Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EFFICIENT TRAINING OF RBF NETWORKS VIA THE KURTOSIS AND SKEWNESS MINIMIZATION LEARNING ALGORITHM

[1]LIN WANG, [2]JINWEN MA

[1] Department of Information Science, School of Mathematical Sciences and LAMA, Peking University

[2] Prof., Department of Information Science, School of Mathematical Sciences and LAMA, Peking University

E-mail: [1]wal@pku.edu.cn , [2] jwma@math.pku.edu.cn

ABSTRACT

Radical Basis Function (RBF) networks have been widely used in time series prediction because of their simplicity, robustness, good approximation and generalization ability. However, it is still rather difficult to select the number and locations of the hidden units of the RBF network appropriately for a specific time series prediction problem. In this paper, the Generalized RBF networks have been established with Gaussian density functions instead of Gaussian functions for the hidden units. Then, we utilize the kurtosis and skewness minimization criterion as well as the corresponding learning algorithm to select the appropriate number and initial parameters of the hidden units from an input data set automatically for the generalized RBF network on a time series prediction task. It has been demonstrated by the experiments that the generalized RBF network trained with the kurtosis and skewness minimization learning algorithm is feasible and efficient on typical chaos time series.

Keywords: *Radical Basis Function (RBF) Network, Time Series Prediction, Kurtosis and Skewness Minimization Criterion*

1. INTRODUCTION

The Radial Basis Function (RBF) networks [1]-[2] are a typical class of forward neural networks widely used in the fields of pattern recognition and information processing. Actually, the RBF network is a two-layer forward neural network such that each hidden unit implements a radial basis function and the output units implement a weighted sum of hidden unit outputs. With such a structure, it can approximate any continuous function as long as the number of hidden units is large enough. Moreover, its structure is simple and the learning process is quite efficient with a good generalization. Therefore, the RBF network has been widely applied to various practical problems involved in speech recognition, clustering analysis, time series prediction, etc., and the most commonly used radial basis functions in the RBF networks are Gaussian activation functions.

However, the training of the RBF network is still a rather difficult task. The key crucial problem is how to select the number and locations of the hidden units appropriately for a practical problem. Although there are many learning methods for training the RBF network, most of them utilize a test-and-growing or evolutionary mechanism to select the number of hidden units for practical applications (e.g., [3]-[5]). Clearly, such learning methods are time consuming and quite easy to be trapped in a local solution. Actually, a good selection of hidden units should appropriately match the structure of input data associated with a practical problem.

Recently, some new learning methods have been proposed to determine the number of hidden units of the RBF network. With the development of competitive learning, the rival penalized competitive learning (RPCL) algorithm was proposed to determine the number of clusters or Gaussians in a dataset automatically [6]-[7]. Thus, it provided a new tool for selecting the number of Gaussians or units from the input data for the RBF network. Alternatively, based on the Bayesian Ying-Yang (BYY) harmony learning theory [8]-[9], a new kind of automated model selection (AMS) learning algorithms have been established for the Gaussian mixture modeling [10]-[12]. As a matter

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of fact, this kind of BYY-AMS learning algorithms can automatically determine the number of Gaussians in a dataset during parameter learning, which can be utilized to select the number of Gaussians as being the hidden units in the RBF network with an input dataset.

For time series prediction, the RBF network plays an important role and often leads to a better result (e.g., [13]-[14]). Since the input data associated with a time series prediction problem are generally subject to a Gaussian mixture model, it is better to generalize the RBF network by replacing the Gaussian kernel functions with Gaussian density functions. In this way, the training of the RBF network can be effectively implemented by an AMS learning algorithm for Gaussian mixtures [15]. In [16], a powerful kurtosis and skewness minimization criterion of model selection as well as the corresponding greedy EM algorithm was established for Gaussian mixture modeling with automated model selection. Moreover, the greedy EM algorithm was further improved in [17]. However, these greedy EM algorithms are time consuming and easy to be trapped in a local solution.

In the current paper, we propose a split-and-merge EM algorithm based on the kurtosis and skewness minimization criterion. This new kurtosis and skewness minimization learning algorithm starts from an appropriate initial number of Gaussians and finally arrives at the correct number of Gaussians in the input dataset automatically and robustly. Moreover, we apply it to the training of the generalized RBF network for time series prediction. It is demonstrated by the experiments on two typical chaos time series datasets that the proposed kurtosis and skewness minimization learning algorithm is effective and efficient on the training of the generalized RBF network for time series prediction.

