BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**🙞 🕮 🙜**

Logo

Description automatically generated

**NIÊN LUẬN NGÀNH**

**KHOA HỌC MÁY TÍNH**

**Đề tài**

**DỰ BÁO THỜI TIẾT VỚI THUẬT TOÁN RNN**

**Sinh viên thực hiện: Nguyễn Việt Hào**

**MSSV: B1812338**

**Khóa: 44**

**Cần Thơ, 05/2022**

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**🙞 🕮 🙜**



**NIÊN LUẬN NGÀNH**

**KHOA HỌC MÁY TÍNH**

**Đề tài**

**DỰ BÁO THỜI TIẾT VỚI THUẬT TOÁN RNN**

**Giáo viên hướng dẫn: Sinh viên thực hiện:**

**Th.S.Phạm Nguyên Hoàng Nguyễn Việt Hào**

**MSSV: B1812338**

**Khóa: 44**

**Cần Thơ, 05/2022**

# NHẬN XÉT CỦA GIẢNG VIÊN

**----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

# LỜI CẢM ƠN

Đầu tiên em xin gửi lời cảm ơn chân thành đến thầy Phạm Nguyên Hoàng đã hướng dẫn em hoàn thành học phần Niên luận ngành này. Học phần đã giúp em mở rộng thêm nhiều kiến thức mới trong quá trình làm và vận dụng tốt những kiến thức đó. Cảm ơn thầy đã tận tình giảng dạy hướng dẫn cũng như tạo điều kiện thuận lợi nhất cho em trong suốt quá trình làm đề tài. Do hạn chế về thời gian và trình độ, bản báo cáo này chắc chắn sẽ có những thiếu sót. Mong thầy góp ý cho bài báo cáo của em để có thể làm hành trang quý giá cho tương lai. Em xin chân thành cảm ơn !

Cần Thơ, ngày 12 tháng 5 năm 2022

# MỤC LỤC

[NHẬN XÉT CỦA GIẢNG VIÊN 1](#_Toc103267098)

[LỜI CẢM ƠN 2](#_Toc103267099)

[MỤC LỤC 3](#_Toc103267100)

[DANH MỤC HÌNH 5](#_Toc103267101)

[DANH MỤC BẢNG 7](#_Toc103267102)

[ABSTRACT 8](#_Toc103267103)

[TÓM TẮT 9](#_Toc103267104)

[PHẦN GIỚI THIỆU 10](#_Toc103267105)

[1. Đặt vấn đề 10](#_Toc103267106)

[2. Lịch sử giải quyết vấn đề. 10](#_Toc103267107)

[2.1 Daily Temperature Prediction Using Recurrent Neural Networks and Long-Short Term Memory 10](#_Toc103267108)

[2.2 Machine Learning – Thử làm Nhà Thiên Văn Dự báo thời tiết. 12](#_Toc103267109)

[3. Mục tiêu đề tài 14](#_Toc103267110)

[4. Đối tượng và phạm vi nghiên cứu 14](#_Toc103267111)

[5. Phương pháp nghiên cứu 15](#_Toc103267112)

[6. Kết quả đạt được 15](#_Toc103267113)

[7. Bố cục niên luận 15](#_Toc103267114)

[PHẦN NỘI DUNG 17](#_Toc103267115)

[Chương I. MÔ TẢ BÀI TOÁN 17](#_Toc103267116)

[1. Mô tả chi tiết bài toán 17](#_Toc103267117)

[2. Các vấn đề và giải pháp liên quan 17](#_Toc103267118)

[2.1 Mạng nơ ron nhân tạo (neural network) 17](#_Toc103267119)

[2.2 Mạng nơ ron truyền thẳng (feed forward neural network) 17](#_Toc103267120)

[2.3 Mạng nơ ron hồi quy (RNN) 18](#_Toc103267121)

[2.3.1 Ví dụ về RNN mô hình hóa trong xử lý ngôn ngữ 19](#_Toc103267122)

[2.3.2 Quá trình lưu thông tin trong RNN 19](#_Toc103267123)

[2.4 Thuật toán lan truyền ngược (BBTT – Backpropagation Through Time) 21](#_Toc103267124)

[2.5 Ưu điểm và nhược điểm của RNN 21](#_Toc103267125)

[2.5.1 Ưu điểm 21](#_Toc103267126)

[2.5.2 Nhược điểm 21](#_Toc103267127)

[2.6 Vấn đề và giải pháp liên quan đến bài toán 22](#_Toc103267128)

[2.6.1 Mô hình tuần tự Sequential 22](#_Toc103267129)

[2.6.2 Long-short term memory – LSTM 22](#_Toc103267130)

[3. Các công cụ sử dụng và một số thư viện. 23](#_Toc103267131)

[3.1 Google Colaboratory 23](#_Toc103267132)

[3.2 Thư viện Keras 24](#_Toc103267133)

[3.3 Thư viện Pandas 25](#_Toc103267134)

[3.4 Thư viện Numpy 25](#_Toc103267135)

[3.5 Thư viện Sklearn 26](#_Toc103267136)

[3.6 Thư viện Matplotlib 26](#_Toc103267137)

[Chương II. THIẾT KẾ VÀ CÀI ĐẶT 28](#_Toc103267138)

[1. Thiết kế hệ thống 28](#_Toc103267139)

[1.1 Đọc tập dữ liệu 28](#_Toc103267140)

[1.2 Tiền xử lý dữ liệu 29](#_Toc103267141)

[1.3 Xây dựng mô hình (thiết kế mạng nơ ron) 29](#_Toc103267142)

[2. Cài đặt hệ thống 29](#_Toc103267143)

[2.1 Đọc tập dữ liệu 29](#_Toc103267144)

[2.1.1 Thông tin cơ bản của tập dữ liệu. 30](#_Toc103267145)

[2.1.2 Thống kê cơ bản về dữ liệu. 30](#_Toc103267146)

[2.1.3 Kiếm tra các giá trị rỗng 31](#_Toc103267147)

[2.2 Tiền xử lý dữ liệu 31](#_Toc103267148)

[2.2.1 Thay đổi chỉ mục cho tập dữ liệu. 31](#_Toc103267149)

[2.2.2 Xử lý các giá trị bị thiếu (missing values) 32](#_Toc103267150)

[2.2.3 Xử lý bằng Label Encoder 32](#_Toc103267151)

[2.2.4 Xóa các cột không cần thiết 33](#_Toc103267152)

[2.2.5 Hiển thị dữ liệu. 34](#_Toc103267153)

[2.2.5.1 Biểu đồ Ta (nhiệt độ trung bình-độ C) 34](#_Toc103267154)

[2.2.5.2 Biểu đồ Tx (biểu đồ nhiệt độ cao nhất-độ C) 34](#_Toc103267155)

[2.2.6 Scale dữ liệu 34](#_Toc103267156)

[2.3 Xây dựng mô hình (model). 35](#_Toc103267157)

[2.3.1 Thiết kế mạng 35](#_Toc103267158)

[2.3.2 Tạo tập dữ liệu huấn luyện 35](#_Toc103267159)

[2.3.3 Tập dữ liệu kiểm tra. 36](#_Toc103267160)

[2.3.4 Biên dịch mô hình và huấn luyện mô hình 36](#_Toc103267161)

[2.4 Dự đoán thời tiết Cần Thơ với tập dữ liệu kiểm tra dựa trên mô hình đã xây dựng 37](#_Toc103267162)

[Chương III. KẾT QUẢ THỰC NGHIỆM 41](#_Toc103267163)

[1. Kiểm thử 41](#_Toc103267164)

[1.1 Trường hợp kiểm thử 1: Scale bằng hàm RobustScaler vs Min-Max Scaler (Cần Thơ) 41](#_Toc103267165)

[1.2 Trường hợp kiểm thử 2: Tăng số nơron ở tầng LSTM 41](#_Toc103267166)

[1.3 Trường hợp kiểm thử 3: Thử mới Station Sapa và Đà Nẵng 42](#_Toc103267167)

[1.3.1 Biểu diễn dự đoán Ta (average temperature): 42](#_Toc103267168)

[1.3.2 Biễu diễn dự đoán rH (relative humidity) 43](#_Toc103267169)

[1.3.3 Biễu diễn dự đoán Sh (hours of sunshine) 44](#_Toc103267170)

[1.4 Trường hợp kiểm thử 4: với timestep = 24 (Cần Thơ) 45](#_Toc103267171)

[2. Đánh giá 46](#_Toc103267172)

[PHẦN KẾT LUẬN 47](#_Toc103267173)

[1. Kết quả đạt được 47](#_Toc103267174)

[1.1 Kỹ năng 47](#_Toc103267175)

[1.2 Chương trình 47](#_Toc103267176)

[2. Hướng phát triển 47](#_Toc103267177)

[TÀI LIỆU THAM KHẢO 48](#_Toc103267178)

# DANH MỤC HÌNH

[Hình 1: Mô hình LSTM 12](#_Toc103266249)

[Hình 2: Dữ liệu tới tiết của Hà Nội 13](#_Toc103266250)

[Hình 3: Ví dụ mô tả tập dữ liệu huấn luyện 14](#_Toc103266251)

[Hình 4: Kết quả dự đoán khá tốt với 15 ngày 15](#_Toc103266252)

[Hình 5: Mạng RNN (bên trái), mạng FFNN (bên phải) 19](#_Toc103266253)

[Hình 6: Mô hình mạng RNN 19](#_Toc103266254)

[Hình 7: Ví dụ về xử lý ngôn ngữ bằng RNN 20](#_Toc103266255)

[Hình 8: Quá trình lưu thông tin trong cell 21](#_Toc103266256)

[Hình 9: Công thức tính h(t) trong xử lý ngôn ngữ 21](#_Toc103266257)

[Hình 10: Công thức tính y(t) trong xử lý ngôn ngữ 21](#_Toc103266258)

[Hình 11: Đạo hàm của L với W 22](#_Toc103266259)

[Hình 12: Mô hình tuần tự Sequential 23](#_Toc103266260)

[Hình 13: Recurrent Neural Network 23](#_Toc103266261)

[Hình 14: Long short term memory 24](#_Toc103266262)

[Hình 15: Giao diện của Google Colab 25](#_Toc103266263)

[Hình 16: Một Pandas dataframe 26](#_Toc103266264)

[Hình 17: Hàm reshape trong Numpy 27](#_Toc103266265)

