**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

---


**Nguyễn Lý Hòa**

# PHÁT HIỆN VÀ PHÂN LOẠI ÂM THANH HO TRÊN CÁC THIẾT BỊ IOT

**Chuyên ngành: Khoa học máy tính**

**Mã số: 8.48.01.01**

**LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH**

**(*Theo định hướng ứng dụng*)**

NGƯỜI HƯỚNG DẪN KHOA HỌC : PGS. TS. PHẠM VĂN CƯỜNG

HÀ NỘI - NĂM 2020

# MỤC LỤC

**MỤC LỤC** ..... **i**
**DANH MỤC HÌNH VẼ** ..... **iii**
**DANH SÁCH BẢNG** ..... **iv**
**DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT** ..... **v**
**BẢN CAM ĐOAN** ..... **vi**
**LỜI CẢM ƠN** ..... **vii**
**LỜI NÓI ĐẦU** ..... **viii**
**CHƯƠNG 1: TỔNG QUAN VỀ PHÂN LOẠI HO** ..... **1**
**1.1 Bài toán phát hiện và phân loại ho** ..... **1**
**1.2 Một số nghiên cứu liên quan** ..... **3**
**1.3 Các dạng ho dựa trên bệnh lý con người** ..... **8**
**1.4 Kết luận** ..... **12**
**CHƯƠNG 2: PHƯƠNG PHÁP PHÂN LOẠI HO** ..... **13**
**2.1 Xử lý âm thanh ho** ..... **13**
**2.2 Mô hình máy học Gaussian hỗn hợp (GMM) cho phát hiện và phân loại ho** ..... **15**
2.2.1 Restricted Boltzmann Machine ..... 16
2.2.2 Mạng học sâu (DNN) ..... 20
**2.3 Mô hình máy học CNN-LSTM sử dụng cho việc phát hiện và phân loại ho** ..... **23**
2.3.1 Mạng học sâu tích chập cho phát hiện và phân loại ho (CNN) ..... 24
2.3.2 Áp dụng mô hình Sequence-to-Sequence cho việc phân loại và phát hiện ho ..... 30
**CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ** ..... **42**
**3.1 Thu thập dữ liệu** ..... **42**
3.1.1 Thu âm và gán nhãn âm thanh ..... 42
3.1.2 Xây dựng và đánh giá âm thanh ..... 45
**3.2 Huấn luyện dữ liệu** ..... **46**
**3.3 Thử nghiệm phát hiện và phân loại ho** ..... **48**
3.3.1 Thử nghiệm 1 ..... 48

3.3.2 Thử nghiệm 2 .....49
3.3.3 Thử nghiệm 3 .....50
3.3.4 Thử nghiệm 4 .....51
3.3.5 Thử nghiệm 5 .....51
**3.4 Kết quả thử nghiệm.....51**
**3.5 Kết luận .....56**
**CHƯƠNG 4: KẾT LUẬN .....59**
**TÀI LIỆU THAM KHẢO .....61**

# DANH MỤC HÌNH VẼ

Hình 1.1: Biểu đồ dịch bệnh Covid – 19 năm 2020 (nguồn: google).....1
Hình 2.1 Thang điểm thể hiện độ đau (VAS) .....13
Hình 2.2: Tổng quan về phương pháp hiện ho thông thường so với phương pháp học sâu.....14
Hình 2.3: Ví dụ đơn giản của RBM với 4 khối ẩn và 3 khối hiển thị.....17
Hình 2.4 Quá trình đào tạo kết hợp giữa DNN và GMM-HMM.....22
Hình 2.5 Một minh họa của mạng nơ-ron tích chập và quy hồi cho hai công thức phát hiện ho. .....23
Hình 2.6: Mô tả kiến trúc CNN.....29
Hình 2.7: Mô hình LSTM .....33
Hình 2.8: Kiến trúc của mô hình Sequence-to-Sequence với câu đầu vào là chuỗi “A B C D” và câu đầu ra là chuỗi “X Y Z” .....35
Hình 2.9: Tổng quan về kiến trúc RNN bộ mã hóa – giải mã để phát hiện ho.....40
Hình 3.1: Thiết bị thu âm được cung cấp tới bệnh nhận.....43
Hình 3.2: Một số các cổng chuyển đổi được sử dụng cho việc kết nới mic với các thiết bị không hỗ trợ cổng cắm 3.5.....43
Hình 3.3: Một số phần đánh giá của các bác sỹ chuyên môn .....44
Hình 3.4: sử dụng phần mềm Audacity thực hiện gán nhãn âm thanh .....44
Hình 3.5: Đồ thị so sánh AUC của CNN và RNN.....52
Hình 3.6: Ma trận nhầm lẫn cho (a) CNN và (b) RNN trong bài toán phân loại nhiều lớp tại thử nghiệm 2. .....53
Hình 3.7: Giảm số lượng lớp của hai mạng .....55
Hình 3.8: Giảm số lượng các đơn vị trong hai mạng.....55
Hình 3.9: Hiệu suất của RNN (LSTM) khi số lượng các đơn vị giảm .....56

# DANH SÁCH BẢNG

Bảng 1.1: Các nguyên nhân hình thành ho không do lây nhiễm .....10
Bảng 1.2: Các nguyên nhân hình thành ho do lây nhiễm .....11
Bảng 2.1: Mô tả thuật toán huấn luyện Mạng học sâu Bayes.....20
Bảng 3.1: So sánh các kết quả của CNN, RNN và MFCC cho việc phân loại ho tại thử nghiệm 1.....52
Bảng 3.2: So sánh kết quả giữa các mạng khi sử dụng các chuỗi dài hơn .....53
Bảng 3.3: So sánh CNN và RNN khi sử dụng .....55

# DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

| Viết tắt | Tiếng Anh | Tiếng Việt |
|---|---|---|
| IoT | Internet of Thing | Internet Vạn Vật |
| HMM | Hidden Markov Model | Mô hình Markov ẩn |
| GMM | Gaussian Mixture Model | Mô hình Gaussian hỗn hợp |
| DNN | Deep Neural Network | Mạng nơ ron sâu |
| ANN | Artificial Neural Network | Mạng nơ ron nhân tạo |
| CNN | Convolutional Neural Network | Mạng học sâu tích chấp |
| RNN | Recurrent Neural Network | Mạng học sâu quy hồi |
| LSTM | Long shot term memory | Bộ nhớ dài – ngắn hạn |
| RBM | Restricted Boltzmann Machine | Máy Boltzmann bị hạn chế |
| DBN | Deep Bayesian Networks | Mạng học sâu Bayes |
| SVM | Support Vector Machine | Máy véc tơ hỗ trợ |
| SFFT | Sparse Fast Fourier Transform | Biến đổi Fourier nhanh |
| MFCC | Mel Frequency Cepstral Coefficients | Phương pháp trích xuất đặc trưng âm thanh |

# BẢN CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

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

Hà Nội, ngày … tháng … năm 2020

HỌC VIÊN CAO HỌC

**Nguyễn Lý Hòa**

# LỜI CẢM ƠN

Trong quá trình học tập, nghiên cứu và hoàn thành luận văn, tôi đã nhận được sự động viên, khuyến khích và tạo điều kiện giúp đỡ nhiệt tình của các cấp lãnh đạo, của các thầy giáo, cô giáo, anh chị em, bạn bè đồng nghiệp và gia đình.

Tôi muốn bày tỏ lòng biết ơn sâu sắc tới các thầy cô giáo, phòng Sau đại học Học viện Công nghệ Bưu chính Viễn Thông và đặc biệt là các thầy cô giáo trực tiếp giảng dạy các chuyên đề của khóa học đã tạo điều kiện, đóng góp ý kiến cho tôi trong suốt quá trình học tập và hoàn thành luận văn thạc sỹ.

Đặc biệt, tôi xin bày tỏ lòng biết ơn sâu sắc tới PGS.TS.Phạm Văn Cường – Người đã trực tiếp hướng dẫn, tận tình chỉ bảo, giúp đỡ tôi tiến hành các hoạt động nghiên cứu khóa học để hoàn thành luận văn này.

Với thời gian nghiên cứu còn hạn chế, thực tiễn công tác lại vô cùng sinh động, luận văn không thể tránh khỏi những thiếu sót, tôi rất mong nhận được các ý kiến đóng góp chân thành từ các thầy giáo, cô giáo, đồng nghiệp, bạn bè.

Hà Nội, ngày tháng năm 2020

Học viên

**Nguyễn Lý Hòa**

# LỜI NÓI ĐẦU

Các loại bệnh dịch trong suốt bề dày lịch sử của loài người đã có sự phát triển, phân cấp các loại bênh biến đổi và thay đổi không ngừng. Một trong số đó không thể không nhắc tới những bệnh liên quan tới đường hô hấp, đặc biệt là dịch bệnh COVID-19 kinh hoàng gần đây đã và đang làm cho thế giới chao đảo. Tính từ đầu năm 2020 đến nay đã có tới 43,9 triệu người nhiễm trên toàn thế giới và hơn 1,16 triệu người tử vong. Tại Việt Nam, chúng ta đã vô cùng thành công khi chỉ có 1169 người nhiễm, chỉ có 35 người tử vong với bệnh nền nặng. Dịch bệnh đã lây lan đến mức kinh hoàng trên quy mô cả thế giới như vậy hầu như là do tính chủ quan của người bệnh, cũng như việc đánh giá sai các triệu chứng của mình.

Các dịch bệnh về hô hấp đã nhiều lần gây ra sự hỗn loạn trên thế giới, chủng bệnh thay đôi liên tục, tăng cường thích nghi với mỗi lần chúng ta tìm ra vacxin điều trị. Ta cũng có thể thấy được các đại dịch lớn đều có các triệu chứng liên quan tới triệu chứng ho tiêu biểu như: bệnh lao, đại dịch cúm Tay Ban Nha (1918), dịch cúm Châu Á (1957),… và đặc biệt dịch SARS (2003) nay đã biến đổi và quay trở lại với tên gọi COVID-19. Triệu chứng ho là một trong các triệu chứng đặc trưng của các dịch bệnh này, với mỗi dịch bệnh về hồ hấp mới sẽ có các đặc trưng ho riêng biệt như đối với dịch COVID-19 là triệu chứng ho khan cùng với các triệu chúng khác ta có thể phân biệt chúng với các triệu chứng cúm thông thường. Chính vì sự thiếu hiểu biết cũng như chủ quan của con người mà đôi khi đã khiến căn bênh trở lên trầm trọng hơn hoặc gây nguy hiểm cho những người xum quanh. Đặc biệt với thực trạng đang quá tải các bệnh viên như hiện nay thì việc có thể nhận dạng bệnh lý còn khó khăn hơn khi không thể nghe những lời khuyên y tế.

Tuy nhiên, sự phát triển của con người cũng đã tạo ra sự phát triển về công nghệ thông tin, sự phát triển của các thiết bị IoT. Chúng ta đã áp dụng công nghệ thông tin vào các ngành nghề khác từ giao thông vận tại, giáo dục, sản xuất chế tạo,… Con người cũng đã áp dụng Công nghệ thông tin trong y tế, dựa trên các hệ thống lưu động, lưu trữ thông tin bệnh nhân; hệ thống báo hiệu sống còn của bệnh nhân; hay cả

đến ứng dụng hệ thống trí tuệ nhân tạo, học sâu để nhận biết các dao động của nhịp tim, nhận dạng tiếng thở, âm thanh ho… Bằng cách nghiên cứu các phương pháp học sâu, tôi mong muốn có thể đưa ra được phương pháp tốt nhất cho việc phân loại âm thanh ho. Nhờ đó, chúng ta có thể phát triển các ứng dụng dựa trên các thiết bị IoT để ai cũng có thể nhận biết được dạng ho của bản thân, cùng với các triệu chứng đi kèm có thể tư đưa ra sơ bộ về tình hình cá nhân để  can thiệp kịp thời với tình trạng của mình cũng như sẽ không gây ra sự lây lan, nguy hiểm đến những người xum quanh.

# CHƯƠNG 1: TỔNG QUAN VỀ PHÂN LOẠI HO

## 1.1 Bài toán phát hiện và phân loại ho

Như chúng ta đã, biết đối với hiện trạng như hiện nay trên thế giới tình trạng đại dịch COVID-19 truyền nhiễm theo cấp số nhân vô cùng nghiêm trọng tăng tới hàng trăm nghìn ca nhiễm và hàng nghìn ca tử vong mỗi này. Dựa vào biểu đồ tại hình 1.1, ta cũng có thể thấy được dịch bệnh này chưa hề có dấu hiệu dừng lại. Theo dữ liệu thống kê hiện tại:

Số ca nhiễm

| Vị trí | Tổng số ca nhiễm | Số ca nhiễm mới (1 ngày*) | Số ca nhiễm mới (60 ngày qua) | Số ca nhiễm trên 1 triệu người | Số ca tử vong |
|---|---|---|---|---|---|
| Toàn thế giới | 43.894.430 | *Không có dữ liệu* | | 5.645 | 1.165.432 |
| Việt Nam | 1.169 | 1 | | 12 | 35 |

Hình 1.1: Biểu đồ dịch bệnh Covid – 19 năm 2020 (nguồn: google)

Và đó chỉ là một trong những bệnh dịch gần đây nhất đang hoành hành trên thế giới tại thời điểm hiện tại. Trong suốt cả quá trình phát triển của loài người đã có những dịch bệnh kinh hoàng hơn như vậy. Bệnh dịch hạch được biết tới từ những năm 541 sau công nguyên từng khiến cho các đế chế Hy Lạp chao đảo, cũng căn bệnh này đã khiến cho cả Châu Âu chao đảo từ 1347 đến 1351 số người chết lên tới 25 triệu người. Bệnh đậu mùa khi thực dân châu Âu xâm chiếm châu Mỹ vào thế kỷ 17 và mang theo cả bệnh đậu mùa (do virus variola gây ra) tới lục địa này. Bệnh đậu mùa đã cướp đi sinh mạng của khoảng 20 triệu người, gần 90% dân số ở châu Mỹ khi đó. Hay đại dịch tả đầu tiên bắt đầu ở Jessore, Ấn Độ (1817-1823) và giết chết hàng triệu người dân Ấn Độ khi ấy. Sau đó, dịch tả bùng phát thêm nhiều đợt mới lan nhanh khắp các châu lục trong thời gian ngắn. Trong số đó, không thể không nhắc tới các đại dịch cúm đã liên tục hoành hành trong loài người như dịch cúm Tây Ban Nha (1918), dịch cúm Châu Á (1957),… Các dịch cúm liên tục thay đổi và thích nghi sau

mỗi lần loại người có thể phòng chống và chữa trị được, như đại dịch COVID-19 cũng là một bản sao sự biến đổi từ chủng của đại dịch SARS năm 2003. Tuy nhiên, mỗi dịch bệnh xảy ra đều có các triệu chứng liên quan để chúng ta có thể dễ dàng phòng ngừa được, tuy nhiên do sự chủ quan và thiếu hiểu biết đã khiến cho loài người rơi vào tình cảnh khó khăn.

Phần lớn các căn bệnh liên quan tới được hô hấp hay dịch cúm đếu có các triệu chứng ho, chẳng hạn như đối với dịch bệnh COVID-19, ngoài các triệu chứng cụ thể ra kèm với đó còn có dạng ho riêng. Tùy theo cơ địa mỗi người mà các triệu chứng dạng ho có thể là rõ hoặc không rõ nhưng vẫn có cùng dạng ho. Mỗi dạng ho, khi đi kèm với những triệu chứng khác nhau có thể là những căn bệnh khác nhau, đôi khi để cá nhân người bệnh có thể tự nhận biết hay phân loại được dạng ho của mình để có thể phòng ngừa cũng là một điều khó khăn khi không có các lời khuyên từ các bác sĩ chuyên ngành. Vì vậy, ứng dụng các kỹ thuật công nghệ để có thể phát hiện và phòng ngừa là một điều cấp thiết.

Với sự phát triển của ngành Công nghệ thông tin, chúng ta đã ứng dụng được vào các ngành nghề khác để có thể hoạt động dễ dàng hơn. Ngoài những ngành nghề khác, việc áp dụng công nghệ thông tin vào ngành y học là vô cùng cần thiết. Ngoài những công việc áp dụng công nghệ thông tin cơ bản, chúng ta cũng đã có nhiều thuật toán học sâu, ứng dụng trí tuệ nhân tạo để hỗ trợ con người trong ngành y học, dễ dàng hơn trong việc chuẩn đoán tình trạng của bệnh nhân và kịp thời phòng ngừa. Giả sử như bằng các thuật toán trí tuệ nhân tạo để kiểm tra và phân loại nhịp tim thai nhi [16], sử dụng để dự đoán ngừng tim đột ngột dựa trên các thông tin thay đổi của nhịp tim [17] hay các thuật toán học máy để kiểm tra tâm lý con người như kiểm tra các khái niệm về tự tử và cảm xúc của thanh thiếu niên để ngăn chặn việc tự tử [18],... Nhận thấy được khả năng khi áp dụng máy học cho các vấn đề về y học cùng với thực trạng hiện nay đối với các dịch bệnh cúm mùa đặc biệt là đại dịch COVID-19, tôi muốn áp dụng các thuật toán học máy để có thể thực hiện bài toán phát hiện và phân loại các dạng ho trên các thiết bị IoT, vậy tại sao là các thiết bị IoT? Đối với các thiết bị IoT thì gần như mọi người đều đang sử dụng chúng chẳng hạn như điện thoại thông

minh Smartphone, thiết bị đeo tay điện tử… Để tiếp cận hệ thống này cho những người sử dụng phổ thông là vô cùng đơn giản, không cần phải có các thiết bị điện tử quá đắt tiền cũng như gây khó chịu cho người sử dụng, không những thế người sử dụng có thể tiếp cận mọi lúc mọi nơi. Việc thực hiện bài toán này chính là tiền đề để tạo ra hệ thống tối ưu nhất và có thể góp phần giúp người sử dụng nhanh chóng biết được tình trạng của bản thân và mau chóng chữa trị, phòng ngừa tránh lây lan ra cộng đồng người xum quanh mình. Xác suất nào đó có thể giảm thiểu số người nhiễm bệnh và người tử vong nếu như có những đại dịch cúm trong tương lai.

## 1.2 Một số nghiên cứu liên quan

Phát hiện, phân loại ho và đánh giá mức độ nghiêm trọng của triệu chứng tự động đã thu hút các nhà nghiên cứu, chuyên gia y tế và bác sỹ trong nhiều năm. Phần lớn người đến chưa trị, hay cần lời khuyên từ bác sỹ đều có chung triệu chứng là ho [3]. Người ta có thể phát hiện âm thanh ho dựa trên việc lắp đặt hàng loạt các cảm biến âm thanh [4], đối với [6], [7], [8], thì họ chỉ cần sử dụng duy nhất một micro gắn lên người có triệu chứng nhiễm bệnh [2] hoặc sử dụng hệ thống mic thu âm của thiết bị di động [1], [11]. Trước đây, các hệ thống cảm biến được lắp đặt theo thứ tự cụ thể trong một môi trường đa chiều để có thể cảm nhận được biến động từ môi trường xum quanh và cảnh báo nguy hiểm [8], trong khi đó các thiết bị cảm biến được đeo trên người thường có tác dụng để có thể nhận biết hành vi của người đeo, cảm biến các tác động, báo hiệu khi người sử dụng bị ngã [9] hoặc là các hệ thống cảm biến này có thể nhận diện được các dấu hiệu sự sống của còn người chẳng hạn như nhịp tim, hơi thở, huyết áp, nồng độ Oxi trong máu để báo hiệu cho người sử dụng khi tới ngưỡng nguy hiểm liên quan tới tính mạng. Tất nhiên, với việc sử dụng nhiều các cảm biến được cài đặt trong môi trường có thể mất khá nhiều kinh phí cũng như khả năng kết nối bởi sự hạn chế của phạm vi cảm biến chỉ có 1 khoảng nhất định (ví dụ như tầm tín hiệu của cảm biến chỉ ở trong phòng hoặc trong nhà). Đối với các thiết bị điện thoại di động hay là các thiết bị IoT có tích hợp cảm biến thì có thể cho phép người dùng phát hiện, phân loại ho và mức độ nghiêm trọng của triệu chứng mọi nơi, mọi lúc.

Các cảm biến hệ thống âm thanh được sử dụng để tạo ra các máy phát hiện ho là tương đối phổ biến, tính tới thời điểm hiện tạo các thiết bị đó có thể nhận dạng được âm thanh ho chuẩn tới 95%. Một số nhà khoa học đã áp dụng phương pháp phát hiện âm thanh ho dựa trên mạng nơ-ron nhân tạo, được tạo ra bởi các vec tơ từ 222 đặc trưng [6], trong khi đó [4] bằng cách đặt các cảm biến ho tại các vị trí trên cơ thể người, so sánh các kết quả và đưa ra kết luận chính việc thay đổi các vị trí đặt máy cũng có thể liên quan tới sự chính xác của máy phát hiện ho, Vizel E. et al. Còn [5] thì đưa ra thông tin về âm thanh ho bằng cách phân tích tổng hợp hai luồng đó là âm thanh được thu từ một chiếc micro được đặt trên ngực và âm thanh được thu từ các cảm biến được cài đặt trong môi trường xum quanh người đeo micro. Tương tự, Zheng, S., et al. [7] CoughLoc phân tích âm thanh ho dựa từ dữ liệu thu được tại mạng cảm biến không dây không xâm nhập, bên cạnh đó CoughLoc cũng phân tích xem tại vị trí thu được các âm thanh ho để tăng độ nhận diện âm thanh chính xác nhất. [6] nhận dạng âm thanh ho bằng cách sử dụng các cảm biến khác nhau bao gồm cả gia tốc kế EMT 25 C (Siemens); Gia tốc kế PPG 201 (PPG); Micro Sony ECM-T150 kết nối với bộ đầu nối nén khí và so sánh chất lượng chuyển đổi và nhận dạng với các thiết bị thu âm thanh của phổi. Bên cạnh đó, chúng ta còn có cách thu âm thanh ho chỉ bằng chiếc micro đeo trên ngực của người bệnh và phân tích dựa trên chính dữ liệu mà chiếc micro đó thu được. Chẳng hạn, [2] Leicester Cough Monitor đề xuất sử dụng máy ghi âm trên ngực bệnh nhân, kế quả LCM đã được đánh giá nghiêm ngặt, đạt được độ nhạy và độ đặc hiệu tỉ lệ cao 91% trên bộ dự liệu ngoại tuyến của 15 bệnh nhân ho mãn tính và 8 người khỏe mạnh. Một nghiên cứu khác, [8] đã sử dụng phương pháp học sâu áp dụng mô hình Markov ẩn với hơn 800 phút ghi âm và đã phát hiện ra các hiện tượng ho chính xác tới 82% cùng với tỉ lệ lỗi thấp khoảng 7 lần mỗi giờ.

