

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN
MACHINE LEARNING**

DỰ ĐOÁN GIÁ CỔ PHIẾU

Người hướng dẫn: **TS LÊ ANH CƯỜNG**

Người thực hiện: **NGUYỄN NGỌC THIÊN - 51900711**

TĂNG KIẾN TRUNG - 51900718

TRẦN BẢO KHA - 51900751

Lớp : 19050201

Khoá : 23

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ MÔN
MACHINE LEARNING**

DỰ ĐOÁN GIÁ CỔ PHIẾU

Người hướng dẫn: **TS LÊ ANH CƯỜNG**

Người thực hiện: **NGUYỄN NGỌC THIÊN - 51900711**

TĂNG KIẾN TRUNG - 51900718

TRẦN BẢO KHA - 51900751

Lớp : 19050201

Khoá : 23

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

LỜI CẢM ƠN

Chúng em xin chân thành gửi lời cảm ơn này đến Thầy Lê Anh Cường giảng viên phụ trách giảng dạy bộ môn Học máy. Nhờ có sự tận tình giảng dạy, truyền đạt kiến thức của quý thầy mà chúng em mới đủ kiến thức để hoàn thành đồ án cuối kỳ này.

Song song với đó, chúng em cũng xin gửi lời cảm ơn đến Khoa Công Nghệ Thông Tin, trường Đại học Tôn Đức Thắng vì đã tạo điều kiện cho chúng em học tập, nghiên cứu trong suốt quá trình học tập môn học này nói riêng và cả quá trình học tại môi trường Đại học nói chung. Một lần nữa chúng em xin gửi lời cảm ơn chân thành đến mọi người và chúc tất cả thật nhiều sức khỏe.

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là công trình nghiên cứu của riêng chúng tôi và được sự hướng dẫn khoa học của 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 luận văn 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 chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung luận văn 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 2022

Tác giả

(ký tên và ghi rõ họ tên)

Nguyễn Ngọc Thiện

Tăng Kiến Trung

Trần Bảo Kha

TÓM TẮT

MỤC LỤC

LỜI CẢM ƠN	iii
TÓM TẮT	v
MỤC LỤC.....	1
DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT	4
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ	5
CHƯƠNG 1 – LÝ THUYẾT CƠ BẢN	8
1.1 K-Nearest Neighbor	Error! Bookmark not defined.
1.1.1 Khái niệm:.....	Error! Bookmark not defined.
1.1.2 Cơ sở lý thuyết:	Error! Bookmark not defined.
1.2 Decision Tree	Error! Bookmark not defined.
1.2.1 Khái niệm:.....	Error! Bookmark not defined.
1.2.2 Cơ sở lý thuyết:	Error! Bookmark not defined.
1.2.3 Overfitting.....	Error! Bookmark not defined.
1.3 Random Forest	Error! Bookmark not defined.
1.3.1 Khái niệm:.....	Error! Bookmark not defined.
1.3.2 Cơ sở lý thuyết:	Error! Bookmark not defined.
1.3.3 Mã giả cho Raindom Forest.....	Error! Bookmark not defined.
CHƯƠNG 2 – CÁC PHƯƠNG PHÁP ĐÁNH GIÁ MÔ HÌNH	Error! Bookmark not defined.
2.1 Accuracy:	Error! Bookmark not defined.
2.1.1 Khái niệm.....	Error! Bookmark not defined.
2.1.2 Cách tính:	Error! Bookmark not defined.
2.2 Loss:	Error! Bookmark not defined.
2.2.1 Khái niệm:.....	Error! Bookmark not defined.
2.2.2 Loss vs Accuracy:	Error! Bookmark not defined.
2.3 Confusion matrix:	Error! Bookmark not defined.

