An Advance on Divide-and-Conquer Based Radial Basis Function Networks

Yiu-ming Cheung[1] and Rong-bo Huang[2]

[1] Department of Computer Science
Hong Kong Baptist University, Hong Kong, P.R. China
ymc@comp.hkbu.edu.hk

[2] Department of Mathematics
Zhong Shan University, Guangzhou, P.R. China
hrongbo@163.net

Abstract. Our recent paper (Huang et al. 2002) has presented a **d**ivide-and-**c**onquer based radial basis function network (DCRBF) that is a hybrid system consisting of several sub-RBF networks, each of which individually takes a sub-input space as its input. Since this system reduces the structural complexity of a RBF network by mapping a high-dimensional modelling problem into several low-dimensional ones, the net's learning speed is considerably improved as a whole with the moderately enhanced generalization capability. However, the performance of DCRBF generally varies with the different input decompositions. Moreover, a DCRBF can be regarded as the decomposing implementation of a conventional RBF. Under the circumstances, the total number of hidden units is a constant. Hence, how to distribute the number of hidden units in each sub-RBF network becomes an important issue in optimizing the DCRBF performance. In this paper, we further explore the decomposition rules on both of inputs and the hidden-unit number. Consequently, an optimal decomposition is presented in a sense of learning speed with the experimental justifications.

1 Introduction

Due to simple architecture and learning, radial basis function (RBF) networks have been intensively studied with a lot of applications, e.g., in data mining [5], pattern recognition [7], and time series forecasting [2, 6]. In general, the structural complexity of a RBF network depends on the number of the hidden units which is further related to the input dimension. Often, the unit number increases along with the increase of the net's input dimension. Hence, effective dimension reduction of the net's input space can considerably reduce the network structural complexity, whereby the network's learning becomes faster with the comparable generalization capability. In the literature, either principal component analysis (PCA) or independent component analysis (ICA) provides a way for input dimension reduction. However, they also meet new difficulties. On the one hand, the PCA technique uses second-order statistics information only, resulting in 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 preprocess [3]. On the other hand, ICA makes the extracted components as independent as possible, but it generally does not assign a specific principle order to the components. To our best knowledge, selecting first several principle independent components is still an open problem.

Alternatively, our recent paper [4] presents a new divide-and-conquer based radial basis function network (DCRBF) that is a hybrid system consisting of several sub-RBF networks, each of which individually takes a sub-input space as its input. The output of DCRBF is a linear combination of the sub-networks' outputs with the linear coefficients learned together with each sub-network system parameters. Since DCRBF reduces the structural complexity of a RBF network by mapping a high-dimensional modelling problem into several low-dimensional ones, the net's learning speed has been considerably improved as a whole with the moderately enhanced generalization capability as shown in [4]. Actually, the DCRBF is a natural extension of our recently proposed dual structural RBF [1] that models a recursive function by using two sub-RBF networks.

However, the DCRBF modelling may naturally arise three questions:

1. How many sub-spaces should the input space be decomposed into?
2. Suppose the input space is decomposed into q sub-spaces. There are still $q!$ decomposition combinations. Which one should be chosen?
3. A DCRBF can be regarded as the decomposing implementation of a conventional RBF. Under the circumstances, the total number of hidden units is a constant. Then, how to distribute the number of hidden units in each sub-RBF network such that the DCRBF performance is optimized in a certain sense?

In this paper, we will concentrate on investigating the latter two problems only. Under a specific q sub-spaces, we have theoretically given out an optimal decomposition in a sense of learning speed, and further justified it by the experimental results.

2 Overview of DCRBF Network

As shown in Figure 1, the input separator of a DCRBF network decomposes the input space $\mathbf{V}$ of a RBF network into the direct sum of q sub-input spaces, written as $\mathbf{V}_r$, $r = 1, 2, \ldots, q$, respectively such that

$$\mathbf{V}_1 \oplus \mathbf{V}_2 \oplus \ldots \oplus \mathbf{V}_q = \mathbf{V} \tag{1}$$

where $\oplus$ means for any $\mathbf{v} \in \mathbf{V}$, there exists a unique $\mathbf{v}_i \in \mathbf{V}_i$ such that $\mathbf{v} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \ldots, \mathbf{v}_q^T]^T$. The DCRBF consists of q sub-RBF network, denoted as RBF_r, $r = 1, 2, \ldots, q$, respectively. Each RBF_r models the relationship between the current output and the input $\mathbf{x}_t(r) \in \mathbf{V}_r$ with

