HỘI NGHỊ SINH VIÊN NGHIÊN CỨU KHOA HỌC HUTECH 2020

ỨNG DỤNG LONG SHORT-TERM MEMORY TRONG DỰ ĐOÁN TÀI CHÍNH

Trương Tấn Phát và Phạm Nguyễn Hoàng Vĩnh Phúc

Khoa Công nghệ Thông tin, Trường Đại học Công Nghệ TP. Hồ Chí Minh (HUTECH)

Giảng viên hướng dẫn: Bùi Mạnh Toàn

TÓM TẮT

Các nhà đầu tư chứng khoán luôn tìm cách dự đoán những chuyển biến tiếp theo của giá trị hàng hoá mà họ đang tham gia theo dõi, giao dịch. Trong quá trình dự đoán, các thuật toán dự đoán bằng máy tính được ứng dụng bên cạnh việc phán đoán dựa trên kĩ thuật và kinh nghiệm của cá nhân. Máy học là một trong những kĩ thuật tiềm năng được nghiên cứu và ứng dụng trong các bài toán tương tự. Trong bài viết này sử dụng kĩ thuật có tên “long short-term memory” (bộ nhớ dài-ngắn hạn), là một kĩ thuật dùng để tìm hiểu khả năng dự đoán các bước chuyển biến của thị trường giao dịch tài chính.

*Từ khoá: finance, long short-term memory, lstm, marchine learning, python, time series forecasting.*

1. GIỚI THIỆU
   1. Long short-term memory
      1. Lý thuyết [4]

Mạng bộ nhớ dài-ngắn (long short term-memory networks), thường được gọi là LSTM - là một dạng đặc biệt của Recurrent neural network, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi Hochreiter & Schmidhuber (1997), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng *tanh*.

LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng 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.

* + 1. Ý tưởng cốt lõi của LSTM

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. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.

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, còn khi là 11 thì có nghĩa là cho tất cả các thông tin đi qua nó. Một LSTM gồm có 3 cổng để duy trì và điều hành trạng thái của tế bào.

* 1. Hoạt động
     1. Bước 1

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 ds rồi đưa ra kết quả là một số trong khoảng [0,1][0,1] cho mỗi số trong trạng thái tế bào C\_{t-1}Ct−1​. Đẩu 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.

* + 1. Bước 2

Bước tiếp theo là quyết định xem thông tin mới nào ta sẽ lưu vào trạng thái tế bào. Việc này gồm 2 phần. Đầu tiên là sử dụng một tầng sigmoid được gọi là “tầng cổng vào” (input gate layer) để quyết định giá trị nào ta sẽ cập nhập. Tiếp theo là một tầng tanh tạo ra một véc-tơ cho giá trị mới  nhằm thêm vào cho trạng thái. Trong bước tiếp theo, ta sẽ kết hợp 2 giá trị đó lại để tạo ra một cập nhập cho trạng thái.

* + 1. Bước 3

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 ta muốn xuất ra. Sau đó, ta đưa nó trạng thái tế bảo qua một hàm tanh để có giá trị nó về khoảng [-1, 1][−1,1], và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn.

1. THỬ NGHIỆM
   1. Dữ liệu thử nghiệm

* Dữ liệu thử nghiệm được dùng trong bài được trích từ nguồn dữ liệu giao dịch của Yahoo Finance.
* Dữ liệu là tập hợp chuỗi giá trị của cột giá đóng (Close) của mỗi công ty, được sắp tuần tự tăng dần theo cột ngày tháng (Date) để dùng trong việc train và test của thuật toán.
* Dữ liệu đầu vào được lưu dưới dạng file CSV, mỗi công ty sẽ có một tập dữ liệu riêng.
* Dữ liệu bắt đầu từ ngày đầu tiên các công ty được cập nhật tên trên Yahoo Finance, kết thúc đồng loạt vào ngày 24/03/2020.
* Dữ liệu dùng để training và testing của thuật toán được chia theo tỉ lệ 80%-20% của mỗi tập đầu vào.

Bảng 1. Giá trị đầu vào

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Thứ tự | Công ty | Mã thị trường | Ngày bắt đầu | Ngày kết thúc | Dữ liệu đầu vào | Dạng file lưu trữ |
| 1 | Apple | AAPL | 12/12/1980 | 24/03/2020 | Close price  Date | File CSV (.csv) |
| 2 | Alphabet | GOOGL | 19/08/2004 |
| 3 | Carriage Services | CSV | 09/08/1996 |
| 4 | IBM | IMB | 02/01/1962 |
| 5 | Electronic Art | EA | 20/09/1989 |
| 6 | Activision blizzard | ATVI | 25/10/1993 |

* 1. Môi trường thử nghiệm
* Môi trường thực hiện: Google Colab.
* Ngôn ngữ thực hiện: Python 3.

