Equivalence between learning in noisy perceptrons and tree committee machines

Mauro Copelli,* Osame Kinouchi,† and Nestor Caticha‡
Instituto de Física, Universidade de São Paulo, Caixa Postal 66318, 05389-970 São Paulo, São Paulo, Brazil

(Received 19 September 1995)

We study learning from single presentation of examples (*on-line* learning) in single-layer perceptrons and tree committee machines (TCMs). Lower bounds for the perceptron generalization error as a function of the noise level ϵ in the teacher output are calculated. We find that local learning in a TCM with K hidden units is simply related to learning in a simple perceptron with a corresponding noise level $\epsilon(K)$. For a large number of examples and finite K the generalization error decays as $\alpha_{\rm CM}^{-1}$, where $\alpha_{\rm CM}$ is the number of examples per adjustable weight in the TCM. We also show that on-line learning is possible even in the $K\rightarrow\infty$ limit, but with the generalization error decaying as $\alpha_{\rm CM}^{-1/2}$. The simple Hebb rule can also be applied to the TCM, but now the error decays as $\alpha_{\rm CM}^{-1/2}$ for finite K and $\alpha_{\rm CM}^{-1/4}$ for $K\rightarrow\infty$. Exponential decay of the generalization error in both the noisy perceptron learning and in the TCM is obtained by using the learning by queries strategy.

PACS number(s): 87.10.+e, 02.50.−r, 05.90.+m, 64.60.Cn

I. INTRODUCTION

The simplest neural network architecture with generalization properties is the single-layer perceptron. This network has been studied with great detail by the statistical mechanics community and provides a test ground for new ideas on learning algorithms and strategies [1].

One of these strategies recently studied is learning from single presentation of examples (often called *on-line*, *incremental*, or *sequential* learning [2–4]) where the examples are used sequentially inducing a single change in the network (being discarded after that). Besides its biological appeal, it is a very cheap computational procedure since no memory space is needed for storing the old examples, nor is time expensive retraining required. On-line learning is also the natural scenario for changing environments where old examples may no longer be representative of the actual rule [5,6] and for selection of examples (*learning by queries*) strategies where each new example is chosen sequentially depending on the present state of the network [2,7,1].

Surprisingly, it leads to generalization performances comparable to those obtained by the conventional minimization of global cost functions defined over all examples (*off-line* learning) [7]. The on-line generalization error for randomly chosen examples decays asymptotically with the same power of $\alpha_p = P/N_p$ (the number of examples P per adjustable parameters N_p of the perceptron) as for off-line learning, the lower bound for this error being only twice the off-line Bayesian bound $e_g^{(p)}(\alpha_p\rightarrow\infty)\approx 0.442/\alpha_p$ in the perceptron [8].

Off-line learning has been extensively studied with methods of equilibrium statistical mechanics, mainly through Gardner coupling space analysis based on the replica formalism [1]. It is interesting to compare these results with those for on-line learning, especially for the case of multilayer nets where the statistical mechanics calculations are more involved.

Indeed, on-line learning has been extended recently to multilayer nets [9,10,4,11–13]. Here we are interested in the determination of the *optimal performance* achievable by on-line learning when applied to a committee machine. The optimal generalization performance in a tree committee machine trained on-line with a *nonlocal* algorithm has been studied in [10]. Here we introduce an optimal *local* algorithm and we show that this problem is nicely related to the case of a single-layer perceptron learning from examples with noisy data.

The paper is organized as follows. In Sec. II the on-line learning scenario is described and a general prescription for calculating on-line optimal performances is given. The optimal performance for a perceptron learning from corrupted examples is determined in Sec. III, as well as the performance of the simple Hebb algorithm. Section IV contains the analysis of the optimal local learning algorithm for a tree committee machine with K hidden units; the relationship between the two problems is discussed in this section. The strategy of learning by queries in both problems is considered in Sec. V. Our conclusions are summarized in Sec. VI.

II. ON-LINE LEARNING IN NEURAL NETWORKS

A. Definitions

We study *supervised learning* of an unknown mapping $M:\mathscr{I}\rightarrow\mathscr{O}$ of an N_p-dimensional input space $\mathscr{I}$ into single output space $\mathscr{O}$ by using only the information from a set of input-output examples. The *training set* $\mathscr{L}=\{(\mathbf{S}^\mu,\xi^\mu)\}$ ($\mu=1,\ldots,P$) is an ordered list of P examples, i.e., pairs of input vectors $\mathbf{S}^\mu=\{S_1^\mu,\ldots,S_{N_p}^\mu\}$ with the corresponding output signals ξ^μ. Average over the training examples will be referred to as an integration over a measure $d\nu_{\mathscr{L}}$.

For simplicity we will describe the formalism for a single perceptron which, whenever necessary, must be regarded to be a branch perceptron of a nonoverlapping or tree committee machine (TCM). Quantities labeled by a superscript (p) refer specifically to the perceptron, while the results for the

*Electronic address: copelli@if.usp.br
†Electronic address: osame@if.usp.br
‡Electronic address: nestor@if.usp.br

committee machine are denoted by (CM). The examples used by the perceptron with adaptive weights $\mathbf{J}\in\mathbb{R}^{N_p}$ (often called the *student* or *hypothesis*) are generated by another network with the same architecture but unknown weights $\mathbf{B}\in\mathbb{R}^{N_p}$ (the *teacher* or *rule*). This corresponds to a *realizable task*, which means that the rule to be inferred lies within the hypothesis space so that some proximity measure in this space can be defined. As usual the relevant measure is the average *rule-hypothesis overlap*

$$\rho=\left\langle\left\langle\frac{\mathbf{J}\cdot\mathbf{B}}{JB}\right\rangle_{\mathcal{N}}\right\rangle_{\mathcal{L}}, \tag{1}$$

where $J=|\mathbf{J}|=\sqrt{\mathbf{J}\cdot\mathbf{J}}$, $B=|\mathbf{B}|=\sqrt{\mathbf{B}\cdot\mathbf{B}}$, $\langle\ \rangle_{\mathcal{N}}$ is the average over the distribution of networks produced by the algorithm, and $\langle\ \rangle_{\mathcal{L}}$ is the average over the possible training sets. It is possible to show that this quantity is self-averaging in the thermodynamic limit $N_p\rightarrow\infty$.

Two general classes of learning behavior appear, depending on whether the transfer function of the neurons is invertible or not. In the former case we have, for example, the linear neuron $g(x)=x$ and the graded response neuron $g(x)=\tanh(x)$. In the last case we have the Boolean neuron $g(x)=\mathrm{sgn}(x)$ where the binary output conveys only partial information about the neuron local field, leading to a qualitatively different learning behavior. On-line learning has been studied both in the invertible [4,11,13] and in the noninvertible [3,7,10,14] cases. Although only machines with Boolean units will be studied in this work, we will show how the optimal performance can be obtained from a common prescription for both types of units.

The local postsynaptic fields in the teacher and student nets are defined as

$$b_\mu\equiv\mathbf{B}\cdot\mathbf{S}^\mu/B,\quad h_\mu\equiv\mathbf{J}\cdot\mathbf{S}^\mu/J, \tag{2}$$

respectively. From the noise free rule response $\sigma_B^\mu\equiv\mathrm{sgn}(b_\mu)$, the output data ξ^μ are generated using a conditional distribution $P(\xi|\sigma_B)$.

In this work we will restrict ourselves to the study of the situation where the teacher output is corrupted by noise, that is,

$$P(\xi|\sigma_B)=\left(1-\frac{\epsilon}{2}\right)\delta(\xi,\sigma_B)+\frac{\epsilon}{2}\delta(\xi,-\sigma_B), \tag{3}$$

where δ is the Kronecker delta. The quantity ϵ (twice the probability of flipping the noise free response σ_B) will be referred to as the noise level in the teacher output.

B. Performance measures

Three different relevant performance measures can be defined. The *training* or *classification error* measures the probability of a misclassification over the training set,

$$e_c^{(p)}\equiv\frac{1}{\alpha N}\sum_{\mu=1}^{\alpha N}\Theta(-\xi^\mu\sigma_J^\mu). \tag{4}$$

Contrasting with off-line learning, on-line algorithms with single presentation of examples always produce networks with nonzero classification error. This represents no problem concerning the generalization properties and perhaps constitutes an advantage in the case of noisy examples by avoiding overfitting. We observe that this quantity is measurable from simulations but up to now there are no analytical results for the on-line classification error except for the simple Hebb rule [15].

The second measure is the *generalization error*, which measures the probability of misclassification of a *noise free* example chosen with a uniform test distribution $d\nu_{\mathcal{T}}(\mathbf{S})$,

$$e_g^{(p)}\equiv\int d\nu_{\mathcal{T}}(\mathbf{S})\Theta(-\sigma_B(\mathbf{S})\sigma_J(\mathbf{S}))=\frac{1}{\pi}\arccos\rho. \tag{5}$$

Although this error is perhaps inaccessible (if we only can obtain noisy outputs ξ as test examples) it has theoretical importance since it measures how successful the hypothesis has been in approximating the rule.

