# Storage capacity and learning algorithms for two-layer neural networks

A. Engel, H. M. Köhler, F. Tschepke, H. Vollmayr, and A. Zippelius
*Institut für Theoretische Physik, Universität Göttingen, W-3400 Göttingen, Germany*
(Received 23 September 1991)

A two-layer feedforward network of McCulloch-Pitts neurons with $N$ inputs and $K$ hidden units is analyzed for $N \to \infty$ and $K$ finite with respect to its ability to implement $p = \alpha N$ random input-output relations. Special emphasis is put on the case where all hidden units are coupled to the output with the same strength (committee machine) and the receptive fields of the hidden units either enclose all input units (fully connected) or are nonoverlapping (tree structure). The storage capacity is determined generalizing Gardner's treatment [J. Phys. A **21**, 257 (1988); Europhys. Lett. **4**, 481 (1987)] of the single-layer perceptron. For the treelike architecture, a replica-symmetric calculation yields $\alpha_c \propto \sqrt{K}$ for a large number $K$ of hidden units. This result violates an upper bound derived by Mitchison and Durbin [Biol. Cybern. **60**, 345 (1989)]. One-step replica-symmetry breaking gives lower values of $\alpha_c$. In the fully connected committee machine there are in general correlations among different hidden units. As the limit of capacity is approached, the hidden units are anticorrelated: One hidden unit attempts to learn those patterns which have not been learned by the others. These correlations decrease as $1/K$, so that for $K \to \infty$ the capacity per synapse is the same as for the tree architecture, whereas for small $K$ we find a considerable enhancement for the storage per synapse. Numerical simulations were performed to explicitly construct solutions for the tree as well as the fully connected architecture. A learning algorithm is suggested. It is based on the least-action algorithm, which is modified to take advantage of the two-layer structure. The numerical simulations yield capacities $p$ that are slightly more than twice the number of degrees of freedom, while the fully connected net can store relatively more patterns than the tree. Various generalizations are discussed. Variable weights from hidden to output give the same results for the storage capacity as does the committee machine, as long as $K = O(1)$. We furthermore show that thresholds at the hidden units or the output unit cannot increase the capacity, as long as random unbiased patterns are considered. Finally we indicate how to generalize our results to other Boolean functions.

PACS number(s): 87.10.+e, 64.60.Cn

## I. INTRODUCTION

The methods of statistical mechanics have been widely used for a quantitative analysis of networks of formal neurons [1]. Among all possible architectures the feedforward layered systems play a central role [2]: they have great computational abilities and at the same time are rather simple, because there is no feedback. The problem of learning in such a network is defined as follows. Given a set of $p$ input-output relations $\{\xi^\mu, \eta^\mu\} (\mu = 1, \ldots, p)$, can one construct a network which produces the correct output $\eta^\mu$ for each input $\xi^\mu$?

The simplest feedforward net is the single-layer perceptron without hidden units. This system is well understood. Geometrical arguments [3] as well as statistical mechanics [4] can be used to determine its storage capacity, i.e., the maximal number of random input-output relations that can be implemented. Furthermore, a learning algorithm is known that finds a solution, if one exists [5].

Much less is known about multilayer perceptrons with one or more layers of hidden units. Whereas single-layer perceptrons can solve only a very limited class of problems [6], any Boolean function of $N$ inputs can be implemented, if one allows for an intermediate layer with sufficiently many hidden neurons [7]. Hence these systems are of great technical importance and have been used extensively in applications of neural nets to real problems. However their theoretical storage capacity is not known in general. Upper bounds for the storage capacity have been derived for some multilayer networks [8,9]. A detailed analysis using methods of statistical mechanics has been performed for the parity machine [10,11]. The main result of Ref. [10] is a storage capacity per synapse, which increases with the logarithm of the number of hidden units, in agreement with the upper bound of Mitchison and Durbin [9].

Several learning algorithms have been suggested to train multilayer networks. The best known one is "backpropagation" [2] which has been used in many applications. In other approaches [12–14] the network is constructed while learning: hidden units or layers of hidden units are added to the network, until the desired mapping is achieved. Convergence can be guaranteed, in contrast to existing algorithms for a *fixed* architecture.

It would also be interesting to consider recurrent neural nets with hidden units. One question to ask is the following: How many patterns can be stabilized in a neural network with attractor dynamics, if a certain fraction of sites is left free to adjust? In the Hopfield model it is known that a macroscopic number of patterns can be stabilized, even though the stability condition is violated at a finite fraction of sites (corresponding to thermalized

 

hidden units) [15].

Here we consider a multilayer perceptron with a single-layer of hidden units (Fig. 1). All neurons of the network are linear threshold units. In general the synaptic weights from the input to hidden and hidden to output layers are free to adapt in the process of learning a given set of $p$ input-output relations. Due to the symmetry of the model all weights from the hidden layer to the output unit can be taken as *positive* without loss of generality. This has led us to consider in detail the so-called committee or consensus machine, where the synapses from the hidden layer to the output unit are all taken to be *equal*.

Such an architecture has important applications in various contexts. For example, Opper and Haussler [16] use a committee machine with a large number of hidden units to implement Bayes's optimal classification algorithm. Sompolinsky and Tishby [17] analyze a closely related architecture in the context of learning a rule—counting domains in one-dimensional patterns.

In this paper we present results of a phase-space analysis to estimate the critical storage capacity $\alpha_c = p_c/N$. For the treelike architecture, a replica-symmetric calculation yields $\alpha_C(K=3) \simeq 4.02$ and $\alpha_c \sim \sqrt{K}$ for a large number $K$ of hidden units. As compared to the parity machine [10], the capacity is strongly reduced, nevertheless it still violates the upper bound of Ref. [9]. For this reason and also because of the structure of solution space, we are led to consider replica-symmetry breaking. In one step we find a reduction in capacity as compared to the replica-symmetric result, for $K=3$ $\alpha_c$ is reduced to $\sim 3.02$. In the fully connected committee machine, there are in general correlations among different hidden units. As the limit of capacity is approached, the hidden units are anticorrelated. One hidden unit attempts to learn those patterns, which have not been learned by the others. These correlations decrease as $1/K$ so that for $K \to \infty$ the capacity per synapse is the same as for the tree architecture [18]. For small $K$ the capacity per synapse is considerably enhanced; for $K=3$ we find $\alpha_c \sim 34.5$ as compared to $\alpha_c^{\text{tree}}(K=3) \times 3 \sim 12$. Numerical simulations were performed to explicitly construct solutions for the tree as well as the fully connected architecture. A learning algorithm is suggested. It is based on the least-action algorithm [9], which is modified to take advantage of the two-layer structure. The numerical simulations yield capacities $p_c$ that are twice the number of degrees of freedom. For the fully connected net, correlations between different hidden units are shown to decrease as $1/K$ as suggested by a simple argument and in agreement with our theoretical analysis. Various generalizations are discussed. Variable weights from the hidden units to the output unit give the same result for the storage capacity as the committee machine gives as long as $K$ remains finite. We indicate how to generalize our results to other Boolean functions and show that thresholds cannot increase the capacity, as long as random unbiased patterns are considered.

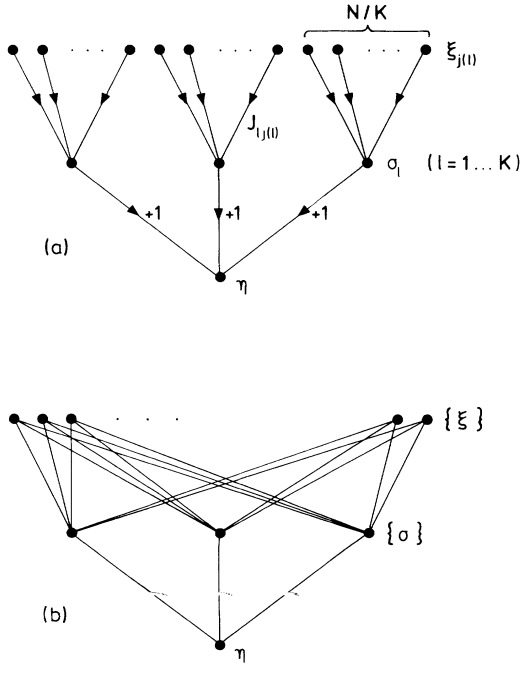


FIG. 1. Committee machine with tree connectivity (a) and fully connected (b).

## II. MODEL

The network under consideration consists of an input layer of $N$ neurons $\xi_i$ $(i=1,\ldots,N)$, a hidden layer of $K$ binary neurons $\sigma_l (l=1,\ldots,K)$, and one output unit $\eta$. The system operates as a feedforward net: synaptic connections from input neuron $\xi_i$ to hidden unit $\sigma_l$ are denoted by $J_{li}$ and synaptic connections from hidden unit $\sigma_l$ to the output unit by $w_l$. We consider real valued variables $\{J_{lj}, w_l\}$ with spherical normalization. Each hidden unit $\sigma_l$ calculates the weighted sum of all its inputs $\{j(l)\}$ and compares it to a threshold $\Theta_l$:

$$\sigma_l = \operatorname{sgn}\left[\sum_{j(l)} J_{lj(l)}\xi_{j(l)} - \Theta_l\right]. \tag{1}$$

Similarly the output unit calculates the weighted sum of all its inputs from the hidden layer and compares it to a threshold $\Theta$:

$$\eta = \operatorname{sgn}\left[\sum_l w_l\sigma_l - \Theta\right]. \tag{2}$$

We assume that there are no direct couplings from input to output. Two architectures will be discussed in detail.

(a) Tree connectivity. The input units are organized in $K$ groups each containing $N/K$ neurons [Fig. 1(a)]. Each hidden unit receives input from its group only, i.e., $j(l)=(l-1)(N/K)+1,\ldots,l(N/K)$. The appropriate normalization of synaptic connections is $\Sigma_{j(l)} J_{lj(l)}^2 = N/K$. The same architecture has been studied for the parity machine [10]. It has the advantage that there are no correlations among different hidden units, because they have no input in common.

(b) Full connectivity. Each hidden unit is connected to each input unit [Fig. 1(b)]. In that case $j(l)=1,\ldots,N$ is independent of $l$ and the appropriate normalization is

$\Sigma_{j(l)} J_{lj(l)}^2 = N$.

For simplicity we take zero thresholds $\Theta_l = \Theta = 0$. For this particular choice, the model has the following symmetry: if a solution $\{w_l, J_{lj}\}$ has been found, then another solution can be constructed by inverting all couplings of one hidden unit, i.e., $-w_{l_0}$, $\{-J_{l_0 j(l_0)}\}$. Without loss of generality we restrict ourselves to positive $w_l$. Moreover we use *fixed* values for the $w_l$ and only adapt the connections $J_{lj}$ between input and hidden units during the learning process. We choose $w_l = 1$, $l = 1, \ldots, K$. Such a network is called a committee machine since the output adjusts to the majority of the hidden units. In Sec. VIII we discuss other fixed Boolean functions between hidden units and output and consider also the effect of variable $w_l$, as well as nonzero thresholds.

The task is to learn $p$ given input-output relations $\{\xi_j^\mu, \eta^\mu\}$ $(\mu = 1, \ldots, p)$. The $\xi_j^\mu$ and $\eta^\mu$ are independent stochastic variables. In most of our discussion we shall focus on binary variables with equal probability for $\pm 1$. In the numerical simulations we shall also consider Gaussian $\{\xi_j^\mu\}$. Two questions will be addressed.

(i) How many relations can be learned for the number $N$ of input sites tending to infinity and a *fixed* number of hidden units $K$?

(ii) How can the couplings $\{J_{lj}\}$ and $\{w_l\}$ be found, which realize a given set of $p$ input-output relations for fixed $N$ and $K$?

The first question will be analyzed in Secs. III and IV by doing statistical mechanics in the phase space of the synaptic interactions $J_{lj(l)}$. In Secs. III and IV we calculate the storage capacity for the tree architecture and in Sec. V for the fully connected net. The second question is discussed in Secs. VI and VII, where different learning algorithms are introduced.

