Open access • Book Chapter • DOI:10.1016/S0924-6509(08)70038-2

On-line learning processes in artificial neural networks — Source link

Tom Heskes, Hilbert J. Kappen

Institutions: Radboud University Nijmegen

Published on: 01 Jan 1993 - North-holland Mathematical Library (Amsterdam : Elsevier)

Topics: Online machine learning, Competitive learning, Stability (learning theory), Unsupervised learning and Instance-based learning

On-line learning processes in artificial neural networks

Tom M. Heskes Bert Kappen

Department of Medical Physics and Biophysics,
University of Nijmegen, Geert Grooteplein 21,
6525 EZ Nijmegen, The Netherlands.

Abstract

We study on-line learning processes in artificial neural networks from a general point of view. On-line learning means that a learning step takes place at each presentation of a randomly drawn training pattern. It can be viewed as a stochastic process governed by a continuous-time master equation.

On-line learning is necessary if not all training patterns are available all the time. This occurs in many applications when the training patterns are drawn from a time-dependent environmental distribution. Studying learning in a changing environment, we encounter a conflict between the adaptability and the confidence of the network's representation. Minimization of a criterion incorporating both effects yields an algorithm for on-line adaptation of the learning parameter.

The inherent noise of on-line learning makes it possible to escape from undesired local minima of the error potential on which the learning rule performs (stochastic) gradient descent. We try to quantify these often made claims by considering the transition times between various minima. We apply our results on the transitions from "twists" in two-dimensional self-organizing maps to perfectly ordered configurations. Finally, we discuss the capabilities of on-line learning for global optimization.

In J. Taylor, editor, *Mathematical Foundations of Neural Networks*, Amsterdam, 1993. Elsevier, pages 199–233.

1 Introduction

1.1 Why a theory for on-line learning?

In neural network models, learning plays an essential role. Learning is the mechanism by which a network adapts itself to its environment. The result of this adaptation process, in both natural as well as in artificial systems, is that the network obtains a representation of its environment. This representation is encoded in its plasticities, such as synapses and thresholds. The function of a neural network can be described in terms of its input-output relation, which in turn is fully determined by the architecture of the network and by the learning rule. Examples of such functions may be classification (as in multi-layered perceptrons), feature extraction (as in networks that perform a principle component analysis), recognition, transformation for motor tasks, or memory. The representation that the network has learned of the environment enables the network to perform its function in a way that is "optimally" suited for the environment on which it is taught.

Despite the apparent differences in their functionalities, most learning rules in the current network literature share the following properties.

1. Neural networks learn from examples. An example may be a picture that must be memorized or a combination of input and desired output of the network that must be learned. The total set of examples or stimuli is called the training set or the environment of the neural network.

2. The learning rule contains a global scale factor, the "learning parameter". It sets the typical magnitude of the weight changes at each learning step.

In this chapter, we set up and work out a theoretical framework based on these two properties. It covers both supervised learning (learning with "teacher", e.g., backpropagation [55], for a review see [33, 65]) and unsupervised learning (learning without "teacher", e.g., Kohonen learning [37], for a review see [6]). The approach taken in this chapter is therefore quite general. It includes and extends results from studies on specific learning rules (see e.g. [3, 9, 48, 53]).

1.2 Outline of this chapter

In artificial neural networks, on-line learning is modeled by *randomly* drawing examples from the environment. This introduces stochasticity in the learning process. The learning process becomes a discrete-time Markov process[1], which can be transformed into a continuous-time master equation. The study of learning processes becomes essentially a study of a particular class of master equations. In section 2 we point out the correct way to approximate this master equation by a Fokker-Plank equation in the limit of small learning parameters. We discuss the consequences of this approach in the case of just one fixed point of the (average) learning dynamics.

Section 3 is more like an intermezzo. Here we discuss two other approaches. The Langevin approach, which leads to an equilibrium Gibbs distribution, has become very popular in neural network literature. However, on-line learning, as we define it, cannot be formulated in terms of a Langevin equation, does not lead to a Gibbs distribution, and is therefore more difficult to study. We will also discuss the more "mathematical" approach which describes on-line learning using techniques from stochastic approximation theory. The mathematical approach has led to many important and rigorously proven theorems, some of which will be mentioned in section 3.

On-line learning, if compared with batch-mode learning where a learning step takes place on account of the *whole* training set, is necessary if not all training patterns are available all

[1] The underlying assumption is that subsequent stimuli are uncorrelated. This is the case for almost all artificial neural network learning rules. However, for biological learning processes and for some applications subsequent stimuli may be correlated. Then the results of our analysis do not apply.

the time. This not only the case for biological learning systems, but also in many practical applications, especially in applications such as financial modeling, economic forecasting, robot control, etcetera, when the training patterns are drawn from a time-dependent environmental distribution. This notion leads to the study of on-line learning in a changing environment in section 4. Using the same techniques as in section 2, we encounter a conflict between the adaptability and the confidence or accuracy of the network's representation. Minimization of a suitable criterion, the so-called "misadjustment", leads to an optimal learning parameter for learning in a changing environment.

The derivation of the optimal learning parameter in section 4 is nice, but of little practical use. To calculate this learning parameter, one needs detailed information about the neural network and its environment, information that is usually not available. In section 5 we try to solve this problem by considering the statistics of the weights. This yields an autonomous algorithm for learning-parameter adjustment.

Another argument in favor of on-line learning, is the possibility to escape from undesired local minima of the energy function or error potential on which the learning rule performs (stochastic) gradient descent. In section 6 we try to quantify these often made claims by considering the transition times between various minima of the error potential. Starting from two hypotheses, based on experimental observations and theoretical arguments, we show that these transition times scale exponentially with some constant, the so-called "reference learning parameter", divided by the learning parameter.

Well-known examples of undesired fixed points of the average learning dynamics are topological defects in self-organizing maps. Using the theory of section 6, we calculate in section 7.1 the reference learning parameters for the transitions from "twists" in two-dimensional maps to perfectly ordered configurations. We compare the theoretically obtained results with results obtained from straightforward simulations of the learning rule.

Finally, we discuss in section 8 to what extent on-line learning might be used as a global optimization method. We derive cooling schedules that guarantee convergence to a global minimum. In these cooling schedules, the reference learning parameters discussed in section 6 play an important role. We compare the optimization capabilities of on-line backpropagation and "Langevin-type" learning for a specific example with profound local minima.

2 Learning processes and their average behavior

2.1 From random walk to master equation

Let the adaptive elements of a neural network, such as synapses and thresholds, be given by a weight vector[2] $\mathbf{w} = (w_1, \ldots, w_N)^T \in \mathbb{R}^N$. At distinct iteration times $\mathbf{w}$ is changed due to the presentation of a training pattern $\vec{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, which is drawn at random according to a probability distribution $\rho(\vec{x})$. The new weight vector $\mathbf{w}' = \mathbf{w} + \Delta\mathbf{w}$ depends on the old weight vector and on the training pattern:

$$\Delta\mathbf{w} \;=\; \eta\, \mathbf{f}(\mathbf{w}, \vec{x})\,. \qquad (1)$$

The function $\mathbf{f}$ is called the learning rule, η the learning parameter.

Because of the *random* pattern presentation, the learning process is a stochastic process. We have to talk in terms of probabilities, averages, and fluctuations. The most obvious probability to start with is the probability $p_i(\mathbf{w})$ to be in state $\mathbf{w}$ after i iterations. This probability obeys a random walk equation

$$p_i(\mathbf{w}') \;=\; \int d^N w\, T(\mathbf{w}'|\mathbf{w})\, p_{i-1}(\mathbf{w}), \qquad (2)$$

[2] We use the notation A^T to denote the transpose of the matrix or vector A.

with $T(\mathbf{w}'|\mathbf{w})$ the transition probability to "walk" in one learning step from state $\mathbf{w}$ to state $\mathbf{w}'$:

$$T(\mathbf{w}'|\mathbf{w}) \;=\; \int d^n x \, \rho(\vec{x}) \, \delta^N(\mathbf{w}' - \mathbf{w} - \eta \mathbf{f}(\mathbf{w}, \vec{x})). \tag{3}$$

The random walk equation (2) gives a description in discrete time steps.

Bedeaux, Lakatos-Lindenberg, and Shuler [7] showed, that a continuous-time description can be obtained through the assignment of *random* values Δt to the time interval between two succeeding iteration steps. If these Δt are drawn from a probability density

$$\varrho(\Delta t) \;=\; \frac{1}{\tau} \exp\left[-\frac{\Delta t}{\tau}\right],$$

the probability $\phi(i,t)$, that after time t there have been exactly i transitions, follows a Poisson process. The probability $P(\mathbf{w}, t)$, that a network is in state $\mathbf{w}$ at *time* t, reads

$$P(\mathbf{w}, t) \;=\; \sum_{i=0}^{\infty} \phi(i,t) p_i(\mathbf{w}).$$

This probability function can be differentiated with respect to time, yielding the master equation

$$\frac{\partial P(\mathbf{w}', t)}{\partial t} \;=\; \int d^N w \; \left[W(\mathbf{w}'|\mathbf{w}) P(\mathbf{w}, t) \;-\; W(\mathbf{w}|\mathbf{w}') P(\mathbf{w}', t) \right], \tag{4}$$

with the transition probability per unit time

$$W(\mathbf{w}'|\mathbf{w}) \;=\; \frac{1}{\tau} T(\mathbf{w}'|\mathbf{w}). \tag{5}$$

Through τ we have introduced a physical time scale. Here we have presented a nice mathematical trick to transform a discrete time random walk equation into a continuous time master equation. It is valid for all values of τ and η. For the rest of this chapter we will choose $\tau = 1$, i.e., the average time between two learning steps is our unit of time.

For notational convenience we introduce the averages over the ensemble $\Xi(t)$ of learning networks

$$\langle \Phi(\mathbf{w}) \rangle_{\Xi(t)} \;\stackrel{\text{def}}{=}\; \int d^N w \, P(\mathbf{w}, t) \, \Phi(\mathbf{w}),$$

and over the set Ω of training patterns

$$\langle \Psi(\vec{x}) \rangle_{\Omega} \;\stackrel{\text{def}}{=}\; \int d^n x \, \rho(\vec{x}) \, \Psi(\vec{x}),$$

for arbitrary function $\Phi(\mathbf{w})$ and $\Psi(\vec{x})$.

The dynamics of equation (4) cannot be solved in general. We will point out the incorrect (section 2.2) and the correct (section 2.3) way to approximate this master equation for small learning parameters η. To simplify the notation, we will only consider the one-dimensional case. In our discussion of the asymptotic dynamics (section 2.4), we will generalize to N dimensions.

2.2 The Fokker-Planck approximation of the Kramers-Moyal expansion

A totally equivalent description of the master equation is given by its full Kramers-Moyal expansion [63]

$$\frac{\partial P(w,t)}{\partial t} \;=\; \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} \left(\frac{\partial}{\partial w}\right)^n \{a_n(w) P(w,t)\}, \tag{6}$$

with the so-called jump moments

$$a_n(w) \;\stackrel{\text{def}}{=}\; \int dw' \, (w - w')^n T(w|w') \;=\; \eta^n \, \langle f^n(w,x) \rangle_{\Omega} \;\stackrel{\text{def}}{=}\; \eta^n \tilde{a}_n(w), \tag{7}$$

where all $\tilde{a}_n$ are of order 1, i.e., independent of η. By terminating this series at the second term, one obtains the Fokker-Planck equation

$$\frac{\partial P(w,t)}{\partial t} = -\eta \frac{\partial}{\partial w} \left\{ \tilde{a}_1(w) P(w,t) \right\} + \frac{\eta^2}{2} \frac{\partial^2}{\partial w^2} \left\{ \tilde{a}_2(w) P(w,t) \right\} . \tag{8}$$

In one dimension, the equilibrium distribution of the Fokker-Planck equation can be written in closed form:

$$P_{\rm s}(w) = \frac{\mathcal{N}}{\tilde{a}_2(w)} \exp\left[\frac{2}{\eta} \int^w dw' \, \frac{\tilde{a}_1(w')}{\tilde{a}_2(w')} \right] , \tag{9}$$

with $\mathcal{N}$ a normalization constant.

Because of the convenience and the simplicity of the result, the Fokker-Planck approach is very popular, also in neural-network literature on on-line learning processes [23, 44, 50, 53]. However, it is *incorrect*! Roughly speaking, this approximation is possible if and only if the average step size $\langle \Delta w \rangle$ and the variance of the step size $\langle (\Delta w - \langle \Delta w \rangle)^2 \rangle$ are proportional to the *same* small parameter [14]. Learning rules of the type (1) have $\langle \Delta w \rangle = \mathcal{O}(\eta)$ but $\langle (\Delta w - \langle \Delta w \rangle)^2 \rangle = \mathcal{O}(\eta^2)$ and thus do not satisfy this so-called "scaling assumption". To convince ourselves, we substitute the equilibrium distribution (9) into the Kramers-Moyal expansion (6) and notice that the third, fourth, $\ldots$, ∞ terms are *all* of the same order as the first and second order terms: formally there is no reason to break off the Kramers-Moyal series after any number of terms.

2.3 A small-fluctuations expansion

Intuitively, a stochastic process can often be viewed as an average, deterministic trajectory, with stochastic fluctuations around this trajectory. Using Van Kampen's system size expansion [63] (see also [14]), it is possible to obtain the precise conditions under which this intuitive picture is valid. We will refer to this as the small-fluctuations expansion. It consists of the following steps.

