Deterministic convergence of conjugate gradient method for feedforward neural networks ☆

Jian Wang[a,b,c], Wei Wu[a], Jacek M. Zurada[b,*]

[a] *School of Mathematical Sciences, Dalian University of Technology, Dalian, 116024, P. R. China*
[b] *Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292, USA*
[c] *School of Mathematics and Computational Sciences, China University of Petroleum, Dongying, 257061, P. R. China*

Abstract

Conjugate gradient methods have many advantages in real numerical experiments, such as fast convergence and low memory requirements. This paper considers a class of conjugate gradient learning methods for backpropagation (BP) neural networks with three layers. We propose a new learning algorithm for almost cyclic BP neural networks based on PRP conjugate gradient method. We then establish the deterministic convergence properties for three different learning fashions, i.e., batch mode, cyclic and almost cyclic learning. There are mainly two deterministic convergence properties including weak and strong convergence which indicate the gradient of the error function goes to zero and the weight sequence goes to a fixed point, respectively. Learning rate plays an important role in the training process of BP neural networks. The deterministic convergence results based on different learning fashions are dependent on different selection strategies of learning rate.

Keywords: Deterministic convergence, conjugate gradient, PRP, feedforward neural networks

1. Introduction

The feedforward neural networks with backpropagation (BP) training procedure have been widely used in various fields of scientific researches and engineer-

☆ Project supported by the National Natural Science Foundation of China (No.10871220)
* Corresponding author.
Email address: jmzura02@louisville.edu (Jacek M. Zurada)

ing applications. The BP algorithm attempts to minimize the least squared error of objective function, defined by the differences between the actual network outputs and the desired outputs (Rumelhart, Hinton, & Williams, 1986). There are two popular ways of learning with training samples to implement the backpropagation algorithm: batch mode and incremental mode (Heskes & Wiegerinck, 1996). In batch training, weight changes are accumulated over an entire presentation of the training samples before being applied, while incremental training updates weights after the presentation of each training sample (Wilson & Martinez, 2003).

There are three incremental learning strategies according to the order that the samples are applied (Heskes & Wiegerinck, 1996; Xu, Zhang, & Jin, 2009; Wu, W. et al, 2010; Wang, Yang, & Wu, 2010). The first strategy is online learning (completely stochastic order). The second strategy is almost-cyclic learning (special stochastic order). The third strategy is cyclic learning (fixed order).

In most cases, the feedforward neural networks are performed by using supervised neural learning techniques, employing the steepest descent method (Park et al, 1991; Saini & Soni, 2002). We mention that the steepest descent method takes consecutive steps in the direction of negative gradient of the performance surface. There has been considerable research on the methods to accelerate the convergence of the steepest descent method (Liu, Liu, & Vemuri, 1993; Papalexopoulos, Hao, & Peng, 1994). Unfortunately, in practice, even with these modifications, the method exhibits oscillatory behavior when it encounters steep valleys, and also progresses too slowly to be effective. An important reason is that the steepest descent method is a simple first order gradient descent method with poor convergence properties.

There have been a number of reports describing the use of second order numerical optimization methods, such as conjugate gradient method (CG), Newton method, to accelerate the convergence of backpropagation algorithm (Saini & Soni, 2002; Goodband, Haas, & Mills, 2008). It is well known that the steepest descent method is the simplest algorithm, but is often slow in converging. Newton method is much faster, but requires the Hessian matrix and its inverse to be calculated. The CG method is something of a compromise; it does not require the calculation of second derivatives, and yet it still has the quadratic convergence property (Hagan, Demuth, & Beale, 1996).

In general, conjugate gradient methods are much more effective than the steepest descent method and are almost as simple to compute. These methods do not attain the fast convergence rates of Newton or quasi-Newton methods, but they have the advantage of not requiring storage of matrices (Nocedal & Wright, 2006). The linear conjugate gradient method was first proposed in (Hestenes & Stiefel,

1952) as an iterative method for solving linear systems with positive definite coefficient matrices. The first nonlinear conjugate gradient method was introduced in (Fletcher & Reeves, 1964). It is one of the earliest known techniques for solving large scale nonlinear optimization problems. Different conjugate gradient methods have been proposed in recent years which depend on the different choices of the descent directions (Dai & Yuan, 2000; Gonzalez & Dorronsor, 2008). There are three classical CG methods such as FR (Fletcher & Reeves, 1964), PRP (Polak & Ribiere, 1969; Polyak, 1969) and HS (Hestenes & Stiefel, 1952) conjugate gradient methods. Among these methods, the PRP method is often regarded as the best one in some practical computations (Shi & Shen, 2007).

The batch mode is common performed by employing the PRP method in feedforward neural networks (Jiang, M. H. et al, 2003). The cyclic learning for PRP method was first presented in (Cichocki, A. et al, 1997). The learning rate is adjusted automatically providing relatively fast convergence at early stages of adaptation while ensuring small final misadjustment for cases of stationary environments. For non-stationary environments, the cyclic learning method proposed have good tracking ability and quick adaptation to abrupt changes, as well as, to produce a small steady state misadjustment.

In (Shen, Shi, & Meng, 2006), a novel algorithm is proposed for blind source separation based on the cyclic PRP conjugate gradient method. The line search method is applied to find the best learning rate. Simulations show the algorithm's ability to perform the separation even with an ill-conditioned mixed matrix. To our best knowledge, the almost cyclic learning for the PRP method has not been discussed until now.

Convergence property for neural networks is an interesting research topic which offers an effective guarantee in practical applications. However, the existing convergence results mainly concentrate upon the steepest descent method. Some weak and strong convergence results based on batch mode training process are proven with special assumptions in recent paper (Xu, Zhang, & Liu, 2010). The convergence results for online learning are mostly asymptotic convergence due to the arbitrariness in the presentation order of the training samples (Terence, 1989; Finnoff, 1994; Chakraborty & Pal, 2003; Zhang, Wu, Liu & Yao, 2009). On the other hand, deterministic convergence lies in cyclic and almost cyclic learning mainly because every sample of the training set is fed exactly once in each training epoch (Xu, Zhang, & Jin, 2009; Wu et al, 2005; Li, Wu, & Tian, 2004; Wu, W. et al, 2010; Wang, Yang, & Wu, 2010).

In this paper, we present a novel study of the deterministic convergence of BP neural networks based on PRP method, including both weak and strong con-

vergence. The weak convergence indicates that the gradient of the error function goes to zero while the weight sequence itself goes to a unique fixed point for the strong convergence. Learning rate is an important parameter in the training process of BP neural networks. From the point of view of mathematical theory, we obtain the convergence results with a constant learning rate for batch mode and a more general choice instead of line search for cyclic and almost cyclic learning. Specially, we demonstrate the following novel contributions:

A) The almost cyclic learning of PRP conjugate gradient method is presented in this paper;

The almost cyclic learning is common for BP neural networks by employing the steepest descent method (Li, Wu, & Tian, 2004). However, there is no reports for almost cyclic learning BP neural networks with PRP conjugate gradient method. We claim that the order of samples can be randomly arranged after each training cycle for almost cyclic PRP learning method.

B) The deterministic convergence of BCG is obtained which includes the strong convergence, i.e., weight sequence goes to a fixed point.