A third measure, which we will call *prediction error*, is the probability of misclassification of a noisy example drawn from the same distribution used during learning,

$$e_p^{(p)}\equiv\int d\nu_{\mathcal{L}}(\mathbf{S},\xi)\Theta(-\xi\sigma_J(\mathbf{S})). \tag{6}$$

If this distribution is uniform, with $P(\xi|\sigma_B)$ given by Eq. (3), we have

$$e_p^{(p)}=(1-\epsilon)\frac{1}{\pi}\arccos\rho+\frac{\epsilon}{2}=(1-\epsilon)e_g^{(p)}+\frac{\epsilon}{2}. \tag{7}$$

Thus in the presence of noise the prediction error is nonzero even if we have a perfect match $\mathbf{J}=\mathbf{B}$. This prediction error is measurable but, since ϵ is unknown, it does not give a direct estimate of the success of the modeling task (how near $\mathbf{J}$ is to $\mathbf{B}$). We observe, however, that both $e_g^{(p)}$ and $e_p^{(p)}$ are monotonic decreasing functions of the overlap ρ, and that the maximization of this overlap is the primordial task for the learning procedure.

C. Learning dynamics and the optimization procedure

We will show that upper bounds for the average evolution of ρ as a function of the number of examples and noise level can be calculated for any distribution of examples.

Our starting point is a fairly general learning dynamics which depends on the two vectors available in the problem ($\mathbf{J}$ and $\mathbf{S}$)

$$J_i(\mu)=\left(1-\frac{\Omega(\mu)}{N_p}\right)J_i(\mu-1)+\frac{1}{N_p}F(\mu)S_i^\mu, \tag{8}$$

where $F(\mu)$ is a function which weights the change induced in the synaptic vector $\mathbf{J}$ by a new example. Due to obvious motivations, we will call it the perceptron *modulation function*. $\Omega(\mu)$ is a decay parameter which may be used to control the length of vector $\mathbf{J}$.

From Eq. (8) it is possible to obtain in the limit $N_p\rightarrow\infty$ differential equations [2,7,5] describing the evolution of ρ and J as a function of the "continuous time" $\alpha_p=\mu/N_p$. These learning equations can be used for the calculation of the performance of any algorithm:

$$\frac{d\rho}{d\alpha_p}=\rho\int d\nu_{\mathscr{L}}\frac{F(\mu)}{J}\left(\frac{b(\mu)}{\rho}-h(\mu)-\frac{F(\mu)}{2J}\right), \tag{9}$$

$$\frac{dJ}{d\alpha_p}=J\int d\nu_{\mathscr{L}}\left(\frac{F^2(\mu)}{2J^2}+\frac{F(\mu)h(\mu)}{J}-\Omega(\mu)\right). \tag{10}$$

Defining $F^{\rm opt}$ as the weight function that optimizes $d\rho/d\alpha_p$, a simple variational calculation leads to

$$F^{\rm opt}(\mu)=\frac{J}{\rho}(\langle b\rangle_{b|\xi,h}-\rho h), \tag{11}$$

where $\langle\ \rangle_{b|\xi,h}$ stands for the average over $P(b_\mu|\xi,h_\mu)$. Note that, although in this paper we will restrict ourselves to the case where the examples are random independent identically distributed variables distributed uniformly the above formal prescription for the optimal weight function is valid for *any* distribution of examples. The only hypothesis is that learning occurs obeying the initial dynamics (8).

Defining $\widetilde{W}\equiv F^{\rm opt}/J$, the evolution equations for the optimal algorithm become

$$\frac{d\rho}{d\alpha_p}=\frac{\rho}{2}\langle\widetilde{W}^2\rangle_{h,\xi}, \tag{12}$$

$$\frac{dJ}{d\alpha_p}=J\left[\frac{1}{2}\langle\widetilde{W}^2\rangle_{h,\xi}-\langle h^2\rangle_h+\frac{1}{\rho}\langle hb\rangle_{h,b}-\Omega\right]. \tag{13}$$

Again, no assumptions on the distribution of examples $P(b,h)$ were made. Equations (12) and (13) are thus also valid for *any* distribution of examples. It follows that, whenever the distribution of examples satisfies the condition

$$\rho\langle h^2\rangle_h=\langle hb\rangle_{h,b}, \tag{14}$$

and learning is done unconstrained $(\Omega=0)$, the evolutions of ρ and J are simply related,

$$\frac{1}{\rho}\frac{d\rho}{d\alpha_p}=\frac{1}{J}\frac{dJ}{d\alpha_p}, \tag{15}$$

as can be seen by inserting Eq. (14) in Eq. (13). This leads to the remarkable property

$$J(\alpha_p)=c\times\rho(\alpha_p), \tag{16}$$

where c is a constant which can be self-consistently set equal to unity. In particular, Eq. (14) holds for the uniform distribution of examples [see Eq. (A2) in the Appendix] and for the selected examples used in the ‘‘learning by queries’’ strategy.

III. PERCEPTRON LEARNING FROM NOISY DATA

A. The optimal algorithm

In the Appendix we show that the optimal modulation function term can be easily calculated as

$$F^{\rm opt}S_i^\mu=-J\lambda^2\frac{\partial}{\partial\hat{J}_i}\ln P(\xi_\mu|h_\mu), \tag{17}$$

where $\hat{J}_i\equiv J_i/|\mathbf{J}|$ and

$$\lambda\equiv\frac{\sqrt{1-\rho^2}}{\rho}. \tag{18}$$

This prescription for calculating the optimal modulation can be used both for Boolean or graded response units and other example distributions [16].

We now give an explicit expression for the optimal weight function for the particular case studied here, where the examples are independent identical uniformly distributed random variables and noise corrupts the teacher output according to (3). Using

$$P(\xi|h)=\sum_{\sigma_B}P(\xi|\sigma_B,h)P(\sigma_B|h)=\sum_{\sigma_B}P(\xi|\sigma_B)P(\sigma_B|h) \tag{19}$$

and the result

$$P(\sigma_B|h)=H\left(\frac{-\sigma_B h}{\lambda}\right), \tag{20}$$

one easily obtains (see Appendix)

$$P(\xi|h)=\frac{\epsilon}{2}+(1-\epsilon)H\left(\frac{-\xi h}{\lambda}\right), \tag{21}$$

where

$$H(x)=\int_x^\infty Dt,\ \ Dt\equiv\frac{1}{\sqrt{2\pi}}e^{-t^2/2}dt.$$

The optimal weight function, following the prescription of Eq. (17), is then

$$F^{\rm opt}(\mu)=J\lambda\frac{(1-\epsilon)}{\sqrt{2\pi}}\frac{e^{-h_\mu^2/2\lambda^2}}{[\epsilon/2+(1-\epsilon)H(-\xi_\mu h_\mu/\lambda)]}\xi_\mu. \tag{22}$$

The optimization procedure has determined not only the form of the modulation function, but most importantly the variables upon which it depends. In particular, note that the presence of ξ means that the learning algorithm amounts to a Hebbian-like term $(\xi\mathbf{S})$ modulated by a function of λ and h.

The optimal weight function (22) presents some interesting properties (see Fig. 1). Its main characteristics are the dependence on (1) the current performance, through the factor $\lambda=\sqrt{1-\rho^2}/\rho$; (2) the ‘‘surprise’’ presented by a new example, as measured by the pretraining stability $\Delta_\mu\equiv\xi_\mu\mathbf{J}(\mu-1)\cdot\mathbf{S}^\mu/J$; and (3) the noise level ϵ present in the environment.

Thus any practical algorithm which intends to approximate the theoretical optimal performance must approximate these characteristics of the optimal modulation function. In particular, the dependence on the unknown quantities λ and ϵ must be replaced by a dependence on measurable ones. This will be discussed at the end of this section, and in the meantime we will focus our attention on the interesting features presented by $F^{\rm opt}$.

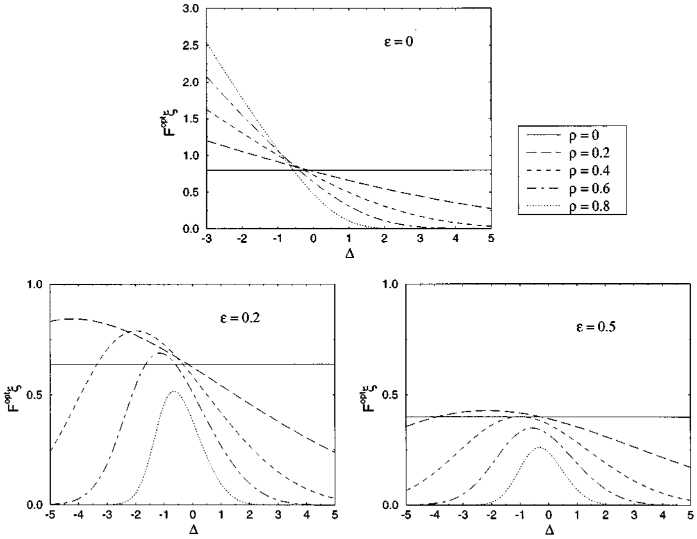

FIG. 1. Norm of the modulation function ξF^{opt} against stability Δ for several values of ρ and ϵ. Note the qualitatively different behavior for $\epsilon = 0$, which can also be observed in Fig. 2.