$$\mathbf{x}_t(r) = [x_t^{(i_1)}, x_t^{(i_2)}, \ldots, x_t^{(i_{d_r})}] \in \mathbf{V}_r \tag{2}$$

where $\{i_1, i_2, \ldots, i_{d_r}\} \subset \{1, 2, \ldots, d\}$, d_r and d are the dimensions of $\mathbf{V}_r$ and $\mathbf{V}$ respectively with $\sum_{r=1}^{q} d_r = d$. Further, $\hat{\mathbf{y}}_t$ is the actual output of the DCRBF network with

$$\hat{\mathbf{y}}_t = \sum_{r=1}^{q} c_r \mathbf{z}_t(r), \tag{3}$$

where $\mathbf{z_t}(r)$ is the RBF_r's output, and c_r is a coefficient of linear combination. At each time step t, given the desired output $\mathbf{y}_t$, we calculate the output residual

$$\hat{\mathbf{e}}_t = \mathbf{y}_t - \hat{\mathbf{y}}_t. \tag{4}$$

Consequently, we can learn the combination coefficients c_rs in Eq.(3) as well as the parameters of each RBF_r's by minimizing the cost function

$$J(\mathbf{\Theta}) = \frac{1}{N} \sum_{t=1}^{N} (\mathbf{y}_t - \hat{\mathbf{y}}_t)^T (\mathbf{y}_t - \hat{\mathbf{y}}_t) \tag{5}$$

where N is the number of inputs, $\boldsymbol{\Theta} = \mathbf{C} \bigcup \boldsymbol{\Theta}_1 \bigcup \boldsymbol{\Theta}_2 \bigcup \ldots \bigcup \boldsymbol{\Theta}_q$ with $\mathbf{C} = \{c_1, c_2, \ldots, c_q\}$, and $\boldsymbol{\Theta}_r$ being the parameters of the RBF_r. In implementation, at each step time t, we adaptively tune $\boldsymbol{\Theta}$ 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 $\boldsymbol{\Theta}$ by

$$c_r^{new} = c_r^{old} + \eta \hat{\mathbf{e}}_t^{\mathbf{T}} \mathbf{z}_t(r), r = 1, 2, \ldots, q \tag{6}$$

$$\boldsymbol{\Theta}_r^{new} = \boldsymbol{\Theta}_r^{old} - \eta \frac{\partial J(\mathbf{\Theta})}{\partial \boldsymbol{\Theta}_r} |_{\boldsymbol{\Theta}_r^{old}}, \tag{7}$$

where η is the learning rate.

The detailed steps of Eq. (7) depend on the explicit implementation of each RBF_r, $r = 1, 2, \ldots, q$. In our recent paper [4], we have presented an algorithm by using an Extended Normalized RBF (ENRBF) network to model each sub-RBF network. Please refer to the paper [4] for more details.

3 The Decomposition Rule for DCRBF

From Section 2, it can be seen that the performance of DCRBF varies with the different input decompositions. Supposing a DCRBF consists of q sub-RBF networks, we can determine an appropriate decomposition by the two separate steps as follows:

Step 1. Give an appropriate order of those d input variables in $\mathbf{V}$;

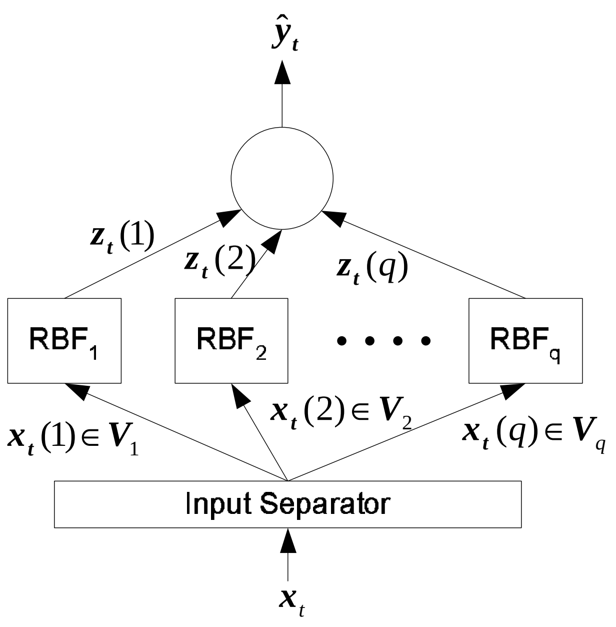

Fig. 1. The DCRBF network model.

