Machine Learning, 32, 179–201 (1998)

Statistical Mechanics of Online Learning of Drifting Concepts: A Variational Approach

RENATO VICENTE rvicente@if.usp.br
Instituto de Física, Universidade de São Paulo, CP66318, CEP 05315-970, São Paulo, SP Brazil

OSAME KINOUCHI osame@ultra3000.ifqsc.sc.usp.br
Instituto de Física de São Carlos, Universidade de São Paulo, CP 369, CEP 13560-970, São Carlos, SP Brazil

NESTOR CATICHA nestor@if.usp.br
Instituto de Física, Universidade de São Paulo, CP66318, CEP 05315-970, São Paulo, SP Brazil

Editors: Gerhard Widmer and Miroslav Kubat

Abstract. We review the application of statistical mechanics methods to the study of online learning of a drifting concept in the limit of large systems. The model where a feed-forward network learns from examples generated by a time dependent teacher of the same architecture is analyzed. The best possible generalization ability is determined exactly, through the use of a variational method. The constructive variational method also suggests a learning algorithm. It depends, however, on some unavailable quantities, such as the present performance of the student. The construction of estimators for these quantities permits the implementation of a very effective, highly adaptive algorithm. Several other algorithms are also studied for comparison with the optimal bound and the adaptive algorithm, for different types of time evolution of the rule.

Keywords: neural networks, concept learning, online algorithms, variational optimization

1. Introduction

The importance of universal bounds to generalization errors, in the spirit of the Vapnik-Chervonenkis (VC) theory, cannot be overstated, since these results are independent of target function and input distribution. These bounds are tight in the sense that for a particular target an input distribution can be found where generalization is as difficult as the VC bound states. However, for several learning problems, by making specific assumptions, it is possible to go further. Haussler et al. (1996) have found tighter bounds that even capture functional properties of learning curves, such as for example the occurrence of discontinuous jumps in learning curves, which cannot be predicted from VC theory alone.

These results were derived by adapting to the problem of learning ideas that arise in the context of statistical mechanics. In recent years many other results (Seung et al., 1992; Watkin et al., 1993; Opper & Kinzel, 1996), bounds or approximations, rigorous or not, have been obtained in the learning theory of neural networks by applying a host of methods originated in the study of disordered materials. These methods permit looking at the properties of large networks, where great analytical simplifications can occur; and also, they afford the possibility of performing averages over the randomness introduced by the training data. They are useful in that they give information about typical rather than, e.g., worst case behavior and should be regarded as complementary to those of computational learning theory.

The statistical mechanics of learning has been formulated either as a problem at thermodynamical equilibrium or as a dynamical process off-equilibrium, depending on the type of learning strategy. Although many intermediate regimes can be identified, we briefly discuss the two dynamical extremes. Batch or offline methods essentially give rise to the equilibrium formulation, while online learning can be better described as an off-equilibrium process.

The offline method begins by constructing a cost or energy function on the parameter space, which depends on *all the training data* simultaneously (Seung et al., 1992; Watkin et al., 1993). Learning occurs by defining a gradient descent process on the parameter space, subject to some (thermal) noise process, which permits to some extent escaping from local traps. In a very simplified way it may be said that, after some time, this process leads to "thermal equilibrium", when essentially all possible information has been extracted by the algorithm from the learning set. The system is now described by a stationary (Gibbs) probability distribution on parameter space.

On the other extreme lies online or incremental learning. Instead of training with a cost function defined over all the available examples, the online cost function depends directly on only *one single example*, independently chosen at each time step of the learning dynamics (Amari, 1967), (for a review, see (Mace & Coolen, 1998)). Online learning occurs also by gradient descent, but now the random nature of the presentation of the examples implies that at each learning step an effectively different cost function is being used. This can lead to good performance even without the costly memory resources needed to keep the information about the whole learning set, as is the case in the offline case.

Although most of the work has concentrated on learning in stationary environments with either online or offline strategies, the learning of drifting concepts has also been modeled using ideas of statistical mechanics (Biehl & Schwarze, 1992; Biehl & Schwarze, 1993; Kinouchi & Caticha, 1993). The natural approach to this type of problem is to consider online learning, since old examples may not be representative of the present state of the concept. It makes little sense, if any, to come to thermal equilibrium with possibly already irrelevant old data, as would be the case with an offline strategy. The possibility of forgetting old information and of preventing the system from reusing it, which are essential features to obtain good performance, are inherent to the online processes, as will be seen below.

We will model the problem of supervised learning, in the sense of Valiant (1984), of a drifting concept by defining a "teacher" neural network. Drift is modeled by allowing the teacher network parameters to undergo a drift that can be either random or deterministic. The dynamics of learning occurs in discrete time. At each time step, a random input vector is chosen independently from a distribution P_D, giving rise to a temporal stream of input-output pairs, where the output is determined by the teacher. From this set of data the student parameters will be built.

The question addressed in this paper concerns the *best* possible way in which the information can be used by the student in order to obtain maximum typical generalization ability. This is certainly too much to ask for and we will have to make some restrictions to the problem. This question will be answered by means of a *variational method* for the following class of exactly soluble models: a feed-forward boolean network learning from a teacher which is itself a neural network of similar architecture and learns by a Hebbian modulated mechanism.

This is still hard and further restrictions will be made. The thermodynamic limit (TL) will always be assumed. This means that the dimension N of parameter space is taken to infinity. This increase brings about great analytical simplifications. The TL is the natural regime to study in condensed matter physics. There the number of interacting units is of the order of $N \approx 10^{23}$ and fluctuations of macroscopic variables of order $\sqrt{N}$. In studying neural networks, results obtained in the TL ought to be considered as the first term in a systematic expansion in powers of $1/N$.

Once this question has been answered in a restricted setting, what does it imply for more general and realistic problems? The variational method has been applied to several models, including boolean and soft transfer functions, single layer perceptrons and networks with hidden units, networks with or without overlapping receptive fields and also for the case of non-monotonic transfer functions (Kinouchi & Caticha, 1992b; Kinouchi & Caticha, 1995; Copelli & Caticha, 1995; Simonetti & Caticha, 1996; Vicente & Caticha, 1997; Van den Broeck & Reimann, 1996). In solving the problem in different cases, different optimal algorithms have been found. But rather than delving in the differences, it is important to stress that a set of features is common to all optimal algorithms. Some of these common features are obvious or at least expected and have been incorporated into algorithms built in an *ad hoc* manner. Nevertheless, it is quite interesting to see them arise from theoretical arguments rather than heuristically. Moreover, the exact functional dependence is also obtained, and this can never be obtained just from heuristics. See (Opper, 1996) for an explicitly Bayesian formulation of online learning which in the TL seems to be similar to the variational method.

The first important result of the variational program is to give lower bounds on the generalization errors. But it gives more; the constructive nature of the method furnishes also an 'optimal algorithm'. However, the direct implementation of the optimal algorithm is not possible, as it relies on information that is not readily accessible. This reliance is not to be thought of as a drawback but rather as indicating what kind of information is needed in order to approximate, if not saturate, the optimal bounds. It indicates directions for further research where the aim should be on developing efficient estimation schemes for those variables.

The procedure to answer what is the best possible algorithm in the sense of generalization is as follows. The generalization error, in the TL, can be written as a function of a set of macroscopic parameters, sometimes referred to as 'order parameters', by borrowing the nomenclature from physics. The online dynamics of the weights (microscopic variables) induces a dynamics of the order parameters, which in the TL is described by a **closed set** of coupled differential equations. The evolution of the generalization error is thus a functional of the cost function gradient which defines the learning algorithm. The gradient of the cost function is usually called the modulation function. The local optimization (see (Rattray & Saad, 1997) for global) is done in the following way. Taking the functional derivative of the rate of decay of the generalization error, with respect to the modulation function, equal to zero, permits determining the modulation function that extremizes the mean decay at each time step. This extremum represents, in many of the interesting cases, a maximum (see (Vicente & Caticha, 1997) for exceptions). We can thus determine the modulation function, i.e., the algorithm, that leads to the fastest local decrease of the generalization error under several restrictions, to be discussed below.

In this paper several online algorithms are analyzed for the boolean single layer perceptron. Other architectures, with e.g., internal layers of hidden units, can be analyzed, although there is a need for laborious modifications of the methods. Examples of random drift, deterministic evolution, changing drift levels and piecewise constant concepts are presented. The paper is organized as follows. In Section 2, the variational approach is briefly reviewed. In Section 3, analytical results and simulations are presented for several algorithms in the cases of random drift and deterministic 'worst-case' drift, where the teacher "flees" from the student in weight space. The asymptotics of the different algorithms are characterized by a couple of exponents, β, the learning exponent and δ, the drift or residual exponent. A relation between these exponents is obtained. A practical adaptive algorithm is discussed in Section 4, where it is applied to a problem with changing drift level. In Section 5, the Wisconsin test for perceptrons is studied. Numerical results for the piecewise constant rule are presented. Concluding remarks are presented in Section 6.

