Calculation of learning curves for inconsistent algorithms

R. Meir
Bellcore 2E-330, 445 South Street, Morristown, New Jersey 07960

J. F. Fontanari
Instituto de Física e Química de São Carlos, Universidade de São Paulo, Caixa Postal 369, 13560 São Carlos, São Paulo, Brazil

(Received 6 November 1991)

The training and generalization errors of three well-known learning algorithms are calculated using methods of statistical physics. We focus in particular on inconsistent algorithms that are unable to perfectly classify the training examples, and show that the asymptotic behavior of these algorithms is different from the case of consistent algorithms. Our results are in agreement with bounds derived by computational learning theorists. We further find that the replica-symmetric theory is stable everywhere for two of the algorithms studied, which leads us to conjecture that it is the exact solution in these cases. We also demonstrate that one of the algorithms studied performs almost indistinguishably from the Bayes learning algorithm, while having the advantage of being implementable in a single-layer network.

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

I. INTRODUCTION

Much recent progress in the theoretical understanding of the learning ability of neural networks has come from two different sources. On the one hand, researchers in the computational learning community have pursued a line of investigation pioneered in the 1970s particularly by Vapnik and Chernovenkins [1], which focused on the analysis of the worst case performance of learning algorithms. We refer in what follows to work along these lines as the VC theory, although many new results have been obtained since the seminal work of Vapnik and Chernovenkins. While these analyses have produced an impressive array of results under very general conditions [2], it has not been clear how these results relate to specific instances, or to the average case performance. In particular, the VC results usually take the form of upper bounds on the worst case performance of *any* consistent learning algorithm for arbitrary probability distributions over both the space of examples and the space of learning tasks. Thus it is not clear how tight these bounds are and how they are related to the average case performance of specific learning algorithms in typical situations. The term *consistent* refers in the learning context to algorithms which are able to learn the training examples perfectly, i.e., do not misclassify any of them.

A second line of research, initiated by the papers of Gardner [3] and Gardner and Derrida [4], is concerned with the calculation of the average case performance of *specific* learning algorithms, under very restrictive conditions. In particular, the results obtained so far are limited to the class of single-layer feedforward neural networks, with particularly simple probability distributions. This approach suffers from the additional problem of not being mathematically rigorous. We note, however, that results obtained with this approach have in general given very good agreement with computer simulations [5,6]. Moreover, at the present time we are not aware of any alternative technique for calculating the average case performance.

An initial attempt at bridging the gap between the two approaches discussed above has been recently taken in Ref. [7]. One of the main thrusts of that paper is that the results obtained from the general VC dimension analysis provide, in fact, rather tight bounds on the average case performance as calculated by the statistical physics approach poineered by Gardner. In this paper, however, we cover a situation which has only very recently been addressed by VC-dimension theorists [8]. In particular, we focus on learning algorithms that produce hypotheses which are not necessarily consistent with the training examples, but that nonetheless produce small generalization errors. Three batch mode algorithms are considered in the present work: (i) the zero-temperature Monte Carlo algorithm [9], (ii) the perceptron learning rule [10], and (iii) the relaxation algorithm [11]. We will show that a careful choice of a single parameter in the latter algorithm yields performance results which are *almost* indistinguishable from the Bayes learning algorithm studied recently by Opper and Haussler [12], while having the merit of being implementable in a single-layer network. The Bayes algorithm, on the other hand, requires a hidden layer in order to be implemented.

Our study incorporates a margin parameter κ which biases the algorithms to favor networks less sensitive to the effects of noise in the training data. However, since in this case the quantity to be minimized is not the classification error [see the definition in Eq. (2.10) below], but rather the training error, such networks can in fact produce nonzero classification errors for large enough training sets. Thus the tuning of κ allows us to distinguish between consistent and inconsistent algorithms. We also find that the asymptotic dependence of the generalization error on the size of the training set is different,

depending on whether the learning algorithm is consistent or not. We note that Griniasti and Gutfreund [13] have recently investigated these learning algorithms in the context of recurrent neural networks, with particular emphasis on their retrieval properties.

The remainder of this paper is organized as follows. In Sec. II we introduce the model in detail, after which we present in Sec. III the results of its analytic solution within the replica symmetric theory. This section also includes a discussion of the stability of the solution, as well as a comparison of the results with the annealed approximation. The results of the replica symmetric theory are discussed in detail in Sec. IV. Finally we present a summary of our results in Sec. V, together with some open questions.

II. MODEL

We consider the problem of learning from examples in single-layer neural networks. We assume the examples presented to the learning network, to be referred to as the *student,* are drawn from another single-layer perceptron, which we name the *teacher.* The examples are drawn from a probability distribution $\nu(S)$, while the teacher function f is drawn from a probability distribution $\mathcal{P}(f)$. The main question we address in this paper is that of obtaining average learning curves for the three algorithms under study. By learning curve we refer to a plot of the generalization error as a function of the size of the training set. The average will be taken with respect to these two probability distributions. We then compare the performance of these algorithms to that of the optimal Bayesian classifier, analyzed recently by Opper and Haussler [12].

We focus on the binary input/output case here, where both the input and the output of the student and teacher networks are binary ± 1 variables. We assume the student is exposed to a stream of $P=\alpha N$ input/output examples $(S^1,t^1),(S^2,t^2),\ldots,(S^P,t^P)$, where $S^l=(S_1^l,S_2^l,\ldots,S_N^l)$ and the outputs t^l are assumed to be generated by a single-layer perceptron. Thus the targets t^l are given by

$$t^l=\mathrm{sgn}\left[\sum_{j=1}^{N} W_j^0 S_j^l\right] . \tag{2.1}$$

The response of the student network to an input S^l is similarly given by

$$\sigma^l=\mathrm{sgn}\left[\sum_{j=1}^{N} W_j S_j^l\right] . \tag{2.2}$$

The objective of the learning process is to be able to predict the outcome of a random input with as small an error as possible. To quantify this notion we define a *generalization error* which measures this probability, and is simply given by

$$\epsilon(\mathbf{W})=\int d\nu(\mathbf{S})\Theta(-t\sigma) \tag{2.3}$$

where the Heaviside function $\Theta(-t\sigma)$ is 0 or 1 depending on whether the network output σ is correct or not.

Another quantity of interest is the *training error* which measures the cumulative error on the training set $\mathbf{S}^1,\ldots,\mathbf{S}^P$ and is usually taken to be of the form

$$E(\mathbf{W})=\sum_{l=1}^{P}\mathcal{E}(\mathbf{W};\mathbf{S}^l) , \tag{2.4}$$

where the specific choice of the function $\mathcal{E}(\mathbf{W};\mathbf{S})$ is up to the trainer. The learning algorithm corresponding to a particular choice of error function would then just be given by the gradient descent rule

$$\frac{dW_i}{dt}=-\eta\frac{\partial E(\mathbf{W})}{\partial W_i} . \tag{2.5}$$

In order to describe the error function it is useful to define the variable

$$\Delta(\mathbf{W},\mathbf{S},t)=\frac{t}{\sqrt{N}}\sum_{j=1}^{N} W_j S_j . \tag{2.6}$$

The first form we consider is just the 0,1 loss function, studied in detail by Gardner and Derrida (GD) [4]. Although this error function does not give rise to a gradient-descent learning algorithm through Eq. (2.5), as it is piecewise constant, it is still useful to investigate its properties since it is directly related to the actual classification error [see Eq. (2.10) below]. Moreover, the GD error function can be minimized through simulated annealing methods [14], although this approach is of course much more computationally expensive than gradient-descent learning. Following Gardner and Derrida [4] we define

$$\mathcal{E}^{\mathrm{GD}}(\mathbf{W};\mathbf{S})=\Theta(\kappa-\Delta) . \tag{2.7}$$

