

TRƯỜNG ĐẠI HỌC PHENIKAA  
KHOA CÔNG NGHỆ THÔNG TIN



**PHENIKAA**  
UNIVERSITY

**Báo cáo Đồ án cơ sở**

**Đề tài :** Tìm hiểu bài toán dự báo chuỗi thời gian trong tương lai sử dụng mô hình Neural Network và Deep Learning

**GVHD:** Nguyễn Văn Thiệu

**SVTH:** Bùi Huy Quyền Anh

**Trịnh Gia Khiêm**

**Lê Hoàng Ngọc Tú**

# Mục lục

<b>I Giới thiệu</b>	2
1: Giới thiệu chung	2
2: Mô tả bài toán	2
<b>II Kiến thức cơ sở</b>	3
1: Tổng quan về times series	3
2: Neural Network	3
2.1: Neural NetWork là gì ?	3
2.2: Đặc điểm của Neural Network	4
2.3: Kiến trúc Neural NetWork	4
2.4: Các hàm kích hoạt	5
3: Mạng Multi-layer Perceptron	6
3.1: Perceptron là gì ?	6
3.2: Giới thiệu về MLP (Multi-layer Perceptron)	7
3.3 :Cách thức hoạt động của mạng mạng Multi-layer	7
4: Mạng Recurrent Neural Network	8
4.1 Khái quát về mạng RNN	8
4.2 Điểm khác biệt giữa RNN và MLP	10
4.3 Điểm khác biệt giữa RNN và LSTM	10
5: Mạng Long Short Term Memory Network	10
5.1: Khái quát về mạng LSTM	10
5.2: Cách thức hoạt động mô hình LSTM	11
<b>III Thực nghiệm</b>	13
1: Giới thiệu về dữ liệu và Kỹ thuật tiền xử lý dữ liệu	13
2: Thiết lập tham số cho mô hình	15
2.1: Long Short Term Memory Model	16
2.2: Multi-layer Perceptron	16
3: Kết quả và nhận xét	17
3.1: So sánh 2 mô hình dựa vào ba chỉ số MSE, MAE, R2	17
3.2: So sánh kết dự đoán và kết quả thực tế cho cả tập train và tập test	18
3.3: Tối ưu hóa tham số mô hình	20
3.4: So sánh hai mô hình sau khi được tối ưu dựa vào ba chỉ số MSE, MAE, R2	20
3.5: So sánh kết dự đoán và kết quả thực tế cho cả tập train và tập test sau khi mô hình được tối ưu	21
<b>IV Kết luận</b>	<b>Error! Bookmark not defined.</b>
<b>V Tài liệu tham khảo</b>	24

# I Giới thiệu

**Đề tài :** Tìm hiểu bài toán dự báo chuỗi thời gian trong tương lai sử dụng mô hình Neural Network và Deep Learning

## 1: Giới thiệu chung

Dự báo chuỗi thời gian trong tương lai là một bài toán đã và đang được ứng dụng rất nhiều trên thế giới vào những năm gần đây. Từ nhiệt độ, độ ẩm, lượng mưa, giá cổ phiếu, Bitcoin ...v.v đã được áp dụng dự báo chuỗi thời gian cho tương lai bởi các dữ liệu có sẵn của quá khứ, hay những dữ liệu có thời gian bất thường. Dự đoán chuỗi thời gian cho tương lai ứng dụng Machine Learning đã giúp rất nhiều cho doanh nghiệp về thị trường chứng khoán.. hay về dự báo thời tiết có tỉ lệ chính xác cao. Việc ứng dụng mô hình Neural Network cho bài toán này đã và đang được các nhà nghiên cứu khoa học hay những doanh nghiệp công nghệ triển khai nghiên cứu những mô hình cho ra những kết quả tốt nhất và áp dụng vào thực tế. Việc dùng mô hình Neural Network giúp cho việc dự báo cho bài toán có độ chính xác cao hơn khi dùng Machine Learning

Đề tài trên sẽ giúp mọi người hiểu về bài toán dự đoán chuỗi thời gian cho tương lai sử dụng mô hình Neural Network là hai mô hình mạng Long Short Term Memory Network và Multi-layer Perceptron. Cụ thể phần báo cáo dưới sẽ nói rõ mô hình mạng với bộ dữ liệu *“Water-Spring-Amiata”* dự đoán nhiệt độ của Fiora trong bộ dữ liệu đó.

## 2: Mô tả bài toán

Đây là bài toán dự báo nhiệt độ cho khu vực Santa Fiora. Sử dụng bộ dữ liệu *Water\_Spring\_Amiata.csv*. Bộ dữ liệu này gồm có 19 cột và 7487 hàng. Nội dung chủ yếu trong bộ dữ liệu là lưu trữ thông tin về các vấn đề như nhiệt độ, tốc độ dòng chảy, lượng mưa của một số khu vực trong đất nước Italy được đo từ 01/01/2000 đến 30/06/2020. Nhưng vì có rất nhiều ô dữ liệu bị thiếu nên trong bài này chúng ta chỉ quan tâm đến việc tạo mô hình dự đoán nhiệt độ cho khu vực Santa Fiora (Italy) với 2 cột dữ liệu chính được sử dụng là Date và Temperature\_S\_Fiora.

Mục tiêu chính của bài toán là so sánh mức độ hiệu quả trong việc dự đoán Nhiệt độ của khu vực Santa Fiora trong bộ dữ liệu

*Water\_Spring\_Amiata.csv* đối với hai mô hình MLP (Multi-layer Perceptron) và LSTM (Long short term memory).

## II Kiến thức cơ sở

### 1: Tổng quan về times series

Time series là một chuỗi các điểm dữ liệu được đo theo từng khoảng khắc thời gian liên nhau theo một tần suất thời gian thống nhất.

Time Series bao gồm một số đặc trưng:

- Trend: Thành phần này chỉ ra xu hướng tổng quan của dữ liệu theo thời gian: lên hoặc xuống, tăng hoặc giảm
- Seasonality: Thành phần chỉ ra các xu hướng theo mùa vị, chỉ ra các pattern theo tháng, theo quý
- Cycle: Thành phần chu kỳ, nó khác seasonality ở chỗ thành phần này có sự vận động trong khoảng thời gian dài
- Irregular remainder: thành phần nhiễu còn lại sau khi trích xuất hết các thành phần ở trên, chỉ ra các điểm bất thường của các điểm dữ liệu

Dự báo chuỗi thời gian bằng Deep Learning: Việc sử dụng Deep Learning để dự báo chuỗi thời gian đã khắc phục được những hạn chế của Machine Learning với nhiều cách tiếp cận khác. Mạng thần kinh hồi quy (RNN) là kiến trúc cổ điển được sử dụng nhiều nhất cho bài toán Dự báo chuỗi thời gian

### 2: Neural Network

#### 2.1: Neural NetWork là gì ?

Neural Network là mạng lưới Nơ-ron nhân tạo. Đây là chuỗi thuật toán nhằm tìm kiếm quan hệ trong tập hợp dữ liệu hệ thống dựa theo cách thức hoạt động não bộ con người. Neural Networks thích ứng với mọi điều chỉnh từ đầu vào, cho ra kết quả đầu ra tốt nhất.

## 2.2: Đặc điểm của Neural Network

Mạng lưới nơ-ron nhân tạo hoạt động như nơ-ron trong não bộ con người.

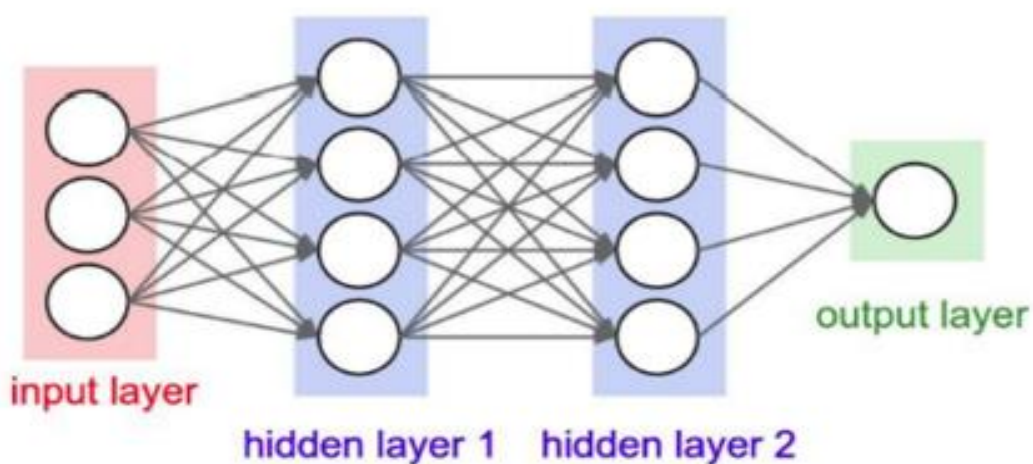
Neural Network tương đồng với những phương pháp thống kê theo đồ thị đường cong hoặc phân tích hồi quy

Mỗi nút là một tập hợp tri giác, cấu tạo tương tự hàm hồi quy đa tuyến tính, được sắp xếp liên kết với nhau.

