An Improved PSO-BP Neural Network and Its Application to Earthquake Prediction

Cao Li [1], Xiaoyu Liu [1]

1. School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081
E-mail: liu_autumn@sina.com

Abstract: This paper presents a way of combining BP (Back Propagation) neural network and an improved PSO (Particle Swarm Optimization) algorithm to predict the earthquake magnitude. It is known that the BP neural network and the normal PSO-BP neural network have some defeats, such as the slow convergence rate, easily falling into local minimum values. For improving the properties of PSO, some proposed the linear decreasing inertia weight strategy. Furthermore, this paper uses a nonlinear decreasing inertia weight in PSO to get a faster training speed and better optimal solutions. Compared with the linear decreasing strategy, the inertia weight in our nonlinear method has a faster declining speed in the early iteration, which can enhance the searching precision. In the late iteration, the inertia weight has a slower declining speed to avoid trapping in local minimum value. Then we apply the improved PSO to optimize the parameters of BP neural network. In the end, the improved PSO-BP neural network is applied to earthquake prediction. The simulation results show that the proposed improved PSO-BP neural network has faster convergence rate and better predictive effect than the BP neural network and the normal PSO-BP neural network.

Key Words: BP neural network, Particle Swarm Optimization Algorithm, Inertia Weight, Earthquake Prediction

1 Introduction

Earthquakes can bring different degrees of damage to society and results in loss of life and property. So the earthquake prediction becomes an important subject in the field of seismological research. BP neural network[1] has the strong nonlinear processing ability, so it is widely used in earthquake prediction research[2]. However, in the optimization process of BP neural network, the gradient descent algorithm , also called the BP algorithm, is easy to cause the BP neural network trapping into local minimum values and has relatively poor generalization ability, in addition, its robustness is bad and its convergence speed of the learning process is slow.

In view of the shortcomings of BP neural network, some suggest to use the PSO (Particle Swarm Optimization) [3] algorithm to optimize the parameters of BP neural network. The PSO algorithm is a new-type swarm intelligence optimization algorithm[4]. The algorithm strategy is simple and easy to be implemented. The parameters of the algorithm which needs to be adjusted are less. But at the same time it has slow convergence rate and it is easy to fall into local minimum values.

So this paper puts forward to improving the PSO algorithm. It is known that the inertia weight ω of PSO has great influence on optimization performance[5]. The inertia weight ω is adjusted by a nonlinear decreasing strategy which makes the inertia weight ω has a faster declining speed in the early period. Such inertia weight adjustment can enhance the local search ability of PSO. In the late period of the algorithm, the inertia weight ω has a slower declining speed which can slows the declining speed of global search ability. Such adjustment is beneficial to jump out of the local minimum values for particles.

In this paper, an improved PSO algorithm is combined with BP neural network, which is used to optimize the parameters of BP neural network. To verify the improved PSO-BP neural network, we apply it to the earthquake prediction. The simulation results show the effectiveness of the proposed algorithm.

2 BP neural network

BP neural network is a multilayer forward network of a one-way transmission. It is a neural network of three layers or more than three layers, such as the input layer, the hidden layer(s) and the output layer. A typical structure of the three-layer-network is shown in Figure 1. The vector $\overline{W}_1$ and $\overline{B}_1$ are the weights and the threshold values between the input layer and the hidden layer. The vector $\overline{W}_2$ and $\overline{B}_2$ are the weights and the threshold values between the hidden layer and the output layer. The number of the input layer nodes and the output layer nodes are determined according to the actual situation. The number of hidden layer nodes is generally obtained by experience formula. The commonly used transfer functions of the BP neural network are sigmoid logarithmic function and sigmoid tangent function or linear function[1].