2. The Variational Approach

The mathematical framework employed in the statistical mechanics of online learning and in the variational optimization are quickly reviewed in this section. We consider only the simple perceptron with no hidden layer. For extensions to other architectures see (Kinouchi & Caticha, 1995; Copelli & Caticha, 1995; Simonetti & Caticha, 1996; Vicente & Caticha, 1997; Van den Broeck & Reimann, 1996).

2.1. Preliminary Definitions

The boolean single layer perceptron is defined by the function $\sigma_B = sign(\mathbf{B} \cdot \mathbf{S})$, with $\mathbf{S} \in \mathcal{R}^N$, parametrized by the *concept* weight vector $\mathbf{B} \in \mathcal{R}^N$, also called *synaptic vector*.

In the student-teacher scenario that we are considering, a perceptron (*teacher*) generates a sequence of statistically independent training pairs $\mathcal{L} = \{(\mathbf{S}^\mu, \sigma_B^\mu) : \mu = 1, ..., p\}$, and another perceptron (*student*) is constructed, using only the examples in $\mathcal{L}$, in order to infer the concept represented by the teacher's vector. The teacher and student are respectively defined by weight vectors $\mathbf{B}$ and $\mathbf{J}$ with norms denoted by B and J.

In the presence of noise, instead of σ_B, the student has access only to a corrupted version $\tilde{\sigma}_B$. For example, for *multiplicative* noise, each teacher output is flipped independently with probability χ (Biehl et al., 1995; Copelli et al., 1996b; Copelli, 1997; Heskes, 1994):

$$P(\tilde{\sigma}_B|\sigma_B) = (1-\chi)\delta(\sigma_B, \tilde{\sigma}_B) + \chi\delta(\sigma_B, -\tilde{\sigma}_B)\,, \tag{1}$$

where $\sigma_B = sign(y)$, and $y = \mathbf{B} \cdot \mathbf{S}/B$ is the *normalized field*. The Kronecker δ is 1 (0) only if the arguments are equal (different). In the same way, for the student, the field $x = \mathbf{J} \cdot \mathbf{S}/J$ and the output $\sigma_J = sign(x)$ are defined.

The definition of a global cost function $E_{\mathcal{L}}(\mathbf{J}) = \sum_\mu E^\mu(\mathbf{J})$, over the entire data set $\mathcal{L}$, is required for batch or offline learning. The interaction among the partial potentials $E^\mu(\mathbf{J})$ may generate spurious local minima, leading to metastable states and possibly very long thermalization times. This can be avoided to a great extent by learning online.

We define a discrete dynamics where at each time step a weight update is performed along the gradient of a partial potential $E^\mu(\mathbf{J})$, which depends on the randomly chosen

μ^{th} example. This random sampling of partial potentials introduces fluctuations which tend to decrease as the system approaches a (hopefully) global minimum. That process has been recently called *self-annealing* (Hondou, 1996) in contrast to the external parameter dependent simulated annealing. The general conditions for convergence of online learning to a global minimum, even in stationary environments, is an open problem of major current interest.

The online dynamics can be represented by a finite difference equation for the update of weight vectors. For each new random example, make a small correction of the current student, in the direction opposite to the gradient of the partial potential and also allow for a restriction of the overall length of the weight vector to prevent runaway behavior:

$$\mathbf{J}^{\mu+1} = \mathbf{J}^{\mu} - \Delta t \Omega^{\mu} \mathbf{J}^{\mu} - \Delta t \nabla_{\mathbf{J}} E^{\mu} \, . \tag{2}$$

Here the partial potential E^{μ} is a function of the scalars that are accessible to the student (field x, norm J and output $\tilde{\sigma}_B$) and corresponds to the randomly sampled example pair $(\mathbf{S}^{\mu}, \tilde{\sigma}_B^{\mu})$. The time scale Δt must be $\Delta t \sim \mathcal{O}(1/N)$ so that in the TL we can derive well-behaved differential equations; in general we will choose $\Delta t = 1/N$. The second term allows to control the norm of the weight vector $\mathbf{J}$.

It is easy to see that $\nabla_{\mathbf{J}} E^{\mu} = (\partial E^{\mu}/\partial x) \nabla_{\mathbf{J}} x$. The calculation of the gradient and the definition $\partial E^{\mu}/\partial x = -J^{\mu} W^{\mu}(\mathcal{V}) \tilde{\sigma}_B^{\mu}$ finally lead to the online dynamics in the form:

$$\mathbf{J}^{\mu+1} = \left(1 - \frac{\Omega^{\mu}}{N}\right) \mathbf{J}^{\mu} + \frac{1}{N} J^{\mu} W^{\mu}(\mathcal{V}) \tilde{\sigma}_B^{\mu} \mathbf{S}^{\mu} \, . \tag{3}$$

Note that each example pair $(\mathbf{S}^{\mu}, \tilde{\sigma}_B^{\mu})$ is used only once to update the student's synapses, $\tilde{\sigma}_B^{\mu} \mathbf{S}^{\mu}$ is called the *Hebbian term* and the intensity of each modification is given by the *modulation function* W. The factor $J \tilde{\sigma}_B^{\mu}$ can be absorbed into the modulation function, but has been explicitly written for convenience. The single most important fact is that the relevant change is made along the direction of the input vector $\mathbf{S}^{\mu}$. This is not the most general update rule since non parallel changes could be considered. However we will concern ourselves with the TL, since it is only there that bounds can be derived. In the TL and in the absence of correlations between different inputs, i.e., $\langle S_i S_j \rangle = 0$, the prefactor of $\mathbf{S}^{\mu}$ is a diagonal matrix (Opper, 1996). Furthermore the class of modulated Hebbian algorithms is interesting, from a biological perspective, even for finite N. The symbol $\mathcal{V}$ denotes the *learning situation* defined by the set of quantities that we are allowed to use in the modulation function, that is, the available information. For boolean perceptrons, $\mathcal{V}$ may contain the corrupted teacher's output $\tilde{\sigma}_B$, the field x, and as discussed below, some information about the generalization error. We can still study the restrictions $\mathcal{V} = \{\tilde{\sigma}_B\}$, $\mathcal{V} = \{\tilde{\sigma}_B, \sigma_J\}$ and $\mathcal{V} = \{\tilde{\sigma}_B, \mid x \mid\}$. Evidently, the more information the student has, the better we expect it to learn.

We consider a specific model of concept drift introduced by Biehl and Schwarze (1992). The drift that can be followed by an online learning system can not be too large for it would be impossible to track, but if too slow it trivially reduces to an effectively stationary problem. Their choice, which makes the problem interesting, is as follows. At each time step the concept vector $\mathbf{B}$ suffers the influence of the changing environment and evolves as

$$\mathbf{B}^{\mu+1} = \left(1 - \frac{\Lambda^{\mu}}{N}\right) \mathbf{B}^{\mu} + \frac{1}{N} \vec{\eta}^{\mu} \, , \tag{4}$$

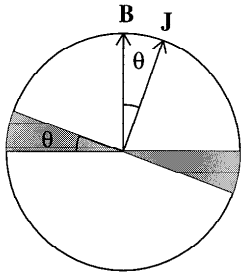

Figure 1. Simple representation of weight vectors in the hyper-sphere. The teacher and the student disagree when the input vector **S** is inside the shaded region.

where Λ controls the norm B and $\vec{\eta} \in \mathcal{R}^N$ is the *drift vector*. Random and deterministic versions of $\vec{\eta}$ will be considered in Section 4.

The performance of a specific student $\mathbf{J}$ on a given concept $\mathbf{B}$ can be measured by the generalization error e_G that is defined as the instantaneous average error $\epsilon = \frac{1}{2}(1-\sigma_J\sigma_B) = \frac{1}{2}\Theta(-xy)$ (σ_B is the non-corrupted output and $\Theta(x)$ is the step function) over inputs extracted from the uniform distribution $P_{\mathcal{U}}(\mathbf{S})$ with support over the hyper-sphere of radius $\sqrt{N}$:

$$e_G(\mathbf{J},\mathbf{B}) = \int d\mathbf{S} P_{\mathcal{U}}(\mathbf{S})\ \epsilon\left(\sigma_J(\mathbf{S}),\sigma_B(\mathbf{S})\right)\ . \tag{5}$$

We make explicit the difference between e_G and the prediction error e_P, which measures the average $\epsilon_P = \langle \frac{1}{2}(1-\sigma_J\tilde{\sigma}_B)\rangle$, over the *true* distribution of examples P_D. It is not difficult to see that the expression for e_G is invariant under rotations of axes in $\mathcal{R}^N$, therefore the integral (5) depends only on the scalars $\rho = \mathbf{B}\cdot\mathbf{J}/BJ$, x, y, B and J. As x and y are sums of independent random variables (S_iJ_i/J and S_iB_i/B, respectively), in the TL a straightforward application of the central limit theorem leads to (see e.g., (Opper et al., 1990; Seung et al., 1992; Watkin et al., 1993)):

$$\begin{aligned} e_G(\rho) &= \int dxdy P_{\mathbf{C}}(x,y)\ \frac{\Theta(-xy)}{2} \\ &= \frac{1}{\pi}\arccos\rho\ . \end{aligned} \tag{6}$$

$P_{\mathbf{C}}(x,y)$ is a Gaussian distribution in $\mathcal{R}^2$ with correlation matrix

$$\mathbf{C} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\ .$$

Note that ρ is a parameter in the probability distribution describing the fields, and J and B define the scale of the fields. In statistical physics, these parameters are called *macroscopic variables*. It is interesting to note that the number of macroscopic variables depends on the symmetry of the input distribution $P_{\mathcal{U}}(\mathbf{S})$.