In the sequel, the kurtosis and skewness minimization criterion as well as the kurtosis and skewness minimization learning algorithm are introduced in Section II. In Section III, we present the model of the generalized RBF network as well as the training method according to the kurtosis and skewness minimization learning algorithm. The time series prediction experimental results of the generalized RBF network with the kurtosis and skewness minimization learning algorithm are demonstrated in Section IV. We finally give a brief conclusion in Section V.

2. KURTOSIS AND SKEWNESS MINIMIZATION LEARNING ALGORITHM

2.1 The Kurtosis and Skewness Minimization Criterion

We begin to introduce the kurtosis and skewness minimization or zeroing criterion suggested in [16] for model selection of the Gaussian mixture modeling on a given dataset. For a univariate Gaussian, both the sample kurtosis and skewness tends to be zero as the number of samples tends to infinity according to the theory of large samples. For a multivariate Gaussian, the samples projected onto any direction are also subject to a univariate Gaussian. Then, we can compute the sample kurtosis and sample skewness for the multivariate Gaussian along some projection directions. In this way, the sample kurtosis and skewness can be also used to measure how well the Gaussian distribution fits the sample data. For the Gaussian mixture modeling, as each component is expected to subject to a Gaussian distribution, we can compute its sample kurtosis and skewness with the samples it occupies and check whether it is small enough. Thus, the sample kurtosis and skewness can always offer us information to decide whether we have got the right number of Gaussians as well as the good estimation of other parameters in the Gaussian mixture model.

To implement the kurtosis and skewness minimization criterion, we begin to make the singular value decomposition for the covariance matrix of each Gaussian and select the first F eigenvectors to be the projection directions. Here, F should be selected such that the projected data of the samples on the F eigenvectors can contain the information of the high dimensional samples as much as possible and the noise within the samples as little as possible. For the mixture model of k Gaussians, we can classify the samples into k Gaussians (i.e., clusters) according to the maximum posterior $p(l \mid x_i)$, that is, x_i belongs to the $l-th$ Gaussian if

$$l = \arg\max_{1 \le j \le k} p(j \mid x_i). \quad (1)$$

Let $X^l = \{x_1^l, ..., x_m^l\}$ be the samples belonging to Gaussian l, $v_1^l, ..., v_F^l$ are the first F eigenvectors of the covariance matrix Σ_l. We can define the sample kurtosis of the Gaussian l along the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

direction v_f^l, named ku_f^l, as follows:

$$ku_f^l = \frac{1}{m}\sum_{i=1}^{m}(\frac{y_{fi}^l - m_f^l}{\sigma_f^l})^4 - 3, \quad (2)$$

where y_{fi}^l are the projected data of x_i^l on the direction of v_f^l, m_f^l and σ_f^l are the mean and standard deviation of the projected data $Y_f^l = \{y_{f1}^l, ..., y_{fm}^l\}$.

In this way, the sample kurtosis of Gaussian l can be defined by

$$ku_l = \sum_{f=1}^{F} abs(ku_f^l). \quad (3)$$

In a similar way, we can define the sample skewness of the Gaussian l along the direction v_f^l, named sk_f^l, as follows:

$$sk_f^l = \frac{1}{m}\sum_{i=1}^{m}(\frac{y_{fi}^l - m_f^l}{\sigma_f^l})^3, \quad (4)$$

and the sample skewnes of Gaussian l by:

$$sk_l = \sum_{f=1}^{F} abs(sk_f^l). \quad (5)$$

Combining the sample kurtosis and skewness of the $l-th$ Gaussian together, we have

$$s_l = ku_l + sk_l. \quad (6)$$

If Gaussian l fits the sample data in its vicinity, the sample kurtosis and skewness sum s_l of the $l-th$ Gaussian should be approximately zero as the number of samples is large enough. In order to test how well the Gaussian mixture model fits all the samples of k Gaussians, we can compute the weighted average of these individual sample kurtosis and skewnesses:

$$S_T = \sum_{l=1}^{k} \pi_l s_l. \quad (7)$$

According to the above definitions, it is clear that the total sample kurtosis and skewness sum S_T can be regarded as a measure on how well a Gaussian mixture fits the sample data. Since a small value of S_T indicates that every Gaussian in the mixture fits the samples well in its vicinity, that is, the Gaussian mixture is a good approximation of the latent mixture distribution. Otherwise, if the value of S_T is large, the samples belonging to certain Gaussians do not really come from a Gaussian distribution so that we should adjust the number of Gaussians or the parameters of the Gaussians. In any way, the true Gaussian mixture model of the sample data leads to the minimization or zeroing of the sample kurtosis and skewness sum, which can serve as a kurtosis and skewness minimization criterion for model selection of the Gaussian mixture modeling.

