Applied Mathematics and Nonlinear Sciences

<https://www.sciendo.com>

Mathematical model of back propagation for stock price forecasting

Feng Li^{1*}, Mohammed Yousuf Abo Keir²

¹WeiNan Normal University, Shaanxi, 714000, China

²Applied Science University, Al Eker, Kingdom of Bahrain

Submission Info

Communicated by Juan Luis García Guirao

Received June 16th 2021

Accepted September 24th 2021

Available online December 27th 2021

Abstract

In order to establish a more accurate Stock Price Prediction Model, the Stock Price Prediction mathematical Model SPPM (Stock Price Prediction Model) based on BP neural network with high frequency data is proposed in this paper. The SPPM integrates several neural networks to predict the movement of stock prices over the next few days. The key problems in SPPM—such as data preprocessing, output fusion and the selection of nodes in the hidden layer of neural network—are discussed in detail. The experimental results show that the SPPM predicted the closing price of 2019-03-19 and 2019-03-20 as 207.16 and 207.22, respectively, which is very close to the actual observed value, and the back propagation mathematical model SPPM has a certain practical value. Our conclusion is that the back propagation model can predict the stock price with high accuracy.

Keywords: Neural network; Mathematical model of back propagation; Stock price forecast.

AMS 2010 codes: 91-08

1 Introduction

The stock market is risky; so, it is necessary to be cautious when transacting in it. Stock market investment requires investors to make prudent decisions. For stock trading, a more rigorous mathematical model can be used to make decisions, which may reduce investment risks and maximise investment returns [1]. The income that stock invests decides at the price of buying and selling, when buy sell, need investor undertakes analysis. Generally speaking, buy to sell a stock to the fundamentals of the stock, policy, trading volume, running trend, the market index to consider, thereby choosing a limited stock to buy. So how to choose stocks and when to buy and sell? According to the relevant data and prediction, according to the corresponding mathematical model to analyse. Here, according to the profit per share, buying volume and selling volume, price-earnings ratio and other analysis, which are built into the mathematical model, the decision is made concerning the

*Corresponding author.

Email address: li_feng012@163.com

timing of buying and selling. In China, people generally accept and widely use statistical analysis methods such as stock price graph analysis method and index analysis method to predict the trend of the stock market. The direction of the traditional analysis method of stock price fluctuations has a certain degree of quantitative or qualitative description. However, these methods are only possible projections for the volatility of the share price, is not clear, it can reach the level of the other in the process of using these methods is highly affected by the subjective factors, so the result explanation often vary from person to person. Reliable quantitative description of stock price fluctuation is a difficult problem in the field of stock forecasting [2, 3]. Since the emergence of the stock market, many scholars and investors have been committed to the prediction of the trend of the stock market, and many forecasting analysis methods have emerged, including the basic analysis method and technical analysis method widely used by investors. With the development of computer technology and artificial intelligence technology, a collection of methods of stock prediction has been introduced, among which the application of neural network to the stock market prediction has been widely studied by the academic circle, and has become a hot topic of academic research. Through the research and demonstration of domestic and foreign scholars, it has been tentatively ascertained that the time series prediction method based on feedforward neural network is the best one at present [4]. Neural network because of its own characteristics in conformity with the characteristics of the stock market, and on the establishment of the model to get rid of the traditional analysis model of dependence on long-term, large sample of data statistics, and to consider only the last a period of history data and the nonlinear relationship between forecast target, make it stand out in the numerous prediction method.

