

Stock Price Trend Prediction Based on RBF Neural Network and Artificial Fish Swarm Algorithm

Wei Yanming^{1,a}, Gan Xusheng^{2,b}, Lei Lei^{3,c}

¹ XiJing College, Xi'an, Shaanxi, 710123, China;

² Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an, Shaanxi, 710051, China;

³ School of Business Administration,

Henan University of Economics and Law, Zhengzhou, Henan, 050046, China

^a350293122@qq.com; ^bganxusheng123@163.com; ^clyx421@yahoo.com.cn

Keywords: RBF Neural Network, Artificial Fish Swarm Algorithm, Stock Price Trend, Prediction.

Abstract. Stock market is a complex nonlinear dynamic system, the traditional stock price prediction method is difficult to reveal its inherent law, and the prediction error is larger. Based on this, a prediction method based on artificial Fish Swarm Algorithm (FSA) and RBF Neural Network (RBFNN) is proposed to predict the stock price trend. In method, firstly, a dynamic adjustment method to the algorithm parameters: visual field and movement step is introduced to improve the search capability of AFS, and then modified FSA is used to train the RBFNN model. The simulation shows that, the proposed method is better than BPNN and RBFNN in prediction accuracy for stock price trend. It provides an effective and feasible method for stock price prediction.

1. Introduction

With the development of the economy and the enhancement of people's investment consciousness, the stock investment has become an important way for many families and to individual management income. However, the stock market has the coexistence characteristics of high risk and high-yield. Therefore, establishing the stock price trend prediction model with high rate and accuracy has the theoretical significance and practical value for investors.

2. RBF Neural Network

In 1985, Powell proposed the Radial Basis Function (RBF) method of multivariable interpolation. In 1988, Broomhead and Lowe first applied RBF to neural network design, compared RBF and multi-layer neural network, Revealed their relationship, and constituted RBF Neural Network (RBFNN) [1].

RBFNN is simple in structure, simple in training, and has a fast convergence rate and can approximate any nonlinear function. The emergence of RBFNN has brought new life to the research and application of neural network. The excellent characteristics of RBFNN make it more vigorous than BP neural network. It is used in more and more areas.

The basic idea of RBFNN is to use RBF as 'basis' of the hidden element to form the hidden layer space, the input vector is transformed in the hidden layer, transform the input data with low-dimensional pattern into high-dimensional space, so that the linear indivisibility problem in the low dimensional space is linearly separable in the high dimension space [2].

The structure of RBFNN is similar to that of other multi-layer forward network, and it is a three-layer forward network. Its structure is show in Fig. 1. The first layer is the input layer, composed of the signal source nodes; the second layer is the hidden layer, the number of nodes

depends on the problem described; the third layer is the output layer, which responds to the effect of the input pattern. The transformation from the input space to the hidden layer is non-linear, however, the transformation from the hidden layer space to the output space transformation is linear.

It is shown from analysis that, the mapping relation of RBFNN is composed of two parts: (suppose that the input dimension is n , the number of hidden nodes is m , the output dimension is p)

The first part of the mapping is the nonlinear transformation from the input layer space to the hidden layer space, the transformation function can be expressed as

$$h_j(x) = \phi(\|x - c_j\|, \sigma_j) = \exp\left(-\frac{\|x - c_j\|^2}{2\sigma_j^2}\right) \quad (1)$$

where $\phi(\cdot)$ is the transformation function of the hidden element (i. e. RBF), generally can be taken as Gaussian function; $\|\cdot\|$ denotes the norm, and usually can be taken as 2-norm; x is the n -dimensional input vector, that is $X = [x_1, x_2, \dots, x_n]$; c_j is the center of the j -th hidden element; σ_j is the width of nonlinear transformation function in the j -th hidden element.

The second part of the mapping is the linear transformation from the hidden layer space to the output layer space, the output is:

$$f(x) = \sum_{j=1}^m h_j(x)w_j \quad (2)$$

where w_j is the connection weight between the j -th hidden element and the output; m is the number of hidden elements.

In RBFNN, a fixed non-linear transformation is performs in hidden layer. The input space is mapped into a new space. The output layer in the new space realizes the function of the linear combiner. The adjustable parameters are the weight of the linear combiner.

Establishing and training a neural network is to make the mapping function through learning to determine the center c_j , and width d of neuron basis function of each hidden layer, and weight w_j from the hidden layer to the output layer, so as to complete the mapping from the required input to the output. Compared with BP neural network, 3 parameters of the RBF network have different role in the mapping. The center c_j and the width d of the hidden layer represent the relative position between the sample space pattern and the center, which finish the nonlinear mapping from the input space to the hidden layer space, and the weight of the output layer realize the linear mapping from the hidden layer space to the output space. It needs to be clear that the core of RBFNN is the design of the hidden layer, and the appropriate selection of the center c_j and width d will fundamentally affect the final performance of RBFNN.

