Generalization and capacity of extensively large two-layered perceptrons

Michal Rosen-Zvi,[1] Andreas Engel,[2] and Ido Kanter[1]
[1]*Minerva Center and Department of Physics, Bar-Ilan University, Ramat-Gan, 52900 Israel*
[2]*Institut für Theoretische Physik, Otto-von-Guericke Universität, PSF 4120, 39016 Magdeburg, Germany*
(Received 25 June 2002; published 27 September 2002)

The generalization ability and storage capacity of a treelike two-layered neural network with a number of hidden units scaling as the input dimension is examined. The mapping from the input to the hidden layer is via Boolean functions; the mapping from the hidden layer to the output is done by a perceptron. The analysis is within the replica framework where an order parameter characterizing the overlap between two networks in the combined space of Boolean functions and hidden-to-output couplings is introduced. The maximal capacity of such networks is found to scale linearly with the logarithm of the number of Boolean functions per hidden unit. The generalization process exhibits a first-order phase transition from poor to perfect learning for the case of discrete hidden-to-output couplings. The critical number of examples per input dimension, α_c, at which the transition occurs, again scales linearly with the logarithm of the number of Boolean functions. In the case of continuous hidden-to-output couplings, the generalization error decreases according to the same power law as for the perceptron, with the prefactor being different.

DOI: 10.1103/PhysRevE.66.036138 PACS number(s): 84.35.+i, 07.05.Mh, 89.70.+c

I. INTRODUCTION

Since the early 1960s, the perceptron, which is the basic element of feed-forward neural networks, was extensively studied as a learning unit with memory capabilities. It was shown that such a feed-forward unit with N input components and an output calculated by the input and the weight vectors can store examples and can learn from them and generalize [1]. It attracted attention in the statistical mechanics field only in the late 1980s. Gardner blazed the trail in her seminal work [2,3], in which she introduced means of quantifying the abilities of the perceptron. The origin of her tool box was models of spin glass. She based her calculations upon the entropy of the network and used the replica trick, in order to overcome the difficulties in calculating the average over the quenched randomness.

After a thorough analysis of the perceptron, the multilayer architecture took center stage [4–12]. The simplest multilayer network (MLN) is composed of two layers, each being perceptronlike. Such networks can be used for more complicated tasks. The number of Boolean mappings that can be implemented in a MLN with binary output is much larger than the number that can be implemented in a binary *perceptron*. It was shown [13] that any mapping can be stored in a large enough MLN and that an unbounded hidden layer only will suffice. Most of the networks that have been studied analytically contain an N-dimensional input vector, where N tends to infinity. The input is connected via a hidden layer with K nodes to the output. The number of nodes is finite or large but even when K is taken to be infinitely large it does not scale with N, which is much larger [4–9,11] (apart from the unique case studied in [10]). It is intriguing to extend the analytical study of MLNs to the case when the number of hidden units scales with N. In all cases studied, the maximal number of patterns that can be stored, divided by the input dimension α_c, becomes larger as the number of hidden layers K grows. We are interested in the case of infinitely large K, when K scales with N and both layers are adaptive. The questions raised in such a model are the following. Can we develop analytical tools to solve such extensively large MLNs? What is the nature of the order parameter in this limit? How to combine into one parameter the quantities of both layers? Will the maximal capacity per weight, α_c, continue to grow in this limit?

It was found that large machines with $K \to \infty$ but when K does not scale with N can generalize. The generalization error ϵ_g that measures the discrepancy between the two machines, the ruler— the teacher — and the student in an explored example, decreases to zero with the same decay typical of the perceptron, independent of K, in the tree parity machine [7] and also in the tree committee machine [11]. It is not clear what happens when K scales with N. Is the generalization decrease similar to the perceptron decrease? What are the methods used to calculate analytically the learning curve? Most of the answers to the questions above were recently introduced in Ref. [12]. In this paper we present a detailed description of the analysis of such extensively large MLNs and include a variety of cases (some of them were omitted in Ref. [12]). We introduce simulation results and include an expanded discussion of the results.

We analyze the $LN{:}N{:}1$ network (see Fig. 1) from several viewpoints. The capacity of the network is examined in the framework of replica calculations [2,3], where an order parameter that incorporates both layers is introduced. It is shown that the order parameter contains the essential information concerning the network performance. Bounds are derived using combinatorial geometry [14,15]. The learning ability of the network is also under discussion, where the replica calculations are used [16–18]. Again, the order parameter involving the capacity calculations is found to be the cornerstone in the generalization analysis. Simulations including exact enumerations are performed and are found to support the results.

Our main finding concerning the capacity is that the maximal capacity of the network, divided by the input dimension α_c, scales with the logarithm of the number of Boolean

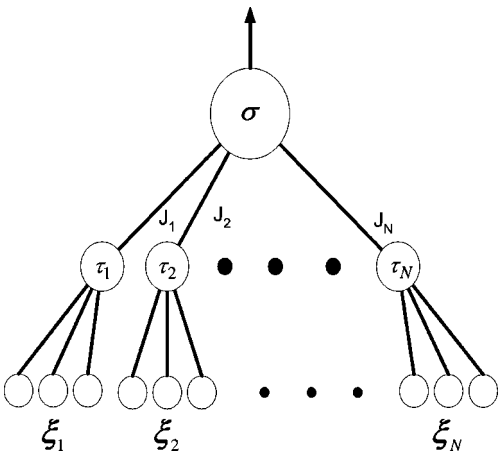

FIG. 1. A two-layered perceptron, $3N{:}N{:}1$.

functions N_B assigned to each unit. The maximal capacity per input dimension α_c was analytically derived for the case of binary hidden-to-output couplings and approximated, using the replica symmetry assumption, in the case of continuous hidden-to-output couplings. In both cases $\alpha_c \sim \ln(N_B)$. We carried detailed simulations and numerical results in the case of $L=3$, the general case when all antisymmetric Boolean functions are admissible ($N_B=16$), and the case of perceptron mapping ($N_B=14$). The hidden-to-output couplings were taken to be either continuous or discrete. We found that α_c is within the analytical bounds and the results are supported by simulations.

The generalization ability in the case of a realizable rule, teacher and student with the same architecture, was derived analytically. Although the student in this case studies from a teacher, which is much more complicated than the perceptron, we found similarities between learning in the perceptron and learning in the case of $3N{:}N{:}1$. In the case of binary hidden-to-output couplings, a phase transition occurs from poor to perfect generalization. Again, the logarithm of the number of Boolean functions determines α_c, the number of examples per input dimension in which the transition occurs. In the case of continuous hidden-to-output couplings, the generalization error obeys the same power law as in the simple perceptron, where the prefactor is inversely proportional to L.

The paper is organized as follows: The architecture is introduced in Sec. II. In Sec. III we define the order parameter that enables calculations in a variety of cases. The storage capacity in the case of discrete and continuous hidden-to-output couplings is discussed in Sec. IV. In Sec. V the generalization ability in all those cases is studied.

II. THE ARCHITECTURE $LN{:}N{:}1$

The architecture of the two-layer feed-forward neural network, $LN{:}N{:}1$, discussed in this paper consists of N binary units $\tau_i = \pm 1$ in the intermediate or so-called hidden layer. Each of these hidden units receives input from a separate subset $\boldsymbol{\xi}_i = \{\xi_{ij}, j=1,\ldots,L\}$ of L units of the input layer. Accordingly, the input layer is of size LN and the receptive fields of the hidden units are nonoverlapping (see Fig. 1). Given the activity in the input layer the states of the hidden units are determined by Boolean functions B_i mapping the L-dimensional binary input $\boldsymbol{\xi}_i$ to a binary output $\tau_i = B_i(\boldsymbol{\xi}_i)$.

The output is a single binary unit σ given by

$$\sigma = \mathrm{sgn}\left(\sum_{i=1}^{N} J_i \tau_i \right). \tag{1}$$

Here $\mathbf{J}$ is the N-dimensional hidden-to-output weight vector.

There are $N_B = 2^{2^L}$ different Boolean functions with L inputs. To keep the connection with more traditional architectures of neural networks which use perceptronlike mappings also between input layer and hidden units, we restrict ourselves to odd functions satisfying $B(-\boldsymbol{\xi}) = -B(\boldsymbol{\xi})$. There are $N_B = 2^{2^{L-1}}$ different odd Boolean functions of L inputs. Only a minute fraction of these, e^{L^2}, can be implemented by a perceptron (see [19]), i.e., for these, there exists an L-dimensional weight vector $\mathbf{W}$ such that

$$B(\boldsymbol{\xi}) = \mathrm{sgn}(\mathbf{W} \cdot \boldsymbol{\xi}). \tag{2}$$

When possible we will give results both for the case when all antisymmetric Boolean functions are available and for the more restricted case when only those implementable by coupling vectors $\mathbf{W}$ may be used.

In a learning process in networks of the proposed architecture both the Boolean functions B_i and the couplings J_i are adapted in order to perform the desired input-output mapping. We will consider in this paper the two standard problems, the capacity and the generalization problem. In both cases the input components are chosen independently at random, $\xi_{ij} = \pm 1$ with equal probability. In the capacity problem the corresponding outputs are generated at random as well and the question is how many of such random input-output mappings one may typically implement by choosing appropriate Boolean functions B_i and values J_i. The threshold is proportional to the dimension of the input space and will be written as $\alpha_c LN$. In the generalization problem one considers two networks of identical architecture. One of these (the teacher) is designed at random choosing Boolean functions B_i^T and couplings J_i^T according to a given probability measure. The second (the student) tries to imitate the teacher as well as possible on the basis of a training set consisting of αLN random inputs together with their classification according to the teacher. The aim is to calculate the generalization error $\epsilon_g(\alpha)$ defined as the probability that the teacher and student disagree on a new random example.