## III. REPLICA-SYMMETRIC THEORY

In this section we determine within the replica-symmetric approximation the typical fractional volume $V_{\rm typ}$ in the space of interactions $\{J_{lj(l)}\}$ of synaptic matrices that can realize $p$ random input-output mappings $\{\xi_{lj(l)}^\mu \to \eta^\mu, \mu = 1, \ldots, p\}$. We are interested in the case $p = \alpha N$ and $N \to \infty$. The method is a straightforward extension of Gardner's analysis of the single-layer perceptron [4]. For given random input patterns $\{\xi_{lj(l)}^\mu\}$ and the corresponding outputs $\eta^\mu$ the quantity

$$V = \frac{M}{D} , \tag{3}$$

where

$$M = \int \prod_{l=1}^{K} \prod_{j(l)} dJ_{lj(l)} \prod_{l=1}^{K} \delta \left[ \sum_{j(l)} J_{lj(l)}^2 - \frac{N}{K} \right] \prod_{\mu} \Theta \left[ \eta^\mu \sum_{l=1}^{K} \mathrm{sgn} \left[ \left[ \frac{K}{N} \right]^{1/2} \sum_{j(l)} J_{lj(l)} \xi_{j(l)}^\mu \right] \right] \tag{4}$$

and

$$D = \int \prod_{l=1}^{K} \prod_{j(l)} dJ_{lj(l)} \prod_{l=1}^{K} \delta \left[ \sum_{j(l)} J_{lj(l)}^2 - \frac{N}{K} \right] \tag{5}$$

gives the fraction of synaptic matrices $\{J_{lj(l)}\}$ which satisfy the constraints

$$\sum_{j(l)} J_{lj(l)}^2 = \frac{N}{K}, \quad l = 1, \ldots, K \tag{6}$$

and implement the desired mapping $\{\xi_i^\mu\} \to \eta^\mu$ for all $\mu = 1, \ldots, p$. Here $\Theta(x)$ denotes the Heaviside function. $\xi_i^\mu$ and $\eta^\mu$ are independent, identically distributed random variables with distribution

$$P(\xi_i^\mu) = \tfrac{1}{2}\delta(\xi_i^\mu - 1) + \tfrac{1}{2}\delta(\xi_i^\mu + 1) . \tag{7}$$

Hence the integrand of $M$ is a product of almost independent random factors and the typical volume $V_{\rm typ}$ is given by

$$V_{\rm typ} \simeq \exp\{\langle\langle \ln V \rangle\rangle\} , \tag{8}$$

where $\langle\langle\ \rangle\rangle$ denotes the average with the distribution (7). Note that the transformation $\xi^\mu \to \eta^\mu \xi^\mu$ leaves the statistical properties of the input-output ensemble invariant so that in calculating $\langle\langle \ln V \rangle\rangle$ the $\eta^\mu$ in (4) can be omitted. The average of the logarithm of $V$ is performed using the replica trick, i.e., by introducing replica indices $\alpha = 1, \ldots, n$ for the synaptic couplings $J_{lj(l)}$ in order to calculate $\langle\langle V^n \rangle\rangle$, and then taking the limit $n \to 0$. Introducing the auxiliary variables

$$\lambda_{l\mu}^\alpha = \left[ \frac{K}{N} \right]^{1/2} \sum_{j(l)} J_{lj(l)}^\alpha \xi_{j(l)}^\mu$$

for the local fields at the hidden units and their conjugate Lagrange multipliers $x_{l\mu}^\alpha$, the average over the patterns can be performed and after standard manipulations [4] one finds to leading order in $N$

$$\langle\langle V^n \rangle\rangle = \int \prod_{l,\alpha} \frac{dE_l^\alpha}{4\pi} \prod_{\substack{l,\alpha,\beta \\ \alpha<\beta}} \frac{dF_l^{\alpha\beta} dq_l^{\alpha\beta}}{2\pi K/N} \exp \left[ N \left[ \frac{1}{2K} \sum_{l,\alpha} E_l^\alpha - \frac{1}{2K} \sum_{\substack{l,\alpha,\beta \\ \alpha<\beta}} F_l^{\alpha\beta} q_l^{\alpha\beta} + \frac{1}{K} G_2(E_l^\alpha, F_l^{\alpha\beta}) + \alpha G_1(q_l^{\alpha\beta}) - \frac{n}{2}[1 + \ln(2\pi)] \right] \right] , \tag{9}$$

where

$$G_2(E_l^\alpha,F_l^{\alpha\beta})=\ln\int\prod_{l,\alpha}dJ_l^\alpha\exp\left[-\tfrac{1}{2}\sum_{l,\alpha}E_l^\alpha(J_l^\alpha)^2+\sum_{\substack{l,\alpha,\beta\\ \alpha<\beta}}F_l^{\alpha\beta}J_l^\alpha J_l^\beta\right] \tag{10}$$

and

$$G_1(q_l^{\alpha\beta})=\ln\int\prod_{l,\alpha}\frac{d\lambda_l^\alpha dx_l^\alpha}{2\pi}\exp\left[i\sum_{l,\alpha}x_l^\alpha\lambda_l^\alpha-\tfrac{1}{2}\sum_{l,\alpha}(x_l^\alpha)^2-\tfrac{1}{2}\sum_{\substack{l,\alpha,\beta\\ \alpha\neq\beta}}x_l^\alpha x_l^\beta q_l^{\alpha\beta}\right]\prod_\alpha\Theta\left[\sum_l\operatorname{sgn}\lambda_l^\alpha\right] . \tag{11}$$

As usual we have introduced the order parameters

$$q_l^{\alpha\beta}=\frac{K}{N}\sum_{j(l)}J_{lj(l)}^\alpha J_{j(l)}^\beta , \tag{12}$$

characterizing the overlap between two synaptic matrices implementing the desired input-output mapping. $E_l^\alpha$ and $F_l^{\alpha\beta}$ are Lagrange multipliers associated with the constraints (6) and (12). The last term in (9) comes from the denominator (5).

The integrals in (9) are dominated by their saddle-point values. In this section we study the ansatz that the values of the order parameters at the saddle point are independent of the replica indices, i.e., we look for a saddle point of the form

$$\begin{aligned} q_l^{\alpha\beta}&=q_l=q ,\\ F_l^{\alpha\beta}&=F_l=F ,\\ E_l^\alpha&=E_l=E . \end{aligned} \tag{13}$$

These quantities are independent of $l$, because the hidden units are equivalent to each other after averaging over the patterns. We do not expect a spontaneous breakdown of this "translational invariance."

With the help of (13) Eqs. (9)–(11) simplify considerably. The saddle-point equations for $E$ and $F$ become algebraic as in the case of the single-layer perceptron [4] and therefore these order parameters can be eliminated. In order to simplify the expression for $G_1$ it is convenient to split the integration variables $\lambda_l$ into their sign and their absolute value:

$$\int_{-\infty}^{\infty}\prod_l d\lambda_l f(\lambda_l)=\operatorname*{Tr}_{\{\tau_l=\pm1\}}\int_0^\infty\prod_l d\lambda_l f(\lambda_l\tau_l) . \tag{14}$$

Taking finally the limit $n\to0$ we get

$$\frac{1}{N}\langle\langle\ln V\rangle\rangle=\operatorname*{extr}_q\left\{\tfrac{1}{2}\ln(1-q)+\frac{q}{2(1-q)}+\alpha\int\prod_l Dt_l\ln\left[\operatorname*{Tr}_{\{\tau_l=\pm1\}}\Theta\left[\sum_l\tau_l\right]\prod_l H(Qt_l\tau_l)\right]\right\} \tag{15}$$

with $Q=[q/(1-q)]^{1/2}$. As usual we have used the abbreviations

$$\int Dt\cdots=\int_{-\infty}^{\infty}\frac{dt}{\sqrt{2\pi}}\exp\left[-\frac{t^2}{2}\right]\cdots$$

and

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

Equation (15) has a simple physical meaning. $\operatorname{Tr}_{\{\tau_l=\pm1\}}\Theta(\Sigma_l\tau_l)$ is the trace over all internal representations $\{\tau_l\}$ that produce the desired output $+1$ and $\prod_l H(Qt_l\tau_l)$ is the product of Gardner volumes of the subperceptrons for a given internal representation.

From (15) we get the maximal storage capacity $\alpha_c$ by taking the limit $q\to1$. In this limit different solutions for the synaptic matrix become highly correlated and the typical fractional volume $V_{\text{typ}}$ shrinks to zero. Quantitatively $\alpha_c$ can be determined from the limit $q\to1$ of the saddle-point equation corresponding to (15). It is, however, simpler to rewrite (15) as

$$\frac{1}{N}\langle\langle\ln V\rangle\rangle=\operatorname*{extr}_q\frac{1}{2(1-q)}\left\{(1-q)\ln(1-q)+q+2\alpha(1-q)\int\prod_l Dt_l\ln\left[\operatorname*{Tr}_{\{\tau_l=\pm1\}}\Theta\left[\sum_l\tau_l\right]\prod_l H(Qt_l\tau_l)\right]\right\} \tag{16}$$

and to observe that in order to have a well-defined extremum for $q\to1$, the expression in the outer bracket should go to zero in this limit. This yields immediately an equation for $\alpha_c^{\text{RS}}$:

$$(\alpha_c^{\text{RS}})^{-1}=-\lim_{q\to1}2(1-q)\int\prod_l Dt_l\ln\left[\operatorname*{Tr}_{\{\tau_l=\pm1\}}\Theta\left[\sum_l\tau_l\right]\prod_l H(Qt_l\tau_l)\right] . \tag{17}$$

Taking the limit $q \to 1$ now requires finding the dominating terms in the trace over $\tau_l$. This is done in Appendix A and gives finally

$$(\alpha_c^{\mathrm{RS}})^{-1} = K \int_0^\infty Dt\, t^2 g(t) \tag{18}$$

with

$$g(t) = \sum_{k=0}^{(K-1)/2} \binom{K-1}{k} [H(t)]^{K-1-k} [1-H(t)]^k . \tag{19}$$

A qualitative interpretation of this formula is given in Sec. VIII together with a discussion of analogous results for networks with other Boolean functions between hidden units and output. A numerical analysis of (18) gives $\alpha_c^{\mathrm{RS}} = 2$, 4.02, 5.78, 7.30, and 8.70 for $K = 1$, 3, 5, 7, and 9, respectively. For large $K$ the binomial distribution in (19) can be approximated by a Gaussian one and one finds for the asymptotic dependence of $\alpha_c$

$$\alpha_c^{\mathrm{RS}}(K) \sim 6 \left[ \frac{2}{\pi} \right]^{1/2} K^{1/2} .$$

In Fig. 2 we have plotted $\alpha_c^{\mathrm{RS}}$ for $K = 1$ to 59 together with this asymptotic behavior as a function of $K^{1/2}$.

The asymptotic behavior violates the upper bound $\alpha_c(K) \sim \ln(K)$ obtained by Mitchison and Durbin [9]. This is a strong indication for replica-symmetry breaking and has led us to investigate a saddle point with broken replica symmetry.

To finish the investigation of the replica-symmetric theory we note that the order parameter $q$ is monotonously increasing with increasing $\alpha$. For $\alpha \to 0$ we find

$$q \sim \frac{2K}{\pi} 2^{-K} \binom{K-1}{\frac{K-1}{2}}^2 \alpha . \tag{20}$$

Figure 3 shows a plot of $q(\alpha)$ for $K = 3$ as obtained by numerically determining the extremum in Eq. (16). Unlike in the case of the parity machine where global symmetries make $q = 0$ for a finite interval $0 \le \alpha \le \alpha_0$ [10], for the committee machine one has $q > 0$ for all $\alpha > 0$.

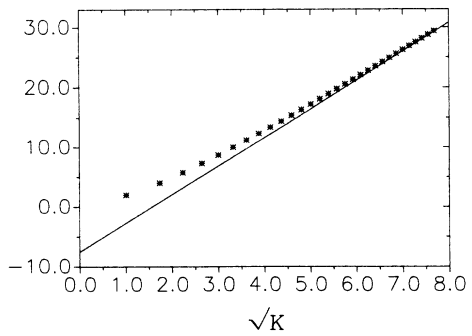


FIG. 2. Storage capacity $\alpha^{\mathrm{RS}}$ vs $\sqrt{K}$ for the tree architecture with $K$ hidden units in replica-symmetric approximation. The straight line shows the asymptotic $\sqrt{K}$ increase of $\alpha_c^{\mathrm{RS}}$.

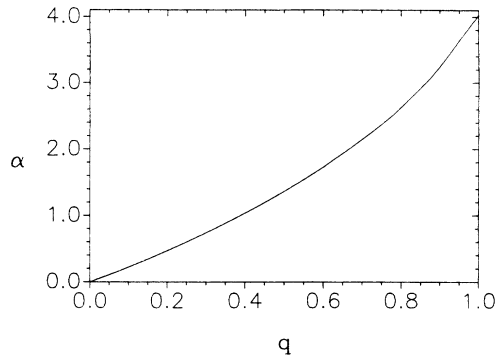


FIG. 3. $\alpha(q)$ for the tree with $K = 3$ in replica-symmetric approximation.

## IV. ONE-STEP REPLICA-SYMMETRY BREAKING

The asymptotic behavior of the storage capacity $\alpha_c^{\mathrm{RS}} \sim K^{1/2}$ for large numbers $K$ of hidden units as obtained within the replica-symmetric theory violates the upper bound $\alpha_c \sim \ln(K)$ derived from geometrical considerations similar to Cover's analysis of the single-layer perceptron [3,9]. It is therefore necessary to see how this result is modified when the symmetry between replicas is broken. According to common wisdom replica symmetry corresponds in the present context to a connected solution space, whereas replica-symmetry breaking is to be expected if the solution space consists of several disconnected parts [19]. For the single-layer perceptron with continuous weights the solution space is known to be connected (even convex) and hence replica symmetry holds. In hidden-unit networks the possibility to encounter replica-symmetry breaking is much larger due to the existence of different internal representations $\{\sigma_l\}$ of the patterns.

Let us assume first that there is a synaptic matrix $J^*_{lj(l)}$ which realizes the internal representation $\sigma_l^\mu = +1$, $l = 1, \ldots, K$ for all patterns $\mu$. It is easy then to show that the solution space is starlike with respect to $J^*_{lj(l)}$ [Fig. 4(a)]. In fact let $J^{(1)}_{lj(l)}$ be any other solution which implements the desired input-output relation, i.e., which makes more $\sigma_l^\mu = +1$ than $\sigma_l^\mu = -1$ for all $\mu$. Then

$$J_{lj(l)} = \gamma J^*_{lj(l)} + (1-\gamma) J^{(1)}_{lj(l)} \tag{21}$$

has for all $\gamma$ with $0 \le \gamma \le 1$ also more $\sigma_l^\mu = +1$ than $\sigma_l^\mu = -1$ for all $\mu$ and is hence a solution. Therefore the solution space is connected (but probably not convex) if the special solution $J^*_{lj(l)}$ exists. Via $J^*_{lj(l)}$ any solution $J^{(1)}_{lj(l)}$ can be continuously deformed into any other solution $J^{(2)}_{lj(l)}$ without leaving the solution space. We can actually determine the value of $\alpha$ for which $J^*_{lj(l)}$ exists. In order to have $\sigma_l^\mu = +1$ for all $l$ and $\mu$ we have to teach the $K$ subperceptrons of our machine $\alpha N$ patterns $\{\xi_j^\mu\}$. Each subperceptron has $N/K$ synapses and using Gardner's results for the single-layer perceptron we find that $J^*_{lj(l)}$ exists for $\alpha \le 2/K$. Hence for $\alpha \le 2/K$ the solution space is connected and the results of the replica-symmetric solution should be correct.