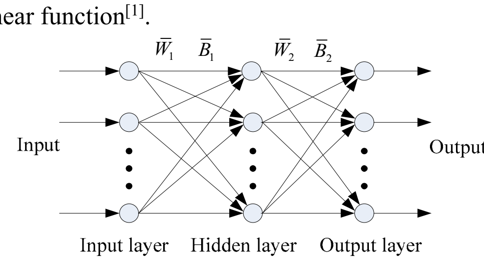


Figure 1: The Structure of Three-Layer-BP Neural Network

The BP neural network usually uses the gradient descent algorithm to adjust the weights and the threshold values, thus has a few drawbacks: 1) The learning process has slow convergence speed; 2) The training network is easy to trap into the local minimum; 3) The structure of the network is difficult to determine; 4) It is difficult to guarantee the network's generalization ability.

3 Particle swarm optimization algorithm

3.1 Theory of PSO algorithm

The mathematical description of the basic particle swarm optimization algorithm is as follows:

In a D dimension search space, a swarm contains N particles. The *i-th* particle at the *t-th* iteration has a current position $X_i(t)=\left(X_{i1}(t),X_{i2}(t),\cdots,X_{iD}(t)\right)$ and a current velocity $V_i(t)=\left(V_{i1}(t),V_{i2}(t),\cdots,V_{iD}(t)\right)$. Each particle memorizes its own best position, called the local best position $P_i(t)=\left(P_{i1}(t),P_{i2}(t),\cdots,P_{iD}(t)\right)$. And it is also has the best position among all particles, called the global position $G(t)=\left(G_1(t),G_2(t),\cdots,G_D(t)\right)$. Each particle updates its own speed and position according to the following evolution equations [6].

$$V_{id}(t+1)=\omega V_{id}(t)+c_1r_1\left[P_{id}(t)-X_{id}(t)\right]+c_2r_2\left[G_{id}(t)-X_{id}(t)\right] \quad (1)$$

$$X_{id}(t+1)=X_{id}(t)+V_{id}(t+1) \quad (2)$$

where $i\in\left[1,N\right]$; d denotes the *d-th* dimension of the particle, $d\in\left[1,D\right]$; t denotes the *t-th* iteration; c_1 and c_2 are the acceleration coefficients [7], $c_1,c_2\in\left[0,2\right]$; ω is the inertia weight; r_1 and r_2 are the two random numbers in the range $\left[0,1\right]$; the velocity $V_i(t)$ is usually limited to a certain range $V_i(t)\in\left[-V_{\max},V_{\max}\right]$; at the same time the searching space of the particles is limited within a certain range $X_i(t)\in\left[-X_{\max},X_{\max}\right]$.

3.2 The improved PSO (IPSO)algorithm

In the basic PSO algorithm, the inertia weight ω can be changed to control the convergence and the exploring ability of the basic particle swarm optimization algorithm. The value of ω determines the current velocity of the particle. A large inertia weight can make the particle have a quick velocity which helps to enhance the global search ability of the algorithm. A small inertia weight can make the particle have a slow velocity which helps to enhance the local search ability of the algorithm. Researchers have proposed linear decreasing inertia weight (LDIW) strategy [8], fuzzy inertia weight (FIW) strategy[9] and random inertia weight (RIW) strategy[10] successively. Among these researches, the most commonly used strategy is linear decreasing inertia weight strategy. Such strategy can make the algorithm have a strong global search ability in the early period, and strong local search ability in the late period. The equation is shown as the following:

$$\omega=\omega_{\max}-\frac{t\times\left(\omega_{\max}-\omega_{\min}\right)}{t_{\max}} \quad (3)$$

where $\omega_{\min}$ and $\omega_{\max}$ are the minimum and maximum value of the inertia weight ω respectively. In general, $\omega_{\max}$ =0.9, $\omega_{\min}$ =0.4, t is the current number of iteration, $t_{\max}$ is the maximum number of iteration.

But the linear decreasing inertia weight strategy has some defeats: 1) In the early stage, the local search ability is poor. Even if the particle is close to the global optimal solution, the particle always misses the solution because the particle's velocity is too fast; 2) In the late stage, the global search ability becomes poor. The algorithm is easy to trap into the local optimum.

