Identification of Dynamical Systems Using Radial Basis Function Neural Networks with Hybrid Learning Algorithm

Jun Li and Feng Zhao

***Abstract*—The paper demonstrates that radial basis function network (RBFN) with adaptive centers and width can be used effectively for identification of nonlinear dynamic system. The proposed RBFN is trained by hybrid learning algorithm, which uses conjugate gradient optimization algorithm to obtain the center and width of each radial basis function and the least squares method to obtain the weights. To avoid capturing a local optimum, regularization error energy function is used and the centers of basis functions are initialized using a fuzzy C-means clustering method. Simulation results reveal that the identification schemes based on RBFN gives considerably better performance and show faster learning in comparison to previous methods.**

I. Introduction

Multilayer feedforward neural networks (MFNN) have been shown to obtain successful results in system identification and control because they provide good approximations when applied to highly nonlinear and complex systems without any prior knowledge [1], [2]. However, MFNN has poor process interpretability and are hindered by problems associated with weight optimization such as slow learning and local minimization. At present, radial basis function networks (RBFN) have been applied as an alternative to conventional artificial neural networks [3], [4]. As is widely known, this is becoming an increasingly popular neural network with diverse applications and is probably the main rival to the multi-layered perceptron.

RBFN consist of two layers with different functional roles: hidden and output layers, Inputs are transformed into a high-dimensional feature space using radial basis functions. Then, the output layer performs a simple weighted sum with a linear output in order to approximate the desired outputs.

In general, the number of radial basis functions should be determined taking into consideration both the generalization performance and the problem of overfitting. Three types of parameters are determined to construct the RBFN: the centers and widths (or spreads) of the radial basis functions in the hidden layer, and the weights connecting the hidden layer, and the weights connecting the hidden layer with the output layer. A number of training algorithms for RBFN have been proposed in [5], [6]. Usually, the two parameters of the radial basis functions and the connecting weights are trained separately in a sequential manner.

The functional behaviour of RBFN is actually equivalent to that of a fuzzy inference system (FIS) under some limitations. Hence, it is possible to apply various structure and parameter identification methods of FIS to RBFN and vice versa [7], [8]. In addition, using support vector machine (SVM) learning to identify the parameters of RBFN is also effective method. This method automatically obtains the location of the centers and weights from a set of training data, More importantly, SVM approximately minimizes structural risk considering generalization performance and SVM learning may have a regularization [9], [10].

As an alternative learning strategy, we consider an extended approach of RBFN that utilizes hybrid learning strategy combined with an unsupervised parameter initialization method. The RBFN used in the experiments are an extension of the method of Moody and Darken [3]. After initialization of the centers and widths of the basis functions using modified K-means clustering method, which also automatically determines the number of basis functions, and these parameters and the connecting weights are tuned by the iterative conjugate-gradient and least squares algorithm respectively. The regularization error energy function is also used to overcome ill-posed problem.

The remainder of this paper is organized as follows. In Section II, The hybrid learning algorithm of adaptive RBFN is given. Several examples for identification of nonlinear systems are presented to verify the performance of the RBFN in Section III. Finally, in Section IV, some concluding remarks are made.

II. Radial Basis Function Neural Networks

RBFN can be trained using several hybrid learning algorithms. Such algorithms are based on the idea of sequentially using nonlinear and linear optimization techniques to train the nonlinear and linear parameters. Hence, we chose to apply the least squares method for the linear weights parameters and conjugate-gradient iterative optimization algorithm for nonlinear ones i.e. centers and widths of radial basis functions.

A. Determinate of Initial Centers and Widths

As we known, both approximation capability and generalization performance should be taken into

This work was supported in part by the Natural Science Foundation of Gansu Province under Grant No. 3ZS042-B25-026 and supported by "Qinglan" Talent Engineering Funds by Lanzhou Jiaotong University.

Jun Li is with the School of Information & Electrical Engineering, Lanzhou Jiaotong University Lanzhou, P. R. China (e-mail: lijun691201@yahoo.com.cn).