It is difficult to characterize the solution space for $\alpha > 2/K$, mainly because then two different solutions $J_{lj(l)}^{(1)}$ and $J_{lj(l)}^{(2)}$ differ by their internal representation for very many patterns. Nevertheless one expects that for some $\bar{\alpha}$ with $2/K < \bar{\alpha} < \alpha_c^{RS}$ the solution space breaks into several disconnected parts [Fig. 4(b)]. Then the replica-symmetric value for $\alpha_c$ is clearly wrong.

In the following we study the implications of the first step of Parisi's hierarchical replica-symmetry-breaking scheme to our problem [20]. The saddle point in (9) is then parametrized by the six order parameters $E$, $F_1$, $F_2$, $q_1$, $q_2$, and $m$, where $m$ denotes the break point of the order parameter function. Again $E$, $F_1$, and $F_2$ can be eliminated with the help of their self-consistent equations, and similar to related problems [21, 10] we get

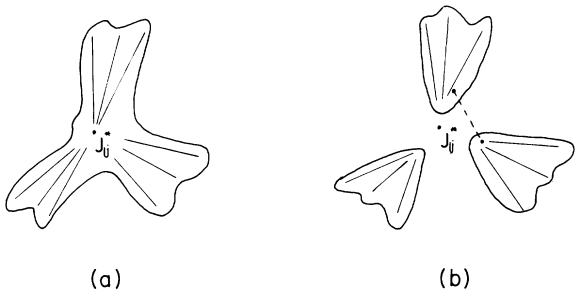


FIG. 4. Sketch of the solution space for $K=3$. (a) $\alpha < \frac{2}{3}$, the solution space is starlike with respect to $J_{lj}^{*}$. (b) $\bar{\alpha} < \alpha$, the solution space breaks into disconnected parts. The dotted line connects two solutions with different internal representations.

$$\frac{1}{N}\langle\langle V\rangle\rangle = \underset{q_0,q_1,m}{\text{extr}} \left\{ \frac{q_0}{2(1-q_1+m\Delta q)} + \frac{m-1}{2m}\ln(1-q_1) + \frac{1}{2m}\ln(1-q_1+m\Delta q) \right.$$
$$\left. + \frac{\alpha}{m}\int \prod_l Dz_l \ln \int \prod_l Dt_l \left[ \underset{\{\tau_l=\pm 1\}}{\text{Tr}} \Theta\left( \sum_l \tau_l \right) \prod_l H\left[ \frac{\tau_l}{(1-q_1)^{1/2}} [q_0^{1/2} z_l + (\Delta q)^{1/2} t_l] \right] \right]^m \right\}, \tag{22}$$

where $\Delta q = q_1 - q_0$. In order to determine $\alpha_c$ from this expression we would have to derive the three saddle-point equations and to study the limit $q_1 \to 1$. This is a rather complicated program. Instead we try again to determine $\alpha_c$ directly from (22) using arguments similar to those that carried us from (15) to (17). First one expects from the structure of the integral in (22) that with $q_1 \to 1$ we have $m \to 0$ so that $c = m/(1-q_1)$ remains finite. This scaling of $m$ is well known from a variety of related systems [20, 22, 10]. Replacing $(1-q_1)$ by $m/c$ in (22) we get

$$\frac{1}{N}\langle\langle \ln V\rangle\rangle$$
$$= \underset{q_0,m,c}{\text{extr}} \frac{1}{2m} \left\{ \frac{cq_0}{1-m+c(1-q_0)} + \ln[1-m+c(1-q_0)] \right.$$
$$+ m\ln\left[\frac{m}{c}\right] + 2\alpha \int \prod_l Dz_l \ln \int \prod_l Dt_l$$
$$\left. \times \left[ \underset{\{\tau_l=\pm 1\}}{\text{Tr}} \Theta\left( \sum_l \tau_l \right) \prod_l H\left[ \frac{\tau_l c^{1/2}}{m^{1/2}} [q_0^{1/2} z_l + (1-q_0)^{1/2} t_l] \right] \right]^m \right\}. \tag{23}$$

In order to have a well-defined extremum with respect to $m$ for $m \to 0$ the expression in the outer curly bracket in (23) should vanish in this limit. This gives an equation for $\alpha_c$:

$$0 = \min_{q_0,c} \left[ \frac{cq_0}{1+c(1-q_0)} + \ln[1+c(1-q_0)] + 2\alpha_c g(q_0,c) \right], \tag{24}$$

where

$$g(q_0,c) = \lim_{m\to 0} \int \prod_l Dz_l \ln \int \prod_l Dt_l \left[ \underset{\{\tau_l=\pm 1\}}{\text{Tr}} \Theta\left( \sum_l \tau_l \right) \prod_l H\left[ \frac{\tau_l c^{1/2}}{m^{1/2}} [q_0^{1/2} z_l + (1-q_0)^{1/2} t_l] \right] \right]^m. \tag{25}$$

Equation (24) determines $\alpha_c$ in an implicit way. It defines a function $f(c,q_0,\alpha)$ and states that $\alpha_c$ is that value of $\alpha$ for which the minimum of $f$ with respect to $c$ and $q_0$ happens to be zero. Since $g(q_0,c)<0$ for all $c$ and $q_0$, we can make the determination of $\alpha_c$ more explicit. We introduce the function $\alpha=\alpha(q_0,c)$ defined by $f(c,q_0,\alpha(q_0,c))=0$ and find $\alpha_c=\min_{q_0,c}\alpha(q_0,c)$, i.e., with (24):

$$\alpha_c=\min_{q_0,c}\left[-\frac{\dfrac{cq_0}{1+c(1-q_0)}+\ln[1+c(1-q_0)]}{2g(q_0,c)}\right], \tag{26}$$

where $g(q_0,c)$ is given by (25).

It is still difficult to determine $g(q_0,c)$ for general $K$. Note that the corresponding expression for the parity machine is much less troublesome since $q_0=0$ in this case [10]. In Appendix B we determine explicitly $g(q_0,c)$ for $K=3$, by performing the limit $m\to 0$ in (25). The numerical solution of (26) then gives $\alpha_c(K=3)\simeq 3.0$, with $q_0\simeq 0.61$ and $c\simeq 50$.

One-step replica-symmetry breaking thus reduces the value of $\alpha_c$ as compared to the replica-symmetric result. This reduction is less than in the case of the parity machine. We find for $K=3$ a decrease of $\alpha_c$ from 4.02 to 3.0. For the parity machine the corresponding values for $K=3$ are 10.3 and 5.0 [10]. This is in accordance with our qualitative discussion of the connectivity of the solution space. For the parity machine it is "checkered" and a replica-symmetric calculation is even less appropriate than in the case of the committee machine. The reduction of $\alpha_c$ in one-step replica-symmetry breaking should also apply to the asymptotic behavior of $\alpha_c$ for $K\to\infty$. It would be very interesting to see whether the modified asymptotics obeys the Mitchison-Durbin bound as in the case of the parity machine [9], but this seems to be very difficult. One can show that $\alpha_c$ is bounded by $\ln K$ for large $K$ if $c$ increases with $K$ at most like a power $c\sim K^x$ with arbitrary $x<\infty$. However, we were not able to extract this behavior of $c$ with $K$ out of the self-consistent equation for $c$.

Another open question concerns the reliability of a one-step symmetry-breaking calculation. We expect that the result for $\alpha_c$ for $K=3$ is a good approximation of the actual value. However, we do not see a convincing argument that one-step replica-symmetry breaking yields already the exact result as, e.g., in the case of the random energy model [22]. To show that two-step replica-symmetry breaking reduces to the one-step solution is very complicated for the committee machine and, in any case, relies on nontrivial numerical work. Moreover our qualitative analysis of the solution space suggests that the number of replica-symmetry-breaking steps necessary to obtain a fair approximation for $\alpha_c$ may increase with increasing $K$.

## V. FULLY CONNECTED ARCHITECTURE

In this section we discuss the replica-symmetric theory for a committee machine where every hidden unit is connected to *all* input units, i.e., $j(l)=j=1,\ldots,N$ for all $l$. The calculation proceeds along the lines of Sec. II. Now however the average over the patterns produces a term of the form

$$\exp\left[-\frac{1}{2N}\sum_{\mu,j}\sum_{l,k}\sum_{\alpha,\beta}x^{\alpha}_{\mu l}x^{\beta}_{\mu k}J^{\alpha}_{lj}J^{\beta}_{kj}\right]. \tag{27}$$

In order to decouple the integrals over $J^{\alpha}_{lj}$ and $x^{\alpha}_{\mu l},\lambda^{\alpha}_{\mu l}$ we have to introduce additional order parameters:

$$C^{\alpha}_{kl}=\frac{1}{N}\sum_j J^{\alpha}_{kj}J^{\alpha}_{lj}\quad(k\neq l) \tag{28}$$

and

$$D^{\alpha\beta}_{kl}=\frac{1}{N}\sum_j J^{\alpha}_{kj}J^{\beta}_{lj}\quad(\alpha\neq\beta,k\neq l) \tag{29}$$

together with their conjugate Lagrange multipliers $\hat{C}^{\alpha}_{kl}$ and $\hat{D}^{\alpha\beta}_{kl}$. These order parameters describe the correlations between synapses which leave the same input unit and arrive at different hidden units. $C^{\alpha}_{kl}$ characterizes these correlations within one solution whereas $D^{\alpha\beta}_{kl}$ correlates different solutions. It is straightforward to rewrite $\langle\langle V^n\rangle\rangle$ in the form of a saddle-point integral. Assuming replica symmetry the saddle point is parametrized by the seven order parameters $E$, $q$, $F$, $C$, $\hat{C}$, $D$, and $\hat{D}$. The saddle-point equations for $E$, $F$, $\hat{C}$, and $\hat{D}$ are algebraic and these variables can be eliminated. The resulting expression for $\langle\langle \ln V\rangle\rangle$ reads

$$\begin{aligned}\frac{1}{N}\langle\langle \ln V\rangle\rangle=\operatorname*{extr}_{q,C,D}\Bigg[&\frac{K-1}{2}\ln(1-q-C+D)+\tfrac{1}{2}\ln[1-q+(K-1)(C-D)]+\frac{K}{2}\frac{q}{1-q+(K-1)(C-D)}\\&+\frac{K(K-1)}{2}\frac{(q-D)(C-D)}{(1-q-C+D)[1-q+(K-1)(C-D)]}\\&+\alpha\int\prod_l Dt_l\int Dy\ln\int Dz\operatorname*{Tr}_{\{\tau_l=\pm1\}}\Theta\left[\sum_l\tau_l\right]\prod_l H\left[\tau_l\frac{(q-D)^{1/2}t_l+(C-D)^{1/2}z+D^{1/2}y}{(1-q-C+D)^{1/2}}\right]\Bigg].\end{aligned} \tag{30}$$

For $\alpha\to\alpha_c$ we expect that the solution is unique up to permutations of the hidden units. Consequently solutions which are related by permutations of the hidden units are not connected in solution space. Hence we have $q\to 1$ and $D\to C$. Introducing

$$a=\frac{C-D}{1-q} \tag{31}$$

we find

$$\frac{1}{N}\langle\langle \ln V\rangle\rangle = \operatorname*{extr}_{q}\left\{\frac{1}{2(1-q)}\operatorname*{extr}_{D,a}\left[(1-q)(K-1)\ln(1-q)(1-a)+(1-q)\ln(1-q)[1+(K-1)a]+\frac{Kq}{[1+(K-1)a]}\right.\right.$$
$$+\frac{K(K-1)(q-D)a}{(1-a)[1+(K-1)a]}+2a_c(1-q)\int\prod_l Dt_l\int Dy\ln\int Dz$$
$$\left.\left.\times\operatorname*{Tr}_{\{\tau_l=\pm1\}}\Theta\left[\sum_l\tau_l\right]\prod_l H\left[\tau_l\frac{(q-D)^{1/2}t_l+(C-D)^{1/2}z+D^{1/2}y}{[(1-q)(1-a)]^{1/2}}\right]\right]\right\}. \tag{32}$$

We now assume that $a$ remains bounded for $q\to1$. This assumption is shown to be self-consistent below. Then the first two terms in (32) do not contribute as $q\to1$. In the last term the limit $q\to1$ can be performed using the techniques of Appendix A. We then get an equation for $\alpha_c$ similar to (18):

$$0=\operatorname*{extr}_{C,a}\left\{\frac{1-a+a(K-1)(1-C)}{1+(K-1)a}-\alpha_c(1-C)\int Dy\int_{(C/1-C)^{1/2}y}^{\infty}Dt\left[t-\left[\frac{C}{1-C}\right]^{1/2}y\right]^2 g(t)\right\} \tag{33}$$

with $g(t)$ defined in Eq. (19). The extremum condition with respect to $a$ gives

$$C=-\frac{1}{(K-1)}\ . \tag{34}$$

The extremum condition with respect to $C$ shows that $a$ remains $O(1)$ for $q\to1$ thus verifying our assumption. From (34) and (33) we find

$$\alpha_c^{-1}=\frac{K}{K-1}\int Dy\int_{iy/K^{1/2}}^{\infty}Dt\left[t-\frac{iy}{K^{1/2}}\right]^2 g(t)\ . \tag{35}$$

Using (19) one can show that this expression is real. For $K=3$ we find the numerical result $\alpha_c^{FC}\simeq34.5$. So in replica symmetry the storage capacity per synapse is for the fully connected net almost three times larger than for the tree.