Bên cạnh đó cũng đã có nhiều phương pháp áp dụng học sâu (Deep Learning Machine) vào việc phát hiện và phân loại các dạng ho. [14] đã phát hiện ho bằng cách sử dụng nhận dạng hình ảnh, dữ liệu âm thanh ho dưới dạng âm thanh sẽ được chuyển đổi thành 1 hình ảnh quang phổ từ máy tính, VD: Sử dụng phép biến đổi Fourier thời

gian ngắn (STFT). Sau đó sử dụng mạng học sâu để phân tích dựa trên hình ảnh quang phổ được chuyển đổi từ âm thanh ho và các âm thanh thông thường. Ưu điểm của việc này là sử dụng Mạng học sâu tích chập (CNN) để áp dụng vào việc nghiên cứu và phát hiện các dạng ho qua hình ảnh, CNN rất dễ huấn luyện và có sẵn rất nhiều phần cứng tài nguyên để hỗ trợ cho việc này. Cùng với đó, áp dụng đồng thời Mạng học sâu hồi quy (RNN) với các nơ-ron chuyên biệt có thể nắm bắt và mô hình hóa các liên kết dài hạn theo trình tự. Bên cạnh đó, RNN cũng khó bị ảnh hưởng bởi hiện tượng nhiễu trong dữ liệu tuần tự. Tuy nhiên, CNN có một nhược điểm lớn đó là yêu cầu dữ liệu đầu vào phải được cố định và rõ ràng, cùng với đó việc học dựa trên RNN là quá trình huấn luyện máy học vô cùng khó khan và lượng mẫu dữ liệu đầu vào là không hề nhỏ. Song song với đó, cũng đó có một số phương án khác như [15] cũng đã sử dụng mạng học sâu để nhận dạng loại ho của bệnh nhân, họ đã chứng minh rằng phân loại ho dựa trên sự kết hợp của mô hình hỗn hợp Gaussian kết hợp với Markov ẩn (GMM – DNN) không thể hoạt động tốt bằng hệ thống sử dụng mạng học sâu cùng mô hình Markov ẩn (HMM – DNN). Ta có thể thấy được bằng cách sử dụng phương pháp học sâu, ta có thể xây dựng được một hệ thống nhanh nhạy và chuẩn xác theo thời gian thực, tuy nhiên để đạt được hệ thống như vậy chúng ta cần một lượng vô cùng lớn các dữ liệu đầu vào để hệ thống máy học có thể sử dụng để tăng khả năng chuẩn đoán. Bênh cạnh đó, toàn bộ các dữ liệu này cần được ghị, thu thập liên tục và không được ngắt quãng.

Con người đã rất thành công khi sử dụng thành thạo và kết hợp các thiết bị như các bộ xử lý, bộ nhớ đệm và nhiều loại cảm biến như gia tốc kế, GPS, con quay hồi chuyển, máy ảnh kỹ thuật số, micro,… để áp dụng vào việc phát hiện, chữa trị các căn bệnh trong những khoảng thời gian không có sự giám sát của bác sĩ, y tá hay các chuyên gia y tế. Bên cạnh đó, hiện nay các thiết bị di động cũng là một trong các thiết bị được nhiều nhà phát triển, khoa học quan tâm tới khi xây dựng các hệ thông phần mềm, ứng dụng phục vụ cho việc phát triển y học và hỗ trợ bệnh nhân. Bằng cách sử dụng thiết bị di động hay các thiết bị IoT, người sử dụng có thể nắm rõ các thông tin tình trạng cá nhân nhanh chóng và đưa ra các quyết định kịp thời. Hiện nay

cũng đã có một số các hệ thống ứng dụng đã có thể phát hiện và phân loại ho bằng các phân tích các dữ liệu âm thanh được thu vào từ các thiết bị di động. Ví dụ: [1] đã đề xuất một phương án phát hiện âm thanh ho dựa trên việc phân tích cường độ âm thanh trung bình, các tính chất thành phần của âm thanh ho đã được trích xuất từ các hệ số biến đổi Fast Fourier của dữ liệu thô thu từ thiết bị di động được đặt trong tay túi áo của người sử dụng hay từ thiết bị đeo cổ tích hợp míc như các thiết bị tai nghe không dây (phần micro hướng về phía miệng của người sử dụng). Kết quả chỉ ra rằng [1] đã đạt tỉ lệ chính xác tới 92%, trong khi đó cách làm này có thể giữ được quyên riêng tư của người sử dụng, nhưng việc sử dụng thiết bị đeo cổ cũng gây ra sự bất tiện cho người sử dụng. Chúng ta có thể thấy được việc sử dụng các cảm biến âm thanh chất lượng cao có thể giúp chúng ta tăng độ chính xác khi nhận dạng âm thanh ho thu được, tuy nhiên có một sự thật là để có thể sử dụng được nhiều cảm biến như vậy ta sẽ mất rất nhiều chi phí, tài nguyên sử dụng, không những vậy các thiết bị này cũng có phạm vi tín hiệu nhất định đôi khi sẽ gây sự khó chịu cho người sử dụng. Đối với thời đại công nghệ 4.0 cùng với tình trạng giãn cách xã hội như hiện nay, việc sử dụng các thiết bị IoT như di động là một phương án tiện nghi và tương đối hữu hiệu đối với những người đã có biểu hiện nghi nhiễm hay những người cần có sự giám sát y tế. Phương pháp đề xuất của tôi cũng được xây dựng dựa trên một thực trạng không thể phủ nhận sự nguy hiểm của dịch bệnh không chỉ tại Việt Nam mà trên cả trên toàn thế giới. Bằng cách sử dụng phương pháp này, ta không chỉ phát hiện âm thanh ho mà còn phận biệt, đánh giá các dạng ho thông qua dữ liệu được thu từ các thiết bị IoT của người sử dụng trên các nền tảng hiện hành đang được phát triển.

Phát hiện ho dựa trên các cảm biến đã được nghiên cứu qua hàng chục năm nay để có thể đưa ra các kết quả chính xác. Tuy vậy, chỉ phát hiện cho là chưa đủ cho cho nhu cầu sử dụng như hiện tại, chúng ta có thể nhận dạng được các dạng ho phổ biến gôm 5 dạng ho xuất hiện trên người [12]. Với mỗi dạng ho, nhưng với các triệu chứng khác nhau lại có thể đưa ra một kết quả lâm sàng khác về bệnh của người đang mắc phải triệu chứng này. Vậy nên, việc có thể phân loại ho là bước đầu tiên để có thể phát hiện kịp thời đến các bệnh liên quan tới đường hô hấp. Để có thể xác nhận

được kiểu dạng ho hay chủng loại ho thì đòi hỏi người mắc triệu chứng trên cần phải có sự phối hợp với bác sỹ chuyên ngành. Đối với thực trạng như hiện nay đôi khi sẽ có nhiều bệnh nhân khi mắc các triệu chứng về ho cũng sẽ chưa vội liên hệ với bệnh viện, tự đánh giá triệu chứng của bản thân và đưa ra quyết định. Đôi khi với chính những suy nghĩ này đã góp phần tăng thêm số lượng người nhiễm phải dịch bệnh này tăng cao. Hay người bệnh nhân khi đến các bệnh viện chuyên ngành lo ngại cũng có thể nhiễm phải dịch bệnh này kể cả không tiếp xúc trực tiếp, vậy nên xây dựng một hệ thống phát hiện và phân loại các dạng ho ngay trên chính các thiết bị di động là một việc vô cùng quan trọng và cần thiết. Chúng ta đã biết rằng nếu sử dụng các thiết cảm biến và biểu đồ ho theo thời gian thực của người bệnh, ta sẽ nhận biết được và phân loại các dạng ho này. Nhưng song song với việc này, chúng ta sẽ phải tốn nhiều kinh phí hơn để sử dụng các thiết bị cảm biến với số lượng lớn với lượng người nhiễm bệnh là quá nhiều, không những vậy các thiết bị này cũng khiến chúng ta mất nhiều thời gian hơn để chuẩn đoán mà còn gây ra sự khó chịu đối với các bệnh nhân (có thể có những người không muốn sử dụng các thiết bị này). Ngược lại đối với các thiết bị IoT, hay các thiết bị di động thì giờ đây là một phần gần như không thể thiếu được với con người hiện đại. Hiện nay, theo như một nghiên cứu đã đưa ra rằng người Việt Nam bỏ ra trung bình 4 tiếng mỗi ngày để sử dụng các thiết bị di động và số người này còn tăng lên khi thống kê với các nước đang phát triển. Chính nhờ sự thông dụng của các thiết bị IoT ngày càng được phát triển và nâng cấp như điện thoại thông minh, thiết bị đeo tay thông minh,… thì việc phát triển hệ thống hỗ trợ người mắc bệnh trên các thiết bị này là vô cùng tiềm năng, có khả năng sẽ đạt được hiệu suất cao với nhiệm vụ được đề ra ban đâu. Bằng các thiết vị IoT, chúng ta có thể thu được các dữ liệu thông tin về âm thanh ho, âm lượng, tần suất triệu chứng của bệnh nhân một cách thụ động và đưa ra được biểu đồ về dạng ho của người mắc triệu chứng này. Các nhà phát triển đã hướng tới việc xây dựng các thiết bị IoT thành một hệ sinh thái chung, dễ dàng chia sẻ các thông tin nhận được giưa chúng qua nền tảng bảo mật cụ thể. Trung nghiên cứu [13] đã đề cập đến việc sử dụng thiết bị đeo tay, và điện thoại di động chia sẻ thông tin cho nhau có thể nhận ra những tác động của cơn ho đến với cơ thể

người trong thời gian thực. Như vậy, ta có thể thấy được sự quan trọng của các thiết bị IoT và bằng cách sử dụng chúng ta có thể giúp nhiều người có thể phát hiện và phân loại ho không chỉ ở người già, trẻ em mà những người khỏe mạng cũng có thể nắm được tình trạng cơ thể mình và mau chóng có biện pháp phòng chống, phòng ngừa hiệu quả, tránh được sự lây lan đáng sợ của các dại dịch nguy hiểm liên quan tới được hô hấp như đại dịch COVID-19 hiện nay.

## 1.3 Các dạng ho dựa trên bệnh lý con người

Ho là một phát thở ra mạnh và đột ngột. Đó là cơ chế tự vệ sinh lý để đưa các dị vật được phát hiện ở phần trên của đường hô hấp có thể gây tắc thở ra phía bên ngoài. Ho cũng là một trong những triệu chứng của việc rối lại hệ thống tuần hoàn trong cơ thể.

Người ta có thể chủ động ho, nhưng trong số đa trường hợp, ho xảy ra ngoài ý muốn và động tác này có tính phản xạ. Tuy nhiên, nhiều virus và vi khuẩn có thể truyền nhiễm từ người này sang người khác thông qua ho.Tại nghiên cứu [19], các nhà chuyên môn đã đưa ra được nghiên cứu cụ thể về triệu chứng này và đưa ra các dạng của triệu chứng ho bên cạnh đó là nguyên nhân lây nhiễm từ ho (Bảng 1.1) và không lây nhiễm từ ho (Bảng 1.2):

- Ho cấp: Là tình trạng ho xảy ra đột ngột, thông thường nhất là do hít phải bụi hoặc chất kích thích. Ho cấp cũng có thể là do các nguyên nhân: Do nhiễm khuẩn, viêm họng, viêm thanh quản, viêm tai, viêm xoan, viêm phế quản, viêm phổi, tràn dịch màng phổi. Cũng có khi các triệu chứng ho xuất hiện trong các bệnh dị ứng tai mũi họng và hen. Triệu chứng ho cấp cũng có thể là do bênh gây ứ máu ở phổi như: Bệnh phù phổi, tim và thường gặp ở người có tiều sử mắc các bệnh tim mạch, tăng huyệt áp,…

- Ho thành cơn: Ho nhiều lần liên tiếp nhau trong một thời gian ngắn, điển hình như là cơn ho gà; người bệnh ho liền một cơn, sau đó hít một hơi dài và tiếp tục ho nữa. Cơ ho kéo dài thường gây gia tăng áp lực trong lồng ngực, gây ứ huyết tĩnh mạch chủ trên làm cho người bệnh đỏ mặt, tĩnh mạch cổ phồng, cơn ho có thể làm

chảy nước mắt, đôi khi còn gây ra phản xạ nôn nữa. Người bệnh có thể đau ê ẩm ngực, lưng và bụng do các cơ hô hấp co bóp quá mức.

- Ho khan kéo dài: là tình trạng ho không khạc ra đờm mặc dù người bệnh có thể ho nhiều. Tuy nhiên, có người nuốt đờm hoặc vì không muốn khạc hay vì không biết khạc đờm. Ho khan kéo dài cần chú ý đến: Bệnh của thanh quản, viêm tai, viêm xương chũm mạn tính; Do ung thư phế quản: xảy ra ở người có thâm niên hút thuốc lá, thuốc lào lâu năm (trên 10 năm).

Ho khan kéo dài còn do các bệnh tổ chức kẽ của phổi như xơ phổi, phù phổi bán cấp, ung thư phổi hoặc lao kê hoặc do tràn dịch mạn tính màng phổi. Ho kéo dài cũng có thể do một số chất độc gây kích thích trực tiếp do cơ chế miễn dịch dị ứng (hen). Một số trường hợp rối loạn tinh thần có biểu hiện ho nhiều, không có tổn thương trên đường hô hấp. Nhưng đó là những trường hợp hiếm gặp. Ho khan kéo dài còn do tác dụng phụ của một số thuốc, nhất là thuốc điều trị bệnh tăng huyết áp (coversyl).

- Ho có đờm: Là tình trạng người bệnh bị ho và cảm thấy nặng ngực, ho thường khạc ra chất nhầy và đờm. Bệnh nhân có cảm giác nghẹt thở và khó thở, mệt lả. Các triệu chứng thường tăng lên khi đi bộ và nói chuyện. Ho có đờm đa số nguyên nhân là do viêm phế quản mạn tính, cũng có khi là triệu chứng ho sau khi viêm họng, viêm mũi và viêm xoang...

Ở một người nghiện thuốc lào, thuốc lá lâu năm bị ho có đờm kéo dài, đồng thời tính chất của ho thay đổi hoặc ho ông ổng là dấu hiệu của báo động ung thư phế quản. Ho khạc đờm nhiều kèm theo bội nhiễm luôn phải chú ý đến ung thư họng - thanh quản, thực quản, khí quản...

- Ho ra máu: Là tình trạng ho khạc thấy máu xuất hiện kèm theo. Nó có nhiều mức độ từ nhẹ đến nặng. Đó có thể là một dấu hiệu của các bệnh viêm phổi cấp và mạn tính, ung thư phổi... Ho ra máu có thể xảy ra đột ngột trong lúc người bệnh cảm thấy khỏe mạnh hoặc sau khi hoạt động mạnh... Thông thường, 90% trường hợp ho ra máu là do bệnh lao đang tiến triển (nếu kèm ho kéo dài, sốt nhẹ, sút cân thì càng

chắc chắn). Nếu ho ra máu chút ít lẫn trong đờm, tái phát một vài lần mà không có sốt hoặc sút cân cũng nên nghĩ đến bệnh lao.

| | |
|---|---|
| Các loại thuốc | Inhibitors of the converting enzyme of the angiotensin |
| | Beta blockers |
| | Interferon peguilado (bronchial mod) |
| | Methotrexate (pneumonitis) |
| Bệnh tim mạch | Pulmonary Edema |
| | Pulmonary Embolism |
| Dạ dày trào ngược | |
| Tắc, nghẹt thở do ngoại vật tác động | |
| Các loại u khối | |
| Hen suyễn | Variant of the asthma with cough[(a)] |
| Bệnh phổi tác nghẽn thâm niên | |
| Hít phải khí độc | Gas mustard, formaldehyde |
| Dị ứng | Silicosis |
| Hiệu ứng ho | After-infectious coughAtopic Cough [(b)]Psychogenic Cough |

Bảng 1.1: Các nguyên nhân hình thành ho không do lây nhiễm

| | | |
|---|---|---|
| Nhiễm các chủng virus | Cảm lạnh | adenovirus, coronavirus, enterovirus, parainfluenza |
| | Cúm | virus influenza A e B |
| | Bronchiolitis | respiratory synctial virus (VSR) |
| | Tranqueobronquitis acute | virus influenza, VSR |
| | Hantavirus | virus Juquitiba, Araraquara, Castelo dos Sonhos, Laguna Negra, Anajatuba |
| Vi khuẩn | Ho gà | Bordetella pertussis |
| | Tranqueobronquitis acute | Mycoplasma pneumoniae |
| | Rinosinusites (syndrome of the cough of the by airmail superior one) | Streptococcus pneumoniae<br>Haemophilus influenzae<br>Moraxella catarrhalis |
| | Vi khuẩn Pneumonia | Streptococcus pneumoniae<br>Mycoplasma pneumoniae<br>Chlamydophila pneumoniae<br>Haemophilus influenzae |
| | Mycobacteriosis typical and atypical | Mycobacterium tuberculosis |
| Ký sinh trùng | Eosinophilia pulmonary parasitic (Syndrome of Loeffler) | Ascaris lumbricoides<br>Ancylostoma duodenale<br>Strongyloides stercoralis |
| | Chronic Schistosomiasis Pulmonary | Schistosoma mansoni |
| | Larva migrans visceral | Toxocara canis, Toxocara cati |
| | Singamus | Syngamus laryngeus |
| Động vật nguyên sinh | Visceral Leishmaniasis | Leishmania chagasi |
| Nấm | Aspergillosis | Aspergillus spp |
| | Blastomycosis | Blastomyces dermatitidis |
| | Cryptococcosis | Cryptococcus neoformans |
| | Histoplasmosis | Histoplasma capsulatum |
| | Paracoccidioidomycosis | Paracoccidioides brasiliensis |
| | Pneumocystosis | Pneumocystis jiroveci |

Bảng 1.2: Các nguyên nhân hình thành ho do lây nhiễm

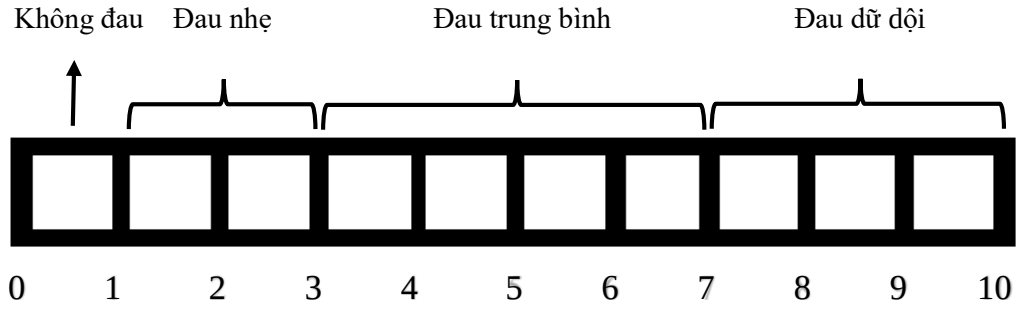
## 1.4 Kết luận

Như vậy, chúng ta có thể thấy được nguy hiểm của các triệu chứng ho, sự cấp thiết của việc đề xuất áp dụng trí tuệ nhận tạo trên các thiết bị IoT để phát hiện và phận loại các dạng ho là vô cùng cần thiết. Bằng việc sử dụng các thiết bị IoT, chúng ta có thể tiếp cận đến người mắc các căn bệnh liên quan tới đường hô hấp hay cụ thể là các triệu chứng ho để đưa ra các kết luận ban đầu về dạng ho của người đang mắc bệnh. Đặc biệt đối với thực trạng hiện nay rằng, dịch bệnh COVID-19 vẫn đang không có dấu hiệu dừng lại trên toàn thế giới thì việc, mỗi người tự trang bị có mình các thông tin cũng như hệ thống nhận dạng, phân biệt chủng ho này sẽ góp phần vào quá tải ở các bệnh viên, giảm thiểu số lượng người nhiễm bệnh hay có để đầy lùi được không chỉ dịch COVID-19 mà còn toàn bộ các dịch bệnh nguy hiểm liên quan tới được hô hấp.

# CHƯƠNG 2: PHƯƠNG PHÁP PHÂN LOẠI HO

## 2.1 Xử lý âm thanh ho

Ho là một triệu chứng thường gặp trong các bệnh liên quan tới được hô hấp. Đó là một phản xạ bảo vệ giúp cơ thể con người thải ra các các chất bài tiết trong đường hô hấp, bảo vệ đường khí di chuyển trực tiếp tới phổi, ví dụ như: đờm, các ngoại vật, các ký sinh hay vi khuẩn có hại,… Trong việc điều trị các bệnh liên quan tới triệu chứng ho, mức ho là yếu tố cần thiết để theo dõi tiến trình phát triển của bênh nhân. Trong các nghiên cứu làm sàng hiện tại, việc đo lường mức độ ho chủ yếu dựa trên các thang điểm tự đánh giá, các thang điểm này tương tự như thang điểm thể hiện mức độ đau của bệnh nhân VAS cùng với các câu hỏi liên quan tới cuộc sống thường ngày của người bị nhiễm bệnh.

