http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (21): 6475-6480, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.6475.6480

Application of Radial Basis Function Optimized by Quantum Particle Swarm Optimization Algorithm in Electric Power System

[1]Yuhong Zhao, [2]Lin Lei and [1]Yifa Sheng
[1]School of Electric Engineering, University of south China, Hengyang, Hunan, China
[2]Institute of Environmental Protection and Safety Engineering, University of south China, Hengyang, Hunan, China

Abstract: The development of smart grid and electricity market requires more accurate and faster short-term load forecasting. Aiming at the problems of Radial Basis Function (RBF) network in electric system short term load forecasting, a novel algorithm integrated the advantages of RBF and Quantum Particle Swarm Optimization algorithm (QPSO) is proposed to improve the short-term load forecasting accuracy and speed. In this study, radial basis function network is trained by QPSO. After confirmed the nodes number of hidden layer, all network parameters are coded to individual particles to optimize learning algorithm. Then, the parameter can search optimal-adaptive value in global space. Using the optimized network to forecast load, the case analysis shows that, compared with the traditional network method, the new algorithm has better predictive ability on power system short-term load due to the higher predict precision and faster convergence.

Key words: Short-term load forecasting, quantum particle swarm optimization, radial basis function

INTRODUCTION

Load forecasting has been widely concerned in modern power system planning, operation, and control. According to the lead time of forecasting, power system load forecasting is divided into short-term, middle-term, and long-term load forecasting. Generally, short-term forecasting is a few hours or one-day ahead prediction. Short-term load forecasting is required for maintenance scheduling, power system security, economical dispatch, and market operation. The accuracy of short-term load forecasting can ensure the security and the efficiency of system operations. Many methods have been proposed for short-term load forecasting in the last decades. (Parvania and Fotuhi-Firuzabad, 2010). Generally, the traditional methods use statistical modes for short- term load forecasting, such as time series, linear regression methods, exponential smoothing and Kalman filters. But these techniques are basically linear models, which suffer from the nonlinearity and only provide reasonable accuracy. Due to the development in artificial intelligence techniques, the artificial neural network, fuzzy theory, genetic algorithm, expert system, the wavelet analysis, and support vector machine have been used for short-term load forecasting (Zhao *et al*., 2013). Among them, radial basis function neural networks have been applied in the short-term load forecasting. Since RBFNNs have only one hidden layer and have fast convergence speed. Besides, the RBFNNs are often referred to as model-free estimators since they can be used to approximate the desired outputs without requiring a mathematical description of how the outputs functionally depend on the inputs. However, optimizing the structure of the RBFNN is still a challenging work (Duan *et al*., 2009). Recently, Particle Swarm Optimization (PSO) has evolved as an important branch of stochastic techniques to explore the search space for optimization. The PSO method has the following advantages over other techniques. The first one is its characteristics of stable convergence. The second one is that it can generate a high-quality solution within shorter calculation time in comparing with other stochastic methods. The third one is its successful applications in many nonlinear and highly complex problems without gradient techniques. Therefore, PSO method has been developed to be real competitors with other well- established techniques for population-based evolutionary computation. Several researchers have successfully applied the PSO algorithm in the learning and structure improvement of the neural network for the short-term load forecasting. However, in the PSO algorithm, the acceleration coefficients are constant and the inertia weight is linearly adaptive (Zeng *et al*., 2011). Sun *et al*. (2004), Quantum-behaved Particle Swarm Optimization (QPSO) is proposed to adjust inertia weight and acceleration coefficients. The major objective is to achieve fast speed of convergence and better solution accuracy with minimal computational burden.

Corresponding Author: Yuhong Zhao, School of Electric Engineering, University of south China, Hengyang, Hunan, China

Integrating the advantages of RBF and Quantum Particle Swarm Optimization algorithm (QPSO), a new method based on QPSO-RBF is adopted to the practical application for predicting the short-term power load in this study. The best value problem of the network structure parameters (center, threshold and width, etc.) of the RBF network is effectively solved.

RADIAL BASIS FUNCTION NETWORK

Radial basis function network is a kind of forward neural network put forward by J.Moody and C.Darken in the 1980s, and it has very good performance. It not only has good generalization ability, the global best approximation ability, and less amount of calculation, has been widely used in pattern recognition, function approximation, adaptive filter, nonlinear time series prediction, etc. (Chang *et al*., 2009).

