

MỤC LỤC

MỞ ĐẦU	2
I. TỔNG QUAN	3
1.1. Tổng quan về thị trường ngoại hối	3
1.2. Các hướng tiếp cận nghiên cứu	6
II. CƠ SỞ NGHIÊN CỨU	9
2.1. Các kiến trúc Recurrent Neural Networks.....	9
2.2. Giới thiệu mạng Long Short-Term Memory	13
2.3. Giới thiệu mạng Gated Recurrent Unit.....	16
III. ĐỀ XUẤT VÀ THỬ NGHIỆM MÔ HÌNH HỌC SÂU	17
3.1. Đề xuất mô hình.....	17
3.2. Thử nghiệm mô hình học sâu	17
IV. KẾT LUẬN.....	24
TÀI LIỆU THAM KHẢO.....	25

MỞ ĐẦU

Thị trường ngoại hối là thị trường tài chính lớn nhất trên thế giới, theo khảo sát ba năm một lần của BIS (BIS's 2019 triennial survey), giao dịch trên thị trường ngoại hối đạt mức đáng kinh ngạc 6,6 nghìn tỷ đô la mỗi ngày vào tháng 4 năm 2019. Nên thị trường ngoại hối có khối lượng giao dịch vượt trội so với sàn giao dịch chứng khoán lớn nhất trên thế giới - Nasdaq với khối lượng giao dịch trung bình hàng ngày khoảng 200 tỷ đô la.

Không giống như chứng khoán, thị trường ngoại hối là một trong những thị trường vô cùng phức tạp vì đặc điểm của nó là có tính biến động cao và phi tuyến tính. Thị trường ngoại hối không bị kiểm soát bởi một cá nhân hay tổ chức nào, mà nó luôn chịu sự tác động của nền kinh tế, chính trị, xã hội của các quốc gia, điều này làm cho việc dự đoán tỷ giá là một vấn đề còn nhiều thách thức. Tuy nhiên, việc dự đoán này sẽ cung cấp cho các nhà đầu tư một nền tảng tham khảo giúp họ tối ưu hóa lợi nhuận và giảm thiểu rủi ro. Trong tương lai, việc dự đoán tỷ giá của các mô hình sẽ là một phần của các hệ thống giao dịch tự động. Đã có nhiều phương pháp được áp dụng để phân tích thị trường ngoại hối như phân tích cơ bản và phân tích kỹ thuật.

Phân tích cơ bản sẽ đưa ra những dự đoán xu hướng của thị trường dựa trên những tin tức ảnh hưởng như là tỷ lệ lạm phát, lãi suất và tăng trưởng kinh tế.

Mặt khác, phân tích kỹ thuật dựa trên dữ liệu quá khứ, nó dựa báo xu hướng của thị trường dựa trên dữ liệu trước.

Có thể nhận thấy, hai hướng phân tích trên đều đã đạt những kết quả nhất định. Tuy nhiên việc kết hợp hai hướng phân tích đó còn đặt ra nhiều vấn đề cần được xử lý, và thử nghiệm. Nghiên cứu mô hình học sâu dự đoán xu hướng giá thị trường ngoại hối có thể kết hợp hai phương pháp phân tích thị trường tạo ra mô hình dự đoán tỷ giá ngoại hối đạt độ chính xác ổn định, phát triển mô hình như một kênh tham khảo giúp nhà đầu tư tối ưu hóa lợi nhuận và giảm thiểu rủi ro, qua đó đánh giá kết quả và khả năng ứng dụng thực tế.

I. TỔNG QUAN

1.1. Tổng quan về thị trường ngoại hối

1.1.1. Các thông tin tổng quát

Thị trường ngoại hối (tiếng Anh: Foreign Exchange Market, viết tắt là FOREX hay FX) là thị trường tiền tệ liên ngân hàng quốc tế, còn được nhắc đến dưới tên Thị trường Tiền mặt (Cash Market) hoặc Thị trường Liên ngân hàng Giao ngay (Spot Interbank Market). Thị trường Forex tồn tại bất cứ nơi nào ở đó tiền tệ của một quốc gia này được chuyển đổi thành tiền tệ của một quốc gia khác. Chính chúng ta sẽ vô tình trở thành người tham gia thị trường Forex nếu như, lấy một ví dụ đơn giản, trong một chuyến đi du lịch nước ngoài, chúng ta đổi tiền tệ của nước mình sang tiền tệ của nước mà chúng ta tới thăm để chi tiêu.

Thị trường Forex như hiện nay hình thành sau một loạt cải cách diễn ra vào năm 1971, sau khi hệ thống quản lý tiền tệ Bretton Woods bị xóa bỏ (thị trấn Bretton Woods ở bang New Hampshire, Mỹ, nơi Hội nghị Tài chính và Tiền tệ Quốc tế diễn ra vào năm 1944 dưới sự bảo trợ của Liên Hợp Quốc. Tại đây, Hiệp ước Bretton Woods đã được ký kết). Các nước có nền kinh tế phát triển đã duy trì tỷ giá hối đoái cố định bằng cách neo giá trị đồng tiền của mình vào vàng cho đến năm 1971. Đây cũng chính là nguyên tắc cốt lõi của thị trường tiền tệ cho đến thời điểm đó. Sau khi hệ thống Bretton Woods bị xóa bỏ, các nước này chuyển sang hệ thống tỷ giá thả nổi, trong đó mỗi đồng tiền đều có tính tự do chuyển đổi và đều có thể được mua bán trên thị trường theo giá cả được xác định dựa vào cung và cầu thực tế đối với đồng tiền đó.

Thị trường Forex:

- **Mang tính toàn cầu:** Do nó không có một trung tâm thanh toán tiền mặt tập trung. Nó bao gồm nhiều thành phần tham gia tại nhiều không gian địa lý khác nhau.

- **Có tính thanh khoản cao:** Số lượng lớn người tham gia thị trường khiến giá trị giao dịch lớn và cho phép bất cứ loại ngoại tệ nào cũng có thể được mua bán theo giá thị trường vào bất kỳ thời điểm nào.

- **Dễ dàng tiếp cận:** Thị trường Forex cũng như thông tin về nó, tin tức hay các chỉ số tài chính, có thể được tiếp cận một cách dễ dàng. Chúng ta có thể mở, đóng hay thay đổi trạng thái giao dịch của mình bất cứ lúc nào trong ngày.

- **Hoạt động 24 giờ mỗi ngày:** Giao dịch được thực hiện 24 giờ một ngày, từ thứ Hai đến thứ Sáu, trừ các ngày cuối tuần và một vài ngày nghỉ khác.

1.1.2. Một số thuật ngữ thường dùng trong Forex

Cặp tiền tệ

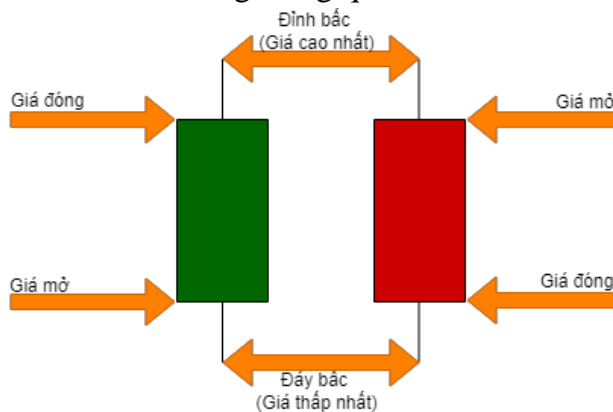
Cặp tiền tệ là sự định giá giữa hai loại tiền tệ khác nhau, giá trị của đồng tiền này được định giá thông qua đồng tiền khác. Tiền tệ thứ nhất trong cặp tiền tệ gọi là tiền cơ sở (base currency), tiền tệ thứ hai là tiền định giá (quote currency). Cặp tiền tệ so sánh giá trị của loại tiền tệ này (base currency) với tiền tệ khác (quote currency). Nó cho biết cần bao nhiêu đồng tiền định giá để mua một đơn vị tiền cơ sở. Đơn vị tiền tệ được xác định bằng

mã đơn vị tiền tệ ISO (ISO currency code) hoặc mã bảng chữ cái gồm ba chữ cái mà chúng được liên kết trên thị trường quốc tế. Vì vậy, đối với đô la Mỹ, mã ISO sẽ là USD. Ví dụ: Tỷ giá EUR/USD có Euro (EURO) là đồng tiền cơ sở, đô la Mỹ (USD) là đồng tiền định giá. Khi EUR/USD = 1.0814, có nghĩa là cần 1.0814 đô la Mỹ để mua 1Euro.

