Learning in Higher Order Boltzmann Machines using Linear Response

Martijn A.R. Leisink* and Hilbert J. Kappen†
Department of Biophysics
University of Nijmegen, Geert Grooteplein 21
NL 6525 EZ Nijmegen, The Netherlands ‡

November 10, 1999

*http://www.mbfys.kun.nl/~martijn
†http://www.mbfys.kun.nl/~bert
‡This research is supported by the Technology Foundation STW, applied science devision of NWO and the technology programme of the Ministry of Economic Affairs.

Abstract

We introduce an efficient method for learning and inference in higher order Boltzmann machines. The method is based on mean field theory with the linear response correction. We compute the correlations using the exact and the approximated method for a fully connected third order network of ten neurons. In addition, we compare the results of the exact and approximate learning algorithm. Finally we use the presented method to solve the shifter problem. We conclude that the linear response approximation gives good results as long as the couplings are not too large.

Keywords: Higher order Boltzmann machines, Linear response, Inference, Learning

1 Introduction

A Boltzmann machine is a network of stochastic binary variables (neurons). All neurons s_i are linked to each other with symmetric couplings $w_{ij} = w_{ji}$. Due to this symmetry the probability distribution is given by the Boltzmann-Gibbs distribution which is a known function of the couplings and thresholds of the network [1].

Sejnowski [15] introduced higher order Boltzmann machines (HOBM), which have also connections like w_{ijk} or w_{ijkl}. Those neural networks can be used to decrease the number of hidden neurons needed in an ordinary Boltzmann machine since the higher order connections can represent higher order correlations in the data. In case the order of the network equals the number of neurons, the system is capable to learn every distribution exactly without hidden neurons [3].

Since learning of a Boltzmann machine requires an amount of time proportional to 2^N, where N is the number of neurons, an approximation is necessary. One can use, for instance, Gibbs sampling [11] where the averages are approximated by taking some samples from the space of all the 2^N possible states of the neurons. This is, however, still a time consuming procedure. Another approximation method, known as naive mean field theory, where the correlations $\langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle$ are assumed to be zero, was applied to ordinary second order Boltzmann machines [13], but gives in general poor results.

Several improvements to mean field theory were made. Thouless, Anderson and Palmer (referred to with TAP) presented a correction to the mean field equations [18] known as the Onsager reaction term [12]. Other approaches are mixture models [10] and tractable graphs [4]. Kappen and Rodríguez [7] have shown an approximation method

called 'linear response' which is much more accurate than naive mean field and comparable to the TAP-approach. Moreover its calculation time is still polynomial. In this paper we extend this method to HOBM.

We describe the theory of an exact HOBM in the next section. In the third section we derive the linear response approximation for HOBM. We introduce the Gibbs free energy G as the Legendre transform of the free energy, F. The derivative of F with respect to the model parameters are the moments of the HOBM. We show that these are related to the derivatives of G through the Legendre transform. The mean field approximation results from a small coupling expansion of the Gibbs free energy [14].

The linear response approximation can be used to approximate the correlations given the couplings of the HOBM (the forward problem). This is useful for inference tasks, where one wants to compute the marginal probability of some neurons given the value of some others.

It can also be used for learning, where the couplings must be estimated from the correlations in the data (the backward problem). In this case a gradient based method must be used to adapt the couplings in such a way that the corresponding correlations are equal to the target correlations. In the case of no hidden units we solve the backward problem for HOBM in the linear response approximation without iteration.

In section 4 we compare the quality of the linear response approximation for the forward and backward problem and we solve the shifter problem as was done by [6].

2 Higher order Boltzmann machines

A higher order Boltzmann machine (HOBM) has a energy depending on its state given by

$$E(\vec{s}) = -\sum_i \theta_i s_i - \frac{1}{2}\sum_{ij} w_{ij} s_i s_j - \frac{1}{6}\sum_{ijk} w_{ijk} s_i s_j s_k - \dots \tag{1}$$

We use the notation θ_i instead of w_i since it will turn out that the thresholds are treated in another way than the other couplings.

