Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230696

The performance analysis of stock predication based on recurrent neural network

Shiying Wang[1, 3, †] and Xinyu Yao[2, †]

[1]School of Economics and Management, East China Jiaotong University, Jiangxi, 330013, China
[2]Information Engineering College, Capital Normal University, Beijing, 100048, China

[†]These authors contributed equally

[3]2020041011000105@ecjtu.edu.cn

Abstract. The stock exchange is unpredictable, and the stock price seems unpredictable. However, with the continuous development of the deep learning model's ability to deal with massive data, forecasting stock prices has become feasible and has reference value for investors. Many factors affect the stock price, and it is a great challenge to define these factors' influence on the price clearly. This paper selects multi-features stock price data sets of different companies. Because of the superiority of recurrent neural networks in dealing with time series problems, this paper compares and analyzes the experimental results of four models, namely Long Short Term Memory, Bi-directional Long Short-Term Memory, Gate Recurrent Unit, and Bi-directional Gate Recurrent Unit, and concludes that the BiLSTM model is the most outstanding one. At the same time, the prediction accuracy under different feature numbers is compared. The experimental results show that the stock price forecasting model with multi-features shows good performance, but the noise brought by it can't be ignored.

Keywords: recurrent neural network, stock prediction, data analysis, LSTM, GRU.

1. Introduction

Stock market forecasts are an essential part of the financial field. The stock forecast reflects the economic situation of a country or region. As the economic situation is complex and changeable, stock forecasting is challenging. The movement trend of financial products directly affects the stability of the financial market, at the same time, influences the whole economy's healthy development. Therefore, the fit and sustainable development of the world economy must use scientific and efficient research methods to guide stock market transactions. At the same time, stock forecasting is the focus of debate in the financial field. To do analysis of stock price's trends, investors always observe all kinds of subtle changes in the stock market. The exponential smoothing method, multiple regression, Autoregressive Integrated Moving Average model, and Generalized Autoregressive Conditional Heteroskedasticity model have all been utilized in stock market research by academics around the world. However, many factors have influence on the stock price, and the influencing mechanism is very complex, so it is hard to explain them using traditional statistical and econometric models. Even if the statistical and econometric models are studied and improved, the model's calculation results are not superior.

In stock market prediction, White first used a neural network to do prediction of the daily return rate of stocks, which was limited to the shortcomings of the traditional Back Propagation neural network model, and the trained model could not predict accurately [1]. Gen Cay established a comprehensive forward neural network model. An empirical study demonstrated that the neural network model's capability was superior to that of the statistical algorithm [2]. An understandable model was found by Rodriguez et al, the moving average method and feedforward artificial neural network [3]. When it comes to predicting investments, the research demonstrates that neural networks provide helpful guidance. In terms of time series prediction, Peter Zhang compares the neural network model to the ARIMA model. The research demonstrates that the neural network model's prediction accuracy is significantly higher than that of the ARIMA model [4]. Comparing with the stock prediction impacts of neural networks and Bayesian estimates, Murat and Bahadir found that the neural network performed better [5]. Muratozbayoglu used the neural network model to predict the stock [6]. The research shows that the neural network model which can establish the optimal portfolio of Markowitz is superior to other models.

Compared with traditional economic and mathematical models, after Lapcdcs and Farber adopted artificial neural networks in forecasting in 1987, the use of neural networks in stock forecasting attracted the attention of numerous academics over time [7]. Because of their multi-influencing, unstable, and unpredictable nature, complex nonlinear issues like stock forecasting are better suited for neural networks to handle. Time series training forecasting is a problem that traditional BP neural network forecasting models theoretically cannot address, despite the fact that neural networks are effective at action prediction. In the meantime, there are a few issues as well, such as the difficulty in determining the network structure and the amount of input data, as well as the ease with which the local minimum can be reached. An innovative variety of neural network is the recurrent neural network which is based on the idea of time series and is capable of implementing multiple inputs and outputs in time series. The interaction between extended time series is analogous to the self-connecting nature of the hidden layer.

A quick gradient decrease and an inability to converge to the ideal solution are just two of the serious issues that Recurrent Neural Network faces. Therefore, Long Short Term Memory Networks (LSTM) and Gated Recurrent Units (GRU), developed from Recurrent Neural Network (RNN), have solved the above problems to some extent. However, the problems, advantages, and disadvantages of the two methods in applying stock market forecasting still need to be analyzed and improved. This paper selects multi-characteristic stock price data sets of different companies. Because of the superiority of recurrent neural networks in dealing with time series problems, this paper compares and analyzes the experimental results of four models, namely LSTM, BiLSTM, GRU, and BiGRU, and concludes that the BiLSTM model is the most outstanding one. At the same time, the prediction accuracy under different feature numbers is compared. The experimental results give information that the stock price forecasting model with multi-features shows good performance, but the noise brought by it can't be ignored.

