An improved long short-term memory neural network for stock forecast

Liujia Lv[1,*], Weijian Kong[1,2], Jie Qi[1,2] and Jue Zhang[1,2]

[1] College of Information Science and Technology, Donghua University, Shanghai 201620, P.R.China
[2] Engineering Research Center of Digitized Textile and Fashion Technology, Ministry of Education, Donghua University, Shanghai 201620, P.R.China

Abstract. This paper presents an improved long short-term memory(LSTM) neural network based on particle swarm optimization (PSO), which is applied to predict the closing price of the stock. PSO is introduced to optimize the weights of the LSTM neural network, which reduces the prediction error. After preprocessing the historical data of the stock, including opening price, closing price, highest price, lowest price, and daily volume these five attributes, we train the LSTM by employing time series of the historical data. Finally, we apply the proposed LSTM to predict the closing price of the stock in the last two years. Compared with typical algorithms by simulation, we find the LSTM has better performance in reliability and adaptability, and the improved PSO-LSTM algorithm has better accuracy.

1 Introduction

The prosperity of the stock market transaction represents the economic status of a country. According to statistics, the total market value of China's stock market accounted for 146% of GDP in 2015, ranked first among developing countries. And the United States accounted for 174%, ranked first in developed countries. The risk and return of the stock market are relatively higher. The stock exchanges of various countries produce large amounts of data every day. Investment institutions also increasingly use data as the primary reference while investing. It has been proved that the stock price changes in the short term is predictable.

In order to predict the trend of stock price, researchers at home and abroad have applied a variety of statistical and econometric methods to the study of stock markets, such as exponential smoothing method[1]，multivariate regression method[2], ARIMA model (autoregressive moving average model)[3] and GARCH model (autoregressive conditional variance model)[4]. However, since there are many factors influencing the stock price and the influence mechanism is complicated, it is difficult to use mathematical model to explain it easily. Traditional statistical model[5] and econometric model[6] should first preprocess the data and change the non-stationary sequence to the stationary sequence, and the amount of data cannot be too large, so the results are not ideal.

Neural network learning[7] is a highly complex nonlinear artificial intelligence system. It is a supervised learning model that simulates the memory of the human brain. It has the ability of distributed storage, self-organization and self-adjustment. The above characteristics of neural network make it more suitable for dealing with problems which are unstable, complex and non-linear, the stock price prediction is such a problem.

Therefore, in the past decade or more, we have used neural network for data analysis and forecast of the stock and obtained a lot of results, especially the BP neural network. For example, the Google Financial Laboratory developed the "wavelet neural network stock market prediction test system"[8] to forecast the net value of equity funds.

Compared with traditional statistics and econometric prediction methods, neural network is better at learning. However, the traditional BP neural network[9] prediction model is not suitable for the problem of time series training and prediction theoretically, at the same time, there are still problems such as the number of input data is difficult to determine and it is easy to fall into local minimum values. As a new type of neural network, LSTM[10] has the concept of time series which can realize multiple input and output in sequence, self-connection between hidden layers. Its selective memory and timing internal influence characteristics are very suitable for the random non-stationary sequences such as stock price prediction.

In this paper, based on the stock price forecasting, we introduce the LSTM and the improved PSO-LSTM neural network models with the time series concept, the improved PSO-LSTM algorithm is using particle swarm optimization to optimize the weights of LSTM neural network so as to improve the accuracy of prediction. Compared with the BP neural network, they have better feasibility and accuracy for prediction. The innovation of this paper mainly lies in: (1) An improved neural network model based on particle swarm optimization is proposed, and its superiority is proved. (2) The LSTM neural network is innovatively used in stock forecasting.

2 The experimental models

LSTM (long short-term memory） neural network introduces the timing into the structure of network, which makes it more adaptable in time-series data analysis.

