Time-Varying Neural Networks for Dynamical Systems Modeling with Application to Image Compression

Ying Kong¹ and Hui-juan Lu²

¹*School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou, China kongying-888@163.com*

²*College of Information Engineering, China Jiliang University, Hangzhou, China hju@cjljlu.edu.cn*

Abstract

The architecture of a class of time-varying neural networks can be determined by simply adopting that of the conventional neural networks, while the weights are allowed to vary with time. The challenge lies how to select the weights, when applying a time-varying neural network. In this paper, we use the iterative learning methodology for training time-varying neural networks, and the neural networks are proposed for modeling and identification of discrete-time time-varying nonlinear systems. Time-varying dynamical neural networks (DNNs) are presented by the architecture of conventional high-order DNNs with connection weights varying with time. Both conventional DNNs and time-varying DNNs are used to identify time-varying systems. The weights are updated by least squares integral learning algorithm with dead-zones. For time-varying case, iterative learning and its improved algorithms are used to update connection weights. The identification error is ensured to converge to the bound, which is proportional to the approximation error.

Keywords: *Identification, Iterative learning algorithms, Time- varying neural networks, Image compression*

1. Introduction

Neural network has powerful approximation, which has been applied for identification and control of nonlinear systems. The weight of conventional neural network is always constant that updated by integral learning law. However, time-varying neural networks have variant weights. After the structure of networks has been known, how to train the connection weights becomes the most important key that directly decides the practical application of time-varying neural network. Both iterative learning control and repetitive control achieve perfect tracking task under some assumptions. Time-varying weights updated by iterative learning algorithm become a feasible alternative. Time-varying nonlinear systems have sophisticated dynamical time-varying properties. Until now, research results in literatures for identification and control of nonlinear time-varying is comparatively few [1,2,3].

Neural network identification is a primary method of nonlinear system identification. For a given excitation function with a sufficient number of neurons, the neural network can approximate arbitrary precise linear continuous function in a very precise way [4,5]. Because of the powerful approximation based on neural network, modeling and identification of nonlinear systems could be realized by neural training [6,7].

The dynamic neural networks have a feedback mechanism itself. System input and output data to on-line measurement obtained by dynamic neural network can be used for

¹ Ying Kong, Liuhe Road No. 318, Hangzhou, Zhejiang, 310023, kongying-888@163.com

nonlinear dynamic system identification. The literature [8] proposed a higher-order recurrent neural network(HONN) model using gradient algorithm and least squares algorithm which is based on the convergence of the identification error. Literature [9] adopts gradient descent algorithm to adjust the network weights and applies passivity analysis training algorithm to prove the stability and robustness for high-end multi-input and multi-output nonlinear systems. Literature [10] uses output system itself, rather than the derivative signal output dynamic neural network structure, to achieve system identification by using projection algorithm for network weights training and design controller. The discussion on dynamic neural network approximation capabilities, and applications seen in terms of feedback linearity was addressed in literatures [11,12].

In practical applications, we often encounter non-stationary process. Iterative learning identification with dynamical time-varying system used for the extraction of hidden digital image recovery and digital image watermarking extraction. For instance, we can make an image hide in another image by using multiple hybrid method which will be embedded into the carrier image at the same time. And we also can compress an image to time-varying weights which store the information of the image with little storage spaces, then use iterative learning identification method to restore the hidden image. At the same time, a digital image watermarking scheme based on chaotic mapping and multi-blending, and the decryption and extraction of the digital watermark can also be carried out from iterative time-varying dynamical system. Such as the watermark image can be added to a parameter of the chaotic system. An iterative time-varying learning neural network would recover the embedded image by training the weights.

Many studies use fixed weights neural networks for system modeling and identification. When taking the non-stationary process variations tracking features into account, variable neural network to solve related problems are welcomed by researchers in recent years. Invariant neural network has fixed neural weights, and time-varying neural network has time-varying weights on the contrary. The key way for time-varying neural networks is training, after the network structure is determined. One approach is to change the weights to a linear combination of a given orthogonal basis functions when the network is represented [13,14]. Training the network is converted to a combination of factors to learn, and can be applied to invariant neural network.

