Revised and expanded version, August 1992.
Original appears in *Neural Networks for Signal Processing 2 — Proceedings of the 1992 IEEE Workshop*, IEEE Press, 445 Hoes Lane, Piscataway, NJ 08854.

LEARNING RATE SCHEDULES FOR FASTER STOCHASTIC GRADIENT SEARCH

Christian Darken*, Joseph Chang ‡ and John Moody*[1]
Yale Departments of Computer Science* and Statistics‡
P.O. Box 2158, New Haven, CT 06520

Abstract. Stochastic gradient descent is a general algorithm that includes LMS, on-line backpropagation, and adaptive k-means clustering as special cases. The standard choices of the learning rate η (both adaptive and fixed functions of time) often perform quite poorly. In contrast, our recently proposed class of "search then converge" (STC) learning rate schedules (Darken and Moody, 1990b, 1991) display the theoretically optimal asymptotic convergence rate and a superior ability to escape from poor local minima However, the user is responsible for setting a key parameter. We propose here a new methodology for creating the first automatically adapting learning rates that achieve the optimal rate of convergence.

INTRODUCTION

The stochastic gradient descent algorithm is

$$\Delta W(t) = -\eta \nabla_W f[W(t), X(t)], \tag{1}$$

where η is the learning rate, t is the "time", and $X(t)$ is the independent random exemplar chosen at time t. The purpose of the algorithm is to find a parameter vector W that minimizes a function $g(W)$, which for learning algorithms has the form $E_X f(W, X)$, i.e. g is the average of an objective function f over the exemplars X. We can rewrite $\Delta W(t)$ in terms of g as

$$\Delta W(t) = -\eta\{\nabla_W g[W(t)] + \xi[t, W(t), X(t)]\}, \tag{2}$$

where the ξ are independent zero-mean random variables ("noise"). Stochastic gradient descent may be preferable to deterministic gradient descent when the exemplar

[1]New addresses effective September 1992: C. Darken, Siemens Corporate Research, 755 College Rd. East, Princeton, NJ 08540; J. Moody, Computer Science and Engineering (CSE) Dept., Oregon Graduate Institute of Science and Technology (OGI), 19600 NW von Neumann Dr., Beaverton, OR 97006-1999. Email addresses: darken@learning.siemens.com, chang@stat.yale.edu, moody@cse.ogi.edu.

set is increasing in size over time or large, making the average over exemplars expensive to compute. Additionally, the noise in the gradient can help the system escape from local minima (Darken and Moody, 1990a, 1990b, 1991). The fundamental algorithmic issue is **how best to adjust η as a function of time and the exemplars**? Our primary goal is to develop η's that cause W to converge quickly to a minimum despite the presence of the noise.

THE OPTIMAL RATE OF CONVERGENCE

"Optimal" is a tricky word, and usually requires explanation. Let w^* be the minimum to which we are converging. Define the "misadjustment" as $M(t) := |W(t) - w^*|^2$, i.e. the squared euclidean distance to the minimum. Near w^*, multiples of this quantity bound the usual sum of squares error measure above and below, so the sum of squares error is roughly proportional to the misadjustment. The implication of a range of theoretical investigations (Chung, 1954; Fabian, 1968; Major and Revesz, 1973; Goldstein, 1987) is that the fastest rate that the misadjustment may be reduced with any η that is a function of time only is $M(t) \propto t^{-1}$. Very little is known theoretically about η's which are allowed to depend upon current or past values of the parameters or exemplars (see however Zhulenev and Medovyi 1978), but experiments indicate that sustained rates of convergence faster than the above are not generally possible. Thus, when we speak of *optimally fast convergence*, we mean that the misadjustment is going to zero proportional to t^{-1}.

STANDARD LEARNING RATE SCHEDULES

The usual non-adaptive choices of η (i.e. η's depending on the time only) often yield poor performance.

The simple expedient of taking η to be constant (the typical choice for the backpropagation and LMS algorithms) results in persistent residual fluctuations. The magnitude of such fluctuations and the resulting degradation of system performance are difficult to anticipate (see fig. 1). Choosing a small value of η reduces the magnitude of the fluctuations, but seriously slows convergence and causes problems with metastable local minima, while choosing a large value of η can result in instability.