2.3.1 Khái niệm:.....	Error! Bookmark not defined.
2.4 Precision vs Recall:	Error! Bookmark not defined.
2.4.1 Precision:.....	Error! Bookmark not defined.
2.4.2 Recall:	Error! Bookmark not defined.
2.5 F-1 Score	Error! Bookmark not defined.
2.5.1 Khái niệm:.....	Error! Bookmark not defined.
2.5.2 Công thức:.....	Error! Bookmark not defined.
2.6 ROC CURVE.....	Error! Bookmark not defined.
2.7 Tổng kết:	Error! Bookmark not defined.
CHƯƠNG 3 - ÁP DỤNG THỰC TIỄN CHO CLASSIFICATION ..	
Error! Bookmark not defined.	
3.1 Bài toán	Error! Bookmark not defined.
3.2 Quá trình xây dựng mô hình:	Error! Bookmark not defined.
3.2.1 Đọc dữ liệu:.....	Error! Bookmark not defined.
3.2.2 Phân tích, đánh giá dữ liệu.....	Error! Bookmark not defined.
3.2.3 Làm sạch dữ liệu	Error! Bookmark not defined.
3.2.4 Tiền xử lý dữ liệu (Data preprocessing)	Error! Bookmark not defined.
defined.	
3.2.5 Training	Error! Bookmark not defined.
CHƯƠNG 4 – FEATURE ENGINEERING	
Error! Bookmark not defined.	
4.1 Giới thiệu về Feature Engineering	Error! Bookmark not defined.
4.2 Lựa chọn đặc trưng (Feature Selection) là gì?	Error! Bookmark not defined.
defined.	
4.3 Độ tương quan (Correlation).....	Error! Bookmark not defined.
4.3.1 Phương pháp lọc	Error! Bookmark not defined.
4.3.2 Correlation Pearson – Tương quan Pearson ..	Error! Bookmark not defined.
defined.	

4.3.3 Chênh lệch tuyệt đối trung bình (MAD).....	Error! Bookmark not defined.
CHƯƠNG 5 – CÁC THUẬT TOÁN TỐI ƯU.....	Error! Bookmark not defined.
5.1 Gradient Descent.....	Error! Bookmark not defined.
5.1.1 Gradient Descent cho hàm 1 biến: ..	Error! Bookmark not defined.
5.1.2 Gradient Descent cho hàm nhiều biến:	Error! Bookmark not defined.
5.1.3 Các thuật toán tối ưu Gradient Descent:	Error! Bookmark not defined.
5.1.3.1 Momentum:.....	Error! Bookmark not defined.
5.1.3.1 Nesterov accelerated gradient (NAG): ..	Error! Bookmark not defined.
5.2 Adam (Adam Optimization Algorithm).....	Error! Bookmark not defined.
5.2.1 Mô tả:	Error! Bookmark not defined.
5.2.2 Thuật toán Adam sinh ra từ đâu?	Error! Bookmark not defined.
5.2.3 Thuật toán tối ưu Adam:	Error! Bookmark not defined.
5.2.3.1 Bias Correction:	Error! Bookmark not defined.
5.2.3.2 AdaMax:	Error! Bookmark not defined.

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

CÁC CHỮ VIẾT TẮT

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

Hình 01. Công thức tính khoảng cách.....	Error! Bookmark not defined.
Hình 02. Sơ đồ tổng quát mô hình Decision Tree.....	Error! Bookmark not defined.
Hình 03. Công thức tính Entropy	Error! Bookmark not defined.
Hình 04. Công thức tính Information Gain.....	Error! Bookmark not defined.
Hình 05. Ví dụ về Decision Tree	Error! Bookmark not defined.
Hình 06. Ví dụ về Overfitting trong Decision Tree	Error! Bookmark not defined.
Hình 07. Ví dụ về cách fix Overfitting trong Decision Tree	Error! Bookmark not defined.
defined.	
Hình 08. Ví dụ về các tạo nhiều cây	Error! Bookmark not defined.
Hình 09. Ví dụ về cách vote cho cây	Error! Bookmark not defined.
Hình 10. Thuật toán Random Forest.....	Error! Bookmark not defined.
Hình 11. Cách tính Accuracy	Error! Bookmark not defined.
Hình 12. Công thức tổng quát cho Accuracy	Error! Bookmark not defined.
Hình 13. Ví dụ về Loss	Error! Bookmark not defined.
Hình 13. So sánh loss giữa 2 model.....	Error! Bookmark not defined.
Hình 15. Ví dụ về dự đoán ung thư.....	Error! Bookmark not defined.
Hình 16. Bảng phân bố dự đoán.....	Error! Bookmark not defined.
Hình 17. Mối tương quan giữa Precision vs Recall	Error! Bookmark not defined.
Hình 18. Công thức tính Precision.....	Error! Bookmark not defined.
Hình 19. Ví dụ tính Precision.....	Error! Bookmark not defined.
Hình 20. Công thức tính Recall.....	Error! Bookmark not defined.
Hình 21. Ví dụ tính Recall	Error! Bookmark not defined.
Hình 22. Tổng quát giữa Precision vs Recall.....	Error! Bookmark not defined.
Hình 22. Công thức tính F1-score.....	Error! Bookmark not defined.
Hình 23. Ví dụ tính F1-score.....	Error! Bookmark not defined.