2.2 The Kurtosis and Skewness Minimization Learning Algorithm

Based on the kurtosis and skewness minimization criterion, we can construct the split-and-merge EM algorithm for the Gaussian mixture modeling with automated model selection. For short, we refer to it as the kurtosis and skewness minimization learning algorithm. Actually, it implements the usual or conventional EM algorithm for parameter estimation and adjusts the number of Gaussians by splitting or merging some unsuitable Gaussians being checked by the kurtosis and skewness measure. The main idea of the proposed algorithm is that, in each iteration, we try to maximize the likelihood by the usual EM algorithm for parameter learning, and to minimize the total sample kurtosis and skewness sum for model selection with the split-and-merge mechanism.

We firstly initialize k with a suitable value which can be a predicted value of the number of actual Gaussians in the dataset. The parameters of k Gaussians in the mixture can be initialized with the k-means algorithm. We then perform the usual EM algorithm until convergence. Furthermore, we select certain most likely unsuitable Gaussians to split or merge. With the new Gaussians, we perform the usual EM algorithm again and again until the sample kurtosis and skewness sum arrives at the minimum value.

1. **The Merging Mechanism**. For any two Gaussians with mean μ_i, μ_j, covariance Σ_i, Σ_j, we define the degree of separation between them as follows (refer to [18]):

$$c_{i,j} = \left\|\mu_i - \mu_j\right\| \Big/ \sqrt{\max\{trace(\Sigma_i), trace(\Sigma_j)\}}. \quad (8)$$

Since $c_{i,j}$ denotes the degree of separation of the two Gaussians, the lower value of $c_{i,j}$ reflects the higher degree of overlap between the two Gaussians. We can choose the two Gaussians with the two lowest values of $c_{i,j}$ to be merged, and set

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the parameters of the new merged Gaussian r by the following rules:

$$\pi_r = \pi_i + \pi_j; \quad (9)$$

$$\mu_r = (\pi_i\mu_i + \pi_j\mu_j)/\pi_r; \quad (10)$$

$$\Sigma_r = (\pi_i\Sigma_i + \pi_j\Sigma_j)/\pi_r. \quad (11)$$

2. **The Split Mechanism**. In order to decrease the sample kurtosis and skewness sum, the Gaussian which contributes most significantly to the high value of the sum, that is, the Gaussian with the maximum value of $\pi_l(ku_l + sk_l)$, should be split. Actually, we can divide such a Gaussian into two Gaussians i' and j' with their parameters designed as follows (refer to [19]).

For convenience, we decompose the covariance matrix Σ_k by $\Sigma_k = USV^T$, where $S = diag[s_1, s_2, ..., s_d]$ is a diagonal matrix with nonnegative diagonal elements in a descent order, U and V are two (standard) orthogonal matrices. We then set $A = U\sqrt{S} = Udiag[\sqrt{s_1}, \sqrt{s_2}, ..., \sqrt{s_d}]$ and get the first column A_1 of A. In this way, the parameters for the two split Gaussians can be given as follows, where γ, μ, β are all set to be 0.5.

$$\pi_{i'} = \gamma\pi_k, \pi_{j'} = (1-\gamma)\pi_k; \quad (12)$$

$$\mu_{i'} = \mu_k - (\pi_{j'}/\pi_{i'})^{1/2}\mu A_1; \quad (13)$$

$$\mu_{j'} = \mu_k + (\pi_{i'}/\pi_{j'})^{1/2}\mu A_1; \quad (14)$$

$$\Sigma_{i'} = (\pi_{j'}/\pi_{i'})\Sigma_k + ((\beta - \beta\mu^2 - 1)(\pi_k/\pi_{i'}) + 1)A_1A_1^T; \quad (15)$$

$$\Sigma_{j'} = (\pi_{i'}/\pi_{j'})\Sigma_k + ((\beta\mu^2 - \beta - \mu^2)(\pi_k/\pi_{j'}) + 1)A_1A_1^T \quad (16)$$

3. **The Procedure of the Proposed Algorithm**. In each iteration, if k is too large, there may be two Gaussians overlapped strongly so that it is likely to fit the samples of one Gaussian in its vicinity with two estimated Gaussians. So, we need to merge these two estimated Gaussians into one. Actually, if the value of $c_{i,j}$ of two Gaussians is less than a threshold value, we just merge the two Gaussians and accept the merge result. As for the spilt operation, if the value of $c_{i,j}$ of the two new Gaussians is less than the threshold value, we will cancel this split operation and find out the next candidate Gaussian to be split.