[Hình 18: Các dạng biểu đồ trong Matplotlib0 28](#_Toc103266266)

[Hình 19: Tập dữ liệu thời tiết ban đầu của Việt Nam 29](#_Toc103266267)

[Hình 20: Dữ liệu tệp dulieuthoitiet.csv 30](#_Toc103266268)

[Hình 21: Thông tin của tập dữ liệu 31](#_Toc103266269)

[Hình 22: Thống kê tập dữ liệu ban đầu 32](#_Toc103266270)

[Hình 23: Hàm strptime() trong datetime 33](#_Toc103266271)

[Hình 24: Thêm cột mã hóa của cột station 33](#_Toc103266272)

[Hình 25: Thống kê số tháng của station 34](#_Toc103266273)

[Hình 26: Tập dữ liệu sau khi xóa các cột không cần thiết 34](#_Toc103266274)

[Hình 27: Biểu đồ nhiệt độ trung bình 35](#_Toc103266275)

[Hình 28: Biểu đồ nhiệt độ cao nhất 35](#_Toc103266276)

[Hình 29: Dữ liệu sau khi đã scale 36](#_Toc103266277)

[Hình 30: Mô hình Stacked LSTM 36](#_Toc103266278)

[Hình 31: Dữ liệu minh họa x\_train và y\_train 37](#_Toc103266279)

[Hình 32: Kết quả của quá trình huấn luyện 38](#_Toc103266280)

[Hình 33: Loss của model 38](#_Toc103266281)

[Hình 34: Biểu diễn giá trị Nhiệt độ Trung bình huấn luyện, giá trị dự đoán và thực tế 39](#_Toc103266282)

[Hình 35:Biểu diễn giá trị Nhiệt độ Trung bình dự đoán và thực tế 39](#_Toc103266283)

[Hình 36: Biểu diễn giá trị Độ ẩm tương đối huấn luyện, giá trị dự đoán và thực tế 40](#_Toc103266284)

[Hình 37: Biểu diễn giá trị Độ ẩm tương đối dự đoán và thực tế 40](#_Toc103266285)

[Hình 38: Biểu diễn giá trị Độ ẩm tuyệt đối huấn luyện, giá trị dự đoán và thực tế 41](#_Toc103266286)

[Hình 39: Biểu diễn giá trị Độ ẩm tuyệt đối dự đoán và thực tế 41](#_Toc103266287)

[Hình 40: Biểu diễn dự đoán Ta của Sapa 43](#_Toc103266288)

[Hình 41: Biểu diễn dự đoán Ta của Đà Nẵng 43](#_Toc103266289)

[Hình 42: Biểu diễn dự đoán rH của Sapa 44](#_Toc103266290)

[Hình 43: Biểu diễn dự đoán rH của Đà Nẵng 44](#_Toc103266291)

[Hình 44: Biểu diễn dự đoán Sh của Sapa 45](#_Toc103266292)

[Hình 45: Biểu diễn dự đoán Sh của Đà Nẵng 45](#_Toc103266293)

# DANH MỤC BẢNG

[Bảng 1: So sánh loss và accuracy của SGD và Adam 12](#_Toc103266233)

[Bảng 2: Bảng so sánh loss và accuracy khi thay đổi lượng dữ liệu được huấn luyện 12](#_Toc103266234)

[Bảng 3: Bảng so sánh loss và accuracy khi thay đổi cách chia dữ liệu huấn luyện và kiểm tra 13](#_Toc103266235)

[Bảng 4: Thống kê giá trị Null 32](#_Toc103266236)

[Bảng 5: RMSE và MAE của mô hình dự đoán cho Cần Thơ 39](#_Toc103266237)

# ABSTRACT

Weather forecasting is one of the most important aspects of modern life. Weather forecasting will let us know the weather only in the future, so that people can prepare for the future, for travels, ... or even organizations that emigrate from disaster will exist, thanks to the advancements of current science and technology.

Because weather is a time series, we may design a weather prediction model for the future utilize data acquired in the past with the advancement of in the contemporary world. As a result, in this topic, I'll use recurrent neural networks in general, and LSTM (long short term memory), which is a type of RNN in particular, as well as the Stacked LSTM model to predict weather parameters like temperature, humidity, rainfall, and so on, using accessible weather data.

Research and apply data preprocessing techniques such as: to encode numeric data, research and implement data preparation techniques such as Label Encoder… The neural network model will next be generated using the Keras library's Sequential model. This model is quite powerful with deep learning, and it is well suited for time series problems, such as weather forecasting.

The project's outcomes include a successful neural network model for time series forecasting in this topic, weather indicators, with high model evaluation indexes (rmse, mae) unexpected, but some data fields may not yield good results because there are weather indicators with a lot of variation between the past and the future. As a result, the built prediction model will be unable to predict these data fields accurately.

# TÓM TẮT

Dự báo thời tiết là một trong những điều cần thiết trong thế giới hiện đại ngày nay. Với sự phát triển của khoa học công nghệ hiện đại, dự báo thời tiết sẽ giúp cho chúng ta biết được các chỉ số thời tiết trong tương lai, nhờ vào đó con người có thể lập kế hoạch cho tương lai, cho các chuyến đi chơi,… hay thậm chí là tổ chức di tán người dân từ những dự báo thiên tai sẽ xảy ra.

Thời tiết có tính chất chuỗi thời gian, nên với sự phát triển của hiện tại ta có thể xây dụng một mô hình dự đoán thời tiết cho tương lai bằng các số liệu thu thập được từ quá khứ. Vì thế trong đề tài này em sẽ sử dụng mạng nơ ron hồi quy nói chung và LSTM (long short term memory) là một dạng của RNN nói riêng, cùng với áp dụng mô hình Stacked LSTM để dự đoán cho các thông số thời tiết như nhiệt độ, độ ẩm, lượng mưa, … bằng những số liệu thời tiết có sẵn.

Nghiên cứu và áp dụng các kỹ thuật tiền xử lý dữ liệu như Label Encoder để mã hóa dữ liệu về kiểu số, ... Mô hình mạng nơ ron sau đó sẽ được xây dựng nhờ vào mô hình tuần tự Sequential của thư viện Keras rất mạnh mẽ với mô hình học sâu, mô hình này được rất thích hợp cho bài toán chuỗi thời gian và ở đề tài sẽ là bài toán dự báo thời tiết.

Kết quả của đề tài đã thu được một mô hình mạng noron hiệu quả cho dự báo chuỗi thời gian ở đề tài này là dự báo các chỉ số thời tiết, đã đạt được các chỉ số đánh giá mô hình (rmse, mae) cao ngoài mong đợi, nhưng đối với một số trường dữ liệu có thể sẽ cho kết quả không cao vì tồn tại các chỉ số thời tiết có sự biến thiên rất cao giữa quá khứ và tương lai dẫn đến mô hình dự đoán đã xây dựng sẽ không thể dự đoán tốt cho các trường dữ liệu này.

# PHẦN GIỚI THIỆU

## Đặt vấn đề

Hiện nay, với sự phát triển của kỹ thuật khoa học, công nghệ tiên tiến thì việc dự báo thời tiết cho mọi địa điểm trên địa cầu đã không còn quá xa lạ. Con người cũng đã nỗ lực trong việc dự báo thời tiết không chính thức trong nhiều thiên niên kỷ trước và mới bắt đầu chính thức từ thế kỷ mười chín. Nhờ sự phát triển của khoa học công nghệ, các thiết bị hiện đại ngày nay thì các dữ liệu thu thập một cách dễ dàng hơn từ các trạm thời tiết, vệ tinh được sử dụng để phân tích và dự đoán các trạng thái, chỉ số thời tiết. Con người dựa trên các quan sát trong quá khứ về thời tiết để có thể dự đoán được thời tiết trong tương lai. Và việc áp dụng máy học đã được sử dụng rộng rãi và có thể hoạt động tốt trong lĩnh vực này.

## Lịch sử giải quyết vấn đề.

* + - 1. Daily Temperature Prediction Using Recurrent Neural Networks and Long-Short Term Memory

Đây là bài viết viết về chủ đề dự báo nhiệt độ với RNN và LSTM, được viết bởi Ike Sri Rahayu, Esmeralda C Djamal, Ridwan Ilyas của khoa tin học, đại học Jenderal Achmad Yani, Indonesia. Dữ liệu của bài báo được lấy từ Cơ quan Địa Vật lý và khí tượng (BMKG) ở Bandung từ năm 2000 đến năm 2019. Bài báo này xây dựng một mô hình dự đoán nhiệt độ hằng ngày trong ba ngày tới bằng 5 lớp: lạnh, mát, bình thường, ấm và nóng bằng cách sử dụng mạng Nơ ron hồi quy và bộ nhớ ngắn hạn dài hạn (LSTM). Nghiên cứu được tiến hành do nhiệt độ là một trong những chỉ số thời tiết quan trọng, nhiệt độ thay đổi có thể ảnh hưởng đến cơ thể con người, đặc biệt là nếu sử dụng quần áo không phù hợp, điều này khiến con người phải thay đổi chất liệu quần áo để phù hợp hơn. Bài báo nghiên cứu xây dụng một mô hình có thể dự đoán nhiệt độ hằng ngày bằng cách sử dụng RNN. Với dữ liệu đầu vào là 4 chỉ số thởi tiết là nhiệt độ, độ ẩm, lượng mưa và tốc độ gió của 20 năm. Dữ liệu trước khi được huấn luyện sẽ được tiền xử lý (tính trung bình cho các giá trị bị thiếu, scale dữ liệu, xử lý các giá trị NaN…) và sau đó huấn luyện bằng mô hình RNN và LSTM dự đoán nhiệt độ cho ba ngày tiếp theo. Sử dụng kỹ thuật phân đoạn, tức là nhóm theo mỗi ba ngày nhóm thành 1 bộ (gồm đầy đủ chỉ số thời tiết), kết quả sẽ tạo ra 2406 bộ dữ liệu, mỗi bộ là một tháng được sắp xếp theo thời gian. Đầu ra của mô hình là hàm Softmax để phân loại cho nhiệt độ đầu ra. Song đó thì RNN là một bộ nhớ ngắn hạn nên sẽ có hạn chế trong xử lý nhiều dữ liệu nên LSTM là một mô hình có thể khắc phục được những hạn chế đó, có khả năng học các phụ thuộc dài hạn và các quá trình tính toán sẽ phức tạp hơn.

