■ 699

Application of Artificial Fish Swarm Algorithm in Radial Basis Function Neural Network

Yuhong Zhou[1], Jiguang Duan[2], Limin Shao[1]*
[1]College of Mechanical and Electrical Engineering, Agricultural University of Hebei,
Baoding 071001, Hebei, China
[2]Institute for Nationalities Attached to Hebei Normal University, Shijiazhuang 050091, Hebei, China
*Corresponding author, e-mail: shaolm@126.com

Abstract
Neural network is one of the branches with the most active research, development and application in computational intelligence and machine study. Radial basis function neural network (RBFNN) has achieved some success in more than one application field, especially in pattern recognition and functional approximation. Due to its simple structure, fast training speed and excellent generalization ability, it has been widely used. Artificial fish swarm algorithm (AFSA) is a new swarm intelligent optimization algorithm derived from the study on the preying behavior of fish swarm. This algorithm is not sensitive to the initial value and the parameter selection, but strong in robustness and simple and easy to realize and it also has parallel processing capability and global searching ability. This paper mainly researches the weight and threshold of AFSA in optimizing RBFNN. The simulation experiment proves that AFSA-RBFNN is significantly advantageous in global optimization capability and that it has outstanding global optimization ability and stability.

Keywords: RBFNN, Artificial Fish Swarm, Applied Optimization

1. Introduction

Artificial neural network is the artificial intelligence technology developed by referring to the biological neural network to simulate the biological process of human brain. RBFNN is a special three-layer feed-forward local approximation network with single hidden layer. It takes the radial basis function as the base of the hidden neuron to form the space of the hidden layer [1]. Its learning algorithm and structure principle are greatly different from those of BP network and it overcomes the shortcomings of BP (Back-Propagation) network to certain extent. To every training sample, it has strong real-time, fast learning speed and extraordinary convergence. RBFNN has been applied to every research field and it has made some progress in robotic intelligent control, pattern recognition, computer image processing and expert system [2].

The study of neural network can be dated back to the 1940s. W.Mc Culloch, an American psychologist and W. Pitts, a mathematician have used the mathematical model of formal neuron to simulate the activity function of biological neurons for the first time and it has such characteristics as the preliminary self-learning, parallel processing and distributed memory [3]. The Hopfieid artificial neural network model raised by Professor Hopfieid from California, USA in 1982 has created quite an upsurge in artificial neural network computer. D.E. Rumelhart and J.L.McClelland have come up with BP algorithm in the year of 1986, which has become the most influential network learning algorithm to date [4]. After that, many scholars have been dedicated to the research of various neural network algorithms so as to obtain the neural network topology and parameters that can satisfy the practical applications, however, because people's research on the biological nervous system is still in the exploratory stage, artificial neural network model cannot realize all the practical functions of human brain, therefore, the theoretical exploration of artificial neural network model is still to be further developed. For the RBF neural network, there are still a lot of problems that are not solved, the performance of the neural network is mainly determined by its structure, the training process of RBF neural network is the process of determining the hidden layer structure and the connection weight values. In the past, the structure of RBF network has to be selected by trial and correcting method, the

---

calculation quantity is large, the speed is low and it is difficult to evaluate whether the network structure is better. This selection method has seriously hindered the application of RBF network. How to improve the operational efficiency and reduce the structure complexity is worth further exploration in the RBF network research area[5].Together with the continuous development of swarm intelligence optimization algorithms in recent years, the research to utilize swarm intelligence algorithms to optimize the neural network structure is still being expanded and explored. To apply swarm intelligence algorithms in the design and training of neural network has been deemed as the complicated optimization problem in multi-dimensional space and every solution means a neural network with different structures and connection weights. AFSA has become a frontier research topic in swarm intelligence algorithm[6].

By using AFSA to optimize the parameters of RBF neural network, we can get the global optimal solution of the RBF parameter set, AFSA can optimize the number of hidden layer nodes in the training process, and then the network structure can be determined. This paper mainly investigates the application of AFSA in RBFNN, makes appropriate adjustments and improvements and uses it in the training of RBFNN for the purpose of using the adaptive ability, parallelism and globality of AFSA to better solve the parameter optimization problem of RBFNN and improve its overall performance. The simulation experiment proves the effectiveness and practicability of the idea and suggestion proposed in this paper.

