TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN NHẬP MÔN HỌC MÁY**

**TRÌNH BÀY LÝ THUYẾT**

*Người hướng dẫn*: **PGS-TS. LÊ ANH CƯỜNG**

*Người thực hiện*: **NGUYỄN ĐỨC TÍN – 51800248**

**NGUYỄN VIỆT TÂN – 51800621**

Lớp**: 18050303**

Khoá**: 22**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn thầy Lê Anh Cường đã giảng dạy tận tình trong suốt quá trình học tập, để chúng em có đủ kiến thức lý thuyết và thực hành. Chúng em xin chúc cho thầy thật nhiều sức khỏe và gặt hái được những thành công trong cuộc sống.

Với lòng biết ơn sâu sắc nhất, chúng em xin gửi đến quý Thầy Cô tại Trường Đại Học Tôn Đức Thắng đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng tôi trong suốt thời gian học tập tại trường.

Một lần nữa, chúng tôi xin chân thành cảm ơn thầy. Bước đầu đi vào thực tế, tìm hiểu về môn Nhập Môn Học Máy, kiến thức của chúng em còn hạn chế và còn nhiều bỡ ngỡ. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, chúng em mong sẽ nhận được sự đóng góp ý kiến từ thầy để có thể tìm ra những vấn đề còn hạn chế và bổ sung, nâng cao kiến thức của mình, phục vụ cho công tác thực tế sau này

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của PGS-TS. Lê 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 Đức Tín*

*Nguyễn Việt Tân*

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

Như chúng ta đã biết, Neural Network được xây dựng để mô phỏng lại cách hoạt động của bộ não con người. Đối với mạng neural thông thường, mỗi sự kiện đầu vào x được xử lý một cách độc lập và đưa ra đầu ra y tương ứng mà không có sự trao đổi thông tin thu thập được tại mỗi đầu vào x trong mạng. Tuy nhiên, bộ não con người hoạt động một cách phức tạp hơn thế nhiều với sự kết hợp của nhiều dạng thông tin và sự kiện với nhau để đưa ra kết luận cuối cùng. Ví dụ như khi bạn đang đọc câu này, mỗi từ bạn vừa đọc sẽ đóng góp 1 phần thông tin tạo nên ý nghĩa của cả câu. Bộ não của bạn lưu giữ những thông tin của những từ bạn vừa đọc và sử dụng chúng để xử lý ngữ nghĩa của những từ tiếp theo. Đây là một quá trình phức tạp mà những Neural Network bình thường không thể mô phỏng lại được. Chính vì vậy, Recurrent Neural Network ra đời để giải quyết vấn đề này.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc104587019)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc104587020)

[TÓM TẮT iv](#_Toc104587021)

[MỤC LỤC 1](#_Toc104587022)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 4](#_Toc104587023)

[CHƯƠNG 1 – LÝ THUYẾT VỀ RECURRENT NEURAL NETWORK 5](#_Toc104587024)

[1.1 Khái niệm về Neural Network 5](#_Toc104587025)

[1.2 Khái niệm về Recurrent Neural Network 6](#_Toc104587026)

[1.3 Cách thức hoạt động của RNN 7](#_Toc104587027)

[1.4 Các dạng RNN 8](#_Toc104587028)

[1.4.1 One to one 8](#_Toc104587029)

[1.4.2 One to many 9](#_Toc104587030)

[1.4.3 Many to one 10](#_Toc104587031)

[1.4.4 Many to many 10](#_Toc104587032)

[1.5 Ưu điểm và nhược điểm của mạng RNN 11](#_Toc104587033)

[CHƯƠNG 2 – LONG SHORT TERM MEMORY 12](#_Toc104587034)

[2.1 Khái niệm về Long Short Term Memory 12](#_Toc104587035)

[2.2 Các thức hoạt động LSTM 12](#_Toc104587036)

[2.2.1 Forget gate - Xác định lượng dữ liệu quá khứ cần bỏ bớt 12](#_Toc104587037)

[2.2.2 Input gate – Xác định đầu vào nên được sử dụng 13](#_Toc104587038)

[2.2.3 Ouput gate – Xác định đầu ra của mô hình 14](#_Toc104587039)

[2.4 Ví dụ về bài toán LSTM 14](#_Toc104587040)

[CHƯƠNG 3 – TRÌNH BÀY LÝ THUYẾT VỀ OVERFITTING VÀ REGULARIZATION 17](#_Toc104587041)

[3.1 Overfitting 17](#_Toc104587042)

[3.2 Những kỹ thuật tránh Overfitting 18](#_Toc104587043)

[3.2.1 Regularization 20](#_Toc104587044)

[3.2.2 Early Stopping 21](#_Toc104587045)

[3.2.3 Dropout 22](#_Toc104587046)

[3.2.4 Các phương pháp khác 24](#_Toc104587047)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC CHỮ VIẾT TẮT**

NN Neural Network

RNN Recurrent Neural Network

LSTM Long Short Term Memory

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1. 1 So sánh mạng RNN (bên trái) và mạng FNN (bên phải) 5](#_Toc104417861)

[Hình 1. 2 Cách thức hoạt động của RNN 6](#_Toc104417862)

[Hình 1. 3 Sơ đồ cho mạng RNN One to one 7](#_Toc104417863)

[Hình 1. 4 Sơ đồ cho mạng RNN One to many 8](#_Toc104417864)

[Hình 1. 5 Sơ đồ mạng RNN Many to one 9](#_Toc104417865)

[Hình 1. 6 Sơ đồ cho mạng RNN Many to many 9](#_Toc104417866)

[Hình 2. 1 Mô hình LSTM 11](#_Toc104409218)

[Hình 2. 2 Forget gate 12](#_Toc104409219)

