

Adaptive Learning of RBF Network Based on Adaptation Complexity

Fan Yang^{1,2}, Xingxing Liu^{2,*} and Fang Deng¹

¹HuBei University of Education, Wuhan, 430205, P.R. China; ²Management School, Wuhan University of Technology, WuHan, P.R. China

Abstract: Radial neural network can be used to decompose complex problems with good biological properties. Adaptive control for neural network is helpful to improve the efficiency of pattern classification. In order to solve pattern classification with different adaptive characteristics, multiple adaptive algorithms are embedded in radial neural network. Through the test of the objective function, it is found that not all of the combinational algorithms can get desired results. After systemic tests, it is found that shifting strategy in the later stage of learning process can get good effect with the appropriate change strategy. Keeping consistent major evolutionary strategy is correct and necessary. Only if the simulation stop optimizing, appropriate other strategy should be taken for reaching better effect.

Keywords: Adaptability, learning step, radial neural network.

1. INTRODUCTION

Biological intelligence is the origin and end-result of intelligent science. Natural intelligence creates bionics, which transform biological structure and function into science and technology for diverse machinery. How to measure the adaptation complexity of various algorithms is crucial. The exploration on it does some contributions to intelligent algorithms. Intelligent algorithm can also be applied to the module clustering. Clustering is the core process of modular design, which is a process of comprehensive multi-objective optimization. The clustering methods can be divided into two kinds as qualitative and quantitative. Qualitative methods are the numerical calculation methods, mainly including heuristic [1] and tree graph [2]. Quantitative are value calculation methods, consisting mainly of the k-means, rough set and neural network [3-6]. These algorithms are technically extended by correlation coefficient, average and variance. The integration of clustering algorithm does not make any exception.

Many problems can be cut down to pattern classification. Clustering is an effective method to solve the problem of approximation and classification. Super large-scale classification problems have the characteristics of massive data, diverse structures, multiple patterns, different scale and dynamic change. Parallel computing and machine learning is helpful to solve the problem of pattern classification. 'Divide and dispose' strategy is the usual way to processing parallel computing on super large-scale pattern classification. Common decomposition schemes can be divided into three kinds, including 'one-versus-all', 'one-versus-one' and 'part-versus-part'. In the strategies' implementation, common classifiers can be mainly divided into two kinds, including the nearest neighbor classifier and radial basis function network (RBF) [7].

Radial basis function network has biological features and fine scalability. It is lack of adaptive mechanism on the learning processing of radial basis function network in current study. A multiple adaptive learning model is designed for radial basis function network, which can be used to solve clustering and approximation problems.

2. MATERIALS AND METHODS

2.1. Basic Algorithm of RBF

BF network [8] with structure of n-h-1 is the major modified model in this paper. In this RBF network, there are n input nodes and h hidden nodes (See Fig. 1).

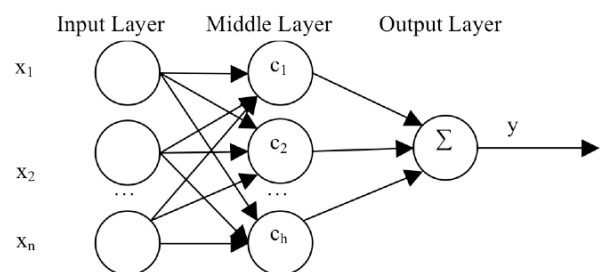


Fig. (1). Classic RBF network.