2. Radial Basis Function Neural Network

RBFNN is a typical local approximation artificial neural network and a forward network constituted by the neurons of input layer, hidden layer and output. Its basic idea is to use the radial basis function as the base of the neuron in the hidden layer and form the space of that layer. The hidden layer transfers the input vectors to transfer the low-dimensional mode input data into the high-dimensional space so as to make the linearly inseparable problems in low dimension linearly separable in the high-dimensional space. RBFNN takes the Euclidean distance between the input vector of the training sample and the weight vector of the nodes in the hidden layer as the input and as a static neural network, it matches the function approximation theory and has the only optimal approximation point. Due to its advantages like simple optimization process, fast training speed and optimal approximation ability, RBFNN can approximate any continuous function as long as there are sufficient neurons in the hidden layer and the units in the hidden layer is the perception units. There are three parameters which affect its network performance, including the weight vector in the output layer, the center of the neurons in the hidden layer and the width (variance) of the neurons in the hidden layer. In the neural network training process, inappropriate parameter selection will cause insufficient fitting and over-fitting. The basis function has several forms: multi-quadratic function, inverse multi-quadratic function, spline function and Gaussian function. To select the iterations and network structure (namely the number of the neurons in the hidden layer) reasonably is key to train the applied neural network model and it will directly affect the prediction accuracy. Generally, the learning of radial basis network usually starts from the network weight and then gradually adjusts the other parameters of the network. The weight bears direct relevancy to the center and width of the neuron [7]. Its structure chart is seen in Figure 1.
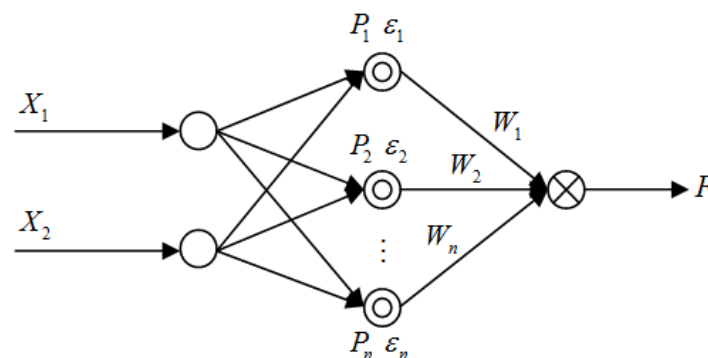

Figure 1. RBF Neural Network Structure

RBFNN is made up of an input layer, a hidden layer including the neurons of a radial basis function (normally, it is Gaussian function) and an output layer of linear neurons. The center of the unit basis function in the hidden layer is P_j, the width is ε_j and the connection weight between the units in the hidden layer and those in the output layer is W_j. Because it has simple structure and its neurons have small sensitive zones, it can be extensively applied in the local approximation of non-linear functions [8].

3. Behavior Description and Principle of Artificial Fish Swarm Algorithm

As an optimization algorithm based on the simulation of the behaviors of fish swarm, artificial fish swarm algorithm (AFSA) is evolved from the fish's behavior in searching for food in the water and it introduces the idea of intelligence algorithm based on behaviors. There are three fish behaviors in the water: preying, swarming and following. This algorithm starts from the behaviors to construct artificial fish and reaches the global optimum through the search of all the individuals in the fish swarm[9]. The descriptions of these behaviors are as follows:

AFSA includes variables and behavior functions, including: X is the AF of the current position, $step$ is the motion step length, $visual$ is the visual distance, $trynumber$ is the number of tries, $detal$ is the congestion factor and $best$ is the range with the highest food within all the areas. $\mathbf{X}_{best}$ is the best status of all AF regions, $\mathbf{X}_c$ is the center position of the artificial fish and $\mathbf{X}_{\min}$ is the position of artificial fish minimum distance. $Y = F(X)$ is the food concentration of the artificial fish . AF's behaviors include Preying, Swarming and Following.

(1) Preying: the artificial fish swims freely in the water and when it finds the food, it will swim towards the food. AFSA extends this fish behavior into the solution of practical optimization problems, namely to get close to the better optimal value gradually [10].

