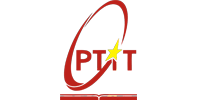
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TP.HỒ CHÍ MINH**

-----------------------------------------------------------------------------------------



**ĐỀ TÀI DỰ BÁO THỜI TIẾT BẰNG MÔ HÌNH**

**RECURRENT NEURAL NETWORK**

**Môn học : Nhập môn trí tuệ nhân tạo**

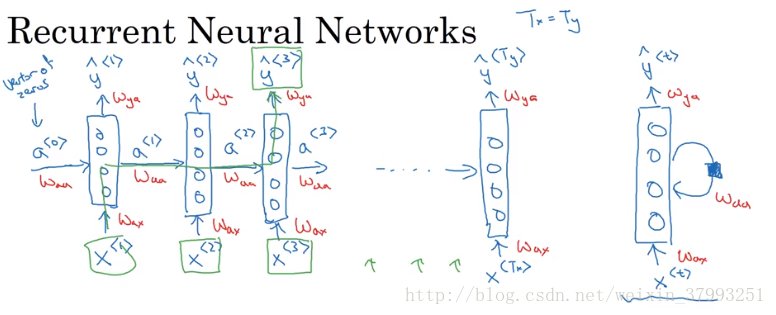
Giảng viên : Nguyễn Thị Bích Nguyên

Họ và tên : Lê Ngọc Huy

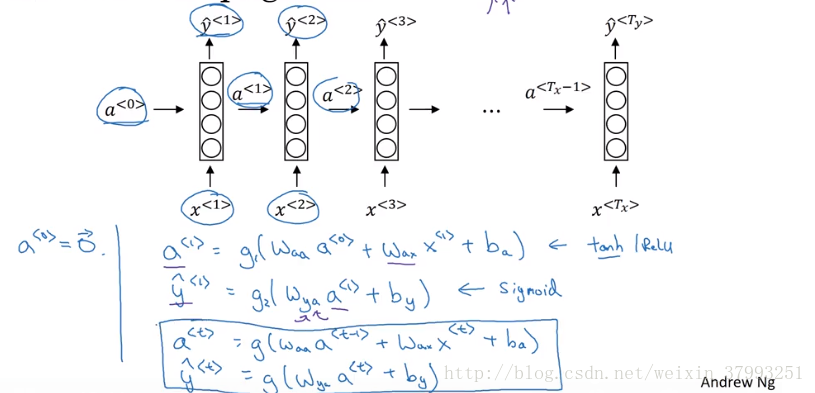
Mã SV : N17DCCN056

TP.Hồ Chí Minh, ngày 6, tháng 6, năm 2020.

1. **MÔ HÌNH RECURRENT NEURAL NETWORK:**



forward propagation



Ta có:

* Mô hình có Tx input và Ty output, các input được cho vào model theo đúng thứ tự  *x*1​,*x*2​,…*x*t​.
* Mỗi thanh chứa được gọi là 1 one-hot, one-hot t có input là *xt*​ và *at*−1​ (output của one-hot trước) là *at*​=*g*(*Wax*∗*xt*​+*Waa*∗*at*−1​+ba). là activation function thường là tanh hoặc ReLu.
* Có thể thấy *at*​ mang cả thông tin từ one-hot trước ( *at*−1​) và input của one-hot hiện tại => *at*​ giống như memory nhớ các đặc điểm của các input từ *x*1​ đến *xt*​
* *a*0 được thêm vào chỉ cho chuẩn công thức nên thường được gán bằng 0 hoặc giá trị ngẫu nhiên. Có thể hiểu là ban đầu chưa có dữ liệu gì để học thì memory rỗng.
* Output của mỗi one-hot sau khi có thông tin từ input là ). g là activation function có thể là softmax.
  + Đánh giá mô hình:

|  |  |
| --- | --- |
| Ưu điểm | Hạn chế |
| * Khả năng xử lý đầu vào có độ dài bất kỳ   • Kích thước mô hình không tăng theo kích thước đầu vào  • Tính toán có tính đến thông tin lịch sử  • Trọng lượng được chia sẻ theo thời gian | • Tính toán chậm  • Khó truy cập thông tin cũ  • Không thể xem xét bất kỳ đầu vào trong tương lai cho trạng thái hiện tại |

1. **ỨNG DỤNG DỰ BÁO THỜI TIẾT BẰNG MÔ HÌNH RECURRENT NEURAL NETWORK:**

Dự báo thời tiết có một vai trò to lớn trong đời sống kinh tế, xã hội. Dự báo đúng sẽ giúp con người đưa ra những quyết định đúng đắn. Dự báo sai, bạn biết rồi đấy, có thể sẽ là những hậu quả vô cùng khủng khiếp.

