

Finite size scaling of the Bayesian perceptron

Arnaud Buhot, Juan-Manuel Torres Moreno, and Mirta B. Gordon*

*Département de Recherche Fondamentale sur la Matière Condensée, CEA/Grenoble, 17 rue des Martyrs,
38054 Grenoble Cedex 9, France*

(Received 12 February 1997)

We study numerically the properties of the Bayesian perceptron through a gradient descent on the optimal cost function. The theoretical distribution of stabilities is deduced. It predicts that the optimal generalizer lies close to the boundary of the space of (error-free) solutions. The numerical simulations are in good agreement with the theoretical distribution. The extrapolation of the generalization error to infinite input space size agrees with the theoretical results. Finite size corrections are negative and exhibit two different scaling regimes, depending on the training set size. The variance of the generalization error vanishes for $N \rightarrow \infty$ confirming the property of self-averaging. [S1063-651X(97)15406-X]

PACS number(s): 87.10.+e, 02.50.-r, 05.20.-y

I. INTRODUCTION

A neural network is able to infer an unknown rule from examples. We address specifically classification tasks in which a single neuron, a *perceptron*, connected to N input units through weights $\mathbf{w}=(w_1, \dots, w_N)$ attributes labels ± 1 given by $\sigma = \text{sgn}(\mathbf{w} \cdot \boldsymbol{\xi})$ to input patterns $\boldsymbol{\xi}=(\xi_1, \dots, \xi_N)$. A perceptron is able to correctly classify *linearly separable* (LS) problems; the hyperplane orthogonal to \mathbf{w} separates, in the input space, the patterns given positive outputs from those given negative outputs. Given a set L_α of $P = \alpha N$ examples, i.e., training patterns $\boldsymbol{\xi}^\mu$ ($\mu = 1, \dots, P$) with their corresponding class τ^μ , the process of finding the weights \mathbf{w} is called *learning*. Generally, if the problem is LS, there is a finite volume of error-free solutions in weights space. This volume is called *version space*.

Most of the learning algorithms proposed so far may be stated as the minimization of a cost function or empirical risk $E(\mathbf{w}; L_\alpha)$ in the weights space. The structure of the problem of learning from examples allows for a statistical mechanics analysis, in which the cost function is considered as an energy. The performance of the learning algorithm is calculated through thermal averages with Boltzman distribution in weights space and quenched averages over all the possible training sets. In the thermodynamic limit $N \rightarrow \infty$, $P \rightarrow \infty$ with $\alpha = P/N$ constant, the zero temperature limit of these averages accounts for the *typical* behavior of the algorithm. The fraction of training errors ϵ_t , the generalization error ϵ_g , and the distribution of distances of the training patterns to the separating hyperplane $\rho(\gamma)$ can be determined with the assumption of self-averaging.

The minimization of the number of training errors, called the Gibbs algorithm, is not the best learning strategy in the case of LS problems, because it picks up one point in version space at random. Its typical generalization error [see (6) below for the definition] vanishes with the size of the training set like $\epsilon_g \approx 0.625/\alpha$ [1]. A more elaborate strategy is to look for those weights in version space that maximize the distance of the separating hyperplane to its closest training patterns [2,3]. These patterns are called the support vectors [4], and

define the *maximal stability* (or maximum margin) perceptron (MSP), that lies at the center of the version space, and whose generalization error vanishes in the large α regime like $\epsilon_g \approx 0.5005/\alpha$ [5,6]. As the training set only contains a small fraction of the information needed to find the underlying rule generating the examples, there is a lower bound to ϵ_g , given by Bayes decision theory [7]. Bayesian performance may be implemented by what is called a committee machine [8]: through the vote of a large number of perceptrons trained with Gibbs algorithm. The Bayesian generalization error vanishes like $\epsilon_g \approx 0.442/\alpha$, in the limit of large α . However, the convergence of the committee machine to the optimum is guaranteed only in the limit of an infinite number of perceptrons. In order to circumvent the complexity of the committee machine, several learning algorithms for single perceptrons, based on the minimization of *ad hoc* cost functions, have been recently proposed [5,6,9]. In these approaches, the cost function is sought within a given class of functions and has a free parameter which has to be optimized for each value of α , the fraction of training patterns. The generalization performance of these algorithms is very *close* to the Bayesian optimal value. Some of them end up with a finite fraction of training errors, suggesting that the optimal solution might lie outside the version space, but it has been established that this is not the case [10]. More recently, the cost function that minimizes the generalization error, was determined through a variational approach, and it was showed that its minimum endows the perceptron with the optimal, Bayesian, generalization performance [11].

In this paper, after a somewhat different derivation of the optimal cost function, we determine the typical distribution of distances of the training patterns to the Bayesian hyperplane, and we present simulation results that confirm the theoretical predictions. We find that the optimal Bayesian student lies close to the *boundary* of the version space. The finite size corrections to the generalization error are negative and present two different scaling behaviors as a function of α .