X_i is the current status of the artificial fish and select a status X_j randomly from the visual distance.

$$\mathbf{X}_j = \mathbf{X}_i + visual \Box Rand() \tag{1}$$

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^t + step \Box Rand() \Box \left(\frac{\mathbf{X}_j - \mathbf{X}_i^t}{\left\| \mathbf{X}_j - \mathbf{X}_i^t \right\|} + \frac{\mathbf{X}_{best} - \mathbf{X}_i^t}{\left\| \mathbf{X}_{best} - \mathbf{X}_i^t \right\|} \right) \tag{2}$$

(2) Swarming: the artificial fish swarm cluster together. Every fish maintains a certain distance with other fish to avoid congestion with and keeps the same direction with the neighborhood fish. The artificial fish searches the number of the current fish, calculates the center position of current fish and moves towards to center [11].

X_i is the current status of the artificial fish, the center of artificial fish has low fitness value and the surrounding environment is not very congested and then the artificial fish moves one step towards to vector sum of $\mathbf{X}_c$ and $\mathbf{X}_{best}$.

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^t + step \Box Rand() \Box \left(\frac{\mathbf{X}_c - \mathbf{X}_i^t}{\left\| \mathbf{X}_c - \mathbf{X}_i^t \right\|} + \frac{\mathbf{X}_{best} - \mathbf{X}_i^t}{\left\| \mathbf{X}_{best} - \mathbf{X}_i^t \right\|} \right) \tag{3}$$

(3) Following: in the fish swarm, the artificial fish searches the optimal position of the surrounding neighborhood. When one or more fish finds food, the adjacent fish will get to the food source following it or them quickly. If the objective function value of the optimal position is bigger than that of the current position and it is not very congested, then the current position moves one step towards the optimal neighborhood fish [12].

X_i is the current status of the artificial fish, artificial fish searches its neighborhood and finds a smaller Y_j.

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^t + step \square Rand() \square \left(\frac{\mathbf{X}_j - \mathbf{X}_i^t}{\left\| \mathbf{X}_j - \mathbf{X}_i^t \right\|} + \frac{\mathbf{X}_{\min} - \mathbf{X}_i^t}{\left\| \mathbf{X}_{\min} - \mathbf{X}_i^t \right\|} \right) \tag{4}$$

The above-mentioned will convert to each other at different moments. Such conversion is usually autonomously realized by the fish's perception to the environment. These behaviors are closely related to the preying and survival of the fish. The flowchart of AFSA is indicated as Figure 2.
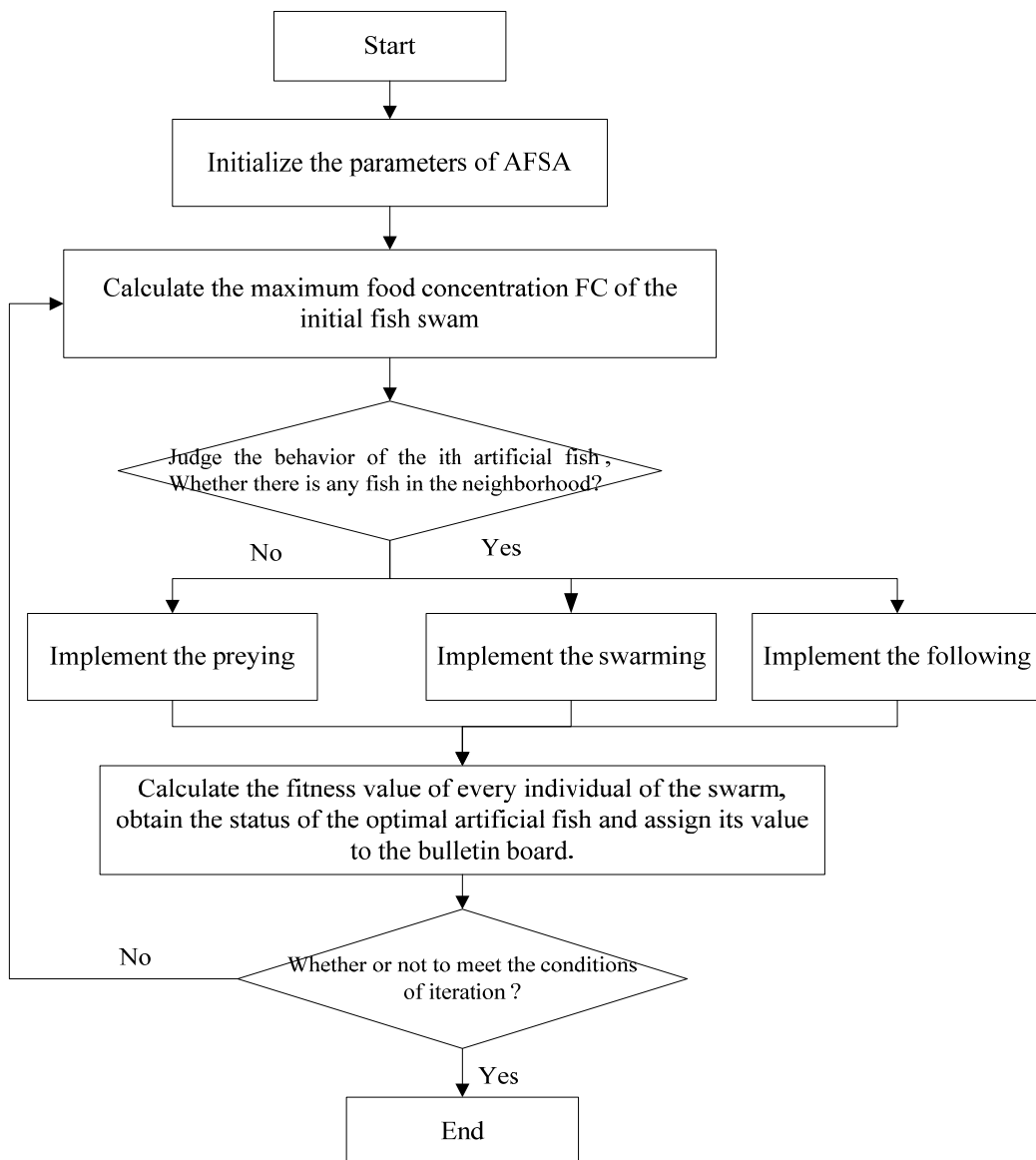
Figure 2. The flowchart of AFSA

