Explicit symmetries and the capacity of multilayer neural networks

1994 J. Phys. A: Math. Gen. 27 2719

(http://iopscience.iop.org/0305-4470/27/8/010)

J. Phys. A: Math. Gen. 27 (1994) 2719–2734. Printed in the UK

Explicit symmetries and the capacity of multilayer neural networks

D Saad

Department of Physics, The University of Edinburgh, Edinburgh EH9 3JZ, UK

Received 6 December 1993

Abstract. Calculating the capacity and generalization capabilities of feed-forward multilayer neural networks requires the use of replica-symmetry-breaking methods, making the calculation practically unfeasible. Replica symmetry is broken because the configuration space is disconnected, which is clearly the case in the capacity limit where the configuration space shrinks to isolated points. Moreover, there is no knowledge about the number of replica-symmetry-breaking steps required to obtain reliable results. Novel approaches to tackle the capacity calculation of feed-forward neural networks avoiding the use of replica-symmetry-breaking methods are presented in this paper. The basic idea behind these approaches is that breaking explicit symmetries of the network prior to the capacity calculation itself restores order-parameter symmetry, at least to a good approximation, and therefore enables the use of the replica-symmetry ansatz. Two methods are presented for breaking the explicit symmetries and restoring replica symmetry; one restricts relations between the various weight elements while the other restricts the values of the order parameters. These methods, which are demonstrated in this work via the capacity calculation of feed-forward neural networks, are applicable to a variety of capacity, learning and generalization capability calculations of such nets. We examine an approximation for carrying out the multi-dimensional Gaussian integrals appearing during the calculation as well as exact results for some simple cases. Numerical results obtained for nets with one to six hidden neurons using the downhill simplex and adaptive simulated-annealing optimization algorithms are in good agreement with simulation results.

1. Introduction

The powerful techniques of statistical mechanics have been used for some time to formulate and calculate properties of simplified neural network models. This work has been related mostly to the single-layer perceptron [1, 2] due to its simplicity and the feasibility of the analysis (for a review see [3]). One of the most important of these methods is the replica method, applicable for many of the calculations examining properties of perceptrons, as well as some simple multilayer configurations [7–9]. The method involves replacing the average over the configuration space by an average over a set of replicas of that space, defining 'order parameters' which represent statistical characteristics of the configuration space. The method is particularly successful and applicable when these order parameters obey some symmetry restrictions, in particular when replica symmetry itself is exact or a good approximation. One of the main limitations of this method for analysing multilayer perceptrons is that this symmetry is not valid, resulting in the use of complicated structures of broken replica symmetry [4]. Replica-symmetry-breaking methods, though useful for a variety of calculations, make the calculation rather difficult and hardly give any indication for the number of replica-symmetry-breaking steps required for obtaining a reliable result.

0305-4470/94/082719+16$19.50

The broken symmetry results from discontinuities in the configuration space which are exposed most clearly in the capacity limit where *single points* in configuration space represent solution parameters to the desired input/output mapping. The same problems appear in learning and generalization tasks below a certain temperature and after learning a large number of training patterns, where the configuration space becomes disconnected and replica symmetry is broken.

In this paper we introduce two novel methods, one for removing the main cause for replica-symmetry breaking in multilayer neural networks and one for enforcing replica symmetry using a microcanonical distribution. The idea is to break *explicit* (and *redundant*) symmetries that exist in these nets prior to the calculation itself. This makes the replica method more applicable for exploring properties of such nets. The two methods introduced in this paper result in an identical end expression and thus equivalent results.

Examining the two-layer perceptron with N neurons in the input layer, M hidden units and a single output neuron, all with a binary activation function, one easily notices that there are several internal symmetries within the group of nets capable of implementing a certain input–output mapping (we shall call them solution nets for a particular mapping). We assume that for a given input–output relation one obtains a certain solution represented by the set of weights w_{ij} connecting input nodes (represented by the index j) to the hidden units (represented by the index i) and the weight vector with components u_i connecting the hidden units to the output unit. This solution is obviously not unique since applying either of the following group operators will produce another solution which is in general different from the initial one.

- Hidden-node permutation in which $w_{ij} \rightarrow w_{kj}$ and $u_i \rightarrow u_k$, where k is an index reorganization. Obviously there are $M!$ operators in this permutation group and therefore $M!$ different solutions are produced from each and every single solution.
- Hidden-node reflection, so that $w_{ij} \rightarrow -w_{ij}$ and $u_i \rightarrow -u_i$ for a certain sub-group of the hidden nodes I, where $i \in I$. This reflection group includes 2^M different operators each generating different equivalent solutions for each single solution.

Provided that there are no redundant hidden neurons in the solution, the number of solutions generated by the two of them together for a single solution is $2^M M!$. Having a certain number of 'true solutions' L, which are not simply generated from the same solution by applying the various operators, one obtains immediately $L2^M M!$ different solutions. These additional solutions significantly complicate the structure of the order parameters used in the replica method to estimate capacities, generalization capabilities, etc. For example, one can show that for a single 'true solution' with one original order parameter calculating the overlap of each solution pair, one obtains 7 and 34 different order parameters for the two and three hidden unit cases, respectively, just by applying the real weight-space symmetries *when there are no other restrictions imposed upon the solutions*; the number of these spurious order parameters grows exponentially with the increase in the number of hidden units.

On top of these symmetry groups there are at least two other equivalence groups, but these groups do not seem to affect the order parameters in the thermodynamic limit. The first of these equivalence groups is the 'tilting' group, resulting from the discreteness of the input and internal representations spaces, allowing the $(N-1)$-dimensional hyperplanes, related to each one of the weight vectors between the input and hidden nodes, to tilt in $N-1$ independent directions between discrete vectors representing the input vectors. A similar equivalence group exists for the $(M-1)$-dimensional hyperplane related to the hidden to output weight vector. The $(N-1)$- or $(M-1)$-dimensional cone of solution vectors generated by the possible tilting shrinks as $\mathcal{O}(1/N)$ (or $\mathcal{O}(1/M)$) and is therefore

not significant in the thermodynamic limit†. The second equivalence group is the group of permutations of *the internal representations*‡; fixing a certain solution vector between the hidden and output layers one can permute the group of internal representations related to the same output among themselves without affecting the actual output, forcing *different* input to hidden relations (and weights) each time. However, this equivalence group keeps the *spatial relation* between the various input to hidden weight vectors (actually the order parameters) constant and therefore does not have a substantial effect on the calculation.

The methods described in this paper suggest that deliberate external symmetry breaking should be performed prior to the calculation itself in order to eliminate this redundancy of solutions and to alleviate the replica-symmetry-breaking problem. Two methods are considered: one, based on restricting relations between the weight elements, is designed only to break the explicit symmetries described above, while the other, based on restricting the order parameters themselves, restores replica symmetry in the capacity limit. For the first method, it can be easily proven that in small systems, with a small number of hidden units, this ordering actually *restores* replica symmetry; however, there is no general proof yet that this is the case for any number of hidden neurons. The second method, for *restoring* replica symmetry is based on forcing additional restrictions on the configuration space, which require replica symmetry themselves. This method, which will be obvious when we reach the order-parameter definition stage, is effectively a microcanonical method which assigns certain order parameters to a fixed value (approaching 1 in the capacity limit)§.

The two methods are demonstrated in this paper via the two-layer perceptron-capacity calculation. However, they are applicable for a wide range of capacity and learning-analysis calculations.

Former attempts to calculate the capacity of multi-layer nets via statistical mechanics methods [7, 8] simplified the net by considering committee and parity machines with non-overlapping receptive fields for the hidden neurons. This avoids part of the computational difficulties resulting from the possible correlation between hidden neurons' representations. In this paper, the problem is tackled by examining *all* of the possible internal representations, in a search for the extreme case in which there is a single set of internal representations and weight matrices (under the restrictions discussed above) that performs the task. An approximation for carrying out the multi-dimensional Gaussian integrals appearing along the calculation is suggested, and special cases which have an exact analytical solution are examined.

Due to the complexity of the expressions we use numerical optimization methods for obtaining the order parameters and the capacity limit. The numerical results are examined in comparison with theoretical and simulation results obtained for similar configurations elsewhere.

In the next section (section 2) we explain the main ideas behind the two methods aimed at alleviating the replica-symmetry-breaking problem and then carry out the calculation itself in section 3. Some simple cases as well as the general case are examined explicitly

† One should note that for nets with a small number of hidden units this *is* a problem since there is no uniqueness for the hidden-to-output vector due to the fact that $O(1/M)$ might not be small enough. One can overcome this problem by enforcing additional restrictions on the solution weights like larger margins from the hyperplane (the parameter κ introduced in (2)) etc.