Most of the detailed numerical results discussed below will refer to the case $L=3$. The 16 possible antisymmetric Boolean functions for this case are presented in Table I. They comprise two groups which are mirror images of each other. We therefore present in Table I only one group — eight Boolean functions. Seven out of the eight Boolean functions can be realized using Eq. (2). The last mapping in Table I is called parity since it is simply the parity of the inputs. It is the well known problem where the mapping cannot be implemented by a perceptron.

TABLE I. Half of the possible antisymmetric Boolean functions in the case of $L=3$. The other eight functions are exactly the opposite, ${}^{8+j}B(\boldsymbol{\xi})=-{}^{j}B(\boldsymbol{\xi})$, $j=1,2,\ldots,8$.

Input	Perceptron [Eq. (2)]							Parity
	${}^{1}B$	${}^{2}B$	${}^{3}B$	${}^{4}B$	${}^{5}B$	${}^{6}B$	${}^{7}B$	${}^{8}B$
$\boldsymbol{\xi}_1=+++$	+	+	+	+	+	+	+	+
$\boldsymbol{\xi}_2=++-$	+	+	+	−	+	−	−	−
$\boldsymbol{\xi}_3=+-+$	+	+	−	+	−	+	−	−
$\boldsymbol{\xi}_4=+--$	+	−	+	+	−	−	−	+
$\boldsymbol{\xi}_5=---$	−	−	−	−	−	−	−	−
$\boldsymbol{\xi}_6=--+$	−	−	−	+	−	+	+	+
$\boldsymbol{\xi}_7=-+-$	−	−	+	−	+	−	+	+
$\boldsymbol{\xi}_8=-++$	−	+	−	−	+	+	+	−

III. THE ORDER PARAMETER

Statistical mechanics analysis of the considered network builds on standard techniques [1]. The central quantity is the entropy averaged over the distribution of the inputs,

$$s=\lim_{N\to\infty}\frac{1}{N}\left\langle\left\langle \ln\int d\mu(\mathbf{J})\mathrm{Tr}_{\{B_i\}} \times\prod_{\mu=1}^{\alpha LN}\theta\left(\sum_i J_iB_i(\boldsymbol{\xi}_i^{\mu})\right)\right\rangle\right\rangle_{\{\boldsymbol{\xi}_i^{\mu}\}}, \tag{3}$$

where $d\mu(\mathbf{J})$ is the proper measure in the space of couplings $\mathbf{J}$ and the trace runs over the set of available Boolean functions. The replica trick

$$\langle\langle\ln\Omega\rangle\rangle=\lim_{n\to 0}\frac{\langle\langle\Omega^n\rangle\rangle-1}{n} \tag{4}$$

with

$$\langle\langle\Omega^n\rangle\rangle=\lim_{N\to\infty}\int\prod_{a=1}^{n}d\mu(J^a)\mathrm{Tr}_{\{B_i^a\}} \times\left\langle\left\langle\prod_{\mu=1}^{\alpha LN}\prod_{a=1}^{n}\theta\left(\sum_i J_i^aB_i^a(\boldsymbol{\xi}_i^{\mu})\right)\right\rangle\right\rangle, \tag{5}$$

is used to perform the quenched average over the input distribution and gives rise to the order parameter

$$q^{ab}=\frac{1}{N}\sum_{i=1}^{N}J_i^aJ_i^b\langle\langle B_i^a(\boldsymbol{\xi})B_i^b(\boldsymbol{\xi})\rangle\rangle_{\boldsymbol{\xi}}. \tag{6}$$

Here the average $\langle\langle f(\boldsymbol{\xi})\rangle\rangle_{\boldsymbol{\xi}}$ runs just over the 2^L different configurations of a *single* input vector $\boldsymbol{\xi}$ of length L. The limit $n\to 0$ in Eq. (4) is appropriate for the capacity problem whereas the generalization error can be obtained by performing the limit $n\to 1$ [1,17]. We will always assume replica symmetry, $q^{ab}=q$ for all $a\neq b$. This is known to be reliable for the generalization problem, whereas it represents a mere approximation in the case of the capacity problem.

The explicit calculations are given in Appendixes A and B. The entropy is found to consist of two major parts. The so-called energetic part G_E,

$$G_E^n(q)=\ln\int DtH^n\left(\sqrt{\frac{q}{1-q}}t\right), \tag{7}$$

is the same as for the simple perceptron. Here we have used the standard abbreviations $Dt=\exp(-t^2/2)/\sqrt{2\pi}dt$ and $H(x)=\int_x^{\infty}Dt$. In the limit $n\to 0$, the capacity problem, the linear term in n yields

$$G_E^{cp}(q)=\int Dt\ln H\left(\sqrt{\frac{q}{1-q}}t\right). \tag{8}$$

In the limit $n\to 1$, the generalization problem, the linear term in $(n-1)$ yields

$$G_E^{gn}(q)=2\int DtH\left(\sqrt{\frac{q}{1-q}}t\right)\ln H\left(\sqrt{\frac{q}{1-q}}t\right). \tag{9}$$

The other part, G_S, is more specific to the network architecture and is in the present case much more involved than for the perceptron. Moreover, it depends on the *a priori* measure $d\mu(\mathbf{J})$ for the couplings. We will therefore discuss separately its explicit form for different *a priori* constraints on the hidden-to-output couplings.

IV. CAPACITY

In this section we discuss the capacity problem. The entropy, Eq. (3), is found to decrease rapidly with an increasing number of random input-output pairs corresponding to less and less flexibility in implementing additional mappings. At a sharp threshold α_c of the storage ratio α no room for further adaptation is left. Within replica symmetry (RS) this is signaled by $q\to 1$, which implies that the available phase space has shrunk to a point, since different solutions of the problem are almost identical. We first investigate the case of binary couplings.

A. Binary couplings

The case where $J_i=\pm 1$ is very special since, due to inversion symmetry, it is exactly equivalent to fixing all the hidden-to-output weights to $J_i=+1$ (the so-called committee machine). Indeed any $J_i=-1$ can be flipped to $J_i=+1$, while at the same time replacing the Boolean function $B_i(\boldsymbol{\xi})$ with its mirror image $\tilde{B}_i(\boldsymbol{\xi})=-B_i(\boldsymbol{\xi})$.

An upper bound for the storage capacity α_c can be obtained from the annealed approximation to the entropy, Eq. (3), given by

$$s_A=-\frac{T}{N}\ln(\langle\langle Z\rangle\rangle_{\boldsymbol{\xi}})=(2^{L-1}-\alpha L)\ln 2, \tag{10}$$

where we assume that all antisymmetric Boolean functions are admissible, $N_B=2^{2^{L-1}}$. Since the entropy must be positive we find

$$\alpha_c \leqslant \alpha_c^{UB} = \frac{2^{L-1}}{L}. \tag{11}$$

As in the case of the Ising perceptron, this bound is related to information theory. The full specification of the network with all $J_i = 1$ requires $N\,2^{L-1}$ bits of information necessary to pin down the N Boolean functions. Therefore the network cannot store more than $N\,2^{L-1}$ bits and α_c cannot exceed $2^{L-1}/L$.

A more detailed characterization of the storage abilities of the network can be obtained from the quenched entropy. Appendix A includes a detailed presentation of the derivations for the general case of discrete values discussed in Sec. IV B. The binary case is a specific case of these general derivations. Therefore, the last two terms appearing in Eq. (A2) are simply $\exp[\Sigma_{a<b}\hat{q}^{ab}\Sigma_j\langle\langle B_j^a(\xi)B_j^b(\xi)\rangle\rangle_{\{\xi\}} - \Sigma_{a,\mu}(\hat{\lambda}_\mu^a)^2]$. In this way we find

$$s = \mathrm{extr}_{q,\hat{q}}\{-\tfrac{1}{2}\hat{q}(1-q) + \alpha L G_E^{cp}(q) + G_s(\hat{q})\}, \tag{12}$$

where G_E^{cp} is given in Eq. (8) and

$$G_s(\hat{q}) = \int \prod_{i=1}^{2^{L-1}} Dz_i \ln \mathrm{Tr}_{\{B\}} \exp\left[\sqrt{\frac{\hat{q}}{2^{L-1}}}\right] \mathcal{Z}_B, \tag{13}$$

where $\mathcal{Z}_B = \langle\langle z_i B(\xi_i)\rangle\rangle_{\boldsymbol{\xi}} = \Sigma_{i=1}^{2^{L-1}} z_i B(\xi_i)$. Note that the sum needs to be taken over half of the possible inputs only, for instance, over only those whose first component is positive (in the case of $L=3$ this means that the sums are over $i = 1, \ldots, 4$, from Table I).

