TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo

Description automatically generated

**DỰ ÁN CUỐI KỲ**

**MÔN NHẬP MÔN HỌC MÁY**

*Người hướng dẫn*: **THẦY LÊ ANH CƯỜNG**

*Người thực hiện*: **NGUYỄN PHAN MINH KHANG – 51900351**

**LÊ NGỌC PHƯƠNG VI -51900464**

**TRẦN MINH PHI -51900408**

Khoá  **: 23**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo

Description automatically generated

**DỰ ÁNCUỐI KỲ**

**MÔN NHẬP MÔN HỌC MÁY**

*Người hướng dẫn*: **THẦY LÊ ANH CƯỜNG**

*Người thực hiện*: **NGUYỄN PHAN MINH KHANG – 51900351**

**LÊ NGỌC PHƯƠNG VI -51900464**

**TRẦN MINH PHI -51900408**

Khoá  **: 23**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

LỜI CẢM ƠN

Lời đầu tiên, nhóm em bao gồm Nguyễn Phan Minh Khang, Lê Ngọc Phương Vi và Trần Minh Phi xin gửi đến thầy (cô) khoa Công Nghệ Thông Tin, trường Đại học Tôn Đức Thắng lời cảm ơn chân thành. Đặc biệt, tập thể nhóm xin gửi đến thầy Nguyễn Anh Cường, người đã tận tình hướng dẫn và giúp đỡ để chúng em có thể hoàn thành bài tiểu luận này, lời cảm ơn chân thành nhất.

Vì kiến thức của nhóm còn nhiều hạn chế, trong quá trình làm bài còn nhiều sai sót, nhóm em kính mong sẽ nhận được những ý kiến đóng góp của quý thầy (cô).

Tập thể nhóm xin chân thành cảm ơn!

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của Thầy Nguyễn Anh Cường. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Phan Minh Khang*

*Lê Ngọc Phương Vi*

*Trần Minh Phi*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

MỤC LỤC

**DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ**

**DANH MỤC HÌNH**

*Hình 1.1.1. Các mô-đun lặp của mạng RNN chứa một layer*

*Hình 1.1.2. Các mô-đun lặp của mạng LSTM chứa bốn layer*

*Hình 1.1.3. Các kí hiệu sử dụng trong mạng LSTM*

*Hình 1.2.1. Tế bào trạng thái LSTM giống như một băng truyền*

*Hình 1.2.2. Cổng trạng thái LSTM*

*Hình 1.2.3. LSTM focus f*

*Hình 1.2.4. LSTM focus i*

*Hình 1.2.5. LSTM focus c*

*Hình 1.3.1. Biến thể nối 2 cổng loại trừ và đầu vào với nhau*

*Hình 1.3.2. Biến thể Gated Recurrent Unit*

**CHƯƠNG 1 – MẠNG BỘ NHỚ DÀI NGẮN**

**(LONG – SHORT TERM MEMORY -LSTM)**

* 1. **Giới thiệu chung**

Là một dạng đặc biệt của mạng Nơ-ron hồi quy (RNN) dựa trên Gradient. Trong quá trình hoạt động nó cho phép cắt bỏ những Gradient dư thừa. Trong quá trình học LSTM có thể thu hẹp thời gian trễ dư thừa của các bước thực hiện thông qua tập hằng số lỗi (theo Hoch Reiter & Schmid Huber- 1997).

LSTM được giới thiệu bởi Hoch Reiter & Schmid Huber (1997), và sau đó đã được cải tiến và phổ biến bởi rất nhiều nhà nghiên cứu trong học máy. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau.

LSTM được thiết kế để giải quyết vấn đề phụ thuộc dài (long-term dependency). Đặc tính của mạng là có thể nhớ thông tin trong suốt thời gian dài chứ không cần phải huấn luyện mạng để nhớ được (nội tại của mạng đã có thể ghi nhớ được mà không cần bất kì can thiệp nào).

LSTM có các thành phần cơ bản sau:

+ Trạng thái tế bào (cell state)

+ Cổng (gates)

+ Sigmoid

+ Tank

LSTM được thiết kế nhằm loại bỏ vấn đề phụ thuộc quá dài. Ta quan sát lại mô hình RNN bên dưới, các layer đều mắc nối với nhau thành các mô-đun neural network. Trong RNN chuẩn, mô-đun repeating này có cấu trúc rất đơn giản chỉ gồm một lớp đơn giản là tầng tank (**tank layer)**.