In this paper, we model the time-varying network to consider the identification problem of nonlinear discrete system over finite interval. The neural network which has the ability to approximate nonlinear dynamic systems when given the structure of the time-varying neural networks to achieve recognition of the nonlinear time-varying systems by finding the optimal weights. These weights change over time while the system is running, but for a fixed time, they do not change with running times. As a result, it can apply iterative learning methodology to get time-varying weights and finish neural network training. The paper first proves the ability of dynamical neural network to fully identify over the finite interval. Then, we use iterative learning least squares with dead-zones algorithm for error approximation to ensure the robustness.

The remainder of this paper is organized as follows. Section 2 presents a special kind of system equation modeled by the time-varying neural network including the structure of the time-varying network. Section 3 presents a learning algorithm which could ensure the effectiveness and stability. Section 4 presents the computer simulation results for time-varying modeling. Finally, Section 5 concludes this paper with final remarks.

2. Time-Varying Neural Networks

A lot of invariant-time networks for nonlinear time-varying system modeling have been proposed in the published literatures [15][16]. Here, we solve the same problem using dynamic characteristics of the network. Differentiating from the invariant-time networks, the weights of the input and output of such a network can change over time. A

class of multi-input and single-output time-varying neural network can be expressed as below

$$y(t) = \varphi^T(x(t))W(t) \quad (1)$$

where $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T \in R^n$ are the input vector, $y(t)$ is the Output Scalar of neural network, $W(t) = [w_1(t), w_2(t), \dots, w_l(t)]^T$ are the adjustable variable weight vector. l represents Neuron nodes. $\varphi(x(t)) = [\varphi_1(x(t)), \varphi_2(x(t)), \dots, \varphi_l(x(t))]^T$ are excitation function vectors.

Let us take the high-order neural network for example, the discrete high-order neural networks can be described as:

$$y_{NN}(t+1) = \sum_{i=1}^N \left(w_i(t) \prod_{j \in I_i} s(z_j(t))^{d_j(i)} \right) \quad (2)$$

where $z(t) = [z_1(t), \dots, z_n(t)]^T$ is the input of HONNs. n is the number of input nodes, and N is the number of NN nodes in the hidden layer. $w(t) = [w_1(t), \dots, w_N(t)]^T$ are time-varying weights connecting the hidden layer to the output layer. $s(\cdot)$ is the active function, $I_i \in \{1, 2, \dots, N\}$ is a collection of not-ordered subsets of $\{1, 2, \dots, n\}$, and $d_j(i)$ are nonnegative integers.

Let us define

$$\begin{aligned} w(t) &= [w_1(t), \dots, w_N(t)]^T \\ v_j(t) &= [v_{j1}(t), \dots, v_{jn}(t)]^T \\ v(t) &= [v_1(t), \dots, v_N(t)]^T \\ s(v^T(t)z(t)) &= [s_1(v_1^T(t)z(t)), \dots, s_N(v_N^T(t)z(t))]^T \end{aligned}$$

then we have

$$y_{NN}(t+1) = w^T(t)s(v^T(t)z(t))$$

As the weights become to be time-invariant, we have the conventional networks. One can give more TVNNs through using time-varying weights in the conventional neural networks, such as RBF, B-spline neural networks, wavelets neural networks, etc. The most common choices for activation function $s(\cdot)$ are

1) Heaviside step function

$$s(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

2) Logistic function $s(z) = 1/(1+e^{-z})$;

3) Hyperbolic tangent function $s(z) = (e^z - e^{-z})/(e^z + e^{-z})$.

There are other alternatives for choosing activation functions. One important feature of the mentioned activation functions is that $s(\cdot)$ presented above is globally bounded with respect to z .

