BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH

BÁO CÁO BẢO VỆ ĐỒ ÁN THỰC TẬP CHUYÊN MÔN

MÔN: THỰC TẬP CHUYÊN MÔN

Đề tài: PHÂN TÍCH, XỬ LÝ VÀ DỰ ĐOÁN LƯU LƯỢNG THAM GIA GIAO THÔNG DỰA TRÊN TẬP DỮ LIỆU TRỰC TUYẾN

GVHD: Th.S Phạm Thị Dung

K.S Trần Quốc Khánh

Sinh viên thực hiện:

Trần Thị Minh Ánh MSV: 6151071001

TPHCM, ngày 20 tháng 06 năm 2023

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN

BÁO CÁO THỰC TẬP CHUYÊN MÔN

ĐỀ TÀI: PHÂN TÍCH, XỬ LÝ VÀ DỰ ĐOÁN LƯU LƯỢNG THAM GIA GIAO THÔNG DỰA TRÊN TẬP DỮ LIỆU TRỰC TUYẾN

Sinh viên thực hiện: TRẦN THỊ MINH ÁNH

Lớp : CQ.61.CNTT

Khoá : **61**

TP. Hồ Chí Minh, ngày 20 tháng 06 năm 2023

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

THIẾT KẾ TỔNG QUAN ĐỀ TÀI

1. Thông tin Sinh viên:

Họ tên: Trần Thị Minh Ánh — Mã sinh viên :6151071001

Lớp : CNTT_K61 — Hệ :4

Ngành đào tạo : Công nghệ thông tin — Khoá :2020

Email : 6151071001@st.utc2.edu.vn — Số điện thoại :0852785547

2. Thông tin Giảng viên hướng dẫn:

Họ tên :Trần Thị Dung — Học vị :Thạc sỹ

Email : ttdung@st.utc2.edu.vn — Số điện thoại :0388389579

Đơn vị công tác: Trường Đại học Giao thông Vận tải PH tại TP Hồ Chí Minh

Họ tên : Trần Quốc Khánh — Học vị : Kỹ sư

Email : — Số điện thoại :

Đơn vị công tác: Trường Đại học Giao thông Vận tải PH tại TP Hồ Chí Minh

GIỚI THIỆU

I. Tên đề tài

Phân tích, xử lý và dự đoán lưu lượng giao thông dựa trên bộ dữ liệu trực tuyến

II. Giới thiệu

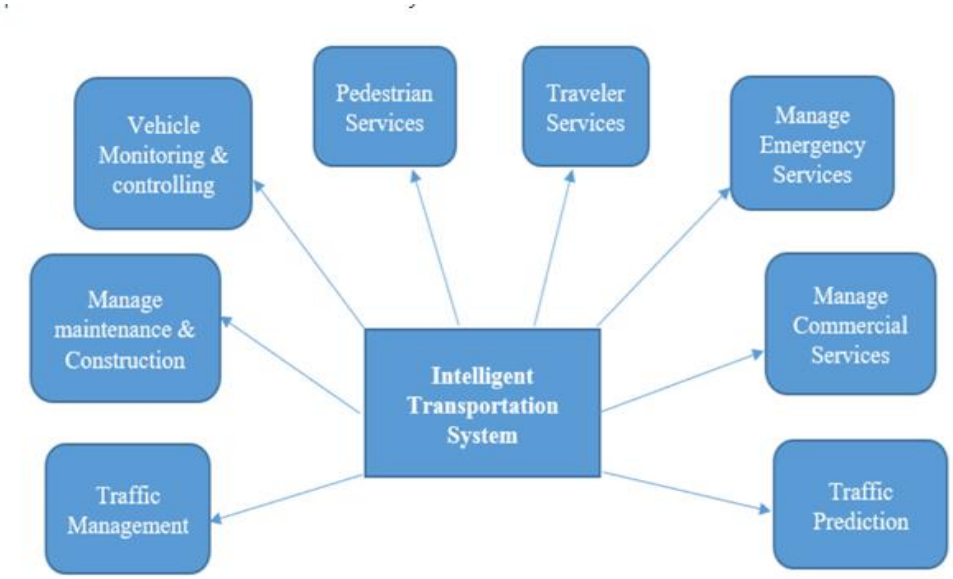
Ở nhiều nơi trên thế giới, tắc nghẽn giao thông là một vấn đề nghiêm trọng. Rất khó để xác định và dự báo chính xác vì nó bị ảnh hưởng bởi một loạt các yếu tố đa dạng. Một số vấn đề chính gây ra bởi tắc nghẽn giao thông là ô nhiễm không khí, đi lại dài, ùn tắc giao thông và tai nạn giao thông.

Để tạo điều kiện lưu lượng giao thông thông suốt trên mạng lưới đường bộ, chúng ta phải giải quyết vấn đề chính là tắc nghẽn giao thông. Do nhu cầu giao thông tăng cao, việc mở rộng cơ sở hạ tầng mạng lưới đường bộ bằng cách mở rộng đường bộ đơn giản là không đủ để quản lý các điều kiện lưu lượng giao thông thông suốt. Để mô hình hóa các trường hợp lưu lượng giao thông, một số loại phương pháp mô hình hóa lưu lượng giao thông là cần thiết mà em sẽ cố gắng thực hiện trong dự án này.

Dự đoán lưu lượng giao thông (Traffic Flow Prediction - TFP) có nghĩa là dự đoán lưu lượng và mật độ của lưu lượng giao thông, thường được sử dụng để điều khiển phương tiện di chuyển, giảm tắc đường và tạo ra tuyến đường tối ưu (ít tốn thời gian hoặc năng lượng nhất). Với sự tiến bộ gần đây trong trí tuệ nhân tạo, học máy (Machine Learning - ML), học sâu (Deep Learning - DL) và dữ liệu lớn (Big data), nghiên cứu về lĩnh vực dự đoán lưu lượng giao thông đã được mở rộng rộng rãi.

ITS là một hệ thống quản lý giao thông tích hợp bao gồm các công nghệ truyền thông dữ liệu, xử lý thông tin và quản lý giao thông tiên tiến. Trong những năm gần đây, sự thành công của học sâu trong thị giác máy tính, nhận dạng tốc độ và xử lý ngôn ngữ tự nhiên khiến việc áp dụng nó vào ITS trở nên tự nhiên. Em chia các ứng dụng trong ITS thành các nhiệm vụ nhận dạng hình ảnh, TFP, dự đoán tốc độ giao thông (TSP), dự đoán thời gian di chuyển (TTP) và các nhiệm vụ khác. Thuật ngữ 'ITS' đề cập đến việc sử dụng các hệ thống liên lạc, thông tin, giao thông vận tải và giao thông đô thị. Hai trong số các mục tiêu chính của ITS là hiệu quả và an toàn giao thông. Giảm tình trạng

tắc nghẽn và chậm trễ của giao lộ, cải thiện thời gian giao thông, cải thiện kiểm soát tốc độ, quản lý năng lực và quản lý sự cố là tất cả những lợi ích của ITS.



Hình 1 Mô tả nhiều nhiệm vụ được bao phủ bởi ITS

- **Lý do chọn đề tài**

Dự án "Phân tích, xử lý và dự đoán lưu lượng giao thông dựa trên bộ dữ liệu trực tuyến" nhằm nghiên cứu và áp dụng các phương pháp, kỹ thuật để hiểu và dự đoán lưu lượng giao thông trong đô thị. Đây là một đề tài hết sức quan trọng và hứa hẹn mang lại những đóng góp đáng kể cho lĩnh vực quản lý giao thông và cải thiện sự di chuyển trong thành phố.

Giao thông đóng vai trò không thể thiếu trong cuộc sống đô thị, tác động trực tiếp đến sự di chuyển của con người và hàng hóa. Để có được quy hoạch và quản lý giao thông hiệu quả, việc hiểu và dự đoán lưu lượng giao thông là điều cần thiết. Điều này giúp tối ưu hóa sự di chuyển, giảm thiểu ùn tắc và tăng cường hiệu suất hệ thống giao thông.

Trong thời đại kỹ thuật số, dữ liệu trực tuyến về giao thông ngày càng trở nên phong phú và đa dạng. Các cảm biến đường, hệ thống giám sát, dữ liệu GPS và các ứng dụng di động cung cấp thông tin liên tục về lưu lượng giao thông. Điều này mở ra cơ

hội cho việc nghiên cứu và sử dụng dữ liệu trực tuyến để phân tích và xử lý thông tin về lưu lượng giao thông.

Phân tích và xử lý dữ liệu lưu lượng giao thông đòi hỏi sự áp dụng của các phương pháp và kỹ thuật tiên tiến. Các thuật toán học máy, khai phá dữ liệu, mạng Neuron và kỹ thuật thống kê được áp dụng để hiểu và dự đoán xu hướng lưu lượng giao thông. Điều này giúp tạo ra các mô hình dự đoán lưu lượng giao thông chính xác và đáng tin cậy.

Tuy nhiên, việc nghiên cứu trong lĩnh vực này còn đối mặt với nhiều thách thức. Tính thời gian thực, độ tin cậy của dữ liệu, kích thước lớn của dữ liệu giao thông, sự biến đổi phức tạp của lưu lượng và tác động của yếu tố bên ngoài như thời tiết và sự kiện đặc biệt đều là những thách thức cần được vượt qua. Đòi hỏi chúng ta phải tiếp tục nghiên cứu và phát triển các phương pháp và công nghệ mới để nâng cao hiệu quả trong việc phân tích và dự đoán lưu lượng giao thông.

Với những lý do trên nên em chọn đề tài "Phân tích, xử lý và dự đoán lưu lượng giao thông dựa trên bộ dữ liệu trực tuyến" để nghiên cứu và thực hiện trong bài báo cáo này. Việc áp dụng các kết quả nghiên cứu vào thực tế có thể giúp quản lý giao thông đô thị hiệu quả hơn, tối ưu hóa hệ thống giao thông và phát triển các giải pháp thông minh trong lĩnh vực giao thông.

- **Mục tiêu đề tài**

Mục tiêu trong đề tài nghiên cứu này là xây dựng RNN nhiều bước với mô hình LSTM để đưa ra dự đoán về lưu lượng truy cập trong 2 giờ tới trong tương lai, với 6 giờ lịch sử trong tập dữ liệu.

- **Đối tượng**

Đối tượng chính trong quá trinh nghiên cứu là dữ liệu về lưu lượng giao thông hàng giờ trên đường cao tốc Interstate 94 hướng về phía Tây của trạm ATR 301 của Bộ Giao thông Minnesota (MN DoT), xấp xỉ ở giữa đường từ Minneapolis đến St Paul, Minnesota.