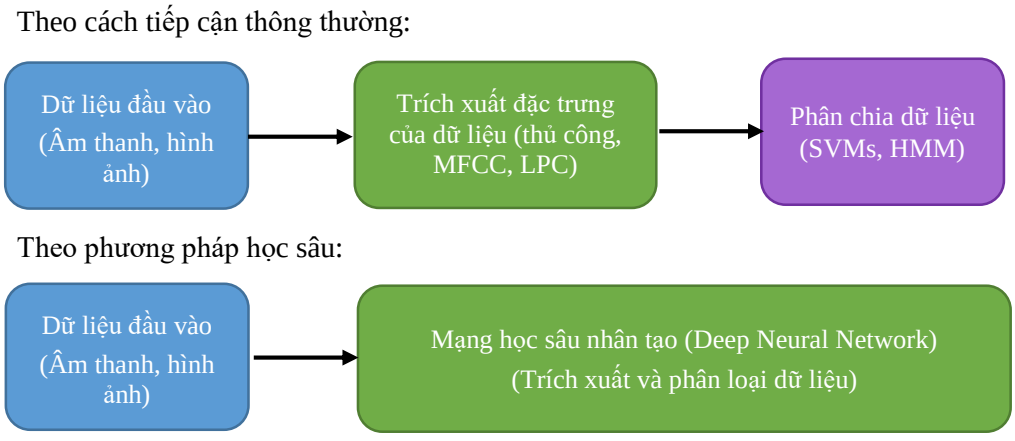



Hình 2.1 Thang điểm thể hiện độ đau (VAS)

Các thang điểm này thu thập thông tin từ các cảm giác chủ quan của bệnh nhân về mức độ năng như của triệu chứng ho. Mặc dù được sử dụng khá là phổ biến trong việc xác định triệu chứng ho, nhưng những thang điểm chủ quan này dễ bị ảnh hưởng bởi chính tâm lý của bệnh nhân, có thể tạo ra chuẩn đoán sai lệnh về triệu chứng ho và hiệu ứng giả dược. Vậy nên, giải pháp để đo lường, chuẩn đoán các triệu chứng ho một cách chính xác, chúng ta sử dụng một thiết bị theo dõi ho như đã nói ở phía trên đó là việc tích hợp một hệ thống phát hiện và phân loại ho ngay trên chính các thiết bị IoT.

Tần suất và cường độ ho của bênh nhân được thu lại có thể là công cụ chuẩn đoán hữu hiệu và có thể giúp cho việc điều trị cho những người có bệnh lý về hô hấp

như hen suyễn, lao và viêm phổi. Mặc dù bệnh nhân có xu hướng cung cấp các tiểu sử thông tin không đang tin cậy, thường bịa đặt hay ghi nhớ không chính xác, cùng với việc sử dụng hệ thông ho tích hợp trên các thiết bị IoT thì hệ thông sẽ thu thập và phân tích các thông tin này. Có một điều đang chú ý đối với việc thiết kế một hệ thống phát hiện ho tự động phải có một độ chính xác cao để từ đó có thể đưa ra chuẩn đoán chính xác. Vì ho là những trường hợp tương đối hiếm, nên ngày cả một máy dò ho có độ đặc hiệu trung bình cũng sẽ tạo ra 1 số lượng lớn các thông tin mang tính sai lệch cao và không thể đưa ra thông tin chính xác. Để có thể đạt được độ đặc hiệu chuyển đổi cao là không hề dễ dàng, bởi vì tiếng ho có cùng đặc tính âm thanh với các âm thanh khác như tiếng hắng giọng, hắt hơi, cười và thậm chí là cả tiếng nói giao tiếp. Ngoài độ đặc hiệu cao, các thiết bị thu tiếng ho cũng cần phải có độ nhạy tốt để có thể kịp thời thu được thông tin về triệu chứng ho của người bị nhiễm bệnh. Có một số phương pháp phổ biến có thể giải quyết các vấn đề liên quan việc phân biệt âm thanh ho và các âm thanh không phải tiếng ho. Các nhà nghiên cứu trước đây đã xem xét các phương pháp nhận dạng âm thanh giọng nói cổ điển như Hệ số phân tích tần số sóng âm (MFCC) và hệ số mã hóa phân tích dự đoán tuyến tính (LPC) [20], [21]; sự thích ứng của việc gán các chức năng nhận dạng giọng nói [22]; và các tính năng thủ công được thiết kế riêng [23].




Hình 2.2: Tổng quan về phương pháp hiện ho thông thường so với phương pháp học sâu.

Tại hình 2.2, Ta có thể thấy được phương pháp học sâu loại bỏ tính năng thu thập thông tin thủ công bằng cách tự động học và khai thác thông tin từ thông tin tho ban đầu được đưa vào. Điểm đặc biệt của việc huấn luyện và tùy chỉnh các tính năng của hệ thống đã khiến cho chúng có thể tận dụng các điểm đặc biệt của âm thanh triệu chứng ho mà có thể không rõ ràng so với sự trùng lặp đặc điểm với các âm thanh không phải là ho. Đi kèm với đó chúng ta cần 1 thiết bị IoT chất lượng phải có micro chất lượng, có khả năng sử dụng lâu dài (kích thước, đem lại sự thoải mái cho người sử dụng, dung lượng pin, tính bảo mật thông tin cao) để có thể đem lại chất lượng thông tin cho việc chuẩn đoán là tốt nhất. Bên trong đó sẽ được tích hợp một thống phát hiện và phân loại âm thanh triệu chứng ho sử dụng mạng học sâu phức hợp để tự động xác định một tập hợp các tính năng được tùy chỉnh thích hợp [24]. Ở đây, chúng ta sử dụng phương phá tiếp cận mạng nơ-ron nhân tạo để xác định các cơn ho một cách rộng rãi hơn bằng cách giải quyết các câu hỏi (1) cấu trúc xử lý tín hiệu nào phù hợp để phân tích  một sự kiện triệu chúng ho tiềm năng, (2) ảnh hưởng của số lớp và số lượng nơ-ron trong một mạng hệ thống và (3) tác động sự phụ thuộc của tín hiệu dài hạn trong hiệu suất của hệ thông phân loại ho.

## 2.2 Mô hình máy học Gaussian hỗn hợp (GMM) cho phát hiện và phân loại ho

Mặc dù có nhiều hệ thống đã được phát triển, tuy nhiên vẫn chưa có hệ thống tối ưu thực sự cho việc phát hiện và phân loại ho. Vấn đề được đưa ra ở đây phần lớn là về khả năng đưa ra chuẩn đoán về triệu chứng ho có độ chính xác không cao. Trong khi đó, hầu hết các nghiên cứu đưa ra đều chưa được thẩm định trên quy mô lớn, dẫn đến kết quả chưa có tính thuyết phục cao. Hệ thống phổ biến nhất, LCM [2], tìm ra sự cân bằng giữa hiệu suất công việc và số lượng người thực hiện. Trong bước phân loại của mình, họ đã thuê các nhà nghiên cứu cơ thể người để ghi ra nhãn tất cả các hiện tượng của cơn ho, không phải âm thanh ho và các đặc tính của dạng âm thanh ho và sau đó được đưa vào mô hình của HMM-GMM để phân loại them. Hiệu suất của công việc này đủ khả năng để đáp ứng lại các ứng dụng trong thực tế. Tuy nhiên, hiệu suất của các mô hình HMM-GMM đã được cải thiện nhiều hơn qua các bước

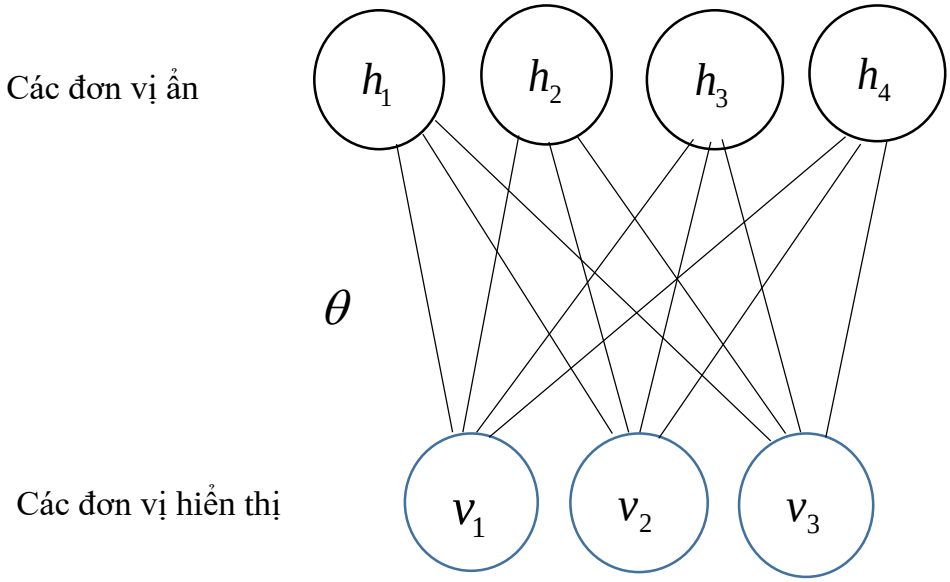
đột phá trong những năm gần đây trong lĩnh vực nhận dạng giọng nói tự động bằng cách thay thế GMM bằng một mô hình mới được phát triển, mạng nơ-ron sâu (DNN) để lập mô hình xác suất quan sát của HMM. Bằng cách huấn luyện các mô hình mạng nơ-ron để phân biệt, thông tin từ các khung sẽ được huấn luyện tốt hơn. Lợi ích của việc sử dụng mạng nơ-ron đã được biết đến từ lâu, nhưng việc huấn luyện cho một mạng nơ-ron sâu là không khả thi cho đến khi phương pháp pretrain được giới thiệu gần đây. Pretrain có thể dịch là tinh chỉnh, mô hình đã được huấn luyện trước đó với một bộ dữ liệu lớn hoặc với các phương pháp tối tân giúp giảm công đoạn khi huấn luyện lại mô hình từ đầu. Mô hình sau đó có thể được huấn luyện để phù hợp với bộ dữ liệu thực tế hoặc sử dụng trong bài toán học máy. Đối với vấn đề phát hiện và phân loại ho, thì điều bắt buộc cần phải làm đó là chuyển đổi mô hình phân loại thành mạng nơ ron sâu. DNN là một mô hình có khả năng học tập mạnh mẽ có thể thay thế GMM khi phân loại các dữ liệu âm thanh ho. Các tín hiệu thô được sử dụng để phân loại các âm thanh ho là các bản ghi âm, được mô hình hóa một cách tự nhiên bằng hệ thống mạng thần kinh sâu.

Trước đây, những trở ngại cho việc huấn luyện một mạng nơ-ron nhân tạo bao gồm 2 khía cạnh quan trọng. Một là tài nguyên để sử dụng thời điểm đó là không nhiều. Vấn đề còn lại là về vấn đề suy biến đạo hàm đã làm các hệ thống kém được huấn luyện đúng cách. Trở ngại đầu tiên đã được xử lý bằng chắc phát triển các thiết bị tính toán ngày càng tốt hơn. Với vấn đề thứ hai đã được giải quyết bằng sự ra đời của phương pháp Pretrain với mạng lưới niềm tin sâu (DBN) được cầu thành từ các Restricted Boltzmann Machines (RBM).

### *2.2.1 Restricted Boltzmann Machine*

Restricted Boltmann Machine (RBM) là một mạng thần kinh ngẫu nhiên. Giả định cơ bản của RBM là tính năng mà tôi quan sát được điều khiển bởi nhiều các yếu tố cấp cao, do đó các yếu tố cấp cao có thể được sử dụng làm các tính năng có mức trìu tượng cao hơn. Nó cũng là một dạng của trường tuyến tính ngẫu nhiên Markov. Các đơn vị có thể nhìn thấy $v$ đại diện cho vector đối tượng thô ban đầu, trong đó $v_j$

là giá trị cả đối tượng thứ i. Các yếu tố cấp cao hơn sẽ được mà hóa trong vector $h$ trong đó $h_i$ là yếu tố thứ $i$. Các mục của RBM sẽ được kết nối qua các trọng só. Về mặt trực quan thì các trọng số sẽ xác định mối tương quan giữa các nút. Trọng số lớn hơn có nghĩa là khả năng lớn các nút sẽ được kết nối tốt. Trong RBM, chỉ có các khối hiển thị và các khối ẩn mới được phép kết nối, đó chính là lý do vì sao nó được gọi là bị hạn chế (Restricted). Một ví dụ đơn giản của của RBM được hiển thị trong hình 2.3.




Hình 2.3: Ví dụ đơn giản của RBM với 4 khối ẩn và 3 khối hiển thị

$$p(v,h,\theta) = \frac{1}{Z(\theta)} e^{-E(v,h,\theta)} \tag{1}$$

$$Z(\theta) = \sum_u \sum_h e^{-E(u,h)} \tag{2}$$

Các nút của RBM được liên kết với các giả định khác nhau để phù hợp với các vấn đề khác nhau. Như trường ngẫu nhiên Markov, các hàm năng lượng tiền năng khác nhau dựa trên các giả định khác nhau. Đối với mỗi cấu hình của tất cả các nút, khả năng của cấu hình đó được xác định thêm bởi hàm thế năng bằng cách chia hàm

phân hoạch là (1) và (2), trong đó u và h đại diện cho tất cả các cầu hình có thể có của các đơn vị khả thi và ẩn. Trong RBM nhị phân, cả các nút trực quan và nút ẩn đề có các giá trị nhỉ phân. Cơ năng được xác định ở (3). Xác xuất có thể dễ dàng suy ra từ (1) và (2).

$$E(v,h,\theta) = -\sum_{i=1}^{V}\sum_{j=1}^{V} W_{ij} v_i h_j - \sum_{i=1}^{V} b_i v_i - \sum_{j=1}^{H} a_j h_j \tag{3}$$

Với xác suất chung, chúng ta có thể suy ra thêm phân phối xác suất cận biên được gán cho vector khả kiến $v$ bằng cách lấy biên $h$. Phân phối biên được thể hiện trong (4). Các tham số của RBM sau đó có thể huấn luyện bằng phương pháp giảm độ dốc. Các đạo hàm của hàm khả năng liên quan tới tham số $W$, $v$ và $h$ được thể hiện trong phương trình (5), (6), (7).

$$p(v \mid \theta) = \frac{1}{Z(\theta)} \sum_{h} e^{-E(v,h)} \tag{4}$$

$$\frac{\partial \log P(M;\theta)}{\partial W} = \sum_{v \in M} (E_{p(h|v)}[vh^T] - E_{P(h|v)}[vh^T]) \tag{5}$$

$$\frac{\partial \log P(M;\theta)}{\partial a} = \sum_{v \in M} (E_{p(h|v)}[h] - E_{P(h|v)}[h]) \tag{6}$$

$$\frac{\partial \log P(M;\theta)}{\partial b} = \sum_{v \in M} (E_{p(h|v)}[v] - E_{P(h|v)}[v]) \tag{7}$$

Trong đó, M là một tệp dữ liệu thô. EP có nghĩa là kỳ vọng phân phối P. P(h|v) là phân phối của các đơn vị ẩn cho vector nhìn thấy được và quan sát. P(h,v) là phân phối chung của các đơn vị quan ẩn và hiển thị. Xác suất có điều kiên P(h|v), được mô tả là (8), (9). Từ các phân phối, tôi có thể tìm thấy kỳ vọng của các xác suất có điều kiện và rất dễ tính toàn. Tuy nhiên, để tính kỳ vọng xác suất chung P(h|v), chúng ta cần lấy mẫu Gibbs bắt đầu từ một vector ngẫu nhiên, các đơn vị mẫu ẩn và các đơn

vị hiển thị lặp đi lặp lại trong một khoảng thời gian dài. Để tăng cường tính toàn các gradient, một quy trình có tên là "Phân kỳ tương phản (CD)" có thể được sử dụng để ước tình các gradient bằng cahs thay thể kỳ vọng chung bằng "tái tao" các đơn vị ẩn và hiển thị. Quy trình chung của CD là vector dữ liệu $v$ làm điểu ban đầu của bộ lấy mẫu trước khi lấy mẫy đệ quy, các đơn vị ẩn và đơn vị hiển thị dựa trên phân phối có điều kiện. Thông thường, $v$ và $h$ sau khi lấy mẫu một lần mà đủ tốt để cập nhật các tham số và các cài đặt sau đó sẽ đặt tên là CD1. Phương pháp này hiệu quả hơn nhiều so với việc lấy mẫu Gibbs.

$$p(h \mid v;\theta) = \prod_j P(h_j \mid v) = \prod_j Ber(h_j \mid \delta(\sum_i W_{ij} v_i + a_i)) \tag{8}$$

$$p(v \mid h;\theta) = \prod_j P(v_j \mid h) = \prod_j Ber(v_j \mid \delta(\sum_i W_{ij} h_i + a_i)) \tag{9}$$

Các đơn vị của RBM có thể kết hợp với các loại giá trị khác nhau. Đối với dữ liệu giá trị thực, Gaussian – Bernoulli RBM là mô hình luôn được sử dụng. Hàm thế năng của Gaussian – Bernoulli RBM được biểu diễn trong (10). Chung ta có thể dễ dàng nhận được phân bố xác suất cho RBM Gaussian – Bernoulli với hàm năng lượng theo các bước tương tự như RBM nhị phân. P(h|v) ở dạng khác với RBM nhị phân, được biểu diễn trong (10).

$$E(v,h,\theta) = -\sum_{i=1}^{D}\sum_{j=1}^{F} W_{ij} h_j \frac{v_i}{\sigma_i} - \sum_{i=1}^{D} \frac{(v_i - b_i)^2}{2\sigma_i^{\,2}} - \sum_{j=1}^{F} a_j h_j \tag{10}$$

$$p(v \mid h;\theta) = \prod_i \mathrm{N}(v_i \mid \sigma \sum_j W_{ij} h_j + b_j, \sigma) \tag{11}$$

Quy tắc có thể cập nhật dễ dàng theo các bước chính xác tương đối với RBM nhị phân. CD1 cũng là một phương pháp hiệu quả để huấn luyện Gaussian – Bernoulli RBM.

### *2.2.2 Mạng học sâu (DNN)*

Trong nguyên mẫu của mạng học sâu (DNN), RBM được sử dụng để học cách khởi tạo DNN. RBM nhiều lớp, còn được gọi là Mạng học sâu Bayes (DBN), có thể được huấn luyện bằng thuật toán tiếp cận tham lam. Trong lớp đầu tiên, Sau khi được huấn luyện tốt, kỳ vọng của các đơn vị ẩn được sử dụng làm vector rữ liệu cho RBM lớp thứ hai. Các lớp trên của RBM đều được huấn luyện theo uy trình này. Sau cùng, các RBM xếp chồng lên nhau tạo thành một DBN. Các bước huấn luyện cho DBN cũng được trình bày trong bảng 2.1.

| |
|---|
| Đầu vào: Dữ liệu D = {x}, lớp K mong muốn và số nút cho mỗi lớp $N_i$ |
| Đầu ra: Cấu trúc và các tham số khởi tạo đã được đào tạo của DNN |
| 1. Học các tham số $\theta_1$ cho lớp dữ liệu đầu tiên của RBM<br>Với k = 2:K<br>2. Khởi tạo RBM lớp thử k bằng cách mở từng lớp RBM, với tham số $W_k = W_{k-1}^T$<br>3. Tinh chỉnh các tham số của RBM ở lớp thứ k bằng các vector dữ liệu được tạo ra từ lớp thứ k-1 |

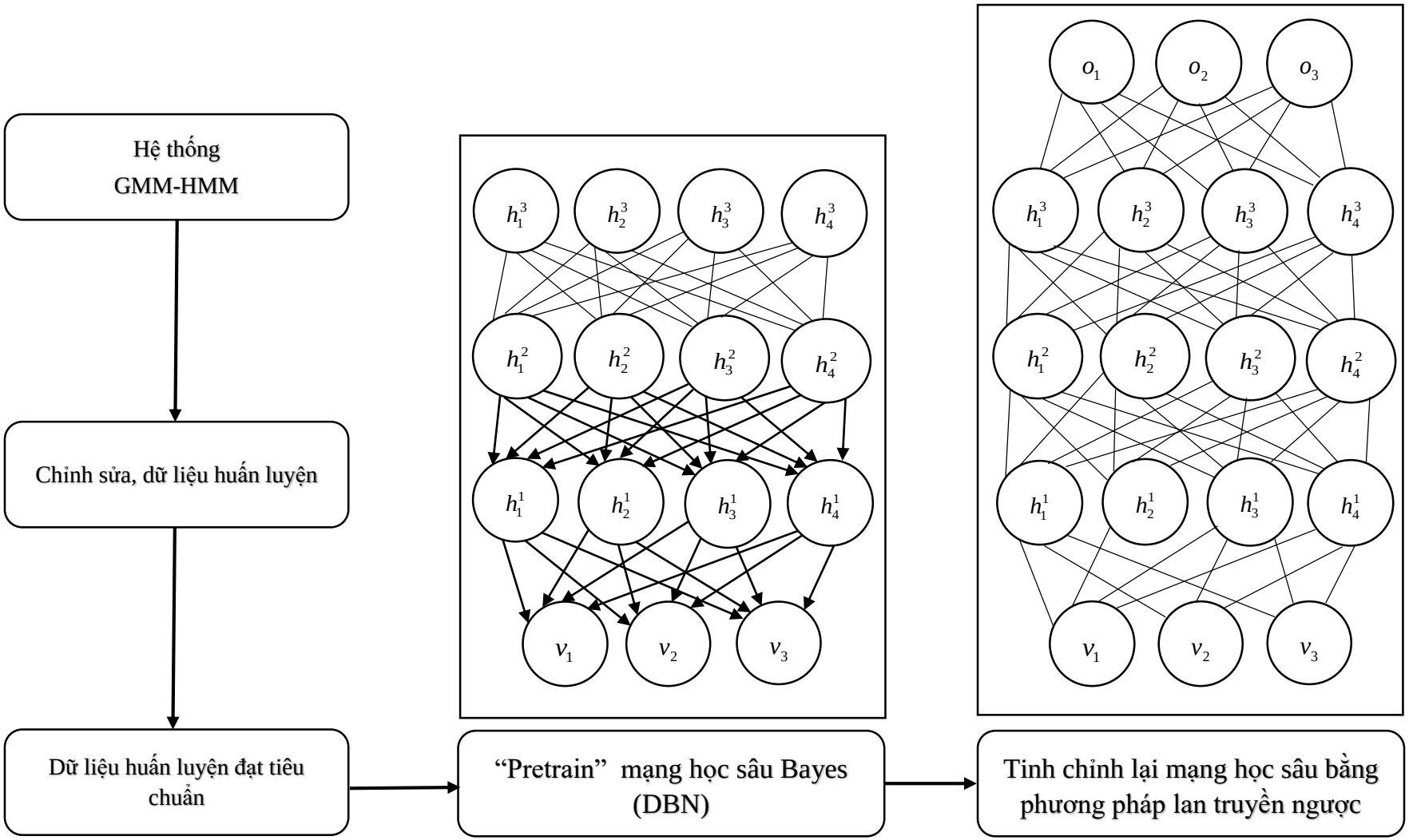
Bảng 2.1: Mô tả thuật toán huấn luyện Mạng học sâu Bayes

Sau khi huấn luyện DBN, các tham số của nó được sao chép vào một mạng nơ-ron có cùng cấu trúc. Lớp đầu ra của DNN là một lớp softmax được xếp chồng lên nhau. Quá trình huấn luyện DBN và sao chép tham số của nó được gọi là "pretrain". Mạng nơ-ron khởi tạo được huấn luyện thêm bằng thuật toán truyền ngược. Bước huấn luyện thêm này được gọi là "tinh chỉnh". Bước pretrain giúp cho DNN theo một số cách. Nó buộc DNN phải ánh xạ không gian đối tượng thôi tới mức trìu tượng cao hơn, không chỉ để học phản ứng vô hướng đối với không gian đầu ra, nó thực sự hoạt động giống như một bộ điều chỉnh do dữ liệu tạo ra.