The number F of project directions is generally set to be slightly greater than half of the sample dimensionality. The termination condition of the algorithm is that either the merge or split operation do not decrease the sample kurtosis and skewness sum any more.

With the preparations all above, we can present the procedure of the kurtosis and skewness minimization learning algorithm (as a split-and-merge EM algorithm with the kurtosis and skewness minimization criterion) as follows:

Step 1. Initialization: set the initial number of k as a reasonable prediction value, and then set the initial parameters Θ_k of the Gaussian mixture via the k-means algorithm.

Step 2. Implement the usual EM algorithm until convergence and compute S_T.

Step 3. Merging Operation:

(a) Merge the two Gaussians i and j with the least $c_{i,j}$ into one Gaussian according to Eqs. (9) - (11).

(b) Implement the usual EM algorithm to obtain the updated parameters Θ_{merge}.

(c) If $c_{i,j} < c_{threshold}$, set $k = k - 1$, $\Theta_k = \Theta_{merge}$, return to Step 3(a); otherwise, compute S_T^{merge}.

Step 4. Split Operation:

(a) Set S_T^{split} a large enough number.

(b) Sort these $\pi_l(ku_l + sk_l)$ in the descending order, and set $p = 1$.

(c) Split the Gaussian with the $p-th$ largest value of $\pi_l(ku_l + sk_l)$ into two new Gaussians i', j' according to Eqs. (12) - (16).

(d) Implement the usual EM algorithm to obtain the updated parameters Θ_{split}.

(e) If $c_{i',j'} > c_{threshold}$, compute S_T^{split}, otherwise, refuse the split result and set $p = p + 1$, if $p \le k$, return to Step 4(c) otherwise go to step 5;

Step 5. Compare the three values of S_T, S_T^{merge} and S_T^{split}:

(i). If $S_T^{split} = \min(S_T, S_T^{merge}, S_T^{split})$, we accept the result of the split operation and set $k = k + 1, \Theta_k = \Theta_{split}$, go to Step 2;

(ii). If $S_T^{merge} = \min(S_T, S_T^{merge}, S_T^{split})$, we accept the result of the merge operation and set $k = k - 1, \Theta_k = \Theta_{merge}$, go to Step 2;

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(iii). If $S_T = \min(S_T, S_T^{merge}, S_T^{split})$, we stop the algorithm with the current Θ_k as the final result of the algorithm.

3. TRAINING OF THE GENERALIZED RBF NETWORK

3.1 The Generalized RBF Network

We begin to introduce the structure of RBF network. The RBF network is a two-layer forward neural network and its outputs are given by

$$y_l(x) = \sum_{j=1}^{k} \omega_{jl} R_j(x), \quad l = 1, ..., m, \tag{17}$$

where k is the number of hidden units or RBF's, ω_{jl} is the weight on the connection from the $j-th$ hidden unit to the $l-th$ output unit. $R_j(x)$ is the $j-th$ Radial Basis Function (RBF) of the input x serving as the output of the $j-th$ hidden unit. When the RBF network has only one single output unit, the output function of the RBF network becomes the following simple version:

$$y(x) = \sum_{j=1}^{k} \omega_j R_j(x). \tag{18}$$

Here, we just use this simple structure of the RBF network for time series prediction.

Actually, the most commonly used radial basis function is Gaussian kernel function given by:

$$R_j(x) = \phi(\| x - \mu_j \|) = \exp\{-\frac{\| x - \mu_j \|^2}{2\sigma_j^2}\}, \tag{19}$$

where μ_j, σ_j are the center and scale of the Gaussian RBF $R_j(x)$, respectively.

For the generalized RBF network, the Gaussian density function is adopted as the radial basis function, instead of Gaussian kernel function. In fact, the Gaussian density function is given by

$$R_j(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)\}, \tag{20}$$

where μ_j, Σ_j are the mean and covariance matrix of the Gaussian distribution.

