Neural Network Prediction Method of Chaotic FH Code Sequence

Xiang Shen
Chengdu Vocational & Technical College of Industry, Chengdu 610218, China
112078295@qq.com

Keywords: Chaotic Frequency Hopping Codes; RBF Neural Network; Prediction.

Abstract. The frequency hopping communication implements the communication in the process of continuous and irregular jumping frequency, widely used in military communications and civilian communications. At present, the research on the modeling and prediction of frequency hopping codes by using chaos analysis and neural network and so on algorithms is very closely developed and has achieved some effects. In this paper, the RBF neural network was used to conduct simulation prediction of the m code, the RS code, and the nonlinear code these three frequency hopping codes, and the simulation experiments were carried out by MATLAB. The performance of the prediction model was analyzed and compared by theoretical analysis and simulation results. The results showed that the RBF neural network was more powerful in approximation ability, classification learning, learning speed and so on aspects.

1. Introduction

The basic problem of prediction is to discover and infer its future according to the law of development of things. In order to explore this rule, the most commonly used method is to create a dynamic mathematical method of the description system [1, 2]. After years of efforts, we have made a deep and detailed research on the prediction model and prediction method of time series, and obtained a lot of theoretical achievements and practical application results. There are many kinds of forecasting methods, which can be divided into two categories according to methodology: point prediction and interval prediction. Point prediction methods can be divided into three parts: nonlinear adaptive prediction method, global prediction method and local prediction method. Local prediction can be divided into two parts: local linear prediction method and local nonlinear prediction method. The global prediction method can be divided into two parts: Global polynomial modeling prediction method and neural network modeling and prediction method. Nonlinear adaptive prediction method can be divided into nonlinear adaptive filtering prediction method based on nonlinear function transformation and adaptive polynomial filtering prediction method based on series expansion [3-5]. In this paper, we discussed the modeling prediction based on the RBF neural network.

2. 2. Methods

2.1 Basic principle and model structure of RBF neural network.

Radial basis function neural network (RBF) is a kind of feed-forward neural network, which has the excellent characteristics of structure adaptive determination and its output does not depend on the initial weight.

RBF neural network has good versatility. Hartman has proved that as long as there are enough hidden layer neurons in RBF neural network, RBF network can approximate any continuous function with any precision [6,7]. Usually, RBF neurons only have partial responses on the input stimuli, that is to say, only when an input falls in the local area of the input space, they can produce an essential non-zero value response. The training speed of RBF neural network is very fast, and it will not shake in training, neither fall into local minimum.

RBF neural network consists of three layers. The first layer is the input layer, which is mainly composed of the nodes of the signal source; the second layer is the hidden layer, in which the transformation function is a locally distributed non-negative nonlinear function. It is decreasing and the center points are radially symmetrical. The third layer is the output layer, and the output of the network is a linear weighted of the hidden unit output. The transformation of the RBF neural network from the input layer space to the hidden layer space is nonlinear, but the transformation from the hidden layer space to the output layer space is linear. RBF neural network has local approximation ability. Its topology is shown in Figure 1.

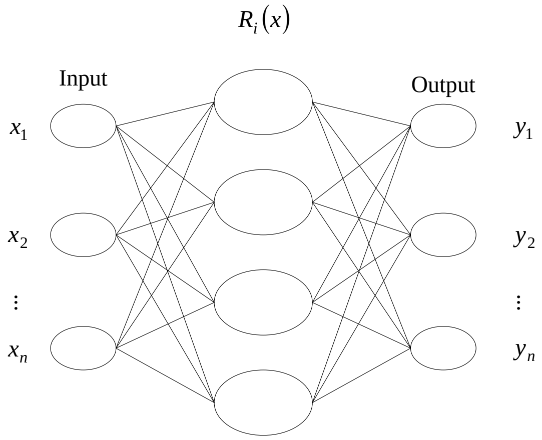

Figure 1 Schematic diagram of RBF neural network structure

2.2 Algorithms of RBF neural network.

1)The transfer function of RBF neural network

The transfer functions of commonly used RBF (namely radial basis function) include the following forms:

$$f(x) = \exp^{-(x/\sigma)^2} \quad (1)$$

$$f(x) = 1/\left(\sigma^2 + x^2\right)^{\alpha}, \alpha > 0 \quad (2)$$

$$f(x) = \left(\alpha^2 + x^2\right)^{\beta}, \alpha < \beta < 1 \quad (3)$$

The above functions are radially symmetrical, but the one with the widest application is Gauss function:

$$R_i(x) = \exp\left(-\|x - c_i\|^2 / 2\sigma_i^2\right), i = 1,2,\cdots,m \quad (4)$$

In (4): x is n dimension input vector, c_i indicates the center of the i-th basis function, having same dimension vector as x, σ_i is the i-th variable perceived, and m refers to the number of perceived units. $\|x - c_i\|$ suggests the norm of vector $x - c_i$, generally representing the distance between x and c_i. In c_i, $R_i(x)$ has a maximum value, which is also the only one. With the increase of $\|x - c_i\|$, $R_i(x)$ rapidly decreases to zero.

RBF algorithm chooses the Gauss function as the transfer function of the hidden layer. The nonlinear mapping from $X \to R_i(x)$ is implemented by the hidden layer, while the linear mapping from $R_i(x) \to y_k$ is realized by the output layer. Set $X = (x_1, x_2, \cdots, x_f, \cdots, x_n)$ to be the input of the input layer, and $Y = (y_1, y_2, \cdots, y_k, \cdots, y_p)$ for the actual output, then the output of the k-th neuron in the output layer is:

$$\hat{y}_k = \sum_{i=1}^{m} \omega_{ik} R_i(X), k = 1,2,\cdots, p \quad (5)$$

In (5), n represents the number of nodes of the input layer, m indicates the number of nodes of the hidden layer, p refers to the number of nodes of the output layer, ω_{ik} suggests the connection weight between the i-th neuron in the hidden layer and the the k-th neuron in the output layer, and $R_i(X)$ is the transfer function of the i-th neuron in the hidden layer. As a result, when the cluster center c_i and the weight ω_{ik} are determined, then we can calculate the output value that corresponds to a given input.

2)The choice of RBF center

How to choose the RBF center is the key for RBF algorithm. In general, there are following several methods:

(1) Random choice method

This method is the simplest method and also a direct calculation method. In this method, the center of the transfer function of the hidden layer unit is randomly selected and center-determined in the input sample data. After determining the center of RBF, then conduct the calculation of variance. If both of the center and variance are determined, the output of the hidden layer unit is known, so the connection weights of the network can be determined by solving the linear equations. In allusion to the given problems, if the distribution of the sample data is representative, then this method is a simple and feasible method.

(2) Self-organizing learning selection method

In this method, the center of RBF is uncertain, and it can be moved. Its location is determined by self-organizing learning. The linear weights of the output layer are calculated by supervised learning rules. As a result, this method is a hybrid learning method. The process of self-organizing learning is the allocation of network resources, and the purpose of learning is to make the center of RBF belong to the important area of the input space. This method is simple in process, fast in speed and suitable for application, with good approximation performance.

(3) Supervised learning selection method

In this method, the center of RBF is determined by supervised learning, which is the most general form of RBF neural network learning. Among them, the supervised learning selection method makes use of the gradient descent method. The requirement of this method for network learning is to optimize the free parameters and weights of the network, so the error objective function can be minimized. By using the gradient descent method, we can solve the optimization problems, and get the optimization formula of network parameters. In the recursive method, the initial value is very important. In order to reduce the possibility that the learning process converges to a local minimum, it is necessary to make a wide range search of parameters in the effective region of the parameter space. For achieving this effect, we are supposed to first of all use RBF neural network algorithm to achieve a regular Gauss classification algorithm, and then use the results of the classification as the starting point of the search.

