh bình thường.

2.2 Khối trích xuất đặc trưng

2.2.1 Tổng quan khối trích xuất đặc trưng

Hình 2.2 thể hiện tổng quan về khối trích xuất đặc trưng



Hình 2.2 Khối trích xuất đặc trưng

Dữ liệu gốc ban đầu được đưa vào khối STFT để tính toán biến đổi Fourier thời gian ngắn (Short-time Fourier Transform – STFT).). STFT được sử dụng để trích xuất phổ công suất của tín hiệu đầu vào. Tín hiệu trong miền thời gian thể hiện về biên độ, nhưng không cho thấy các thành phần tần số. Ngược lại, tín hiệu trong miền tần số cho các thành phần tần số nhưng không biểu thị thứ tự các tần số đó. STFT thể hiện được cả các thành phần tần số và thứ tự xuất hiện. Tiếp theo, khối tạo Filter banks tạo ra các bộ lọc gọi là Filter banks, là các bộ lọc tam giác được sử dụng để bắt chước hoạt động tai người. Tiêp tục thực hiện nhân các bộ lọc này với phổ năng lượng của tín hiệu để thu được các đặc trưng Mel filter banks. Cuối cùng, lấy logarit của Mel filter banks, ta thu được Log Mel filter banks.

2.2.2 Biến đổi STFT

Biến đổi Fourier thời gian ngắn (Short-time Fourier transform – STFT) là một chuỗi các biến đổi Fourier của tín hiệu được cửa sổ hóa hay còn gọi là windowing. STFT cung cấp thông tin của tần số được định vị theo thời gian cho các tình huống trong đó các thành phần tần số của tín hiệu thay đổi theo thời gian [30].

Công thức cho STFT thời gian rời rạc:

$$\mathbf{STFT}\{\mathbf{x}[\mathbf{n}]\}(\mathbf{m},\mathbf{w}) = \sum_{\mathbf{n}=-\infty}^{\infty} \mathbf{x}[\mathbf{n}]\mathbf{w}[\mathbf{n}-\mathbf{m}]\mathbf{e}^{-\mathbf{jwn}} \qquad (2.1)$$

Trong đó:

x[n]: Tín hiệu rời rạc trong miền thời gian

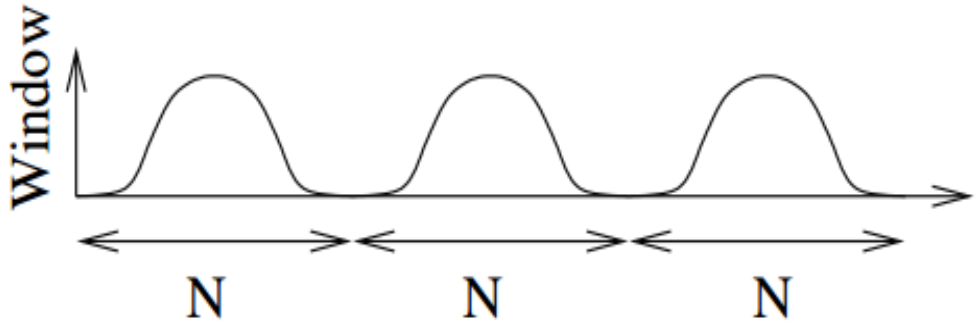
w: Loại cửa sổ sử dụng

m: Chỉ số của cửa sổ

2.2.2.1 Windowing

Bước này sẽ phân chia tín hiệu thành thành các khung sử dụng cửa số Hamming thay vì sử dụng cửa sổ hình chữ nhật, vì sử dụng cửa sổ hình chữ nhật khiến tín hiệu bị sai trong miền tần số. Trong đề tài này, cửa sổ Hamming được sử dụng để phân chia tín hiệu, cửa sổ Hamming có công thức:

$$W_{ham}(n) = \begin{cases} 0,54 - 0,46\cos\left(\frac{2\pi n}{N-1}\right), & 0 < n < N-1 \\ 0 & , \quad n \text{ còn lại} \end{cases} \quad (2.2)$$

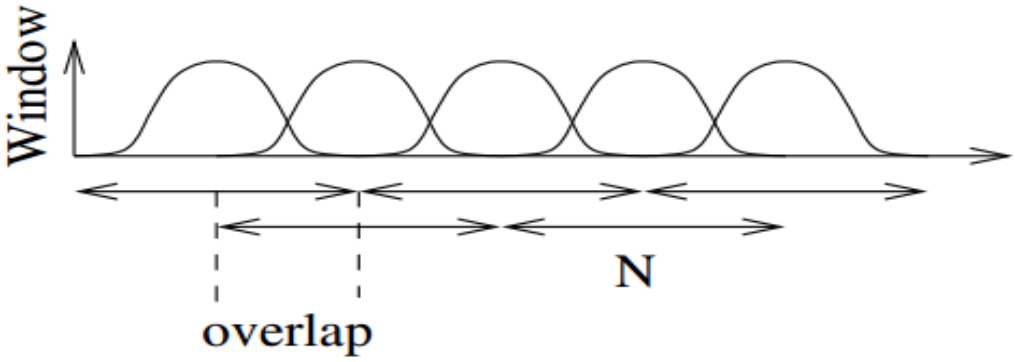


Hình 2.3 Cửa sổ và kích thước cửa sổ [30]

Dựa vào Hình 2.3, chúng ta có thể thầy thấy cường độ tín hiệu giảm xuống 0 ở cả hai đầu của mỗi cửa sổ. Nó có nghĩa là tín hiệu bị suy giảm biên độ trong miền thời gian. Vì vậy overlap là một trong những giải pháp tốt để giải quyết vấn đề này.

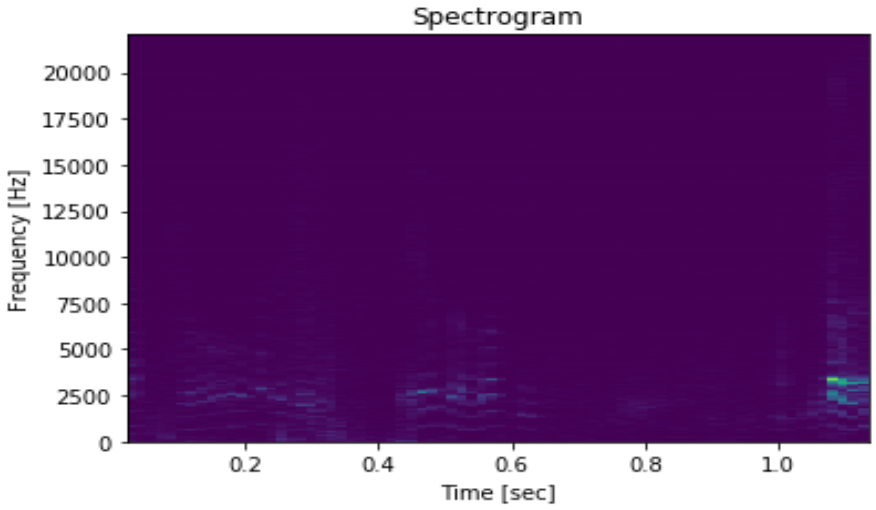
2.2.2.2 Overlap

Chồng chéo hay overlap là một kỹ thuật xếp chồng các cửa sổ lên nhau với tỷ lệ nhất định như là 20%, 30%, 50% để tránh suy giảm tín hiệu ở cả hai đầu của mỗi cửa sổ. Hình 2.4 đây minh họa cửa sổ với 50% chồng lên nhau.



Hình 2.4 Các cửa số với 50% overlap [30]

Để minh họa STFT, một biểu đồ gọi là spectrogram được sử dụng. Trên hình 2.5 là spectrogram của phổ công suất của tín hiệu rời rạc.



Hình 2.5 Phổ của tín hiệu rời rạc [30]

Công thức phổ công suất của tín hiệu rời rạc là:

$$\mathbf{P} = \frac{|\mathbf{FFT}(\mathbf{x}[\mathbf{n}])|^2}{\mathbf{NFFT}} \tag{2.3}$$

Trong đó:

P: phổ công suất

x: tín hiệu rời rạc

NFFT: chiều dài cửa sổ

2.2.3 Mel filter bank

Trong các nghiên cứu về tiếp nhận âm thanh của con người, người ta thấy rằng tai người với mỗi tần số hấp thụ là khác nhau. Do đó, để nâng cao chất lượng cho các hệ thống phát hiện âm thanh, các bộ lọc Filter banks được tạo ra để bắt chước hoạt động của tai người [31]. Để mô hình hóa Mel filter banks, khái niệm về thang đo Mel được đưa ra. Thang đo Mel có ảnh hưởng của phi tuyến trong miền tần số của tín hiệu, mô phỏng hiệu ứng của tai người với từng tần số.

Công thức chuyển đổi từ thang đo tần số f (Hz) sang thang đo m (Mel) như sau:

$$\mathbf{m} = \mathbf{2595}\,\mathbf{log}\left(\mathbf{1} + \frac{\mathbf{f}}{\mathbf{700}}\right) \tag{2.4}$$

Để chuyển đổi ngược lại từ thang đo Mel sang thang đo Hz, ta dùng công thức:

$$\mathbf{f = 700\left(10^{\frac{m}{2595}} - 1\right)} \tag{2.5}$$

Mel filter banks bao gồm các bộ bộ lọc tam giác, được thể hiện dưới dạng:

$$H_m(k) = \begin{cases} 0, & k<f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)}, & f(m-1) \le k < f(m) \\ 1, & k = f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)}, & f(m) \le k < f(m+1) \\ 0, & k > f(m+1) \end{cases} \tag{2.6}$$

Trong đó:

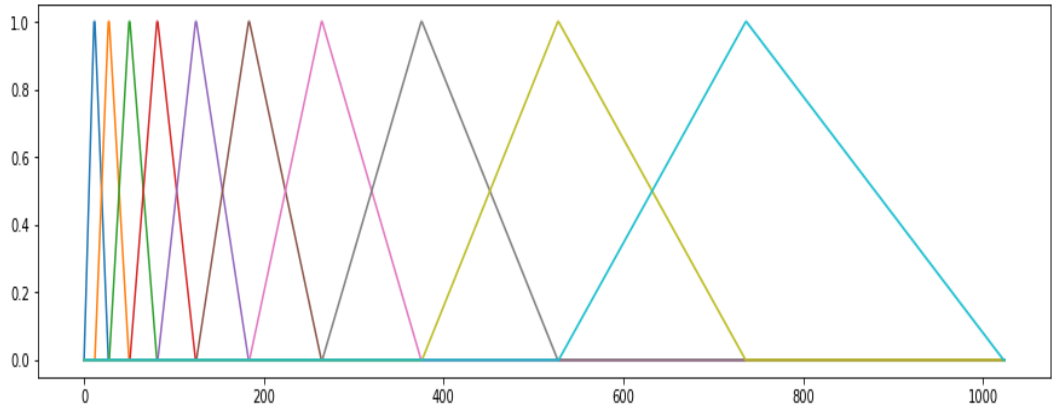
m: Chỉ số bộ lọc tam giác

k: giá trị thứ k của bộ lọc tam giác

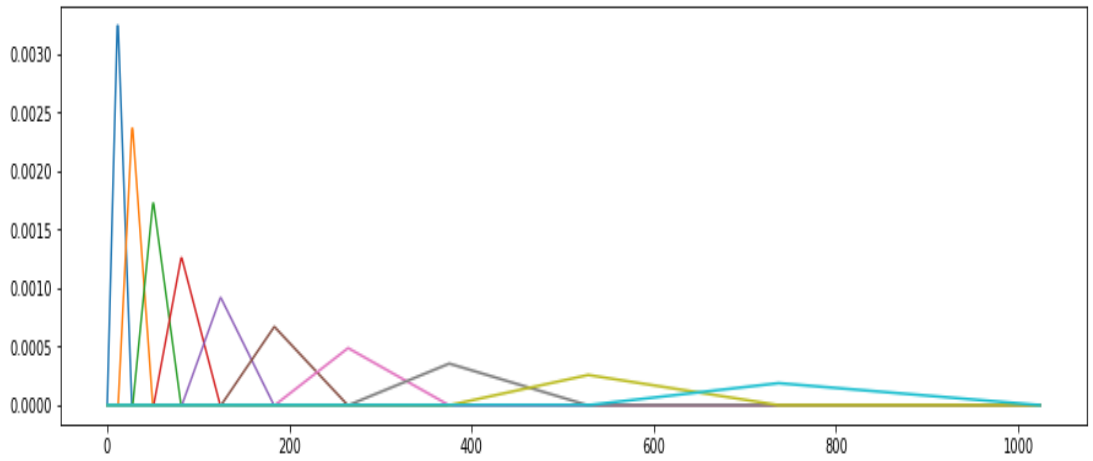
H_m (k): Bộ lọc tam giác

f (m): Hàm chuyển đổi từ miền thang đo Mel sang miền tần số

Hình 2.6 minh họa các filter bank Mel với NFFT = 2048 và số lượng bộ lọc tam giác là 10, mỗi bộ lọc được thể hiện bằng một màu. Tuy nhiên, nếu chúng ta muốn sử dụng các bộ lọc này một cách hiệu quả, chúng ta cần chuẩn hóa các bộ lọc tam giác vào cùng một khu vực (năng lượng), để tránh tăng giá trị của nhiễu trong bộ lọc năng lượng cao. Để làm điều này, cần chia chiều cao của chúng cho chiều dài tương ứng (Hình 2.7).

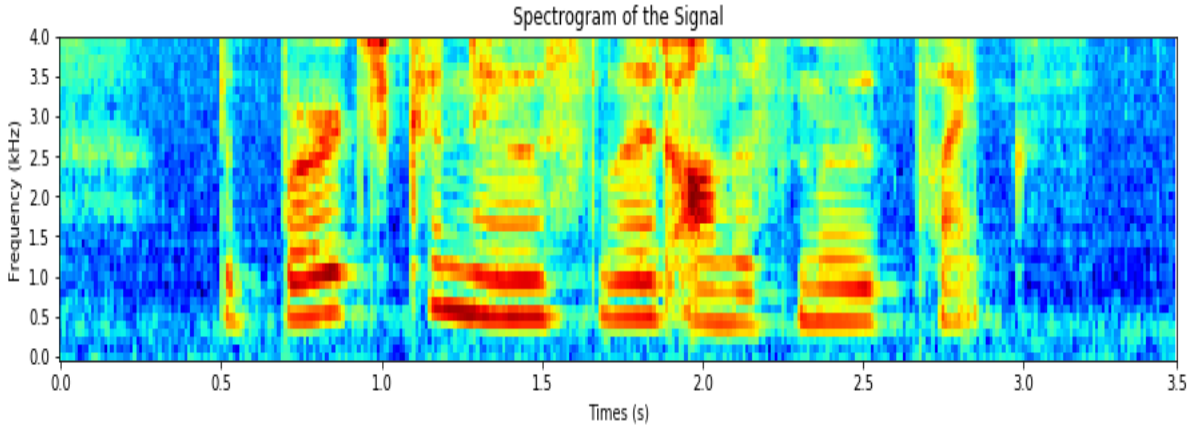


Hình 2.6 Các filter banks với NFFT = 2048 và nframe = 10



Hình 2.7 Các filter banks được chuẩn hóa

Sau khi nhân các bộ lọc này với phổ công suấtcủa tín hiệu và lấy logarithm của chúng, ta thu được Log Mel filter banks như trên hình 2.8.



Hình 2.8 Các đặc trưng Log Mel filter banks

2.2.4 Chuẩn hóa dữ liệu

Các điểm dữ liệu đôi khi được đo đạc với những đơn vị khác nhau, chẳng hạn như m và feet, hoặc có hai thành phần của vector dữ liệu chênh lệch nhau quá lớn, chẳng hạn một thành phần có khoảng giá trị từ 0 đến 1000, trong khi thành phần kia chỉ có khoảng giá trị từ 0 đến 1. Lúc này, chúng ta cần chuẩn hóa dữ liệu trước khi thực hiện các bước tiếp theo. Dưới đây là một vài phương pháp chuẩn hóa thường được sử dụng [32].

2.2.4.1 Rescaling (Min-max Normalization):

Rescaling, hay còn gọi là chuẩn hóa min-max, là phương pháp đơn giản nhất với việc đưa tất cả các thành phần dữ liệu về cùng một khoảng là [0,1], công thức sẽ là:

$$\mathbf{x}' = \frac{\mathbf{x} - \mathbf{min}(\mathbf{x})}{\mathbf{max}(\mathbf{x}) - \mathbf{min}(\mathbf{x})} \quad (2.7)$$

Trong đó:

x: đặc trưng ban đầu

x': đặc trưng sau khi chuẩn hóa

min(x), max(x) được tính trên toàn bộ dữ liệu huấn luyện ở cùng một thành phần vector đặc trưng x.

2.2.4.2 Standardization:

Standardization là một phương pháp cũng hay được sử dụng với việc giả sử mỗi thành phần đều tuân theo phân phối chuẩn (phân phối Gauss) với kỳ vọng μ là 0 và phương sai σ^2 là 1. Khi đó, công thức sẽ là:

$$\mathbf{x} = \frac{\mathbf{x} - \mu}{\boldsymbol{\sigma}} \quad (2.8)$$

Trong báo cáo này, phương pháp rescaling được sử dụng để chuẩn hóa dữ liệu.

2.3 Tổng quan về mạng neural

2.3.1 Giới thiệu mạng neural nhân tạo

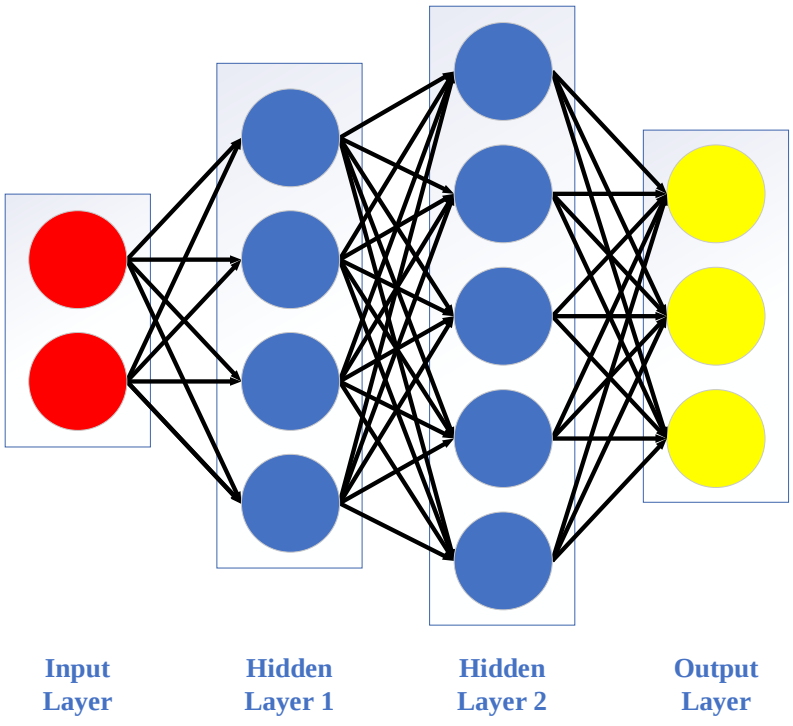
Thuật ngữ neuron thần kinh có nguồn gốc từ các tế bào hệ thống thần kinh của con người (động vật). Về cơ bản, mạng lưới thần kinh sinh học, là một mạng lưới liên kết của hàng tỷ tế bào thần kinh với hàng nghìn tỷ kết nối giữa chúng. Mạng neural nhân tạo (Artificial neural networks – ANN) là sự mô phỏng được lấy cảm hứng từ neuron sinh học và được thực hiện trên máy tính để thực hiện một số tác vụ cụ thể như phân cụm, phân loại, nhận dạng mẫu, v.v.

ANN hoạt động giống với bộ não của con người. ANN thu nhận kiến thức thông qua học tập, các kiến thức của ANN được lưu trữ trong các kết nối giữa neurons mà được gọi là trọng số.

2.3.2 Mô hình mạng neuron nhân tạo cơ bản

Hình 2.9 thể hiện mô hình cơ bản của một mạng neuron thông thường. Trong đó, mỗi cột hình chữ nhật được gọi là các layer, cột đầu tiên là input layer, các cột ở giữa được gọi là các hidden layer và cột cuối cùng là output layer. Các hình tròn trong mỗi cột đó được gọi là các nodes hay các units.

- Input layer: Input layer bao gồm các node đầu vào và đưa dữ liệu ban đầu vào hệ thống để xử lý thêm bởi các lớp tiếp theo.
- Hidden layer: Mỗi hidden layer được gọi là *fully connected layer*, tên gọi theo đúng ý nghĩa, mỗi node trong hidden layer được kết nối với tất cả các node trong layer trước. Hidden layer là một lớp ở giữa các lớp đầu vào và các lớp đầu ra, trong đó các node trong lớp này sẽ lấy một tập hợp các đầu vào có trọng số và tạo ra một đầu ra thông qua một hàm kích hoạt hay tên tiếng anh là *activation function*.
- Output layer: Output layer là lớp neuron nhân tạo cuối cùng có tác dụng tạo ra các đầu ra cho mô hình.