4. Application of Artificial Fish Swarm Algorithm in RBFNN

RBFNN is a forward network with excellent performance. It has the optimal approximation performance. The hidden layer is constituted by a group of radial basis functions. What is relevant to every node in the hidden layer are the parameter vector and the width [13]. The combination mode of AFSA and RBFNN is shown as Figure 3.
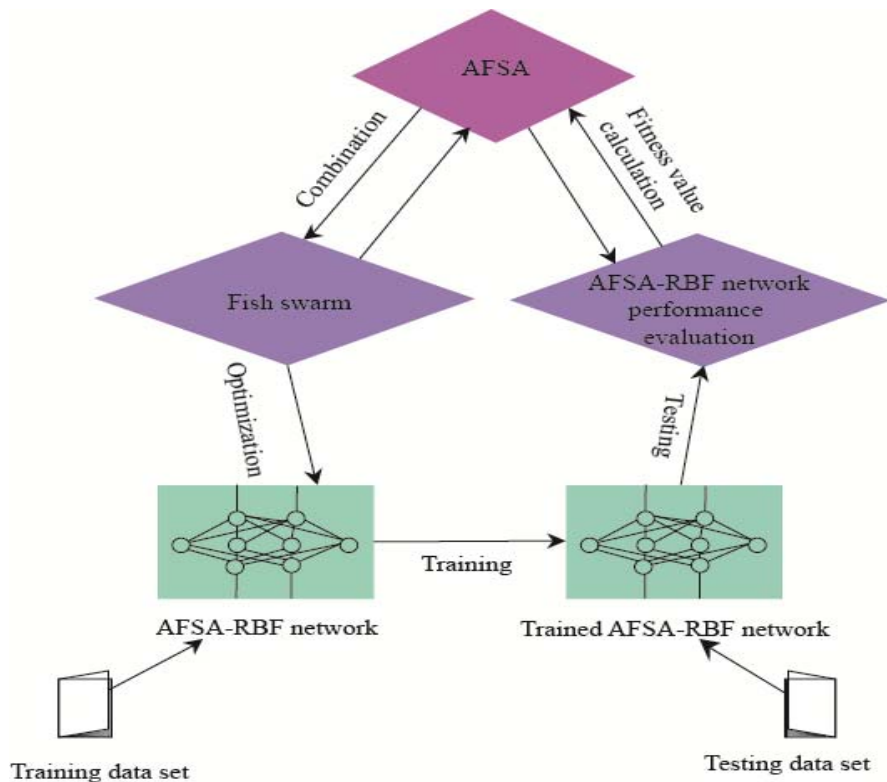
Figure 3. The combination diagram of AFSA and RBFNN

The learning of the common RBFNN always starts from the network weight and then gradually adjusts to other parameters of the network since the weight is closely related to the center and width of the neurons. The transfer function of the hidden layer selects Gaussian function, namely $R_i(x) = \exp[-\dfrac{\|x - c_i\|^2}{2\delta_i^2}]$.