- **Phạm vi nghiên cứu**

Phạm vi nghiên cứu sẽ tập trung vào việc phân tích, xử lý và dự đoán lưu lượng giao thông trong khu vực đô thị dựa trên tập dữ liệu về lưu lượng giao thông khu vực trên

III. Phương pháp nghiên cứu

Thu thập dữ liệu: Em sẽ thu thập dữ liệu giao thông trực tuyến từ các nguồn khác nhau như cảm biến đường, camera giám sát, dữ liệu GPS và ứng dụng di động. Dữ liệu này sẽ cung cấp thông tin liên tục về lưu lượng giao thông và các yếu tố liên quan khác như tốc độ, thời gian di chuyển và lưu lương phương tiện tham gia giao thông.

Tiền xử lý dữ liệu: Trước khi tiến hành phân tích, em sẽ tiền xử lý dữ liệu để loại bỏ nhiễu, xử lý các giá trị bị thiếu và chuẩn hoá dữ liệu. Quá trình này giúp đảm bảo tính chính xác và đáng tin cậy của dữ liệu.

Phân tích dữ liệu: Em sẽ áp dụng các phương pháp và kỹ thuật khai phá dữ liệu và thống kê để tìm ra các mẫu, liên kết và xu hướng trong dữ liệu giao thông. Các phương pháp này bao gồm phân tích đường cong thời gian, phân tích hồi quy, phân tích chuỗi thời gian và phân tích tương quan.

Xây dựng mô hình dự đoán: Dựa trên các phân tích dữ liệu, em sẽ xây dựng các mô hình dự đoán lưu lượng giao thông. Các phương pháp học máy như học sâu, học kỹ thuật thống kê và mạng Neuron sẽ được áp dụng để xây dựng những mô hình chính xác và linh hoạt.

Đánh giá mô hình: Cuối cùng, em sẽ đánh giá hiệu suất của các mô hình dự đoán. Em sẽ sử dụng các Đo đạt, đánh giá như sai số dự đoán, độ chính xác và độ tin cậy để đánh giá mô hình. Ngoài ra, em cũng sẽ so sánh kết quả dự đoán với dữ liệu thực tế để đánh giá khả năng ứng dụng của mô hình trong thực tế.

Sử dụng các phương pháp và kỹ thuật nghiên cứu trên, em hy vọng có thể hiểu và dự đoán lưu lượng giao thông một cách chính xác và đáng tin cậy, từ đó giúp cải thiện quản lý giao thông và tối ưu hóa sự di chuyển trong đô thị.

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC CHỮ VIẾT TẮT

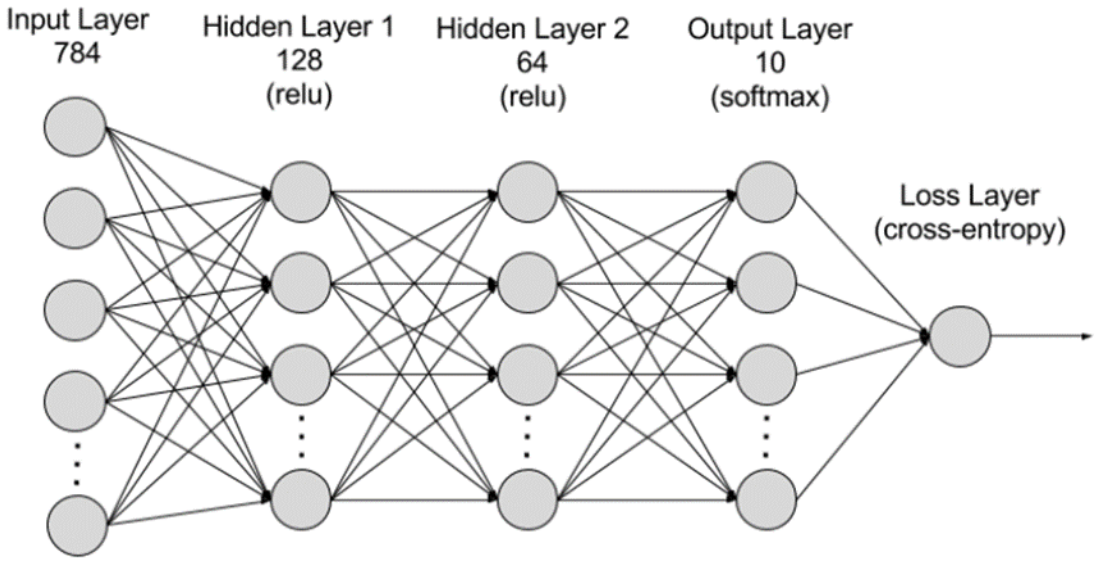
SỐ THỨ TỰ	TÊN VIẾT TẮT	TÊN ĐẦY ĐỦ
1	CNN	Convolutional Neural Networks
2	RNN	Recursive Neural Networks
3	LSTM	Long Short-Term Memory

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về Học sâu

Kỹ thuật Học sâu (Deep Learning) là một lĩnh vực của trí tuệ nhân tạo (Artificial Intelligence) tập trung vào việc xây dựng và huấn luyện các mạng Neuron nhân tạo sâu với nhiều lớp ẩn. Mục tiêu chính của Học sâu là tự động học và rút trích các đặc trưng phức tạp từ dữ liệu đầu vào mà không cần sự can thiệp tay người.

Mạng nơ-ron nhân tạo trong Học sâu thường có kiến trúc mạnh mẽ, với hàng hoặc hàng trăm lớp ẩn giữa lớp đầu vào và lớp đầu ra. Các mô hình Học sâu phổ biến bao gồm Convolutional Neural Networks (CNN) cho xử lý hình ảnh, Recurrent Neural Networks (RNN) cho xử lý dữ liệu chuỗi, và Transformers cho xử lý dữ liệu có cấu trúc.



Hình 2 Mạng nơ-ron nhân tạo

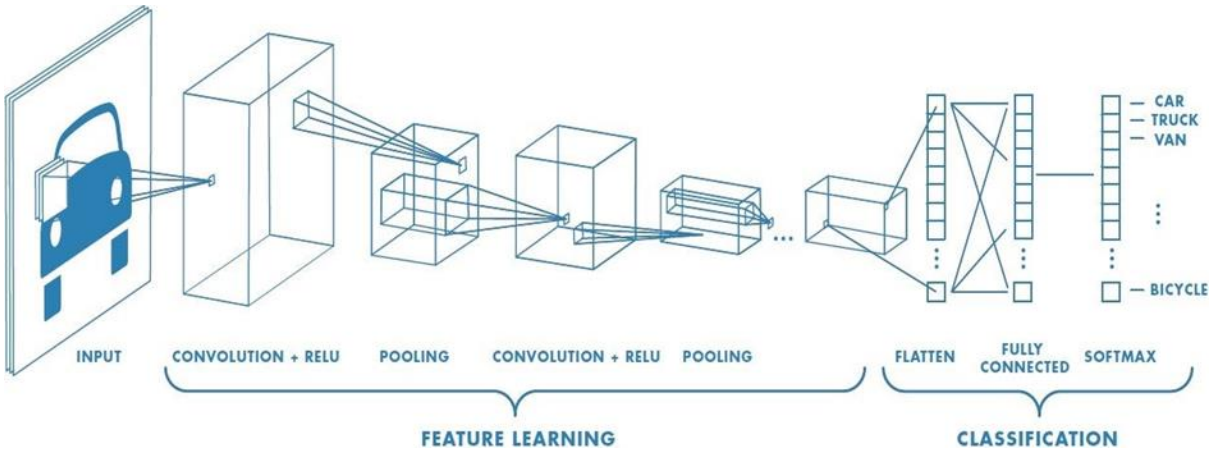
Quá trình huấn luyện mô hình Học sâu thường dựa trên một lượng lớn dữ liệu huấn luyện và thuật toán lan truyền ngược (backpropagation) để điều chỉnh trọng số của các nơ-ron. Quá trình này giúp mô hình học cách biểu diễn các đặc trưng phức tạp từ dữ liệu và tối ưu hóa để đạt được độ chính xác cao trong việc dự đoán hoặc phân loại.

Học sâu đã đạt được những thành tựu đáng kể trong nhiều lĩnh vực, bao gồm nhận dạng hình ảnh, nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên, tự lái xe, xử phân

loại sản phẩm trong công nghiệp và nhiều ứng dụng khác. Với khả năng học tập sâu và khả năng tự động học các đặc trưng, Học sâu mang lại tiềm năng lớn trong việc giải quyết các bài toán phức tạp và đưa ra dự đoán chính xác từ dữ liệu không cấu trúc và không rõ ràng.

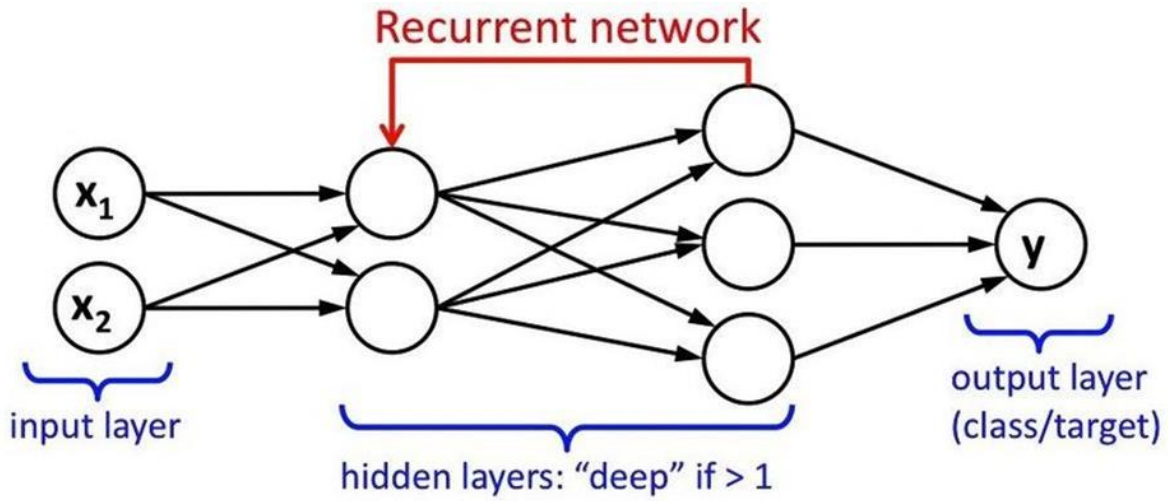
1.2. Các phương pháp và mô hình học sâu phổ biến

- Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN): Được sử dụng chủ yếu trong xử lý ảnh và video. CNN có khả năng tự động học các đặc trưng cấu trúc của hình ảnh thông qua việc áp dụng các lớp tích chập và lớp gộp. CNN thường được sử dụng trong các bài toán như phân loại ảnh, phát hiện đối tượng và nhận dạng khuôn mặt.