Với bài toán dự báo như thế này, mình quy nó về Time Series Analysis để có thể dễ hiểu và dễ áp dụng hơn. Bạn có thể tìm thấy một số bài toán thuộc thể loại này, như phân tích tài chính, tỉ số chứng khoán, hay áp dụng nó để phân tích tỉ giá Bitcoin để đầu tư đúng chỗ, đúng thời điểm.

Với bài toán này, bạn có thể thử nghiệm trên các thông số thời tiết.

* nhiệt độ (°C)
* độ ẩm (%)
* lượng mưa (mm)
* tốc độ gió (km/h)
* hướng gió (độ)
* lượng mây (%)

…….

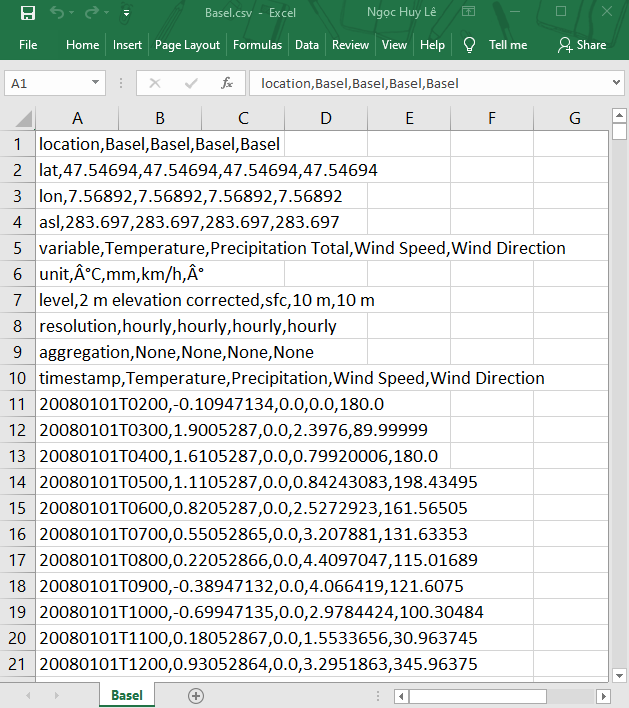
Output: thông số thời tiết của 720h

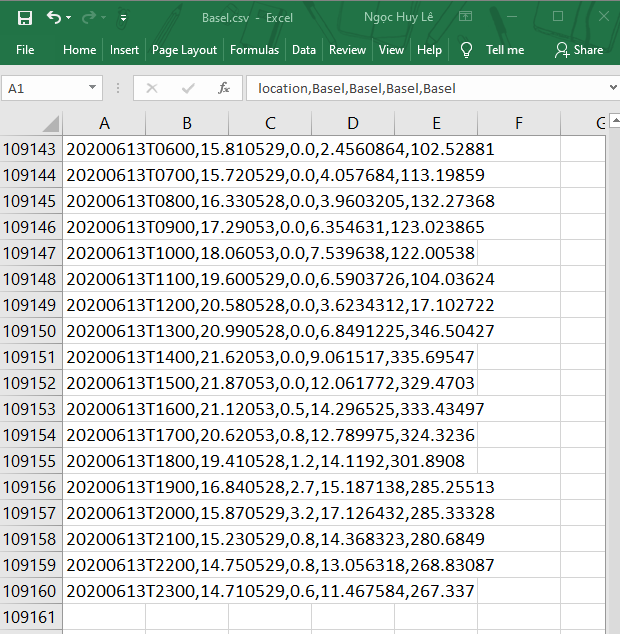
Ta sẽ làm theo các bước sau đây để xây dựng được ứng dụng:

* Bước 1: Chuẩn bị dữ liệu

Dữ liệu là một yếu tố bắt buộc trong tất cả các bài toán Machine Learning. Với bài toán dự báo thời tiết, bạn có thể download dữ liệu thời tiết Hà Nội https://www.meteoblue.com. Trang này cung cấp đầy đủ và chính xác cho các bạn cấc thông tin lượng mưa, độ ẩm, nhiệt độ, nắng gió… Nhưng cũng có một nhược điểm đó là họ chỉ cho phép bạn download thông tin thời tiết của 15 ngày gần nhất.

Nhưng trang web cho phép tải về bản thử nghiệm về thành phố Basel với bộ dữ liệu thời tiết trong 30 năm. Sau khi download về, tôi có được file dữ liệu dạng như sau:





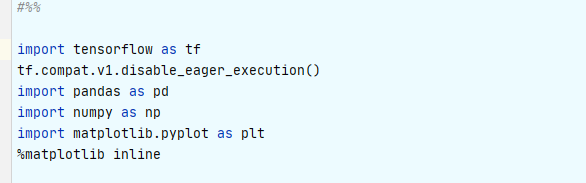
Dữ liệu tôi tải về từ 02h00 ngày 01/01/2008 đến 23h00 ngày 13/06/2020.

Nhiều bạn sẽ thắc mắc với dữ liệu như trên thì chúng ta sẽ dự đoán nhiệt độ cho ngày hôm sau thế nào đây? Có lẽ chúng ta cần xử lý nó một chút mới có thể sử dụng được dữ liệu này.

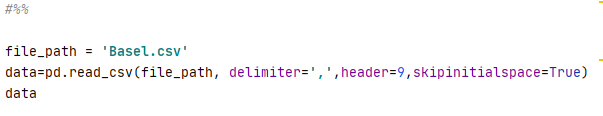
* Bước 2: Xử lý dữ liệu

Tôi sử dụng thư viện tensorflow để hỗ trợ quá trình training, Về cơ bản thì tensorflow đã được tích hợp rất nhiều các thuật toán khác nhau, dễ dàng sử dụng, và giúp giảm thời gian xây dựng các hệ thống deep learning. Đồng thời kết hợp với pandas và numpy để phân tích, và xử lý cấu trúc data, và matplotlib dùng để về đồ thị.

Việc vẽ đồ thị rất quan trọng đối với các bài toán thuộc dạng Time Series Analysis như thế này. Vì dĩ nhiên việc đoán trước không thể trả về kết quả chính xác 100% được, Kết quả sẽ là tương đối và có thể có một chút sai số không đáng kể. Vì thế việc vẽ đồ thị sẽ giúp bạn dễ dàng so sánh giữa kết quả dự đoán và thực tế.

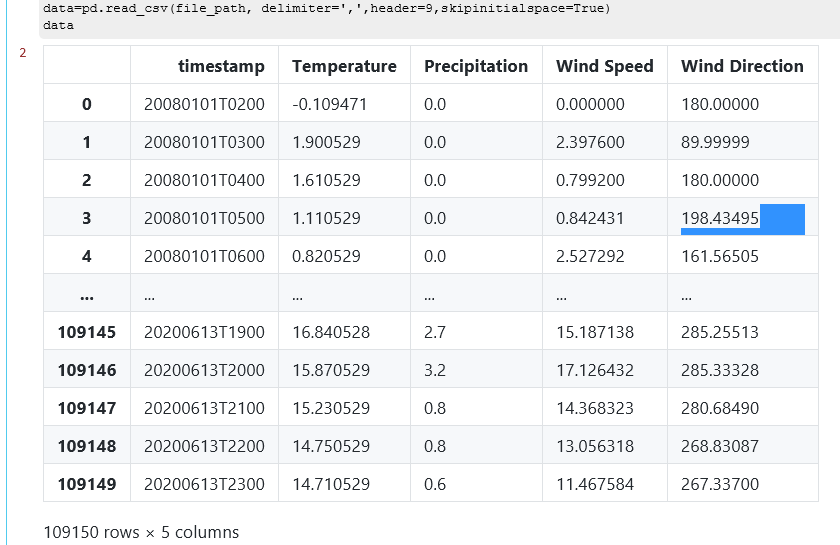
Đầu tiên import các thư viện sẽ được dùng:

Load training data:



Header sẽ là vị trí header trong file dữ liệu. Như bạn thấy trong file csv bên trên Header của dữ liệu ở dòng số 10, nói các khác nếu xét theo 1 array thì index=9. Nếu file dữ liệu của bạn không có header thì có thể đổi thành header=None

Ta được dữ liệu như sau:



Như vậy ta đã có được file dữ liệu training. bây giờ bạn chỉ cần gọi data [‘Temperature’] hoặc data[‘Wind Speed’] … để sử dụng:



Ở đây tôi lấy dữ liệu là Temperature để sử dụng, nghĩa là ta có:

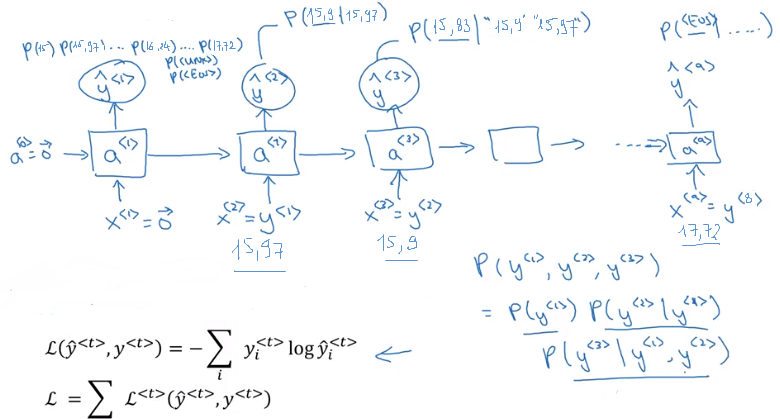


Nhiều bạn có thể sẽ thắc mắc với dữ liệu chỉ có X như vậy thì chúng ta sẽ lấy Y ở đâu để training, để predict đây. Thật ra nó cũng không hoàn toàn mơ hồ khó hiểu vậy đâu, hãy cứ nghĩ đơn giản như thế này:

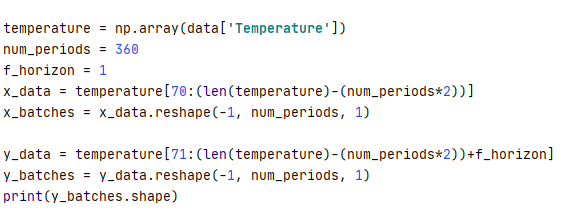
Bạn có 1 chuỗi các giá trị array, các số tiếp theo sẽ phụ thuộc vào các số trước đó, cũng như vậy chúng ta sẽ có các giá trị thời tiết tại thời điểm trước đó sẽ là X và giá trị thời tiết tại thời điểm trên sau 1 giờ sẽ là Y. Ví dụ:

| **X** | **Y** |
| --- | --- |
| 15.97 | 15.9 |
| 15.9 | 15.83 |
| 15.83 | 15.85 |
| 15.85 | 16.24 |
| 16.24 | 16.69 |
| 16.69 | 17.22 |
| 17.22 | 17.72 |
| 17.72 | 18.25 |

Ta sử dụng mô hình one-to-more để giải thích:

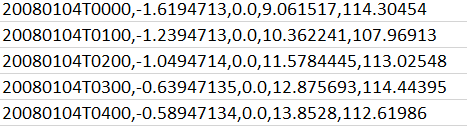


Vì dữ liệu quá lớn 109149 giá trị nên tôi chia nhỏ thành các mảng dữ liệu nhỏ để train, mỗi mảng có num\_periods là 360. Vì chia đều nên sẽ dư ra 70 giá trị đầu vì vậy giá trị đầu tiên lúc này sẽ sở vị trí 71 xét theo array = 70.