At $\rho \approx 0$ the weight function F^{opt} is a constant, i.e., the student perceptron starts learning in a pure Hebbian way, while at later times F^{opt} acquires a highly structured shape, as can be seen in Fig. 1. The modulation is not static but performance dependent, varying along the learning process.

The dependence on the ''surprise'' content of a new example can be seen by considering the pretraining stability, defined by $\Delta \equiv h\xi$. When $\Delta > 0$, the student output equals the teacher noisy final output, i.e., the student is correctly predicting the example. In this situation the weight function is small, indicating that no major change in the synaptic couplings should be made. But for $\Delta < 0$ the student network is mispredicting the example, and thus the weight function strongly increases with increasing $|\Delta|$, i.e., the surprise of being wrong makes the student attach importance to the example.

The presence of noise, however, changes this scenario. If Δ happens to be very negative, the weight function decreases, since, for such large $|h|$, the misprediction is probably due to noise in the teacher answer. This is a very nice analytical result which provides a theoretical justification for the heuristical procedure, developed for this same problem, of an exponentially decaying term for examples with very negative stabilities (''thermal perceptron'' [17]) and the stubborn strategy of ignoring, with some probability, highly deviant data [18]. Indeed, algorithms (e.g., relaxation) without this weight decrease for discrepant data fail to achieve the α_p^{-1} power law for $e_g^{(p)}$ [18].

The dependence on the ''surprise'' and noise level can be seen independently of the learning stage in Fig. 2, where the rescaled weight function $F^{\mathrm{opt}}\xi/J\lambda$ is plotted against the rescaled stability Δ/λ for several values of ϵ.

From Eqs. (12) and (13) the evolution of ρ and J is obtained:

$$\frac{d\rho}{d\alpha_p} = \frac{(1-\rho^2)^{3/2}}{\rho^2}\frac{(1-\epsilon)^2}{2\pi}\int \frac{Dx\, e^{-x^2/2\rho^2}}{\epsilon/2+(1-\epsilon)H(x)}, \tag{23}$$

$$\frac{dJ}{d\alpha_p} = J\frac{(1-\rho^2)^{3/2}}{\rho^3}\frac{(1-\epsilon)^2}{2\pi}\int \frac{Dx\, e^{-x^2/2\rho^2}}{\epsilon/2+(1-\epsilon)H(x)} - \Omega J. \tag{24}$$

Since the ρ equation is not coupled to the J equation, the evolution of $\rho(\alpha_p)$ is obtained by a simple numerical integration. The asymptotical behavior for $\rho \to 1$ is described by

$$\frac{d\rho}{d\alpha_p} \simeq \frac{(1-\rho^2)^{3/2}}{\rho^2}\frac{I(\epsilon)}{2\pi}, \tag{25}$$

where

$$I(\epsilon) = (1-\epsilon)^2 \int \frac{Dx\, e^{-x^2/2}}{\epsilon/2+(1-\epsilon)H(x)}. \tag{26}$$

A simple power counting on Eq. (25) shows that

$$\rho(\alpha_p) \simeq 1 - \frac{2\pi^2}{I^2(\epsilon)}\frac{1}{\alpha_p^2}, \tag{27}$$

so that

$$e_g^{(p)} \simeq \frac{2}{I(\epsilon)}\alpha_p^{-1} \tag{28}$$

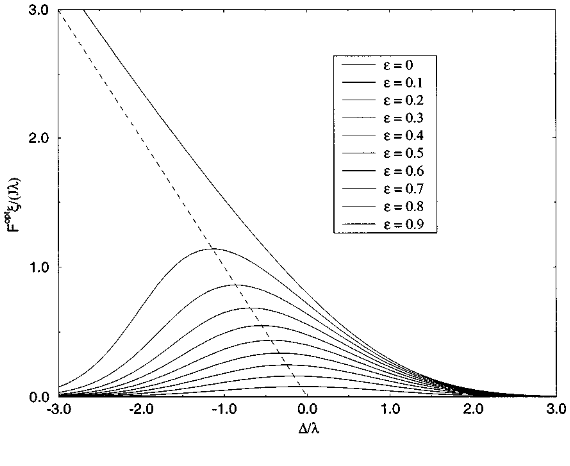

FIG. 2. Rescaled weight function $\xi F/(J\lambda)$ against rescaled stability Δ/λ. From top to bottom, ϵ=0, 0.1, . . . , 0.9. The dashed straight line corresponds to the maxima of the functions.

for large α_p. The above equation shows that for any $\epsilon \neq 1$ asymptotical perfect generalization ($e_g^{(p)} \to 0$) is possible, and the error decays always with the same power law (α_p^{-1}). For $\epsilon \simeq 1$, $I(\epsilon)$ goes to zero quadratically,

$$I(\epsilon) \overset{\epsilon\to 1}{\simeq} \sqrt{2}(1-\epsilon)^2. \tag{29}$$

The above result will be important for the analysis of the tree committee machine, in the next section.

We stress that these results represent lower bounds to the on-line performance of the perceptron in the specific scenario described. But in a real situation, one can hardly access the ''noise level'' of a system, while our optimal weight function explicitly makes use of ϵ. An on-line noise estimator through which this problem is overcome has been proposed in [14].

The optimal function also depends on the inaccessible overlap ρ. There are several ways out of this problem. A simple way for stationary rules is to take advantage of property $J(\alpha_p) = \rho(\alpha_p)$ (16) presented by unconstrained learning. The value of the perceptron norm J is a measurable quantity and the optimal function in this case reads

$$F^{\mathrm{opt}} = \sqrt{1-J^2}\,\frac{(1-\epsilon)}{\sqrt{2\pi}}\,\frac{\exp\left[-\frac{1}{2}h^2J^2/(1-J^2)\right]}{[\epsilon/2+(1-\epsilon)H(-\xi hJ/\sqrt{1-J^2})]}\,\xi. \tag{30}$$

The simulations presented in this paper have used this form for F^{opt}. In the case of time-dependent rules these solutions do not work, but for those cases a method for on-line estimation of λ has been developed in [5].

B. The Hebb algorithm

The Hebb algorithm, initially studied by Vallet [15] for the perceptron, has recently been applied in the tree committee machine with three hidden units [10]. We now study the performance of a perceptron learning with the Hebbian algorithm in the presence of output noise. This result, through the equivalence property to be presented later, enables a generalization of our results to TCMs with any K.

Instead of looking directly at the pure Hebb algorithm ($F = \xi$) we allow for a possibly varying function of the learning stage

$$F_H = W(\alpha_p)\xi. \tag{31}$$

Inserting this weight function in the evolution equation for ρ one obtains

$$\frac{d\rho}{d\alpha_p} = (1-\epsilon)\sqrt{\frac{2}{\pi}}(1-\rho^2)\frac{W(\alpha_p)}{J} - \rho\frac{W^2(\alpha_p)}{2J}. \tag{32}$$

The weight function W^* that optimizes $d\rho/d\alpha_p$ is then

$$W^* = J\sqrt{\frac{2}{\pi}}\frac{(1-\rho^2)}{\rho}(1-\epsilon), \tag{33}$$

in which case the evolution of ρ can be easily seen to be given by

$$\frac{d\rho}{d\alpha_p} = \frac{1}{\pi}(1-\epsilon)^2\frac{(1-\rho^2)^2}{\rho}. \tag{34}$$

For the initial condition $\rho(0) = 0$ we have then

$$\rho(\alpha_p) = \left[1+\frac{\pi}{2(1-\epsilon)^2\alpha_p}\right]^{-1/2}. \tag{35}$$

Note that again the optimization procedure left the ρ equation uncoupled from the J equation. It is very interesting that the obtained weight function (33) corresponds to the usual Hebbian algorithm ($F = \xi$). To see this first calculate the J evolution (10) for $W = W^*$:

$$\frac{dJ}{d\alpha_p} = J\left[\frac{1}{\pi}(1-\epsilon)^2\frac{(1-\rho^2)^2}{\rho^2} + \frac{2}{\pi}(1-\epsilon)^2(1-\rho^2) - \Omega\right]. \tag{36}$$

From Eq. (33) one obtains

$$\frac{dW^*}{d\alpha_p} = (1-\epsilon)\sqrt{\frac{2}{\pi}}\left[\frac{1-\rho^2}{\rho}\frac{dJ}{d\alpha_p} - \left(2+\frac{1-\rho^2}{\rho^2}\right)J\frac{d\rho}{d\alpha_p}\right]. \tag{37}$$

Equations (34) and (36) lead then to

$$\frac{dW^*}{d\alpha_p} = -\Omega W^*, \tag{38}$$

meaning that if $\Omega = 0$ the optimal algorithm is simply the Hebb rule ($W^* =$ const). However, if Ω is a nonzero constant, then the optimal algorithm prescribes an exponentially decaying weight $W^\star = \exp(-\Omega\alpha_p)$ to the newly presented example.