Radial basis function network is a local approximation network, generally including 3 layers (n inputs, m hidden node, p output). The structure is shown in Fig. 1. The first layer is the input layer which is composed by the signal source node; the second layer is the hidden layer, it will map the input space to a new space, its unit number depends on the needs of the problem described; the third layer is the output layer, it make response to the function of the input mode. A transformation from the input space to the hidden layer space is nonlinear, it is a kind of nonlinear transformation for feature extraction, and the space from the hidden layer to the output layer space transformation is linear. The transformation function of the hidden layer unit is RBF function, which is a non-negative nonlinear function of a local distribution center point radial symmetric attenuation.

Basis function of RBF network is commonly used Gaussian function, which can be expressed as:

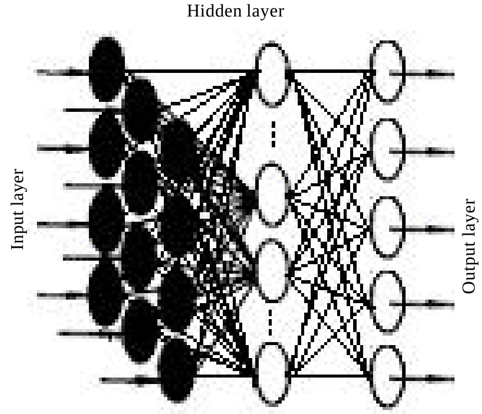

Fig. 1: Radial gaussian function network topology

$$\phi_i(x) = \exp\left[\frac{-\|X - c_i\|^2}{2\sigma_i^2}\right], i = 1, 2 \cdots, m \quad (1)$$

where, $\phi_i(x)$ is the output of the first i nodes of hidden layers; X is the input sample, $x = (x_1, x_2, \ldots, x_n)^T$; c_i is the first i hidden layer node of Gaussian kernel centers and X has the same dimension; σ_i is the first i variables of hidden layers nodes, called normalization constant or the base width. RBF network output is the output of hidden layer nodes in a linear combination:

$$y_k = \sum_{i=1}^{m} w_{ik}\phi_i(x), k = 1, 2, \cdots, p \quad (2)$$

QUANTUM PARTICLE SWARM OPTIMIZATION ALGORITHM

Particle swarm optimization (PSO) algorithm: Particle Swarm Optimization (PSO) is an evolution computing technology based on swarm intelligence proposed by Eberhart and Kennedy in 1995, and it is based on the simulation of the flock. Described as follows: a flock of birds search food in random, a region where only a piece of food, the most simple and effective strategies to find food is to search the area around the recent bird from food, which constitutes one of the basic concepts of the PSO. When PSO is used to solve the optimization problems, the solution of the problem corresponds to a bird in the search space position, saying that these birds as "particle."Each particle has its own position and velocity (determine the direction and distance of flight) and there is a fitness decided by optimal function. First of all, the PSO is initialized to a group of random particles (random solutions). And then the optimal solution is found by iteration .In each iteration, the particles update themselves by tracking two "extreme". The first one is the optimal solution found by the particle itself. This solution is called individual extreme value p_{Best}. Another extreme value is the optimal solution of the entire population found currently. This is global extreme value g_{Best}. When the optimal value was found, the two particles update their speed and position according to the Eq. 3-4:

$$V_{ij}^{(k+1)} = \omega v_{ij}^{(k)} + c_1 r_1^{(k)}(p_{ij}^{(k)} - x_{ij}^{(k)}) + c_2 r_2^{(k)}(p_{g}^{(k)} - x_{ij}^{(k)}) \quad (3)$$

$$x_{ij}^{(k+1)} = x_{ij}^{(k)} + V_{ij}^{(k+1)} \quad (4)$$

where, j expressed the jth-dimension of particles;i expressed particles; r_{1j}, r_{2j} expressed the random number in the interval [0, 1]; c_1 is called cognitive factor and expressed believe degree on experience; it can adjust the step size for particles to fly the direction of its own best position; c_2 is known as the coefficient of social learning, and expressed the believe degree on individuals around, it can adjust the step size for particles to fly the direction of its global best position. The algorithm iteration termination condition is generally chosen as the maximum number of iterations or fitness value satisfies a predetermined threshold value of the minimum fitness after searching the optimal location so far. ω expressed the inertia weight coefficient. The study found large ω can enhance global search ability of PSO, and smaller ω can enhance the local search ability of PSO.

