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

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

where $g_j(t) \equiv \frac{1}{2}(O^j - g(t))^2,\ j = 1, \cdots, J$. The aim of the network training is to find (w^*, V^*) such that

$$E(w^*, V^*) = \min E(w, V). \tag{2.4}$$

The gradients of the error function with respect to w and V respectively are as follows

$$E_w(w, V) = \sum_{j=1}^{J} g_j'(w \cdot G(V\xi^j))G(V\xi^j), \tag{2.5}$$

$$E_{v_i}(w, V) = \sum_{j=1}^{J} g_j'(w \cdot G(V\xi^j))w_i g'(v_i \cdot \xi^j)\xi^j, \quad i = 1, \cdots, n. \tag{2.6}$$

Given arbitrarily the initial weights w^0, w^1 and V^0, V^1, BPM method updates the weights w and V iteratively by

$$\begin{cases} w^{k+1} = w^k - \eta \sum_{j=1}^{J} g_j'(w^k \cdot G(V^k\xi^j))G(V^k\xi^j) + \tau_k(w^k - w^{k-1}), \\ v_i^{k+1} = v_i^k - \eta \sum_{j=1}^{J} g_j'(w^k \cdot G(V^k\xi^j))w_i^k g'(v_i^k \cdot \xi^j)\xi^j \\ \qquad + \gamma_{k,i}(v_i^k - v_i^{k-1}), \quad i = 1, \cdots, n, \quad k = 1, 2, \cdots, \end{cases} \tag{2.7}$$

where $\eta \in (0, 1)$ is the learning rate, and τ_k and $\gamma_{k,i}$ are the momentum coefficients to be determined. Denote

$$\Delta w^{k+1} = w^{k+1} - w^k, \tag{2.8}$$

$$\Delta v_i^{k+1} = v_i^{k+1} - v_i^k, \quad i = 1, \cdots, n, \tag{2.9}$$

$$p^k = E_w(w^k, V^k) \equiv \sum_{j=1}^{J} g_j'(w^k \cdot G(V^k\xi^j))G(V^k\xi^j), \tag{2.10}$$

$$q_i^k = E_{v_i}(w^k, V^k) \equiv \sum_{j=1}^{J} g_j'(w^k \cdot G(V^k\xi^j))w_i^k g'(v_i^k \cdot \xi^j)\xi^j, \quad i = 1, \cdots, n. \tag{2.11}$$

Then (2.7) can be rewritten as follows:

$$\begin{cases} \Delta w^{k+1} = \tau_k \Delta w^k - \eta p^k, \\ \Delta v_i^{k+1} = \gamma_{k,i}\Delta v_i^k - \eta q_i^k, \quad i = 1, \cdots, n, \quad k = 1, 2, \cdots. \end{cases} \tag{2.12}$$

Similar to [15], we choose the momentum coefficients τ_k and $\gamma_{k,i}$ as follows:

$$\tau_k = \begin{cases} \frac{\tau\|p^k\|}{\|\Delta w^k\|} & \text{if } \|\Delta w^k\| \neq 0, \\ 0 & \text{else,} \end{cases} \quad \gamma_{k,i} = \begin{cases} \frac{\tau\|q_i^k\|}{\|\Delta v_i^k\|} & \text{if } \|\Delta v_i^k\| \neq 0, \\ 0 & \text{else,} \end{cases} \tag{2.13}$$

where $\tau \in (0, 1)$ is the momentum factor and $\|\cdot\|$ is the Euclidian norm.

The following assumptions will be used:

($A1$) $|g(t)|, |g'(t)|$ and $|g''(t)|$ are uniformly bounded for $t \in R$.

($A2$) $\|w^k\|$ $(k = 1, 2, \cdots)$ are uniformly bounded.

