Turbo-Generator Vibration Fault Diagnosis Based on PSO-BP Neural Networks

YOUPENG ZHANG, HONGSHENG SU*
School of Automation and Electrical Engineering
Lanzhou Jiaotong University
Lanzhou 730070
P.R.CHINA
*shsen@163.com

Abstract: - To overcome the flaws of traditional BP learning algorithm of its low convergence speed and easy falling into local extremum during turbo-generator vibration faults diagnosis, a novel algorithm called PSO-BP is proposed for artificial neural network (ANN) learning based on particle swarm optimization (PSO) in this paper. The algorithm covers the two phases. Firstly, PSO algorithm is applied to optimize the weight values of neural networks based on training samples. BP algorithm is then applied for farther optimization based on verifying samples till the best weight values are achieved. In addition, to simplify the structure of the networks, rough set theory (RST) is applied to implement the attributes reduction from the diagnostic decision table quantified by Kohonen networks. Thus the learning speed and the misjudgment rate of the diagnostic networks are improved, dramatically. Finally, the trials in turbo-generator vibration faults diagnosis indicate that the proposed method possesses better forecasting accuracy and lower misjudgment rate, and is an ideal pattern classifier.

Key-Words: - Turbo-Generator; Particle Swarm Optimization (PSO); BP neural networks; Rough set; Kohonen networks; Evaluation

1 Introduction

With the development of national economy, single turbo-generator capacity is ever-growing, and the reliable, safe and efficient operation of the generators is required more, consequently. It is very significant to research on fault diagnosis and supervision technology of the turbo-generator. The vibration status of the generators is an important technology index for its continual operation. For many years, the instances are not unusual that the damage of the generators is made due to vibration abnormity. Clearly, the reinforcement and abnormity of the vibration is often the boding of the generators faults[1]. Traditionally, the diagnostic methods for the turbo-generator vibration faults mainly include neural networks(NN)[2-3], genetic algorithm(GA)[4-5], expert systems(ES)[6-7], fuzzy set(FS)[8-9], rough set(RS) [10], and some other new methods[11-15]. However, all these methods are conditionally applied and the better results are not achieved. To change the situation and improve the diagnostic accuracy, a PSO-BP algorithm based neural network is proposed in this paper. The PSO-BP algorithm overcomes the flaws of traditional BP gradient calculation of low convergence speed and falling into local extremum, and seeks the best weight values through particle swarm iteration, the accuracy of the algorithm is therefore improved, dramatically. In addition, to simplify the structure of the networks, rough set theory (RST) is applied to implement the attributes reduction from the diagnostic decision table quantidied by Kohonen networks. The learning and diagnosis speed of the neural networks is therefore improved, greatly. Investigations show that new PSO-BP algorithm can quicken the convergence procedure and find the global extremum, rapidly, and is an ideal classification algorithm.

2 Rough Set Theory

The attributes set which is composed of the conditional attribute C and the decision attribute D, $A=C\cup D$, $C\cap D=\varnothing$, $a\in A$, $V=Va$, Va is scope of In RST, knowledge denotation system may be described by.

$$S=<U,A,V,F> \quad (1)$$

where U is the universe and expresses a set of the finite objects, A is ta, f: $U\times A\rightarrow V$ is a information

function, it specifies attribute values of every object in U.

Information systems based on rough sets definition can be denoted by the use of table format, where columns express attributes and rows represent objects, and every row describes information of an object. The table therefore is called decision table, which can generalize the relationships among data and educe the classification rules of the concepts. In rough sets, binary indivisible relationship ind(R) determined by R⊆A can be expressed by

$$\text{ind}(R)=\{(x,y) \in U\times U| \ \forall a\in A, f(x,a)=f(y,a)\} \qquad (2)$$

It is very clearly that if $(x,y) \in \text{ind}(R)$, then x and y can't be differentiated in accordance with existing information, they are an equivalent relation in U.
Let $S=<U, C\cup D>$, if $C1\in C$, $C1\neq\varnothing$, and the following two conditions hold.

(1) $\text{ind}_{C1}(D)=\text{ind}_{C}(D)$

(2) $\forall C2\subseteq C1$, $\text{ind}_{C2}(D)\neq\text{ind}_{C1}(D)$

According to (1) and (2), we can say $C1$ is a reduction of C with regard to D, the intersection of all these reductions is called core, and defined as $\text{core}_D(C)= \cap\text{red}_D(C)$.

Through the above reduction we can get several reduction attribute sets. The best attributes combination is considered to possess the smallest average intervolving information, whose basic steps are described as follows.

1) Firstly, the intervolving information between all the two in each attribute combination is worked out, and the results are then totaled and averaged. The acquired average quantity is considered as the average intervolving information of the reduction attribute combination.

2) Secondly, the average intervolving information of all reduction attribute combinations are worked out, afterwards, the attribute combination possessing the smallest average intervolving information may be selected as the best reduction attribute set. The smallest means the appropriately inter-independent attribute combination. In [16], the intervolving information based on information entropy is defined as follows.