[Hình 2. 3 Input gate 12](#_Toc104409220)

[Hình 2. 4 Output gate 13](#_Toc104409221)

[Hình 3. 1 y=sin(2πx) model. Underfit: degree 1 (left), Goodfit: degree 3 (center), Overfit: degree 15 (right) 17](#_Toc104417881)

[Hình 3. 2 Phương pháp dừng sớm (early stopping) 21](#_Toc104417882)

[Hình 3. 3 Phương pháp dropout 22](#_Toc104417883)

CHƯƠNG 1 – LÝ THUYẾT VỀ RECURRENT NEURAL NETWORK

1.1 Khái niệm về Neural Network

Neural Network (mạng nơ-ron nhân tạo) là tập hợp các thuật toán gần giống với não người và được thiết kế để nhận dạng các mẫu dữ liệu. NN sẽ phân tích dữ liệu thông qua nhận thức máy móc, ghi nhãn hoặc phân cụm đầu vào thô. Nó có thể nhận ra các mẫu số chứa trong các vectơ, trong đó có tất cả dữ liệu trong thế giới thực như hình ảnh, âm thanh, văn bản hoặc chuỗi thời gian. NN sẽ bao gồm một số lượng lớn các phần tử xử lý có tính kết nối cao (nơ-ron) làm việc cùng nhau để giải quyết một vấn đề.

Cấu tạo của một NN sẽ bao gồm một số lượng lớn các bộ xử lý hoạt động song song và được sắp xếp theo các tầng. Tầng đầu tiên nhận thông tin đầu vào thô, tương tự như các dây thần kinh thị giác trong quá trình xử lý thị giác của con người. Dữ liệu của mỗi tầng kế tiếp sẽ nhận được từ đầu ra từ tầng trước nó. Tầng cuối cùng tạo kết quả đầu ra của hệ thống.

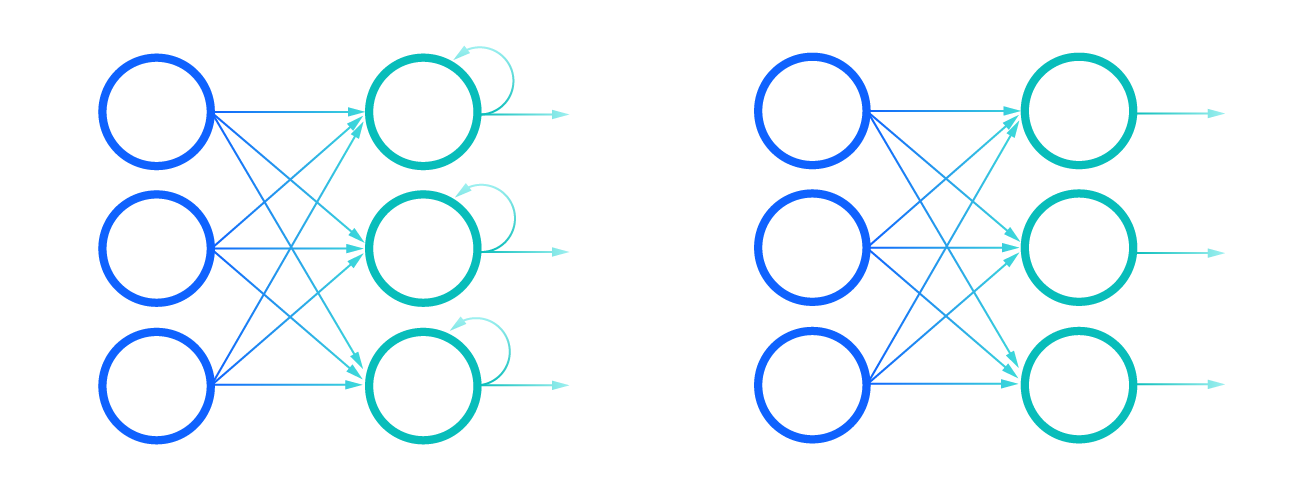
Các mô hình Neural Network phổ biến:

* **Feed-Forward Neural Network:** Được sử dụng cho các bài toán hồi quy và phân loại.
* **Convolutional Neural Network:** Được sử dụng để phát hiện đối tượng và phân loại hình ảnh.
* **Deep Belief Network:** Được sử dụng trong lĩnh vực chăm sóc sức khỏe để phát hiện ung thư.
* **Recurrent Neural Network:** Được sử dụng để nhận dạng giọng nói, nhận dạng giọng nói, dự đoán chuỗi thời gian và xử lý ngôn ngữ tự nhiên.

