International Journal of Control and Automation
Vol. 8, No. 7 (2015), pp. 159-166
http://dx.doi.org/10.14257/ijca.2015.8.7.17

Integrating Particle Swarm Algorithm and Artificial Fish Swarm Algorithm to Optimize BP Algorithm

Zhengli Zhai and Mingyue Pang

[1]*College of Computer Engineering, Qingdao Technological University, Qingdao, Shandong 266520, China*
[2]*College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China*
zhaizhl@163.com

Abstract

A strategy which using the particle swarm algorithm improved by the artificial fish swarm algorithm to optimize the BP (Back propagation) algorithm was proposed. It can conquer the shortcomings that the convergence rate of BP is too slow and it is easy to fall into local extreme value, and can improve the learning ability of BP neural network. Finally, the improved algorithm has been used to analysis the earthquake prediction. The results of simulation and test show that the optimized algorithm can improve the predicting accuracy of the BP network.

Keywords: *Artificial Swarm, Particle Swarm, BP algorithm, Earthquake Prediction*

1. Introduction

BP algorithm is one of the most frequently used learning method in neural network, it uses the back propagation of the network error to adjust the network link weight and node threshold in order to achieve the desired effect of minimizing the sum of differences between actual output and expected output of the network [1], BP algorithm has the characteristics such as simple structure, strong plasticity, good adaption, self-learning, fault-tolerance, and so on. Though the BP algorithm has the absolute advantage, it often presented some questions in practical use, for example, easy to trap into local minimum, weak generalization ability, slow convergence rate, *etc*[2]. Many algorithms had been proposed to improve the algorithm and the particle swarm algorithm is also put up as one of the improved algorithm, which is a kind of swarm intelligence global search optimize algorithm, owning the following advantages: robustness, rapidity, interoperability and global convergence. The basic idea of particle swarm improving BP is that the weight and threshold of the BP network can be adjusted using the particle swarm algorithm, *i.e.*, the present location of the particle in the swarm is used as the weight and threshold of the network to calculate, and the error of the BP network is used as fitness value of the particle swarm algorithm, which is the judgment of the update of the article swarm algorithm. However, the traditional particle swarm algorithm can be lead to the situation trapped into local extreme value because of the “aggregate” of the particle in the progress of searching the optimization. The improved Particle Swarm optimization algorithm based on the Artificial Fish swarm (AFPSO for short) has been proposed, it utilizes the aggregate of the artificial fish swarm to step out of the local extreme value, updates the network parameters of BP network by using the improved particle swarm algorithm so that it can conquer the “aggregate” phenomenon within limits, enhance the learning ability, convergence rate and the ability of stepping out of local extreme of the BP network.

ISSN: 2005-4297 IJCA

2. Preliminary Knowledge

2.1. BP Neural Network Algorithm

BP algorithm is the training algorithm of multilevel network. Here we do the study based on the three-level BP network: input, hidden, and output layer, the adjacent layer processing full connection, the neuron in the same layer being connectionless. The number of nodes of the input and output layer is confirmed by the input and output of the practical problem and the number of nodes of the hidden layer is got by the formula $N_H = \sqrt{N_I + N_O} + L$ ($L = 0,1,\cdots 10$, where N_I means the nodes of input layer, N_O means the nodes of output layer) [3].

The training progress of the BP algorithm mainly includes two steps: the first step is the forward propagation of the input sample, the sample value introducing from the input layer, going through the hidden layer and outputting through the output layer at last, the data was processed by the every level, including weighting multiply and calling the activation function to get the output value of every neural cell [4]; the second step is the back propagation of the error signal, there must be the error between practical output got by the first step and the target output, the error is defined as:

$$E = \frac{1}{2}\sum_{k=1}^{m}(t_k - o_k)^2 \qquad (1)$$

Where, m is the number of nodes of the output layer, t_k and o_k are the target output value and the practical output of output neuron k respectively.

