Optimization Methods and Software, 1994, Vol.4, pp.103-116

SERIAL AND PARALLEL BACKPROPAGATION CONVERGENCE VIA NONMONOTONE PERTURBED MINIMIZATION

O.L. MANGASARIAN and M.V. SOLODOV

Computer Sciences Department, University of Wisconsin,
1210 West Dayton Street, Madison, WI 53706,
email: olvi@cs.wisc.edu, solodov@cs.wisc.edu. *

A general convergence theorem is proposed for a family of serial and parallel nonmonotone unconstrained minimization methods with perturbations. A principal application of the theorem is to establish convergence of backpropagation (BP), the classical algorithm for training artificial neural networks. Under certain natural assumptions, such as divergence of the sum of the learning rates and convergence of the sum of their squares, it is shown that every accumulation point of the BP iterates is a stationary point of the error function associated with the given set of training examples. The results presented cover serial and parallel BP, as well as modified BP with a momentum term.

1 INTRODUCTION

We consider the following unconstrained optimization problem

$$\min_{x \in \Re^n} f(x) \tag{1}$$

where $f : \Re^n \to \Re$ is a continuously differentiable function from the n-dimensional real space $\Re^n$ to the real numbers $\Re$.

We start with a nonmonotone convergence theorem for unconstrained optimization algorithms (Theorem 2.1 below). This result generalizes the monotone Theorem 2.1 of [12] by adding perturbations to the algorithms that result in a nonmonotone sequence of function values. This is a key generalization that allows the proposed theorems to apply to a wider class of algorithms including backpropagation. We shall establish that every accumulation point of the sequence generated by such perturbed algorithms is a stationary point for the minimization problem (1). We further show that algorithms based on appropriately perturbed gradient directions fall within the presented framework (Corollary 2.1).

* This material is based on research supported by Air Force Office of Scientific Research Grant AFOSR-89-0410 and National Science Foundation Grant CCR-9101801.

Of particular interest for us will be the special case when the function $f(x)$ is given by a summation of a finite number of functions $f_j(x), j = 1, \ldots, N$ for some $N \geq 1$. That is

$$f(x) = \sum_{j=1}^{N} f_j(x) \tag{2}$$

We note that this is exactly the case for the BP algorithm for training artificial neural networks [11], where N is the number of examples in the training set.

Motivated by the parallel BP we present a parallel version of Corollary 2.1 (Theorem 2.2). In particular, we consider the case when the functions $f_j(x), j = 1, \ldots, N$ are distributed among k parallel processors. Under the same assumptions as those of Corollary 2.1, convergence of the gradient of the objective function of the problem (1) to zero is established (Theorem 2.2).

We note that a primary goal of this paper is the convergence analysis of the classical BP algorithm [16],[7],[17],[8]. BP has long been successfully used by the artificial intelligence community for training artificial neural networks. Curiously, there seems to be no published *deterministic* convergence results for this method. The primary reason for this seems to be the nonmonotonic nature of the process. Every iteration of on-line BP is a step in the direction of negative gradient of a partial error function associated with a single training example (e.g. $f_j(x)$ in (2)). It is clear that there is no guarantee that such a step will decrease the full objective function $f(x)$, which is the sum of the errors for *all* the training examples . Therefore a single iteration of BP may, in fact, increase rather than decrease the objective function $f(x)$ we are trying to minimize. This difficulty makes convergence analysis of BP a challenging problem that has currently attracted interest of many researchers [5],[6],[10],[4],[18].

In [18] by using *stochastic* approximation ideas [1],[3] it is shown that, under certain stochastic assumptions, the sequence of weights generated by BP either diverges or converges almost surely to a point that is a stationary point of the error function. Some stronger stochastic results were recently obtained in [5]. We emphasize that our approach is purely ***deterministic***. In fact, we show that BP can be viewed as an ordinary perturbed gradient-type algorithm for unconstrained optimization (Theorem 3.1). We give a convergence result for serial and parallel online BP as well as the modified BP with a *momentum term*. There seems to be a consensus in the artificial intelligence community that the use of a momentum term generally yields superior computational results than pure BP [16],[6]. We note that under the same assumptions on the learning rate that we use here, the first two conditions of (27) below, convergence of the sequence of weights for linear feedforward networks without a hidden layer is established in [9].

The paper is organized as follows. In Section 2 we establish the serial and parallel versions of our nonmonotone convergence theorem for unconstrained optimization. We also show that this theorem can be applied to the analysis of a family of optimization methods with perturbed gradients. In Section 3 we concentrate on the online BP algorithm for training neural networks, and apply the results of Section 2 to establish its convergence. Section 4 contains some concluding remarks.