- Hình 24. Công thức tính TPR vs FPR.....**Error! Bookmark not defined.**
- Hình 25. Sơ đồ biểu thị sự thay đổi TPR và FPR**Error! Bookmark not defined.**
- Hình 26. Import các thư viện cần thiết và đọc dataset dưới dạng file csv. **Error! Bookmark not defined.**
- Hình 27. 5 dòng dữ liệu đầu tiên trong dataset**Error! Bookmark not defined.**
- Hình 28. Tổng số các dòng dữ liệu chứa giá trị null theo cột.....**Error! Bookmark not defined.**
- Hình 29. Mô tả về các đặc trưng của tập dữ liệu.**Error! Bookmark not defined.**
- Hình 30. Biểu đồ biểu thị tương quan giữa giới tính, tuổi với chỉ số BMI..... **Error! Bookmark not defined.**
- Hình 31. Tạo cột age_group dựa trên độ tuổi**Error! Bookmark not defined.**
- Hình 32. Biểu đồ box giữa đặc trưng age_group và chỉ số bmi ...**Error! Bookmark not defined.**
- Hình 33. Tính toán sinh chỉ số bmi dựa trên độ tuổi và giới tính dựa trên giá trị trung bình (mean)**Error! Bookmark not defined.**
- Hình 34. Kiểm tra lại các dòng dữ liệu chứa giá trị null và kiểm tra có trùng lặp dòng dữ liệu nào không.....**Error! Bookmark not defined.**
- Hình 35. Biểu đồ cột so sánh tỉ lệ giữa người bị đột quỵ và không bị đột quỵ trong dataset.....**Error! Bookmark not defined.**
- Hình 36. Biểu đồ tròn so sánh tỉ lệ giữa người bị đột quỵ và không bị đột quỵ trong dataset.....**Error! Bookmark not defined.**
- Hình 37. Bỏ cột id và age_group ra khỏi dataframe.**Error! Bookmark not defined.**
- Hình 38. Đếm thống kê giá trị của từng cột.**Error! Bookmark not defined.**
- Hình 39. Xóa dòng dữ liệu có giới tính là Other**Error! Bookmark not defined.**
- Hình 40. Chuẩn hóa dữ liệu về dạng 0 và 1**Error! Bookmark not defined.**
- Hình 41. Tập dữ liệu sau khi chuẩn hóa.....**Error! Bookmark not defined.**
- Hình 42. Tách tập train và tập test đồng thời chuẩn hóa**Error! Bookmark not defined.**

- Hình 43. Import các thư viện model cần thiết và tiến hành train..**Error! Bookmark not defined.**
- Hình 44. Tính các thông số để so sánh giữa các mô hình.....**Error! Bookmark not defined.**
- Hình 45. Kết quả so sánh giữa các mô hình.....**Error! Bookmark not defined.**
- Hình 46. Ma trận tương quan biểu diễn dưới dạng heatmap**Error! Bookmark not defined.**
- Hình 47. Ma trận tương quan của một tập dữ liệu.**Error! Bookmark not defined.**
- Hình 48. Độ tương quan giữa các đặc trưng**Error! Bookmark not defined.**
- Hình 49. Biểu diễn MAD dưới dạng biểu đồ cột.....**Error! Bookmark not defined.**
- Hình 50. Phương trình và đồ thị hàm 1 biến.....**Error! Bookmark not defined.**
- Hình 51. Code minh họa**Error! Bookmark not defined.**
- Hình 52. Code minh họa**Error! Bookmark not defined.**
- Hình 53. Minh họa vấn đề Gradient Descent.....**Error! Bookmark not defined.**
- Hình 54. Minh họa thể hiện sự khác nhau giữa thuật toán GD và thuật toán GD với Momentum**Error! Bookmark not defined.**
- Hình 55. Đồ thị NAG**Error! Bookmark not defined.**
- Hình 56. Vector giữa momentum và gradient.....**Error! Bookmark not defined.**

CHƯƠNG 1 – Dataset

Giá cả chứng khoán luôn liên tục biến động, không ngừng thay đổi do chịu ảnh hưởng của nhiều yếu tố cả vi mô lẫn vĩ mô, như chính trị, chiến tranh, kinh tế, tình hình tài chính công ty,,,. Đồng nghĩa với việc có rất nhiều dữ liệu để dự đoán được giá chứng khoán. Trong đề tài này chúng em sẽ sử dụng kiến thức học máy đã học để dự đoán giá cổ phiếu.

