A Dual Structural Radial Basis Function Network for Recursive Function Estimation

Yiu-ming Cheung
Email:ymc@comp.hkbu.edu.hk

Department of Computer Science, Hong Kong Baptist University, Hong Kong, PRC

ABSTRACT
We present a dual structural radial basis function (RBF) network for recursive function estimation. This network is a hybrid system consisting of two sub-RBF networks. One sub-network models the relationship between the current network output and the past ones, and the other one describes the relationship between the current network output and the inputs. We propose a new variant of extended normalized RBF (ENRBF) network to implement each sub-RBF net. This variant uses an adjustable $p$-order single-term polynomial to fit the relations between each hidden unit and each output unit. It not only includes the existing ENRBF net as its special case, but also has better fitting ability in general under the moderate number of hidden units. The experiments have shown the proposed net’s performance.

1. Introduction
Due to less complexity and simple learning, radial basis function (RBF) network has been extensively studied with a lot of applications and many theoretical results [2, 3, 4, 5, 8].

Typically, the existing approaches [2, 3, 7, 12] assume that the outputs of a RBF network is the function of the inputs only. However, in some practical problems such as nonlinear adaptive noise cancelation problem [1] and the representation of finite state automata [9], the net’s output depends on the past ones as well as the inputs. Under the circumstances, these existing techniques cannot work well. In the literature, one way is to use the existing techniques to train the RBF network by taking the net’s input and the past outputs as an augmented input [1]. However, this method requires that the scale of network’s output is the same as the inputs. Otherwise, it can result in poor clustering results in the hidden layer, whereby the net’s performance deteriorates.

In this paper, we present a dual structural radial basis function (RBF) network, which is a hybrid system consisting of two sub-RBF networks. One sub-network models the relationship between the current network’s output and the past ones, and the other one describes the relationship between the current output value and the inputs. Since this network clusters the network’s inputs and the past outputs separately rather than regarding them as an augmented inputs, it can therefore work well even if the scale of network’s outputs is different from the inputs. We implement each sub-RBF network by proposing a new ENRBF variant, in which we use an adjustable $p$-order single-term polynomial to fit the relations between each hidden unit and each output unit. Such a polynomial regression extends the fitting ability of the fixed first-order one used in most existing ENRBF networks [10, 11], it is therefore generally expected that this new variant has better performance in function approximation under the moderate number of hidden units. Successful experimental results have shown the performance of the dual structural RBF network in recursive function estimation.

2. Problem
Given a set of training data set $\{x_t, y_t\}_{t=1}^N$, where $x_t = [x_{t,1}, x_{t,2}, \ldots, x_{t,d}]^T$ and $y_t \in \mathbb{R}^n$ are the network’s input and desired output respectively, we describe the relations between $y_t$’s and $x_t$’s by the following recursive function:

$$y_t = F(Y_{t-1}, x_t) + e_t$$

with $Y_{t-1} = [y_{t-1}^T, y_{t-2}^T, \ldots, y_{t-q}^T]^T$, where $F(.)$ is an unknown deterministic nonlinear function and $e_t$ is white noise. In this paper, we suppose $F(Y_{t-1}, x_t)$ to be linear separable. That is, $F(Y_{t-1}, x_t)$ can be further decomposed into other two functions $f(.)$ and $g(.)$ with

$$F(Y_{t-1}, x_t) = \gamma_1 f(Y_{t-1}) + \gamma_2 g(x_t),$$

where $\gamma_1$ and $\gamma_2$ are two unknown constants. The task of a proposed RBF network is to approximate this function.
through the given training data set with the network’s
generality as good as possible under a certain measure-
ment.

3. A New Variant of ENRBF Network

The general architecture of an ENRBF network is shown
in Figure 1, which consists of a \( k \)-unit hidden layer
and an \( n \)-unit output layer. The net’s output \( \mathbf{z}_t = [z_{t,1}, z_{t,2}, \ldots, z_{t,n}]^T \) is

\[
\mathbf{z}_t = \sum_{j=1}^{k} g_j(\mathbf{x}_t) O_j(\mathbf{x}_t),
\]

where \( \mathbf{x}_t = [x_{t,1}, x_{t,2}, \ldots, x_{t,d}]^T \) is an input, \( g_j(\mathbf{x}_t) \) is
an \( n \times 1 \) vector function whose \( r \)th component describes
the relations between hidden unit \( j \) and output unit \( r \).
\( O_j(\mathbf{x}_t) \) is the output of unit \( j \) in the hidden layer with

\[
O_j(\mathbf{x}_t) = \frac{\phi_j(\mathbf{x}_t - \mathbf{m}_j)^T \Sigma_j^{-1}(\mathbf{x}_t - \mathbf{m}_j)}{\sum_{i=1}^{k} \phi_i(\mathbf{x}_t - \mathbf{m}_i)^T \Sigma_i^{-1}(\mathbf{x}_t - \mathbf{m}_i)},
\]