We briefly describe our notation. All the vectors are column-vectors. For x in $\Re^n$, x^T denotes its transpose. Throughout the paper, $\|\cdot\|$ denotes the two-norm, that is $\|x\| = (x^T x)^{\frac{1}{2}}$ for x in $\Re^n$. For a differentiable function $f : \Re^n \to \Re$, ∇f will denote its gradient. If a function f is continuously differentiable on $\Re^n$, we shall write $f \in C^1(\Re^n)$. If f has Lipschitz continuous partial derivatives on $\Re^n$ with some constant $L > 0$, that is

$$\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\| \ \ \forall\ x, y \in \Re^n,$$

we write $f \in LC^1_L(\Re^n)$. $\Re_+$ will denote the nonnegative real line, that is $x \in \Re$ and $x \geq 0$.

2 CONVERGENCE OF ALGORITHMS WITH PERTURBATIONS

We start with a convergent nonmonotone algorithm theorem for the solution of the unconstrained minimization problem (1). Our result is much in the spirit of [12], except for the key difference of nonmonotonicity. We first define a forcing function.

Definition 2.1. *A continuous function $\sigma : \Re_+ \to \Re_+$ such that $\sigma(0) = 0, \sigma(t) > 0$ for $t > 0$, and such that $t^i \geq 0$ and $\{\sigma(t^i)\} \to 0$ imply that $\{t^i\} \to 0$, is said to be a forcing function.*

Some typical examples of forcing functions are ct, ct^2 for some $c > 0$.

We now state a classical lemma ([14],p.144, [15],p.6) that will be used later, as well as another lemma used in the proof of Theorem 2.1.

Lemma 2.1. *Let $f \in LC^1_L(\Re^n)$, then*

$$|f(y) - f(x) - \nabla f(x)^T(y - x)| \leq \frac{L}{2}\|y - x\|^2 \quad \forall x, y \in \Re^n$$

Lemma 2.2.[2] *Let $\{a^i\}$ and $\{\epsilon^i\}$ be two sequences of nonnegative real numbers with $\sum_{i=0}^{\infty} \epsilon^i < \infty$ and $0 \leq a^{i+1} \leq a^i + \epsilon^i$ for $i = 0, 1, \dots$, then the sequence $\{a^i\}$ converges.*

We are now ready to state and prove our first result.

Theorem 2.1. *Let $f \in C^1(\Re^n)$ and let $\inf_{x\in\Re^n} f(x) = \bar{f} > -\infty$. Start with any $x^0 \in \Re^n$. Having x^i stop if $\nabla f(x^i) = 0$, else compute $x^{i+1} = x^i + \eta_i d^i$ according to a direction d^i and stepsize η_i chosen as follows*

***Direction** d^i :*

$$-\nabla f(x^i)^T d^i \geq \sigma(\|\nabla f(x^i)\|) - \lambda_i \tag{3}$$

where $\lambda_i \geq 0$ and $\sigma(\cdot)$ is a forcing function .

***Stepsize** η_i :*

$$f(x^i) - f(x^{i+1}) \geq -\eta_i \nabla f(x^i)^T d^i - \nu_i, \quad \eta_i > 0, \nu_i \geq 0 \tag{4}$$

If

$$\sum_{i=0}^{\infty} \eta_i = \infty, \quad \sum_{i=0}^{\infty} \lambda_i \eta_i < \infty, \quad \sum_{i=0}^{\infty} \nu_i < \infty \tag{5}$$

then the sequence $\{f(x^i)\}$ *converges, and* $\inf_i \|\nabla f(x^i)\| = 0$. *If, in addition,* $f \in LC^1_L(\Re^n)$ *and* $\|d^i\| \leq c, \ \forall\, i, \quad c > 0$, *it follows that* $\{\nabla f(x^i)\} \to 0$, *and for each accumulation point point* $\bar{x}$ *of the sequence* $\{x^i\}$, $\nabla f(\bar{x}) = 0$.

Proof. If $\nabla f(x^{\bar{i}}) = 0$ for some $\bar{i}$, then the algorithm terminates at a stationary point. Suppose now that it does not terminate.

Combining (3) and (4) we have

$$f(x^i) - f(x^{i+1}) \geq \eta_i \sigma(\|\nabla f(x^i)\|) - \lambda_i \eta_i - \nu_i \tag{6}$$

Hence

$$0 \leq f(x^{i+1}) - \bar{f} \leq f(x^i) - \bar{f} + \lambda_i \eta_i + \nu_i$$

By (5) and Lemma 2.2, the sequence $\{f(x^i) - \bar{f}\}$ converges, and so does the sequence $\{f(x^i)\}$.