A picture containing text, clock, device, gauge

Description automatically generated

Hình 1: Mô hình LSTM

Bài báo này xây dựng mô hình dự đoán nhiệt độ mỗi ba ngày với dữ liệu của 20 năm, 14 năm và 4 năm. Mô hình sử dụng 2 bộ tối ưu hóa (optimizer) là Stochastic Gradient Descent (SGD)và Adaptive Moment Estimation (Adam). Trong đó mô hình Adam qua thử nghiệm sẽ cho kết quả tốt hơn mô hình SGD. Sau đó các thử nghiệm sẽ được thử nghiệm với bộ optimizer Adam

Table

Description automatically generated

Bảng 1: So sánh loss và accuracy của SGD và Adam

Kiểm thử với dữ liệu huấn luyện là 20 năm, 12 năm và 4 năm cho ra được độ chính xác cao nhất là 90.92% trên tập dữ liệu huấn luyện và 80,36% trên tập dữ liệu kiểm tra. Kết quả cho thấy được việc giảm lượng tập dữ liệu huấn luyện đi sẽ làm ảnh hưởng đến độ chính xác của mô hình (loss tăng lên accuracy giảm xuống)

Table

Description automatically generated

Bảng 2: Bảng so sánh loss và accuracy khi thay đổi lượng dữ liệu được huấn luyện

Và cách chia lượng tập dữ liệu huấn luyện và kiểm tra cũng sẽ ảnh hưởng đến độ chính xác của mô hình:

Table

Description automatically generated

Bảng 3: Bảng so sánh loss và accuracy khi thay đổi cách chia dữ liệu huấn luyện và kiểm tra

Qua các thử nghiệm, có thể kết luận rằng mô hình tối ưu hóa, lượng dữ liệu và cách chia dữ liệu có thể ảnh hưởng đến kết quả thu được.

* + - 1. Machine Learning – Thử làm Nhà Thiên Văn Dự báo thời tiết.

Bài viết này được tác giả Lavender nói về việc sử dụng các thông tin thời tiết trong quá khứ để dự đoán cho thời tiết hôm sau. Dữ liệu của bài viết này là dữ liệu của thời tiết Hà Nội được lấy từ <https://www.meteoblue.com/> với dữ liệu miễn phí thì chỉ có thể download được dữ liệu thời tiết của 15 ngày gần nhất và có thể trả phí để download dữ liệu trong 30 năm gần nhất.

Table

Description automatically generated

Hình 2: Dữ liệu tới tiết của Hà Nội

Tác giả sử dụng thư viện Tensorflow để hỗ trợ việc huấn luyện vì đã được tích hợp nhiều thuật toán khác nhau giúp tối ưu được thời gian xây dựng hệ thống học sâu (deep learning). Cùng với các thư viện Pandas, Numpy để tiền xử lý dữ liệu, cũng như đưa ra cái nhìn tổng quan về dữ liệu bằng cách vẽ các đồ thị bằng thư viện Matplotlib. Vì dữ liệu thời tiết là dữ liệu chuỗi thời gian nên các dữ liệu ở tương lai sẽ phụ thuộc vào các dữ liệu trong quá khứ, mang trong mình tính chất chuỗi thời gian, nên tác giả sử dụng các dữ liệu trong thời gian trước cụ thể là 1 giờ trước làm X (tập dữ liệu huấn luyện) và dữ dữ liệu trong tương lai 1 giờ liền kề làm Y

Table, calendar

Description automatically generated

Hình 3: Ví dụ mô tả tập dữ liệu huấn luyện

Do đó, tác giả sẽ sử dụng giá trị thuộc tính Temperature (nhiệt độ) độ C từ 15-01-2018 00:00:00 đến 29-01-2018 00:00:00 làm tập X\_train với y\_train sẽ là giá trị từ 15-01-2018 01:00:00 và hai ngày cuối cùng làm tập X\_test (tương đương với 48 giờ) – mô hình xây dựng sẽ dự đoán chỉ số thời tiết cho 48 giờ này. Biến đổi tập dữ liệu về dạng (samples, time\_steps, feature) để phù hợp với đầu vào của mô hình mạng nơron ở đây sẽ là (14, 24, 1) tương đương 14 ngày, 1 ngày 24 giờ, và 1 giờ ứng với 1 thuộc tính thời tiết làm dữ liệu đầu vào.

Tác giả lựa chọn thuật toán Recurrent Neural Network (RNN) hay mạng nơ ron hồi quy do tính chất sử dụng các dữ liệu trong quá khứ để dự đoán tương lai, có khả năng nhớ được các thông tin trước đó (do đầu ra của dữ liệu sẽ phụ thuộc vào các tính toán trước đó). Tác giả sử dụng nơ ron mô hình RNN truyền thống với 100 nơ ron đầu ra của tầng này, hàm activation là relu, phương pháp huấn luyện là Adam và hàm mất mát là MSE. Sau đó tác giả huấn luyện với 1000 epoch và đã thu được kết quả khá tốt chỉ với 15 ngày làm dữ liệu đầu vào và có thể sẽ hiệu quả hơn với dữ liệu đầu vào nhiều hơn thay vì 15 ngày.

Chart, scatter chart

Description automatically generated

Hình 4: Kết quả dự đoán khá tốt với 15 ngày

## Mục tiêu đề tài

Dự báo thời tiết, trong đó thời tiết có tính chất chuỗi thời gian. Cùng với sự phát triển của các ngành như khoa học dữ liệu, các công nghệ hiện đại để thu tập dữ liệu thì chúng ta có thể xây dựng một mô hình dự đoán trạng thái thời tiết như nhiệt độ, độ ẩm… từ các dữ liệu ta thập thập được. Mục tiêu của đề tài sẽ sử dụng dữ liệu thời tiết đã có sẵn (nhiệt độ, độ ẩm,..), xây dựng một mô hình mạng nơ ron hồi quy, dự đoán và đánh giá mô hình.

## Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài :

* Mạng nơ ron hồi quy (RNN).
* Mô hình Long short term memory của mạng Mạng nơ ron hồi quy.
* Các kỹ thuật tiền xử lý dữ liệu.
* Thư việc Keras.
* Dữ liệu chuỗi thời gian.
* Vẽ biểu đồ bằng thư viện Matplotlib

Phạm vi nghiên cứu:

* Tiền xử lý dữ liệu (xử lý dữ liệu bị thiếu)
* Nghiên cứu áp dụng thư việc Keras
* Xây dựng mô hình dự báo thời tiết với LSTM
* Đánh giá mô hình bằng các chỉ số như rmse, mae

## Phương pháp nghiên cứu

* Xác định đề tài: Để tránh mất nhiều thời gian đọc các đề tài nội dụng không liên quan hoặc không phục cho mục đích đề tài của mình thì việc đầu tiên cần làm là xác định chỉ rõ đề tài cần nghiên cứu. Để có được hướng đi cũng như cái nhìn tổng quan cho bài báo cáo.
* Xác định các từ khoá: Xác định từ khoá liên quan đến đến đề tài để có thể thu hẹp phạm vi, dễ dàng tìm kiếm xây dựng chủ đề tránh mất nhiều thời gian.
  + RNN, Time series, Sequential, Long short term memory
* Tìm các đề tài tượng tự và chọn lọc: Chọn lọc lại các đề tài, tài liệu tìm được, đặc biệt là đọc phần tóm tắt xem có giúp ta thu được giá trị phù hợp liên quan không?
* Đề ra các hướng làm: Xây dụng mô hình dự báo thời thiết tiết theo cách tiếp cận với thuật toán RNN
* Tìm hiểu về RNN, LSTM, Timeseries
* Đọc tập dữ liệu, hiển thị để có cái nhìn tổng quan về dữ liệu.
* Tiền xử lý dữ liệu
* Xây dựng tập dữ liệu huấn luyện và tập dữ liệu kiểm thử.
* Xây dựng mô hình LSTM
* Đánh giá mô hình (rmse, vẽ biểu đồ)
* Nhận góp ý từ người khác để có nhiều góc nhìn hơn để hoàn thiện chỉnh sửa.
* Chỉnh sửa lại từ các góp ý
* Viết báo cáo

## Kết quả đạt được

ây dựng được một mô hình dự báo thời tiết từ tập dữ liệu thời tiết có sẵn bằng mô hình mạng nơ ron hồi quy long short term memory – một mô hình nâng cao của mạng nơ ron hồi quy. Đánh giá mô hình đã được xây dựng.

## Bố cục niên luận

Bố cục gồm 3 phần với nội dung như sau:

**PHẦN GIỚI THIỆU**

Giới thiệu tổng quát về đề tài

**PHẦN NỘI DUNG**

**Chương 1:** Mô tả bài toán

**Chương 2:** Thiết kế, cài đặt giải thuật.

**Chương 3:** Kết quả thực nghiệm

**PHẦN KẾT LUẬN**

          Kết quả đạt được và hướng phát triển

**TÀI LIỆU THAM KHẢO**

# PHẦN NỘI DUNG

# MÔ TẢ BÀI TOÁN

## Mô tả chi tiết bài toán

Chuỗi thời gian là một chuỗi các phép đo được thực hiện theo thời gian, thường thu được ở các khoảng cách đều nhau, có thể là hàng ngày, hàng tháng, hàng quý hoặc hàng năm. Nói cách khác, chuỗi thời gian là một chuỗi các điểm dữ liệu được ghi lại tại các thời điểm cụ thể. Các điểm dữ liệu này thường bao gồm các phép đo liên tiếp được thực hiện trong một thời gian và được sử dụng để theo dõi sự thay đổi theo thời gian như lưu lượng dòng chảy hằng năm, dữ liệu dân số hằng năm, lãi suất hằng tuần,… Ở bài toán này cũng sẽ giải quyết bài toán dữ liệu chuỗi thời gian, tức sẽ dự đoán cho chỉ số thời tiết cho một số tỉnh thành của Việt Nam trong tập dữ liệu 65 tỉnh thành được thu thập theo hằng tháng (ứng với mỗi dòng dữ liệu) trong 50 năm (từ 1960 đến 2010) bằng cách sử dụng mô hình mạng noron hồi quy kết hợp ứng dụng Stacked LSTM để dự đoán cho 7 chỉ số thời tiết (nhiệt độ trung bình, nhiệt độ cao nhất, nhiệt độ thấp nhất, lượng mưa, độ ẩm tương đối, thời lượng nắng, độ ẩm tuyệt đối) để dự đoán cho một số tỉnh thành (Cần Thơ, Đà Nẵng, Sapa).

