LSTM model optimization on stock price forecasting

Yifeng Wang Yuying Liu Meiqing Wang Rong Liu

College of Mathematics and Computer Science, Fuzhou University, Fuzhou, 350116, China

Email:WangYifeng_AI@163.com

***Abstract*—In this paper, we mainly study the application of Long Short-Term Memory (LSTM) algorithms in the stock market. LSTM originates from the recurrent neural network (RNN) and has a significant effect on the time series problems. In this paper, the BP neural network model and the LSTM model are established respectively. Then we combine them with the stock data, a series of prediction results are obtained. Obviously, the prediction results of LSTM model are more accurate, and the prediction accuracy rate can reach 60%-65%. In the modeling process, in order to solve the 'saw-tooth phenomenon' of the gradient descent algorithm which is inevitable, we have improved the traditional gradient descent algorithm and specially designed the input data of the neural network. In addition, we defined a parameter combination library and use the skill of dropout to get the more ideal prediction results.**

Keywords:LSTM; BP neural network; stock forecasting

I. INTRODUCTION

The stock market has a strong regularity, but it is difficult to predict due to a large number of factors. In the past few decades, many algorithms have been applied to stock market prediction, but it is difficult to have a definitive model. Since the stock market is a complex asset complex, traditional econometrics and statistics models are not suitable for stock prediction problems [1]. But with the rapid development of artificial intelligence field, stock forecast base on BP neural network has indeed achieved certain results [2], the accuracy can achieve 50%-55%. Under some circumstance, it can even hit 56%-58%. Moreover, the RNN and LSTM developed on the basis of BP neural network, have unique advantages in predicting time series! The accuracy of stock prediction using LSTM at home and abroad can be stable at about 60%.

In this paper, wavelet analysis method is included to denoise the stock data. Meanwhile, a lot of improvements are made to the gradient descent function of LSTM and BP models, and a new data input method and training method are proposed for the LSTM model. We also study the selection of neural network parameters, and establish a 'parameter database'. The accuracy of the model which is proposed in this paper can be stabilized at 60%-65%.we not only use the TensorFlow library to implement the algorithm, but also re-program the algorithm based on the mathematics of the algorithm and implements the algorithm. Despite the heavy workload, the advantage of the latter is that it can fully integrate the practical problems with the mathematical principles, with more pertinence, smaller memory footprint, more opportunities for design and optimization. And its forecasting results can be compared with the results achieved with TensorFlow, and ultimately get an optimal result.

II. STOCK FORECASTING BASED ON LSTM MODEL

All the stock forecasts described in this paper are intended to predict the closing price on the next trade day. This chapter adopts three models and uses different combinations of parameters to forecast multiple stocks. Because these stocks belong to different fields and their influencing factors are different, the stock prediction difficulty is also varied.

A. BP Neural Network

1) Establishment and Solution of BP Algorithm Model

In machine learning, BP neural network algorithms are widely applied. Generally, the data set is divided into three groups—training set, validation set, and test set. The neural network is trained on the training set and tested with the test set. When the performance of the neural network on the validation set degrades, the training is stopped. This method we adopt can prevent 'overfitting' from happening effectively. The BP neural network model we constructed has 4 layers, which is: one input layer、two hidden layers and one output layer.

In this paper, we choose daily data after the previous restoration as the stock data, and the length of time varies from 3 to 6 years. Specifically, the daily data consists of six dimensions, which are: open-price、high-price、low-price、close-price、volume and amount. Daily data of stock A on day k is regarded as the input of BP neural network: $\mathbf{x_k} = (open_k, high_k, close_k, low_k, volume_k, amount_k)$. Then output is $\mathbf{y_{k+1}}$, the closing price of day k+1,then $(\mathbf{x_k}, \mathbf{y_{k+1}})$ can train the neural network for one time.