1. Following Van Kampen, we make the "small-fluctuations Ansatz", i.e., we choose a new variable ξ such that
$$w = \phi(t) + \sqrt{\eta}\xi \tag{10}$$
with $\phi(t)$ a function to be determined. Equation (10) says that the time-dependent stochastic variable w is given by a deterministic part $\phi(t)$ plus a term of order $\sqrt{\eta}$ containing the (small) fluctuations. *A posteriori*, this Ansatz should be verified. The function $\Pi(\xi,t)$ is the probability $P(w,t)$ in terms of the variable ξ:
$$\Pi(\xi,t) \overset{\rm def}{=} P(\phi(t) + \sqrt{\eta}\xi, t) .$$

2. Using simple chain rules for differentiation, we transform the Kramers-Moyal expansion (6) for $P(w,t)$ into a differential equation for $\Pi(\xi,t)$:
$$\frac{\partial \Pi(\xi,t)}{\partial t} - \frac{1}{\sqrt{\eta}} \frac{d\phi(t)}{dt} \frac{\partial \Pi(\xi,t)}{\partial \xi} = \sum_{n=1}^{\infty} \frac{(-1)^n \eta^{n/2}}{n!} \left(\frac{\partial}{\partial \xi} \right)^n \left\{ \tilde{a}_n(\phi(t) + \sqrt{\eta}\xi) \, \Pi(\xi,t) \right\} .$$

3. We choose the function $\phi(t)$ such that the lowest order terms on the left- and righthandside cancel, i.e.,
$$\frac{1}{\eta} \frac{d\phi(t)}{dt} = \tilde{a}_1(\phi(t)) . \tag{11}$$
This is called the deterministic equation.

4. We make a Taylor expansion of $\tilde{a}_n(\phi(t)+\sqrt{\eta}\xi)$ in powers of $\sqrt{\eta}$. After some rearrangements we obtain

$$\frac{1}{\eta}\frac{\partial\Pi(\xi,t)}{\partial t} = \sum_{m=2}^{\infty}\sum_{n=1}^{m}\frac{(-1)^n\eta^{(m-2)/2}}{n!\,(m-n)!}\tilde{a}_n^{(m-2)}(\phi(t))\left(\frac{\partial}{\partial\xi}\right)^n\{\xi^{m-n}\,\Pi(\xi,t)\}\ ,$$

5. In the limit $\eta \to 0$ only the term $m=2$ survives on the righthandside. This is called the linear noise approximation. The remaining differential equation for $\Pi(\xi,t)$ is the Fokker-Planck equation

$$\frac{1}{\eta}\frac{\partial\Pi(\xi,t)}{\partial t} = -\tilde{a}_1'(\phi(t))\frac{\partial}{\partial\xi}\{\xi\Pi(\xi,t)\} + \frac{1}{2}\tilde{a}_2(\phi(t))\frac{\partial^2}{\partial\xi^2}\Pi(\xi,t)\ , \tag{12}$$

   where the prime denotes differentiation with respect to the argument.

6. From equation (12) we calculate the dynamics of the average fluctuations $\langle\xi\rangle_{\Xi(t)}$ and the size of the fluctuations $\langle\xi^2\rangle_{\Xi(t)}$:

$$\begin{aligned}\frac{1}{\eta}\frac{\partial\langle\xi\rangle_{\Xi(t)}}{\partial t} &= a_1'(\phi(t))\,\langle\xi\rangle_{\Xi(t)}\\ \frac{1}{\eta}\frac{\partial\langle\xi^2\rangle_{\Xi(t)}}{\partial t} &= 2a_1'(\phi(t))\left\langle\xi^2\right\rangle_{\Xi(t)} + a_2(\phi(t))\ .\end{aligned} \tag{13}$$

7. We started with the Ansatz that ξ is of order 1. From equation (13) we conclude that the final result is consistent with the Ansatz, provided that both evolution equations converge, i.e., that

$$a_1'(\phi(t))\ <\ 0\ . \tag{14}$$

   So, there are regions of weight space where the small-fluctuations expansion is valid ($a_1' < 0$) and where it is invalid ($a_1' \geq 0$).

Let us summarize what we have done so far. We have formulated the learning rule (1) in terms of a discrete time Markov process (2). Introducing Poisson distributed time steps we have transformed this discrete random walk equation into a continuous time master equation (4). Making a small-fluctuations Ansatz for small learning parameters η, we have derived equation (11) for the deterministic behavior and equation (12) for the probability distribution of the fluctuations around this deterministic behavior. At the same time we have derived the condition (14) which must be satisfied for this description to be valid in the limit of small learning parameters η.

Now that we have made a rigorous expansion of the master equation, we can refine our bold statement that the Fokker-Planck approximation is incorrect. If we substitute the small-fluctuations Ansatz (10) into the Fokker-Planck equation (8), then the lowest-order Fokker-Planck equation for ξ is exactly the same as the lowest-order term (12) in the small-fluctuations expansion. So, if we are only interested in the lowest order, we might as well use the Fokker-Planck approximation of section 2.2, as long as we keep in mind that only the small-noise approximation, i.e., the lowest order term (12) has any validity [14]. So, all features beyond that approximation are spurious and cannot be taken seriously [63]. In practice this means that we may still apply the Fokker-Planck approximation if we study learning with just one minimum, but must suppress the temptation to extend this approach to learning with various minima.

2.4 Asymptotic results in N dimensions

The first two jump moments defined in equation (7) play an important role and are therefore given special names: the drift vector, which is just the average learning rule

$$\mathbf{f}(\mathbf{w}) \stackrel{\text{def}}{=} \langle \mathbf{f}(\mathbf{w}, \vec{x}) \rangle_{\Omega} \ ,$$

and the diffusion matrix

$$D \stackrel{\text{def}}{=} \left\langle \mathbf{f}(\mathbf{w}, \vec{x})\, \mathbf{f}^T(\mathbf{w}, \vec{x}) \right\rangle_{\Xi(t)} \ , \tag{15}$$

containing the fluctuations in the learning rule. Furthermore, we define the Hessian matrix $H(\mathbf{w})$ with components

$$H_{ij}(\mathbf{w}) \ = \ -\frac{\partial f_i(\mathbf{w})}{\partial w_j} \ . \tag{16}$$

If and only if the Hessian matrix is symmetric[3], an energy function or error potential $E(\mathbf{w})$ can be defined such that the learning rule performs (stochastic) gradient descent on this error:

$$\mathbf{f}(\mathbf{w}) \ = \ -\nabla E(\mathbf{w}) \ , \tag{17}$$

where ∇ stands for differentiation with respect to the weight vector $\mathbf{w}$. The Hessian matrix gives the curvature of the error potential in the different directions. The condition (14) says that the small-fluctuations expansion is valid in regions of weight space with positive definite Hessian $H(\mathbf{w})$. These regions will be called attraction regions.

The deterministic equation (11) reads in N dimensions

$$\frac{1}{\eta}\frac{d\boldsymbol{\phi}(t)}{dt} \ = \ \mathbf{f}(\boldsymbol{\phi}(t)) \ . \tag{18}$$

The attractive fixed point solutions of this "average learning dynamics" will be denoted by $\mathbf{w}^*$. If there exists an error potential, then these fixed points are simply the (local) minima. At a fixed point $\mathbf{w}^*$ we have no drift, i.e., $\mathbf{f}(\mathbf{w}^*) = \mathbf{0}$ and a positive definite Hessian $H(\mathbf{w}^*)$. The typical local relaxation time towards these fixed points is

$$\tau_{\text{local}} = \frac{1}{\eta \lambda_{\min}(\mathbf{w}^*)} \ , \tag{19}$$

with $\lambda_{\min}(\mathbf{w}^*)$ the smallest eigenvalue of the Hessian $H(\mathbf{w}^*)$. To study the asymptotic convergence, we can make an expansion around the minimum $\mathbf{w}^*$. In [28, 63] it is shown that, after linearization of $\phi(t)$ around the fixed point $\mathbf{w}^*$, the evolution equations (11) and (13) are equivalent with

$$\begin{aligned} \frac{1}{\eta}\frac{d\mathbf{m}(t)}{dt} \ &= \ -H\mathbf{m}(t) \\ \frac{1}{\eta}\frac{d\Sigma^2(t)}{dt} \ &= \ -H\Sigma^2(t) - \Sigma^2(t)H^T + \eta D \ , \end{aligned} \tag{20}$$

where the Hessian and the diffusion matrix are both evaluated at the fixed point $\mathbf{w}^*$ and with definitions for the bias $\mathbf{m}(t)$ and covariance matrix $\Sigma^2(t)$

$$\mathbf{m}(t) \stackrel{\text{def}}{=} \langle \mathbf{w} \rangle_{\Xi(t)} - \mathbf{w}^* \ , \qquad \Sigma^2(t) \stackrel{\text{def}}{=} \left\langle \left[\mathbf{w} - \langle \mathbf{w} \rangle_{\Xi(t)}\right] \left[\mathbf{w} - \langle \mathbf{w} \rangle_{\Xi(t)}\right]^T \right\rangle_{\Xi(t)} \ . \tag{21}$$

It can be shown that this linearization is allowed for η small enough and t large enough [28].

[3] In the literature, the matrix $H(\mathbf{w})$ is only called Hessian if it is indeed symmetric.

From the linear Fokker-Planck equation (12) and the asymptotic evolution equations (20) we conclude that the asymptotic probability distribution for small learning parameters η is a simple Gaussian, with its average at the fixed point $\mathbf{w}^*$ and a covariance matrix Σ^2 obeying

$$H\Sigma^2 + \Sigma^2 H^T \;=\; \eta D\,. \tag{22}$$

So, there are persistent fluctuations of order η that will only disappear in the limit $\eta \to 0$. These theoretical predictions are in good agreement with simulations (see [28] and simulations in the following sections).

3 Intermezzo: other approaches

3.1 The Langevin approach

In this section we will point out the difference between the "intrinsic" noise due to the random presentation of training patterns and the "artificial" noise in studies on the generalization capabilities of neural networks (see e.g. [57, 64]). In the latter case, the noise is added to the deterministic equation (18), i.e., the weights evolve according to the Langevin equation

$$\frac{d\mathbf{w}(t)}{dt} \;=\; -\nabla E(\mathbf{w}(t)) + \sqrt{2T}\boldsymbol{\xi}(t)\,, \tag{23}$$

where $\boldsymbol{\xi}(t)$ is white noise obeying

$$\langle \xi_i(t)\xi_j(t')\rangle \;=\; \delta_{ij}\,\delta(t-t')\,.$$

The Langevin equation (23) is equivalent to the Fokker-Planck equation [63]

$$\frac{\partial P(\mathbf{w},t)}{\partial t} \;=\; -\nabla\left\{\mathbf{f}(\mathbf{w})\,P(\mathbf{w},t)\right\} \;+\; T\,\nabla^2 P(\mathbf{w},t)\,.$$

The equilibrium distribution is [compare with equation (9)]

$$P_{\rm s}(\mathbf{w}) \;=\; \frac{1}{Z}\,\exp\left[-\frac{E(\mathbf{w})}{T}\right]\,, \tag{24}$$

with Z a normalization constant. The existence of this Gibbs distribution raises the idea to put learning in the framework of statistical mechanics [45, 57, 64]. In these studies, the Langevin equation (23) is more an "excuse" to arrive at the Gibbs distribution (24) than an attempt to study the dynamics of learning processes in artificial neural networks. The equilibrium distribution of the master equation for on-line learning processes is *not* a simple Gibbs distribution (see also [23]), which makes the analysis of on-line learning processes much more difficult.

Because of the equilibrium Gibbs distribution (24), the Langevin equation (23) has also been proposed as a method for global optimization [2, 15, 17, 21]. The discrete-time version

$$\mathbf{w}(t+\Delta t) - \mathbf{w}(t) \;=\; \mathbf{f}(\mathbf{w})\,\Delta t \;+\; \sqrt{2T}\,\boldsymbol{\xi}\,\sqrt{\Delta t}\,, \tag{25}$$

with $\boldsymbol{\xi}$ Gaussian white noise of variance 1, can be simulated easily. The smaller Δt, the closer the correspondence with the continuous Langevin equation. We will call this "Langevin-type learning" and we will come back on it in section 8.3. Note that equation (25) does indeed satisfy the "scaling assumption" mentioned in section 2.2: both the average step size and the variance of the step size are proportional to Δt. This scaling property explains why equation (25) can indeed be approximated by a globally valid Fokker-Planck equation, and the learning rule (1) not.

3.2 Mathematical approach

Besides the "physical" approach which starts from the master equation, there is the "mathematical" approach which treats on-line learning in the context of stochastic approximation theory. The starting point in this approach is the so-called interpolated process. With $\mathbf{w}_n$ the network state and η_n the learning parameter after n iterations, the interpolated process $\mathbf{w}(t)$ is defined by

$$\mathbf{w}(t) \; = \; \frac{t_n - t}{\eta_n}\mathbf{w}_{n-1} + \frac{t - t_{n-1}}{\eta_n}\mathbf{w}_n \; , \qquad \text{for } t_{n-1} \leq t < t_n \; ,$$

with $t_0 \stackrel{\text{def}}{=} 0$ and $t_n \stackrel{\text{def}}{=} \eta_1 + \ldots + \eta_n$. This approach has led to many important, rigorously proven theorems. For example, if η_n tends to zero at a suitable rate, i.e., such that

$$\lim_{n\rightarrow\infty} \eta_n = 0 \; , \qquad \sum_{n=0}^{\infty} \eta_n = \infty \; , \qquad \sum_{n=0}^{\infty} \eta_n^p < \infty \qquad \text{for some } p > 1 \; , \tag{26}$$

then the interpolated process of $\mathbf{w}_n$ eventually follows the solution trajectory of the ordinary differential equation (18) with probability 1 [43, 47]. Furthermore, if these and some additional technical requirements are satisfied, the learning process will always converge to one of the fixed points $\mathbf{w}^*$ of this differential equation. In neural-network literature this method has been applied to the analysis of feature extraction algorithms [34, 56]. Similar techniques have been used to study the convergence of general learning algorithms for small constant learning parameters [40]. In the context of global optimization (see section 8) the work of Kushner [41, 42] is worth mentioning. In particular Kushner shows that convergence to the global optimum occurs almost surely, provided that in the limit $n \rightarrow \infty$ the learning parameter decreases proportional to $1/\log n$ [42, 65].

4 A conflict in a changing environment

4.1 Motivation and mathematical description

Equation (22) states that we must drop the learning parameter to zero in order to prevent asymptotic fluctuations in the network state. This has been the usual strategy in the training of artificial neural networks. But this is certainly not the kind of behavior one would expect from a true adaptive system that a neural network, based on real biological systems, should be. A true adaptive system can always adapt itself to changes in the environment. Biological neural systems are famous for their ability to correct for the lengthening of limbs during growth, or their ability to recover (at least partially) after severe damage or surgery. This kind of adaptability is also desirable for artificial neural networks, e.g., for networks for the control of robots that suffer from wear and tear, or for neural networks for the modeling of economic processes. In this section we will therefore discuss the performance of neural networks learning in a changing environment [28].

