DANH MỤC

MỞ ĐẦU

[1. Tính cấp thiết , tình hình nghiên cứu liên quan đến đề tài 3](#_Toc35002375)

[2. Mục tiêu , nhiệm vụ đề tài 4](#_Toc35002376)

[3. Phương pháp nghiên cứu 4](#_Toc35002376)

[4. Đối tượng và phạm vi nghiên cứu 4](#_Toc35002376)

[5. Lời cảm ơn 4](#_Toc35002376)

[CHƯƠNG 1. MÔ TẢ BÀI TOÁN 6](#_Toc35002377)

[1.1. Giới thiệu về thương mại điện tử 6](#_Toc35002378)

[1.1.1 Một số hệ thống thương mại điện tử ở Việt Nam 7](#_Toc35002378)

[1.1.2 Tầm quan trọng của hệ thống phân tích ý kiến người tiêu dùng trong thương mại điện tử 1](#_Toc35002378)

[1.2. Giới thiệu bài toán phân tích ý kiến người tiêu dùng 1](#_Toc35002380)

[1.3. Yêu cầu, Chức năng , cấu trúc hệ thống 1](#_Toc35002380)

[CHƯƠNG 2. KIẾN THỨC CƠ SỞ 1](#_Toc35002381)

[2.1 Tổng quan về mạng nơ-ron nhân tạo 1](#_Toc35002378)

[2.2. Mạng nơ-ron hồi quy(Recurrent Neural Network) 1](#_Toc35002378)

[2.3. Mạng nơ-ron Long-Short Term Memory 1](#_Toc35002378)

[2.4. Mạng nơ-ron Gated Recurrent Units 1](#_Toc35002378)

[CHƯƠNG 3. PHƯƠNG PHÁP GIẢI QUYẾT 1](#_Toc35002381)

[3.1. Thu thập dữ liệu 1](#_Toc35002382)

[3.1.1 Phương pháp thu thập 1](#_Toc35002383)

[3.1.2 Thu thập dữ liệu trên Tiki 1](#_Toc35002384)

[3.1.3 Thu thập dữ liệu trên Shopee 1](#_Toc35002385)

[3.1.4 Thu thập dữ liệu trên Lazada 1](#_Toc35002386)

[3.2. Phân tích sắc thái phản hồi bằng các mô hình học sâu 1](#_Toc35002387)

[3.2.1 Bài toán đánh giá quan điểm của người dùng 1](#_Toc35002388)

[3.2.2 Tạo tập dữ liệu để huấn luyện 1](#_Toc35002389)

[3.2.3 Mô hình LSTM 1](#_Toc35002390)

[3.2.4 Mô hình BiLSTM 1](#_Toc35002391)

[3.2.5 Mô hình GRU 1](#_Toc35002392)

[3.3. Thống kê , xây dựng biểu đồ so sánh các mối quan tâm 1](#_Toc35002393)

[CHƯƠNG 4. XÂY DỰNG PHẦN MỀM THU THẬP, PHÂN TÍCH, DỰ ĐOÁN XU HƯỚNG CỦA THỊ TRƯỜNG 1](#_Toc35002394)

[4.1. Giới thiệu phần mềm 1](#_Toc35002395)

[4.2. Các kết quả đã đạt được 1](#_Toc35002396)

[KẾT LUẬN 1](#_Toc35002397)

[1.Các kết quả đã đạt được 1](#_Toc35002398)

[2.Hạn chế của đề tài 1](#_Toc35002399)

[3.Hướng phát triển của đề tài 1](#_Toc35002400)

[TÀI LIỆU THAM KHẢO 1](#_Toc35002401)

**LỜI NÓI ĐẦU**

1. **Tính cấp thiết , tình hình nghiên cứu liên quan đến đề tài**

Mọi lĩnh vực trong xã hội đều đang chuyển mình và phát triển chóng mặt dưới đầu tàu là cuộc Cách mạng công nghiệp 4.0. Cuộc Cách mạng này cùng với sự hỗ trợ của sức mạnh số hóa và công nghệ thông tin đã tạo nên những thay đổi đột phá, trở thành chuỗi cung ứng thông minh, đem lại hiệu quả cao cho nền thương mại điện tử.

Theo dự kiến, đến năm 2020 sẽ có khoảng 30% dân số Việt Nam tham gia mua sắm online với mức khoảng 350 USD/ người. Hai mảng là thương mại điện tử di động và thương mại điện tử định vị vẫn đang là xu thế chủ đạo với tổng mức bán lẻ toàn cầu chiếm tới 25%.

Đó là chưa kể đến số lượng Smartphone mà 1 người sở hữu cũng tăng lên. Hiện nay mỗi người Việt trung bình sở hữu đến 1,3 chiếc điện thoại và 70% trong đó là Smartphone. Do có sẵn hạ tầng công nghệ, Việt Nam vẫn đạt khoảng 4 tỷ USD cho doanh mảng thương mại điện tử dù sự cách biệt giữa các địa phương vẫn khá lớn. Các doanh nghiệp đang ồ ạt nhảy vào tranh giành “miếng bánh ngon” này, có thể kể đến các thương hiệu lớn như: Tiki, Shopee, Adayroi, Sendo, …

Như vậy, có thể thấy tiềm năng của thương mại điện tử ở Việt Nam là rất lớn. Vấn đề của các doanh nghiệp là làm sao đưa được những sản phẩm tốt nhất tiếp cận được đến người dùng để có thể tìm kiếm, gia tăng khách hàng và nâng cao doanh số và doanh thu bán hàng. Còn đối với từng cá nhân khách hàng – là một thành viên trong cộng đồng mua sắm trực tuyến . Việc có thể tìm kiếm một sản phẩm rất dễ đối với họ , nhưng để đưa ra quyết định chọn lựa sản phẩm đáng tin cậy và hãng sản xuất uy tín thì lại gây khó cho những khách hàng không am hiểu về lĩnh vực đó. Thực tế , theo số liệu của nguồn Nghiên cứu Toàn cầu xu hướng mua hàng cao cấp Nielsen 2018 . Khi chọn mua các sản phẩm cao cấp mới , người tiêu dùng đánh giá sự giới thiệu của bạn bè, người thân là yếu tố quan trọng nhất , với 50% người được khảo sát nhận đình rằng các đề xuất của bạn bè và gia đình ảnh hưởng đến quyết định dùng thử của họ . Từ đó , ta nghĩ đến một ý tưởng rằng có một số lượng lớn đề xuất khác gấp ngàn lần các đề xuất từ người thân khách hàng đem lại độ tin tưởng cao hơn hay không? Và Đó chính là việc lắng nghe ý kiến của cộng đồng người tiêu dùng trên khắp mọi miền tổ quốc.

Nhận thấy đây là một đề tài không chỉ mang tính khoa học lớn mà còn có tính thực tế rất cao , là nhu cầu cấp thiết của toàn xã hội ngày nay , tôi quyết định chọn đề tài

**“** *Nghiên cứu xây dựng hệ thống phân tích ý kiến phản hồi của khách hàng trên mạng* **“** để thực hiện nghiên cứu.

Đề tài tập trung vào việc nghiên cứu , khảo sát , đánh giá và đề xuất ra một số phương pháp hỗ trợ tư vấn khách hàng mua sắm trực tuyến , giúp nhà sản xuất đưa ra quyết định nên áp dụng mẫu sản phẩm mới hay không.

1. **Mục tiêu , nhiệm vụ đề tài**

*Mục tiêu*

Đề tài tập trung vào việc nghiên cứu đưa ra các sắc thái của ý kiến phản hồi, thống kê mối quan tâm của khách hàng sau khi mua các sản phẩm trên các trang mua sắm trực tuyến. Tỉ lệ tích cực và tiêu cực sẽ giúp cho nhà sản xuất quyết định tới việc thay đổi chiến lược của mình.

Thống kê số lượng mối quan tâm , đánh giá của cộng đồng người tiêu dùng sẽ giúp cho khách hàng mới nhận biết được xu hướng mua sắm và đâu là sản phẩm tốt được tin dùng.

*Nhiệm vụ*

Thứ nhất : Thu thập ý kiến người tiêu dùng trên mạng

Thứ hai : Phân tích phản hồi tích cực , tiêu cực của khách hàng với từng sản phẩm.

Thứ ba : Xây dựng hệ thống xác định xu hướng , nhu cầu mua sắm của khách hàng.

1. **Phương pháp nghiên cứu**

Căn cứ và mục đích , yêu cầu , nội dung đặt ra của đề tài , tôi xác định phương pháp nghiên cứu ở đây bao gồm:

* Xây dựng một sơ đồ hệ thống tổng quan mang lại cái nhìn chung nhất.
* Tiến hành thu thập thông tin
* Nghiên cứu dữ liệu , áp dụng các thuật toán , mô hình để xử lý dữ liệu , và đưa ra kết quả tới người dùng
* Xây dựng hệ thống , thử nghiệm và cải thiện cái thuật toán , mô hình đã áp dụng

1. **Đối tượng và phạm vi nghiên cứu**

*Đối tượng nghiên cứu:*

\_ Các phương pháp thu thập , tổ chức dữ liệu.

\_ Các phương pháp xử lý ngôn ngữ tự nhiên.

\_ Các phương pháp hỗ trợ , tư vấn sản phẩm.

*Phạm vi nghiên cứu:*

\_ Đề tài tập trung nghiên cứu phân tích sắc thái ý kiến phản hồi của người tiêu dùng trên các kênh thương mại điện tử và đề xuất , định hướng mua bán sản phẩm.

1. **Lời cảm ơn**

Để có thể hoàn thành đề tài nghiên cứu khoa học một cách hoàn chỉnh , bên cạnh sự nỗ lực cố gắng của nhóm nghiên cứu còn có sự giúp đỡ của quý thầy cô , sự ủng hộ động viên của đơn vị , gia đình và bạn bè trong suốt thời gian học tập và nghiên cứu , thực hiện đề tài nghiên cứu.

Xin chân thành bày tỏ lòng biết ơn đến thầy TS. Phan Việt Anh , người đã luôn động viên , giúp đỡ , hướng dẫn chỉ dạy và tạo mọi điều kiện thuận lợi nhất cho nhóm chúng em hoàn thành đề tài này. Chân thành bày tỏ lòng biết ơn đến toàn bộ thầy cô trong khoa Công nghệ thông tin học viện Kỹ Thuật Quân sự và các thủ trưởng các cấp đã tận tình chỉ bảo , truyền đạt kiến thức quý báu cũng như tạo điều kiện , quan tâm động viên cho nhóm nghiên cứu trong quá trình học tập nghiên cứu và cho đến khi hoàn thiện đề tài.

**Chương 1**

**MÔ TẢ BÀI TOÁN**

* 1. **Giới thiệu về thương mại điện tử**

Thương mại điện tử (E-Commerce) là hình thức kinh doanh trực tuyến sử dụng nền tảng công nghệ thông tin với sự hỗ trợ của Internet để thực hiện các giao dịch mua bán, trao đổi, thanh toán trực tuyến. Thương mại điện tử (TMĐT) là xu hướng của thời đại toàn cầu hóa, đây là lĩnh vực tiềm năng để các doanh nghiệp vừa và nhỏ sinh lợi và phát triển, cơ hội cho những ai muốn khởi nghiệp kinh doanh theo mô hình mới.

Việt Nam được đánh giá là một trong những thị trường tiềm năng của TMĐT do dân số trẻ, lượng người dùng smartphone chiếm tỷ trọng lớn. Tại Việt Nam, doanh thu từ TMĐT năm 2013 đạt chỉ đạt 2,2 tỷ USD, đến năm 2018 con số này đã tăng lên 8,06 tỷ USD. Số lượng người mua sắm trực tuyến cũng tăng từ 30,3 triệu người (năm 2015) lên 39,9 triệu người (năm 2018).

***1.1.1 Một số hệ thống thương mại điện tử ở Việt Nam***

**Tiki**

Tiki là doanh nghiệp kinh doanh trong ngành thương mại điện tử được thành lập vào năm 2010, với khởi điểm ban đầu chủ yếu kinh doanh sách trực tuyến, sau 7 năm, Tiki.vn hiện đã trở thành một nhà bán lẻ trực tuyến đa ngành với 9 ngành hàng tiêu dùng cơ bản, trong đó sách vẫn là một sản phẩm “nhận diện” thương hiệu doanh nghiệp.

Có thể nói Tiki đang là đối thủ lớn của Lazada với tốc độ phát triển chóng mặt. Website Tiki.vn có hơn 300.000 sản phẩm thuộc 12 lĩnh vực điện tử, phong cách sống và sách.

