LỜI CẢM ƠN

Trước hết, em xin được chân thành cảm ơn thầy Ngô Thanh Huy, là nguồn động viên và người hướng dẫn tận tình của em trong suốt hành trình nghiên cứu và hoàn thành đồ án chuyên ngành. Sự am hiểu sâu sắc của thầy về chủ đề này đã giúp em vượt qua những thách thức khó khăn, từ việc chọn đề tài đến quá trình triển khai và đánh giá kết quả. Em rất biết ơn những lời chỉ bảo chân thành và những lời khuyên quý báu của thầy, đã giúp em không chỉ hoàn thành đồ án mà còn có được những hiểu biết sâu rộng về lĩnh vực này.

Ngoài ra, em xin gửi lời cảm ơn chân thành đến tất cả các bạn. Sự giúp đỡ, hỗ trợ, và chia sẻ kiến thức của các bạn đã tạo nên một môi trường làm việc tích cực và tràn đầy năng lượng để em có thể phấn đấu hoàn thành đồ án.

Một lần nữa, em xin chân thành cảm ơn thầy và các bạn đã đóng góp vào sự thành công của đồ án này. Em hy vọng rằng kiến thức và kinh nghiệm thu được từ đồ án này sẽ là nguồn động lực mạnh mẽ, làm nền tảng cho sự phát triển và thành công trong tương lai.

MỤC LỤC

DANH MỤC TỪ VIẾT TẮT

LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
BPTT	Backpropagation Through Time
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error

DANH MỤC HÌNH ẢNH – BẢNG BIỂU

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Đồ án trình bày kết quả của nghiên cứu xây dựng một mô hình LSTM đơn giản để dự báo dữ liệu dòng thời gian với kiểu dữ liệu đơn biến. Số liệu sử dụng cho mô hình là số liệu chỉ số chất lượng không khí của Thành phố Hồ Chí Minh trong 2 năm 2021–2022. Kết quả thiết lập mô hình cho ra các chỉ số đánh giá RMSE và MAPE tốt nhưng vẫn có hạn chế đáng kể ở tập huấn luyện, do đó mô hình cần được chỉnh sửa lại để nâng cao hiệu suất dự báo, từ đó hoàn thiện để ứng dụng trong dự đoán dữ liệu dòng thời gian.

MỞ ĐẦU

1. Lý do chọn đề tài

Mạng LSTM (viết tắt của Long Short-Term Memory) là một trong những mô hình mạng neural phổ biến nhất hiện nay trong lĩnh vực dự báo dữ liệu dòng thời gian. Khả năng của LSTM trong việc ghi nhớ thông tin đã học trước đó và lựa chọn các thông tin quan trọng từ dữ liệu giúp nâng cao hiệu suất dự đoán.

Trong thời đại công nghệ hiện đại, với sự phát triển của trí tuệ nhân tạo, việc áp dụng mô hình LSTM để dự đoán dữ liệu dòng thời gian có ý nghĩa lớn. Đối với nhiều lĩnh vực như tài chính, y tế, thời tiết, và nhiều lĩnh vực khác, khả năng dự đoán chính xác về dữ liệu dòng thời gian là chìa khóa quan trọng để tối ưu hóa quy trình, giảm rủi ro và tăng cường hiệu suất.

Vì thế em đã chọn đề tài "Mô hình LSTM cho dự đoán dữ liệu dòng thời gian". Bài viết sẽ trình bày các kiến thức cơ bản về mạng LSTM và ứng dụng của nó trong lĩnh vực dự đoán dữ liệu dòng thời gian, cũng như giới thiệu phương pháp tiếp cận để giải quyết bài toán dự đoán. Em hy vọng rằng việc nghiên cứu và triển khai mạng LSTM giúp nắm bắt những tiến bộ mới nhất và hiểu rõ hơn về cách xử lý và dự đoán dữ liệu dòng thời gian, mang lại kết quả độ chính xác cao và ứng dụng rộng rãi trong các lĩnh vực thực tế.. Đây là một đóng góp quan trọng và có thể mở ra những hướng nghiên cứu mới trong tương lai.

2. Mục đích

Mục đích của đề tài này là tìm hiểu các kiến thức cơ bản về mạng LSTM như cấu trúc và cách hoạt động của mô hình, từ đó áp dụng mô hình LSTM trong việc dự đoán dữ liệu dòng thời gian thông qua tập dữ liệu và đánh giá hiệu suất của nó.

3. Đối tượng

Đối tượng của nghiên cứu là các nhà nghiên cứu, sinh viên, giảng viên, chuyên gia làm việc trong lĩnh vực phân tích dữ liệu và dự đoán xu hướng hoặc bất cứ ai khác có quan tâm đến ứng dụng của LSTM.

4. Phạm vi nghiên cứu

Phạm vi nghiên cứu sẽ bao gồm tìm hiểu về nghiên cứu lý thuyết về mô hình LSTM, cách hoạt động và cấu trúc của nó. Đồng thời, đồ án cũng sẽ nêu bài toán dự đoán dòng thời gian, xây dựng một mô hình mạng LSTM đơn giản cho bài toán này, thử nghiệm và đánh giá hiệu suất của mô hình trong việc dự đoán dữ liệu dòng thời gian.

Tuy nhiên, đồ án này chỉ tập trung vào dữ liệu dòng thời gian đơn biến và mô hình LSTM cổ điển mà không đi sâu vào các biến thể của nó.

CHƯƠNG 1: TỔNG QUAN

1.1 Sơ lược về sự hình thành và phát triển của mạng LSTM

Mạng LSTM (Long Short-Term Memory – bộ nhớ ngắn hạn dài) là một dạng mạng neural học sâu có khả năng xử lý và hiểu được thông tin từ dữ liệu dòng thời gian một cách hiệu quả. Nó được đề xuất lần đầu tiên bởi Hochreiter và Schmidhuber vào năm 1997 nhằm giải quyết vấn đề biến mất đạo hàm của các mạng RNN (Recurrent Neural Network – mạng thần kinh hồi quy) truyền thống trong quá trình huấn luyện mô hình.

Từ khi được đề xuất đến nay, mạng LSTM đã trải qua nhiều bước phát triển và cải tiến để nâng cao hiệu suất và khả năng ứng dụng. Một số bước phát triển quan trọng của mạng LSTM có thể kể đến như sau:

- Năm 1999, cổng quên được bổ sung cho mạng LSTM, cho phép mạng có thể quyết định khi nào nên bỏ đi các thông tin không cần thiết trong bộ nhớ.
- Năm 2000, kỹ thuật học phối hợp được đề xuất (learning to forget), giúp mạng LSTM có thể tự động điều chỉnh các trọng số của cổng quên để quên các thông tin không liên quan.
- Năm 2009, kỹ thuật dropout (loại bỏ ngẫu nhiên một số neural) được áp dụng cho mạng LSTM, giúp giảm thiểu hiện tượng quá khớp (overfitting) và tăng khả năng khái quát hóa của mạng.
- Năm 2014, một phiên bản đơn giản hơn của mạng LSTM được đề xuất, gọi là mạng GRU (Gated Recurrent Unit), bằng cách gộp cổng đầu vào và cổng quên thành một cổng cập nhật (update gate) và loại bỏ đường kết nối bên trong đơn vị bộ nhớ.
- Năm 2015, Greff và cộng sự tiến hành so sánh và đánh giá các biến thể của mạng LSTM, và kết luận rằng không có sự khác biệt đáng kể về hiệu suất giữa các biến thể, và mạng LSTM tiêu chuẩn vẫn là lựa chọn tốt nhất cho hầu hết các bài toán.

