Fuzzy Neural Networks Learning by Variable-dimensional Quantum-behaved Particle Swarm Optimization

Jing Zhao*[1], Ming Li[2], Zhihong Wang[3]
[1]School of Information, Qilu University of Technology
Jinan 250353, Shandong, China, Ph./Fax: +86-531-89631256
[2]College of Science and Technology, Shandong University of Traditional Chinese Medicine
Jinan 250355, Shandong, China, Ph./Fax: +86-531-89628261
[3]Shandong Yingcai University
Jinan 250104, Shandong, China, Ph./Fax: +86-531-88253001
*Corresponding author, e-mail: zjstudent@126.com

Abstract

The evolutionary learning of fuzzy neural networks (FNN) consists of structure learning to determine the proper number of fuzzy rules and parameters learning to adjust the network parameters. Many optimization algorithms can be applied to evolve FNN. However the search space of most algorithms has fixed dimension, which cannot suit to dynamic structure learning of FNN. We propose a novel technique, which is named variable-dimensional quantum-behaved particle swarm optimization algorithm (VDQPSO), to address the problem. In the proposed algorithm, the optimum dimension, which is unknown at the beginning, is updated together with the position of swarm. The optimum dimension converged at the end of the optimization process corresponds to a unique FNN structure where the optimum parameters can be achieved. The results of the prediction of chaotic time series experiment show that the proposed technique is effective. It can evolve to optimum or near-optimum FNN structure and optimum parameters.

Keywords: *fuzzy neural networks,parameters learning,quantum-behaved particle swarm optimization algorithm, structure learning,variable dimension*

1. Introduction

Fuzzy neural networks (FNN) [1] has the advantage of fuzzy system and neural networks, which includes the exact fitting and learning ability of neural networks and the powerful knowledge representation of fuzzy logic inference. The learning of FNN consists of structure learning to determine the number of fuzzy rules, and parameters learning to adjust the FNN's parameters. Too many fuzzy rules make the network structure become complex and have the poor generalization capability and over fitting. Whereas too few fuzzy rules can not better fit the network. So it is important to optimize structure of FNN. The parameters learning of FNN is a function optimization problem after the structure of FNN is obtained. Many algorithms can be applied to optimize the number of fuzzy rules and the parameters learning, such as GA [2], particle swarm optimization (PSO) [3] and hybrid algorithm [4, 5].

PSO is an evolutionary computation technique developed by Eberhart and Kennedy in 1995 [6]. However, the algorithm cannot converges to the global minimum point with probability one [7]. Jun Sun *et al.* propose a global convergence-guaranteed PSO, quantum-behaved particle swarm optimization (QPSO) algorithm [8, 9, 10], which is inspired in quantum mechanics. It has been shown that QPSO is a strong universal optimization technique and can be applied in various complicated optimum problem.

In general, most intelligent algorithms can be applied to a search space with fixed dimensions. However the dimension of some optimization problem is dynamic, such as the structure learning of FNN. In order to address this problem, we present a novel variable-dimensional quantum-behaved particle swarm optimization algorithm (VDQPSO), which can optimize both the structure learning and parameters learning of FNN.

The rest structure of this paper is as follows. In section 2, a brief introduction of the structure of FNN is presented. PSO, QPSO and the novel algorithm is described in section 3.

Received April 19, 2013; Revised July 15, 2013; Accepted July 25, 2013

Next, the structure learning and parameters learning of FNN are depicted in section 4. Then the experiment results are given in section 5. Finally, the conclusion is put forward in section 6.

2. The Structure of Fuzzy Neural Networks

The Takagi-Sugeno-Kang (TSK)[11,12] fuzzy inference model of FNN is proposed in our technique. Each fuzzy rules corresponds a sub-FNN. The form of the rule R_j, which is the *j*th fuzzy rules of first order TSK fuzzy neural network, is as follows:

R_j : if x_1 is A_{1j} and x_2 is A_{2j} ... and x_n is A_{nj}

Then $y_j = \sum_{k=1}^{K} v_{kj}\varphi_k$

Where x_i and y_j are input and local output variables respectively, A_{ij} is the fuzzy set of input universe with the membership function $\mu_{A_{ij}}$, n is the number of input variables, v_{kj} is the link weight of local output, φ_k is the trigonometric function of input variables, K is the number of function variables.