Biểu đồ nến

Khác với các loại biểu đồ thông thường, biểu đồ nến được cấu tạo bởi nhiều hình trụ có đường kẻ ở giữa, khá giống cây nến, nên được gọi là biểu đồ nến. Một cây nến trong thị trường Forex thường mang 4 thông tin dữ liệu chính:

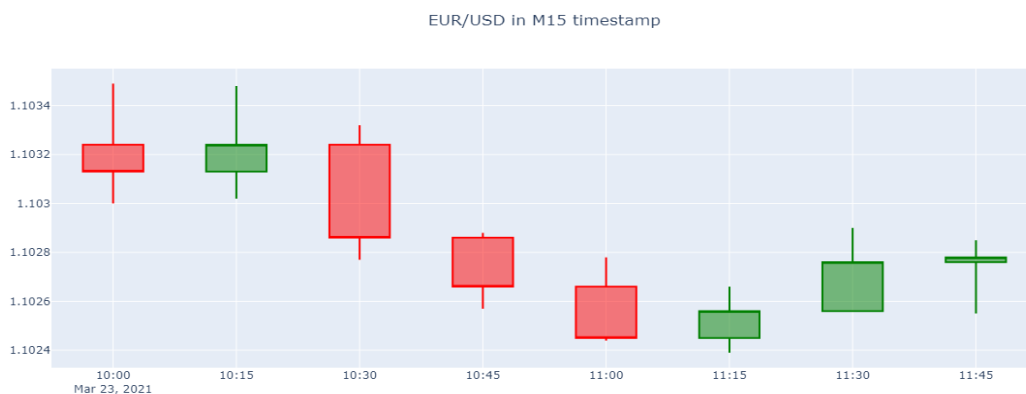
- Giá mở: Tỷ giá giao dịch đầu tiên trong quá trình hình thành một cây nến mới.
- Giá cao nhất: Tỷ giá lớn nhất trong một khoảng thời gian nhất định.
- Giá thấp nhất: Tỷ giá nhỏ nhất trong một khoảng thời gian nhất định.
- Giá đóng: Tỷ giá giao dịch cuối cùng trong quá trình hình thành một cây nến.

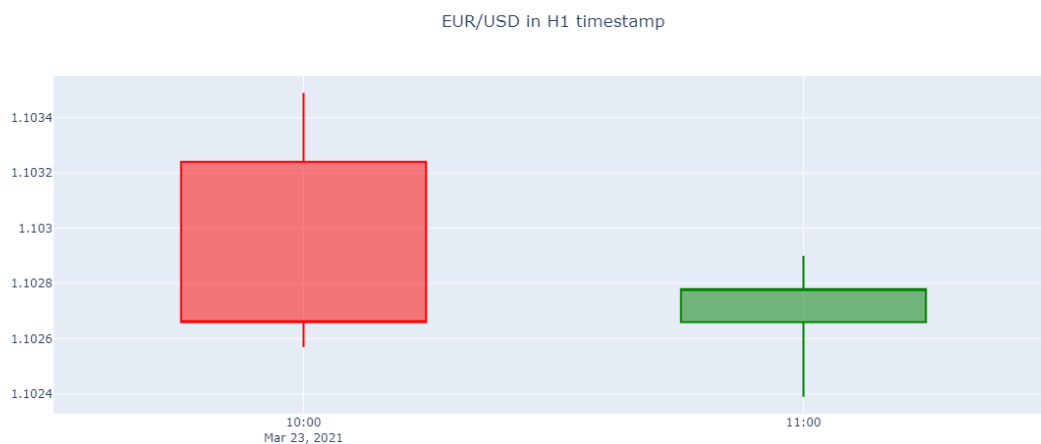


Hình 1.1 – Minh họa biểu đồ nến. Nến tăng (màu xanh); Nến giảm (màu đỏ)

Các khung thời gian

Trên hầu hết các sàn giao dịch, chúng ta có thể lựa chọn khung thời gian phù hợp để hiển thị các cây nến trên biểu đồ nến. Các khung thời gian ngắn hạn phổ biến như: M1 (1 phút), M15 (15 phút), H1 (1 tiếng) và một số khung thời gian khác. Việc lựa chọn khung thời gian thích hợp để tiện cho việc quan sát thị trường của cá nhân nhà đầu tư cũng là một yếu tố giúp tăng khả năng nắm bắt thị trường.





Hình 1.2 – Biểu đồ nến tương ứng ở hai khung thời gian (M15 và H1)

Giá Bid (bán) và giá Ask (mua)

Khi chọn một cặp tiền để giao dịch, sẽ luôn xuất hiện ở phần tỷ giá có 2 mức giá khác nhau, đó chính là giá mua (Buy hoặc Ask) và giá bán (Sell hoặc Bid). Giá Ask là giá sàn giao dịch chấp nhận bán ra hay sẽ là giá để chúng ta mua vào khi mở hoặc đóng lệnh giao dịch, giá Bid là giá mà sàn giao dịch chấp nhận mua từ chúng ta, hay bán cho sàn đó nếu muốn mở hoặc đóng lệnh. Trên thực tế thì giá Bid sẽ luôn luôn nhỏ hơn giá Ask, nếu thực hiện một lệnh Buy, ta phải mua với giá Ask, khi đã khớp lệnh, giá Bid là giá để tính lời lỗ của lệnh đang giao dịch. Ngược lại, nếu thực hiện một lệnh Sell, ta phải mua với giá Bid, khi đã khớp lệnh, giá Ask là giá để tính lời lỗ của lệnh đang chạy.

Spread

Spread là khoảng chênh lệch giữa giá mua và giá bán của một cặp tiền tệ. Khi đặt lệnh giao dịch trên thị trường, phí spread chính là chi phí chính của lệnh này. Hiện nay, ngoài phí hoa hồng, phí swap thì nguồn lợi nhuận của các sàn giao dịch Forex thu về từ các giao dịch của khách hàng trên sàn của mình là Spread, với mỗi sản phẩm, mỗi cặp tiền sẽ có mức Spread khác nhau. Do vậy, khi giao dịch các nhà đầu tư thường cần phải lựa chọn sàn giao dịch dựa trên tiêu chí Spread để tối đa hóa lợi ích cho mình bởi nó ảnh hưởng trực tiếp đến mức lợi nhuận khi giao dịch. Phí spread càng nhỏ, chi phí giao dịch càng thấp. Spread càng rộng thì chi phí càng cao.

Spread là sự chênh lệch giữa giá mua và giá bán. Vì vậy để tính spread, đơn giản chỉ cần lấy giá mua trừ giá bán. Giá sử dụng cặp tiền EUR/USD tại thời điểm có giá mua vào là 1,1135 và giá bán ra là 1,1133, khi đó phí spread được tính bằng cách lấy 1,1135 trừ 1,1133 thì được chênh lệch là 0,0002 tương đương 2 pips. Đây cũng là lý do mà ngay khi bạn mới mở lệnh, sẽ chịu một khoảng thua lỗ nhỏ tương ứng với số pip chênh lệch giữa giá mua và giá bán. Một số loại Spread:

- Spread nổi (Floating Spread): Mức chênh lệch giữa giá mua và giá bán của cặp tiền tệ luôn thay đổi tùy theo điều kiện thị trường.

- Spread cố định (Fixed Spread): Mức chênh lệch giữa giá mua và giá bán không thay đổi bất kể điều kiện thị trường là gì.

Pip

Pip là từ viết tắt của "percentage in point" hoặc "price interest point", là đơn vị nhỏ nhất dùng để đo lường sự biến động của tỷ giá cặp tiền tệ. Pip là mức di chuyển giá nhỏ nhất mà tỷ giá hối đoái có thể thực hiện dựa trên quy ước thị trường Forex. Hầu hết các cặp tiền tệ được định giá đến bốn chữ số thập phân và sự thay đổi pip là dấu thập phân cuối cùng (thứ tư). Do đó, một pip tương đương với 1/100 của 1% hoặc một điểm cơ bản. Nếu tỷ giá EUR/USD dao động từ 1.1080 đến 1.1081 có giá trị tăng 0.0001 USD tương đương với 1 pip.

Sự chuyển động của một cặp tiền tệ xác định liệu một nhà giao dịch có lãi hay lỗ từ các vị trí vào cuối ngày. Một nhà giao dịch thực hiện lệnh Buy với cặp tiền tệ EUR/USD sẽ thu được lợi nhuận nếu đồng EURO tăng giá trị so với đô la Mỹ. Nếu nhà giao dịch mua đồng euro với giá 1,1050 và đóng lệnh giao dịch ở mức 1,1085, họ sẽ kiếm được $1,1085 - 1,1050 = 35$ pips trên giao dịch.