Obviously for $\kappa=0$ this choice of error function corresponds to the number of misclassifications. The second function we are interested in is the perceptron function (P) [10,15],

$$\mathcal{E}^{P}(\mathbf{W};\mathbf{S})=(\kappa-\Delta)\Theta(\kappa-\Delta) , \tag{2.8}$$

while the third function considered is the relaxation function (R) [15],

$$\mathcal{E}^{R}(\mathbf{W};\mathbf{S})=(\kappa-\Delta)^2\Theta(\kappa-\Delta) . \tag{2.9}$$

The three error measures we focus on have been well known for many years, although except for the first one we do not know of any analytic calculation of the learning curves for these algorithms. The learning algorithm corresponding to the choice (2.8) is just the celebrated perceptron learning algorithm [10] (performed in batch mode), while that corresponding to (2.9) is the relaxation algorithm, introduced by Agmon [11] as a method to solve a set of linear inequalities, and later generalized by Mays [16] to include the margin parameter κ. In fact these authors were concerned with the on-line version of the learning process, where the weight adaptations are made after each pattern presentation. The relaxation algorithm has recently also been studied in the context of neural networks by Anlauf and Biehl [17].

It is important at this stage to emphasize the difference between the *training* error given by (2.4) and the *classification* error $E_c(\mathbf{W})$, which measures the number of

misclassifications on the training set. This distinction is only relevant, of course, in the case $\kappa > 0$. For $\kappa = 0$, both the classification and the training error will be shown to vanish for any α. Thus we define

$$E_c(\mathbf{W}) = \sum_{l=1}^{P} \Theta(-t^l \sigma^l) , \tag{2.10}$$

with t^l and σ^l being given by (2.1) and (2.2), respectively. As we point out in Appendix B, the classification error (2.10) becomes nonzero above a critical value of α, after which it increases to a maximum and then decreases to zero for $\alpha \to \infty$. The fact that the classification error is nonzero substantiates our claim that for $\kappa > 0$ the algorithms are in fact inconsistent. It may be argued at this stage that introduction of the margin term κ does not seem to make much sense, since it produces a nonzero classification error. However, as we show below, if the training set is of limited size, it is always better to train the network with nonzero κ, thus sacrificing a nonzero classification error for a minimal generalization error. Moreover, Mays [16] has shown that training with $\kappa > 0$ speeds up the learning, although this is an issue we do not address here.

It should further be noted that if a ground-state energy of zero exists for one of these error functions, it exists for all three. Thus we expect that as long as the number of training examples is not large, so that the training error is zero, all three error functions will produce identical results. The difference between them becomes apparent when the training error is nonzero. It should also be noted that with these definitions it is quite possible to have the generalization error smaller than the training error for $\kappa > 0$. This is due to the fact that we have defined the generalization error in Eq. (2.3) as the probability of misclassification, without reference to κ, while the training error vanishes only if $\Delta(\mathbf{W},\mathbf{S},t) > \kappa$ for all training patterns. We believe that this is the sensible definition in this case, since the generalization error should not be defined with respect to the learning parameter κ.

At this stage it is useful to introduce a probabilistic element into the problem. Since the only information available to us is the training error, we would like to consider the space of all networks of a *given* training error. As we are assuming no prior knowledge, it is natural to require that all networks of a given training error are equivalent. Within the space of all networks (of a given architecture) it is useful to introduce a probability distribution which is constrained solely by the given average training error. Using the maximum entropy principle, which requires that we pick the probability distribution of maximal entropy which obeys the constraints (in this case on the training errors), we are immediately led to the standard Gibbs distribution

$$\mathcal{P}(\mathbf{W}) = Z^{-1} e^{-\beta E(\mathbf{W})} , \tag{2.11}$$

where the partition function Z is given by

$$Z = \int d\mu(\mathbf{W}) e^{-\beta E(\mathbf{W})} \tag{2.12}$$

and $\beta = 1/T$ is the inverse "temperature." In order to impose a measure $d\mu(\mathbf{W})$ on weight space, we follow Gardner [3] and restrict the weights to be real variables, confined to lie on the surface of the N-dimensional sphere of radius N,

$$d\mu(\mathbf{W}) = \prod_{i=1}^{N} \frac{dW_i}{\sqrt{2\pi e}} \delta(W \cdot \mathbf{W} - N) . \tag{2.13}$$

Having defined a probability distribution over the space of networks we may proceed now to define the average training and generalization errors as follows:

$$\epsilon_t(T,P,\kappa) = P^{-1} \langle\langle \langle E(\mathbf{W}) \rangle_T \rangle\rangle , \tag{2.14}$$

$$\epsilon_g(T,P\kappa) = \langle\langle \langle \epsilon(\mathbf{W}) \rangle_T \rangle\rangle , \tag{2.15}$$

where the thermal average $\langle \ \rangle_T$ is taken with respect to the probability distribution $\mathcal{P}(\mathbf{W})$ and $\langle\langle \ \rangle\rangle$ is an average over the probability distribution of the examples $\nu(\mathbf{S})$. We assume that the examples S_i^l are independent, identically distributed ± 1 variables, with Prob(1) $=$Prob$(-1) = \frac{1}{2}$. Note that in principle we should also average over the probability distribution of the teacher $\mathcal{P}(f)$. However, as we point out in Appendix A, if certain assumptions are made about this distribution one can do away with this further average. The free energy is given by the *quenched* average

$$F(T,P,\kappa) = -T \langle\langle \ln Z \rangle\rangle , \tag{2.16}$$

for which the training error as well as the entropy may be obtained by the thermodynamic formulas

$$\epsilon_t = \frac{1}{P} \frac{\partial(\beta F)}{\partial \beta} , \tag{2.17}$$

$$S = -\frac{\partial F}{\partial T} . \tag{2.18}$$

It is also useful to define the average classification error ϵ_c,

$$\epsilon_c(t,P,\kappa) = P^{-1} \langle\langle \langle E_c(\mathbf{W}) \rangle_T \rangle\rangle , \tag{2.19}$$

where $E_s(\mathbf{W})$ is given by (2.10). It is not difficult to see that ϵ_c obeys the relations $\epsilon_c \le \epsilon_t$, as well as $\epsilon_c \le \epsilon_g$.

In what follows we will concern ourselves with the zero-temperature limit $\beta \to \infty$. However, as we are interested in deriving explicit expressions for the training errors we must consider the nonzero temperature case first, then taking the limit in Eq. (2.17) *after* calculating the derivative. We note that learning at nonzero temperature has been shown to be advantageous when the training examples themselves are noisy [18]. We limit ourselves in this study to the case of error free examples.

Probably the most interesting question one can ask about the above learning algorithms is how well they generalize. Using the framework of statistical physics, as applied by Gardner and Derrida [4] to this class of problems, we are able to answer this question. As we show in Sec. IV, the three algorithms have different generalization abilities. We do not address here the issue of the convergence properties of these algorithms. For a discussion of this issue the reader may consult the textbook by Duda and Hart [15].