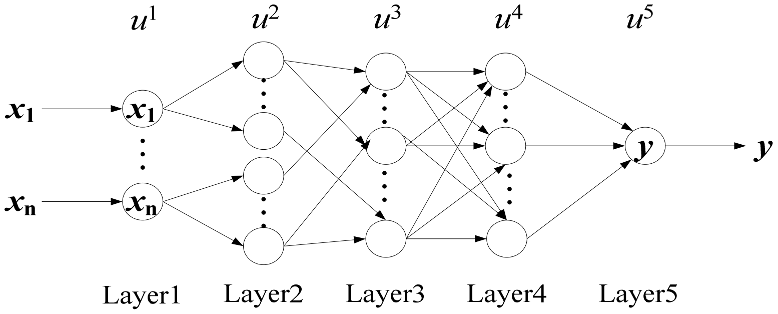

Figure 1. The structure of FNN

Layer 1, the input layer, transmits input value to the next layer. The output of this layer is $u_i^1 = x_i$.

Layer 2 is fuzzed layer. Membership function is used to obtain membership value on a fuzzy set in this layer. Gaussian function is implemented in our technique. That is $u_{ij}^2 = \mu_{A_{ij}}^2 = \exp(\frac{-(u_i^1 - m_{ij})^2}{\sigma_{ij}^2})$, m_{ij} and σ_{ij} are the mean and variance of the Gaussian function, respectively, of the *j*th term of the *i*th input variable x_i.

Layer 3 is fuzzy inference layer. The firing strength of corresponding rules is given by the output of layer 3. That is $u_j^3 = \prod_i u_{ij}^2$, where $\prod_i u_{ij}^2$ is multiplication of input variables.

The output of Layer 4 is $u_j^4 = u_j^3 \sum_{k=1}^{K} v_{kj}\varphi_k$ .Where v_{kj} is the corresponding weight of FNN, and φ_k is a trigonometric function, given by $[x_i \quad \sin(\pi x_i) \quad \cos(\pi x_i)]$. Therefore $K = 3*n$, n is the number of input variables.

Layer 5 is de-fuzzy layer. In this layer, the node integrates all of above layer output as a center of area and de-fuzzy with the expression $y = u^5 = \frac{\sum_{j=1}^{R} u_j^4}{\sum_{j=1}^{R} u_j^3}$, where R denotes the number of rules, y is the output of the FNN model.

3. Variable-Dimensional Quantum-Behaved Particle Swarm Optimization Algorithm

3.1. Quantum-Behaved Particle Swarm Optimization Algorithm

In PSO algorithm, the population has *M* particles in the *D*-dimensional space. Each particle represents a potential optimum solution vector in solution space, whose value is evaluated by objective function. The position vector and velocity vector of the particle i at generation t are represented as $x_i(t) = (x_{i1}(t), x_{i2}(t), \cdots, x_{iD}(t))$ and $v_i(t) = (v_{i1}(t), v_{i2}(t), \cdots, v_{iD}(t))$. The particle moves according to the equations:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_1 (pbest_{i,j} - x_{i,j}(t)) + c_2 r_2 (gbest_j - x_{i,j}(t)) \quad (1)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad (2)$$

Where $i = 1,2,...,M$, $j = 1,2,...,D$, w is the inertia weight. c_1 and c_2 are called the acceleration coefficients. r_1 and r_2 are random number. $pbest_i = (pbest_{i1}, pbest_{2i}, \cdots, pbest_{iD})$ denotes the previous best position of particle i with the name personal best position($pbest$), while vector $gbest = (gbest_1, gbest_2, \cdots, gbest_D)$ is recorded as the global best particle position($gbest$) in the whole swarm.

The state of particle is depicted by its position vector and velocity vector, which determine the trajectory of the particle in PSO algorithm. However in quantum world, the position and the velocity of a particle cannot be determined simultaneously according to *uncertainty principle*. Therefore, if individual particles in a PSO system have quantum behavior, the PSO algorithm is bound to work in a different fashion.

