

**A DIVIDE-AND-CONQUER FAST IMPLEMENTATION OF
RADIAL BASIS FUNCTION NETWORKS WITH
APPLICATION TO TIME SERIES FORECASTING***

RONG-BO HUNAG

*Department of Mathematics
Zhong Shan University
Guangzhou, PRC
E-mail: hrongbo@163.net*

YIU-MING CHEUNG AND LAP-TAK LAW

*Department of Computer Science
Hong Kong Baptist University
Hong Kong, PRC
E-mail: ymc@comp.hkbu.edu.hk, tllaw@comp.hkbu.edu.hk*

From the dual structural radial basis function network (DSRBF) (Cheung and Xu 2001), this paper presents a new **divide-and-conquer** learning approach to radial basis function networks (DCRBF). The DCRBF network is a hybrid system consisting of several sub-RBF networks, each of which takes a sub-input space as its input. Since this system divides a high-dimensional modeling problem into several low-dimensional ones, it can considerably reduce the structural complexity of a RBF network, whereby the net's learning is much faster. We have experimentally shown its outstanding learning performance on forecasting two real time series as well as synthetic data in comparison with a conventional RBF one.

1. Introduction

Radial basis function (RBF) networks are one of the most popular models in neural network. In the literature, RBF nets have been intensively studied with a lot of applications, e.g. in data mining⁸, pattern recognition¹¹, and time series forecasting^{4,9}. In general, the structural complexity of a RBF network depends on the number of the hidden nodes which is further related to the input dimension. Often, the node number increases along with the

*The work described in this paper was fully supported by a Faculty Research Grant of Hong Kong Baptist University with Project Number: FRG/02-03/I-06.

increase of the net's input dimension. Hence, effective dimension reduction of the net's input space can considerably decrease the network structural complexity, whereby the network's learning converges faster. Traditionally, principle component analysis (PCA) is a prevalent statistical tools for input dimension reduction. The basic rule is to select first several principal components of the observations as the RBF inputs. Since the PCA technique only uses second-order statistics information, it renders the principal components de-correlated but not really independent. That is, some useful information in the non-principal components may be discarded as well during the dimension reduction process. Consequently, the performance of the RBF network may become worse after PCA preprocess⁵.

In the past decade, independent component analysis (ICA) has been widely studied in the fields of neural networks and signal processing. It uses high-order statistics to map the multivariate observations into new representations with their redundancy as reduced as possible. In the literature, it has been shown that ICA outperforms PCA in extracting the hidden feature information and structures from the observations^{1,2,6,14}. Actually, our recent paper⁵ has successfully applied ICA to reduce the input dimension of a RBF network without deteriorating the net's generalization ability. However, ICA generally does not assign a specific principle order to the extracted components. To our best knowledge, selecting first several principle independent components is still an open problem.

Recently, Kai Tokkola¹⁰ applied a nonlinear dimension-reducing transformation to map high dimensional space to low dimensional one, i.e., $T : \mathfrak{R}^D \rightarrow \mathfrak{R}^d, d < D$. In his work, he coupled a nonparametric density estimator with a mutual information criterion based on Renyi's entropy to learn discriminative dimension-reducing transforms. However, this technique has at least two drawbacks to reduce the RBF input dimension. One is that the nonlinear dimension-reducing transforms lies in computational difficulties since it involves density estimation, resulting in a large amount of computing cost. The other drawback is that it makes the RBF performance degraded as the output dimension increases.

In our recent paper³, a dual structural radial basis function network has been proposed to accomplish a recursive RBF by two sub-networks. In this dual system, the input is divided into two parts with each modelled by a sub-network. The preliminary studies have shown its success on recursive function estimation. In this paper, we further extend its concept and give out a **divide-and-conquer** approach to **radial basis function (DCRBF)** network. This DCRBF is a hybrid system consisting of several sub-RBF

networks, each of which takes a sub-input as its input. That is, such a system has decomposed the original large input space into a direct sum of sub-input spaces with the output being a linear combination of these sub-RBF networks' ones. We give out an algorithm to learn the combination coefficients as well as the parameters in each sub-network. We have experimentally shown its outstanding learning performance on forecasting two real time series as well as synthetic data in comparison with a conventional RBF network.

