Send Orders for Reprints to reprints@benthamscience.net

The Open Mechanical Engineering Journal, 2014, *8*, 81-84

Open Access

Application of Neural Network Integration in Fault Diagnosis

Wei Xiong*[,1,2], Xuehui Xian[2] and Lijing Zhang[2]

[1]*School of Control and Computer Engineering, North China Electric Power University Baoding, 071003, China*

[2]*Information and Network Management Center, North China Electric Power University Baoding, 071003, China*

Abstract: According to the generation methods of individual neural network and the methods of generating conclusions from integrated neural network, an effective neural network integration system can be constructed. An optimization method for neural network integration is proposed. In the generation of individuals in the network integration, a variety of genetic algorithms and particle swarm optimization algorithm are used to train individual networks, thus to improve the precision of network members and reduce the correlation among the network members; in the conclusion generation, weight of the individual neural network is dynamically determined. The simulation results show that the effectiveness and feasibility of the method in fault diagnosis.

Keywords: Dynamic selectivity, fault diagnosis, kernel fuzzy C-means clustering integration, neural network integration, particle swarm optimization algorithm, weight.

1. INTRODUCTION

In recent years, with the development of artificial intelligence, expert system, neural network technology and knowledge discovery theory, intelligent condition monitoring and fault diagnosis technology has become a hot research in fault diagnosis. Neural network has been widely used in the fault diagnosis because of its self-learning, nonlinear pattern recognition, associative ability, fault tolerance and functional very strong approximation ability [1,2]. But it is a prominent problem to solve how to construct and train the optimized neural network, and to improve the generalization ability of neural network.

In 1990, Hansen and Salamon put forward neural network integration method innovatively [3], which showed that the training of multiple neural networks and their results could be merged and significantly improved the generalization ability of neural network system, which was considered to be a very effective engineering neural computing method. In recent years, many researchers have used neural network integration in fault diagnosis field, and achieved good results. However, in the process of neural network integration method, it still needs to carry out deep research in the theory and practice about how to combine the results of multiple neural networks, how to generate individual networks in the integration network and how to effectively use the finite samples.

2. THE STATUS QUO OF NEURAL NETWORK INTEGRATION METHOD IN FAULT DIAGNOSIS

2.1. The Method Based on Traditional Neural Network Integration

Hongzhi Zang, *et al.* [4] studied the neural network integration using radial basis functions, which has been successfully applied in transformer fault diagnosis. Qihua Xu, *et al*. [5] proposed a kind of neural network integration fault diagnosis method based on AdaBoost. Caili Song [6] constructed a neural network integration with 3 levels which were connected in series through the establishment of information distribution networks, sub neural network and decision information fusion network, and improved the equipment fault diagnosis. M.A. El-Gamal, *et al.* [7] applied the neural network integration to fault diagnosis of analog circuits. Di Xiao, *et al.* [8] also had good performance for fault diagnosis using the rough radial basis function integration. Tianyu Liu, *et al.* [9] introduced the selective integration learning technique, and used the clustering algorithm for the selective neural network integration learning technique.

The traditional neural network integration method means that individual neural networks will be fixed if they are selected, and they will be permanently out of neural network integration if they are not selected, which will result in losing a lot of useful information. Some neural network removed will get even better test effect than the selected individual network for some test samples. If these networks are removed ahead of time, the loss is not only a lot of useful information, but also the cost of training these neural networks. In addition, the analysis of test samples is a very important work, and each sample has its own characteristics, while the selectivity of the traditional integration method only has partial meaning, it cannot be moved after once chosen, and then a set of network integration are used for any test sample, which will cause a great loss of useful information.

2.2. The Method Based on Kernel Fuzzy C-means Clustering

The idea of Kernel fuzzy c-means clustering (KFCM) [10] algorithm is that the use of nonlinear mapping Φ (*) is used to transfer the input pattern vector space into a high dimensional feature space, and then the fuzzy C- means algorithm is used in the feature space to do fuzzy cluster analysis for the transformed feature vector Φ (xi). The KFCM integration technique extended fuzzy C- means algorithm in the feature space using the kernel method, did fuzzy clustering analysis for the transformed feature vector, which makes the similarity within groups be maximum and makes that between groups be minimum, and then to find the inherent characteristics of the object.