This paper utilizes more than 10 different stocks and data ranging from 3 years to 6 years to test the model performance. For example, we select stock 'Yunnan Baiyao' (stock code 000538) as forecast target and apply 1400 days stock data for training and next 100 days data for testing. Because the BP neural network may easily fall into a local minimum, the predictive value may appear to drift slightly in the average line during the training process, we call this phenomenon 'not dare to predict' in this paper, as shown in Figure 2-1.

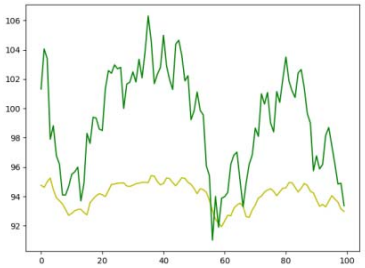

Figure 2-1 BP neural network predicts the closing price of 'Yunnan Baiyao' stock

978-1-5386-7445-1/18/$31.00 ©2018 IEEE
DOI 10.1109/DCABES.2018.00052

While the number of iterations is very large, the prediction results will be close to the actual value. Generally speaking, when the performance of the neural network is degraded on the validation set, the number of iterations has often ranged from 400,000 to 500,000.For example, training and testing performance for the stock 'Yunnan Baiyao' are shown below:

Table 2-1 BP neural network prediction accuracy of fluctuations

Data Set	Forecast Accuracy
Training Set	56.3%
Validation Set	61.2%
Test Set	57.6%

When the training stopped, a total of 435,768 iterations were experienced during the training. Although the forecasting value is very close to actual value, it can be found that the forecasting performance is not satisfactory when referring to the accuracy of the next trade day's price change rate. Because the neural network stops training when the performance on the validation set declines, the accuracy of the neural network in the validation set is generally higher than that of training set. After many trials, the prediction accuracy of the stock closing price for stock 000536 can be stabilized at 98%-99%, the prediction accuracy of the stock change of 000538 basically maintained at 55% -58%.

2) Analysis of BP Neural Network Model

Because a three-layer neural network can fit any nonlinear continuous function with arbitrary precision [4], it's very suitable for complex systems, such as the stock market. However, this algorithm is very easy to converge to a local minimum. Although we have improved it by combining random gradient descent methods, it still be impossible to avoid experiencing several local minima in training sessions, which is one of the reasons for the need of a great scale of iterations.

This paper adopts the non-precise linear search method, in the later iterations, the 'saw-tooth phenomenon' becomes extremely noticeable, which makes the BP algorithm inefficient. Besides, because the objective function is complex, when the output of the neuron is close to the actual value in the later iterations, some 'flat- areas' will inevitably appear. In these areas, the weight parameter changes very little each time, thus the training almost stopped. All these are important reasons for the excessive number of iterations of BP neural networks.

What's more, the stock data is of strong time correlation, and the stock movement in the previous time period hides some important market characteristics. However, BP neural network model only uses stock data as some completely independent input which cause training performance and training speed inevitably greatly reduced. Therefore, in order to make full use of its time correlation and trend characteristics and improve its convergence speed as much as possible, this paper further propose another forecasting model based on Long Short-Term Memory neural network (LSTM).

B. One-dimensional Long-Short Term Memory(LSTM)

1) recurrent neural network(RNN)

LSTM evolved from recurrent neural network (RNN). The RNN loops the regular BP neural network several times and effectively utilizes the previous output, which can describe dynamic time behavior. Compared with BP algorithm, RNN can make better use of its internal memory to process the input of any time sequence. Unlike the BP network, RNN algorithm has to perform multiple inputs during one forward calculation, and the input is divided into two parts. One part is the element in prepared training sample. The other part is the calculation result. One calculation is completed after many recurrences. Chained features imply that the nature of RNN is to dealing with continuous data, especially for sequences and lists. Therefore RNN and LSTM are especially suitable for handling the above types of data.