Taking $\eta(t) = c/t$ (the usual choice in the stochastic approximation literature for the last forty years beginning with seminal work Robbins and Monro (1951)), typically results in slow convergence to bad solutions (high-lying local minima) for small c, and parameter blow-up for small t if c is too large (Darken and Moody, 1990b, 1991).

The available adaptive schedules (i.e. η's depending on the time *and* on previous exemplars) have problems as well. A schedule developed by Urasiev is proven to converge in principle, but in practice it may converge slowly if at all (see fig. 2). The delta-bar-delta learning rule, which was developed in the context of deterministic gradient descent (Jacobs, 1988), is often useful in locating the general vicinity of a solution in the stochastic setting. However, it hovers about the solution without converging (see fig. 3). Methods such as Kesten's (1958) require the user to specify an entire sequence of free parameters and thus are difficult to compare to alternative techniques. However, with a specific, reasonable choice of sequence, Zhulenev and Medovyi (1978) have proven that the optimal convergence rate is not generally achieved.

The literature for both non-adaptive and adaptive learning rates is widely scattered over time and disciplines, however to our knowledge, no published technique is guaranteed to attain the optimal convergence speed.

Also available are online nongradient (e.g. pseudo-newton) techniques for solving optimization problems (Ljung and Söderström, 1983, etc.). While these techniques may be much more efficient asymptotically than gradient techniques, they require a minimum of $O(N^2)$ operations for each recursive update, where N is the number of trainable parameters ("weights") in the system, as compared to $O(N)$ for the stochastic gradient techniques above. Since a neural network may have a very large number of parameters (thousands to hundreds of thousands of parameters are not uncommon), $O(N^2)$ updates may be too slow for general use. Worse, it is unclear that the extra computation is helpful for nonquadratic $g(W)$ when far from a minimum.

QUALITATIVE BEHAVIOR OF SCHEDULES

We compare several fixed and adaptive learning rate schedules on a toy problem. The problem is learning a two parameter adaline (gain=1.0, bias=1.0) in the presence of independent uniformly distributed $[-0.5, 0.5]$ noise on the exemplar labels. The exemplars are independently sampled and uniformly distributed on $[-1.8, 0.2]$. The objective function has a condition number of 10, indicating the presence of the ravine indicated by the elliptical contours in the figures. $c^* = 5$ for this problem. All runs start from the same parameter (weight) vector and receive the same sequence of exemplars. Results are presented in figs. 1–6.

SEARCH-THEN-CONVERGE SCHEDULES

Our recently proposed solution to the problems of escaping from "bad" metastable local minima, finding a "good" local minimum, and achieving the asymptotically optimal rate of convergence are the "search then converge" (STC) learning rate schedules (Darken and Moody, 1990b, 1991). With the STC schedules, η is chosen to be a fixed function of time, such as the following:

$$\eta(t) = \eta_0 \frac{1 + \frac{c}{\eta_0}\frac{t}{\tau}}{1 + \frac{c}{\eta_0}\frac{t}{\tau} + \tau\frac{t^2}{\tau^2}} \tag{3}$$

This function is approximately constant with value η_0 at times small compared to τ (the "search phase"). At times large compared with τ (the "converge phase"), the function decreases as c/t. See for example the eta vs. time curves for figs. 4 and 5. This schedule has demonstrated a dramatic improvement in convergence speed and quality of solution as compared to the traditional learning rate schedules for adaptive k-means clustering, an unsupervised learning algorithm (Darken and Moody,

