Brief Papers

Gradient Radial Basis Function Networks for Nonlinear and Nonstationary Time Series Prediction

E. S. Chng, S. Chen, and B. Mulgrew

Abstract—**We present a method of modifying the structure of radial basis function (RBF) network to work with nonstationary series that exhibit homogeneous nonstationary behavior. In the original RBF network, the hidden node's function is to sense the trajectory of the time series and to respond when there is a strong correlation between the input pattern and the hidden node's center. This type of response, however, is highly sensitive to changes in the level and trend of the time series. To counter these effects, the hidden node's function is modified to one which detects and reacts to the gradient of the series. We call this new network the gradient RBF (GRBF) model. Single and multistep predictive performance for the Mackey–Glass chaotic time series were evaluated using the classical RBF and GRBF models. The simulation results for the series without and with a time-varying mean confirm the superior performance of the GRBF predictor over the RBF predictor.**

I. Introduction

THE radial basis function (RBF) network has enjoyed considerable success in application to nonlinear time-series prediction [1]–[4]. Most of the successful results, however, are obtained when the network is applied to predict signals that are stationary. The performance of the RBF predictor for nonstationary signal is less satisfactory [5]. This is because the RBF network, like many other neural-network models, does not characterize temporal variability well. Since real-world signals are often not only highly nonlinear but also highly nonstationary, it is desired to develop predictors which can handle signals that exhibit both such characteristics.

For nonstationary time series involving variations of local mean and trend, the series can be made stationary by applying a suitable difference on the signal. This is the implementation behind the linear auto-regressive integrated moving average (ARIMA) model [6] to predict nonstationary signals. By incorporating a similar mechanism into the RBF network, the resulting model will have better predictive performance for nonstationary homogeneous time series. As well as using the difference of the signal as the input vector to the network, we also modify the response of each hidden node with a term which can be interpreted as a local one-step predictor. We refer to this new network as the GRBF network.

In Section II, after a summary of the classical RBF network, the GRBF network is introduced and some geometric interpretations of this model are given. The construction of the GRBF network using the orthogonal least squares (OLS) subset selection technique [2] is briefly discussed. Simulation results using the classical RBF and GRBF networks to predict the Mackey–Glass chaotic time series with and without time-varying mean/trend are given in Section III to demonstrate the effectiveness of the GRBF network. Section IV contains some concluding remarks.

II. The RBF Network

The RBF network is a single-hidden-layer feedforward neural network [4]. Each node of the hidden layer has a parameter vector called center. This center is used to compare with the network input vector to produce a radially symmetrical response. Response of the hidden layer are scaled by the connection weights of the output layer and then combined to produce the network output.

To predict the signal value y_i, the RBF network input vector

$$\mathbf{x}_i = [y_{i-1}, y_{i-2}, \cdots, y_{i-M}]^T \tag{1}$$

is an M-dimensional vector consisting of past signal samples. M is often referred to as the embedding vector length. In the present study, we choose the Gaussian function as the nonlinearity of the hidden nodes. The response of the jth hidden node to $\mathbf{x}_i$ is given by

$$\phi_{ij} = \exp(-\alpha\|\mathbf{x}_i - \mathbf{c}_j\|^2) \tag{2}$$

where $\mathbf{c}_j$ is an M-dimensional center vector and α is a positive constant which determines the width of the symmetric response of the hidden node. Theoretical investigations have shown that the choice of nonlinearity for hidden nodes is not crucial [7], [8] and a uniform width for every hidden node is sufficient for universal approximation [8]. The network output is defined as

$$\hat{y}_i = \sum_{j=1}^{K} \phi_{ij} h_j \tag{3}$$

where h_j are the network connection weights and K is the number of hidden nodes.

Manuscript received October 12, 1993; revised May 10, 1994 and October 11, 1994.

E. S. Chng is with the Laboratory for Artificial Brain Systems, Frontier Research Programme, Riken, 2–1 Hirosawa, Wako-shi, Saitama 351–01, Japan.

S. Chen is with the Department of Electrical and Electronic Engineering, The University of Portsmouth, Portsmouth PO1 3DJ, England.