The intuitive meanings of ρ and of the Eq. 6 can be verified with the help of Figure 1. Observing that $\rho = \cos\theta$ and that for the boolean perceptron the weight vectors are normal to a hyper-plane that divides the hyper-sphere in two differently labeled hemispheres, it is easy to see that the student and the professor disagree on the labeling of input vectors $\mathbf{S}$ inside the shaded region, thus trivially $e_G = \theta/\pi = \frac{1}{\pi}\arccos\rho$.

2.2. Emergence of the Macroscopic Dynamics

The dimensionality of the dynamical system involved can be reduced by using (4) and (3) to write a system of coupled difference equations to the macroscopic variables:

$$\begin{aligned}\rho^{\mu+1} &= \rho^{\mu} + \frac{\rho^{\mu}}{N}\left[W^{\mu}(\mathcal{V})\left(\frac{y^{\mu}\tilde{\sigma}_B^{\mu}}{\rho^{\mu}} - \Delta^{\mu}\right) - \frac{1}{2}(W^{\mu}(\mathcal{V}))^2\right] \\ &+ \frac{1}{N}\left[\frac{\mathbf{J}^{\mu}\cdot\vec{\eta}^{\mu}}{J^{\mu}} - \rho^{\mu}\Lambda^{\mu} + \frac{\mathbf{S}^{\mu}\cdot\vec{\eta}^{\mu}}{N}\tilde{\sigma}_B^{\mu}W^{\mu}(\mathcal{V})\right] + \mathcal{O}\left(\frac{1}{N^2}\right) , \qquad (7)\\ J^{\mu+1} &= J^{\mu} + \frac{J^{\mu}}{N}\left(W^{\mu}(\mathcal{V})\Delta^{\mu} + \frac{1}{2}(W^{\mu}(\mathcal{V}))^2 - \Omega^{\mu}\right) + \mathcal{O}\left(\frac{1}{N^2}\right) , \qquad (8)\\ B^{\mu+1} &= B^{\mu} + \frac{1}{N}\left(\frac{\vec{\eta}^{\mu}\cdot\mathbf{B}^{\mu}}{B} - \Lambda^{\mu}B^{\mu}\right) + \frac{\vec{\eta}^{\mu}\cdot\vec{\eta}^{\mu}}{2BN^2} + \mathcal{O}\left(\frac{1}{N^2}\right) . \qquad (9)\end{aligned}$$

In the above equations the *local stability* $\Delta = x\tilde{\sigma}_B$ was introduced. Positive stability means that the student classification $\sigma_J = sign(x)$ agrees with the (noisy) learning data $\tilde{\sigma}_B$.

The usefulness of the TL lies in the possibility of transforming the stochastic difference equations into a closed set of deterministic differential equations (Kinzel & Ruján, 1990; Kinouchi & Caticha, 1992a). The idea is to choose a continuous time scale α such that for the TL regime $p/N \to \alpha$, where p is the number of examples already presented. The equations are then averaged over the input vectors $\mathbf{S}$ and drift vector $\vec{\eta}$ distributions, leading to:

$$\begin{aligned}\frac{d\rho}{d\alpha} &= \rho\left\langle W\left(\frac{y\tilde{\sigma}_B}{\rho} - \Delta\right) - \frac{1}{2}W^2\right\rangle + \left\langle\frac{\mathbf{J}\cdot\vec{\eta}}{J} - \rho\Lambda + C_{S\eta}\tilde{\sigma}_B W\right\rangle , \qquad (10)\\ \frac{dJ}{d\alpha} &= J\left\langle W\Delta + \frac{1}{2}W^2 - \Omega\right\rangle , \qquad (11)\\ \frac{dB}{d\alpha} &= \left\langle\frac{\vec{\eta}\cdot\mathbf{B}}{B} - \Lambda B + C_{\eta\eta}\right\rangle , \qquad (12)\end{aligned}$$

where $\langle ...\rangle = \int d\vec{\eta}\, d\mathbf{S}\,(\ldots)P(\vec{\eta},\mathbf{S})$ and the definitions $C_{S\eta} = \lim_{N\to\infty}(\mathbf{S}\cdot\vec{\eta})/(NB)$ and $C_{\eta\eta} = \lim_{N\to\infty}(\vec{\eta}\cdot\vec{\eta})/(2BN)$ have been used.

The fluctuations in the stochastic equations vanish in the TL and the above equations become exact (*self-averaging property*). This can be proved by writing the Fokker-Planck equations for the finite N stochastic process defined in (7), (8) and (9), and showing that the diffusive term vanishes in the TL (Mace & Coolen, 1998).

2.3. *Variational Optimization of Algorithms*

The variational approach was proposed in (Kinouchi & Caticha, 1992b) as an analytical method to find learning algorithms with optimal mean decay (per example) of the generalization error. The same method was applied in several architectures and learning situations.

The idea is to write:

$$\frac{\mathrm{d}e_{\mathrm{G}}}{\mathrm{d}\alpha} = \frac{\partial e_{\mathrm{G}}}{\partial \rho}\frac{\mathrm{d}\rho}{\mathrm{d}\alpha}\,, \tag{13}$$

and use the macroscopic dynamics equation (10) to build:

$$\frac{de_G}{d\alpha}[W] = \frac{\partial e_G}{\partial \rho}\left[\rho\left\langle W\left(\frac{y\tilde{\sigma}_B}{\rho}-\Delta\right)-\frac{1}{2}W^2\right\rangle+\left\langle\frac{\mathbf{J}\cdot\vec{\eta}}{J}-\rho\Lambda+C_{S\eta}\tilde{\sigma}_B W\right\rangle\right]. \tag{14}$$

Each modulation function leads to a specific macroscopic dynamics and, correspondingly, to a mean decay error. The above equation captures explicitly the dependence of the decay on the modulation function. To emphasize that $de_G/d\alpha$ is a function of a function W we enclose the argument in square brackets and refer to $de_G/d\alpha[W]$ as a *functional*. Thus, the optimization is attained by imposing the extremum condition:

$$\frac{\delta}{\delta W(\mathcal{V})}\left(\frac{de_G}{d\alpha}[W(\mathcal{V})]\right)_{W=W^*}=0\,, \tag{15}$$

where $\delta/\delta W(\mathcal{V})$ stands for the *functional derivative* in the subspace of modulation functions W with dependence in the set $\mathcal{V}$. The above equation is analogous to those involving usual derivatives and can be solved observing that $\partial e_G/\partial\rho \neq 0$ and

$$\frac{\delta}{\delta W(\mathcal{V})}\langle f(\mathcal{H},\mathcal{V})W^n(\mathcal{V})\rangle = n\langle f(\mathcal{H},\mathcal{V})\rangle_{\mathcal{H}|\mathcal{V}}\,W^{n-1}(\mathcal{V})\,, \tag{16}$$

f is an arbitrary function, $\mathcal{H}$ is the set of *hidden* variables, that is, in contrast to the set $\mathcal{V}$, the variables not accessible to the student (e.g., fields y, drift vector $\vec{\eta}$, etc ...) and $\langle ...\rangle_{\mathcal{H}|\mathcal{V}} = \int d\mathcal{H}P(\mathcal{H}|\mathcal{V})...$, where for a given set $\mathcal{H}=\{a_1,a_2,..\}$, $d\mathcal{H}=da_1da_2....$ The solution is given by:

$$W^*(\mathcal{V}) = \left\langle\frac{(y+C_{S\eta})\tilde{\sigma}_B}{\rho}-\Delta\right\rangle_{\mathcal{H}|\mathcal{V}}. \tag{17}$$

By writing $y+C_{S\eta}=(\mathbf{B}+\vec{\eta})\cdot\mathbf{S}/\sqrt{B}$ it can be seen that the optimal algorithm "tries" to pull the example stability Δ, not to an estimative of the present teacher stability, but to one already corrected by the effect of the drift $\vec{\eta}$. It seems natural to concentrate on cases where the drift and the input vectors are uncorrelated ($C_{S\eta}=0$).