The generalization error asymptotic behavior for large α_p can be obtained from Eqs. (35) and (5):

$$e_g^{(p)} \simeq \frac{1}{\sqrt{2\pi}|1-\epsilon|}\alpha_p^{-1/2}. \tag{39}$$

Again it is worth pointing out that no matter how much noise corrupts the teacher output, the Hebbian algorithm will always give an $\alpha_p^{-1/2}$ decay for the generalization error (unless, obviously, for $\epsilon = 1$ where no learning can occur).

IV. LOCAL LEARNING FROM NOISELESS DATA IN THE TREE COMMITTEE MACHINE

In this section we study the tree committee machine. The special case in which there are only three hidden units has already been studied in [10], where it was shown that the optimal algorithm for on-line learning requires nonlocal information. That means that the optimal weight function for some branch perceptron depends on information of the other branches. We investigate now to what amount such nonlocality is important, what the optimal *local* procedure is, and how this problem is connected to learning in the perceptron from noisy data.

A. The TCM model

The K tree committee machine we deal with is a set of K independent Boolean perceptrons (branches) with $N_p = N_{\rm CM}/K$ input units each. They do not share any input component. The notation we use is such that every $N_{\rm CM}$-dimensional vector $\vec{V}$ can be thought of as K N_p-dimensional branch vectors, i.e., $\vec{V} = (\mathbf{V}_1, \ldots, \mathbf{V}_K)$.

We consider the case where a student network learns from a learning set provided by a teacher network, and we assume they have the same architecture. The learning set is a set of P pairs $\{(\vec{S}^\mu, \Sigma_B^\mu)\}$ $(\mu = 1, \ldots, P)$ where $\vec{S} = (\mathbf{S}_1, \ldots, \mathbf{S}_K)$ with $S_{kj} = \pm 1$, $k = 1, \ldots, K$, $j = 1, \ldots, N_p$, and Σ_B is the teacher output [see Eq. (41) below]. The synaptic weights of the teacher are denoted by $\vec{B} = (\mathbf{B}_1, \ldots, \mathbf{B}_K)$.

Given an input branch vector $\mathbf{S}_k$ each branch perceptron $\mathbf{B}_k$ of the teacher net gives a partial output

$$\sigma_{Bk} = \operatorname{sgn}(b_k), \quad b_k \equiv \mathbf{B}_k \cdot \mathbf{S}_k / B_k . \tag{40}$$

The normalization $\Sigma_{j=1}^{N_p} B_{kj}^2 \equiv B_k^2 = 1$ can be imposed without loss of generality.

The set $\{\sigma_{Bk}\}$ (the so called ''internal representation'' of $\vec{S}$ in the teacher net) is inaccessible for the student net. The only quantity it can access (besides the input vector $\vec{S}$) is the teacher's final output, which is made up from the internal representation:

$$\Sigma_B = \operatorname{sgn}(\mathscr{B}), \quad \mathscr{B} \equiv \sum_{k=1}^{K} \sigma_{Bk} . \tag{41}$$

The student net is defined by a vector of real connections $\vec{J}$ and upon presentation of an input vector $\vec{S}$, it gives an output

$$\Sigma_J = \operatorname{sgn}(\mathscr{H}), \quad \mathscr{H} \equiv \sum_{k=1}^{K} \sigma_{Jk} , \tag{42}$$

where

$$\sigma_{Jk} = \operatorname{sgn}(h_k), \quad h_k \equiv \mathbf{J}_k \cdot \mathbf{S}_k / J_k . \tag{43}$$

As in the perceptron case, the aim of training the student TCM is to obtain $\vec{J}$, using information from the learning set, such that $P(\Sigma_J = \Sigma_B)$, upon the presentation of a random input vector uncorrelated with the learning set, is maximized. Mato and Parga [19] have calculated the generalization error $e_g^{(\rm CM)}$ for a TCM with K hidden units as a function of the overlaps $\rho_k \equiv \mathbf{J}_k \cdot \mathbf{B}_k / J_k B_k$, where $J_k = |\mathbf{J}_k|$. In the situation where the K overlaps have the same value, i.e., $\rho_k = \rho$, $k = 1, \ldots, K$, the generalization error is given [19] by

$$e_g^{(\rm CM)}(\rho) = \frac{1}{2} - \frac{1}{2\pi^2} \sum_{n=0,2,4,\ldots}^{K-1} \binom{K}{n} \left[B\left(\frac{K-n}{2}, \frac{n+1}{2} \right) \right]^2 \times (1 - 2e_g^{(p)})^{K-n}, \tag{44}$$

where $e_g^{(p)}$ is the generalization error that each student branch perceptron would present with respect to the corresponding teacher branch perceptron, and B is Euler's beta function. In the $K \to \infty$ limit, it can be shown that

$$e_g^{(\rm CM)}(\{\rho_k\}) \overset{(\rho_k=\rho)}{=} \frac{1}{\pi} \arccos\left[\frac{2}{\pi} \arcsin(\rho) \right]. \tag{45}$$

Since we are interested in maximizing the rate at which the generalization error decreases with the presentation of examples, what Eq. (44) suggests is to maximize $d\rho_k / d\alpha_{\rm CM}$ for $k = 1, \ldots, K$, where now $\alpha_{\rm CM}$ stands for the number of presented examples per machine connection,

$$\alpha_{\rm CM} = \frac{\mu}{N_{\rm CM}} = \frac{\alpha_{\rm p}}{\rm K} . \tag{46}$$

Maximizing $de_g^{(\rm CM)}/d\alpha_{\rm CM}$ implies the maximization of $de_g^{(p)}/d\alpha_p$, turning the TCM on-line learning problem into a simple perceptron one. This property is due to the fact that the receptive fields are nonoverlapping. But on-line learning with the simple perceptron requires knowledge of the teacher perceptron output, which in this case belongs to the inaccessible internal representation of a given example. The central issue about the equivalence between on-line learning in the committee machine and in perceptrons with noisy data is that for *uniform distributions of examples* the final machine output Σ_B can be viewed as a corrupted version of the inaccessible σ_{Bk}, which is necessary for learning at each student branch perceptron $\mathbf{J}_k$. One can then use the results of Sec. III if the effective ''noise level'' ϵ is known. Clearly it depends on the size of the machine and one expects $\epsilon(K)$ to be a monotonic increasing function of K.

B. Local learning in TCMs

We now discuss the equivalence property sketched above in detail. A learning algorithm in the kth TCM branch is said to be local when, besides the teacher output Σ_B, it makes use only of a variable $\mathscr{F}_k$ which does not depend on the fields $\{h_l\}_{l \neq k}$ of the other branches. $\mathscr{F}_k$ is more generally defined by some conditional probability distribution

$$P(\mathscr{F}_k | \{h_l\}, \{b_l\}) \xrightarrow{\rm LOCAL} P(\mathscr{F}_k | h_k) \tag{47}$$

and may contain only partial information about h_k. We also define a variable $\mathscr{G}_k$ through its conditional probability distribution

$$P(\mathscr{G}_k | \{h_l\}, \{b_l\}) \xrightarrow{\rm LOCAL} P(\mathscr{G}_k | h_k). \tag{48}$$

$\mathscr{G}_k$ is a random variable whose informational value about the receptive field h_k is complementary to that of $\mathscr{F}_k$. For example, the optimal algorithm uses all the information contained in the local field, $\mathscr{F} = h$ and $\mathscr{G} = 1$; the Hebb algorithm has $\mathscr{F} = 1$ and $\mathscr{G} = h$; and the Rosenblatt perceptron algorithm has $\mathscr{F} = \sigma_J$ and $\mathscr{G} = |h|$.