Diagram

Description automatically generated

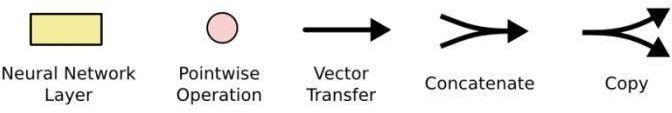
*Hình 1.1.1. Các mô-đun lặp của mạng RNN chứa một layer*

Về kiến trúc mạng LSTM**:** giống như RNN, nó là một chuỗi các mô-đun được lặp đi lặp lại. Tuy nhiên, xét về cấu trúc thì LSTM có 4 tầng mạng nơ-ron tương tác với nhau một cách rất đặc biệt, gọi đó là các tầng ẩn (hidden layer). Một số biến thể của LSTM được thực hiện dựa trên việc thay đổi vị trí kết nối giữa các tầng và cổng.

Diagram

Description automatically generated

*Hình 1.1.2. Các mô-đun lặp của mạng LSTM chứa bốn layer*



Lớp ẩn mạng

Nơ-ron

Toán tử

Pointwise

Truyền

Véc-tơ

Phép nối các

toán hạng

Sao chép

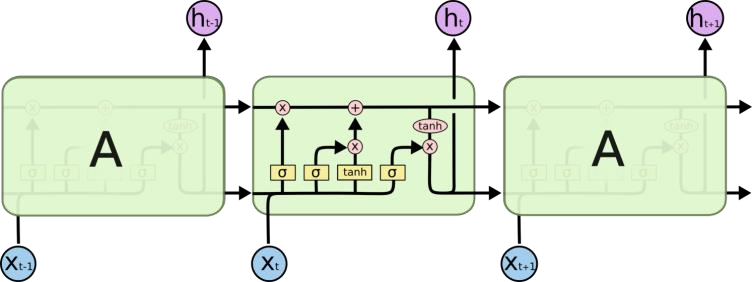
*Hình 1.1.3. Các kí hiệu sử dụng trong mạng LSTM*

Các ký hiệu sử dụng trong mạng LSTM được như hình 1.1.3 sau đây:

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

#### **Phân tích mô hình LSTM**

LSTM là một mạng nơ ron hổi quy có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.



*Hình 1.2.1. Tế bào trạng thái LSTM giống như một băng truyền*

Chìa khóa của LSTM là trạng thái tế bào (cell state). Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Nhờ đó các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.

A picture containing text, clock

Description automatically generated

LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate).

Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.

Tầng sigmoid sẽ cho đầu ra là một số trong khoản [0, 1] [0,1], mô tả có bao nhiêu thông tin có thể được thông qua.

Khi đầu ra là 00 thì có nghĩa là không cho thông tin nào qua cả Khi là 11 thì có nghĩa là cho tất cả các thông tin đi qua nó.

Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

Graphical user interface, application

Description automatically generated

*Hình 1.2.2. Cổng trạng thái LSTM*

Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là “tầng cổng quên” (forget gate layer).

Nó sẽ lấy đầu vào là ht-1 và xt rồi đưa ra kết quả là một số trong khoảng [0, 1] [0,1] cho mỗi số trong trạng thái tế bào Ct-1.

Đầu ra là 11 thể hiện rằng nó giữ toàn bộ thông tin lại, còn 00 chỉ rằng toàn bộ thông tin sẽ bị bỏ đi.

*Diagram

Description automatically generated**Hình 1.2.3. LSTM focus f*

Đầu tiên là sử dụng một tầng sigmoid được gọi là “tầng cổng vào” (input gate layer) để quyết định giá trị nào sẽ cập nhật.

Diagram

Description automatically generated with low confidence

*Hình 1.2.4. LSTM focus i*

Tiếp theo là một tầng tanh tạo ra một véc-tơ cho giá trị mới nhằm thêm vào cho trạng thái. Sau đó kết hợp 2 giá trị đó lại để tạo ra một cập nhập cho trạng thái.