2 Research Methods

2.1 BP neural network

(1) BP neural network model

BP algorithm is used to train multi-layer feedforward neural network (the feedforward network using BP learning algorithm is called BP neural network), which belongs to the learning algorithm with tutors. BP network has the characteristics of clear structure, easy implementation, powerful computing function and superior performance, so it is widely used in many fields such as pattern recognition and text classification. BP neural network adopts a parallel network structure, including input layer, hidden layer and output layer. BP network has been proved to have strong nonlinear mapping ability and generalisation function, and multi-layer network can approximate arbitrary nonlinear function. Before the training of BP neural network, the parameters of the network should be determined and initialised first, and the training of the network can only be started when everything is ready. The input signal enters the network from the input layer, and is output through the output layer after the weighted sum of each layer and the transformation of activation function [5]. This process is the forward propagation of the input signal. In this process, the input of each layer of neurons is only affected by the output of the previous layer of neurons and the weight and Min value of the network remain unchanged. If the error between the actual output and the expected output of the network is large, the error signal is transferred via back propagation to reduce the error and make the actual output approach the expected output gradually.

Figure 1 shows a three-layer feedforward neural network, in which the input layer and the output layer each have two nodes. The hidden layer has three nodes. Each node in the hidden layer and the output layer is a sigmoid cell, which is based on a smooth differentiable threshold function. For each Sigmoid cell, its output is calculated as follows: $o = \sigma(\vec{w} \cdot \vec{x})$, among them $\sigma(y) = \frac{1}{1+e^{-y}}$, $\vec{x}$ is the input vector of this node, and $\vec{w}$ is its weight vector. σ is often called the sigmoid function or we can also call it the logistic function.

The input of the hidden layer node comes from the input layer. When the weight of each input layer node to each hidden layer node is determined, the output of the hidden layer node is determined. The output of the hidden layer is used as the input of the output layer. Similarly, when the weight of each hidden layer node to each output layer node is determined, the output value of the output layer node is also determined. Therefore,

Fig. 1 Feedforward neural network with three-layer structure

the weight vector learning is the key. In essence, the weight learning problem is a search problem, i.e. it is necessary to find a reasonable $\vec{w}$ in the $R^{|\vec{w}|}$ space to minimise the error of the corresponding network on the training samples. The formal expression is, find $\vec{w}$, i.e. minimise this expression.

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in b} \sum_{k \in \text{outputs}} (t_{kd} - o_{kd})^2 \quad (1)$$

where D is the training sample set; D is a training sample; outputs are collections of network output units; t_{kd} is the KTH dimension of the expected output vector of d ; and o_{kd} is the value of the KTH dimension in the output vector of the neural network to D . It is noteworthy to observe that genetic algorithm, particle swarm optimisation algorithm and so on can be used to find the approximate optimal solution of $\vec{w}$. In this paper, BP algorithm based on stochastic gradient descent is used to search $\vec{w}$.

(2) BP algorithm

BP algorithm is a learning algorithm with a tutor. Taking three-layer BP neural network as an example, the derivation of BP algorithm is made. We assume that the input learning samples are P , $x_1, x_2, \dots, x_p$, the corresponding expected output is $t_1, t_2, \dots, t_p$, the actual output is $y_1, y_2, \dots, y_p$ and the number of neurons in the hidden layer is s . The idea of BP algorithm is to correct the connection weight and Min value by calculating the mean square error between the actual output and the expected output, so that the actual output and the expected output are as close as possible.

(1) Forward propagation of input signals

Output of the i th neuron in the hidden layer is as follows:

$$a_i = f \left(\sum_{j=1}^p x_j w_{ij} - \theta_j \right) \quad (2)$$

where w_{ij} is the connection weight between the input layer and the hidden layer, and θ_j is the Min value of neurons in the input layer.