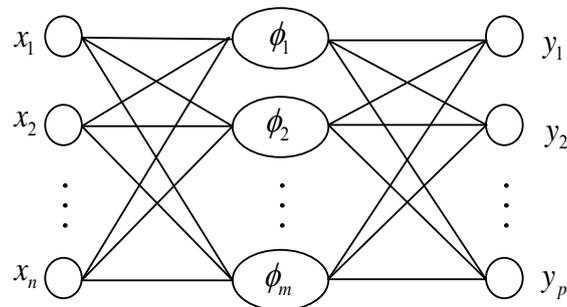


Fig. 1. Basic structure of RBFNN

3. Artificial Fish Swarm Algorithm

AFS algorithm is an intelligent random search algorithm [3][4], its mathematical model can be described as follows: suppose that there is a artificial fish swarm with N individuals in n

dimensions search space, and the state of the artificial fish i can be expressed as $X_i = (x_{i1}, x_{i2}, \dots, x_{in})^T$; the food concentration of current position of the artificial fish i can be expressed as $Y_i = F(X_i)$ where $F(\cdot)$ is the objective function, the distance between the individuals is $d_{i,j} = \|X_i - X_j\|$; $visual$ is the visible domain of the artificial fish, $step$ is the movement step of the artificial fish, δ is the factor of congestion degree; $trynumber$ is the maximum try numbers that the artificial fish feeds every time.

In each iteration, the artificial fish finish itself update by foraging, clustering and pileup behavior etc. to realize the optimization process, these specific behaviors can be described as follows:

1. Foraging behavior: the artificial fish swim about randomly within its visual field. When the food is found, it swims along the direction of gradual increase of the food. Suppose that X_i is the current state of an artificial fish, and a random state X_j is selected in its visible domain ($d_{ij} \leq visual$), when the food concentration of X_j is larger than that of X_i , the artificial fish moves one step along the direction; otherwise, re-selects a random state X_j , and judges if the movement condition is satisfied; after repeat several times, if the condition is not still satisfied, randomly move one step. The formulas can be expressed as:

$$x_{ik}^{next} = x_{ik} + rand(step) \cdot \frac{x_{jk} - x_{ik}}{\|X_j - X_i\|}, \quad Y_j > Y_i \quad (3)$$

$$x_{ik}^{next} = x_{ik} + rand(step), \quad Y_j < Y_i \quad (4)$$

where $k=1,2,\dots,n$, x_{ik}^{next} denotes the next state of x_{ik} , $rand(step)$ denotes the random number in $[0, Step]$.

2. Clustering behavior: in the process of fish swim, in order to ensure its survival and escape from danger, the fishes spontaneously cluster together, the fishes clustering follow three rules: one is the separation rule that can avoid the close fellows too crowded as far as possible; two is the alignment rules that can be consistent with average direction as far as possible; three is the cohesion rule that can move toward the center of close fellows as far as possible.

Suppose that X_i is the current state of an artificial fish, the number of fellows within its visible domain is n_f , then we can have

$$J_i = \{X_j \mid \|X_j - X_i\| \leq visual\} \quad (5)$$

If J_i is a non-empty set, it indicates that there exist other fellows within its visible domain, namely $n_f \geq 1$, then the center position of these fellows can be given by

$$X_{ck} = \sum_{j=1}^{n_f} \frac{x_{jk}}{n_f} \quad (6)$$

where X_{ck} denotes k -th element of X_c ; x_{jk} denotes k -th element of j -th fellow $X_j (j=1,2,\dots,n_f)$. Calculate the food concentration Y_c of the center position, if the condition

$$\frac{Y_c}{\delta} > Y_i, \quad \delta > 1 \quad (7)$$

is satisfied, it shows that the center position X_c has a higher safety degree, and is not also crowded, then we can calculate

$$x_{ik}^{next} = x_{ik} + rand(step) \cdot \frac{x_{ck} - x_{ik}}{\|X_c - X_i\|} \quad (8)$$

Otherwise, the artificial fish conducts the foraging behavior. If J_i is an empty set, it indicates that

there is not other fellows within its visible domain, the foraging behavior can be performed.

