2

Conf-940699--2

On Stochastic Approximation Algorithms for Classes of PAC Learning Problems †

Nageswara S.V. Rao, V. R. R. Uppuluri‡, E. M. Oblow

Center for Engineering Systems Advanced Research

P. O. Box 2008

Oak Ridge National Laboratory

Oak Ridge, Tennessee 37831-6364

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Submitted to

IEEE World Congress on Computational Intelligence, June 26 - July 2, 1994, Orlando, Florida — federation of *IEEE International Conference on Neural Networks*, *IEEE International Conference on Fuzzy Systems*, *IEEE International Conference on Evolutionary Computation*.

"The submitted manuscript has been authored by a contractor of the U.S. Government under contract No. DE-AC05-84OR21400. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes."

†Research sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U.S. Department of Energy, under Contract No. DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc. In addition, this work is also partially funded by National Science Foundation under grant No IRI-9108610.

‡Consultant

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

872

On Stochastic Approximation Algorithms for Classes of PAC Learning Problems

Nageswara S.V. Rao, V. R. R. Uppuluri, E. M. Oblow

Center for Engineering Systems Advanced Research

P. O. Box 2008

Oak Ridge National Laboratory

Oak Ridge, Tennessee 37831-6364

Abstract— The classical stochastic approximation methods are shown to yield algorithms to solve several formulations of the PAC learning problem defined on the domain $[0,1]^d$. Under some assumptions on differentiability of the probability measure functions, simple algorithms to solve some PAC learning problems are proposed based on networks of non-polynomial units (e.g. artificial neural networks). Conditions on the sizes of the samples required to ensure the error bounds are derived using martingale inequalities.

Keywords— probably and approximately correct learning, stochastic approximation, neural networks, learning algorithms, martingales.

I. INTRODUCTION

THE paradigm of *Probably and Approximately Correct* (PAC) concept learning has attracted considerable attention over the past years since the pioneering works of Valiant [22]. Several basic results in PAC learning are existential in nature: informally, one of the main results states that any hypothesis that minimizes the empirical error, based on a sufficiently large but *finite* sample, will approximate the underlying concept with a high probability. Even for simple cases, the problem of computing such hypothesis could be of varying complexities [3,17]; thus, algorithms that can handle significant classes of PAC learning problems will be of both practical and theoretical interest. This paradigm is very useful in the study of practical learning algorithms which have to be terminated after a finite number of steps.

We show that some important classes of the PAC learning problem can be solved by stochastic approximation algorithms based on networks of sigmoidal units studied by Cybenko [5]. Thus, we illustrate an explicit connection between the area of PAC learning, feedforward neural networks (White [24]) and stochastic approximation (Wasan [25], Kushner and Clark [10], and Benveniste et al. [2]); the connection between the latter two has been well-known due to the works of Neljkovic [15], and Stankovic and Milosavljevic [21], Finoff [8], Darken and Moody [6] and White [25].

Incremental algorithms for learning problems are well-known in a deterministic framework in terms of classical perceptron algorithms (for example, Nilsson [16], Minski

and Papert [14]) and more recently in PAC learning context (for example, Littlestone [12], Rao et al. [19]). Our approach, under suitable conditions on the measure functions, subsumes several versions of [16,19], and moreover illustrates how several standard results on martingales can be used to design and analyze incremental learning algorithms in the PAC learning context. Compared to the on-line PAC algorithms (for example, Littlestone and Warmuth [13] and Vovk [23]) that attempt to minimize the total number of errors made during the iterations, our algorithm is weaker in that it provides bounds on the error committed after the algorithm has been terminated.

There have been a number of works that analyze learning algorithms based on neural networks [25]. In particular, the convergence properties of various back-propagation type algorithms has been shown by Neljkovic [15], and Stankovic and Milosavljevic [21], Finoff [8] and Darken and Moody [6] (also see White [25]). These works typically show the asymptotic convergence of the algorithm by utilizing an ordinary differential equation that characterizes the algorithm. It is unclear if these works can be used to give error bounds on the parameters of an algorithm that is terminated after a finite number of steps; such bounds are required in solving the PAC learning problem. Also, finite sample results are very useful in practical applications where the learning algorithms are always terminated after a finite number of steps. Further, the methods used here for obtaining finite sample results could be applied to the other learning algorithms.

We consider the concept classes that can be closely approximated by fixed-size networks, and classes of distributions that satisfy some conditions on the gradient (detailed conditions are given in Section 3). We present a basic algorithm that uses the (fixed-size) network representation of a concept, whose weights are incrementally updated in response to its predicted classification of the next example. We estimate bounds on the precision and accuracy of the current estimate as a function of sample size and/or number of iterations. Then we present a variant of this algorithm that uses "batches" of examples in each update; in some cases better bounds for this algorithm can be obtained compared to the basic algorithm.

