

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----



**PHẠM CHÍ HÙNG**

**NGHIÊN CỨU CÔNG NGHỆ LSTM  
VÀ GIẢI PHÁP CHO BÀI TOÁN DỰ ĐOÁN  
LƯỢNG HÀNH KHÁCH ĐI MÁY BAY**

**Chuyên ngành: Hệ thống Thông tin**

**Mã số: 8.48.01.04**

**TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT**  
*(Theo định hướng ứng dụng)*

**HÀ NỘI - 2019**

Luận văn được hoàn thành tại:

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

Người hướng dẫn khoa học: **TS. NGUYỄN VĂN THỦY**

Phản biện 1: .....

Phản biện 2: .....

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ... giờ ... ngày ... tháng ... năm .....

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông

## MỞ ĐẦU

### 1. Tính cấp thiết của đề tài

Ngày nay, ngành hàng không là một ngành công nghiệp vô cùng phát triển, nó phục vụ nhu cầu đi lại, giao thương giữa các quốc gia, cùng với đó là sự phát triển kinh tế của các nước có đường bay quốc tế. Theo dự đoán, vào năm 2035, lượng hành khách sử dụng dịch vụ hàng không sẽ đạt 7,2 tỷ lượt, tăng gần gấp đôi con số 3,8 tỷ trong năm nay (số liệu do Hiệp hội Vận tải Hàng không Quốc tế (IATA) cung cấp). Ông Alexandre De Juniac, Tổng Giám đốc điều hành của IATA nhận xét nhu cầu đi lại bằng đường hàng không trong hai thập kỷ tới sẽ gấp đôi. Cũng theo dự báo trên, IATA cho rằng khu vực châu Á - Thái Bình Dương sẽ là nơi có nhu cầu di chuyển bằng đường không cao nhất thế giới. Để đáp ứng nhu cầu phục vụ hành khách một cách tốt nhất, việc phải có một hệ thống dự đoán lượng hành khách đi máy bay là rất cần thiết.

### 2. Tổng quan vấn đề nghiên cứu

Theo dự báo của IATA, ông Alexandre De Juniac đã đưa ra ba kịch bản dự báo về lĩnh vực hàng không trong giai đoạn 20 năm tới. Kịch bản thứ nhất dự báo tăng gấp đôi lượng hành khách. Kịch bản thứ hai đưa ra nhịp độ tăng trưởng hành khách hàng không gần gấp ba lần so với năm 2016. Kịch bản cuối cùng ước tính 7,2 tỷ lượt khách sử dụng dịch vụ hàng không vào năm 2035.

Cùng với đó, thị trường hàng không Trung Quốc sẽ thay thế Mỹ trở thành thị trường hàng không lớn nhất thế giới (tính cả đường bay nội địa và quốc tế) vào năm 2029. Ấn Độ cũng sẽ thay nước Anh chiếm vị trí thứ ba vào năm 2026, trong khi Indonesia sẽ lọt vào top 10 trong danh sách của IATA.

Nhận thấy nhu cầu quan trọng của việc dự đoán lượng hành khách có nhu cầu đi lại bằng đường hàng không, tôi đề xuất một phương pháp sử dụng công nghệ LSTM để dự đoán lượng hành khách đi máy bay quốc tế.

### 3. Mục đích nghiên cứu

- Nghiên cứu về vấn đề dự báo chuỗi thời gian, áp dụng dự đoán lượng hành khách đi máy bay quốc tế.
- Nghiên cứu và ứng dụng công nghệ LSTM.

### 4. Đối tượng và phạm vi nghiên cứu

#### 4.1. Đối tượng nghiên cứu

- Công nghệ LSTM (Long Short-Term Memory).
- Vấn đề dự báo lượng hành khách đường bay quốc tế.

#### **4.2. Phạm vi nghiên cứu**

- Giới hạn nghiên cứu về công nghệ LSTM (Long Short-Term Memory).
- Nghiên cứu bài toán dự đoán chuỗi theo thời gian.

### **5. Phương pháp nghiên cứu**

#### **5.1. Phương pháp nghiên cứu lý thuyết**

- Đọc và phân tích tài liệu về công nghệ LSTM, nghiên cứu vấn đề dự đoán chuỗi thời gian thực.

#### **5.2. Phương pháp thực nghiệm**

- Xây dựng ứng dụng xem xét vấn đề dự đoán lượng hành khách quốc tế.
- Thử nghiệm, đánh giá kết quả.

# CHƯƠNG 1

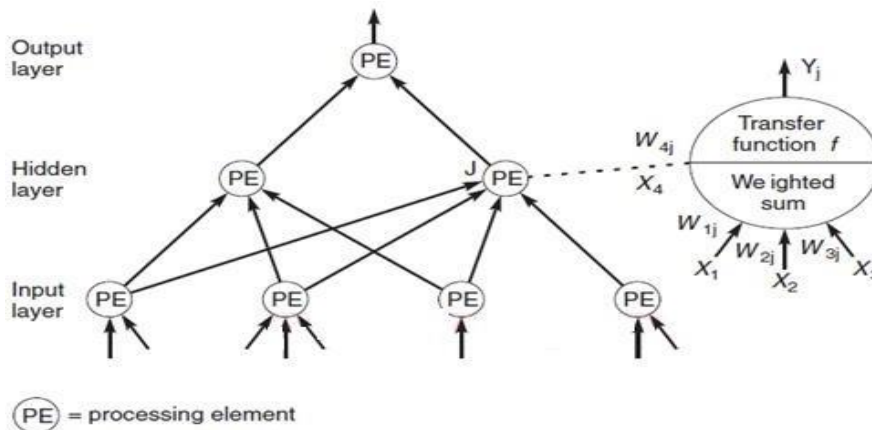
## TỔNG QUAN MẠNG NƠ-RON HỒI QUY

Trong chương này tôi sẽ giới thiệu về cơ sở lý thuyết về mạng nơ-ron nhân tạo, cách thức hoạt động của mạng nơ-ron, các phiên bản mở rộng của mạng nơ-ron nhân tạo.

### 1.1 Mạng nơ-ron nhân tạo

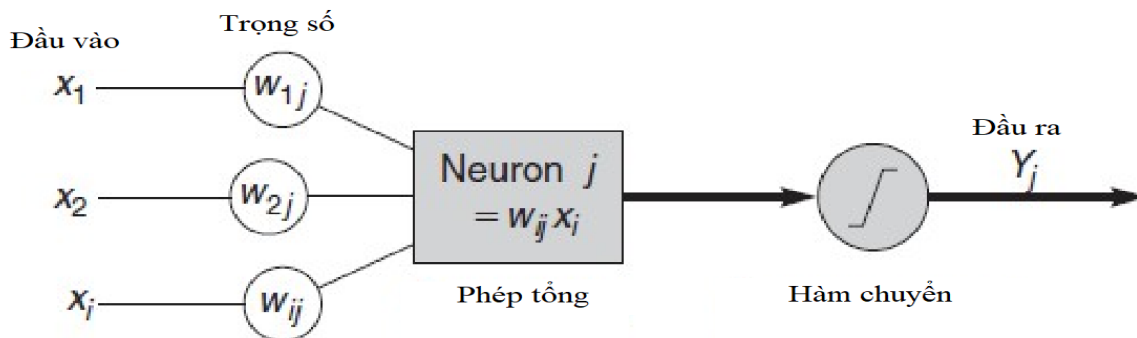
#### 1.1.1 Kiến trúc mạng nơ-ron nhân tạo

Mạng nơ-ron nhân tạo (Artificial Neural Network – ANN) là một mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh của sinh vật, bao gồm số lượng lớn các Nơ-ron được gắn kết để xử lý thông tin. ANN được giới thiệu năm 1943 bởi nhà thần kinh học Warren McCulloch và nhà logic học Walter Pits, ANN hoạt động giống như bộ não của con người, được học bởi kinh nghiệm (thông qua việc huấn luyện), có khả năng lưu giữ các tri thức và sử dụng các tri thức đó trong việc dự đoán các dữ liệu chưa biết.