II. THEORETICAL RESULTS

In this section, we present an alternative derivation of the optimal potential for learning linearly separable tasks for

*Also at Centre National de la Recherche Scientifique.

completeness and we deduce the training patterns distance distribution. The theoretical problem is formulated as follows: the probability that the classifier assigns class σ to pattern ξ after training with a set L_α of αN examples is $P(\sigma|\{\xi, \mathbf{w}, L_\alpha\}) = \Theta(\sigma \mathbf{w} \cdot \xi) P(\mathbf{w}|L_\alpha) P(\xi)$, where $\Theta(x)$ is the Heaviside function. In general the *posterior* probability $P(\mathbf{w}|L_\alpha)$ is determined through the minimization of a cost function $E(\mathbf{w}; L_\alpha)$ which depends on the training set. In order to derive the properties of a training algorithm minimizing a cost function, it is useful to introduce a fictitious temperature $1/\beta$, and consider the finite temperature probability

$$P(\mathbf{w}|L_\alpha; \beta) = p(\mathbf{w}) \frac{e^{-\beta E(\mathbf{w}; L_\alpha)}}{Z(L_\alpha; \beta)}, \quad (1)$$

where $p(\mathbf{w})$, called the *prior* probability density, allows to impose constraints to the weights, and $Z(L_\alpha; \beta)$ is the partition function

$$Z(L_\alpha; \beta) = \int \exp[-\beta E(\mathbf{w}; L_\alpha)] p(\mathbf{w}) d\mathbf{w}. \quad (2)$$

The *typical* behavior of any intensive quantity $X(\mathbf{w})$ is obtained under the assumption of self-averaging through the quenched average over *all* the possible training sets L_α of the same size α , in the thermodynamic limit $N \rightarrow \infty$ (taken at constant $\alpha = P/N$) and in the zero temperature limit:

$$\langle\langle X \rangle\rangle = \lim_{N \rightarrow \infty} \int P(L_\alpha) dL_\alpha \left[\lim_{\beta \rightarrow \infty} \int X(\mathbf{w}) P(\mathbf{w}|L_\alpha; \beta) d\mathbf{w} \right]. \quad (3)$$

where $\langle\langle \dots \rangle\rangle$ stands for the double average, over the weights \mathbf{w} and the training sets L_α .

If the cost function has a unique minimum $\mathbf{w}^*(L_\alpha)$ (this may not be the case, as happens when the cost function is the number of training errors), then $P(\mathbf{w}|L_\alpha) = \delta(\mathbf{w} - \mathbf{w}^*(L_\alpha))$. In this case, the average between brackets in (3) is reduced to $X(L_\alpha, N)$, which is a random variable that depends on the particular training set realization through $\mathbf{w}^*(L_\alpha)$. The width of its probability distribution function is expected to vanish in the thermodynamic limit, i.e., all the training sets endow the perceptron with the same properties, with probability one. This property is called self-averaging. As a consequence, $X(L_\alpha, N)$ may be calculated by averaging over all the possible training sets to get rid of the particular training set realization. The replica method of statistical mechanics has been developed to cope with the averages over so-called quenched variables which in this case correspond to the realizations L_α .

Consider the paradigm of learning a LS rule from examples: for each pattern ξ^μ , a *teacher* perceptron of weight vector \mathbf{v} defines the corresponding target $\tau^\mu = \text{sgn}(\mathbf{v} \cdot \xi^\mu)$. As usual, we assume that the $P = \alpha N$ training patterns are independently selected with a probability density function $P(\xi^\mu)$, and that the cost function the *student's* weights \mathbf{w} have to minimize is an additive function of the examples,

$$E(\mathbf{w}; L_\alpha) = \sum_{\mu=1}^P V(\gamma^\mu), \quad (4)$$

where the potential V depends on the training pattern μ and its class through the stability

$$\gamma^\mu = \tau^\mu \frac{\mathbf{w} \cdot \xi^\mu}{\sqrt{\mathbf{w} \cdot \mathbf{w}}}. \quad (5)$$

As the outputs σ^μ and τ^μ are invariant under the transformations $\mathbf{w}, \mathbf{v} \rightarrow a\mathbf{w}, a'\mathbf{v}$ with $a, a' > 0$, the teacher's and student's weights spaces may be restricted to the hyperspheres $\mathbf{w}^2 = N$ and $\mathbf{v}^2 = N$ respectively without any loss of generality. Most training algorithms can be cast in the form (4). If the minimum of (4) is unique and $V(\gamma)$ is differentiable, the weights \mathbf{w} can be obtained by a gradient descent. This is not the case for Gibbs algorithm, whose potential is the nondifferentiable error-counting function $V^G(\gamma) = \Theta(-\gamma)$.

The generalization error $\epsilon_g(\mathbf{w})$ is the probability that a pattern, chosen at random with the same probability density as the training patterns, be misclassified by the student perceptron. Its typical value depends on the overlap $R = \langle\langle \mathbf{v} \cdot \mathbf{w} / N \rangle\rangle$ between the student and the teacher weight vectors \mathbf{w} and \mathbf{v} ,

$$\epsilon_g \equiv \langle\langle \epsilon_g(\mathbf{w}) \rangle\rangle = \frac{1}{\pi} \arccos R. \quad (6)$$