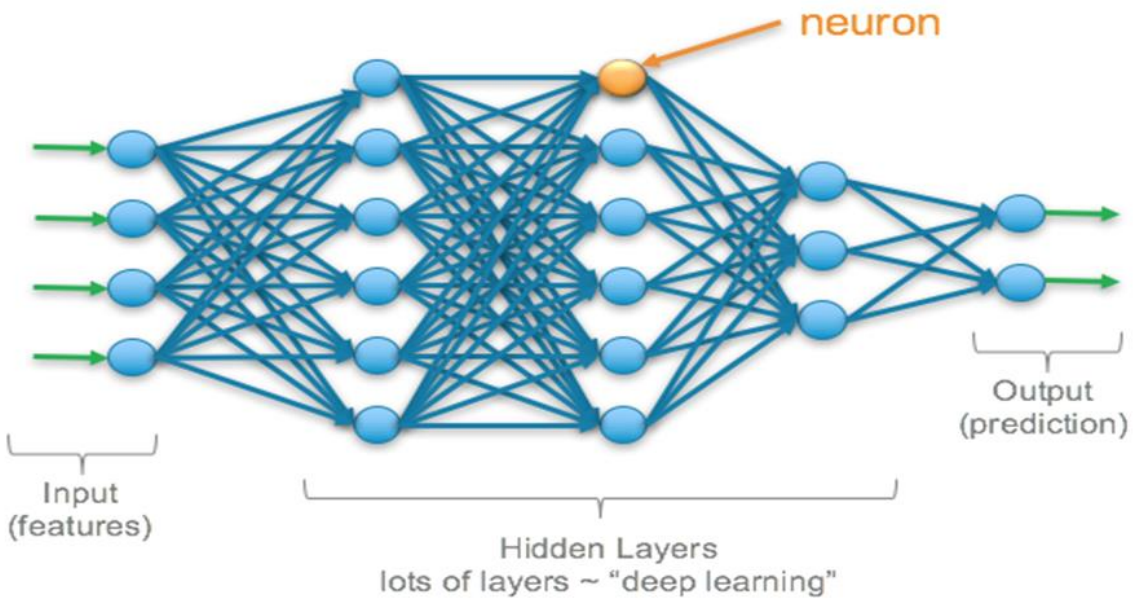
Hình 3 Convolutional Neural Networks - CNN

- Mạng nơ-ron tuần tự (Recurrent Neural Networks - RNN): Được sử dụng trong các tác vụ liên quan đến dữ liệu chuỗi, ví dụ như xử lý ngôn ngữ tự nhiên, dịch máy, nhận dạng giọng nói và dự báo chuỗi thời gian. RNN có khả năng lưu trữ thông tin liên quan đến quá khứ và sử dụng nó để dự đoán các sự kiện trong tương lai.



Hình 4 Recurrent Neural Networks - RNN

- Mạng nơ-ron tái phát (Recursive Neural Networks - Recursive NN): Được sử dụng trong xử lý cây cú pháp và dữ liệu có cấu trúc phức tạp. Mô hình này có khả năng tổ chức dữ liệu theo cấu trúc phân cấp và tìm hiểu các mối quan hệ phụ thuộc giữa các thành phần của dữ liệu.



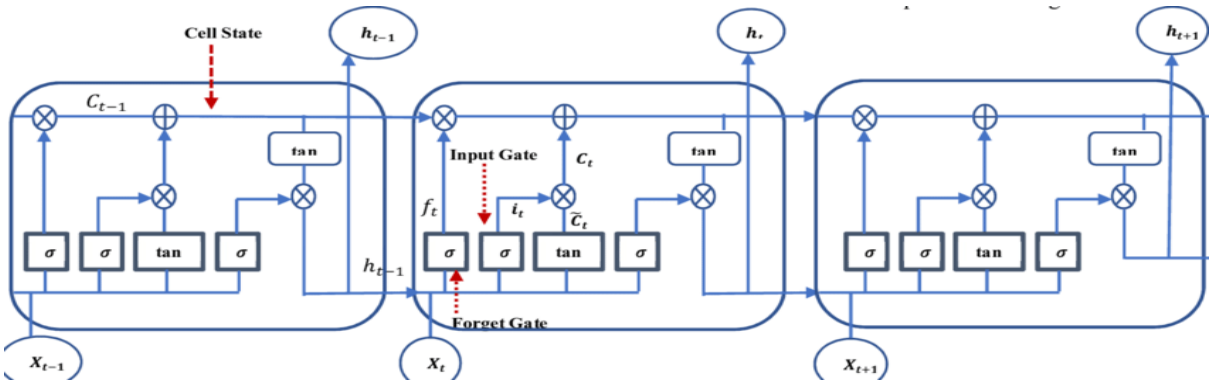
Hình 5 Recursive Neural Networks - Recursive NN

- LSTM (Long Short-Term Memory)

LSTM (Long Short-Term Memory) là một loại mạng Neuron RNN (Recurrent Neural Network) cải tiến được giới thiệu để giải quyết vấn đề mất mát thông tin và khả năng học phụ thuộc xa trong mạng RNN tiêu chuẩn.

Trong RNN tiêu chuẩn, thông tin chỉ được truyền qua một số lớp nơ-ron liên tiếp. Khi thông tin đi qua nhiều lớp, có thể xảy ra hiện tượng mất mát thông tin (vanishing gradient) do sự lặp lại của ma trận trọng số trong quá trình lan truyền ngược. Điều này làm cho RNN khó khăn trong việc giữ và sử dụng thông tin từ quá khứ trong quá trình dự đoán.

LSTM giải quyết vấn đề này bằng cách sử dụng cơ chế "cổng" (gating mechanism) để kiểm soát thông tin đi qua các đơn vị nơ-ron. LSTM có cấu trúc chứa các đơn vị nhớ (memory cell) và các cổng đầu vào (input gate), cổng quên (forget gate) và cổng đầu ra (output gate). Các cổng này giúp điều chỉnh lưu lượng thông tin đi qua và quyết định thông tin nào sẽ được giữ lại và thông tin nào sẽ bị loại bỏ.



Hình 6 The Long Short-Term Memory (LSTM) based Recurrent Neural Networks

Cụ thể, các cổng của LSTM hoạt động như sau:

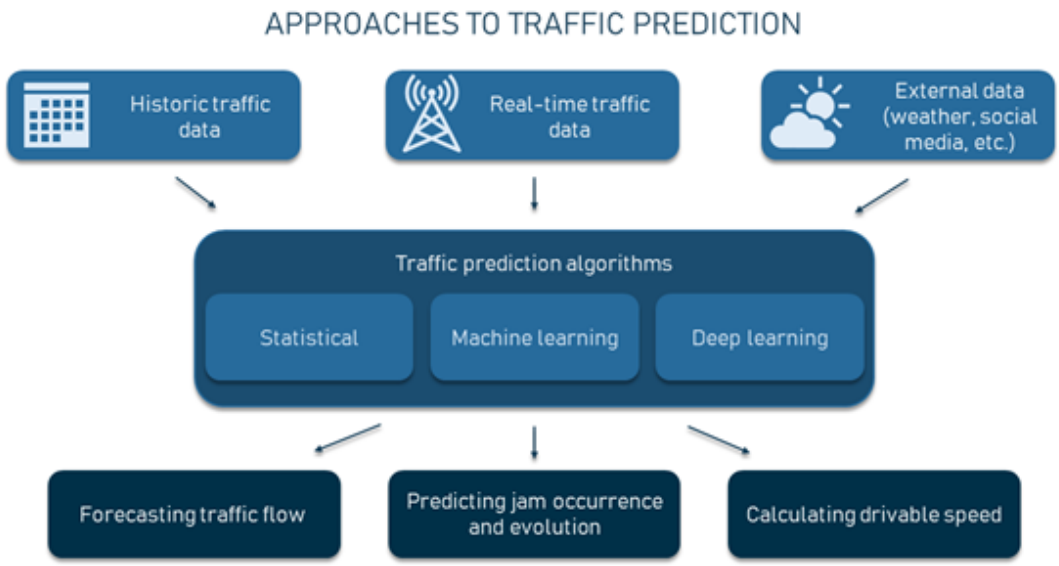
- Cổng quên (forget gate): Xác định thông tin nào sẽ được loại bỏ khỏi đơn vị nhớ.
- Cổng đầu vào (input gate): Xác định thông tin mới nào sẽ được thêm vào đơn vị nhớ.
- Cổng đầu ra (output gate): Xác định thông tin nào sẽ được truyền ra khỏi đơn vị nhớ để đóng góp vào dự đoán hoặc quá trình tiếp theo.

Nhờ vào cơ chế cổng này, LSTM có khả năng giữ và sử dụng thông tin quan trọng từ quá khứ trong quá trình dự đoán, giúp cải thiện khả năng mô hình hóa các phụ thuộc xa trong dữ liệu tuần tự.

LSTM đã được sử dụng rộng rãi trong nhiều lĩnh vực như xử lý ngôn ngữ tự nhiên, dịch máy, nhận dạng giọng nói, dự đoán chuỗi thời gian, và nhiều ứng dụng khác. Nó đã chứng minh khả năng mô hình hóa và dự đoán tốt trong các tác vụ liên quan đến dữ liệu tuần tự và dữ liệu có mối quan hệ thời gian.

CHƯƠNG 2: XÂY DỰNG THUẬT TOÁN VÀ MÔ HÌNH

Dự đoán giao thông bao gồm dự báo tốc độ có thể lái được trên các đoạn đường cụ thể, cũng như tình trạng tắc đường và diễn biến. Chúng ta hãy xem xét các cách tiếp cận khác nhau cho nhiệm vụ này.



Hình 7 Approaches To Traffic Prediction

2.1. Phương pháp Backpropagation

Backpropagation (phản hồi ngược) là một thuật toán quan trọng được sử dụng trong huấn luyện mạng nơ-ron để tính toán đạo hàm của hàm mất mát theo các trọng số của mô hình. Thuật toán này cho phép chúng ta điều chỉnh các trọng số để giảm thiểu lỗi và cải thiện hiệu suất của mô hình. Dưới đây là mô tả chi tiết về thuật toán backpropagation:

Lan truyền tiến (Forward Propagation):
Đầu tiên, chúng ta thực hiện lan truyền tiến để tính toán đầu ra của mô hình dựa trên dữ liệu đầu vào và các trọng số hiện tại của mô hình.
Dữ liệu được truyền qua mạng nơ-ron từ lớp đầu vào đến lớp cuối cùng, và mỗi lớp tính toán đầu ra của nó bằng cách áp dụng hàm kích hoạt tương ứng và tính tổng trọng số đầu vào của nó.

Tính toán độ lỗi (Compute Loss):

Sau khi có đầu ra dự đoán của mô hình, chúng ta tính toán độ lỗi bằng cách so sánh đầu ra dự đoán với đầu ra thực tế sử dụng hàm mất mát (trong trường hợp của bạn, hàm mất mát là MeanSquaredError).
Hàm mất mát đo lường sự sai khác giữa đầu ra dự đoán và đầu ra thực tế và trả về giá trị độ lỗi.

Lan truyền ngược (Backward Propagation):
Tiếp theo, chúng ta thực hiện lan truyền ngược để tính toán đạo hàm của hàm mất mát theo các trọng số của mô hình.
Lan truyền ngược bắt đầu từ lớp cuối cùng và di chuyển ngược lên các lớp trước đó trong mạng nơ-ron.
Ở mỗi lớp, chúng ta tính toán đạo hàm của hàm mất mát theo đầu ra của lớp và áp dụng quy tắc chuỗi để tính toán đạo hàm theo các trọng số của lớp.
Đạo hàm này cho biết làm thế nào sự thay đổi của mỗi trọng số ảnh hưởng đến độ lỗi của mô hình.

