J. Phys. A: Math. Gen. 25 (1992) 367–380. Printed in the UK

On the capacity of neural networks with binary weights

I Kocher and R Monasson

Laboratoire de Physique Théorique de l'Ecole Normale Supérieure[†], 24 rue Lhomond, 75231 Paris Cedex 05, France

Received 26 April 1991

Abstract. We study the critical capacity (α_c) of multilayered networks with binary couplings. We show that, for any network presenting a tree-like architecture after the first hidden layer, no fixed internal representation is required. Using Gardner's calculations, we apply statistical mechanics to the simplest network with two layers of adaptive weights. Following the same approach as for the binary perceptron we find from the zero-entropy point a critical capacity $\alpha_c = 0.92$. We discuss the validity of this result in view of exhaustive search simulations on small networks.

1. Introduction

Over the past few years, a lot of work has been devoted to the storage capacity of neural networks. The simplest feedforward network, namely the perceptron with real synaptic weights [1], has been studied using statistical mechanics tools developed by Gardner [2, 3], which allow the computation of the number of independent random patterns which can be stored in this network. Nevertheless, such a network is unable to solve nonlinearly separable problems. Thus, one has to consider more complicated architectures, including hidden units, which are much more interesting from both biological and computational points of view.

Unfortunately, nobody has succeded until now in finding capacities of neural networks which are really multilayered, i.e. with free couplings between the first hidden layer and the output. In order to get around this problem, a useful idea has been to choose *a priori* the internal representation: one freezes the weights below the first hidden layer, building a specific decoder which matches these hidden units and the output [4]. The computation is then reduced to the capacity of a single perceptron (input-first hidden layer), which produces the desired internal representation in the first hidden layer. The most studied internal representation is the parity representation, where the output is the product of the hidden units [4–7]. In this paper we shall show that the computation of the capacity of some really multilayered networks with binary weights needs no arbitrary internal representation choice; this one is indeed already imposed by the network's architecture itself.

We propose here to consider networks with one hidden layer and binary weights. Such weights are interesting from a practical standpoint because one can easily implement them in electronic circuits. They also represent the simplest way to take into account the notion of synaptic depth (the limited accuracy of the synaptic couplings), which is biologically motivated [8].

[†] Unité Propre de Recherche du CNRS, associée à l'Ecole Normale Supérieure et à l'Université de Paris-Sud.

We show in section 2 that, for this class of networks, the internal representations are automatically fixed, which allows the computation of statistical quantities. We will see that, more generally, in the case of networks presenting a tree-line architecture after the first hidden layer, the principle of the calculations is also valid. A wide range of neural networks is obviously dismissed by this constraint and many complicated tasks (as for instance the encoding problem) cannot be solved by tree-like structures. However, one hidden-layer networks are already able to implement usual Boolean functions (for example, the parity decoder) and their study remains of interest.

We then restrict our study to the case where there are three hidden units and non-overlapping receptive fields. Gardner's method allows us to compute the number of errors made by the best possible network as a function of the size of the training set (section 3).

We find some estimates of the critical capacity, particularly that given by the zero-entropy point within the replica symmetric approximation [8, 10]. We show, as in the case of the binary perceptron, that this value seems to be exact (section 4). However, numerical simulations are not in very good agreement with it (section 5) and lead to the conclusion that layered neural networks might exhibit richer overlap distributions than previously thought.

2. Networks with binary weights and internal representations

In this section we consider networks with one hidden layer and binary weights. Let A be such a network. The neurons are binary valued too, with zero thresholds.

Thus, each neuron σ is updated according to

$$\sigma = \operatorname{sgn}\left(\sum_i W_i S_i\right) \tag{2.1}$$

where $S_i = \pm 1$ are the neurons in previous layer and $W_i = \pm 1$ are the weights. We shall restrict ourselves to cases where the number of neurons in each layer is odd, in order to avoid ambiguities in (2.1).

The network we study is composed of one input layer of N neurons, one hidden layer of K neurons and one output layer. For simplicity, we restrict ourselves to the case where there is only one neuron in the output layer. K is finite, while the results we derive below are obtained in the large-N limit.

For l between 1 and K, we define the weights $W^0_{l,m}$ between the input neuron m $(1 \leqslant m \leqslant N)$ and the neuron l of the hidden layer $(1 \leqslant l \leqslant K)$ and the weights W^1_l between the neuron l of the hidden layer and the single output. For a given input ξ_m $(1 \leqslant m \leqslant N)$, the corresponding output is therefore

$$\sigma(\{W\}, \boldsymbol{\xi}) = \operatorname{sgn}\left[\sum_{l=1}^{K} W^1_l \operatorname{sgn}\left(\sum_{m=1}^{N} W^0_{l,m}\xi_m\right)\right]. \tag{2.2}$$

We consider here a network characterized by the couplings $\{W\}$, and we want to compute the capacity in the case of random patterns: each pattern presented to the network is a pair $(\boldsymbol{\xi}, \sigma)$ where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)$ is a configuration of the input layer, and σ is the desired output for this configuration. This pattern will be considered as stored if the output $\sigma(\{W\}, \boldsymbol{\xi})$ realized by the network coincides with the desired output σ.

