

# Failure diagnosis of the steam generator based on improved BP network and Particle

Fujun Qi<sup>1</sup>, Qingsong Niu<sup>1</sup>, Jing Peng<sup>1</sup>, Zhiguang Tian<sup>2\*</sup>

<sup>1</sup>Linyi Electric Power Corporation Liyin, 276000, China;

<sup>2</sup>School of Automation and Electrical Engineering, Linyi University, Linyi, 276000, China

**Abstract** In order to overcome the problems of slow rate of convergence and falling easily into local minimum in BP algorithm, this paper introduces the adaptive particle swarm optimization algorithm and combining model. The paper applies it to steam turbine-generators fault diagnosis. The experiment data shows that the algorithm converges quickly and recognizes faults efficiently; it has a reference value for faults diagnosis.

## 1 Introduction

BP network technology is a ripen one, but it is insufficient in two ways. First, Multilayer perception is a non-linear optimization problem in perspective of mathematics. It adopts the traditional unconstrained minimum method to realize the minimization of the error function, which inevitably leads to some extreme minimization problems in some parts. Second, there is no theoretical guide when choosing the latent layer of network nodes, it is only chosen by experience. So, people propose various kinds of methods to improve it, such as adding the momentum, adjust the studying rate, improving function, and etc., by which certain achievements have been made, but the problem has yet not been solved essentially. Particle swarm optimization (PSO) is a kind of technology which is based on swarm optimization algorithm of Evolutionary Computation and derived from artificial life and evolutionary computation. Realizing the bright prospect of PSO applied in the realms like function optimization, many scholars do researches into this aspect following the footstep of Eberhart and Kennedy. Up to now lots of adaptive algorithm of PSO has been proposed and PSO algorithm is widely applied in the realm of function optimization, neural network practice, classifications of models, fuzzy system control. The advantage of PSO is that it's simple to carry out and has a deeply automatic background. What's more, it is suitable for both scientific research and project apply. A model based on PSO optimizes BP neural network parameter is proposed and the model algorithm about the faults samples of steam turbine generator is built in this paper, based on the consideration above. It proves that the results are much better than the traditional BP neural network.

## 2 Classification of BP neural network faults

Learning process of BP consists of propagate and back propagate. In the course of propagate, the input mode spreads from input layer undergo to output layer through the process of latent layer of neurons, with every layer of neuron influencing the next layer only. If the expected output can not be achieved at the output layer, it will convert to back propagate. In this way, the error signal transmits from output layer to input layer and meanwhile adjust connection right value along the way, so that the error can be monished constantly to the precision required.

Fig 1 is the basic composition of 3 layers of BP neural network .BP network algorithm is as follows:  $W_{ij}$  is the connection right value between the  $i$  pieces of neuron at the input layers and the  $j$  pieces of neuron at the latent layers, and  $V_{kj}$  is that between the  $k$  pieces of neuron at the output layers and the  $j$  pieces of neurons at latent layers.

Course of propagation: Input layer: output value of unit  $i$  is equal to its input value  $x_i$ ; Latent layer: As to  $j$  pieces of latent unit, its output value  $net_j$  is the weighted sum of the output value of former layers

$$net_j = \sum_i w_{ji} o_i + \theta_j \quad (1)$$

Output:  $a_j = f(net_j)$ .  $f$  is function of sigmoid

Output layer. output  $y_k$  is

$$y_k = \sum_j v_{kj} a_j \quad (2)$$

Course of the back propagation Error function  $E_p$  :

\*Corresponding author's e-mail: [tianzhiguang@lyu.edu.cn](mailto:tianzhiguang@lyu.edu.cn)

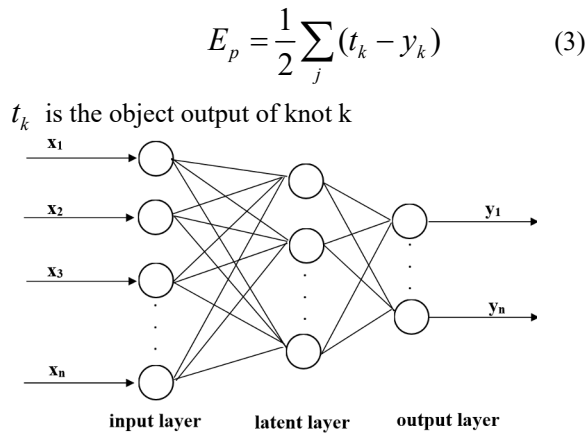


Fig. 1 Artificial neural network

Back propagation adopts a principle of Widrow-hoffdelta, the revision quantum and the square error of connecting right is pro rata relative to change quantum of connecting right, therefore, the revision quantum of the connecting right between the latent layer and output layer is as follows:

$$\Delta w_{kj} = -\eta_{kj} \frac{\partial E^2}{\partial w_{kj}} \quad (4)$$

The revision quantum of connecting right value between the nodes of input layer and latent layer is:

$$\Delta w_{kj} = -\eta_{ji} \frac{\partial E^2}{\partial w_{ji}} \quad (5)$$

$\eta_{kj}, \eta_{ji}$  are learning rate, generally  $\eta_{kj} = \eta_{ji}, 0 < \eta_{kj}, \eta_{ji} < 1$

### 3 Classification of BP neural network faults

The basic idea of PSO is that every problem of optimized potential solution is all the particle which search the space, and every particle has a optimized function to determine its Fitness Value. Also every particle has a rate vector to decide their direction of flying and distance. At first PSO means a group of random particles (random solution), but gradually they find the best answer through iterative. Among every iterative, particles follow two extremes in order to refresh itself. The first one is particle itself could find best solution at that moment, which is called personal best particle; another extreme is the best answer in the whole species at that moment, which is the best answer in the all. Otherwise we can also use part of them instead of the whole species as swarm's neighbor, so the local extreme is equal to all the neighborhood of extreme. We can get that PSO is a kind of search which is based on the individual's corporation and competition to complete the search in complex search space. It also is a kind of evolutionary computation based on swarm intelligence. PSO is like genetic algorithm. It is a tool which is based on swarm optimization. However, PSO doesn't concern the operations of Crossover and Mutation which are used by genetic algorithm. It is the

particle that does the search of solution space follows best particle, so it is easy to realize without the advantages needed to be adjusted by the parameter. In particle swarm optimization, every questions of optimized potential solution is either PSO or through consistent updating extreme points.

Suppose in a searching space about N dimension, there are M particles forming a group, in which number m particles refers to a vector of N dimension  $x_m = (x_{m1}, x_{m2} \dots x_{mN})$ ,  $m = 1, 2, \dots M$ , that is number m particles is located as  $x_m$  in the searching space about N dimension. And every particles' location is a potential answer. Put  $x_m$  to an end function and then you can get the answer, according to which you can figure whether  $x_m$  is the best answer. Number m particles' "flying" speed is also a vector of N dimension, written as  $v_m = (v_{m1}, v_{m2} \dots v_{mN})$ . Take down the best location number m particles has searched so far as  $pb_m = (pb_{m1}, pb_{m2} \dots pb_{mN})$ , the whole particles group searching so far the best location as  $gb = (gb_1, gb_2 \dots gb_N)$ . The basic formula about PSO proposed by the early scholars is:

$$v_{mn}^k = v_{mn}^k + c_1 r_1^k (pb_{mn}^k - x_{mn}^k) + c_2 r_2^k (pb_n^k - x_{mn}^k) \quad (6)$$

$$x_{mn}^{k+1} = x_{mn}^k + v_{mn}^{k+1} \quad (7)$$

In the formula,  $m = 1, 2, \dots, M, n = 1, 2, \dots, N$ ,  $c_1$  and  $c_2$  are the acceleration coefficient, that is to adjust the longest footsteps on flying towards the best particle in the whole group and the best particle of the individuals. If it is too short, the particles may fly far from the destination zone while if it is too long, the particles will fly to the destination zone suddenly or fly over it. Proper  $c_1$  and  $c_2$  could not only accelerate but also restrain from falling the local optimization. Often put  $c_1 = c_2 = 2$ ,  $r_1$  and  $r_2$  are the random numbers among [0,1].

The request of stopping iterative is determined by specific problems, usually iterative is the biggest iterations or use the searched best location to figure out the adaptive number, which satisfies the certain adaptive number and then stops.