Cập nhật trọng số (Weight Update):
Sau khi tính toán được đạo hàm của hàm mất mát theo các trọng số, chúng ta có thể cập nhật các trọng số để giảm thiểu độ lỗi và cải thiện hiệu suất của mô hình.
Cập nhật trọng số được thực hiện bằng cách di chuyển ngược qua mạng nơ-ron và điều chỉnh các trọng số theo hướng giảm độ lỗi.
Đạo hàm theo trọng số chỉ ra cách thay đổi của trọng số ảnh hưởng đến độ lỗi của mô hình.

Thuật toán cập nhật trọng số thường sử dụng phương pháp gradient descent (gradient giảm dần) để điều chỉnh các trọng số dựa trên đạo hàm.

Một thuật toán phổ biến trong gradient descent là thuật toán Stochastic Gradient Descent (SGD), trong đó các trọng số được cập nhật dựa trên từng mẫu dữ liệu đầu vào một cách ngẫu nhiên.

Công thức cập nhật trọng số trong gradient descent là:
weight_new = weight_old - learning_rate * gradient,
trong đó learning_rate là tỷ lệ học (learning rate) quyết định tốc độ cập nhật trọng số.

Lặp lại quá trình huấn luyện:
Quá trình lan truyền tiến, tính toán độ lỗi, lan truyền ngược và cập nhật trọng số được lặp lại cho mỗi mẫu huấn luyện trong tập huấn luyện và trong nhiều epoch.
Mỗi epoch đại diện cho một lần chạy qua toàn bộ tập dữ liệu huấn luyện.
Mục tiêu là giảm thiểu độ lỗi qua các epoch để mô hình hội tụ và đạt được hiệu suất tốt trên dữ liệu huấn luyện và dữ liệu xác thực.
Thông qua quá trình backpropagation, mạng nơ-ron có khả năng cập nhật các trọng số để tìm ra các giá trị tối ưu và cải thiện hiệu suất của mô hình trong việc dự đoán chuỗi thời gian.

2.2. Phương pháp thống kê

Sử dụng các kỹ thuật thống kê, em có thể nhận ra các mô hình giao thông ở các tỷ lệ khác nhau, chẳng hạn như trong ngày, vào các ngày trong tuần, theo mùa, vv. So với các phương pháp học máy, chúng thường nhanh hơn, đơn giản và dễ thực hiện hơn. Tuy nhiên, vì chúng không xử lý được nhiều dữ liệu đa biến hơn, nên độ chính xác của chúng thấp hơn.

Từ những năm 1970, mô hình trung bình chuyển động tích hợp tự hồi quy (ARIMA) - được dễ dàng sử dụng và hiển thị độ chính xác cao hơn so với các kỹ thuật thống kê khác - đã được sử dụng rộng rãi để dự đoán giao thông.

Nó sử dụng phương pháp thống kê truyền thống để phân tích quá khứ và dự báo tương lai. Nó thu thập dữ liệu từ một loạt khoảng thời gian thường xuyên và giả định

rằng các mô hình quá khứ sẽ tiếp tục trong tương lai. Tuy nhiên, luồng giao thông là một cấu trúc phức tạp với nhiều biến không thể xử lý hiệu quả bằng các mô hình ARIMA một biến.

Để tính toán và thống kê dữ liệu em sử dụng hàm **describe()** trong Python. Hàm **describe()** trong thư viện pandas của Python được sử dụng để tạo ra một tóm tắt thống kê mô tả cho dữ liệu số học trong một Series hoặc DataFrame. Hàm này cung cấp các thông tin quan trọng về phân phối, trung bình, độ lệch chuẩn và các phân vị của dữ liệu.

Cú pháp:

dataframe.describe()

Hàm **describe()** trả về một DataFrame chứa các thông số thống kê mô tả như:

- **COUNT**: Là tổng số lượng mẫu không bị thiếu trong cột. Nó được tính bằng cách đếm số lượng giá trị không phải NaN (Not a Number) trong cột

  Công thức tính count:

  count = số lượng giá trị không phải NaN trong cột

Lưu ý rằng NaN thường được sử dụng để biểu thị giá trị bị thiếu hoặc không xác định trong dữ liệu. Khi tính toán count, NaN sẽ không được tính trong số lượng mẫu.

- **MEAN**: là tổng của các giá trị trong cột chia cho tổng số mẫu.

  Công thức tính mean:

  mean = tổng các giá trị / tổng số mẫu

- **STD**: Độ lệch chuẩn, đo lường mức độ phân tán của dữ liệu, là một đại lượng thống kê được sử dụng để đo độ biến thiên của dữ liệu so với giá trị trung bình. Nó cho biết mức độ phân tán của dữ liệu quanh giá trị trung bình.

  Công thức tính độ lệch chuẩn:

1. Tính giá trị trung bình của dữ liệu (mean).
2. Tính khoảng cách của mỗi giá trị đến giá trị trung bình.
3. Bình phương các khoảng cách đã tính.
4. Tính trung bình của các bình phương khoảng cách.

5. Lấy căn bậc hai của trung bình bình phương khoảng cách để tính toán độ lệch chuẩn.

Công thức toán học:

$$std = sqrt((1/N) * sum((x\_i - mean)^2))$$

Trong đó:

- std là độ lệch chuẩn.
- N là tổng số mẫu.
- x_i là giá trị mẫu thứ i trong dữ liệu.
- mean là giá trị trung bình của dữ liệu.

Độ lệch chuẩn cho biết mức độ phân tán của dữ liệu. Giá trị std càng lớn, dữ liệu càng phân tán rộng từ giá trị trung bình. Ngược lại, giá trị std càng nhỏ, dữ liệu càng gần giá trị trung bình.

Độ lệch chuẩn là một đại lượng quan trọng trong phân tích thống kê và được sử dụng rộng rãi để mô tả tính biến động của dữ liệu.

- **MIN**: Giá trị nhỏ nhất trong cột.
- **25%, 50%, 75%:** Các phân vị 25%, 50% (hoặc median) và 75% của dữ liệu.

Giải thích rõ hơn về cách tính các phân vị (Q1, Q2, Q3) trong dữ liệu:

1. Phân vị 25% (Q1):
   - Đầu tiên, sắp xếp dữ liệu theo thứ tự tăng dần.
   - Tính vị trí của phân vị 25% trong dữ liệu: vị trí = 0.25 * (số lượng mẫu + 1).
   - Nếu vị trí không là một số nguyên, ta lấy giá trị trung bình của hai giá trị xung quanh vị trí để xác định vị trí thực tế.

Lấy giá trị tại vị trí đã tính là giá trị của phân vị 25% (Q1).

2. Phân vị 50% (Q2) hay giá trị trung vị:
   - Đầu tiên, sắp xếp dữ liệu theo thứ tự tăng dần.
   - Tính vị trí của phân vị 50% trong dữ liệu: vị trí = 0.50 * (số lượng mẫu + 1).
   - Nếu vị trí không là một số nguyên, ta lấy giá trị trung bình của hai giá trị xung quanh vị trí để xác định vị trí thực tế.
   - Lấy giá trị tại vị trí đã tính là giá trị của phân vị 50% (Q2).

3. Phân vị 75% (Q3):

- Đầu tiên, sắp xếp dữ liệu theo thứ tự tăng dần.
- Tính vị trí của phân vị 75% trong dữ liệu: vị trí = 0.75 * (số lượng mẫu + 1).
- Nếu vị trí không là một số nguyên, ta lấy giá trị trung bình của hai giá trị xung quanh vị trí để xác định vị trí thực tế.
- Lấy giá trị tại vị trí đã tính là giá trị của phân vị 75% (Q3).

Tóm lại, để tính các phân vị 25%, 50% và 75% trong dữ liệu, ta cần sắp xếp dữ liệu, xác định vị trí tương ứng và lấy giá trị tại vị trí đó. Các phân vị này giúp chúng ta hiểu rõ hơn về phân phối và độ biến động của dữ liệu trong một tập hợp.

- **MAX:** Giá trị lớn nhất trong cột.

Hàm **describe()** chỉ tính toán thống kê mô tả cho các cột số học và bỏ qua các cột không phải số học trong kết quả. Nó thường được sử dụng để có cái nhìn tổng quan về phân phối và các giá trị cơ bản của dữ liệu.

Ví dụ:

```
def describe(df):
    return pd.concat([df.describe().T, df.skew().rename('skew'),], axis=1)

describe(df_raw)
```

Hình 8 Hàm Describe

Đoạn code trên định nghĩa một hàm có tên là ‘describe’, nhận đối số là một DataFrame `df`. Hàm này được sử dụng để tính toán các thông số mô tả thống kê và độ lệch của các cột trong DataFrame.

Cụ thể, hàm `describe` sử dụng hai phương thức của DataFrame:

‘df.describe()’ - Phương thức này tính toán các thông số mô tả thống kê của các cột trong DataFrame, bao gồm số lượng mục, giá trị trung bình, độ lệch chuẩn, giá trị tối thiểu, các phần vị và giá trị tối đa. Kết quả của phương thức này là một DataFrame với các thông số mô tả thống kê cho từng cột.

`df.skew()` - Phương thức này tính toán độ lệch (skewness) của các cột trong DataFrame. Độ lệch là một đo lường cho hình dạng của phân phối dữ liệu. Kết quả của phương thức này là một Series chứa độ lệch của mỗi cột.

Sau đó, hàm `**pd.concat()**` được sử dụng để kết hợp hai kết quả trên thành một DataFrame duy nhất. Tham số `axis=1` chỉ định rằng việc kết hợp sẽ được thực hiện theo chiều cột.

Cuối cùng, khi gọi '**describe(df_raw)',** chúng ta truyền DataFrame **'df_raw'** vào hàm **'describe'** để tính toán các thông số mô tả thống kê và độ lệch của các cột trong DataFrame. Kết quả là một DataFrame chứa thông tin mô tả thống kê và độ lệch của các cột.

Kết quả thu được:

	count	mean	std	min	25%	50%	75%	max	skew
temp	48204.0	281.205870	13.338232	0.0	272.16	282.45	291.806	310.07	-2.247226
rain_1h	48204.0	0.334264	44.789133	0.0	0.00	0.00	0.000	9831.30	219.389036
snow_1h	48204.0	0.000222	0.008168	0.0	0.00	0.00	0.000	0.51	48.367484
clouds_all	48204.0	49.362231	39.015750	0.0	1.00	64.00	90.000	100.00	-0.197257
traffic_volume	48204.0	3259.818355	1986.860670	0.0	1193.00	3380.00	4933.000	7280.00	-0.089381

Hình 9 Kết quả dùng hàm Describe

2.3. Phương pháp học máy

Em có thể xây dựng các mô hình dự đoán bằng cách sử dụng học máy (ML) để xem xét lượng lớn dữ liệu không đồng nhất từ nhiều nguồn khác nhau. Việc sử dụng các thuật toán ML để dự đoán giao thông đã được nghiên cứu rất nhiều. Dưới đây là một số ví dụ hiệu quả.

