SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
ISSN 2054-7390

An Online Gradient Method with Smoothing L_0 Regularization for Pi-Sigma Network

[1]Khidir Shaib Mohamed, [2,3,4]Yousif Shoaib Mohammed
[1]Department of Mathematics Computer, Faculty of Science, Dalanj University, Dalanj, Sudan;
[2]Department of Physics, College of Science & Art, Qassim University, Oklat Al- Skoor, Saudi Arabia;
[3]Physics Department, College of Education, Dalanj University, Dalanj, Sudan;
[4]Physics Department, Africa City for Technology – Khartoum, Sudan
Khshm7@yahoo.com; yshm@yahoo.com

ABSTRACT

The description of this study is to make possibility analysis solution of online gradient method with smoothing L_0 regularization for pi-sigma network training. Due to the effectiveness computational and theoretical analysis are a very important issues to improve the generalization performance of networks and the gradient descent algorithm with regularization is widely used method. However, L_0 regularization is reefed to NP-hard nature problems, which has not differentiable objective functional-penalty term. In this paper to avoid this trick, we use a smoothing function to recover the origin L_0 regularization into smoothing L_0 regularization. Under this condition, the resulting obtained as a good decreases solution when compared with others. The monotonically of the error function, weak and strong convergence theorems are proved.

Keywords: Convergence; Online gradient method; Pi-Sigma networks; Smoothing L_0 regularization.

1 Introduction

Pi-sigma network (PSN) is one of the most popular higher-order feedforward neural networks, which has generated substantial interest in a wide range of research communities, including function approximation [1], time series prediction [2], and pattern recognition [3]. Contrasting to traditional regularization term adding a penalty functional to the cost function, is that to improve the generalization performance and sometimes to control the size of the network weights to decrease the error between the desired and real outputs of the networks [4,5]. Due to the training process, practical there are two ways to implement the weights updating in the networks as online gradient [6] and batch gradient [7]. The modified cost error function with the regularization term is defined as follows

$$E(W) = \bar{E}(W) + \lambda \|W\|^p \tag{1}$$

where λ is the regularization parameter, $\bar{E}(W)$ is a usual error function depending on the W is the weights of the network, $\|W\|^p = \sum_{k=1}^{n} |w_k|^p$ is the p-norm ($0 \leq p \leq 2$) of the weights of the network. It is well known that the L_0 regularization ($p = 0$) is the earliest regularization method used to variable selection and feature extraction. The L_0 regularization yields the most sparse solutions, but it faces the problem of combinatory optimization is a NP-hard nature [8]. To overcome this difficulty, a typical relaxation of the L_0 regularization term is introduced, which is the L_1 regularization term [9]. L_1 regularization is also called Lasso [10] and has been accepted as one of the most useful tools for sparse optimization. The L_2

DOI: 10.14738/tmlai.66.5838
Publication Date: 27th December, 2018
URL: http://dx.doi.org/10.14738/tmlai.66.5838

regularization term is common term introduced into the training procedure for neural networks, which has computational efficient due to having analytical solution [11, 12].

More related works, in [13] study the online gradient method with smoothing L_0 regularization for FNNs training and shows how the absolute value approximated by a series of smoothing function. In [14] proposed online gradient method with smoothing $L_{1/2}$ regularization for training FNN. The objective function of this term is the sum of a non-convex, non-smooth, which causes oscillation of the error function and the norm and difficulty in convergence analysis. However, the $L_{1/2}$ regularization is approximated by smoothing function.

The organization of this paper is as follows. In Section 2, we describe PSN algorithm and the online gradient method (OG) with smoothing L_0 regularization. In Section 3, the convergence theorems and its analysis are presented. Contains some supporting simulation results in Section 4. The summarized of this work in Section 5.