The optimization under different conditions of information availability, i.e., different specifications of the sets $\mathcal{V}$ and $\mathcal{H}$, leads to different learning algorithms. This can be seen by performing the appropriate averages in (17), as we proceed to show:

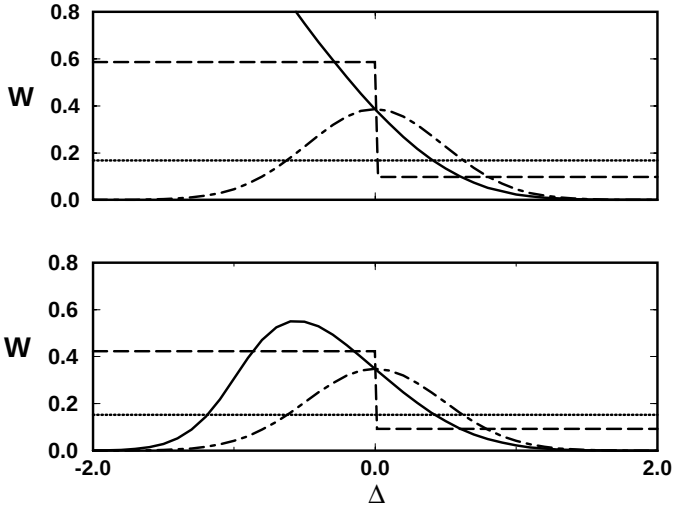

Figure 2. Modulation functions exemplified for $\rho = 0.9$ with noise levels $\chi = 0$ (top) and $\chi = 0.1$ (bottom): Annealed Hebb (dots), Step (dashes), Symmetric (dashes-dots) and Optimal (solid).

Annealed Hebb Algorithm: Suppose that the available information is limited to the corrupted teacher output. This corresponds to the learning situation such that $\mathcal{H} = \{y, \vec{\eta}, |x|, \sigma_J\}$ and $\mathcal{V} = \{\tilde{\sigma}_B\}$. The optimal algorithm for this situation is given by the following modulation function (see appendix A for a detailed derivation):

$$W_{AH}(\tilde{\sigma}_B; \rho, \chi) = \sqrt{\frac{2}{\pi}} \lambda^2 \rho (1 - \chi) \, . \tag{18}$$

The weight changes are proportional to the Hebb factor, but the modulation function does not depend on the example stability Δ (see Figure 2). Hence the name Hebb. However this function is not constant in time, the temporal evolution (*annealing*) is automatically incorporated into the modulation function. Optimal annealing is achieved by having the modulation function depend on the parameter ρ, the normalized overlap between the student $\mathbf{J}$ and the concept $\mathbf{B}$. Since this quantity is certainly not available to the student, there will be a need to complement the learning algorithm with an efficient estimator of the present level of performance by the student (see Section 4).

Step Algorithm (Copelli, 1997): This algorithm, obtained under the restriction $\mathcal{H} = \{y, \vec{\eta}, |x|\}$ and $\mathcal{V} = \{\tilde{\sigma}_B, \sigma_J\}$, is a close relative to Rosenblatt's original perceptron algorithm, which works by error correcting and treats all the errors in the same manner. There are two important differences, however, since correct answers also cause (smaller) corrections and furthermore, the size of the corrections evolves in time in a similar manner to the annealed Hebb algorithm.

The modulation function is

$$W_{Step}(\tilde{\sigma}_B, \sigma_J; \rho, \chi) = \frac{1}{\sqrt{2\pi}} \lambda^2 \rho (1-\chi) \frac{1}{\left[\frac{\chi}{2} + \frac{(1-\chi)}{\pi} \arccos(-\rho \tilde{\sigma}_B \sigma_J)\right]} . \quad (19)$$

Note that the step algorithm has access to the student's output and can differentiate between right ($\Delta > 0$) and wrong ($\Delta < 0$) classifications. The name arises from the form of its modulation function (Figure 2). The annealing increases the height of the step, i.e., the difference between right and wrong, as the overlap ρ goes to one.

Symmetric Weight Algorithm (see (Kinouchi & Caticha, 1992a)): This is the optimal algorithm for the learning situation described by $\mathcal{H} = \{y, \vec{\eta}, \sigma_J\}$ and $\mathcal{V} = \{\tilde{\sigma}_B, |x|\}$. The resulting modulation function is given by:

$$W_{SW}(\tilde{\sigma}_B, |x|; \rho, \chi) = \sqrt{\frac{2}{\pi}} \lambda (1-\chi) e^{-x^2/2\lambda^2}. \quad (20)$$

That algorithm cannot discern between wrong and right classifications, but only differentiates between "easy" (large $\mid \Delta \mid$) and "hard" (small $\mid \Delta \mid$) classifications, concentrating the learning in "hard" examples (Figure 2).

Optimal Algorithm (see Kinouchi & Caticha, 1992b; Biehl et al., 1995; Copelli et al., 1996b)): When all the available information is used we have the learning situation described by $\mathcal{H} = \{y, \vec{\eta}\}$ and $\mathcal{V} = \{\tilde{\sigma}_B, |x|, \sigma_J\}$. The optimal algorithm is then given by:

$$W_{OPT}(\Delta = \tilde{\sigma}_B x; \rho, \chi) = \frac{1}{\sqrt{2\pi}} \lambda (1-\chi) \frac{e^{-\Delta^2/2\lambda^2}}{\chi/2 + (1-\chi) H(-\Delta/\lambda)} . \quad (21)$$

In the presence of noise, a crossover is built into the optimal modulation function. This crossover is from a regime where the student classification is not strongly defined (Δ negative but small)—and the information from the teacher is taken seriously—to a regime where the student is confident on its own answer and any strong disagreement (very negative Δ) with the teacher will be attributed to noise, and thus effectively disregarded. The scale of the stabilities where the crossover occurs depends on the level of performance ρ and therefore is also annealed.

The learning mechanisms are highly adaptive and remain the same in the case of drifting rules, where the common features described above, mainly the ρ dependent annealing, lead automatically to a forgetting mechanism without the need to impose it, based on heuristic expectations, in an *ad hoc* manner.

It is interesting to note that the heuristically proposed algorithms are approximations of these optimized modulation functions. For instance, the simple Hebb rule is the annealed Hebb when $\vec{\eta} = 0$, since it can be shown in this case that $WJ = 1$, and corresponds to the $\rho \to 0$ regime of all the optimized algorithms; Rosenblatt's perceptron algorithm is qualitatively similar to the step algorithm; Adatron (Anlauf & Biehl, 1989; Watkin et al., 1993) approximates the optimal algorithm for $\chi = 0$ and $\rho \to 1$; OLGA (Kim & Sompolinsky, 1996) and thermal perceptron (Frean, 1992) algorithms resemble the optimal modulation with $\chi > 0$.

3. Learning Drifting Concepts

The important result from last section is that, under the assumption of uncorrelated drift and input vectors, the modulation functions do not depend on the drift parameters (in contrast to the explicit dependence on the examples' noise level χ). So, they are expected to be robust to continuous or abrupt, random or deterministic, concept changes. In this section simple instances of drifting concepts are examined; abrupt changes are studied in Section 5.

3.1. Random Drift

In this scenario, the concept weight vector $\mathbf{B}$ performs a random walk on the surface of a N-dimensional unit sphere. The drift vector has random components with zero mean and variance $2D$,

$$\langle \eta_i^\mu \eta_j^\nu \rangle = 2D\delta_{ij}\delta_{\mu\nu} \,, \tag{22}$$

The condition $\mathbf{B}^{\mu+1}\cdot\mathbf{B}^{\mu+1}=1$ is imposed in (12) by considering that $\mathbf{B}^0\cdot\mathbf{B}^0=1$ and with the choice $\Lambda=\vec{\eta}\cdot\mathbf{B}+D$.

The order of magnitude of the scaling of the drift vector with N is important since it gives the correct time scale for non-trivial behavior: if smaller, the drift would be irrelevant in the time scale of learning, while if larger it would not allow any tracking. In the relevant regime, the task is nontrivial also in another sense: the autocorrelation of the concept vector decays exponentially in the α-scale, $\langle \mathbf{B}(\alpha)\mathbf{B}(\alpha')\rangle \propto e^{-D(\alpha-\alpha')}$.

For the optimized algorithms, the equation for ρ decouples from the equation for J. After the proper averages, the learning equation for this type of drift reduces to

$$\frac{d\rho}{d\alpha} = \rho\left\langle W\left(\frac{y\tilde{\sigma}_B}{\rho}-\Delta\right)-\frac{1}{2}W^2\right\rangle - \rho D \,, \tag{23}$$

This equation can be solved for each particular modulation function W. The generalization errors $e_G=\frac{1}{\pi}\arccos(\rho)$ for the several algorithms described in the last section are compared in Figure 3 for the noiseless case. Solid curves refer to integrations of the above learning equation and symbols correspond to simulation results. Although the rule is continuously changing, it can be tracked within a stationary error e_G^∞ which depends on the drift amplitude D. The functions $e_G^\infty(D)$ for the various algorithms can be found from the condition $d\rho/d\alpha=0$ and are shown in Figure 4.

