Proceedings of the 2023 International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/8/20230277

Exploiting long short-term memory neural network for stock price prediction

Zihui Chen

Faculty of Information Science and Engineering, Ocean University of China, Qingdao, Shandong, 266000, China

921018234@qq.com

Abstract. Stock as a high yield, high risk investment has been favored by the public. In order to increase the return on investing in stocks, investors need to predict stock prices. In the past, investors used traditional mathematical methods to make predictions. Now, neural networks are used by investors to predict stocks, which can improve the accuracy of stock forecasting. To further verify the effectiveness of these methods, this work discusses the effects of different network structures and hyperparameters on stock prediction models using short-term memory (LSTM) neural networks. The results show that deeper network layer can get better training effect, but it needs more training time, resulting in a lot of time waste. In addition, this experiment tests the prediction effect under different dropout parameters. The results show that the dropout function should not be too large or too small. Multiple experiments are needed to find an appropriate dropout value.

Keywords: stock price prediction, neural network, deep learning, LSTM

1. Introduction

As an investment method, compared with other investment methods such as funds and bonds, stocks bear higher risks, but they also have the characteristics of high returns [1,2]. Therefore, the stock as a way of investment is very popular with the public. In order for investors to get higher returns, the timing of their buying and selling of stocks becomes very important. Investors never stop searching for stock predictions. With the improvement of computer computing power and breakthroughs in relevant theories in the field of mathematics, people find that the non-machine learning method is not ideal for the prediction effect of large and complex data such as stocks [3,4]. Therefore, people use machine learning to predict stocks, among which LSTM neural network can efficiently process nonlinear data, and its time series characteristics are consistent with the characteristics of stocks, so LSTM neural network has relatively excellent performance in stock prediction [5-7]. This experiment mainly studies whether there is an impact on the training effect of LSTM networks after changing various parameters or changing the structure of LSTM models, to help investors to boost the accuracy of stock return prediction, and to study the prediction effect of LSTM neural networks on stocks of different companies in China. The data set for this experiment came from MSN Finance, which collected daily stock data such as the opening price, the highest price, etc., since 2013 [8,9]. In addition, this study provides suggestions for stock forecasting to improve the accuracy of investors' stock forecasting. In this study, LSTM neural

DOI: 10.54254/2755-2721/8/20230277

network model is used to predict stock returns. The training of the model leverages mean absolute error (MAE) to measure the loss, and ADMA as the optimization algorithm. Three evaluation matrixes, including MAE, mean square error (MSE), and Root mean square error (RMAE). In this experiment, five characteristics are leveraged, which are the opening, closing, lowest, and highest prices together with the volume of trade.

2. Method

2.1. Dataset

The dataset from MSN Finance and Economics was used as a dataset for training in this experiment. The site records detailed stock parameters at different times. Such as growth rate, opening price, profitability, etc. At the same time, MSN Money is very large in terms of the volume of stock trading data. So, the experiment used stock data from MSN Finance from 2013 to 2020.

2.2. LSTM

This model could date back to 1997, designed by Sepp Hochreit et, al. It is offered to solve the problem of prolonged artificial time task which is difficult for RNN to solve, and the problem that RNN is prone to gradient disappearance [10].

LSTM is a new neural network developed on the basis of RNN. The feature of LSTM is that there are four states inside a single cell of LSTM, and a persistent state is maintained between LSTM cycle structures to keep the data, and the structural characteristics of LSTM determine which information to discard or continue to save. RNNS don't store information easily. The structure of LSTM itself is characterized by information memory and transmission and status update through input, forgetting, and output gates.

The oblivion gate will control the information that passes through it and make a judgment about whether it is retained or not. It reads the entry of the current instant and the exit of the previous instant which was retained in the message. After the activation function, generate a value, based on this value, to determine whether or not the information is stored.

$$\mathrm{f}_t = \sigma(\mathrm{W}_f \times (\mathrm{h}_{t-1}, \mathrm{X}_t) + \mathrm{b}_f) \tag{1}$$

Among them of h_{t-1} represents the output at the t stage, X_t denotes the corresponding input, W_f said forget the door to the weight of information, the b_f respectively oblivion gate bias, σ said use Sigmoid function.