The generalized RBF network can make better use of the statistic properties of the input data which may be or approximately be distributed as a Gaussian mixture. So, it has more extensive adaptability. Therefore, we can directly utilize the kurtosis and skewness minimization learning algorithm on the input data to determine the number of hidden units and to set the initial mean and covariance matrix of each Gaussian density function.

3.2 The Least Mean Square (LMS) Algorithm

After determining the number of the hidden units and setting the initial value of the parameters of the generalized RBF network, we can implement the Least Mean Square (LMS) algorithm to learn the parameters of the generalized RBF network ω_j, μ_j Σ_j for $j = 1, 2, ..., k$.

In fact, the mean square error of the generalized RBF network on a sample data set $S = \{(x_t, \bar{y}_t)\}_{t=1}^{N}$ can be given as follows:

$$E = \frac{1}{2}\sum_{t=1}^{N} (\hat{y}_t - y_t)^2 = \frac{1}{2}\sum_{t=1}^{N} (\hat{y}_t - \sum_{j=1}^{k} \omega_j R_j(x_t))^2, \tag{21}$$

where y_t is the output of the network with input x_t. Then, we have the derivatives of E with respect to ω_j, μ_j and Σ_j respectively, as follows:

$$\frac{\partial E}{\partial \omega_j} = \sum_{t=1}^{N} (\hat{y}_t - y_t) R_j(x_t); \tag{22}$$

$$\frac{\partial E}{\partial \mu_j} = \sum_{t=1}^{N} (\hat{y}_t - y_t) \omega_j \frac{\partial R_j(x_t)}{\partial \mu_j}; \tag{23}$$

$$\frac{\partial E}{\partial \Sigma_j} = \frac{1}{2}\sum_{t=1}^{N} (\hat{y}_t - y_t) \omega_j \frac{\partial R_j(x_t)}{\partial \Sigma_j}. \tag{24}$$

According to the above derivatives as well as the further derivatives on $R_j(x)$ and using the gradient descent method, we have the LMS learning rules on the parameters ω_j, μ_j of the generalized RBF network as follows:

$$\Delta\omega_j = \eta \sum_{t=1}^{N} (\hat{y}_t - y_t) R_j(x_t); \tag{25}$$

and

$$\Delta\mu_j = \eta \sum_{t=1}^{N} (\hat{y}_t - y_t) \omega_j R_j(x_t) \Sigma_j^{-1} (x_t - \mu_j), \tag{26}$$

where $\eta > 0$ is the learning rate.

As for the covariance matrix Σ_j, since it is constrained to be positive definitiveness, we can decompose it by

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\Sigma_j = B_j B_j^T, \tag{27}$$

and then the LMS learning rule of B_j can be given by

$$\Delta \mathrm{vec}[B_j] = \frac{\eta}{2}\sum_{t=1}^{N}(\hat{y}_t - y_t)\omega_j R_j(x_t)\frac{\partial (B_j B_j^T)}{\partial B_j} \cdot \mathrm{vec}[\Sigma_j^{-1}(x_t-\mu_j)(x_t-\mu_j)^T\Sigma_j^{-1} - \Sigma_j^{-1}] \tag{28}$$

where $\mathrm{vec}[A]$ denotes the connect column vectors in order, that is, if

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \cdots & \cdots & \cdots \\ a_{d1} & \cdots & a_{dd} \end{pmatrix}, \tag{29}$$

we have

$$\mathrm{vec}[A] = (a_{11},...,a_{d1},...,a_{1d},...,a_{dd})^T. \tag{30}$$

And

$$\frac{\partial BB^T}{\partial B} = I_{d\times d}\otimes B_{d\times d}^T + E_{d^2\times d^2}\cdot B_{d\times d}^T \otimes I_{d\times d} \tag{31}$$

where $\otimes$ denotes the Kronecker product, and

$$E_{d^2\times d^2} = \frac{\partial B^T}{\partial B} = (\Gamma_{ij})_{d^2\times d^2} = \begin{pmatrix} \Gamma_{11} & \Gamma_{12} & \cdots & \Gamma_{1d} \\ \Gamma_{21} & \Gamma_{22} & \cdots & \Gamma_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ \Gamma_{d1} & \Gamma_{d2} & \cdots & \Gamma_{dd} \end{pmatrix} \tag{32}$$

where Γ_{ij} is a matrix of $d\times d$, where the $(j,i)^{th}$ is 1, others are 0.

Summing up Equations (25), (26), (27) and (28) together, we have the LMS learning rules of the parameters ω_j, μ_j and Σ_j. Actually, we will implement it to learn the final parameters of the generalized RBF network for time series prediction in next section.

