



# Journal of Applied Sciences

ISSN 1812-5654

**science**  
alert

**ANSI***net*  
an open access publisher  
<http://ansinet.com>

## The Effect of Several Parameters on Radial Basis Function Networks for Time Series Prediction

Mitat Uysal

Department of Computer Engineering Doğuş University, Kadıköy 34722, İstanbul, Turkey

**Abstract:** In this study, several radial basis function networks are compared according to their approximation ability in time series forecasting problems. Optimal values for the tested parameters are obtained using computer simulation runs. Effects of width selection in Gaussian Kernels, of the number of neurons in the hidden layer, and of selection of kernel function are investigated.

**Key words:** Radial basis functions, forecasting, time series, prediction, function approximation

### INTRODUCTION

There are many applications of time series in science and engineering, like electrical load estimation, risk prediction, river flood forecasting, stock market prediction, etc.

For making a prediction using time series, a large variety of approaches are available. Prediction of scalar time-series  $\{x(n)\}$  refers to the task of finding an estimate  $\hat{x}(n+1)$  of the next future sample  $x(n+1)$  based on the knowledge of the history of time-series, i.e, the samples  $x(n), x(n-1), \dots$  (Rank.,2003).

Linear prediction, where the estimate is based on a linear combination of  $N$  past samples can be represented as below:

$$\hat{x}(n+1) = \sum_{i=0}^{N-1} \alpha_i x(n-i)$$

with the prediction coefficients  $\alpha_i, i = 0,1, \dots N-1$ .

Introducing a general nonlinear function  $f(\cdot): \mathfrak{R}^n \rightarrow \mathfrak{R}$  applied to the vector  $x(n) = [x(n), x(n - M), \dots, x(n-(N-1))]^T$  of past samples, we arrive at the nonlinear prediction approach

$$\hat{x}(n+1) = f(x(n))$$

### RADIAL BASIS FUNCTION NETWORK

The RBF network consists of 3 layers: An input layer, a hidden layer and an output layer. A typical RBF network is shown in Fig. 1.

Mathematically, the network output for linear output nodes can be expressed as below:

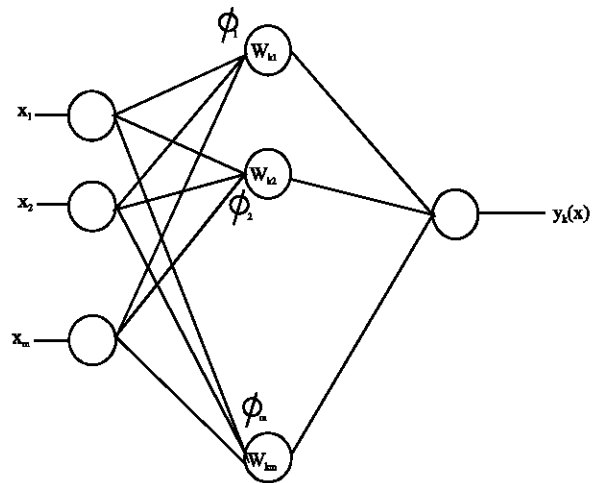


Fig. 1: Typical RBF network

$$y_k(x) = \sum_{j=1}^m W_{kj} \phi_j(\|x - \bar{x}_j\|) + W_{k0}$$

where,  $x$  is the input vector with elements  $x_i$  (where  $i$  is the dimension of the input vector);  $\bar{x}_j$  is the vector to determine the center of the basis function;  $\phi_j$  with elements  $\bar{x}_{ij}$ ;  $W_{kj}$ 's are the weights and  $W_{k0}$  is the bias (Harpham *et al.*, 2006). The basis function  $\phi_j(\cdot)$  provides the non-linearity.

### BASIS FUNCTIONS

The most used basis functions are Gaussian and multiquadratic functions. They are given below:  
 Gaussian

$$\phi(x) = \exp\left(-\frac{x^2}{2\delta^2}\right) \text{ for } \delta > 0 \text{ and}$$

$x \in \mathfrak{R}$

Multiquadratic

$$\phi(x) = (x^2 + \delta^2)^p \text{ for } \delta > 0 \text{ and } x \in \mathfrak{R}$$

p is between 0 and 1. Usually p is taken as 1/2.

**CALCULATING THE OPTIMAL VALUES OF WEIGHTS**

A very important property of the RBF Network is that it is a linearly weighted network in the sense that the output is a linear combination of m radial basis functions, written as below:

$$f(x) = \sum_{i=1}^m w^{(i)} \phi_i^{(i)}(x)$$

The main problem is to find the unknown weights  $\{W^{(i)}\}_{i=1}^m$ .