The output of the KTH neuron in the output layer is:

$$y_k = f \left(\sum_{r=1}^p a_r w'_{kr} - \theta'_r \right) \quad (3)$$

If $net'_k = \sum_{r=1}^p a_r w'_{kr} - \theta'_r$ is taken, Eq. (3) is converted into Eq. (4), as follows:

$$y_k = f(net'_k) \quad (4)$$

where w'_{ir} is the connection weight between the hidden layer and the output layer, and θ'_r is the Min value of the neurons in the hidden layer. The error function is:

$$E(w, \theta) = \frac{1}{2} \sum_{k=1}^p (t_k - y_k)^2 \quad (5)$$

(2) Back propagation of error signals

When the actual network output is not consistent with the expected output, the gradient descent method is used to correct the network connection weight. The connection weight adjustment formula between the hidden layer and the output layer is:

$$\Delta w'_{kr} = -\eta \frac{\partial E}{\partial w'_{kr}} \quad (6)$$

The weight adjustment formula from the input layer to the hidden layer is:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \quad (7)$$

According to the properties of partial derivatives, Eq. (6) can be written as:

$$\Delta w'_{kr} = -\eta \frac{\partial E}{\partial w'_{kr}} = -\eta \frac{\partial E}{\partial net'_k} \cdot \frac{\partial net'_k}{\partial w'_{kr}} \quad (8)$$

According to Eqs (3) and (5), the required formula can be obtained as the following in Eq. (9):

$$\begin{aligned} E &= \frac{1}{2} \sum_{k=1}^p (t_k - y_k)^2 \\ &= \frac{1}{2} \sum_{k=1}^p \left(t_k - f \left(\sum_{r=1}^p a_r w'_{kr} - \theta'_r \right) \right)^2 \end{aligned} \quad (9)$$

For the output layer, there is the formula:

$$\frac{\partial E}{\partial net'_k} = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial net'_k} = \frac{\partial E}{\partial y_k} \cdot f'(net'_k) \quad (10)$$

And because

$$\begin{aligned} \frac{\partial E}{\partial y_k} &= -(t_k - y_k) \\ \frac{\partial net'_k}{\partial w'_{kr}} &= a_r \end{aligned} \quad (11)$$

the formula of weight adjustment between hidden layer and output layer is:

$$\Delta w'_{kr} = \eta (t_k - y_k) \cdot f'(net'_k) \cdot a_r \quad (12)$$

2.2 The theoretical basis of BP neural network for predicting stock price

The so-called prediction means to estimate the value of unknown data in the future through some known historical data, and we set the time series $\{x_i\}$, in which the historical data $x_n, x_{n+1}, \dots, x_{n+m}$. The neural network uses data $x_n, x_{n+1}, \dots, x_{n+m}$ to fit the function and predicts the value at the moment of $n+m+k$ ($k > 0$) in the future, i.e. it predicts some nonlinear function relation of $x_{n+m+k}: x_{n+m+k} = f(x_n, x_{n+1}, \dots, x_{n+m})$. The neural network is used to fit the function relation and deduce the future value. This is the basic idea of time series prediction by artificial neural network. The neural network structure for time series prediction can be divided into single step prediction and multi-step prediction. The number of network output of single-step prediction is 1, which only predicts the data of one day in the future. The number of network output of multi-step prediction is multiple, which can predict the data of many days in the future. Application of neural network is the basic principle of stock price prediction; application of neural network is a strong nonlinear approximation ability, which will determine the stock price factors as the input matrix, and the stock price as the output of target matrix, with historical data as the training data of network training; the training is actually the result of the fitting of the nonlinear mapping between the output of an input [6, 7]. Then, using this input-output function, the new input is given, and the output is the predicted result.

The traditional linear prediction method takes the weighted sum of several past observations as the prediction result, while the artificial neural network is a highly parallel nonlinear system, which is composed of a large number of simple processing elements connected with each other, and has the characteristics of large-scale parallel processing. Although the functions of each processing unit are very simple, the parallel activities of a large number of simple processing units endow the network with rich functions and high speed. The extensive interconnection and parallel work of neurons inevitably make the whole network highly nonlinear. The self-learning of neural network means that when the external environment changes, after a period of training or perception, the neural network can automatically adjust the structural parameters to produce the desired output for the given input. Training through self-learning is the natural way for neural networks to learn; and so, the two words 'learning' and 'training' are often used interchangeably. By adjusting the nonlinear action of neurons, neural network approximates the nonlinear mapping within the system more accurately, which makes the prediction accuracy of chaotic time series several orders of magnitude higher than that of traditional methods.

