Journal of Mathematical Research with Applications
Jan., 2014, Vol. 34, No. 1, pp. 114–126
DOI:10.3770/j.issn:2095-2651.2014.01.012
Http://jmre.dlut.edu.cn

A Modified Gradient-Based Neuro-Fuzzy Learning Algorithm for Pi-Sigma Network Based on First-Order Takagi-Sugeno System

Yan LIU[1,2], Jie YANG[1], Dakun YANG[1], Wei WU[1,*]

1. *School of Mathematical Sciences, Dalian University of Technology, Liaoning* 116024*, P. R. China;*
2. *School of Information Science and Engineering, Dalian Polytechnic University, Liaoning* 116034*, P. R. China*

Abstract This paper presents a Pi-Sigma network to identify first-order Tagaki-Sugeno (T-S) fuzzy inference system and proposes a simplified gradient-based neuro-fuzzy learning algorithm. A comprehensive study on the weak and strong convergence for the learning method is made, which indicates that the sequence of error function goes to a fixed value, and the gradient of the error function goes to zero, respectively.

Keywords first-order Takagi-Sugeno inference system; Pi-Sigma network; convergence.

MR(2010) Subject Classification 47S40; 93C42; 82C32

1. Introduction

The combination of neural networks and fuzzy set theory is showing special promise and is a growing hot topic in recent years. The Takagi-Sugeno (T-S) fuzzy model was proposed by Takagi-Sugeno [1], which is characterized as a set of IF-THEN rules. Pi-Sigma Network (PSN) [2] is a class of high-order feedforward network and is known to provide inherently more powerful mapping abilities than traditional feedforward neural networks.

A hybrid Pi-Sigma network was introduced by Jin [3], which is capable of dealing with the nonlinear systems more efficiently. Combination of the benefits of high-order network and Takagi-Sugeno inference system makes Pi-Sigma network have a simple structure, less training epoch and fast computational speed [4]. Despite numerous works dealing with T-S systems analysis on identification and stability, fewer studies have been done concerning the convergence of the learning process.

In this paper, a comprehensive study on convergence results for Pi-Sigma network based on first-order T-S system is presented. In particular, the monotonicity of the error function in the learning iteration is proven. Both the weak and strong convergence results are obtained,

Received July 19, 2012; Accepted November 25, 2012
Supported by the Fundamental Research Funds for the Central Universities, the National Natural Science Foundation of China (Grant No. 11171367) and the Youth Foundation of Dalian Polytechnic University (Grant No. QNJJ 201308).
* Corresponding author
E-mail address: wuweiw@dlut.edu.cn (Wei WU)

indicating that the gradient of the error function goes to zero and the weight sequence goes to a fixed point, respectively.

The remainder of this paper is organized as follows. A brief introduction of first-order Takagi-Sugeno system and Pi-Sigma neural network is proposed in the next section. Section 3 demonstrates our modified neuro-fuzzy learning algorithm based on gradient method. The main convergence results are provided in Section 4. Section 5 presents a proof of the convergence theorem. Some brief conclusions are drawn in Section 6.

2. First-order Takagi-Sugeno inference system and high-order network

2.1. First-order Takagi-Sugeno inference system

A general fuzzy system, which is a T-S model, is comprised of a set of IF-THEN fuzzy rules having the following form:

$$R^i : \text{If } x_1 \text{ is } A_{1i} \text{ and } x_2 \text{ is } A_{2i} \text{ and} \ldots \text{and } x_m \text{ is } A_{mi} \text{ then } y_i = f_i(\cdot), \tag{1}$$

where R^i $(i = 1, 2, \ldots, n)$ denotes the i-th implication, n is the number of the fuzzy implications of the fuzzy model, $x_1, \ldots, x_m$ are the premise variables, $f_i(\cdot)$ is the consequence of the i-th implication, which is a nonlinear or linear function of the premises, and A_{li} is the fuzzy subset whose membership function is continuous piecewise-polynomial function.

In the first-order Takagi-Sugeno system (T-S1), the output function $f_i(\cdot)$ is a first order polynomial of the input variables $x_1, \ldots, x_m$ and the corresponding output y_i is determined by [5, 6]

$$y_i = p_{0i} + p_{1i}x_1 + \cdots p_{mi}x_m. \tag{2}$$

Given an input $\mathbf{x} = (x_1, x_2, \ldots, x_m)$, the final output of the fuzzy model is expressed by

$$y = \sum_{i=1}^{n} h_i y_i, \tag{3}$$

where h_i is the overall truth value of the premises of the i-th implication calculated as

$$h_i = A_{1i}(x_1)A_{2i}(x_2)\ldots A_{mi}(x_m) = \prod_{l=1}^{m} A_{li}(x_l). \tag{4}$$

We mention that there is another form of (3), that is [7],

$$y = \Big(\sum_{i=1}^{n} h_i y_i\Big) / \Big(\sum_{i=1}^{n} h_i\Big). \tag{5}$$

For simplicity of learning, a common strategy is to obtain the fuzzy consequence without computing the center of gravity [8]. Therefore, we adopt the form of (3) throughout our discussions.

2.2. Pi-Sigma neural network

The conventional feed-forward neural network has summary nodes which is difficult to identify some complex problem. A hybrid Pi-Sigma neural network is shown in Figure 1. In this

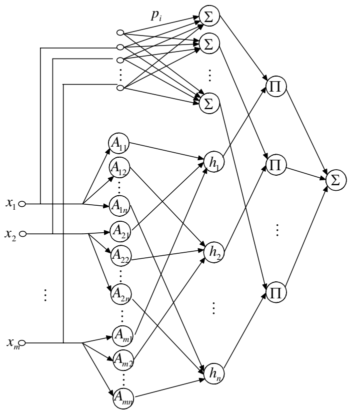

Figure 1 Topological structure of the first-order Takagi-Sugeno inference system

high-order network, Σ denotes the summing neurons, Π denotes the product neurons. The output of pi-sigma neural network is

$$y=\sum_{i=1}^{n}h_iy_i=\sum_{i=1}^{n}\Big(\prod_{l=1}^{m}\mu_{A_{li}}(x_l)\Big)(p_{0i}+p_{1i}x_1+\cdots+p_{mi}x_m)=\mathbf{hPx}, \tag{6}$$

