

ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN - TIN



ĐỒ ÁN II

PHÂN TÍCH GIÁ CHỨNG KHOÁN
BẰNG MÔ HÌNH LSTM

Chuyên ngành: Toán ứng dụng

Giảng viên hướng dẫn: ThS. Lê Xuân Lý

Sinh viên thực hiện: Vũ Thành Đạt

MSSV: 20206275

Lớp: Hệ thống thông tin 01-K65

HÀ NỘI, 01/2024

NHẬN XÉT CỦA GIẢNG VIÊN

1. Mục tiêu

(a)

(b)

(c)

2. Nội dung

(a)

(b)

(c)

3. Đánh giá kết quả đạt được

(a)

(b)

(c)

Hà Nội, ngày ... tháng ... năm 2024

Giảng viên hướng dẫn

ThS. LÊ XUÂN LÝ

Lời cảm ơn

Báo cáo này được thực hiện và hoàn thành tại Đại học Bách Khoa Hà Nội, nằm trong nội dung học phần Đồ án II của kì học 2023-1.

Em xin được gửi lời cảm ơn chân thành tới ThS. Lê Xuân Lý, là giảng viên đã trực tiếp hướng dẫn và gợi ý cho em đề tài rất thú vị này, đồng thời thầy cũng đã giúp đỡ tận tình và có những góp ý, định hướng bổ ích .

Hà Nội, ngày ... tháng ... năm 2024

Tác giả đồ án

Vũ Thành Đạt

Lời mở đầu

Trong thời đại công nghệ hiện đại bùng nổ, đặc biệt với sự ra đời của trí tuệ nhân tạo đang trở nên phổ biến hơn bao giờ hết. Sự ra đời của máy học, trí tuệ nhân tạo, mạng nơ-ron và các công nghệ học tập hiện đại đã giúp cuộc sống con người trở nên ngày càng dễ dàng.

Nhưng để có thể từ một mô hình trí tuệ nhân tạo, có thể đưa ra đánh giá, dự báo không phải đơn giản. Do đó, em quyết định nghiên cứu, tìm hiểu về mô hình Long Short Term Memory dự báo giá chứng khoán. Từ đó, đặt nền móng cơ bản về các mô hình học tập như Trí tuệ nhân tạo, máy học, học sâu, mạng nơ-ron và nghiên cứu sâu hơn trong tương lai. Bài báo cáo này trình bày tổng quan hiểu biết của em về mô hình Long Short Term Memory (LSTM) và kết quả nghiên cứu xây dựng mô hình LSTM dự đoán chỉ số giá chứng khoán VN-index.

Cấu trúc của báo cáo

Báo cáo đề án II này sẽ được trình bày gồm các chương như sau:

- **Chương 1: Đặt vấn đề và định hướng giải pháp:** Chương này giới thiệu chung về bài toán phân tích dự đoán giá chứng khoán dựa trên mô hình mạng nơ-ron nhân tạo Long Short Term Memory và nhiệm vụ cụ thể của đề án.
- **Chương 2: Mô hình Long Short Term Memory:** Trình bày tổng quan về các khái niệm chứng khoán và các lý thuyết của Trí tuệ nhân tạo, Máy học, Học sâu, mạng nơ-ron liên quan đến mô hình LSTM. Sau đó đi vào chi tiết mô hình LSTM.
- **Chương 3: Thử nghiệm số:** Trình bày quá trình xây dựng mô hình bằng ngôn ngữ Python và đánh giá kết quả thu được.

Quy tắc viết báo cáo và ký hiệu

Trong báo cáo này có sử dụng một số tên viết tắt cho các thuật ngữ và ký hiệu như sau:

LSTM	Long Short Term Memory
σ	Hàm sigmoid
tanh	Hàm tanh
b	Sai số
∂L	Đạo hàm riêng của hàm L
W	Ma trận trọng số
g_t	Gradient tại bước thời gian t
\mathbb{R}	Tập hợp các số thực
$\mathbb{R}^{m \times n}$	Tập hợp các ma trận có m hàng, n cột

Mục lục

Lời Cảm Ơn	i
Mở đầu	ii
Chương 1. Đặt vấn đề và định hướng giải pháp	2
1.1 Tổng quan bài toán	2
1.2 Mục tiêu và giới hạn bài toán	2
Chương 2. Mô hình Long Short Term Memory	4
2.1 Một số khái niệm cơ bản về chứng khoán	4
2.1.1 Khái niệm	4
2.1.2 Thị trường chứng khoán	5
2.2 Giới thiệu về mạng Nơ-ron	6
2.2.1 Trí tuệ nhân tạo, Machine learning, Deep learning	6
2.2.2 Giới thiệu về mạng Nơ-ron	7
2.2.3 Hàm kích hoạt - <i>Activation function</i>	9
2.2.4 Hàm mất mát - <i>Loss function</i>	10
2.2.5 Cập nhật trọng số	11
2.3 Mô hình LSTM - <i>Long Short Term Memory</i>	12
2.3.1 Ý tưởng mô hình LSTM	12
2.3.2 Các thành phần của một đơn vị trong mạng LSTM	14

2.3.3	Quá trình thực hiện mô hình LSTM	15
2.3.4	Huấn luyện mô hình	18
Chương 3. Thử nghiệm số		22
3.1	Tổng quan về bộ dữ liệu	23
3.2	Xây dựng mô hình LSTM	25
3.3	Dự báo kết quả và đánh giá mô hình	28
Tài liệu tham khảo		32

Danh sách hình vẽ

2.1	Quan hệ giữa AI, ML, Deep learning, Neural networks	7
2.2	Mô hình tổng quát của mạng nơ-ron	8
2.3	Hàm sigmoid	9
2.4	Hàm tanh	10
2.5	Mô hình Long Short Term Memory	13
2.6	Các thành phần của mô hình Long Short Term Memory	15
2.7	Một số ký hiệu	15
2.8	Xóa các thông tin không cần thiết.	16
2.9	Quyết định các thông tin mới được thêm vào trạng thái tế bào	17
2.10	Cập nhật trạng thái ô C_t	17
2.11	Điều chỉnh thông tin đầu ra.	18
3.1	Tổng quan về bộ dữ liệu	23
3.2	Giá đóng cửa theo thời gian	24
3.3	Chạy mô hình	26
3.4	Giá trị hàm mất mát giảm dần qua các epoch	26
3.5	Kết quả dự báo của mô hình LSTM	27
3.6	Hiển thị kết quả dự đoán của mô hình	27
3.7	So sánh giá trị dự đoán và giá trị thực tế	28

Chương 1

Đặt vấn đề và định hướng giải pháp

1.1 Tổng quan bài toán

Bài toán dự báo giá chứng khoán là một trong những bài toán phức tạp và thú vị trong lĩnh vực tài chính. Nó liên quan đến việc sử dụng các phương pháp, mô hình và dữ liệu lịch sử để dự đoán giá chứng khoán trong tương lai. Thị trường chứng khoán là một thị trường biến động, đem lại nhiều lợi nhuận, thu hút nhiều nhà đầu tư. Bên cạnh đó, rủi ro khi tham gia thị trường chứng khoán là lớn, dẫn đến mô hình phân tích, dự báo chứng khoán ra đời nhằm hạn chế rủi ro khi đầu tư chứng khoán.

Mục tiêu của bài toán dự báo giá chứng khoán là cung cấp thông tin về xu hướng và biến động giá của một công ty hoặc một tài sản tài chính cụ thể. Dự báo giá chứng khoán có thể được sử dụng để đưa ra quyết định đầu tư, giao dịch chứng khoán và quản lý rủi ro.

Từ đó có thể thấy được xây dựng mô hình dự đoán hỗ trợ đầu tư chứng khoán là một bài toán hữu ích, cũng là bài toán em nghiên cứu trong đề án này.

1.2 Mục tiêu và giới hạn bài toán

Mục tiêu của đề án:

- Tìm hiểu tổng quan về mô hình Long Short Term Memory

- Áp dụng mô hình Long Short Term Memory xây dựng chương trình dự đoán giá cổ phiếu.
- Kiểm tra tính khả thi của mô hình Long Short Term Memory thông qua xây dựng chương trình và thử nghiệm.

Nội dung công việc:

- Nghiên cứu mô hình Long Short Term Memory.
- Ứng dụng mô hình Long Short Term Memory xây dựng chương trình dự báo giá chứng khoán bằng ngôn ngữ Python.

Chương 2

Mô hình Long Short Term Memory

2.1 Một số khái niệm cơ bản về chứng khoán

2.1.1 Khái niệm

Chứng khoán là bằng chứng xác nhận quyền và lợi ích hợp pháp của người sở hữu đối với tài sản hoặc phần vốn của tổ chức phát hành. Chứng khoán được thể hiện dưới hình thức chứng chỉ, bút toán ghi sổ hoặc dữ liệu điện tử. Chứng khoán bao gồm các loại: cổ phiếu, trái phiếu, chứng chỉ quỹ đầu tư, chứng khoán phái sinh. Xem xét chứng khoán có thể thấy rằng chứng khoán đại biểu cho một số tiền nhất định mà người đầu tư đã ứng ra và điều cơ bản là nó đưa lại cho người sở hữu chứng khoán quyền được hưởng những khoản thu nhập nhất định trong tương lai. Vì thế, chứng khoán có thể được mua bán và được lưu thông với tư cách là hàng hoá.

