Journal of Physics: Conference Series

Stock price prediction using the RNN model

Stock price prediction using the RNN model

Yongqiong Zhu*

School of Art, Wuhan Business University, Wuhan, China

*Corresponding author: yongqiongzhu@gmail.com

Abstract. This paper proposes a deep learning technique to predict the stock market. Since RNN has the advantage of being able to process time series data, it is very suitable for forecasting stocks. Therefore, we use the RNN network and use Apple's stock price in the past ten years as data set to predict. Experiments show that the prediction accuracy is over 95%, and the loss close to 0.1%.

Keywords: deep learning, RNN, stock prediction.

1. Introduction

Stock is a form of trading that can express rhythm with numbers. Because of the law of large numbers, it defines that all behaviors in the world can be represented by numbers, and there are certain objective laws. Stocks are no exception. What quantitative trading needs to do is to find the trend of stocks through mathematical models, that is, through the fall or rise of stocks in the past period of time, it is concluded that when there is a certain fluctuation, the stock will have a corresponding rise or fall trend.

RNN [1]is a deep learning network structure. The advantage of RNN is that it considers the context of the data during the training process, can process time series data, and is very suitable for stock prediction. Because stock price fluctuations at a certain moment often have some connection with previous trends.

The structure of this paper is as follows. In the second part, the related work of the research and the objectives of this research are introduced. The third part introduces the system model of this paper, and gives the way to achieve it. The fourth part is experimental simulation to prove our method. The fifth part is the conclusion.

2. Related work

Anshul Mittal [2] apply sentiment analysis and machine learning principles to find the correlation between“public sentiment”and “market sentiment” in 2012. They propose a new cross validation method for financial data and obtain 75.56% accuracy using Self Organizing Fuzzy Neural Networks (SOFNN) on the Twitter feeds and DJIA values from the period June 2009 to December 2009. In 2013, Jianfeng Si [3] proposes a technique to leverage topic-based sentiments from Twitter to help predict the stock market. They first utilize a continuous Dirichlet Process Mixture model to learn the daily topic set. For each topic they build a sentiment time series. Then, they regress the stock index and the Twitter sentiment time series to predict the market. Ayodele A. Adebiyi [4] presents extensive process of building stock price predictive model using the ARIMA model in 2014. Results obtained revealed that the ARIMA model has a strong potential for short-term prediction and can compete favorably with existing techniques for stock price prediction. In 2015，Xiao Ding [5] propose a deep learning method for event driven stock market prediction. First, events are extracted from news text, and represented as dense vectors, trained using a novel neural tensor network. Second, CNN is used to model both short-term and long-term influences of events on stock price movements. There are some similar studies. Amin Hedayati Moghaddam [6] use the ability of artificial neural network (ANN) in forecasting the daily NASDAQ stock exchange rate in 2016. In 2017, Eunsuk Chong [7] offer a systematic analysis of the use of deep learning networks for stock market analysis and prediction. In the same year，David M. Q. Nelson [8] usage of LSTM networks on that scenario, to predict future trends of stock prices based on the price history, alongside with technical analysis indicators.

In this article, we also use deep learning-based method to predict the price of stocks, but we use the RNN network, because the RNN network can process time series data, which is very suitable for stock prediction.

3. Design

3.1. Data preprocessing.
For the downloaded stock price list, the data needs to be preprocessed and normalized uniformly to scale the data to the range [0, 1].

For the input data matrix X, for each data $x_{i,j}$ in X:

$$x_{i,j} = \frac{(x_{i,j} - x_{\min(axis=j)})}{x_{\max(axis=j)} - x_{\min(axis=j)}} \cdot (max - min) + min \tag{1}$$

Here $axis = j$ means to do such a normalization operation for each column of the matrix X.

3.2. Network model.
The purpose of RNN is to process sequence data. In the traditional neural network model, from the input layer to the hidden layer to the output layer, the layers are fully connected, and the nodes between each layer are not connected. But this kind of ordinary neural network is powerless for many problems. The reason why RNN is called recurrent neural network means that the current output of a sequence is also related to the previous output. The specific form of expression is that the network will memorize the previous information and apply it to the calculation of the current output, that is, the nodes between the hidden layers are no longer unconnected but connected, and the input of the hidden layer not only includes the output of the input layer It also includes the output of the hidden layer at the previous moment. In theory, RNN can process sequence data of any length.