B. Mulgrew is with the Department of Electrical Engineering, The University of Edinburgh, Edinburgh EH9 3J1, Scotland.

Publisher Item Identifier S 1045-9227(96)00160-9.

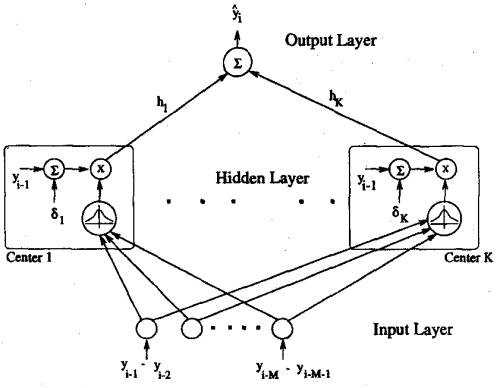

Fig. 1. Topology of first-order GRBF network.

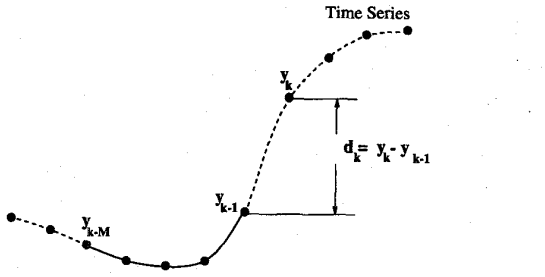

Fig. 2. Training mode of jth hidden node. If the kth training input $\mathbf{x}'_k$ is chosen as the center $\mathbf{c}'_j$, δ_j is set to d_k.

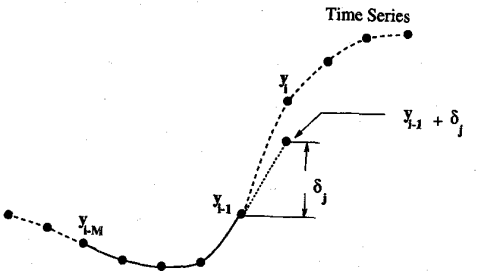

Fig. 3. Predictive mode of jth hidden node. If the center $\mathbf{c}'_j$ matches $\mathbf{x}'_i$, $(y_{i-1} + \delta_j)$ is a good approximation of y_i.

Assume that N samples of the signal, $\{y_i\}_{i=1}^N$, are available for training. The centers, $\mathbf{c}_j$, $1 \le j \le K$, can be selected from the network training input $\mathbf{x}_i$, $1 \le i \le N$. The weights can then be solved for using the least squares method [9]. A constructive approach is to use the OLS algorithm to simultaneously determine RBF centers and weights [2].

A. The GRBF Network

The GRBF network, like the RBF case, is a single-hidden-layer feedforward neural network. In the GRBF network, however, the network input vector is generated by differencing the raw data. The order of differencing determines the order of the GRBF network. For example, the input vector of the first-order GRBF network at time i is given by

$$\begin{aligned}\mathbf{x}'_i &= \mathbf{x}_i - \mathbf{x}_{i-1} \\ &= [y_{i-1} - y_{i-2}, y_{i-2} - y_{i-3}, \cdots, y_{i-M} - y_{i-M-1}]^T \end{aligned} \quad (4)$$

where $\mathbf{x}_i$ and $\mathbf{x}_{i-1}$ are the original input vectors to the RBF network at time i and $(i-1)$, respectively. The elements of $\mathbf{x}'_i$ show the rate of change in the trajectory of the time series for the past M samples.

Fig. 1 depicts the structure of the first-order GRBF network. Although the Gaussian function still serves as the nonlinear function which compares the similarity of the input vector to the hidden node's center, the response of the Gaussian function is now multiplied by an additional term $(y_{i-1} + \delta)$. The response of the jth hidden node of a first-order GRBF network to the input vector $\mathbf{x}'_i$ is therefore given by

$$\phi'_{ij} = \exp(-\alpha \|\mathbf{x}'_i - \mathbf{c}'_j\|^2) \times (y_{i-1} + \delta_j) \quad (5)$$

where $\mathbf{c}'_j$ is an M-dimensional center vector and δ_j is a constant value associated with the center. The term $(y_{i-1}+\delta_j)$ can be interpreted as a local single-step prediction of y_i by the jth hidden node. From (5), if the input vector is very similar to the jth center, the value of the Gaussian function will be close to 1.0 and the predictor $(y_{i-1}+\delta_j)$ becomes fully active. As in the case of the RBF network, the output layer is a linear combiner with weights h_j, $1 \le j \le K$.