where " $\cdot$ " denotes the usual inner product, $\mathbf{p}_i=(p_{0i},p_{1i},\ldots,p_{mi})^T$, $i=1,2,\ldots,n$, $\mathbf{h}=(h_1,h_2,\ldots,h_n)$, $\mathbf{P}=(\mathbf{p}_1^T,\mathbf{p}_2^T,\ldots,\mathbf{p}_n^T)^T$, $\mathbf{x}=(1,x_1,x_2,\ldots,x_m)^T$. From (6), it can be seen that this Pi-Sigma network is one form of T-S type fuzzy system. For fuzzy system implemented by this type network, the degree of membership function and parameters can be updated indirectly. For more efficient identification of nonlinear systems, Gaussian membership function is commonly used for the fuzzy judgment "x_l *is* A_{li}" which is defined by

$$A_{li}(x_l)=\exp\big(-(x_l-a_{li})^2/\sigma_{li}^2\big)=\exp\big(-(x_l-a_{li})^2b_{li}^2\big), \tag{7}$$

where a_{li} is the center of $A_{li}(x_l)$, and σ_{li} is the width of $A_{li}(x_l)$, b_{li} is the reciprocal of $\sigma_{li}(x_l), i=1,2,\ldots,n, l=1,2,\ldots,m$. What we need is to preset some initial weights, which will be updated to their optimal values when some learning algorithm is implemented.

3. Modified gradient-based neuro-fuzzy learning algorithm

Let us introduce an operator "$\odot$" for the description of the learning method.

Definition 3.1 ([8]) *Let* $\mathbf{u}=(u_1,u_2,\ldots,u_n)^T\in\mathbb{R}^n$, $\mathbf{v}=(v_1,v_2,\ldots,v_n)^T\in\mathbb{R}^n$. *Define the operator "$\odot$" by*

$$\mathbf{u}\odot\mathbf{v}=(u_1v_1,u_2v_2,\ldots,u_nv_n)^T\in\mathbb{R}^n.$$

It is easy to verify the following properties of the operator "$\odot$":

1) $\|\mathbf{u}\odot\mathbf{v}\|\le\|\mathbf{u}\|\|\mathbf{v}\|$,
2) $(\mathbf{u}\odot\mathbf{v})\cdot(\mathbf{x}\odot\mathbf{y})=(\mathbf{u}\odot\mathbf{v}\odot\mathbf{x})\cdot\mathbf{y}$,
3) $(\mathbf{u}+\mathbf{v})\odot\mathbf{x}=\mathbf{u}\odot\mathbf{x}+\mathbf{v}\odot\mathbf{x}$,

where $\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, and "$\cdot$" and "$\|\ \|$"represent the usual inner product and Euclidean norm, respectively.

Suppose that the training sample set is $\{\mathbf{x}^j, O^j\}_{j=1}^J \subset \mathbb{R}^m \times \mathbb{R}$ for Pi-Sigma network, where $\mathbf{x}^j$ and O^j are the input and the corresponding ideal output of the j-th sample, respectively. The fuzzy rules are provided by (1). We denote the weight vector connecting the input layer and the Σ layer by $\mathbf{p}_i = (p_{0i}, p_{1i}, \ldots, p_{mi})^T, (i = 1, 2, \ldots, n)$ (see Figure 1). Similarly, we denote the centers and the reciprocals of the widths of the corresponding Gaussian membership functions by

$$\mathbf{a}_i = (a_{1i}, a_{2i}, \ldots, a_{mi})^T,$$

$$\mathbf{b}_i = (b_{1i}, b_{2i}, \ldots, b_{mi})^T = (\frac{1}{\sigma_{1i}}, \frac{1}{\sigma_{2i}}, \ldots, \frac{1}{\sigma_{mi}})^T, \quad 1 \le i \le n, \tag{8}$$

respectively, and take them as the weight vector connecting the input layer and membership layer. For simplicity, all parameters are incorporated into a weight vector

$$\mathbf{W} = \left(\mathbf{p}_1^T, \mathbf{p}_2^T, \ldots, \mathbf{p}_n^T, \mathbf{a}_1^T, \ldots, \mathbf{a}_n^T, \mathbf{b}_1^T, \ldots, \mathbf{b}_n^T\right)^T. \tag{9}$$

The error function is defined as

$$E(\mathbf{W}) = \frac{1}{2}\sum_{j=1}^{J}(y^j - O^j)^2 = \sum_{j=1}^{J} g_j(\sum_{i=1}^{n} h_i^j(\mathbf{p}_i \cdot \mathbf{x}^j)) = \sum_{j=1}^{J} g_j(\mathbf{h}^j \mathbf{P} \mathbf{x}^j), \tag{10}$$

where O^j is the desired output for the j-th training pattern $\mathbf{x}^j$, y^j is the corresponding fuzzy reasoning result, J is the number of training patterns, and

$$\mathbf{h}^j = (h_1^j, h_2^j, \ldots, h_n^j)^T = \mathbf{h}(\mathbf{x}^j), \ g_j(t) = \frac{1}{2}(t - O^j)^2, \quad t \in \mathbb{R}, \ 1 \le j \le J. \tag{11}$$

The purpose of the network learning is to find $\mathbf{W}^*$ such that

$$E(\mathbf{W}^*) = \min E(\mathbf{W}). \tag{12}$$

The gradient descent method is often used to solve this optimization problem.

Remark 3.1 ([8]) Due to this simple simplification, the differentiation with respect to the denominator is avoided, and the cost of calculating the gradient of the error function is reduced.

Let us describe our modified gradient-based neuro-fuzzy learning algorithm. Noting (8) is valid, then we have

$$h_q^j = \prod_{l=1}^{m} A_{lq}(x_l^j) = \prod_{l=1}^{m} \exp\left(-(x_l^j - a_{lq})^2 b_{lq}^2\right) = \exp\left(\sum_{l=1}^{m}\left(-(x_l^j - a_{lq})^2 b_{lq}^2\right)\right). \tag{13}$$

The gradient of the error function $E(\mathbf{W})$ with respect to $\mathbf{p}_i$ is given by

$$\frac{\partial E(\mathbf{W})}{\partial \mathbf{p}_i} = \sum_{j=1}^{J} g_j'(\mathbf{h}^j \mathbf{P} \mathbf{x}^j) h_i^j \mathbf{x}^j. \tag{14}$$