1.2 Khái niệm về Recurrent Neural Network

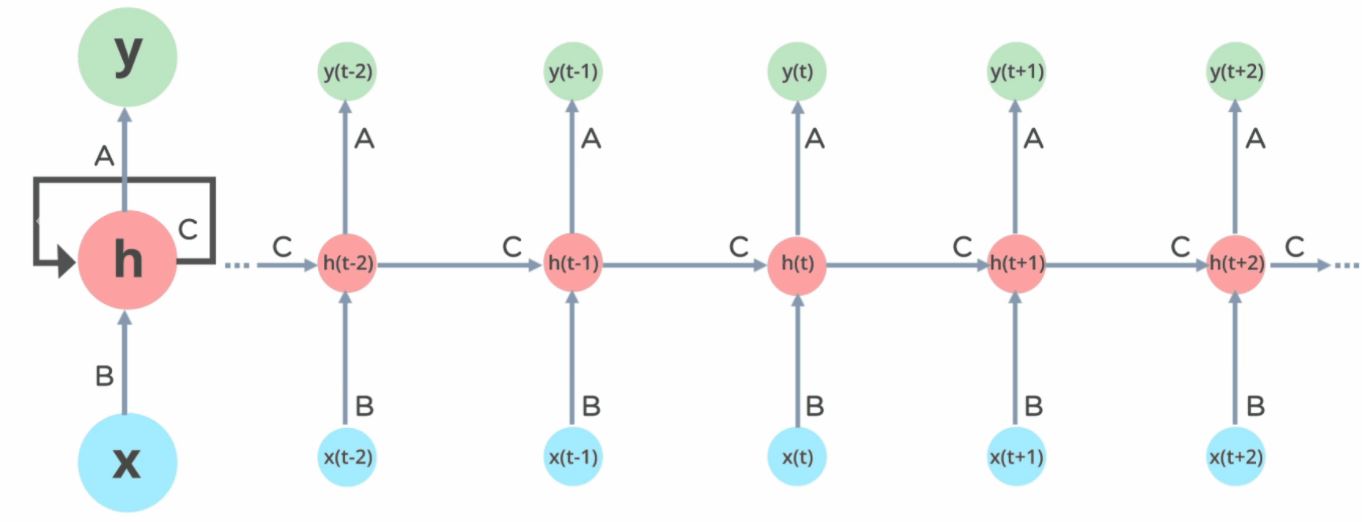
Recurrent Neural Network (mạng nơ-ron hồi quy) có thể như là mô hình tổng quát hơn của Feed-Forward Neural Network (mạng nơ-ron truyền thắng) khi sở hữu thêm một bộ nhớ trong. RNN có bản chất là lai đi lặp lại vì nó thực hiện cùng một chức năng cho mọi đầu vào của dữ liệu trong khi đầu ra của đầu vào hiện tại sẽ phụ thuộc vào lần tính toán trong của nhẫn lần trước đó. Sau khi có kết quả đầu ra, nó được sao chép và gửi lại vào mạng định kỳ. Để RNN đưa ra quyết định, nó xem xét đầu vào hiện tại và đầu ra mà nó đã học được từ đầu vào trước đó.

Không giống như mạng FNN, RNN có thể sử dụng trạng thái bên trong bộ nhớ để xử lý chuỗi đầu vào. Điều này làm cho RNN có thể áp dụng cho các tác vụ như nhận dạng chữ viết tay hoặc nhận dạng giọng nói. Trong các mạng nơ-ron khác, tất cả các đầu vào là độc lập với nhau. Nhưng trong RNN, tất cả các đầu vào đều liên quan đến nhau.



Hình 1.  So sánh mạng RNN (bên trái) và mạng FNN (bên phải)

1.3 Cách thức hoạt động của RNN



Hình 1.  Cách thức hoạt động của RNN

Với ***X*** là lớp đầu vào, ***h*** là lớp ẩn và ***y*** là lớp đầu ra. ***A***, ***B*** và ***C*** là các tham số được sử dụng để cải thiện đầu ra của RNN. Tại một thời điểm t bất kỳ, đầu vào hiện tại là sự kết hợp của đầu vào tại ***x(t)*** và ***x(t-1)***. Kết quả đầu ra tại bất kỳ thời điểm nào sẽ được nạp trở lại mạng RNN để cải thiện kết quả đầu ra.

Công thức tính trạng thái hiện tại:

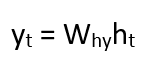


Áp dụng hàm kích hoạt:



Với W là trọng số, h là vectơ ẩn đơn lẻ, Whh là trọng số của trạng thái ẩn trước đó, Whx là trọng số của trạng thái đầu vào hiện tại, tanh là hàm kích hoạt, thực hiện tính không tuyến tính làm giảm kích hoạt đến phạm vi [-1,1].

Đầu ra của mô hình:



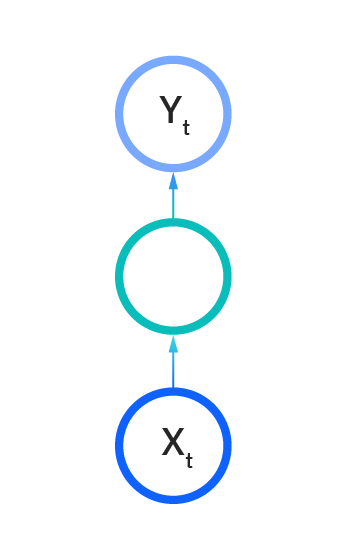
Với Yt là đầu ra.Why là trọng số của trạng thái đầu ra.

1.4 Các dạng RNN

Mạng RNN có 4 dạng chính:

1.4.1 One to one

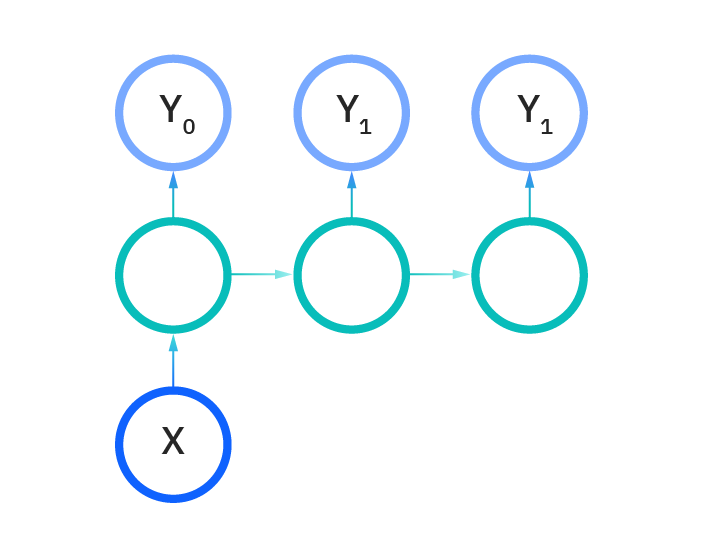
Dạng RNN này còn được biết đến với tên gọi Vanila Neural Network. Mạng RNN dạng One to one thường được sử dụng cho hầu hết các bài toán học máy hiện nay, với một đầu vào và một đầu ra duy nhất.