In order to compute the capacity of this network, we present to it a set of P random patterns $(\boldsymbol{\xi}^\mu, \sigma^\mu)$ $(1 \leq \mu \leq P)$ and we define an energy which corresponds to the number of unstored patterns among the set [3]

$$E(\{W\}, \{\boldsymbol{\xi}\}) = \sum_{\mu=1}^{P} \theta(-\sigma(\{W\}, \boldsymbol{\xi}^\mu)\sigma^\mu) \tag{2.3}$$

where $\theta(x)$ is the Heaviside function. We then introduce a Boltzmann measure in the space of networks, characterized by the partition function

$$Z_\mathrm{A} = \sum_{\{W\}} \exp(-\beta E(\{W\}, \{\boldsymbol{\xi}\})) \tag{2.4}$$

where β is an auxiliary parameter which plays the same role as the inverse of a temperature. The minimum number of mistakes which can be realized by the best possible network is the internal energy (or the free energy f) at the limit $\beta \to \infty$. Therefore, one wants to compute the free energy

$$-\beta f_\mathrm{A} = \lim_{N\to\infty} \frac{1}{N} \overline{\ln Z_A} \tag{2.5}$$

where the overbar denotes the average over the pattern distribution [3].

Due to the choice of binary weights, to each path (m, l) from the input to the output, we can associate an effective coupling

$$J(l, m, \{W\}) = W_l^1 W_{l,m}^0. \tag{2.6}$$

We obtain from (2.2)

$$\sigma(\{W\}, \boldsymbol{\xi}) = \mathrm{sgn}\left[\sum_{l=1}^{K} \mathrm{sgn}\left(\sum_{m=1}^{N} J(l, m, \{W\})\xi_m\right)\right]. \tag{2.7}$$

A straightforward computation leads to $Z_\mathrm{A} = 2^K Z_\mathrm{B}$ with

$$Z_\mathrm{B} = \sum_{\{J=\pm 1\}} \exp\left(\sum_{\mu=1}^{P} \theta(-\sigma(\{J_{l,m}\}, \boldsymbol{\xi}^\mu)\sigma^\mu)\right). \tag{2.8}$$

Z_B is the partition function of the network B defined as follows: it has the same architecture as A; all the weights beyond the hidden layer are fixed to unity and form a decoder which imposes an internal representation to B; the $\{J\}$ couplings between the input and the hidden layer are binary.

As K remains finite when $N \to \infty$, A and B have exactly the same free energies and therefore the same properties.

We have actually proved that any network with two layers of adaptative binary weights may be related to another one with a given internal representation, whose study is much easier.

Let us notice that the above argument may be repeated with weights equal to $-1, 0, 1$. This allows us to choose incompletely connected networks for A. It is thus possible to generalize the above result to a more general class of networks, composed of all the multilayered networks without overlapping fields after the first hidden layer.

Such networks can be fully connected between the input and the first hidden layer, and have a tree architecture between this first hidden layer and the output. The important fact is that, for each cell of the hidden layer, there is one and only one synaptic path to reach the output. They are composed of one input layer of N neurons, $H-1$ hidden layers, and one single output.

Let k_h $(1 \leqslant h \leqslant H)$ be the number of hidden cells in the hth layer. The input layer (numbered zero) has $k_0 = N$ neurons, while the output layer has $k_H = 1$. Each k_h $(h \geqslant 1)$ is finite.

For the layer $h \leqslant H-1$, the architecture of the network defines a partition $\{C_1^h, \ldots, C_{k_{h+1}}^h\}$ of the k_h neurons of the layer: each neuron l $(l = 1, \ldots, k_{h+1})$ of the layer $h+1$ is connected to the neurons of C_l^h, and only to them (this is indeed compulsory for $h \geqslant 1$). We define as $W_{l,m}^h$ the weight between the neuron m of the layer h $(m \in C_l^h)$ and neuron l of the layer $h+1$ $(1 \leqslant l \leqslant k_{h+1})$. The case $H = 1$ gives the binary perceptron.

To each path $\{l\} = l_0, l_1, \ldots, l_H$ *from the input to the output we can associate an* effective coupling

$$J(\{l\},\{W\}) = W_{1,l_{H-1}}^{H-1} W_{l_{H-1},l_{H-2}}^{H-2} \cdots W_{l_1,l_0}^{0}. \tag{2.9}$$

We obtain from (2.9)

$$\sigma(\{W\},\boldsymbol{\xi}) = \mathrm{sgn}\left[\!\!\left[\sum_{l_{H-1}=1}^{k_{H-1}} \mathrm{sgn}\left\{ \sum_{l_{H-2}=1}^{k_{H-2}} \mathrm{sgn}\left[\ldots \mathrm{sgn}\left(\sum_{l_0=1}^{N} J(\{l\},\{W\})\xi_{l_0} \right) \right] \right\} \right]\!\!\right]. \tag{2.10}$$

We see obviously that the network has the same partition function as the one where all weights between the first hidden layer and the output are fixed to unity (with a proportionality factor 2^K where $K = \Sigma_{h=1}^{H-1} k_h$ is finite). We can conclude that any binary network which has a tree architecture after the first hidden layer is equivalent to a two-layered binary network with a particular internal representation, which is fixed by the architecture of the initial network.

In the following, we illustrate this property on the simplest tree-like network, which contains only one hidden layer.