3. Pileup behavior: when one or a few fish in the swarm find the food, the close fellows quickly follow to arrive to the position of the food. Suppose that the current state of the artificial fish is X_i and X_{\max} that Y is the largest in the fellows within its visible domain ($d_{ij} \leq visual$) is searched. If $Y_{\max} > \delta \cdot Y_i$, it shows that X_{\max} has a high food concentration, and its surrounding is not too crowded, then we can calculate

$$x_{ik}^{next} = x_{ik} + rand(step) \cdot \frac{x_{\max k} - x_{ik}}{\|X_{\max} - X_i\|} \quad (9)$$

Otherwise, the artificial fish executes the foraging behavior. If there is no other fellow within its visible domain, the foraging behavior can be carried out.

4. Bulletin board: it is used to record the optimal state of the individual in the swarm. After an iterative calculation, the state of each artificial fish is compared to the record in bulletin board. If better, then the record in bulletin board is substituted; otherwise, the bulletin board is unchanged. After the termination of all iterations, the result recorded in bulletin board is the calculated optimal solution.

In view of above-mentioned artificial fish and its behavior, each artificial fish explores itself current environment condition (including changes of the objective function and the fellows) to determine a behavior, and ultimately, the artificial fishes cluster around several local extremum. In general, for solving the maximum problem, the artificial fish with high food concentration is around extremum range that the objective value is larger, and this range can usually gather a lot of artificial fish, which helps to determine and obtain the global extremum.

4. Modified FSA

In the foraging behavior, the individuals of artificial fish always try to move in a better direction, which laid the foundation for the convergence of the algorithm. The artificial fish randomly visits the state of a point X_j within its visual field. If the state is better than the current state X_i , the state X_i moves a step to the state X_j ; if the state X_j is worse than the state X_i , it continues to search the state within its visual field. If it reaches a certain search times *try-number* and still does not find a better state, then do random swimming. In the above process, the step size *step* and the visual field *visual* have great influence on the speed and precision of the convergence of FSA. Because of improper setting, the algorithm may fall into the local extreme or not reach the accuracy. In the foraging behavior, when the individual of artificial fish does not find a better state, it will randomly select a new state, but it cannot make full use of the information obtained, resulting in increased computational complexity and slow convergence. In view of the above shortcomings, a modified FSA (MFSA) is introduced to train the RBFNN model [5].

The change of visual field has a great influence on the convergence of algorithms. When *visual* is very small, the foraging behavior of artificial fish and the random swimming are more prominent, at this time, the local search ability of the algorithm is strong. When *visual* is very large, the rear-end behavior and clustering behavior of the artificial fish are more prominent, at this time, the global search ability of the algorithm is strong and fast convergence. In addition, the impact of *step* cannot be ignored, if *step* is large, the convergence is fast, but it may be oscillating occasionally; if *step* is small, the convergence is slow, but it can obtain the solution of high precision.

It can be known from the analysis that, the more difficult to optimize the function, the more imperative to strengthen the global search capabilities. Once the approximately optimal location is positioned, it needs to improve the local search capabilities to enhance the fine search. Therefore, the dynamic adjustment method of *visual* and *step* can be used to improve the performance of the algorithm. In the early period of the algorithm, it can be used to improve the global search performance and convergence speed of the algorithm, so that the artificial fish roughly search in a large range, with the search, *visual* and *step* gradually are reduced. In the later period of the algorithm, the global search is gradually evolved into local search, and the fine search is performed

near the optimal solution, which improves the local search ability and optimization precision of the algorithm. $visual$ and $step$ can be dynamically adjusted by

$$\begin{cases} visual = visual \cdot a + visual_{\min} \\ step = step \cdot a + step_{\min} \\ a = \exp[-30 \cdot (t/T_m)^s] \end{cases} \quad (10)$$

where $visual_{\min}=0.001$, $step_{\min}=0.0002$, $step=visual/8$, t is the current iteration number, T_m is the maximum number of iterations. The initial value of $visual$ is usually taken as $Z_m/4$ (Z_m is the maximum value of the search range). $visual$ and $step$ function consists of three segments. The algorithm firstly keeps the maximum value, and then gradually becomes smaller, and finally remains the smallest. This method is a good way to balance the global search ability and local search ability, and can improve the convergence speed and improve the accuracy of the algorithm. The change rate of the value of the function a from large to small is an integer (greater than 1), and is usually in the range $[1, 30]$. Fig. 3 shows the change curves of the function a at $T_m = 1000$ and $s = 3, 6, 12$, respectively.