To compute the partial gradient $\frac{\partial E(\mathbf{W})}{\partial \mathbf{a_i}}$, we note, $\forall 1 \le i \le n,\ 1 \le q \le n$

$$\frac{\partial h_q^j}{\partial \mathbf{a}_i} = \frac{\partial \exp\Big(\sum\limits_{l=1}^{m}\big(-(x_l^j - a_{lq})^2 b_{lq}^2\big)\Big)}{\partial \mathbf{a}_i} = \begin{cases} \dfrac{\partial \exp\Big(\sum\limits_{l=1}^{m}\big(-(x_l^j - a_{li})^2 b_{li}^2\big)\Big)}{\partial \mathbf{a}_i}, & q = i,\\ 0, & q \ne i, \end{cases} \tag{15}$$

and

$$\begin{aligned} \frac{\partial \exp\Big(\sum\limits_{l=1}^{m}(-(x_l^j - a_{li})^2 b_{li}^2)\Big)}{\partial \mathbf{a}_i} &= (2h_i^j(x_1^j - a_{1i})b_{1i}^2, \cdots, 2h_i^j(x_m^j - a_{mi})b_{mi}^2)^T \\ &= 2h_i^j\big((\mathbf{x}^j - \mathbf{a}_i)\odot \mathbf{b}_i \odot \mathbf{b}_i\big). \end{aligned} \tag{16}$$

It follows from (10), (15), (16), that, for $1 \le i \le n$, the partial gradient of the error function $E(\mathbf{W})$ with respect to $\mathbf{a}_i$ is

$$\begin{aligned} \frac{\partial E(\mathbf{W})}{\partial \mathbf{a}_i} &= \sum_{j=1}^{J} g_j'(\mathbf{h}^j \mathbf{P}\mathbf{x}^j)\Big(\sum_{q=1}^{n}(\mathbf{p}_q \cdot \mathbf{x}^j)\frac{\partial h_q^j}{\partial \mathbf{a}_i}\Big) \\ &= 2\sum_{j=1}^{J} g_j'(\mathbf{h}^j \mathbf{P}\mathbf{x}^j)(\mathbf{p}_i \cdot \mathbf{x}^j)h_i^j\big((\mathbf{x}^j - \mathbf{a}_i)\odot \mathbf{b}_i \odot \mathbf{b}_i\big). \end{aligned} \tag{17}$$

Similarly, for $1 \le i \le n$, the partial gradient of the error function $E(\mathbf{W})$ with respect to $\mathbf{b}_i$ is

$$\begin{aligned} \frac{\partial E(\mathbf{W})}{\partial \mathbf{b}_i} &= \sum_{j=1}^{J} g_j'(\mathbf{h}^j \mathbf{P}\mathbf{x}^j)\Big(\sum_{q=1}^{n}(\mathbf{p}_q \cdot \mathbf{x}^j)\frac{\partial h_q^j}{\partial \mathbf{b}_i}\Big) \\ &= -2\sum_{j=1}^{J} g_j'(\mathbf{h}^j \mathbf{P}\mathbf{x}^j)(\mathbf{p}_i \cdot \mathbf{x}^j)h_i^j\big((\mathbf{x}^j - \mathbf{a}_i)\odot (\mathbf{x}^j - \mathbf{a}_i)\odot \mathbf{b}_i\big). \end{aligned} \tag{18}$$

Combined with (14), (17) and (18), the gradient of the error function $E(\mathbf{W})$ with respect to $\mathbf{W}$ is constructed as follows

$$\begin{aligned} \frac{\partial E(\mathbf{W})}{\partial \mathbf{W}} = \Big(&(\frac{\partial E(\mathbf{W})}{\partial \mathbf{p}_1})^T, \ldots, (\frac{\partial E(\mathbf{W})}{\partial \mathbf{p}_n})^T, (\frac{\partial E(\mathbf{W})}{\partial \mathbf{a}_1})^T, \ldots, (\frac{\partial E(\mathbf{W})}{\partial \mathbf{a}_n})^T, \\ &(\frac{\partial E(\mathbf{W})}{\partial \mathbf{b}_1})^T, \ldots, (\frac{\partial E(\mathbf{W})}{\partial \mathbf{b}_n})^T\Big)^T. \end{aligned} \tag{19}$$

Preset an arbitrary initial value $\mathbf{W}^0$, the weights are updated in the following fashion based on the modified neuro-fuzzy learning algorithm

$$\mathbf{W}^{k+1} = \mathbf{W}^k + \Delta \mathbf{W}^k, \quad k = 0, 1, 2, \ldots, \tag{20}$$

where

$$\Delta \mathbf{W}^k = \big((\Delta \mathbf{p}_1^k)^T, \ldots, (\Delta \mathbf{p}_n^k)^T, (\Delta \mathbf{a}_1^k)^T, \ldots, (\Delta \mathbf{a}_n^k)^T, (\Delta \mathbf{b}_1^k)^T, \ldots, (\Delta \mathbf{b}_n^k)^T\big)^T,$$

and

$$\Delta \mathbf{p}_i^k = -\eta\frac{\partial E(\mathbf{W})}{\partial \mathbf{p}_i}, \ \Delta \mathbf{a}_i^k = -\eta\frac{\partial E(\mathbf{W})}{\partial \mathbf{a}_i}, \ \Delta \mathbf{b}_i^k = -\eta\frac{\partial E(\mathbf{W})}{\partial \mathbf{b}_i}, \quad 1 \le i \le n, \tag{21}$$

$\eta > 0$ is a constant learning rate. (20) is also given by

$$\mathbf{W}^{k+1} = \mathbf{W}^k - \eta\frac{\partial E(\mathbf{W})}{\partial \mathbf{W}}, \quad k = 0, 1, 2, \ldots. \tag{22}$$

4. Convergence theorem

To analyze the convergence of the algorithm, we need the following assumption:

(A) There exists a constant $C_0 > 0$ such that $\|\mathbf{p}_i^k\| \leq C_0;\ \|\mathbf{a}_i^k\| \leq C_0;\ \|\mathbf{b}_i^k\| \leq C_0$ for all $i = 1; 2; :::; n$, $k = 1; 2; :::$

Let us specify some constants to be used in our convergence analysis as follows:

$$
\begin{aligned}
M &= \max_{1 \leq j \leq J} \{\|\mathbf{x}^j\|; \|O^j\|\}; \\
C_1 &= \max\{C_0 + M; (C_0 + M)C_0\}; \\
C_2 &= 2JC_0C_1(nC_0C_1 + C_1)\max\{C_0^1; C_1^2\} + \\
&\quad 2C_0^2C_1^2(nC_0C_1 + C_1) + 2JC_0C_1^2(C_0 + C_1)(nC_0C_1 + C_1); \\
C_3 &= J(nC_0C_1 + C_1)\max\{\frac{1}{2}; 4C_0^2C_1^2; 4C_1^4\}; \\
C_4 &= 4nJC_1^2\max\{C_0^2C_1^2; C_1^2; 1\}; \\
C_5 &= C_2 + C_3 + C_4
\end{aligned}
$$

where $\mathbf{x}^j$ is the j-th given training pattern, O^j is the corresponding desired output, n and J are the numbers of the fuzzy rules and the training patterns, respectively.