Quantum particle swarm optimization algorithm: Through the analysis shows that the particles in the classical PSO algorithm find the optimal value through approach the Pi and Pg. For some problem of the optimum point away from Pi and Pg, it is more difficult to find the optimal solution using this algorithm, that is, the algorithm can guarantee the global convergence. Aiming at the convergence problem of PSO algorithm, a new model of the PSO algorithm was proposed from the perspective of quantum science by Sun *et al.* (2004).This model is based DELTA potential well, which is considered that particle has a quantum behavior. According to this model, Quantum-behaved Particle Swarm Optimization was proposed; the experimental results show that QPSO convergence performance has been greatly improved. The algorithm changes the evolutionary search strategy of the whole of PSO algorithm. It can be to search in the whole feasible solution space, and don't need the velocity vector in the evolution equation, the form of evolution equations is simpler, smaller, and the parameters are easier to control. So the searching ability of QPSO algorithm is better than that of all the PSO algorithm has been developed. In order to guarantee the convergence of the algorithm, each particle must converge to their respective point P, P = (p_1, p_2,..., p_n,), this is determined by the following of particles and the aggregation of particle swarm. The j-dimensional coordinate of point P of particle i is:

$$P_j = \frac{(c_1 \times r_{1j} \times p_{ij}^{(k)} + c_2 \times r_{2j} \times p_{gj}^{(k)})}{c_1 \times r_{1j} + c_2 \times r_{2j}} \tag{5}$$

A global point is introduces to calculate the next iteration of particles in particle swarm, which is defined as the average m_{best} of the local optimal location of all the particles:

$$\text{mbest} = \sum_{i=1}^{M} P_i / M = \left(\sum_{i=1}^{M} P_{i1} / M, \sum_{i=1}^{M} P_{i2} / M, \ldots, \sum_{i=1}^{M} P_{iD} / M \right) \tag{6}$$

The particle iterative equation becomes:

$$X(k+1) = P \pm \beta * |\text{mbest-}X(k)| * \ln\frac{1}{u}, u = \text{rand} \tag{7}$$

$$\beta = (\beta_1 - \beta_2) \times (\text{MAXITER} - k) / \text{MAXITER} + \beta_2 \tag{8}$$

Wherein: M is the number of particles; n is the dimension of the particles; P_i is the best position of the particle i; u is the random number in the interval; β expressed the coefficient of contraction expansion, which is an important QPSO convergence parameter, β_1, β_2, respectively the initial and final values. P_i is p_{best} of the ith particle, MAXITER is the maximum number of iterations, k is the current iteration number. By the literature, when β_1 and β_2 are more than 1.7, QPSO algorithm cannot be made convergence; when β_1 is 1.2 and β_2 is 0.7 the algorithm can be made to achieve better convergence. Equation 5-8 is called the quantum particle swarm algorithm, referred QPSO.

The algorithm steps are as follows:

- The position value X (k) of the particle swarm is randomly initialized, to determine the number of particles and the maximum allowable number of iterations k_{max}
- According to the optimized function f (.), the fitness of each particle (X (k)) is obtained and compared with the optimal value f (P_i) of the individual history; if the current value of fitness is better than that of the optimal value of individual history, the current value have replaced for individual optimal value, otherwise don't replace
- Fitness values of all particles are evaluated to obtain g_{Best}
- m_{best} is obtained by using Eq. 5-6
- The particles are updated according to the Eq. 5-8
- When the iterations achieve to the maximum iteration times or meet the minimum error requirement, the iteration will be stop. Otherwise, go to Eq. 2

QPSO-RBF LOAD FORECASTING MODEL AND APPLICATION

Load forecasting issues related description: Learning and training on the basis function RBF network, taking into account the respective input data of the network often have different dimensions, in order to avoid saturation phenomenon and the complexity of the training of the neurons,the input data must be pretreated. First Load value is transformed with normalization method; followed by the value of date type, temperature and precipitation quantify are making between interval [0, 1]. In addition, the adverse load data caused by some kind of reason to deviate from the reasonable value is pseudo-data, which should be removed in the forecast. Comparison method is used in this article. We compare the load value of a certain moment and its before and after moment load value, if the difference is greater than a certain threshold, automatically correct pseudo data by Matlab program; If one day without data or pseudo data too much, the days of data can be regarded as defect, then the day data can use a few days before and after the normal data repair, in order to ensure data integrity further standardizationl.

