HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐẶNG ĐÌNH QUÂN

XÁC ĐỊNH TỶ LỆ TIN XẤU TRÊN BÁO ĐIỆN TỬ TIẾNG VIỆT BẰNG PHƯƠNG PHÁP HỌC SÂU

LUẬN VĂN THẠC SĨ KỸ THUẬT
(Theo định hướng ứng dụng)

HÀ NỘI – NĂM 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

ĐẶNG ĐÌNH QUÂN

XÁC ĐỊNH TỶ LỆ TIN XẤU TRÊN BÁO ĐIỆN TỬ TIẾNG VIỆT BẰNG PHƯƠNG PHÁP HỌC SÂU

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH
MÃ SỐ: 8.48.01.01

LUẬN VĂN THẠC SĨ KỸ THUẬT
(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS. TS. TRẦN QUANG ANH

HÀ NỘI – NĂM 2020

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi. Nội dung của luận văn có tham khảo và sử dụng các tài liệu, thông tin được đăng tải trên những tạp chí và các trang web theo danh mục tài liệu tham khảo. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Hà Nội, ngày tháng năm 2020

Người cam đoan

Đặng Đình Quân

LỜI CẢM ƠN

Trong quá trình thực hiện luận văn này, học viên luôn nhận được sự hướng dẫn, chỉ bảo rất tận tình của PGS. TS. Trần Quang Anh là cán bộ trực tiếp hướng dẫn khoa học. Thầy đã giành nhiều thời gian trong việc hướng dẫn học viên cách đọc tài liệu, thu thập và đánh giá thông tin cùng phương pháp nghiên cứu để hoàn thành một luận văn cao học.

Học viên xin chân thành cảm ơn các thầy, cô giáo trong Học viện Công nghệ Bưu chính Viễn thông đã luôn nhiệt tình giúp đỡ và tạo điều kiện tốt nhất cho em trong suốt quá trình học tập tại trường.

Xin chân thành cảm ơn các anh, các chị và các bạn học viên lớp Cao học – trong Học viện đã luôn động viên, giúp đỡ và nhiệt tình chia sẻ với em những kinh nghiệm học tập, công tác trong suốt khoá học.

Học viên cũng xin chân thành cảm ơn các vị lãnh đạo và các bạn đồng nghiệp tại cơ quan đã luôn tạo mọi điều kiện tốt nhất để em có thể hoàn thành tốt đẹp khoá học Cao học này.

Em xin chân thành cảm ơn!

Hà Nội, ngày tháng năm 2020

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Từ viết tắt	Nghĩa tiếng Anh	Nghĩa tiếng Việt
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
MLP	Multilayer Perceptron	Mạng nơ-ron nhiều lớp
RNN	Recurrent Neural Network	Mạng nơ-ron hồi qui
LSTM	Long Short-Term Memory	Mạng bộ nhớ tạm dài hạn
SGD	Stochastic Gradient Descent	Thuật toán giảm độ dốc ngẫu nhiên
MSE	Mean Squared Error	Bình phương lỗi trung bình
CSDL	Database	Cơ sở dữ liệu
TF-IDF	Term Frequency – Inverse Document Frequency	Tần suất từ – tần suất văn bản nghịch đảo
TP	True Positives	Dự đoán dương tính đúng
FP	False Positives	Dự đoán dương tính sai (cảnh báo nhầm)
TN	True Negatives	Dự đoán âm tính đúng
FN	False Negatives	Dự đoán âm tính sai (bỏ sót)

DANH MỤC CÁC HÌNH VẼ VÀ BẢNG

MỞ ĐẦU

Với sự phổ biến của Internet, báo điện tử đã trở thành kênh thông tin quan trọng đối với đời sống xã hội ngày nay. Chức năng chính của báo điện tử là phản ánh mọi mặt của xã hội, cung cấp thông tin thời sự, chính xác cho độc giả. Khác với tạp chí là chủ yếu cung cấp thông tin mang tính tham khảo/học thuật về một lĩnh vực chuyên biệt, ví dụ như: tạp chí khoa học, tạp chí công nghệ, tạp chí văn học, tạp chí thể thao… [24]. Như vậy có thể nói báo điện tử là sự phản ánh về hiện thực xã hội.

Một trang thông tin điện tử (website) là hệ thống thông tin dùng để thiết lập một hoặc nhiều trang thông tin được trình bày dưới dạng ký hiệu, số, chữ viết, hình ảnh, âm thanh và các dạng thông tin khác phục vụ cho việc cung cấp và sử dụng thông tin trên Internet (Nghị định 72/2013/NĐ-CP). Từ năm 2015 đến tháng 3/2017, có 168 trang thông tin điện tử được cấp phép tại Việt Nam [25]. Báo điện tử là một loại hình báo chí được xây dựng dưới hình thức của một trang thông tin điện tử và phát hành trên mạng Internet. Tính đến tháng 6/2017, cả nước có 150 báo điện tử [26]. Chưa có số liệu thống kê chính xác về những trang thông tin điện tử tiếng Việt chưa được cấp phép đang hoạt động trên mạng Internet. Với số lượng trang thông tin điện tử như vậy, khối lượng thông tin được đăng tải cho độc giả hằng ngày là rất lớn.

Bộ Thông tin và Truyền thông (TT&TT) đã đưa ra quan điểm rằng "cái xấu xuất hiện với tỉ lệ 30% trên mặt báo nghĩa là cái xấu trở thành cái chính của xã hội; cái xấu chiếm 20% là biểu hiện cái xấu có xu hướng trở thành cái chính trong xã hội; còn cái xấu chiếm 10% tuy không phải là cái chính nhưng đủ sức tác động đến con người". Nếu tỷ lệ cái xấu đăng tải trên một tờ báo điện tử không phản ánh phù hợp với thực tế xã hội, tờ báo đó sẽ góp phần cung cấp cho độc giả cái nhìn sai lệch về thực trạng xã hội và làm "xói mòn niềm tin" của người dân [23].

Như vậy, việc đánh giá tỷ lệ cái xấu trên mặt báo điện tử là vô cùng cấp thiết. Tuy nhiên, với khối lượng thông tin khổng lồ trên báo điện tử như đã đề cập, cần thiết có một phương pháp để tự động thực hiện công việc này một cách chính xác và kịp thời. Trong luận văn này, học viên đi tìm một phương pháp hiệu quả để giải quyết vấn đề đánh giá tỷ lệ thông tin tiêu cực trên báo điện tử một cách tự động.

Vấn đề đặt ra trong luận văn là một vấn đề mới đang được Bộ TT&TT quan tâm, tìm giải pháp. Tuy nhiên, có thể dễ dàng nhận thấy bài toán cần giải nằm trong lĩnh vực phân loại văn bản. Từ một trang báo điện tử, ta có thể thu thập được những thông tin không gắn liền với một bài báo cụ thể như: số lượng bài viết được đăng trong ngày, số bài viết được đăng của từng chuyên mục, danh sách các chuyên mục… Tuy nhiên, những thông tin này không đủ để ước lượng tỷ lệ thông tin tiêu cực của cả trang báo. Như vậy, ta cần phải dựa vào lượng thông tin chính đó là tiêu đề, nội dung… của từng bài báo để xác định bài báo đó có nói về cái xấu trong xã hội hay không. Sau đó, ta tính tỷ lệ các bài báo nói về cái xấu trên tổng số các bài báo.

Trong khai phá văn bản, ngoài phân loại văn bản ra còn có các hướng nghiên cứu khác rất gần với vấn đề cần giải quyết là: trích rút chủ đề (topic/concept/entity extraction), khai phá quan điểm (opinion mining) và phân cụm văn bản (clustering). Thứ nhất, ta có thể coi vấn đề cần giải quyết là một bài toán trích rút chủ đề với 2 chủ đề (xấu, tốt). Tuy nhiên, ta không thể coi cái xấu và cái tốt là các chủ đề. Khi nói đến cùng một chủ đề, một bài viết có thể phản ánh mặt tốt trong khi bài viết khác có thể phản ánh mặt xấu. Thứ hai, mục tiêu của bài toán khai phá quan điểm là xác định quan điểm chủ quan của người viết. Tuy nhiên, cái tốt/cái xấu trong nội dung các bài báo mạng về bản chất không phải là quan điểm chủ quan (mang tính cảm xúc) mà là các thông tin thời sự khách quan. Cái xấu/cái tốt ở đây không phải là ý kiến cá nhân của tác giả bài báo mạng về một sự vật, hiện tượng, mà là một bản tin tường thuật, phản ánh chính xác một sự việc xảy ra trong xã hội. Cuối cùng, cách tiếp cận của bài toán phân cụm văn bản có thể được áp

dụng trong vấn đề này. Các bài viết từ một trang báo điện tử sẽ được phân thành 2 cụm. Tuy nhiên, cần tìm ra một độ đo sao cho các bài viết về cái xấu có khoảng cách gần nhau và cách xa các bài viết về cái tốt, đồng thời nghiên cứu thêm phương pháp để xác định cụm nào trong hai cụm chứa các bài viết nói về cái xấu.

Các phương pháp học máy thống kê cổ điển: SVM, kNN, mạng nơ-ron, LLSF (Linear Least Squares Fitting) và máy phân loại Bayes đơn giản đã được áp dụng để phân loại văn bản theo chủ đề (category) với kết quả tốt [10]. Các kỹ thuật học sâu (CNN, RNN, LSTM) tuy chưa vượt qua được các phương pháp cổ điển trong bài toán phân loại văn bản nhưng là một lựa chọn khả quan vì một số lý do. Thứ nhất, các kỹ thuật học sâu đã được chứng minh là có khả năng hiểu ngôn ngữ tự nhiên ngang bằng và thậm chí tốt hơn các phương pháp cổ điển tốt nhất [12]. Thứ hai, con người không cần tham gia vào việc lựa chọn đặc trưng, bởi vì các đặc trưng được học tự động từ dữ liệu. Cuối cùng, khi dữ liệu càng lớn thì hiệu quả của kỹ thuật học sâu càng được phát huy [6].

Từ những lý do trên, học viên lựa chọn đề tài "XÁC ĐỊNH TỶ LỆ TIN XẤU TRÊN BÁO ĐIỆN TỬ TIẾNG VIỆT BẰNG PHƯƠNG PHÁP HỌC SÂU" cho luận văn tốt nghiệp trình độ đào tạo thạc sĩ.

Mục đích, đối tượng và phạm vi nghiên cứu:

Mục đích nghiên cứu của luận văn là nghiên cứu các phương pháp học sâu dành cho dữ liệu dạng văn bản và ứng dụng vào bài toán xác định tin xấu trên báo điện tử tiếng Việt.

Đối tượng nghiên cứu của luận văn là các phương pháp học sâu dành cho dữ liệu dạng văn bản và bài toán xác định tin xấu dành cho bài báo điện tử tiếng Việt.

Phạm vi nghiên cứu của luận văn là các bài viết thuộc hai chuyên mục "đời sống" và "kinh doanh" trên báo điện tử tiếng Việt.

Phương pháp nghiên cứu:

- Về mặt lý thuyết: Thu thập, khảo sát, phân tích các tài liệu và thông tin có liên quan đến bài toán xác định tỷ lệ tin xấu trên báo điện tử tiếng Việt và các phương pháp học sâu áp dụng cho dữ liệu văn bản.

- Về mặt thực nghiệm: Xây dựng tập dữ liệu tin xấu tiếng Việt, làm thí nghiệm cài đặt và huấn luyện một số mô hình dự đoán, tổng hợp và so sánh kết quả thí nghiệm giữa các mô hình khác nhau để tìm ra ưu, nhược điểm và khả năng áp dụng của từng phương pháp.

Kết cấu của luận văn gồm 3 chương chính như sau.

Chương 1: Sơ lược về học máy, học sâu và bài toán xác định tỷ lệ tin xấu.

Chương 2: Phương pháp xác định tỷ lệ bài viết nói về cái xấu trên báo điện tử tiếng Việt.

Chương 3: Đánh giá phương pháp xác định tỷ lệ bài viết nói về cái xấu trên báo điện tử tiếng Việt.

Chương 1 – SƠ LƯỢC VỀ HỌC MÁY, HỌC SÂU VÀ BÀI TOÁN XÁC ĐỊNH TỶ LỆ TIN XẤU

1.1. GIỚI THIỆU BÀI TOÁN XÁC ĐỊNH TỶ LỆ TIN XẤU

Để xác định tỷ lệ tin xấu của một trang báo điện tử, bài toán đặt ra đó là làm sao để gán nhãn tốt/xấu cho mỗi bài viết trên trang báo đó. Thông tin trên mỗi bài báo điện tử thường bao gồm cả chữ viết, hình ảnh, âm thanh và video. Trong đó, phần lớn các bài báo điện tử có chứa cả nội dung chữ viết và hình ảnh. Nội dung video ngày càng trở lên phổ biến nhưng chưa chiếm đa số trong các trang báo điện tử. Trên hầu hết các trang báo điện tử, hình ảnh trong bài viết đều được ghi chú bằng phụ đề miêu tả nội dung bức ảnh. Trong phạm vi thời gian cho phép của luận văn, học viên lựa chọn tập trung nghiên cứu về nội dung văn bản của các bài báo.

Cảnh sát phong toả hiện trường và đưa các bình gas từ trong nhà ra ngoài. Ảnh: *Gia Chính*

Hình 1.1: Ví dụ về phụ đề của hình ảnh trên một bài báo điện tử.
(Nguồn: Báo điện tử VnExpress)

Bài toán tổng quát mà luận văn cần giải quyết đó là bài toán phân loại với một nhãn và hai lớp. Giải pháp cần đưa ra được nhãn chính xác đối với đầu vào là nội dung dạng

text của một bài báo, từ đó tính được tỷ lệ phần trăm tin xấu trong tổng số các bài viết trên một trang báo điện tử.

Sai số gây ra bởi mô hình phân loại được chia làm hai trường hợp: một tin xấu được dự đoán là tin tốt (bỏ sót) và một tin tốt được dự đoán là tin xấu (cảnh báo nhầm). Trong nhiều bài toán phân loại, tổn thất gây ra bởi hai loại sai số nói trên là khác nhau. Ví dụ trong bài toán lọc thư rác thì cảnh báo nhầm là trường hợp dự đoán sai nghiêm trọng, gây ra thiệt hại lớn. Trong khi đó điều ngược lại xảy ra ở bài toán phát hiện xâm nhập, người ta thường cho phép tỷ lệ cảnh báo nhầm cao để có thể giảm xác suất của trường hợp bỏ sót bởi vì bỏ sót hành vi xâm nhập là sai số có tổn thất lớn hơn. Với bài toán xác định tỷ lệ tin xấu, cảnh báo nhầm làm kết quả tổng hợp về tỷ lệ tin xấu của một trang báo điện tử tăng cao, khiến cho nhiều trang báo có khả năng bị vượt ngưỡng tin xấu cho phép hơn. Ngược lại, sai số bỏ sót làm tỷ lệ tin xấu bị hạ thấp, năng một trang báo bất kỳ bị vượt ngưỡng tin xấu sẽ giảm xuống. Sự cân bằng giữa hai loại sai số này có thể được kiểm soát nhờ điều chỉnh tham số của mô hình phân loại, nếu giảm tỷ lệ sai số này thì sẽ làm tăng tỷ lệ kia và ngược lại.

1.1.1. Định nghĩa về tin xấu

Tin nói về sự việc, hiện tượng tiêu cực (không phân biệt trong nước hay thế giới), nói về những sự việc mang tính chất phản cảm, không hay, không đẹp, gây tác hại đến môi trường, kinh tế, xã hội… Tin xấu là tin nói về thực trạng đáng buồn của xã hội, khó khăn của nền kinh tế, thiên tai.