**Hình 1.1 Kiến trúc mạng nơ-ron nhân tạo**

#### 1.1.2 Hoạt động của mạng nơ-ron nhân tạo



**Hình 1.2 Quá trình xử lý thông tin của một mạng nơ-ron nhân tạo.**

Hoạt động của mạng nơ-ron nhân tạo được thể hiện ở hình 1.2 với 3 chu trình:

Hàm tổng: Tính tổng trọng số của tất cả các đầu vào được đưa vào mỗi nơ-ron. Hàm tổng của một nơ-ron đối với  $n$  đầu vào được tính theo công thức sau:

$$Y = \sum_{i=1}^n X_i W_i \quad (1)$$

Việc lựa chọn hàm chuyển đổi có tác động lớn đến kết quả đầu ra của mạng ANN. Hàm chuyển đổi phi tuyến được sử dụng phổ biến trong mạng ANN là *sigmoid* hoặc *tanh*.

Hàm Sigmoid: 
$$f(z) = \frac{1}{1 + \exp(-z)}$$

Hàm Tanh: 
$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2)$$

Trong đó, hàm *tanh* là phiên bản thay đổi tỉ lệ của *sigmoid*, tức là khoảng giá trị đầu ra của hàm chuyển đổi thuộc khoảng  $[-1, 1]$  thay vì  $[0, 1]$  nên chúng còn gọi là hàm chuẩn hóa.

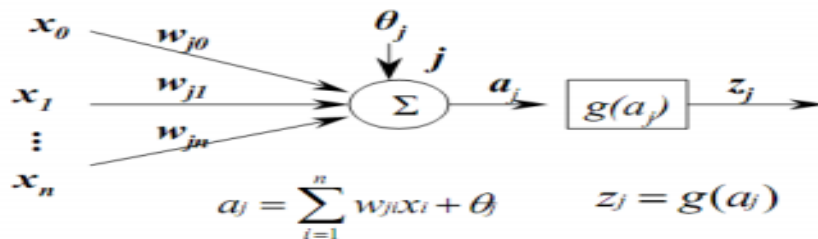
Áp dụng lan truyền ngược ta có các đại lượng:

$$\begin{aligned} \delta_3 &= y - \hat{y} \\ \delta_2 &= (1 - \tanh^2 z_1) * \delta_3 W_2^T \\ \frac{\partial L}{\partial W_2} &= a_1^T \delta_3 \\ \frac{\partial L}{\partial b_2} &= \delta_3 \\ \frac{\partial L}{\partial W_1} &= x^T \delta_2 \\ \frac{\partial L}{\partial b_1} &= \delta_2 \end{aligned} \quad (5)$$

## 1.2 Các thành phần cơ bản của mạng nơ-ron nhân tạo

### 1.2.1 Đơn vị xử lý

Còn được gọi là một nơ-ron hay một nút, thực hiện một công việc rất đơn giản: nó nhận tín hiệu vào từ các đơn vị phía trước hay một nguồn bên ngoài và sử dụng chúng để tính tín hiệu ra sẽ được lan truyền sang các đơn vị khác.



Hình 1.3 Đơn vị xử lý

Trong đó:

$x_i$  : các đầu vào

$w_{ji}$  : các trọng số tương ứng với các đầu vào

$\theta_j$  : độ lệch

$a_j$  : đầu vào mạng

$z_j$  : đầu ra của nơon

$g(x)$ : hàm chuyển (hàm kích hoạt).

### 1.2.2 Hàm kết hợp

Mỗi một đơn vị trong một mạng kết hợp các giá trị đưa vào nó thông qua các liên kết với các đơn vị khác, sinh ra một giá trị gọi là đầu vào mạng. Hàm thực hiện nhiệm vụ này gọi là hàm kết hợp, được định nghĩa bởi một luật lan truyền cụ thể. Trong phần lớn các mạng nơon, chúng ta giả sử rằng mỗi một đơn vị cung cấp một bộ cộng như là đầu vào cho đơn vị mà nó có liên kết. Tổng đầu vào đơn vị  $j$  đơn giản chỉ là tổng trọng số của các đầu ra riêng lẻ từ các đơn vị kết nối cộng thêm ngưỡng hay độ lệch  $\theta_j$ :

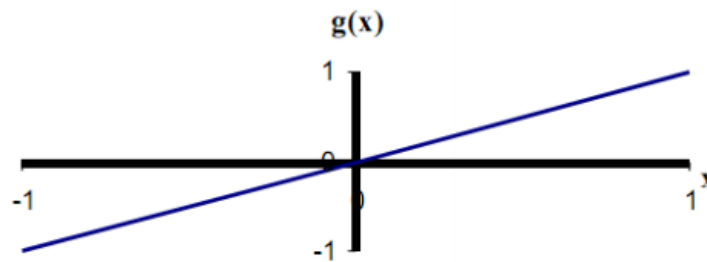
$$a_j = \sum_{i=1}^n w_{ji}x_i + \theta_j \quad (6)$$

### 1.2.3 Hàm kích hoạt

\* Hàm đồng nhất

$$g(x) = x \quad (8)$$

Nếu coi các đầu vào là một đơn vị thì chúng sẽ sử dụng hàm này. Đôi khi một hằng số được nhân với đầu vào mạng để tạo ra một hàm đồng nhất.



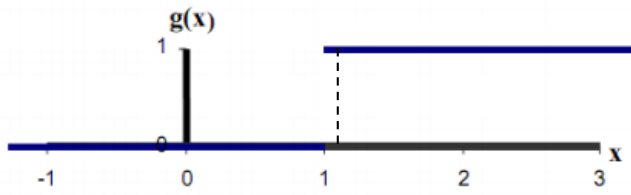
Hình 1.4 Hàm đồng nhất

\* Hàm bước nhị phân

Hàm này cũng được biết đến với tên "Hàm ngưỡng". Đầu ra của hàm này được giới hạn vào một trong hai giá trị:

$$g(x) = \begin{cases} 1, & \text{nếu } (x \geq \theta) \\ 0, & \text{nếu } (x < \theta) \end{cases}$$

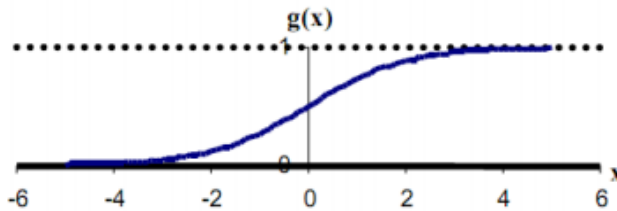
Dạng hàm này được sử dụng trong các mạng chỉ có một lớp. (9) Trong hình vẽ sau,  $\theta$  được chọn bằng 1.



Hình 1.5 Hàm bước nhị phân

\* Hàm sigmoid

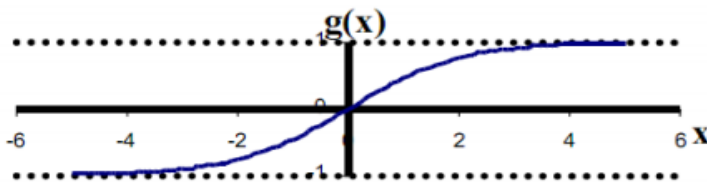
$$g(x) = \frac{1}{1 + e^{-x}} \quad (10)$$



Hình 1.6 Hàm Sigmoid

\* Hàm sigmoid lưỡng cực

$$g(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (11)$$



Hình 1.7 Hàm Sigmoid lưỡng cực

Hàm này có các thuộc tính tương tự hàm sigmoid. Nó làm việc tốt đối với các ứng dụng có đầu ra yêu cầu trong khoảng  $[-1, 1]$ .