Mặc dù DNN có thể được huấn luyện tốt hơn với quy trình trước đó, nhưng DNN không thể tự xử lý tốt dữ liệu tạm thời. Khi xử lý dữ liệu âm thanh, tôi kết hợp DNN và HMM để tạo ra bộ giải mã tốt hơn. Các đặc điểm thô từ một khung và các

khung liên tiếp của nó được gửi tới DNN cùng nhau, theo đó DNN có thể học được sự phụ thuộc các khung liên tiếp. Mỗi đơn vị đầu ra đại diện cho một trạng thái HMM, nghĩa là DNN dự đoán xác suất quan sát mà khung hiện tại thuộc về mỗi trạng thái. Ba HMM ho và một HMM không ho được huấn luyện. Sau khi các xác suất quan sát được tính toán, một thuật toán giải mã Viterbi sẽ được áp dụng với tất cả các HMM. Đối với mỗi mẫu, một phiên mã, chứa trình tự giải mã HMM có khả năng xảy ra nhất, có thể được tạo ra từ một quá trình giải mã. Một mẫu được dán nhãn là ho nếu HMM ho được tìm thấy trong trình tự mã hóa.

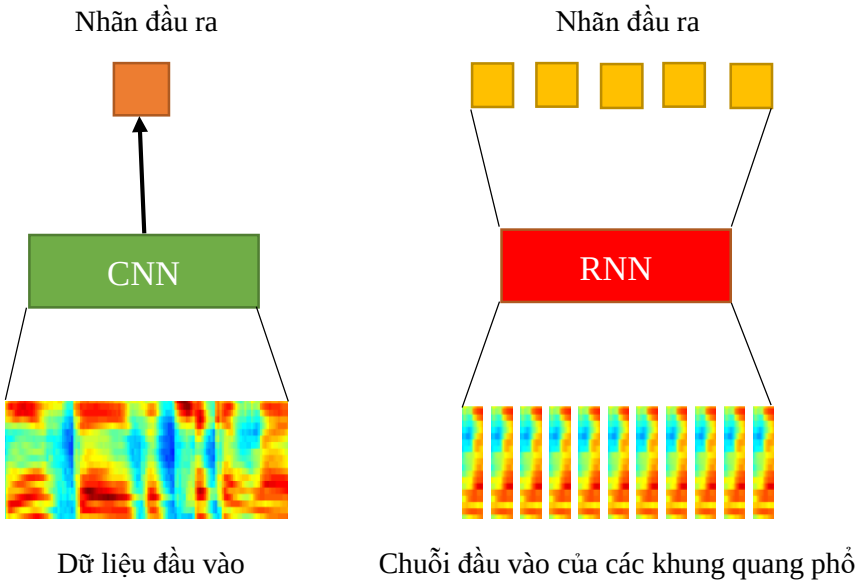
"Ground truth" cho các trạng thái được tạo ra bằng cách sử dụng hệ thống cơ sở GMM-HMM để ghi trước các dữ liệu huấn luyện. Trong nguyễn mẫu kết hợp DNN và GMM-HMM, DNN được sử dụng để tính toán khả năng tham dò qua sát, và GMM-HMM được sử dụng để giải mã cấu trúc thời gian. Quy trình huấn luyện cho hệ thông DNN-GMM-HMM được thể hiện trong hình 2.4.




Hình 2.4 Quá trình đào tạo kết hợp giữa DNN và GMM-HMM

## 2.3 Mô hình máy học CNN-LSTM sử dụng cho việc phát hiện và phân loại ho

Phát hiện âm thanh triệu chứng ho có thể thực hiện dưới dạng trực quan bằng cách chuyển đổi âm thanh 1 chiều thành dạng "hình ảnh" thời gian quang phổ 2 chiều, bằng máy tính chẳng hạn Biến đổi Fourier thời gian ngắn (STFT). Sau đó, mạng lưới nơ-ron thần kinh nhân tạo có thể hoạt động tốt với việc nhận dạng hình ảnh để so sánh các âm thanh của triệu chứng ho các âm thanh không phải triệu chứng ho trên các mẫu hình ảnh trong nội dung quang phổ 2 chiều. Ưu điểm của phương pháp này là biến thể mạng thân kinh nhân tạo được sử dụng cho việc nghiên cứu kỹ lưỡng hình ảnh, Mạng học sâu tích chập (CNN) [14], có thể sử dụng cho việc phát hiện các triệu chứng ho (Hình 2.5)




Hình 2.5 Một minh họa của mạng nơ-ron tích chập và quy hồi cho hai công thức phát hiện ho.

Trong CNN, đầu vào là một hình ảnh mang kích thước cố định, một đoạn của quang phổ và đầu ra là một nhãn duy nhất. RNN nhận vào một chuỗi các khung phổ và xuất ra một chuỗi các nhãn. Với việc gần đây được sử dụng vô cùng phổ biến, CNN rất dễ đào tạo và có rất nhiều tài nguyên phần mềm và phần cứng sẵn có cho việc này. Tuy nhiên, có một nhược điểm tương đối lớn với các công thức hình ảnh. Nó yêu cầu các kích thuốc dữ liệu đầu vào để huấn luyện là phải được chỉnh sửa hoàn

thành và được định nghĩa trước. Mặc dù điều này là khá đơn giản và phù hợp với các hình ảnh truyền thống vì ban đầu chúng đã định dạng ở kích thước 2D cố định, nhưng đó cũng chính là 1 điều khá đáng lưu ý dối với các dữ liệu hiển thị theo thời gian thực như âm thanh. Đối với 1 một cái máy ảnh, việc ghi lại hình ảnh sẽ luyên có cùng kích thước và độ phân giải, tuy nhiên việc này ngược lại đối với các dữ liệu âm thanh, các âm thanh ho có những bản ghi khác nhau về mặt thời lượng thu được. Do đó, một công việc luôn phải thực hiện vô cùng cần thiết đó là đàm bảo dữ liệu thời gian từ các tin hiệu âm thanh tín hiệu phải luôn được cố định trước khi đưa vào huấn luyện cho mạng nơ-ron nhân tạo. Khi chúng ra chỉnh sửa, chắc chắn sẽ có những đoạn âm thanh dài hơn hoặc ngắn hơn so với tiêu chuẩn đã đề ra ban đầu, chúng ra sẽ phải cắt ngắn bớt các phần âm thanh dài hơn và ghép nối những phần này vào các đoạn âm thanh bị thiếu thời lượng so với tiêu chuẩn. Đối với việc loại bỏ các phần đoạn thừa nhiều, sẽ làm giảm dữ liệu có sẵn để đào tạo và điều này là không được khuyến khích vì mạng học sâu nhân tạo sẽ được huấn luyện tốt hơn khi có càng nhiều dữ liệu mẫu được đưa vào. Mặt khác đối với các phần dữ liệu không được chỉnh sửa tốt có quá nhiều tạp âm hay là nhiễu so với các nội dung nguyên bản ban đầu cũng làm giảm độ chính xác của việc phát hiện và phân loại triệu chứng ho. Bên cạnh các yêu cầu về chia cắt các đoạn âm thanh, hình ảnh được đưa vào cũng cần một giai đoạn xử lý hậu kỳ khi chuyển đổi từ âm thanh sang dạng trực quan để căn chỉnh các dán nhãn đầu ra được chuẩn với các tín hiệu âm thanh được đưa vào và chuyển đổi.

### *2.3.1 Mạng học sâu tích chập cho phát hiện và phân loại ho (CNN)*

#### 2.3.1.1 Giới thiệu về mạng nơ-ron tích chập (CNN)

Một trong các ứng dụng quan trọng của mạng nơ-ron tích chập đó là cho phép các máy tính có khả năng “nhìn” và “phân tích”. Nó được lấy cảm hứng từ vỏ não thị giác. Nghĩa là Convnets (Convolutional Neural Networks) được sử dụng để nhận dạng hình ảnh bằng cách đưa nó qua mạng nơ-ron với nhiều layer, mỗi layer là các bộ lọc tích chập. Sau khi đi qua các layer này chúng ta có được đặc trưng và dùng nó nhận dạng ra đối tượng.

Mỗi khi chúng ta nhìn thấy một cái gì đó, một loạt các lớp tế bào thần kinh được kích hoạt, và mỗi lớp thần kinh sẽ phát hiện một tập hợp các đặc trưng như đường thẳng, cạnh, màu sắc,… của đối tượng. Lớp nơ-ron càng cao sẽ phát hiện các đặc trưng phức tạp hơn để nhận ra những gì chúng ta đã thấy.

ConvNet có 02 phần chính: Lớp trích lọc đặc trưng của ảnh (Conv, Relu và Pool) và Lớp phân loại (FC và softmax).

### a. Feature (Đặc trưng)

Features có thể hiểu là các đặc trưng, Ở đây, ta thấy CNN so sánh hình ảnh theo từng mảnh, mỗi mảnh đó được gọi là Feature. So với việc khớp các bức ảnh lại với nhau thì CNN làm việc nhìn ra sự tương đồng trong việc tìm kiếm tho các Feature khớp với nhau trong hai hình ảnh tốt hơn.

Mỗi Feature được coi như một hình ảnh nhỏ, tức là chúng cũng là những mảng hai chiều nhỏ. Các Feature sẽ được khớp với các khính cạnh chung của bức ảnh đó nghĩa là feature này sẽ tương ứng với khía cạnh nào đó của hình ảnh và chúng sẽ được khớp lại với nhau.

### b. Tích chập (Convolutional)

Nói theo một cách đơn giản thì khi xem một hình ảnh mới, CNN sẽ không biết nó ở vị trí nào, và các feature sẽ khớp với nhau ở đâu, vì vậy nó thử sắp xếp các feature tại các vị trí khác nhau. Trong quá trình đó, chúng ta sẽ tạo thành được một bộ lọc được gọi là Filter. Và để thực hiện được điều này, chúng ta đã sử dụng phần toán gọi là nơ-ron tích chập.

Để hoàn tất được quá trình tích chập, chúng ta phải lặp lại hành động trên. Đó chính là việc ta sẽ xếp tất cả các Feature vào tất cả mọi mảnh hình ảnh có thể thực hiện được. Kết quả của quá trình này chính là chúng ta sẽ có được bộ lọc, và với mỗi bài toàn khác nhau chúng ta sẽ có một filter tương ứng. Có thể nói, quá trình tích chệp diễn ra theo từng lớp và nó được gọi là layer.

### c. Các lớp cơ bản trong CNN

- *Lớp tích chập:*

Có thể nói đây là một lớp cực kỳ quan trọng trong CNN, bởi ở lớp này sẽ thực hiện mọi phép tính toán. Một số khái niệm cần nhắc tới ở Lớp tích chập là filter map, stride, padding, feature map.

Nếu như ANN kết nói với từng pixel của hình ảnh đầu vào thì CNN sử dụng những filter để áp vào những vùng của hình ảnh. Các filter map này chính là một ma trận ba chiều, trong đó bao gồm các parameter.

Stride ở đây có thể hiểu là khi bạn dịch chuyển filter map theo pixel dựa vào một giá trị từ trái qua phải. Stride chính là sự dịch chuyển này.

Padding là những giá trị 0 được thêm vào lớp input

Feature map thể hiện kết quả qua mỗi lần filter map quét qua input. Mỗi lần quét như thế sẽ xảy ra quá trình tính toán.

- *Lớp tổng hợp (pooling):*

Một công cụ mạnh mẽ khác mà CNNs sử dụng được gọi là pooling. Pooling là một cách lấy những hình ảnh lớn và làm co chúng lại trong khi vẫn giữ các thông tin quan trọng nhất trong đó. Pooling chỉ dùng kiến thức toán của lớp hai. Nó bao gồm việc duyệt bước một ô vuông cửa sổ nhỏ dọc trên một hình ảnh và lấy giá trị lớn nhất từ cửa sổ ở mỗi bước. Trong thực tế, một cửa sổ có cạnh 2 hoặc 3 điểm ảnh và duyệt bước mỗi 2 điểm ảnh là được.

Sau khi pooling, một hình ảnh sẽ có khoảng một phần tư số điểm ảnh so với lúc bắt đầu. Vì nó giữ các giá trị lớn nhất từ mỗi cửa sổ, nó sẽ bảo toàn tính khớp của mỗi feature bên trong cửa sổ. Nghĩa là nó không quan tâm quá nhiều về vị trí chính xác nơi feature khớp, miễn là nó khớp ở chỗ nào đó trong cửa sổ. Kết quả là CNNs có thể tìm xem liệu một feature có nằm trong hình ảnh mà không cần lo nó nằm ở đâu. Điều này giúp giải quyết vấn đề của máy tính là quá trực nghĩa.

Một layer pooling là hoạt động thực hiện pooling trên một hình ảnh hoặc một tập các hình ảnh. Đầu ra sẽ có cùng số lượng hình ảnh, nhưng mỗi cái sẽ có điểm ảnh ít hơn. Điều này cũng rất hữu ích trong việc quản lý tải trọng tính toán. Hạ một tấm ảnh 8 megapixel xuống còn 2 megapixel sẽ giúp mọi xử lý tải về trở nên dễ dàng.

- *Lớp Relu:*

Relu layer chính là một hàm kích hoạt trong mạng neural network. Hàm kích hoạt còn được gọi là activation function. Tác dụng chính của hàm kích hoạt này chính là việc mở kích hoạt mô phỏng các neuron có tỷ lệ truyền xung qua axon. Trong activation function có các hàm cơ bản như: sigmoid, Tanh, Relu, Leaky relu, Maxout.

Hiện nay, hàm relu đang được sử dụng khá phổ biến và thông dụng. Đặc biệt là trong việc huấn luyện các neuron thì relu có những ưu điểm nổi bật. Có thể kể đến như việc tính toán nhanh hơn,…

### d. Cấu trúc của mạng CNN

Mạng CNN gồm nhiều lớp Convolution chồng lên nhau, sử dụng các hàm và tanh để kích hoạt các trọng số. Mỗi một lớp sau khi được kích hoạt sẽ cho ra kết quả trừu tượng cho các lớp tiếp theo. Mỗi layer kế tiếp chính là thể hiện kết quả của layer trước đó.

Thông qua quá trình huấn luyện, các lớp CNN tự động học các giá trị được thể hiện qua các lớp filter.

Có hai điều cần quan tâm ở mô hình CNN là tính bất biến và tính kết hợp. Trong trường hợp, cùng một đối tượng mà chiếu theo những góc khác nhau thì sẽ cho độ chính xác bị ảnh hưởng.

Đối với phép dịch chuyển, quay và co dãn sẽ sử dụng lớp tổng hợp để sử dụng làm bất biến các tính chất kia. Vì vậy mà CNN đưa ra kết quả có độ chính xác cao ở các mô hình.

Cấu trúc của CNN gồm 3 phần chính: Local receptive field, shared weight and bias, pooling.

- Local receptive field: hay còn gọi là các trường cục bộ. Tác dụng của lớp này chính là nó giúp chúng ta tách lọc các dữ liệu, thông tin của ảnh và chọn được những vùng có giá trị sử dụng nhất.

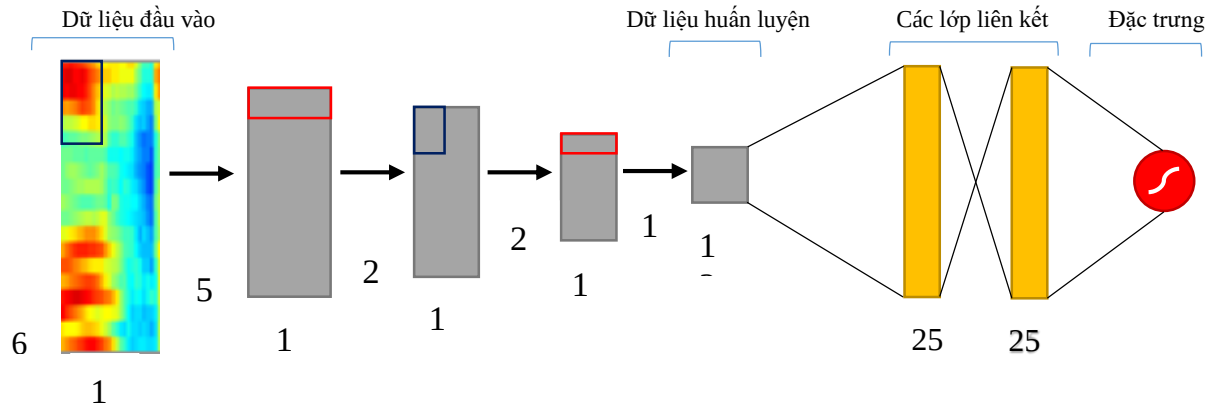
- Shared weight and bias: Trọng số chia sẻ làm giảm tối đa số lượng các tham số là tác dụng chính của yếu tố này trọng mạng CNN hiện nay. Bởi trong mỗi

convolution có những feature map khác nhau, mỗi feature map lại giúp phát hiện một vài đặc trưng trong ảnh.

- Lớp tổng hợp: Gần đây như là lớp cuối cùng trước khi cho ra kết quả. Vì vậy, để có được kết quả dễ hiểu và dễ dùng nhất thì lớp tổng hợp sẽ có tác dụng làm đơn giản hóa thông tin đầu ra. Tức là sau khi hoàn tất các quá trình tính toán và quét các lớp thì sẽ đi đến lớp tổng hợp để giản lược bớt những thông tin không cần thiết, sau đó sẽ cho ra kết quả mà chúng ta mong muốn.

#### 2.3.1.2 Kiến trúc phát triển cho bài toán phát hiện và phân loại ho

Trong quá trình tiền xử lý, luồng dữ liệu âm thanh sẽ được phân tích thành các khung hình mỗi khung hình có độ dài mỗi khung là 4ms. Để loại bỏ các dữ liệu không liên quan như các dải âm thanh ồn ào và những phần không hiển thị âm thanh, ta sử dụng bộ tiền xử lý thực hiện qua trình chuyển đổi khung hình bởi Lu et al.[26]. Với mỗi 16 khung hình (64 ms), năng lượng RMS được tính toán và so sánh với ngưỡng xác định trước. Các khung có năng lượng thấp được coi như sự chuyển đổi từ các phần không có âm thanh và sẽ bị loại bỏ khi các phần năng lượng cao được "công nhận". Vì một số âm thanh như lời nói có thể gián đoán hoặc không liên tục, Có khả năng các khung hình với năng lượng thấp có thể lẫn vào các sự kiện âm thanh. Để tránh việc loại bỏ các khung như vậy, một khi các khung được chấp nhận, nó được coi là mô tả sự bắt đầu của một sự kiện âm thanh và 4 khung tiếp theo sẽ được chấp nhận bất kể năng lượng của chúng có thấp hay cao. Do đó, sự kiện tối thiểu 320 ms cũng sẽ được chấp nhận, đó cũng là độ dài trung bình của âm thanh ho. Các sự kiện âm thanh đã được thừa nhận cũng sẽ được chuẩn hóa bằng giá trị RMS đang chạy, và sau đó trải qua quá trình chuyển đổi quang phổ - thời gian (spectro – temporal).




Hình 2.6: Mô tả kiến trúc CNN

Đầu vào cho mạng là một biểu đồ phổ STFT 64 ms. Mạng bao gồm có hai lớp chập, hai lớp dày đặc và một lớp phân loại softmax. Mỗi lượt chuyển đổi có 16 bộ lọc. Đối với các sự kiện được chấp nhận, mỗi 128 bin biến đổi Fourier thời gian ngắn. được thực hiện để tạo ra một biểu đồ quang phổ với 64 điểm tần số và khung thời gian khác nhau tùy thuộc vào độ dài của sự kiện. Các bảng quang phổ được phân thành các đoạn 16 khung hình và 4 khung hình này được chồng lên nhau đáp ứng với 64 ms dữ liệu âm thanh gốc. Đói với các trường hợp mà có ít hơi 16 khung nội dung, các phần dư thiếu sẽ được đặt là 0. Các phân đoạn phổ 64x16 được gán một nhãn duy nhất liên quan đến lớp (ho | không ho). Từ các đoạn quang phổ này cùng với cá nhãn của ta sẽ là dữ liệu đầu vào để CNN phân loại.

Kiến trúc mạng nơ-ron của tôi được lấy cảm hứng từ kiến trúc khá phổ biến đó là LeNet-5 [27] mang lại hiệu suất nghệ thuật cao đối với tập dữ liệu chữ viết tay MNIST. So với các kiến trúc nổi tiếng khác như AlexNet [28]. LeNet-5 là một mạng nhỏ hơn nhiều và phù hợp hơn với các tập dữ liệu nhỏ hơn, Tuy nhiên, vì tập dữ liệu của tôi thậm chí còn nhỏ hơn tập dữ liệu MNIST, nên tôi đã giảm số lượng tế bào thần kinh trong mỗi lớp theo phương pháp heuritics thông thường, chẳng hạn như đảm bảo số lượng đơn vị ẩn chỉ là một phần nhỏ của đầu vào.