Step 2. Separate these ordered d input variables into q groups, each of which spans an sub-input space accordingly.

In Step 1, we use PCA ordering, i.e., the inputs $x^{(1)}, x^{(2)}, \ldots, x^{(d)}$ in $\mathbf{V}$ is transformed into $x^{(i_1)}, x^{(i_2)}, \ldots, x^{(i_d)}$, where $i_1, i_2, \ldots, i_d$ is the PCA order. In Step 2, we uniformly decompose the ordered input variables into q groups. That is, the dimension of each sub-input space should be as equal as possible. In the following, we will present a theorem to show that such a uniform input separation leads to an optimal decomposition in a sense of training time cost under a specific input order. Moreover, when the total number of hidden units is fixed, the theorem tells us that the number of hidden units in each sub-RBF network should be uniformly distributed as well. Before presenting the theorem, we first give out some definitions:

Definition 1. *Let d_i, $i = 1, 2, \ldots, q$ be positive integers. A q-tuple $(d_1, d_2, \ldots, d_q)$ is called $q_{\text{decomposition}}$ of d if $d_1 + d_2 + \cdots + d_q = d$.*

Definition 2. *The total number k of hidden units in DCRBF is defined as the module of DCRBF, written as $\|DCRBF\| = k$, where k is the sum of those hidden radial function units in each sub-RBF network.*

Definition 3. *A product among the parameters and constants in the DCRBF is called a basic computation term(BCT).*

Definition 4. *If a $q_{\text{decomposition}}$ makes the training time cost of a DCRBF minimized, it is called* **t***ime* **o***ptimal* **d***ecomposition(TOD).*

The theorem is then given as follows:

Theorem 1.
1. *Let the dimension of input space* $\mathbf{V}$ *of RBF be* d *and* $(d_1, d_2, \ldots, d_q)$ *be a* $q_{\text{decomposition}}$ *of* d*. The RBF is therefore decomposed into the DCRBF consisting of* q *sub-networks with input dimension* $d_1, d_2, \ldots, d_q$ *respectively.*
2. $\|DCRBF\| = k$*. Let* $(k_1, k_2, \ldots, k_q)$ *be a* $q_{\text{decomposition}}$ *of* k*, where* k_r *is the number of hidden units in sub-network* RBF_r*;*
3. *The time cost of each BCT in the DCRBF to be equal.*

The $q_{\text{decomposition}}$ *of* $(d_1, d_2, \ldots, d_q)$ *and* $(k_1, k_2, \ldots, k_q)$ *is TOD if and only if*

$$d_1 = d_2 = \cdots = d_q = \frac{d}{q} \tag{8}$$

$$k_1 = k_2 = \cdots = k_q = \frac{k}{q}. \tag{9}$$

To prove Theorem 1, we first give out the lemma as follows:

Lemma 1. *Given the object function*

$$f(x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n) = \sum_{i=1}^{n} x_i y_i^2 \tag{10}$$

with the constraints:

$$\sum_{i=1}^{n} x_i = Q, \qquad \sum_{i=1}^{n} y_i = K, \qquad x_i > 0, \quad y_i > 0, i = 1, 2, \ldots, n, \tag{11}$$

the necessary and sufficient condition for a minimum point of $f(x_1, x_2, \cdots, x_n)$ *is*

$$x_1 = x_2 = \cdots = x_n = \frac{Q}{n}, \tag{12}$$

$$y_1 = y_2 = \cdots = y_n = \frac{K}{n}. \tag{13}$$

Proof. We construct the Lagrange function as follows:

$$\begin{aligned} L(x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n, \lambda_1, \lambda_2) = {} & \textstyle\sum_{i=1}^{n} x_i y_i^2 + \lambda_1 (Q - \sum_{i=1}^{n} x_i) \\ & + \lambda_2 (K - \textstyle\sum_{i=1}^{n} y_i). \end{aligned} \tag{14}$$