($A3$) The following set has finite number of elements:

$$\Omega = \{(w, V)|\ \ E_w(w, V) = 0,\ E_{v_i}(w, V) = 0,\ i = 1, \cdots, n\}. \tag{2.14}$$

Remark 2.1. Condition $(A1)$ is valid for Sigmoid functions which are the most commonly used activation functions. Condition $(A2)$ will be needed to guarantee the weak convergence (2.16)-(2.18) below, that is, boundedness implies (weak) convergence in the learning iteration process. Further investigation is necessary if one wants to drop this condition. We point out that a condition similar to $(A2)$ is assumed for nonlinear problems in [6]. Condition $(A3)$ requires that the error function has only finite number of local minimums, which is used to guarantee the strong convergence (2.19)-(2.20). From $(A1)$ and $(A2)$, it is easy to verify that $\|G(x)\|$ is bounded for $x \in R^n$, that $\|p^k\|$ and $\|q_i^k\|$ $(i = 1, \cdots, n;\ k = 1, 2, \cdots)$ are uniformly bounded, and that $|g_j(t)|, |g_j'(t)|$ and $|g_j''(t)|$ $(j = 1, \cdots, J)$ are also uniformly bounded for $t \in R$. These observations will be frequently used later in our proofs.

The following theorem is our main results, whose proof is postponed to Section 4.

Theorem 2.1. *Assume that* $(A1)$ *and* $(A2)$ *are valid. Then, there exist constants* $C^* > 0$ *and* $E^* \geq 0$ *such that for* $0 < s < 1,\ \tau = s\eta$ *and*

$$\eta < \min\Big\{1,\ \frac{1-s}{C^*[(1+s)(3+2s+s^2)+(2+s+s^2)^2]}\Big\}, \tag{2.15}$$

the following results hold for the iteration process (2.12)*:*

$$\lim_{k\to\infty} E(w^k, V^k) = E^*, \tag{2.16}$$

$$\lim_{k\to\infty} \|E_w(w^k, V^k)\| = 0, \tag{2.17}$$

$$\lim_{k\to\infty} \|E_{v_i}(w^k, V^k)\| = 0, \quad i = 1, \cdots, n. \tag{2.18}$$

Furthermore, if $(A3)$ *is also satisfied, then the iteration process* (2.12) *converges to a local minimum* (w^*, V^*)*:*

$$\lim_{k\to\infty} w^k = w^*, \quad \lim_{k\to\infty} V^k = V^*, \tag{2.19}$$

$$E_w(w^*, V^*) = 0, \quad E_{v_i}(w^*, V^*) = 0, \quad i = 1, \cdots, n. \tag{2.20}$$

3. Numerical Experiment

Let us illustrate the convergence behavior of our BPM with a hidden layer by using two numerical examples simulating two benchmark problems, i.e., the 8-bit parity problem and the Sonar problem. In all cases, the logistic activation function $g(x) = 1/(1 + \exp^{-x})$ is used for the hidden and output nodes of the following network structures. The learning rate η is 0.1 and the momentum coefficient τ is 0.05. The maximum number of epochs was set to 5000, and the training was considered successful whenever Fahlman's "40-20-40" criterion was satisfied [3] (i.e., values in the lowest 40% of the output range were treated as logical zero, values in the highest 40% of the output range were treated as logical one, and values in the middle 20% of the output range were treated as indeterminate and therefore considered as incorrect).

The 8-bit parity function is a mapping defined on the set of $2^8 = 256$ distinct 8-dimensional binary vectors, and its value is 1 or 0 when the sum of the 8 components of a binary vector is odd or even, respectively. The network is of three layers with the structure 8-8-1. We shall compare our results with those in [1], where two very efficient second order learning methods, namely, Levenberg-Marquardt with adaptive momentum (LMAM) and optimized Levenberg-Marquardt

with adaptive momentum (OLMAM), are proposed. It is easy to show that $(w, V) = 0$ is a local minimum point of the error function $E(w, V)$ defined by (2.3). So the initial values of (w^0, V^0) are chosen stochastically in $[-2, 2]$ rather than in a much smaller region $[-0.1, 0.1]$ as in [1]. In all the 100 trials, the learning iteration tended to a local minimum as predicted by our convergence prediction. 5 trials were completely successful, i.e., all the 256 binary vectors were correctly classified according to the "40-20-40" criterion. So the performance of our first order method BPM in this respect is not as good as the second order methods LMAM and OLMAM, for which the successful trials were 14 and 94 respectively as reported in [1]. But in average we correctly classified 98.68% of the binary vectors, which seems not too bad. On the other hand, we point out that the computational time of each epoch is $\mathcal{O}(m^3)$ for LMAM and OLMAM methods (m is the total number of the weights and bias of the network) since it needs to solve a linear system of order m, while it is only $\mathcal{O}(m)$ for our BPM since it only involves computations such as inner products of m-dimensional vectors.