Ta không thể coi cái xấu và cái tốt là các chủ đề vì khi nói đến cùng một chủ đề, một bài viết có thể phản ánh mặt tốt trong khi bài viết khác có thể phản ánh mặt xấu. Hơn nữa, cái tốt/cái xấu trong nội dung các bài báo mạng về bản chất không phải là quan điểm chủ quan (mang tính cảm xúc) của tác giả bài viết, mà là các thông tin thời sự khách quan. Cái xấu/cái tốt ở đây không phải là ý kiến cá nhân của tác giả bài báo mạng về một sự vật, hiện tượng, mà là một bản tin tường thuật, phản ánh chính xác một sự việc xảy

ra trong xã hội. Tuy vậy, khi nói về một sự việc mang tính tiêu cực, việc sử dụng những từ ngữ mang tính tiêu cực là không thể tránh khỏi. Đây là cơ sở để hình thành luận điểm rằng các mô hình học máy có khả năng phân biệt được cái tốt, cái xấu trong bài viết, đặc biệt là các mô hình học máy có khả năng nhớ được thông tin theo trục tọa độ thời gian (temporal).

Luận văn không có mục đích đưa ra định nghĩa chuẩn về tin xấu. Thay vào đó, nghiên cứu này đặt mục tiêu thử nghiệm hiệu quả của các mô hình học máy trong việc phân biệt/phát hiện tin xấu theo một định nghĩa cụ thể.

1.1.2. Phân loại văn bản

Phân loại văn bản là bài toán cổ điển và phổ biến trong khoa học máy tính nói chung và trong lĩnh vực học máy nói riêng. Mục tiêu của bài toán là xây dựng mô hình phần mềm để tự động phân loại văn bản thành hai hoặc nhiều lớp. Đây là một tác vụ được coi là dễ đối với con người nhưng khó đối với máy tính bởi sự phức tạp về logic của nó. Độ khó của bài toán phân loại văn bản phụ thuộc trực tiếp vào đối tượng dữ liệu của bài toán. Trong đó, nội dung cụ thể của văn bản, ngôn ngữ của văn bản, độ dài văn bản, kích thước tập dữ liệu, chất lượng của quá trình gán nhãn… tất cả đều góp phần quyết định độ khó của bài toán phân loại. Bài toán xác định tỷ lệ tin xấu có khối lượng dữ liệu lớn và sẵn có đó là những tin tức đã được xuất bản trên báo điện tử. Độ dài văn bản ở mức trung bình (khoảng 200 - 500 từ), dài hơn so với các ý kiến bình luận (một vài câu) nhưng ngắn hơn so với các văn bản chuyên ngành hoặc tác phẩm văn học (chẳng hạn, trong bài toán phát hiện đạo văn). Các bài báo điện tử thường thông qua quy trình soạn thảo, kiểm duyệt và xuất bản nên nhìn chung đều tuân thủ quy tắc ngữ pháp, sử dụng từ ngữ theo chuẩn mực cao so với những dữ liệu khác như các bài viết, bình luận trên mạng xã hội hoặc các review sản phẩm. Về ngôn ngữ, Tiếng Việt được xếp vào nhóm có ngữ pháp khó trong các ngôn ngữ trên thế giới.

Bài toán phân loại văn bản được giải quyết phổ biến với các phương pháp học máy. Phương pháp này đặt mục tiêu mô phỏng cách mà con người tiếp nhận và xử lý thông tin để đưa ra kết luận về loại của văn bản. Mỗi mô hình học máy sẽ mô phỏng tư duy của con người theo một cơ chế nhất định. Mô hình này có chức năng tiếp nhận và xử lý văn bản theo cơ chế đã đặt ra, và đưa ra kết luận của nó về loại của văn bản. Để giúp mô hình đưa ra được các dự đoán đúng hơn, nó sẽ được huấn luyện bằng dữ liệu mẫu. Tùy vào cấu tạo bên trong của một mô hình mà khả năng học tập của nó có thể khả nhau. Một mô hình với cơ chế không phù hợp sẽ không có khả năng ghi nhận kiến thức từ dữ liệu. Mô hình học máy không nhất thiết phải mô phỏng toàn bộ chức năng của não bộ con người, mà chỉ cần mô phỏng cơ chế đủ để nó "hiểu" được dữ liệu của bài toán.

Gần đây, học sâu đã trở thành một phương pháp phổ biến để giải quyết bài toán này. Nó đã nổi lên như một kỹ thuật học máy mạnh mẽ, có khả năng học nhiều hình thái biểu diễn khác nhau của dữ liệu hay nói cách khác đó là có thể tự động học được đặc trưng của dữ liệu với thành công vượt trội so với những phương pháp cũ. Cùng với sự thành công của học sâu trong nhiều lĩnh vực ứng dụng khác, học sâu cũng được sử dụng phổ biến trong phân loại văn bản những năm gần đây.

1.1.3. Phân tích cảm xúc

Phân tích cảm xúc hoặc khai phá quan điểm là nghiên cứu tính toán về ý kiến của con người, tình cảm, cảm xúc, đánh giá và thái độ đối với các thực thể như sản phẩm, dịch vụ, tổ chức, cá nhân, vấn đề, sự kiện, chủ đề và thuộc tính của họ. Sự khởi đầu và sự phát triển nhanh chóng của lĩnh vực này trùng khớp với các phương tiện truyền thông xã hội trên web. Lần đầu tiên trong lịch sử của ngành khoa học máy tính, chúng ta có một khối lượng lớn quan điểm được ghi lại dưới dạng dữ liệu số. Từ đầu những năm 2000, phân tích cảm xúc đã trở thành một trong những lĩnh vực nghiên cứu được quan tâm nhất trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Nó cũng được nghiên cứu rộng

rãi trong các lĩnh vực khác như khai phá dữ liệu, khai phá web, khai phá văn bản và truy xuất thông tin.

Các nhà nghiên cứu đang ngày càng trở nên quan tâm hơn trong việc tự động phân tích cảm xúc của công chúng về một chủ đề cụ thể. Thách thức của vấn đề nghiên cứu này đó là phải trích xuất được chiều hướng cảm xúc từ các dữ liệu văn bản. Đây là một bài toán trong lĩnh vực khai phá quan điểm (opinion mining) hoặc phân tích cảm xúc (sentiment analysis). Những khó khăn cụ thể trong bài toán này bao gồm: (1) tính chủ quan trong việc định nghĩa quan điểm và (2) các hiện tượng ngôn ngữ ảnh hưởng đến tính phân cực của câu từ.

Trên thực tế, do tầm quan trọng của nó đối với toàn bộ doanh nghiệp và xã hội, nó đã lan rộng từ khoa học máy tính sang khoa học quản lý và khoa học xã hội như tiếp thị, tài chính, khoa học chính trị, truyền thông, khoa học y tế và thậm chí cả lĩnh vực lịch sử. Sự phổ biến này đến từ thực tế các ý kiến là trung tâm của hầu hết các hoạt động của con người và là nhân tố chính ảnh hưởng đến hành vi của chúng ta. Niềm tin và nhận thức của chúng ta về thực tế, và những lựa chọn chúng ta đưa ra, ở một mức độ đáng kể, dựa trên cách người khác nhìn và đánh giá thế giới. Vì lý do này, bất cứ khi nào chúng ta cần đưa ra quyết định, chúng ta thường tìm kiếm ý kiến của người khác. Điều này không chỉ đúng với cá nhân mà còn đúng với các tổ chức [11].

1.2. SƠ LƯỢC VỀ HỌC MÁY

Nền tảng của trí tuệ nhân tạo là khả năng máy móc có thể nhận thức như con người nhờ việc "học" từ các ví dụ. Việc học của một cỗ máy thông minh có nhiều điểm tương đồng với quá trình học của con người. Để học một khái niệm mới, chúng ta ghi nhớ các đặc điểm của đối tượng và gắn đối tượng đó với một khái niệm mà ta được dạy. Ta hình thành khả năng khái quát khi được học nhiều bảm thể của một khái niệm nào đó. Chẳng hạn, sau khi gặp được nhiều người đàn ông và phụ nữ khác nhau, một đứa trẻ dần dần khái quát được các đặc điểm nhận dạng giữa nam giới và nữ giới. Nó hình thành khả

năng tự phân biệt giới tính của một người lạ mặt mà không cần người lớn hỗ trợ. Khả năng phân biệt của con người đôi khi có sự nhầm lẫn đến từ thiếu xót trong việc học hoặc thiếu xót trong khả năng quan sát, ghi nhớ… của chúng ta.

Học máy (machine learning) mô phỏng lại quá trình học nói trên để khiến cho phần mềm máy tính có thể học và nhận thức được các dữ liệu số (văn bản, hình ảnh, âm thanh…). Mô hình học máy là một chương trình máy tính có chứa một tập bất kỳ các tham số và có hai chức năng cơ bản là *học* và *dự đoán*. Mỗi mô hình học máy đều có mục tiêu xác định, một tác vụ cụ thể mà nó cần thực hiện (phân loại, phân cụm, phát hiện, lọc, khôi phục…). Tùy vào mục tiêu, mối với mỗi đầu vào x, mô hình học máy có nhiệm vụ tìm ra một kết quả y. Chức năng *học* có nhiệm vụ xử lý những mẫu ví dụ (gọi ngắn gọn là *mẫu*) để điều chỉnh các tham số bên trong chương trình cho khớp với đặc điểm của những mẫu ví dụ đó. Mẫu ở đây chính là đối tượng học của mô hình học máy, ví dụ: giá chứng khoán, nội dung tin tức, hình ảnh động vật, giọng nói… Chức năng dự đoán áp dụng bộ tham số trên một mẫu x để đưa ra kết quả y của mẫu đó. Như vậy, chức năng dự đoán chính là thành phần thực hiện mục tiêu của mô hình. Ta có thể biểu diễn chức năng dự đoán dưới dạng một hàm số như sau: $y = f(x; \theta)$, trong đó θ chính là bộ tham số bên trong của mô hình học máy.

Tập hợp các ví dụ mẫu được gọi là *tập dữ liệu huấn luyện* (training data). Ngày nay, có rất nhiều định dạng dữ liệu được dùng trong học máy: số liệu, văn bản, hình ảnh, âm thanh, video… Mỗi định dạng dữ liệu lại chia thành nhiều kiểu dữ liệu, ví dụ: dữ liệu văn bản có thể là dài hoặc ngắn, đơn giản (plain) hoặc có cấu trúc (structured), được viết bằng các ngôn ngữ khác nhau… Các đặc tính được chú ý của tập dữ liệu huấn luyện đó là độ lớn và tính đại diện (representativeness). Tính đại diện của tập dữ liệu là đặc tính cho phép người học có thể học được kiến thức một cách đầy đủ và tổng quan từ tập dữ liệu đó. Ví dụ, một tập dữ liệu về hình ảnh lá cây được coi là đại diện nếu nó có chứa

đầy đủ mẫu của các loại lá cây và các dạng biến thể một cách tiêu biểu nhất. Để có được tính đại diện thì điều kiện cần đó là tập dữ liệu phải đủ lớn.

Não bộ của con người có khả năng lựa chọn các đặc tính để dựa vào đó nhận dạng một đối tượng. Khi phân biệt giới tính của một người, não của chúng ta chỉ chọn ra một vài đặc điểm có sự khác biệt rõ rệt để đánh giá, và bỏ qua những đặc điểm không liên quan như: số lượng răng, màu tóc, nước da, chiều cao… Quá trình đó trong học máy được gọi là trích chọn thuộc tính. Hiệu quả của mô hình đầu ra phụ thuộc rất nhiều vào việc lựa chọn những thuộc tính tốt. Với học sâu (deep learning), quá trình trích chọn thuộc tính được tự động hóa. Điều này khiến cho những bài toán khó giải đối với học máy truyền thống có thể được giải quyết khi sử dụng học sâu. Ở các phần tiếp theo của luận văn, các khái niệm về học máy và học sâu sẽ giải thích chi tiết và đầy đủ hơn.

1.2.1. Học máy có giám sát

Hình thức phổ biến nhất trong học máy là học máy có giám sát (supervised learning). Trong học máy có giám sát, ví dụ mẫu được cung cấp kèm theo kết quả (gọi là *nhãn*) chuẩn cho chức năng học. Điều này tương tự với việc cho học sinh biết đáp án của của bài tập khi dạy học. Ta có thể biểu diễn chức năng học trong học máy có giám sát bằng công thức như sau: $\theta = g(x, \theta_0)$, trong đó g đại diện cho chức năng học, θ_0 là tập tham số trước khi học mẫu x, θ là tập tham số được thay đổi sau khi học. Giả thiết của học máy có giám sát là khi máy học một số lượng mẫu đủ lớn, tập tham số θ sẽ chứa đầy đủ kiến thức (knowledge) để máy có thể tự suy ra đáp án đúng cho các mẫu mới. Nền tảng sâu xa của học máy có giám sát là từ lý thuyết xác suất thống kê [1].

Các bài toán tiêu biểu được giải bằng phương pháp học máy có giám sát là:

- **Phân loại (classification):** Cho một mẫu x, mô hình học máy phải dự đoán một kết quả y dành cho x trong số k lớp hữu hạn. Nếu $k = 2$, ta gọi bài toán là phân

loại nhị phân. Bài toán phân loại còn có một biến thể trong đó kết quả đầu ra của mô hình là một tập các xác suất dành cho các lớp.

- **Hồi quy (regression):** Cho một mẫu x, mô hình học máy phải đưa ra kết quả dạng số liên tục. Đầu ra của mô hình có dạng liên tục thay vì rời rạc như đối với bài toán phân loại. Bài toán dự đoán giá cổ phiếu là một ví dụ về hồi quy.
- **Phát hiện hành vi bất thường (anomaly detection):** Mô hình học máy được huấn luyện bằng các hoạt động bình thường. Khi đó, nếu đầu vào x là một hoạt động bình thường, kết quả của mô hình sẽ cho thấy x đã được "học" trước đó. Khi kết quả của mô hình cho thấy đầu vào chưa được học, điều đó chứng tỏ hoạt động đầu vào là một hoạt động bất thường. Ví dụ về phát hiện hành vi bất thường là những hệ thống bảo mật và các phần mềm diệt virus máy tính.

1.2.2. Học máy không giám sát

Khác biệt lớn nhất giữa học máy không giám sát và có giám sát là sự vắng mặt của nhãn trong tập mẫu. Trong học máy không giám sát, chức năng học phải tự điều chỉnh bộ tham số mà không có nhãn chuẩn cho trước. Chính vì thiếu đi phương hướng để dựa vào khi điều chỉnh các tham số, học máy không giám sát đặt mục tiêu là tìm ra mô hình mật độ xác suất của (tập) mẫu [18]. Có thể hiểu mục tiêu đó là đi tìm các mối liên hệ giữa các mẫu trong tập dữ liệu. Ví dụ đơn giản nhất về mối liên hệ giữa các mẫu đó là khoảng cách giữa các mẫu và phân bổ trong không gian của các mẫu. Người ta có thể sử dụng học máy không giám sát để tìm ra vùng phân bổ của những dữ liệu chứa thông tin quan trọng nhất nhằm mục đích loại bỏ dữ liệu dư thừa. Học máy không giám sát có thể được sử dụng để tách nhiễu ra khỏi một tín hiệu, ví dụ như ứng dụng lọc nhiễu từ âm thanh.

Một vài bài toán được giải bằng học máy không giám sát là:

- **Phân cụm (clustering):** Cho một tập mẫu và một số k, thuật toán phân cụm phải chia tập mẫu thành k nhóm.

- **Giảm chiều dữ liệu (dimensionality reduction):** Cho dữ liệu đầu vào ở không gian $\mathbb{R}^N$, mô hình học máy phải biểu diễn dữ liệu đó ở không gian $\mathbb{R}^M$ với $M < N$ mà vẫn giữ được đặc tính của dữ liệu gốc.