## Các vấn đề và giải pháp liên quan

* + - 1. Mạng nơ ron nhân tạo (neural network)

Mạng nơ ron nhân tạo (neural network) là một mô hình toán học hay mô hình tính toán được xây dựng dựa trên mạng nơ ron sinh học, là một hệ thống tính toán lấy cảm hứng từ sự hoạt động của các nơ-ron trong hệ thần kinh của con người.

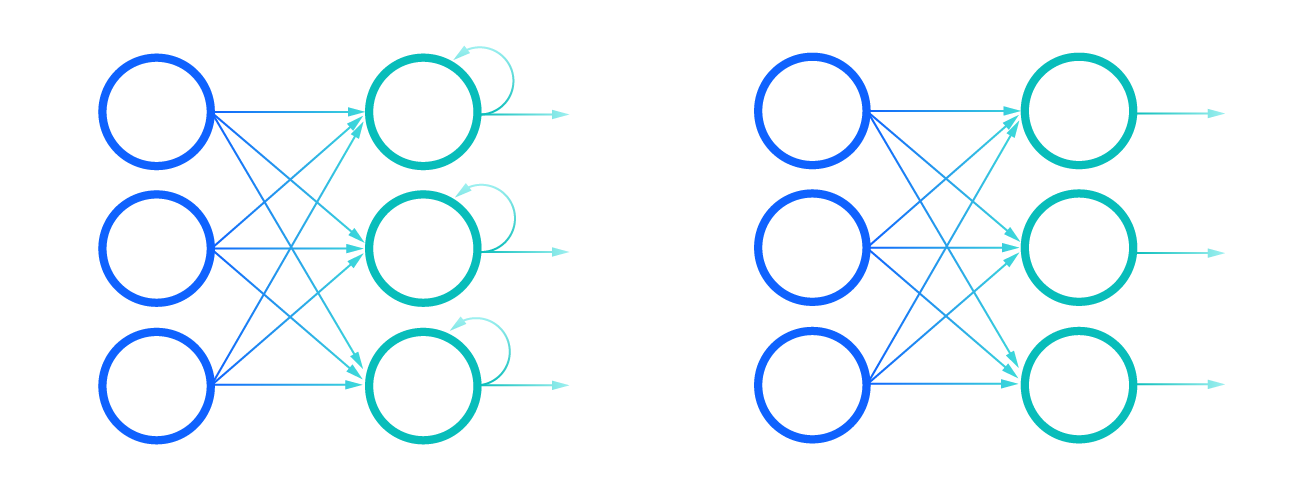
* + - 1. Mạng nơ ron truyền thẳng (feed forward neural network)

Trước hết nói về mạng nơron truyền thẳng (feed forward neural network), mạng nơ ron sẽ gồm 3 thành phần chính là: tầng đầu vào (input layer), tầng ẩn (hidden layer), tầng đầu ra (output layer)

* Thông tin chỉ chảy theo một hướng: tư tầng đầu vào, qua các lớp ẩn, đến lớp đầu ra
* Các mạng nơ ron này chuyển tiếp nguồn cấp dữ liệu mà không giữ lại bộ nhớ về các đầu vào mà chúng đã xử lý.
* Đầu vào và đầu ra của mạng nơ ron truyền thẳng độc lập với nhau.

Mạng nơ ron truyền thằng (FFNN) không phù hợp với những bài toán dạng chuỗi (mô tả nội dung, hoàn thành câu, chuỗi thời gian…) vì các dự đoán tiếp theo sẽ phụ thuộc và tính toán dựa trên các dữ liệu trước nó (như vị trí trong câu, từ nào ở trước nó…)

Do đó, mạng nơ ron hồi quy (RNN) đã được ra đời để xử lý các dạng dữ liệu mà FFNN sẽ làm không tốt như dữ liệu chuỗi thời gian, dữ liệu có tính phụ thuộc nhau…



Hình 5: Mạng RNN (bên trái), mạng FFNN (bên phải)

* + - 1. Mạng nơ ron hồi quy (RNN)

Với mạng nơ ron thông thường thì dữ liệu chúng ta cho vào cùng một lúc, nhưng đôi khi dữ liệu chúng ta lại có quan hệ trình tự với nhau, lúc này khi chúng ta thay đổi vị trí, trình tự của dữ liệu làm mất đi ý nghĩa trình tự sẽ dẫn đến kết quả sai khác. Ví dụ: “Bạn đi học chưa” và “Bạn chưa đi học” khi tách từ ta sẽ được bộ [‘Bạn’, ‘đi’, ‘học’, ‘chưa’], ta thấy sẽ không có sự phân biệt giữa 2 câu trên. Lúc này chúng ta cần một mô hình mạnh mẽ hơn là RNN.

Mạng nơ ron hồi quy (RNN) sử dụng bộ nhớ để lưu giữ lại thông tin, tính toán trước đó phục vụ cho các tính toán dự đoán hiện tại và tương lai để đạt được kết quả tối ưu nhất. Như hình bên dưới, các x đại diện cho đầu vào được chia theo các bước thời gian (time\_step), x(t) là đầu vào cho time\_step t, y(t) là đầu ra của time\_step t (x3 sẽ là vector đầu vào đại diện cho từ thứ 3 trong câu)

A picture containing text, clock, sign

Description automatically generated

Hình 6: Mô hình mạng RNN

Một ví dụ nhỏ khi chưa có RNN về việc hoàn thành câu sử dụng mô hình đếm n-grams language model: Để dự đoán được từ “xe” khi biết từ “tôi” và từ “lái”, sẽ bằng cách lấy số lượng câu “tôi lái xe” xuất hiện trong từ điển chia (/) số lượng câu “tôi lái” trong từ điển ⬄ P(“học” | “tôi”, “lái”) = count(“tôi lái xe”) / count(“tôi lái”). Mô hình đếm như trên có một vài vấn đề là nếu câu “tôi lái xe” chưa bao giờ xuất hiện trong từ điển sẽ cho xác suất bằng 0, nếu từ “tôi lái” không xuất hiện trong từ điển sẽ cho xác suất không tồn tại. Cho nên sau này với sự phát triển của Deep learning thì RNN đã ra đời.

* + - * 1. Ví dụ về RNN mô hình hóa trong xử lý ngôn ngữ

Diagram

Description automatically generated with medium confidence

Hình 7: Ví dụ về xử lý ngôn ngữ bằng RNN

Trong ví dụ trên, đầu tiên vì chưa có giá trị đầu tiên nên có thể sẽ khởi tạo h(0) = 0, sau đó sẽ học giá trị của x(1) - từ “Việt” sau khi word embedding (mỗi từ sẽ có các giá trị riêng của nó trong vector embedding), cho ra được xác suất y(1) là xác suất của từ nào đó có khả năng xuất hiện sau từ “Việt” là cao nhất (ở đây sẽ là từ “Nam” ).

Bước tiếp theo h(2) lúc này sẽ học giá trị của h(1) và x2 sẽ là “Việt” và “Nam”, sau đó tính xác suất của từ nào mà có xác suất đứng sau hai từ “Việt” và “Nam” cao nhất. Tương tự với các từ còn lại.

* + - * 1. Quá trình lưu thông tin trong RNN

Quá trình lưu thông tin được diễn ra trong một cell được thể hiện như hình bên dưới:

Diagram

Description automatically generated

Hình 8: Quá trình lưu thông tin trong cell

Trong đó,

* x(i): có kích thước x
* h(t-1) và h(t): có kích thước h.
* W(hh): h x h
* W(hx): h x x
* y(t): có kích thước y.
* W(S): S = h x y

Như hình bên trên, mong muốn của chúng ta là từ chuỗi lịch sử h(t-1) kết hợp với x(t) cho ra được output output có xác suất mong muốn nhất.

Text

Description automatically generated

Hình 9: Công thức tính h(t) trong xử lý ngôn ngữ

Đầu ra y(t) sẽ được tính theo công thức: softmax là một activation function khá là phổ biến thường được dùng cho các bài toán phân loại (classification) nhiều lớp.

Text

Description automatically generated with medium confidence

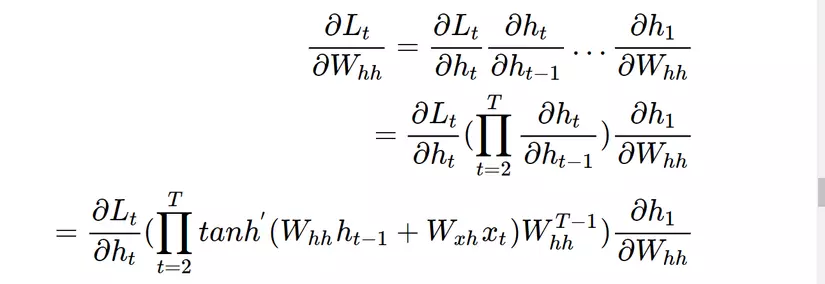
Hình 10: Công thức tính y(t) trong xử lý ngôn ngữ

Trọng số W ở đây chỉ có một và sẽ được dùng cho tất các các cell của mô hình RNN, chỉ khác kích thước để khi kết hợp với các giá trị tạo ra vector có số chiều nhất quán phục vụ cho các phép toán trên vector.

Có thể nói một cell ở đây được coi như là mô phỏng cho việc lưu trữ thông tin để tạo ra một lịch sử mới là h(t), có thể nói các mô hình nâng cao sau này như LSTM đều chỉ tìm ra hàm h(t) tốt nhất, phát triển cho hàm h(t) phức tạp hơn để cho ra được kết quả mong muốn hơn.

* + - 1. Thuật toán lan truyền ngược (BBTT – Backpropagation Through Time)

Trong quá trình training thì chúng ta có ba tham số cần phải tìm là W(hh), W(hx), W(S) hay W(hy), chúng ta cần tính đạo hàm của L theo các W trên (L là loss function) để thực hiện Gradient Descent.

