COMPUTATIONAL PROPERTIES AND CONVERGENCE ANALYSIS OF BPNN FOR CYCLIC AND ALMOST CYCLIC LEARNING WITH PENALTY[☆]

Jian Wang[a,b], 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, U.S.*

Abstract

Weight decay method as one of classical complexity regularizations is simple and appears to work well in some applications for backpropagation neural networks (BPNN). This paper shows results for the weak and strong convergence for cyclic and almost cyclic learning BPNN with penalty term (CBP-P and ACBP-P). The convergence is guaranteed under certain relaxed conditions for activation functions, learning rate and under the assumption for the stationary set of error function. Furthermore, the boundedness of the weights in the training procedure is obtained in a simple and clear way. Numerical simulations are implemented to support our theoretical results and demonstrate that ACBP-P has better performance than CBP-P on both convergence speed and generalization ability.

Keywords: Weight decay, Backpropagation, Cyclic, Almost cyclic, Convergence

[☆]Project supported by the National Natural Science Foundation of China (No.11171367) and China Scholarship Council (CSC).

[*]Corresponding author.

Email address: jmzura02@louisville.edu (Jacek M. Zurada)

1. Introduction

Multilayer perceptron network trained with a highly popular algorithm known as the error backpropagation (BP) has been successfully applied to solve some difficult and diverse problems [1, 2]. This algorithm, based on the error-correction learning rule can be viewed as a generalization of the least-mean-square (LMS) algorithm. There are two main modes to implement it: batch learning, in which optimization is carried out with respect to all training samples simultaneously, and incremental learning, where it follows the presentation of each training sample [3].

There are three different incremental BP learning strategies: on-line learning, cyclic learning, and almost cyclic learning [4]. Incremental learning strategies require less storage capacity than batch mode learning. Due to the random presentation order of the training samples, incremental learning implementing the instant gradient of the error function is a stochastic process, whereas batch mode learning corresponds to the standard gradient descent method and is deterministic [4, 5, 6].

It is well known that the general drawbacks of gradient-based BPNN training methods are their more likely divergence and weak generalization. In real-world problems, the BP method is usually prone to require the use of highly structured networks of a rather large size [2]. Thus, it is requisite to reach an appropriate tradeoff between reliability of the training and the goodness of the model. Knowing that the network design is statistical in nature, the tradeoff can be achieved by minimizing the overall risk with regularization theory [7]. A general setting is to add an extra regularization term which is called *penalty term* for BPNN [2].

There are three classical different penalty terms for BPNN: *weight decay* [8], *weight elimination* [9] and *approximate smoother* [10]. In the weight decay procedure, the penalty term is stated as the squared norm of the weights in the BPNN [8, 11]. All the weights in the networks are treated equally. Some of the weights are forced to take values close to zero, while other weights maintain reasonably large values, and consequently improve the generalization of BPNN [2]. In the weight elimination procedure, the complexity penalty represents the complexity of the network as function of the weight magnitudes relative to a pre-assigned parameter [12]. The approximate smoother approach is proposed in [10] for BPNN with a hidden layer and a single output neuron. This method appears to be more accurate than weight decay or weight elimination for the complexity regularization of BPNN. However, it is much more computationally complex than its counterparts [2].

Below we discuss the convergence of BPNN with penalty term from a mathematical point of view. Insofar as the satisfying performance in weight decay method, there are quantitative studies of the convergence property with different BP learning strategies [13, 14, 15, 16, 17, 18, 19].

For batch mode learning, the weak convergence and monotonicity are proved as a special case for the typical gradient descent method of optimization theory. A highlight in [13] is that the boundedness of the weights between input and hidden layers are guaranteed. As an extension, the boundedness of the total weights in the BP feedforward neural networks based on batch learning has been proved in [14]. For online learning, [15] focuses on the linear output of BPNN, while an extension that the activation function satisfies twice continuously differentiable is proposed in [16]. The main contribution of these two papers is to theoretically prove the boundedness of the weights and an almost sure convergence of the approach to the zero set of the gradient of the error function.

Assuming the training samples are supplied in random order in each cycle (almost cyclic), the monotonicity and weak convergence of the almost cyclic learning for BPNN with penalty term (ACBP-P) are guaranteed based on restricted conditions for activation functions and learning rates [17]. Additionally, the results in [17] are valid for BPNN without hidden layer. On the basis of cyclic learning BPNN with penalty term (CBP-P), the convergence results are proved in [18, 19]. A momentum term to speed up the training procedure is considered as well in [19].

Within the framework of BPNN with cyclic and almost-cyclic learning, the latest convergence results concentrate on the regular BPNN [21] and on BPNN with momentum term [22] under much relaxed conditions such as activation functions and learning rates. The training method of BPNN based on the common gradient descent without any additional term is considered in [21]. Furthermore, the strong convergence result was first proved which allows the stationary points of error function to be uncountable somehow. In [22], the weak and strong convergence results have been obtained for BPNN with momentum term which performs much better than regular BPNN. None of the earlier studies focused on convergence results for similar learning modes with penalty term based on relaxed conditions. This paper attempts to fill this gap.

The aim of this paper is to present a comprehensive study for CBP-P and ACBP-P of weak and strong convergence with the identical relaxed training conditions [21, 22], indicating that the gradient of the error function goes to zero and the weight sequence goes to a fixed point, respectively. In comparison to the convergence results which consider the CBP-P and ACBP-P [17, 18, 19], quite simple and general conditions are formulated below for the learning rate and the activation functions to guarantee the convergence. The main points and novel contributions of this paper are as follows:

1) The derivatives g', f' of the activation functions g, f are Lipschitz continuous on $\mathbb{R}$. This improves the corresponding conditions in [17, 18, 19], which requires the boundedness of the second derivatives g'', f''.

From a mathematical point of view, we mention that different analytical tools are employed in [13, 17, 18, 19] and this study for the convergence analysis. The differential Taylor expansion in [13, 17, 18, 19], which requires the boundedness of the second derivative of the activation function g, is considered, while in this paper, we discuss the integral Taylor expansion and hence require the Lipschitz continuity of g', f' on $\mathbb{R}$ [20].

2) The condition on the learning rate in this paper is extended to a more general case: $\sum_{m=0}^{\infty} \eta_m = \infty$; $\sum_{m=0}^{\infty} \eta_m^2 < \infty$, $(\eta_m > 0)$, which is identical to those in [21] for cyclic learning without penalty.

Learning rate is an important criterion in the convergence analysis of BPNN. The convergence results in [19] for cyclic learning with penalty and momentum term focus on no hidden layer feedforward neural networks, and require $\frac{1}{\eta_{k+1}} = \frac{1}{\eta_k} + \beta$, $(k \in \mathbb{N}, \beta > 0)$, where η_k is the learning rate of the k-th training cycle. Basically, this condition is equivalent to $\eta_k = O\left(\frac{1}{k}\right)$. It is easy to see that the conditions on the learning rate are more relaxed in this paper than those in [17, 18, 19].

3) The restrictive assumptions for the strong convergence in [13, 17, 19] are relaxed such that the stationary points set of the error function is only required not to contain any interior point.

To obtain the strong convergence result, which means that the weight sequence converges to a fixed point, an extra condition is considered in [13, 17, 18, 19]: the gradient of the error function has finitely many stationary points. Thus, this additional assumption is a special case in this paper (cf. $(A3)$).

