Improved Clustering and Anisotropic Gradient Descent Algorithm for Compact RBF Network*

Delu Zeng, Shengli Xie, and Zhiheng Zhou

College of Electronic & Information Engineering, South China University of Technology
510641, Guangzhou, China
Donald_zeng@163.com

Abstract. In the formulation of radial basis function (RBF) network, there are three factors mainly considered, i.e., centers, widths, and weights, which significantly affect the performance of the network. Within thus three factors, the placement of centers is proved theoretically and practically to be critical. In order to obtain a compact network, this paper presents an improved clustering (IC) scheme to obtain the location of the centers. What is more, since the location of the corresponding widths does affect the performance of the networks, a learning algorithms referred to as anisotropic gradient descent (AGD) method for designing the widths is presented as well. In the context of this paper, the conventional gradient descent method for learning the weights of the networks is combined with that of the widths to form an array of couple recursive equations. The implementation of the proposed algorithm shows that it is as efficient and practical as GGAP-RBF.

1 Introduction

Radial Basis Function (RBF) networks, due to their simple topological structures while retaining outstanding ability of approximation, are being used widely in function approximation, pattern recognition, and time series prediction.

Generally there are several crucial factors which seriously affect the performance of the RBF networks, i.e., the number of the hidden neurons, the center and the width for each neuron, and the weights. The original RBF networks [1] require that there be as many neurons as the observations (inputs).Thus they bring on high computational cost particularly for the case of bulky observations. In order to reduce the number of the hidden neurons, some compact RBF networks have been proposed [2]-[5].

However, among the above mentioned factors, the choice of the centers has the most critical effect on the performance of the network and plenty of study has been done on the choice of centers [2]-[7].

The algorithms proposed by S.Chen [2], [5] obtain a more compact network by orthogonal least square (OLS) method. The scheme has educed the contribution to the output variance from each neuron.

* The work is supported by the National Natural Science Foundation of China for Excellent Youth (Grant 60325310), the Guangdong Province Science Foundation for Program of Research Team (Grant 04205783), the Specialized Prophasic Basic Research Projects of Ministry of Science and Technology, China (Grant 2005CCA04100).

I. King et al. (Eds.): ICONIP 2006, Part II, LNCS 4233, pp. 806 – 813, 2006.

Integrated with forward subset selection and 0^{th}-order regularization, Orr [3] presented a regularized forward selection (RFS) algorithm for RBF networks. The algorithm used only one preset parameter, the basis function width.

To further study on the influence on the error by the locations of centers, Panchapakesan [6] proposed a new result on the bounds for the gradient and Hessian of the error considered as a function the centers, the widths, and the weights, for justification of moving the centers.

G. Huang [4] proposed a generalized growing and pruning RBF (GGAP-RBF) network. In their paper, a definition of "significance" is provided to judge the significance of the hidden neurons. Using this definition to grow or prune the hidden neurons one can establish a parsimonious RBF network with most significant ones. It functions well and it seems the GGAP-RBF is a great theoretical breakthrough on establishing a compact RBF network.

This paper looks into the problem of learning the centers with an improved clustering (IC) scheme and the widths with an anisotropic gradient descent (AGD) method.

2 RBF Network

The validity of RBF network is guaranteed by the theory of Reproducing Kernel Hilbert Space (RKHS), in which the dot product is computed by the kernels. In the field of RBF network one use series of RBFs to play the role of kernels as in RKHS.

Let $X = \{x_1, x_2, \cdots, x_N\}$, where $x_i = (x_{i1}, x_{i2}, \cdots x_{il}) \in R^l$, be the observations, and $Y = \{y_1, y_2, \cdots, y_N\}$ be the corresponding desired outputs. Without loss of generality, the Gaussian $g(\cdot) = \exp(-\frac{\|\cdot\|^2}{2\sigma^2})$ is usually chosen to take the role of RBF.

Then output of the system is:

$$f(X, C, \sigma, W) = \sum_j w_j \exp(-\frac{\| x_i - c_j \|^2}{2\sigma_j^2}), \quad (1)$$

where $C = \{c_1, c_2, \cdots, c_M\}$ and $\sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_M\}$ are centers and widths of the hidden neurons, respectively, $\|\cdot\|$ is the Euclidean norm, and $W = (w_1, w_2 \cdots, w_M)$ are the weights connecting the hidden neurons with the output.