Feng Zhao is with the School of Information & Electrical Engineering, Lanzhou Jiaotong University Lanzhou, P. R. China (e-mail: zhao_wind818@263.net).

consideration when we determine the proper choice of the number of the radial basis functions. Hence, in the paper, the Fuzzy C-means cluster algorithm is applied to determine the initial parameters. The available data samples are collected in matrix Z formed by concatenating the regression data matrix X and output vector *y*:

$$X=\begin{bmatrix}\boldsymbol{x}_1^{\mathrm{T}}\\ \vdots\\ \boldsymbol{x}_N^{\mathrm{T}}\end{bmatrix}, y=\begin{bmatrix}y_1\\ \vdots\\ y_N\end{bmatrix}, Z^{\mathrm{T}}=\left[\boldsymbol{X}y\right].$$

Each observation thus is an n+1 dimensional column vector

$$\boldsymbol{z}_k=\left[x_{1,k},\cdots,x_{n,k},y_k\right]^{\mathrm{T}}=\left[\boldsymbol{x}_k{}^{\mathrm{T}}y_k\right].$$

The clustering algorithm is based on the minimization of an objective function call C-mean functional. It is defined as follows:

$$J\left(\boldsymbol{X},\boldsymbol{U},\boldsymbol{V}\right)=\sum_{i=1}^{c}\sum_{j=1}^{N}\mu_{i,k}{}^{m}D_{i,k}{}^{2} \tag{1}$$

where contains the clusters and $m\in[1,\infty]$ is a weighting exponent that determines the fuzziness of the resulting clusters and it is often chosen as m=2. The fuzzy partition matrix has to satisfy the following conditions:

$$\begin{aligned}&\boldsymbol{U}\in R^{c\times N}\ with\quad \mu_{i,k}\in\left[0,1\right],\forall i,k;\\ &\sum_{i=1}^{c}\mu_{i,k}=1,\forall k;0<\sum_{k=1}^{N}\mu_{i,k}<N,\forall i\end{aligned} \tag{2}$$

A fuzzy C-means cluster scheme as follows [11]:

1) Initialization: Given the data set X, choose the number of clusters 1<c <N, initialization with random cluster centers chosen from data set X.

2) Repeat for l=1, 2,…,

Step 1: Compute the cluster prototypes (means)

$$\boldsymbol{v}_i^{(l)}=\frac{\sum_{k=1}^{N}\left(\mu_{i,k}{}^{(l-1)}\right)^{m}\boldsymbol{z}_k}{\sum_{i=1}^{N}\mu_{i,k}{}^{(l-1)}},1\le i\le c. \tag{3}$$

Step 2: Compute the distance measure $D_{i,k}$ .The distance to prototype is calculated based the covariance matrices of the cluster

$$\boldsymbol{F}_i^{(l)}=\frac{1}{N}\sum_{k=1}^{N}\left(\boldsymbol{z}_k-\boldsymbol{v}_i^{(l)}\right)\left(\boldsymbol{z}_k-\boldsymbol{v}_i^{(l)}\right)^{\mathrm{T}}$$

The distance function is chosen as

$$D_{i,k}{}^{2}\left(\boldsymbol{z}_k,\boldsymbol{v}_i\right)=\left(\boldsymbol{z}_k-\boldsymbol{v}_i^{(l)}\right)^{\mathrm{T}}\boldsymbol{F}_i^{-1}\left(\boldsymbol{z}_k-\boldsymbol{v}_i^{(l)}\right) \tag{4}$$

Step 3: Update the partition matrix

$$\mu_{i,k}{}^{(l)}=\frac{1}{\sum_{j=1}^{c}\left(D_{i,k}\left(\boldsymbol{z}_k,\boldsymbol{v}_i\right)/D_{j,k}\left(\boldsymbol{z}_k,\boldsymbol{v}_j\right)\right)^{2/(m-1)}} \tag{5}$$

Until $\left\|\boldsymbol{U}^{(l)}-\boldsymbol{U}^{(l-1)}\right\|<\varepsilon$ are obtained.

B. Hybrid Learning Algorithms of RBFN

The RBFN used in here is an extension of the method of Moody and Darken [3], since centers and widths are also adapted. That is, the output weights of RBFN and the RBF centers and widths can be adjusted simultaneously. In the way, the network fine-tunes itself to the data after the above clustering steps [6].