When all antisymmetric Boolean functions are at our disposal the above expression can be simplified using

$$\mathrm{Tr}_{\{B\}} \exp\left[A \sum_{\boldsymbol{\xi}} z_i B(\xi_i)\right] = \prod_i \{2\cosh(Az_i)\}. \tag{14}$$

Then G_s is found to be given by

$$G_s(\hat{q}) = 2^{L-1} \int Dz \ln\left[2\cosh\left(\sqrt{\frac{\hat{q}}{2^{L-1}}} z\right)\right]. \tag{15}$$

The transformations $\hat{q} \mapsto 2^{L-1}\hat{q}$ and $\alpha \mapsto 2^{L-1}\alpha/L$ now map the expression for the entropy onto the corresponding expression for the so-called Ising perceptron [20]. Using the results of this case we immediately find that from the limit $q \to 1$ we get

$$\alpha_c^{RS}(L, 2^{2^{L-1}}) = \alpha_c^{RS}(1,2)\,2^{L-1}/L, \tag{16}$$

with $\alpha_c^{RS}(1,2) = 4/\pi$. However, this result is known to overestimate the storage capacity since the entropy becomes negative and replica symmetry is broken for $\alpha < \alpha_c^{RS}$. The correct value for α_c is given by the value at which the replica symmetric entropy vanishes. This implies

$$\alpha_c(L, 2^{2^{L-1}}) = \alpha_c(1,2)\,2^{L-1}/L, \tag{17}$$

TABLE II. Upper bound for α_c, the replica result and the correct result derived according to the zero-entropy criterion (see text) in the case of $L=3$ and binary hidden-to-output weights.

	α_c^{UB}	α_c^{RS}	α_c
$L=3$, $N_B=16$	1.33	1.70	1.11
$L=3$, $N_B=14$	1.27	1.40	1.06

where $\alpha_c(1,2) \cong 0.83$ is the storage capacity of the Ising perceptron [20]. The most important point following from this result is that the storage capacity of the proposed network scales with the *logarithm of the number of admissible Boolean functions*.

If we restrict ourselves to the set of Boolean functions which may be implemented by perceptrons, cf. Eq. (2), the identity appearing in Eq. (14) no longer holds. Nevertheless explicit results can be obtained from the numerical solution of the saddle point equations

$$\hat{q}(1-q) = \frac{\alpha L}{2\pi} \int Dt \frac{\exp\left(-\dfrac{qt^2}{1-q}\right)}{H^2\left(\sqrt{\dfrac{q}{1-q}}\, t\right)}, \tag{18}$$

$$2^{\frac{L-1}{2}}\sqrt{\hat{q}}(1-q) = \int \prod Dz_i \frac{\mathrm{Tr}_B \mathcal{Z}_B \exp\left[\sqrt{\dfrac{\hat{q}}{2^{L-1}}}\mathcal{Z}_B\right]}{\mathrm{Tr}_B \exp\left[\sqrt{\dfrac{\hat{q}}{2^{L-1}}}\mathcal{Z}_B\right]}.$$

Again, $\alpha_c(L, N_B)$ is determined as the value of α at which the entropy becomes zero. The upper bound is derived either from the annealed approximation or according to information theory and is again given by $\alpha_c^{UB}(L, N_B) = \log_2 N_B / L$.

The results for the special case $L=3$ are collected in Table II. Note that the values for the storage capacity in the two cases again scale as the respective logarithms of the number of admissible Boolean functions, $\alpha_c(3,14)/\alpha_c(3,16) = 1.06/1.11 \cong \ln 14/\ln 16$.

In Fig. 2 we compare the analytical result $\alpha_c(3,14)$ with numerical simulations using exact enumerations. We determine $f(\alpha)$, the fraction of learning sessions in which the complete training set is learned for $N=5$ (7). The data points are obtained by performing in any given α four groups of 250 (50) experiments which are 4×250 (50) choices of patterns. The standard deviation of the calculated quantities over the four different results are used to produce error bars for the depicted mean quantities. Even for the small sizes accessible to this numerical technique we find a steepening of the transition with increasing N and a crossing point of the curves close to the theoretical prediction.

B. Discrete couplings

We can generalize the above analysis to the case of discrete couplings in the hidden-to-output layer

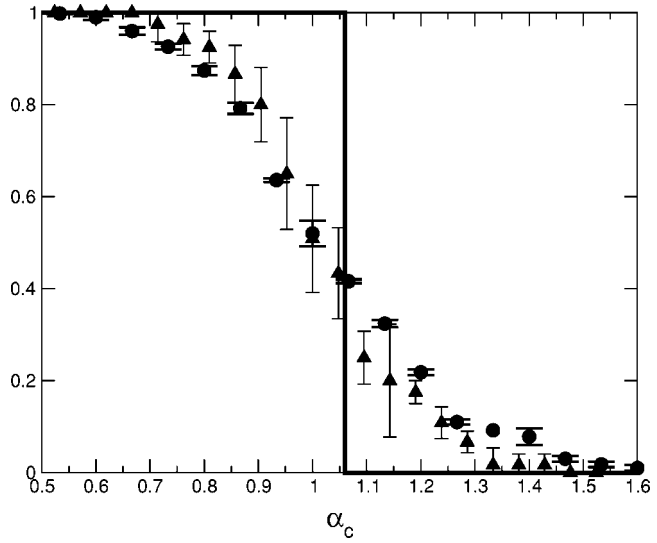

FIG. 2. Fraction f of the runs in which all $\alpha 3N$ random input-output mappings were embedded by a MLN with binary hidden-to-output weights and $N_B=14$. Averages over 4×250 realizations in the case of $N=5$ (circles) and 4×50 in the case of $N=7$ (triangles) are compared with the analytical result (solid line).

$$J_i=\pm\frac{1}{\mathcal{L}},\pm\frac{2}{\mathcal{L}},\ldots,\pm 1. \tag{19}$$

In a manner similar to the binary case, we use the zero entropy criterion that was found to give the best estimation for the storage capacity in the case of finite synaptic depth [21,22]. In this case there are four order parameters in the analytical equations, q [Eq. (6)], its conjugate $\hat{q}$, $\overline{q}=\Sigma_i(J_i^a)^2/N$, and its conjugate, $\hat{\overline{q}}$. A detailed derivation of α_c is given in Appendix A.

We determine explicit numerical results for the storage capacity $\alpha_c(\mathcal{L})$ for the simple cases $L=3$ and $N_B=16$ and $N_B=14$ only. The equations for the order parameters in the case of $L=3$ and $N_B=16$ are given by Eqs. (A10), α_c is found by setting the entropy (A7) to zero. The case $N_B=14$ was treated in a similar manner. The results for $\alpha_c(\mathcal{L})$ for $\mathcal{L}=1,2,3,4,5$ are shown against each other in the inset of Fig. 3. The solid line is a linear fit, $\alpha_c(3,14)=a\,\alpha_c(3,16)$ with $a=0.96\pm0.01$. This is in good agreement with our assumption that $a\sim\ln 14/\ln 16\cong0.95$ for any $\mathcal{L}$.

The capacity increases monotonically with $\mathcal{L}$ in both cases. As $\mathcal{L}$ becomes large, the numerical solution of Eqs. (A10) becomes very sensitive. In Fig. 3 we present the analytical results for $L=3$ and $N_B=16$. To extract the asymptotic behavior for large $\mathcal{L}$, we fitted the dependence $\alpha_c(3,16)=1.90+0.51/\mathcal{L}-1.42\ln(\mathcal{L})/\mathcal{L}$ to the data points starting from (and including) $\mathcal{L}=8$. For $\mathcal{L}\to\infty$ we get $\alpha_c\sim1.9$, which is close to the result for continuous couplings (see Table III).

It is rather difficult to compare these analytical findings with numerical simulations, since the effects of the finite synaptic depth do not show up at the small values of N accessible to exact enumerations [19].

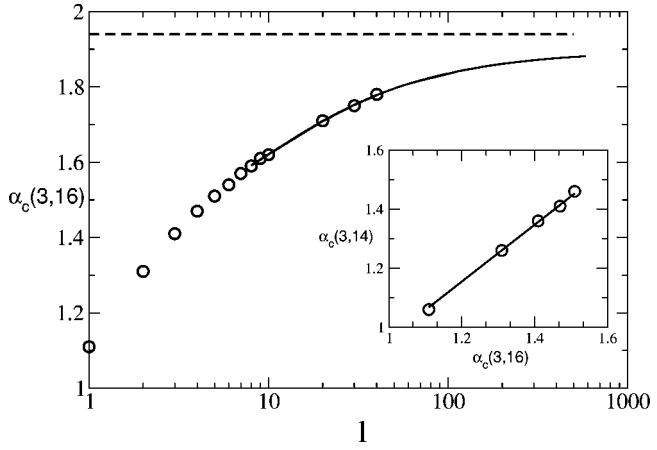

FIG. 3. Analytical results of $\alpha_c(3,16)$, derived according to the zero-entropy criterion as a function of the synaptic depth in the hidden-to-output layer $\mathcal{L}$, are presented in a semilog plot (circles). The solid line is a fit to the asymptotic behavior, the dashed line is the RS result for continuous hidden-to-output couplings. The inset shows the proportionality of $\alpha_c(3,14,\mathcal{L})$ and $\alpha_c(3,16,\mathcal{L})$ for $\mathcal{L}=1,2,3,4,5$ (from bottom to top).