III. THE REPLICA APPROACH

The mathematical problem we are faced with at this stage is the calculation of the average free-energy density in the thermodynamic limit $N\to\infty$, where the average is taken with respect to the probability distribution of the examples $d\nu(S)$. Since the free energy is related to the logarithm of the partition function, we resort to the well-known replica trick

$$\lim_{N\to\infty}\langle\langle \ln Z\rangle\rangle=\lim_{N\to\infty}\lim_{n\to 0}\frac{\langle\langle Z^n\rangle\rangle-1}{n} . \tag{3.1}$$

In order to proceed we follow a common strategy in the physics literature, by first calculating $\lim_{N\to\infty}\langle\langle Z^n\rangle\rangle$ and then taking the limit $n\to 0$. While this interchange of limits has been shown to be valid for the spin-glass problem [19], we do not know of any arguments for its validity in general.

Using the standard approach to these problems [3,4] we proceed to evaluate $\lim_{N\to\infty}\langle\langle Z^n\rangle\rangle$, obtaining the following expression:

$$\lim_{N\to\infty}\langle\langle Z^n\rangle\rangle=\int\prod_{\substack{\alpha,\beta\\ \alpha<\beta}}\frac{dq_{\alpha\beta}d\hat{q}_{\alpha\beta}}{2\pi i/N}\times\int\prod_{\alpha}\frac{dR_\alpha d\hat{R}_\alpha}{2\pi i/N}\times e^{-NF(q_{\alpha\beta},\hat{q}_{\alpha\beta},R_\alpha,\hat{R}_\alpha)} . \tag{3.2}$$

The integrals are then evaluated in the limit $N\to\infty$ by the standard saddle-point method [20]

$$\int\prod_{i=1}^{m}dx_i e^{-NF(x)}\approx(2\pi)^{m/2}N^{-m/2}\left|\mathrm{Det}\left[\frac{\partial^2 F}{\partial x_i\partial x_j}\right]\right|_{x_0}^{-1/2}\times e^{-NF(x_0)}\left[1+O\left[\frac{1}{N}\right]\right] , \tag{3.3}$$

where x_0 is the saddle point at which $\partial F/\partial x=0$, and it is assumed that the Hessian matrix $\partial^2F/\partial x_i\partial x_j$, evaluated at x_0, is a positive definite matrix (i.e., the point x_0 is a true minimum).

Since the expression for F is rather lengthy, we direct the reader to Appendix A where it is explicitly displayed. The physically meaningful order parameters $q_{\alpha\beta}$ and R_α are given by

$$q_{\alpha\beta}=\left\langle\frac{1}{N}\mathbf{W}^\alpha\cdot\mathbf{W}^\beta\right\rangle_\eta , \tag{3.4}$$

$$R_\alpha=\left\langle\frac{1}{N}\mathbf{W}^\alpha\cdot\mathbf{W}^0\right\rangle_\eta , \tag{3.5}$$

where the averages are with respect to the probability measure

$$d\eta(\mathbf{W})=\prod_{\alpha=1}^{n}d\mu(\mathbf{W}^\alpha)\exp\left[\sum_{\substack{\alpha,\beta\\ \alpha<\beta}}\hat{q}_{\alpha\beta}\mathbf{W}^\alpha\cdot\mathbf{W}^\beta+\sum_\alpha\hat{R}_\alpha\mathbf{W}^\alpha\cdot\mathbf{W}^0\right] . \tag{3.6}$$

Thus we see that $q_{\alpha\beta}$ is related to the correlation between two solutions α and β, while R_α is related to the overlap of a solution α with the teacher.

A. The replica symmetric solution

To proceed further, one is required to solve the saddle-point equations for general n and then take the limit $n\to 0$. This procedure is in general very complex. In order to make progress, however, it has been customary to make an ansatz concerning the solution of these equations. In particular, the most commonly used *replica-symmetric* assumption is just that all the variables take on values independent on the replica index, i.e.,

$$q_{\alpha\beta}=q\ ,\quad \hat{q}_{\alpha\beta}=\hat{q}\quad\forall\alpha<\beta , \tag{3.7}$$

$$R_\alpha=R\ ,\quad \hat{R}_\alpha=\hat{R}\quad\forall\alpha . \tag{3.8}$$

With this ansatz in mind, one proceeds to obtain the following expression for the free-energy density in the limits $N\to\infty$ and $n\to 0$:

$$-\beta f=\frac{1}{2}\left[\frac{1-r^2}{1-q}+\ln(1-q)\right]+\alpha G_1(q,r) . \tag{3.9}$$

The specific forms for the functions G_1 at nonzero temperature, for the three cases studied, are again given in Appendix A, as they are rather lengthy. The variable r appearing in (3.9) is related to R by

$$r=\frac{R}{\sqrt{M}} \tag{3.10}$$

where $M=N^{-1}\sum_i(W_i^0)^2$. We further assume that the teacher weight components W_i^0 obey the *law of large numbers* [21] and thus M approaches a definite limit when $N\to\infty$. We elaborate on this assumption in Appendix A. It is interesting to note at this stage that all reference to the teacher probability distribution has vanished, since M was eliminated from the expression for the free energy through a rescaling of the order parameter R. Thus we conclude that if the replica theory is correct, the properties of the system are independent of the specific teacher distribution, *as long as* the random variable M approaches a definite limit for large N. We note that this situation occurs only because the student weights are real. In the case where these weights are constrained to a discrete and bounded set of values [6] we need to consider the full probability distribution of the teacher weights.

Following Gardner and Derrida [4] we realize that in the zero-temperature limit $\beta\to\infty$ we may expect two types of solutions. The first one, with $q<1$, is given simply by setting $\beta=\infty$ in Eqs. (A9)–(A11) of Appendix A, obtaining an equation valid in all three cases,

$$\lim_{\beta\to\infty}(-\beta f)=\frac{1}{2}\left[\frac{q-r^2}{1-q}+\ln(1-q)\right]+\int Dt\,H(\xi_1)\ln H(\xi_2) \tag{3.11}$$

where

$$\xi_1=\frac{rt}{\sqrt{q-r^2}} , \tag{3.12}$$

$$\xi_2=\frac{\kappa+\sqrt{q}\,t}{\sqrt{1-q}} \ . \tag{3.13}$$

We have also introduced the standard definitions

$$Dt=\frac{e^{-t^2/2}}{\sqrt{2\pi}}dt$$

and

$$H(x)=\int_x^\infty Dt \ .$$

Solving the saddle-point equations resulting from setting the derivatives of f with respect to q and r to zero we find that for $\alpha<\alpha_c(\kappa)$ they possess solutions with $q<1$. At α_c, however, $q\to 1$ and the equations become singular. Note that for $\kappa=0$, $\alpha_c=\infty$, so that there always exists a solution with $q<1$ in this case. This transition to $q=1$ at a finite value of α has been observed already by Gardner and Derrida [4] in the context of a random mapping and by Györgi and Tishby [18] in the context of learning from examples with noisy input patterns. For $\alpha>\alpha_c$, we must thus consider the limit when $\beta\to\infty$ and $1-q\to 0$ while $\beta(1-q)<\infty$. Defining

$$x=\beta(1-q) \ , \tag{3.14}$$

we find the following expressions in the zero-temperature limit:

$$f^{\mathrm{GD}}=-\frac{1-r^2}{2x}+\frac{\alpha}{x}\int_{-\kappa}^{-\kappa+\sqrt{2x}}Dt\,H(\xi)(\kappa+t)^2 +2\alpha\int_{-\kappa+\sqrt{2x}}^{\infty}Dt\,H(\xi) \ , \tag{3.15}$$

$$f^{P}=-\frac{1-r^2}{2x}+\frac{\alpha}{x}\int_{-\kappa}^{-\kappa+x}Dt\,H(\xi)(\kappa+t)^2 -\alpha\int_{-\kappa+x}^{\infty}Dt\,H(\xi)[x-2(\kappa+t)] \ , \tag{3.16}$$

$$f^{R}=-\frac{1-r^2}{2x}+\left[\frac{2\alpha}{1+2x}\right]\int_{-\kappa}^{\infty}Dt\,H(\xi)(\kappa+t)^2 \ , \tag{3.17}$$

where

$$\xi=\frac{rt}{\sqrt{1-r^2}} \ . \tag{3.18}$$

The values of the order parameter x and r are then obtained by setting the derivatives of f with respect to each of them to zero.