There are two technical issues that need to be addressed in applying stochastic algorithms to PAC learning problems.

- (a) *Finite Representation of Hypotheses*: Stochastic approximation algorithms require that a hypothesis be

This research is sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U.S. Department of Energy, under Contract No. DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc. In addition, the work of the first author is also partially funded by National Science Foundation under grant No IRI-9108610.

V. R. R. Uppuluri is a consultant.

The submitted manuscript has been authored by a contractor of the U.S. Government under contract No. DE-AC05-84OR21400. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

finitely represented, typically by a real vector, in order to support the incremental update structure [2,24]. This issue is addressed by employing the recent results on approximation of several important classes of mappings by neural networks and related structures (e.g., see Cybenko [5], Leshno et al. [11]). For concreteness, we utilize the approximations of [5] to indicator functions (which are not continuous).

- (b) *Finite Sample Results*: PAC learning paradigm requires results based on finite samples, but explicit finite sample results are fairly infrequent in the area of stochastic algorithms. We employ finite sample results of Polyak [18] which are in a convenient form for this paper.

In terms of computational complexity, some of our algorithms have the sample sizes polynomial in the precision, confidence and other parameters; as a result of constant update time (with respect to sample size), the algorithms run in polynomial time. The corresponding class of PAC learning problem defines the concepts that are arbitrarily approximated by networks of fixed number of nodes. This class is interesting, in the light of results from Pitt and Valiant [17] that such guarantees cannot be given even in the case of Boolean problems (unless $R = NP$, where R is the class of sets accepted in random polynomial time by a deterministic Turing machine and NP is the class of sets accepted in polynomial time by a non-deterministic Turing machine).

In terms of sample complexity, as a function of the problem parameters (Lipschitz constant of the gradient and the size of the approximating network), the sample size for fixed precision and confidence can be made almost linear by a suitable choice of the step size in our algorithm (Corollary 2, Part (iib)); this aspect makes it very suitable for scale-ups in terms of these parameters.

The organization of the paper is as follows. Some preliminary facts on PAC learning, feedforward network approximation and stochastic approximation algorithms are presented in Section 2. The algorithms for PAC learning are presented in Section 3; the basic and batching algorithms are presented in Section 3.1 and 3.2 respectively.

II. PRELIMINARIES

A. PAC Learning

We are given a set $X = [0, 1]^d$ called the *domain*, and $C \subseteq 2^X$ and $H \subseteq 2^X$ called the *concept class* and *hypothesis class* respectively; members of C and H are measurable under a distribution P_X on X . A *concept* is any $c \in C$ and a *hypothesis* is any $h \in H$. For $s \subseteq X$, an *indicator function* $1_s : X \rightarrow \{0, 1\}$ is defined such that for any $x \in X$, $1_s(x) = 1$ (0) if and only if $x \in s$ ($x \notin s$). A pair $(x, 1_c(x))$ is an *example of* $c \in C$, and set of m such examples is *m-sample of* c . For $a, b \subseteq X$, we have $a \Delta b = (a \cap b) \cup (a \cap \bar{b})$. Then for any $s \in C \cup H \cup \{c \Delta h | c \in C, h \in H\}$ we have the *actual measure of* s given by $\mu(s) = \int dP_X$.

In the framework of *concept learning*, a finite-sized sample, produced according to an unknown distribution P_X

using an unknown concept $c \in C$, is given. The concept class C is said to be *learnable* [3] if given a finite sample, a hypothesis $h \in H$, $C \subseteq H$, can be produced such that for any $0 < \epsilon, \delta < 1$, we have the following condition satisfied

$$P[\mu(h \Delta c) \leq \epsilon] \geq 1 - \delta \quad (2.1.1)$$

where P is the product measure on the set of all independently and identically distributed m -samples. Note that $\mu(h \Delta c)$ is the probability that a randomly chosen test point $x \in X$ will be classified differently by h and c , i.e. $\mu(h \Delta c) = \int_{1_c(x) \neq 1_h(x)} dP_X$. Here, ϵ and δ are called the *precision* and *confidence* parameters respectively, and the pair (ϵ, δ) is called the *performance pair*. The condition of equation (2.1.1) is referred to as the (ϵ, δ) -condition.