BP algorithm adjusted the connection weight between the neurons in the way of minimizing the error, the change of the link weight value is:

$$\Delta w_k = \eta \frac{\partial E}{\partial w_k} \qquad (2)$$

Where, η is the learning rate, its value is the random among [0, 1]. If the error value meets the conditions that stop the training, then the output can be thought satisfied and stop training. Otherwise, the back propagation training will be go on until meet stop conditions.

2.2. Particle Swarm Optimization

Particle swarm optimization (PSO) [5] is a bionic optimization algorithm proposed by Kennedy and Eberhart in American in 1995. In PSO, each potential value of question to be solved is a "particle" in the search space, each particle has a fitness decided by the question, and the speed of each particle decides its direction and location, and then particle searching the global optimal solution in the space according to the present optimal particle. The basic idea of PSO describes as following: in the d-dimensional searching space, there is a particle swarm with scale m, if the location of the ith particle in the d-dimensional space is $x_i = (x_{i1}, x_{i2} \cdots x_{id})$, whose speed is $v_i = (v_{i1}, v_{i2} \cdots v_{id})$, and its individual extremum is $p_i = (p_{i1}, p_{i2} \cdots p_{id})$, global extremum is $p_g = (p_{g1}, p_{g2} \cdots p_{gd})$. The ith particle updates its speed and location according to the following formula:

$$v_{ij}^{k+1} = w v_{ij}^{k} + c_1 r_1 (p_{ij}^{k} - x_{ij}^{k}) + c_2 r_2 (p_{gj}^{k} - x_{ij}^{k}) \qquad (3)$$

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1} \qquad (4)$$

Where, $j = (1, 2, \cdots, d)$, the parameter $w \in [0,1]$ is the inertia weight; c_1 and c_2 are the learning factor; r_1 and r_2 are the random among [0, 1] [6].

3. The Improvement of the BP Algorithm Based on the Artificial Fish Swarm and Particle Swarm

3.1. Artificial Fish Swarm Algorithm

Artificial fish swarm algorithm (AFSA), a behaviorism artificial intelligence algorithm, is proposed through inspired by fish swarm behavior, owning the ability that conquer local extremum and get the global extremum [7]. The basic fish swarm algorithm mainly uses the forage of the fish, bunching and tailgating to finish optimizing procedure. The bunching makes the artificial fish trapped into the local extremum to gather toward the artificial fish trend to global extremum, so that it can escape local extremum; the tailgating makes the artificial fish trapped into the local extremum to followed in the wake of the global optimal artificial fish to escape local extremum [8].

The individual location of artificial fish is represented as vector $X = (x_1, x_2, \cdots x_n)$; the food concentration of present location is $Y = f(X)$, where, Y is the target function value; the distance between individuals is $d_{i,j} = \left\| X_i - X_j \right\|$; $Visable$ is the perception distance of the artificial fish; $Step$ is the moving step-size of the artificial fish; δ is the crowding degree factor. Here we mainly introduce the bunching of the artificial fish:

If the present state of artificial fish is X_i, searching the number of partner n_f in present neighborhood ($d_{i,j} < Visable$) and the center location X_c, if the food concentration Y_c in center location is larger than the food consistency Y_i in the location the artificial fish lives ($Y_c > Y_i$) and $\frac{Y_c}{n_f} > \delta Y_i$, it shows that partner center owning more food and it have more spare space and then artificial fish X_i step a step size step toward the partner center location X_c; the formula describes as following:

$$X_{\text{next}} = X_i + Rand() \times Step \times \frac{X_c - X_i}{\left\| X_c - X_i \right\|} \tag{5}$$

3.2. Description of BP Algorithm Based on the Artificial Fish Swarm-Particle Swarm (AFPSO-BP)

The implementation steps of AFPSO-BP algorithm:
Input: the input and output of training samples
Output: the BP neural network after training
Step 1: initializing parameters

1) Initializing BP neural network, confirming the number of neuron of every level, training number, training target and so on.
2) Initializing particle swarm, initializing the scale of particle swarm, the location of the particle, and the dimension of the particle (decided by the weight and threshold of the BP neural network), iterations number, inertia weight w, speed bound, learning factor c_1 and c_2, and so on.
3) Initializing the parameter of the artificial fish swarm, initializing the try number *try_number*, perception distance $Visable$, crowding degree factor δ, step size

$step$, and so on.