where \( \mathbf{m}_j \) is the center vector, and \( \Sigma_j \) is the receptive
field of the basis function \( \phi(.) \). In general, one common
choice of function \( \phi(.) \) is the Gaussian function \( \phi(s^2) = \exp(-0.5s^2) \). Consequently, Eq.(3) becomes

\[
\mathbf{z}_t = \sum_{j=1}^{k} g_j(\mathbf{x}_t) \frac{\exp(-0.5(\mathbf{x}_t - \mathbf{m}_j)^T \Sigma_j^{-1}(\mathbf{x}_t - \mathbf{m}_j))}{\sum_{i=1}^{k} \exp(-0.5(\mathbf{x}_t - \mathbf{m}_i)^T \Sigma_i^{-1}(\mathbf{x}_t - \mathbf{m}_i))}.
\]

In the existing ENRBF networks as shown in [10, 11],
\( g_j(\mathbf{x}_t) \) is a linear function:

\[
g_j(\mathbf{x}_t) = \mathbf{W}_j \mathbf{x}_t + \beta_j, \quad j = 1, 2, \ldots, k,
\]

where \( \mathbf{W}_j \) is an \( n \times d \) parameter matrix, and \( \beta_j \) is an
\( n \times 1 \) constant vector. Here, we extend Eq.(6) to

\[
g_j(\mathbf{x}_t) = \mathbf{W}_j \text{dg}[\text{sign}(\mathbf{x}_t)]|\mathbf{x}_t|^{p_j} + \beta_j
\]

with

\[
\text{sign}(\mathbf{x}_t) = [\text{sign}(x_{t,1}), \text{sign}(x_{t,2}), \ldots, \text{sign}(x_{t,d})]^T,
\]

\[
|\mathbf{x}_t|^{p_j} = [|x_{t,1}|^{p_j}, |x_{t,2}|^{p_j}, \ldots, |x_{t,d}|^{p_j}]^T,
\]

where \( \text{dg}(\mathbf{x}_t) \) denotes the diagonal matrix whose \( (i,i) \)th
element is \( x_{t,i} \). That is, we use a single \( p \)-order poly-
nomial term to model the relations between each hid-
den unit and each output unit. By putting Eq.(7) into
Eq.(5), we then have

\[
\mathbf{z}_t = \sum_{j=1}^{k} (\mathbf{W}_j \text{dg}[\text{sign}(\mathbf{x}_t)]|\mathbf{x}_t|^{p_j} + \beta_j) O_j(\mathbf{x}_t),
\]

\[
O_j(\mathbf{x}_t) = \frac{\exp(-0.5(\mathbf{x}_t - \mathbf{m}_j)^T \Sigma_j^{-1}(\mathbf{x}_t - \mathbf{m}_j))}{\sum_{i=1}^{k} \exp(-0.5(\mathbf{x}_t - \mathbf{m}_i)^T \Sigma_i^{-1}(\mathbf{x}_t - \mathbf{m}_i))}.
\]

In Eq.(9), we have two parameter sets: \( \{\mathbf{m}_j, \Sigma_j\} \)’s in
the hidden layer and \( \{\mathbf{W}_j, p_j, \beta_j\} \)’s in the output layer.
We learn the parameters in the same way as the existing
approaches [2, 3, 7, 12] with the two steps:

Step 1 Learn \( \{\mathbf{m}_j, \Sigma_j\} \)’s in the hidden layer via a clus-
tering algorithm such as k-means [6] or RPCL [12];

Step 2 Learn \( \{\mathbf{W}_j, p_j, \beta_j\} \)’s in the output layer under
the least mean square criteria.

4. Dual Structural RBF Network

4.1. Structure

As shown in Figure 2, this new network consists of two
sub-RBF networks denoted as \( RBF_1 \) and \( RBF_2 \) respectively.
At each time step \( t \), the input of \( RBF_1 \) is \( \mathbf{x}_t \) whereas that of \( RBF_2 \) is the time-delayed past desired
outputs \( y_{t-1}, y_{t-2}, \ldots, y_{t-q} \). \( \mathbf{z}_t^1 \) is the output of \( RBF_1 \)
whereas \( \mathbf{z}_t^2 \) is \( RBF_2 \)’s output. \( \hat{y}_t \) is the network’s actual
output, which is the linear combination of \( \mathbf{z}_t^1 \) and \( \mathbf{z}_t^2 \):

\[
\hat{y}_t = c_1 \mathbf{z}_t^1 + c_2 \mathbf{z}_t^2,
\]

where \( c_1 \) and \( c_2 \) are the coefficients.