The output of the network is computed as a linear superposition of C basis function

$$f\left(\boldsymbol{x}\right)=\sum_{k=1}^{c}w_k h_k\left(\boldsymbol{x}\right) \tag{6}$$

where w_k, k=1,2,…,c, denotes the weights of the weights of the output layer.The Gaussian radial basis functions h_k are defined as

$$h_k\left(\boldsymbol{x}\right)=\exp\left(-\frac{\left\|\boldsymbol{x}-\boldsymbol{v}_k\right\|^2}{2\sigma_k{}^{2}}\right) \tag{7}$$

where $\boldsymbol{v}_k$ and σ_k denotes centers and widths,respectively. In a first step,the centers $\boldsymbol{v}_k$ are initialized with fuzzy C-means clustering and the widths σ_k is determined as the distance between $\boldsymbol{v}_k$ and the closest $\boldsymbol{v}_i$, $i\neq k$, $i\in(1,\ldots,c)$.

The regularized error energy function i.e. weight decay is defined as follows:

$$E=\frac{1}{2}\sum_{n=1}^{N}\left(y_n-f\left(\boldsymbol{x}_n\right)\right)^2+\frac{\lambda}{2N}\sum_{k=1}^{c}w_k{}^{2} \tag{8}$$

Taking the derivate of (8) with respect to $\boldsymbol{v}_k$ and σ_k, we obtain

$$\frac{\partial E}{\partial\boldsymbol{v}_k}=\sum_{n=1}^{N}\left(f\left(\boldsymbol{x}_n-y_n\right)\right)w_k\frac{\boldsymbol{x}_n-\boldsymbol{v}_k}{\sigma_k{}^{2}}h_k\left(\boldsymbol{x}_n\right) \tag{9}$$

$$\frac{\partial E}{\partial\sigma_k}=\sum_{n=1}^{N}\left(f\left(\boldsymbol{x}_n-y_n\right)\right)w_k\frac{\left\|\boldsymbol{x}_n-\boldsymbol{v}_k\right\|^2}{\sigma_k{}^{3}}h_k\left(\boldsymbol{x}_n\right) \tag{10}$$

The two derivatives are employed in the minimization of (8) by a conjugate gradient decent with line search, where we always compute the optimal output weights in every evaluation of the error energy function during the line search.The optimal output weights $\boldsymbol{w}$=[w_1,w_2, …,w_k]$^{\mathrm{T}}$ in matrix notation can be computed in closed form by

$$\boldsymbol{w}=\left(H^{\mathrm{T}}H+2\frac{\lambda}{N}\boldsymbol{I}\right)^{-1}H^{\mathrm{T}}\boldsymbol{y} \tag{11}$$

where $H_{i,k}=h_k\left(\boldsymbol{x}_i\right)$ and $\boldsymbol{y}$=[y_1,y_2, …,y_N]$^{\mathrm{T}}$ denotes the output vector, and $\boldsymbol{I}$ an identity matrix.

Though careful tuning of the regularized parameter, the number of centers C and the numbers of iterations, overfitting

phenomenon may be avoided.

A hybrid learning algorithms of the RBFN is as follows:

1) Initialization: Set the number of RBFN centers c and then run fuzzy C-means clustering to find initial value for $\boldsymbol{v}_k$ and determine σ_k , k=1,2, …,c, as the distance between $\boldsymbol{v}_k$ and the closest $\boldsymbol{v}_i$ ($i \neq k$). Set the regularization constant and the maximum number of iterations l.

2) Repeat for l=1, 2,…,

Step 1: Using (11) Compute optimal output weights.

Step 2: Compute (9) and (10) with optimal $\boldsymbol{w}$ and form a gradient vector $\boldsymbol{v}$ and estimate the conjugate direct $\overline{\boldsymbol{v}}$ with Fletcher-Reeves Method [12].

Step 3: Perform a line search to find the minimizing step δ in direction $\overline{\boldsymbol{v}}$ and in each evaluation of E recomputed the optimal output weights as in Step 1.

Step 4: Update $\boldsymbol{v}_k$ and σ_k with $\overline{\boldsymbol{v}}$ and δ .Until the maximum number of iteration is obtained.

III. Application to Identification of Nonlinear Dynamic Systems

To illustrate the performance of the presented RBFN approach, three numerical instances are provided and all these examples can find their origin in the existing literature, which is convenient for the purpose of comparison.

Example 1: The first example is taken from [13] in which the plant to be identified is given by the second-order nonlinear difference equation

$$y_{k+1} = \frac{y_k y_{k-1}\left(y_k + 2.5\right)}{1 + y_k^{\,2} + y_{k-1}^{\,2}} + u_k \tag{12}$$

700 simulated data points are generated from the plant model (12).The first 500 data points are obtained by assuming a random input signal uniformly distributed in the interval [-2, 2], and the last 200 data points are obtained by using a sinusoid input signal, i.e. $u_k = \sin\left(2\pi k/25\right)$. We use the first 500 data points to build a identifier model based on RBFN. The performance of the RBFN model is verified using the remaining 200 data points.