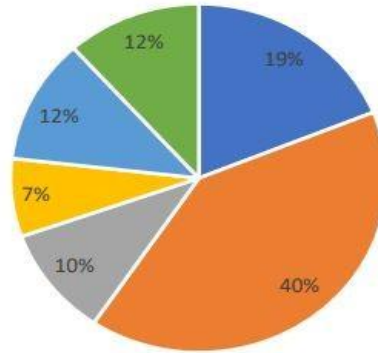
Đòn bẩy (Leverage)

Trong giao dịch và đầu tư, đòn bẩy là công cụ cung cấp khả năng đầu tư hoặc kiểm soát các quỹ lớn hơn nhiều so với các quỹ hiện đang nắm giữ. Đòn bẩy rất có lợi cho nhà giao dịch vì nó không yêu cầu phải sở hữu vốn lớn ngay từ đầu để thực hiện các giao dịch có giá trị cao và có thể mang lại lợi nhuận rất cao trong một thời gian ngắn. Đòn bẩy trong Forex được thể hiện dưới dạng tỷ lệ như từ 1:2, 1:5, 1:100, 1:500, 1:1000 cho đến đòn bẩy vô cực. Ví dụ khi giao dịch cặp EUR/USD với đòn bẩy là 1:100 (tương đương số tiền gấp 100 lần), ta đặt lệnh Buy ở mức giá 1.1050, cặp tiền tăng và sau đó đóng lệnh ở mức giá 1.1070, nghĩa là ta đã kiếm được 20 pips. Thay vì với mức lợi nhuận thông thường 1 pip ta thu được 0,1 USD thì với đòn bẩy gấp 100 lần thì 1 pip ta được 10 USD, 20 pips tương đương với 200 USD lợi nhuận mà ta thu được. Tuy nhiên, đây cũng là một "con dao hai lưỡi" vì đòn bẩy càng cao thì rủi ro càng lớn, khi chơi với đòn bẩy gấp 1000 lần nếu lãi chúng ta sẽ lãi rất lớn nhưng nếu lỗ ta cũng sẽ lỗ gấp 1000 lần.

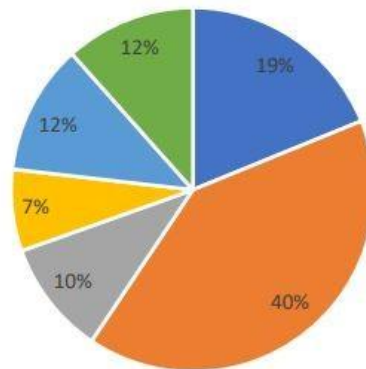
1.2. Các hướng tiếp cận nghiên cứu

Các công trình nghiên cứu của Khushi và các cộng sự [43] đã đề xuất nhiều phương pháp học sâu để phát triển mô hình dự đoán xu hướng giá Forex.

■ CNN ■ LSTM ■ DNN ■ RNN ■ Reinforcement learning ■ Other deep learning methods



■ CNN ■ LSTM ■ DNN ■ RNN ■ Reinforcement learning ■ Other deep learning methods



Hình 1.3 – Sự phân bố của các phương pháp học sâu áp dụng trong Forex được nghiên cứu trong các hội nghị lớn

Convolutional Neural Network

Convolutional Neural Network (CNN) là mạng nơ-ron có một hoặc nhiều lớp tích chập và được sử dụng chủ yếu để xử lý hình ảnh (image processing), phân loại (classification), phân đoạn (segmentation) và cả dữ liệu tương quan tự động khác vì khả năng nhận dạng mạnh mẽ của nó, việc sử dụng nó cũng được mở rộng sang lĩnh vực dự đoán kinh tế. Rút trích đặc trưng từ dữ liệu tài chính là một vấn đề quan trọng trong lĩnh vực dự đoán thị trường, và CNN là một trong những phương pháp tốt nhất để rút trích đặc trưng hình ảnh và dự đoán xu hướng giá.

Nếu các đặc điểm của thị trường chứng khoán tại một thời điểm cụ thể được coi là một biểu đồ đặc trưng, CNN có khả năng trích xuất các đặc điểm của thị trường chứng khoán tại thời điểm tương ứng từ các biểu đồ đặc trưng này. Do đó, CNN có thể được sử dụng để xây dựng mô hình lựa chọn thời gian và cuối cùng được sử dụng để hoàn thành việc xây dựng chiến lược lựa chọn thời gian.

Recurrent Neural Network

Recurrent Neural Network (RNN) bản chất là một mạng Neural Network và nó rất thích hợp trong việc lập mô hình và xử lý dữ liệu tuần tự. RNN có thể ghi nhớ trạng thái trước đó và nó có thể được sử dụng để tính toán trạng thái hiện tại, các lớp ẩn (hidden layers) không độc lập và đầu vào của lớp ẩn hiện tại không chỉ bao gồm đầu ra của lớp đầu vào mà còn bao gồm đầu ra của lớp ẩn trước đó. Vì thế, RNN sẽ xử lý rất tốt dữ liệu tuần tự. Ưu điểm của RNN là nó sẽ xem xét bối cảnh của dữ liệu trong quá trình huấn luyện, nên RNN rất phù hợp với sự biến động của chứng khoán và Forex vì xu hướng tại một thời điểm cụ thể thường có mối liên hệ nào đó với những xu hướng trước đó.

Long Short-Term Memory

Mô hình Long Short-Term Memory (LSTM) là một trong những biến thể của RNN được đề xuất bởi Sepp Hochreiter và Jürgen vào năm 1997. LSTM giới thiệu thiết kế vòng tự lặp để tạo ra đường đi của gradient, có thể liên tục chạy trong một thời gian dài. Trọng số của vòng tự lặp cũng được cập nhật trong mỗi lần lặp, điều này giải quyết vấn đề Gradient Vanishing dễ dàng gặp trong mô hình RNN khi cập nhật trọng số. Trong mô hình dự đoán xu hướng Forex, LSTM có thể học được các đặc trưng mà giá di chuyển tương ứng từ dữ liệu lịch sử Forex.

Deep Neural Network

Deep Neural Network (DNN) là một mạng Neural Network có ít nhất một lớp ẩn (hidden layer), cho phép mô hình thực hiện các hàm phi tuyến phức tạp và sở hữu tính trừu tượng rất cao, nên các lớp ẩn càng sâu, càng nhiều thì mô hình càng cải thiện. Vì DNN có khả năng dự đoán với dữ liệu lớn và có các mối quan hệ ánh xạ phi tuyến phức tạp, nên một hệ thống dự đoán xu hướng Forex thông minh có thể được thiết kế dựa trên DNN. Kỳ vọng rằng mô hình có thể đạt được độ chính xác cao hơn nhiều so với con người.

Reinforcement Learning

Reinforcement Learning là một trong những phương pháp học sâu tập trung vào việc hành động theo tình hình hiện tại để tối đa hóa lợi nhuận. Trong Reinforcement Learning, có hai thành phần cơ bản: trạng thái và hành động. Chiến lược được định nghĩa là thực hiện một hành động cụ thể trong một trạng thái cụ thể. Tất cả những gì người học phải làm là học một chiến lược tốt bằng cách liên tục khám phá và học hỏi.

Nếu trạng thái được coi là thuộc tính và hành động là nhãn, thì thật dễ để biết rằng cả Supervised Learning và Reinforcement Learning sẽ cố gắng tìm một ánh xạ và suy ra nhãn từ thuộc tính đã biết. Theo cách này, chiến lược trong học tăng cường tương đương với các bài toán phân lớp, hồi quy trong học có giám sát. Tuy nhiên, trong các vấn đề thực tế, Reinforcement Learning không có thông tin gắn nhãn như học có giám sát, và kết quả thường thu được sau khi thực hiện hành động. Do đó, Reinforcement Learning sẽ liên tục điều chỉnh chiến lược trước đó dựa vào kết quả của hành động trước, vì lý do này, thuật toán sẽ tìm hiểu: ở trạng thái nhất định thì nên thực hiện bước nào sẽ có kết quả có lợi nhất.

Vì thế, Reinforcement Learning có thể học được cách giao dịch như việc lựa chọn

giá để khớp lệnh, chọn thời gian giao dịch và kích thước lệnh tốt nhất theo phản ứng của thị trường rất tốt. Nó có thể xem thông tin ngữ cảnh (giá cả, tin tức, dư luận, lãi suất, phí, hành động giao dịch, lợi nhuận, thua lỗ) của giao dịch như một môi trường học hỏi củng cố. Lãi hoặc lỗ có thể được coi là phần thưởng cho quá trình học tăng cường, việc giao dịch là hành động và các yếu tố ngữ cảnh là trạng thái hoặc quan sát để đưa ra dự đoán về xu hướng Forex.

II. CƠ SỞ NGHIÊN CỨU

2.1. Các kiến trúc Recurrent Neural Networks

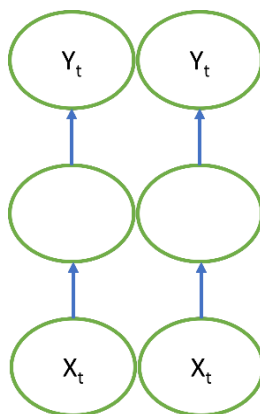
2.1.1 Giới thiệu

Mạng Recurrent Neural Networks (RNN) được phát triển dựa trên công trình của David Rumelhart vào năm 1986. RNN ra đời với ý tưởng chính là sử dụng một bộ nhớ để lưu lại chuỗi các thông tin từ những bước xử lý trước đó để đưa ra dự đoán cho bước hiện tại.