It includes three network structure layers (input layer, hidden layer and output layer). On the basis of this, we propose an improved LSTM prediction model based on particle swarm optimization to minimize prediction error. The network training mainly takes the hidden layer as the research object. First in the input layer, we define the original fault time sequence as $F_0 = (f_1, f_2, ..., f_n)$, the divided training set and test set can be expressed as $F_{tr} = (f_1, f_2, ...f_m)$ and $F_{te} = (f_{m+1}, f_{m+2}, ..., f_n)$, satisfaction constraint conditions are $m < n$ and $m, n \in N$. Then for the element f_t in the training set, we use the classical z-score[1] standardization formula to process (mean is 0, standard deviation is 1, denoted as zscore), the standardized set can be expressed as follows:

$$F_{tr}^{'} = (f_1^{'}, f_2^{'}, ..., f_m^{'}) \quad (1)$$

$$f_t^{'} = (f_t - \sum_{t=1}^{n}) / \sqrt{\sum_{t=1}^{n}((f_t - \sum_{t=1}^{n})^2 / n)} \quad (2)$$

$$1 \le t \le m \quad (3)$$

$$t \in N \quad (4)$$

2.1 LSTM (Long Short-Term Memory) neural network

The LSTM model is to replace the hidden layer of RNN cells with LSTM cells, so that they have capacity of long-term memory. The most widely used structure of LSTM model hidden layer cell is shown in Figure 1.

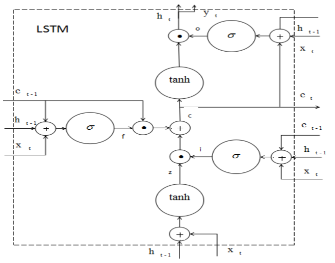

Figure1. The structure of LSTM model hidden layer cell

Its forward formulas are as follows:

$$i_t = \sigma(\omega_{xi} x_t + \omega_{hi} h_{t-1} + \omega_{ci} c_{t-1} + b_i) \quad (5)$$

$$f_t = \sigma(\omega_{xf} x_t + \omega_{hf} h_{t-1} + \omega_{cf} c_{t-`1} + b_f) \quad (6)$$

$$c_t = f_t c_{t-1} + i_t \tanh(\omega_{xc} x_t + \omega_{hc} h_{t-1} + b_c) \quad (7)$$

$$o_t = \sigma(\omega_{xo} x_t + \omega_{ho} h_{t-1} + \omega_{co} c_t + b_o) \quad (8)$$

$$h_t = o_t \tanh(c_t) \quad (9)$$

In the above formulas, i means the input gate, f means the forgotten door, c means the cell state, o means the output gate, ω is the corresponding weight coefficient matrices and b is bias term, σ, tanh are sigmoid and hyperbolic tangent activation functions. LSTM can add or delete information to cell through a gate control unit. These valves can selectively determine whether the information is passed. It takes the previous state of network as input to be calculated by the sigmoid function, if the result reach the threshold of the valve, it will multiply this result with current layer's calculation by elements. Then it will be the new input of next layer. If not, forget the output. The weight of each layer is updated during each model backpropagation training process.

The LSTM model training process uses a BPTT algorithm whose principle is similar to the classical back propagation (BP), it can be divided into four steps: (1) Calculate the output of LSTM cells according to the forward calculation method; (2) Reverse calculate the error of each LSTM cell, including time and network level these two reverse propagation directions; (3) Calculate the gradient of each weight according to the corresponding error term; (4) Update weights with gradient-based optimization algorithm;

The gradient-based optimization algorithm used in our LSTM model is adaptive momentum estimation algorithm[11]. It can compute adaptive learning rate for different parameters and occupy less storage resources. Adam has a better overall performance in practical applications than other stochastic optimization methods.

The whole framework of LSTM prediction model is shown in Figure 2, including input layer, hidden layer, output layer, network training and network prediction five functional modules. The input layer processes the original fault time series to meet the network input requirements. The hidden layer uses the LSTM cells in Figure 1 to construct a single-layer recurrent neural network. The output layer provides the prediction results. The network training uses the Adam optimization method. The network prediction uses the iterative method to forecast point by point.