The hidden layer structure of RNN as figure 1:

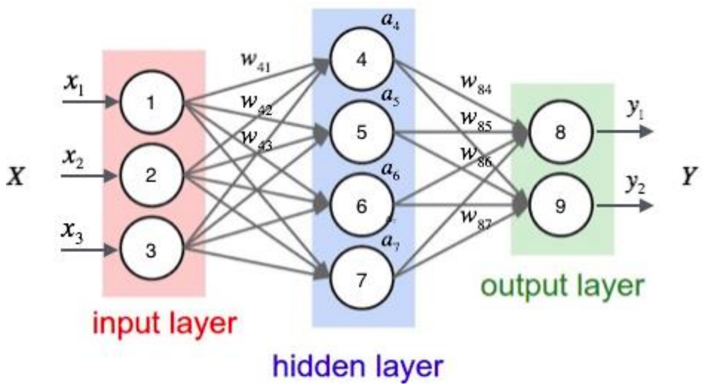


Figure 1. RNN hidden layer.

The value s of the hidden layer of RNN not only depends on the current input x, but also depends on the value s of the last hidden layer, as shown in figure 2:

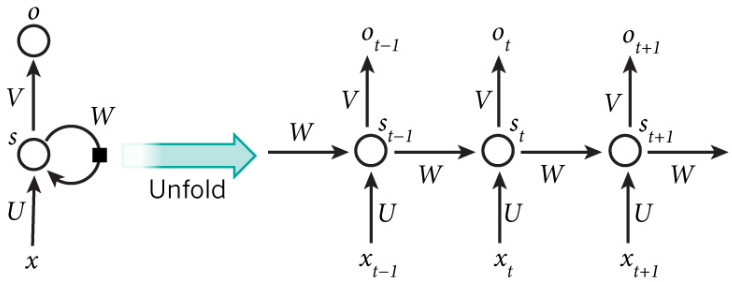


Figure 2. RNN hidden layer calculate process.

Among them, t is the time, x is the input layer, s is the hidden layer, o is the output layer, and the matrix W is the last value of the hidden layer as the weight of this input.

RNN training also uses the BP error back propagation algorithm, but there is a little difference. In the training process, the parameters W, U, V are shared, but the traditional fully connected neural network is not. And in using the gradient descent algorithm, the output of each step not only depends on the current step of the network, but also depends on the state of the previous steps of the network.

3.3. Performance evaluation index.

Experimental performance is evaluated using MSE, RMSE and MAE.

MSE is mean_squared_error:

$$MSE = E[(predictvalue_j - truevalue_j)^2] \tag{2}$$

RMSE is the square root of MSE:

$$RMSE = \sqrt{MSE} \tag{3}$$

MAE stands for mean absolute error, which is the difference between the predicted value and the true value:

$$MAE = \frac{\sum_1^j E[|predictvalue_j - truevalue_j|]}{j} \tag{4}$$

4. Simulation

This experiment is based on tensorflow2.0 [9] simulation environment. The programming language is Python. The CPU is i7-9750H and graphics card is RTX2070, 8G video memory.

4.1. Prepare the data set.

The experiment uses Apple's stock (AAPL). Download the daily trend of this stock from Yahoo Finance [10] for the past 10 years, that is, the data from August 9, 2009 to August 12, 2020. The screenshot of the data is as figure 3:

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Adj Close	Volume
2	2010/8/9	37.35429	37.45	37.08143	37.39286	32.31264	75782000
3	2010/8/10	37.12143	37.20714	36.79286	37.05857	32.02377	112980000
4	2010/8/11	36.48571	36.52714	35.68714	35.74143	30.88557	155013600
5	2010/8/12	35.24143	36.15714	35.16	35.97	31.08309	133730100
6	2010/8/13	35.95	35.98286	35.58429	35.58571	30.75102	88717300
7	2010/8/16	35.36857	35.71571	35.23143	35.37714	30.57079	79607500
8	2010/8/17	35.72572	36.37571	35.6	35.99572	31.10532	105660100
9	2010/8/18	36.05143	36.38143	35.94	36.15286	31.2411	84924000
10	2010/8/19	36.12	36.21143	35.52572	35.69714	30.84731	106676500
11	2010/8/20	35.62714	36.27428	35.57143	35.66286	30.81768	96057500
12	2010/8/23	35.97	36	35.03571	35.11429	30.34363	103510400
13	2010/8/24	34.66714	34.71429	34.09286	34.27572	29.619	150641400
14	2010/8/25	34.00571	34.85571	33.88572	34.69857	29.98441	149216900
15	2010/8/26	35.06429	35.10714	34.32571	34.32571	29.6622	116626300
16	2010/8/27	34.53571	34.65857	33.65143	34.51714	29.82762	137097800
17	2010/8/30	34.39429	35.10714	34.38286	34.64286	29.93625	95822300
18	2010/8/31	34.55	34.93714	34.33571	34.72857	30.01033	105196700
19	2010/9/1	35.35286	35.92286	35.18286	35.76143	30.90286	174259400

Figure 3. AAPL stock price.

The downloaded data file is AAPL.csv, which can be opened and viewed in EXCEL. As can be seen from the table, each row of the CSV data contains data such as the trading time, opening price, highest price, lowest price, and closing price of the stock. This experiment only predicts the opening price of the stock, so you only need to care about the open field in the table.

Read the file AAPL.csv and use 65% of the data as the training set, and the other 35% as the test set. The validation set uses the data set of the test set.

The construction of train_x, train_y and test_x , test_y in the data set depends on the timesteps parameter, as table 1. When timesteps=5, the stock price of the day is predicted based on the stock price of the previous 5 days. For example, predict the value of August 16, 2010 based on the value of August 9, 2010 to August 13, 2010.

Table 1. Train_x and train_y constructed when timesteps=5.

train_X	train_Y
[37.354286, 37.121429, 36.485714, 35.241428, 35.950001]	[35.368572]
[37.121429, 36.485714, 35.241428, 35.950001， 35.368572]	[35.725716]
[36.485714, 35.241428, 35.950001， 35.368572， 35.725716]	[36.05143]
…	…

The data construction method of test_x and test_y is the same.

4.2. Network parameter setting.

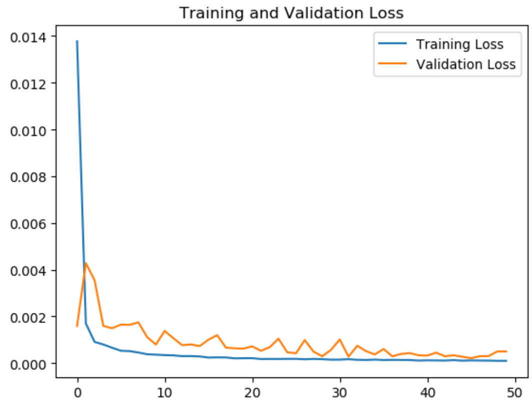
We use a two-layer RNN network to simulate the experiment. The dimensions of the input and output data vectors are both 1, that is, Input_size is 1. Timesteps is the step size, and the input data and label value are set according to the value of the timesteps. The first-layer RNN network has 50 unites nodes, and the second-layer RNN network has 100 unites nodes. The loss is judged by MSE. Network parameter settings are shown in Table 2.

Table 2. RNN Parameters.