3 Proposed IC-AGD Algorithm

Besides learning the centers, we study how to design the corresponding widths as well since they do affect the performance of the RBF networks either and can not be neglected.

In the section, an improved clustering (IC) scheme for locating the centers is described first, followed by the anisotropic gradient descent (AGD) method to decide the relative widths.

3.1 An Improved Clustering Scheme

Given a set of distinct observations $\{x_i\}_{i=1}^N$, where $x_i \in R^l \quad (i = 1, \cdots N)$, we need to cluster the observations into some categories without any a priori information of the number of category. A proposed scheme for efficiently clustering the observations is formulated as follows:

A. Compute the mean position point $\bar{x}$ for all the observations as follows:

$$\bar{x} = \frac{1}{N} \sum_{i}^{N} x_i \,. \tag{2}$$

B. Compute the distance d_i between each of the observations $\{x_i\}_{i=1}^N$ and the mean position point $\bar{x}$, i.e., $d_i^2 = \| x_i - \bar{x} \|^2$. Then rank $\{d_i\}_{i=1}^N$ from small to large by quit sort scheme. Without loss of generality, they are still denoted by $d_1 \le d_2 \le \cdots \le d_N$.

C. Define

$$d_{\min} \triangleq \min\{d_i, i = 1, \cdots, N\} = d_1, \tag{3}$$

and

$$d_{\max} \triangleq 2\max\{d_i, i = 1, \cdots, N\} = 2d_N \,. \tag{4}$$

Let

$$d \triangleq \lambda \cdot d_{\min} + (1-\lambda) \cdot d_{\max} = \lambda \cdot d_1 + 2(1-\lambda) \cdot d_N \,, \tag{5}$$

where $\lambda \in [0,1]$ is a preset parameter.

D. Set a relationship matrix $\{r_{i,j}\}_{i,j=1}^N$, which intends to indicate the cluster membership of each point from the observations, namely a Cluster Indication Matrix (CIM). And initialize the CIM as a zero matrix.

E. In order not to calculate all of the distances between every two points, we firstly intend to find a coarse CIM for the observations.

For each observation $x_i \in X = \{x_i\}_{i=1}^N$, do the following steps from $i = 1$ to N:

a. Denote $O(x_i)$ the neighborhood centered at x_i and with radius d.

b. Compute the distance between x_i and x_j, if x_j satisfies that $r_{ij} = 0$ and $x_j \in X_1 = \{x_j \mid \ | d_j - d_i | < d \quad and \quad i \ne j\} \subset X$ (Fig.1). The way that we

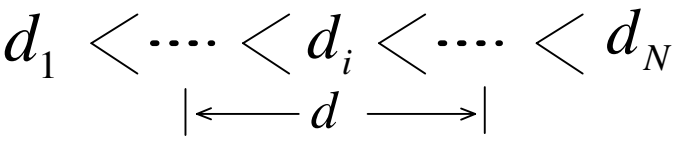

Fig. 1. Search the points in every neighborhood

search the points within the subset X_1 of X would greatly lower the computation complexity.

c. Then judge whether the observation x_j is in $O(x_i)$, and let

$$r_{ij} = r_{ji} = \begin{cases} |x_i - x_j|, & if \quad x_j \in O(x_i) \\ -1 & , else \end{cases}, \tag{6}$$

where x_j is a suspicious neighborhood point of x_i when $r_{ij} = -1$.

F. For the partition obtained from step **E** will end up with intersections between some coarse clusters, It is necessary to refine the CIM to make sure the **rule** is guaranteed, which i and k should belong to the same cluster, if i and j, j and k belong to the same cluster respectively. Thus the suspicion for membership of certain points is eliminated .To tackle this, we refine the CIM in the following way:

$$\text{If} \begin{cases} r_{ij} > 0 \\ r_{jk} > 0 \\ r_{ik} = 0 \end{cases} \text{ or } \begin{cases} r_{ij} > 0 \\ r_{jk} > 0 \\ r_{ik} = -1 \end{cases} \text{, then set } r_{ki} = r_{ik} = 1. \tag{7}$$

From the refined CIM, we know the observations $\{x_i\}_{i=1}^N$ can be partition into M clusters, namely $\{Q_1, Q_2, \cdots, Q_M\}$ and it is obvious that the lower bound for the distance of any two clusters is d.

