Learning processes in neural networks[1]

Tom M. Heskes Bert Kappen

Department of Medical Physics and Biophysics,
University of Nijmegen, Geert Grooteplein Noord 21,
6525 EZ Nijmegen, The Netherlands,
e-mail: tom@mbfys.kun.nl

Abstract

We study the learning dynamics of neural networks from a general point of view. The environment from which the network learns is defined as a set of input stimuli. At discrete points in time, one of these stimuli is presented and an incremental learning step takes place. If the time between learning steps is drawn from a Poisson distribution, the dynamics of an ensemble of learning processes is described by a continuous-time master equation. A learning algorithm that enables a neural network to adapt to a changing environment, must have a nonzero learning parameter. This constant adaptability, however, goes at the cost of fluctuations in the plasticities, such as synapses and thresholds. The ensemble description allows us to study the asymptotic behavior of the plasticities for a large class of neural networks. For small learning parameters we derive an expression for the size of the fluctuations in an unchanging environment. In a changing environment, there is a trade-off between adaptability and accuracy (i.e., size of the fluctuations). We use the networks of Grossberg [J. Stat. Phys. **48**, 105 (1969)] and Oja [J. Math. Biol. **15**, 267 (1982)] as simple examples to analyze and simulate the performance of neural networks in a changing environment. In some cases an optimal learning parameter can be calculated.

[1] Physical Review A, Volume 44, Number 4, Pages 2718-2726

I Introduction

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 this environment. The representation is encoded in the plasticities of the network, 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 determined by the architecture of the network and by the learning rule. Examples of such functions may be classification (as in multilayered perceptrons), feature extraction (as in networks that perform a principle component analysis), recognition, transformation for motor tasks, or memory. The representation of the environment that the network has learned enables the network to perform its function in a way that is "optimally" suited for the environment on which it is taught.

The environment can be defined as a set of examples or stimuli, and learning is usually modeled as the process of randomly drawing examples from the environment and presenting them to the neural network. Thus learning becomes a stochastic process. So far the learning process in artificial neural networks has been considered almost exclusively for the case when the network is given examples from a *fixed unchanging* environment. The aim of these learning algorithms has been to find the *one static* representation of the environment, in terms of synapses and thresholds, that optimizes the function of the network for that specific environment. This requires that for large times the learning parameter, which controls the amount of learning, should go to zero, since otherwise fluctuations in the representation will persist and thus optimality in the above sense is never achieved. Conditions for convergence to an asymptotic solution are derived by Ljung [1] and Kushner and Clark [2] for general stochastic processes. More specifically, Ritter and Schulten [3] discuss the convergence properties of Kohonen's topology conserving maps and Clark and Ravishankar [4] give a convergence theorem for Grossberg learning.

Such algorithms, for which the learning parameter vanishes asymptotically, are clearly not the ones that are used in natural neural networks. Natural adaptive systems always learn. Examples of such learning exist on very large time scales (people learn with age) as well as on short time scales (attention for details, discovery of regularities). This constant tendency to learn accounts for the adaptability of biological neural systems to a *changing* environment.

In order to implement such behavior in artificial neural networks, the learning parameter should not go to zero asymptotically, but should take a constant nonzero value. The *adaptability* of the neural network is best served with a large learning parameter: The larger the learning parameter, the faster the response of the neural network to the changing environment. On the other hand, a large learning parameter gives rise to large fluctuations around the desired optimal representation. This has a negative effect on the *accuracy* of the network's representation of the environment at a given time. Given some criterion for the network's adaptability and accuracy, there is an optimal learning parameter that is certainly nonzero for a neural network operating in a time-dependent environment. It is interesting to note that similar ideas have been proposed by Wiener [5] in connection with his work on linear prediction theory.

We propose to study the learning dynamics of a large class of neural networks for constant learning parameter η. In Sec. II, we define the class of learning algorithms that we will consider. If the time between learning steps is drawn from a Poisson distribution, the dynamics of an ensemble of learning processes is described by a continuous-time master equation [6].

From this we can calculate in Sec. III the dynamics of macroscopic quantities such as the expected representation and its fluctuations. We illustrate our formalism with Grossberg learning [7], for which the evolution of the macroscopic quantities is exactly solvable.

For general learning rules, the asymptotic solutions cannot be calculated. In Sec. IV we therefore make an approximation valid for small fluctuations, as proposed by Van Kampen [8]. If it is assumed that the asymptotic solution is peaked around the "noise free" limit, the expected representation and its fluctuations obey a coupled set of linear differential equations of which the asymptotic solution can be calculated. We compare our analytical results with simulations for the Oja learning rule [9], which calculates the principal component of the covariance matrix of the input distribution.

In Sec. V we discuss the performance of learning rules in a gradually changing environment. The formalism, as developed in Sec. II and III is applicable to this case as long as changes in the environment are slow in comparison with the time scale of the learning algorithm. For a simple changing environment and Grossberg learning, the asymptotic solution can be calculated exactly, and illustrates the conflicting goals of accuracy and adaptability.

In Sec. VI, the analysis of Sec. IV is repeated for a changing environment. Again a set of linear differential equations is obtained. The usefulness of the analytical results are illustrated with Oja's learning rule, which receives its input from a slowly rotating environment.

In Sec. VII, some conclusions are drawn.

II The learning process

In this section we will define the class of learning algorithms that we consider. Let the representation that the neural network builds of the environment be given by a N-dimensional vector $\mathbf{w} = (w_1, \ldots, w_N)^T$. This vector $\mathbf{w}$ contains all the synaptic strengths and thresholds of the neural network, and completely specifies the *state* of the neural network in the learning process. The environment of the network is assumed to be a set of stimuli $\vec{x}$ to be taken from a subset $\Omega \subset \Re^n$. Here n denotes the dimension of the stimulus space, which will often be equal to the number of input neurons. The environment is fixed. The probability that the network gets exposed to a stimulus $\vec{x}$ is given by a probability distribution $\rho(\vec{x})$, which for the moment is time independent.

We consider the following learning mechanism. At distinct points in time a stimulus $\vec{x}$ is presented to the network and a learning step takes place. The network changes its weight vector $\mathbf{w}$ to $\mathbf{w}' = \mathbf{w} + \Delta\mathbf{w}$, obeying

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

where $\mathbf{f}(\mathbf{w}, \vec{x})$, the so-called "stochastic force", is an arbitrary function $\mathbf{f}(\mathbf{w}, \vec{x}) : \Re^N \times \Re^n \to \Re^N$ ($\Re$ representing the set of all real numbers) and η is the learning parameter. Eq. (1) simply states that the new network state $\mathbf{w}'$ after the learning step is a function of the state $\mathbf{w}$ before this learning step and the randomly drawn input vector $\vec{x}$.

Eq. (1) applies to most of the learning rules in neural network theory. Depending on the particular choice of the stochastic force $\mathbf{f}(\mathbf{w}, \vec{x})$, learning processes of neural networks with quite different functionalities can be described. A few illustrative examples are the following.

(i) Kohonen's topological feature map [10] as used in Ritter and Schulten [3]:

$$\Delta\vec{w}_i \;=\; \eta\,(\vec{x} - \vec{w}_i) h_\sigma(i, i_{\max}(\vec{x}))\,,$$

with i labeling the neurons, $\vec{w}_i$ a set of feedforward connections, $i_{\max}(\vec{x})$ the neuron that fires maximally when stimulus $\vec{x}$ is presented, and h_σ a bell-shaped function of width σ.