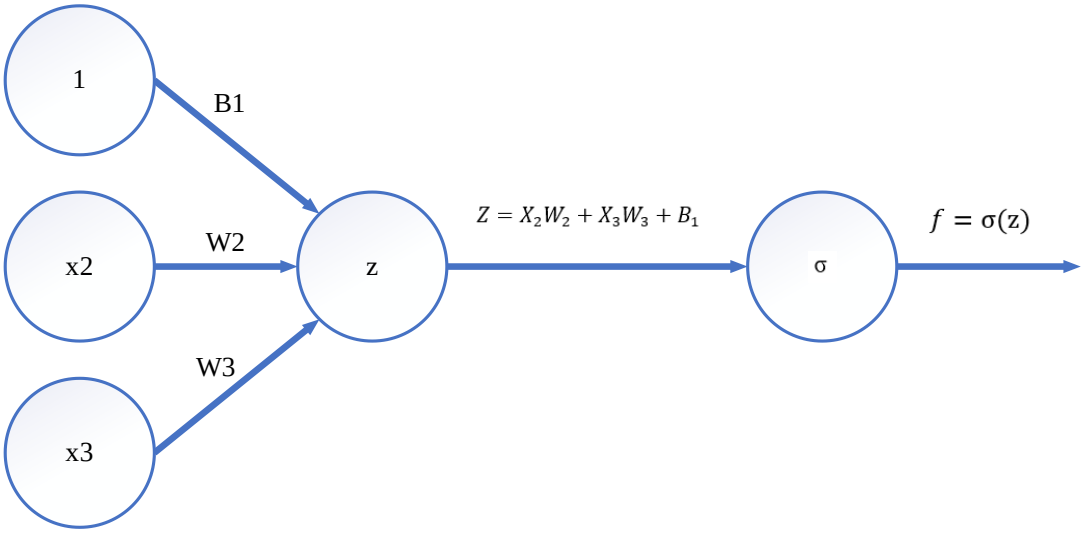


Hình 2.9. Một mô hình neuron network cơ bản

Trong mỗi mô hình mạng neuron luôn chỉ có 1 input layer và 1 output layer, số hidden layer ở giữa có thể thay đổi số lượng tùy vào người thiết kế, thậm chí trong một số trường hợp có thể không có lớp hidden layer. Tổng số layer trong một mô hình sẽ được tính bằng tổng số layer của hidden layer và output layer. Ví dụ nếu như một mô hình mạng có 1 input layer, 5 hidden layer và 1 output layer thì ta có thể nói số layer trong mô hình này là 6 layer.

2.3.3 Node

Mạng neuron nhân tạo có thể được xem như các đồ thị có hướng có trọng số w trong đó các neuron nhân tạo là các nút và các cạnh có hướng có trọng số w là các kết nối giữa đầu ra neuron và đầu vào neuron (Hình 2.10).



Hình 2.10. Node trong các hidden layer và output layer

Trong Hình 2.10, đầu ra của các node 2, 3 tương ứng với x_2 và x_3 với trong trọng số tương ứng là W_2 và W_3. Ngoài ra, node đầu tiêng giá trị luôn cố định bằng 1 thì trọng số ứng với nó giờ đây có tên gọi là bias hay hệ số tự do, ký hiệu là B1. Việc mỗi node thêm một bias sẽ làm tăng tính tổng quát hóa quả phương trình. Ví dụ như các bài toán tìm đường thẳng đi qua các điểm có tọa độ là (1,3) và (2,16) với phương trình bậc 1 là:

$$\mathbf{y = ax + b} \qquad (2.9)$$

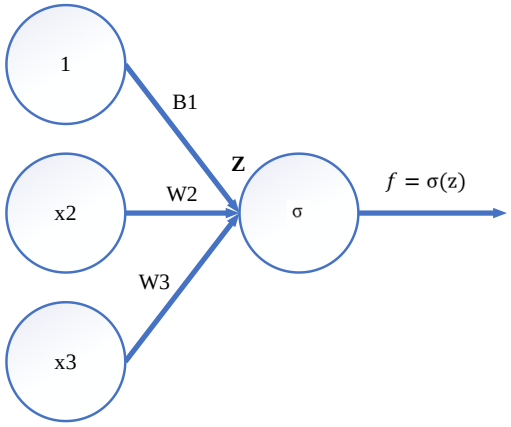
Trong đó:

a: trọng số

b: hệ số tự do

Giả sử nếu như không có hệ số tự do b thì đường thẳng luôn đi qua điểm gốc tọa độ vì vậy việc tìm đường thẳng phù hợp với điều kiện đề bài là rất khó khăn và nó cũng mất đi tính tổng quát của phương trình bậc 1. Quay lại với mô hình, với việc thêm bias vào các node sẽ làm tăng tính tổng quát của phương trình z, huấn luyện mô hình sẽ dễ dàng hơn trong việc tính toán và tìm ra các trọng số w mà bài toán mong muốn.

Mỗi node đều thực hiện 2 bước tính toán đầu tiên là tính tổng *linear* và tiếp tục áp dụng *activation function*. Hình 2.11 là mô hình gộp giữa hai bước tình toán của Hình 2.12 và được gọi là mô hình *logistic regression*. Logistic regression là mô hình neural network đơn giản nhất chỉ với input layer và output layer.



Hình 2.11. Mô hình logistic regression

Công thức của hai bước sẽ được tóm gọn lại như sau:

- Tính tổng linear:

$$\mathbf{Z} = \sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{X_j W_j} + \mathbf{B} \qquad (2.10)$$

Trong đó:

X_j: đầu ra của mỗi node thuộc lớp layer trước

W_j: trọng số tương ứng với mỗi node

B là bias của layer trước

n: số lượng node của layer trước

Z là đầu ra của bước 1

- Áp dụng activation function:

$$\mathbf{X_{out}} = \boldsymbol{\sigma}(\mathbf{Z}) \quad (2.11)$$

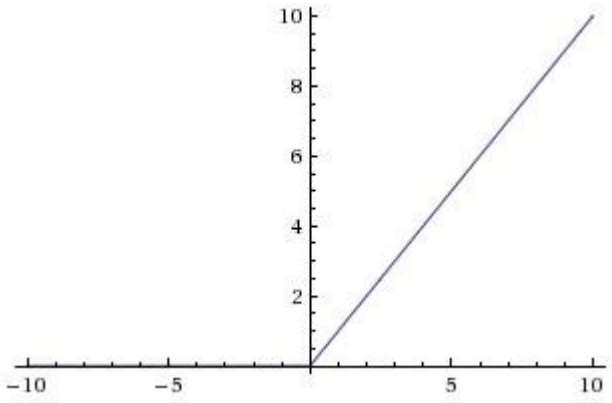
Trong đó:

X_{out}: đầu ra của node.

σ: hàm activation function.

2.3.4 Activation function

Trong ANN, hàm activation function có rất nhiều và rất đa dạng, chúng được sử dụng tùy thuộc vào đặc điểm của dữ liệu đầu vào mà chọn một hàm phù hợp. Trong đề tài này, dữ liệu là giọng nói của con người, là một tín hiệu liên tục và phi tuyến tính, vì vậy, hàm activation là hàm ReLU và hàm Sigmoid được sử dụng (Hình 2.12 và Hình 2.13).



Hình 2.12. Đồ thị hàm ReLU [32]

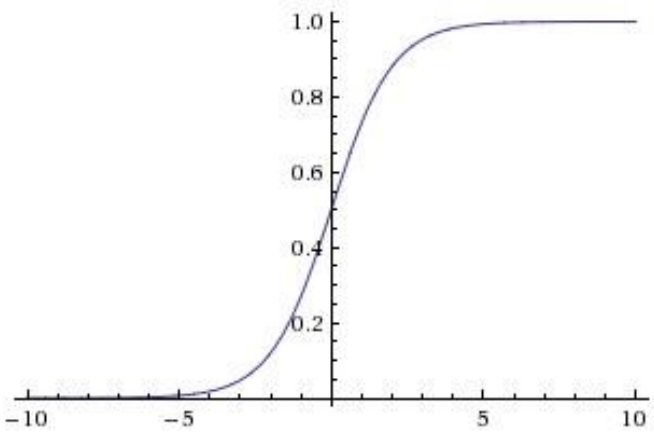
ReLU (Rectified Linear Unit) được sử dụng rộng rãi gần đây vì tính đơn giản của nó. Đồ thị của hàm ReLU được minh họa trên Hình 2.10. Nó có công thức toán học:

$$\mathbf{f}(\mathbf{s}) = \mathbf{max}(\mathbf{0}, \mathbf{s}) \quad (2.12)$$

Ưu điểm chính của nó là:

- ReLU được chứng minh giúp cho việc training các Deep Networks nhanh hơn rất nhiều (theo Krizhevsky et al.). ReLU được tính toán gần như tức thời và gradient của nó cũng được tính cực nhanh với gradient bằng 1 nếu đầu vào lớn hơn 0, bằng 0 nếu đầu vào nhỏ hơn 0.

- Mặc dù hàm ReLU không có đạo hàm tại s = 0, trong thực nghiệm, người ta vẫn thường định nghĩa ReLU′ = 0 và khẳng định thêm rằng, xác suất để input của một unit bằng 0 là rất nhỏ.



Hình 2.13. Đồ thị hàm sigmoid [32]

Hàm sigmoid có phương trình:

$$\mathbf{f(s) = \frac{1}{1 + e^{-s}}} \qquad (2.13)$$

Hàm sigmoid có đồ thị như trong Hình 2.13. Nếu đầu vào lớn, hàm số sẽ cho đầu ra gần với 1. Với đầu vào nhỏ (rất âm), hàm số sẽ cho đầu ra gần với 0. Hàm số này được sử dụng nhiều trong quá khứ nhưng gần đây ít khi được sử dụng.

2.3.5 Hàm mất mát

Mối quan hệ giữa phép đánh giá và các tham số của mô hình được biểu diễn thông qua một hàm số được gọi là hàm mất mát (loss function). Một phép đánh giá cho kết quả tốt khi hàm mất mát nhỏ, vì vậy nhiệm vụ của ta là phải tối thiểu hàm mất mát.

Gọi θ là tập các tham số mô hình, θ* là bộ tham số của mô hình có hàm mất mát tối ưu, kí hiệu hàm mât mát của mô hình là $\mathcal{L}(\theta)$, ta cần giải bài toán tối ưu sau:

$$\boldsymbol{\theta^* = \mathbf{argmin}\big(\mathcal{L}(\theta)\big)} \qquad (2.14)$$

Hàm số $\mathcal{L}(\theta)$ có thể không có chặn dưới hoặc đạt giá trị nhỏ nhất với nhiều bộ tham số θ khác nhau. Thậm chí, việc tối ưu hàm số này đôi khi không thực hiện được. Trên thực tế, ta chỉ cần tìm được một bộ tham số θ khiến hàm mất mát đạt giá trị nhỏ nhất hoặc cực tiểu cũng mang lại kết quả tốt.

CHƯƠNG 3. CÁC MÔ HÌNH MẠNG NEURAL ĐƯỢC SỬ DỤNG

Chương này trình bày về các mô hình mạng neural trong bài toán phát hiện âm thanh bất thường, bao gồm: Bộ mã hóa tự động (Autoencoder – AE) và Bộ mã hóa tự động biến thể (Variational Autoencoder – VAE).

3.1 Giới thiệu

Trong những năm gần đây, các mô hình sinh dựa trên học sâu đã phát triển và thu hút được nhiều sự chú ý trong lĩnh vực trí tuệ nhân tạo. Dựa vào lượng dữ liệu khổng lồ hiện có, các kiến trúc mạng và thuật toàn huấn luyện tối ưu, các mô hình sinh đã cho thấy khả năng có thể tái tạo dữ liệu rất giống thật, ví dụ như văn bản, âm thanh, hình ảnh. Trong các mô hình sinh, bộ mã hóa tự động (Autoencoder – AE) là mô hình nổi bật và được chú ý hơn cả.