Applying (6) to the first summation below we obtain

$$\begin{aligned} f(x^0) - \bar{f} &\geq f(x^0) - f(x^i) = \sum_{j=0}^{i-1} (f(x^j) - f(x^{j+1})) \\ &\geq \sum_{j=0}^{i-1} \eta_j \sigma(\|\nabla f(x^j)\|) - \sum_{j=0}^{i-1} (\lambda_j \eta_j + \nu_j) \\ &\geq \inf_{0 \leq j \leq i-1} \sigma(\|\nabla f(x^j)\|) \sum_{j=0}^{i-1} \eta_j - \sum_{j=0}^{i-1} \lambda_j \eta_j - \sum_{j=0}^{i-1} \nu_j \end{aligned} \tag{7}$$

By letting $i \to \infty$ we obtain

$$f(x^0) - \bar{f} \geq \inf_{j \geq 0} \sigma(\|\nabla f(x^j)\|) \sum_{j=0}^{\infty} \eta_j - \sum_{j=0}^{\infty} \lambda_j \eta_j - \sum_{j=0}^{\infty} \nu_j \tag{8}$$

Since the left-hand-side and the last two terms of the right-hand-side in (8) are finite numbers, it follows from the divergence of $\sum_{j=0}^{\infty} \eta_j$ that $\inf_j \sigma(\|\nabla f(x^j)\|) = 0$. By Definition 2.1 of the forcing function we immediately have that

$$\inf_i \|\nabla f(x^i)\| = 0 \tag{9}$$

Now assume that $f \in LC^1_L(\Re^n)$ and $\|d^i\| \leq c, \quad \forall i, \ c > 0$. Suppose the sequence $\{\nabla f(x^i)\}$ does not converge to zero. Then there exists some $\epsilon > 0$ and some increasing sequence of integers $\{i_l\}$ such that $\|\nabla f(x^{i_l})\| \geq \epsilon$ for all l. On the other hand, (9) guarantees that for every l there exists some $j > i_l$ such that $\|\nabla f(x^j)\| \leq \frac{\epsilon}{2}$. For each l let $j(l)$ denote the least integer which satisfies these conditions. By the triangle inequality, the fact that $f \in LC_L^1(\Re^n)$ and (5), we have

$$\begin{aligned}\frac{\epsilon}{2} &\leq \|\nabla f(x^{i_l})\| - \|\nabla f(x^{j(l)})\| \leq \|\nabla f(x^{i_l}) - \nabla f(x^{j(l)})\| \\ &\leq L\|x^{i_l} - x^{j(l)}\| \leq L\sum_{t=i_l}^{j(l)-1} \eta_t \|d^t\| \leq Lc\sum_{t=i_l}^{j(l)-1} \eta_t\end{aligned}$$

Hence

$$\sum_{t=i_l}^{j(l)-1} \eta_t \geq \frac{\epsilon}{2Lc} = \bar{c} > 0 \tag{10}$$

By making use of (6) and (10), we have

$$\begin{aligned}f(x^{i_l}) - f(x^{j(l)}) &\geq \sum_{t=i_l}^{j(l)-1} \eta_t \sigma(\|\nabla f(x^t)\|) - \sum_{t=i_l}^{j(l)-1} (\lambda_t \eta_t + \nu_t) \\ &\geq \bar{c} \inf_{i_l \leq t \leq j(l)-1} \sigma(\|\nabla f(x^t)\|) - \sum_{t=i_l}^{\infty}(\lambda_t \eta_t + \nu_t)\end{aligned}$$

Since the sequence $\{f(x^i)\}$ converges and the last summation above converges to zero as $l \to \infty$, it follows that

$$\lim_{l\to\infty} \inf_{i_l \leq t \leq j(l)-1} \sigma(\|\nabla f(x^t)\|) = 0 \tag{11}$$

However, by the choice of i_l and $j(l)$, $\|\nabla f(x^t)\| \geq \frac{\epsilon}{2}$, $\forall t : i_l \leq t < j(l)$. This contradicts (11) since $\sigma(\cdot)$ is a forcing function. Hence the assumption that $\nabla f(x^i)$ does not converge to zero is invalid. Taking into account continuity of the gradient of f, we conclude that if $\bar{x}$ is an accumulation point of $\{x^i\}$, then $\nabla f(\bar{x}) = 0$. The proof is complete.

∎

Remark 2.1. Assumptions (3), (4) and (5) can be combined into the following simpler and more general condition, where θ_i replaces $\lambda_i \eta_i + \nu_i$:

$$f(x^i) - f(x^{i+1}) \geq \eta_i \sigma(\|\nabla f(x^i)\|) - \theta_i$$

$$\sum_{i=0}^{\infty} \eta_i = \infty, \quad \sum_{i=0}^{\infty} \theta_i < \infty$$

These new conditions also guarantee that the assertions of Theorem 2.1 hold. However, we have chosen to state Theorem 2.1 in a direction – stepsize form because it is easier to implement. See [12] for specific instances of directions d^i and stepsize η_i choices without perturbation terms.

We now show that Theorem 2.1 can be applied to the analysis of the perturbed gradient-type methods. It is important to point out that the assumptions (12) below of Corollary 2.1 of boundedness of f from below, and the Lipschitz continuity and boundedness of ∇f are all satisfied in the context of BP, the convergence of which is established in Section 3.