Definition1. The information entropy of the equivalent relation $G(U|I(G)=\{x_1, x_2,\ldots, x_n\})$ can be expressed by $H(G)$, the conditional entropy that the equivalent relation $Q(U|I(Q)=\{y_1, y_2,\ldots, y_n\})$ is relative to G can be expressed by $H(Q|G)$, then the involving information between G and Q is defined by

$$I(Q,G)=H(Q)-H(Q\mid G)=\sum_{x,y}p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)} \qquad (3)$$

where p expresses the probability.

3 Kohonen Network Quantification Method

The diagnostic decision table possibly comprises some redundant characteristics. Therefore, how to identify these redundant characteristics and ignore them and not influence the origin content of the diagnosis will be a very significant job. RS is considered as a powerful tool for data mining and knowledge discovery, and can be therefore used to simplify the decision table. Considering the quantification characteristic of RS, the continuous attribute values must be quantified in the decision table. Due to the excellent characteristics, say, simple structure, short learning time, high classification precision, and strong anti-interference ability, Kohonen networks are therefore applied to quantify the decision table, broadly. The algorithm is described as follows [17].

Step 1: Let the largest classification number of Kohonen network be I, each attribute value set V_i is classified as two classes from the start, the classification boundary point $P(j)$ is then achieved.

1) The initial weigh value $\boldsymbol{W}(0)$ is given a small stochastic quantity;

2) The distance d_j between input $\boldsymbol{X}_i(t)$ and each output node j can be then worked out;

3) The connection weighs of the minimal distance node is then adjusted.

Step 2: When classification boundary point is $P(j)$.

1) According to $P(j)$, region is partitioned, and each region is partitioned as a class, the variance $D1$ inside one class and $D2$ between the classes are worked out, respectively;

2) Let classification number be K ($I\geq K\geq 2$), when $K\geq I$, turn to step 1; when $K<I$, the classification is terminated. Hence, K-1 is the largest classification number of this attribute set;

3) For each attribute set V_i, K is selected as the best classification number only since it can let $D1$ up to the minimum and $D2$ up to the maximum.

Step 3:

1) According to the best classification of each V_i, the best classification boundary points may be gained;

2) Region partition is done and marked.

3) The quantified decision table is acquired.

4 PSO-BP Learning Algorithm

4.1 BP algorithm

BP algorithm belongs to δ algorithm, and is a supervised learning algorithm [18]. Its main ideal is to apply the errors between the practical outputs of the network and aim vector to modify its weight values so as to let the mean square of the output neurons up to minimum. During the adjustment every time, the varieties of the weight values and deviations has a direct proportion to its error, and the error influence is backward propagated to former each layer.

BP learning algorithm comprises the two parts, that is, forward propagation for information and backward propagation for error. During forward propagation, the input information is propagated to the output layer after calculation each layer, the neuron states each layer only influence the ones in its later layer. If the desired values in output layer haven't been acquired, the changes of the error then are worked out, afterwards, the algorithm shifts to backward propagation process. Across the backward propagation, the error signals in output layer is backward propagated to former each layer to adjust the weight values of the neurons till the expected aim is achieved. For convenience analysis, we take a two-layer neural network for example to explain, its simplified network structure is shown in Fig.1.

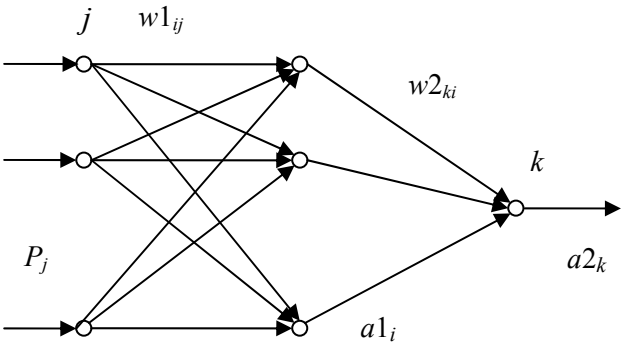

Fig.1 Simple network structure with one hidden layer

Let input vector be $\boldsymbol{P}$, and the input layer have r neurons, then $\boldsymbol{P}=(p_1,p_2,\ldots,p_r)$. Let the hidden layer have $s1$ neurons, its activation function be $f1$, and $s2$ neurons in output layer, and its activation function be $f2$. Let the actual output of the networks be $\boldsymbol{A}$, and $\boldsymbol{T}$ be the aim vector. The detailed learning process is described below.

4.1.1 Forward Propagation of Information

According to Fig.1, the output of the i^{th} neuron in hidden layer is calculated by

$$a1_i = f1(\sum_{j=1}^{r} w1_{ij}\, p_j + b1_i), \quad i=1, 2,\ldots,s1. \tag{4}$$

The output of the k^{th} neuron in output layer then is described by

$$a2_k = f2(\sum_{i=1}^{s1} w2_{ki}\, a1_i + b2_k), \quad k=1, 2,\ldots,s2. \tag{5}$$