* + 1. **Cập nhập trạng thái tế bào cũ Ct-1 thành trạng thái mới Ct.**

Thực hiện nhân trạng thái cũ với *ft* để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm it Ct.

Trạng thái mới thu được này phụ thuộc vào việc ta quyết định cập nhập mỗi giá trị trạng thái ra sao.

A screenshot of a computer

Description automatically generated with low confidence

*Hình 1.2.5. LSTM focus c*

**1.2.2. Quyết định đầu ra.**

Giá trị đầu ra sẽ dựa vào trạng thái tế bào, nhưng sẽ được tiếp tục sàng lọc. Đầu tiên, ta chạy một tầng sigmoid để quyết định phần nào của trạng thái tế bào muốn xuất ra. Sau đó, ta đưa nó trạng thái tế bảo qua một hàm *tank* để có giá trị nó về khoảng [-1, 1], và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn.

A picture containing text, clock

Description automatically generated

#### **Một số biến thể của LSTM**

Một dạng LTSM phổ biến được giới thiệu bởi [Gers & Schmidhuber](ftp://ftp.idsia.ch/pub/juergen/TimeCount-IJCNN2000.pdf) [(2000)](ftp://ftp.idsia.ch/pub/juergen/TimeCount-IJCNN2000.pdf) được thêm các đường kết nối “peephole connections”, làm cho các tầng cổng nhận được giá trị đầu vào là trạng thái tế bào.

A picture containing diagram

Description automatically generated

* + 1. **Biến thể nối 2 cổng loại trừ và đầu vào với nhau.**

Thay vì phân tách các quyết định thông tin loại trừ và thông tin mới thêm vào, ta sẽ quyết định chúng cùng với nhau luôn. Bỏ đi thông tin khi mà ta thay thế nó bằng thông tin mới đưa vào. Đưa thông tin mới vào khi ta bỏ thông tin cũ nào đó đi.

*Diagram

Description automatically generated**Hình 1.3.1. Biến thể nối 2 cổng loại trừ và đầu vào với nhau.*

* + 1. **Biến thể Gated Recurrent Unit (GRU)**

Nó kết hợp các cổng loại trừ và đầu vào thành một cổng “cổng cập nhập” (update gate). Nó cũng hợp trạng thái tế bào và trạng thái ẩn với nhau tạo ra

A picture containing text, clock

Description automatically generatedmột thay đổi khác. Kết quả là mô hình của ta sẽ đơn giản hơn mô hình LSTM chuẩn và ngày càng trở nên phổ biến.

*Hình 1.3.2. Biến thể Gated Recurrent Unit*

* 1. **Những khác biệt của LSTM so với mô hình truyền thống RNN**

\*Trong Vanilla RNN, chúng ta chỉ sử dụng tanh function với dữ liệu đầu vào là Current input (xt) và thông tin lưu trữ từ timestep trước (Hidden state ht-1). Tuy nhiên trong LSTM và GRU, ta sử dụng kết hợp tanh và sigmoid function cùng với các thuật toán để quyết định thông tin nào nên được lưu trữ và thông tin nào nên được loại bỏ.

\*Ngoài việc sử dụng hai dữ liệu đầu vào là Current input (xt) và Hidden state ht-1 như mạng RNN thông thường, LSTM còn sử dụng thêm một đầu vào nữa là Cell state (c t-1). Ở đây chúng ta có thể coi Cell state là ‘memory’ của mạng LSTM và nó có thể lưu trữ thông tin của các timestep đầu tiên, do đó hạn chế ảnh hưởng của short-term memory như trong Vanilla RNN.

A picture containing text, clock, device

Description automatically generated

Diagram, schematic

Description automatically generated

Graphical user interface, text, application

Description automatically generated

\*Tất cả các RNN có các vòng phản hồi trong lớp lặp lại. Điều này cho phép chúng duy trì thông tin trong bộ nhớ theo thời gian. Nhưng, có thể khó đào tạo RNN tiêu chuẩn để giải quyết các vấn đề đòi hỏi phải học phụ thuộc thời gian dài hạn. Điều này là do độ dốc của hàm mất phân rã theo cấp số nhân theo thời gian (được gọi là vấn đề độ dốc biến mất). Mạng LSTM là một loại RNN sử dụng các đơn vị đặc biệt ngoài các đơn vị tiêu chuẩn. Các đơn vị LSTM bao gồm một "ô nhớ" có thể duy trì thông tin trong bộ nhớ trong thời gian dài. Một bộ cổng được sử dụng để kiểm soát khi thông tin đi vào bộ nhớ, khi nó xuất ra và khi nó bị lãng quên. Kiến trúc này cho phép chúng học các phụ thuộc dài hạn.