3. A special case: a two-layered network with non-overlapping fields

3.1. Presentation of the network

The network we study here in more detail is given in figure 1. It has one hidden layer of K units. Each of them is connected to N/K input neurons [5-7]. Following the idea developed in section 2 we can fix the weights between the hidden units and the output σ to $+1$. Thus, we obtain

$$\sigma(\boldsymbol{\xi}) = \mathrm{sgn}\left[\sum_{l=1}^{K} \mathrm{sgn}\left(\sum_{i=1}^{N/K} J_{i,l}\xi_{i,l} \right) \right] \tag{3.1}$$

where the $\boldsymbol{J}_l$ and $\boldsymbol{\xi}_l$ are, respectively, the couplings and the input patterns of the K disconnected perceptrons.

We define the capacity α per synapse by

$$\alpha = \frac{P}{N} \tag{3.2}$$

where P is the number of presented patterns $\boldsymbol{\xi}^\mu = (\xi_1^\mu, \xi_2^\mu, \ldots, \xi_K^\mu)$, $\mu = 1, \ldots, P$. All $\xi_{i,l}^\mu$ and σ^μ are unbiased random binary variables.

One sees obviously that σ is the most frequent sign among the hidden cells σ_l, $(l = 1, \ldots, K)$. Any two-layered network with binary weights obeys, therefore, a majority rule.

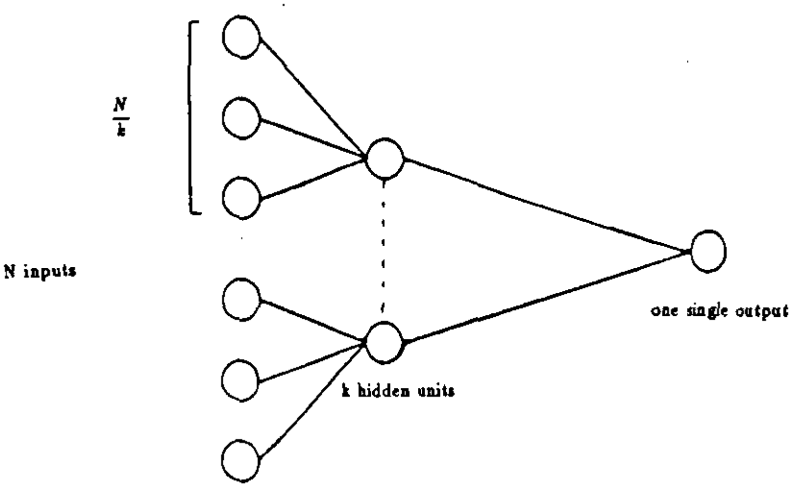

Figure 1. The network which we study has one hidden layer of K units. Each of them is connected to N/K input neurons. Notice that K is odd so that the field incoming onto the output is not vanishing.

3.2. Computation of the partition function

Using the replica method, we have to compute the nth powers of Z defined in (2.8) and average over the patterns.

Since inputs of different hidden units are independent [4, 5], the only kind of parameter is

$$q_l^{a,b}=\frac{K}{N}\,\overline{\langle \boldsymbol{J}_l^a\cdot \boldsymbol{J}_l^b\rangle} \tag{3.3}$$

where the brackets denote a thermal average over the $\boldsymbol{J}$ weights. $q_l^{a,b}$ is the typical overlap between the couplings incoming into the hidden neuron l in replicas a and b (a and b are replica indices which run from 1 to n).

Introducing conjugate parameters $\hat{q}_l^{a,b}$ we find

$$\overline{Z^n}=\int\prod_{l,a<b}\left(\frac{\mathrm{d}q_l^{a,b}\,\mathrm{d}\hat{q}_l^{a,b}}{2\pi}\right)\exp[N(g_0+g_J+g_\beta)] \tag{3.4}$$

where

$$\begin{aligned}
g_0&=-\frac{1}{K}\sum_{l,a<b}\hat{q}_l^{a,b}q_l^{a,b}\\
g_J&=\frac{1}{K}\ln\left[\sum_{\{J_l^a=\pm1\}}\exp\left(\sum_{l,a<b}\hat{q}_l^{a,b}J_l^aJ_l^b\right)\right]\\
g_\beta&=\alpha\ \ln\left\{\int\prod_{l,a}\left(\frac{\mathrm{i}\mathrm{d}t_l^a\,\mathrm{d}\hat{t}_l^a}{2\pi}\right)\exp\left(\sum_{a,l}\hat{t}_l^at_l^a-\beta\sum_a\theta\left(-\sum_l\operatorname{sgn}(t_l^a)\right)\right.\right.\\
&\qquad\left.\left.+\tfrac{1}{2}\sum_{a,l}(\hat{t}_l^a)^2+\sum_{l,a<b}q_l^{a,b}\hat{t}_l^a\hat{t}_l^b\right)\right\}.
\end{aligned} \tag{3.5}$$

Finally, we determine the free energy f by

$$-\beta f=\lim_{n\to0,N\to\infty}\frac{\overline{Z^n}-1}{nN} \tag{3.6}$$

3.3. Replica symmetric and replica symmetry broken calculations

In the following we will note the two solutions as, respectively, RS and RSB. We give here the free energy of the network, up to one step of replica symmetry breaking.

We consider first the RS approximation where for all $a \neq b$, and for all $l = 1, \dots, K$, $q_l^{a,b} = q$. We get rid of the l dependence since, after averaging over disorder, all units play the same role [4–7].