## 2.3: Kiến trúc Neural Network

Mỗi một mạng lưới Nơ-ron nhân tạo là một Perceptron đa tầng, một Neural Network thường bao gồm 3 kiểu tầng cụ thể như sau:

- Input Layer (tầng đầu vào): Nằm bên trái của hệ thống, bao gồm dữ liệu thông tin đầu vào.
- Output Layer (tầng đầu ra): Nằm bên phải của hệ thống, bao gồm dữ liệu thông tin đầu ra.
- Hidden Layer (tầng ẩn): Nằm ở giữa tầng đầu vào và đầu ra, thể hiện quá trình suy luận và xử lý thông tin của hệ thống.



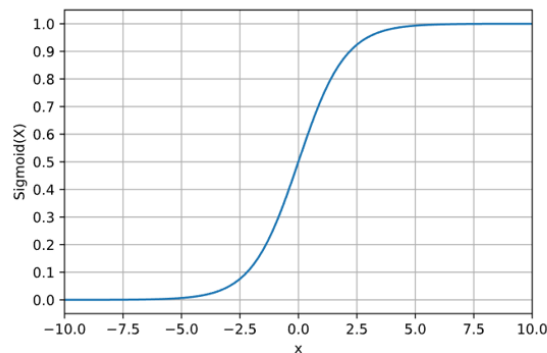
Hình 1: Cấu trúc Mạng nơ-ron

## 2.4: Các hàm kích hoạt

- Hàm Sigmoid

Công thức:

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



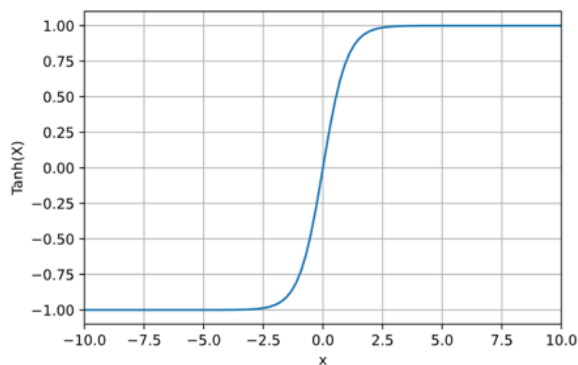
**Đồ thị hàm Sigmoid**

Hàm Sigmoid nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (0;1) (xem đồ thị phía trên). Đầu vào là số thực âm rất nhỏ sẽ cho đầu ra tiệm cận với 0, ngược lại, nếu đầu vào là một số thực dương lớn sẽ cho đầu ra là một số tiệm cận với 1. Trong quá khứ hàm Sigmoid hay được dùng vì có đạo hàm rất đẹp

- Hàm Tanh

Công thức:

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

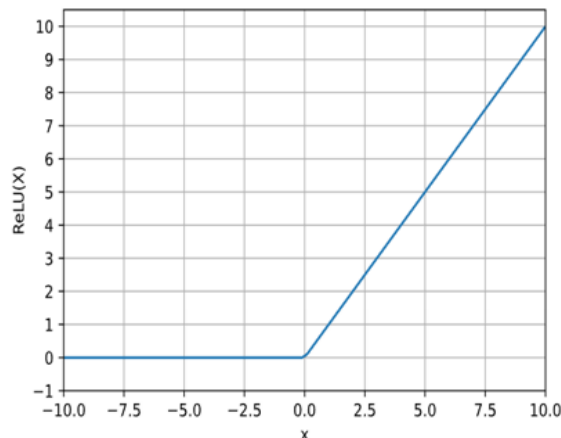


**Đồ thị hàm Tanh**

Hàm Tanh nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (-1; 1). Cũng như Sigmoid, hàm Tanh bị bão hoà ở 2 đầu (gradient thay đổi rất ít ở 2 đầu). Tuy nhiên hàm Tanh lại đối xứng qua 0 nên khắc phục được một nhược điểm của Sigmoid. Hàm tanh còn có thể được biểu diễn bằng hàm sigmoid như sau:  $\text{Tanh}(x) = 2\sigma(2x) - 1$

- Hàm ReLu  
Công thức:

$$f(x) = \max(0, x)$$



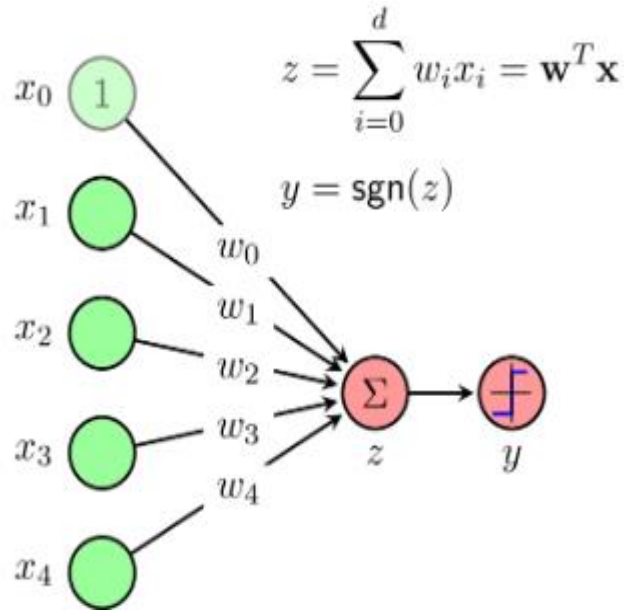
**Đồ thị hàm ReLu**

Hàm ReLU đang được sử dụng khá nhiều trong những năm gần đây khi huấn luyện các mạng neuron. ReLU đơn giản lọc các giá trị < 0. Nhìn vào công thức chúng ta dễ dàng hiểu được cách hoạt động của nó.

### 3: Mạng Multi-layer Perceptron

#### 3.1: Perceptron là gì ?

Perceptron là một thuật toán học máy tuyến tính được sử dụng để học có giám sát (supervised learning) cho các bộ phân loại nhị phân khác nhau. Một perceptron có thể được định nghĩa là một mạng nơ-ron với một lớp duy nhất phân loại dữ liệu tuyến tính.

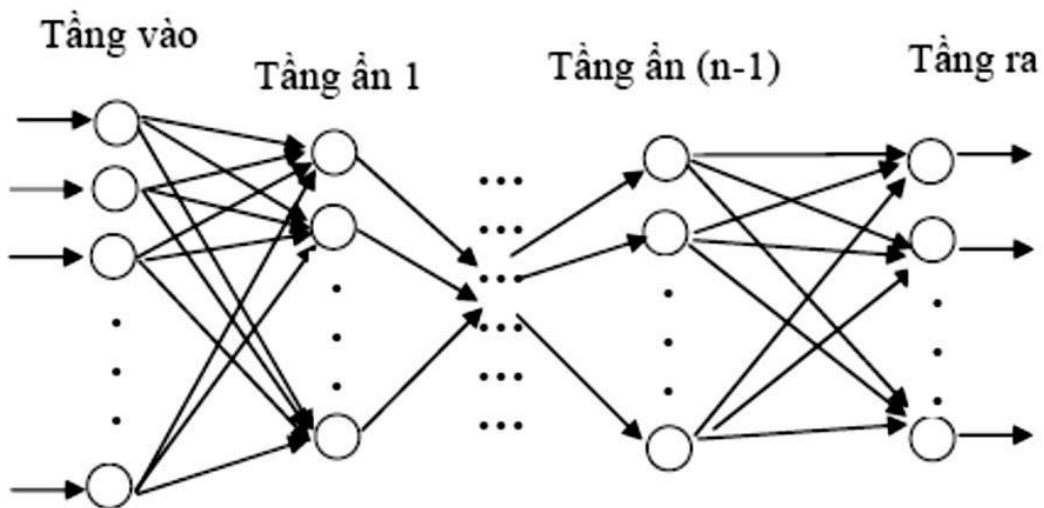


Mô hình đơn Perceptron

### 3.2: Giới thiệu về MLP (Multi-layer Perceptron)

Là sự kết hợp của nhiều perceptron hay còn gọi là perceptron đa tầng. Là mạng được sử dụng rộng rãi với nhiều tầng truyền thẳng. Một mạng MLP tổng quát là mạng có  $n$  ( $n \geq 2$ ) tầng (thông thường tầng đầu vào không được tính đến): trong đó gồm một tầng đầu ra (tầng thứ  $n$ ) và  $(n-1)$  tầng ẩn.