2.3 Establishment of mathematical model of back propagation for stock price prediction

(1) Prediction model

The stock price prediction model (SPPM) designed in this paper is shown in Figure 2. The model is divided into two stages: training stage and application stage. The training stage refers to the process of stock price prediction based on the neural network learned in the training stage [8,9]. In order to improve the generalisation ability of SP-PM and enhance the effectiveness of the model applied to stock price prediction, the neural network integration technology is adopted. Neural network integration has proved to be a very effective approach to improve the processing power of the learning system even if it is only a simple voting or averaging of a set of networks. In this paper, the integration of neural network is embodied in two aspects: individual generation and result fusion.

Fig. 2 Stock price forecasting model

(2) Data preprocessing

We assume that $t_1, t_2, \dots, t_n (n \geq 2)$ is a continuous time series. At the moment t_i , any attribute of a stock (such as opening price, highest price, lowest price, closing price, trading volume, transaction amount etc.) can be obtained. For the data sequence obtained on the time series $t_1, t_2, \dots, t_n$ by an attribute (for instance, the closing price indicated below), we can tentatively ascertain the data sequence to be: $x_1, x_2, \dots, x_n$.

First, we normalise each dimension of the data sequence:

$$x_i = \frac{x_i - \min}{\max - \min} \quad (13)$$

where $1 \leq i \leq n$, max and min respectively represent $x_1, x_2, \dots$, the maximum and minimum values of x_n . Second, $n-N-M+1$ samples are constructed according to Eq. (13), where n and M represent the future closing prices of $M(M \geq 1)$ days predicted by the closing prices of the previous $N(N \geq 1)$ days; $n-N-M+1 \geq 2$.

$$\begin{cases} x_1 \\ x_2 \\ \dots \\ x_3 \end{cases} \Rightarrow \begin{cases} X_1 = \langle x_1, x_2, \dots, x_N \rangle, Y_1 = \langle x_2, x_3, \dots, x_{M+1} \rangle \\ X_2 = \langle x_2, x_3, \dots, x_{N+1} \rangle, Y_2 = \langle x_3, x_4, \dots, x_{M+2} \rangle \\ \dots \\ X_{n-N-M+1} = \langle x_{n-N-M+1}, x_{n-N-M+2}, x_{n-N-M+3}, \dots, x_{n-M} \rangle \\ Y_{n-N-M+1} = \langle x_{n-M+1}, x_{n-M+2}, \dots, x_n \rangle \end{cases} \quad (14)$$

The sample set we get is denoted by D , and obviously $|D|$ is $n-N-M+1$. The i ($1 \leq i \leq n-n-m+1$) sample in D is expressed as $\langle X_i, Y_i \rangle$, where $|X_i|$ is N and $|Y_i|$ is M . For a time series, as shown in Figure 3, when the window size is adjusted, different sample sets can be obtained. Different neural network individuals can be trained by using different sample sets [10]. The following, $D_1, D_2, \dots$ and D_k represent k sample sets, respectively, and k represents the number of Windows or the total number of neural networks.

Fig. 3 Time series based on windowing technique

(3) Establishment of network structure

If the input layer and output layer adopt linear transformation function and the hidden layer adopts Sigmoid transformation function, then the multi-layer neural network with a hidden layer can approximate any rational function with arbitrary accuracy. Therefore, the feedforward network used in this paper has three layers of structure: input layer, hidden layer and output layer. The number of nodes in the input layer of the network is determined by the dimension of the input vector. Since the dimension of the input vector is N , the number of nodes in the input layer is determined to be N . The number of nodes in the output layer is larger by the dimension of the output vector, which is the direct cause of ‘overfitting’ in training. However, it is a pity that there is no scientific and universal method to determine it in theory at present. In order to avoid the phenomenon of ‘overfitting’ during training as much as possible and ensure high enough network performance and generalisation ability, the most basic principle to determine the number of nodes in the hidden layer is rendered as compact as possible under the premise of meeting the accuracy requirements, i.e. the nodes in the hidden layer are kept to as few a number as possible. The following conditions must be satisfied when determining the number of hidden layer nodes: (1) The number of hidden layer nodes must be less than $(|D| - 12)$; The number of training samples must be more than the connection weight of the network model, which is generally around 2–10 times.