In quantum time-space framework, the quantum state of a particle is depicted by wave function, instead of position and velocity. Inspired PSO in quantum mechanics, Jun Sun *et al.* proposed QPSO. The equations are as follows:

$$p_i = \varphi \times pbest_i + (1-\varphi) \times gbest \quad (3)$$

$$mbest = \frac{1}{M}\sum_{i=1}^{M} pbest_i \quad (4)$$

$$x_i(t+1) = p_i \pm \beta \left| mbest - x_i(t) \right| * \ln(\frac{1}{u}) \quad (5)$$

Where φ and u are random number uniformly distributed in $(0,1)$. p_i is called local attractor. $mbest$ is mean best position of the population.Parameter β is called the contraction-expansion coefficient. In the process of iteration, ± is decided by the random number, when it is bigger than 0.5, minus sign (—) is proposed, others plus sign (+) is proposed.

3.2. Variable-Dimensional QPSO

The major drawback of basic and some improved QPSO algorithms is that the dimension of search space is fixed. In many optimization problems, the optimum dimension is dynamic. In order to address this problem, we present a variable-dimensional QPSO (VDQPSO) technique, which negates the need of fixing the dimension of the solution space in advance [13]. The VDQPSO can seek the best dimension and the best position on the optimum dimension in optimum process.

In VDQPSO, as an optimum objective, the dimension is updated together with the position of the particle. In order to accomplish this, each particle has two sets of components. The first one is the position of the particle in *D*-dimensional search space. The second one is a positive integer N , which denotes the dimension of the particle, limited in the range $[Nmin, Nmax]$. The two components, each of which has been subjected to two independent and consecutive processes, have its personal best position ($pbest$). Accordingly, the first process is positional QPSO. It means the standard QPSO positional shifts in *D*-dimensional search space. Each particle records its last position and personal best position on the given dimension so that this information can be used when it revisits the same dimension at a later time. The second process is dimensional QPSO. In this process, the dimension of each particle can be chose freely in the particular range. The particle will remember its current positional status and keep up the positional QPSO process on the new dimension.

In VDQPSO, the characteristics of each particle i at generation t in the swarm X are represented:

$dx_i(t)$: the dimension component of particle i at generation t .

$dpbest_i(t)$: the personal best dimension component of particle i at generation t .

$dgbest(t)$: the global best dimension at generation t .

$xx_{i,j}^{dx_i(t)}(t)$: the *j*th dimension of the position of particle i on dimension $dx_i(t)$.

$xpbest_{i,j}^{dx_i(t)}(t)$: the *j*th dimension of the personal best position of particle i on dimension $dx_i(t)$.

$xgbest_j^d(t)$: the *j*th dimension of the global best position of swarm on dimension d .

At generation t, the positional components update of particle i is performed on its current dimension $dx_i(t)$, and then the dimensional components are updated to determine the next dimension $dx_i(t+1)$. The equations of positional update are as follows:

$$xp_i^{dx_i} = \varphi \times xpbest_i^{dx_i} + (1-\varphi) \times xgbest^{dx_i} \tag{6}$$

$$xmbest^{dx_i} = \frac{1}{M}\sum_{i=1}^{M} xpbest_i^{dx_i} \tag{7}$$

$$xx_i^{dx_i(t)}(t+1) = xp_i^{dx_i} \pm \beta \left| xmbest^{dx_i} - xx_i^{dx_i(t)}(t) \right| * \ln(\frac{1}{u}) \tag{8}$$

Note that the particle's new position $xx_i^{dx_i(t)}(t+1)$ will still be on the same dimension $dx_i(t)$. The particle may move to another dimension with the update of the dimension. However the equation of the dimensional QPSO need to improve because the dimension must be an integer. The equations are as following in our method:

$$dp_i = \varphi \times dpbest_i + (1-\varphi) \times dgbest \tag{9}$$

$$dmbest = \frac{1}{M}\sum_{i=1}^{M} dpbest_i \tag{10}$$

$$dx_i(t+1) = \left\lfloor dp_i \pm \beta \left| dmbest - dx_i(t) \right| * \ln(\frac{1}{u}) \right\rfloor \tag{11}$$

where $\lfloor \cdot \rfloor$ is the floor operator.