Nonlinear time-varying systems are widely existed in the actual system, We can

approximate the nonlinear mapping using the above time-varying neural network. For each particular moment t , $f(x(t), t)$ is a nonlinear function with respect to x . The time-varying neural network can degenerate into time-invariant one at each t . where $t=0,1,2,\dots,N$ is the time, $u(t), y(t)$ are the measurable scalar input and output of the system, respectively, and f is an unknown smooth function of its arguments. Note that as f is not t -dependent explicitly, (1) becomes the conventional time-invariant nonlinear system, and many results can be applied for its identification.

3. Nonlinear Time-varying System Modeling

Consider the following nonlinear discrete repeatable time-varying system

$$y_k(t+1) = f(y_k(t), \dots, y_k(t-n+1), u_k(t), \dots, u_k(t-m+1), t) \quad (3)$$

where $m > 0, n > 0$; $f(\cdot)$ is the nonlinear time-varying function;
 $t \in \{0, 1, \dots, T\}$ represents the time interval; k denotes the iteration number.

$x_k(t) = [y_k(t), \dots, y_k(t-n+1), u_k(t), \dots, u_k(t-m+1)]^T$, so the dynamical system (2) can be rewritten as:

$$y_k(t+1) = f(x_k(t), t) \quad (4)$$

Let us consider the time-varying network model below:

$$\hat{y}_k(t+1) = \hat{f}(x_k(t), W(t)) \quad (5)$$

where $\hat{y}_k(t)$ represents the output of the neural network; $\hat{f}(\cdot)$ is determined by the structure of the time-varying neural network and the number of the neurons;
 $W(t)$ denotes the time-varying weights of the neural network. Our purpose of this manuscript is to construct an iterative learning algorithm to estimate the value of the variable weights by repeating the varying system.

For a given set of $S \subset R^{n+m}$, function $\hat{f}(\cdot)$ can approximate the the objective function $f(\cdot)$, the identification error $\bar{\varepsilon}$ is ensured to converge to the bound. Note that $\forall x_k(t) \in S, \exists W(t)$, the solution of (5) satisfies:

$$y_k(t+1) = \hat{f}(x_k(t), W(t)) + \varepsilon(x_k(t), t) \quad (6)$$

where $\varepsilon(x_k(t), t) \leq \bar{\varepsilon}$, $\bar{\varepsilon} > 0$.

if $W_k(t)$ is the value of the k th iteration, the output of the k th iterative neural modeling can be expressed:

$$y_k^*(t+1) = \hat{f}(x_k(t), W_k(t)) = \varphi^T(x_k(t))W_k(t) \quad (7)$$

the estimating error is: $e_k(t+1) = y_k(t+1) - y_k^*(t+1)$. Iterative learning algorithm with dead zone modification are proposed for updating the time-varying weights, and the identification error is ensured to converge to the bound, which is proportional to the approximation error.

3.1. Iterative Learning Algorithms with Dead-zones Modification

To overcome the impact of non-zero identification error, iterative learning algorithm with dead zone modification are proposed for updating the time-varying weights along iterative axis. We adopt the algorithm as follows:

$$\hat{w}_{k+1}(t) = \hat{w}_k(t) + \frac{\alpha_k(t)\beta_k(t)e_k(t)P_k(t)\varphi_k(t)}{1 + \alpha_k(t)\beta_k(t)\varphi_k^T(t)P_k(t)\varphi_k(t)} \quad (8)$$

$$P_{k+1}(t) = P_k(t) - \frac{\alpha_k(t)\beta_k(t)P_k(t)\varphi_k(t)\varphi_k^T(t)P_k(t)}{1 + \alpha_k(t)\beta_k(t)\varphi_k^T(t)P_k(t)\varphi_k(t)} \quad (9)$$

where $\alpha_k(t)$ and $\beta_k(t)$ are given as follows:

$$\alpha_k(t) = \frac{|e_k(t)|\bar{\varepsilon}^{-1} - 1}{1 + \varphi_k^T(t)P_k(t)\varphi_k(t)} \quad (10)$$

$$\beta_k(t) = \begin{cases} 0 & , |e_k(t)| < \bar{\varepsilon} \\ 1 & , |e_k(t)| \geq \bar{\varepsilon} \end{cases} \quad (11)$$