Constant η=0.1

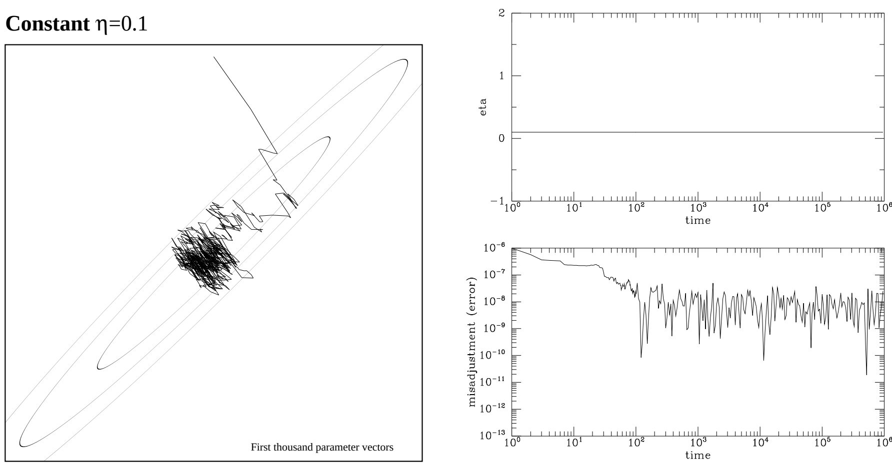

Figure 1: The constant η schedule, commonly used in training backpropagation networks, does not converge in the stochastic setting.

Urasiev

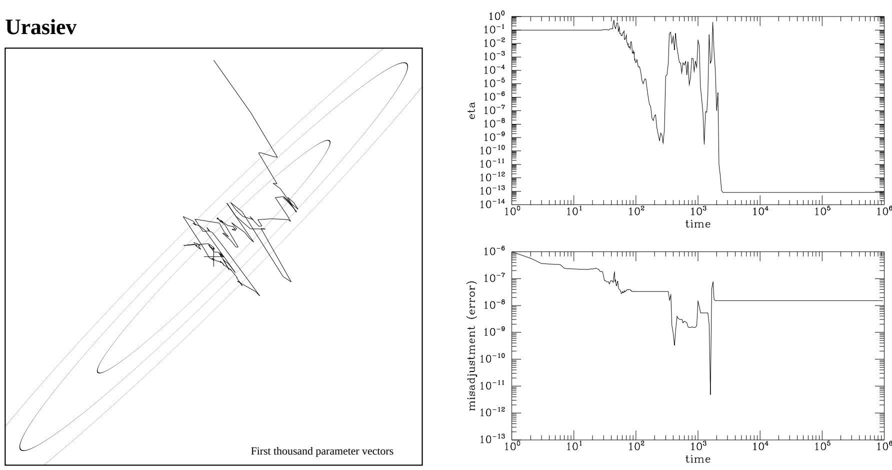

Figure 2: Urasiev's technique (Urasiev, 1988) varies η erratically over several orders of magnitude. The large fluctuations apparently cause η to completely stop changing after a while due to the finite precision of the implementation. Parameters used were $D = 0.2$, $R = 2$, and $U = 1$.

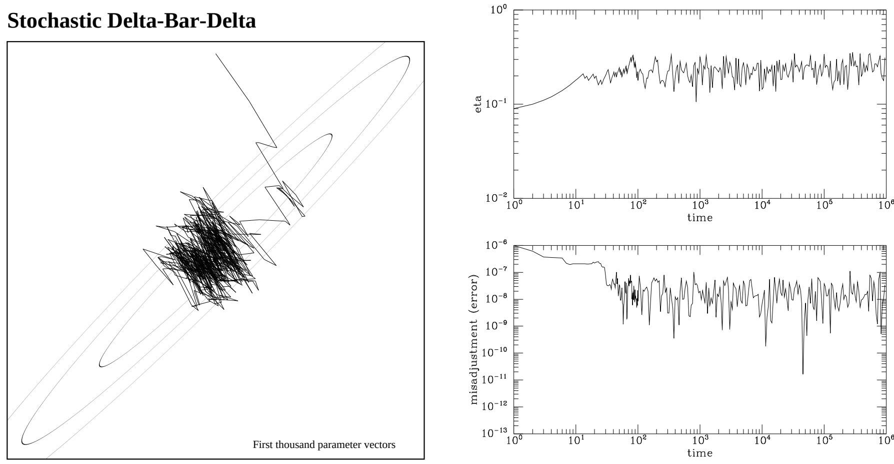

Figure 3: Delta-bar-delta (Jacobs, 1988) was developed for use with deterministic gradient descent. It is also useful for stochastic problems with little noise, which is not the case for this test problem however. In this example η *increases* from its initial value, and then stabilizes. We use the algorithm exactly as it appears in Jacobs' paper with noisy gradients substituted for the true gradient (which is unavailable in the stochastic setting). Parameters used were $\eta_0 = 0.1$, $\theta = 0.3$, $\kappa = 0.01$, and $\phi = 0.1$.