1.2.3. Học máy bán giám sát

Học máy bán giám sát là trường hợp chỉ có một phần nhỏ mẫu trong tập dữ liệu huấn luyện có nhãn kèm theo. Lượng dữ liệu được gán nhãn quá nhỏ để có thể huấn luyện có giám sát một cách hiệu quả. Tuy nhiên, so với bài toán học máy không giám sát thì ít nhất ta cũng có một chút ít cơ sở để hỗ trợ cho chức năng học. Một cách tiếp cận trong hướng này là sử dụng các mẫu có nhãn để huấn luyện ra một mô hình thô, sau đó dùng mô hình chưa hoàn thiện này để gán nhãn cho những mẫu còn lại. Cuối dùng, khi tất cả các mẫu đã có nhãn, mô hình được huấn luyện như trong học máy có giám sát. Đối với cách làm này, độ chính xác của các nhãn được gán tự động là khá thấp khi mà chúng được sinh ra bởi một mô hình không tối ưu. Các nghiên cứu trong lĩnh vực học máy bán giám sát chú trọng vào việc đi tìm những cách để tăng chất lượng của quá trình gán nhãn tự động nói trên.

1.2.4. Hàm mục tiêu, hàm tổn thất, hàm chi phí

Hàm mục tiêu (objective function) là một khái niệm cơ bản trong học máy. Trong cả học máy có giám sát và không giám sát thì ta đều phải thực hiện công đoạn thiết kế hàm mục tiêu. Hàm mục tiêu chính là hàm dự đoán trong đó có chứa bộ tham số tối ưu mà ta cần đi tìm. Như vậy, hàm mục tiêu là một hàm chưa biết mà ta hy vọng có thể tìm ra. Đầu tiên, ta thiết kế hàm dự đoán với số lượng tham số và cách tổ chức, tính toán tham số mà ta giả định rằng giống với hàm mục tiêu cần tìm. Các tham số được khởi tạo ngẫu nhiên và được điều chỉnh bằng cách học từ các mẫu. Việc huấn luyện (điều chỉnh tham số) này không đảm bảo tìm ra được hàm mục tiêu bởi vì hai lý do. Thứ nhất, thiết kế của chức năng dự đoán có thể không giống với hàm mục tiêu thực sự, dẫn tới bất khả

thi trong việc mô phỏng hàm mục tiêu. Thứ hai, thông tin chứa đựng trong tập dữ liệu huấn luyện không đầy đủ nên không thể xây dựng lại hàm mục tiêu một cách toàn vẹn.

Hàm tổn thất (loss function) và hàm chi phí (cost function) là hai khái niệm cơ bản trong học máy. Đối với học máy có giám sát, hàm tổn thất là một hàm số của sự khác biệt giữa kết quả dự đoán và nhãn chuẩn. Đối với học máy không giám sát, hàm tổn thất là đặc thù đối với từng bài toán cụ thể. Hàm chi phí là một hàm tổng hợp các giá trị hàm tổn thất trên toàn bộ tập dữ liệu. Nói một cách nôm na, hàm tổn thất có phạm vi trên một mẫu đơn lẻ còn hàm chi phí có phạm vi trên toàn tập dữ liệu.

Hàm chi phí có vai trò đặc biệt quan trọng trong quá trình huấn luyện mô hình. Nhờ có nó, thuật toán huấn luyện biết được hướng điều chỉnh tập tham số sao cho giá trị của hàm chi phí giảm xuống, đồng nghĩa với việc kết quả dự đoán gần hơn với nhãn chuẩn.

1.2.5. Overfitting

Một vấn đề quan trọng trong học máy là làm sao mô hình huấn luyện ra phải hoạt động tốt trên các mẫu mới chưa từng thấy trước đây chứ không chỉ các mẫu mà mô hình đã được học. Khả năng thực hiện tốt trên các mẫu chưa được học gọi là khả năng tổng quát hóa (generalization).

Thông thường, khi huấn luyện một mô hình học máy, chúng ta có một tập huấn luyện. Chúng ta có thể tính toán hàm chi phí và điều chỉnh tập tham số để giảm giá trị này. Đến đây, việc huấn luyện thực chất là một bài toán tối ưu. Tuy nhiên, học máy khác với tối ưu ở chỗ mục tiêu của học máy là giảm giá trị hàm chi phí trên cả những mẫu mới chứ không chỉ trên tập mẫu huấn luyện. Giá trị hàm chi phí trên các mẫu nằm ngoài tập huấn luyện được gọi là giá trị lỗi thực nghiệm (testing error). Để ước lượng giá trị này, người ta thường chia dữ liệu ra hai phần là phần huấn luyện (training set) và phần thử nghiệm (testing set). Hiệu quả của mô hình trên tập dữ liệu thử nghiệm chính là điều mà chúng ta quan tâm khi ứng dụng các phương pháp học máy. Một tập thử nghiệm tốt là tập thử nghiệm nằm trong cùng không gian dữ liệu với tập huấn luyện và các mẫu

được phân chia đồng đều giữa tập thử nghiệm và tập huấn luyện. Hay nói cách khác, tập thử nghiệm tốt có tính đại diện tương tự với tập huấn luyện.

Khi một mô hình học máy có hiệu quả kém trên tập huấn luyện, ta gọi trường hợp đó là underfitting. Khi một mô hình có hiệu quả rất cao trên tập huấn luyện nhưng hiệu quả trên tập thử nghiệm lại thấp, ta gọi trường hợp đó là overfitting. Hai chiều hướng này được coi là ngược nhau và trên thực tế ta có thể điều khiển xu hướng dẫn đến hai tình huống nói trên bằng cách điều chỉnh độ lớn hay độ phức tạp (capacity) của mô hình. Mô hình quá đơn giản sẽ không có đủ khả năng ghi nhớ hết được các đặc điểm của tập dữ liệu, dẫn đến underfitting, trong khi mô hình quá phức tạp sẽ ghi nhớ cả những chi tiết quá cụ thể của dữ liệu mẫu, khiến cho mô hình mất đi tính khái quát và dẫn đến overfitting. Việc thiết kế mô hình học máy sao cho vừa đủ phức tạp để tiếp nhận kiến thức từ dữ liệu huấn luyện sẽ giúp huấn luyện ra được mô hình với tính khái quát cao.

1.3. SƠ LƯỢC VỀ HỌC SÂU

Phương pháp học máy từ lâu đã được ứng dụng trong bài toán phân loại văn bản. Tuy nhiên, độ phong phú và phức tạp của dữ liệu làm cho tỷ lệ lỗi của các mô hình học máy tăng cao. Để khắc phục vấn đề này, người ta thiết kế ra các phương pháp trích chọn thuộc tính để giữ lại những thuộc tính dễ phân loại và loại bỏ những thuộc tính gây nhiễu loạn. Chất lượng của quá trình trích chọn thuộc tính quyết định rất nhiều đến mức độ hiệu quả của một mô hình học máy. Cách làm truyền thống này tuy đã đạt được những thành công nhất định, nhưng để thiết kế được phương pháp trích chọn thuộc tính tốt là công việc yêu cầu kiến thức chuyên gia.

Ngày nay, dữ liệu huấn luyện ngày càng nhiều và tốc độ phần cứng ngày càng cao, cộng thêm sự ra đời của các phương pháp mới cho phép huấn luyện các mạng nơ-ron nhiều lớp hơn, khái niệm "học sâu" đã ra đời và trở thành một đột phá trong lĩnh vực học máy có giám sát. Học sâu đề cập tới việc ứng dụng các mạng nơ-ron sâu để giải quyết các bài toán nhận diện, phân loại… và đã đạt được nhiều thành công đáng kể [6].

Ở phần này, học viên trình bày các kiến thức cơ bản về học sâu làm nền tảng cho các phương pháp được áp dụng thực nghiệm trong các chương sau của luận văn.

1.3.1. Mạng nơ-ron

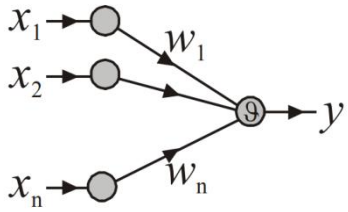
Ứng dụng có thể coi là cơ bản và phổ biến nhất trong lĩnh vực machine learning chính là mạng nơ-ron nhân tạo (gọi tắt là mạng nơ-ron). Lấy cảm hứng từ cấu trúc của bộ não sinh học, mạng lưới thần kinh bao gồm một số lượng lớn các đơn vị xử lý thông tin (được gọi là nơ-ron) được tổ chức thành các lớp, hoạt động đồng nhất với nhau. Nó có thể được huấn luyện để thực hiện các tác vụ, như phân loại văn bản, bằng cách điều chỉnh các trọng số của kết nối giữa các nơ-ron trong mạng.

Học sâu là ứng dụng của mạng nơ ron vào các nhiệm vụ học tập bằng cách sử dụng mạng nhiều lớp. Nó có thể khai thác nhiều sức mạnh hơn từ mạng nơ ron trong việc học kiến thức từ dữ liệu so với trước kia, khi mà mạng nơ ron chỉ có thể áp dụng được với một hoặc hai lớp và một lượng nhỏ dữ liệu.

Dựa trên cấu trúc liên kết mạng, các mạng nơ ron nói chung có thể được phân loại thành các mạng feed-forward và mạng nơ-ron hồi quy (recurrent) / đệ quy (recursive). Các cấu trúc mạng cũng có thể được trộn lẫn và kết hợp với nhau. Các cấu trúc mạng nơ-ron khác nhau sẽ được đề cập trong các phần sau của luận văn này.

1.3.1.1. Perceptron

Mô hình mạng nơ-ron đầu tiên được công bố bởi một nghiên cứu do hải quân Hoa Kỳ tài trợ [19] vào năm 1958. Nó có tên là perceptron, được tạo ra để mô phỏng hoạt động não bộ con người. Thực chất, perceptron là một mạng nơ-ron một lớp đơn giản, chỉ có khả năng giải những bài toán tuyến tính hoặc “học” trên không gian dữ liệu tuyến tính.



Hình 1.2: Cấu trúc của perceptron

Một perceptron tương đương với hàm sau:

$$f(x) = \sum_{i=1}^{n} w_i x_i + \alpha$$

và

$$y = \begin{cases} 1, & f(x) \geq 0 \\ 0, & f(x) < 0 \end{cases}$$

Việc huấn luyện một perceptron nghĩa là đi tìm các giá trị trong vector $w \in \mathbb{R}^N$ phù hợp để từ một vector đầu vào $x \in \mathbb{R}^N$, perceptron đưa ra được giá trị y chính xác nhất. Mô hình perceptron phù hợp cho ứng dụng học máy có giám sát để giải bài toán phân loại nhị phân.

1.3.1.2. Mạng nơ-ron truyền thẳng nhiều lớp

Các mạng nơ-ron truyền thẳng sâu – Multilayer Perceptron (MLP) – là mô hình học sâu điển hình nhất. Để giải những bài toán không tuyến tính, ví dụ như hàm XOR, thì perceptron là không đủ. Mạng nơ-ron với nhiều lớp ẩn (hidden layers) có thể giải những bài toán không tuyến tính [20]. Như vậy, mạng nơ-ron thoát được những hạn chế của perceptron về việc chỉ biểu diễn được các quan hệ tuyến tính. Cùng với phát hiện này, mạng nơ-ron trở lại với nhiều ứng dụng đột phá.

Mục tiêu của MLP là để mô phỏng một hàm f^* nào đó. Ví dụ một hàm phân loại $y = f^*(x)$ ánh xạ đầu vào x thành một lớp y. MLP mô phỏng hàm này dưới dạng

$y = f(x; \theta)$ và học các tham số θ sao cho hàm f mô phỏng hành vi của hàm f^* một cách gần đúng nhất có thể. Một mô hình như vậy được gọi là mạng truyền thẳng bởi vì thông tin đi qua mạng từ x, qua các lớp tính toán trong hàm f, tới đầu ra y. Trong mô hình không tồn tại những kết nối truyền ngược (feedback) – khi đầu ra của mô hình được truyền ngược lại làm đầu vào của chính nó. Khi mạng nơ-ron truyền thẳng có bao gồm các kết nối truyền ngược, nó được nhắc đến bằng một cái tên khác đó là mạng nơ-ron hồi quy (recurrent neural network, RNN). Mạng RNN sẽ được đề cập ở những phần sau của chương này.

Mạng truyền thẳng là những mô hình quan trọng đối với học máy. Chúng được sử dụng rộng rãi trong các ứng dụng thương mại. Ví dụ, mạng nơ-ron tích chập (convolutional neural network, CNN) là một loại mạng truyền thẳng được ứng dụng nhiều trong tác vụ nhận diện khuôn mặt từ hình ảnh.

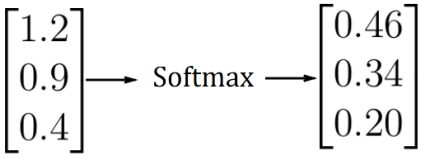
1.3.2. Hàm kích hoạt

1.3.2.1. Softmax

Hàm softmax có công thức như sau:

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

Hàm softmax lấy đầu vào là một vector số và chuẩn hóa các thành phần của vector đó về trong khoảng [0; 1] sao cho tổng của vector đầu ra là 1. Hình 1.3 minh họa ví dụ trong đó hàm softmax chuyển hóa một vector 3 phần tử.



Hình 1.3: Minh họa cách hoạt động của hàm kích hoạt Softmax.
Nguồn: https://towardsdatascience.com/@ManishChablani

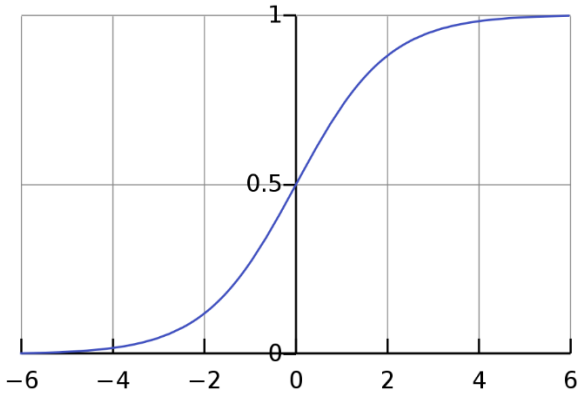
Hàm softmax được sử dụng cho lớp đầu ra của mạng nơ-ron có nhiều hơn một nơ-ron. Giá trị thô của mỗi nơ-ron của lớp đầu ra có thể ở một khoảng giá trị rất khác với khoảng [0; 1] nhưng trong các ứng dụng mạng nơ-ron thì ta cần giá trị ở trong khoảng này bởi vì nó thể hiện được xác suất của lớp (class) tương ứng với nơ-ron trong lớp đầu ra.

1.3.2.2. Sigmoid

Hàm sigmoid là hàm kích hoạt được sử dụng trên một nơ-ron. Hàm sigmoid cũng thường được dùng để làm hàm kích hoạt cho lớp đầu ra của mạng nơ-ron, đặc biệt là mạng perceptron. Hàm sigmoid có công thức như sau:

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$

Hàm sigmoid có đồ thị là một đường cong đối xứng tại điểm [0, 0.5] (Hình 1.4). Hàm có tác dụng chuyển một giá trị số thực về trong khoảng [0; 1].



Hình 1.4: Đồ thị của hàm sigmoid.
Nguồn: https://en.wikipedia.org/wiki/Sigmoid_function

1.3.2.3. Hàm tanh

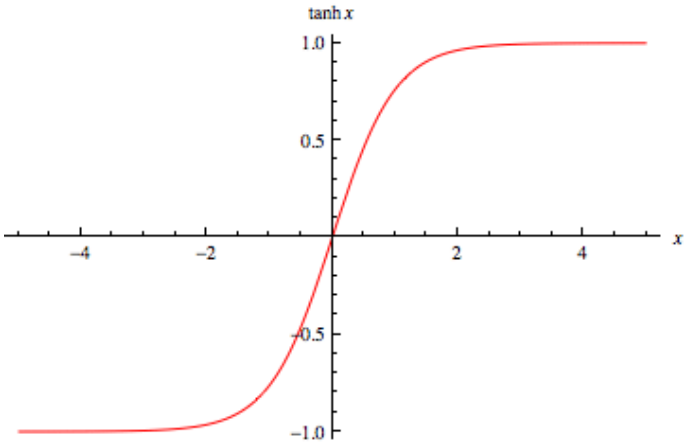
Hàm tanh có công thức như sau:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Hàm tanh chuyển giá trị đầu vào về khoảng [-1; 1], đối xứng tại gốc tọa độ (Hình 1.5). Tuy cần nhiều bước tính toán để tính ra được giá trị của hàm tanh, đạo hàm của nó lại rất dễ tính toán và hoàn toàn không phụ thuộc vào giá trị đầu vào x, mà chỉ phụ thuộc vào giá trị đầu ra.

$$\tanh'(x) = 1 - \tanh(x)^2$$

Đây là một tính chất thú vị dành cho những thuật toán tối ưu dựa trên GD. Bởi vì sự dễ dàng trong tính toán đạo hàm của nó và tính chất đối xứng, tanh được lựa chọn nhiều để làm hàm kích hoạt trong các mạng nơ-ron.