(4) Orthogonal least squares method

Orthogonal least squares method is also an important learning method of RBF neural network, whose source is linear regression model. The basic idea is to use the regression model to express the input and output relations of the network, and to analyze the contribution to the reduction of variance by orthogonal regression operator. Study and select the appropriate regression vector and the regression operator number, so as to make the network output able to meet the performance requirements. In addition, the RBF function directly forms the regression operator, so once the regression operator is determined, the parameters of the RBF function can be determined.

3) Description for RBF algorithm

In the following, we will analyze a RBF algorithm, whose RBF center is based on self-organizing learning. The method is divided into two steps: the first step is the unsupervised self-organizing learning stage, that is, the stage to study base function of the center and the variance of the hidden layer, and also the stage for determining the weights between the training input layer and the hidden layer. The second step is the supervised learning stage, which is the stage for determining the weights between the training hidden layer and the output layer. Generally speaking, before the training, the input vector X and the corresponding target vector T as well as the expansion constant C of the radial basis function. The purpose of the training is to solve the final weights and thresholds between the two layers.

The steps for RBF algorithm are shown as follows:

(1) Determine the center $T_k\left(k=1,2,\cdots,l\right)$ of learning

In the process of self-organizing learning, clustering algorithm is used, and the K- mean clustering algorithm is usually used. Assuming that there is one cluster center, set the center of basis function in the n-th iteration to be $T_k\left(n\right)\left(k=1,2,\cdots,l\right)$. The concrete steps of K- means clustering algorithm are as follows:

① Initialize the center of clustering. In general, set $T_k(0)$ to be the initial sample, and the iteration step for $n=0$;

② Randomly input the sample X_i of the training;

③ Find the closet center of the training sample X_i, that is to say, find $T_k(x_i)$ to make it meet:

$$K(X(i))=\min_k \|X(i)-T_k(n)\|, k=1,2,\cdots,l \quad (6)$$

In (6), $T_k(n)$ is the k-th center of the basis function in the n-th iteration.

① Adjust the center of the basis function with the following equation:

When $k=K(X(i))$: $T_k(n+1)=T_k(n)+a[X(i)-T_k(n)]$ (7)

Other cases: $T_k(n+1)=T_k(n)$ (8)

In (7), a refers to the learning step length, and $0<a<0$.

② Judge whether all the training samples are learned, and judge if the distribution of the center will not change. If yes, then end; otherwise, $n=n+1$ turns to ②.

At last, $T_k(k=1,2,\cdots,l)$ obtained is the final basis function center of RBF neural network.

(2) Determine the variance $\sigma_k(k=1,2,\cdots,l)$

After the center is learned, it is fixed. Then, it is necessary to determine the variance of the basis function. In this paper, RBF uses Gauss function, then we use the following formula to calculate the variance:

$$\sigma_1=\sigma_2=\cdots=\sigma_l=\frac{d_{\max}}{\sqrt{2l}} \quad (9)$$

l refers to the number of the hidden units, and $d_{\max}$ represents the maximum distance between the chosen centers.

(3) Learn the weight $W_{kj}(k=1,2,\cdots,l; j=1,2,\cdots,l)$

We can make use of LMS algorithm to learn the weight, and the steps are shown as follows:

① Set the variables and parameters. The input parameters are $X(n)=[x_1(n),x_2(n),\cdots,x_m(n)]$, also called the training sample. The weight vector is $W(n)=[w_1(n),w_2(n),\cdots,w_m(n)]$, the actual output is $Y(n)$, the expectation output is $d(n)$, the learning efficiency is η, and the iteration time is n.

② Initialize and give $W_j(0)$ with a random and smaller non-zero value, $n=0$.