However, there are also some problems in the application of RNN. The main problems are the gradient explosion and the disappearance of gradients. The RNN is composed of multiple BP neural networks. Therefore, the calculation requires multiple activation functions, which are tanh or sigmoid functions. They limit the function value between [-1,1] and [0,1] respectively, and limit their derivatives between [0,1] and [0,0.25].Therefore, error E_t propagates along time t will cause its value to go to 0,which leads to the gradient to disappear; for the same reason, error E_t propagates will also lead to a sharp increase in its value, which causes the gradient explosion.

2) Establishment of LSTM algorithm model

For gradient explosion and gradient disappearance problem, changing activation function or the method of setting threshold has little effect on improving the training performance, and the effect of data time correlation is not exploited. Compared with the traditional RNN, in addition to having more BP neural network modules, the Long Short-Term Memory neural network also increases a cell state C_t which memorizes information transmitted over time. During transmission, the information of cell at the current moment is determined by the output of the previous cell and the current input. The gate structure is trained by the BP neural network layer and can control the degree of addition or deletion of information.

We input training data (**x**,**y**) into LSTM model, completing the forward calculation, then get the output value of the model, denoted **Y**. Let E be the variance of the output value and the true value, which is:

$$E = \frac{1}{2}\sum_{j=1}^{l}(Y_j - y_j)^2$$

It is our objective function. Just as the adjusting parameters ways of BP neural network, LSTM calculates the gradient decline direction of:

W_{hf}、W_{xf}、b_f、W_{hi}、W_{xi}、b_i、W_{hc}、W_{xc}、b_c、W_{ho}、W_{xo}、b_o、W_y、b_y

According to the objective function, by solving the back propagation of the error along the time, the formulae of each parameter iterative in LSTM is obtained.

LSTM overcomes the gradient disappearing problem. Since the essence of RNN is to maintain a state C_t within its structure where t represents time and C_t can be calculated by recursion. The traditional RNN always uses the 'overwrite' method to calculate the state. Based on error function to determine the weight parameters according to the chain rule would directly cause the gradient to be expressed as a continuous product.

Therefore, the gradient disappears, that is, a large number of items less than 1 are multiplied with each other, and the result is close to 0; The LSTM employs the 'accumulation' method to calculate the state so that its derivative also has an accumulated form. In summary, LSTM models effectively avoid the gradient disappearance problem. Besides, the concept of 'cell memory' has been introduced to LSTM, which makes it more natural to handle long-term sequences and can make full use of the temporal correlation of time series.

3) One-Dimensional LSTM Stock Model

The training data used in this section is basically the same as the previous BP model. The only difference is that since the LSTM model established in this section is one-dimensional, that is, the input training data is one-dimensional, so we only select 'close price' as input from the stock daily data(opening price, closing price, highest price, lowest price, trading volume, transaction amount).

Selecting L+1days stock closing price as input, then generates a vector which has the shape $1\times L$ from the previous L-day data as the input vector: $\mathbf{X}=(close_1,close_2\ldots close_L)$. Set the number of time nodes t (t divides L) and truncate $\mathbf{X}$ to t segments to form t vectors $\mathbf{x_1},\mathbf{x_2}\ldots\mathbf{x_t}$ as the input for each time node. At the same time, use $\mathbf{x_2}(1)$, $\mathbf{x_3}(1)\ldots\mathbf{x_t}(1)$ and data of the (L+1)day as the output vector $\mathbf{Y}$, which has the shape$1\times t$,as the figure2-2 shows. Put $\mathbf{X}$ and $\mathbf{Y}$ into the LSTM model. Perform several trainings, after the training is completed, when the stock is closed for L consecutive days, the model will make a prediction on the L+1th day's closing price. Let the LSTM memory cell dimension be mem, and the input vector at each time node $\mathbf{x_i}$'s dimension is x_dim(x_dim=L/t). Generally, it is more appropriate to have mem slightly larger than x_dim. Then we constantly adjust the parameters and apply several stocks' data to analysis model. Among them, the forecast results for stock real estate index (SH000006) are as follows:

Table 2-2 Real estate index forecast results

Accuracy	L	x_dim	mem	Number of iterations
Validation set:58%, Test set:56%	1600	400	500	5000
Validation set:57%, Test set:54%	1600	320	400	8000
Validation set:61%, Test set:59%	500	250	150	3000
Validation set:65%, Test set:62%	500	100	100	4000
Validation set:59%, Test set:54%	100	25	30	1000
Validation set:54%, Test set:51%	100	10	15	1000

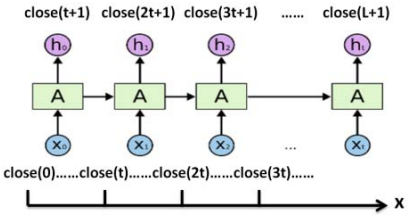

Figure 2-2 Construction of Input and Output of One-Dimensional LSTM Model

Table 2-2 shows that the impact of parameters on the one-dimensional LSTM model. When the training set becomes larger, the training time span increases(that is, L increases),the number of time nodes increases(that is. t increases), the performance of the model does not rise but decreases, although compared with the BP model, the model performance has been greatly improved and the number of iterations also dropped significantly. However, we should also make more use of stock daily data, i.e. the other five dimensions of daily stock information. A multi-dimensional LSTM model is established.

C. Multidimensional (LSTM) model

Since the one-dimensional LSTM model only make little use of stock daily data, and the time span of data is short (In section Ⅱ.C, when time span exceeds 500 days, the model performance has dropped sharply). So in this section we will make some model improvements and add the dimensions of input data.

1) Input data

Select L+1days stock daily data as input, which consists of six dimensions of characteristics, rather than just the 'closing price' feature. Set the time period to t, and process the data of the previous L days as follows: select data from day 1 to day t to constitute a matrix $\mathbf{x_1}$. Similarly, the data from the 2th day to the t+1th day is selected to form a matrix $\mathbf{x_2}$……Select the data from day L-t to day L from L to form a matrix $\mathbf{x_{L-t}}$. As above, a total of L-t input matrices are constructed. Each input data is shifted backward by the previous input data one day, and the input matrix $\mathbf{x_i}$ has the shape of $(t\times 6)$.

2) Output data

Corresponding to the input data, the output data is the closing price of the next trade day. Therefore, in the data of length L+1 days, the data from the 2th day to the t+1th day is selected to constitute the vector $\mathbf{y_1}$. Similarly, select the data from day 3 to day t+2 to form vector $\mathbf{y_2}$…Select the data from the L+1-th day to the L+1th day to form vector $\mathbf{y_{L-t}}$. The output vector has the shape of $t\times 1$.

In summary, the training set can be written as:

$$\{(\mathbf{x_1},\mathbf{y_1}),(\mathbf{x_2},\mathbf{y_2}),(\mathbf{x_3},\mathbf{y_3})\cdots\cdots(\mathbf{x_{L-t}},\mathbf{y_{L-t}})\}$$

In the ith training process, input $(\mathbf{x_i},\mathbf{y_i})$ to the multi-dimensional LSTM model, as the figure2-3 shows:

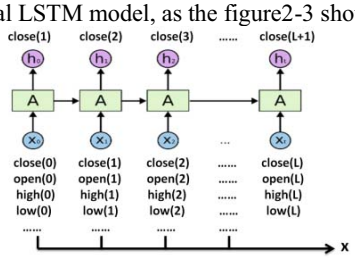

Figure 2-3 The Input and Output of Multidimensional LSTM Model

During training, it can be divided into several batches to increase training speed and training performance. After above work is done, the model will make a prediction on the Lth day closing price of the stock when the L-days stock data is entered continuously in this model.