C. Continuous couplings

For continuous couplings we enforce as usual the spherical constraint $\Sigma_{i=1}^N J_i^2=N$. We try to determine the maximum number $\alpha_c LN$ of input-output mappings that can be stored in such a network. The zero-entropy criterion cannot be used in this case since s can be negative when the version space is continuous. We start by deriving an upper bound for α_c. A lower bound is given by the results for finite depth obtained above. Clearly, the possibilities in a network with finite depth are limited compared to the continuous weights, and therefore its maximal capacity should be smaller. We chose to introduce in Table III the results derived for $\mathcal{L}=5$ as a lower bound for α_c in the case of continuous couplings. In principle, any discrete set (i.e., any value of $\mathcal{L}$) can serve as a lower bound when the limit $\mathcal{L}\to\infty$ is supposed to be the closest lower bound (see the discussion in Sec. IV D).

We derive an upper bound for α_c by counting the different configurations that may be generated for given inputs $\boldsymbol{\xi}_i^\mu$, $\mu=1,\ldots,\alpha LN$. There are at most $2^{N(\log_2 N_B-1)}$ different configurations of hidden units, using different combinations of the N_B Boolean functions. Since the mapping from the hidden layer to the output is performed by a perceptron, each hidden configuration gives rise to the desired output with probability $C(p,N)/2^p$ with $p=\alpha LN$. Here $C(p,N)$ denotes the number of dichotomies calculated in Ref. [14],

TABLE III. Upper bound for α_c, the replica symmetry result, and lower bound derived from the case of discrete couplings with $\mathcal{L}=5$, in the case of $L=3$ and continuous hidden-to-output weights.

	α_c^{UB}	α_c^{RS}	α_c^{LB}
$L=3$, $N_B=16$	2.39	1.95	1.51
$L=3$, $N_B=14$	2.32	1.85	1.46

$$\frac{1}{N}\ln C(p,N)\sim \alpha L\,\ln(\alpha L)-(\alpha L-1)\ln(\alpha L-1). \quad (20)$$

Setting the probability, $2^{N(\log_2 N_B-1)}C(p,N)/2^p$, equal to 1/2 we find that α_c is bounded for $N\to\infty$ by the solution $\alpha^{MD}(L,N_B)$ of the equation

$$\ln\frac{N_B}{2}=(\alpha L-1)\ln(\alpha L-1)-\alpha L\,\ln\frac{\alpha L}{2}. \quad (21)$$

The result is an upper bound rather than an exact result, since we neglected correlations between the different dichotomies (see Ref. [15]).

For $L=1$, $N_B=2$, we get the expected result $\alpha^{MD}=2$. In the case of $L=3$ we find $\alpha^{MD}(3,16)\cong 2.394$ and $\alpha^{MD}(3,14)\cong 2.315$ (as appears in Table III). In the limit of large L, the bound is

$$\lim_{L\to\infty}\alpha^{MD}(L,N_B)\sim\frac{\log_2 N_B}{L}. \quad (22)$$

This result shows the same scaling with the number of Boolean functions as the lower bound derived from the zero-entropy result in the case of binary couplings, Eq. (17). Hence we can summarize at this stage, without even calculating the capacity directly, that the maximal capacity in the continuous case scales with $\log N_B/L$ and the prefactor is larger than 0.83.

If the mapping from the input to the hidden layer is done by perceptrons we know that the number of implementable Boolean functions scales like $N_B\sim e^{L^2}$ for large L. Therefore, in this limit the upper bound assumes the form $\alpha^{MD}\sim L$ implying that adding more inputs to each hidden unit linearly enlarges the maximal storage capacity.

The analysis of the replica calculations in the case of continuous weights is given in Appendix B. Equations (B6) are the equations for the order parameters in the general case. In the small-α regime, the order parameter q is given by

$$q\sim\frac{2L}{\pi 2^{L-1}}\alpha. \quad (23)$$

This relation holds in both the binary and the discrete cases. The overlap parameter q grows with increasing α with a slope decreasing proportionally to the number of inputs per unit, 2^{L-1}, independent of N_B and the measure in the couplings space, $\mu(\mathbf{J})$.

We carried out numerical simulations in the case of $L=3$, $N_B=14$, and $N=5$. We determined the behavior of the order parameter q for small α as shown in Fig. 4 (circles). Error bars are half of the standard deviation obtained from 1000 different runs. The linear approximation, Eq. (23), is given by the dashed line. The simulation results compare well with the analytic result Eq. (23) (solid line) and the linear approximation. As α increases there is a deviation from the analytical curve; the better learning performance of the simulations is due to finite size effects.

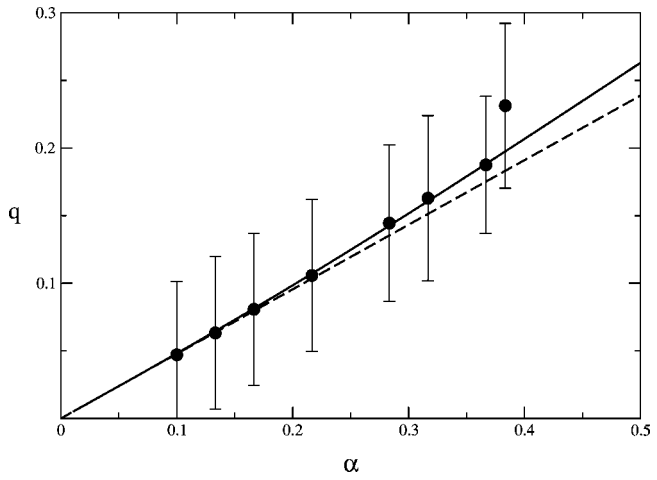

FIG. 4. Analytical results of q, as a function of α derived from Eqs. (B6), in the case of $L=3$, $N_B=14$, and continuous hidden-to-output couplings (solid line). The dashed line shows the linear approximation around $\alpha\to 0$, Eq. (23). Simulation results in the case of $N=5$ (circles) are in good agreement for small α. Error bars are half of the standard deviation obtained from 1000 different runs.

As soon as q approaches 1 the numerical integrals diverge, and α_c is found from the asymptotic expansion of the functions for $q\to 1$ and $\hat{q}\sim 1/(1-q)^2\to\infty$. In the case of $L=3$ if $N_B=14$ we get $\alpha_c\cong 1.85$, whereas if $N_B=16$, the critical α is somewhat larger, $\alpha_c\cong 1.95$, and the ratio between the results is again connected to the ratio between the logarithm of N_B. The general result when all the antisymmetric Boolean functions are admissible is

$$\alpha_c(L,2^{2^{L-1}})=\frac{2+\frac{4}{\pi}(2^{L-1}-1)}{L}. \quad (24)$$

Simulations

A great computational effort is demanded in performing simulations of the kind of learning by choice of internal representations [23] in an extensive large network when the Boolean functions in the first layer are defined by perceptron mapping. Moreover, when the Boolean functions in the first layer can be any antisymmetric Boolean function, the last method seems to be inappropriate. It appears that in such a case, the natural algorithm will be to go through all the possible mappings in the first layer and in each possibility to try to teach the network using a traditional learning algorithm that is known to perform well in the perceptron. Such partial exact enumerations are time consuming and therefore are performed only for small N.

It has been proved that in the case of $N=3$ and in the case of $N=5$ one can confine the hidden-to-output layer $\mathbf{J}$ to a finite number of values and that this network, although restricted, is capable of implementing the same Boolean functions of the input as the network with no restrictions on its second-layer weights [6,19]. We used the aforementioned equivalence and made exact enumeration calculations in the case of $N=3$ and $N=5$ as shown in Fig. 3. In the case of

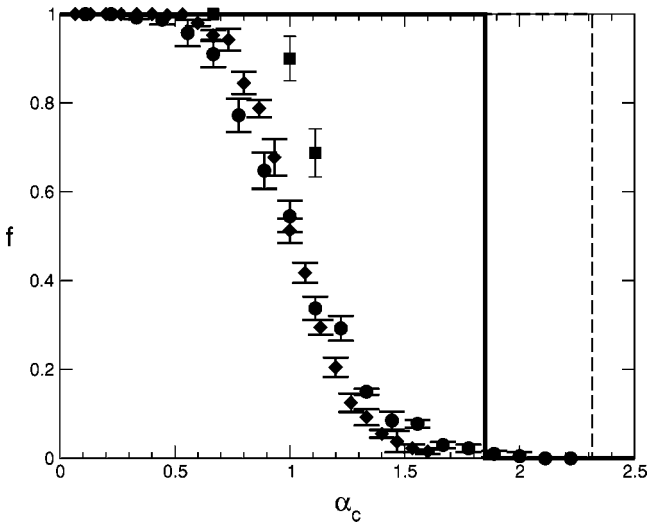

FIG. 5. The fraction of success embedding procedures, *f*, as a function of the number of patterns per input dimension, α, in the case of continuous hidden-to-output weights, $L=3$ and $N_B=14$. A comparison between the analytical result of α_c under the RS assumption (solid line), exact enumeration results in the case of $N=3$ (circles), $N=5$ (diamonds), and partial learning (see text) in the case of $N=9$ (squares). Error bars are half of the standard deviation. The dashed line is the upper bound for α_c.