4) The deterministic convergence results are valid for ACBP-P as well.

We mention that CBP-P is typically a deterministic iteration procedure in that the updating fashion is deterministic for fixed order of samples. Due to the random order of samples in each training cycle, the experiment shows that ACBP-P behaves numerically better than CBP-P [17]. In this paper, our convergence results are generalizations of both the results of [18], which considers CBP-P, and of the results of [17, 19], which considers ACBP-P.

Remark: Considering the batch learning BPNN with penalty term we note that this method corresponds to the standard gradient descent algorithm. The convergence results are valid as well once the differential Taylor expansion in [13] is replaced by the integral Taylor expansion in this paper. In addition, a simple and clear proof for the boundedness of the weights is presented.

5) Illustrated experiments have been done to verify the theoretical results of this paper, such as boundedness of the weights, convergence property of BPFNN with penalty term.

Comparing to [23], three different simulations have been performed to demonstrate clearly the important properties of BPFNN with penalty term. Furthermore, one of the classification simulations shows that ACBP-P performs generally much better than CBP-P.

The rest of this paper is organized as follows: Section 2 introduces the two weights updating algorithms: CBP-P and ACBP-P. The main convergence results are presented in Section 3. The performance of the presented two algorithms are reported and discussed in Section 4. The detailed proofs of the main results are stated as Appendix for interested readers.

2. Algorithm Description

Denote the numbers of neurons of the input, hidden and output layers of BPNN are p, n and 1, respectively. Suppose that 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 target output of the j-th sample, respectively. Let $\mathbf{V} = (v_{i,j})_{n\times p}$ be the weight matrix connecting the input and the hidden layer, 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 the 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 the activation functions for the hidden and output layers, respectively. For convenience, we introduce the following function

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

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

$$y = f\left(\mathbf{u} \cdot G(\mathbf{V}\mathbf{x})\right). \tag{2}$$

For fixed weights $\mathbf{w}$, the output error is defined as

$$\begin{aligned} E(\mathbf{w}) &= \frac{1}{2}\sum_{j=0}^{J-1}(O^j - f(\mathbf{u} \cdot G(\mathbf{V}\mathbf{x}^j)))^2 + \lambda\|\mathbf{w}\|^2 \\ &= \sum_{j=0}^{J-1} f_j(\mathbf{u} \cdot G(\mathbf{V}\mathbf{x}^j)) + \lambda\|\mathbf{w}\|^2, \end{aligned} \tag{3}$$

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

$$\begin{aligned} E_{\mathbf{u}}(\mathbf{w}) &= -\sum_{j=0}^{J-1}\left(O^j - y^j\right) f'(\mathbf{u} \cdot G(\mathbf{V}\mathbf{x}^j))G(\mathbf{V}\mathbf{x}^j) + 2\lambda\mathbf{u} \\ &= \sum_{j=0}^{J-1} f_j'(\mathbf{u} \cdot G(\mathbf{V}\mathbf{x}^j))G(\mathbf{V}\mathbf{x}^j) + 2\lambda\mathbf{u}, \end{aligned} \tag{4}$$

$$\begin{aligned} E_{\mathbf{v}_i}(\mathbf{w}) &= -\sum_{j=0}^{J-1}\left(O^j - y^j\right) f'(\mathbf{u} \cdot G(\mathbf{V}\mathbf{x}^j))u_i g'(\mathbf{v}_i \cdot \mathbf{x}^j)\mathbf{x}^j + 2\lambda\mathbf{v}_i \\ &= \sum_{j=0}^{J-1} f_j'(\mathbf{u} \cdot G(\mathbf{V}\mathbf{x}^j))u_i g'(\mathbf{v}_i \cdot \mathbf{x}^j)\mathbf{x}^j + 2\lambda\mathbf{v}_i, \end{aligned} \tag{5}$$

where $y^j = f(\mathbf{u} \cdot G(\mathbf{V}\mathbf{x}^j))$, $i = 1, \cdots, n$ and $j = 0, 1, \cdots, J-1$.
Write

$$E_{\mathbf{V}}(\mathbf{w}) = \left(E_{\mathbf{v}_1}(\mathbf{w})^T, E_{\mathbf{v}_2}(\mathbf{w})^T, \cdots, E_{\mathbf{v}_n}(\mathbf{w})^T\right)^T, \tag{6}$$

$$E_{\mathbf{w}}(\mathbf{w}) = \left(E_{\mathbf{u}}(\mathbf{w})^T, E_{\mathbf{V}}(\mathbf{w})^T\right)^T. \tag{7}$$

2.1. Cyclic Learning of BP with Penalty (CBP-P)

Given an initial weight $\mathbf{w}^0 \in \mathbb{R}^{n(p+1)}$, the cyclic learning of BPNN with penalty term (CBP-P) updates the weights iteratively by

$$\mathbf{u}^{mJ+j+1} = \mathbf{u}^{mJ+j} - \eta_m \nabla_j \mathbf{u}^{mJ+j}, \tag{8}$$

$$\mathbf{v}_i^{mJ+j+1} = \mathbf{v}_i^{mJ+j} - \eta_m \nabla_j \mathbf{v}_i^{mJ+j}. \tag{9}$$

where $\eta_m > 0$ is the learning rate for m-th cycle,

$$\nabla_k \mathbf{u}^{mJ+j} = f_j' \left(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,\, k} \right) G^{mJ+j,\, k} + 2\lambda \mathbf{u}^{mJ+j}, \tag{10}$$

$$\nabla_k \mathbf{v}_i^{mJ+j} = f_j' \left(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,\, k} \right) u_i^{mJ+j} g' \left(\mathbf{v}_i^{mJ+j} \cdot \mathbf{x}^k \right) \mathbf{x}^k + 2\lambda \mathbf{v}_i^{mJ+j}, \tag{11}$$

$$\begin{cases} G^{mJ+j,\, k} = G(\mathbf{V}^{mJ+j}\mathbf{x}^k), \\ y^{mJ+j,\, k} = f(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,\, k}), \end{cases} \tag{12}$$

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

For brevity, the above weights updating indicates

$$\mathbf{w}^{mJ+j+1} = \mathbf{w}^{mJ+j} - \eta_m \nabla_j \mathbf{w}^{mJ+j}, \tag{13}$$

where

$$\mathbf{w}^{mJ+j} = \left(\left(\mathbf{u}^{mJ+j}\right)^T, \left(\mathbf{v}_1^{mJ+j}\right)^T, \cdots, \left(\mathbf{v}_n^{mJ+j}\right)^T \right)^T, \tag{14}$$

$$\nabla_j \mathbf{w}^{mJ+j} = \left(\left(\nabla_j \mathbf{u}^{mJ+j}\right)^T, \left(\nabla_j \mathbf{v}_1^{mJ+j}\right)^T, \cdots, \left(\nabla_j \mathbf{v}_n^{mJ+j}\right)^T \right)^T. \tag{15}$$

2.2. Almost Cyclic Learning of BP with Penalty (ACBP-P)

The order of the training samples for CBP-P is fixed in the whole training procedure. For almost cyclic learning of BP with penalty term (ACBP-P), each sample is chosen with a stochastic order and is fed exactly once in each training 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 is fixed as $\eta_m > 0$ in the m-th cycle. The weights updating follows as:

$$\mathbf{w}^{mJ+j+1} = \mathbf{w}^{mJ+j} - \eta_m \nabla_{m(j)} \mathbf{w}^{mJ+j}, \tag{16}$$