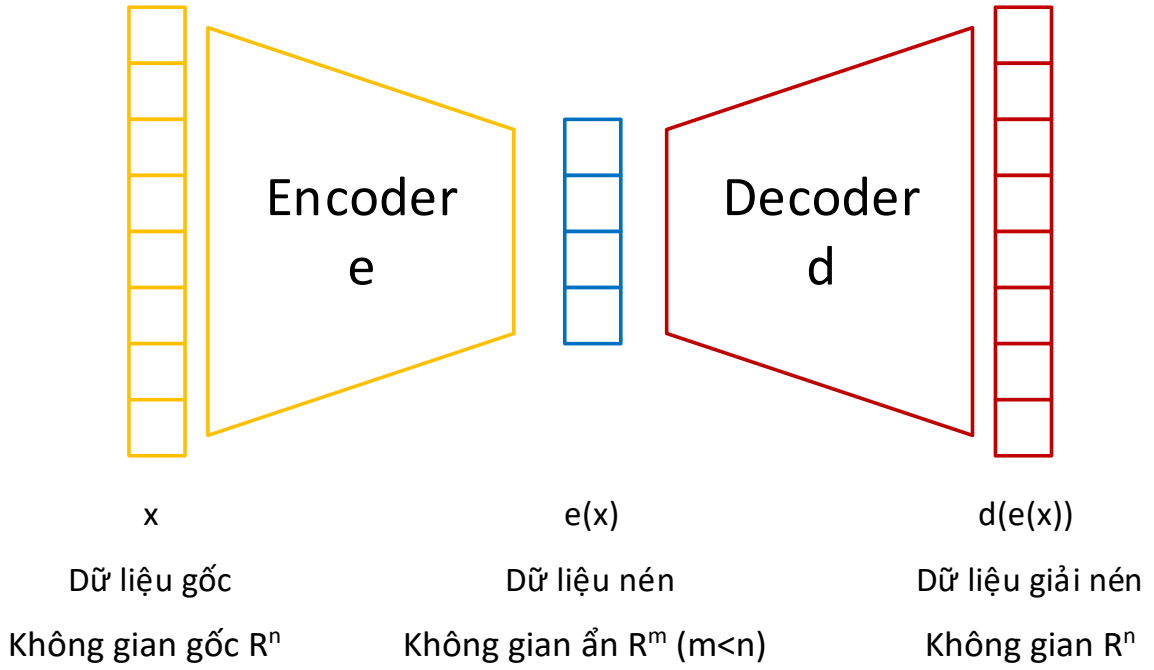
3.2 Giảm chiều dữ liệu

Phần này được viết chủ yếu dựa vào tài liệu tham khảo [33].

Trước khi trình bày sâu về AE, một khái niệm cần tìm hiểu kĩ trước là giảm chiều dữ liệu (Dimensionality Reduction). Theo khái niệm trong học máy, giảm chiều dữ liệu là một tiến trình giảm số lượng đặc trưng của dữ liệu. Số lượng đặc trưng có thể giảm bằng cách lựa chọn những đặc trưng chính hoặc bằng cách trích xuất, nghĩa là tạo ra các đặc trưng mới dựa trên những đặc trưng sẵn có, sao cho số lượng đặc trưng giảm đi. Việc giảm chiều dữ liệu này đặc biệt hữu ích trong những trường hợp mà dữ liệu cần có ít chiều (ví dụ: lưu trữ dữ liệu, tính toán khối lượng lớn, …). Mặc dù có rất nhiều phương pháp khác nhau để giảm chiều dữ liệu, nhưng chúng đều dựa trên một khuôn mẫu, bao gồm bộ mã hóa (Encoder) và bộ giải mã (Decoder).

Một cách tổng quát nhất, bộ mã hóa có nhiệm vụ tạo ra sự biểu diễn của các đặc trưng mới dựa trên sự biểu diễn của các đặc trưng cũ (có thể bằng cách chọn lựa hoặc trích xuất). Ở chiều ngược lại, bộ giải mã tái tạo lại những đặc trưng ban đầu dựa vào các đặc trưng mới được tạo ra bởi bộ mã hóa. Việc giảm chiều dữ liệu có thể được hiểu như việc nén dữ liệu từ không gian ban đầu (initial space) thành không gian mã hóa

(encoded space) hay không gian ẩn (latent space), sau đó giải nén để thu được dữ liệu ban đầu. Việc nén và giải nén dữ liệu như thế này đương nhiên sẽ có mất mát dữ liệu, khi mà một phần thông tin bị mất trong quá trình nén và sẽ không thể được khôi phục khi giải nén. Nguyên lí giảm chiều dữ liệu được minh họa trên hình 3.1.



Hình 3.1 Nguyên lí giảm chiều dữ liệu

Trên hình 3.1, dữ liệu gốc x trong không gian gốc R^n được nén bằng bộ mã hóa encoder e thành dữ liệu e(x) trong không gian ẩn R^m, với m<n vì mục đích giảm chiều dữ liệu. Sau đó, dữ liệu nén được giải nén bằng bộ giải mã decoder d, thu được dữ liệu d(e(x)) trong không gian R^n giống không gian gốc. Trong trường hợp lí tưởng khi x = d(e(x)), dữ liệu thu được giống hệt dữ liệu gốgc vì không bị mất mát thông tin, tuy nhiên trên thực tế sự mất mát thông tin luôn xảy ra, khi đó x ≠ d(e(x)). Công việc cần làm ở đây là tìm ra cặp bộ mã hóa/bộ giải mã sao cho việc mất mát là tối thiểu. Cụ thể là, cần tìm bộ mã hóa có khả năng giữ được tối đa thông tin khi nén, và từ đó mất mát là tối thiểu khi giải nén. Sự mất mát thông tin được xác định bằng Lỗi tái tạo (Reconstruction Error) giữa dữ liệu gốc và dữ liệu thu được sau khi giải nén, kí hiệu là $\epsilon(x, d(e(x)))$. Kí hiệu E và D lần lượt là tập các bộ mã hóa và bộ giải mã, e* và d* lần lượt là bộ mã hóa và bộ giải mã cần tìm, ta có:

$$(\mathbf{e}^*, \mathbf{d}^*) = \underset{(\mathbf{e},\mathbf{d}) \in (\mathbf{E},\mathbf{D})}{\mathbf{argmin}} \; \boldsymbol{\epsilon}\left(\mathbf{x}, \mathbf{d}\left(\mathbf{e}(\mathbf{x})\right)\right) \tag{3.1}$$

3.3 Phương pháp phân tích thành phần chính

3.3.1 Giới thiệu

Một trong những phương pháp giảm chiều dữ liệu đầu tiên áp dụng theo khuôn mẫu trên có tên là "Phân tích thành phần chính" (Principal Component Analysis – PCA). Ý tưởng chính của PCA là xây dựng các đặc trưng độc lập là các tổ hợp tuyến tính của các đặc trưng gốc. Đây có thể coi tương đương như một phép chiếu dữ liệu từ không gian gốc lên một không gian con, sao cho dữ liệu trên không gian con này gần nhất có thể với dữ liệu trên không gian gốc ("gần" ở đây được tính theo khoảng cách Euclid). Nói cách khác, PCA xác định không gian con tuyến tính của không gian gốc, sao cho lỗi do xấp xỉ dữ liệu vì phép chiều là nhỏ nhất có thể.

3.3.2 Cơ sở toán học của PCA

Giả sử ta có vector dữ liệu gốc là x như sau:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \vdots \\ \mathbf{x_n} \end{bmatrix} \in \mathbf{R^n} \tag{3.2}$$

Bộ mã hóa ở đây được biểu diễn dưới dạng một ma trận E_{mxn} mà các hàng là trực giao, nghĩa là:

$$\mathbf{E} = \begin{bmatrix} \mathbf{e_{11}} & \mathbf{e_{12}} & \cdots & \mathbf{e_{1n}} \\ \mathbf{e_{21}} & \mathbf{e_{22}} & \cdots & \mathbf{e_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{e_{m1}} & \mathbf{e_{m2}} & \cdots & \mathbf{e_{mn}} \end{bmatrix} \in \mathbf{R^{mxn}} \tag{3.3}$$

Kí hiệu hàng thứ i của ma trận E là:

$$\mathbf{e_i} = [\mathbf{e_{i1}} \quad \mathbf{e_{i2}} \quad \cdots \quad \mathbf{e_{in}}] \tag{3.4}$$

Ta có tích vô hướng của các vector hàng bằng 0:

$$\mathbf{e_i e_j} = \mathbf{0} \; \forall \; \mathbf{i} \neq \mathbf{j} \tag{3.5}$$

Các hàng của ma trận E trực giao với nhau thể hiện m đặc trưng trong không gian ẩn là độc lập với nhau.

Đặt z là vector dữ liệu sau khi được bộ mã hóa nén. Ta có mối quan hệ sau:

$$\mathbf{z} = \mathbf{Ex} = \begin{bmatrix} \mathbf{e_{11}} & \mathbf{e_{12}} & \cdots & \mathbf{e_{1n}} \\ \mathbf{e_{21}} & \mathbf{e_{22}} & \cdots & \mathbf{e_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{e_{m1}} & \mathbf{e_{m2}} & \cdots & \mathbf{e_{mn}} \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \vdots \\ \mathbf{x_n} \end{bmatrix} = \begin{bmatrix} \mathbf{z_1} \\ \mathbf{z_2} \\ \vdots \\ \mathbf{z_m} \end{bmatrix} \in \mathbf{R^m} \quad (3.6)$$

Ở chiều giải mã, bộ giải mã được biểu diễn dưới dạng một ma trận D_{nxm}:

$$\mathbf{D} = \begin{bmatrix} \mathbf{d_{11}} & \mathbf{d_{12}} & \cdots & \mathbf{d_{1m}} \\ \mathbf{d_{21}} & \mathbf{d_{22}} & \cdots & \mathbf{d_{2m}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{d_{n1}} & \mathbf{d_{n2}} & \cdots & \mathbf{d_{nm}} \end{bmatrix} \in \mathbf{R^{nxm}} \quad (3.7)$$

Đặt x’ là vector dữ liệu sau khi giải mã, ta có:

$$\mathbf{x'} = \mathbf{Dz} = \begin{bmatrix} \mathbf{d_{11}} & \mathbf{d_{12}} & \cdots & \mathbf{d_{1m}} \\ \mathbf{d_{21}} & \mathbf{d_{22}} & \cdots & \mathbf{d_{2m}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{d_{n1}} & \mathbf{d_{n2}} & \cdots & \mathbf{d_{nm}} \end{bmatrix} \begin{bmatrix} \mathbf{z_1} \\ \mathbf{z_2} \\ \vdots \\ \mathbf{z_m} \end{bmatrix} = \begin{bmatrix} \mathbf{x'_1} \\ \mathbf{x'_2} \\ \vdots \\ \mathbf{x'_m} \end{bmatrix} \in \mathbf{R^n} \quad (3.8)$$

Hàm mất mát, hay lỗi tái tạo được xác định là:

$$\boldsymbol{\epsilon}(\mathbf{x}, \mathbf{x'}) = \|\mathbf{x} - \mathbf{x'}\|_2^2 = \|\mathbf{x} - \mathbf{DEx}\|_2^2 = \|(\mathbf{I} - \mathbf{DE})\mathbf{x}\|_2^2 \quad (3.9)$$

