BATCH GRADIENT METHOD FOR TRAINING OF PI-SIGMA NEURAL NETWORK WITH PENALTY

Kh. Sh. Mohamed[1,2], Yan Liu[3], Wei Wu[1], Habtamu Z. A[1]

[1]School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China
[2]Mathematical Department, College of Sciences, Dalanj University, Dalanj, Sudan
[3] School of Information Science and Engineering, Dalian Polytechnic University, Dalian 116034, China

ABSTRACT

In this letter, we describe a convergence of batch gradient method with a penalty condition term for a narration feed forward neural network called pi-sigma neural network, which employ product cells as the output units to inexplicit amalgamate the capabilities of higher-order neural networks while using a minimal number of weights and processing units. As a rule, the penalty term is condition proportional to the norm of the weights. The monotonicity of the error function with the penalty condition term in the training iteration is firstly proved and the weight sequence is uniformly bounded. The algorithm is applied to carry out for 4-dimensional parity problem and Gabor function problem to support our theoretical findings.

KEYWORDS

Pi-sigma Neural Network, Batch Gradient Method, Penalty& Convergence

1. INTRODUCTION

Introduced another higher order feed forward polynomial neural network called the pi-sigma neural network (PSN)[2], which is known to provide naturally more strongly mapping abilities than traditional feed forward neural network. The neural networks consisting of the PSN modules has been used effectively in pattern classification and approximation problems [1,7,10,13]. There are two ways of training to updating weight: The first track, batch training, the weights are updating after each training pattern is presented to the network in [9]. Second track, online training, the weights updating immediately after each training sample is fed (see [3]). The penalty condition term is oftentimes inserted into the network training algorithms has been vastly used so as to amelioration the generalization performance, which refers to the capacity of a neural network to give correct outputs for untrained data and to control the magnitude of the weights of the network structure [5,6,12]. In the second track the online training weights updating become very large and over-fitting resort to occur, by joining the penalty term into the error function [4,8,11,14], which acts as a brute-force to drive dispensable weights to zero and to prevent the weights from taking too large in the training process. The objective of this letter to prove the strong and weak convergence main results which are based on network algorithm prove that the weight sequence generated is uniformly bounded.

For related work we mention [15] where a sigma-pi-sigma network is considered. The pi-sigma network considered in this paper has different structure as sigma-pi-sigma network and leads to

different theoretical analysis. But some techniques of proofs in [15] are also used here in this paper.

The rest of this paper is organized as follows. The neural network structure and the batch gradient method with penalty is described in the Section 2. In Section 3 the main convergence results are presented. Simulation results are provided in Section 4. In Section 5 the proofs of the main results are provided. Finally, some conclusions are drawn in Section 6.

2. BATCH GRADIENT METHOD WITH A PENALTY TERM

In this paper, we are concerned with a PSNs with the structure *p-n-1*, where p, n and 1 are the dimensions of the input, hidden and output layers, respectively. Let $w_k = \left(w_{k1}, w_{k2} \dots, w_{kp}\right)^T \in \mathbb{R}^P (1 \le k \le n)$ the weight vectors connecting the input and summing units, and write $w = (w_1^T, w_2^T, \dots, w_N^T) \in \mathbb{R}^{np}$. Corresponding to the biases w_{kP}, with fixed value-1. The structure of PSN is shown in Fig.1.

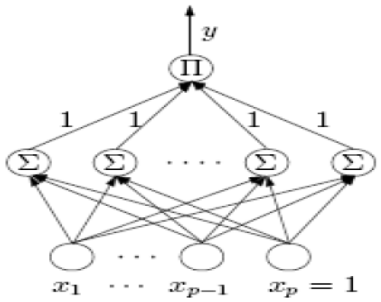

Figure 1. A pi-sigma network with p-n-1 structure

Assume g: $\mathbb{R} \to \mathbb{R}$ is a given activation function. In specially, for an input $x^j \in \mathbb{R}^P$, the output of the pi-sigma network is

$$y = \mathrm{g}\left(\prod_{i=1}^{n}\left(w_i \cdot x^j\right)\right) \tag{1}$$

The network keeping with a given set of training examples $\left\{\xi^j, y^j\right\}_{j=1}^{J} \subset \mathbb{R}^p \times \mathbb{R}^J$, J is the numbers of training examples. The error function with a penalty is given by:

$$\begin{aligned} E(w) &= \frac{1}{2}\sum_{j=1}^{J}\left(\left(o^j - \mathrm{g}\left(\prod_{i=1}^{n}\left(w_i \cdot x^j\right)\right)\right)^2 + \lambda\sum_{k=1}^{n}\|w_k\|^2\right. \\ &= \sum_{j=1}^{J}\mathrm{g}_j\left(\prod_{i=1}^{n}\left(w_i \cdot x^j\right)\right) + \lambda\sum_{k=1}^{n}\|w_k\|^2 \end{aligned} \tag{2}$$

where $\lambda > 0$ is a penalty coefficient and $\mathrm{g_j(t)} = \frac{1}{2}\left(o^j - \mathrm{g}(t)\right)^2$. The gradient of $E(w)$ with respect to w_k is written as :

$$\nabla E_k(w) = \sum_{j=1}^{J} \mathrm{g}_j \left(\prod_{i=1}^{n} (w_i \cdot x^j) \right) \prod_{\substack{i=1 \\ i \neq k}}^{n} (w_i \cdot x^j) x^j + 2\lambda w_k \tag{3}$$

Then, the weights updating rule is

$$w_k^{m+1} = w_k^m - \Delta w_k^m, \qquad m = 0,1,\ldots. \tag{4}$$

$$\begin{aligned} \Delta w_k^m &= -\eta \nabla E_k(w_k^m) \\ &= \sum_{j=1}^{J} \mathrm{g}_j' \left(\prod_{i=1}^{n} (w_i^m \cdot x^j) \right) \prod_{\substack{i=1 \\ i \neq k}}^{n} (w_i^m \cdot x^j) x^j + 2\lambda w_k^m \end{aligned} \tag{5}$$

m denotes mth update and $\eta > 0$ is the learning rate. In this paper, we suppose that $\eta$is a fixed constant and $\|\cdot\|$ denotes the Euclidean norm.

3. MAIN RESULTS

In this section we present some convergence theorems of the batch gradient method with penalty (4). These proofs are given in next section. Some sufficient conditions for the convergence are as follows:

(A1) $|\mathrm{g}(t)|, |\mathrm{g}_j'(t)|, |\mathrm{g}_j''(t)| \leq C \ \ \forall t \in \mathbb{R}, 1 \leq j \leq J.$
(A2) $\|x^j\| \leq \mathrm{C}, |w_k^i \cdot x^j| \leq C, \forall 1 \leq j \leq J, 1 \leq k \leq n, i = 0,1,\ldots$
(A3) η and λ are chosen to satisfy the condition: $0 < \eta < \frac{1}{\lambda + C}$
(A4) There exists a closed bounded region Ω such $\{w^m\} \subset \Omega$, and the set $\Omega_0 = \{w | E_w(w) = 0\}$ contains only finite points.

Theorem 1 If Assumptions (A1) – (A3) are valid, let the error function is given by (2), and the weight sequence $\{w^m\}$ be generated by the iteration algorithm (4) for an arbitrary initial value, then we have

$(i) E(w^{m+1}) \leq E(w^m), m = 0,1,2,\ldots.$
(ii) There exists $E^* \geq 0$ such that $\lim_{m\to\infty} E(w^m) = E^*$.
$(iii) \lim_{m\to\infty} \|E_k(w^m)\| = 0, \ , k = 1,2,..,n.$
Furthermore, if Assumption (A4) is also valid, then we have the following strong convergence
(iv) There exists a point $w^* \in \Omega_0$ such that $\lim_{m\to\infty} w^m = w^*$
The monotonicity and limit of the error function sequence $\{E(w^m)\}$ are shown in Statements (i) and (ii), respectively. Statements (ii) (ii) and (iii) indicate the convergence of $\{E_k(w^m)\}$, referred to as weak convergence. The strong convergence of $\{w^m\}$ is described in Statement (iv).

4. SIMULATIONS RESULTS

To expound the convergence of batch gradient method for training pi-sigma neural network, numerical example experiments are executed for 4-parity problem and regression problem.

4.1. Example 1: Parity Problem

Parity problem is a difficult classification problem. The famous XOR problem is completely the two-parity problem. In this example, we use the four-parity problem to test the performance of PSNs. The network is of three layers with the structure 5-4-1, and the logistic activation function $g(t) = 1/(1 + e^{-t})$ is used for the hidden and output nodes. The initial weights are chosen in $[-0.5, 0.5]$ and the learning rate with different value $\eta = 0.05$, 0.07 and 0.09 and the penalty parameter $\lambda = 0.0001$. The maximum number of epoch 3000.