Giống như LeNet-5, mạng của tôi bao gồm năm lớp: 2 lớp chập, 2 lớp được kết nối đầy đủ và một lớp phân loại softmax. Mỗi lớp chập có 16 đơn vị tuyến tính được chỉnh lưu (ReLU). Lớp chập đầu tiên lấy các phân đoạn quang phổ 64x16 làm

đầu vào và có các bộ lọc có kích thước 9x3. Tiếp theo là lớp tổng hợp tối đa 2x1. Lớp tích chập thứ hai có các bộ lọc có kích thước 5x3 và cũng được theo sau bởi một lớp tổng hợp tối đa 2x1. Các phép biến đổi được thực hiện với bước sóng là 1. Các lớp chập được theo sau bởi 2 lớp được kết nối đầy đủ với 256 đơn vị tuyến tính được chỉnh lưu cho mỗi lớp. Các lớp được kết nối đầy đủ cũng sử dụng quy định bỏ lớp (p = 0,5) để giảm việc trang bị quá mức. Cuối cùng, lớp cuối cùng lấy các đầu ra của lớp thứ hai được kết nối đầy đủ và phân loại đầu vào là một sự kiện ho hoặc lời nói bằng cách sử dụng hàm softmax. Kiến trúc mạng được minh họa trong Hình 2.6.

Tôi đã chọn cách thức ReLU thay vì các hàm tanh hoặc sigmoid truyền thống vì ReLU không có vấn đề về độ dốc biến mất và thường dẫn đến hội tụ nhanh hơn [28]. Các kích thước bộ lọc tích hợp được chọn để kích hoạt tính năng tích hợp 2D: trên cả miền tần số và miền thời gian. Các ứng dụng trước đây của mạng phức hợp trong âm thanh đôi khi biến đổi dọc theo trục thời gian hoặc tần số. Tuy nhiên, đối với ứng dụng của tôi, vì tôi biết cả mô hình thời gian và quang phổ ngắn hạn có thể phân biệt đối với các sự kiện ho và lời nói, tôi xoay quanh cả hai chiều. Ngoài ra, vì các phân đoạn đầu vào của tôi bao phủ một khoảng thời gian tương đối ngắn (16 khung hình, 64 mili giây), tôi cố định kích thước của các bộ lọc dọc theo trục thời gian (ở 3 khung hình). Tổng hợp các lớp xuống các kết quả đầu ra của mẫu phức hợp để làm cho các phép tính có thể quản lý được trong các lớp tiếp theo. Tương tự như định cỡ bộ lọc của tôi, tôi không thực hiện gộp theo trục thời gian để tránh làm giảm thêm độ phân giải tạm thời hạn chế của các phân đoạn.

### *2.3.2 Áp dụng mô hình Sequence-to-Sequence cho việc phân loại và phát hiện ho*

#### 2.3.2.1 Giới thiệu về mạng nơ-ron quy hồi (RNN)

Con người không bắt đầu suy nghĩ của họ từ đầu tại tất cả các thời điểm. Cũng như bạn đang đọc bài viết này, bạn hiểu mỗi chữ ở đây dựa vào từ bạn đã hiểu các chữ trước đó chứ không phải là đọc tới đâu ném hết đi tới đó, rồi lại bắt đầu suy nghĩ

lại từ đầu tới chữ bạn đang đọc. Tức là tư duy đã có một bộ nhớ để lưu lại những gì diễn ra trước đó.

Tuy nhiên các mô hình mạng nơ-ron truyền thống thì không thể làm được việc đó, đó có thể coi là một khuyết điểm chính của mạng nơ-ron truyền thống. Ví dụ, bạn muốn phân loại các bối cảnh xảy ra ở tất cả các thời điểm trong một bộ phim, thì đúng là không rõ làm thế nào để có thể hiểu được một tình huống trong phim mà lại phụ thuộc vào các tình huống trước đó nếusử dụng các mạng nơ-ron truyền thống.

Mạng nơ-ron hồi quy (Recurrent Neural Network) sinh ra để giải quyết vấn đề đó. Mạng này chứa các vòng lặp bên trong cho phép thông tin có thể lưu lại được. Các vòng lặp này khiến cho mạng nơ-ron hồi quy trông có vẻ khó hiểu. Tuy nhiên, nếu bạn để ý một chút thì nó không khác mấy so với các mạng nơ-ron thuần. Một mạng nơ-ron hồi quy có thể được coi là nhiều bản sao chép của cùng một mạng, trong đó mỗi đầu ra của mạng này là đầu vào của một mạng sao chép khác. Chuỗi lặp lại các mạng này chính là phân giải của mạng nơ-ron hồi quy, các vòng lặp khiến chúng tạo thành một chuỗi danh sách các mạng sao chép nhau. Trong vài năm gần đây, việc ứng dụng RNN đã đưa ra được nhiều kết quả không thể tin nổi trong nhiều lĩnh vực: nhận dạng giọng nói, mô hình hóa ngôn ngữ, dịch máy, mô tả ảnh,… Danh sách vẫn còn đang được mở rộng tiếp. Đằng sau sự thành công này chính là sự đóng góp của LSTM. LSTM là một dạng đặc biệt của mạng nơ-ron hồi quy, với nhiều bài toán thì nó tốt hơn mạng hồi quy thuần. Hầu hết các kết quả thú vị thu được từ mạng RNN là được sử dụng với LSTM.

### a. Vấn đề phụ thuộc xa

Một điểm nổi bật của RNN chính là ý tưởng kết nối các thông tin phía trước để dự đoán cho hiện tại. Việc này tương tự như ta sử dụng các cảnh trước của bộ phim để hiểu được cảnh hiện thời. Đôi lúc ta chỉ cần xem lại thông tin vừa có thôi là đủ để biết được tình huống hiện tại. Trong tình huống này, khoảng cách tới thông tin có được cần để dự đoán là nhỏ, nên RNN hoàn toàn có thể học được.

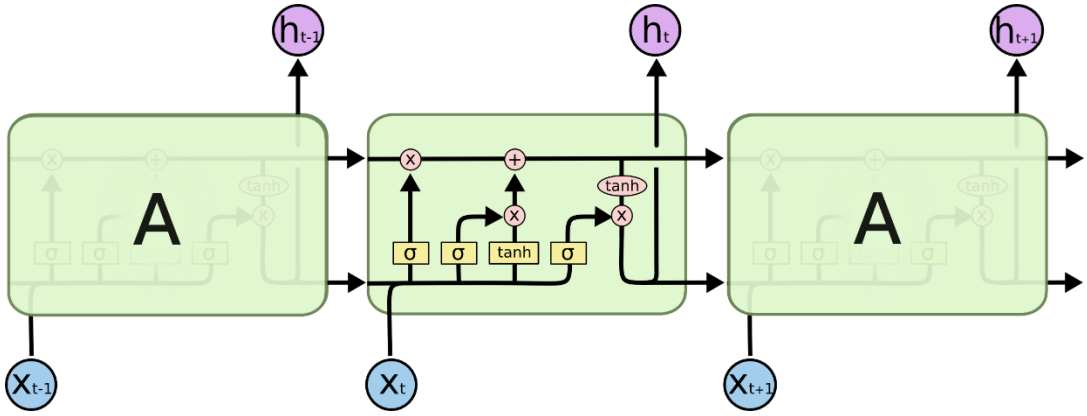
Nhưng trong nhiều tình huống ta buộc phải sử dụng nhiều ngữ cảnh hơn để suy luận. Ví dụ, dự đoán chữ cuối cùng trong đoạn: “I grew up in France… I speak fluent French.”. Rõ ràng là các thông tin gần (”I speak fluent”) chỉ có phép ta biết được đằng sau nó sẽ là tên của một ngôn ngữ nào đó, còn không thể nào biết được đó là tiếng gì. Muốn biết là tiếng gì, thì ta cần phải có thêm ngữ cảnh “I grew up in France” nữa mới có thể suy luận được. Rõ ràng là khoảng cách thông tin lúc này có thể đã khá xa rồi. Thật không may là với khoảng cách càng lớn dần thì RNN bắt đầu không thể nhớ và học được nữa. Về mặt lý thuyết, rõ ràng là RNN có khả năng xử lý các phụ thuộc xa (long-term dependencies). Chúng ta có thể xem xét và cài đặt các tham số sao cho khéo là có thể giải quyết được vấn đề này. Tuy nhiên, đáng tiếc trong thực tế RNN có vẻ không thể học được các tham số đó.

#### 2.3.2.2 Mạng LSTM

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & Schmidhuber (1997), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng tanh. LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.




Hình 2.7: Mô hình LSTM

b. Ý tưởng cốt lõi của LSTM

Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình 7. Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi. LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate). Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân. Tầng sigmoid sẽ cho đầu ra là một số trong khoản [0, ], mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 0 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 1 thì có nghĩa là cho tất cả các thông tin đi qua nó. Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

c. Bên trong LSTM

Bước đầu tiên của LsTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid – gọi là “Tầng cổng quên” (forget gate layer). Nó lấy đầu vào là $h_{t-1}$ và $x_t$ rồi đưa ra kết quả là một số trong khoảng [0, 1] cho mỗi số trạng thái tế bào . Đầu ra là 1 thể hiện rằng nó giữ toàn bộ thông tin lại, còn 0 chỉ rằng toàn bộ thông tin sẽ bị bỏ đi. Quay trở lại với ví dụ mô hình ngôn ngữ dự đoán từ tiếp theo dựa trên tất cả các từ trước đó, với những bài toán

như vậy, thì trạng thái tế bào có thể sẽ mang thông tin về giới tính của một nhân vật nào đó giúp ta sử dụng được đại từ nhân xưng chuẩn xác. Tuy nhiên, khi đề cập tới một người khác thì ta sẽ không muốn nhớ tới giới tính của nhân vật nữa, vì nó không còn tác dụng gì với chủ thế mới này.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{12}$$

Bước tiếp theo là quyết định xem thông tin nào ta sẽ lưu vào trạng thái tế bào. Việc này bao gồm 2 phần. Đầu tiên là sử dụng một tầng sigmoid được gọi là "Tầng cổng vào" (input gate layer) để quyết định giá trị nào ta sẽ cập nhật. Tiếp theo là một tầng tanh tạp ra một vector cho giá trị mới $\tilde{C}_t$ nhằm thêm vào cho trạng thái. Trong bước tiếp theo, ta sẽ kết hợp hai giá trị đó để tạo ra một cập nhật cho trạng thái. Chẳng hạn với ví dụ mô hình ngôn ngữ của ta, ta sẽ muốn thêm giới tính của nhân vật mới này vào trạng thái tế bào và thay thế giới tính của nhân vật trước đó.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{13}$$

$$\tilde{C}_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{14}$$

Giờ là lúc cập nhật trạng thái tế bào cũ $C_{t-1}$ thánh trạng thái mới $C_t$. Ở các bước trước đó đã quyết định những việc cần làm nên ta chỉ cần thực hiện là xong. Ta sẽ nhận trạng thái cũ với $f_t$ để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm $i_t * \tilde{C}_t$. Trạng thái mới thu được này sẽ phụ thuộc vào việc ta quyết định cập nhật mỗi giá trị trạng thái ra sao. Với bài toàn mô hình ngôn ngữ, chính là việc ta bỏ đi thông tin về giới tính của nhân vật cũ, và thêm thông tin về giới tính của nhân vật mới như ta đã quyết định ở các bước trước đó.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{15}$$

Cuối cùng, ta cần quyết định xem ta muốn đầu ra là gì. Giá trị đầu ra sẽ dựa vào trạng thái tế bào, nhưng sẽ được tiếp tục sàng lọc. Đầu tiên, ta chạy một tầng sigmoid để quyết định phần nào của trạng thái tế bào ta muốn xuất ra. Sau đó, ta đưa nó trạng thái tế bảo qua một hàm tanh để co giá trị nó về khoảng [-1,1], và nhân nó

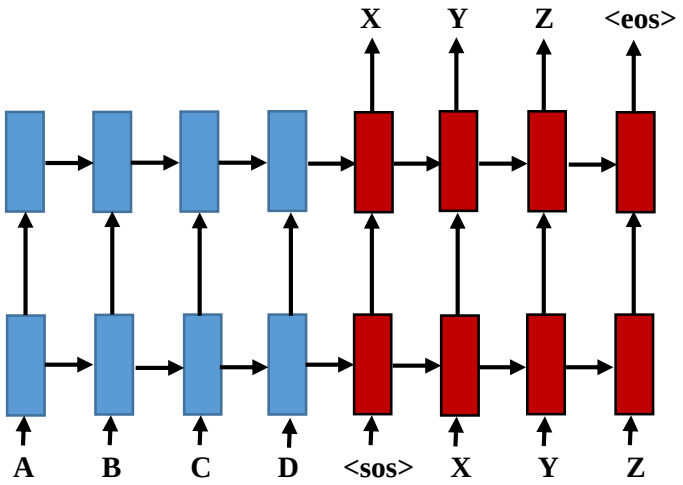
với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn. Với ví dụ về mô hình ngôn ngữ, chỉ cần xem chủ thể mà ta có thể đưa ra thông tin về một trạng từ đi sau đó. Ví dụ, nếu đầu ra của chủ thể là số ít hoặc số nhiều thì ta có thể biết được dạng của trạng từ đi theo sau nó phải như thế nào.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{16}$$

$$h_t = o_t * \tanh(C_t) \tag{17}$$

### 2.3.2.3 Mô hình Sequence-to-Sequence

Mô hình Sequence-to-Sequence được đề xuất bởi Sutskever et al. vào năm 2014 và được sử dụng tạo ra một chuỗi các token của câu trong ngôn ngữ đích $y = \{y_1, ..., y_m\}$ làm câu bản dịch tương ứng cho một chuỗi các token của câu ngôn ngữ nguồn $x = \{x_1, ..., x_n\}$ được cung cấp trước. Mục tiêu của quá trình huấn luyện là tối ưu hóa xác suất có điều kiện $p(y_1, ..., y_m \mid x_1, ..., x_n)$ với giá trị của $m$ là độ dài của chuỗi đầu ra có thể khác với $n$ là độ dài của chuỗi đầu vào. Mô hình này sử dụng kiến trúc Encoder-Decoder và thông thường thì mạng RNN hoặc những mạng như mạng LSTM và GRU sẽ được dử dụng cho cả bộ Encoder và bộ Decoder. Đặc biệt, mạng LSTM được sử dụng để giải quyết các vấn đề phụ thuộc dài, ghi nhớ và biểu diễn mối quan hệ của các thông tin phụ thuộc vào ngữ cảnh trong câu văn bản.




Hình 2.8: Kiến trúc của mô hình Sequence-to-Sequence với câu đầu vào là chuỗi “A B C D” và câu đầu ra là chuỗi “X Y Z”

Các thành phần chính của mô hình Sequence-to-Sequence bao gồm:

- Bộ Encoder được sử dụng để ánh xa chuỗi token trong ngôn ngữ nguồn đầu vào thành một vector có kích thước cố định. Tại mỗi bước mã hóa, Encoder sẽ nhận vector tương ứng với mỗi token trong chuỗi đầu vào để tạo ra vector trạng thái ẩn $s$ đại diện cho chuỗi đầu vào tại bước mã hóa cuối cùng.

- Bộ Decoder sử dụng vector $s$ như khởi tạo cho trạng thái ẩn đầu tiên và tạo ra chuỗi các token ở ngôn ngữ đích tại mỗi bước giải mã. Do đó, hàm xác suất có điều kiện có thể được phân tích như sau:

$$p(y_1,...,y_m \mid x_1,...,x_n) = \prod_{j=1}^{m} p(y_j \mid s, y_1,...,y_{j-1}) \tag{18}$$

Trong vế phải của công thức trên, mỗi phân bố $p(y_j \mid s, y_1,...,y_{j-1})$ mô tả xác suất xuất hiện của token $y_j$ với vector đại diện cho câu đầu vào $s$ và các token trong chuỗi đầu ra đứng trước nó. Phân bố này được biểu diễn bằng một hàm softmax trên tất cả các token trong tập từ vựng ở ngôn ngữ đích.

Công thức trên có thể viết thành dạng như sau:

$$\log p(y \mid x) = \sum_{j=1}^{m} \log p(y_j \mid y_{j<s}, s) \tag{19}$$

Mỗi token $y_j$ có xác suất xuất hiện được tính như sau:

$$p(y_j \mid y_{j<s}, s) = soft\max(g(h_j)) \tag{20}$$

Trong đó $g$ là hàm dùng để biến đổi trạng thái ẩn $h_j$ của decoder tại vước giải mã tương ứng thành vector có kích thước bằng kích thước của tập từ vựng trong ngôn ngữ đích. Trạng thái ẩn $h_j$ được tính như sau:

$$h_j = f(h_{j-1}, s) \tag{21}$$

Trong đó $f$ là hàm biểu diễn chung cho quá trình tính trạng thái ẩn tại bước hiện tại của trạng thái ẩn đầu ra của bước trước bằng mạng RNN hoặc bằng những

cải tiến khác như LSTM và GRU. Trong mô hình của Sutskever et al., vector $s$ đại diện cho câu nguồn chỉ được sử dụng một lần để làm trạng thái ẩn đầu tiên cho bộ Decoder. Trong mô hình của tác giả Bahdanau et al. Và tác giả Luong et al., $s$ là một vector đặc biệt được sử dụng xuyên suốt tại mỗi bước trong quá trình giải mã.

Hàm mất mát cần tối ưu hóa trong quá trình huấn luyện là một hàm có dạng tích của các hàm crossEntropy:

$$L = -\sum_{j=1}^{m}\sum_{i=1}^{V} q_{j,i} \log(p_{j,i}) \tag{22}$$

Trong đó, $q_{j,i}$ là phân tử thứ $i$ của vector one-hot $q_j$ có kích thước $V$ tại bước giải mã thứ $j$. Vector $q_j$ biểu diễn cho token thứ $j$ trong chuỗi đầu ra từ tập huấn luyện. $p_{j,i}$ là phần tử thứ $i$ của vector $p_{j,i}$ cũng có kích thước $V$ với $p_j = soft\max(g(h_j))$

Về cơ bản sau khi huấn luyện hoàn tất, chúng ta sẽ tạo ra bản dịch từ một chuỗi đầu vào chưa biết trước bằng cách tính toán sinh ra bản dịch có khả năng xuất hiện cao nhất dựa vào mô hình thu được sau huấn luyên:

$$\hat{y} = \arg\max_{y}(p(y \mid \hat{x})) \tag{23}$$

a. Cơ chế giải mã với thuật toán Greedy Search

Trong quá trình giải mã của mô hình Sequence-to-Sequence, thuật toán Greedy Search là một giải pháp đơn giản để mô hình dự đoán phân tử của chuỗi đầu ra tại mỗi bước của quá trình giải mã. Ở mỗi bước thời gian, trạng thái ẩn ở mạng RNN của bộ Decoder sẽ được ánh xạ thành một vector có kích thước $V$ bằng với kích thước $V$ của tập từ vựng ở ngôn ngữ đích. Hàm softmax sẽ chuẩn hóa vector này thành vector $p$ với mỗi phần tử là giá trình xác suất xuất hiện của mỗi token tương ứng trong tập từ vựng ở ngôn ngữ đích với chuỗi đầu vào trước và chuỗi các phần tử đã được giải mã tại các bước trước. Hàm argmax sẽ tính ra vị trí của phần tử trong vector $p$ có xác suất cao nhất và chúng ta sẽ chọn ra được token có vị trí tương đương với giá trị này

trong tập từ vựng ở ngôn ngữ đích. Quá trình giải mã dừng lại khi trong chuỗi đầu ra xuất hiện token đặc biệt "< EOS >".

b. Cơ chế giải mã với thuật toán Beam Search

Một vấn đề khi mô hình Sequence-to-Sequence áp dụng thuật toán Greedy Search trong quá trình giải mã nếu những phần tử đầu tiên trong chuỗi đầu ra được dự đoán thiếu chính xác, chất lượng toàn chuỗi đầu ra sẽ bị ảnh hưởng nghiêm trọng do quá trình giải mã tạo ra các phần tử tiếp theo trong chuỗi đầu ra đều được tính toán dựa trên các phần tử đầu tiên. Với thuật toán Beam Search, thay vì chỉ chọn ra một phần tử duy nhất có xác suất cao nhất tại mỗi bước giải mã, chúng ta dữ lại $k$ giả thuyết có xác suất cao nhất cho các bước giải mã tiếp theo với $k$ là tham số chiều rộng (beam width). Khi token đặc biệt "< EOS >" xuất hiện trong mọi giả thuyết, chúng ta kết thúc quá trình giải mã và chọn ra giả thuyết có giá trị xác suất $p(y_1, y_2, ..., y_{<EOS>} \mid x_1, x_2, ..., x_n)$ cao nhất làm kết quả cuối cùng cho chuỗi đầu ra. Ý tưởng này khắc phục được vấn đề khi mô hình Sequence-to-Sequence áp dụng thuật toán Greedy Search cho quá trình giải mã, cho phép quá trình giải mã có thể tạo được chuỗi đầu ra có chất lượng tốt hơn nếu như những phần tử đầu tiên của chuỗi đầu ra thiếu chính xác.