Sự hình thành và phát triển của mạng LSTM đánh dấu bước đột phá quan trọng trong lĩnh vực học máy. LSTM không chỉ phá vỡ các kỷ lục về đổi mới dịch máy, mô hình hóa ngôn ngữ và xử lý ngôn ngữ đa ngôn ngữ mà còn cho thấy những tiến bộ đáng chú ý trong việc kết hợp với mạng CNN (Convolution Neural Network). Sự hợp tác này đã cải thiện đáng kể độ chính xác của chú thích hình ảnh tự động, góp phần đáng kể vào tiến bộ của lĩnh vực này.

Trong phần tiếp theo, đồ án sẽ trình bày về mạng LSTM và tìm hiểu về iệc xây dựng một mô hình mạng LSTM để dự đoán dữ liệu dòng thời gian. Vấn đề này đang được quan tâm rất nhiều bởi tính ứng dụng cao của nó trong các lĩnh vực khác nhau, nhất là lĩnh vực dự báo biến động và xu hướng. Ý tưởng chính để thực hiện được dựa trên khả năng ghi nhớ thông tin trong thời gian dài mà không bị mất mát của mạng LSTM, từ đó kết hợp thông tin từ quá khứ để đưa ra dự đoán cho hiện tại.

1.2 Một số ứng dụng của mạng LSTM

Mạng LSTM đã được chứng minh là một công cụ mạnh mẽ và linh hoạt để giải quyết các bài toán trong nhiều lĩnh vực ứng dụng. Một số ứng dụng tiêu biểu của mạng LSTM là:

- Dự đoán dữ liệu dòng thời gian: Mạng LSTM được sử dụng rộng rãi để dự đoán các giá trị trong tương lai dựa trên dữ liệu quá khứ như dự đoán giá cổ phiếu, nhu cầu điện, lưu lượng giao thông, ….
- Xử lý ngôn ngữ tự nhiên: LSTM thường được sử dụng để mô hình hóa và dự đoán các chuỗi từ trong các tác vụ như dịch máy, tóm tắt văn bản, sinh văn bản, phân tích cảm xúc, ….
- Phân loại dữ liệu dòng thời gian: Mạng LSTM có thể phân loại các dòng thời gian thành các nhãn hay hạng mục, ví dụ như phân loại hoạt động của người dùng, phát hiện bất thường, nhận diện chữ viết tay, ….
- Sinh dữ liệu dòng thời gian: Mạng LSTM có thể sinh ra các dòng thời gian mới có tính chất tương tự với dữ liệu huấn luyện, ví dụ như sinh âm thanh, âm nhạc,
- Nhận diện giọng nói: Mạng LSTM có khả năng nhận diện và xử lý các đặc trưng trong dữ liệu âm thanh theo thời gian.

1.3 Giới thiệu về dữ liệu dòng thời gian

Dữ liệu dòng thời gian là tập hợp các điểm dữ liệu được thu thập theo các khoảng thời gian (ngày, tháng, năm, …) biểu thị sự thay đổi của một hay nhiều yếu tố nào đó theo thời gian. Mỗi quan sát được thu thập có mối quan hệ liên tục với các quan sát trước và sau đó. Đây có thể là dữ liệu về giá cổ phiếu, thời tiết, sản lượng hàng ngày, lưu lượng truy cập trang web, hoặc bất kỳ loại dữ liệu nào có tính chất thời gian.

Dữ liệu dòng thời gian thể hiện tính biến động, có sự tăng giảm bất định, theo thời gian hay bị ảnh hưởng với thời gian. Các yếu tố quan trọng cần lưu ý khi phân tích

dữ liệu dòng thời gian bao gồm: xu hướng (trend), tính mùa vụ (seasonality), chu kỳ (cycle) và nhiễu (noise).

Dữ liệu dòng thời gian được ứng dụng rộng rãi trong nhiều lĩnh vực. Trong kinh doanh, dữ liệu dòng thời gian được ứng dụng nhiều trong việc phân tích giao dịch, số liệu bán hàng, hành vi của khách hàng, xu hướng trong các hoạt động trong các doanh nghiệp. Trong khoa học, dữ liệu dòng thời gian được sử dụng để theo dõi các hiện tượng tự nhiên như lượng mưa, nhiệt độ. Trong công nghệ thông tin, dữ liệu dòng thời gian được sử dụng để theo dõi lưu lượng truy cập website, số lượng người dùng sử dụng ứng dụng.

Phân tích dữ liệu dòng thời giúp hiểu đặc điểm của tệp dữ liệu, sự thay đổi của tệp dữ liệu theo thời gian. Nó cũng giúp xác định được những yếu tố ảnh hưởng đến các biến tại các thời điểm khác nhau. Một trong những ứng dụng quan trọng của phân tích dữ liệu dòng thời gian là dự đoán giá trị tương lai của các biến trong dòng thời gian dựa vào xu hướng của dữ liệu trong quá khứ.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Cơ sở lý thuyết

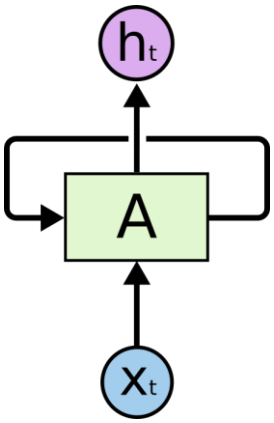
2.1.1 Giới thiệu sơ lược về RNN

Trước khi đi sâu vào tìm hiểu chi tiết về mạng LSTM, chúng ta sẽ giới thiệu sơ qua về mạng RNN.

Suy nghĩ của con người không bắt đầu từ đầu mỗi giây. Khi chúng ta đọc một bài luận, chúng ta hiểu mỗi từ dựa trên kiến thức của mình về những từ trước đó, mà không cần phải bắt đầu suy nghĩ lại từ đầu. Điều này gọi là sự tiếp nối trong suy nghĩ.