The Sonar benchmark is a well-known classification problem. The task is to classify reflected sonar signals in two categories (metal cylinders and rocks). The related data set comprises 208 input vectors, each with 60 components. We use a 60-7-1 feedforward network (60 inputs, 7 hidden nodes, one output unit). The training set consists of 104 input vectors and 10 trails are performed as in [7]. The initial values of (w^0, V^0) are chosen stochastically in $[-1, 1]$. All the 10 trials were completely successful, i.e., all the 104 binary vectors were correctly classified according to the "40-20-40" criterion. We can not compare this result with that in [1], since the network with hidden layer is not used in [1] for this case. Instead, we compare it with the result in [7] which correctly classified 99.8% of the input vectors when 12 hidden nodes are applied in a standard feedforward neural network and the best performance of 100% was attained by the network with 24 hidden units. So this comparison indicates that our momentum method can improve the performance of the training for feedforward neural networks.

4. Proof of Theorem 2.1

4.1. Useful lemmas

We shall use the following abbreviations:

$$G^{k,j} = G(V^k \xi^j), \quad \omega^{k,j} = w^k \cdot G^{k,j}. \tag{4.1}$$

Notice

$$\omega^{k+1,j} - \omega^{k,j} = G^{k,j} \cdot \Delta w^{k+1} + (G^{k+1,j} - G^{k,j}) \cdot w^{k+1}.$$

Then, it follows from (2.12) that

$$\begin{aligned}
&\sum_{j=1}^{J} g_j'(\omega^{k,j})(\omega^{k+1,j} - \omega^{k,j}) \\
&= \sum_{j=1}^{J} g_j'(\omega^{k,j}) G^{k,j}(\tau_k \Delta w^k - \eta p^k) + \sum_{j=1}^{J} g_j'(\omega^{k,j})(G^{k+1,j} - G^{k,j}) \cdot w^{k+1} \\
&= -\eta \|p^k\|^2 + \tau_k p^k \cdot \Delta w^k + \sum_{j=1}^{J} g_j'(\omega^{k,j})(G^{k+1,j} - G^{k,j}) \cdot w^{k+1}.
\end{aligned}$$

By Taylor's formula, expanding $g_j(\omega^{k+1,j})$ at $\omega^{k,j}$ gives

$$\begin{aligned} g_j(\omega^{k+1,j}) &= g_j(\omega^{k,j}) + g_j' \\ &= (\omega^{k,j})(\omega^{k+1,j} - \omega^{k,j}) + \frac{1}{2} g_j''(t_{k,j})(\omega^{k+1,j} - \omega^{k,j})^2, \end{aligned}$$

where $t_{k,j}$ lies between $\omega^{k+1,j}$ and $\omega^{k,j}$. Consequently,

$$\begin{aligned} & E(w^{k+1}, V^{k+1}) \\ &= E(w^k, V^k) + \sum_{j=1}^{J} g_j'(\omega^{k,j})(\omega^{k+1,j} - \omega^{k,j}) + \frac{1}{2} \sum_{j=1}^{J} g_j''(t_{k,j})(\omega^{k+1,j} - \omega^{k,j})^2 \\ &= E(w^k, V^k) - \eta \|p^k\|^2 + \tau_k p^k \cdot \Delta w^k + \sum_{j=1}^{J} g_j'(\omega^{k,j})(G^{k+1,j} - G^{k,j}) \cdot w^{k+1} \\ &\quad + \frac{1}{2} \sum_{j=1}^{J} g_j''(t_{k,j})(\omega^{k+1,j} - \omega^{k,j})^2. \end{aligned} \tag{4.2}$$

