Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (22): 5384-5390, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.5384.5390

Neural Network Algorithm Based Method for Stock Price Trend Prediction

[1]Nan Ma, [2]Yun Zhai, [1]Wen-Fa Li, [3]Cui-Hua Li, [1]Shan-shan Wang and [1]Lin Zhou
[1]College of Information Technology, Beijing Union University, Beijing, 100101,China
[2]E-Government Research Center, Chinese Academy of Governance, Beijing, 100089, China
[3]Department of GSH Information-Based, Military Office in Beijing, Beijing, 100840, China

Abstract: Neural network algorithm is very suitable for stock prediction as a model for dealing with complicated relationship. However, the prediction accuracy of neural network algorithm depends largely on the number of hidden nodes and the terminal condition. To follow up the changes in stock prices, a new method is proposed in this study to find out the optimal parameter. The recommended solution is setting fewer hidden nodes and lower holdout percentage. Results show that the proposed method can lessen about 60% of the forecast error such that it can ensure the efficiency and accuracy of the algorithm.

Key words: Data mining, Neural network algorithm, stock prediction, hidden nodes

INTRODUCTION

Data mining is the key member of business intelligence and its main purpose is to extract the model from the existing data to improve the intrinsic value and refine the data into knowledge (Tang and Jamie, 2007).

With the rapid development of data mining, this kind of technology is used in business management, government offices, scientific research, engineering and many other areas.

There are many approaches in data mining, such as neural networks, decision trees, Bayes algorithm and so on. Besides, each algorithm has different scope of application and characteristics of itself.

In recent years, neural network has been widely used in time series analysis and financial forecasting. The reason is that neural network has very strong ability of nonlinear function approximation. It overcomes the gaps on processing data in traditional methods. Therefore it can be successful in forecasting application (Naskas and Papananos, 2004; Yi and Lv, 2009).

There are a mass of reliable and accurate raw data in the field of stock, while the data has virtual predictable meaning. If we can predict the movements of stock price, we can help many investors to analyze the market and make investment decisions.

This study uses the neural network algorithm to carry out contrast experiments to predict the stock price and ultimately determines a more optimal parameter setting solution which can improve the accuracy of prediction efficiently.

NEURAL NETWORK

Neural network originated in the biological neural network (Han and Kamber, 2007). It is a simple system that it is the simulation of biological nervous system structure, process and system functions. It is a branch of artificial intelligence and it began in the 1940s and included mathematics, electronics and control, computer science, neurophysiology, cognitive science, nonlinear dynamics and many other disciplines. As we all know, the human brain is the material basis of mental activity and thinking is the embodiment of human intelligence. It was a long term that people tried to understand the working mechanism of the human brain and tried to imitate the human brain functions. Neural network is such a class of network that consists of a large number of processing elements (neurons) (Zhang *et al*., 2010). It is similar to brain synapses and can be a mathematical model of information processing. It is the abstraction, simplification and simulation of the human brain and reflecting the basic characteristics of the human brain.

Similar to humans, neural network has the following features (Xie, 2008):

- The characteristics of parallel processing
- The characteristics of fault tolerance
- The characteristics of associative memory
- The characteristics of optimization
- The characteristics of VLSI Implementation
- The characteristics of handling difficult issues

Corresponding Author: Nan Ma, College of Information Technology, Beijing Union University, Beijing, 100101, China

Because neural network has memory and learning functions, it can be used in various fields, such as product quality analysis, experimental data modeling, analysis and design of engineering, commercial credit assessment, signal classification and so on.

Neural network is a computational model consists of a large number of nodes (also called neurons) and mutual weighted connections. Each node represents a specified output function, called activation function. Every connection between two nodes represents a weight of connection signal. It is equivalent to the memory of neural network. Outputs of the network are according to the connections, weights and activation functions. The network is usually approximation of some algorithms and functions or may be an expression of logical strategy (Zhu and Wang, 2010).

In a Multilayer Perceptron neural network, each neuron receives one or more inputs and produces one or more identical outputs (Li *et al*., 1999). Each output is a simple non-linear function of the sum of the inputs to the neuron. Inputs pass forward from nodes in the input layer to nodes in the hidden layer and then pass from the hidden layer to the output layer; there are no connections between neurons within a layer. If no hidden layer is included, as in a logistic regression model, inputs pass forward directly from nodes in the input layer to nodes in the output layer.