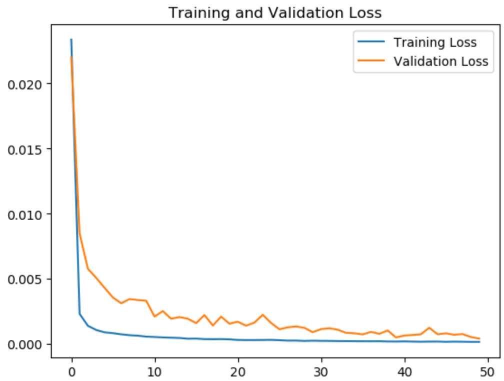
Input_size	1
timesteps	5/10
Units1	50
Units2	100
dropout	0.2
batch_size	64
epochs	50
optimize	Adam
loss	MSE

4.3. Experiment performance.

Now we compare the network performance when timesteps=5 and timesteps=10. Figure 4 shows the decrease in losses in both cases. The blue curve is the loss rate of the training set, and the yellow curve is the loss rate of the validation set. It can be seen from the figure 4 that after the epoch is 30, the loss rate is basically stable, reaching about 0.1%.



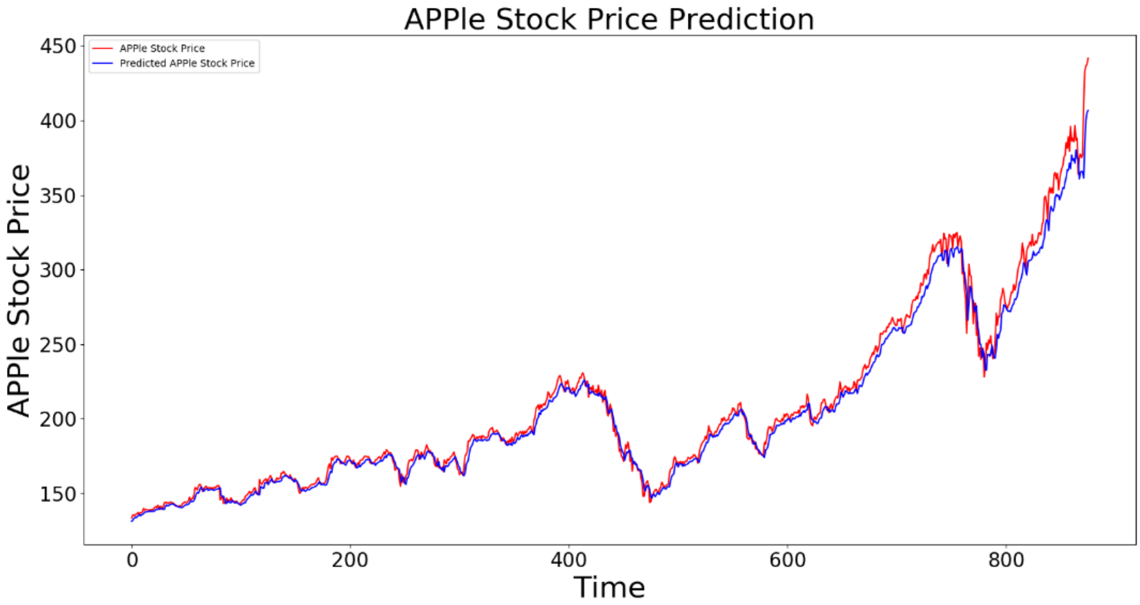
timesteps=5



timesteps=10

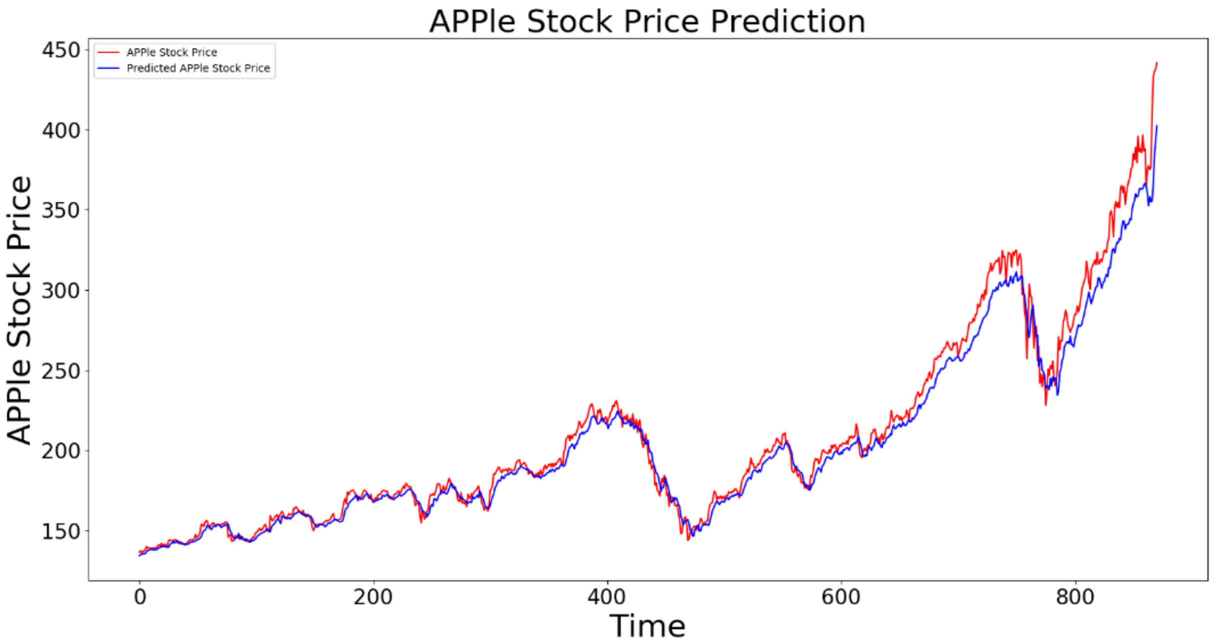
Figure 4. Loss rate.

Figures 5 and 6 show the curves of the predicted value and the true value of the stock price under the conditions of timesteps=5 and timesteps=10. The red curve is the true value of Apple stock, and the blue curve is the predicted value of the stock. It can be seen from Figure 5 and Figure 6 that the curve fitting between the predicted value and the true value is better, and the predicted value can accurately predict the trend of stock prices.



timesteps=5

Figure 5. Comparison of the predicted value and the true value .



timesteps=10

Figure 6. Comparison of the predicted value and the true value.

The comparison of MSE, RMSE and MAE values is given as table 3:

Table 3. Comparison of MSE, RMSE and MAE .

	timesteps=5	timesteps=10
MSE	68.492027	109.218124
RMSE	8.275991	10.450748
MAE	5.903805	7.835381

It can be seen from the table that the MAE value calculated by the predicted value and the true value within acceptable range. But as the timesteps increases, the MAE value will become larger.

5. Conclusion