Here, x is a n-dimensional input vector; c is the center of the basis function and it is the vector with the same dimensions as x and δ determines the width that the basis function surrounds the center point. The output network structure of RBFNN is 10-5-1, namely that there are 10 nodes, 5 nodes and 1 node in the input layer, hidden layer and output layer respectively. The output obtained is the prediction value of the prediction moment.

This paper applies AFSA into the parameter adjustment of RBFNN and detailed steps for the specific algorithm can be divided into three phases.

Phase I: Initialization phase of AFSA-RBFNN.

Step 1: Generate the training sample $F = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$.

Step 2: Clarify the number of neurons in every layer and the network objective error.

Step 3: Set the parameters of the fish swarm, including the size of the fish swarm m, the maximum number of iterations gen, the perception range of the artificial fish $visual$, the maximum shift step length $step$.

Phase II: Train the neural network model by using AFSA with coupling prior information.

Step 1: The individuals update themselves through preying, swarming and following and generate new fish swarm.

Step 2: Calculate the fitness value of every individual of the swarm; obtain the status of the optimal artificial fish and assign its value to the bulletin board.

Step 3: Evaluate all the individuals. If a certain individual is better than the bulletin board, then replace the bulletin board with this individual.

Step 4: When the optimal solution in the bulletin board reaches the satisfactory error bound, the algorithm ends; otherwise, turn to Step 1.

Phase III: The Retraining of RBFNN.

Step 1: Correspond the global optimal value obtained from the iteration of AFSA to the weight and threshold of RBFNN.

Step 2: Continue to train the neural network with RBF algorithm until the objective error is reached.

5. Simulation Experiment and Analysis

This paper uses AFSA to optimize the center, width, weight and threshold of RBFNN and establishes the model (AFSA-RBF) of RBFNN to be optimized by artificial fish swarm. The computer hardware CPU in the experiment uses Intel Core 2 Duo E8400 3.0GHz with an internal memory of 4G and the software uses MatlabR2012a simulation platform. This experiment uses two test functions to verify the RBFNN algorithm to be optimized by improved AFSA. The number of dimensions is D and the two functions are shown as Table 1.

Table 1. Test Function Set

$$f_1 = \sum_{i=1}^{D} 2x_i \qquad\qquad x_i \in [-2,2], D = 5,8$$

$$f_2 = 2x_i + 12\sin(6x_i) + 9\cos(5x_i) \qquad x_i \in [0,15], D = 5,8$$

This paper takes 300 groups of 5- and 8-dimensional data from the two functions respectively. Among the 300 groups of data, randomly select 270 groups of data to train the neural network and the rest 30 groups of data as the prediction samples. Figure 4 and Figure 7 are the comparison diagram of the error change curves in the training processes of AFSA-RBFNN and RBFNN corresponding to the 5- and 8-dimensioanl input data of the two functions respectively.
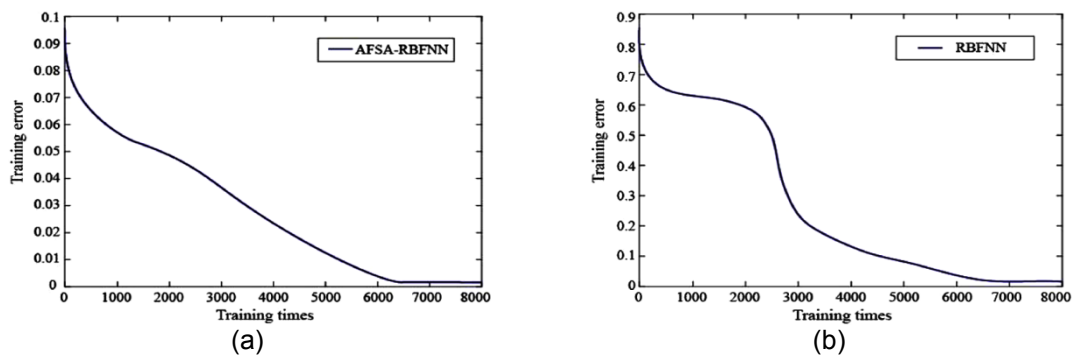

(a)   (b)