Theorem *If Assumption (A) is valid, the error function $E(\mathbf{W})$ is defined in (10) and the learning rate ´ is chosen such that $0 <$ ´ $< \frac{1}{C_5}$ is satisfied, then starting from an arbitrary initial value $\mathbf{W}^0$, the learning sequence $\{\mathbf{W}^k\}$ is generated by (22) and (19), and we have*

(i) $E(\mathbf{W}^{k+1}) \leq E(\mathbf{W}^k)$, $k = 0; 1; 2; :::$; *there exists* $E^* > 0$ *such that* $\lim_{k\to\infty} E(\mathbf{W}^k) = E^*$;

(ii) $\lim_{k\to\infty} E_{\mathbf{W}}(\mathbf{W}^k) = 0$.

5. Proof of the convergence theorem

The proof is divided into two parts dealing with Statements (i) and (ii), respectively.

Proof of Statement (i) For any $1 \leq j \leq J$, $1 \leq i \leq n$ and $k = 0; 1; 2; :::$ we define the following notations for convergence:

$$
\Phi_0^{k,j} = \mathbf{h}^{k,j}\mathbf{P}^k\mathbf{x}^j;\ \Psi^{k,j} = h^{k+1,j} - h^{k,j};\ »_i^{k,j} = \mathbf{x}^j - \mathbf{a}_i^k;\ \Phi_i^{k,j} = »_i^{k,j} \odot \mathbf{b}_i^k: \tag{23}
$$

Noticing error function (10) is valid, and applying the Taylor mean value theorem with Lagrange remainder, we have

$$
\begin{aligned}
&E(\mathbf{W}^{k+1}) - E(\mathbf{W}^k) \\
&\quad = \sum_{j=1}^{J} \left(g_j(\Phi_0^{k+1,j}) - g_j(\Phi_0^{k,j})\right) \\
&\quad = \sum_{j=1}^{J} \left[g_j'(\Phi_0^{k,j})(\mathbf{h}^{k+1,j}\mathbf{P}^{k+1}\mathbf{x}^j - \mathbf{h}^{k,j}\mathbf{P}^k\mathbf{x}^j) + \frac{1}{2}g_j''(s_{k,j})(\Phi_0^{k+1,j} - \Phi_0^{k,j})^2\right]
\end{aligned}
$$

$$= \sum_{j=1}^{J} \Big[g_j'(\Phi_0^{k,j}) \big(\mathbf{h}^{k,j} \Delta \mathbf{P}^{k+1} \mathbf{x}^j + \Psi^{k,j} \mathbf{P}^{k+1} \mathbf{x}^j + \Psi^{k,j} \Delta \mathbf{P}^k \mathbf{x}^j \big) \Big] +$$
$$\frac{1}{2} \sum_{j=1}^{J} g_j''(s_{k,j}) (\Phi_0^{k+1,j} - \Phi_0^{k,j})^2,$$

where $s_{k,j} \in \mathbb{R}$ is a constant between $\Phi_0^{k,j}$ and $\Phi_0^{k+1,j}$.

Employing (14), we have

$$\begin{aligned} \sum_{j=1}^{J} g_j'(\Phi_0^{k,j}) (\mathbf{h}^{k,j} \Delta \mathbf{P}^k \mathbf{x}^j) &= \sum_{j=1}^{J} g_j'(\Phi_0^{k,j}) \sum_{i=1}^{n} \big(h_i^{k,j} (\Delta \mathbf{p}_i^k)^T \big) \mathbf{x}^j \\ &= \sum_{i=1}^{n} (\Delta \mathbf{p}_i^k)^T \sum_{j=1}^{J} g_j'(\Phi_0^{k,j}) h_i^{k,j} \mathbf{x}^j \\ &= \sum_{i=1}^{n} \frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{p}_i} \cdot \Big(-\eta \frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{p}_i} \Big) \\ &= -\eta \sum_{i=1}^{n} \Big\| \frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{p}_i} \Big\|^2. \end{aligned}$$

Using the Taylor expansion, and noticing $h_i^{t,j} = \exp(-\Phi_i^{t,j} \cdot \Phi_i^{t,j})$, partly similar with the proof of Lemma 2 in [16], we have

$$\begin{aligned} \Psi^{k,j} \mathbf{P}^k \mathbf{x}^j &= \sum_{i=1}^{n} (h_i^{k+1,j} - h_i^{k,j}) (\mathbf{p}_i^k \cdot \mathbf{x}^j) = \sum_{i=1}^{n} (\mathbf{p}_i^k \cdot \mathbf{x}^j) \Big(\exp(-\Phi_i^{k+1,j} \cdot \Phi_i^{k+1,j}) - \exp(-\Phi_i^{k,j} \cdot \Phi_i^{k,j}) \Big) \\ &= \sum_{i=1}^{n} (\mathbf{p}_i^k \cdot \mathbf{x}^j) h_i^{k,j} \big(-(\Phi_i^{k+1,j} \cdot \Phi_i^{k+1,j} - \Phi_i^{k,j} \cdot \Phi_i^{k,j}) \big) + \\ &\quad \frac{1}{2} \sum_{i=1}^{n} (\mathbf{p}_i^k \cdot \mathbf{x}^j) \exp(\widetilde{t}_i^{s,j}) (\Phi_i^{k+1,j} \cdot \Phi_i^{k+1,j} - \Phi_i^{k,j} \cdot \Phi_i^{k,j})^2, \end{aligned} \tag{24}$$

where $\widetilde{t}_i^{s,j}$ lies between $-\Phi_i^{k+1,j} \cdot \Phi_i^{k+1,j}$ and $-\Phi_i^{k,j} \cdot \Phi_i^{k,j}$.