For large $K$, $C$ tends to zero and we get

$$\alpha_c^{FC}=K\alpha_c^{\text{tree}}\ . \tag{36}$$

The result (34) for the order parameter $C$ measuring the correlations between synapses leaving the same input unit but arriving at different hidden units is remarkably simple. It can be understood qualitatively as follows. At $\alpha_c$ most patterns have internal representations with $(K+1)/2$, $\sigma_l=+1$, and $(K-1)/2$, $\sigma_l=-1$. That means that on the average two different hidden units will have the same value for a fraction $[(K-1)/2K]$ and different values for a fraction $[(K+1)/2K]$ of the patterns. This gives rise to an anticorrelation $\langle\langle\sigma_l\sigma_k\rangle\rangle\cong-1/K$. This result shows that a committee machine is indeed more complex than just $K$ perceptrons. In particular $C<0$ demonstrates that there is a genuine "division of labor" between the hidden units. It is also interesting that Mézard and Patarnello find for the fully connected parity machine $C=0$ for all $K$ [18].

We expect that the order parameters $C$ and $D$ are decreasing functions of $K$, also if replica-symmetry breaking is taken into account. Then the storage capacity per synapse should be the same for the fully connected and the tree architecture in the limit of *large* $K$. This has been shown explicitly for the replica-symmetric solution.

## VI. SIMPLE LEARNING ALGORITHM

In this section we present a simple learning algorithm for the committee tree, which is able to learn $p\sim O(N/\sqrt{K})$ patterns. The motivation of this chapter is to show that the tree can learn more patterns than one of its branches ($p_{\text{branch}}=2N/K$ [4]). The idea of the algorithm is very simple. Each hidden unit learns $2N/K$ patterns. The units are trained one after the other; each new one learns only those patterns, which do not yet produce the right majority vote at the output unit. If one is interested in high storage capacities, this algorithm is useless. It is best for no hidden units at all, i.e., $K=1$. However the suggested algorithm allows us to make use of all the available know-how in training of single-layer perceptrons also in two-layer nets with tree architecture. This may be advantageous in applications, where for technical reasons one may favor more units with low connectivity instead of few units with high connectivity.

We start with a description of the learning algorithm for given $N$, $K$, and $p$, where $p\le p_c(N,K)$. Then we go on to calculate $p_c(K/N)$.

We assume $N/K\gg K$ and denote an optimal algorithm for the single-layer perceptron by $\mathcal{A}_{\text{branch}}$. Without loss of generality the desired output for very pattern is $+1$.

First the lower synapses are all set equal to 0. Then we successively set the lower synapses at $+1$ and learn the upper synapses in the corresponding branch. After each step the field at the output unit for each pattern is calculated. As all lower synapses have weight 1 or 0, the fields can only take integer values. The field distribution is used to select those patterns, which will be learned by the

next unit. The detailed procedure is as follows. In the beginning all patterns have field $h^{\mu}(0)=0$. The first branch is learned with $\mathcal{A}_{\text{branch}}$ such that the first $2N/K$ patterns set the hidden unit to $+1$. Half of the other patterns will have hidden output $+1$ (they have been learned "by accident") and the other half will have $-1$, because the patterns are stochastically independent. Now the lower synapse connecting the first hidden unit to the final output unit is set to $+1$. So after the first step the field distribution is

$$P_1(h)=\left[\frac{p}{2}-\frac{N}{K}\right]\delta(h+1)+\left[\frac{p}{2}+\frac{N}{K}\right]\delta(h-1)=\left\langle\!\left\langle \sum_{\mu}\delta(h-h^{\mu}(1))\right\rangle\!\right\rangle_{\xi} . \tag{37}$$

In the next step the second unit learns $2N/K$ patterns with field $h^{\mu}(1)=-1$. After training the second unit, the set of all $p$ patterns can be classified according to their local fields $h^{\mu}(2)=-2,0,+2$. The third unit learns those $2N/K$ patterns, which have the smallest fields and so on. In Fig. 5 we show the field distribution after each step for $p=5N/K$. In the second step $2N/K$ patterns with field $-1$ are learnt by $\mathcal{A}_{\text{branch}}$ to give $+1$ at the second hidden unit. In the example (Fig. 5) there are only $1.5N/K$ patterns with field $-1$. So all these patterns are learned at the second hidden unit together with $0.5N/K$ patterns with field $+1$. The remaining $3N/K$ patterns with field $+1$ have been divided: half of them will give $+1$ at the second hidden unit, the other half will give $-1$. The second lower synapse is finally set $+1$. Let us summarize the effect on the field distribution of step 2: $1.5N/K$ patterns with field $-1$ are learned, so their fields become 0. $0.5N/K$ patterns with field $+1$ are learned to field $+2$. $1.5N/K$ patterns with field $+1$ are shifted to fields 0 and $+2$, respectively. All subsequent steps are performed in the same spirit. $\mathcal{A}_{\text{branch}}$ is applied to learn $2N/K$ of the patterns with the worst (most negative) field. If those are less than $2N/K$, patterns with the next worse field are learned. Half of all other fields will be increased by $+1$, the other half decreased by $-1$.

The algorithm will be successful if after $K$ steps all fields are positive. This will be true if $p\leq p_c(K,N)$. The storage capacity is most easily calculated by determining $K_{\min}(p,N/K)$, defined as the minimal number of branches which is needed to learn $p$ patterns for fixed $N/K$. One starts with $p$ patterns, calculates the field distributions after each step, and counts the number of steps $K_{\min}$ needed to assure that all fields be positive. The result is shown in Fig. 5 for $pK/N=5$. Note that $K_{\min}$ only depends on the ratio $pK/N$ (instead of two variables $p$ and $N/K$), because if $p$ and $N/K$ are multiplied by the same factor, the field distribution before the first step $P_0(h)=p\delta(h)$ and the number of patterns, which are learned in one step $2N/K$, are scaled by the same factor. Hence all field distributions are just rescaled. $K_{\min}$ is always odd, because the last step always learns patterns, which had field 0 to field $+1$. For each odd $K$ the maximum storage capacity $p_c$ can be determined by calculating $K_{\min}(pK/N)$ in the neighborhood of $p_c$. At $p_c$ $K_{\min}$ jumps from $K$ to $K+2$.

In Fig. 6 we show $p_cK/N$ as a function of $K$. In order to verify the guess $p\sim N/\sqrt{K}$ for $K\to\infty$ we plotted $p\sqrt{K}/N$ against $N/pK$. The result is

$$p_c=N\left[3.086-\frac{1}{\sqrt{K}}-0.87\frac{1}{K}+O(K^{-3/2})\right] , \tag{38}$$

where the coefficient of $K^{-1}$ has been determined by the slope of the curve in Fig. 7. We present the following remarks.

(1) One can generalize the algorithm in selecting patterns which do not have the most negative field, e.g., $N/K$ patterns which have a negative field with the lowest absolute value could be learned together with $N/K$ patterns with the most negative field. For small, fixed $K$ it can be shown analytically that the most-negative-field rule yields the maximum $p$. The proof involves the solution of a linear optimization problem with $O(K^2)$ variable. We performed the calculation up to $K=11$. As a by-product one gets $p_c(K)$ directly

| $K$ | $\frac{pK}{N}$ |
|---|---|
| 1 | 2 |
| 3 | 4 |
| 5 | $5\frac{1}{2}$ |
| 7 | $6\frac{24}{33}$ |
| 9 | 7.80 |
| 11 | 8.76 |

(2) The algorithm can be applied if the lower synapses have nonuniform weights. We shall not investigate this possibility systematically. We just mention that for given $K$ one might be able to increase $p_c$, e.g.,

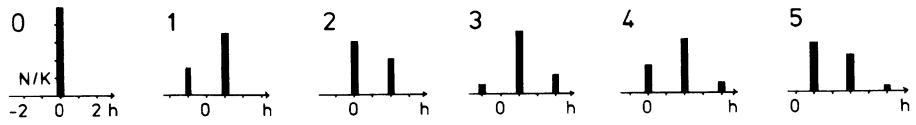


FIG. 5. Field distribution $P_i(h)$ at each learning step for $pK/N=5$.

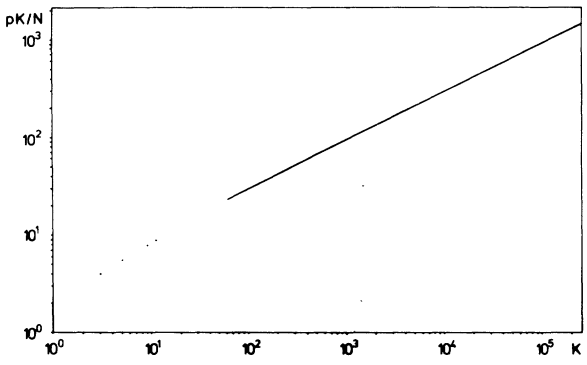


FIG. 6. Storage capacity $\alpha K = pK/N$ as a function of the number of hidden units $K$ as obtained by the simple algorithm. For finite $K$ the graph has a slight curvature.

$p_c(K=5)=6N/K$ (instead of $5.5N/K$) if the lower synapses have values 2,2,1,1,1.

(3) Note that $\alpha_c(N,K)$ *decreases* with increasing $K$. One reason why the algorithm is not optimal is the following. It does not contain any back-correlations. Consider an arbitrary hidden unit $l$ with $1<l<K$. It has information about parts of the patterns, which have been learned by the first $(l-1)$ hidden units, because it is trained only with patterns, which did not produce the desired output so far. However unit $l$ lacks any information about parts of patterns which will be learned later, i.e., by hidden units $l+1,\ldots,K$.

## VII. TOWARD A BETTER LEARNING ALGORITHM

Learning in a two-layer network is in general a hard combinatorial optimization problem, so approximate techniques are needed to find good solutions in reasonable time. We will have a closer look at the least-action algorithm of Mitchison and Durbin [9] and improve it by taking special advantage of the two-layer structure. Least action goes back to a learning algorithm for the committee machine that was described by Nilsson [23]. It was reformulated [9] for the parity machine, which requires three layers of simple-summing-threshold units, but can also be applied to the committee machine and other two-layer nets. It is a generalization of

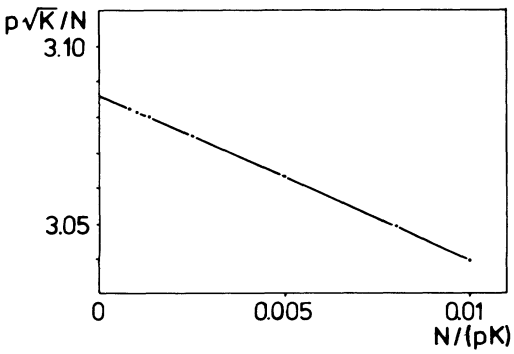


FIG. 7. Finite-size scaling of the storage capacity of the simple algorithm. The asymptotic behavior of $p(K,N)$ [Eq. (38)] can be extracted from the plot.

Rosenblatt's perceptron algorithm [5]. Patterns are cyclically presented to the network; whenever a pattern needs to be learned, i.e., if the parity of the hidden layer is not as desired or the field of the output neuron lower than its threshold, a hidden unit is selected and a Hebbian term is added to the couplings of the selected unit. In a simple perceptron there is only one unit (or, respectively, $N$ independent units). The selection of the hidden unit in the layered net is done according to the rule that the local field $h_l^\mu$ of the selected unit is closest to its threshold. Then a little change of the couplings may be sufficient to change the sign of the unit's output, which in turn would change the parity or the field of the output unit. This however is not always the case. The algorithm has been applied to the fully connected parity and the committee machine (with an even number of hidden units) [9] and to the tree-structured parity machine [10]. Mitchison *et al.* also showed that it is much more efficient than the back-propagation of Rumelhart *et al.* [24].

In a two-layer network we have more information on how well a particular pattern is already learned than just the binary output of a parity machine—we know the field at the output unit. Since we want to learn all patterns of a specific set we can always select a pattern with index $\mu$ that is *worst* (or one of the worst), in the sense that its field $h^\mu$ has the most undesirable (e.g., smallest) value of all fields $h^\nu$ $(\nu=1,\ldots,p)$. Now not only the hidden unit but also the pattern is selected in each step. The step itself can be made *adaptive,* that is instead of adding a simple Hebbian term to the couplings we multiply it with the distance to the desired local field 1 or $-1$ as in Adaline learning [25–27], see also [28]). For the fully connected network the change of couplings is explicitly given by

$$J_{lj}(t+1)=J_{lj}(t)+\frac{1}{\sqrt{N}}[1-h_l^\mu(t)]\xi_j^\mu\eta^\mu \tag{39}$$

with

$$h_l^\mu(t)=\frac{1}{\sqrt{N}}\sum_j J_{lj}(T)\xi_j^\mu\eta^\mu \ .$$

Thereby we ensure that the learning step is successful in the sense that the sign of the hidden unit is changed, and at the same time that it does not overachieve its requirements, i.e., produce a very large field. Note that the target state of the hidden unit is $\eta^\mu$, because the output unit just calculates the majority of the hidden units and a hidden unit is selected in the algorithm only if it gives the wrong vote.