From Figures 2(a), (b) and 3(c) we observe that the error function and gradient of norm decrease monotonically, respectively, and that both norm of the gradient error function approaches zero, as depicted by the convergence Theorem 1. and from Figures Figure 3(d), (e) and (f), we can see the that the valid function approximation.

4.2. Example 2: Function Regression Problem

In this section we test the performance of batch gradient with penalty for a multi-dimensional Gabor function has the following form (see Figure. 5):

$$a(x,y) = \frac{1}{2\pi(0.5)^2} exp\left(\frac{x^2+y^2}{2(0.5)^2}\right) cos\big(2\pi(x+y)\big).$$

In this example, 256 input points are selected from an evenly 16×16 grid on $-0.5 \le x \le 0.5$ and $-0.5 \le y \le 0.5$ and the 16 input points are randomly selected from the 256 points as training patterns. The number of neurons for input, summation and product layer are p=3, N=6 and 1, respectively. The parameters in this example take the values $\eta = 0.9,$ and $\lambda = 0.0001$. when the number of training iteration epochs reaches 30000.

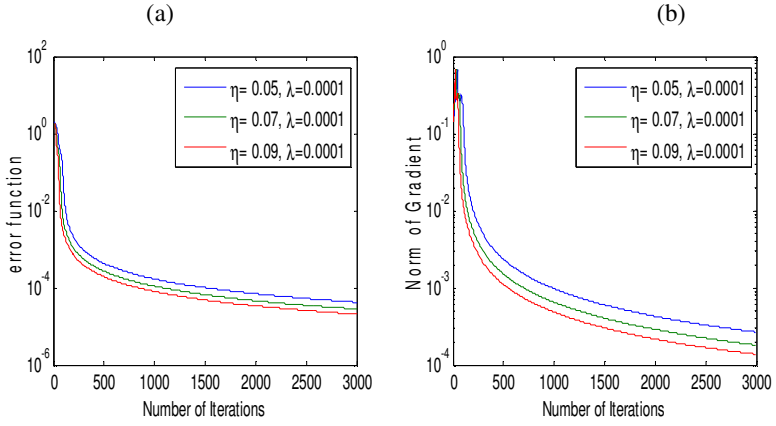

Figure 2. Example 1: (a) Error function with penalty (b) Norm of gradient with penalty