Tiki đạt được tỷ lệ hài lòng cao nhất của khách hàng và tỷ lệ hoàn vốn thấp nhất trong số tất cả các nhà khai thác thương mại điện tử ở Việt Nam do khả năng kiểm soát chuỗi cung cấp hết sức hợp lý từ các nhà chuyên môn, từ việc hợp tác với các nhãn hiệu nổi tiếng chuyên cung cấp hàng hóa chất lượng cao đến việc quản lý kho và hậu cần, thanh toán và cung cấp dịch vụ khách hàng cao cấp.

Thành công của Tiki đến từ việc mô hình kinh doanh linh hoạt, và tương tác rất tốt với khách hàng, và điều này không thể không nhắc đến hệ thống thương mại điện tử trên 2 nền tảng web và ứng dụng di động được xây dựng với các cách thức hỗ trợ, tư vấn người dùng, quảng cáo sản phẩm đã đưa sản phẩm đến gần hơn với người dùng.

**Lazada**

Tháng 2 năm 2012, Lazada.vn bắt đầu tham gia thị trường mua sắm trực tuyến của Việt Nam. Mặc dù chỉ xuất hiện chưa đầy 2 tháng nhưng Lazada.vn đã đứng đầu về các thị phần các website bán hàng trực tuyến theo thống kê năm 2014.

Lazada sử dụng mô hình B2C để triển khai các hoạt động kinh doanh thương mại điện tử của mình cùng một loạt chiến dịch tiếp thị nhằm tăng mức độ thân thiện, gần gũi với khách hàng. Nhờ các chiến lược này, Lazada đã đạt doanh thu 600 tỷ đồng trên tổng số doanh thu hơn 1600 tỷ đồng của các sàn giao dịch năm 2014.

Dù với bất cứ chiến lược nào, điều quan trọng nhất của Lazada vẫn nằm ở vấn đề bán hàng trực tuyến, và trải nghiệm của người mua trên hệ thống của công ty này vẫn là quan trọng nhất. Lazada cũng có một nền tảng web và ứng dụng di động rất tốt trong việc gợi ý các sản phẩm hoặc đáp ứng nhu cầu tìm kiếm cho người mua.

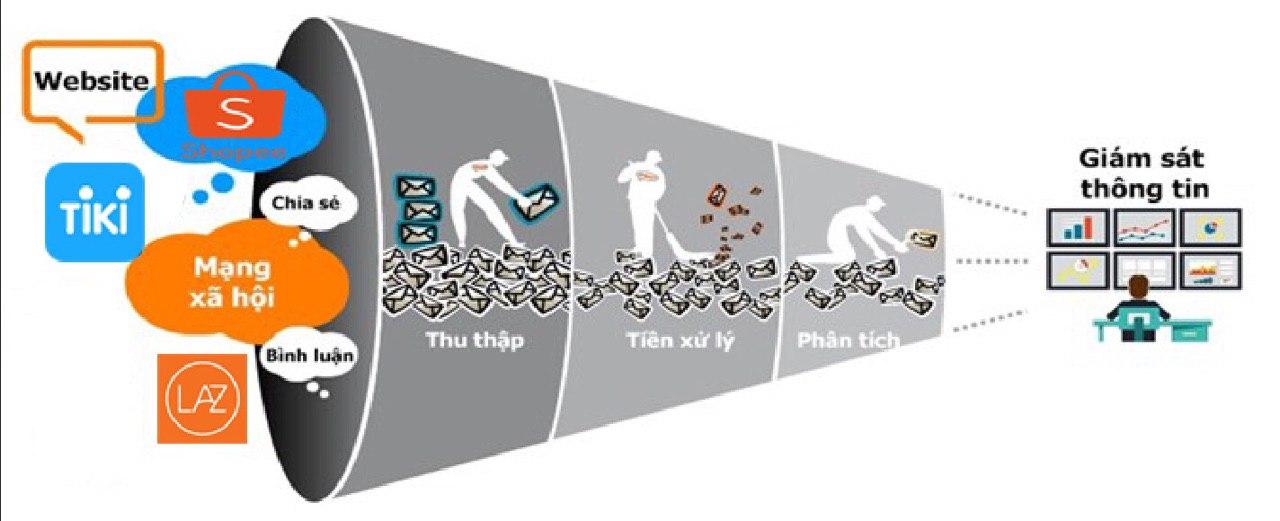
***1.1.2*** ***Tầm quan trọng của hệ thống phân tích ý kiến người tiêu dùng trong thương mại điện tử***

Ngoài các chiến lược kinh doanh , marketing và các cách thức quản lý , tổ chức doanh nghiệp , về bản chất các doanh nghiệp thương mại điện tử vẫn tồn tại và phát triển xoay quanh hệ thống thương mại điện tử trực tuyến của họ. Vì vậy , sở hữu một hệ thống có khả năng phân tích , đánh giá , tư vấn và hỗ trợ người dùng tốt khi người dùng mua sắm vẫn là một yếu tốt quan trọng đóng góp vào thành công của một doanh nghiệp thương mại điện tử.

* 1. **Giới thiệu bài toán phân tích ý kiến người tiêu dùng**

Bài toán phân tích ý kiến người tiêu dùng đã được nghiên cứu và phát triển bởi nhiều đội ngũ khoa học và các doanh nghiệp. Vấn đề quan tâm lớn nhất của người dùng khi mua hàng là họ cần phải đảm bảo rằng hàng hóa là hàng thật , chất lượng. Niềm tin là thứ cần được xây dựng và xem xét nghiêm túc dưới con mắt của các doanh nghiệp phát triển thương mại điện tử.

Vì vấn đề niềm tin mà thực tế hiện nay nhiều đại lý bán lẻ , cá nhân bán hàng trên các kênh thương mại điện tử lợi dụng việc bình luận đánh giá về sản phẩm để tạo uy tín cho cửa hàng của mình. Nhưng liệu bao nhiêu trong số bình luận đó là thật ? bao nhiêu là do cửa hàng đó tự tạo ra để tâng bốc thương hiệu của mình. Không may , chúng ta có thể chắc chắn được rằng , không ai lại tự đi chê sản phẩm của mình không tốt , không đúng chất lượng cả. Dựa vào yếu tố này , chúng ta có thể phân tích số lượng phản hồi tiêu cực từ khách hàng và từ đó có thể đánh giá chung cho chất lượng cửa hàng. Cuối cùng, đưa ra lời khuyên cho người tiêu dùng.



Hình 1.1 : *Hệ thống phân tích ý kiến phản hồi của khách hàng*

Như chúng ta có thể thấy trong hình 1.1 , Các câu bình luận của các trang thương mại điện tử khác nhau như Tiki , Lazada , Shopee được thu thập và tổ chức thành CSDL , Dữ liệu sẽ được áp dụng một số thuật toán để tạo. Hệ thống sẽ xử lý đầu vào này bằng các thuật toán xử lý ngôn ngữ tự nhiên , và các mô hình Deep learning . Ngoài ra , người dùng sẽ được hỗ trợ tư vấn xem liệu đâu là sản phẩm được ưu chuộng và tiêu dùng thịnh hành hiện nay , giá cả phù hợp và phải chăng.

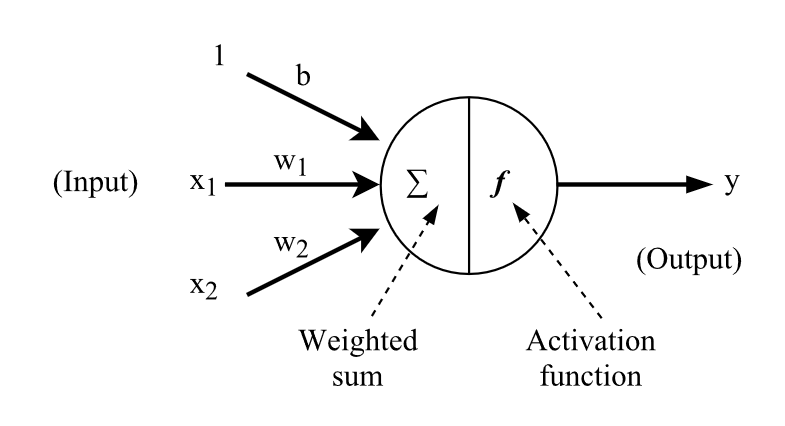
**Chương 2**

**KIẾN THỨC CƠ SỞ**

**2.1 Tổng quan về mạng nơ-ron nhân tạo**

Mạng nơ-ron nhân tạo là một mô hình tính toán được xây dựng nên từ tập các phần tử liên kết với nhau được gọi là các nơ-ron. Ta sẽ bắt đầu bằng việc mô tả cấu trúc của một nơ-ron cơ bản nhất được sử dụng để xây dựng nên mạng nơ-ron. Cấu trúc cơ bản của một nơ-ron bao gồm *đầu vào, hàm kích hoạt* và *đầu ra. Đầu vào* của nơ-ron có thể đến từ các nơ-ron khác hoặc đến từ nguồn bên ngoài, mỗi *đầu vào* này được gắn liền với một trọng số thể hiện sự quan trọng của chúng với các *đầu vào* khác. Để tính toán *đầu ra,* nơ-ron áp dụng *hàm kích hoạt* với tổng các trọng số của *đầu vào. Đầu vào* x với n chiều, *đầu ra* sẽ được tính toán bởi hàm số sau:

Trong hình 4.27 nơ-ron nhận hai giá trị *đầu vào* là x1 và x2, cùng với trọng số lần lượt là w1 và w2. Cùng với đó, còn có *đầu vào* khác là 1 với trọng số b được gọi là *bias.* Vai trò của giá trị *bias* này giúp dịch chuyển hàm số sang trái hoặc phải.



Hình 2.1 Kiến trúc một nơ-ron cơ bản

*Hàm kích hoạt f*  được sử dụng để biến *đầu vào* thành *đầu ra* phi tuyến tính. Có một vài *hàm kích hoạt* được sử dụng phổ biến trong thực tế. Các *hàm kích hoạt* cần có các tính chất như sau:

*Phi tuyến tính.* Đây là tính chất rất quan trọng cho phép mạng nơ-ron học được các hàm số phức tạp. Thực tế, mạng nơ-ron 2 lớp với *hàm kích hoạt* phi tuyến được chứng minh có thể xấp xỉ bất kỳ hàm số nào. (Universal Approximator).

*Khả vi liên tục*. Thuật toán phổ biến nhất để huấn luyện mạng nơ-ron là thuật toán Gradient Descent. Thuật toán Gradient Descent tìm điểm cực tiểu của hàm số bằng cách đi ngược lại với dấu của đạo hàm của hàm số tại điểm đó. Vì vậy tính chất *khả vi liên tục* là rất cần thiết để áp dụng các phương pháp tối ưu dựa trên tính toán đạo hàm. Một số *hàm kích hoạt* không khả vi tại một số điểm nhất định. Trong trường hợp này, các phương pháp tối ưu dựa trên tính toán đạo hàm sẽ không thể thực hiện được trên các điểm này.

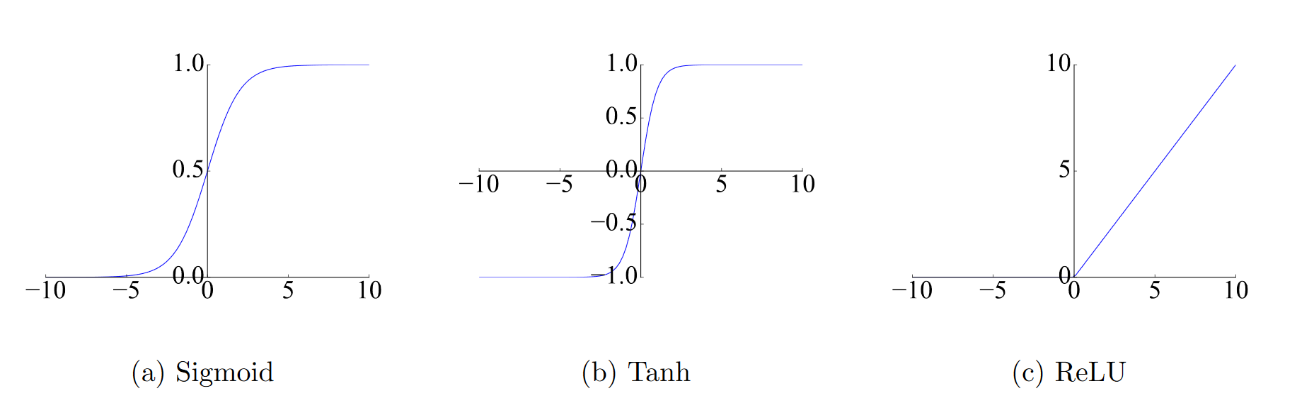
*Đơn điệu.* Tính chất này đảm bảo rằng mặt lỗi gắn với mô hình đơn lớp là mặt lồi.