We compute the above quantities g_0, g_J and g_β and find afterwards the saddlepoint of (3.4):

$$-\beta f_{\mathrm{RS}} = \mathrm{Max}_{q,\hat{q}} \left[\!\!\left[-\tfrac{1}{2}\hat{q}(1-q) + \int \mathrm{D}z \ln[2\cosh(z\sqrt{\hat{q}})] + \alpha \int \sum_{l=1}^{K} \mathrm{D}x^l \right.\right.$$
$$\left.\left. \times \ln\left\{ \mathrm{e}^{-\beta} + (1-\mathrm{e}^{-\beta}) \int \prod_{l=1}^{K} \mathrm{D}u^l \theta \left[\sum_{l=1}^{K} \mathrm{sgn}\left(u^l - \sqrt{\frac{q}{1-q}}\, x^l \right) \right] \right\} \right]\!\!\right] \tag{3.7}$$

where

$$\mathrm{D}z = \frac{\mathrm{d}z}{\sqrt{2\pi}}\, \mathrm{e}^{-z^2/2} \tag{3.8}$$

is the Gaussian measure.

With one step of replica symmetry breaking, one has five parameters q_0, $\hat{q}_0$, q_1, $\hat{q}_1$, m [10] and

$$-\beta f_{\mathrm{RSB}} = \mathrm{Max}_{q_0,\hat{q}_0,q_1,\hat{q}_1,m} \left[\!\!\left[\frac{m}{2}\, q_0\hat{q}_0 - \tfrac{1}{2}\hat{q}_1[1+(m-1)q_1] \right.\right.$$
$$+ \frac{1}{m} \int \mathrm{D}z \ln\left(\int \mathrm{D}x[2\cosh(z\sqrt{\hat{q}_0} + x\sqrt{\hat{q}_1 - \hat{q}_0})]^m \right)$$
$$+ \frac{\alpha}{m} \int \prod_{k=1}^{K} \mathrm{D}x^k \ln\left\{ \int \prod_{j=1}^{K} \mathrm{D}y^j \left[\mathrm{e}^{-\beta} + (1-\mathrm{e}^{-\beta}) \int \prod_{i=1}^{K} \mathrm{D}t^i \right.\right.$$
$$\left.\left.\left. \times\, \theta\left(\sum_{l=1}^{K} \mathrm{sgn}(t^l - x'^l - y'^l) \right) \right]^m \right\} \right]\!\!\right] \tag{3.9}$$

with

$$x'^l = \sqrt{\frac{q_0}{1-q_1}}\, x^l \qquad y'^l = \sqrt{\frac{q_1 - q_0}{1-q_1}}\, y^l.$$

4. Some estimates of the critical capacity

In this section, we try to predict the theorical critical capacity from the previous calculations. In order to have numerical values we choose $K = 3$, which is the lowest possible integer allowing non-vanishing field on the output.

4.1. Bounds on α_c

Since the number of binary synapses is N, the information theory tells us that

$$\alpha_c \leqslant 1. \tag{4.1}$$

This upper bound can easily be obtained from the annealed approximation [10], which consists of calculating $\overline{\ln Z} \simeq \ln \overline{Z}$.

One can also look for lower bounds of α_c.

Let us consider the binary perceptron $\{W\}$, including N input neurons and fed with $P = \alpha N$ random patterns ξ^μ. As N grows to infinity we define $f(\alpha)$ as the highest fraction of the P patterns which can be stored.

With the notation of section 2

$$f(\alpha) = 1 - \frac{1}{P} \min_{\{W\}}(E(\{W\}, \{\xi\}). \tag{4.2}$$

Obviously, if $\alpha \leqslant \alpha_0$ (α_0 is the capacity of the binary perceptron [10] and is assumed to be nearly equal to 0.83), one obtains $f(\alpha) = 1$.

For $\alpha > \alpha_0$, there exists $\{W_0\}$ storing exactly all the ξ^μs ($\mu \leqslant \alpha_0 N$). Since the pattern distribution is random, half of the remaining ξ^μs ($\mu > \alpha_0 N$) are stored by $\{W_0\}$. This algorithm leads to

$$f(\alpha) \geqslant p(\alpha) = \frac{\alpha_0 N + \frac{1}{2}(P - \alpha_0 N)}{P} = \frac{\alpha + \alpha_0}{2\alpha}. \tag{4.3}$$

Moreover, choosing α and α' such that $\alpha' \leqslant \alpha_0 \leqslant \alpha$, we define $g(\alpha, \alpha')$ as the highest storable fraction of the $P = \alpha N$ patterns when imposing the criterion that the $\alpha' N$ first patterns must be stored. We have

$$f(\alpha) \geqslant g(\alpha, \alpha') \geqslant p(\alpha) \geqslant \frac{\alpha'}{\alpha}. \tag{4.4}$$

Now, we are looking to store $P = \alpha N$ patterns in our two-layered network. We number as A_1, A_2 and A_3 the three perceptrons of $N/3$ inputs, whose outputs σ_1, σ_2 and σ_3 are the three hidden cells. One pattern $\sigma^\mu \xi^\mu$ is stored if at least two σ_is are equal to $+1$.