Methods based on deep learning are very suitable for predicting stocks. RNN can predict stock prices well when timesteps is small, with high accuracy. But when the timesteps increase, the error will also increase. Further improvements are needed.

Acknowledgement

This paper is supported by grant from the academic team of Wuhan Business University (2020TD001).

References

[1] Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization[J]. arXiv preprint arXiv:1409.2329, 2014.
[2] Mittal A, Goel A. Stock prediction using twitter sentiment analysis[J]. Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf), 2012, 15.
[3] Si J, Mukherjee A, Liu B, et al. Exploiting topic based twitter sentiment for stock prediction [C]//Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2013: 24-29.
[4] Ariyo A A, Adewumi A O, Ayo C K. Stock price prediction using the ARIMA model [C]//2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. IEEE, 2014: 106-112.
[5] Ding X, Zhang Y, Liu T, et al. Deep learning for event-driven stock prediction [C]//Twenty-fourth international joint conference on artificial intelligence. 2015.
[6] Moghaddam A H, Moghaddam M H, Esfandyari M. Stock market index prediction using artificial neural network [J]. Journal of Economics, Finance and Administrative Science, 2016, 21(41): 89-93.
[7] Chong E, Han C, Park F C. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies [J]. Expert Systems with Applications, 2017, 83: 187-205.
[8] Nelson D M Q, Pereira A C M, de Oliveira R A. Stock market's price movement prediction with LSTM neural networks [C]//2017 International joint conference on neural networks (IJCNN). IEEE, 2017: 1419-1426.
[9] Sergeev A, Del Balso M. Horovod: fast and easy distributed deep learning in TensorFlow [J]. arXiv preprint arXiv:1802.05799, 2018.
[10] https://finance.yahoo.com/quote/AAPL