In our simulations and also in the formulation of the algorithm we restrict ourselves to the simplest case of a two-layer network, a committee machine with zero thresholds, and an odd number of hidden units, in order not to burden the model with too many details. The algorithm, which we call the adaptive least-action (ALA) algorithm, can later be modified to take additional freedoms like threshold or variable couplings from hidden to output into consideration. We will formulate and apply the algorithm to the tree and also to the fully connected net in a unified way. Without loss of generality we may

again map all patterns on $\eta^\mu=+1$. It is convenient to store the correlation matrix of the patterns $(\mu,\nu=1,\ldots,p)$ for the tree,

$$C_l^{\mu\nu}=\frac{K}{N}\sum_{j(l)}\xi_j^\mu\xi_j^\nu\ , \tag{40}$$

and for the fully connected network,

$$C^{\mu\nu}=\frac{1}{N}\sum_{j}\xi_j^\mu\xi_j^\nu\ . \tag{41}$$

We can write the couplings as $J_{lj}=\sum_\nu x_l^\nu\xi_j^\nu$ and therefore $h_l^\nu=\sqrt{N/K}\sum_\mu x_l^\mu C_l^{\mu\nu}$ or, respectively, $h_l^\nu=\sqrt{N}\sum_\mu x_l^\mu C^{\mu\nu}$. All calculations can be done with the embedding strengths $x_l^\nu$, so we do not need to store $J_{lj}$ until we start calculating order parameters. Slightly formalized, the main part of the algorithm reads

set $x_k^\nu\equiv 0$ for all $\nu$ and $k$

for $t=0,1,\ldots$

find first pattern $\mu$ with $h^\mu=\min_\nu h^\nu$

if $h^\mu>0$ solution found, STOP

find first hidden unit $l$ with $h_l^\mu=\max_{k,h_k^\mu\le 0}h_k^\mu$

let $x_l^\mu(t+1)=x_l^\mu(t)+(1-h_l^\mu)(C_{(l)}^{\mu\mu})^{-1}$

update local fields $h_k^\nu$ and fields $h^\nu$.

The algorithm is equally well applicable for binary and for real-valued patterns with or without a bias. Note that $C_{(l)}^{\mu\mu}=1$ for binary patterns, where $C_{(l)}^{\lambda\nu}$ stands for $C_l^{\lambda\nu}$ or $C^{\lambda\nu}$, respectively, for the tree and the fully connected network. Also a stopping criterion is needed for the case that no solution can be found. It would be too crude to stop the algorithm after a fixed number of steps $T$, because a significant stopping time will depend on system sizes $N$ and $K$ and also on the special choice of the set of patterns. We therefore check the change of the couplings of the last $T$ steps against their $L_2$ norm $|J_l|=\sum_{\lambda,\nu}x_l^\lambda C_{(l)}^{\lambda\nu}x_l^\nu$. The algorithm is stopped with no success after $\tau T$ steps, if for some small $\epsilon>0$

$$\frac{K\sum_{t=(\tau-1)T+1}^{\tau T}\delta x_{l(t)}^{\mu(t)}}{T\sum_{\lambda,\nu,k}x_k^\lambda C_{(k)}^{\lambda\nu}x_k^\nu}<\epsilon\ . \tag{42}$$

where $\delta x_{l(t)}^{\mu(t)}$ is the change of $x_k^\mu$ in step $t,\tau=1,2,\ldots$. Convergence problems were never observed. For a given pattern and hidden unit the algorithm is equivalent to the quadratic optimization problem of Adaline learning. The nonlinear and nonsmooth learning problem is fitted with quadratic cost functions in every step and therefore belongs to the class of the trust region methods (see, e.g., [29]).

Usually all hidden units are changed approximately equally often during one learning cycle. We did not see the effect that the algorithm is caught frequently in local traps, where only a single or a few hidden units are repeatedly changed without improving the total solution significantly. This effect makes problems in Mitchison and Durbin's nonadaptive and therefore linear least action (LLA).

Simulations were performed for the tree and the fully connected network as well as for binary and Gaussian unbiased patterns. Gaussian patterns produce results with a weaker dependence on the system size. Note that the theory holds for both kinds of patterns. Figure 8 shows the distribution of the local fields $h_l^\nu$ normalized to $|J_l|=1$ for all $l$ after learning in a fully connected system and a tree with $K=3$, $N_{\text{tree}}=147$, and $N_{\text{full}}=49$ of 200 sample sets of Gaussian unbiased patterns each. It demonstrates the way learning works: fields are taken from small negative values and shuffled to $1/|J_l|$, which is a large change in the beginning of the learning process and decreasing, while $|J_l|$ is growing. During this shuffling other local fields are dropping back with the tendency to become Gaussian. The distribution of the corresponding embedding strengths $x_l^\nu$ after learning normalized in the same way is depicted in Fig. 9; 77% and, respectively, 74% of all $x_l^\nu$ have a value of zero for the tree and for the fully connected network. The distribution is characteristic for the learning algorithm; it is much smoother than the one of LLA, where we see sharp peaks at zero and also at 1. Differences between the tree and the fully connected network are apparently not significant in $x$ and $h$.

Storage capacities were determined by interpolating success rates to 50% in learning all patterns of a set. To be more precise, we determined a $p$ so that slightly more than 50% of all tested sets with $p$ patterns could be learned *completely,* then we increased $p$ so that the success rate went down. The intersection of the connecting line with 50% defines the critical number of patterns $p_c^{\text{ALA}}$. The critical storage capacity is defined as $\alpha_c^{\text{ALA}}(N,K)=p_c^{\text{ALA}}(N,K)/N$. The stopping parameter was chosen to be small enough to have little influence on $\alpha_c$ and also big enough to not waste too much computer time: $\epsilon=5\times10^{-5}$. Figure 10 shows storage capacities

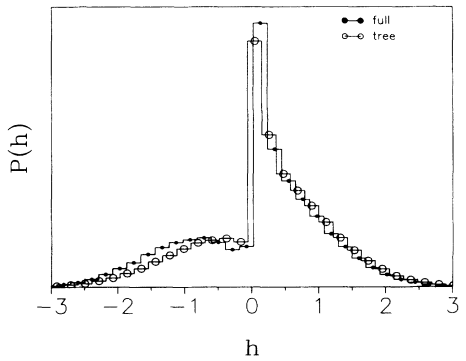


FIG. 8. Distribution of local fields $h_k^\nu$, $k=1,\ldots,K$, $\nu=1,\ldots,p$ for the fully connected and the tree-structured committee machine, after learning with adaptive least action. System sizes are $N=63$, $K=3$, and the numbers of patterns $P$ are at the critical storage capacity. Both distributions have been averaged over 200 independent sets of patterns.

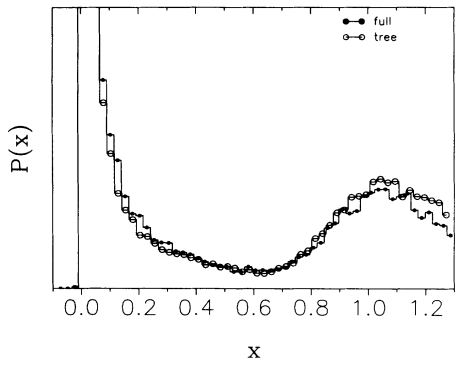


FIG. 9. Distribution of embedding strengths $x_k^\nu$, $k=1,\ldots,K, \nu=1,\ldots,p$ of the simulations of Fig. 8.

for the tree committee machine for $K=3$, 5, and 7 hidden units. The solid symbols show results for Gaussian distributed patterns with zero mean and variance 1. Open symbols give results for binary unbiased patterns, which show a stronger dependence on $N$, when $N$ is small. For sufficiently large $N$ we see a maximal possible storage capacity of $\alpha_c^{\mathrm{ALA}} \simeq 2$ for all $K$. This storage capacity is approximately twice the number of degrees of freedom, as for a single-layer perceptron, see also Fig. 12. The discrepancy to the theoretical result of $\alpha_c \approx 3.0$ may be due to further replica-symmetry-breaking effects or to insufficiencies of the algorithm. Even if the capacity is not significantly larger than 2, this does not mean that the tree could be replaced by a simple perceptron, since also nonlinear separable problems can be handled.

A slightly different result is found for the fully connected committee machine. Figure 11 shows its storage capacities, again for Gaussian and binary patterns (solid and open symbols). We see a storage capacity which reaches a value of approximately $2K$ for not too small $N$ ($\epsilon = 5\times 10^{-5}$). In order to check whether the algorithm is able to store more than this at the cost of more computer time, we tested the scaling behavior of $\alpha_c(\epsilon)$ for $K=3$ and sufficiently large $N$ (43 fully connected and 129 for the tree), see Fig. 12. We observe linear scaling of the fully connected storage capacity with $\epsilon^{1/2}$, which reaches $\alpha_c^{\mathrm{full}}/K \to \approx 2.2$ for $\epsilon \to 0$. The data points of $\alpha_c^{\mathrm{tree}}(\epsilon^{1/2})$ are not on a straight line; still the plot suggests for small

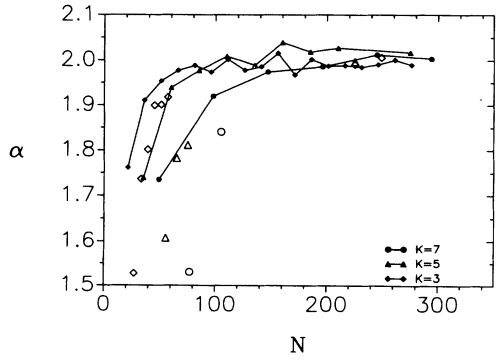


FIG. 10. Critical storage capacity of the tree committee machine for Gaussian patterns (full symbols connected with lines) and binary patterns (open symbols) and finite-size effects.

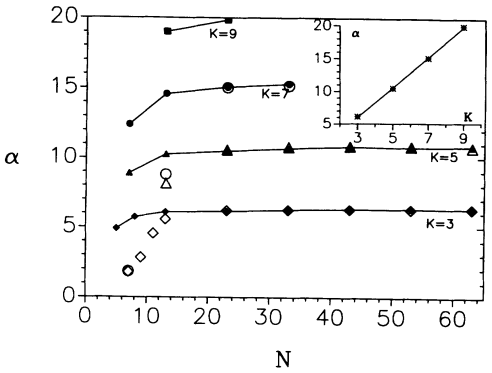


FIG. 11. Critical storage capacity of the full connected committee machine for Gaussian and binary patterns as in Fig. 10. Also shown is the storage capacity over $K$ for $N=27$, which indicates a linear scaling in that region with gradient 2.0.

$\epsilon$ that $\alpha_c^{\mathrm{tree}} \to \approx 2.02$. Compared to LLA we find that critical storage capacities are higher and can be reached with fewer iterations. In an example of $N=14$, $K=3$, and Gaussian patterns we found $\alpha_c^{\mathrm{LLA}}=5.48$, while ALA reaches $\alpha_c^{\mathrm{ALA}}=5.81$ in $52p$ compared to $50p$ iterations on the average for $\epsilon = 5\times 10^{-4}$. We always took $T=p$.

The algorithm as defined is deterministic. We find that the storage capacity cannot be increased by changing the initial conditions $x_0 \equiv x(t=0)$. We took all $x_0=+1$ or all $x_0=-1$ or Gaussian random $x_0$ with mean zero and variance 1. The worst results are obtained for $x_0=-1$, while random initial conditions reduce $\alpha_c^{\mathrm{LA}}$ by $\sim 10\%$ and increase the number of necessary iterations by a factor of 6.

In contrast to single-layer perceptrons it is not known whether the algorithm finds a solution, if one exists. This may well be the reason for the observed discrepancy between the calculated capacities and the capacities reached by ALA. On the other hand, we cannot exclude that the theoretical capacity is further reduced by more steps of replica-symmetry breaking.

For the full connected net it is of interest to study correlations among different hidden units:

$$C_{kl} = \frac{1}{N} \sum_j J_{kj} J_{lj}, \quad k \neq l \ . \tag{43}$$

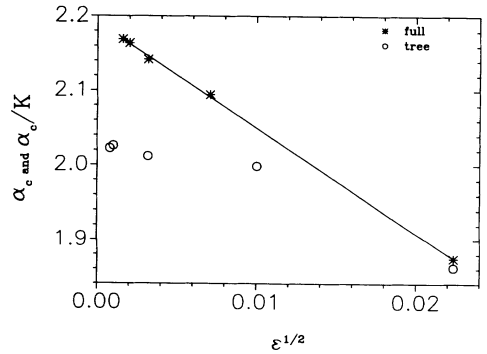


FIG. 12. Scaling of $\alpha_c$ for the fully connected and the tree-structured net with $\epsilon^{1/2}$ for $N=43$ (full) and $N=139$ (tree).

Its distributions are depicted in Fig. 13 for two systems with $K=3$, $N=11$, and $N=63$ (200 sets of patterns). It shows that the distributions get sharper with growing $N$ at a value of $\approx -0.35$. Finite-size effects are depicted in Fig. 14, where average values of $C$ are plotted versus $N$. The small inset graph shows $C$ of $N=22$ as a function of $1/(K-1)$. The data points are fairly well on a straight line with the gradient $-1.2$, which is remarkably close to $C=-1/(K-1)$, the replica-symmetric result in the limit $\alpha \to \alpha_c$, and supported by a simple argument (see Sec. V).

We would like to learn more about the solution space and in particular the other order parameters from numerical simulations. This is difficult, because we do not know whether the dynamics of Eq. (39) is ergodic or not. Hence it is not clear whether averaging over random initial conditions is equivalent to the average of statistical mechanics with equal *a priori* probabilities. Nevertheless we have seen that different initial conditions generate different fixpoints of the dynamics for the *same set of patterns*. To investigate correlations of these fixpoints, we first generate several solutions $\{J_{lj}^{\alpha}\}$ $(\alpha=1,\ldots,M)$ for a given set of patterns, using ALA with random Gaussian $x_0$ and $x_0(\alpha=1)=0$. For the tree we calculate their mutual overlaps

$$q_k^{\alpha\beta}=\frac{K}{N}\sum_{j}^{N/K} J_{kj}^{\alpha}J_{kj}^{\beta}, \quad \alpha<\beta \tag{44}$$

for each hidden unit $k$. The distribution of overlaps

$$P(q)=[P_{\xi}^{l}(q)]_{\xi,l}=\left[\frac{2}{M(M-1)}\sum_{\alpha,\beta}\delta(q-q_l^{\alpha\beta})\right]_{\xi,l} \tag{45}$$

averaged over patterns and hidden units is shown in Fig. 15 with $K=3$ and 5, $N=49$, and 200 sets of patterns. The storage ratio was $\alpha=1.85$, which is somewhat below the critical storage capacity of the algorithm, so that solutions can be found with acceptable success rates also for random initial conditions. Since $q$ will not change rapidly at $\alpha_c$, all sets of couplings were correlated, whether learning was successful or not. A growing number of solutions $M$ improves the statistics of the distribution, but takes more of the second-best solutions into account, with the result of a decreasing $q$ ($K=3$: $M=3$, solid triangles; $M=4$, solid circles). Both $K=3$ distributions are still well peaked at $q=0.2$. For $K=5$ we see the peak at $q\simeq 0.14$ ($M=4$). If the learning dynamics (39) is ergodic these results would suggest that at $\alpha=1.85$ replica symmetry is still unbroken.