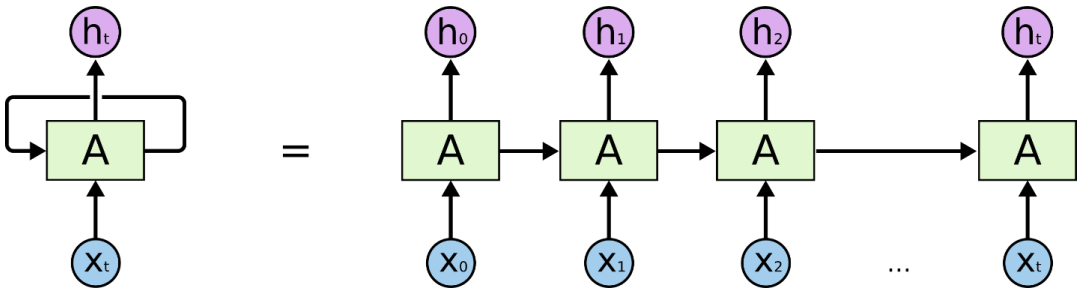
Các mạng neural truyền thống gặp khó khăn khi thực hiện điều này, và đây là một vấn đề lớn. Chẳng hạn, nếu chúng ta cần biết về tất cả các sự kiện đang diễn ra trong một bộ phim tại mọi thời điểm, mạng neural truyền thống khó có thể sử dụng kiến thức về các sự kiện trước đó để dự đoán những sự kiện sau đó.

Mạng RNN được đề xuất để giải quyết vấn đề này. Nó có các vòng lặp bên trong, cho phép thông tin tồn tại lâu dài. Trong mạng RNN, trạng thái ẩn tại mỗi bước thời gian sẽ được xác định dựa vào dữ liệu đầu vào tại bước thời gian hiện tại và các thông tin có được từ bước thời gian trước đó, tạo ra khả năng ghi nhớ các thông tin đã được tính toán ở những bước thời gian trước cho mạng.



Hình 1. Cấu tạo chung của mạng RNN

Hình trên thể hiện một đoạn mạng RNN A, nhận đầu vào x_t và xuất đầu ra một giá trị h_t. Vòng lặp cho phép thông tin được truyền từ bước này sang bước tiếp theo của mạng. Nếu chúng ta khai triển hình trên ra chi tiết hơn, sẽ được hình sau:



Hình 2. Cấu tạo khai triển của mạng RNN

Chúng ta có thể thấy rằng trong hình trên mạng RNN có thể được coi là nhiều bản sao của cùng một mạng, trong đó mỗi đầu ra của mạng bản sao này là đầu vào của một mạng bản sao khác, tương tự như mạng neural truyền thống.

Một trong những điểm nổi bật của RNN là ý tưởng về việc kết nối các thông tin trước đó với sự kiện hiện tại, chẳng hạn như việc sử dụng các cảnh trước đó của một bộ phim có thể giúp hiểu được cảnh hiện tại. Nếu RNN có thể làm được điều này thì chúng sẽ cực kỳ hữu ích, nhưng thực tế thì lại là một vấn đề khác.

Đôi khi, chúng ta chỉ cần xem thông tin gần đây để thực hiện nhiệm vụ hiện tại. Ví dụ: Nếu chúng ta thử đoán từ cuối cùng trong câu "Những đám mây ở trên …", thì chúng ta không cần thêm ngữ cảnh nào nữa - khá rõ ràng từ tiếp theo sẽ là "bầu trời". Trong những trường hợp như vậy, khi khoảng cách giữa thông tin liên quan và nơi cần thông tin đó là nhỏ, RNN có thể học cách sử dụng thông tin trong quá khứ.

Nhưng cũng có những trường hợp chúng ta cần thêm ngữ cảnh, chẳng hạn ta thử đoán từ cuối cùng trong câu "Tôi có sở thích tìm hiểu về lĩnh vực trí tuệ nhân tạo, nên gần đây tôi đã học rất nhiều ngôn ngữ lập trình, đặc biệt là ngôn ngữ …". Thông tin gần nhất cho thấy từ tiếp theo có lẽ là tên của một ngôn ngữ lập trình, nhưng nếu muốn thu hẹp ngôn ngữ lập trình nào, chúng ta cần thông tin của ngữ cảnh "lĩnh vực trí tuệ nhân tạo", vốn đến từ khoảng cách xa hơn. Đây là một ví dụ cho việc xảy ra trường hợp khoảng cách giữa thông tin liên quan và điểm cần thiết sẽ trở nên rất lớn. Và khi khoảng cách đó ngày càng lớn, RNN không thể học cách kết nối thông tin được nữa.

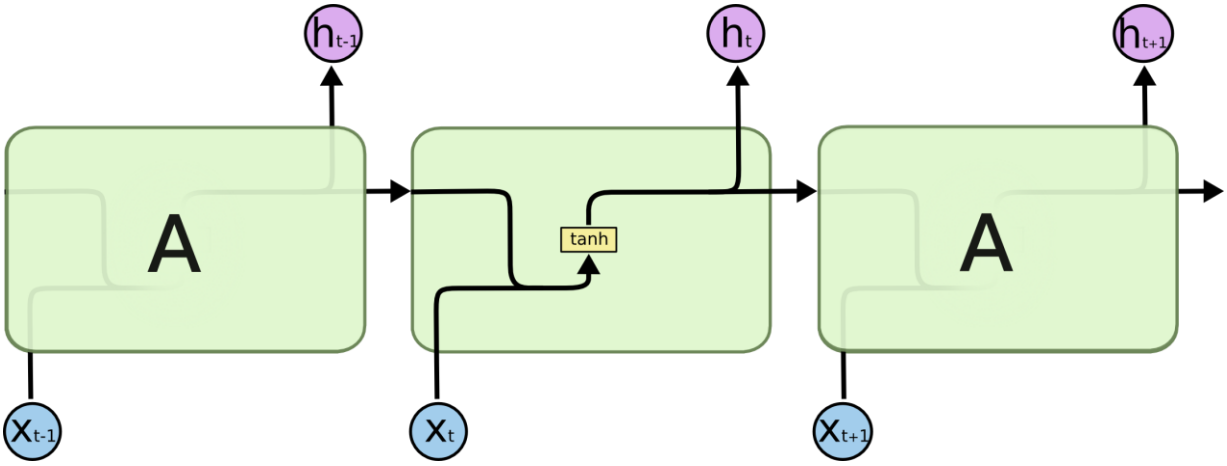
Về lý thuyết, RNN hoàn toàn có khả năng xử lý sự "phụ thuộc xa" (long-term dependencies) như vậy. Chúng ta có thể khéo léo chọn các tham số để giải các bài toán thuộc dạng này. Tuy nhiên trên thực tế, RNN dường như không thể học được các tham số này. Do vậy, mạng LSTM đã được ra đời nhằm giải quyết vấn đề trên.