\*Các RNN tiêu chuẩn (Mạng thần kinh tái phát) bị biến mất và bùng nổ các vấn đề về độ dốc. Các LSTM (Bộ nhớ dài hạn) giải quyết các vấn đề này bằng cách giới thiệu các cổng mới, chẳng hạn như cổng đầu vào và cổng quên, cho phép kiểm soát tốt hơn dòng chảy gradient và cho phép bảo quản tốt hơn các phụ thuộc tầm xa của Node.

\*Các LSTM thường được gọi là RNN ưa thích. Vanilla RNNs không có trạng thái tế bào. Chúng chỉ có các trạng thái ẩn và những trạng thái ẩn đó đóng vai trò là bộ nhớ cho RNN.

Trong khi đó, LSTM có cả trạng thái tế bào và trạng thái ẩn. Trạng thái ô có khả năng loại bỏ hoặc thêm thông tin vào ô, được điều chỉnh bởi "cổng". Và vì "tế bào" này, theo lý thuyết, LSTM sẽ có thể xử lý sự phụ thuộc lâu dài (trong thực tế, rất khó để làm như vậy).

\*Khi chuyển từ RNN sang LSTM (Bộ nhớ ngắn hạn dài hạn), ta sẽ có nhiều nút điều khiển hơn, điều khiển luồng và trộn. Đầu vào theo trọng lượng được đào tạo. Và do đó, mang lại sự linh hoạt hơn trong việc kiểm soát đầu ra. Vì vậy, LSTM cung cấp cho chúng ta khả năng kiểm soát nhiều nhất và do đó, kết quả tốt hơn. Nhưng cũng đi kèm với sự phức tạp và chi phí hoạt động nhiều hơn.

* 1. **Kết luận**

LSTM là một cải tiến cho RNN với mục đích giúp cho tất cả các bước của RNN có thể truy vấn được thông tin từ một tập thông tin lớn hơn.

Ví dụ, nếu chúng ta sử dụng RNN để tạo mô tả cho một bức ảnh, nó có thể lấy một phần ảnh để dự đoán mô tả từ tất cả các từ đầu vào.LSTM hoạt động thực sự tốt hơn nhiều so với RNN cho nhiều bài toán.

**CHƯƠNG 2 - ỨNG DỤNG, MÔ TẢ DỮ LIỆU VÀ MÔ TẢ THUẬT TOÁN**

**BÀI 1**

**Mô tả dữ liệu và mục tiêu bài toán:**

Dữ liệu sử dụng mô hình LSTM để giải quyết bài toán thuộc loại Time Series. Đây là dữ liệu để dự đoán nhiệt độ của thời tiết qua từng khoảng thời gian trong ngày.

Lấy dữ liệu:

Dữ liệu được lấy từ trang: <https://www.worldweatheronline.com/ho-chi-minh-city-weather-history/vn.aspx>

Mô tả dữ liệu:

* Dữ liệu được lưu trong file weather\_data.csv
* là dữ liệu về thời tiết trong năm 2018.
* Dữ liệu gồm 12 cột 2920 dòng
* Ý nghĩa của các cột:
  + time: thời gian (str, format hh:mm)
  + month: tháng (float) (tạm thời, có thể chuyển về str)
  + temperature: nhiệt độ môi trường (float, độ C)
  + feelslike: nhiệt độ cảm nhận được (float, độ C)
  + wind: tốc độ gió (float, km/h)
  + direction: hướng gió (str)
  + gust: tốc độ gió tối đa (float, km/h)
  + cloud: độ che phủ của mây (float, %)
  + humidity: độ ẩm không khí (float, %)
  + precipitation: lượng mưa (float, mm)
  + pressure: áp suất không khí (float, bm)
  + weather: tình hình thời tiết hiện tại (str)

## Data Train:

Về bộ dữ liệu để cho training thì lấy 80% từ bộ dữ liệu weather tải về.