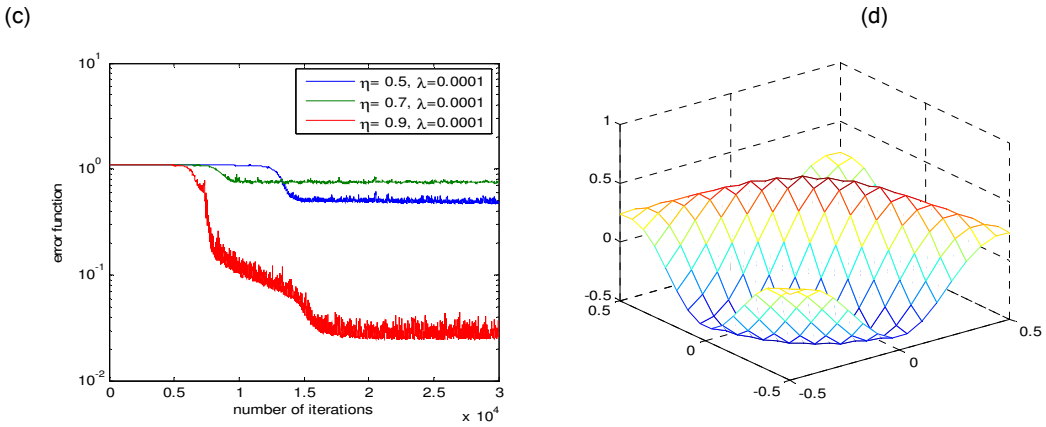

Figure 3. Example 2: **(c)** Error function with penalty (d) Gabor function

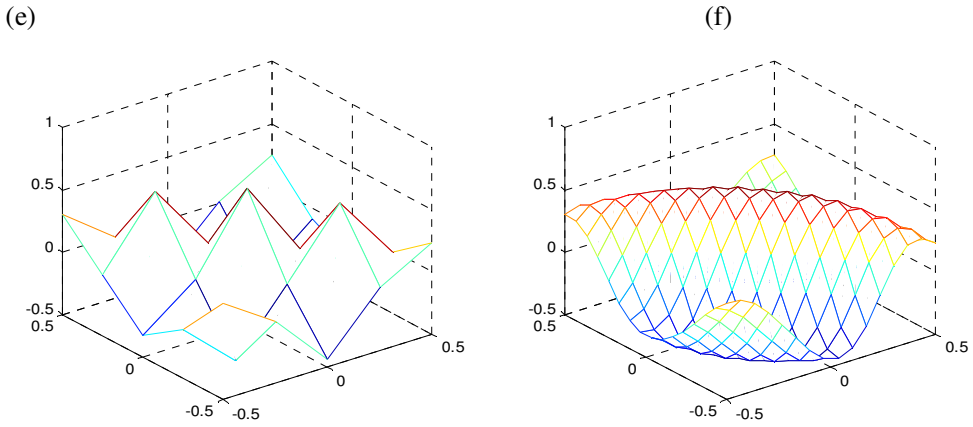

Figure 3. The approximation: (e) training pattern results (f) test pattern results

5. PROOFS

To proof Theorem 1, First we present a important lemma which contribute to the our analysis, which is basically the same as Theorem 14.1.5 in [16]. Their proof is thus omitted.

Lemma 1 Suppose that h: $\mathbb{R}^K \to \mathbb{R}$ is continuous and differentiable on a compact set $\breve{D} \subset \mathbb{R}^K$ and that $\Omega = \{\mathcal{Z} \in \breve{D} | \nabla h(\mathcal{Z}) = 0\}$ has only finite number of point. If a sequence $\{\mathcal{Z}^m\}_{m=1}^{\infty} \in \breve{D}$ satisfies then $\lim_{m\to\infty} \|\mathcal{Z}^{m+1} - \mathcal{Z}^m\| = 0$, $\lim_{m\to\infty} \|\nabla h(\mathcal{Z}^m)\| = 0$. Then there exists a point $\mathcal{Z}^* \in \Omega$ such that $\lim_{m\to\infty} \mathcal{Z}^m = \mathcal{Z}^*$.

Proof of Theorem 1 For sake of convenience, we show the notations

$$\sigma^m = \sum_{k=1}^{n} \|\Delta w_k^m\|^2 \tag{6}$$

$$r_k^{m,j} = \Delta_j^m w_k^{m+1} - \Delta_j^m w_k^m \tag{7}$$

Proof Applying Taylor's formula to extend $\mathrm{g}_j\left(\prod_{i=1}^{n}(w_i^{m+1}\cdot x^j)\right)$ at $\prod_{i=1}^{n}(w_i^{m+1}\cdot x^j)$, we have

$$\begin{aligned}\mathrm{g}_j\left(\prod_{i=1}^{n}(w_i^{m+1}\cdot x^j)\right) &= \mathrm{g}_j\left(\prod_{i=1}^{n}(w_i^{m}\cdot x^j)\right)\\ &+\mathrm{g}_j^{'}\left(\prod_{i=1}^{n}(w_i^{m}\cdot x^j)\right)\sum_{k=1}^{n}\left(\prod_{\substack{i=1\\ i\neq k}}^{n}(w_i^{m}\cdot x^j)\right)\left(w_k^{\mathrm{m+1}}-w_k^m\right)x^j\\ &+\frac{1}{2}\mathrm{g}^{''}(t_1)\left(\prod_{i=1}^{n}(w_i^{m+1}\cdot x^j)-\prod_{i=1}^{n}(w_i^{m}\cdot x^j)\right)^2\\ &+\frac{1}{2}\sum_{\substack{k_1,k_2=1\\ k_1\neq k_2}}^{n}\left(\prod_{\substack{i=1\\ i\neq k_1,k_2}}^{n} t_2\right)\left(w_{k_1}^{\mathrm{m+1}}-w_{k_1}^m\right)\left(w_{k_2}^{\mathrm{m+1}}-w_{k_2}^m\right)\left(x^j\right)^2\end{aligned} \tag{8}$$