The defined error function is expressed by

$$E(\boldsymbol{W}, B) = \frac{1}{2}\sum_{k=1}^{s2}(t_k - a2_k)^2 \tag{6}$$

4.1.2 Backward Propagation of Error

1) The Weight Adjustment in Output Layer

Considering the weight alteration from the i^{th} neuron in hidden layer to the k^{th} output in output layer, then

$$\begin{aligned}\Delta w2_{ki} &= -\eta\frac{\partial E}{\partial w2_{ki}} = -\eta\frac{\partial E}{\partial a2_k}\bullet\frac{\partial a2_k}{\partial w2_{ki}} \\ &= \boldsymbol{\eta}(t_k - a2_k)\bullet f2'\bullet a1_i = \eta\bullet\delta_{ki}\bullet a1_i\end{aligned} \tag{7}$$

where

$$\delta_{kj} = (t_k - a2_k)\bullet f2 = e_k\bullet f2' \tag{8}$$

$$e_k = t_k - a2_k \tag{9}$$

Similarly, we have

$$\begin{aligned}\Delta b2_{ki} &= -\eta\frac{\partial E}{\partial w2_{ki}} = -\eta\frac{\partial E}{\partial a2_k}\bullet\frac{\partial a2_k}{\partial b2_{ki}} \\ &= \eta(t_k - a2_k)\bullet f2'_i = \eta\bullet\delta_{ki}\end{aligned} \tag{10}$$

2) The Weight Adjustment in Hidden Layer

Considering the weight alteration from the j^{th} input in input layer to the i^{th} output in hidden layer, we then have

$$\begin{aligned}\Delta w1_{ij} &= -\eta\frac{\partial E}{\partial w1_{ij}} = -\eta\frac{\partial E}{\partial a2_k}\bullet\frac{\partial a2_k}{\partial a1_i}\bullet\frac{\partial a1_i}{\partial w1_{ij}} \\ &= \eta\sum_{k=1}^{s2}(t_k - a2_k)\bullet f2'\bullet w2_{ki}\bullet f1'\bullet p_j = \eta\bullet\delta_{ij}\bullet p_j\end{aligned} \tag{11}$$

$$\delta_{ij} = e_i\bullet f1', e_i = \sum_{k=1}^{s2}\delta_{ki}w2_{ki}, \delta_{ki} = e_k\bullet f2', e_k = t_k - a2_k \tag{12}$$

Similarly, we have

$$\Delta b1_i = \eta \delta_{ij} \quad (13)$$

According to (**7**),(**10**),(**11**) and (**13**), we can see that the alteration in weight values is direct proportional to its error e_k, that is, if the larger the absolution value of e_k is, the larger its influence on related weight and threshold values is. The above descriptions are basic BP algorithm. Due to application of gradient algorithm basic BP algorithm possesses many flaws, for instance, easy falling into local optimization. To tackle this problem many improved BP algorithm is recommended, such as self-adaptive learning rate, accessional momentum law, and so on. However, these improvements do not resolve this problem, thoroughly. New global optimization algorithms still expect to be found.

4.2 PSO Algorithm

Particle Swarm Optimization (PSO) is a global random optimization algorithm [19]. Its basic thinking comes from intelligent behavior of the swarm. Specifically speaking, it mainly emulates the characteristics of the migration and gathering of the birds during seeking food. The algorithm generates swarm intelligence to optimize the seeking aim by the cooperation and competition among the particles. PSO algorithm not only retains global scout strategy with swarm-based, the applied operation model called as displacement-speed is comparatively simple and programming is easily realized, but also it holds the unique optimization properties such as fast operation speed and relatively simple structure. PSO is a high efficient parallel seeking algorithm, and has some prominent behaviors in tackling non-linear optimization problems. A basic PSO algorithm is presented as follows.

Set a swarm composed of n particles in D-dimensional space, the i^{th} particle may be expressed as a D-dimensional vector $\boldsymbol{x}_i=(x_{i1},x_{i2},\ldots,x_{iD})$, $i=1,2,\ldots,n$, namely, the position of the i^{th} particle in D-dimensional space is $\boldsymbol{x}_i$, and each such position is named as a potential solution. The adaptability function value of $\boldsymbol{x}_i$ is calculated by substituting it into the aim function $f(\boldsymbol{x}_i)$, then, according to the value size, $\boldsymbol{x}_i$ can be weighed to be the good or the bad. The flight speed of the i^{th} particle is also a D-dimensional vector, and written as and $\boldsymbol{v}_i=(v_{i1},v_{i2},\ldots,v_{iD})$. Set until now, the optimal position sought by the i^{th} particle is $\boldsymbol{p}_i=(p_{i1},p_{i2},\ldots,p_{iD})$, and the optimal position sought by the overall particle swarm is $\boldsymbol{p}_g=(p_{g1},p_{g2},\ldots,p_{gD})$. Then the position and speed of the particle i can be evolved according to the following equation (**14**) and (**15**).

$$v_{id}^{k+1} = \omega \times v_{id}^{k} + c_1 r_1 (p_{id}^{k} - x_{id}^{k}) + c_2 r_2 (p_{gd}^{k} - x_{id}^{k})] \quad (14)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \quad (15)$$

In (**14**), ω is inertia weight, and indicates an influence yielded by the present speed of the particle on next generation. A suitable ω can make the particle hold balanced exploration abilities. Parameters c_1 and c_2 are non-negative learning factors, the values of which usually are limited in range of one to two, if the values are too small, the particle is far away from the optimal aim area, inversely, if too large, the particle can suddenly or possibly fly over aim area. $r_1$and r_2 are random variables with a scope of zero to one.