Because  $gb$  is the best location of the whole particle group, The PSO is better at the rate but sometimes falls into local best optimization. Formula 6 can be divided into three parts. The first part is the previous speed. The second part is the acknowledgment part of the particle itself, that is the influence on the present location. The third part is the social one, which refers to the  $gb$  of the particle about corporation. PSO is better at the rate of constringency, especially at the early stage of the arithmetic. But there also lies some shortcomings such as low precision and easy radiation and so on. Through research, if the parameters such as

acceleration coefficient and maximum velocity are too big, PSO will properly miss the best solution, and the arithmetic will not constringe; under the circumstance of constringency, all the particles tend to be similar for they all fly to the best solution, thus losing the diversity. As a result, in the later stage, the constringency slows down, and at the same time, when the arithmetic constringes to a certain degree of precision, it can not go on optimizing, which leads to a low precision. In order to better control the exploration and the development of the arithmetic, Shi and Eberhart introduced the inertia weight  $\omega$  for the first time in velocity of PSO, the renewed formula of inertia weight method is as follows:

$$v_{mn}^{k+1} = \omega v_{mn}^k + c_1 r_1^k (pb_{mn}^k - x_{mn}^k) + c_2 r_2^k (pb_n^k - x_{mn}^k) \quad (8)$$

Among the formula,  $\omega$  is not-allowance, which is called momentum, a kind of weight to control speed.

Therefore, the basic PSO is the special of inertia weight  $\omega = 1$ . The introduction of inertia weight  $\omega$  has greatly improved PSO, which lays a solid foundation for the successful settlement of many problems in reality. For this reason, people usually regard PSO with inertia weight as standard PSO. Inertia weight  $\omega$  illustrates the influence of the speed of the last generation upon that of the present generation. Controlling the size of the value can adjust both the overall and partial ability to seek optimization of PSO. Shi found that dynamic inertia weight can achieve a better optimization result than fixed-value. He suggests adopting LDW (Linearly Decreasing Weight). The LDW formula is as follows:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} \times t_n \quad (9)$$

In it,  $\omega_{\max}$  and  $\omega_{\min}$  are respectively the maximization and the minimum of  $\omega$ ;  $t_n$  and  $t_{\max}$  are respectively the present iterative steps and the maximization iterative steps.  $\omega$  controls how the speed of last generation affects the present generation. When  $\omega$  is bigger, the influence becomes greater and the ability of overall search is stronger; when  $\omega$  is smaller, the influence becomes smaller, and the ability of overall search is weaker. Through the adjustment of the value of weight  $\omega$ , we can pick out the local minimum. We can use weight  $\omega$  to control the influence of the speed of the last generation on that of the present generation. The result of experiments illustrates that PSO obtains a higher speed of constringency when  $\omega$  is located in [0.8,1.2]. According to experience,  $\omega$  is set as linear decrease from 0.9 to 0.4, which enables PSO can search a comparatively large area and to locate quickly the proper place of the best solution. With  $\omega$  reducing, the flying speed of particles slows down, and starts to subtle partial searching. This method enhances the speed of constringency, and receives an obvious effect in improving the function of function, but PSO is complicated in practical searching process, so that the LDW can not reflect the real searching process of optimization. In fact,  $\omega$  should vary with the evolution speed of particle swarm and the conglomeration of

particles, namely,  $\omega$  can be indicated as the function of  $h$  and  $s$ ,

$$\omega = f(h, s) \quad (10)$$

$\omega$  should become large as particles enhances conglomeration and becomes small as particles slows down the evolution.

This article adopts the method brought up by Shi that changes the inertia weight by using the fuzzy controller. The improvement of this method is correct the  $\omega$  by fuzzy rules, that is suppose a fuzzy inference engine with 2 inputs, 1 output to change inertia weight. The two input of fuzzy inference engine are the present  $\omega$ , and the Normalized Current Best Performance Evaluation, NCBPE; and output the increase of  $\omega$ . CBPE is the best estimation discovered by the particle swarm optimization so far. CBPE has different definitions, but the general is the best adaptive answer. NCBPE is figured by this:

$$NCBPE = \frac{CBPE - CBPE_{\min}}{CBPE_{\max} - CBPE_{\min}} \quad (11)$$

In it  $CBPE_{\max}$  and  $CBPE_{\min}$  are the above line and the bottom line of NCBPE. The input and output of fuzzy inference engine is: LOW, MEDIUM and HIGH, relating fuzzy membership grade function is left Triangle, Triangle, right Triangle, fuzzy inference engine defined 9 rules to fuzzy inference, determining the present  $w$ . The illustrations of some reference function suggest that particle swarm optimization can all receive satisfactory results, and it also covers much optimization issue. By self-adaptive adjusting global coefficient, searching efficiency and searching accuracy, it's a efficient optimization.