Selection of the network input node and network model: The short-term load forecasting model based on QPSO-RBF uses a three layers network, its composition is: The input layer is a layer with 17 neurons (as described in Table 1, except the load values use normalized data, and the rest use quantized coefficient); the output layer is a layer with 1 neurons, corresponding to the predictive value at moment on the forecast date.

The RBF network based on QPSO learning algorithm design:

- To determine radial basic unit number: To determine the number of hidden units in the network training is to determine the optimal number of clustering. This article adopts the Competitive Learning RPCL (originally described Penalized Competitive Learning) algorithm of opponents punished to determine the radial basic unit number. The algorithm can not only modified winning entries to fit the input values, the clustering number can also be determined automatically by punishing time item vector (smaller) the method of forced redundant nodes from clustering data. According to the literature (Xu *et al*., 1993), RPCL algorithm is adopted to define the radial grass-roots unit number
- The key of using QPSO to train the RBF network weights and threshold is to establish the mapping relationship between QPSO particle dimensions space and the neural network connection weights, threshold values. According to RBF network shown in Fig. 1, each particle position vector is: $X = (w_1, w_I, c_1, c_I, \sigma_1,..., \sigma_I)$, where I is the number of neurons in the hidden layer. All the parameters of RBF network are then coded into individual represented real digital string, according to the particle size of the group, in accordance with the individual structures randomly generated a certain number of individuals (particles) populations, in which different individuals on behalf of a different set of neural network parameters, meanwhile initialization p_{best}, g_{best} (Shi *et al*., 2009)
- Corresponding to each individual neural network input training samples for training. Computing mean square error generated each network on the training set (that is, the particle's fitness) which is defined as the squared error between the k-th iteration actual network output Y 'and the ideal output Y, as shown in Eq. 9

$$F(t) = \sum_{t=1}^{n} (Y_t' - Y_t)^2 \quad (9)$$

where, n is the number of samples, F(t) is the objective function to evaluate particle swarm all individuals, to find the best individual (that is, the particle swarm the least mean square error of the individual) is used to determine whether need to update the g_{best} and p_{best} of particles
- According to the QPSO model, position vector of each individual is updated to produce new individual particles. The newly generated particles continue to be mapped to the parameters of the network and enter the training samples to train the network, so the algorithm is repeated until the termination condition is satisfied, the output of a set of parameters with the best fitness value is the final result, RBF network training algorithm ends

Table 1: Network input information

Input neurons	Corresponding input
1	maximum temperature of the week before forecast date
2	minimum temperature of the week before forecast date
3	the average temperature of the week before forecast date
4	Weather conditions during the period of t in the week before forecast date
5	load value during the period of t in the week before forecast date
6	load type of the week before forecast date
7	maximum temperature of the day before forecast date
8	minimum temperature of the day before forecast date
9	the average temperature of the day before forecast date
10	the weather conditions during the period of t in the day before forecast date
11	load value during the period of t in the day before forecast date
12	load type of the day before forecast date
13	maximum temperature of the forecast date
14	minimum temperature of the forecast date
15	the average temperature of the forecast date
16	Weather conditions during the period of t in the forecast date
17	load type of the forecast date

Forecasting model simulation: In order to verify the feasibility and effectiveness of load forecasting based QPSO_RBF learning algorithm,in this study, we uses the historical load data and the corresponding weather data of a city in Hunan Province in March 2008-March 2009 (except holidays) as training data to predict the area March 27, 2009 24-point load. Set up network initial

Table 2: Result of load forecasting

Time (h)	Actual value (Mw)	Algorithm of this article	
		Predictive value (Mw)	Relative error (%)
01:00	666.04	670.77	0.71
02:00	622.82	632.91	1.62
03:00	656.35	657.6	0.19
04:00	625.40	622.40	-0.48
05:00	659.93	650.36	-1.45
06:00	653.70	649.97	-0.57
07:00	776.84	785.00	1.05
08:00	755.77	760.98	0.69
09:00	784.51	804.20	2.51
10:00	785.84	789.22	0.43
11:00	820.35	830.77	1.27
12:00	839.38	849.37	1.19
13:00	772.91	783.65	1.39
14:00	744.53	751.16	0.89
15:00	661.23	669.69	1.28
16:00	704.23	717.68	1.91
17:00	719.88	714.41	-0.76
18:00	790.81	778.32	-1.58
19:00	823.20	842.63	2.36
20:00	871.53	877.98	0.74
21:00	799.17	809.64	1.31
22:00	792.93	810.93	2.27
23:00	694.28	704.62	1.49
24:00	732.15	739.69	1.03