Đặc trưng của chứng khoán

Sự phát triển thị trường chứng khoán làm nảy sinh nhiều loại chứng khoán và rất đa dạng. Tuy nhiên, các loại chứng khoán đều có những đặc trưng sau:

- Chứng khoán luôn gắn với khả năng thu lợi
- Chứng khoán luôn gắn với rủi ro
- Chứng khoán có khả năng thanh khoản

2.1.2 Thị trường chứng khoán

Thị trường chứng khoán trong điều kiện của nền kinh tế hiện đại, được quan niệm là nơi diễn ra các hoạt động giao dịch mua bán chứng khoán trung và dài hạn. Việc mua bán chứng khoán được thực hiện ở thị trường sơ cấp khi người mua mua chứng khoán lần đầu tiên từ những người phát hành và ở thị trường thứ cấp khi có sự mua bán lại các chứng khoán đã được phát hành ở thị trường sơ cấp. Như vậy, xét về mặt hình thức, thị trường chứng khoán chỉ là nơi diễn ra các hoạt động trao đổi, mua bán, chuyển nhượng các loại chứng khoán, qua đó thay đổi chủ thể nắm giữ chứng khoán.

Chức năng cơ bản của thị trường chứng khoán

- Cung cấp môi trường đầu tư cho công chúng
- Tạo tính thanh khoản cho các chứng khoán
- Đánh giá hoạt động của doanh nghiệp
- Tạo môi trường giúp Chính phủ thực hiện các chính sách vĩ mô
- Huy động vốn đầu tư cho nền kinh tế

Chủ thể tham gia thị trường chứng khoán

Các tổ chức và cá nhân tham gia thị trường chứng khoán có thể được chia thành các nhóm: nhà phát hành, nhà đầu tư và các tổ chức có liên quan đến chứng khoán.

- Nhà phát hành: Nhà phát hành là các tổ chức thực hiện huy động vốn thông qua thị trường chứng khoán. Nhà phát hành là người cung cấp các chứng khoán - hàng hoá của thị trường chứng khoán.
- Nhà đầu tư: Nhà đầu tư là những người thực sự mua và bán chứng khoán trên thị trường chứng khoán nhằm mục đích thu lời. Nhà đầu tư được chia thành 2 loại: nhà đầu tư cá nhân và nhà đầu tư có tổ chức.
- Các tổ chức có liên quan đến thị trường chứng khoán: Cơ quan quản lý Nhà nước, Sở Giao dịch Chứng khoán, tổ chức tài trợ chứng khoán, ...

Các thông số giá và khối lượng giao dịch của chứng khoán:

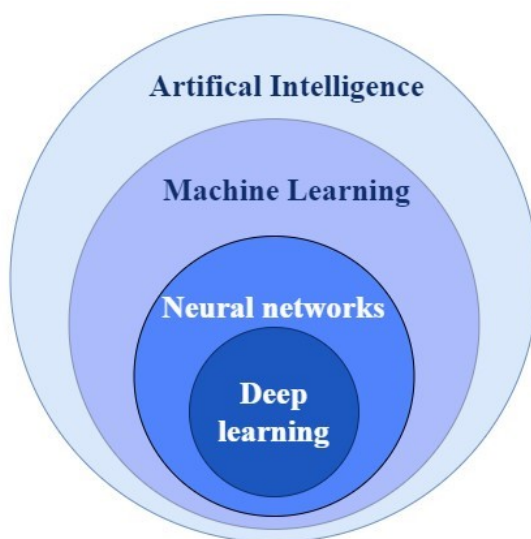
- **Giá mở cửa:** Là mức giá của giao dịch đầu tiên trong kỳ của phiên giao dịch trên thị trường cụ thể. (ví dụ: giao dịch đầu tiên của ngày). Khi phân tích dữ liệu hàng ngày, giá mở cửa có vai trò đặc biệt quan trọng vì đây là giá đã được các bên liên quan đồng thuận.
- **Giá cao nhất:** Là giá cao nhất trong mộ phiên giao dịch hoặc trong một chu kỳ theo dõi biến động giá.
- **Giá thấp nhất:** Là giá cao nhất trong một phiên giao dịch hoặc trong một chu kỳ theo dõi biến động giá..
- **Giá đóng cửa:** Là mức giá thị trường của cổ phiếu vào thời điểm đóng cửa hay kết thúc một phiên giao dịch trên thị trường cụ thể.
- **Khối lượng giao dịch** Là số lượng cổ phần (hay hợp đồng) được giao dịch trong kỳ. Mối liên quan giữa giá và khối lượng giao dịch rất quan trọng (chẳng hạn như giá tăng đi kèm với sự gia tăng của khối lượng).

2.2 Giới thiệu về mạng Nơ-ron

2.2.1 Trí tuệ nhân tạo, Machine learning, Deep learning

Trí tuệ nhân tạo (*Artificial Intelligence - AI*) là lĩnh vực của khoa học máy tính nhằm nghiên cứu và phát triển các hệ thống hoạt động giống hoặc tương tự như con người trong việc thực hiện các tác vụ thông minh. Mục tiêu của trí tuệ nhân tạo là thiết kế và xây dựng các máy tính hoặc hệ thống có khả năng học hỏi, lý thuyết, tư duy và thực hiện các nhiệm vụ mà trước đây chỉ có con người mới có thể làm được.

Học máy (*Machine learning*) là một lĩnh vực con của trí tuệ nhân tạo. Nó tập trung vào việc xây dựng các mô hình và thuật toán để cho máy tính học từ dữ liệu mà không cần được lập trình cụ thể. Máy tính học từ dữ liệu bằng cách phân tích



Hình 2.1: Quan hệ giữa AI, ML, Deep learning, Neural networks

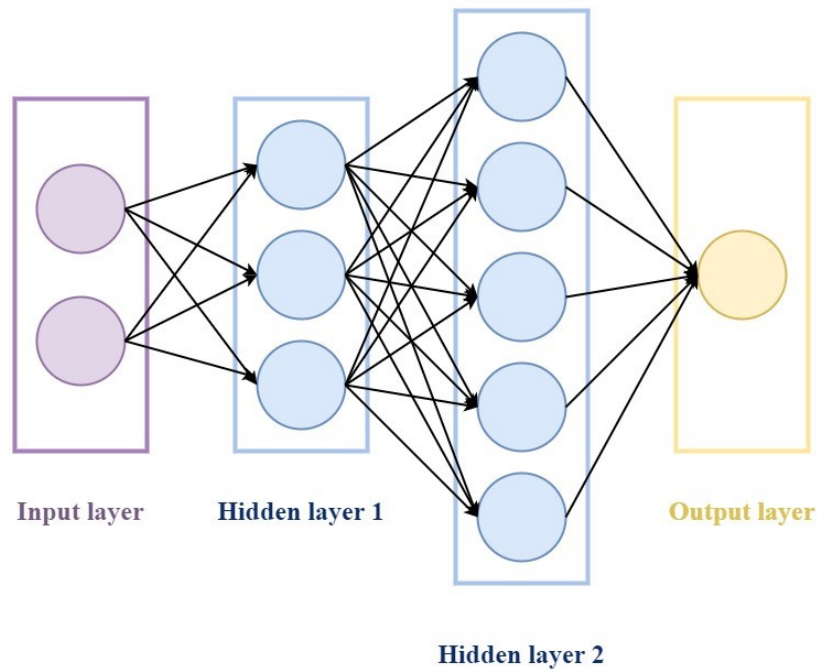
và nhận biết các mẫu, quy luật hoặc thông tin ẩn trong dữ liệu để đưa ra dự đoán, phân loại hoặc tìm ra các giải pháp.

Học sâu (*Deep learning*) là một phương pháp học máy đặc biệt, trong đó mô hình học máy được xây dựng dựa trên mạng nơ-ron nhân tạo sâu có nhiều tầng ẩn. Deep learning chú trọng vào việc tự động học các đặc trưng từ dữ liệu đầu vào thông qua một số lớp nơ-ron ẩn, cho phép nó phát hiện và biểu diễn các mức độ trừu tượng và phức tạp hơn so với các phương pháp truyền thống.