2. DCRBF Network

2.1. Architecture

The architecture of the DCRBF network is shown in Figure 1. We decompose a RBF network into q sub-networks denoted as $RBF_1, RBF_2, \dots, RBF_q$, respectively. Let k_r represent the number of radial basis functions in the r^{th} sub-network, denoted as RBF_r , where $r = 1, 2, \dots, q$. In the DCRBF, the input separator divides the input space into q sub-ones by direct sum decompositions. That is, the input of the RBF_r is

$$\mathbf{x}_t(r) = [x_t^{(i_1)}, x_t^{(i_2)}, \dots, x_t^{(i_{d_r})}] \in \mathbf{V}_r \quad (1)$$

where $\{i_1, i_2, \dots, i_{d_r}\} \subset \{1, 2, \dots, d\}$. \mathbf{V}_r is the r^{th} direct sum subspace of \mathbf{V} such that

$$\mathbf{V}_1 \oplus \mathbf{V}_2 \oplus \dots \oplus \mathbf{V}_q = \mathbf{V} \quad (2)$$

where \oplus means for any $\mathbf{v} \in \mathbf{V}$, there exists a unique $\mathbf{v}_i \in \mathbf{V}_i$ such that $\mathbf{v} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \dots, \mathbf{v}_q^T]^T$, and d_r is the dimension of subspace \mathbf{V}_r with

$$\sum_{r=1}^q d_r = d. \quad (3)$$

We let $\hat{\mathbf{y}}_t$ be the actual output of the DCRBF network with

$$\hat{\mathbf{y}}_t = \sum_{r=1}^q c_r \mathbf{z}_t(r), \quad (4)$$

where $\mathbf{z}_t(r)$ is the RBF_r 's output, and c_r is the linear combination coefficient.

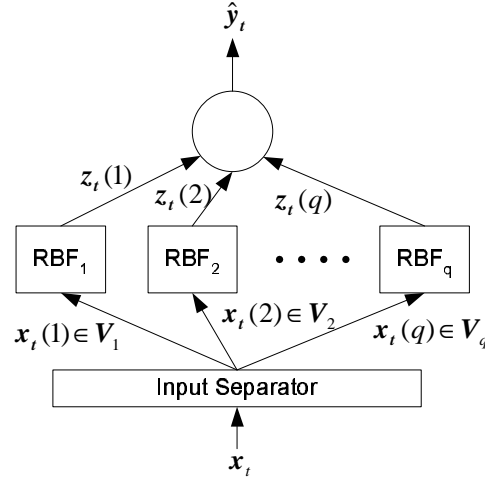


Figure 1. The DCRBF network model.

2.2. Learning Algorithm

Given the desired output \mathbf{y}_t at time step t , we calculate the output residual

$$\hat{\mathbf{e}}_t = \mathbf{y}_t - \hat{\mathbf{y}}_t. \quad (5)$$

Consequently, we can learn the combination coefficients c_r in Eq. (4) as well as the parameters of each RBF_r 's by minimizing the cost function

$$J(\Theta) = \frac{1}{N} \sum (\mathbf{y}_t - \hat{\mathbf{y}}_t)^T (\mathbf{y}_t - \hat{\mathbf{y}}_t) \quad (6)$$

where N is the number of inputs, $\Theta = \mathbf{C} \cup \Theta_1 \cup \Theta_2 \cup \dots \cup \Theta_q$ with $\mathbf{C} = \{c_1, c_2, \dots, c_q\}$, and Θ_r being the parameters of the RBF_r . In implementation, at each step time t , we adaptively tune Θ with a little small step along the descent direction of minimizing $(\mathbf{y}_t - \hat{\mathbf{y}}_t)^T (\mathbf{y}_t - \hat{\mathbf{y}}_t)$. That is, we adjust Θ by

$$c_r^{new} = c_r^{old} + \eta \hat{\mathbf{e}}_t^T \mathbf{z}_t(r), r = 1, 2, \dots, q \quad (7)$$

$$\Theta_r^{new} = \Theta_r^{old} - \eta \frac{\partial J(\Theta)}{\partial \Theta_r} \Big|_{\Theta_r^{old}} \quad (8)$$

where η is the learning rate.