B. Approximation by Networks

The general architecture of a multilayer feedforward network consists of an input layer with d units and output layer with m units, and one or more hidden layers; here we consider $m = 1$ and a single hidden layer. The hidden unit j has a weight vector $b_j \in \mathbb{R}^d$ and a threshold $t_j \in \mathbb{R}$. The output of the j th hidden unit is $\sigma(b_j^T x - t_j)$, where $x \in \mathbb{R}^d$ is the input vector, $b_j^T x$ denotes the scalar product, and $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is called an *activation function*. The output of the output node is given by

$$h(w, x) = \sum_{j=1}^M a_j \sigma(b_j^T x - t_j)$$

where $a = (a_1, a_2, \dots, a_M)$, $t = (t_1, t_2, \dots, t_M)$, M is the number of neurons in the hidden layer, and w is the *weight vector* of the network given by $w^T = [a^T t^T b_1^T \dots b_M^T]$. Then $h(w, x)$ is called the *output* of the network with the *weight vector* w .

Cybenko [5] considered the *sigmoid function* which is a specific form of continuous $\sigma(\cdot) : \mathbb{R} \rightarrow [0, 1]$ such that $\sigma(t) \rightarrow 1$ as $t \rightarrow +\infty$ and $\sigma(t) \rightarrow 0$ as $t \rightarrow -\infty$. The following theorem is a specialized version of Theorem 3 of [5] that is applicable to indicator functions.

Theorem 1 (5) *Let σ be a continuous sigmoidal function. Let f be an indicator function of any finite measurable set of $[0, 1]^d$. For any $\epsilon > 0$, for some finite M , there exists*

$$g(x) = h(w, x) = \sum_{j=1}^M a_j \sigma(b_j^T x - t_j)$$

and a set $D \subset [0, 1]^d$ with Lebesgue measure at least $1 - \epsilon$ and $|g(x) - f(x)| < \epsilon$ for all $x \in D$.

This theorem enables us to represent the indicator function of a concept $c \in C$ with an arbitrary precision as a network of sigmoidal functions of one hidden layer. The recent results of Leshno et al. [11] establish that it is sufficient for the activation function to be non-polynomial in order that the approximation results of the above kind are valid. For concreteness, we utilize the above theorem in our treatment, and the results of [11] can be similarly applied.

C. Stochastic Approximation Algorithms

Some of the simplest stochastic approximation algorithms take the following form

$$w_{n+1} = w_n - \gamma_n s_n(w_n, \zeta_n) \quad (2.3.1)$$

where the real vector w_n is an estimate of the parameter of interest at the n th step, $\{\gamma_n\}$ is a sequence of scalars, $\{\zeta_n\}$ is a sequence of random variables, and $s_n(w_n, \zeta_n)$ is a random variable called the *update rule*. For example, in solving $\min_w f(w)$, where gradient estimates of $f(\cdot)$ contain random error terms, $s_n(\cdot)$ could correspond to the noisy estimate of the gradient. The conditions for convergence (in different ways such as in probability, in expectation, etc.) of this type of algorithms have been extensively studied (e.g., Kushner and Clark [10], Wasan [24]).

The following results from Polyak [18] will be utilized in the next section.

Lemma 1 (18) For a sequence of real numbers $\{v_n\}$ such that $v_n \geq 0$, $0 \leq \nu_n \leq 1$, $0 \leq \eta_n \leq \eta$, and

$$v_{n+1} \leq (1 - \nu_n)v_n + \nu_n \eta_n,$$

we have

$$v_n \leq v_0 \prod_{j=0}^{n-1} (1 - \nu_j) + \eta \left(1 - \prod_{j=0}^{n-1} (1 - \nu_j) \right).$$

Corollary 1 (18) Under the hypothesis of Lemma 1, if $\nu_n \rightarrow 0$ and $\sum_{n=0}^{\infty} \nu_n = \infty$, then $v_n \rightarrow 0$.

Lemma 2 (18) Consider a sequence of real random variables $\{w_n\}$, $w_n \geq 0$, $E[w_0] < \infty$, and

$$E[w_{n+1} | w_0, w_1, \dots, w_n] \leq (1 - \nu_n)w_n + \nu_n \eta_n$$

$0 \leq \nu_n \leq 1$, $\sum_{n=0}^{\infty} \nu_n = \infty$, $\eta_n \geq 0$, $\eta_n \rightarrow 0$, $\lim_{n \rightarrow \infty} \bar{\rho}_n \geq \rho \geq 1$, $\bar{\rho}_n = \left(1 - \frac{\eta_{n+1}}{\eta_n}\right) \frac{1}{\nu_{n+1}}$. Then for every $\epsilon > 0$, there exists c such that

$$P \left[w_n \leq (c + \epsilon) \prod_{j=0}^{n-1} (1 - \nu_j) \text{ for all } n \right] \geq 1 - \frac{c}{\epsilon}.$$

In particular, if $\bar{\rho}_n \geq \rho > 1$ for all n , then $c = E[w_0] + \eta_0/(\rho - 1)$.

III. ALGORITHMS FOR PAC LEARNING

We first develop conditions required to implement a stochastic approximation algorithm for PAC learning problem.