2.2.2 Giới thiệu về mạng Nơ-ron

Mạng nơ-ron là một tập hợp con của học máy và là trung tâm của các thuật toán học sâu. Tên và cấu trúc của chúng được lấy cảm hứng từ bộ não con người, bắt chước cách các tế bào thần kinh sinh học truyền tín hiệu cho nhau. Mạng nơ-ron bao gồm các lớp nút kết nối với các nút khác, có trọng số và các ngưỡng liên quan. Có thể coi một nút là một nơ-ron nhân tạo và đường kết nối giữa các nút là các dây thần kinh. Mạng nơ-ron dựa vào dữ liệu đào tạo để tìm hiểu và cải thiện độ chính xác của chúng theo thời gian. Do đó, khi các thuật toán được tinh chỉnh và đạt độ chính xác cao, mạng nơ-ron sẽ trở thành một công cụ mạnh mẽ trong công nghệ khoa học máy tính hiện đại.

Mô hình tổng quát của mạng nơ-ron



Hình 2.2: Mô hình tổng quát của mạng nơ-ron

- Lớp (*Layer*) đầu tiên là lớp đầu vào (*Input layer*). Lớp này nhận đầu vào của mạng, có thể là các đặc trưng hoặc dữ liệu đầu vào khác.
- Các lớp ở giữa là lớp ẩn (*Hidden layer*). Mỗi lớp ẩn có thể có nhiều nút, và các nút trong cùng một lớp được kết nối với nhau. Số lượng lớp ẩn và số lượng nút trong mỗi lớp là các tham số cần được định nghĩa trước.
- Lớp cuối cùng được gọi là lớp đầu ra (*Output layer*). Lớp này trả về kết quả cuối cùng của mạng nơ-ron.
- Mỗi hình tròn là một nút (*node*). Mỗi nút trong lớp ẩn và lớp đầu ra liên kết với tất cả các nút ở lớp trước đó với các hệ số w riêng và có 1 hệ số độ lệch (*bias*) b .

Mỗi nút trong mạng nơ-ron thực hiện các phép tính trên đầu vào của nó, kết hợp với các trọng số tương ứng. Sau đó, một hàm kích hoạt (*Activation function*) được áp dụng để tính toán đầu ra của nơ-ron. Các trọng số trong mạng nơ-ron được điều chỉnh trong quá trình huấn luyện để tối ưu hóa đầu ra của mạng cho các mục tiêu cụ thể, như giảm thiểu sai số hay tối đa hóa độ chính xác.

2.2.3 Hàm kích hoạt - *Activation function*

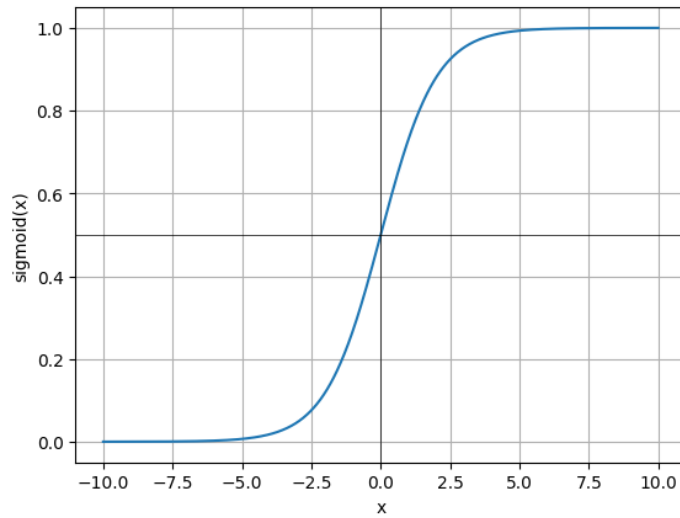
Hàm kích hoạt (*Activation function*) là những hàm phi tuyến được áp dụng vào đầu ra của các nơ-ron trong tầng ẩn của một mô hình mạng. Nếu chỉ áp dụng 1 hàm tuyến tính thì phép biến đổi không có tính chất phi tuyến, sẽ không thể phát hiện ra những quan hệ phức tạp của dữ liệu. Việc kết hợp của các hàm kích hoạt giữa các tầng ẩn để giúp mô hình học được các quan hệ phi tuyến phức tạp trong dữ liệu.

Một số hàm kích hoạt thông dụng:

- Hàm sigmoid $f(x)$ có công thức:

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

Giá trị của hàm $\sigma(x)$ nằm trong khoảng $(0, 1)$.



Hình 2.3: Hàm sigmoid

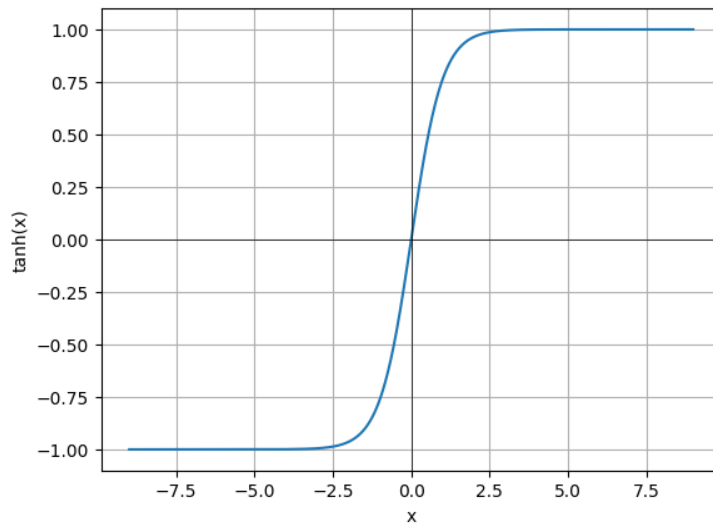
Đạo hàm của hàm sigmoid $\sigma(x)$ là:

$$\frac{d(\sigma(x))}{dx} = \frac{e^{-x}}{(1 + e^{-x})^2} = \sigma(x) - \sigma(x)^2 \leq \frac{1}{4}$$

- Hàm tanh $g(x)$ có công thức:

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

Giá trị của hàm $g(x)$ nằm trong khoảng $(-1, 1)$.



Hình 2.4: Hàm tanh

Đạo hàm của hàm tanh $g(x)$ là:

$$\frac{d(g(x))}{dx} = 1 - g(x)^2 \leq 1$$

2.2.4 Hàm mất mát - *Loss function*

Hàm mất mát (*Loss function*) là một hàm được sử dụng để đo lường sự khác biệt giữa giá trị dự đoán của mô hình và giá trị thực tế. Nó thường được sử dụng để đánh giá hiệu suất của mô hình trong quá trình huấn luyện mạng nơ-ron và điều chỉnh các tham số của mô hình để tối thiểu giá trị của hàm mất mát và đạt được kết quả dự đoán chính xác.

Hàm mất mát thể hiện sự chênh lệch giữa hai đại lượng:

- \hat{y} : giá trị dự đoán.
- y : giá trị thực tế.

Một số hàm mất mát thông dụng:

- Mean Square Error(MSE) - Trung bình bình phương sai số:

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

- Mean Absolute Error(MAE) - Trung bình sai số tuyệt đối:

$$L = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

2.2.5 Cập nhật trọng số

Sau khi có giá trị dự đoán từ mô hình và giá trị thực tế, ta tính toán được hàm mất mát, sử dụng thuật toán lan truyền ngược (*Backpropagation*) để giảm giá trị hàm mất mát xuống thấp nhất, tối ưu mạng nơ-ron.

Phương pháp phổ biến nhất để tối ưu mạng nơ-ron chính là *Gradient descent*. Giả sử ta cần tìm cực tiểu toàn cục cho hàm $f(\theta)$ trong đó θ là tập hợp các tham số cần tối ưu. *Gradient descent* tìm giá trị tiệm cận giá trị nhỏ nhất của hàm số bằng cách giảm dần các giá trị θ . Để áp dụng *Gradient descent*, chúng ta cần tính đạo hàm của $f(\theta)$ theo từng tham số θ được ký hiệu là $\nabla_{\theta}f(\theta)$.

Thuật toán *Gradient descent* bắt đầu bằng một điểm dự đoán θ_0 , sau đó ở vòng lặp thứ t , quy tắc cập nhật là:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta}f(\theta_t) \quad (2.1)$$

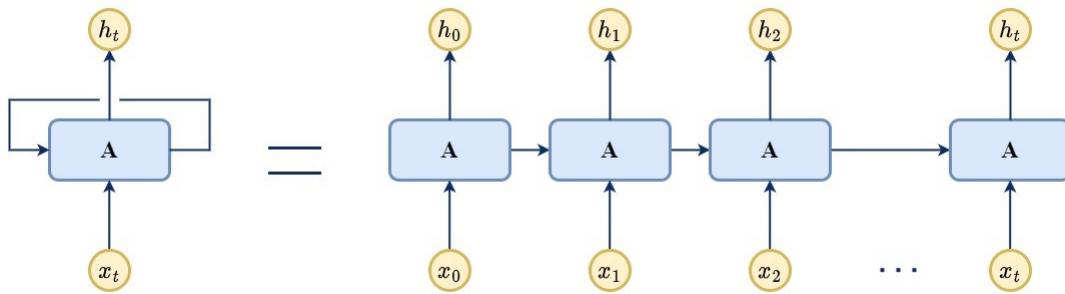
Trong đó: η là siêu tham số chỉ tốc độ học.