Employing the property 2) of the operator "$\odot$" in the Definition 3.1, we deduce

$$\begin{aligned} &\sum_{i=1}^{n} (\mathbf{p}_i^k \cdot \mathbf{x}^j) h_i^{k,j} \Big(-(\Phi_i^{k+1,j} \cdot \Phi_i^{k+1,j} - \Phi_i^{k,j} \cdot \Phi_i^{k,j}) \Big) \\ &= \sum_{i=1}^{n} (\mathbf{p}_i^k \cdot \mathbf{x}^j) h_i^{k,j} \Big[- \Big(2\Phi_i^{k,j} \cdot (\Phi_i^{k+1,j} - \Phi_i^{k,j}) + (\Phi_i^{k+1,j} - \Phi_i^{k,j}) \cdot (\Phi_i^{k+1,j} - \Phi_i^{k,j}) \Big) \Big] \\ &= -2 \sum_{i=1}^{n} (\mathbf{p}_i^k \cdot \mathbf{x}^j) h_i^{k,j} \big(\Phi_i^{k,j} \cdot (\Phi_i^{k+1,j} - \Phi_i^{k,j}) \big) - \sum_{i=1}^{n} (\mathbf{p}_i^k \cdot \mathbf{x}^j) h_i^{k,j} \| \Phi_i^{k+1,j} - \Phi_i^{k,j} \|^2 \\ &= -2 \sum_{i=1}^{n} (\mathbf{p}_i^k \cdot \mathbf{x}^j) h_i^{k,j} \Big((\xi_i^{k,j} \odot \mathbf{b}_i^k) \cdot \big((-\Delta \mathbf{a}_i^k) \odot \mathbf{b}_i^{k+1} + \xi_i^{k,j} \odot \Delta \mathbf{b}_i^k \big) \Big) - \\ &\quad \sum_{i=1}^{n} (\mathbf{p}_i^k \cdot \mathbf{x}^j) h_i^{k,j} \| \Phi_i^{k+1,j} - \Phi_i^{k,j} \|^2 \end{aligned}$$

$$
\begin{aligned}
&= 2\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}\big(\xi_i^{k,j}\odot\mathbf{b}_i^k\big)\cdot\big(\Delta\mathbf{a}_i^k\odot\mathbf{b}_i^{k+1}\big)-\\
&\quad 2\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}(\xi_i^{k,j}\odot\mathbf{b}_i^k)\cdot(\xi_i^{k,j}\odot\Delta\mathbf{b}_i^k)-\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}\|\Phi_i^{k+1,j}-\Phi_i^{k,j})\|^2\\
&= 2\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}(\xi_i^{k,j}\odot\mathbf{b}_i^k\odot\mathbf{b}_i^k)\cdot\Delta\mathbf{a}_i^k+2\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}(\xi_i^{k,j}\odot\mathbf{b}_i^k\odot\Delta\mathbf{b}_i^k)\cdot\Delta\mathbf{a}_i^k-\\
&\quad 2\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}(\xi_i^{k,j}\odot\xi_i^{k,j}\odot\mathbf{b}_i^k)\cdot\Delta\mathbf{b}_i^k-\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}\|\Phi_i^{k+1,j}-\Phi_i^{k,j})\|^2.
\end{aligned}
$$

So we have

$$
\begin{aligned}
&\sum_{j=1}^{J}g_j'(\Phi_0^{k,j})\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}\big(-(\Phi_i^{k+1,j}\cdot\Phi_i^{k+1,j}-\Phi_i^{k,j}\cdot\Phi_i^{k,j})\big)\\
&\quad= 2\sum_{j=1}^{J}g_j'(\Phi_0^{k,j})\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}(\xi_i^{k,j}\odot\mathbf{b}_i^k\odot\mathbf{b}_i^k)\cdot\Delta\mathbf{a}_i^k+\\
&\qquad 2\sum_{j=1}^{J}g_j'(\Phi_0^{k,j})\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}(\xi_i^{k,j}\odot\mathbf{b}_i^k\odot\Delta\mathbf{b}_i^k)\cdot\Delta\mathbf{a}_i^k-\\
&\qquad 2\sum_{j=1}^{J}g_j'(\Phi_0^{k,j})\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}(\xi_i^{k,j}\odot\xi_i^{k,j}\odot\mathbf{b}_i^k)\cdot\Delta\mathbf{b}_i^k-\sum_{j=1}^{J}g_j'(\Phi_0^{k,j})\delta\\
&\quad=\sum_{i=1}^{n}\frac{\partial E(\mathbf{W}^k)}{\partial\mathbf{a}_i}\cdot\Delta\mathbf{a}_i^k+2\sum_{j=1}^{J}g_j'(\Phi_0^{k,j})\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}(\xi_i^{k,j}\odot\mathbf{b}_i^k\odot\Delta\mathbf{b}_i^k)\cdot\Delta\mathbf{a}_i^k+\\
&\qquad\sum_{i=1}^{n}\frac{\partial E(\mathbf{W}^k)}{\partial\mathbf{b}_i}\cdot\Delta\mathbf{b}_i^k-\sum_{j=1}^{J}g_j'(\Phi_0^{k,j})\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}\|\Phi_i^{k+1,j}-\Phi_i^{k,j})\|^2. \qquad (25)
\end{aligned}
$$

We can easily get $\|\Phi_0^{k,j}\|=\|\Sigma_{i=1}^{n}h_i^{k,j}(\mathbf{p}_i^k\cdot\mathbf{x}^j)\|\le\Sigma_{i=1}^{n}\|\mathbf{p}_i^k\cdot\mathbf{x}^j\|\le\Sigma_{i=1}^{n}\|\mathbf{p}_i^k\|\|\mathbf{x}^j\|=nMC_0$, $\|\xi_i^{k,j}\|=\|\mathbf{x}^j-\mathbf{a}_i^k\|\le M+C_0$. By definition of $g_j(t)$ in (11), it is easy to find that $g_j'(t)=t-O^j$, then we can get $|g_j'(\Phi_0^{k,j})|\le(nMC_0+M)\le nC_0C_1+C_1$.

Together with Assumption (A), and the property 1) of the operator "$\odot$", we get

$$
\begin{aligned}
&2\sum_{j=1}^{J}g_j'(\Phi_0^{k,j})\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}(\xi_i^{k,j}\odot\mathbf{b}_i^k\odot\Delta\mathbf{b}_i^k)\cdot\Delta\mathbf{a}_i^k\\
&\quad\le 2JC_0^2C_1^2(nC_0C_1+C_1)\sum_{i=1}^{n}\|\Delta\mathbf{b}_i^k\|\|\Delta\mathbf{a}_i^k\|\\
&\quad\le JC_0^2C_1^2(nC_0C_1+C_1)\sum_{i=1}^{n}\big(\|\Delta\mathbf{a}_i^k\|^2+\|\Delta\mathbf{b}_i^k\|^2\big), \qquad (26)
\end{aligned}
$$

and

$$
-\sum_{j=1}^{J}g_j'(\Phi_0^{k,j})\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}\|\Phi_i^{k+1,j}-\Phi_i^{k,j})\|^2
$$