Một số *hàm kích hoạt* phổ biến bao gồm: *sigmoid, tanh* và *ReLU.* Biểu thức của chúng được trình bày dưới đây:

*sigmoid:* đưa giá trị của các *đầu vào* về khoảng [0,1]

*tanh:* đưa giá trị các *đầu vào* về khoảng [-1,1]

*ReLU:* loại bỏ các giá trị *đầu vào* âm và đưa chúng về 0



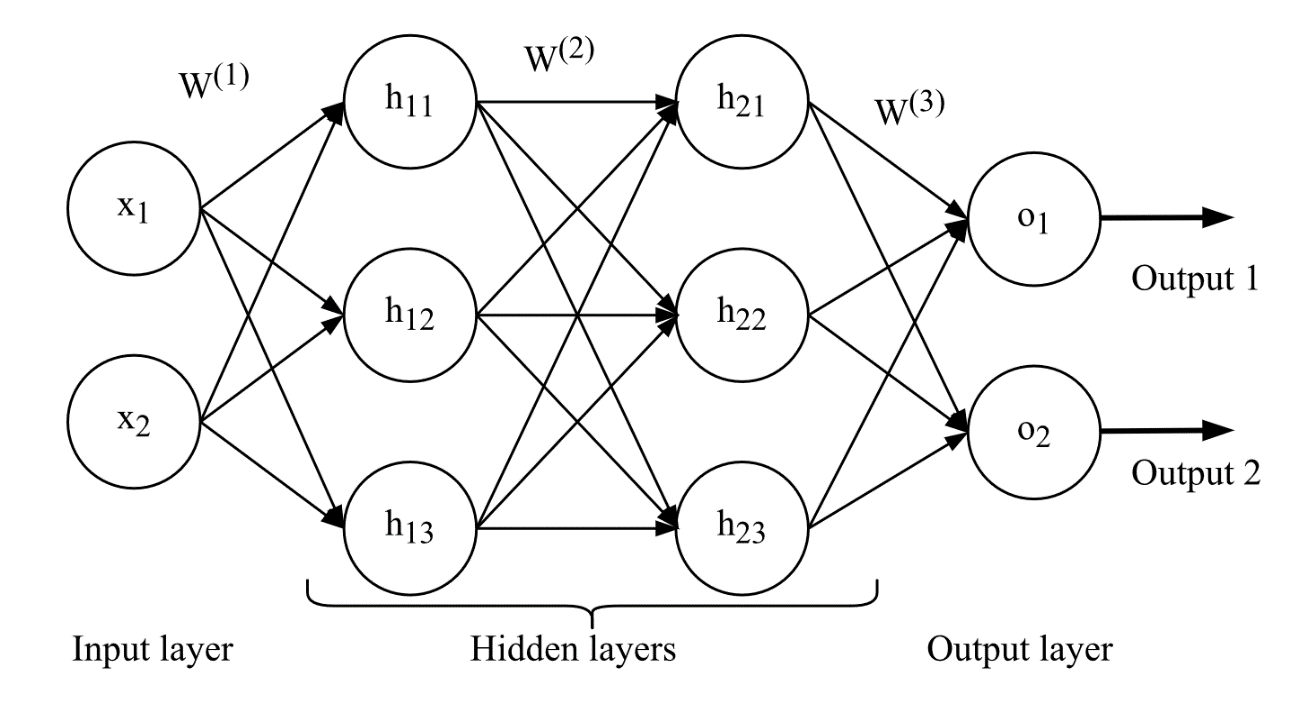
Hình 2.2 Đồ thị các hàm kích hoạt sigmoid, tanh, ReLU

Mạng nơ-ron là mạng bao gồm các lớp nơ-ron, mỗi lớp nơ-ron bao gồm các nơ-ron được sắp xếp, trong đó *đầu ra* của một nơ-ron có thể là *đầu vào* của một nơ-ron khác. Các nơ-ron giữa 2 lớp kết nối với nhau và mỗi kết nối này được gắn liền với một giá trị trọng số. Mạng nơ-ron nhận *đầu vào* từ bên ngoài tại lớp đầu tiên, thực hiện các tính toán, và đưa *đầu ra* đến lớp cuối cùng của mạng. Giá trị kích hoạt *a* của nơ-ron *i* là *hàm kích hoạt* được áp dụng với giá trị tổng của các tích giữa *đầu vào* và trọng số, cộng với giá trị *bias: ai = f(Wi.x + bi).* Giả sử các lớp nơ-ron là loại lớp nơ-ron kết nối toàn bộ (fully-connected layer), ta có thể suy ra công thức tính toán giá trị *kích hoạt* của các nơ-ron trong cùng một lớp theo công thức tính toán ma trận sau:

Với x và  *f* được áp dụng cho từng thành phần trong z.

*f* (z) = *f* ([z1, z2, …, zm]) = [*f* (z1), *f* (z2), …, *f* (zm)] (2.7)

Hình sau mô tả cấu trúc của một mạng nơ-ron suy luận tiến (feedforward).



Hình 2.3 Kiến trúc của một mạng nơ-ron

Trong mạng nơ-ron suy luận tiến, dữ liệu được tính toán và truyền theo một chiều từ lớp đầu vào (input layer), qua các lớp ẩn (hidden layer), và đến các lớp đầu ra (output layer). Lớp đầu vào chịu trách nhiệm nhận dữ liệu đầu vào từ bên ngoài và chuyển chúng đến lớp ẩn mà không thực hiện bất kỳ một tính toán nào. Lớp ẩn thực hiện các tính toán, áp dụng các *hàm kích hoạt*, và chuyển dữ liệu lần lượt qua các lớp ẩn tiếp theo và đến lớp đầu ra. Quá trình tính toán của mạng nơ-ron trong hình trên có thể được tính dựa trên công thức sau:

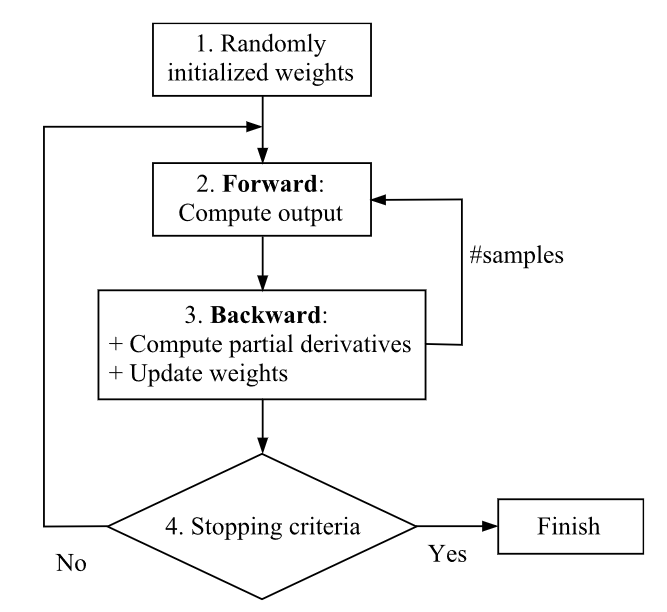
*Đầu ra* là kết quả của quá trình tính toán *đầu vào* qua từng lớp trong mạng nơ-ron. Quan sát các giá trị *đầu ra* này, chúng ta sẽ đánh giá lỗi của mô hình theo một số phương pháp nhất định. Để thu được mô hình tối ưu, ta cần một lượng lớn mẫu dữ liệu, các tham số của mô hình sẽ được điều chỉnh trong quá trình huấn luyện với mục tiêu giảm lỗi mô hình trên tập dữ liệu đó.

**Huấn luyện mạng nơ-ron**

Mục tiêu của huấn luyện một mạng nơ-ron là tìm ra bộ các tham số và *bias* trong đó với mỗi véc-tơ *đầu vào* mô hình sẽ tính toán và đưa ra véc-tơ *đầu ra* mong muốn. Để làm được điều đó, chúng ta cần phải định nghĩa một *hàm lỗi* (Error function, đôi khi có thể được gọi là cost function hay loss function) để tính toán mỗi véc-tơ *đầu ra* thành một giá trị số thực. Trong quá trình huấn luyện, người huấn luyện sẽ cố gắng để giảm thiểu giá trị lỗi để thu được mô hình tối ưu. Xét trường hợp của học có giám sát, mỗi mẫu dữ liệu được gắn với một nhãn. Giá trị lỗi của mỗi mẫu dữ liệu có thể được tính bằng độ chênh lệch giữa *đầu ra* và nhãn thực tế của mẫu đó. Có nhiều loại *hàm lỗi* được sử dụng để đánh giá lỗi mô hình. Lựa chọn *hàm lỗi* hợp lý cũng là một vấn đề quan trọng trong việc huấn luyện mô hình. Một *hàm lỗi* khá phổ biến là hàm Mean Square Error (MSE) tính toán khoảng cách giữa *đầu ra* và nhãn thực tế bằng khoảng cách Euclidean.Công thức MSE cho n mẫu dữ liệu được tính toán như sau:

Trong đó là véc-tơ *đầu ra* dự đoán bởi mạng, là nhãn thực tế.

Một thuật toán rất hiệu quả để tối ưu mạng nơ-ron là thuật toán **Lan truyền ngược (Backpropation)** kết hợp với **Gradient Descent.** Phương pháp thực hiện lặp lại để điều chỉnh các tham số mô hình để *hàm lỗi* hội tụ đến giá trị cực tiểu (minimum). Với mỗi bước, ta xác định đạo hàm của hàm số tại điểm hiện tại theo công thức:

 Hình 2.4 Sơ đồ giải thuật lan truyền ngược

Sau đó, trọng số sẽ được điều chỉnh thêm một lượng ngược với dấu của đạo hàm hàm lỗi tại điểm đó.

Với *l* là số trọng số, là tốc độ học (learning rate) để điều chỉnh tốc độ hội tụ đến giá trị cực tiểu của *hàm lỗi.*

Do việc tính toán từ *đầu vào* đến *đầu ra* phải trải qua nhiều bước tính toán, và qua nhiều hàm số, nên việc tính đạo hàm theo trọng số không thể thực hiện trực tiếp. Trong trường hợp này, giải thuật **lan truyền ngược** được áp dụng theo chiều ngược lại với chiều suy luận tiến để tính toán đạo hàm theo các trọng số.

Để tính toán đạo hàm theo trọng số, phương pháp này dựa theo quy tắc **chuỗi đạo hàm của hàm hợp**.

Giải thuật **lan truyền ngược** được thực hiện như sau:

Giả sử ta có mạng nơ-ron có Llớp.

**Bước 1**: Suy luận tiến

Lần lượt tính các từ = 2 L theo công thức:

Trong đó, lớp chính là giá trị *đầu vào.*

**Bước 2:** Tính đạo hàm theo z ở *lớp đầu ra*

Trong đó, đã tính được ở bước 1.

**Bước 3:** Lan truyền ngược

Tính đạo hàm theo z ngược lại từ = (L – 1) 2 theo công thức:

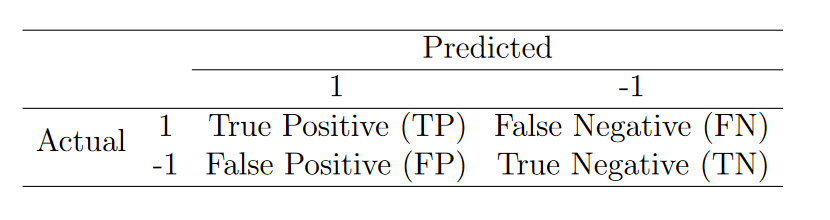
Với tính được ở bước 1 và tính được ở vòng lặp trước.

**Bước 4:** Tính đạo hàm theo bằng công thức:

Với tính được ở bước 1 và tính được ở bước 3. [4]

**Phương pháp đánh giá mô hình học sâu**

Để so sánh hiệu quả của các mô hình máy học, có nhiều phương pháp đánh giá được áp dụng. Nhiều phương pháp đánh giá thông thường được tính toán dựa trên confusion matrix. Confusion matrix là một bảng, trong đó mỗi dòng là giá trị nhãn dự đoán và mỗi cột là giá trị các nhãn thực tế. Hình sau là một confusion matrix cho bài toán phân loại 2 lớp (Binary Classification) trong đó dữ liệu được phân vào 2 loại.