Condition 1 Let $X = [0, 1]^d$ and for a fixed M , the set of functions of the form

$$h(w, x) = \sum_{j=1}^M a_j \sigma(b_j^T x - t_j)$$

approximate the set of indicator functions $\{1_c(\cdot)\}_{c \in C}$ of the concept class C in the sense of Theorem 2.1, where w is the weight vector of known dimension $N = (d + 2)M$, i.e., for each $1_c(\cdot)$ and $\epsilon > 0$, there exists some $h(w, \cdot)$ such that $|1_c(x) - h(w, x)| < \epsilon$ for all $x \in D \subseteq X$ such that the Lebesgue measure of D is at least $1 - \epsilon$.

Notice that this condition requires that the parameter vector be of fixed and known dimension N . If N is not known, for the sigmoidal activation function, estimates of N can be obtained by using the Fourier coefficients of the indicator functions using the works of Barron [1].

A. Basic Algorithm

The structure of the *basic algorithm* is as follows

$$w_{n+1} = w_n - \Gamma_n [h(w_n, x_n) - 1_c(x_n)] \quad (3.1.1)$$

where $w_n, w_{n+1} \in \mathbb{R}^N$, each component of $\Gamma_n \in \mathbb{R}^N$ consists of a scalar γ_n (same in all components of Γ_n) called the *step size*, and $(x_n, 1_c(x_n))$ is the n th example. The expression $|h(w_n, x_n) - 1_c(x_n)|$ is the *update rule*. Each step of this algorithm can be implemented in $O(Md) = O(N)$ time, and thus this algorithm has a time complexity of $O(nMd)$, when terminated after n th iteration.

A number of algorithms similar to (3.1.1) have been studied by Nedeljkovic [15], Finnoff [8], Darken and Moody [6], and Stankovic and Milosavljevic [21] (also see Fargo and Lugosi [7] for more complex algorithms). Only asymptotic analysis of these algorithms has been carried out, and it is unclear as how to guarantee PAC style results based on these works.

Let $\mu(w) = \int |h(w, X) - 1_c(X)| dP_X$ denote the expected error made by the hypothesis $h(w, \cdot)$. Here we choose to provide (ϵ, δ) -condition for $\mu(w)$, and note that if $h(w, \cdot)$ takes values of 0 or 1 only, then this (ϵ, δ) -condition is identical to the condition (2.1.1). Also we have $\mu(w) \geq 0$, and $\inf_w \mu(w) = 0$ by Condition 3.1. Further $E[|h(w_n, x_n) - 1_c(x_n)|] = \mu(w_n)$.

In order to ensure the convergence of the algorithm (3.1.1), we require the following conditions on the probability measure generated by P_X .

Condition 2 Let $\mu(w)$ be differentiable and its gradient satisfy the following Lipschitz condition. For all $u, v \in \mathbb{R}^N$, there exists a positive constant L such that

$$\|\nabla \mu(u) - \nabla \mu(v)\| \leq L \|u - v\|.$$

Condition 3 There exists a scalar θ such that for any w and N -dimensional vector $\Psi(w) = (\mu(w), \dots, \mu(w))^T$, we have

$$\nabla \mu(w)^T \Psi(w) \geq \theta \mu(w),$$

where $\nabla \mu(w)^T = \left(\frac{\partial \mu(w)}{\partial w_1}, \dots, \frac{\partial \mu(w)}{\partial w_N} \right)$. This condition implies $\nabla \mu(w)^T \mathbf{1} \geq \theta$, every where except at $\mu(w) = 0$, where $\mathbf{1}$ is a column vector of all 1's.

We now state a basic version of our main result. Here the conditions on the sample sizes are expressed as functions of the step size γ_n , and two specific cases are then illustrated in the following corollary.

Theorem 2 Under the Conditions 1-3, the basic algorithm of (3.1.1) satisfies the following conditions:

- (i) (a) Under the condition $\sum_{j=0}^{\infty} \gamma_j^2 < \infty$, the sequence of random variables $\{\mu(w_n)\}$ converges to a finite random variable $\mu(w)$.
- (b) Under the conditions $\gamma_n \rightarrow 0$ as $n \rightarrow \infty$ and $\sum_{i=1}^{\infty} \gamma_i = \infty$, we have $E[\mu(w_n)] \rightarrow 0$ as $n \rightarrow \infty$.
- (c) In addition to conditions in (b), under the condition $\lim_{n \rightarrow \infty} \frac{1}{\theta} \left(\frac{1}{\gamma_{n+1}} - \frac{1}{\gamma_n} \right) \geq \gamma \geq 1$, we have $\mu(w_n) \rightarrow 0$ with probability one.
- (ii) For $\gamma_n < 1/\theta$ for all n , we have $P[\mu(w_n) < \epsilon] > 1 - \delta$ for sufficiently large sample of size n such that

$$\prod_{j=0}^{n-1} (1 - \gamma_j \theta) = \frac{\delta \epsilon - \frac{LN}{2\theta^2}}{1 - \frac{LN}{2\theta^2}}.$$

The required sample size n is also given by the condition $\sum_{j=0}^{n-1} \gamma_j = \frac{1}{\theta} \ln \left(\frac{1 - \frac{LN}{2\theta^2}}{\delta \epsilon - \frac{LN}{2\theta^2}} \right)$.

- (iii) Under the conditions, $\gamma_n \leq 1/\theta$ for all n , $\gamma_n \rightarrow 0$, $\sum_{j=0}^{\infty} \gamma_j = \infty$, and $1/\gamma_{n+1} - 1/\gamma_n \geq \rho \theta \geq \theta$ we have $P[\mu(w_n) < \epsilon \text{ for all } n] > 1 - \delta$ for sufficiently large sample of size n such that

$$\sum_{j=0}^{n-1} \gamma_j = \ln \left[\frac{(1 + \delta)}{\epsilon \delta} \left(1 + \frac{LN \gamma_0}{2\theta} \right) \right].$$

Proof: Using Taylor's series expansion (see [24]) and Condition 3.2 we obtain for $s_n = |h(w_n, x_n) - 1_c(x_n)|$, and $\xi_n = (s_n, \dots, s_n)^T \in [0, 1]^N$,

$$\begin{aligned} \mu(w_{n+1}) &= \mu(w_n - \Gamma_n s_n) = \mu(w_n - \gamma_n \xi_n) \\ &\leq \mu(w_n) - \gamma_n \nabla \mu(w_n)^T \xi_n + \frac{L}{2} \gamma_n^2 \|\xi_n\|^2 \end{aligned} \quad (3.2.1)$$

Since $\nabla \mu(w_n)^T \xi_n = s_n \nabla \mu(w_n)^T \mathbf{1}$, by Condition 3.3 we have

$$E[\nabla \mu(w_n)^T \xi_n] = \mu(w_n) \nabla \mu(w_n)^T \mathbf{1} \geq \theta \mu(w_n)$$

Also since $s_n \leq 1$, we have $\|\xi_n\|^2 \leq N$. Let $\mathcal{F}_n$ be the σ -algebra generated by $w_1, w_2, \dots, w_n$. Now taking conditional expectations on both sides of the equation (3.2.1) we obtain the following condition:

$$E[\mu(w_{n+1}) | \mathcal{F}_n] \leq (1 - \gamma_n \theta) \mu(w_n) + \frac{LN \gamma_n^2}{2} \quad (3.1.2)$$

Under the conditions $\gamma_n < 1/\theta$ and $\sum_{j=1}^{\infty} \gamma_j^2 < \infty$, the sequence $\{\mu(w_n)\}$ forms an almost supermartingale, hence $\mu(w_n) \rightarrow \mu_w < \infty$ almost surely by the lemma of Robbins and Siegmund [20]; differentiability of measures then implies $\mu_w = \mu(w)$ for some w . The Part (ib) follows from

Corollary 1 as shown below, and Part (ic) is a direct consequence of Lemma 2. Hence Part (i) is true.

For parts (ii) and (iii), we obtain a bound on $E[\mu(w_n)]$, and use Chebychev's inequality that

$$P[\mu(w_n) > \epsilon] \leq \frac{E[\mu(w_n)]}{\epsilon}$$

; by equating the right hand side or its suitable upperbound to δ , we obtain a condition on n .

We take the expectation on both sides of the equation (3.1.2) to obtain

$$E[\mu(w_{n+1})] \leq (1 - \gamma_n \theta) E[\mu(w_n)] + \gamma_n \theta \frac{LN \gamma_n}{2\theta}.$$

Now Part (ib) is a direct consequence of Corollary 1. First bound of Part (ii) follows directly from Lemma 1. For the second bound, we note that $e^{-z} \geq 1 - z$ for $0 \leq z \leq 1$

which yields the upper bound $e^{-\sum_{j=0}^{n-1} \gamma_j} \geq \prod_{j=0}^{n-1} (1 - \theta \gamma_j)$.

Part (iii) is a direct consequence of Lemma 2 by noting

that $\delta = c/\epsilon$ and $\epsilon = (c + \epsilon)e^{-\sum_{j=0}^{n-1} \nu_j}$, and the event

$$\left[w_n \leq (c + \epsilon) \prod_{j=0}^{n-1} (1 - \nu_j) \right]$$

implies the event

$$\left[w_n \leq (c + \epsilon) e^{-\sum_{j=0}^{n-1} \nu_j} \right].$$

Thus the theorem is true. $\square$

Part (i) of this theorem characterizes the asymptotic behavior of the algorithm in (3.1.1); the condition (ib) corresponds to the well-known Robbins-Monro conditions on the step size. Part (ii) and (iii) yield the sample sizes needed to ensure (ϵ, δ) -condition at the n step of the algorithm. The condition in (ii) is just on $\mu(w_n)$ at the n th step, and this condition might not have been satisfied before n th step, whereas the condition in (iii) is valid uniformly for all n , and hence is stronger than (ii).