For the swarm, firstly, the global best position $xgbest$ is recorded in positional QPSO. Then in dimensional QPSO, the global best dimension ($dgbest$) is achieved by comparing the fitness score of each particle to each other on its personal best dimension. The positional QPSO process alternates with dimensional QPSO process until the VDQPSO process terminates. Finally, the optimum solution $xgbest$ on the optimum dimension $dgbest$ will be achieved.

4. The Structure Learning and Parameter Learning of FNN

In this section, we apply the VDQPSO technique to evolve the fuzzy rules and parameters of FNN. Figure 2 represents the coding of a particle. Where i and j are the *i*th input variable and the *j*th rule. R denotes the number of rules. Suppose the length of each rule is RS, and the initialization swarm is *X(popsize, dimension)*. Where *popsize* and *dimension* is the size and dimensions respectively in swarm X. Thus $dimension = R \times RS$, $Rmin \le R \le Rmax$. $Rmin$ and $Rmax$ are the minimum and maximum of rules of FNN.

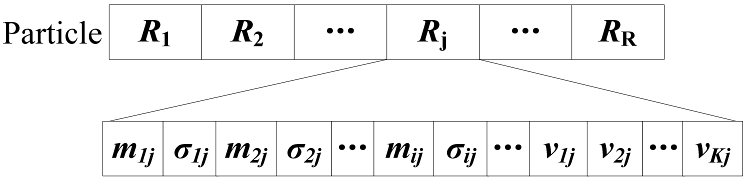

Figure 2. The coding structure of a particle

In VDQPSO, the number of rules update instead of the dimensional update for FNN is recorded. The process is described step-by-step as follows.

Step 1 Initialization phase. Firstly, initialize the number of rules dx and the particles' position xx. Then the best personal dimension $dpbest$, the global best dimension $dgbest$, the best personal position $xpbest$ and the global best position $xgbest$ are initialized too.

Step 2 Update the particles' position used equation (6), (7) and (8).

Step 3 Update the particles' dimension used equation (9), (10) and(11).

Step 4 Evaluate the fitness function value of each particle in its corresponding rules. In the particular rules, if the new score is better than previous positional score, then update $xpbest$ and $xgbest$. If the new score is better than previous dimensional score, then update $dpbest$ and $dgbest$.

Step 5 Repeat Step 2-4 until stop criterion is satisfied.

5. Simulation Examples and Its Results

5.1. Prediction of Chaotic Time Series

In order to test and evaluate the proposed technique, the prediction of the chaotic time series, a benchmark problem, is performed. The chaotic time series is generated by chaotic Mackey-Glass [14] differential equation defined as follows:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{12}$$

Mackey-Glass equation is called the function of the time delay parameter τ. It produces chaotic behavior with $\tau > 16.8$, so $\tau = 17$ is adopted in our technique. The prediction of the chaotic time series is to use past values of the time series up to time t to predict the value in the future $t+p$. The standard of prediction is to create a mapping from S points of the time series spaced Δ apart, that is $[x(t-(S-1)\Delta), \cdots, x(t-\Delta), x(t)]$, to a predicted future value $x(t+p)$. $S=4$ and $\Delta = 6$ are used usually. The fourth-order Runge-Kutta method is applied to obtain the time series value at each integer time point. From the time series $x(t)$, 1500 input-output data pairs are extracted to test the novel algorithm performance. The format of the series is $[x(t-18), x(t-12), x(t-6), x(t); x(t+6)]$. Where $x(0) = 1.2$. There are four input variables and one output in the FNN model. Divide the data pairs into 3 cross-validation groups, each group have 1000 data pairs. The first group is $[x(1) - x(1000)]$. The second group is $[x(251) - x(1250)]$, and the third group is $[x(501) - x(1500)]$. The first 500 data pairs are used as the training data set, while the remaining 500 pairs are the testing data set to validate the novel method. The fitness function of the series is defined as the following formula:

$$F = \sqrt{\frac{1}{H}\sum_{h=1}^{H}(y_h - \overline{y_h})^2} \tag{13}$$

Where y_h is the *h*th model output, $\overline{y_h}$ represents the *h*th desired output, and H is the number of training data.