Using Taylor's formula again, we have that there exists $\hat{t}_{k,i,j}$ between $v_i^{k+1} \cdot \xi^j$ and $v_i^k \cdot \xi^j$ such that

$$\begin{aligned} & g(v_i^{k+1} \cdot \xi^j) - g(v_i^k \cdot \xi^j) \\ &= g'(v_i^k \cdot \xi^j) \Delta v_i^{k+1} \cdot \xi^j + \frac{1}{2} g''(\hat{t}_{k,i,j})(\Delta v_i^{k+1} \cdot \xi^j)^2 \\ &= g'(v_i^k \cdot \xi^j)(\gamma_{k,i} \Delta v_i^k - \eta q_i^k) \cdot \xi^j + \frac{1}{2} g''(\hat{t}_{k,i,j})(\Delta v_i^{k+1} \cdot \xi^j)^2 \\ &= -\eta g'(v_i^k \cdot \xi^j) q_i^k \cdot \xi^j + g'(v_i^k \cdot \xi^j) \gamma_{k,i} \Delta v_i^k \cdot \xi^j + \frac{1}{2} g''(\hat{t}_{k,i,j})(\Delta v_i^{k+1} \cdot \xi^j)^2. \end{aligned} \tag{4.3}$$

Then it is easy to prove the following Lemma 4.1.

Lemma 4.1. *For the second last term in* (4.2)*, there holds*

$$\begin{aligned} & \sum_{j=1}^{J} g_j'(\omega^{k,j})(G^{k+1,j} - G^{k,j}) \cdot w^{k+1} \\ &= -\eta \sum_{i=1}^{n} \|q_i^k\|^2 + \sum_{j=1}^{J} \sum_{i=1}^{n} g_j'(\omega^{k,j}) w_i^k g''(v_i^k \cdot \xi^j) \gamma_{k,i} \Delta v_i^k \cdot \xi^j \\ &\quad + \frac{1}{2} \sum_{j=1}^{J} \sum_{i=1}^{n} g_j'(\omega^{k,j}) w_i^k g''(\hat{t}_{k,i,j})(\Delta v_i^{k+1} \cdot \xi^j)^2 \\ &\quad + \sum_{j=1}^{J} g_j'(\omega^{k,j})(G^{k+1,j} - G^{k,j}) \cdot \Delta w^{k+1}, \end{aligned} \tag{4.4}$$

where $\hat{t}_{k,i,j}$ lies between $v_i^k \cdot \xi^j$ and $v_i^{k+1} \cdot \xi^j$.

Lemma 4.2. *If* (A1) *and* (A2) *are valid, then there exists a constant $C_0 > 0$ such that*

$$\|G^{k+1,j} - G^{k,j}\| \le C_0(\eta + \tau + \eta^2 + \tau^2) \left(\sum_{i=1}^{n} \|q_i^k\|^2 \right)^{\frac{1}{2}}. \tag{4.5}$$

Proof. Denote

$$\varphi_{k,i,j} = g(v_i^{k+1} \cdot \xi^j) - g(v_i^k \cdot \xi^j).$$

Then we have

$$\|G^{k+1,j} - G^{k,j}\| = \left(\sum_{i=1}^{n} \varphi_{k,i,j}^2\right)^{\frac{1}{2}}.$$

It follows from (4.3) that

$$\varphi_{k,i,j} = -\eta g'(v_i^k \cdot \xi^j) q_i^k \cdot \xi^j + g'(v_i^k \cdot \xi^j)\gamma_{k,i}\Delta v_i^k \cdot \xi^j + \frac{1}{2} g''(\hat{t}_{k,i,j})(\Delta v_i^{k+1} \cdot \xi^j)^2.$$

It is easy to see that there exists $C_1 > 0$ such that

$$\begin{aligned} |\varphi_{k,i,j}| &\le C_1\Big(\eta\|q_i^k\| + |\gamma_{k,i}|\|\Delta v_i^k\| + \|\Delta v_i^{k+1}\|^2\Big) \\ &\le C_1(\eta+\tau)\|q_i^k\| + C_1\|\Delta v_i^{k+1}\|^2. \end{aligned} \tag{4.6}$$