2.1.2 Giới thiệu mạng LSTM

Mạng LSTM là một loại mạng neural học sâu được thiết kế cải tiến từ mạng RNN để khắc phục nhược điểm về phụ thuộc xa của mạng RNN truyền thống. Được xây dựng

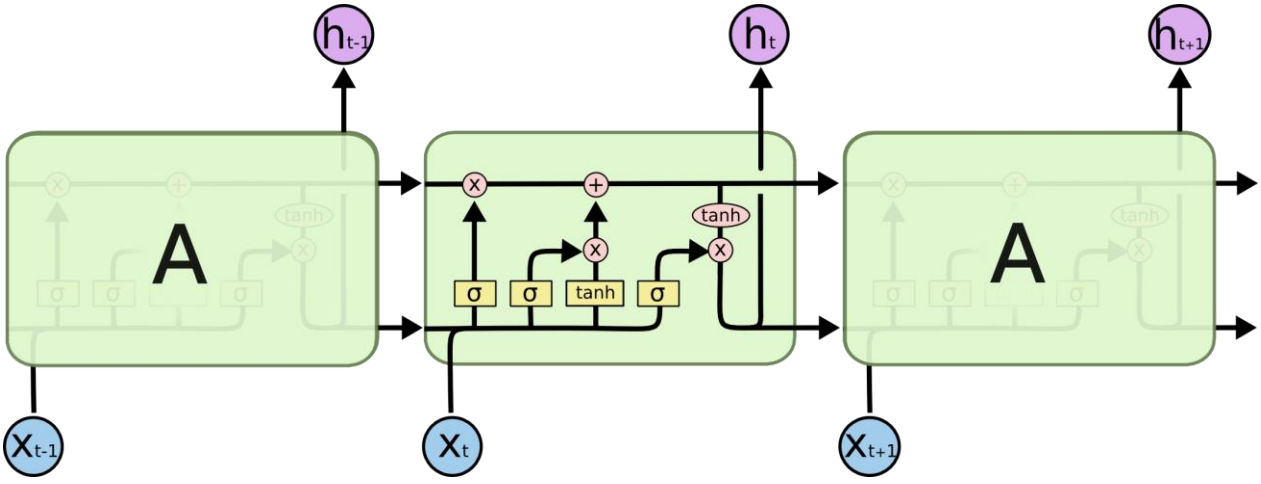
trên ý tưởng của mạng RNN, LSTM giữ được thông tin qua thời gian một cách hiệu quả, làm cho nó trở thành lựa chọn phổ biến cho dự đoán dữ liệu dòng thời gian.

Tất cả các mạng nơ-ron hồi quy đều có dạng là một chuỗi các module lặp của mạng neural. Trong các mạng RNN tiêu chuẩn, module lặp này sẽ có cấu trúc rất đơn giản, thường là một lớp tanh đơn.



Hình 3. Module lặp của mạng RNN, có một lớp

LSTM cũng có cấu trúc giống chuỗi này, nhưng module lặp có cấu trúc khác. Thay vì chỉ có một lớp mạng neural duy nhất thì nó có bốn lớp tương tác với nhau

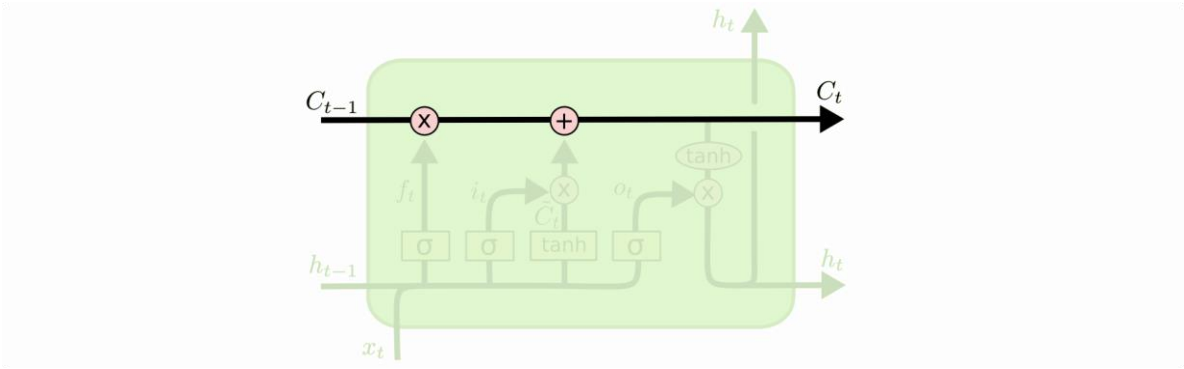


Hình 4. Module lặp của mạng LSTM, có bốn lớp

Ý tưởng chính trong kiến trúc LSTM là tìm hiểu khi nào nên nhớ và khi nào nên quên thông tin thích hợp mà không phải ghi nhớ tất cả thông tin tại mọi thời điểm. Nói cách khác, mạng tìm hiểu một cách hiệu quả thông tin nào có thể cần thiết sau này theo trình tự và khi nào thông tin đó không còn cần thiết nữa.

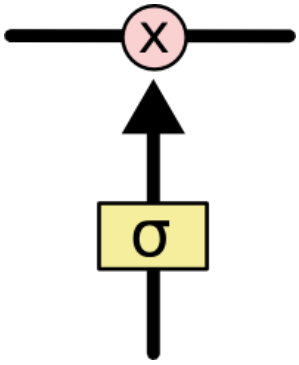
Chìa khóa của LSTM là trạng thái ô, đường ngang chạy qua phía trên của Hình 4. Trạng thái ô giống như một băng chuyền, nó chạy thẳng xuống toàn bộ chuỗi và chỉ

đi qua một số tương tác tuyến tính nhỏ. Vì vậy mà thông tin có thể dễ dàng di chuyển theo nó mà không bị thay đổi.



Hình 5. Trạng thái ô

LSTM có khả năng loại bỏ hoặc thêm thông tin vào trạng thái ô, được điều chỉnh cẩn thận bởi các cấu trúc được gọi là cổng (gate). Cổng là nơi sàng lọc thông tin đi qua nó. Chúng bao gồm một lớp sigmoid và một phép toán nhân.