#### 4 Classification of BP neural network faults

BP optimization itself uses error squares as objective function, which uses gradient to get its mitigation. Unless error squares function is positive definite, there must be local minimum. It is a tortuous line in the process when iterative of gradient descent algorithm approaches minimum, which is called as "saw teeth" [3]. As a result it will make the convergence speed slow. When the neural network is used, first you should define the location of the swarm group  $x$  element as the every level of the neural network connects the weight value and threshold. Fitness function is the neural network mean square error indices, the formula is followed:

$$J = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C (y_{ji}^d - y_{ji})^2 \quad (12)$$

In it,  $N$  is the sample;  $C$  is the output of the neural network;  $y_{ji}^d$  is the idle output of sample I of network j;

$y_{ji}$  is the practical of sample I of network  $j$ . When using the PSO to train the BP network,  $x$  is the network weight value and threshold training BP.

Swam algorithm doesn't use gradient and other supplementary information during the process of network

study; it only handles neural network weight coefficient and threshold particle according to the degree of adaptation and get best answer.

The main parameter of BP neural network is study efficiency and momentum factor. Study efficiency is the speed of network, momentum factor can control the occurrence of swing.

$$\Delta w_{ji}^{(l+1)} = \eta \delta_j o_j + \alpha \Delta w_{ji}^{(l)} \quad (13)$$

$\alpha$  is momentum factor,  $\eta$  is studying efficiency, the course is:

1) Choose swarms  $m$ , adaptive threshold, maximum allowable iterations,  $t_{max}$ ,  $c_1$ ,  $c_2$ , put  $X$  as a random in  $(0,1)$ .

2) for  $(t=0; J_g > \epsilon \& t < t_{max}; t++)$

3) for  $i = 1: 1: m$

4) According to the training sample, figure out  $J_i$

6) if  $J_i < J_p(i)$ ,  $J_p(i) = J_i$ ;  $p_i = x_i$ ; end if

7) if  $J_i < J_g$ ,  $J_g = J_i$ ;  $p_g = x_i$ ; end if

8) end for

9) figure out  $z$  and  $p_t$

In it  $X = (x_1, x_2, \dots, x_m)$ ,  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$  is the location of the swarm  $I$ ;  $p = (p_1, p_2, \dots, p_m)$ ;  $J_g$  is the best adaptive answer in the present swarm group, the swarm is  $p_g$ .

## 5 Examples of diagnosis

This paper takes the malfunctions of steam generator for example. Measure the vibration on each axletree, and the vibration spectra of steam generator can be indicated in fig. 2. This paper takes eight common malfunctions such as quality disparity, rotor thermal bow and rubbing and impacting fault in the axial as the network output. Typical samples are listed in form 1 and fault samples as form 2. In the object output of samples such as form3, 1 designates that relevant malfunctions occur and 0 designates that malfunctions do not occur. Taking the advantage of learning samples, train the NN by means of BP and PSO-BP.

Input the malfunction mode to be discerned indicated in fig. 2 as a network into the network that is well trained, and a result indicated in form 3 will be achieved.

From Tab.3, as to the BP model, after introducing the particle swarm optimization, the accuracy is surpassing the original optimization and the situation of network's falling into the local minimum won't happen easily so that the accuracy has improved.

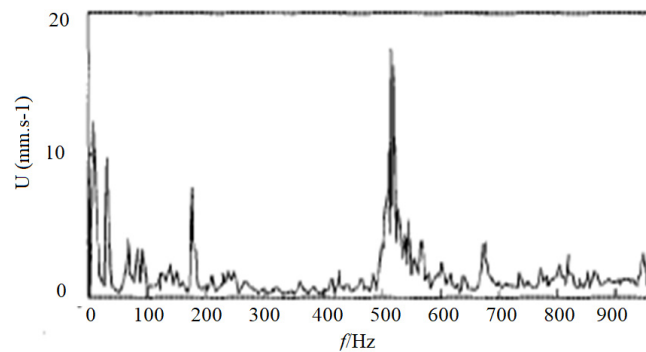


Fig. 2 Turbine machinery vibration speed spectra

Tab.1 Standard mode eigenvector of the common faults occurring in the rotating machinery