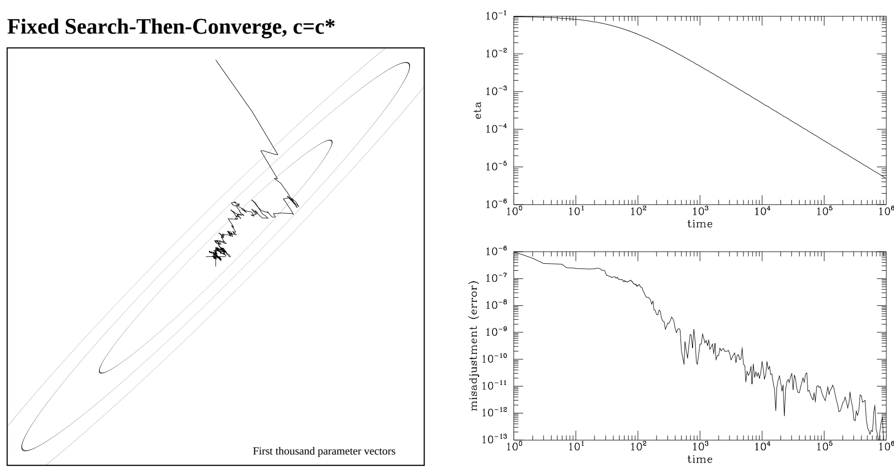

Figure 4: The deterministic search-then-converge schedule with $c = c^*$ gives excellent performance. $c = 2c^*$ would be even better (see footnote 2). However if c^* is not known, you may get performance as in the next two examples. An adaptive technique is called for.

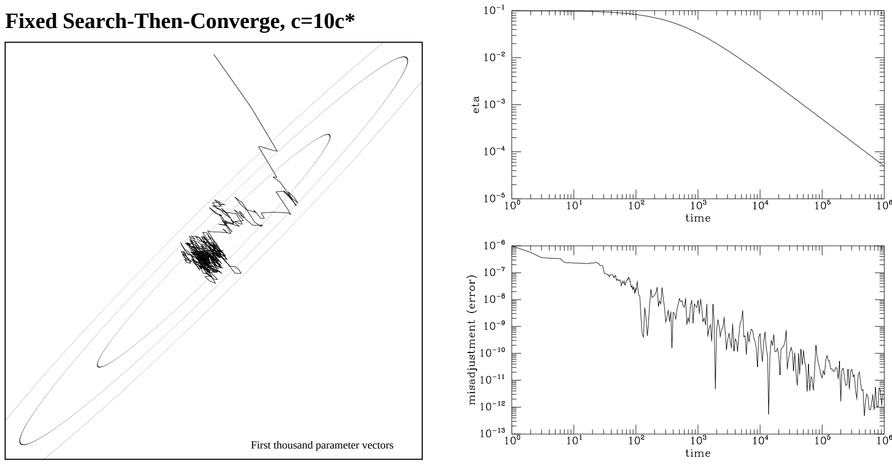

Figure 5: Note that taking $c > c^*$ slows convergence a bit as compared to the $c = c^*$ example in fig. 4, though it could aid escape from bad local minima in a nonlinear problem.

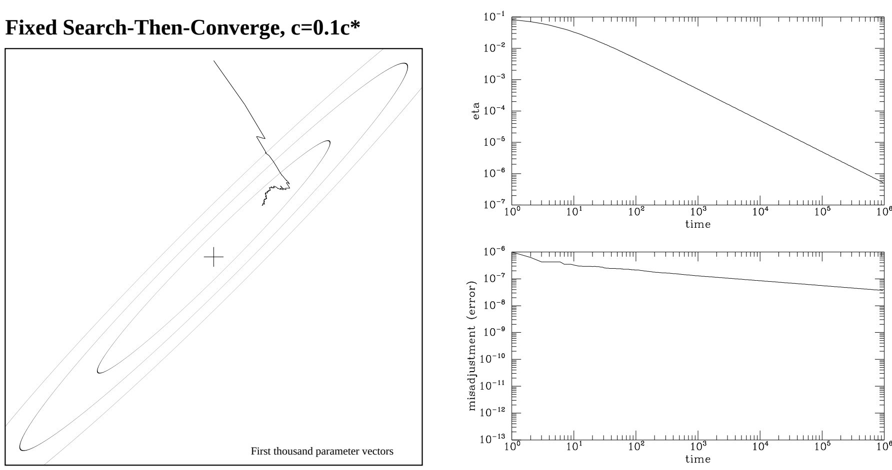

Figure 6: This run illustrates the penalty to be paid if $c < c^*$.