That is,

$$\mathbf{u}^{mJ+j+1} = \mathbf{u}^{mJ+j} - \eta_m \nabla_{m(j)} \mathbf{u}^{mJ+j}, \tag{17}$$

$$\mathbf{v}_i^{mJ+j+1} = \mathbf{v}_i^{mJ+j} - \eta_m \nabla_{m(j)} \mathbf{v}_i^{mJ+j}. \tag{18}$$

where

$$\nabla_{m(k)} \mathbf{u}^{mJ+j} = f_j' \left(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,m(k)} \right) G^{mJ+j,m(k)} + 2\lambda \mathbf{u}^{mJ+j}, \tag{19}$$

$$\nabla_{m(k)} \mathbf{v}_i^{mJ+j} = f_j' \left(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,m(k)} \right) u_i^{mJ+j} g' \left(\mathbf{v}_i^{mJ+j} \cdot \mathbf{x}^{m(k)} \right) \mathbf{x}^{m(k)} + 2\lambda \mathbf{v}_i^{mJ+j}, \tag{20}$$

$$\begin{cases} G^{mJ+j,m(k)} = G(\mathbf{V}^{mJ+j} \mathbf{x}^{m(k)}), \\ y^{mJ+j,m(k)} = f(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,m(k)}), \end{cases} \tag{21}$$

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

3. Summary of 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 $\Omega_0 = \{\mathbf{w} : E_{\mathbf{w}}(\mathbf{w}) = 0\}$ be the stationary point set of the error function $E(\mathbf{w})$. Let $\Omega_{0,s} \subset \mathbb{R}$ be the projection of Ω_0 onto the s-th coordinate axis, that is,

$$\Omega_{0,s} = \left\{ w_s \in \mathbb{R} : \mathbf{w} = (w_1, \cdots, w_s, \cdots, w_{n(p+1)})^T \in \Omega_0 \right\} \tag{22}$$

for $s = 1, 2, \cdots, n(p+1)$. To analyze the convergence of the algorithm, following assumptions are needed:

(A1) $g'(t)$ and $f'(t)$ are Lipschitz continuous on $\mathbb{R}$;

(A2) $\eta_m > 0$, $\sum_{m=0}^{\infty} \eta_m = \infty$, $\sum_{m=0}^{\infty} \eta_m^2 < \infty$;

(A3) $\Omega_{0,s}$ does not contain any interior point for every $s = 1, 2, \cdots, n(p+1)$.

Theorem 3.1. *Assume the Conditions* $(A1)$ *and* $(A2)$ *are valid. Then, starting from an arbitrary initial weight* $\mathbf{w}^0$*, the learning sequence* $\{\mathbf{w}^m\}$ *generated by* (8) *and* (9) *or by* (17) *and* (18) *is uniformly bounded, that is, there exists a positive constant* $C > 0$ *such that*

$$\|\mathbf{w}^m\| < C, \tag{23}$$

and satisfies the following weak convergence

$$\lim_{m\to\infty} \|E_{\mathbf{w}}(\mathbf{w}^m)\| = 0; \tag{24}$$

Moreover, if the assumption $(A3)$ *is also valid, there holds the strong convergence: There exists an unique* $\mathbf{w}^* \in \Omega_0$ *such that*

$$\lim_{m\to\infty} \mathbf{w}^m = \mathbf{w}^*. \tag{25}$$

4. Simulations

In this section, three different simulations are presented to verify the convergence property of CBP-P and ACBP-P. In addition, the performance of CBP-P and ACBP-P with and without penalty are compared for: 4-Parity problem, regression and benchmark classifications. The network architectures for each of the above problems are demonstrated below, respectively. The logistic function *tansig(·)* is employed as the activation function of hidden layer for all of the preceding networks, while the output activation function is different and depends on the network output in terms of the following different applications. To illustrate the convergence results in this paper, we have performed different trials: one trial for the first two simulations, while twenty trials for the third classification problems. We note that the performance of CBP-P and ACBP-P is very similar with slight differences such as effectiveness and stochastic property. Thus, we verify the theoretical results of this paper based on CBP-P in the first two examples and compare the performance of CBP-P and ACBP-P in the last example.

4.1. Example 1: 4-Parity Problem.

In this example, the 4-Parity problem is considered for five inputs (including bias), nine hidden units (including bias) and one output. All transfer functions are *tansig(·)*. This experiment has been conducted by selecting the learning rate η and penalty factor λ with different values from 0.1 to 0.5,

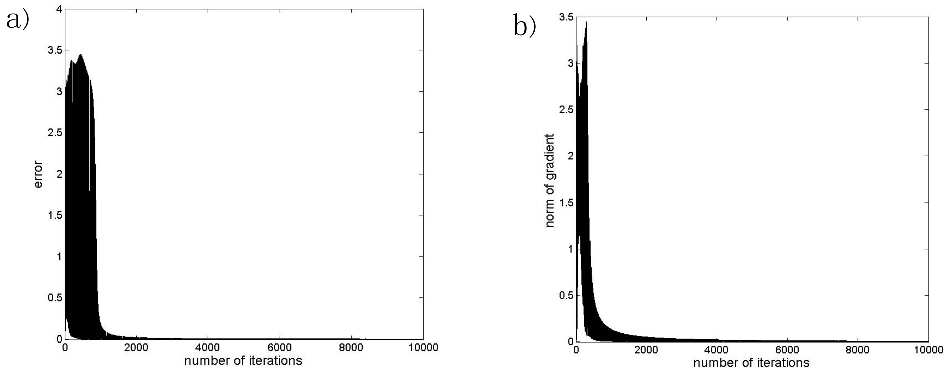


Figure 1: Performance behavior of CBP-P for Example 1, a) Error, b) Norm of gradient.

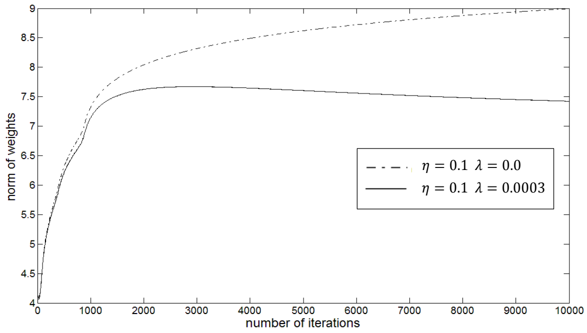


Figure 2: Comparison between CBP-P and CBP of norm of weights for Example 1.

and 0.001 to 0.0001, respectively. The initial weights are randomly chosen in $[-1, 1]$. The training procedure is stopped after $10,000$ iterations or when the error is less than $1e - 6$. We note that the performance behavior of the above tests is consistent with the convergence results which proved in Theorem 3.1. We select one parity of the parameters to show and compare the performance with and without the penalty factor.

The performance results of CBP-P are shown in Fig. 1 for $\eta = 0.1, \lambda = 0.0003$ for 4-Parity problem. It can be seen that the error function decreases monotonically in Fig. 1(a), and the norm of the gradient of error function approaches zero in Fig. 1(b), as depicted by the convergence results in (24). Fig. 2 demonstrates the effectiveness of the algorithm in controlling the magnitude of weights. The norm of weights increases during the training procedure without the penalty term, while the norm of weights initially increases and then remains bounded with penalty term as indicated in the theoretical results (23).