‡ Internal representations are defined as the representations of input vectors in the hidden layer(s) for a given net characterized by a certain set of weights and activation functions.

§ It is important to note that forcing the system to possess replica symmetry might result in an oversimplification of the configuration space and uninteresting results. In the present calculation we show that forcing replica symmetry produces similar end results to eliminating *redundant* symmetries; we therefore do not expect such an oversimplification in the current calculation.

in section 4; these are solved numerically and compared to simulation results for several small nets in section 5.

2. Breaking explicit symmetries versus restoring replica symmetry

The two methods introduced in this paper are significantly different, though their final goal is similar, i.e. to alleviate the replica-symmetry-breaking problem.

The first method is aimed at the redundant explicit symmetries of the solution space. The goal is to break the symmetries by adding a restriction on the configuration space, admitting only a single exemplar from the entire permutation–reflection group.

The easiest method of breaking the explicit symmetries is by forcing the system's weight vector u to have *positive* values ordered according to their values, i.e. $u_i > u_{i+1}$. This ordering leaves no residual symmetry in real weight-space; it keeps a single exemplar from each and every set generated by the permutation–reflection group. Moreover, it was recently proven [5, 6] that for the case of common continuous activation functions, eliminating the permutation–reflection equivalence group implies that the solution is unique and thus that the symmetry of the system's order parameter is restored. As we mentioned earlier, in the case of binary-activation functions, this method restores replica symmetry for small systems but there is no proof that this is the case in general. Using this method one can carry out the calculation with no residual symmetries left.

The second method, designed to *restore* replica symmetry is based on restricting the order-parameter space (and the configuration space as well) in a manner that requires replica symmetry. A necessary condition for replica symmetry is that the configuration space for the solution parameters is connected. Considering a certain *continuous* normalized solution-weight vector in configuration space one can always find a connected region of solution parameters confined within a multi-dimensional sphere with a radius ρ in the vicinity of the original solution. Restricting the various (unit) solutions to have an overlap greater than $1-2\rho^2$ forces *all* solutions (represented in the various replicas) to be confined to the same connected region in the vicinity of the end solution vector.

While for sub-optimal capacities, one can define a minimal overlap between different replicas, sufficient for keeping the configuration space connected, in the capacity limit this restriction results in a *unique* solution, i.e. an overlap of 1 between the same weight vectors of different replicas.

It is interesting to note that recently Meir and Fontanari [10, 11], calculating the capacity of a perceptron with discrete weights, showed that by using the micro-canonical distribution and the replica-symmetry ansatz they obtain similar results to those obtained by replica-symmetry-breaking methods. Using the micro-canonical distribution forces the energy of the solution to be in a certain value and thus filters out all other solutions with different energies. Since for systems with discrete weights, examined in their paper, each different combination of weights produces a different energy, this restriction actually chooses a *certain* solution, restoring replica symmetry. In their work Meir and Fontanari [10, 11] actually show that in breaking the system's symmetry by restricting the configuration space one can retrieve replica symmetry and obtain similar results to those achieved by considering the entire configuration space and using the replica-symmetry-breaking ansatz.

3. Computing the capacity

We start by computing the capacity of the two-layer perceptron using the first method, designed simply to break explicit symmetries. Later on we explain the second method,

designed to restore replica symmetry explicitly, noting the variations obtained in the expressions.

In order to compute the capacity of the two-layer perceptron with binary activation functions we use Gardner's method [1], based on measuring the normalized weight-space volume enabling the mapping of p input/output relations of vectors:

$$V = \frac{\int \left(\prod_{ij} \mathrm{d}w_{ij}\right)\left(\prod_i \mathrm{d}u_i\right)\ \sum_{\forall \epsilon_i^\mu} \prod_{\mu i}\left[\Theta_{\epsilon_i^\mu,\xi_j^\mu}\Theta_{\zeta^\mu,\epsilon_i^\mu}\right] \prod_{i=1}^{M} \Theta_{u_i}\,\delta_w\delta_u}{\int \left(\prod_{ij} \mathrm{d}w_{ij}\right)\left(\prod_i \mathrm{d}u_i\right)\delta_w\delta_u} \tag{1}$$

defining

$$\begin{aligned}
\Theta_{\epsilon_i^\mu,\xi_j^\mu} &\equiv \Theta\left(\epsilon_i^\mu \frac{1}{\sqrt{MN}} \sum_{j=1}^{N} w_{ij}\xi_j^\mu - \kappa\right) \\
\Theta_{\zeta^\mu,\epsilon_i^\mu} &\equiv \Theta\left(\zeta^\mu \frac{1}{\sqrt{MN}} \sum_{i=1}^{M} u_i\epsilon_i^\mu - \kappa\right) \\
\Theta_{u_i} &\equiv \Theta\left(u_i - u_{i+1} - \varrho\right) .
\end{aligned} \tag{2}$$

The numerator in (1) represents the volume of the weight space that together with a certain set of internal representations enables the storage of p patterns under the spherical constraint and ordering restrictions, while the denominator represents the volume of the entire relevant weight space. We expect the relevant volume of the configuration space to shrink to a single point in the capacity limit. The first two step functions (represented by Θ), assure the required mapping of the input vectors $\boldsymbol{\xi}^\mu$ onto the set of internal representations $\boldsymbol{\epsilon}^\mu$ and of the internal representations onto the output values ζ^μ, where μ is the pattern index, with a certain margin κ from the separating hyperplane; the third step function keeps an interval ϱ between successive-ordered weight elements. The indices i and j are the hidden and input site indices, w and $\boldsymbol{u}$ are the sets of weights connecting the input to hidden nodes and the hidden to output nodes, respectively; N and M are the number of input and hidden neurons, respectively, and $1/\sqrt{MN}$ is used for normalizing the expressions. The vector component u_{M+1} is set to zero thus forcing the vector $\boldsymbol{u}$ to be positive.

The denominator, as well as the last terms in the numerator, results from the spherical constraint for the two sets of weights

$$\delta_w \equiv \prod_{i=1}^{M} \delta\left(\sum_{j=1}^{N} w_{ij}^2 - N\right) \qquad \delta_u \equiv \delta\left(\sum_{i=1}^{M} u_i^2 - M\right). \tag{3}$$

The numerator includes two new elements in comparison with the conventional expression. The first is a summation over the entire space of possible internal representations that might connect the two sets of weights w and $\boldsymbol{u}$, while the second is an ordering term, forcing the weight vector $\boldsymbol{u}$ to be positive and ordered according to the values of its components. Note that the summation over all internal representations selects only *one* set of internal representations for each choice of weight and input vectors so that double counting does not occur.

Since the statistically relevant quantity is the average over the pattern distribution of the logarithm of V (due to its relation to the free energy [1]), we introduce replicas of the set of weights and internal representations, w_{ij}^α, u_i^α and $\epsilon_i^{\mu\alpha}$, where α is the replica index, in averaging the value of V^n.

The principle of the calculation involves the introduction of integral representations which permit the discrete summation to be carried out. Normally the parameters of the resulting multi-dimensional integrals are evaluated by solving the saddle-point equations.

Here, due to the complexity of the expression, we will find the optimal parameters using a numerical optimization technique.

Replacing all of the Θ functions by their integral form

$$\Theta\left(\epsilon_i^{\mu\alpha}\frac{1}{\sqrt{MN}}\sum_{j=1}^{N}w_{ij}^{\alpha}\xi_j^{\mu}-\kappa\right)=\int_{\kappa}^{\infty}\frac{\mathrm{d}\lambda_i^{\mu\alpha}}{2\pi}\int \mathrm{d}x_i^{\mu\alpha}\,\mathrm{e}^{\mathrm{i}x_i^{\mu\alpha}\lambda_i^{\mu\alpha}}\mathrm{e}^{-\mathrm{i}x_i^{\mu\alpha}\epsilon_i^{\mu\alpha}(1/\sqrt{MN})\sum_j w_{ij}^{\alpha}\xi_j^{\mu}}$$

$$\Theta\left(\zeta^{\mu}\frac{1}{\sqrt{MN}}\sum_{i=1}^{M}u_i^{\alpha}\epsilon_i^{\mu\alpha}-\kappa\right)=\int_{\kappa}^{\infty}\frac{\mathrm{d}\phi^{\mu\alpha}}{2\pi}\int \mathrm{d}y^{\mu\alpha}\,\mathrm{e}^{\mathrm{i}y^{\mu\alpha}\phi^{\mu\alpha}}\mathrm{e}^{-\mathrm{i}y^{\mu\alpha}\zeta^{\mu}(1/\sqrt{MN})\sum_i u_i^{\alpha}\epsilon_i^{\mu\alpha}}$$