3.2 Center Locating and Coarse Width Setting

Let

$$c_i = \frac{1}{|Q_i|}\sum_{j=1}^{|Q_i|} x_j \text{, and } \sigma_i^2 = \frac{1}{|Q_i|}\sum_{j=1}^{|Q_i|} \| x_j - c_i \|^2 \tag{8}$$

where $x_j \in Q_i$ and $i = 1, \cdots, M$.

3.3 Widths and Weights Learning

Many papers have focused on locating the centers as well as the weights, while it is noted that the designing of the widths is still important. In this part, we formulate an

anisotropic gradient descent (AGD) method. According to the distribution of the observations in R^l, the change of the observations mainly processed by those RBFs may vary between directions. Our AGD method is to update the widths of the kernels with a width scaling factor for the above coarse widths.

Let us define the refined widths of the RBF:

$$\sigma_*^2 = (s_1\sigma_1^2, s_2\sigma_2^2, \cdots, s_M\sigma_M^2), \tag{9}$$

where $S = (s_1, s_2, \cdots, s_M)$ is a scaling vector and each of its components is positive.

Then the mean square error (MSE) of the system is switched to a function of S and W, i.e.:

$$E(\sigma_*, W) = \frac{1}{N}\sum_{i=1}^{N} \| Y - f(X, C, \sigma_*, W) \|^2$$

$$\Rightarrow \quad E(S, W) = \frac{1}{N}\sum_{i=1}^{N} (y_i - f(X, C, S, W))^2 \ . \tag{10}$$

Replace f with Gaussian radial function, and differentiate the equation (10) with respect to w_j and s_j, $j = 1, \cdots M$, respectively, then we have:

$$\frac{\partial E(S,W)}{\partial w_j} = 2\frac{\partial \gamma(S,W)}{\partial w_j}\gamma^T(S,W) = -2G(X, s_j)\gamma^T(S,W), \tag{11}$$

$$\frac{\partial E(S,W)}{\partial s_j} = 2\frac{\partial \gamma(S,W)}{\partial s_j}\gamma^T(S,W) = -2w_j\frac{\partial G(X, s_j)}{\partial s_j}\gamma^T(S,W), \tag{12}$$

where $\gamma(S,W) = (\gamma_i(S,W))_{i=1}^N$, $\gamma_i(S,W) = y_i - \sum_{j=1}^{M} w_j g(x_i, s_j)$, and $G(X, s_j) = (g(x_i, s_j))_{i=1}^N$, $g(x_i, s_j) = \exp(-\frac{\| x_i - c_j \|^2}{2 s_j \sigma_j})$, $i = 1, \cdots, N$.

Apply the gradient descent for $E(S,W)$ with respect to S and W respectively, we obtain the coupled recursive equations for S and W as follows:

$$\begin{cases} w_j(n+1) = w_j(n) + 2\eta_1 G(X, s_j(n))\gamma^T(S(n), W(n)) \\ s_j(n+1) = s_j(n) + 2\eta_2 w_j(n)\dfrac{\partial G(X, s_j(n))}{\partial s_j}\gamma^T(S(n), W(n)) \end{cases}, \tag{13}$$

where η_1 , η_2 are two different learning steps with non-negative value and $j = 1, \cdots M$.

4 Implementation

In this section, an implementation of our algorithm is applied to a time series problems in function approximation area.

The chaotic Markey-Glass time series prediction is a well known benchmark prediction problem which is generated from the following delay differential equation:

$$\begin{cases} \dfrac{du(t)}{dt} = -0.1u(t) + \dfrac{0.2u(t-17)}{1+u^{10}(t-17)} \\ u(t-17) = 0.3 \end{cases} . \tag{14}$$

Table 1. Performance Comparisons between IC-AGD and GGAP-RBF

Algorithms	CPU Time (s)	Training RMSE	Testing RMSE	Number of neurons
GGAP(2-norm)	9.432	0.031342	0.058692	10
IC-AGD	8.198	0.011465	0.062481	9