The detailed steps in Eq. (8) depend on the implementation of each RBF_r , $r = 1, 2, \dots, q$. In general, each sub-RBF networks can be realized by a variety of RBF network models. In this paper, we adopt the Extended

Normalized RBF (ENRBF) network proposed in ¹². The general architecture of an ENRBF network is shown in Figure 2, which consists of a k -units hidden layer and an n -units output layer. The net's output is

$$\mathbf{z}_t = \sum_{j=1}^k (\mathbf{W}_j \mathbf{x}_t + \beta_j) O_j(\mathbf{x}_t) \quad (9)$$

where $\mathbf{z}_t = [z_t^{(1)}, z_t^{(2)}, \dots, z_t^{(n)}]^T$, $\mathbf{x}_t = [x_t^{(1)}, x_t^{(2)}, \dots, x_t^{(d)}]^T$ is an input, \mathbf{W}_j is an $n \times d$ matrix and β_j is an $n \times 1$ vector. $O_j(\mathbf{x}_t)$ is the output of unit j in the hidden layer with

$$O_j(\mathbf{x}_t) = \frac{\phi[(\mathbf{x}_t - \mathbf{m}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{x}_t - \mathbf{m}_j)]}{\sum_{i=1}^k \phi[(\mathbf{x}_t - \mathbf{m}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_t - \mathbf{m}_i)]} \quad (10)$$

where \mathbf{m}_j is the center vector, and $\boldsymbol{\Sigma}_j$ is the receptive field of the basis function $\phi(\cdot)$. In common, the Gaussian function $\phi(s) = \exp(-0.5s^2)$ is chosen. Consequently, Eq. (9) becomes

$$\mathbf{z}_t = \sum_{j=1}^k (\mathbf{W}_j \mathbf{x}_t + \beta_j) \frac{\exp[-0.5(\mathbf{x}_t - \mathbf{m}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{x}_t - \mathbf{m}_j)]}{\sum_{i=1}^k \exp[-0.5(\mathbf{x}_t - \mathbf{m}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_t - \mathbf{m}_i)]}. \quad (11)$$

In the above equation, two parameter sets should be learned. One is $\{\mathbf{m}_j, \boldsymbol{\Sigma}_j | j = 1, 2, \dots, k\}$ in the hidden layer, and the other is $\{\mathbf{W}_j, \beta_j | j = 1, 2, \dots, k\}$ in the output layer. In the paper ¹², these parameters learning has been connected with the mixture-of-experts model, whereby an expectation-maximization (EM) based single-step learning algorithm is proposed. Here, for simplicity, we prefer to learn the two parameter sets in the same way as the traditional approaches with the two separate steps:

Step 1: Learn $\{\mathbf{m}_j, \boldsymbol{\Sigma}_j | j = 1, 2, \dots, k\}$ in the hidden layer via a clustering algorithm such as k -means ⁷ or RPCL ¹³;

Step 2: Learn $\{\mathbf{W}_j, \beta_j | j = 1, 2, \dots, k\}$ in the output layer under the least mean square criteria. That is, we learn them as well as \mathbf{C} by minimizing Eq. (6). Consequently, the detailed implementations of Step 2 (i.e., Eq. (8)) are given as follows:

Step 2.1: Given \mathbf{x}_t and \mathbf{y}_t , we calculate $\hat{\mathbf{y}}_t$ by Eq. (4).