Neural network is mainly to solve the classification and regression tasks, it can identify the smooth, continuous and nonlinear relationship between the input attributes and predictable attributes.

As is known that the stock market is a complex, uncertain and nonlinear system. Ups and downs of the stock affected by many factors, like economy, policy and investors. Therefore there are some very complex and hidden relationships among these factors. And most important point is that all these changes and effects are concentrated reflected in the stock price, like opening price and closing price. Unfortunately, even if these changes and effects are concentrated to the stock's price, one cannot obtain the useful relationships and the regulations of the prices by simple observation and analysis of these massive stock data (Wu *et al*., 2001). According to the feature of neural network algorithm, it can effectively handle these complex relationships and excavate the rules including the price variation and the relationship between opening price and closing prices and so on. Finally, it can provide more scientific and efficient references for the investment decisions (Zeng, 2009).

NEURAL NETWORK IN STOCK PRICE FORECASTING PROCESS

Neural network algorithm: Neural network algorithm uses a Multilayer Perceptron network and it composed of up to three layers of neurons. These layers are input layer, hidden layer and output layer. Input layer formed by the input neurons, hidden layer formed by the hidden neurons, output layer formed by the output neurons and neurons connected by the edge with weight (Isaksson *et al*., 2005a).

Each input neuron is mapped to an input attribute. The value of the property was converted to a floating point number between-1 to 1 before processing. They provide input attribute values for the data mining model. For discrete input attributes, an input neuron typically represents a single state from the input attribute. This includes missing values, if the training data contains nulls for that attribute. A discrete input attribute that has more than two states generates one input neuron for each state and one input neuron for a missing state, if there are any nulls in the training data. A continuous input attribute generates two input neurons: one neuron for a missing state and one neuron for the value of the continuous attribute itself. Input neurons provide inputs to one or more hidden neurons.

Hidden neuron receives the outputs from the input neurons or hidden neurons in front of it and after some calculation, the result will be passed to the next level. It allows the network to learn nonlinear relationships (Isaksson *et al*., 2005b).

Output neuron is usually on behalf of predictable properties, the value is usually a floating point number between 0 and 1. For discrete input attributes, an output neuron typically represents a single predicted state for a predictable attribute, including missing values. A neuron receives input from other neurons, or from other data, depending on which layer of the network it is in. An input neuron receives inputs from the original data. Hidden neurons and output neurons receive inputs from the output of other neurons in the neural network. Inputs establish relationships between neurons and the relationships serve as a path of analysis for a specific set of cases. Each input has a value assigned to it, called the weight, which describes the relevance or importance of that particular input to the hidden neuron or the output neuron. The greater the weight that is assigned to an input, the more relevant or important the value of that input. Weights can be negative, which implies that the

input can inhibit, rather than activate, a specific neuron. The value of each input is multiplied by the weight to emphasize the importance of an input for a specific neuron. For negative weights, the effect of multiplying the value by the weight is to deemphasize the importance.

Neural Network algorithm uses feedforward networks, there does not exist directed ring in the network.

Figure 1 shows a feedforward network of three-tier structure, where node 1, 2, 3 are the input neurons, node 4, 5 are hidden neurons, node 6, 7 are the output neurons.

A neuron has one or more inputs but only one output, neural network algorithm uses the method of weighted sum (each input value is multiplied by the weight associated with it and then summing the product) combines multiple input values, then compute output values (activate) according to the different types of neuron:

For hidden neurons, using the function named tanh:

$$O = (e^{a}-e^{-a})/(e^{a}+e^{-a}) \quad (1)$$

For output neurons, using the function named sigmoid:

$$O = 1/(1+e^{-a}) \quad (2)$$

where, a is the input value, O is the output value.

Figure 2 shows the calculation of combination and output of internal neurons. First, the input value 1, 2 and 3 will be combined by the method of weighted sum. Then it will choose tanh or sigmoid according to the type of neuron to get the output.