The update laws for $P_k(t)$ and $w_k(t)$, given by (8) and (9), indicate the iterative learning least squares algorithm for network training, where $t=0,1,\dots,N$ and $k=0,1,\dots$, $w_0(t)$ and $P_0(t)>0$ are given a priori for $t=0,1,\dots,N$. Since $\varphi_k^T(t)P_k(t)\varphi_k(t)$ in (8) is a scalar, no matrix inversion is necessary in the computation.

3.2 Convergence Analysis

In this section, an important lemma about the convergence properties of least squares algorithm is established for identification problems. It presents properties of the proposed time-varying neural networks.

Lemma 1: The time-varying neural network and the learning algorithm, including (7), (8) and (9), ensure the following properties:

(i) For $t=0,1,\dots,N$ and $k=0,1,\dots$, $\|\tilde{w}_k(t)\|^2 \leq \frac{\lambda_{\max}(P_0^{-1}(t))}{\lambda_{\min}(P_0^{-1}(t))} \|\tilde{w}_0(t)\|^2$, where $\lambda_{\max}(P_0^{-1}(t))$ is the maximum eigenvalue of $P_0^{-1}(t)$ and $\lambda_{\min}(P_0^{-1}(t))$ is the minimum eigenvalue of $P_0^{-1}(t)$, respectively.

(ii) For $t=0,1,L,N$,

$$\lim_{k \rightarrow \infty} \beta_k(t) \frac{(|e_k(t)| - \bar{\varepsilon})^2}{1 + \varphi_k^T(t)P_k(t)\varphi_k(t)} = 0 \quad (12)$$

Proof. (i) subtracting $w^*(t)$ from both sides of (8)

$$\tilde{w}_{k+1}(t) = \tilde{w}_k(t) + \frac{\alpha_k(t)\beta_k(t)e_k(t)P_k(t)\varphi_k(t)}{1 + \alpha_k(t)\beta_k(t)\varphi_k^T(t)P_k(t)\varphi_k(t)} \quad (13)$$

From the matrix inversion lemma in [15], (9) implies

$$P_{k+1}^{-1}(t) = P_k^{-1}(t) + \alpha_k(t)\beta_k(t)\varphi_k(t)\varphi_k^T(t) \quad (14)$$

Define a Lyapunov candidate function

$$V_k(t) = \tilde{\Theta}_k^T(t)P_k^{-1}(t)\tilde{\Theta}_k(t) \quad (15)$$

Using equations (13), (14), and (15), yields

$$V_{k+1}(t) - V_k(t) = \alpha_k(t)\beta_k(t)[\varepsilon_k^2(t) - \frac{e_k^2(t)}{1 + \alpha_k(t)\beta_k(t)\varphi_k^T(t)P_k(t)\varphi_k(t)}] \quad (16)$$

Now we will discuss two cases according to (11)

Case 1: if $|e_k(t)| < \bar{\varepsilon}$, $\beta_k(t) = 0$, hence

$$V_{k+1}(t) - V_k(t) = 0 \quad (17)$$

Case 2: if $|e_k(t)| \geq \bar{\varepsilon}$, $\beta_k(t) = 1$, $\alpha_k(t) \geq 0$, hence

$$V_{k+1}(t) - V_k(t) = \alpha_k(t) \left[\varepsilon_k^2(t) - \frac{e_k^2(t)}{1 + \alpha_k(t) \varphi_k^T(t) P_k(t) \varphi_k(t)} \right] \quad (18)$$