For BCG method, we consider the case of a constant learning rate rule. The weak convergence for general nonlinear optimal problems is proved in (Dai & Yuan, 2000). However, we extend the convergence result including the strong convergence result as well in this paper. Xu, Zhang, & Liu (2010) prove the weak and strong convergence based on batch mode learning of steepest descent method for three complex-valued recurrent neural networks. We mention that BCG method would become the steepest descent method once the conjugate coefficients are set to zero. In additional, the assumptions of the activation functions and the stationary points of error function in this paper are more relaxed and easier to extend the results to the complex-valued recurrent neural networks.

C) The deterministic convergence including weak convergence and strong convergence of CCG for feedforward neural networks are obtained for the first time.

The PRP conjugate gradient method has no global convergence in many situations. Some modified PRP conjugate gradient methods with global convergence were proposed (Grippo & Lucidi, 1997; Khoda, Liu, & Storey, 1992; Nocedal, 1992; Shi, 2002) via adding some strong assumptions or using complicated line

searches. To our best knowledge, the deterministic convergence results in this paper are novel for CCG and ACCG for feedforward neural networks. We note that cyclic learning with steepest descent method is a special case of CCG which based on PRP conjugate gradient method. A dynamic learning strategy is considered in (Cichocki, A. et al, 1997) which depends on the instant conjugate direction. However, from mathematical point of view, a more general case for learning rate is presented below instead of line search strategy.

D) The above deterministic convergence results are also valid for ACCG.

Similarly, almost cyclic learning for steepest descent method is a special case of ACCG once the conjugate coefficients are set to zero in this paper. The convergence assumptions of ACCG for feedforward neural networks are more relaxed than those in (Li, Wu, & Tian, 2004).

The rest of this paper is organized as follows. In Section 2, three updating methods including BCG, CCG and ACCG are introduced. The main convergence results are presented in Section 3 and their proofs are gathered in the Appendix. Some conclusions are drawn in Section 4.

2. Algorithms

2.1. Conjugate Gradient Methods

Consider an unconstrained minimization problem

$$\min f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n. \tag{1}$$

where $\mathbb{R}^n$ denotes an n-dimensional Euclidean space and $f : \mathbb{R}^n \to \mathbb{R}^1$ is a continuously differentiable function.

Generally, a line search method takes the form

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}_k, \quad k = 0, 1, \cdots, \tag{2}$$

where $\mathbf{d}_k$ is a descent direction of $f(\mathbf{x})$ at $\mathbf{x}^k$ and α_k is a step size. For convenience, we denote $\nabla f(\mathbf{x}^k)$ by $\mathbf{g}_k$, $f(\mathbf{x}^k)$ by f_k and $\nabla^2 f(\mathbf{x}^k)$ by G_k . If G_k is available and inverse, then $\mathbf{d}_k = -G_k^{-1}\mathbf{g}_k$ leads to the Newton method and $\mathbf{d}_k = -\mathbf{g}_k$ results in the steepest descent method (Nocedal & Wright, 2006).

In line search methods, the well-known conjugate gradient method has the formulation (2) in which

$$\mathbf{d}_k = \begin{cases} -\mathbf{g}_k, & \text{if} \quad k = 0, \\ -\mathbf{g}_k + \beta_k \mathbf{d}_{k-1}, & \text{if} \quad k \geq 1. \end{cases} \tag{3}$$

where

$$\beta_k^{HS} = \frac{\mathbf{g}_k^T (\mathbf{g}_k - \mathbf{g}_{k-1})}{\mathbf{d}_{k-1}^T (\mathbf{g}_k - \mathbf{g}_{k-1})}, \tag{4}$$

$$\beta_k^{FR} = \frac{\|\mathbf{g}_k\|^2}{\|\mathbf{g}_{k-1}\|^2}, \tag{5}$$

$$\beta_k^{PRP} = \frac{\mathbf{g}_k^T (\mathbf{g}_k - \mathbf{g}_{k-1})}{\|\mathbf{g}_{k-1}\|^2}, \tag{6}$$

or β_k is defined by other formulae (Shi & Guo, 2008). The corresponding methods are called the HS (Hestenes & Stiefel, 1952), FR (Fletcher & Reeves, 1964) and PRP (Polak & Ribiere, 1969; Polyak, 1969) conjugate gradient methods.

2.2. BCG, CCG and ACCG

In this paper, we consider the backpropagation neural networks with three layers based on the PRP conjugate gradient method which corresponding to the three different weight updating fashions: batch mode, cyclic and almost cyclic learning. Suppose the numbers of neurons for the input, hidden and output layers are p, n and 1, respectively. The training sample set is $\{\mathbf{x}^j, O^j\}_{j=0}^{J-1} \subset \mathbb{R}^p \times \mathbb{R}$, where $\mathbf{x}^j$ and O^j are the input and the corresponding ideal output of the *j*-th sample, respectively. Let $\mathbf{V} = (v_{i,j})_{n\times p}$ be the weight matrix connecting the input and hidden layers, and write $\mathbf{v}_i = (v_{i1}, v_{i2}, \cdots, v_{ip})^T$ for $i = 1, 2, \cdots, n$. The weight vector connecting the hidden and output layers is denoted by $\mathbf{u} = (u_1, u_2, \cdots, u_n)^T \in \mathbb{R}^n$. To simplify the presentation, we combine the weight matrix $\mathbf{V}$ with the weight vector $\mathbf{u}$, and write $\mathbf{w} = \left(\mathbf{u}^T, \mathbf{v}_1^T, \cdots, \mathbf{v}_n^T\right)^T \in \mathbb{R}^{n(p+1)}$. 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

$$G(\mathbf{z}) = (g(z_1), g(z_2), \cdots, g(z_n))^T, \quad \forall\, \mathbf{z} \in \mathbb{R}^n. \tag{7}$$

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

$$y = f(\mathbf{u} \cdot G(\mathbf{V}\mathbf{x})). \tag{8}$$

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

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2}\sum_{j=0}^{J-1}(O^j - f(\mathbf{u}\cdot G(\mathbf{V}\mathbf{x}^j)))^2 \\ &= \sum_{j=0}^{J-1} f_j(\mathbf{u}\cdot G(\mathbf{V}\mathbf{x}^j)), \end{aligned} \tag{9}$$

where $f_j(t) = \frac{1}{2}(O^j - f(t))^2,\ j = 0, 1, \cdots, J-1,\ t \in \mathbb{R}$. The gradients of the error function with respect to $\mathbf{u}$ and $\mathbf{v}_i$ are, respectively, given by

$$\begin{aligned} E_{\mathbf{u}}(\mathbf{w}) &= -\sum_{j=0}^{J-1}\left(O^j - y^j\right) f'(\mathbf{u}\cdot G(\mathbf{V}\mathbf{x}^j))G(\mathbf{V}\mathbf{x}^j) \\ &= \sum_{j=0}^{J-1} f_j'(\mathbf{u}\cdot G(\mathbf{V}\mathbf{x}^j))G(\mathbf{V}\mathbf{x}^j), \end{aligned} \tag{10}$$

$$\begin{aligned} E_{\mathbf{v}_i}(\mathbf{w}) &= -\sum_{j=0}^{J-1}\left(O^j - y^j\right) f'(\mathbf{u}\cdot G(\mathbf{V}\mathbf{x}^j))u_i g'(\mathbf{v}_i\cdot\mathbf{x}^j)\mathbf{x}^j \\ &= \sum_{j=0}^{J-1} f_j'(\mathbf{u}\cdot G(\mathbf{V}\mathbf{x}^j))u_i g'(\mathbf{v}_i\cdot\mathbf{x}^j)\mathbf{x}^j. \end{aligned} \tag{11}$$

Write

$$E_{\mathbf{V}}(\mathbf{w}) = \left(E_{\mathbf{v}_1}(\mathbf{w})^T, E_{\mathbf{v}_2}(\mathbf{w})^T, \cdots, E_{\mathbf{v}_n}(\mathbf{w})^T\right)^T, \tag{12}$$

$$E_{\mathbf{w}}(\mathbf{w}) = \left(E_{\mathbf{u}}(\mathbf{w})^T, E_{\mathbf{V}}(\mathbf{w})^T\right)^T. \tag{13}$$

For the sake of brevity, we define the three different learning algorithms based on PRP conjugate gradient method: BCG for batch mode, CCG for cyclic and ACCG for almost cyclic learning, respectively.