## Data Test:

Về bộ test thì sẽ lấy 20% từ bộ dữ liệu weather tải về.

**Mô tả thuật toán:**

Do tải file dữ liệu lên google driver nên cần liên kết google colab với google driver có thể lấy dữ liệu ra từ google driver và tiến hành xử lý dữ liệu trước khi cho vào mô hình mạng. Graphical user interface, application

Description automatically generated

Import các thư viện cần thiết và load dữ liệu

Chúng ta import 1 số thư viện quan trọng, để tí nữa chúng ta sử dụng. Ở đây tụi em sử dụng các thư viện:

* numpy: Giúp xử lý số liệu
* panda: dùng để đọc dữ liệu
* pyplot: Dùng để vẽ biểu đồ
* MinMaxScaler: Dùng để chuẩn hóa dữ liệu
* Và một số thư viện để lập trình cho mô hình mạng dựa trên thư viện keras

Graphical user interface, text

Description automatically generated

Tiếp theo là load data từ tập dữ liệu weather.csv được lưu trong đường link trên google drive /content/driver/My Drive/Deep-Learning/stock5/weather.csv.

Table

Description automatically generated with medium confidence

Thực hiện lấy data bằng thư viện panda, chuyển data về dạng numpy sau đó tiến hành chia dữ liệu thành 2 bộ, một bộ train chiếm 80% dữ liệu chính, và một bộ test chiếm 20% còn lại của dữ liệu.

Text

Description automatically generated

Thực hiện chia dữ liệu X, X là các cột đầu

Calendar

Description automatically generated with medium confidence

Thực hiện chia dữ liệu y, y là cột cuối

Graphical user interface, text, application, email

Description automatically generated

Thực hiện đếm các giá trị trong y

Graphical user interface, text, application

Description automatically generated

Chuẩn hóa dữ liệu

Application

Description automatically generated with low confidence

Tiếp theo là đến phần xử lý dữ liệu. time\_step: Trong bài toán này thì ta hiểu là nếu muốn cứ 30 giá trị của Open thì đoán 1 giá trị open tiếp theo thì time\_step ở đây bằng 30. Còn num\_predict là 1. Từ đó ta thấy hàm get\_data ở trên mục đích là định dạng lại dữ liệu để tí có thể đưa vào mạng. Ví dụ, sau khi qua hàm get\_data thì: x\_train = [[1,2,3,4,5,6,7,8,9,10],[2,3,4,5,6,7,8,9,10,11]] và y\_train = [11,12] Bản chất là muốn dùng [1,2,3,4,5,6,7,8,9,10] để đoán ra 11, [2,3,4,5,6,7,8,9,10,11] để đoán ra 12.

Graphical user interface, text, application

Description automatically generated

Chia bộ dữ liệu thành tập train và tập test. Lấy 30 dữ liệu để đoán 1 dữ liệu tiếp theo. Sau đó chuẩn hóa dữ liệu về dạng từ 0 đến 1, theo hàm MinMaxScaler() cho bộ train và test. Mục đích của chuẩn hóa là để tí nữa vào mô hình mạng nó tối ưu tốt hơn.

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

Graphical user interface, application

Description automatically generated

Table

Description automatically generated

Tiếp theo, chúng ta sẽ reshape lại cho x\_train và y\_train.Tại sao phải reshape, bởi vì theo quy chuẩn của keras thì đầu vào của LSTM có dạng (batch\_size, time\_steps, feature) như sau:

Batch size: Cứ hiểu là có bao nhiêu cặp (time\_steps, feature) ấy time\_steps: Như trình bày ở trên rồi feature: là có bao nhiêu thuộc tính của mỗi phần tử trong time\_step. Ví dụ : time\_step có 10 giá trị (mỗi giá trị là một vector), mỗi vector là một giá trị 2 chiều chẳng hạn, thì feature ở đây là 2 (tức 2 chiều đó ). Tóm lại, thì feature cứ hiểu là số thuộc tính của mỗi phần tử time\_step. Còn reshape đầu ra mục đích là tí cho hợp với shape đầu ra của mô hình mạng. Ở trên ta thấy dùng 30 để đoán 1, nên đầu ra ở đây phải reshape theo (-1,1).