Since $|\varepsilon_k(t)| \leq \bar{\varepsilon}$, $|e_k(t)| \geq \bar{\varepsilon}$

$$\varepsilon_k^2(t) (1 + \alpha_k(t) \varphi_k^T(t) P_k(t) \varphi_k(t)) \leq \frac{\varepsilon_k^2(t)}{\bar{\varepsilon}} |e_k(t)| \leq |e_k(t)| \bar{\varepsilon} \quad (19)$$

Using (18), equation (19) leads to

$$\begin{aligned} V_{k+1}(t) - V_k(t) &\leq \frac{\alpha_k(t)}{1 + \alpha_k(t) \varphi_k^T(t) P_k(t) \varphi_k(t)} |e_k(t)| (\bar{\varepsilon} - |e_k(t)|) \\ &= - \frac{(|e_k(t)| - \bar{\varepsilon})^2}{\frac{\bar{\varepsilon}}{|e_k(t)|} + \varphi_k^T(t) P_k(t) \varphi_k(t)} \\ &\leq - \frac{(|e_k(t)| - \bar{\varepsilon})^2}{1 + \varphi_k^T(t) P_k(t) \varphi_k(t)} \end{aligned} \quad (20)$$

Summarize these two cases, we can conclude that

$$V_{k+1}(t) - V_k(t) \leq -\beta_k(t) \frac{(|e_k(t)| - \bar{\varepsilon})^2}{1 + \varphi_k^T(t) P_k(t) \varphi_k(t)} \leq 0 \quad (21)$$

From (14)

$$\lambda_{\min}(P_{k+1}^{-1}(t)) \geq \lambda_{\min}(P_k^{-1}(t)) \geq \dots \geq \lambda_{\min}(P_0^{-1}(t)) \quad (22)$$

From (8)

$$\lambda_{\max}(P_{k+1}(t)) \leq \lambda_{\max}(P_k(t)) \leq \dots \leq \lambda_{\max}(P_0(t)) \quad (23)$$

Using (21), together with (22) and (23), yields

$$\square \tilde{w}_k(t) \square^2 \leq \frac{\lambda_{\max}(P_0^{-1}(t))}{\lambda_{\min}(P_0^{-1}(t))} \square \tilde{w}_0(t) \square^2 \quad (24)$$

which ensures property (i).

Since $\tilde{w}_0(t)$ is bounded for all $t \in \{0, 1, \dots, T\}$ and $k \in \{0, 1, \dots\}$, as a result, (23) indicates the boundedness of $\tilde{w}_k(t)$.

(ii) Summing (21) from 0 to k results in

$$\sum_{i=0}^k \beta_i(t) \frac{(|e_i(t)| - \bar{\varepsilon})^2}{1 + \varphi_i^T(t) P_i(t) \varphi_i(t)} \leq V_0(t) \quad (25)$$

Since $V_0(t)$ is bounded, therefore

$$\lim_{k \rightarrow \infty} \beta_k(t) \frac{(|e_k(t)| - \bar{\varepsilon})^2}{1 + \varphi_k^T(t) P_k(t) \varphi_k(t)} = 0 \quad (26)$$

Thus property (ii) is established. It is observed that the presented algorithm is comparable with the properties of recursive algorithms in [22]. The fundamental results in Lemma 1 are presented in the iteration-domain, while those for recursive algorithms are given in

time-domain.

Theorem 1. The iterative learning neural network described by (7), (8) and (9) ensures that for $t=0,1,\dots,N$,

$$\limsup_{k \rightarrow \infty} |e_k(t)| \leq \bar{\varepsilon} \quad (27)$$

Proof. With the convergence result (ii) of Lemma 1, the theorem follows directly from the boundedness of $\varphi_k(t)$.

The identification error is converged to the neighborhood of the zero over the entire interval. The neighborhood radius depends on the accuracy of neural network modeling.

4. Simulation Results and Discussions

In the previous sections, we have presented the theoretical results about (1), in addition to showing that the least square with dead zone algorithm is given as in (8) and (9). In this section, we present two illustrative examples, which could substantiate superiority that time-invariant network can not reach.