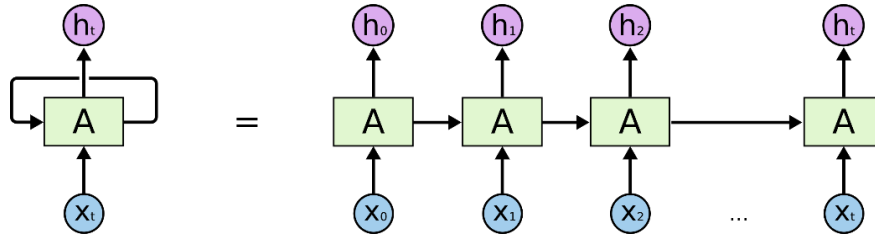
### 1.3 Mạng nơ-ron hồi quy

Mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) là một trong những mô hình học sâu được đánh giá có nhiều ưu điểm trong các tác vụ xử lý ngôn ngữ tự nhiên

#### 1.3.1 Khái niệm RNN

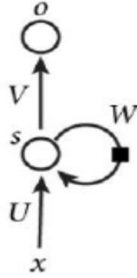
Ý tưởng của RNN đó là thiết kế một mạng nơ-ron sao cho có khả năng xử lý được thông tin dạng chuỗi, ví dụ một câu là một chuỗi gồm nhiều từ.





Hình 1.8 Mô hình mạng nơ-ron hồi quy

### 1.3.2 Quá trình xử lý thông tin của RNN



Hình 1.9 Quá trình xử lý thông tin trong RNNs

Quá trình này có thể được biểu diễn bằng mô hình toán sau (Mikolov *et al.*, 2014):

$$S_t = f(Ux_t + WS_{t-1})$$

### 1.3.3 Các ứng dụng của RNN

(12)

- **Mô hình ngôn ngữ và phát sinh văn bản**

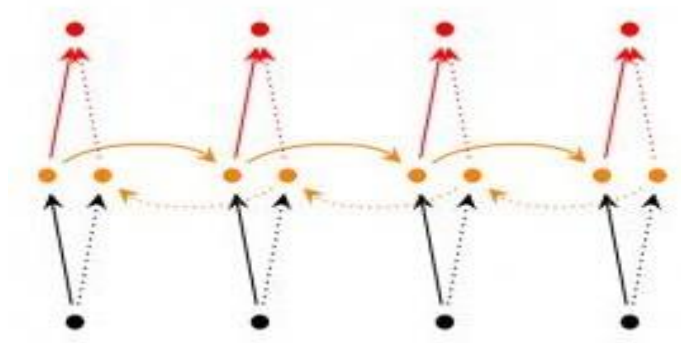
Mô hình ngôn ngữ cho ta biết xác suất của một câu trong một ngôn ngữ là bao nhiêu (ví dụ xác suất  $p(\text{"hôm qua là thứ năm"}) = 0.001$ ;  $p(\text{"năm thứ hôm là qua"}) = 0$ ). Đây cũng là bài toán dự đoán xác suất từ tiếp theo của một câu cho trước là bao nhiêu.

- **Dịch máy**
- **Phát sinh mô tả cho ảnh (Generating Image Descriptions)**

### 1.3.4 Các phiên bản mở rộng của RNN

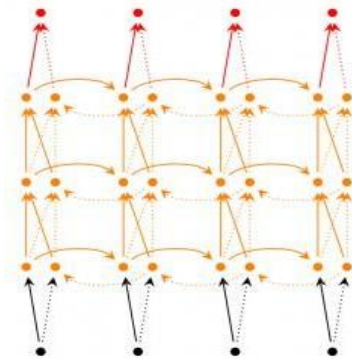
- **RNN hai chiều:**

Dựa trên ý tưởng đầu ra tại thời điểm  $t$  không chỉ phụ thuộc vào các thành phần trước đó mà còn phụ thuộc vào các thành phần trong tương lai. Ví dụ, để dự đoán một từ bị thiếu trong chuỗi, ta cần quan sát các từ bên trái và bên phải xung quanh từ đó. Mô hình này chỉ gồm hai RNNs nạp chồng lên nhau. Trong đó, các trạng thái ẩn được tính toán dựa trên cả hai thành phần bên trái và bên phải của mạng.



Hình 1.10 Mạng RNN hai chiều.

- RNN hai chiều sâu



Hình 1.11 Mạng RNN nhiều tầng.

- Mạng bộ nhớ ngắn hạn (LSTM)

## 1.4 Kết luận chương 1

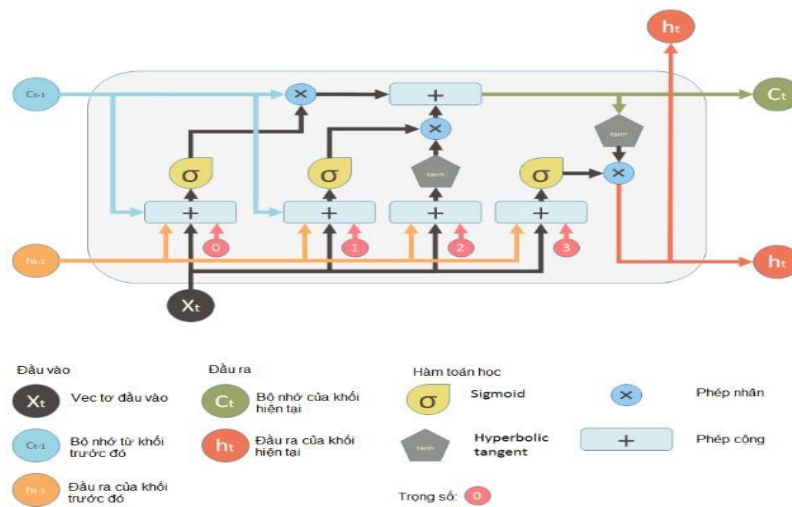
LSTM là một bước tiến lớn trong việc sử dụng RNN. Ý tưởng của nó giúp cho tất cả các bước của RNN có thể truy vấn được thông tin từ một tập thông tin lớn hơn, nó giúp giải quyết vấn đề dự đoán chuỗi thời gian. Cho nên trong đề án này chúng tôi tập trung nghiên cứu cho bài toán dự đoán hành khách lượng hành khách đi máy bay quốc tế. Chi tiết mô hình mạng này được giới thiệu trong Chương 2.

## CHƯƠNG 2

### ỨNG DỤNG CÔNG NGHỆ LSTM CHO VIỆC DỰ ĐOÁN LƯỢNG HÀNH KHÁCH ĐI MÁY BAY QUỐC TẾ

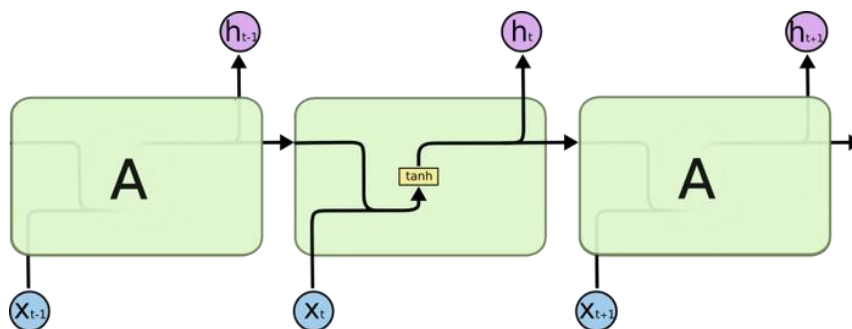
Chương này sẽ giới thiệu về bài toán ước lượng hành khách đi máy bay quốc tế và về mạng cải tiến LSTM: Kiến trúc, mô hình, quy trình hoạt động. Đây cũng là cơ sở để xây dựng thực nghiệm trong Chương 3.