The behavior for small drift is shown in Table 1. Note the abrupt change in the exponents due to the inclusion of more information than the output $\tilde{\sigma}_B$.

Table 1. Small drift exponents: Random case.

	$e_G^\infty(D)$	$e_G(D=0,\chi=0)$
Annealed Hebb	$\left(\frac{D}{\pi^3}\right)^{1/4} \approx 0.42\, D^{1/4}$	$0.40\,\alpha^{-1/2}$
Symmetric	$\left(\frac{\sqrt{2}}{\pi^2}\right)^{1/3} D^{1/3} \approx 0.52\, D^{1/3}$	$1.41\,\alpha^{-1}$
Step	$\frac{(4)^{1/3}}{\pi} D^{1/3} \approx 0.51\, D^{1/3}$	$1.27\,\alpha{-}1$
Optimal	$\left(\frac{2}{\pi^2 A^2}\right)^{1/3} D^{1/3} \approx 0.45\, D^{1/3}$	$0.88\,\alpha^{-1}$

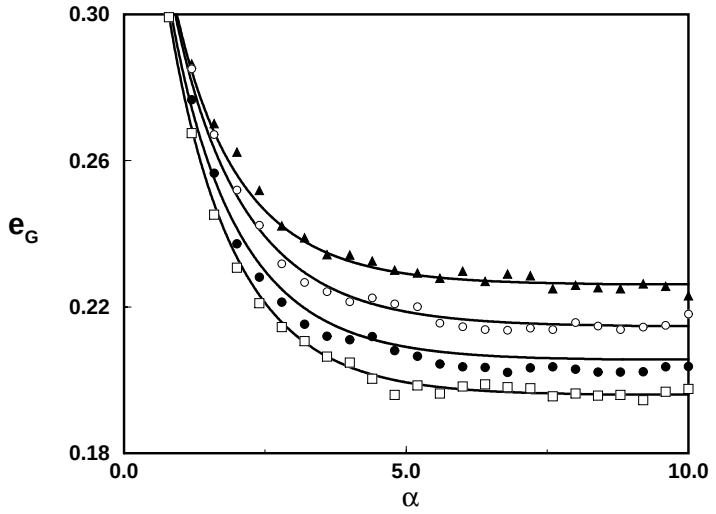

Figure 3. Integration of learning equations and simulation results ($N = 5000$) for random drift $D = 0.1$: Annealed Hebb (triangles), Symmetric (white circles), Step (black circles) and Optimal (white squares). The self-averaging property is clear since the simulation results refer to only one run.

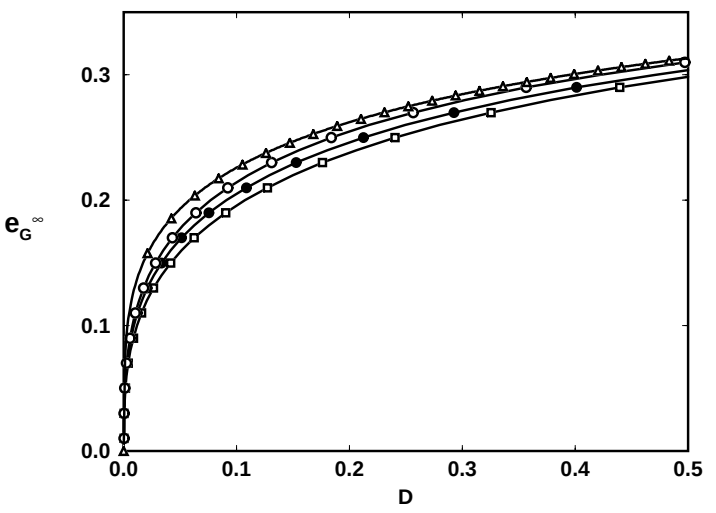

Figure 4. Asymptotic error $e_G^\infty(D)$ for random drift: Annealed Hebb (triangles), Symmetric (white circles), Step (black circles) and Optimal (white squares).

3.2. Deterministic Drift

In the learning scenario considered so far, the worst case drift will occur when at each time step the concept is changed deterministically so that the overlap with the current student vector is minimized. In this situation, previously examined by Biehl & Schwarze (1993), the new concept is chosen by minimizing $\mathbf{B}^{\mu+1} \cdot \mathbf{J}^{\mu}$ subject to the conditions

$$\begin{aligned} \mathbf{B}^{\mu+1} \cdot \mathbf{B}^{\mu} &= 1 - \frac{D}{N^2} , \\ \mathbf{B}^{\mu+1} \cdot \mathbf{B}^{\mu+1} &= 1 , \end{aligned} \tag{24}$$

where now D is the drift amplitude for the deterministic case. Note the different scaling with N for non trivial behavior.

Clearly $\mathbf{B}^{\mu+1}$ lies in the same plane which contains $\mathbf{B}^{\mu}$ and $\mathbf{J}^{\mu}$. The solution of this constrained minimization problem is

$$\begin{aligned} \mathbf{B}^{\mu+1} &= a\mathbf{B}^{\mu} - b\mathbf{J}^{\mu} , \\ a &= 1 - \frac{D}{N^2} + bJ\rho, \\ b &= \frac{1}{JN}\left[\frac{2D - (D/N)^2}{1-\rho^2}\right]^{1/2} . \end{aligned} \tag{25}$$

In terms of Λ and $\vec{\eta}$ of eq. 4 is

$$\Lambda = (1-a)N , \quad \vec{\eta} = -b\mathbf{J}N . \tag{26}$$

The learning equation reduces to

$$\frac{d\rho}{d\alpha} = \rho\left\langle W\left(\frac{y\tilde{\sigma}_B}{\rho} - \Delta\right) - \frac{1}{2}W^2\right\rangle - \sqrt{2D(1-\rho^2)} , \tag{27}$$

and the analysis is similar to the previous subsection. Theoretical error curves confirmed by simulations are shown in Figure 5 for the different algorithms with fixed drift amplitude. The stationary tracking error $e_G^{\infty}(D)$ is shown in Figure 6.

Table 2. Small drift exponents: Deterministic case.

	$e_G^{\infty}(D)$	$e_G(D=0,\chi=0)$
Annealed Hebb	$\left(\frac{\sqrt{2D}}{\pi^2}\right)^{1/3} \approx 0.52\, D^{1/6}$	$0.40\, \alpha^{-1/2}$
Symmetric	$\left(\frac{4D}{\pi^2}\right)^{1/4} \approx 0.80\, D^{1/4}$	$1.41\, \alpha^{-1}$
Step	$\frac{2^{5/4}}{\pi} D^{1/4} \approx 0.76\, D^{1/4}$	$1.27\, \alpha^{-1}$
Optimal	$\left(\frac{8D}{A^2}\right)^{1/4} \approx 0.63\, D^{1/4}$	$0.88\, \alpha^{-1}$

The behavior for small drift D is shown in Table 2. Again we can note the occurrence of an abrupt change in the exponents after the inclusion of information on the student's fields.

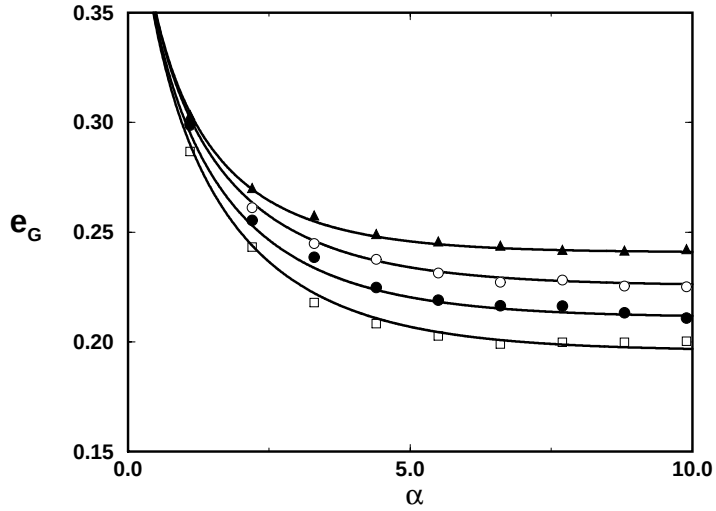

Figure 5. Integration of learning equations and simulation results ($N = 5000$) for deterministic drift $D = 0.01$: Annealed Hebb (triangles), Symmetric (white circles), Step (black circles) and Optimal (white squares).