2. Methods

2.1. Long short term memory networks

Long Short-Term Memory Networks are one of the most well-known Recurrent Neural Networks. It was first introduced by Hochreiteretal and Schmidhuber [1]. The RNN architecture is designed to process neural networks with time series and variable-length input sequence data. RNNs are successful in many sequence modeling tasks, and LSTM is the key to success. LSTM focuses on solving the long-term dependencies problem in R NN compared to the standard RNN model. Bengio's paper confirms that in a standard RNN, the information performance of the long term gets worse as the distance increases. The hidden layer of an LSTM is a gated cell, which makes the architectures of RNNs and LSTMs distinct. LSTMs are a different structure from RNNs. There are four interacting especially. Each base RNN contains a gate of output, a gate of input, and a gate of forget, which together make up the

DOI: 10.54254/2755-2721/6/20230696

LSTM. Gates are a way to let information through optionally. In addition, long-term cellular memory states and candidate states are waiting to be stored. The concept of cell state has been added to the LSTM, and each cell can add or delete information. Forget gate decides how to delete information, the external input gate determines updated information, and the output gate outputs information for the cell. The weights of the LSTM are subject to change, thus avoiding the problem of learning time or gradient explosion that occurs in standard RNNs.

Forget Gate Layer: Forget Gate Layer is to control what new information to store in the neural network. When the data is entered into the cell, the algorithm inside the cell will filter the input data, removing some unwanted information data. Specifically, the data that meets the algorithm rules are retained, and the data that does not conform to the algorithm rules are zeroed to remove this part of the data.

Given the current moment t, $h_{t-1}$is the hidden state that the cell receives at the previous moment. At the same time, the data x_t is entered into the cell and it's passed down after the algorithm updates the data. The resulting data is calculated again with the sigmoid function. At this point it gets a value in [0,1]. 0 means to discard all information or completely keep this. 1 means to receive and pass all information down or completely get rid of this. By combining the values with s_{t-1}, the cell state at the previous moment, it gets the specific gravity of the information passed down in the previous cell state. The forget gate formula representation in formula (1). b_f is the bias term, U_t is the input weight, and U_t is the cyclic weight of the forget gate.

$$f_t = \sigma(W_f \cdot h_{t-1} + U_f \cdot x_t + b_f) \tag{1}$$

Input Gate Layer: LSTM determines the updated data by entering the gate. The gate of input is used to update the state of the cell. It is divided into two parts. The sigmoid layer is the initial part. The decision of whether to update the data is made using it. The tanh layer is the second part. To supplement the present state, this layer generates a new candidate values $\widetilde{s_t}$.

$$g_t = \sigma(W_g \cdot h_{t-1} + U_g \cdot x_t + b_g) \tag{2}$$

$$\widetilde{s_t} = \tanh\,(W \cdot h_{t-1} + \textstyle\sum_j U \cdot x_t + b) \tag{3}$$

The gate of forget and the gate of input will update the cell state. The updated cell state is shown in Equation 4. It inputs s_{t-1} and then output s_t.

$$s_t = f_t \cdot s_{t-1} + g_t \cdot \widetilde{s_t} \tag{4}$$

Output Gate Layer: The first step is the sigmoid layer. This layer defines which part of the cell state it wants to output. In the second step, let the cell state go through the tanh layer so that the output is between -1 and 1.

$$q_t = \sigma(W_q \cdot h_{t-1} + U_q \cdot x_t + b_g) \tag{5}$$

Then multiplying them together to get the end result s_t.

$$h_t = q_t \cdot tanh(s_{t-1}) \tag{6}$$

2.2. Gated recurrent units

Gated Recurrent Units are the new generation of RNN, evolved from LSTM. GRU is somewhat similar in structure to LSTM in the cell [8-10]. GRU does not use the cell state but instead uses the hidden state to transmit information. Unlike LSTM, which has three gates, the reset and update gates are all that GRU has. With only two gates, GRU has fewer parameters than LSTM, simplifying the calculation process and speeding up training.

GRU does not use the linear self-updating memory cell of the LSTM but instead uses the door to update automatically. LSTM uses the update gate to control how much new memory is added independently, also called cell state, which is the same as the old Cell state is irrelevant. But GRU's

DOI: 10.54254/2755-2721/6/20230696

newly added cell state will be limited by the old cell state. When the old cell the more states, fewer cell states are added later. The specific schematic diagram of GRU is shown below.