Step 2: pass the initial location of the particle to BP neural network as the initial weight and threshold of BP network.

Step 3: do the iteration, taking the error of the BP network as the fitness of the particle swarm. Confirm the individual extremum and the global extremum of the particle: for every particle *i* in generation *j*, calculating its fitness $P_{i,j}$, noting the serial number of the particle owning smallest fitness as *I*, doing the bunching algorithm of artificial fish swarm for the particle *I*, judging whether the particle *I* has been trapped into local extremum or not. If it meets the condition that $n_f > 0$ and the fitness of the center location X_c satisfies $\frac{Y_c}{n_f} > \delta Y_i$ and $Y_c < Y_I$, then step forward one step toward partner center location to get to X_{next} , X_{next} is the optimal location of the generation *j*; otherwise, the optimal location of generation *j* is X_I . Comparing the individual extremum $P'_{i,j}$ of generation *j* with global optimal value P_g , if $P'_{i,j} < P_g$, then $P_g = P'_{i,j}$, recording the location of the present best particle.

Step 4: updating the speed and location of every particle, and judging whether the speed and location of the particle after updating is out of range or not.

Step 5: checking if the conditions of stopping the algorithm iteration or the error requirements is satisfied, stopping the algorithm, the weight and threshold of every dimension of global optimum P_g is what is required. Otherwise, go to **step 3**.

4. Experiments Analysis

This section selects the earthquake data [9] as sample, according these data, we use the 7 prediction factors Significantly affecting the result of earthquake as the input vector. These 7 factors are the earthquake accumulated frequency (short for A) happening in the half year and whose scale no less than 3, the accumulated value the energy released in half year (short for B), b value (short for C), the number of abnormal earthquake swarm (short for D), the number of earthquake zone (short for E), whether in active time (short for F), the earthquake scale about the earthquake zone (short for G), predicting the earthquake scale as the output vector (short for H). We have collected 17 samples, among which the first 7 item are the input of sample, the last one is the output of sample. In order to make the identification accuracy not influenced by the network, we use the formula $x' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$ $(i = 1, 2, 3 \cdots, n)$ before the training test. The earthquake prediction studying sample normalized shows as the Table 4.1.

Table 4.1 Earthquake Prediction Studying Sample

sample	A	B	C	D	E	F	G	H
1	0	0	0.62	0	0	0	0	0
2	0.3915	0.4741	0.77	0.5	0.5	1	0.3158	0.5313
3	0.2835	0.5402	0.68	0	0.5	1	0.3158	0.5938
4	0.6210	1.0000	0.63	1	0.5	1	1.0000	0.9375
5	0.4185	0.4183	0.67	0.5	0	1	0.7368	0.4375
6	0.2160	0.4948	0.71	0	0	1	0.2632	0.5000

7	0.9990	0.0383	0.75	0.5	1	1	0.9474	1.0000
8	0.5805	0.4925	0.71	0	0	0	0.3684	0.3750
9	0.0810	0.0692	0.76	0	0	0	0.0526	0.3125
10	0.3915	0.1230	0.98	0.5	0	0	0.8974	0.6563
11	0.0270	0.0742	0.62	0	0	0	0.2105	0.1875
12	0.1755	0.3667	0.77	0	0.5	1	0.7368	0.4062
13	0.4320	0.3790	0.68	0.5	0	1	0.2632	0.4375
14	0.4995	0.4347	0.63	0	0	1	0.6842	0.5938
15	0.6885	0.5842	0.67	0.5	0.5	1	0.4211	0.6250
16	0.5400	0.8038	0.71	0.5	0.5	1	0.5789	0.7187
17	0.1620	0.2565	0.75	0	0	1	0.4737	0.3750