$$\begin{aligned}
&\leq C_0C_1(nC_0C_1+C_1)\sum_{j=1}^{J}\sum_{i=1}^{n}\|\Phi_i^{k+1,j}-\Phi_i^{k,j}\|^2\\
&= C_0C_1(nC_0C_1+C_1)\sum_{j=1}^{J}\sum_{i=1}^{n}\|(-\Delta\mathbf{a}_i^k)\odot\mathbf{b}_i^{k+1}+»_i^{k,j}\odot\Delta\mathbf{b}_i^k\|^2\\
&\leq 2JC_0C_1(nC_0C_1+C_1)\sum_{i=1}^{n}\big(C_0^2\|\Delta\mathbf{a}_i^k\|^2+C_1^2\|\Delta\mathbf{b}_i^k\|^2\big)\\
&\leq C_{21}\sum_{i=1}^{n}\big(\|\Delta\mathbf{a}_i^k\|^2+\|\Delta\mathbf{b}_i^k\|^2\big); \qquad (27)
\end{aligned}$$

where $C_{21}=2JC_0C_1(nC_0C_1+C_1)\max\{C_0^2;C_1^2\}$. The combination of (29)–(31) leads to

$$\begin{aligned}
&\sum_{j=1}^{J}g_j'(\Phi_0^{k,j})\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)h_i^{k,j}\big(-(\Phi_i^{k+1,j}\cdot\Phi_i^{k+1,j}-\Phi_i^{k,j}\cdot\Phi_i^{k,j})\big)\\
&\quad\leq\sum_{i=1}^{n}\frac{@E(\mathbf{W}^k)}{@\mathbf{a}_i}\cdot\Delta\mathbf{a}_i^k+\sum_{i=1}^{n}\frac{@E(\mathbf{W}^k)}{@\mathbf{b}_i}\cdot\Delta\mathbf{b}_i^k+C_{22}\sum_{i=1}^{n}\big(\|\Delta\mathbf{a}_i^k\|^2+\|\Delta\mathbf{b}_i^k\|^2\big);
\end{aligned}$$

where $C_{22}=C_{21}+JC_0^2C_1^2(nC_0C_1+C_1)$. Furthermore,

$$\begin{aligned}
&g_j'(\Phi_0^{k,j})\frac{1}{2}\sum_{j=1}^{J}\sum_{i=1}^{n}(\mathbf{p}_i^k\cdot\mathbf{x}^j)\exp(\tilde{t}_i^{s,j})(\Phi_i^{k+1,j}\cdot\Phi_i^{k+1,j}-\Phi_i^{k,j}\cdot\Phi_i^{k,j})^2\\
&\leq\frac{C_0C_1(nC_0C_1+C_1)}{2}\sum_{j=1}^{J}\sum_{i=1}^{n}(\Phi_i^{k+1,j}\cdot\Phi_i^{k+1,j}-\Phi_i^{k,j}\cdot\Phi_i^{k,j})^2\\
&=\frac{C_0C_1(nC_0C_1+C_1)}{2}\sum_{j=1}^{J}\sum_{i=1}^{n}\big[(\Phi_i^{k+1,j}+\Phi_i^{k,j})\cdot(\Phi_i^{k+1,j}-\Phi_i^{k,j})\big]^2\\
&\leq 2C_0C_1^2(nC_0C_1+C_1)\sum_{j=1}^{J}\sum_{i=1}^{n}\|\Phi_i^{k+1,j}-\Phi_i^{k,j}\|^2\\
&\leq C_{23}\sum_{i=1}^{n}\big(\|\Delta\mathbf{a}_i^k\|^2+\|\Delta\mathbf{b}_i^k\|^2\big); \qquad (28)
\end{aligned}$$

where $C_{23}=2JC_0C_1^2(C_0+C_1)(nC_0C_1+C_1)$. Combining (28) and (32) leads to

$$\begin{aligned}
&\sum_{j=1}^{J}g_j'(\Phi_0^{k,j})(\mathbf{\Psi}^{k,j}\mathbf{P}^k\mathbf{x}^j)\\
&\leq\sum_{i=1}^{n}\frac{@E(\mathbf{W}^k)}{@\mathbf{a}_i}\cdot\Delta\mathbf{a}_i^k+\sum_{i=1}^{n}\frac{@E(\mathbf{W}^k)}{@\mathbf{b}_i}\cdot\Delta\mathbf{b}_i^k+C_2\sum_{i=1}^{n}\big(\|\Delta\mathbf{a}_i^k\|^2+\|\Delta\mathbf{b}_i^k\|^2\big)\\
&=-´\sum_{i=1}^{n}\Big(\Big\|\frac{@E(\mathbf{W}^k)}{@\mathbf{a}_i}\Big\|^2+\Big\|\frac{@E(\mathbf{W}^k)}{@\mathbf{b}_i}\Big\|^2\Big)+C_2´^2\sum_{i=1}^{n}\Big(\Big\|\frac{@E(\mathbf{W}^k)}{@\mathbf{a}_i}\Big\|^2+\Big\|\frac{@E(\mathbf{W}^k)}{@\mathbf{b}_i}\Big\|^2\Big);
\end{aligned}$$

where $C_2=C_{22}+C_{23}$.

Notice $\|\Phi_i^{t,j}\|=\|»_i^{t,j}\odot b_i^t\|\leq C_1$, $\|\mathbf{b_i}\|\leq C_0$, $\|»_i\|\leq C_0+M=C_1$. By the Taylor mean value theorem with Lagrange remainder and the properties of Euclidean norm, we get

$$
\begin{aligned}
\|\Psi^{t,j}\|^2 &= \left\| \begin{pmatrix} h_1^{t+1,j} - h_1^{t,j} \\ h_2^{t+1,j} - h_2^{t,j} \\ \cdots \\ h_n^{k+1,j} - h_n^{t,j} \end{pmatrix} \right\|^2 \\
&= \sum_{i=1}^{n} (h_i^{t+1,j} - h_i^{t,j})^2 = \sum_{i=1}^{n} \Big(\exp\big(- \Phi_i^{t+1,j} \cdot \Phi_i^{t+1,j} \big) - \exp\big(- \Phi_i^{t,j} \cdot \Phi_i^{t,j} \big) \Big)^2 \\
&= \sum_{i=1}^{n} \Big(- \exp(\tilde{s}_i^{t,j}) \Big(\big(\Phi_i^{t+1,j} + \Phi_i^{t,j}\big) \cdot \big(\Phi_i^{t+1,j} - \Phi_i^{t,j}\big) \Big)^2 \\
&\le \sum_{i=1}^{n} \big(|2C_1| \big(\xi_i^{t+1,j} \odot b_i^{t+1,j} - \xi_i^{t,j} \odot b_i^{t,j} \big) \big)^2 \\
&= \sum_{i=1}^{n} \Big(|2C_1| \big\| (\xi_i^{t+1,j} - \xi_i^{t,j}) \odot b_i^{t+1,j} + \xi_i^{t,j} \odot (b_i^{t+1,j} - b_i^{t,j}) \big\| \Big)^2 \\
&\le \sum_{i=1}^{n} \big(|2C_1 C_0| \|\xi_i^{t+1,j} - \xi_i^{t,j}\| + 2C_1^2 \|b_i^{t+1} - b_i^{t}\| \big)^2 \\
&\le \sum_{i=1}^{n} \Big(C_{31} (\|\Delta \mathbf{a_i}^t\| + \|\Delta \mathbf{b_i}^t\|) \Big)^2 \le 2C_{31}^2 \sum_{i=1}^{n} (\|\Delta \mathbf{a_i}^t\|^2 + \|\Delta \mathbf{b_i}^t\|^2),
\end{aligned}
$$