The centers $\mathbf{c}'_j$ and the scalars δ_j, $1 \le j \le K$, can be chosen during the training from the training data $\{\mathbf{x}'_k\}_{k=1}^N$ as illustrated in Fig. 2. For each training input vector $\mathbf{x}'_k$, define

$$d_k = y_k - y_{k-1}. \quad (6)$$

If $\mathbf{x}'_k$ is selected as the jth center $\mathbf{c}'_j$, we set $\delta_j = d_k$ to ensure that the jth hidden node is a perfect predictor of y_k. In this way, the problem of constructing a network is equivalent to the task of selecting a K-term subset model $\{\mathbf{c}'_j, \delta_j\}_{j=1}^K$ from the full N-term model $\{\mathbf{x}'_k, d_k\}_{k=1}^N$. The OLS algorithm can readily be applied to perform this subset selection task.

The rationale behind the GRBF model become obvious when the network performs predictive operation. Each hidden node compares the network input vector $\mathbf{x}'_i$ with its center $\mathbf{c}'_j$. The Gaussian response of each hidden node indicates the degree of matching between $\mathbf{x}'_i$ and $\mathbf{c}'_j$. The hidden nodes thus sense the gradient of the time series rather than the series itself as in the case of the RBF model. The term $(y_{i-1} + \delta_j)$ also has a clear geometric meaning. Referring to Fig. 3, if the jth center $\mathbf{c}'_j$ matches the gradient $\mathbf{x}'_i$ of the series, $(y_{i-1} + \delta_j)$ is likely to be a very good prediction of y_i.

Fig. 4 illustrates different behavior of the RBF and GRBF nodes. The task is to sense peaks of a sinusoid signal embedded in a varying mean. Both the RBF and GRBF centers were set according to a segment of signal containing a peak. The results clearly show that the GRBF node was able to respond to each peaks while the RBF node was unable to track the time series.

Although the GRBF has better generalization property, the complexity of the GRBF hidden node is greater than that of a RBF hidden node. This, however, is not a significant disadvantage as the better generalization property often results in smaller network. Therefore, the overall complexity of the GRBF network may not necessarily be greater than that of the RBF network in practical applications.

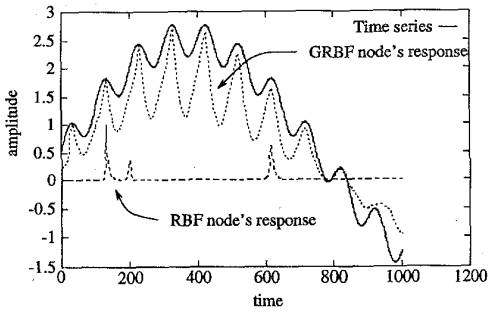

Fig. 4. RBF and GRBF node's response.

B. Higher-Order GRBF Networks

We can extend the idea of mapping the data's gradient by the first-order GRBF network to that of matching higher-order gradient by a higher-order GRBF network. For instance, the input vector to the second-order GRBF network at time i can be defined as

$$\mathbf{x}_i'' = \mathbf{x}_i' - \mathbf{x}_{i-1}' = [(y_{i-1} - y_{i-2}) - (y_{i-2} - y_{i-3}), \cdots, (y_{i-M} - y_{i-M-1}) - (y_{i-M-1} - y_{i-M-2})]^T. \quad (7)$$

The response of the jth hidden node of the second-order GRBF network to $\mathbf{x}_i''$ is calculated according to

$$\phi_{ij}'' = \exp(-\alpha \|\mathbf{x}_i'' - \mathbf{c}_j''\|) \times ((y_{i-1} - y_{i-2}) + \delta_j'). \quad (8)$$

The M-dimensional center vectors $\mathbf{c}_j''$ and the scalars δ_j', $1 \le j \le K$, can similarly be selected from the training data $\{\mathbf{x}_k''\}_{k=1}^N$. For each training input vector $\mathbf{x}_k''$, define

$$d_k' = d_k - d_{k-1} = (y_k - y_{k-1}) - (y_{k-1} - y_{k-2}). \quad (9)$$

If $\mathbf{x}_k''$ is selected as the jth center $\mathbf{c}_j''$, the value of δ_j' is set to d_k'. The OLS algorithm is well suited for this subset selection problem.