Baohai Huang, *et al*. [11] researched on the fault feature extraction based on Kernel Principal Component Analysis (KPCA), and proposed a neural network integration method based on KPCA-KFCM, which solved the selection problem of fault feature value and individual network in neural network integration, and was successfully applied in fault diagnosis of steam turbine. KPCA-KFCM integration method firstly uses Gauss function to calculate the eigenvalue and the eigenvector, then use kernel fuzzy c-means clustering algorithm to classify multiple BP individual neural networks which are independently trained on the training sample set, and finally calculated the generalization error of all the individuals in each category on the validation set, and set the BP neural network with minimum average generalization error as a representative of this kind of neural network to do generation, which ensure the accuracy and the difference of individual networks.

2.3. The Method Based on Dynamic Selectivity

The method independently trains multiple BP neural networks by the training sample set; then calculates the generalization error of each neural network in each sample of the validation set, and constitutes the performance matrix; next searches the sample closest to the test sample from the validation sample set, predicts the generalization error of the corresponding test samples according to the generalization error of different neural networks for these samples, and constitutes the performance matrix using the generalization error prediction information of various neural network for each test sample; finally, dynamically selects the individual neural network the generalization error of which is not larger than certain threshold for each test sample, the final output is obtained by the majority voting method. Thus the neural network integration for some sample can be dynamically selected according to the different test samples, using multiple neural networks at any time.

3. THE OPTIMIZED NEURAL NETWORK INTEGRATION METHOD

In these two methods, kernel fuzzy c-means clustering integration and dynamic selectivity integration, the single BP neural network is used. To further improve the integration difference and more efficiently integrate the conclusion of individual networks, an optimized neural network integration method is proposed in this paper.

3.1. Training the Members

Considering two aspects, improving the precision of network members in neural network integration and reducing the degree of correlation among the network members, two kinds of adaptive genetic algorithms (AGA) [12], two kinds of chaos genetic algorithms (CGA) [13], fuzzy genetic algorithm (FGA) [14], immune genetic algorithm (IGA) [15] and simulated annealing genetic algorithm (SAGA) [16] used to train 7 network members, and training samples of each network member were selected by random method. At the same time, the particle swarm optimization algorithm (c-PSOA) [17] was used to train the eighth network members, and the training samples of the eighth network members were the misclassified samples in the front seven network members.

The specific process of c-PSOA algorithm to train network members is as follows:

(1) To initialize the certain neural network structure, that is, to determine the neuron number in the input layer, the hidden layer and the output layer, and the layer number of hidden layer;

(2) To randomly initialize the position and velocity of each particle in the standard particle swarm, and each particle individual in the differential evolution particle swarm, which is represented by a M dimensional vector including M parameters need be optimized by neural network;

(3) Every vector is mapped as a set of parameters of the network to construct the network; and the training samples are input to train the network to evaluate each individual and calculate the fitness value;

(4) To evaluate the initial fitness value of each particle in the standard particle swarm, to save the historical best position of the initial individual and the corresponding fitness value, and to save the historical best position of the initial group and the corresponding fitness value;

(5) To evaluate the fitness value of each particle individual in the differential evolution particle swarm, and the best fitness value and the best individual are calculated;

(6) To calculate the new speed of the particle in the standard particle swarm, and to do the limiting process for the velocity of each particle. To calculate the new position of each particle;

(7) To update the individual historical best position of each particle in the standard particle swarm and the corresponding best fitness value, and that of the whole swarm and the corresponding fitness value;

(8) To do evolutionary computation for differential evolution particle swarm;

(9) To calculate the fitness change rate of the standard PSO particle swarm, and to determine whether it is lower than the set threshold, if less, then go to step (10), otherwise go back to step (6) to continue the search;

(10) To rank the two subgroups of particles by the fitness values, and to exchange the front good 0.382N

individuals in the differential evolution particle subgroup in the fitness value before with the back poor 0.382N individuals in the standard particle subgroup;

(11) If the best fitness function value is less than the given threshold, the search stops, the optimized neural network is output; otherwise return to step (6) to (9) to continue searching. If the fitness function value less than a given threshold is not obtained through the maximum iteration number, return to step (1) to change the structure of neural network, that is, increase the number of the hidden layer neurons, and then the changed neural network is optimized through step (2) to (11).

