Neural Networks 24 (2011) 91–98

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Convergence analysis of online gradient method for BP neural networks☆

Wei Wu[a,*], Jian Wang[a,b], Mingsong Cheng[a], Zhengxue Li[a]

[a] School of Mathematical Sciences, Dalian University of Technology, Dalian, 116024, PR China

[b] School of Mathematics and Computational Sciences, Petroleum University of China, Dongying, 257061, PR China

ARTICLE INFO

Article history:
Received 28 March 2010
Received in revised form 8 September 2010
Accepted 9 September 2010

Keywords:
Neural networks
Backpropagation learning
Online gradient method
Weak convergence
Strong convergence

ABSTRACT

This paper considers a class of online gradient learning methods for backpropagation (BP) neural networks with a single hidden layer. We assume that in each training cycle, each sample in the training set is supplied in a stochastic order to the network exactly once. It is interesting that these stochastic learning methods can be shown to be deterministically convergent. This paper presents some weak and strong convergence results for the learning methods, indicating that the gradient of the error function goes to zero and the weight sequence goes to a fixed point, respectively. The conditions on the activation function and the learning rate to guarantee the convergence are relaxed compared with the existing results. Our convergence results are valid for not only S–S type neural networks (both the output and hidden neurons are Sigmoid functions), but also for P–P, P–S and S–P type neural networks, where S and P represent Sigmoid and polynomial functions, respectively.

1. Introduction

Artificial neural network has been a hot topic in recent years in cognitive science, computational intelligence and intelligent information processing. Backpropagation (BP) is the most broadly used learning method for feedforward neural networks. It was first proposed by Werbos (1974) in his Ph.D. thesis, and has been rediscovered several times (LeCun, 1985; Parker, 1982; Rumelhart, Hinton, & Williams, 1986). There are two practical ways to implement the backpropagation algorithm: batch updating approach and online updating approach. Corresponding to the standard gradient method, the batch updating approach accumulates the weight correction over all the training samples before actually performing the update. On the other hand, the online updating approach updates the network weights immediately after each training sample is fed. Some authors compare the two different training schemes for feedforward neural networks (Heskes & Wiegerinck, 1996; Nakama, 2009; Wilson & Martinez, 2003). Heskes and Wiegerinck (1996) reveal several asymptotic properties of the two schemes. Wilson and Martinez (2003) explain why batch training is almost always slower than online training (often orders of magnitude slower) especially on large training sets. Nakama (2009) theoretically analyzes the convergence properties of the two schemes applied to quadratic loss functions and shows the exact degrees to which the training set size, the variance of the per-instance gradient, and the learning rate affect the rate of convergence for each scheme.

There are three approaches for online training of BP neural networks according to different fashions of sampling. The first approach is OGM-CS (completely stochastic order): At each learning step, one of the samples is drawn at random from the training set and presented to the network (Finnoff, 1994; Heskes & Wiegerinck, 1996; Terence, 1989; Wilson & Martinez, 2003). The second approach is OGM-SS (special stochastic order): In each training cycle, each sample in the training set is supplied in a stochastic order to the network exactly once (Heskes & Wiegerinck, 1996; Li & Ding, 2005; Li, Wu, & Tian, 2004; Nakama, 2009). The third approach is OGM-F (fixed order): In each training cycle, each sample in the training set is supplied in a fixed order to the network exactly once (Heskes & Wiegerinck, 1996; Mangasarian & Solodov, 1994; Wu & Xu, 2002; Wu, Feng, Li, & Xu, 2005; Xu, Zhang, & Jin, 2009).

Naturally, the existing convergence results for OGM-CS are mostly asymptotic convergence with a probabilistic nature as the size of training samples goes to infinity (Bertsekas & Tsitsiklis, 1996; Chakraborty & Pal, 2003; Fine & Mukherjee, 1999; Finnoff, 1994; Liang, Feng, Lee, Lim, & Lee, 2002; Tadic & Stankovic, 2000; Terence, 1989; Zhang, Wu, Liu, & Yao, 2009). Deterministic convergence can be obtained for OGM-SS and OGM-F (Li et al., 2004; Mangasarian & Solodov, 1994; Shao, Wu, & Liu, 2007; Wu & Xu, 2002; Wu et al., 2005; Wu, Feng, & Li, 2002; Wu & Shao, 2003; Wu, Shao, & Qu, 2005; Xu et al., 2009). It is interesting to see that the learning method OGM-SS with stochastic nature enjoys deterministic convergence. The convergence result is a

☆ Project supported by the National Natural Science Foundation of China (No. 10871220).