3) results and analysis

Select several stocks, constantly adjust the parameters and the number of iterations. Actually the implementation of the asset allocation is more concerned about the accuracy of stock price change rates. The accuracy rate shown below is the accuracy of model forecasting value for the next trade day's closing price changes: If the forecast result states that stock will rise (fall) and the stock indeed rise (fall), then the forecast is accurate.

In order to improve the accuracy, we propose a 'threshold' filtering method. Set the threshold be a, the prediction value of LSTM be b. we only take model forecasting results into accuracy statistics if this restriction is satisfied: $|b| \geqslant a$. The essence of this 'threshold' method is that we only focus on the large forecasting values of LSTM model .In this way, the model accuracy is improved. It is not difficult to conclude that LSTM model has higher prediction accuracy on forecasting large stock change. However, if the threshold is too large, the number of predictions left is too small to be thought convincing. After many experiments, the threshold is set to 1%.

The LSTM was trained based on 1400 days data from January 2012 to May 2017 and test based on 150 days data from June 2017 to 2018. The model performance is displayed as follows:

Table 2-3 Multidimensional LSTM Multi-stock Forecast Results

	000538	000069	000895
T=150, N=10, iterations:500	58%, 62%	52.7%, 59.3%	54.3%, 56.2%
T=150, N=10, iterations:1000	60.8%, 70.8%	57.4%, 70%	56.1%, 65.8%
T=150, N=10, iterations:3000	65.5%, 71%	54.1%, 60%	57.4%, 61.6%
T=150, N=10, iterations:10000	57.4%, 62%	57.1%, 62%	52.4%, 58%

T=150 indicates that the number of time nodes in the model is 150, that is, the cell structure is repeated 150 times; N=10 indicates that the number of hidden nodes in each BP neural network layer is 10. It can be seen from the table that the prediction of 'Yunnan Baiyao' stock by the model is more accurate. There are many reasons for this result. They will be analyzed in the later chapters. Have stock 'Yunnan Baiyao' stock as model test data and we change parameters or the training set of LSTM model, then make predictions for next 250 trade days. The prediction performance are shown as follows:

Table 2-4 Multi-dimensional LSTM model performance for 'Yunnan Baiyao'

	2012-2018					2009-2017	2010-2016	1998-2006
	T=100 N=10	T=150 N=30	T=200 N=60	T=250 N=100	T=250 N=10	T=250 N=10	T=250 N=10	T=250 N=10
iterations: 1000	57.6% 63.4%	62.1% 66.3%	59.3% 65.2%	57.4% 61.5%	64.7% 72.5%	59.7% 75.6%	59.1% 63.2%	60% 61.8%
iterations: 3000	61% 64.5%	63.4% 60.1%	66.4% 69.1%	59.1% 63.8%	65.5% 69.3%	58.3% 65.1%	63.7% 72.5%	58.3% 62%
iterations: 10000	54.3% 58%	57.6% 58.5%	58.5% 61.1%	61.6% 57.1%	64.3% 66.6%	55% 63.8%	57.7% 66.7%	56.5% 55.6%

Table 2-4 shows that the number of hidden layer nodes should not be too large. In the case that input dimension is 6, the number of hidden layer nodes should be 6-12. Meanwhile, based on multi-dimensional information, enough time nodes can enhance the model performance, especially facing stock data with complex time correlation. If model lack sufficient time nodes, it may not be able to learn long-term trend from data. So 200-250 can be reasonable. In addition, the optimal number of iterations varies around 2000-6000 and increase with the number of time nodes. Figure 2-4 represents part of the prediction performance based on training set of 2010-2016 and 2009-2017 with the parameters set to T=250, N=10.The red line is the data value and the blue line is the forecasting value. In general, if the parameters are reasonable, only based on stock daily data, the accuracy of forecasting for a particular stock can be stabilized at about 60% to 65%. If we adopt threshold filtering method, the accuracy can reach 66%-72%.