3.2. Determining the Weight

To make a conclusion using the method that dynamically determines the weight of integration individual neural network. The procedure identifying the dynamic weight of the individual neural network is as follows:

(1) According to the learning results of N training samples for all the M individual network members, for every network member, each sample is set a reliability function f_{mn} (m=1, 2, ..., M; n=1, 2, ..., N), the interval of which is [0,1]. The reliability function f_{mn} shows that the error characteristic of the n^{th} sample P_n after the m^{th} network member learning, the smaller the error is, the larger the reliability function f_{mn}.

$$f_{mn} = sat(x) = \begin{cases} 1 & x > 1 \\ x & 0 \le x \le 1 \\ 0 & x < 0 \end{cases} \tag{1}$$

$$x = 1 - \theta(T_n - \overline{T}_{nm})^2 \tag{2}$$

where, T_n means the ideal output of network member corresponded with the n^{th} sample P_n; $\overline{T}_{nm}$ represents the actual output of the n^{th} sample P_n by the m^{th} network member learning; θ is an adjustable positive constant, usually taken as 1.

(2) For a new set of the i^{th} data to be detected P_{N+i}, the distances D_{in} between P_{N+I} and all the N samples (Euclidean norm) are calculated, to find the training samples P_L corresponded with the minimum value D_{iL} = *minimum* (D_{i1}, D_{i2}, …, D_{iN}), then the dynamic weight of the i^{th} data to be detected P_{N+i} for M individual network members is $\omega_{mi} = f_{mL}$ (m=1, 2, …, M).

Thus, the neural network integration output y_i of the M individual network members for the i^{th} data to be detected P_{N+i} is calculated as:

$$y_i = \sum_{m=1}^{M} \omega_{mi} y_{mi} \Big/ \sum_{m=1}^{M} \omega_{mi} \tag{3}$$

where, y_{mi} is the diagnostic output of the m^{th} of individual network member for the i^{th} data to be detected P_{N+i}.

3.3. The Simulation Experiment

Six kinds of faults were simulated in the rotor test rig, including rotor imbalance, rub, rotor crack, deviation, bearing loose and oil film breakdown. The fault features were the various frequency components, including (0.1-0.39)f1, (0.40-0.49)f1, 0.5f1, (0.51-0.99)f1, 1f1, 2f1, (3-5)f1, (5-)f1, where f1 is the basic frequency. In the experiment, 98 fault feature data were used, each individual neural network randomly selected 60 samples from 80 samples, and the remaining 18 samples were used to evaluate the effect of the neural network integration

According to the optimized integration method, the training and test results of steam turbine fault samples are shown in Table **1**, where AGA1-INN, AGA2-INN, CGA1-INN, CGA2-INN, FGA-INN, IGA-INN, SAGA-INN, PSOA-INN respectively represent 8 individual network member through 8 neural network training algorithms;

Table 1. Results of fault diagnosis of turbo-generator using neural network integration.

Method	Mean Square Error of Sample Training	Misclassification Number when Threshold is 0.5		Misclassification Number when Threshold is 0.3	
		Train Sample	Test Sample	Train Sample	Test Sample
AGA1-INN	0.0195	0	0	8	2
AGA2-INN	0.016	2	0	10	2
CGA1-INN	0.0187	2	1	3	1
CGA2-INN	0.0162	0	0	2	1
FGA-INN	0.0155	2	1	5	1
IGA-INN	0.0159	3	1	4	1
SAGA-INN	0.0136	0	0	6	2
PSOA-INN	0.0036	0	0	0	0
Integration A-INN	—	0	0	2	1
Integration B-INN	—	0	0	1	0
Integration C-INN	—	0	0	0	0

Integration A-INN means the simple average weighted integrated network output, that is, the output of each network member is integrated in the same weight; Integration B-INN represents the average integrated weighted network output based on the training error, that is, the output of each network member is integrated according to the sum of the mean square error of the network to all the training sample; Integration C-INN is the network output based on the dynamic weighted average integration. "Mean square error of sample training" represents the sum of the overall error square after the network training. From Table **1**, for the training samples or test samples, the threshold is 0.5 or 0.3, the proposed method has the completely correct fault classification.