The prosperous key of PSO algorithm is the selection and adjustment of its parameters, including collective size P, particle dimension d, learning factors c_1 and c_2, inertia weight ω, the largest speed $v_{\max}$, the largest iteration times $T_{\max}$, and calculating precision ε, where $T_{\max}$ and ε is the termination conditions, and confirmed by optimization quality and seeking efficiency. In general, P is in range of 10 to 40; d is dimension of solution space determined by concrete problems; $v_{\max}$ is the largest flying speed of the particles, if v_{max} is too large, the particles likely fly over better solution, and easily fall into a local optimization conversely. Clearly, it directly influences the global exploration ability. To make certain the influence of ω on the algorithm, through analyzing a quantity of experiment data Eberhart and Shi give out the conclusions below: if $v_{\max}\leq 2$, then ω tends to 1; if $v_{\max}\geq 3$, then ω tends to 0.8 better; when $\omega\in(0.9, 1.2)$, the ideal results may be attained. The inertia weight ω is usually considered as ≤ 1.4 so as to make particles hold locomotion inertia and possess capabilities to break a new exploration space, or mounted as linearly decreasing with iterative time called LDW strategy [20], that is,

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{G} \times g \quad (16)$$

where G expresses the gross iterative time, g represents the current iterative time. Learning factors c_1 and c_2 represent statistic speeding weight of each particle towards P_{best} and g_{best} localization. Compatible c_1 and c_2 can speed convergence and may

not fall into local minimum. $c_1+ c_2$ may as well be approximate to 4, generally, $c_1= c_2\approx 2.05$.

4.3 PSO-BP Algorithm

PSO is applied to neural networks for the two functions: one is the optimization of the network weight; the other is the optimization of the network structure. Below we mostly consider the optimization of the network weight.

Typical BP algorithm easily falls into local minimum, this leads to a low convergence speed. In addition, the network learning is very sensitive to initial weight values, whose slightly changes will cause the oscillation of the network. It requires constant training for these parameters to be fixed, but excess training leads to over-fitting. Due to quick convergence speed and better global exploration capability, PSO algorithm is applied to optimize the weights of the neural network so as to overcome the flaws of BP algorithm. Thus, the generalized ability of neural network not only can be expanded, but also its learning ability and convergence speed are improved, dramatically.

The two key points must be seized when PSO is applied to optimize the weights of the networks.

1) To establish a mapping connection between the particle dimensions and the network weights. The dimension weight of each particle in particle swarm is corresponding to a connecting weight in NN, in other words, the number of the weights in NN should be the same as the one of the particles PSO.

2) To select MSE of NN as the fitting function of PSO algorithm.

Let d neurons be in input layer, m in hidden layer, and n in output layer, neural networks therefore possess $d\times m+m\times n+m+n$ weights and thresholds in all. Correspondingly, the dimensions of each particle of PSO algorithm should also be $d\times m+m\times n+m+n$. Let the network possess N training samples, mean square error (MSE) is then expressed by

$$MSE_T=\frac{1}{N}\sum_{i=1}^{N}\ [\sum_{j=1}^{n}\ (t_{ij}-y_{ij})^2] \qquad (17)$$

The above equation may serve as the fitness function in PSO algorithm, where t_{ij} is the desired export and y_{ij} is the practical output of the network. In PSO-BP algorithm, all weights and thresholds are firstly coded as a real number vector to express individual in colony. Stochastically generating the colony of these vectors, newly generated individual reverting as the weights of the network during evolution, MSE serving as fitness function, thus learning is changed as optimizing problem, that is, to seek a group of the optimal weights to make MSE minimum. If MSE is lower than the given precision beforehand or iterative time is lower than the largest time, training process then stops, iteration continues implementation till the largest iterative time is arrived otherwise. At the moment, the achieved parameters is quite approximate to the best combination, on the basis of it, BP algorithm is used to optimize the parameters of the network further till the best parameters are obtained, that is, the misjudgment rate of testing sample group is minimum. The flow chart of PSO-BP algorithm is shown in Fig.2.

5 Performance Evaluation of Learning Algorithm

For feed-forward neural networks, its performance is evaluated by classification or function approximation problems. Generally, there are four kinds of indexes to be adopted for learning algorithm test.

1) Classification error rate in training set, expressed by

$$\varepsilon_T= \frac{n_T}{N_T}\times 100\% \qquad (18)$$

where n_T is the misjudged samples number, and N_T is the gross samples number in training set.

2) Classification error rate in test set, expressed by

$$\varepsilon_G= \frac{n_G}{N_G}\times 100\% \qquad (19)$$

where n_G is the misjudged samples number, and N_G is the gross samples number in training set.

3) MSE_T in training set, expressed by

$$MSE_T= \frac{1}{2N_T}\sum_{i=1}^{N_T}\sum_{j=1}^{C}\ (y^d_{j,i}-y_{j,i})^2 \qquad (20)$$

4) MSE_G in test set, expressed by

$$MSE_G= \frac{1}{2N_G}\sum_{i=1}^{N_G}\sum_{j=1}^{C}\ (y^d_{j,i}-y_{j,i})^2 \qquad (21)$$

For classification, the above four indexes are adopted. For function approximation, the only MSE_T and MSE_G may be applied for the latter.