2.3.2.4 Áp dụng mô hình cho việc phát hiện và phân loại ho

Việc gán nhãn là một bước cần thiết trước khi phân đoạn và hỗ trợ cho việc nhận diện hình ảnh. Mục tiêu của việc gán nhãn cho mô hình này đó là ánh xạ một chuỗi chưa được phân đoạn (dữ liệu đầu vào) với một chuỗi khác (nhãn đầu ra). Nhận dạng giọng nói, nhận dạng chữ viết tay và dịch máy là những ví dụ điển hình của các bài toán áp dụng mô hình này. Cụ thể với những ứng dụng, việc ghi nhãn có khả năng mô hình hóa các đặc trưng dài hạn của âm thanh ho. Đặc biệt, nó có thể ghi lại sự phụ thuộc về thời gian và phổ giữa ba pha đặc trưng (pha ban đầu, pha giữa và pha cuối cùng) của một cơn ho. Để thực hiện nhiệm vụ nhận dạng và phân loại âm thanh ho, dữ liệu thời gian quang phổ từ tín hiệu âm thanh vẫn có thể sử dụng bất chấp những đặc tính về thời gian của dữ liệu. Không giống như việc nhận dạng trực quan, chúng

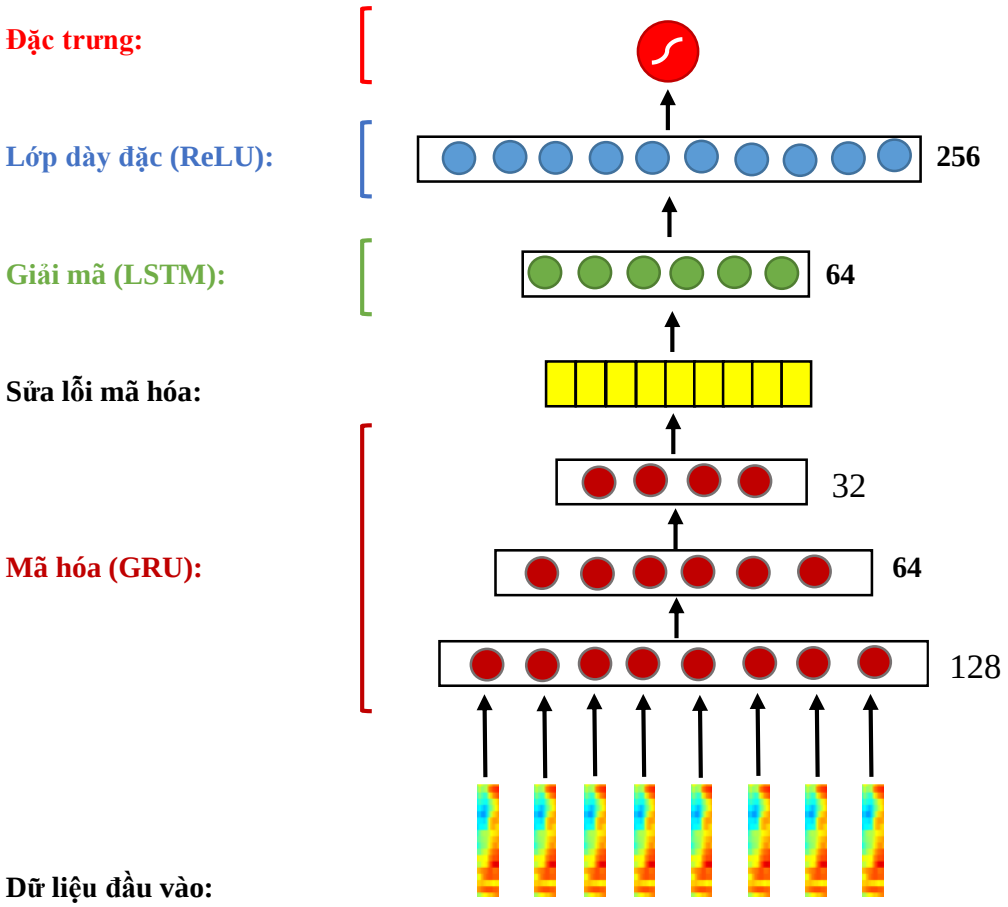
ta có thể có các dữ liệu đầu vào đa dạng về thời gian thu được; điều này giúp chúng ta phải thực hiện việc loại bỏ hoặc thay thế các dữ liệu sử dụng cho việc huấn luyện. Hơn nữa, với việc dữ liệu đầu ra là dữ liệu tuần tự, nên chúng ta không cần xử lý hậu kỳ hay căn chỉnh các dán nhãn dự đoán. Có một mô hình cổ điển, giúp chúng ta giải quyết các vấn đến cho việc dán nhãn đó là mô hình Markov ẩn (HMM). Tuy nhiên, các HMM bị hạn chế bởi việc thu nhận các đặc tính dài hạn. Trong quá trình huấn luyện các HMM, các sự kiện trong quá khứ sẽ có ít nhiều sự ảnh hưởng đối với các biến dữ liệu hơn các sự kiện gần hơn. Do đó, mô hình kết quả khá độc lập với các dữ liệu đầu vào và đầu ra.

Tại hình 2.5, Mạng học sâu quy hồi (RNN) là biến thể của mạng nơ – ron nhân tạo đã xử lý các vấn đề gán nhãn cho mô hình Sequence-to-Sequence bằng cách sử dụng các phương pháp học sâu. Mặc dù, mô hình mạng học sâu quy hồi (RNN) cũng bị hạn chế đối với các đặc trưng dài hạn, tuy nhiên đã có 2 nghiên cứu gần đây đã giúp chúng ta giải quyết vấn đề đó. Đầu tiên, họ đã chỉ ra rằng sự phân cấp của các RNN (mạng sâu hơn) có thể mô hình hóa các đặc trưng dài hạn tốt hơn vì chúng có khả năng phân giải tốt các dữ liệu theo có đặc tính thời gian [24]. Thứ hai, hiện nay có nhiều loại nơ – ron mới đặc biệt cho phép RNN kiểm soát nhiều hơn bộ nhớ trong của chúng [25]. Kết quả là, không giống như HMM, các RNN hiện tại với các tế bào thần kinh chuyên việt có thể nắm bắt và mô hình hóa các bối cảnh dài hạn theo trình tự. Bên cạnh đó, RNN cũng không bị ảnh hưởng quá nhiều bởi nhiễu trong các dữ liệu tuần tự.

Nhược điểm của việc học dán nhãn trình tự bằng mạng học sâu quy hồi (RNN) là quá trình đào tạo có thể khó khăn và cần nhiều mẫu. Do đó, các ứng dụng của RNN trước đây hoạt động đối với các tác vụ tuần tự không mong đợi như nhận dạng giọng nói. Tuy nhiên, các nghiên cứu gần đây về mạng nơ-ron nhân tạo đã giới thiệu các kỹ thuật chuyển đổi, bổ sung như cắt độ dốc (TensorFlow), các dạng của tế bào thần kinh khác đã làm cho việc huấn luyện RNN trở nên dễ dàng và hiệu quả hơn. Cùng với đó, sự tăng trưởng theo cấp số nhân về tài nguyên của máy tính trong những năm cũng đã góp phần đáng kể trong việc huấn luyện RNN cũng như tất cả các phương

pháp học sâu khác. Hiện tại, các RNN đã đặt được những hiệu quả cao trong hầu hết các bài toán như nhận dạng giọng nói và dịch máy.

Tín hiệu được veotor hóa thành các khung hình 4ms và có một giao thức được sử dụng cho việc tiếp nhận các khung hình. Giao thức này cũng đóng vai trò cho việc phân loại như loại bỏ các phần âm thanh tĩnh lặng hay các khung có năng lượng thấp. Tuy nhiên trong trường hợp này, các khung hình được chấp nhận có thể có các độ dài khác nhau (trái ngược với việc toàn bộ các khung phải là 64ms cố định). Các khung được xác nhận và chuyển đổi thành chuỗi 64 phần tần số trên mỗi khung hình, là phối cảnh tuần tự của các phân đoạn phổ STFT từ khi các thành phần cho việc nhận dạng được thiết lập. Các chuỗi này và các nhãn tương ứng của chúng được sử dụng để huấn luyện mạng nơ-ron quy hồi.




Hình 2.9: Tổng quan về kiến trúc RNN bộ mã hóa – giải mã để phát hiện ho

Bộ mã hóa bao gồm ba lớp; hai lớp đầu tiên có đơn vị ha chiều và thứ ba là đơn hướng. Tất cả các tế bào thần kinh trong bộ mã hóa đều là Gated Recurrent Unit (GRU). Bộ giải mã là một lớp duy nhất LSTM có tích hợp sẵn cơ chế Attention. Tiếp theo là một lớp dày đặt các đơn vị ReLU, và cuối cùng là một lớp phân loại softmax.

Tôi đã triển khai RNN của mình với kiến trúc bộ mã hóa – giải mã 6 lớp, cho phép mạng xử lý và phân loại các chuỗi đầu vào và chuỗi đầu ra có độ dài tùy ý. Bộ mã hóa được tạo ra từ ba lớp: 2 lớp quy hồi hai chiều với 128 và 64 đơn vị tương ứng và một lớp quy hồi đơn chiều với 32 đơn vị. Bộ mã hóa của tôi được thiết lập để xử lý các chuỗi có độ dài tối đa được cố định và tôi đặt tùy thuộc vào thử nghiệm (xem phần thử nghiệm phía bên dưới). Tất cả các nơ-ron quy hồi trong bộ mã hóa là Gated Recurrent Unit (GRU), có thể xác định các phần mang tính chất dài hạn tron một chuỗi dữ liệu đầu vào. Lớp cuối cùng của bộ mã hóa xuất ra một phần thông tin cố định, sau đó sẽ được sử dụng để tạo bộ giải mã. Bộ giải mã là một lớp quy hồi duy nhất gồm 64 đơn vị bộ biến đổi dài ngắn (LSTM), kết hợp với cơ chế Attention. Cơ chế Attention cho phép mạng tập trung vào các phần nổi bật của tính năng đầu vào và cuối cùng dẫn đến cải thiện hiệu suất phân loại. Hiện tại, bộ giải mã của tôi được thiết lập để xuất một nhẫn duy nhất cho một chuỗi đầu vào. Sau bộ giải mã, chúng ta có một lớp được kết nối đầy đủ với 256 tế bào thần kinh ReLU. Cuối cùng lớp phân loại xuất ra một nhãn lớp bằng cách sử dụng hàm softmax. Mô hình bộ mã hóa – giải mã cũng được minh họa trong hình 2.8.

# CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ

## 3.1 Thu thập dữ liệu

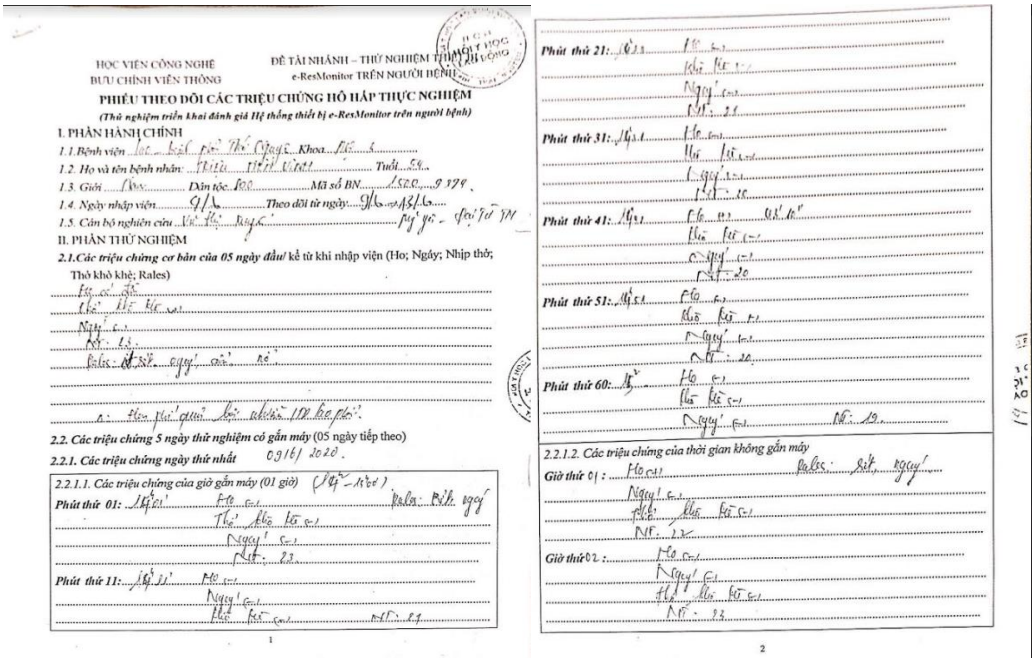
### *3.1.1 Thu âm và gán nhãn âm thanh*

Để đo âm thanh ho, chúng tôi đã sử dụng một thiết bị thu âm thanh thu âm liên tục quá trình của các bệnh nhân nhiễm bệnh bao gồm cả âm thanh ho và các âm thanh ngoại cảnh. Chiến micro thu âm này được gắn trực tiếp trên ngực áo của bệnh nhân và kết nối với phần mềm thu âm trên điện thoại. Sau khi đeo các thiết trên người, đối với mỗi bênh nhân chúng tôi sẽ yêu cầu việc thực hiện ghi âm quá trình tại giường bệnh trong vòng 30 phút tới một tiếng. Toàn bộ các âm thanh trên khoảng thời gian bệnh nhân được yêu cầu đeo sẽ được ghi âm lại bao gồm cả âm thanh ho và các âm thanh ngoại cảnh.

Thiết bị thu âm sẽ bao gồm hệ thống micro thu âm đẳng hướng liên kết trực tiếp với thiết bị di động của bệnh nhân và gắn tại cổ áo của bệnh nhân như hình 3.1. Phần lớn thiết bị này sẽ tương thích với cổng âm thanh 3.5 trên thiết bị di động của bệnh nhân. Đối với các thiết bị đặc biệt mới như các thiết bị Android được sản xuất sau năm 2018 hay một số các thiết bị sử dụng hệ điều hành IOS từ Iphone 7 trở lên, chúng tôi sẽ cung cấp cho các bệnh nhân các jack cắm chuyển đổi để có thể kết nối thiết bị thu âm với thiết bị di động cá nhân của họ (Hình 3.2). Tất cả các âm thanh được thu lại sẽ được chuyển lại với định dạng WAV. Hệ thống âm thanh được thu lại này sẽ được chuyển tới các bác sỹ chuyên môn nghe lại và đưa ra các đánh giá sơ bộ về quá trình thu âm cũng như dạng ho của người bệnh trong một mẫu đánh giá được cung cấp sẵn như hình 3.3. Sau khi các dữ liệu được các bác sĩ đã chuẩn đoán chính xác, tôi sẽ sử dụng dữ liệu này cũng với các file âm thanh đã thu được và tiến hành gán nhãn âm thanh trên phần mềm Audacity (Tại hình 3.4 là quá trình gán nhãn một tệp âm thanh đã được chỉnh sửa và truy xuất sau khi lấy ra từ hệ thống thu âm). Việc gán nhãn âm thanh của tôi dữ trên sự khác biệt của âm trường trong file gán nhãn kết hợp với việc nghe liên tục file ghi âm.


Hình 3.1: Thiết bị thu âm được cung cấp tới bệnh nhận

Hình 3.2: Một số các cổng chuyển đổi được sử dụng cho việc kết nới mic với các thiết bị không hỗ trợ cổng cắm 3.5



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐỀ TÀI NHÁNH – THỬ NGHIỆM TRÊN NGƯỜI BỆNH e-ResMonitor

**PHIẾU THEO DÕI CÁC TRIỆU CHỨNG HÔ HẤP THỰC NGHIỆM**

*(Thử nghiệm triển khai đánh giá Hệ thống thiết bị e-ResMonitor trên người bệnh)*

I. PHẦN HÀNH CHÍNH

1.1. Bệnh viện ... Khoa ...

1.2. Họ và tên bệnh nhân: ... Tuổi ...

1.3. Giới ... Dân tộc ... Mã số BN ...

1.4. Ngày nhập viện ... Theo dõi từ ngày ...

1.5. Cán bộ nghiên cứu ...

II. PHẦN THỬ NGHIỆM

**2.1.Các triệu chứng cơ bản của 05 ngày đầu** kể từ khi nhập viện (Ho; Ngáy; Nhịp thở; Thở khò khè; Rales)

**2.2. Các triệu chứng 5 ngày thử nghiệm có gắn máy** (05 ngày tiếp theo)

**2.2.1. Các triệu chứng ngày thứ nhất** 03/6/2020.

2.2.1.1. Các triệu chứng của giờ gắn máy (01 giờ)

Phút thứ 01: ...

Phút thứ 11: ...

Phút thứ 21: ...

Phút thứ 31: ...

Phút thứ 41: ...

Phút thứ 51: ...

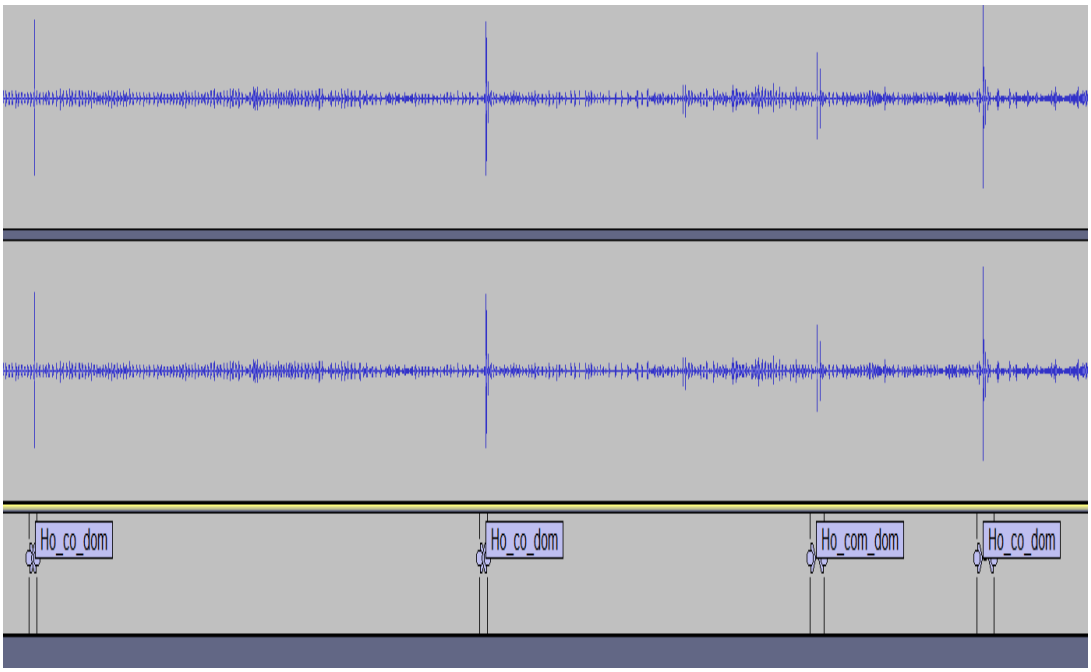
Phút thứ 60: ...

2.2.1.2. Các triệu chứng của thời gian không gắn máy

Giờ thứ 01: ...

Giờ thứ 02: ...

Hình 3.3: Một số phần đánh giá của các bác sỹ chuyên môn




Hình 3.4: sử dụng phần mềm Audacity thực hiện gán nhãn âm thanh

### *3.1.2 Xây dựng và đánh giá âm thanh*

Để xây dựng và đánh giá dữ liệu âm thanh, tôi đã tạo ra một cơ sở dữ liệu ghi âm các bệnh nhân từ các bệnh viên lao phổi tại Hà Nội và tại Thái Nguyên. Các đối tượng có độ tuổi từ 18 – 65 tuổi, bao gồm tất cả các dân tộc của Việt Nam. Tất cả các dữ liệu cũng như quá trình thu thập dữ liệu đều được Thầy giáo hướng dẫn và cơ quan nơi sở tại phê duyệt cho phép thực hiện. Các thiết bị thu âm được sử dụng để thu thập dữ liệu âm thanh khi các bệnh nhân có triệu chứng ho dài hạn và đang điều trị tại bệnh viện sở tại. Môi đối tượng được gắn và sử dụng các thiết bị tối đa trong 5 ngày, mỗi ngày đeo trong vòng 30 phút tới một tiếng. Các thiết bị không hề gây ra sự khó chịu và các bệnh nhân đều sử dụng một cách chính xác tự nhiên không có sự cưỡng bức. Hệ thống thu âm sẽ thu lại toàn bộ âm thanh bao gồm cả âm thanh ho và các âm thanh ngoại cảnh bao gồm tất cả các âm thanh không phải tiếng ho và không phải lời nói mà cảm biến thu được như tiếng thở, nhịp tim, tiếng nổ lách tách và thậm chí cả tiếng bíp được sử dụng để báo hiệu đối tượng trong quá trình thu âm. Nguồn âm thanh này sau khi được thu lại, sẽ được các bác sỹ chuyên môn nghe và đánh giá các âm thanh liên quan tới triệu chứng ho kèm với thời gian xảy ra triệu chứng này. Các nhãn được sử dụng sẽ là: Kho_khe, Ho_co_dom, Ho_khan, Ngay.

Đối với tất cả các bản ghi, thiết bị thu âm được lấy mẫu ở băng tần 44,1 kHz và sau đó được lấy mẫu xuống 16 kHz. Tất cả các sự kiện trong bản ghi (ho, lời nói hoặc các sự kiện khác) được gắn nhãn thủ công trên PC bằng trình chỉnh sửa âm thanh Audacity. Thời gian ho trong tất cả các dữ liệu thu thập được dao động từ 250 ms đến tối đa khoảng 800 ms. Mặt khác, lời nói và các âm thanh khác có xu hướng dài hơn nhiều trong khoảng thời gian. Để đảm bảo các ví dụ đào tạo của chúng tôi có cùng độ dài, chúng tôi chia bài phát biểu và các bản ghi âm khác thành các đoạn nhỏ hơn với thời lượng ngẫu nhiên được tạo ra từ phân phối Gaussian về thời lượng của các ví dụ ho. Điều này dẫn đến thời lượng trung bình khoảng 320 ms cho tất cả các ví dụ trong cơ sở dữ liệu của tôi, không phân biệt lớp của chúng.

## 3.2 Huấn luyện dữ liệu

Trước khi huấn luyện, tôi đã tối ưu hoa bằng cách sử dụng một tập hợp nhỏ của cơ sở dữ liệu được chạy nhiền lần trong các cấu hình khác nhau để tìm rat ham số huấn luyện tối ưu (ví dụ như: tốc độ học,…). Khi các siêu tham số này được thiết lập, chúng tôi tiến hành xây dựng các mạng thực tế với dữ liệu đầy đủ. Để huấn luyện cả hai mạng nơ-ron này, một lược đồ xác nhận chéo được sử dụng 10 lần. Cơ sở dữ liệu được chia thành 10 phần: 9 phần để xây dựng mô hình, phần còn lại để thử nghiệm mô hình. Điều này được thực hiện lặp đi lặp lại trên tất cả các phần trong 10 lần chạy khác nhau và sau đí chúng ta sẽ tính toán các chỉ số trung bình từ các chỉ số đánh giá mỗi lần thu được. Tất cả các dữ liệu được xáo trộn trước khi phân vùng và do đó có khả năng trùng lặp giữa các dữ liệu huấn luyện và các dữ liệu kiểm tra. Nói cách khác, có thể các mẫu âm thanh từ cùng một nguồn nằm trong cả dữ liệu huấn luyện và dữ liệu kiểm tra. Tuy nhiên, trong phần thử nghiệm, chúng tôi đã giám sát xem mạng của chúng tôi tổng quát đạt được hiệu quả như thế nào đối với các âm thanh, một dạng dữ liệu mà chúng ta không thể nhìn thấy.