Corollary 2.1. *Let*

$$f \in LC^1_L(\Re^n), \ \ \|\nabla f(x)\| \le M, \ \ f(x) \ge \bar{f} \quad \forall x \in \Re^n \tag{12}$$

for some $M > 0$ *and some* $\bar{f}$. *Start with any* $x^0 \in \Re^n$. *Having* x^i, *stop if* $\nabla f(x^i) = 0$, *else compute*

$$x^{i+1} = x^i + \eta_i d^i \tag{13}$$

where

$$d^i = -\nabla f(x^i) + e^i \tag{14}$$

for some $e^i \in \Re^n$, $\eta_i \in \Re$, $\eta_i > 0$ *and such that*

$$\sum_{i=0}^{\infty} \eta_i = \infty, \ \ \sum_{i=0}^{\infty} \eta_i^2 < \infty, \ \ \sum_{i=0}^{\infty} \eta_i \|e^i\| < \infty, \ \ \|e^i\| \le \gamma \ \ \forall i, \ \ \gamma > 0 \tag{15}$$

It follows that $\{f(x^i)\}$ *converges,* $\{\nabla f(x^i)\} \to 0$ *and for each accumulation point* $\bar{x}$ *of the sequence* $\{x^i\}$, $\nabla f(\bar{x}) = 0$.

Proof. It suffices to show that conditions (3)–(5) of Theorem 2.1 are satisfied. We first note that, by (12), (14) and (15), $\|d^i\| \le M + \gamma$ for all i.

By the Cauchy-Schwartz inequality, (12) and (13), we have with $\sigma(s) = s^2$,

$$\begin{aligned} -\nabla f(x^i)^T d^i &= \|\nabla f(x^i)\|^2 - \nabla f(x^i)^T e^i \\ &\ge \sigma(\|\nabla f(x^i)\|) - \|\nabla f(x^i)\|\|e^i\| \\ &\ge \sigma(\|\nabla f(x^i)\|) - M\|e^i\| \end{aligned} \tag{16}$$

By Lemma 2.1, (13) and (14), it follows

$$\begin{aligned} f(x^i) - f(x^{i+1}) &\ge -\nabla f(x^i)^T (x^{i+1} - x^i) - \frac{L}{2}\|x^{i+1} - x^i\|^2 \\ &= -\eta_i \nabla f(x^i)^T d^i - \frac{L}{2}\eta_i^2 \|d^i\|^2 \\ &\ge -\eta_i \nabla f(x^i)^T d^i - \frac{L}{2}\eta_i^2 (M + \gamma)^2 \end{aligned} \tag{17}$$

Relations (16), (17) and (15) establish the assumptions (3)–(5) of Theorem 2.1 with $\lambda_i = M\|e^i\|$, $\nu_i = \frac{L}{2}(M + \gamma)^2 \eta_i^2$. The proof is complete. ∎

Remark 2.2. Under appropriate assumptions, other well known direction choices, such as conjugate and quasi-Newton directions [15] can also be perturbed similarly as in Corollary 2.1.

Remark 2.3. Similar to [12], a parallel version of Theorem 2.1 can be established where portions of the gradient are distributed among the processors. However, having in mind the analysis of the BP algorithm, we shall instead here concentrate on parallel distribution of the objective function in the form (2).

We now turn our attention to the case when the objective function of the problem is represented by the sum of a finite number of functions as in (2). Suppose that we have k parallel processors, $k \geq 1$. Let J_l be a partition of $\{1, \ldots, N\}$ such that $J_l \subseteq \{1, \ldots, N\}$, $\cup_{l=1}^{k} J_l = \{1, \ldots, N\},$; $J_{l_1} \cap J_{l_2} = \emptyset$ *for* $l_1 \neq l_2$. Let N_l be the number of elements in J_l. We define the function f^l associated with J_l as follows

$$f^l(x) = \sum_{j \in J_l} f_j(x) \tag{18}$$

With this definition we have

$$f(x) = \sum_{l=1}^{k} f^l(x) \tag{19}$$

We are now ready to state and prove a parallel version of Corollary 2.1.

Theorem 2.2. *Let each f_j, $j = 1, \ldots, N$, satisfy the assumptions (12) of Corollary 2.1. Start with any $x^0 \in \Re^n$. Having x^i, stop if $x^i = x^{i-1}$. Else compute x^{i+1} as follows:*
(i) Parallelization. *For each processor $l \in \{1, \ldots, k\}$ compute*

$$y_l^{i+1} = x^i + \eta_i d_l^i \tag{20}$$

where

$$d_l^i = -\nabla f^l(x^i) + e_l^i, \quad \eta_i > 0 \tag{21}$$

(ii) Synchronization. *Let*

$$x^{i+1} = \frac{1}{k} \sum_{l=1}^{k} y_l^{i+1} \tag{22}$$

If for some $\gamma > 0$

$$\sum_{i=0}^{\infty} \eta_i = \infty, \ \sum_{i=0}^{\infty} \eta_i^2 < \infty, \ \sum_{i=0}^{\infty} \eta_i \|e_l^i\| < \infty, \ \|e_l^i\| \leq \gamma, \ \forall\, i, \ \ l = 1, \ldots, k \tag{23}$$

all the conclusions of Corollary 2.1 hold.