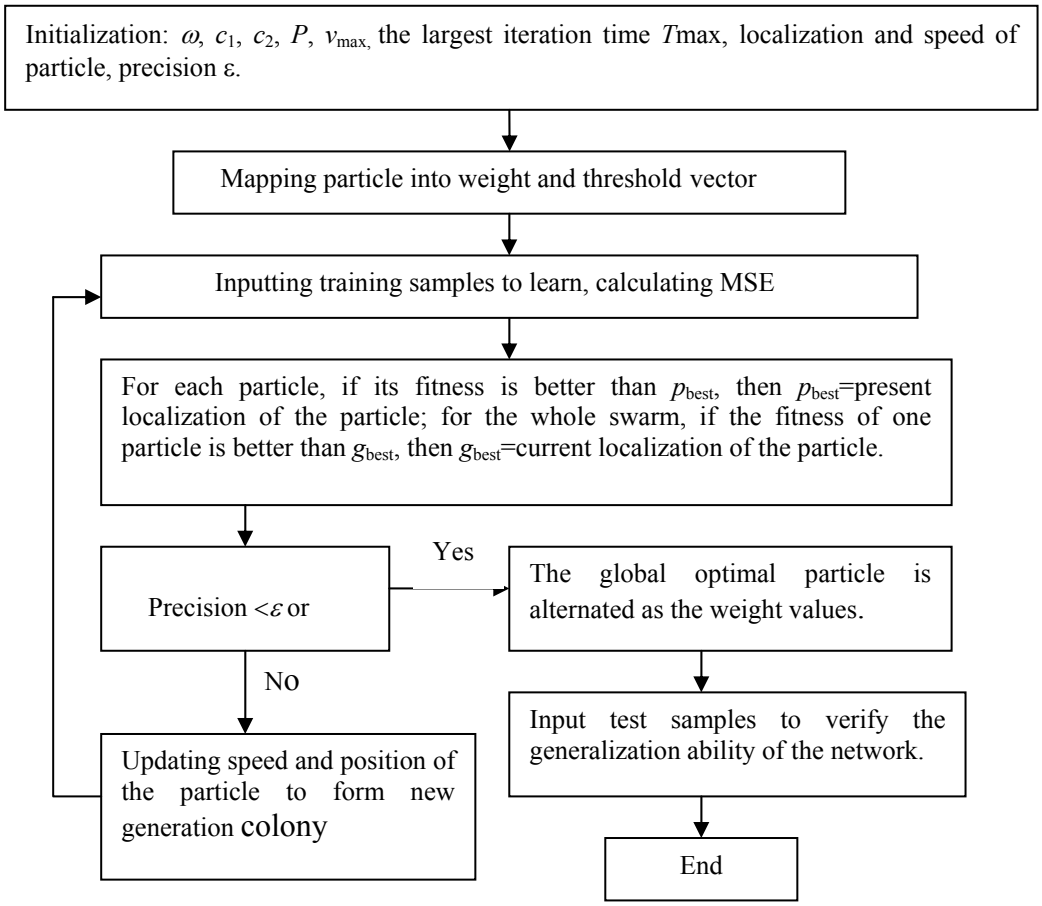

Fig.2 The flow chart of PSO-BP algorithm

6 Examples

According to practice and practical cases in resent years, some familiar fault sources and fault symptoms information of stream turbine vibration faults are listed in Tab.1 [2]. Ten kinds of typical faults in rotation machines are selected as outputs of neural networks, and the score peak energy values in the range of nine frequency ranges are used to act as input characteristic vector in Tab.1. It is necessary to say that frequency score energy values are obtained by three-time wavelet package decomposition to faults wave-shape in the range of given frequency and be probabilistically completed.

Tab.1 Fault sources and symptoms table of stream turbine

Sample (*D*)	*a* (0.01-0.39)f	*b* (0.40-0.49)f	*c* 0.50f	*d* (0.51-0.99)f	*e* f	*f* 2f	*g* (3/5)f	*h* odd-time f	*i* high frequency f
1	0.00	0.00	0.00	0.00	0.90	0.05	0.05	0.00	0.00
2	0.00	0.30	0.10	0.60	0.00	0.00	0.00	0.00	0.10
3	0.00	0.00	0.00	0.00	0.40	0.50	0.10	0.00	0.00
4	0.10	0.80	0.00	0.10	0.00	0.00	0.00	0.00	0.00
5	0.10	0.10	0.10	0.10	0.20	0.10	0.10	0.10	0.10
6	0.00	0.00	0.00	0.00	0.20	0.15	0.40	0.00	0.25
7	0.00	0.00	0.10	0.90	0.00	0.00	0.00	0.00	0.00
8	0.00	0.30	0.10	0.60	0.00	0.00	0.00	0.00	0.00
9	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.00
10	0.00	0.00	0.00	0.00	0.00	0.80	0.20	0.00	0.00

f: 50HZ, D1: unbalance, D2: vapour impulsion force even, D3: deflection centre, D4: oil film whorling, D5: rotator collision, D6: symbiosis looseness fault, D7: push force bearing fault, D8: gasping vibration, D9: bearing seat looseness fault, D10:unequal bearing rigidity.