The process of neural network algorithm is described as follows:

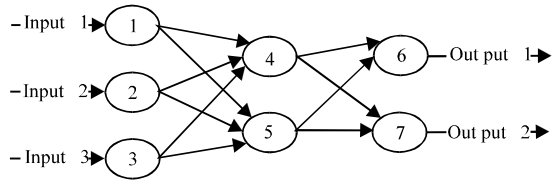

Fig. 1: Feedforward Network

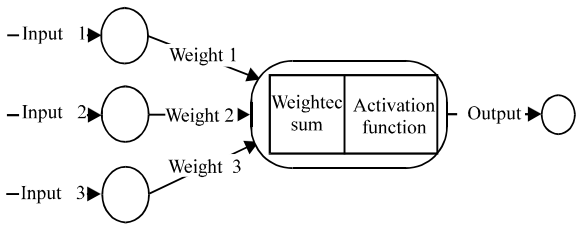

Fig. 2: Calculation In Neuron

Step 1: Assigned weights randomly and the range is -0.0 to 1.0

Step 2: Based on the current weight to calculated the output (use the weighted sum and sigmoid or tanh)

Step 3: Calculated output error for each output and hidden neurons (compared with the actual data) and update the weights and back-propagation

Step 4: Repeat from step 2 until satisfying so far

Here are a few possible termination conditions:

- Sufficient accuracy to meet the test set
- Maximum number of iterations
- Weight convergence
- Timeout

Error calculated function is: For output neurons:

$$Err_i = O_i (1-O_i) (T_i-O_i) \quad (3)$$

where, O_i is the output of output neuron i, T_i is the actual value of the output neuron i.

For hidden neurons:

$$Err_i = O_i (1-O_i) \Sigma_j Err_j w_{ij} \quad (4)$$

where, (O_i is the output of hidden neuron i, it has j outputs passed to the lower layer. Err_j is the error of neuron j and w_{ij} is the weight between the two neurons.)

Adjust the weight:

$$w_{ij} = w_{ij}+l*Err_j*O_i \quad (5)$$

where, l for the learning rate is in the range 0 to 1.

Through the process of neural network algorithm, we can know clearly that neural network must first go through a learning process, namely training and then it can work, which means prediction.

To further elaborate the workflow of neural network algorithm, a practical example will be illustrated:

Suppose the algorithm will identify the type "A" and "B" and provide that if "A" is input, the output should be 1 and if the input is "B", the output should be 0.

First, the connection weights of the network are assigned randomly and the range of value is between 0 and 1.

Second, suppose the data regarding type "A" is sent to the network. The network uses the current weights and function like sigmoid or tanh to calculate the output.

Third, it will calculate the output error for each output and hidden neurons and compare them with the actual data (to check if the output is close to 1 for type "A"). If the error is higher than the expected minimum error and the maximum count of circulation is not reached, the network will adjust the weights of the whole network until the terminal condition is touched.

The process of learning type "B" is similar to the above.

After the process of training, this network can obtain the characteristics of both type "A" and "B". By this time, if the data about type "B" is sent to the network, the network has great probability to return 0.

Data preprocessing: The first step of data mining and knowledge discovery process is data preprocessing. Statistics found that in data mining and knowledge discovery process, the data pre-processing accounted for 60% of the entire workload. This is due to that data from real world are largely incomplete and inconsistent which cannot carry out data mining directly or mining result is unsatisfactory. In order to improve the quality of data mining, data pre-processing techniques comes out. There are several methods of data preprocessing: data cleaning, data integration, data transformation, data reduction and so on. These data processing techniques used before the data mining can greatly improve the quality of data mining models and reducing the time required for the actual mining.

In this study, test data is from stock transaction data named "Gehuayouxian" since February 8, 2001 to February 11, 2011. Data schema is GHYX (date, opening price, closing price, maximum, minimum, volume, turnover).

The meaning of each attribute as shown in Table 1.

In fact, there are many properties associated with the stock price but we select only those attributes for data mining. Because these attributes do not like financial data, they change frequently and are the most basic, essential properties of the stock. Since the source data we collected using the format of "YYYY-MM-DD" as date which can lead the failure of model training. We need preprocess data which means transform date to string. However, how to reprocess these data depends on the source data, data mining algorithm and the experiment's environment. The source data used in this study summarized in Appendix A.

Table 1: Attribute meaning

Opening price	First deal price on each business day
Closing price	the last deal price on each business day
Maximum	Maximum deal price on each business day
Minimum	Minimum deal price on each business day
Volume	No. of deal
Turnover	Total amount of deal

Create and train model: Standardization of data mining language is a key to develop a new generation of data mining system. DMX (Data Mining extension) is proposed byMicrosoft in SSAS (SQL Server Analysis Services) and it is a language that can achieve compliance with OLE DB for Data Mining. DMX supports data mining system processes relational databases directly and it provides the methods of creating, accessing andmanagingdataminingalgorithms and the system of open source. This is a breakthrough of the development of standardization in data mining (Li and Sun, 2011).