Proof. We shall establish assumptions (13)– (15) of Corollary 2.1.
By (19) and (20)–(22), we have

$$\begin{aligned} x^{i+1} - x^i &= \frac{1}{k} \sum_{l=1}^{k} y_l^{i+1} - x^i = \frac{1}{k} \sum_{l=1}^{k} (x^i + \eta_i d_l^i) - x^i \\ &= \frac{1}{k} \sum_{l=1}^{k} \eta_i d_l^i = \frac{\eta_i}{k} \sum_{l=1}^{k} (-\nabla f^l(x^i) + e_l^i) \end{aligned}$$

$$= \frac{\eta_i}{k}\left(-\nabla f(x^i) + \sum_{l=1}^{k} e_l^i\right)$$

Now, in view of (23), Corollary 2.1 applies with $e^i = \sum_{l=1}^{k} e_l^i$ and the proof is complete.

∎

Remark 2.4. Theorem 2.2 can be easily generalized so that each processor takes an arbitrary but finite number of steps before any synchronization is made. The changes needed to extend Theorem 2.2 to these asynchronous methods are straightforward, and are thus omitted. See [12] for details.

Remark 2.5. A similar parallel version can be stated for Theorem 2.1.

3 CONVERGENCE OF THE BACKPROPAGATION ALGORITHM

We now turn our attention to the classical BP algorithm for training feedforward artificial neural networks with one layer of hidden units [16],[7],[17],[8]. The number of hidden units is assumed to be fixed.

Suppose we have N training examples and k processors with $N \geq 1$ and $k \geq 1$. In a manner similar to that of Section 2 we consider a partition of the set $\{1, \ldots, N\}$ into the subsets J_l, $l = 1, \ldots, k$, so that each example is assigned to at least one processor. Note that if examples are assigned to more than one processor, a different error function may be generated, which nevertheless measures the error of the same training problem. The variables of the problem here are the weights associated with the arcs of the neural network and the thresholds of the hidden and output units. The objective is to minimize a certain error function [11] which for our purposes here is the same as (18)–(19), that is

$$\min_{x \in \Re^n} f(x) := \sum_{l=1}^{k} f^l(x) = \sum_{l=1}^{k} \sum_{j \in J_l} f_j(x)$$

where $J_l \subseteq \{1, \ldots, N\}$, $l = 1, \ldots, k$, $\cup_{l=1}^{k} J_l = \{1, \ldots, N\}$. We note that this function is the sum of individual error functions each of which is associated with a single training example. Each component f_j of the objective function is a squared composition of the sigmoid and linear functions [11], and therefore satisfies the assumptions (12) on any bounded set.

Each iteration of the *serial online* BP algorithm consists of a step in the direction of negative gradient of an error function associated with a single training example. In the *parallel* BP each processor performs one (or more) cycles of serial BP on its set of training examples. Then a synchronization step is performed that consists of averaging the iterates computed by all the k processors.

Below we state a parallel BP algorithm with an added *momentum term* which consists of the difference between the current and previous iterates. For simplicity and in a similar manner to the method of conjugate gradients [13] we reset this term to zero periodically (see Algorithm 3.1). It has been observed that introduction of momentum term usually leads to faster convergence and adds stability to problems with noisy data [8].

We now summarize and describe our notation for stating and establishing convergence of the parallel BP algorithm with a momentum term :

$\boldsymbol{i = 1, 2, \ldots}$: Index number of major iterations of BP, each of which consists of going through the entire set of error functions $f_1(x), \ldots, f_N(x)$. This is achieved serially or in parallel by k processors with processor l handling the error function $f^l(x)$, $l = 1, \ldots, k$.

$\boldsymbol{j = 1, \ldots, N_l}$: Index of minor iterations performed by parallel processor l, $l = 1, \ldots, k$. Each minor iteration j consists of a step in the direction of negative gradient $-\nabla f^l_{m(j)}(z_l^{i,j})$ and a momentum step, where $m(j)$ is an element of the permuted set J_l. Note that in general the map $m(\cdot)$ depends on the index i and processor l. For simplicity, we skip this dependence in our notation. Recall that N_l is the number of elements in the set J_l.

$\boldsymbol{x^i}$: Iterate in $\Re^n$ of major iteration $i = 1, 2, \ldots$.

$\boldsymbol{z_l^{i,j}}$: Iterate in $\Re^n$ of minor iteration $j = 1, \ldots, N_l$, within major iteration $i = 1, 2, \ldots$, computed by processor $l = 1, \ldots, k$.

By $\boldsymbol{\eta_i}$ we shall denote the ***learning rate*** (the coefficient multiplying the gradient), and by $\boldsymbol{\alpha_i}$ the ***momentum rate*** (the coefficient multiplying the momentum term). For simplicity we shall assume that the learning and momentum rates remain fixed within each major iteration. Table 1 gives a flowchart of the parallel BP algorithm.