In Tab.1, let input characteristic vector be expresses by $\boldsymbol{X}$={*a*, *b*, *c*, *d*, *e*, *f*, *g*, *h*, *i* }, and output vector be $\boldsymbol{Y}$={*D*}. According to quantitative method of Kohonen network, a quantitative decision table is obtained as follows.

Tab. 2. Quantitative decision table

Sample (*D*)	*a* (0.01-0.39)f	*b* (0.40-0.49)f	*c* 0.50f	*d* (0.51-0.99)f	*e* f	*f* 2f	*g* (3/5)f	*h* odd-time f	*i* high fre. f
1	0	0	0	0	1	0	0	0	0
2	0	1	1	1	0	0	0	0	1
3	0	0	0	0	1	1	0	0	0
4	1	1	0	1	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1
6	0	0	0	0	1	1	1	0	1
7	0	0	1	1	0	0	0	0	0
8	0	1	1	1	0	0	0	0	0
9	1	0	0	0	0	0	0	1	0
10	0	0	0	0	0	1	1	0	0

Tab. 3. Simplified decision table

D	*a*	*b*	*e*	*f*	*i*
1	0	0	1	0	0
2	0	1	0	0	1
3	0	0	1	1	0
4	1	1	0	0	0
5	1	1	1	1	1
6	0	0	1	1	1
7	0	0	0	0	0
8	0	1	0	0	0
9	1	0	0	0	0
10	0	0	0	1	0

Known after rough set reduction, {*b*,*e*,*i*} is core attribute set, we therefore can get reduction attribute sets {*a*,*b*,*e*,*f*,*i*},{*a*,*b*,*e*,*g*,*i*}, {*b*,*c*,*e*,*f*,,*i*}, and {*b*,*c*,*e*,*g*,*i*}. According to (5), the attribute set *X*'={*a*,*b*,*e*,*f*,*i*} possesses the smallest average intervolving information, it is for that selected as conditional attributes set and *D* serves for decision attributes set, a simplified decision table is gained as shown Tab. 3.

Seen from the Tab. 3, the original 9 attributes now become 5, the redundant conditional attributes are ignored, and the decision table is simplified greatly, the structure of neural network is therefore simplified If the condition attributes serve as the inputs of BP network and the decision attribute act as the outputs of the network, and the samples are used to train the network in Tab. 3, the trained network may then be applied to implement fault diagnosis. Presently, to select the suitable structure of neural network still hasn't theoretical instructions, according to prior experience, the sum of the nodes in hidden layer shouldn't be smaller than the sum of the nodes in input layer. For convenient analysis, here the structure of the neural networks is selected as 5-6-10. Thus, there are 5×6 +6×10 +6+10=106 weights and thresholds to require to be optimized in all, and so, the dimension of the particle in PSO algorithm should also be 106. Let the gross number of the particles be 30, learning factor $c_1$and c_2 be 2.05, the initial location of the particles be given according to 2**rand*()－1, and the initial location of the particles be given according to *rand*()*×0.02－0.01. Let ω linearly decrease with iteration time in range of 0.45 to 0.95, and *V*max=0.5. The transfer function of the neurons in middle layer selects S type tangent

function *tansig*, and the one in output layer is S type logarithm function *logsig*. The reason to do this is that outputs of the functions is within the scope of 0 to 1, output requirements of the networks can just be well satisfied. Learning function is *trainlm* function. Let learning times T_{max} of the networks be 1000, training aim ε be 0.001, and learning rate equal 1. Thus we get the learning process of the three networks below.

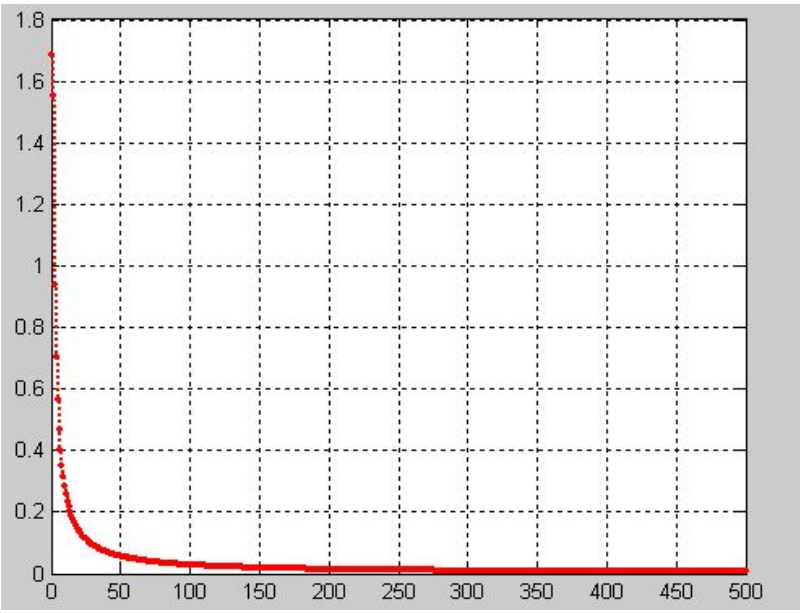

Fig.3 Error graph of BP neural network with PSO based

Seen from Fig.3, after 500-time training the error requirement is met. When inputting the training in Tab.3, the practical output of PSO-BP algorithm is