Graphical user interface, application

Description automatically generated

Tiếp theo là chỉnh lại đầu vào tập train

Graphical user interface, text

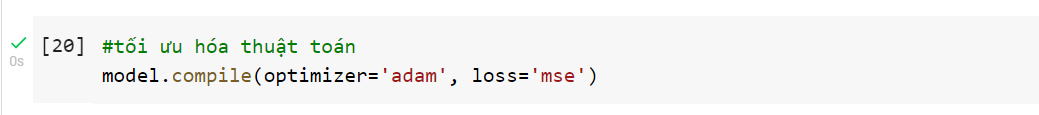
Description automatically generated

Tiếp theo chúng ta sẽ xây dựng mô hình mạng. Bước đầu tiên, chúng ta cần khởi tạo lớp Sequential. Đây sẽ là lớp mô hình của chúng em và chúng em sẽ thêm các lớp LSTM, Dropout và Dense cho mô hình này. Ở trên ta thêm vào 3 lớp LSTM liên tiếp, và cứ qua 1 lớp là có 1 dropout 0.3 . Cuối cùng ta cho qua một tầng Dense với đầu ra là 1 chiều.

Graphical user interface, text, application, email

Description automatically generated

Sử dụng hàm loss bình phương trung bình là hàm mất mát và để tối ưu hóa thuật toán, chúng em sử dụng trình tối ưu hóa adam.



Sau đó thực hiện việc huấn luyện mạng. Thực hiện huấn luyện mạng với validation là 0.2, epochs =100 và với batch\_size =30, tiến hành lưu lại mô hình trong file 30\_to\_1.h5 được lưu theo đường dẫn /content/driver/My Drive/Deep-Learning/stock5/30\_to\_1.h5 trong google driver.

Table

Description automatically generated

Cuối cùng chúng ta sẽ chạy kết quả test

Graphical user interface

Description automatically generated with medium confidence

Ta thực hiện lấy model đã lưu ra và tiến hành dự đoán bằng test\_output = model.predict(x\_test) Sau khi dự đoán xong thì chúng ta thực hiện inverse transform lại, do lúc nãy ta đã chuẩn hóa chúng nằm trong 0->1, bây giờ ta chuyển nó về đúng giá trị thật của chúng bằng đoạn mã:

test\_1 = scaler.inverse\_transform(test\_output)

test\_2=scaler.inverse\_transform(y\_test)

**BÀI 2**

**Mô tả dữ liệu và mục tiêu bài toán:**

Dữ liệu sử dụng ở đây là Breast Cancer Wisconsin (Diagnostic) từ Kaggle. Đây là dữ liệu dự đoán xem bệnh ung thư lành tính hay ác tính. Các đặc điểm được tính toán từ hình ảnh số hóa của một kim hút nhỏ (FNA) của khối vú.

Thông tin của thuộc tính:

1. ID number
2. Diagnosis (M = malignant, B = benign)

Mười đặc điểm có giá trị thực được tính toán cho mỗi nhân tế bào:

a) radius (giá trị trung bình của khoảng cách từ tâm đến các điểm trên chu vi)

b) texture (độ lệch chuẩn của các giá trị gray-scale)

c) perimeter

d) area

e) smoothness (biến thể cục bộ trong độ dài bán kính)

f) compactness (chuvi2 / diện tích - 1.0)

g) concavity (mức độ nghiêm trọng của các phần lõm của đường viền)

h) concave points (số phần lõm của đường viền)

i) symmetry

j) fractal dimension ("coastline approximation" - 1)

**Mô tả thuật toán:**

Đầu tiên, import các thư viện cần thiết và load dữ liệu. Dữ liệu được download từ Kaggle (nguồn dữ liệu được lấy ở: <https://www.kaggle.com/uciml/breast-cancer-wisconsin-data>).

Graphical user interface, text, application

Description automatically generated

Graphical user interface, table

Description automatically generated

Tiếp theo, thu thập dữ liệu ở các giá trị trong cột diagnosis để hình dung số lượng giá trị của các cột 'diagnosis' dễ hiểu hơn.