Ngoài dạng tổng quát như trên, công thức cập nhật *Gradient descent* còn có nhiều biến thể khác như *Stochastic Gradient Descent (SGD)*. Khác với *Gradient descent* khi cập nhật tham số cần tính đạo hàm trên tất cả tập dữ liệu (dữ liệu lớn việc tính toán trở nên rất tốn thời gian), thì *Stochastic Gradient Descent* cập nhật tham số bằng việc tính đạo hàm trên từng điểm dữ liệu, chấp nhận việc sai số lớn hơn nhưng có lợi ích về mặt tính toán. Công thức *Stochastic Gradient Descent* cũng tương tự như *Gradient Descent* nhưng thực hiện trên từng điểm dữ liệu.

2.3 Mô hình LSTM - *Long Short Term Memory*

2.3.1 Ý tưởng mô hình LSTM

Các hoạt động trong cuộc sống thường được diễn ra liên tiếp, tuần tự. Ví dụ như trong một câu văn, nếu ta chỉ có 1 từ thì rất khó để có thể dự đoán được từ tiếp theo, nhưng nếu chúng ta có nhiều dữ liệu về văn bản, từ ngữ trước đó thì ta có thể sử dụng để dự đoán ra từ kế tiếp. Để giải quyết vấn đề này, mạng Long Short Term Memory (LSTM) - Bộ nhớ dài ngắn hạn ra đời là một mạng tuần tự, ghi nhớ thông tin trước đó và sử dụng thông tin đó để xử lý đầu vào hiện tại, đưa ra dự đoán.



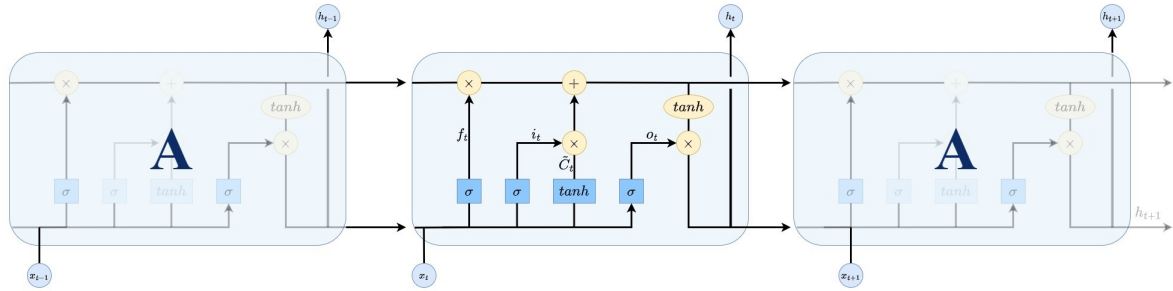
Long Short Term Memory (LSTM) là một kiểu đặc biệt của mạng nơ-ron hồi quy. Ý tưởng chính trong các loại mạng nơ-ron hồi quy là kết hợp các thông tin trong quá khứ để đưa ra dự đoán cho hiện tại.

LSTM là kiến trúc mạng được đề xuất để xử lý vấn đề này được giới thiệu bởi Hochreiter và Schmidhuber (1997). LSTM được thiết kế để tránh được vấn đề phụ thuộc xa, ghi nhớ thông tin trong suốt thời gian dài. LSTM không ghi nhớ tất cả các thông tin tại tất cả các thời điểm của chuỗi mà sẽ có cơ chế chọn lọc thông tin nào cần ghi nhớ, thông tin nào sẽ quên. Điều này đảm bảo thông tin ở thời điểm gần nhất so với thời điểm cần dự báo vẫn mang đầy đủ thông tin trong quá khứ, thông tin được lưu giữ tại thời điểm t là trạng thái tế bào tại thời điểm t . Trạng thái tế bào chính là điểm cốt lõi của LSTM.

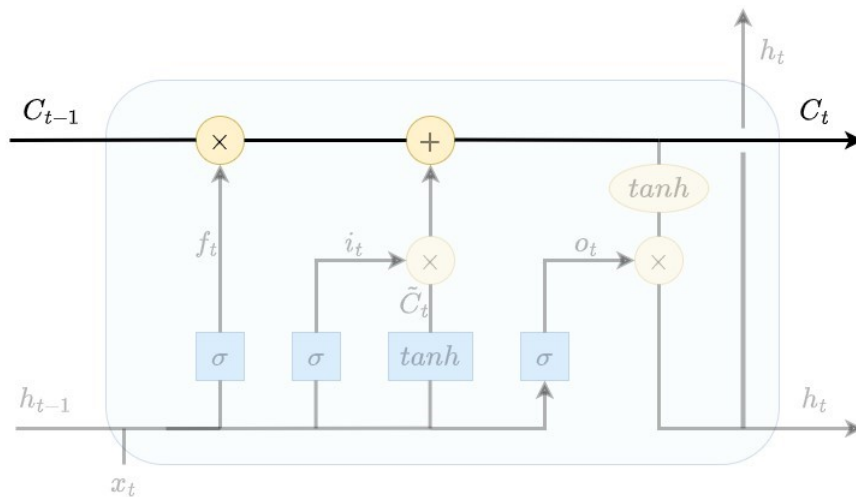
Mô hình LSTM là một kiến trúc mạng nơ-ron hồi quy được sử dụng rộng rãi trong cho các bài toán với dữ liệu dạng chuỗi

- Xử lý ngôn ngữ tự nhiên.
- Dự báo chuỗi thời gian.
- Tạo văn bản.
- Nhận dạng hình ảnh và video.

LSTM có dạng một chuỗi các nút lặp đi lặp lại của mạng thần kinh:



Từ mô hình trên, tại nút thứ t thấy được mô hình LSTM nhận các giá trị đầu vào là x_t và h_{t-1} là trạng thái ẩn của nút $t - 1$. Điểm đặc biệt của mô hình LSTM là trạng thái tế bào (Cell state) C_t - chính đường chạy thông ngang phía trên của sơ đồ. Trạng thái tế bào là một dạng giống như băng truyền, chạy xuyên suốt qua các nút mạng. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt.

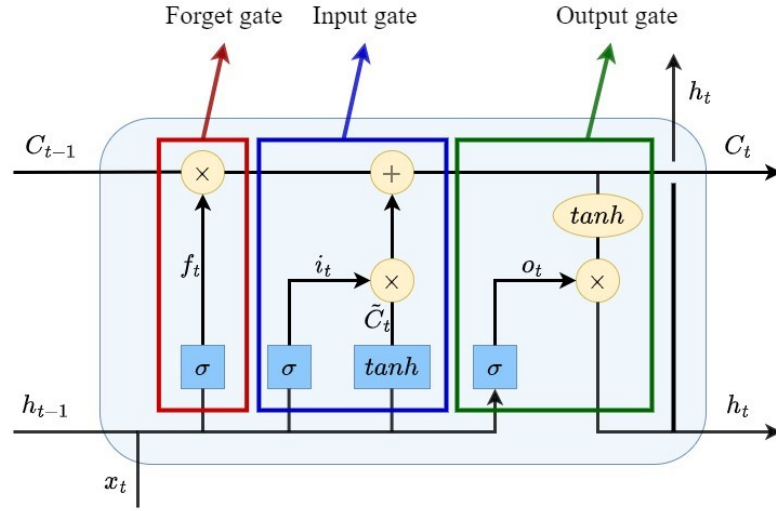


Hình 2.5: Mô hình Long Short Term Memory

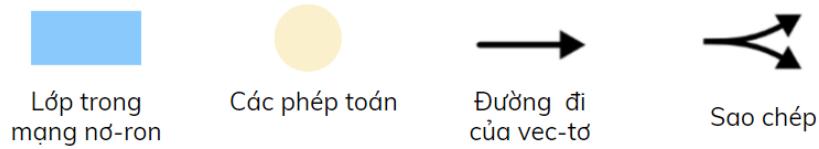
2.3.2 Các thành phần của một đơn vị trong mạng LSTM

LSTM có 5 thành phần để mô hình hóa cả dữ liệu ngắn hạn và dữ liệu dài hạn:

- Trạng thái tế bào - *Cell state* (C_t): được sử dụng để lưu trữ thông tin từ các thời điểm trước đó và giúp mạng LSTM có khả năng xử lý và nhớ thông tin trong quá khứ trong quá trình dự đoán tại thời điểm hiện tại.
- Trạng thái ẩn - *Hidden state* (h_t): được xem như một bộ nhớ trong mô hình, giúp mô hình ghi nhớ thông tin liên quan đến trạng thái ẩn trước h_{t-1} và sử dụng thông tin đó để đưa ra dự đoán hoặc phân tích chuỗi dữ liệu tiếp theo.
- Cổng đầu vào - *Input gate* (i_t): Cổng này xem xét thông tin đầu vào hiện tại và trạng thái trước đó và quyết định thông tin nào nên được ghi vào trạng thái mới của đơn vị bộ nhớ. Nó giúp mô hình xác định những phần quan trọng của thông tin mới.
- Cổng quên - *Forget gate* (f_t): Cổng này xem xét thông tin đầu vào hiện tại và trạng thái trước đó và quyết định thông tin nào nên được quên đi. Nó giúp mô hình xác định những phần quan trọng và không quan trọng của thông tin trong quá khứ.
- Cổng đầu ra - *Output gate* (o_t): Cổng này xem xét thông tin đầu vào hiện tại và trạng thái hiện tại của đơn vị bộ nhớ và quyết định thông tin nào nên được truyền qua đầu ra. Nó giúp mô hình xác định những phần quan trọng của thông tin cần được sử dụng trong các tác vụ tiếp theo.