We are now ready to state and prove convergence of the parallel BP algorithm.

Algorithm 3.1. ***Parallel BP with Momentum Term.***
Start with any $x^0 \in \Re^n$. Having x^i, stop if $\nabla f(x^i) = 0$, else compute x^{i+1} as follows :
(i) for each processor $l \in \{1, \ldots, k\}$ do

$$z_l^{i,j+1} = z_l^{i,j} - \eta_i \nabla f^l_{m(j)}(z_l^{i,j}) + \alpha_i \Delta z_l^{i,j}, \quad j = 1, \ldots, N_l \tag{24}$$

$$\textit{where } z_l^{i,1} = x^i, \quad 0 < \eta_i < 1, \quad 0 \le \alpha_i < 1$$

$$\Delta z_l^{i,j} = \begin{cases} 0 & \textit{if } j = 1 \\ z_l^{i,j} - z_l^{i,j-1} & \textit{otherwise} \end{cases} \tag{25}$$

(ii) Synchronization

$$x^{i+1} = \frac{1}{k} \sum_{l=1}^{k} z_l^{i,N_l+1} \tag{26}$$

We note that for $k = 1$, Algorithm 3.1 becomes the serial BP, while the choice of $\alpha_i = 0$ reduces it to the simple BP. Note that since each parallel processor can use the same program for its computations, load balancing can be easily achieved.

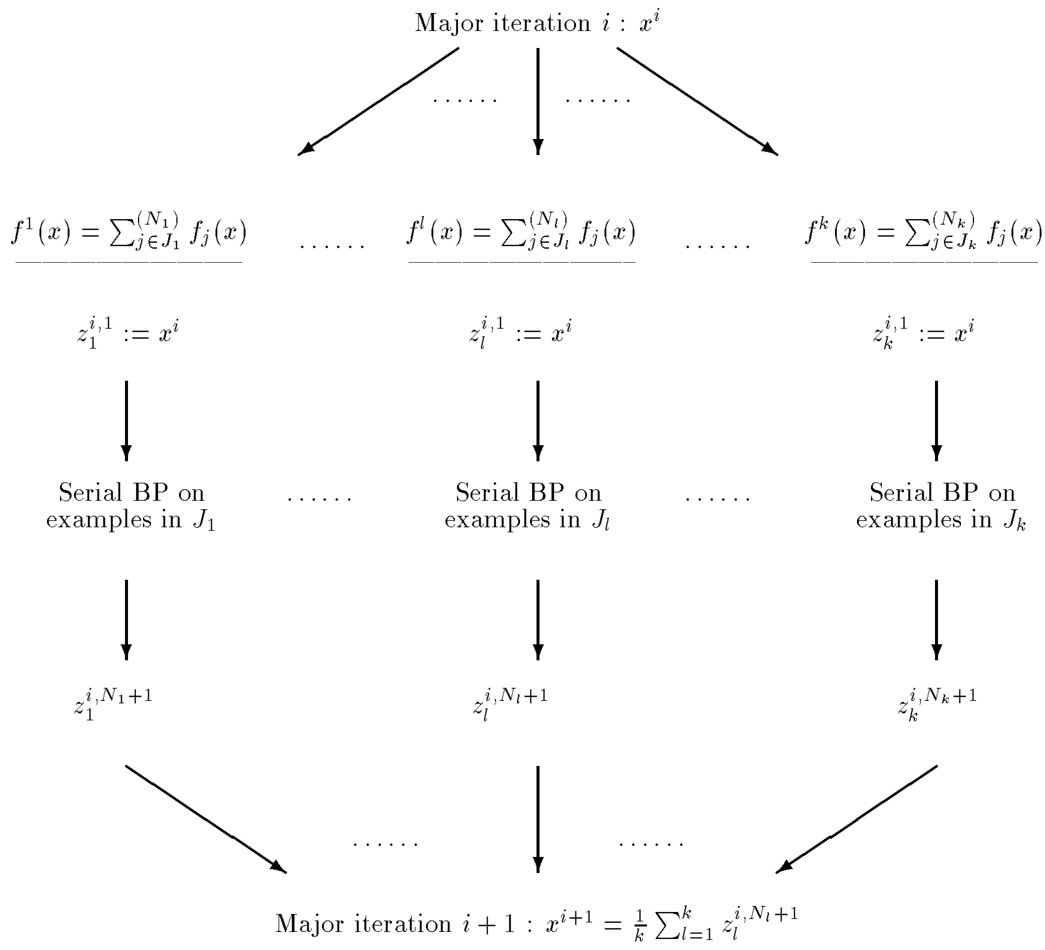

Table 1. Flowchart of the Parallel BP

To the best of our knowledge there are no published deterministic convergence proofs for either the parallel or serial BP algorithm. In [18] it is proven that the sequence of weights generated by the serial BP either converges to a point that is almost surely stationary or it diverges. In contrast, our approach is deterministic. Our proof which is based on the results of Section 2, covers both serial and parallel cases as well as the computationally important methods with a momentum term. We note the equivalence of BP to a deterministic perturbed gradient algorithm.