Renumbering the patterns, A_1 stores the ξ^μs, $\mu \leqslant P_1 = \alpha_1 N$ where $P_1 = f(3\alpha)P$. Thus,

$$\alpha_1 = \alpha f(3\alpha). \tag{4.5}$$

As a consequence, A_2 must store the patterns $\mu > P_1$. This is possible if $P - P_1 \leqslant \alpha_0 N/3$:

$$f(3\alpha) \geqslant 1 - \frac{\alpha_0}{3\alpha}. \tag{4.6}$$

When this condition is satisfied, all patterns $\mu > P_2 = \alpha_2 N$ may be stored by A_2 with $P - P_2 = g(3\alpha, 3\alpha - 3\alpha_1)P$ and

$$\alpha_2 = \alpha(1 - g(3\alpha, 3\alpha - 3\alpha_1)). \tag{4.7}$$

We see in figure 2 that A_3 can store the patterns $\mu \leqslant P_2$ and $\mu > P_1$, provided that $P_2 + (P - P_1) \leqslant \alpha_0 N/3$. Using (4.5) and (4.7),

$$f(3\alpha) + g[3\alpha, 3\alpha(1 - f(3\alpha)] \geqslant 2 - \frac{\alpha_0}{3\alpha}. \tag{4.8}$$

First, we notice that, for $\alpha \leqslant \alpha_0$, $p(3\alpha) \geqslant 1 - \alpha_0/3\alpha$. From (4.3) we conclude that condition (4.6) is verified up to $\alpha = \alpha_0$.

Furthermore, referring to (4.3) and (4.4), as long as we have

$$p(3\alpha) + p(3\alpha) \geqslant 2 - \frac{\alpha_0}{3\alpha} \tag{4.9}$$

condition (4.8) will be true. Solving (4.9), we find $\alpha \leqslant \frac{2}{3}\alpha_0$.

<table>
<tr><td>$\frac{\mu}{N}$</td><td colspan="3">0 α_2 α_1 α</td></tr>
<tr><td>σ_1</td><td colspan="2">+</td><td>-</td></tr>
<tr><td>σ_2</td><td>−</td><td colspan="2">+</td></tr>
<tr><td>σ_3</td><td>+</td><td>?</td><td>+</td></tr>
</table>

Figure 2. The outputs σ_l of the three perceptrons A_l for the patterns $\sigma^\mu \xi^\mu$ ($\mu = 1 \ldots \alpha N$). *Since our network obeys a majority rule at least two outputs must be positive for each* μ. One easily sees conditions (4.6) and (4.8) on the diagram. Notice that, as σ_3 is random between α_2 and α_1, our algorithm cannot reach the highest capacity α_c.

The lower bound of α_c is therefore

$$\alpha_c \geqslant \tfrac{2}{3}\alpha_0 \simeq 0.55. \tag{4.10}$$

4.2. *Analysis of the* RS *approximation*

Starting from the free energy given in (3.7), we see that the zero-temperature energy (i.e. the number of unstored patterns) is zero as long as $q < 1$. A possible critical capacity is thus α_E, the lowest α for which q reaches 1.

We find

$$\alpha_E = \frac{16}{5\pi - 6 - 2\sqrt{3}} \simeq 2.56. \tag{4.11}$$

This value being inconsistent with (4.1), the RS solution surely becomes wrong below α_E. We have studied its local stability with regard to transverse fluctuations [2, 3, 11] (longitudinal stability is always verified up to α_E). We have found that the RS solution is locally stable up to

$$\alpha_{AT} \simeq 1.3. \tag{4.12}$$

We can look for the zero-temperature entropy S_{RS}, which is the logarithm of the number of couplings storing all the patterns. As soon as it becomes negative, the RS solution is wrong. This occurs for $\alpha > \alpha_S$ where

$$\alpha_S \simeq 0.92. \tag{4.13}$$

The curves $q(\alpha)$ and $S_{RS}(\alpha)$ are shown in figures 3(a) and 3(b).

4.3. *The one-step* RSB *solution*

We proceed exactly as in the binary perceptron case and look for solution for (3.5) with one stage of replica symmetry breaking in the hierarchical scheme of Parisi [9]. As we seek for the critical capacity, we consider the case $q_1 \to 1$ where

$$q_1 = \frac{K}{N} \overline{\langle J \rangle_\gamma^2} \tag{4.14}$$

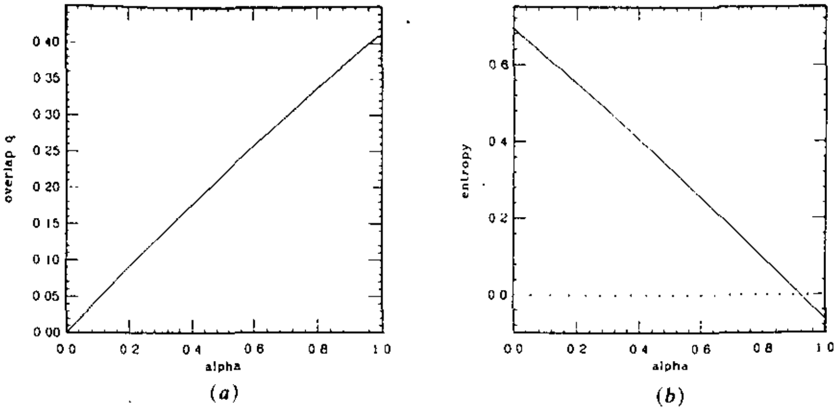

Figure 3. Evolution with α of (a) the order parameter q and (b) the entropy within the RS approximation. The entropy vanishes for $\alpha_S \simeq 0.92$ and we obtain $q(\alpha_S) \simeq 0.38$.

is the overlap inside one pure state γ and look for the saddlepoint over

$$q_0 = \frac{K}{N} \overline{\langle J \rangle_\gamma \langle J \rangle_\delta} \tag{4.15}$$

which is the overlap between the average solutions of two different valleys γ and δ.