Bảng 2. Giá trị cài đặt của thử nghiệm

|  |  |  |
| --- | --- | --- |
| Thứ tự | Chỉ số | Giá trị |
| 1 | Batch\_size train | 10 |
| 2 | Batch size test | 10 |
| 3 | Epochs | 5 |
| 4 | Lookback | 5 |
| 5 | Optimizer | adam |
| 6 | Loss | mean\_squared\_error |
| 7 | Lượng dữ liệu training | 80% |
| 8 | Lượng dữ liệu testing | 20% |
| 9 | Số ngày dự đoán | 30 |
| 10 | Thời gian dự đoán | 24/03/2020 ~ 24/04/2020 |

* 1. Kết quả dự đoán

|  |  |
| --- | --- |
|  |  |
| *Hình 1. Dự đoán của Apple cho 30 ngày tiếp theo* | *Hình 2. Giá trị thực của Apple* |
|  |  |
| *Hình 3. Dự đoán của Alphabet cho 30 ngày tiếp theo* | *Hình 4. Giá trị thực của Alphabet* |
|  |  |
| *Hình 5. Dự đoán của Carriage Services cho 30 ngày tiếp theo* | *Hình 6. Giá trị thực của Carriage Services* |
|  |  |
| *Hình 7. Dự đoán của IBM cho 30 ngày tiếp theo* | *Hình 8. Giá trị thực của IBM* |
|  |  |
| *Hình 9. Dự đoán của Electronic Art cho 30 ngày tiếp theo* | *Hình 10. Giá trị thực của Electronic Art* |
|  |  |
| *Hình 11. Dự đoán của Activistion Blizzard cho 30 ngày tiếp theo* | *Hình 12. Giá trị thực của Activistion Blizzard* |

* Các kết quả dự đoán có hướng phát triển trùng với hướng phát triển của dữ liệu thật, nhưng số liệu dự đoán đa phần thì không khớp.

1. KẾT LUẬN

Mô hình long short-term memory cho khả năng dự đoán trong thời gian ngắn có tính chính xác ở mức khá và có thể tin tưởng. Trong trường hợp dự đoán những dữ liệu về lâu dài thì thông tin dự đoán chỉ mang tính chất tương đối. Mô hình được thực hiện nhiều lần thử và các kết quả có thể thay đổi giữa các lần. Mô hình hứa hẹn sẽ có thể trở thành một tham chiếu để tham khảo trong quá trình dự đoán các giá trị tương lai của thị trường giao dịch.

Dựa vào mô hình dự đoán, ta có thể nắm được xu thế của thị trường sẽ đi lên hoặc đi xuống, nhưng không thể nắm được tình hình thay đổi của xu hướng thị trường. Mô hình cần được thực hiện thêm nhiều lần nữa cùng với những cải tiến khác để có thể đưa ra dự đoán chính xác hơn.

TÀI LIỆU THAM KHẢO

1. Keras: The Sequential model API (truy cập lần cuối: 20/04/2020)

<https://keras.io/guides/sequential_model/>

1. Keras: Getting started with the Keras Sequential model (truy cập lần cuối: 20/04/2020)

<https://faroit.com/keras-docs/1.0.1/getting-started/sequential-model-guide/>

1. George V Jose (2019) Predicting Sequential Data using LSTM: An Introduction

<https://towardsdatascience.com/time-series-forecasting-with-recurrent-neural-networks-74674e289816>

1. Do Minh Hai, Hai's Blog (2017) [RNN] LSTM là gì? (truy cập lần cuối: 20/04/2020) <https://dominhhai.github.io/vi/2017/10/what-is-lstm/>
2. Nttuan8 (2019) Long short-term memory (LSTM) (truy cập lần cuối: 20/04/2020)

<https://nttuan8.com/bai-14-long-short-term-memory-lstm/>

1. phamduytung (2018) Phân biệt Epoch – Batch (truy cập lần cuối: 20/04/2020)

<https://www.phamduytung.com/blog/2018-10-02-understanding-epoch-batchsize-iterations/>

1. plotly: Scatter Plots in Python (truy cập lần cuối: 20/04/2020)

<https://plotly.com/python/line-and-scatter/>

1. randerson112358 (2019) Stock Price Prediction Using Python & Machine Learning (truy cập lần cuối: 20/04/2020)

<https://medium.com/@randerson112358/stock-price-prediction-using-python-machine-learning-e82a039ac2bb>

1. Dữ liệu: <https://bit.ly/30n9WKL> - <https://bit.ly/3eYIw1W>