Let the number of nodes in the hidden layer be h , and the connection weight be: $N \times h + M \times h$. The second condition can be expressed as: $2 \leq \frac{|D|}{N \times h + M \times h} \leq 10$, $\frac{|D|}{10 \times (N+M)} \leq h \leq \frac{|D|}{2 \times (N+M)}$.

Thus, the above two conditions can be expressed as:

$$\frac{|D|}{10 \times (N+M)} \leq h \leq \frac{|D|}{2 \times (N+M)} \quad (15)$$

To sum up, this paper adopts N-H-M neural network, with hidden layer node as sigmoid element and output layer node as linear element.

(5) Prediction of fusion

K data sets can be trained to obtain K neural networks. We assume that for an unknown sample $\langle X, ? \rangle$, let SPPM output be $rY_1, Y_2, \dots, Y_k$, respectively. Next, we will discuss how to ‘merge’ these outputs and finally give the specific method of the stock price trend in the future M days. One of the easiest ways to do this is to take the average value of each output, as follows:

$$Y = \frac{\sum_{i=1}^k Y_i}{k} \quad (16)$$

where the magnitude of Y is M, and each dimension represents the predicted value for a certain day in the future.

In many cases, we don’t need to know the exact value of the future – only whether it is going up or down. In this case, we can just look at the value of the current day in Y compared to the previous day.

3 Result Analysis

In this paper, the experimental hardware environment used is the following— CPU: Intel-Core2DuoProcessorT5500; Memory: 1GB; Programming environment: Matlab7.1. The experimental data are taken from the ‘flush’. In this paper, we discuss the case of N=10 and M=2, i.e. the trend of the next 2 days is predicted based on the data of the first 10 days. In Table 1, SPPM predicted that the Shanghai Composite Index would close at 2374.64 on March 19, compared to the actual closing price of the previous trading day (2404.74), when the forecast trend was down. This is a departure from the actual running results of the day. In Table 1, SPPM predicted that the Shanghai Composite Index would close at 2373.84 on March 20, compared with the actual closing price of the previous trading day (2374.64), when SPPM predicted a downward trend. This is exactly in line with the actual operating results of the day, and the predicted value is very close to the actual closing price. Overall, SPPM expects the next two days to be on the downside relative to the current date (03-16), which is in line with the actual operation. In order to further verify the performance of SPPM, we also forecast the closing price of Kweichow Moutai 2019-03-19 and 2019-03-20. Table 2 shows the prediction results of SP-PM for Kweichow Moutai. On March 16, the actual closing price of Kweichow Moutai was 207.59, while on March 19 and 20, Kweichow Moutai rose slightly. SPPM forecast the closing price of 2019-03-19 and 2019-03-20 at 207.16 and 207.22, respectively, which were very close to the actual observed value.

Table 1 Forecast results of ‘Shanghai Composite Index’ (parameter setting: k=5, N=10, M=2)

The date of	The real value (N/A)	The real price	NN1(h=8)	NN2(h=6)	NN3(h=4)	NN4(h=2)	NN5(h=3)	Predictive value (N/A)	Forecast as
2019-03-16	2404.74								
2019-03-19	2410.18	rose	2355.0	2363.0	2345.6	2400.3	2409.3	2374.64	fall
2019-03-20	2376.84	fall	2352.7	2358.7	2346.0	2402.1	2409.7	2373.84	fall

Table 2 Forecast results of ‘Kweichow Moutai’ (parameter setting: k=5, N=10, M=2)