We select y_k, y_{k-1} and u_k as the input variables and the number of clusters are set to ten. Therefore, the RBFN model consists of ten radial basis functions with three-dimensional premises. The regularized parameter $\lambda = 10^{-6}$ and up to 50 CG iterations.

The mean squared error (MSE) is 0.0086 in training region and the MSE is 0.0096 in test region, respectively.

Fig. 1 shows the outputs of the identified model and the actual model. It can be seen from Fig.1 that the outputs from identified model attains a rather good match with the actual model, not only in the modeling region bur also in the testing region. Only 500 data points have been used to identify the model here; while in [13], 100,000 data points have been used to identify a multilayer feedforward neural network model. If the number of data points used to identify the model is increased, the performance of identified model based on RBFN can be further improved.

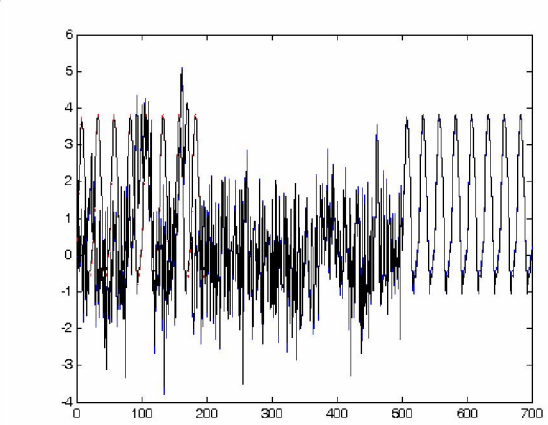

Fig. 1. Outputs of the identified model (solid line) and the actual model (dashed line) in example 1.

Example 2: We consider the gas furnace data of Box-Jenkins [14]. The data set is well known and frequently used as a benchmark instance to illustrate identification algorithms. It consists of 296 pairs of input-output measurements taken from a laboratory furnace with a sampling time of 9 seconds. The process input is the methane flow rate and the output is the percentage of CO_2 in the off gas. Many researchers drew a conclusion that a structure of the dynamic model for this system is

$$y_{k+1} = f\left(y_k, u_{k-3}\right) \tag{13}$$

We apply RBFN to the above dynamic model, and also set the numbers of clusters to ten. The regularized parameter $\lambda = 10^{-6}$ and up to 50 CG iterations. The mean squared error (MSE) of the training data is 0.1203. Fig.2 shows the identified model behavior in comparison with the actual process. It can be seen from Fig.2 that the performance is very good. The approximation power of the model can be appreciated if we compare the achieved modeling performance (MSE) with other results [15], [16].

Example3: In the example, the input is seen to occur nonlinearly in the difference equation describing the plant. The plant has the form [13]:

$$y_{k+1} = \frac{y_k}{1 + y_k^{\,2}} + u_k^{\,3} \tag{14}$$

Similarly, a series-parallel identification model based on RBFN can be applied, which is described as follows:

$$\widehat{y}_{k+1} = N_f\left[y_k, y_{k-1}, y_{k-2}, u_k, u_{k-1}\right] \tag{15}$$

3100 simulated data points are generated from the plant model (12).The first 3000 data points are obtained by assuming a random input signal uniformly distributed in the interval [-2, 2], and the last 100 data points are obtained by using a sinusoid input signal, i.e.

$u_k = \sin\left(2\pi k/25\right) + \sin\left(2\pi k/10\right)$.

We use the first 3000 data points to build an identifier model based on RBFN. The performance of the RBFN model is verified using the remaining 100 data points. The numbers of clusters are set to ten. Hence, the RBFN model consists of ten radial basis functions with five-dimensional premises. Fig. 3 shows the outputs of the identified model and the actual model. It can be seen from Fig.3 that the outputs from identified model attains a rather good match with the actual model.