We assume that the training patterns are identically distributed random variables whose components have zero mean $\langle \xi_i^\mu \rangle = 0$ and unit variance $\langle \xi_i^\mu \xi_j^\nu \rangle = \delta_{\mu\nu} \delta_{ij}$. The free energy per neuron is averaged over the training sets with the replica method under the assumption of *replica symmetry*, which will be shown to be stable. The extremum conditions on the free energy that determine the overlap R are [5]

$$1 - R^2 = 2\alpha \int_{-\infty}^{\infty} H\left(\frac{-Rt}{\sqrt{1-R^2}}\right) (\lambda(t; c) - t)^2 Dt, \quad (7a)$$

$$R = 2\alpha \int_{-\infty}^{\infty} \exp\left(-\frac{t^2}{2(1-R^2)}\right) \frac{(\lambda(t; c) - t) dt}{2\pi\sqrt{1-R^2}}; \quad (7b)$$

with $Du = \exp(-u^2/2) du / \sqrt{2\pi}$ and $H(t) = \int_t^\infty Du = (1/2) \text{erfc}(t/\sqrt{2})$. The parameter c is the $\beta \rightarrow \infty$ limit of $\beta(1-q)$, where q is the overlap between two solutions in the student's space. If the cost function (4) has a single global minimum, $q \rightarrow 1$ and c is finite. The function $\lambda(t; c)$, determined by the saddle point equation of the free energy for $\beta \rightarrow \infty$, minimizes $W(\lambda) = V(\lambda) + (\lambda - t)^2/2c$ with respect to λ . For cost functions having continuous derivatives $\lambda(t; c)$ satisfies

$$t = \lambda + c \frac{dV}{d\lambda}(\lambda). \quad (8)$$

The solution to (7) has to verify the necessary condition for local stability of the replica symmetric solution [6]:

$$2\alpha \int_{-\infty}^{+\infty} Dt H\left(\frac{-Rt}{\sqrt{1-R^2}}\right) (\lambda'(t; c) - 1)^2 < 1, \quad (9)$$

where $\lambda' = \partial\lambda/\partial t$. It has recently been shown that (9) can only be satisfied if (8) is invertible, which imposes [12]:

$$cV'' \equiv c \frac{d^2V}{d\lambda^2} > -1. \quad (10)$$

If $V(\lambda)$ is known, ϵ_g can be calculated through the solution of Eqs. (7a) and (7b). Instead of solving this direct problem, we are interested in finding the *best* potential within the class of functions having continuous derivatives. Instead of using Schwartz inequality as in [11], we show that a straightforward functional maximization of R leads to the same result. As only the product βV appears in the partition function (2), we can multiply the potential V and the temperature $1/\beta$ by the same constant $a > 0$ leaving $Z(L_\alpha; \beta)$ invariant. This transformation changes $c \rightarrow c/a$ in (8) and (10), leaving R unchanged. Thus, we may impose $c = 1$ throughout without any loss of generality, which amounts to choosing the energy units.

A further simplification arises from considering R as a functional of V through $g(t) \equiv \lambda(t) - t$. For then we can write

$$g(t) = -\frac{dV}{d\lambda}(\lambda(t)), \quad (11)$$

where $\lambda(t)$ is the solution to (8). Equations (7) and (9) become respectively

$$1 - R^2 = 2\alpha \int_{-\infty}^{\infty} g^2(t) H\left(\frac{-Rt}{\sqrt{1-R^2}}\right) Dt \equiv \alpha f(R, g), \quad (12a)$$

$$R = \frac{\alpha}{\pi} \int_{-\infty}^{\infty} g(t) \exp\left(-\frac{t^2}{2(1-R^2)}\right) \frac{dt}{\sqrt{1-R^2}} \equiv \alpha h(R, g), \quad (12b)$$

$$2\alpha \int_{-\infty}^{\infty} (g'(t))^2 H\left(-\frac{Rt}{\sqrt{1-R^2}}\right) Dt < 1. \quad (13)$$

Given α , Eqs. (12a), (12b), and (13) must be simultaneously verified by the function $g(t)$ that maximizes R . We look for solution $g(t)$ that maximizes (12b), with (12a) considered as a constraint, introduced through a Lagrange multiplier η . As it is not easy to impose inequality (13) as supplementary constraint, we maximize $R = \alpha h(R, g) + \eta[1 - R^2 - \alpha f(R, g)]$ and we will show that our result is consistent, i.e., that the $g(t)$ obtained does indeed verify conditions (10) and (13). The function $g(t)$ that maximizes R satisfies $\delta R / \delta g(t) = 0$, which implies $\delta h / \delta g = \eta \delta f / \delta g$, where $\delta(\dots) / \delta g$ stands for the functional derivative of (\dots) with respect to $g(t)$. It is straightforward to deduce the expression for g :

$$g(t) = \frac{\eta^{-1} \exp\left(-\frac{R^2 t^2}{2(1-R^2)}\right)}{2\sqrt{2\pi(1-R^2)} H\left(-\frac{Rt}{\sqrt{1-R^2}}\right)}, \quad (14)$$

where η and R depend implicitly on α . After introduction of (14) into (12), we find the solutions $R(\alpha) \equiv \mathcal{R}$ and $\eta(\alpha)$:

$$\frac{\mathcal{R}^2}{\sqrt{1-\mathcal{R}^2}} = \frac{\alpha}{\pi} \int_{-\infty}^{\infty} Dt \frac{\exp\left(-\frac{t^2 \mathcal{R}^2}{2}\right)}{H(-\mathcal{R}t)}, \quad (15a)$$

$$\eta^{-1}(\alpha) = 2 \frac{1-\mathcal{R}^2}{\mathcal{R}}. \quad (15b)$$

They determine, through (14), the function g for each value of α :