③According to the input sample $X(n)=[x_1(n),x_2(n),\cdots,x_m(n)]$ and the corresponding expectation output *d*, calculate:

$$e(n)=d(n)-X^T(n)W(n) \quad (10)$$

$$W(n+1)=W(n)+\eta X(n)e(n) \quad (11)$$

④ Judge whether it meets the above conditions. If meets, the algorithm ends. Otherwise, increase 1 of n value, and turn to ③ for re-operation.

3. Results

M sequence, RS sequence and nonlinear sequence three sets of data use 200 state vectors as the input, and the corresponding 200 prediction values are used as the output. The multilayer perceptron prediction model is trained by RBF algorithm. The simulation results are shown in Figure 2, Figure 3 and Figure 4 (the value is represented by *, and the true value is represented by O).

Figure 2, Figure 3 and Figure 4 show the comparison between the predicted value and the true value of the m sequence, RS sequence and the nonlinear sequence, respectively. It can be seen that the prediction effect of the nonlinear sequence is not good, while other sequences have accurate prediction. This shows that although the RBF network is close to the other three sequences, it does not approach the function generating nonlinear sequence. This inspires us to recognize that there is no uniform model of nonlinear time series, and it is necessary to choose different models for different sequences. At the same time, the prediction results of RBF network show that if it is possible to choose different models according to the specific time sequence to achieve correct prediction results.