The activation function of i th input node is equation (1).

$$\phi_i(x) = e^{-\frac{\|x-c_i\|^2}{r_i^2}} \quad (1)$$

x is the input and $x \in R_n$. c_i is the center of i th hidden node and $c_i \in R_h$. r_i is extension constant of the hidden nodes. The function of output y is equation (2).

$$y_k = f(x, \theta_{(s)}) = \sum_{i=1}^h w_i \phi_i(\|x - c_i\|) \quad (2)$$

w_i is output weight of the i th hidden node. $\theta(h)$ represents h hidden nodes' parameters, including data centers, extension

constants and output weights. A RBF network can be identified with exclusive hidden nodes' parameters. $\theta_{(h)} = \{\theta_i, i=1,2,\dots,h\}, \theta_i = \{c_i, w_i, r_i\}$. A single output RBF network with forgetting factor is used for testing the modified algorithm. The objective function is E as equation (3).

$$E = \frac{1}{2} \sum_{j=1}^N \beta_j e_j \quad (3)$$

β_j is a forgetting factor, e_j is an error signal as equation (4) shows.

$$e_j = y_j - \sum_{i=1}^h w_i \phi_i(X_j) \quad (4)$$

Considering the influence of all the training sample and forgetting factor. The adjustments of each parameter in θ_i are as equation (5),(6) and (7).

$$\Delta c_i = \eta \frac{w_i}{r_j^2} \sum_{j=1}^N \beta_j e_j \phi_i(X_j) (X_j - c_i) \quad (5)$$

$$\Delta r_i = \eta \frac{w_i}{r_j^3} \sum_{j=1}^N \beta_j e_j \phi_i(X_j) \|X_j - c_i\|^2 \quad (6)$$

$$\Delta w_i = \eta \sum_{j=1}^N \beta_j e_j \phi_i(X_j) \quad (7)$$

η is the learning step, which is often set as a constant. It will be a function with error as input. This is the major improvement for RBF network in this paper.

2.2. Adaptive Design

In generalization method for neural network, structure optimization design is the mainstream. There are other optimization design methods, including the initiative learning, noise interference, integration, optimal stopping and implication learning. The use of random noise in the sample input can improve the generalization ability of neural network, similar to the regularization method for structure design of the neural network. Stopping learning at the appropriate time can improve the generalization ability of neural network, which is also a kind of implicit regularization method. Adaptive adjustment on learning can make neural network adapt to the new features of the problem quickly, which is beneficial to improve the generalization ability of neural network.

The adaptive neural network proposed in this paper is mainly to check and determine whether the revised weight to reduce the effect of function value error. If error value reduces, learning speeds in relative low. Conversely, if the error does not reduce, learning speed is over-modulated. Then, a reverse adjustment corresponding to the situations is needed for achieving convergence.

A linear learning step function is a common alternative. However, it is complex on choosing learning step function in practical problem. It is necessary to make more practical learning step function. Pan Wang [9, 10] provided some alternatives as general modes (see Fig. 2).

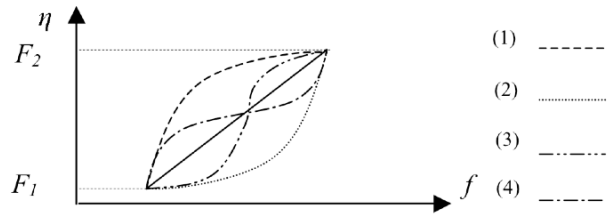


Fig. (2). Learning step function alternatives.

Assume F_1, F_2 are the learning step values as the objective function reaches maximum and minimum. F_1, F_2 . The two parameters should be selected manually by previous experience and current design. It is common and necessary that the value should be adjusted since the changes in different evolutionary stages. Assume individual initial value is E . $\eta(E)$ represents learning step. While $\max(E)$ and $\min(E)$ represent the objective's maximum and minimum of the current stage respectively.