Các loại mô hình RNN

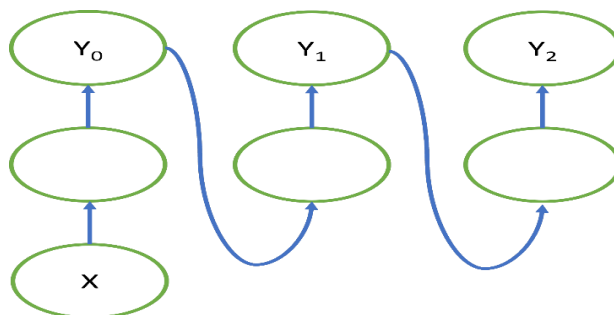
Dựa vào số lượng xử lý của chuỗi đầu vào và chuỗi đầu ra, mạng RNN được chia thành 4 loại chính với nhiều kiến trúc khác nhau.

One-to-One: Đây là mạng RNN truyền thống sử dụng kiến trúc One-to-One, cặp (x_t, y_t) duy nhất.



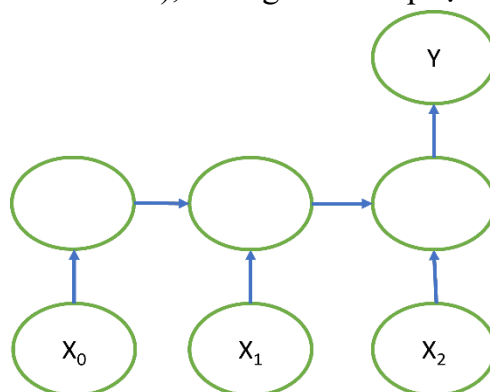
Hình 2.1 – Mô phỏng One-to-One RNN

One-to-Many: Đối với One-to-Many, một đầu vào duy nhất tại x_t có thể tạo ra nhiều đầu ra: (y_0, y_1, y_2) . Music Generation là một ví dụ.



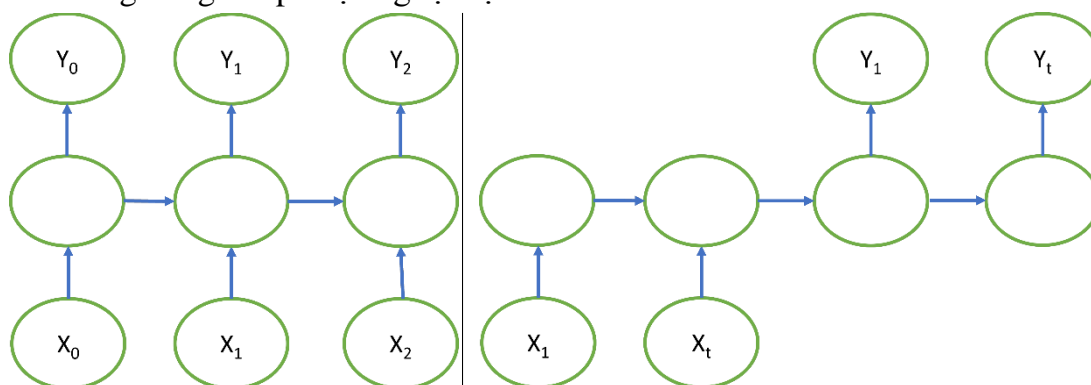
Hình 2.2 – Mô phỏng One-to-Many RNN

Many-to-One: Trong trường hợp này, nhiều đầu vào từ các bước thời gian khác nhau tạo ra một đầu ra duy nhất. Ví dụ: (x_t, x_{t+1}, x_{t+2}) có thể tạo ra một đầu ra duy nhất y_t . Các mạng như vậy được sử dụng trong phân tích cảm xúc (sentiment analysis) hoặc phát hiện cảm xúc (emotion detection),.. trong đó nhãn phụ thuộc vào một chuỗi các từ.



Hình 2.3 – Mô phỏng Many-to-One RNN

Many-to-Many: Có nhiều loại mô hình thuộc nhóm. Hình 2.4 là một ví dụ, trong đó ba đầu vào tạo ra ba đầu ra, hai đầu vào tạo ra hai đầu ra, lưu ý là số lượng đầu vào và đầu ra có thể khác nhau. Một ví dụ thực tế nhưng dịch máy, ví dụ: hệ thống dịch các từ tiếng Anh sang tiếng Pháp hoặc ngược lại.



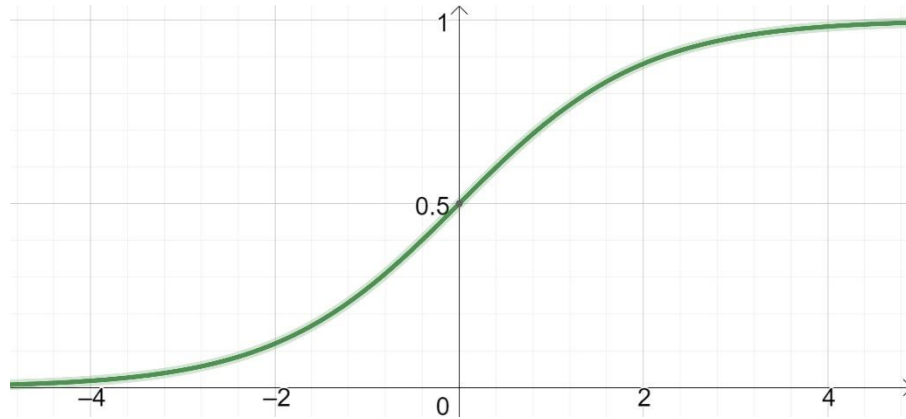
Hình 2.4 – Mô phỏng Many-to-Many RNN

Các hàm kích hoạt phổ biến trong các mô hình RNNs

Các hàm kích hoạt (Activation function) quyết định khi nào một nơ-ron được kích hoạt, khi nào thông tin được truyền qua nơ-ron khác dựa trên một phép biến đổi phi tuyến được tính toán bằng giá trị đầu vào, đầu ra sau khi biến đổi được dùng làm đầu vào cho nơ-ron tiếp theo.

Hàm Sigmoid: Nếu đầu vào lớn, hàm số sẽ cho đầu ra gần với 1; với đầu vào nhỏ (rất âm), hàm số sẽ cho đầu ra gần với 0. Trước đây, hàm kích hoạt này được sử dụng nhiều vì có đạo hàm rất đẹp. Tuy nhiên gần đây, hàm sigmoid chỉ được sử dụng ở output layer khi yêu cầu của đầu ra là các giá trị nhị phân.

$$Sigmoid(x) = \frac{1}{1+e^{(-x)}} \quad (2.1)$$

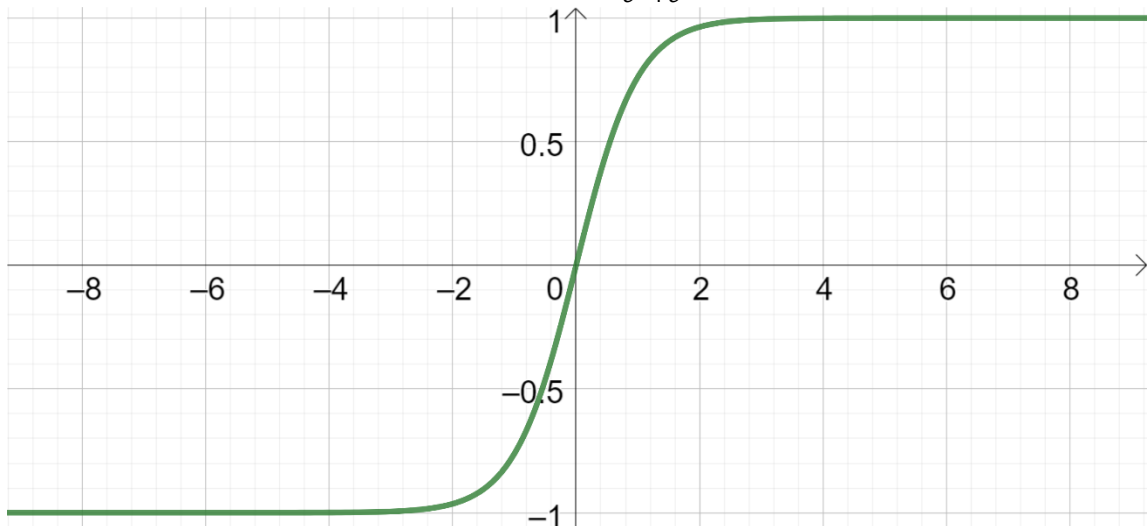


Hình 2.5 – Hàm kích hoạt Sigmoid

Hàm Tanh: Một hàm tương tự thường được sử dụng và mang lại hiệu quả tốt hơn là hàm tanh, hàm số này có tính chất đầu ra chạy từ -1 đến 1, khiến cho nó có tính chất zero-centered, thay vì chỉ dương như hàm sigmoid.