2 Description of the Proposed Method

The description of the numbers neurons of PSN for the input, summation and product nodes are p, n and 1, respectively. Take $w_j = (w_{j1}, w_{j2}, \ldots, w_{jp})^T \in \mathbb{R}^P (1 \leq j \leq n)$ the weight vector connecting the input layer, the $k-th$ is summation unit and write $w = (w_1^T, w_2^T, \ldots, w_n^T) \in \mathbb{R}^{np}$. Note that the weights from summing units to product unit are fixed to be 1 and the topological structure of SPNN is given below.

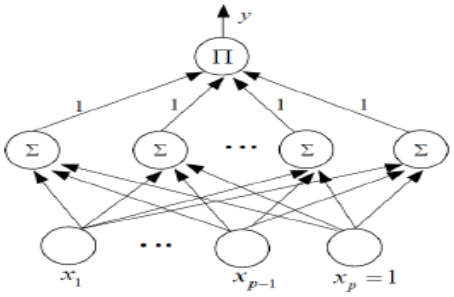

Figure 1. Pi-Sigma network algorithm

The network outputs calculated for input data $x = (x_1, \ldots, x_p) \in \mathbb{R}^p$ and given activation function $g: \mathbb{R} \to \mathbb{R}$ by

$$y = g\left(\prod_{j=1}^{n}(w_j \cdot x)\right) \tag{2}$$

Given a set of the training samples $\{x^l, O^l\}_{l=1}^{L} \subset \mathbb{R}^p \times \mathbb{R}$, where O^l is the desired ideal output for the input x^l. The modified error with L_0 regularization is defined by

$$\tilde{E}(w) = \sum_{l=1}^{L}\left(O^l - g\left(\prod_{j=1}^{n}(w_j \cdot x^l)\right)\right)^2 + \lambda\|w\|_0$$

$$= \sum_{l=1}^{L} g_l\left(\prod_{j=1}^{n}(w_j \cdot x^l)\right) + \lambda\|w\|_0 \tag{3}$$

where $g_l(t) = \frac{1}{2}(O^l - g(t))^2$, λ > 0 is regularization coefficient and $\|\cdot\|_0$ is the L_0-norm of a vector, defined as $\| w \|_0 = \sum_{j\in w}|w_j|^0$ with the absolute value term denoted by $|\cdot|$. The problem mathematical of this term is not differentiable and the online gradient method can be directly used, which it difficult to fine w^*, such as

$$\tilde{E}(w^*) = \min \tilde{E}(w) \tag{4}$$

To better understand the efficiency of the proposed methods we have reasonable way to fine practically needed to converge to a solution in (3). To this end, we use differentiable function to approximate the absolute value term; there are many smooth functions can be applied to various these kinds of problems. The modification of the error function with smoothing L_0 regularization can be given by

$$E(w) = \sum_{l=1}^{L}\left(O^l - g\left(\prod_{j=1}^{n}(w_j \cdot x^l)\right)\right)^2 + \lambda F(w)$$
$$= \sum_{l=1}^{L} g_l\left(\prod_{j=1}^{n}(w_j \cdot x^l)\right) + \lambda F(w) \tag{5}$$

where $F(w) = \sum_{j\in w} f(w_j)$ is a differentiable function and the smoothing function given by

$$f(x) = \begin{cases} -x & x \le -\varepsilon, \\ -\frac{1}{8\varepsilon^3}x^4 + \frac{1}{4\varepsilon}x^2 + \frac{3}{8}x & -\varepsilon < x < \varepsilon, \\ x & x \ge \varepsilon, \end{cases} \tag{6}$$

where $\varepsilon > 0$ is a small positive constant. The gradient of the error $E(w)$ with respect to w_j as

$$E_{w_j}(w) = \sum_{l=1}^{L} g_l'\left(\prod_{j=1}^{n}(w_j \cdot x^l)\right)\prod_{\substack{k=1\\k\neq j}}^{n}(w_k \cdot x^l)\, x^l + \lambda f'(w_j) \tag{7}$$

The network updates the weights $\{w^m\}$ iteratively starting from $m = 0$ by

$$w_j^{m+1} = w_j^m - \eta \Delta w_j^m \tag{8}$$

$$\Delta w_j^m = g_l'\left(\prod_{j=1}^{n}(w_j^m \cdot x^l)\right)\prod_{\substack{k=1\\k\neq j}}^{n}(w_k^m \cdot x^l)\, x^l + \lambda f'(w_j^m)$$

where $\eta > 0$ is the learning rate.