Bộ dữ liệu bao gồm 30 mã cổ phiếu chứa thông tin về giá cả (open, high, low, close) và khối lượng giao dịch chứng khoán từ ngày 04/01/2021 đến ngày 29/10/2021 được lấy từ package [vnquant](#).

Bài toán gồm các dataset:

- price_train.csv: Giá cả và khối lượng giao dịch chứng khoán, dùng để training mô hình
- price_test.csv: Tập dữ liệu test dùng để đánh giá kết quả mô hình trên Kaggle.
- business_train.csv: Báo cáo kinh doanh.
- finance_train.csv: Báo cáo tài chính.
- sample_submission.csv: Tập mẫu kết quả nộp lên Kaggle (30% dữ liệu Public và 70% dữ liệu Private trên Leaderboard Score)
- Trong đó mỗi dataset có các thuộc tính sau:
 - price_train:
 - date: thời gian giao dịch
 - open: giá mở cửa (giá thực hiện tại lần khớp lệnh đầu tiên trong ngày giao dịch chứng khoán)
 - high: giá cao nhất trong ngày
 - low: giá thấp nhất trong ngày

- close: giá đóng cửa (giá thị trường của các cổ phiếu vào thời điểm đóng cửa một phiên giao dịch) (target_variable)
- volume: khối lượng cổ phiếu giao dịch trong ngày (target variable)
- symbol: mã cổ phiếu
- business_train:
- index: chỉ số kết quả hoạt động kinh doanh, các cột còn lại dạng YYYY-MM: tháng báo cáo tài chính, thường là 3, 6, 9, 12 hàng năm.
- finance_train:
- index: chỉ số tài chính
- các cột còn lại dạng YYYY-MM: tháng báo cáo tài chính, thường là 3, 6, 9, 12 hàng năm.
- price_test: Đây là dữ liệu chứa các ngày và mã chứng khoán tương ứng mà chúng ta cần phải dự báo giá close.

Lưu ý sau khi dự báo trên template của price_test thì phải chuyển sang template sample_submission để nộp bài.

- Các trường bao gồm:
- date: ngày dự báo yyyy-mm-dd
- symbol: mã chứng khoán
- close: giá đóng cửa, đây là trường mục tiêu cần dự báo.
- sample_submission: Template được biến đổi từ price_test, dùng để submit kết quả.
- Id: Mã id của một quan sát dự báo, là kết hợp giữa date và symbol ngăn cách bởi dấu ":".
- Predicted: giá trị dự báo của giá close tương ứng với Id.

Chương 2: Tiền xử lý dữ liệu

Import thư viện

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import pandas as pd
```

Hình 2.1

Đọc dữ liệu

```
[3] df_business = pd.read_csv('business_train.csv', header = 0)
df_finance = pd.read_csv('finance_train.csv', header = 0)
df_price = pd.read_csv('price_train.csv', header = 0)
print('df_business.shape: ', df_business.shape, '; total symbols: ', len(df_business['symbol'].unique()))
print('df_finance.shape: ', df_finance.shape, '; total symbols: ', len(df_finance['symbol'].unique()))
print('df_price.shape: ', df_price.shape, '; total symbols: ', len(df_price['symbol'].unique()))
```

Hình 2.2

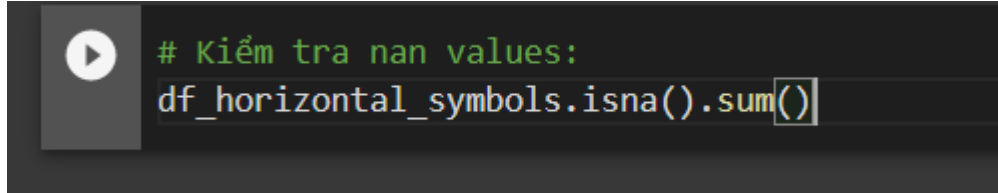
Để thuận tiện xử lý dữ liệu, chúng ta cần sắp xếp những mã chứng khoán này về dạng bảng có các cột là các mã chứng khoán và dòng là chuỗi thời gian.