Hình 1.5: Đồ thị của hàm tanh.

1.3.3. Huấn luyện mạng nơ-ron

1.3.3.1. SGD

Gần như tất cả ứng dụng về học sâu đều sử dụng một thuật toán rất quan trọng: tối ưu giảm độ dốc ngẫu nhiên (Stochastic Gradient Descent). SGD được phát triển từ thuật toán gốc Gradient Descent. Một vấn đề thường gặp trong học máy đó là cần phải có các tập huấn luyện lớn để có thể huấn luyện được mô hình có tính khái quát hóa tốt. Tuy nhiên, các tập huấn luyện lớn đồng nghĩa với việc mất nhiều thời gian tính toán hơn. Hàm chi phí (cost function) trong các thuật toán học máy thường có thể được phân tích

dưới dạng tổng của các hàm tổn thất (loss function) áp dụng trên tất cả các mẫu của tập huấn luyện. Khi kích thước tập huấn luyện tăng lên rất nhiều, thời gian để thực hiện một bước tối ưu trong Gradient Descent trở nên quá dài.

Nguyên lý hoạt động của SGD coi độ dốc (gradient) là một giá trị ước lượng. Giá trị này có thể được ước tính một cách gần đúng với một tập con nhỏ của tập mẫu mẫu. Cụ thể, trên mỗi bước của thuật toán GD, chúng ta lấy ngẫu nhiên một số lượng mẫu nhất định (gọi là mini-batch) rải đều trên tập huấn luyện (uniformly distributed). Kích thước của mini-batch thường là một con số nhỏ, thường từ một mẫu cho tới một vài trăm mẫu. Con số này thường không thay đổi khi kích thước tập huấn luyện tăng [1]. Như vậy, ta có thể huấn luyện mô hình trên tập dữ liệu kích thước hàng tỷ mẫu trong khi chỉ phải tính toán độ dốc trên vài trăm mẫu ở mỗi bước cập nhật. Tiếp theo, thuật toán SGD sẽ di chuyển tập trọng số trong không gian trọng số theo chiều xuống dốc dựa vào độc dốc vừa tìm được. Tốc độ di chuyển tập trọng số được quy định bởi một giá trị gọi là *learning rate*.

Algorithm: Stochastic gradient descent (SGD)

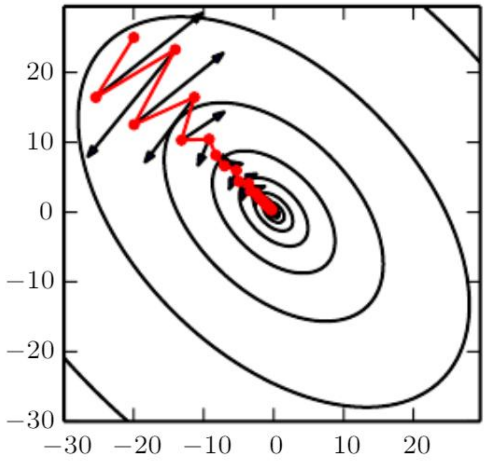
Require: Learning rate schedule $\epsilon_1, \epsilon_2, \ldots$
Require: Initial parameter $\boldsymbol{\theta}$
  $k \leftarrow 1$
  while stopping criterion not met **do**
    Sample a minibatch of m examples from the training set $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$.
    Compute gradient estimate: $\hat{\boldsymbol{g}} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i L(f(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), \boldsymbol{y}^{(i)})$
    Apply update: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon_k \hat{\boldsymbol{g}}$
    $k \leftarrow k + 1$
  end while

Hình 1.6: Pseudo-code của thuật toán SGD [1]

Tham số *learning rate* là một tham số rất quan trọng cho thuật toán này. SGD thường được mô tả với tốc độ học cố định. Trong thực tế, cần phải giảm dần tốc độ học

tập theo thời gian [1]. Khi mô hình tiếp cận với điểm tối ưu thì tốc độ di chuyển cần giảm xuống để tránh việc mô hình không thể đến được điểm tối ưu do bước nhảy quá lớn.

Momentum cũng là một cơ chế được thiết kế để tăng tốc độ học cho SGD [4]. Nguyên lý của momentum là lưu giữ lại hướng di chuyển của một số bước cập nhật gần nhất trong quá khứ để điều chỉnh hướng đi hiện tại của SGD, giữ cho SGD không bị đi lệch hướng cũ và có thể đi sâu vào vị trí tối ưu cục bộ. Tác dụng của momentum được minh họa trong Hình 1.7.



Hình 1.7: Minh họa tác dụng của momentum trong SGD [1].
Đường có mũi tên là hướng mà đáng lẽ SGD sẽ chọn nếu không sử dụng momentum.

Thuật toán GD nói chung thường được coi là chậm hoặc không đáng tin cậy. Trong quá khứ, việc áp dụng GD cho các vấn đề tối ưu hóa những hàm không lồi được coi là vô căn cứ và bất khả thi. Ngày nay, chúng ta biết rằng các mô hình học máy hoạt động rất tốt khi được huấn luyện bằng các phương pháp dựa trên GD. Thuật toán tối ưu hóa GD không thể đảm bảo việc tìm được giải pháp tối ưu cục bộ trong một khoảng thời gian hợp lý, nhưng nó thường tìm được một giá trị rất nhỏ của hàm chi phí, trong thời gian đủ nhanh để được coi là hữu ích. Ngoài ứng dụng trong học sâu, thuật toán SGD còn có nhiều ứng dụng quan trọng khác bên ngoài. Nó là cách phổ biến nhất để huấn luyện các mô hình tuyến tính (linear models) lớn trên các tập dữ liệu rất lớn. Đối với kích thước

mô hình cố định, chi phí tính toán của SGD không phụ thuộc vào kích thước tập dữ liệu. Trong thực tế, ta thường thường sử dụng một mô hình lớn hơn khi kích thước tập huấn luyện tăng lên, nhưng điều này là không bắt buộc. Số lượng bước (nói cách khác, thời gian huấn luyện) cần thiết để đạt được trạng thái hội tụ (convergence) thường tăng theo kích thước tập huấn luyện. Tuy nhiên, khi kích thước tập huấn luyện tăng dần tới vô hạn, mô hình có xu hướng là sẽ hội tụ trước khi SGD xét hết các mẫu của tập huấn luyện. Khi đó, thời gian huấn luyện sẽ không tăng cùng với kích thước của tập huấn luyện nữa.

1.3.3.2. Backpropagation

Để có thể tối ưu một mạng nơ-ron bằng SGD thì ta tinh chỉnh các tham số trong ma trận tham số của các lớp mạng dựa vào đạo hàm của đầu ra đối với đầu vào (phương pháp gradient descent). Như vậy, để huấn luyện một mạng nhiều lớp, ta phải tính được đạo hàm của đầu ra đối với giá trị đầu vào. Việc này có thể thực hiện dễ dàng đối với mạng nơ-ron một lớp nhưng lại không đơn giản đối với mạng có nhiều lớp. Nghiên cứu [20] chỉ ra rằng một mạng như vậy được huấn luyện một cách hiệu quả dựa trên một quy trình đơn giản được gọi là *back-propagation* (việc tính đạo hàm chuỗi).

Khi một mạng nơ-ron truyền thẳng nhận đầu vào x và tạo ra kết quả y, thông tin được truyền qua các lớp mạng. Giá trị của vector đầu vào được chuyển đổi dần dần qua từng lớp mạng, quá trình này được gọi là *forward propagation*. Khi có kết quả đầu ra y, ta có thể tính được chi phí $\delta = J(\theta)$. Quá trình *back-propagation* thực hiện ngược lại, từ giá trị hàm chi phí quay trở lại các lớp mạng từ cuối về đầu, cho tới lớp đầu vào, để tính đạo hàm của hàm chi phí đối với từng tham số của từng lớp. Thuật toán back-propagation dựa trên nguyên lý đạo hàm chuỗi trong giải tích, được phát biểu như sau:

Xét $z = f(y)$ và $y = g(x)$, ta có: $d(z)dx = d(z)dy * d(y)dx$.

Bằng cách tính đạo hàm chuỗi như vậy, trên lý thuyết, đạo hàm của hàm chi phí có thể được tính cho tất cả các trọng số có tham gia vào việc tính toán ra kết quả. Tuy nhiên,

với giới hạn về độ chính xác của kiểu dữ liệu float thì điều này không đúng khi mạng nơ-ron có nhiều lớp.

1.3.3.3. Hàm kích hoạt ReLU

ReLU là viết tắt của cụm từ Rectified Linear Unit, là một hàm kích hoạt phi tuyến tính được dùng phổ biến trong các mạng nơ-ron sâu, thay thế các hàm *sigmoid* hoặn *tanh* trước đây. Công thức của hàm *ReLU* là $g(z) = \max\{0, z\}$. Hàm ReLU giữ được giá trị đạo hàm lớn trong quá trình backpropagation nên nó không gặp phải vấn đề đạo hàm biến mất (vanishing gradient) như các hàm kích hoạt khác. Cộng thêm với cách tính đạo hàm đơn giản và không bị chặn trên bởi giá trị 1.0, hàm *ReLU* giúp cho việc huấn luyện mạng nơ-ron nhiều lớp trở nên đơn giản và nhanh chóng.

1.3.3.4. Adam

Tham số learning rate trong SGD là một trong những tham số quan trọng và cũng rất khó để tối ưu. Để tăng tốc độ và hiệu quả cho SGD, cơ chế momentum đã được thêm vào. Tuy nhiên, với momentum, ta có thêm tham số để tinh chỉnh, vấn đề trở lên phức tạp hơn. Từ vấn đề này, một loạt biến thể của SGD được ra đời với nhiều giá trị *learning rate* cho từng chiều không gian và các giá trị này cũng tự động thích nghi trong quá trình huấn luyện. Những phương pháp kể trên được gọi là nhóm thuật toán tối ưu thích nghi (adaptive optimizers).

Adam [14] là một trong những thuật toán tối ưu thích nghi như vậy, được giới thiệu vào năm 2014. Một cách khái quát, Adam kết hợp khả năng tự động thích nghi learning rate cho từng trục tọa độ với cơ chế momentum. Trong thực nghiệm, thuật toán Adam có tốc độ tìm ra kết quả nhanh hơn SGD nhưng lại có xu hướng dễ bị overfitting hơn và mô hình đào tạo bởi Adam thường không tốt bằng mô hình huấn luyện bởi SGD [21].

1.3.4. Một số hàm chi phí

1.3.4.1. MSE

Mean Squared Error (trung bình cộng của bình phương sai số) là hàm chi phí được sử dụng rất phổ biến trong học máy. Nó có công thức như sau:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Trong đó y_i là nhãn của mẫu thứ i trong tập huấn luyện, $\hat{y}_i$ là kết quả dự đoán của mô hình dành cho mẫu đó.

1.3.4.2. Categorical Cross Entropy

Hàm chi phí này áp dụng cho đầu ra của mô hình phân loại dạng vector one-hot.

$$CE = -\sum_{i=1}^{C} y_i * \log(\hat{y}_i)$$

Trong công thức trên, y_i là phần tử thứ i trong vector nhãn, $\hat{y}_i$ là phần tử thứ i trong vector kết quả dự đoán. Sparse Categorical Cross Entropy là phiên bản Categorical Cross Entropy dành cho trường hợp tập nhãn được biểu diễn dưới dạng số thứ tự của các lớp (ví dụ [1, 0, 1, 0, 0, 1]) còn đầu ra của mô hình lại có dạng vector one-hot. Binary Cross Entropy là phiên bản Categorical Cross Entropy dành cho bài toán phân loại nhị phân.

Chương 2 – PHƯƠNG PHÁP XÁC ĐỊNH TỶ LỆ BÀI VIẾT NÓI VỀ CÁI XẤU TRÊN BÁO ĐIỆN TỬ TIẾNG VIỆT

2.1. BIỂU DIỄN THUỘC TÍNH

Mỗi thuật toán, mỗi phương pháp đều có định dạng dữ liệu đầu vào nhất định. Để một văn bản có thể sử dụng được với một thuật toán, văn bản cần được biểu diễn bằng định dạng phù hợp cho thuật toán đó. Ngoài ra, các cách biểu diễn thuộc tính khác nhau

còn có những tính chất khác nhau và các ưu điểm riêng. Trong phần này, học viên tập trung hơn vào việc miêu tả những phương pháp biểu diễn thuộc tính phổ biến cho các phương pháp học sâu.

2.1.1. Character-level, word-level

Một mẫu ví dụ được biểu diễn dưới dạng tập hợp các thuộc tính. Cách làm phổ biến nhất là biểu diễn mẫu dưới dạng vector. Dạng vector của mẫu có thể có kiểu dữ liệu rời rạc (số nguyên) hoặc liên tục (số thực). Mỗi thuộc tính có thể đại diện cho một từ hoặc một ký tự trong văn bản. Trong ngôn ngữ tự nhiên, từ là đơn vị ngôn ngữ nhỏ nhất có ý nghĩa. Những ký tự đứng đơn lẻ không có khả năng mang ý nghĩa. Tuy vậy, mạng CNN [2] và nhiều phương pháp khác [5] đã được áp dụng thành công cho bài toán phân loại văn bản ở cấp độ ký tự. Những phương pháp này vẫn cho kết quả thực nghiệm tốt mặc dù rất khó để giải thích cơ chế để tạo ra kết quả đó. Các phương pháp học sâu để phân loại văn bản ở cấp độ ký tự thường sử dụng cấu trúc mạng nhiều lớp với số lượng tham số rất lớn để khai thác sức mạnh xử lý của phần cứng. Một kết luận chung có thể đưa ra được đó là sự phân bổ từ ngữ cũng chính là được tạo ra từ phân bổ của các ký tự cấu thành nên các từ ngữ đó. Vì vậy, nếu một phương pháp dựa trên thống kê có hiệu quả ở cấp độ từ ngữ thì nó cũng sẽ có hiệu quả ở cấp độ ký tự.

2.1.2. One-hot encoding

Có thể nói one-hot encoding là cách biểu diễn thuộc tính nguyên thủy đối với các mạng nơ-ron. Cách biểu diễn này còn được gọi là cách biểu diễn “thưa thớt” (sparse representation) bởi vì một vector thuộc tính thường có rất nhiều phần tử, chỉ một vài phần tử trong đó có giá trị 1, còn lại đều là 0. Trong một thời gian rất dài, các nghiên cứu về truy xuất thông tin đã sử dụng cách biểu diễn này một cách rất phổ biến. One-hot encoding cũng góp phần giới hạn khả năng của các mô hình học máy trước kia bởi vì nó tốn nhiều bộ nhớ và giữ lại được ít thông tin hữu ích của văn bản. Các kỹ thuật đánh giá mức độ quan trọng của từ ngữ (term weighting) mà điển hình là TF-IDF [15] thường

được sử dụng để thay thế one-hot encoding trong nhiều phương pháp học máy truyền thống [10].

2.1.3. Word Embedding

Các phương pháp học sâu dựa trên mạng nơ-ron hồi quy (RNN) thường sử dụng đầu vào ở cấp độ từ ngữ bởi vì có một kỹ thuật cho phép huấn luyện ra những vector số thực có thể dùng để đại diện cho từ ngữ. Kỹ thuật đó gọi là word embedding [16], ra đời cùng với sự bùng nổ của các phương pháp học sâu.