Một nhược điểm dễ nhận thấy là khi đầu vào có trị tuyệt đối lớn (rất âm hoặc rất dương), đạo hàm của cả sigmoid và tanh sẽ rất gần với 0. Điều này đồng nghĩa với việc các hệ số tương ứng với unit đang xét sẽ gần như không được cập nhật khi sử dụng công thức cập nhật gradient descent. Thêm nữa, khi khởi tạo các hệ số cho multilayer neural network với hàm kích hoạt sigmoid, chúng ta phải tránh trường hợp đầu vào một hidden layer nào đó quá lớn, vì khi đó đầu ra của hidden layer đó sẽ rất gần với 0 hoặc 1, dẫn đến đạo hàm bằng 0 và gradient descent hoạt động không hiệu quả.

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$

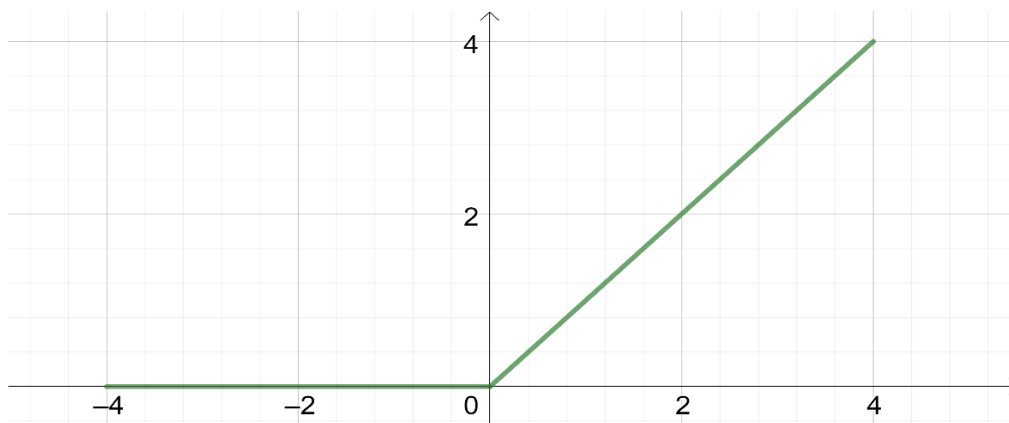


Hình 2.6 – Hàm kích hoạt Tanh

Hàm ReLU: ReLU (Rectified Linear Unit) được sử dụng rộng rãi gần đây vì tính đơn giản của nó. Hàm ReLU có công thức toán học rất đơn giản, rất lợi về mặt tính toán,

đạo hàm của nó bằng 0 tại các điểm âm, bằng 1 tại các điểm dương. ReLU được chứng minh giúp cho việc huấn luyện các multilayer neural network và deep network (rất nhiều hidden layer) nhanh hơn rất nhiều so với hàm tanh.

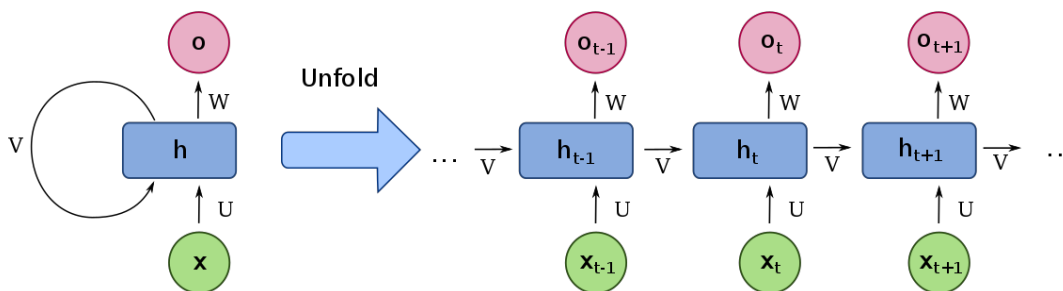
$$\text{ReLu}(x) = \max(0, x) \quad (2.3)$$



Hình 2.7 – Hàm kích hoạt ReLU

2.1.2. Mạng Recurrent Neural Networks

Khác với mạng nơ-ron truyền thống có tất cả các đầu vào và cả đầu ra là độc lập với nhau tức là chúng không liên kết thành chuỗi với nhau, mạng nơ-ron tuần hoàn là một loại mạng nơ-ron trong đó đầu ra của bước trước được cung cấp làm đầu vào cho bước hiện tại. Một vòng lặp cho phép thông tin có thể truyền đi từ bước này sang bước tiếp theo.



Hình 2.8 – Mô hình RNN truyền thống

Hình 2.8 mô tả dữ liệu được đưa vào một cách tuần tự, ở mỗi bước thời gian t ta sẽ đưa dữ liệu đầu vào x_t và nhận đầu ra là o_t với g là hàm kích hoạt thường là Tanh hoặc ReLU), có thể thấy h_t mang thông tin từ trạng thái trước đó h_{t-1} và đầu vào của trạng thái hiện tại x_t nên h_t giống như một bộ nhớ các đặc điểm của các đầu ra từ x₁ đến x_t.

$$h_t = g_h(W_h x_t + U_h h_{t-1} + b_h) \quad (2.4)$$

$$o_t = g_o(W_o h_t + b_o) \quad (2.5)$$

Các vấn đề về Gradient trong quá trình huấn luyện của mạng RNN

Về lý thuyết, RNN có thể sử dụng thông tin theo chuỗi dài tùy ý, nhưng trên thực tế, chúng bị hạn chế khi chỉ nhìn lại một vài bước trong các trường hợp dữ liệu quá dài mang một lượng thông tin lớn thì mạng RNN không thể nhớ được các thông tin xa trước đó. Vanishing Gradient (Gradient biến mất) và Exploding Gradient (Gradient bùng nổ) là những vấn đề thường gặp phải khi sử dụng các kỹ thuật tối ưu hóa trọng số dựa trên gradient để huấn luyện mạng nơ-ron xuất phát từ việc lựa chọn hàm kích hoạt không hợp lý hoặc số lượng các lớp ẩn của mạng quá lớn.

Ta có thể coi gradient là độ dốc của một hàm. Gradient càng cao, độ dốc càng lớn và mô hình càng có thể học nhanh hơn. Nhưng nếu độ dốc bằng 0, mô hình ngừng học. Một gradient chỉ đơn giản là đo lường sự thay đổi của tất cả các trọng số liên quan đến sự thay đổi của sai số.

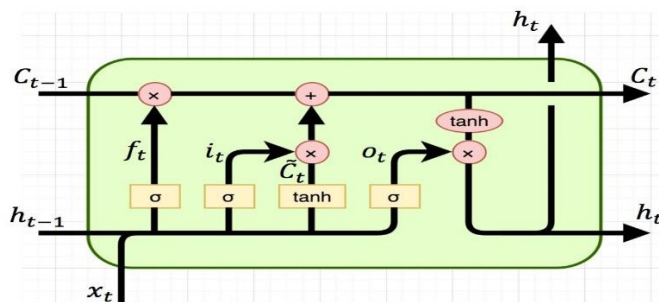
Vấn đề Exploding Gradient xảy ra khi thuật toán đánh trọng số quá cao, may mắn thay vấn đề này có thể được giải quyết bằng kỹ thuật Gradient Clipping.

Vấn đề Vanishing Gradient xảy ra khi các giá trị của gradient quá nhỏ và kết quả là mô hình ngừng học hoặc mất quá nhiều thời gian. Đây là một vấn đề lớn trong những năm 1990 và khó giải quyết hơn nhiều so với vấn đề Exploding Gradient. Ta có thể xử lý vấn đề này bằng cách sử dụng hàm kích hoạt ReLU có đạo hàm bằng 0 tại các điểm âm hoặc 1 tại các điểm dương, nên ta phần nào có thể kiểm soát vấn đề mất mát đạo hàm.

May mắn thay, những vấn đề của mạng RNN đã được giải quyết thông qua các biến thể khác của RNN, phổ biến có thể kể đến là mạng LSTM của Sepp Hochreiter và Juergen Schmidhuber và mạng GRU.

2.2. Giới thiệu mạng Long Short-Term Memory

Mạng Long Short-Term Memory (LSTM) là một phiên bản mở rộng của RNN được đề xuất bởi Sepp Hochreiter và Juergen vào năm 1997. LSTM được thiết kế để giải quyết vấn đề Gradient Vanishing trong mạng RNN do các bài toán phụ thuộc xa (Long-term dependencies) gây ra. Mạng LSTM có thể bao gồm nhiều tế bào LSTM (LSTM memory cell) liên kết với nhau và kiến trúc cụ thể của mỗi tế bào được biểu diễn như trong Hình 2.9. Kiến trúc LSTM bao gồm một trạng thái tế bào (cell state) C_t và ba cổng (gate) - một cổng quên (forget gate) f_t một cổng vào (input gate) i_t và một cổng ra (output gate) o_t .