So this paper proposes a nonlinear decreasing inertia weight strategy. The equation is shown as follows.

$$\omega=\omega_{\max}-\frac{\sqrt{t+1}\times\left(\omega_{\max}-\omega_{\min}\right)}{\sqrt{t_{\max}+1}} \quad (4)$$

Figure 2 shows the decreasing curves of equation (3) and (4). From the curves we can see: 1) In the early iterations, the inertia weight ω adjusted by the nonlinear decreasing strategy declines faster than that by the linear decreasing strategy; 2) In the late iterations, the inertia weight ω adjusted by the nonlinear decreasing strategy declines slower than that by the linear decreasing strategy.

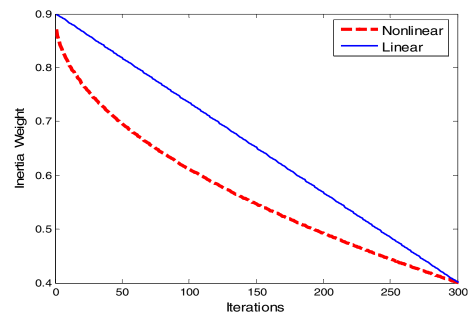


Figure 2: Linear and nonlinear decreasing curves

In the early stage of the improved PSO algorithm, t is small, thus ω is close to $\omega_{\max}$ and the algorithm has strong global searching ability. With the increase of t, the inertia weight ω decreases nonlinearly. The inertia weight ω in the nonlinear decreasing algorithm declines faster than that in the linear decreasing algorithm. So the local search ability of the improved algorithm is stronger. At the same time, the velocity of the particle declines faster. When the particle is close to the global optimal solution, it can search for the optimal solution finely. So that the convergence precision becomes high and the particle is not easy to miss the optimal solution.

In the late stage of the improved PSO algorithm, the inertia weight ω decreases nonlinearly also. But inertia weight ω declines slower than that in the linear decreasing algorithm. It can ensure that the global search ability and diversity slowly weaken. So that the particle can jump out of the local minimum value which can effectively avoid algorithm's trapping in local minimum value.

The improved PSO algorithm proposed in this paper can help to find a better balanced inertia weight between the global search and the local search. In the following statement we call the improved PSO algorithm as IPSO algorithm.

3.3 BP parameter optimization based on IPSO

Without loss of generality, given a BP neural network with three layers. The number of neurons in the input layer, hidden layer and output layer is I, H and O respectively. The weights and the threshold values of the BP neural network are regarded as particles in the searching space. The IPSO algorithm is used to search the best solution of these particles. The combination of BP neural network and IPSO is denoted as IPSO-BPNN in the following.

The optimization procedure of the IPSO-BPNN is described as follows.

Step1: Initialize parameters: the total numbers of particle N; the maximum number of iterations; the minimum mean square error of BP neural network ε=0.05; the dimension of the particle swarm $D = IH + H + HO + O$.

Step2: Select the *MSE* (Mean Square Error) of the BP neural network as the fitness function *Fitness* of the particle swarm.

$$MSE = \frac{1}{k}\sum_{m=1}^{k}\sum_{n=1}^{s}\left(y_{mn} - q_{mn}\right)^2 \tag{5}$$

where y_{mn} is the actual output of the neural network, q_{mn} is the target output, s is the number of the output nodes and k is the number of the samples in the training set.

Step3: Initialize the position and velocity of each particle randomly.

Step4: Update the local best $P_{\mathrm{i}}(t)$ and global best $G(t)$ according to equation(5). Update the particle's velocity $V_{\mathrm{i}}(t)$ and position $X_{\mathrm{i}}(t)$ according to equations (1) and (2). Update the inertia weight ω according to equation (4). Then the number of iterations adds one.

Step5: Judge whether maximum number of iterations $t_{\max}$ or the minimum mean square error ε is attained. If it is attained, perform Step 6, otherwise perform Step4.

Step6: The algorithm is over and we get the global optimal solution G_{best}. It is the optimal solution of the weights and the threshold values of the BP neural network.

4 Modeling and simulation of earthquake prediction based on IPSO-BP

4.1 Modeling of the earthquake prediction

In order to test the effect of IPSO algorithm, this paper chooses the seismic data[11] of a coastal area, located at 117 ° ~ 120 ° east longitude, 22 ° ~ 26 °east longitude, as sample source. According to this data, we extract 6 representative predictive indexes in terms of time, space and intensity, i.e., earthquake frequency, creep, releasing energy, value b (a symbol to measure the levels of seismic activity in a given area), lack of shock, value η (a useful indicator to judge the deflected degree of actual datum and equation G-R).