Figure 4. Comparison of the test experiment results of 5-dimensional f1 function

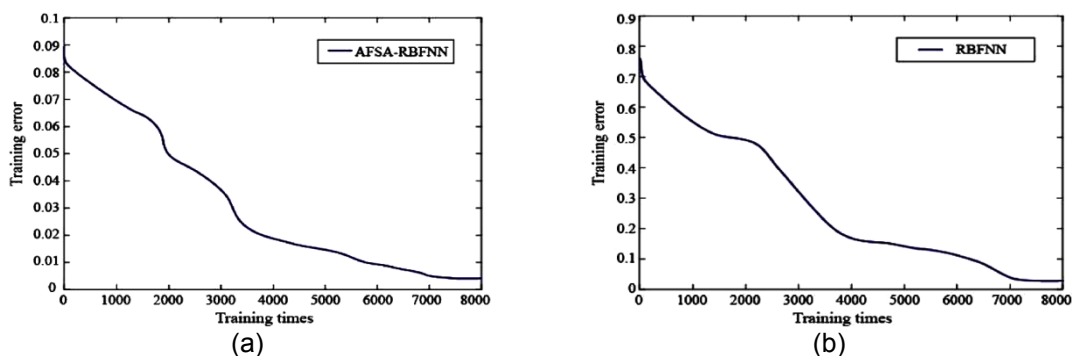(a)   (b)

Figure 5. Comparison of the test experiment results of 8-dimensional f1 function
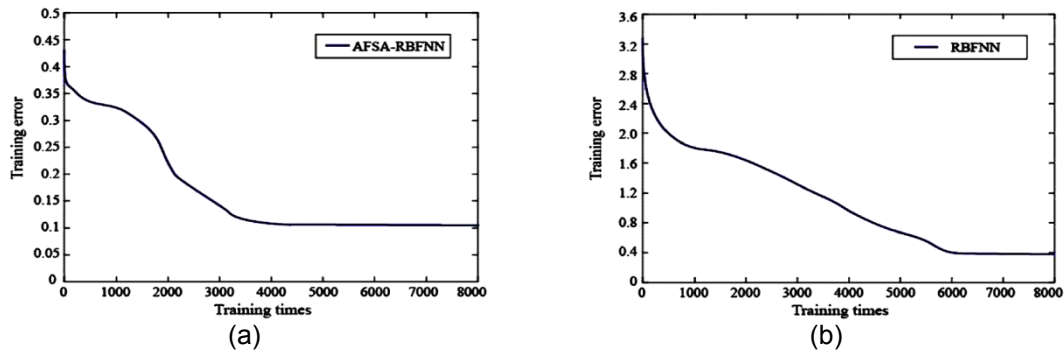
Figure 6. Comparison of the test experiment results of 5-dimensional f2 function
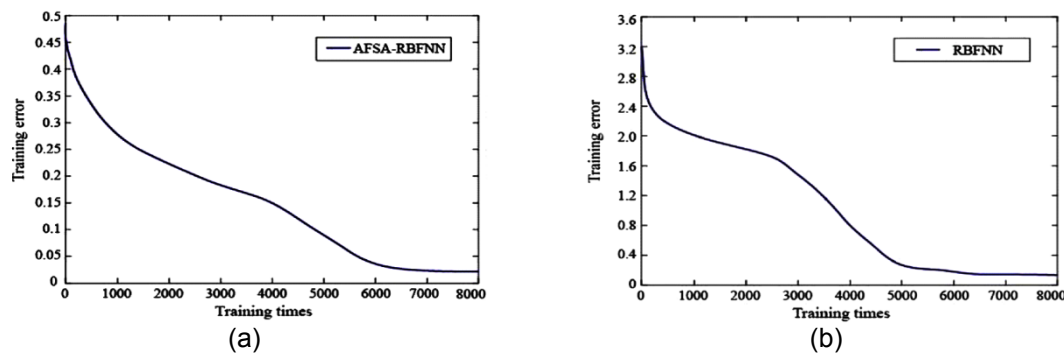

Figure 7. Comparison of the test experiment results of 8-dimensional f2 function

In RBF neural network, the number of hidden layer nodes and the choice of center and width are the key to the whole network performance, directly impact on the network approximation ability. It doesn't need to make any assumptions of the center and width of RBF function, using AFSA to learn the network connection weights, and can determine the network topology, the hidden neural unit location and the corresponding width value.