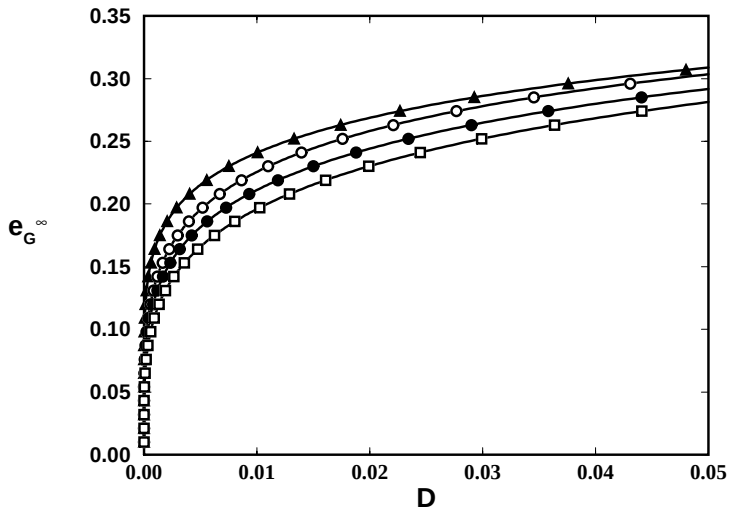

Figure 6. Asymptotic error $e_G^\infty(D)$ for deterministic drift: Annealed Hebb (triangles), Symmetric (white circles), Step (black circles) and Optimal (white squares).

3.3. *Asymptotic Behavior: Critical Exponents and Universality Classes for Unlearnable Problems*

A simple, although partial, measure of the performance of a learning algorithm can be given by the asymptotic decay of the generalization error in the driftless case and alternatively by the residual error dependence on D in the presence of drift. These are not independent aspects, but rather linked in a manner reminiscent of the relations between the different exponents that describe power law behavior in critical phenomena.

In the absence of concept drift, the generalization error decays to zero when α approaches α_c (which here happens to be infinity but may have a finite value in other situations (Watkin et al., 1993)) as a power law of the number of examples with the so called *learning* or static exponent β,

$$e_G \propto \tau^{\beta} \,, \tag{28}$$

where $\tau \equiv \frac{1}{\alpha} - \frac{1}{\alpha_c}$. Thus, we may think of e_G as a kind of *order parameter* in the sense that it is a quantity which changes from zero to a finite value at $\tau = 0$. We may think of $1/\alpha$ as the analog of the *control parameter* T (temperature) in critical phenomena.

Any amount of drift in the concept changes the problem from a learnable to an unlearnable one, with a residual error $e_G^{\infty} \equiv e_G(\tau = 0)$. We have seen that this behavior at the critical point $\tau = 0$ also obeys a power law

$$e_G^{\infty} \propto D^{1/\delta} \,, \tag{29}$$

where δ has been called the *drift exponent*.

In principle, we can classify different unlearnable situations (due to say, various kinds of concept drift), by the two exponents β and δ. Different algorithms and learning situations may have the same exponents. We can thus define, in a spirit similar to that in the study of critical phenomena, the so-called *universality classes* of behavior. In this paper we have seen four classes of behavior summarized in Table 3. In the absence of drift, $\beta = 1/2$ for the Hebb algorithm and $\beta = 1$ for the symmetric, step and optimal algorithms we have introduced above. There exist, however, other classes. For example, for the standard Rosenblatt perceptron algorithm with fixed learning rate, $\beta = 1/3$.

Table 3. Universality classes.

algorithm	β	random δ	deterministic δ
Rosenblatt	$1/3$	5	8
AH	$1/2$	4	6
SW, Step, OPT	1	3	4

We have observed that, in general, the two exponents are independent. However, if a simple condition holds, then there exists a relation connecting β and δ for each kind of drift. It can be shown that for the scenarios we have presented:

$$\begin{aligned} \delta &= \frac{1}{\beta} + 2 \,, \quad \text{(random drift)} \\ \delta &= \frac{2}{\beta} + 2 \,, \quad \text{(deterministic drift)} \,. \end{aligned} \tag{30}$$

The reader interested in the details of how to derive the above relations is referred to (Kinouchi & Caticha, 1993) and appendix B.

4. Practical Considerations

The most important question that arises in the implementation of the variational ideas as a guide to construct algorithms is how to measure the several unavailable quantities that go into the construction of the modulation function. The problem of inferring the example distribution will not be considered and only a simple method to measure the student-teacher overlap ρ will be presented. This is done by adding a 'module' to the perceptron in order to estimate online the generalization error, as studied in (Kinouchi & Caticha, 1993). Algorithms that rely on this kind of module are quite robust with respect to changes in the distribution of examples and even to lack of statistical independence (Kuva et al., in press). Consider an online estimator (a 'running average') which uses the instantaneous error $\epsilon^\mu = (1 - \sigma_B^\mu \sigma_J^\mu)/2$ to update the current estimate of the generalization error:

$$\hat{e}_G^{(\mu+1)} = (1 - \frac{\omega}{N})\hat{e}_G^{(\mu)} + \frac{\omega}{N}\epsilon^\mu \, . \tag{31}$$

This estimator incorporates exponential memory loss through the ω parameter. In the perceptron, due to the factor $\lambda = \tan(\pi e_G)$ that appears in the modulation function, fluctuations around $e_G \approx \frac{1}{2}$ may lead to spurious divergences. Therefore it is natural to consider the truncated Taylor expansion

$$\hat{\lambda}_k = \tan^{(k)}(\pi \hat{e}_G) = \pi \hat{e}_G + \frac{1}{3}(\pi \hat{e}_G)^3 + \frac{2}{15}(\pi \hat{e}_G)^5 + \cdots + c_k(\pi \hat{e}_G)^k \, . \tag{32}$$

Then, the modulation function for an adaptive algorithm inspired by the noiseless optimal algorithm is

$$W(\hat{\lambda}_k, \Delta_\mu) = \frac{1}{\sqrt{2\pi}} \frac{\hat{\lambda}_k}{H(\frac{-\Delta_\mu}{\hat{\lambda}_k})} \exp(-\frac{\Delta_\mu^2}{2\hat{\lambda}_k^2}) \, . \tag{33}$$

In Figure 7 we present the results of applying this algorithm to a problem where the drift itself is non-stationary. We have dubbed this non-stationarity *drift acceleration*. The algorithm is quite uninterested in the particular type of drift acceleration, and as an illustration we chose a drift given by $D = D_0 \sin^2(2\pi\nu t)$. The adaptive algorithm makes **no** use of this knowledge. There has been no attempt at optimizing the estimator itself, but a reasonable and robust choice is $\omega = 2$ and $k = 3$. Simulations were done for $N = 1000$, a size regime where, for all practical purposes, the central limit theorem holds. Note that the Hebb algorithm is not able to keep track of the rule since it has no internal forgetting mechanism.

We have not studied the mixed case of drift in the presence of noise. The nature of the noise process corrupting the data is essential in determining the asymptotic learning exponent (β). While multiplicative (flip) noise does not alter β for the optimized algorithms, additive (weight) noise does. This extension deserves a separate study. See (Biehl et al., 1995) for the behavior of the optimized algorithm and noise level estimation in the presence of *noise acceleration* in the absence of drift; see also (Heskes, 1994) where it is shown that learning is possible even in the mixed drift-noise case.

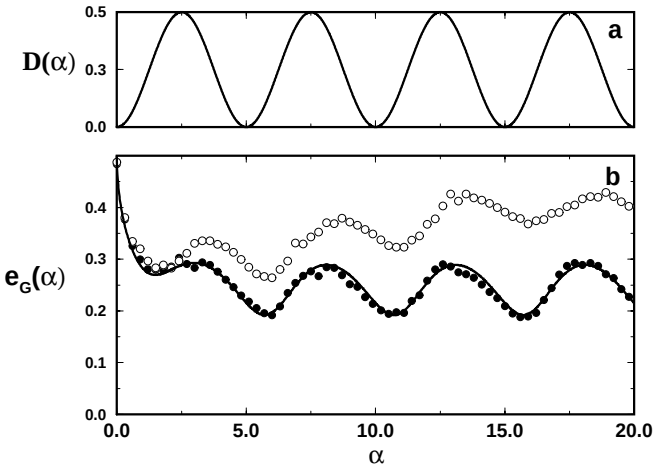

Figure 7. a) Oscillating drift level $D = D_0 \sin^2(2\pi\nu t)$ for $D_0 = 0.5, \nu = 0.1$. b) Integration of the differential equation for the oscillating case $D_0 = 0.5, \nu = 0.1$ (thick solid). Adaptive optimal algorithm with $\omega = 2$ and $k = 3$ (black circles) and simple Hebb algorithm (white circles).

5. The Wisconsin Test for Perceptrons: Piecewise Constant Rules

How do the algorithms studied in the previous sections perform in the case of abrupt changes (piecewise constant rules)? The interest is in determining how the optimal algorithms fare in a task for which they were not optimized. The Wisconsin test (WT) for perceptrons (WTP) to be studied here is used in the diagnostics of pre-frontal lobe (PFL) syndrome in human patients and will now be described very briefly (for details see e.g., (Shallice, 1988; Levine et al., 1992)).