Mathematically speaking, a changing environment corresponds to a time-dependent input probability $\rho(\vec{x}, t)$. The probability density of network states $\mathbf{w}$ still follows a continuous-time master equation, but now with time-dependent transition probability $T_t(\mathbf{w}'|\mathbf{w})$:

$$T_t(\mathbf{w}'|\mathbf{w}) \; = \; \int d^n x \; \rho(\vec{x}, t) \, \delta^N(\mathbf{w}' - \mathbf{w} - \eta \mathbf{f}(\mathbf{w}, \vec{x})) \; \stackrel{\text{def}}{=} \; \left\langle \delta^N(\mathbf{w}' - \mathbf{w} - \eta \mathbf{f}(\mathbf{w}, \vec{x})) \right\rangle_{\Omega(t)} \; ,$$

where $\Omega(t)$ stands for the set of training patterns, the "environment", at time t. The fixed points $\mathbf{w}^*(t)$ of the deterministic equation

$$\frac{1}{\eta}\frac{d\mathbf{w}(s)}{ds} \; = \; \langle \mathbf{f}(\mathbf{w}(s), \vec{x}) \rangle_{\Omega(t)} \; , \tag{27}$$

may depend on time. We define the "misadjustment" $\mathcal{E}$ as the average squared Euclidian distance with respect to this fixed point $\mathbf{w}^*(t)$ [11]:

$$\mathcal{E} \overset{\text{def}}{=} \frac{1}{T}\int_0^T dt \, \left\langle |\mathbf{w} - \mathbf{w}^*(t)|^2 \right\rangle_{\Xi(t)} = \frac{1}{T}\int_0^T dt \, |\mathbf{m}(t)|^2 + \mathrm{Tr}\left[\Sigma^2(t)\right] . \tag{28}$$

The bias $\mathbf{m}(t)$, defined in equation (21) but now with time-dependent $\mathbf{w}^*(t)$ instead of fixed $\mathbf{w}^*$, is a measure of how well the ensemble of learning networks follows the environmental change *on the average*. It gives the typical delay between what the average network state is, $\langle \mathbf{w} \rangle_{\Xi(t)}$, and what it should be, $\mathbf{w}^*(t)$. The covariance matrix $\Sigma^2(t)$ gives the width of the distribution and thus a measure of "confidence". T is a time window to be discussed later.

4.2 An example: Grossberg learning in a changing environment

Let us first discuss a simple model that can be solved without any approximations. We consider the Grossberg learning rule [19]

$$\Delta w = \eta(x - w) ,$$

in a time-dependent environment. The input distribution is moving along the axis with a constant velocity v, i.e., $\rho(x,t) = \tilde{\rho}(x - vt)$. We choose

$$\tilde{\rho}(x) = \frac{1}{2l} \, \theta(l + x) \, \theta(l - x) ,$$

with $\theta(x) = 1$ for $x > 0$ and $\theta(x) = 0$ for $x < 0$. So, x is drawn with equal probability from the interval $[vt - l, vt + l]$. The input standard deviation $\chi = l/\sqrt{3}$, is constant. The aim of this learning rule is to make w coincide with the mean value of the probability distribution $\rho(x,t)$, i.e., the fixed point $w^*(t)$ of the deterministic equation (27) obeys

$$w^*(t) = \langle x \rangle_{\Omega(t)} = vt .$$

So, $\dot{w}^* = v$, the rate of change of the fixed point solution is equal to the rate of change of the environment.

Straightforward calculations show that the evolution of the bias $m(t)$ and the variance $\Sigma^2(t)$ is governed by

$$\begin{aligned} \frac{dm(t)}{dt} &= -\eta \, m(t) + v , \\ \frac{d\Sigma^2(t)}{dt} &= -(2 - \eta)\eta \, \Sigma^2(t) \; + \; \eta^2 m^2(t) \; + \; \eta^2 \chi^2 . \end{aligned}$$

This set of differential equations decays exponentially to the stationary solution

$$m = \frac{v}{\eta} , \quad \Sigma^2 = \frac{\eta^2\chi^2 + v^2}{\eta(2 - \eta)} . \tag{29}$$

Note that this behavior is really different from the behavior in a fixed environment. In a fixed environment ($v = 0$) the asymptotic bias is negligible if compared with the variance[4]. However, in a changing environment ($v > 0$), the bias is inversely proportional to the learning parameter η, and can become really important if this learning parameter is chosen too small. In figure 1 we have shown the (simulated) probability density $P(w - w^*(t))$ for three different values of the speed v. For zero velocity the bias is zero and the distribution is sharply peaked. For a relatively small velocity, the influence on the width of the distribution is negligible, but the effect on the bias is clearly visible. For a relatively large speed, the variance is also affected and can get pretty large.

[4] For the linear learning rule discussed in this example it is even zero. In general, the nonlinearity of the learning rule leads to a bias of $\mathcal{O}(\eta)$ whereas the standard deviation is of $\mathcal{O}(\sqrt{\eta})$.

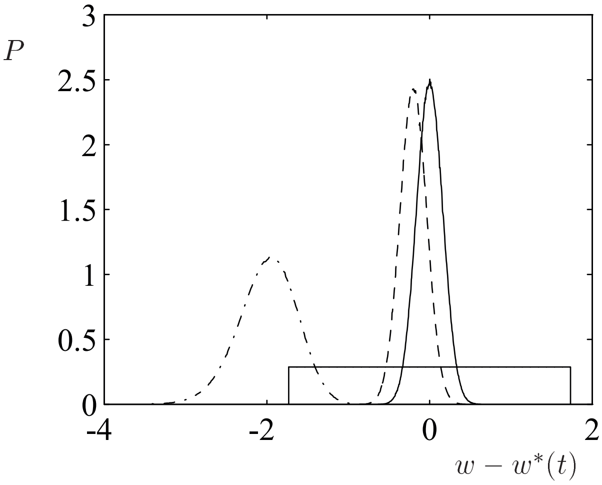

Figure 1: Probability distribution for time-dependent Grossberg learning. Learning parameter $\eta = 0.05$, standard deviation $\chi = 1.0$. The input probability $\rho(x,t)$ is drawn for reference (solid box). Zero velocity (solid line). Small velocity: $v = 0.01$ (dashed line). Large velocity: $v = 0.1$ (dash-dotted line).

A good measure for the learning performance is the misadjustment defined in equation (28). In the limit $T \rightarrow \infty$, we can neglect the exponential transients to the stationary state (29). We obtain

$$\mathcal{E} \; = \; \frac{\eta^3 \chi^2 + 2v^2}{\eta^2(2-\eta)} \; .$$

This misadjustment is sketched in figure 2, together with simulation results. For small learning parameters the bias dominates the misadjustment and we have

$$\mathcal{E} \; \approx \; \frac{v^2}{\eta^2} \quad \text{for} \quad \eta \ll (v/\chi)^{2/3} \; .$$

On the other hand, for larger learning parameters the variance yields the most important contribution:

$$\mathcal{E} \; \approx \; \frac{\eta \chi^2}{2} \quad \text{for} \quad (v/\chi)^{2/3} \ll \eta \ll 2 \; .$$

Somewhere in between these two limiting cases, the misadjustment has a minimum at the *optimal* learning parameter η_{optimal} which is for this particular example the solution of the cubic equation

$$2\chi^2 \eta_{\text{optimal}}^3 + 3v^2 \eta_{\text{optimal}} - 2v^2 \; = \; 0 \; .$$

Reasonable performance of the learning systems can only be expected if $v \ll \chi$, i.e., if the displacement of the input probability distribution per learning step is much smaller than its width. In this limit, we obtain

$$\eta_{\text{optimal}} \; \approx \; \left[\frac{v}{\chi}\right]^{2/3} \quad \text{for} \quad v \ll \chi \; .$$

This optimal learning parameter gives the best compromise between fast adaptability, which asks for a large learning parameter, and high confidence, which requires a small (but not too small!) learning parameter. A similar "accuracy conflict" is noted by Wiener in his work on linear prediction theory [67].

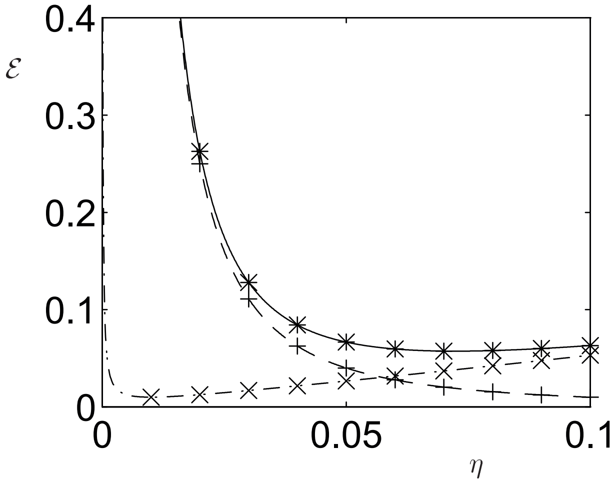

Figure 2: Misadjustment as a function of the learning parameter for Grossberg learning in a changing environment. Squared bias (computed, dashed line; simulated, +), variance (computed, dash-dotted line; simulated, x), and error (computed, solid line; simulated, $*$). Simulations were done with 5000 neural networks. Standard deviation input: $\chi = 1.0$. Velocity: $v = 0.01$.

4.3 Nonlinear learning rules in nonstationary environments

The Grossberg learning rule is linear and therefore exactly solvable. Of course, most practical learning rules in neural networks are nonlinear and high dimensional. For nonlinear high-dimensional learning rules the basic idea is still the same: there is a conflict between fast adaptability (a small bias) and high confidence (a small variance). In order to calculate a learning parameter that yields a good compromise between these two competing goals, we have to make approximations, similar to the ones made in section 2. So, we have to require that the learning parameter is so small that it is allowed to make the usual small-fluctuations expansion. To linearize around to fixed point, we must now also require that the rate of change $\mathbf{v} \equiv \dot{\mathbf{w}}^*$ is much smaller than the typical weight change ηf. Provided these requirements are fulfilled, the evolution of the bias $\mathbf{m}(t)$ and the covariance $\Sigma^2(t)$ is governed by [28]

$$\begin{aligned} \frac{d\mathbf{m}(t)}{dt} &= -\eta H(t)\mathbf{m}(t) - \mathbf{v}(t) \, , \\ \frac{d\Sigma^2(t)}{dt} &= -\eta H(t)\Sigma^2(t) - \eta\Sigma^2(t)H^T(t) + \eta^2 D(t) \, , \end{aligned} \tag{30}$$

with notation $H(t) \overset{\text{def}}{=} H(\mathbf{w}^*(t))$, and so on. Let us furthermore assume that the changes in the "speed" $\mathbf{v}$, the diffusion D, and the curvature H are so slow that they can be considered constant on the local relaxation time τ_{local} [see equation (19)]. Then the bias and covariance matrix tend to stationary values. The stationary bias is inversely proportional to the learning parameter η and proportional to the speed v, whereas the variance is proportional to the learning parameter and more or less independent of the speed. So, for nonlinear learning rules we also obtain a misadjustment of the form [28]

$$\mathcal{E} \approx \frac{\alpha v^2}{\eta^2} + \beta\eta \, ,$$

with α and β constants that depend on the diffusion D and the curvature H at the fixed point. Here the time window T must be larger than the local relaxation time τ_{local} and smaller than the time in which at least one of the quantities $\mathbf{v}$, D, or H, changes substantially. Again, the optimal learning parameter is proportional to $v^{2/3}$. For slow changes v and learning parameters

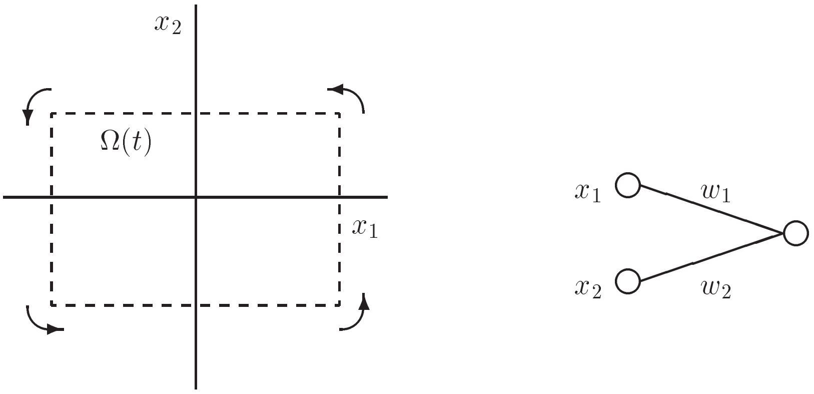

Figure 3: Oja learning. A unit is taught with two-dimensional examples from a rectangle which is rotating around the origin. The principal component of the covariance matrix lies parallel to the longest side of the rectangle.

near the optimal learning parameter all conditions for the validity of the evolution equations (30) are fulfilled [28].