Hình 1.  Sơ đồ cho mạng RNN One to one

1.4.2 One to many

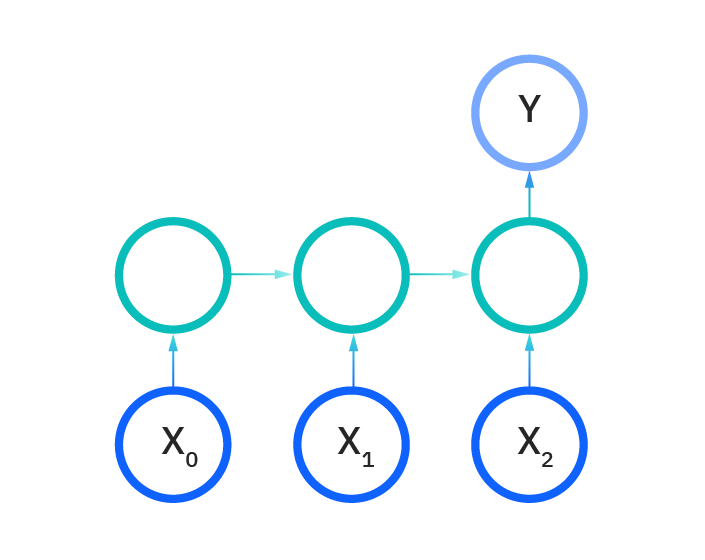
Dạng mạng RNN này chỉ có một đầu vào duy nhất và nhiều đầu ra. Mạng RNN One to many thường được dùng cho để chú thích hình ảnh.



Hình 1.  Sơ đồ cho mạng RNN One to many

1.4.3 Many to one

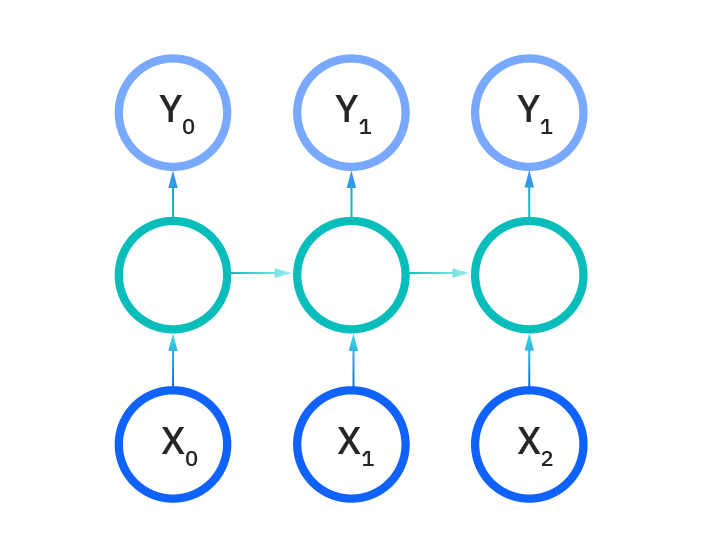
Mạng RNN Many to one sẽ nhận một chuỗi các đầu vào và tạo ra một đầu ra duy nhất. Phân tích cảm xúc trong câu nói là một ví dụ điển hình của loại mạng này. Với một câu nhất định, ta có thể được phân loại xem nó thể hiện tình cảm tích cực hay tiêu cực.



Hình 1.  Sơ đồ mạng RNN Many to one

1.4.4 Many to many

Mạng RNN Many to many sẽ nhận một chuỗi các đầu vào và tạo một chuỗi các đầu ra. Dịch tự động là một trong những ví dụ về dạng mạng này.



Hình 1.  Sơ đồ cho mạng RNN Many to many

1.5 Ưu điểm và nhược điểm của mạng RNN

Ưu điểm của RNN:

* RNN có thể lập mô hình chuỗi dữ liệu để mỗi mẫu có thể được giả định là phụ thuộc vào các mẫu trước đó.
* RNN được sử dụng với các lớp phức hợp để mở rộng vùng lân cận pixel hiệu quả.

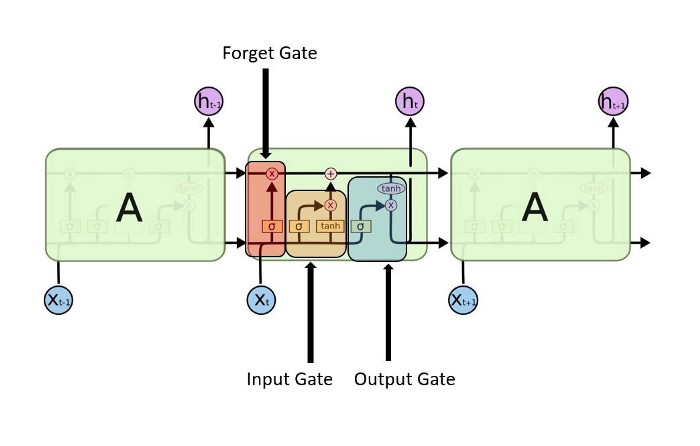
Nhược điểm của RNN:

* Lỗi Gradient Vanishing.
* Lỗi Gradient Exploding
* Training một RNN rất khó khăn.
* RNN không thể xử lý các chuỗi rất dài nếu sử dụng hàm tanh hoặc relu làm hàm kích hoạt.