Hình 2.6: Các thành phần của mô hình Long Short Term Memory



Hình 2.7: Một số ký hiệu

2.3.3 Quá trình thực hiện mô hình LSTM

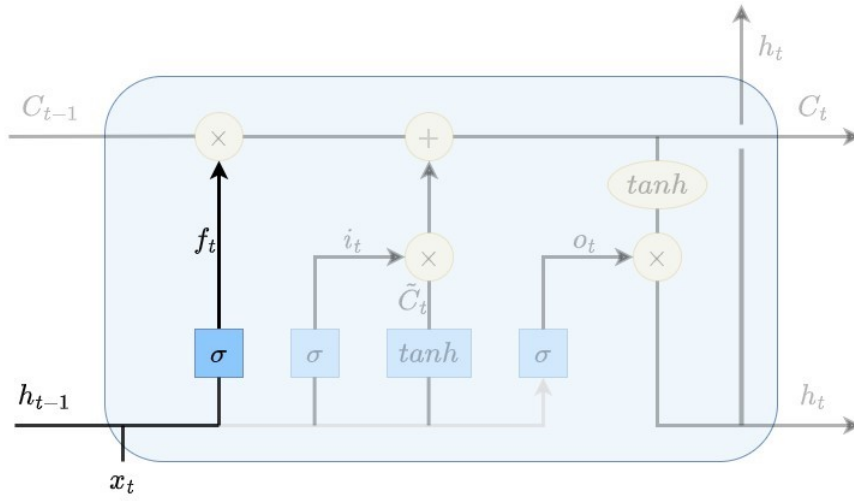
Chúng ta sẽ phân tích sâu hơn cách hoạt động tại thời điểm t trong mạng LSTM.

- Đầu vào của bài toán là dữ liệu dạng chuỗi $\{x_1, x_2, \dots, x_t\}$, $x_t \in \mathbb{R}^{k \times 1}$ là giá trị đầu vào tại thời điểm t , trong đó k là số đặc trưng của vectơ x .
- Trạng thái ẩn h_{t-1} làm đầu vào cho quá trình tính toán tại thời điểm t .
- Tương ứng với hai giá trị đầu vào x_t và h_{t-1} ta có ma trận trọng số $W \in \mathbb{R}^{d \times k}$ và $U \in \mathbb{R}^{d \times d}$ và vec-tơ độ lệch $b \in \mathbb{R}^{d \times 1}$. Trong đó, k là số đặc trưng, d là kích thước ẩn.
- Trạng thái tế bào C_t được cập nhật thông qua các bước quyết định của cổng trong mạng LSTM, bao gồm cổng quên, cổng đầu vào và cổng đầu ra. Các cổng này quyết định cách thông tin mới sẽ được thêm vào trạng thái tế bào và cách thông tin cũ sẽ được giữ lại hoặc loại bỏ.

Quá trình thực hiện mô hình LSTM được trình bày như sau:

- Bước 1: LSTM sẽ quyết định thông tin nào sẽ được loại bỏ khỏi tế bào thông qua tầng cổng quên. Cổng dùng hàm *sigmoid* với đầu vào là h_{t-1} , x_t và trả về một số nằm trong khoảng $(0, 1)$. Giá trị này càng gần 0 nghĩa là thông tin ít quan trọng, giá trị càng gần 1 nghĩa là thông tin càng quan trọng.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2.2)$$



Hình 2.8: Xóa các thông tin không cần thiết.

- Bước 2: Xác định thông tin mới nào sẽ được lưu trữ trong trạng thái tế bào. Quá trình xác định này được chia làm 2 giai đoạn.

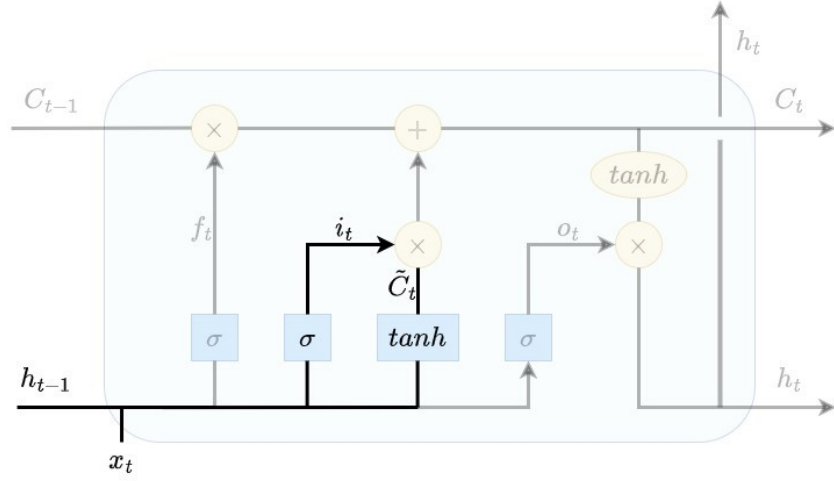
Giai đoạn 1: Sử dụng hàm *sigmoid* σ của cổng vào quyết định giá trị nào sẽ được cập nhật.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2.3)$$

Giai đoạn 2: Sử dụng hàm *tanh* tạo ra một vector trạng thái mới \tilde{C}_t sao cho \tilde{C}_t có thể thêm vào trạng thái.

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (2.4)$$

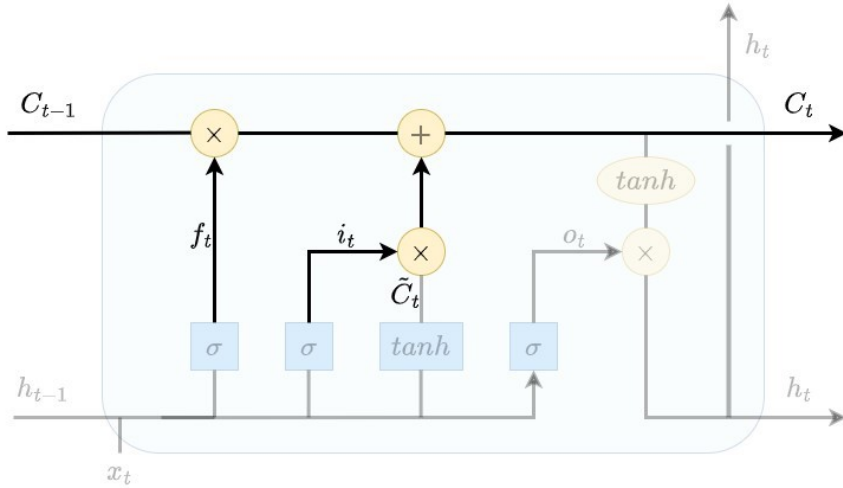
Trong bước tiếp theo sẽ kết hợp 2 giá trị này để tạo ra cập nhật cho trạng thái tế bào.



Hình 2.9: Quyết định các thông tin mới được thêm vào trạng thái tế bào

- Bước 3: Cập nhật trạng thái tế bào cũ C_{t-1} sang trạng thái tế bào mới C_t . Chúng ta sẽ nhân trạng thái cũ C_{t-1} với f_t để xác định thông tin nào sẽ được loại bỏ khỏi trạng thái tế bào cũ, sau đó sẽ cộng thêm trạng thái cập nhật mới $i_t * \tilde{C}_t$. Trạng thái tế bào sẽ được cập nhật như hình dưới đây.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.5)$$