(1) Decelerated good learning step function. The learning step value drops along with the increase of individual initial values. Fitness decreases along with the increase of the values.

$$\eta_1(E) = F_1 + \frac{F_2 - F_1}{1 - e^{-1}} \left(1 - e^{\frac{E - \min(E)}{\min(E) - \max(E)}} \right) \quad (8)$$

(2) Accelerated good learning step function. The speed of the learning step value increases along with the improvement of individual initial values.

$$\eta_2(E) = \frac{F_2 - e \cdot F_1}{1 - e} + \frac{F_1 - F_2}{1 - e} \left(e^{\frac{E - \min(E)}{\max(E) - \min(E)}} \right) \quad (9)$$

(3) Variable tendency of increase-drop learning step function. The speed of the learning step value first increases. Then it drops along with the improvement of individual initial values.

$$\eta_3(E) = F_1 + \frac{F_2 - F_1}{1 - e^{(\min(E) - \max(E))}} \left(1 - e^{\frac{(E - \min(E))^2}{\min(E) - \max(E)}} \right) \quad (10)$$

(4) Variable tendency of drop-increase learning step function. The speed of learning step first drops. Then it increases along with the improvement of individual value.

$$\eta_4(E) = F_1 + \sqrt{\ln(1 - (1 - e^{(F_2 - F_1)^2}) \cdot \frac{E - \min(E)}{\max(E) - \min(E)}} \quad (11)$$

In order to display different evolutionary strategy, different learning step function modes in selection operator set is conducive to the smooth fine-tuning, which can create more possibilities of composed or decomposed. Considering relative higher error tolerance of the neural network is in the early stage of the training, significant changes are suitable. On the contrary, smaller error tolerance in the later period allows only fine-tuning. If over-modulated in the early stage and relative-low in the later stage happen, fine tuning strategies have two kinds. Accelerated learning in low-section of the learning curve and decelerated learning in high-section. Or decelerated learning in low-section of the learning curve and accelerated learning in high-section. Random noise and optimal stopping and adaptive learning

Table 1. Learning combinational algorithm (A) of neural network.

(1-1)

	Early Stage (Higher Tolerance)	Later Stage (Small Tolerance)
Relative-low(should improve)	$\eta_1(E)$	$\eta_3(E)$
over-modulated(should reduce)	$\eta_4(E)$	$\eta_2(E)$

Table 2. Learning combinational algorithm (B) of neural network.

(1-2)

	Early Stage (Higher Tolerance)	Later Stage (Small Tolerance)
Relative-low(should improve)	$\eta_1(E)$	$\eta_4(E)$
over-modulated(should reduce)	$\eta_3(E)$	$\eta_2(E)$

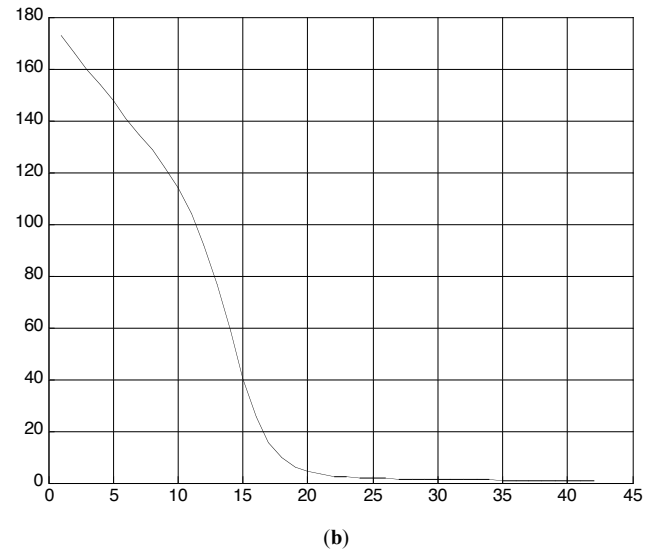
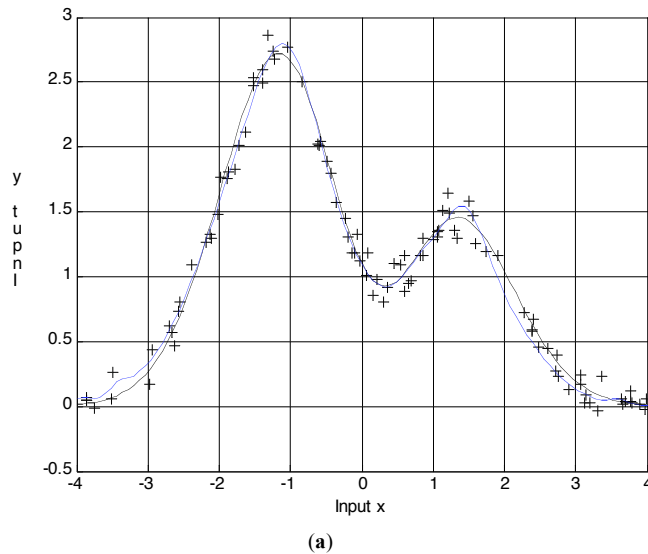