The probability to find the system in equilibriuim in some state $\vec{s}$ is the Boltzmann-Gibbs distribution

$$P(\vec{s}) = \frac{1}{Z}\exp\left(-E(\vec{s})\right) \tag{2}$$

where Z is a normalization constant

$$Z = \sum_{\text{all } \vec{s}} \exp\left(-E(\vec{s})\right) \tag{3}$$

The learning problem is to set the thresholds and all the couplings such that some target distribution $Q(\vec{s})$ is approximated as closely as possible. Therefore one minimizes the distance between P and Q. A suitable measure of this distance is the Kullback-Leibler divergence, given by

$$K(Q,P) = \sum_{\text{all } \vec{s}} Q(\vec{s}) \log \frac{Q(\vec{s})}{P(\vec{s})} \tag{4}$$

since it stands for the average distance (in information theoretical sense) between the model and the observed data. Since the model is considered to be an approximation of the reality, the average is taken with respect to the data. For details see [9]. An important property is that $K = 0 \Leftrightarrow P = Q$ and $K > 0$ otherwise. Therefore a learning algorithm can be defined by minimizing this function [1]

$$\Delta\theta_i \sim \langle s_i \rangle_{\mathrm{c}} - \langle s_i \rangle_{\mathrm{f}}$$

$$\begin{aligned} \Delta w_{ij} &\sim \langle s_i s_j \rangle_{\mathrm{c}} - \langle s_i s_j \rangle_{\mathrm{f}} \\ \Delta w_{ijk} &\sim \langle s_i s_j s_k \rangle_{\mathrm{c}} - \langle s_i s_j s_k \rangle_{\mathrm{f}} \\ &\vdots \end{aligned} \tag{5}$$

The subscript f denotes an average taken in the free phase which corresponds to an average with respect to the distribution P. The subscript c denotes the clamped phase in which the visible neurons are set to values in the training set. In case of no hidden units this corresponds to an average with respect to the distribution Q.

3 Mean field for HOBM

We define the free energy F of the network as

$$F(\theta, w) = -\log Z = -\log \sum_{\text{all } \vec{s}} \exp\left(-E(\vec{s})\right) \tag{6}$$

We use the α-expansion as introduced by Plefka [14] to approximate the free energy. The energy of some state of the network is given by

$$E(\vec{s}, \alpha) = -\sum_i \theta_i s_i + \alpha E_{\mathrm{int}}(\vec{s}) \tag{7}$$

where E_{int} stands for the interaction energy given by

$$E_{\mathrm{int}}(\vec{s}) = -\frac{1}{2}\sum_{ij} w_{ij} s_i s_j - \frac{1}{6}\sum_{ijk} w_{ijk} s_i s_j s_k - \dots \tag{8}$$

For $\alpha = 1$ we have our original model; for $\alpha = 0$ we have a factorised model which is tractable.

We perform a Legendre transformation [5] where we replace the independent variable θ_i by $m_i \stackrel{\text{def}}{=} -\frac{\partial F}{\partial \theta_i} = \langle s_i \rangle_\alpha$ and hence we obtain

$$G(m, w, \alpha) = F(\theta, w, \alpha) + \sum_i \theta_i m_i \tag{9}$$

where G is the Legendre transform of F known as the Gibbs free energy. Instead of $m_i(\theta, w, \alpha)$ being a function of the thresholds and the couplings, now m_i is an independent variable and $\theta_i = \theta_i(m, w, \alpha)$.

We expand G around $\alpha = 0$:

$$G(m, w, 1) = G(m, w, 0) + \alpha G'(m, w, 0) + \mathcal{O}(\alpha^2) \tag{10}$$

where a prime denotes a differentiation with respect to α. Since G and G' are evaluated at $\alpha = 0$ (the factorised model), one can derive [14]

$$G(m, w, 0) = \sum_i \left\{ \frac{1+m_i}{2} \log \frac{1+m_i}{2} + \frac{1-m_i}{2} \log \frac{1-m_i}{2} \right\} \tag{11}$$

$$G'(m, w, 0) = -\frac{1}{2} \sum_{ij} w_{ij} m_i m_j - \frac{1}{6} \sum_{ijk} w_{ijk} m_i m_j m_k - \ldots = \langle E_{\text{int}} \rangle_0 \tag{12}$$

where $\langle \ldots \rangle_0$ denotes an expectation value taken with respect to $P(\vec{s})$ with $\alpha = 0$.

We approximate the Gibbs free energy by setting $\alpha = 1$

$$\begin{aligned} G(m, w) = & \sum_i \left\{ \frac{1+m_i}{2} \log \frac{1+m_i}{2} + \frac{1-m_i}{2} \log \frac{1-m_i}{2} \right\} \\ & + \langle E_{\text{int}} \rangle_0 + \mathcal{O}(w^2) \end{aligned} \tag{13}$$

From the Legendre transformation we furthermore know

$$\frac{\partial G}{\partial m_i} = \theta_i \tag{14}$$

Using the approximation for G (13) we obtain the generalised mean field equations:

$$m_i = \tanh\left(\theta_i - \frac{\partial}{\partial m_i} \langle E_{\text{int}} \rangle_0 \right) \tag{15}$$

In principle one can expand upto higher orders in α. For second order Boltzmann machines this was done by [8].

