Radial Basis Function Neural Network With Dynamic Optimal Learning Rate & Genetic Algorithm

[1]HAN-LEIH LIU, [2]SENG KAH PHOOI
[1]School of Microelectronics, Griffith University, Kessels Rd, Nathan QLD 4111,AUSTRALIA
[2]School of Engineering, Monash University, Malaysia campus, 2 Jalan Kolej, Bandar Sunway, 46150 PJ, Selangor, MALAYSIA
kp_seng@yahoo.com, sikuo@lycos.com

Abstract: - This paper presents a training method for radial basis function neural network (RBFNN) based on GA and dynamic optimal learning rate. Genetic algorithm (GA) is applied to search for the optimal centers from input space of the RBF. Dynamic optimal learning rate approach is also appended to the genetic algorithms' searching procession to obtain a set of weighting factors so that the network actual output converges to the designed signal. Simulation results have revealed that the good performance of the proposed scheme.

Key-Words: Neural network, Radial Basis Function, Learning rate, Genetic Algorithm

1 Introduction

In the past decades, feedforwarded layered neural networks have been used in many fields, for example, signal processing, automatic control, image processing, etc. Many researchers have worked on neural networks and good achievements have been reached. The designed transfer function works in each neuron such that the proper value can be pumped out. Moody and Darken [1] proposed a RBFNN structure that can employ local receptive fields to perform function mappings.

Gradient techniques [2]-[5] is widely applied to do learning (or training) so that a set of expected optimal weighting factors can be computed and then the network actual output can converge the desired signal. During the procession of learning, the learning rate is a vital factor, which affects the speed of learning. Authors [2] proposed an approach that dynamic optimal learning rates for training a fuzzy neural network can be found. On the other hand, genetic algorithms [6]-[9], which have globally searching characteristic, are also commonly used in many areas.

In this paper, we presented a methodology that Gas are applied to seek optimal centers of RBFN while the approach for finding dynamic optimal learning rates is appended into them. Therefore the total error between the actual and desired data can be dramatically decreased. The optimal weighting factors and centers of RBFNN can be obtained faster. A simulation example is presented so that it can illustrate the performance of the proposed method.

2. RBF Neural Network Based On GA And Dynamic Optimal Learning Rate

In 1988, Moody and Darken [1] proposed a network structure, namely, a radial basis function network, which employs local receptive fields to perform function mappings $F:\Re^N \to \Re$..

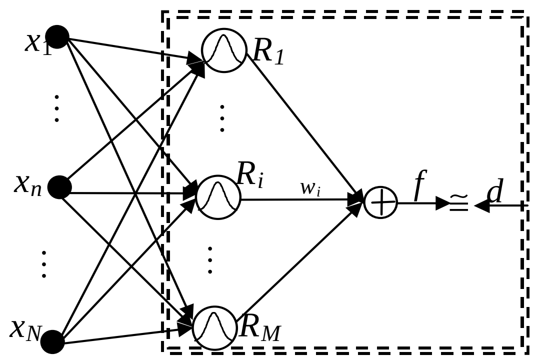


Fig. 1: Radial basis function network.

As shown in Figure 1, d is desired signal and response function R_i and output of network f are defined as follows:

$$R_i(\underline{x}) = \exp[-\|\underline{x} - \underline{c}_i\|^2 / \sigma_i^2] \qquad (1)$$

and $$y_j = f_j(\underline{x}_j) = \sum_{i=1}^{M} w_i R_i(\underline{x}_j) \qquad (2)$$

where $\underline{x} \in \Re^N$ is a input vector, $\underline{c}_i \in \Re^N$ is a center vector in the input space, $\sigma_i \in R$ is a width parameter, $W=[w_1,\cdots,w_M] \in \Re^M$ is a weighting vector.
Given a set of training vectors, which forms the training matrix $X \in \Re^{N\times P}$, it is desired to use the gradient technique [2]-[5] to train the above network so that the actual outputs converge to the desired outputs d's. The error function e_i is defined as

$$e_i = y_i - d_i \qquad (3)$$

Then we have the total squared error J:

$$J = \frac{1}{2P} \underline{e}^T \underline{e} \tag{4}$$

where $\underline{e} = [e_1, \ldots e_P]^T$.

To update weighting factor W, the gradient method is applied:

$$W(t+1) = W(t) - \beta_t \frac{\partial J}{\partial W(t)} \tag{5}$$

$$= W(t) - \beta_t \frac{1}{P} \underline{e}_{1\times p} R_{p\times M}$$

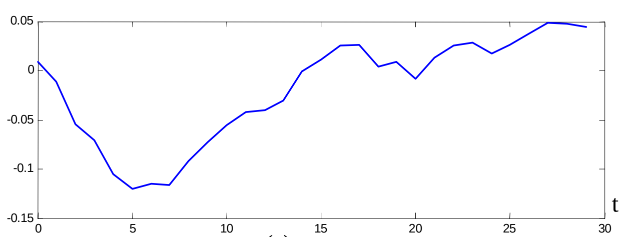
where β_t is a learning rate at iteration t.

We apply the theorem 1 in [5] so that the optimal learning rate β_t can be obtained. Therefore, we can guarantee that the system is stable and converges faster. In order to find the optimal centers c_i, we apply the genetic algorithm [6]-[9]. By redefining population in [5] as $Pop = \{c_i\}$, $c_i \in [c_{min}, c_{max}]$, we mostly follow the algorithm II proposed in [5].

