Gradient descent learning in and out of equilibrium

Nestor Caticha and Evaldo Araújo de Oliveira
Instituto de Física, Universidade de São Paulo, CP 66318 São Paulo, SP, CEP05389-970 Brazil
(Received 16 June 2000; revised manuscript received 2 January 2001; published 24 May 2001)

Relations between the off thermal equilibrium dynamical process of on-line learning and the thermally equilibrated off-line learning are studied for potential gradient descent learning. The approach of Opper to study on-line Bayesian algorithms is used for potential based or maximum likelihood learning. We look at the on-line learning algorithm that best approximates the off-line algorithm in the sense of least Kullback-Leibler information loss. The closest on-line algorithm works by updating the weights along the gradient of an effective potential, which is different from the parent off-line potential. A few examples are analyzed and the origin of the potential annealing is discussed.

DOI: 10.1103/PhysRevE.63.061905 PACS number(s): 87.10.+e, 84.35.+i, 89.70.+c, 05.50.+q

The interest in the application of statistical mechanics to the study of learning in neural networks (NN) stems partly from the fact that the extraction of information from data (examples) can be modeled by a dynamical process of minimization of an energy function, possibly in the presence of (thermal) noise. In the case where the system is allowed to equilibrate, roughly all the possible information has been extracted from the data by the learning algorithm. In a very important sense learning theory is different from, e.g., magnetism. In the latter the interactions are fixed by the physical constraints, and the equilibrium state and how it is reached is the object of study. In the former, the energy function can be chosen in order to achieve a certain property in the equilibrium state, such as the largest possible typical generalization or memorization capability.

Techniques originated in the study of disordered systems, such as the replica and cavity methods, TAP equations, as well as Monte Carlo techniques, have been borrowed and extended, leading to several results in what has become known as off-line learning (OFL). Since disordered systems may take too long to equilibrate, implying a high computational cost, the search for efficient nonequilibrium learning algorithms has been undertaken. An interesting class of methods—where essentially, examples are used one at a time—is collected under the name of on-line learning (ONL) [1]. These bring the possibility of efficient performance and low computational cost.

Opper [2,3] has offered a new theoretical way of studying the relation between OFL and ONL. He applied his ideas to Bayes learning. The posterior probability distribution for the set of weights obtained after T examples is used as the prior for the next example. If the full posterior is maintained, any calculation amounts to an OFL one. But by projecting the posterior into a restricted family of parametric distributions, huge computational gains can be achieved, transforming the process into an effective ONL one. Now only a set of parameters and an auxiliary set of hyperparameters have to be updated. The changes in the hyperparameters induce automatically an effective annealing of tensorial learning rates. In the case of continuous weights, he applied these ideas by projecting to a Gaussian space of posteriors. Solla and Winther [4] generalized it by extending it so that information about, e.g., the binary nature of weights, can be included in a consistent way. This is simply achieved by projecting into another family of posteriors and again imposing that the information loss be minimized.

There is however no reason to limit these studies to the case of Bayes learning and the aim of this paper is to extend Opper's method to include the problem of learning by gradient descent. From a non-Bayesian point of view, Bayes learning is anything that uses a ''likelihood''—or for that matter, a Gibbs term $\exp(-\beta V)$ for some potential based learning method—in a Bayes theorem inversion formula. Without entering in such controversial arenas, we adopt, in this paper, a more restricted position on what we mean by Bayes learning. We consider learning under the conditions where we have a model in mind. This is structural prior information. We choose to implement a given architecture over others based on prior information or prejudice. No matter why, this imposes a model on the practitioner, who is still at liberty to choose the potential. But if this freedom is exercised it is at the cost of not being able to be called a Bayesian, for there is only one choice—given the *a priori* structural information—of the likelihood. There is still the possibility, that for (almost) any given potential, a set of prior informations can be identified so that the likelihood agrees with the Gibbs exponential for that potential. But prior information should remain so and not be identified *a posteriori* by reference to the potential. Nevertheless, the reader maybe left with the probably correct impression that this is only a problem of labeling the method, upon which the results will not depend. This is not the issue we address.