CONCLUSION AND FUTURE WORKS

The neural network integration based on particle swarm optimization algorithm has strong global search ability, which can improve the effectiveness and efficiency of the integration, provides an effective method for fault diagnosis of the steam turbine, and is worth being a further study. Petri net theory is based on the network theory, algebraic theory etc., and can systematically analyze and describe the structure, function and process of the fault diagnosis system using the graphic language, is an important tool of modeling and analysis of asynchronous concurrent diagnosis system. The combination of neural network and Petri net is yet another important research direction of neural network integration method and its practical application in the fault diagnosis is worthy of further study.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] Y. Xuemin, and P. Bo, "Cause analysis of increased carbon content of fly ash and countermeasure in fired coal boiler", *Boiler Technology*, vol. 35, pp. 49-51, 58, 2004.

[2] K. Arthur, "Robust soft sensors on integration of genetic programming analytical neural networks and support vector machines evolutionary computation", *Proceedings of the 2002 Congress*, Vol. 1, pp. 896-901, 2002.

[3] W. Dongfeng, H. Baohai, and D. Ze, *et al*. "Application of optimized neural network ensemble in fault diagnosis of turbo-generator," *Intelligent Information Management Systems and Technologies*, vol. 4, pp. 80-83, 2008.

[4] Z. Hongzhi, H. Yuhua, and Y. Xiaodong, "Integrated ann based on radial bases function applied in transformer fault diagnosis," *Proceedings of the EPSA*, vol. 15, pp. 51-77, 2003.

[5] X. Qihua, and Y. Rui, "Simulation research on fault diagnosis using adaboost algorithm", *Computer Engineering and Design*, vol. 26, pp. 3210-3212, 2005.

[6] S. Caili, "Integrated ueural technique and network diagnosis its application", *Journal of Xi'an University of Science and Technology*, vol. 28, pp. 569-572, 2009.

[7] M.A. El-Gamal, M. Abdulghafour, "Fault isolation in analog circuits using a fuzzy inference system", *Computers & Electrical Engineering*, vol. 29, pp. 213- 229, 2003.

[8] X. Di, and H. Shousong, "Ellipsoidal basis functional neural network based on rough K-Means", *Journal of Nanjing University of Aeronautics & Astronautics*, vol. 38, pp. 321-325, 2006.

[9] L. Tianyu, and Z. Yanchi, "Clustering algorithm based selective ensemble techniques for gear fault diagnosis", *Journal of Shanghai Dianji University*, vol. 12, pp.118-124, 2009.

[10] S. Murry, and T. Johansen, "Multiple model approaches to modeling and control", London: Taylor and Francis, 1997.

[11] H. Baohai, L. Yan, and W. Dongfeng, *et al*. "Steam turbine fault diagnosis based on KPCA and KFCM ensemble", *Electric Power Automation Equipment*, vol. 30, pp. 84-87, 2010.

[12] M. Scrinvas, and L.M. Patanaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 24, pp. 656-667, 1994.

[13] L. Chenzhong, and S. Huihe, "Evolutionary algorithms with chaotic mutations", *Control and Decision*, vol. 15, pp. 557-560, 2000.

[14] I. Tazawa, S. Koakutsu, and H. Hirata, "An evolutionary optimization based on the immune system and its application to the VLSI floor-plan design problem", *Trans. of the Institute of Electrical Engineers of Japan*, Part C, vol.117-C (7), pp. 821-828, 1997.

[15] I. K. Jeong, and J. J. Lee, "Adaptive simulated annealing genetic algorithm for control application", *International Journal of System Science*, vol. 27, pp. 241-253, 1996.

[16] F.T. Lin, C.Y. Kao, and C.C. Hsu, "Applying the genetic approach to simulated annealing in solving some NP-hard," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, pp. 1752-1767, 1993.

[17] M. Heng, and W. Yongchu, "A Cooperative particle swarm optimization algorithm based on differential evolution", *Information and Control*, vol. 37, pp. 176- 180, 2008.

Received: November 11, 2013 Revised: February 11, 2014 Accepted: March 3, 2014