2.2.1. BCG

For BCG, we assume that the learning rate of the training process is a positive constant η. For simplicity, we introduce the following notations:

$$G^{m,\,k} = G(\mathbf{V}^m\mathbf{x}^k), \quad E_{\mathbf{w}}^m = E_{\mathbf{w}}(\mathbf{w}^m), \tag{14}$$

$$E_{\mathbf{u}}^m = E_{\mathbf{u}}(\mathbf{w}^m), \quad E_{\mathbf{v}_i}^m = E_{\mathbf{v}_i}(\mathbf{w}^m), \tag{15}$$

$$m = 1, 2, \cdots;\ k = 0, 1, \cdots, J-1;\ i = 1, 2, \cdots, n.$$

Starting from an arbitrary initial weight $\mathbf{w}^0$, we proceed to refine it iteratively by the sequential expressions

$$\mathbf{w}^{m+1} = \mathbf{w}^m + \eta \mathbf{d}^m, \quad m = 1, 2, \cdots, \tag{16}$$

$$\mathbf{d}^m = -E_{\mathbf{w}}^m + \beta^m \mathbf{d}^{m-1}, \quad m = 1, 2, \cdots \tag{17}$$

where

$$\beta^m = \begin{cases} 0 & m = 1, \\ \frac{(E_{\mathbf{w}}^m)^T \left(E_{\mathbf{w}}^m - E_{\mathbf{w}}^{m-1}\right)}{\left\|E_{\mathbf{w}}^{m-1}\right\|^2} & m \geq 2, \end{cases} \tag{18}$$

2.2.2. CCG

Suppose that the learning rate of the training procedure for each cycle is fixed as $\eta_m > 0$. Some notations are simplified as follows:

$$G^{mJ+j,\,k} = G(\mathbf{V}^{mJ+j}\mathbf{x}^k), \quad y^{mJ+j,\,k} = f(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,\,k}), \tag{19}$$

$$g_{\mathbf{u},\,k}^{m,\,j} = f_k' \left(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,\,k}\right) G^{mJ+j,\,k}, \tag{20}$$

$$g_{\mathbf{v}_i,\,k}^{m,\,j} = f_k' \left(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,\,k}\right) u_i^{mJ+j} g'(\mathbf{v}_i^{mJ+j} \cdot \mathbf{x}^k)\mathbf{x}^k, \tag{21}$$

$$g_k^{m,\,j} = \left(\left(g_{\mathbf{u},\,k}^{m,\,j}\right)^T, \left(g_{\mathbf{v}_1,\,k}^{m,\,j}\right)^T, \cdots, \left(g_{\mathbf{v}_n,\,k}^{m,\,j}\right)^T\right)^T. \tag{22}$$

$$\beta^{m,\,j,\,k} = \begin{cases} 0 & m = 0,\ j = 0, \\ \frac{g_k^{m,\,j} \cdot \left(g_k^{m,\,j} - g_k^{m,\,j-1}\right)}{\left\|g_k^{m,\,j-1}\right\|^2} & \text{others.} \end{cases} \tag{23}$$

$$m \in \mathbb{N};\ i = 1, 2, \cdots, n;\ j,\ k = 0, 1, \cdots, J-1.$$

With the above notations and equations, the cyclic conjugate gradient (CCG) training procedure can be formulated as the following iteration:

$$\mathbf{w}^{mJ+j+1} = \mathbf{w}^{mJ+j} + \eta_m \mathbf{d}^{mJ+j}, \tag{24}$$

$$\mathbf{d}^{mJ+j} = -g_j^{m,\,j} + \beta^{m,\,j,\,j} \mathbf{d}^{mJ+j-1}, \tag{25}$$

$$m \in \mathbb{N},\ j = 0, 1, \cdots, J-1.$$

where

$$\beta^{m,\,j,\,j} = \begin{cases} 0 & m = 0,\ j = 0, \\ \frac{g_j^{m,\,j} \cdot \left(g_j^{m,\,j} - g_j^{m,\,j-1}\right)}{\left\|g_j^{m,\,j-1}\right\|^2} & \text{others.} \end{cases} \tag{26}$$

2.2.3. ACCG

We observe that the order of the training samples is fixed for CCG. On the other hand, we can also choose the samples in a special stochastic order (almost cyclic) as follows: For each cycle, let $\{\mathbf{x}^{m(0)}, \mathbf{x}^{m(1)}, \cdots, \mathbf{x}^{m(J-1)}\}$ be a stochastic permutation of the samples set $\{\mathbf{x}^0, \mathbf{x}^1, \cdots, \mathbf{x}^{J-1}\}$. The learning rate of the training procedure for each cycle is fixed as $\eta_m > 0$. Some notations are simplified as follows:

$$G^{mJ+j,\,m(k)} = G(\mathbf{V}^{mJ+j}\mathbf{x}^{m(k)}), \quad y^{mJ+j,\,m(k)} = f(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,\,m(k)}), \tag{27}$$

$$g^{m,\,j}_{\mathbf{u},\,m(k)} = f'_m(k)\left(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,\,m(k)}\right) G^{mJ+j,\,m(k)}, \tag{28}$$

$$g^{m,\,j}_{\mathbf{v}_i,\,m(k)} = f'_{m(k)}\left(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,\,m(k)}\right) u_i^{mJ+j} g'(\mathbf{v}_i^{mJ+j} \cdot \mathbf{x}^{m(k)})\mathbf{x}^{m(k)}, \tag{29}$$

$$g^{m,\,j}_{m(k)} = \left(\left(g^{m,\,j}_{\mathbf{u},\,m(k)}\right)^T, \left(g^{m,\,j}_{\mathbf{v}_1,\,m(k)}\right)^T, \cdots, \left(g^{m,\,j}_{\mathbf{v}_n,\,m(k)}\right)^T\right)^T. \tag{30}$$

$$\beta^{m,\,j,\,m(k)} = \begin{cases} 0 & m = 0,\ j = 0, \\ \dfrac{g^{m,\,j}_{m(k)} \cdot \left(g^{m,\,j}_{m(k)} - g^{m,\,j-1}_{m(k)}\right)}{\left\|g^{m,\,j-1}_{m(k)}\right\|^2} & \text{others}, \end{cases} \tag{31}$$

$$m \in \mathbb{N};\ i = 1, 2, \cdots, n;\ j,\ k,\ m(k) = 0, 1, \cdots, J-1.$$

Now, in place of the above weights updating formulae for CCG, we have the following iteration:

$$\mathbf{w}^{mJ+j+1} = \mathbf{w}^{mJ+j} + \eta_m \mathbf{d}^{mJ+j}, \tag{32}$$

$$\mathbf{d}^{mJ+j} = -g^{m,\,j}_{m(j)} + \beta^{m,\,j,\,m(j)}\mathbf{d}^{mJ+j-1}, \tag{33}$$

$$m \in \mathbb{N},\ j,\ m(j) = 0, 1, \cdots, J-1.$$

where

$$\beta^{m,\,j,\,m(j)} = \begin{cases} 0 & m = 0,\ j = 0, \\ \dfrac{g^{m,\,j}_{m(j)} \cdot \left(g^{m,\,j}_{m(j)} - g^{m,\,j-1}_{m(j)}\right)}{\left\|g^{m,\,j-1}_{m(j)}\right\|^2} & \text{others}. \end{cases} \tag{34}$$