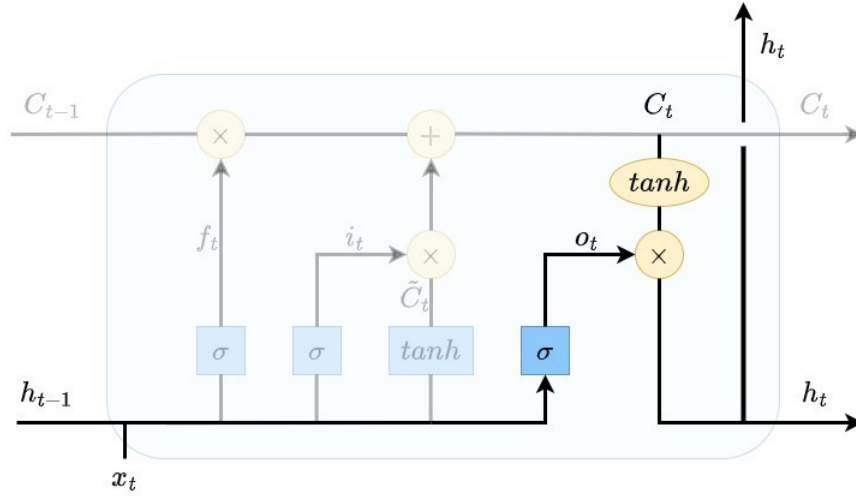
Hình 2.10: Cập nhật trạng thái ô C_t

- Bước 4: Xác định đầu ra của nút, đầu ra này phụ thuộc vào trạng thái của tế bào vừa được cập nhật nhưng vẫn tiếp tục được chọn lọc. Đầu tiên, sử dụng

hàm *sigmoid* với đầu vào x_t và h_{t-1} để quyết định phần nào của trạng thái tế bào ta muốn xuất ra thu được o_t . Sau đó, chúng ta đưa trạng thái tế bào vừa mới được cập nhật qua hàm *tanh* để có được giá trị nằm trong khoảng $(-1, 1)$, rồi nhân với giá trị đầu ra o_t vừa tìm được. Ta thu được giá trị cần dự đoán tại bước thứ t như hình dưới đây.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (2.6)$$

$$h_t = o_t * \tanh(C_t) \quad (2.7)$$



Hình 2.11: Điều chỉnh thông tin đầu ra.

Quá trình trên là cách hoạt động của một nút mạng LSTM áp dụng cho bài toán dự báo chuỗi thời gian, quá trình thực hiện tính toán dự báo giá trị đầu ra như trên được coi là quá trình tiến (*Forward pass*).

2.3.4 Huấn luyện mô hình

Với quá trình dự báo giá trị đầu ra của mô hình LSTM như trên, ta tiến hành huấn luyện mạng. Quá trình huấn luyện mạng hay điều chỉnh trọng số để giá trị hàm mất mát đạt giá trị nhỏ nhất là quá trình lùi (*Backward pass*).

LSTM sử dụng thuật toán lan truyền ngược thông qua thời gian để tìm ra bộ trọng số tối ưu. Với L là hàm mất mát, tính toán đạo hàm của hàm mất mát theo các trọng số đầu vào của mạng. Đạo hàm này cho biết cách mà hàm mất mát thay đổi

theo các tham số của mạng, từ đó ta có thể điều chỉnh các tham số này để tối ưu hóa mô hình.

Quá trình lùi được mô tả dưới đây, đặt:

$$\hat{C}_t = W_C x_t + U_C h_{t-1}$$

$$\hat{i}_t = W_i x_t + U_i h_{t-1}$$

$$\hat{f}_t = W_f x_t + U_f h_{t-1}$$

$$\hat{o}_t = W_o x_t + U_o h_{t-1}$$

Từ đó ta có:

$$\begin{aligned} z_t = \begin{bmatrix} \hat{C}_t \\ \hat{i}_t \\ \hat{f}_t \\ \hat{o}_t \end{bmatrix} &= \begin{bmatrix} W_C & U_C \\ W_i & U_i \\ W_f & U_f \\ W_o & U_o \end{bmatrix} \times \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \\ &= W \times I_t \end{aligned}$$

với I là vecto đầu vào liên kết, W là ma trận trọng số.

Để đơn giản tính toán ta tạm thời bỏ qua độ lệch, các công thức trong quá trình tiến được viết lại như sau:

$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1}) = \tanh(\hat{C}_t) \quad (2.8)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1}) = \sigma(\hat{i}_t) \quad (2.9)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1}) = \sigma(\hat{f}_t) \quad (2.10)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1}) = \sigma(\hat{o}_t) \quad (2.11)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.12)$$

$$h_t = o_t * \tanh(C_t) \quad (2.13)$$

Thực hiện tính đạo hàm hàm mất mát L từ đầu ra của mạng đến các lớp trước đó, sử dụng quy tắc chuỗi của đạo hàm và đạo hàm của các hàm kích hoạt *sigmoid*, *tanh*. Quá trình thực hiện được trình bày như sau:

- Với $h_t = o_t * \tanh(C_t)$ và $\delta h_t = \frac{\partial L}{\partial h_t}$, ta có đạo hàm của L theo o_t và C_t :

$$\begin{aligned}
\frac{\partial L}{\partial o_t} &= \frac{\partial L}{\partial h_t} * \frac{\partial h_t}{\partial o_t} \\
&= \delta h_t * \tanh(C_t) \\
\Rightarrow \delta o_t &= \delta h_t * \tanh(C_t)
\end{aligned} \tag{2.14}$$

$$\begin{aligned}
\frac{\partial L}{\partial C_t} &= \frac{\partial L}{\partial h_t} * \frac{\partial h_t}{\partial C_t} \\
\delta c_t &= \delta h_t * o_t * (1 - \tanh^2(C_t)) \\
\Rightarrow \delta C_t &= \delta h_t * o_t * (1 - \tanh^2(C_t))
\end{aligned} \tag{2.15}$$

- Với $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$ và $\delta C_t = \frac{\partial L}{\partial C_t}$ được tính như trên, ta có đạo hàm của L theo i_t , f_t , \tilde{C}_t và C_{t-1} :

$$\begin{aligned}
\frac{\partial L}{\partial i_t} &= \frac{\partial L}{\partial C_t} * \frac{\partial C_t}{\partial i_t} \\
&= \delta C_t * \tilde{C}_t \\
\Rightarrow \delta i_t &= \delta C_t * \tilde{C}_t
\end{aligned} \tag{2.16}$$

$$\begin{aligned}
\frac{\partial L}{\partial f_t} &= \frac{\partial L}{\partial C_t} * \frac{\partial C_t}{\partial f_t} \\
&= \delta C_t * C_{t-1} \\
\Rightarrow \delta f_t &= \delta C_t * C_{t-1}
\end{aligned} \tag{2.17}$$

$$\begin{aligned}
\frac{\partial L}{\partial \tilde{C}_t} &= \frac{\partial L}{\partial C_t} * \frac{\partial C_t}{\partial \tilde{C}_t} \\
&= \delta C_t * i_t \\
\Rightarrow \delta \tilde{C}_t &= \delta C_t * i_t
\end{aligned} \tag{2.18}$$

$$\begin{aligned}
\frac{\partial L}{\partial C_{t-1}} &= \frac{\partial L}{\partial C_t} * \frac{\partial C_t}{\partial C_{t-1}} \\
&= \delta C_t * f_t \\
\Rightarrow \delta C_{t-1} &= \delta C_t * f_t
\end{aligned} \tag{2.19}$$

- Có $\mathbf{z}_t = \begin{bmatrix} \hat{C}_t \\ \hat{i}_t \\ \hat{f}_t \\ \hat{o}_t \end{bmatrix} = W \times I_t$ và $\delta\tilde{C}_t, \delta i_t, \delta f_t, \delta o_t$ ở trên, sử dụng đạo hàm hàm kích hoạt *sigmoid*, *tanh* ta tính được δz_t như sau:

$$\delta\hat{C}_t = \delta\tilde{C}_t * (1 - \tanh^2(\hat{C}_t))$$

$$\delta\hat{i}_t = \delta i_t * i_t * (1 - i_t)$$

$$\delta\hat{f}_t = \delta f_t * f_t * (1 - f_t)$$

$$\delta\hat{o}_t = \delta o_t * o_t * (1 - o_t)$$

$$\Rightarrow \delta z_t = [\delta\hat{a}_t, \delta\hat{i}_t, \delta\hat{f}_t, \delta\hat{o}_t]^T \quad (2.20)$$

- Ta có $z_t = W \times I_t$, từ đó ta có

$$\delta W_t = \delta z_t * (I_t)^T \quad (2.21)$$

Nếu đầu vào của chúng ta có T bước thời gian hay T chính là số đơn vị trong kiến trúc mạng LSTM thì:

$$\delta W = \sum_{t=1}^T \delta W_t \quad (2.22)$$

Sau khi tính được δW , tham số W sẽ được cập nhật bằng cách sử dụng thuật toán tối ưu thích hợp. Từ đó, ta sẽ có được mô hình LSTM tối ưu để thực hiện dự báo.

Chương 3

Thử nghiệm số

Sau khi tìm hiểu lý thuyết về LSTM, trong báo cáo này em áp dụng vào việc xây dựng mô hình dự báo chỉ số giá chứng khoán VN-Index.

Chỉ số VN-Index là một chỉ số thị trường chứng khoán của Việt Nam, đại diện cho tất cả cổ phiếu niêm yết tại Sở Giao dịch Chứng khoán TP.HCM (HoSE). Chỉ số này được tính từ ngày thị trường chứng khoán Việt Nam đi vào hoạt động vào ngày 28/7/2000, với giá trị cơ sở ban đầu là 100 điểm.