$$g(t; \alpha) = T^2 \frac{d}{dt} \ln H\left(-\frac{t}{T}\right), \quad (16)$$

where we wrote $g(t; \alpha)$ to stress the α dependence, and $T^2 \equiv (1-\mathcal{R}^2)/\mathcal{R}^2$. It is straightforward to verify that $g(t; \alpha)$ satisfies the stability condition (13) for all α , justifying our assumption of replica symmetry. A comparison of (15a) with previous results [8,13] shows that $\mathcal{R}(\alpha) = \sqrt{R_G(\alpha)}$, where R_G corresponds to Gibb's algorithm. The same equation relates the Bayesian generalizer to Gibb's algorithm, as was demonstrated by Opper and Haussler [8] with a method that makes explicit use of the committee machine architecture. The potential $V(\lambda)$ may be obtained by integration of (11):

$$V(\lambda) = \int_{t(\lambda)}^{+\infty} g(t') \left(1 + \frac{dg(t')}{dt'}\right) dt', \quad (17)$$

where $t(\lambda)$ is given by the inversion of $\lambda = t + g(t; \alpha)$, and we imposed that $V(+\infty) = 0$. This optimal potential endows the perceptron with Bayesian generalization performance and depends *implicitly* on the size of the training set through T . It presents a logarithmic divergence $V(\lambda) \approx -T^2 \ln(\lambda)$ for $\lambda \rightarrow 0^+$. As $V(\lambda) = \infty$ for negative stabilities to ensure that $\lambda(t)$ is single valued, the optimal weight vector lies *within* the version space. For $\lambda \rightarrow \infty$, $V(\lambda) \approx T^3 \exp(-\lambda^2/2T^2)/\lambda$. Thus the range of the potential decreases for increasing values of α and vanishes as $\alpha \rightarrow \infty$ showing that the most relevant patterns for learning are located within a narrow window, on both sides of the student's hyperplane, whose width shrinks like T for increasing α ($T \sim 1/\alpha$ for $\alpha \gg 1$). With this cost function, the optimal generalizer may be found by a simple gradient descent, with neither the need to train an infinite number of perceptrons for implementing a committee machine, as was suggested by Opper and Haussler [8], nor to determine a large number of "samplers" of the version space, as proposed by Watkin [10].

Once the potential is known, it is straightforward to calculate the distribution of stabilities of the training set:

$$\rho(\gamma) = \left\langle \left\langle \frac{1}{P} \sum_{\mu} \delta(\gamma - \gamma^\mu) \right\rangle \right\rangle. \quad (18)$$

Its general expression is [5]

$$\rho(\gamma) = 2 \int_{-\infty}^{\infty} Dt H\left(-\frac{t}{T}\right) \delta[\lambda(t) - \gamma], \quad (19)$$

with $\lambda(t) = t + g(t; \alpha)$. In terms of $t(\gamma)$,

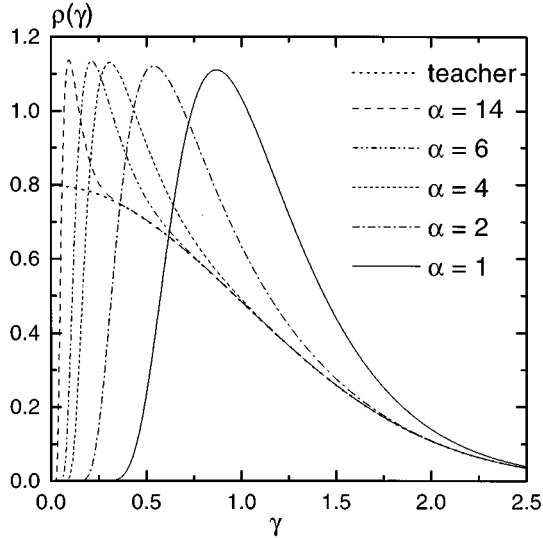


FIG. 1. Distribution of distances of the training patterns to the Bayesian separating hyperplane for different values of α .

$$\rho(\gamma) = \sqrt{\frac{2}{\pi}} \exp\left(-\frac{t^2(\gamma)}{2}\right) H\left(\frac{-t(\gamma)}{T}\right) \frac{dt}{d\gamma}(\gamma), \quad (20)$$

which depends on α through T . In the present case, as all the patterns have positive stabilities, $\rho(\gamma)$ is the distribution of the distances of the training patterns to the student's hyperplane. Distributions obtained through a numerical inversion of $\lambda(t)$, for several values of α , are plotted in Fig. 1 [14]. The density of patterns is exponentially small, $\rho(\gamma) \approx [T/\pi\gamma] \exp[-T^2/2(\mathcal{R}\gamma)^2]$ at small distance to the hyperplane. It increases with γ up to a maximum at $\gamma_M(\alpha)$. At larger γ there is a crossover to a Gaussian distribution, $\rho(\gamma) \approx (\sqrt{2/\pi}) \exp(-\gamma^2/2)$ identical to the teacher's one. Both $\gamma_M(\alpha)$ and the crossover distance get closer to the hyperplane with increasing α . In the large α limit, both quantities vanish like $1/\alpha$, with $\gamma_M \approx 1.769/\alpha$. Thus the region of disagreement between the student's and the teacher's distributions decreases for increasing size of the training set. In the limit $\alpha \rightarrow \infty$, the Bayesian distribution is identical to the teacher's one.