Hình 2.5 Confusion Matrix

Từ confusion matrix chúng ta có thể thấy có 4 khả năng xảy ra với *đầu ra* của bài toán này. Nếu cả nhãn dự đoán và nhãn thực tế đều là positive, ta gọi trường hợp này là True Positive, nếu nhãn dự đoán là positive và nhãn thực tế là negative, ta có trường hợp False Positive, nếu nhãn dự đoán là negative và nhãn thực tế là positive, ta gọi là trường hợp False Negative, trường hợp còn lại với nhãn dự đoán là negative và nhãn thực tế là negative gọi là True Negative.

Tỷ lệ **True Positive** (TPR): hay còn gọi là **recall**, được tính theo công thức sau:

Tỷ lệ **True Negative** (TNR):

Tỷ lệ **False Positiv**e (FPR): hay còn gọi là tỷ lệ báo động nhầm, có công thức:

**Precision**: Tỷ lệ các mẫu được gán nhãn đúng trên các mẫu được gán 1 nhãn nhất định.

**Accuracy**: Còn gọi là **độ chính xác,** là tỷ lệ các nhãn được dự đoán đúng tất cả nhãn được dự đoán.

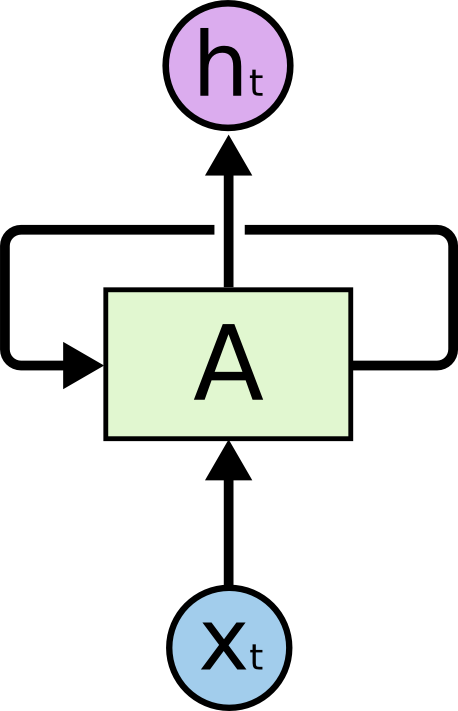
**F1 score**: Trung bình điều hòa của precision và recall. Đại lượng này được tính từ precision và recall.

### **2.2 Mạng nơ-ron hồi quy (Recurrent Neural Network)**

Con người không bắt đầu suy nghĩ của họ từ đầu tại tất cả các thời điểm. Cũng như khi đang đọc bài viết này, bộ não con người hiểu mỗi chữ ở đây dựa vào từ mà bộ não đã hiểu các chữ trước đó chứ không phải là với mỗi từ mới chúng ta sẽ có một nhận thức hoàn toàn mới mà bỏ đi những thông tin về những từ cũ. Tức là tư duy đã có một bộ nhớ để lưu lại những gì diễn ra trước đó.

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

Mạng nơ-ron hồi quy (RNN) sinh ra để giải quyết vấn đề đó. Mạng này chứa các vòng lặp bên trong cho phép thông tin có thể lưu lại được.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/RNN-rolled.png)

Hình 2.6 Một nút mạng RNN cơ bản

Hình vẽ trên mô tả một đoạn của mạng nơ-ron hồi quy A với đầu vào ​và đầu ra là . Một vòng lặp cho phép thông tin có thể được truyền từ bước này qua bước này qua bước khác của mạng nơ-ron.

Các vòng lặp này khiến cho mạng nơ-ron hồi quy trông có vẻ khó hiểu. Tuy nhiên một mạng nơ-ron hồi quy có thể được coi là nhiều bản sao chép của cùng một mạng, trong đó mỗi đầu ra của mạng này là đầu vào của một mạng sao chép khác. Chúng ta hãy xem mô hình sau:

Chuỗi lặp lại các mạng này chính là phân giải của mạng nơ-ron hồi quy, các vòng lặp khiến chúng tạo thành một chuỗi danh sách các mạng sao chép nhau. Trong vài năm gần đây, việc ứng dụng RNN đã đưa ra được nhiều kết quả rất tốt trong nhiều lĩnh vực: nhận dạng giọng nói, mô hình hóa ngôn ngữ, dịch máy, mô tả ảnh,...

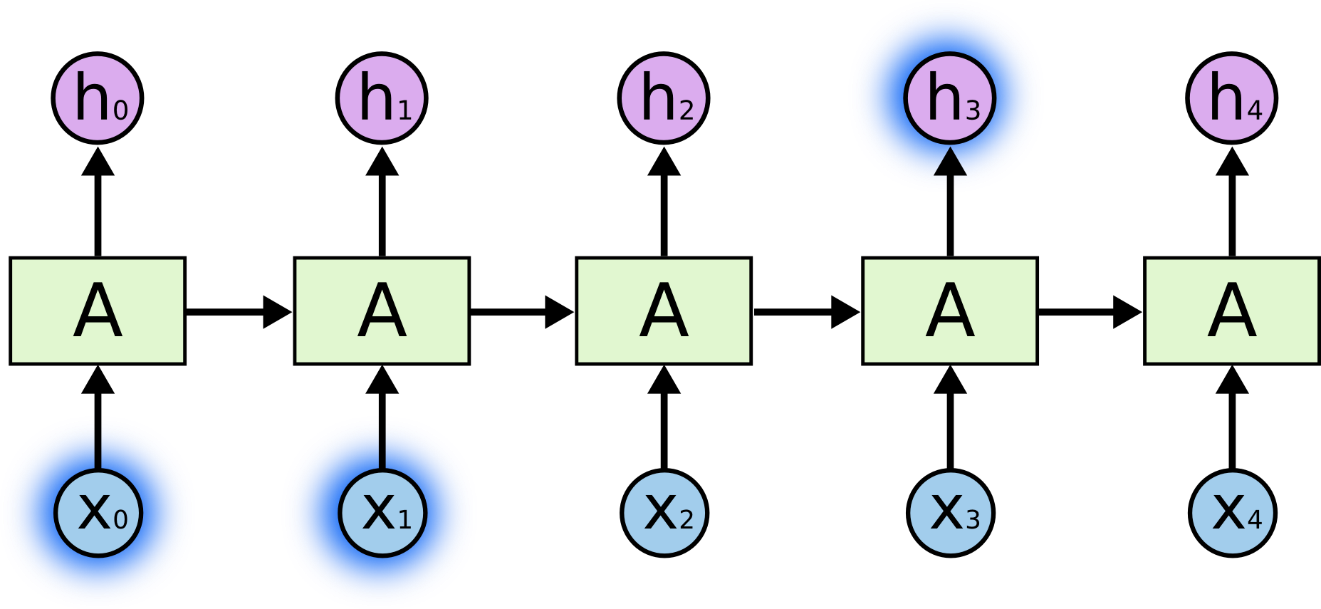
Đằng sau sự thành công này chính là sự đóng góp của mạng bộ nhớ dài-ngắn ([LSTM](https://en.wikipedia.org/wiki/Long_short-term_memory)). LSTM là một dạng đặc biệt của mạng nơ-ron hồi quy, với nhiều bài toán thì nó tốt hơn mạng hồi quy thuần. Hầu hết các kết quả thú vị thu được từ mạng RNN là được sử dụng với LSTM.[5]

**Vấn đề của mạng nơ-ron hồi quy**

Nhược điểm của mạng nơ-ron hồi quy

Một điểm nổi bật của RNN chính là ý tưởng kết nối các thông tin phía trước để dự đoán cho hiện tại. Việc này tương tự như ta sử dụng các cảnh trước của bộ phim để hiểu được cảnh hiện thời. Nếu mà RNN có thể làm được việc đó thì chúng sẽ cực kì hữu dụng, tuy nhiên liệu chúng có thể làm được không?

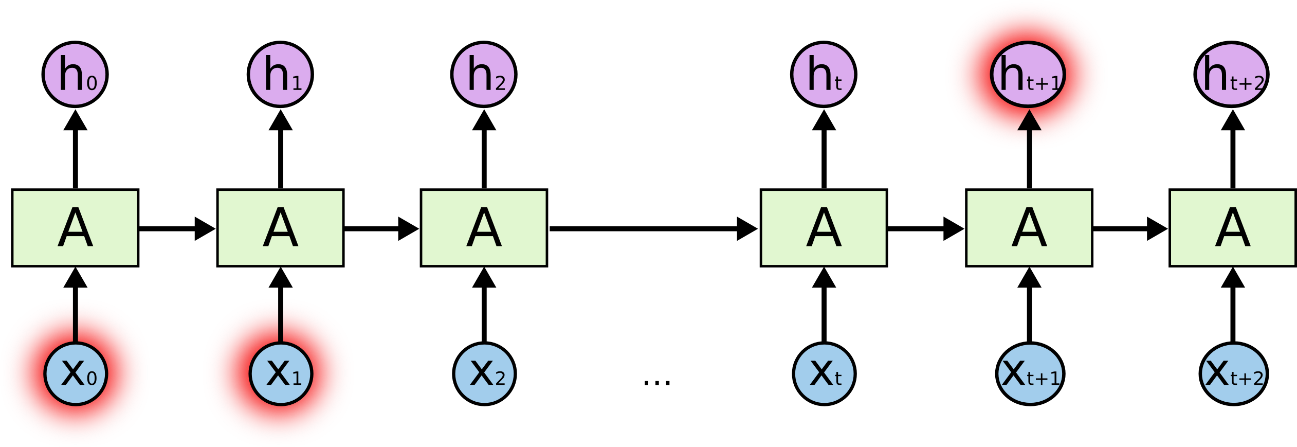
Đôi lúc ta chỉ cần xem lại thông tin vừa có thôi là đủ để biết được tình huống hiện tại. Ví dụ, ta có câu: “các đám mây trên bầu trời” thì ta chỉ cần đọc tới “các đám mây trên bầu” là đủ biết được chữ tiếp theo là “trời”. Trong tình huống này, khoảng cách tới thông tin có được cần để dự đoán là nhỏ, nên RNN hoàn toàn có thể học được.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/RNN-shorttermdepdencies.png)

Hình 2.7 Ví dụ về mạng RNN

Nhưng trong nhiều tình huống ta buộc phải sử dụng nhiều ngữ cảnh hơn để suy luận. Ví dụ, dự đoán chữ cuối cùng trong đoạn: “I grew up in France… I speak fluent French.”. Rõ ràng là các thông tin gần (”I speak fluent”) chỉ có phép ta biết được đằng sau nó sẽ là tên của một ngôn ngữ nào đó, còn không thể nào biết được đó là tiếng gì. Để biết là từ gì, thì ta cần phải có thêm ngữ cảnh “I grew up in France” nữa mới có thể suy luận được. Rõ ràng là khoảng cách thông tin lúc này có thể đã là rất xa so với khả năng học của RNN

Khoảng cách càng lớn dần thì RNN bắt đầu không thể nhớ và học được nữa.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/RNN-longtermdependencies.png)

Hình 2.8 Vấn đề phụ thuộc xa

Về mặt lý thuyết, rõ ràng là RNN có khả năng xử lý các phụ thuộc xa (long-term dependencies). Chúng ta có thể xem xét và cài đặt các tham số sao cho khéo là có thể giải quyết được vấn đề này. Tuy nhiên, đáng tiếc trong thực tế RNN có vẻ không thể học được các tham số đó. Vấn đề này đã được khám phá khá sâu bởi [Hochreiter](http://people.idsia.ch/~juergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf) và Bengio, trong các bài báo của mình, họ đã tìm được những lý do căn bản để giải thích tại sao RNN không thể học được. Vấn đề này được gọi là **phụ thuộc xa.**

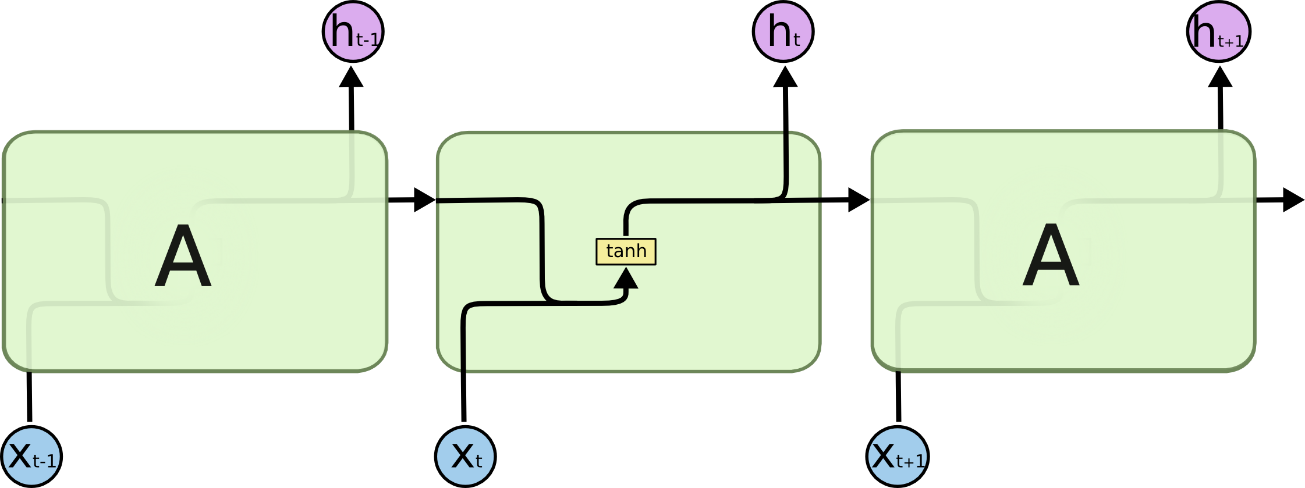
Tuy nhiên, LSTM không vấp phải vấn đề đó!