3.1 The forward problem

We will now show how to approximate the forward problem, i.e. to compute the correlations given the couplings and thresholds of the network.

First we define

$$F_{ijk\ldots} = \frac{\partial^{\cdots} F}{\partial\theta_i \partial\theta_j \partial\theta_k \ldots} \tag{16}$$

and

$$G_{ijk\ldots} = \frac{\partial^{\cdots} G}{\partial m_i \partial m_j \partial m_k \ldots} \tag{17}$$

F is the cumulant generating function of the distribution P and hence $F_{ijk\ldots}$ are the connected parts of the correlations. For instance

$$F_{ij} = -\langle s_i s_j\rangle + \langle s_i\rangle \langle s_j\rangle \tag{18}$$

$$F_{ijk} = -\langle s_i s_j s_k\rangle + \langle s_i s_j\rangle \langle s_k\rangle + \langle s_j s_k\rangle \langle s_i\rangle + \langle s_k s_i\rangle \langle s_j\rangle - 2\langle s_i\rangle \langle s_j\rangle \langle s_k\rangle \tag{19}$$

The expression

$$F_{ij} = -{G^{-1}}_{ij} \tag{20}$$

is a property of the Legendre transformation, as one can easily see by differentiating

$$m_i = -\frac{\partial F}{\partial\theta_i} \quad \text{and} \quad \theta_i = \frac{\partial G}{\partial m_i} \tag{21}$$

with respect to θ_j and m_j respectively.

When we differentiate both sides of equation (20) with respect to θ_k we obtain

$$\begin{aligned} F_{ijk} &= \sum_{\alpha\beta} {G^{-1}}_{i\alpha} {G^{-1}}_{j\beta} \frac{\mathrm{d}G_{\alpha\beta}}{\mathrm{d}\theta_k} \\ &= \sum_{\alpha\beta} {G^{-1}}_{i\alpha} {G^{-1}}_{j\beta} \left(\underbrace{\frac{\partial G_{\alpha\beta}}{\partial\theta_k}}_{=0} + \sum_{\gamma} \frac{\partial m_\gamma}{\partial\theta_k} \frac{\partial G_{\alpha\beta}}{\partial m_\gamma} \right) \\ &= \sum_{\alpha\beta\gamma} {G^{-1}}_{i\alpha} {G^{-1}}_{j\beta} {G^{-1}}_{k\gamma} G_{\alpha\beta\gamma} \end{aligned} \tag{22}$$

which expresses the third order derivatives of F in terms of derivatives of G. Differentiating again with respect to θ_l gives us an expression for the fourth order derivative and so on.

Thus given the couplings and thresholds of a HOBM we can compute m_i from (15) and compute G and its derivatives with respect to m_i from (13). Using (22) we compute the correlations. In this way we approximately solve the forward problem. Note that the linear response algorithm has a computational complexity which is polynomial in the number of neurons, whereas the exact learning method is exponential.

3.2 The backward problem

When there are no hidden neurons, all the clamped correlations in the learning rule (5) are known. Since the converged learning process requires

$$\Delta\theta_i \sim \langle s_i \rangle_{\mathrm{c}} - \langle s_i \rangle_{\mathrm{f}} = 0 \tag{23}$$

$$\Delta w_{ij} \sim \langle s_i s_j \rangle_{\mathrm{c}} - \langle s_i s_j \rangle_{\mathrm{f}} = 0 \quad i \neq j \tag{24}$$

$$\Delta w_{ijk} \sim \langle s_i s_j s_k \rangle_{\mathrm{c}} - \langle s_i s_j s_k \rangle_{\mathrm{f}} = 0 \quad i \neq j, i \neq k, j \neq k \tag{25}$$

$$\vdots$$

we know the free correlations have to be equal to the clamped ones. These equations can only be obtained, when all indices are different, since a Boltzmann distribution (2) does not depend on couplings with two or more indices equal (hereafter referred to as 'diagonal couplings').

However, we can find a solution to equation (23), (24) and (25) by imposing the stronger condition that (23), (24) and (25) must hold for any set of indices and allowing non-zero diagonal couplings. This 'trick' was succesfully used for second order

Boltzmann machines. In that case, these diagonal contributions effectively estimate the second order self coupling contributions [7]. Here, such a motivation is less evident.

Step 1

From (23) we can compute m_i, since $m_i = -F_i = \langle s_i \rangle_{\mathrm{f}}$. From (24) and (25) we can compute the derivatives of F using (18) and (19).