4. EXPERIMENTAL RESULTS

In this section, we implement the kurtosis and skewness minimization learning algorithm as well as the LMS algorithm on the training of the generalized RBF network for nonlinear time series prediction. Actually, two famous chaos time series, Mackey-Glass time series and Rossler time series, are used for test. For comparison, we also use the BP network and the conventional RBF network for the same prediction problem on the two time series. Here, we use the root mean square error (RMSE) as the index of the prediction accuracy.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[\hat{y}_i - f(x_i)]^2} \tag{33}$$

where $\hat{y}_i$ is the expected output, and $f(x_i)$ the actual output of the network, for input x_i.

4.1 On the Mackey-Glass Time Series Prediction

We firstly train the generalized RBF network with the kurtosis and skewness minimization learning algorithm and the LMS algorithm for the Mackey-Glass time series prediction. As shown in Fig. 1, a piece of the Mackey-Glass time series is generated via the delay differential equation:

$$\dot{x} = \frac{ax(t-\tau)}{1+x(t-\tau)^c} - bx(t), \tag{34}$$

where $a = 0.2$, $b = 0.1$, $\tau = 17$. Particularly, 1000 sample data are generated to form pieces of time series as $\{x(t-18), x(t-12), x(t-6), x(t), x(t+6)\}$, $118 \le t \le 1117$, where the first four data of each sample are considered as an input data of the generalized RBF network, while the last one is considered as the expected output of the generalized RBF network. Mathematically, the expected output $\hat{y}_i = x(t+6)$, the input sample $x_i = \{x(t-18), x(t-12), x(t-6), x(t)\}$. In the experiment, we divide these 1000 sample data into two sets: the training and test sets with the preceding and remaining 500 sample data, respectively.

We implement the kurtosis and skewness minimization learning algorithm to design the generalized RBF network and set the initial parameters for the prediction of the Mackey-Glass time series. Actually, the number of hidden units is determined to be 8. The LMS algorithm is further implemented to obtain the final parameters of the network and the prediction result on the test data is given in Fig. 2, with the prediction mean square error 0.0044, which is the lowest prediction error among the four neural networks on the Mackey-Glass time series (see Table I for details).

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

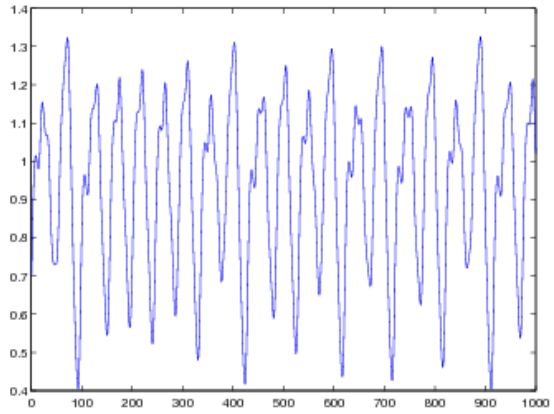

Fig. 1 The Sketch Of The Mackey-Glass Time Series.

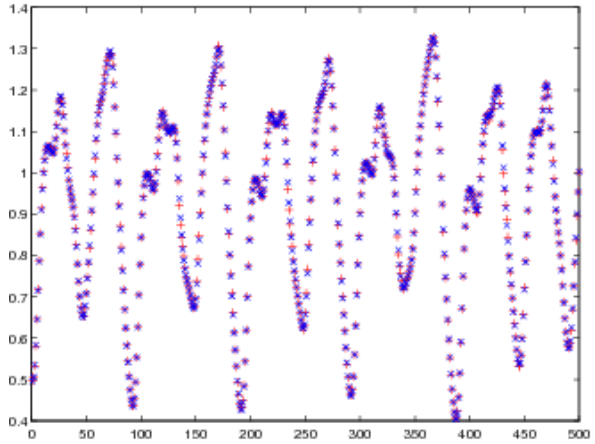

Fig. 2 The Prediction Result Of The Generalized RBF Network With The Kurtosis And Skewness Minimization Learning Algorithm And The LMS Algorithm For The Mackey-Glass Time Series, Where + Represents The Sample Datum, While × Represents The Prediction Datum.