4.4 An example: Oja learning in a changing environment

A simple example of a nonlinear learning rule is the Oja learning rule [49]

$$\Delta \mathbf{w} \;=\; \eta\, \mathbf{w}^T\mathbf{x}\,[\mathbf{x} - (\mathbf{w}^T\mathbf{x})\mathbf{w}]\;.$$

This learning rule searches for the principal component of the input distribution, i.e., the eigenvector of the covariance matrix $C \overset{\text{def}}{=} \left\langle \mathbf{x}\mathbf{x}^T \right\rangle_\Omega$ with the largest eigenvalue. The network structure and input space is pictured in figure 3. We take a network with one output neuron, two input neurons and two weights. The inputs are drawn with equal probability from a two-dimensional box with sides $2l_1$ and $2l_2$:

$$\tilde{\rho}(x_1,x_2) \;=\; \frac{1}{4l_1l_2}\theta(l_1+x_1)\,\theta(l_1-x_1)\,\theta(l_2+x_2)\,\theta(l_2-x_2)\;.$$

The covariance matrix of this input distribution is diagonal:

$$\tilde{C} \;=\; \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix}\,,$$

with $\Lambda_\alpha \overset{\text{def}}{=} l_\alpha^2/3$ for $\alpha = 1,2$. If we choose $l_1 > l_2$, then the two fixed point solutions of the differential equation (27) are $\mathbf{w}^*(t) = \pm(1,0)^T$. So, the fixed point solution is normalized, but is still free to lie along the positive or negative axis. To model learning in a changing environment, the box is rotated around an axis perpendicular to the box, going through the origin, with angular velocity ω. The principal component of this time-dependent input distribution obeys

$$\mathbf{w}^*(t) \;=\; \begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \end{bmatrix}\;.$$

For small angular velocities ω and small learning parameters η, we can apply the approximations discussed above to calculate the squared bias and the variance. We obtain

$$|\mathbf{m}|^2 \;=\; \frac{1}{\eta^2}\left[\frac{\omega}{\Lambda_1-\Lambda_2}\right]^2\,, \quad \mathrm{Tr}\,[\Sigma^2] \;=\; \frac{\eta}{2}\frac{\Lambda_1\Lambda_2}{\Lambda_1-\Lambda_2}\;.$$

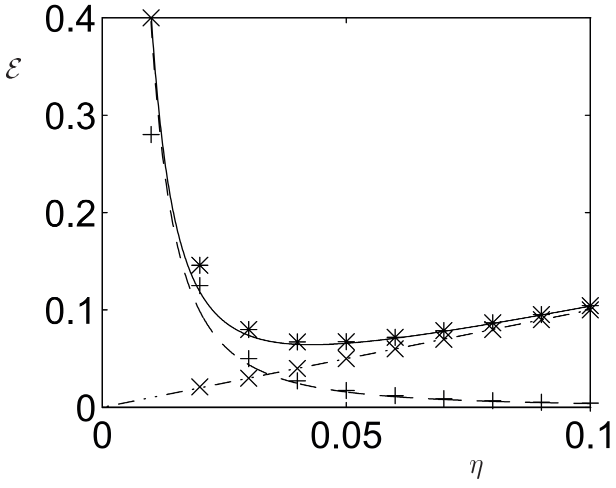

Figure 4: Misadjustment as a function of the learning parameter for Oja learning in a changing environment. Squared bias (computed, dashed line; simulated, +), variance (computed, dash-dotted line; simulated, x) and error (computed, solid line; simulated, $*$). Simulations were done with 5000 neural networks. Eigenvalues of the covariance matrix of the input distribution, $\Lambda_1 = 2.0$ and $\Lambda_2 = 1.0$. Angular velocity, $\omega = 2\pi/1000$.

The sum of these terms yields the misadjustment $\mathcal{E}$. Within this approximation, the minimum of the misadjustment is found for the optimal learning parameter

$$\eta_{\text{optimal}} = \left[\frac{2\omega^2}{(\Lambda_1 - \Lambda_2)\Lambda_1\Lambda_2}\right]^{1/3} .$$

The "theoretical" misadjustment is compared with results from simulations in figure 4. Especially in the vicinity of the optimal learning parameter, the approximations seem to work quite well.

5 Learning-parameter adjustment

5.1 Estimating the misadjustment

The method described above to calculate the optimal learning parameter looks simple and elegant and may work fine for the small examples discussed there, but is in practice useless since it requires detailed information about the environment (the diffusion and the curvature at the fixed point) that is usually not available. In this section we will point out how this information can be estimated from the statistics of the network weights and can be used to yield an autonomous algorithm for learning-parameter adaptation [29].

Suppose we have estimates for the bias and the variance, M_{estimate} and $\Sigma^2_{\text{estimate}}$, respectively, while learning with learning parameter η. We know that (in a gradually changing environment) the bias is inversely proportional to the learning parameter, whereas the variance is proportional to the learning parameter. So, with a new learning parameter η_{new}, our estimate for the misadjustment $\mathcal{E}$ is

$$\mathcal{E} = \frac{\eta^2}{\eta^2_{\text{new}}} M^2_{\text{estimate}} + \frac{\eta_{\text{new}}}{\eta} \Sigma^2_{\text{estimate}} . \tag{31}$$

Minimization of this misadjustment with respect to the new learning parameter η_{new} yields

$$\eta_{\text{new}} = \left[\frac{2M^2_{\text{estimate}}}{\Sigma^2_{\text{estimate}}}\right]^{1/3} \eta . \tag{32}$$

How do we obtain these estimates for the bias and the variance? First, we set the lefthandside of the evolution equations (30) equal to zero, i.e., we assume that the bias and the variance are more or less stationary. Then, to calculate the bias we must have an idea of the curvature H. To estimate it, we can use the asymptotic solution of equation (30) that relates the covariance matrix (the fluctuations in the network state) to the diffusion (the fluctuations in the learning rule). Since we can calculate both the diffusion and the covariance, we might try to solve the remaining matrix equation to compute the curvature. This seems to solve the problem but leads us directly to another one: solving an $N \times N$-matrix equation, where N is the number of weights, is computationally very expensive. Kalman-filtering, when applied to learning in neural networks [59], and other second-order methods for learning [5] have similar problems. Here, it seems even worse since we are only interested in updating *one global* learning parameter. Therefore, we will not consider all weights, but only a simple (global) function of the weights, e.g.,

$$W \stackrel{\text{def}}{=} \sum_{i=1}^{N} a_i w_i \, ,$$

with $\mathbf{a}$ a random vector that is kept fixed after it is chosen. During the learning process, we keep track of $\langle \Delta W \rangle$, $\langle \Delta W^2 \rangle$, $\langle W \rangle$, and $\langle W^2 \rangle$. From these averages, we can estimate a new learning parameter. The last problem concerns the averaging. In theory, the average must be over an ensemble of learning networks. Yet, it seems very unprofitable to learn with say 100 networks if one is just interested in the performance of one of them. Some authors do suggest to train an ensemble of networks for reasons of cross-validation [24], but although it would certainly improve the accuracy of the algorithm, it seems too much effort for simple learning-parameter adaptation. Instead, we estimate the averages by replacing the ensemble averages by time averages over a period T for the network that is trained. The time period T must be large enough to obtain accurate averages, but cannot be much larger than the typical time scale on which the diffusion, the curvature, or the "speed" changes significantly (see the discussion in section 4.3).

The final algorithm for learning-parameter adjustment consists of the following steps [29].

1. Gather statistics from learning with learning parameter η during time T, yielding $\langle W \rangle_T$, $\langle W^2 \rangle_T$, $\langle \Delta W \rangle_T$, and $\langle (\Delta W)^2 \rangle_T$.

2. Estimate the variance from

$$\Sigma^2_{\text{estimate}} = \left\langle W^2 \right\rangle_T - \langle W \rangle_T^2 - \frac{T^2 \langle \Delta W \rangle_T^2}{12} \, ,$$

   where the last term is a correction for the average change of W, and the bias from

$$M_{\text{estimate}} = - \frac{2 \langle \Delta W \rangle_T \Sigma^2_{\text{estimate}}}{\langle (\Delta W)^2 \rangle_T - \langle \Delta W \rangle_T^2} \, ,$$

   which can be obtained directly from the stationary solution of the evolution equations (30) for a one-dimensional system.

3. Calculate the new learning parameter η_{new} from equation (32).

5.2 Updating the learning parameter of a perceptron

As an example, we apply the adjustment algorithm to a perceptron [54] with two input units, one output unit, two weights (w_1 and w_2), and a threshold (w_0). The output of the network reads

$$y(\mathbf{w}, \vec{x}) = \tanh \left(\sum_{i=0}^{2} w_i x_i \right) \, ,$$

with the input vector $\vec{x} \overset{\text{def}}{=} (x_1, x_2)^T$ and $x_0 \equiv -1$. The learning rule is the so-called delta rule or Widrow-Hoff learning rule [66]

$$\Delta w_i \;=\; \eta \, [y_{\text{desired}} - y(\mathbf{w}, \vec{x})] \, [1 - y^2(\mathbf{w}, \vec{x})] \, x_i \; .$$

Backpropagation [55] is the generalization of this learning rule for neural networks with hidden units. The desired output y_{desired} depends on the class from which a particular input vector is drawn. There are two classes of inputs: "diamonds" corresponding to positive desired outputs $y_{\text{desired}} = 0.9$ and "crosses" corresponding to negative desired outputs $y_{\text{desired}} = -0.9$. We draw the input vectors $\vec{x}$ from Gaussian distributions with standard deviation σ around the center points $\vec{c}_{\pm} \overset{\text{def}}{=} \pm(\sqrt{2}\sin\phi, \sqrt{2}\cos\phi)^T$:

$$\rho(\vec{x}, y_{\text{desired}}) \;=\; \frac{1}{2} \sum_{+,-} \frac{1}{2\pi\sigma^2} \exp\left[\frac{|\vec{x} - \vec{c}_{\pm}|^2}{2\sigma^2}\right] \delta(y_{\text{desired}} \pm 0.9) \; .$$

In the optimal situation, the weights and the threshold yield a decision boundary going through the origin and perpendicular to the line joining the two center points. In other words, the fixed point solution $\mathbf{w}^*$ of the differential equation (27) corresponds to a decision boundary that is described by the line

$$x_1 \, \sin\phi + x_2 \, \cos\phi \;=\; 0 \; .$$

We can model learning in a changing environment by choosing a time-dependent angle $\phi(t)$, i.e., by rotating the center points.

Figures 5(a)-(c) show snapshots of the perceptron learning in a fixed, a suddenly changing, and a continuously changing environment, respectively. All simulations start with random weights, input standard deviation $\sigma = 1$, angle $\phi(0) = \pi/4$, a constant time window $T = 500$, and an initial learning parameter $\eta = 0.1$. After this initialization, the algorithm described in section 5.1 takes care of the recalibration of the learning parameter.

In a fixed environment [figure 5(a)], i.e., with a time-independent input probability density $\rho(y_{\text{desired}}, \vec{x})$, the weights of the network rapidly converge towards their optimal values. So, after a short while the bias is small and the decision boundary wiggles around the best possible separatrix. Then the algorithm decreases the learning parameter to reduce the remaining fluctuations. Theoretical considerations show that in a fixed environment the algorithm tends to decrease the learning parameter as [29]

$$\eta(t) \;\propto\; \frac{1}{t} \quad \text{for large } t,$$

which, according to the conditions (26) in section 3.2, is the fastest possible decay that can still guarantee convergence to the fixed point $\mathbf{w}^*$.

The second simulation [figure 5(b)] shows the response of the algorithm to a sudden change in the environment. The first 5000 learning steps are the same as in figure 5(a). But now the center points are suddenly displaced from $\phi = \pi/4$ to $\phi = -\pi/4$. This means that at time $t = 5000$ the decision boundary is completely wrong. The algorithm measures a larger bias, i.e., notices the "misadjustment" to the new environmental conditions, and raises the learning parameter. Psychologists might call this "arousal detection" (see e.g. [20]). It can be shown that, for this particular adjustment algorithm, the quickness of the response strongly depends on the learning parameter at the time of the change [29]. The lower the learning parameter, the slower the response. Therefore, it seems better to keep the learning parameter always above some lower bound, say $\eta_{\text{min}} = 0.001$, instead of letting it decrease to zero.

Figure 5(c) depicts the consequences of the algorithm in a gradually changing environment, the situation from which the algorithm was derived. In this simulation, we rotate the center points with a constant angular velocity $\omega = 2\pi/1000$. Simple theory, assuming perfect "noiseless" measurements, tells us that the learning parameter should decrease exponentially towards

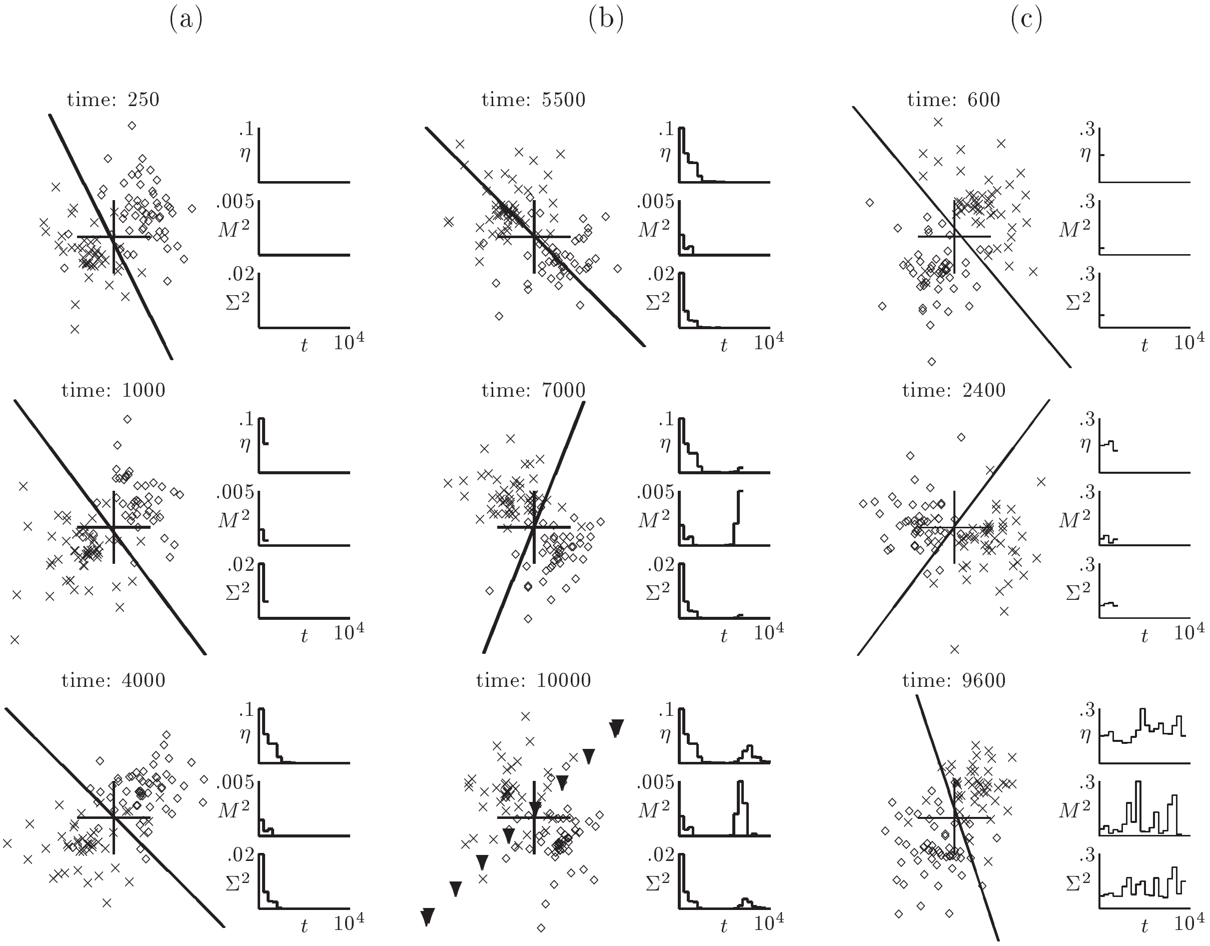

Figure 5: Learning-parameter adjustment for a perceptron. The last 150 training patterns are shown. Diamonds ($\diamond$) denote positive desired outputs, crosses ($\times$) negative desired outputs. The bold line shows the decision boundary found by the network. Graphs on the right give the learning parameter η, the squared bias M^2_{estimate}, and the variance $\Sigma^2_{\text{estimate}}$, all estimated from the statistics of the network weights. (a) A fixed environment: $\phi(t) = \phi(0) = \pi/4$. (b) A sudden change in the environment: $\phi(t)$ changes abruptly from $\pi/4$ to $-\pi/4$. (c) A continuously changing environment: $\phi(t) = \pi/4 + 2\pi t/1000$.