The zero-temperature training energy can be obtained by Eq. (3.9) together with the identity

$$\epsilon_t(T=0)=\frac{1}{\alpha}\lim_{\beta\to\infty}\frac{\partial(\beta f)}{\alpha\beta} \ . \tag{3.19}$$

In the range $\alpha<\alpha_c$, where Eq. (3.11) is valid, we find $\epsilon_t=0$, while for $\alpha>\alpha_c$ we obtain

$$\epsilon_t^{\mathrm{GD}}=2\int_{-\kappa+\sqrt{2x}}^{\infty}Dt\,H(\xi) \ , \tag{3.20}$$

$$\epsilon_t^{P}=2\int_{-\kappa+x}^{\infty}Dt\,H(\xi)[(\kappa+t)-x] \ , \tag{3.21}$$

$$\epsilon_t^{R}=2\int_{-\kappa}^{\infty}Dt\,H(\xi)\left[\frac{\kappa+t}{1+2x}\right]^2 \ . \tag{3.22}$$

We immediately observe that when $x\to\infty$ the training errors vanish in all cases, which is consistent with the results obtained below α_c. The fact that the three error functions give rise to the same free energy below α_c is expected, since it is clear from the basic definitions [Eqs. (2.7)–(2.9)] that if a zero ground-state energy exists, it exists for all models simultaneously.

B. Stability of the replica symmetric solution

As we mentioned above, the calculation of the integral (3.2) in the limit $N\to\infty$ was performed by the saddle-point method. In order to check that the replica symmetric solution is in fact a minimum and not just a saddle point, it is necessary to calculate the Hessian matrix appearing in Eq. (3.3) and check that it is in fact positive definite. Following the analysis of Gardner and Derrida [4] one can show that the local stability of the replica symmetric solution is determined by the condition

$$\alpha\gamma_0\gamma_1<1 \tag{3.23}$$

where γ_0 and γ_1 are the transverse eigenvalues [22] of the matrices $\partial^2 G_0$ and $\partial^2 G_1$ evaluated at the replica-symmetric saddle point. The expressions for G_0 and G_1 are given in Appendix A.

After some algebra, we obtain the following expressions for the stability conditions in the three cases of interest:

$$2\alpha\int_{-\kappa}^{-\kappa+\sqrt{2x}}Dt\,H(\xi)<1 \ , \quad \text{Gardner-Derrida} \tag{3.24}$$

$$2\alpha\int_{-\kappa}^{-\kappa+x}Dt\,H(\xi)<1 \ , \quad \text{perceptron} \tag{3.25}$$

$$2\alpha\left[\frac{2x}{1+2x}\right]^2\int_{-\kappa}^{\infty}Dt\,H(\xi)<1 \ , \quad \text{relaxation} \tag{3.26}$$

where ξ is defined in Eq. (3.18).

C. The annealed approximation

For the sake of completeness we present the results obtained for the annealed approximation. As is well known, in this case we replace $\langle\langle \ln Z\rangle\rangle$ by $\ln\langle\langle Z\rangle\rangle$, which greatly facilitates the calculation. The result, at zero temperature, obtained for all three learning algorithms is

$$\lim_{\beta\to\infty}(-\beta f)=\tfrac{1}{2}\ln(1-r^2)+2\alpha\int_0^\infty Dt\,H\left[\frac{\kappa-rt}{\sqrt{1-r^2}}\right] \ . \tag{3.27}$$

The saddle-point equation resulting from setting df/dr to zero is

$$r=\frac{\alpha}{2\pi}\sqrt{1-r^2}\,\frac{e^{-k^2/2(1-r^2)}}{\int_0^\infty Dt\,H\left[\frac{\kappa-rt}{\sqrt{1-r^2}}\right]} \ , \tag{3.28}$$

which possesses a single solution with $r<1$ for any finite α. This should be compared to the situation arising in the replica symmetric theory where the saddle-point

equations resulting from Eq. (3.11) become singular at a finite value of α, thus giving rise to a phase transition in that case. Moreover, the annealed approximation also predicts that the training error vanishes for any α, which contradicts the replica symmetric results. These results supply further support to the observation by Seung, Sompolinsky, and Tishby [5] that the annealed approximation gives qualitatively correct results in the consistent case, while failing for the inconsistent one.

IV. ANALYSIS OF THE RESULTS

The first question we address in the context of the replica-symmetric solution is its stability. Thus, using Eqs. (3.24)–(3.26) we would like to know what are the regions in the (α,κ) plane where the replica-symmetric solution is stable. We plot in Fig. 1 the curve $\kappa_{AT}(\alpha)$, above which the replica-symmetric solution becomes unstable for the Gardner-Derrida error function (2.7). The second line appearing in the figure, $\kappa_c(\alpha)$, is the critical line above which learning with zero training error is impossible. As we can see, there is a large area of the (α,κ) space in which the replica-symmetric solution is stable. The situation in the case of the perceptron and relaxation algorithms is even better. We find that the replica symmetric solution is stable for every α and κ. Thus it would seem like the replica-symmetric solution is the exact solution for this problem. To show this we present the stability conditions (3.25) and (3.26) for $\alpha\to\infty$, thereby demonstrating that our solution is stable even in this regime. We find in this limit

$$\frac{e^{-\kappa^2/2}}{\alpha}<1\ ,\quad \text{perceptron} \tag{4.1}$$

$$\frac{\pi[1-2H(\kappa)]}{2\alpha\kappa^2}<1\ ,\quad \text{relaxation}\ . \tag{4.2}$$

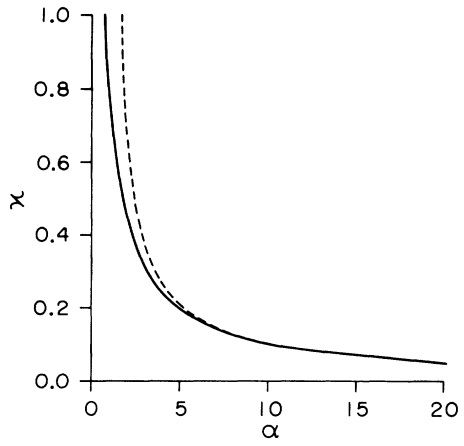

FIG. 1. Phase diagram in the (α,κ) plane. The solid line marks the critical above which the training and classification errors are nonzero. The dashed line is the instability line for the Gardner-Derrida error function, above which the replica-symmetric solution becomes unstable. The replica symmetric solution for the perceptron and relaxation models is stable everywhere.

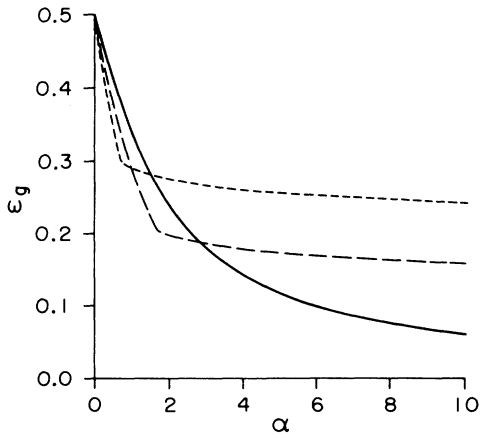

FIG. 2. Generalization error versus α for the Gardner-Derrida error function for $\kappa=0$ (solid line), $\kappa=0.5$ (long dashes), and $\kappa=1$ (short dashes).