Trong đó, I là ma trận đơn vị kích thước nxn:

$$\mathbf{I} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \end{bmatrix} \quad (3.10)$$

Trong trường hợp tổng quát, đầu vào gồm N điểm dữ liệu khác nhau. Khi đó, ma trận biểu diễn đầu vào là:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_{11}} & \mathbf{x_{12}} & \cdots & \mathbf{x_{1N}} \\ \mathbf{x_{21}} & \mathbf{x_{22}} & \cdots & \mathbf{x_{2N}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x_{n1}} & \mathbf{x_{n2}} & \cdots & \mathbf{x_{nN}} \end{bmatrix} \in \mathbf{R^{nxN}} \quad (3.11)$$

Mỗi cột của ma trận X biểu diễn một điểm dữ liệu n chiều. Tượng ứng với đó, ta có ma trận dữ liệu được nén:

$$\mathbf{Z} = \mathbf{EX} = \begin{bmatrix} \mathbf{z_{11}} & \mathbf{x_{12}} & \cdots & \mathbf{x_{1N}} \\ \mathbf{z_{21}} & \mathbf{x_{22}} & \cdots & \mathbf{x_{2N}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z_{m1}} & \mathbf{x_{m2}} & \cdots & \mathbf{x_{mN}} \end{bmatrix} \in \mathbf{R^{mxN}} \quad (3.12)$$

Ma trận dữ liệu giải nén là:

$$\mathbf{X'} = \mathbf{DZ} = \begin{bmatrix} \mathbf{x'_{11}} & \mathbf{x'_{12}} & \cdots & \mathbf{x'_{1N}} \\ \mathbf{x'_{21}} & \mathbf{x'_{22}} & \cdots & \mathbf{x'_{2N}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x'_{n1}} & \mathbf{x'_{n2}} & \cdots & \mathbf{x'_{nN}} \end{bmatrix} \in \mathbf{R^{nxN}} \quad (3.13)$$

Và cuối cùng, lỗi tái tạo được xác định là:

$$\boldsymbol{\epsilon}(\mathbf{X}, \mathbf{X'}) = \frac{\mathbf{1}}{\mathbf{N}} \|(\mathbf{I} - \mathbf{DE})\mathbf{X}\|_2^2 \quad (3.14)$$

Đặt $L_{nxN} = (I - DE)X$, ta có:

$$\boldsymbol{\epsilon}(\mathbf{X}, \mathbf{X'}) = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} \|\mathbf{L_i}\|_2^2 \quad (3.15)$$

Trong đó L_i vector cột thứ i của ma trận L

I. T. Jolliffe trong cuốn sách *Principal Component Analysis* [34] đã chứng minh rằng lỗi tái tạo đạt giá trị nhỏ nhất khi các vector riêng đơn vị tương ứng với m trị riêng lớn nhất của ma trận hiệp phương sai là trực giao, xác định không gian con m chiều để thực hiện phép chiếu. Do đó, m vector riêng này có thể được chọn là các đặc trưng mới. Ngoài ra, ma trận của Bộ giải mã là chuyển vị của ma trận của bộ mã hóa:

$$\mathbf{D} = \mathbf{E^T} \quad (3.16)$$

Các bước thực hiện PCA có thể được tóm tắt như sau [35]:

Bước 1: Tính vector kì vọng của toàn bộ dữ liệu

$$\hat{\mathbf{e}} = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{x_i} \quad (3.17)$$

Trong đó x_i là vector cột thứ i của ma trận X, ê là một ma trận cột n hàng tướng ứng với n chiều của dữ liệu ban đầu

Bước 2: Trừ mỗi điểm trong từng cột (từng chiều) của ma trận X cho kì vọng tương ứng của chiều đó

$$\hat{\mathbf{x}}_{\mathbf{ij}} = \mathbf{x_{ij}} - \hat{\mathbf{e}}_{\mathbf{i}}, \mathbf{với i} = \overline{\mathbf{1, n}}, \mathbf{j} = \overline{\mathbf{1, N}} \quad (3.18)$$

Gọi ma trận thu được sau bước 2 là $\hat{X}$, ma trận này có kích thước giống ma trận X và trung bình của từng cột bằng 0, thể hiện kì vọng của từng chiều dữ liệu bằng 0. Việc

làm cho kì vọng từng chiều dữ liệu bằng 0 này giúp cho PCA được tính toán dễ dàng hơn khi thực hiện tối thiểu lỗi tái tạo.

Bước 3: Tính ma trận hiệp phương sai

$$\mathbf{S} = \frac{\mathbf{1}}{\mathbf{N}}\widehat{\mathbf{X}}\widehat{\mathbf{X}}^{\mathbf{T}} \tag{3.19}$$

Bước 4: Tính các trị riêng và vector riêng có norm 2 (độ dài Eclid) bằng 1 của ma trận hiệp phương sai, sắp xếp các vector riêng này theo thứ tự giảm dần của trị riêng.

Bước 5: Chọn m vector riêng ứng với m trị riêng lớn nhất để xây dựng ma trận E có các hàng tạo thành một hệ trực giao. Các vector riêng còn được gọi là các thành phần chính (Principal Component), tạo thành một không gian con *gần* với phân bố của dữ liệu ban đầu đã chuẩn hoá.

Bước 6: Chiếu dữ liệu ban đầu đã chuẩn hoá $\hat{X}$ xuống không gian con tìm được.

$$\mathbf{Z} = \mathbf{E}\widehat{\mathbf{X}} \tag{3.20}$$

Bước 7: Dữ liệu gốc có thể được khôi phục

$$\mathbf{X}' = \mathbf{E}^{\mathbf{T}}\mathbf{Z} = \mathbf{E}^{\mathbf{T}}\mathbf{E}\widehat{\mathbf{X}} \tag{3.21}$$

Vì E là một ma trận trực giao, nên $\mathrm{E}^{\mathrm{T}}\mathrm{E} = \mathrm{I}$, do đó:

$$\mathbf{X}' = \widehat{\mathbf{X}} \tag{3.22}$$

Sự sai khác giữa dữ liệu gốc và dữ liệu khôi phục nằm ở vector kì vọng ê:

$$\mathbf{X} = \mathbf{X}' + \hat{\mathbf{e}} \tag{3.23}$$

Đó là lí do vì sao ta mong muốn vector kì vọng $\hat{e}$ bằng 0. Đối với ma trận dữ liệu $\hat{X}$ có kì vọng của từng chiều bằng 0, nên theo lí thuyết có thể nén và giải nén mà không bị mất mát thông tin.

3.3.3 Những hạn chế của PCA

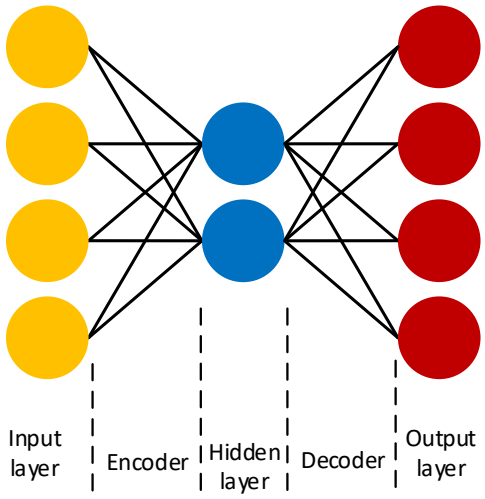
Vì ý tưởng chính của PCA là đi tìm không gian con tuyến tính của không gian gốc, nên nó chỉ có thể thực hiện nếu dữ liệu gốc là tuyến tính. Trên thực tế, các chiều của dữ liệu thường ở dạng phi tuyến, nên PCA sẽ không sử dụng được. Một hạn chế khác của PCA và việc trừ đi kì vọng của từng chiều dữ liệu để thu được dữ liệu mà kì vọng từng chiều bằng 0. Trong các lĩnh vực vật lí, ví dụ thiên văn học, giá trị các tín hiệu là

không âm, việc làm này có thể dẫn tới giá trị âm, từ đó tạo ra các thông lượng phi vật lí, và cần có mô hình chuyển tiếp để hồi phục các gia trị thực của tín hiệu. Nhìn chung, PCA có điểm mạnh là dễ thực hiện và thời gian chạy nhanh, tuy nhiên với ràng buộc phụ thuộc vào dữ liệu tuyến tính khiến cho nó khó có thể được triển khai trên nhiều loại dữ liệu thực tế.

3.4 Bộ mã hóa tự động

3.4.1 Giới thiệu

Ý tưởng về bộ mã hóa tự động (Autoencoder – AE) đã có từ thập niên 80 của thế kỉ XX [36], tuy nhiên chỉ thực sự phát triển và có tính ứng dụng thực tế trong khoảng 10 năm trở lại đây với sự ra đời của Học sâu [37]. Về cơ bản, AE có mô hình giống PCA, nhưng với bộ mã hóa và bộ giải mã là các mạng neural, và việc tìm ra mô hình AE tốt nhất được thực hiện dựa trên một thuật toán tối ưu. Một kiến trúc AE đơn giản được thể hiện trên hình 3.2.



Hình 3.2 Kiến trúc AE đơn giản

Trên hình 3.2 là một mạng neural đơn giản với một lớp ẩn với hai neurons, hai lớp đầu vào vào đầu ra đều có bốn neurons. Lớp ẩn có số neurons ít hơn lớp đầu vào và đầu ra, do đó có thể biểu diễn dữ liệu trên số chiều ít hơn dữ liệu gốc. Đây là kiến trúc cơ bản nhất của AE, là một mạng neural truyền thẳng không truy hồi (Feedforward Non-recurrent Neural Network), có một tầng đầu vào, một tầng đâu ra, có thể có một hoặc nhiều tầng ẩn. Số neurons ở tầng đầu vào và tầng đầu ra luôn bằng nhau, vì mục đích của AE là tái tạo đầu vào với sự sai khác nhỏ nhất. AE là mô hình học không giám sát, vì nó không yêu cầu có nhãn dữ liệu để cho mạng có thể học.

3.4.2 Cơ sở toán học của AE

Xét một mạng neural với lớp ẩn trong cùng có m neurons, lớp đầu vào và lớp đầu ra có n neurons. Giả sử vector dữ liệu ban đầu là x ϵ R^n như ở (). Gọi W_{mxn}, b_m, σ lần lượt là ma trận trọng số, ma trận bias và hàm kích hoạt của bộ mã hóa. Đặt z là vector dữ liệu ở tầng ẩn, ta có:

$$\boldsymbol{z = \sigma(Wx + b) \in R^m} \tag{3.24}$$

Tương tự với bộ mã hóa, gọi W'_{nxm}, b'_n và σ' lần lượt là ma trận trọng số, ma trận bias và hàm kích hoạt của bộ giải mã, ta thu được vector dữ liệu đầu ra là:

$$\boldsymbol{x' = \sigma'^{(W'z+b')} \in R^n} \tag{3.25}$$

Hàm mất mát được xác định là:

$$\boldsymbol{\mathcal{L}(x, x') = \|x - x'\|_2^2 = \left\|x - \sigma'^{(W'(\sigma(Wx+b))+b')}\right\|_2^2} \tag{3.26}$$

Hàm mất mát này có thể được tối thiểu nhờ thuật toán lan truyền ngược được áp dụng cho mạng neural truyền thẳng không phản hồi.