$$\Theta(u_i^{\alpha}-u_{i+1}^{\alpha}-\varrho)=\int_{\varrho}^{\infty}\frac{\mathrm{d}\nu_i^{\alpha}}{2\pi}\int \mathrm{d}z_i^{\alpha}\,\mathrm{e}^{\mathrm{i}z_i^{\alpha}\nu_i^{\alpha}}\mathrm{e}^{-\mathrm{i}z_i^{\alpha}(u_i^{\alpha}-u_{i+1}^{\alpha})} \tag{4}$$

where we introduce the integration variables $\lambda_i^{\mu\alpha}$, $x_i^{\mu\alpha}$, $\phi^{\mu\alpha}$, $y^{\mu\alpha}$, ν_i^{α} and z_i^{α}. Similarly, the spherical constraint integral representations introduce the order parameters ε^{α} and E_i^{α} for the weight vectors u^{α} and matrices w^{α}, respectively. Averaging over all input and output vectors ξ^{μ} and ζ^{μ} in the large-N limit, one obtains for the main term of the numerator

$$\sum_{\forall\epsilon_i^{\mu\alpha}}\prod_{\mu}\int\left(\prod_{j=1}^{N}\mathrm{D}r_j^{\mu}\right)\mathrm{D}s^{\mu}\prod_{\alpha,i=1\ldots M}\left[\exp\left\{\frac{\mathrm{i}}{\sqrt{MN}}\epsilon_i^{\mu\alpha}\left(\sum_{j=1}^{N}w_{ij}^{\alpha}x_i^{\mu\alpha}r_j^{\mu}+u_i^{\alpha}y^{\mu\alpha}s^{\mu}\right)\right\}\right] \tag{5}$$

using the convention $\mathrm{D}x\equiv(\mathrm{d}x/\sqrt{2\pi})\mathrm{e}^{-x^2/2}$.

Summing over all possible internal representations and applying the Gaussian trick again, the entire numerator term becomes (omitting the μ index and the weight integrations):

$$\begin{aligned}\prod_{\mu}\Bigg\{&\int_{\kappa}^{\infty}\left(\prod_{\alpha,i}\frac{\mathrm{d}\lambda_i^{\alpha}}{2\pi}\right)\int\left(\prod_{\alpha,i}\mathrm{d}x_i^{\alpha}\right)\left(\prod_{\alpha}\frac{\mathrm{d}\phi^{\alpha}}{2\pi}\right)\int\left(\prod_{\alpha}\mathrm{d}y^{\alpha}\right)\mathrm{e}^{\mathrm{i}\sum_{\alpha,i}x_i^{\alpha}\lambda_i^{\alpha}+\mathrm{i}\sum_{\alpha}y^{\alpha}\phi^{\alpha}}\\
&\times\int\left(\prod_{\alpha,i}\mathrm{D}t_i^{\alpha}\right)\left(\prod_{j=1}^{N}\mathrm{D}r_j\right)\mathrm{D}s\,\mathrm{e}^{-(\mathrm{i}/\sqrt{MN})\sum_{\alpha,i,j}w_{ij}^{\alpha}x_i^{\alpha}r_jt_i^{\alpha}-(\mathrm{i}/\sqrt{MN})\sum_{\alpha,i}u_i^{\alpha}y^{\alpha}st_i^{\alpha}}\Bigg\}\\
&\times\int\left(\prod_{i,\alpha}\frac{\mathrm{d}E_i^{\alpha}}{4\pi\mathrm{i}}\right)\mathrm{e}^{(N/2)\sum_{\alpha,i}E_i^{\alpha}-\sum_{\alpha,i}(E_i^{\alpha}/2)\sum_j(w_{ij}^{\alpha})^2}\\
&\times\int\left(\prod_{\alpha}\frac{\mathrm{d}\varepsilon^{\alpha}}{4\pi\mathrm{i}}\right)\mathrm{e}^{(M/2)\sum_{\alpha}\varepsilon^{\alpha}-\sum_{\alpha}(\varepsilon^{\alpha}/2)\sum_i(u_i^{\alpha})^2}\\
&\times\int_{\varrho}^{\infty}\left(\prod_{\alpha,i=1\ldots M}\frac{\mathrm{d}\nu_i^{\alpha}}{2\pi}\right)\int\left(\prod_{\alpha,i=1\ldots M}\mathrm{d}z_i^{\alpha}\right)\mathrm{e}^{\mathrm{i}\sum_{\alpha,i}z_i^{\alpha}\nu_i^{\alpha}}\mathrm{e}^{-\mathrm{i}\sum_{\alpha,i}z_i^{\alpha}(u_i^{\alpha}-u_{i+1}^{\alpha})}.\end{aligned} \tag{6}$$

The entire numerator term consists of two main expressions: the first, in the curly brackets, depends on the patterns (μ-dependent) representing restrictions imposed by the input/output vectors themselves; the second represents general restrictions imposed on the solution vectors/matrices such as spherical constraint and weight ordering.

Carrying out the s and r_j integrations one can rewrite the main term (μ-dependent) of (6) as

$$\begin{aligned}\int\left(\prod_{\alpha,i=1\ldots M}\mathrm{D}t_i^{\alpha}\right)\exp\Bigg\{&-\frac{1}{2}\sum_{\alpha,k,i}\widehat{P}_{ik}^{\alpha}t_i^{\alpha}t_k^{\alpha}\,(y^{\alpha})^2-\sum_{\alpha<\beta,i,k}P_{ik}^{\alpha\beta}t_i^{\alpha}t_k^{\beta}y^{\alpha}y^{\beta}-\frac{1}{2M}\sum_{\alpha,i}\left(t_i^{\alpha}x_i^{\alpha}\right)^2\\
&-\frac{1}{M}\sum_{\alpha<\beta,i}t_i^{\alpha}t_i^{\beta}x_i^{\alpha}x_i^{\beta}\widetilde{Q}_i^{\alpha\beta}-\frac{1}{M}\sum_{\alpha,i<k}t_i^{\alpha}t_k^{\alpha}x_i^{\alpha}x_k^{\alpha}\widehat{Q}_{ik}^{\alpha}-\frac{2}{M}\sum_{\alpha<\beta,i<k}t_i^{\alpha}t_k^{\beta}x_i^{\alpha}x_k^{\beta}Q_{ik}^{\alpha\beta}\Bigg\}\end{aligned} \tag{7}$$

where the order parameters are defined by

$$\widehat{P}_{ik}^{\alpha}=\frac{1}{NM}u_i^{\alpha}u_k^{\alpha} \qquad P_{ik}^{\alpha\beta}=\frac{1}{NM}u_i^{\alpha}u_k^{\beta} \qquad \alpha\neq\beta$$

$$\widehat{Q}_{ik}^{\alpha}=\frac{1}{N}\sum_{j=1}^{N}w_{ij}^{\alpha}w_{kj}^{\alpha} \qquad \widetilde{Q}_{k}^{\alpha\beta}=\frac{1}{N}\sum_{j=1}^{N}w_{kj}^{\alpha}w_{kj}^{\beta} \qquad Q_{ik}^{\alpha\beta}=\frac{1}{N}\sum_{j=1}^{N}w_{ij}^{\alpha}w_{kj}^{\beta} \qquad \alpha\neq\beta\,.$$

The incentive for introducing these order parameters is that they represent statistical properties of the configuration space, some of which are of order 1 (like the overlap of two weight vectors of the same replica and hidden node—$\widetilde{Q}_k^{\alpha\alpha}$) while others are significantly smaller, representing weaker correlations between weight vectors related to different nodes and different replicas.

These order-parameter definitions are expressed explicitly by introducing δ functions; these functions take an integral form, defining a set of coupled integration variables: $F_{ik}^{\alpha\beta}$, $\widehat{F}_{ik}^{\alpha}$, $\widetilde{F}_{i}^{\alpha\beta}$, $D_{ik}^{\alpha\beta}$, $\widehat{D}_{ik}^{\alpha}$ for the following order parameters $Q_{ik}^{\alpha\beta}$, $\widehat{Q}_{ik}^{\alpha}$, $\widetilde{Q}_{i}^{\alpha\beta}$, $P_{ik}^{\alpha\beta}$ and $\widehat{P}_{ik}^{\alpha}$, respectively.