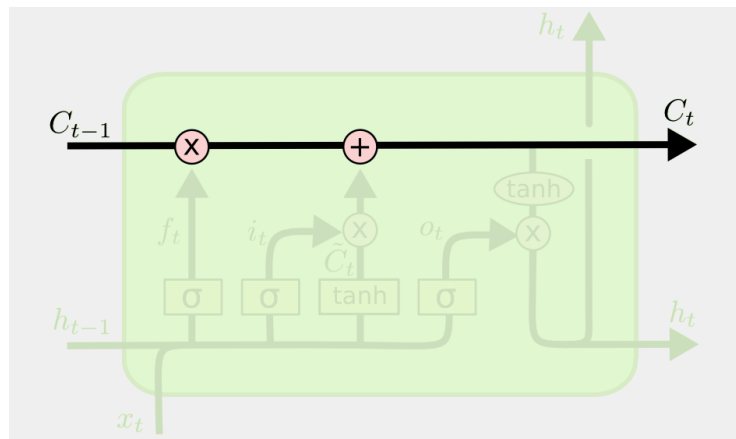


Hình 2.9 – Sơ đồ biểu diễn kiến trúc bên trong của một tế bào LSTM

Tại mỗi bước thời gian t , các cổng đều lần lượt nhận giá trị đầu vào x_t và giá trị h_{t-1} có được từ giá trị đầu ra của trạng thái trước đó.

Trạng thái tế bào (Cell state)

Trạng thái tế bào chính là đường chạy ngang như trong Hình 2.10. Các thông tin có thể dễ dàng truyền đi thông suốt qua nhiều tế bào LSTM mà không sợ bị thay đổi do trạng thái tế bào đi xuyên suốt qua các nơ-ron và chỉ tương tác tuyến tính đôi chút.

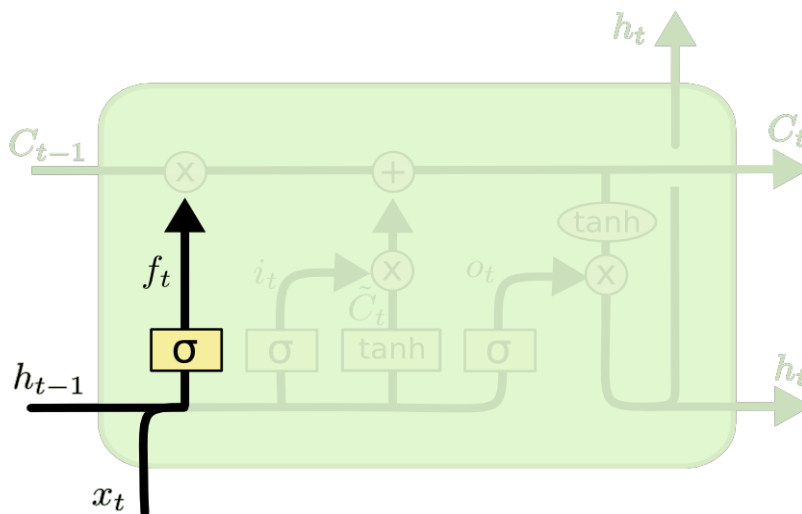


Hình 2.10 – Sơ đồ biểu diễn trạng thái tế bào LSTM

Cổng quên (Forget gate)

Cổng quên kiểm tra trạng thái ẩn trước đó h_{t-1} và đầu vào hiện tại x_t , sau đó xác định bộ phận nào sẽ được loại bỏ khỏi trạng thái tế bào. Quyết định này sẽ phụ thuộc bởi hàm sigmoid f_t luôn giữ kết quả là một con số có giá trị từ trong khoảng $[0,1]$ trong đó giá trị 1 là dấu hiệu để "giữ lại" toàn bộ thông tin và giá trị 0 là dấu hiệu để "loại bỏ" toàn bộ thông tin. Công thức được biểu diễn như sau:

$$f_t = \sigma(W_f * h_{t-1} + W_f * x_t + b_f) \tag{2.6}$$



Hình 2.11 – Sơ đồ biểu diễn cổng quên trong LSTM

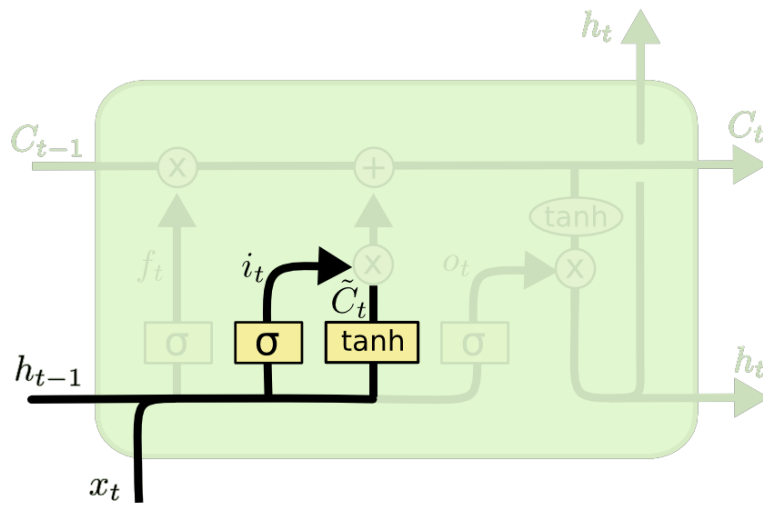
Cổng đầu vào (Input gate)

Cổng đầu vào quyết định thông tin mới nào sẽ được lưu trữ lại trạng thái tế bào. Hàm Sigmoid với công thức như bên dưới sẽ quyết định giá trị nào sẽ được cập nhật:

$$i_t = \sigma(W_i * h_{t-1} + W_i * x_t + b_i) \quad (2.7)$$

Sau đó, hàm Tanh tạo một vector \tilde{C}_t kết hợp với i_t để cập nhật cho trạng thái tế bào. Công thức \tilde{C}_t được biểu diễn như sau:

$$\tilde{C}_t = \tanh(W_c * h_{t-1} + W_c * x_t + b_c) \quad (2.8)$$



Hình 2.12 – Sơ đồ biểu diễn cổng đầu vào trong LSTM

Trạng thái tế bào mới C_t được cập nhật theo những thông tin được mang theo từ trạng thái tế bào trước đó C_{t-1} kết hợp với thông tin được tính toán tại trạng thái hiện tại và sẽ được truyền vào những tế bào LSTM tiếp theo. Công thức tính C_t được biểu diễn như sau:

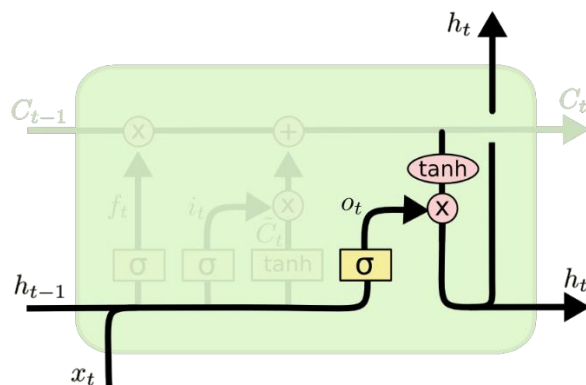
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (2.9)$$

Cổng đầu ra (Output gate)

Và cuối cùng, cổng đầu ra sẽ quyết định xuất cái gì. Hàm Sigmoid O_t sẽ lọc trạng thái tế bào để tiếp tục quyết định phần nào sẽ được truyền đến đầu ra. Để tạo đầu ra h_t , nhân o_t với trạng thái tế bào đi qua hàm Tanh để giữ tất cả các giá trị trong khoảng $[-1,1]$. Chi tiết có thể được biểu diễn như sau:

$$O_t = \sigma(W_o * h_{t-1} + W_o * x_t + b_o) \quad (2.10)$$

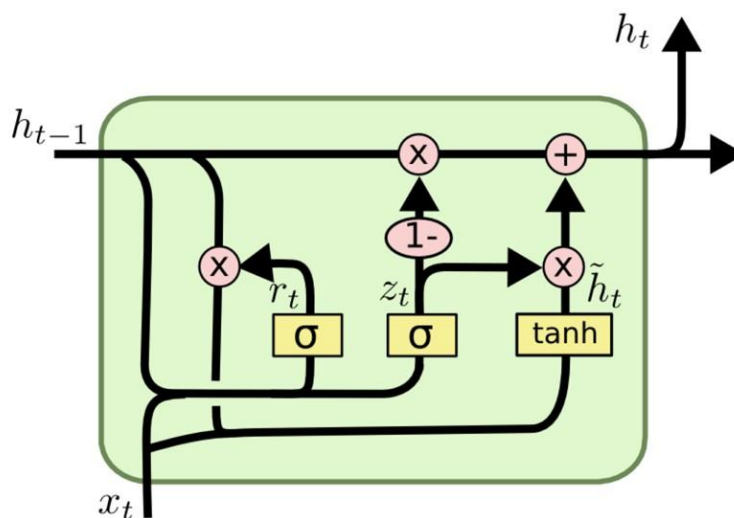
$$h_t = o_t * \tanh(C_t) \quad (2.11)$$



Hình 2.13 – Sơ đồ biểu diễn cổng đầu ra trong LSTM

2.3. Giới thiệu mạng Gated Recurrent Unit

Mạng Gated Recurrent Unit (GRU) được đề xuất bởi Cho và các cộng sự vào năm 2014. Đây là một biến thể đơn giản hơn của LSTM nhưng vẫn giữ được các đặc điểm tương tự như LSTM vì nó cũng hoạt động để giải quyết vấn đề bộ nhớ ngắn hạn của các mô hình RNN. Thay vì sử dụng thông tin điều chỉnh “trạng thái tế bào (state cell)”, nó sử dụng các trạng thái ẩn và thay vì ba cổng, nó có hai - một cổng đặt lại (reset gate) và một cổng cập nhật (update gate). Tương tự như các cổng trong LSTM, các cổng đặt lại và cập nhật kiểm soát số lượng và thông tin nào cần giữ lại, gửi đi.