4.2. Example 2: An approximation problem.

In this subsection, we consider the following function demonstrated in [24] to show the function approximation capability of BPNN with penalty term.

$$F(x) = 0.5x - \sin(x), \quad x \in [-4, 4]. \tag{26}$$

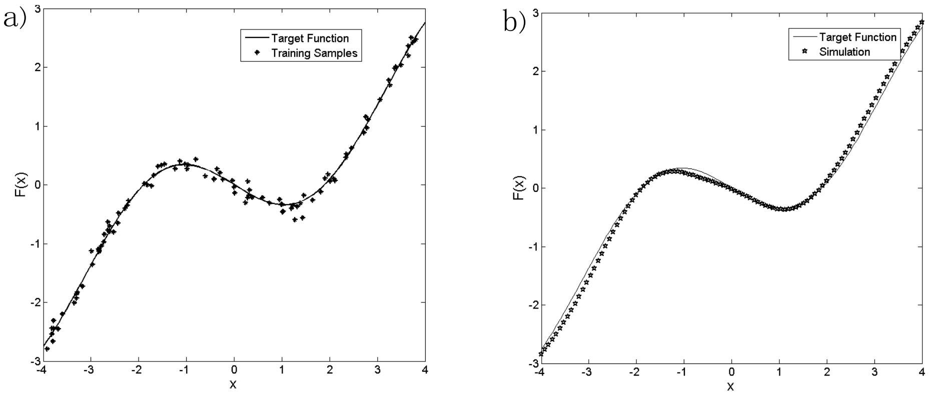


Figure 3: Approximation performance of CBP-P for Example 2, a) Target function and training samples, b) Approximation.

The training pairs are generated as follows: 100 inputs ($x^i, i = 1, \cdots, 100$) are randomly chosen from the interval $[-4, 4]$ with the corresponding outputs $F(x^i) + e_i$, where $e_i \in N(0, 0.1)$ is noise and $N(0, 0.1)$ stands for the normal distribution with expectation and variance being 0 and 0.1, separately. The desired function and the training pairs ("$*$") are shown in Fig. 3(a).

We construct one CBP-P network with 2 input neurons (including bias), 8 hidden neurons and 1 output neuron to implement this approximation problem. The activation function *purelin($\cdot$)* is employed for the output layer in terms of the special approximation problem (26). The initial weights are chosen stochastically in $[-1, 1]$. The training parameters take the following settings: $\eta = 0.02$ and $\lambda = 0.0005$, respectively. The stop criteria are set to be: $10,000$ training cycles or the desired error below $1e - 6$.

Fig. 3(b) shows that CBP-P approximates the presented nonlinear function (26) very well, which demonstrates that BPNN with penalty term can

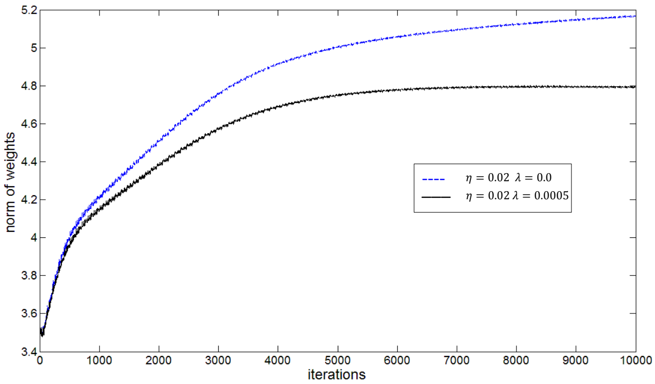
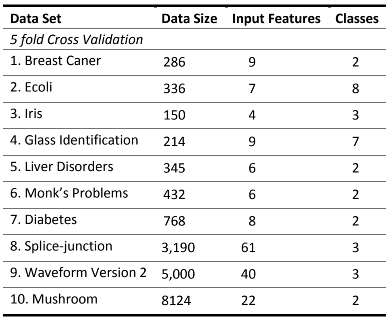


Figure 4: Comparison between CBP-P and CBP of norm of weights for Example 2.

be successfully used for approximation problems. It can be seen that CBP-P in Fig. 4 can effectively control the magnitude of the weights in the training procedure, which shows that the norm of weights for CBP-P tends to be steady.

4.3. Example 3: Benchmark classification problems.

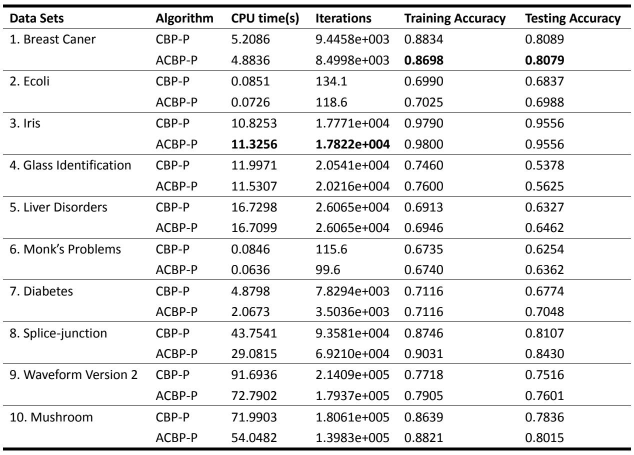
The CBP-P and ACBP-P methods have also been compared using 10 benchmark classification datasets from the UCI Machine Learning Repository [25] as shown in Fig. 5. In this example, 5-fold cross-validation has been performed, i.e. each dataset is randomly split 5 subsets with one set of the five used as testing set while the four remaining subsets as training sets.



Data Set	Data Size	Input Features	Classes
5 fold Cross Validation			
1. Breast Caner	286	9	2
2. Ecoli	336	7	8
3. Iris	150	4	3
4. Glass Identification	214	9	7
5. Liver Disorders	345	6	2
6. Monk's Problems	432	6	2
7. Diabetes	768	8	2
8. Splice-junction	3,190	61	3
9. Waveform Version 2	5,000	40	3
10. Mushroom	8124	22	2

Figure 5: Benchmark classification datasets for Example 3.

To compare the computational performance, all training parameters are identically chosen except for the order of the training, as indicated in Section 2. The original learning rate and penalty factor are set to be 0.1 and 0.0001, separately. The termination criteria are: $30,000$ training cycles or maximum error of $1e-5$ for the first 7 small size datasets and $400,000$ training cycles or maximum error of $1e-5$ for the last 3 datasets.