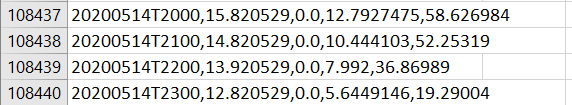


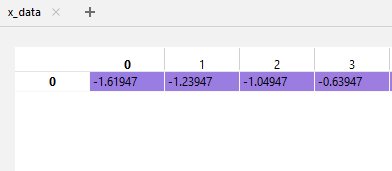
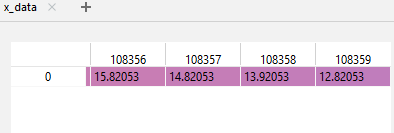
Như vậy trong bài toán tôi sẽ dùng dữ liệu x\_train từ giá trị 04/01/2008 00:00

đến ngày hiện tại – 720h là 23h00 ngày 14/05/2020, 720h còn lại này sẽ là thời gian để lại test với hết quả dự đoán, khi đó chúng ta sẽ có đến 720 giá trị test để so sánh. Và dữ liệu y sẽ là giá trị từ 04/01/2008 01:00

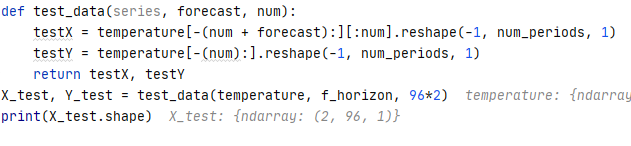


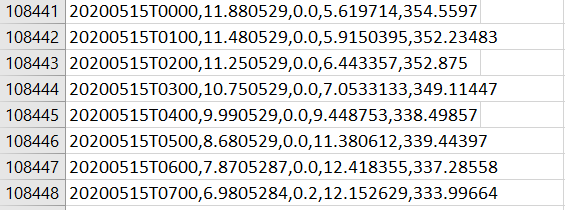
……..



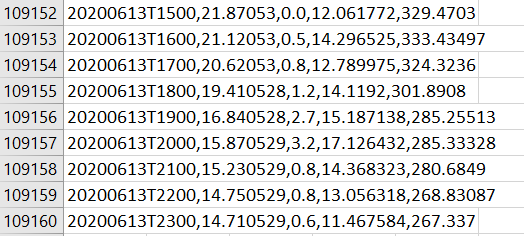


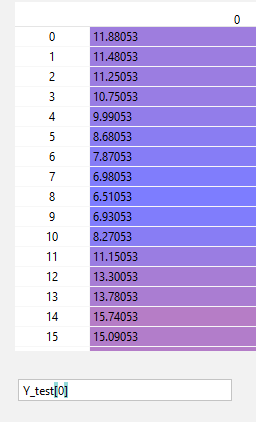
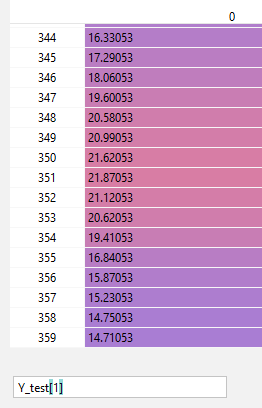
* Dữ liệu test:





…





Như vậy file dữ liệu Basel.csv sẽ chia ra như sau:

00h 4/1/2008

00h 15/5/2020…

23h 14/5/2020

70h bỏ

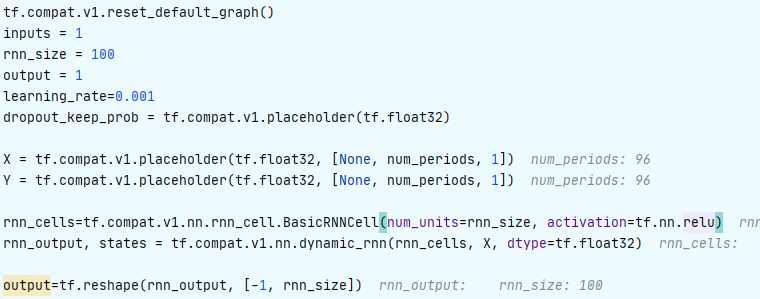
720h test

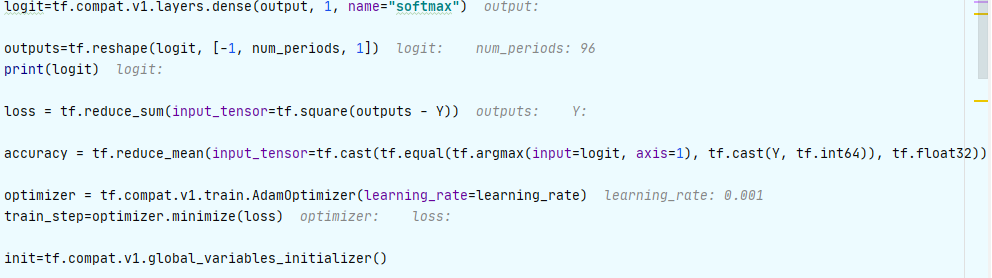
108360h train

Bước 3: Training Model

Do đặc thù của bài toán này là dùng các thông tin trong quá khứ, để dự đoán tương lai, nên thuật toán ở đây tôi lựa chọn là Recurrent Neeural Network (RNN) hay trong tiêng Việt chúng ta gọi nó là Mạng Nơ-ron hồi quy Ý tưởng chính của RNN (Recurrent Neural Network) là sử dụng chuỗi các thông tin. Ví dụ, nếu muốn đoán từ tiếp theo có thể xuất hiện trong một câu thì ta cũng cần biết các từ trước đó xuất hiện lần lượt thế nào. RNN được gọi là hồi quy (Recurrent) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó.

Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó. Trong nhiều mạng neural truyền thống khác, dữ liệu đầu vào và đầu ra hoàn toàn độc lập với nhau, tức là chúng không có liên kết thành chuỗi. Do đó khi áp dụng vào bài toán dự báo thời tiết của tôi sẽ rất khó để đưa ra kết quả dự đoán.





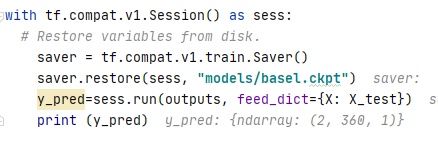
Ở đây khi khai báo và sử dụng mạng neural, tôi sử dụng mô hình RNN truyền thống với hàm activation là relu. Bạn có thể chuyển qua và test thử với mô hình khác của RNN như LSTM (Long Short-Term Memory).

Trainning và save model:

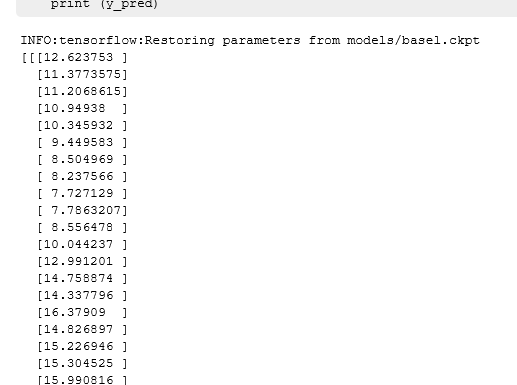
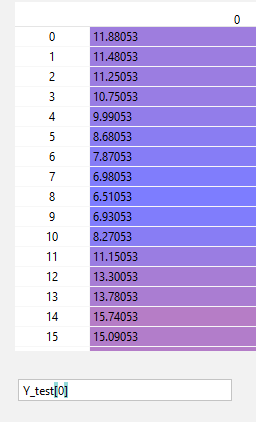


Bước 4: Test dữ liệu

Vậy là quá trình training đã hoàn thành. bây giờ cần phải test lại thành quả của chúng ta. Khi test dữ liệu, bạn cần restore lại models mà mình vừa training lúc nãy để sử dụng.



Ta có được kết quả là y\_pred và so sánh với dữ liệu 720h test lúc trước ta có:



Nếu chỉ nhìn vào dữ liệu đầu ra như thế kia tôi cũng không biết liệu kết quả dự đoán của chúng ta sẽ ra sao. Do đó để dễ dàng so sánh, việc vẽ đồ thị sẽ rất cần thiết để giúp chúng ta có cái nhìn chính xác để đánh giá mô hình dự đoán này. Kết quả:

