

Generalized Regression Neural Network Based Predictive Model of Nonlinear System

Yibin Song*, Zhenbin Du

School of Computer, Yantai University

Yantai, Shandong, 264005, China

*Corresponding author, e-mail: syb2874@163.com

Abstract

Generalized Regression Neural Network (GRNN) is usually applied to the Function approximation. This paper, based on the principle of GRNN, presents a method for the predictive model of nonlinear complex system. The presented algorithm is applied to the learning and predicting process for the system modeling. The simulations show the described method has good effects on predicting the dynamic process of the nonlinear model, and could be applied on the predictive control for nonlinear systems satisfactorily.

Keywords: predictive model, grnn, nonlinear system, function approximation

Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

It is known the Generalized Regression Neural Network (GRNN) is a much efficient method for fitting or approximating the output of complex plants. Usually, GRNN is considered to be made up of a layer of radial base network and a layer of linear network. It is known that Predictive Control is referred to as model-based control. It combines the control strategy with predictive model, optimizing algorithm and feedback adjusting. The predictive control method for linear systems had been resolved perfectly, but it is difficult for the control of nonlinear complex systems. The first key work performed is to realize predictive model accurately for the nonlinear dynamic process. On the basis of the active I/O information, the GRNN-based predictive model could be used to describe the dynamic action of nonlinear systems and to predict the intending outputs of system. The I/O information is to be the base of predictive control. So, setting the predictive model is the precondition for realizing predictive control of complex systems. For the structure-known linear models, there are many known methods for getting predictive information. However, it is difficult to obtaining predicting information from the structure-unknown and definition-deficiently nonlinear complex systems [1-3]. So, the predictive control for the system will be difficult to realize usually. The approximating ability of GRNN, as we known, will makes it possible to get the predictive model for the complex systems quickly and accurately.

On the other hand, GRNN is similar as multi-layer BP-NN in network structure. The BP-NN needs usually large amounts of calculation of linking weights for getting the global approach. The main characteristics of GRNN are its output has the local linear relation to the parameters. It is a typical local approaching network. So, the learning speed of GRNN is more quickly than that of BP-NN, especially for the function approaching and model prediction for structure-unknown objects. It is a powerful tool for realizing the predictive control of the complex objects or systems.

Based on the principle of GRNN, this paper presents a method for getting the predictive model of nonlinear complex system. The described algorithm is applied to the predicting process of a nonlinear model. The simulations show that GRNN-based predictive model has good effects on getting the predictive model of complex systems, and could be applied on the prediction control for nonlinear systems satisfactorily.

2. GRNN-Based Predictive Control

2.1. GRNN Structure Analysis

The typical structure of GRNN is shown as Figure 1. It is often used in function approximation for complex models. It consists of a layer of radial base network and a layer of linear network normally. Direct mappings are adopted by GRNN between input layer and hidden layer. But the mapping between hidden layer and output layer adopts the weighted linearly sum of hidden layers as the mapping mode (shown as Figure 1). In Figure 1, $IW_{1,i} \in Q \times R$ are elements of the i th row of the weighting matrix in the first layer. “ $\cdot*$ ” stands for the dot product of input Vectors. $LW_{2,i} \in Q \times Q$ is the weight matrix. $a^1 (\in Q \times 1)$ is the output of first layer. R is the dimension of network input. Q is the neuron number of every layer and is also the number of training samples. Nprod presents the normal dot matrix for the calculating the output vectors $n^2 (\in Q \times 1)$. $a^2 (\in Q \times 1)$ is the output of GRNN.

GRNN adopts direct mapping from input cell to hidden layer. But the mapping between hidden layer and output layer adopts the linearly weighted sum of hidden layers as the mapping mode (shown as in Figure 1). This structure of NN can reduce the complexity of computational problems so as to speed up the learning process. It is suitable especially for accomplishing the function approximating and model identification quickly.

Usually, the Neural Network Predictive Control (NNPC) system consists of predictive model, looping optimizing and feedback adjusting [4]. Taking the errors between intending expectation and prediction as the input value, the predictive controller realizes the desired performance. The function of NN predictor model is to provide the output predictive information accurately and quickly. According to the prediction $\hat{y}(t)$ and the expectation $r(t)$, the designed target function J could be optimized by the Nonlinear Optimizer so as to reach the intending control goal. The general structure of NNPC system based on GRNN is shown as Figure 2, where $r(t)$ is the expectation value of output, $y(t)$ is actual value of output and $\hat{y}(t+k)$ is the prediction value of output [5].

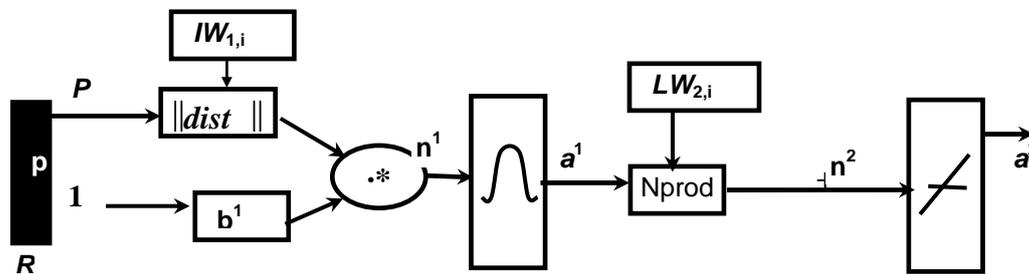


Figure 1. Structure of GRNN

2.2. GRNN-Based Predictive Model for Nonlinear System

In general, GRNN is made up of two layers, i.e. hidden layer and output layer. It is similar to the BP-NN in the model frame. The differences are the mapping function of GRNN adopts the Generalized Regression Algorithm for the training of Neural Network, which has the features of its output having the local linear relation to the parameters.

The network training could avoid the calculation of nonlinear optimizing and, therefore, there will not be the problem of local particle. So, the topological structure of neural network can be defined during the learning process, which is more quickly than BP-NN. To complete predictive control, the steps of NNPC are following:

- Obtain the expectation of output sequence $r(t+j)$ ($j=K_1, K_1+1, \dots, P$)
- Create the prediction by RBF-NN model $\hat{y}(t+j|t)$ ($j=K_1, K_1+1, \dots, P$)

And then, modify the errors:

$$y(t + j|t) = \hat{y}(t + j|t) + \alpha_j[y(t + j) - y(t + j|t - 1)] \quad (j= K_1, K_1+1, \dots, P)$$

c) Calculate prediction errors:

$$e(t + j) = r(t + j) - y(t + j|t) \quad (j= K_1, K_1+1, \dots, P)$$

d) Optimize the desired error goal function J so as to get the optimal control sequence:

$$u(t + j) \quad (j=0, 1, 2, 3, \dots, M)$$

e) Put the u(t) to the system, and back to step (1).

It is known the key function of GRNN predictor is to realize the prediction of system output quickly and accurately. In GRNN, the hidden layer adopts the nonlinear optimizing strategy. So, the learning speed of the function parameters will be slower accordingly.

Oppositely, the output layer adopts the linear optimizing strategy for the adjusting of weights. The learning speed is quicker than that of the hidden layer. The input vectors of lower dimensions are mapped to the hidden space of higher dimensions first. The hidden cells select the generalized regression basis function to realize the vector conversion, and then, sorted or identified by output layer. The form of base function F can be described as:

$$F(X) = \sum_{i=1}^N w_i \varphi(\| X - X_i \|) \quad (1)$$

Where:

$\varphi(\| X - X_i \|)$ is the generalized radial basis function (RBF), usually uses Gaussian function, $\| \bullet \|$ denotes the Euclid Norm.

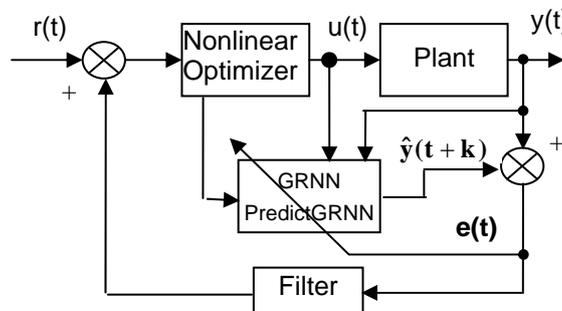


Figure 2. Structure of NNPC

To take known data $X_i \in R^{n_i}$ as the center of RBF and φ is radial-symmetrical to the center point. If Gauss function is adopted as RBF, it can be described as:

$$G(\| X - t_i \|^2) = \exp(-\frac{M}{d_m^2} \| X - t_i \|^2) \quad (2)$$

Where:

M is the number of center, t_i is the center of RBF and d_m is the maximal distance between selected centers. The mean-square-error of RBF takes $\sigma = d_m / \sqrt{2M}$.

According to the different ways for selection of RBF center, GRNN is usually applied the methods as random selection, self-organization learning (SOL) or supervisory learning, etc [6,

7]. This paper applies Orthogonal-least-square (OLS) learning algorithm to choose the center of RBF so as to complete the network training. In OLS algorithm, the central position of RBF is set automatically. The weights of output layer can be calculated by error-correction-learning algorithm. So, it is known the OLS algorithm is a mixed learning algorithm essentially. Its function is to adjust the center of RBF to the key area of input space.

Clustering least distance is the learning goal of OLS algorithm. The steps of training algorithm are:

- (1) Pre-elect randomly M as the cell number of hidden layer;
- (2) Pre-elect a set of center vector t_i ($1 < i < M$) for Radial Base Function. Distribute input samples X_n ($n=1, 2, \dots, N$) to every t_i and to get the regression operators matrix P :

$$P = [p_i(n)] = G(\|X_n - t_i\|) \quad (n=1,2,\dots,N, i=1,2,\dots,M) \quad (3)$$

(3) Orthogonalize the matrix P by the following equations:

$$\begin{cases} u_i = P_i \\ \alpha_{ik} = \frac{u_i^T \cdot P_k}{u_i^T \cdot u_i} \\ u_k = P_k - \sum_{i=1}^{k-1} \alpha_{ik} \cdot u_i \end{cases} \quad (4)$$

$(1 \leq i \leq k; \quad k = 2 \dots \dots M)$

Where: M_i is the number of input sample in θ_i

- (4) Calculate g_i and ε_i by using the following equation:

$$\begin{cases} g_i = \frac{u_i^T \cdot d}{u_i^T \cdot u_i} \\ \varepsilon_i = \frac{g_i^2 \cdot u_i^T \cdot u_i}{d^T \cdot d} \end{cases} \quad (1 \leq i \leq M) \quad (5)$$

Get the upper triangle matrix A , and then calculate the linking weights W by the equation: $A \cdot W = g$.

Where: $g = [g_1, g_2, \dots, g_M]^T$.

- (5) Check the whether the following equation is met. If it doesn't be met, turn back to step (2) for reselecting the center of RBF.

$$1 - \sum_{i=1}^M \varepsilon_i < \rho \quad (6)$$

Where: $0 < \rho < 1$ is the fixed tolerance.

The spread coefficient (Sc) is one of important parameters in the GRNN learning algorithm. The different decision of Sc will affect the learning performance directly. In fact, a suitable Sc can get smoother process for the signal approximating obviously. This paper presents an improved GRNN learning method based on the adaptive spread coefficient for the signal prediction of complex models. The adaptive spread coefficient is described as:

$$Sc = Sc + k_{\xi} \cdot e^{\xi} \quad (7)$$

Where:

k_ζ is the fitting coefficient ,

$$\zeta = |\min(err) - err^*|,$$

err^* is the desired learning error.

In order to get a suitable spread factor, the relative error of learning ζ is introduced into the training algorithm. The value of Sc can be modified adaptively with the training errors. Simulations show the amended algorithm can improve the training performance and simplify the structure of neural network satisfactorily.

3. Simulations

According to the sampling data of a nonlinear system, this paper applies a generalized regression neural network to structure a predictive model for a nonlinear system [8, 9, 11]. Suppose a nonlinear system can be described by following equations:

Input:

$$u(k) = 0.25 \cdot [\sin(\frac{2k\pi}{250}) + \sin(\frac{3k\pi}{100})] \quad (8)$$

Output:

$$y(k) = \frac{y(k-1) \cdot y(k-2) \cdot u(k-1)[y(k-3)-1] + u^2(k-2)}{1 + y^2(k-1) + y^2(k-2)} \quad (9)$$

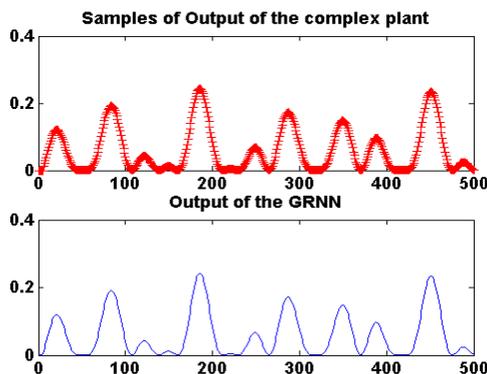


Figure 3. Learning Result of After 500 Pairs Samples

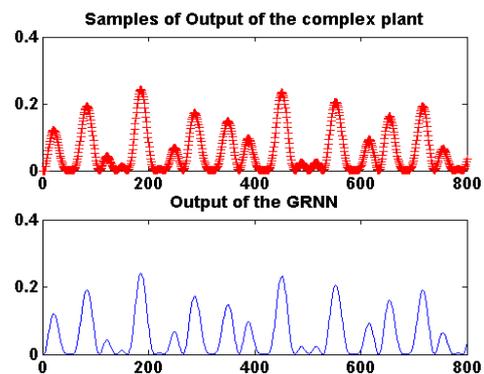


Figure 4. Learning Result of After 800 Pairs Samples

Suppose the system structure is unknown. GRNN needs to model the unknown system by learning the sampling data, to Sample respectively 500 pairs and 800 pairs of model I/O data, from a nonlinear system, as the learning samples for the GRNN. The training results and predictive output of GRNN are shown in Figure 3 and Figure 4. Where, '+' stands for sampling data, '—' shows the results after learning by the GRNN. The learning mean variance is $1.7576e-005$ and $1.3860e-005$ respectively. Meanwhile, after training finished, some pairs of new samples are sent to the trained GRNN so as to test its performance [10]. Simulations show the trained GRNN has the fine generalization ability obviously (shown in Figure 5).

It is shown the training algorithm based on GRNN is an effective way to realize the predictive control for the complex systems. The main strong feature of GRNN is its speediness and accuracy on function approximating of complex model, especially suitable for constructing the predictor model. It can provide the predict information for the predictive control system quickly and accurately.