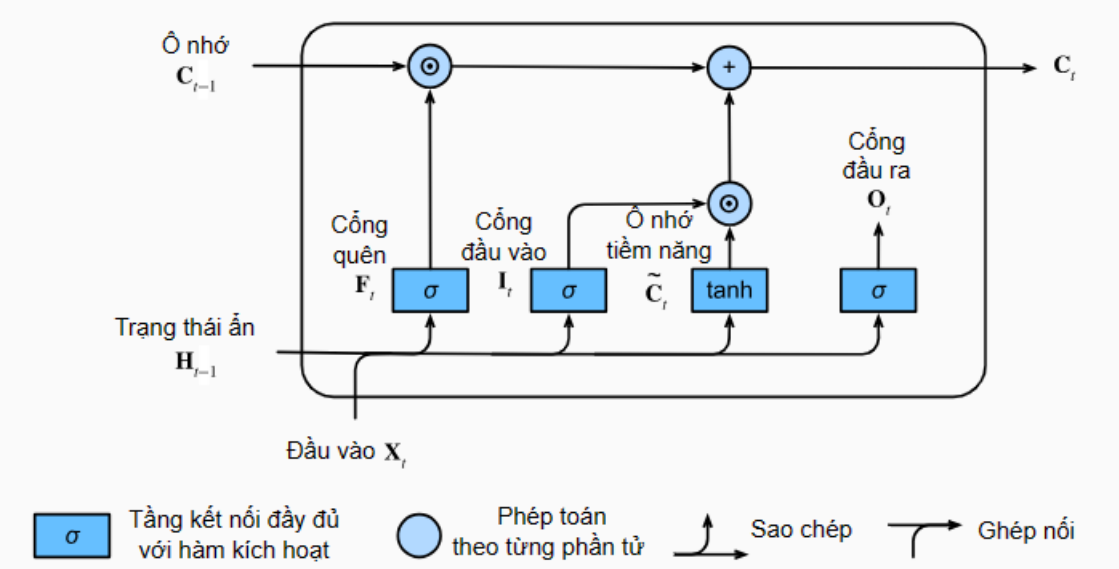


Hình 6. Cấu trúc một cổng

Lớp sigmoid xuất ra một giá trị từ 0 đến 1, mô tả lượng thông tin được cho phép đi qua. Giá trị bằng 0 có nghĩa là "không cho thông tin nào đi qua", trong khi giá trị bằng 1 có nghĩa là "cho mọi thông tin đi qua". LSTM có ba cổng này để duy trì và kiểm soát trạng thái ô.

2.1.3 Cấu tạo mạng LSTM

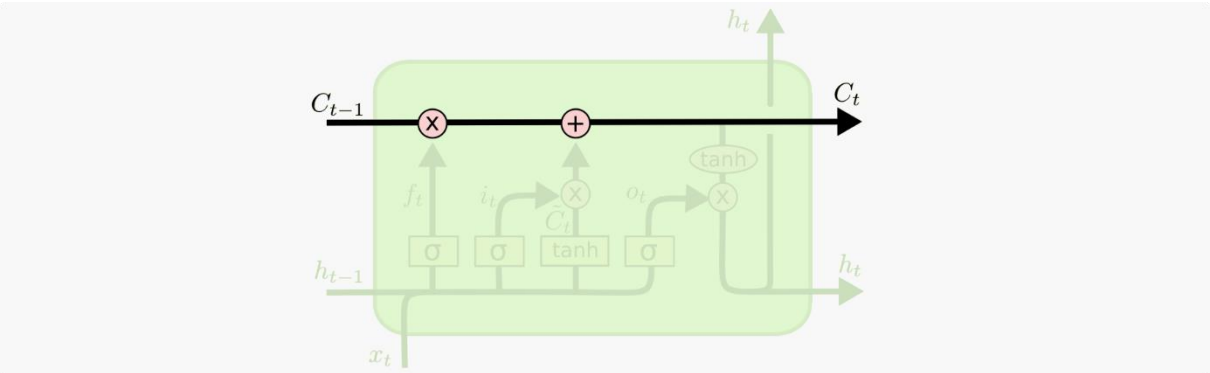
Mạng LSTM có cấu tạo gồm các ô nhớ (memory cell) có chứa các cổng (gate) để điều khiển luồng thông tin. Các cổng bao gồm cổng đầu vào (input gate), cổng quên (forget gate) và cổng đầu ra (output gate), giúp mạng LSTM có thể lưu trữ, quên hoặc truy xuất các thông tin quan trọng trong quá trình học.



Hình 7. Cấu tạo chung của mạng LSTM

a) Ô nhớ

Ô nhớ là thành phần quan trọng nhất của mạng LSTM, nó chịu trách nhiệm lưu trữ và cập nhật thông tin trạng thái ẩn của mạng theo thời gian. Ô nhớ có thể được coi như một bộ nhớ tạm thời, giúp mạng giữ lại thông tin quan trọng và loại bỏ thông tin không cần thiết. Ô nhớ được ký hiệu là C_t ở thời điểm t.



Hình 8. Ô nhớ

Ô nhớ được điều khiển bởi ba cổng là cổng quên, cổng đầu vào và cổng đầu ra. Mỗi cổng là một lớp neural có hàm kích hoạt là hàm sigmoid, cho ra một giá trị trong khoảng từ 0 đến 1, biểu thị mức độ quan trọng của thông tin tương ứng.

- Cổng quên quyết định thông tin nào sẽ được giữ lại và thông tin nào sẽ bị loại bỏ từ ô nhớ trước đó.
- Cổng đầu vào quyết định thông tin mới nào sẽ được thêm vào ô nhớ.
- Cổng đầu ra quyết định thông tin nào sẽ được xuất ra từ ô nhớ.

b) Cổng quên

Cổng quên quyết định thông tin nào trong ô nhớ sẽ được giữ lại hoặc bỏ đi. Nó nhận đầu vào là trạng thái ẩn trước đó h_{t-1}và dữ liệu đầu vào hiện tại x_t, và tính toán một vector f_t có cùng kích thước với ô nhớ. Mỗi phần tử của vector có giá trị trong khoảng từ 0 đến 1, với giá trị 0 có nghĩa là quên hoàn toàn và ngược lại, giá trị 1 có nghĩa là giữ lại hoàn toàn. Công thức tính cổng quên như sau:

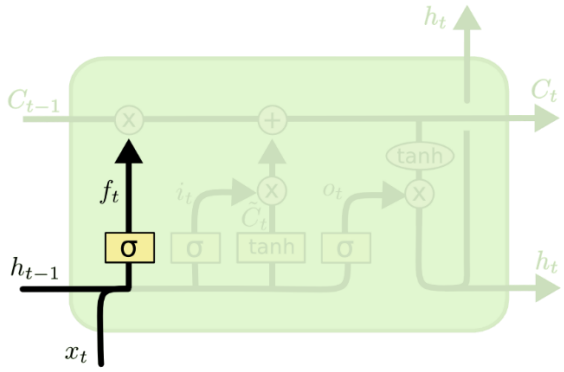
$$f_t = \sigma(W_f \cdot [h_{t-1}, w_t] + b_f)$$

Trong đó:

- f_t là giá trị của cổng quên tại thời điểm t.
- σ là hàm sigmoid
- W_f là ma trận trọng số cho cổng quên.
- h_{t-1} là trạng thái ẩn tại thời điểm t−1.
- w_t là đầu vào tại thời điểm t.
- b_f là độ lệch cho cổng quên.

W_f và b_f là các tham số cần được học trong quá trình huấn luyện.

Vector f_t sau đó được nhân với trạng thái ô nhớ trước đó c_{t-1} để lọc ra những thông tin không cần thiết.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Hình 9. Cổng quên

c) Cổng đầu vào

Cổng đầu vào giữ chức năng quyết định thông tin nào sẽ được giữ lại và thông tin nào sẽ bị loại bỏ từ ô nhớ trước đó.