4.2 On the Rossler Time Series

We further train the generalized RBF network with the kurtosis and skewness minimization learning algorithm and the LMS algorithm for the Rossler time series prediction. As shown in Fig. 3, a piece of the Rossler time series is generated via the differential equations:

$$\begin{cases} \dot{x} = -y - z \\ \dot{y} = x + ay \\ \dot{z} = b + z(x - c) \end{cases} \quad (35)$$

where $a = 0.2$, $b = 0.1$, $c = 5.7$. Particularly, 1000 sample data are generated to form pieces of time series as $\{x(t-18), x(t-12), x(t-6), x(t), x(t+6)\}$, $318 \le t \le 1317$, where the first four data of each sample are considered as an input data of the generalized RBF network, while the last one is considered as the expected output of the generalized RBF network. Mathematically, the expected output $\hat{y}_i = x(t+6)$, the input sample $x_i = \{x(t-18), x(t-12), x(t-6), x(t)\}$. In the experiment, we divide these 1000 sample data into two sets: the training and test sets with the preceding and remaining 500 sample data, respectively. For convenience, we can preprocess these Rossler time series data so that the value scope of $x(t)$ is $[0,1.6]$ according to the transformation:

$$x(t)' = (x(t) + 10)/15 \quad (36)$$

Table I The Prediction Accuracies Of Four Neural Networks On The Mackey-Glass Time Series.

The Neural Network	RMSE
The BP network	0.0109
The RBF (with 10 hidden units)	0.0198
The RBF (with 23 hidden units)	0.0107
The Generalized RBF	0.0044

Table II The Prediction Accuracies Of Four Neural Networks On The Rossler Time Series.

The Neural Network	RMSE
The BP network	0.0049
The RBF (with 10 hidden units)	0.0568
The RBF (with 23 hidden units)	0.0146
The Generalized RBF	0.0040

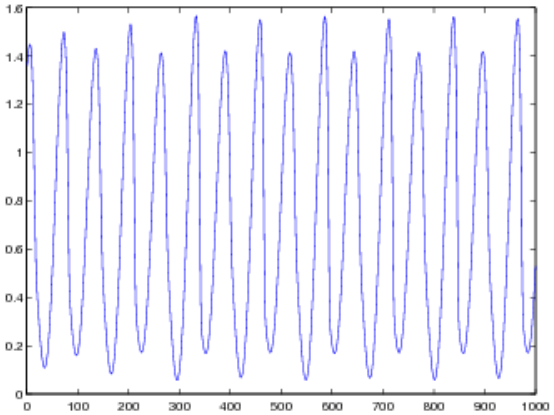

Fig. 3 The sketch of the Rossler time series.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

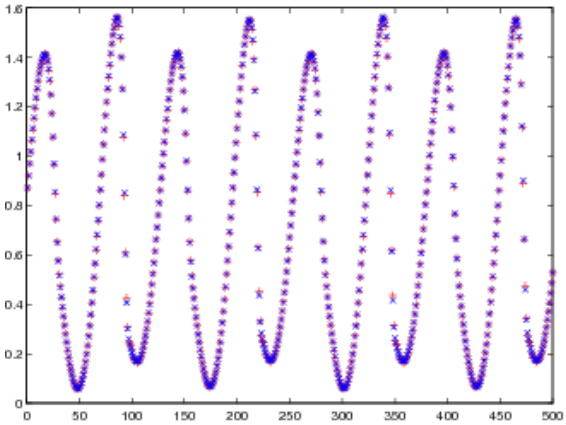

Fig. 4 The Prediction Result Of The Generalized RBF Network With The Kurtosis And Skewness Minimization Learning Algorithm And The LMS Algorithm For The Rossler Time Series, Where + Represents The Ample Datum, × Represents The Prediction Datum.

In the same way, we implement the kurtosis and skewness minimization learning algorithm to design the generalized RBF network and set the initial parameters for the prediction of the Rossler time series. Actually, the number of hidden units is determined to be 9. We further implement the LMS algorithm to obtain the final parameters of the network. The prediction result of the generalized RBF network on the test data is given in Fig. 4, with the prediction mean square error 0.0040, which is the lowest prediction error among the four neural networks on the Rossler time series (see Table II for details).

Based on the experimental results on two typical chaos time series, we have found that the generalized RBF network with the kurtosis and skewness minimization learning algorithm really improves the prediction accuracy in comparison with the BP network and the conventional RBF networks.