Step 2.2: We update

$$\mathbf{W}_j^{new}(r) = \mathbf{W}_j^{old}(r) + \eta \Delta \mathbf{W}_j(r) \quad (12)$$

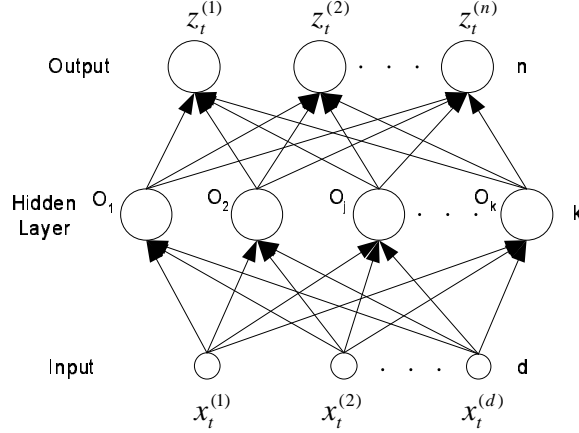


Figure 2. ENRBF network model.

$$\boldsymbol{\beta}_j^{new}(r) = \boldsymbol{\beta}_j^{old}(r) + \eta \Delta \boldsymbol{\beta}_j(r) \quad (13)$$

with

$$\Delta \mathbf{W}_j(r) = c_r^{old} O_j(\mathbf{x}_t(r)) \hat{\mathbf{e}}_t \mathbf{x}_t(r)^T \quad (14)$$

$$\Delta \boldsymbol{\beta}_j(r) = c_r^{old} O_j(\mathbf{x}_t(r)) \hat{\mathbf{e}}_t \quad (15)$$

where $\{\mathbf{W}_j(r), \boldsymbol{\beta}_j(r) | j = 1, 2, \dots, k_r, r = 1, 2, \dots, q\}$ is the parameter set of the RBF_r .

The iterations of Step 2.1 and 2.2 do not stop until the parameters converge.

3. Experimental Results

3.1. Experiment 1

We investigated the performance of the DCRBF network in time series forecasting. We generated 5,100 data points of a time series data with order 9 as follows:

$$\begin{aligned} u(t) = & 0.08u^2(t-1) - 0.33u(t-2) + \sin(u(t-3)) + 0.08u(t-4) \\ & + 0.2u(t-5) + 0.064u^2(t-6)u(t-7) - 0.6u(t-8)u(t-9) \end{aligned} \quad (16)$$

Let,

$$\begin{aligned} x_t = & [x_t^{(1)}, x_t^{(2)}, \dots, x_t^{(9)}] \\ = & [u(t-1), u(t-2), \dots, u(t-9)] \end{aligned}$$

be the input of the RBF network and $y_t = u(t)$ be the output. We let the first 5,000 data points be the training set, and the remaining 100 data points be the testing set. The input space of RBF network is decomposed into three subspaces with the input dimension $d_1 = 2, d_2 = 3, d_3 = 4$ respectively. Meanwhile, the RBF network is decomposed into three sub-networks. Let the size of hidden units in the conventional ENRBF network be $k = 6$ while each of the sub-network in the DCRBF network to be $k_1 = 2, k_2 = 2, k_3 = 2$ respectively. In the experiment, we fixed the learning rate $\eta = 0.0001$ and measured the net's performance under the MSE criterion. After repeatedly scanning the training data set 300 times, the performance of ENRBF and DCRBF under the MSE criterion is shown in Figure 3. We found that the DCRBF network converges much faster than the ENRBF network.