In this study, we use DMX to train the network and predict the price of stock. The specific steps as follows:

First, using DMX language to create the model which contains the date, opening price, maximum, minimum, closing price, volume, turnover and date as the key, the closing price as the predictable attribute:

```
CREATE Mining Model Stock_Model
(DealDate text key, --date as key
    OpenPrice double continuous, --opening price, continuous
    Maximum double continuous, --maximum, continuous
    Minimum double continuous, --minimum, continuous
    ClosePrice double continuous predict, --closing price, continuous, as predictable attribute
    Volume double continuous, --volume, continuous
    Turnover double continuous--turnover, continuous
)
```

Using Microsoft_Neural_Network--neural network algorithm.

This piece of DMX code is very easy to understand. Its function is to create a mining model which contains the necessary fields with correct data type.

We use neural network algorithm which provided by Microsoft to create the model, the algorithm parameters are set to default.

After the model is set up we should train the model. The purpose of training is to make the model adapt to the source data and improve itself. In theory, more source data can strengthen the model but the training process lasts longer.

Before we train the model, we should ensure that the source data we collected is stored in the database (in this sample, we use database named Stock to store the source data).

Then we use following DMX code to train the model:

- INSERT INTO Stock_Model (DealDate, OpenPrice, Maximum, Minimum, ClosePrice, Volume, Turnover)
- OPENROWSET ('SQLNCLI', 'Integrated Security = SSPI; Data Source = localhost; Initial Catalog = Stock', 'select date, opening price, maximum, minimum, closing price, volume, turnover from gehuayouxian')

The above code means that we train the model by the source data we collected. Although it looks like a simple SQL insert statements, it really did a lot of things behind. To begin with, it connected to the SQL Server and located the specific database that the statement assigned. Moreover, it searched the dataset from the table named gehuayouxian and the dataset included all the fields, such as date, volume, turnover and so on. Furthermore, the dataset was filled into the data mining model, namely Stock_Model and trained the model with the algorithm named Microsoft Neural Network. This process will continue for unpredictable time, because the span of training process based on the number of source data. The greater the amount of training data, the longer the training time will be.

Interpretation of the model: After execute the above code, the model is created and trained. Then we will interpret the model. The rules obtained by training the model can be explained by mining model viewer as shown in Fig. 3 and Table 2-3.

Table 2 illustrates the value of input property which has decided when explain the model. We can select input properties on describing the model by ourselves. Figure 3 shows the two states of predictable property, this range can be selected too. This selection will impact on the display of the rules. Table 2 shows the rules with above premise.

For example, as shown in Fig. 3 and Table 2 and 3, if a sample's opening price is between 7.65 and 14.986 while its volume is between 379 and 7048.116 and its maximum between 25.840 and 43.998, then this sample's closing price is tend to 25.340-43.143.

If we change the range of input properties or output properties, the rules displayed in Table 3 will be changed too. But all the changes depend on the result of the model training.

Prediction: Now, we have created and trained the model. Training process aims to process data and this process enable the model to gain some valuable experience and rule. But all the works are the preparation for the next step, which is prediction. So our next step is using the historical data to carry out prediction what can help us to analyze the accuracy of the model.

Figure 4 shows the accuracy of prediction this time.

Abscissa represents the actual value of closing price of the stock, the ordinate represents the predicted value of closing price, the 45-degree line represents the precise prediction and each point represents a sample. Therefore, we can know the actual value and predicted value from the position of the point. The fitting of points and line reflects the accuracy of prediction.

It can be seen from the figure that the accuracy of this model's prediction is not high. Especially there is a great error on the range of 15 yuan and 35 yuan.

Table 4 shows the prediction directly. In Table 4, the actual closing price, predicted closing price and their difference from 2001-2-8 to 2001-3-13 are listed. These values have been rounded for easier comparison.

As can be seen from Table IV, the average error of prediction is more than 0.7 yuan. In stock trading system, people restrain the number of stock transaction by multiples of 100, so the error cannot be ignored.