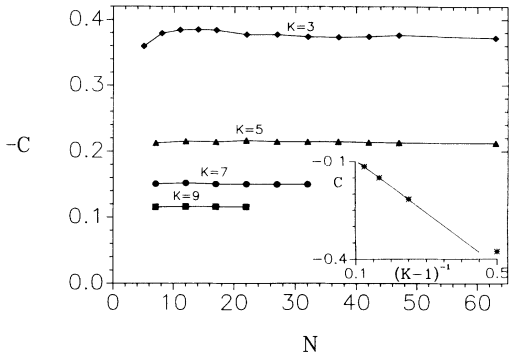


FIG. 14. Finite-size effects for the order parameter $C$ of the fully connected net. Plotted are average values of $C$ from 200 sets of Gaussian patterns for $K=3$, 5, 7, and 9 and $N$ up to 63. The inset graph shows data for $C$ over $(K-1)^{-1}$ for $N=22$, which lie on a straight line with gradient 1.2.

The order parameters $q$ and $D$ of the fully connected committee machine were obtained by correlating four different solutions; one of them was obtained from $x_0=0$. Since the full net has a permutational symmetry in the hidden units, any solution can prefer any permutation of the hidden units at random. We therefore determined for all pairings $(\alpha,\beta)$ an individual mapping of the hidden units $\{k\}\to\{k'\}$, which maximizes

$$q_k^{\alpha\beta}=\frac{1}{N}\sum_{i} J_{ki}^{\alpha}J_{k'i}^{\beta}, \quad \alpha<\beta\ . \tag{46}$$

The remaining combination of hidden units forms the order parameter $D$:

$$D_{kl}^{\alpha\beta}=\frac{1}{N}\sum_{i} J_{ki}^{\alpha}J_{li}^{\beta} \quad l\neq k', \quad \alpha<\beta\ . \tag{47}$$

Figure 16 shows the distribution of $q$ and $D$ of systems with $N=49$, $K=3$ averaged over 200 independent sets of patterns again; $q$ appears to be a little smaller than in the

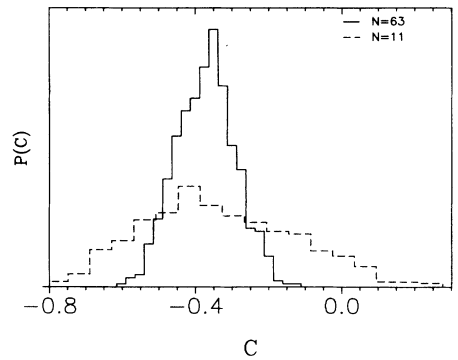


FIG. 13. Distribution of the order parameter $C$ of the fully connected net with Gaussian patterns. $K=3$, $\alpha=1.85$. For $N=11$ and 63, the distributions get sharp with growing $N$.

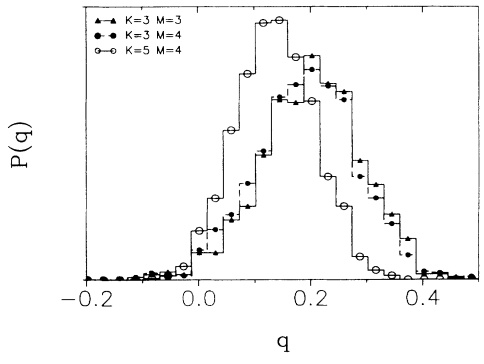


FIG. 15. Distribution of the order parameter $q$ of the tree, $N=49$, $K=3$, $\alpha=1.85$. Three and four solutions of 200 sets of binary patterns were correlated.

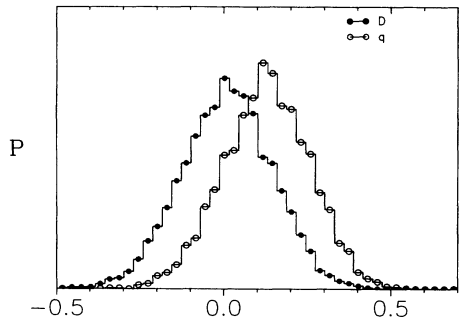


FIG. 16. Distributions of the order parameters $q$ and $D$ of the fully connected net, $N=49$, $K=3$, $\alpha=1.85$. Four solutions of 200 sets of Gaussian patterns were correlated.

tree. The average value is approximately 0.15, while $D$ is distributed around zero.

Also for the parity machine adaptive step lengths are possible. Advantages are not as clear as in the committee case. In a simulation of a system with $N=23$ we saw an increase of the critical storage capacity from 5.2 to 5.6 at the cost of 80% more necessary iterations. Further simulations need to be performed to get a clearer picture of the matter.

The suggested algorithm ALA can also be used for fully connected two-layer networks with variable couplings $\{w_k\}$ from hidden to output layer. We again restrict ourselves to positive signs of the $\{w_k\}$ and order them according to their absolute value, since the net is invariant under permutations of the hidden units. Nonuniform values of the $\{w_k\}$ are prescribed in advance and fixed during learning. The algorithm takes the different values of $w_k$ into account: Concerning the patterns, the fields $h^\mu$ reflect the influence of $\{w_k\}$. Hidden units can be selected as for uniform couplings. In this way simulations can be used to learn more about the appropriate distribution of $\{w_k\}$.

## VIII. GENERALIZATIONS AND OUTLOOK

So far we have mainly discussed a special case of a two-layer network, where the output unit adjusts according to the majority of hidden units. In this section we shall try to put our work in a more general context and investigate some possible extensions. In particular we shall discuss other Boolean functions, variable weights from the hidden layer to the output unit, and the effects of thresholds. Finally some open questions will be pointed out.

### A. General Boolean function from hidden units to output

It is easy to generalize our results for the maximal storage capacity to a feedforward network with an arbitrary but fixed relation between hidden units and output. Consider the same model as defined in Sec. II but with

$$\eta=F(\{\sigma_l\}) \tag{48}$$

instead of (2), where $F$ is an arbitrary Boolean function of $K$ variables. In complete analogy to Sec. III we then get for the replica-symmetric value of $\alpha_c$ instead of (17)

$$(\alpha_c^{\mathrm{RS}})^{-1}=-\lim_{q\to 1} 2(1-q)\int \prod_l Dt_l \left\langle\!\!\left\langle \ln\left[\operatorname*{Tr}_{\{\tau_l=\pm 1\}} \Theta(\eta F(\{\tau_l\}))\prod_l H(Qt_l\tau_l)\right]\right\rangle\!\!\right\rangle_\eta . \tag{49}$$

For a symmetric Boolean function $F(\{-\sigma_l\})=-F(\{\sigma_l\})$ the $\eta$ average in (49) can be performed just by omitting $\eta$ in the argument of the $\Theta$ function. To get an explicit result for $\alpha_c$ similar to (18) one has to take the limit $q\to 1$ in (49) which now depends on the detailed form of $F$. It is however possible to find the result by inspection, as we now show. This will not only facilitate the derivation of $\alpha_c$ for arbitrary Boolean functions, but at the same time provide a deeper understanding of the storage properties of networks with hidden units.

We rewrite Eq. (18) in the following form:

$$\alpha_c^{-1}=K\int_0^\infty Dt\, t^2 g(t) . \tag{50}$$

For $g(t)=1$ this is precisely Gardner's formula for the storage capacity of a perceptron with $N/K$ synapses storing $p=\alpha N$ patterns [4]. The advantage of a two-layer network is that not all subperceptrons have to adapt to all patterns. If, e.g., one pattern produces a large negative field at hidden unit $\sigma_l$, then the correct output can be achieved with the help of other subperceptrons without changing the synaptic couplings of $\sigma_l$. The capacity of the two-layer net is reached if one of the subperceptrons, which *has to be adapted,* is at its limit of capacity. We then say that this subperceptron becomes saturated or critical. Hence the storage capacity of the network with hidden units is given by Gardner's formula with the interpretation of $g(t)$ as the probability that the critical subperceptron has to be modified.

To see how we can construct $g(t)$ if $F(\{\sigma_l\})$ is known, we consider the parity machine as an example. In this case $F(\{\sigma_l\})=\prod_l \sigma_l$ and the correct output can be achieved by adapting only one subperceptron. To maximize the capacity, this particular subperceptron should be chosen as the one with the lowest value of the noise $\{t_l\}$. (Note that this is also done in the more elaborate calculation of Appendix A.) Without loss of generality we choose subperceptron 1. It will be modified only if $|t_1|<|t_l|$ for $l=2,3,\ldots,K$. The probability that for a given value of $t_1$ we have $|t_1|<|t_2|$ is given by $2H(t_1)$. This implies

$$g(t_1)=[2H(t_1)]^{K-1} \tag{51}$$

with a factor $2H(t_1)$ for each $l=2,\ldots,K$. (This is of course in agreement with the results of Ref. [10].)

As a second example we consider the committee machine, i.e., $F(\{\sigma_l\})=\sum_l \sigma_l$. If a pattern does not

yield the correct output, we may have to change up to $(K+1)/2$ subperceptrons. This is the worst of all cases with all $\sigma_l=-1$. Again we want to adapt those perceptrons, which have the smallest values of the noise $\{t_l\}$. Hence the first subperceptron has to be modified, if $t_1$ belongs to the $(K+1)/2$ smallest $t_l$. Since all $\{t_l\}$ are independent Gaussian variables with variance 1 we find

$$g(t_1)=[H(t_1)]^{K-1} + \begin{bmatrix} K-1 \\ 1 \end{bmatrix}[H(t_1)]^{K-2}[1-H(t_1)]+\cdots + \begin{bmatrix} K-1 \\ \frac{K-1}{2} \end{bmatrix}(H(t_1))^{K-1-(K-1)/2}[1-H(t_1)]^{(K-1)/2} \tag{52}$$

which coincides with Eq. (19).

Similarly one can construct expressions for the replica-symmetric $\alpha_c$ for other Boolean functions $F(\{\sigma_l\})$. Two more examples will be discussed in the next paragraph, where we study the effects of thresholds. It is also possible to get results for $\alpha_c$ in one-step replica-symmetry breaking along the lines of Sec. IV. In particular it is easy to generalize Eqs. (22) to (26) to arbitrary Boolean functions $F(\{\sigma_l\})$.

### B. Thresholds

We can generalize the main results of Secs. III and IV to networks of formal neurons with thresholds. Denoting the threshold of the hidden units by $\theta_h$ and that of the output unit by $\theta_0$ we find in replica symmetry similar to (16)

$$\frac{1}{N}\langle\langle \ln V\rangle\rangle = \operatorname*{extr}_q \left\{ \tfrac{1}{2}\ln(1-q)+\frac{q}{2(1-q)}+\alpha\left\langle\!\!\left\langle \int \prod_l Dt_l \ln \left[ \operatorname*{Tr}_{\{\tau_l=\pm 1\}} \Theta\left[\eta\left(\sum_l \tau_l-\theta_0\right)\right]\prod_l H\left[\tau_l\frac{\theta_h+q^{1/2}t_l}{(1-q)^{1/2}}\right]\right]\right\rangle\!\!\right\rangle_\eta \right\}. \tag{53}$$

The interpretation is the same as for (16). The $\tau_l$ trace is restricted to internal representations that yield the desired output and the Gardner volumes of the subperceptrons are now characterized by a stability parameter $\theta_h$ [4]. In the derivation of Eq. (53) we have assumed that all hidden units have the same threshold. If this is not the case, one has to expect an $l$ dependence of the order parameters $E$, $F$, and $q$. Note that thresholds break the $S_i\to -S_i$ symmetry so that the statistics of the output unit $\eta^\mu$ must be taken into account explicitly in (53). Similar to (17) we then find

$$(\alpha_c^{\mathrm{RS}})^{-1}=-\lim_{q\to 1} 2(1-q)\left\langle\!\!\left\langle \int \prod_l Dt_l \ln \left[ \operatorname*{Tr}_{\{\tau_l=\pm 1\}} \Theta\left[\eta\left(\sum_l \tau_l-\theta_0\right)\right]\prod_l H\left[\tau_l\frac{\theta_h+q^{1/2}t_l}{(1-q)^{1/2}}\right]\right]\right\rangle\!\!\right\rangle_\eta . \tag{54}$$

The limit $q\to 1$ can now be performed using the techniques of Appendix A or the intuitive argument of the previous paragraph. We discuss two cases in detail.

(a) $\theta_0=0,\theta_h>0$. If the desired output is positive, $\eta=+1$, the inverse capacity is given by Gardner's result for a single-layer perceptron with stability parameter $\kappa=\theta_h$, multiplied by $g(t)$. If the desired output is negative, $\eta=-1$, the threshold $\theta_h$ facilitates the right decision of the committee, corresponding to a negative stability parameter in Gardner's calculation, i.e., $\kappa=-\theta$. Adding both terms with weight $\frac{1}{2}$ we obtain

$$\alpha_c^{-1}(\theta_h)=\frac{K}{2}\left[\int_{-\theta_h}^{\infty} Dt\,(t+\theta_h)^2+\int_{\theta_h}^{\infty} Dt\,(t-\theta_h)^2\right]g(t), \tag{55}$$

with $g(t)$ given in Eq. (52). The capacity has a maximum for $\theta_h=0$, as can be seen by differentiating Eq. (55) twice.