The saddlepoint condition over $\hat{q}_1$ gives $\hat{q}_1 \to \infty$ when $q_1 \to 1$ and, from (3.9), we obtain

$$f_{\mathrm{RSB}}(q_0, \hat{q}_0, 1, \infty, m, \beta) = \frac{1}{m} f_{\mathrm{RS}}(q_0, m^2 \hat{q}_0, \beta m). \tag{4.16}$$

We deduce from this equality a phase diagram, shown in figure 4. It indicates that the (α, T) plane is divided into two parts separated by the line $T_c(\alpha)$ defined by

$$S_{\mathrm{RS}}(\alpha, T_c(\alpha)) = 0. \tag{4.17}$$

In the first area $S_{\mathrm{RS}} > 0$ and the RS solution seems to be correct. In the second area one needs one step of breaking. We get $S_{\mathrm{RSB}}(\alpha, T) = 0$ and

$$m = 1 - \sum_\gamma \overline{P_\gamma^2} = \frac{T}{T_c(\alpha)} \tag{4.18}$$

where P_γ is the weight of the pure state numbered γ. This analysis (see [6, 7, 10]) leads us to believe that the critical capacity is α_S given in (4.13).

5. Numerical simulations

From a numerical point of view, dealing with binary networks is much harder than for continuous ones. Already in the perceptron case there is no reliable algorithm (guaranteed to converge if there exists a solution). We have therefore decided to use exact enumeration methods. But such exhaustive scannings forbid large-size systems and $N \simeq 25$ is a typical upper limit for the number of input neurons [12, 13]. In the present case, the number of neurons connected to each hidden unit must be odd. Only simulations with $N = 9$, 15 or 21 are allowed, if one wants to average the results over a reasonable number of samples.

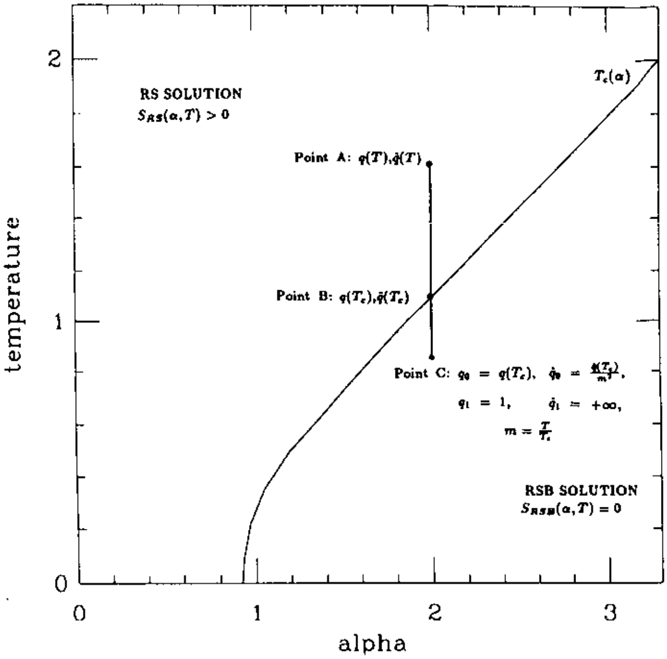

Figure 4. Phase diagram (α, T). The plane is divided into two parts separated by the line $T_c(\alpha)$. Beyond T_c the RS solution seems to be exact while, below, one needs one step of replica symmetry breaking.

5.1. *First simulation*

The first approach we resort to is the one given in [6]. Choosing randomly Q samples of $P=\alpha N$ patterns, one counts the fraction $f_N(\alpha, Q)$ which can be stored (i.e. for which there exist suitable couplings). As a function of α, $f_N(\alpha)$ (obtained with large Q) decreases from 1 ($\alpha=0$) to 0 (large α). In the large-N limit one expects

$$\lim_{N\to\infty} f_N(\alpha)=\theta(\alpha_c-\alpha). \tag{5.1}$$

A simple estimate of the critical capacity α_c is thus given by α_N defined by $f_N(\alpha_N)=\frac{1}{2}$.

The simulations we did with binary patterns exhibit big fluctuations for $N=21$. However, we checked that the slopes of f_N were increasingly sharper and α_N decreased with N. After averaging over Q samples equal respectively to 10 000, 1000 and 100, we found

$$\begin{aligned}\alpha_9&=0.93\pm0.004\\ \alpha_{15}&=0.90\pm0.006\\ \alpha_{21}&=0.87\pm0.015.\end{aligned} \tag{5.2}$$

Although these values are not in good agreement with the prediction $\alpha_c=0.92$, one must consider finite-size effects, which might be important. Even in the binary perceptron case, data obtained from simulations with binary patterns up to $N=21$ extrapolate to a value lower than 0.83 [13].