Phương pháp ngẫu nhiên rừng (Random Forest) xây dựng nhiều cây quyết định và kết hợp dữ liệu của chúng để tạo ra dự báo chính xác. Với đủ dữ liệu huấn luyện, nó có thể tạo ra kết quả hiệu quả nhanh chóng. Trong trường hợp này, biến đầu vào của mô hình bao gồm thời tiết, thời gian, điều kiện đường cụ thể, chất lượng đường và ngày lễ.

Ngoài ra, phương pháp k trong thuật toán K-Nearest Neighbor (KNN) sử dụng ý tưởng về độ tương đồng của các đặc trưng để dự đoán giá trị trong tương lai.

2.4. Phương pháp học sâu

Phương pháp học sâu (DL) đã chứng minh hiệu quả cao trong dự đoán giao thông so với các phương pháp ML hoặc thống kê, luôn cho thấy độ chính xác dự đoán khoảng 90% trở lên. Các thuật toán DL dựa trên mạng nơ-ron.

Mạng nơ-ron nhân tạo (ANN) hoặc mạng nơ-ron (NN) được tạo thành từ các nút (neuron) tương tác được sắp xếp trong hai hoặc nhiều lớp và được thiết kế để mô phỏng hành vi của não người. Có nhiều loại mạng nơ-ron được phát triển cho các mục đích khác nhau. Dưới đây là một số mạng nơ-ron đã được sử dụng trong phân tích và dự đoán giao thông.

Mạng nơ-ron tích chập (CNN) được coi là những người tiên phong trong việc phân tích và nhận dạng hình ảnh. Sử dụng hình ảnh từ camera giám sát trên đường để phát hiện tắc nghẽn là một trong những ứng dụng tự nhiên của CNN trong các vấn đề giao thông. CNN không phải là lựa chọn hàng đầu cho dự báo giao thông. Tuy nhiên, những nỗ lực phát triển mô hình dựa trên CNN để dự đoán tốc độ mạng giao thông đã rất hiệu quả. Để làm điều này, các nhà nghiên cứu đã tạo ra một ma trận hình ảnh hai chiều từ dữ liệu thời gian và không gian mô tả lưu lượng giao thông.

Mạng nơ-ron hồi quy dài hạn (RNN), so với mạng nơ-ron tích chập (CNN), được thiết kế để phân tích dữ liệu chuỗi thời gian hoặc quan sát thu thập trong các khoảng thời gian cụ thể. Những hiểu biết này có thể được thấy rõ trong các mô hình dự đoán giao thông. Mô hình RNN đã chứng minh khả năng dự đoán sự phát triển tắc nghẽn với độ chính xác cao. Vấn đề biến mất gradient, đó là lý do tại sao RNN được coi là "có bộ nhớ ngắn hạn", là nhược điểm của nó vì nó làm mất một số dữ liệu từ các lớp trước. Do đó, việc huấn luyện mô hình khó khăn và tốn thời gian hơn.

LSTM (Long short-term memory) và GRU (Gated Recurrent Unit) là các biến thể của RNN giải quyết vấn đề biến mất gradient. Một nghiên cứu so sánh hiệu suất của các mô hình này cho thấy mô hình GRU có độ chính xác cao hơn trong dự đoán lưu lượng giao thông và dễ huấn luyện hơn.

Nhiều nghiên cứu đã đề xuất phát triển các mô hình NN khác nhau cho dự đoán giao thông, bao gồm mạng nơ-ron đồ thị (Graph neural networks), mạng nơ-ron mờ (fuzzy NN), mạng nơ-ron Bayesian và các phương pháp kết hợp hai hoặc nhiều thuật toán. Hiện tại, chưa có một kỹ thuật duy nhất lý tưởng có thể được sử dụng trong tất cả các tình huống để tạo ra những dự đoán chính xác nhất.

Còn một vài điều nữa cần đề cập khi thực hiện các kỹ thuật ML cho dự đoán giao thông. Chúng ta phải nhớ rằng các thuật toán ML/DL hoạt động tốt nhất khi có đủ dữ liệu để huấn luyện mô hình và điều chỉnh chúng để đạt được độ chính xác tối đa. Do đó, chúng ta càng có được tập dữ liệu lớn, kết quả càng tốt.

CHƯƠNG 3: PHÂN TÍCH VÀ XỬ LÝ DỮ LIỆU

3.1. Bộ dữ liệu

Hiện tại mô hình của em sử dụng bộ dữ liệu "Metro Interstate"

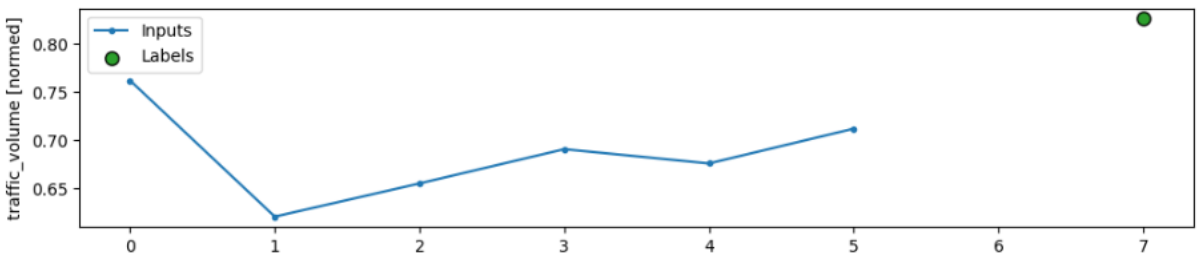
Bộ dữ liệu về lưu lượng giao thông liên bang chứa thông tin về lưu lượng giao thông hàng giờ trên làn đường hướng Tây của Xa lộ Liên tiểu bang-94 (I-94) ở Hoa Kỳ. Bộ dữ liệu bao gồm các báo cáo thời tiết và nhiệt độ hàng giờ từ năm 2012 đến 2018.

Thông tin trong bộ dữ liệu có thể được sử dụng để hiểu lưu lượng giao thông trên đường liên bang theo ngày giờ và có thể hữu ích trong việc dự đoán giờ cao điểm, dự báo thời tiết cũng như lập kế hoạch mở rộng đường liên bang và đường cao tốc ở Hoa Kỳ.

Hơn nữa, các tính năng thời tiết hàng giờ và ngày lễ cũng được đưa vào để xem xét các tác động đến lưu lượng giao thông.

3.2. Phân tích và xử lý dữ liệu

Bộ dữ liệu lưu lượng giao thông liên bang Metro là một lưu lượng giao thông ở Xa lộ Liên tiểu bang 94 cho trạm MN DOT ATR 301, gần giữa Minneapolis và St Paul, MN.Các tính năng thời tiết hàng giờ và ngày lễ được đưa vào để tác động đến lưu lượng giao thông.Mục đích chính của em là xây dựng một RNN nhiều bước với mô hình LSTM đưa ra một điểm dự đoán duy nhất về khối lượng lưu lượng truy cập 2 giờ trong tương lai, được đưa ra cửa sổ 6 giờ trước đó.Điều này có thể được chứng minh trong (Hình), trong đó các đầu vào biểu thị điểm dữ liệu trong 6 giờ đã cho và nhãn là đầu ra dự kiến 2 giờ sau đó.



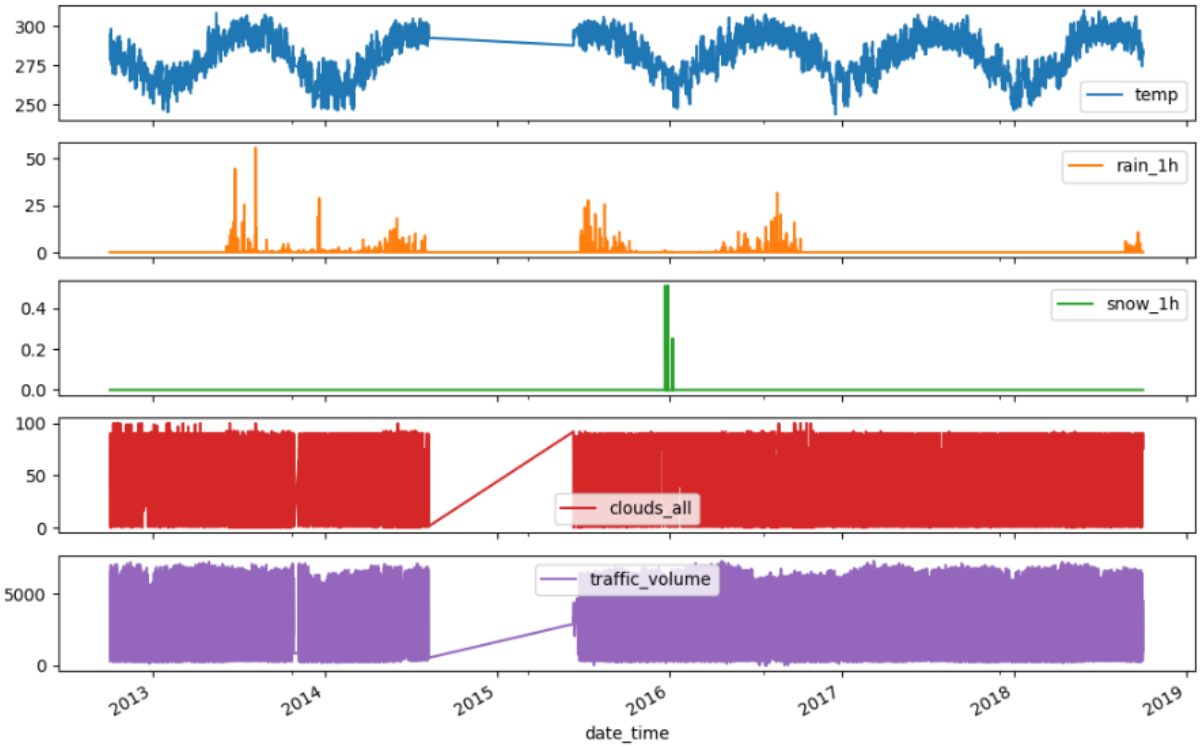
Hình 10 Biểu thị điểm dữ liệu trong 6 giờ đã cho và nhãn

Tại quá trình xử lý dữ liệu ban đầu, gần 7629 mục trùng lặp hàng giờ đã được tìm thấy trong bộ dữ liệu, điều đó có nghĩa là lưu lượng lưu lượng được lặp lại nhiều lần trong cùng một ngày.