These equations are obviously satisfied for any κ, and thus we conclude that the replica-symmetric solution is stable for the perceptron and relaxation error functions. We note that in the context of the random mapping problem, Griniasti and Gutfruend [13] have also demonstrated the increased range of stability of the replica-symmetric solution for the perceptron and relaxation error functions as opposed to the Gardner-Derrida function.

Having established the stability of the replica-symmetric solution in the three cases, we wish to study their performances as predicted by the theory. In Fig. 2 we plot the generalization error against α for the Gardner-Derrida error functions for $\kappa=0,0.5,1.0$. As can be seen in the figure, if α is small, the generalization error is minimized for large κ. It is also interesting to note that for $\kappa>0$ the generalization error changes its behavior drastically at α_c, the point at which the training error starts to deviate from zero. Qualitatively similar behavior can be observed in Figs. 3 and 4 for the perceptron and relaxation algorithms, respectively, although the

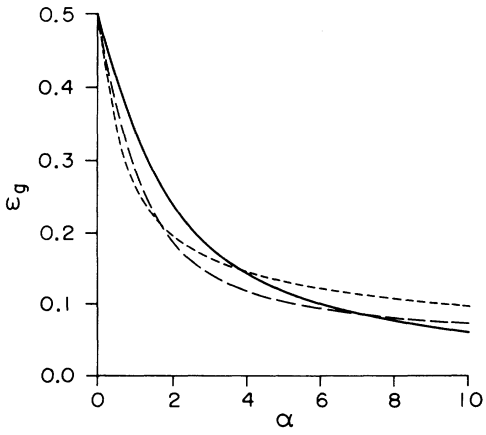

FIG. 3. Same as Fig. 2 for the perceptron error function.

curves are much smoother in this case.

In order to compare the error functions we have plotted in Fig. 5 the generalization errors for the three models studied at $\kappa=0.5$. For comparison we also present the results for the Bayes algorithm, taken from Opper and Haussler [12]. Similar curves can be obtained for other values of κ. We have found that the generalization error for the relaxation algorithm is always smaller than the other two. We also present a comparison of the training errors at $\kappa=1$ for the three error functions in Fig. 6. We observe that for large values of α the training error of the relaxation algorithm is always lowest, followed by that of the perceptron.

Looking at Fig. 5 we observe that the Bayes learning algorithm seems to decrease faster than the other algorithms we have presented. In fact, solving the saddle-point equations for large α we find to leading order in $1/\alpha$

$$\epsilon_g(\kappa>0)\approx\left[\frac{1}{\pi}\int_0^{\kappa}Dt\right]^{1/2}\frac{1}{\sqrt{\alpha}}\ , \quad \text{perceptron} \tag{4.3}$$

$$\epsilon_g(\kappa>0)\approx\left[\frac{1}{\pi\kappa^2}\int_0^{\kappa}Dt\,(\kappa-t)^2\right]^{1/2}\frac{1}{\sqrt{\alpha}}\ , \quad \text{relaxation} \tag{4.4}$$

while Opper and Haussler [12] find for the Bayes algorithm

$$\epsilon_g^{\text{Bayes}}\approx\frac{0.44}{\alpha}\ . \tag{4.5}$$

We note, however, that the prefactors to the $1/\sqrt{\alpha}$ terms in Eqs. (4.3) and (4.4) vanish for $\kappa=0$. It is thus clear that a nonzero value for κ changes the asymptotic behavior from α^{-1} to $\alpha^{-1/2}$. In fact, for $\kappa=0$ we get

$$\epsilon_g(\kappa=0)\approx\frac{0.62}{\alpha}\ , \tag{4.6}$$

which holds for all learning algorithms, and agrees with results of Opper and Haussler [12] and Seung, Sompolinsky, and Tishby [5]. An interesting observation at this point is that the asymptotic expressions we obtained in

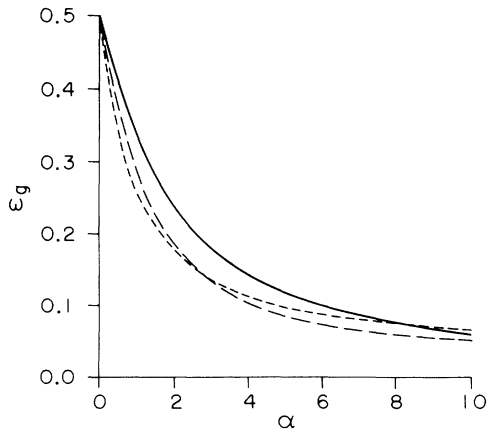

FIG. 4. Same as Fig. 2 for the relaxation error function.

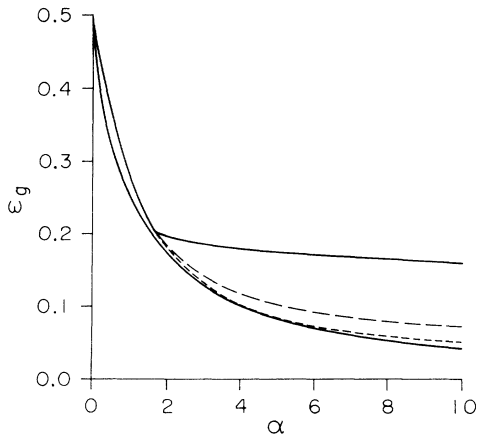

FIG. 5. Generalization error vs α for the Gardner-Derrida (upper solid line), perceptron (long dashes), relaxation (short dashes), and Bayes (lower solid line) learning algorithms. We used $\kappa=0.5$ for the first three algorithms.

Eqs. (4.3) and (4.4) seem to violate the upper bounds obtained from VC dimension analysis by Haussler, Kearns, and Schapire [7]. However, this is not the case. The results of the above authors assume that the learning algorithm learns the training set perfectly, while we show in Appendix B, that with a nonzero value for κ the classification error is positive above α_c. Thus the algorithms we have discussed do not fall in the realm of problems investigated by Haussler, Kearns, and Schapire, and thus need not obey the bounds. We note that Seung, Sompolinsky, and Tishby [5] have recently also found an $\alpha^{-1/2}$ asymptotic decay of the generalization error for a related problem. However, in their case the replica-symmetric solution was unstable in the regime $\alpha\to\infty$ and it was not clear whether this behavior would persist if the exact solution were obtained. In our case, on the other hand, since we find the replica-symmetric theory to

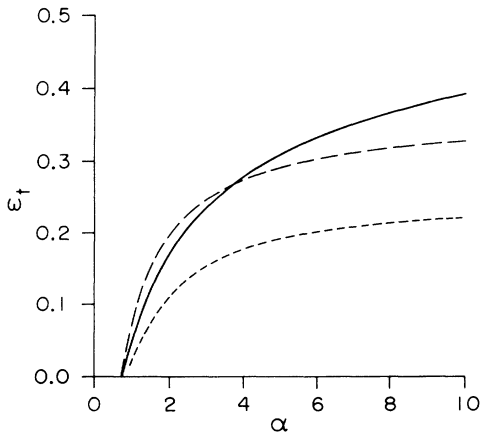

FIG. 6. Training error vs α for the Gardner-Derrida (solid line), perceptron (long dashes), and relaxation (short dashes) learning algorithms at $\kappa=1$.

be stable everywhere, we believe the above asymptotic results are exact.