a constant "optimal" learning parameter [29]. In practice, the fluctuations are too large and the theory cannot be taken very seriously. Nevertheless, the pictures show that the overall performance is quite acceptable.

5.3 Learning of a learning rule

The algorithm described in section 5.1 and tested in section 5.2 is an example of the "learning of a learning rule" [3]. It shows how one can use the statistics of the weight variables to estimate a new learning parameter. This new learning parameter is found through minimization of the "expected misadjustment" [see equation (31)]. The underlying theory is valid for any learning rule of the form

$$\Delta\mathbf{w} \;=\; \eta\, \mathbf{f}(\mathbf{w},\vec{x}) \;,$$

which makes the algorithm widely applicable. Although originally designed for learning in changing environment, it also works fine in a fixed environment and in case of a sudden environmental change. The qualitative features of the algorithm (turning down the learning parameter if there is no new information, "arousal detection" in case of a sudden change) seem very natural from a biological and psychological point of view.

It is difficult to compare our algorithm with the many heuristic learning-rate adaptation algorithms that have been proposed for specific learning rules in a fixed environment (see e.g. [35] for a specific example or [5, 26] for reviews on learning-rate adaptation for backpropagation). Usually, these algorithms are based on knowledge of the whole error landscape and cannot cope with pattern-by-pattern presentation, let alone with a changing environment. Furthermore, most of these heuristic methods lack a theoretical basis, which does not necessarily affect the performance on the reported examples, but makes it very difficult to judge their "generalization capability", i.e., their performance on other (types of) problems.

The "learning of the learning rule" of Amari [3] is related to our proposal. Amari argues that the weight vector is far from optimal when two successive weight changes are (likely to be) in almost the same direction, whereas the weight vector is nearly optimal when two successive weight changes are (likely to be) in opposite directions. In our notation, this idea would yield an update of the learning parameter of the form (the original idea is slightly more complicated)

$$\eta(t+1) \;=\; \eta(t) + \gamma \, \frac{\Delta W(t)}{\eta(t)} \frac{\Delta W(t-1)}{\eta(t-1)} \;,$$

with $\Delta W(t) \overset{\text{def}}{=} W(t) - W(t-1)$ and γ a small parameter. The "learning of the learning rule" leads to the same kind of behavior as depicted in figures 5(a)-(c): "the rate of convergence automatically increases or the degree of accuracy automatically increases according to whether the weight vector is far from the optimal or nearly optimal" [3]. Amari's algorithm is originally designed with reference to a linear perceptron operating in a fixed environment, but might also work properly for a larger class of learning rules in a changing environment.

The more recent "search then converge" learning rate schedules of Darken et al. [11] are asymptotically of the form

$$\eta(t) \;\approx\; \frac{c}{t} \qquad \text{for large } t \,.$$

These schedules are designed for general learning rules operating in a fixed environment and guarantee convergence to a fixed point $\mathbf{w}^*$. The parameter c must be chosen carefully, since convergence is much slower for $c \leq c^*$ than for $c > c^*$, with c^* a usually unknown problem-dependent key parameter. To judge whether the parameter c is chosen properly, they propose to keep track of the "drift" F (again rewritten in our notation, their notation is slightly different and more elaborate)

$$F(t) \;\overset{\text{def}}{=}\; \left\langle \sqrt{s}\, \Delta W(s) \right\rangle^2_{T(t)} \;,$$

where the average is over the last T learning steps before time t. They argue that the "drift $F(t)$ blows up like a power of t when c is too small, but hovers about a constant value otherwise" [11]. This provides a signal for ensuring that c is large enough. Although not directly applicable to learning in a changing environment, it is another example of the idea to use the statistics of the weights for adaptation of the learning parameter. This general idea definitely deserves further attention and has great potential for practical applications.

6 Transition times between local minima

6.1 Context and state of the art

In the preceding sections, we have only discussed learning in the vicinity of one fixed point solution of the average learning dynamics. Learning rules with only one fixed point form a very limited class. Nowadays popular learning rules, such as backpropagation [55] and Kohonen learning [37], can have many fixed points. Some of these fixed points appear to be better than others. A well-defined measure for how good a particular network state $\mathbf{w}$ is, is the error potential $E(\mathbf{w})$. Often, one starts by defining an error potential, such as the (average) squared distance between the network's output and the desired output for backpropagation, and derives a learning rule from this error by calculating the gradient ∇ with respect to the network state $\mathbf{w}$ as in equation (17). With batch-mode learning, the network gets stuck in a minimum; in which minimum only depends on the initial network state. Many authors (see e.g. [5, 13, 24, 44]) share the feeling that random pattern presentation, i.e., on-line instead of batch-mode learning, introduces noise that helps to escape from "bad" local minima and favors lower lying minima. In this section, we will try to point out a theory that refines and quantifies these statements. We will restrict ourselves to learning rules for which equation (17) holds. Generalization to learning rules that cannot be derived from a global error potential is straightforward, except that there is no obvious, unbiased global measure of how good a network state is.

The results of section 2 give a purely local description of the stochastic process, i.e., the analysis yields unimodal distributions. This is a direct consequence of the "small-fluctuations Ansatz" (10). For an error potential with multiple minima, we obtain an approximate description around each minimum, but not a global description of a multimodal distribution. Standard theory on stochastic processes [12, 14, 63] cannot provide us with a general expansion method for unstable systems, i.e., stochastic systems with multiple fixed points. As we noted in section 2.2, the Fokker-Planck approximation, although often applied, does not offer an alternative since its validity is also restricted to the so-called attraction regions with positive curvature. Leen and Orr [44], for example, report simulations in which the Fokker-Planck approach breaks down even for extremely low learning parameters. Our approach [31] is based on two hypotheses which are supported by experimental and theoretical arguments. These hypotheses enable us to calculate asymptotic expressions for the transition times between different minima.

6.2 The hypotheses

Again, we start with the master equation (4) in a fixed environment. In section 2 we showed that in the attraction regions, where the Hessian $H(\mathbf{w})$ is positive definite, Van Kampen's system size expansion can be applied for small learning parameters η. Each attraction region contains exactly one minimum of the error $E(\mathbf{w})$. We say that minimum α lies inside attraction region A_α. $T_{\alpha\beta}$ stands for the transition region connecting attraction regions α and β. In the transition regions the Hessian has one negative eigenvalue. We can expand the probability density $P(\mathbf{w},t)$:

$$P(\mathbf{w},t) \;=\; \sum_{\alpha} P_\alpha(\mathbf{w},t) \;+\; \sum_{\alpha\beta} P_{\alpha\beta}(\mathbf{w},t)\;,$$

where $P_\alpha(\mathbf{w},t)$ is equal to $P(\mathbf{w},t)$ inside attraction region A_α and zero outside, and similar definitions for $P_{\alpha\beta}(\mathbf{w},t)$ in the transition regions[5]. For proper normalization, we define the occupation numbers

$$n_\alpha(t) \overset{\text{def}}{=} \int_{A_\alpha} d^N w\; P(\mathbf{w},t)\;,$$

i.e., the occupation number $n_\alpha(t)$ is the probability mass in attraction region A_α. From the master equation (4), we would now like to extract the evolution of these occupation numbers $n_\alpha(t)$.

Figure 6 shows the histogram of 10000 independently learning one-dimensional networks at three different times (see [31] for details). We use this simple example to give an idea of the evolution of the master equation in the presence of multiple minima and to point at a few characteristic properties of unstable stochastic systems (see [63]). The learning networks

[5] We neglect the probability mass outside the attraction and transition regions since it is negligible if compared with the probability mass inside these regions and has no effect on our calculation of transition times anyway.

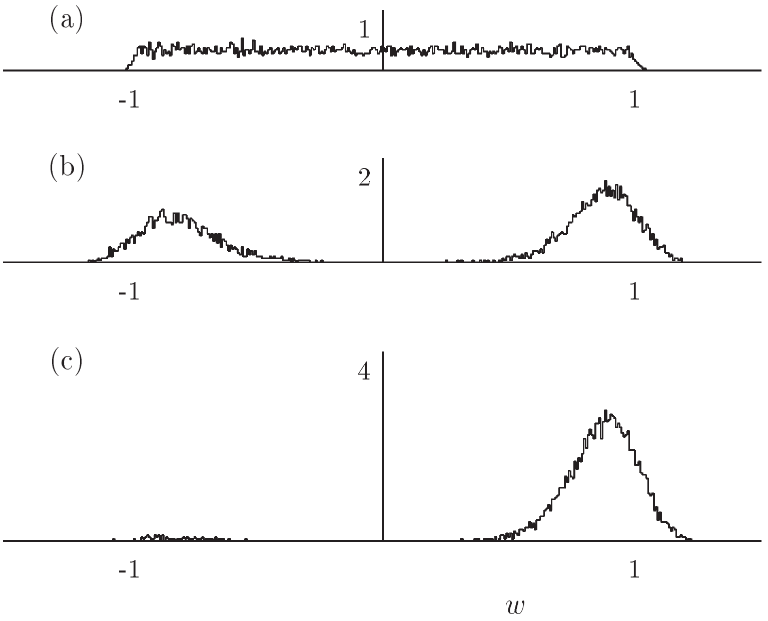

Figure 6: Histogram found by simulation of 10000 one-dimensional neural networks learning on an error potential with a local and a global minimum. (a) $t = 1$: Initial distribution. (b) $t = 10^3$: Two peaks. (c) $t = 10^6$: Stationary distribution.

perform stochastic gradient descent on a one-dimensional error potential with a local minimum at $w \approx -1$ and a global minimum at $w \approx 1$. The weights are initialized with equal probability between -1 and 1 (figure 6(a): $t = 1$). On a time scale of order $1/\eta$, the local relaxation time $\tau_{\rm local}$ in equation (19), $P(w, t)$ evolves to a distribution with peaks at the two minima (figure 6(b): $t = 10^3$). The probability mass in the transition region is much smaller than the probability mass in the attraction regions: transitions between the minima are very rare. The global relaxation time to the equilibrium distribution (figure 6(c): $t = 10^6$) is much larger than the local relaxation time.

Our first hypothesis is well-known in the theory of unstable stochastic processes [63]. It says that the rare transitions may affect the probability *mass*, but not the *shape* of the distribution in the attraction regions. In other words, we assume that after the local relaxation time, we are allowed to "decouple time and space" in the attraction regions:

$$P_\alpha(\mathbf{w}, t) \;=\; n_\alpha(t)\, p_\alpha(\mathbf{w}) \;.$$

This assumption seems to be valid when the attraction regions are well separated and when the transitions between them are rare. Substitution of this assumption into the master equation yields

$$\begin{aligned} \frac{dn_\alpha(t)}{dt} \;&=\; -\sum_\beta \left[\int_{T_{\alpha\beta}} d^N w' \int_{A_\alpha} d^N w \; T(\mathbf{w}'|\mathbf{w})\, p_\alpha(\mathbf{w}) \right] n_\alpha(t) \\ &\quad + \int_{A_\alpha} d^N w' \sum_\beta \int_{T_{\alpha\beta}} d^N w \; T(\mathbf{w}'|\mathbf{w})\, P_{\alpha\beta}(\mathbf{w}, t) \;. \end{aligned}$$

The first term in this equation corresponds to probability mass leaving attraction region A_α, the second term to probability mass entering A_α.

Let us concentrate on the first term alone and neglect the second term. This corresponds to a simulation in which all networks that leave the attraction region A_α are taken out. The term between brackets is the probability per unit time to go from attraction region A_α to transition region $T_{\alpha\beta}$. The inverse of this term is called the transition time $\tau(A_\alpha \to T_{\alpha\beta})$ from attraction region A_α to transition region $T_{\alpha\beta}$:

$$\tau(A_\alpha \to T_{\alpha\beta}) \;=\; \left[\int_{T_{\alpha\beta}} d^N w' \int_{A_\alpha} d^N w \, T(\mathbf{w}'|\mathbf{w})\, p_\alpha(\mathbf{w})\right]^{-1} . \tag{33}$$

Below we will sketch how to calculate this transition time for small learning parameters η. We will show that it is of the form

$$\tau(A_\alpha \to T_{\alpha\beta}) \;\sim\; \exp\left[\frac{\tilde{\eta}_{\beta\alpha}}{\eta}\right] \qquad \text{for small } \eta,$$

with $\tilde{\eta}_{\beta\alpha}$, the so-called reference learning parameter, a constant independent of the learning parameter η. If the learning parameter is chosen much smaller than the reference learning parameter, the probability to go from the attraction to the transition region within a finite number of learning steps is negligible. Furthermore, the reference learning parameters play an important role in the derivation of cooling schedules that guarantee convergence to the global minimum (see section 8).

