Journal of Computational Mathematics, Vol.26, No.4, 2008, 613–623.

CONVERGENCE OF GRADIENT METHOD WITH MOMENTUM FOR BACK-PROPAGATION NEURAL NETWORKS*

Wei Wu
Department of Applied Mathematics, Dalian University of Technology, Dalian 116024, China
Email: wuweiw@dlut.edu.cn
Naimin Zhang
Mathematics and Information Science College, Wenzhou University, Wenzhou 325035, China
Email: naiminzhang@yahoo.com.cn
Zhengxue Li and Long Li
Department of Applied Mathematics, Dalian University of Technology, Dalian 116024, China
Email: lizx@dlut.edu.cn, long_li1982@163.com
Yan Liu
College of Information Science and Engineering, Dalian Institute of Light Industry, Dalian 116034, China
Email: liuyan_3001@hotmail.com

Abstract

In this work, a gradient method with momentum for BP neural networks is considered. The momentum coefficient is chosen in an adaptive manner to accelerate and stabilize the learning procedure of the network weights. Corresponding convergence results are proved.

Mathematics subject classification: 68Q32, 68T05.
Key words: Back-propagation (BP) neural networks, Gradient method, Momentum, Convergence.

1. Introduction

Back-propagation (BP) algorithm is widely used in neural network training, and its convergence is discussed in, e.g., [4, 5]. A momentum term is often added to the BP algorithm in order to accelerate and stabilize the learning procedure [2, 10, 11], in which the present weight updating increment is a combination of the present gradient of the error function and the previous weight updating increment.

Phansalkar and Sastry [8] give a stability analysis for the BP algorithm with momentum (BPM in short). They show that the stable points of BPM are local minima of the least squares error, and other equilibrium points are unstable. Qian [9] also discusses BPM, showing that the behavior of the system near a local minimum is equivalent to a set of coupled and damped harmonic oscillators. The momentum term improves the speed of convergence by bringing some eigen components of the system closer to critical damping. These two results are local convergence results describing the behavior of the learning iteration *near* the local minima of the error function. They can not be directly used for the usual situation when the initial weights are chosen stochastically.

The convergence of BPM is also considered by Bhaya [2] and Torii [12]. They require the gradient of the error function $E_w(w)$ to be a linear function of the weight w. Especially in [12]

* Received March 19, 2007 / Revised version received August 15, 2007 / Accepted September 19, 2007 /

the learning rate and the momentum coefficient are restricted to be constants. Consequently, the iteration procedure of BPM can be expressed as a stationary iteration. The convergence property is then determined by the eigenvalues of its iterative matrix. Unfortunately, for usual activation functions such as Sigmoid functions, the gradient of the error function is not a linear function of the weight. We mention that Bhaya [2] reveals an interesting fact that BPM is equivalent to the conjugate gradient method in a certain sense.

In [15], some convergence results are given for BPM in a simple case where the network has no hidden layer. These results are of global nature in the sense that they are valid for any arbitrarily given initial values of the weights. Moreover, it is not required that the gradient of the error function is linear. The key for the convergence analysis is the monotonicity of the error function during the learning iteration, which is proved under the uniformly boundedness assumption of the activation function and its derivatives.

The aim of this paper is to generalize the results in [15] to a more general and more important case, that is, the BP neural network with a hidden layer. Due to the involvement of the hidden layer, we shall need an extra assumption that the weight vectors connecting the hidden and the output layers of the BP neural network are uniformly bounded. Then, we are able to establish the convergence of BPM.

The rest part of the paper is organized as follows. In Section 2 we introduce BPM and discuss its convergence property. In Section 3 we make some numerical experiments to verify our theoretical result. The details of the convergence proof are provided in Section 4.

2. BPM and Its Convergence

Consider a BP neural network with three layers. The numbers of neurons for the input, hidden and output layers are l, n and 1, respectively. Let the input training examples be $\xi^j \in R^l$ $(j = 1, \cdots, J)$, and the corresponding desired outputs be $O^j \in R$ $(j = 1, \cdots, J)$. We denote the weight matrix connecting the input and the hidden layers by $V = (v_{ij})_{n\times l}$, and we write $v_i = (v_{i1}, v_{i2}, \cdots, v_{il}) \in R^l$ $(i = 1, \cdots, n)$. The weight vector connecting the hidden and the output layers is denoted by $w = (w_1, w_2, \cdots, w_n) \in R^n$. Let $g : R \longrightarrow R$ be a given activation function for the hidden and output layers. For convenience, we introduce the following vector function for $x = (x_1, \cdots, x_n) \in R^n$

$$G(x) = (g(x_1), g(x_2), \cdots, g(x_n)). \tag{2.1}$$

For any given input $\xi \in R^l$, the output of the hidden neurons is $G(V\xi)$, and the final output of the network is

$$\zeta = g(w \cdot G(V\xi)). \tag{2.2}$$

We remark that, in practice, there should be bias involved in the above formulas for the output and hidden neurons. Here we have dropped the bias so as to simplify the presentation and derivation.

The usual square error function is defined by

$$\begin{aligned} E(w, V) &:= \frac{1}{2}\sum_{j=1}^{J}\left[O^j - g\left(w \cdot G(V\xi^j)\right)\right]^2 \\ &\equiv \sum_{j=1}^{J} g_j(w \cdot G(V\xi^j)), \end{aligned} \tag{2.3}$$