Consider a deck of cards, each one having a set of pictures. The cards can be arranged into two categories in several different ways. The different possible classifications can be done according to, e.g., color (black or red pictures), parity (even or odd number of figures in the picture), sharpness (figures can be round or pointed) etc. The examiner chooses a rule and a patient is shown a sequence of cards and asked to classify them. The information whether the patient's classification is correct or not is made available before the next presentation. After a few trials (5-10) normal subjects are able to infer the desired rule. PFL patients are reported to infer correctly the rule after as little as 15 trials. Now a new rule is chosen at random by the examiner but the patient is not informed about the change. Normal patients are quick to pick up the change and after a small number of trials (5-10) are again able to correctly classify the cards. PFL patients are reported to persevere in the old rule and after as much as 60 trials still insist in the old classification rule.

Our WTP is designed as a direct implementation of these ideas, by considering learning from a piecewise constant teacher, without resetting the couplings to *tabula rasa*, i.e., without letting the patient know that the rule has changed.

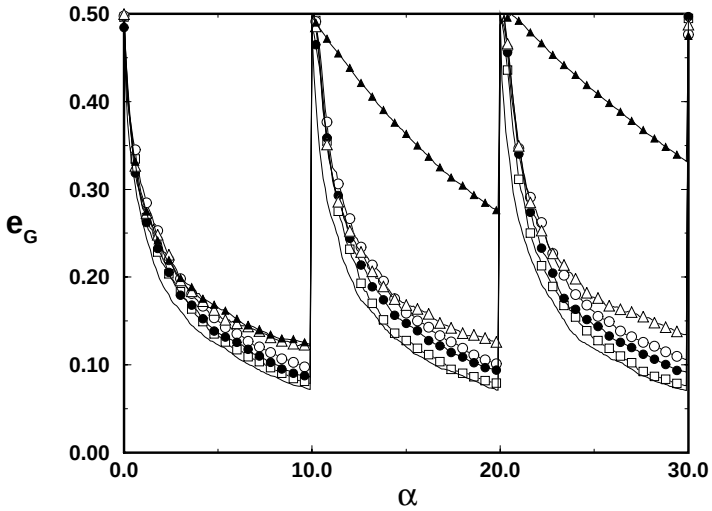

Figure 8. Simulations for $N = 500$ in single runs. Bottom line: lower bound given by the optimal algorithm with the true values of λ. Symbols: optimal algorithm (white squares), step algorithm (black circles), symmetric weight algorithm (white circles), annealed Hebb (white triangles), all with $\omega = 2$ and $k = 3$ estimator, and Hebb (black triangles). The rule is piecewise constant, it changes abruptly at $\alpha = 10$ and $\alpha = 20$.

Figure 8 shows the results of simulations with the adaptive algorithm of (33). The rule is constant up to $\alpha = 10$, it then suddenly jumps to another, uncorrelated vector and stays again unchanged until $\alpha = 20$ and so on. The most striking feature is that the perceptron with pure Hebbian algorithm works quite efficiently for the first rule but perseveres in that state and adapts poorly to a change. It can not detect performance degradation and is not surprised by the errors. The reason for that is that the scale of the weight changes is the same independently of the length of the **J** vector. The other algorithms are able to adapt to the new conditions as they incorporate the estimate of the performance of the student.

6. Conclusions

The necessary ingredients for the online tracking of drifting concepts, adaptation to non-stationary noise etc., emerge naturally and in an integrated way in the optimized algorithms. These ingredients have been *theoretically derived* rather than heuristically introduced. Many of the current ideas in machine learning of changing concepts can be viewed as playing a role similar to the ideal features discussed here for the perceptron. Among the important ideas arising from the variational approach are:

- **Learning algorithms from first principles:** For each one of these simple learning scenarios an ideal and optimal learning algorithm can be found from first principles. These optimized algorithms do not have arbitrary parameters like learning rates, acceptance

thresholds, parameters of learning schedules, forgetting factors, etc. Instead, they have a set of features which play similar roles to those heuristic procedures. The exact form of these features may suggest new mechanisms for better learning.

- **Learn to learn in changing environments:** The optimal modulation function W indeed represents a parametric family of functions, the (non-free) parameters being the same as those present in the probability distribution of the learning problem: J, ρ, χ, etc. The modulation function changes during learning, that is, the algorithm moves in this parametric space during the learning process. The student "learns to learn" by online estimation of the necessary parameters of its modulation function.

- **Robustness of optimized algorithms:** Historically, a multitude of learning algorithms has been suggested for the perceptron: Rosenblatt's perceptron, Adaline, Hebb, Adatron, thermal perceptron, OLGA, etc. From the variational perspective, these practical algorithms can be viewed as more or less reliable approximations of the ideal ones in the TL. For example, simple Hebb corresponds to the optimal algorithm in the limit $\rho \to 0$; the Adatron (relaxation) algorithm is related to the limit $\rho \to 1$; OLGA (Kim & Sompolinsky, 1996) and Thermal Perceptron (Frean, 1992) include an acceptance threshold which mimics the optimal algorithm in the presence of multiplicative noise χ. Thus, although the optimal algorithms are derived for very specific distributions of examples, it does not mean that they are fragile, non-robust when applied in other environments. They are indeed very robust (Kuva et al., in press), at least for the environments in which the standard algorithms work, since these practical algorithms are 'particular cases' of a more general modulation function. But since new learning situations (new types of noise, drifting processes, general non-stationarity, etc.) can be theoretically examined from the variational viewpoint, it is possible that new features emerge, and that these suggest new practical ideas for more robust and efficient learning.

- **Emergence of 'cognitive' modules:** Do the variational ideas have any relevance to 'biological machine learning'? Probably not for the biological *structures*, which are produced by opportunistic 'evolution bricolage', but perhaps they might apply in understanding biological cognitive *functions*. The variational approach brings forth a suggestion that, even if not new, acquires a more concrete form due to the transparent nature of the simple models studied: *optimization of the learning ability leads to the emergence of 'cognitive functional modules', here defined as components of the modulation function and accessory estimators of relevant quantities of the probability distribution related to the learning situation*. A tentative list of such estimators suggested by the variational approach may be: a) a *mismatch* (surprise) module for detection of discrepant examples; b) an emotional/attentional module for providing differential memory weight for these discrepant examples; c) 'constructivist' filters which accommodate or downplay the highly discrepant data; d) noise level estimators for tuning these filters; e) a working memory system for online estimation of current performance which enables detection of environmental changes. In conclusion, the variational approach suggests that the *necessity* of certain *cognitive functions* may be related to statistical inference principles already present in simple learning machines.

- **Extensions:** All the results presented here have been obtained under a rather severe set of restrictions from a practical point of view. The main points concern the TL; noise, order parameter and example distribution estimation; larger architecture complexity. At present we don't know how to handle finite size effects. That the parameter estimation problem is probably easier than the others is suggested by the robustness found in (Copelli et al., 1996a). The extension of the variatonal program to experimentally more relevant architectures, such as those that include hidden units and/or soft transfer functions is possible (Vicente & Caticha, 1997; Rattray & Saad, 1997). This extension is however a difficult task, since the evaluation of the appropriate modulation functions requires a rapidly increasing amount of work when the number of hidden units grows. The effects that drift may have are not known, but it could even induce faster breaking of the permutation symmetry among the hidden nodes, thus affecting the plateau structure.

Important remaining questions concern whether the variational approach can be successfully applied to other learning models (radial basis functions, mixture models, etc.). The answers will help in determining the difference between universal and particular features of the learning systems.

Acknowledgments

O. Kinouchi and R. Vicente were supported by FAPESP fellowships. N. Caticha was partially supported by CNPq and FINEP/RECOPE.

Appendix A

In this appendix we exemplify the derivation of the annealed Hebb algorithm. This algorithm is optimal when the learning situation is given by $\mathcal{H} = \{y, \vec{\eta}, |x|, \sigma_J\}$ and $\mathcal{V} = \{\tilde{\sigma}_B\}$. Considering that $C_{S\eta} = 0$, we need to perform the average:

$$W^*(\tilde{\sigma}_B) = \frac{\tilde{\sigma}_B}{\rho} \left\langle y - \rho x \right\rangle_{\{x,y\}|\tilde{\sigma}_B}, \tag{A.1}$$

which involves $\int dy\, y\, P\left(y \mid \tilde{\sigma}_B\right)$; $\int dx\, x\, P\left(x \mid \tilde{\sigma}_B\right)$. The probability distributions are easily obtained using Bayes theorem:

$$P(y \mid \tilde{\sigma}_B) = \frac{P(\tilde{\sigma}_B \mid y) P(y)}{\int dy\, P(\tilde{\sigma}_B \mid y)}. \tag{A.2}$$

By the central limit theorem we know that, in the TL, $P(y)$ and $P(x)$ are Gaussians with unit variance and it is not difficult to verify, using (1), that :

$$P(\tilde{\sigma}_B \mid y) = \frac{\chi}{2} + (1-\chi)\Theta(\tilde{\sigma}_B y);\ P(\tilde{\sigma}_B \mid x) = \frac{\chi}{2} + (1-\chi)H(-\frac{\tilde{\sigma}_B x}{\lambda}), \tag{A.3}$$

where $\lambda = \sqrt{1-\rho^2}/\rho$ and $H(x) = \int_x^\infty \frac{dt}{\sqrt{2\pi}} e^{-t^2/2}$.