By (2.12) and (2.13) we have

$$\|\Delta v_i^{k+1}\|^2 \le (\|\gamma_{k,i}\Delta v_i^k\| + \|\eta q_i^k\|)^2 \le 2(\tau^2+\eta^2)\|q_i^k\|^2.$$

By $(A1)$ and $(A2)$, $\|q_i^k\|$ is uniformly bounded, i.e., there exists $C_2 > 0$ such that $\|q_i^k\| \le C_2$ for any i and k. Consequently,

$$\|\Delta v_i^{k+1}\|^2 \le 2C_2(\tau^2+\eta^2)\|q_i^k\|. \tag{4.7}$$

Using (4.6)-(4.7) and setting $C_0 = \max\{C_1, 2C_1C_2\}$ yield

$$|\varphi_{k,i,j}| \le C_0(\eta+\tau+\eta^2+\tau^2)\|q_i^k\|,$$

which leads to the desired result (4.5). □

Lemma 4.3. *Let* $(A1)$ *and* $(A2)$ *be valid. Then, there exists a constant* $C^* > 0$ *such that*

$$\left|\sum_{j=1}^{J}\sum_{i=1}^{n} g_j'(\omega^{k,j}) w_i^k g'(v_i^k \cdot \xi^j)\gamma_{k,i}\Delta v_i^k \cdot \xi^j\right| \le \tau\sum_{i=1}^{n}\|q_i^k\|^2, \tag{4.8}$$

$$\left|\frac{1}{2}\sum_{j=1}^{J}\sum_{i=1}^{n} g_j'(\omega^{k,j}) w_i^k g''(\hat{t}_{k,i,j})(\Delta v_i^{k+1} \cdot \xi^j)^2\right| \le C^*(\tau+\eta)^2\sum_{i=1}^{n}\|q_i^k\|^2, \tag{4.9}$$

$$\begin{aligned} &\left|\sum_{j=1}^{J} g_j'(\omega^{k,j})(G^{k+1,j} - G^{k,j}) \cdot \Delta w^{k+1}\right| \\ &\quad\le C^*(\tau+\eta)(\eta+\eta^2+\tau+\tau^2) \cdot \Big(\|p^k\|^2 + \sum_{i=1}^{n}\|q_i^k\|^2\Big), \end{aligned} \tag{4.10}$$

$$\begin{aligned} &\left|\frac{1}{2}\sum_{j=1}^{J} g_j''(t_{k,j})(\omega^{k+1,j} - \omega^{k,j})^2\right| \\ &\quad\le C^*(\tau+\eta)^2\|p^k\|^2 + C^*(\eta+\eta^2+\tau+\tau^2)^2\sum_{i=1}^{n}\|q_i^k\|^2. \end{aligned} \tag{4.11}$$

Proof. (4.8) can be shown as follows:

$$\left|\sum_{j=1}^{J}\sum_{i=1}^{n} g_j'(\omega^{k,j}) w_i^k g'(v_i^k\cdot\xi^j)\gamma_{k,i}\Delta v_i^k\cdot\xi^j\right| = \left|\sum_{i=1}^{n} q_i^k\cdot(\gamma_{k,i}\Delta v_i^k)\right|$$
$$\le \sum_{i=1}^{n}|\gamma_{k,i}|\|\Delta v_i^k\|\|q_i^k\| \le \tau\sum_{i=1}^{n}\|q_i^k\|^2.$$

To prove (4.9) we take $C_3=\max\{\|\xi^1\|,\|\xi^2\|,\cdots,\|\xi^J\|\}$, so

$$|\Delta v_i^{k+1}\cdot\xi^j| \le C_3\Big(|\gamma_{k,i}|\|\Delta v_i^k\|+\eta\|q_i^k\|\Big) \le C_3(\tau+\eta)\|q_i^k\|.$$