(ii) Hopfield's associative memory [11]: $\Delta w_{ij} = \eta\, x_i x_j$, with i labeling the neurons x_i the stimulus value at neuron i, and w_{ij} the lateral connections.

(iii) An input-output relation such as the multilayered perceptron with backpropagation [12]: $\Delta\mathbf{w} = -\eta\, \partial_w E$, with $\mathbf{w}$ the weights and thresholds of the network, and E an error function which should be minimized.

Two other examples, Oja's principal component network and Grossberg's "center-of-mass" network will be used as specific examples to illustrate our theory in the subsequent sections.

The learning process as defined in Eq. (1) is a stochastic process since at each learning step the input vector $\vec{x}$ is drawn *at random.* In order to describe this learning process we must therefore talk in terms of probabilities, expectation values, and fluctuations. The most obvious probability to look at is $p_i(\mathbf{w})$: the probability that the network is in state $\mathbf{w}$ after i learning steps. Thus, the learning process becomes a Markov process (see, e.g., Ref. [13]):

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

where $T(\mathbf{w}'|\mathbf{w})$ is the transition probability to go in one learning step from state $\mathbf{w}$ to state $\mathbf{w}'$:

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

Eq. (2) describes a random walk with discrete iteration steps labeled by i.

It can be shown [1-4] that, under certain conditions including a slowly vanishing of the learning parameter,

$$\lim_{i\to\infty} \eta_i = 0 \, , \quad \sum_{i=1}^{\infty} \eta_i = \infty \, ,$$

the learning process converges to a stationary solution

$$p_S(\mathbf{w}) \;=\; \delta^N(\mathbf{w} - \mathbf{w}^*) \, , \tag{4}$$

where the points $\mathbf{w}^*$ are stable fixed points of the differential equation

$$\frac{d\mathbf{w}(t)}{dt} \;=\; \langle \mathbf{f}(\mathbf{w}(t), \vec{x}) \rangle_\Omega \;\equiv\; \mathbf{f}(\mathbf{w}(t)) \, . \tag{5}$$

These stable fixed points $\mathbf{w}^*$ are, by definition of the learning rule, locally optimal representations of the environment. If a global energy function $E(\mathbf{w})$ exists such that $f_i(\mathbf{w}) = -\partial_{w_i} E(\mathbf{w})$ for all $\mathbf{w}$, then the stable fixed points $\mathbf{w}^*$ are local minima of this energy function $E(\mathbf{w})$.

Instead of the above approach, we will discuss learning processes with small but non-vanishing learning parameters. Therefore we need a continuous-time description that is valid for all values of the learning parameter η. Bedeaux, Lakatos-Lindenberg and Shuler [6] showed, that such a continuous-time description can be obtained through the assignment of *random* values Δt to the time interval between two succeeding iteration steps labeled by i. 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. Now the probability $P(\mathbf{w}, t)$, that a network is in state $\mathbf{w}$ at *time* t, is defined

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

This probability function $P(\mathbf{w}, t)$ can be differentiated with respect to time, yielding the master equation

$$\frac{\partial P(\mathbf{w}', t)}{\partial t} \;=\; \int d^N w \; \left[W(\mathbf{w}'|\mathbf{w}) P(\mathbf{w}, t) \; - \; W(\mathbf{w}|\mathbf{w}') P(\mathbf{w}', t) \right] \, , \tag{6}$$

with the transition probability per unit time

$$W(\mathbf{w}'|\mathbf{w}) \;=\; \frac{1}{\tau} T(\mathbf{w}'|\mathbf{w}) \, . \tag{7}$$

This result is valid independently of τ, the average time between two successive learning steps, and the learning parameter η. Through τ we have introduced a physical time scale, which is also reflected in the transition probability *rate* $W(\mathbf{w}'|\mathbf{w})$ in Eq. (7).

Through the assignment of random time values to the learning steps, we have obtained a continuous-time master equation (6) describing the evolution of an ensemble of learning neural networks. We will denote the distribution of states $\mathbf{w}$ at time t by $\Xi(t)$. The expectation value for an arbitrary function $\psi(\mathbf{w})$ at time t is written

$$\langle \psi(\mathbf{w}) \rangle_{\Xi(t)} \;\equiv\; \int d^N w \, P(\mathbf{w},t) \psi(\mathbf{w}) \, . \tag{8}$$

III Learning in a fixed environment

A consequence of an asymptotically constant nonzero learning parameter is that fluctuations will persist and the learning process, in general, will not converge to a deterministic solution like the one in Eq. (4). So local optimality is not likely to be achieved. As an indication of the deviation from local optimality, we define the error

$$\mathcal{E} \;\equiv\; \left\langle \|\mathbf{w}-\mathbf{w}^*\|^2 \right\rangle_{\Xi(\infty)} \;=\; \|\mathbf{m}(\infty)\|^2 + \mathrm{Tr}\left[\Sigma^2(\infty)\right] \, , \tag{9}$$

with definitions of the bias and the covariance matrix, respectively,

$$\begin{aligned} m_i(t) &\equiv \langle w_i \rangle_{\Xi(t)} - w_i^* \, , \\ \Sigma^2_{ij}(t) &\equiv \left\langle \left(w_i - \langle w_i \rangle_{\Xi(t)} \right) \left(w_j - \langle w_j \rangle_{\Xi(t)} \right) \right\rangle_{\Xi(t)} \, . \end{aligned}$$

Note that the error $\mathcal{E}$ as defined in Eq. (9) gives a measure of the performance of the network in the neighborhood of $\mathbf{w}^*$: it does not give any information about the global performance of the network. In order to compute this error, we will focus on the evolution equations of the macroscopic quantities $\langle w \rangle_{\Xi(t)}$ and $\Sigma^2(t)$.

Using the master equation (6) and the definition (8), we obtain

$$\begin{aligned} \frac{\tau}{\eta} \frac{d \langle w_i \rangle_{\Xi(t)}}{dt} &= \langle f_i(\mathbf{w}) \rangle_{\Xi(t)} \, , \\ \frac{\tau}{\eta} \frac{d \Sigma^2_{ij}(t)}{dt} &= \left\langle f_i(\mathbf{w}) \left(w_j - \langle w_j \rangle_{\Xi(t)} \right) \right\rangle_{\Xi(t)} + \left\langle \left(w_i - \langle w_i \rangle_{\Xi(t)} \right) f_j(\mathbf{w}) \right\rangle_{\Xi(t)} + \eta \, \langle D_{ij}(\mathbf{w}) \rangle_{\Xi(t)} \, , \end{aligned} \tag{10}$$

with the drift vector $\mathbf{f}(\mathbf{w})$ already defined in Eq. (5) and the diffusion tensor $D(\mathbf{w})$,

$$D_{ij}(\mathbf{w}) \;\equiv\; \langle f_i(\mathbf{w},\vec{x}) f_j(\mathbf{w},\vec{x}) \rangle_\Omega \, .$$

In Eq. (10), $\langle w \rangle_{\Xi(t)}$ describes the "mean tendency" of the learning system and $\Sigma^2(t)$ the superimposed fluctuations. The diffusion tensor $D(\mathbf{w})$ is a positive definite matrix. It contains the fluctuations in the learning rule.

The evolution equations in Eq. (10) are exact, i.e., they are valid for all values of η. The exact evolution equations for higher-order cumulants can be derived in the same way. For our purposes, the expectation value of the state and the covariance matrix provide adequate information about the learning process. Note that, since terms of order η^3 and higher do not contribute to the evolution of $\langle \mathbf{w} \rangle_{\Xi(t)}$ and $\Sigma^2(t)$, Eq. (10) can also be derived from a Fokker-Planck approach which results from a Taylor expansion including terms up to order η^2.