It follows that

$$\langle y\rangle_{\{x,y\}|\tilde{\sigma}_B} = \sqrt{\frac{2}{\pi}}\tilde{\sigma}_B(1-\chi); \quad \langle \rho x\rangle_{\{x,y\}|\tilde{\sigma}_B} = \sqrt{\frac{2}{\pi}}\tilde{\sigma}_B(1-\chi)\rho^2. \tag{A.4}$$

Combining the above results in (A.1) finally gives:

$$W_{AH}(\tilde{\sigma}_B;\rho,\chi) = \sqrt{\frac{2}{\pi}}\lambda^2\rho(1-\chi)\,. \tag{A.5}$$

Appendix B

To derive the relations between the exponents we remember that $e_G \propto \sqrt{1-\rho^2}$ when $\rho \to 1$, so that we may write in this limit the learning equation as

$$\frac{de_G}{d\alpha} \approx CD^m e_G^{-n} - C_1(D)e_G^{n_1} - C_2(D)e_G^{n_2} - \dots\,, \tag{B.1}$$

where C is a constant, $C_k(D)$ are functions of D and n and n_k are positive numbers. Now, denote by $C_*(D)$ the first function which survives in the limit $D \to 0$, $C_*(D \to 0) = C_*$. Then, the learning equation is

$$\frac{de_g}{d\alpha} \approx -C_* e_G^{n_*}\,, \tag{B.2}$$

$$e_G(\alpha) \approx (C_*(n_*-1)\alpha)^{-1/(n_*-1)} \quad (n_* > 1)\,,$$

$$e_G(\alpha) \propto e^{-C_*\alpha} \quad (n_* = 1)\,. \tag{B.3}$$

Thus,

$$\beta = 1/(n_*-1)\,, \tag{B.4}$$

with $\beta \to \infty$ denoting exponential decay.

In the presence of small drift, we may write $C_1(D) \propto D^{m_1}$. The stationary condition $de_G/d\alpha = 0$ leads to

$$e_G^{\infty}(D) \approx D^{\frac{m-m_1}{n_1+n}}\,, \tag{B.5}$$

$$\delta = \frac{n_1+n}{m-m_1}\,. \tag{B.6}$$

This shows that, in principle, the two exponents are independent. However, if it happens that the first surviving function is $C_*(D) = C_1(D)$ (which seems to be a very common situation), then $n_* = n_1$ and $m_1 = 0$, so that

$$\delta = \frac{1}{m}\left(\frac{1}{\beta} + (1+n)\right)\,. \tag{B.7}$$

The relations given by Eq. (30) follow from the fact that $n = 1, m = 1$ for random drift and $n = 0$ and $m = 1/2$ for deterministic drift. Other drift scenarios may define other universality classes.

In the case where $C_* \neq C_1$, we only can conclude that

$$\delta > \frac{1}{m}\left(\frac{1}{\beta} + (1+n)\right) . \tag{B.8}$$

It is important to note that an exponential decay of the error ($\beta = \infty$) leads to the limiting value $\delta = 2$ both for deterministic and random drift. It is known that the error cannot decay faster than exponential in these learning problems.

References

Amari, S. (1967). Theory of adaptive pattern classifiers. *IEEE Transactions, EC-16*, 299–307.

Anlauf, J.K. & Biehl, M. (1989). The AdaTron: an adaptive perceptron algorithm. *Europhysics Letters, 10*, 687–692.

Biehl, M. & Schwarze, H. (1992). Online learning of a time-dependent rule. *Europhysics Letters, 20*, 733–738.

Biehl, M. & Schwarze, H. (1993). Learning drifting concepts with neural networks. *Journal of Physics A: Mathematical and General, 26*, 2651–2665.

Biehl, M., Riegler, P. & Stechert, M. (1995). Learning from noisy data: an exactly solvable model. *Physical Review E 52*, R4624–R4627.

Copelli, M. (1997). Noise robustness in the perceptron. *Proceedings of the ESANN'97*, Belgium.

Copelli, M. & Caticha, N. (1995). Online learning in the committee machine. *Journal of Physics A: Mathematical and General, 28*, 1615–1625.

Copelli, M., Eichhorn, R., Kinouchi, O., Biehl, M., Simonetti, R., Riegler, P. & Caticha, N. (1996a) Noise robustness in multilayer neural networks *Europhysics Letters, 37*.

Copelli, M., Kinouchi, O. & Caticha, N. (1996b). Equivalence between learning in noisy perceptrons and tree committee machines. *Physical Review E, 53*, 6341–6352.

Frean, M. (1992). A 'thermal' perceptron learning rule. *Neural Computation, 4*, 946–957.

Haussler, D., Kearns, M., Seung, H. S. & Tishby, N. (1996). Rigorous learning curve bounds from statistical mechanics. *Machine Learning, 25*, 195–236.

Heskes, T. (1994). The use of being stubborn and introspective, In J. Dean, H. Cruse & H. Ritter (Eds.) *Proceedings of the ZiF Conference on Adaptive Behavior and Learning* . University of Bielefeld, Bielefeld, Germany.

Hondou, T. (1996). Self-annealing dynamics in a multistable system. *Progress in Theoretical Physics, 95*, 817–822.

Kim, J.W. & Sompolinsky, H. (1996). Online Gibbs learning. *Physical Review Letters, 76*, 3021–3024.

Kinouchi, O. & Caticha, N. (1992a). Biased learning in boolean perceptrons. *Physica A, 185*, 411–416.

Kinouchi, O. & Caticha, N. (1992b). Optimal generalization in perceptrons. *Journal of Physics A: Mathematical and General, 25*, 6243–6250.

Kinouchi, O. & Caticha, N. (1993). Lower bounds on generalization errors for drifting rules. *Journal of Physics A: Mathematical and General, 26*, 6161–6171.

Kinouchi, O. & Caticha, N. (1995). Online versus offline learning in the linear perceptron: A comparative study. *Physical Review E, 52*, 2878–2886.

Kinzel, W. & Ruján, P. (1990). Improving a network generalization ability by selecting examples. *Europhysics Letters, 13*, 473–477.

Kuva, S., Kinouchi, O., & Caticha, N. (in press). Learning a spin glass: determining Hamiltonians from metastable states. *Physica A*.

Levine, D. S., Leven, S. J. & Prueit, P. S.(1992). Integration, disintegration, and the frontal lobes. In Levine, D.S. & Leven, S.J. (Eds.), *Motivation, Emotion and Goal Direction in Neural Networks*. Hillsdale, NJ: Erlbaum.

Mace, C.W.H. & Coolen, A.C.C. (1998). Statistical mechanical analysis of the dynamics of learning in perceptrons. *Statistics and Computing*, 8, 55-68.

Opper, M., Kinzel, W., Kleinz, J. & Nehl, R. (1990). On the ability of the optimal perceptron to generalize. *Journal of Physics A: Mathematical and General, 23*, L581–L586.

Opper, M. & Kinzel, W. (1996). Statistical mechanics of generalization. In van Hemmen, J.L., Domany, E. & Schulten, K. (Eds.), *Physics of Neural Networks*. Berlin: Springer.

Opper, M. (1996). Online versus offline learning from random examples: general results. *Physical Review Letters, 77*, 4671–4674.

Rattray, M. & Saad, D. (1997). Globally optimal online learning rules for multi-layer neural networks. *Journal of Physics A: Mathematical and General*, L771–776.
Seung, H.S., Sompolinsky, H. & Tishby, N. (1992). Statistical mechanics of learning from examples. *Physical Review A, 45*, 6056–6091.
Shallice, T. (1988). *From Neuropsychology to Mental Structures*. Cambridge: Cambridge University Press.
Simonetti, R. & Caticha, N. (1996). Online learning in parity machines. *Journal of Physics A: Mathematical and General, 29*, 4859–4867.
Valiant, L.G. (1984). A theory of the learnable. *Communications of ACM, 27*, 1134– 1142.
Van den Broeck, C. & Reimann P. (1996). Unsupervised learning by examples: online versus offline. *Physical Review Letters, 76*, 2188–2191.
Vicente, R. & Caticha, N. (1997). Functional optimization of online algorithms in multilayer neural networks. *Journal of Physics A: Mathematical and General, 30*, L599–L605.
Watkin, T.L.H., Rau, A. & Biehl, M. (1993). The statistical mechanics of learning a rule. *Reviews of Modern Physics, 65*, 499–556.

Received September 22, 1997
Accepted December 12, 1997
Final Manuscript February 26, 1998