### 3.3 :Cách thức hoạt động của mạng mạng Multi-layer



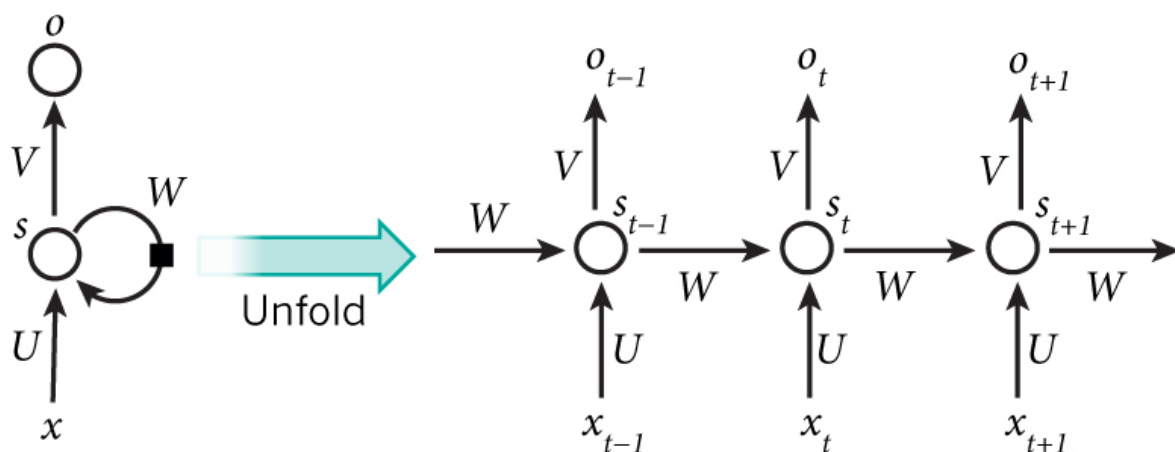


Hoạt động của mạng MLP như sau: Tại tầng đầu vào các neural nhận tín hiệu vào xử lý (tính tổng trọng số, gửi tới hàm truyền) rồi cho ra kết quả (là kết quả của hàm truyền); kết quả này sẽ được truyền tới các neural thuộc tầng ẩn thứ nhất; các neuron tại đây tiếp nhận như là tín hiệu đầu vào, xử lý và gửi kết quả đến tầng ẩn thứ 2. Quá trình tiếp tục cho đến khi các neural thuộc tầng ra cho kết quả.

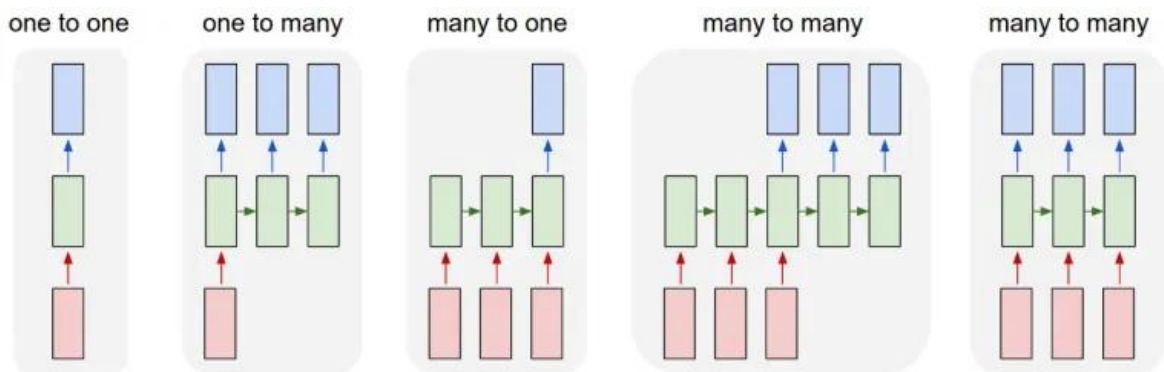
## 4: Mạng Recurrent Neural Network

### 4.1 Khái quát về mạng RNN

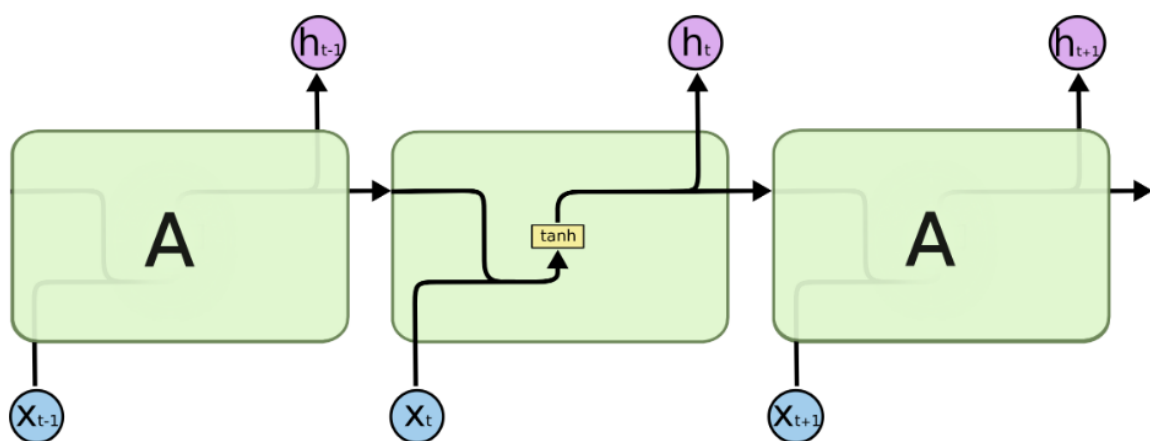
Ý tưởng chính của RNN (Recurrent Neural Network) là sử dụng chuỗi các thông tin. Trong các mạng nơ-ron truyền thống tất cả các đầu vào và cả đầu ra là độc lập với nhau. Tức là chúng không liên kết thành chuỗi với nhau. Nhưng các mô hình này không phù hợp trong rất nhiều bài toán. Ví dụ, nếu muốn đoán từ tiếp theo có thể xuất hiện trong một câu thì ta cũng cần biết các từ trước đó xuất hiện lần lượt thế nào chứ nhỉ? RNN được gọi là hồi quy (Recurrent) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó. Trên lý thuyết, RNN có thể sử dụng được thông tin của một văn bản rất dài, tuy nhiên thực tế thì nó chỉ có thể nhớ được một vài bước trước đó (ta cùng bàn cụ thể vấn đề này sau) mà thôi. Về cơ bản một mạng RNN có dạng như sau:



RNN được phân thành các loại chính:



- **One to one:** mẫu bài toán cho Neural Network (NN) và Convolutional Neural Network (CNN), 1 input và 1 output, ví dụ với CNN input là ảnh và output là ảnh được segment.
- **One to many:** bài toán có 1 input nhưng nhiều output, ví dụ: bài toán caption cho ảnh, input là 1 ảnh nhưng output là nhiều chữ mô tả cho ảnh đấy, dưới dạng một câu.
- **Many to one:** bài toán có nhiều input nhưng chỉ có 1 output, ví dụ bài toán phân loại hành động trong video, input là nhiều ảnh (frame) tách ra từ video, output là hành động trong video
- **Many to many:** bài toán có nhiều input và nhiều output, ví dụ bài toán dịch từ tiếng anh sang tiếng việt, input là 1 câu gồm nhiều chữ: “I love Vietnam” và output cũng là 1 câu gồm nhiều chữ “Tôi yêu Việt Nam”.



Cấu trúc RNN

## 4.2 Điểm khác biệt giữa RNN và MLP

RNN là một kiến trúc neural network phù hợp temporal data. RNN phù hợp với các tác vụ liên quan đến time-series data, brain-wave data, hoặc stock market data. Để làm được điểm này, RNN sử dụng state matrices liên quan đến data đang xử lý. Nói cách khác, RNN sẽ sử dụng trạng thái trước đó (previous state) của dữ liệu đang xử lý. Nhìn chung, chúng ta có thể hình dung RNN như MLP kết hợp với state matrices để dự đoán.