A three-layer BP neural network in Figure 1 is used to build the earthquake prediction modeling. The BP neural network has 6 input nodes which represent the 6 predictive indexes respectively. The BP neural network has 1 output node, that is, the predictive earthquake magnitude. The number of hidden layer nodes is 15 according to the empirical formula. The transfer function of hidden layer is tan-sigmoid and the output layer's transfer function is log-sigmoid.

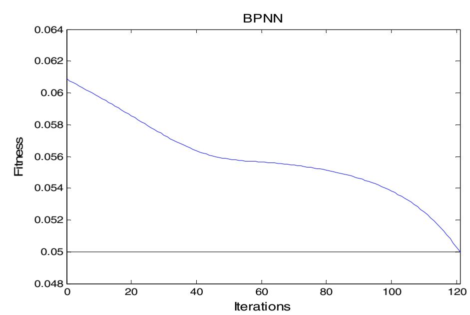
Seismic data has 29 groups in total, where groups 1-20 of the seismic data are chosen as training sample of BP neural network, and groups 21-29 are the test sample of the BP neural network to verify the effectiveness of the earthquake predictor. 29 groups of seismic data have been already normalized.

The PSO and ISPO algorithm have the same initial parameters of the particle swarm: total numbers of particle N=20; acceleration coefficients c_1=c_2=2; $\omega_{\min}$=0.4, $\omega_{\max}$=0.9 ,$t_{\max}$=300 in equation (3) and (4). Then we conduct the simulation.

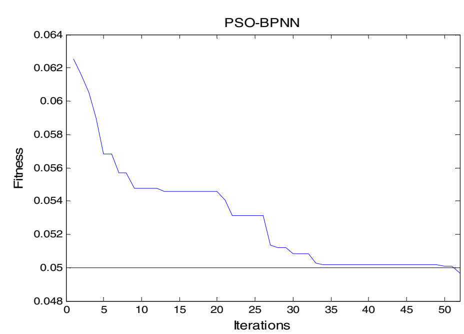
4.2 Analysis of the simulation result

4.2.1 Training result's analysis

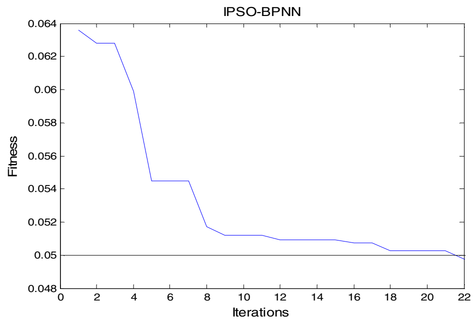
After building the prediction modeling, we start to train the three networks. Figure 3 shows respectively the training convergence result of the basic BP neural network, the BP neural network optimized by PSO with the inertia weight ω declining linearly and the BP neural network optimized by IPSO with inertia weight ω declining nonlinearly. The three networks are called in the following simulation as BPNN, PSO-BPNN and IPSO-BPNN respectively.



(a) Training convergence result of BPNN



(b) Training convergence result of PSO-BPNN



(c) Training convergence result of IPSO-BPNN

Figure 3: The training convergence curves of three networks

From Figure 3 we can get that the BPNN has a slow convergence speed. It needs about 120 training iterations until the minimum mean square error is attained. And the PSO-BPNN has the faster convergence speed. It needs about 51 iterations to attain the minimum mean square error. Compared to the BPNN and the PSO-BPNN, the IPSO-BPNN has the fastest convergence speed. It only needs about 22 iterations to attain the minimum mean square error.

4.2.2 Test result's analysis

After the training procedure, we choose the seismic data, groups 21-29, to test the above three networks and examine their learning performance. Table 1 is the prediction result of BPNN, PSO-BPNN and IPSO- BPNN.