Công thức tính VN-Index. là như sau:

$$\text{VN-Index} = \frac{\text{Tổng giá trị thị trường của các cổ phiếu niêm yết hiện tại}}{\text{Tổng giá trị của các cổ phiếu niêm yết cơ sở}} * 100 \quad (3.1)$$

Chỉ số VN-Index được tính toán và thay đổi trong quá trình diễn ra giao dịch trên thị trường chứng khoán. Sự biến động về giá cổ phiếu sẽ làm thay đổi giá trị của chỉ số này.

Sự biến động của chỉ số VN-Index phản ánh rủi ro hệ thống trong thị trường chứng khoán Việt Nam. Việc dự báo sự tăng giảm của VN-Index. có thể giúp các nhà đầu tư nhận biết chiều hướng biến động giá của các cổ phiếu trên thị trường này, đồng thời cung cấp thông tin về xu hướng và tình hình thị trường chứng khoán.

Bộ dữ liệu sử dụng của chỉ số chứng khoán VN-Index trong khoảng thời gian từ ngày 1/10/2020 đến ngày 29/12/2023. Bộ dữ liệu lấy từ trang web *investing.com*

3.1 Tổng quan về bộ dữ liệu

	time	close	open	high	low
0	2020-10-01	914.09	910.13	914.09	909.13
1	2020-10-02	909.91	914.09	916.67	897.72
2	2020-10-05	914.68	913.77	915.55	910.96
3	2020-10-06	915.67	918.11	918.93	912.77
4	2020-10-07	919.72	913.62	923.57	912.69
...
809	2023-12-25	1117.66	1103.06	1118.31	1103.06
810	2023-12-26	1122.25	1117.66	1123.11	1117.66
811	2023-12-27	1121.99	1122.25	1126.47	1121.99
812	2023-12-28	1128.93	1121.99	1130.58	1120.50
813	2023-12-29	1129.93	1128.93	1135.05	1128.93

814 rows × 5 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 814 entries, 0 to 813
Data columns (total 5 columns):
#   Column  Non-Null Count  Dtype
---  -
0   time    814 non-null    datetime64[ns]
1   close   814 non-null    float64
2   open    814 non-null    float64
3   high    814 non-null    float64
4   low     814 non-null    float64
dtypes: datetime64[ns](1), float64(4)
```

Hình 3.1: Tổng quan về bộ dữ liệu

- Bộ dữ liệu bao gồm 814 dòng, 5 cột đại diện cho các trường dữ liệu:
 - Time: thời gian
 - Open: giá mở cửa.
 - High: giá cao nhất trong phiên giao dịch.
 - Low: giá thấp nhất trong phiên giao dịch.

– Close: giá đóng cửa.

Mô hình sử dụng chỉ số giá đóng cửa trong quá khứ để dự đoán giá đóng cửa trong tương lai, vì thế cần loại bỏ các trường không cần thiết và giữ lại trường giá đóng cửa để tiến hành xây dựng và huấn luyện mô hình. Chỉ số giá đóng cửa được thể hiện theo thời gian như sau:



Hình 3.2: Giá đóng cửa theo thời gian

- Tiến hành chia bộ dữ liệu thành 2 tập *train* và *test*, với tỷ lệ 80% cho tập *train* và 20% cho tập *test*.
- Mô hình LSTM yêu cầu bộ dữ liệu là 1 chuỗi, mô hình sẽ sử dụng 30 ngày trước đó làm yếu tố quyết định và dự đoán giá cổ phiếu cho ngày thứ 31.

```
train_data = scaled_data[0:int(training_data_len),:]
x_train = []
y_train = []
for i in range(30, len(train_data)):
    x_train.append(train_data[i-30:i, 0])
    y_train.append(train_data[i, 0])
    if i <= 31:
        print(x_train)
        print(y_train)
        print()
x_train, y_train = np.array(x_train), np.array(y_train)
```

3.2 Xây dựng mô hình LSTM

Sử dụng thư viện Keras trong Python để xây dựng mô hình LSTM:

```
from keras.models import Sequential
from keras.layers import Dense, LSTM

model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape=(x_train.shape[1], 1)))
model.add(Dense(25))
model.add(Dense(1))
```

Mô hình sử dụng hàm mất mát *Mean Squared Error* và thuật toán tối ưu *Adam* để huấn luyện mô hình.

```
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(x_train, y_train, batch_size=32, epochs=100)
```

Sau khi tạo mô hình và chuẩn bị dữ liệu, mô hình còn yêu cầu một số tham số như sau:

- `batchsize`: là số lượng dữ liệu cho mỗi lần tính và cập nhật hệ số, huấn luyện mô hình với `batchsize = 32`
- `epochs`: số lượng epoch thực hiện trong quá trình training. (Một epoch là một lần duyệt qua hết các dữ liệu trong tập training set). Mô hình thực hiện qua 100 lần.

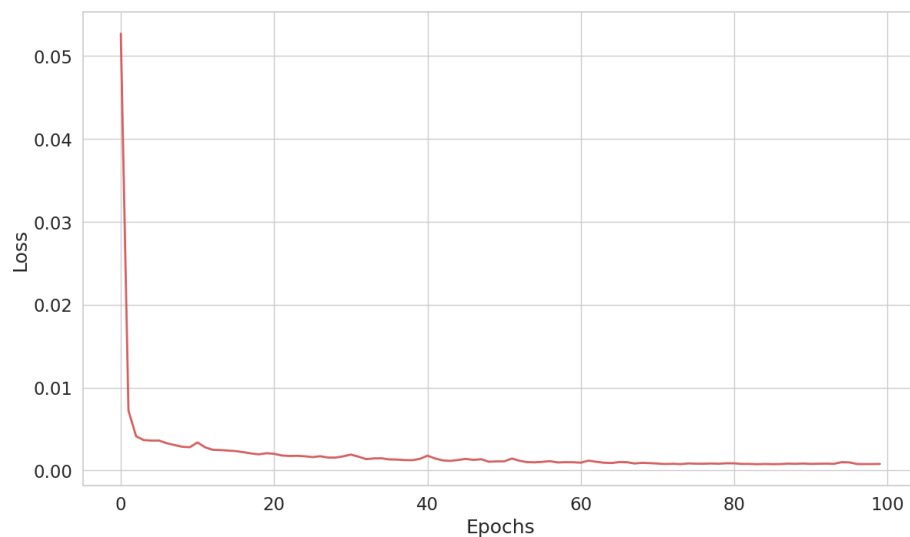