Graphical user interface, text

Description automatically generated

Tìm trong data trong các cột có giá trị null để dễ thực hiện các tính toán trong data

Table

Description automatically generated

A picture containing chart

Description automatically generated

Quan sát thấy được trong data có 2 biến: biến độc lập và biến phụ thuộc nên ta chia ra làm 2 loại: x là data không có cột giá trị diagnosis và y là cột giá trị diagnosis

A picture containing company name

Description automatically generated

Kế tiếp, nhằm thuận tiện cho tính toán ta sẽ chuyển các giá trị trong y về binary

Text

Description automatically generated with low confidence

Sau đó, chuẩn hoá dữ liệu để chuẩn bị cho bước áp dụng vào các mô hình học máy

Graphical user interface, text, application, Word

Description automatically generated

**Áp dụng các phương pháp học máy khác nhau:**

**1.ANN:**

Thuật toán Artificial Neural Network (ANN) gồm 3 thành phần chính: Input layer và output layer chỉ gồm 1layer, hidden layer có thể có 1 hay nhiều layer tùy vào bài toán cụ thể. ANN hoạt động theo hướng mô tả lại cách hoạt động của hệ thần kinh với các neuron được kết nối với nhau

Trong ANN, trừ input layer thì tất cả các node thuộc các layer khác đều full-connected với các node thuộc layer trước nó. Mỗi node thuộc hidden layer nhận vào ma trận đầu vào từ layer trước và kết hợp với trọng số để ra được kết quả.

Graphical user interface, text

Description automatically generated

Table

Description automatically generated

1. **Clustering:**

Thuật toán K-Means Clustering để phân cụm các điểm dữ liệu dựa theo vị trí tương đối của chúng so với nhau. K-Means Clustering là thuật toán đơn giản và phổ biến, được biết đến như một trong những thuật toán Unsupervised Learning được sử dụng nhiều nhất.

Text, table

Description automatically generated with medium confidence

Graphical user interface, text, application, email

Description automatically generated

1. **SVM:**

SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại.

Text

Description automatically generated

Ta tính độ chính xác trên dữ liệu và kết quả cho ra là 0.972Graphical user interface, text, application

Description automatically generated

1. **LR:**

Kỹ thuật này được sử dụng rộng rãi trong lĩnh vực y tế, nơi xử lý các vấn đề phân loại nhị phân. Mô hình LR cho ta thấy độ chính xác là 0.97

Graphical user interface, text, application

Description automatically generated

1. **kNN:**

Thuật toán K láng giềng gần nhất (tiếng Anh: K-Nearest Neighbor - KNN) là một kĩ thuật học có giám sát (supervised learning) dùng để phân loại quan sát mới bằng cách tìm điểm tương đồng giữa quan sát mới này với dữ liệu sẵn có.

Graphical user interface, text, application

Description automatically generated

Kết quả cho cho thấy độ chính xác đạt 0.962

1. **Perceptron:**

Perceptron Learning Algorithm, gọi ngắn là Perceptron, là một thuật toán giúp chúng ta thực hiện công việc phân loại với hai lớp. Perceptron sẽ đọc từng dữ liệu và điều chỉnh biên giới sao cho tất cả các điểm nằm cùng một phía của biên giới có nhãn giống nhau.

Graphical user interface, text, application

Description automatically generated

Độ chính xác của Perceptron là 0.957

1. **Random Forest:**

Random Forests là thuật toán học có giám sát (supervised learning). Nó có thể được sử dụng cho cả phân lớp và hồi quy. Nó cũng là thuật toán linh hoạt và dễ sử dụng nhất. Random forests tạo ra cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và chọn giải pháp tốt nhất bằng cách bỏ phiếu. Nó cũng cung cấp một chỉ báo khá tốt về tầm quan trọng của tính năng.

Graphical user interface, text, application, email

Description automatically generated

Độ chính xác của Random Forest mang lại là 0.962

1. **Naïve Bayes:**

Bộ phân lớp Bayes là một giải thuật thuộc lớp giải thuật thống kê, nó có thể dự đoán xác suất của một phần tử dữ liệu thuộc vào một lớp là bao nhiêu. Phân lớp Bayes được dựa trên định lý Bayes (định lý được đặt theo tên tác giả của nó là Thomas Bayes)