Table 1. Prediction result and error of BPNN, PSO-BPNN and IPSO-BPNN

Group	Actual earthquake magnitude	BPNN		PSO-BPNN		IPSO-BPNN	
		Predictive value	Absolute error	Predictive value	Absolute error	Predictive value	Absolute error
21	4.9	4.322	0.578	4.462	0.438	4.525	0.375
22	3.9	4.286	0.386	4.211	0.311	4.167	0.267
23	4.4	4.191	0.209	4.232	0.168	4.21	0.19
24	5.5	3.993	1.507	4.759	0.741	5.16	0.34
25	4.7	4.388	0.312	4.55	0.15	4.57	0.13
26	4.1	4.48	0.38	4.346	0.246	4.301	0.201
27	5.1	4.401	0.699	4.867	0.233	5.187	0.087
28	4.3	4.186	0.114	4.345	0.045	4.321	0.021
29	4.6	4.021	0.579	4.484	0.116	4.569	0.031

By comparing the predictive values and absolute error of the three kinds of network in Table 1, we can conclude that: compared to the BPNN and the BP neural network optimized by PSO algorithm, the IPSO-BP neural network has a better predictive effect.

5 Conclusion

This paper is based on the BP neural network and an improved PSO (IPSO) algorithm to predict the earthquake magnitude. The BP neural network is used to build the earthquake prediction modeling. The IPSO algorithm is applied to optimize the parameters of the BP neural network. In the IPSO algorithm, the inertia weight declines nonlinearly along the iteration time. The IPSO algorithm can better make up for the shortcomings of trapping in local minimum value and slow convergence speed of BP neural network and PSO-BP neural network. The simulation result shows that compared with the BP neural network and the BP neural network optimized by PSO algorithm. The BP neural network optimized by IPSO algorithm has a better ability to predict the earthquake magnitude.

References

[1] Bingxin Liu, Ning Wang, Dong Zhang. Research on Improved Algorithm based on BP Neural Network in Smart Grid Distribution System. Modern Electronic Technique,Vol.35,No.21,143-144(148),2012.(In Chinese)

[2] Cheng Xu, Jinhui Zhou. Research and Application of BP Neural Network in Earthquake Prediction. Process Automation Instrumentation, Vol. 33, No. 6, 12-14, 2012. (In Chinese)

[3] James Kennedy, Russell Eberhart. Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks, Vol.IV, 1942-1948, 1995.

[4] Haiyan Chen. Task Scheduling in Cloud Computing Based on Swarm Intelligence Algorithm. Computer Science, Vol.41, No.6A, 83-86, 2014. (In Chinese)

[5] Yuhui Shi，Russell C Eberhart. Parameter Selection in Particle Swarm Optimization. Evolutionary Programming(VII), Vol.1447, No.25, 591-600, 1998.

[6] Chunqing Meng. A New Particle Swarm Optimization with Multi-speed Updating Formulas. Journal of Network New Media, Vol.2, No.4, 60-64, 2013. (In Chinese)

[7] Fuling Guan, Lu Dai, Meimeng Xia. Pretension Optimized and Verification Test of Double-ring Deployable Cable Net Antenna based on Improved PSO.Aerospace Science and Technology, Vol 32, No.1, 19-25,2014.

[8] Hanhong Zhu, Yi Wang ,Kesheng Wang. Particle Swarm Optimization(PSO) for The Constrained Portfolio Optimization Problem. Expert System with Applications, Vol.38, No.8, 10161-10169,2011.

[9] Songhua Zhou, Chunjuan OuYang, Changxin Liu, Ping Zhu. Adaptive Fuzzy Particle Swarm Optimization Algorithm. Computer Engineering and Applications, Vol. 46, No.33, 46-48, 2010. (In Chinese)

[10] Yong Feng, Yongmei Yao, Aixin Wang,Comparing with Chaotic Inertia Weights in Particle Swarm Optimization. In: International Conference on Machine Learning and Cybernetics, 329-333, 2007.

[11] Ka Zhu, Zhenglin Wang. Skillful at MATLAB Neural Network. Beijing: Publishing House of Electronics Industry, 378-381, 2010.