Hình 2.14 – Sơ đồ biểu diễn mạng GRU

Ở cổng đặt lại (reset gate), hàm r_t quyết định bao nhiêu thông tin quá khứ cần quên, công thức của nó như sau:

$$r_t = \sigma(W_r * [h_{t-1}, x_t] + b_r) \quad (2.12)$$

Sau đó, cổng cập nhật (update gate) sẽ quyết định loại bỏ thông tin nào thông qua hàm z_t và thông tin mới nào cần thêm qua hàm h_t , cuối cùng hàm h_t đưa ra dữ liệu đầu ra.

$$z_t = \sigma(W_z * [h_{t-1}, x_t] + b_z) \quad (2.13)$$

$$h'_t = \tanh(W_h * [h_{t-1}, x_t] + b_h)$$

$$h_t = (1 - z_t) * h_t + z_t * h'_t$$

Thời gian huấn luyện mô hình GRU nhanh hơn LSTM và hiệu quả hơn do đã được cắt giảm bớt 1 công tính toán so với LSTM. Tuy nhiên, LSTM hoạt động tốt hơn GRU khi bài toán yêu cầu bộ nhớ dài hạn (long-term memory).

III. ĐỀ XUẤT VÀ THỬ NGHIỆM MÔ HÌNH HỌC SÂU

3.1. Đề xuất mô hình

Nghiên cứu này đề xuất và thử nghiệm hướng tiếp cận để xây dựng mô hình học sâu dự đoán xu hướng giá Forex bằng mô hình Long Short-term Memory là một trong những mô hình được áp dụng nhiều nhất bên cạnh Recurrent Neural Network (RNN), Convolutional Neural Network (CNN). Do hạn chế về mặt thời gian nên đề xuất và thử nghiệm chủ yếu tập trung vào mô hình LSTM.

3.2. Thử nghiệm mô hình học sâu

Chuẩn bị dữ liệu

Dữ liệu đầu vào được lấy từ website chuyên cung cấp thông tin về giá cả thị trường ngoại hối là: https://www.dukascopy.com/plugins/fxMarketWatch/?historical_data

The screenshot shows the Dukascopy website interface. The top navigation bar includes 'LIVE Telesision', 'Forex Data & Tools', and app download buttons for Google Play and the App Store. The main content area is titled 'Historical Data Export' and 'Historical Data Feed'. A sidebar on the left lists various market tools, with 'Historical Data Export' selected. The main list displays instruments with their symbols and names, such as WWD.US/USD (WOODWARD INC), WYNN.US/USD (WYNN RESORTS LTD), X.US/USD (UNITED STATES STEEL CORP), XAG/USD (Spot silver), XAU/USD (Spot gold), XLE.US/USD (Energy Select Sector SPDR Fund), XLF.US/USD (Financial Select Sector SPDR Fund), XLI.US/USD (Industrial Select Sector SPDR Fund), XLK.US/USD (Technology Select Sector SPDR Fund), XLM/USD (Stellar vs US Dollar), XLNX.US/USD (XILINX INC), XLP.US/USD (Consumer Staples Select Sector SPDR Fund), XLU.US/USD (Utilities Select Sector SPDR Fund), and XLV.US/USD (Health Care Select Sector SPDR Fund).

Chọn cặp tiền tệ và khung giờ cho dữ liệu dạng nến(Candlestick), khung thời gian

càng lớn càng ít nhiễu. Chọn khoảng thời gian lấy dữ liệu trong phần From date và To date:

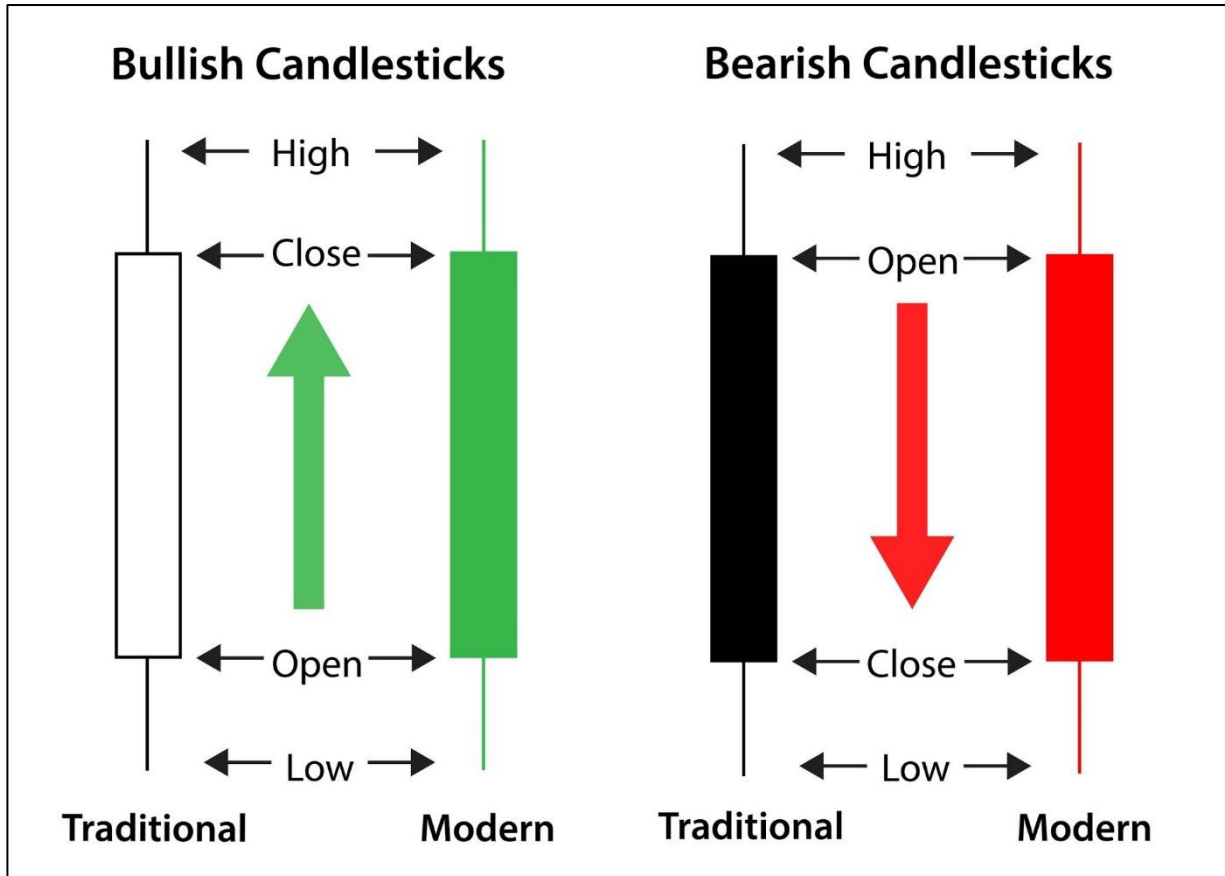
Candlestick: <input type="text" value="4"/> <input type="text" value="Hour"/>	From date: <input type="text" value="2023-06-19"/>	Filter flats: <input type="text" value="Disable"/>	Units <input type="text"/>
Offer side: <input type="text" value="Unit count"/>	To date: <input type="text" value="2023-06-24"/>	Day start time: <input type="text" value="EET"/>	<input type="text" value="Local"/> <input type="text" value="GMT"/>
<input type="button" value="Download"/>			

Khi chọn xong, có thể quan sát dữ liệu bằng cách nhấn vào nút "Show on chat", ta sẽ thấy biểu đồ hình nến:



Giải thích: Biểu đồ thông thường mà chúng ta hay gặp là dạng line, nhưng trong thị trường ngoại hối, giá cả không ngừng biến động khiến cho việc dùng biểu đồ line rất khó quan sát. Biểu đồ nến là phân tích trong giá từng khoảng thời gian thành 4 yếu tố: Giá mở cửa, đóng cửa, giá cao nhất, thấp nhất giống như hình trên. Mức giá tại điểm bắt đầu của khung thời gian gọi là Open, tại điểm kết thúc gọi là Close, giá cao nhất trong khung này gọi là High, và giá thấp nhất trong khung thời gian này gọi là Low.