$N=3$ we had to examine four different $\mathbf{J}$ only, (1 1 1), (1 0 0), (0 1 0), (0 0 1). In the case of $N=5$ we examined the following seven prototype families, (1 1 1 1 1), (1 0 0 0 0), (1 1 1 0 0), (2 1 1 1 0), (3 1 1 1 1), (2 2 1 1 1), (3 2 2 1 1), and all of its permutations. The data points presented in Fig. 5 were obtained by performing 100 experiments four times in any given α.

The discrepancy between the exact enumeration results and the analytical curve in Fig. 5 may be due to finite size effects. The equivalence described above that is the basis for the use of exact enumeration, instead of some sort of learning procedure, actually shows that carrying simulations for small N and continuous hidden-to-output couplings is equivalent to carrying simulations with discrete hidden-to-output couplings, whereas we found that α_c in the discrete case is smaller than α_c in the continuous case. Therefore, we also performed partial exact enumerations for $N=9$. We examined half of the possible Boolean functions (the other half is redundant due to the inversion symmetry indicated above). For each possible evaluation of the Boolean functions in the first-layer units we tried to teach the second-layer according to the ADATRON learning procedure [24]. As α becomes larger the time it took to find whether there is a solution or not becomes longer. Therefore we have results only for $\alpha=0.667,1,1.111$, the squares presented in Fig. 5. The results for $N=9$ were far better than the exact enumerations carried out for $N=3$ and $N=5$. This result is indeed consistent with our observation that the differences between α_c of the continuous and discrete cases become negligible only for very large $\mathcal{L}$ (see Fig. 3).

D. Discussion

The crux of our findings in this section is the property that determines the maximal capacity of networks of the type described above, which was found to be the logarithm of the number of Boolean functions embedded in each unit of the first layer, $\ln(N_B)$. That term was found to determine α_c where only the free factor depends on the kind of limitation one has on the couplings in the net. In the discrete case we have exact results for the critical α from the zero-entropy criterion.

In the case of continuous couplings it appears that there should be a regime in which the RS is unstable. We know, as confirmed by simulation, that in the small-α regime the RS solution is correct (see Fig. 4). Moreover, in the case of $L=1$ the RS solution is stable for $\alpha<\alpha_c$ and is unstable for $\alpha>\alpha_c$ (see [1,25–27] and references therein). The question is whether the RS remains stable in the regime where $\alpha \leqslant \alpha_c$. For $L=1$, the perceptron, the answer is definitely positive. As L becomes very large, the RS solution in the continuous case Eq. (24) meets that of the binary case [Eq. (16)]. Clearly, this solution is unstable since it overestimates the bound [Eq. (22)]. In this paper we specifically examine the case of $L=3$. As one can see in Fig. 3, it appears that the solution in the discrete case with a large synaptic depth, $\mathcal{L} \gg 1$, which may serve as a lower bound, almost coincides with the RS solution for the continuous case. The correcting procedure appears to be very complicated since it was shown that one-step replica symmetry breaking (RSB) [25,26] is not sufficient to solve the storage capacity calculations in the perceptron and one has to solve the perceptron within the full Parisi scheme [27]. The question of stability of the replica and the kind of RSB assumption to be made are not within the realm of this study.

V. GENERALIZATION

We only consider the simplest setup in which the teacher and student network have the same architecture. Accordingly the teacher is defined by a $LN:N:1$ MLN with Boolean functions B_i^T and couplings J_i^T generated at random. The student is given a set of αLN random inputs together with the corresponding outputs of the teacher. The task is to choose the Boolean functions B_i^S and the couplings J_i^S of the student such that the probability for misclassifying a new random example, the generalization error, is small. In Appendix C it is shown that the generalization error is given by

$$\epsilon_g = \frac{1}{\pi}\cos^{-1}\rho, \tag{25}$$

with the normalized overlap $\rho = q/(||\mathbf{J}^T||\ ||\mathbf{J}^S||)$ and

$$q = \frac{1}{N}\sum_i J_i^T J_i^S \langle\langle B_i^T(\boldsymbol{\xi}) B_i^S(\boldsymbol{\xi})\rangle\rangle_{\boldsymbol{\xi}}. \tag{26}$$

Assuming the same *a priori* measures for the teacher and student, the problem exhibits teacher-student symmetry such that replica symmetry holds and the overlap Eq. (26) is identical with the student-student overlap defined in Eq. (6) [1]. It can be derived by taking the limit $n\to 1$ instead of $n\to 0$ in the same expression Eq. (4) for the quenched entropy already used in the capacity problem.

A. Binary couplings

Learning with binary hidden-to-output couplings is expected to show a first-order phase transition, similar to the findings in the discrete perceptron [16]. Here we study only the generalization ability of discrete networks whose hidden-to-output couplings are constrained to binary couplings, $J_i^{T/S}=\pm 1$. The learning features of a discrete network with $2\mathcal{L}$ possible values are easily derived by generalizing to that case using similar methods to those described in Appendix A.

In order to find the overlap ρ as a function of α we calculate the entropy. We start with the terms in Eq. (A5) and substitute $\overline{q}=1$ (hence $\rho=q$). Expanding around $n=1$ results in

$$s=\mathrm{extr}_{q,\hat{q}}\{-\tfrac{1}{2}\hat{q}(1+q)+\alpha L G_E^{gn}(q)+G_S(\hat{q})\}, \quad (27)$$

where G_E^{gn} is defined in Eq. (9) and

$$G_S=\int \prod_{i=1}^{2^{L-1}} Dz_i \ln \mathrm{Tr}_{\{B_i\}} \exp\left[\sqrt{\frac{\hat{q}}{2^{L-1}}}\mathcal{Z}_B+\frac{\hat{q}}{2^{L-1}}\mathcal{B}\right], \quad (28)$$

with $\mathcal{B}=\Sigma_i B(\xi_i)$.

In the case where all $2^{2^{L-1}}$ antisymmetric Boolean functions can be used, the expression for G_S can again be simplified using Eq. (14). In this way we find

$$G_S=2^{L-1}\int Dz \ln 2\cosh\left[\sqrt{\frac{\hat{q}}{2^{L-1}}}z+\frac{\hat{q}}{2^{L-1}}\right]. \quad (29)$$

Using the rescaling $\hat{q}\mapsto 2^{L-1}\hat{q}$ and $\alpha\mapsto 2^{L-1}\alpha/L$ the result for the entropy again maps perfectly on the known result for the Ising perceptron. Hence there is a first-order phase transition from poor to perfect learning at

$$\alpha_c^{learn}(L,2^{2^{L-1}})=\alpha_c^{GD}(1,2)2^{L-1}/L, \quad (30)$$

where $\alpha_c^{GD}(1,2)\cong 1.245$. This value was first found for the perceptron by Gardner and Derrida on the basis of numerical simulations [3], and was shortly afterwards derived analytically in Ref. [16]. In the case of $L=3$, Eq. (30) yields a phase transition to perfect generalization at $\alpha_c\cong 1.66$.

In the case of perceptron mappings between the input and hidden layer, i.e., general N_B, one has the following set of equations:

$$\hat{q}=\frac{\alpha L}{\pi\sqrt{1-q}}\int Dt e^{-qt^2/2}H(\sqrt{q}t), \quad (31)$$

$$1+q=\int \prod_{i=1}^{2^{L-1}} Dz_i \frac{\mathrm{Tr}_B\left(\frac{1}{\sqrt{\hat{q}2^{L-1}}}\mathcal{Z}_B+2^{2-L}\mathcal{B}\right)\exp\Sigma_{\mathcal{Z}B}}{\mathrm{Tr}_B\exp\Sigma_{\mathcal{Z}B}},$$

where $\Sigma_{\mathcal{Z}B}=\sqrt{\hat{q}/2^{L-1}}\mathcal{Z}_B+(\hat{q}/2^{L-1})\mathcal{B}$. Like in the case of the binary perceptron, this set of equations has two solutions: $q\to 1$, $\hat{q}\to\infty$, which is the result for any finite α and gives identical zero entropy. The other solution is $q(\alpha)\neq 1$ and is physically correct up to α_c, where the entropy vanishes.

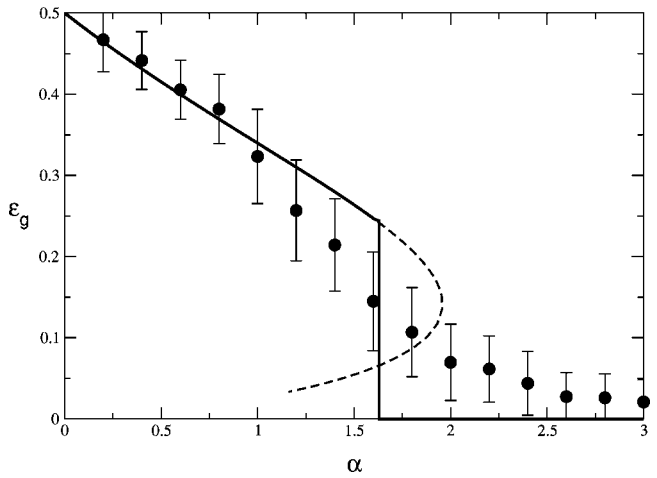

FIG. 6. Analytical results of ϵ_g as a function of α in the case of $L=3$, $N_B=14$, and binary hidden-to-output couplings (solid line). The dashed line shows the nonphysical solution for $\alpha>\alpha_c$. We ran exact enumerations with $N=5$ (circles). Error bars are half of the standard deviations obtained from 100 runs.