3. Main Results

For any vector $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n$, we write its Euclidean norm as $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^n x_i^2}$. Let $\Omega_0 = \{\mathbf{w} \in \Omega : E_{\mathbf{w}}(\mathbf{w}) = 0\}$ be the stationary point set of the error function $E(\mathbf{w})$, where $\Omega \subset \mathbb{R}^{n(p+1)}$ is a bounded region satisfying

(A3) below. Let $\Omega_{0,s} \subset \mathbb{R}$ be the projection of Ω_0 onto the s-th coordinate axis, that is,

$$\Omega_{0,s} = \left\{ w_s \in \mathbb{R} : \mathbf{w} = (w_1, \cdots, w_s, \cdots, w_{n(p+1)})^T \in \Omega_0 \right\} \tag{35}$$

for $s = 1, 2, \cdots, n(p+1)$. Let constants C_1 and C_2 be defined by (cf. Assumption (A3))

$$\max_{1 \le j \le J} \left\{ \|\mathbf{x}^j\|,\ |O^j| \right\} = C_1,\ \sup_{m \in \mathbb{N}} \|\mathbf{w}^m\| = C_2. \tag{36}$$

To analyze the convergence of the algorithms, we need the following assumptions.

(A1) $g'(t)$ and $f'(t)$ are local Lipschitz continuous;

(A2) The constant learning rate η satisfies

$$0 < \eta < \frac{2L+1}{(3L+1)^2} \tag{37}$$

where $L > 0$ is the corresponding Lipschitz constant (cf. Lemma 4.1).

(A2)′ The diminishing learning rate $\eta_m > 0$ satisfies

$$\sum_{m=0}^{\infty} \eta_m = \infty,\ \sum_{m=0}^{\infty} \eta_m^2 < \infty \text{ and } \sup_{m \in \mathbb{N}} \eta_m \le \frac{1}{4L'}, \tag{38}$$

where L' is the Lipschitz constant (cf. Lemma 4.8).

(A3) There exists a bounded region $\Omega \subset \mathbb{R}^n$ such that $\{\mathbf{w}^m\}_{m=0}^{\infty} \subset \Omega$;

(A4) $\Omega_{0,s}$ does not contain any interior point for every $s = 1, 2, \cdots, n(p+1)$.

Theorem 3.1. *Let assumptions (A1), (A2) and (A3) be valid. For the BCG algorithm, denote the weight sequence starting from arbitrary* $\mathbf{w}^0$ *by* $\mathbf{w}^m$*. We have the following convergence results:*

$$a. \quad E\left(\mathbf{w}^{m+1}\right) \le E\left(\mathbf{w}^m\right), \quad k = 0, 1, 2, \cdots, \tag{39}$$

$$b. \quad \lim_{m \to \infty} E\left(\mathbf{w}^m\right) = E^*, \tag{40}$$

$$c. \quad \lim_{m \to \infty} \|E_{\mathbf{w}}^m\| = 0. \tag{41}$$

Furthermore, if assumption $(A4)$ *is also valid, then there exists a fixed point* $\mathbf{w}^* \in \Omega_0$ *such that*

$$d. \quad \lim_{m \to \infty} \mathbf{w}^m = \mathbf{w}^*. \tag{42}$$

Theorem 3.2. *Suppose that the assumptions (A1), (A2)′ and (A3) are valid. For the CCG and ACCG algorithms, starting from an arbitrary initial value* $\mathbf{w}^0$*, the weight sequence* $\mathbf{w}^m$ *satisfies the following convergence:*

$$1). \quad \lim_{m\to\infty} E\left(\mathbf{w}^m\right) = E^*, \tag{43}$$

$$2). \quad \lim_{m\to\infty} E_{\mathbf{w}}\left(\mathbf{w}^m\right) = 0. \tag{44}$$

Moreover, if the assumption (A4) is also valid, there holds the strong convergence: There exists $\mathbf{w}^* \in \Omega_0$ *such that*

$$3). \quad \lim_{m\to\infty} \mathbf{w}^m = \mathbf{w}^*. \tag{45}$$

Let us make the following three remarks on the above deterministic convergence results:

(I). The learning rate is the big difference of the convergence assumptions between BCG and CCG, ACCG. It is well known that the convergence results are guaranteed with constant learning rate for batch mode (Xu, Zhang, & Liu, 2010) while the diminishing learning rate for incremental mode for neural networks based on the steepest gradient method (Chakraborty & Pal, 2003; Wu et al, 2005; Xu, Zhang, & Jin, 2009; Zhang, Wu, Liu & Yao, 2009; Wu, W. et al, 2010). For the similar reason, we consider the case of a constant learning rate for BCG. However, the diminishing learning rule is adopted for CCG and ACCG.

(II). The monotonicity of the error function is the remarkable difference of the convergence results between BCG and CCG, ACCG. The main reason is that different updating directions are adopted in these different learning fashions. We mention that the descent direction of BCG is dependent on the true conjugate gradient while the instant conjugate gradient is employed in CCG and ACCG. Basically, the BCG method in this paper is a class of solutions for standard unconstrained optimal problems.

(III). We claim that the same convergence results are available for both CCG method and ACCG method. The only difference between these two algorithms is the order of samples which are fed in each training process. It is easy to see that the corresponding lemmas are not influenced by the different updating formulae.

4. Proofs

To be clear, we show in detail the convergence proof for BCG in the following Subsection 4.1. Sequentially, in Subsection 4.2, the detailed proof procedure is manifested for CCG and ACCG.