```
df_horizontal_symbols = pd.pivot_table(df_price,
    index = 'date',
    columns = 'symbol',
    values = 'close',
    aggfunc = {
        'close': lambda x: x
    }
)

df_horizontal_symbols.head(5)
```

Hình 2.3

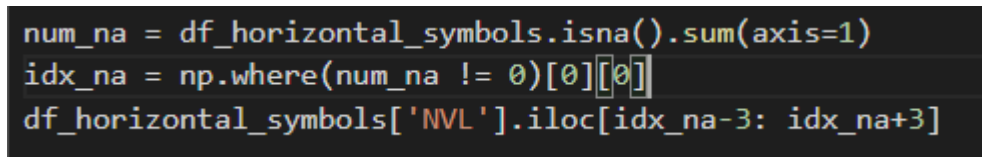
Kiểm tra nan values



```
# Kiểm tra nan values:  
df_horizontal_symbols.isna().sum()
```

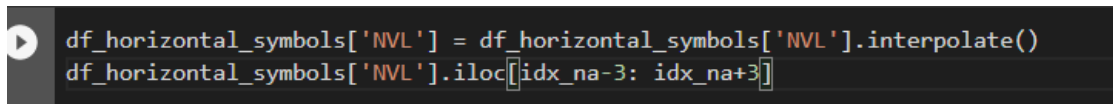
Hình 2.4

Do có cột NVL có một phần tử là NaN nên xử lý dữ liệu bằng phương pháp feed forward và sau đó sẽ fill dữ liệu bằng nội suy tuyến tính



```
num_na = df_horizontal_symbols.isna().sum(axis=1)  
idx_na = np.where(num_na != 0)[0][0]  
df_horizontal_symbols['NVL'].iloc[idx_na-3: idx_na+3]
```

Hình 2.5

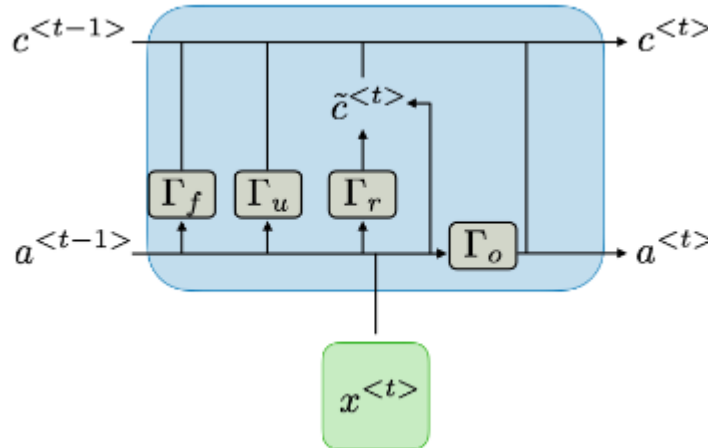