5.2. Parameters Setting of Algorithm and Results Discussions

In this section the effectiveness of the novel algorithm is illustrated. The algorithm parameters setting are described as follow: In our technique, the value of β varies from 1.0 to 0.5 linearly. $Rmin = 2$ and $Rmax = 12$ are proposed. The swarm size is set to 50. The number of function variables $K = 3*4 = 12$ is defined. The maximum generation is set to 2000. Because of the stochastic characteristic of VDQPSO algorithm, the experiment is run 10 independent times.

Firstly, we choose the data group 2 randomly to determine the number of fuzzy rules. Then the first 500 data pairs of data group 2 are set used to simulate the results as the training data set. The iterative process of the number of rules, minimum errors, $xgbest$ and $dgbest$ are keep track of in the VDQPSO implementation process. Table 1 records the $dgbest$ at the end of each iterative process. The optimum dimension is the number that is repeated most often in Table 1. Then the determined number of rules is used to validate the performance of the novel technique on data group 1 and data group 3 so that the results can be achieved. As can be seen from Table 1, the number 6, 7 and 11 are selected two times. The errors of VDQPSO, called the best Root Mean Square (RMS) error, between predictive value and the actual value are reported in our experiment over 2 times of running.

The selection process of rules are illustrated in Figure 3, Figure 4 and Figure 5. In Table 2 the minimal training data RMS error, corresponding testing data RMS error and the average RMS error on three data groups are given. Figure 6, Figure 7 and Figure 8 illustrate respectively the convergence process of average RMS error with different number of rules on three data groups. The graph of predictive value and actual value on data group 1 is presented in Figure 9. The detailed graph of the circle on the Figure 9 is described in Figure 10.

From Table 2, Figure 6-10 we can obtain some analysis results. As can be seen from Table 2, the minimum and average RMS error are increasing with the augment of number of rules on three data groups in general. However the performance on data group 1 and data group 3 with 7 rules is superior to other number of rules. As is illustrated in Figure 6 and Figure 8, it makes no difference to the RMS on data group 1 and data group 3 whether the number of rules is 6 or 7. In Figure 7, the result with 7 rules outperforms the result with 6 rules and 11 rules On data group 2. Note that the worst performance on all data group is that the number of rules is 11. It means that too many number of rules result in too many errors. As is can be seen from the detailed graph, the fitting effect with 7 rules out perform others. However in some generations, the result with 11 rules is the best.

All the experimental results show the effectiveness of the proposed VDQPSO.

Table 1. The number of rules

Number of run	Number of rules	Number of run	Number of rules
1	6	6	7
2	11	7	7
3	11	8	2
4	6	9	9
5	8	10	5

Table 2. The RMS error

Number of rules	Data sets	Data1 error	Data2 error	Data3 error	Average error
6	Training	0.0231	0.0226	0.0191	0.0216
	Test	0.0229	0.0230	0.0196	0.0218
7	Training	0.0215	0.0251	0.0184	0.0217
	Test	0.0215	0.0260	0.0193	0.0223
11	Training	0.0306	0.0328	0.0271	0.0302
	Test	0.0311	0.0335	0.0270	0.0305