#### 2.1 Kiến trúc mạng LSTM

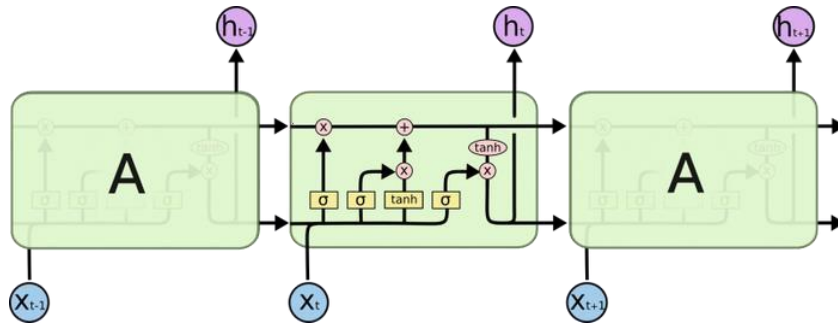


**Hình 2.1** Cấu trúc của mô hình LSTM

Trong hình 2.2, cấu trúc mạng LSTM gồm có 4 tầng và tương tác với nhau một cách đặc biệt. Cốt lõi của mạng LSTM bao gồm trạng thái nhớ và cổng. Trạng thái tế bào giống như băng chuyền, chạy xuyên suốt qua tất cả các nút mạng giúp thông tin được truyền đạt dễ dàng, còn cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid.



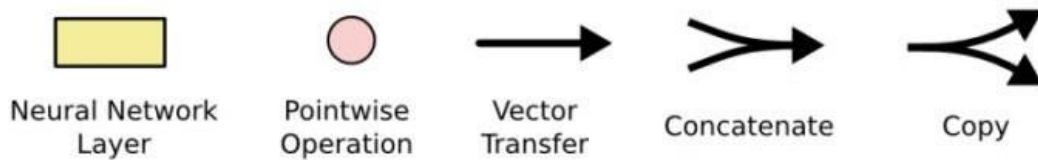
**Hình 2.2** Các mô-đun lặp của mạng RNN chứa một lớp



**Hình 2.3 Các mô-đun lặp của mạng LSTM chứa bốn lớp**

Trong đó, các ký hiệu sử dụng trong mạng LSTM được giải nghĩa sau đây:

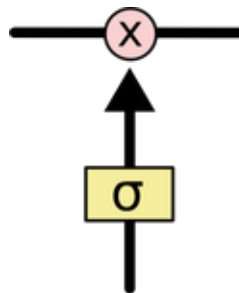
- Hình chữ nhật nền vàng là các lớp ẩn của mạng nơ-ron
- Hình tròn nền hồng biểu diễn toán tử theo từng điểm
- Đường kẻ gộp lại với nhau biểu thị phép nối các toán hạng
- Và đường rẽ nhánh biểu thị cho sự sao chép từ vị trí này sang vị trí khác



**Hình 2.4 Các kí hiệu sử dụng trong mạng LSTM**

## 2.2 Quá trình xử lý thông tin của LSTM

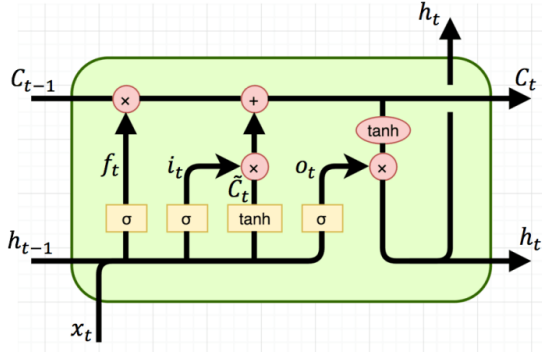
Mạng LSTM có khả năng thêm hoặc bớt thông tin vào trạng thái hạt nhân, được quy định một cách cẩn thận bởi các cấu trúc gọi là cổng. Các cổng này là một cách (tùy chọn) để định nghĩa thông tin băng qua. Chúng được tạo bởi hàm sigmoid và một toán tử nhân theo từng điểm.



**Hình 2.5 Cổng trạng thái LSTM.**

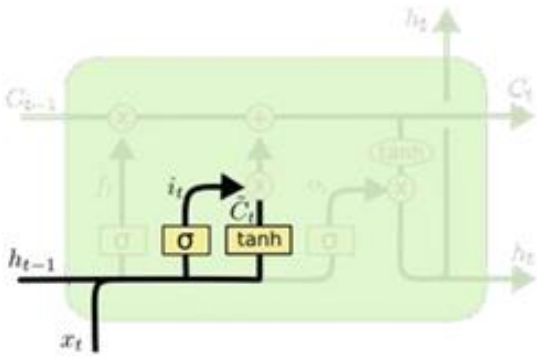
Hàm kích hoạt Sigmoid có giá trị từ 0 – 1, mô tả độ lớn thông tin được phép truyền qua tại mỗi lớp mạng. Nếu ta thu được 0, điều này có nghĩa là “không cho bất kỳ cái gì đi qua”, ngược lại nếu thu được giá trị là 1, thì có nghĩa là “cho phép mọi thứ đi qua”.

Một mạng LSTM gồm có 3 cổng để duy trì hoạt động trạng thái của hạt nhân.



**Hình 2.6 Bước thứ 1 quy trình xử lý của LSTM**

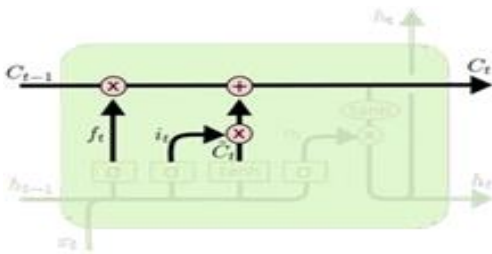
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (12)$$



**Hình 2.7 Bước thứ 2 quy trình xử lý của LSTM**

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \end{aligned} \quad (13)$$

Tiếp theo, chúng tôi sẽ kết hợp hai thành phần này lại để cập nhật vào trạng thái hạt nhân. Đây là giá trị mới, tỉ lệ số lượng giá trị mà chúng tôi muốn cập nhật  $f_t$  cho mỗi trạng thái.

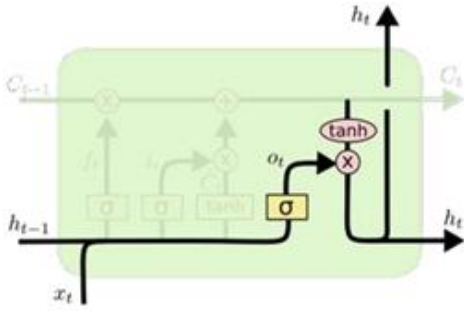


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (14)$$

**Hình 2.8 Bước thứ 3 quy trình xử lý của LSTM**

Cuối cùng, chúng tôi cần quyết định xem thông tin đầu ra là gì. Đầu ra này cần dựa trên trạng thái hạt nhân, nhưng sẽ được lọc bớt thông tin. Đầu tiên, áp dụng lớp sigmoid đơn để quyết định xem phần nào của trạng thái hạt nhân sẽ ra đầu ra. Sau đó, ta sẽ đẩy trạng thái hạt nhân qua (đẩy giá trị vào khoảng -1 và 1) và nhân với một cổng sigmoid đầu ra, để giữ lại

những phần ta muốn ra ngoài.



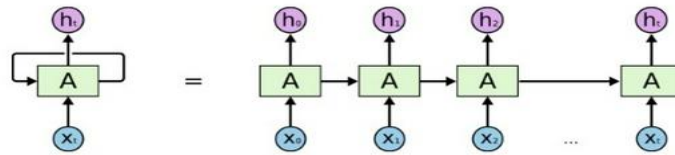
$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \quad (15)$$

**Hình 2.9 Bước cuối cùng quy trình xử lý của LSTM**

## 2.3 Các kỹ thuật LSTM sử dụng trong thử nghiệm

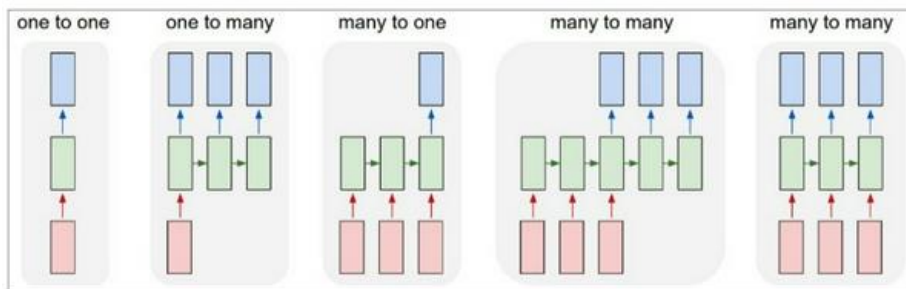
### 2.3.1 LSTM hồi quy

Thông thường dữ liệu của bài toán là tập dữ liệu hai cột: cột đầu tiên chứa số lượng hành khách (t) của tháng này và cột thứ hai chứa số lượng hành khách của tháng tới (t + 1), được dự đoán.