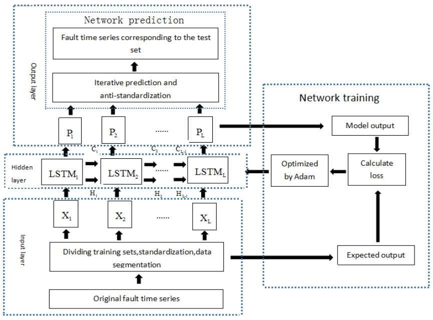

Figure 2. The whole framework of LSTM prediction model

We use data segmentation method to process $F_{tr}^{'}$, and set the split length value to L. The segmented model is as follows:

$$X = (X_1, X_2, ..., X_L) \tag{10}$$

$$X_p = (f'_p, f'_{p+1}, ... f'_{m-L+p-1}) \tag{11}$$

$$1 \le p \le L \tag{12}$$

$$p, L \in N \tag{13}$$

The corresponding theoretical output is:

$$Y = (Y_1, Y_2, ..., Y_L) \tag{14}$$

$$Y_p = (f'_{p+1}, f'_{p+2}, ..., f_{m-L+p}{}') \tag{15}$$

Next, we will enter x into the hidden layer, that the hidden layer contains L homogeneous LSTM cells which are connected by front and back moment, and the output of X after the hidden layer can be expressed as:

$$P = (P_1, P_2, ..., P_L) \tag{16}$$

$$P_p = LSTM_{fo}(X_p, C_{p-1}, H_{p-1}) \tag{17}$$

Among these, C_{p-1} and H_{p-1} are the state and output of the previous LSTM cell, $LSTM_{fo}$ is the forward calculation of LSTM cells, set the size of the vector of cell states is S_{st}, both size of the C_{p-1} and H_{p-1} are S_{st}, we can see that the hidden layer output P, the model input X and the theoretical output Y are all two-dimensional arrays with dimension $(m-L, L)$. Then we choose the mean square error as the error calculation formula, the loss function of the training process can be defined as :

$$loss = \sum_{i=1}^{L(m-L)} (p_i - y_i)^2 /(L(m-L)) \tag{18}$$

First, we set the minimum loss function as the optimization target, and set the network initialized random number of seed $seed$, the learning rate η and the training step length $steps$. Then we apply Adam optimization algorithm to continuously update the network weights to obtain the final hidden layer network.

The trained LSTM network $(LSTM^*_{net})$ is used to predict, and the prediction process is based on iterative method. First, the last line data of theoretical output Y is:

$$Y_f = (f'_{m-L+1}, f'_{m-L+2}, ..., f'_m) \tag{19}$$

Enter Y_f into $LSTM^*_{net}$, the result is:

$$P_f = LSTM^*_{net}(Y_f) = (p_{m-L+2}, p_{m-L+3}, ..., p_{m+1}) \tag{20}$$

The prediction value at moment $m+1$ is p_{m+1}, the last $L-1$ data points and p_{m+1} of Y_f are merged into a new row of data:

$$Y_{f+1} = (f'_{m-L+2}, f'_{m-L+3}, ..., p_{m+1}) \tag{21}$$

Enter the Y_{f+1} into $LSTM^*_{net}$, the prediction value at moment $m+2$ is p_{m+2}, in turn, the prediction sequence is:

$$p_o = (p_{m+1}, p_{m+2}, ..., p_n) \tag{22}$$

Next, through the z-score anti-standardization of $X(dez)$, the final prediction sequence corresponding to the test set is as follows:

$$P_{te} = dez(P_o) = (p^*_{m+1}, p^*_{m+2}, ..., p^*_n) \tag{23}$$

$$P^*_k = P_k \sqrt{\sum_{t=1}^{n} (f_t - \sum_{t=1}^{n} f_t / n)^2 / n + \sum_{t=1}^{n} f_t / n} \tag{24}$$

$$m + 1 \le k \le n \tag{25}$$

$$k \in N \tag{26}$$

Similarly, we make each row of X as a model input can obtain the fitting sequence P_{tr} corresponding to training set F_{tr}. Finally, the model fitting and prediction accuracy are obtained by calculating deviation quantitation F_{tr} and P_{tr}, F_{tr} and P_{te}.

2 2 LSTM neural network algorithm based on particle swarm optimization (PSO-LSTM)

It involves many parameters in constructing the LSTM prediction model，of which the weights are the most important. In order to achieve better prediction results, we use particle swarm optimization algorithm to optimize these parameters. PSO can avoid network convergence falling into the local optimal solution.