CHƯƠNG 2 – LONG SHORT TERM MEMORY

2.1 Khái niệm về Long Short Term Memory

Long Short Term Memory (LSTM) là một phiên bản cải tiến của RNN. LSTM giúp cho mô hình dễ dàng ghi nhớ dữ liệu quá khứ trong bộ nhớ. Với LSTM, các vấn đề về gradient của RNN được nêu ở trên sẽ được giải quyết. Mô hình LSTM rất phù hợp để phân loại, xử lý và dự đoán chuỗi thời gian cho trước các độ trễ thời gian không xác định. Nó training mô hình bằng cách sử dụng truyền ngược back-propagation.



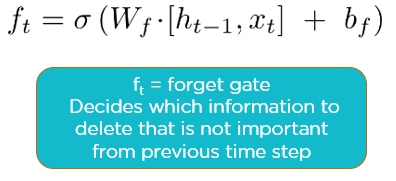
Hình 2.  Mô hình LSTM

2.2 Các thức hoạt động LSTM

Mô hình LSTM hoạt động theo quy trình 3 bước:

2.2.1 Forget gate - Xác định lượng dữ liệu quá khứ cần bỏ bớt

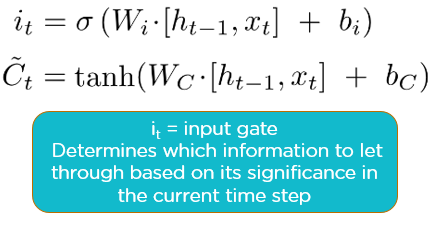
Bước đầu tiên trong LSTM quyết định thông tin nào nên được bỏ qua trong khoảng thời gian cụ thể đó. Để thực hiện điều này, LSTM sẽ sử dụng hàm Sigmoid. Nó sẽ xem xét trạng thái trước đó ***(ht-1)*** cùng với ***xt*** đầu vào hiện tại và thực tính toán.



Hình 2.  Forget gate

2.2.2 Input gate – Xác định đầu vào nên được sử dụng

Bước này gồm 2 bước nhỏ. Đầu tiên, hàm Sigmoid sẽ quyết định giá trị nào sẽ được cho qua (0 hoặc 1). Tiếp theo, hàm Tanh sẽ gán trọng số quan trọng cho các giá trị được cho qua, quyết định mức độ quan trọng của chúng (-1 đến 1).

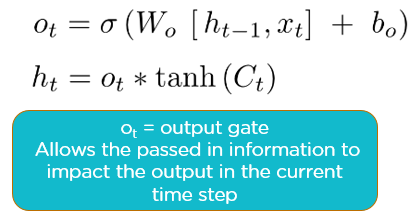


Hình 2.  Input gate

Giá trị đầu vào tại bước này ***x(t)***, Input gate sẽ phân tích những thông tin quan trọng dựa vào 2 bước được nêu trên.

2.2.3 Ouput gate – Xác định đầu ra của mô hình

Bước này là quyết định đầu ra của mô hình sẽ là gì. Đầu tiên, chúng ta tiếp tục dùng hàm Sigmoid để quyết định giá trị nào sẽ được cho qua (0 hoặc 1). Nếu được cho qua, giá trị đó sẽ tiếp tục được hàm Tanh gán trọng số quan trọng (-1 đến 1), sau đó nhân nó với giá trị đầu ra của hàm Sigmoid.



Hình 2.  Output gate

2.4 Ví dụ về bài toán LSTM

Chúng ta sẽ tiến hành triển khai một mô hình LSTM để dự đoán giá cổ phiếu của Google. Dựa trên dữ liệu từ năm 2012 đến 2016, chúng ta sẽ dự đoán giá cổ phiếu của năm 2017.

Đầu tiên, chúng ta sẽ import những thư viện cần thiết:



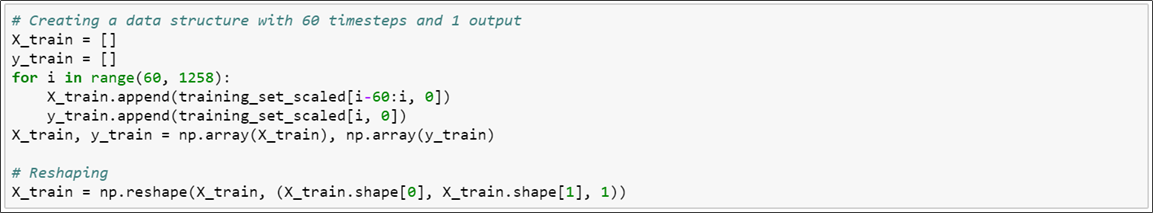
Tiếp theo, chúng ta import trainng dataset:



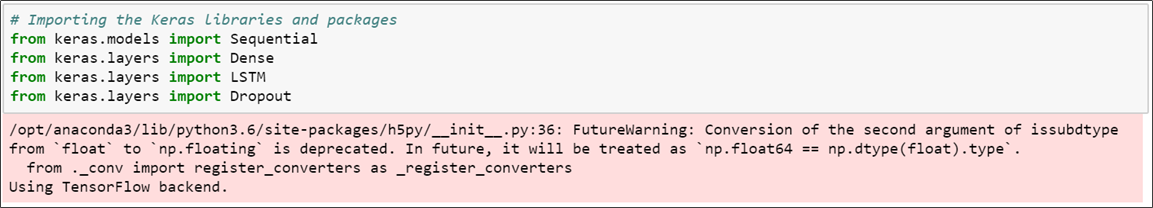
Chuẩn hóa dữ liệu:



Tạo cấu trúc cho dữ liệu với 60 time steps và 1 đầu ra:



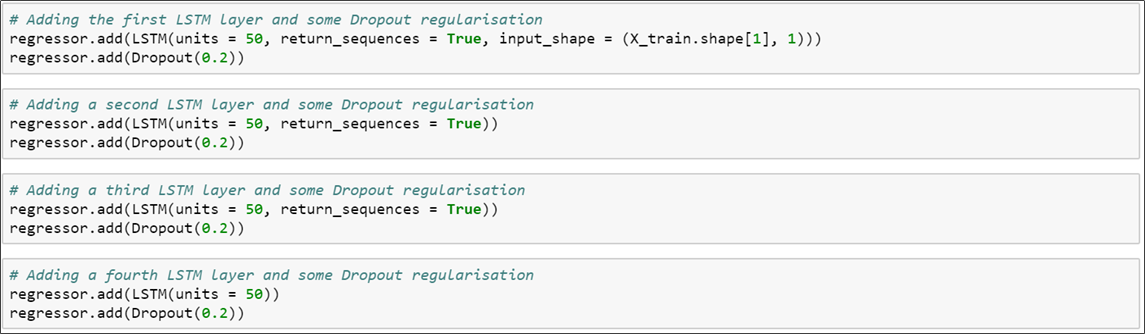
Import thư viện Keras:



Khởi tạo mô hình RNN:



Thêm các lớp LSTM và Dropout:



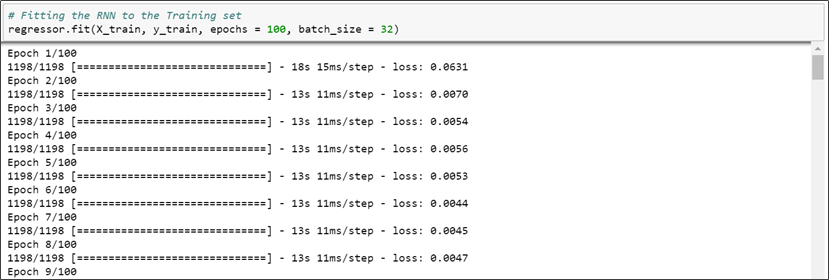
Khởi tạo đầu ra output:



Khởi tạo trình biên dịch mô hình RNN:



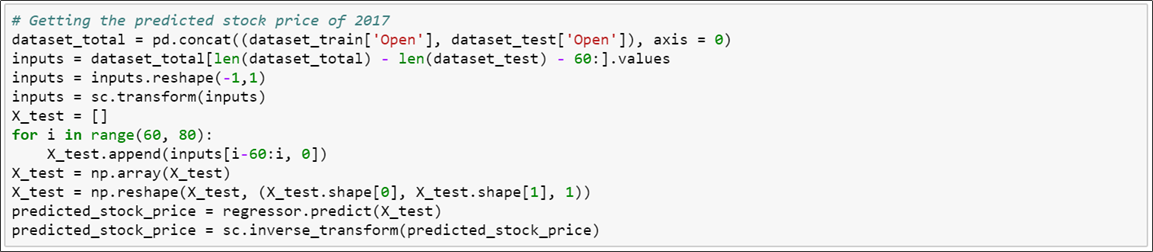
Training:



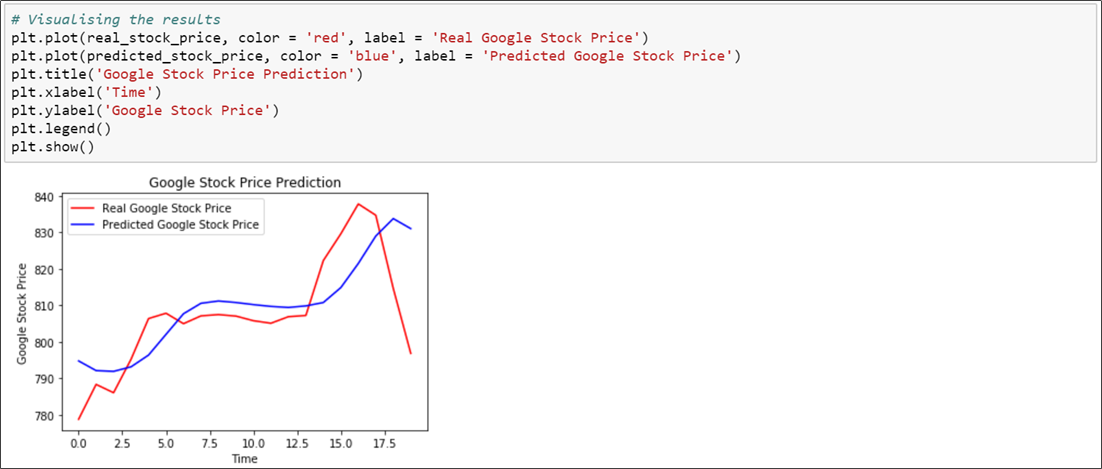
Tải dữ liệu cổ phiếu của năm 2017:



Dự doán giá cổ phiếu cho năm 2017:



Sử dụng sơ đồ so sánh kết quả dự đoán và kết quả thực tế:



CHƯƠNG 3 – TRÌNH BÀY LÝ THUYẾT VỀ OVERFITTING VÀ REGULARIZATION

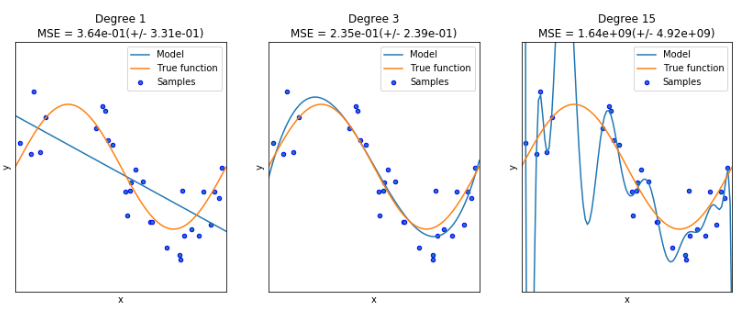
3.1 Overfitting

Overfitting là hiện tượng khi mô hình xây dựng thể hiện được chi tiết bộ dữ liệu huấn luyện. Điều này có nghĩa là cả dữ liệu nhiễu hoặc dữ liệu bất thường trong tập huấn luyện đều được chọn và học để đưa ra quy luật mô hình. Những quy luật này sẽ không có ý nghĩa nhiều khi áp dụng với bộ dữ liệu mới có thể có dạng dữ liệu nhiễu khác. Khi đó, nó ảnh hưởng tiêu cực tới độ chính xác của mô hình nói chung.

