A Theoretical Comparison of Batch-Mode, On-Line, Cyclic, and Almost-Cyclic Learning

Tom Heskes and Wim Wiegerinck

Abstract— **We study and compare different neural network learning strategies: batch-mode learning, on-line learning, cyclic learning, and almost-cyclic learning. Incremental learning strategies require less storage capacity than batch-mode learning. However, due to the arbitrariness in the presentation order of the training patterns, incremental learning is a stochastic process; whereas batch-mode learning is deterministic. In zeroth order, i.e., as the learning parameter η tends to zero, all learning strategies approximate the same ordinary differential equation for convenience referred to as the "ideal behavior." Using stochastic methods valid for small learning parameters η, we derive differential equations describing the evolution of the lowest-order deviations from this ideal behavior. We compute how the asymptotic misadjustment, measuring the average asymptotic distance from a stable fixed point of the ideal behavior, scales as a function of the learning parameter and the number of training patterns. Knowing the asymptotic misadjustment, we calculate the typical number of learning steps necessary to generate a weight within order ϵ of this fixed point, both with fixed and time-dependent learning parameters. We conclude that almost-cyclic learning (learning with random cycles) is a better alternative for batch-mode learning than cyclic learning (learning with a fixed cycle).**

I. Introduction

In most neural-network applications, learning plays an essential role. Through learning, the weights of the network are adapted to meet the requirements of its environment. Usually, the environment consists of a finite number of examples, the training set. We consider two popular ways of learning with this training set: incrementally and batch-mode. With incremental learning, a pattern x^μ is presented to the network, and the weight vector w is updated before the next pattern is considered

$$\Delta w = \eta f(w, x^\mu) \tag{1}$$

with η the learning parameter and $f(\cdot, \cdot)$ the learning rule. This learning rule can be either supervised, e.g., the backpropagation learning rule [1] where x^μ represents an input–output combination, or unsupervised, e.g., the Kohonen learning rule [2] where x^μ stands for an input vector. In batch-mode, we first average the learning rule over all P training patterns before changing the weights

$$\Delta w = \eta \frac{1}{P} \sum_{\mu=1}^{P} f(w, x^\mu) \equiv \eta F(w) \tag{2}$$

where we have defined the average learning rule or drift $F(w)$. Both incremental and batch-mode learning can be viewed as an attempt to realize, or at least approximate, the ordinary differential equation

$$\frac{dw}{dt} = F(w). \tag{3}$$

In [3] it was rigorously established that the sequence of weight vectors following (1) can be approximated by the differential (3) in the limit $\eta \to 0$. However, choosing an infinitesimal learning parameter is not realistic, since the smaller the learning parameter, the longer the time needed to converge. In this paper, we will therefore go one step further and calculate the lowest-order deviations from the differential (3) for small learning parameters η. For convenience, we will refer to (3) as the "ideal behavior." If the drift $F(w)$ can be written as the gradient of an error potential $E(w)$, i.e., if

$$F(w) = -\nabla E(w)$$

the ideal behavior will lead the weights w to a (local) minimum of $E(w)$.

Batch-mode learning is completely deterministic but requires additional storage for each weight which can be inconvenient in hardware applications. Incremental learning strategies, on the other hand, are less demanding on the memory side, but the arbitrariness of the order in which the patterns are presented makes them stochastic. We will consider three popular incremental learning strategies: on-line learning, cyclic learning, and almost-cyclic learning. At each on-line learning step, one of the patterns is drawn at random from the training set and presented to the network. The training process for (almost) cyclic learning consists of training cycles in which each of the P patterns is presented exactly once. Cyclic learning is learning with a fixed cycle, i.e., before learning starts, a particular order of pattern presentation is drawn at random and then fixed in time. In almost-cyclic learning, on the other hand, the order of pattern presentation is continually drawn at random after each training cycle.

On-line learning has been studied using stochastic methods borrowed from statistical physics (see, e.g., [4]–[6]). These studies are not restricted to on-line learning on finite pattern

Manuscript received July 18, 1995; revised April 20, 1995.

The authors are with the Department of Medical Physics and Biophysics, RWCP (Real World Computing Program), Novel Functions SNN (Dutch Foundation for Neural Networks) Laboratory, University of Nijmegen, Geert Grooteplein 21, NL 6525 EZ Nijmegen, The Netherlands.