3 Convergence analysis

The following presumptions are required to prove the convergence Theorem 1.

Assumption (A1) $|g(t)|, |g'(t)|, |g''(t)|$ are uniformly bounded for $t \in \mathbb{R}$

Assumption (A2) $\left\|w_j^m \cdot x^l\right\|$ $(m = 0,1, \dots)$ is uniformly bounded.

Assumption (A3) Chosen η and λ to satisfy: $0 < \eta < 1/(M\lambda + C)$, where

$$C = \frac{1}{2}C_1 C_2^2 C_3^{2(n-1)} L + \frac{1}{2}C_1^{n-1} C_2^2 L(n-1),$$
$$C_1 = max\left\{\sup_{t\in\mathbb{R}}|g(t)|, \sup_{t\in\mathbb{R}}|g'(t)|, \sup_{t\in\mathbb{R}}|g''(t)|, \sup_{t\in\mathbb{R},1\le l\le L}|g_l'(t)|, \sup_{t\in\mathbb{R},1\le l\le L}|g_l''(t)|\right\},$$

$$C_2 = \min_{1\le l\le L} \|x^l\|, \qquad C_3 = \sup_{m\in\mathbb{N},1\le l\le L} \|w_j^m \cdot x^l\| \tag{9}$$

Assumption (A4) There exists a closed bounded region Φ such that $\{w^m\} \subset \Phi$, and set $\Phi_0 = \{w \in \Phi: E_w(w) = 0\}$ contains only finite points.

Theorem 1 Let the error function $E(\mathrm{w})$ is defined by (5) and the weight $\{w^m\}$ be generated by the iteration algorithm (8) for an arbitrary initial value $m = 0$. If Assumptions (A1) - (A3) are valid, then the following estimate exists:

(a) $E(w^{m+1}) \le E(w^m)$, $m = 0,1,\dots;$

(b) $\lim_{m\to\infty} \|E_w(w^m)\| = 0;$

Moreover, if Assumption (A4) is valid, the strong convergence will be established: There exists a point Φ_0 satisfying that

(c) $\lim_{m\to\infty} w^m = w^*.$

4 Experiment and Analysis

The purpose of this section is to carry out the numerical experiments of the proposed OG with the smoothing L_0 regularization term ($\mathrm{OGSL_0}$) to performance of lean practices, which compared with OG with L_1 regularization ($\mathrm{OGL_1}$) and L_2 regularization ($\mathrm{OGL_2}$). Each of the three algorithms takes 10 trials as well as to be able to provide more powerful mapping capability. In this test, we use identification of the nonlinear function $y = sin(\pi x)$ with different closed interval $x \in [a, -a]$. By choosing the learning rate $\eta = 0{,}05, \lambda = 0{,}0001$ the penalty parameter and max number of iterations 2000.

From Figures 2-5, we see that the proposed $\mathrm{OGSL_0}$ decreases monotonically and the corresponding gradient tends to zero it's better than the $\mathrm{OGL_1}$ and $\mathrm{OGL_2}$ when the number of iteration increases. From Table 1, we also can see that the average error training, average error testing and the average numbers of neurons eliminated (ANE in brief) by the pruning over the 10 trials and the resulting of $\mathrm{OGSL_0}$ is more suitable.