The point we want to stress is the relation between the thermal equilibrium OFL and the out of equilibrium ONL, independently of whether the method is Bayesian or not. The main result in this paper is determining the relation between the potentials used in OFL and ONL. We look at the zero-temperature limit and, for a class of architectures, construct the potential, which gives the closest (in some sense to be discussed) on-line approximation to an OFL problem with a given potential. These two potentials, unexpectedly, are not the same.

We obtain equations that describe the evolution of the weights and hyperparameters for general potentials. Then we look at some applications. We analyze the relation between the off equilibrium and thermal equilibrium for a special

case, which is Bayes optimal with a nondifferentiable transfer function, the noiseless Boolean perceptron. The on-line algorithm is automatically annealed and we discuss how the annealing is related to a performance estimate. Finally, we apply the resulting equations to the same architecture using Rosenblatt's perceptron algorithm. The generalization exponent changes from 1/3 in the pure ONL to 1 in the minimum information loss projection ONL.

Let y_k be an example. In the case of supervised learning it is to be thought of as an input-output pair $y_k=(S_k,\sigma_k)$ and we assume, based on the available prior structural information, that the data pairs are generated by a map $\sigma=f_{w*}(S)$, which might be deterministic or stochastic so as to include the possibility of noise corrupted data. For unsupervised learning or density estimation it is an input vector $y_k=S_k$. The learning set is formed by μ such random examples $D_\mu=(y_1,y_2,\ldots,y_\mu)$, drawn independently from identical distributions. The purpose of learning is to make an estimate $\hat{w}$ of the true N dimensional vector of parameters or weights $w*$. To do so a cost function or potential $V[\sigma,f_w(S)]=V(w,y)$ is introduced. Usually one seeks a minimum of the total energy $E(w)=\Sigma_{k=1}^{\mu}V[\sigma_k,f_w(S_k)]$, so that learning is stated as an optimization problem. The additive form is adequate in the case of independent (or noninteracting) examples. There is also the possibility that aside from the learning set, and the model, other information about the possible weight vectors is available. It might be encoded in the prior probability $p_0(w)$, that is, the probability that can be attributed to any w, of being the true parameter vector, based on information other than D_μ. The information contained in the prior and in the learning set can be taken into account simultaneously by using Bayes theorem and imposing the equivalence of the minimum-energy prescription and that of maximizing the likelihood of the examples, which as shown by Levin *et al.* [5], leads to a functional equation whose solution is the Gibbs distribution

$$P_V(w|D_\mu)=\frac{1}{Z_\mu}p_o(w)P(D_\mu|w) \tag{1}$$

$$=\frac{1}{Z_\mu}p_o(w)\exp\left[-\beta\sum_{k=1}^{\mu}V(w,y_k)\right], \tag{2}$$

where β measures the sensibility of the likelihood, and of course, plays the role of the inverse temperature, and the partition function is given by $Z_\mu=\int p_o(w')P(D_\mu|w')d^Nw'$.

Thus, the problem has been formulated as one of statistical mechanics, in this case of disordered systems due to the random nature of the data. Spin-glass behavior for this type of system has been found in many different cases. Estimation of parameters may turn into a computationally hard problem, as suggested by the long thermalization times encountered while doing, e.g., Monte Carlo estimates. This also happens for the prediction of the output σ to a new (statistically independent) input vector. A neural network, on the other hand, once it has been trained, and a reasonable $\hat{w}$ been determined, permits rapid estimation of σ. The fact that the determination of $\hat{\mathbf{w}}$, using the full Gibbs distribution, may itself be hard, seems to imply that there is no way out. However, suppose a reasonable estimate has been achieved for a learning set D_μ, then the incorporation of the information carried by a new example $y_{\mu+1}$ can be efficiently and easily done at least in an approximate way. This is the idea behind ONL and we now study this from the same perspective Opper has used to analyze Bayes learning. That these estimates are in general hard to do, leads to an approximation of the Gibbs distribution $P_V(w|D_\mu)$ by $P_g(w|D_\mu)$. The type of problem dictates what is a useful approximation. In many cases the fluctuations, at least for large μ, will be Gaussian, and so we study this case. Still the approximation can be done in many ways. To limit the loss of hard gained information, as measured by the Kullback-Leibler [6] divergence,

$$D_{KL}=\int P_V\log\left(\frac{P_V}{P_g}\right)d^Nw, \tag{3}$$

we follow [2–4] and project the current version of the Gibbs distribution to a Gaussian with the same mean $\hat{w}(\mu)$ and covariance $C_{ij}(\mu)$. To check this, look at the variations of D_{KL} with respect to P_g.