So, we can compute how the transition time $\tau(A_\alpha \to T_{\alpha\beta})$ from the attraction region to the transition region scales as a function of the learning parameter η. But we are more interested in the transition time $\tau(A_\alpha \to A_\beta)$ from attraction region A_α to attraction region A_β, i.e., the average time it takes to get *over* transition region $T_{\alpha\beta}$. What happens in this transition region? In the transition regions the small-fluctuations expansion of section 2.3 is not valid. If we still try to apply it, we notice that (in this approximation scheme) the fluctuations tend to explode [see equation (13)]. On the other hand, in the attraction regions the (asymptotic) fluctuations are proportional to the learning parameter. The idea is now that, for small learning parameters η, the transition time from attraction region A_α to A_β is dominated by the transition time from A_α to transition region $T_{\alpha\beta}$. More specifically, our second hypothesis states that

$$\lim_{\eta\to 0} -\eta \, \ln \tau(A_\alpha \to A_\beta) \;\approx\; \lim_{\eta\to 0} -\eta \, \ln \tau(A_\alpha \to T_{\alpha\beta}) \;=\; \tilde{\eta}_{\beta\alpha} \,,$$

i.e., that the reference learning parameter for the total transition from one attraction region to another can be estimated by calculating the reference learning parameter for the transition from the attraction region to the transition region.

6.3 Calculation of the reference learning parameter

In this section we will sketch how to calculate the reference learning parameter

$$\tilde{\eta}_{\beta\alpha} \;=\; -\lim_{\eta\to 0} \eta \ln \left\{ \int_{T_{\alpha\beta}} d^N w' \int_{A_\alpha} d^N w \int d^n x \; \rho(\vec{x})\; \delta^N(\mathbf{w}' - \mathbf{w} - \eta \mathbf{f}(\mathbf{w}, \vec{x}))\, p_\alpha(\mathbf{w}) \right\} \tag{34}$$

for the transition from attraction region A_α to transition region $T_{\alpha\beta}$. We recall from section 2.4 that the local probability distribution $p_\alpha(\mathbf{w})$ can be approximated by a Gaussian with its average at the minimum $\mathbf{w}^*_\alpha$ and variance $\Sigma^2_\alpha = \eta K_\alpha$ obeying

$$H_\alpha K_\alpha + K_\alpha H_\alpha \;=\; D_\alpha \,, \tag{35}$$

where the Hessian $H_\alpha \overset{\text{def}}{=} H(\mathbf{w}^*_\alpha)$ and the diffusion matrix $D_\alpha \overset{\text{def}}{=} D(\mathbf{w}^*_\alpha)$ are both evaluated at the minimum $\mathbf{w}^*_\alpha$.

In equation (34), we have to integrate over all $\mathbf{w}$ and $\vec{x}$ such that

$$\mathbf{w} \in A_\alpha \quad \text{and} \quad \mathbf{w}' = \mathbf{w} + \eta \mathbf{f}(\mathbf{w}, \vec{x}) \in T_{\alpha\beta} \, .$$

So[6], both $\mathbf{w}$ and $\mathbf{w}'$ are within order η of the boundary $B_{\beta\alpha}$ between attraction region A_α and transition region $T_{\alpha\beta}$. Now it is easy to prove [31] that, for small learning parameters η, the integral in (34) converges to an integral over the boundary $B_{\beta\alpha}$ times some term of order η. This latter term disappears if we take the logarithm, multiply with η, and take the limit $\eta \rightarrow 0$. Finally, in the limit $\eta \rightarrow 0$, the only remaining term is

$$\tilde{\eta}_{\beta\alpha} \;=\; -\lim_{\eta \rightarrow 0} \eta \ln \left\{ \int_{B_{\beta\alpha}} d^{N-1}w \, \exp \left[-\frac{(\mathbf{w} - \mathbf{w}^*_\alpha)^T K_\alpha^{-1} (\mathbf{w} - \mathbf{w}^*_\alpha)}{2\eta} \right] \right\} .$$

The integral can be approximated using the method of steepest descent. The largest contribution is found when the term between brackets is maximal on the boundary $B_{\beta\alpha}$. So, the largest contribution comes from the "easiest" path from the local minimum $\mathbf{w}^*_\alpha$ to the transition region $T_{\alpha\beta}$. The matrix K_α^{-1} defines the local "metric". The final result is

$$\tilde{\eta}_{\beta\alpha} \;=\; \inf_{\mathbf{w} \in B_{\beta\alpha}} \left[\frac{(\mathbf{w} - \mathbf{w}^*_\alpha)^T K_\alpha^{-1} (\mathbf{w} - \mathbf{w}^*_\alpha)}{2} \right] . \tag{36}$$

Roughly speaking, the reference learning parameter is proportional to the height of the error barrier and inversely proportional to the local fluctuations. The result is similar to the classical Arrhenius factor for unstable stochastic (chemical) processes [63]. In the next section we will apply this formula to calculate the reference learning parameter for the transition from a twist ("butterfly") to a perfectly ordered configuration in a self-organizing map.

7 Unfolding twists in a self-organizing map

7.1 Twists are local minima of an error potential

The Kohonen learning rule [37, 38] tries to capture important features of self-organizing processes. It has not only applications in robotics, data segmentation, and classification tasks, but may also help to understand the formation of sensory maps in the brain. In these maps, the external information is represented in a topology-preserving manner, i.e., neighboring units code similar input signals. Properties of the Kohonen learning procedure have been studied in great detail [10, 52]. Most of these studies focussed on the convergence properties of the learning rule, i.e., asymptotic properties of the learning network in a perfectly ordered configuration. In this context, Ritter and Schulten [51, 53] were the first to use the master equation for a description of on-line learning processes.

It is well-known that not only perfectly ordered configurations, but also topological defects, like kinks in one-dimensional maps or twists in two-dimensional maps, can be fixed point solutions of the learning dynamics [16]. With a slight change, the Kohonen learning rule can be written as the gradient of a global error potential [30]. Then the topological defects correspond to local minima of this error potential, whereas global minima are perfectly ordered configurations. The unfolding of a twist in a two-dimensional map is now simply a transition from a local minimum to a global minimum. Using the theory developed in section 6, we will calculate the reference learning parameters for these transitions and compare them with straightforward simulations of the learning rule.

As an example, we consider a network of 4 units. Each unit has a two-dimensional weight vector, so, the total eight-dimensional network state vector is written $\mathbf{w} = (\vec{w}_1, \ldots, \vec{w}_4)^T = (w_{11}, w_{12}, w_{21}, \ldots, w_{42})^T$. Each learning iteration consists of the following steps.

[6] For simplicity, we will only consider the case in which the learning rule is bounded. i.e., for which there exists an $M < \infty$ such that $|\mathbf{f}(\mathbf{w}, \vec{x})| < M$, for all $\mathbf{w}$ and all $\vec{x} \in \Omega$.

1. An input $\vec{x} = (x_1, x_2)^T$ is drawn with equal probability from a square:

$$\rho(x_1, x_2) \; = \; \frac{1}{4}\theta(1+x_1)\,\theta(1-x_1)\,\theta(1+x_2)\,\theta(1-x_2)\,.$$

2. The "winning unit" is the unit with the smallest local error

$$e_i(\mathbf{w}, \vec{x}) \; = \; \frac{1}{2}\sum_j h_{ij}|\vec{w}_j - \vec{x}|^2\,.$$

Here h is called the lateral-interaction matrix. The closer two units i and j in the "hardware" network configuration, the stronger the lateral interaction h_{ij}. We choose it of the form

$$h \; = \; \frac{1}{(1+\sigma)^2}\begin{pmatrix} 1 & \sigma & \sigma^2 & \sigma \\ \sigma & 1 & \sigma & \sigma^2 \\ \sigma^2 & \sigma & 1 & \sigma \\ \sigma & \sigma^2 & \sigma & 1 \end{pmatrix},$$

with $0 \leq \sigma < 1$ the so-called lateral-interaction strength. $\sigma = 0$ means no lateral interaction. Which unit "wins" depends on the network state $\mathbf{w}$ and on the particular input vector $\vec{x}$. We will denote the winning by $\kappa(\mathbf{w}, \vec{x})$ or just κ.

3. The weights are updated with

$$\Delta w_{i\alpha} \; = \; \eta\, f_{i\alpha}(\mathbf{w}, \vec{x}) \; = \; -\eta\,\frac{\partial e_\kappa(\mathbf{w}, \vec{x})}{\partial w_{i\alpha}} \; = \; \eta\, h_{\kappa i}(x_\alpha - w_{i\alpha})\,. \qquad (37)$$

So, in principal all weights are moved towards the input vector. To what extent depends on the lateral interaction between the particular unit and the winning unit.

Equation (37) is exactly the Kohonen learning rule. The difference is step 2: the determination of the winning unit. In Kohonen's procedure the winner is the unit with the smallest Euclidian distance to the input vector. We propose to determine the winning unit on account of the local error $e_i(\mathbf{w}, \vec{x})$, the *same* error that is differentiated to yield the learning rule (37). Then, and only then, it can be shown [27, 30] that this learning procedure performs (stochastic) gradient descent on the global error potential[7]

$$E(\mathbf{w}) \; = \; \left\langle e_{\kappa(\mathbf{w},\vec{x})}(\mathbf{w}, \vec{x})\right\rangle_\Omega\,.$$

For $\sigma = 0$ the local error $e_i(\mathbf{w}, \vec{x})$ is just the Euclidian distance between the weight $\vec{w}_i$ and the input $\vec{x}$ which makes both learning procedures totally equivalent.

Careful analysis shows that, for $0 < \sigma < \sigma^* \approx 0.240$, the error potential has $4! = 24$ different possible minima: 8 global minima and 16 local minima. To visualize these network states, we draw lines between the positions of the (two-dimensional) weight vectors of neighboring units, i.e., between 1-2, 2-3, 3-4, and 4-1. As can be seen in figure 7(a), the global minima correspond to perfectly ordered configurations. They are called "rectangles". The "twist" or "butterfly" in figure 7(b) is an example of a topological defect: a local minimum. For $\sigma = 0$, i.e., no interaction, all minima are equally deep. At $\sigma = \sigma^*$ the local minima, representing twists, disappear and only global minima, representing rectangles, remain.

[7] The gradient of $E(\mathbf{w})$ consists of two parts: the differentiation of the local error and the differentiation of the "winner-take-all mechanism". This latter term, which is the most difficult one, exactly cancels if and only if the "winner" is determined on account of the local errors $e_i(\mathbf{w}, \vec{x})$ [30].

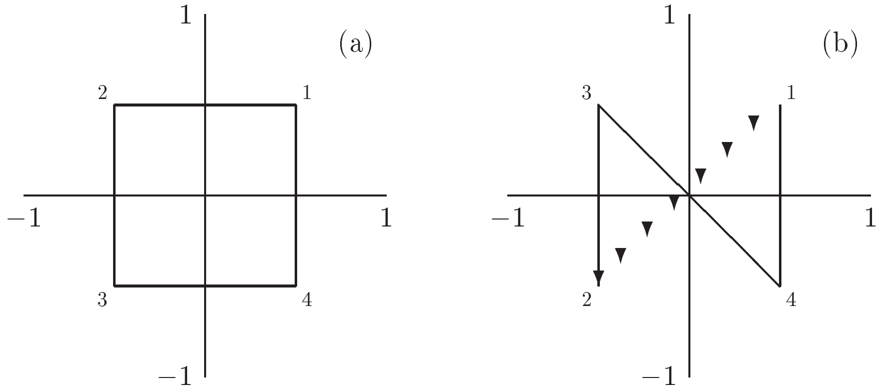

Figure 7: Configurations in a two-dimensional map. (a) Rectangle. (b) Twist.

7.2 Theory versus simulations

We will calculate the reference learning parameter $\tilde{\eta}$ for the transition from the local to the global minimum, i.e., from a twist to a rectangle, for different values of σ. This reference learning parameter tells us how the average time needed to unfold a twist scales as a function of the learning parameter η. We go through the following steps.

1. Choose the lateral-interaction strength σ.

2. Determine the position of the local minimum $\mathbf{w}^*$, i.e., the exact network weights of the twist in figure 7(b).

3. Calculate the Hessian H and the diffusion matrix D at this minimum from equation (16) and (15), respectively.

4. Solve equation (35) to find the covariance matrix K and its inverse K^{-1}.

5. On the boundary between the attraction and the transition region, the determinant of the Hessian of the error potential $E(\mathbf{w})$ is exactly zero. Find the point $\mathbf{w}$ on this boundary with the smallest distance $(\mathbf{w} - \mathbf{w}^*)^T K^{-1}(\mathbf{w} - \mathbf{w}^*)$.

6. Compute the reference learning parameter $\tilde{\eta}(\sigma)$ from equation (36).

The fifth step, optimization under the awkward constraint that the determinant of the Hessian matrix must be zero, is by far the most difficult one. For larger problems, i.e., a higher dimensional weight space, this may become too difficult. The solid line in figure 8 gives the "theoretically" obtained reference learning parameter as a function of the strength σ.

Straightforward simulations of the learning procedure are used for comparison. For each choice of the interaction strength σ, we train 500 independently operating networks for 4 different learning parameters. For each learning parameter, we determine the transition time $\tau(\eta)$. The reference learning parameter $\tilde{\eta}$ follows from the best possible fit of the form

$$\ln \tau(\eta) \;=\; \tilde{\eta}\eta^{-1} + d \ln \eta^{-1} + c\,.$$

The reference learning parameters $\tilde{\eta}(\sigma)$ obtained in this way are indicated by an asterix in figure 8. The theoretically obtained reference learning parameters are somewhat smaller than the ones obtained from straightforward simulations. This might be due to the neglect of the transition region.