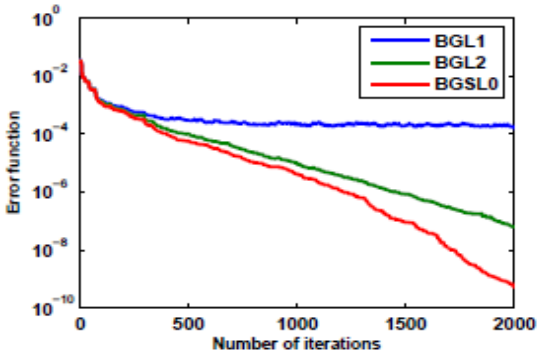

Figure 2. Results of errors for $\mathrm{OGL_1}$, $\mathrm{OGL_2}$ and $\mathrm{OGSL_0}$ in interval [0.2,-0.2].

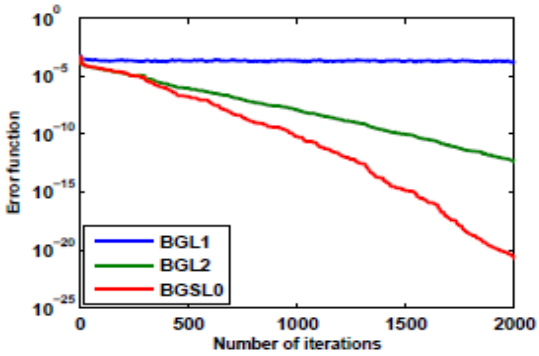

Figure 3. Results of errors for $\mathrm{OGL_1}$, $\mathrm{OGL_2}$ and $\mathrm{OGSL_0}$ in interval [0.5,-0.5].

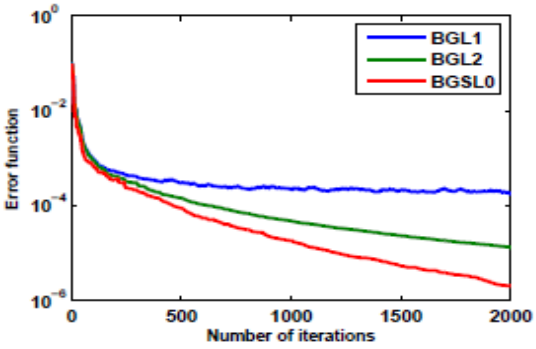

Figure 4. Results of errors for OGL_1, OGL_2 and $OGSL_0$ in interval [1,-1].

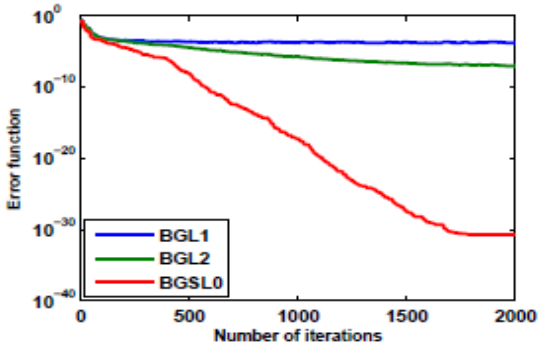

Figure 5. Results of errors for OGL_1, OGL_2 and $OGSL_0$ in interval [1.3,-1.3].

Table 1. Effect results of average error training/ testing with different interval.

Algorithm	Average Error of training patterns	Average Error of testing patterns	CAN of zero	Initial weights $W \in [a, -a]$
OGL_1	1.7090e-04	1.8748e-04	1.4	[0.2,-0.2]
OGL_2	6.3672e-08	4.8842e-08	2.0	[0.2,-0.2]
$OGSL_0$	5.4650e-10	3.1593e-10	5.4	[0.2,-0.2]
OGL_1	1.5960e-04	1.7445e-04	0	[0.5,-0.5]
OGL_2	4.7631e-13	2.9299e-13	0	[0.5,-0.5]
$OGSL_0$	2.2005e-21	1.3043e-21	3.0	[0.5,-0.5]
OGL_1	1.7684e-04	1.9067e-04	1.0	[1,-1]
OGL_2	1.3566e-05	1.1340e-05	1.8	[1,-1]
$OGSL_0$	2.0794e-06	1.2024e-06	4.0	[1,-1]
OGL_1	1.6778e-04	1.6639e-04	1.3	[1.3,-1.3]
OGL_2	9.7101e-08	7.8366e-08	2.4	[1.3,-1.3]
$OGSL_0$	2.0367e-31	5.7173e-29	5.1	[1.3,-1.3]