Data Sets	Algorithm	CPU time(s)	Iterations	Training Accuracy	Testing Accuracy
1. Breast Caner	CBP-P	5.2086	9.4458e+003	0.8834	0.8089
	ACBP-P	4.8836	8.4998e+003	**0.8698**	**0.8079**
2. Ecoli	CBP-P	0.0851	134.1	0.6990	0.6837
	ACBP-P	0.0726	118.6	0.7025	0.6988
3. Iris	CBP-P	10.8253	1.7771e+004	0.9790	0.9556
	ACBP-P	**11.3256**	**1.7822e+004**	0.9800	0.9556
4. Glass Identification	CBP-P	11.9971	2.0541e+004	0.7460	0.5378
	ACBP-P	11.5307	2.0216e+004	0.7600	0.5625
5. Liver Disorders	CBP-P	16.7298	2.6065e+004	0.6913	0.6327
	ACBP-P	16.7099	2.6065e+004	0.6946	0.6462
6. Monk's Problems	CBP-P	0.0846	115.6	0.6735	0.6254
	ACBP-P	0.0636	99.6	0.6740	0.6362
7. Diabetes	CBP-P	4.8798	7.8294e+003	0.7116	0.6774
	ACBP-P	2.0673	3.5036e+003	0.7116	0.7048
8. Splice-junction	CBP-P	43.7541	9.3581e+004	0.8746	0.8107
	ACBP-P	29.0815	6.9210e+004	0.9031	0.8430
9. Waveform Version 2	CBP-P	91.6936	2.1409e+005	0.7718	0.7516
	ACBP-P	72.7902	1.7937e+005	0.7905	0.7601
10. Mushroom	CBP-P	71.9903	1.8061e+005	0.8639	0.7836
	ACBP-P	54.0482	1.3983e+005	0.8821	0.8015

Figure 6: Comparison between CBP-P and ACBP-P for Example 3.

Four performance metrics have been listed in Fig. 6. The CPU time measures the time when perform the training procedure. It can be seen that the training times for ACBP-P are less than CBP-P except for the "Iris" set. The main reason is that the stochastic nature survives in ACBP-P in which the order of training samples are randomly chosen. This shows that ACBP-P training runs much faster than CBP-P in terms of the stochastic property. Training and testing accuracies play a crucial role in measuring the performance of feedforward neural networks. Training accuracy presents the classification capability of BPNN in training procedure, while testing accuracy shows the generalization of BPNN. It can also be seen from Fig. 6 that ACBP-P does much better than CBP-P on the selected benchmark classification problems except for the first "Breast Cancer" set. This demonstrates that ACBP-P has better generalization performance in terms of the stochastic property.

5. Conclusions

Cyclic and almost cyclic learning of BPNN with penalty term (weight decay) are considered in this paper. The weak convergence which indicates that the gradient of the error function goes to zero as the iteration goes to infinity is proved under relaxed conditions of the activation functions and the learning rate. In comparison to existing convergence results, the assumption for the strong convergence in this study is a big-step extension as well. Illustrative experiments are implemented to illustrate theoretical results, and the comparison between CBP-P and ACBP-P shows that stochastic nature plays an important role in improving the performance of ACBP-P.

Appendix

The convergence proof for CBP-P is presented in the following Subsection **A**. Then, in Subsection **B**, we briefly point out how to extend the results to ACBP-P.

The following three lemmas are very useful in convergence analysis for CBP-P and ACBP-P methods, and the specific proofs are presented in [21].

Lemma 1: 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

$$|q''(x)| \leq K, \ \ a.e. \ [a, b]. \tag{27}$$

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

$$q(x) \leq q(x_0) + q'(x_0)(x - x_0) + T(x - x_0)^2, \tag{28}$$

where $x_0, x \in [a, b]$.

Lemma 2: 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

$$\lim_{m\to\infty} a_m = 0. \tag{29}$$

Lemma 3: 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

$$S = [\gamma_1, \gamma_2]. \tag{30}$$

Lemma 4: 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

$$Y_{t+1} \leq Y_t - W_t + Z_t, t = 0, 1, \cdots . \tag{31}$$

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 follows directly from [20].

The following lemma is crucial for the strong convergence analysis, and it basically follows the same proof as in (21) of Theorem 3.1 in [21]. Its proof is thus omitted.

Lemma 5: 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

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

A. *Convergence Analysis for CBP-P*

For brevity, we introduce the following notations:

$$R^{m,\,j} = -\eta_m \left(\nabla_j \mathbf{u}^{mJ+j} - \nabla_j \mathbf{u}^{mJ}\right), \tag{32}$$

$$r_i^{m,\,j} = -\eta_m \left(\nabla_j \mathbf{v}_i^{mJ+j} - \nabla_j \mathbf{v}_i^{mJ}\right), \tag{33}$$

$$d^{m,\,l} = \mathbf{u}^{mJ+l} - \mathbf{u}^{mJ} = -\eta_m \sum_{k=0}^{l-1} \nabla_k \mathbf{u}^{mJ} + \sum_{k=0}^{l-1} R^{m,\,k}, \tag{34}$$

$$h_i^{m,\,j} = \mathbf{v}_i^{mJ+j} - \mathbf{v}_i^{mJ} = -\eta_m \sum_{k=0}^{j-1} \nabla_k \mathbf{v}_i^{mJ} + \sum_{k=0}^{j-1} r_i^{m,\,k}, \tag{35}$$

$$\psi^{m,\,l,\,j} = G^{mJ+l,\,j} - G^{mJ,\,j}, \tag{36}$$

$$\phi^{m,\,J,\,j} = \mathbf{u}^{(m+1)J} \cdot G^{(m+1)J,\,j} - \mathbf{u}^{mJ} \cdot G^{mJ,\,j}. \tag{37}$$

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

The boundedness of the weight sequence is an important property for CBP-P. We firstly give the proof of the boundedness of the weight sequence.

Proof to (23): By the assumption $(A2)$, it is easy to know that $\lim_{m\to\infty} \eta_m = 0$. There exists a positive constant $M_1 \in \mathbb{N}$ such that

$$1 - 2\lambda\eta_m > 0, (m > M_1) \tag{38}$$

Let $A_1 = \max\left\{\left\|\mathbf{u}^{mJ+j}\right\|, m \le M_1, j = 0, \cdots, J-1\right\}$. Applying the assumption $(A1)$, there exists a constant $A_2 > 0$ such that

$$A_2 = \sup\left\{\frac{1}{2\lambda}\left\|g_j'\left(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,\,k}\right) G^{mJ+j,\,k}\right\|\right\},$$

where $m \in \mathbb{N},\ j = 0, \cdots, J-1$. Let $A = \max\{A_1, A_2\}$. By the updating formulas (8) and (10), we have

$$\begin{aligned} \left\|\mathbf{u}^{mJ+j+1}\right\| &\le (1 - 2\lambda\eta_m)\left\|\mathbf{u}^{mJ+j}\right\| + \eta_m \left\|g_j'\left(\mathbf{u}^{mJ+j} \cdot G^{mJ+j,\,k}\right) G^{mJ+j,\,k}\right\| \\ &\le (1 - 2\lambda\eta_m)A + 2\lambda\eta_m A = A. \end{aligned} \tag{39}$$

Using mathematical induction, it is easy to conclude that $\left\|\mathbf{u}^{mJ+j}\right\| \le A$, $j = 0, 1, \cdots, J-1$, $m \in \mathbb{N}$. Similarly, we can get that $\left\|\mathbf{v}_i^{mJ+j}\right\|$ $(m \in \mathbb{N}, i = 1, 2, \cdots, n, j = 0, 1, \cdots, J-1)$ is also bounded. Immediately, we obtain the uniform boundedness of the weight sequence $\{\mathbf{w}^m\}$

$$\left\|\mathbf{w}^{mJ+j}\right\| \le C,\ j = 0, 1, \cdots, J-1,\ m \in \mathbb{N}, \tag{40}$$

where $C > 0$ is a suitable constant. This proof is complete.