To illustrate the dynamics of a learning process, we consider Grossberg learning [7]. The network consists of one neuron with n inputs. Its weight vector $\mathbf{w}$ follows the learning rule,

$$\Delta \mathbf{w} \;=\; \eta \, (\mathbf{x} - \mathbf{w}) \, . \tag{11}$$

Obviously the dimension of the weight vector $\mathbf{w}$ is equal to the dimension of the input space, the set of stimuli $\mathbf{x} \in \Omega \subset \Re^n$. Since the different dimensions in Eq. (11) are uncoupled,

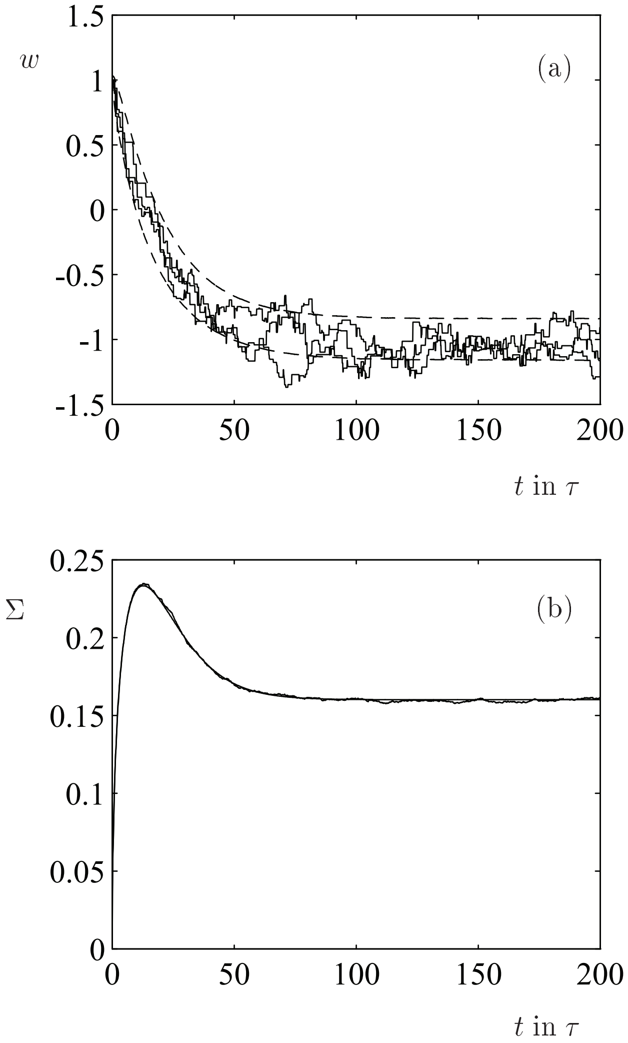

Figure 1: Mean and standard deviation for time-independent Grossberg learning as a function of time in units τ. Learning parameter $\eta = 0.05$. Standard deviation input $\chi = 1.0$. Probabilistic mean $\langle x \rangle_\Omega = -1.0$. All 10 000 neural networks started with $w = 1.0$, so $\langle w \rangle_{\Xi(0)} = 1.0$ and $\Sigma^2(0) = 0.0$. (a) Three examples of individually learning networks (solid lines) and simulated mean ± standard deviation (dashed lines). (b) Variance $\Sigma^2(t)$, simulated (solid line) and calculated (dashed line).

we can restrict ourselves to one dimension. The convergence of the Grossberg learning rule in case of equally spaced time intervals between the learning steps has been studied by Clark and Ravishankar [4]. The stable fixed point of Eq. (5) is the probabilistic mean of the input distribution: $w^* = \langle x \rangle_\Omega$.

The evolution equations for the mean $\langle w \rangle_{\Xi(t)}$ and the standard deviation $\Sigma(t)$ can be calculated exactly; Eq. (10) yields

$$\begin{aligned} \tau \frac{d \langle w \rangle_{\Xi(t)}}{dt} &= -\eta\, m(t)\,, \\ \tau \frac{d\Sigma^2(t)}{dt} &= -(2-\eta)\eta\, \Sigma^2(t) \;+\; \eta^2 m^2(t) \;+\; \eta^2 \chi^2\,, \end{aligned} \tag{12}$$

with definitions $m(t) \equiv \langle w \rangle_{\Xi(t)} - \langle x \rangle_\Omega$ and $\chi^2 = \left\langle (x - \langle x \rangle_\Omega)^2 \right\rangle_\Omega$, the variance of the input distribution. The solution of Eq. (12) is

$$\begin{aligned} m(t) &= m(0)\, \mathrm{e}^{-\eta t/\tau}\,, \\ \Sigma^2(t) &= \frac{\eta}{2-\eta}\chi^2 \;+\; \left[\Sigma^2(0) - \frac{\eta}{2-\eta}\chi^2 + m^2(0)\left(1 - \mathrm{e}^{-\eta^2 t/\tau}\right)\right] \mathrm{e}^{-(2-\eta)\eta t/\tau}\,. \end{aligned} \tag{13}$$

So the ensemble of learning neural networks converges for $\eta < 2$ to the asymptotically stable solution

$$\begin{aligned} \langle w \rangle_{\Xi(\infty)} &= \langle x \rangle_\Omega\,, \\ \Sigma^2(\infty) &= \frac{\eta}{2-\eta}\chi^2\,. \end{aligned}$$

The expectation value converges to $\langle x \rangle_\Omega = w^*$. The error $\mathcal{E}$ is therefore equal to the final variance in the weights w that is proportional to the variance of the input distribution and for small learning parameters also to the learning parameter. The standard deviation diverges at $\eta = 2$.

We have simulated this Grossberg learning for an ensemble of 10 000 independently operating neural networks looking at an environment with $\rho(x) = 1/2l$ for $|x| < l$ and $\rho(x) = 0$ elsewhere. Three examples of individual networks and $\langle w \rangle_{\Xi(t)} \pm \Sigma(t)$ are plotted as a function of time in Fig. 1(a). Fig. 1(b) shows the variance $\Sigma^2(t)$, both calculated and simulated. The results are in excellent agreement with Eq. (13).

IV A Gaussian approximation

Eq. (10), which describes the evolution of the mean and the covariance matrix, is elegant but in general unsolvable. Therefore we make a Gaussian approximation valid for small fluctuations as proposed by Van Kampen [8]. For this approximation to be valid, one must assume that the learning process converges to a stationary solution of the master equation (6). Convergence can be proved in case of a finite number of possible states $\mathbf{w}$ (see, e.g., Ref. [13]). The convergence proof for a continuous state space requires the *a priori* existence of a stationary solution. We will show the existence of stationary solutions within our approximation scheme. This justifies, *a posteriori*, the Van Kampen approximation.