4.1. Convergence Proof for BCG

Lemma 4.1. *Suppose that the assumptions* $(A1), (A3)$ *are valid, then* $E_{\mathbf{w}}(\mathbf{w})$ *satisfies* $Lipschitz$ *condition, i.e., there exists a positive constant* L*, such that*

$$\left\|E_{\mathbf{w}}\left(\mathbf{w}^{m+1}\right)-E_{\mathbf{w}}\left(\mathbf{w}^{m}\right)\right\| \leq L\left\|\mathbf{w}^{m+1}-\mathbf{w}^{m}\right\|. \tag{46}$$

In turn, we have

$$\left\|E_{\mathbf{w}}\left(\mathbf{w}^{m+1}+t\left(\mathbf{w}^{m+1}-\mathbf{w}^{m}\right)\right)-E_{\mathbf{w}}\left(\mathbf{w}^{m}\right)\right\| \leq L(t+1)\left\|\mathbf{w}^{m+1}-\mathbf{w}^{m}\right\|. \tag{47}$$

PROOF. According to $(A1), (A3)$, it is easy to know that $G^{m,j}$, $f_j'\left(\mathbf{u}^m \cdot G\left(\mathbf{V}^m \mathbf{x}^j\right)\right)$, $g'\left(\mathbf{v}_i^m \cdot \mathbf{x}^j\right) (m \in \mathbb{N}; j = 0, 1, \cdots, J-1; i = 1, 2, \cdots.)$ are all bounded. Furthermore, f_j' is also Lipschitz continuous. Let the positive constant L_1 is the corresponding Lipschitz constant.

$$\begin{aligned}
&\left\|f_j'\left(\mathbf{u}^{m+1} \cdot G^{m+1,j}\right) G^{m+1,j}-f_j'\left(\mathbf{u}^m \cdot G^{m,j}\right) G^{m,j}\right\| \\
&\leq\left|f_j'\left(\mathbf{u}^{m+1} \cdot G^{m+1,j}\right)-f_j'\left(\mathbf{u}^m \cdot G^{m,j}\right)\right|\left\|G^{m+1,j}\right\| \\
&\quad+\left|f_j'\left(\mathbf{u}^m \cdot G^{m,j}\right)\right|\left\|G^{m+1,j}-G^{m,j}\right\| \\
&\leq L_1\|G^{m+1,j}\|\left|\mathbf{u}^{m+1} \cdot G^{m+1,j}-\mathbf{u}^m \cdot G^{m,j}\right| \\
&\quad+\left|f_j'\left(\mathbf{u}^m \cdot G^{m,j}\right)\right|\left\|G^{m+1,j}-G^{m,j}\right\| \\
&\leq L_1\|G^{m+1,j}\|^2\left\|\mathbf{u}^{m+1}-\mathbf{u}^m\right\| \\
&\quad+\left(L_1\|G^{m+1,j}\|\|\mathbf{u}^m\|+|f_j'(\mathbf{u}^m \cdot G^{m,j})|\right)\|G^{m+1,j}-G^{m,j}\| \\
&\leq L_1\|G^{m+1,j}\|^2\left\|\mathbf{u}^{m+1}-\mathbf{u}^m\right\| \\
&\quad+\max_{1 \leq i \leq n} g'(t_i)\|\mathbf{x}^j\| C_2 \sum_{i=1}^{n}\|\mathbf{v}_i^{m+1}-\mathbf{v}_i^m\| \\
&\leq L_2\|\mathbf{w}^{m+1}-\mathbf{w}^m\|.
\end{aligned} \tag{48}$$

where $C_2 = L_1\|G^{m+1,j}\|\|\mathbf{u}^m\|+|f_j'(\mathbf{u}^m \cdot G^{m,j})|$, $L_2 = \sqrt{n+1}\max\{L_1\|G^{m+1,j}\|^2, \max_{1 \leq i \leq n} g'(t_i)\|\mathbf{x}^j\| C_2\}$.

Similarly, we have

$$\begin{aligned}&\left\|f_j'\left(\mathbf{u}^{m+1}\cdot G^{m+1,j}\right)u_i^{m+1}g'\left(\mathbf{v}_i^{m+1}\cdot\mathbf{x}^j\right)\mathbf{x}^j-f_j'\left(\mathbf{u}^m\cdot G^{m,j}\right)u_i^m g'\left(\mathbf{v}_i^m\cdot\mathbf{x}^j\right)\mathbf{x}^j\right\|\\&\leq L_3\left\|\mathbf{w}^{m+1}-\mathbf{w}^m\right\|.\end{aligned}\tag{49}$$

By (12), (13), (48) and (49), we get

$$\begin{aligned}&\left\|E_{\mathbf{w}}\left(\mathbf{w}^{m+1}\right)-E_{\mathbf{w}}\left(\mathbf{w}^m\right)\right\|\\&\leq\left\|E_{\mathbf{u}}\left(\mathbf{w}^{m+1}\right)-E_{\mathbf{u}}\left(\mathbf{w}^m\right)\right\|+\sum_{i=1}^{n}\left\|E_{\mathbf{v}_i}\left(\mathbf{w}^{m+1}\right)-E_{\mathbf{v}_i}\left(\mathbf{w}^m\right)\right\|\\&\leq L\left\|\mathbf{w}^{m+1}-\mathbf{w}^m\right\|\end{aligned}\tag{50}$$

where $L=JL_2+nJL_3$.
Naturally, the formula (47) is valid.

Lemma 4.2. *If $m\geq 1$, then there is the following estimation*

$$\left\|\mathbf{d}^m\right\|\leq C_3\left\|E_{\mathbf{w}}^m\right\|\tag{51}$$

where $C_3=\frac{1-\sqrt{(1-4\eta L)}}{2\eta L}$.

PROOF. It is easy to see that $1<C_3<2$. Then (51) is obvious valid if $m=1$. Suppose that (51) is valid for some $m\geq 1$. Hence, employing Lemma 4.1, we have that

$$\begin{aligned}&\left\|\mathbf{d}^{m+1}\right\|\leq\left\|E_{\mathbf{w}}^{m+1}+\mathbf{d}^{m+1}\right\|+\left\|E_{\mathbf{w}}^{m+1}\right\|\\&\leq\left\|\beta^{m+1}\mathbf{d}^m\right\|+\left\|E_{\mathbf{w}}^{m+1}\right\|\\&\leq\left\|E_{\mathbf{w}}^{m+1}\right\|\frac{\left\|E_{\mathbf{w}}^{m+1}-E_{\mathbf{w}}^m\right\|\left\|\mathbf{d}^m\right\|}{\left\|E_{\mathbf{w}}^m\right\|^2}+\left\|E_{\mathbf{w}}^{m+1}\right\|\\&\leq\left(1+C_3^2\eta L\right)\left\|E_{\mathbf{w}}^{m+1}\right\|\\&=C_3\left\|E_{\mathbf{w}}^{m+1}\right\|.\end{aligned}\tag{52}$$

By mathematical induction, (51) is valid for all $m\geq 1$.

The following lemma is a crucial tool for our analysis, which is basically the same proof process as in (21) of Theorem 3.1 in (Wu, W. et al, 2010). Its proof is thus omitted.

Lemma 4.3. *Let $F:\ \Phi \subset \mathbb{R}^p \to \mathbb{R}, (p \geq 1)$ be continuous for a bounded closed region (Φ), and $\Phi_0 = \{\mathbf{z} \in \Phi:\ F(\mathbf{z}) = 0\}$. If the projection of Φ_0 on each coordinate axis does't contain any interior point. Let the sequence $\{\mathbf{z}^n\}$ satisfy:*