Geometric properties of the first-order GRBF network can similarly be extended to a higher-order GRBF network. If we view that the first-order GRBF network uses a matching of gradient to predict the next value of the time series, then the second-order GRBF network predicts the next rate of change based on a matching of second-order gradient. This interpretation can be generalized to higher-order GRBF networks.

C. Subset Model Selection

In the previous work [2], the problem of constructing a RBF network from training data is formulated as one of selecting a K-term subset network from the N-term full network based on the OLS algorithm. The same approach can readily be applied to construct a GRBF network from training data. In fact, the OLS algorithm can be applied to any model which has a linear-in-the-parameter structure.

The OLS algorithm is an efficient way of implementing the forward regression procedure [10] and, therefore, it does not guarantee to find the best K-term subset model from the N-term full model. This, however, is not a serious deficiency as the subset model found are normally very good. Furthermore, the algorithm is computationally very efficient. This allows us to work on a very large initial model set, which can combine the linear, RBF, and GRBF expansions. Attempt to find the optimal K-term subset model from such a large model set would almost certainly be impractical.

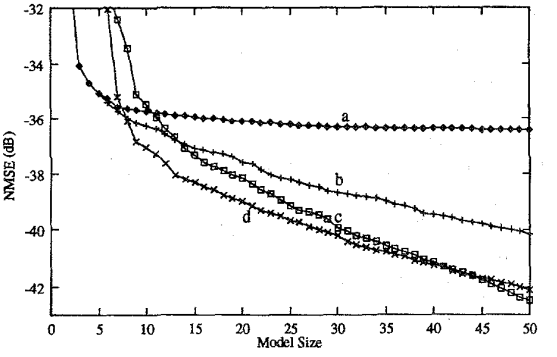

Fig. 5. Performance of predictors in training phase for Mackey–Glass series. a) linear model, b) linear and RBF model, c) linear, RBF, and first-order GRBF model, and d) linear, RBF, first- and second-order GRBF model.

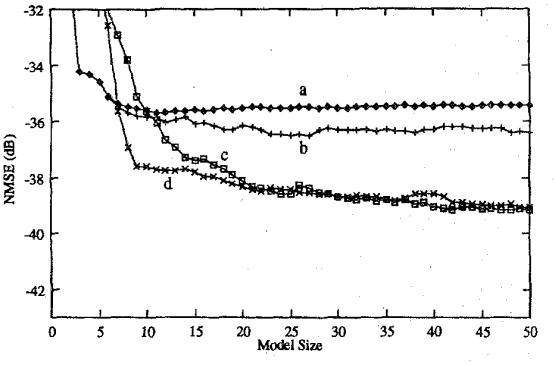

Fig. 6. Performance of predictors in testing phase for Mackey–Glass series. a) linear model, b) linear and RBF model, c) linear, RBF, and first-order GRBF model, and d) linear, RBF, first- and second-order GRBF model.

III. Simulation Results

We present some simulation results of time series prediction using the RBF and GRBF predictors in this section. Initial full models were created by using all the available data in the training set as RBF and/or GRBF centers. Some linear terms were also included into the full models. Subset models were then selected from these large full models using the OLS scheme, and used to evaluate single-step and multistep prediction performance.

The time series used to evaluate the performance of the various models is the Mackey–Glass time series in chaotic mode generated using the following equation

$$\frac{ds(t)}{dt} = -bs(t) + \frac{\rho s(t-\tau)}{1 + s^{10}(t-\tau)} \quad (10)$$

where $\tau = -21$, $\rho = 0.2$, $b = 0.1$, initial conditions $s(t-\tau) = 0.5$ for $0 \le t \le \tau$ and step size $= 2$ sec. Gaussian white noise was added to the series to create a signal to noise ratio (SNR) of 50 dB.