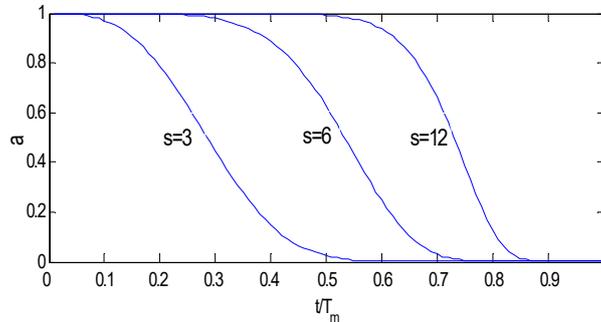


Fig. 2. Change curve of function a

5. MAFS Learning Algorithm of RBF Neural Network

MAFS algorithm is applied to conduct the learning of RBFNN model, namely optimize the parameter vector $\theta = \{c_j, \sigma_j, w_j\}$ of RBFNN. Its simplified learning process is shown in Fig.3, where the termination condition is the maximum number of iterations or target error. Before training RBFNN, the network parameters to be optimized need to be coded, and each dimension correspond to a parameter, then the number of dimensions of search space $N=(n+1)m+(m+1)p+2m$.

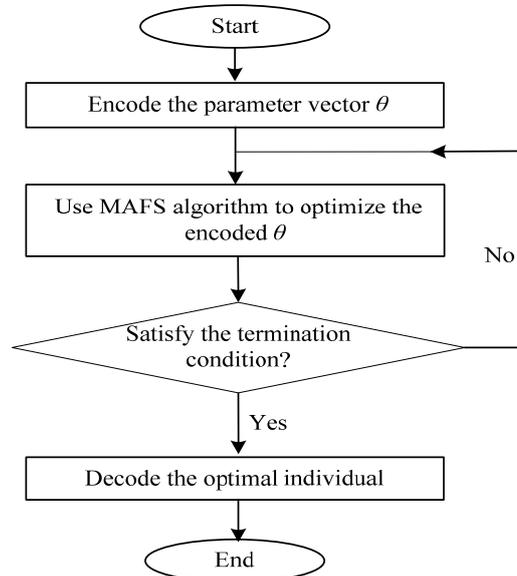


Fig. 3 Learning process of MAFS-RBFNN

6. Experiment Simulation

To verify MAFS-RBFNN prediction method for stock price trend, the experimental data is from: 10/22/2009 ~ 07/18/2014 from Dow Jones Index (U S A). To facilitate the modeling, 15 features are applied as the inputs of MAFS-RBFNN, the output of MAFS-RBFNN is the stock price trend T , which can be defined by the percent yield R , namely Drop: $R \in (-\infty, -0.5\%]$, Stability: $R \in (-0.5\%, 0.5\%)$ and Rise: $R \in [0.5\%, \infty)$, R can be calculated by

$$R_t = (P_t - P_{t-1}) / P_{t-1} \times 100\% \quad (11)$$

where P_t is the series of stock price.

According to the learning process in Fig. 3, the stock price trend prediction model based on MAFS-RBFNN is established and tested. The performance comparison between BPNN, RBFNN and MAFS-RBFNN is shown in Table1. Compared with BPNN and RBFNN models, the proposed MAFS-RBFNN model has obvious advantages in stock price trend prediction.

Table 1 Performance Comparison between Different Models for Dow Jones Index

Assessment index	BPNN	RBFNN	MAFS-RBFNN
Training time (s)	21.5	12.0	14.5
RMSE	188.20	152.15	135.48
MAD	195.68	170.33	152.06
MAPE	4.84	4.25	4.07
DS%	60.95	63.70	65.02
CP%	58.74	62.55	66.51
CD%	61.75	62.09	65.14

7. Conclusion

To predict the stock price trend accurately, RBFNN prediction method based on MAFS is proposed. In the method, AFS is used to optimize the parameters of RBFNN, namely complete the training of RBFNN, and a dynamic adjustment method to the parameter *visual* and *step* is introduced to improve the traditional AFS. The simulation result shows that MAFS-RBFNN method has the superiority in prediction accuracy for stock price trend.

References

- [1] D. E. Rumelhart, J. L. McClelland, "Parallel distributed processing," MIT Press, Cambridge, 1986.
- [2] S. Chert, P. M. Crant, C. F. N. Cown, "Orthogonal least square algorithm for radial basis function networks," IEEE Transaction on Neural Networks, vol. 2, No. 2, 1991, pp. 302-309.
- [3] X. L. Li, Z. J. Shao, J. X. Qian, "An optimizing method based on autonomous animats: fish-swarm algorithm," Engineering Theory & Practice, vol. 22, No. 11, 2002, pp. 32-38.
- [4] X. L. Li, J. X. Qian, "Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques," Journal of Circuits and System, vol. 8, No. 1, 2003, pp. 1-6.
- [5] L. G. Wang, Y. Hong, F. Q. Zhao, D. M. Yu, "Improved artificial fish swarm algorithm," Computer Engineering, vol. 34, No. 19, 2008, pp. 192-194.