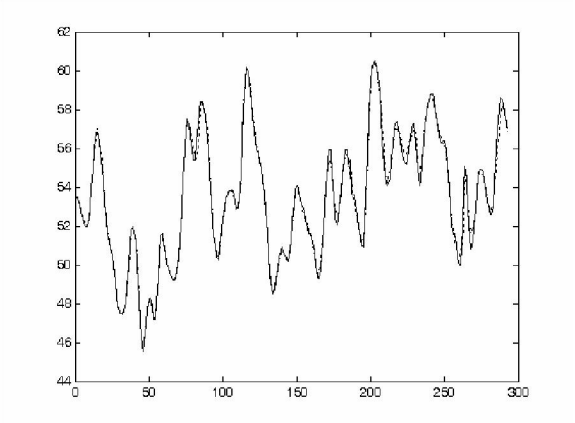

Fig. 2. Outputs of the identified model (solid line) and the actual gas furnace model (dashed line) in example 2.

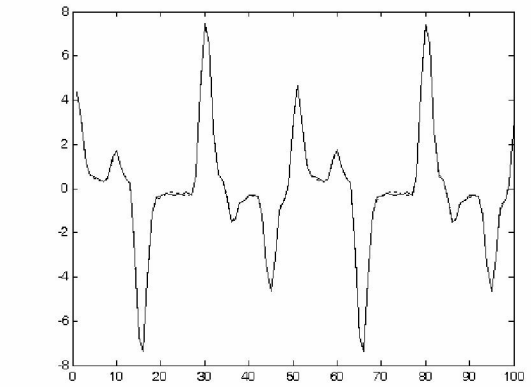

Fig. 3. Outputs of the identified model (solid line) and the actual model (dashed line) in example 3.

IV. Conclusion

An extended version of RBFN with an efficient hybrid learning algorithm has been given in this paper. The proposed RBFN is trained by a hybrid learning algorithm, which uses conjugate gradient optimization algorithm to obtain the center and width of each radial basis function and the least squares method to obtain the weights. To avoid capturing a local optimum, the centers of basis functions are initialized using a fuzzy C-means clustering method and regularization error energy function is also used. In application, the RBFN has smaller network structure and a small number of turning parameters than the multiply feedforward neural network. We apply the RBFN model to identify the nonlinear system. Experiment results show that the identification schemes based on RBFN gives considerably better performance and show faster learning in comparison to previous methods. Furthermore, the RBFN can also uses as adaptive controllers for nonlinear systems.

In conclusion, RBFN exhibits powerful function approximation capabilities, reasonable identification and prediction power for modeling real systems. RBFN with adaptive centers and width can be used effectively for identification and control of nonlinear dynamic system.

Acknowledgment

The authors would like to thank Dr. Gunnar Raetsch for his help.

References

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.1995, pp. 164–191.

[2] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., New Jersey: Prentice-Hall, 1999, pp. 156–392.

[3] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, no.2, pp. 281-294, 1989.

[4] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol.78, pp1481-1497, 1990.

[5] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.

[6] K. –R. Muller, A .Smola, and G. Ratsch etc, "Using support vector machines for time series prediction," In B. Scholkopf, C. Burges, and A. Smola, ed., *Advances in Kernel Methods-Support Vector Learning*, pp. 243-254, Cambridge, MA, MIT Press, 1999.

[7] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stable Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1994, ch. 7.

[8] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. New Jersey: Prentice-Hall, 1997, ch. 9.

[9] W. Kaminski, P. Strumillo, "Kernel orthonomalization in radial basis function neural networks," *IEEE Trans. on Neural Networks*, vol.8, no. 5, pp. 1177-1183,1997

[10] B. Schölkopf, K. -K Sung and C J. C.Burges, "Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers," *IEEE Trans on Signal Processing*, vol.45, no.11, pp. 2758-2765, 1997.

[11] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, 1981.

[12] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes in C*, 2nd ed., Cambridge: Cambridge University Press, 1992.

[13] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamic systems using neural networks," *IEEE Trans. on Neural Networks*, vol.1, no.1, pp.4-27, 1990.

[14] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forcasting and Control*, 2nd ed., San Francisco: Holden-Day, 1976.

[15] M. Sugeno and K. Tanaka, "Successive identification of a fuzzy model and its application to prediction of a complex system," *Fuzzy Sets and Systems*, vol 42, pp. 315-334, 1991.

[16] L. Wang and R. Langari, " Complex systems modeling via fuzzy logic," *IEEE Trans. Syst., Man, and Cyb.-Part B*, vol. 26, no. 1, pp. 100-106, 1996