The 17 samples given above is divided into training sample and testing sample, among which, the sample data of row 1-12 are the training samples, used to train the weight and threshold of the BP neural network, the row 13-17 are the testing samples. We do simulation with these samples by utilizing standard particle swarm BP algorithm and AFPSO-BP respectively. The hidden nodes of BP network is 30; the scale of the particle swarm N=50, the biggest iterations number *itmax*=200; the trying number *try_number*=50, the perception distance $Visable$ =40, crowding factor δ = 41.618, the step size $step$ =10.25. The result of the experiment shows as the following picture in Figure 4.1.

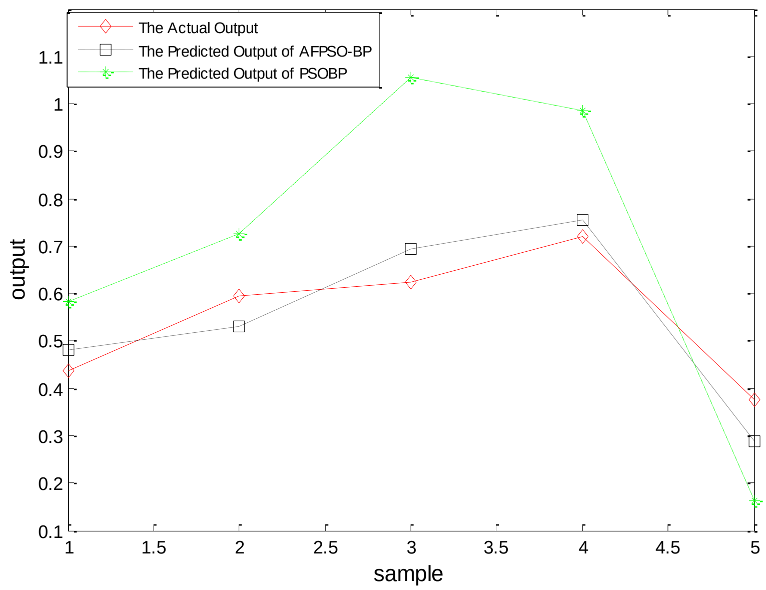

Figure 4.1. Comparison of Earthquake Prediction Output of Sample in Row 13-17

Figure 4.1 shows the comparison diagram of earthquake prediction, from which it can be seen that the improved algorithm AFPSO-BP put forward by us is closer to the practical output curve than the standard PSOBP. The Figure 4.1 shows that the error of AFPSO-BP is small in comparison. In order to compare the size of the error between AFPSO-BP and PSOBO, we compute the predicting error of the two algorithms through simulation and do the comparison, showing as Figure 4.2.