5.2. Second simulation

In order to reduce finite-size effects, we now choose Gaussian patterns, for which better results have been obtained for the perceptron [12, 13] (from the replica method one expects that Gaussian or binary patterns lead to the same values of the storage capacity α_c). For one pattern $\boldsymbol{\xi}=(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3)$, and one network $\boldsymbol{J}=(\boldsymbol{J}_1, \boldsymbol{J}_2, \boldsymbol{J}_3)$, we compute the stabilities of the three independent inputs

$$K_l=\frac{\boldsymbol{J}_l \cdot \boldsymbol{\xi}_l}{\|\boldsymbol{J}_l\| \cdot \|\boldsymbol{\xi}_l\|} \qquad l=1,2,3. \tag{5.3}$$

Ordering the K_l so as to obtain $K_1 \leqslant K_2 \leqslant K_3$, we define

$$K(\boldsymbol{J}, \boldsymbol{\xi})=K_2. \tag{5.4}$$

As our network follows a majority rule, we see that $\boldsymbol{J}$ stores the pattern $\boldsymbol{\xi}$ (i.e. $\boldsymbol{J} \cdot \boldsymbol{\xi} > 0$) if and only if $K(\boldsymbol{J}, \boldsymbol{\xi}) > 0$. For one sample $S=\{\boldsymbol{\xi}^\mu\}$ ($\mu=1, \ldots, P$) consider the optimal stability $K_{\text{opt}}(S)$, which is positive if there exists one set of couplings suitable for the whole sample:

$$K_{\text{opt}}(S)=\max_{\boldsymbol{J}}[\min_\mu(K(\boldsymbol{J}, \boldsymbol{\xi}^\mu))]. \tag{5.5}$$

using an exact enumeration based on the Gray code [12], we plot the distribution of $K_{\text{opt}}(S)$ for fixed α and N.

The curves corresponding to $N=9$, 15 and 21 are given in figure 5. The numbers of samples are, respectively, equal to 10^5, 10^4 and 1500.

First, we notice that $K_{\text{opt}}(S)$ may be relatively well fitted by a Gaussian, whose variance scales roughly as $1/N$. This strongly indicates that, in the large-N limit, one obtains sharply peaked distributions and thus provides a good indication for the critical capacity [13].

We show in figure 6 the fitted curves $K_{\text{opt}}(\alpha)$ (i.e. the average of $K_{\text{opt}}(S)$ over S) for the three values of N under study. Their intersections with the axis give us new estimates α_N of α_c. We find

$$\begin{aligned} \alpha_9 &= 0.902 \pm 0.001 \\ \alpha_{15} &= 0.896 \pm 0.002 \\ \alpha_{21} &= 0.898 \pm 0.003. \end{aligned} \tag{5.6}$$

As for the binary perceptron, these values are higher than the ones given by binary patterns. In fact, no rigorous proof of the equality of the critical capacities obtained with Gaussian and Ising inputs is available.

One sees that the numerical results are slightly lower than the expected value. However, the relative error over α_{21} does not allow us to conclude whether critical capacities obtained for finite N are decreasing or not. In the first case, which would be similar to the perceptron one [12, 13], this would indicate that the storage capacity is not 0.92. In the second case, one could expect important finite-size effects in multilayered networks, even with Gaussian patterns.

6. Conclusion

In this paper we have focused on the storage capacity of multilayered networks with binary weights. We have shown that all networks with tree-like structures after the first