Publisher Item Identifier S 1045-9227(96)04394-9.

sets, but also discuss learning in changing environments [7], [8], learning with time-correlated patterns [9], and learning with momentum term [10], [11]. The method we use in this paper to derive the (well-known) results on on-line learning can also be applied to learning with cycles of training patterns. Learning with cycles has been studied in [12] for the linear LMS learning rule. Our results are valid for any (nonlinear) learning rule that can be written in the form of (1) Furthermore, we will point out and quantify the important difference between cyclic and almost-cyclic learning.

In Section II, we will apply a mixture between the time-averaging method proposed in [5] and Van Kampen's expansion [13] explained in [6] to derive the lowest-order deviations from the ideal behavior (3) for the various learning strategies. At first reading, the reader may want to skip this section or view it as an appendix. Section III focuses on the asymptotic behavior. The asymptotic misadjustment measures the asymptotic local deviations between the weight vector and a stable fixed point of the ideal behavior (3) and is therefore a useful indication of the network's performance. Another closely related performance measure is the typical number of learning steps n_ϵ necessary to generate a weight within order ϵ of the stable fixed point. We will calculate n_ϵ for the three incremental learning strategies, both with fixed and with time-dependent learning parameters.

II. Theory

In this section we will study batch-mode learning, on-line learning, cyclic learning, and almost-cyclic learning in the limit of small learning parameters η. We will focus on the lowest-order deviations from the ordinary differential equation found in the limit $\eta \to 0$. For convenience, we will use one-dimensional notation. It is straightforward to generalize the results to higher dimensions.

A. Batch-Mode Learning

In the following, we will use subscripts n and m to indicate that time is measured in number of presentations of one pattern and in number of cycles of P patterns, respectively, i.e., $n = mP$. With this convention, the batch-mode learning rule (2) can be written

$$\begin{aligned} w_{n+P} - w_n &= w_{m+1} - w_m \\ &= \eta \sum_{\mu=1}^{P} f(w_m, x^\mu) \\ &= \nu F(w_n) \end{aligned} \tag{4}$$

where we have defined the rescaled learning parameter $\nu \equiv \eta P$. To turn (4) into a set of differential equations, we make the ansatz

$$w_n = \phi(t) + \nu\theta(t)$$

with

$$\begin{aligned} t &\equiv \nu m \\ &= \eta n. \end{aligned}$$

Up to the two lowest orders in the learning parameter η, (4) is equivalent to

$$\frac{d\phi(t)}{dt} = F[\phi(t)] \tag{5}$$

$$\frac{d\theta(t)}{dt} = F'[\phi(t)]\theta(t) - \frac{1}{2}F'[\phi(t)]F[\phi(t)]. \tag{6}$$

For batch-mode learning, the deviation from the ideal behavior (3) is of order $\nu = \eta P$. This deviation, which is a consequence of the discretization of the learning steps, is well known as the error of Euler's method in numerical analysis [14].

B. On-Line Learning

On-line learning is an incremental learning strategy where a weight update takes place after each presentation of one randomly drawn training pattern. Given training pattern x_n at learning step n, the weight change reads

$$w_{n+1} - w_n = \eta f(w_n, x_n). \tag{7}$$

We start with the ansatz in which the fluctuations are small, i.e., we write (see, e.g., [6])

$$w_n = \phi_n + \sqrt{\eta}\xi_n$$

with ϕ_n a deterministic part and ξ_n a noise term. After T iterations of the learning rule (7), we have

$$\begin{aligned} &\phi_{n+T} - \phi_n + \sqrt{\eta}[\xi_{n+T} - \xi_n] \\ &\quad = \eta \sum_{i=0}^{T-1} f(\phi_{n+i}, x_{n+i}) \\ &\qquad + \eta\sqrt{\eta} \sum_{i=0}^{T-1} f'(\phi_{n+i}, x_{n+i})\xi_{n+i} + \mathcal{O}(\eta^2 T) \\ &\quad = \left[\eta \sum_{i=0}^{T-1} f(\phi_n, x_{n+i}) + \mathcal{O}(\eta^2 T^2)\right] \\ &\qquad + \sqrt{\eta}\left[\eta \sum_{i=0}^{T-1} f'(\phi_n, x_{n+i})\xi_n + \mathcal{O}(\eta^2 T^2)\right] \end{aligned} \tag{8, 9}$$

where in the last step we used that $\phi_{n+i} = \phi_n + \mathcal{O}(i\eta)$ and similarly for ξ_{n+i}. We write the first sum on the right-hand side as an average part plus a noise term

$$\eta \sum_{i=0}^{T-1} f(\phi_n, x_{n+i}) = (\eta T)F(\phi_n) + \sqrt{\eta}\sqrt{\eta T}\chi_n(\phi_n) \tag{10}$$

with the drift $F(w)$ defined in (2) and noise $\chi_n(w)$ defined by

$$\chi_n(w) \equiv \frac{1}{\sqrt{T}} \sum_{i=0}^{T-1} [f(w, x_{n+i}) - F(w)].$$

For large T the noise $\chi_n(w)$, consisting of T independent terms, is Gaussian distributed with zero average and variance

$$\begin{aligned} D(w) &\equiv \langle f^2(w, x)\rangle_x - F^2(w) \\ &= \frac{1}{P}\sum_{\mu=1}^{P} f^2(w, x^\mu) - F^2(w). \end{aligned}$$