In [27] we also try to calculate the transition times for the transition from a "kink", a topological defect in a one-dimensional map, to a "line", a perfectly ordered configuration. Again,

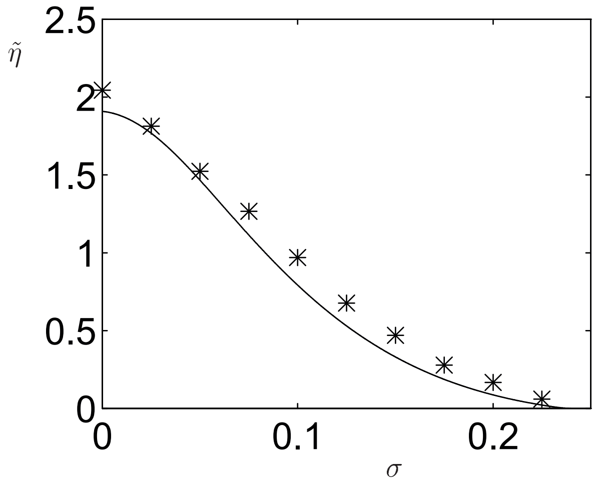

Figure 8: The reference learning parameter $\tilde{\eta}$ as a function of the lateral-interaction strength σ. Solid lines show the theoretically obtained results. Simulation results are indicated by an "$*$".

this is a transition from a local minimum, the kink, to a global minimum, the line. For small σ, when this transition becomes very improbable (for $\sigma = 0$ the dynamics of the learning rule is such that a kink cannot be removed), the reference learning parameters predicted by theory do no longer agree with the results obtained from simulations. A possible explanation is the violation of the first assumption explained in section 6.2: in the limit $\sigma \to 0$ the transition region, which normally acts as a buffer between the two attraction regions, vanishes and the assumption that transitions only affect the masses and not the shapes of the probability distributions in the attraction regions is no longer valid. Further study is necessary to solve this problem.

In all this, we must not forget that, if we really want to calculate the reference learning parameter, detailed knowledge about the environment and the network structure is needed. The same notion came up in section 4, where we tried to calculate the optimal learning parameter for learning in a changing environment. To a certain extent we could solve this problem in section 5 by considering the statistics of the weights. Here it is much more difficult, since we need to extract *global* information from the network dynamics. A solution might be a pre-learning phase, similar to the ones proposed for simulated annealing processes (see e.g. [1]).

8 On-line learning and global optimization

8.1 The analogy with simulated annealing

On-line learning is a stochastic process. The "intrinsic noise" due to the random pattern presentation enables transitions between different minima. The larger the learning parameter, the greater this noise, so the easier the transitions. We might compare this with simulated annealing [4, 36] or Langevin equations [17, 21] (see also section 3.1). In the simulated annealing approach a candidate $\mathbf{w}$ is picked at random according to some "generating probability function". The error $E(\mathbf{w}')$ of the candidate $\mathbf{w}'$ is compared with the error $E(\mathbf{w})$ of the current state $\mathbf{w}$. Downhill steps are always accepted, uphill steps are accepted with a probability proportional to

$$\exp\left[-\frac{E(\mathbf{w}') - E(\mathbf{w})}{T}\right] .$$

The noise parameter T is called the temperature. Using this dynamics, it can be shown that after sufficient time, the probability distribution $P(\mathbf{w}, t)$ resembles a Gibbs distribution. With a proper cooling schedule, i.e., a smart choice for the temperature as a function of time, convergence to the global optimum can be guaranteed for these processes. Can we draw an analogy to on-line learning in neural networks, even when simulated annealing is really different from the learning procedure (1)? Or more specifically, how should we choose our noise parameter, the learning parameter, to get the fastest possible convergence to the global minimum? Starting from the transition times derived in section 6, we will try to answer these questions.

8.2 Derivation of a cooling schedule

For simplicity, we will first consider a two-level system with one global minimum $E_1 \equiv E(\mathbf{w}_1^*)$ and one local minimum $E_2 \equiv E(\mathbf{w}_2^*)$. The average error potential $E(t)$ is defined

$$E(t) \stackrel{\text{def}}{=} \langle E(\mathbf{w}) \rangle_{\Xi(t)} \; = \; n_1(t) E_1 + n_2(t) E_2 + \mathcal{O}(\eta) \; = \; E_1 + n_2(t)(E_2 - E_1) + \mathcal{O}(\eta) \; , \qquad (38)$$

where we use $n_1(t) + n_2(t) = 1$, i.e., we neglect the probability mass in the transition region. This is correct for times t much larger than the local relaxation time of order $1/\eta$ (see the discussion in section 6.2). Then the probability distribution $P(\mathbf{w}, t)$ is strongly peaked in the vicinity of the minima of the error potential. The variance of these local probability distributions is of order η and thus the average error potential of the networks in the vicinity of a particular minimum $\mathbf{w}^*$

$$\langle E(\mathbf{w}) \rangle_{\Xi(\infty)} - E(\mathbf{w}^*) \; \approx \; \frac{1}{2} \Big\langle (\mathbf{w} - \mathbf{w}^*)^T H (\mathbf{w} - \mathbf{w}^*) \Big\rangle_{\Xi(\infty)} \; \approx \; \text{Tr} \; [H \Sigma^2(\infty)] \; \approx \; \eta \text{Tr} \; D \; ,$$

is also of order η. For the moment, we will neglect this term. It will only play a significant role when either $n_1(t)$ or $n_2(t)$ becomes of order η.

The occupation number $n_2(t)$ obeys the differential equation

$$\frac{d n_2(t)}{dt} \; = \; - \frac{n_2(t)}{\tau_{12}} + \frac{n_1(t)}{\tau_{21}} \; , \qquad (39)$$

with transition time τ_{12} for the transition from attraction region A_2 to A_1 of the form (see section 6)

$$\tau_{12} \; \sim \; \exp \left[\frac{\tilde{\eta}_{12}}{\eta} \right] \qquad \text{for small } \eta \; , \qquad (40)$$

and similarly for τ_{21}. From (38) and (39), we can derive a differential equation for the average error $E(t)$:

$$\frac{dE(t)}{dt} \; = \; - \left[\frac{E(t) - E_1}{\tau_{12}} - \frac{E_2 - E(t)}{\tau_{21}} \right] \; .$$

We would now like to choose the learning parameter η as a function of time t such that the average error potential $E(t)$ decreases as fast as possible, i.e., to choose $\eta(t)$ such that the term between brackets is as large as possible [58]:

$$[E(t) - E_1] \, \frac{d}{d\eta(t)} \left[\frac{1}{\tau_{12}(\eta(t))} \right] \; = \; [E_2 - E(t)] \, \frac{d}{d\eta(t)} \left[\frac{1}{\tau_{21}(\eta(t))} \right] \; .$$

This defines a relationship between $E(t)$ and $\eta(t)$, which can be used to transform the differential equation for $E(t)$ into a differential equation for the time trajectory of the optimal $\eta(t)$. Using the form (40), we obtain [for small learning parameters $\eta(t)$]

$$\frac{d\eta(t)}{dt} \; = \; -\eta^2(t) \, \left[\frac{1}{\tilde{\eta}_{21} \tau_{12}(\eta(t))} + \frac{1}{\tilde{\eta}_{12} \tau_{21}(\eta(t))} \right] \; .$$

Now, suppose that the transition from the local to the global minimum is "easier" than vice versa, i.e., has a shorter transition time and thus a smaller reference learning parameter[8]. Then

[8] As we will argue in section 8.3, a transition from a higher minimum to a lower minimum is in almost all cases easier than vice versa. If the reverse is true, then the local minimum is the "most attractive" minimum and, by replacing $\tilde{\eta}_{12}$ for $\tilde{\eta}_{21}$ in what follows, we can only guarantee convergence to this minimum.

we can neglect the second term between brackets if compared with the first term. For large t, the lowest order solution of the remaining differential equation yields [32]

$$\eta(t) \;=\; \frac{\tilde{\eta}_{12}}{\ln t} \;+\; \mathcal{O}\left[\frac{\ln\ln t}{(\ln t)^2}\right] \;. \tag{41}$$

This constitutes our final optimal cooling schedule. It only depends on the reference learning parameter $\tilde{\eta}_{12}$ for the transition from the local to the global minimum.

In a sense, the derived cooling schedule is indeed optimal. A "faster" cooling schedule, e.g. $\eta(t) = \tilde{\eta}_{12}/5 \ln t$, cannot guarantee that a network starting at the local minimum will indeed reach the global minimum. We could say that the transition from the local to the global minimum is "closed". The optimal cooling schedule keeps this transition just "open". A "slower" cooling schedule, e.g. $\eta(t) = 5\,\tilde{\eta}_{12}/\ln t$, gives also an open transition, but convergence will take longer than with the optimal cooling schedule. By looking at the transition times we can easily check whether a particular transition is open or closed. If the transition time grows at most linearly with time t the transition is open, if it grows faster than linearly with time t the transition is closed. For the optimal cooling schedule (41) the transition time τ_{12} from the local to the global minimum grows linearly with time t.

Generalization to more minima is tedious. Nevertheless, the final cooling schedule is of the same form [32]

$$\eta(t) \;=\; \frac{\eta^*}{\ln t} \qquad \text{for large } t\;,$$

The optimal η^* depends on the reference learning parameters between the various minima. It is bounded by [32]

$$\tilde{\eta}_{\rm min} \;\leq\; \eta^* \;\leq\; \tilde{\eta}_{\rm min} + (M-1)(\tilde{\eta}_{\rm max} - \tilde{\eta}_{\rm min})\;,$$

with $\tilde{\eta}_{\rm min}$ and $\tilde{\eta}_{\rm max}$ the smallest and the largest finite reference learning parameter, respectively, and M the number of minima. This kind of "exponentially slow" cooling schedule is common ground in the theory of stochastic processes for global optimization [17, 36][9]. In cooling schedules for simulated annealing the optimal η^* is called "the critical depth" [8]. It is the depth (suitably defined) of the deepest local minimum which is not a global minimum state [22]. In this context, the approach taken in [62] is most similar to ours: the critical depth is computed from the structure of a Markov chain, i.e., from transition probabilities between different states. Neither we, nor other authors, claim that it is easy to calculate the optimal parameter η^* for practical optimization problems. We only try to give an intuitive feeling of the factors that determine this parameter.

8.3 Global optimization and on-line backpropagation

In this last section we will discuss an example of on-line backpropagation with profound local minima. The structure of the network is depicted in figure 9(a). There are 6 synapses and 3 thresholds, so, $N = 9$ adaptive elements. These elements are combined in the weight vector $\mathbf{w} = (w_{10}, w_{11}, w_{12}, w_{20}, w_{21}, w_{22}, w_{30}, w_{31}, w_{32})^T$. The network has 2 variable inputs: x_1 and x_2. Thresholds are incorporated by defining $x_0 = y_0 = -1$. The outputs y_1 and y_2 of the hidden units are given by

$$y_i \;=\; \tanh\left[\sum_{j=0}^{2} w_{ij}x_j\right]\;,$$

[9] There is a method called fast simulated annealing [60, 61] based on a cooling schedule that decreases with $1/t$. The difference is the use of a Cauchy distribution (with an infinite variance!) instead of a Gaussian distribution (with a finite variance which is more similar to on-line learning processes) for the generation of new states.

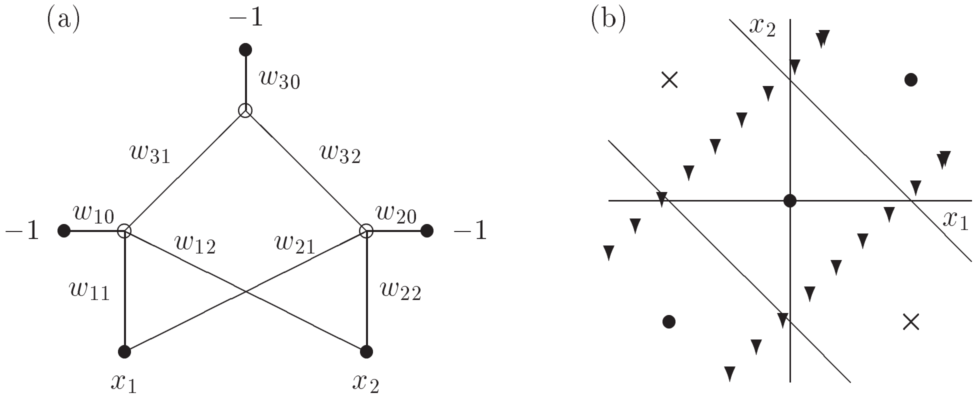

Figure 9: (a) Network structure. (b) XOR problem.

the output y_3 of the network by

$$y_3 \;=\; \tanh\left[\sum_{j=0}^{2} w_{3j} y_j\right] \,.$$

The goal of the backpropagation learning rule is to minimize the quadratic error potential [55]

$$E_0(\mathbf{w}) \;=\; \frac{1}{2p}\sum_{\mu=1}^{p}\left[y_3(\mathbf{w},x_1^\mu,x_2^\mu) - x_3^\mu\right]^2 \;\stackrel{\text{def}}{=}\; \frac{1}{p}\sum_{\mu=1}^{p} E_0(\mathbf{w},\vec{x}^\mu) \,, \tag{42}$$

where the sum is over all p training patterns, indicated by three-dimensional vectors $\vec{x}^\mu = (x_1^\mu, x_2^\mu, x_3^\mu)^T$. The components x_1^μ and x_2^μ give the input values of the network for pattern μ, the component x_3^μ the desired output value. We will use desired output values of ±0.8 instead of ±1 to prevent divergence of the weights. Rather than minimizing the error (42), it is often convenient to minimize an error of the form

$$E(\mathbf{w}) \;=\; E_0(\mathbf{w}) + \lambda E_1(\mathbf{w}) \,, \tag{43}$$

with $E_1(\mathbf{w})$ an extra term, the so-called bias [25] (not to be confused with the bias $\mathbf{m}$ of sections 2, 4, and 5). We will use the bias

$$E_1(\mathbf{w}) \;=\; \frac{1}{4}\sum_{i=0}^{2}\sum_{j=0}^{2}\left[w_{ij}^2 - \alpha\right]^2 \,, \tag{44}$$

with $\alpha = 0.1$ and $\lambda = 0.01$. Incorporation of this bias has a few advantages among which there are prevention of local minima with infinite weights and reduction of training times [39].