4.1. First-Order System Identification Case Study

For illustration and comparison, a nonlinear system using HONN is considered. The dynamical system is described as:

$$y(t) = 0.72 * \sin(0.2 * \pi * t) * y(t-1) + 0.5 * \sin(0.2 * \pi * t) * y(t-2) * xu(t-1) + 0.2 * \sin(0.2 * \pi * t) * xu(t-2)^2 + 0.4 * \sin(0.2 * \pi * t) * xu(t-3) \quad (28)$$

System identification includes finding the relation between the input and output of the system. Here, $y(t)$ is the target output of the system. HONN model parameters are updated according to prediction error $e(t)$. The identification is performed for two cases. In the first case, the above system is trained with time-variant HONN network and, in the second case the same system is also trained with time-invariant HONN network. The least squares algorithm with dead-zone algorithm is applied for updating of the parameter values of the HONN. The initial values of the parameters of the HONN are generated in the interval $[-1, 1]$. For the identification of the system, the following incentive signal is used:

$$xu(t) = \begin{cases} \sin(\pi * t / 25), & t < 25 \\ 1.0, & 25 \leq t < 50 \\ -1.0, & 50 \leq t < 75 \\ 0.3 * \sin(\pi * t / 25) + 0.1 * \sin(\pi * t / 32) + 0.6 * \sin(\pi * t / 10), & 75 \leq t < 100 \end{cases} \quad (29)$$

When $k=63$, the convergent index satisfies $J_k = 1 \times 10^{-5}$. From Figure1 and Figure2 we can see when the number of parameters of the time-variant HONN is $N=48$, the value of RMSE of identification obtained after iterative times 63 was 8.02×10^{-6} . Figure1 show the three dimensional weights of the neural network.

Furthermore, in order to verify the effectiveness of the time-varying neural network mode, $xu(t) = 0.15 \sin(2t * \pi / 250) + 0.22 \sin(2t * \pi / 35) + 0.2$ is introduced as testing signals. The root mean square error (RMSE) as a performance criterion is given in (29):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{time} (y(t+1) - y_p(t+1))^2}{time}} \quad (30)$$

Figure3 shows us the $RMSE=8.56 \times 10^{-5}$ with the testing data using the training model. The tracking trajectories of the dynamical system are depicted in Figure4. So $y_p(t)$ is the

predicted output with HONN models and $y_k(t)$ is the output of the time-varying neural network.

At the same time, using the parameter update algorithm derived above, the learning of the parameter values of time-invariant HONN for the given input signals is also carried out. The training is continued for 100 epochs with 100 times steps in each epoch. As a result of training, when the number of parameters is $N=48$, the value of RMSE of identification obtained after training was 0.1796. RMSE value of HONN for test data was 0.3329. When the number of parameters is taken 72, the value of RMSE of identification obtained after training was 0.1678. RMSE value of HONN for test data was 0.2894. Figure5 demonstrates the error between true value and the testing value. From the Figure5, we know the time-varying neural network approximates the value accurately. Figure6 shows the evolution of RMSE values over 100 epochs. For comparison, it is worth pointing out that the data trained along the iterative axis with the system approximate better than the one trained along the time axis.

4.2. Second-order System Identification Case Study

The nonlinear dynamic system to be identified is described by

$$y(t) = 0.02 * \sin(0.2 * \pi * t) * (y(t-1) * y(t-2) * y(t-3) * xu(t-2) * (y(t-3) - 1) + xu(t-1)) / (1 + y(t-3)^2 + y(t-2)^2) \quad (31)$$

The equilibrium state of the unforced system given by (30) is (0,0). The training input $xu(t)$ is uniformly selected in the $[-1,1]$ range, and the testing input $xu(t)$ is given by (28). The initial values of the parameters of the HONN are generated in the interval $[0,1]$.