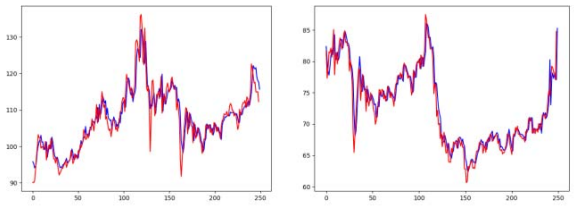

Figure 2-4 Multi-dimensional LSTM prediction results for 2010-2016(left) and 2009-2017(right), T=250, N=10

D. The selection of training stocks

Table 2-5 BP neural network forecast performance of some stocks in Shanghai and Shenzhen stock markets

parameter settings		Shanghai Stock Exchange			Shenzhen Stock Exchange		
Number of nodes	Number of iterations	600900	600016	600006	000402	000725	000651
5	1000	50.2%	52.1%	50.9%	52.4%	52.3%	51.5%
	10000	52.4%	53.2%	51.7%	53.1%	54.5%	52.4%
10	1000	51.8%	53.1%	52.4%	53.5%	51.7%	53.9%
	10000	51%	54.4%	53.6%	54.9%	53.6%	56.4%
15	1000	50.6%	52.8%	52%	51.1%	54.7%	52.8%
	10000	51.3%	53.2%	52.7%	52.4%	55.1%	53.4%

As we can see from the table 2-5, stocks of Shenzhen Stock Market has stronger regularity and higher prediction accuracy than those of Shanghai Stock Exchange. Therefore, the focus of this paper is stocks in Shenzhen stock market. In particular, the prediction accuracy of stock 'Yunnan Baiyao' stands out, we mainly concentrates on its prediction and investment.

III. MODEL OPTIMIZATION

A. Parameters Parameter combination library

The neural network model performance is closely related to the selection of its parameters, such as the number of hidden layers, the number of hidden layer nodes, the learning rate etc. It is generally believed that increasing the number of hidden layers can reduce training errors and improve training performance, but it will increase the time of iteration. Consequently, the selection of the number of hidden layer is extremely important. However, there is no scientific and universal method for determining it [5]. So far, many theories have been proposed for the case that the model has sufficient training data [6], which is difficult to achieve in practice[7]. From experiment results, it can be implied that excessive number of layers or nodes can also lead to model performance declined. So selecting a structure that is as compact as possible can effectively avoid the 'overfitting' in training and ensure good generalization ability.

In parameter adjustment, grid search method can greatly improve the model performance. However, it is not suitable under some circumstance for the calculation speed is very slow. Therefore, we choose to apply existing empirical formulas to determine the hidden layer and node numbers. Plenty of experiments at home and abroad demonstrate that it is relatively effective to adopt the following empirical formulas to determine the number of neural network hidden layer and nodes [8]:

$$m = \sqrt{n+l} + \alpha; \quad m = \log_2 n; \quad m = \sqrt{nl}$$

Where m is the number of hidden layer nodes, n is the number of input layer nodes, l is the number of output layer nodes, α is a constant between 1 and 10. Combined with several experiments in this paper, the model performs relatively better, when the parameters meets the above empirical formula. As for the learning rate parameter, it can be decided dynamically. Through the experiments, it is revealed that the learning rate (i.e., the step size) has a great relationship with the scale of the problem. As the problem scale becomes larger, the learning rate shall be smaller appropriately.

In summary, we have obtained an appropriate selection range of parameters and the effect of parameters' changes on the model. Because the hidden information from the time series is different of different stocks, it is not possible to have a set of parameter settings that are reasonable for all stocks. Therefore, the 'parameter combination library' is built to provide reasonable parameter combinations for different stocks neural network model.