### **2.3 Mạng nơ-ron Long-Short Term Memory**

Mạng nơ-ron Long-Short Term Memory (LSTM) là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi [Hochreiter & Schmidhuber (1997)](http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

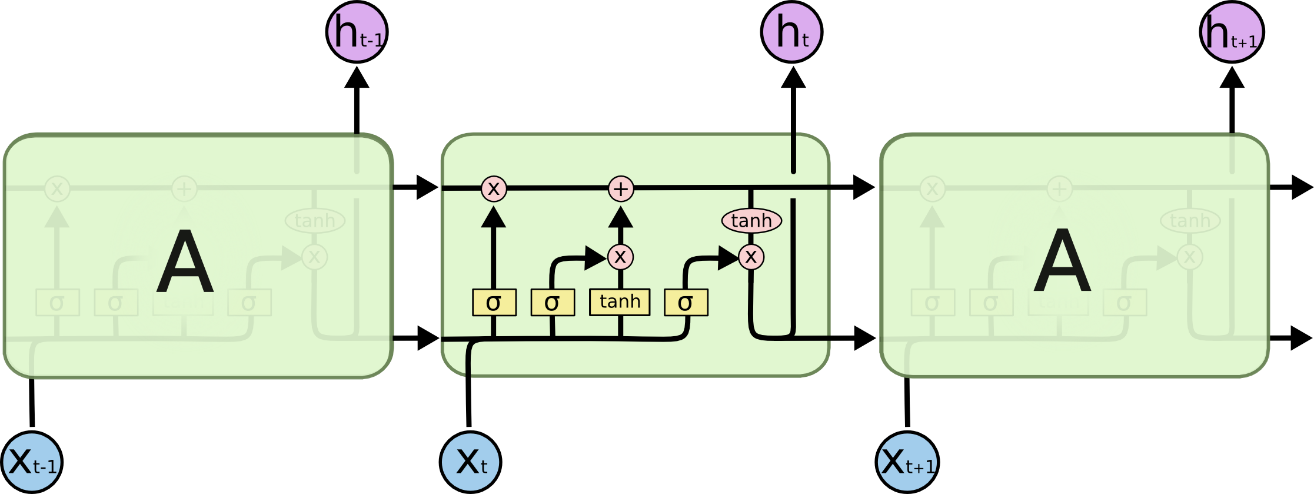
LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một cổng *tanh*.



Hình 2.9 Cấu trúc của một nút mạng RNN

LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron để nhận *đầu vào* và tính toán đưa ra *đầu ra*, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt. Chi tiết các tầng này sẽ được trình bày ở phần dưới

****

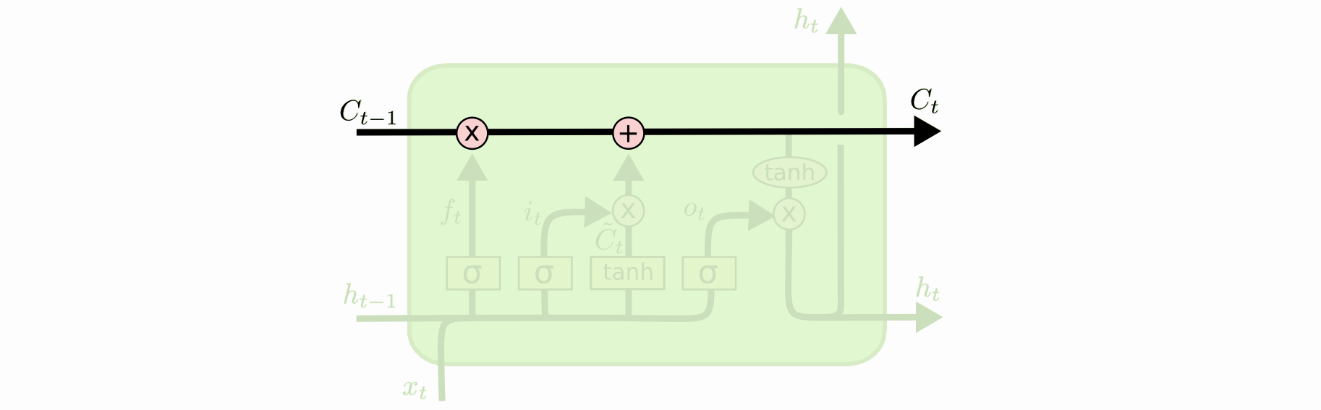
Hình 2.10 Cấu trúc của một nút mạng LSTM

Quy ước các ký tự trong sơ đồ trên như sau:



Hình 2.11 Các quy ước trong sơ đồ

Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ.



Hình 2.12 Trạng thái tế bào

Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate).

Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng *sigmoid* và một phép nhân.



Hình 2.13 Cấu trúc cổng sigmoid

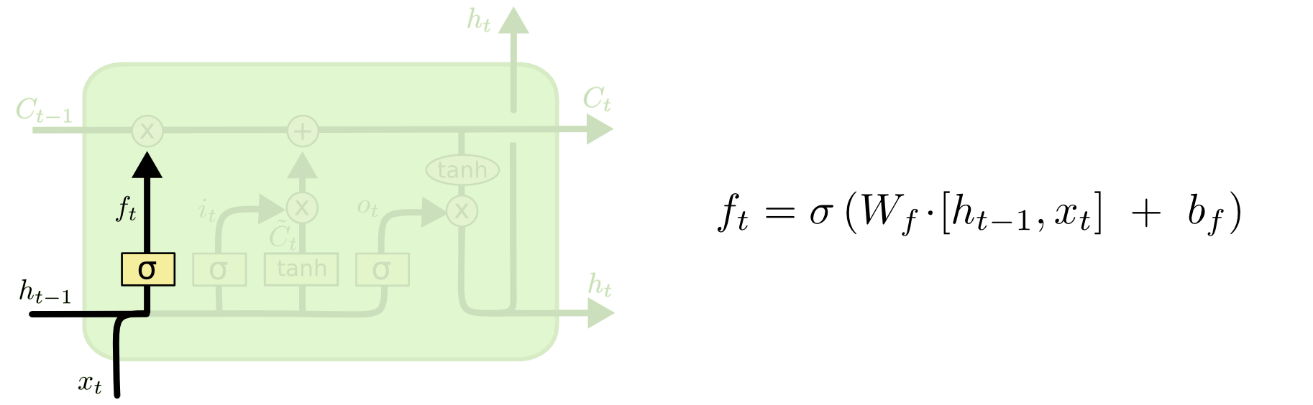
Tầng *sigmoid* sẽ cho đầu ra là một số trong khoản [0,1], mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 0 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 1 thì có nghĩa là cho tất cả các thông tin đi qua nó.

Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

Nguyên lý của mạng LSTM

Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng *sigmoid* - gọi là “tầng cổng quên” (forget gate layer). Nó sẽ lấy đầu vào là và rồi đưa ra kết quả là một số trong khoảng [0,1] cho mỗi số trong trạng thái tế bào . Đầu ra là 1 thể hiện rằng nó giữ toàn bộ thông tin lại, còn 0 chỉ rằng toàn bộ thông tin sẽ bị bỏ đi.

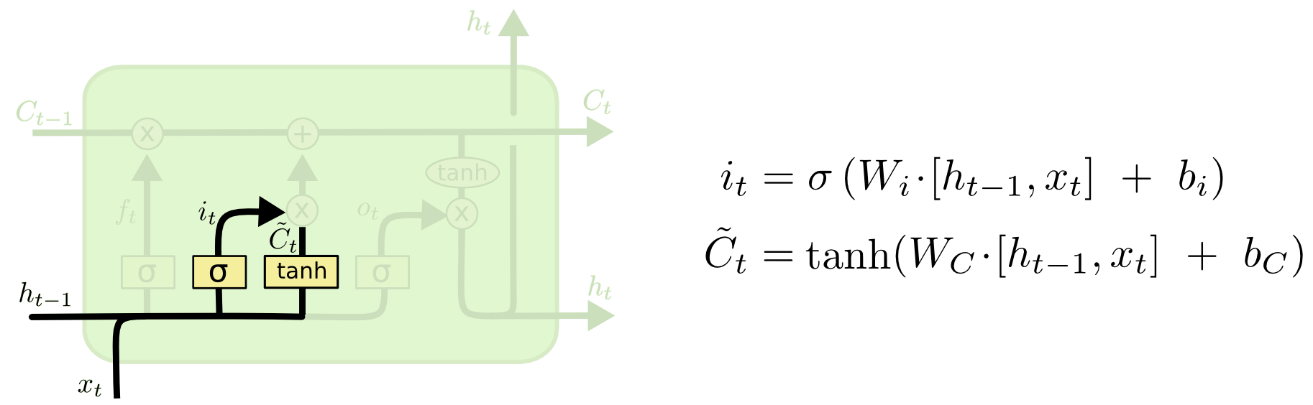
Quay trở lại với ví dụ mô hình ngôn ngữ dự đoán từ tiếp theo dựa trên tất cả các từ trước đó, với những bài toán như vậy, thì trạng thái tế bào có thể sẽ mang thông tin về giới tính của một nhân vật nào đó giúp ta sử dụng được đại từ nhân xưng chuẩn xác. Tuy nhiên, khi đề cập tới một người khác thì ta sẽ không muốn nhớ tới giới tính của nhân vật nữa, vì nó không còn tác dụng gì với chủ thế mới này.



Hình 2.14 Dữ liệu đi vào cổng “quên”

Bước tiếp theo là quyết định xem thông tin mới nào ta sẽ lưu vào trạng thái tế bào. Việc này gồm 2 phần. Đầu tiên là sử dụng một tầng *sigmoid* được gọi là “tầng cổng vào” (input gate layer) để quyết định giá trị nào ta sẽ cập nhập. Tiếp theo là một tầng *tanh* tạo ra một véc-tơ cho giá trị mới nhằm thêm vào cho trạng thái. Trong bước tiếp theo, ta sẽ kết hợp 2 giá trị đó lại để tạo ra một cập nhập cho trạng thái.

Chẳng hạn với ví dụ mô hình ngôn ngữ của ta, ta sẽ muốn thêm giới tính của nhân vật mới này vào trạng thái tế bào và thay thế giới tính của nhân vật trước đó.