Trong biểu đồ nến, cả khoảng dao động giá như hình trên được mô tả thành hình một cây nến:



- Nến biểu hiện giá tăng (bên trái hình): Cây nến có mức giá đóng cửa cao hơn mức giá mở cửa.
- Nến biểu hiện giá giảm (bên phải hình): Cây nến có mức giá đóng cửa thấp hơn mức giá mở cửa.
- Phần thân nến: Phần giá trong khoảng Open-Close
- Phần biến động hình que gọi là bóng nến nhé (shadow candlestick)

Qua biểu đồ nến, phần nào đó chúng ta có thể thấy thị trường đang giằng co như thế nào trong 1 khoảng thời gian, cây nến càng dài thì thị trường biến động càng mạnh, ngược lại, cây nến ngắn cho thấy thị trường bình ổn.

Tiền xử lý

Sau khi tải file csv về, ta tiến hành đọc và quan sát dữ liệu dưới dạng số:

```
raw_data = pd.read_csv("/content/drive/MyDrive/BCHT-2023/XAUUSD.csv")
raw_data.head()
```

Tại mỗi time step, dữ liệu của chúng ta có 5 đặc trưng:

	Local time	Open	High	Low	Close	Volume
0	29.05.2023 00:00:00.000 GMT+0700	1945.788	1945.788	1945.788	1945.788	0.000000e+00
1	29.05.2023 04:00:00.000 GMT+0700	1942.078	1946.255	1940.905	1941.485	1.922930e+06
2	29.05.2023 08:00:00.000 GMT+0700	1941.475	1946.135	1940.145	1945.705	2.939090e+06
3	29.05.2023 12:00:00.000 GMT+0700	1945.735	1948.835	1940.755	1945.205	3.016710e+06
4	29.05.2023 16:00:00.000 GMT+0700	1945.195	1949.575	1944.125	1948.685	3.082300e+06

- Open: Giá mở cửa
- High: Giá cao nhất trong khung thời gian đó
- Low: Giá thấp nhất trong khung thời gian đó
- Close: Giá đóng cửa
- Volume: Khối lượng giao dịch

Ngoài các đặc trưng nền, mà data cung cấp sẵn, ta có thể đưa vào một số đặc trưng khác như:

- Độ dài thân nến, bóng nến, đặc trưng nến là tăng hay giảm ...
- Trung bình giá qua các time step (MA)
- Chỉ báo liên quan sức mua, bán. Ví dụ như RSI
- Các tín hiệu phân kì trên các chỉ báo
- Và rất nhiều đặc trưng khác.

Trong báo cáo này chúng ta sẽ thử nghiệm các đặc trưng cơ bản của nến bao gồm: Open, High, Low, Close, các độ dài thân nến, bóng nến.

Kiểm tra trong data có trường nào rỗng hay không:

```
raw_data.isna().sum()
```

Loại bỏ những ngày thị trường đóng cửa, những ngày này, giá sẽ đi ngang và volume = 0:

```
raw_data = raw_data[raw_data.Volume !=0]
```

Tính toán một số đặc trưng:

```
[ ] high = raw_data["High"]
low = raw_data["Low"]
open_ = raw_data["Open"]
close = raw_data["Close"]
volume = raw_data["Volume"]
candle_body = open_ - close

candle_top_shadow = candle_body.copy()
candle_bot_shadow = candle_body.copy()

for i in range(len(candle_body)):
    if candle_body[i] > 0:
        candle_top_shadow[i] = high[i] - close[i]
        candle_bot_shadow[i] = open_[i] - low[i]
    else:
        candle_top_shadow[i] = high[i] - open_[i]
        candle_bot_shadow[i] = close[i] - low[i]

candle_values = [open_, high, low, close, candle_body, candle_top_shadow, candle_bot_shadow]
candle_values = np.asarray(candle_values).T
```

Ý tưởng lấy input và output là: Sử dụng một cửa sổ trượt trên `candle_values`, tại mỗi thời điểm, các giá trị trên khung đó sẽ lấy làm input train, bốn nền sau đó được đưa ra để tính output train.

```

length_x = 20
length_y = 4
step = 3
index = 0

while length_x + length_y + index < len(raw_data):
    x.append(candle_values[index: length_x + index])
    y_buffer = candle_values[index + length_x: index + length_x + length_y]
    result = []
    for i in range(length_y):
        mean = (y_buffer[i][1] + y_buffer[i][2])/2
        result.append(mean)

    y.append(result)
    index = index + step

num_split = int(0.85*len(y))
x = np.asarray(x)
y = np.asarray(y)

x_test = x[num_split:]
y_test = y[num_split:]

x_train = x[0: num_split]
y_train = y[0: num_split]

```

Mô hình

Chúng ta sử dụng một mô hình LSTM đơn giản để giải quyết:

```

model = Sequential()
model.add(TimeDistributed(Dense(128, activation="relu"), input_shape=(time_steps, features)))
model.add(LSTM(128))
model.add(Dense(16, activation="relu"))
model.add(Dense(4, activation="linear"))
model.compile(loss="mse", optimizer="adam")
model.summary()

```

Tranning với loss function là "mean squared error", optimizer là "adam" với 100 epoch:

```

hist = model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=100)
model.save_weights("lstm_stock_market_weight.h5")
model.save("lstm_stock_market_model.h5")
model_json = model.to_json()
with open("lstm_stock_market_json.json", "w") as json_file:
    json_file.write(model_json)

```

Test thử:

Chúng ta lấy 100 nến gần nhất để dự đoán kết quả:

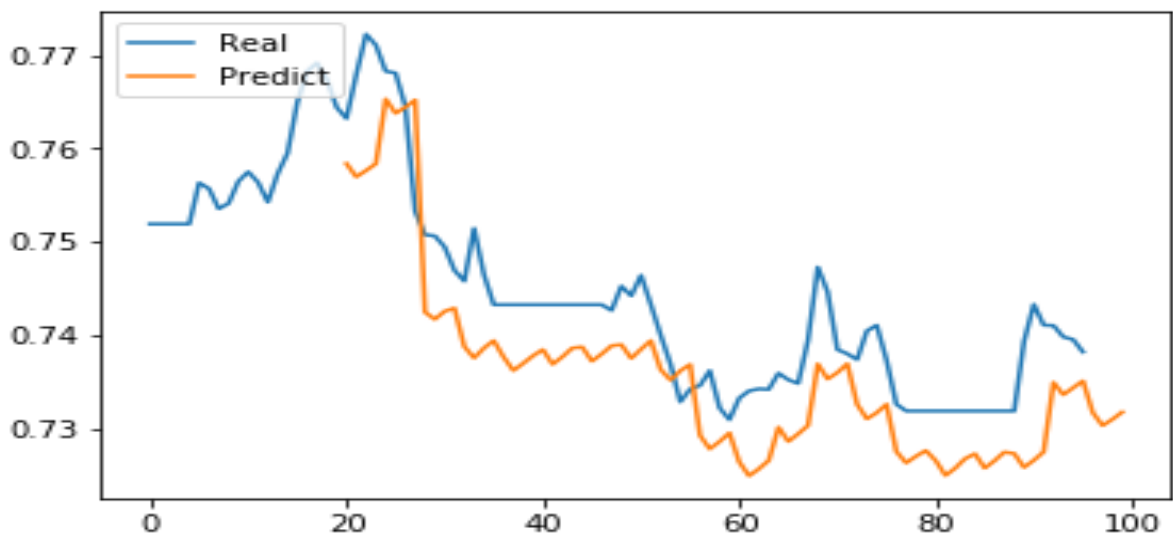
```
▶ data_test = candle_values[-100:]  
x_test = []  
length_x = 20  
step = 4  
index = 0  
while length_x + index < len(data_input):  
    x_test.append(data_input[index: length_x + index])  
    index = index + step  
x_test = np.array(x_test)  
result = model.predict(x_test)
```

Nổi kết quả tính được để vẽ biểu đồ:

```
▶ predict_plot = []  
for item in result:  
    predict_plot.extend(item)
```

Vẽ biểu đồ đường trung bình thực tế và đường do máy học dự đoán:

```
▶ plt.plot(simple_ma[-100:-4])  
plt.plot(range(20,20+len(predict_plot)), predict_plot)
```



Như chúng ta thấy, tuy giá trị thực tế với giá trị do máy dự đoán khác biệt nhau,

nhưng điểm bù của mô hình này là nó dự đoán trước 4 cây nến, từ đó có thể tận dụng để dự đoán xu hướng. Khác với một số mô hình dự đoán trước 1 cây nến - rất khó để áp dụng trong giao dịch thực tế.

IV. KẾT LUẬN

Trong thị trường ngoại hối có rất nhiều biến cố, không có 1 mô hình nào có thể dự đoán chính xác 100% nên trong giao dịch thực tế, mọi phương pháp hay mô hình nào đó chỉ mang tính tham khảo.

TÀI LIỆU THAM KHẢO

Zexin Hu, Yiqi Zhao & Matloob Khushi, “*A Survey of Forex and Stock Price Prediction Using Deep Learning*”, Applied System Innovation, Vol 4; (1), 2021.