(i). $\lim_{n\to\infty} F(\mathbf{z}^n) = 0$*;*

(ii). $\lim_{n\to\infty} \|\mathbf{z}^{n+1} - \mathbf{z}^n\| = 0$*.*

Then, there exists an unique $\mathbf{z}^* \in \Phi_0$ such that

$$\lim_{n\to\infty} \mathbf{z}^n = \mathbf{z}^*.$$

Proof to (39).
Using the differential mean value theorem, there exists a constant $\theta \in [0, 1]$, such that

$$\begin{aligned}
&E\left(\mathbf{w}^{m+1}\right) - E\left(\mathbf{w}^m\right) \\
&= \left(E_{\mathbf{w}}\left(\mathbf{w}^m + \theta\left(\mathbf{w}^{m+1} - \mathbf{w}^m\right)\right)\right)^T \left(\mathbf{w}^{m+1} - \mathbf{w}^m\right) \\
&= \eta\left(E_{\mathbf{w}}^m\right)^T \mathbf{d}^m + \eta\left[E_{\mathbf{w}}\left(\mathbf{w}^m + \theta\left(\mathbf{w}^{m+1} - \mathbf{w}^m\right)\right) - E_{\mathbf{w}}\left(\mathbf{w}^m\right)\right]^T \mathbf{d}^m \\
&\leq -\eta\left\|E_{\mathbf{w}}^m\right\|^2 + \eta\frac{\left\|E_{\mathbf{w}}^m\right\|^2 \left\|E_{\mathbf{w}}^m - E_{\mathbf{w}}^{m-1}\right\| \left\|\mathbf{d}^{m-1}\right\|}{\left\|E_{\mathbf{w}}^{m-1}\right\|^2} + \eta^2(L+1)\left\|\mathbf{d}^m\right\|^2 \\
&\leq -\eta\left(1 - C_3^2\eta(2L+1)\right)\left\|E_{\mathbf{w}}^m\right\|^2,
\end{aligned} \tag{53}$$

Applying (37) and $C_3 = \frac{1-\sqrt{(1-4\eta L)}}{2\eta L}$, we have that $1 - C_3^2\eta(2L+1) > 0$. The monotone decreasing property of the error function of BCG is thus proved. □

Proof to (40).
By (9), it is to see that $E\left(\mathbf{w}^m\right) \geq 0$ $(m \in \mathbb{N})$. Combining with (39), we then conclude that there exists $E^* \geq 0$ such that

$$\lim_{m\to\infty} E\left(\mathbf{w}^m\right) = E^*.$$

□

Proof to (41).
According to (53) and the fact that (40), we deduce that $\sum_{m=1}^{\infty}\left\|E_{\mathbf{w}}^m\right\|^2 < \infty$. Then, we have

$$\lim_{m\to\infty}\left\|E_{\mathbf{w}}^m\right\| = 0, \tag{54}$$

i.e., the weak convergence for BCG is valid. □

Proof to (42).
By the assumption $(A1)$, it is easy to see that $E_{\mathbf{w}}(\mathbf{w})$ is continuous. Applying (16), (41) and Lemma **??**, we have

$$\lim_{m\to\infty} \|\mathbf{w}^{m+1} - \mathbf{w}^m\| = \eta \lim_{m\to\infty} \|\mathbf{d}^m\| = 0. \tag{55}$$

Furthermore, the assumption $(A4)$ is valid. Then, using the Lemma 4.3, there exists an unique $\mathbf{w}^* \in \Omega_0$, such that

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

This completes the strong convergence proof for BCG.

4.2. Convergence Proof for CCG and ACCG

The following three lemmas are very useful tools in convergence analysis for CCG and ACCG methods, and the specific proofs are presented in (Wu, W. et al, 2010).

Lemma 4.4. Let $q(x)$ be a function defined on a bounded closed interval $[a, b]$ such that $q'(x)$ is Lipschitz continuous with Lipschitz constant $K > 0$. Then, $q'(x)$ is differentiable almost everywhere in $[a, b]$ and

$$|q''(x)| \leq K, \ a.e.\ [a, b]. \tag{56}$$

Moreover, there exists a constant $C > 0$ such that

$$q(x) \leq q(x_0) + q'(x_0)(x - x_0) + C(x - x_0)^2, \quad \forall x_0, x \in [a, b]. \tag{57}$$

Lemma 4.5. *Suppose that the learning rate η_m satisfies (A2) and that the sequence $\{a_m\}$ $(m \in \mathbb{N})$ satisfies $a_m \geq 0$, $\sum_{m=0}^{\infty} \eta_m a_m^{\beta} < \infty$ and $|a_{m+1} - a_m| \leq \mu\eta_m$ for some positive constants β and μ. Then we have*

$$\lim_{m\to\infty} a_m = 0. \tag{58}$$

Lemma 4.6. *Let $\{b_m\}$ be a bounded sequence satisfying $\lim_{m\to\infty}(b_{m+1} - b_m) = 0$. Write $\gamma_1 = \lim_{n\to\infty} \inf_{m>n} b_m$, $\gamma_2 = \lim_{n\to\infty} \sup_{m>n} b_m$ and $S = \{a \in \mathbb{R} :$ There exists a subsequence $\{b_{i_k}\}$ of $\{b_m\}$ such that $b_{i_k} \to a$ as $k \to \infty\}$. Then we have*

$$S = [\gamma_1, \gamma_2]. \tag{59}$$

Lemma 4.7. *Let Y_t, W_t and Z_t be three sequences such that W_t is nonnegative and Y_t is bounded for all t. If*

$$Y_{t+1} \leq Y_t - W_t + Z_t, t = 0, 1, \cdots . \tag{60}$$

and the series $\Sigma_{t=0}^{\infty} Z_t$ is convergent, then Y_t converges to a finite value and $\Sigma_{t=0}^{\infty} W_t < \infty$

PROOF. This Lemma is directly from (Xu, Zhang, & Jin, 2009).

Lemma 4.8. *If the assumptions $(A1)$ and $(A3)$ are valid, then $E_{\mathbf{w}}(\mathbf{w})$ satisfies the Lipschitz condition, i.e., there exists a positive constant L', such that*

$$\|g_j^{m,J+1} - g_j^{m,j}\| \leq L' \|\mathbf{w}^{mJ+j+1} - \mathbf{w}^{mJ+j}\|, \tag{61}$$

$$\left\|E_{\mathbf{w}}\left(\mathbf{w}^l\right) - E_{\mathbf{w}}\left(\mathbf{w}^k\right)\right\| \leq L'' \left\|\mathbf{w}^l - \mathbf{w}^k\right\|, \ l, \ k \in \mathbb{N}. \tag{62}$$

PROOF. The procedure of the detailed proof for this Lemma is similar with the Lemma 4.1 and then omitted.

Lemma 4.9. *Consider the learning algorithm* (24)*,* (25) *and* (26)*. The assumptions $(A1)$, $(A2)'$ and $(A3)$ are valid, then*

$$\left\|\mathbf{d}^{mJ+j}\right\| \leq 2 \left\|g_j^{m,j}\right\|, \ m \in \mathbb{N}, \ j = 0, 1, \cdots, J-1. \tag{63}$$

PROOF. By (26), it is easy to see that $\left\|\mathbf{d}^{mJ+j}\right\| = \left\|g_j^{m,j}\right\|$ for $m = 0$, $j = 0$, and (63) is valid in turn.

Suppose that (63) is valid for $mJ + j > 0$, then according to the assumption $(A2)'$ and the Lipschitz property in Lemma 4.8, we have

$$\begin{aligned}
&\left\|\mathbf{d}^{mJ+j+1}\right\| \leq \left\|g_j^{m,j+1} + \mathbf{d}^{mJ+j+1}\right\| + \left\|g_j^{m,j+1}\right\| \\
&\leq \left\|\beta^{mJ+j+1}\mathbf{d}^{mJ+j}\right\| + \left\|g_j^{m,j+1}\right\| \\
&\leq \left\|g_j^{m,j+1}\right\| \frac{\left\|g_j^{m,j+1} - g_j^{m,j}\right\| \left\|\mathbf{d}^{mJ+j}\right\|}{\left\|g_j^{m,j}\right\|^2} + \left\|g_j^{m,j+1}\right\| \\
&\leq (1 + 4\eta_m L') \left\|g_j^{m,j+1}\right\| \\
&\leq 2 \left\|g_j^{m,j+1}\right\|.
\end{aligned} \tag{64}$$

Thus, by mathematical induction, (63) is valid for $m \in \mathbb{N}, j = 0, 1, \cdots, J-1$.