Tab. 4 The true output of the training samples

1	2	3	4	5	6	7	8	9	10
1.0126	-0.0087	-0.0087	-0.0060	-0.0063	0.0043	0.0008	0.0024	0.0040	0.0009
0.0010	0.9713	0.0053	-0.0123	-0.0092	0.0080	-0.0173	0.0052	0.0007	-0.0032
0.0047	-0.0035	1.0189	-0.0060	-0.0058	-0.0094	-0.0096	0.0092	0.0039	-0.0033
0.0004	0.0042	0.0067	1.0094	-0.0065	0.0016	-0.0187	0.0059	0.0024	0.0012
0.0002	0.0065	0.0020	0.0058	0.9801	0.0088	-0.0095	0.0092	0.0002	0.0033
0.0047	-0.0032	0.0008	-0.0057	-0.0152	1.0055	-0.0416	0.0097	0.0057	-0.0044
0.0020	-0.0105	0.0081	-0.0097	-0.0337	0.0120	0.9127	0.0098	0.0008	-0.0087
-0.0055	0.0026	-0.0089	0.0090	0.0044	-0.0188	0.0665	0.9558	-0.0026	0.0091
-0.0015	-0.0037	-0.0061	-0.0034	0.0055	-0.0096	0.0028	-0.0052	0.9969	-0.0004
0.0070	-0.0020	0.0018	-0.0003	-0.0028	0.0037	-0.0190	0.0004	0.0021	0.9979

Known from Tab.4, the classification correctness rate of PSO-BP neural network is 100% for the training samples group. The faults nodes are close to 1, and non-faults nodes are close to zero, the gross training results are for that feasible.

Tab.5 gives out ten test samples to test the well-trained PSO-BP neural network, and Tab.6 is the export results. It is noted that Tab.6 is the average of 20-time trials.

Tab. 5 The test samples group

Fault Type	0	1	2	3	4	5	6	7	8
Fre. Region	0.01/0.39 f	0.40/0.49 f	0.5f	0.51/0.99 f	f	2f	3-5f	Odd-time f	high fre. f
D1	0.00256	0.00122	0.00993	0.01826	0.81123	0.07904	0.04958	0.04958	0.0039
	0.05129	0.00267	0.00227	0.01846	0.7578	0.09388	0.03373	0.03373	0.0059
	0.00493	0.00162	0.00131	0.01049	0.84174	0.05299	0.01962	0.01962	0.0032
D5	0.11805	0.01598	0.00831	0.12527	0.56643	0.01725	0.04653	0.02356	0.0786
	0.11670	0.00545	0.00523	0.17401	0.56365	0.02107	0.05358	0.01288	0.0534
	0.03012	0.01275	0.02175	0.16904	0.61279	0.01977	0.05657	0.02518	0.0520
D0	0.02475	0.18273	0.39201	0.19642	0.05736	0.09657	0.02254	0.02254	0.0051
	0.00482	0.24000	0.5058	0.07214	0.08549	0.03526	0.02535	0.02535	0.0059
	0.01321	0.23394	0.48800	0.06358	0.09938	0.03841	0.02777	0.02777	0.0079
	0.02363	0.14473	0.5394	0.10211	0.05216	0.09117	0.02069	0.02069	0.0054

f: 50HZ, D0: normal, D1: unbalance, D5: rotator collision,

Tab. 6 The test samples group

Fault Type	1	2	3	4	5	6	7	8	9	10
D1	0.8685	-0.0118	0.1985	-0.0970	0.0926	-0.0244	-0.0610	0.0941	-0.0057	-0.0241
	0.8512	0.0064	0.0970	-0.0348	0.0295	-0.0893	0.0115	-0.0474	0.0258	-0.1936
	0.8716	-0.0147	0.1260	0.0208	0.1340	-0.0084	0.0225	-0.0373	-0.0371	-0.0607
D5	0.0894	0.0064	-0.1519	0.3352	0.8087	-0.0929	0.0421	-0.0854	-0.2894	0.2064
	0.0444	-0.2212	0.0255	0.0142	1.0241	-0.1500	0.1459	0.2516	-0.0444	-0.2212
	0.2866	-0.1034	0.1212	-0.0614	0.8579	-0.0308	0.2036	-0.0295	-0.2866	-0.1034
D0	-0.3275	0.2050	-0.0958	-0.0020	0.1520	-0.1850	-0.2020	0.0593	0.0275	0.2350
	0.0468	-0.1384	-1.1232	0.3640	0.3805	0.0957	-0.2875	0.0062	-0.0468	0.1384
	-0.0905	0.0902	-0.3105	-0.0263	0.0761	-0.0450	0.1129	-0.2167	0.1905	0.0902
	0.0561	-1.0210	-0.5212	0.3541	1.6196	-0.3754	0.0036	0.0167	-0.1276	0.0021

Known from Tab. 6, the misjudgment rate in single time is 0%.

Below we review the two performance indexes of PSO-BP: ε_T and ε_G. are calculated as shown in Table 7.

Tab. 7 ε_T and ε_G of PSO-BP algorithm

Algorithm	ε_T	ε_G
PSO-BP	0	0.15

Applying the same training and test samples group, the compared results are given as shown in Table 8 between PSO-BP neural networks and BP neural networks.