* Corresponding author.
E-mail address: wuweiw@dlut.edu.cn (W. Wu).

0893-6080/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2010.09.007

bit easier to prove for OGM-F than for OGM-SS. But we have reason to believe, and our experience shows, that OGM-SS behaves numerically better than OGM-F since the stochastic nature of the learning procedure survives in OGM-SS (Li & Ding, 2005; Li et al., 2004).

To guarantee the convergence, it is commonly required that the learning rate η_m satisfies the assumptions $\sum_{m=1}^{\infty}\eta_m = \infty$ and $\sum_{m=1}^{\infty}\eta_m^2 < \infty$ as in Bertsekas and Tsitsiklis (1996) and Tadic and Stankovic (2000) for OGM-CS. An extra assumption $\lim_{m\to\infty}\eta_m/\eta_{m+1} = 1$ was introduced by Xu et al. (2009) for OGM-F. A special condition which is basically $\eta_m = O(1/m)$ was required in Li et al. (2004), Shao et al. (2007), Wu and Xu (2002), Wu et al. (2005), Wu et al. (2002), Wu and Shao (2003) and Wu et al. (2005) for OGM-F and OGM-SS.

To obtain the strong convergence result, which means that the weight sequence converges to a fixed point, Wu et al. (2005) introduced an additional assumption: the number of the stationary points of the error function is finite. A more relaxed condition is used in Xu et al. (2009): the gradient of the error function has at most countably infinite number of stationary points.

The aim of this paper is to present a comprehensive study on the weak and strong convergence for OGM-F and OGM-SS, indicating that the gradient of the error function goes to zero and the weight sequence goes to a fixed point, respectively. These convergence results improve the existing results in Li et al. (2004), Shao et al. (2007), Wu and Xu (2002), Wu et al. (2005), Wu et al. (2002), Wu and Shao (2003), Wu et al. (2005) and Xu et al. (2009) such that the conditions on the activation function and the learning rate to guarantee the convergence are much relaxed. Specifically, we make the following contributions:

- The extra condition $\lim_{m\to\infty}\eta_m/\eta_{m+1} = 1$ for the learning rate is removed which is a requisite in Xu et al. (2009).
- The convergence results are valid for both OGM-F and OGM-SS.
- The convergence results apply not only to S–S type neural networks (both the output and hidden neurons are Sigmoid functions), but also to P–P, P–S and S–P type neural networks, where S and P represent Sigmoid and polynomial functions, respectively.
- The restrictive assumptions for the strong convergence in Wu et al. (2005) and Xu et al. (2009) are relaxed such that the stationary points set of the error function is only required not to contain any interior point.
- We assume that the derivative g' of the activation function is Lipschitz continuous on any bounded closed interval. This improves the corresponding conditions in Wu et al. (2005), which require the boundedness of the second derivative g'', and in Xu et al. (2009), which require g' to be Lipschitz continuous and uniformly bounded on the whole R.