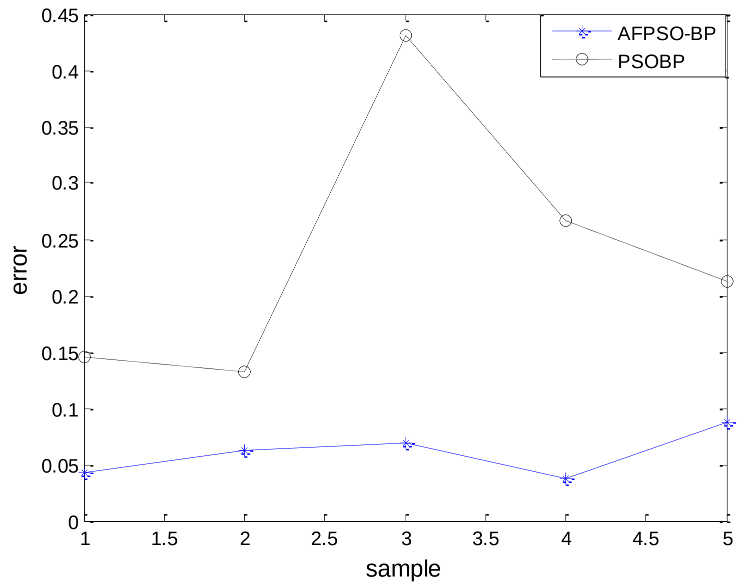

Figure 4.2. Comparison of Predicting Error of the Five Test Samples

It can be seen obviously that the predicting error of the standard PSOBP is bigger than AFSPO-BP, and the distribution of the errors of 5 samples is uneven, from which a conclusion can be got that the improved AFPSO-BP have improved in predicting error and error distribution, not only made the error smaller but also made them distributed even, hence improved the studying ability and the ability of jumping out of the local extremum. Besides, the improved algorithm also improved in convergence speed, the convergence curve of two algorithms showing as Figure 4.3 and Figure 4.4.

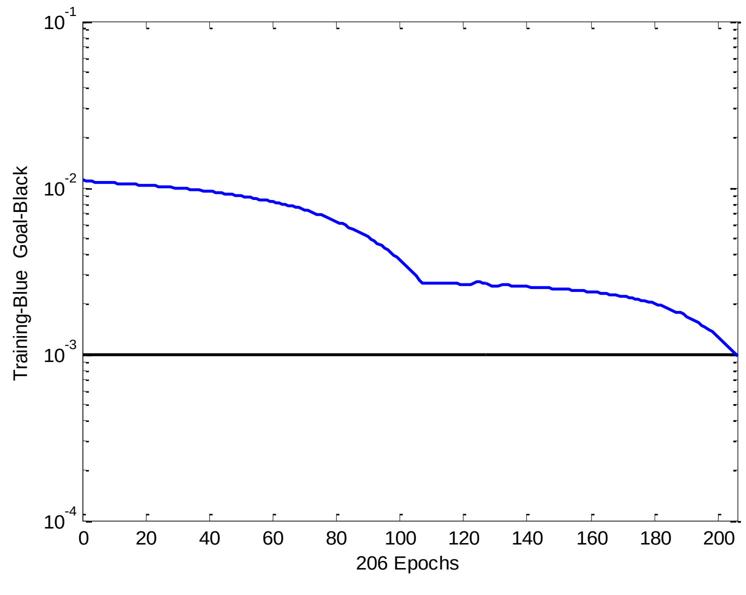

Figure 4.3. Convergence curve of PSOBP

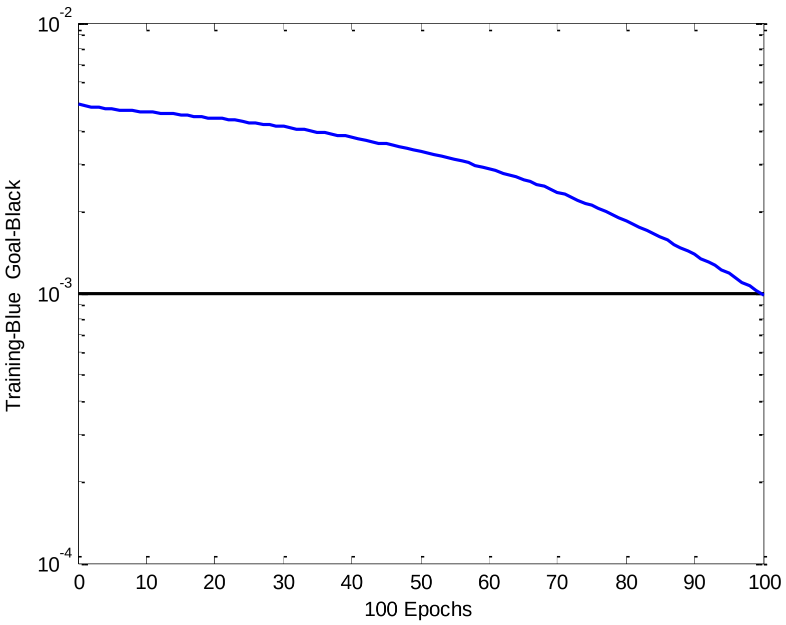

Figure 4.4. Convergence Curve of AFPSO-BP

By comparing the Figure 4.3 and Figure 4.4, it can be seen that when in the same convergence target, the convergence speed of the improved AFPSO-BP is much faster than the standard PSOBP. From the result of the simulation, the improved algorithm achieved a certain optimization result.