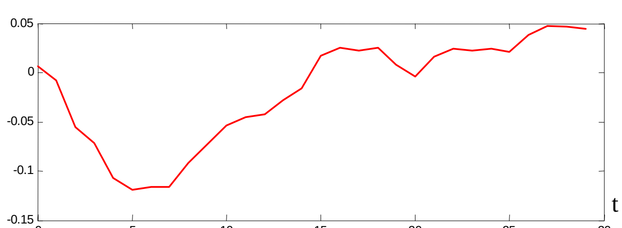
3. Simulation Example

For simulation, we get a segment signal data from Matlab file "mtlb" to do the filtering processing. The parameters are given as the number of hidden layer M=15, the length of input vector N=6, population size *Pop_size*=30, *Max_gen*=40, *Iteration*=400, c_{min}= -2, c_{max}=0.7, fixed σ_i= 0.5, *Threshold* =0.01.

After 400 iterations, we show the network output and the desired signal in Fig. 2. The trajectory of total error J is showed in Fig. 3. We notice that the value of J is always decreased and not much changed after 300th iteration. The norm trajectory of weighting factor vector W is showed in Fig. 4.



(a)



(b)

Fig. 2. (a) The output of the network, (b) the desired signal.

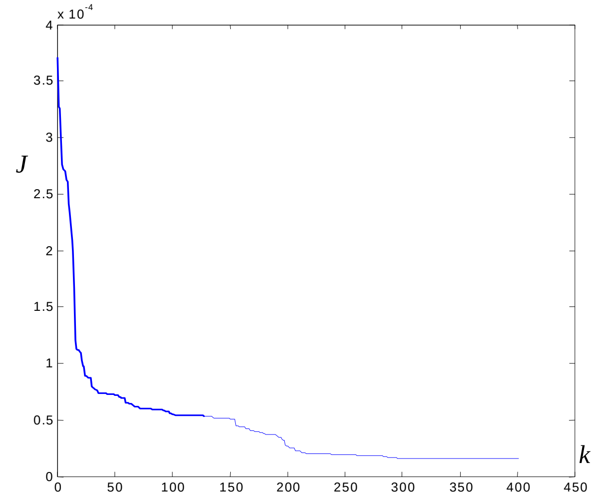


Fig. 3. The trajectory of the total error J

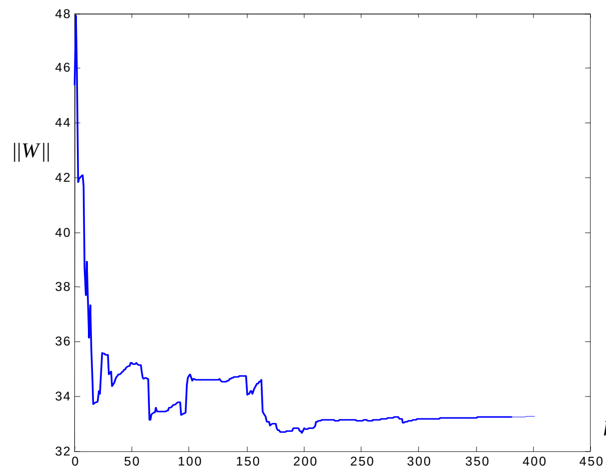


Fig. 4 The norm trajectory of weighting factor vector W.

4 Conclusion

In this paper, we constructed a RBF neural network and applied dynamic optimal learning approach to obtain optimal learning rates such that the proposed technique can train the network efficiently. The optimal center vector for hidden layer is search by genetic algorithm. The simulation example has shown that our approach is applicable. Not only the total error runs to convergent but the norm trajectories of weighting factor vector can also converge.

References:

[1] Moody, J. and C. Darken, "Learning with localized receptive fields,", *in Proc. 1988 Connectionist Models Summer School*, Carnegie Mellon University, 1988.

[2] C. H. Wang, W. Y. Wang, T. T. Lee, and P. S. Tseng, "Fuzzy B-spline membership function (BMF) and its applications in fuzzy-neural control," *IEEE Trans. Syst. Man Cyber.*, vol. 25, no. 5, pp. 841-851, May 1995.

[3] B. Kosko, *Neural Network and Fuzzy Systems*, Prentice Hall, Eaglewood Cliffs, NJ. 1992.

[4] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, Prentice Hall, Eaglewood Cliffs, NJ. 1994.

[5] C. H. Wang, H. L. Liu and C. T. Lin, "Dynamic Optimal Learning Rates of a Certain Class of Fuzzy Neural Networks and its Applications with Genetic Algorithm", *IEEE Trans. Syst. Man Cyber-B.*, Vol. 31, No. 3, pp. 467-475 ,June. 2001.

[6] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York, 3rd edition, 1996.

[7] K. S. Tang, K. F. Man, and D. W. Gu, "Structured genetic algorithm for robust H^{∞} control systems design," *IEEE Trans. Industrial Electronics* , Vol. 43, No. 5, pp. 575-582, October 1996.

[8] Chen-Chien Hsu, Kai-Ming Tse and Chi-Hsu Wang, "Digital redesign of continuous systems with improved suitability using genetic algorithms," *IEE Electronics Letters*, Vol. 33, No. 15, pp. 1345-1347 , July 1997.

[9] T. L. Seng, M. B. Khalid, and R. Yusof, "Tuning of a Neural-Fuzzy Controller by Genetic Algorithm", *IEEE Trans. on Syst. Man Cyber.*, Vol. 29, No. 2, pp. 226-236, Apr. 1999.