Improve the model: The algorithm uses default parameter setting solution which leads the great prediction error and it cannot make contribution to investment decision. So we should improve the model in order to improve the accuracy of prediction by means of changing parameters' value of the algorithm..

Among the parameters provided by the algorithm, Hidden_Node_Ratio and Holdout_Percentage

Table 2: Set input property

Attribute	Value
Opening price	7.650-14.98600
Volume	379.000-7048.116

Table 3: Show the relues

Attribute	Value	Tend 15.013-20.176	Tend 25.340-43.143
Volume	17613604.222-129062333.098		
Volume	240511061.973-624764731.511		
Volume	129062333.098-240511061.973		
Maximum	25.840-43.998		
Maximum	20.574-25.840		
Maximum	19.785-24.865		

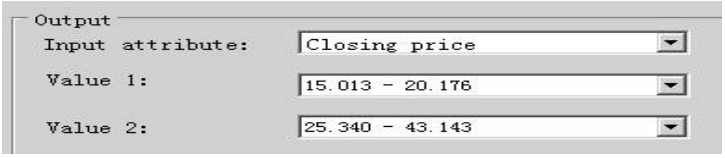

Fig. 3: Set output property

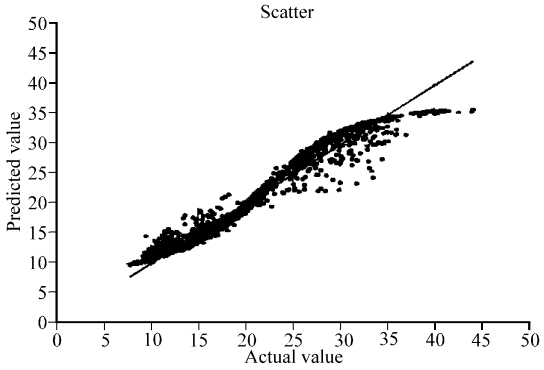

Fig. 4: Accuracy chart

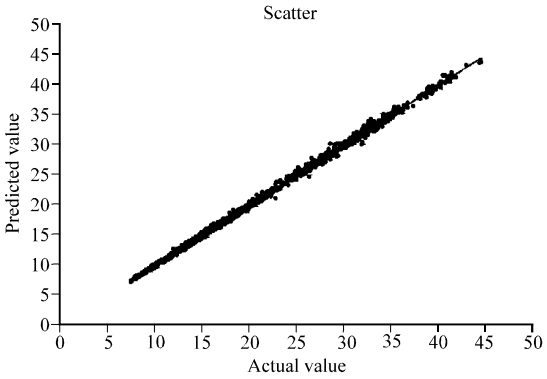

Fig. 5: Accuracy chart

Table 4:Prediction results

Date	Actual closing price (yuan)	Predicted closing price (yuan)	Difference (yuan)
2001-2-8	28.45	25.86	2.59
2001-2-9	30.09	29.09	1.00
2001-2-12	31.12	31.27	0.15
2001-2-13	30.99	32.58	1.59
2001-2-14	31.99	32.72	0.73
2001-2-15	31.25	32.78	1.53
2001-2-16	33.25	32.12	1.13
2001-2-19	32.90	33.28	0.38
2001-2-20	32.86	33.03	0.17
2001-2-21	32.79	33.54	0.75
2001-2-22	32.96	33.57	0.61
2001-2-23	33.60	33.73	0.13
2001-2-26	33.65	33.94	0.29
2001-2-27	33.00	33.78	0.78
2001-2-28	32.70	33.60	0.90
2001-3-1	32.70	33.60	0.90
2001-3-2	33.30	33.69	0.39
2001-3-5	32.78	33.67	0.89
2001-3-6	33.55	33.62	0.07
2001-3-7	33.50	33.94	0.44
2001-3-8	34.99	33.71	1.28
2001-3-9	34.82	34.40	0.42
2001-3-12	35.10	34.40	0.70
2001-3-13	34.59	34.40	0.19

have the greatest impacts on the accuracy of prediction. Hidden_Node_Ratio decides the number of hidden nodes in this model. Enhance the hidden nodes can improve the accuracy of prediction but it leads a longer training time. Holdout_Percetage specifies the percentage of training data and it would be used as a part of terminal condition.

In this study, we try to modify the two parameters in order to find the relationship between the accuracy of prediction and the values of the two parameters.