(b) $\theta_h=0,\theta_0>0$. In this case $\alpha_c(\theta_0)$ changes discontinuously if $\Delta\theta_0=2$ and remains constant in between. Hence we only consider $\theta_0=0,2,\ldots,K-1$. The probability $g(t)$ that the critical subperceptron has to be modified is given by the probability that it belongs to the $(K+\theta_0+1)/2$ $[(K-\theta_0+1)/2]$ subperceptrons with smallest noise $t_l$ for output $\eta=+1(\eta=-1)$. Adding both terms with weight $\frac{1}{2}$, we obtain

$$\alpha_c^{-1}(\theta_0)=\frac{K}{2}\int_0^\infty Dt\,t^2\left[\sum_{l=0}^{(K+\theta_0-1)/2}+\sum_{l=0}^{(K-\theta_0-1)/2}\right]\times\begin{bmatrix} K-1 \\ l \end{bmatrix}[H(t)]^{K-l-1}[1-H(t)]^l . \tag{56}$$

For example, if $\theta_0=(K-1)$ this implies

$$2\alpha_c^{-1}=\frac{K}{2}+K\int_0^\infty Dt\,t^2[H(t)]^{K-1} . \tag{57}$$

For positive output, $\eta=+1$, all hidden units must be $+1$ to overcome the threshold. In this case the capacity is that of a single perceptron $2/K$. If the output is negative, $\eta=-1$, it is sufficient to have one $\tau_l=-1$. Hence the critical perceptron only has to be modified if it has the smallest value of the noise.

The general expression for $\alpha_c(\theta_0)$ [Eq. (54)] is an even function of $\theta_0$, as one would expect for random unbiased output $\eta$. To show that it has a maximum at $\theta_0=0$ we calculate

$$\alpha_c^{-1}(2)-\alpha_c^{-1}(0)=\frac{K}{2}\int_0^\infty Dt\, t^2\left[[H(t)]^{(K-3)/2}[1-H(t)]^{(K+1)/2}\frac{(K-1)!}{\left[\frac{K+1}{2}\right]!\left[\frac{K-3}{2}\right]!}\right.$$
$$\left.-[H(t)]^{(K-1)/2}[1-H(t)]^{(K-1)/2}\frac{(K-1)!}{\left[\frac{K-1}{2}\right]!\left[\frac{K-1}{2}\right]!}\right]$$
$$=\sqrt{2\pi}\begin{bmatrix}K\\ \frac{K-1}{2}\end{bmatrix}\int_0^\infty dt\, t[H(t)]^{(K-1)/2}[1-H(t)]^{(K+1)/2}>0\ . \tag{58}$$

We conclude that thresholds cannot increase the capacity for random, unbiased input-output relations. Carefully chosen thresholds should, however, improve the storage abilities of the network if the patterns are drawn from a biased probability distribution. Although we have only studied the replica-symmetric theory these qualitative conclusions should remain true also in the context of replica-symmetry breaking. Note, however, that already in the single-layer perceptron replica symmetry is broken for $\kappa<0$ [4] [cf. (55)].

### C. Adaptive hidden-to-output couplings

It is of special interest to generalize our results to two-layer networks for which the couplings $w_l$ from the hidden units to the output are not fixed but can be adapted in the process of learning. The fractional volume in the phase space of interactions stabilizing all patterns is then given by [cf. (3)–(5)]

$$V=\frac{M}{D}\ , \tag{59}$$

where

$$M=\int_{-\infty}^{\infty}\prod_{l=1}^{K}\prod_{j(l)}dJ_{lj(l)}\int_0^\infty\prod_{l=1}^{K}dw_l\prod_{l=1}^{K}\delta\left[\sum_{j(l)}J_{lj(l)}^2-\frac{N}{K}\right]\delta\left[\sum_{l=1}^{K}w_l^2-K\right]\prod_\mu\Theta\left\{\eta^\mu\sum_{l=1}^{K}w_l\mathrm{sgn}\left[\left[\frac{K}{N}\right]^{1/2}\sum_{j(l)}J_{lj(l)}\xi_{j(l)}^\mu\right]\right\} \tag{60}$$

and

$$D=\int_{-\infty}^{\infty}\prod_{l=1}^{K}\prod_{j(l)}dJ_{lj(l)}\int_0^\infty\prod_{l=1}^{K}dw_l\prod_{l=1}^{K}\delta\left[\sum_{j(l)}J_{lj(l)}^2-\frac{N}{K}\right]\delta\left[\sum_{l=1}^{K}w_l^2-K\right]\ . \tag{61}$$

Since the $w_l$ do not couple directly to the disorder variables $\xi_{j(l)}^\mu$ the average can be performed as in Sec. III and we get similar to (9)

$$\langle\langle V^n\rangle\rangle=\int\prod_{l,\alpha}\frac{dE_l^\alpha}{4\pi}\prod_{l,\alpha<\beta}\frac{dF_l^{\alpha\beta}dq_l^{\alpha\beta}}{2\pi K/N}\prod_{l,\alpha}dw_l^\alpha\prod_\alpha\delta\left[\sum_{l=1}^{K}(w_l^\alpha)^2-K\right]$$
$$\times\exp\left[N\left[\frac{1}{2K}\sum_{l,\alpha}E_l^\alpha-\frac{1}{2K}\sum_{l,\alpha<\beta}F_l^{\alpha\beta}q_l^{\alpha\beta}+\frac{1}{K}G_2(E_l^\alpha,F_l^{\alpha\beta})+\alpha G_1(q_l^{\alpha\beta},w_l^\alpha)-\frac{n}{2}[1+\ln(2\pi)]\right]\right]\ , \tag{62}$$

where $G_2(E_l^\alpha,F_l^{\alpha\beta})$ is again given by (10) and

$$G_1(q_l^{\alpha\beta},w_l^\alpha)=\ln\int\prod_{l,\alpha}\frac{d\lambda_l^\alpha dx_l^\alpha}{2\pi}\exp\left[i\sum_{l,\alpha}x_l^\alpha\lambda_l^\alpha-\tfrac{1}{2}\sum_{l,\alpha}(x_l^\alpha)^2-\tfrac{1}{2}\sum_{\substack{l,\alpha,\beta\\ \alpha\neq\beta}}x_l^\alpha x_l^\beta q_l^{\alpha\beta}\right]\times\prod_\alpha\Theta\left[\sum_l w_l^\alpha\mathrm{sgn}\lambda_l^\alpha\right]\ . \tag{63}$$

(63) differs from (11) just by the argument of the $\Theta$ function.

From (62) we infer that the $w_l^\alpha$ integral can be performed by the saddle-point method. Since after averaging all hidden units are equivalent the saddle-point values of the $w_l^\alpha$ should not depend on $l$ [analogous to (14) for the other integration variables]. Note that the independence of $l$ of the saddle-point values holds for the replica-symmetric solution and the solution with broken replica symmetry. Then the $\delta$ function in (62) requires $w_l^\alpha=1$ for all $l$ and $\alpha$ and the storage properties are the same as for the committee tree.

The reason for this at first sight surprising result is the special architecture of our network. The number of in-

puts of the hidden units is $O(N)$ whereas the output unit receives input from $O(1)$ hidden units only. It is impossible to adapt the $K$ variables $w_l$ to the correlations of infinitely many patterns $\{\xi^{\mu}_{j(l)}\}$. In other words, the committee tree has $N$ adjustable parameters, a tree with a variable hidden-to-output relation has $N+K$ and to leading order in $N$ the same number of input-output mappings can be implemented. Consequently for $K=O(1)$ the committee machine has the highest storage capacity of all two-layer networks of the discussed type.

Many other generalizations and extensions are of interest. We just mention two:

*Other pattern statistics.* For example, the tree architecture seems not well suited for random uncorrelated patterns. One may hope to gain considerably in capacity, if patterns are used, which are correlated within a receptive field [i.e., within of $O(N/K)$ bits] but uncorrelated for different receptive fields. In general the question what architecture to choose for a given patterns statistics or vice versa which class of patterns is easily learned by a given architecture is not well understood.

*Other architectures.* An example is a committee machine with a large number of hidden units, each connected to a finite number of inputs only. In this case a replica analysis is more complicated since higher cumulant order parameters have to be introduced similar to the case of diluted spin glasses [30]. Another architecture, which presumably can be treated analytically, is characterized by random dilution of the fully connected net, so that the hidden units have partially overlapping receptive fields. Finally one may want to consider hidden units, whose receptive fields are defined in one- or two-dimensional space and which can be used to handle geometrical objects.

*Note added.* After completion of this work we received a copy of unpublished work by E. Barkai, D. Hansel, and H. Sompolinsky in which similar questions are discussed.

## ACKNOWLEDGMENTS

Simulations were performed on a Cray Y/MP of the Forschungszentrum in Jülich, and on Ultrix work stations of the Institut für Numerische Mathematik, Göttingen.

## APPENDIX A

In this appendix we perform explicitly the limit $q\to 1$ in the replica-symmetric expression for $\alpha_c$ (17) to arrive at (18). It is convenient to first rewrite the $t_l$ integrals using once more (14) in the form

$$\operatorname*{Tr}_{\{\eta_l=\pm 1\}}\int_0^{\infty}\prod_l Dt_l \ln\left[\operatorname*{Tr}_{\{\tau_l=\pm 1\}}\Theta\left[\sum_l \tau_l\right]\prod_l H(Qt_l\eta_l\tau_l)\right]. \tag{A1}$$

For $q\to 1$ we have $Q=(q/1-q)^{1/2}\to\infty$ and, depending on the sign of $\eta_l\tau_l$, the $H$ functions either tend to 1 or zero. Consider first an $\eta_l$ configuration with $\sum_l \eta_l<0$, i.e., with more $\eta_l=-1$ than $\eta_l=+1$. Then $\tau_l=-\eta_l$ is an allowed $\tau_l$ configuration (since $\sum_l \tau_l>0$) which dominates the $\tau_l$ trace because all $H$ functions have negative arguments and tend to 1 for $Q\to\infty$. Taking the logarithm one realizes that these $\eta_l$ configurations give negligible contributions to the $\eta_l$ trace. For all $\eta_l$ configurations with $\sum_l \eta_l>0$ all allowed $\tau_l$ configurations will produce some $H$ functions with positive arguments. These tend to zero for $Q\to\infty$ giving rise to contributions to the $\eta_l$ trace of large absolute value. Now

$$H(x)\sim\frac{1}{(2\pi)^{1/2}x}\exp\left[-\frac{x^2}{2}\right]\quad\text{for } x\to\infty \tag{A2}$$

and hence

$$\frac{H(Qt_l)}{H(Qt_m)}\sim\frac{t_m}{t_l}\exp\left[-\frac{Q^2}{2}(t_l^2-t_m^2)\right]. \tag{A3}$$

Hence for every $\eta_l$ configuration with $\sum_l \eta_l>0$ the $\tau_l$ trace is for $Q\to\infty$ dominated by a single term. This term has $\tau_l=-1$ for as many $l$ as possible, i.e., for $(K-1)/2$. These $\tau_l=-1$ are distributed such that $\tau_l=-1$ only if $\eta_l=+1$ and that those $l$ with $\tau_l=\eta_l=+1$ correspond to the smallest $t_l$. In order to extract the dominant term from the $\tau_l$ trace we have therefore to know the *relative order* of the integration variables $t_l$. It is convenient to fix this order by using

$$\int_0^{\infty}\prod_l Dt_l f(t_l)=K!\int_0^{\infty}Dt_1\int_{t_1}^{\infty}Dt_2\cdots\int_{t_{K-1}}^{\infty}Dt_K f(t_l), \tag{A4}$$

which holds for all integrands $f(t_l)$ which are symmetric with respect to permutations of the $t_l$. So we can write (17) in the form

$$\alpha_c^{-1}=-\lim_{q\to 1}2(1-q)\operatorname*{Tr}_{\{\eta_l=\pm 1\}}\Theta\left[\sum_l \eta_l\right]K!\int_0^{\infty}Dt_1\cdots\int_{t_{K-1}}^{\infty}Dt_K\ln\operatorname*{Tr}_{\{\tau_l=\pm 1\}}\Theta\left[\sum_l \tau_l\right]\prod_l H(Qt_l\tau_l\eta_l). \tag{A5}$$

Now we can calculate the asymptotic behavior of all allowed contributions to the $\eta_l$ trace one after another. It is useful to introduce the abbreviation $K'=(K+1)/2$. We start with $\eta_l=+1$ for all $l$: in this case the integral in (A5) is given by

$$K!\int_0^\infty Dt_1\cdots\int_{t_{K-1}}^\infty Dt_K \ln[H(Qt_1)H(Qt_2)\cdots H(Qt_{K'})]$$
$$=K!\int_0^\infty Dt_1 \ln H(Qt_1)\frac{1}{(K-1)!}[H(t_1)]^{K-1}+K!\int_0^\infty Dt_1\int_{t_1}^\infty Dt_2 \ln H(Qt_2)\frac{1}{(K-2)!}[H(t_2)]^{K-2}+\cdots$$
$$+K!\int_0^\infty Dt_1\int_{t_1}^\infty Dt_2\cdots\int_{t_{K'-1}}^\infty Dt_{K'}\ln H(Qt_{K'})\frac{1}{(K-K')!}[H(t_{K'})]^{K-K'}$$
$$=K!\int_0^\infty Dt_1 \ln H(Qt_1)\frac{1}{(K-1)!}[H(t_1)]^{K-1}+K!\int_0^\infty Dt_2 \ln H(Qt_2)\mathrm{erf}(t_2)\frac{1}{(K-2)!}[H(t_2)]^{K-2}+\cdots$$
$$+K!\int_0^\infty Dt_{K'}\ln H(Qt_{K'})\frac{1}{(K'-1)!}[\mathrm{erf}(t_{K'})]^{K'-1}\frac{1}{(K-K')!}[H(t_{K'})]^{K-K'}, \tag{A6}$$

where

$$\mathrm{erf}(x)=\int_0^x Dt=\tfrac{1}{2}-H(x). \tag{A7}$$

Therefore the contribution of this $\eta_l$ configuration to the $\eta_l$ trace is given by

$$\int_0^\infty Dt \ln H(Qt)\left[\sum_{k=0}^{K'-1}\frac{K!}{(K-k-1)!k!}[H(t)]^{K-k-1}[\mathrm{erf}(t)]^k\right]. \tag{A8}$$