At this point, we can actually introduce the second method, designed to *restore* replica symmetry. As a condition for replica symmetry we require, in the capacity limit, an overlap of 1 between the same weight vectors of different replicas. Forcing $\widetilde{Q}_i^{\alpha\beta}=\gamma$, where $\gamma\to 1$, requires effectively replica symmetry. Practically we should simply replace the ordering term $\prod_{i=1}^{M}\Theta_{u_i}$ in the numerator of (1) by a term that will take the form $\delta(\gamma-(1/N)\sum_{j=1}^{N}w_{kj}^{\alpha}w_{kj}^{\beta})$, where $\gamma\to 1$ at the present stage. This results in a similar expression to the one obtained using the first method, excluding the ordering term (the last term of (6)).

We shall carry on the calculation using the first method of breaking explicit symmetries, indicating the differences in the expressions obtained by using the second method (basically, omitting ordering terms and replacing $\widetilde{Q}_i^{\alpha\beta}$ by γ, where $\gamma\to 1$).

Applying the replica-symmetry ansatz [1] we can accumulate the various terms: terms which include the weight matrix elements w_{ij}^{α}, those including the weight vector elements u_k^{α}, the free terms which include the order parameter and the main term including all terms with the vector index μ. At this stage we set the margin parameter κ to zero for carrying out the basic capacity calculation (this parameter is anyway only 'nice to have' for imposing *additional* restrictions on the configuration space).

Integrating the main term (7) and the ordering term using the Gaussian trick and rewriting the entire expression as e^{nMNG}, where $n\to 0$ is the number of replicas, one obtains the following expression for G (omitting constants and setting κ to zero):

$$\begin{aligned}G=&-\frac{1}{2M}\log|\mathcal{E}-\mathcal{F}|-\frac{1}{2M}\mathrm{tr}\left[(\mathcal{E}-\mathcal{F})^{-1}\mathcal{F}\right]\\&+\frac{1}{MN}\int\left(\prod_k\frac{\mathrm{d}v_k}{\sqrt{2\pi}}\right)\frac{\mathrm{e}^{-\frac{1}{2}v^{\mathrm{T}}\mathcal{S}^{-1}v}}{|\mathcal{S}|^{\frac{1}{2}}}\log\left[\int_{Q}^{\infty}\left(\prod_k\frac{\mathrm{d}v_k}{\sqrt{2\pi}}\right)\mathrm{e}^{-\frac{1}{2}\eta^{\mathrm{T}}(\mathcal{R}-\mathcal{S})\eta+i\eta\cdot v}\right]\\&+\alpha_{\mathrm{c}}\int\left(\prod_k\frac{\mathrm{d}s_k}{\sqrt{2\pi}}\right)\frac{\mathrm{e}^{-\frac{1}{2}s^{\mathrm{T}}\mathcal{B}^{-1}s}}{|\mathcal{B}|^{\frac{1}{2}}}\log\left[\int_{s_k}^{\infty}\left(\prod_k\frac{\mathrm{d}\lambda_k}{\sqrt{2\pi}}\right)\frac{\mathrm{e}^{-\frac{1}{2}\lambda^{\mathrm{T}}(\mathcal{A}-\mathcal{B})^{-1}\lambda}}{|\mathcal{A}-\mathcal{B}|^{\frac{1}{2}}}\right]\\&+\alpha_{\mathrm{c}}\int\left(\prod_k\frac{\mathrm{d}r_k}{\sqrt{2\pi}}\right)\frac{\mathrm{e}^{-\frac{1}{2}r^{\mathrm{T}}\mathcal{D}^{-1}r}}{|\mathcal{D}|^{\frac{1}{2}}}\log\left[\frac{1}{2}\int\left(\prod_k\mathrm{D}l_k\right)\mathrm{erfc}\left(\frac{l\cdot r}{\sqrt{2l^{\mathrm{T}}(\mathcal{C}-\mathcal{D})l}}\right)\right]\end{aligned}$$

$$+\frac{1}{2M}\sum_{i<k} F_{ik}Q_{ik} - \frac{1}{2M}\sum_{i\neq k}\widehat{F}_{ik}\widehat{Q}_{ik} + \frac{1}{2M}\sum_{k}\widetilde{F}_k\widetilde{Q}_k$$

$$+\frac{1}{2}\sum_{ik} D_{ik}P_{ik} - \sum_{ik}\widehat{D}_{ik}\widehat{P}_{ik} + \frac{1}{2N}\varepsilon + \frac{1}{2M}\sum_{k} E_k . \tag{8}$$

In this equation α_c represents the number of stored patterns normalized with respect to MN, the matrices $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$, $\mathcal{R}$ and $\mathcal{S}$ and the vector η are

$$\eta_k \equiv \sum_{j=k}^{M} v_j \qquad \mathcal{A}_{jk} \equiv \frac{1}{M}\left[\delta_{jk} + (1-\delta_{jk})\widehat{Q}_{jk}\right] \qquad \mathcal{B}_{jk} \equiv \frac{1}{M}\left[\delta_{jk}\widetilde{Q}_k + (1-\delta_{jk})Q_{jk}\right]$$

$$\mathcal{D}_{jk} \equiv P_{jk} \qquad \mathcal{E}_{jk} \equiv \delta_{jk}E_k - (1-\delta_{jk})\widehat{F}_{jk} \qquad \mathcal{F}_{jk} \equiv -\delta_{jk}\widetilde{F}_k - (1-\delta_{jk})\frac{F_{jk}}{2}$$

$$\mathcal{S}_{jk} \equiv -D_{jk} \qquad \mathcal{R}_{jk} \equiv \delta_{jk}\varepsilon - 2\widehat{D}_{jk} \qquad \mathcal{C}_{jk} \equiv \widehat{P}_{jk} .$$

We assume that the matrices are positive-definite since they cause a divergence of the term G otherwise.

Equation (8) is the cornerstone of the calculation and represents an extention of Gardner's expression [1] to the multi-dimensional case. Using the second method, designed to restore replica symmetry, results in a similar expression, excluding the following ordering term integrals:

$$\frac{1}{MN}\int\left(\prod_k \frac{\mathrm{d}v_k}{\sqrt{2\pi}}\right)\frac{\mathrm{e}^{-\frac{1}{2}v^{\mathrm{T}}\mathcal{S}^{-1}v}}{|\mathcal{S}|^{\frac{1}{2}}}\log\left[\int_{\varrho}^{\infty}\left(\prod_k \frac{\mathrm{d}\nu_k}{\sqrt{2\pi}}\right)\mathrm{e}^{-\frac{1}{2}\eta^{\mathrm{T}}(\mathcal{R}-\mathcal{S})\eta + \mathrm{i}\eta\cdot v}\right]$$

and replacing $\widetilde{Q}_k$ by γ.

4. Simple configurations and the general case

Examining expression (8) for the single hidden neuron case it is easy to see the similarity between (8) and Gardner's expression; most of the expression keeps its form, with the multi-dimensional integrals reducing to simple integrals and the matrix expressions to corresponding scalars. In this case there is no ordering and the hidden to output weight is always 1, so the entire expression collapses back to Gardner's expression giving the same results:

$$G = -\frac{1}{2}\log(E+\widetilde{F}) + \frac{1}{2}\frac{\widetilde{F}}{E+\widetilde{F}} + \frac{1}{2}\widetilde{F}\widetilde{Q} + \frac{1}{2}E + \alpha_c\int \mathrm{D}s\log\left[\int_{s\sqrt{\widetilde{Q}}/\sqrt{1-\widetilde{Q}}}^{\infty}\mathrm{D}\lambda\right]. \tag{9}$$

The two hidden-neuron case looks at first sight identical to the previous case since when we order the weights we create one *dominant* weight. However, if we allow equality of weights ($\varrho = 0$) and define an activation of zero to result in a positive output (we could equally define it as negative), we actually have a larger capacity which can be computed using different methods as well (see appendix A). Although, in this case there is no ordering and the hidden-to-output weight vector is unique, there is no simple analytical result since two-dimensional Gaussian integrals with a general lower limit cannot be carried out. A slight simplification of (8) for the two-dimensional case can be introduced:

$$G = -\frac{1}{4}\log\begin{vmatrix} E_1+\widetilde{F}_1 & F/2-\widehat{F} \\ F/2-\widehat{F} & E_2+\widetilde{F}_2 \end{vmatrix}$$

$$+\frac{1}{4}\mathrm{tr}\left[\begin{pmatrix} E_1+\widetilde{F}_1 & F/2-\widehat{F} \\ F/2-\widehat{F} & E_2+\widetilde{F}_2 \end{pmatrix}^{-1}\begin{pmatrix} \widetilde{F}_1 & F/2 \\ F/2 & \widetilde{F}_2 \end{pmatrix}\right]$$

$$+\alpha_c \int_{-\infty}^{\infty} \frac{ds_1}{\sqrt{2\pi}} \frac{ds_2}{\sqrt{2\pi}} \frac{e^{-\frac{1}{2}s^T\mathcal{B}^{-1}s}}{|\mathcal{B}|^{\frac{1}{2}}} \log\left[\int \frac{d\lambda}{2\pi} \frac{e^{-\frac{1}{2}(\tilde{s}_1^2+\tilde{s}_2^2)/(1-\lambda^2)+(\lambda\tilde{s}_1\tilde{s}_2)/(1-\lambda^2)}}{\sqrt{1-\lambda^2}}\right]$$
$$+\tfrac{1}{4}FQ - \tfrac{1}{2}\widehat{F}\widehat{Q} + \tfrac{1}{4}\widetilde{F}_1\widetilde{Q}_1 + \tfrac{1}{4}\widetilde{F}_2\widetilde{Q}_2 + \tfrac{1}{4}E_1 + \tfrac{1}{4}E_2$$

where $\lambda \equiv [\widehat{Q} - Q]/(\sqrt{1-\widetilde{Q}_1}\sqrt{1-\widetilde{Q}_2})$, $\tilde{s}_k \equiv s_k/\sqrt{\widetilde{Q}_k}$, all vectors are two dimensional, and the λ integral is indefinite.