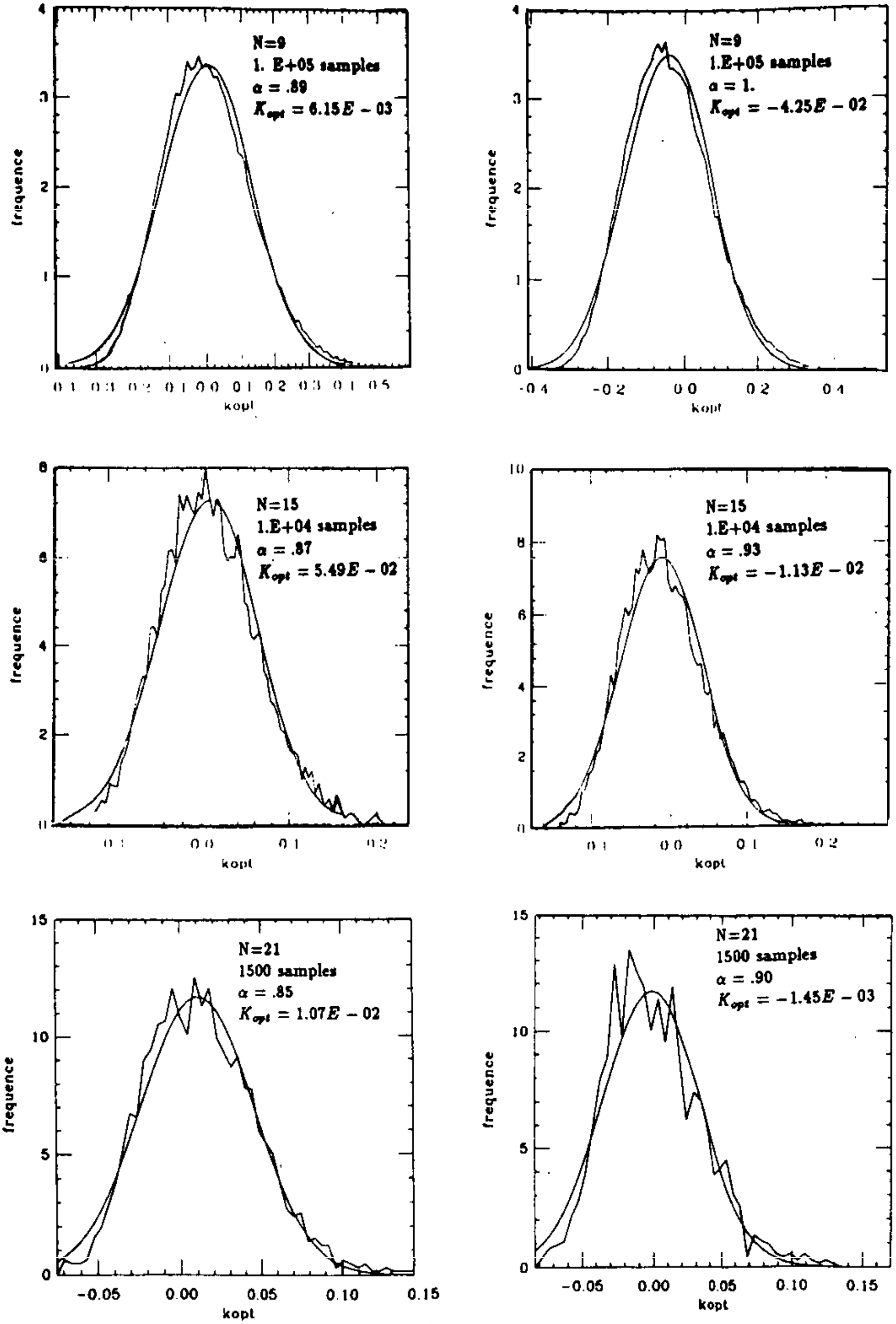

Figure 5. The distribution of $K_{\rm opt}(S)$: (a) for $N = 9$; (b) for $N = 15$; (c) for $N = 21$. For each size N, the values of α we have chosen are as close as possible to 0.92.

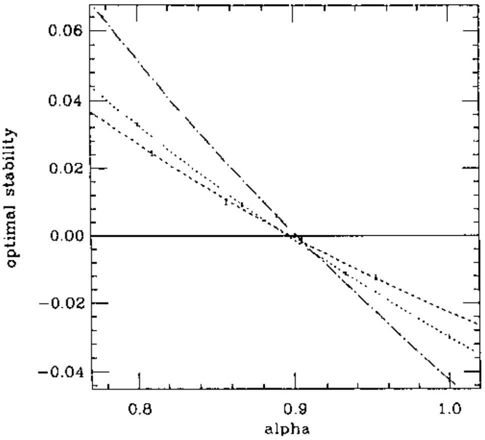

Figure 6. Evolution of K_{opt} with the size of the training set α, for $N=9$ (chain line), 15 (dotted line) and 21 (broken line). These curves are the best quadratic fits obtained from the numerical results.

hidden layer may be exactly studied, without fixing an internal representation *a priori*. We have illustrated this property for the simplest two-layered network with non-overlapping receptive fields, which works as a majority decoder (calculations are indeed feasible for any tree-like network: one will obtain more complicated decoders).

Applying statistical mechanics tools developed recently [2, 10] we found that this network exhibits the same behaviour as the binary perceptron. Its zero-temperature entropy computed within the RS approximation vanishes for $\alpha_S = 0.92$ and a complete freezing occurs, described by one step of replica symmetry breaking. This result is interesting since it suggests that a small modification of the architecture (three added neurons) may lead to a substantial improvement of the storage capacity per synapse of the network ($\alpha_c = 0.83$ for the perceptron).

In order to check this estimate, numerical simulations have been carried out. Up to N, the number of input neurons, equal to 21 they give values which are lower than 0.92. So as to elucidate this situation, one must take into account finite-size effects. Recent studies [13] have stressed their importance, especially in the case of binary patterns. When dealing with Gaussian patterns, however, all the results obtained for the perceptron may be extrapolated to the zero-entropy point with good accuracy [12, 13]. We have shown this not to be the case for our majority network. But one cannot exclude that, for $N > 21$, the α_N values increase up to 0.92 (non-monotonous variations with Gaussian patterns may occur in other problems like the generalization for the binary perceptron [13]). So, without asserting that 0.92 is a wrong value, we may doubt that the zero-entropy point provides us with the correct α_c. If this were the case the solution with one step of replica symmetry breaking would be wrong. One should thus attempt to compute up to two steps of breaking, i.e. to look for a more complicated saddlepoint.

Acknowledgments

We thank M Mézard for numerous stimulating discussions and his critical comments on the manuscript. We also thank W Krauth for his help concerning numerical simulations and M Biehl for useful remarks.

References

[1] Minsky M L and Papert S 1969 *Perceptrons* (Cambridge, MA: MIT Press)
[2] Gardner E 1988 *J. Phys. A: Math. Gen.* **21** 257
[3] Gardner E and Derrida B 1988 *J. Phys. A: Math. Gen.* **21** 271
[4] Mézard M and Patarnello S 1989 Unpublished findings
[5] Barkai E, Hansel D and Kanter I 1990 *Phys. Rev. Lett.* **65** 2312
[6] Barkai E and Kanter I 1991 *Europhys. Lett.* **14** 107
[7] Kanter I 1991 Information theory of a multilayer neural network with discrete weights *Preprint* Bar-Ilam University, 11/90
[8] Gutfreund H and Stein Y 1990 *J. Phys. A: Math. Gen.* **23** 2613
[9] Mézard M, Parisi G and Virasoro M A 1987 *Spin Glass Theory and Beyond* (Singapore: World Scientific)
[10] Krauth W and Mézard M 1989 *J. Physique* **50** 3057
[11] de Almeida J R L and Thouless D J 1978 *J. Phys. A: Math. Gen.* **11** 983
[12] Krauth W and Opper M 1989 *J. Phys. A: Math. Gen.* **22** L519
[13] Derrida B, Griffiths R B and Prugel-Bennet A 1991 *J. Phys. A: Math. Gen.* **24** 4907