By $(A1)$ and $(A2)$, there exists a $C_4>0$ such that

$$\left|\frac{1}{2}\sum_{j=1}^{J}\sum_{i=1}^{n} g_j'(\omega^{k,j}) w_i^k g''(\hat{t}_{k,i,j})(\Delta v_i^{k+1}\cdot\xi^j)^2\right|$$
$$\le \sum_{i=1}^{n}\sum_{j=1}^{J} C_4(\Delta v_i^{k+1}\cdot\xi^j)^2 \le \sum_{i=1}^{n} JC_4C_3^2(\tau+\eta)^2\|q_i^k\|^2$$
$$= C_5(\tau+\eta)^2\sum_{i=1}^{n}\|q_i^k\|^2,$$

where $C_5=JC_4C_3^2$. Now we prove (4.10). Recalling Remark 2.1, we have $C_6>0$ such that

$$\left|\sum_{j=1}^{J} g_j'(\omega^{k,j})(G^{k+1,j}-G^{k,j})\cdot\Delta w^{k+1}\right| \le C_6\sum_{j=1}^{J}\|G^{k+1,j}-G^{k,j}\|\|\Delta w^{k+1}\|. \tag{4.12}$$

From (2.12) and (2.13) it is easy to know that

$$\|\Delta w^{k+1}\| \le (\tau+\eta)\|p^k\|. \tag{4.13}$$

This, together with (4.5) and (4.12), gives

$$\left|\sum_{j=1}^{J} g_j'(\omega^{k,j})(G^{k+1,j}-G^{k,j})\cdot\Delta w^{k+1}\right|$$
$$\le JC_0C_6(\tau+\eta)(\eta+\tau+\eta^2+\tau^2)\|p^k\|\left(\sum_{i=1}^{n}\|q_i^k\|^2\right)^{\frac{1}{2}}$$
$$\le \frac{1}{2}JC_0C_6(\tau+\eta)(\eta+\tau+\eta^2+\tau^2)\Big(\|p^k\|^2+\sum_{i=1}^{n}\|q_i^k\|^2\Big)$$
$$= C_7(\tau+\eta)(\eta+\tau+\eta^2+\tau^2)\Big(\|p^k\|^2+\sum_{i=1}^{n}\|q_i^k\|^2\Big),$$

where $C_7=\frac{1}{2}JC_0C_6$. Next, we prove (4.11). Note that

$$|\omega^{k+1,j}-\omega^{k,j}|$$
$$= |w^{k+1}\cdot G^{k+1,j}-w^k\cdot G^{k,j}-w^{k+1}\cdot G^{k,j}+w^{k+1}\cdot G^{k,j}|$$
$$\le |G^{k,j}\cdot\Delta w^{k+1}|+|(G^{k+1,j}-G^{k,j})\cdot w^{k+1}|.$$

There exists $C_8 > 0$ such that

$$|G^{k,j} \cdot \Delta w^{k+1}| \le C_8 \|\Delta w^{k+1}\| \le C_8(\tau+\eta)\|p^k\|.$$

Moreover, by $(A2)$ and (4.5), there exists $C_9 > 0$ such that

$$\begin{aligned} &|(G^{k+1,j} - G^{k,j}) \cdot w^{k+1}| \le C_9 \|G^{k+1,j} - G^{k,j}\| \\ &\le C_9 C_0(\eta+\tau+\eta^2+\tau^2)\Big(\sum_{i=1}^{n}\|q_i^k\|^2\Big)^{\frac{1}{2}}. \end{aligned}$$

Consequently,

$$\begin{aligned} &\left|\omega^{k+1,j} - \omega^{k,j}\right|^2 \\ &\le \left[C_8(\tau+\eta)\|p^k\| + C_9C_0(\eta+\tau+\eta^2+\tau^2)(\sum_{i=1}^{n}\|q_i^k\|^2)^{\frac{1}{2}})\right]^2 \\ &\le 2C_8^2(\tau+\eta)^2\|p^k\|^2 + 2C_9^2C_0^2(\eta+\tau+\eta^2+\tau^2)^2\sum_{i=1}^{n}\|q_i^k\|^2. \end{aligned} \tag{4.14}$$

There also exists $C_{10} > 0$ such that

$$\left|\frac{1}{2}\sum_{j=1}^{J} g_j''(t_{k,j})(\omega^{k+1,j} - \omega^{k,j})^2\right| \le C_{10}\sum_{j=1}^{J}\left|\omega^{k+1,j} - \omega^{k,j}\right|^2.$$