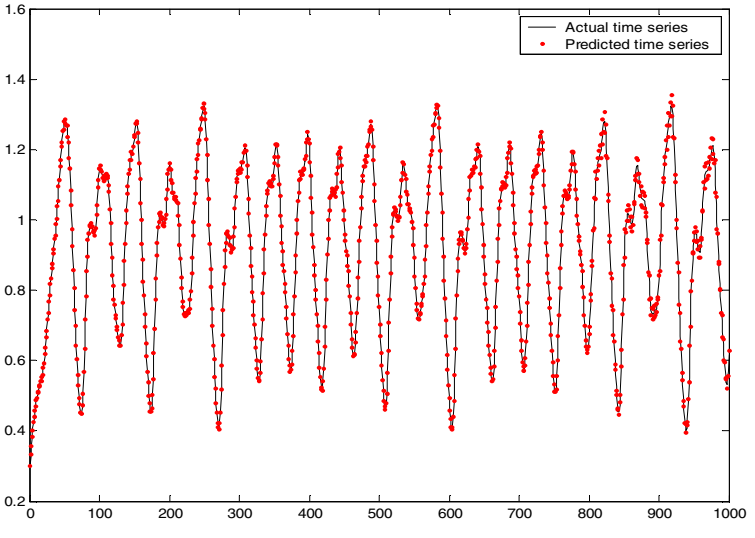

Fig. 2. The predicted with the IC-AGD and actual time series

Resample $N = 1000$ points from the above equation according to 1 sample period. With 20 sample steps ahead, we aim to predict the value of $u(t+20)$ by the values $\{u(1), u(2), \cdots, u(t)\}$. Let $\{x_i = (u_{i-20}, u_{i-20-6}, u_{i-20-12}, u_{i-20-18})^T\}_{i=1}^{1000}$ be the multiple inputs for the RBF network, while $\{y_i\}_{i=1}^{1000}$ be the corresponding outputs, where the first 800 pairs are for training, and the last 200 pairs are for testing. We have the following results:

Table 1 gives an comparison of the performance between the proposed algorithm and GGAP-RBF algorithm[4], where the parameters of GGAP is $e_{\min} = 0.01$, $\kappa = 0.85$, $\varepsilon_{\max} = 0.7$, $\varepsilon_{\min} = 0.07$, and the parameter in the proposed algorithms is $\lambda = 0.5$. Fig.2 shows the performance for the approximation ability of the proposed algorithm for the above time series prediction problems, where the red points denote the predicted time series, and the black curve denotes the actual time series.

The simulations show that the proposed algorithm performs as efficient as the GGAP-RBF, and will be practical in function approximation area either.

5 Conclusions

In this paper, an efficient algorithm IC-AGD for establishing a compact RBF network is presented. This algorithm consists of an improved clustering (IC) scheme for placing the centers, and an anisotropic gradient descent (AGD) method for learning the widths combined with learning the weights by conventional gradient descent. In the IC scheme, we take advantage of the relation between the observations and the mean position point to lower the computational complexity. And in the AGD method, we quote a scaling vector for the widths since the clusters may vary between directions.

Note that in the process of IC scheme for locating centers, the value of λ for threshold d will be much more favorable when it is optimized from maximizing the distance between clusters.

In conclusion, implementation shows that the proposed IC-AGD algorithm is as efficient and practical as the newly presented GGAP-RBF [4]. What is more, the proposed IC scheme can be utilized for other clustering problem.

References

1. Broomhead, D.S., Lowe, D.: Multivariable functional interpolation and adaptive networks. Complex Syst.,vol.2, (1988) 321-255
2. Chen, S., Cowan, C. F. N., Grant, P. M.: Orthogonal least squares learning algorithm for radial basis function networks. IEEE Trans. Neural Netw., vol. 2, no. 2, (1991) 302–309
3. Orr, M. J. L.: Regularization on the selection of radial basis function centers. Neural Computat., vol. 7, (1995) 606–623
4. Huang, G. B., Saratchandran, P., Sundararajan, N.: A Generalized Growing and Pruning RBF(GGAP-RBF) Neural Network for Function Approximation. IEEE Trans. Neural network, vol.16, no. 1, (2005) 57-67

5. Chen, S., Chng, E. S., Alkadhimi, K.: Regularized orthogonal least squares algorithm for constructing radial basis function networks. Int. J. Control, vol. 64, no. 5, (1996) 829–837
6. Panchapakesan, C., Palaniswami, M., Manzie. C.: Effects of moving the centers in an RBF network. IEEE Trans. Neural Network, vol.13, No.6, (2002) 1299-1307.
7. Xie, S.L., He, Z.S., Gao, Y.: Adaptive Theory of Signal Processing. 1st ed. Chinese Science Press, Beijing (2006).