1990b). However, these benefits apply to supervised learning as well (Darken and Moody, 1991). Compare the error curve of the constant learning rate in fig. 1 with those of figs. 4 and 5.

The STC schedules yield the optimal asymptotic rate of convergence if $c > c^* \equiv 1/2\alpha$, where α is the smallest eigenvalue of the hessian of the function g (defined above) at the pertinent minimum (Chung, 1954) (Major and Revesz, 1973).[2] Let "excess error" describe the difference between the current value of the function to be minimized (called g above) and the value at the minimum to which the system is converging. The penalty for choosing $c < c^*$ is that the ratio of the excess error ($\tilde{g}_{c<c^*}$) given c too small to the excess error ($\tilde{g}_{c>c^*}$) with c large enough gets arbitrarily large as training time grows, i.e.

$$\lim_{t \to \infty} \frac{\tilde{g}_{c<c^*}}{\tilde{g}_{c>c^*}} = \infty \quad . \tag{4}$$

The same holds for the ratio of the two distances to the location of the minimum in parameter space.

While the STC schedules work well, their asymptotic performance depends upon the user's choice of c. Since neither η_0 nor τ affects the asymptotic behavior of the system, we will discuss their selection elsewhere. Setting $c > c^*$, however, is vital. Can such a c be determined automatically? Directly estimating α with conventional methods (by estimating the smallest eigenvalue of the hessian at our current estimate of the minimum) is too computationally demanding. This would take at least $O(N^2)$ storage and computation time for each estimate, and would have to be done repeatedly (N is the number of parameters). We are investigating the possibility of a low complexity direct estimation of α by performing a second optimization. However here we take a more unusual approach: we shall determine whether c is large enough by observing the trajectory of the parameter (or "weight") vector.

ON-LINE DETERMINATION OF WHETHER $c < c^*$

We propose that excessive correlation in the parameter change vectors (i.e. "drift") indicates that c is too small (see fig. 7). There are many ways one can quantify the notion of drift. We discuss only two here. Our first definition of the drift $D1(t)$ is

$$D1(t) \equiv ||d(t)||^2 \quad , \tag{5}$$

[2] In the regime that $c > c^*$, Chung (1954) showed that the asymptotic misadjustment goes as $M(t) \propto [c^2 c^*/(c - c^*)]\, t^{-1}$. $M(t)$ is minimized when $c = 2c^*$. For $c = c^*$ and $c < c^*$, Major and Revesz (1973) showed that the asymptotic misadjustment goes as $M(t) \propto (\log t)/t$ and $M(t) \propto t^{-c/c^*}$ respectively. Given these differing asymptotic behaviors, we are concerned only with setting $c > c^*$ and are not attempting to solve the more difficult problem of setting $c = 2c^*$. In a practical stochastic context, setting $c = 2c^*$ may be difficult or even impossible for an $O(N)$ algorithm to achieve without prior knowledge of c^* (this is an open question). Our goal of ensuring that $c > c^*$ avoids the poor relative asymptotic behavior indicated in equation 4.

Figure 7: Two contrasting parameter vector trajectories illustrating the notion of drift.