During the experiment, we recorded the values of Hidden_Node_Ratio and Holdout_Percentage, training time and we calculated the average prediction error after every experiment.

In Table 5, line 2 shows the default parameter value and the result. We can see line 1 to line 5 from this table that the average prediction error is not increasing or decreasing with the parameter named Hidden_Node_Ratio. And we can see line 6 to line 10 from this table that the average prediction error is not increasing or decreasing with the parameter named Holdout_Percentage too. Line 11 is the optimum value of the integrated two experimental parameters but the result is still not satisfactory. Last 2 lines take a more extreme value but the prediction error is small enough as a reference for investment decisions.

Table 5: Paramemers impact on training time and error

Ine No.	Hidden_Node _Ratio	Holdout_ Percentage	TrainingTime (sec)	Average prediction error
1	2	30	18	0.96
2	4	30	17	1.21
3	8	30	17	0.83
4	12	30	18	0.80
5	30	30	18	1.01
6	4	5	17	1.13
7	4	15	17	1.22
8	4	10	17	1.13
9	4	80	17	1.08
10	4	60	17	1.11
11	12	80	17	0.89
12	0	99	17	0.24
13	0	1	17	0.19

Figure 5 is the chart of predictive accuracy which used Hidden_Node_Ratio = 0 and Holdout_Percentage = 1.

It can be seen from the figure that points and 45 degree line fitting in a good condition. Only a few points far from the 45-degree line, most of the points are in a good fit with this line. So this parameter setting solution is suitable for stock prediction.

CONCLUSION

This study compared the impact of the accuracy of prediction and training time with parameter Hidden_Node_Ratio and Holdout_Percentage. There

were a lot of experiments are done to analyze the relationship between prediction accuracy and parameters' values. The result shows that it is effective on improving accuracy of prediction to use less Hidden_Node_Ratio and Holdout_Percentage which has little influence on training time. This parameter setting solution can reduce 60% average prediction error compared to the default solution. So, it can be a preferred option on prediction of stock when using neural network algorithm.

ACKNOWLEDGMENTS

This research is partially supported by the National Natural Science Foundation of China under grants 61300078 and 61175048 and the project of new starting point of Beijing Union University under grants ZK10201312 (Research on fuzzy cognitive map ensemble classifiers and its application).

REFERENCE

Han, J.W. and M. Kamber, 2007. Concept and Technology of Data Mining. China Machine Press, Beijing.

Isaksson, M., D. Wisell and D. Ronnow, 2005. Nonlinear behavioral modeling of power amplifiers using radial-basis function neural networks. Proceedings of the International Microwave Symposium Digest, June 12-17, 2005, Long Beach, CA.

Isaksson, M., D. Wisell and D. Ronnow, 2005. Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks. IEEE Trans. Microwave Theo. Tech., 53: 3422-3428.

Li, M.Q., B.Y. Xu and J.S. Kou, 1999. On the combination of genetic algorithms and neural networks. Syst. Eng. Theo. Pract., 19: 20-24.

Li, Y. and L. Sun, 2011. Design and implementation of data mining algorithms package based on DMX. Comput. Technol. Dev., Vol. 21.

Naskas, N. and Y. Papananos, 2004. Neural-network-based adaptive baseband predistortion method for RF power amplifiers. Trans. Circuit. Syst. Ii-Exp. Briefs, 51: 619-623.

Tang, Z. and M. Jamie, 2007. Data Mining with SQL Server 2005. Tsinghua University Press, Beijing.

Wu, W., W.Q. Chen and B. Liu, 2001. Prediction of ups and downs of stock market by BP neural networks. J. Dalian Univ. Technol., 41: 9-15.

Xie, B.C., 2008. Bussiness and Data Mining in Application of Microsoft SQL Server. China Machine Press, Beijing.

Yi, M. and W. Lv, 2009. Application of data mining technology based on BP algorithm for forecasting stock price. Mod. Comput., 2: 106-110.

Zeng, D.H., 2009. The application of BP neural network in predicting the stock price. China Sci. Technol. Inform., Vol. 21

Zhang, Y., Y. Yin and W. Li, 2010. Directly Determine Method of Neural Network Weights. Zhongshan University Press, Guangzhou..

Zhu, K. and Z.L. Wang, 2010. Proficient in Matlab Neural Network. Electronic Industry Press, Beijing.