Lemma 6: Assume condition $(A1)$ is valid, and let the weight sequence $\mathbf{w}^{mJ+j}$ be generated by (8)-(11). Then there are some positive constants C_1-C_6 such that

$$\left\|G^{mJ+j,\,k}\right\| \le C_1, \tag{41}$$

$$\left\|d^{m,\,l}\right\| \le C_2\eta_m, \tag{42}$$

$$\left\|\psi^{m,\,l,\,j}\right\| \le C_3\eta_m, \tag{43}$$

$$\left\|\phi^{m,\,J,\,j}\right\| \le C_4\eta_m, \tag{44}$$

$$\left\|R^{m,\,j}\right\| \le C_5\eta_m^2, \tag{45}$$

$$\left\|r_i^{m,\,j}\right\| \le C_6\eta_m^2, \tag{46}$$

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

The following lemma demonstrates an almost monotonicity of the error function during the updating procedure.

Lemma 7: Let the weight sequence $\left\{\mathbf{w}^{mJ+j}\right\}$ be generated by (8)-(11). Under condition $(A1)$, there holds

$$E\left(\mathbf{w}^{(m+1)J}\right) \leq E\left(\mathbf{w}^{mJ}\right) - \eta_m \left\|E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\|^2 + C_7\eta_m^2, \tag{47}$$

where $m \in \mathbb{N}$ and $C_7 > 0$ is a constant independent of m and η_m.

Proof: According to assumption $(A1)$ and Lemma 1, we observe that $g''\left(\mathbf{v}_i^{mJ} \cdot \mathbf{x}^j + t\left(h_i^{mJ} \cdot \mathbf{x}^j\right)\right)$ is integrable almost everywhere on $t \in [0,1]$. Thus,

$$\begin{aligned} & f_j'\left(\mathbf{u}^{mJ} \cdot G^{mJ,\,j}\right)\mathbf{u}^{mJ} \cdot \psi^{m,\,J,\,j} \\ &= f_j'\left(\mathbf{u}^{mJ} \cdot G^{mJ,\,j}\right)\sum_{i=1}^{n}\mathbf{u}_i^{mJ} g'(\mathbf{v}_i^{mJ} \cdot \mathbf{x}^j) h_i^{mJ} \cdot \mathbf{x}^j \\ &\quad + f_j'\left(\mathbf{u}^{mJ} \cdot G^{mJ,\,j}\right)\sum_{i=1}^{n}\mathbf{u}_i^{mJ}\left(h_i^{mJ} \cdot \mathbf{x}^j\right)^2 \int_0^1 (1-t) g''\left(\mathbf{v}_i^{mJ} \cdot \mathbf{x}^j + t\left(h_i^{mJ} \cdot \mathbf{x}^j\right)\right) dt. \end{aligned} \tag{48}$$

By virtue of (34) and (35), we have

$$\left\|\mathbf{w}^{(m+1)J}\right\|^2 = \left\|\mathbf{u}^{(m+1)J}\right\|^2 + \sum_{i=1}^{n}\left\|\mathbf{v}_i^{(m+1)J}\right\|^2, \tag{49}$$

$$\left\|\mathbf{u}^{(m+1)J}\right\|^2 = \left\|\mathbf{u}^{mJ}\right\|^2 + 2d^{m,J} \cdot \mathbf{u}^{mJ} + \left\|d^{m,J}\right\|^2, \tag{50}$$

$$\left\|\mathbf{v}_i^{(m+1)J}\right\|^2 = \left\|\mathbf{v}_i^{mJ}\right\|^2 + 2h_i^{m,J} \cdot \mathbf{v}_i^{mJ} + \left\|h_i^{m,J}\right\|^2. \tag{51}$$

Under assumption $(A1)$, it is easy to see that f_j' is Lipschitz continuous. By (10), (11), (34), (35), (48), (49)-(51) and Lemma 1, we obtain that

$$\begin{aligned} & f_j\left(\mathbf{u}^{(m+1)J} \cdot G^{(m+1)J,j}\right) \\ =& f_j(\mathbf{u}^{mJ} \cdot G^{mJ,j}) + f_j'(\mathbf{u}^{mJ} \cdot G^{mJ,j})\left(d^{m,J} \cdot G^{mJ,j} + \mathbf{u}^{mJ} \cdot \psi^{m,J,j} + d^{m,\,J} \cdot \psi^{m,\,J,\,j}\right) \\ &+ \left(\phi^{m,J,j}\right)^2 \int_0^1 (1-t) f_j''\left(\mathbf{u}^{mJ} \cdot G^{mJ,j} + t\phi^{m,J,j}\right) dt. \end{aligned} \tag{52}$$

Furthermore, we have

$$\left(-\eta_m \sum_{j=0}^{J-1} \nabla_j \mathbf{u}^{mJ}\right) \cdot d^{m,J} = \left\|\sum_{j=0}^{J-1} \nabla_j \mathbf{u}^{mJ}\right\|^2 - \eta_m \sum_{j=0}^{J-1} \nabla_j \mathbf{u}^{mJ} \cdot \sum_{j=0}^{J-1} R^{m,\,j}, \tag{53}$$

$$\left(-\eta_m \sum_{j=0}^{J-1} \nabla_j \mathbf{u}^{mJ}\right) \cdot h_i^{m,J} = \left\|\eta_m \sum_{j=0}^{J-1} \nabla_j \mathbf{v}_i^{mJ}\right\|^2 - \eta_m \sum_{j=0}^{J-1} \nabla_j \mathbf{v}_i^{mJ} \cdot \sum_{j=0}^{J-1} r_i^{m,\,j}. \tag{54}$$

On the basis of the above results, we can get that

$$\begin{aligned} E\left(\mathbf{w}^{(m+1)J}\right) &= \sum_{j=0}^{J-1} f_j\left(\mathbf{u}^{(m+1)J} \cdot G^{(m+1)J,\,j}\right) + \lambda \left\|\mathbf{w}^{(m+1)J}\right\|^2 \\ &= E\left(\mathbf{w}^{mJ}\right) - \eta_m \left\|E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right)\right\|^2 + \delta_m, \end{aligned} \tag{55}$$

where

$$\begin{aligned} \delta_m = & \sum_{j=0}^{J-1} \nabla_j \mathbf{u}^{mJ} \cdot \sum_{j=0}^{J-1} R^{m,\,j} + \sum_{i=1}^{n}\left(\sum_{j=0}^{J-1} \nabla_j \mathbf{v}_i^{mJ} \cdot \sum_{j=0}^{J-1} r_i^{m,\,j}\right) \\ & + \sum_{i=1}^{n}\sum_{j=0}^{J-1} f_j'\left(\mathbf{u}^{mJ} \cdot G^{mJ,\,j}\right) \mathbf{u}_i^{mJ} \left(h_i^{mJ} \cdot \mathbf{x}^j\right)^2 \int_0^1 (1-t) g''\left(\mathbf{v}_i^{mJ} \cdot \mathbf{x}^j + t\left(h_i^{mJ} \cdot \mathbf{x}^j\right)\right) dt \\ & + \lambda d^{m,\,J} \cdot d^{m,\,J} + \lambda \sum_{i=1}^{n} h_i^{m,\,J} \cdot h_i^{m,\,J} + \sum_{j=0}^{J-1} f_j'\left(\mathbf{u}^{mJ} \cdot G^{mJ,\,j}\right) d^{m,\,J} \cdot \psi^{m,\,J,\,j} \\ & + \sum_{j=0}^{J-1} \left(\phi^{m,\,J,\,j}\right)^2 \int_0^1 (1-t) f_j''\left(\mathbf{u}^{mJ} \cdot G^{mJ,\,j} + t\phi^{m,\,J,\,j}\right) dt. \end{aligned}$$