We define h_t as memory cell containing the previous information h_{t-1} and the current information $\widetilde{h_t}$ which is also called hiding state information. The expression for $\widetilde{h_t}$ and h_t are as follows.

$$\widetilde{h_t} = \tanh\,(W \cdot [r_t \cdot h_{t-1}, x_t]) \tag{7}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \widetilde{h_t} \tag{8}$$

Update Gate Layer: In the LSTM, the forget gate and the input gate are combined into the update gate. It is used to calculate the maximum amount of cell state data that can be stored, including data on both the current and prior cell states.

$$z_t = \sigma(W_z \cdot h_{t-1} + U_z \cdot x_t) \tag{9}$$

Reset Gate Layer: This gate determines how much old data or infomation should be forgotten. The results of update gate and reset gate are between 0 and 1.

$$r_t = \sigma(W_r \cdot h_{t-1} + U_r \cdot x_t) \tag{10}$$

After the GRU filters the data, the final data is output by the following equation.

$$y_t = \sigma(W_o \cdot h_t) \tag{11}$$

Besides testing LSTM and GRU, this paper tests the derivative versions of the above two models, BiLSTM and BiGRU. Compared with the above two models, BiLSTM and BiGRU model time series in both directions, so they can learn more sequence characteristics.

3. Experimental results and analysis

The dataset all comes from the waizao website(waizaowang.com). The dataset selects the performance of 65 stocks in Shanghai, Shenzhen, and Beijing A shares in the past year, deletes some features for subsequent use, and finally retains 44 attributes. The dataset is then normalized. The training set to test set ratio is 7:3. The feature description as shown in Table 1.

Table 1. The display of some features.

Feature	Description
tdate	The time of stock trading
price	The latest price at the current time
zdfd	amount of increase and amount of decrease
Zded	Limit of ups and downs
ttmsyl	Trailing Twelve Months
zgj	The highest price
zdj	The lowest price
mgsy	Earning Per Share

In this paper, min-max standardization—a linear adjustment of the original data—is used to ensure that the results fall inside the [0,1] interval. Formula 4.1 illustrates the conversion function as follows:

$$x^* = \frac{x - min}{max - min} \tag{12}$$

Max means the maximum value, and min means the minimum value of the sample data.

DOI: 10.54254/2755-2721/6/20230696

3.1. Similarity matrix

Then the cosine similarity calculation is used on the dataset, if the cosine similarity is found for the $\vec{a}$ and $\vec{b}$, the result falls on [0-1], and the larger the result, the higher the similarity. As shown in Equation (13).

$$cos\theta = \frac{\vec{a}\times\vec{b}}{||\vec{a}|\times|\vec{b}||} \tag{13}$$

The results as shown in Figure 1.

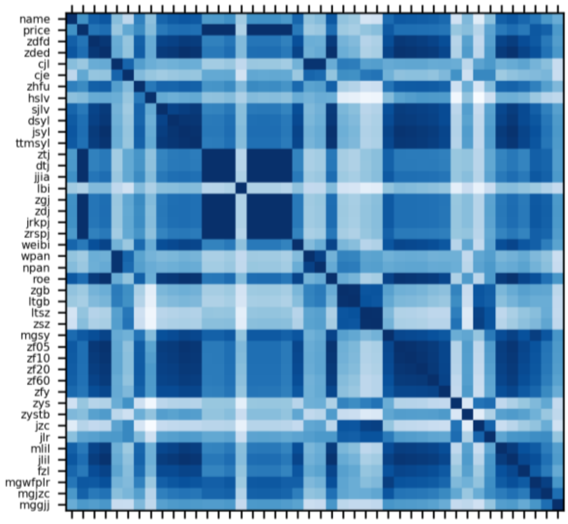

Figure 1. Results of cosine similarity.

The abscissa is a transpose of ordinates, and darker colors indicate the higher correlation between attributes.

3.2. Evaluation criteria

Root Mean Square Error (RMSE) is used for this experiment's evaluation criteria. The ratio of the squared variation of the anticipated prices from the actual prices to the ratio of n observations is known as the root mean square error, or RMSE. It measures the deviation between the forecast and actual prices and is sensitive to outliers in the data. The calculation formula:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(Y_i - f(x_i))^2} \tag{14}$$

where Y_i is the actual price, $f(x_i)$ is the forecast price.