Secondly, the information goes through the main door, which determines how the information is updated and the status. There are two stages: a sigmod layer identifies the information to be updated; A tanh layer generates content that requires updating.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2}$$

Leveraging them, the model is updated. The C_t said updated information, i_t for C_t coefficient, C_{t-1} table shown on the status of the moment $\tilde{C}_t$ said the updated status.

The final message will pass through the output gate to get the final output value, which will be based on the updated state. First, A sigmod control point is used to identify which part of the information will be generated, and then the updated information is processed and multiplied with the sigmod door release to obtain the final release.

$$i_t = \sigma(\mathrm{W}_i \times (\mathrm{h}_{t-1}, \mathrm{X}_t) + \mathrm{b}_i) \tag{3}$$

$$\tilde{C}_t = \tanh(\mathrm{W}_C \times (\mathrm{h}_{t-1}, \mathrm{X}_t) + \mathrm{b}_C) \tag{4}$$

The LSTM uses special implicit units to hold long-term inputs and controls the required information by placing units in gates, oblivion gates, and output gates, allowing information to be stored for long periods of time. It solves the problem that RNNS are prone to gradient disappearance and can handle

DOI: 10.54254/2755-2721/8/20230277

long-term sequences. The characteristics of RNN model are inherited by LSTM, and the problem of RNN gradient disappearing is solved.

But it also has drawbacks. Its special cellular structure means that it stores four fully connected layers in each cell. If the LSTM network model has a very large number of layers, it means that the amount of computation of the model will become rather large and that the time consumed will also become very large.

In this experiment, the keras deep learning setting was used to build the template quickly. The sequential model is built, the LSTM layer is appended, with dropout rate at 0.5, and the Dense layer is added, and its dimension is aggregated at 1. Use Relu as the activating feature. The loss function is based on the mean absolute error (MAE). Adam was used for the optimisation algorithm, and 50 epoch were used for the template, with each batch size of 100. In this experiment, different Dropout parameters are set for training, and the results are sorted out and analyzed. And the same data set is used for training under the LSTM model with different layers, and the results are analyzed and compared.

2.3. Evaluation index

In this paper, MSE, RMSE and MAE are used to evaluate the performances.

MSE returns the mean sum of squares of the subtraction results within the estimated and the actual output results. MSE can assess the extent to which data has been modified. The lower the MSE value, the more accurate the prediction model can describe the experimental data.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{5}$$

The square root of MSE is called RMSE, and its practical meaning is to measures the standard deviation and RMSE is leveraged for depicting the sample dispersion. Smaller parameter value of RMSE indicates better the fitting effect.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{6}$$

MAE between the forecast and the actual data is another measurement. The lower the MAE value, the better the prediction effect of the model.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{7}$$

3. Result

3.1. Training process

In this experiment, stock data of 600,000. SH from 2013 to 2017 were selected for research. The closing, opening, highest, and lowest prices together with the volume of transactions were selected as features, and the data were normalized and other operations. In this experiment, data for January 4, 2013 to February 15, 2015 were used as a training package., and data from December 29, 2016 to March 14, 2017 were used as a set of predictions to test the effect of the predictive capacity of the model.

LSTM model is used to predict long-term trend items. Its loss curves are illustrated in Figure 1. It can be seen that during the training, the first ten times of loss has a significant decline, and after 20 times, the training loss is close to the test loss.

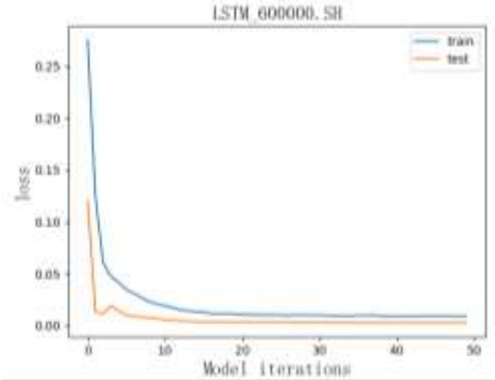

Figure 1. Loss curves for training and testing.