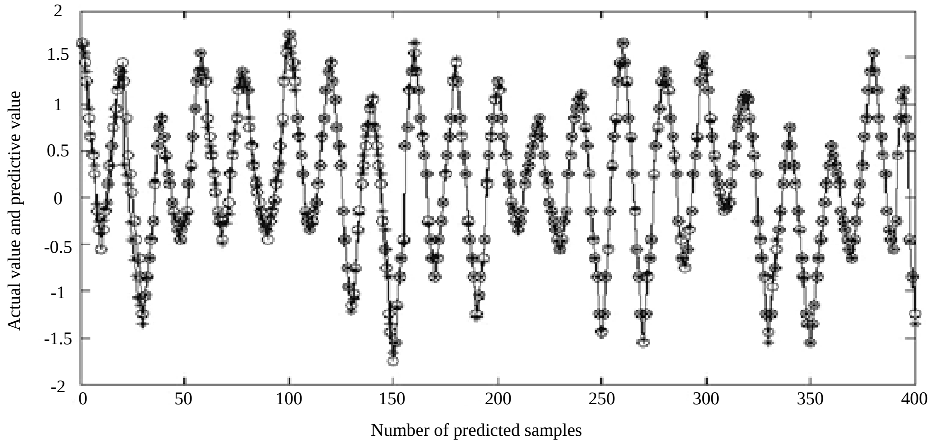

Figure 2 Comparison of the predicted value and the real value of m sequence

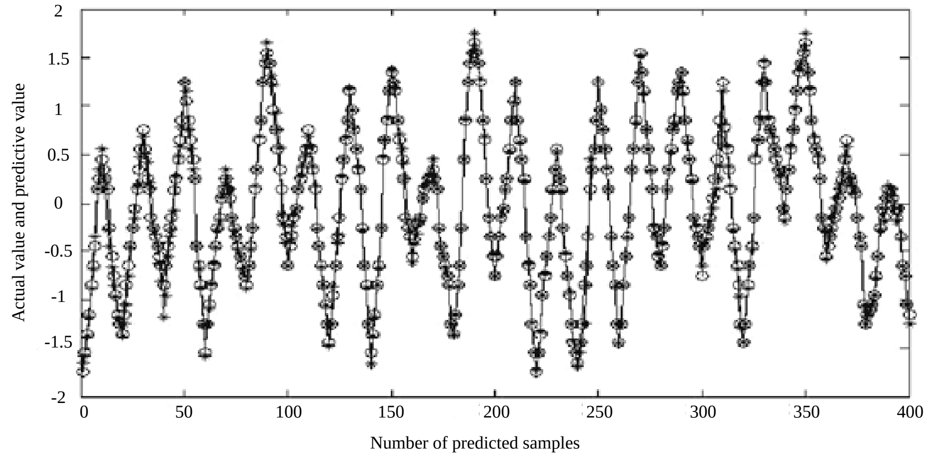

Figure 3 Comparison of the predicted value and the real value of RS sequence RS

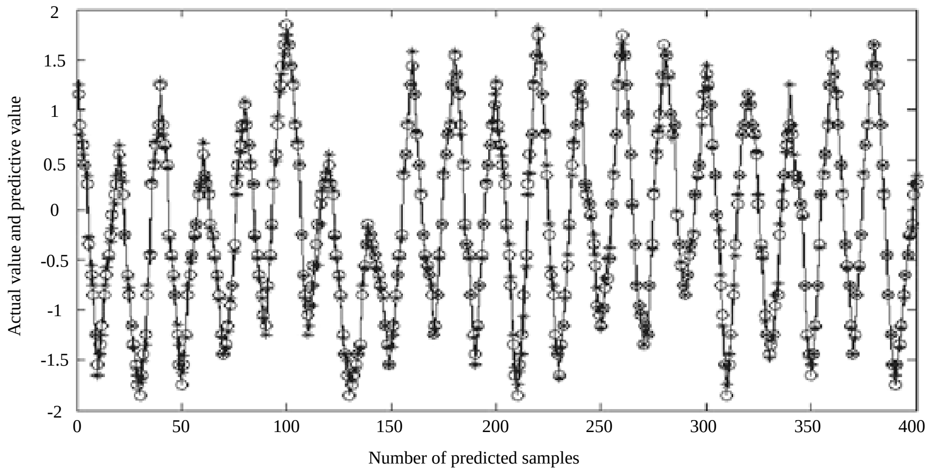

Figure 4 Comparison of the predicted value and the real value of nonlinear sequence

4. Conclusion

Frequency hopping communication is an important technology for wireless communication system to improve the anti-interference ability and anti-interception capability, which has important application value and broad development prospects in military communication, mobile communication, wireless LAN and so on. In this paper, we introduced the basic principle and model structure of RBF neural network, and under the background of RBF neural network algorithm, used MATLAB to realize the simulation experiment of RBF neural network prediction for common chaotic FH code (m code, RS code and nonlinear frequency hopping sequence). The simulation results showed that the RBF neural network could effectively predict the frequency hopping codes, and had high prediction speed and high prediction accuracy.

References

[1]. Wiggins, & Stephen. (2013). *Introduction to applied nonlinear dynamical systems and chaos* =. Springer-Verlag.
[2]. Broer, H. W. (2013). Nonlinear dynamical systems and chaos. *Progress in Nonlinear Differential Equations & Their Applications, 19*.
[3]. David G. Stork. (2011). Book review: "introduction to the theory of neural computation", john hertz, anders krogh, and richard g. palmer. *International Journal of Neural Systems, 2*(01n02), 157-158.
[4]. Carandini, M., & Heeger, D. J. (2012). Normalization as a canonical neural computation. *Nature Reviews Neuroscience, 13*(1), 51-62.
[5]. Howarth, C., Gleeson, P., & Attwell, D. (2012). Updated energy budgets for neural computation in the neocortex and cerebellum. *Journal of Cerebral Blood Flow & Metabolism Official Journal of the International Society of Cerebral Blood Flow & Metabolism, 32*(7), 1222-32.
[6]. Veeramuthuvel, P., Shankar, K., & Sairajan, K. K. (2015). Application of rbf neural network in prediction of particle damping parameters from experimental data. *Journal of Vibration & Control*.
[7]. Giantomassi, A., Ippoliti, G., Longhi, S., Bertini, I., & Pizzuti, S. (2011). On-line steam production prediction for a municipal solid waste incinerator by fully tuned minimal rbf neural networks. *Journal of Process Control, 21*(1), 164-172.