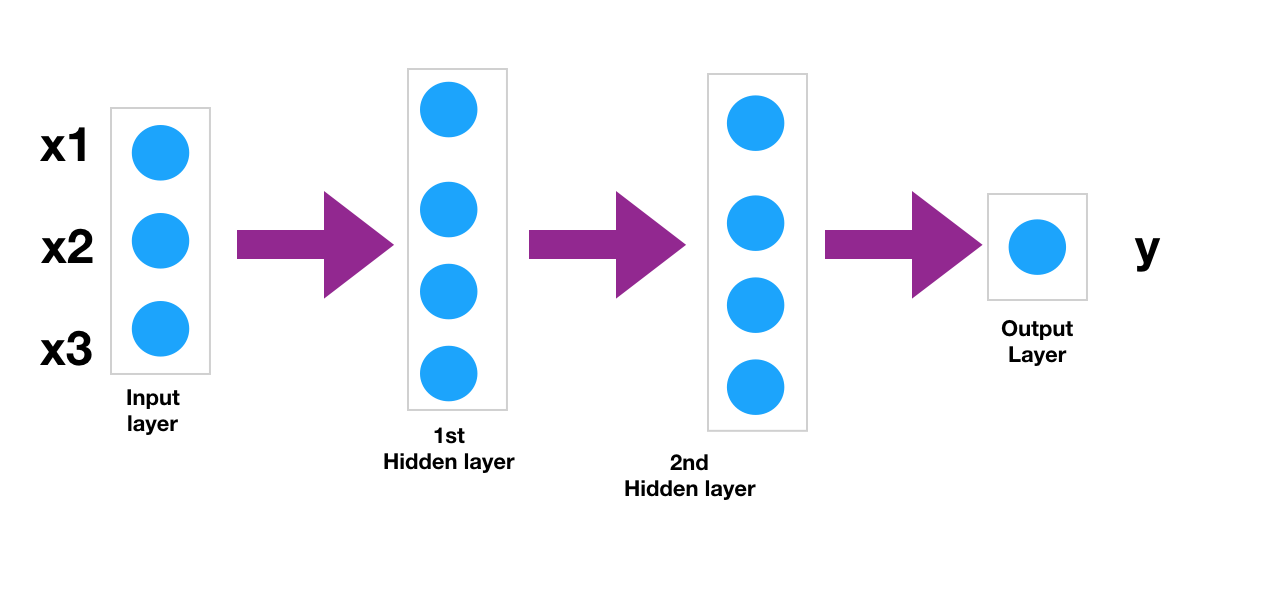


Hình 11: Đạo hàm của L với W

* + - 1. Ưu điểm và nhược điểm của RNN
         1. Ưu điểm
* Có thể áp dụng với các dữ liệu chuỗi nói chung và dữ liệu chuỗi thời gian nói riêng.
* Mạnh mẽ hơn mô hình mạng nơ ron truyền thống.
  + - * 1. Nhược điểm
* Các cell phải tuần tự, phụ thuộc với nhau, nghĩa là muốn tính h(t) là phải tính xong h(t-1), muốn tính h(t+1) phải tính xong h(t) nên mô hình có thể chậm.
* Hiện tượng vanishing: nếu chuỗi (sequence) quá dài thì sẽ có quá nhiều phép nhân và khi trọng số w bé hơn một (do các các hàm activation function trong thuật toán lan truyền ngược BBTT sẽ có giá trị bé hơn một) thì tích của nhiều số bé hơn một sẽ xấp xỉ không, dẫn đến việc cập nhận trọng số sẽ trở nên vô nghĩa.
* Hiện tượng Exploding Gradient: tùy thuộc vào hàm activation function mà làm cho ma trận trở nên lớn hơn.
  + - 1. Vấn đề và giải pháp liên quan đến bài toán
         1. Mô hình tuần tự Sequential

Mô hình Sequential được được gọi là “sequential” vì nó liên quan đến việc xác định một kiến trúc sequential và trong kiến trúc ta có thể thêm vào từng tầng theo mô hình tuyến tính từ đầu vào cho đến đầu ra theo thứ tự.

Recurrent Neural Network được sử dụng phổ biến ở mô hình Sequential này.



Hình 12: Mô hình tuần tự Sequential

* + - * 1. Long-short term memory – LSTM

LSTM là mạng trí nhớ ngắn hạn định hướng dài hạn là một mô hình nâng cấp của RNN có khả năng học đc sự phụ thuộc trong dài hạn.

Bản chất vẫn là RNN nhưng hàm các cell trong LSTM sẽ phức tạp hơn để cso thể nhận đc h(t) tốt nhất.

Diagram

Description automatically generated

Hình 13: Recurrent Neural Network

LSTM khắc phục được những hạn chế của RNN vì RNN mang thông tin từ tầng (layer) trước ra layer sau nhưng chỉ mang qua được một số trạng thái (state) nhất định sau đó sẽ bị vannishing gradient => RNN là short term memory, LSTM sẽ có thêm các cổng (gate) để có thể giữ lại các thông tin cần thiết sẽ được dùng sau (có thể hạn chế được khả năng bị vanishing gradient) khắc phục được những hạn chế của RNN => LSTM sẽ mang được thông tin ở xa hơn => Long short term memory.

Diagram

Description automatically generated

Hình 14: Long short term memory

LSTM có các cổng (gates) bên trong cho phép mô hình huấn luyến tốt hơn với thuật toán BPTT so với với RNN, điều chỉnh luồng thông tin được đưa qua (thông tin nào được lưu trữ, thông tin nào bị xóa đi). LSTM nổi bật khi có thêm một trạng thái ô được sử dụng như một đường dẫn (pathway) để kết nối luồng dữ liệu từ mỗi cổng. Mặt khác LSTM vẫn có thể bị vanishing gradient nhưng mô hình LSTM này sẽ mạnh mẽ hơn, hạn chế được hiện tượng đó.

## Các công cụ sử dụng và một số thư viện.

* + - 1. Google Colaboratory

Google Colaboratory được gọi tắt là Google Colab là một thành quả của Google Research. Mục đích Google Colab là giúp chúng ta chạy code Python trực tiếp thông qua trình duyệt, chúng phù hợp với các công việc liên quan đến phân tích dữ liệu (Data Analysis), máy học (Machine Learning).

Đặc biệt đối với lĩnh vực AI – Deep learning thì sẽ cần đến hàng trăm nghìn phép tính, việc đó đòi hỏi máy phải có cấu hình trung bình đến cao. Để giải quyết vấn đề đó, Google Colab giúp chúng ta chạy thông qua trình duyệt và các tài nguyên cần sử dụng như CPU, GPUs, TPUs sẽ được hệ thống Google Colab cung cấp, được cài đặt sẵn rất nhiều thư viện phổ biến như Pandas, Numpy…, có thể chạy theo từng ô (block) rất thuận tiện cho việc kiểm thử và đánh giá và có thể kết nối trực tiếp với Google Drive.

Hạn chế lớn nhất của hệ thống này sẽ chỉ làm việc liên tục được 12 giờ và với bản Google Colab Pro sẽ là 24 giờ và có thể trùy chọn RAM lên đến 32 GB.

Tóm lại thì Google Colab sẽ phù hợp với những người đang tiếp cận đến lĩnh vực Data, Machine learning và đặc biệt là Deep learning mà chưa đủ kinh phí để đầu tư cấu hình.

Graphical user interface, text, application

Description automatically generated

Hình 15: Giao diện của Google Colab

* + - 1. Thư viện Keras

Keras là một thư viện Python mạnh mẽ và dễ sử dụng được xây dựng dựa trên các thư việc học sâu phổ biến như TensorFlow, Theano… để tạo các mô hình học sâu. Cấu trúc tối thiểu, đơn giản, cộng đồng hỗ trợ lớn. Được phát triển vào năm 2005 bởi Francois Chollet, là một kỹ sư nghiên cứu Deep learning.

Thư viện Keras có hai mô hình là Sequential và API function, với bài toán dạng chuỗi nói chung và dạng chuỗi thời gian nói riêng (được sử dụng cho bài toán dự báo thời tiết) sẽ sử dụng mô hình Sequential (mô hình tuần tự) rất hiệu quả.

Thư viện Keras giúp xây dụng một mô hình học sâu (deep learning) nhanh và dễ dàng. Giúp giúp ta tạo một mô hình theo từng tầng một (layer by layer)

Hỗ trợ xây dựng Convolution Neural Network – CNN (mạng nơ ron tích chập) và cả Recurrent Neural Network – RNN

* CNN là chủ yếu dành cho các vấn đề thị giác máy tính, mạnh mẽ trong xử lý ảnh, giúp xác định các đối tượng, vị trí của các đối tượng, quan hệ của các đối tượng trong hình ảnh.
* RNN chủ yếu xử lý dữ liệu dạng chuỗi, có tính tuần tự như chuỗi thời gian.
  + - 1. Thư viện Pandas

Thư viện Pandas là một thư viện Python cung cấp cấu trúc dữ liệu nhanh, mạnh, linh hoạt, là một thư viện mã nguồn mở được sử dụng rộng rãi bởi khả năng hỗ trợ mạnh mẽ trong các thao tác với dữ liệu. Là một công cụ giúp phân tích và xử lý dữ liệu với ngôn ngữ Python.

Khai báo thư viện sau khi cài đặt một cách dễ dàng: import pandas

Pandas sử dụng một cấu trúc dữ liệu riêng là Dataframe, cung cấp nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này.

Là một công cụ cho phép đọc/ghi (read/write) dữ liệu ở nhiều dạng tập tin như csv, text, excel, sql database,... vào dataframe của Pandas.

Dễ dàng xử lý các giá trị bị thiếu, thêm, sửa, xóa với các trường dữ liệu. Thống kê đánh giá dễ dàng tập dữ liệu bằng hàm info() hoặc describe(), tìm kiếm, sắp xếp, truy xuất có điều kiện với các dữ liệu.

Text, calendar

Description automatically generated

Hình 16: Một Pandas dataframe

* + - 1. Thư viện Numpy

Thư viện Numpy là một thư viện mạnh mẽ, được sử dụng rộng rãi của Python. Cho phép người dùng làm việc hiệu quả với các ma trận (matrix) và mảng (array) với tốc độ xử lý nhanh hơn rất nhiều.