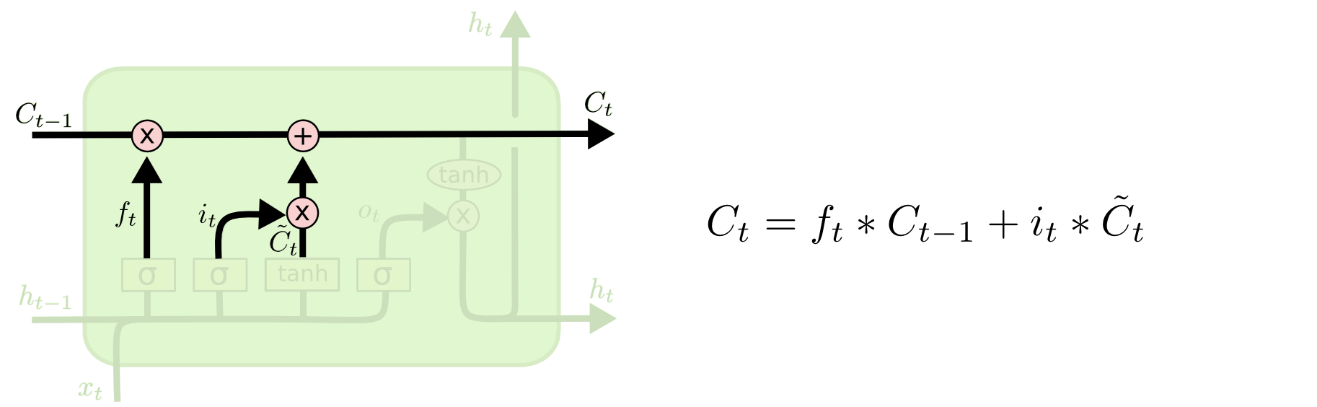


Hình 2.15 Dữ liệu đi qua cổng đầu vào

Giờ là lúc cập nhập trạng thái tế bào cũ ​ thành trạng thái mới ​. Ở các bước trước đó đã quyết định những việc cần làm, nên giờ ta chỉ cần thực hiện là xong.

Ta sẽ nhân trạng thái cũ với ​ để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm ∗​. Trạng thái mơi thu được này phụ thuộc vào việc ta quyết định cập nhập mỗi giá trị trạng thái ra sao.

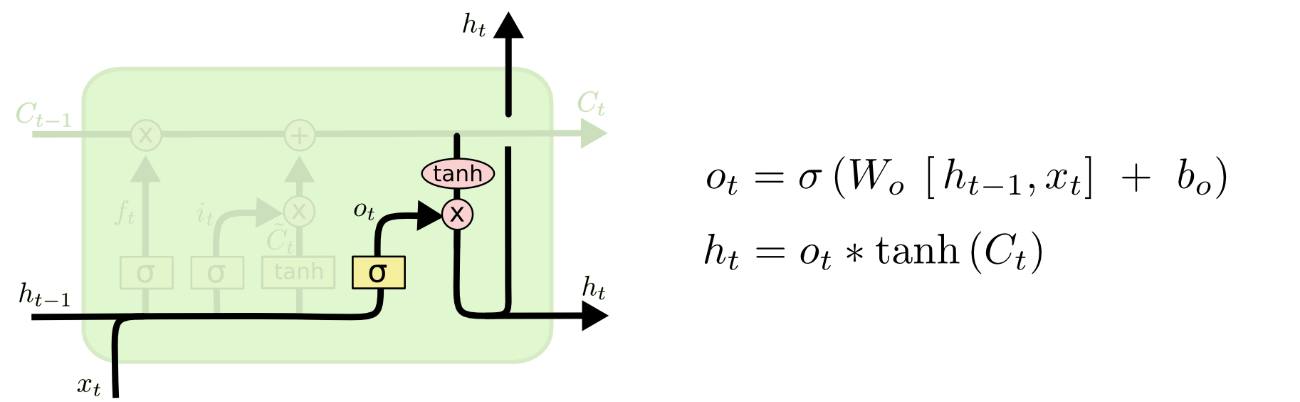
Với bài toàn mô hình ngôn ngữ, chính là việc ta bỏ đi thông tin về giới tính của nhân vật cũ, và thêm thông tin về giới tính của nhân vật mới như ta đã quyết định ở các bước trước đó.



Hình 2.16 Cập nhật trạng thái tế bào

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

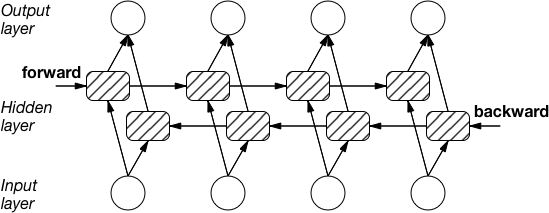
Với ví dụ về mô hình ngôn ngữ, chỉ cần xem chủ thể mà ta có thể đưa ra thông tin về một trạng từ đi sau đó. Ví dụ, nếu đầu ra của chủ thể là số ít hoặc số nhiều thì ta có thể biết được dạng của trạng từ đi theo sau nó phải như thế nào. [6]



Hình 2.17 Dữ liệu đi qua cổng đầu ra

Mạng nơ-ron Bidirectional Long Short Term Memory Neural Network

Việc nhận dạng chính xác tên riêng trong một đoạn văn bản phụ thuộc không chỉ vào các thông tin phía trước của từ đang xét mà còn cả các thông tin phía sau. Tuy nhiên, một kiến trúc LSTM truyền thống với một lớp duy nhất chỉ có thể dự đoán nhãn của từ hiện tại dựa trên thông tin có được từ các từ nằm trước đó. Bidirectional Long-Short Term Memory (BiLSTM) đã được tạo ra để khắc phục điểm yếu trên. Một kiến trúc BiLSTM thường chứa 2 mạng LSTM đơn được sử dụng đồng thời và độc lập để mô hình hoá chuỗi đầu vào theo 2 hướng: từ trái sang phải (forward LSTM) và từ phải sang trái (backward LSTM).

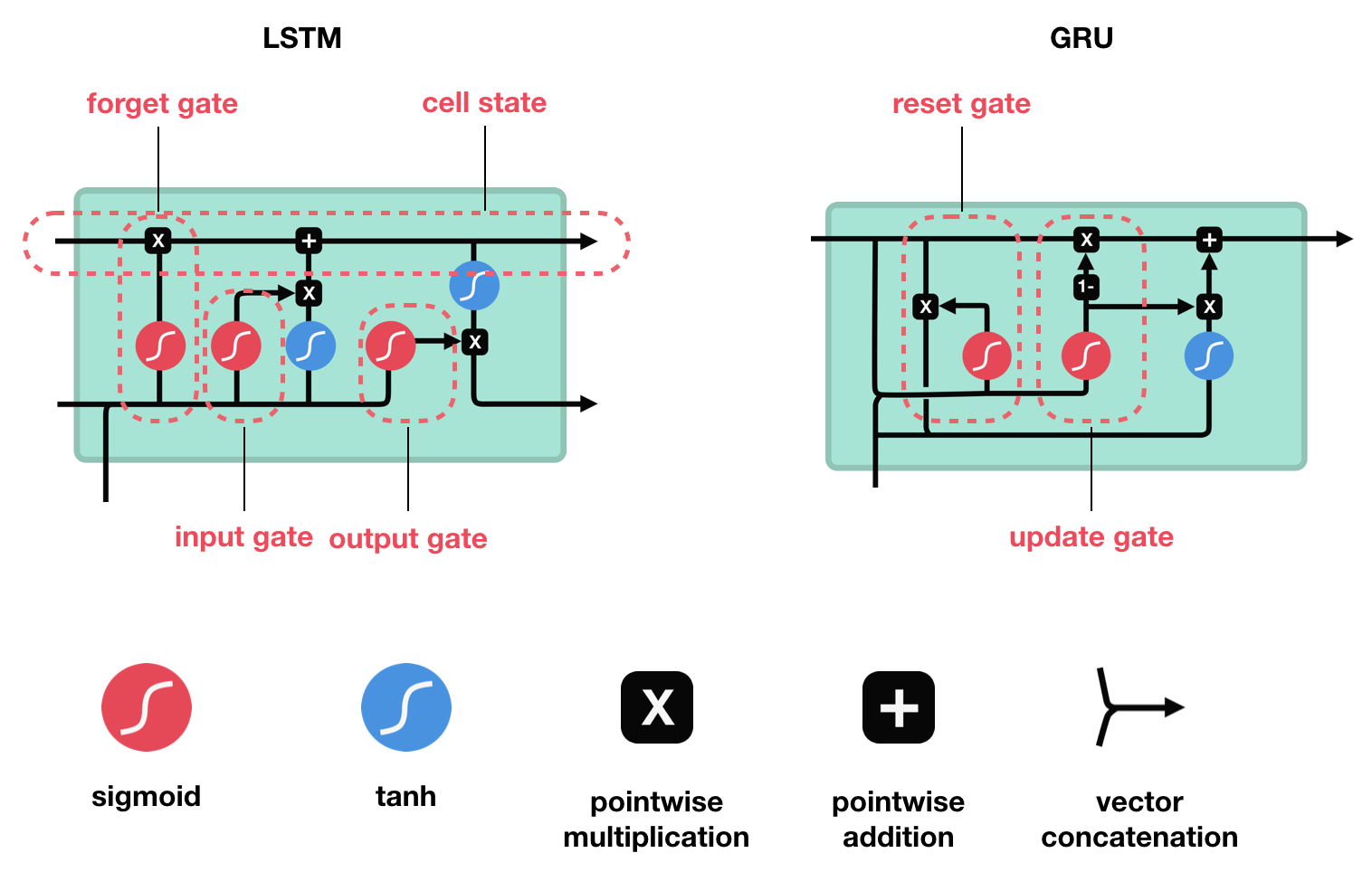


Hình 2.18 Sơ đồ mạng BiLSTM

### **2.4 Mạng nơ-ron Gated Recurrent Units**

Sau khi đã hiểu về cơ chế của LSTM, giờ hãy nhìn sơ qua về GRU. GRU chỉ có 2 cổng để chắt lọc thông tin, cổng reset và cổng update. Nó không có trạng thái tế bào mà chỉ có đầu ra vừa dùng để đưa ra quyết định vừa dùng để thông tin cho các bước tiếp theo.

Để so sánh cấu trúc của GRU với LSTM. Hãy quan sát hình sau là sơ đồ cấu trúc của một đơn vị cơ bản trong mạng LSTM và mạng GRU.

****

Hình 2.19 So sánh cấu trúc 2 nút mạng LSTM và GRU

Do mất đi trạng thái tế bào, nên tác dụng của 2 cổng trên khá khó để phân biệt rõ ràng, có thể nói là cả 2 cổng đều có tác dụng chắt lọc thông tin từ các đầu vào của tế bào để đưa ra một output thỏa mãn cả 2 tiêu chí là lưu giữ thông tin quá khứ và có khả năng đưa ra quyết định hiện tại một cách chính xác nhất. Do giảm bớt tính phức tạp nên GRU hoạt động nhanh hơn đôi chút so với LSTM. Về tính hiệu quả thì rất khó để đưa ra kết luận chính xác, nhưng có thể nói mức độ phổ biến LSTM là vượt trội khi so với GRU.

**Chương 3**

**PHƯƠNG PHÁP GIẢI QUYẾT**

**3.1 Thu thập dữ liệu**

Để có thể xây dựng hệ thống, việc đầu tiên là phải chuẩn bị được lượng dữ liệu sản phẩm. Tôi quyết định lựa chọn Tiki, Lazada, Shopee vì đây là 3 trang thương mại điện tử lớn với lượng người dùng cao, tương tác với hệ thống nhiều hơn sẽ có nhiều dữ liệu để đánh giá sản phẩm hơn.

Do cấu trúc mỗi trang web thương mại điện tử này khác nhau, nên cần phát triển các chương trình riêng để thu thập dữ liệu cho từng trang web này. Chúng ta gọi các chương trình này là các spider.

**3.1.1 Phương pháp thu thập**

Chương trình thu thập sẽ được lập trình trên ngôn ngữ lập trình Python phiên bản Python 3, tôi sẽ sử dụng các thư viện chính scarpy, Selenium, pyautogui, BeautifulSoup4 để thực hiện thu thập dữ liệu.

Chức năng các thư viện này có thể khái quát như sau:

*Scrapy* : Thư viện scrapy là thư viện dùng để thu thập dữ liệu web, cơ chế của thư viện này là tạo request đến các trang web và phân tích response của trang web đó để trích ra thông tin. Thư viện này sẽ được sử dụng để trích xuất url đến các trang sản phẩm.

*Selenium:* Đây là thư viện thao tác trên các trình duyệt web. Bao gồm các thao tác duyệt web, nhập văn bản, điều hướng trên web, phân tích các thẻ html trên trang web, thực thi mã javascript,…Thư viện này sẽ được sử dụng để tự động duyệt đến các trang web sản phẩm và tìm các thông tin từ các thẻ html trên trang sản phẩm đó.

*pyautogui:* Đây là thư viện tự động hóa thao tác với giao diện đồ họa của máy tính, các thao tác trên bàn phím, nhập ký tự,… Thư viện này được sử dụng để lưu lại các trang web sản phẩm để tiến hành trích xuất thông tin sau.