Nó bao gồm hai phần: phần thứ nhất tính toán vector i_t giống như cổng quên, để xác định mức độ cập nhật của mỗi phần tử trong ô nhớ, phần này sử dụng hàm sigmoid để quyết định thông tin nào được cập nhật, có giá trị trong khoảng từ 0 đến 1; phần thứ hai tính toán vector $\tilde{C}_t$ có cùng kích thước với ô nhớ, để tạo ra trạng thái ô nhớ tiềm năng

cho việc cập nhật ô nhớ. Vector này có kích hoạt là hàm tanh, cho ra các giá trị trong khoảng từ -1 đến 1. Các công thức tính cổng đầu vào như sau:

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

Trong đó:

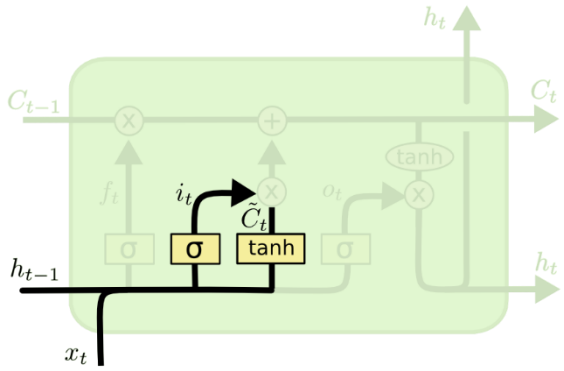
- i_t là giá trị của cổng đầu vào tại thời điểm t.
- σ là hàm sigmoid.
- W_i là ma trận trọng số cho cổng đầu vào.
- h_{t-1} là trạng thái ẩn tại thời điểm t−1.
- x_t là đầu vào tại thời điểm t.
- b_i là độ lệch cho cổng đầu vào.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Trong đó:

- $\tilde{C}_t$ là trạng thái ô nhớ tiềm năng tại thời điểm t.
- W_c là ma trận trọng số cho trạng thái ô nhớ tiềm năng.
- h_{t-1} và x_t giống như công thức trên.
- b_C là độ lệch cho trạng thái ô nhớ tiềm năng.

W_i, b_i, W_C, b_C cũng là các tham số cần được học trong quá trình huấn luyện.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Hình 10. Cổng đầu vào, tính toán trạng thái ô nhớ tiềm năng

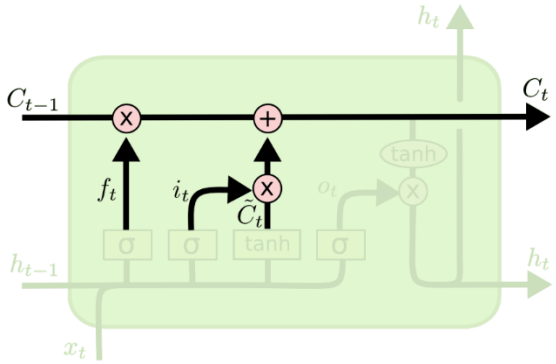
Phép nhân giữa các vector này với trạng thái ô nhớ tiềm năng (theo từng phần tử) sẽ cho ra một cập nhật cho trạng thái ô nhớ hiện tại, xác định thông tin nào sẽ được giữ lại và thông tin nào sẽ bị loại bỏ:

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

Trong đó:

- C_t là trạng thái ô nhớ tại thời điểm t.
- f_t là giá trị của cổng quên tại thời điểm t.
- C_{t-1} là trạng thái ô nhớ tại thời điểm t−1.

- i_t là giá trị của cổng đầu vào tại thời điểm t.
- $\tilde{C}_t$ là trạng thái ô nhớ tiềm năng tại thời điểm t.



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

Hình 11. Cổng đầu vào, quyết định giữ lại hoặc loại bỏ thông tin

d) Cổng đầu ra

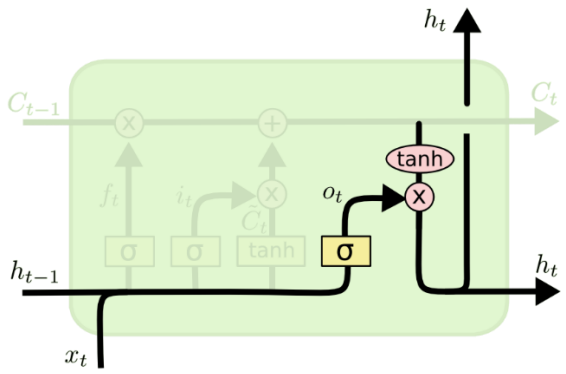
Cổng đầu ra chịu trách nhiệm quyết định thông tin nào sẽ được truyền ra ngoài từ ô nhớ. Cổng này được tính toán bằng cách sử dụng hàm sigmoid, nhận đầu vào là trạng thái ẩn hiện tại và đầu vào hiện tại để tính toán vector o_t tương tự như cổng quên và cổng đầu vào. Sau đó, ô nhớ sẽ được kích hoạt bởi hàm tanh, để tạo ra một vector C_t có giá trị trong khoảng từ -1 đến 1. Phép nhân giữa vector này với vector o_t sẽ cho ra được trạng thái ẩn h_t, được sử dụng làm đầu vào cho các lớp tiếp theo của mạng LSTM hoặc làm đầu ra cuối cùng y_t của mạng. Các công thức tính cổng đầu ra như sau:

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

$$h_t = o_t * tanh(C_t)$$

$$y_t = h_t \text{ (nếu t là thời điểm cuối cùng trong mạng)}$$

- o_t là giá trị của cổng đầu ra tại thời điểm t
- σ là hàm sigmoid.
- W_o là ma trận trọng số cho cổng đầu ra.
- h_{t-1} là trạng thái ẩn tại thời điểm t−1.
- x_t là đầu vào tại thời điểm t.
- b_o là độ lệch cho cổng đầu ra
- h_t là trạng thái ẩn tại thời điểm t.
- y_t là đầu ra cuối cùng tại thời điểm t.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Hình 12. Cổng đầu ra

2.1.4 Cơ chế hoạt động của mạng LSTM

Mạng LSTM hoạt động dựa trên cách ô nhớ duy trì thông tin qua nhiều chu kỳ thời gian và cách các cổng quyết định xem thông tin nào sẽ được truyền tiếp và xử lý. Qua các cơ chế này, mạng tránh được hiện tượng biến mất đạo hàm và bùng nổ đạo hàm, những vấn đề thường gặp khi huấn luyện các mạng RNN truyền thống.

Mạng LSTM còn có khả năng học mối quan hệ phụ thuộc xa trong dữ liệu dòng thời gian bằng cách điều khiển quá trình quên, thêm, và đưa ra thông tin một cách linh hoạt và hiệu quả. Nhờ vào những tính năng này, mạng LSTM trở thành một công cụ mạnh mẽ cho việc xử lý và dự báo dữ liệu vượt trội so với các mô hình truyền thống khác.

2.1.5 Phương pháp huấn luyện mạng LSTM

Quá trình huấn luyện mạng LSTM thường sử dụng hàm mất mát (loss function) và thuật toán tối ưu hóa (optimization algorithm). Hàm mất mát đo lường sự khác biệt giữa giá trị dự đoán và giá trị thực tế của dữ liệu. Thuật toán tối ưu hóa cập nhật các trọng số của mạng LSTM để giảm thiểu hàm mất mát.