From (9), we obtain the set of difference equations

$$\begin{aligned}\phi_{n+T} - \phi_n &= (\eta T) F(\phi_n) + \mathcal{O}(\eta^2 T^2) \\ \xi_{n+T} - \xi_n &= (\eta T) F'(\phi_n)\xi_n + \sqrt{\eta T}\chi_n(\phi_n) \\ &\quad + \mathcal{O}(\eta^2 T^2). \end{aligned} \tag{11}$$

For small learning parameters η, we can replace the difference equation for ϕ_n by the differential (5). The deviation due to discretization is of order η (see the analysis of batch-mode learning) which is negligible in comparison with the noise term of order $\sqrt{\eta}$. The difference equation (11) for the noise ξ_n is a discretized version of a Langevin equation which, in the limit $\eta T \to 0$, becomes a continuous-time Langevin equation (see, e.g., [15]). The corresponding Fokker–Planck equation for the probability $\Pi(\xi, t)$ is

$$\frac{\partial \Pi(\xi, t)}{\partial t} = -F'[\phi(t)] \frac{\partial}{\partial \xi} [\xi \Pi(\xi, t)] + \frac{1}{2} D[\phi(t)] \frac{\partial^2 \Pi(\xi, t)}{\partial \xi^2}. \tag{12}$$

In "zeroth order" the weights follow the ideal behavior (5). The randomness of the sampling, however, leads to deviations of order $\sqrt{\eta}$. This result is not new and has been derived in many different ways (see, e.g., [7], [8], and [16]). Our derivation combines the ansätze suggested by Van Kampen's expansion [6], [13] with the time-averaging procedure applied in [5]. In the following section we will show how a similar procedure can be used to study learning with cycles.

C. Learning with Cycles

Let $\vec{x} \equiv \{x_0, \cdots, x_i, \cdots, x_{P-1}\}$ denote a cycle of patterns. There are $P!$ possible different training cycles. Given such a training cycle $\vec{x}$, the weight change can be written as

$$w_{n+P} - w_n = \eta \sum_{i=0}^{P-1} f(w_{n+i}, x_i)$$

which, after substitution of the ansatz

$$w_n = v_n + (\eta P) z_n$$

can be turned into

$$\begin{aligned} &v_{n+P} - v_n + (\eta P)[z_{n+P} - z_n] \\ &\quad = \eta \sum_{i=0}^{P-1} f(v_{n+i}, x_i) + \eta^2 P \sum_{i=0}^{P-1} f'(v_{n+i}, x_i) z_{n+i} \\ &\qquad + \mathcal{O}(\eta^3 P^3) \\ &\quad = \eta \sum_{i=0}^{P-1} f(v_n, x_i) + \eta^2 \sum_{i=1}^{P-1} f'(v_n, x_i) \sum_{j=0}^{i-1} f(v_n, x_j) \\ &\qquad + \eta^2 P \sum_{i=0}^{P-1} f'(v_n, x_i) z_n + \mathcal{O}(\eta^3 P^3) \\ &\quad = (\eta P) F(v_n) + (\eta P)^2 [F'(v_n) z_n + b(v_n, \vec{x}) \\ &\qquad + \tfrac{1}{2} F'(v_n) F(v_n)] + \mathcal{O}(\eta^3 P^3) \end{aligned} \tag{13}$$

with definition

$$b(w, \vec{x}) \equiv \frac{1}{P^2} \sum_{i=1}^{P-1} \sum_{j=0}^{i-1} f'(w, x_i) f(w, x_j) - \frac{1}{2} F'(w) F(w).$$

In terms of the rescaled learning parameter $\nu \equiv \eta P$ and the time m measured in cycles, (13) yields

$$v_{m+1} - v_m = \nu F(v_m) \tag{14}$$

$$\begin{aligned} z_{m+1} - z_m &= \nu [F'(v_m) z_m + b(v_m, \vec{x}) \\ &\quad + \tfrac{1}{2} F'(v_m) F(v_m)] + \mathcal{O}(\nu^2). \end{aligned} \tag{15}$$