Fig. (3). Simulation results of original algorithm (A).

step are integrated into design a better adaptive RBF network. This network can adaptively adjust learning step according to the learning phase. Two preliminary combinational algorithms are as Tables 1 and 2 shows. These two algorithms are based on authors' experience on simulation with expectation of improvement.

3. RESULTS AND DISCUSSION

The approximation test of Hermit polynomial function (see equation 12) is used for testing new adaptive RBF network.

$$F(x) = 1.1(1 - x + 2x^2)e^{-x^2/2} \tag{12}$$

Total number of samples $N = 100$. Input sample x_i obeys uniform distribution $[4, 4]$. The output sample is $F_{(x_i)+e_i}$. Initial value of η is 0.001. Terminal objective function value is $E_0=0.9$. Through to simulate a variety of combinational algorithms, the possibility of performance improvement of neural network in pattern recognition with different combi-

national algorithms is explored. Simulation on the original algorithm gets optimal variance value of 0.8992 in the 42nd cycle (see Fig. 3).

In Fig. (3a), is a fitting figure, while (3b) is a curve of the objective function values. For learning combinational algorithm (A) of neural network, the conversion time (t_s) is set to 20 cycles. The simulation gets the optimal value of 1.0822, which does not meet the requirements in 5000 cycle. Obviously, combinational algorithm (A) does not reach requirement. If the conversion time (t_s) is set to 30, the result gets great improvement with this delay. The optimal value 0.9 can be reached at 300th cycle. If the conversion time (t_s) is set to more than 40, the simulation can get no more improvement. For learning combinational algorithm (B) of neural network, the conversion time (t_s) is set to 20 cycles. The simulation get the optimal value of 0.8993 at 42nd cycle. This result is similar to simulation results of original algorithm. This result shows learning combinational algorithm (B) is more suitable for intrinsic dynamic learning of RBF (See Fig. 4).