Step 2

The derivatives of G expressed in terms of the derivatives of F can be obtained as in 22. For instance

$$G_{ij} = F_{ij}^{-1} \tag{26}$$

$$G_{ijk} = \sum_{\alpha\beta\gamma} F_{\alpha i}^{-1} F_{\beta j}^{-1} F_{\gamma k}^{-1} F_{\alpha\beta\gamma} \tag{27}$$

Step 3

The derivatives of G are approximately given in terms of m_i and w by differentiating equation (13):

$$\begin{aligned}
\theta_i = \frac{\partial G}{\partial m_i} &= \tanh^{-1} m_i + \frac{\partial}{\partial m_i} \langle E_{\mathrm{int}} \rangle_0 \\
\frac{\partial^2 G}{\partial m_i \partial m_j} &= \frac{\delta_{ij}}{1 - m_i^2} + \frac{\partial^2}{\partial m_i \partial m_j} \langle E_{\mathrm{int}} \rangle_0 \\
\frac{\partial^3 G}{\partial m_i \partial m_j \partial m_k} &= \frac{\delta_{ijk} 2 m_i}{(1 - m_i^2)^2} + \frac{\partial^3}{\partial m_i \partial m_j \partial m_k} \langle E_{\mathrm{int}} \rangle_0 \\
&\vdots
\end{aligned} \tag{28}$$

Suppose the HOBM contains all couplings upto order k. The k-th order derivative of G only depends on m_i and the k-th order couplings. Since m_i is known we can calculate

these couplings. The $k-1$-th derivative of G only depends on m_i and the k-th and $k-1$-th order couplings. From this we can compute the couplings of order $k-1$. We repeat this and finally we find the thresholds (i.e. the first order couplings).

For instance, for a third order Boltzmann machine we used the following equations, which follow directly from (28).

$$\begin{aligned} w_{ijk} &= \frac{\delta_{ijk} 2 m_i}{\left(1-m_i^2\right)^2} - G_{ijk} \\ w_{ij} &= \frac{\delta_{ij}}{1-m_i^2} - \sum_k w_{ijk} m_k - G_{ij} \\ \theta_i &= \tanh^{-1} m_i - \sum_j w_{ij} m_j - \frac{1}{2} \sum_{jk} w_{ijk} m_j m_k \end{aligned} \tag{29}$$

Thus given the clamped correlations of the HOBM without hidden neurons we can approximate the couplings without iteration. We refer to this method as 'linear response with diagonal couplings' or simply 'linear response'.

4 Results

In this section we apply the above theory to third order Boltzmann machines.

4.1 The forward problem

We assess the quality of the approximation to compute the correlations of a given Boltzmann distribution Q with second and third order couplings:

$$Q(\vec{s}) = \frac{1}{Z} \exp\left(\frac{1}{2} \sum_{ij} w_{ij} s_i s_j + \frac{1}{6} \sum_{ijk} w_{ijk} s_i s_j s_k \right) \tag{30}$$

The couplings are randomly drawn from a Gaussian with zero mean and standard deviation $\sigma/\sqrt{N}$ for the second order interactions and σ/N for the third order ones. We plot the exact correlations versus the approximated ones for $\sigma = 0.2$, 0.3 and 0.4.

Since it is known that for second order models the mean field approximation breaks down at $\sigma > 1/2$ for large N [17], we took $\sigma < 1/2$. The network consists of 10 neurons, since in that case it is possible to compare the results with the exact calculations.

Figure 1 shows the graphs for $\langle s_i s_j \rangle$ and $\langle s_i s_j s_k \rangle$. One can see that the approximated second and third order correlations are almost equal to the exact ones for $\sigma = 0.2$. For $\sigma = 0.4$ the third order approximations are less accurate.

4.2 The backward problem

A Boltzmann machine with distribution $P(\vec{s})$ which has thresholds, second and third order couplings, is trained using the method from section 3.2. The target distribution $Q(\vec{s})$ in equation (4) is a Boltzmann distribution itself with only second and third order interactions as in equation (30). The couplings are random from a Gaussian with zero mean and a standard deviation $\sigma_2/\sqrt{N}$ for the second order and σ_3/N for the third order couplings. N is the number of neurons, which is ten again. For $\sigma_3 = 0$ this model corresponds to the Sherrington-Kirkpatrick (SK) model [16], which was used to train second order Boltzmann machines in [8].