The numerical result of $\epsilon_g(\alpha)$ in the case of $L=3$ and $N_B=14$, derived by Eqs. (31), the vanishing entropy criteria, and Eq. (25) are presented in Fig. 6. The solid line is the analytical curve $\epsilon_g(\alpha)$ where the phase transition from poor to perfect generalization occurs. The transition occurs at $\alpha_c\cong 1.62$. As expected, a smaller number of Boolean functions in each unit of the first layer results in faster learning, $\alpha_c(3,14)<\alpha_c(3,16)$. A smaller value of the critical storage ratio α_c determined in the capacity problem usually gives rise to quicker generalization. The reason is that the network cannot reproduce many input-output pairs without having a key to how they are produced (*generalization starts where learning ends*).

We ran exact enumerations in this case for $N=5$. Despite the fact that N is small, in the small-α regime there is good agreement between the analytical curve and the averaged simulation results. The averaged results obtained from 100 runs and the standard deviations are presented in Fig. 6. The first-order transition is in the simulation smoothed by finite size effects.

B. Continuous couplings

The entropy of a $3N{:}N{:}1$ network with continuous hidden-to-output weights as a function of n is given in Eqs. (B2) and (B3). As indicated above, taking the limit $n\to 1$ is appropriate for the learning problem. We redefine the parameters, $\hat{Q}=\hat{q}/(k+\hat{q})$, and find that

$$\hat{q}=\frac{\hat{Q}}{1-\hat{Q}}, \quad (32)$$

since the zero order, $(n-1)^0$ of the entropy should vanish. The entropy calculated to first order $(n-1)^1$ is given by

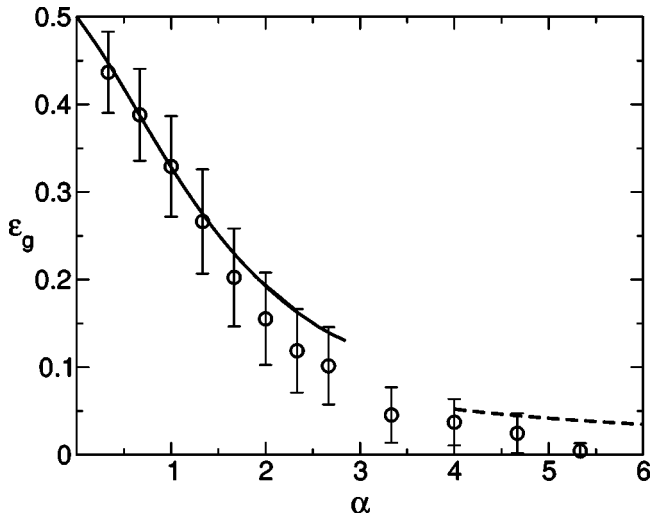

FIG. 7. The generalization error as a function of α for $L=3$, $N_B=16$, and continuous hidden-to-output couplings derived from the analytical Eqs. (35) (solid line) together with the asymptotic expansion of ϵ_g, Eq. (36) (dashed line). The circles are results of exact enumerations with $N=5$, with error bars obtained from 100 runs.

$$\underset{q\hat{Q}}{\mathrm{extr}}\left\{-\frac{q\hat{Q}}{2(1-\hat{Q})}+\frac{1}{2}\ln(1-\hat{Q})+\alpha L G_E^{gn}(q)+G_S^{gn}(\hat{Q})\right\}, \tag{33}$$

where G_E^{gn} is given by Eq. (9) and

$$\frac{G_S}{\sqrt{1-\hat{Q}}}=\int\prod_{i=1}^{2^{L-1}}Dz_i\mathrm{Tr}_B\exp\left(\frac{\hat{Q}}{2^L}\mathcal{Z}_B\right)\ln\mathrm{Tr}_B\exp\left(\frac{\hat{Q}}{2^L}\mathcal{Z}_B\right). \tag{34}$$

The equations derived by taking the extremum are

$$\frac{\hat{Q}}{(1-\hat{Q})}=\frac{\alpha L}{2\pi\sqrt{1-q}}\int Dt\frac{\exp\left\{-\frac{qt^2}{2}\right\}}{H(\sqrt{q}t)},$$

$$q=2(1-\hat{Q})^2\frac{\partial G_S}{\partial\hat{Q}}-(1-\hat{Q}). \tag{35}$$

At the end of the learning procedure, when $q\to1$, one also finds that $\hat{Q}\to1$. We derived the generalization error from Eq. (25) and assumed that all the antisymmetric Boolean functions are available for the first layer. In that case, $\partial G_S/\partial\hat{Q}\sim1/[2(1-\hat{Q})^2]$ and

$$\epsilon_g\sim\frac{0.625}{L\alpha}. \tag{36}$$

Not surprisingly, the generalization error decays according to a power law, as in the spherical perceptron [18]. The decay is slower for larger L, again reflecting the enhanced storage abilities. The numerical derivation of $\epsilon_g(\alpha)$ given by Eqs. (35) and (25) in the case of $L=3$ and $N_B=16$ is presented in Fig. 7 (solid line). For large α, the derivation of ϵ_g from the numerical integrals becomes impossible, due to the sensitive integrals involved. Therefore, we present the asymptotic expansion (dashed line) for large α, Eq. (36). The averaged exact enumeration results taken from 100 samples with $N=5$ are in good agreement for small α (circles), whereas for large α the generalization error in the simulations vanishes faster to zero due to finite size effects.

C. Discussion

In summary, we found that learning in large two-layered perceptrons is possible. The learning curve behaves in the same way as in the case of a simple perceptron — phase transition in the binary case and power law decay in the continuous case. Such a similarity was observed in the case of a large number of hidden units $K\to\infty$ when $K\ll N$ [7]. However, in the two-layered perceptrons presented in this paper, the power-law decay in the continuous case depends on the number of inputs to each hidden unit, L. Moreover, the discontinuous transition in the discrete case occurs at a value of α, which scales with the logarithm of the number of Boolean functions in each unit in the first layer, $\ln N_B$.

In this work we used the most simple learning algorithms. We counted on exact enumerations in small N at least for the first layer and then the second one was treated as a simple perceptron. Such exact enumerations are performed by repeating the whole set of examples for each realization of the Boolean functions in the first layer, and trying to embed the input-output relations by training the second layer. As shown in Fig. 7 such procedures yield reliable results only for small α. To address the question of whether there is an efficient algorithm which achieves an α^{-1} decay of ϵ_g in the continuous case, on-line learning schemes should be used, as shown in the Committee Machine [9]. The on-line analysis of the ability of the extensively large two-layered perceptrons warrants further study.

ACKNOWLEDGMENT

The partial support of the GIF is acknowledged.

APPENDIX A

In this appendix we calculate the dependence on α of the order parameter q describing the overlap between different networks that can embed αLN random examples. All networks have components in the hidden-to-output layer that are confined to a finite set of values. The general description is exemplified for the values given in Eq. (19), where the binary case is a special case with $\mathcal{L}=1$.

Our starting point is Eq. (5). First, we rescale the argument of the θ function by a factor of $1/\sqrt{N}$. In such a way we ensure that in the thermodynamic limit the argument, which is the local field, will be in the appropriate order. We rewrite the equation by using the integral representation of the θ function, using λ_μ^a and $\hat{\lambda}_\mu^a$ for that purpose,

$$\langle\langle\Omega^n\rangle\rangle=\lim_{N\to\infty}\frac{1}{N}\int\prod_{\mu=1}^{\alpha LN}\prod_{a=1}^{n}\left\{\frac{d\lambda_\mu^a d\hat{\lambda}_\mu^a}{2\pi}\exp[i\lambda_\mu^a\hat{\lambda}_\mu^a]\right\}\times\prod_{\mu=1}^{\alpha LN}\prod_{j=1}^{N}\left\langle\left\langle\exp\left[-i\frac{1}{\sqrt{N}}\sum_{a=1}^{n}\hat{\lambda}_\mu^a J_j^a B_j^a(\boldsymbol{\xi}_j^\mu)\right]\right\rangle\right\rangle_{\boldsymbol{\xi}}. \tag{A1}$$

We take the Taylor expansion of the last exponent in the right-hand side of the equation above up to the quadratic order. The linear term vanishes and therefore, by recollecting everything to an exponent form, we have a Gaussian. Introducing the order parameter, Eq. (6), we have

$$\begin{aligned}\langle\langle\Omega^n\rangle\rangle=&\lim_{N\to\infty}\frac{1}{N}\int\prod_{\mu=1}^{\alpha LN}\prod_{a=1}^{n}\left\{\frac{d\lambda_\mu^a d\hat{\lambda}_\mu^a}{2\pi}\exp[i\lambda_\mu^a\hat{\lambda}_\mu^a]\right\}\\&\times\int\prod_{a<b}\frac{dq^{a,b}d\hat{q}^{a,b}}{2\pi/N}\exp\left[-\sum_{a<b}Nq^{a,b}\hat{q}^{a,b}-\sum_{\mu,a<b}q^{ab}\hat{\lambda}_\mu^a\hat{\lambda}_\mu^b\right]\times\int\prod_a d\mu(\mathbf{J}^a)\mathrm{Tr}_{\{B_j\}}\\&\times\exp\left[\sum_{a<b}\hat{q}^{ab}\sum_j J_j^a J_j^b\langle\langle B_j^a(\boldsymbol{\xi})B_j^b(\boldsymbol{\xi})\rangle\rangle_{\{\xi\}}\right]\\&\times\exp\left[-\sum_{\mu,a,j}(\hat{\lambda}_\mu^a J_j^a)^2/2\right].\end{aligned} \tag{A2}$$