**Hình 2.10 Mạng nơ-ron hồi quy**

Trong hình 2.10, A là mạng nơ-ron hồi quy. Nó nhận một đầu vào  $x_t$ , tiến hành xử lý và đưa ra đầu ra  $h_t$ . Điểm đặc biệt của A là nó sẽ lưu lại giá trị của  $h_t$  để sử dụng cho đầu vào tiếp theo. Có thể coi một mạng neural hồi quy là một chuỗi những mạng con giống hệt nhau, mỗi mạng sẽ truyền thông tin nó vừa xử lý cho mạng phía sau nó.

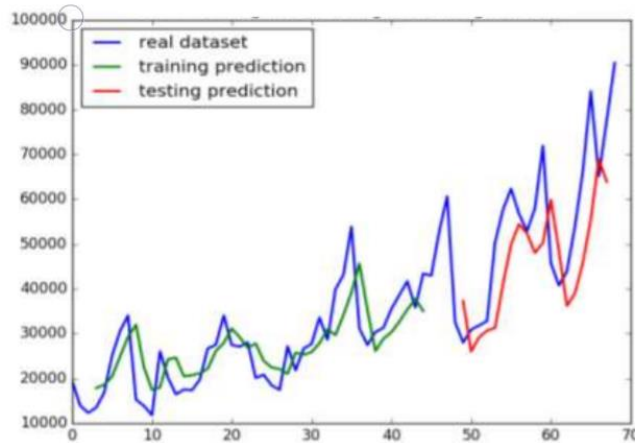


**Hình 2.11 Hoạt động của mạng nơ-ron hồi quy**

Kiểu hoạt động của mạng nơ-ron hồi quy được thể hiện ở hình 2.11. Mỗi hình chữ nhật là 1 vector và các mũi tên thể hiện các hàm biến đổi. Vector đầu vào có màu đỏ, vector đầu ra có màu xanh biển và vector trạng thái thông tin trao đổi giữa các mạng con có màu xanh lá.

### 2.3.2 LSTM hồi quy sử dụng phương thức cửa sổ

Một mô hình dự đoán có nhiều biến số một lần để dự đoán bước tiếp theo là phương thức cửa sổ. Ví dụ: giá trị tại  $t$  và giá trị tại  $t + 1$  được sử dụng để dự đoán giá trị tại thời điểm  $t + 2$ , có thể được phát triển bằng cách sử dụng thời gian hiện tại  $t$  và các lần trước  $t-1$  làm biến đầu vào để dự đoán  $t + 1$ . Khi được tạo thành mô hình hồi quy, biến đầu vào là  $t-1$  và  $t$  và biến đầu ra là  $t + 1$ .



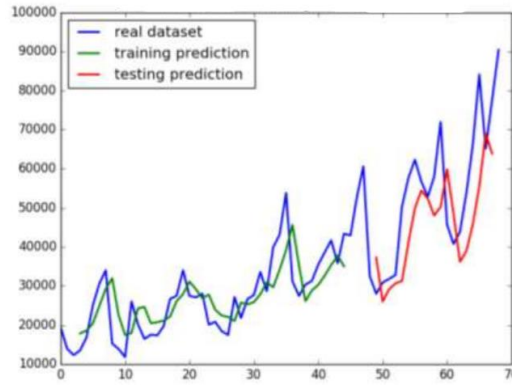
Hình 2.12 LSTM sử dụng phương thức cửa sổ

### 2.3.3 LSTM hồi quy sử dụng bước thời gian

Các quan sát được thực hiện ở ba bước trước ( $t-1$ ,  $t-2$ ,  $t-3$ ) và được sử dụng làm đầu vào để dự đoán quan sát tại thời điểm hiện tại ( $t$ ). Dự đoán sử dụng bước đa thời gian theo khái niệm trình tự để trình tự với một chuỗi dài hơn. Trình tự được sử dụng trong nghiên cứu này được thực hiện như sau:

January	February	March	→	April
February	March	April	→	May
March	April	May	→	June
April	May	June	→	July
...	...	...	...	...
Month $t-2$	Month $t-1$	Month $t$	→	Month $t+1$

Hình 2.13 Trình tự bước thời gian



**Hình 2.14 LSTM sử dụng bộ nhớ giữa các bước**

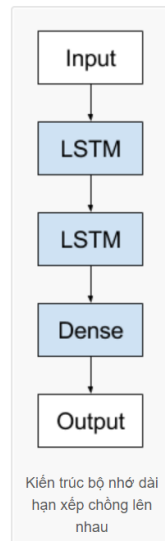
### **2.3.4 LSTM sử dụng bộ nhớ giữa các bước**

Mạng LSTM có bộ nhớ có khả năng ghi nhớ trong các chuỗi dài. Thông thường, trạng thái trong mạng được đặt lại sau mỗi đợt đào tạo khi khớp mô hình, cũng như mỗi lệnh gọi dự đoán hoặc đánh giá. Chúng ta có thể giành quyền kiểm soát tốt hơn khi trạng thái bên trong của mạng LSTM bị xóa trong Keras bằng cách làm cho lớp LSTM có trạng thái. Điều này có nghĩa là nó có thể xây dựng trạng thái trên toàn bộ chuỗi đào tạo và thậm chí duy trì trạng thái đó nếu cần để đưa ra dự đoán. Nó đòi hỏi dữ liệu đào tạo không được xáo trộn khi lắp mạng. Nó cũng yêu cầu thiết lập lại rõ ràng trạng thái mạng sau mỗi lần tiếp xúc với dữ liệu huấn luyện bằng cách gọi đến trạng thái đặt lại.

### **2.3.5 LSTM xếp chồng sử dụng bộ nhớ giữa các bước**

Kiến trúc LSTM xếp chồng là những lợi ích tương tự có thể được khai thác với LSTM. Các mạng LSTM hoạt động trên dữ liệu chuỗi, điều đó có nghĩa là việc thêm các lớp sẽ thêm các mức độ trừu tượng của các quan sát đầu vào theo thời gian. Trong thực tế, quan sát khúc dữ liệu theo thời gian hoặc đại diện cho vấn đề ở quy mô thời gian khác nhau. Sự kiện xây dựng một RNN sâu bằng cách xếp chồng nhiều trạng thái ẩn lặp lại lên nhau. Cách tiếp cận này có khả năng cho phép trạng thái ẩn ở mỗi cấp hoạt động ở các khoảng thời gian khác nhau.