We let the partial derivatives of L with respect to $x_i, y_i, \lambda_1, \lambda_2, i = 1, 2, \ldots, n$ be zero, i.e.,

$$\frac{\partial L}{\partial x_i} = y_i^2 - \lambda_1 = 0, \tag{15}$$

$$\frac{\partial L}{\partial y_i} = 2x_i y_i - \lambda_2 = 0, \tag{16}$$

$$\frac{\partial L}{\partial \lambda_1} = (Q - \sum_{i=1}^{n} x_i) = 0, \tag{17}$$

$$\frac{\partial L}{\partial \lambda_2} = (K - \sum_{i=1}^{n} y_i) = 0. \tag{18}$$

From Eq.(15) and Eq.(16),we have

$$x_1 = x_2 = \cdots = x_n, \tag{19}$$

$$y_1 = y_2 = \cdots = y_n. \tag{20}$$

Substitute Eq. (19)(20) into Eq. (17)(18), Lemma 1 is therefore proved.

■

Now we present the proof of Theorem 1 as follows.

Proof. Let $(d_1, d_2, \ldots, d_q)$ be a $q_{\text{decomposition}}$ of d, T be the training time of DCRBF and $T_r, r = 1, 2, \ldots, q$ be the training time of RBF_r, $r = 1, 2, \ldots, q$ respectively. We have

$$T = \sum_{r=1}^{q} T_r. \tag{21}$$

Since T_r depends on the number of BCT only in the learning process, we therefore just need to consider the BCT in the following terms:

$$[\mathbf{x}(r) - \mathbf{m}_j(r)]^T \boldsymbol{\Sigma}_j^{-1} [\mathbf{x}(r) - \mathbf{m}_j(r)], \tag{22}$$

where $j = 1, 2, \ldots, d_r$, $r = 1, 2, \ldots, q$, and $\mathbf{x}(r) \in \mathbf{V}_r$ is a vector. It can be seen that the number of BCT in (22) is d_r^2. Without loss of generality, we suppose the computing time cost of each BCT is 1 time unit. Hence, T_r is a function of d_r and k_r, which can be further expressed as

$$T_r(d_r, k_r) = k_r d_r^2. \tag{23}$$

Putting Eq. (23) into Eq. (21), we then have

$$T(d_1, d_2, \cdots, d_q, k_1, k_2, \cdots, k_q) = \sum_{r=1}^{q} k_r d_r^2. \tag{24}$$

That is, we try to optimize the following constraint problem:

$$\text{minimize } T(d_1, d_2, \cdots, d_q, k_1, k_2, \cdots, k_q) = \sum_{r=1}^{q} k_r d_r^2, \tag{25}$$

$$\text{subject to } \sum_{r=1}^{q} d_r = d, \quad \sum_{r=1}^{q} k_r = k, \quad d_r > 0, \quad k_r > 0. \tag{26}$$

It can be seen that the solution of this problem is actually a special case of Lemma 1 when x_i and y_i with $i = 1, 2, \ldots, n$ both take integer values. Hence, Theorem 1 is held.

■

4 Experimental Simulations

4.1 Experiment 1 and 2

To justify the above theorem, we showed two experiments to compare the train time cost of DCRBF with the different decompositions. The experimental environment is given in Table 1. In Experiment 1, we used 8,100 data points generated from the nonlinear function:

$$\begin{aligned} y_t = {} & x_t^{(1)} \cos(x_t^{(2)}) + x_t^{(3)} \sin(x_t^{(4)}) - 0.4(x_t^{(5)})^2 \\ & + 0.5 x_t(6) x_t^{(7)} + 0.2(x_t^{(8)})^2 x_t^{(9)} + \varepsilon_t \end{aligned}$$

where $\mathbf{x_t} = [x_t^{(1)}, x_t^{(2)}, x_t^{(3)}, x_t^{(4)}, x_t^{(5)}, x_t^{(6)}, x_t^{(7)}, x_t^{(8)}, x_t^{(9)}]$ is the input of RBF, y_t is the desired output of RBF and ε_t is zero-mean Gaussian white noise with the variance being 0.001. We let $\|DCRBF\| = 9$ and decomposed the input space of the RBF into the direct sum of three sub-input spaces. The decomposition of input space and experimental results are shown in Table 2. It can be seen that the learning speed of uniform decomposition is the fastest in all cases we have tried so far.