Let us make a few remarks on the above contribution points. For the first contribution point, as an example, we recall a well-known adaptive technique for the learning rate η_m: $\eta_m = (1+a)\eta_{m-1}$ if the error is decreasing, and $\eta_m = (1-a)\eta_{m-1}$ otherwise, where $a < 1$ is a positive number. Xu's condition $\lim_{m\to\infty}\eta_m/\eta_{m+1} = 1$ (Xu et al., 2009) is not valid in this case, while our convergence results remains valid. For the second contribution point, it is interesting to see that the learning method OGM-SS with stochastic nature enjoys deterministic convergence. We observe that OGM-F is actually a deterministic iteration procedure in that the iteration sequence is determined uniquely by the initial value and the fixed order of the samples. The convergence result is a bit easier to prove for OGM-F than for OGM-SS. We have reason to believe, and our experience shows, that OGM-SS behaves numerically better than OGM-F since the stochastic nature of the learning procedure survives in OGM-SS (Li & Ding, 2005; Li et al., 2004). Our convergence results are generalizations of both the results of Xu et al. (2009), which considers OGM-F, and the results of Li et al. (2004), which considers OGM-SS with an unpleasant condition $\eta_m = O(1/m)$ on the learning rate. Our third contribution allows the activation functions for both hidden and output layers to be more flexible. Here we remark that typically, S–S type networks are used for classification problems, and S–P type networks with Sigmoid hidden neurons and linear output neurons are used for approximation problems. The existing convergence results (Li et al., 2004; Shao et al., 2007; Wu & Xu, 2002; Wu et al., 2005, 2002; Wu & Shao, 2003; Wu et al., 2005; Xu et al., 2009) are mostly for either S–S type or S–P type alone but not for both of them. In this paper, we give a uniform treatment for all types of networks. The fourth and fifth contribution points are mainly of theoretical interest. From a theoretical point of view, we mention that different analytical tools are employed in Wu et al. (2005) and Xu et al. (2009) and this study for the convergence analysis, might explain, at least in part, why different conditions are obtained for the convergence. The differential Taylor expansion is used in Wu et al. (2005), which requires the boundedness of the second derivative g'' of the activation function g; the mean value theorem of integrals is employed in Xu et al. (2009), which requires g' to be Lipschitz continuous and uniformly bounded; and in this paper, we use the integral Taylor expansion and hence require the Lipschitz continuity of g' on any bounded closed interval. Finally, we point out that Xu et al. (2009) is a big step forward for the convergence study of OGM-F and that Xu et al. (2009) also includes another convergence result under the condition that the error function is directionally convex. This convex condition is not considered in this paper.

The rest of this paper is organized as follows. In Section 2, online updating methods including OGM-F and OGM-SS are introduced. The main convergence results are presented in Section 3 and their proofs are gathered in Section 4. Some conclusions are drawn in Section 5.

2. OGM-F and OGM-SS

Let us begin with an introduction of a feedforward neural network with three layers. The numbers of neurons for the input, hidden and output layers are p, n and 1, respectively. Suppose that the training sample set is $\left\{\mathbf{x}^j, O^j\right\}_{j=1}^{J} \subset \mathbb{R}^p \times \mathbb{R}$, where $\mathbf{x}^j$ and O^j are the input and the corresponding ideal output of the jth sample, respectively. Let $\mathbf{V} = \left(v_{i,j}\right)_{n\times p}$ be the weight matrix connecting the input and the hidden layers, and write $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{ip})^T$ for $i = 1, 2, \ldots, n$. The weight vector connecting the hidden and the output layers is denoted by $\mathbf{u} = (u_1, u_2, \ldots, u_n)^T \in \mathbb{R}^n$. To simplify the presentation, we combine the weight matrix $\mathbf{V}$ with the weight vector $\mathbf{u}$, and write $\mathbf{w} = \left(\mathbf{u}^T, \mathbf{v}_1^T, \ldots, \mathbf{v}_n^T\right)^T \in \mathbb{R}^{n(p+1)}$. Let $g,\ f : \mathbb{R} \to \mathbb{R}$ be given activation functions for the hidden and output layers, respectively. For convenience, we introduce the following vector valued function

$$G(\mathbf{z}) = (g(z_1), g(z_2), \ldots, g(z_n))^T, \quad \forall\, \mathbf{z} \in \mathbb{R}^n. \tag{1}$$

For any given input $\mathbf{x} \in \mathbb{R}^p$, the output of the hidden neurons is $G(\mathbf{Vx})$, and the final actual output is

$$y = f(\mathbf{u} \cdot G(\mathbf{Vx})). \tag{2}$$

For any fixed weights $\mathbf{w}$, the error of the neural networks is defined as

$$E(\mathbf{w}) = \frac{1}{2}\sum_{j=1}^{J}(O^j - f(\mathbf{u}\cdot G(\mathbf{Vx}^j)))^2 = \sum_{j=1}^{J} f_j(\mathbf{u}\cdot G(\mathbf{Vx}^j)), \tag{3}$$

where $f_j(t) = \frac{1}{2}(O^j - f(t))^2,\ j = 1, 2, \ldots, J,\ t \in \mathbb{R}$. The gradients of the error function with respect to $\mathbf{u}$ and $\mathbf{v}_i$ are, respectively,