3.3. Results and analysis

Four different models—LSTM (Long Short-Term Memory), GRU (Gate Recurrent Unit), BiLSTM (Bi-directional Long Short-Term Memory) and BiGRU(Bi-directional Gate Recurrent Unit)—are employed in this experiment to predict the most recent stock price. Inputs consist of 44 features, including the (latest) price. All experimental results are the result of 5000 epochs. Table 2 1) presents the experimental outcomes. The superiority of deep learning models. The dataset has been normalized earlier, so the error of 10^{-6} level is already deficient, which shows the superiority of deep learning models in handling such tasks. 2). BiLSTM and BiGRU performed better. Because of the better use of bidirectional information, the BiLSTM model is superior to the LSTM model, and the BiGRU model is better than the GRU model.

3). LSTM outperforms GRU. It can be seen that the RMSE of both LSTM and BiLSTM is smaller than GRU and BiGRU because GRU is a simplified version of LSTM.

Table 2. Experimental results of 4 models (with one feature).

Model	RMSE
LSTM	$6.9083*10^{-6}$
BiLSTM	$6.8108*10^{-6}$
GRU	$7.0001*10^{-6}$
BiGRU	$6.9234*10^{-6}$

From Table 2, the optimal choice has been BiLSTM and then based on BiLSTM, experiment 2 controls the number of input attributes. The test results are displayed in Table 3. One feature means that only the price attribute is used as input, and ten features mean that the first ten attributes in the dataset are used as input.

Table 3. Experimental results of 4 models (with multi-features).

Model	RMSE
BiLSTM (1 feature)	$6.8108*10^{-6}$
BiLSTM (10 feature)	$5.2508*10^{-6}$
BiLSTM (20 feature)	$4.7150*10^{-6}$
BiLSTM (30 feature)	$5.5562*10^{-6}$
BiLSTM (all feature)	$7.8468*10^{-6}$

It is evident from comparing the experimental findings that the number of feature inputs does not increase linearly connected to RMSE. In the first 20 features, RMSE gradually decreases as the number of features increases, while at 30 features and all features, the noise caused by the rich input of the model increases RMSE. Therefore, it is necessary to filter features according to methods such as similarity calculations, which is the next step of this paper.

4. Conclusion

Through the comprehensive analysis of the experimental results of the four models, the BiLSTM model has the best performance in this experiment. Compared with LSTM, BiLSTM can make full use of two-way information; Compared with BiGRU, BiLSTM has one more gate; Combining the above two advantages, BiLSTM makes the RMSE smaller and the result more accurate in predicting the latest stock price. However, in the choice of the number of features, it is not that the more features, the more accurate the result. Between 20-30 features, the advantage brought by adding features will be maximized. Because as the number of features grows, the influence of noise on the prediction results likewise grows, bringing the prediction model closer to reality. Therefore, it is necessary to screen features according to similarity calculation and other methods, which is also the next step of this paper.

References

[1] H.White. Economic prediction using neural networks: the case of IBM daily stock returns. Neural Networks, IEEE International Conference on. 1988.2(6). 451-458
[2] Gen Cay. R.Non-linear prediction of security returns with moving average rules.Journal of Forecasting,1996,15(3).43-46
[3] Rodriguez. F, Martel.C. On the profitability of technical trading rules based on artificial neural networks: Evidence form the madrid stock market. Economics Letters.2000.69(1).35-37
[4] G. Peter Zhang. Time series forecasting using a hybrid ARIMA and neural network model|J]. Neurocomputing.2003(50). 159-175
[5] A.Murat. Bahadir. Comparison of bayesian estimation and neural network model in stock market

DOI: 10.54254/2755-2721/6/20230696

trading.Intelligent Engineering Systems through Artificial Neural Networks.2011.20. 74-81

[6] A.Muratozbayoglu. Neural based techical analysis in stock market forecasting.Intelligent Engineering Systems through Artificial Neural Networks.2008(18). 261-265

[7] Lapedes, A., & Farber, R. (1987). Nonlinear signal processing using neural networks: Prediction and system modelling (No. LA-UR-87-2662; CONF-8706130-4).

[8] Fu, R., Zhang, Z., & Li, L. (2016, November). Using LSTM and GRU neural network methods for traffic flow prediction. In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC) (pp. 324-328). IEEE.

[9] Dey, R., & Salem, F. M. (2017, August). Gate-variants of gated recurrent unit (GRU) neural networks. In 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS) (pp. 1597-1600). IEEE.

[10] Bansal, T., Belanger, D., & McCallum, A. (2016, September). Ask the gru: Multi-task learning for deep text recommendations. In proceedings of the 10th ACM Conference on Recommender Systems (pp. 107-114).