The output of the system depends on two previous output values and three previous input values. The value of $xu(t)$ is generated in the interval $[-1,1]$, $y(1)=0$, $y(2)=0$, $y(3)=0$. The training is continued for time=100. As a performance criterion, the root mean square error (RMSE) given in (30) is used. We set $RMSE < 1e-5$, $e1=1e-6$. As a result of training, when the number of parameters of the time-variant HONN is $N=8$, the value of RMSE of identification obtained after iterative times $k=6$ was $8.02 * e-6$. RMSE value of time-variant HONN for test data was $6.5 * e-5$.

Figure7 demonstrates the identification results, where the solid line denotes the output of the system, and the dashed line denotes the HONN output. Figure8 shows the time-varying parameter along the iterative axis. Figure9 shows the RMSE values obtained during learning. Training RMSE results with HONN model for system identification are shown in Figure10.

However, using the parameter update algorithm derived above, the learning of the parameter values of time-invariant HONN for the given input signals is also carried out. The training is continued for 100 epochs with 100 times steps in each epoch. As a result of training, when the number of parameters is $N=32$, the value of RMSE of identification obtained after training was 0.0062. RMSE value of HONN for test data was 0.0097. When the number of parameters is taken 48, the value of RMSE of identification obtained after training was 0.0057. RMSE value of HONN for test data was 0.009. Clearly, from Figure11 we can see the result does not approximate to zero.

4.3. Image Compression Identification Problems

Image compression may be lossy or lossless. Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics. Iterative learning with dead zone algorithm using time-varying neural network can also apply to the image compression problems.

This example presents time-varying RBF neural networks with capabilities applicable to compress an image size of $98 * 120$ pixels. For neural network training, the subtracting clustering algorithm is used, and Iterative Learning Least Squares (ILLS)

algorithm is applied for updating time-varying weights. The results show that the time-varying weights of time-varying RBF neural networks get a good compressed effect, on the contrary, the time-invariant RBF neural networks with recursive least squares algorithm cannot dispose the image. The image is divided into 3 big blocks, and in each block we get 98×40 pixels. RBF network has three key parameters: Basic function center, basic function width and weights. Here we suppose that the basic function widths and centers are unchanged.

The input of the system $[u(t), y(t)]^T$, $u(t)$ is the position of each pixel and $y(t)$ presents the value of the pixel. We select x of $[0, 4]$, with an interval of 0.001 spaced from the coordinates of 3920 points. $y_p(t)$ are the output of the network, $e(t) = y(t) - y_p(t)$, $w(0) = 0$. Using subtracting clustering algorithm for constructing neural networks, we get 6 time-varying weights in each block. By storing 6 time-varying weights instead of store 98×40 pixels, the storage capacity is greatly reduced. In Figure 12 the original image is shown. It tells us the recovered image is well matched with the original image.

5. Conclusion

This paper applied time-varying neural network to model and identify of nonlinear systems and it fully demonstrated effectiveness of iterative learning methodology. The iterative learning algorithm was proposed to modify the weights along iteration axis, using the data obtained in the system for a fixed time. The zero identification error was achieved over the entire interval. The iterative learning least squares algorithm has been analytically shown to be applicable for the neural network training. Time-varying dynamical system using iterative learning identification algorithm can be applied to many meaningful aspects such as the extracting of compressed image. Future work may lie in the application of the time-varying dynamical system of iterative learning algorithm to engineering problems, the study on the extracting of video image in the solution.

Figure 1. Time-varying Parameter

Figure 2. Convergence History for Example 1

Figure 3. RMSE Values Obtained During Learning with Time-varying System

Figure 4. Test Results with HONN Model for System Identification

Figure 5. Results of Identification, where Green Line Denotes the Output of the Plant, Red line Denotes the HONN Output

Figure 6. RMSE Values Obtained during Learning with Time-invariant System

Figure 7. The Training Results with HONN Model for System Identification where the Red Line is the System Output, Green Line is the HONN Output