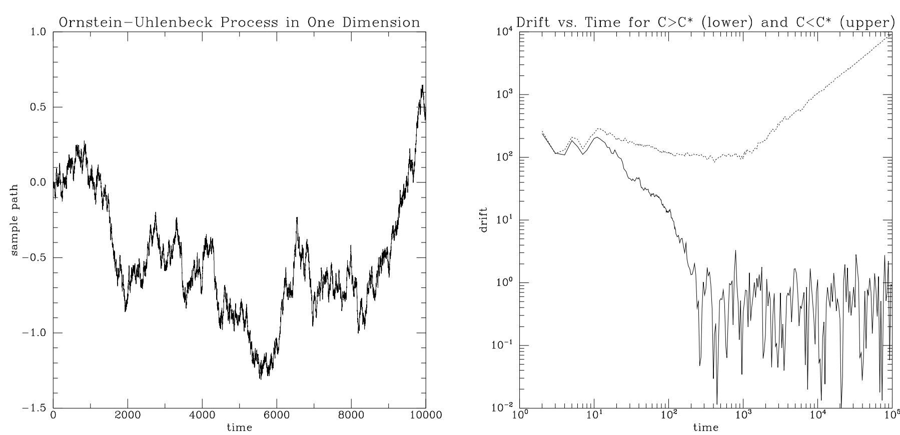

Figure 8: (Left, Fig. 8a) An Ornstein–Uhlenbeck process. This process is zero-mean, gaussian, and stationary (in fact, ergodic). It may be thought of as a random walk with a restoring force towards zero. (Right, Fig. 8b) Measurement of the drift (definition $D1$) for the runs $c = 10c^*$ and $c = .1c^*$, which are discussed in figs. 5 and 6 above.

where $d(t)$ is the drift vector with components

$$d_k(t) \equiv \sqrt{T} \frac{\langle \delta_k(s) \rangle_t}{[\langle (\delta_k(s) - \langle \delta_k(s) \rangle_t)^2 \rangle_t]^{1/2}} \quad . \tag{6}$$

Here, $\delta_k(s)$ is the change in the kth component of the parameter vector at time s, and the angled brackets with subscript t denote an average over the last T parameter changes before time t. We will define T itself as a function of t below. Notice that the numerator is the average parameter step while the denominator is the standard deviation of the steps. Thus, a drift much larger than one may be taken as indicating a "significant" amount of correlation in the change vectors. As a point of reference, if the δ_k are independent and identically distributed, then the d_k would approach unit-variance normals for large T. Since for poorly conditioned problems a small but significant drift may be masked by noise, it is necessary to allow T to grow with time if one desires to detect arbitrarily small drifts. Thus we take $T = \lceil at \rceil$, where $0 < a < 1$.

An alternative, more theoretically tractable, definition of drift is

$$D2(t) \equiv ||d(t)||^2 \quad , \tag{7}$$

where the components of the drift vector $d(t)$ are

$$d_k(t) \equiv \langle \sqrt{s} \delta_k(s) \rangle_t \quad . \tag{8}$$

This definition lacks the normalization of the previous one, and has a somewhat different weighting of the parameter changes. In compensation for the lack of normalization, δ_k is defined to be the kth component of the noisy gradient, which is different from parameter changes by a factor whose time-dependence is known (the learning rate).

Asymptotically, we will take the learning rate to go as c/t. Choosing c too small results in a slow drift of the parameter vector towards the solution in a relatively linear trajectory. When $c > c^*$ however, the trajectory is much more jagged. Compare figs. 5 and 6. In terms of the definitions of drift above, $D1(t)$ **and $D2(t)$ blow up like a power of t when c is too small, but hover about a constant value otherwise**. For an empirical example, see fig. 8b. This fact provides us with a signal to use in future adaptive learning rate schemes for ensuring that c is large enough.

The bold-printed statement above implies that an arbitrarily small change in c which moves it to the opposite side of c^* has dramatic consequences for the behavior of the drift. The following rough argument outlines how one might convince oneself analytically that this interesting discontinuity of behavior is real. We simplify the argument by using the second, simpler definition of the drift and considering a one-dimensional problem. We will study $d(t) = d_1(t)$ for which $D2(t) \equiv [d_1(t)]^2$. Then

$$d_1(t) \equiv \langle \sqrt{s} \delta_1(s) \rangle_t = \langle \sqrt{s} \{ g'[W(s)] + \xi(s) \} \rangle_t \tag{9}$$

where g and ξ are defined in the introduction above. Since $W(t) \to w^*$, as $t \to \infty$, $g'[W(t)] = g''(w^*)[W(t) - w^*] + O[(W(t) - w^*)^2]$. In fact, while not immediately obvious, we can approximate

$$d_1(t) \approx g''(w^*)\langle\sqrt{s}[W(s) - w^*]\rangle_t + \langle\sqrt{s}\xi(s)\rangle_t \tag{10}$$