We are now ready to apply the analysis of Section 2 to backpropagation.

Theorem 3.1. *Let $S \subset \Re^n$ be any bounded set. If the learning and momentum* *rates are chosen such that*

$$\sum_{i=0}^{\infty} \eta_i = \infty, \ \ \sum_{i=0}^{\infty} \eta_i^2 < \infty, \ \ \sum_{i=0}^{\infty} \alpha_i \eta_i < \infty, \tag{27}$$

then for any sequence $\{x^i\} \subset S$ *generated by the BP Algorithm 3.1 it follows that* $\{f(x^i)\}$ *converges,* $\{\nabla f(x^i)\} \to 0$*, and for each accumulation point* $\bar{x}$ *of the sequence* $\{x^i\}$*,*
$\nabla f(\bar{x}) = 0$.

Proof. We shall show that the assumptions of Theorem 2.2 are satisfied.

First note that the error function is smooth and thus its gradient satisfies assumptions (12) on a bounded set S. Also f is nonnegative, hence it is bounded below. Using (24) and (25), for any cycle i, any processor l, and any j such that $2 \leq j \leq N_l + 1$ we obtain

$$\begin{aligned}
z_l^{i,j} - x^i &= z_l^{i,j} - z_l^{i,1} = \sum_{t=1}^{j-1} (z_l^{i,t+1} - z_l^{i,t}) \\
&= \sum_{t=1}^{j-1} (-\eta_i \nabla f_{m(t)}^l (z_l^{i,t}) + \alpha_i \Delta z_l^{i,t}) \\
&= -\eta_i \sum_{t=1}^{j-1} \nabla f_{m(t)}^l (z_l^{i,t}) + \alpha_i (z_l^{i,j-1} - x^i) \qquad (28) \\
&= -\eta_i \sum_{t=1}^{j-1} \nabla f_{m(t)}^l (z_l^{i,t}) - \eta_i \sum_{s=1}^{j-2} \left(\alpha_i^{j-1-s} \sum_{t=1}^{s} \nabla f_{m(t)}^l (z_l^{i,t}) \right) \qquad (29)
\end{aligned}$$

where (29) is obtained by repeated use of (28) with j replaced by $j-1,\ j-2, \ldots, 2$. By (28) and (18), for $j = N_l + 1$ we have

$$\begin{aligned}
z_l^{i,N_l+1} - x^i &= -\eta_i \sum_{t=1}^{N_l} \nabla f_{m(t)}^l (z_l^{i,t}) + \alpha_i (z_l^{i,N_l} - x^i) \\
&= -\eta_i (\nabla f^l (x^i) + a_l^i + \frac{\alpha_i}{\eta_i} b_l^i) \qquad (30)
\end{aligned}$$

where

$$a_l^i = \sum_{t=2}^{N_l} (\nabla f_{m(t)}^l (z_l^{i,t}) - \nabla f_{m(t)}^l (x^i)) \tag{31}$$

and

$$b_l^i = x^i - z_l^{i,N_l} \tag{32}$$

Let

$$e_l^i = -a_l^i - \frac{\alpha_i}{\eta_i} b_l^i \tag{33}$$

Now, in view of Theorem 2.2, assumptions (27) and (30), all we have to do is to verify that

$$\sum_{i=0}^{\infty} \eta_i \|e_l^i\| < \infty, \quad \|e_l^i\| \leq \gamma, \quad \gamma > 0, \ l = 1, \ldots, k \tag{34}$$

By (31), (29), (12), the triangle inequality and $\alpha_i \leq 1$ it follows

$$\begin{aligned}
\|a_l^i\| &\leq \sum_{t=2}^{N_l} \|\nabla f_{m(t)}^l(z_l^{i,t}) - \nabla f_{m(t)}^l(x^i)\| \leq L \sum_{t=2}^{N_l} \|z_l^{i,t} - x^i\| \\
&\leq L \sum_{t=2}^{N_l} \left(\eta_i \sum_{r=1}^{t-1} \|\nabla f_{m(r)}^l(z_l^{i,r})\| + \eta_i \sum_{s=1}^{t-2} \left(\alpha_i^{t-1-s} \sum_{r=1}^{s} \|\nabla f_{m(r)}^l(z_l^{i,r})\| \right) \right) \\
&\leq L\eta_i(N_l^2 M + N_l^3 M) = c_1 \eta_i
\end{aligned} \tag{35}$$