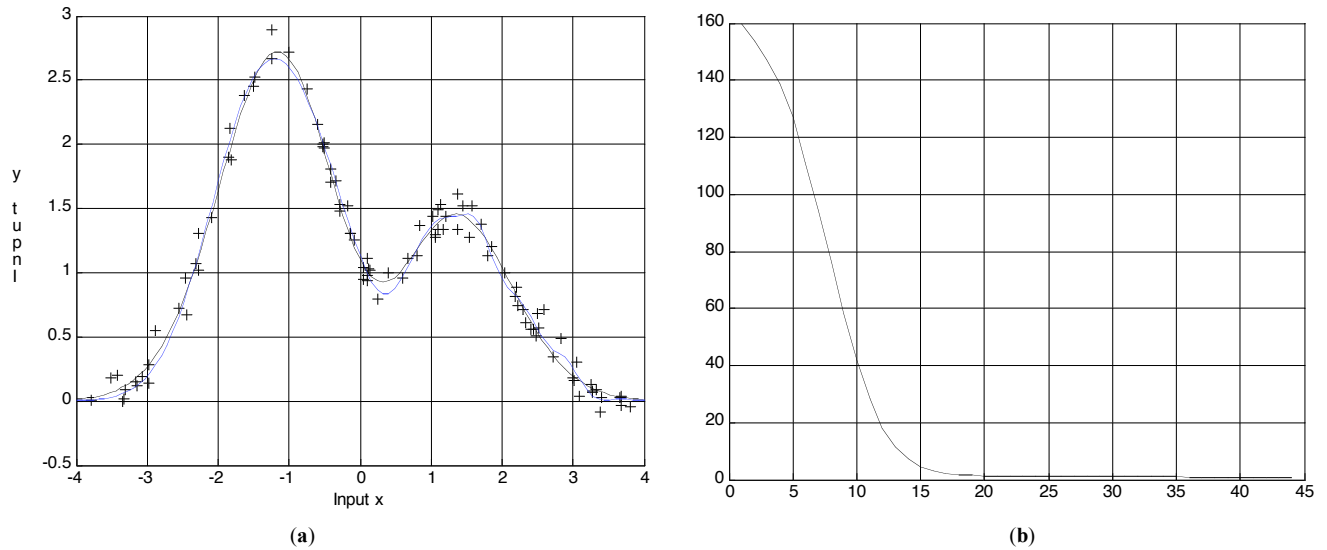


Fig. (4). Simulation results of original algorithm (B).

Table 3. Learning combinational algorithm (C) of neural network.

(1-3)

	Early Stage (Higher Tolerance)	Later Stage (Small Tolerance)
Relative-low(should improve)	$\eta_1(E)$	--
over-modulated(should reduce)	$\eta_4(E)$	$\eta_2(E)$

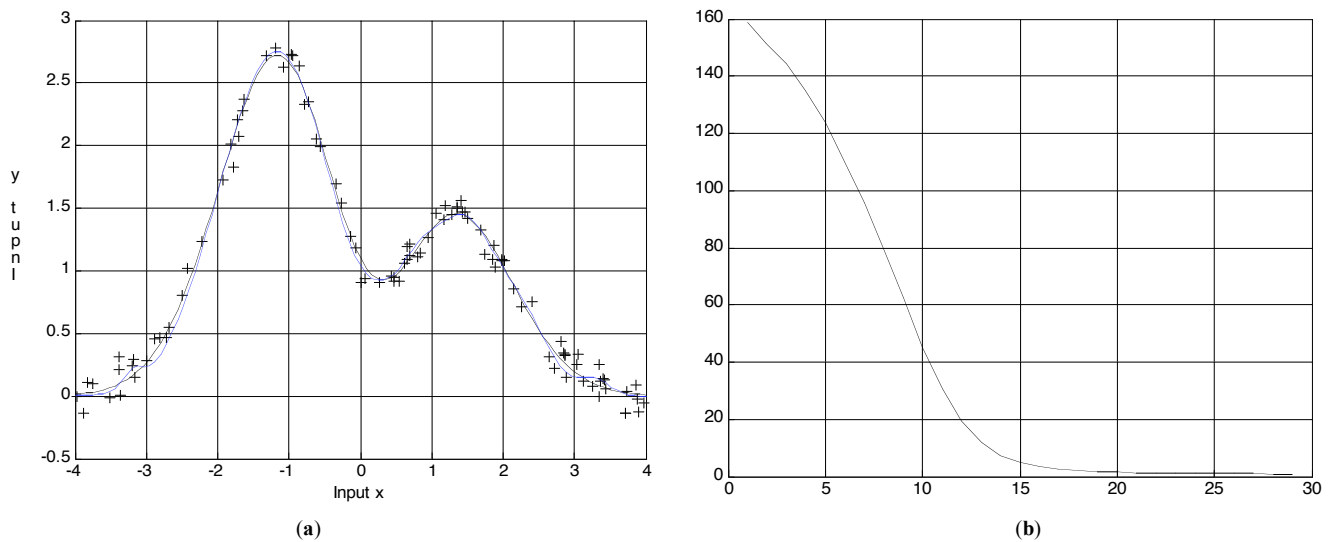


Fig. (5). Simulation results of original algorithm (C).