In the annealed approximation we find that the asymptotic results obtained for the consistent case $\kappa=0$ behave similarly to the replica-symmetric results. On the other hand, in the inconsistent case $\kappa>0$, we find very different asymptotic behaviors in the two theories. This result provides extra support to the observation of Seung, Sompolinsky, and Tishby [5] that the annealed approximation provides a reasonable approximation in the consistent case, while failing badly in the inconsistent case. In particular we find for the annealed approximation

$$\epsilon_g(\kappa=0)\approx\frac{1}{\alpha}\ , \tag{4.7}$$

$$\epsilon_g(\kappa>0)\approx\frac{\kappa}{\pi\sqrt{2}}\frac{1}{\sqrt{\ln\alpha}}\ . \tag{4.8}$$

In Fig. 7 we present the learning curves for $\kappa=0,0.5,1.0$ obtained using the annealed approximation. While they seem to agree qualitatively with the replica-symmetric results, they scale very differently for $\alpha\rightarrow\infty$. In fact, the result (4.8) for the inconsistent case violates the upper bound of Haussler [8]. Moreover, as mentioned above, this approximation predicts zero training error, and thus zero classification error, which contradicts the replica-symmetric results.

From the above results it would seem that while it is advantageous to set $\kappa>0$ for small α, keeping κ positive give very poor asymptotic behavior compared to the $\kappa=0$ case. It is therefore interesting to choose an optimal κ in such a way that for every α the generalization error is minimized. We plot in Fig. 8 the curve $\kappa^{\rm opt}(\alpha)$ for the three algorithms studied. Note that the curve for the GD case coincides with the curve for the critical κ, $\kappa_c(\alpha)$, given in Fig. 1. In all cases we find that to minimize the generalization error, for a given α, it is always best to train with $\kappa>0$, and slowly decrease κ as the size of the training set increases. It is interesting to observe that minimization of the generalization error for the perceptron and relaxation algorithms is achieved with nonzero

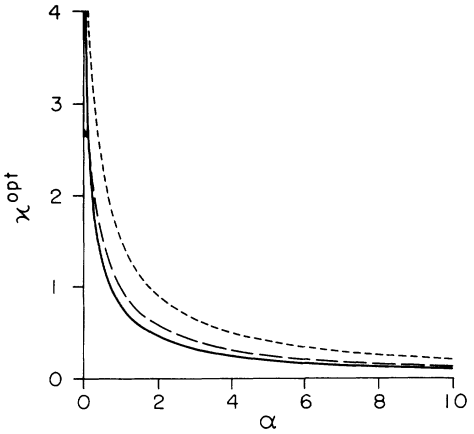

FIG. 8. Optimal value for κ vs α for the Gardner-Derrida (solid line), perceptron (long dashes), and relaxation (short dashes) learning algorithms.

training (and classification) errors, even in the absence of noise in the examples themselves. We believe that in this case the finiteness of the training set (i.e., the fact that α is finite) introduces an effective sampling noise. This result is similar to that of Györgi and Tishby [18], who found that optimal generalization for noisy examples is achieve by training at nonzero temperature, i.e., forcing the training error to be nonzero. We show here that this is the case even in the noise-free situation. In distinction to the results of Ref. [18], however, we find that as the size of the training set increases the classification error needed to achieve optimal performance decreases to zero, since the sampling noise vanishes for large α. We plot in Fig. 9 the optimal generalization curves for the three algorithms, taken at $\kappa=\kappa^{\rm opt}$, and again compare them to the Bayes algorithm. On the scale of the figure, the performance of the Bayes and relaxation algorithms and of the Gardner-Derrida and perceptron algorithms are indistinguishable. We note that in order to get a transition

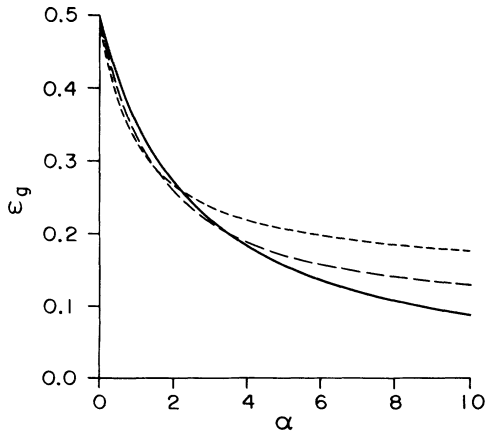

FIG. 7. Generalization error vs α in the annealed approximation for $\kappa=0$ (solid line), $\kappa=0.5$ (long dashes), and $\kappa=1$ (short dashes).

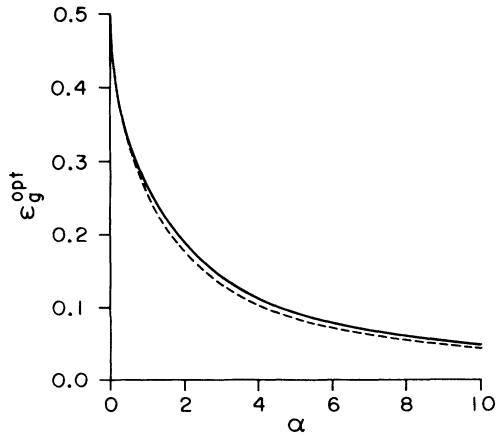

FIG. 9. Minimal generalization error vs α for the Gardner-Derrida and perceptron (solid line) and relaxation and Bayes (dashes) learning algorithms.

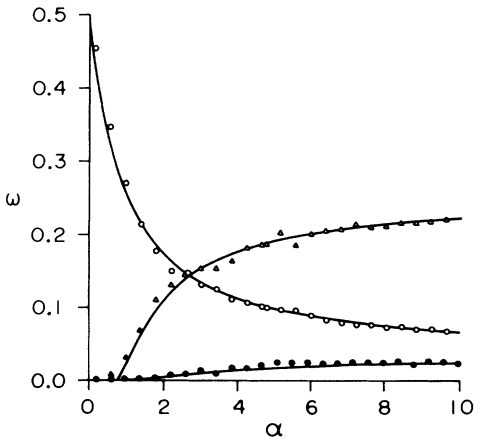

FIG. 10. Simulation results for training (△), generalization (○), and classification (●) errors for the relaxation algorithm with $\kappa=1$. The simulations were performed with $N=69$ and averaged over 150 cases. The lines are the predictions of the replica-symmetric theory.

from the $1/\sqrt{\alpha}$ behavior for $\kappa>0$ to the $1/\alpha$ behavior for $\kappa=0$, we must have $\kappa^{\rm opt}\propto 1/\alpha$ for large α. In their analysis of the Bayes algorithm Opper and Haussler [12] show that to get the optimal Bayesian performance requires the addition of a layer of hidden units with $H\to\infty$ hidden units. What we find here is that, provided we are able to determine $\kappa^{\rm opt}$, we can achieve a performance almost identical to Bayes, with a single-layer perceptron. We hasten to add that we are aware of the fact that the generalization error for the choice $\kappa^{\rm opt}$ still scales as $0.62/\alpha$ compared to the Bayesian $0.44/\alpha$. However, for any *finite* α we can choose κ so that the generalization error is very close to the Bayesian result. For example, in the range $0\le\alpha\le 10$ we find the two results never differ by more that 10^{-3}.

Finally, in order to demonstrate the validity of our results we compare the theoretical predictions for the training, classification, and generalization errors with computer simulations. In Fig. 10 we present simulation results for the relaxation algorithm with margin parameter $\kappa=1$. As can be seen in the figure, already for $N=69$ the results fit the theory very well. The increase in the classification error beyond α_c can be seen clearly in the simulation results. The expected asymptotic decay in ϵ_c occurs beyond the range of α investigated. Similar agreement has been observed for other values of the parameter κ.