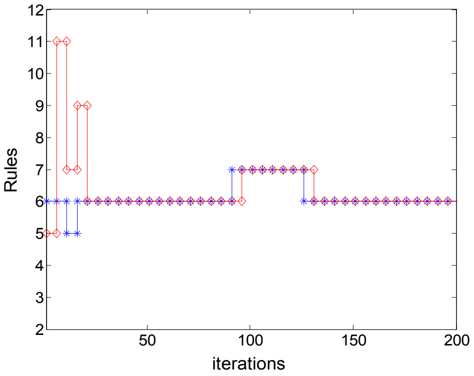

Figure 3. The selection process of rule=6

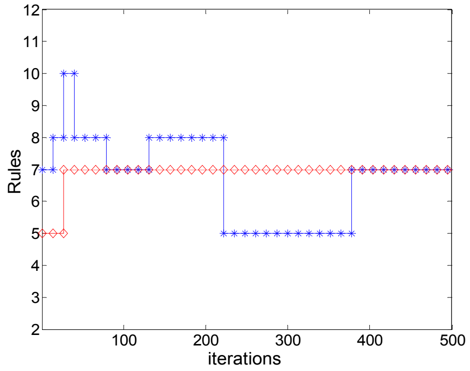

Figure 4. The selection process of rule=7

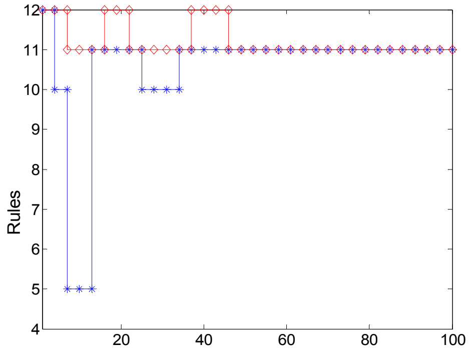

Figure 5. The selection process of rule=11

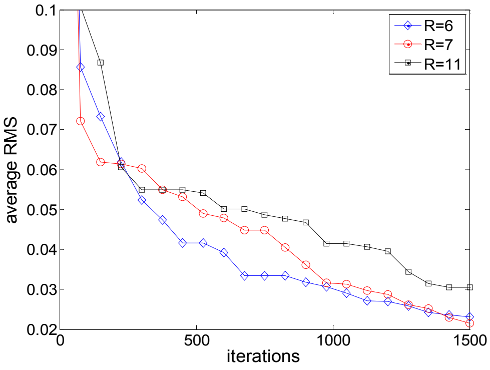

Figure 6. The convergence graph of average RMS on data group 1

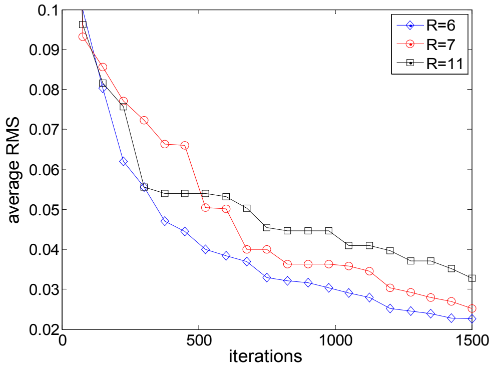

Figure 7. The convergence graph of average RMS on data group 2

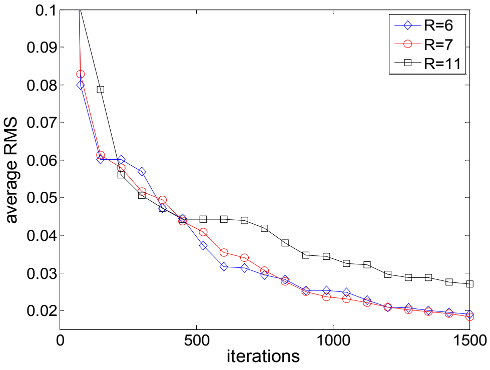

Figure 8. The convergence graph of average RMS on data group 3

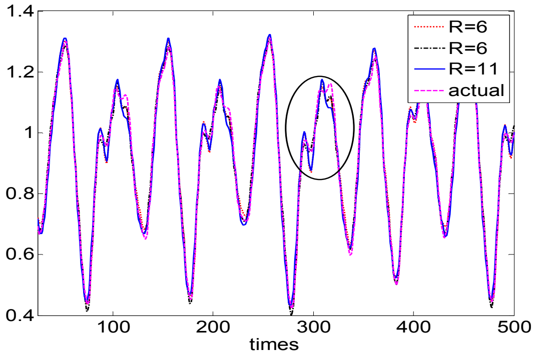

Figure 9. The graph of predictive and actual value on data group 1

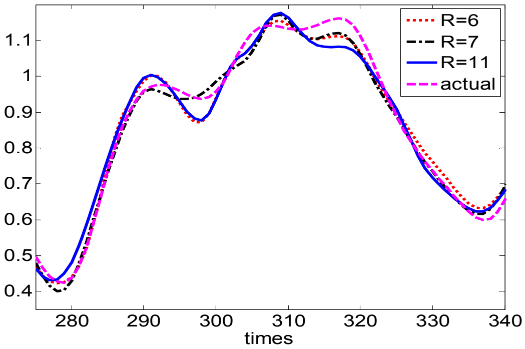

Figure 10. The detailed graph of circle on Figure 9

6. Conclusion

In all kinds of optimization algorithms, the dimension of the search space must be fixed in advance. However the structure learning of FNN is a dynamic process. It is difficult to determine the proper number of fuzzy rules in advance. In this paper, we propose a novel evolutionary method named variable-dimensional quantum-behaved particle swarm optimization algorithm (VDQPSO) for structure and parameters learning of FNN. The novel technique promises an especial solution to address the above drawback. In VDQPSO, the dimension of the particle, i.e. the number of fuzzy rules, which is regarded as an optimization objective, is updated together with the position of the particle. At last, all the particles can converge to the global solution on the optimum dimension in a simultaneous way. Experimental results show that the proposed method is effective.

References

[1] Tawafan A, Sulaiman MB, Ibrahim ZB. Adaptive Neural Subtractive Clustering Fuzzy Inference System for the Detection of High Impedance Fault on Distribution Power System. *IAES International Journal of Artificial Intelligence (IJ-AI)*. 2012; 1(2): 63-72.

[2] Tang AM, Quek C, Ng GS. GA-TSKfnn: Parameters Tuning of Fuzzy Neural Network Using Genetic Algorithms. *Expert Systems with Applications*. 2005; 29(4): 769-781.