Where $t_1 \in \mathbb{R}$ is on the line segment between $\prod_{i=1}^{n}(w_i^{\mathrm{m+1}}\cdot x^j)$ and $\prod_{i=1}^{n}(w_i^{m}\cdot x^j)$. After dealing with (8) by accumulation $\mathrm{g}_j\left(\prod_{i=1}^{n}(w_i^{m+1}\cdot x^j)\right)$ for $1 \le j \le J$, we obtain from (2), (4) and Taylor's formula, we have

$$\begin{aligned}E(w^{m+1}) &= \sum_{j=1}^{J}\mathrm{g}_j\left(\prod_{i=1}^{n}\left(w_i^{m+1}\cdot x^j\right)\right)+\lambda\sum_{k=1}^{n}\left\|w_k^{m+1}\right\|^2\\ &= E(w^m)-\frac{1}{\eta}\sum_{k=1}^{n}\|\Delta w_k^m\|^2+\frac{\lambda}{2}\mathrm{C}^2\sum_{k=1}^{n}\|\Delta w_k^m\|^2+\Delta_1+\Delta_2\end{aligned} \tag{9}$$

Where

$$\Delta_1 = \frac{1}{2}\sum_{j=1}^{J}\mathrm{g}_{\mathrm{j}}^{'}\left(\prod_{i=1}^{n}\left(w_i^m\cdot\xi^j\right)\right)\sum_{\substack{k_1,k_2=1\\ k_1\neq k_2}}^{n}\left(\prod_{\substack{i=1\\ i\neq k_1,k_2}}^{n} t_2\right)\left(\Delta w_{k_1}^{\mathrm{m}}\right)\left(\Delta w_{k_2}^{\mathrm{m}}\right)\left(x^j\right)^2 \tag{10}$$

$$\Delta_2 = \frac{1}{2}\sum_{j=1}^{J}\mathrm{g}^{''}(t_1)\left(\prod_{i=1}^{n}(w_i^{m+1}\cdot x^j)-\prod_{i=1}^{n}(w_i^{m}\cdot x^j)\right)^2 \tag{11}$$

$$\Delta_3 = \frac{1}{\eta}\sum_{k=1}^{n}(\Delta w_k^m)\cdot\sum_{k=1}^{n}\left(r_k^{m,j}\right) \tag{12}$$

It follows from (A1), (A2), (5) and Taylor's formula to first and second orders, we obtain

$$|\Delta_1| \le \frac{1}{2}C^{n+1}\sum_{j=1}^{J}\sum_{\substack{k_1,k_2=1\\ k_1\neq k_2}}^{n}\left\|\Delta w_{k_1}^m\right\|\cdot\left\|\Delta w_{k_2}^m\right\|$$

$$\leq \frac{1}{2} C^{n+1}(n-1)J \sum_{k=1}^{n} \|\Delta w_k^m\|^2$$
$$\leq \mathrm{C}_3 \sum_{k=1}^{n} \|\Delta w_k^m\|^2 \tag{13}$$

Where $\mathrm{C}_3 = \frac{1}{2} C^{n+1}(n-1)J$. By Assumption (A2) , (A2) and Cauchy- Schwartz inequality, we have

$$\left| \prod_{i=1}^{n} (w_i^{m+1} \cdot x^j) - \prod_{i=1}^{n} (w_i^m \cdot x^j) \right|$$
$$\leq \left| \prod_{i=1}^{n} (w_i^{m+1} \cdot x^j) \right| \left| (w_n^{m+1} - w_n^m) x^j \right| + \left| \prod_{i=1}^{n-2} \left(w_i^{m+1} \cdot x^j\right) \left(w_i^m \cdot x^j\right) \right|$$
$$\times \left| (w_{n-1}^{m+1} - w_{n-1}^m) x^j \right| + \cdots + \left| \prod_{\mathrm{i}=1}^{n} \left(w_i^m \cdot x^j\right) \right| \left| (w_1^{m+1} - w_1^m) x^j \right|$$
$$\leq C^{n-1} \left\| x^j \right\| \sum_{k=1}^{n} \|\Delta w_k^m\|$$
$$\leq C_4 \sum_{k=1}^{n} \|\Delta w_k^m\| \tag{14}$$

Where $C_4 = C^n (1 \leq j \leq J, 1 \leq k \leq n\,, m = 0,1,2, \dots)$. Similarly, we get

$$\left| \prod_{\substack{i=1 \\ i \neq k}}^{n} (w_i^m \cdot x^j) - \prod_{\substack{i=1 \\ i \neq k}}^{n} (w_i^m \cdot x^j) \right| \leq \tilde{C}_4 \sum_{k=1}^{n} \|\Delta w_k^m\| \tag{15}$$