It is worthwhile to compare the present results with the MSP, whose weight vector is the one with maximal distance from all the hyperplanes that define the version space. The corresponding distribution $\rho(\gamma)$ presents a gap for $\gamma < \kappa$, and a δ peak at $\gamma = \kappa$, which is precisely half the smallest width of the version space. In the large α limit, $\kappa \approx 1.004/\alpha$ is smaller than γ_M . The fact that the Bayesian student has patterns at vanishing distance from the hyperplane, and has most patterns at distances larger than κ , allows us to conclude that its weight vector lies close to the boundary of the version space. It has been shown [10] that the Bayesian weight vector is the barycenter of the (strictly convex) version space. Our result means that the barycenter of the version space is far from its center, which is rather surprising, and might indicate that the version space is highly nonspherical. Notice that the teacher weight vector lies even closer to the version space boundary, as it has a finite distribution of stabilities for all $\gamma > 0$. This explains why some potentials recently proposed [5,9] may reach a generalization error

lower than the MSP, in spite of the fact that they find a solution *outside* the version space, i.e., without correctly learning the complete training set.

III. SIMULATION RESULTS

The theoretical results of the preceding section were obtained in the thermodynamic limit, $N \rightarrow \infty$, $P \rightarrow \infty$, with $\alpha = P/N$ finite. In this section we present results of thorough numerical simulations that confirm very nicely the theoretical predictions, and are precise enough to determine the finite size corrections.

We describe first our implementation of the learning procedure. Given a training set, the optimal student is found by a gradient descent on the cost function (4) with potential (17). In practice, only the derivative of the potential is needed, and we do not need to perform the integration in (17). As $dV/d\lambda$ is the function $-g(t; \alpha)$ defined by Eq. (16), evaluated at $t = t(\lambda)$ given by (8), we only have to invert the equation $\lambda(t) = t + g(t; \alpha)$. We calculated numerically $dV/d\lambda$ for each value of α considered. As the optimal potential diverges for negative stabilities, the minimization has to be started with a weight vector $\mathbf{w}(0)$ already inside the version space. In our simulations, we determined $\mathbf{w}(0)$ by minimization of the cost function (4) with potential $V(\lambda) = 1 - \tanh(\beta\gamma/2)$, in which the value of β has to be optimally tuned [5]. We used the implementation called Minimizer [15], that finds the best value of β together with the weights $\mathbf{w}(0)$ through a deterministic annealing. Starting from $\mathbf{w}(0)$, the weights are iteratively modified through

$$\mathbf{w} = \mathbf{w}(k) - \epsilon(k) \delta \mathbf{w}, \quad (21a)$$

$$\delta \mathbf{w} = \sum_{\mu} \frac{dV}{d\lambda}(\gamma^{\mu}; \alpha) \tau^{\mu} \xi^{\mu}, \quad (21b)$$

$$\mathbf{w}(k+1) = \sqrt{N} \frac{\mathbf{w}}{\sqrt{\mathbf{w} \cdot \mathbf{w}}}, \quad (21c)$$

where γ^{μ} is the stability (5) of pattern μ . Actually, the derivative $\partial E/\partial \mathbf{w}$ has two terms, and only one of them is taken into account in Eq. (21b). The neglected term, that contributes to keep $\mathbf{w} \cdot \mathbf{w}$ constant only to first order in ϵ , has been replaced by the normalization (21c). A straightforward calculation shows that the component of $\delta \mathbf{w}$ [Eq. (21b)] orthogonal to $\mathbf{w}(k)$, $\delta \mathbf{w}_{\perp} \equiv \delta \mathbf{w} - \mathbf{w}(k) \delta \mathbf{w} \cdot \mathbf{w}(k)/N$, is proportional to $\partial E/\partial \mathbf{w}$. Thus, at convergence, $\delta \mathbf{w}_{\perp}^2 \equiv \delta \mathbf{w}_{\perp} \cdot \delta \mathbf{w}_{\perp}$ vanishes. Actually, the stopping condition in all our simulations was $\delta \mathbf{w}_{\perp}^2 \leq 10^{-14}$.

The variable learning rate $\epsilon(k)$, introduced to speed up the convergence, is determined as follows. At each iteration, we calculate (21a) for three different values of ϵ : $\epsilon(k-1)/2$, $\epsilon(k-1)$, and $5\epsilon(k-1)$. The value $\epsilon(k-1)/2$ should prevent the oscillations that may appear for too large learning rates, whereas $5\epsilon(k-1)$ allows to accelerate the convergence in regions where the potential is flat. At each iteration, we keep for $\epsilon(k)$ the value that minimizes $\delta \mathbf{w}_{\perp}^2$. With this procedure, the initialization of ϵ is irrelevant; we used $\epsilon(0) = 10^{-2}$ in all our tests.

In our simulations, we determined the generalization error $\epsilon_g(\alpha, N)$ and the distribution of stabilities $\rho(\gamma; \alpha, N)$ as a

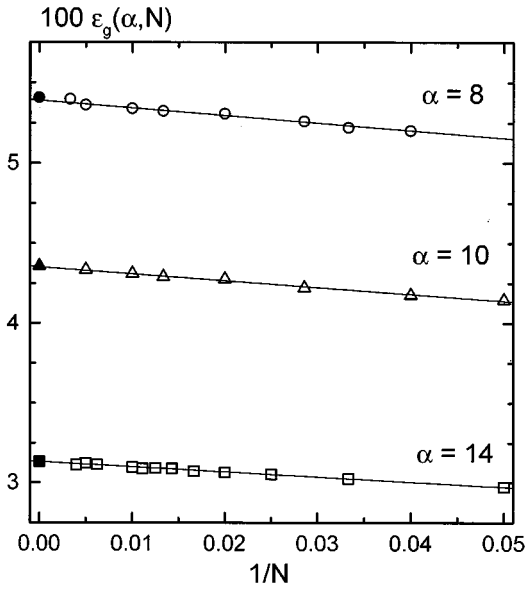


FIG. 2. Average generalization error vs $1/N$. Error bars are not visible at the scale of the figure. Lines are least squared fits to the numerical data, which are extrapolated to $1/N=0$. Full symbols correspond to the theoretical values.