Như vậy, điểm khác biệt dễ nhận thấy nhất của AE so với PCA là AE có hàm kích hoạt ở các neurons. Các hàm kích hoạt này có vai trò bẻ gãy sự tuyến tính của dữ liệu, từ đó không còn ràng buộc về điều kiện tuyến tính như PCA. Về lí thuyết, khi có càng nhiều lớp ẩn thì AE càng có thể giảm số chiều dữ liệu đi nhiều lần mà vẫn có thể tái tạo dữ liệu với mất mát nhỏ. Tất nhiên, việc tái tạo dữ liệu với mất mát càng nhỏ phải đánh đổi bằng sự phức tạp của mạng. Ngoài ra, việc chọn số neurons ở lớp ẩn cũng rất quan trọng, nếu chọn quá ít thì khả năng tái tạo dữ liệu sẽ bị giới hạn, nhưng nếu chọn quá nhiều thì mạng sẽ không chọn được những đặc trưng quan trọng của dữ liệu.

3.4.3 Những hạn chế của AE

Mặc dù AE đã tốt hơn PCA rất nhiều, nhưng nó vẫn có những hạn chế nhất định. Hạn chế lớn nhất của AE là nó chỉ có khả năng tái tạo lại những dữ liệu tương tự với dữ liệu đã được huấn luyện. Hơn nữa, AE rất dễ bị hiện tượng quá khớp vì mục đích của nó là tối thiểu nhất có thể sự sai khác giữa đầu vào và đầu ra. AE không tính đến phân bố dữ liệu của lớp ẩn, nên nó có phần cứng nhắc khi tái tạo dữ liệu. Vì thế, AE không thể được sử dụng như một mô hình sinh, bởi nếu tách riêng Bộ giải mã ra để tạo

dữ liệu mới, thì chưa chắc nó có thể tạo được dữ liệu như mong muốn nếu đầu vào của bộ giải mã có sai khác (thậm chí rất nhỏ) so với dữ liệu được nén khi huấn luyện.

3.5 Bộ mã hóa tự động biển thể

3.5.1 Giới thiệu

Để khắc phục những nhược điểm của AE, vào năm 2013, Diederik P. Kingma và Max Welling đã đưa ra bộ mà hóa tự động biến thể (Variational Autoencoder – VAE) [38]. VAE hoạt động cũng giống như AE, nhưng dữ liệu ở tầng ẩn được biểu diễn dưới dạng một phân phối xác suất hay vì các neurons cố định, do đó tránh được hiện tượng quá khớp và có khả năng sinh dữ liệu mới.

Đối với VAE, bộ mã hóa được huấn luyện để tính toán kì vọng và phương sai mô tả phân phối Gaussians (hay phân phối chuẩn), từ đó biểu diễn dữ liệu của không gian ẩn. Hàm mất mát của VAE là kết hợp của hai phần: phần tái tạo (Reconstruction) giống như của AE, và phần cơ chế kiểm soát (Regularization), sử dụng để xây dựng phân phối của không gian ẩn gần với phân phối chuẩn. Phần cơ chế kiểm soát này được thể hiện dưới dạng phân kì Kullback-Leibler (Kulback-Leibler Divergence) giữa phân phối tìm được và phân phối chuẩn.

3.5.2 Cơ sở toán học của VAE

Phần này được viết chủ yếu dựa vào tài liệu tham khảo [38].

Gọi x là biến ngẫu nhiên biểu diễn dữ liệu ban đầu, z là biến ngẫu nhiên trong không gian ẩn. Giả sử z được lấy mẫu từ phân phối $p_{\theta^*}(z)$, x được lấy mẫu từ phân phối hợp lí có điều kiện (Conditional Likelihood Distribution) $p_{\theta^*}(x|z)$, trong đó θ là tham số của phân phối (kì vọng, phương sai). Thêm nữa, giả sử rằng $p_{\theta^*}(z)$ và $p_{\theta^*}(x|z)$ thuộc các họ tham số của phân phối $p_\theta(z)$ và $p_\theta(x|z)$ mà hàm mật độ xác suất của chúng khả vi tại mọi điểm. Khác với bộ mã hóa và bộ giải mã của AE ở dạng xác định, thì của VAE sẽ ở dạng xác suất. Bộ mã hóa của VAE được xác định bởi $p_\theta(z|x)$ mô tả phân phối của biến ngẫu nhiên z. Trong khí đó, bộ giải mã của VAE được xác định bởi $p_\theta(x|z)$ mô tả phân phối của biến ngẫu ngẫu nhiên được tái tạo lại.

Theo định lí Bayes trong xác suất, ta có:

$$\mathbf{p_\theta(z|x)} = \frac{\mathbf{p_\theta(x|z)p_\theta(z)}}{\mathbf{p_\theta(x)}} \tag{3.27}$$

Vì $p_\theta(x) = \int p_\theta(x|z)p_\theta(z)dz$, thay vào (3.27) được:

$$\mathbf{p_\theta(z|x)} = \frac{\mathbf{p_\theta(x|z)p_\theta(z)}}{\int \mathbf{p_\theta(x|z)p_\theta(z)dz}} \quad (3.28)$$

Với những giả sử vừa nêu trên, thì việc tính toán $p_\theta(z|x)$ thực sự khó khắn, vì ta chưa biết θ^* và z, vì thế ở đây cần sử dụng kĩ thuật xấp xỉ. Gọi $q_\phi(z|x)$ là một mô hình xấp xỉ với $p_\theta(z|x)$. Trong không gian ẩn, dữ liệu được lấy mẫu và đưa vào Bộ giải mã để sinh dữ liệu.

Một phương pháp phổ biến để xấp xỉ hai phân phối là Phân kì Kullback-Leibler. Hàm mất mát do xấp xỉ phân phối bằng Phân kì Kullback-Leibler là:

$$\mathbf{KL}\left(\mathbf{q_\phi(z|x), p_\theta(z|x)}\right) = \sum \mathbf{q_\theta(z|x)} \log \frac{\mathbf{q_\phi(z|x)}}{\mathbf{p_\theta(z|x)}} \quad (3.29)$$

Biến đổi (3.29) một chút, ta được:

$$\mathbf{KL}\left(\mathbf{q_\phi(z|x), p_\theta(z|x)}\right) = \mathbf{E}\left(\log \mathbf{q_\phi(z|x)} - \log \mathbf{p_\theta(z|x)}\right) \quad (3.30)$$

Thay (3.27) vào (3.30), ta có:

$$\mathbf{KL}\left(\mathbf{q_\phi(z|x), p_\theta(z|x)}\right) = \mathbf{E}\left(\log\ \mathbf{q_\phi(z|x)} - \log \mathbf{p_\theta(z)} - \log \mathbf{p_\theta(x|z)} + \log \mathbf{p_\theta(x)}\right) \quad (3.31)$$

Tiếp tục biến đổi (3.30), thu được:

$$\mathbf{KL}\left(\mathbf{q_\phi(z|x), p_\theta(z|x)}\right) = \mathbf{E}(\log \mathbf{p_\theta(x)} - \log \mathbf{p_\theta(x|z)}) + \mathbf{KL}\left(\mathbf{q_\phi(z|x), p_\theta(z)}\right) \quad (3.32)$$

Do kì vọng được lấy theo biến z, nên $p_\theta(x)$ không phụ thuộc vào z, cho nên:

$$\log \mathbf{p_\theta(x)} = \mathbf{KL}\left(\mathbf{q_\phi(z|x), p_\theta(z|x)}\right) + \mathbf{E}(\log \mathbf{p_\theta(x|z)}) - \mathbf{KL}\left(\mathbf{q_\phi(z|x), p_\theta(z)}\right) \quad (3.33)$$

Ở đây, $\log p_\theta(x)$ được gọi là sự hợp lí cận biên (Marginal Likelihood).

Đặt $\mathcal{L}(\theta, \phi) = E(\log p_\theta(x|z)) - KL\left(q_\phi(z|x), p_\theta(z)\right)$, ta có thể viết lại công thức (3.33):

$$\log \mathbf{p_\theta(x)} = \mathbf{KL}\left(\mathbf{q_\phi(z|x), p_\theta(z|x)}\right) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) \quad (3.34)$$

Vì số hạng đầu tiên của (3.34) luôn không âm, nên $\log p_\theta(x) \geq \mathcal{L}(\theta, \phi)$, và $\mathcal{L}(\theta, \phi)$ được gọi là cận phía dưới (lower bound) của Sự hợp lí cận biên $\log p_\theta(x)$. Mục đích của VAE là tối ưu $\mathcal{L}(\theta, \phi)$ cho cả hai tham số θ và ϕ.

Số hạng thứ nhất của $\mathcal{L}(\theta, \phi)$, $E(\log p_\theta(x|z))$ là một Ước lượng hợp lí cực đại, có vai trò giống như hàm mất mát của AE. Nhưng thay vì tối thiểu hóa, ta mong muốn tối đa hóa số hạng này vì nó thể hiện phân phối của biến ngẫu nhiên được sinh ra từ Bộ giải mã hóa.

Số hạng thứ hai của $\mathcal{L}(\theta, \phi)$, $KL\left(q_\phi(z|x), p_\theta(z)\right)$ thể hiện sự khác nhau của hai phân phối $q_\phi(z|x)$ và $p_\theta(z)$, do đó ta mong muốn số hạng này càng nhỏ càng tốt. Để đơn giản nhất, cho rằng $p_\theta(z)$ tuân theo phân phối chuẩn $\aleph(0, 1)$, từ đó $q_\phi(z|x)$ cần được tối ưu để gần với phân phối chuẩn $\aleph(0, 1)$ nhất có thể. Giả sử $q_\phi(z|x)$ tuân theo phân phối Gaussian với kì vọng và phương sai lần lượt là μ và σ^2. Sử dụng thuật toán tối ưu Bayes biến thể mã hóa tự động (Auto-encoding Variational Bayesian – AEVB) được các tác giả trình bày trong [38], $KL\left(q_\phi(z|x), p_\theta(z)\right)$ được tối ưu như sau:

$$\mathbf{KL}\left(\mathbf{q_\phi(z|x)}, \mathbf{p_\theta(z)}\right) = -\frac{\mathbf{1}}{\mathbf{2}}\left(\mathbf{1} - \boldsymbol{\mu}^{\mathbf{2}} - \boldsymbol{\sigma}^{\mathbf{2}} + \mathbf{log}\,\boldsymbol{\sigma}^{\mathbf{2}}\right) \quad (3.35)$$

3.6 Kết luận chương

Chương này đã trình bày lí thuyết về giảm chiều dữ liệu, các phương pháp PCA, AE, VAE. Những cơ sở lí thuyết này là nền tảng để xây dựng các mô hình mạng neural trong bài toán phát hiện âm thanh bất thường

CHƯƠNG 4. THÍ NGHIỆM VÀ KẾT QUẢ

Chương này trình bày về tập dữ liệu sử dụng, các thiết lập thí nghiệm cho mô hình AE và VAE, các tham số sử dụng để đánh giá mô hình, cuối cùng trình bày và so sánh các kết quả khi thực hiện với AE và VAE.