When $\tilde{C}_4 = C^{n-1}$. By Assumptions (A1), (14) and Cauchy Schwartz inequality, we obtain

$$|\Delta_2| \leq \frac{1}{2} C \left| \left(\prod_{i=1}^{n} (w_i^{m+1} \cdot x^j) - \prod_{i=1}^{n} (w_i^m \cdot x^j) \right)^2 \right|$$
$$\leq \frac{1}{2} \mathrm{C} \mathrm{C}_4^2 \sum_{k=1}^{n} \|\Delta w_k^m\|^2 \tag{16}$$

By Assumption (A1), (A2), (7), (14) and (15) for $m = 0,1, \cdots$, we have

$$|\Delta_3| \leq \left| \sum_{k=1}^{n} (\Delta w_k^m) \cdot \sum_{k=1}^{n} \left(r_k^{m,j}\right) \right|$$
$$\leq (C^{n+1} C_4 + C^{n+1} + 2\lambda) \sum_{k=1}^{n} \|\Delta w_k^m\|^2$$
$$\leq \mathrm{C}_5 \sum_{k=1}^{n} \|\Delta w_k^m\|^2 \tag{17}$$

Where $\mathrm{C}_5 = (C^{n+1}C_4 + C^{n+1} + 2\lambda)$. Transfer (13), (16) and (17) into (9), there holds

$$\begin{aligned} E(w^{m+1}) - E(w^m) &\leq -\left(\frac{1}{\eta} - \frac{\lambda}{2}\mathrm{C}^2 - \mathrm{C}_3 - \mathrm{C}_5 - \frac{1}{2}\mathrm{C}\mathrm{C}_4^2\right)\sum_{k=1}^{n}\|\Delta w_k^m\|^2 \\ &\leq -\left(\frac{1}{\eta} - \mathrm{C}\right)\sum_{k=1}^{n}\|\Delta w_k^m\|^2 \\ &\leq 0 \end{aligned} \tag{18}$$

This completes the proof to report $(i)$of the Theorem 1.

Proof to (ii)of the Theorem 1 From the conclusion (i), we know that the non-negative sequence $\{E(w^m)\}$ is monotone. But it also bounded below. Hence, there must exist $E^* \geq 0$ such that $\lim_{k\to\infty} E(w^m) = E^*$. The proof to $(ii)$is thus completed.

Proof to (iii) of the Theorem 1 It the follows from Assumption (A3) that $\beta > 0$. Taking $\beta = \frac{1}{\eta} - \mathrm{C}$ and using (18), we get

$$E(w^{m+1}) \leq E(w^m) - \beta\rho^m \leq \cdots \leq E(w^0) - \beta\sum_{k=1}^{m}\rho^k$$

Since $E(w^{m+1}) > 0$, then we have

$$\beta\sum_{k=1}^{m}\rho^k \leq E(w^0) < \infty.$$

Setting $m \to \infty$, we have

$$\sum_{k=1}^{\infty}\rho^k \leq \frac{1}{\beta}E(w^m) < \infty.$$

Thus

$$\lim_{m\to\infty}\rho^m = 0$$

It follows from (5) and Assumption (A1)

$$\lim_{m\to\infty}\|\Delta w_k^m\| = 0, \quad k = 1,2,\dots,n \tag{19}$$

This completes the proof.

Proof to (iv) of the Theorem 1 Note that the error function $E(w^m)$ defined in (2) is continuous and differentiable. According to (16), Assumption (A4) and Lemma 1, we can easily get the desired result, i.e ., there exists a point $w^* \in \Omega_0$ such that

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

This completes the proof to (iv).

6. CONCLUSION

Convergence results are decided for the batch gradient method with penalty for training pi-sigma neural network (PSN). The penalty term is a condition proportional to the magnitude of the weights. We prove under moderate conditions, if Assumptions (A1) - (A3) hold, then

that the weights of the networks are deterministically bounded in the learning process. With the help of this conclusion, to point strongly purpose, if Assumption (A4) is also valid, then we prove that the suggested algorithm converges with probability one to the set of zeroes of the gradient of the error function in (2). Resemblance, the existing similar convergence results require the boundedness of the weights as a precondition.

ACKNOWLEDGEMENTS

We gratefully acknowledge to thank the anonymous referees for their valuable comments and suggestions on the revision of this paper.