*BeautifulSoup4:* Thư viện phân tích cú pháp html và trích ra thông tin từ các thẻ html. Thư viện này sẽ được sử dụng để phân tích các tệp html đã được lưu lại để trích xuất thông tin sản phẩm.

Đặc điểm chung của các trang web thương mại điện tử Shopee, Tiki, Lazada là thông tin trên sản phẩm sẽ được tải lên trang web bởi mã javascript, như vậy phương pháp thu thập dữ liệu bằng việc phân tích response của trang web sẽ không thể thực hiện được. Vì vậy những thư viện thu thập thông tin trên web theo cơ chế gửi request và phân tích response như scrapy sẽ không có hiệu quả trong trường hợp này. Sau quá trình nghiên cứu các trang web và các thư viện, tôi quyết định sẽ sử dụng các thư viện thao tác với máy tính, trình duyệt web để tự động hóa quá trình lấy dữ liệu trên các trang thương mại điện tử này.

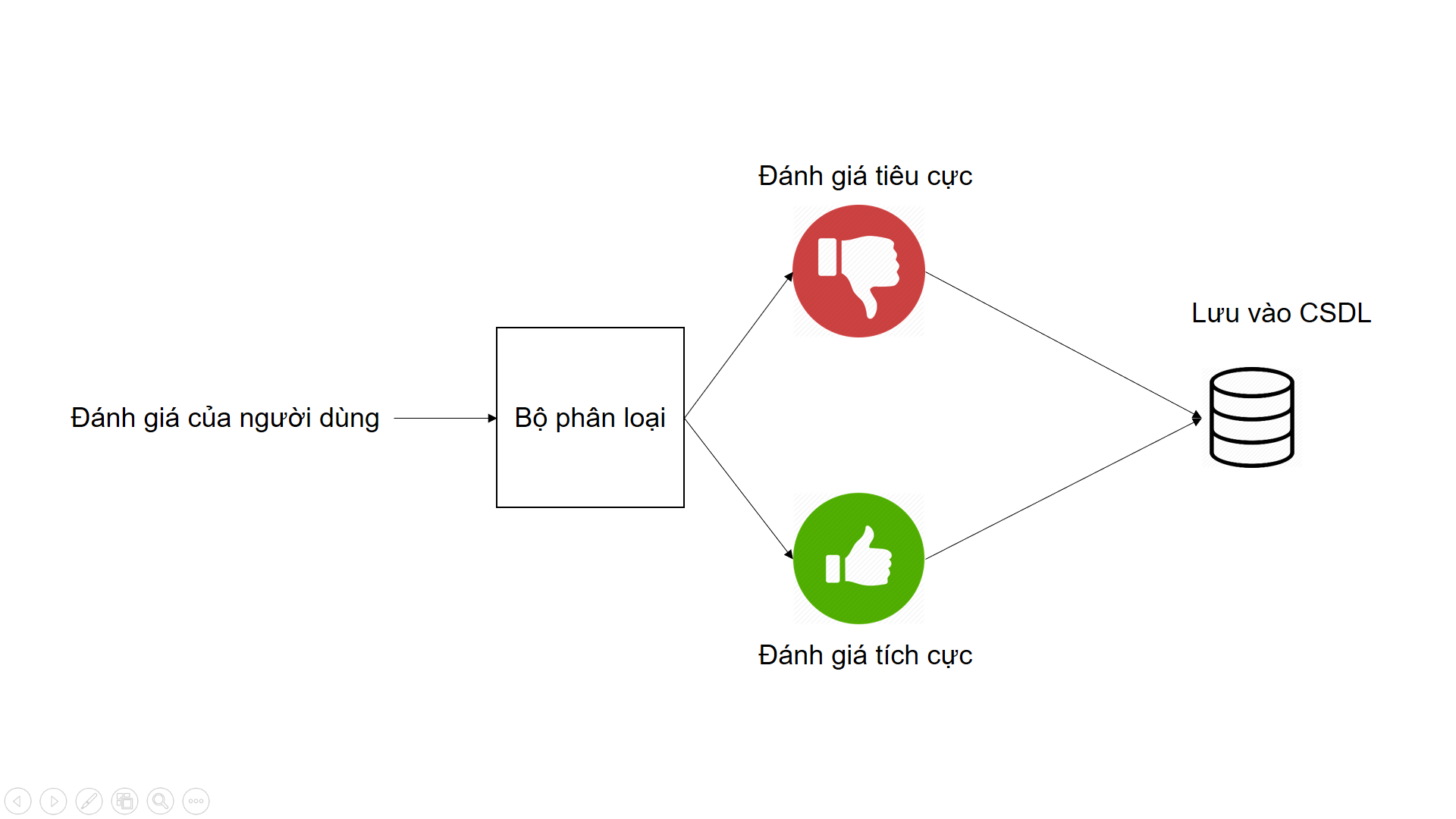
**3.2 Phân tích sắc thái phản hồi bằng các mô hình học sâu**

Ngoài việc đưa ra các thông tin về sản phẩm để hỗ trợ cho người dùng, một hệ thống thương mại điện tử tốt cần cho người dùng tiếp cận được với các đánh giá về sản phẩm đó được đưa ra bởi người dùng đã từng mua và trải nghiệm dịch vụ và sản phẩm. Các thông tin này là các thông tin hết sức thực tế và chiếm được niềm tin khá lớn từ người dùng. Vì người dùng khi đi mua hàng thường không có nhiều niềm tin vào các chiêu trò marketing của các nhà cung cấp thương mại điện tử, thay vào đó, họ có xu hướng chú ý đến đánh giá của những người đã từng mua và trải nghiệm sản phẩm.

Bài toán đặt ra đối với các hệ thống thương mại điện tử là phải có một cơ chế tiếp nhận, quản lý và phân loại các đánh giá này.

**3.2.1 Bài toán đánh giá quan điểm của người dùng.**

Đánh giá quan điểm người dùng đã mua và trải nghiệm sản phẩm và đưa các thông tin này đến người dùng mới giúp cho người dùng có thể có thêm các kinh nghiệm khi lựa chọn sản phẩm hoặc tránh đi các sản phẩm kém chất lượng. Hệ thống của đồ án được xây dựng mô đun để tiếp nhận và phân loại các đánh giá từ người dùng. Bài toán đặt ra như sau:



Hình 3.2.1 Bài toán đánh giá quan điểm người dùng

Người dùng nhập dữ liệu đầu vào là một đánh giá. Đánh giá sẽ được đưa vào một bộ phân loại đã được huấn luyện sẵn. Bộ phân loại sẽ gán nhãn đánh giá là **tích cực** hay **tiêu cực**, sau đó lưu chúng vào CSDL.

Hệ thống sẽ sử dụng mạng học sâu để huấn luyện mô hình.

### **3.2.2 Tạo tập dữ liệu để huấn luyện**

Để có được tập dữ liệu để huấn luyện, chương trình đã thu thập bình luận trên các trang Điện máy xanh: <https://www.dienmayxanh.com/>, Thế giới di động: <https://www.thegioididong.com/>, FPT Shop: <https://fptshop.com.vn/>, Tiki: <https://tiki.vn/>. , shopee: https://shopee.vn

Các đánh giá trên 4 trang này có đặc điểm chung là đều được đánh giá theo thang điểm từ 1 đến 5. Các đánh giá được chấm 4 hoặc 5 điểm đều là các đánh giá có thái độ tích cực, các đánh giá được chấm 1 hoặc 2 điểm đều là các đánh giá có thái độ tiêu cực, riêng đánh giá 3 điểm phụ thuộc nhiều trường hợp mà đánh giá này có thái độ tiêu cực hay tích cực.

Chương trình lấy các các đánh giá được chấm 4 hoặc 5 điểm và gán nhãn chúng là đánh giá tích cực, và lấy ra các đánh giá được chấm 1 hoặc 2 điểm và gán nhãn chúng là đánh giá tiêu cực.

Sau khi thực hiện lấy dữ liệu và gán nhãn, chương trình đã lấy ra được tập dữ liệu 20000 đánh giá với 10000 đánh giá tích cực và 10000 đánh giá tiêu cực. Để tiến hành áp dụng các mô hình học sâu trên bộ dữ liệu này. Tôi sẽ chia bộ dữ liệu thành 16000 mẫu để huấn luyện và 4000 để tiến hành đánh giá. Số lượng mẫu của mỗi lớp tiêu cực và tích cực được chia đều theo tỷ lệ 50:50 trong cả 2 tập train và test.

Sau khi chuẩn bị xong tập dữ liệu, chúng ta sẽ áp dụng các mô hình học sâu trên các tập dữ liệu này.

**3.2.3 Mô hình LSTM**

Mô hình LSTM đã đạt được rất nhiều kết quả tốt trong nhiều lĩnh vực, đặc biệt là Xử lý Ngôn ngữ Tự nhiên nhờ khắc phục được vấn đề phụ thuộc xa (Long term dependencies). Tôi sẽ áp dụng mô hình LSTM với bộ dữ liệu đánh giá người dùng đã thu thập được từ các trang thương mại điện tử để giải quyết bài toán đánh giá quan điểm người dùng.

Công việc được thực hiện theo 3 bước:

1. Chuẩn bị dữ liệu

2. Xây dựng và huấn luyện mô hình

3. Kiểm tra và đánh giá mô hình.

**Chuẩn bị dữ liệu**

Giai đoạn chuẩn bị dữ liệu bao gồm:

*Chuẩn hóa dữ liệu:* Các đánh giá được làm sạch bằng cách loại bỏ đi các stopword (VD: “vừa”, “nên”, …), các từ có ký tự kéo dài (VD “Cái áo đẹp quaaaaaaaaaaaaaaa”,…), các từ theo kiểu teencode được chuyển về các từ chuẩn hơn (VD: “hok” thành “không”, “ok” thành “tốt”,…), các kí tự icon được chuyển đổi thành nhãn tích cực tiêu cực ( VD: :D thành “positive” , ☹ thành “negative”,…) , chuyển câu có nghĩa phủ định như “ Áo này chẳng đẹp “ thành “ Áo này negative”.

*Tạo từ điển các từ và biến các câu văn bản thành véc tơ:* Tiến hành tạo từ điển các từ xuất hiện trong tập dữ liệu huấn luyện và véc tơ hóa các câu văn bản bằng chỉ mục của các từ trong từ điển. Loại bỏ đi các từ chỉ xuất hiện 1 lần (Các từ này có xác suất là các từ người dùng viết sai, VD như “thuộ tính”,…) Các từ không có trong từ điển sẽ đưa về giá trị 0

VD: Ta có từ điển sau: {“Yêu”: 1, “Hát”: 2, “Tôi”: 3, “Thích”: 4 }

Với câu văn bản “Tôi rất thích hát” sau khi map với từ điển ta sẽ có véc tơ

[3, 0, 4, 2]

*Tiến hành padding hoặc truncating véc tơ dữ liệu.* Các véc tơ có số chiều khác nhau vì đánh giá có câu dài câu ngắn. Điều này khiến cho dữ liệu không thể đưa vào mô hình huấn luyện được, ta cần sử dụng zero padding khi véc tơ có độ dài ngắn hơn yêu cầu của mô hình và ngược lại cắt bỏ nếu véc tơ có chiều dài dài hơn.

*Chuyển nhãn dữ liệu về dạng 0,1:* Các đánh giá tích cực sẽ có nhãn là 0, các đánh giá tiêu cực sẽ có nhãn là 1.

**Huấn luyện mô hình.**

Giai đoạn này bao gồm:

*Xây dựng mô hình:*

Mô hình được xây dựng có kiến trúc như sau:

1. Lớp Embedding: Chuyển các véc tơ của câu văn bản đầu vào thành véc tơ embedding của văn bản với số chiều được định nghĩa trước.

2. Lớp LSTM: Nhận đầu vào từ lớp Embedding, tiến hành tính toán và đưa ra đầu ra có số chiều được định nghĩa từ trước.

3. Lớp Fully-connected: Lớp nhận đầu vào từ lớp LSTM và xử lý và đưa ra *đầu ra* với chiều được định nghĩa từ trước.

4. Lớp Sigmoid: Biến đầu ra từ lớp Fully-connected về khoảng từ 0 đến 1

5. Lớp đầu ra: Giá trị đầu ra từ lớp Sigmoid sẽ quyết định xem câu văn bản thuộc về lớp nào.

Triển khai mô hình sử dụng ngôn ngữ lập trình Python cùng thư viện Keras. Các siêu tham số được định nghĩa như sau:

Độ dài véc tơ văn bản trích từ từ điển 200

Độ dài véc tơ embedding: 200

Số chiều của đầu ra của lớp LSTM: 100

Số chiều của đầu ra của lớp Fully-connected: 1

*Huấn luyện mô hình.*

Mô hình được huấn luyện sử dụng thuật toán tối ưu Adam và hàm lỗi Binary Cross Entropy do đây là bài toán phân loại 2 lớp.