Lemma 4.10. *Let the sequence* $\{\mathbf{w}^{mJ+j}\}$ *be generated by* (24)*,* (25) *and* (26)*. Under assumptions* $(A1)$*,* $(A2)'$ *and* $(A3)$*, there holds*

$$E\left(\mathbf{w}^{(m+1)J}\right) \leq E\left(\mathbf{w}^{mJ}\right) - \eta_m \left\|E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\|^2 + C_4\eta_m^2. \tag{65}$$

where C_4 *is a positive constant independent of* m *and* η_m

PROOF. Applying the assumptions $(A1), (A3)$ and the Lemma 4.4, we get

$$\begin{aligned} g_j\left(\mathbf{w}^{(m+1)J}\cdot\mathbf{x}^j\right) \leq\ & g_j\left(\mathbf{w}^{mJ}\cdot\mathbf{x}^j\right) + g_j'\left(\mathbf{w}^{mJ}\cdot\mathbf{x}^j\right)\left(\mathbf{w}^{(m+1)J}-\mathbf{w}^{mJ}\right)\cdot\mathbf{x}^j \\ & + C_5\left(\left(\mathbf{w}^{(m+1)J}-\mathbf{w}^{mJ}\right)\cdot\mathbf{x}^j\right)^2. \end{aligned} \tag{66}$$

Summing the above equation from $j=0$ to $j=J-1$, we deduce that

$$\begin{aligned} E\left(\mathbf{w}^{(m+1)J}\right) &\leq E\left(\mathbf{w}^{mJ}\right) + E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\cdot D^{m,J} + JC_1^2C_5\left\|D^{m,J}\right\|^2 \\ &\leq E\left(\mathbf{w}^{mJ}\right) - \eta_m\left\|E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\|^2 + \delta_m. \end{aligned} \tag{67}$$

where

$$D^{m,J} = \mathbf{w}^{(m+1)J} - \mathbf{w}^{mJ} = \eta_m\sum_{j=0}^{J-1} d^{mJ+j},$$

$$\begin{aligned} \delta_m =& -\eta_m E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\cdot\left(E_{\mathbf{w}}\left(\mathbf{w}^{mJ+j}\right) - E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right) \\ & + \eta_m E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\cdot\left(\sum_{j=0}^{J-1}\beta^{mJ+j}d^{mJ+j-1}\right) + JC_1^2C_5\left\|D^{m,J}\right\|^2. \end{aligned}$$

By (36) and the boundedness of $g_j'(t)$, $\left\|E_{\mathbf{w}}\left(\mathbf{w}^{mJ+j}\right)\right\|$ $(m\in\mathbb{N},\ j=0,1,\cdots,J-1)$ is also bounded. According to Lemma 4.8 and Lemma 4.9, we conclude that

$$\begin{aligned} & -\eta_m E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\cdot\left(E_{\mathbf{w}}\left(\mathbf{w}^{mJ+j}\right) - E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right) \\ & \leq L''\eta_m\left\|E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\|\left\|\mathbf{w}^{mJ+j}-\mathbf{w}^{mJ}\right\| \\ & \leq L''\eta_m^2\left\|E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\|\sum_{k=0}^{j-1}\left\|d^{mJ+k}\right\| \leq C_6\eta_m^2. \end{aligned} \tag{68}$$

Similarly, there exists $C_7,\ C_8 > 0$ such that

$$\eta_m E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right) \cdot \left(\sum_{j=0}^{J-1} \beta^{mJ+j} d^{mJ+j-1}\right) \leq C_7 \eta_m^2, \tag{69}$$

$$JC_1^2 C_5 \left\|D^{m,\,J}\right\|^2 \leq C_8 \eta_m^2. \tag{70}$$

Thus, the desired estimate (65) is deduced by setting $C_4 = C_6 + C_7 + C_8$.

Proof to (43) .
According to Lemma 4.7 and Lemma 4.10, there exists a fixed value $E^* \geq 0$ such that

$$\lim_{m\to\infty} E\left(\mathbf{w}^{mJ}\right) = E^*. \tag{71}$$

By the assumption $(A2)$, there holds $\lim_{m\to\infty} \eta_m = 0$. Similar to the proving process of (67), it is easy to conclude that

$$\lim_{m\to\infty} \left|E\left(\mathbf{w}^{mJ+j}\right) - E\left(\mathbf{w}^{mJ}\right)\right| = 0, j = 0, 1, \cdots, J-1.$$

Sequentially, we have

$$\lim_{m\to\infty} E\left(\mathbf{w}^{mJ+j}\right) = \lim_{m\to\infty} E\left(\mathbf{w}^{mJ}\right) = E^*.$$

This completes the proof. □

Proof to (44)
According to Lemma 4.7 and Lemma 4.10, we get

$$\sum_{m=0}^{\infty} \eta_m \left\|E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\|^2 < \infty, \tag{72}$$

Applying the Lemma 4.8, we find that

$$\begin{aligned}\left|\left\|E_{\mathbf{w}}\left(\mathbf{w}^{(m+1)J}\right)\right\| - \left\|E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\|\right| &\leq \left\|E_{\mathbf{w}}\left(\mathbf{w}^{(m+1)J}\right) - E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\| \\ &\leq L_2 \left\|D^{m,\,J}\right\| \leq C_9 \eta_m.\end{aligned} \tag{73}$$

Using (72), (73) and the Lemma 4.5, we see that

$$\lim_{m\to\infty} \left\|E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\| = 0. \tag{74}$$

Similar to the proving process of (73), there exists a constant $C_{10} > 0$, such that

$$\left\| E_{\mathbf{w}}\left(\mathbf{w}^{mJ+j}\right) - E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\| \leq C_{10}\eta_m. \tag{75}$$

It is easy to see that

$$\begin{aligned}\left\| E_{\mathbf{w}}\left(\mathbf{w}^{mJ+j}\right)\right\| &\leq \left\| E_{\mathbf{w}}\left(\mathbf{w}^{mJ+j}\right) - E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\| + \left\| E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\| \\ &\leq C_{10}\eta_m + \left\| E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\|,\end{aligned} \tag{76}$$

Thus, we have $\lim_{m\to\infty}\left\| E_{\mathbf{w}}\left(\mathbf{w}^{mJ+j}\right)\right\| = 0,\ j = 1, 2, \cdots, J-1$. The weak convergence of CCG method is completed. □

Proof to (45).
By the assumption $(A1)$, we can deduce that $E_{\mathbf{w}}(\mathbf{w})$ is continuous. Combining (24), (25) and (44), we get

$$\lim_{m\to\infty} \|\mathbf{w}^{(m+1)J} - \mathbf{w}^{mJ}\| = 0. \tag{77}$$

According the assumption $(A4)$ and Lemma 4.3, there exists an unique $\mathbf{w}^*$, such that

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

By (24), (25), (44) and Lemma 4.9, it is easy to see that

$$\lim_{m\to\infty} \|\mathbf{w}^{mJ+j} - \mathbf{w}^{mJ}\| = 0, \quad j = 0, 1, \cdots.$$

Thus, we conclude that

$$\lim_{m\to\infty} \mathbf{w}^{mJ+j} = \mathbf{w}^*, \quad j = 0, 1, \cdots. \tag{78}$$

This immediately gives the strong convergence for CCG. □

Remark for deterministic convergence of ACCG:
We note that the big difference between CCG and ACCG is the order of the samples in each training cycle. However, every sample is fed exactly once in each cycle. On the basis of this factor, the convergence results for CCG are all available for ACCG. The detailed proof is omitted.