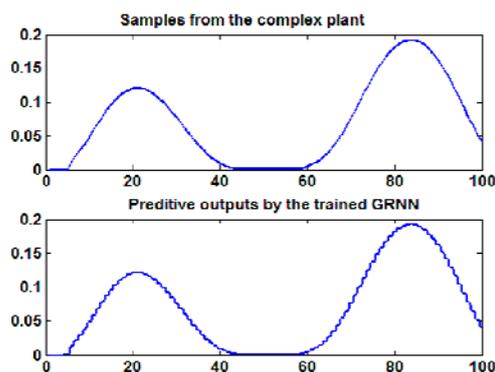


Figure 5. Generalization Ability of the Designed GRNN

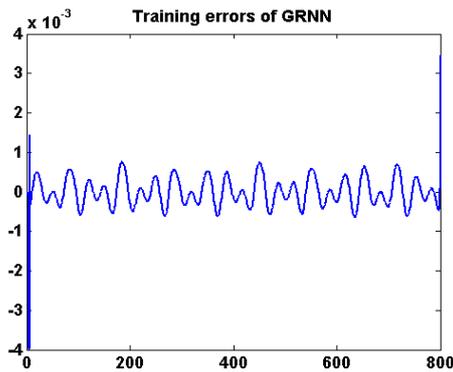


Figure 6. Learning Errors of the Trained GRNN

It can be found the learning algorithm based on GRNN shows a clear advantage on learning performances, especially on the model formation of predictor. It can improve performances obviously. It proved that the RBF-NN is more suitable for the modeling and identifying of complex systems. It is an effective method for the prediction and control of complex systems.

4. Conclusion

The characteristics of GRNN algorithm are the rate of convergence and the accuracy of learning results. Based on GRNN, this paper presents a training algorithm for the neural network predictor. After applying to the modeling of a typical nonlinear system and simulations, it is found that the GRNN-based predictor has a quicker learning process and an excellent training performance. In the aspect of function fitting and signal predicting for complex models, GRNN has more excellent performance than that of normal BP-NN. It is more suitable for getting the predictive model for complex systems. Therefore, GRNN is an effective way to realize the real-time predictive control for complex systems.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 61175053) and the Natural Science Foundation of Shandong Province, China (Grant No. Y2006G22).

References

- [1] Philippe De Wilde: Neural Network Models, An analysis. Springer-Verlag. 1996.
- [2] Biehl M, Ghosh A, Hammer B. Dynamics and Generalization Ability of LVQ Algorithms. *Journal of Machine Learning Research*. 2007; 8: 323-360
- [3] Takahiro Emoto, Masatake Akutagawa, Udantha R Abeyratne, Hirofumi Nagashino, Yohsuke Kinouchi. *Analysis of a Nonlinear System via Internal-State of a Neural Network*, Proceedings of EEE/EURASIP International Workshop on Nonlinear Signal and Image Processing, Sapporo, Japan. 2005; 931-935.
- [4] Schneider P, Biehl M, Hammer B. Distance Learning in Discriminative Vector Quantization. *Neural Computation*. 2009; 21(10): 2942-2969.
- [5] Younes C, Suranjan P, Leonard F: A generalized regression neural network and its application for leaf wetness prediction to forecast plant disease. *Chemometrics and Intelligent Laboratory System*. 1999; 48: 47-58.
- [6] Siefkes C, Siniakov P. An Overview and Classification of Adaptive Approaches to Information Extraction. *Journal of Data Semantics*. 2005; 4: 172-212.
- [7] M Bianchini, P Frasconi, M Gori. Learning without local minima in radial basis function networks. *IEEE Trans. on Neural Networks*. 1995; 6(3): 749-756.

-
- [8] Xue zhen-kuang, Li Shao-yuan. A Multi-Model modeling Approach to MIMO Nonlinear Systems. *Acta Electronica Sinica*. 2005; 33(1): 52-56.(in Chinese)
 - [9] Kosmatopoulos EB. High-Order Neural Network Structures for Identification of Dynamical Systems. *IEEE Trans. On Neural Networks*. 1995; 6(2): 422-431.
 - [10] Mithun M Bhaskar, Maheswarapu Sydulu. A Hybrid Genetic Algorithm Approach for Optimal Power Flow. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 9(2): 211-216.
 - [11] CHEN Zhong-xiao, CUI Ke, CHEN Xing-yu, LI Yan-fang. Distribution Network Fault Diagnosis Method Based on Granular Computing-BP. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(3): 1181-1188.