5. CONCLUSIONS

We have investigated the training of the RBF network from the input dataset and proposed the kurtosis and skewness minimization learning algorithm for determining the number of hidden units and setting the initial values of the parameters in the generalized RBF network. Actually, the proposed kurtosis and skewness minimization learning algorithm is based on the newly established kurtosis and skewness minimization criterion and operates in a way of the split-and merge EM algorithm. When the structure and initial parameters of the generalized RBF network are obtained, the least mean square algorithm can be implemented to get the final parameters. It is demonstrated by the experiments on the two typical chaos time series that the generalized RBF network with the kurtosis and skewness minimization learning algorithm really improves the prediction accuracy and outperforms the BP network and the conventional RBF networks.

ACKNOWLEDGMENT

This work was supported by the Ph.D. Programs Foundation of Ministry of Education of China for grant 20100001110006.

REFRENCES:

[1] J. Moody and C. Darken, "Fast learning in networks of locally tuned processing units", *Neural Computation*, Vol. 1, 1989, pp. 281-294.

[2] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multiplayer networks", *Science*, Vol. 247, 1990, pp. 978-982.

[3] J. Platt, "A resource-allocating network for function interpolation", *Neural Computation*, Vol. 3, 1991, pp.213-225.

[4] A. Esposito, M. Marinaro, D. Oricchio and S. Scarpetta, "Approximation of continuous and discontinuous mapping by a growing neural RBF-based algorithm", *Neural Networks*, Vol. 13, 2000, pp. 651-665.

[5] A.V.D. Sanchez, "Searching for a solution to the automatic RBF network design problem", *Neurocomputing*, Vol. 42, 2002, pp. 147-170.

[6] L. Xu, A. Krzyzak and E. Oja, "Rival penalized competitive learning for clustering analysis, RBF net and curve detection", *IEEE Trans. on Neural Networks*, Vol. 4, 1993, pp. 636-649.

[7] J. Ma and T. Wang, "A cost-function approach to rival penalized competitive learning (RPCL)", *IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics*, Vol. 36, No. 4, 2006, pp. 722-737.

[8] L. Xu, "Ying-Yang machine: a Bayesian Kullback scheme for unified learning and new results on vector quantization", *Proceedings of International Conference on Neural Information Processing (ICONIP95)*, Vol. 2, 1995, pp. 977-988.

[9] L. Xu, "Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

three-layer nets and ME-RBF-SVM models", *International Journal of Neural Systems*, Vol. 11, No. 1, 2001, pp. 43-69.

[10] J. Ma, T. Wang, and L. Xu, "A gradient BYY harmony learning rule on Gaussian mixture with automated model selection", *Neurocomputing*, Vol. 56, No. 1, 2004, pp. 481-487.

[11] J. Ma and J. Liu, "The BYY annealing learning algorithm for Gaussian mixture with automated model selection", *Pattern Recognition*, Vol. 40, 2007, pp. 2029-2037.

[12] J. Ma and L. Wang, "BYY harmony learning on finite mixture: adaptive gradient implementation and a floating RPCL mechanism", *Neural Processing Letters*, Vol. 24, 2006, pp. 19-40.

[13] E. S. Chuq, S. Chen B. Mulgrew, "Gradient radial basis function networks for nonlinear and nonstationary time series prediction", *Neural Networks*, Vol. 7, No. 1, 1996, pp. 190-194.

[14] S.A. Billings and X. Hong, "Dual-orthogonal radial basis function networks for nonlinear time series prediction", *Neural Networks*, Vol. 11, 1989, pp. 479-493.

[15] K. Huang, L. Wang, and J. Ma, "Efficient training of RBF networks via the BYY automated model selection learning algorithms", , *Lecture Notes in Computer Science*, Vol. 4491, 2007, pp. 1183-1192.

[16] L. Wang and J. Ma, "A kurtosis and skewness based criterion for model selection on Gaussian mixture", *Proceedings of the 2nd International Conference on Biomedical Engineering and Informatics (BMEI, 2009)*,17-19 October 2009, Tianjin, China.

[17] L. Wang and J. Ma, "An efficient greedy EM algorithm for Gaussian mixture for adaptive model selection using the kurtosis and skewness criterion", *Advanced Materials Research*, Vol. 452-453, 2012, pp. 1501-1506.

[18] N. Vlassis and A. Likas, "A greedy EM algorithm for Gaussian mixture learning", *Neural Processing Letters*, Vol. 15, 2002, pp. 77-87.

[19] Z. Zhang, C. Chen and J. Sun, "EM algorithm for Gaussian mixtures with split-and-merge operation", *Pattern Recognition*, Vol. 36, 2003, pp. 1973-1983.