ONL proceeds by storing all the information in the previous μ examples in the vector $\hat{w}(\mu)$. Other auxiliary quantities [in this case the covariance $C_{ij}(\mu)$] usually termed hyperparameters, will be needed and their natural appearance and evolution naturally justify the annealing of learning rates.

The basic idea already in Ref. [2] is to consider the Gibbs distribution as the prior for the new, the $(\mu+1)$th example. Even when $P_V(w|D_\mu)$ is substituted by the Gaussian $P_g(w|D_\mu)$, in general $P_V(w|D_{\mu+1})$ will not be Gaussian. Therefore it is projected into a Gaussian of mean $\hat{w}(\mu+1)$ and covariance $C_{ij}(\mu+1)$ The procedure can then be iterated to include the next example. Of course this update will change the covariance of the posterior, leading to a new set of equations relating $C_{ij}(\mu+1)$ and $C_{ij}(\mu)$.

The introduction of a new example, if the system is allowed to thermalize, can be the starting point for a cavity analysis as studied by Griniasty [7]. We do not, by doing the Gaussian approximation, allow the system to thermalize.

In order to calculate the approximate change in the expected value of $\mathbf{w}$, start with [5]

$$P_V(w|D_{\mu+1})=\frac{P_V(w|D_\mu)\exp[-\beta V(w,y_{\mu+1})]}{\int P_V(w'|D_\mu)\exp[-\beta V(w',y_{\mu+1})]d^Nw'} \tag{4}$$

and substitute it by

$$\tilde{P}_V(w|D_{\mu+1})=\frac{P_g(w|D_\mu)\exp[-\beta V(w,y_{\mu+1})]}{\int P_g(w'|D_\mu)\exp[-\beta V(w',y_{\mu+1})]d^Nw'}, \tag{5}$$

then project $\tilde{P}_V(w|D_{\mu+1})$ to $P_g(w|D_{\mu+1})$. Call the initial conditions to this iteration procedure $\hat{w}(0)$ for the mean and

for covariance $\mathbf{C}(0)$. We call our current estimates of the weights and the covariance $\hat{w}(\mu)$ and $\mathbf{C}(\mu)$, respectively. Then

$$\hat{w}_i(\mu+1)=\int w_i P_g(w|D_{\mu+1})d^N w, \tag{6}$$

$$C_{ij}(\mu+1)=\int [w_i-\hat{w}_i(\mu+1)][w_j-\hat{w}_j(\mu+1)]P_g(w|D_{\mu+1})d^N w' \tag{7}$$

Let $\mathbf{u}$ measure the Gaussian fluctuations of w around $\hat{\mathbf{w}}(\mu)$

$$\hat{w}_i(\mu+1)=\frac{\int w_i P_g(w|D_\mu)\exp[-\beta V(w,y_{\mu+1})]d^N w}{\int P_g(w'|D_\mu)\exp[-\beta V(w',y_{\mu+1})]d^N w'}=\hat{w}_i(\mu)+\frac{\int u_i \exp[-\frac{1}{2}\mathbf{u}^t\mathbf{C}^{-1}\mathbf{u}]\exp\{-\beta V[\hat{w}(\mu)+\mathbf{u},y_{\mu+1}]\}d^N\mathbf{u}}{\int \exp[-\frac{1}{2}\mathbf{u}^t\mathbf{C}^{-1}\mathbf{u}]\exp\{-\beta V[\hat{w}(\mu)+\mathbf{u},y_{\mu+1}]\}d^N\mathbf{u}}.$$