REFERENCES

[1] Hussaina A. J & Liatsisb P (2002)'' Recurrent pi-sigma networks for DPCM image coding'',Neurocomputing, Vol. 55, pp 363-382.
[2] Shin Y & Ghosh J (1991) ''The pi-sigma network: An efficient higher-order neural network forpattern classification and function approximation'', International Joint Conference on Neural Networks, Vol. 1, pp13–18.
[3] Wu. W & Xu Y. S(2002) '' Deterministic convergence of an online gradient method for neural networks''Journal of Computational and Applied MathematicsVo1. 144 (1-2),pp335-347.
[4] Geman S, Bienenstock E & Doursat R (1992)''Neural networks and the bias/variance dilemma''Neural Computation Vo1. 4, pp1–58.
[5] ReedR (1997)''Pruning algorithms-a survey''IEEE Transactions on Neural NetworksVo1. 8, pp185–204.
[6] G. Hinton G (1989) ''Connectionist learning procedures''Artificial Intelligence Vo1.40, pp185-243.
[7] Sinha M, Kumar K &Kalra P.K (2000) '' Some new neural network architectures with improved learning schemes'', Soft Computing, Vo1.4, pp214-223.
[8] Setiono R (1997) ''A penalty-function approach for pruning feedforward neural networks''Neural Networks Vo1.9, pp185–204.
[9] Wu. W, Feng G. R &Li X. Z (2002)''Training multiple perceptrons via minimization of sum of ridgefunctions' 'Advances in Computational Mathematics, Vo1.17 pp331-347.
[10] Shin Y& Ghosh J(1992)''Approximation of multivariate functions using ridge polynomial networks'' International Joint Conference on Neural Networks, Vo1. 2, pp380-385.
[11] Bartlett P. L'' For valid generalization, the size of the weights is more important than the size of the network'' Advances in Neural Information Processing Systems Vo1. 9, pp134–140.
[12] Loone S & Irwin G(2001)''Improving neural network training solutions using regularisation, Neurocomputing Vo1. 37, pp71-90.
[13] Jiang L. J Xu F & Piao S. R (2005) ''Application of pi-sigma neural network to real-time classification of seafloor sediments'' Applied Acoustics, Vo1. 24, pp346–350.
[14] Zhang H.S & Wu W (2009) ''Boundedness and convergence of online gradient method with penalty for linear output feed forward neural networks'' Neural Process Letters Vo1. 29, pp205–212.
[15] Liu Y, Li Z. X. Yang D.K, Mohamed Kh. Sh, Wang J & Wu W (2015) '' Convergence of batch gradient learning algorithm with smoothing L1/2 regularization for Sigma-Pi-Sigma neural networks'', Neurocmputing Vol. 151, pp333-341.
[16] Yuan Y & Sun W (2001) ''Optimization Theory and Methods'', Science Press, Beijing.

Authors

Kh. Sh. Mohamed received the M.S. degree from Jilin University, Changchun, China, in applied mathematics in 2011. He works as a lecturer of mathematics at College of Science Dalanj University, since 2011. Now he is working toward Ph.D. degree in computational mathematics at Dalian University of Technology, Dalian, China. His research interests include theoretical analysis and regularization methods for neural networks

Yan Liu received the B.S. degree in computational mathematics from the Dalian University of Technology in 2004. She is currently with the School of Information Sciences and Engineering, Dalian Polytechnic University, Dalian,China. In2012 she has obtained the Ph.D. degree in computational mathematics from Dalian University of Technology. Her research interests include machine learning, fuzzy neural networks and regularization theory.

Wei Wu received the Bachelor's and Master's degrees from Jilin University, Changchun,China,in1974 and1981, respectively, and the Ph.D. degree from Oxford University, Oxford,U.K.,in1987.He is currently with the School of Mathematical Sciences, Dalian University of Technology, Dalian, China. He has published four books and 90 research papers. His current research interests include learning methods of neural networks.

Habtamu Z.A received the bachelor degree from Wollega University, Ethiopia, in mathematics education in 2009 and his Master's degree in Mathematics education from Addis Ababa University, Ethiopia in 2011. He worked as a lecturer of applied mathematics at Assosa University, Ethiopia. Currently he is working toward Ph.D. degree in Applied mathematics at Dalian University of Technology, Dalian, China. His research interests include numerical optimization methods and neural networks