Hàm mất mát phổ biến trong việc huấn luyện mạng là hàm sai số bình phương trung bình (mean squared error - MSE), được định nghĩa như sau:

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

Trong đó:

- N là số lượng điểm dữ liệu huấn luyện hoặc kiểm tra
- y_i là giá trị thực tế của mẫu thứ i
- $\hat{y_i}$ là giá trị dự đoán của mẫu thứ i

Thuật toán tối ưu hóa phổ biến cho việc huấn luyện mạng LSTM là thuật toán lan truyền ngược qua thời gian (Backpropagation Through Time – BPTT). Điều này

giúp mạng LSTM học được cấu trúc và xu hướng trong dữ liệu huấn luyện, từ đó có thể dự đoán tốt trên dữ liệu mới. Thuật toán BPTT tính toán đạo hàm riêng bằng cách lan truyền ngược lỗi từ tầng đầu ra qua các tầng ẩn từ cuối dòng thời gian về đầu chuỗi, giúp cập nhật trọng số của mạng dựa trên lỗi tại mỗi bước thời gian, từ đó giúp mạng "học" từ lỗi và cải thiện hiệu suất dự đoán.

2.2 Phương pháp nghiên cứu

2.2.1 Dữ liệu sử dụng trong đồ án

Đồ án này sử dụng tập dữ liệu về chất lượng không khí ngoài trời được thu thập từ mạng lưới giám sát chất lượng không khí gồm sáu trạm quan trắc chất lượng không khí trên khắp Thành phố Hồ Chí Minh, Việt Nam.

Dữ liệu thô chứa 52.549 bản ghi được thu thập trong khoảng thời gian từ giữa tháng 2 năm 2021 đến giữa tháng 6 năm 2022. Tập dữ liệu về chất lượng không khí bao gồm số liệu về bụi mịn PM2.5, tổng số lượng bụi lơ lửng (Total Suspended Particulates – TSP), Sulphur dioxide (SO2), Ozone (O3), Nitrogen Dioxide (NO2), Carbon Monoxide (CO) (đơn vị $\mu g/m^3$) và hai thông số khí tượng là nhiệt độ (đơn vị °C) và độ ẩm (đơn vị %). Bộ dữ liệu này có thể được sử dụng để lập mô hình chất lượng không khí, phân tích không gian, thời gian và đánh giá chất lượng không khí trên các khu vực khác nhau như các khu vực giao thông, dân cư và công nghiệp trên toàn thành phố.

Tuy nhiên, đồ án này chỉ trích xuất số liệu về bụi mịn PM2.5 trên trạm số 1 làm tập dữ liệu để đáp ứng với phạm vi của đồ án là dữ liệu đơn biến. Do đó, biến bụi mịn PM2.5 cũng đồng thời là biến mục tiêu để dự đoán. Tập dữ liệu này sẽ có 7892 bản ghi tương ứng với 7892 giá trị bụi mịn PM2.5. Tiếp theo, chúng tôi chia tập dữ liệu mới này thành hai phần: tập huấn luyện và tập kiểm tra với tỷ lệ 70:30. Cuối cùng dữ liệu được tiền xử lý bằng cách chuẩn hóa về khoảng [0, 1] để tăng hiệu quả của mô hình.

2.2.2 Xây dựng mô hình mạng LSTM để dự đoán dữ liệu

Đồ án này sử dụng kiến trúc mạng LSTM sau:

- Lớp LSTM đầu tiên: Lớp này chứa 50 đơn vị LSTM và trả về chuỗi đầu ra đầy đủ. Đầu vào cho lớp này có kích thước tương ứng với số lượng bước thời gian và số lượng đặc trưng trong dữ liệu dòng thời gian của bạn.
- Lớp Dense: Lớp này chứa một đơn vị Dense và được sử dụng để dự đoán giá trị tiếp theo trong dòng thời gian.
- Hàm kích hoạt: hàm ReLU

- Thuật toán tối ưu hóa: Adam
- Hàm mất mát: sai số bình phương trung bình
- Các thông số cho quá trình huấn luyện: epochs = 50, batch_size = 60

2.2.3 Đánh giá mô hình

Các chỉ số được sử dụng để đánh giá mô hình:

- Sai số phần trăm tuyệt đối trung bình (MAPE – Mean absolute percentage error): được tính như sau:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \widehat{y_i}}{y_i}\right|$$

Giá trị MAPE càng thấp thì mô hình dự đoán càng tốt.

- RMSE: là căn bậc hai của sai số bình phương trung bình (MSE), được tính như sau:

$$RMSE = \sqrt{\text{MSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}$$

Giá trị RMSE càng nhỏ thì mô hình càng phù hợp với dữ liệu.

Trong các công thức trên:

- N là số lượng điểm dữ liệu huấn luyện hoặc kiểm tra
- y_i là giá trị thực tế của mẫu thứ i
- $\widehat{y_i}$ là giá trị dự đoán của mẫu thứ i

CHƯƠNG 3: ĐÁNH GIÁ KẾT QUẢ

3.1 Môi trường cài đặt

Để xây dựng và đánh giá mô hình mạng LSTM, đồ án đã sử dụng một trong những ngôn ngữ phổ biến nhất trong lĩnh vực học máy. Python hỗ trợ nhiều thư viện quan trọng giúp thực hiện các công việc nghiên cứu; cụ thể đồ án đã sử dụng Keras để định nghĩa kiến trúc mô hình, huấn luyện mô hình, và lưu trữ, tải dữ liệu. Đồng thời, chúng tôi cũng sử dụng các công cụ khác như Pandas và Numpy để tiền xử lý dữ liệu đầu vào và phân tích kết quả đầu ra.

Để tạo môi trường chứa và chạy mã nguồn mô hình, chúng tôi đã lựa chọn Google Colab. Google Colab là một dịch vụ miễn phí của Google, cho phép chúng tôi tạo và chia sẻ các notebook Jupyter một cách thuận tiện. Nó giúp người dùng viết và chạy mã Python trực tiếp trên máy chủ của Google mà không cần cài đặt phần mềm phức tạp trên máy tính cá nhân. Môi trường Jupyter Notebook của Google Colab được tích hợp tốt với nhiều thư viện và công cụ, đặc biệt là TensorFlow, Keras, và PyTorch, giúp chúng tôi dễ dàng thực hiện các nhiệm vụ phức tạp như huấn luyện mô hình deep learning trên dữ liệu lớn mà không mất thời gian cho việc cài đặt và cấu hình môi trường. Bên cạnh đó, nó cũng cho phép tích hợp với Google Drive giúp lưu trữ và chia sẻ notebook trở nên đơn giản.