where $C_{31} = 2C_1 \max\{C_0, C_1\}$ and $\tilde{s}_i^{t,j}$ lies between $-\Phi_i^{t+1,j} \cdot \Phi_i^{t+1,j}$ and $-\Phi_i^{t,j} \cdot \Phi_i^{t,j}$. A combination of Cauchy-Schwartz inequality and (33) gives

$$
\begin{aligned}
\sum_{j=1}^{J} g_j'(\Phi_0^{k,j}) \left(\Psi^{k,j} \Delta \mathbf{P}^k \mathbf{x}^j \right) =& \sum_{j=1}^{J} g_j'(\Phi_0^{k,j}) \sum_{i=1}^{n} \Psi_i^{k,j} (\Delta \mathbf{p}_i^k)^T \mathbf{x}^j \\
=& \sum_{j=1}^{J} \sum_{i=1}^{n} g_j'(\Phi_0^{k,j}) \Psi_i^{k,j} (\Delta \mathbf{p}_i^k)^T \mathbf{x}^j \\
\le& C_1 (nC_0C_1 + C_1) \sum_{j=1}^{J} \sum_{i=1}^{n} \|\Psi_i^{k,j}\| \|(\Delta \mathbf{p}_i^k)^T\| \\
\le& \frac{C_1 (nC_0C_1 + C_1)}{2} \sum_{j=1}^{J} \sum_{i=1}^{n} (\|\Psi_i^{k,j}\|^2 + \|(\Delta \mathbf{p}_i^k)^T\|^2) \\
\le& J (nC_0C_1 + C_1) C_{31}^2 \sum_{i=1}^{n} \left(\|\Delta \mathbf{a}_i^k\|^2 + \|\Delta \mathbf{b}_i^k\|^2 \right) + \\
& J \frac{(nC_0C_1 + C_1)}{2} \sum_{i=1}^{n} \left(\|(\Delta \mathbf{p}_i^k)^T\|^2 \right. \\
\le& C_3 \Big(\sum_{i=1}^{n} \left(\|(\Delta \mathbf{p}_i^k)^T\|^2 \right. + \sum_{i=1}^{n} \|\Delta \mathbf{a}_i^k\|^2 + \sum_{i=1}^{n} \|\Delta \mathbf{b}_i^k\|^2 \Big) \\
=& C_3 \eta^2 \left\| \frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{W}} \right\|^2,
\end{aligned}
$$

where $C_3 = J(nC_0C_1 + C_1)\max\{\frac{1}{2}, 4C_0^2C_1^2, 4C_1^4\}$.

$g_j''(t) = 1$ is easily deduced from the definition of $g_j(t)$ in (11), and we get

$$
\begin{aligned}
&\frac{1}{2}\sum_{j=1}^{J} g_j''(s_{k,j})(\Phi_0^{k+1,j} - \Phi_0^{k,j})^2\\
&= \frac{1}{2}\sum_{j=1}^{J} \|\Phi_0^{k+1,j} - \Phi_0^{k,j}\|^2\\
&= \frac{1}{2}\sum_{j=1}^{J} \|\mathbf{h}^{k+1,j}\mathbf{P}^{k+1}\mathbf{x}^j - \mathbf{h}^{k,j}\mathbf{P}^{k}\mathbf{x}^j\|^2\\
&= \frac{1}{2}\sum_{j=1}^{J} \|\sum_{i=1}^{n} h_i^{k+1,j}(\mathbf{p}_i^{k+1}\cdot\mathbf{x}^j) - \sum_{i=1}^{n} h_i^{k,j}(\mathbf{p}_i^{k}\cdot\mathbf{x}^j)\|^2\\
&= \frac{1}{2}\sum_{j=1}^{J} \|\sum_{i=1}^{n} (h_i^{k+1,j} - h_i^{k,j})(\mathbf{p}_i^{k+1}\cdot\mathbf{x}^j) + \sum_{i=1}^{n} (h_i^{k+1,j} - h_i^{k,j})(\mathbf{p}_i^{k+1}\cdot\mathbf{x}^j)\|^2\\
&\le \frac{1}{2}\sum_{j=1}^{J}\sum_{i=1}^{n} \|(h_i^{k+1,j} - h_i^{k,j})(\mathbf{p}_i^{k+1}\cdot\mathbf{x}^j) + (h_i^{k+1,j} - h_i^{k,j})(\mathbf{p}_i^{k+1}\cdot\mathbf{x}^j)\|^2\\
&\le \frac{1}{2}\sum_{j=1}^{J}\sum_{i=1}^{n} [\|(h_i^{k+1,j} - h_i^{k,j})(\mathbf{p}_i^{k+1}\cdot\mathbf{x}^j)\|^2 + \|(h_i^{k+1,j} - h_i^{k,j})(\mathbf{p}_i^{k+1}\cdot\mathbf{x}^j)\|^2]\\
&\le C_4\eta^2 \sum_{i=1}^{n}\Big(\Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{p}_i}\Big\|^2 + \Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{a}_i}\Big\|^2 + \Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{b}_i}\Big\|^2\Big)\\
&= C_4\eta^2\Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{W}}\Big\|^2,
\end{aligned}
$$

where $C_4 = 4nJC_1^2\max\{C_0^2C_1^2, C_1^2, 1\}$.