If the conversion time (ts) is set to 30, the result gets the optimal value 0.8990 at 73th cycle. If the conversion time (ts) is set to more than 40, the simulation can get no more improvement. Preliminary judgment can be summarized to that the application of combinational algorithm (B) is available. It seems that no certain judgments from the above simulations. More attempts is necessary.

After restructuring algorithms, it is found that the use of $\eta_2(E)$ and $\eta_4(E)$ is quite negative for convergence when learning step is relative-small. According many attempts, a new combinational algorithm (C) is found as Table 3 shows.

In the early stage, learning step function is set to $\eta_1(E)$ when learning step is relative-small, and set to $\eta_4(E)$ when learning step is over-modulated. In the later stage, learning step function is set to $\eta_2(E)$ when learning step is over-modulated. With combinational algorithm (C), the simulation can get optimal value 0.8789 at 29th cycle (See Fig. 5).

CONCLUSION

Adaptive set on dynamic learning of the RBF network can get some progresses. In the approximation process, it is found that different problems need to choose appropriate

adaptation function through simulating different adaptive model of learning process. The details are as follow:

(1) Frequent adjustments on learning step may be not beneficial to reduce error or early convergence. (2) Shifting strategy in the later stage of learning process can get good effect with the appropriate change strategy. (3) In the process of learning process, keeping consistent major evolutionary strategy is correct and necessary. Only if the situation stop optimizing, appropriate change strategy should be taken for reaching better effect.

CONFLICT OF INTEREST

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

ACKNOWLEDGEMENTS

This work was supported in part by the Chinese National Natural Science Foundation (No. 71371148), and by China postdoctoral Fund Projects (No. 2013 M542080 and No. 2014 T70750). Financially it supported by self-determined and innovative research funds of WUT (Grant No. 2012-YB-020).

REFERENCES

- [1] R.B. Stone, K.L. Wood, and R.H. Crawford, "A heuristic method for identifying modules for product architectures," *Design Studies*, vol. 21, no. 1, pp. 5-31, 2000.
- [2] H. Bunke, P. Dickinson, C. Irniger, and M. Kraetzl, "Recovery of missing information in graph sequences by means of reference pattern matching and decision tree learning," *Pattern Recognition*, vol. 39, no. 4, pp. 573-586, 2006.
- [3] H.E. Tseng, C.C. Chang, and J.D. Li, "Modular design to support green life-cycle engineering," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2524-2537, 2008.
- [4] V.B. Kreng, and T.P. Lee, "Modular product design with grouping genetic algorithm-a case study," *Computers and Industrial Engineering*, vol. 46, no. 3, pp. 443-460, 2004.
- [5] Y. Liu, J. Ding, and S. Zhang, "Method for determining the optimal module of product family based on right degree," *Journal of Machine Design*, vol. 27, no. 4, pp. 29-32, 2010.
- [6] X. Wang, Y. Chen, H. Zhao, and B.L. Lu, "Parallelized extreme learning machine ensemble based on min-max modular network," *Neurocomputing*, vol. 128, no. 1, pp. 31-41, 2014.
- [7] B.L. Lu, and K. Ito, "Converting general nonlinear programming problems into separable programming problems with feed forward neural networks," *Neural Networks*, vol. 16, no. 1, pp. 1059-1074, 2003.
- [8] G.B. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 57-67, 2005.
- [9] P. Wang, J. Zhang, and L. Xu, "How to measure adaptation complexity in evolvable systems-A new synthetic approach of constructing fitness functions," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10414-10419, 2011.
- [10] D.A. Silva, G.I. Alves, P.S.G. De Mattos Neto, and T.A.E. Ferreira, "Measurement of fitness function efficiency using data envelopment analysis," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7147-7160, 2014.

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© Yang et al.; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.