function of N and α . Given α and N , we generated training sets L_α of $P = \alpha N$ binary patterns. The components of the input patterns ξ_i^μ of each training sample were chosen at random with probability $p(\xi_i^\mu = 1) = p(\xi_i^\mu = -1) = 1/2$ for all $1 \leq i \leq N$ and $1 \leq \mu \leq P$. The corresponding outputs $\tau^\mu = \text{sgn}(\mathbf{v} \cdot \xi^\mu)$ are determined by a randomly selected teacher of normalized weights \mathbf{v} ($\mathbf{v} \cdot \mathbf{v} = N$). We made simulations for $\alpha = 1, 2, 4, 6, 8, 10$, and 14 , and we considered at least seven different values of N for each value of α . Each training set was learnt with the optimal potential using (21), as explained before. The overlap between the obtained normalized weights $\mathbf{w}^*(L_\alpha)$ and the teacher's weights \mathbf{v} , $R(L_\alpha, N) = \mathbf{w}^* \cdot \mathbf{v} / N$, determines the generalization error of the student perceptron, $\epsilon_g(L_\alpha, N) = \arccos[R(L_\alpha, N)] / \pi$.

We determined the generalization error for each pair (α, N) , averaged over $M(\alpha, N)$ training sets, $\epsilon_g(\alpha, N) = \sum_{\{L_\alpha\}} \epsilon_g(L_\alpha, N) / M(\alpha, N)$. The number of samples $M(\alpha, N)$ was chosen large enough to have a good precision in the extrapolation to $1/N \rightarrow 0$. Values of $M(\alpha, N)$ ranging from 500 to 20 000 (the larger number of samples corresponding to the smaller values of $P = \alpha N$) were used. Most of the simulations were done on a parallel computer that allows for 64 samples to be processed simultaneously. The obtained values of $\epsilon_g(\alpha, N)$ are represented on Fig. 2 as a function of $1/N$. All the investigated values of α show the same behavior, and only some of them are reported on the figure for reasons of clarity. The generalization errors are linear in $1/N$ because for each α we only considered values of N large enough that the second order corrections in $1/N$ be negligible. The fits to the numerical results extrapolate correctly to the theoretical values $\epsilon_g(\alpha)$ obtained in the thermodynamic limit $N \rightarrow \infty$, $P \rightarrow \infty$ with $\alpha = P/N$ constant. The finite size corrections are *negative*, meaning that in finite dimension the expected generalization error is *lower* than predicted by the theory. This result can be understood if one

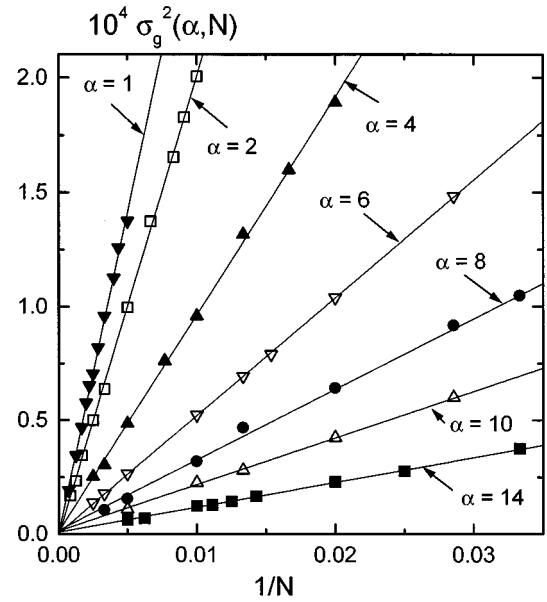


FIG. 3. Variance of the generalization error vs $1/N$. Lines are least squared fits to the numerical data.

considers the information content of the training set instead of its size. As the number of possible training patterns is 2^N , the training set carries (on the average) a fraction of information $\alpha N / 2^N$ which, at constant α , is larger the smaller N . Moreover, given $\alpha > 2$, there is always a value N_α small enough that $\alpha N_\alpha > 2^{N_\alpha}$, i.e., such that *all* the possible patterns belong to the training set. One expects that $\epsilon_g(\alpha, N_\alpha) = 0$, and that $\epsilon_g(\alpha, N)$ increases smoothly for increasing N to reach $\epsilon_g(\alpha, \infty)$ from below.

The variance of the generalization error, $\sigma_g^2(\alpha, N) = \sum_{\{L_\alpha\}} (\epsilon_g(L_\alpha, N) - \epsilon_g(\alpha, N))^2 / M(\alpha, N)$, is represented as a function of $1/N$ on Fig. 3 for all the values of α considered. The fact that all the lines extrapolate to zero shows that, in the thermodynamic limit, the distribution of $\epsilon_g(L_\alpha, N)$ is a delta function: any randomly selected training set corresponding to the same α endows the perceptron with the same typical generalization error, with probability one. In other words, the hypothesis of self-averaging, underlying the statistical mechanics calculations, is correct.