In the case of discrete couplings $d\mu(\mathbf{J})=\mathrm{Tr}_J$, we define, similarly to the perceptron [21], an additional order parameter $\overline{q}^a=\Sigma_j(J_j^a)^2/N$ and its conjugate $\hat{\overline{q}}^a$. Counting on the replica symmetry assumption we derive

$$\begin{aligned}\langle\langle\Omega^n\rangle\rangle=&\lim_{N\to\infty}\frac{1}{N}\int\prod_{\mu=1}^{\alpha LN}\prod_{a=1}^{n}\frac{d\lambda_\mu^a d\hat{\lambda}_\mu^a}{2\pi}[i\lambda_\mu^a\hat{\lambda}_\mu^a]\\&\times\int\frac{dqd\hat{q}}{2\pi/N}\int\frac{d\overline{q}d\hat{\overline{q}}}{2\pi/N}\exp\left[-N\frac{n(n-1)q\hat{q}}{2}-Nn\overline{q}\hat{\overline{q}}\right]\\&\times\exp\left[-q\sum_{\mu,a<b}\hat{\lambda}_\mu^a\hat{\lambda}_\mu^b-\frac{\overline{q}-q}{2}\sum_{\mu,a}(\lambda_\mu^a)^2\right]\\&\times\mathrm{Tr}_{\{B_j\}}\mathrm{Tr}_{\{J\}}\exp\left[-\frac{N}{2}\hat{q}\sum_a(J^a)^2+N\overline{q}\sum_a(J^a)^2\right]\\&\times\exp\left[\hat{q}\sum_{j,a<b}J_j^a J_j^b\langle\langle B_j^a(\boldsymbol{\xi})B_j^b(\boldsymbol{\xi})\rangle\rangle_{\{\xi\}}\right].\end{aligned} \tag{A3}$$

At this stage it is impossible to calculate the integrals over $\hat{\lambda}_\mu^a$ and to perform the trace over J^a since both appear in mixed exponents that contain different replicas. We circumvent this difficulty by using the Gaussian integral

$$\begin{aligned}&\exp\left[\hat{q}\sum_{a<b}J_j^a J_j^b\langle\langle B_j^a(\boldsymbol{\xi})B_j^b(\boldsymbol{\xi})\rangle\rangle_{\{\xi\}}\right]\\&=\exp\left[\frac{\hat{q}}{2^{L-1}}\sum_{a,b,\xi_i}J_j^a J_j^b B_j^a(\boldsymbol{\xi}_i)B_j^b(\boldsymbol{\xi}_i)-\frac{\hat{q}}{2}\sum_a(J_j^a)^2\right]\\&=\int\prod_{i=1}^{2^{L-1}}Dz_i\exp\left[\sqrt{\frac{\hat{q}}{2^{L-1}}}\sum_{a,i}J_j^a B_j^a(\boldsymbol{\xi}_i)z_i-\frac{\hat{q}}{2}\sum_a(J_j^a)^2\right].\end{aligned} \tag{A4}$$

The mixed terms involving $\hat{\lambda}_\mu^a$ are treated in the same manner. The product and the sum at the end of Eq. (A4) are due to the average over ξ. The possible inputs are divided into two groups, one being the opposite of the other. It can be shown that as a result of the inversion symmetry of the Boolean functions, it is sufficient to go through one of the groups — half of the input [e.g., to evaluate the terms for the input 1 to 4 in the case of $L=3$ (Table I)]. This leads to

$$\begin{aligned}\langle\langle\Omega^n\rangle\rangle=&\int\frac{dqd\hat{q}}{2\pi/N}\int\frac{d\overline{q}d\hat{\overline{q}}}{2\pi/N}\exp\left\{-N\left[\frac{n(n-1)}{2}q\hat{q}\right]\right\}\\&\times\exp\left\{-N\left[\frac{n(n-1)}{2}q\hat{q}+\overline{q}\hat{\overline{q}}n-\alpha LG_E^n-G_S^n\right]\right\},\end{aligned} \tag{A5}$$

where

$$G_E^{n,Disc}=\ln\int Dt H^n\left(\sqrt{\frac{q}{\overline{q}-q}}t\right),$$

$$G_S^{n,Disc}=\ln\int\prod_{i=1}^{2^{L-1}}Dz_i\{\ln[\mathrm{Tr}_{J,B}e^{-(\hat{q}/2-\hat{\overline{q}})J^2}e^{J\sqrt{\hat{q}/2^{L-1}}\mathcal{Z}_B}]\}^n. \tag{A6}$$

We use redefinitions of the parameters similar to those in Ref. [21], $F_1=\hat{q}, F_2=\frac{1}{2}F_1-\hat{\overline{q}}$. The entropy is rewritten as a function of the last parameters and in the limit of $n\to\infty$,

$$s^{Disc}=\operatorname*{extr}_{F_1,F_2,q,\overline{q}}\left\{\alpha G_E^{Disc}+G_S^{Disc}-F_2\overline{q}-\frac{F_1}{2}(\overline{q}-q)\right\}, \tag{A7}$$

where G_E^{Disc} is similar to Eq. (8),

$$G_E^{Disc}=\int Dt\ln H\left(\sqrt{\frac{q}{\overline{q}-q}}t\right), \tag{A8}$$

and is the same expression derived for the discrete perceptron. For the Ising perceptron, $\overline{q}=1$ by definition, and one gets Eq. (8) exactly.

However, G_S^{Disc} is unique to MLN. It can be rewritten in a comparatively simple manner if we assume that all the

antisymmetric Boolean functions are possible: $N_B=2^{2^{L-1}}$. We then use the identity Eq. (14). The generalization of G_S^{Disc} to the case where only perceptron mappings are admissible is straightforward but tedious,

$$G_S^{Disc}=\int\prod_{i=1}^{2^{L-1}}Dz_i\ln \mathrm{Tr}_J\left[e^{-F_2J^2}\prod_{i=1}^{2^{L-1}}\cosh\left(\sqrt{\frac{F_1}{2^{L-1}}}Jz_i\right)\right]. \tag{A9}$$

The four equations for the set of parameters $\{q,\bar{q},F_1,F_2\}$ are derived by finding the extremum of Eq. (A7) with respect to the parameters

$$F_2=\frac{F_1(\bar{q}-q)}{2\bar{q}},$$

$$F_1=\frac{\alpha L}{\sqrt{2\pi}(\bar{q}-q)^{3/2}}\frac{\bar{q}}{q}\int tDt\frac{e^{-[q/2(\bar{q}-q)]t^2}}{H\left(\sqrt{\frac{q}{\bar{q}-q}}t\right)},$$

$$\bar{q}=\int\prod_{i=1}^{2^{L-1}}Dz_i\langle J^2\rangle,$$

$$\bar{q}-q=\frac{1}{\sqrt{2^{L-1}F_1}}\int\prod_{i=1}^{2^{L-1}}Dz_i\left\langle J\sum_i z_i\tanh(C_i)\right\rangle, \tag{A10}$$

where the average is defined as follows:

$$\langle A(J)\rangle\equiv\frac{\mathrm{Tr}_J A(J)e^{-F_2J^2}\prod_i\cosh(C_i)}{\mathrm{Tr}_J e^{-F_2J^2}\prod_i\cosh(C_i)} \tag{A11}$$

and

$$C_i=\sqrt{\frac{F_1}{2^{L-1}}}Jz_i. \tag{A12}$$

The maximum capacity α_c is found by calculating the number of examples per input dimension α in which the entropy vanishes.

APPENDIX B

In the following we calculate the order parameter q for networks that try to store random examples. The hidden-to-output weight vectors in these networks are subject to the spherical constraint, i.e.,

$$d\mu(\mathbf{J})=\prod_i\frac{dJ_i}{\sqrt{2\pi e}}\delta\left(\sum_{i=1}^N J_i^2-N\right). \tag{B1}$$

The above distribution is substituted in Eq. (A2) by employing the integral representation of the δ function and using the parameter k (see Ref. [1]). By applying the Gaussian integrals, Eq. (A4), and assembling everything we derive

$$\langle\langle\Omega^n\rangle\rangle=\int\frac{dqd\hat{q}}{2\pi/N}\int\frac{dk}{4\pi}\exp\left\{-N\left[\frac{n(n-1)}{2}q\hat{q}+Nn\frac{k}{2}\right]\right\}$$
$$\times\exp\{N[\alpha LG_E^n(q)+G_S^n(k,\hat{q})]\}, \tag{B2}$$

where $G_E^n(q)$ is given in Eq. (7) and

$$G_S^n=\ln\int\prod_{i=1}^{2^{L-1}}Dz_i\left[\mathrm{Tr}_B\exp\left\{\frac{\hat{q}}{2^L(k+\hat{q})}\mathcal{Z}_B^2\right\}\right]^n$$
$$-\frac{n}{2}-\frac{n}{2}\ln(k+\hat{q}). \tag{B3}$$