5. Conclusion

The improved algorithm presented by us adds the artificial fish swarm algorithm to the particle swarm BP algorithm, through taking advantage of the bunching of the artificial fish to conquer the traditional algorithm's disadvantage of easy to trap in local extremum. Through the simulation, the improved algorithm have improved the predicting accuracy of the BP network.

Acknowledgement

This work was supported by the National Basic Research Program of China "973 Program" (No.2010CB328101), National Nature Science Foundation of China (No. 61173181), Qingdao Science and Technology Program (No.12-1-4-4-(7)-JCH, No. KJZD-13-31-JCH), Excellent Youth Scientist Research Reward Fund of Shandong Province (No. BS2010DX009), Shanghai Nature Science Foundation (No. 13ZR1443100).

References

[1] W. W. Sun, "Study on Improved Algorithm and Application of BP neural network", Chongqing University, **(2009)**.
[2] L. Jiang, "Research on BP Neural Network Learning Based on Particle Swarm Optimization and Simulated Annealing Algorithm", Anhui University, **(2013)**.
[3] X. L. Shen, H. Y. Zhang and J. S. Zhang, "Improved Particle Swarm Optimization Algorithm", Computer system applications, vol. 19, no.2, **(2010)**, pp.57-60.
[4] Z. L. Jiang, "Introduction to Artificial Neural Networks", Beijing: Higher Education Press, **(2003)**.
[5] J. Kennedy, "Particle swarm optimization", Encyclopedia of Machine Learning, Springer US, **(2010)**, pp. 760-766.
[6] X. B. Sun, "Study on the Fusion of BP Neural Network Algorithm with other Algorithms and its Application", Chongqing University, **(2011)**.

[7] X. M. Ma and N. Liu, "Improved artificial fish-swarm algorithm based on adaptive vision for solving the shortest path problem", Journal on Communications, vol.35, no.1, **(2014)**, pp.1-6.
[8] Y. B. Zheng, J. J. Liu and Y. Wang, "Improved artificial fish swarm algorithm based on social learning mechanism", Journal of Computer Applications, vol. 33, no.5, **(2013)**, pp.1305-1307, 1329.
[9] Z. X. Ge and Z. Q. Sun, "Neural Network Theory and Matlab R2007 implementation", Beijing: Electronic Industry Press, **(2007)**.

Authors

Zhengli Zhai, he received the B.S degree in thermal applied mathematics from Shandong University, Ji'nan, China, in 1994, and M.S. degree in computer software and theory from Shandong University of Science and Technology, Tai'an, China, in 2003, and Ph.D. degree in computer application technology from University of Science and Technology Beijing, Beijing, China, in 2007. In 1994, he became a Research Fellow in Linyi University, Linyi, China. Currently, he is an associate Professor with College of Computer Engineering, Qingdao Technological University, Qingdao, China, the M.S. supervisor from 2008. His research interests include Petri net theory and applications, Grid/Cloud Computing, intelligent information processing, multi-agent systems.

Mingyue Pang, she received the B.S degree in Computer science and technology from Binzhou University, Binzhou, China, in 2012. She is currently pursuing the M.S degree in Software Engineering from College of Computer Engineering, Qingdao Technological University, China. She won the State Scholarship and the outstanding graduate Award. Her research interests include Grid/Cloud Computing, intelligent algorithms, multi-agent systems.