Using assumption $(A1)$, (23) and Lemma 6, we can evaluate the first term of δ_m as follows:

$$\left\|\sum_{j=0}^{J-1} \nabla_j \mathbf{u}^{mJ} \cdot \sum_{j=0}^{J-1} R^{m,\,j}\right\| \le \sum_{j=0}^{J-1} \left\|\nabla_j \mathbf{u}^{mJ}\right\| \cdot \sum_{j=0}^{J-1} \left\|R^{m,\,j}\right\| \le C_{7,1}\eta_m^2. \tag{56}$$

where $C_{7,1} > 0$ is a suitable constant.

Similarly, the evaluations for the other terms of δ_m can be accessed with corresponding constants $C_{7,t} > 0$ for $t = 2, \cdots, 7$. Thus, the desired evaluation (47) is obtained by setting $C_7 = \sum_{t=1}^{7} C_{7,t}$.

Proof of (24): By the assumption $(A2)$, Lemma 4 and Lemma 7, we conclude that

$$\sum_{m=0}^{\infty} \eta_m \left\| E_{\mathbf{w}} \left(\mathbf{w}^{mJ} \right) \right\|^2 = \sum_{m=0}^{\infty} \eta_m \left(\| E_{\mathbf{u}} \left(\mathbf{w}^{mJ} \right) \|^2 + \| E_{\mathbf{V}} \left(\mathbf{w}^{mJ} \right) \|^2 \right) < \infty. \tag{57}$$

Naturally, it holds

$$\sum_{m=0}^{\infty} \eta_m \left\| E_{\mathbf{u}} \left(\mathbf{w}^{mJ} \right) \right\|^2 < \infty. \tag{58}$$

Combining (4), (10), there exists a suitable constant $C_8 > 0$ such that

$$\left| \left\| E_{\mathbf{u}} \left(\mathbf{w}^{(m+1)J} \right) \right\| - \left\| E_{\mathbf{u}} \left(\mathbf{w}^{mJ} \right) \right\| \right| \leq \sum_{j=0}^{J-1} \left\| \nabla_j \mathbf{u}^{(m+1)J} - \nabla_j \mathbf{u}^{mJ} \right\| \leq C_8 \eta_m. \tag{59}$$

A combination of (58), (59) and Lemma 2 immediately gives

$$\lim_{m \to \infty} \left\| E_{\mathbf{u}} \left(\mathbf{w}^{mJ} \right) \right\| = 0. \tag{60}$$

Considering the assumption $(A2)$, it is to see that $\lim_{m \to \infty} \eta_m = 0$. By (41), we conclude that

$$\left\| E_{\mathbf{u}} \left(\mathbf{w}^{mJ+j} \right) \right\| \leq \frac{1}{\eta_m} \sum_{j=0}^{J-1} \left\| R^{m,j} \right\| + \left\| E_{\mathbf{u}} \left(\mathbf{w}^{mJ} \right) \right\| \leq J C_5 \eta_m + \left\| E_{\mathbf{u}} \left(\mathbf{w}^{mJ} \right) \right\|. \tag{61}$$

A combination of the above two Eqs. (60) and (61) gives that

$$\lim_{m \to \infty} E_{\mathbf{u}} \left\| \left(\mathbf{w}^{mJ+j} \right) \right\| = 0$$

for $j = 0, 1, \cdots, J-1$. Similarly, it holds that $\lim_{m \to \infty} E_{\mathbf{v}_i} \left\| \left(\mathbf{w}^{mJ+j} \right) \right\| = 0$ Thus, we have

$$\lim_{m\to\infty} \|E_{\mathbf{w}}(\mathbf{w}^m)\| = 0. \tag{62}$$

This completes the proof of weak convergence for CBP-P.

Proof of (25): By the assumptions $(A1)$, it indicates that $E_{\mathbf{w}}(\mathbf{w})$ is continuous. Combining (8), (9), (13) and $(A2)$, we obtain that

$$\lim_{m\to\infty} \left\|\mathbf{w}^{(m+1)J} - \mathbf{w}^{mJ}\right\| = 0. \tag{63}$$

According to the assumption $(A3)$, (24), (63) and Lemma 5, there exists an unique $\mathbf{w}^* \in \Omega_0$ such that

$$\lim_{m\to\infty} \mathbf{w}^{mJ} = \mathbf{w}^*. \tag{64}$$

By the assumption $(A2)$, (13) and the boundedness of $\nabla_j \mathbf{w}^{mJ+j}$ for $m \in \mathbb{N}$, $j = 0, 1, \cdots, J-1$, it holds that

$$\lim_{m\to\infty} \left\|\mathbf{w}^{mJ+j} - \mathbf{w}^{mJ}\right\| = 0. \tag{65}$$

Thus, we conclude that

$$\lim_{m\to\infty} \mathbf{w}^{mJ+j} = \mathbf{w}^*, \quad j = 0, 1, \cdots, J-1. \tag{66}$$

This immediately indicates the strong convergence of CBP-P.

B. *Convergence Analysis for ACBP-P*

Let the weight sequence $\left\{\mathbf{w}^{mJ+j}\right\}$ $(m \in \mathbb{N}; j = 0, 1, \cdots, J-1)$ be updated by (17) and (18). The following notations for ACBP-P are introduced as:

$$R^{m,j} = -\eta_m \left(\nabla_{m(j)} \mathbf{u}^{mJ+j} - \nabla_{m(j)} \mathbf{u}^{mJ}\right), \tag{67}$$

$$r_i^{m,j} = -\eta_m \left(\nabla_{m(j)} \mathbf{v}_i^{mJ+j} - \nabla_{m(j)} \mathbf{v}_i^{mJ}\right), \tag{68}$$

$$\begin{aligned} d^{m,l} &= \mathbf{u}^{mJ+l} - \mathbf{u}^{mJ} \\ &= -\eta_m \sum_{k=0}^{l-1} \nabla_{m(k)} \mathbf{u}^{mJ} + \sum_{k=0}^{l-1} R^{m,k}, \end{aligned} \tag{69}$$

$$\begin{aligned} h_i^{m,j} &= \mathbf{v}_i^{mJ+j} - \mathbf{v}_i^{mJ} \\ &= -\eta_m \sum_{k=0}^{j-1} \nabla_{m(k)} \mathbf{v}_i^{mJ} + \sum_{k=0}^{j-1} r_i^{m,k}, \end{aligned} \tag{70}$$

$$\psi^{m,l,m(j)} = G^{mJ+l,m(j)} - G^{mJ,m(j)}, \tag{71}$$

$$\phi^{m,J,m(j)} = \mathbf{u}^{(m+1)J} \cdot G^{(m+1)J,m(j)} - \mathbf{u}^{mJ} \cdot G^{mJ,m(j)}. \tag{72}$$