For a given set $\{\mathscr{F}_k, \mathscr{G}_k\}$, the probability distribution $P(b_k, \mathscr{G}_k, \Sigma_B, \mathscr{F}_k)$ is the relevant quantity to be calculated. From it, the determination of an optimized algorithm and the calculation of the corresponding performance are straightforward procedures. Starting from the Gaussian distribution P_0 given by Eq. (A2), its calculation follows as

$$P(b_k,\mathcal{G}_k,\Sigma_B,\mathcal{F}_k)$$
$$=P_0(b_k)\int\left(\prod_{l\neq k}P_0(b_l,h_l)db_l\,dh_l\right)P_0(h_k|b_k)dh_k$$
$$\times\delta\left(\Sigma_B,\text{sgn}\left[\sum_{l=1}^{K}\text{sgn}(b_l)\right]\right)P(\mathcal{F}_k,\mathcal{G}_k|h_k). \qquad (49)$$

Integrals over $\{h_l\}_{l\neq k}$ are trivial and the resulting expression can be factorized in a simple way:

$$P(b_k,\mathcal{G}_k,\Sigma_B,\mathcal{F}_k)$$
$$=P_0(b_k)\int\left(\prod_{l\neq k}P_0(b_l)db_l\right)\delta\left(\Sigma_B,\text{sgn}\left[\sum_{l=1}^{K}\text{sgn}(b_l)\right]\right)$$
$$\times\int P_0(h_k|b_k)dh_kP(\mathcal{F}_k,\mathcal{G}_k|h_k)$$
$$=P_0(b_k)P(\Sigma_B|b_k)P(\mathcal{F}_k,\mathcal{G}_k|b_k). \qquad (50)$$

Similarly, in the perceptron learning from noisy data described in the preceding section the probability distribution $P(b,\mathcal{G},\xi,\mathcal{F})$ is the relevant quantity. Since $P(\xi|b,\mathcal{F},\mathcal{G})=P(\xi|b)$, it follows that

$$P(b,\mathcal{G},\xi,\mathcal{F})=P_0(b)P(\xi|b)P(\mathcal{F},\mathcal{G}|b). \qquad (51)$$

So, the probability distributions for both problems factorize in a very similar manner. It remains to show that $P(\Sigma_B|b_k)$ has the same structure as $P(\xi|b)$. We can write

$$P(\xi|b)=P(\xi=-\text{sgn}(b))$$
$$+[1-2P(\xi=-\text{sgn}(b))]\Theta(\xi\text{sgn}(b)), \qquad (52)$$

where $P(\xi=-\text{sgn}(b))=\epsilon/2$. From Eq. (50), we see that for uniformly distributed examples [in which case $P(\sigma_{Bk}=\pm1)=1/2$], we can write

$$P(\Sigma_B|b_k)=P(\Sigma_B=-\text{sgn}(b_k))$$
$$+[1-2P(\Sigma_B=-\text{ sgn}(b_k))]\Theta(\Sigma_B\text{ sgn}(b_k)), \qquad (53)$$

where $P(\Sigma_B=-\text{sgn}(b_k))$ is also a constant which depends on K. Therefore, for a TCM with K hidden units, there is an ‘‘effective noise level’’ $\epsilon(K)$ such that $P(\Sigma_B|b_k,K)=P(\xi|b,\epsilon(K))$. The study of a local algorithm in the TCM defined by some modulation function $F_k(\Sigma_B,\mathcal{F}_k)$ is *equivalent* to the study of the same algorithm in a perceptron with examples corrupted by some noise level $\epsilon(K)$.

For this equivalence to hold it is fundamental that the algorithm be fully *local*, i.e., the weight function for a given branch perceptron k does not make use of information from the other branch students' internal fields $\{h_l\}_{l\neq k}$. These internal fields are correlated with the teacher branch perceptron fields $\{b_l\}_{l\neq k}$ [see Eq. (A2)], and thus nonlocal rules allow us to infer (in some sense) what the internal representation of the teacher net is. This detailed information about the relation between Σ_B and $\{b_l\}$ has no equivalent in the simpler noisy perceptron situation.

C. The equivalent noise level $\epsilon(K)$

The quantity in the TCM problem which corresponds to ξ in the simple perceptron problem is Σ_B. To obtain the equivalent noise level ϵ we must then calculate $P_K(\sigma_{Bk}=\Sigma_B)\equiv P(\sigma_{Bk}=\Sigma_B|K)$, which we now proceed to do. Defining

$$\tilde{\mathcal{B}}_k\equiv\sum_{l\neq k}\sigma_{Bl}, \qquad (54)$$

which is the contribution to $\mathcal{B}$ of all the hidden units except the kth one, and recalling that the sum has $K-1$ terms (and thus is an even number), it is easy to see that

$$\begin{aligned}\sigma_{Bk}\tilde{\mathcal{B}}_k>0 &\;\Rightarrow\; \Sigma_B=\sigma_{Bk},\\ \sigma_{Bk}\tilde{\mathcal{B}}_k=0 &\;\Rightarrow\; \Sigma_B=\sigma_{Bk},\\ \sigma_{Bk}\tilde{\mathcal{B}}_k<0 &\;\Rightarrow\; \Sigma_B=-\sigma_{Bk},\end{aligned} \qquad (55)$$

so that

$$P_K(\sigma_{Bk}\neq\Sigma_B)=P_K(\sigma_{Bk}\tilde{\mathcal{B}}_k<0). \qquad (56)$$

In the uniform distribution of examples studied here, hidden units are set to ±1 with equal probabilities, i.e., $P(\sigma_{Bk})=1/2$, so that $P_K(\sigma_{Bk}\tilde{\mathcal{B}}_k<0)=P_K(\tilde{\mathcal{B}}_k<0)$. In this case, a simple counting leads to

$$P_K(\tilde{\mathcal{B}}_k)=\left(\frac{1}{2}\right)^{K-1}\frac{(K-1)!}{\left(\frac{K-1+\tilde{\mathcal{B}}_k}{2}\right)!\left(\frac{K-1-\tilde{\mathcal{B}}_k}{2}\right)!}. \qquad (57)$$

As expected, $P_K(\tilde{\mathcal{B}}_k)=P_K(-\tilde{\mathcal{B}}_k)$, so that

$$P_K(\tilde{\mathcal{B}}_k>0)=P_K(\tilde{\mathcal{B}}_k<0)=\frac{1}{2}[1-P_K(\tilde{\mathcal{B}}_k=0)]. \qquad (58)$$

Finally,

$$P_K(\sigma_{Bk}\neq\Sigma_B)=\frac{1}{2}[1-D(K)], \qquad (59)$$

where

$$D(K)\equiv P_K(\tilde{\mathcal{B}}_k=0)=\left(\frac{1}{2}\right)^{K-1}\frac{(K-1)!}{\left[\left(\frac{K-1}{2}\right)!\right]^2} \qquad (60)$$

is the probability that the kth branch decides the machine output. Since the noise level ϵ was defined in Sec. III as twice the probability that the original output σ_{Bk} is flipped, the final expression of the ‘‘equivalent noise level’’ is

$$\epsilon(K)=1-D(K)=1-\left(\frac{1}{2}\right)^{K-1}\frac{(K-1)!}{\left[\left(\frac{K-1}{2}\right)!\right]^2}. \qquad (61)$$

Thus the optimal local on-line learning for a TCM has been solved for any number of branches K. The evolution of the branch overlaps is calculated from Eq. (23) using the

TABLE I. Noise level ϵ and the corresponding asymptotic integral $I(\epsilon)$ as functions of K.

K	ϵ	$I(\epsilon)$
1	0	2.264
3	1/2=0.5	0.375
5	5/8=0.625	0.205
7	11/16=0.6875	0.141
9	93/128≃0.727	0.108
11	0.754	0.087
13	0.774	0.073
15	0.790	0.063
17	0.803	0.055
19	0.814	0.049
21	0.824	0.044

effective noise level $\epsilon(K)$. This overlap gives the branch error $e_g^{(p)}(\alpha_p)$ which must be inserted in Eq. (44) for obtaining the learning curve $e_g^{(\mathrm{CM})}(\alpha_{\mathrm{CM}})$. Observe that in this process the change $\alpha_p = K\alpha_{\mathrm{CM}}$ should be done.

D. Asymptotic behavior

For finite K, $I(\epsilon(K))$ is nonzero [see Eq. (26)], implying an $\alpha_{\mathrm{CM}}^{-1}$ decay for the generalization error per branch perceptron $e_g^{(p)}$, according to Eqs. (28) and (46). According to Eq. (44), the asymptotic generalization error of the machine behaves like

$$e_g^{(\mathrm{CM})} \simeq C(K) e_g^{(p)}, \tag{62}$$

i.e., a constant (which grows with K) times the asymptotic generalization error per branch perceptron (e.g., for $K=3$ we have $e_g^{(\mathrm{CM})} \simeq 3/2 e_g^{(p)}$, asymptotically). This happens because the sum on Eq. (44) has a finite number of terms, so that the term of order $[e_g^{(p)}]^1$ dominates. Then, the optimal local algorithm always leads to an $\alpha_{\mathrm{CM}}^{-1}$ decay for the generalization error, as long as K remains finite. In Table I the noise level ϵ and its corresponding integral $I(\epsilon)$ are shown for several values of K.

A subtle point concerns the meaning of $K\to\infty$ and the onset of the asymptotical behavior. Due to the nonuniform convergence of Eq. (44) as $K\to\infty$, at $\rho\to 1$ there is a crossover in behavior of $e_g^{(\mathrm{CM})}(\rho)$. For finite but large K two regimes can be detected. For intermediate values of α_{CM}, the error decays as $\alpha_{\mathrm{CM}}^{-1/2}$, while for sufficiently large α_{CM} it crosses over to $\alpha_{\mathrm{CM}}^{-1}$. The crossover occurs for increasingly larger values of α_{CM} as $K\to\infty$. Thus, in the limit of infinite K the $\alpha_{\mathrm{CM}}^{-1}$ regime cannot be reached (see Fig. 3).