[3] Lin CJ, Lee CY. Non-linear System Control Using A Recurrent Fuzzy Neural Network Based on Improved Particle Swarm Optimization. *International Journal of Systems Science*. 2010; 41(4): 381-395.

[4] Jiao SH, Jiang ZY, Bu JL. Application of Artificial Neural Network Based on GA-BP Algorithm to Predicting Mechanical Properties of TC4. *Advanced Materials Research*. 2010; 146-147: 1698-1701.

[5] Juang CF, Wang CY. A Self-generating Fuzzy System with Ant and Particle Swarm Cooperative Optimization. *Expert Systems with Applications*. 2009; 36(3): 5362–5370.

[6] Kennedy J, Eberhart R. *Particle Swarm Optimization*. Proceedings of the IEEE International Conference on Neural Networks. Perth. 1995; 4: 1942-1948.

[7] Frans VDB. An Analysis of Particle Swarm Optimizers, PhD. Thesis, University of Pretoria, South Africa; 2002.

[8] Sun J, Feng B, Xu WB. *Particle Swarm Optimization with Particles Having Quantum Behavior*. Proceedings of the IEEE Congress on Evolutionary Computation. Portland. 2004; 1: 325-331.

[9] Sun J, Xu WB, Feng B.*A Global Search Strategy of Quantum-behaved Particle Swarm Optimization*. Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems. Singapore. 2004; 1: 111-116.

[10] Fang W, Sun J, Ding YR, Wu XJ, Xu WB. A Review of Quantum-behaved Particle Swarm Optimization. *IETE Technical Review*. 2010; 27(7): 336-348.

[11] Rezaee B, Zarandi MHF. Data-driven Fuzzy Modeling for Takagi-Sugeno-Kang Fuzzy System. *Information Sciences*.2010; 180(2): 241-255

[12] Tewari A, Macdonald MU. Knowledge-based Parameter Identification of TSK Fuzzy Models. *Applied Soft Computing.* 2010; 10(2): 481-489.

[13] Kirabyaz S, Pulkkinen J, Gabbouj M. Multi-dimensional Particle Swarm Optimization in Dynamic Environments, *Expert Systems with Applications*. 2011; 38(3): 2212–2223.

[14] Farzad M, Tahersima H, Khaloozadeh H. *Predicting the Mackey Glass Chaotic Time Series Using Genetic Algorithm*. Proceedings of the International Joint Conference SICE-ICASE. Busan. 2006: 5460-5463.