Tab. 8 Compared results between PSO-BP and BP neural networks

	BP	PSO-BP
MSE_{T}	0.0005064	0.0003085
MSE_{G}	0.1931	0.03113
ε_T	0	0
ε_G	0.23	0.15

Clearly, the compared results indicate that MSE_{T} and MSE_{G} of PSO-BP algorithm are lower than ones of BP algorithm, and for training samples and test samples, the misjudgment rates of PSO-BP algorithm are smaller than the ones of BP neural networks, and the training time is also improved, markedly.

7 Conclusion

PSO-BP algorithm effectively overcomes the flaws of conventional BP algorithm during neural networks learning, such as improving learning precision and reducing learning speed, and is an effective global learning method. Moreover, PSO-BP algorithm also overcomes the deficiencies of PSO algorithm such as slow convergence speed and low classification precision. In terms of PSO and BP, they have many improved models, how to match these models better and to make them work more effectively is our future work.

Acknowledgements

This work is partially supported by the 'Qing Lan' Talent Engineering Funds supported by Lanzhou Jiaotong University. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References:

[1] W. Li, *Examination Analysis and Prevention for Stream Turbine Generators Faults*, China Electrical Power Publication House, 2002.

[2] H. J. Yu, *Intelligent Diagnosis Based on Neural Networks*, Metallurgy Industry Publication House, 2000.

[3] J. Y. Yan, *Research on Turbine Vibration Faults Diagnosis Method Based on ANN*(D), Hehai University, 2006.

[4] Hongsheng Su, Qunzhan Li, Application study of composite neural networks in fault diagnosis systems based on GA learning, *Computer Science*, Vol.32, No.8A, 2005, pp.255-256.

[5] B. Zhang, M. Li, and G. Zhang, The Multipile-Layer Fuzzy Model for Tubro-Generator Faulty Diagnosis, *Turbine Technology*, Vol.45, No.5, 2003, pp. 25-28.

[6] X. Zhang, C. X. Miao, A Trouble Diagnosis Eexpert System for Stream-Turbine Based on Fuzzy Theory and Synthesis Analysis, *Coal Engineering*, No.12, 2002, pp.28-30.

[7] Z. Dong, N. Zhang, P. Han, and D. Zhang, Design of Turboset Vibration Fault Diagnosis Expert System of Power Plant and Its Realization, *Electric Power Science and Engineering,* Vol.23, No.2, 2007, pp.44-47

[8] X. Wang, N. Zhang, Z, Dong, Turbine Vibration Fault Diagnosis Based on Fuzzy Clustering Analysis, *Journal of Northchina Electric Powers University*, Vol.35, No.2, 2008, pp. 57-60.

[9] Z. Y. Wang, T. Y. Wang, Y. G. Xu, and Q. Li, Multipile Symptom Fuzzy Fault Diagnosis for Stream Turbine Vibration Based on CLIPS, *Electric Power*, Vol.41, No.4, 2008, pp.67-70.

[10] W. Huang, X. Zhao, W. Wang, and L. Dai, Rough Set Model for Vibration Fault Diagnosis of Stream Turbine-Generator Set, *Automation of Electric Power Systems*, Vol.28, No.15, 2004, pp.80-84+103.

[11] Z. Dong, N. Zhang, and P. Han. Stream Turbine Vibration Fault Diagnosis Using Rough Sets and Bayesian Network, *Turbine Technology*, Vol. 50, No.3, 2008, pp.238-240.

[12]P. Liang, X. Long, and G. Wu, A Study of Turbine Vibration Fault Diagnosis Based on ARMA and a Neural Network, *Journal of engineering for thermal energy and power*, Vol.22, No.1,2007, pp.6-11.

[13]C. Chen, Q. Liu, Applications of Probabilistic Causal Network in Fault Diagnosis Turbine Machinery, *Proceedings of the CSEE*, Vol.21, No.3, 2001, pp.78-81.

[14] W. Zhao, R. Wan, and R. Ye, The Comprehensive Diagnosis for the Vibration Troubles of Turbo-Operator Set, *Northeastern electric power technology*, Vol.22, No.8, 2001, pp.31-33.

[15] H. Su, and Q. Li, Cooperative Multi-Immune-Agent Systems for Substation Fault Diagnosis Based on Ontology, *WSEAS Trans. on Power Systems*, Vol.1, No.2, 2006, pp.446-453.

[16] H. Su, and H. Dong, Transformer Fault Diagnosis Based on Reasoning Integration of Rough Set and Fuzzy Set and Bayesian Optimal Classifier, *WSEAS Transactions on Circuits and Systems*, Vol.8, No.1, 2009, pp.136-145.

[17] L. Pan, L. Zhang, J. Zhang, Application of Rough Sets in Image Feature Selection, *Journal of Data Acquisition & Processing*, Vol.17, No.1, 2002, pp.42-45.

[18] S. Cong, *Theory and Application of Neural Networks*, USTC Press, 2003.

[19] J.Kennedy and R C Eberhart, Particle Swarm Optimization, *Proceedings of IEEE International conference on Neural Networks,*1995, 1942-1948.

[20] Y. Shi, and R C Eberhart, A Modified Particle Swarm Optimizer, Institute of Electrical and Electronics Engineers,No.5,1998, pp.69-73.