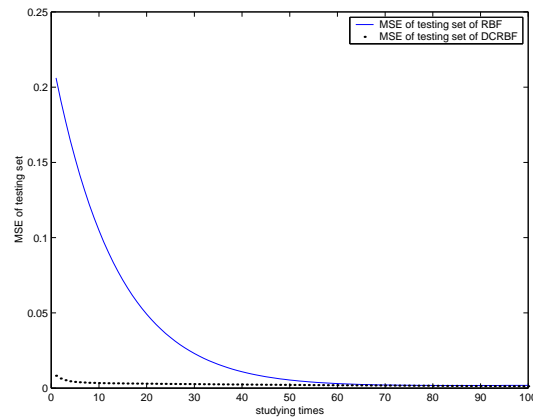


Figure 3. The comparison between the performance of ENRBF network and DCRBF network on the synthetic time-series data.

3.2. Experiment 2

We performed an experiment on the benchmark data getting from the famous Rob Hyndman's Time Series Data Library. We used the FOREX daily foreign exchange rates of 9 countries from 31st December, 1979 to 31st December, 1998 with size 4,774 data in this experiment. We let the first 4,674 data be the training set, and the remaining 100 data be the testing set. Also, we set the dimension of input space of ENRBF network at $d = 9$, which was further decomposed into three subspaces with the input

dimension $d_1 = 2, d_2 = 3, d_3 = 4$ respectively. We let the number of hidden units in the ENRBF network be $k = 8$, while the number of hidden units in the three sub-networks of the DCRBF be $k_1 = 2, k_2 = 3, k_3 = 3$ respectively. The experimental result is shown in Figure 4. It can be seen again that the DCRBF network converges much faster than the ENRBF with a slight improvement of net's generalization ability.

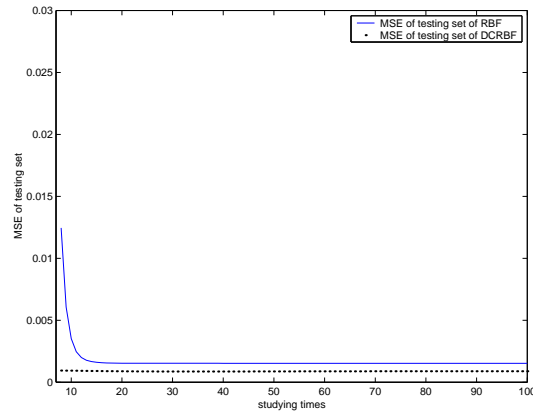


Figure 4. The comparison between the performance of RBF network and DCRBF network on FOREX daily foreign exchange data.

3.3. Experiment 3

We applied the DCRBF in the famous time series of annual average number of the sunspot from year 1700 to 1979 observed by Rudolph Wolf. We used the first 250 data to be the training set, and the remaining 30 to be the testing set. The number of hidden units of the ENRBF network was $k = 8$, while the hidden units of the three sub-networks in the DCRBF were $k_1 = 2, k_2 = 3, k_3 = 3$. We let the input dimension of the ENRBF be $d = 9$, and the input dimension of three decomposed sub-network in DCRBF be $d_1 = 3, d_2 = 3, d_3 = 3$. The experimental results are shown in Figure 5. Once again, we found that the DCRBF converges much faster than the ENRBF with a slight better generalization ability.

4. Concluding Remarks

We have presented a divide-and-conquer learning approach for RBF network (DCRBF), which is a hybrid system consisting of several sub-RBF

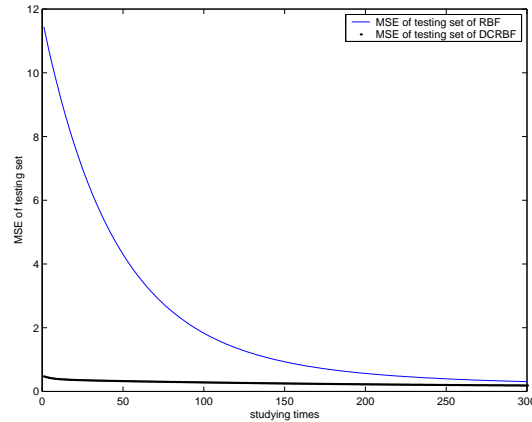


Figure 5. The comparison between the performance of the ENRBF and the DCRBF networks on sunspot data.

networks. Each sub-RBF network takes a sub-input spaces as its own input. The whole DCRBF network output is a combination of sub-RBF networks' outputs. Since this system divides a high-dimensional modelling problem into several low-dimensional ones, its structural complexity is generally simpler than a conventional RBF network. The experiments have shown that the proposed approach gives a slight better generalization ability with a much faster learning speed. In this paper, we just decompose the input space into sub-input spaces heuristically rather than having a general rule to follow. It is therefore expected that a more appropriate decomposition method exists to give a better net's performance.