Difference equation (14) for v_m is just the batch-mode learning rule (4). Lowest-order correction (6) to the ideal behavior (5) can be incorporated in the difference equation for z_m. Up to order ν^2, (14) and (15) are therefore equivalent to (5) and

$$z_{m+1} - z_m = \nu \{F'[\phi(t)] z_m + b[\phi(t), \vec{x}_m]\} + \mathcal{O}(\nu^2) \tag{16}$$

with $\vec{x}_m$ the particular cycle presented at "cycle step" m. Neglecting the higher-order terms, (7) can be viewed as an incremental learning rule for training *cycle* $\vec{x}_m$, just as (7) is the learning rule for training *pattern* x_n. All necessary information about the cycle $\vec{x}_m$ is contained in the term $b[\phi(t), \vec{x}_m]$. With cyclic learning, we have the same cycle $\vec{x}_m = \vec{x}$ at all cycle steps with almost-cyclic learning we draw the cycle $\vec{x}_m$ at random at each cycle step.

1) Almost-Cyclic Learning: With almost-cyclic learning, subsequent training cycles are drawn at random; almost-cyclic learning is on-line learning with training cycles instead of training patterns. We can apply a similar time-averaging procedure as in our study of on-line learning. Starting from the ansatz

$$z_m = \frac{1}{P} \psi_m + \sqrt{\eta} \zeta_m$$

we obtain, after T iterations of the "learning rule" (16) and neglecting all terms of order $\nu^2 T^2$ and higher

$$\psi_{m+T} - \psi_m = (\nu T) \{F'[\phi(t)] \psi_m + B[\phi(t)]\} \tag{17}$$

$$\zeta_{m+T} - \zeta_m = (\nu T) F'[\phi(t)] \zeta_m + \sqrt{\nu T} \chi_m[\phi(t)] \tag{18}$$

with definitions

$$B(w) \equiv P \langle b(w, \vec{x}) \rangle_{\vec{x}}$$

and

$$\chi_m(w) \equiv \sqrt{\frac{P}{T}} \sum_{i=0}^{T-1} [b(w, \vec{x}_{m+i}) - \langle b(w, \vec{x}) \rangle_{\vec{x}}].$$

The variance $Q(w)$ for the white noise $\chi_m(w)$ is given by

$$Q(w) \equiv P[\langle b^2(w, \vec{x}) \rangle_{\vec{x}} - \langle b(w, \vec{x}) \rangle_{\vec{x}}^2]. \tag{19}$$

Calculation of the average $B(w)$ and the variance $Q(w)$ is tedious but straightforward. In terms of

$$C(w) \equiv \frac{1}{P} \sum_{\mu=1}^{P} f'(w, x^\mu) f(w, x^\mu)$$

and

$$G(w) \equiv \frac{1}{P} \sum_{\mu=1}^{P} [f'(w, x^\mu)]^2$$

we obtain

$$B(w) = -\frac{1}{2}C(w)$$
$$Q(w) = \frac{1}{12}\{[F'(w)]^2 D(w) + F^2(w)G(w) + 2F'(w)F(w)C(w)\} + \frac{1}{12P}[D(w)G(w) - C^2(w)]. \quad (20)$$

We can, similar to what we did for on-line learning, turn (17) for ψ_m and (18) for ζ_m into a differential equation for $\psi(t)$ and a Fokker–Planck equation for $\Pi(\zeta, t)$

$$\frac{d\psi(t)}{dt} = F'[\phi(t)]\psi(t) + B[\phi(t)] \quad (21)$$
$$\frac{\partial \Pi(\zeta, t)}{\partial t} = -F'[\phi(t)]\frac{\partial}{\partial \zeta}[\zeta \Pi(\zeta, t)] + \frac{1}{2}Q[\phi(t)]\frac{\partial^2 \Pi(\zeta, t)}{\partial \zeta^2}. \quad (22)$$

Recalling all our definitions and ansätze, we conclude that this set of equations, in combination with (5), can be used to predict the behavior of

$$w_n = \phi(\eta n) + \eta\psi(\eta n) + (\eta P)\sqrt{\eta}\zeta(\eta n) + \mathcal{O}(\eta^2 P^2).$$

For almost-cyclic learning, the deviation from the ideal behavior due to discretization of learning steps is of order η, and the deviation due to the randomness of the sampling is of order $\eta^{3/2}P$.