For this purpose, the general least squares principal can be used to minimize the sum squared error:

$$SSE = \sum_{i=1}^n [y^{(i)} - f(x^{(i)})]^2$$

With respect to the weights of f, resulting in a set of m simultaneous linear algebraic equations in the m unknown weights

$$(A^T A) w = A^T y$$

Where,

$$A = \begin{bmatrix} \phi^{(1)}(x^{(1)}) & \phi^{(2)}(x^{(1)}) & \dots & \phi^{(m)}(x^{(1)}) \\ \phi^{(1)}(x^{(2)}) & \phi^{(2)}(x^{(2)}) & \dots & \phi^{(m)}(x^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi^{(1)}(x^{(n)}) & \phi^{(2)}(x^{(n)}) & \dots & \phi^{(m)}(x^{(n)}) \end{bmatrix}$$

$$w = [w^{(1)}, w^{(2)}, \dots, w^{(m)}]^T$$

$$y = [y^{(1)}, y^{(2)}, \dots, y^{(n)}]^T$$

In the special case where n = m, the resultant system is just

$$Aw = y \text{ (Day et al., 2003)}$$

The output y(x) represents the next value of y in time t taking input values  $x_1, x_2, \dots, x_n$  that represent the

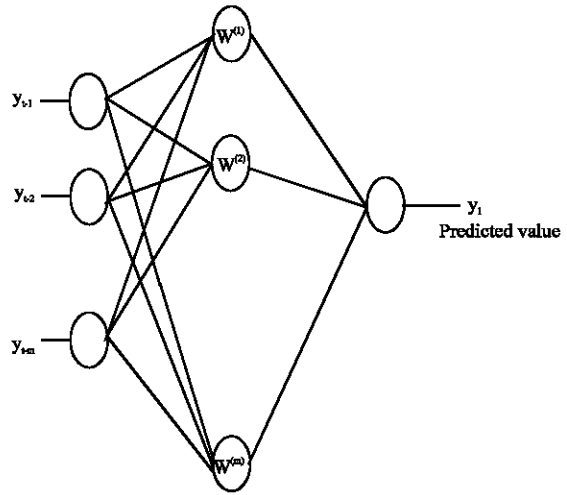


Fig. 2: Finding predicted value y<sub>i</sub>

previous function values set with values  $y_{t-1}, y_{t-2}, \dots, y_{t-n}$ . So,  $x_n$  corresponds to  $y_{t-1}$ ,  $x_{n-1}$  corresponds to  $y_{t-2}$  etc. as in Fig. 2.

**SIMULATION RESULTS**

Several computer simulation runs are carried out to find the optimal values of parameters in radial basis functions such as width ( $\delta$ ) and centers ( $\bar{x}_i$ 's).

The effect of type of radial basis functions (Gaussian, multiquadratic etc.) in function approximation is also investigated.

The last parameter to be investigated is the number of neurons in the hidden layer. The effect of the number of neurons in the hidden layer on performance of neural network for time series prediction is studied.

**EFFECT OF WIDTH SELECTION**

For this work, the time-series data of American Express Bank is used. Monthly log data consists of 324 data items. The first 162 data items are used for training and the remaining 162 data items are used for forecasting.

Figure 3 shows the results of simulation run with  $\delta = 0.5$  and 18 neurons in the hidden layer for the last 50 data items.

In Fig. 4, similar results for  $\delta = 1.2$  and 18 neurons in the hidden layer are shown.

For  $\delta = 1.5$  and 18 neurons in the hidden layer an optimal solution is obtained with minimum error rate. This result is shown in Fig. 5.

In Fig. 6, for the optimal solution, all the real and predicted values are shown.