The asymptotic behavior of $D(K)$, for large K, is

$$D(K) = 1 - \epsilon(K) \simeq \sqrt{\frac{2}{\pi}}\frac{1}{\sqrt{K}}, \tag{63}$$

so that $\epsilon(K)\to 1$ in this limit, as expected. From Eq. (29) it follows that

$$I(\epsilon(K)) \overset{K\to\infty}{\simeq} \frac{2\sqrt{2}}{\pi}\frac{1}{K}, \tag{64}$$

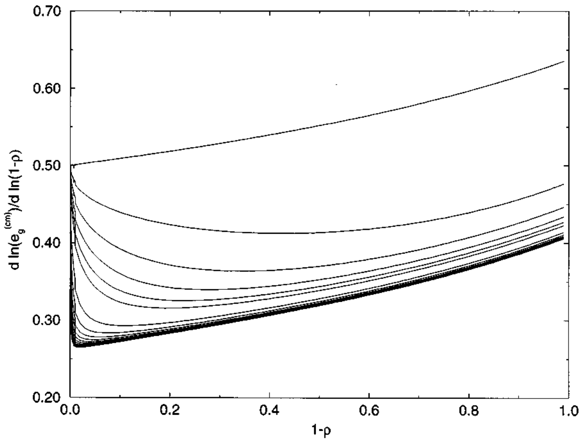

FIG. 3. Nonuniform convergence of $e_g^{(\mathrm{CM})}(\rho)$ as $K\to\infty$. For large but finite K, $d\ln(e_g^{(\mathrm{CM})})/d\ln(1-\rho)$ approaches 1/4 for small values of $1-\rho$. Yet, for ρ sufficiently close to 1, the expression converges to 1/2. From top to bottom, K=1, 3, 5, 7, 9, 11, 21, 31, 41, 51, 61, 71, 81, 91, 101, 111, 121.

so that the $e_g^{(p)}$ asymptotic decay can be obtained from Eqs. (28) and (64):

$$e_g^{(p)} \simeq \frac{\pi}{\sqrt{2}}\frac{K}{\alpha_p} = \frac{\pi}{\sqrt{2}}\alpha_{\mathrm{CM}}^{-1}. \tag{65}$$

The above equation shows that if the number of presented examples is proportional to the number of connections in each branch perceptron (that is, finite α_p), no learning can occur for infinitely many hidden units. The proper scaling in this case happens to be the number of presented examples per machine connection (α_{CM}). Since the noise level is close to unity, the modulation function prevents the occurrence of major synaptic changes in a broader region of the Δ axis. This could be interpreted as if each branch perceptron learned in the average only once every K examples. Equations (45) and (65) lead then to an asymptotic ($\alpha_{\mathrm{CM}}\to\infty$) generalization error

$$e_g^{(\mathrm{CM})} \simeq \frac{2}{\pi}\sqrt{e_g^{(p)}} \simeq \frac{2^{3/4}}{\sqrt{\pi}}\alpha_{\mathrm{CM}}^{-1/2}. \tag{66}$$

Simulations were performed to check the equivalence described, and results are shown in Fig. 4. While using the optimal weight (22), the inaccessible overlap ρ was replaced by the norm J as measured during run time, and the agreement between theoretical calculations and simulations shows that fluctuations on J are irrelevant. Note that while symbols refer to simulations on TCMs (with K from one to 11), solid lines represent the numerical integration of Eq. (23) for the corresponding $\epsilon(K)$, confirming the equivalence property.

A similar analysis can be applied to the simple Hebb learning. From Eqs. (39) and (62) it follows that for finite K the machine generalization error has an $\alpha_{\mathrm{CM}}^{-1/2}$ asymptotic decay. For large K, one finds again that the number of examples presented must be proportional to the number of input units of the committee machine for learning to occur. Using result (63) it follows that

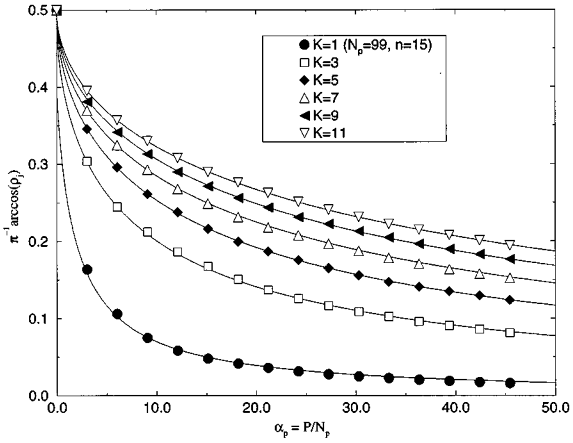

FIG. 4. Average of $e_g^{(p)}$ taken over K branches against α_p, for the optimal algorithm. N_p is the number of units per perceptron and n is the number of runs, for each K. See text for details.

$$e_g^{(p)} \simeq \frac{1}{2}\sqrt{\frac{K}{\alpha_p}} = \frac{1}{2}\alpha_{\rm CM}^{-1/2}. \tag{67}$$

In the $K \to \infty$ limit, $e_g^{(\rm CM)}$ is given by Eq. (45). Near $\rho = 1$, result (67) can be used, leading to

$$e_g^{(\rm CM)} \simeq \frac{\sqrt{2}}{\pi}\alpha_{\rm CM}^{-1/4}. \tag{68}$$

Simulations for the simple Hebb algorithm on the TCM can be seen in Fig. 5.

E. Discussion

We stress that the ‘‘optimal algorithm’’ with the specific form $F^{\rm opt}$ presented here is optimal only for the example distribution and the type of noise assumed. There is no optimal ‘‘panacea’’ algorithm, only perhaps an optimal Bayesian prescription to construct algorithms which depend on prior information (see Appendix). It is interesting, however, to compare certain traits of the optimal algorithm (in the belief that they may be generic or generalizable characteristics) with the heuristical algorithms proposed for these machines (TCMs with Boolean units), such as Mitchison and Durbin’s least action procedure [20] and its variants.

Learning a TCM. The modulation function $F^{\rm opt}$ produces major synaptic changes in the hidden units which have negative but small stabilities Δ. The least action algorithm follows a procedure which crudely resembles the optimal function: only the synapses of the unit with lowest (in modulo) negative stability are updated, so greatly discrepant internal representations in some branches do not produce synaptic changes.

The decreasing of $F^{\rm opt}$ along the learning process produced by the prefactor λ is also approximated by the conventional algorithms by using a decreasing learning step. We think that the form of the optimal local algorithm gives some theoretical insight for the success of these standard procedures.

More specifically, suppose that we make use of the least action procedure with an Adatron (relaxation) algorithm to change the individual branch perceptrons of a committee. It is known that standard Adatron does not perform well for single-layer perceptrons with this type of output noise [18] and, due to the equivalence property, we may ask whether it also presents a bad performance when used in committees. From the insight furnished by the optimal algorithm, we may expect the TCM student to succeed in learning the rule, since least action in some sense mimics the modulated cutoff, an essential ingredient for learning ‘‘noisy’’ data. The least action procedure is necessary to overcome the relaxation algorithm handicap.

However, if some algorithm similar to $F^{\rm opt}$ is used (with a ‘‘cutoff’’ for discrepant data) then the ‘‘principle of least action’’ is no longer necessary: all the branch perceptrons can be simultaneously updated. The optimal *nonlocal* algorithm is even more sophisticated: in some circumstances the lowest stability branch is not the one that gets the largest correction [10].

Learning from noisy examples. The form of $F^{\rm opt}$ suggests how to construct an algorithm robust to this type of output noise and its learning curve gives lower bounds for the performance of the on-line heuristical algorithms. Instead of a probabilistic cutoff [18], however, the optimal function prescribes a time-dependent continuous modulation of the weight given for the discrepant data. This agrees well with some proposals appearing in the ‘‘robust statistics’’ literature [21].

Since each environment (with well defined type and level of noise, distribution of examples, etc.) determines its specialized optimal modulation function, we must address the issue of algorithm robustness. Perhaps the heuristical algorithms (or even the simple Hebb rule), being less specialized, turn out to be more robust when changing the learning situation. The tradeoff specialization-robustness issue is important and is currently under study.

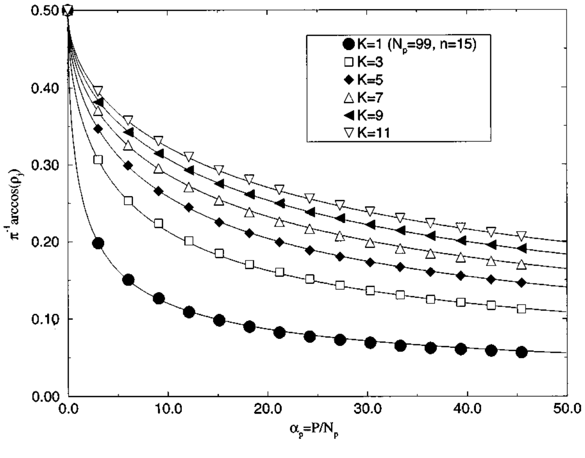

FIG. 5. Average of $e_g^{(p)}$ taken over K branches against α_p, for the Hebbian algorithm. N_p is the number of units per perceptron and n is the number of runs, for each K.