2) Cyclic Learning: With cyclic learning, a particular cycle $\vec{x}$ is drawn at random from the set of $P!$ possible cycles and then kept fixed at all times. The "learning rule" is, up to order ν^2, given in (16). Given a particular training cycle $\vec{x}^\alpha$ with corresponding $b^\alpha(w) \equiv b(w, \vec{x}^\alpha)$, the evolution of the deviation z^α is completely deterministic

$$\frac{dz^\alpha(t)}{dt} = F'[\phi(t)]z^\alpha(t) + b^\alpha[\phi(t)].$$

We can split $z^\alpha(t)$ in an average part $\psi(t)$, common to all cycles, and a specific part $\gamma^\alpha(t)$

$$z^\alpha(t) = \frac{1}{P}\psi(t) + \frac{1}{\sqrt{P}}\gamma^\alpha(t).$$

The evolution of $\psi(t)$ follows (21), and the evolution of $\gamma^\alpha(t)$ is given by

$$\frac{d\gamma^\alpha(t)}{dt} = F'[\phi(t)]\gamma^\alpha(t) + q^\alpha[\phi(t)] \quad (23)$$

where we have defined the term

$$q^\alpha(w) \equiv \sqrt{P}[b^\alpha(w) - \langle b(w, \vec{x})\rangle_{\vec{x}}]$$

with zero average and variance $Q(w)$ defined in (19) and computed in (20). From

$$w_n^\alpha = \phi(\eta n) + \eta\psi(\eta n) + \eta\sqrt{P}\gamma^\alpha(\eta n)$$

we conclude that for learning with a particular fixed cycle $\vec{x}^\alpha$, the weight vector w_n^α follows the ideal behavior $\phi(\eta n)$ with correction terms of order $\eta\sqrt{P}$. These correction terms for cyclic learning are larger than those for almost-cyclic learning.

D. A Note on the Validity

Let us reconsider our ansätze. We assumed that deviations from the ideal behavior $\phi(t)$ scale with some positive power of η, i.e., the smaller η the smaller these deviations. Looking at the differential and Langevin-type equations for these deviations, we see that the deviations remain bounded if and only if $F'[\phi(t)] < 0$. Assuming that the drift $F(w)$ can be written as minus the gradient of some error potential $E(w)$, this implies that the theory is valid in regions of weight space where the error potential is convex which is true in the vicinity of the local minima w^*. Outside these so-called "attraction regions," our derivations are only valid on short time scales (see [6] for a more detailed discussion on the validity of Fokker–Planck approaches of on-line learning processes).

A second notion concerns perfectly learnable problems. For these problems there exists a weight or a set of weights w^* such that $f(w^*, x^\mu) \equiv 0$ for all patterns μ in the training set. An example is a perceptron learning a linearly separable problem. For these perfectly learnable problems the "perfect" weight w^* acts like a sink—all learning processes will end up in this state. In our analysis, a perfectly trainable network has a vanishing asymptotic diffusion. Most practical problems, however, are not perfectly learnable, and a minimum of the error potential w^* corresponds to the best compromise on the training set. In this paper, we therefore restrict ourselves to networks that are not perfectly trainable.

III. Asymptotic Properties

The ideal behavior (3) leads w to a stable fixed point w^* obeying

$$F(w^*) = 0 \quad \text{and} \quad F'(w^*) < 0$$

i.e., to a (local) minimum of the error potential $E(w)$ (assuming such an error potential exists). In this section we will study asymptotic properties of the various learning strategies. We will focus on the asymptotic behavior of the misadjustment

$$M_n \equiv \langle (w - w^*)^2\rangle_w = [\langle w\rangle_w - w^*]^2 + [\langle w^2\rangle_w - \langle w\rangle_w^2] \equiv \Delta + \Sigma \quad (24)$$

where the average is over the distribution of the weights after a large number of learning steps n. Δ and Σ are called the bias and the variance, respectively.

A. The Asymptotic Misadjustment

First, we will consider the asymptotic misadjustment $A \equiv M_\infty$ for the various learning strategies. We will concentrate on training sets with a large number of patterns and small learning parameters, i.e., we will consider the situation $1 \ll P \ll 1/\eta$. In the following we will only give the results in leading order.

1) Batch-Mode Learning: A stable fixed point w^* of the differential (2) is also a stable fixed point of the batch-mode (2). Therefore, all deviations from the ideal behavior will completely vanish, and the batch-mode learning rule yields zero asymptotic misadjustment.