We take the weights of LSTM hidden layer as input of particle swarm. The initial output error of LSTM is used as fitness of particle swarm, then judge particle's performance according to condition. The random initial particle swarm updates its own parameter according to the individual extremum and global extremum.

In each iteration process, the formula for the particle to update its velocity and position is as follows:

$$V_{id}^{k+1} = \omega V_{id}^{k} + c_1 r_1 (P_{id}^{k} - X_{id}^{k}) + c_2 r_2 (P_{gd}^{k} - X_{gd}^{k}) \tag{27}$$

$$X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k+1} \tag{28}$$

In the above formulas, $d = 1,2,...,D$; $i = 1,2,...,n$; P_{id} is the individual extremum; P_{gd} is the global extremum; V_{id} is the velocity of particles; X_{id} is the position of the particle; ω is the inertia weight; k is the current iteration times; c_1, c_2 are learning factors; r_1, r_2 are random numbers that are distributed in [0,1].

When the PSO algorithm optimizes the LSTM algorithm, the value of the optimal particle position vector in the particle swarm is used in sequence as the initial value of each weight in the LSTM network . The dimension of each particle can be calculated according to the structure of the neural network model, and the mean square error of each output neuron of the given training set is taken as the fitness function of particle swarm. Then we calculate the fitness value of each particle according to the fitness function, the smaller the value is, the smaller the error of network output is. It also means the better performance of the corresponding particles. The particle position is continuously updated so that the error of the output layer of the network is gradually reduced. In each iteration, the particle with the smallest error is taken as the current optimal particle.

The PSO-LSTM flowchart is shown in Figure 3.

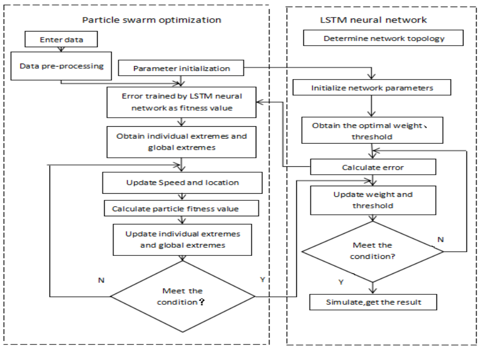
Particle swarm optimization
Enter data
Data pre-processing
Parameter initialization
Error trained by LSTM neural network as fitness value
Obtain individual extremes and global extremes
Update Speed and location
Calculate particle fitness value
Update individual extremes and global extremes
N
Y
Meet the condition?
LSTM neural network
Determine network topology
Initialize network parameters
Obtain the optimal weight、threshold
Calculate error
Update weight and threshold
Meet the condition?
N
Y
Simulate,get the result

Figure 3. The PSO-LSTM algorithm flowchart

The steps are as follows:

(1) We input training samples and test samples, then normalize them, finally determine the number of neurons in each layer of LSTM network;

(2) Determine the relevant parameters of the PSO and then determine the corresponding parameters of the LSTM network;

(3) Randomly generate initial particle position and velocity;

(4) Train the neural network and calculate the initial output error which is the fitness value of the particle swarm;

(5) Obtain the individual extreme and global extremum;

(6) Update the velocity and position of each particle;

(7) Whether meet the conditions for the PSO to stop the iteration, if it is, save the results and go to step (8), otherwise go to step (4);

(8) Take the global optimal value of PSO as the weights of LSTM neural network;

(9) Using LSTM algorithm to train neural network and judge whether meet the conditions for the LSTM to stop the iteration, if not, repeat the steps from (4) to (9), otherwise, save the results and go to step (10);

(10) Simulate with the test sample and get the output.

In this process, the main combination steps of the two algorithms are as follows: (1) The parameters of PSO determine the relevant parameters of LSTM network; (2) The output error of the latter is taken as the fitness value of the former; (3) The optimal solution obtained by PSO is set as the training parameters of neural network. So that the PSO-LSTM model can have advantages of two algorithms, it not only has better global search ability but also better local search ability.