Hiện tượng Overfitting thường xảy ra trong các mô hình phi tham số hoặc phi tuyến, những mô hình có sự linh hoạt cao trong xây dựng hàm mục tiêu.

Như vậy, rất nhiều thuật toán học máy phi tham số sẽ bao gồm những thông số và kĩ thuật để hạn chế và giới hạn mức độ học chi tiết của mô hình.

Ví dụ, bài toán cây quyết định là một thuật toán học máy phi tham số. Đây là thuật toán thường xảy ra hiện tượng Overfitting. Ta có thể tránh hiện tượng này bằng phương pháp cắt tỉa cây (pruning).



Hình 3. y=sin(2πx) model. Underfit: degree 1 (left), Goodfit: degree 3 (center), Overfit: degree 15 (right)

Ví dụ như ở hình phía bên phải ở trên. Mô hình của ta gần như mong muốn bao được hết tất cả các điểm làm cho biên độ dao động của nó lớn quá mức. Mô hình này mà dự đoán với 1 giá trị mới của x thì khả năng y sẽ bị lệch đi rất nhiều.

3.2 Những kỹ thuật tránh Overfitting

* Tăng số lượng Training Data
* Giảm số lượng biến giải thích (feature)
* Tăng độ lớn của parameter chuẩn hóa λ

Đầu tiên cần một vài đại lượng để đánh giá chất lượng của mô hình trên *training data* và *test data*. Giả sử là đầu ra thực sự (có thể là vector), và là đầu ra dự đoán bởi mô hình:

*với p thường bằng 1 hoặc 2*

Với Classification, trung bình cộng của cross entropy có thể được sử dụng.

**Test Error**: Giống như trên nhưng áp dụng mô hình tìm được vào *test data*. Khi xây dựng mô hình, ta không được sử dụng thông tin trong tập dữ liệu test. Dữ liệu test chỉ được dùng để đánh giá mô hình. Với Regression, đại lượng này thường được định nghĩa:

*với p giống như p trong cách tính train error phía trên*

Việc lấy trung bình là quan trọng vì lượng dữ liệu trong hai tập hợp training và test có thể chênh lệch rất nhiều.

Một mô hình được coi là tốt (fit) nếu cả train error và test error đều thấp. Nếu train error thấp nhưng test error cao, ta nói mô hình bị overfitting. Nếu train error cao và test error cao, ta nói mô hình bị underfitting. Nếu train error cao nhưng test error thấp, tôi không biết tên của mô hình này, vì cực kỳ may mắn thì hiện tượng này mới xảy ra, hoặc có chỉ khi tập dữ liệu test quá nhỏ.

Cả hai hiện tượng Overfitting và Underfitting đều khiến mô hình xây dựng có độ chính xác kém. Nhưng hiện nay, vấn đề phổ biến nhất xuất hiện là Overfitting.

Overfitting thực sự là một vấn đề quan trọng bởi vì việc đánh giá mô hình học máy trên bộ dữ liệu huấn luyện sẽ khác biệt với việc đánh giá độ chính xác của tổng thể ( những dữ liệu mà mô hình chưa gặp bao giờ).

*Có hai kỹ thuật quan trọng trong việc đánh giá mô hình học máy và tránh hiện tượng overfitting:*

* Sử dụng kỹ thuật lấy lại mẫu để ước lượng độ chính xác của mô hình
* Sử dụng tập Validation test

**Lấy lại mẫu** (Resampling Methods) là kỹ thuật phổ biến hơn. Khi đó, ta sẽ chia tập dữ liệu thành k tập con. Cách này được gọi là **k-fold cross validation**. Điều này cho phép bạn thực hiện huấn luyện trên các tập dữ liệu khác nhau k lần, và từ đó, xây dựng ước lượng độ chính xác của mô hình học máy với dữ liệu mới.

Sử dụng **Cross-validation** là một tiêu chuẩn tốt trong học máy để ước lượng độ chính xác của mô hình với bộ dữ liệu mới. Còn trường hợp ta có nhiều dữ liệu, việc sử dụng tập Validation sẽ là một phương pháp tuyệt vời.

3.2.1 Regularization

Regularization, một cách cơ bản, là thay đổi mô hình một chút để tránh overfitting trong khi vẫn giữ được tính tổng quát của nó (tính tổng quát là tính mô tả được nhiều dữ liệu, trong cả tập training và test). Một cách cụ thể hơn, ta sẽ tìm cách di chuyển nghiệm của bài toán tối ưu hàm mất mát tới một điểm gần nó. Hướng di chuyển sẽ là hướng làm cho mô hình ít phức tạp hơn mặc dù giá trị của hàm mất mát có tăng lên một chút.