Note that $u_i \exp[-\frac{1}{2}\mathbf{u}^t\mathbf{C}^{-1}\mathbf{u}]=-C_{ij}\partial_{u_j}(\exp[-\frac{1}{2}\mathbf{u}^t\mathbf{C}^{-1}\mathbf{u}])$, then one integration by parts leads to

$$\hat{w}_i(\mu+1)=\hat{w}_i(\mu)+C_{ij}\frac{\int \exp[-\frac{1}{2}\mathbf{u}^t\mathbf{C}^{-1}\mathbf{u}]\partial_{u_j}\exp\{-\beta V[\hat{w}(\mu)+\mathbf{u},y_{\mu+1}]\}d^N\mathbf{u}}{\int \exp[-\frac{1}{2}\mathbf{u}^t\mathbf{C}^{-1}\mathbf{u}]\exp\{-\beta V[\hat{w}(\mu)+\mathbf{u},y_{\mu+1}]\}d^N\mathbf{u}},$$

where a summation over repeated indices is implied. Then using

$$\partial_{u_i} f(\hat{\mathbf{w}}+u)=\partial_{\hat{w}_i} f(\hat{\mathbf{w}}+\mathbf{u}), \tag{8}$$

the on-line algorithm that results is

$$\hat{w}_i(\mu+1)=\hat{w}_i(\mu)+C_{ij}(\mu)\partial_j \ln\langle\exp\{-\beta V[\hat{w}(\mu)+u]\}\rangle, \tag{9}$$

where $\langle\cdots\rangle$ means the average with respect to the Gaussian distribution with zero mean and covariance $C_{ij}(\mu)$.

The next step is to determine the evolution of the covariance. In terms of the Gaussian distributed fluctuations u_i of zero mean and the variation $\Delta\hat{w}=\hat{w}_i(\mu+1)-\hat{w}_i(\mu)$, given by Eq. (9)

$$C_{ij}(\mu+1)=\int (u_i-\Delta\hat{w}_i)(u_j-\Delta\hat{w}_j)P_g(w|D_\mu)d^N w'. \tag{10}$$

Now use the identity

$$u_i u_j \exp[-\tfrac{1}{2}\mathbf{u}^t\mathbf{C}^{-1}\mathbf{u}]=C_{ij}\exp[-\tfrac{1}{2}\mathbf{u}^t\mathbf{C}^{-1}\mathbf{u}]+C_{ik}C_{jl}\partial_{u_k}\partial_{u_l}(\exp[-\tfrac{1}{2}\mathbf{u}^t\mathbf{C}^{-1}\mathbf{u}]), \tag{11}$$

then two integrations by parts and the use of Eq. (8) determines the prescription for the covariance update

$$C_{ij}(\mu+1)=C_{ij}(\mu)+C_{ik}(\mu)C_{jl}(\mu)\partial_k\partial_l \times\ln\langle\exp\{-\beta V[\hat{w}(\mu)+u]\}\rangle. \tag{12}$$

On one hand, this set of equations describe a first (Gaussian) approximation to the problem of OFL learning with the potential $E_\mu=\Sigma_{\mu=1}^{\mu}V(w;y_\mu)$. On the other hand it describes an ONL learning prescription for the update of the weight vector, and a set of hyperparameters that are useful in improving performance.

We now consider the widely popular class of problems where the network is a classifier into two categories $\sigma=\pm 1$ and the dimension of S is N. We study the case where the potential $V(\lambda)$ is a differentiable function of the stability $\lambda=\sigma w S/\sqrt{N}$. How is the resulting algorithm related to the usual ONL schemes? Let $t=\sigma\hat{w}S/\sqrt{N}$, denote the stability of an example previous to its presentation to the network so that $\lambda=t+\sigma\mathbf{u}S/\sqrt{N}$, the stability of example S in the network parametrized by w. Introduce $\tilde{C}_{ij}(\mu)=\beta C_{ij}(\mu)$ and $\mathsf{x}=S_i\tilde{C}_{ij}S_j/N$. An explicit form for $\langle\exp(-\beta V)\rangle$ can be obtained. Introduce a 1 in the form $1=\int d\lambda\,\delta(\lambda-\sigma w S/\sqrt{N})\propto\int d\lambda\,d\hat{\lambda}\exp i\hat{\lambda}(\lambda-\sigma w S/\sqrt{N})$. A pair of quadratic integrations show that

$$\langle\exp(-\beta V)\rangle\propto\int d\lambda\exp-\beta\left[V(\lambda)+\frac{(\lambda-t)^2}{2\mathsf{x}}\right], \tag{13}$$

thus for the estimate of the weights we have

$$\hat{w}_i(\mu+1)=\hat{w}_i(\mu)+\frac{1}{\beta\sqrt{N}}\tilde{C}_{ij}(\mu)S_j\sigma(\mu)\partial_t\ln\int d\lambda \times\exp-\beta\left[V(\lambda)+\frac{(\lambda-t)^2}{2\mathsf{x}}\right], \tag{14}$$

while for the annealing equation

$$\tilde{C}_{ij}(\mu+1)=\tilde{C}_{ij}(\mu)+\frac{1}{\beta N}\tilde{C}_{ik}(\mu)\tilde{C}_{jl}(\mu)S_kS_l\partial_t^2\ln\int d\lambda \times\exp-\beta\left[V(\lambda)+\frac{(\lambda-t)^2}{2\mathsf{x}}\right]. \tag{15}$$