Returning to the general case of M hidden units, note that up until now no approximations were made. However, one cannot carry out multi-dimensional integrals from a general lower limit to infinity analytically. We therefore use Kendall's [12] approximation (see appendix B) to simplify the expression for G. In order to use the two most dominant terms of Kendall's approximations the following two assumptions were used.

(i) $[\mathcal{A}-\mathcal{B}]_{jk} \ll [\mathcal{A}-\mathcal{B}]_{kk}$ for $k \neq j$. This assumption can be explained easily by the different forms of the diagonal and non-diagonal terms in $\mathcal{A}-\mathcal{B}$. The diagonal terms are of the form $1-\widetilde{Q}_k$, where $\widetilde{Q}_k \leqslant 1$ and the difference is therefore always a positive small value, constantly decreasing as one approaches the capacity limit. The off-diagonal terms on the other hand are of the form $\widehat{Q}_{jk} - Q_{jk}$, where Q_{jk} can approach $\widehat{Q}_{jk}$ either from above or from below since $\widehat{Q}_{jk}$ is not expected to have a particularly high value (obviously if it does the probabilities that Q_{jk} will reach $\widehat{Q}_{jk}$ from above and from below will be significantly different due to the fact that $Q_{jk} < 1$). Therefore we expect the off-diagonal terms to be always small, particularly in comparison with the diagonal terms.
(ii) $[\widetilde{\mathcal{R}-\mathcal{S}}]^{-1}_{jk} \ll [\widetilde{\mathcal{R}-\mathcal{S}}]^{-1}_{kk}$ for $k \neq j$, where $[\widetilde{\mathcal{R}-\mathcal{S}}]^{-1}$ is defined as

$$[\widetilde{\mathcal{R}-\mathcal{S}}]^{-1}_{jk} \equiv \frac{[\widehat{\mathcal{R}-\mathcal{S}}]^{-1}_{kj}}{\sqrt{[\widehat{\mathcal{R}-\mathcal{S}}]^{-1}_{kk}[\widehat{\mathcal{R}-\mathcal{S}}]^{-1}_{jj}}}$$
$$[\widehat{\mathcal{R}-\mathcal{S}}]_{jk} \equiv \sum_{l\leqslant k, m\leqslant j} [\mathcal{R}-\mathcal{S}]_{ml}. \tag{10}$$

This assumption results from the relations between the diagonal and off-diagonal elements in these matrices. Diagonal elements are of the form $\varepsilon + D_{kk} - 2\widehat{D}_{kk}$, while off-diagonal elements are of the form $D_{jk} - 2\widehat{D}_{jk}$ and in the capacity limit $D_{jk} \to \widehat{D}_{jk}$ for all k and j. Examining these integration parameters closely one can show that, in that limit, $[\widetilde{\mathcal{R}-\mathcal{S}}]^{-1}_{jk} \ll [\widetilde{\mathcal{R}-\mathcal{S}}]^{-1}_{kk}$.

At this stage we can set the second margin parameter ϱ to zero for carrying out the basic capacity calculation (like κ, ϱ is only 'nice to have' for imposing *additional* restrictions on the configuration space). Setting ϱ to zero actually makes the weight ordering slightly different, allowing equality of adjacent weights; carrying out the calculation for $\varrho \neq 0$ is feasible (see appendix C) and produces minor modifications to expression (8). As explained earlier, the parameter ϱ can be useful, especially for small-system calculations, adding restrictions to the configuration space and forcing it to converge to a single point.

Carrying out the approximations to the multi-dimensional Gaussian integrations (see appendix C), expression (8) can then be reduced to

$$G = \frac{1}{2M}\log|\mathcal{A}-\mathcal{B}| + \frac{1}{2M}\operatorname{tr}[(\mathcal{A}-\mathcal{B})^{-1}\mathcal{B}]$$
$$-\frac{1}{2MN}\log|\mathcal{R}-\mathcal{S}| - \frac{1}{2MN}\operatorname{tr}[(\mathcal{R}-\mathcal{S})^{-1}\mathcal{S}]$$

$$-\frac{1}{MN\pi}\mathrm{tr}(\psi\mathcal{S})+\left(\frac{2}{\pi}\right)^{M/2}\frac{1}{MN}\sum_{k,j\neq k}\left[\widetilde{\mathcal{R}-\mathcal{S}}\right]_{jk}^{-1}|I-\mathcal{S}\psi|^{-\frac{1}{2}}$$

$$+\frac{1}{2}\sum_{ik}D_{ik}P_{ik}-\sum_{ik}\widehat{D}_{ik}\widehat{P}_{ik}+\frac{1}{2N}\varepsilon+\frac{1}{2}\alpha_c\int_0^\infty d\sigma\frac{M-\mathrm{tr}\left[I+\sigma(\mathcal{C}-\mathcal{D})\right]^{-1}}{\sigma\,|I-\sigma(\mathcal{C}-\mathcal{D})|^{\frac{1}{2}}}$$

$$+\alpha_c\Bigg\{-\frac{1}{4}\sum_k\frac{\mathcal{B}_{kk}}{(\mathcal{A}-\mathcal{B})_{kk}}+\frac{1}{4}\sum_k\log\frac{(\mathcal{A}-\mathcal{B})_{kk}}{\mathcal{B}_{kk}}$$

$$+\left[\log\sum_{k,j\neq k}\widetilde{(\mathcal{A}-\mathcal{B})}_{jk}-\frac{1}{2}\sum_k\log(\mathcal{A}-\mathcal{B})_{kk}\right]\int_0^\infty\left(\prod_k\frac{dr_k}{\sqrt{2\pi}}\right)\frac{e^{-\frac{1}{2}r^T\mathcal{B}^{-1}r}}{|\mathcal{B}|^{\frac{1}{2}}}\Bigg\} \tag{11}$$

where I is the identity matrix and

$$\left[\widetilde{\mathcal{A}-\mathcal{B}}\right]_{jk}\equiv\frac{[\mathcal{A}-\mathcal{B}]_{jk}}{\sqrt{[\mathcal{A}-\mathcal{B}]_{kk}[\mathcal{A}-\mathcal{B}]_{jj}}}$$

$$\psi\equiv[\mathcal{T}^T(\mathcal{R}-\mathcal{S})]^{-1}[\widetilde{\mathcal{R}-\mathcal{S}}][(\mathcal{R}-\mathcal{S})\mathcal{T}]^{-1}$$

$$\left[\widetilde{\mathcal{R}-\mathcal{S}}\right]_{jk}\equiv\left[\left(\widetilde{\mathcal{R}-\mathcal{S}}\right)_{jj}^{-1}\left(\widetilde{\mathcal{R}-\mathcal{S}}\right)_{kk}^{-1}\right]^{-\frac{1}{2}}\qquad\mathcal{T}_{jk}\equiv\begin{cases}1 & j\geqslant k\\ 0 & \text{otherwise}.\end{cases}$$

The corresponding expression, using the second method for restoring replica symmetry is similar, excluding the $\mathcal{R}$, $\mathcal{S}$, $\mathcal{T}$ and ψ terms.

Expression (11) can now be optimized since there are no more multi-dimensional integrations (the Gaussian integration over r can be easily approximated using Kendall's expression, and in any case it doesn't play a significant role in the optimization process). Moreover, *since we ordered the weights* we can assume that, in the capacity limit, all of the matrix elements $\mathcal{D}_{jk}\to\mathcal{C}_{jk}$. These matrix elements are products of the various components in the weight vector u which converges now to a *single point* in configuration space. Note that this limit can be taken *only* if there are no redundant neurons and if the weights are ordered since it requires a convergence of the parameters to a single point in configuration space. For example, in the two-hidden-neuron case this forces the outgoing weights of the hidden neurons *to be equal* since one hidden neuron is otherwise dominant, making the other neuron redundant.