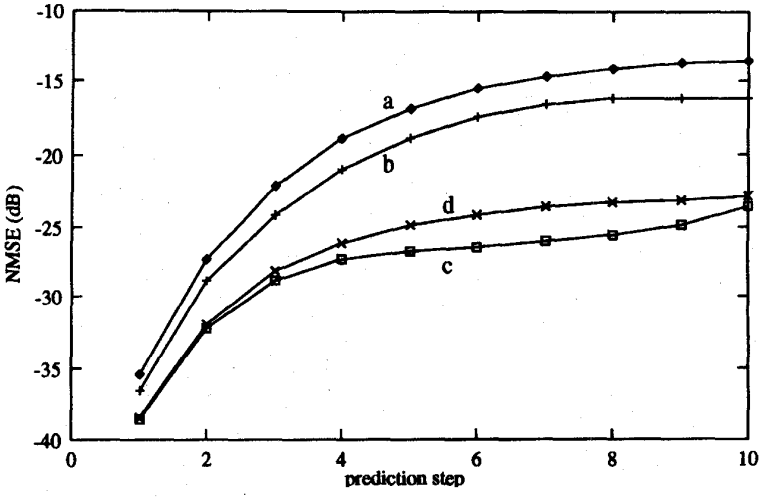

Fig. 7. Multistep performance of predictors with a model size of 25 for Mackey–Glass series. a) linear model, b) linear and RBF model, c) linear, RBF, and first-order GRBF model, and d) linear, RBF, first- and second-order GRBF model.

Data samples of point 100–600 were used as the training set ($N = 500$) and samples 601 to 1100 were used as the validation set. The embedding vector's dimension was chosen to be $M = 6$[1] and the width of Gaussian function was set to $\alpha = 1.0$. The following types of models were used:

L-model	Linear model of order 50.
L0-model	Combinations of the linear mode and the classical RBF model.
L01-model	Combinations of the linear model, the classical RBF and first-order GRBF models.
L012-model	Combinations of the linear model, the classical RBF model, the first and second-order GRBF models.

A. Simulation for Stationary Series

For the Mackey–Glass time-series, the results of single-step performance for the predictors in training phase are shown in Fig. 5, where the vertical axis indicates the normalized mean square error (NMSE) in dB. As expected, as the size of each selected subset model increases, the accuracy of the model continued to improve. The rate of improvement, however, was not the same for each model. The predictors with GRBF expansion, i.e., L01 and L012-models, achieved better error reduction with a smaller model size. These two GRBF subset models also performed better on the validation set compared with the classical RBF model, as can be seen in Fig. 6. The multistep prediction performance on the validation set for the models were tested using a model size of 25 (Fig. 7), and the results show that the two GRBF models had better multistep predictive accuracy.

B. Simulation for Nonstationary Series

To examine how the predictors behave for nonstationary series, we used a modified Mackey–Glass time-series. This new series was formed by adding sinusoid with amplitude 0.3 and a period of 3000 samples to the Mackey–Glass time series

[1]The embedding dimension must be chosen to be greater than $2 \times D$, where D is the attractor's dimension [11].

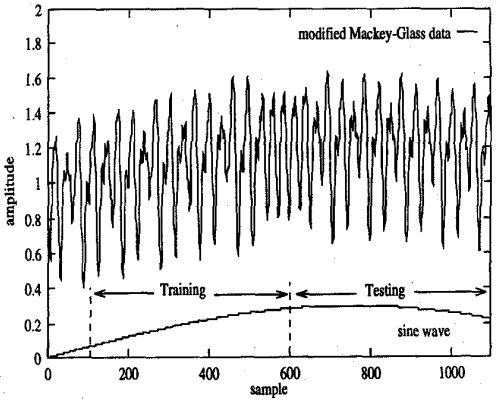

Fig. 8. Modified Mackey–Glass time series.