5 Conclusion

Regularization is the one of the most popular method used in neural networks applications. The L_0 regularization is the earliest regularization term applied to feature extraction and variable selection. This term yields the sparsest solutions, but it NP-hard problems in combinatory optimization. This study presents a novel online gradient method with smoothing L_0 regularization for pi-sigma neural network (OGSL_0). The key contribution of paper is to address the absolute value term of the L_0 regularization at the origin by using a smoothing function. The propose of OGSL_0 it's able to make a feasible analysis solution and shown a good effectiveness convergence results when compared with OGL_1 and OGL_2.

6 Acknowledgment

The authors would like to express our thanks to the anonymous reviewers and the editors for their previous helpful comments and valuable suggestions, which greatly improves this paper.

7 Appendix

The following lemma is a crucial tool for our analysis. Its proof is thus omitted (see [15]).

Lemma 1 Let $\mathrm{F}: \mathbb{R}^{\mathrm{Q}} \to \mathbb{R}$ is continuous and differentiable on a compact set $H \subset \mathbb{R}$ and that $\mathcal{M} = \{U \in H | \nabla F(U) = 0\}$ has only finite number of points. If a sequence $\{U^k\}_{k=1}^{\infty} \in H$ satisfies $\lim_{k\to\infty} \|U^{k+1} - U^k\| = 0$ and $\lim_{k\to\infty} \|\nabla \mathrm{F}(U^k)\| = 0$, then there exists a point $\mathrm{U}^* \in \mathcal{M}$ such that $\lim_{k\to\infty} U^k = U^*$.

The proof of Theorem 1 is divided into three steps. For convenience, we let

$$\rho_j^m = \sum_{j=1}^{n} \left(\Delta w_j^m\right)^2 \tag{10}$$

By using the Taylor's formula to extend $g_l\left(\prod_{j=1}^{n}\left(w_j^{m+1} \cdot x^l\right)\right)$ at $\prod_{j=1}^{n}\left(w_j^m \cdot x^l\right)$, we have

$$\begin{aligned}
&g_l\left(\prod_{j=1}^{n}\left(w_j^{m+1} \cdot x^l\right)\right) - g_l\left(\prod_{j=1}^{n}\left(w_j^m \cdot x^l\right)\right) \\
&= g_l'\left(\prod_{j=1}^{n}\left(w_j^m \cdot x^l\right)\right)\prod_{\substack{k=1 \\ k\neq j}}^{n}\left(w_k^m \cdot x^l\right)\sum_{j=1}^{n}\left(\Delta w_j^m \cdot x^l\right) \\
&+\frac{1}{2}\sum_{\substack{j_1,j_2=1 \\ j_1\neq j_2}}^{n}\left(\prod_{\substack{k=1 \\ k\neq j}}^{n}(t_1)\right)\left(\Delta w_{j_1}^m \cdot \Delta w_{j_1}^m\right)\left(x^l\right)^2 \\
&+\frac{1}{2}g''(t_2)\left(\prod_{j=1}^{n}\left(w_j^{m+1} \cdot x^l\right) - \prod_{j=1}^{n}\left(w_j^m \cdot x^l\right)\right)^2
\end{aligned} \tag{11}$$

where $t_1 \in \mathbb{R}$ is between $w_j^{m+1} \cdot x^l$ and $w_j^m \cdot x^l$ and $t_2 \in \mathbb{R}$ is between $\prod_{j=1}^{n}\left(w_j^{m+1} \cdot x^l\right)$ and $\prod_{j=1}^{n}\left(w_j^m \cdot x^l\right)$.