At finite size, the average generalization error and its variance depend on $P = \alpha N$. To first order in $1/P$, we may write

$$\epsilon_g(\alpha, N) = \epsilon_g(\alpha, \infty) - \phi(\alpha) / P, \quad (22)$$

$$\sigma_g^2(\alpha, N) = \psi(\alpha) / P. \quad (23)$$

The behavior of $\phi(\alpha)$ and $\psi(\alpha)$, displayed on Figs. 4 and 5, presents a crossover at $\alpha \approx 2$, i.e., in the neighborhood of the perceptron's capacity. At large α , $\phi(\alpha)$ is constant and $\psi(\alpha)$ decreases smoothly, whereas at small α , both quantities increase with α . Thus, as a function of P , finite size corrections to ϵ_g vanish slower at $\alpha \leq 2$ than at large α . This is the reason why we needed a larger number of samples for low α in our simulations.

As N decreases, the mean value of the generalization error distribution, $\epsilon_g(\alpha, N)$, shifts towards lower values, proportionally to $1/N$. However, the broadening of the distribu-

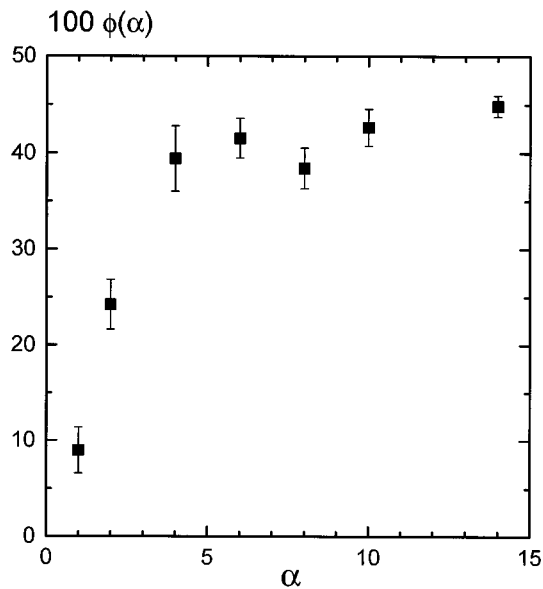


FIG. 4. Slope of the finite size corrections to the generalization error.

tion, $\sigma_g(\alpha, N) \sim 1/\sqrt{N}$, overcompensates this effect. Thus, in spite of the negative correction to $\epsilon_g(\alpha, \infty)$ at finite N , there is a finite probability that a particular trained perceptron generalize worse than the theoretical prediction.

The distributions of stabilities follow the same trends as the generalization error. Histograms, determined with some of our results, are compared to the theoretical density distributions, on Figs. 6 and 7. On Fig. 6, numerical results for both the student and the teacher perceptrons, are displayed. Although not clearly visible on the figure, the finite size teacher has less patterns at small distances to the separating hyperplane, the tail of the distribution being slightly higher,

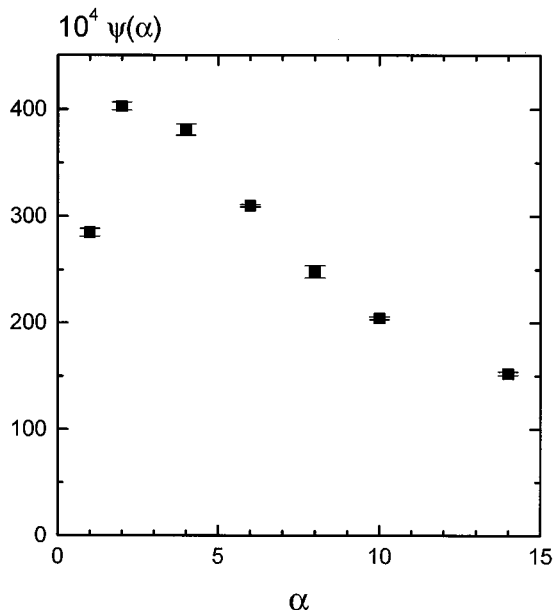


FIG. 5. Slope of the finite size variance of the generalization error.

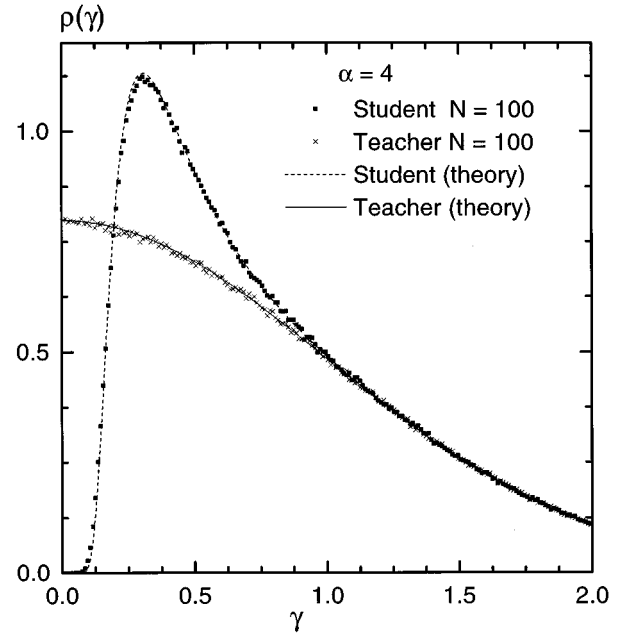


FIG. 6. Theoretical and numerical ($N=100$) distribution of stabilities for the optimal student and the teacher for $\alpha=4$.

than the theoretical distribution. These discrepancies are much smaller than the finite size effects on the student perceptrons, which exhibit an increase of the pattern density closer to the hyperplane, with a corresponding depletion of the peak at γ_M . These effects are enhanced at smaller N , as may be seen on Fig. 7.