Application of the approximation method introduced by Van Kampen to the evolution equations (10) yields

$$\begin{aligned} \frac{\tau}{\eta} \frac{d \langle w_i \rangle_{\Xi(t)}}{dt} &= f_i(\langle \mathbf{w} \rangle_{\Xi(t)}) \;+\; \frac{1}{2} \sum_{jk} Q_{ijk}(\langle \mathbf{w} \rangle_{\Xi(t)}) \Sigma^2_{jk}(t)\,, \\ \frac{\tau}{\eta} \frac{d\Sigma^2_{ij}(t)}{dt} &= -\sum_k G_{ik}(\langle \mathbf{w} \rangle_{\Xi(t)}) \Sigma^2_{kj}(t) \;-\; \sum_k \Sigma^2_{ik}(t) G_{jk}(\langle \mathbf{w} \rangle_{\Xi(t)}) \;+\; \eta D_{ij}(\langle \mathbf{w} \rangle_{\Xi(t)})\,, \end{aligned} \tag{14}$$

with definitions

$$G_{ij}(\mathbf{w}) \equiv -\frac{\partial f_i(\mathbf{w})}{\partial w_j}, \quad Q_{ijk}(\mathbf{w}) = \frac{\partial f_i(\mathbf{w})}{\partial w_j \partial w_k}.$$

In Eq. (14) higher order terms are omitted. According to this approximation $|\Sigma^2(t)|$ tends to become $\eta|D|/|G|$ for $t \to \infty$ (by $|\ldots|$ we mean the order of magnitude of the tensor). This value can be used to check the self-consistency of this approximation. The equations are approximately valid if the largest neglected terms are much smaller than the terms we take into account, i.e.,

$$\begin{aligned} \eta|\partial_w^3 \mathbf{f}|^2|D| &\ll |\partial_w \mathbf{f}||\partial_w^2 \mathbf{f}|^2 , \\ \eta|\partial_w^2 D| &\ll |\partial_w \mathbf{f}| , \\ \eta|\partial_w^2 \mathbf{f}|^2|D| &\ll |\partial_w \mathbf{f}|^3 . \end{aligned} \tag{15}$$

If the drift term $\mathbf{f}(\mathbf{w})$ and the diffusion tensor $D(\mathbf{w})$ are sufficiently smooth, we can always find a learning parameter η such that the requirements (15) are fulfilled. Eq. (14) is a set of two coupled nonlinear differential equations which describe the evolution of the expected state and the superimposed fluctuations for small fluctuations. Note that Eq. (14) is, strictly speaking, only valid for $t < \infty$ as long as $\Sigma^2(t)$ is of the same order of magnitude as $\Sigma^2(\infty)$. In many cases this is true for the entire learning process [see, e.g., Fig. 1(b)].

We will show that for small learning parameters there exist stationary solutions of the master equation (6) that are peaked in the neighborhood of the stable fixed points $\mathbf{w}^*$ of Eq. (5). We expand Eq. (14) with respect to the bias $\mathbf{m}(t) \equiv \langle \mathbf{w} \rangle_{\Xi(t)} - \mathbf{w}^*$:

$$\begin{aligned} \frac{\tau}{\eta}\frac{dm_i(t)}{dt} &= -\sum_j G_{ij} m_j(t) + \frac{1}{2}\sum_{jk} Q_{ijk}\Sigma^2_{jk}(t) , \\ \frac{\tau}{\eta}\frac{d\Sigma^2_{ij}(t)}{dt} &= -\sum_k G_{ik}\Sigma^2_{kj}(t) - \sum_k \Sigma^2_{ik}(t) G_{jk} + \eta D_{ij} , \end{aligned} \tag{16}$$

with all tensors evaluated at $\mathbf{w}^*$. Note that the stability of $\mathbf{w}^*$ implies that the symmetric part of the matrix $G(\mathbf{w}^*)$ must be positive semidefinite. For convenience, we will exclude matrices $G(\mathbf{w}^*)$ with zero eigenvalues. The analysis including “flat directions” should be restricted to the eigenspace spanned by the eigenvectors with nonzero eigenvalues.

In Eq. (16), higher orders are omitted and self-consistency can be checked using $|\Sigma^2(\infty)| = \eta|D|/|G|$ and $|\mathbf{m}(\infty)| = \eta|\mathbf{Q}||D|/|G|^2$. The approximate validity of Eq. (16) requires (15) and

$$\begin{aligned} \eta|D||\partial_w^3 \mathbf{f}| &\ll |\partial_w \mathbf{f}|^2 , \\ \eta|\partial_w^2 \mathbf{f}||\partial_w D| &\ll |\partial_w \mathbf{f}|^2 . \end{aligned} \tag{17}$$

Under these conditions, the set of linear differential equations (16) gives an approximate description of the convergence of the learning process [Eqs. (6) and (10)] to a stationary state.

The stationary solution of Eq. (16) obeys

$$\begin{aligned} \sum_j G_{ij} m_j(\infty) &= \frac{1}{2}\sum_{jk} Q_{ijk}\Sigma^2_{jk}(\infty) , \\ \sum_k G_{ik}\Sigma^2_{kj}(\infty) + \sum_k \Sigma^2_{ik}(\infty) G_{jk} &= \eta D_{ij} . \end{aligned}$$

In closed form the asymptotic solution is

$$\begin{aligned} m_i(\infty) &= \eta \sum_{jklmn} \left[G^{-1}\right]_{ij} Q_{jkl} \int_0^\infty dy \, \left[\mathrm{e}^{-Gy}\right]_{km} D_{mn} \left[\mathrm{e}^{-G^T y}\right]_{nl} , \\ \Sigma^2_{ij}(\infty) &= \eta \sum_{kl} \int_0^\infty dy \, \left[\mathrm{e}^{-Gy}\right]_{ik} D_{kl} \left[\mathrm{e}^{-G^T y}\right]_{lj} . \end{aligned} \tag{18}$$

The existence of this stationary solution of the master equation (6) *a posteriori* justifies our approximation scheme as outlined in this section. Note that in this approximation the asymptotic mean representation $\langle \mathbf{w} \rangle_{\Xi(\infty)}$ deviates from the locally optimal representation $\mathbf{w}^*$ proportional to η. The asymptotic standard deviation is proportional to $\sqrt{\eta}$, which is significantly larger than η for small learning parameters.

If the matrix $G(\mathbf{w}^*)$ is symmetric, the error $\mathcal{E}$ defined in Eq. (9) can be calculated yielding

$$\mathcal{E} \;=\; \frac{\eta}{2} \mathrm{Tr} \left[G^{-1} D \right] \;, \tag{19}$$

where terms of order η^2 are neglected. Eq. (19) summarizes a few characteristics of the asymptotic solution of the learning process. First of all, the error is proportional to the diffusion matrix $D(\mathbf{w}^*)$ which contains the "noisiness" of the environment. Secondly, the error is inversely proportional to the curvature at the stable solution, i.e., the steeper the valley of the minimum, the smaller the error.

The "perfectly trainable" neural networks form a special class of learning neural networks. These networks have a stable fixed point $\mathbf{w}^*$ such that $\mathbf{f}(\mathbf{w}^*, \vec{x}) = \mathbf{0} \; \forall_{\vec{x}} \in \Omega$. In this point, the diffusion $D(\mathbf{w}^*) = 0$, so there are no fluctuations. Since there is no way to escape from this point, as can be seen from the transition probability $T(\mathbf{w}|\mathbf{w}^*) = \delta^N(\mathbf{w} - \mathbf{w}^*)$ in Eq. (3), $\mathbf{w}^*$ acts like a sink. In this particular case there is no harm in choosing a relatively large learning parameter. An example is a backpropagation network [14] that obtains a representation $\mathbf{w}^*$ of the environment such that all input vectors $\vec{x} \in \Omega$ are transformed exactly into the desired output vectors, i.e., for which the backpropagation error $E(\mathbf{w}^*) = 0$.