V. LEARNING BY QUERIES

Kinzel and Ruján [2] pointed out that by appropriately choosing the new examples, with a distribution $P(h) = \delta(h)$, the generalization error of a single-layer perceptron could be reduced from an asymptotic decay of

$0.399\alpha_p^{-1/2}$ to $0.199\alpha_p^{-1/2}$. In [7] it was shown that if the modulation function was permitted to evolve along the learning process, a vast qualitative improvement could be achieved, with a resulting exponential decay. We extend this result now in two directions. We first show that the exponential decay of the perceptron generalization error is preserved even in the presence of the particular type of noise studied here and secondly that this results in an exponentially decaying error for the TCM, according to the equivalence discussed in the preceding section. For large K this decay is independent of K.

In obtaining Eq. (23) we made explicit use of the distribution $P_0(h)$. For an arbitrary distribution function $P(h)$ [for which the conditional distribution function $P(b|h)$ is still $P_0(b|h)$] the evolution of ρ is given by

$$\frac{d\rho}{d\alpha_p}=\frac{(1-\rho^2)}{\rho}\frac{(1-\epsilon)^2}{4\pi}\int dhP(h)\left[\frac{e^{-h^2/\lambda^2}}{\epsilon/2+(1-\epsilon)H(-h/\lambda)}+\frac{e^{-h^2/\lambda^2}}{\epsilon/2+(1-\epsilon)H(h/\lambda)}\right]. \tag{69}$$

Defining the term between square brackets as $f(h)$, $d\rho/d\alpha_p$ is maximized if the distribution of h is chosen to be $P(h)=\delta(h-h_0)$, where h_0 is the maximum of f. The solution of $df(h_0)/dh_0=0$ is easily seen to be $h_0=0$, leading then to the optimized distribution $P_{\rm opt}(h)=\delta(h)$, a generalization of the results of Ref. [7]. Inserting $P_{\rm opt}(h)$ into Eq. (69), one obtains

$$\frac{d\rho}{d\alpha_p}=\frac{(1-\epsilon)^2}{\pi}\frac{(1-\rho^2)}{\rho}, \tag{70}$$

which for the initial condition $\rho(0)=0$ has the solution

$$\rho=\{1-\exp[-2(1-\epsilon)^2\alpha_p/\pi]\}^{1/2}. \tag{71}$$

When $\alpha_p\to\infty$, the corresponding branch errors are described by

$$e_g^{(p)}\simeq\frac{1}{\pi}\exp[-(1-\epsilon)^2\alpha_p/\pi]. \tag{72}$$

This means that for finite K the committee machine error decays exponentially with $\alpha_{\rm CM}$, with a K-dependent prefactor. For large K, results (63) and (72) can be used, leading to the branch error

$$e_g^{(p)}\simeq\frac{1}{\pi}\exp(-2\alpha_{\rm CM}/\pi^2), \tag{73}$$

which is independent of K. In the $K\to\infty$ limit we have

$$e_g^{(\rm CM)}\simeq\frac{2}{\pi^{3/2}}\exp(-\alpha_{\rm CM}/\pi^2). \tag{74}$$

To implement this strategy it is necessary to select input vectors **S** orthogonal to **J**, which may be a costly search. We observe that the exponential decay can be achieved by a less rigorous choice of **S**, say, vectors which have $h=c\lambda$. In this case we obtain from Eq. (69)

$$\frac{d\rho}{d\alpha_p}=\frac{(1-\epsilon)^2}{\pi}\frac{(1-\rho^2)}{\rho}\mathscr{T}(c), \tag{75}$$

$$\mathscr{T}(c)=\frac{e^{-c^2/2}}{4}\left[\frac{1}{\epsilon/2+(1-\epsilon)H(-c)}+\frac{1}{\epsilon/2+(1-\epsilon)H(c)}\right], \tag{76}$$

which also leads to an exponential decay for the generalization error with a prefactor $\mathscr{T}(c)<1$ in front of $\alpha_{\rm CM}$ in expressions (74) and (73).

VI. SUMMARY AND CONCLUSIONS

The variational approach to the determination of optimal algorithms has already been applied to several machines with different architectures: linear [16] and Boolean perceptrons [7], parity [22], and committee machines with Boolean units [10]. Although this approach has been mainly used in conjunction with on-line learning, it can also be applied to off-line learning [23,24].

The main idea behind this approach is that given a certain amount of information about the learning conditions, such as the distribution of examples or the type and level of noise, optimal learning algorithms can be theoretically derived instead of being proposed only from *ad hoc* and heuristical considerations. It may be argued though, that in a real learning problem this type of information can only be obtained approximately and possibly by *ad hoc* methods. We nevertheless feel that considerable theoretical insight can be achieved by studying exactly solvable models under restrictive conditions, an usual methodology in the statistical physics literature. The interpretation of the motive for the success of the least action procedure is one of these insights.

The study of increasingly richer architectures should be undertaken based on the hope that some properties will remain valid, being more a reflection of generic learning behavior rather than lucky outcomes due to the restricted set of architectures examined. One general result which has emerged is the form of the optimal cost function, $E_{\rm opt}=-\lambda^2\ln P(\xi|\{h_k\})$, which is valid for all learning situations and all machines studied so far. This is a maximum-likelihood method, but our proposal is a more general Bayesian cost function which reduces to the above form when we have a uniform prior $P(\mathbf{J})$ (see Appendix). The connection with Bayesian inference ideas deserves further clarification.

We have presented lower bounds on generalization errors. That these performances might be attained hinges on the knowledge of the noise level and example distribution. We have shown that the determination of noise level in the perceptron translates into architecture determination in the committee machine. The *a priori* knowledge of an architecture amounts to the determination of confidence levels on the data used for training and thus any restriction in the determination of one should apply to the other. Therefore methods to determine the noise level ϵ, such as the one developed by Biehl *et al.* [14], are important since they may also suggest methods for estimation of the teacher architecture complexity in other problems. How these algorithms fare in the absence of such detailed or precise information will be the subject of future work.

The equivalence between noisy perceptron and TCM leads to the possibility of studying on-line learning in the limit of infinite K. Since the limit of infinite N_p was taken before the large K limit, it must be stressed that our results should be good approximations for finite $N_{\rm CM}$ and K only when $N_{\rm CM} \gg K$. We have found a crossover from an $\alpha_{\rm CM}^{-1}$ behavior for finite K to an $\alpha_{\rm CM}^{-1/2}$ one for infinite K. This crossover is due to a nonuniform convergence of the series that defines the generalization error at ρ equal to one.

Although we have not addressed the issue of nonstationary rules they could be included in the same way as in [6,5]. The study of the learning by queries strategy is easily done. Optimization leads to an exponentially fast decaying error even when examples are not chosen at the decision border $h \neq 0$.

Although local algorithms for TCMs work well and have the nice equivalence with algorithms for the noisy perceptron problem, the study of *nonlocal* algorithms for the general K TCM does not lose its importance. On the contrary, the performance difference between local ($e_g^{(\rm CM)} \simeq 2.67/\alpha_{\rm CM}$) and nonlocal ($e_g^{(\rm CM)} \simeq 0.88/\alpha_{\rm CM}$) optimal on-line algorithms in the $K=3$ case [10] signals that nonlocality plays an important role during learning. Recent results [25] indicate that the asymptotical result $e_g^{(\rm CM)} \simeq 0.88/\alpha_{\rm CM}$ remains valid for optimal nonlocal on-line learning in general K TCMs.

Note added: After the main results of this work had been obtained we received a manuscript from Biehl *et al.* [14] which contains some overlapping results about learning in the perceptron with noise.