Mạng học sâu đã có các thuận lợi nhất định khi có được bộ dữ liệu được sử dụng cho việc huấn luyện lớn tới vậy. Vì vậy, các ý tưởng về việc gia tăng số lượng các mẫu, các ví dụ huấn luyện có thể hữu ít khi chúng ta huấn luyện cho hệ thống. Trong ứng dụng của chúng tôi, dữ liệu đầu vào được tăng cường để góp phần cho việc bổ xung cho quá trình chuyển đổi thông tin khi huấn luyện. Điều này được thực hiện bằng cách đảo ngược các đoạn quang phổ từ các sự kiện giống nhau có sự trung lặp tối đa là 25%. Cơ sở dữ liệu của chúng tôi phân dải thành 11,125 phân đoạn mà chúng tôi huấn luyện cho mạng của mình. Chúng tôi cũng chuẩn hóa toàn bộ dữ liệu đào tạo trên tất cả các thành phần như thường được thực hiện trong việc đào tạo các mạng nơ-ron nhân tạo. Dưới đây, chúng tôi nêu các chi tiết bổ sung các phần dữ liệu cho việc huấn luyện của hai mạng khác nhau.

1) Mạng học sâu tích chập (CNN): Mạng học sâu tích chập được huấn luyện bằng cách sử dụng các gốc ngẫu nhiên (SGD), với tốc độ là 0,001, batch size là 20 và gia tốc Nesterov là 0,9. Mạng có 660,690 thông số có thể học được và quá trình huấn

luyện sau khoảng 50 chu kỳ, với thời gian chạy trung bình khoảng 5 tiếng cho tát cả 10 lần chạy.

2) Huấn luyện Mạng học sâu quy hồi (RNN): Mạng quy hồi với 323,983 tham số và được huấn luyện bằng cách sử dụng quy trình tối ưu hóa 'adadelta'.: một phương pháp để giảm dần độ dốc với tốc độ học thích ứng, ít nhạy hơn các quy trình tối ưu hóa khác khi đối với tham số tốc độ học ban đầu [29]. Mặc dù adadelta không phải là trình tối ưu hóa nhanh nhất cho ứng dụng, nhưng nó được phát hiện là có thể đưa quá trình huấn luyện mượt mà hơn, và mạng lại độ chính xác khi thử nghiệm tốt hơn so với các trình tối ưu khác đã thử như vani SGD, rmsprop [30] và adagrad [31]. Tốc độ học ban đầu là 0.005 và batch size là 40 được sử dụng, Mạng được đào tạo trong 35 chu kỳ, mất khoảng 7 tiếng cho 10 lần huấn luyện, Các kỹ thuật khác được sử dụng để làm cho quá trình huấn luyện định kỳ có hiệu quả là cắt giảm độ dốc và chuẩn hóa hàng loạt. Gradient clipping áp đặt một ngưỡng đối với gradient như một phương tiện để hạn chế sự ảnh hưởng khi mà bất kỳ phần tử thần kinh được kích hoạt có thể tạo ra hiện tượng mất thông tin trong tổng thể quá trình lan truyền ngược [32]. Việc cắt bớt độ đốc được phát hiện để làm cho việc huấn luyện các lớp quy hồi có thể truy xuất được như các báo cáo trong nghiên cứu khác [33]. Chuẩn hóa hàng loạt là một kỹ thuật được giới thiệu gần đây để giải quyết vấn đề sự thay đổi trong phân phối quá trình kích hoạt mạng trong quá trình huấn luyện [34]. Nó liên quan đến việc chuẩn hóa các đầu vào cho mỗi lớp bằng cách thống kê các mini-batch. Chuẩn hóa hàng loạt cải thiện tốc độ huấn luyện. hoạt động như một bộ điều chỉnh để giảm việc lắp quá mức, và thường dẫn đến độ chính xác xác nhận cao hơn ngay cả trong mạng quy hồi [35].

Cả hai mạng đều được triển khai bằng Lasange [36], một thư viện python dựa trên Theano để đào tạo mạng nơ-ron. Huấn luyện được thực hiện trên một PC duy nhất với CPU R7 2700x 3.7GHz và bộ nhớ ram là 32 GB.

## 3.3 Thử nghiệm phát hiện và phân loại ho

Để so sánh và đánh giá hiệu suất của hai công thức phát hiện ho áp dụng phương pháp học sâu, chúng tôi đã thực hiện 5 lần thực nghiệm. Đầu tiên, chúng tôi nghiên cứu và kết luận rằng cả hai mạng lưới thần kinh đều trích xuất các đặc trưng hiệu quả để xác định và phân loại ho. Tiếp theo, chúng tôi so sánh CNN và RNN với nhau trong một nhiệm vụ phân loại nghiêm ngặt hơn để khám phá thêm khả năng phân biệt ho của chúng. Trong thử nghiệm thứ ba, chúng tôi điều tra xem cả CNN và RNN nắm bắt sự phụ thuộc dài hạn tốt như thế nào bằng cách thử nghiệm cả hai mô hình trên các chuỗi dài hơn. Trong thử nghiệm thứ tư, chúng tôi xác minh xem mô hình của chúng tôi hoạt động tốt thế nào trên dữ liệu từ các đối tượng nằm ngoài cơ sở dữ liệu của chúng tôi. Cuối cùng, trong thử nghiệm thứ năm, chúng tôi kiểm tra xem hiệu suật của cả hai mạng bị ảnh hưởng như thế nào bởi kích thước của chúng. Tất cả các thử nghiệm trên được thực hiện theo sơ đồ xác thực chéo 10 lần và các chỉ số hiệu suất được tính trung bình trên tất cả các lần.

Các chỉ số mà chúng tôi sử dụng để đánh giá mô hình đó là độ nhạy, độ đặc hiệu và độ chính xác. Độ nhạy được tính bằng tỉ số các cơn ho được xác định chính xác trên tổng số các cơn ho trong một tập hợp thử nghiệm. Độ đặc hiệu, là tỷ lệ các trường hợp xác định chính xác các trường hợp không ho trên tổng số các trường hợp không ho. Độ chính xác là tổng hợp của cả độ nhạy và độ đặc hiệu, là tỉ lệ các mẫu được xác định chính xác (dù cho ho, không ho, các dạng ho) trên tổng số mẫu trên bộ thử nghiệm.

### *3.3.1 Thử nghiệm 1*

Để xác minh mức độ hiệu quả các tính năng của RNN và CNN đã học đối với việc phân loại ho, chúng tôi so sánh chúng với các tính năng MFCC thường được sử dụng. Trong thí nghiệm này, chúng tôi chỉ tập trung vào hai lớp: tiếng ho và tiếng nói. Chúng tôi trích xuất 13 hệ số MFCC từ các âm thanh trong cơ sở dữ liệu của chúng tôi bằng cách sử dụng chiều rộng cửa sổ phân tích là 32ms và độ dài các bước nhảy là 16ms (chồng chéo là 50%). Các thông số phân tích này tương tự như các

thống số thường được sử dụng trong các nghiên cứu ho GIAO DỊCH IEEE TRÊN MẠCH VÀ HỆ THỐNG SINH HỌC 6 và nhận dạng giọng nói [37], [21]. Phân tích MFCC tạo ra 3 khung hình đối với mỗi khung hình ứng vơi 64ms âm thanh. Do đó, các đặc trưng của MFCC được chia thành phân đoạn 13x3 để tạo ra một thiết lập có thể so sánh được với các phân đoạn phổ được sử dụng để huấn luyện CNN và RNN. Ngoài ra, mặc dù RNN có thể sxuer lý các chuỗi có độ dài tùy ý, chúng tôi đã đặt độ dài tối đa mỗi chuỗi là 16 khung hình (64ms) để cho phép so sánh trực tiếp với các đặc trung CNN và MFCC với độ phân dải tại thời điểm đó. Với cách tiếp cận này, mỗi phương pháp trích xuất đặc trung (CNN, RNN, MFCC) mang lại các đặc trung có ý nghĩ để phân loaijbatas ký đoạn âm thanh 64 ms nào có phải sự kiện ho hay là không. Vì lớp phân loại của cả RNN và CNN đều là các hàng Softmax, nên một hàm softmax (SM) cũng được huấn luyện sử dụng các đạc trưng của MDCC. Với cùng một loại bộ phận phân loại và số lượng các đặc trung gần giống nhau, việc so sánh trực tiếp độ chính xác phân loại sẽ đưa ra kết luận về khả năng đại diện của MFCC, CNN và RNN đối với nhiệm vụ phát hiện và phân loại ho của chúng tôi. Chúng tôi cũng huấn luyện một hàm cơ sở xuyên tâm với máy vector hỗ trợ (SVM) trên MFCC để quan sát cách so sánh giữa một bộ phân loại phức tạp hơn với các mạng nơ-ron sâu. Ngoại ra, chúng tôi huấn luyện một SVM trên dữ liệu SFFT thô dùng làm thanh tham chiếu để so sánh với RNN và CNN.

### *3.3.2 Thử nghiệm 2*

Trong thử nghiệm thứ hai, chúng tôi sử dụng thiết lập tương tự như thử nghiệm đầu tiên, ngoaoij trừ chúng tôi chỉ tập trung vào CNN và RNN. Nhiệm vụ phân biệt được thực hiện thực tế hơn bằng cách bao gồm các âm thanh khác không phải là tiếng ho và tiếng nói mà cảm biến âm thanh sẽ đo được khi sử dụng để thu âm thực tế. Những âm thanh này bao gồm: Tiếng nhịp tim, hơi thở, tiếng cười, tiếng hắng giọng, tiếng nghiến rang và âm thanh từ điều kiện ngoại cảnh do các tác động vật lý tới thiết bị thu âm (ví dụ: Khi bệnh nhân cạm vào cảm biến). Mục đích là đánh giá hiệu suất của mạng học sâu trong bài toàn phân loại nhiều lớp.

### *3.3.3 Thử nghiệm 3*

Ở đây chúng tôi kiểm nghiệm ở mức độ mà cả hai kiến trúc mạng đề có thể nắm bắt được sự phụ thuộc lâu dài và liệu điều này có thể cải thiện khả năng phát hiện và phân loại ho hay không. Điều này được thự hiện bằng cách chạy cả hai mô hình trên các chuỗi dài hơn. Vì thiết lập thử nghiệm này cho phép cả hai mô hình được chạy trên toàn bộ sự kiện ho, chúng tôi cũng so sánh với phương pháp phát hiện ho Mô hình Markov ẩn thông thường [21]. Như trong thử nghiệm đầu tiên, chúng tôi chỉ tập trung vào hai lớp: tiếng ho và tiếng nói. Trong khi Mô hình Markov hỗn hợp (GMM) và RNN đều có thể xử lý các chuỗi đầu vào có độ dài thay đổi, mô hình CNN cũng yêu cầu một đầu vào cố định. Do đó, chúng tôi đặt độ dài trình tự tối đa làm thời gian trung bình của các lần ho trong cơ sở dữ liệu của chúng tôi: 320 ms. Đây là gấp 5 lần độ dài cửa sổ được sử dụng trước đó (64 ms, 16 khung hình) và mang lại các phân đoạn quang phổ 64x80. Các mục nhập cơ sở dữ liệu có thời lượng dài hơn được chia thành hai với 25% chồng chéo và không được đệm nếu cần. Để điều chỉnh mô hình CNN cho phù hợp, chúng tôi chia tỷ lệ chiều rộng của số chập theo thời gian của nó bằng 5. Độ dài chuỗi đầu vào của RNN cũng được tăng cho tối đa 80. Sau đó, cả hai mạng đều được huấn luyện lại trên dữ liệu đã sửa đổi. Sử dụng cùng một dữ liệu và khung thử nghiệm, một mô hình GMM-HMM được triển khai để so sánh. Một GMM-HMM với 10 trạng thái được đào tạo cho mỗi lớp. Trạng thái đầu tiên và trạng thái cuối cùng là không phát xạ, nhưng tất cả các trạng thái giữa đều có phân bố xác suất phát xạ được mô hình hóa bởi hỗn hợp Gaussian 7 chiều. Đối với mỗi ví dụ huấn luyện, 13 hệ số MFCC được tính theo cách tương tự như trong thí nghiệm 3, ngoại trừ điều này dẫn đến chuỗi khung dài hơn 15 cho các ví dụ huấn luyện kéo dài. Sau đó, các tính năng MFCC 13x15 được sử dụng để đào tạo GMM-HMM. Tại thời điểm thử nghiệm, một chuỗi vector đặc trưng tương tự được trích xuất từ ví dụ thử nghiệm được lắp cho cả hai GMM-HMM. Các giá trị khả năng nhật ký kết quả của cả hai đều phù hợp xác định xem âm thanh có liên quan đến sự kiện ho hoặc lời nói hay không. Cấu hình GMM-HMM này khá phổ biến trong các nghiên cứu về ho và nhận dạng giọng nói [21].

### *3.3.4 Thử nghiệm 4*

Trong thử nghiệm thứ tư, chúng tôi điều tra hiệu suất mạng khi không có sự trùng lặp thông tin giữa dữ liệu huấn luyện và thử nghiệm. Cả hai mô hình đều được thử nghiệm trên các mẫu từ hai đối tượng bên ngoài cơ sở dữ liệu; Một bệnh nhân nam và một bệnh nhân nữ. Dữ liệu thử nghiệm trong thiết lập này bao gồm 128 mẫu mỗi âm thanh tiếng ho và tiếng nói. Âm thanh được thu thập bằng cách sử dụng thiết bị di động theo cách giống như âm thanh cơ sở dữ liệu ban đầu được thu thập. Thử nghiệm này nhằm xác minh rằng các mô hình của chúng tôi phét hiện và phân loại tốt cho tiếng ho.

### *3.3.5 Thử nghiệm 5*

Kích thước mạng nơron được đặc trưng bởi hai tham số: số lượng đơn vị ẩn trong một lớp và tổng số lớp trong mạng. Trong thử nghiệm cuối cùng, chúng tôi kiểm tra mức độ ảnh hưởng của việc sửa đổi một trong hai thông số này đến hiệu suất mô hình. Để khảo sát ảnh hưởng của số lớp trong mạng, chúng tôi huấn luyện mạng có số lớp bằng một nửa số lớp trong mô hình ban đầu. Điều này dẫn đến mạng 3 lớp nhỏ hơn cho cả RNN và CNN, so với CNN 5 lớp ban đầu và RNN 6 lớp. Ba lớp là: lớp chập hoặc lặp lại đầu tiên từ các mô hình ban đầu, lớp kết nối đầy đủ 256 đơn vị và lớp phân loại sigmoid cuối cùng. Chúng tôi cũng huấn luyện mạng nơ-ron dày đặc thường xuyên 3 lớp để so sánh. Đối với số lượng đơn vị, chúng tôi tạo ra nhiều mô hình mạng bằng cách giảm số lượng đơn vị trong mỗi lớp của bản gốc theo hệ số 2, 4 và 8. Ví dụ: mô hình RNN “giảm một nửa số lớp”, tương ứng với giảm 2, có 64, 32, 16, 32, 128, 1 số đơn vị trong 6 lớp tương ứng (từ cấu hình 128, 64, 32, 64, 256, 1 ban đầu).

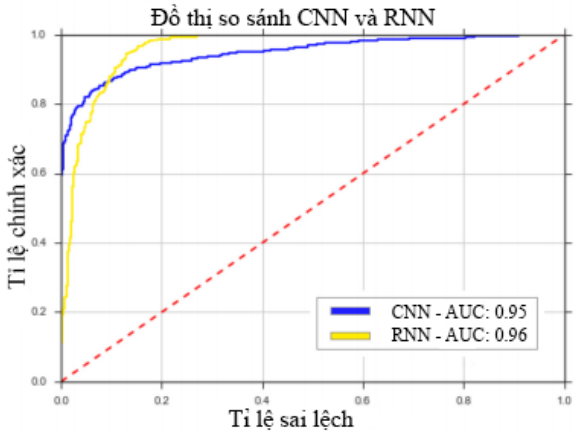
## 3.4 Kết quả thử nghiệm

Kết quả cho Thử nghiệm 1 được báo cáo trong Bảng 3.1. Đầu tiên, chúng tôi nhận thấy rằng cả hai mô hình mạng nơ-ron đều hoạt động tốt hơn so với việc chỉ huấn luyện một SVM trên dữ liệu thô, đây là thử nghiệm cơ bản. Ngoài ra, hai mạng dường như hoạt động tốt hơn cả hai mô hình dựa trên MFCC. Mặc dù MFCC với

softmax (MFCC + SM) dường như có độ nhạy cao, nó thừa nhận rất nhiều kết quả sai và dẫn đến độ chính xác kém. So sánh trực tiếp giữa CNN và RNN, CNN mang lại độ chính xác tổng thể cao hơn 89,7%. Trong khi RNN dường như mang lại độ nhạy trung bình tốt hơn một chút trên 10 lần, nó có phương sai cao hơn nhiều so với CNN. Mặt khác, CNN đạt được độ đặc hiệu lớn hơn đáng kể so với RNN, với độ lệch chuẩn tương đối tối thiểu. Chúng tôi cũng tạo đồ thị đặc tính hoạt động của máy thu (ROC) cho cả hai mạng bằng cách thay đổi ngưỡng trên đầu ra của đơn vị sigmoid cuối cùng (Hình 3.5). Điều này thông báo về khoảng cách giữa các mạng phân tách hai lớp. Từ các đồ thị chúng tôi quan sát thấy cả hai mạng hoạt động khá tốt về mặt này, với giá trị ROC Area Under the Curve (AUC) là 0,96 (RNN) và 0,95 (CNN). Đường cong ROC được tạo bởi ngưỡng thay đổi trên đầu ra của nút cuối cùng trong mạng. RNN dường như có AUC cao hơn một chút là 0,96 so với CNN.

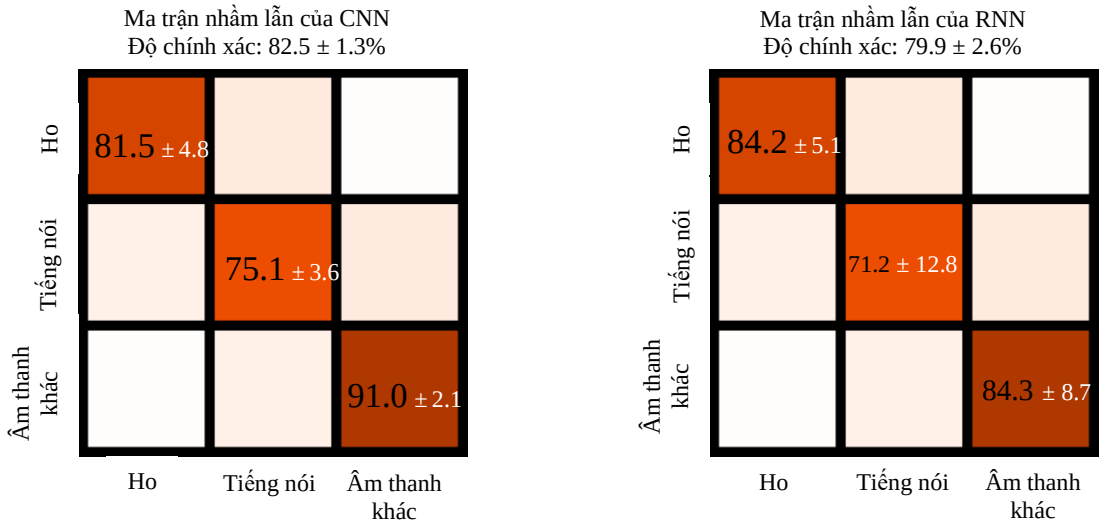
| **Hệ thống** | **Độ nhạy (%)** | **Độ đặc hiệu (%)** | **Độ chính xác (%)** |
|---|---|---|---|
| MFCC+SM | 94.3 ± 3.1 | 68.5 ± 9.4 | 81.4 ± 3.6 |
| MFCC+SVM | 74.9 ± 7.6 | 91.1 ± 1.5 | 87.6 ± 4.8 |
| STFT+SVM | 76.9 ± 3.4 | 74.4 ± 4.8 | 77.2 ± 3.3 |
| STFT+CNN | 86.8 ± 1.5 | 92.7 ± 2.4 | 89.7 ± 1.5 |
| STFT+RNN | 87.7 ± 7.9 | 82.0 ± 11.6 | 84.9 ± 3.6 |

Bảng 3.1: So sánh các kết quả của CNN, RNN và MFCC cho việc phân loại ho tại thử nghiệm 1




Hình 3.5: Đồ thị so sánh AUC của CNN và RNN

Hình 3.6 cho thấy ma trận nhầm lẫn cho cả CNN và RNN trong bài toán phân loại nhiều lớp khó hơn trong Thử nghiệm 2. Nhiệm vụ liên quan đến việc phân biệt ba loại: tiếng ho, tiếng nói và các âm thanh khác. CNN đạt được độ chính xác tổng thể cao hơn 82,5%, mặc dù RNN so sánh tốt trên tất cả các lớp. Như mong đợi, độ chính xác phân loại đã bị giảm xuống đối với cả hai mạng. Tuy nhiên, chúng tôi vẫn quan sát thấy độ chính xác của CNN (82,5%) cao hơn so với RNN (79,9%). Trên cả ba lớp, chúng tôi quan sát xu hướng tương tự như trong thí nghiệm đầu tiên, trong đó độ nhạy cảm của ho cao hơn một chút trong trường hợp RNN trong khi độ chính xác không ho (giọng nói và các hoạt động khác) vẫn cao hơn đáng kể trong CNN.