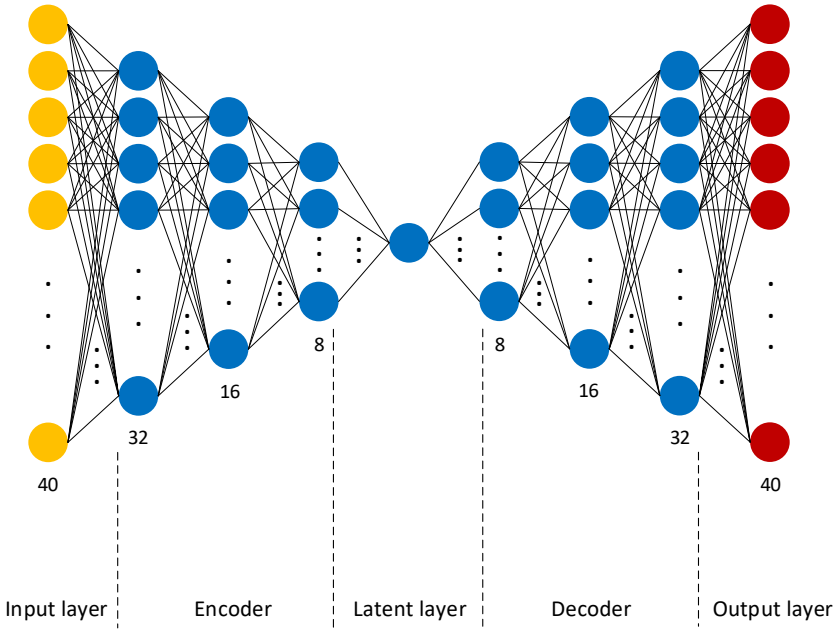
4.1 Tập dữ liệu

Để triển khai và so sánh hoạt động của AE và VAE trong việc phát hiện âm thanh bất thường, một tập dữ liệu nhỏ là các bản ghi âm cuộc họp bằng tiếng Nhật được sử dụng, có thêm những tiếng động lạ như tiếng gõ bàn, tiếng bút rơi, tiếng đập cửa, … Âm thanh thu được là âm thanh đơn âm (Monophonic Sound), tổng độ dài các bản ghi là 58 phút, trong đó tập huấn luyện dài 43 phút và tập kiểm tra dài 15 phút.

4.2 Thiết lập thí nghiệm

4.2.1 Thiết lập mạng AE

Sau khi trích xuất được đặc trưng 40 chiều là 40 log mel filter banks của mỗi frame âm thanh, các đặc trưng này được đưa vào một mạng AE như trên hình 4.1.

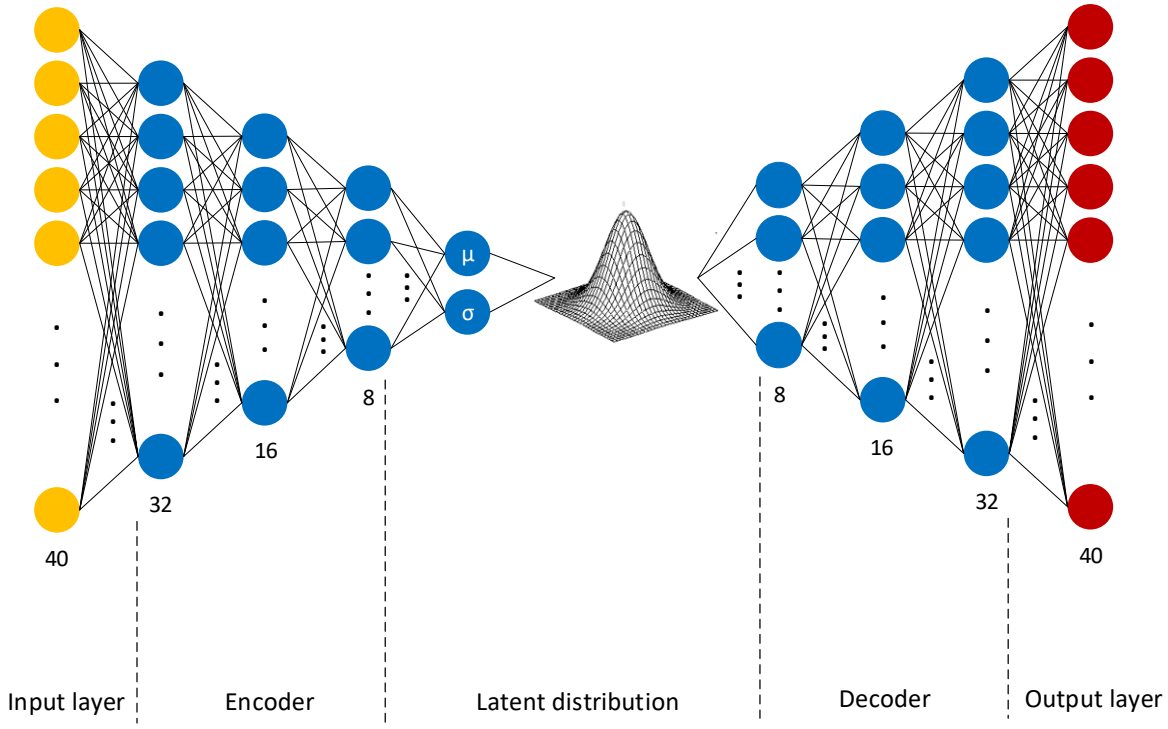


Hình 4.1 Thiết lập mạng AE

40 đặc trưng log Mel filter banks được đưa qua các lớp trung gian của bộ mã hóa xuống còn một chiều ở lớp ẩn trong cùng. Các lớp trung gian của bộ giải mã được thiết lập đối xứng với bộ mã hóa, lớp đầu ra gồm 40 neurons thể hiện 40 đặc trưng Mel filter bank được tái tạo.

4.2.2 Thiết lập mạng VAE

Mạng VAE được thiết lập giống như AE, ngoại trừ tầng ẩn trong cùng không còn là neuron cố định mà là một phân phối có kì vọng μ và phương sai σ^2, hai tham số này được tính toán từ hai neurons ở lớp trước đó (hình 4.2). Dữ liệu sau đó được lấy mẫu từ phân phối này và đưa qua bộ giải mã để sinh dữ liệu đầu ra.



Hình 4.2 Thiết lập mạng VAE

4.3 Các tham số đánh giá

Phần này được viết chủ yếu dựa vào tài liệu tham khảo [39].

4.3.1 Các tham số thống kê trung gian

Để so sánh kết quả xác định một frame âm thanh là bất thường hay không với nhãn thực tế của nó, các tham số thống kê theo đoạn (segment-based) được sử dụng. Dựa trên kết quả xác định bất thường, bốn tham số sau đây được định nghĩa trong bảng 4.1:

Bảng 4.1 Các tham số thống kê trung gian

Tham số	Nhãn	Dự đoán
True positive (TP)	Bất thường	Bất thường
False positive (FP)	Bình thường	Bất thường
True negative (TN)	Bình thường	Bình thường
False negative (FN)	Bất thường	Bình thường

Trên lí thuyết có bốn tham số thống kê, nhưng trên thực tế ta không sử dụng tham số True negative (TN) vì nó không có ý nghĩa trong phát hiện âm thanh bất thường

4.3.2 Các tham số đánh giá

4.3.2.1 Precision, Recall và F-Score

Precision (P) và Recall (R) ban đầu được sử dụng để đo độ hiệu quả của việc khôi phục thông tin, nhưng cũng có thể được sử dụng trong các bài toán phân loại. Chúng được định nghĩa như sau:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4.1}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4.2}$$

F-Score (còn được gọi là F1-Score) được tính toán dựa trên P và R:

$$\mathrm{F} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}} \tag{4.3}$$

Thay (4.1) và (4.2) vào (4.3), công thức tính F-Score được viết lại là:

$$\mathrm{F} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{4.4}$$

F-Score là một giá trị nằm trong đoạn [0, 1], ta mong muốn F-Score càng cao càng tốt.

4.3.2.2 Error Rate

Tham số Error Rate (ER) xác định tổng số lỗi thuộc các dạng “Đưa vào” (Insertions – I), “Xóa bỏ” (Deletions – D) và “Thay thế” (Substitutions – S). Để tính

ER, các lỗi được đếm trên từng segment. Trên segment k, các lỗi trên được định nghĩa như sau:

$$S(k) = \min\big(FN(k), FP(k)\big) \quad (4.5)$$

$$D(k) = \max\big(0, FN(k) - FP(k)\big) \quad (4.6)$$

$$I(k) = \max\big(0, FP(k) - FN(k)\big) \quad (4.7)$$

Giả sử N(k) là số frame có nhãn bất thường trên segment k, ER được tính là:

$$ER = \frac{\sum_{k=1}^{K} S(k) + \sum_{k=1}^{K} D(k) + \sum_{k=1}^{K} I(k)}{\sum_{k=1}^{K} N(k)} \quad (4.8)$$

Giá trị của ER nằm trong nửa khoảng [0, ∞), ta mong muốn ER càng nhỏ càng tốt

4.3.3 Kết quả thí nghiệm

Bảng () so sánh kết quả thí nghiệm phát hiện tiếng động lạ khi sử dụng AE và VAE

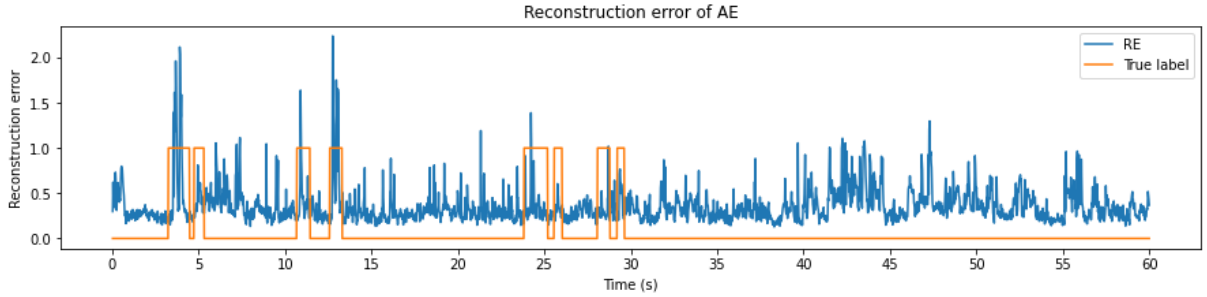
Bảng 4.2 Kết quả thí nghiệm

	AE	**VAE**
Ngưỡng cho F-Score tốt nhất	1.06	1.49
F-Score tốt nhất	0.5169	0.4658
ER tương ứng với F-Score tốt nhất	1.0639	1.0372
Ngưỡng cho ER tốt nhất	1.41	1.97
F-Score tương ứng với ER tốt nhất	0.5138	0.2963
ER tốt nhất	0.7447	0.9096

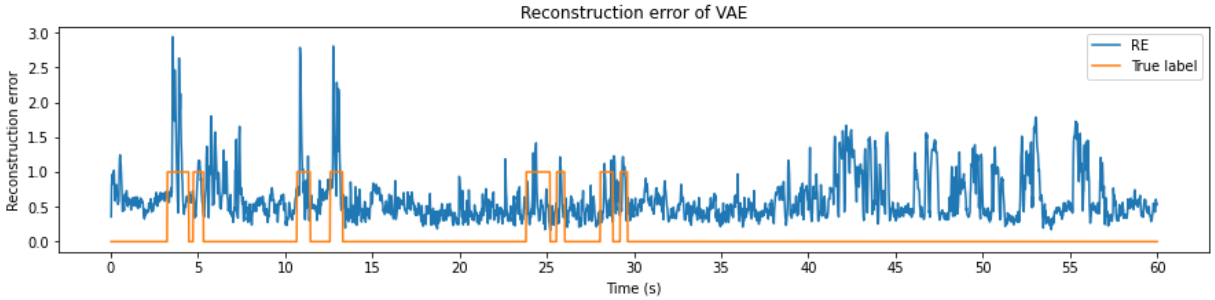
AE và VAE là các phương pháp học bán giám sát, vì việc chọn ngưỡng để xác định bất thường cần có sự tác động của con người, khi lỗi tái tạo lớn hơn ngưỡng được chọn thì tại đó được xác định là bất thường. Như ở trên bảng 2, ta có thể chọn ngưỡng khác nhau để cho ra các tham số F-Score tốt nhất cũng như ER tốt nhất. Dữ liệu trên bảng 2 đã chỉ ra AE có kết quả tốt hơn VAE trong việc phát hiện âm thanh bất thường đối với tập dữ liệu sử dụng. Điều này có thể được giải thích là do tập dữ liệu nhỏ, mà VAE lại

là một mô hình xác suất, nên việc có ít dữ liệu huấn luyện ảnh hưởng không nhỏ đến kết quả của VAE.