Using the Taylor expansion theorem, for $k = 0, 1, 2, \ldots$ we get

$$
\begin{aligned}
&E(\mathbf{W}^{k+1}) - E(\mathbf{W}^k)\\
&\quad\le -\eta\sum_{i=1}^{n}\Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{p}_i}\Big\|^2 - \eta\sum_{i=1}^{n}\Big(\Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{a}_i}\Big\|^2 + \Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{b}_i}\Big\|^2\Big) +\\
&\qquad C_3\eta^2\Big(\sum_{i=1}^{n}\Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{p}_i}\Big\|^2 + \sum_{i=1}^{n}\Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{a}_i}\Big\|^2 + \sum_{i=1}^{n}\Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{b}_i}\Big\|^2\Big) +\\
&\qquad (C_2 + C_4)\eta^2\Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{W}}\Big\|^2\\
&\quad\le -(\eta - C_5\eta^2)\Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{W}}\Big\|^2,
\end{aligned}
$$

where $C_5 = C_2 + C_3 + C_4$, and $s_{k,j} \in \mathbb{R}$ lies between $\Phi_0^{k,j}$ and $\Phi_0^{k+1,j}$. Write $\beta = \eta - C_5\eta^2$, then

$$
E(\mathbf{W}^{k+1}) \le E(\mathbf{W}^k) - \beta\Big\|\frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{W}}\Big\|^2. \tag{30}
$$

Suppose the learning rate η satisfies

$$
0 < \eta < \frac{1}{C_5}. \tag{31}
$$

Then there holds

$$E(\mathbf{W}^{k+1}) \leq E(\mathbf{W}^k), \quad k = 0, 1, 2, \ldots. \tag{32}$$

From (32), we know the nonnegative sequence $\{E(W^k)\}$ is monotone, adding it is bounded below, hence, there exists $E^* > 0$ such that $\lim_{k\to\infty} E(\mathbf{W}^k) = E^*$. The statement (i) is proved.

Proof of Statement (ii) Using (35), we have

$$E(\mathbf{W}^{k+1}) \leq E(\mathbf{W}^k) - \eta \left\| \frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{W}} \right\|^2 \leq \cdots \leq E(\mathbf{W}^0) - \eta \sum_{t=0}^{k} \left\| \frac{\partial E(\mathbf{W}^t)}{\partial \mathbf{W}} \right\|^2.$$

For $E(\mathbf{W}^{k+1}) \geq 0$, we get

$$\eta \sum_{t=0}^{k} \left\| \frac{\partial E(\mathbf{W}^t)}{\partial \mathbf{W}} \right\|^2 \leq E(\mathbf{W}^0).$$

Set $k \to \infty$, then

$$\sum_{t=0}^{\infty} \left\| \frac{\partial E(\mathbf{W}^t)}{\partial \mathbf{W}} \right\|^2 \leq \frac{1}{\eta} E(\mathbf{W}^0) < \infty.$$

This immediately gives

$$\lim_{k\to\infty} \left\| \frac{\partial E(\mathbf{W}^k)}{\partial \mathbf{W}} \right\| = 0. \tag{33}$$

The Statement (ii) is proved. And this completes the proof of Theorem.

6. Conclusion

First-order Takagi-Sugeno (T-S) system has recently been a powerful practical engineering tool for modeling and control of complex systems. Ref. [7] showed that Pi-Sigma network is capable of dealing with the nonlinear systems more efficiently, and it is a good model for first-order T-S system identification.

We note that the convergence property for Pi-Sigma neural network learning is an interesting research topic which offers an effective guarantee in real application. To further enhance the potential of Pi-Sigma network, a modified gradient-based algorithm based on first-order T-S inference system has been proposed to reduce the computational cost of learning. Our contribution is to provide a rigorous convergence analysis for this learning method, and some convergence results are given which indicate that the gradient of the error function goes to zero and the fuzzy parameter sequence goes to a local minimum of the error function, respectively.

References

[1] T. TAKAGI, M. SUGENO. *Fuzzy identification of systems and its applications to modeling and control.* IEEE Trans. Syst., Man, Cyber., 1985, SMC-15(1): 116–132.

[2] Y. SHIN, J. GHOSH. *The Pi-Sigma networks: an efficient higher-order neural network for pattern classification and function approximation.* in: Proceedings of International Joint Conference on Neural Networks, 1991, **1**: 13–18.

[3] Yaochu JIN, Jingping JIANG, Jing ZHU. *Neural network based fuzzy identification and its application to modeling and control of complex systems.* IEEE Transactions on Systems, Man, and Cybernetics, 1995, **25**(6): 990–997.

[4] Weixin YU, M. Q. LI, Jiao LUO, et al. *Prediction of the mechanical properties of the post-forged Ti–6Al–4V alloy using fuzzy neural network.* Materials and Design, 2012, **31**: 3282–3288.

[5] D. M. LIU, G. NAADIMUTHU, E. S. LEE. *Trajectory tracking in aircraft landing operations management using the adaptive neural fuzzy inference system.* Comput. Math. Appl., 2008, **56**(5): 1322–1327.

[6] I. TURKMEN, K. GUNEY. *Genetic tracker with adaptive neuro-fuzzy inference system for multiple target tracking.* Expert Systems with Applications, 2008, **35**(4): 1657–1667.

[7] K. CHATURVED, M. PANDIT, L. SRIVASTAVA. *Modified neo-fuzzy neuron-based approach for economic and environmental optimal power dispatch.* Applied Soft Computing, 2008, **8**(4): 1428–1438.

[8] Wei WU, Long LI, Jie YANG, et al. *A modified gradient-based neuro-fuzzy learning algorithm and its convergence.* Information Sciences, 2010, **180**(9): 1630–1642.

[9] E. CZOGALA, W. PEDRYCZ. *On identification in fuzzy systems and its applications in control problems.* Fuzzy Sets and Systems, 1981, **6**(1): 73–83.

[10] W. PEDRYCZ. *An identification algorithm in fuzzy relational systems.* Fuzzy Sets and Systems, 1984, **13**(2): 153–167.

[11] Yongyan CAO, P. M. FRANK. *Analysis and Synthesis of Nonlinear Time-Delay Systems via Fuzzy Control Approach.* IEEE Transactions on Fuzzy Systems, 2000, **8**(2): 200–211.

[12] Yuju CHEN, Tsungchuan HUANG, Reychue HWANG. *An effective learning of neural network by using RFBP learning algorithm.* Inform. Sci., 2004, **167**(1-4): 77–86.

[13] Chao ZHANG, Wei WU, Yan XIONG. *Convergence analysis of batch gradient algorithm for three classes of sigma-pi neural networks.* Neural Processing Letters, 2007, **26**(3): 177–189.

[14] Chao ZHANG, Wei WU, Xianhua CHEN, et al. *Convergence of BP algorithm for product unit neural networks with exponential weights.* Neurocomputing, 2008, **72**(1–3): 513–520.

[15] J. S. WANG, C. S. G. LEE. *Self-adaptive neuro-fuzzy inference systems for classification applications.* IEEE Transactions on Fuzzy Systems, 2002, **10**(6): 790–802.

[16] H. ICHIHASHI, I. B. TKSEN. *A neuro-fuzzy approach to data analysis of pairwise comparisons.* Internat. J. Approx. Reason., 1993, **9**(3): 227–248.