Khai báo thư viện sau khi cài đặt: import numpy

Numpy hỗ trợ tính toán trên các mạng nhiều chiều có kích thước lớn bằng các hàm đã được tối ưu đối với các mảng đó. (tính tổng của mảng, tìm giá trị lớn nhất, nhỏ nhất trong mảng theo từng chiều…)

Ví dụ: đổi mảng 2x3 thành mảng 3x2 như hình bên dưới:

A picture containing diagram

Description automatically generated

Hình 17: Hàm reshape trong Numpy

* + - 1. Thư viện Sklearn

Sklearn hay Scikit-learn là một thư viện Python mã nguồn mở dành cho lĩnh vực máy học (machine learning). Nó cung cấp các công cụ hiệu quả để học máy , các mô hình phân lớp, hồi quy, gom cụm.

Sklearn được phát triển bởi David Cournapeau trong một dự án mùa hè năm 2007, sau đó được một nhóm nghiên cứu trong Viện nghiên cứu Khoa học máy tính và Tự động hóa của Pháp phát triển thêm và công bố bản phát hành đầu tiên vào đầu năm 2010.

Sklearn hỗ trợ các thuật toán hiện đại như KNN, random forest, SVM, neural network,… Các tập dữ liệu được tích hợp sẵn trong thư viện (iris,…), các phương pháp tiền xử lý dữ liệu như PCA (Principal component analysis), đánh giá mô hình (metrics) và con nhiều công cụ khác.

* + - 1. Thư viện Matplotlib

Vẽ biểu đồ là một trong những điều cực kỳ cần thiết trong khi làm việc với dữ liệu, đặc biệt là các lập trình hay không phải lập trình viên muốn phát triển trong lĩnh vực Data Analysis hay Data Science. Thì thư viện Matplotlib là một thư viện không thể bỏ qua.

Matplotlib là một thư viện giúp chúng ta trực quan hóa dữ liệu mạnh mẽ, vẽ đồ đồ họa 2D bằng ngôn ngữ Python. Matplotlib cung cấp một module mạnh mẽ để vẽ biểu đồ là Pyplot Có thể vẽ các biểu đồ để có cái nhìn tổng quan về dữ liệu bằng thư viện này, một số biểu đồ như: bar graph, histogram, scatter plot, area plot, pie plot.

Chart, scatter chart

Description automatically generated

Hình 18: Các dạng biểu đồ trong Matplotlib0

# THIẾT KẾ VÀ CÀI ĐẶT

## Thiết kế hệ thống

* + - 1. Đọc tập dữ liệu

Tập dữ liệu ban đầu có phần mở rộng là rdata, chứa thông tin thời tiết của ba quốc gia trong khoảng thời gian quá khứ là Lào, Thái Lan và Việt Nam

Ta chỉ quan tâm đến tập dữ liệu thời tiết của Việt Nam, tập dữ liệu này được ghi lại trong khoảng thời gian từ 1960 đến năm 2010 của Việt Nam ở nhiều tỉnh thành.

Trong đề tài này ta chỉ quan tâm đến bảy thuộc tính của tập dữ liệu này:

* Ta (average temperature in °C): Nhiệt độ trung bình ở độ C
* Tx (maximal temperature in °C): Nhiệt độ cao nhất ở độ C
* Tm (minimal temperature in °C): Nhiệt độ thấp nhất ở độ C
* Rf (rainfall in mm): Lượng mưa đơn vị mm
* rH (relative humidity in %): Độ ẩm tương đối đơn vị %
* Sh (hours of sunshine): Thời lượng có nắng đơn vị giờ
* aH (absolute humidity in mm Hg): Độ ẩm tuyệt đối.

A black screen with white text

Description automatically generated with low confidence

Hình 19: Tập dữ liệu thời tiết ban đầu của Việt Nam

Tập dữ liệu thời tiết Việt Nam có 32736 dòng dữ liệu, mỗi dòng đại diện cho một tháng của một tỉnh thành nào đó và có các thuộc tính ghi lại các chỉ số thời tiết.

Để thuận tiện cho việc đọc tập dữ liệu và xử lý ta xuất dữ liệu ra tệp mới với phần mở rộng là csv, với tên là “dulieuthoitiet.csv”

**A black screen with white text

Description automatically generated with low confidence**

Hình 20: Dữ liệu tệp dulieuthoitiet.csv

* + - 1. Tiền xử lý dữ liệu

Tiến hành đọc tập dữ liệu thời tiết, sau đó biến đổi sao cho dễ xử lý (gộp cột month và year thành cột chỉ mục(index) của dữ liệu. cái nhìn tổng quan, xử lý các giá trị bị rỗng (NaN), chuẩn hóa dữ liệu (rất cần thiết cho mô hình mạng nơ ron). Tạo tập dữ liệu huấn luyện (training set) và tập dữ liệu kiểm tra (testing set), các tập dữ liệu này phải ở dạng mảng ba chiều (3D array) mà mô hình LSTM có thể xử lý được.

* + - 1. Xây dựng mô hình (thiết kế mạng nơ ron)

Sử dụng mô hình Sequential (mô hình tuần tự) thiêt kế một kiến trúc gồm các tầng ẩn (hidden layers) bên trong mô hình, ứng dụng Stacked LSTM để tạo “độ sâu” cho mô hình, làm mô hình phức tạp hơn (giúp đối tượng cần dự đoán “rõ nét” hơn). Việc làm này sẽ giảm thiểu đi việc sử dụng nhiều nơ ron, giúp mô hình chạy nhanh và chính xác hơn và trong một số nghiên cứu thì “độ sâu” của mạng sẽ quan trọng hơn số lượng nơron.

Mô hình LSTM hoạt động trên dữ liệu tuần tự (sequence data), việc bổ sung thêm các tầng ẩn sẽ làm tăng thêm mức độ trừu tượng hóa của các quan sát đầu vào theo thời gian.

## Cài đặt hệ thống

* + - 1. Đọc tập dữ liệu

Sử dụng pyreadr một gói của Python cho phép đọc và ghi các tệp RData vào hoặc từ khung dữ dữ liệu (dataframe) của thư viện Pandas

Sử dụng hàm read\_r và truyền vào đường dẫn của tệp có phần mở rộng là rdata.

Vì tập dữ liệu này chứa dữ liệu thời tiết của ba quốc gia, nên ta sử dụng hàm keys() để kiểm tra ta được: odict\_keys(['data\_laos', 'data\_thailand', 'data\_vietnam', 'stations\_laos', 'stations\_vietnam']), trong đề tài này ta sẽ làm việc với dữ liệu thời tiết của Việt Nam (data\_vietnam) và lưu lại thành tệp csv.

* + - * 1. Thông tin cơ bản của tập dữ liệu.

Sử dụng hàm info() để xem thông tin của dataframe (sau khi xử lý gộp cột year và month làm chỉ mục):

Text

Description automatically generated

Hình 21: Thông tin của tập dữ liệu

Tập dữ liệu cho thấy có 32736 quan sát, có chỉ mục từ 1960-01-01 đến 2010-12-01, tập dữ liệu gồm có tám cột, trong bảng thông tin trên còn cho biết đối kiểu dữ liệu (Dtype) và số giá trị không rỗng (Non-Null Count) của từng cột.

* + - * 1. Thống kê cơ bản về dữ liệu.

Sử dụng hàm describe() trong thư viện pandas để tạo một bản thống kê dữ liệu, giúp ta có cái nhìn tổng quan ban đầu về dữ liệu, qua thống kê ta thấy độ lệch chuẩn (std) của Rf (lượng mưa) và Sh (thời lượng có năng) khá cao, có thể sẽ ảnh hưởng đến mô hình chúng ta xây dựng.

A picture containing text, screen, silver, screenshot

Description automatically generated

Hình 22: Thống kê tập dữ liệu ban đầu

* Bảng dữ liệu cho biết các số liệu theo từng cột như:
* Count: số lượng giá trị không rỗng.
* Mean: giá trị trung bình.
* Std (standard deviation): độ lệch chuẩn của giá trị.
* Min: giá trị nhỏ nhất.
* 25%, 50%, 75%: các khoảng giá trị.
* Max: giá trị lớn nhất.
  + - * 1. Kiếm tra các giá trị rỗng

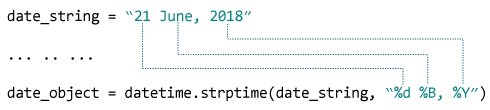
Sử dụng hàm isna() để kiểm tra xem liệu rằng trong ô đó có giá trị hay không, nếu không có giá trị thì trả về TRUE, ngược lại thì FALSE. Sau đó kết hợp với hàm sum() để đếm tổng số giá trị TRUE của mỗi cột dữ liệu ta được số lượng các giá trị rỗng:

|  |  |
| --- | --- |
| Columns | Null values count |
| station | 0 |
| Ta | 1080 |
| Tx | 1381 |
| Tm | 1380 |
| Rf | 984 |
| rH | 1140 |
| Sh | 1896 |
| aH | 804 |

Bảng 4: Thống kê giá trị Null

* + - 1. Tiền xử lý dữ liệu
         1. Thay đổi chỉ mục cho tập dữ liệu.

Nhận thấy dữ liệu có hai cột là year và month quá rườm rà, ta tiến hành gộp hai cột này thành một và lấy chúng làm chỉ mục (index) cho tập dữ liệu. Ta sử dụng hàm strptime(string, format code) tạo đối tượng thời gian (datetime object) từ chuỗi đã cho, nhận vào hai đối số: string, format code:



Hình 23: Hàm strptime() trong datetime

* + - * 1. Xử lý các giá trị bị thiếu (missing values)

Nhận thấy các giá trị NaN không đáng kể nên ta tiến hành xóa các dòng nào có ghi nhận giá trị NaN bằng hàm dropna() của thư viện Pandas với inplace = True để thao tác trực tiếp với dataframe. Sau khi xóa các gái trị NaN thì dữ liệu sẽ còn lại 30479 dòng so với 32736 ban đầu.

* + - * 1. Xử lý bằng Label Encoder

Chúng ta tiến hành thêm một cột là “staion\_encoded” là dữ liệu mã hóa cột dữ liệu “station” về dạng số bằng hàm LabelEncoder(), hàm này mã hóa dữ liệu về dữ liệu số có giá trị từ 0 đến n (n là số lớp của dữ liệu):

Text

Description automatically generated

Hình 24: Thêm cột mã hóa của cột station

Sử dụng hàm values\_count() để thống kê dữ liệu của mỗi “station” xem có đủ số tháng từ năm 1961 đến năm 2010 không:

A screenshot of a phone

Description automatically generated with medium confidence

Hình 25: Thống kê số tháng của station

Thống kê trên cho thấy dữ liệu ở các địa điểm (station) có thể bị thiếu do quá trình ghi nhận và xử lý giá trị NaN phục vụ cho mô hình, có 17 địa điểm là đủ 600 tháng (50 năm) và 48 địa điểm không ghi nhận đủ 50 năm. Sẽ sử dụng dữ liệu của Cần Thơ với ghi nhận là 384 tháng để đánh giá trước.