Nguyên lý cơ bản của word embedding đó là sử dụng một số từ ở phía trước và phía sau của một từ w bất kỳ để huấn luyện một ma trận tham số của một mạng nơ-ron sao cho mạng này có khả năng dự đoán từ phía sau hoặc phía trước của từ w. Tuy nhiên, mục đích chính của quá trình này không phải là để huấn luyện ra một máy dự đoán, bởi vì việc đó là bất khả thi khi mà đối với một từ có quá nhiều nhãn khác nhau dành cho nó. Ma trận thuộc tính của mô hình nói trên có kích thước là N × M với N là kích thước bảng từ vựng và M là một tham số tùy biến. Mỗi dòng của ma trận tương ứng với một từ trong từ điển và với mỗi từ đó ta có một vector gồm M giá trị trọng số. Mỗi khi một cặp giá trị đầu vào và đầu ra (x, y) với x là từ gốc và y là một từ ở gần nó trong câu được đưa vào mô hình để huấn luyện, vector trọng số tương ứng với từ x lại được cập nhật để quan hệ giữa nó và từ y trở nên gần gũi hơn. Điều tương tự xảy ra với gặp (y, x) bởi vì hai từ này đứng cạnh nhau. Cuối cùng, vector ứng với x và y trở thành hai vector có quan hệ gần gũi với nhau. Chính những vector trọng số có độ dài M đó là mục tiêu của phương pháp word embedding.

Trên lý thuyết, word embedding có ưu điểm vượt trội hơn so với one-hot encoding và term weighting. Vị trí tương đối giữa các từ được mô phỏng lại trong word embedding, những từ trái nghĩa sẽ có vị trí đối xứng trong không gian vector M chiều, những từ gần nghĩa sẽ có vị trí gần nhau và các phép tính không gian áp dụng trên các vector gần như thể hiện được nghĩa của các từ. Ví dụ [16]:

vector(“king”) - vector(“man”) + vector(”woman”) ≈ vector(“queen”)

2.1.4. Word2Vec

Một ứng dụng nổi tiếng của word embedding là thuật toán word2vec và bộ vector từ ngữ tiếng Anh được huấn luyện sẵn của Google. Các vector từ ngữ trong tập word2vec có độ dài 300 phần tử và được huấn luyện từ tập dữ liệu Google News có chứa 100 tỷ từ. Bởi vì các tính chất ưu việt của bộ vector được huấn luyện sẵn này mà nó đã được sử dụng trong rất nhiều nghiên cứu liên quan đến ngôn ngữ tự nhiên [11] [5]. Thuật toán word2vec cũng được nhiều nhà nghiên cứu phân tích và giải thích.

2.2. CÁC CẤU TRÚC MẠNG NƠ-RON SÂU

2.2.1. CNN

Mạng nơ-ron tích chập (Convolutional Neural Network) là một kiến trúc mạng nơ-ron đặc biệt dùng để xử lý các dữ liệu dạng lưới, ví dụ như hình ảnh 2D hoặc dữ liệu số theo chu kỳ thời gian (time-series data). Mạng nơ-ron tích chập là mạng nơ-ron có ứng dụng một toán tử có tên là tích chập (convolution) trong ít nhất một lớp mạng.

Toán tử tích chập là phép toán áp dụng một cửa sổ (hoặc lõi) *w* trên một vector hoặc ma trận *z* có kích thước các chiều đều lớn hơn hoặc bằng cửa sổ *w* nói trên. Cửa sổ *w* được di chuyển lướt trên bề mặt của *z* và tích vô hướng của *w* với từng phần bề mặt tương ứng trên *z* được tính để tạo ra *z*’ có kích thước nhỏ hơn *z*.

Cửa sổ *w* đóng vai trò như một bộ lọc hoạt động trên *z*. Giá trị của các phần tử trong *w* quyết định tính chất của bộ lọc *w*.

2.2.1.1. Lớp tích chập

Một lớp tích chập điển hình trong các mạng nơ-ron có các tham số sau:

- **Số lượng bộ lọc:** Trên cùng một ma trận, ví dụ như một bức ảnh, ta có thể áp dụng toán tử tích chập với nhiều bộ lọc để tạo ra nhiều phiên bản *z*’ có đặc điểm

khác nhau tùy thuộc vào tính chất của từng bộ lọc. Điều này giống như việc nhìn nhận một vấn đề từ nhiều khía cạnh hoặc nhìn nhận một bức ảnh ở nhiều trạng thái màu sắc khác nhau vậy.

- **Kích thước cửa sổ:** Bộ lọc có kích thước lớn chứa nhiều tham số hơn và từ đó sẽ có hành vi phức tạp hơn.

2.2.1.2. Pooling

Kỹ thuật pooling có tác dụng làm giảm độ phức tạp của dữ liệu đầu vào bằng cách chọn lấy một giá trị duy nhất từ một cửa sổ. Pooling cũng sử dụng các cửa sổ có kích thước xác định nhưng khác với lớp tích chập, cửa sổ của lớp pooling không chứa trọng số trong mình mà thay vào đó nó được gắn với một hàm tổng hợp kết quả, ví dụ như hàm max hoặc hàm avg. Khi thực hiện kỹ thuật pooling, ta thay giá trị tại một vị trí bằng giá trị được tổng hợp từ những giá trị xung quanh nó trong phạm vi kích thước của cửa sổ pooling. Khi hàm max được sử dụng ở một lớp pooling, ta gọi lớp mạng đó là một lớp *max pooling*.

2.2.2. RNN

Khác với mạng nơ-ron truyền thẳng, trong mạng nơ-ron hồi quy (recurrent neural network, RNN) tồn tại các lớp mà đầu ra của nó được dùng làm đầu vào của chính nó. Hay nói cách khác, cấu trúc mạng hình thành một vòng tròn khép kín. Như vậy, dữ liệu đầu vào sẽ đi nhiều lần qua cùng một hàm. Việc đầu ra được đưa trở lại làm đầu vào như vậy giống như nó được đi qua nhiều lớp (giống nhau) của mạng nơ-ron. Cấu trúc mạng như thế không chỉ có tiềm năng *mô phỏng được những logic phức tạp* (sâu, nhiều lớp) mà còn có *tính khái quát rất cao* vì một bộ tham số (parameters) được sử dụng cho tất cả các bước biến đổi dữ liệu. Thêm nữa, với dữ liệu dạng chuỗi (ví dụ: văn bản) thì tất cả các đặc trưng (các từ) của dữ liệu đầu vào đều được áp dụng cùng một tập tham số đó. Trên lý thuyết, RNN có khả năng đưa ra kết quả dựa vào chuỗi tất cả các dữ liệu đầu vào trước đó [1]. Nói cách khác, RNN có khả năng nhớ được dữ liệu đầu vào trong quá

khứ. RNN có "trí nhớ", giống như việc chúng ta đọc một cuốn sách từ đầu đến cuối. Ta hiểu được phần kết của cuốn sách bởi vì ta đã đọc qua và ghi nhớ các đoạn trước đó. Như vậy, mạng RNN có tiềm năng mô phỏng quá trình đọc hiểu của con người.

2.2.3. Dropout

Khi huấn luyện mạng nơ-ron sâu, ta đi tìm giải pháp ở một không gian rất rộng lớn và phức tạp. Ngoài việc có rất nhiều điểm tối ưu cục bộ, hay còn gọi là "bẫy" dành cho các thuật toán tối ưu dựa vào nguyên lý xuống dốc (GD-based) khiến cho việc tìm giải pháp tối ưu trở nên khó khăn thì còn có vấn đề về overfitting. Đó là tình huống khi mà mô hình quá phù hợp với dữ liệu huấn luyện nhưng chưa chắc đã tốt trong thực tế. Một trong những nguyên nhân dẫn đến overfitting đó là sự phụ thuộc lẫn nhau giữa các trọng số trong một lớp mạng. Dropout là kỹ thuật sinh ta để khắc phục tình trạng này. Nguyên lý của Dropout đó là ngẫu nhiên bỏ qua một phần nơ-ron của một lớp mạng (tạm thời "tắt" các nơ-ron đó ở một bước) để làm giảm sự phụ thuộc lẫn nhau giữa các nơ-ron trong một lớp. Khi đó việc huấn luyện được tiếp diễn với sự tham gia của những nút còn lại. Điều đó mô phỏng lại khá gần với hoạt động của não bộ, không phải lúc nào não bộ cũng sử dụng toàn bộ các nơ-ron thần kinh.

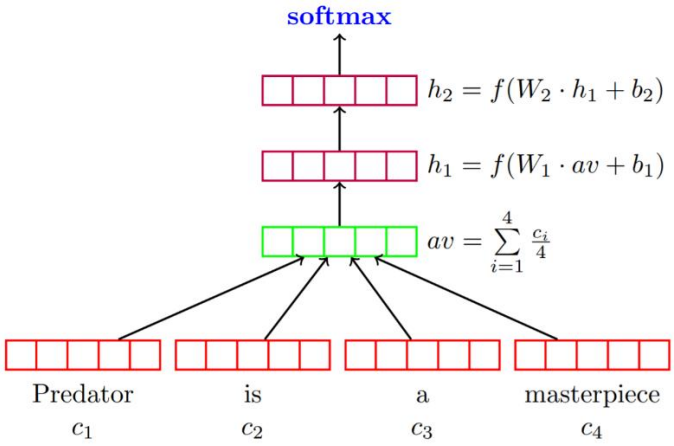
Một kỹ thuật khác để tránh tình trạng overfitting là kỹ thuật dừng sớm (early stopping) [1]. Khi huấn luyện mạng nơ-ron, ta thường theo dõi quá trình huấn luyện bằng cách thử nghiệm mô hình trên tập dữ liệu thử nghiệm sau mỗi epoch. Khi quan sát thấy tỷ lệ phù hợp giữa hiệu quả trên tập huấn luyện và tập thử nghiệm là khi mà ta nên dừng quá trình huấn luyện và sử dụng bộ trọng số tại thời điểm đó.

2.3. MỘT SỐ PHƯƠNG PHÁP PHÂN LOẠI VĂN BẢN BẰNG HỌC SÂU

Phần này sẽ tóm tắt một số nghiên cứu khoa học về ứng dụng các phương pháp học sâu để phân loại văn bản. Các mô hình mạng nơ-ron sâu được chia thành các nhóm khác nhau, trong đó nổi bật là nhóm các mạng nơ-ron truyền thẳng [2] [12] và nhóm các mạng nơ-ron hồi qui [9]. Đối với mỗi nhóm mô hình, các kỹ thuật khác nhau như word embedding [16], attention [5]… lại được ứng dụng thêm để khắc phục điểm hạn chế và tăng hiệu quả của mô hình.

Mạng nơ-ron truyền thẳng là những mô hình đơn giản nhất được ứng dụng để mô phỏng văn bản. Tuy vậy, những mô hình này vẫn đạt được độ chính xác cao khi thử nghiệm trên một số tập dữ liệu. Khi ứng dụng những mô hình này, văn bản được coi là một tập từ ngữ (bag-of-words). Các từ ngữ có thể được biểu diễn dưới dạng *one-hot encoding*, hoặc có dưới dạng vector từ ngữ (word embedding). Những phương pháp phổ biến để thực hiện kỹ thuật word embedding là word2vec [16] và GloVe [17]. Khi sử dụng word embedding, dạng biểu diễn vector của một văn bản thường được tạo ra bằng cách tính tổng hoặc trung bình cộng của tất cả các vector từ ngữ trong văn bản. Cách làm này được dựa trên giả thiết rằng vector từ ngữ có khả năng chứa đựng thông tin về nghĩa của từ và ý nghĩa của từ ngữ trong các vector đó có tính định lượng và có thể thực hiện các phép toán trên các vector này.

Một ví dụ về mạng MLP dùng cho phân loại văn bản là Deep Average Network [13] với cấu trúc được minh họa trong Hình 2.1. Bằng việc tính trung bình cộng của các word vector, mô hình này đã bỏ qua thông tin về thứ tự sắp xếp của các từ trong văn bản. Bằng việc thử nghiệm trên các tập dữ liệu chuẩn về phân loại văn bản bằng tiếng Anh như tập dữ liệu phê bình phim của Rotten Tomatoes, tập dữ liệu Stanford Sentiment Treebank và tập dữ liệu IMDB, mô hình [13] cho kết quả tuy không phải là vượt trội nhưng có thể so sánh với nhiều phương pháp khác có độ phức tạp cao hơn.



Hình 2.1: Mô hình mạng MLP với đầu vào dạng word vector dùng cho phân loại văn bản.
Nguồn: [13].

Các mô hình dựa trên mạng RNN cho phép xử lý văn bản dưới dạng chuỗi các từ ngữ. Những mô hình này được thiết kế để học sự phụ thuộc giữa các từ trong câu hoặc trong văn bản cũng như cấu trúc văn bản [5]. Tuy nhiên, mạng RNN nguyên bản không có hiệu quả cao trong việc hiểu văn bản dài và thường cho hiệu quả kém hơn so với các mạng truyền thẳng. Tuy nhiên, trong số các loại mạng RNN thì mạng LSTM vượt trội hơn cả vì nó có khả năng nắm bắt được ràng buộc giữa các từ trong văn bản dài hơn rất nhiều nhờ vào việc đưa thêm các ô nhớ để nhớ thêm nhiều giá trị cũng như đưa vào 3 loại "cổng" (cổng vào, cổng ra, cổng quên) để điều tiết luồng của thông tin ra vào mỗi đơn vị LSTM (LSTM cell).

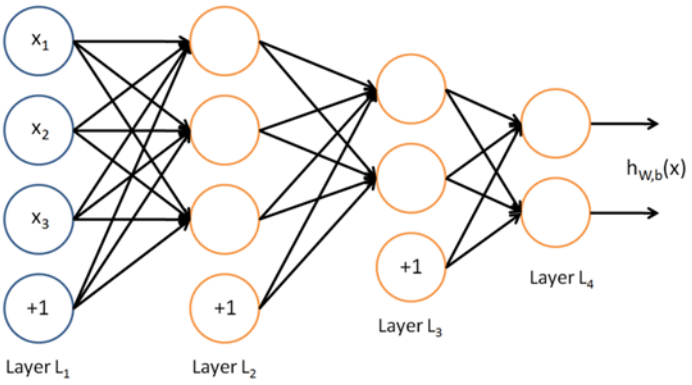
Phương pháp Tree-LSTM [3] kết hợp nhiều đơn vị LSTM thành cấu trúc dạng cây để nắm bắt thông tin về ngữ nghĩa phức tạp trong văn bản. Các tác giả của [3] cho rằng cấu trúc dạng cây có nhiều ưu điểm hơn việc kết hợp các đơn vị LSTM theo mạch nối tiếp bởi vì ngữ pháp trong ngôn ngữ tự nhiên thường được phân tích theo dạng cây, trong đó nhiều đơn vị ngữ pháp nhỏ kết hợp lại thành đơn vị ngữ pháp lớn hơn.

Trong khi mạng RNN nhận diện đặc trưng theo thời gian thì mạng CNN lại nhận diện các dấu hiệu hiện hữu trên không gian [6]. Một trong những mô hình mạng CNN đầu tiên được ứng dụng cho bài toán phân loại văn bản là mô hình DCNN [7]. Mô hình

này có mục tiêu là phân loại câu. Nó sử dụng các lớp pooling có kích thước động (dynamic k-max pooling). Đầu vào của mạng DCNN là một ma trận được hợp thành từ word vector của các từ trong một câu. Sau một số lớp tích chập và max pooling, phần đuôi của mạng DCNN bao gồm một lớp dày đặc với hàm kích hoạt softmax.

2.4. PHƯƠNG PHÁP MLP

MLP thường được nhắc đến là loại mạng nơ-ron cơ bản nhất (vanilla). Mạng MLP được cấu tạo từ các lớp mạng kết nối toàn phần (fully connected). Trong các bộ công cụ học sâu, lớp kết nối toàn phần được nhắc đến với cái tên *dense layer* (lớp dày đặc). Mỗi nơ-ron của một lớp dày đặc kết nối với tất cả các nơ-ron của lớp trước đó. Mỗi nơ-ron của lớp dày đặc cũng sử dụng một hàm kích hoạt không tuyến tính. Thông thường, các nơ-ron của cùng một lớp dày đặc sử dụng chung một hàm kích hoạt.