The following corollary discusses two important special cases where the step size is a constant and a constant multiple of the components of Harmonic numbers respectively.

Corollary 2 Under the Conditions 1-3, the basic algorithm of (3.1.1) satisfies the following conditions:

- (i) Under the condition of constant step size $\gamma_n = \gamma$, $\gamma < 1/\theta$, we have $P[\mu(w_n) < \epsilon] > 1 - \delta$ for sufficiently large sample of size n given by

$$n = \ln \left(\frac{1 - \frac{LN \gamma}{2\theta}}{\delta \epsilon - \frac{LN \gamma}{2\theta}} \right) / \ln \left(\frac{1}{1 - \gamma \theta} \right).$$

- (ii) Consider $\gamma_n = \gamma/(n + 1)$, for $n \geq 0$, and $\gamma < 1/\theta$.

- (a) We have $P[\mu(w_n) < \epsilon] > 1 - \delta$ for sufficiently large sample of size n given by

$$n = \left\lceil \frac{1 - \frac{L\gamma N}{2\theta}}{\epsilon\delta - \frac{L\gamma N}{2\theta}} \right\rceil^{\frac{2}{\theta}}$$

- (b) We have $P[\mu(w_n) < \epsilon \text{ for all } n] > 1 - \delta$ for sufficiently large sample of size n given by

$$n = \left\lceil \frac{(1 + \delta)}{\epsilon\delta} \left(1 + \frac{L\gamma^2 N}{2(1 - \gamma\theta)} \right) \right\rceil^{\frac{1}{\theta}}$$

Proof: Part (i) follows along the lines of Part (ii) of Theorem 2 by using $\gamma_n = \gamma$. Now consider Part (ii). First $\eta_n = \frac{L\gamma N}{2\theta(n+1)}$, and $\eta = \frac{L\gamma N}{2\theta}$. By following along the lines of Part (ii) of Theorem 2, we obtain

$$E[\mu(w_n)] \leq (1 - \eta) \prod_{i=0}^{n-1} (1 - \gamma_i \theta) + \eta.$$

Now $\sum_{i=0}^{n-1} \gamma_i \geq 1/2 \ln 2$ (since $H_n = 1 + \frac{1}{2} + \dots + \frac{1}{n} > 1/2(\ln n + 1)$ [7]) which yields

$$\prod_{i=0}^{n-1} (1 - \gamma_i \theta) \leq e^{-\sum_{i=0}^{n-1} \theta \gamma_i} \leq n^{-\theta \gamma/2}.$$

Thus $E[\mu(w_n)] \leq (1 - \eta)n^{-\theta \gamma/2} + \eta$, and the condition on n follows by equating the right hand side to $\epsilon\delta$. Second equation of Part (ii) similarly follows from Lemma 2 by noting that $\bar{\rho}_n = \rho = \frac{1}{\theta\gamma}$. $\square$

Notice that the constant step algorithm does not satisfy either conditions of Part (i) of Theorem 2. The step size $\gamma_n = \gamma/(n+1)$, satisfies all three sets of conditions in Part (i) of Theorem 3.1; first, the sum $\gamma \sum_{j=1}^{\infty} \frac{1}{j^2}$ is the Riemann's zeta function $\zeta(2)$ which converges to a limit, and second, the Harmonic number $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ satisfies the condition $\frac{\ln n + 1}{2} < H_n \leq \lfloor \ln n \rfloor + 1$ which implies $\sum_{j=1}^{\infty} \gamma_j$ is unbounded (more information on Harmonic numbers can be found in Graham et al. [7]). In terms of the scale-up with respect to L and N , by choosing $\frac{1}{\theta(1+\alpha)} < \gamma < \frac{1}{\theta}$ for small $\alpha > 0$, by Part (iib) we have $n = O(L^{1+\alpha} N^{1+\alpha})$ for fixed ϵ , δ and θ ; this almost linear nature makes this algorithm scalable in terms of N and L .