Next we consider $\eta_l$ configurations with all but one $\eta_l=+1$: There are $K$ equivalent configurations of this type, so without loss of generality we can take $\eta_l=+1$ for $l=1,\ldots,K-1$ and $\eta_K=-1$. The dominant term in the $\tau_l$ trace has $\tau_K=+1$ hence $H(Q\tau_K\eta_K t_K)\to 1$ and the integral over $t_K$ gives just a factor $\frac{1}{2}$ in (A1). Moreover we have one $H$ function with positive argument less than in the previous case. The remaining integrals are handled as in (A6) and we find for this particular $\eta_l$ configuration

$$\tfrac{1}{2}(K-1)!\int_0^\infty Dt_1\cdots\int_{t_{K-2}}^\infty Dt_{K-1}\ln[H(Qt_1)H(Qt_2)\cdots H(Qt_{K'-1})]$$
$$=\tfrac{1}{2}\int_0^\infty Dt \ln H(Qt)\left[\sum_{k=0}^{K'-2}\frac{K!}{(K-k-2)!k!}[H(t)]^{K-k-2}[\mathrm{erf}(t)]^k\right]. \tag{A9}$$

In order to get the contribution from all $\eta_l$ configurations with a single $\eta_l=-1$ we have to multiply (A9) by $K$. It is straightforward to obtain the leading term of the integral in (A5) for an $\eta_l$ configuration with $m$, $\eta_l=-1$, and $(K-m)$, $\eta_l=+1$: In this case we find

$$2^{-m}\binom{K}{m}\int_0^\infty Dt \ln H(Qt)\left[\sum_{k=0}^{K'-m-1}\frac{(K-m)!}{(K-k-m-1)!k!}[H(t)]^{K-k-m-1}[\mathrm{erf}(t)]^k\right]. \tag{A10}$$

For $m=0$ and $m=1$ we recover (A8) and (A9) multiplied by $K$, respectively.

The largest possible $m$ is $m=K'-1$ since otherwise $\sum_l\eta_l<0$. Therefore we get for (A5)

$$\alpha_c^{-1}=-\lim_{q\to 1}2(1-q)\int_0^\infty Dt \ln H(Qt)\left[\sum_{m=0}^{K'-1}\frac{K!}{m!}2^{-m}\left[\sum_{k=0}^{K'-m-1}\frac{1}{(K-m-1-k)!k!}[H(t)]^{K-m-k-1}[\mathrm{erf}(t)]^k\right]\right]. \tag{A11}$$

It is easy now to perform the limit $q\to 1$ using (A2). Moreover introducing $n=(m+k)$ we can simplify (A11) to

$$\alpha_c^{-1}=K\int_0^\infty Dt\, t^2\sum_{n=0}^{K'-1}\frac{(K-1)!}{(K-1-n)!n!}[H(t)]^{K-n-1}\sum_{k=0}^{n}\frac{n!}{(n-k)!k!}2^{k-n}[\mathrm{erf}(t)]^k, \tag{A12}$$

which is the same as (18).

## APPENDIX B

In this appendix we determine the critical capacity $\alpha_c$ for a committee machine with $K=3$ hidden units using (26) and (25). The main problem is to accomplish the limit $m\to 0$ in (25) explicitly. This is turn requires us to find the dominating terms in the trace over $\tau_l$ for $m\to 0$. To this end we first rewrite (25) in the form

$$g(q_0,c)=\lim_{m\to 0}\int\prod_l Dz_l \ln\left\{\mathrm{Tr}_{\{\eta_l=\pm 1\}}\int_0^\infty\prod_l dt_l f(t_l,z_l,\eta_l)\left[\mathrm{Tr}_{\{\tau_l=\pm 1\}}\Theta\left[\sum_l\tau_l\right]\prod_l H\left[\frac{t_l\tau_l\eta_l c^{1/2}}{m^{1/2}}\right]\right]^m\right\}, \tag{B1}$$

where

$$f(t_l,z_l,\eta_l)=[2\pi(1-q_0)]^{-1/2}\exp\left[-\frac{(t_l-\eta_l q_0^{1/2}z_l)^2}{2(1-q_0)}\right]. \tag{B2}$$

As in the replica-symmetric case of Appendix A there is for all $\eta_l$ configurations and all values of the integration variables $t_l$ a unique $\tau_l$ configuration, which dominates the trace over the $\tau_l$. To find it we have to fix the relative order of the $t_l$. This is now, however, more difficult because the integrand of the $t_l$ integral is no longer a symmetric function of the $t_l$ due to the appearance of the functions $f(t_l,z_l,\eta_l)$. Instead of (A4) we must therefore use the identity

$$\int_0^\infty \prod_l dt_l F(t_l)=\int_0^\infty dt_1\int_{t_1}^\infty dt_2\cdots\int_{t_{K-1}}^\infty dt_K F(t_l)+(\text{perm.}), \tag{B3}$$

where perm. stands for all analogous terms with permutations of the indices $1,\ldots,K$. Moreover *all* $\eta_l$ configurations contribute in (B1) and not only half of them as in the replica-symmetric case. These complications are the reason why we restrict ourselves in the following to $K=3$. Let us then calculate the $t_l$ integrals in (B1) for all eight possible $\eta_l$ configurations and take the limit $m\to 0$. We start with $\eta_1=\eta_2=\eta_3=+1$: The dominating term in the $\tau_l$ trace has $\tau_l=-1$ for the index $l$ which corresponds to the largest $t_l$ and $\tau_k=+1$ for $k\neq l$. Hence we get

$$\begin{aligned}
&\int_0^\infty dt_1dt_2dt_3 f(t_1,z_1,+1)f(t_2,z_2,+1)f(t_3,z_3,+1)\left[\operatorname*{Tr}_{\{\tau_l=\pm1\}}\Theta\left(\sum_l\tau_l\right)\prod_l H\left(\frac{t_l\tau_l\eta_l c^{1/2}}{m^{1/2}}\right)\right]^m\\
&=\int_0^\infty dt_1 f(t_1,z_1,+1)\int_{t_1}^\infty dt_2 f(t_2,z_2,+1)\int_{t_2}^\infty dt_3 f(t_3,z_3,+1)H^m\left(\frac{t_1c^{1/2}}{m^{1/2}}\right)H^m\left(\frac{t_2c^{1/2}}{m^{1/2}}\right)+(\text{perm.})\\
&=\int_0^\infty dt_1 f(t_1,z_1,+1)e^{-ct_1^2/2}\int_{t_1}^\infty dt_2 f(t_2,z_2,+1)e^{-ct_2^2/2}\int_{t_2}^\infty dt_3 f(t_3,z_3,+1)+(\text{perm.}),
\end{aligned} \tag{B4}$$

where in the last step we performed the limit $m\to 0$ using (A2).

Next consider the $\eta_l$ configuration $\eta_1=-1$, $\eta_2=\eta_3=+1$. Now the dominating term in the $\tau_l$ trace has $\tau_1=+1$ and $\tau_2=+1$, $\tau_3=-1$ if $t_2<t_3$ or $\tau_2=-1$, $\tau_3=+1$ if $t_2>t_3$. Hence we get for the $t_l$ integrals in this case

$$\begin{aligned}
&\int_0^\infty dt_1 f(t_1,z_1,-1)\left[\int_0^\infty dt_2 f(t_2,z_2,+1)e^{-ct_2^2/2}\int_{t_2}^\infty dt_3 f(t_3,z_3,+1)\right.\\
&\qquad\left.+\int_0^\infty dt_3 f(t_3,z_3,+1)e^{-ct_3^2/2}\int_{t_3}^\infty dt_2 f(t_2,z_2,+1)\right].
\end{aligned} \tag{B5}$$

Similar expressions result for the other $\eta_l$ configurations with exactly one $\eta_l=-1$ and their joint contribution to the $\eta_l$ trace is therefore

$$\int_0^\infty dt_1 f(t_1,z_1,-1)\int_0^\infty dt_2 f(t_2,z_2,+1)e^{-ct_2^2/2}\int_{t_2}^\infty dt_3 f(t_3,z_3,+1)+(\text{perm.}). \tag{B6}$$

For the remaining $\eta_l$ configurations the dominant term in the $\tau_l$ trace tends always to 1 for $m\to 0$. The relative order of the $t_l$ is no longer relevant and therefore the $t_l$ integrals factorize. Observing finally that

$$\int_x^\infty dt\, f(t,z,\eta)=H\left(\frac{x-\eta q_0^{1/2}z}{(1-q_0)^{1/2}}\right) \tag{B7}$$

and introducing $Q_0=[q_0/(1-q_0)]^{1/2}$ we find from (B1), (B4), and (B6)

$$\begin{aligned}
g(q_0,c)=\int Dz_1Dz_2Dz_3\ln\Bigg\{&\Bigg[\int_0^\infty dt_1 f(t_1,z_1,+1)e^{-ct_1^2/2}\\
&\times\int_{t_1}^\infty dt_2 f(t_2,z_2,+1)e^{-ct_2^2/2}H\left(\frac{t_2}{(1-q_0)^{1/2}}-Q_0t_3\right)+(\text{perm.})\Bigg]\\
&+\left[H(Q_0z_1)\int_0^\infty dt_2 f(t_2,z_2,+1)e^{-ct_2^2/2}H\left(\frac{t_2}{(1-q_0)^{1/2}}-Q_0z_3\right)+(\text{perm.})\right]\\
&+[H(Q_0z_1)H(Q_0z_2)H(-Q_0z_3)+(\text{perm.})]+[H(Q_0z_1)H(Q_0z_2)H(Q_0z_3)]\Bigg\}.
\end{aligned} \tag{B8}$$

The $t_1$ integral in the first term under the logarithm can still be performed analytically. The remaining four integrals as well as the minimization in $c$ and $q_0$ have to be done numerically. The results are discussed in Sec. IV.

[1] See, for example, D. Amit, *Modelling Brain Function* (Cambridge University Press, New York, 1989); J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Reading, MA, 1991); *Physics of Neural Networks,* edited by J. L. van Hemmen, E. Domany, and K. Schulten (Springer, Berlin, 1991).

[2] See, for example, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* edited by J. McClelland, D. E. Rumelhart, and the PDP Research Group (Bradford Books/ MIT, Cambridge, MA, 1986).

[3] T. M. Cover, IEEE Trans. Electron. Comput. **EC-14**, 326 (1965).

[4] E. Gardner, J. Phys. A **21**, 257 (1988); Europhys. Lett. **4**, 481 (1987).

[5] F. Rosenblatt, *Principles of Neurodynamics* (Spartan, New York, 1962).

[6] M. Minsky and S. Papert, *Perceptrons* (MIT, Cambridge, MA, 1969).

[7] A. N. Kolmogorov, Dokl. Akad. Nauk. USSR **114**, 953 (1957); R. Hecht-Nielsen, Proceedings of the IEEE First Annual Conference on Neural Networks, edited by M. Cardill and C. Butler, San Diego, 1987 (unpublished).

[8] E. B. Baum, J. Complex. **4**, 193 (1988).

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

[10] F. Barkai, D. Hansel, and I. Kanter, Phys. Rev. Lett. **65**, 2312 (1990).

[11] E. Barkai and I. Kanter, Europhys. Lett. **14**, 107 (1991).

[12] P. Rujan and M. Marchand, Complex Systems **3**, 229 (1989).

[13] M. Mézard and J. P. Nadal, J. Phys. A **22**, 2191 (1989).

[14] M. Biehl and M. Opper, Phys. Rev. A **44**, 6888 (1991).

[15] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Ann. Phys. (N.Y.) **173**, 30 (1987).

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

[17] H. Sompolinsky and N. Tishby, Europhys. Lett. **13**, 567 (1990).

[18] M. Mézard and S. Patarnello found the same result for the parity machine for *all K* (unpublished).

[19] E. Gardner and B. Derrida, J. Phys. A **21**, 271 (1988).

[20] G. Parisi, J. Phys. A **13**, 1101 (1980).

[21] W. Krauth and M. Mézard, J. Phys. (Paris) **50**, 3057 (1989).

[22] D. J. Gross and M. Mézard, Nucl. Phys. **240**, 431 (1984).

[23] N. J. Nilsson, *Learning Machines* (McGraw-Hill, New York, 1965).

[24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Nature **323**, 533 (1986).

[25] B. Widrow and M. E. Hoff, WESCON Convention Report IV, p. 96, 1960 (unpublished).

[26] S. Diederich and M. Opper, Phys. Rev. Lett. **58**, 949 (1987).

[27] W. Kinzel and M. Opper, in *Physics of Neutral Networks,* edited by J. L. van Hemmen, E. Domany, and K. Schulten (Springer, Berlin, 1991).

[28] J. K. Anlauf and M. Biehl, Europhys. Lett. **10**, 687 (1989); **11**, 387 (1990).

[29] R. Fletcher, *Practical Methods of Optimization* (Wiley, Chichester, 1987).

[30] L. Viana and A. J. Bray, J. Phys. C **18**, 3037 (1985); M. Mézard and G. Parisi, Europhys. Lett. **3**, 1067 (1987); I. Kanter and H. Sompolinsky; Phys. Rev. Lett **58**, 164 (1987).