Figure 8. Time-varying Parameters

Figure 9. Convergence History for Example 2

Figure10. RMSE Values Obtained During Learning

Figure 11. RMSE Values Obtained During Learning

Figure 12. The Original Image with Time-invariant Network

Acknowledgement

This study was partly supported by National Natural Science Foundation of China (No.61272315, No. 60842009), Zhejiang Provincial Natural Science Foundation (No. Y1110342), Zhejiang Provincial Science and Technology Department of International Cooperation Project (No.2012C24030),

References

- [1] S. Haykin, Neural Networks: A Comprehensive Foundation, Upper Saddle River, New Jersey: Prentice-Hall, 1999.
- [2] M. Sun and X. He, "Iterative learning identification and control of discrete time-varying systems", Proceedings of the 26th Chinese Control Conference, Kunming, China, Jul. pp. 520-524, 2007.
- [3] M. Sun, "Iterative learning neurocomputing," in Proceedings of 2009 International Conference on Wireless Networks and Information Systems, Shanghai, China, Dec. 28-29, pp. 158-161, 2009.
- [4] Garces F, A dynamic neural network approach.. London: Springer-Verlag, 2003.
- [5] Y. Zhang, G. Ruan "Robustness analysis of the Zhang neural network for online time-varying quadratic optimization," J. Phys. A, Math. Theory, Vol. 43, no. 24, pp. 1639-1643, Jun, 2010.
- [6] Hui-juan Lu. "Dissimilarity based ensemble of extreme learning machine for gene expression data Classification," Neurocomputing. vol. 12, pp. 130-139, 2013.
- [7] Hui-juan Lu. "ELM-based gene expression classification with misclassification cost," Neural Comput & Applic. vol. 10, pp. 2079-2089, 2013.
- [8] Guang-bin Huang. "Extreme learning machine: theory and applications," Neurocomputing. vol. 70, pp. 489-501, 2006.
- [9] D. Guo, Y. Zhang, "Zhang Neural Network for Online Solution of Time-Varying Linear Matrix Inequality Aided With an Equality Conversion," in IEEE Transactions on Neural Networks and Learning Systems, pp. 370-382, May, 2014.
- [10] Guang-bin Huang. "Optimization method based extreme learning machine for classification," Neurocomputing. Vol. 74, pp. 155-163, 2010.
- [11] Y. Zhang, "Different Zhang functions leading to different ZNN models illustrated via time-varying matrix square roots finding," Expert Systems with Applications, pp. 4393-4403, 2013.
- [12] M. Sun, "Time-Varying Neurocomputing: An Iterative learning perspective," in Proceedings of International Conference on intelligent computing, pp. 18-26, 2012.
- [13] S. Li, "Nonlinearly Activated Neural Network for Solving Time-Varying Complex Sylvester Equation," IEEE Transactions on Cybernetics. vol. 44, pp. 1397 - 1407, Aug, 2014.
- [14] H. Zeng, "Improved Conditions for Passivity of Neural Networks With a Time-Varying Delay," IEEE Transactions on Cybernetics. vol. 44, pp. 785 - 792, Aug, 2014.
- [15] Z. WenleX, "MIMO system identification with extended MADALINE neural network trained by Levenberg-Marquardt Method," in Proceedings of the 26th Chinese Control and Decision Conference, Changsha, China, pp. 1191-1196, May, 2014.
- [16] F. Pan, "Application of neural network for time-varying large delay system," in Proceedings of Machine Learning and Cybernetics, USA, pp. 1191-1196, Aug, 2004.

Authors

Kong Ying was born in Hangzhou, China, in 1980. She received the B.S. Degree in computer science and technology from Zhejiang Gongshang University, Hangzhou, China, in 2003. She received the M.S. Degree in computer technology from Zhejiang university in 2008. She is currently working toward Ph.D. degree in iterative learning systems at the school of Information and Technology , Zhejiang University of Technology, Hangzhou, China.

Her main researches include neural network, pattern recognition, intelligent information processing.

Lu Hui-juan is a teacher of China Jiliang university. She received B.S.

Degree in industrial electrical automation from China University of Mining and Technology,