The set of equations (16) describes the exponential decay of the expected bias and fluctuations in the representation of the neural network. The *response time*, the typical time constant of these exponential decays, is different in the different eigenvector directions of the matrix G. Let us denote the response time in the eigenvector direction α with corresponding eigenvalue λ_α by θ_α; then,

$$\theta_\alpha \;=\; \frac{\tau}{\eta \; \mathrm{Re} \left(\lambda_\alpha \right)} \;. \tag{20}$$

The response time, which is an indication for the adaptability of a neural network to a changing environment, is inversely proportional to the learning parameter. Combining Eq. (19) and (20) we see that in order to reduce the response time by a factor 2, the learning parameter must be increased by a factor 2, yielding a twice as large error $\mathcal{E}$.

We conclude this section by calculating the asymptotic solutions $\mathbf{m}(\infty)$ and $\Sigma^2(\infty)$ for the nonlinear learning rule of Oja [9]. This algorithm computes the principal component of the covariance matrix of the stimulus set Ω. The network consists of one neuron with n inputs. Its weight vector $\mathbf{w}$ follows the learning rule

$$\Delta \mathbf{w} \;=\; \eta \, \mathbf{w}^T \mathbf{x} \left[\mathbf{x} - (\mathbf{w}^T \mathbf{x}) \mathbf{w} \right] . \tag{21}$$

With the definition of the covariance matrix of the input distribution $C \equiv \left\langle \mathbf{x}\mathbf{x}^T \right\rangle_\Omega$, it is easy to show that the normalized eigenvector of C with the largest eigenvalue is the only stable fixed point $\mathbf{w}^*$ of Eq. (5).

We take $n = 2$ and draw our stimuli at random from a rectangle:

$$\rho(x_1, x_2) = \rho_1(x_1) \rho_2(x_2) \text{ with } \rho_\alpha(x) = \left\{ \begin{array}{cl} 1/2l_\alpha & \text{for } |x| < l_\alpha \\ 0 & \text{otherwise} \;, \end{array} \right.$$

with $l_1 > l_2$. The covariance matrix of the input distribution has the form

$$C \;=\; \left(\begin{array}{cc} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{array} \right) ,$$

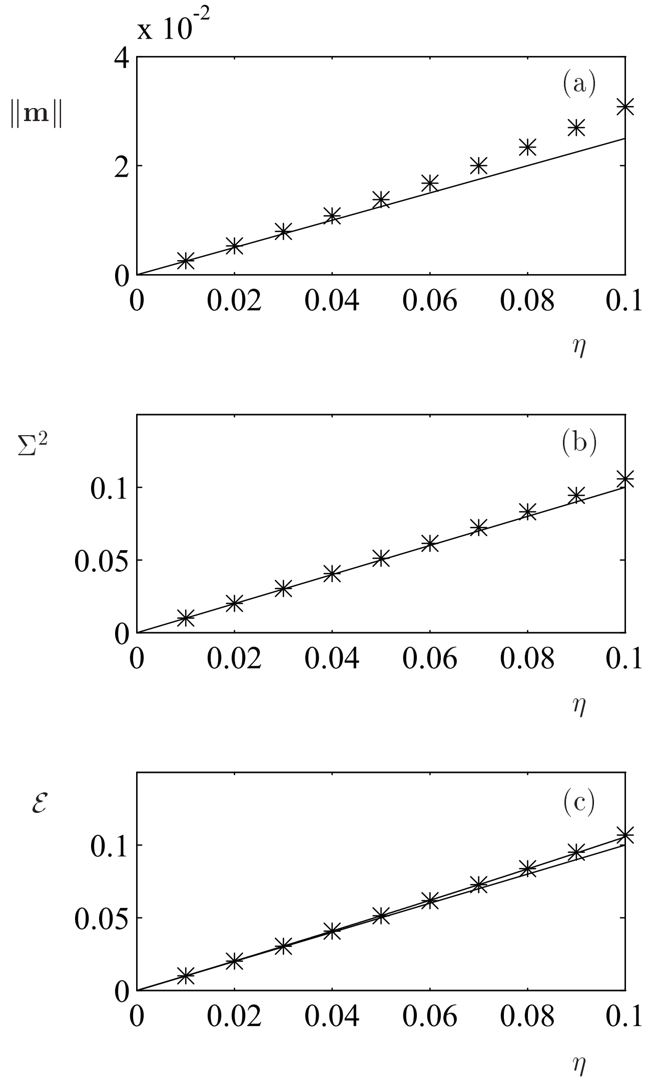

Figure 2: Asymptotic bias, variance, and error for Oja learning as a function of the learning parameter. Eigenvalues of the covariance matrix of the input distribution: $\Lambda_1 = 2.0$ and $\Lambda_2 = 1.0$. Simulations were done with 5 000 neural networks. (a) Bias $\|\mathbf{m}(\infty)\|$ computed from Eq. (22) (solid line) and simulated ($*$). (b) Variance Tr $\left[\Sigma^2(\infty)\right]$ computed from Eq. (22) (solid line) and simulated ($*$). (c) Error $\mathcal{E}$ computed up to order η (solid line), including all η^2 contributions (dashed line), and simulated ($*$).

where $\Lambda_\alpha = l_\alpha^2/3$. The eigenvectors of the covariance matrix of the input distribution with the largest eigenvalue Λ_1 are: $\mathbf{w}^* = (1\ ,\ 0)^T$ and $\mathbf{w}^* = (-1\ ,\ 0)^T$. Calculation of the stationary solution (18) is straightforward and leads to

$$\begin{aligned}\mathbf{m}(\infty) &= -\frac{\eta}{4}\frac{\Lambda_2^2}{\Lambda_1 - \Lambda_2}\mathbf{w}^* \ , \\ \Sigma^2(\infty) &= \frac{\eta}{2}\begin{pmatrix} 0 & 0 \\ 0 & \frac{\Lambda_1\Lambda_2}{\Lambda_1-\Lambda_2}\end{pmatrix} \ .\end{aligned} \tag{22}$$

In Fig. 2 $\|\mathbf{m}(\infty)\|$ and Tr $[\Sigma^2(\infty)]$ are plotted as a function of the learning parameter η, both calculated [Eq. (22), solid line] and simulated (with 5 000 neural networks, asterisk). The deviation between simulation and computation is less than 10% up to about $\eta = 0.05$.

The approximation scheme outlined in this section can be extended including higher-order terms of η. Since the term $\|\mathbf{m}(\infty)\|^2$ is already of order η^2, one only has to compute the second-order terms of Tr $[\Sigma^2(\infty)]$ to obtain a second-order estimate of the error. Straightforward calculation, using the first-order terms computed in Eq. (22), yields

$$\mathcal{E} = \frac{\eta}{2}\frac{\Lambda_1\Lambda_2}{\Lambda_1-\Lambda_2} + \frac{\eta^2}{4}\frac{\Lambda_1\Lambda_2^2}{\Lambda_1-\Lambda_2} + \frac{\eta^2}{16}\left[\frac{\Lambda_2^2}{\Lambda_1-\Lambda_2}\right]^2 .$$

This expression yields the dashed line in Fig. 2(c), which, of course, gives a better prediction of the simulations than the solid line which shows the error calculated up to order η [according to Eq. (19)].

The stationary probability distribution for $\eta = 0.05$ and 5 000 neural networks is plotted in Fig. 3(a). In Fig. 3(b) contour lines are drawn. The contour lines of a Gaussian with bias and covariance matrix (calculated up to order η^2) are drawn in Fig. 3(c). It is clear that the real probability distribution is not a simple Gaussian, but nevertheless the deviation of the simulated bias and variance from the values predicted by theory is small (Fig. 2).