3.2. Impact of various network structures

3.2.1. Impact of different LSTM layers. The experiment was completed by data acquisition, data preprocessing, neural network model construction and training, network structure adjustment and stock prediction. First of all, this experiment used python to obtain data on opening price, closing price, high point, low point and turnover from 2013 to 2019 on MSN Finance. From 2013-01-04 to 2015-02-15, a total of 812 data including opening, closing, highest point, lowest point and trading volume were selected for training.

Under different neural network structures, LSTM neural network models show different prediction effects. As shown in Table 1, in the first test, the LSTM neural network model is leveraged for prediction. It contains only one hidden layer. The outcome of the predictive model evaluation is depicted in Figure 1. In the last two experiments, one layer and two layers of hidden LSTM neural network were added for training. The latter two experiments were rated better than the first.

Table 1. Impact of different number of LSTM layers.

	MSE	RMSE	MAE
One layer	0.4061	0.6372	0.3641
Two layers	0.3979	0.6308	0.3436
Three layers	0.3934	0.6272	0.3399

3.2.2. Impact of different dropout rates. Under the same neural network structure, different parameter LSTM neural network models will also show different prediction effects. This experiment tested the effect of parametric abandonment on the LSTM neural array, as shown in Table 2. The training package and test package used in this experiment are the same as the prior experience. And the effect on the predicted results is observed using different dropout values of 0.2 to 0.8. Evaluation of prediction models with different dropout is shown below. The prediction effect varies with different dropout values. For example, the RMSE evaluation results using the 0.2dropout value show an evaluation of 0.635 in RMSE. By using the prediction results using the 0.4dropout value, the evaluation in the RMSE drops to 0.633. For the prediction using the 0.8dropout value, the evaluation on the RMSE increases to 0.748.

Table 2. Impact of different dropout rates.

Dropout Rate	MSE	RMSE	MAE
0.2	0.403	0.635	0.354
0.3	0.406	0.637	0.355
0.4	0.401	0.633	0.350
0.5	0.406	0.637	0.358
0.6	0.405	0.636	0.366

Table 2. (continued).

0.7	0.440	0.663	0.409
0.8	0.560	0.748	0.531

3.3. Impact of various network structures

In this experiment, the keras deep learning setting was used to build the template quickly. The sequential model is built, the LSTM layer is appended, with dropout rate 0.5, and the Dense layer is added, and its dimension is aggregated at 1. Use Relu as the activating feature. Finally, the predicted outcomes are illustrated in Figure 2 below.

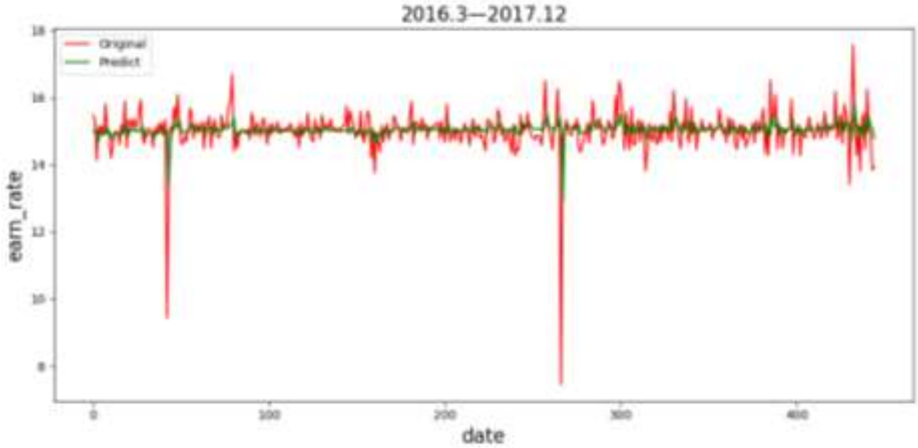

Figure 2. Visualization of prediction.