From the training error curve comparison diagrams of AFSA-RBFNN and RBFNN, it can be seen that AFSA-RBFNN enhances the learning efficiency of the network weight, increases the convergence, avoids vibration of RBFNN in the training process and makes the network incapable of convergence. It is quite effective to extract prior information from the data by using AFSA-RBFNN to improve the search efficiency of the algorithm. Since AFSA can search the global optimal value accurately, the location and number of neurons in RBFNN can be determined automatically in the process of parameter optimization, it improves the accuracy of the trained network classification model and the convergence speed, meanwhile enhances algorithm efficiency and generalization performance. Using AFSA to optimize the structure of RBFNN, it's effective to improve the efficiency of network.

6. Conclusion

Artificial fish swarm algorithm has many characteristics, including excellent ability to search the global extremum, strong robustness, easiness to be realized and parallel processing capability. To solve the deficiency of RBFNN, this paper introduces AFSA, which has strong global optimization ability, to train RBFNN and solve such problems of traditional RBFNN as that it is easy to get trapped into local optimum. It can be seen from the experiment result that AFSA-RBFNN can enhance the network learning accuracy and has better global convergence performance.

Acknowledgements

References

[1] Sufang Li, Mingyan Jiang. The New Complex-Valued Wavelet Neural Network. *TELKOMNIKA Telecommunication, Computing, Electronics and Control*. 2014; 12(3): 613-621.

[2] Saeid Asgari Taghanaki, Ahmad Hatam, Behzad Bahraminejad. Synthetic Feature Transformation with RBF Neural Network to Improve the Intrusion Detection System Accuracy and Decrease Computational Costs. *International Journal of Information and Network Security (IJINS)*. 2012; 1(1): 28-36.

[3] Patricia Melin, Fevrier Valdez. Genetic Optimization of Neural Networks for Person Recognition Based on the Iris. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(2): 309-320.

[4] Susmita Mall, S Chakraverty. Chebyshev Neural Network Based Model for Solving Lane–Emden Type Equations. *Applied Mathematics and Computation*. 2014; 247(15): 100-114.

[5] Hu Xiaoping, Wang Chao, Zhang Wenhui, Ma Jing. Self-learning PID Control for X-Y NC Position Table with Uncertainty Base on Neural Network. *TELKOMNIKA Telecommunication, Computing, Electronics and Control*. 2014; 12(2): 343-348.

[6] Amir Hossein Zaji, Hossein Bonakdari. Performance Evaluation of Two Different Neural Network and Particle Swarm Optimization Methods for Prediction of Discharge Capacity of Modified Triangular Side Weirs. *Flow Measurement and Instrumentation*. 2014; 40(12): 149-156.

[7] Buse Melis Ozyildirim, Mutlu Avci. Logarithmic Learning for Generalized Classifier Neural Network. *Neural Networks*. 2014; 60(12): 133-140.

[8] Paweł Pławiak. An Estimation of The State of Consumption of A Positive Displacement Pump Based on Dynamic Pressure or vibrations using Neural Networks. *Neuro Computing*. 2014; 144(20): 471-483.

[9] Md Abul Kalam Azad, Ana Maria AC Rocha, Edite MGP Fernandes. Improved Binary Artificial Fish Swarm Algorithm for the 0–1 Multidimensional Knapsack Problems. *Swarm and Evolutionary Computation*. 2014; 14(2): 66-75.

[10] M Fernanda, P Costa, Ana Maria AC Rocha, Edite MGP Fernandes. An Artificial Fish Swarm Algorithm Based Hyperbolic Augmented Lagrangian Method. *Journal of Computational and Applied Mathematics*. 2014; 259(15): 868-876.

[11] Weiling Zhu, Jingqing Jiang, Chuyi Song, Lanying Bao. *Clustering Algorithm Based on Fuzzy C-means and Artificial Fish Swarm*. Procedia Engineering. 2012; 29: 3307-3311.

[12] Na Fang, Jianzhong Zhou, Rui Zhang, Yi Liu, Yongchuan Zhang. A Hybrid of Real Coded Genetic Algorithm and Artificial Fish Swarm Algorithm for Short-term Optimal Hydrothermal Scheduling. *International Journal of Electrical Power & Energy Systems*. 2014; 62(11): 617-629.

[13] Hrvoje Krstić, Željko Koški, Irena Ištoka Otković, Martina Španić. Application of Neural Networks in Predicting Airtightness of Residential Units. *Energy and Buildings*. 2014; 84(12): 160-168.