ACKNOWLEDGMENTS

O.K. would like to thank Rosângela Loschi for useful discussions about Bayesian inference. The work of M.C. was supported by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), while O.K. and N.C. received partial support from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

APPENDIX: TRAINING ENERGY FOR THE OPTIMAL ALGORITHM IN THE CASE OF RANDOM EXAMPLES

The dynamic evolution may be thought of as a stochastic gradient descent minimization of some cost function $H(\mu)$ defined in the space of normalized vectors $\hat{\mathbf{J}} \equiv \mathbf{J}/J$,

$$\hat{J}_i(\mu) = \hat{J}_i(\mu-1) - \frac{1}{N_p}\frac{\partial H}{\partial \hat{J}_i}, \tag{A1}$$

where $\hat{J}_i \equiv J_i/J$ is the ith component of the unit vector $\hat{\mathbf{J}}$. The total energy $H = V + E$ has two parts, the potential $V^\mu = \frac{1}{2}\Omega(\mu)\hat{\mathbf{J}}\cdot\hat{\mathbf{J}}$ controlling the synaptic vector length and the *training energy* E^μ which depends only on the example μ. We list here the most studied algorithms: Hebb: $E_H^\mu = -\Delta_\mu$; $F_H(\mu) = \xi^\mu$; Adaline: $E_A^\mu = \frac{1}{2}(\kappa - \Delta_\mu)^2$; $F_A(\mu) = (\kappa - \Delta_\mu)\xi^\mu$; perceptron: $E_P^\mu = (\kappa - \Delta_\mu)\Theta(\kappa - \Delta_\mu)$; $F_P(\mu) = \Theta(\kappa - \Delta_\mu)\xi^\mu$; and relaxation: $E_R^\mu = (\kappa - \Delta_\mu)^2\Theta(\kappa - \Delta_\mu)$; $F_R(\mu) = (\kappa - \Delta_\mu)\Theta(\kappa - \Delta_\mu)\xi^\mu$; where κ could be a function of α_p.

Consider distributions of input vectors such that the fields b and h [as defined in (2)] are Gaussian correlated variables with zero mean and unit variance (if the means are nonzero, we can work with the normalized fields $\hat{b} = b - \langle b \rangle$ and $\hat{h} = h - \langle h \rangle$):

$$\begin{aligned} P_0(b,h) &= P_0(h)P_0(b|h) \\ &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{h^2}{2}\right)\frac{1}{\sqrt{2\pi(1-\rho^2)}} \\ &\quad \times \exp\left(-\frac{(b-\rho h)^2}{2(1-\rho^2)}\right). \end{aligned} \tag{A2}$$

Starting from this probability distribution, we wish to show that the optimal weight function (11) has a corresponding training energy

$$E_{\rm opt} = -\lambda^2 \ln P(\xi|h), \quad \lambda \equiv \frac{\sqrt{1-\rho^2}}{\rho}. \tag{A3}$$

The deduction that follows is strictly valid for the uniform distribution of examples (A2) above, though we have reasons to believe that Eq. (A3) is a general prescription (see below).

According to Eqs. (8) and (A1), the weight function is related to the $\hat{\mathbf{J}}$ gradient of the energy function. Starting then from Eq. (A3), it follows that

$$\begin{aligned} \frac{\partial E_{\rm opt}}{\partial \hat{J}_i} &= -\frac{\lambda^2}{P(\xi|h)}\frac{\partial}{\partial \hat{J}_i}P(\xi|h) \\ &= -\frac{\lambda^2}{P(\xi|h)}\frac{\partial}{\partial \hat{J}_i}\int db\; P(\xi|b)P(b|h), \end{aligned} \tag{A4}$$

where we have used that $P(\xi|b,h) = P(\xi|b)$, so that the only dependence on $\hat{J}_i$ lies on the student receptive field h of $P(b|h)$. Only now must we introduce the specific example distribution (A2). For $P(b|h) = P_0(b|h)$ we have

$$\frac{\partial}{\partial \hat{J}_i}P_0(b|h) = \frac{S_i}{\lambda^2}\left(\frac{b}{\rho} - h\right)P_0(b|h), \tag{A5}$$

so that, returning to Eq. (A4),

$$\begin{aligned} \frac{\partial E_{\rm opt}}{\partial \hat{J}_i} &= -\frac{\lambda^2}{P(\xi|h)}\int dbP(\xi|b)P_0(b|h)\frac{S_i}{\lambda^2}\left(\frac{b}{\rho} - h\right) \\ &= -S_i\int dbP(b|\xi,h)\left(\frac{b}{\rho} - h\right) \\ &= -S_i\frac{F^{\rm opt}}{J}. \end{aligned} \tag{A6}$$

Since we have not specified what ξ is, this holds both for Boolean as well as graded response perceptrons. For example, ξ could be the field b itself, or a noisy version of it, so that we would be dealing with the linear perceptron problem [16].

Result (A6) enables us then to give an alternative prescription for obtaining the optimal weight function $F^{\rm opt}$. It is easier to calculate the derivative

$$F^{\rm opt} = -S_i J \frac{\partial E_{\rm opt}}{\partial \hat{J}_i} = S_i \lambda^2 J \frac{\partial}{\partial \hat{J}_i} \ln P(\xi|h) \tag{A7}$$

than to perform the integration over b prescribed by Eq. (11).

It is worth pointing out that prescriptions (11) and (A7) hold independently of our particular choice of noise (3). For example, noise could be added to the teacher field b instead, and both prescriptions would still be valid.

Now we show that the Bayesian prescription of maximizing the probability of occurrence of the hypothesis given the data leads to $E_{\rm opt}$. Note that

$$P(\xi|\hat{\mathbf{J}},\mathbf{S}) = \int dh P(\xi|h) P(h|\hat{\mathbf{J}},\mathbf{S}) = \int dh P(\xi|h)\, \delta(h - \hat{\mathbf{J}}\cdot\mathbf{S})$$

$$= P(\xi|\hat{\mathbf{J}}\cdot\mathbf{S}), \tag{A8}$$

so that

$$\ln P(\hat{\mathbf{J}}|\xi,\mathbf{S}) = \ln P(\xi|\hat{\mathbf{J}},\mathbf{S}) + \ln P(\hat{\mathbf{J}}) - \ln P(\xi)$$

$$= \ln P(\xi|h) + \text{const}, \tag{A9}$$

once the *a priori* distribution for $\hat{\mathbf{J}}$ is uniform. This may not be the case in more involved situations with a nonuniform hypothesis space.

It is important to note that the relevant quantity is $\hat{\mathbf{J}}$, not h. The Bayesian prescription is to maximize the function $\ln P(\hat{\mathbf{J}}|\xi,\mathbf{S})$, which leads to the maximization of $\ln P(\xi|h)$, *not* $\ln P(h|\xi)$. The latter would lead to an extra term, $\ln P_0(h) = -h^2/2$, which is *not* a constant. The difference between $P(\hat{\mathbf{J}}|\xi,\mathbf{S})$ and $P(h|\xi)$ lies in the fact that the sum $h = \Sigma_i y_i$ of a large number of variables ($y_i = J_i S_i$) with *uniform* priors has a *nonuniform* (Gaussian) prior (central limit theorem).

This relationship with Bayesian inference leads us to conjecture that the prescription given by Eq. (A7) continues to be valid for other network architectures and learning situations where $P(\mathbf{J})$ is uniform. Of course, if $P(\mathbf{B})$ is known to be nonuniform, this may lead us to use an adequate prior $P(\mathbf{J})$ so that improved performances may be attainable.

[1] T. L. H. Watkin, A. Rau, and M. Biehl, Rev. Mod. Phys. **65**, 2 (1993).

[2] W. Kinzel and P. Ruján, Europhys. Lett. **13**, 473 (1990).

[3] O. Kinouchi and N. Caticha, Physica A **185**, 411 (1992).

[4] M. Biehl and H. Schwarze, J. Phys. A **28**, 643 (1995).

[5] O. Kinouchi and N. Caticha, J. Phys. A **26**, 6161 (1993).

[6] M. Biehl and H. Schwarze, J. Phys. A **26**, 2651 (1993).

[7] O. Kinouchi and N. Caticha, J. Phys. A **25**, 6243 (1992).

[8] M. Opper and D. Haussler, Phys. Rev. Lett. **66**, 2677 (1991).

[9] Y. Kabashima, J. Phys. A **27**, 1917 (1994).

[10] M. Copelli and N. Caticha, J. Phys. A **28**, 1615 (1995).

[11] D. Saad and S. Solla, Phys. Rev. Lett. **74**, 4337 (1995).

[12] D. Saad and S. Solla, Phys. Rev. E **52**, 4225 (1995).

[13] P. Riegler and M. Biehl, J. Phys. A **28**, L507 (1995).

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

[15] F. Vallet, Europhys. Lett. **8**, 747 (1989).

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

[17] M. A. Frean, Neural Comput. **4**, 946 (1992).

[18] T. Heskes, in *Proceedings of the ZiF Conference on Adaptive Behavior and Learning*, edited by J. Dean, H. Cruse, and H. Ritter (University of Bielefeld, Bielefeld, Germany, 1994).

[19] G. Mato and N. Parga, J. Phys. A **25**, 5047 (1992).

[20] G. J. Mitchison and R. M. Durbin, Biol. Cybern. **60**, 345 (1989).

[21] D. S. Chen and R. C. Jain, IEEE Trans. Neural Networks **5**, 467 (1994).

[22] R. Simonetti and N. Caticha (unpublished).

[23] O. Kinouchi and N. Caticha, Phys. Rev. E (to be published).

[24] C. Van den Broeck and P. Reimann, Phys. Rev. Lett. **76**, 2188 (1996).

[25] M. Copelli and N. Caticha (unpublished).