V. CONCLUSION

We have focused in this paper on the analysis of three well-known learning algorithms for single-layer feedforward neural networks. While much work has been devoted to the convergence properties of these algorithms [15], we have studied their generalization performance as the size of the training set α increases. Interestingly we find that the algorithms which learn faster also generalize better. Our main results can be summarized as follows.

(i) The replica-symmetric theory has been shown to be stable for the perceptron and relaxation algorithms. It would thus be challenging to try to derive these results in a mathematically rigorous way (i.e, without using replicas).

(ii) The scaling behavior of the generalization errors in all three learning algorithms differ depending on whether the margin parameter κ is zero or not, or equivalently on whether the algorithm is consistent or not. In particular, for $\kappa=0$, $\epsilon_g\propto\alpha^{-1}$ while for $\kappa>0$, $\epsilon_g\propto\alpha^{-1/2}$.

(iii) The three learning algorithms studied yield identical behavior in the consistent case $\kappa=0$ at zero temperature. Thus the easily implementable perceptron and relaxation algorithms should be preferred. Obviously for nonzero temperature, the algorithms behave differently even for $\kappa=0$, since they give rise to distinct free energy functions. In the inconsistent case $\kappa>0$, the performance of the algorithms is markedly different, even at zero temperature, with the relaxation algorithm clearly outperforming the other two.

(iv) It is possible to choose the margin parameter κ so as to minimize the generalization error, in which case the relaxation algorithm is *almost* indistinguishable from the Bayes algorithm. The optimal value of κ is a decreasing function of the training set size α, approaching zero when $\alpha\to\infty$ with a power-law behavior α^{-1}. This result is important since the latter algorithm has been shown to be exactly implementable in a neural network setting only with $H\to\infty$ hidden units. We hasten to add that we do not claim, of course, that the two algorithms yield identical results.

(v) We find that for *finite* α, optimal generalization is achieved with a nonzero classification error of the training set, even though the examples are noise free. We believe this is due to the effective sampling noise introduced by the finiteness of α. As the number of examples increases, the minimal generalization error is obtained for vanishing classification error.

(vi) The annealed approximation has been shown to give qualitatively correct results for the generalization error in the consistent case $\kappa=0$. However, for $\kappa>0$ it fails to predict the transition to a state with nonzero training error, and thus yields grossly incorrect training errors, as well as the wrong asymptotic results for the generalization errors. Moreover, the latter results contradict the upper bound of Ref. [8].

We first observe that the asymptotic behavior $\epsilon_g\propto\alpha^{-1}$ of the generalization error in the consistent case (where the training error is zero for arbitrarily large α) agrees with the results derived from the VC-dimension analysis. As expected, the results fit neatly between the upper bound [2] and the lower bound [23] derived by VC theory for consistent learning algorithms. While the VC approach is very general and assumes very little about the learning process, the statistical approach we have followed has made very restrictive assumptions. In particular, we have (i) assumed a single-layer "network," (ii) restricted the probability distributions of the examples and of the target function to particularly simple forms. Moreover, and perhaps more importantly, the VC results are *worst* case results, while our results are average case

results. It would be interesting to see whether this kind of agreement still holds for multilayered problems, for which to the best of our knowledge, we at present have no average case results (although some results have been obtained for the random map problem in multilayered networks [24–26]).

Remarkably we also find our average case results in the inconsistent case $\kappa > 0$ to be in agreement with a recent VC theory analysis by Haussler [8]. The above author finds that the generalization error in this case is bounded above by a constant times $\alpha^{-1/2}$ which agrees with our average case results. It thus seems that the results obtained by the VC theory and the statistical physics approach agree both in the consistent as well as the inconsistent cases. Since we do not know of a lower bound in the inconsistent case, it is still possible that there are cases for which the generalization error decreases faster than $\alpha^{-1/2}$.

ACKNOWLEDGMENTS

The research of R.M. is supported by DARPA Contract No. F49620-90-0042 (DEF). J.F.F. is supported in part by Conselho de Desenvolvimento Científico e Tecnológico (CNPq).

APPENDIX A: THE REPLICA EQUATIONS

We give below the expressions for $\langle\langle Z^n \rangle\rangle$ within the full replica space, together with the expressions obtained within the replica symmetric framework. We assume the examples S_i are independent, identically distributed ± 1 random variables with $\text{Prob}(1)=\text{Prob}(-1)=\frac{1}{2}$. While the calculations are rather tedious, they have become pretty standard by now and will not be repeated.

$$\langle\langle Z^n \rangle\rangle = \int \prod_{\substack{\alpha,\beta \\ \alpha<\beta}} \frac{dq_{\alpha\beta} d\hat{q}_{\alpha\beta}}{2\pi i/N} \int \prod_{\alpha} \frac{dR_\alpha d\hat{R}_\alpha}{2\pi i/N} \exp\left[N \left[- \sum_{\substack{\alpha,\beta \\ \alpha<\beta}} q_{\alpha\beta}\hat{q}_{\alpha\beta} - \sum_\alpha R_\alpha \hat{R}_\alpha + G_0(\hat{q}_{\alpha\beta}, \hat{R}_\alpha) + \alpha G_1(q_{\alpha\beta}, R_\alpha) \right] \right] \tag{A1}$$

where

$$G_0 = \frac{1}{N} \ln \int \prod_{\alpha=1}^{n} d\mu(\mathbf{W}^\alpha) \exp\left[\sum_{\alpha<\beta} \hat{q}_{\alpha\beta} \mathbf{W}^\alpha \cdot \mathbf{W}^\beta + \sum_\alpha \hat{R}_\alpha \mathbf{W}^\alpha \cdot \mathbf{W}^0 \right] \tag{A2}$$

and

$$G_1 = \ln \int Dy \int \prod_\alpha \frac{dx_\alpha d\hat{x}_\alpha}{2\pi} \exp\left[-\beta f(y, x_\alpha) + i \sum_\alpha \hat{x}_\alpha (x_\alpha - y R_\alpha / \sqrt{M}) \right] \times \exp\left[- \sum_{\substack{\alpha,\beta \\ \alpha<\beta}} \hat{x}_\alpha \hat{x}_\beta (q_{\alpha\beta} - R_\alpha R_\beta / M) - \tfrac{1}{2} \sum_\alpha \hat{x}_\alpha^2 (1 - R_\alpha^2 / M) \right] \tag{A3}$$

with

$$M = \frac{1}{N} \sum_{i=1}^{N} (W_i^0)^2 . \tag{A4}$$

At this stage we make the assumption that the random variable M approaches a definite limit as $N \to \infty$. This assumption translates in fact into an assumption about the probability distribution of the teacher function. For example, assuming the components W_i^0 are independent, identically distributed random variables, one can use the law of large numbers [21] to prove that the fluctuations of M vanish in the large-N limit. However, the law of large numbers may hold under more general conditions.