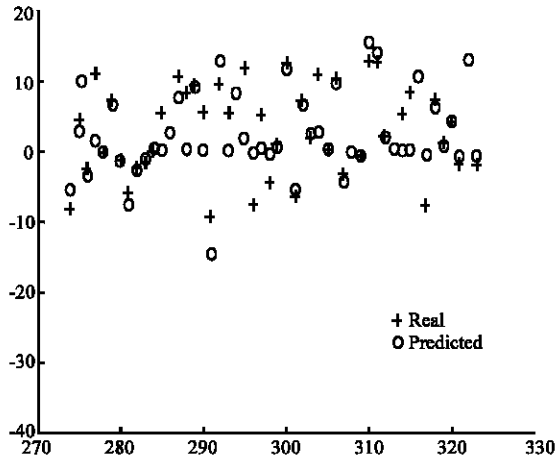


Fig. 3:  $\delta = 0.5$  and 18 neurons in the hidden layer. Last 50 data items

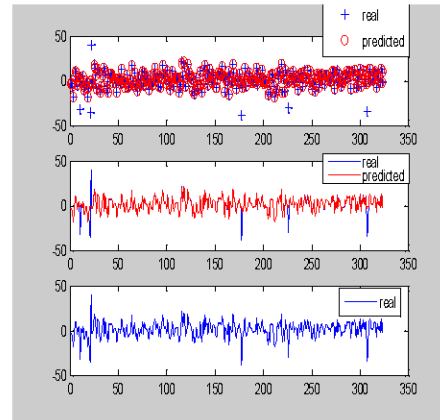


Fig. 6:  $\delta = 1.5$  and 18 neurons in the hidden layer. Optimal solution

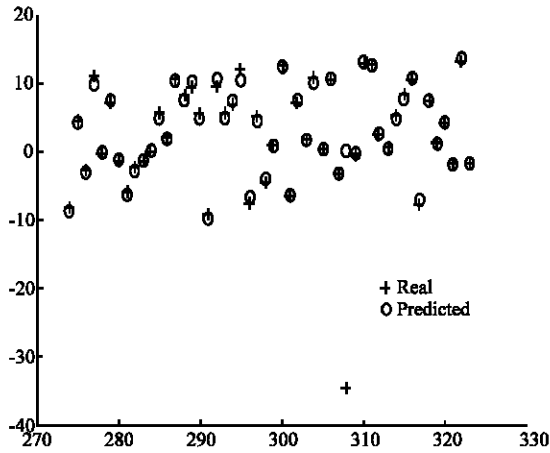


Fig. 4:  $\delta = 1.2$  and 18 neurons in the hidden layer. Last 50 data items

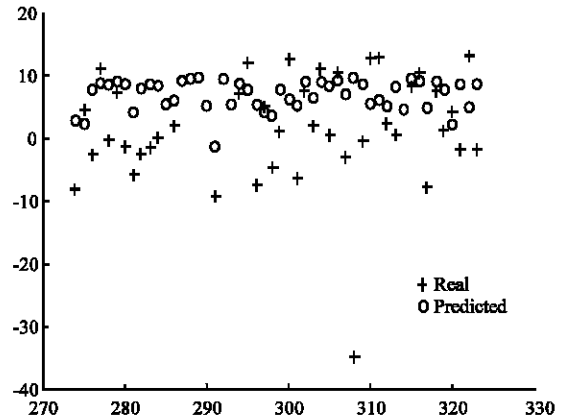


Fig. 7:  $\delta = 1.5$  and 9 neurons in the hidden layer. Last 50 data items

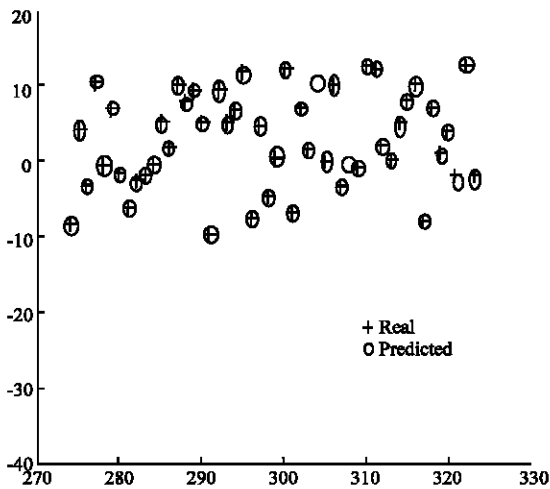


Fig. 5:  $\delta = 1.5$  and 18 neurons in the hidden layer. Last 50 data items

As can be seen from simulation results presented above, the width parameter ( $\delta$ ) has an important effect on optimal solution.

### EFFECT OF NUMBER OF NEURONS IN THE HIDDEN LAYER

The second important parameter is the number of neurons that are used in the hidden layer of the RBF network.

In Fig. 7, simulation results for  $\delta = 1.5$  and 9 neurons in the hidden layer are shown.

If we compare these results with the results that is given in Fig. 5 for  $\delta = 1.5$  and 18 neurons in the hidden layer, then we can see big differences between two figures.

If we increase number of number of neurons in the hidden layer while  $\delta$  remains fixed, then we can obtain better results in the prediction problem.

### **EFFECT OF KERNEL FUNCTIONS**

The effect of the type of kernel function (Gaussian, multivariate etc) is problem dependent, which means it can change from one problem to another.

### **CONCLUSIONS**

In this study, different radial basis function networks are compared according to their ability to predict results in time series forecasting problem.

Optimal values for the tested parameters are obtained using simulation runs. Optimal width value of the Gaussian function is obtained as 1.5 for the data file to be processed and the optimal number of neurons in the hidden layer of RBF network is found as 18 for the same problem.

In future research, the relationships between the statistical parameters of data points (average, standard deviation, etc.) and parameters of the RBF network will be investigated for the optimal solution in time series forecasting problems.

### **REFERENCES**

- Day, N.M. and T.T. Cang, 2003. Approximation of function and its derivatives using radial basis function networks. *Applied Mathematical Modeling*, 27: 197-220.
- Harpham, C. and C.W. Dawson, 2006. The effect of different basis functions on a radial basis function network for time series prediction: A comparative study. *Neurocomputing*. (In Press).
- Rank, E., 2003. Application of Bayesian trained RBF networks to nonlinear time-series modeling. *Signal Processing*, 83: 1393-1410.