Table 1. The experimental environment

CPU	Pen.III 650MHZ
Memory	256M
Operating System	Windows 2000
Running software	Matlab 5.3

In Experiment 2, we performed an experiment on the benchmark data acquired from the famous Rob Hyndman's Time Series Data Library. We used the FOREX daily foreign exchange rates of Australia to USA from December 31, 1979 to December 31, 1998 with the 4,774 data points. In the experiment, we let

Table 2. The training time cost of DCRBF with the different input decompositions

3_decomposition of d	3_decomposition of k	$\sum_{r=1}^{q} k_r d_r^2$	mean training time
(3,3,3)	**(3,3,3)**	**81**	**11.1723**
(3,3,3)	(2,2,5)	81	11.2679
(2,2,5)	(3,3,3)	99	11.6658
(2,3,4)	(2,2,5)	106	11.7269
(2,2,5)	(2,2,5)	141	12.0171

the input of RBF be $\mathbf{x}_t = [x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-5}, x_{t-6}, x_{t-7}, x_{t-8}, x_{t-9}]^T$, and $y_t = x_t$ be the output. Similar to Experiment 1, we also let $\|DCRBF\| = 9$, and decomposed the input space of RBF into three sub-input spaces. Table 3 shows the results under the different decompositions. Again, the DCRBF with the uniform decomposition needs the least training time cost.

Table 3. The training time cost of DCRBF with different decompositions

3_decomposition of d	3_decomposition of k	$\sum_{r=1}^{q} k_r d_r^2$	mean training time
(3,3,3)	**(3,3,3)**	**81**	**15.4161**
(3,3,3)	(2,2,5)	81	15.4914
(2,2,5)	(3,3,3)	99	15.7441
(2,3,4)	(2,2,5)	106	15.9068
(2,2,5)	(2,2,5)	141	16.2129

4.2 Experiment 3

In the above experiments, we have investigated the uniform decomposition on the learning speed of DCRBF without considering the net's generalization capability. In the following, we will further demonstrated the generalization capability of DCRBF when uniform decomposition and PCA input-variable ordering are used. We set $\mathbf{x}_t = [x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(11)}]$ to be the input of DCRBF, where $x_t^{(2)}, x_t^{(4)}, x_t^{(6)}, x_t^{(7)}, x_t^{(8)}, x_t^{(9)}$ were uniform, $x_t^{(1)}, x_t^{(3)}, x_t^{(11)}$ were Gaussian and $x_t^{(5)}, x_t^{(10)}$ were two Gaussian mixture. The desired outputs of the network were given by:

$$\begin{aligned} y_t = {} & x_t^{(1)} \cos x_t^{(2)} + x_t^{(3)} \sin x_t^{(4)} \cos x_t^{(10)} - (x_t^{(5)})^2 \sin x_t^{(12)} + 0.5 x_t^{(6)} x_t^{(7)} \\ & + 0.2 (x_t^{(8)})^2 x_t^{(9)} \sin x_t^{(11)} + e_t \end{aligned}$$

where e_t is white noise.

We generated $1,100$ data points. The first $1,000$ were the training data, and the remaining 100 data were the testing data. In the experiment, the PCA order of input is $x^{(11)}, x^{(10)}, x^{(9)}, x^{(8)}, x^{(7)}, x^{(6)}, x^{(3)}, x^{(1)}, x^{(2)}, x^{(4)}, x^{(5)}$. We decomposed the inputs into two parts: $\mathbf{x}(1) = x^{(11)}, x^{(10)}, x^{(9)}, x^{(8)}, x^{(7)}, x^{(6)}$ and $\mathbf{x}(2) = x^{(3)}, x^{(1)}, x^{(2)}, x^{(4)}, x^{(5)}$. Further, we fixed the learning rate be 0.001 and measured the net's performance under the criterion of mean-square error (MSE). During the net's learn process, the MSE curve on the testing set is shown in Figure 2. After scanning training set data 200 times, a snapshot of the MSE value on the testing set is 0.32 as shown in the Trial 1 of Table 4. In contrast, we also investigated another input uniform decomposition in Trial 2 without considering PCA ordering. Table 4 shows that the performance of DCRBF deteriorates a little bit in comparison with Trial 1. Further, in Trial 3 and Trial 4, we tested two other different input decompositions without uniform decomposition and PCA ordering. It can be seen that the performance of DCRBF has been further moderately deteriorated, but they were better than the DCRBF without input decomposition (i.e., a DCRBF has degenerated to a conventional RBF) as shown in Trial 5 of Table 4. Actually, Figure 2 has shown that the DCRBF involving uniform decomposition and PCA ordering learns much faster than the conventional RBF with a moderately improved generalization capability.