This, together with (4.14), yields

$$\begin{aligned} &\left|\frac{1}{2}\sum_{j=1}^{J} g_j''(t_{k,j})(\omega^{k+1,j} - \omega^{k,j})^2\right| \\ &\le C_{11}(\tau+\eta)^2\|p^k\|^2 + C_{11}(\eta+\tau+\eta^2+\tau^2)^2\sum_{i=1}^{n}\|q_i^k\|^2, \end{aligned}$$

where $C_{11} = \max\{2JC_{10}C_8^2,\ 2JC_{10}C_9^2C_0^2\}$. Finally we take $C^* = \max\{C_5, C_7, C_{11}\}$ to complete the proof. □

Lemma 4.4. *If* $(A1)$ *and* $(A2)$ *are satisfied, then for the iteration process* (2.12) *there holds*

$$\begin{aligned} &E(w^{k+1}, V^{k+1}) \\ &\le E(w^k, V^k) - \alpha\|p^k\|^2 - \beta\sum_{i=1}^{n}\|q_i^k\|^2, \quad k = 1, 2, \cdots, \end{aligned} \tag{4.15}$$

where

$$\alpha = \eta - \tau - C^*(\tau+\eta)(\eta+\eta^2+\tau+\tau^2) - C^*(\tau+\eta)^2, \tag{4.16}$$

$$\begin{aligned} \beta = {} & \eta - \tau - C^*(\tau+\eta)^2 - C^*(\tau+\eta)(\eta+\eta^2+\tau+\tau^2) \\ & -C^*(\eta+\eta^2+\tau+\tau^2)^2, \end{aligned} \tag{4.17}$$

and C^* *is the constant defined in Lemma* 4.3.

Proof. From (4.2) and (4.4) we have

$$\begin{aligned}
E(w^{k+1},V^{k+1}) &= E(w^k,V^k) - \eta\|p^k\|^2 - \eta\sum_{i=1}^{n}\|q_i^k\|^2 + \tau_k p^k\cdot\Delta w^k \\
&\quad + \sum_{j=1}^{J}\sum_{i=1}^{n} g_j'(\omega^{k,j}) w_i^k g'(v_i^k\cdot\xi^j)\gamma_{k,i}\Delta v_i^k\cdot\xi^j + \frac{1}{2}\sum_{j=1}^{J}\sum_{i=1}^{n} g_j' \\
&= (\omega^{k,j}) w_i^k g''(\hat{t}_{k,i,j})(\Delta v_i^{k+1}\cdot\xi^j)^2 + \sum_{j=1}^{J} g_j' \\
&= (\omega^{k,j})(G^{k+1,j}-G^{k,j})\cdot\Delta w^{k+1} + \frac{1}{2}\sum_{j=1}^{J} g_j'' = (t_{k,j})(\omega^{k+1,j}-\omega^{k,j})^2.
\end{aligned}$$

Using Lemma 4.3 and noticing

$$|\tau_k p^k\cdot\Delta w^k| \le \tau\|p^k\|^2,$$

we can obtain the desired result (4.15). □

Lemma 4.5. ([14]) *Let $f: R^n \to R$ be continuously differentiable, and assume that the number of the elements of the set $\Omega = \{x|\ f_x(x) = 0\}$ be finite. If the sequence $\{x^k\}$ satisfies*

$$\lim_{k\to\infty}\|x^k - x^{k+1}\| = 0, \quad \lim_{k\to\infty}\|f_x(x^k)\| = 0,$$

then we have

$$\lim_{k\to\infty} x^k = x^*, \quad f_x(x^*) = 0. \tag{4.18}$$

4.2. Proof to Theorem 2.1

It is easy to check that if

$$\eta < \frac{1-s}{C^*[(1+s)(3+2s+s^2)+(2+s+s^2)^2]}, \tag{4.19}$$

then α and β defined by (4.16) and (4.17) are positive. Thus, by (4.15), the sequence $E(w^k,V^k)$ is monotonically decreasing. Note that $E(w^k,V^k)$ is nonnegative, so there exists $E^* \ge 0$ such that

$$\lim_{k\to\infty} E(w^k,V^k) = E^*.$$

It follows from (4.15), (2.10) and (2.11) that

$$\begin{aligned}
&\sum_{k=1}^{\infty}\|E_w(w^k,V^k)\|^2 = \sum_{k=1}^{\infty}\|p^k\|^2 < \infty, \\
&\sum_{k=1}^{\infty}\sum_{i=1}^{n}\|E_{v_i}(w^k,V^k)\|^2 = \sum_{k=1}^{\infty}\sum_{i=1}^{n}\|q_i^k\|^2 < \infty.
\end{aligned}$$