B. Optimization of Gradient descent algorithm in LSTM

To reduce the number of iterations, this paper introduces the idea of stochastic gradient descent algorithm. Every time the model decides the direction of gradient descent, besides selecting based on the formula of BPTT algorithm, it can also be generated by randomly selecting with a certain probability. In this way, we not only reduce the amount of calculation, but also increase the speed of iteration and the probability of reaching the optimal point or jumping out of the local optimum when there is a local optimum. For the same purpose, the step size of iterations is also generated in the same way [3]. Furthermore, in the later stage of iterations, the error function would drop to a very small value, and the adjustment of the weight value is obtained by deriving the error function through t, thus each adjustment of the weight is minimal, training is almost stagnant. To solve this problem, we introduce the function of iterative step size and error:

$$L_k = \frac{1}{\sqrt[\alpha]{loss_k}}$$

Where $loss_k$ is the error of the k-th training, L_k is the step length of the k-th iteration, and α is a parameter. Since the error in the previous training process is generally greater than 1, the training effect is not sensitive to the adjustment of α. When loss is small, α can greatly control the range of step size. By doing this, the problem of option value updating slow in the late iterations can be overcome.

C. Optimization of training-dropout

Due to the fact that the neural network training speed becomes slower when the scale of the problem increases and it easily falls into local minimums (especially for BP neural networks) and it has inherent 'saw-tooth phenomenon' of the gradient descent algorithm, the neural network model demands a large number of iterations. In other words, although the performance of the neural network in the validation set does not decline, the training is close to stagnation and the improvement of the error is negligible. After the iteration is completed, the neural network has been 'overfitting'. So we introduce 'dropout' in neural networks. This approach is equivalent to forcing some hidden-layer nodes to cooperate with some randomly selected nodes rather than just matching a fixed "partner", which greatly weakening the joint adaptability between neuron nodes and greatly enhancing model's generalization. In the practically work, we make some neurons multiply 0 with a certain probability in the forward calculation so that they do not play any role in this training.

IV. CONCLUSION

In this paper, we construct the Long Short-Term Memory neural network model to have effective forecasting on the stock market, which accuracy can reach 60% - 65% or more.

At the same time, we have found that the functions in the TensorFlow library are not always optimal. From the perspective of training speed, the functions we write from the mathematics principle has a faster speed, and they can have more detailed designs or improvements under different situation. However, the functions in the TensorFlow library are more adaptable.

From our work, it can be inferred that there does exist some regular rules or patterns in the stock market and there are predictable stocks. This is also the reason why there are many private placement institutions in the market. The work done in this paper coincides with these agencies, so it has a broad market application prospects.

There is still much room for improvements in stock forecasting models. For example: optimization of neural network parameters, optimization of neural network training process…These will be the next step in the research.

REFERENCES

[1] E.F. FAMA, K.R. FRENCH. Dividend yields and expected stock returns [J].Journal of Financial Economics, 1988,22 (1): 3-26.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Z Jiang,D Xx,J Liang.A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem[J]. Papers. 2017.

[3] Junfeng Yao,Xianyong Yang,Xiaoqi Peng,Tian Zhang,Shunbin Zheng.Decreasing gradient optimization algorithm with variable step length based on chaotic variables[A].Journal of Tsinghua University(Science and Technology). 2003,43(12):1676-1678.

[4] Cybenko G. Approximation by superpositions of a sigmoidal function[J]. Analysis in Theory & Applications, 1989, 2(4):303-314.

[5] Qinglang Zhang,Xianming Li.A New Method to Determine the Number of Hidden Layer Nodes in Neural Networks[A].Journal of Jishou University(Nature Science Edition),2002,23(1):89-91.

[6] Grossberg S. Neural networks and natural intelligence[J]. Quarterly Review of Biology, 1988.

[7] Grossberg, Stephen. Nonlinear neural networks: Principles, mechanisms,and architectures[J].Neural Networks,1988,1(1):17-61.

[8] Liming Zhang.Artificial neural network model and its application[M].Fudan University Press,1993.