* + - * 1. Xóa các cột không cần thiết

Sau khi sử dụng các cột station để để thông kê và phân tích, ta tiến hành xóa các cột bằng hàm drop() với inplace = True. Dữ liệu lúc này chỉ còn 7 thuộc tính dùng để huấn luyện mô hình và index là chỉ thời gian ghi nhận của các thuộc tính đó.

Graphical user interface, text, application

Description automatically generated

Hình 26: Tập dữ liệu sau khi xóa các cột không cần thiết

* + - * 1. Hiển thị dữ liệu.

Biểu đồ Ta (nhiệt độ trung bình-độ C)

A picture containing text, antenna, measuring stick

Description automatically generated

Hình 27: Biểu đồ nhiệt độ trung bình

Biểu đồ Tx (biểu đồ nhiệt độ cao nhất-độ C)

A screenshot of a computer

Description automatically generated with low confidence

Hình 28: Biểu đồ nhiệt độ cao nhất

* + - * 1. Scale dữ liệu

Scale dữ liệu là một thao tác rất cần thiết trong các mô hình mạng nơron, vì trong dữ liệu có thể các thuộc tính chênh lệch nhau rất nhiều, ví dụ một thuộc tính nằm trong khoản 0 đến 1000 còn một cột có giá trị trong khoản 0 đến 1 thì sẽ làm cho quá trình huấn luyện không ổn định cho ra mô hình có hiệu quả không cao.

Ta tiến hành scale dữ liệu bằng hàm MinMaxScaler() của thư viện sklearn về trong khoảng 0 tới 1 với fearture\_range=(0, 1), ta được:

Text

Description automatically generated

Hình 29: Dữ liệu sau khi đã scale

* + - 1. Xây dựng mô hình (model).
         1. Thiết kế mạng

Khởi tạo model bằng hàm Sequential() để tạo mô hình tuần tự từng tầng một cùng với mô hình Stacked LSTM (xếp chồng các tầng LSTM lại với nhau như đã giới thiệu ở phần trên)

Mỗi tầng mạng sẽ đòi hỏi một kích thước đầu vào khác nhau, ở đây ta sử dụng hai tầng mạng là LSTM và Dense, có kiến trúc như sau: [input] LSTM (64 noron) => LSTM (32 noron) => LSTM (16 noron) => Dense (7 noron) [output]

Diagram

Description automatically generated

Hình 30: Mô hình Stacked LSTM

Chú ý kiến trúc Stacked LSTM này thì hai tầng LSTM đầu tiên return\_sequential phải là True để có thể tiếp tục tạo ra mảng ba chiều làm đầu vào cho tầng LSTM tiếp theo. LSTM đòi hỏi đầu vào là một mảng ba chiều (3D array) có dạng (samples, time\_steps, features) nên các đầu vào của mô hình (x\_train và x\_test) phải ở dạng mảng ba chiều. Đầu vào của tầng LSTM đầu tiên sẽ được xác định bằng input\_shape(time\_steps, features). LSTM cần xử lý các samples, mỗi samples là một chuỗi thời gian. Tầng Dense cuối cùng yêu cầu đầu vào là một mảng hai chiều (2D array), nên tầng LSTM trước đó sẽ return\_sequential = Flase, đầu ra của tầng Dense này sẽ là bảy noron tương ứng với 7 thuộc tính cần dự đoán.

* + - * 1. Tạo tập dữ liệu huấn luyện

Từ tập dữ liệu thời tiết của Cần Thơ, ta dung 80% dữ liệu làm tập huấn luyện (training) và 20% dữ liệu còn lại làm tập kiểm tra (testing).

Ta có 80% dữ liệu sẽ ứng với 307 dòng, và 20% còn lại sẽ là 77 dòng.

Vì mô hình dự báo thời tiết là mô hình học có giám sát, thuật toán ta sử dụng là LSTM nên ta sẽ dùng 24 tháng trước để dự đoán cho tháng 25 liền kề tương ứng với x\_train và y\_train. Sử dụng vòng lặp từ 12 đến 307, trong từng vòng lặp sử dụng hàm append để thêm dữ liệu vào từng list, sau đó chuyển về dạng numpy array, với sự mạnh mẽ của thư viện Numpy nó giúp ta tự động chuyển về mảng ba chiều. Lúc này dữ liệu của bộ dữ liệu huấn luyện chỉ còn 283 bộ.

Bộ dữ liệu huấn luyện: x\_train: (283, 24, 7), y\_train (283,7)

Text

Description automatically generated

Hình 31: Dữ liệu minh họa x\_train và y\_train

* + - * 1. Tập dữ liệu kiểm tra.

Tương tự với dữ liệu của tập huấn luyện ta tiến hành tạo tập dữ liệu kiểm tra và sẽ lấy lùi về sau 24 dữ liệu để dự đoán cho tháng đầu tiên của tập dữ liệu kiểm tra và chú ý phải chuyển dữ liệu về mảng Numpy tương đương với mảng ba chiều để phù hợp với mô hình.

* + - * 1. Biên dịch mô hình và huấn luyện mô hình

Biên dịch mô hình với thuật toán để tối ưu hóa mô hình: bộ tối ưu hóa (optimizer) là Adam – một dạng của SGD (Stochastic Gradient Descent) và hàm mất mát (loss function) là RMSE (Root Mean Squared Error), huấn luyện mô hình với batch\_size = 1 và epochs = 50. Kết quả của quá trình huấn luyện khá tốt trên tập dữ liệu huấn luyện, tức cho ra loss = 0.0078 ở epochs thứ 50. Mô hình có tổng cộng 34103 tham số và tất cả 34103 tham số này đều được dùng để huấn luyện mô hình.

Graphical user interface, text

Description automatically generated

Hình 32: Kết quả của quá trình huấn luyện

Shape

Description automatically generated with low confidence

Hình 33: Loss của model

* + - 1. Dự đoán thời tiết Cần Thơ với tập dữ liệu kiểm tra dựa trên mô hình đã xây dựng

|  |  |  |
| --- | --- | --- |
| **Attribute** | **RMSE** | **MAE** |
| Ta | 0.6820195674395932 | 0.5068000050334189 |
| Tx | 0.8564067502419317 | 0.6827068824272651 |
| Tm | 0.49036516061814206 | 0.37873618386008523 |
| Rf | 67.53338178414319 | 51.27342019383009 |
| rH | 3.2687604855167676 | 2.4607227315531146 |
| Sh | 114.97666349530546 | 73.52516249619521 |
| aH | 0.9395616201391798 | 0.6787397409414317 |
| Loss | 0,0078 | |

Bảng 5: RMSE và MAE của mô hình dự đoán cho Cần Thơ

Kết quả cho thấy mô hình dự đoán rất tốt cho các thuộc tính Ta, Tx, Tm, rH và aH. Đối với hai thuộc tính Rf và Sh lại cho kết quả không cao do hai chỉ số thởi tiết này có độ lệch chuẩn quá lớn ⬄ độ biến thiên cao, dẫn đến mô hình đã dự báo kông tốt cho hai chỉ số này.

A picture containing text, antenna, measuring stick, needle

Description automatically generated

Hình 34: Biểu diễn giá trị Nhiệt độ Trung bình huấn luyện, giá trị dự đoán và thực tế

**Chart, line chart

Description automatically generated**

Hình 35:Biểu diễn giá trị Nhiệt độ Trung bình dự đoán và thực tế

A picture containing text, needle, tool

Description automatically generated

Hình 36: Biểu diễn giá trị Độ ẩm tương đối huấn luyện, giá trị dự đoán và thực tế

**Chart, line chart

Description automatically generated**

Hình 37: Biểu diễn giá trị Độ ẩm tương đối dự đoán và thực tế

**A picture containing text, measuring stick, needle

Description automatically generated**

Hình 38: Biểu diễn giá trị Độ ẩm tuyệt đối huấn luyện, giá trị dự đoán và thực tế

Chart, line chart

Description automatically generated

Hình 39: Biểu diễn giá trị Độ ẩm tuyệt đối dự đoán và thực tế

# KẾT QUẢ THỰC NGHIỆM

## Kiểm thử

* + - 1. Trường hợp kiểm thử 1: Scale bằng hàm RobustScaler vs Min-Max Scaler (Cần Thơ)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **RobustScaler** | | **Min-Max Scaler** | |
| **Attribute** | **RMSE** | **MAE** | **RMSE** | **MAE** |
| Ta | 0.8005539275029574 | 0.6119152217716366 | 0.6820195674395932 | 0.5068000050334189 |
| Tx | 1.0553750387834868 | 0.8427532790543197 | 0.8564067502419317 | 0.6827068824272651 |
| Tm | 0.7732462960758721 | 0.5384800997647372 | 0.49036516061814206 | 0.37873618386008523 |
| Rf | 105.09567887538918 | 81.27903365098037 | 67.53338178414319 | 51.27342019383009 |
| rH | 4.0570610861972005 | 3.096407415216619 | 3.2687604855167676 | 2.4607227315531146 |
| Sh | 112.7809865776253 | 75.432732995764 | 114.97666349530546 | 73.52516249619521 |
| aH | 1.3032713902349096 | 0.9168649450525062 | 0.9395616201391798 | 0.6787397409414317 |
| Loss (50 epochs) | 0.0633 | | 0.0078 | |