Table 4. The MSE of DCRBF with different decompositions on testing set

Trials	input space	k	MSE
1	$(\mathbf{x_3}, \mathbf{x_1}, \mathbf{x_2}, \mathbf{x_4}, \mathbf{x_5}) \bigoplus (\mathbf{x_{11}}, \mathbf{x_{10}}, \mathbf{x_9}, \mathbf{x_8}, \mathbf{x_7}, \mathbf{x_6})$	**(5,6)**	**0.3200**
2	$(x_3, x_4, x_5, x_{10}, x_{11}) \bigoplus (x_1, x_2, x_6, x_7, x_8, x_9)$	(5,6)	0.3235
3	$(x_1, x_2, x_{11}) \bigoplus (x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$	(3,8)	0.3451
4	$(x_4, x_5) \bigoplus (x_1, x_2, x_3, x_6, x_7, x_8, x_9, x_{10}, x_{11})$	(2,9)	0.3534
5	$(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11})$	11	0.4322

5 Conclusion

We have further studied the decompositions in the DCRBF network. Not only is a PCA input-variable ordering suggested, but also a uniform decomposition is presented on both of inputs and the hidden-unit number, which is optimal in a sense of learning speed. The experiments have shown that the DCRBF involving PCA ordering and uniform decomposition learns much faster than the other decomposition as well as the conventional RBF one with the generalization capability moderately improved.

Acknowledgment

This work was supported by a Faculty Research Grant of Hong Kong Baptist University with the project code: FRG/02-03/II-40.

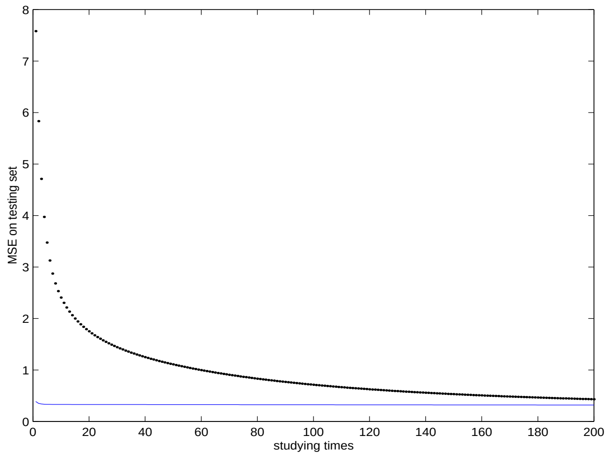

Fig. 2. The MSE curve of DCRBF and a conventional RBF on the testing set in Experiment 3, where the solid line is from the DCRBF, and the dashed line is from the conventional one.

References

1. Y.M. Cheung and L. Xu, "A Dual Structural Radial Basis Function Network for Recursive Function Estimation", *Proceedings of International Conference on Neural Information Processing* (ICONIP'2001), Vol. 2, pp. 1903–1097, 2001.
2. N. Davey, S.P. Hunt and R.J. Frank, "Time Series Prediction and Neural Networks", *Journal of Intelligent and Robotic Systems*, Vol. 31, pp. 91–103, 2001.
3. R.B. Huang, L.T. Law and Y.M. Cheung, "An Experimental Study: On Reducing RBF Input Dimension by ICA and PCA", *Proceedings of* 1^{st} *International Conference on Machine Learning and Cybernetics 2002* (ICMLC'02), Vol. 4, pp. 1941–1946, 2002.
4. R.B. Huang, Y.M. Cheung and L.T. Law, "A Divide-and-Conquer Fast Implementation of Radial Basis Function Networks with Application to Time Series Forecasting", *Advances in Data Mining and Modeling*, pp. 97–106, Hong Kong, June 27-28, 2002.
5. K.J. McGarry, S. Wermter and 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.
6. A. Saranli and 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, 1996.
7. B. Verma, "Handwritten Hindi Character Recognition using RBF and MLP Neural Networks", *IEEE International Conference on Neural Networks* (ICNN), Perth, pp. 86–92, 1995.