2) On-Line Learning: The most important contribution to the asymptotic misadjustment for on-line learning stems from the noise ξ due to the randomness of the sampling. The Fokker–Planck (12) yields

$$\begin{aligned} A &= \Sigma \\ &= \eta\langle\xi^2\rangle \\ &= \frac{D(w^*)}{2|F'(w^*)|}\,\eta \end{aligned}$$

i.e., the asymptotic misadjustment is of order η. The diffusion $D(w^*)$ measures the local fluctuations in the learning rule, the derivative $F'(w^*)$ the local curvature of the error potential. The bias Δ is of order η^2 and thus negligible [8].

3) Almost-Cyclic Learning: For almost-cyclic learning, the bias Δ follows from the stationary solution of the average deviation ψ. Equation (21) gives

$$\begin{aligned} \Delta &= \eta^2\psi^2 \\ &= \left[\frac{C(w^*)}{2F'(w^*)}\right]^2 \eta^2 \end{aligned}$$

where $C(w^*)$ measures the correlation between the learning rule and its derivative. The variance is of higher order in η but strongly depends on the number of patterns P

$$\begin{aligned} \Sigma &= \eta^3 P^2\langle\zeta^2\rangle \\ &= \eta^3 P^2\,\frac{Q(w^*)}{2|F'(w^*)|} \\ &= \frac{|F'(w^*)|D(w^*)}{24}\,\eta^3 P^2 \end{aligned}$$

which follows from the Fokker–Planck equation (22) and (20). Depending on the term ηP^2, the asymptotic misadjustment is dominated by either the bias or the variance.

4) Cyclic Learning: With cyclic learning, we first have to calculate the asymptotic misadjustment for a particular cycle $\vec{x}^\alpha$. In lowest-order, we obtain

$$A^\alpha = \eta^2[\sqrt{P}\gamma^\alpha]^2$$

with γ^α the asymptotic solution of the differential (23)

$$\gamma^\alpha = \frac{q^\alpha(w^*)}{|F'(w^*)|}.$$

The average asymptotic misadjustment, which is the average over all possible cycles, thus obeys

$$\begin{aligned} A &\equiv \langle A^\alpha\rangle_\alpha \\ &= \frac{D(w^*)}{12}\,\eta^2 P. \end{aligned} \tag{25}$$

The (average) asymptotic misadjustment for cyclic learning is (for small learning parameters and a considerable number of patterns P) always an order of magnitude larger than the asymptotic misadjustment for almost-cyclic learning. Almost-cyclic learning is therefore a better alternative for batch-mode learning than cyclic learning.

B. Necessary Number of Learning Steps

In this section we consider the decay of the misadjustment to its asymptotic solution for the three incremental learning strategies, both with fixed and with time-dependent learning parameters. We will work in the limit $1 \ll P^2 \ll 1/\eta_n \ll n$. The analysis in Section II shows that the convergence rate is the same for all learning strategies discussed in this paper, i.e., in lowest-order we can write

$$M_{n+1} - M_n = 2\eta_n|F'(w^*)|[A(\eta_n) - M_n] \tag{26}$$

with η_n the learning parameter at learning step n and $A(\eta)$ the asymptotic misadjustment for the various learning strategies. We recall from Section III that $A(\eta) \propto \eta$ for on-line learning, $A(\eta) \propto \eta^2$ for almost-cyclic learning, and $A(\eta) \propto P\eta^2$ for cyclic learning. Following [12], we consider the concept of an ϵ-optimal weight, i.e., a weight w within order ϵ of the local minimum w^*. We will calculate the typical number of learning steps n_ϵ necessary to reach such an ϵ-optimal weight, i.e., the typical number of learning steps until the misadjustment is of order ϵ^2. Our analysis is a generalization of [12] to general (nonlinear) learning rules and points out the important difference between the three incremental learning strategies.

1) Fixed Learning Parameters: With a fixed learning parameter, i.e., $\eta_n = \eta$ for all n, the misadjustment M_n obeying (26) can be written

$$M_n = A(\eta) + \mathcal{O}[e^{-2|F'(w^*)|n\eta}]. \tag{27}$$

To make sure that the asymptotic misadjustment is of order ϵ^2, we have to choose $\eta = \mathcal{O}[A^{-1}(\epsilon^2)]$ with $A^{-1}(\cdot)$ the inverse of $A(\cdot)$. Substitution into (27) yields

$$e^{-\lambda n\eta} = \mathcal{O}(\epsilon^2)$$

with $\lambda = \mathcal{O}(1)$. Thus, the typical number of learning steps n_ϵ necessary to generate an ϵ-optimal weight is

$$n_\epsilon = \mathcal{O}\left\{\frac{1}{A^{-1}(\epsilon^2)}\log\left[\frac{1}{\epsilon}\right]\right\}.$$

We have $n_\epsilon \sim 1/\epsilon^2 \log(1/\epsilon)$ for on-line learning, $n_\epsilon \sim 1/\epsilon \log(1/\epsilon)$ for almost-cyclic learning, and $n_\epsilon \sim \sqrt{P}/\epsilon \log(1/\epsilon)$ for cyclic learning.