V A gradually changing environment

We will now discuss the master equation describing the learning process in a gradually changing environment. By this we mean that we will assume that the set of stimuli $\Omega(t)$ and the probability density $\rho(\vec{x},t)$ change as a function of time t. Therefore, the transition probability for the network to go from a state $\mathbf{w}$ to a state $\mathbf{w}'$ at time t becomes

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

For a gradually changing environment (i.e., such that changes on a time scale τ are insignificant: $|\tau\partial_t\rho| \ll |\rho|$), we can write

$$\tau\frac{dP(\mathbf{w}',t)}{dt} = \int d^N w\, T_t(\mathbf{w}'|\mathbf{w})P(\mathbf{w},t) \ - \ P(\mathbf{w}',t) \ .$$

With the obvious definitions

$$\begin{aligned}\mathbf{f}(\mathbf{w},t) &= \langle \mathbf{f}(\mathbf{w},\vec{x})\rangle_{\Omega(t)} \ , \\ D(\mathbf{w},t) &= \langle D(\mathbf{w},\vec{x})\rangle_{\Omega(t)} \ ,\end{aligned}$$

the evolution equations for the expectation value of $\mathbf{w}$ and the covariance matrix $\Sigma^2(t)$ are written

$$\frac{\tau}{\eta}\frac{d\langle w_i\rangle_{\Xi(t)}}{dt} = \langle f_i(\mathbf{w},t)\rangle_{\Xi(t)} \ ,$$

$$\frac{\tau}{\eta}\frac{d\Sigma_{ij}^2(t)}{dt} = \Big\langle f_i(\mathbf{w},t)\left(w_j - \langle w_j\rangle_{\Xi(t)}\right)\Big\rangle_{\Xi(t)} + \Big\langle \left(w_i - \langle w_i\rangle_{\Xi(t)}\right) f_j(\mathbf{w},t)\Big\rangle_{\Xi(t)} + \eta\,\langle D_{ij}(\mathbf{w},t)\rangle_{\Xi(t)} \ .$$

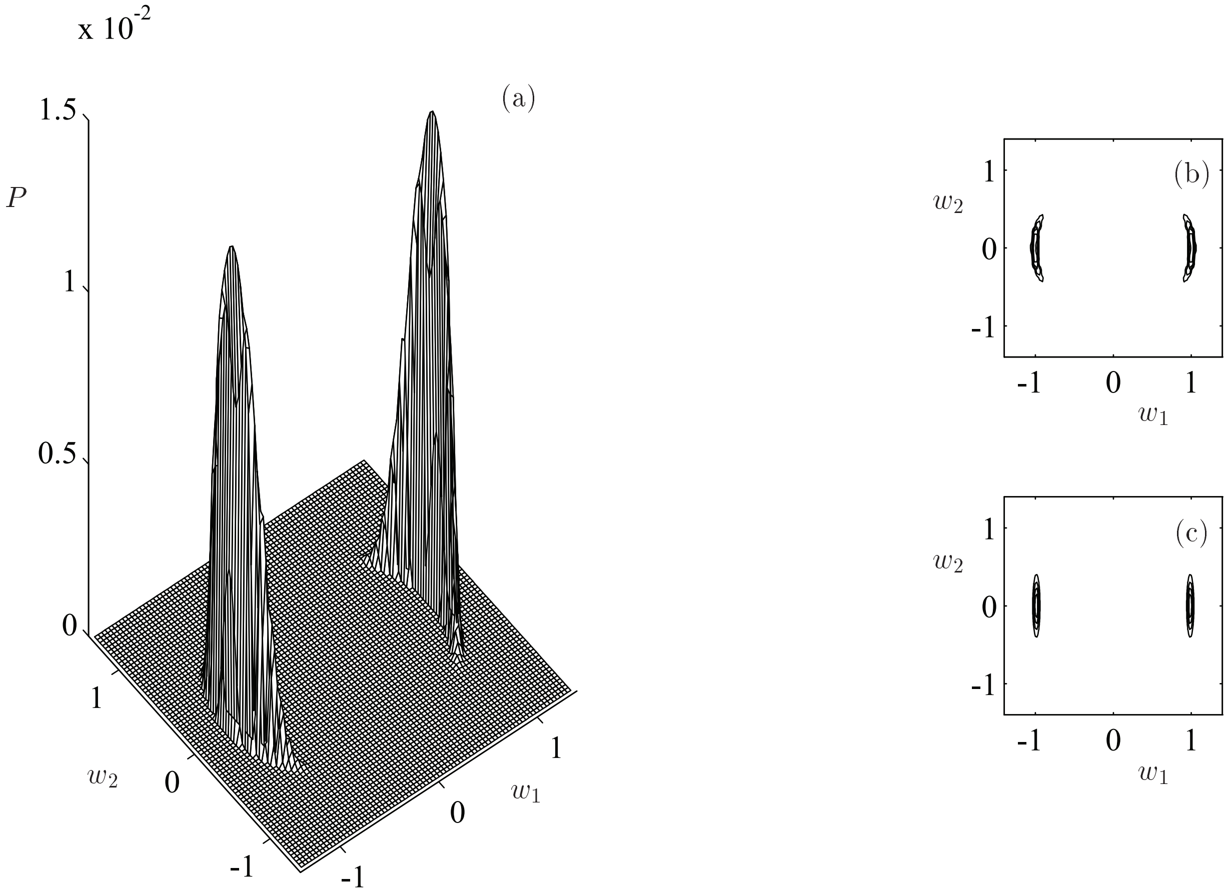

Figure 3: Asymptotic probability distribution for Oja learning. Learning parameter $\eta = 0.05$. Simulations were done with 5 000 neural networks. (a) Simulated probability distribution and (b) corresponding contour map. (c) Contour map of Gaussian probability distribution with calculated bias and covariance matrix including terms up to order η^2.

The time dependency of the environment leads to a time dependency of the stable fixed points $\mathbf{w}^*(t)$. Similar to the error defined in Eq. (9), we can define an error $\mathcal{E}$ indicating the performance of a neural network operating in a time-dependent environment:

$$\mathcal{E} = \lim_{T\to\infty} \frac{1}{T}\int_0^T dt \left\langle \|\mathbf{w} - \mathbf{w}^*(t)\|^2 \right\rangle_{\Xi(t)} = \lim_{T\to\infty} \frac{1}{T}\int_0^T dt \left\{ \|\mathbf{m}(t)\|^2 + \mathrm{Tr}\left[\Sigma^2(t)\right]\right\} , \tag{23}$$

with $\mathbf{m}(t) \equiv \langle \mathbf{w}\rangle_{\Xi(t)} - \mathbf{w}^*(t)$, as usual. The idea is that minimization of this error leads to an optimal learning parameter.