Hình 2.2: Minh họa cấu trúc mạng MLP với các lớp dày đặc và các kết nối mang trọng số giữa các nơ-ron.

Trong phương pháp này, học viên sử dụng mạng MLP với 4 lớp ẩn. Lớp đầu vào có kích thước 3000 nơ-ron. Lớp ẩn thứ nhất là lớp dày đặc có 128 nơ-ron, sử dụng hàm kích hoạt *ReLU*. Lớp ẩn thứ hai là lớp dày đặc có 32 nơ-ron, sử dụng hàm kích hoạt *ReLU*. Lớp ẩn thứ ba là lớp Dropout với tỷ lệ 0.5. Lớp đầu ra có 1 nơ-ron, sử dụng hàm kích hoạt *sigmoid*.

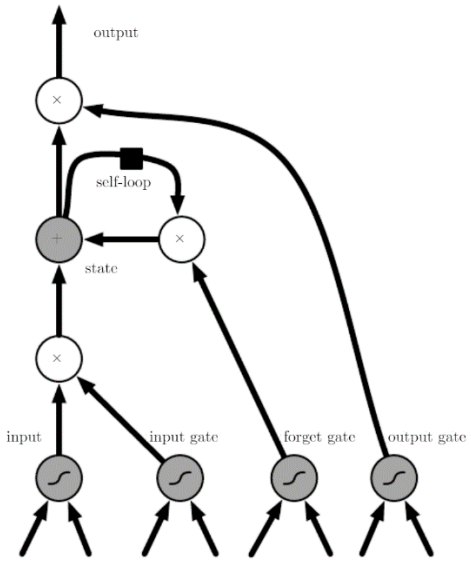
Dữ liệu đầu vào được biểu diễn ở dạng one-hot encoding, lấy 3000 từ xuất hiện nhiều nhất trong bảng từ vựng, sau khi đã tiền xử lý dữ liệu. Chi tiết về quá trình tiền xử lý dữ liệu được đề cập ở Chương 3.

Hai thuật toán tối ưu được sử dụng là thuật toán SGD có áp dụng momentum với tham số *learning_rate* = 0.03 và *momentum* = 0.6 và thuật toán Adam với các tham số mặc định. Hàm chi phí là hàm Mean Squared Error (MSE). Ba độ đo để đánh giá hiệu quả của mô hình là Accuracy, Recall và Precision.

2.5. PHƯƠNG PHÁP LSTM

Phương pháp thứ hai được học viên áp dụng cho bài toán là mô hình mạng LSTM với đầu vào có độ dài thay đổi. Mạng LSTM đã được giới thiệu từ năm 1997 [8] nhưng đến nay nó vẫn là một trong những cấu trúc học sâu được nhắc đến nhiều nhất, đặc biệt là trong những bài toán về ngôn ngữ tự nhiên. LSTM được biết đến với khả năng học được sự ràng buộc trên chuỗi dữ liệu có độ dài lớn. Mạng LSTM đã đạt được thành công trong các bài toán về dịch thuật tự động, nhận diện chữ viết tay, nhận diện giọng nói [1].

Trong luận văn này, học viên thử nghiệm trên bài toán xác định tin xấu để xác định hiệu quả của phương pháp LSTM. Hình 2.3 mô tả cấu trúc của một tế bào LSTM (LSTMCell), mỗi đơn vị này có khả năng mô phỏng và ghi nhớ đặc tính của một chuỗi đầu vào dạng số. Với chuỗi đầu vào dạng các word vector có kích thước M phần tử thì cần có M tế bào như vậy, mỗi tế bào có nhiệm vụ đảm nhiệm dữ liệu trên một chiều không gian. Để các tham số trong những tế bào này có định hướng để thay đổi, ta gắn vector chứa đầu ra của M tế bào với một lớp output. Với output dạng 0 (tin xấu) và 1 (tin bình thường) thì hàm kích hoạt sigmoid là một lựa chọn phù hợp. Một lớp Dropout ở giữa đầu ra của M tế bào và lớp output có thể được sử dụng để tăng tính độc lập cho các tế bào LSTM và giảm khả năng mô hình bị overfitting.



Hình 2.3: Cấu trúc của một đơn vị (cell) trong mạng LSTM.
Nguồn: [1].

Có hai phương án khả thi dành cho lớp đầu vào phía trước của lớp LSTM. Cách thứ nhất đó là trực tiếp sử dụng lớp Embedding để vừa huấn luyện word vector, vừa huấn luyện các tế bào LSTM. Cách thứ hai là sử dụng word vector được huấn luyện từ trước (pre-trained). Đối với tình huống dữ liệu huấn luyện có nhiều và dữ liệu có chất lượng tốt, phương án sử dụng word vector được huấn luyện từ trước có nhiều tiềm năng hơn. Huấn luyện word vector là một quá trình không giám sát nên khối lượng dữ liệu lớn không gây trở ngại vì không phải tốn thời gian gán nhãn cho dữ liệu. Với điều kiện dữ liệu huấn luyện còn ít, học viên tiến hành thí nghiệm để đánh giá hiệu quả giữa hai phương án nói trên trong Chương 3.

2.6. PHƯƠNG PHÁP BI-LSTM-CNN

Cấu trúc RNN rất phù hợp để xử lý văn bản có độ dài thay đổi. Một văn bản được cấu thành bởi chuỗi các từ ngữ. Đối với mạng RNN, tất cả các từ của văn bản không được tiếp nhận cùng một lúc ở lớp đầu vào. Thay vào đó, mạng RNN đọc từng từ một, từ đầu tới cuối văn bản, và cho kết quả đầu ra sau khi đã đọc từ cuối cùng. Mỗi lần đọc

một từ như vậy được gọi là một bước thời gian (timestep). Đầu ra của mạng RNN thay đổi theo từng bước thời gian và dữ liệu đầu ra có thể được ghi lại theo lịch sử bước thời gian để tạo thành một ma trận với một chiều thể hiện bước thời gian và chiều kia là chiều của vector đặc trưng (trong trường hợp này đó là word vector). Các tác giả của [9] cho rằng việc hầu hết các phương pháp trước đó chỉ sử dụng vector đầu ra của mạng RNN ở bước thời gian cuối cùng là sự bỏ phí những đặc tính hữu ích. Thay vì ứng dụng pooling trên 1 chiều không gian thì [9] đề xuất lấy toàn bộ output của các bước thời gian để hình thành một ma trận và ứng dụng pooling trên ma trận đó (2D pooling). Đến đây, ta nhận được một ma trận với kích thước cố định, biểu diễn đặc trưng của văn bản. Từ đây, ta có thể xử lý văn bản giống như xử lý một hình ảnh và có thể áp dụng các kỹ thuật từ mạng CNN vào đó.

LSTM được biết đến với khả năng học được sự ràng buộc trên chuỗi dữ liệu có độ dài lớn. Tuy nhiên, đối với bài toán phân loại văn bản, ta cần mô phỏng các ràng buộc với độ dài lớn hơn. Chính vì vậy, phương pháp này ứng dụng cấu trúc mạng LSTM hai chiều – Bidirectional LSTM, viết tắt là BI-LSTM. Nghiên cứu [5] cho thấy BI-LSTM có khả năng hiểu văn bản dạng ngôn ngữ tự nhiên tốt hơn so với LSTM nguyên bản.

Ngoài ra, trong phương pháp này, học viên còn áp dụng thêm lớp Flatten để chuyển đầu ra của lớp Pooling từ 2D thành 1D và kỹ thuật Dropout để tránh tình trạng overfit. Với cấu trúc mạng sâu và phức tạp, thuật toán tối ưu với learning rate thích nghi cho từng trục tọa độ trong không gian tìm kiếm – Adam [14] – được lựa chọn để giúp tăng tốc quá trình huấn luyện.

Chương 3 – ĐÁNH GIÁ PHƯƠNG PHÁP XÁC ĐỊNH TỶ LỆ TIN XẤU TRÊN BÁO ĐIỆN TỬ TIẾNG VIỆT

3.1. TẬP DỮ LIỆU

3.1.1. Phạm vi dữ liệu thử nghiệm

Dữ liệu thử nghiệm được lấy từ chuyên mục "Thời sự" của báo điện tử VnExpress (*https://vnexpress.net*). Học viên lựa chọn chuyên mục này để lấy dữ liệu thử nghiệm bởi vì thông qua khảo sát nhanh thì học viên nhận thấy đây là chuyên mục có tỷ lệ tin xấu và tin bình thường cân bằng. Những chuyên mục khác như "Văn hóa", "Giải trí"… có tỷ lệ tin xấu ở mức rất thấp. Điều này sẽ ảnh hưởng đến việc huấn luyện các mô hình học máy. Học viên không lựa chọn toàn bộ các chuyên mục trên báo điện tử VnExpress bởi vì thời gian làm luận văn cũng như năng lực của bản thân không cho phép thực hiện nghiên cứu ở quy mô lớn hơn.

3.1.2. Thu thập dữ liệu

Dữ liệu được thu thập bằng công cụ lấy tin tự động do học viên tự phát triển. Công cụ này gồm có 2 phần là phần URL Crawler (lấy đường dẫn tin) và phần Content Crawler (lấy nội dung tin). Chức năng URL Crawler tự động tải các trang của chuyên mục "Thời sự" trên báo điện tử VnExpress với đường dẫn có dạng:

```
https://vnexpress.net/thoi-su-p{x}
```

Trong đó `{x}` là số trang và được công cụ tự động tăng tiến. Mỗi đường dẫn tin tức có dạng:

```
https://vnexpress.net/tieu-de-tin-{d}.html
```

Trong đó `{d}` là id của bài báo trên VnExpress. Công cụ lưu lại id của bài báo để tránh lấy tin tức trùng lặp vào tập dữ liệu. Chức năng Content Crawler lấy mã nguồn HTML của từng bài viết về, dựa vào một số dấu hiệu nhận biết được lập trình thủ công

bởi học viên để tự động lấy tiêu đề, tóm tắt, nội dung, thời gian và tên tác giả rồi sau đó lưu dữ liệu vào một cơ sở dữ liệu MySQL.

Thời gian giữa hai lần lấy tin được điều chỉnh là khoảng 10 giây để tránh công cụ lấy tin bị hệ thống bảo vệ DDoS của VnExpress chặn.

3.1.3. Xử lý & gán nhãn dữ liệu

Khi đã có tập dữ liệu thô trong CSDL, học viên tiến hành loại bỏ những tin trùng lặp bằng cách sắp xếp tin tức theo tiêu đề, những tin có tiêu đề trùng lặp sẽ được đọc để xác định trùng lặp, sau đó tin bị trùng sẽ được xóa bỏ.

Học viên gán nhãn thủ công cho tất cả các bài viết với hai lớp là "Tin xấu" và "Tin bình thường". Việc gán nhãn được tiến hành với các tiêu chí được soạn trước. Những tin được gán nhãn là "Tin xấu" thỏa mãn một trong những điều kiện sau:

- Tin nói về sự việc, hiện tượng tiêu cực (không phân biệt trong nước hay thế giới).
- Tin nói về những sự việc mang tính chất phản cảm, không hay, không đẹp.
- Tin phản ánh về sự kiện gây tác hại xấu đến môi trường, kinh tế, xã hội.
- Tin nói về thực trạng đáng buồn của xã hội, khó khăn của nền kinh tế, thiên tai, dịch bệnh.

Một số ví dụ về tin xấu:

- Sơ chế tôm chết thành tôm nõn để bán
- Thủy điện chặn dòng, hơn 100 ha cây trồng 'khát' nước
- Cháy tàu chở dầu, hai người chết
- Nước máy ở Bến Tre nhiễm mặn
- Mưa đá ở nhiều tỉnh
- Rác ùn ứ nhiều nơi ở Quảng Nam
- Không khí Hà Nội ở mức 'rất xấu'

Trong khi gán nhãn cho các bài viết, học viên kết hợp rà soát và chỉnh lại những bài viết bị tình trạng một đoạn văn lặp hai lần cũng như loại bỏ tên tác giả còn sót lại ở cuối bài viết mà công cụ tự động chưa loại bỏ hết. Toàn bộ quá trình gán nhãn được hỗ trợ bởi công cụ gán nhãn trên giao diện web do học viên tự phát triển. Sau khi gán nhãn, học viên thực hiện kiểm tra lại toàn bộ tin đã gán nhãn để đảm bảo nhãn gán đúng theo các tiêu chí đã đặt ra.

Dữ liệu được sau đó được loại bỏ định dạng HTML. Tiêu đề tin, phần tóm tắt và phần nội dung được gộp chung vào thành một văn bản với tiêu đề là câu đầu tiên, sau đó đến phần tóm tắt và cuối cùng là nội dung tin. Tất cả dữ liệu được chuyển thành dạng chữ viết thường (lower-case) nhằm mục đích đơn giản hóa việc tách từ. Những từ viết hoa trong tiếng Việt rơi vào hai trường hợp đó là từ ở đầu câu và tên riêng. Trong hai trường hợp này, các từ ở đầu câu chiếm đa số. Hơn nữa, từ góc nhìn của con người, ngay cả khi tất cả các từ được chuyển về dạng chữ thường thì việc hiểu văn bản vẫn không bị ảnh hưởng nhiều.

Loại bỏ các các dấu chấm, phẩy, xuống dòng… (punctuations), ký tự đặc biệt thông thường và những ký tự đặc biệt hiếm gặp như dấu nháy đơn, nháy kép và chấu 3 chấm cách điệu do chức năng Auto Correct của Microsoft Word, dấu cách không xuống dòng (non-breaking space)… Sau bước này, các văn bản trong tập dữ liệu chỉ còn lại từ ngữ và dấu cách.

Để phục vụ công đoạn word embedding với thuật toán word2vec ở các phương pháp dựa trên mạng LSTM, nội dung các bài báo được tách thành từng câu dựa vào việc chia nhỏ văn bản ra bằng các ký tự dấu chấm câu (bao gồm chấm than, chấm hỏi, ba chấm…).

Sau khi thực hiện các bước thu thập và xử lý dữ liệu nêu trên, tập dữ liệu cuối cùng bao gồm tổng số 8546 bài viết. Số bài viết được gán nhãn tin xấu theo tiêu chí: 5200 bài viết.

3.2. THIẾT KẾ THÍ NGHIỆM

Toàn bộ thí nghiệm mô tả trong phần này được thực hiện trên bộ thư viện TensorFlow và thư viện Gensim (*https://radimrehurek.com/gensim*) trên ngôn ngữ lập trình Python. TensorFlow (*https://tensorflow.org*) là một bộ công cụ dành cho học sâu đến từ hãng Google và được sử dụng rất rộng rãi trên thế giới. Gensim là bộ thư viện về mô phỏng chủ đề (topic modeling) trên ngôn ngữ Python nhưng trong thí nghiệm chỉ dùng một phần của Gensim đó là mô hình Word2Vec.

3.2.1. Thí nghiệm 1

Phương pháp LSTM và BI-LSTM-CNN đều sử dụng kỹ thuật word embedding để chuyển hóa từ ngữ trong văn bản thành các vector dày đặc. Tuy nhiên, có hai phương án khác nhau để thực hiện word embedding. Cách thứ nhất là đưa trực tiếp lớp mạng Embedding vào mô hình học sâu để huấn luyện các vector từ ngữ và huấn luyện khả năng phân loại đồng thời. Cách thứ hai là huấn luyện bộ vector từ ngữ trước bằng một tập dữ liệu lớn (huấn luyện không giám sát) rồi sử dụng bộ vector đó để biểu diễn các bài viết làm đầu vào cho mô hình. Nhằm khảo sát hiệu quả giữa hai phương án nói trên, học viên tiến hành thí nghiệm 1 như mô tả sau đây.