According to the experience and experimental comparison, we set the particle size to 60, the maximum number of iterations to 100, the inertia weight to 0.5, the acceleration factors c_1 and c_2 to 1.5, the position restriction interval to [-5, 5], and the speed restriction interval to [-1, 1]. We use the output error of LSTM neural network model as the fitness function of particle swarm optimization, as the time of iterations increases, the function value becomes smaller and smaller, and the individual's fitness will be higher and higher.

3 Experimental data

We select the daily stock data of the SP 500 from January 2, 2007 to January 2, 2017. The data includes the time, opening price, closing price, highest price, lowest price, and daily volume these six most critical attributes.

We set the time as the index of the time series and store the remaining five attributes as independent variables in the database, then input them to the LSTM and PSO-LSTM neural network models for training prediction.

The prediction method is to use the data of the first 10 trading days to predict the closing price data of the 11th day, this is due to the fact that more than 10 days of data have less impact on the 11th day of data, and the maximum probability will affect the accuracy of the prediction. Therefore, in the training of the LSTM neural network, the ten year data is divided into pieces each includes ten days, input 10 days' data to predict the closing price data of the 11th day and compare with the original data. The mainly training mode is to use the designed random function to disrupt the 10-year data set three times, so that the piece and piece are completely independent and have no correlation. We take the first 80% of the dataset for training, the remaining 20% for prediction.

This paper mainly compare and analyse the abilities of BP Neural network, LSTM Neural Network and PSO-LSTM Neural network in stock price prediction, we fit the predict results and actual results of stock, and calculate the errors of the three models.

4 The experimental simulation and result analysis

First of all, we use the classical z-score standardization formula (mean is 0, standard deviation is 1, denoted as zscore) normalize the data and control the data dimension between 0 and 1.

In this experiment, there is some difference in the input and output of the data. When using the BP neural network to predict, we input data includes five independent variables which are the opening price, the closing price, the highest price, the lowest price, and the daily trading volume. However, when we use the LSTM and PSO-LSTM neural network to predict, the data of input except for the first set of data includes the opening price, the closing price, the highest price, the lowest price, and the daily trading volume these five parameters, the remaining nine sets input only includes the opening price, the highest price, the lowest price, and the daily trading volume these four parameters, do not include the closing price.

In the output, the BP neural network is an input corresponding to an output, while the LSTM and PSO-LSTM are inputing the data from the 1th to 10th days to predict the data from 2nd to 11th days (the closing price of 11th day is predicted by the first 10 days). This is because there is no concept of timing in the BP neural network, so it is necessary to add the closing price to the training prediction in order to influence the prediction of the next day. However, in the training prediction of LSTM and PSO-LSTM neural networks, there is no need to enter the closing price because it has concept of timing, and it can convery the previous day's forecast to the next day through the hidden layer, so it is not necessary to input the closing price except the first day.

4.1 The prediction of the three models

We use the trained models to predict the last two years’ data and fit the predicted values with actual values into a graph. The blue curve is the predicted value, the red curve is the actual value. In order to make the experimental result more obvious, we enlarge the graph and select the 100-day result, a 100-day fitting chart of the BP Neural network is shown in figure 4.

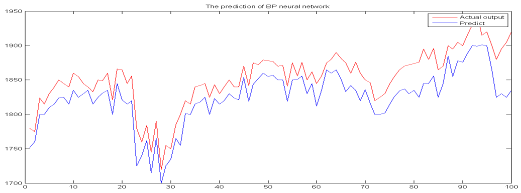

Figure 4. The prediction of BP neural network

A 100-day fitting chart of the LSTM Neural network is shown in figure 5.

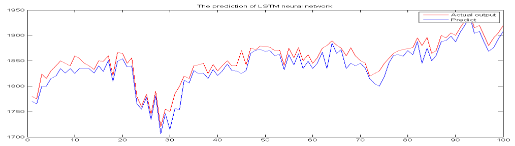

Figure 5. The prediction of LSTM neural network

A 100-day fitting chart of the PSO-LSTM Neural network is shown in figure 6.