IV. CONCLUSION

In this paper, we presented numerical simulations of the simplest neural network, the perceptron, learning optimally a linear separation task from examples. They confirm the the-

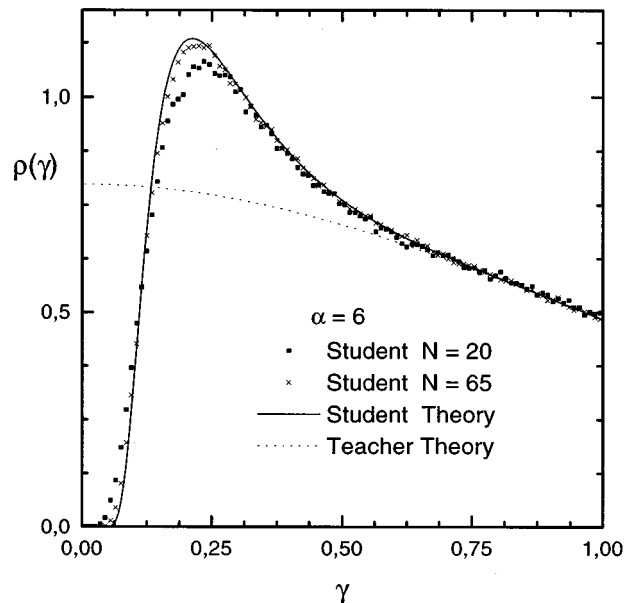


FIG. 7. Theoretical and numerical ($N=20$ and 65) distribution of stabilities for the optimal student for $\alpha=6$.

oretical predictions and present interesting finite size scaling behaviors.

After a derivation of the optimal learning potential, we deduced the theoretical distribution of distances of the learned patterns to the separating hyperplane, $\rho(\gamma)$. Surprisingly, the optimal student is predicted to be close to the boundary of the version space instead of being near of its center, as currently believed.

We presented extensive numerical simulations with the aim of clarifying to which extent the theoretical results, which predict the typical behaviour of the generalization error and the distribution of stabilities in the thermodynamic limit, are valid for finite size systems. In particular, the numerically determined distribution of stabilities shows that finite size optimal perceptrons lie even closer to the version space boundary than the theoretical prediction for $N \rightarrow \infty$. The extrapolation of the generalization error ϵ_g to $1/N \rightarrow 0$ averaged over a large number of samples, confirm the theoretical predictions with very high accuracy. The variance of ϵ_g vanishes in that limit, showing that all the training sets endow the perceptron with the same generalization error, with probability one. This is just what is meant by the hypothesis of self-averaging underlying the replica approach, which is thus numerically validated.

At finite N the mean generalization error is *smaller* than the theoretical value. As the argument that allows to understand such result is independent of any learning scheme, for it takes into account only the information content of the training set, we expect it to be also valid for statistical mechanics predictions of ϵ_g for other learning algorithms. However, it is worth pointing out that the width of the generalization error distribution grows with decreasing N faster than the shift of the mean value.

As a function of α , $\epsilon_g(\alpha, N)$ shows two different scaling regimes, depending on whether $\alpha > 2$ or $\alpha < 2$. The crossover at $\alpha_c = 2$ might be correlated to the perceptron capacity. As below α_c any training set is expected to be linearly separable, it seems likely that the generalization error presents a different scaling at $\alpha < \alpha_c$. Theoretical calculations of finite size corrections remain to be done, to clarify the observed scaling regimes.

Although the simulations were done for binary random input vectors, the behavior of the generalization error should be the same for continuous input vectors whose components have zero mean and unit variance, as the theoretical results only depend on the two first moments of the pattern distribution. It would be interesting to see whether the observed crossover at $\alpha \approx 2$ persists in this case.

-
- [1] H. S. Seung, H. Sompolinsky, and N. Tishby, *Phys. Rev. A* **45**, 6056 (1992).
- [2] W. Krauth and M. Mézard, *J. Phys. (France)* **50**, 3057 (1989).
- [3] L. Bottou and V. Vapnik, *Neural Comput.* **4**, 888 (1992).
- [4] V. N. Vapnik, *The Nature of Statistical Learning Theory* (Springer-Verlag, New York, 1995).
- [5] M. B. Gordon and D. Grempel, *Europhys. Lett.* **29**, 257 (1995).
- [6] M. Bouten, J. Schietse, and C. Van den Broeck, *Phys. Rev. E* **52**, 1958 (1995).
- [7] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973).
- [8] M. Opper and D. Haussler, *Phys. Rev. Lett.* **66**, 2677 (1991).
- [9] R. Meir and J.F. Fontanari, *Phys. Rev. A* **45**, 8874 (1992).
- [10] T. L. H. Watkin, *Europhys. Lett.* **21** (8), 871 (1993).
- [11] O. Kinouchi and N. Caticha, *Phys. Rev. E* **54**, R54 (1996).
- [12] M. Bouten, *J. Phys. A* **27**, 6021 (1994).
- [13] G. Györgyi and N. Tishby, in *Neural Networks and Spin Glasses*, edited by W.K. Theumann and R. Köberle (World Scientific, Singapore, 1990), pp. 3–36.
- [14] After completion of this work, we learned that O. Kinouchi derived a similar expression for the distribution of stabilities (unpublished).
- [15] B. Raffin and M. B. Gordon, *Neural Comput.* **7**, 1182 (1995).