```
df_horizontal_symbols['NVL'] = df_horizontal_symbols['NVL'].interpolate()  
df_horizontal_symbols['NVL'].iloc[idx_na-3: idx_na+3]
```

Hình 2.6

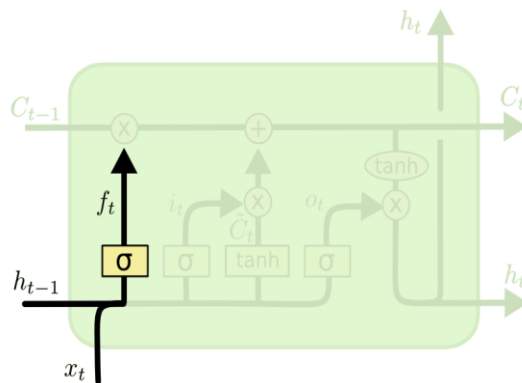
Chương 3: Mô hình học máy

3.1 LSTM



Mạng trí nhớ ngắn hạn định hướng dài hạn còn được viết tắt là LSTM làm một kiến trúc đặc biệt của RNN. Bước đầu tiên trong LSTM sẽ quyết định xem thông tin nào chúng ta sẽ cho phép đi qua ô trạng thái (cell state). Nó được kiểm soát bởi hàm sigmoid trong một tầng gọi là tầng quên (*forget gate layer*).

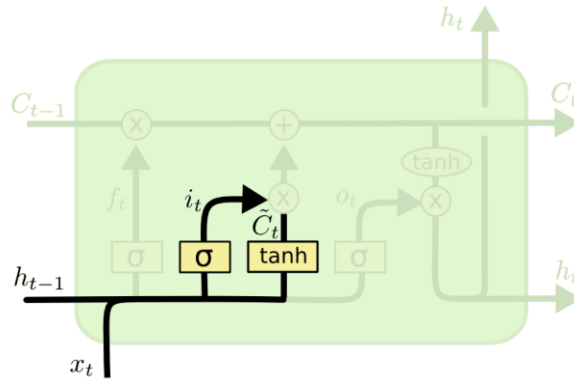
Đầu tiên nó nhận đầu vào là 2 giá trị h_{t-1} và x_t và trả về một giá trị nằm trong khoảng 0 và 1 cho mỗi giá trị của ô trạng thái C_{t-1} . Nếu giá trị bằng 1 thể hiện ‘giữ toàn bộ thông tin’ và bằng 0 thể hiện ‘bỏ qua toàn bộ chúng’.



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Bước tiếp theo chúng ta sẽ quyết định loại thông tin nào sẽ được lưu trữ trong ô trạng thái. Bước này bao gồm 2 phần. Phần đầu tiên là một tầng ẩn của hàm sigmoid

được gọi là tầng cổng vào (*input gate layer*) quyết định giá trị bao nhiêu sẽ được cập nhật. Tiếp theo, tầng ẩn hàm tanh sẽ tạo ra một véc tơ của một giá trị trạng thái mới \tilde{C}_t mà có thể được thêm vào trạng thái. Tiếp theo kết hợp kết quả của 2 tầng này để tạo thành một cặp nhật cho trạng thái.

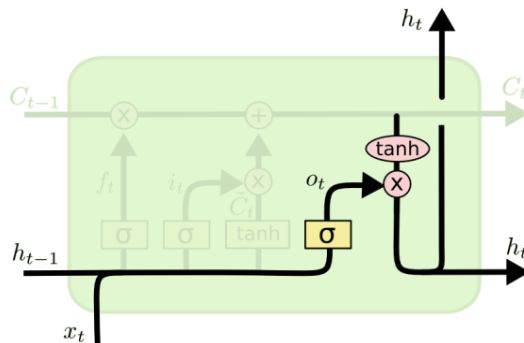


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Đây là thời điểm để cập nhật một ô trạng thái cũ, C_{t-1} sang một trạng thái mới C_t . Những bước trước đó đã quyết định làm cái gì, và tại bước này chỉ cần thực hiện nó.

Chúng ta nhân trạng thái cũ với f_t tương ứng với việc quên những thứ quyết định được phép quên sớm. Phần tử đề cử $i_t \cdot \tilde{C}_t$ là một giá trị mới được tính toán tương ứng với bao nhiêu được cập nhật vào mỗi giá trị trạng thái.

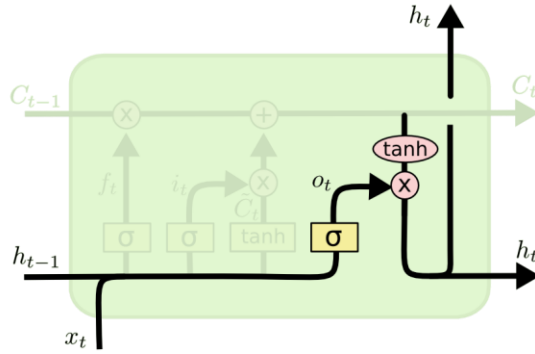


$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Cuối cùng cần quyết định xem đầu ra sẽ trả về bao nhiêu. Kết quả ở đầu ra sẽ dựa trên ô trạng thái, nhưng sẽ là một phiên bản được lọc. Đầu tiên, chúng ta chạy qua một tầng sigmoid nơi quyết định phần nào của ô trạng thái sẽ ở đầu ra. Sau đó, ô trạng

thái được đưa qua hàm tanh (để chuyển giá trị về khoảng -1 và 1) và nhân nó với đầu ra của một cổng sigmoid, do đó chỉ trả ra phần mà chúng ta quyết định.



$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Xây dựng model