Ban đầu, em nhận thức được điều đó, nhưng em đã quyết định coi bộ dữ liệu là một giờ cho mỗi bản ghi, không phải là chuỗi thời gian thực tế với bộ dữ liệu được lập chỉ mục đơn vị thời gian, nhưng điều này dẫn đến kết quả xác thực khiêm tốn (MAE giữa 300s).Sau đó, em quyết định xử lý trước bộ dữ liệu đào tạo và xác thực đúng cách khi thời gian được lập chỉ mục trong khoảng thời gian 1 giờ và kết quả xác nhận tốt hơn nhiều (MAE 200s thấp).

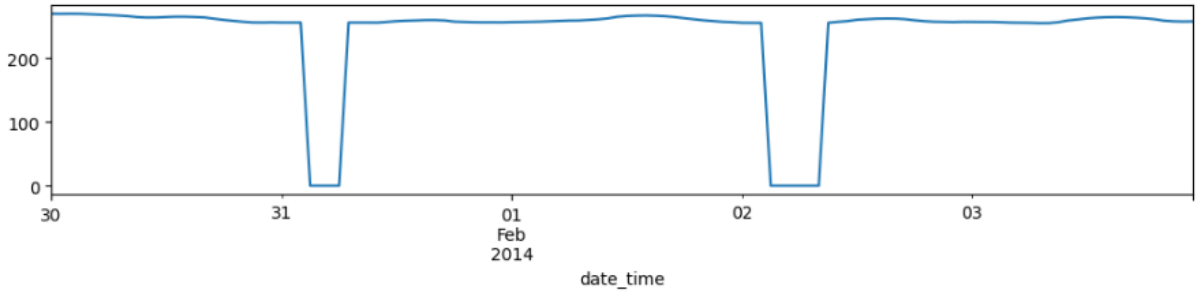
Lateron, em chia dữ liệu thành đào tạo, xác nhận và thử nghiệm.Để theo dõi các bản ghi thuộc bộ dữ liệu phù hợp.

Hơn nữa, trong quá trình xử lý trước dữ liệu dữ liệu đã được tìm thấy từ năm 2014 đến 2015 (10 tháng) và các bản ghi xác thực được thực hiện vì một khoảng cách có thể được quan sát rõ ràng giữa dữ liệu như trong hình dưới đây.

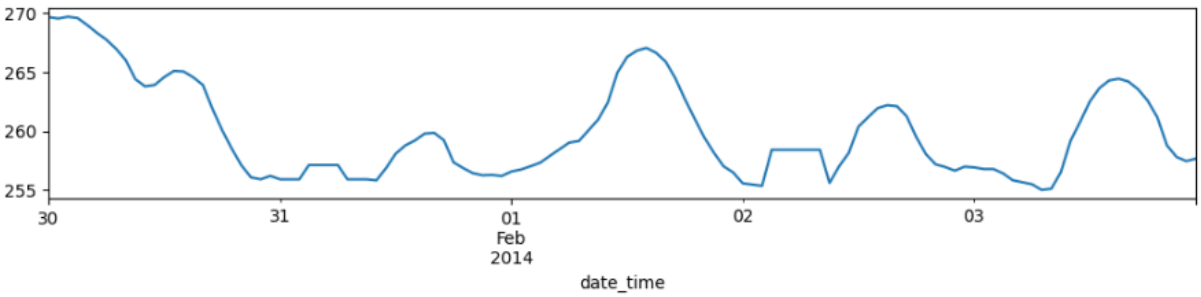


Hình 11 Quá trình xử lý dữ liệu

Em đã sử dụng các neuron để đối phó với nó và đối với các ngoại lệ đã được xử lý trong một vài hồ sơ, ví dụ như hai ngày có trường nhiệt độ được đặt ở mức 0 của mỗi ngày.



Hình 12 Xử lý ngoại lệ



Hình 13 Xử lý ngoại lệ

Rain Field cũng có một giá trị cực trị duy nhất, mà em cũng đặt thành giá trị trung bình của ngày hôm đó. Tuyết có các giá trị cực đoan, nhưng em không thể xác định xem các giá trị có phải là ngoại lệ hay không. Đó có thể là một mùa đông đặc biệt, và vì em không biết về điều kiện thời tiết trong khu vực đó, em quyết định giữ nó như vậy.

Hơn nữa, trong quá trình chuyển đổi dữ liệu và kỹ thuật tính năng,em đã quyết định chuyển đổi Weather_Main thành các biến được mã hóa một lần và bỏ mô tả Weather_Descript khi em thấy rằng nó thêm một loại thông tin dư thừa với Weather_Main.Ngoài ra, em nghĩ rằng thông tin có giá trị để nắm bắt là liệu ngày là ngày lễ hay cuối tuần.Chúng ta không cần phải theo dõi kỳ nghỉ nào.Vì vậy, một tính năng mới is_holiday được tạo ra và tính năng ngày lễ trong dữ liệu cũ đã bị loại bỏ. Tương tự, chúng ta không cần phải theo dõi nó vào cuối tuần.Vì vậy, em đã tạo một tính năng mới, is_weekend.Trường DATE_TIME được chuyển đổi thành tín hiệu sử dụng Sin và COS để chuyển đổi thời gian để xóa tín hiệu "thời gian trong ngày" và "thời gian

trong năm".Điều này cho phép truy cập mô hình vào các tính năng tần số quan trọng nhất.

Cuối cùng, em đã chia các thành phần DATE_TIME thành các trường nguyên tố khác, Dayofweek, Day, Tháng, Year và Day_Hour.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 48204 entries, 2012-10-02 09:00:00 to 2018-09-30 23:00:00
Data columns (total 27 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   traffic_volume       48204 non-null  int64
 1   Day sin              48204 non-null  float64
 2   Day cos              48204 non-null  float64
 3   Year sin             48204 non-null  float64
 4   Year cos             48204 non-null  float64
 5   temp                 48204 non-null  float64
 6   clouds_all           48204 non-null  int64
 7   rain_1h              48204 non-null  float64
 8   snow_1h              48204 non-null  float64
 9   is_weekend           48204 non-null  int64
 10  is_holiday           48204 non-null  int64
 11  weather_Clear        48204 non-null  uint8
 12  weather_Clouds       48204 non-null  uint8
 13  weather_Drizzle      48204 non-null  uint8
 14  weather_Fog          48204 non-null  uint8
 15  weather_Haze         48204 non-null  uint8
 16  weather_Mist         48204 non-null  uint8
 17  weather_Rain         48204 non-null  uint8
 18  weather_Smoke        48204 non-null  uint8
 19  weather_Snow         48204 non-null  uint8
 20  weather_Squall       48204 non-null  uint8
 21  weather_Thunderstorm 48204 non-null  uint8
 22  dayofweek            48204 non-null  int64
 23  day                  48204 non-null  int64
 24  month                48204 non-null  int64
 25  year                 48204 non-null  int64
 26  day_hour             48204 non-null  int64
dtypes: float64(7), int64(9), uint8(11)
memory usage: 7.8 MB
```

Hình 14 Dữ liệu sau khi đã xử lý

Trong phần Kiểu chỉ số chuỗi thời gian, như đã đề cập, đã có 7629 các mục trùng lặp hàng giờ.Và em đã quyết định sửa lỗi này bằng cách lấy mẫu lại dữ liệu trên cơ sở 1 giờ để mỗi bản ghi chỉ giống với một giờ.Hồ sơ giờ trùng lặp được tính trung bình trong cùng một giờ.Các phép biến đổi được thực hiện sau khi bộ dữ liệu chia thành đào tạo, xác nhận và thử nghiệm.

Em đã có kết quả tốt nhất bằng cách sử dụng bình thường hóa Min-Max so với tỷ lệ tiêu chuẩn. Đối với hầu hết các thử nghiệm, em nhằm mục đích đánh giá hiệu quả của việc điều chỉnh từng siêu đồng tính bằng hiệu suất của các mô hình dựa trên RNN và LSTM. Em quyết định không sử dụng các phương pháp tìm kiếm siêu phân tích tự động để hiểu rõ hơn về cách mỗi siêu đồng tính ảnh hưởng đến mô hình.

Cho rằng việc tiền xử lý dữ liệu đã tiêu thụ nhiều giờ, em phải áp dụng các phương pháp nhanh chóng và vẩn để giúp em hiểu cách các mạng LSTM hoạt động, thay thế cho đánh giá hệ thống tự động lai em đã áp dụng thử nghiệm thủ công hệ thống của các siêu âm bằng cách sử dụng các giá trị cực nhỏ và lớn cho số lượng đơn vị LSTM. Sau đó, em đã thêm các loại lớp khác như tích chập, dày đặc, GRU và các lớp hai chiều. Em cũng đã thử nghiệm với các kích thước lô khác nhau. Sau đó, em bắt đầu trộn và kết hợp các quan sát của em dựa trên cách các lớp và các siêu âm đã ảnh hưởng đến mô hình.

Dataset split before timeseries resampling:

train_df:	33204
val_df:	10000
test_df:	5000

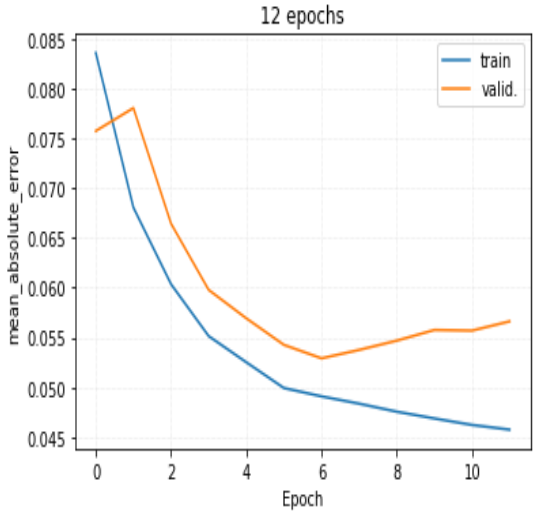
Dataset split after 1-hour timeseries resampling:

train_df:	40110
val_df:	8359
test_df:	4083

CHƯƠNG 4: KẾT QUẢ CỦA XÂY DỰNG MÔ HÌNH

4.1. Mô hình Dense

```
Training vs. Validation:
```

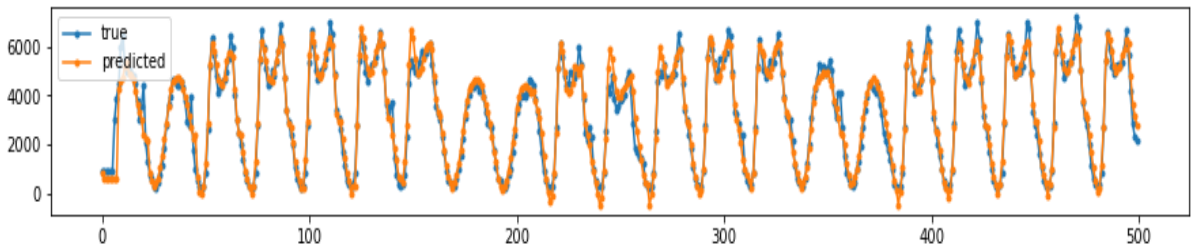



```
Validation Scores:

261/261 - 0s - loss: 0.0054 - mean_absolute_error: 0.0529

Predictions Evaluation:

Predictions: 8352
MAE: 385.16 (0.0529)
```

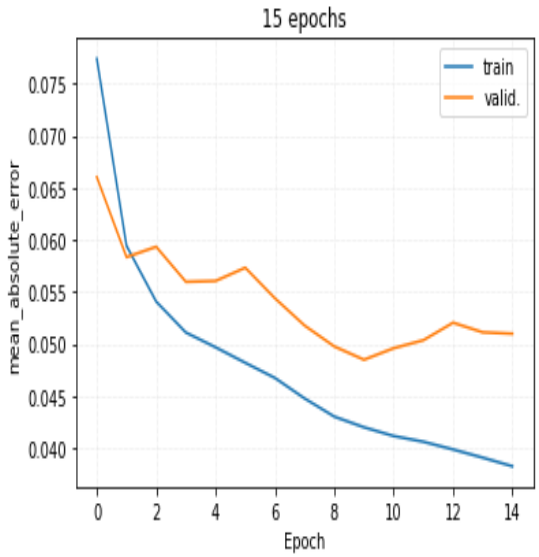



```
Wall time: 42.1 s
```

Hình 15 Kết quả của mô hình Dense

4.2. Mô hình Tích Chập Conv(CNN)

Mô hình tích chập đưa ra dự đoán dựa trên lịch sử chiều rộng cố định, điều này có thể dẫn đến hiệu suất tốt hơn mô hình dày đặc vì nó có thể thấy mọi thứ đang thay đổi như thế nào theo thời gian

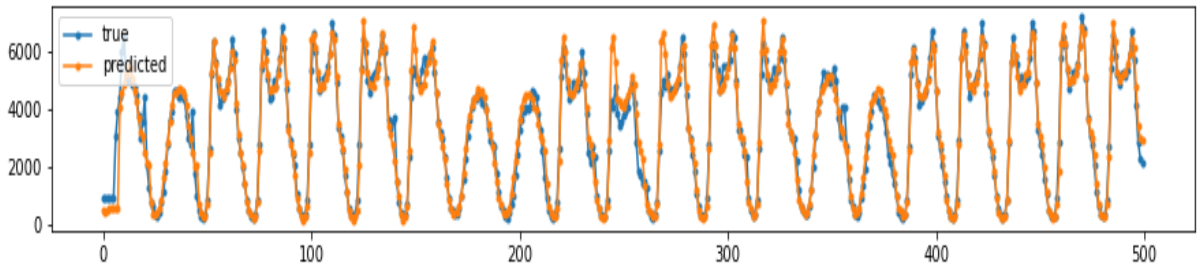



```
Validation Scores:

261/261 - 1s - loss: 0.0048 - mean_absolute_error: 0.0485

Predictions Evaluation:

Predictions: 8352
MAE: 352.99 (0.0485)
```

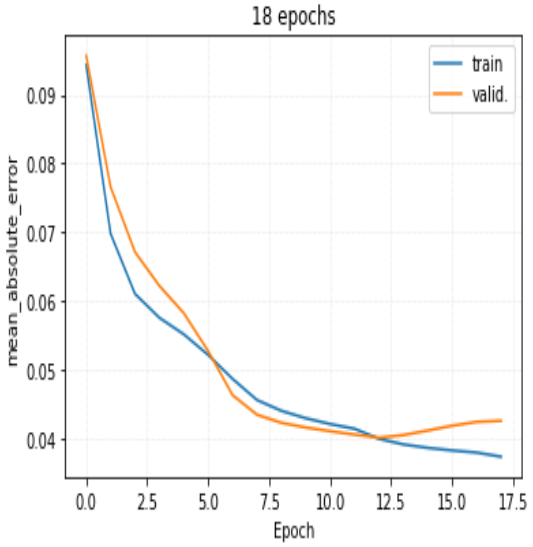



```
Wall time: 1min 12s
```

Hình 16 Kết quả của mô hình CNN

4.3. Mô hình LSTM(RNN)

```
Training vs. Validation:
```

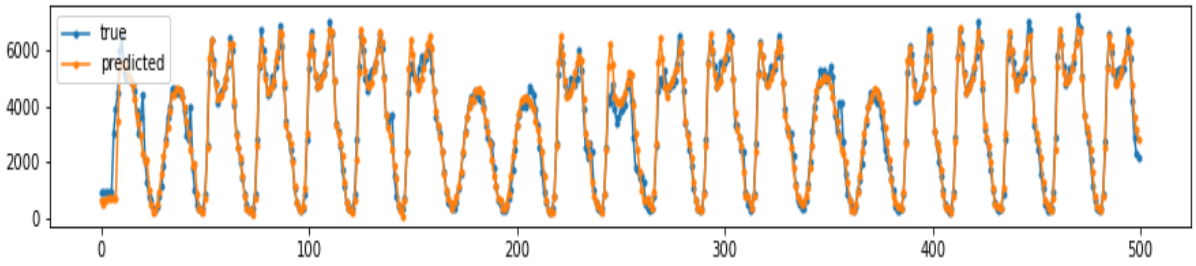



```
Validation Scores:

261/261 - 1s - loss: 0.0034 - mean_absolute_error: 0.0401

Predictions Evaluation:

Predictions: 8352
MAE: 291.92 (0.0401)
```

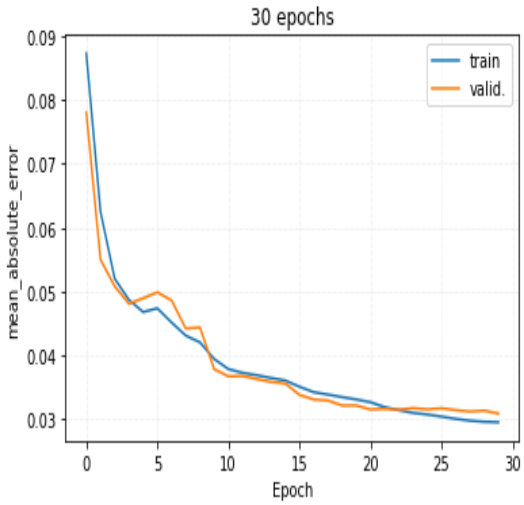