Các siêu tham số như sau:

batch size (Kích thước từng lô dữ liệu huấn luyện): 32

epoch (Số lần huấn luyện qua toàn bộ tập dữ liệu): 3

**Kiểm tra và đánh giá mô hình**

Tôi sử dụng tập kiểm tra với 2400 đánh giá, trong đó có 1200 đánh giá tích cực và 1200 đánh giá tiêu cực.

Đánh giá trên bộ dữ liệu test, tôi thu được độ chính xác (accuracy) là %

Bảng 3.1 Đánh giá mô hình LSTM

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F1-score |
| Tích cực (0) | **0.913** | 0.927 | 0.920 |
| Tiêu cực (1) | 0.928 | 0.914 | 0.921 |
| avg / total | 0.92 | 0.92 | 0.92 |

Kết quả thu được ban đầu khá tốt đối với bộ dữ liệu này.

### **3.2.2 Mô hình BiLSTM**

Mô hình BiLSTM không chỉ học được các đặc trưng liên kết theo chiều xuôi của câu văn bản, mà còn có khả năng học được các đặc trưng liên kết theo chiều ngược lại của câu văn bản. Tôi sẽ áp dụng mô hình BiLSTM trên bộ dữ liệu để giải quyết bài toán đánh giá quan điểm người dùng để đánh giá khả năng áp dụng của mô hình BiLSTM.

Công việc được thực hiện theo 3 bước:

1. Chuẩn bị dữ liệu

2. Xây dựng và huấn luyện mô hình

3. Kiểm tra và đánh giá mô hình.

**Chuẩn bị dữ liệu**

Giai đoạn chuẩn bị dữ liệu bao gồm:

*Chuẩn hóa dữ liệu:* Các đánh giá được làm sạch bằng cách loại bỏ đi các stopword (VD: “vừa”, “nên”, …), các từ có ký tự kéo dài (VD “Cái áo đẹp quaaaaaaaaaaaaaaa”,…), các từ theo kiểu teencode được chuyển về các từ chuẩn hơn (VD: “hok” thành “không”, “ok” thành “tốt”,…)

*Tạo từ điển các từ và biến các câu văn bản thành véc tơ:* Tiến hành tạo từ điển các từ xuất hiện trong tập dữ liệu huấn luyện và véc tơ hóa các câu văn bản bằng chỉ mục của các từ trong từ điển. Loại bỏ đi các từ chỉ xuất hiện 1 lần (Các từ này có xác suất là các từ người dùng viết sai, VD như “thuộ tính”,…) Các từ không có trong từ điển sẽ đưa về giá trị 0

VD: Ta có từ điển sau: {“Yêu”: 1, “Hát”: 2, “Tôi”: 3, “Thích”: 4 }

Với câu văn bản “Tôi rất thích hát” sau khi map với từ điển ta sẽ có véc tơ

[3, 0, 4, 2]

*Tiến hành padding hoặc truncating véc tơ dữ liệu.* Các véc tơ có số chiều khác nhau vì đánh giá có câu dài câu ngắn. Điều này khiến cho dữ liệu không thể đưa vào mô hình huấn luyện được, ta cần sử dụng zero padding khi véc tơ có độ dài ngắn hơn yêu cầu của mô hình và ngược lại cắt bỏ nếu véc tơ có chiều dài dài hơn.

*Chuyển nhãn dữ liệu về dạng 0,1:* Các đánh giá tích cực sẽ có nhãn là 0, các đánh giá tiêu cực sẽ có nhãn là 1.

**Huấn luyện mô hình.**

Giai đoạn này bao gồm:

*Xây dựng mô hình:*

Mô hình được xây dựng có kiến trúc như sau:

1. Lớp Embedding: Chuyển các véc tơ của câu văn bản đầu vào thành véc tơ embedding của văn bản với số chiều được định nghĩa trước.

2. Lớp BiLSTM: Nhận đầu vào từ lớp Embedding, tiến hành tính toán và đưa ra đầu ra có số chiều được định nghĩa từ trước. Kiến trúc của mô hình này không khác so với mô hình LSTM đã được triển khai ở mục trước, chỉ khác ta thay lớp LSTM bằng lớp BiLSTM, tức lớp mạng LSTM 2 chiều.

3. Lớp Fully-connected: Lớp nhận đầu vào từ lớp LSTM và xử lý và đưa ra *đầu ra* với chiều được định nghĩa từ trước.

4. Lớp Sigmoid: Biến đầu ra từ lớp Fully-connected về khoảng từ 0 đến 1

5. Lớp đầu ra: Giá trị đầu ra từ lớp Sigmoid sẽ quyết định xem câu văn bản thuộc về lớp nào.

Triển khai mô hình sử dụng ngôn ngữ lập trình Python cùng thư viện Keras. Các siêu tham số được định nghĩa như sau:

Độ dài véc tơ văn bản trích từ từ điển 200

Độ dài véc tơ embedding: 100

Số chiều của đầu ra của lớp LSTM: 100

Số chiều của đầu ra của lớp Fully-connected: 1

*Huấn luyện mô hình.*

Mô hình được huấn luyện sử dụng thuật toán tối ưu Adam và hàm lỗi Binary Cross Entropy do đây là bài toán phân loại 2 lớp.

Các siêu tham số như sau:

batch size (Kích thước từng lô dữ liệu huấn luyện): 32

epoch (Số lần huấn luyện qua toàn bộ tập dữ liệu): 3

**Kiểm tra và đánh giá mô hình**

Tôi sử dụng tập kiểm tra với 2400 đánh giá, trong đó có 1200 đánh giá tích cực và 1200 đánh giá tiêu cực.

Đánh giá trên bộ dữ liệu test, tôi thu được độ chính xác (accuracy) là 92.7%

Bảng 3.2 Đánh giá mô hình BiLSTM

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F1-score |
| Tích cực (0) | 0.931 | 0.920 | 0.925 |
| Tiêu cực (1) | 0.923 | 0.933 | 0.928 |
| avg / total | 0.925 | 0.925 | 0.925 |

Như vậy, BiLSTM cho kết quả tốt hơn rõ rệt so với LSTM,

**3.2.3 Mô hình GRU**

Khác với LSTM, GRU là mô hình mới được đề xuất vào năm 2014, đây là mô hình khá mới nếu so sánh với LSTM được đề xuất vào năm 1997. GRU cũng sử dụng các cổng để thực hiện việc học và quên dữ liệu, nhưng thay vì có 3 cổng như LSTM, GRU chỉ có 2 cổng là cổng thiết lập lại và cổng cập nhật. Do sử dụng ít cổng nên GRU có ít tham số hơn LSTM và GRU sẽ chạy nhanh hơn LSTM. Nhưng nếu so về hiệu quả thì một số trường hợp GRU lại tỏ ra tốt hơn LSTM, đặc biệt là các trường hợp ít dữ liệu. Tuy nhiên nếu có lượng dữ liệu đủ lớn LSTM lại tỏ rõ sức mạnh của mình và kết quả tốt hơn GRU. Tôi sẽ áp dụng mô hình GRU để đánh giá khả năng của GRU trên bộ dữ liệu này.

Công việc được thực hiện theo 3 bước:

1. Chuẩn bị dữ liệu

2. Xây dựng và huấn luyện mô hình

3. Kiểm tra và đánh giá mô hình.

**Chuẩn bị dữ liệu**

Giai đoạn chuẩn bị dữ liệu bao gồm:

*Chuẩn hóa dữ liệu:* Các đánh giá được làm sạch bằng cách loại bỏ đi các stopword (VD: “vừa”, “nên”, …), các từ có ký tự kéo dài (VD “Cái áo đẹp quaaaaaaaaaaaaaaa”,…), các từ theo kiểu teencode được chuyển về các từ chuẩn hơn (VD: “hok” thành “không”, “ok” thành “tốt”,…)

*Tạo từ điển các từ và biến các câu văn bản thành véc tơ:* Tiến hành tạo từ điển các từ xuất hiện trong tập dữ liệu huấn luyện và véc tơ hóa các câu văn bản bằng chỉ mục của các từ trong từ điển. Loại bỏ đi các từ chỉ xuất hiện 1 lần (Các từ này có xác suất là các từ người dùng viết sai, VD như “thuộ tính”,…) Các từ không có trong từ điển sẽ đưa về giá trị 0

VD: Ta có từ điển sau: {“Yêu”: 1, “Hát”: 2, “Tôi”: 3, “Thích”: 4 }

Với câu văn bản “Tôi rất thích hát” sau khi map với từ điển ta sẽ có véc tơ

[3, 0, 4, 2]

*Tiến hành padding hoặc truncating véc tơ dữ liệu.* Các véc tơ có số chiều khác nhau vì đánh giá có câu dài câu ngắn. Điều này khiến cho dữ liệu không thể đưa vào mô hình huấn luyện được, ta cần sử dụng zero padding khi véc tơ có độ dài ngắn hơn yêu cầu của mô hình và ngược lại cắt bỏ nếu véc tơ có chiều dài dài hơn.

*Chuyển nhãn dữ liệu về dạng 0,1:* Các đánh giá tích cực sẽ có nhãn là 0, các đánh giá tiêu cực sẽ có nhãn là 1.

**Huấn luyện mô hình.**

Giai đoạn này bao gồm:

*Xây dựng mô hình:*

Mô hình được xây dựng có kiến trúc như sau:

1. Lớp Embedding: Chuyển các véc tơ của câu văn bản đầu vào thành véc tơ embedding của văn bản với số chiều được định nghĩa trước.

2. Lớp GRU: Nhận đầu vào từ lớp Embedding, tiến hành tính toán và đưa ra đầu ra có số chiều được định nghĩa từ trước.

3. Lớp Fully-connected: Lớp nhận đầu vào từ lớp GRU và xử lý và đưa ra *đầu ra* với chiều được định nghĩa từ trước.

4. Lớp Sigmoid: Biến đầu ra từ lớp Fully-connected về khoảng từ 0 đến 1

5. Lớp đầu ra: Giá trị đầu ra từ lớp Sigmoid sẽ quyết định xem câu văn bản thuộc về lớp nào.

Triển khai mô hình sử dụng ngôn ngữ lập trình Python cùng thư viện Keras. Các siêu tham số được định nghĩa như sau:

Độ dài véc tơ văn bản trích từ từ điển 200

Độ dài véc tơ embedding: 100

Số chiều của đầu ra của lớp GRU: 100

Số chiều của đầu ra của lớp Fully-connected: 1

*Huấn luyện mô hình.*

Mô hình được huấn luyện sử dụng thuật toán tối ưu Adam và hàm lỗi Binary Cross Entropy do đây là bài toán phân loại 2 lớp.

Các siêu tham số như sau:

batch size (Kích thước từng lô dữ liệu huấn luyện): 32

epoch (Số lần huấn luyện qua toàn bộ tập dữ liệu): 3

**Kiểm tra và đánh giá mô hình**

Tôi sử dụng tập kiểm tra với 2400 đánh giá, trong đó có 1200 đánh giá tích cực và 1200 đánh giá tiêu cực.

Đánh giá trên bộ dữ liệu test, tôi thu được độ chính xác (accuracy) là 91,66%

Bảng 3.3 Đánh giá mô hình GRU

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F1-score |
| Tích cực (0) | 0.929 | 0.912 | 0.921 |
| Tiêu cực (1) | 0.916 | 0.932 | 0.924 |
| avg / total | 0.92 | 0.92 | 0.92 |

GRU cho kết quả tốt hơn LSTM trong việc dự đoán các đánh giá tích cực . Do đây là bộ dữ liệu nhỏ nên GRU thể hiện tốt hơn LSTM.

Vậy kết luận trong 3 mô hình thì BiLSTM cho ta kết quả tốt hơn cả. Bởi vì lúc tiền xử lý dữ liệu đầu vào chúng ta đã gán thêm các đặc trưng positive hay negative vào cuối của câu. Điều này giúp cho BiLSTM đoán chuẩn xác hơn.

**3.3. Thống kê , xây dựng biểu đồ so sánh các mối quan tâm**