Similarly, by (32), (29), (12), the triangle inequality and $\alpha_i \leq 1$, we have

$$\begin{aligned}
\|b_l^i\| &= \|z_l^{i,N_l} - x^i\| \leq \eta_i \left(\sum_{t=1}^{N_l-1} \|\nabla f_{m(t)}^l(z_l^{i,t})\| + \sum_{s=1}^{N_l-2} \left(\alpha_i^{N_l-1-s} \sum_{t=1}^{s} \|\nabla f_{m(t)}^l(z_l^{i,t})\| \right) \right) \\
&\leq \eta_i \left(\sum_{t=1}^{N_l-1} M + \sum_{s=1}^{N_l-2} M N_l \right) \leq \eta_i (M N_l + M N_l^2) = c_2 \eta_i
\end{aligned} \tag{36}$$

By (33), (35), (36), and the triangle inequality, we obtain

$$\|e_l^i\| \leq c_1 \eta_i + c_2 \alpha_i$$

The latter combined with (27) implies (34), and the proof is complete.

∎

Remark 3.1. The boundedness assumption in Theorem 3.1 was made in order to ensure that f has Lipschitz continuous and bounded gradient. Since the principal result of the theorem deals with accumulation points, boundedness of the iterates is needed in order to ensure the existence of such accumulation points. There are a number of ways to ensure that the sequence of iterates produced by BP be bounded. In [6] a regularization term consisting of the squared 2-norm of x is added to the error function so that the modified objective function has bounded level sets. In [10] a simple projection onto a box is introduced which ensures that the iterates remain in the box.

4 CONCLUDING REMARKS

A general theorem for the nonmonotone convergence of a family of unconstrained optimization methods has been presented. It was established that the serial or parallel backpropagation algorithm with or without a momentum term for training feedforward artificial neural networks with one layer of hidden units can be viewed as a *deterministic* perturbed gradient method. Each accumulation point of the sequence of weights generated by BP is shown to be a stationary point of the error function associated with the given set of training examples. The results of the paper are equally applicable to feedforward neural networks with more than one layer of hidden units.

REFERENCES

1. C. C. Blaydon, R. L. Kashyap & K. S. Fu: "Applications of the stochastic approximation methods", in Adaptive, Learning, and Pattern Recognition systems, J. M. Mendel & K. S. Fu (eds), Academic Press, 1970.
2. Y. C. Cheng: "On the gradient projection method for solving the nonsymmetric linear complementarity problem", Journal of Optimization Theory and Applications 43, 1984, 527-541.
3. Yu. Ermoliev & R. J.-B. Wets (editors): "Numerical Techniques for Stochastic Optimization", Springer-Verlag, Berlin, 1988.
4. W. Finnoff: "Diffusion approximations for the constant learning rate backpropagation algorithm and resistance to local minima", 1992 Conference on Neural Information Processing Systems, Denver, Colorado, November 30–December 3, 1992.
5. A. A. Gaivoronski: "Convergence properties of backpropagation for neural networks via theory of stochastic gradient methods. Part 1.", Symposium on Parallel Optimization 3, Madison July 7-9, 1993.
6. L. Grippo: "A class of unconstrained minimization methods for neural network training", Symposium on Parallel Optimization 3, Madison July 7-9, 1993.
7. J. Hertz, A. Krogh & R. G. Palmer: "Introduction to the Theory of Neural Computation", Addison-Wesley, Redwood City, California, 1991.
8. T. Khanna: "Foundations of Neural Networks", Addison-Wesley, Reading, Massachusetts, 1989.
9. Z.-Q. Luo: "On the convergence of the LMS algorithm with adaptive learning rate for linear feedforward networks", Neural Computation 3, 1991, 226-245.
10. Z.-Q. Luo & P. Tseng: "Analysis of an approximate gradient projection method with applications to the backpropagation algorithm", Symposium on Parallel Optimization 3, Madison July 7-9, 1993.
11. O. L. Mangasarian: "Mathematical programming in neural networks", ORSA Journal on Computing, 5(4), 1993, 349-360.
12. O. L. Mangasarian: "Parallel gradient distribution in unconstrained optimization", Computer Sciences Department, University of Wisconsin, Madison, Technical Report # 1145, April 1993. SIAM Journal on Control, submitted.
13. G. P. McCormick & K. Ritter: "Alternate proofs of the convergence properties of the conjugate gradient method", Journal of Optimization Theory and Applications 13, 1974, 497-518.
14. J. M. Ortega: "Numerical Analysis: A Second Course", Academic Press, New York, New York 1972.
15. B. T. Polyak: "Introduction to Optimization", Optimization Software, Inc., New York, New York 1987.

16. D. E. Rumelhart, G. E. Hinton & R. J. Williams: "Learning internal representations by error propagation", in "Parallel Distributed Processing", D. E. Rumelhart, J. L. McClelland eds., MIT Press, Cambridge, 1986, 318-362.
17. P. K. Simpson: "Artificial Neural Systems", Pergamon Press, New York, 1990.
18. H. White: "Some asymptotic results for learning in single hidden-layer feedforward network models", Journal of the American Statistical Association, vol.84, No. 408, 1989, 1003-1013.