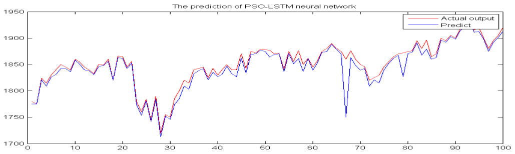

Figure 6. The prediction of PSO-LSTM neural network

It can be seen from the graph that the prediction results of LSTM and PSO-LSTM models are more accurate than BP neural network. Using particle swarm to optimizate neural network weights is valid.

4.2 The error of the three models

We calculate the error between predicted values and actual values, the errors of them are compared in figure 7.

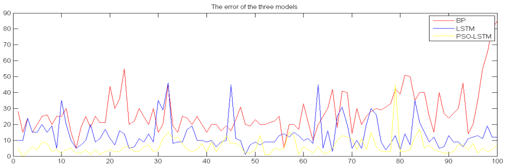

Figure 7. The error of the three models

Through calculating, we can see that the average error of the LSTM neural network prediction is about 7.3% (the data took as the error research object is from January 2015 to January 2017), the average error of the PSO-LSTM is about 5.9%, while the average error of the BP is about 14.7%, therefore the LSTM neural network is much more accurate than BP, and the improved PSO-LSTM algorithm has the highest accuracy.

5 Conclusion

This paper proposes an improved LSTM neural network algorithm which use PSO to optimize the parameters of LSTM. It takes the output error of the neural network as the fitness of PSO to find the optimal weights for the LSTM neural network. Then we set the global optimal value as the initial，in order to achieve the purpose of optimizing the LSTM neural network by using the PSO algorithm, thus establish a neural network model based on the PSO-LSTM algorithm. The experiment shows that the PSO-LSTM algorithm has better performance and higher accuracy.

References

1. E. L. De Faria, Marcelo P. Albuquerque, J. L. Gonzalez, J. T. P. Cavalcante, and Marcio P. Albuquerque. Predicting the brazilian stock market through neural networks and adaptive exponential smoothing methods. Expert Systems with Applications, **36**(10): 12506-12509, 2015.
2. R W Rebello and Y V Reddy. Multivariate regression: A tool for forecasting stock prices. *Iup Journal of Accounting Research Audit Practices*, ix: 7-32, 2010.
3. Yichen Dong, Siyi Li, and Xueqin Gong. Time series analysis: An application of arima model in stock price forecasting. In International Conference on Innovations in Economic Management and Social Science, 2017.
4. Bhowmik Roni, Chao Wu, Roy Kumar Jewel, and Shouyang Wang. A study on the volatility of the bangladesh stock market based on garch type models. *Journal of Systems Science Information*, **5**, 2017.
5. H. U. Zhongyi, Yukun Bao, Chiong Raymond, and Tao Xiong. Profit guided or statistical error guided a study of stock index forecasting using support

vector regression. *Journal of Systems Science Complexity*, **30**(6): 1-18, 2017.

6. Xiaoxia Xie. Research on the explanation power of accounting information of annual report of listed companies of stock pricebased on econometric model concept of empirical accounting theory. In *International Conference on Business Management and Electronic Information*, pages 588-592, 2011.
7. Kazuhiro Kohara, Yoshimi Fukuhara, and Yukihiro Nakamura. Selective prasentation learning for neural network forecasting of stock markets. *Neural Computing Applications*, **4**(3): 143-148, 2016.
8. Yu Fang, Kamaladdin Fataliyev, Lipo wang, Xiuju Fu, and Yaoli Wang. Improving the genetic-algorithm-optimized wavelet neural network for stock market prediction. In *International Joint Conference on Neural Networks*, pages 3038-3042, 2014.
9. Xing Chen, Lifang Wu, and Fuming Wang. The research of stock prediction based on ga-bp neural network. *Shanxi Electronic Technology*, 2014.
10. Kai Sheng Tai, Richard Socher, and Christopher D.Manning. Improved semantic representations from tree-structured long short-term memory networks. *Computer Science*, **5**(1): 36, 2015.
11. Hongmei Shao and Gaofeng Zheng. Convergence analysis of a back-propagation algorithms with adaptive momentum. Neurocomputing, **74**(5): 749-752, 2011.