Using this limit results in a much simpler expression for (11), having only the $\mathcal{A}$ and $\mathcal{B}$ terms and no ordering terms. The second method for restoring replica symmetry yields a similar end result, using a similar assumption for the convergence of the matrix elements $\mathcal{D}_{jk}\to\mathcal{C}_{jk}$. The remaining expression is

$$G=\frac{1}{2M}\log|\mathcal{A}-\mathcal{B}|+\frac{1}{2M}\mathrm{tr}\left[(\mathcal{A}-\mathcal{B})^{-1}\mathcal{B}\right]$$

$$+\alpha_c\Bigg\{-\frac{1}{4}\sum_k\frac{\mathcal{B}_{kk}}{(\mathcal{A}-\mathcal{B})_{kk}}+\frac{1}{4}\sum_k\log\frac{(\mathcal{A}-\mathcal{B})_{kk}}{\mathcal{B}_{kk}}$$

$$+\left[\log\sum_{k,j\neq k}\widetilde{(\mathcal{A}-\mathcal{B})}_{jk}-\frac{1}{2}\sum_k\log(\mathcal{A}-\mathcal{B})_{kk}\right]\int_0^\infty\left(\prod_k\frac{dr_k}{\sqrt{2\pi}}\right)\frac{e^{-\frac{1}{2}r^T\mathcal{B}^{-1}r}}{|\mathcal{B}|^{\frac{1}{2}}}\Bigg\}. \tag{12}$$

Obviously, for the case of a single hidden neuron, expression (12) coincides with the related approximation to Gardner's expression [1] giving similar results for the net capacity and the order parameters. However, for the general case it is difficult to extremize the expression analytically and one should use numerical methods.

5. Numerical solutions and simulations

Due to the complexity of expression (12) we did not try to optimize it analytically using the saddle-point equations, though for some of the order parameters this can be done (we actually use the saddle-point equations for E_i, F_{ik}, $\widehat{F}_{ik}$ and $\widetilde{F}_i$), simplifying the expression significantly. Also, optimizing expression (11) numerically is not particularly simple since the sensitivity of the expression to the various parameters varies significantly from one parameter to another.

We optimized these terms using two methods: the 'downhill simplex' method and the adaptive simulated-annealing algorithm [13] (which is a generalization of the conventional simulated-annealing algorithm [14]). We used these algorithms for optimizing expression (12) for various cases, obtaining the dependence of the capacity (α_c, the number of stored patterns divided by MN) on the number of hidden units for the cases $M = 1, \ldots, 6$, shown in figure 1. The results coincide for a single hidden neuron with Gardner's result and for the two-hidden-neuron case with theoretical results achieved using a different method. This method based on the capacity of two groups of correlated patterns is explained explicitly in appendix A.

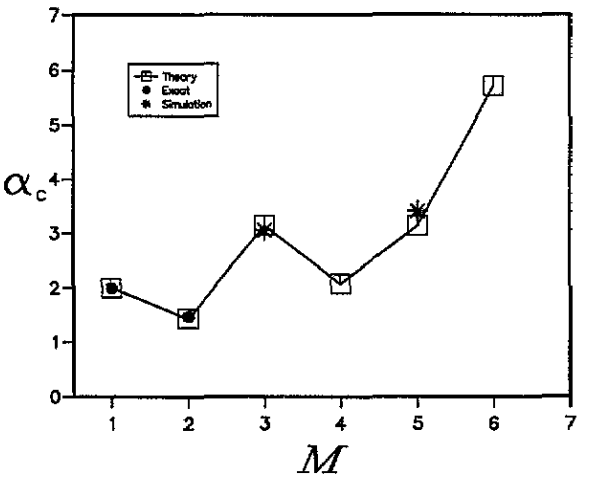

Figure 1. The dependence of the capacity (α_c) of a two-layer feed-forward binary neural net with a single output neuron on the number of hidden neurons. The full line (and boxes) represents theoretical results, the full circles represent exact theoretical results for the one- and two-neuron cases derived in [1] and appendix A, respectively, and the asterisks represent simulation results for the three- and five-neuron cases presented in [15].

We compared these results for the three- and five-hidden-neuron cases with those presented recently for the two-layer perceptron (and the fully connected committee machine) [15]. The two sets of results are in good agreement. The capacity shows a moderate increase for larger number of hidden neurons resulting from the increase in the combinations of internal representations†. One can relate the lower relative capacity of small nets with an even number of hidden neurons, in comparison to those with an odd number, to the reduced number of different combinations of internal representations. We expect this gap to narrow as the number of hidden neurons increases. Though it is hard to draw general conclusions from the behaviour of the capacity results for several small systems, the analytic results suggest that there is a moderate increase in the capacity α_c with the increase in the number of hidden neurons M. As the number of hidden neurons increases we expect the capacity to rise increasingly rapidly.

† One should note that in spite of the close relation between the capacity as defined by statistical mechanics and the VC dimension [16], the result obtained is not the VC dimension of the multilayer net. The statistical mechanics capacity represents the capacity of the *average* case while the VC dimension represents the *worst* case.

Though there is no difficulty in principle in optimizing expressions (11) and (12) for *any number of hidden units (there are no multi-dimensional integrals after using Kendall's* approximation), there are practical difficulties resulting from the large number of parameters. Moreover, it is unlikely that the mutual dependence of the parameters in the expression could be disentangled since this will break the multilayer net into separated perceptrons. However, one can hope that some more general approximations or bounds for the relations between the various parameters can emerge, resulting in general results for the large-M case.

6. Concluding remarks

In this paper we examine the capacity calculation of a binary two-layer feed-forward neural net with a single output neuron. We offer a method for carrying out the calculation based on summing over all of the internal representations and breaking explicit symmetries existing in such nets. Two methods, one for alleviating and one restoring replica symmetry, are introduced. The first relates the various weight components in a way that breaks the most significant explicit symmetries, while the second relies on forcing restrictions on the order parameters enforcing replica symmetry.

In addition, we offer approximations for avoiding some of the practical obstacles appearing along the calculation, such as the multi-dimensional Gaussian integrals with finite limits. We obtain numerical results by optimizing the expressions with respect to the order parameters using numerical optimization algorithms (downhill simplex, adaptive simulated annealing).

The methods suggested in this paper for alleviating the replica-symmetry-breaking problem by restricting the configuration space are applicable to a variety of calculations dealing with multilayer nets both of continuous and discrete representations. Applying these methods to other capacity problems is rather simple, though in the case of continuous activation functions (where replica symmetry is *restored* by breaking explicit symmetries) one can expect some extremely complicated calculations due to the continuous activation functions used. In the case of discrete weights, breaking explicit symmetries is obviously *insufficient* since additional symmetry breaking occurs due to the discreteness of the configuration space and microcanonical techniques might be more appropriate.

Breaking explicit symmetries in multilayer training and generalization calculations alleviates the replica-symmetry-breaking problem, which occurs in the course of training when node symmetry is broken; however, it is not clear whether it actually restores replica symmetry in this case, which probably depends on the way node symmetry is broken, and *additional/different symmetry breaking might be required for restoring replica symmetry.* Obviously, the permutation–reflection group generates replicas of each one of the solutions, as in the capacity-limit case (some of which are identical due to node symmetry subgroups), however, it is not clear whether different original solutions represent a connected region in configuration space. In the case of learnable training problems, where a *certain* end solution is defined, one can show that apart from the permutation–reflection symmetry all solution hyperspaces, in any stage of the training procedure, are connected since they should all include the end solution. Hence, *breaking explicit symmetries does restore replica* symmetry in this case.

The basic concept, presented in this paper, of breaking explicit symmetries in the configuration space offers a method for simplifying complicated neural network calculations; however, the identification of the symmetries of the problem and a clever choice of methods for breaking them varies from one problem to another and needs to be worked out differently for each problem.

Acknowledgments

The author would like to thank D J Wallace for stimulating discussions, encouragement and support as well as for valuable comments regarding the manuscript, and J Simone, C Morningstar and D Barber for helpful suggestions. The work is supported by SERC grant No. GRF 79719.