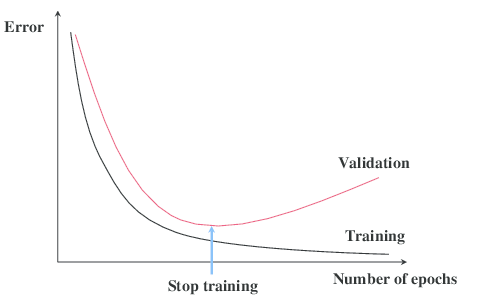
L1 có dạng:

L2 có dạng:

3.2.2 Early Stopping

Khi huấn luyện càng lâu thì giá trị hàm mất mát của mô hình trên tập huấn luyện càng nhỏ và mô hình có xu hướng khớp tốt dữ liệu trên tập huấn luyện hơn. Mặc dù sai số trên tập huấn luyện có xu hướng giảm dần theo thời gian nhưng trên tập kiểm định (validation data) điều này sẽ chưa chắc là đúng. Lý do là vì thời điểm mô hình đạt tới một độ phức tạp nhất định nó sẽ không còn khái quát hoá tốt dữ liệu kiểm định. Như vậy trên tập kiểm định tới một giai đoạn epoch nào đó sai số sẽ tăng lên.

Phương pháp dừng sớm sẽ xác định đâu là epoch được lựa chọn làm điểm dừng phù hợp căn cứ vào sai số trên tập kiểm định. Đó là thời điểm mà sai số trên tập kiểm định bắt đầu có xu hướng tăng lên. Khi đó một quyết định dừng sớm quá trình huấn luyện sẽ giúp tránh hiện tượng quá khớp.

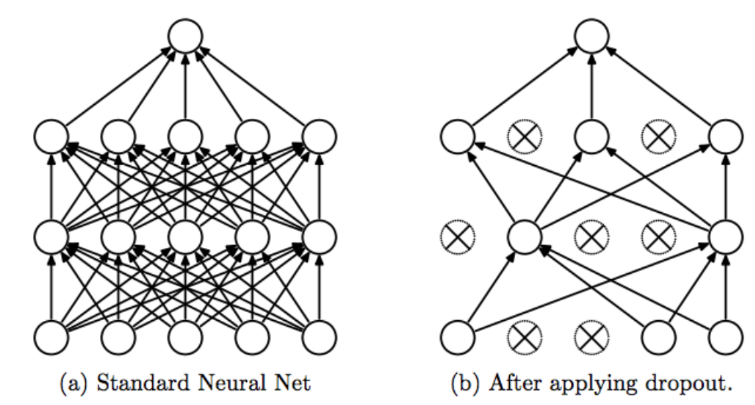


Hình 3. Phương pháp dừng sớm (early stopping)

Phương pháp dừng sớm (early stopping) được thể hiện qua sai số trên tập kiểm định bắt đầu tăng lên tại một epoch nào đó, chẳng hạn như điểm dừng (stop training) trong hình. Theo qui luật tổng quát, trong dài hạn sai số trên tập huấn luyện có xu hướng giảm dần theo epoch bởi quá trình huấn luyện luôn tìm cách tối thiểu hàm mất mát trên tập huấn luyện.

Phương pháp dừng sớm thường được áp dụng trong quá trình huấn luyện các mô hình deep learning không chỉ giảm thiểu quá khớp mà còn tiết kiệm chi phí huấn luyện. Để tìm được vị trí dừng phù hợp chúng ta sẽ kiểm tra mức độ gia tăng của sai số trên tập kiểm tra. Điều kiện dừng được thiết lập là ngưỡng gia tăng hàm mất mát của epoch sau so với epoch trước lớn hơn . Trong quá trình huấn luyện cũng cần liên tục lưu lại các checkpoint cho mô hình sau mỗi epoch cho tới khi đạt được điểm dừng.

3.2.3 Dropout



Hình 3. Phương pháp dropout

Phương pháp dropout được áp dụng trên mạng nơ ron. Mạng nơ ron sẽ bao gồm nhiều layers mỗi layer gồm nhiều units. Các trọng số mô hình được thể hiện bởi một mũi tên liên kết units giữa các layers. Hình bên phải là toàn bộ mạng được sử dụng trong quá trình suy luận. Trong khi hình bên trái là áp dụng dropout trên các layers của mạng và chỉ được áp dụng trong quá trình huấn luyện. Trọng số mô hình bị loại bỏ được thể hiện qua những kết nối mũi tên bị xoá bỏ đi. Mô hình mới được tạo thành để huấn luyện sẽ trở nên thưa hơn, đồng thời mức độ phức tạp giảm nhằm giúp giảm quá khớp. Ngoài ra việc lựa chọn tham số để loại bỏ là ngẫu nhiên nên kiến trúc mô hình dropout được biến đổi rất đa dạng. Như vậy mô hình sau cùng thu được về bản chất là một sự kết hợp (ensembling) của các mô hình dự báo.

Mỗĩ một lượt huấn luyện chúng ta loại bỏ một tỷ lệ thiểu số các trọng số mô hình thì sẽ tạo ra một mô hình mới ít phức tạp hơn. Như vậy mô hình được huấn luyện theo kỹ thuật này sẽ là kết hợp của rất nhiều các mô hình biến thể sau dropout và chúng ta có thể xem chúng như là một phương pháp kết hợp mô hình (ensemble model) nhằm giảm quá khớp.

3.2.4 Các phương pháp khác

* regularization
* Tikhonov regularization
* Regularizers for sparsity
* Regularization trong sklearn

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. Tài liệu môn học Nhập Môn Học Máy – Trường Đại học Tôn Đức Thắng
2. Recurrent neural network – Nttuan8

**Tiếng Anh**

1. Aditi Mittal - *Understanding RNN and LSTM*
2. Avijeet Biswal - *Recurrent Neural Network (RNN) Tutorial for Beginners* – Simplilearn
3. Srivastava, Nitish, et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting” Journal of Machine Learning Research 15.1 (2014): 1929-1958.
4. "Overfitting - Wikipedia". En.Wikipedia.Org, 2017