5. Conclusion

Almost cyclic learning for BP neural networks based on conjugate gradient method is presented in this paper. From the mathematical point view, it is novel that the deterministic convergence results for BCG, CCG and ACCG are obtained. On the basis of the selecting strategies of learning rate, the convergence results are guaranteed for BCG and CCG, ACCG, respectively. It makes a big difference to the monotonicity property of the error function whether the updating direction is the true conjugate gradient or not.

References

Chakraborty, D., & Pal, N. R. (2003). A novel training scheme for multilayered perceptrons to realize proper generalization and incremental learning. *IEEE Transactions on Neural Networks*, 14, 1-14.

Cichocki, A., Orsier, B., Back, A., & Amari, S. I. (1997). On-line adaptive algorithms in non-stationary environments using amodified conjugate gradient approach. *Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop*, 316–325.

Dai, Y. H., & Yuan, Y. X. (2000). *Nonlinear conjugate gradient methods*. Shanghai: Shanghai Scientific and Technical Press.

Finnoff, W. (1994). Diffusion approximations for the constant learning rate backpropagation algorithm and resistance to local minima. *Neural Computation*, 6, 242-254.

Fletcher, R., Reeves, C. M. (1964). Function minimization by conjugate gradients. *The Computer Journal*, 7, 149C154.

González, A., Dorronsoro, J. R. (2008). Natural conjugate gradient training of multilayer perceptrons. *Neurocomputing*, 71, 2499-2506.

Goodband, J. H., Haas, O. C. L., & Mills, J. A. (2008). A comparison of neural network approaches for on-line prediction in IGRT. *The international Journal of Medical Physics Research and Practice*, 35, 1113-1112.

Grippo, L., Lucidi, S. (1997). A globally convergent version of the Polak-Ribiére conjugate gradient method. *Mathematical Programming*, 78, 375-391.

Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). *Neural network design*. Boston: PWS Publishing.

Heskes, T., & Wiegerinck, W. (1996). A Theoretical Comparison of Batch-Mode, On-Line, Cyclic, and Almost-Cyclic Learning. *IEEE Transactions on Neural Networks*, 7, 919-925.

Hestenes, M. R., Stiefel, E. (1952). Method of conjugate gradient for solving linear systems. *Journal of Research of the National Bureau of Standards*. 49, 409C436.

Jiang, M. H., Gielen, G., Zhang, B., & Luo, Z. S. (2003). Fast learning algorithms for feedforward neural networks. *Applied Intelligence*, 18, 37-54.

Khoda, K. M., Liu, Y., & Storey, C. (1992). Contributed Papers Generalized Polak-Ribiére algorithm. *Journal of Optimization Theory and Applications*, 75, 345-354.

Li, Z. X., & Ding, X. S. (2005). Prediction of Stock Market by BP Neural Networks with Technical Indexes as Input. *Numerical Mathematics: A Journal of Chinese Universities*, 27, 373-377.

Li, Z. X., Wu, W., & Tian, Y. L. (2004). Convergence of an online gradient method for feedforward neural networks with stochastic inputs. *Journal of Computational and Applied Mathematics*, 163, 165-176.

Liu, C. N., Liu, H. T., & Vemuri, S. (1993). Neural network-based short term load forecasting. *IEEE Transactions on Power Systems*, 8, 336C342.

Nakama, T. (2009). Theoretical analysis of batch and on-line training for gradient descent learning in neural networks. *Neurocomputing*, 73, 151-159.

Nocedal, J. (1992). Theory of algorithms for unconstrained optimization. *Acta Numerica*, 1, 199-242.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (2nd ed.). New York: Springer, (Chapter).

Papalexopoulos, A. D., Hao, S., & Peng, T. M. (1994). An implementation of a neural network-based load forecasting model for the EMS. *IEEE Transactions on Power Systems*, 9, 1956C1962.

Park, D. C., Marks, R. J., Atlas, L. E., & Damborg, M. J. (1991). Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems*, 6, 442C449.

Polak, E., Ribiére, G. (1969). Note sur la convergence de directions conjuguées. *Rev Francaise Infomat Recherche Operatonelle*, 16, 35-43.

Polyak, B. T. (1969). The conjugate gradient method in extreme problems. *USSR Computational Mathematics and Mathematical Physics*. 9, 94-112.

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

Saini, L. M., & Soni, M. K. (2002). Artificial Neural Network-Based Peak Load Forecasting Using Conjugate Gradient Methods. *IEEE Transactions on Power Systems*, 17, 907-912.

Shen, X. Z., Shi, X. Z., & Meng, G. (2006). Online algorithm of blind source separation based on conjugate gradient method. *Circuits Systems Signal Processing*, 25, 381-388.

Shi, Z. J. (2002). Restricted PR conjugate gradient method and its global convergence. *Advances In Mathematics*, 1, 47-55. (in Chinese).

Shi, Z. J., & Guo, J. H. (2008). A new algorithm of nonlinear conjugate gradient method with strong convergence. *Computational & Applied Mathematics*, 27, 93-106.

Shi, Z. J., & Shen, J. (2007). Convergence of the Polak-Ribiére-Polyak conjugate gradient method. *Nonlinear Analysis*, 66, 1428-1441.

Terence, D.S. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2, 459-473.

Wang, J., Yang, J., & Wu, W. (2010). Convergence of Cyclic and Almost-Cyclic learning with momentum for feedforward neural networks. *Ieee Transactions on Neural Networks, Submitted*.

Wilson, D. R., & Martinez, T. R. (2003). The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16, 1429-1451.

Wu, W., Feng, G. R., Li, Z. X., & Xu, Y. S. (2005). Deterministic convergence of an online gradient method for BP neural networks. *IEEE Transactions on Neural Networks*, 16, 533-540.

Wu, W., Wang, J., Cheng, M., & Li, Z. (2010). Convergence analysis of online gradient method for BP neural networks. *Neural Networks, In Press, Corrected Proof*.

Wu, W., & Xu, Y. S. (2002). Deterministic convergence of an on-line gradient method for neural networks. *Journal of Computational and Applied Mathematics*, 144, 335-347.

Xu, D. P., Zhang, H. S., & Liu, L. J. (2010). Convergence Analysis of Three Classes of Split-Complex Gradient Algorithms for Complex-Valued Recurrent Neural Networks. *Neural Computation*, 22, 2655-2677.

Xu, Z. B., Zhang, R., & Jin, W. F. (2009). When Does Online BP Training Converge?. *IEEE Transactions on Neural Networks*, 20, 1529-1539.

Zhang, H. S., Wu, W., Liu, F., & Yao, M. C. (2009). Boundedness and Convergence of Online Gradient Method With Penalty for Feedforward Neural Networks. *IEEE Transactions on Neural Networks*, 20, 1050-1054.