B. Batching Algorithm

We now allow the update rule to depend on n ; unlike in (3.1.1) where w_n is updated with every example, we "batch" x_i 's before updating w_n as follows.

$$w_{n+1} = w_n - \gamma_n \sum_{j=1}^{i_n} |h(w_n, x_{t_{n-1}+j}) - l_c(x_{t_{n-1}+j})| \quad (3.2.1)$$

where i_n is a positive integer, and $(x_{t_{n-1}+1}, l_c(x_{t_{n-1}+1})), (x_{t_{n-1}+2}, l_c(x_{t_{n-1}+2})), \dots, (x_{t_{n-1}+n}, l_c(x_{t_{n-1}+n}))$ is the set

of examples of n th step, disjoint from the set of any other step such that $t_n = t_{n-1} + i_n$. This algorithm will be referred to as the *batching algorithm* and its performance is described in the following theorem.

Theorem 3 Under the Conditions 1-3, the batching algorithm of (3.2.1) satisfies the following conditions:

- (i) If $1/(\theta+1) \leq \gamma_n i_n \leq 1/\theta$ for all n , we have $P[\mu(w_n) < \epsilon] > 1 - \delta$ for sufficiently large number of steps n given by

$$n = \ln \left(\frac{1 - \frac{L\gamma N}{2\theta^2}}{\epsilon\delta - \frac{L\gamma N}{2\theta^2}} \right) / \ln(1 + 1/\theta).$$

- (ii) If $\gamma_n = \theta/i_n$ for all n , we have $P[\mu(w_n) < \epsilon] > 1 - \delta$ for sufficiently large number of steps n given by

$$n = \frac{1}{\theta^2} \ln \left(\frac{1 - \frac{L\gamma N}{2}}{\epsilon\delta - \frac{L\gamma N}{2}} \right).$$

- (iii) Consider a constant a such that $a > 1$ if $\theta > 1$ and $a < 1$ if $\theta < 1$. For $\gamma_n i_n = \frac{1}{\theta \cdot (n+1)^a}$, we have $P[\mu(w_n) < \epsilon] > 1 - \delta$ for sufficiently large number of steps n given by

$$n = \left\lceil \frac{(1 + \delta)}{\epsilon\delta} \left(1 + \frac{L\gamma N}{2\theta^{a+1}(\theta^{a-1} - 1)} \right) \right\rceil^{2\theta^{a-1}}.$$

Proof: The proof follows along the lines of Theorem 2.

Let $s_n = \sum_{j=1}^{i_n} |h(w_n, x_{t_{n-1}+j}) - l_c(x_{t_{n-1}+j})|$. Then we have $E[s_n \nabla \mu(w_n)^T \mathbf{1}] \geq \theta i_n \gamma_n$, and the equation (3.1.2) for this case can be written as follows

$$E[\mu(w_{n+1})|w_n] \leq (1 - \gamma_n i_n \theta) \mu(w_n) + \frac{L\gamma_n^2 i_n^2}{2}.$$

Then we apply the lemmas of Section IIB. For Part (i) we note that $\gamma_n i_n \leq 1/\theta$ which yields $\eta = \frac{L\gamma N}{2\theta^2}$, and $\prod_{j=0}^{n-1} (1 - \gamma_j \theta_j) \leq \left(\frac{\theta}{\theta+1} \right)^n$, and the required bound follows. For Part

(ii), we have $\eta = \frac{L\gamma N}{2}$ and $\prod_{j=0}^{n-1} (1 - \gamma_j \theta_j) \leq (1 - \theta^2)^n$ which yields the required bound. Part (iii) is based on Lemma 2. The choice of a based on θ ensures that $\bar{\rho}_n = \theta^{a-1} > 1$. Then we have $\sum_{j=0}^{n-1} \gamma_j i_j \theta \geq \theta^{1-a} \sum_{j=0}^{n-1} \frac{1}{j+1} \geq \frac{1}{2} \theta^{1-a} \ln n$. This

in turn implies $e^{-\sum_{j=0}^{n-1} \gamma_j i_j \theta} \leq n^{-\frac{1}{2} \theta^{1-a}}$. $\square$

In Theorem 3, extra information on θ is used. The bounds on the number of steps and/or sample size here could be better, in some cases, than those in Theorem 2. We illustrate some specific cases. Consider that L is small so that $\frac{1 - \frac{L\gamma N}{2\theta^2}}{\epsilon\delta - \frac{L\gamma N}{2\theta^2}}$ can be approximated by $\frac{1}{\epsilon\delta}$. Then the number of steps in Theorem 2 (i) no more than those of constant step algorithm (of Corollary 2(i)) under the condition $\gamma \leq \frac{1}{\theta(\theta+1)}$. Under the same condition on L , we can compare the number of steps in Theorem 3 (ii) with

those of Corollary 2 (iia); it can be shown that the former is smaller under the condition $\theta > \gamma/2$.