Proof to $(\boldsymbol{a})$ of Theorem 1. After dealing with (11) by accumulation (5) and Taylors formula

$$\begin{aligned}
E(w^{m+1}) - E(w^m) &= \sum_{l=1}^{L}\left(g_l\left(\prod_{j=1}^{n}\left(w_j^{m+1} \cdot x^l\right)\right) - g_l\left(\prod_{j=1}^{n}\left(w_j^m \cdot x^l\right)\right)\right) \\
&+\lambda(F(w^{m+1}) - F(w^m)) \\
&= \sum_{l=1}^{L}\left(g_l'\left(\prod_{j=1}^{n}\left(w_j^m \cdot x^l\right)\right)\prod_{\substack{k=1 \\ k\neq j}}^{n}\left(w_k^m \cdot x^l\right)x^l\right) \cdot \Delta w_j^m \\
&+\lambda\sum_{j=1}^{n}\left(f'\left(w_j^m\right) + f''(t_3)\Delta w_j^m\right) \cdot \Delta w_j^m \\
&+\frac{1}{2}g''(t_2)\left(\prod_{j=1}^{n}\left(w_j^{m+1} \cdot \mathrm{x}^l\right) - \prod_{j=1}^{n}\left(w_j^m \cdot x^l\right)\right)^2 + \delta \\
&\leq -\left(\frac{1}{\eta} - M\lambda\right)\rho_j^m \\
&+\frac{1}{2}g''(t_2)\left(\prod_{j=1}^{n}\left(w_j^{m+1} \cdot x^l\right) - \prod_{j=1}^{n}\left(w_j^m \cdot x^l\right)\right)^2 + \delta
\end{aligned} \tag{12}$$

where $t_3 \in \mathbb{R}$ is between $w_j^{m+1} \cdot x^l$ and $w_j^m \cdot x^l$, $f''(t_3) = M$ and

$$\delta = \frac{1}{2}\sum_{l=1}^{L} g_l'\left(\prod_{j=1}^{n}(w_j^m \cdot x^l)\right)\sum_{\substack{j_1,j_2=1 \\ j_1 \neq j_2}}^{n}\prod_{\substack{k=1 \\ k\neq j}}^{n}(t_1)\left(\Delta w_{j_1}^m \cdot \Delta w_{j_1}^m\right)(x^l)^2$$

By Cauchy-Schwartz inequality and Assumption (A2), we have the following estimation

$$\begin{aligned}\left(\prod_{j=1}^{n}(w_j^{m+1} \cdot x^l) - \prod_{j=1}^{n}(w_j^m \cdot x^l)\right) &\leq \left(\prod_{j=1}^{n-1}(w_j^{m+1} \cdot x^l)\right)(w_j^{m+1} - w_j^m)x^l \\ &+ \left(\prod_{j=1}^{n-2}(w_j^{m+1} \cdot x^l)(w_j^m \cdot x^l)\right)(w_j^{m+1} - w_j^m)x^l \\ &+ \cdots + \left(\prod_{j=2}^{n}(w_j^{m+1} \cdot x^l)\right)(w_j^{m+1} - w_j^m)x^l \\ &\leq C_3^{n-1}\|x^l\|\sum_{j=1}^{n}(w_j^{m+1} - w_j^m) \\ &\leq C_2 C_3^{n-1}\sum_{j=1}^{n}(\Delta w_j^m) \qquad (13)\end{aligned}$$

and

$$\begin{aligned}\delta &\leq \frac{1}{2}\sum_{l=1}^{L} g_l'\left(\prod_{j=1}^{n}(w_j^m \cdot x^l)\right)\sum_{\substack{j_1,j_2=1 \\ j_1 \neq j_2}}^{n}\left(\prod_{\substack{k=1 \\ k\neq j}}^{n}(t_1)\right)\left(\Delta w_{j_1}^m \cdot \Delta w_{j_1}^m\right)(x^l)^2 \\ &\leq \frac{1}{2}C_1^{n-1}C_2^2 L(n-1)\sum_{\substack{j_1,j_2=1 \\ j_1 \neq j_2}}^{n}\left(\Delta w_{j_1}^m \cdot \Delta w_{j_1}^m\right) \\ &\leq \frac{1}{2}C_1^{n-1}C_2^2 L(n-1)\rho_j^m \qquad (14)\end{aligned}$$