Taking the limit $n\to 0$ one gets the following expression for the entropy:

$$\underset{\hat{q},q,k}{\mathrm{extr}}\left\{\frac{q\hat{q}}{2}+\frac{k}{2}-\frac{1}{2}\ln(k+\hat{q})+\alpha LG_E^{cp}(q)+G_S(k,\hat{q})\right\}, \tag{B4}$$

where G_E^{cp} is given in Eq. (8) and

$$G_S=\int\prod_{i=1}^{2^{L-1}}Dz_i\mathrm{Tr}_B\exp\left\{\frac{\hat{q}}{2^L(k+\hat{q})}\mathcal{Z}_B^2\right\}. \tag{B5}$$

Taking the extremum over the parameters yields three equations:

$$k=1-q\hat{q},$$

$$\frac{\hat{q}(1-q)}{k+q\hat{q}}=\frac{\alpha L}{2\pi}\int Dt\frac{e^{-(qt^2/1-q)}}{H^2\left(\sqrt{\frac{q}{1-q}}t\right)},$$

$$\frac{1-q}{1-\hat{q}\frac{1-q}{k+\hat{q}}}=\frac{1}{2^{L-1}}\int\prod_{i=1}^{2^{L-1}}Dz_i\frac{\mathrm{Tr}_B\mathcal{Z}_B^2\exp\left\{\frac{\hat{q}}{2^L(k+\hat{q})}\mathcal{Z}_B^2\right\}}{\mathrm{Tr}_B\exp\left\{\frac{\hat{q}}{2^L(k+\hat{q})}\mathcal{Z}_B^2\right\}}. \tag{B6}$$

The result of the saddle point equations is the evolution of the overlap between different networks capable of storing α random examples, $q(\alpha)$.

APPENDIX C

In this appendix the joint probability distribution of x and y [defined in Eq. (C2)] is calculated under the spherical assumption ($q=\rho$). Having this probability, $P(x,y|\rho)$, enables calculation of the generalization error according to its definition,

$$\epsilon_g = \langle\langle \theta(-xy)\rangle\rangle_{xy}. \tag{C1}$$

The parameters, x and y represent the local fields

$$x \equiv \frac{1}{\sqrt{N}} \sum_i J_i^S B_i^S(\boldsymbol{\xi}_i), \quad y \equiv \frac{1}{\sqrt{N}} \sum_i J_i^T B_i^T(\boldsymbol{\xi}_i), \tag{C2}$$

and since the output σ is the sign of the local fields, Eq. (C1) simply states that the generalization error is the averaged discrepancy between the teacher's and the student's output. We show that although the $LN{:}N{:}1$ network is different from the perceptron, the final function $P(x,y|\rho)$ is the same and therefore one can find a simple expression for the generalization error, Eq. (25).

Under the spherical constraint, the assumptions are as follows:

$$\frac{1}{N}\sum_i (J_i^T)^2 = 1, \quad \frac{1}{N}\sum_j (J_i^s)^2 = 1, \tag{C3}$$

$$\frac{1}{N}\sum_j J_i^T J_j^S \langle\langle B_j^T(\boldsymbol{\xi}) B_j^S(\boldsymbol{\xi})\rangle\rangle_{\boldsymbol{\xi}} = \rho.$$

We calculate the joint probability distribution according to the definitions of x and y, Eq. (C2),

$$P(x,y|\rho) = \left\langle\left\langle \delta\left(\frac{\sum_j J_j^S B_j^S(\boldsymbol{\xi}_j)}{\sqrt{N}} - x \right) \delta \times \left(\frac{\sum_j J_j^T B_j^T(\boldsymbol{\xi}_j)}{\sqrt{N}} - y \right) \right\rangle\right\rangle_{\{\boldsymbol{\xi}_j\}}. \tag{C4}$$

Representing the δ functions by integrals, one can rewrite the average above in a single-site manner

$$P(x,y|\rho) = \int \frac{d\hat{x}d\hat{y}}{4\pi^2} \exp(-ix\hat{x} - iy\hat{y}) \times \prod_j \left\langle\left\langle \exp\left(i\hat{x}\frac{J_j^S B_j^S(\boldsymbol{\xi})}{\sqrt{N}} + i\hat{y}\frac{J_j^T B_j^T(\boldsymbol{\xi})}{\sqrt{N}} \right)\right\rangle\right\rangle_{\boldsymbol{\xi}}. \tag{C5}$$

Since in this paper we restricted the Boolean functions to those that are antisymmetric, one can take the average over the input in two steps. The first step is to divide the inputs into two groups, $\boldsymbol{\xi}_+$ and $\boldsymbol{\xi}_-$, such that for any input vector in the first group there is the opposite one in the second group, and then to take the average over these two groups. In the case of $L=3$, the division may be $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3, \boldsymbol{\xi}_4$ from Table I as one group, and the other four as the other group. One then takes the average over one specific group, say, $\boldsymbol{\xi}_+$. Deriving the probability after taking only the first average yields

$$P(x,y|\rho) = \int \frac{d\hat{x}d\hat{y}}{4\pi^2} \exp(-ix\hat{x} - iy\hat{y}) \times \prod_j \left\langle\left\langle \cos\left[\frac{1}{\sqrt{N}} \{\hat{x} J_j^S B_j^S(\boldsymbol{\xi}) + \hat{y} J_j^T B_j^T(\boldsymbol{\xi})\} \right]\right\rangle\right\rangle_{\boldsymbol{\xi}_+}. \tag{C6}$$

Taking the expansion over the cosing function in the thermodynamic limit, one gets

$$P(x,y|\rho) = \int \frac{d\hat{x}d\hat{y}}{4\pi^2} \exp(-ix\hat{x} - iy\hat{y}) \times \exp\left\{ -\hat{x}^2 \frac{1}{2N}\sum_j (J_j^S)^2 - \hat{y}^2 \frac{1}{2N}\sum_j (J_j^T)^2 \right\} \times \exp\left\{ -\hat{x}\hat{y}\frac{1}{N}\sum_j J_j^T J_j^S \langle\langle B_j^T(\boldsymbol{\xi}) B_j^S(\boldsymbol{\xi})\rangle\rangle_{\boldsymbol{\xi}} \right\}. \tag{C7}$$

The result after introducing the definitions, Eq. (C3), and taking the integrals over $\hat{x}$, $\hat{y}$ is

$$P(x,y|\rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left[-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)} \right], \tag{C8}$$

the same function as for the perceptron. Therefore, the relation between ϵ_g and ρ is the same [Eq. (25)].

[1] A. Engel and C. Van den Broeck, *Statistical Mechanics of Learning* (Cambridge University Press, Cambridge, 2001).
[2] E. Gardner, J. Phys. A **21**, 257 (1988).
[3] E. Gardner and B. Derrida, J. Phys. A **22**, 1983 (1989).
[4] E. Barkai, D. Hansel, and I. Kanter, Phys. Rev. Lett. **18**, 2312 (1990).
[5] E. Barkai, D. Hansel, and H. Sompolinsky, Phys. Rev. A **45**, 4146 (1992).
[6] A. Engel, H.M. Köhler, F. Tschepke, H. Vollmayr, and A. Zippelius, Phys. Rev. A **45**, 5790 (1992).
[7] H. Schwarze and J. Hertz, Europhys. Lett. **20**, 375 (1992).
[8] R. Monasson and R. Zecchina, Phys. Rev. Lett. **75**, 2432 (1995).
[9] R. Urbanczik, Europhys. Lett. **35**, 553 (1996).
[10] A. Bethge, R. Kühn, and H. Horner, J. Phys. A **27**, 1929 (1994).
[11] M. Opper, Phys. Rev. Lett. **72**, 2113 (1994).
[12] M. Rosen-Zvi, A. Engel, and I. Kanter, Phys. Rev. Lett. **87**, 078101 (2001).
[13] G. Cybenko, Math. Control, Signals, Syst. **2**, 303 (1989).

[14] T.M. Cover, IEEE Trans. Electron. Comput. **EC-14**, 326 (1965).
[15] G.J. Mitchson and R.M. Durbin, Biol. Cybern. **60**, 345 (1989).
[16] G. Györgyi, Phys. Rev. A **41**, 7097 (1990).
[17] M. Opper and D. Haussler, Phys. Rev. Lett. **66**, 2677 (1991).
[18] M.S. Seung, H. Sompolinsky, and N. Tishby, Phys. Rev. A **45**, 6056 (1992).
[19] A. Priel, M. Blatt, T. Grossman, E. Domany, and I. Kanter, Phys. Rev. E **50**, 577 (1994).
[20] W. Krauth and M. Mezard, J. Phys. (Paris) **50**, 3057 (1989).
[21] H. Gutfreund and Y. Stein, J. Phys. A **23**, 2613 (1990).
[22] I. Kanter, Europhys. Lett. **17**, 181 (1992).
[23] T. Grossman, R. Meir, and E. Domany, Complex Syst. **2**, 555 (1989).
[24] M. Biehl and P. Riegler, Europhys. Lett. **28**, 525 (1994).
[25] R. Erichsen and W.K. Theumann, J. Phys. A **26**, L61 (1993).
[26] R. Majer, A. Engel, and A. Zippelius, J. Phys. A **26**, 7405 (1993).
[27] G. Györgyi and P. Reimann, Phys. Rev. Lett. **79**, 2746 (1997).