Sau đây là phần phân tích những vị trí bị dự đoán sai của AE và VAE. Hình 4.3 và 4.4 thể hiện lỗi tái tạo của một đoạn âm thanh trong tập kiểm tra.



Hình 4.3 Lỗi tái tạo của AE



Hình 4.4 Lỗi tái tạo của VAE

Có thể thấy, AE và VAE đều phát hiện rất tốt một số vị trí âm thanh bất thường, đó là những âm thanh đơn lẻ không bị lẫn tiếng nói, có một vài chỗ những tiếng động lạ nhỏ và bị lẫn vào tiếng nói AE không phát hiện được nếu chọn ngưỡng cao. Ngoài ra, khoảng thời gian từ giây thứ 40 trở đi có nhiều chỗ có lỗi tái tạo cao, đó là những chỗ có tiếng nói lớn, rất dễ bị nhầm thành bất thường nếu chọn ngưỡng thấp. Ở những vị trí này, VAE có lỗi tái tạo cao hơn AE, dẫn đến kết quả xâu hơn AE.

4.4 Kết luận chương

Trong chương này, các thí nghiệm đã được thực hiện để đánh giá kết quả của việc phát hiện âm thanh bất thường đối với AE và VAE. Mặc dù kết quả thí nghiệm đã chứng minh AE cũng như VAE có khả năng phát hiện âm thanh bất thường, tuy nhiên độ hiệu quả chỉ dừng ở mức chấp nhận được. Tập dữ liệu sử dụng chưa đủ lớn nên kết quả có được chưa thực sự tốt.

KẾT LUẬN

Kết luận chung

Sau 3 tháng tìm hiểu và thực hành trên các mô hình mạng và dữ liệu thật, em đã đạt được một số kết quả tốt trong việc nghiên cứu các phương pháp phát hiện âm thanh bất thường. Ngoài ra, em cũng đã tích lũy cho mình một lượng kiến thức và kinh nghiệm nhất định trong thời gian làm đề tài, những kinh nghiệm này chắc chắn sẽ giúp ích cho cá nhân em rất nhiều cho công việc sau này. Trong đề tài lần này, vẫn còn nhiều vấn đề tồn đọng trong quá trình thực hiện mà em chưa giải quyết được, em sẽ cố gắng khắc phục trong thời gian tới. Em em xin cảm ơn thầy Hàn Huy Dũng và các bạn trong SPARC Laboratory đã hỗ trợ hết mình để có được kết quả như hôm nay.

Hướng phát triển

Nhóm sẽ tiếp tục tìm hiểu và nâng cấp hiệu suất của mô hình thông qua một số phương pháp như là thu thập thêm dữ liệu để mô hình huấn luyện, thay đổi số lượng các node trong mỗi layer của mô hình để nhằm đạt được hiệu suất cao nhất. Ngoài ra, nhóm sẽ tìm hiểu thêm một số mô hình mạng có chức năng tương tự và đánh giá, so sánh với các mô hình đã nghiên cứu.

TÀI LIỆU THAM KHẢO

[1] J. McCarthy, M. L. Minsky, N. Rochester and C. Shannon, "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence," Nikos Drakos, Computer Based Learning Unit, University of Leeds, England, 1956.

[2] M. Minsky and S. Papert, Perceptrons: an introduction to computational geometry, Cambridge, Massachusetts, USA: MIT Press, 1969.

[3] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," in *Neurocomputing: foundations of research*, Cambridge, Massachusetts, USA, MIT Press, 1988, pp. 696-699.

[4] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210-229, July 1959.

[5] T. M. Mitchel, Machine learning, New York, USA: McGraw-Hill Education, 1997.

[6] M. Copeland, "What's the Difference Between Artificial Intelligence, Machine Learning and Deep Learning?," NVIDIA, 29 July 2016. [Online]. Available: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/.

[7] P. Coucke, B. D. Ketelaere and J. D. Baerdemaeker, "Experimental analysis of the dynamic, mechanical behavior of a chicken egg," *Journal of Sound and Vibration*, vol. 266, pp. 711-721, 2003.

[8] Y. Chung, S. Oh, J. Lee, D. Park, H. H. Chang and S. Kim, "Automatic Detection and Recognition of Pig Wasting Diseases Using Sound Data in Audio Surveillance Systems," *Sensors*, p. 12929–12942, 2013.

[9] A. Yamashita, T. Hara and T. Kaneko, "Inspection of Visible and Invisible Features of Objects with Image and Sound Signal Processing," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2006)*, Beijing, China, October 2006.

[10] Y. Koizumi, S. Saito, H. Uematsu and N. Harada, "Optimizing Acoustic Feature Extractor for Anomalous Sound Detection Based on Neyman-Pearson Lemma," in *Proceedings of European Signal Processing Conference (EUSIPCO)*, Greek island of Kos, September 2017.

[11] C. Clavel, T. Ehrette and G. Richard, "Events Detection for an Audio-Based Surveillance System," in *Proceedings of 2005 IEEE International Conference on Multimedia and Expo*, Amsterdam, Netherlands, July 2005.

[12] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci and A. Sarti, "Scream and Gunshot Detection and Localization for Audio-Surveillance," in *Proceedings of Advanced Video and Signal based Surveillance*, London, UK, September 2007.

[13] S. Ntalampiras, I. Potamitis and N. Fakotakis, "Probabilistic Novelty Detection for Acoustic Surveillance Under Real-World Conditions," *IEEE Transactions on Multimedia* , vol. 13, no. 4, pp. 713-719, August 2011.

[14] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio and M. Vento, "Audio Surveillance of Roads: A System for Detecting Anomalous Sounds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 279-288, January 2016.

[15] H. Eghbal-Zadeh, B. Lehner, M. Dorfer and G. Widmer, "CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors," DCASE2016 Challenge, Tech. Rep., Budapest, Hungary, September 2016.

[16] Y. Han and J. Park, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," DCASE2017 Challenge, Tech. Rep., Munich, Germany, September 2017.

[17] T. Komatsu, T. Toizumi, R. Kondo and Y. Senda, "Acoustic event detection method using semi-supervised non-negative matrix factorization," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop*, Budapest, Hungary, September 2016.

[18] T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. L. Roux and K. Takeda, "Duration-controlled LSTM for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 11, pp. 2059 - 2070, August 2017.

[19] S. Adavanne, A. Politis and T. Virtanen, "Multichannel Sound Event Detection Using 3D Convolutional Neural Networks for Learning Inter-channel Features," in *arXiv preprint arXiv:1801.09522, 2018*, January 2018.

[20] M. Basseville and I. V. Nikiforov, Detection of Abrupt Changes: Theory and Application, vol. 104, New Jersey, USA: Prentice Hall Englewood Cliffs, 1993.

[21] S. Liu, M. Yamada, N. Collier and M. Sugiyama, "Change-Point Detection in Time-Series Data by Relative Density-Ratio Estimation," *Neural Networks*, vol. 43, pp. 72-83, 2013.

[22] Y. Ono, Y. Onishi, T. Koshinaka, S. Takata and O. Hoshuyama, "Anomaly detection of motors with feature emphasis using only normal sounds," in *Acoustics, Speech, and Signal Processing (ICASSP), International Conference on*, Vancouver, Canada, May 2013.

[23] L. Tarassenko, P. Hayton, N. Cerneaz and M. Brady, "Novelty detection for the identification of masses in mammograms," in *1995 Fourth International Conference on Artificial Neural Networks*, Cambridge, UK, June 1995.

[24] J. Foote, "Automatic audio segmentation using a measure of audio," in *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE*, New York, USA, August 2000.

[25] M. Markou and S. Singh, "Novelty detection: a review-part 1: statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481-2497, December 2003 .

[26] D. W. Scott, "Outlier detection and clustering by partial mixture modeling," in *COMPSTAT 2004 Proceedings in Computational Statistics*, 2004.

[27] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in *Proceedings of the International Joint Conference on Neural Networks*, Oregon, USA, July 2003.

[28] M. Sakurada and T. Yairi, "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, New York, USA, December 2014.

[29] E. Marchi, F. Vesperini, F. Eyben, S. Squartini and B. Schuller, "A novel approach

for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, Brisbane, QLD, Australia , April 2015.

[30] N. Kehtarnavaz, Digital Signal Processing System Design: LabVIEW-Based Hybrid Programming, Cambridge, Massachusetts, USA: Academic Press, 2008.

[31] X. Huang, A. Acero and H. Hon., Spoken Language Processing: A guide to theory, algorithm, and system development, Prentice Hall, 2001.

[32] V. H. Tiệp, Machine Learning cơ bản, Hà Nội, VIệt Nam: NXB Khoa học kỹ thuật, 2018.

[33] J. Rocca, "Understanding Variational Autoencoders (VAEs)," Towards Data Science Inc., 24 September 2019. [Online]. Available: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73.

[34] I. T. Jolliffe, "Properties of Population Principal Components," in *Principal Component Analysis*, USA, Springer Publishing, 2002, pp. 10-28.

[35] L. I. Smith, "A tutorial on Principal Components Analysis," Computer Science Technical Report, 2002.

[36] R. S. Z. Geoffrey E. Hinton, "Autoencoders, Minimum Description Length and Helmholtz Free Energy," in *Advances in Neural Information Processing Systems 6*, Denver, Colorado, USA, November 1993.

[37] P. Baldi, "Autoencoders, Unsupervised Learning, and Deep Architectures," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, Bellevue, Washington, USA, July 2011.

[38] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *International Conference on Learning Representations*, Banff, Canada, April 2014.

[39] M. A., H. T. and V. T., "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, 2016.