in an appropriate sense. Define $X(t) \equiv \sqrt{t}[W(t) - w^*]$. For the case $c > c^*$, Kushner (1978) shows how to interpolate the $X(t)$ into a function defined on $[0, \infty)$, such that left-shifting this function by increasingly greater amounts yields a sequence of functions converging in distribution to an Ornstein-Uhlenbeck process (fig. 8a). In the case $c < c^*$, there exists a $p > 0$ such that $t^{-p}X(t)$ converges in distribution to a random variable which is nonzero with probability one (i.e. the sample paths of $t^{-p}X(t)$ are "flat" at large times), so that $X(t)$ grows like t^p. Thus it can be shown that the middle term in (10) converges in distribution to a normal random variable if $c > c^*$, but blows up like a power of t if $c < c^*$. Since the ξ's are independent and have uniformly bounded variances (a traditional assumption), the last term in (10) has a uniformly bounded variance. Thus, the $d_1(t)$ are uniformly bounded variance random variables if $c > c^*$ and blow up like a power of t otherwise.

FINAL REMARKS

While our deterministic *STC* schedules are capable of obtaining excellent, in fact asymptotically optimal, performance provided that $c > c^*$, adaptive schedules which automatically set $c > c^*$ would be greatly advantageous. Our empirical tests support our theoretical expectations that drift can be used to determine whether the crucial parameter c is large enough. Using drift statistics such as $D1(t)$ or $D2(t)$, it will be possible to produce the first adaptive learning rate algorithms which converge at optimal speed, while requiring only $O(N)$ operations per update. We are continuing to investigate candidate algorithms which we expect to be useful for many real-world optimization problems.

A significantly expanded theoretical discussion and additional empirical results (including applications of both *STC* schedules and drift-based adaptive learning rate schedules to multilayer perceptrons) will be presented elsewhere.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge support from AFOSR grant 89-0478, NSF grant ECS-9114333, and ONR grant N00014-89-J-1228.

REFERENCES

K. Chung. (1954) On a stochastic approximation method. *Annals of Mathematical Statistics.* **25**:463-483.

C. Darken and J. Moody. (1990a) Fast adaptive k-means clustering: some empirical results. *Proceedings of the IEEE IJCNN Conference,* IEEE Press, Piscataway, New Jersey. **2**:233-238.

C. Darken and J. Moody. (1990b) Note on learning rate schedules for stochastic optimization. *Advances in Neural Information Processing Systems 3,* Morgan Kauffman, San Mateo, California. 832-838.

C. Darken and J. Moody. (1991) Towards faster stochastic gradient search. *Advances in Neural Information Processing Systems 4,* Morgan Kauffman, San Mateo, California. 1009-1016.

V. Fabian. (1968) On asymptotic normality in stochastic approximation. *Annals of Mathematical Statistics* **39**(4):1327-1332.

L. Goldstein. (1987) Mean square optimality in the continuous time Robbins Monro procedure. Technical Report DRB-306. Department of Mathematics, University of Southern California.

R. Jacobs. (1988) Increased rates of convergence through learning rate adaptation. *Neural Networks.* **1**:295-307.

H. Kesten. (1958) Accelerated stochastic approximation. *Annals of Mathematical Statistics.* **29**:41-59.

H. Kushner. (1978) Rates of convergence for sequential Monte Carlo optimization methods. *SIAM J. Control and Optimization.* **16**:150-168.

L. Ljung and T. Söderström. (1983) *Theory and Practice of Recursive System Identification.* The MIT Press, Cambridge, MA.

P. Major and P.Revesz. (1973) A limit theorem for the Robbins-Monro approximation. *Z. Wahrscheinlichkeitstheorie verw. Geb.* **27**:79-86.

H. Robbins and S. Monro. (1951) A Stochastic Approximation Method. *Ann. Math. Stat.* **22**:400-407.

S. Urasiev. (1988) Adaptive stochastic quasigradient procedures. In *Numerical Techniques for Stochastic Optimization.* Y. Ermoliev and R. Wets Eds. Springer-Verlag.

S. Zhulenev and V. Medovyi. (1978) The strong law of large numbers and normality of Kesten's procedure. *Theor. Prob.* **23**:615-621.