2) Time-Dependent Learning Parameter: With time-dependent learning parameters we can choose the learning parameter η_n yielding the fastest possible decay of the misadjustment M_n. Optimizing the right-hand side of (26) with respect to η_n, we obtain a relationship between M_n and η_n

$$A(\eta_n) - M_n + \eta_n A'(\eta_n) = 0$$

and thus

$$M_n = (r+1)A(\eta_n) \tag{28}$$

with $r = 1$ for on-line learning and $r = 2$ for learning with cycles. Substitution of (28) into (26) yields a difference

equation for η_n. For large n, the solution of this difference equation reads

$$\eta_n = \frac{r+1}{2|F'(w^*)|n}. \tag{29}$$

Combining (28) and (29), we obtain $M_n \sim 1/n$ and thus $n_\epsilon \sim 1/\sqrt{\epsilon}$ for on-line learning, $M_n \sim 1/n^2$ and $n_\epsilon \sim 1/\epsilon$ for almost-cyclic learning, and $M_n \sim P/n^2$ and $n_\epsilon \sim \sqrt{P}/\epsilon$ for cyclic learning. With time-dependent learning parameters, the necessary number of learning steps n_ϵ is order $\log(1/\epsilon)$ smaller than with fixed learning parameters. In both cases, cyclic learning requires about $\sqrt{P}$ more learning steps than almost-cyclic learning.

IV. Discussion

We have studied the consequences of different learning strategies. For *local optimization*, learning with cycles is a better alternative for batch-mode learning than on-line learning. The asymptotic misadjustment for learning with cycles scales with η^2, whereas the asymptotic misadjustment for on-line learning is proportional to η. Furthermore, learning with cycles requires less learning steps to get close to the minimum. Almost-cyclic learning yields, in this respect, even better performances than cyclic learning.

Learning with cycles can be interpreted as a more "conservative" learning strategy, since within each cycle the network receives information about all training patterns. With on-line learning, on the other hand, the time span between two presentations of a particular pattern can be much larger than the period of one cycle. As a result, the fluctuations in the network weights are larger for on-line learning than for learning with cycles. The asymptotic bias for cyclic learning can be viewed as an artifact of the fixed presentation order. Almost-cyclic learning prevents this artifact by randomizing over the presentation orders at the (lower) price of small asymptotic fluctuations.

In our presentation, we have used one-dimensional notation. However, it is straightforward to generalize the results to general high-dimensional weight vectors. The asymptotic misadjustment for the various learning strategies then depends on, most importantly, the Hessian matrix and the diffusion matrix. The Hessian matrix is related to the local curvature of the error potential, the diffusion matrix to the fluctuations in the learning rule. Training sets with lots of redundant information have a lower diffusion and thus a lower asymptotic bias than training sets with very specific and contradictory information.

For backpropagation as well as for other learning rules minimizing the loglikelihood of training patterns, the diffusion matrix is, up to a global scale factor, proportional to the Hessian matrix (see, e.g., [17]). As a consequence, for on-line learning the asymptotic distribution of the weights is isotropic (see [5]). Similarly, it can be shown that the asymptotic covariance matrix for cyclic learning (averaged over all possible representation orders) is, in a lowest-order approximation, proportional to the diffusion matrix as could be guessed from the one-dimensional result (25). Therefore, this covariance matrix is, for learning rules based on loglikelihood procedures, also proportional to the Hessian matrix. The asymptotic covariance matrix for almost-cyclic learning is proportional to the square of the Hessian matrix. The covariance matrix that would result from a local lowest-order approximation of a Gibbs distribution is proportional to the inverse of the Hessian matrix.

Our analysis is valid in the limit of small learning parameters η, but even then only locally, i.e., in the vicinity of local minima and on short time scales. The theory can therefore not be used to make quantitative predictions about *global* properties of the learning behavior such as escape times out of local minima or stationary distributions. However, the above local description may be helpful to explain some aspects of global properties. For instance, the larger the local fluctuations, the higher the probability to escape (see, e.g., [18] and [19]). This might be one of the reasons why on-line learning, the learning strategy with the largest fluctuations, often yields the best results, especially in complex problems with many local minima (see, e.g., [20]).