The functions $f(y, x_\alpha)$ take on the following forms in the three cases studied:

$$f^{\text{GD}}(y, x_\alpha) = \Theta(\kappa - \text{sgn}(y) x_\alpha) , \tag{A5}$$

$$f^{P}(y, x_\alpha) = (\kappa - \text{sgn}(y) x_\alpha) \Theta(\kappa - \text{sgn}(y) x_\alpha) , \tag{A6}$$

$$f^{R}(y, x_\alpha) = (\kappa - \text{sgn}(y) x_\alpha)^2 \Theta(\kappa - \text{sgn}(y) x_\alpha) . \tag{A7}$$

Using the replica symmetric assumption [Eqs. (3.7) and (3.8)], we obtain the following expression for the free-energy density:

$$-\beta f = \frac{1}{2} \left[\frac{q - r^2}{1 - q} + \ln(1 - q) \right] + \alpha G_1(q, r) \tag{A8}$$

where we have defined $r = R/\sqrt{M}$ and

$$G_1^{\text{GD}} = 2 \int_{-\infty}^{\infty} Dt\, H(\xi_1) \ln[H(\xi_2) + e^{-\beta} H(-\xi_2)] , \tag{A9}$$

$$G_1^{P} = 2 \int_{-\infty}^{\infty} Dt\, H(\xi_1) \ln[H(\xi_2) + e^{\beta^2 (1-q)/2 - \beta(\kappa + \sqrt{q}\, t)} \times H(\sqrt{1-q} - \xi_2)] , \tag{A10}$$

$$G_1^{R} = 2 \int_{-\infty}^{\infty} Dt\, H(\xi_1) \times \ln\left[H(\xi_2) + \frac{e^{-\beta(\kappa + \sqrt{q}\, t)^2 / [1 + 2\beta(1-q)]}}{\sqrt{1 + 2\beta(1-q)}} \times H\left[- \frac{\xi_2}{\sqrt{1 + 2\beta(1-1)}} \right] \right] , \tag{A11}$$

with ξ_1 and ξ_2 being given by Eqs. (3.12) and (3.13).

APPENDIX B: CALCULATING THE CLASSIFICATION ERROR

In order to calculate the average classification error defined by (2.19) we use the following identities:

$$\epsilon_c = P^{-1}\langle\langle\ \langle E_c(W)\rangle_T \rangle\rangle$$
$$= P^{-1}\left\langle\!\!\left\langle Z^{-1}\int d\mu(\mathbf{W})E_c(\mathbf{W})e^{-\beta E(\mathbf{W})}\right\rangle\!\!\right\rangle \tag{B1}$$

where

$$Z = \int d\mu(\mathbf{W})e^{-\beta E(\mathbf{W})} . \tag{B2}$$

Multiplying numerator and denominator by Z^{n-1} we get

$$\epsilon_c = P^{-1}\left\langle\!\!\left\langle Z^{-n}\int \prod_{\alpha=1}^{n} d\mu(\mathbf{W}^\alpha)E_c(\mathbf{W}^1)\right.\right.$$
$$\left.\left.\times\exp\left[-\beta\sum_\alpha E(\mathbf{W}^a)\right]\right\rangle\!\!\right\rangle . \tag{B3}$$

In the limit $n\to 0$, $Z^{-n}\to 1$, and thus

$$\epsilon_c = P^{-1}\lim_{n\to 0}\left\langle\!\!\left\langle \int \prod_{\alpha=1}^{n} d\mu(\mathbf{W}^\alpha)E_c(\mathbf{W}^1)\right.\right.$$
$$\left.\left.\times\exp\left[-\beta\sum_\alpha E(\mathbf{W}^\alpha)\right]\right\rangle\!\!\right\rangle . \tag{B4}$$

At this stage we may go ahead with the usual replica manipulations, finally obtaining within the replica symmetric theory

$$\epsilon_c = \int Dt\int Dy\frac{\int Dv\,\Theta(-yz)e^{-\beta f(y,z)}}{\int Dv\,e^{-\beta f(y,z)}} \tag{B5}$$

where

$$z = v\sqrt{1-q}+yr-t\sqrt{q-r^2} , \tag{B6}$$

and $f(y,z)$ are given by Eqs. (A5)–(A7) depending on the error function used (and replacing x_α by z).

It is not difficult to see that in the region $\alpha<\alpha_c$ where $q<1$, one obtains $\epsilon_c=0$ by setting $\beta\to\infty$. This result is expected since we know that $\epsilon_c\le\epsilon_t$ and $\epsilon_t=0$ in this region. For $\alpha>\alpha_c$ one must make the limit carefully, bearing in mind that $x=\beta(1-q)$ is finite. After some algebra we get the following simple expressions for the perceptron and relaxation classification errors:

$$\epsilon_c^P = 2\int_0^\infty Dt\,H\left[\frac{rt+x}{\sqrt{1-r^2}}\right] , \tag{B7}$$

$$\epsilon_c^R = \frac{2}{\sqrt{1+2x}}\int_0^\infty Dt\,H\left[\frac{rt+2\kappa x}{\sqrt{1-r^2}}\right] . \tag{B8}$$

It is thus clear that the average classification error ϵ_c is nonzero above α_c. In fact, it displays a rather interesting behavior, increasing from zero at α_c until reaching a maximum, after which it slowly decrease to zero. All this time it remains bounded above by both ϵ_t and ϵ_g decaying asymptotically as $\alpha^{-1/2}$. As mentioned in the text, the fact that the classification error is nonzero proves our claim that for $\kappa>0$ the algorithms are inconsistent.

[1] V. N. Vapnik and A. Y. Chervonenkis, Theor. Prob. Appl. **16**, 264 (1971).
[2] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, J. Assoc. Comput. Mach. **36**, 929 (1989).
[3] E. Gardner, J. Phys. A **21**, 257 (1988).
[4] E. Gardner and B. Derrida, J. Phys. A **21**, 271 (1988).
[5] S. Seung, H. Sompolinsky, and N. Tishby, Phys. Rev. A (to be published).
[6] R. Meir and J. F. Fontanari, J. Phys. A **25**, 1149 (1992).
[7] D. Haussler, M. Kearns, and R. Schapire, in *Computational Learning Theory: Proceedings of the Fourth Annual Workshop,* edited by L. Valiant and M. Warmuth (Kaufmann, San Mateo, CA, 1991).
[8] D. Haussler, University of California Technical Report No. UCSC-CRL-91-02 (unpublished).
[9] H. Sompolinsky, N. Tishby, and H. S. Seung, Phys. Rev. Lett. **65**, 1683 (1990).
[10] F. Rosenblatt, *Principles of Neurodynamics* (Spartan Books, New York, 1962).
[11] S. Agmon, Can. J. Math. **6**, 382 (1954).
[12] M. Opper and D. Haussler, Phys. Rev. Lett. **66**, 2677 (1991).
[13] M. Griniasti and H. Gutfreund, J. Phys. A **24**, 715 (1991).
[14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Science **220**, 671 (1983).
[15] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973).
[16] C. H. Mays, IEEE Trans. Electron Commput. **EC-13**, 465 (1964).
[17] J. K. Anlauf and M. Biehl, Eurphys. Lett. **10**, 687 (1989).
[18] 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.
[19] J. L. van Hemmen and R. G Palmer, J. Phys. A **12**, 563 (1979).
[20] N. G. de Bruijn, *Asymptotic Methods in Analysis* (Dover, New York, 1981).
[21] W. Feller, *An Introduction to Probability Theory and its Applications* (Wiley, New York, 1957).
[22] J. R. L. de Almeida and D. J. Thouless, J. Phys. **11**, 983 (1978).
[23] A. Ehrenfreucht, D. Haussler, M. Kearns, and L. Valiant, in *Computational Learning Theory: Proceedings of the First Annual Workshop,* edited by D. Haussler and L. Pitt (Kaufmann, San Mateo, CA, 1988).
[24] E. Barkai, D. Hansel, and I. Kanter, Phys. Rev. Lett. **65**, 2312 (1990).
[25] E. Barkai and I. Kanter, Europhys. Lett. **14**, 107 (1991).
[26] M. Griniasti and T. Grossman (unpublished).