```
Wall time: 2min 5s
```

Hình 17 Kết quả của mô hình RNN

4.4. Mô hình My Models

Training vs. Validation:

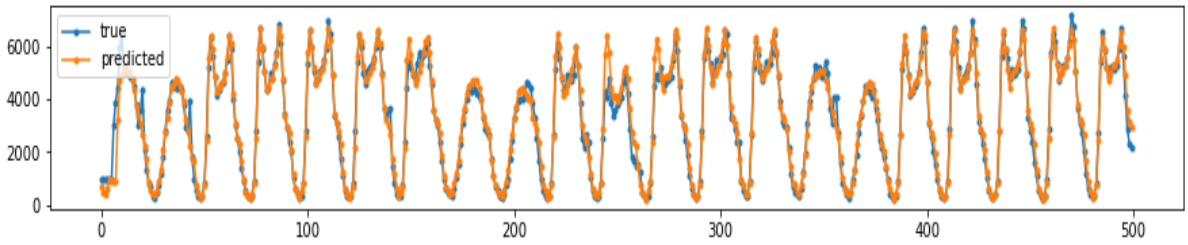


Validation Scores:

261/261 - 12s - loss: 0.0023 - mean_absolute_error: 0.0309

Predictions Evaluation:

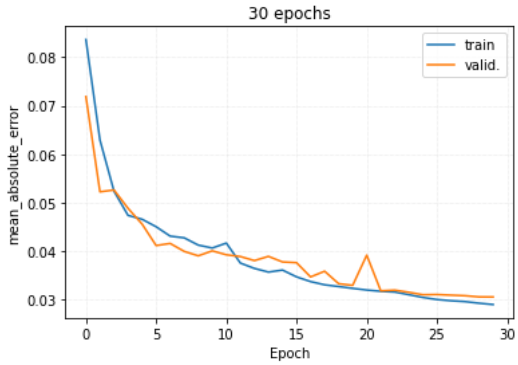
Predictions: 8352
MAE: 224.94 (0.0309)



Wall time: 1h 38min 1s

Hình 18 Kết quả của mô hình mylstm_1

Training vs. Validation:

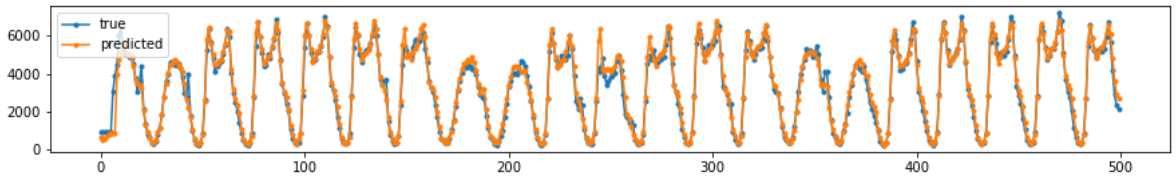


Validation Scores:

261/261 - 24s - loss: 0.0021 - mean_absolute_error: 0.0306

Predictions Evaluation:

Predictions: 8352
MAE: 222.69 (0.0306)



Wall time: 3h 48min 24s

Hình 19 Kết quả của mô hình mylstm_2

4.5. So sánh hiệu suất mô hình

So sánh hiệu suất của mô hình của em với các thuật toán mới khác

- LSTM (Mạng nơ-ron hồi quy dài hạn)

Dự đoán lưu lượng giao thông trong thời gian ngắn là rất quan trọng đối với các hệ thống giao thông thông minh, và thông tin giao thông không gian và thời gian bổ sung có thể được sử dụng trong các nghiên cứu tương lai để ước tính chính xác lưu lượng giao thông trên một mạng lớn hơn. Nhìn chung, mô hình LSTM có nhiều lớp sẽ cải thiện khả năng học của mô hình, nhưng cũng dễ bị Overfitting.

- CNN (Mạng nơ-ron tích chập)

CNN, hay mạng nơ-ron tích chập, thêm các lớp "lọc" bổ sung trong đó trọng số của bộ lọc được xác định. Backpropagation vẫn thực hiện nhiệm vụ này cho chúng ta, nhưng sẽ rất không khăn cho việc thực hiện thuật toán Backpropagation.

Một CNN có nhiều bộ lọc song song có thể được điều chỉnh để trích xuất các khía cạnh quan trọng khác nhau.

- CNN so với LSTM

CNN có thể được sử dụng để giảm số lượng tham số cần thiết cho việc huấn luyện trong khi vẫn duy trì hiệu suất - đây là sức mạnh của việc kết hợp xử lý tín hiệu với học sâu, trong khi LSTM yêu cầu nhiều tham số hơn CNN, lợi ích của nó đến từ khả năng kiểm tra các chuỗi đầu vào dài mà không phải mở rộng kích thước mạng.

- LSTM RNN (Mạng nơ-ron hồi quy dài hạn)

Kiến trúc RNN truyền thống có vấn đề về Vanishing Gradient. Để vượt qua nhược điểm này, các cấu trúc của RNN như LSTM được đề xuất, được thiết kế để cho phép các ô nhớ quyết định khi nào nên quên một số thông tin cụ thể, từ đó xác định các mốc thời gian tối ưu cho các vấn đề chuỗi thời gian. Những đặc điểm này đặc biệt mong muốn trong dự đoán lưu lượng giao thông trong thời gian ngắn trong lĩnh vực giao thông vận tải vì khả năng ghi nhớ lâu của nó. Sử dụng LSTM RNN cho dự đoán lưu lượng giao thông và cho thấy rằng LSTM RNN có hiệu suất tốt hơn hầu hết các mô hình phi tham số.

CHƯƠNG 5: KẾT LUẬN

5.1. Kết quả

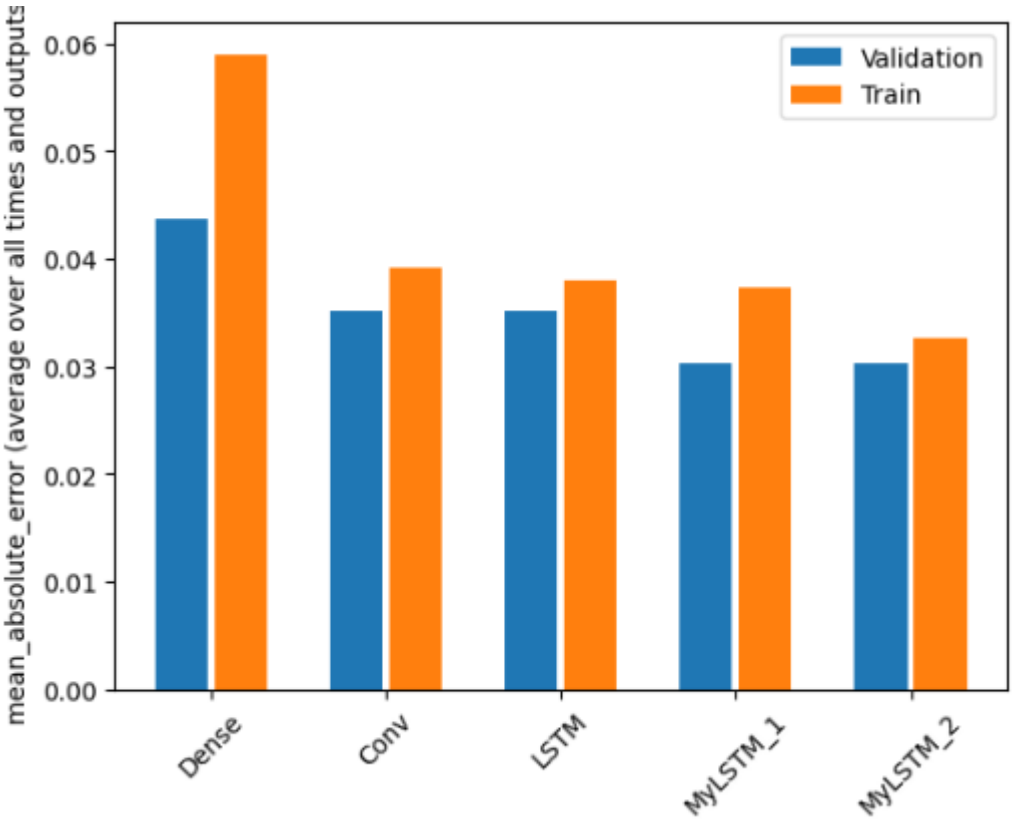
Em đã cố gắng tìm các kiến trúc mô hình LSTM nổi tiếng như trước đây em đã tìm thấy cho các mạng thần kinh tích chập.Tuy nhiên, em không thể tìm thấy bất cứ điều gì ngoài một vài lớp LSTM là tài liệu tham khảo.

Hyperparameter đã được kiểm tra một cách có hệ thống bằng các giá trị khác nhau là:

- Số lượng các đơn vị và lớp LSTM, và các đơn vị dày đặc ở các lớp cuối cùng (Mô hình 1)
- LSTM hai chiều (Mô hình 2)

Hơn nữa, em đã giữ một biến thể của ba mô hình tham chiếu được cung cấp trong hướng dẫn Tensorflow (Dense, Conv và LSTM).Em đã xây dựng hai mô hình khác (mylstm_1, mylstm_2) hoạt động tốt hơn so với các mô hình tham chiếu.

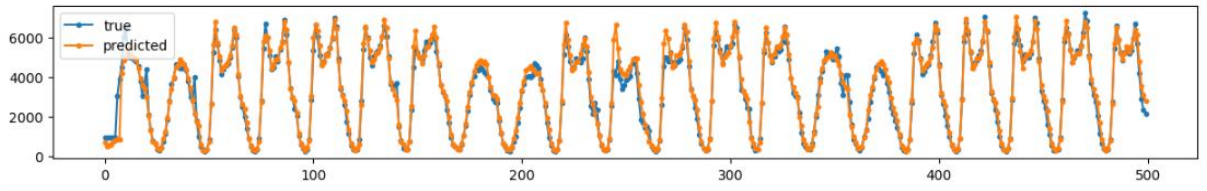
So sánh hiệu suất của các mô hình



Hình 20 So sánh hiệu suất của các mô hình

Mô hình tốt nhất của em sử dụng các LSTM hai chiều với hai lớp chuyển tiếp và lùi tùy chỉnh, và hai lớp dày đặc với 512 đơn vị mỗi lớp và một lớp đầu ra dày đặc với một đơn vị.Mô hình đã thực hiện phương sai thấp nhất và được duy trì cho nhiều kỷ nguyên hơn trong quá trình đào tạo .

Các dự đoán trong (Hình) cho thấy mô hình đã nắm bắt được tất cả các mẫu quan trọng với các lần bỏ lỡ nhỏ của một số dị thường.



Hình 21 Mô hình nắm bắt các điểm dị thường

5.2. Hạn chế

Độ chính xác của dự đoán: Dự đoán lưu lượng giao thông trên bộ dữ liệu trực tuyến có thể Không hoàn toàn chính xác một cách hoàn hảo. Các biến thể và biến động

của giao thông thường xuyên xảy ra, và dữ liệu trực tuyến có thể không đầy đủ hoặc không chính xác. Do đó, việc dự đoán chính xác lưu lượng giao thông trên một khoảng thời gian cụ thể có thể là một thách thức.

Độ trễ: Khi phân tích và xử lý dữ liệu trực tuyến, việc thu thập và xử lý dữ liệu có thể gặp độ trễ. Thời gian mất đi từ khi dữ liệu được thu thập cho đến khi nó được xử lý và đưa ra dự đoán có thể gây trễ đáng kể. Điều này có thể làm giảm hiệu quả và tính ứng dụng của hệ thống trong một số tình huống cần đưa ra phản ứng nhanh chóng.

Sự biến đổi không đồng đều: Lưu lượng giao thông có thể thay đổi theo thời gian và địa điểm. Trong một số khu vực, sự biến đổi có thể rất không đồng đều và khó dự đoán. Các yếu tố như sự kiện đặc biệt, công trình xây dựng, thời tiết, và các yếu tố khác có thể ảnh hưởng đến lưu lượng giao thông một cách không thường xuyên và khó khăn cho việc xử lý và dự đoán.

Quản lý và xử lý dữ liệu lớn: Dữ liệu giao thông trực tuyến thường rất lớn và phức tạp. Quá trình phân tích và xử lý dữ liệu lớn yêu cầu tài nguyên mạnh mẽ và hệ thống tính toán hiệu quả. Việc thu thập, lưu trữ và xử lý dữ liệu lớn có thể tốn kém và đòi hỏi kiến thức chuyên môn về cơ sở dữ liệu, hệ thống phân tán, và các công nghệ liên quan khác.

Bảo mật và riêng tư: Dữ liệu giao thông thường chứa thông tin cá nhân và có tính nhạy cảm. Việc bảo vệ dữ liệu và đảm bảo tính riêng tư là một thách thức quan trọng. Cần có các biện pháp bảo mật mạnh mẽ để đảm bảo rằng dữ liệu không bị lộ và được xử lý một cách an toàn và đúng đắn.

5.3. Hướng phát triển

Có thể áp dụng được mô hình dựa trên tập dựa liệu và các tuyến đường giao thông ở Việt Nam. Tích hợp hệ thống quản lý giao thông thông minh, đề tài có thể được mở rộng để tích hợp với hệ thống quản lý giao thông thông minh (ITS) để tạo ra giải pháp toàn diện. ITS cung cấp các công nghệ và giải pháp quản lý giao thông thông minh, như đèn giao thông thông minh, hệ thống điều phối giao thông và cơ chế thu phí điện tử. Tích hợp với ITS có thể cung cấp lợi ích lớn cho việc phân tích, xử lý và dự đoán lưu

lượng giao thông. Phân tích và dự đoán thời gian hành trình, ngoài việc dự đoán lưu lượng giao thông, cũng có thể mở rộng để phân tích và dự đoán thời gian hành trình của các phương tiện. Điều này có thể giúp người dùng và hệ thống đưa ra các quyết định thông minh về lộ trình và thời gian đi lại.

5.4. Lời kết

Như vậy bài cáo này là đại diện cho quá trình tìm hiểu và nghiên cứu của em về những kiến thức chuyên ngành và những kiến thúc ngoài chuyên ngành trong thời gian thực tập chuyên môn. Đồng thời đây cũng một bước xác định và định hướng nghiên cứu trong tương lai khi kết thúc bài báo cáo này ta lại rút ra những khuyết điểm thứ mà có thể khắc phục và cải thiện trong bài báo cáo tới.

---------- Xin Cảm Ơn Quý Thầy cô đã xem xét. ----------

TÀI LIỆU THAM KHẢO

1. A Research of Traffic Prediction using Deep Learning Techniques. International Journal of Innovative Technology and Exploring Engineering, 2019. 8(9S2): p. 725-728.
2. Apurv Chandel, S.S., Badavath Uday Kiran, Prabhas Prasad, Nidhi Lal, An Accurate Estimation of Interstate Traffic of Metro City Using Linear Regression Model of Machine Learning.
3. Data set. Available from: https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume#.
4. Ahmad, U.K. Metro Interstate Traffic Volume Time-series Forecasting Using Recurrent for Neural Networks (RNNs). Available from: https://medium.com/@umaimakhurshidahmad/metro-interstate-traffic-volume-time-series-forecasting-using-recurrent-for-neural-networks-rnns-a73732276d1a.
5. Garlan, E.; Available from: https://www.kaggle.com/code/ramyahr/metro-interstate-traffic-volume/notebook.
6. Rohit Singh, A., Mr. Sibi Amaran, Dr. K Sree Kumar, Analysis of traffic flow in different weather conditions. april, 2021.
7. Jiang, R., et al., DL-Traff: Survey and Benchmark of Deep Learning Models for Urban Traffic Prediction, in Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021. p. 4515-4525.