To compare to previous work we look at the zero-temperature limit. The λ integral can be calculated by the saddle-point method. Let $\lambda_o(t)$ be the minimum of $V(\lambda)+(\lambda-t)^2/2\mathsf{x}$, that is the solution of

$$\left[\frac{\partial V}{\partial\lambda}+\frac{\lambda-t}{\mathsf{x}}\right]_{\lambda=\lambda_o}=0, \tag{16}$$

then $\lim_{\beta\to\infty}1/\beta\ln\langle\exp(-\beta V)\rangle=-[V(\lambda_o)+(\lambda_o-t)^2/2\mathsf{x}]$. Define what will be shown to be the effective on-line potential

$$\mathcal{E}_{\mathsf{x}}(t)\equiv V(\lambda_o)+\frac{(\lambda_o-t)^2}{2\mathsf{x}}. \tag{17}$$

Note that from Eqs. (16) and (17) it is easy to see that

$$\left.\frac{\partial V}{\partial\lambda}\right|_{\lambda=\lambda_o}=\frac{\partial\mathcal{E}_{\mathsf{x}}(t)}{\partial t}. \tag{18}$$

The algorithm equations can now be written as

$$\hat{w}_i(\mu+1)=\hat{w}_i(\mu)-\frac{1}{\sqrt{N}}\tilde{C}_{ij}(\mu)S_j\sigma(\mu)\frac{\partial\mathcal{E}_{\mathsf{x}}(t)}{\partial t}, \tag{19}$$

$$\tilde{C}_{ij}(\mu+1)=\tilde{C}_{ij}(\mu)-\frac{1}{N}\tilde{C}_{ik}(\mu)\tilde{C}_{jl}(\mu)S_kS_l\frac{\partial^2\mathcal{E}_{\mathsf{x}}(t)}{\partial t^2}. \tag{20}$$

The update of $\hat{w}$ [Eq. (19)] can be identified with an annealed (time or number of examples μ dependent $\tilde{C}_{ij}$) tensorial learning rate Hebbian-like algorithm modulated by $\partial V/\partial\lambda|_{\lambda_o}$, the gradient of the original potential calculated, not at the point t where it would be expected since it is the pretraining stability, but at the posterior stability λ_o. However, the need to calculate the gradient at a future point λ_o would render this algorithm useless. But in its stead [see Eq. (18)] the gradient $\partial\mathcal{E}_{\mathsf{x}}(t)/\partial t$ of a related potential is used. The OFL potential is transmuted to the effective ONL potential, and the gradient of the latter can be calculated at the accessible value of t.

Equation (14) reminds others that have appeared in related but different places and a few comments are in order. It is not totally unrelated to those obtained in the cavity analysis of learning by Griniasty [7]. The cavity and replica methods are not constructive, they are used to determine the OFL performance of gradient descent learning algorithms. The parameter x plays the role of the stiffness parameter in the cavity analysis and that of $x=\lim_{\beta\to\infty}\beta(1-q)$ in the replica (symmetric) calculations. With respect to the latter, Bouten *et al.* have, in their analysis of OFL gradient descent learning, stressed the interpretation of replica results in terms of cavity arguments.

But this effect of transmutation of potentials has been seen before in [8,9]. These works were done in the context of the variational-optimization method. Its purpose is to determine a potential that leads to maximum performance by functionally extremizing a performance measure such as the generalization error with respect to the potential. For some architectures it has been applied to both ONL and OFL learning in the thermodynamic limit in order to determine maximum possible generalization. It was found [9], that for the single layer perceptron, Eq. (17) gives precisely the relation between the optimal generalization ONL and OFL potentials. The same relation holds in unsupervised learning [10]. Up to now this relation [Eq. (17)] seemed little more than accidental, but now can be seen as a consequence of approximating OFL by the closest (in the sense of Kullback-Leibler divergence) ONL learning scheme.

Equation (20) describes the annealing of the tensorial learning rate. Several works (e.g., Refs. [1,11]) have stressed the need for an ONL learning rate annealing. The need comes from the fact that once an estimate is close to a minimum of the potential, the step size should be reduced in order not to overshoot. The analogous of an annealing rate in an OFL problem appears e.g., in Ref. [12], where a performance is improved by choosing a parameter of the potential (there, the threshold κ of a relaxation algorithm) from the knowledge of the size of the learning set. This appears automatically in the variational optimized potentials, both ONL and OFL [13,9]. The origin of the need for annealing was thought to be the same. However, here, as in the work of Opper, it can be seen that even if an OFL potential is not annealed, the imposition of minimal information loss will anneal the ONL learning rate.

To understand how the annealing is working, we analyze a smooth potential V that is flat for large absolute values of the stability. For negative values it saturates at a positive value, while for positive stabilities it goes to zero. In the transition region it decays monotonically. This kind of potential is quite sensible, actually the optimal one discussed in Ref. [9] for the Boolean perceptron in the presence of multiplicative noise, is of this type. The second derivative that enters the annealing equation is positive if the example is correctly classified, and negative if not. This means that the system is estimating on-line its performance. If in error, it reacts by increasing the estimate of the variance of the posterior distribution and in that manner, allowing larger corrections to be made to the current estimate $\hat{w}$. When an example is correctly classified, then the system will start making smaller weight estimate adjustments. Actually this is consistent with the idea, exposed, e.g., in Refs. [13,14], that adaptive annealing schemes should depend on the estimate of the generalization error.

From an argument similar to Opper [3], the covariance annealing is governed by

$$\lim_{\mu\to\infty}\frac{C^{-1}}{\mu}=J_V(w^*), \tag{21}$$

where the matrix $[J_V(w^*)]_{ij}=\overline{\partial_i \partial_j \mathcal{E}_x(t)}$, and the overbar indicates average over the examples distribution. This is not in general Fisher's Information matrix, but it is expected to be so for some cases. These include the additive noise case for the perceptron with the optimal potential [15], the unsupervised learning case [10], and the linear perceptron [16], where the ONL performance is asymptotically efficient. It is expected to differ in cases such as the perceptron learning from a spherical distribution of examples in the presence of multiplicative noise, since then ONL can achieve only twice the error of the Bayes algorithm. It is possible that further studies of this system of equations can shed light on this exact factor of 2.

We now apply these equations to the particularly interesting case of the perceptron algorithm of Rosenblatt applied to a perceptron in a noiseless student-teacher scenario. The OFL potential can be defined by $V_R(\lambda)=-\lambda\Theta(-\lambda)$, where $\lambda=\sigma \mathbf{w}S/N$. A possible prescription for the weights can be obtained by simulated annealing. Let, as usual, $\alpha=P/N$, where P is the number of examples. The interest resides in the fact that the generalization error decays as α^{-1} in OFL, but only as $\alpha^{-1/3}$ for pure ONL without annealing. The relevant quantity is the effective ONL energy $\mathcal{E}_x(t)$. The modulation function, $-\partial_t \mathcal{E}_x(t)$ is

$$\lim_{\beta\to\infty} -\partial_t \ln \int d\lambda \exp -\left[\beta V_R(\lambda)+\frac{(\lambda-t)^2}{2\tilde{x}}\right] = \frac{1}{\sqrt{2\pi\tilde{x}}}\frac{e^{-t^2/(2\tilde{x})}}{H\left(\frac{-t}{\sqrt{\tilde{x}}}\right)}, \tag{22}$$

where $\tilde{x}=S_i C_{ij} S_j/N$ and $H(x)=\int_{-\infty}^{x}\exp(-z^2/2)dz/\sqrt{2\pi}$. This is surprisingly close to the optimal ONL modulation function. Even the annealing, which affects $\tilde{x}$, is similar, and from Eqs. (9) and (12) the ONL generalization error decays as α^{-1} asymptotically.