(a) (b)

Hình 3.6: Ma trận nhầm lẫn cho (a) CNN và (b) RNN trong bài toán phân loại nhiều lớp tại thử nghiệm 2.

| **Hệ thống** | **Độ nhạy (%)** | **Độ đặc hiệu (%)** | **Độ chính xác (%)** |
|---|---|---|---|
| GMM-HMM | 79.1 ± 11.7 | 80.8 ± 5.9 | 79.9 ± 4.0 |
| CNN | 76.2 ± 24.6 | 82.2 ± 6.4 | 79.2 ± 15.0 |
| RNN | 81.7 ± 16.9 | 89.20 ± 18.4 | 85.5 ± 8.6 |

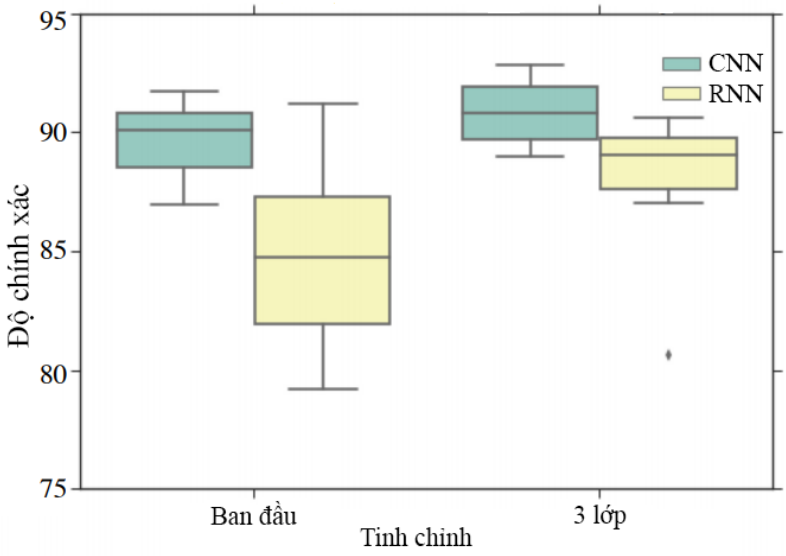
Bảng 3.2: So sánh kết quả giữa các mạng khi sử dụng các chuỗi dài hơn

Trong Bảng 3.2, chúng tôi báo cáo hiệu suất của các mô hình CNN, RNN và GMM-HMM trong thử nghiệm 3. Lưu ý rằng RNN, với độ chính xác phân loại 85,5% dường như hoạt động tốt hơn cả CNN và mô hình GMM-HMM. CNN dường như hoạt động tốt gần như mô hình GMM-HMM, mặc dù với một phương sai lớn hơn nhiều. So với RNN, hiệu suất CNN được điều chỉnh trong dài hạn thấp hơn đáng kể và điều này dường như cho thấy CNN thiếu khả năng nắm bắt sự phụ thuộc lâu dài.

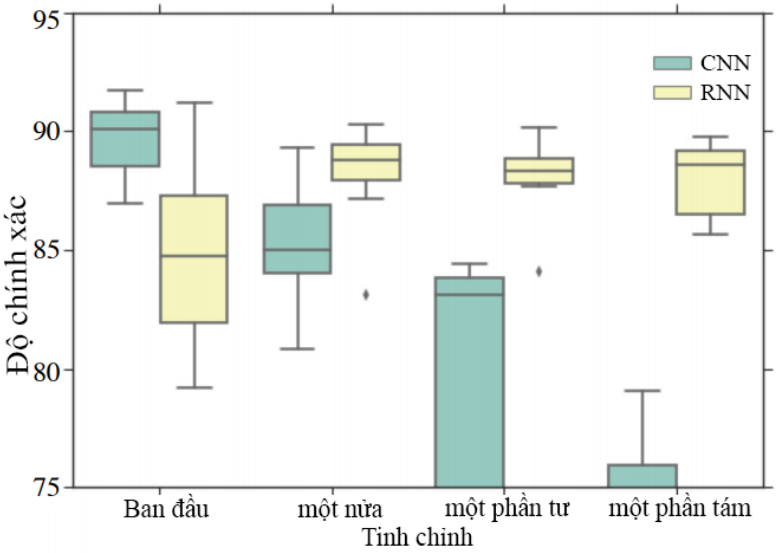
Ngoài ra, Bảng 3.3 cho thấy hiệu suất của các mạng trên các mẫu từ các đối tượng không nhìn thấy. Các giá trị độ nhạy, độ đặc hiệu và độ chính xác cho cả RNN và CNN đều nằm trong phạm vi được quan sát cho các thử nghiệm gấp 10 lần của Thử nghiệm 1. Cuối cùng, các hình phía dưới cho thấy các ô hộp so sánh hiệu suất của cả CNN và RNN với cấu hình kích thước khác nhau hàng tấn trên 10 nếp gấp. Mạng 3 lớp đạt được hiệu suất tốt hơn so với các mô hình ban đầu với độ chính xác là 90,9% đối với CNN và 88,2% đối với RNN (Hình 3.7). Trong khi đó, mạng 3 lớp được kết nối đầy đủ thông thường có độ chính xác là 82,8% ± 2,5. Mặt khác, khi số lượng đơn vị giảm đi một nửa, độ chính xác của CNN giảm xuống trong khi hiệu suất RNN được cải thiện (Hình 3.8). Việc giảm thêm số lượng đơn vị vượt quá một nửa dẫn đến hiệu suất kém hơn trong cả hai mô hình. Xu hướng này rõ ràng hơn đối với CNN so với RNN vì độ chính xác của RNN dường như vẫn cố định. Tuy nhiên, quan sát các giá trị độ nhạy và độ đặc hiệu cho các mô hình RNN cho thấy rằng ngoài việc giảm đi hai lần, độ đặc hiệu chỉ tăng khi độ nhạy phải trả (Hình 3,9). Do đó, số lượng đơn vị tối ưu cho RNN dường như bằng một nửa số đơn vị trong mô hình ban đầu. Lý do điều này là tối ưu là vì độ đặc hiệu được tối đa hóa, với độ chính xác vẫn gần như nhau; chúng tôi muốn độ đặc hiệu rất cao cho một trường hợp hiếm gặp như ho. Các mô hình CNN và RNN nửa đơn vị mang lại độ chính xác lần lượt là 85,3% và 87,6%. Nói chung, người ta lưu ý rằng RNN dường như hoạt động tốt hơn CNN trong các mô hình có ít đơn vị hơn, trong khi ngược lại, đúng với các mô hình có ít lớp hơn.

| Hệ thống | Độ nhạy (%) | Độ đặc hiệu (%) | Độ chính xác (%) |
|---|---|---|---|
| CNN | 82.0 | 93.2 | 87.6 |
| RNN | 84.2 | 75.2 | 79.7 |

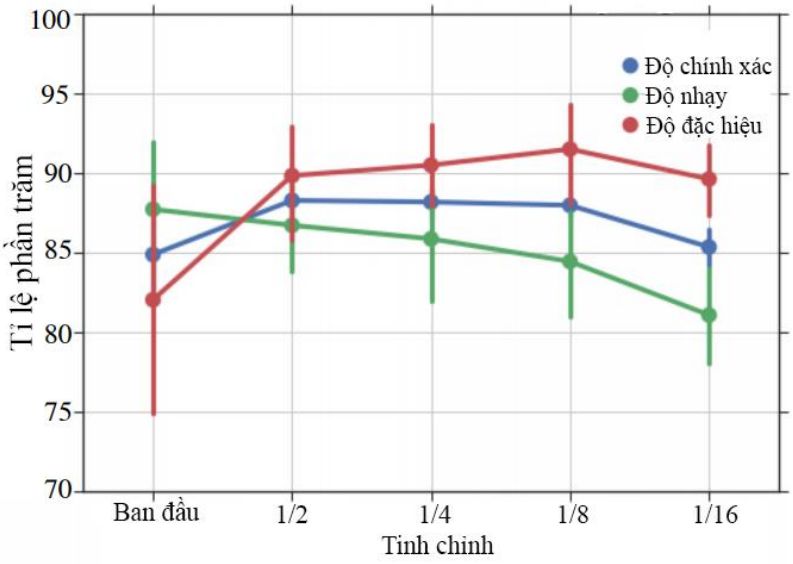
Bảng 3.3: So sánh CNN và RNN khi sử dụng




Hình 3.7: Giảm số lượng lớp của hai mạng




Hình 3.8: Giảm số lượng các đơn vị trong hai mạng




Hình 3.9: Hiệu suất của RNN (LSTM) khi số lượng các đơn vị giảm

Thí nghiệm 5: Đồ thị so sánh độ chính xác của các mạng có cấu hình kích thước khác nhau. Hình 3.7 cho biểu đồ hộp cho độ chính xác của mô hình ban đầu và so sánh với mô hình 3 lớp. Mạng 3 lớp đạt hiệu suất tốt hơn với độ chính xác là 90,9% (CNN) và 88,2% (RNN). Hình 3.8 là ô dạng hộp cho các mô hình có số đơn vị giảm đi 2 (một nửa), 4 (phần tư) và 8 (phần tám) so với mô hình ban đầu. Hình 3.9 so sánh độ chính xác, độ nhạy và độ đặc hiệu của RNN giữa các mô hình với số lượng đơn vị giảm.

## 3.5 Kết luận

Từ thử nghiệm đầu tiên và kết quả trong Bảng 3.1, chúng tôi có thể khẳng định rằng các mô hình mạng nơ-ron của chúng tôi thực sự đang học được các đặc trưng hiệu quả cao. Điều này được thể hiện rõ ràng bằng thực tế là chúng hoạt động tốt hơn bộ phân loại SVM trên STFT thô. Hơn nữa, cả hai mô hình hoạt động tốt hơn so với các mô hình dựa trên MFCC, chứng minh quan điểm rằng các tính năng được học sâu sẽ hiệu quả hơn so với chế tạo thủ công một lần để phát hiện ho. Tuy nhiên, một quan sát thú vị là mô hình MFCC + SVM đạt được độ đặc hiệu rất cao. Một lý do có thể cho điều này có thể là vì các MFCC được thiết kế đặc biệt cho nhận dạng giọng nói, chúng trích xuất các đặc trưng tốt cho nhận dạng giọng nói. Vì tính cụ thể trong thử

nghiệm 1 đề cập đến việc xác định chính xác âm thanh giọng nói, lợi ích bổ sung của các MFCC được quan sát thấy khi sử dụng cùng bộ phân loại SVM là khá mạnh mẽ.

Trong hầu hết các thí nghiệm (1, 2 và 5), chúng tôi đã quan sát thấy mô hình hấp dẫn trong đó CNN cho độ đặc hiệu cao hơn nhiều trong khi RNN (LSTM) cho độ nhạy ho tốt hơn. Một ý kiến có thể giải thích điều này là CNN thực hiện tốt hơn nhiều trong việc phát hiện giọng nói vì phổ giọng nói có các sóng đặc trưng và các mẫu được xác định rõ hơn phổ của âm thanh ho. Mạng CNN, thực sự giỏi trong việc nắm bắt các mẫu hình ảnh, có thể lập mô hình tốt hơn các tín hiệu như vậy trong quang phổ so với RNN (LSTM). Mặt khác, có thể lý giải rằng RNN (LSTM) mang lại độ nhạy tốt hơn vì công thức ghi nhãn trình tự đúng hơn khi thực hiện nhiệm vụ phát hiện ho thực tế.

Một quan sát khác là RNN hoạt động tốt hơn cả CNN và GMM-HMM trên các chuỗi dài hơn. Các đơn vị GRU và LSTM của RNN cho phép nó mô hình hóa tốt hơn sự phụ thuộc dài hạn trong âm thanh ho. CNN cung cấp độ chính xác tương tự như HMM, đặc biệt là khi người ta xem xét phương sai cao của độ chính xác của nó trên 10 lần. Hiệu suất của CNN rất thú vị vì chúng tôi đã chỉ ra rằng CNN có thể mang lại hiệu suất tốt trên các chuỗi dài nếu các nhãn đầu ra của chúng cho các cửa sổ ngắn được tính trung bình trên toàn bộ chuỗi dài hơn. Xem xét cả hai yếu tố, chúng tôi cho rằng sử dụng CNN trên các chuỗi ngắn sẽ tốt hơn so với các chuỗi dài hơn. Nói chung, hiệu suất giảm đối với các chuỗi dài hơn và điều này có ý nghĩa khi số lượng các ví dụ huấn luyện giảm khi các chuỗi được kéo dài.

Liên quan đến các tham số mạng, chúng tôi lưu ý rằng các mô hình 3 lớp hoạt động tốt hơn các mô hình ban đầu của chúng tôi. Điều này có thể có nghĩa là các mô hình ban đầu của chúng tôi đã trang bị quá nhiều dữ liệu của chúng tôi. Nhiều lớp hơn làm cho mạng nơ-ron phi tuyến tính hơn và do đó làm tăng độ phức tạp của mạng. Một mô hình phức tạp hơn có thể dễ dàng trang bị quá mức cho bất kỳ dữ liệu đào tạo nhất định nào nhưng sẽ hoạt động kém trên dữ liệu thử nghiệm. Các mô hình 3 lớp ít phức tạp hơn so với các mô hình ban đầu và do đó, hiệu suất tốt hơn của chúng gợi ý rằng mô hình ban đầu có khả năng trang bị quá nhiều dữ liệu huấn luyện. Điều

đó nói rằng, việc giảm số lượng đơn vị được coi là ảnh hưởng đến hiệu suất nói chung mặc dù số đơn vị tối ưu cho RNN chỉ bằng một nửa số đơn vị trong mô hình ban đầu. Cuối cùng, chúng tôi cho thấy rằng mạng của chúng tôi tổng quát tốt cho các đối tượng ngoài cơ sở dữ liệu, mang lại hiệu suất gần với các kết quả trong tập huấn luyện.

# CHƯƠNG 4: KẾT LUẬN

Chúng ta có thể thấy được nguy hiểm của các triệu chứng ho, sự cấp thiết của việc đề xuất áp dụng trí tuệ nhận tạo trên các thiết bị IoT để phát hiện và phận loại các dạng ho là vô cùng cần thiết. Bằng việc sử dụng các thiết bị IoT, chúng ta có thể tiếp cận đến người mắc các căn bệnh liên quan tới đường hô hấp hay cụ thể là các triệu chứng ho để đưa ra các kết luận ban đầu về dạng ho của người đang mắc bệnh. Đặc biệt đối với thực trạng hiện nay rằng, dịch bệnh COVID-19 vẫn đang không có dấu hiệu dừng lại trên toàn thế giới thì việc, mỗi người tự trang bị có mình các thông tin cũng như hệ thống nhận dạng, phân biệt chủng ho này sẽ góp phần vào quá tải ở các bệnh viên, giảm thiểu số lượng người nhiễm bệnh hay có để đầy lùi được không chỉ dịch COVID-19 mà còn toàn bộ các dịch bệnh nguy hiểm liên quan tới được hô hấp.

Công việc của chúng tôi đầu tiên sử dụng khả năng phát hiện ho dưới dạng quang phổ và các vấn đề ghi nhãn trình tự. Chúng tôi đã triển khai một mạng nơ-ron tích tụ và lặp lại để giải quyết hai công thức tương ứng. Từ đánh giá mô hình của chúng tôi, chúng tôi cho thấy rằng cả hai mạng đều có thể tìm hiểu các tính năng tốt cho nhiệm vụ phát hiện và phân biệt ho. Chúng tôi đã xác định rằng đối với tập dữ liệu của chúng tôi và thiết lập, CNN mang lại độ đặc hiệu tốt hơn trong khi RNN tạo ra độ nhạy tốt hơn. Chúng tôi cũng chỉ ra các yếu tố thay đổi như độ dài chuỗi đầu vào, nhiệm vụ phân loại và các tham số mạng ảnh hưởng như thế nào đến hiệu suất mô hình. Mặc dù chúng tôi đã chọn các mô hình và giá trị siêu tham số của mình theo cách thủ công, các mạng kết quả vẫn hoạt động tốt hơn các bộ phân loại truyền thống.

Đối với mục tiêu đã đề ra ban đầu “Phát hiện và phân loại âm thanh ho trên các thiết bị IoT”. Đã thực hiện được các nội dung sau:

- Trinh bày các dạng ho, các phân biệt các loại ho dựa trên các đặc trưng
- Các mô hình tiềm năng cho việc phát hiện và phân loại âm thanh ho
- Các thí nghiệm đánh giá các mô hình cho việc phát hiện ho.

Khi nghiên cứu và thực hiện đề tài,tôi đã mong muốn có thể đưa ra một phương pháp tối ưu cho việc phát hiện và phân loại âm thanh ho. Tuy nhiên, do gặp nhiều khó khăn do thời gian có hạn và dịch bệnh COVID-19 đã trở thành những cản trở lớn trong quá trình nghiên cứu đề tài.

Tôi mong muốn phần hoàn thiện của đề tài này sẽ có thể thúc đẩy cho các nghiên cứu sau đưa ra các phương pháp tối ưu hơn, xây dựng được một hệ thống hoàn chỉnh trên các thiết bị IoT phục vụ cho việc đánh giá sức khỏe con người nhằm đưa ra các chuẩn đoán nhanh và chính xác nhất.

# TÀI LIỆU THAM KHẢO

[1] Larson, E. C., et. al. : Accurate and Privacy Preserving Cough Sensing Using a Low Cost Microphone. In: Proc. of UbiComp, pp. 375-384. Beijing, 2011

[2] Birring, S. S., et al.: The Leicester Cough Monitor: Preliminary Validation of an Automated Cough Detection System in Chronic Cough. In: European Respiratory Journal, 31 (5), pp. 1013-1018

[3] Schappert, S., Burt, C.: Ambulatory Care Visits to Physician Offices, Hospital Outpatient and Emergence. In: Vital Health statistics, 13, pp. 1-66

[4] Drugman, T., et al.: Audio and Contact Microphone for Cough Detection. In: Pro c. Of INTERSPEECH, pp. 1303-1306. IEEE Press. Portland, 2012

[5] Vizel, E., et al.: Validation of an Ambulatory Cough Detection and Counting Application Using Voluntary Cough under Different Conditions. In: Cough 6(3), (2008)

[6] Kraman, S. S., et al.: Comparisons of Lung Sound Transducers Using a Bioacoustic Transducer Testing System. In: Journal of Appl Physiol., 101(2), pp. 169-176 (2006)

[7] Zheng, S., et al.: CoughLoc: Location-Aware Indoor Acoustic Sensing for Non-intrusive Cough Detection. In: Int'l Workshop on MobiSys,2011

[8] Pham, C., et al.: The Ambient Kitchen: A Pervasive Sensing Environment for Situated Services. In: Proc. of ACM Conf. on Designing Interactive Systems, Newcastle, UK, 2012

[9] Pham, C., et al.: A Wearable Sensor based Approach to Real-Time Fall Detection and Fine-Grained Activity Recognition. In: Journal of Mobile Multimedia 9, pp. 15-26 (2013)

[10] Drugman, T., et al.: Assessment of Audio Features for Automatic Cough Detection. In: Proc. of 19th European Signal Processing Conference, pp. 1289 – 1293, 2011

[11] Mark, S., Hyekyun, H., Mark, B.: Automated Cough Assessment on a Mobile Platform. In: Journal of Medical Engineering (2014)

[12] https://dantri.com.vn/suc-khoe/moi-loai-ho-mot-kieu-benh

[13] Akane Sano ; Rosalind W. Picard: Stress Recognition Using Wearable Sensors and Mobile Phones (2013)

[14] Justice Amoh; Kofi Odame: Neural Networks For Identifying Cough Sounds (2016)

[15] Jia-Ming Liu, Mingyu You, Zheng Wang, Guo-Zheng Li, Xianghuai Xu, and Zhongmin Qiu: Cough event classification by pretrained deep neural network (2015)

[16] Jianqiang Li; Zhuang-Zhuang Chen; Luxiang Huang; Min Fang; Bing Li; Xianghua Fu; Huihui Wang; Qingguo Zhao: Automatic Classification of Fetal Heart Rate Based on Convolutional Neural Network (2018)

[17] Feng Xiao; Yimin Chen; Ming Yuchi; Mingyue Ding; Jun Jo: Heart rate prediction model based on physical activities using evolutionary neural network (2010)

[18] Harish S. Bhat, Sidra J. Goldman-Mellor: Predicting adolescent suicide attempts with neural networks (2017)

[19] Aracy Pereira Silveira Balbani: Cough: neurophysiology, methods of research, pharmacological therapy and phonoaudiology (2012)

[20] S. J. Barry, A. D. Dane, A. H. Morice, and A. D. Walmsley, “The automatic recognition and counting of cough.,” Cough (London, England), vol. 2, p. 8, jan 2006.

[21] S. Matos, S. Member, S. S. Birring, I. D. Pavord, D. H. Evans, and S. Member, “Detection of Cough Sounds in Continuous Audio Recordings Using Hidden Markov Models,” vol. 53, no. 6, pp. 1078–1083, 2006.

[22] T. Drugman, J. Urbain, and T. Dutoit, “Assessment of audio features for automatic cough detection,” 19th European Signal Processing . . . , no. 32, 2011.

[23] E. C. Larson, T. Lee, S. Liu, M. Rosenfeld, and S. N. Patel, “Accurate and privacy preserving cough sensing using a low-cost microphone,” Proceedings of the 13th international conference on Ubiquitous computing - UbiComp ’11, p. 375, 2011.

[24] J. Amoh and K. Odame, “DeepCough: A Deep Convolutional Neural Network in A Wearable Cough Detection System,” in IEEE Biomedical Circuits and Systems Conference (BioCAS), pp. 1–4, IEEE, 2015.

[25] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1–32, 1997.

[26] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. Campbell, “SoundSense: scalable sound sensing for people-centric applications on mobile phones,” Proceedings of the 7th international conference on Mobile systems, applications, and services, pp. 165–178, 2009.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Advances In Neural 2Information Processing Systems, pp. 1–9, 2012.

[29] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method,” arXiv, p. 6, dec 2012.

[30] Y. N. Dauphin, H. de Vries, J. Chung, and Y. Bengio, “RMSProp and equilibrated adaptive learning rates for non-convex optimization,” arXiv preprint arXiv:1502.04390, 2015.

[31] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” The Journal of Machine Learning Research, vol. 12, pp. 2121–2159, 2011.

[32] R. Pascanu, T. Mikolov, and Y. Bengio, “Understanding the exploding gradient problem,” Computing Research Repository (CoRR) abs/1211.5063, 2012.

[33] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, “Advances inoptimizing recurrent networks,” ICASSP, IEEE International Conferenceon Acoustics, Speech and Signal Processing - Proceedings, pp. 8624–8628, 2013.

[34] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” arXiv, 2015

[35] C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, and Y. Bengio, “Batch Normalized Recurrent Neural Networks,” arXiv preprint arXiv:1510.01378, 2015.

[36] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. K. Sønderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. D. Fauw, M. Heilman, Diogo149, B. McFee, H. Weideman, Takacsg84, Peterderivaz, Jon, Instagibbs, D. K. Rasul, CongLiu, Britefury, and J. Degrave, “Lasagne: First release.,” aug 2015.

[37] C. Ittichaichareon, S. Suksri, and T. Yingthawornsuk, “Speech Recognition using MFCC,” International Conference on Computer Graphics, Simulation and Modeling, pp. 135–138, 2012.