Substation (13) and (14) into (12), we fine that

$$\begin{aligned}E(w^{m+1}) - E(w^m) &\leq -\left(\frac{1}{\eta} - M\lambda - \frac{1}{2}C_1C_2^2C_3^{2(n-1)}L - \frac{1}{2}C_1^{n-1}C_2^2L(n-1)\right)\rho_j^m \\ &\leq -\left(\frac{1}{\eta} - M\lambda - C\right)\rho_j^m \\ &\leq 0, \qquad (15)\end{aligned}$$

where $C = \frac{1}{2}C_1C_2^2C_3^{2(n-1)}L + \frac{1}{2}C_1^{n-1}C_2^2L(n-1)$, thus leads to

$$E(w^{m+1}) \leq E(w^m), m = 0,1,2,\ldots \qquad (16)$$

The proof to (a) of Theorem 1 is thus completed.

Proof to (b) of Theorem 1. According to the Assumption (A3), (15) and fine $\beta > 0$ satisfied:

$$\beta = \frac{1}{\eta} - M\lambda - C. \qquad (17)$$

In view of (16), (17), there holds

$$E(w^{m+1}) \le E(w^m) - \beta\rho_j^m \le \cdots \le E(w^0) - \beta\sum_{q=0}^{m}\rho_j^m.$$

From $E(w^{m+1}) > 0$, then

$$\beta\sum_{q=0}^{m}\rho_j^m \le E(w^0) < \infty.$$

Called $m \to \infty$, we obtain

$$\sum_{q=0}^{\infty}\rho_j^m \le \frac{1}{\beta}E(w^0) < \infty,$$

and thus leads to

$$\lim_{m\to\infty}\left\|\Delta w_j^m\right\| = \lim_{m\to\infty}\left\|E_{w_j}(w^m)\right\| = 0. \tag{18}$$

Thus proof to (b) of the Theorem 1 is omitted.

Proof to (c) of Theorem 1. Note that the error function $E(w)$ defined in (5) is continuous and differentiable. According to (18), Assumption (A4) and Lemma 1, we can easily get the desired result, i.e., there exists a point $w^* \in \Phi$ such that

$$\lim_{m\to\infty} w^m = w^*$$

This proof is completed.

REFERENCES

[1]. *Shin, Y., et al., The pi-sigma network: An efficient higher-order neural network for pattern classification and function approximation. In Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on IEEE, 1991. 1: p. 13-18.*

[2]. *Akram, U., et al., A comprehensive Survey on Pi-Sigma Neural Network for Time Series Prediction. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 2017. 9(3-3): p.57-62.*

[3]. *Shin, Y., et al., Computationally efficient invariant pattern recognition with higher order Pi-Sigma Networks. The University of Texas at Austin, 1992.*

[4]. *Setiono, R., A penalty-function approach for pruning feedforward neural networks. Neural computation, 1997. 9(1): p.185-204.*

[5]. *Reed, R., Pruning algorithms-a survey, IEEE Trans Neural Network, 1993. 4 (5): p.740–7.*

[6]. *Wu, W., et al., Deterministic convergence of an online gradient method for BP neural networks. IEEE Transactions on Neural Networks, 2005. 16(3): p.533-540.*

[7]. *Hagan, M.T., et al., Training feedforward networks with the Marquardt algorithm. IEEE transactions on Neural Networks, 1994. 5(6): p.989-993.*