Appendix A. Capacity of binary two-layer nets with two hidden neurons

Defining an activation of zero to produce a + output and the two weight components from the hidden layer to the output to be equal, one can separate the internal representations in the two-hidden-neuron case into two groups. For example, the groups $\{(++), (-+), (+-)\}$ and $\{(--)\}$ represent the response of the two hidden neurons to the various input vectors. The two groups of internal representations are mapped to a + and − output, respectively (we can choose any other grouping equally by changing the sign of the hidden-to-output vector components or defining the zero activation response differently). For convenience we define two groups of vectors $\xi^- \in P$ and $\xi^+ \in N$, mapped to posivite and negative outputs, respectively (P and N represent the entire set of stored vectors, mapped to a positive and negative output, respectively).

Since we assume random input–output relations, half of the input vectors are mapped to the first group of internal representations while the other half is mapped to $(--)$. Obviously, this requires input vectors represented by an internal representation vector $(--)$ to be assigned to − *in both of the hidden layer neurons*, while vectors from the first group represented by other internal representations must be assigned to + *for at least one of the hidden neurons*. This division creates a situation in which the image of the input vectors in the hidden units representation is the following.

- *First hidden neuron.* The group of vectors N is mapped to a negative hidden layer output while part of the vectors in P, $P_1 \subset P$ is mapped to a positive output.
- *Second hidden neuron.* The group of vectors N is mapped to a negative output (at the hidden layer) while the rest of the vectors in P, $P_2 \subset P$, is mapped to a positive output.

So, at the end, stored vectors assigned to a negative *output* are mapped to a $(--)$ internal representation while those assigned to a positive *output* are mapped to one of the other internal representations. Clearly $P_1 \cup P_2 = P$, representing *all* of the stored vectors with a positive output, and the number of patterns in P equals the number of patterns in N (random distribution). This division creates a storage of correlated patterns in the two input-to-hidden perceptrons, which we can estimate explicitly [1, 17] as $\alpha_c(\beta)$ for a certain ratio β between positively and negatively assigned vectors for each one of these two perceptrons. Using the capacity expression $\alpha_c(\beta)$, the overall capacity $\alpha_c^{\{2,2\}}$ (two layer, two hidden neurons) is therefore:

$$\alpha_c^{\{2,2\}} = \alpha_c(\beta_1) + \alpha_c(\beta_2)(1 - \beta_2) \tag{A1}$$

where $\beta_k = N/(N + P_k)$, adding the number of vectors with a negative output stored in both perceptrons $\beta_1\alpha_c(\beta_1)$ to the two groups of vectors with positive outputs stored in the two perceptrons separately, $\alpha_c(\beta_1)(1 - \beta_1)$ and $\alpha_c(\beta_2)(1 - \beta_2)$. Examining the capacity $\alpha_c^{\{2,2\}}$ under the constraint $P_1 \cup P_2 = P$ (which includes the same number of vectors as N) one can calculate the extremum for $\alpha_c^{\{2,2\}}$. This maximum appears where $\beta_1 = \beta_2 = \frac{2}{3}$ and results in $\alpha_c^{\{2,2\}} = 1.437$.

Appendix B. Kendall's approximation

Kendall approximated multi-dimensional Gaussian integrals with general lower limits using a binomial distribution; for simplicity we introduce the approximation for the three-dimensional case though it can be easily generalized to any number of integrations (and variables):

$$\int_{s_k}^{\infty}\left(\prod_{k=1}^{3}\frac{\mathrm{d}\lambda_k}{\sqrt{2\pi}}\right)\frac{\mathrm{e}^{-\frac{1}{2}\lambda^{\mathrm{T}}\mathcal{A}^{-1}\lambda}}{|\mathcal{A}|^{\frac{1}{2}}}$$
$$=\sum\frac{\mathcal{A}_{12}^{j}\mathcal{A}_{23}^{k}\mathcal{A}_{13}^{l}}{j!k!l!}H_{j+k-1}(s_1)f(s_1)H_{j+l-1}(s_2)f(s_2)H_{l+k-1}(s_3)f(s_3) \qquad \text{(B1)}$$

where the summation is over all possible indices j, k, and l; $f(s_k)\equiv \mathrm{e}^{-\frac{1}{2}s_k^2}/\sqrt{2\pi}$ and the matrix $\mathcal{A}$ is of the form

$$\mathcal{A}\equiv\begin{bmatrix} 1 & \mathcal{A}_{12} & \mathcal{A}_{13} & \dots & \mathcal{A}_{1M} \\ \mathcal{A}_{21} & 1 & \mathcal{A}_{23} & & \mathcal{A}_{2M} \\ \vdots & & \ddots & & \vdots \\ \mathcal{A}_{M1} & \dots & & & 1 \end{bmatrix}. \qquad \text{(B2)}$$

The functions $H_j(x)$ are Hermite polynomials defined as

$$H_j f(x)\equiv\left(-\frac{\mathrm{d}}{\mathrm{d}x}\right)^{j} f(x). \qquad \text{(B3)}$$

The zero-order term can be defined similarly resulting with a multiplication of complementary error functions.

In the reference [12] one can find a convergence proof for the series as well as other useful approximations.

Appendix C. Obtaining the final expression

One can use the saddle-point equations for E_i, F_{ik}, $\widehat{F}_{ik}$ and $\widetilde{F}_i$, replacing terms related to these order parameters by others. Expression (8) then includes three main terms.

(i) The hidden output term

$$\alpha_{\mathrm{c}}\int\left(\prod_k\frac{\mathrm{d}r_k}{\sqrt{2\pi}}\right)\frac{\mathrm{e}^{-\frac{1}{2}r^{\mathrm{T}}\mathcal{D}^{-1}r}}{|\mathcal{D}|^{\frac{1}{2}}}\log\left[\frac{1}{2}\int\left(\prod_k \mathrm{D}l_k\right)\mathrm{erfc}\left(\frac{l\cdot r}{\sqrt{2l^{T}(\mathcal{C}-\mathcal{D})l}}\right)\right]. \qquad \text{(C1)}$$

This term can be calculated analytically using integral identities for the complementary error function and modifying variables to polar coordinates. This results with the following expression:

$$\frac{1}{2}\alpha_{\mathrm{c}}\int_0^{\infty}\mathrm{d}\sigma\frac{M-\mathrm{tr}\left[I+\sigma(\mathcal{C}-\mathcal{D})\right]^{-1}}{\sigma\,|I-\sigma(\mathcal{C}-\mathcal{D})|^{\frac{1}{2}}} \qquad \text{(C2)}$$

where I is the identity matrix.

(ii) The input hidden term

$$\alpha_{\mathrm{c}}\int\left(\prod_k\frac{\mathrm{d}s_k}{\sqrt{2\pi}}\right)\frac{\mathrm{e}^{-\frac{1}{2}s^{\mathrm{T}}\mathcal{B}^{-1}s}}{|\mathcal{B}|^{\frac{1}{2}}}\log\left[\int_{s_k}^{\infty}\left(\prod_k\frac{\mathrm{d}\lambda_k}{\sqrt{2\pi}}\right)\frac{\mathrm{e}^{-\frac{1}{2}\lambda^{\mathrm{T}}(\mathcal{A}-\mathcal{B})^{-1}\lambda}}{|\mathcal{A}-\mathcal{B}|^{\frac{1}{2}}}\right]. \qquad \text{(C3)}$$

This term cannot be calculated analytically. We therefore use Kendall's approximation for the multi-dimensional integrations, obtaining the following expression:

$$\alpha_c \int \left(\prod_k \frac{ds_k}{\sqrt{2\pi}}\right) \frac{e^{-\frac{1}{2}s^T\mathcal{B}^{-1}s}}{|\mathcal{B}|^{\frac{1}{2}}} \log\left[\prod_k \tfrac{1}{2}\mathrm{erfc}\left(\frac{\tilde{s}_k}{\sqrt{2}}\right) + \sum_{j\neq k} \widetilde{(\mathcal{A}-\mathcal{B})}_{jk} \prod_l \left(\frac{e^{-\frac{1}{2}\tilde{s}_l^2}}{\sqrt{2\pi}}\right)\right] \tag{C4}$$

where

$$\widetilde{[\mathcal{A}-\mathcal{B}]}_{jk} \equiv \frac{[\mathcal{A}-\mathcal{B}]_{jk}}{\sqrt{[\mathcal{A}-\mathcal{B}]_{kk}[\mathcal{A}-\mathcal{B}]_{jj}}} \qquad \text{and} \qquad \tilde{s}_l \equiv \frac{s_l}{\sqrt{[\mathcal{A}-\mathcal{B}]_{ll}}}.$$