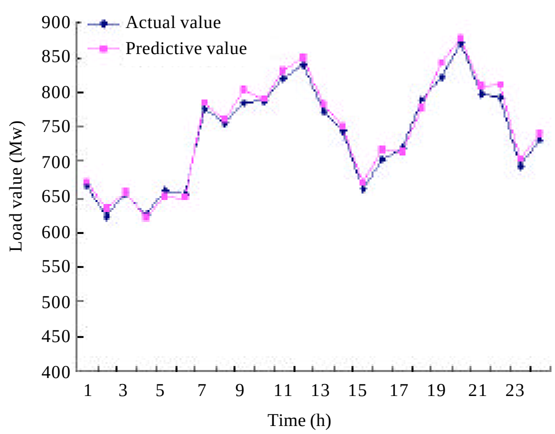

Fig. 2: Comparison of the load actual value and forecast value

hidden node number is 20; the number of particles initialized is 50; the weight values ω is selected between 0.4~0.8; iteration times is 230 times. After learning samples 226 times, the effective number of hidden nodes is obtained to be 16. The predicted results are shown in Table 2 and Fig. 2.

It can be seen from the Table 2 and Fig. 2, Load forecasting curve of the date and the actual load curve shape is roughly the same. The relative error of all points are less than 3%, mainly in plus or minus 1.2%, the maximum relative error is 2.51%, the minimum relative error is 0.19%. Predict obtained more satisfactory results.

CONCLUSION

The key of accurate short-term load forecasting of power system is to build up the forecast model. This study proposes a new RBF neural network based on quantum particle swarm algorithm to use in load forecasting, various parameters of the RBF network composed of a multi-dimensional vector, as particles are optimized, so that can be search the optimal solution within the scope of the solution space. Compared with basic particle swarm optimization algorithm, this algorithm calculation is easier and has faster convergence speed, and stronger global searching ability, parameter estimation is more accurate. The proposed method is applied to the training of RBF neural network power system load forecasting, simulation examples show that the method has good prediction accuracy and has a certain value in the project.

ACKNOWLEDGMENTS

This study was financially supported by Hunan Provincial Education Science "twelfth five-year" plan of 2013 projects (Project No. [XJK013CGD111]) research, the Scientific Research Fund of Hunan Provincial Science and Technology Programme (Project No. [2014FJ3157]) and Hengyang City Science and Technology Programme (Project No. [2013KJ31]).

REFERENCES

Chang, Y.J., S.H. Wang and H.Y. Sun, 2009. Research of short-term load forecasting algorithm based on wavelet analysis and radial basis function neural network. Proceedings of the 2nd International Conference on Power Electronics and Intelligent Transportation System, Volume 1, December 19-20, 2009, Shenzhen, pp: 81-84.

Duan, Q.C., M. Zhao, D.X. Wang and P. Duan, 2009. Short-term electricity price forecasting based on hybrid particle swarm optimization and normalized radial basis function neural network. Power Syst. Prot. Control, 37: 37-39.

Parvania, M. and M. Fotuhi-Firuzabad, 2010. Demand response scheduling by stochastic SCUC. IEEE Trans. Smart Grid, 1: 89-98.

Shi, B., Y.X. Li, X.H. Yu, W. Yan, P. Li and X. Meng, 2009. A modified particle swarm optimization and radial basis function neural network hybrid algorithm model and its application. Proceedings of the WRI Global Congress on Intelligent Systems, Volume 1, May 19-21, 2009, Xiamen, pp: 134-138.

Sun, J., B. Feng and W.B. Xu, 2004. Particle swarm optimization with particles having quantum behavior. Proceedings of the Congress on Evolutionary Computation, June 19-23, 2004, Portland OR, USA., pp: 325-331.

Xu, L., A. Krzyzak and E. Oja, 1993. Rival penalized competitive learning for clustering analysis, rbf net and curve detection. IEEE Trans. Neural Networks, 4: 636-649.

Zeng, Q.W., Z.H. Xu and J. Wu, 2011. Forecasting of electricity load based on particle swarm optimization and support vector machine. Microelectronics Comput., 28: 147-153.

Zhao, Y.H., X.C. Zhao and W. Cheng, 2013. The application of chaotic particle swarm optimization algorithm in power system load forecasting. Adv. Mater. Res., 614-615: 866-869.