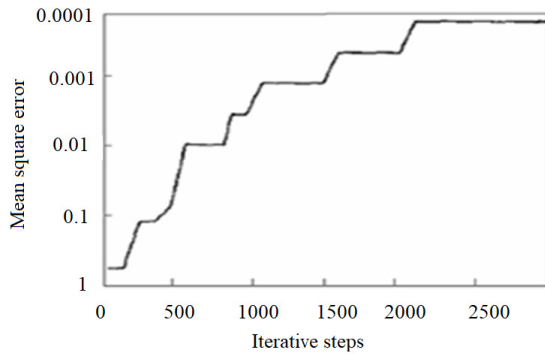
malfunction samples	1	2	3	4	5	6	7	output
vibration frequency	<0.4f	0.4/0.49f	0.5f	0.51/0.99f	1f	2f	3/5f	
disparity	0.00	0.00	0.00	0.00	0.89	0.04	0.05	10000000
misalignment	0.00	0.00	0.00	0.00	0.39	0.52	0.11	01000000
coupling damage	0.10	0.20	0.00	0.10	0.20	0.30	0.10	00100000
oil-film oscillation	0.00	1.00	0.00	0.00	0.00	0.00	0.00	00010000
axial fault	0.10	0.10	0.00	0.10	0.20	0.10	0.20	00001000
steam vibrating	0.00	0.20	0.10	0.70	0.00	0.00	0.00	00000100
loosing signals	0.60	0.40	0.00	0.00	0.00	0.00	0.10	00000010
basic resonance	0.00	0.00	1.00	0.00	0.00	0.00	0.00	00000001

Tab. 2 Eigenvector of malfunctions to be discerned input node

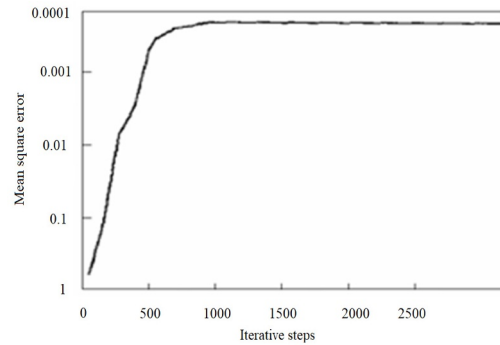
input node	1	2	3	4	5	6	7
input data	0.00	0.00	0.00	0.00	0.45	0.54	<b>0.07</b>
	0.11	0.10	0.00	0.10	0.20	0.09	<b>0.22</b>
	0.41	0.36	0.02	0.00	0.00	0.00	<b>0.03</b>
	0.00	0.96	0.00	0.17	0.01	0.00	<b>0.00</b>

**Tab. 3** Network diagnosis result malfunction 7

output node	1	2	3	4	5	6	7	8	malfunction
	0.0000	0.9467	0.0000	0.0000	0.0000	0.0001	0.0001	0.0011	2
output result	0.0006	0.0005	0.0713	0.0001	0.8714	0.0001	0.0000	0.0064	5
	0.3049	0.1856	0.0375	0.1097	0.2009	0.0135	0.7000	0.1200	7
	0.0419	0.0561	0.1059	0.8639	0.0357	0.2344	0.0312	0.2613	4



**Fig.3** BP neural network output



**Fig.4** PSO-BP neural network output

Through the use of the learning samples, BP optimization and the mixed PSO-BP optimization train the neural train network. The relationship between BP optimized by PSO and the typical BP neural network output is as picture 3 and picture 4: Seen from picture 3 and 4 we can see that to reach the same faults accuracy, the BP network and the PSO-BP neural network need distinctive iterations. So compared to BP neural network, PSO-BP neural network have a higher speed on the whole.

## 6 Conclusions

The theoretical analysis and the examples show that PSO-BP neural network faults diagnose is a better diagnose. It can not only improve BP neural network's weight value but also improve the structure of the neural network, overcoming the traditional BP neural network faults diagnoses' disadvantages and improving the speed of the neural network training. What's more, it's good for the practical use of the BP neural network faults diagnose.

## Acknowledgments

This work is supported by Shandong Electric Power Corporation.

## References

1. Zhao Bo, Cao Yija, "An improved paricle swarm optimization algorithm for power system unit commitment". *Power System Technology*, 2004, 28(21):6-11
2. Wei Yu, Liu Zhenxing, "Nonlinear blind source separation using improved particle swarm optimization." *Aerospace Electronics Information Engineering and Control* 2006,28(1):139-143

3. Bulter K L, Sarma N D R. "General Reconfiguration methodology for AC radial shipboard power systems." *IEEE 2000 Power Engineering Society Winter Meeting*, 2000, 2,1226-1230.
4. Wei Xing, Shu Naiqiu, Cui Pengcheng, "Based on improving PSO-BP neural network and D-S faults diagnose." *Automation of electricity system*, 2006, 30(7):46~50
5. Pan Hao, Hou Qinglan, "Based on BP network study." *Computer project and apply* 2006,16:41-45
6. NatsukiH, Hitoshi. "Particle Swarm optimization with Gaussian mutation." *Proc of the Congresson Evolutionary Computation*, 2003:72-7