The date of	The real value (yuan)	The real price	NN1(h=8)	NN2(h=6)	NN3(h=4)	NN4(h=2)	NN5(h=3)	Predictive value	Forecast as
2019-03-16	207.59								
2019-03-19	207.64	rose	207.35	206.67	207.41	207.13	207.24	207.16	fall
2019-03-20	207.75	rose	207.15	206.85	207.56	207.71	206.84	207.22	rose

Figure 4 shows the comparison curve between the predicted results and the actual results of SPPM on the Shanghai Composite Index and Kweichow Moutai D1 dataset, respectively. On the whole, SPPM is better than Kweichow Maotai in fitting the Shanghai Composite Index. The possible reason is that compared with individual stocks, the SSE index is less likely to be controlled by a few institutions, i.e. the SSE index more realistically reflects the overall law of the market. In other words, the running regularity of the Shanghai Composite Index is easier to be captured and learned by SPPM. And because of the reason of speculation, even if the law of the stock was captured, the predicted value may also have a gap with the actual value. The worse the fitting effect is, the less the component of market regularity is reflected, and the more difficult the SP-PM is to predict accurately.

Fig. 4 Comparison curve between predicted SPPM value and actual value

4 Conclusion

This paper presents a backpropagation mathematical model of stock price prediction, SPPM, which can predict the stock price in the future, up to several days. Due to the integration of multiple neural networks, the predicted prices have a high degree of accuracy. The experimental results prove this. Future work includes: (1) Studying the relationship between N and M on prediction accuracy; (2) The relationship between sample size and prediction accuracy.

References

- [1] Laughlin J M. Some new transformations for Bailey pairs and WP-Bailey pairs[J]. *Central European Journal of Mathematics*, 2020, 8(3):474-487.
- [2] Abozaid A A, Selim H H, Gadallah K, et al. Periodic orbit in the frame work of restricted three bodies under the asteroids belt effect[J]. *Applied Mathematics and Nonlinear Sciences*, 2020, 5(2):157-176.
- [3] Bing H, Xu Y, Hu J. Crank–Nicolson finite difference scheme for the Rosenau–Burgers equation[J]. *Applied Mathematics & Computation*, 2018, 204(1):311-316.
- [4] Kanna M, Kumar R P, Nandappa S, et al. On Solutions of Fractional order Telegraph Partial Differential Equation by Crank-Nicholson Finite Difference Method[J]. *Applied Mathematics and Nonlinear Sciences*, 2020, 5(2):85-98.
- [5] Pochinka O V, Shubin D D. On 4-dimensional flows with wildly embedded invariant manifolds of a periodic orbit[J]. *Applied Mathematics and Nonlinear Sciences*, 2020, 5(2):261-266.
- [6] Alghamdi M H, Alshaery A A. Mathematical Algorithm for Solving Two–Body Problem[J]. *Applied Mathematics and Nonlinear Sciences*, 2020, 5(2):217-228.
- [7] Saouli M A. Existence of solution for Mean-field Reflected Discontinuous Backward Doubly Stochastic Differential Equation[J]. *Applied Mathematics and Nonlinear Sciences*, 2020, 5(2):205-216.

- [8] Khalifeh M H, Yousefi-Azari H, Ashrafi A R. The first and second Zagreb indices of some graph operations[J]. *Discrete Applied Mathematics*, 2019, 157(4):804-811.
- [9] V Medvedev, Zhuzhoma E. Any closed 3-manifold supports A-flows with 2-dimensional expanding attractors[J]. *Applied Mathematics and Nonlinear Sciences*, 2020, 5(2):307-310.
- [10] Malkin M I, Safonov K A. Monotonicity and non-monotonicity regions of topological entropy for Lorenz-like families with infinite derivatives[J]. *Applied Mathematics and Nonlinear Sciences*, 2020, 5(2):293-306.

This page is intentionally left blank