As an example we will discuss the performance of the Grossberg learning rule [Eq. (11)] in a time-dependent one-dimensional environment, where the input distribution is moving along the axis with constant velocity v and constant standard deviation χ: $\rho(x,t) = \tilde{\rho}(x - vt)$, with $\tilde{\rho}(x) = 1/2l$ for $|x| < l$ and $\tilde{\rho}(x) = 0$ elsewhere. The aim of this learning rule is to make w coincide with the mean value of the probability distribution $\rho(x,t)$, i.e., $w^*(t) = \langle x\rangle_{\Omega(t)}$. The evolution equations for $\langle w\rangle_{\Xi(t)}$ and $\Sigma^2(t)$ are given in Eq. (12), but now with definition

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

The asymptotic solution of these evolution equations is

$$\begin{aligned} \langle w\rangle_{\Xi(t)} &= \langle x\rangle_{\Omega(t)} - \frac{\tau v}{\eta} = \langle x\rangle_{\Omega(t-\tau/\eta)} \\ \Sigma^2(t) &= \Sigma^2 = \frac{\chi^2}{\eta(2-\eta)}\left[\eta^2 + \gamma^2\right] , \end{aligned} \tag{24}$$

with the typical constant

$$\gamma = \frac{v\tau}{\chi} ,$$

the ratio between the distance covered in the average time between two learning steps and the standard deviation. From Eq. (24) we see that, on the average, the representation which the network has of the environment, $\langle w\rangle_{\Xi(t)}$, lags a time τ/η behind the best possible representation, $\langle x\rangle_{\Omega(t)}$. Second, the standard deviation diverges at $\eta = 2$, as in the static case, but diverges also at $\eta = 0$ for nonzero velocities.

Eq. (24) is illustrated in Fig. 4, where the simulated probability distribution of the weight w is sketched for three different cases: zero velocity, small velocity ($v = 0.01/\tau$) and relatively large velocity ($v = 0.1/\tau$). Simulations were done with 5 000 neural networks for $\eta = 0.05$ and $\chi = 1$. For zero velocity the probability distribution of the difference between the weights and the probabilistic mean is symmetric around the origin. A slowly moving environment gives rise to a small delay and a slightly broader distribution. If the environmental change is relatively large, the probability distribution sincerely lags behind and is much broader.

The error defined in Eq. (23) yields in this example

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

For nonzero velocity the error diverges at $\eta = 2$ and $\eta = 0$ and has a global minimum for some $0 < \eta < 2$. This error is plotted in Fig. 5(a) as a function of the learning parameter. For small learning parameters, the error is dominated by the bias, for large learning parameters by the standard deviation. The optimal learning parameter can be found by minimization of this error $\mathcal{E}$. For small γ the optimal learning parameter is proportional to $\gamma^{2/3}$.

VI Nonlinear learning rules

In this section we will discuss the performances of neural networks operating in a time-dependent environment with a nonlinear learning rule. We will show that for slow changes and small

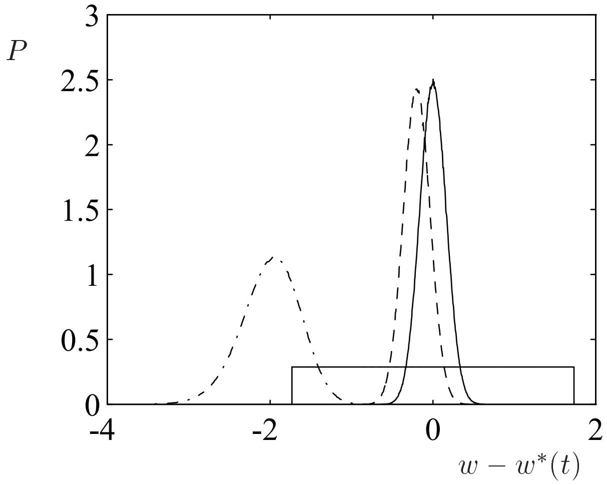

Figure 4: Simulated probability distribution for time-dependent Grossberg learning. Learning parameter $\eta = 0.05$, standard deviation input $\chi = 1.0$, 5 000 neural networks. The input probability distribution $\rho(x,t)$ is drawn for reference (solid box). Zero velocity (solid line). Small velocity: $v = 0.01/\tau$ (dashed line). Relatively large velocity: $v = 0.1/\tau$ (dash-dotted line).

learning parameters linear differential equations still give a useful description of the learning process.

Making again the expansions described in Sec. IV, we find

$$\begin{aligned} \frac{\tau}{\eta}\frac{dm_i(t)}{dt} &= -\sum_j G_{ij}(t)m_j(t) + \frac{1}{2}\sum_{jk} Q_{ijk}(t)\Sigma^2_{jk}(t) - \frac{\tau}{\eta}v_i(t)\;, \\ \frac{\tau}{\eta}\frac{d\Sigma^2_{ij}(t)}{dt} &= -\sum_k G_{ik}(t)\Sigma^2_{kj}(t) - \sum_k \Sigma^2_{ik}(t)G_{jk}(t) + \eta D_{ij}(t)\;, \end{aligned} \tag{25}$$

with definition $\mathbf{v}(t) \equiv \dot{\mathbf{w}}^*(t)$ and notation $G(t) \equiv G(\mathbf{w}^*(t))$, and so on. The approximate validity of these equations requires not only (15) and (17), but also

$$\begin{aligned} |\tau\mathbf{v}||\partial_w^2\mathbf{f}| &\ll \eta|\partial_w\mathbf{f}|^2\;, \\ |\tau\mathbf{v}||\partial_w D| &\ll \eta|D||\partial_w\mathbf{f}|\;. \end{aligned} \tag{26}$$

These conditions may be summarized as follows. Changes in the environment of order $v\tau$ must be small compared to the size of a learning step ηf. Eq. (25) then gives an approximate description of the learning process for times $t \gg \tau/\eta$, i.e., such that terms of the form $\exp[-\eta t/\tau]$ can be neglected.

For symmetric $G(t)$ we may rewrite Eq. (25) in the eigenvector directions α of the matrix $G(t)$. We make some further simplifications by assuming that the environmental changes are such that λ_α, v_α, $D_{\alpha\beta}$ and $Q_{\alpha\beta\gamma}$ are independent of time. This is true for the moving distribution in the Grossberg example (Sec. V) and for the example we will discuss next: an Oja network operating in a slowly rotating environment. Furthermore, the following analysis can be viewed as a zeroth-order approximation, which is valid if the changes of these parameters are insignificant on a time scale of θ_α, the response time defined in Eq. (20). We can calculate the asymptotic solution

$$m_\alpha = -\frac{\tau v_\alpha}{\eta\lambda_\alpha} + \frac{\eta}{2\lambda_\alpha}\sum_{\beta\gamma}\frac{Q_{\alpha\beta\gamma}D_{\beta\gamma}}{\lambda_\beta+\lambda_\gamma}\;,$$

$$\Sigma^2_{\alpha\beta} \;=\; \frac{\eta D_{\alpha\beta}}{\lambda_\alpha + \lambda_\beta} \,. \tag{27}$$

If in the expression for the bias m_α the second term due to the nonlinearity of the learning rule is much smaller than the first term due the environmental change, this solution corresponds to

$$\langle w_\alpha \rangle_{\Xi(t)} \;=\; w^*_\alpha(t - \theta_\alpha)\,,$$

with the delay θ_α equal to the response time defined in Eq. (20). The delay is inversely proportional to the learning parameter. In this special case we can also calculate the error $\mathcal{E}$, defined in Eq. (23), neglecting terms of order η^2:

$$\mathcal{E} \;=\; \frac{1}{\eta^2} \sum_\alpha \left(\frac{\tau v_\alpha}{\lambda_\alpha} \right)^2 \;+\; \eta \sum_\alpha \frac{D_{\alpha\alpha}}{2\lambda_\alpha} \,. \tag{28}$$

The optimal learning parameter is the learning parameter for which $\mathcal{E}$ is minimal:

$$\eta_{\rm optimal} \;=\; \sqrt[3]{\frac{\sum_\alpha \left(\tau v_\alpha/\lambda_\alpha\right)^2}{\sum_\alpha D_{\alpha\alpha}/4\lambda_\alpha}} \,. \tag{29}$$

The optimal learning parameter is proportional to $v^{2/3}$. Substitution of this optimal learning parameter in Eq. (28) yields $\mathcal{E}_{min} \propto v^{2/3}$. For minimal error the contributions of the bias and the standard deviation are of the same order of magnitude. Note that the requirements (15), (17) and (26) are all fulfilled for small changes v and $\eta \approx \eta_{\rm optimal}$.