Examining this term one can see that there are two different terms in the argument of the logarithm; since the denominator of $\tilde{s}$ approaches zero, the second term in the logarithm's argument is dominant where all of the integration parameters are positive, while the first term is dominant in all other integration regions. Separating the integration in all of the various regions, one obtains the following expression for this term:

$$\alpha_c\left\{-\frac{1}{4}\sum_k \frac{\mathcal{B}_{kk}}{(\mathcal{A}-\mathcal{B})_{kk}} + \frac{1}{4}\sum_k \log\frac{(\mathcal{A}-\mathcal{B})_{kk}}{\mathcal{B}_{kk}} + \left[\log\sum_{k,j\neq k}\widetilde{(\mathcal{A}-\mathcal{B})}_{jk} - \frac{1}{2}\sum_k \log(\mathcal{A}-\mathcal{B})_{kk}\right]\int_0^\infty \left(\prod_k \frac{dr_k}{\sqrt{2\pi}}\right)\frac{e^{-\frac{1}{2}r^T\mathcal{B}^{-1}r}}{|\mathcal{B}|^{\frac{1}{2}}}\right\}. \tag{C5}$$

(iii) The ordering term

• For the $\varrho = 0$ case:

$$\frac{1}{MN}\int \left(\prod_k \frac{dv_k}{\sqrt{2\pi}}\right)\frac{e^{-\frac{1}{2}v^T\mathcal{S}^{-1}v}}{|\mathcal{S}|^{\frac{1}{2}}} \log\left[\int_0^\infty \left(\prod_k \frac{dv_k}{\sqrt{2\pi}}\right) e^{-\frac{1}{2}\eta^T(\mathcal{R}-\mathcal{S})\eta + i\eta\cdot v}\right]. \tag{C6}$$

Applying Kendall's approximation, the small argument expanding the complementary error function for small argument, and using the fact that

$$\Xi_k \equiv \frac{\left[(\mathcal{T}^T)^{-1}(\mathcal{R}-\mathcal{S})^{-1}v\right]_k}{\sqrt{\widetilde{[\mathcal{R}-\mathcal{S}]}^{-1}_{kk}}} \ll 1$$

one obtains for the first two powers in $\widetilde{[\mathcal{R}-\mathcal{S}]}^{-1}$:

$$\frac{1}{MN}\int \left(\prod_k \frac{dv_k}{\sqrt{2\pi}}\right)\frac{e^{-\frac{1}{2}v^T\mathcal{S}^{-1}v}}{|\mathcal{S}|^{\frac{1}{2}}}\left\{-\frac{1}{2}v^T[\mathcal{R}-\mathcal{S}]^{-1}v - \frac{1}{2}\log|\mathcal{R}-\mathcal{S}| + \log\left[\prod_k \frac{1}{2}\left(1 - i\sqrt{\frac{2}{\pi}}\Xi_k + \cdots\right) + \sum_{j,k\neq j}\widetilde{[\mathcal{R}-\mathcal{S}]}^{-1}_{jk}\prod_l\left(\frac{e^{\frac{1}{2}\Xi_l^2}}{\sqrt{2\pi}}\right)\right]\right\}. \tag{C7}$$

Using the small-argument approximation for the logarithm, one obtains the end term:

$$-\frac{1}{2MN}\log|\mathcal{R}-\mathcal{S}| - \frac{1}{2MN}\mathrm{tr}\left[(\mathcal{R}-\mathcal{S})^{-1}\mathcal{S}\right] - \frac{1}{MN\pi}\mathrm{tr}(\psi\mathcal{S}) + \left(\frac{2}{\pi}\right)^{M/2}\frac{1}{MN}\sum_{k,j\neq k}\widetilde{[\mathcal{R}-\mathcal{S}]}^{-1}_{jk}|I-\mathcal{S}\psi|^{-\frac{1}{2}} \tag{C8}$$

where $\psi \equiv [\mathcal{T}^T(\mathcal{R}-\mathcal{S})]^{-1}\widetilde{[\mathcal{R}-\mathcal{S}]}[(\mathcal{R}-\mathcal{S})\mathcal{T}]^{-1}$.

• For the $\varrho \neq 0$ case:

In the case of $\varrho \neq 0$ the expression becomes slightly different. Assuming that $\varrho \gg \sqrt{[\widetilde{\mathcal{R}-\mathcal{S}}]^{-1}_{kk}}$ (since ϱ is a small finite constant while $\sqrt{[\widetilde{\mathcal{R}-\mathcal{S}}]^{-1}_{kk}}$ approaches zero) we can use the large (positive) argument expansion for the complementary error function, obtaining

$$\frac{1}{MN}\int\left(\prod_k \frac{\mathrm{d}v_k}{\sqrt{2\pi}}\right)\frac{\mathrm{e}^{-\frac{1}{2}v^{\mathrm{T}}\mathcal{S}^{-1}v}}{|\mathcal{S}|^{\frac{1}{2}}}\left\{-\frac{1}{2}v^{\mathrm{T}}[\mathcal{R}-\mathcal{S}]^{-1}v-\frac{1}{2}\log|\mathcal{R}-\mathcal{S}|\right.$$
$$\left.+\log\left[\left(\prod_k\sqrt{\frac{2}{\pi}}\frac{\mathrm{e}^{-\frac{1}{2}\Xi_k^2}}{\Xi_k}\right)+\sum_{j,k\neq j}\left[\widetilde{\mathcal{R}-\mathcal{S}}\right]^{-1}_{jk}\prod_l\left(\frac{\mathrm{e}^{-\frac{1}{2}\Xi_l^2}}{\sqrt{2\pi}}\right)\right]\right\} \tag{C9}$$

redefining

$$\Xi_k \equiv \varrho/\sqrt{[\widetilde{\mathcal{R}-\mathcal{S}}]^{-1}_{kk}}-\mathrm{i}\frac{\left[\left(\mathcal{T}^{\mathrm{T}}\right)^{-1}(\mathcal{R}-\mathcal{S})^{-1}v\right]_k}{\sqrt{[\widetilde{\mathcal{R}-\mathcal{S}}]^{-1}_{kk}}}.$$

Considering the dominant term of the logarithm we obtain for the entire term:

$$-\frac{1}{2MN}\log|\mathcal{R}-\mathcal{S}|-\frac{1}{2MN}\mathrm{tr}\left[(\mathcal{R}-\mathcal{S})^{-1}\mathcal{S}\right]+\frac{1}{2MN}\mathrm{tr}\left[\psi S\right]$$
$$+\frac{1}{MN}\log\sum_{k,j\neq k}[\widetilde{\mathcal{R}-\mathcal{S}}]^{-1}_{jk}-\frac{1}{2MN}\frac{\varrho^2}{\mathrm{tr}(\widetilde{\mathcal{R}-\mathcal{S}})^{-1}} \tag{C10}$$

replacing (C8).
Combining expressions (C2), (C5) and (C8) yields (11) in the text.

References

[1] Gardner E 1987 *J. Phys. A: Math. Gen.* **21** 257
[2] Seung H S, Sompolinsky H and Tishby N 1993 *Phys. Rev.* A **45** 6056
[3] Hertz J A, Krogh A and Palmer R G 1990 *Introduction to the theory of neural computation* (Redwood City, CA: Addison-Wesley)
[4] Parisi G 1979 *Phys. Rev. Lett.* **43** 1754
[5] Albertini F, Sontag E D and Maillot V 1993 *Neural Networks* **6** 975
[6] Chen A M, Lu H and Hecht-Nielsen R 1993 *Neural Comput.* **5** 910
[7] Barkai E, Hansel D and Sompolinsky H 1992 *Phys. Rev.* A **45** 4146
[8] Engel A, Köhler H M, Tschepke F, Vollmayr H and Zippelius A 1992 *Phys. Rev.* A **45** 7590
[9] Schwarze H and Hertz J A 1992 *Europhys. Lett.* **20** 375
[10] Meir R and Fontanari J F 1993 *Network* **4** 381
[11] Fontanari J F and Meir R 1993 *J. Phys. A: Math. Gen.* **26** 1077
[12] Kendall M G 1941 *Biometrika* **32** 196
Kendall M G, Stuart A and Ord J K 1987 *Kendall's Advanced Theory of Statistics* vol 1, 5th edn (London: Griffin)
[13] Ingber L 1989 *Math. Comput. Modelling* **12** 967
[14] Kirkpatrick S, Gelatt C D and Vecchi M P 1983 *Science* **220** 671
[15] Priel A, Blatt M, Grossman T, Domany E and Kanter I 1993 Computational capabilities of restricted two layered perceptrons *Preprint*
[16] Vapnik V N and Chervonenkis A Ya 1971 *Th. Prob. Appl.* **16** 264
[17] Saad D 1993 *J. Phys. A: Math. Gen.* **26** 3757