Graphical user interface, text, application, email

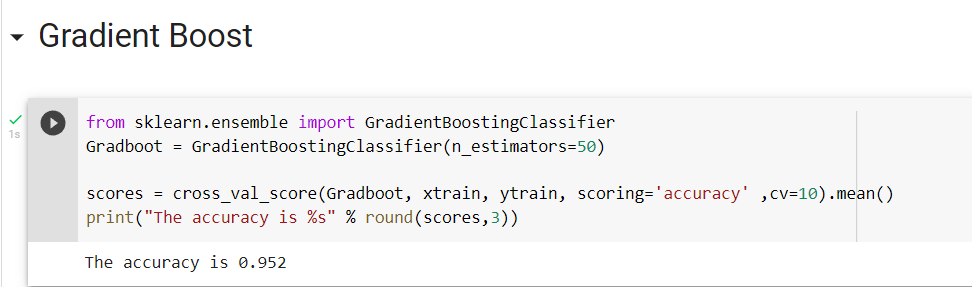
Description automatically generated

Và độ chính xác Naïve Bayes có được là 0.937

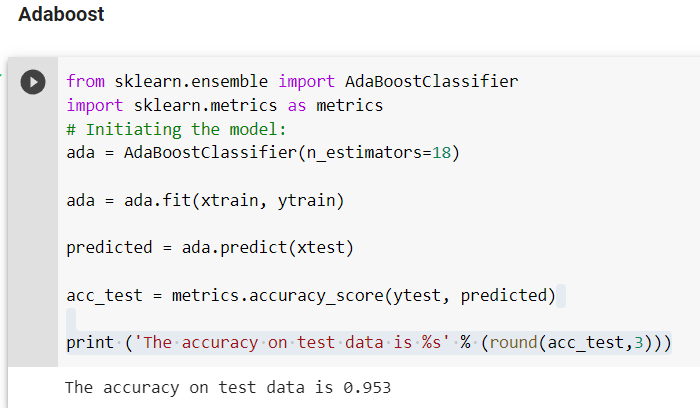
1. **Adaboost:**

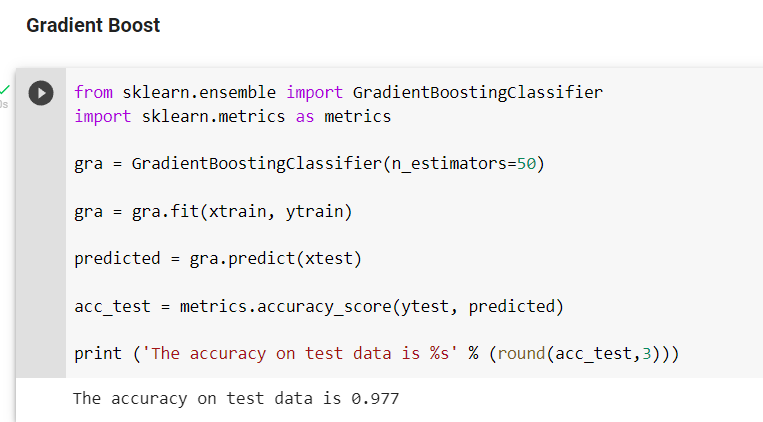
Adaboost là một thuật toán boosting dùng để xây dựng bộ phân lớp (classifier). Một thuật toán tổng hợp sử dụng các phương pháp bagging và boosting để phát triển một công cụ dự đoán nâng cao.

AdaBoost tương tự như Random Forest theo nghĩa là các dự đoán được lấy từ nhiều cây quyết định ****

1. **Gradient boost:**Gradient Boost cũng là một thuật toán tổng hợp sử dụng các phương pháp thúc đẩy (boosting) để phát triển một công cụ dự đoán nâng cao. Theo nhiều cách, Gradient Boost tương tự như AdaBoost ****

**Thử trên test data:**

****

****

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

**TÀI LIỆU THAM KHẢO**

[**https://qastack.vn/stats/222584/difference-between-feedback-rnn-and-lstm-gru**](https://qastack.vn/stats/222584/difference-between-feedback-rnn-and-lstm-gru)

[**https://lib.hpu.edu.vn/bitstream/handle/123456789/34455/Lieu-Vuong-Phuc-Minh-CT2101M.pdf?sequence=1**](https://lib.hpu.edu.vn/bitstream/handle/123456789/34455/Lieu-Vuong-Phuc-Minh-CT2101M.pdf?sequence=1)