## 4.3 Điểm khác biệt giữa RNN và LSTM

- Cấu trúc RNN đơn giản hơn LSTM vì chỉ có một tầng là hàm Tanh
- LSTM có bốn tầng: Ba tầng là hàm Sigmoid 1 tầng là hàm Tanh
- LSTM chậm hơn so với RNN

# 5: Mạng Long Short Term Memory Network

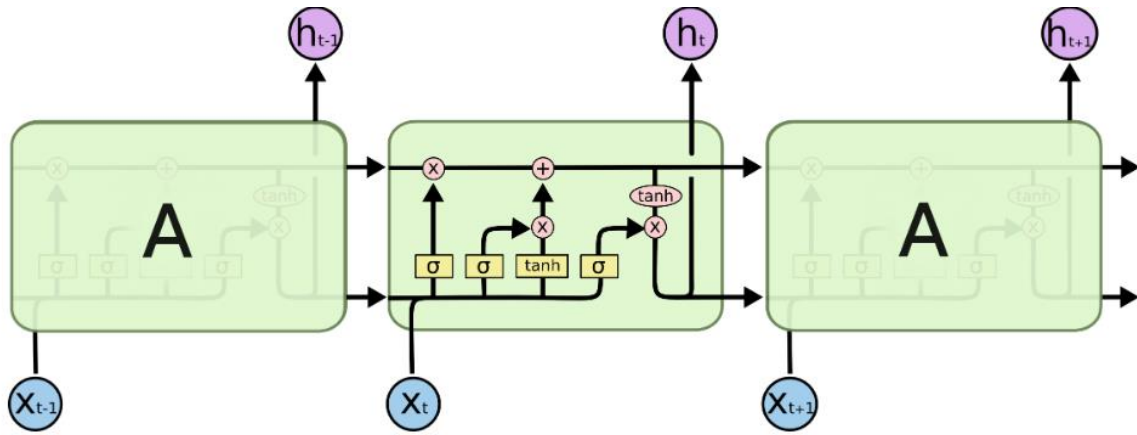
## 5.1: Khái quát về mạng LSTM

Mạng LSTM (Long Short Term Memory network) là một dạng đặc biệt của RNN, có khả năng học được các phụ thuộc xa.

LSTM được thiết kế tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính của LSTM, chứ không cần huấn luyện để có thể nhớ được nó.

Mạng hồi quy đều có dạng chuỗi các modul lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các module có cấu trúc rất đơn giản, thường là một tầng Tanh

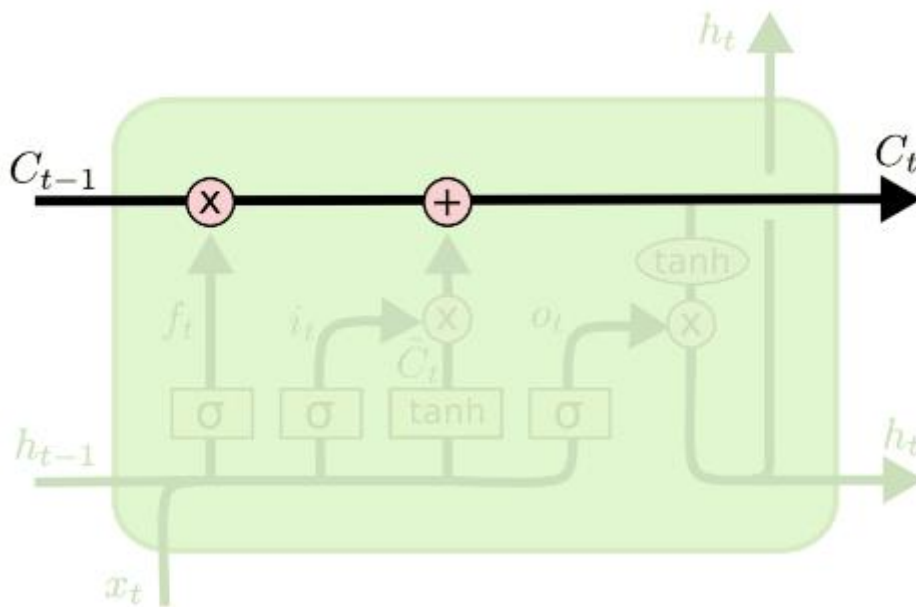
LSTM cũng có kiến trúc dạng chuỗi nhưng mô-đun của nó khác với cấu trúc mạng RNN. Thay vì có 1 tầng thì LSTM có 4 tầng tương tác với nhau 1 cách đặc biệt



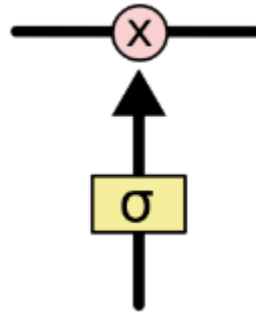
Hình 3: Cấu trúc mạng LSTM

### 5.2: Cách thức hoạt động mô hình LSTM

Ý tưởng của LSTM là thành phần ô trạng thái (cell state) được thể hiện qua đường chạy ngang qua đỉnh của đồ thị (Hình 4)



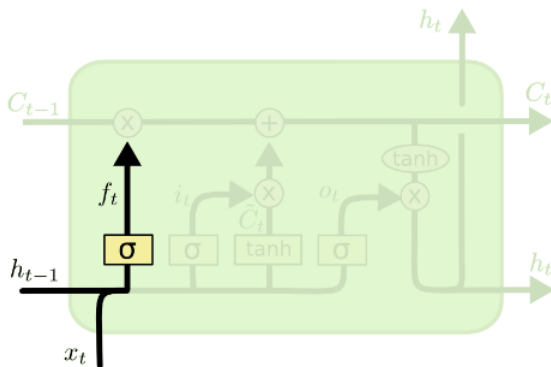
Đường đi của cell state trong LSTM



Công của hàm sigmoid trong LSTM

### Thứ tự các bước của LSTM

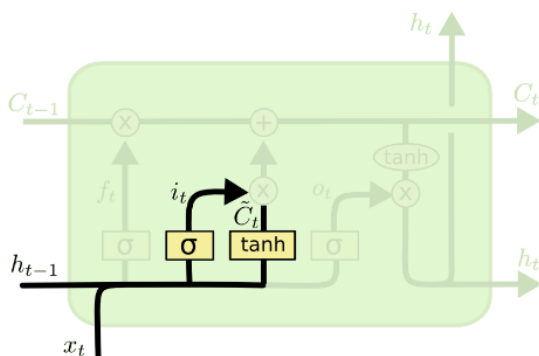
Bước 1: LSTM quyết định thông tin sẽ cho phép đi qua ô trạng thái (cell state).  
Kiểm soát bởi hàm sigmoid trong Forget gate layer)



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

Forget gate layer

Tiếp quyết định thông tin được lưu trữ trong ô trạng thái (cell state). Gồm 2 phần, một tầng ẩn của hàm sigmoid là tầng input gate layer quyết định giá trị sẽ được cập nhật. Tầng ẩn hàm tanh tạo ra 1 vectơ của một giá mà có thể được thêm vào trạng thái mới. Kết hợp 2 tầng tạo thành 1 cập nhật cho trạng thái

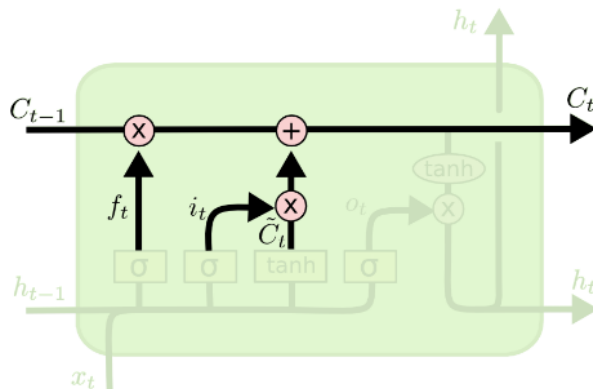


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

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

Cập nhật giá trị trạng thái kết hợp từ input gate layer và tầng ẩn hàm tanh

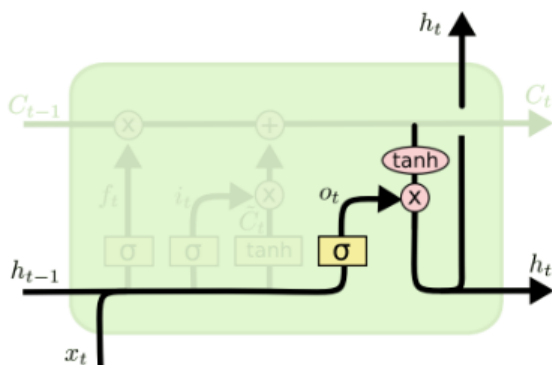
Nhân trạng thái cũ  $f_t$  tương ứng. Phần tử đề cử  $i_t * C_t$  là một giá trị được tính toán tương ứng với bao nhiêu giá trị cập nhật mỗi trạng thái



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

Ô trạng thái (cell state) mới

Cuối cùng cần quyết định đầu ra sẽ trả về như nào. Kết quả đầu ra sẽ dựa trên ô trạng thái. Đầu tiên cho chạy qua một tầng sigmoid nói quyết định của cell state sẽ ở đầu ra. Sau đó, cell state đưa qua hàm tanh (chuyển về giá trị -1 đến 1) và nhân với đầu ra của một cổng sigmoid



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

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

Điều chỉnh thông tin ở đầu ra thông qua hàm tanh

### III Thực nghiệm

1: Giới thiệu về dữ liệu và Kỹ thuật tiền xử lý dữ liệu

**Dataset: Water-Spring-Amiata.csv**

Chuỗi các điểm dữ liệu được đo theo từng khoảng thời gian liên nhau và khoảng cách giữa các lần bằng nhau, gọi là chuỗi thời gian (Time series). Nhiệt độ của

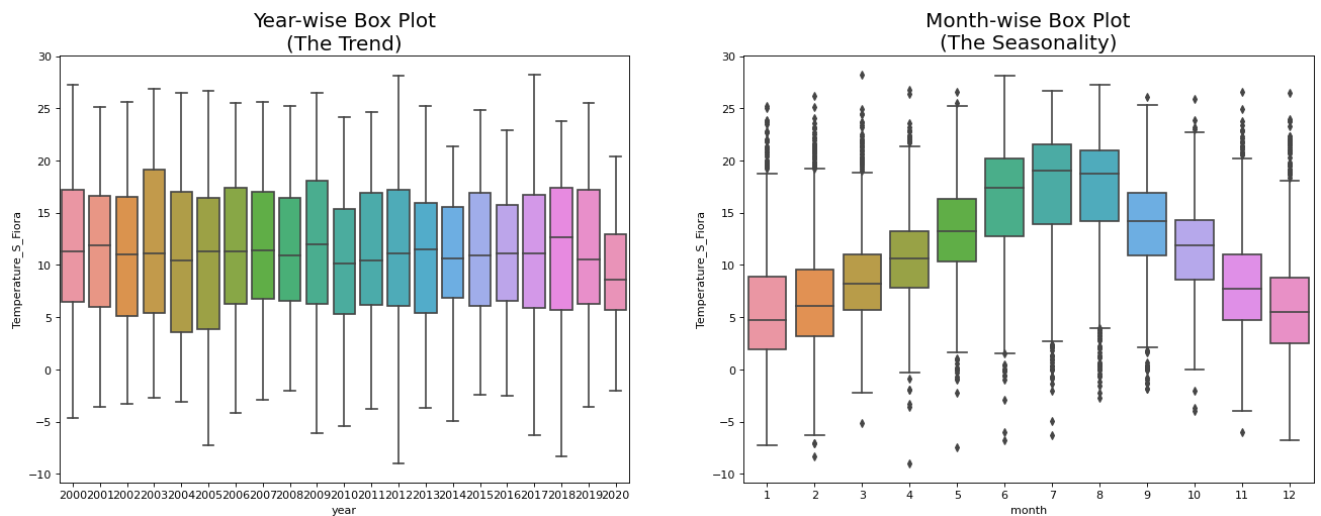
Fiora trong mỗi ngày từ những năm 2000 - 2020 được thể hiện rõ ở tập data ở cột Temperature\_S\_Fiora.

Date	Temperature_S_Fiora
01/01/2000	-0,15
02/01/2000	1,55
03/01/2000	2,70
04/01/2000	4,00
05/01/2000	4,60

Năm dòng đầu tiên của Dataset

Đề dữ báo chuỗi thời gian cho tương lai cso rất nhiều phương pháp để tiên xử lí dữ liệu của dataset, thương ta sẽ sử dụng phương pháp “Sliding window” để tiên xử lí dữ liệu.

Biểu đồ biểu diễn phân bố dữ liệu theo năm và tháng của tập dữ liệu



Tái tạo cấu trúc tập dữ liệu chuỗi thời gian trên dưới dạng học có giám sát bằng cách sử dụng giá trị ở bước thời gian trước đó để dự đoán giá trị thời gian tiếp theo hoặc là biến đổi thành thời gian của quá khứ. Khi tổ chức lại chuỗi thời gian theo cách trên ta sẽ được

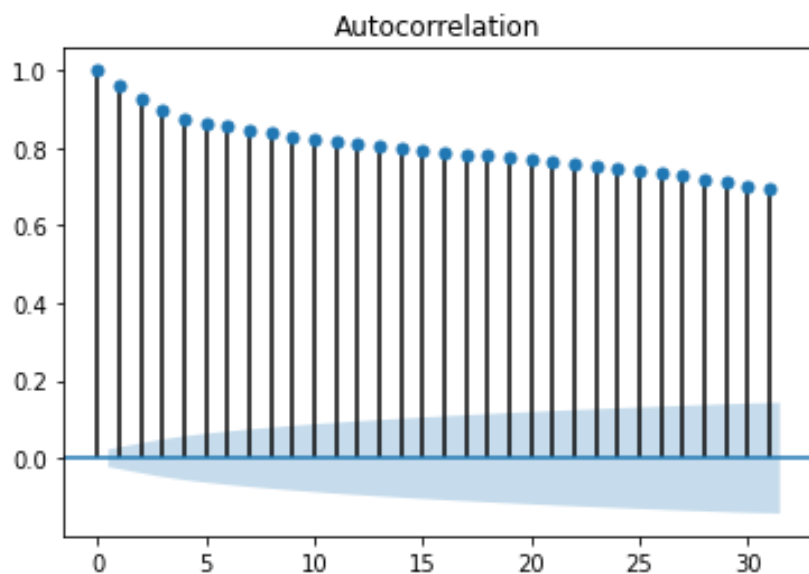
Temperature_S_Fiora	Temperature_S_Fiora (t +1)
-0,15	1,55
1,55	2,70
2,70	4,00
4,00	4,60
4,60	NaN

Biến đổi ở trên và so sánh với chuỗi ban đầu quan sát được:

- Bước thời gian đầu vào là **Temperature\_S\_Fiora** và bước thời gian tiếp theo là đầu ra là **Temperature\_S\_Fiora(t+1)**
- Thứ tự giữa các quan sát được giữ nguyên và tiếp tục giữ nguyên khi sử dụng tập dữ liệu để huấn luyện mô hình, nhưng đối với dữ liệu trên thì ta sẽ tái cấu trúc nó về quá khứ để dự đoán cho hiện tại và tương lai

.....(t-2)	Temperature_S_Fiora(t-1)	Temperature_S_Fiora	Temperature_S_Fiora (t +1)
	....	-0,15	1,55
	-0,15	1,55	2,70
	1,55	2,70	4,00
	2,70	4,00	4,60
	4,00	4,60	....

Biểu đồ biểu thị sự tương quan độ trễ của tập dữ liệu



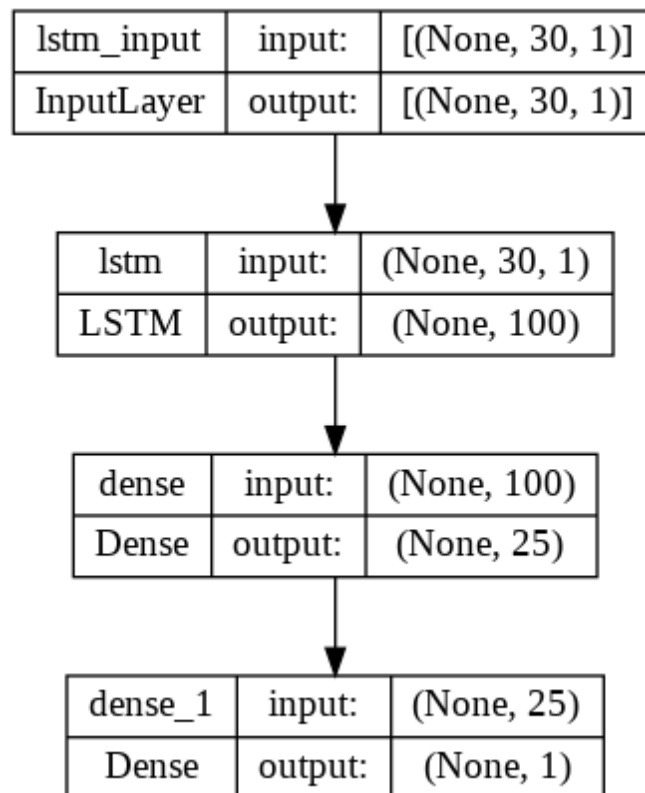
## 2: Thiết lập tham số cho mô hình

Với bộ dữ liệu trên khi tiền xử lý dữ liệu xong sẽ áp dụng với vài mô hình dữ liệu đặc biệt với đề tài này ta sẽ áp dụng hai mô hình Neural Network vào tập dữ liệu để dự đoán.



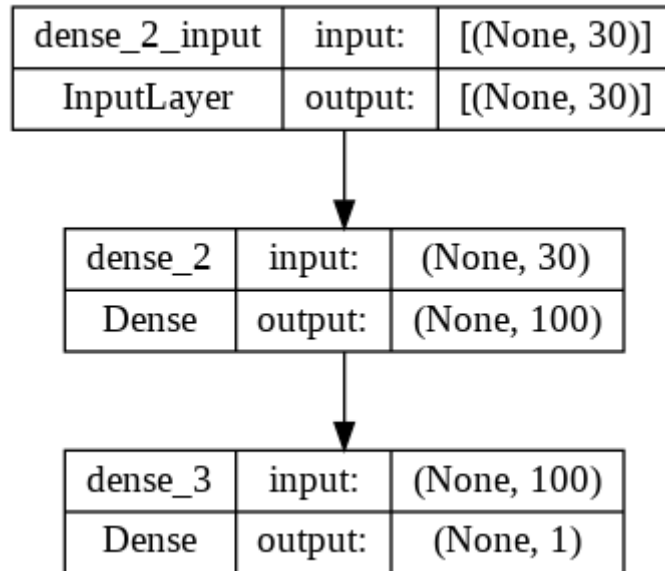
## 2.1: Long Short Term Memory Model

Chúng tôi xác định LSTM với 100 nơ-ron trong lớp ẩn đầu tiên và sử dụng 25 nơ-ron cho lớp ẩn thứ hai. Sử dụng 1 nơ-ron trong lớp đầu ra để dự đoán nhiệt độ. Chúng tôi sẽ sử dụng hàm loss MSE(mean squared error) là hàm loss được sử dụng phổ biến và rộng rãi cho các mô hình hồi quy. Mô hình sẽ được fit với số epochs bằng 300 và số batch size bằng 500



## 2.2: Multi-layer Perceptron

Mô hình Multilayer Perceptron, viết tắt là MLP, là một mô hình mạng nơ-ron được kết nối đầy đủ tiêu chuẩn. Mô hình của chúng tôi được tạo bởi 2 lớp dense với lớp dense đầu tiên chúng tôi sử dụng 100 nơ-ron và lớp dense thứ 2 sử dụng 1 nơ-ron với tác dụng như 1 lớp đầu ra.



### 3: Kết quả và nhận xét

#### 3.1: So sánh 2 mô hình dựa vào ba chỉ số MSE, MAE, R2

##### 3.1.1 So sánh trên tập train

Dataset	Mô hình	MSE	MAE	R2
Temperature_S_Fiora	LSTM	0.78	0.64	0.98
	MLP	2.46	1.16	0.94

So sánh ba giá trị MSE, MAE, R2 trước khi tối ưu mô hình

Nhận xét: Dựa vào ba chỉ số MSE, MAE, R2 thì ta có thể thấy LSTM vượt trội hơn hẳn so với MLP khi nó có chỉ số MSE và MAE thấp hơn còn R2 thì cao hơn

##### 3.1.2 So sánh trên tập test

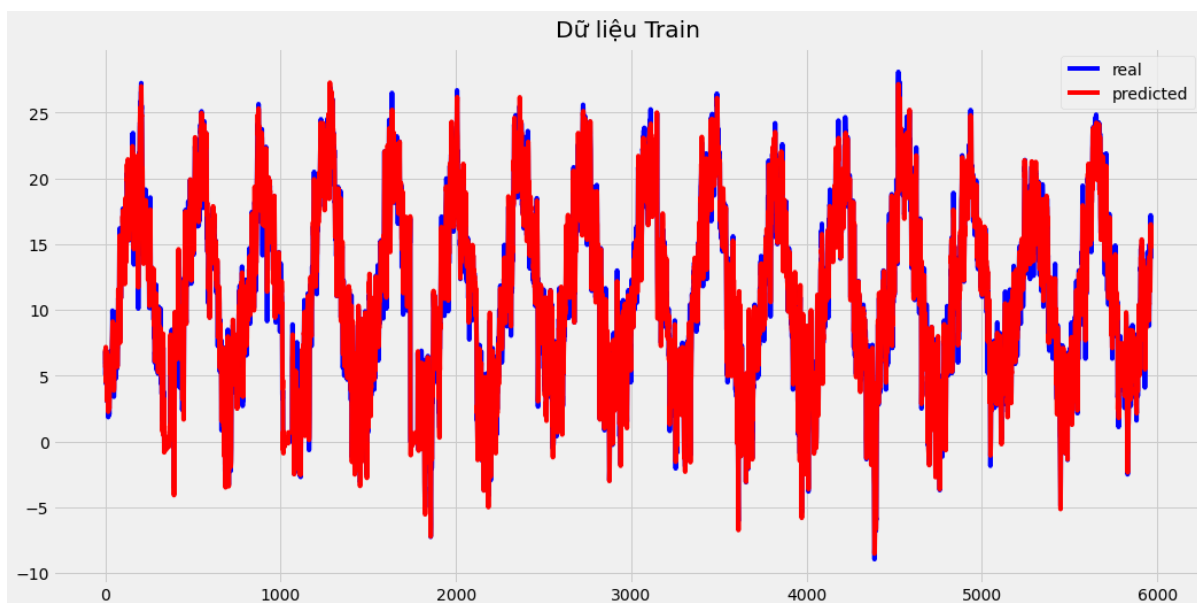
Dataset	Mô hình	MSE	MAE	R2
Temperature_S_Fiora	LSTM	5.72	1.77	0.86
	MLP	4.11	1.49	0.90

So sánh ba giá trị MSE, MAE, R2 trước khi tối ưu mô hình

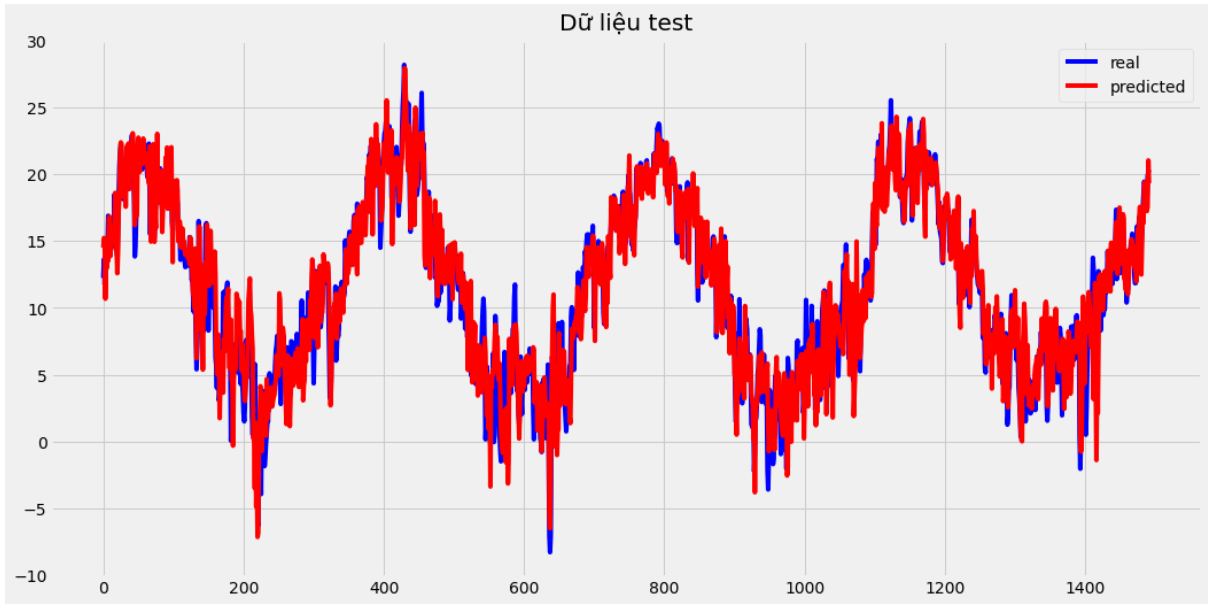
Nhận xét: Còn trong tập test thì MLP lại tỏ ra vượt trội hơn hẳn so với LSTM khi nó có chỉ số MSE và MAE thấp hơn còn R2 thì cao hơn LSTM