References

[1] D. Rumelhart, J. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Cambridge, MA: MIT Press, 1986.

[2] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics,* vol. 43, pp. 59–69, 1982.

[3] C. Kuan and K. Hornik, "Convergence of learning algorithms with constant learning rates," *IEEE Trans. Neural Networks,* vol. 2, pp. 484–489, 1991.

[4] G. Radons, "On stochastic dynamics of supervised learning," *J. Phys. A,* vol. 26, pp. 3455–3461, 1993.

[5] L. Hansen, R. Pathria, and P. Salamon, "Stochastic dynamics of supervised learning," *J. Phys. A,* vol. 26, pp. 63–71, 1993.

[6] T. Heskes, "On Fokker–Planck approximations of on-line learning processes," *J. Phys. A,* vol. 27, pp. 5145–5160, 1994.

[7] S. Amari, "A theory of adaptive pattern classifiers," *IEEE Trans. Elec. Comput.,* vol. EC-16, 299–307, 1967.

[8] T. Heskes and B. Kappen, "Learning processes in neural networks," *Phys. Rev. A,* vol. 44, pp. 2718–2726, 1991.

[9] W. Wiegerinck and T. Heskes, "On-line learning with time-correlated patterns," *Europhysics Lett.,* vol. 28, pp. 451–455, 1994.

[10] G. Orr and T. Leen, *Momentum and Optimal Stochastic Search,* M. Mozer, P. Smolensky, D. Touretzky, J. Elman, and A. Weigend, Eds., in *Proc. 1993 Connectionist Models Summer School.* Hillsdale: Erlbaum, 1993.

[11] W. Wiegerinck, A. Komoda, and T. Heskes, "Stochastic dynamics of learning with momentum in neural networks," *J. Phys. A,* vol. 27, pp. 4425–4437, 1994.

[12] Z. Luo, "On the convergence of LMS algorithm with adaptive learning rate for linear feedforward networks," *Neural Computation,* vol. 3, pp. 226–245, 1991.

[13] N. van Kampen, *Stochastic Processes in Physics and Chemistry.* Amsterdam: North-Holland, 1992.

[14] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C,* 2nd ed. Cambridge: Cambridge Univ. Press, 1992.

[15] C. Gardiner, *Handbook of Stochastic Methods,* 2nd ed. Berlin: Springer, 1985.

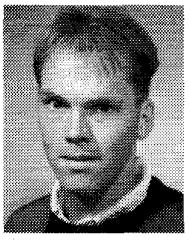
[16] G. Radons, H. Schuster, and D. Werner, "Fokker–Planck description of learning in backpropagation networks," in *Proc. Int. Neural Network Conf. 1990 Paris.* Dordrecht: Kluwer Academic, 1990, pp. 993–996.

[17] W. Buntine and A. Weigend, "Computing second derivatives in feed-forward networks: A review," *IEEE Trans. Neural Networks,* vol. 5, pp. 480–488, 1994.

[18] T. Heskes, E. Slijpen, and B. Kappen, "Learning in neural networks with local minima," *Phys. Rev. A,* vol. 46, pp. 5221–5231, 1992.

[19] ——, "Cooling schedules for learning in neural networks," *Physical Rev. E,* vol. 47, pp. 4457–4464, 1993.

[20] E. Barnard, "Optimization for neural nets," *IEEE Trans. Neural Networks,* vol. 3, pp. 232–240, 1992.



Tom Heskes received both the M.Sc. and the Ph.D. degrees in physics from the University of Nijmegen, The Netherlands, in 1988 and 1993, respectively.

From August 1993 until June 1994, he did postdoctoral work at the Beckman Institute, University of Illinois at Champaign-Urbana. Currently, he has a postdoctoral position at the Department of Medical Physics and Biophysics of the University of Nijmegen, The Netherlands. His research interests include theory on neural-network dynamics, with regard to both industrial applications and to biological modeling.

Wim Wiegerinck received the M.Sc. degree (hons.) in physics from the University of Amsterdam, The Netherlands, in 1988, and the Ph.D. degree in physics from the University of Nÿmegen, the Netherlands.

From October 1988 until April 1990, he worked at the Royal Netherlands Meteorological Institute on the predictability of weather forecasting. Since 1990 he has worked at the Department of Medical Physics and Biophysics of the University of Nÿmegen. His research interests include the theory of neural networks and dynamical processes.