3.2 Kết quả nghiên cứu

Điều chỉnh số bước nhảy trong dữ liệu huấn luyện và dữ liệu kiểm thử được các kết quả như sau:

Bảng 1. Các chỉ số đánh giá sau khi huấn luyện

Số bước nhảy	MAPE		RMSE	
	Tập huấn luyện	Tập kiểm thử	Tập huấn luyện	Tập kiểm thử
3	15442592836994.90	0.35	7.30	5.44
6	14813734689815.56	0.35	7.26	5.48
10	17523712317078.44	0.39	7.22	5.55
12	18666286301826.24	0.33	7.19	5.40
15	22403049157093.71	0.37	7.17	5.51

Dựa vào kết quả trên, có thể thấy được khi số bước nhảy tăng, các giá trị MAPE và RMSE đều giảm xuống, chứng tỏ mô hình học tốt khi dữ liệu đầu vào tăng. Nhìn chung ở bước nhảy là 10, thì mô hình là tối ưu nhất.

Giá trị RMSE cho cả tập huấn luyện và tập kiểm thử đều nằm trong khoảng từ 5 đến 8, cho thấy mô hình có độ chính xác tương đối.

Tuy nhiên, giá trị MAPE cho tập huấn luyện rất lớn so với tập kiểm thử ở tất cả các bước nhảy, có nghĩa là mô hình có khả năng đã bị quá khớp trên tập dữ liệu huấn luyện.

CHƯƠNG 4: KẾT LUẬN

Đồ án đã thực hiện giới thiệu về mạng LSTM và xây dựng và đánh giá mô hình mạng LSTM cho việc dự đoán dữ liệu dòng thời gian.

Dựa trên kết quả huấn luyện và kiểm thử mô hình LSTM, chúng ta có thể thấy rằng mô hình đã học được một số mẫu từ dữ liệu. Tuy nhiên, có sự khác biệt lớn giữa giá trị MAPE của tập huấn luyện và tập kiểm thử, cho thấy mô hình có thể chưa được tối ưu hoá đầy đủ, cần phải cải thiện hiệu suất thông qua các nỗ lực tối ưu hóa và fine-tuning . Mặc dù vậy, mô hình vẫn cho thấy độ chính xác tương đối với giá trị RMSE từ 5 đến 8 cho cả hai tập.

CHƯƠNG 5: HƯỚNG PHÁT TRIỂN

Một số hướng phát triển tiềm năng có thể xem xét tiếp theo để nâng cao hiệu suất của mô hình:

- Tối ưu hóa mô hình: Tìm hiểu và áp dụng các kỹ thuật tối ưu hóa mô hình LSTM như thay đổi kiến trúc, tinh chỉnh siêu tham số, các kỹ thuật như điều chỉnh tốc độ học, sử dụng dropout,…
- Sử dụng mô hình phức tạp hơn: Kết hợp mô hình LSTM với các kiến trúc mô hình học sâu khác, chẳng hạn như mạng CNN, để tận dụng các đặc trưng không gian trong dữ liệu.

Ngoài ra có thể mở rộng ứng dụng của mô hình để tích hợp yếu tố bền vững, phân tích tác động xã hội và kinh tế, và áp dụng công nghệ mới.

DANH MỤC TÀI LIỆU THAM KHẢO

1. L. X. Hiền và H. V. Hùng, "Ứng dụng mạng Long Short-Term Memory (LSTM) để dự báo mực nước tại trạm Quang Phục và Cửa Cấm, Hải Phòng, Việt Nam," *Tạp chí Khoa học Kỹ thuật Thủy lợi và Môi trường*, số 62, trang 10-11, 2018.
2. H. V. Thông và N. V. Kiên, "Thiết kế mạng học sâu Long Short-Term Memory (LSTM) để dự báo lưu lượng và phát hiện bất thường trong mạng cấp nước sạch," *Tạp chí Khoa học & Công Nghệ ĐHTN*, tập 225, số 14, trang 135-137, 2020.
3. N. V. Hưng, V. H. Nam, V. Đ. Anh, T. Q. Hiệp, và L. T. Long, "Dự đoán giá trị cảm biến chất lượng không khí sử dụng mạng nơ ron tích chập một chiều và mạng bộ nhớ dài ngắn hạn," *Tạp chí Khoa học Công nghệ Thông tin và Truyền thông*, tập 04, số 01, trang 130-132, 2021.
4. D. T. Hà và N. T. Nghe, "Ứng dụng mô hình đa biến bộ nhớ dài – ngắn hạn trong dự báo nhiệt độ và lượng mưa," *Tạp chí Khoa học Trường Đại học Cần Thơ*, tập 58, số 4A, trang 10-13, 2022.
5. N. T. Tuan, T. H. Nguyen, và T. T. H. Duong, "Stock Price Prediction in Vietnam using Stacked LSTM," trong *Hội nghị Quốc tế về Vạn vật Thông minh (ICIT)*, 2022.
6. N. H. Ly và H. T. T. Hà, "Kết hợp mô hình học máy và mô hình thống kê trong dự báo dòng thời gian: Trường hợp lạm phát tại Việt Nam giai đoạn 2000 – 2021," *Tạp chí Khoa học Kinh tế*, tập 10, số 01, trang 20-22, 2022.
7. C. Olah, "Understanding LSTM Networks," *GitHub Blog*, 2015. http://colah.github.io/posts/2015-08-Understanding-LSTMs/.
8. P. Đ. Khanh, "Lý thuyết về mạng LSTM," *GitHub Blog*, 2019. https://phamdinhkhanh.github.io/2019/04/22/Ly_thuyet_ve_mang_LSTM.html.
9. R. Rajnish, L. Quan, H. Bang, V. Khue, và S. Ricardo, "The HelthyAir Dataset: Outdoor Air Quality in Ho Chi Minh City, Vietnam," *Mendeley Data*, tập 1, 2022. doi: 10.17632/pk6tzrjks8.1.
10. "Bài 16: Dự đoán chuỗi thời gian bằng LSTM RNN - VNCoder.vn." VNCoder.vn
11. "Sử dụng mô hình bộ nhớ ngắn hạn dài hạn (LSTM) của Keras để dự đoán giá cổ phiếu," ICHI.PRO, 2023.
12. "What is LSTM," dominhhai.github.io, 2017.
13. "Dự đoán dữ liệu dạng chuỗi sử dụng mạng thần kinh LSTM," Zun.vn.

14. "Machine Learning - Thử làm Nhà Thiên Văn Dự Báo Thời Tiết," Viblo, 29 tháng 1 năm 2018.
15. "LSTM in forecasting," Academia.edu.
16. "How to Develop LSTM Models for Time Series Forecasting," Machine Learning Mastery.
17. "Sử dụng mạng LSTM (Long Short Term Memory) để dự đoán số liệu hướng thời gian," Trituevietvn.com.
18. "Bài 14: Long short term memory (LSTM)," nttuan8.com, 02 tháng 6 năm 2019.
19. "A complete guide to understand Long Short Term Memory (LSTM) networks," Sefidian.com, 15 tháng 8 năm 2019.
20. "LSTM," D2l.ai.
21. "Recurrent Neural Network: Từ RNN đến LSTM," Viblo.
22. "Giải thích chi tiết về mạng Long Short Term Memory (LSTM)," Nguyentruonglong.net.
23. "LSTM model," LinkedIn, Armando Aguilar Lopez.