Mô hình LSTM được sử dụng để thử nghiệm trên cùng tập dữ liệu như mô tả ở phần trước. Học viên cấu hình để mỗi mô hình lần lượt sử dụng hai phương án dữ liệu đầu vào. Với phương án sử dụng bộ vector huấn luyện trước, các bài viết được chia ra thành đơn vị câu. Mô hình Word2Vec trong bộ công cụ gensim (*https://radimrehurek.com/gensim*) được sử dụng để thực hiện word embedding với các tham số sau:

- size = 128. Kích thước vector từ ngữ. Mỗi vector từ ngữ thu được sau khi huấn luyện sẽ có độ dài 128 phần tử số thực.

- window = 5. Kích thước cửa sổ quét. Mỗi từ sẽ được huấn luyện với 5 từ phía trước và 5 từ phía sau của nó.
- min_count = 1. Tần số tối thiểu của từ được huấn luyện là 1, đồng nghĩa với việc huấn luyện tất cả các từ. Giá trị này được chọn bởi vì kích thước tập dữ liệu huấn luyện nhỏ.

Với thí nghiệm này, để đánh giá được cả độ bao phủ và độ tin cậy của kết quả dự đoán của hai mô hình, học viên sử dụng hai độ đo là recall và precision. Kỹ thuật *k-fold cross validation* với $k = 5$ được thực hiện để đánh giá kết quả trong thí nghiệm này.

3.2.2. Thí nghiệm 2

Với ba phương pháp được trình bày trong Chương 2 của luận văn, học viên tiến hành thí nghiệm này nhằm tìm ra phương pháp phù hợp nhất cho bài toán xác định tỷ lệ tin xấu trên báo điện tử tiếng Việt. Thử nghiệm trên dữ liệu là một cách khách quan để đánh giá chất lượng của mô hình học máy.

Tuy được đánh giá trên cùng tập dữ liệu nhưng trong 3 phương pháp thì MLP có cách biểu diễn dữ liệu đầu vào khác với hai phương án còn lại. MLP có thể sử dụng nhiều cách biểu diễn đầu vào khác nhau nhưng MLP không thể nhận dữ liệu đầu vào dạng chuỗi theo trục tọa độ thời gian như hai phương pháp LSTM và BI-LSTM-CNN.

Từ kết quả của thí nghiệm 1, cách sử dụng vector từ ngữ huấn luyện sẵn với word2vec tỏ ra ưu thế hơn so với cách đưa thêm lớp Embedding vào mô hình. Vì vậy, trong thí nghiệm này, hai mô hình LSTM và BI-LSTM-CNN đều sử dụng vector từ ngữ huấn luyện sẵn bởi word2vec. Các thông số cài đặt mô hình MLP đã được mô tả trong thí nghiệm 1. Cấu hình dùng để huấn luyện bộ từ vựng vector word2vec cũng được mô tả trong thí nghiệm 1.

Đối với cả hai mô hình LSTM và BI-LSTM-CNN, một mẫu tin tức đều được giới hạn độ dài tối đa là 200 từ. Mẫu tin có độ dài ngắn hơn 200 từ sẽ được độn thêm (padding)

các vector với toàn giá trị 0 để cho đủ độ dài 200. Như vậy, đầu vào của hai mô hình này đều là các mảng có kích thước $B \times L \times E$ với B là kích thước tập dữ liệu, L là độ dài của mỗi mẫu tin ($L = 200$) và E là độ dài vector từ ($E = 128$).

Mô hình LSTM được cấu hình với đầu vào có độ dài thay đổi bằng cách sử dụng lớp Masking của thư viện TensorFlow (tf.keras.layers.Masking). Lớp này có tác dụng tạm dừng lớp LSTM phía sau nó khi chuỗi đầu vào không còn dữ liệu. Lý do ta phải sử dụng lớp Masking đến từ hạn chế của việc huấn luyện mô hình mặc định của TensorFlow. Lớp LSTM của thư viện Keras vốn có hỗ trợ lớp đầu vào với độ dài thay đổi, nhưng chức năng huấn luyện mô hình lại huấn luyện mô hình theo loạt (batch) và phải chuyển hóa loạt dữ liệu huấn luyện thành kiểu dữ liệu Tensor. Kiểu dữ liệu Tensor không cho phép các phần tử có độ dài khác nhau, điều này là để đảm bảo hiệu năng tính toán. Việc sử dụng lớp Masking vẫn đảm bảo được mục tiêu đó là có độ dài của các mẫu thay đổi mà không phải lập trình lại một loạt công cụ huấn luyện trên TensorFlow.

Sau lớp Masking là lớp LSTM với số lượng đơn vị ẩn (hidden units) là 128, bằng với kích thước vector từ ngữ. Đây là cấu hình được sử dụng rộng rãi và cho kết quả tốt trong nhiều bài toán [1].

Sau lớp LSTM là lớp Dropout với tham số 0.3 để giảm khả năng bị overfitting. Cuối cùng là lớp đầu ra dày đặc (Dense) với 01 nơ-ron sử dụng hàm kích hoạt sigmoid. Cấu trúc phần đuôi của mô hình này chính là một mạng perceptron với lớp đầu vào có kích thước 128, lấy dữ liệu đầu ra của lớp LSTM làm dữ liệu đầu vào.

Mô hình BI-LSTM-CNN có lớp LSTM được bọc ngoài bằng lớp Bidirectional để biến lớp LSTM thành hai chiều, giúp cho nó có khả năng đọc hiểu văn bản theo cả hai hướng. Lớp LSTM cũng được cấu hình để trả về toàn bộ ma trận trọng số bên trong nó với tham số $\mathrm{return\_sequences} = \mathrm{True}$. Tham số này có tác dụng biến đầu ra của mạng LSTM từ dạng vector trở thành dạng ma trận. Ma trận đầu ra này thực chất chính là việc ghép nối các vector ở mỗi bước thời gian (timestep) tạo thành. Như vậy đầu ra của lớp

LSTM sẽ có kích thước 200×128 và bởi vì nó được bọc ngoài bởi lớp Bidirectional nên đầu ra của cả lớp BI-LSTM là gấp đôi kích thước nói trên (200×256).

Tiếp sau lớp BI-LSTM là lớp tích chập Conv2D nhưng lớp tích chập của Keras được thiết kế dùng để xử lý dữ liệu hình ảnh, đối với mỗi pixel hình ảnh định dạng RGB lại có 3 số nguyên thể hiện 3 màu sắc cơ bản. Như vậy đầu vào của lớp Conv2D yêu cầu thêm một chiều không gian nữa. Để đầu ra của lớp BI-LSTM tương thích với đầu vào của lớp Conv2D thì học viên sử dụng một lớp Reshape làm trung gian ở giữa. Lớp Reshape thực chất không làm thay đổi tính chất của dữ liệu. Sự thay đổi về cách bố trí dữ liệu này là để phù hợp với cách đọc dữ liệu của lớp Conv2D. Lớp Conv2D có số lượng bộ lọc (filter) là 1 với kích thước cửa sổ lọc là 2×2 (theo như cấu hình trong [9]).

Sau lớp tích chập là một lớp MaxPool2D (max pooling) với kích thước cửa sổ là 2×2. Lớp max pooling có tác dụng lọc những đặc trưng nổi bật ra từ kết quả của các bộ lọc của lớp tích chập. Lớp MaxPool2D có đầu ra là một ma trận hai chiều.

Đầu ra của lớp MaxPool2D trở thành đầu vào của một lớp dày đặc có 2 nơ-ron sử dụng hàm kích hoạt *softmax*. Kết quả đầu ra của cả mô hình là một vector dạng one-hot có 2 phần tử. Hàm chi phí MSE được sử dụng trong quá trình huấn luyện mô hình bằng thuật toán Adam [14].

Đối với 3 mô hình trong thí nghiệm này, độ đo recall, precision, accuracy và điểm số F_1 được sử dụng làm độ đo chung để so sánh các mô hình. Độ đo accuracy và F_1 được chọn bởi vì trong bài toán xác định tỷ lệ tin xấu trên báo điện tử, sai số fp và sai số fn có ảnh hưởng không khác nhau. Mục tiêu cuối cùng của việc phân loại là để ước lượng tỷ lệ phần trăm của tin xấu trong toàn bộ các bài viết. Kỹ thuật *k-fold cross validation* với $k = 5$ được thực hiện để đánh giá kết quả trong thí nghiệm này.

3.2.3. Các độ đo để đánh giá kết quả

Có bốn tiêu chí được sử dụng để đánh giá kết quả của các mô hình trong luận văn. Các tiêu chí đó lần lượt là: accuracy, recall, precision, F_1. Những tiêu chí này được tính toán dựa trên bốn chỉ số kết quả của bài toán phân loại nhị phân đó là: true positive, false positive, true negative, false negative. Trong bài toán xác định tin xấu, ta coi kết quả tin xấu là dương tính và tin bình thường là âm tính. tp là trường hợp kết quả dự đoán là dương tính cho mẫu có nhãn dương tính. fp là khi kết quả dự đoán là dương tính cho mẫu có nhãn âm tính. tn là trường hợp dự đoán đúng cho mẫu có nhãn âm tính. fn là trường hợp dự đoán sai cho mẫu có nhãn âm tính.

Bảng 3.1: Bảng chân lý cho các trường hợp kết quả dự đoán

	Tin xấu (nhãn)	**Tin bình thường (nhãn)**
Tin xấu (dự đoán)	True positive	False positive
Tin bình thường (dự đoán)	False negative	True negative

Tiêu chí accuracy có công thức như sau:

$$\mathrm{ACC} = \frac{\mathrm{tp} + \mathrm{tn}}{N}$$

Trong đó N là tổng số mẫu trong tập dữ liệu.

Tiêu chí recall là tỷ lệ dự đoán đúng trên tổng số các mẫu dương tính. Recall có ý nghĩa thể hiện độ bao phủ của kết quả dự đoán. Recall càng cao thì càng nhiều tin xấu được xác định. Recall có công thức như sau:

$$\mathrm{Recall} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}}$$

Tuy nhiên có những tình huống khi mà mô hình dự đoán sai rất nhiều nhưng recall có giá trị cao. Đó là khi fn nhỏ nhưng fp lại rất lớn. Bởi vì fp không ảnh hưởng đến recall

nên ngay cả khi hầu hết các dự đoán dương tính đều là dự đoán sai thì recall vẫn có thể có giá trị gần với 100%. Chính vì vậy, recall thường được đi kèm với precision để đánh giá mô hình một cách toàn vẹn hơn. Tiêu chí precision là tỷ lệ dự đoán đúng trên toàn bộ các dự đoán dương tính. Tiêu chí này có ý nghĩa thể hiện độ tin cậy của kết quả dự đoán. Precision có công thức như sau:

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}$$

Trong trường hợp ví dụ trên, khi recall đạt 100% bởi vì $\text{fn} = 0$ và fp lớn, khi đó precision sẽ có giá trị rất nhỏ, gần bằng 0.

Tiêu chí F_1 thường được sử dụng như là một sự cân bằng giữa hai tiêu chí recall và precision. Với một mô hình dự đoán không hoàn hảo, khi một trong hai tiêu chí recall hoặc precision đạt mức rất cao (100%) thì tiêu chí kia sẽ có giá trị thấp. Một báo cáo kết quả thí nghiệm chỉ sử dụng một trong hai tiêu chí sẽ dẫn đến kết quả không khách quan. Tiêu chí F_1 được tổng hợp từ recall và precision sẽ giúp loại bỏ những tình huống như vậy. Công thức của điểm số F_1 như sau:

$$\text{F}_1 = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

Ngoài tiêu chí F_1, một tiêu chí tổng quan hơn cũng thường được sử dụng đó là điểm số F_β. Tiêu chí F_β cho phép đặt trọng số khác nhau cho các sai số loại fp và fn, để dùng cho các bài toán mà sai số fp và fn có chi phí khác nhau. Tham số β được dùng để xác địch tỷ lệ chi phí giữa hai loại sai số. Tiêu chí F_1 chính là một trường hợp của tiêu chí F_β với $\beta = 1$. Công thức của F_β như sau:

$$\text{F}_\beta = \frac{(1 + \beta^2) \times \text{tp}}{(1 + \beta^2) \times \text{tp} + \beta^2 \times \text{fn} + \text{fp}}$$

Trong công thức trên, fn được coi là có chi phí cao gấp β lần so với fp.

3.2.4. Kiểm chứng chéo

Kiểm chứng chéo (cross-validation) là một kỹ thuật phổ biến dùng để đánh giá hiệu quả của các mô hình phân loại trong phương pháp học máy. Tập dữ liệu mẫu được chia thành hai phần *train* và *test* có phân bố đồng đều giữa các nhãn và có đặc tính tương đồng với nhau (số lượng mẫu trong hai phần có thể không bằng nhau). Trong quá trình huấn luyện, mô hình học máy chỉ được học từ các mẫu trong phần *train* và không được tiếp cận các mẫu trong phần *test*. Sau khi huấn luyện đạt kết quả như ý muốn với tập *train*, mô hình được áp dụng trên tập *test* để đánh giá hiệu quả với dữ liệu mới mà mô hình phân loại chưa từng thấy.

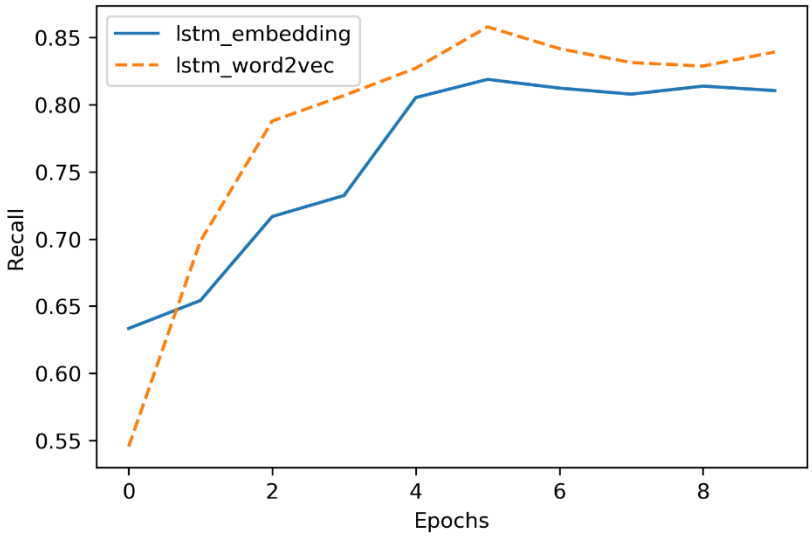
Trên thực tế, một cách làm khác thường được áp dụng nhiều hơn đó là *k-fold cross validation*. Với kỹ thuật này, tập dữ liệu mẫu được chia làm *k* phần đồng đều và mô hình được huấn luyện và thử nghiệm *k* lần rồi lấy kết quả trung bình. Ở mỗi lần, một trong *k* phần được sử dụng làm tập *test*, những phần còn lại được hợp lại dùng làm tập *train*. Phương pháp *k-fold cross validation* đem lại tính khách quan hơn so với cross validation thông thường vì toàn bộ tập mẫu lần lượt có cơ hội được sử dụng để thử nghiệm. Kết quả đo được có mức độ tự tin (confidence) cao hơn xét trên quan điểm thống kê [22].