```

import torch
from sklearn.preprocessing import MinMaxScaler

# Configure the window size
seq_length = 3 # input time step
num_predict = 1 # number of step in target

def sliding_windows(data, seq_length, num_predict=1):
    x = []
    y = []

    for i in range(data.shape[0]-seq_length-num_predict):
        # _x = data.iloc[i:(i+seq_length), :].values # shape (seq_length, num_symbols)
        # _y = data.iloc[(i+seq_length):(i+seq_length+num_predict), :].values[0] # shape (num_symbols,)
        _x = data[i:(i+seq_length), :] # shape (seq_length, num_symbols)
        _y = data[(i+seq_length):(i+seq_length+num_predict), :][0] # shape (num_symbols,)
        x.append(_x)
        y.append(_y)

    return np.array(x), np.array(y)

# Scaling dataset
sc = MinMaxScaler()
training_data = sc.fit_transform(df_horizontal_symbols)

X, y = sliding_windows(training_data, seq_length=seq_length, num_predict=num_predict)

print(X.shape, y.shape)
# Train/test split
train_size = int(len(y) * 0.9)
test_size = len(y) - train_size

# Convert into variable
dataX = torch.Tensor(np.array(X))
dataY = torch.Tensor(np.array(y))

trainX = torch.Tensor(np.array(X[0:train_size]))
trainY = torch.Tensor(np.array(y[0:train_size]))

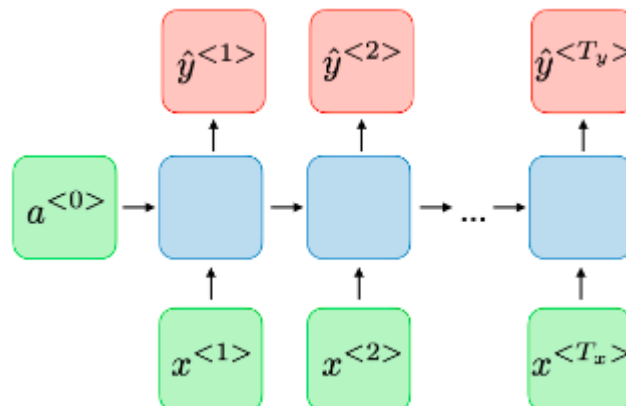
valX = torch.Tensor(np.array(X[train_size:len(X)]))
valY = torch.Tensor(np.array(y[train_size:len(y)]))

print(trainX.shape, valX.shape)
print(trainY.shape, valY.shape)

```

Many-to-Many

Dựa trên mô hình LSTM chúng ta có thể xem xét bài toán là một tác vụ dự báo 'many-to-many'. Trong đó mỗi một giá trị đầu vào là một véc tơ window của một mã chứng khoán thứ i. Các đầu ra là giá trị dự báo của phiên tiếp theo của chính mã chứng khoán đó.



Xây dựng model

```
import matplotlib.pyplot as plt

# Convert tensor to numpy
y_predict = train_test_predict.data.numpy()
y_target = dataY.data.numpy()

# Inverse scaling
y_predict = sc.inverse_transform(y_predict)
y_target = sc.inverse_transform(y_target)

# Plot
def plt_line(target, pred, symbols):
    colors = ['green', 'blue', 'black', 'red', 'yellow']
    def _label(symbol, actual = True):
        label = symbol + '-actual' if actual else symbol + '-predict'
        return label
    plt.figure(figsize=(18, 12))
    plt.axvline(x=train_size, c='r', linestyle='--')
    plt.text(x=train_size, y=40, s='train test split')
    for i, symbol in enumerate(symbols):
        plt.plot(target[:, i], label=_label(symbol, True), color=colors[i])
        plt.plot(pred[:, i], label=_label(symbol, False), linestyle='--', color=colors[i])
    plt.legend()
    plt.suptitle('Vnquant Time-Series Prediction')
    plt.show()

all_symbols = list(df_horizontal_symbols.columns)
for i in range(6):
    symbols = all_symbols[i*5:(i+1)*5]
    plt_line(y_target[:, i*5:(i+1)*5], y_predict[:, i*5:(i+1)*5], symbols)
```

TÀI LIỆU THAM KHẢO

1. [Khoa học dữ liệu \(phamdinhhkhanh.github.io\)](https://github.com/phamdinhhkhanh/vnquant)
2. https://github.com/phamdinhhkhanh/vnquant?fbclid=IwAR08gL691NgRftf0Qi1Jxjt5lesCVN6xdlAgR4dyZOvtdWYMG090Uk3Q_9s
3. https://www.kaggle.com/competitions/stock-market-prediction/data?fbclid=IwAR08gL691NgRftf0Qi1Jxjt5lesCVN6xdlAgR4dyZOvtdWYMG090Uk3Q_9s

PHỤ LỤC