References

1. A. D. Back, A. S. Weigend, "A First Application of Independent Component Analysis to Extracting Structure from Stock Returns," *International Journal of Neural System*, Vol. 8(4), pp. 473–484, 1997.
2. M. S. Bartlett, H. M. Lades, T. J. Sejnowski, "Independent Component Representations for Face Recognition," *Proceedings of the SPIE Symposium on Electronic Imaging: Science and Technology; Conference on Human Vision and Electronic Imaging III*, pp. 528–539, 1998.
3. Y. M. Cheung, L. Xu, "A Dual Structure Radial Basis Function Network for Recursive Function Estimation," *Proceedings of International Conference on Neural Information Processing (ICONIP'2001)*, Vol. 2, pp. 1903–1097, 2001.
4. N. Davey, S. P. Hunt, R. J. Frank, "Time Series Prediction and Neural Networks," *Journal of Intelligent and Robotic Systems*, Vol. 31, pp. 91–103, 2001.
5. R. B. Huang, L. T. Law, Y. M. Cheung, "An Experimental Study: On Reduc-

- ing RBF Input Dimension by ICA and PCA,” to be appeared in *Proceedings of 1st International Conference on Machine Learning and Cybernetics 2002 (ICMLC’02)*, Beijing, November 4-5, 2002.
6. G. J. Jang, T. W. Lee, Y. H. Oh, “Learning Statistically Efficient Features for Speaker Recognition,” *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Salt Lake City, Utah, May, 2001.
 7. J. MacQueen, “Some Methods for Classification and Analysis of Multivariate Observations,” *Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability*, Berkeley, University of California Press, Vol. 1, pp. 281–297, 1967.
 8. K. J. McGarry, S. Wermter, J. MacIntyre, “Knowledge Extraction from Radial Basis Function Networks and Multilayer Perceptrons,” *Proceeding of International Joint Conference on Neural Networks*, Vol. 4, pp. 2494–2497, 1999.
 9. A. Saranli, B. Baykal, “Chaotic time-series prediction and the relocating LMS (RLMS) algorithm for radial basis function networks,” *European Signal Processing Conference (EUSIPCO)*, Vol. 2, pp. 1247–1250, September 1996.
 10. K. Torkkola, W. M. Campbell, “Mutual Information in Learning Feature Transformations,” *Proceedings of International Conference on Machine Learning (ICML)*, Stanford, CA, June 29–July 2, 2000.
 11. B. Verma, “Handwritten Hindi Character Recognition using RBF and MLP Neural Networks,” *IEEE International Conference on Neural Networks (ICNN)*, Perth, pp. 86–92, 1995.
 12. L. Xu, “RBF Nets, Mixture Experts, and Bayesian Ying-Yang Learning,” *Neurocomputing*, Vol. 19, No. 1-3, pp. 223–257, 1998.
 13. L. Xu, “Rival Penalized Competitive Learning, Finite Mixture, and Multisets Clustering,” *Proceedings International Joint Conference on Neural Networks*, May 5-9, 1998, Anchorage, Alaska, Vol. II, pp. 2525–2530, 1998.
 14. A. Ziehe, G. Nolte, T. Sander, K. R. Muller, G. Curio, “A Comparison of ICA-based Artifact Redction Methods for MEG,” *12th International Conference on Biomagnetism*, Helsinki University of Technology, Finland, 2000.