To conclude, we have studied the first approximation ONL, which is (Kullback-Leibler) closest to potential learning OFL. Somewhat surprisingly the ONL potential $\mathcal{E}_x(t)$ is not the same as the OFL $V(\lambda)$. The most striking feature is that they depend on different quantities. The former on t, the stability prior to learning, and it could not be otherwise for the post presentation stability is unknown. The latter, on the stability, which will tend, in equilibrium to the OFL (equilibrium) post presentation stability. A second feature is expected, the energy consists of a pure energy term V associated to the new term plus another that reflects the presence of previously presented examples.

The equations have been applied to the perceptron learning with Rosenblatt's perceptron algorithm. It was shown that the minimum (KL-) information loss induces an annealing, which changes the generalization learning exponent from 1/3 in pure ONL to 1 in the annealed ONL, in the same class as the full OFL.

We refer to this as a first approximation since a systematic expansion can be implemented [17]. The infinite (formal) series shows that OFL equilibrium is attained by parameters and hyperparameters updates that involve only the effective ONL potential without making reference to the OFL potential. In connection to this we look at the question [19] of what it means to learn OFL with a potential that is infinite for negative stabilities. Can gradient descent only start if the current estimate is within version space? This is the case of the noiseless perceptron optimal potential mentioned above [9]. While this issue is not totally closed, a tentative answer starts by noticing that the effective ONL potential can be used even outside version space, since it is well defined for negative prior stabilities.

A related question that arises, and which might be attacked in the future by the techniques of dynamical replicas [18], is If the effective ONL potential is used iteratively in learning from a restricted learning set, what will be the asymptotic time state? Is it obviously going to the offline limit [19,20]?

We acknowledge interesting discussions with M. Copelli, O. Kinouchi, P. Riegler, R.Vicente, and also with participants of the Workshop on Statistical Mechanics, Max Planck Institute, Dresden, 1999 where an early version of this work was presented. E.A.O. was supported by financial assistance from the Fundação de Apoio à Pesquisa do Estado de São Paulo (FAPESP) and the research of N.C. was partially supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

[1] S. Amari, IEEE Trans. Electron. Comput. **16**, 299 (1967).
[2] M. Opper, Phys. Rev. Lett. **77**, 4671 (1996).
[3] M. Opper, *On-Line Learning in Neural Networks*, edited by D. Saad (Cambridge University Press, Cambridge, England, 1998).
[4] S. Solla and O. Winther, *On-Line Learning in Neural Networks*, edited by D. Saad (Cambridge University, Cambridge, 1998).
[5] E. Levin, N. Tishby, and S. Solla, Proc. IEEE **78**, 1568 (1990).
[6] S. Kullback, *Information Theory and Statistics* (Wiley, New York, 1959).
[7] M. Griniasty, Phys. Rev. E **47**, 4496 (1993).
[8] O. Kinouchi and N. Caticha, J. Phys. A **25**, 6243 (1992).
[9] O. Kinouchi and N. Caticha, Phys. Rev. E **54**, R54 (1996).
[10] C. Van den Broeck and P. Riemann, Phys. Rev. Lett. **76**, 2188 (1996).
[11] N. Barkai, H. Seung, H. Sompolinsky, Phys. Rev. Lett. **75**, 1415 (1995).
[12] R. Meir and J. Fontanari, Phys. Rev. A **45**, 8874 (1992).
[13] O. Kinouchi and N. Caticha, J. Phys. A **26**, 6161 (1993).
[14] N. Caticha and O. Kinouchi, Philos. Mag. B **77**, 5 1565 (1998).

[15] M. Biehl, P. Riegler, and M. Stechert, Phys. Rev. E **52**, R4624 (1995).

[16] O. Kinouchi and N. Caticha, Phys. Rev. E **52**, 2878 (1995).

[17] E. A. Oliveira, R. Vicente, and N. Caticha (unpublsihed).

[18] C. W. H. Mace and A. C. C. Coolen, Statistical and Computation **8**, 55 (1998).

[19] A. Engel and C. van den Broeck, *The Statistical Mechanics of Learning* (Cambridge University Press, Cambridge, England, 2000).

[20] M. Opper and D. Haussler, Phys. Rev. Lett. **66**, 2677 (1991); in *Proceedings of the 4th Annual Workshop on Computational Learning Theory (COLT91)* (Morgan Kauffman, San Mateo, 1991).