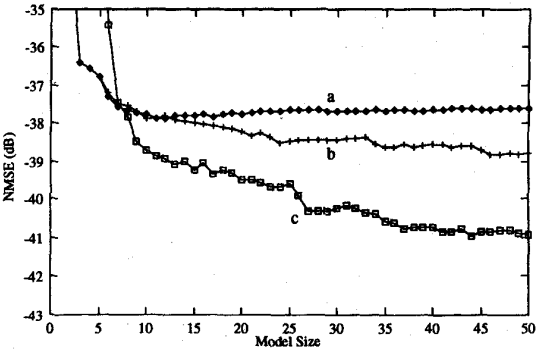

Fig. 9. Performance of predictors in testing phase for modified Mackey–Glass series. a) linear model, b) linear and RBF model, and c) linear, RBF, and first-order GRBF model.

used in the previous example (Fig. 8). As the training data were formed from samples 100 to 600 and the validation data consisted of samples 601–1100, the predictors were trained without being exposed to the change in the level and trend of the test data. The results for the single-step prediction in the validation phase (Fig. 9) and the multistep predictive

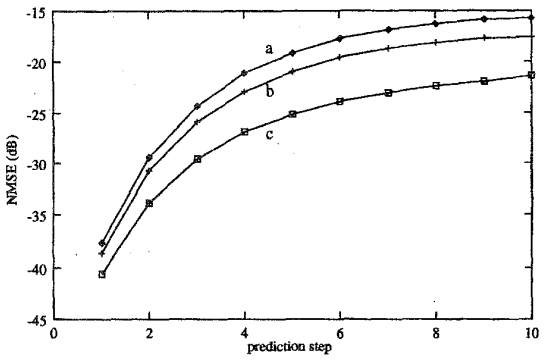

Fig. 10. Multistep performance of predictors with a model size of 35 for modified Mackey–Glass series. a) linear model, b) linear and RBF model, and c) linear, RBF, and first-order GRBF model.

performance on the validation set using a model size of 35 (Fig. 10) suggest that the GRBF network can perform better than the classical RBF network in a nonstationary environment.

IV. Conclusion

We have presented a GRBF network for nonlinear and nonstationary time series prediction. The hidden layer of this GRBF network is designed to respond to the gradient of time-series rather than the trajectory itself. This can usually improve predictive accuracy, particularly for homogeneous nonstationary time series as are demonstrated in the simulation results. The construction of the GRBF predictor from the time series observations has been proposed using the OLS subset selection algorithm. Although the discussion was based on time series prediction, this GRBF network can be applied to other signal processing applications.

Acknowledgment

The authors would like to thank D. Hughes for providing the Mackey–Glass data generator used in the simulations. They also thank the reviewers for their comments and suggestions.

References

[1] M. A. S. Potts and D. S. Broomhead, "Time series prediction with a radial basis function neural network," *SPIE Adaptive Signal Processing*, vol. 1565, pp. 255–266, 1991.
[2] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 302–309, 1991.
[3] M. Casdagli, "Nonlinear prediction of chaotic time-series," *Physica D*, vol. 35, pp. 335–356, 1989.
[4] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Syst.*, vol. 2, pp. 321–355, 1988.
[5] E. Levin, "Hidden control neural architecture modeling of nonlinear time varying systems and its applications," *IEEE Trans. Neural Networks*, vol. 4, pp. 109–116, 1993.
[6] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control.* Oakland, CA: Holden-Day, 1976.
[7] M. J. D. Powell, "Radial basis functions for multivariable interpolation: A review," in *Algorithms for Approximation*, J. C. Mason and M. G. Cox, Eds. Oxford, UK: Oxford University Press, 1987, pp. 143–167.
[8] J. Park and I. W. Sandberg, "Universal approximation using radial basis function networks," *Neural Computa.*, vol. 3, pp. 246–257, 1991.
[9] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math.*, vol. 14, pp. 403–420, 1970.
[10] R. R. Hocking, "The analysis and selection of variables in linear regression," *Biometrics*, vol. 32, pp. 1–49, 1976.
[11] F. Takens, "Detecting strange attractors in turbulence," in *Lecture Notes in Mathematics*. New York: Springer-Verlag, 1980, pp. 366–381.