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

We mention that the only difference between CBP-P and ACBP-P is the order of training samples. Basically, the related Lemmas can be proved by adjusting the corresponding indexes of the formulas. In contrast to Lemma 6 and Lemma 7, we have the following Lemmas:

Lemma 8: Assume condition $(A1)$ is valid, and let the weight sequence $\mathbf{w}^{mJ+j}$ be generated by (17)-(20). Then there exist some positive constants C_1-C_6 such that

$$\left\| G^{mJ+j,m(k)} \right\| \le C_1, \tag{73}$$

$$\left\| d^{m,l} \right\| \le C_2 \eta_m, \tag{74}$$

$$\left\| \psi^{m,l,m(j)} \right\| \le C_3 \eta_m, \tag{75}$$

$$\left\| \phi^{m,J,m(j)} \right\| \le C_4 \eta_m, \tag{76}$$

$$\left\| R^{m,j} \right\| \le C_5 \eta_m^2, \tag{77}$$

$$\left\| r_i^{m,j} \right\| \le C_6 \eta_m^2, \tag{78}$$

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

Proof: According to the assumption $(A1)$, it is obvious that the activation function $g(t)$ is uniformly bounded on $\mathbb{R}$. Thus, we obtain the same inequality result as (41):

$$\left\| G^{mJ+j,m(k)} \right\| = \left\| G\left(\mathbf{V}^{mJ+j} \mathbf{x}^{m(k)}\right) \right\| \le \sqrt{n} \sup_{t \in \mathbb{R}} g(t) = C_1. \tag{79}$$

Similarly, the remaining inequalities (74)-(78) can be estimated by adjusting the corresponding superscripts.

*Lemma 9:*Let the weight sequence $\left\{\mathbf{w}^{mJ+j}\right\}$ be generated by (17)-(20). Under condition $(A1)$, it holds

$$E\left(\mathbf{w}^{(m+1)J}\right) \le E\left(\mathbf{w}^{mJ}\right) - \eta_m \left\| E_{\mathbf{w}}\left(\mathbf{w}^{mJ}\right) \right\|^2 + C_7 \eta_m^2, \tag{80}$$

where $m \in \mathbb{N}$ and $C_7 > 0$ is the same constant as in Lemma 7.

Proof: It is easy to see that the proof can be completed by replacing the corresponding superscripts in Lemma 7. The details are left to interest readers and thus omitted.

Proof of (24) and (25): For ACBP-P, the weak and strong convergence results can be similarly obtained in terms of Lemmas 1-5 and Lemmas 8-9.

References

[1] D.E. Rumelhart, et al., *Parallel distributed processing: explorations in the microstructure of cognition.* Cambridge, Mass.: MIT Press, 1986.

[2] S.S. Haykin, *Neural networks: a comprehensive foundation.* Upper Saddle River, N.J.: Prentice Hall, 1999.

[3] D. Saad, *On-line learning in neural networks.* Cambridge [England]; New York: Cambridge University Press, 1998.

[4] T. Heskes and W. Wiegerinck, “A theoretical comparison of batch-mode, on-line, cyclic, and almost-cyclic learning”, *IEEE Transactions on Neural Networks*, vol. 7, no. 4, pp. 919-925, 1996.

[5] D.R. Wilson and T.R. Martinez, “The general inefficiency of batch training for gradient descent learning”, *Neural Networks*, vol. 16, no. 10, pp. 1429-1451, 2003.

[6] T. Nakama, “Theoretical analysis of batch and on-line training for gradient descent learning in neural networks”, *Neurocomputing*, vol. 73, no. 1-3, pp. 151-159, 2009.

[7] A.N. Tikhonov, “On solving incorrectly posed problems and method of regularization”, *Doklady Akademii Nauk USSR*, vol. 151, pp. 501-504, 1963.

[8] G.E. Hinton, “Connectionist Learning Procedures”, *Artificial Intelligence*, vol. 40, no. 1-3, pp. 185-234, 1989.

[9] A.S. Weigend, et al., “Generalization by weight-elimination applied to currency exchange rate prediction”, *Proc. of Intern. Joint Conference on Neural Networks, Seattle, WA*, vol. 1, pp. 837-841, 1991.

[10] J.E. Moody and T.S. Rognvaldsson, “Smoothing Regularizers for Projective Basis Function Networks”, *Advances in Neural Information Processing Systems*, 1997.

[11] K. Saito and R. Nakano, “Second-order learning algorithm with squared penalty term”, *Neural Computation*, vol. 12, no. 3, pp. 709-729, 2000.

[12] R. Reed, “Pruning Algorithms - a Survey”, *IEEE Transactions on Neural Networks*, vol. 4, no. 5, pp. 740-747, 1993.

[13] W. Wu, et al., “Convergence of Batch BP Algorithm with Penalty for FNN Training”, *in Neural Information Processing.* vol. 4232, I. King, et al., Eds., ed: Springer Berlin / Heidelberg, pp. 562-569, 2006.

[14] H.S. Zhang, et al., “Boundedness of a Batch Gradient Method with Penalty for Feedforward Neural Networks”, *12th WSEAS Int. Conf. on APPLIED MATHEMATICS, Cairo.*, pp. 175-178, 2007.

[15] H. Zhang and W. Wu, “Boundedness and Convergence of Online Gradient Method with Penalty for Linear Output Feedforward Neural Networks”, *Neural Processing Letters*, vol. 29, no. 3, pp. 205-212, 2009.

[16] H.S. Zhang, et al., “Boundedness and Convergence of Online Gradient Method With Penalty for Feedforward Neural Networks”, *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 1050-1054, 2009.

[17] H.M. Shao, et al., “Convergence of online gradient method with a penalty term for feedforward neural networks with stochastic inputs”, *Numerical Mathematics: A Journal of Chinese Universities*, vol. 14, no. 1, p. 10, 2005.

[18] H.M. Shao, et al., “Convergence and monotonicity of an online gradient method with penalty for neural networks”, *WSEAS Trans. Math.*, vol. 6, pp. 469-476, 2007.

[19] H. Shao and G. Zheng, “Boundedness and convergence of online gradient method with penalty and momentum”, *Neurocomputing*, vol. 74, no. 5, pp. 765-770, 2011.

[20] Z.B. Xu, et al., “When Does Online BP Training Converge?”, *IEEE Transactions on Neural Networks*, vol. 20, no. 10, pp. 1529-1539, 2009.

[21] W. Wu, et al., "Convergence analysis of online gradient method for BP neural networks," *Neural Networks*, vol. 24, no. 1, pp. 91-98, 2011.

[22] J. Wang, J. Yang and W. Wu., "Convergence of Cyclic and Almost-Cyclic Learning With Momentum for Feedforward Neural Networks", *IEEE Transactions on Neural Networks*, vol. 22, pp. 1297-1306, 2011.

[23] J. Wang, W. Wu and J. M. Zurada, "Boundedness and Convergence of MPN for Cyclic and Almost Cyclic Learning with Penalty", *Proc. of Intern. Joint Conference on Neural Networks, San Jose, California*, pp.125-132, 2011, .

[24] Y.J. Ren, "Numerical analysis and the implementations based on Matlab," Beijing, Higer Education Press, 2007.

[25] http://archive.ics.uci.edu/ml.