* + - 1. Trường hợp kiểm thử 2: Tăng số nơron ở tầng LSTM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **256-128-64-7** | | **128-64-32-7** | |
| **Attribute** | **RMSE** | **MAE** | **RMSE** | **MAE** |
| Ta | 0.8084615511744739 | 0.6411370909059203 | 0.6391199510513551 | 0.46261956351143974 |
| Tx | 0.9593927030209705 | 0.785188848322088 | 0.7423062156070208 | 0.582739396528764 |
| Tm | 0.658748965285054 | 0.4948633615072672 | 0.48124296144901113 | 0.3616143610570337 |
| Rf | 90.19831232548658 | 68.5609094378236 | 65.07490345473119 | 47.37608209616178 |
| rH | 3.3511581454776502 | 2.495662386014864 | 2.813973656383875 | 2.0028855400580863 |
| Sh | 122.15625465775132 | 79.48428320946631 | 121.7413191742295 | 78.07914829006442 |
| aH | 0.9374722333846823 | 0.713417355425946 | 0.8603358072708774 | 0.6417407345462156 |
| Loss (50 epochs) | 0.0072 | | 0.0078 | |

* + - 1. Trường hợp kiểm thử 3: Thử mới Station Sapa và Đà Nẵng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Sapa** | | **Đà Nẵng** | |
| **Attribute** | **RMSE** | **MAE** | **RMSE** | **MAE** |
| Ta | 2.1327712226775692 | 1.341699981689453 | 1.6495030021813497 | 1.0976337482402851 |
| Tx | 1.590471665401553 | 1.1115931590398154 | 1.2061724687066815 | 0.9550602355560694 |
| Tm | 2.268391667483213 | 1.3808624807993573 | 4.287177251193953 | 2.3256918969092433 |
| Rf | 107.71739067481766 | 81.55222116470337 | 195.5404338795344 | 111.50807363336736 |
| rH | 5.497920403824476 | 3.9227962493896484 | 3.1198568951997365 | 2.3240452833299514 |
| Sh | 51.49426125516092 | 30.92209696451823 | 40.6630364727855 | 32.2770713112571 |
| aH | 1.0547134626085792 | 0.8363718398412069 | 1.3994597899780283 | 1.062551795662224 |
| Loss (50 epochs) | 0.0065 (7 phút ) | | 0.0079 (4 phút) | |

* + - * 1. Biểu diễn dự đoán Ta (average temperature):

A picture containing diagram

Description automatically generated

Hình 40: Biểu diễn dự đoán Ta của Sapa

Chart, line chart

Description automatically generated

Hình 41: Biểu diễn dự đoán Ta của Đà Nẵng

* + - * 1. Biễu diễn dự đoán rH (relative humidity)

Chart

Description automatically generated

Hình 42: Biểu diễn dự đoán rH của Sapa

Chart, line chart

Description automatically generated

Hình 43: Biểu diễn dự đoán rH của Đà Nẵng

* + - * 1. Biễu diễn dự đoán Sh (hours of sunshine)

Chart

Description automatically generated

Hình 44: Biểu diễn dự đoán Sh của Sapa

Chart, line chart

Description automatically generated

Hình 45: Biểu diễn dự đoán Sh của Đà Nẵng

* + - 1. Trường hợp kiểm thử 4: với timestep = 24 (Cần Thơ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Time\_step=12** | | **Time\_step=24** | | **Time\_step=36** | |
| **Attribute** | **RMSE** | **MAE** | **RMSE** | **MAE** | **RMSE** | **MAE** |
| Ta | 0.8005539275029574 | 0.6119152217716366 | 0.6404739158377026 | 0.4674148460487268 | 0.66451464299506 | 0.5175058389638926 |
| Tx | 1.0553750387834868 | 0.8427532790543197 | 0.7760883785638966 | 0.6191423886782162 | 0.8204805291002916 | 0.6650717698134385 |
| Tm | 0.7732462960758721 | 0.5384800997647372 | 0.5109919232066868 | 0.39369462248566856 | 0.49949668466031016 | 0.3993187693806437 |
| Rf | 105.09567887538918 | 81.27903365098037 | 56.91604584941483 | 42.09016306837658 | 71.38465196030575 | 53.98196908034287 |
| rH | 4.0570610861972005 | 3.096407415216619 | 3.3965317991749937 | 2.5738469269987823 | 3.6995220368908446 | 2.894262643789316 |
| Sh | 112.7809865776253 | 75.432732995764 | 113.57518919741231 | 72.75863366065086 | 115.14351748805868 | 75.07292428945566 |
| aH | 1.3032713902349096 | 0.9168649450525062 | 0.8705836878937037 | 0.6459246845988484 | 0.8989355881917336 | 0.7043441970627031 |
| Loss | 0.0079 | | 0.0078 | | 0.0078 | |

## Đánh giá

* Đối với kiểm thử tăng số nơron sẽ làm cho model huấn luyện lâu hơn ( với mô hình 256-128-64-7 thời gian huấn luyện với 50 epoch sẽ mất 9 phút 39 giây ) mà kết quả cũng không thay đổi nhiều.
* Trường hợp kiểm thử 1: thay đổi phương pháp scale, kết quả cho thấy phương pháp scale bằng Min-Max Scaler sẽ cho kết quả tốt hơn khá rõ rệt.
* Thay đổi time\_step sẽ cho kết quả của các chỉ số đánh giá có thay đổi tốt ở time\_step bằng 24, cụ thể là ở chỉ số thời tiết Rf cho ra RMSE khoảng 57. Và ở các chỉ số khác cũng cho chỉ số RMSE tốt hơn so với 2 trường hợp time\_step = 12 và time\_step = 36
* Thay đổi các địa điểm dự đoán sẽ cho kết quả khác nhau vì mỗi tỉnh thành sẽ có thời tiết khác nhau nhưng nhìn chung các tỉnh thành đều có độ lệch chuẩn cao ở Rf (Rainfall) và Sh (Hours of sunshine). Trong 3 địa điểm dự đoán trong đề tài này thì Đà Nẵng có độ lệch chuẩn của Rf cao nhất (249.152402) và thấp nhất là Cần Thơ (119.470364), và độ lệch chuẩn của Sh ở Sapa sẽ là thấp nhất (44.285443), Cần Thơ và Đà Nẵng thì tương đương nhau (khoảng 66)

# PHẦN KẾT LUẬN

## Kết quả đạt được

* + - 1. Kỹ năng
* Chủ động sắp xếp thời gian để xây dựng đề tài đúng yêu cầu đề ra hằng tuần.
* Tham khảo được nhiều bài báo khoa học, bài luận trong nước và ngoài nước trong quá trình tìm kiếm tài liệu.
* Hiểu được cách hoạt động của mô hình RNN, cũng như LSTM
* Nâng cao khả năng tự học, tự tìm kiếm tài liệu, phân tích tài liệu, cũng như định hướng được mục tiêu sẽ làm cho chủ đề.
  + - 1. Chương trình
* Hiểu và xây dựng được mô hình mạng nơ ron.
* Khả năng phân tích dữ liệu tốt hơn.
* Ứng dụng được các kiến thức đã học ở môn Khai khoáng dữ liệu và Nguyên lý máy học.

## Hướng phát triển

* Xây dựng xử lý các dữ liệu có độ lệch chuẩn lớn.
* Xây dựng thêm ứng dụng (ứng dụng web) để demo mô hình
* Thay đổi thiết kế mạng khác: thêm nhiều tầng (layers) hơn thay đổi số noron khác hơn.
* Tìm hiểu và ứng dụng các pre-trained model.

# TÀI LIỆU THAM KHẢO

1. Wikipedia – Bách khoa toàn thư mở. Bộ nhớ dài-ngắn hạn
2. Jason Brownlee (2017), Multivariate Time Series Forecasting with LSTMs in Keras, Deep Learning for Time Series, https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/
3. Jason Brownlee (2017), How to Convert a Time Series to a Supervised Learning Problem in Python, Deep Learning for Time Series, https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/
4. Quy Nguyen (2021), 11. Các phương pháp scale dữ liệu trong machine learning, https://ndquy.github.io/posts/cac-phuong-phap-scaling/
5. To Duc Thang (2020), Làm quen với Keras, https://viblo.asia/p/lam-quen-voi-keras-gGJ59mxJ5X2
6. FuchsMichaelAndi (2019), Feature Scaling with Scikit-Learn, https://michael-fuchs-python.netlify.app/2019/08/31/feature-scaling-with-scikit-learn/#normalize-or-standardize
7. Santhoopa Jayawardhana (2020), Sequence Models & Recurrent Neural Networks (RNNs), https://towardsdatascience.com/sequence-models-and-recurrent-neural-networks-rnns-62cadeb4f1e1#:~:text=Sequence
8. Tanvir (2020), Word Embedding and One Hot Encoding, https://medium.com/intelligentmachines/word-embedding-and-one-hot-encoding-ad17b4bbe111
9. Christopher Olah (2015), Understanding LSTM Networks, https://colah.github.io/posts/2015-08-Understanding-LSTMs/
10. Huynh Chi Chung (2020), Giới thiệu về Deep learning, thư viện Keras, https://viblo.asia/p/gioi-thieu-ve-deep-learning-thu-vien-keras-63vKjDGAl2R
11. Jason Brownlee (2017), Difference Between Return Sequences and Return States for LSTMs in Keras, https://machinelearningmastery.com/return-sequences-and-return-states-for-lstms-in-keras/
12. Jason Brownlee (2017), Stacked Long Short-Term Memory Networks, https://machinelearningmastery.com/stacked-long-short-term-memory-networks/
13. Trần Trung Trực (2020), Optimizer- Hiểu sâu về các thuật toán tối ưu ( GD,SGD,Adam,..), https://viblo.asia/p/optimizer-hieu-sau-ve-cac-thuat-toan-toi-uu-gdsgdadam-Qbq5QQ9E5D8
14. Tek4 (2021), API Mô Hình Tuần Tự – Sequential (Đơn Giản) – Keras Cơ Bản, https://tek4.vn/api-mo-hinh-tuan-tu-sequential-don-gian-keras-co-ban
15. Shipra Saxena (2021), Introduction to Long Short Term Memory (LSTM), https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/
16. Aakarsha Chug (2021), Deep Learning | Introduction to Long Short Term Memory, https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/
17. Ike Sri Rahayu, Esmeralda C Djamal, Ridwan Ilyas (2020), Jenderal Achmad Yani university, Indonesia, Daily Temperature Prediction Using Recurrent Neural Networks and Long-Short Term Memory, http://www.ieomsociety.org/detroit2020/papers/540.pdf
18. What is time series data?, influxdata, https://www.influxdata.com/what-is-time-series-data/
19. Tuan Nguyen (2019), Long short term memory (LSTM), https://nttuan8.com/bai-14-long-short-term-memory-lstm/