After [18], we choose the set of $p = 5$ training patterns sketched in figure 9(b). Circles indicate negative output, crosses positive output. This is just the usual XOR truth table with one additional pattern at the origin. Because of this additional pattern, the error potential (43) has not only global minima, but also profound local minima[10]. The thick lines in figure 9(b) show the separation lines of the hidden units that lead to the optimal solution. At the global minima all five training patterns are correctly classified. The thin lines give the separation lines corresponding to the local minima. At the local minima only four patterns are correctly classified. For symmetry reasons there are 8 local and 8 global minima.

We will compare the optimization capabilities of the following two learning procedures.

[10] At the local minima in the original XOR problem, carefully analyzed in [46], at least one of the weights is either infinite or zero. After incorporation of the bias (44), we did not encounter any of these "stupid" minima.

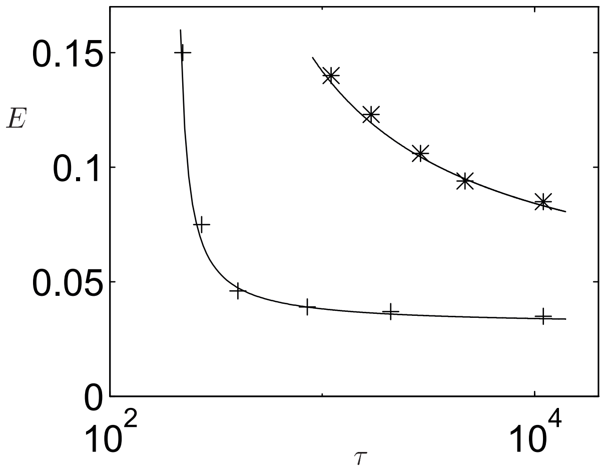

Figure 10: Asymptotic performance E versus transition time τ for on-line learning ($+$) and Langevin-type learning ($*$). The lines serve to guide the eye.

1. At each learning step, one of the patterns, say ν, is drawn at random from the set of 5 patterns, and a learning step is performed:

   $$\Delta \mathbf{w} \; = \; -\eta \, \nabla \left[E_0(\mathbf{w}, \vec{x}^\nu) + \lambda E_1(\mathbf{w}) \right] \; .$$

   This, of course, is an on-line learning process of the type discussed in this chapter.

2. Artificial noise is added to the gradient of the *total* error potential, averaged over all training patterns:

   $$\Delta \mathbf{w} \; = \; -\nabla \left[E_0(\mathbf{w}) + \lambda E_1(\mathbf{w}) \right] \; \Delta t \; + \; \sqrt{2T} \, \boldsymbol{\xi} \, \sqrt{\Delta t} \, ,$$

   with $\boldsymbol{\xi}$ noise of variance 1. This is called "Langevin-type learning"; it is a discretized version of the Langevin equation (see section 3.1). We will choose $\Delta t = 1$.

For both learning procedures we take an ensemble of 1000 independently operating neural networks, all starting at a local minimum. We train this ensemble for a few different values of η and T. From the dynamics of the occupation numbers at the local and global minima, we measure the transition times $\tau(\eta)$ and $\tau(T)$. Besides this, we collect the average error potential at the stationary situation, so, for very long learning times. These are denoted $E(\eta)$ and $E(T)$. The average error E can be viewed as a measure of the asymptotic performance of the learning procedure, the transition time τ as the typical time to reach it. As can be seen from figure 10, where the asymptotic performance E is plotted as a function of the transition time τ, on-line learning is highly preferable above Langevin-type learning: the same transition time yields a much better asymptotic performance for on-line learning than for Langevin-type learning.

The inhomogeneous intrinsic noise due to the random pattern presentation explains the better performance of on-line learning processes. For Langevin-type learning, the noise is homogeneous, i.e., the same at each minimum, whereas for on-line learning the noise is related to the diffusion D, the fluctuations in the learning rule, which is a function of the weights. Usually we will have that the higher the error potential, the more there is to learn, the larger the fluctuations in the learning rule, the higher the noise level, and the easier to escape. Roughly speaking, the reference learning parameter for a transition from minimum α to β is proportional to the height of the barrier between α and β and inversely proportional to the local fluctuations at α. In the backpropagation example of this section, the reference learning parameter for the transition from the global to the local minimum is much larger than the reference learning parameter for the reverse transition, whereas the "reference temperature" for both transitions is of the same order of magnitude. This explains the form of figure 10.

Generalization of these arguments suggests that the inhomogeneous noise coming from the random presentation of patterns in on-line learning processes *helps* to find the global minimum. The comparison made above is just a simplistic and specific example, but it gives a nice idea of the usefulness of on-line learning if compared with other optimization techniques.

Acknowledgment

This work was partly supported by the Dutch Foundation for Neural Networks.

References

[1] E. Aarts and P. van Laarhoven. A pedestrian review on the theory and applications of the simulated annealing algorithm. In J. van Hemmen and I. Morgenstern, editors, *Heidelberg Colloquium on Glassy Dynamics*, pages 288–307, Berlin, 1987. Springer-Verlag.

[2] F. Aluffi-Pentini, V. Parisi, and F. Zirilli. Global optimization and stochastic differential equations. *Journal of optimization theory and applications*, 47:1–16, 1985.

[3] S. Amari. A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, 16:299–307, 1967.

[4] S. Amato, B. Apolloni, G. Caporali, U. Madesani, and A. Zanaboni. Simulated annealing approach in backpropagation. *Neurocomputing*, 3:207–220, 1991.

[5] R. Battiti. First- and second-order methods for learning: between steepest descent and Newton's method. *Neural Computation*, 4:141–166, 1992.

[6] S. Becker. Unsupervised learning procedures for neural networks. *International Journal of Neural Systems*, 2:17–33, 1991.

[7] D. Bedeaux, K. Lakatos-Lindenberg, and K. Shuler. On the relation between master equations and random walks and their solutions. *Journal of Mathematical Physics*, 12:2116–2123, 1971.

[8] O. Catoni. Rough large deviation estimates for simulated annealing: application to exponential schedules. *The Annals of Probability*, 20:1109–1146, 1992.

[9] D. Clark and K. Ravishankar. A convergence theorem for Grossberg learning. *Neural Networks*, 3:87–92, 1990.

[10] M. Cottrell and J. Fort. A stochastic model of retinotopy: a self-organizing process. *Biological Cybernetics*, 53:405–411, 1986.

[11] C. Darken, J. Chang, and J. Moody. Learning rate schedules for faster stochastic gradient search. In *Neural Networks for Signal Processing 2*, New York, 1992. IEEE.

[12] J. Doob. *Stochastic Processes.* Wiley, New York, 1953.

[13] W. Finnoff. Diffusion approximations for the constant learning rate backpropagation algorithm and resistance to local minima. In S. Hanson, J. Cowan, and L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 459–466, San Mateo, 1993. Morgan Kaufmann.

[14] C. Gardiner. *Handbook of Stochastic Methods.* Springer, Berlin, second edition, 1985.

[15] S. Geman and C. Hwang. Diffusions for global optimization. *Siam Journal on Control and Optimization*, 24:1031–1043, 1986.

[16] T. Geszti. *Physical Models of Neural Networks.* World Scientific, Singapore, 1990.

[17] B. Gidas. The Langevin equation as a global minimization algorithm. In E. Bienenstock, F. Fogelman Soulié, and G. Weisbuch, editors, *Disordered Systems and Biological Organization*, pages 321–326, Berlin, 1986. Springer-Verlag.

[18] M. Gori and A. Tesi. On the problem of local minima in backpropagation. *IEEE Transactions on PAMI*, 14:76–86, 1992.

[19] S. Grossberg. On learning and energy-entropy dependence in recurrent and nonrecurrent signed networks. *Journal of Statistical Physics*, 48:105–132, 1969.

[20] S. Grossberg. *The Adaptive Brain I.* North-Holland, Amsterdam, 1986.

[21] T. Guillerm and N. Cotter. A diffusion process for global optimization in neural networks. In *IJCNN*, volume 1, pages 335–340, New York, 1991. IEEE.

[22] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13:311–329, 1988.

[23] L. Hansen, R. Pathria, and P. Salamon. Stochastic dynamics of supervised learning. *Journal of Physics A*, 26:63–71, 1993.

[24] L. Hansen and P. Salomon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:993–1001, 1990.

[25] S. Hanson and L. Pratt. A comparison of different biases for minimal network construction with back-propagation. In D. Touretzky, editor, *Advances in Neural Information Processing Systems 1*, pages 177–185. Morgan Kaufmann, 1989.

[26] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, 1991.

[27] T. Heskes. Transition times in self-organizing maps. *Submitted to Biological Cybernetics*, 1992.

[28] T. Heskes and B. Kappen. Learning processes in neural networks. *Physical Review A*, 44:2718–2726, 1991.

[29] T. Heskes and B. Kappen. Learning-parameter adjustment in neural networks. *Physical Review A*, 45:8885–8893, 1992.

[30] T. Heskes and B. Kappen. Error potentials for self-organization. In *International Conference on Neural Networks, San Francisco*, volume 3, pages 1219–1223, New York, 1993. IEEE.

[31] T. Heskes, E. Slijpen, and B. Kappen. Learning in neural networks with local minima. *Physical Review A*, 46:5221–5231, 1992.

[32] T. Heskes, E. Slijpen, and B. Kappen. Cooling schedules for learning in neural networks. *Physical Review E*, 47:4457–4464, 1993.

[33] G. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40:185–234, 1989.

[34] K. Hornik and C. Kuan. Convergence analysis of local feature extraction algorithms. *Neural Networks*, 5:229–240, 1992.

[35] R. Jacobs. Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1:295–307, 1988.

[36] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[37] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[38] T. Kohonen. *Self-organization and Associative Memory.* Springer, New York, 1988.

[39] A. Kramer and A. Sangiovanni-Vincentelli. Efficient parallel learning algorithms for neural networks. In D. Touretzky, editor, *Advances in Neural Information Processing Systems 1*, pages 40–48. Morgan Kaufmann, 1989.

[40] C. Kuan and K. Hornik. Convergence of learning algorithms with constant learning rates. *IEEE Transactions on Neural Networks*, 2:484–89, 1991.

[41] H. Kushner. Robustness and approximation of escape times and large deviations estimates for systems with small noise effects. *SIAM Journal of Applied Mathematics*, 44:160–182, 1984.

[42] H. Kushner. Asymptotic global behavior for stochastic approximation and diffusions with slowly decreasing noise effects: global minimization via Monte Carlo. *SIAM Journal of Applied Mathematics*, 47:169–185, 1987.

[43] H. Kushner and D. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Springer, New York, 1978.

[44] T. Leen and G. Orr. Weight-space probability densities and convergence times for stochastic learning. In *International Joint Conference on Neural Networks.* IEEE, 1992.

[45] E. Levin, N. Tishby, and S. Solla. A statistical approach to learning and generalization in layered neural networks. *Proceedings IEEE*, 78:1568–1574, 1990.

[46] P. Lisboa and S. Perantonis. Complete solution of the local minima in the xor problem. *Network*, 2:119–124, 1991.

[47] L. Ljung. Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, AC-22:551–575, 1977.

[48] Z. Luo. On the convergence of LMS algorithm with adaptive learning rate for linear feedforward netyworks. *Neural Computation*, 3:226–245, 1991.

[49] E. Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273, 1982.

[50] G. Radons, H. Schuster, and D. Werner. Fokker-Planck description of learning in backpropagation networks. In *International Neural Network Conference 90 Paris*, pages 993–996, Dordrecht, 1990. Kluwer Academic.

[51] H. Ritter, K. Obermayer, K. Schulten, and J. Rubner. Self-organizing maps and adaptive filters. In E. Domany, J. van Hemmen, and K. Schulten, editors, *Models of Neural Networks*, pages 281–306, Berlin, 1991. Springer.

[52] H. Ritter and K. Schulten. On the stationary state of Kohonen's self-organizing sensory mapping. *Biological Cybernetics*, 54:99–106, 1986.

[53] H. Ritter and K. Schulten. Convergence properties of Kohonen's topology conserving maps: fluctuations, stability, and dimension selection. *Biological Cybernetics*, 60:59–71, 1988.

[54] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

[55] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[56] T. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2:459–473, 1989.

[57] H. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical Review A*, 45:6056–6091, 1992.

[58] S. Shinomoto and Y. Kabashima. Finite time scaling of energy in simulated annealing. *Journal of Physics A*, 24:L141–L144, 1991.

[59] S. Singhal and L. Wu. Training multilayered perceptrons with the extended Kalman algorithm. In D. Touretzky, editor, *Advances in Neural Information Processing Systems 1*, pages 133–140, San Mateo, 1989. Morgan Kaufmann.

[60] M. Styblinski and T. Tang. Experiments in nonconvex optimization: stochastic approximation with function smoothing and simulated annealing. *Neural Networks*, 3:467–483, 1990.

[61] H. Szu. Fast simulated annealing. In J. Denker, editor, *Neural Networks for Computing*, pages 420–425, New York, 1986. American Institute of Physics.

[62] J. Tsitsiklis. Markov chains with rare transitions and simulated annealing. *Mathematics of Operations Research*, 14:70–71, 1989.

[63] N. van Kampen. *Stochastic Processes in Physics and Chemistry.* North-Holland, Amsterdam, 1981.

[64] T. Watkin, A. Rau, and M. Biehl. The statistical mechanics of learning a rule. *Reviews of Modern Physics*, 65:499–556, 1993.

[65] H. White. Learning in artificial neural networks: a statistical perspective. *Neural Computation*, 1:425–464, 1989.

[66] B. Widrow and M. Hoff. Adaptive switching circuits. *1960 IRE WESCON Convention Record, Part 4*, pages 96–104, 1960.

[67] N. Wiener. *I am a Mathematician.* Doubleday, New York, 1956.