We discuss some simulations with the nonlinear learning rule of Oja [Eq. (21)] in order to see whether they support our analysis. The neural network consisting of one neuron is still taught with random samples from a rectangle as in Sec. IV. But now we are rotating the rectangle with constant angular velocity ω around the axis which goes through the origin and is perpendicular to the rectangle. The principal component of the covariance matrix of the input distribution and thus $\mathbf{w}^*(t)$ follows:

$$\mathbf{w}^*(t) \;=\; \pm \left[\begin{array}{c} \cos(\omega t) \\ \sin(\omega t) \end{array} \right] .$$

Since in this example λ_α, v_α, $D_{\alpha\beta}$ and $Q_{\alpha\beta\gamma}$ are indeed independent of time, the results given by Eq. (27) are approximately valid if all corresponding conditions are satisfied. We can calculate the squared bias and the variance up to order η^2:

$$\begin{aligned} \|\mathbf{m}\|^2 \;&=\; \frac{1}{\eta^2} \left[\frac{\omega \tau}{\Lambda_1 - \Lambda_2} \right]^2 + \frac{\eta^2}{16} \left[\frac{\Lambda_2^2}{\Lambda_1 - \Lambda_2} \right]^2 \,, \\ {\rm Tr}\left[\Sigma^2\right] \;&=\; \frac{\eta}{2} \frac{\Lambda_1 \Lambda_2}{\Lambda_1 - \Lambda_2} + \frac{\eta^2}{4} \frac{\Lambda_1 \Lambda_2^2}{\Lambda_1 - \Lambda_2} \,. \end{aligned} \tag{30}$$

These terms are plotted in Fig. 5(b), together with the error $\mathcal{E}$ (solid line), which is just the sum of the squared bias (dashed line) and the variance (dash-dotted line). The computed values are reasonable estimates for the values found by simulations for learning parameters $\eta > 0.02$. For smaller learning parameters the conditions (26) are violated and agreement is not to be expected. From Eqs. (15), (17) and (26) it can be estimated that $0.01 \ll \eta \ll 0.3$. Substitution of all relevant parameters in Eq. (29) leads to the optimal learning parameter $\eta_{\rm optimal} = 0.043$. The optimal learning parameter in simulations is not much different.

VII Conclusions and discussion

We have set up a general framework for studying the asymptotic solutions of a large class of learning neural networks for nonzero asymptotic learning parameters η. The conditions for

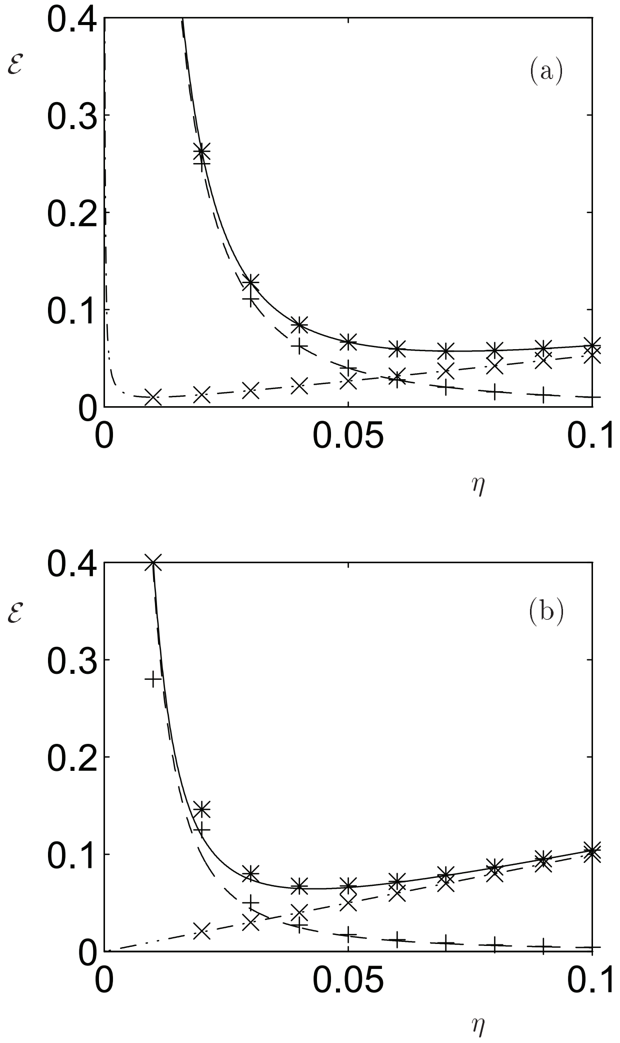

Figure 5: Error as a function of the learning parameter for learning processes in a changing environment. Squared bias (computed, dashed line; simulated, +), variance (computed, dash-dotted line; simulated, x) and error (computed, solid line; simulated, $*$). Simulations were done with 5 000 neural networks. (a) Grossberg learning. Standard deviation input: $\chi = 1.0$. Velocity: $v = 0.01/\tau$. (b) Oja learning. Eigenvalues of the covariance matrix of the input distribution, $\Lambda_1 = 2.0$ and $\Lambda_2 = 1.0$. Angular velocity, $\omega = 2\pi/1\ 000\tau$.

the validity of the framework in a fixed environment are given in Eqs. (15) and (17) and are roughly equivalent to $\eta\,\partial_w f \ll 1$. If the network has obtained a stationary representation of the environment, a nonzero learning parameter gives rise to fluctuations in the representation of the network proportional to η and allows the network to adapt to a new, different environment in a time which is inversely proportional to η. The size of these effects can be calculated analytically.

In a constantly changing environment, the analysis holds approximately [see the conditions in Eq. (26)] as long as the rate of change in the environment v is small in comparison with the "learning rate" $\eta f/\tau$. There is a trade-off between adaptability and accuracy: the more adaptable the network is, the less accurate it is, and vice versa. If an error criterion is defined which takes these two effects into account, the learning parameter has an optimal value which is proportional to $v^{2/3}$.

To be able to do the analysis above, we had to the make the following essential assumptions.

(i) Learning is described by a first order process as given by Eq. (1): the new network state $\mathbf{w} + \Delta\mathbf{w}$ depends only on the *present* network state $\mathbf{w}$ and on the training pattern $\vec{x}$.

(ii) At each learning step a training pattern $\vec{x}$ is drawn *at random* from the probability distribution $\rho(\vec{x}, t)$, i.e., the value of $\vec{x}$ drawn at time t is independent of previous values of $\vec{x}$. This assumption, in combination with the first assumption, enabled us to describe learning as a Markov process. Violation of either of these complicates the analysis significantly. For example, it is clear that this analysis is not directly applicable to learning processes concerning the storage of temporal sequences.

(iii) A physical time scale is introduced by drawing the time intervals between successive learning steps from a *Poisson distribution.* This is an elegant way to transform a discrete random-walk equation into a continuous time master equation for any value of the learning parameter η. It may also be applied to describe the dynamics of spin states in Hopfield-type neural networks in differential form, even for a finite number of neurons.

Acknowledgment

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

References

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

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

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

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

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

[6] 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.

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

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

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

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

[11] J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79:2554–2558, 1982.

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

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

[14] 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.