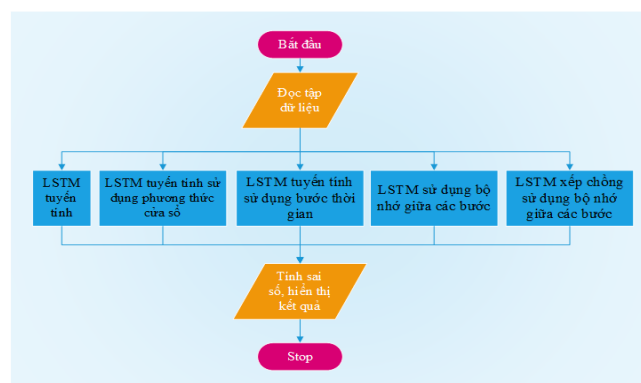
Hình 2.15 LSTM xếp chồng sử dụng bộ nhớ giữa các bước

## 2.4 Nghiên cứu vấn đề dự báo chuỗi thời gian, lượng hành khách đi máy bay quốc tế

### 2.4.1 Phân tích yêu cầu

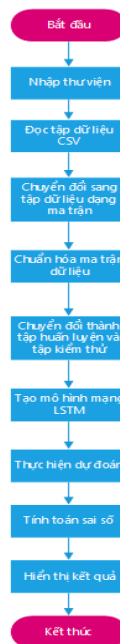
Lượng hành khách đi máy bay quốc tế không chỉ phụ thuộc vào nhu cầu đi lại thực tế của hành khách mà còn chịu ảnh hưởng bởi nhiều yếu tố khác như tình hình kinh tế, thời tiết, các dịp nghỉ lễ, các sự kiện diễn ra trong năm... Hệ thống dự báo lượng hành khách đi máy bay quốc tế được xây dựng nhằm giúp hãng hàng không chuẩn bị đáp ứng với lưu lượng hành khách trong tương lai

### 2.4.2 Mô hình thử nghiệm



Hình 2.16 Mô hình thực nghiệm hệ thống dự đoán lượng hành khách đi máy bay

### 2.3.3 Các bước xử lý



Hình 2.17 Các bước xử lý của mô-đun dự đoán lượng hành khách đi máy bay

## 2.5 Cài đặt ứng dụng

Thực nghiệm được tiến hành trên 1 máy tính với cấu hình như sau:

- CPU Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz
- RAM 2 GB
- OS Windows 10
- Ngôn ngữ lập trình Python

Các thư viện và phần mềm hỗ trợ học sâu được sử dụng trong thực nghiệm là Anaconda, Keras, Tensorflow và PyCharm.

```

Anaconda Prompt (Anaconda3) - conda create -n tensorflow python=3.5 - pip install --ignore-installed --upgrade tensorflow
Proceed ([y]/n)? y

Downloading and Extracting Packages
ca-certificates-2019 | 166 KB | ##### | 100%
certifi-2019.6.16 | 156 KB | ##### | 100%
pip-19.2.2 | 1.9 MB | ##### | 100%
python-3.7.4 | 18.2 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
# $ conda activate tensorflow
#
# To deactivate an active environment, use
#
# $ conda deactivate

(base) C:\Users\HATV>pip install --ignore-installed --upgrade tensorflow
Collecting tensorflow
  Downloading https://files.pythonhosted.org/packages/f7/08/25e47a53692c2e0dcd2211a493ddfe9007a5cd92e175d6dffa6169a0b392/tensorflow-1.14.0-cp37-cp37m-win_amd64.whl (68.3MB)
    | 67.4MB 2.2MB/s eta 0:00:01
  
```

Hình 2.18 Cài đặt Tensorflow

```

root@bb599330d88e: /
14ca88e9f672: Pull complete
a31c3b1caad4: Pull complete
b054a26005b7: Pull complete
ce020afbc108: Pull complete
ff2859ce8b51: Pull complete
fcb44a17b2cb: Pull complete
2ba920e8c8f8: Pull complete
c7618033dbfa: Pull complete
879aa60b376b: Pull complete
Digest: sha256:87463fd80faae7979b78d2f1a26d62262210653eb166b638069ed06ae68dacb
Status: Downloaded newer image for tensorflow/tensorflow:latest
docker.io/tensorflow/tensorflow:latest

(base) C:\Windows\system32>docker run -it -p 8888:8888 tensorflow/tensorflow

TensorFlow

WARNING: You are running this container as root, which can cause new files in
mounted volumes to be created as the root user on your host machine.

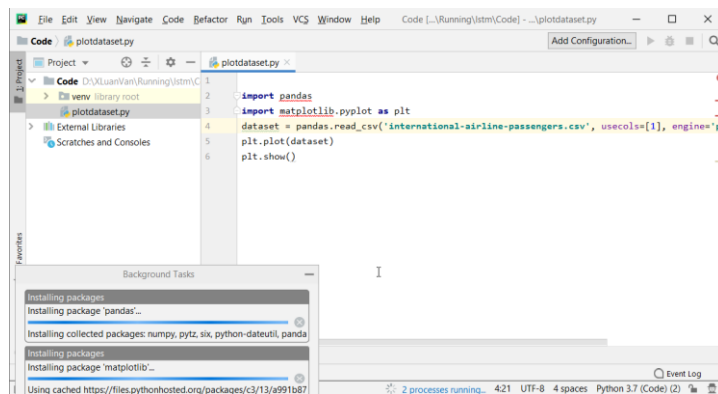
To avoid this, run the container by specifying your user's userid:

$ docker run -u $(id -u):$(id -g) args...

root@bb599330d88e: /#

```

**Hình 2.19 Môi trường phát triển Tensorflow**



**Hình 2.20 Phần mềm IDE Pycharm**

## 2.6 Kết luận chương

Trong chương 2, chúng ta đã tìm hiểu được kiến trúc và quy trình xử lý thông tin của mạng LSTM. Vấn đề được đặt ra để giải quyết bài toán dự báo chuỗi thời gian, lượng hành khách đi máy bay quốc tế. Chúng tôi đã nêu ra được mô hình thử nghiệm và các bước xử lý của mô-đun dự đoán. Thử nghiệm và đánh giá kết quả sẽ được chúng tôi nêu ra trong Chương 3.

## CHƯƠNG 3

### THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

#### 3.1 Giới thiệu bài toán ước lượng hành khách đi máy bay quốc tế

Với dữ liệu đầu vào một tập số lượng hành khách đi máy bay quốc tế theo tháng và năm, từ 07/2005 đến 03/2016, với 129 lượt quan sát. Bộ dữ liệu có sẵn miễn phí từ trang web <https://data.world/data-society/air-traffic-passenger-data> dưới dạng tệp tin CSV.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		Adjusted Passenger Count												
2	2005-7	160890												
3	2005-8	157700												
4	2005-9	135944												
5	2005-10	138552												
6	2005-11	135008												
7	2005-12	130393												
8	2006-1	110409												
9	2006-2	101163												
10	2006-3	124880												
11	2006-4	141717												

**Hình 3.1 Dữ liệu đầu vào số hành khách đi máy bay quốc tế**

Cột đầu tiên là số liệu thời gian theo đơn vị tháng (yyyy-mm)

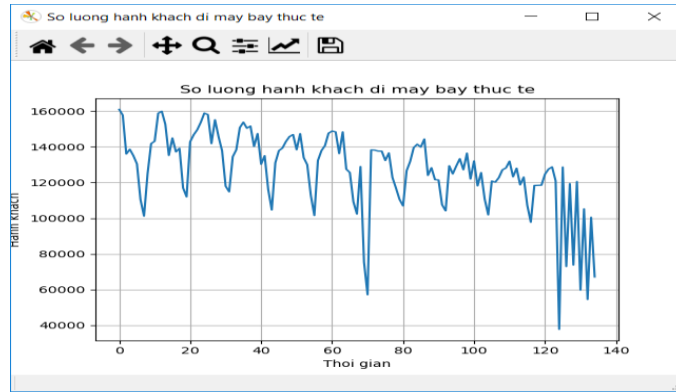
Cột thứ hai là số lượng hành khách đi máy bay trong thời gian tháng đó

#### 3.2 Các kịch bản thử nghiệm

- LSTM hồi quy
- LSTM hồi quy sử dụng phương thức cửa sổ
- LSTM hồi quy sử dụng bước thời gian
- LSTM sử dụng bộ nhớ giữa các bước
- LSTM xếp chồng sử dụng bộ nhớ giữa các bước

#### 3.3 Một số kết quả thực nghiệm

##### 3.3.1 Kết quả thử nghiệm độ chính xác dự đoán lượng hành khách đi máy bay



Hình 3.2 Đồ thị đầu vào số hành khách đi máy bay quốc tế

### 3.3.2 Phương pháp LSTM hồi quy

Theo phương pháp này đầu ra của thử nghiệm:

Epoch 98/100

- 0s - loss: 0.0020

Epoch 99/100

- 0s - loss: 0.0020

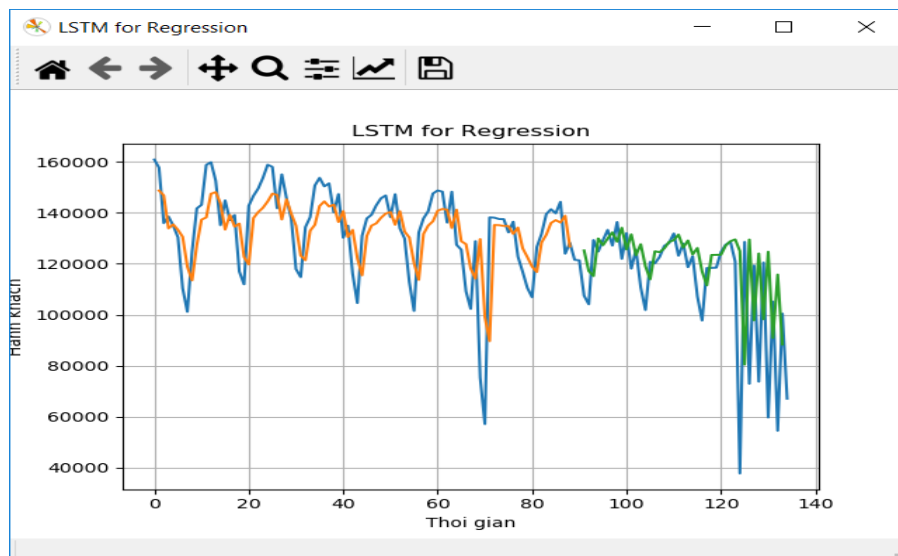
Epoch 100/100

- 0s - loss: 0.0020

Train Score: 22.92 RMSE

Test Score: 47.53 RMSE

Chúng ta có thể thấy rằng mô hình có lỗi trung bình khoảng 22.92 hành khách (tính bằng nghìn) trên tập dữ liệu huấn luyện và khoảng 47.53 hành khách (tính bằng nghìn) trên tập dữ liệu thử nghiệm.



Hình 3.3 LSTM hồi quy

### 3.3.3 Phương pháp LSTM hồi quy sử dụng phương thức cửa sổ

Theo phương pháp này đầu ra của thử nghiệm:

Epoch 98/100

- 0s - loss: 0.0021

Epoch 99/100

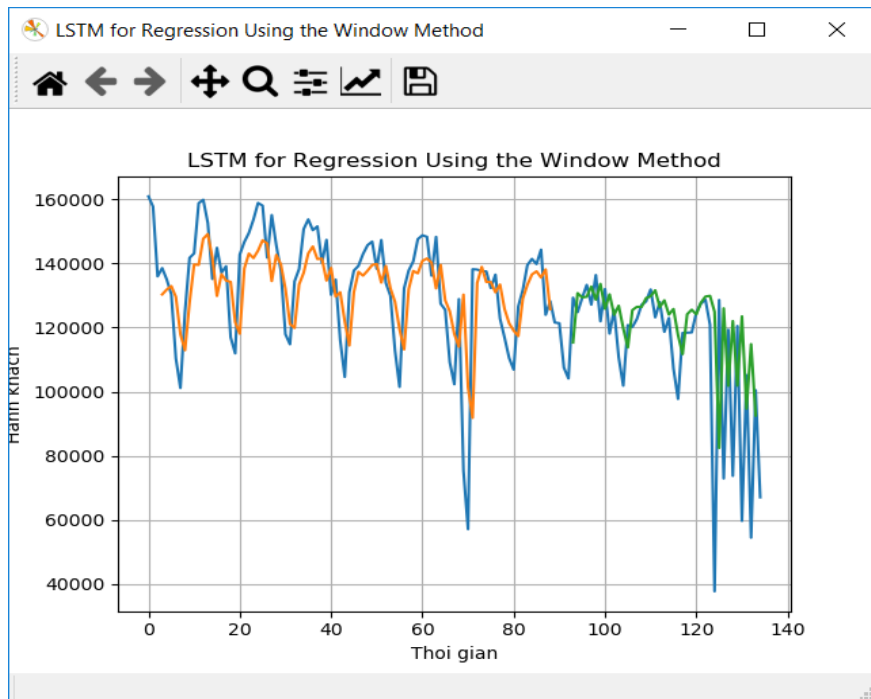
- 0s - loss: 0.0022

Epoch 100/100

- 0s - loss: 0.0020

Train Score: 24.19 RMSE

Test Score: 58.03 RMSE



Hình 3.4 LSTM hồi quy sử dụng phương thức cửa sổ

### 3.3.4 Phương pháp LSTM hồi quy sử dụng bước thời gian

Theo phương pháp này đầu ra của thử nghiệm:

Epoch 98/100

- 0s - loss: 0.0021

Epoch 99/100

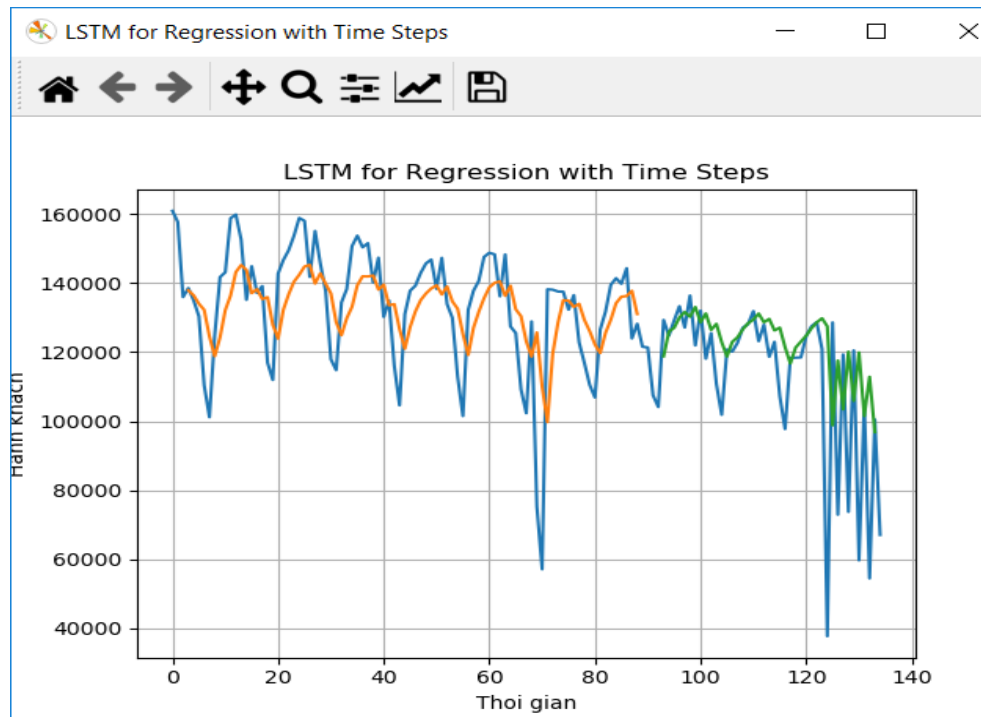
- 0s - loss: 0.0022

Epoch 100/100

- 0s - loss: 0.0020

Train Score: 23.73 RMSE

Test Score: 58.79 RMSE



**Hình 3.5 LSTM hồi quy sử dụng bước thời gian**

### ***3.3.5 Phương pháp LSTM sử dụng bộ nhớ giữa các bước***

Theo phương pháp này đầu ra của thử nghiệm:

Epoch 98/100

- 0s - loss: 0.0017

Epoch 99/100

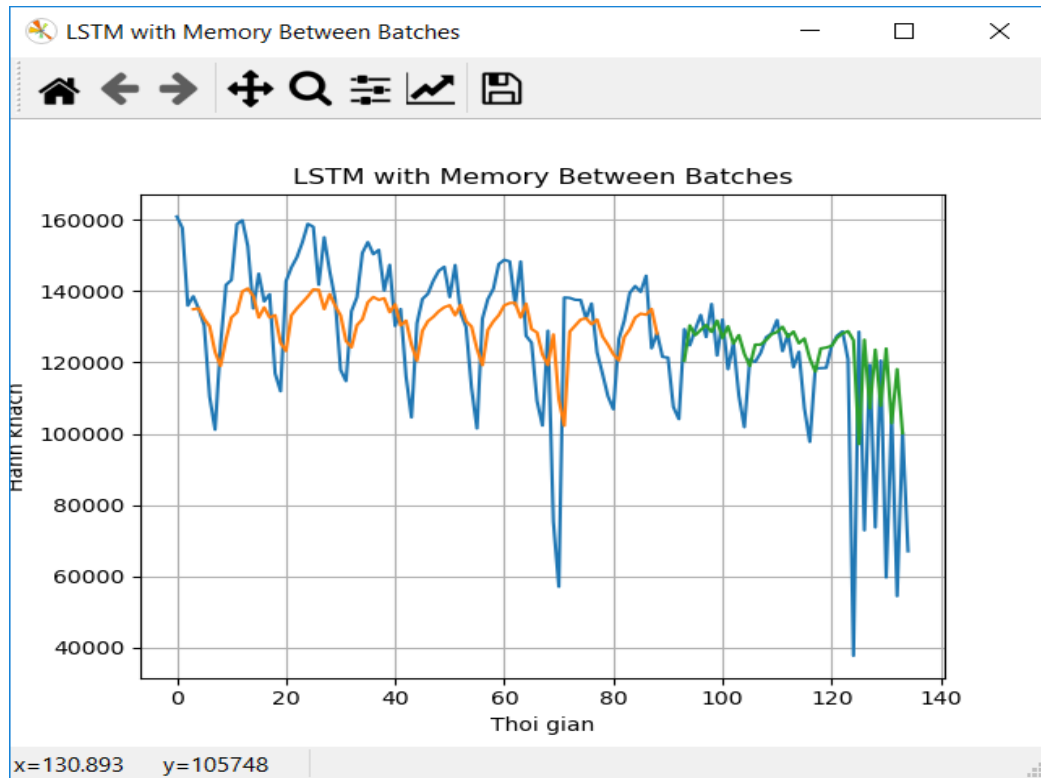
- 0s - loss: 0.0017

Epoch 100/100

- 0s - loss: 0.0017

Train Score: 20.80 RMSE

Test Score: 55.25 RMSE



**Hình 3.6 LSTM sử dụng bộ nhớ giữa các bước**

### ***3.3.6 LSTM xếp chồng sử dụng bộ nhớ giữa các bước***

Theo phương pháp này đầu ra của thử nghiệm:

Epoch 98/100

- 0s - loss: 0.0038

Epoch 99/100

- 0s - loss: 0.0037

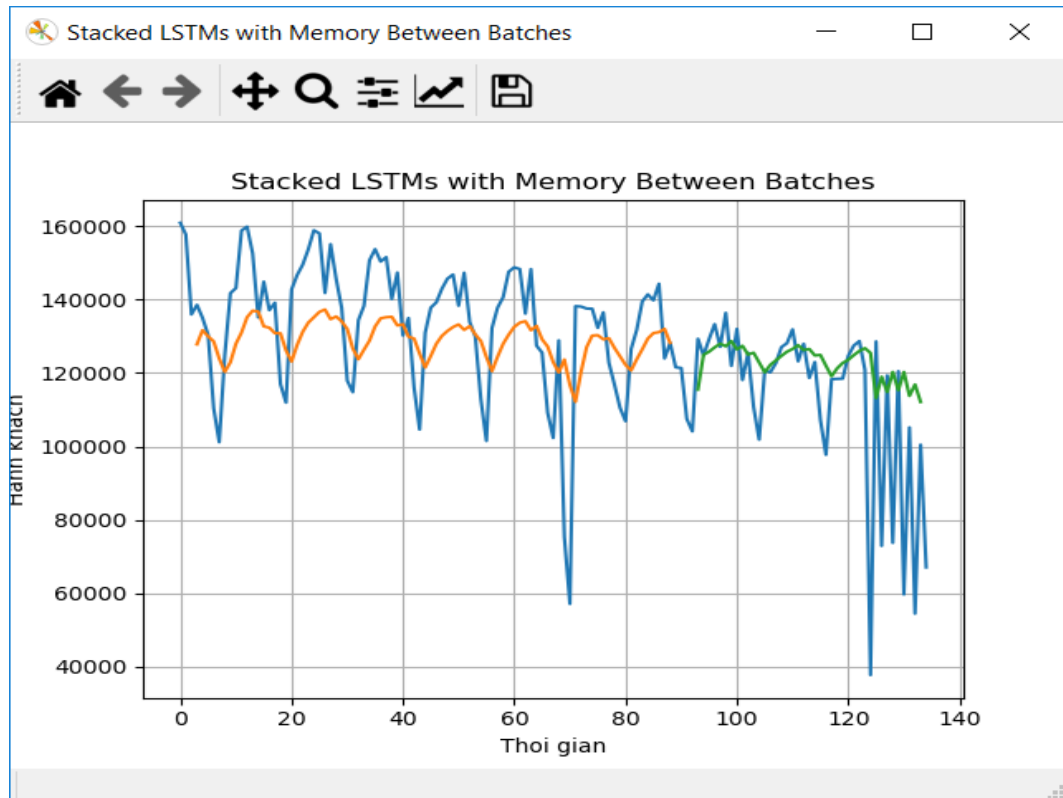
Epoch 100/100

- 0s - loss: 0.0037

Train Score: 30.10 RMSE

Test Score: 79.68 RMSE





Hình 3.7 LSTM xếp chồng sử dụng bộ nhớ giữa các bước

### 3.4 Kết luận chương

Tổng kết dữ liệu lỗi trung bình của 5 kịch bản cho ta Bảng 3.1

Bảng 3.1 Đánh giá kết quả dự đoán hành khách đi máy bay quốc tế

	Lỗi trung bình (Tập huấn luyện)	Lỗi trung bình (Tập kiểm nghiệm)
<b>LSTM hồi quy</b>	22.92	47.53
<b>LSTM hồi quy sử dụng phương thức cửa sổ</b>	24.19	58.03
<b>LSTM hồi quy sử dụng bước thời gian</b>	23.73	58.79
<b>LSTM sử dụng bộ nhớ giữa các bước</b>	20.80	55.25
<b>LSTM xếp chồng sử dụng bộ nhớ giữa các bước</b>	30.10	79.68

))

## KẾT LUẬN

Luận văn đã trình bày được khái niệm, kiến thức cơ bản về công nghệ LSTM, áp dụng công nghệ mạng nơ-ron nhân tạo trong việc dự đoán lượng hành khách đi máy bay quốc tế. Các kết quả của Luận văn bao gồm:

Đã đề xuất và thử nghiệm phương pháp dự đoán lượng hành khách đi máy bay áp dụng công nghệ LSTM. Làm thế nào để phát triển các mạng LSTM cho hồi quy, khung cửa sổ và thời gian dựa trên các vấn đề dự đoán chuỗi thời gian. Làm thế nào để phát triển và đưa ra dự đoán bằng cách sử dụng các mạng LSTM duy trì trạng thái qua các quá trình. Phát triển ứng dụng ứng dụng trên hệ điều hành windows.

Về hạn chế của luận văn, do thời gian và nguồn lực và kiến thức của học viên còn có hạn trong việc nghiên cứu nên các kết quả của luận văn mới thực hiện được việc dự đoán hành khách đi máy bay của hãng. Chưa thực nghiệm hệ thống với các thuật toán khác để có cơ sở so sánh, đánh giá hiệu quả

Trong thời gian tới học viên mong muốn nghiên cứu sâu hơn để cải thiện hiệu suất, tăng tốc độ xử lý dữ liệu với dữ liệu lớn. Nghiên cứu các phương pháp nâng cao độ chính xác dự đoán lượng hành khách đi máy bay. Xây dựng hệ thống hoàn chỉnh với tập dữ liệu lớn và triển khai thử nghiệm ứng dụng trên các nền tảng khác.