### 3.2: So sánh kết dự đoán và kết quả thực tế cho cả tập train và tập test

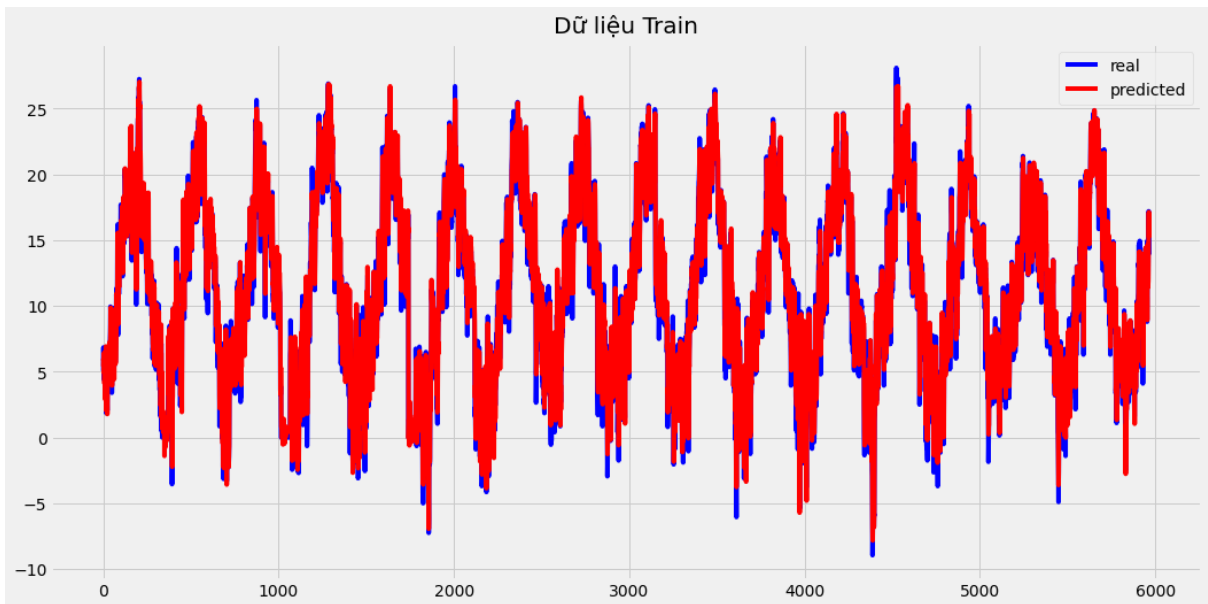
Vì đã có sự so sánh hàm loss trước đây nên chúng tôi tiến hành dự đoán ở cả tập train và tập test bằng mô hình LSTM và MLP. Quan sát hình ảnh dưới đây, có thể thấy được cả hai mô hình đã hoạt động rất hiệu quả và có kết quả dự đoán rất đáng kinh ngạc.



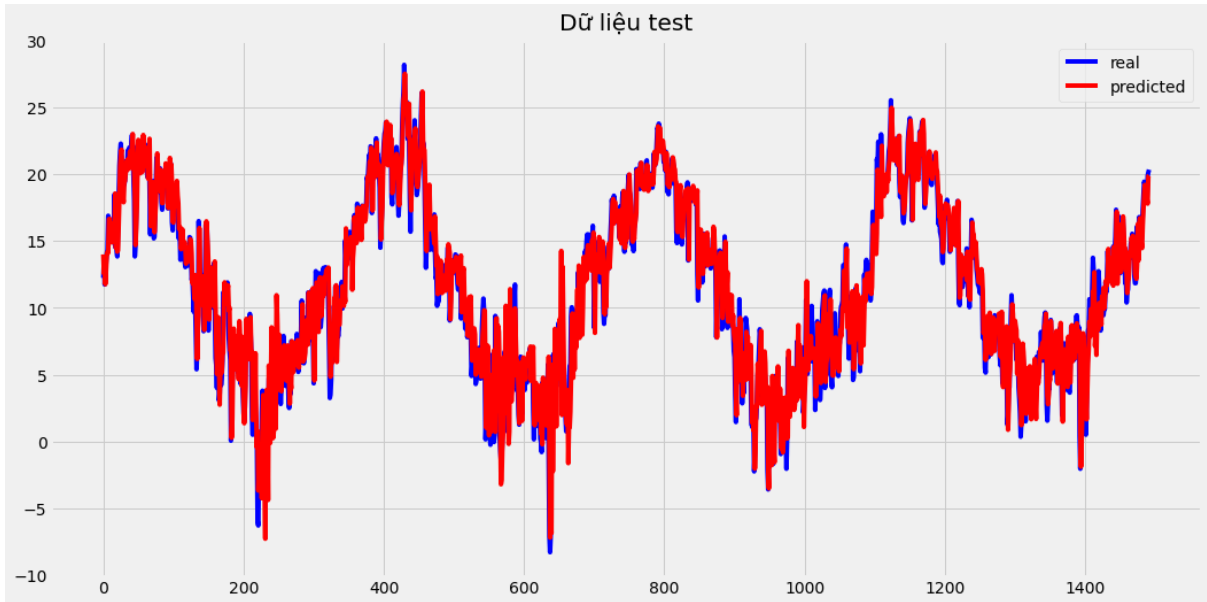
So sánh giá trị thực tế và giá trị dự đoán của LSTM ở tập train



So sánh giá trị thực tế và giá trị dự đoán của LSTM ở tập test



So sánh giá trị thực tế và giá trị dự đoán của MLP ở tập train



So sánh giá trị thực tế và giá trị dự đoán của MLP ở tập test

### 3.3: Tối ưu hóa tham số mô hình

Chúng tôi sẽ đi tìm các giá trị tối ưu cho các siêu tham số của một mô hình bằng cách sử dụng GridSearchCV. GridSearchCV là quá trình thực hiện điều chỉnh siêu tham số để xác định các giá trị tối ưu cho một mô hình nhất định. Chúng tôi tiến hành tối ưu các tham số là Epochs và Batch\_size nhằm để hoàn thiện hơn về mô hình. Với kết quả đầu ra thì chúng tôi nên giảm số lượng Epochs xuống mức 100 và giữ nguyên Batch\_size là 500.

### 3.4: So sánh hai mô hình sau khi được tối ưu hóa dựa vào ba chỉ số MSE, MAE, R2

#### 3.4.1 So sánh trên tập train

Dataset	Mô hình	MSE	MAE	R2
Temperature_S_Fiora	LSTM	2.89	1.28	0.93
	MLP	2.91	1.28	0.93

So sánh ba giá trị MSE, MAE, R2 sau khi tối ưu mô hình

Nhận xét: Trong tập train việc thay đổi một số chỉ số như epoch để tối ưu hóa mô hình không những không làm cho mô hình tốt lên mà thậm chí nó còn làm cho mô hình tệ đi ở cả 3 chỉ số. Trong đó LSTM vẫn tỏ ra nhỉnh hơn đôi chút so với MLP ở chỉ số MSE còn MAE và R2 thì gần như tương đương với nhau.

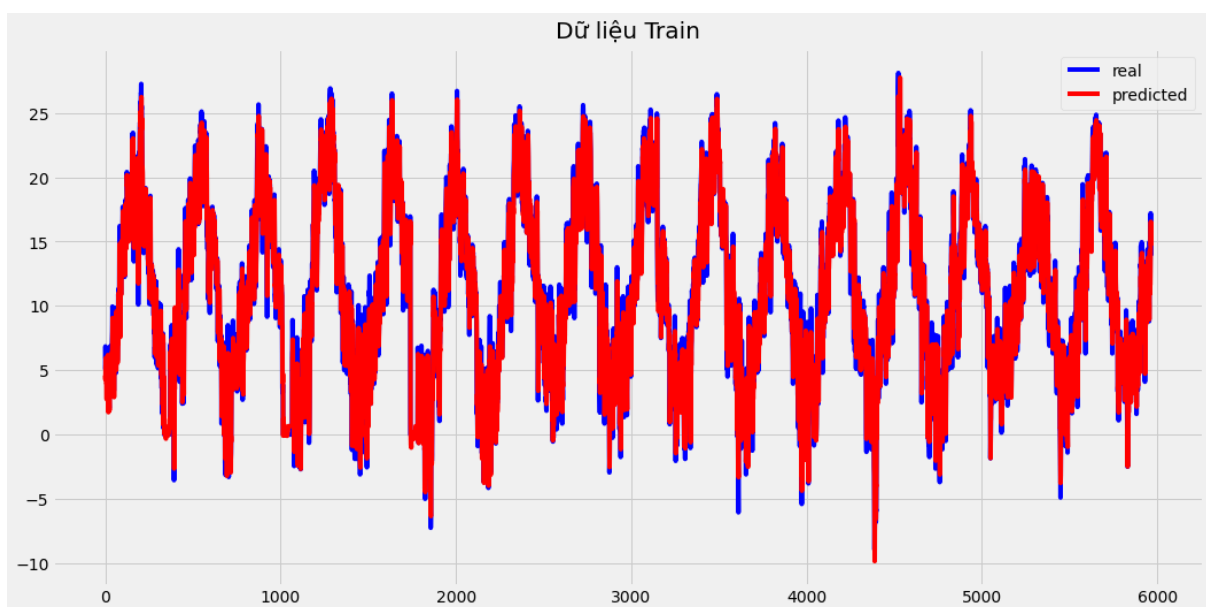
### 3.4.2 So sánh trên tập test

Dataset	Mô hình	MSE	MAE	R2
Temperature_S_Fiora	LSTM	3.87	1.40	0.90
	MLP	3.26	1.36	0.92