```

Epoch 1/100
15/15 [=====] - 5s 58ms/step - loss: 0.0921
Epoch 2/100
15/15 [=====] - 1s 90ms/step - loss: 0.0081
Epoch 3/100
15/15 [=====] - 1s 90ms/step - loss: 0.0043
Epoch 4/100
15/15 [=====] - 1s 90ms/step - loss: 0.0026
Epoch 5/100
15/15 [=====] - 1s 88ms/step - loss: 0.0022
Epoch 6/100
15/15 [=====] - 1s 51ms/step - loss: 0.0022
Epoch 7/100
15/15 [=====] - 1s 50ms/step - loss: 0.0022
Epoch 8/100
15/15 [=====] - 1s 49ms/step - loss: 0.0021
Epoch 9/100
15/15 [=====] - 1s 51ms/step - loss: 0.0021
Epoch 10/100
15/15 [=====] - 1s 53ms/step - loss: 0.0020
Epoch 11/100
15/15 [=====] - 1s 50ms/step - loss: 0.0020

```

Hình 3.3: Chạy mô hình

Từ kết quả hiển thị trên ta thấy được qua quá trình huấn luyện, lần lượt qua các lần duyệt, giá trị của hàm mất mát đang có xu hướng giảm dần và đạt tới giá trị tối ưu gần về 0. Chúng ta có thể thấy rõ hơn về sự giảm dần của hàm mất mát qua đồ thị sau:



Hình 3.4: Giá trị hàm mất mát giảm dần qua các epoch

- Sau khi xây dựng xong mô hình LSTM, tiến hành sử dụng mô hình để dự

báo kết quả cho tương lai với input là tập dữ liệu *test*:

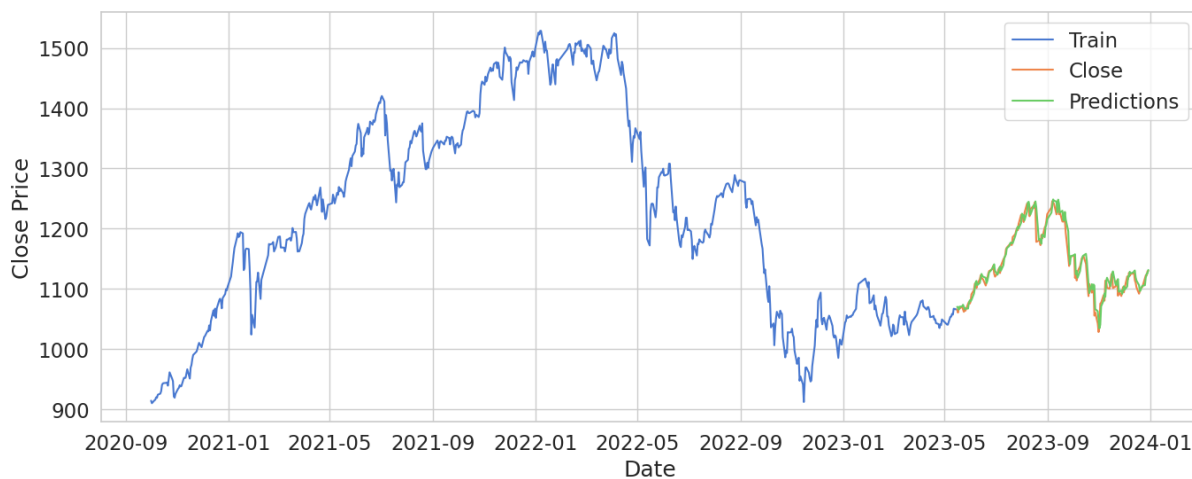
```
predictions = model.predict(x_test)
```

Kết quả dự báo của mô hình như sau:

	time	close	Predictions	Chênh lệch
0	2023-05-16	1065.91	1064.752441	1.157559
1	2023-05-17	1060.44	1064.735718	-4.295718
2	2023-05-18	1068.31	1060.074707	8.235293
3	2023-05-19	1067.07	1065.610352	1.459648
4	2023-05-22	1070.64	1065.658081	4.981919
...
185	2023-12-25	1117.66	1099.952515	17.707485
186	2023-12-26	1122.25	1113.004639	9.245361
187	2023-12-27	1121.99	1119.068237	2.921763
188	2023-12-28	1128.93	1119.555542	9.374458
189	2023-12-29	1129.93	1125.820923	4.109077

Hình 3.5: Kết quả dự báo của mô hình LSTM

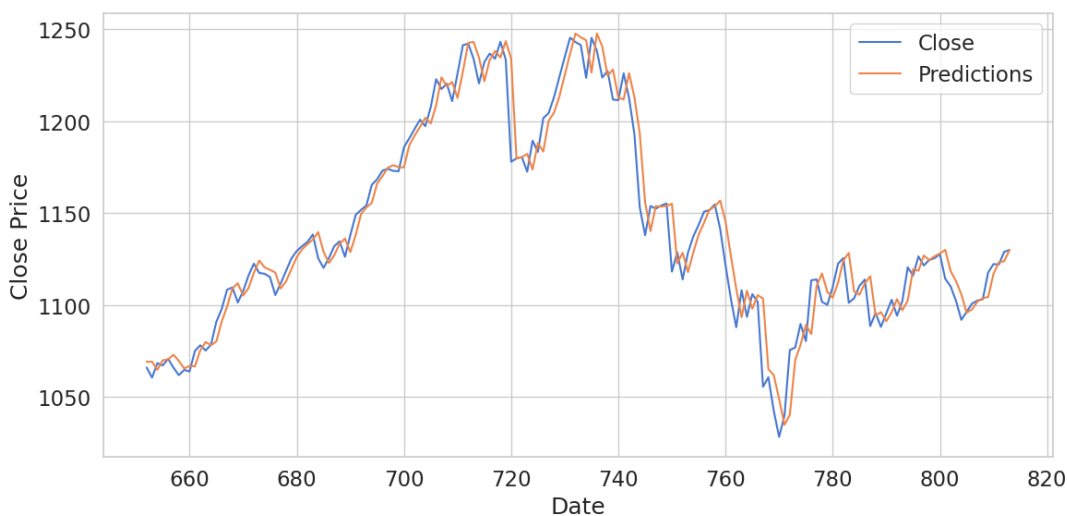
Biểu đồ so sánh giá dự báo và giá cổ phiếu thực tế:



Hình 3.6: Hiển thị kết quả dự đoán của mô hình

3.3 Dự báo kết quả và đánh giá mô hình

So sánh giá trị dự đoán và giá trị thực tế qua đồ thị theo thời gian:



Hình 3.7: So sánh giá trị dự đoán và giá trị thực tế

Từ đồ thị trên, ta thấy mô hình dự báo mới chỉ có dáng điệu tương tự với dữ liệu thực tế và chưa thực sự hoạt động tốt. Thực hiện tính toán hệ số $RMSE$ (*Root Mean Squared Error*) - Trung bình căn sai số để đánh giá mô hình. $RMSE$ là một độ đo thường được sử dụng để đánh giá mức độ chính xác của mô hình dự báo. Nó là một phép toán để tính độ lệch trung bình giữa các giá trị dự báo và giá trị thực tế, với mục tiêu là đạt giá trị thấp nhất.

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

Trong đó, \hat{y}_i là giá trị dự đoán, y_i là giá trị thực tế.

```
rmse = np.sqrt(np.mean(valid['Chênh lệch'] ** 2 ))  
print(rmse)
```

12.16365964713854

Giá trị sai lệch trung bình là 12.16 có thể cho thấy mô hình có sự chênh lệch đáng kể giữa giá trị dự báo và giá trị thực tế, mô hình chưa đủ tin cậy để dự đoán chính xác, cần được điều chỉnh và cải thiện. Tuy nhiên, mô hình cũng đã đánh giá được xu hướng tăng giảm của chỉ số VN-Index.

Dự báo chỉ số thị trường chứng khoán Việt Nam sử dụng mô hình LSTM có thể có sự khác biệt lớn so với giá trị thực tế trong một số ngày giao dịch. Điều này là do thị trường chứng khoán Việt Nam chịu ảnh hưởng mạnh từ nhiều yếu tố như tâm lý nhà đầu tư, tác động từ các thị trường chứng khoán khác và thông tin về sự thay đổi chính sách.

Do đó, nhà đầu tư nên kết hợp kết quả từ mô hình dự báo với phân tích kỹ thuật và thường xuyên quan sát để có cái nhìn đúng đắn và chính xác về sự biến động của thị trường chứng khoán. Mô hình dự báo có thể cung cấp một cơ sở dự đoán, nhưng không đảm bảo rằng các dự báo sẽ hoàn toàn chính xác trong mọi trường hợp.

Bằng cách kết hợp phân tích kỹ thuật và theo dõi thường xuyên, nhà đầu tư có thể nhận biết các yếu tố tác động đến thị trường và điều chỉnh quyết định đầu tư một cách linh hoạt. Điều này giúp tăng khả năng đưa ra quyết định đúng đắn và tận dụng mọi cơ hội trong môi trường thị trường chứng khoán đầy biến động.

Kết luận

Kết quả đạt được

Sau thời gian nghiên cứu với phạm vi đề án, em đã đạt được kết quả như sau:

- Tìm hiểu sơ lược về mạng nơ-ron và đi vào tìm hiểu mô hình Long Short Term Memory.
- Xây dựng chương trình dự đoán giá chứng khoán sử dụng mô hình LSTM và ngôn ngữ python.

Kỹ năng đạt được

- Bước đầu biết tìm kiếm, đọc, dịch tài liệu chuyên ngành liên quan đến nội dung đề án.
- Biết tổng hợp các kiến thức đã học và kiến thức trong tài liệu tham khảo để viết báo cáo đề án.
- Biết cài đặt thuật toán bằng ngôn ngữ lập trình hiện đại.

Hướng phát triển đề án trong tương lai

- Cải thiện độ chính xác trong phân tích dự báo của mô hình đã xây dựng
- Áp dụng Mô hình LSTM vào các bài toán với các dữ liệu khác như văn bản, hình ảnh,... .

- Xây dựng mô hình LSTM kết hợp với các mô hình dự báo khác như ARIMA,... để tăng độ chính xác cho bài toán dự báo.
- Kết hợp thêm các kỹ thuật phân tích giá chứng khoán để có thể đưa ra phân tích, dự báo chính xác hơn.

Tài liệu tham khảo

- [1] Charu C. Aggarwal. *Neural Networks and Deep Learning* (2018).
- [2] Jason Brownlee. *Long Short-Term Memory Networks With Python* (2017).
- [3] Diederik P. Kingma, Jimmy Ba. *Adam: A Method for Stochastic Optimization* (2017).
- [4] Hum Nath Bhandari, Binod Rimal, Nawa Raj Pokhrel, Ramchandra Rimal, Keshab R. Dahal, Rajendra K.C. Khatri. *Predicting stock market index using LSTM* (2022).
- [5] AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference? URL: <https://www.ibm.com/cloud/blog/ai-vsmachine-learning-vs-deep-learning-vs-neural-networks>.
- [6] Simple RNN vs GRU vs LSTM :- Difference lies in More Flexible control. URL: <https://medium.com/@saurabh.rathor092/simplernn-vs-gru-vs-lstm-differencelies-in-more-flexible-control-5f33e07b1e57>.