3.3. KẾT QUẢ THÍ NGHIỆM

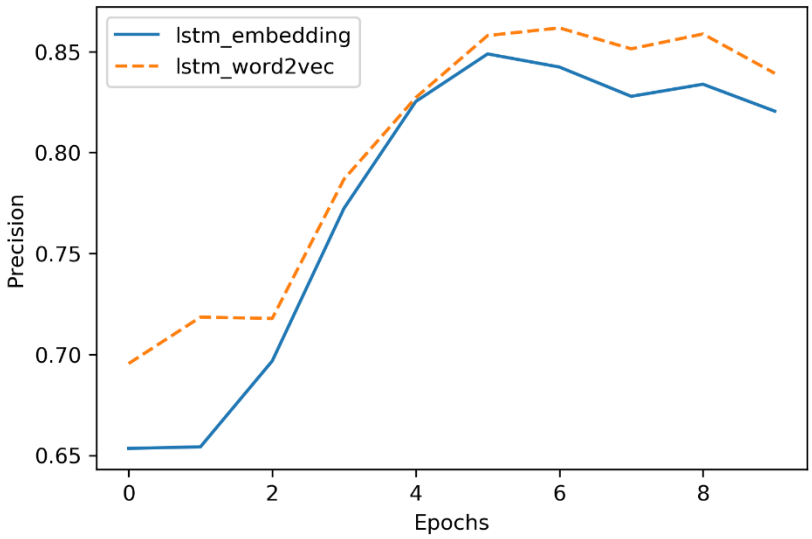
3.3.1. Thí nghiệm 1

Kết quả thí nghiệm với *k-fold cross validation* đo được trên tập thử nghiệm được trình bày trong Hình 3.1 (tiêu chí recall) và Hình 3.2 (tiêu chí precision). Với cả hai tiêu chí đánh giá, phương án sử dụng vector word2vec được huấn luyện trước đều có hiệu quả tốt hơn. Cụ thể, recall cao nhất khi áp dụng vector word2vec được huấn luyện trước với phương pháp LSTM đạt được là 0.857943 và precision cao nhất đạt được là 0.861723. Hiệu quả tốt nhất đạt được ở khoảng epoch thứ 5 và 6, sau đó hiểu quả dần giảm xuống. Độ đo precision có giá trị cao hơn so với recall ở cả 2 phương án. Đáng

chú ý, với phương án sử dụng lớp Embedding, tiêu chí recall chỉ đạt tối đa 0.818845. Độ bao phủ thấp cho thấy tỷ lệ dự đoán chính xác thấp, mô hình học được ít thông tin hữu ích hơn so với phương pháp còn lại. Chiều hướng giảm dần kết quả khi kéo dài thời gian huấn luyện có thể là biểu hiện của hiện tượng overfitting, khi mà hiệu quả trên tập *train* vẫn tăng theo quá trình huấn luyện nhưng hiệu quả trên tập *test* lại giảm. Để đạt được mô hình phù hợp, có hiệu quả tốt nhất trên thực tế, ta nên dừng huấn luyện ở khoảng 5 epochs.



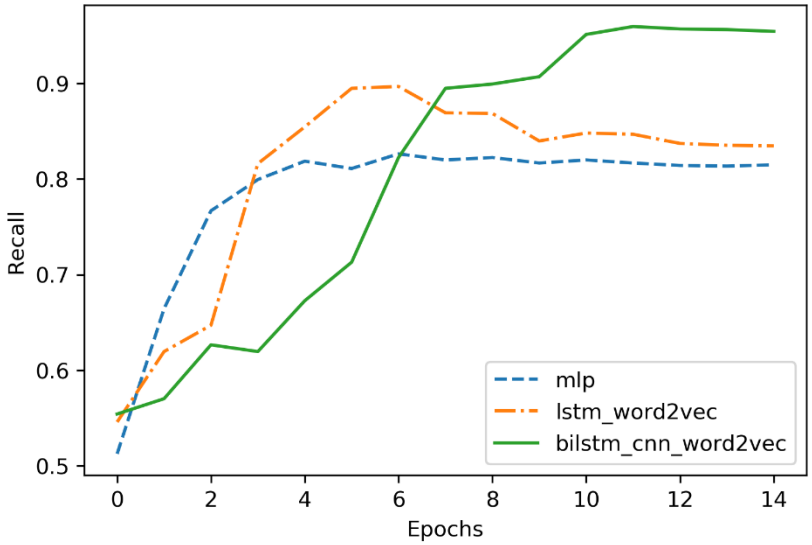
Hình 3.1: Biểu đồ độ đo Recall qua 10 epochs huấn luyện mô hình LSTM với lớp Embedding và với vector word2vec huấn luyện trước.



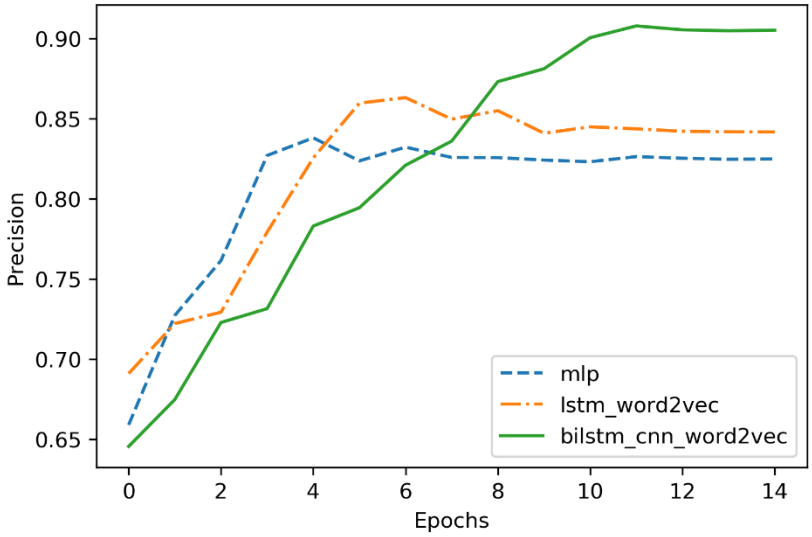
Hình 3.2: Biểu đồ độ đo Precision qua 10 epochs huấn luyện mô hình LSTM với lớp Embedding và với vector word2vec huấn luyện trước.

3.3.2. Thí nghiệm 2

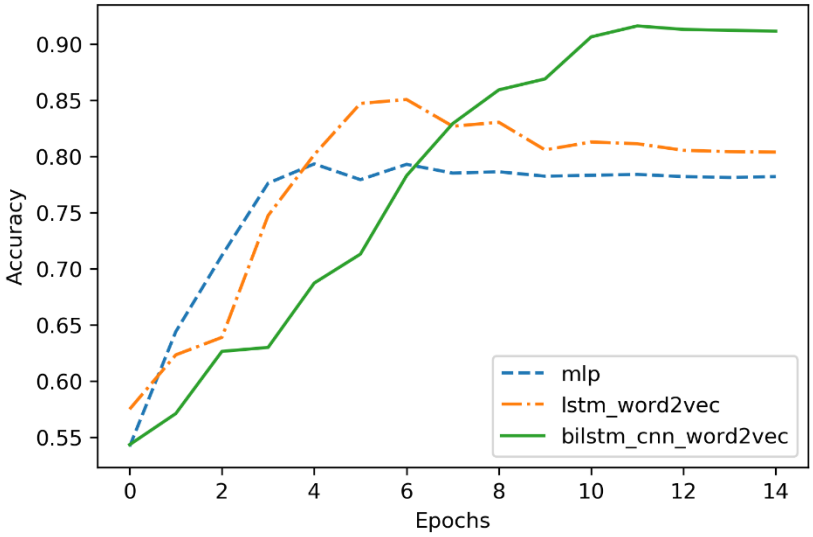
Thí nghiệm 2 so sánh ba phương pháp bằng nhiều tiêu chí khác nhau. Từ Hình 3.6, có thể thấy rõ phương pháp MLP và LSTM tìm ra giải pháp nhanh hơn, đạt được hiệu quả cao nhất chỉ sau khoảng 5 – 6 vòng huấn luyện (epochs). Phương pháp BI-LSTM-CNN thường tìm được giải pháp tốt bắt đầu từ vòng thứ 11 nhưng giải pháp tìm được tốt hơn đáng kể so với hai phương pháp còn lại. Với cả 2 phương pháp MLP và BI-LSTM-CNN, chỉ số recall thường cao hơn precision trong khi đó hai chỉ số này là tương tự nhau ở phương pháp LSTM (Hình 3.3 và Hình 3.4). Phương pháp LSTM có hiệu quả cao hơn so với MLP nhưng cũng gặp phải vấn đề overfitting nặng nề hơn. Trong khi đó, phương pháp BI-LSTM-CNN có hiệu quả cao hơn rõ rệt so với hai phương án còn lại và không có dấu hiệu rõ rệt của overfitting. Ở các chỉ số, phương pháp BI-LSTM-CNN đều đạt mức xấp xỉ 0.9 (90%) với accuracy cao nhất đạt 0.91615, điểm số F_1 đạt cao nhất 0.93304.



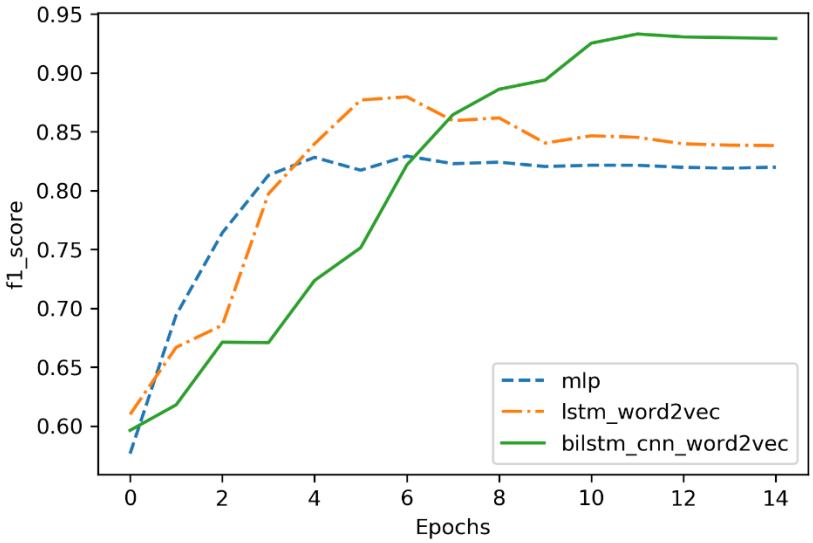
Hình 3.3: So sánh tiêu chí Recall qua 15 epochs huấn luyện giữa 3 mô hình phân loại trong Thí nghiệm 2.



Hình 3.4: So sánh tiêu chí Precision giữa 3 mô hình phân loại qua 15 epochs huấn luyện – thí nghiệm 2.



Hình 3.5: So sánh 3 mô hình phân loại bằng tiêu chí Accuracy, huấn luyện qua 15 epochs.



Hình 3.6: Kết quả thí nghiệm 2 với tiêu chí F_1 của 3 mô hình phân loại.

KẾT LUẬN

Trong luận văn này, học viên đã tiến hành nghiên cứu tài liệu về các phương pháp giải bài toán phân loại văn bản để áp dụng cho bài toán xác định tỷ lệ tin xấu trên báo điện tử tiếng Việt. Các kiến thức nền tảng về học máy và học sâu đã được trình bày theo trình tự từ cơ bản đến nâng cao. Luận văn cũng đi từ những vấn đề lý thuyết đến các ứng dụng mang tính thực nghiệm với mục tiêu xuyên suốt là để giải quyết bài toán đề ra một cách hiệu quả nhất. Thông qua quá trình tham khảo tài liệu, trong nhiều phương pháp được tóm tắt và thảo luận thì học viên đã lựa chọn và điều chỉnh ba phương pháp học sâu tiêu biểu để giải bài toán xác định tỷ lệ tin xấu: (1) phương pháp MLP đại diện cho nhóm các mạng nơ-ron truyền thẳng truyền thống, (2) phương pháp LSTM đại diện cho nhóm mạng nơ-ron hồi qui và (3) phương pháp BI-LSTM-CNN đại diện cho nhóm mô hình kết hợp nhiều cấu trúc mạng khác nhau. Qua thử nghiệm trên tập dữ liệu do học viên tự thu thập và xử lý, phương pháp BI-LSTM-CNN cho hiệu quả tốt và ổn định hơn hai phương pháp còn lại một cách đáng kể.

Tuy trọng tâm của luận văn là để giải quyết bài toán phân biệt giữa tin xấu và tin không xấu, mục tiêu ứng dụng lại là đi tìm tỷ lệ phần trăm các tin xấu trên một trang báo điện tử. Tuy rằng một mô hình học máy với độ chính xác tuyệt đối là điều bất khả thi với khoa học máy tính ở thời điểm hiện tại, bài toán này không thực sự cần một mô hình hoàn hảo. Cách mà chúng ta sử dụng kết quả của mô hình có thể khiến cho một mô hình với độ chính xác chưa cao vẫn có thể trở thành hữu ích trên thực tế. Chẳng hạn, với accuracy là 0.95, giả sử một trang báo có 100 bài viết và trên thực tế 30 trong số đó là tin xấu. Có hai trường hợp cho sai số lớn nhất đó là 5% phát hiện sai rơi cả vào 30 tin xấu (phát hiện 25 tin xấu, 75 tin tốt) hoặc 5% phát hiện sai rơi cả vào 70 tin tốt (phát hiện 35 tin xấu, 65 tin tốt). Sai số trong trường hợp này là ±16.7%. Như vậy, để khẳng định một trang báo có vượt quá tỷ lệ 30% tin xấu hay không, ta cần kết quả dự đoán ít

nhất là 35% tin xấu. Để khẳng định một trang báo chưa vượt quá tỷ lệ 30% tin xấu, ta cần kết quả dự đoán nhỏ hơn 25%.

Nghiên cứu trong luận văn này của học viên còn nhiều thiếu xót cả về mặt lý thuyết và thực hành. Nếu có thêm thời gian nghiên cứu, học viên sẽ tìm hiểu thêm về các mảng lý thuyết sâu về học sâu như các kỹ thuật học cấu trúc dữ liệu (representation learning), các lý thuyết về huấn luyện mô hình để thiết kế mô hình học sâu phù hợp nhất với đặc điểm của bài toán và dữ liệu, tránh overfitting, các cấu trúc mạng nơ-ron mang tính nền tảng như các mạng auto-encoder, mạng deep belief network, RBMs… Học viên tin rằng, khi tiếp cận được đến những lý thuyết chuyên sâu hơn, các hướng nghiên cứu mới sẽ mở ra và học viên sẽ có thể nâng cao khả năng ứng dụng học sâu để giải quyết được nhiều bài toán khó hơn trên thực tế.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[2] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems* (pp. 649-657).

[3] Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

[4] Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, *4*(5), 1-17.

[5] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2020). Deep Learning Based Text Classification: A Comprehensive Review. *arXiv preprint arXiv:2004.03705*.

[6] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436.

[7] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

[8] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[9] Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., & Xu, B. (2016). Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*.

[10] Yang, Y., & Liu, X. (1999, August). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 42-49). ACM.

[11] Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1253.

[12] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493-2537.

[13] Iyyer, M., Manjunatha, V., Boyd-Graber, J., & Daumé III, H. (2015, July). Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 7th IJCNLP - ACL (Volume 1: Long Papers)* (pp. 1681-1691).

[14] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[15] Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of documentation.

[16] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[17] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

[18] Hastie, T., Tibshirani, R., & Friedman, J. (2008). Unsupervised learning. In *The elements of statistical learning* (pp. 485-585). Springer, New York, NY.

[19] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. In *Psychological Review, 65*(6), 386.

[20] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. In *Nature, 323*(6088), 533.

[21] Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., & Recht, B. (2017). The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems* (pp. 4148-4158).

[22] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1-47.

[23] Giang, P. (2018). Bộ TT&TT giao ban quản lý nhà nước tháng 7 năm 2018. In *Cổng thông tin điện tử Bộ Thông tin và Truyền thông*. Retrieved from *http://mic.gov.vn/Pages/TinTuc/137560/Bo-TT-TT-giao-ban-quan-ly-nha-nuoc-thang-7-nam-2018.html*

[24] Ca, D. (2017). Báo và Tạp chí khác nhau thế nào? In *Nhà quản lý*. Retrieved from *http://nhaquanly.vn/bao-va-tap-chi-khac-nhau-nao-d23155.html*

[25] Authority of Broadcasting and Electronic Information (2017). Tổng hợp giấy phép Trang Thông tin điện tử tổng hợp (từ năm 2015 đến tháng 3/2017). Received from *http://abei.gov.vn/danh-sach-cap-phep/tong-hop-giay-phep-t/106467*

[26] Minh, B. (2017). Số liệu thống kê mới nhất về lĩnh vực TT&TT tính đến tháng 6/2017. In *Infonet*. Retrieved from *https://infonet.vn/so-lieu-thong-ke-moi-nhat-ve-linh-vuc-tttt-tinh-den-thang-62017-post232004.info*