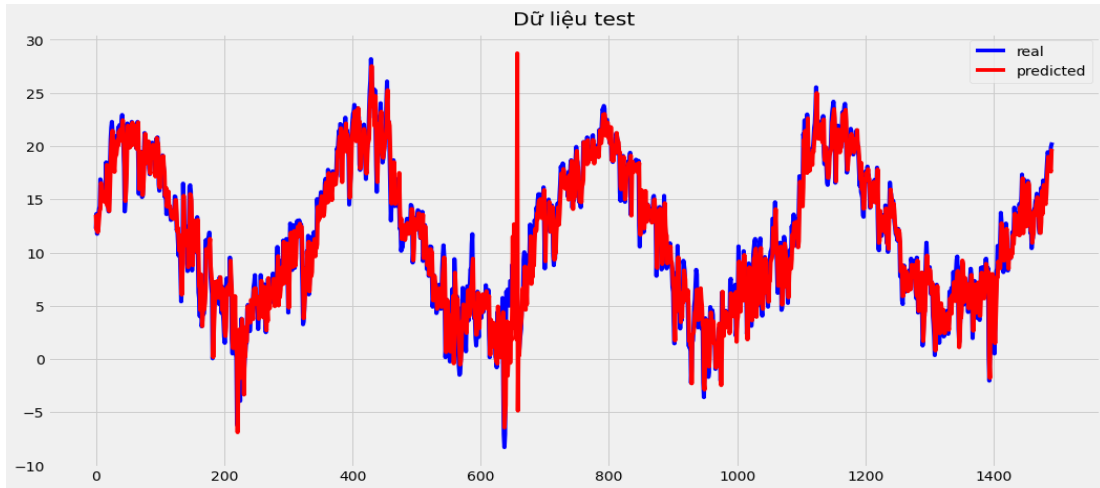
So sánh ba giá trị MSE, MAE, R2 sau khi tối ưu mô hình

Nhận xét: Ngược lại trong tập test thì việc tối ưu hóa lại trở nên rất hiệu quả đối với cả hai mô hình LSTM và MLP khi mà các chỉ số đều tốt hơn so với lúc chưa được tối ưu hóa tham số. Cụ thể trong tập test thì MLP vẫn tỏ ra vượt trội hơn hẳn so với LSTM ở cả ba chỉ số.

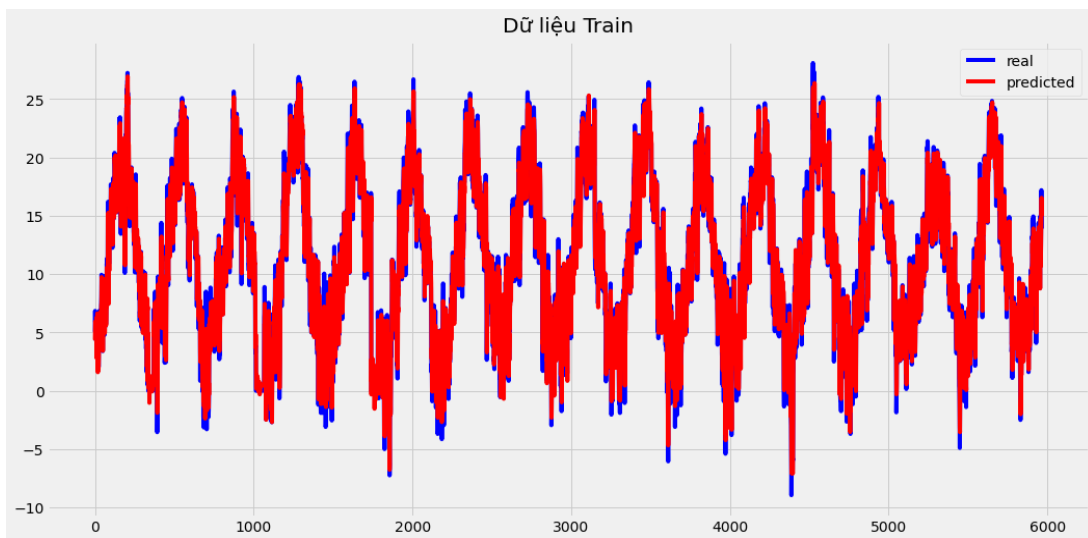
### 3.5: So sánh kết dự đoán và kết quả thực tế cho cả tập train và tập test sau khi mô hình được tối ưu



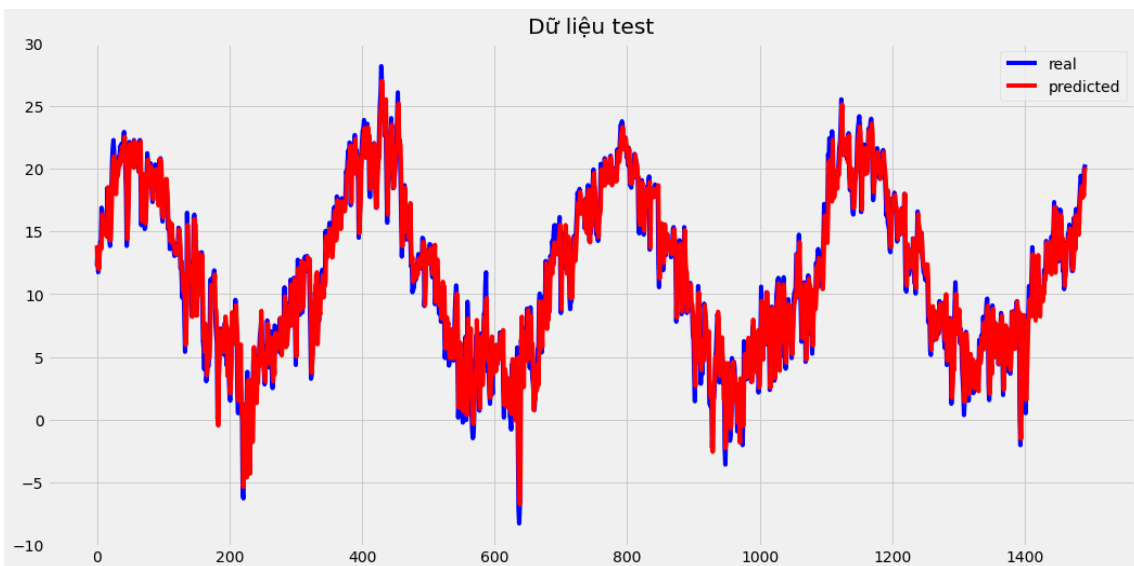
So sánh giá trị thực tế và giá trị dự đoán của LSTM ở tập train



So sánh giá trị thực tế và giá trị dự đoán của LSTM ở tập test



So sánh giá trị thực tế và giá trị dự đoán của MLP ở tập train



So sánh giá trị thực tế và giá trị dự đoán của MLP ở tập test

Nhận xét: Các biểu đồ cho thấy sau khi tối ưu hóa tham số mô hình xong kết quả không có quá nhiều khác biệt so với lúc chưa tối ưu hóa nhưng cả hai mô hình vẫn dự đoán rất chính xác kết quả ở cả tập train và tập test.

## IV Kết luận

- Đề tài trên nhóm đã tìm hiểu về time series sử dụng mô hình Neural Network cũng chính là hai mô hình mạng MLP (Multi-layer Perceptron) và LSTM (Long Short Term Memory Model). Nhóm đã tìm hiểu về cơ chế hoạt động của 2 mạng nơ-ron đó.
- Sử dụng hai mô hình mạng cho một tập dữ liệu liên quan đến time series cho thấy kết quả dự đoán của hai mô hình gần tương đương nhau nhưng vẫn có đôi chút sự chênh lệch, đối với bộ dữ liệu được sử dụng ở trên thì ta có thể thấy trong tập test mô hình mạng MLP dự đoán chính xác hơn đôi chút so với mô hình mạng LSTM, ngược lại trong tập train mô hình mạng LSTM lại tỏ ra nổi bật hơn hẳn so với mô hình mạng MLP.
- Nhóm cũng đã tìm hiểu thêm về cách tối ưu tham số cho 2 mô hình mạng và cũng đã áp dụng vào bài toán trên cho thấy kết quả không khả thi lắm. Có thể bài toán trên cách tối ưu chưa thực sự cần thiết hoặc là nhóm đang nhầm lẫn ở cách hoạt động khiến hai mô hình mạng MLP và LSTM dự đoán cho kết quả không như mong đợi
- Việc sử dụng Neural Network vào bài toán dự báo chuỗi thời gian cho tương lai cũng rất phù hợp vì độ chính xác của dự báo được tăng lên so với mô hình Machine Learning.



## V Tài liệu tham khảo

<https://dominhhai.github.io/vi/2017/10/what-is-rnn/>

<https://nttuan8.com/bai-13-recurrent-neural-network/>

<https://viblo.asia/p/recurrent-neural-networkphan-1-tong-quan-va-ung-dung-jvElaB4m5kw>

<https://viblo.asia/p/recurrent-neural-network-tu-rnn-den-lstm-gGJ597z1ZX2>