After the desired output \( y_t \) is available, we can calculate the
output residual

\[
\hat{e}_t = y_t - \hat{y}_t.
\]
Then, based on \( \hat{e}_t \), we learn the parameters \( c_1, c_2 \) as well as those in each sub-RBF network under a certain measurement criterion. We hereafter denote the parameters \( \{W_j, p_j, \beta_j\} \)’s in the RBFs as \( \Theta_r = \{W_j(r), p_j(r), \beta_j(r)\}_{j=1}^k \), where \( r = 1 \) and \( 2 \). In the next sub-section, we will give out the detailed learning algorithm by implementing RBF1 and RBF2 nets individually by our proposed ENRBF variant in Section 3, although the internal structure and implementation of RBF1 are independent from those of RBF2 in general.

4.2. Learning Algorithm

We learn the parameters by minimizing the mean square error (MSE) between network’s actual outputs \( \hat{y}_t \) and the desired outputs \( y_t \) with the cost function:

\[
J(\Theta) = \frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^T (y_t - \hat{y}_t),
\]

where \( \Theta = \{c_1, c_2, \Theta_1, \Theta_2\} \). In implementation, at each time step \( t \), we adaptively tune \( \Theta \) with a little small step along the descent direction of minimizing \( (y_t - \hat{y}_t)^T (y_t - \hat{y}_t) \). That is, we adjust \( \Theta \) by

\[
\Theta_{\text{new}} = \Theta_{\text{old}} - \eta \frac{\partial J(\Theta)}{\partial \Theta}_{\Theta_{\text{old}}}.
\]

Specifically, when the outputs of RBF1 and RBF2 are both given by Eq.(9), we have the following detailed algorithm:

Step 1  Fix \( \Theta \), given \( x_t \) and \( Y_{t-1} \), we obtain \( \hat{y}_t \) by Eq.(10).

Step 2  We adjust the coefficients \( c_1 \) and \( c_2 \) by

\[
c_r^{\text{new}} = c_r^{\text{old}} + \eta \hat{e}_t^T z_t^r, \quad r = 1, 2
\]

and

\[
W_j^{\text{new}}(r) = W_j^{\text{old}}(r) + \eta \Delta W_j(r), \quad p_j^{\text{new}}(r) = p_j^{\text{old}}(r) + \eta \Delta p_j(r), \quad \beta_j^{\text{new}}(r) = \beta_j^{\text{old}}(r) + \eta \Delta \beta_j(r)
\]

with

\[
\Delta W_j(r) = c_r^{\text{old}} O_j(u_t) \hat{e}_t |u_t|^p \text{d}g[\text{sign}(u_t)]
\]

\[
\Delta p_j(r) = c_r^{\text{old}} O_j(u_t) v_t^T \text{d}g[\text{sign}(u_t)] W_j^T \hat{e}_t
\]

\[
\Delta \beta_j(r) = c_r^{\text{old}} O_j(u_t) \hat{e}_t
\]

where \( j = 1, 2, \ldots, k \), \( v_{t,j} = [ |u_{t,1}|^p \text{ln} |u_{t,1}|, |u_{t,2}|^p \text{ln} |u_{t,2}|, \ldots, |u_{t,k}|^p \text{ln} |u_{t,k}| ]^T \), \( \zeta = d \) in the RBF1 and \( \zeta = nq \) in the RBF2.

5. Experimental Results

We demonstrated the performance of the proposed network in the recursive function estimation. We generated 1,100 data points \( \{x_t, y_t\} \)’s from the following equation:

\[
y_t = 0.7(\sin x_t)^3 + 0.3 y_{t-1}^2 + e_t, \quad t \geq 1
\]

with \( y_0 = 0 \), where \( x_t \in [1, 10] \), \( e_t \in [-0.1, 0.1] \) is white noise with uniformly distributed. We let the first 1,000 data points be training set, and the remaining 100 points be testing set. The data distribution graph is shown in Figure 3. In the experiment, we fix \( \eta = 0.001 \). Furthermore, we set the size of each sub-RBF network to be \( k = 5 \) and \( n = 1 \). We measured the net’s performance under the MSE criterion.

After repeatedly scanning the training data set 100 times, the net’s performance on training data set tends to converge as shown in Figure 4. We then applied the net to the testing set. We obtained the MSE value to be 0.0082. In comparison, we also applied a classic ENRBF network to the same data set. We found that this network’s performance deteriorates significantly with the MSE value on the testing set to be 0.057. In this case, our proposed approach outperforms the classic one with the performance improved about 85.6%.
6. Conclusions

We have presented a dual structural RBF network, which is a hybrid system that consists of two sub-RBF networks. One sub-RBF network models the relationship between the current network output and the past ones, while the other one describes the relationship between the current network output and the inputs. We have implemented each sub-RBF network by proposing a new ENRBF variant, where an adjustable $p$-order single-term polynomial is used to fit the relations between each hidden unit and each output unit. The preliminary experiments have shown that this dual structural RBF network outperforms the classical one in recursive function estimation.

References