Consequently,

$$\lim_{k\to\infty}\|E_w(w^k,V^k)\| = \lim_{k\to\infty}\|p^k\| = 0, \tag{4.20}$$

$$\lim_{k\to\infty}\|E_{v_i}(w^k,V^k)\| = \lim_{k\to\infty}\|q_i^k\| = 0, \quad i = 1,\cdots,n. \tag{4.21}$$

Finally, we use (2.8), (2.9), (4.7), (4.13), (4.20) and (4.21) to obtain

$$\lim_{k\to\infty} \|w^k - w^{k+1}\| = 0, \quad \lim_{k\to\infty} \|V^k - V^{k+1}\| = 0. \tag{4.22}$$

A combination of Lemma 4.5 and (4.22) yields (2.19)-(2.20). From Lemma 4.4 we know that (w^*, V^*) is a local minimum. We thus complete the proof. □

Acknowledgment. This work was supported by National Natural Science Foundation of China (10471017) and Zhejiang Provincial Natural Science Foundation (Y606009).

References

[1] N. Ampazis and S.J. Perantonis, Two highly efficient second-order algorithms for training feedforward networds, *IEEE T. Neural Networ.*, **13**:5 (2002), 1064-1074.

[2] A. Bhaya and E. Kaszkurewicz, Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method, *Neural Networks*, **17** (2004), 65-71.

[3] S.E. Fahlman, Faster learning variations on back propogation: AN empirical study, in Proc. 1988, Connectionist Models Summer School, San Mateo, CA: Morgan Kaufmann, 38-51.

[4] T.L. Fine and S. Mukherjee, Parameter convergence and learning curves for neural networks, *Neural Comput.*, **11** (1999), 747-769.

[5] W. Finnoff, Diffusion approximations for the constant learning rate backpropagation algorithm and resistance to locol minima, *Neural Comput.*, **6** (1994), 285-295.

[6] M. Gori and M. Maggini, Optimal convergence of on-line backpropagation, *IEEE T. Neural Networ.*, (1996), 251-254.

[7] R.P. Gorman and T.J. Sejnowski, Analysis of hidden units in a layered network trained to classify sonar targets, *Neural Networks*, **1** (1988), 75-89.

[8] V.V. Phansalkar and P.S. Sastry, Analysis of the back-propagation algorithm with momentum, *IEEE T. Neural Networ.*, **5**:3 (1994), 505-506.

[9] N. Qian, On the momentum term in gradient descent learning algorithms, *Neural Networks*, **12** (1999), 145-151.

[10] D.E. Rumelhart and J.L. McClelland, eds., Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, Cambrige, MA: MIT Press, 1986.

[11] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Learning representations by back-propagating errors, *Nature*, **323** (1986), 533-536.

[12] M. Torii and M.T. Hagan, Stability of steepest descent with momentum for quadratic functions, *IEEE T. Neural Networ.*, **13**:3 (2002), 752-756.

[13] W. Wu, H.M. Shao and D. Qu, Strong convergence for gradient methods for BP networks training, Proceedings of 2005 International Conference on Neural Networks and Brains (ICNN-B'05), Edited by M.-S. Zhao and Z.-C. Shi, Beijing, China, 2005, IEEE Press. pp. 332-334.

[14] Y.X. Yuan, W.Y. Sun, Optimization Theory and Methods, Science Press, Beijing, 2001.

[15] N.M. Zhang, W. Wu, and G.F. Zheng, Convergence of gradient method with momentum for two-layer feedforward neural networks, *IEEE T. Neural Networ.*, **17**:2 (2006), 522-525.