4. Discussion

After observing the experimental results in Section 3.2 of this paper, it is known that different network models or network parameters have certain influence on the prediction model trained by the network.

In 3.2.1 of this paper, the network structure under different LSTM layers and the same network parameters is trained, and the trained prediction model is used for prediction. MSE, RMSE and MAE were used for evaluation. It can be found that under the network structure with more layers, the evaluation of MSE, RMSE and MAE is higher, indicating that the prediction effect is better. It shows that with the growing of network layers, the performances also improved. However, when the number of layers is increased, the time consumption also getting higher. In addition, when layers are increaseing too much, the improvement of prediction effect will become smaller and smaller, and finally almost no improvement. Therefore, the method of adding network layers cannot be used to improve the effect of prediction model, which will lead to low time efficiency.

In 3.2.2, the network structure is trained with a hidden layer of LSTM and different dropout parameters, and use the trained prediction model to make predictions. MSE, RMSE and MAE were also used for evaluation. By comparing the prediction models trained in the interval of dropout parameters from 0.2 to 0.8, the prediction evaluation of the model first increases and then decreases. It demonstrates that the prediction performances are first improved and then decreased. When the dropout parameter is 0.4, the prediction effect is the best of these experiments. Therefore, to boosting the LSTM performances, directly leveraging the method of improving the dropout parameter is unacceptable. Multiple experiments should be done to test the relatively efficient dropout parameter.

In the stock price prediction model of this experiment, different stock data will have different impacts on the prediction effect, so the selection of data set is also very important. And the stock parameters selected in this experiment are five characteristic data. The limitation of data selection leads to poor prediction results in this experiment. More other characteristic data can be added to improve the accuracy of model prediction, such as stock market fund sentiment characteristics, technical indicators, etc.

DOI: 10.54254/2755-2721/8/20230277

5. Conclusion

In this experiment, LSTM neural network model was used for training with 600000.SH stock data, and stock prediction was conducted on the remaining data. By changing the network structure and parameters of LSTM, the influence on the prediction model is tested. And MSE, RMSE and MAE are used as matrixes. Through the experiment, it was found that under the model with different LSTM layers, the more the model layers, the better the prediction effect, but the higher the training time would be. For different dropout parameters, the prediction effect of the model is getting worse after achieving its climax, as the dropout parameter increases. So, adjusting the dropout function is not always better with bigger, and multiple experiments are needed to predict the optimal parameters. Only five different characteristics of the stock market are used in this experiment, and there are some limitations in data selection. If other experiments are carried out, other relevant parameters can be added to make the process of model prediction more efficient and concise, and obtain more accurate prediction results.

References

[1] Yuniningsih, Y., Widodo, S., & Wajdi, M. B. N. (2017). An analysis of decision making in the stock investment. Economic: Journal of Economic and Islamic Law, 8(2), 122-128.

[2] Barber, B. M., & Odean, T. (2000). Trading is hazardous to your wealth: The common stock investment performance of individual investors. The journal of Finance, 55(2), 773-806.

[3] Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019). Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. Intelligent Systems in Accounting, Finance and Management, 26(4), 164-174.

[4] Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. Procedia computer science, 167, 599-606.

[5] Ghosh, A., Bose, S., Maji, G., Debnath, N., & Sen, S. (2019). Stock price prediction using LSTM on Indian share market. In Proceedings of 32nd international conference on, 63, 101-110.

[6] Ding, G., & Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. International Journal of Machine Learning and Cybernetics, 11, 1307-1317.

[7] Mehtab, S., & Sen, J. (2020). Stock price prediction using CNN and LSTM-based deep learning models. In 2020 International Conference on Decision Aid Sciences and Application (DASA), 447-453.

[8] Feldman, J., Muthukrishnan, S., Sidiropoulos, A., Stein, C., & Svitkina, Z. (2010). On distributing symmetric streaming computations. ACM Transactions on Algorithms (TALG), 6(4), 1-19.

[9] Hsu, H. C. (2010). Using MSN money to perform financial ratio analysis. Journal of College Teaching & Learning (TLC), 7(9), 25-36.

[10] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.