Now consider the sample size. If $i_j = c$, c is a constant, then by Theorem 3 (i) the sample size is cn ; under the condition of small L above, we can show that this sample size is upperbounded by n of Corollary 1(i) under the condition $\gamma \leq \frac{1}{\theta} \left(1 - \frac{\epsilon}{\log_c(1+1/\theta)}\right)$. For the case $i_j = j$, similar computation yields the condition

$$\gamma \leq \frac{1}{\theta} \left[1 - (1 + 1/\theta)^{2 \ln(1+1/\theta) / \ln(1/(\delta\epsilon))}\right].$$

By utilizing the specific values of N , L and θ , similar improvements are possible in certain other cases by using suitable sequences $\{i_n\}$ and $\{\gamma_n\}$.

IV. CONCLUSIONS

We showed that the well-known techniques of stochastic approximation algorithms can be used to solve some classes of PAC learning problems. We considered the concept classes that can be closely approximated by a fixed-size networks, and classes of distributions that satisfy some conditions on the gradient. We presented a basic algorithm that uses a network representation of a concept, whose weights are updated in response to the next example. Then we presented a variant of this algorithm that uses "batches" of examples in each update; in some cases better bounds for this algorithm are obtained compared to the basic algorithm.

This work can be viewed as only a *starting point* for a vast number of conceivable future investigations in a number of ways. First, the extensive varieties of stochastic approximation algorithms can yield a number of variations of the proposed basic algorithm. Second, approximations based on radial basis networks (Broomhead and Lowe [4]), wavelet networks (Zhang and Benveniste [26]), etc. could provide some more useful versions of the basic algorithm. Also, the PAC paradigm could possibly motivate finite (small) sample studies of stochastic approximation algorithms.

REFERENCES

- [1] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):931-945, 1993.
- [2] A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations*. Springer-Verlag, New York, 1990.
- [3] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association of Computing Machinery*, 36(4):929-965, 1989.
- [4] D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321-355, 1988.
- [5] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303-314, 1989.
- [6] C. Darken and J. Moody. Towards faster stochastic gradient search. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances In Neural Information Processing Systems*, volume 4, pages 1010-1016. Morgan Kaufmann Publishers, San Mateo, California, 1992.

- [7] A. Farago and G. Lugosi. Strong universal consistency of neural network classifier. *IEEE Transactions on Information Theory*, 39(4):1146-1151, 1993.
- [8] W. Finnoff. Diffusion approximations for the constant learning rate backpropagation algorithm and resistance to local minima. In S. T. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances In Neural Information Processing Systems*, volume 5, pages 459-466. Morgan Kaufmann Publishers, San Mateo, California, 1993.
- [9] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley Pub. Co., New York, 1988.
- [10] H. J. Kushner and D. S. Clark. *Stochastic Approximations for Constrained and Unconstrained systems*. Springer-Verlag, New York, 1979.
- [11] M. Leshno, V. Ya. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6:861-867, 1993.
- [12] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285-318, 1988.
- [13] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. In *Proceedings of Symp. on Foundations of Computer Science*, pages 256-261, 1989.
- [14] M. L. Minsky and S. A. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1988. Second Printing.
- [15] V. Nedeljkovic. A novel multilayer neural networks training algorithm that minimizes the probability of classification error. *IEEE Transactions on Neural Networks*, 4(4):650-659, 1993.
- [16] N. J. Nilsson. *Learning Machines*. McGraw-Hill Book Co., 1965.
- [17] L. Pitt and L. G. Valiant. Computational limitations on learning from examples. *Journal of the Association for Computing Machinery*, 35(4):965-984, 1988.
- [18] B. T. Polyak. Convergence and convergence rate of iterative stochastic algorithms I: General case. *Automation and Remote Control*, (12):1858-1868, 1976.
- [19] N. S. V. Rao, E. M. Oblow, and C. W. Glover. Learning separations by Boolean combinations of hyperplanes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994. to appear.
- [20] H. Robbins and D. Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In J. S. Rustagi, editor, *Optimizing Methods in Statistics*, pages 233-258. Academic Press, New York, 1971.
- [21] S. Stankovic and M. Milosavljevic. Training of multilayer perceptrons by stochastic approximation. In V. Milutinovic and P. Antognelli, editors, *Neural Networks, Concepts, Applications and Implementations*, volume IV, pages 201-239. Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [22] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134-1142, 1984.
- [23] V. G. Vovk. Aggregating strategies. In *Proceedings of Third Ann. Workshop on Computational Learning Theory*, pages 371-383, 1990.
- [24] M. T. Wasan. *Stochastic Approximation*. Cambridge University Press, 1969.
- [25] H. White. *Artificial Neural Networks*. Blackwell Publishers, Cambridge, Massachusetts, 1992.
- [26] Q. Zhang and A. Benveniste. Wavelet networks. *IEEE Transactions on Neural Networks*, 3(6):889-898, 1992.

DATE

FILMED

5/4/94

END