Linear response is used to obtain the couplings and thresholds of $P(\vec{s})$. The difference between the target and $P(\vec{s})$ is measured using the Kullback divergence. If our approximation is perfect, the Kullback divergence is zero, since the task is realizable. Indeed, using the exact learning rule, the Boltzmann machine converges to a $K = 0$ solution.

In figure 2 the logarithm of the Kullback is shown versus σ_2 and σ_3. It is clear that in the case of large σ_2 and σ_3 our approximation is no longer valid, since it was based on a Taylor expansion with respect to the couplings. We conclude from the figure that

the linear response approximation is only allowed as long as $\sigma_2 < 0.6$ and $\sigma_3 < 0.8$ (since in that region the Kullback is small enough). The fact that the method does not break down at $\sigma_2 = 1/2$ but gradually deteriorates between $1/2 < \sigma_2 < 1$ can apparently be attributed to the fact that the diagonal couplings effectively play the role of the Onsager reaction term (TAP) [7].

4.3 The shifter problem

We applied the linear response method to the shifter problem as described in [6]. For this problem the neurons are divided into three groups. Group P_1 is a group of p neurons which are clamped to one of the 2^p binary patterns with equal probability. Group P_2 is again a group of p neurons and their states are determined by shifting the states of the neurons in P_1. Wrap-around is used so that a bit that is shifted out appears at the opposite side. The pattern can be shifted upto s positions to the left, upto s positions to the right or no shift with equal probability $\frac{1}{2s+1}$. The third group, S, has $2s+1$ neurons, which represent the possible shifts. In the original problem $p = 8$ and $s = 1$ (only left, right or no shift was allowed). The network had to recognize the shift, given the two patterns on P_1 and P_2.

Hinton and Sejnowski [6] used a second order Boltzmann machine with 24 hidden neurons to solve the problem. More recently, Albizuri et al. used a third order machine without hidden neurons [2]. We trained a third order network without hidden neurons for the original problem using the method from section 3.2 with one small change: Since there is always exactly one neuron in S which is on, the matrix of second order correlations, F_{ij}, has a zero eigenvalue. Therefore, in equation (20), we take the pseudo inverse instead of the true inverse. After training, the network had to solve two tasks:

1. The neurons in P_1 and P_2 are clamped and the network has to recognize the shift.

2. The neurons in P_1 and S are clamped and the neurons in P_2 have to output the shifted pattern.

The first task was perfectly done by the network. All shifts were recognized except 10 ambiguous cases: Six cases in which all neurons in P_1 are either plus or minus one (where the shifts can not be distinguished) and four cases where the neurons in P_1 are alternately plus and minus one (where a left and right shift can not be distinguished).

The second task was harder to solve. About 80% of the patterns were shifted correctly, i.e. the sign of all mean firing rates in P_2, computed using equation (15), was equal to the shifted pattern in the data set. In figure 3 is shown how many bits were correctly shifted in which percentage of the cases. In figure 4 there is an example of the output of the network: The input pattern that should be shifted to the right is shown at the left, at the right the mean firing rates of the neurons in P_2.

Our results on task 1 are much better than those obtained in [6], where an exact trained second order Boltzmann machine with 24 hidden neurons was used. On the contrary, the exact trained third order Boltzmann machine in [2] gave better results on task 1 and 2. This is not suprising, since linear response is an approximation.

The complexity of the linear response algorithm is $\mathcal{O}\left(N^4\right)$ instead of exponential. Therefore we can train much larger networks than the exact methods can. We have trained a large shifter problem where the size of the pattern, p, is 53 and the maximum shift, s, is 13. Hence, the size of the network is $N = 2p + 2s + 1 = 133$. There are 221,184 possible input patterns. It took 30 minutes on a pentium computer to train

the network with 5000 training and 1000 validation patterns. The shift was recognized correctly for 99% of the validation patterns.

5 Discussion

In this paper we have extended the linear response approximation to Boltzmann machines with higher order interactions. We have shown that good approximations can be obtained as long as the couplings needed to model the target distribution are not too large. Moreover we presented a solution for the backward problem without iteration, when there are only visible neurons. The calculation time of the algorithm is polynomial, whereas the time needed for the exact learning rule is exponential. For the specific case of third order Boltzmann machines it is $\mathcal{O}\left(N^4\right)$.

Although the theory presented is in principle valid for any order of the Boltzmann machine, only third and maybe fourth order will be useful in practice. The calculation time is polynomial but can take long anyway. Furthermore the storage of all the correlations might be a problem for networks of very high orders.

It is possible to expand the Gibbs free energy G upto the second order of α in equation (10). For a second order Boltzmann machine this brings in the TAP-term as was shown by Plefka [14] and Kappen et al. [8]. The same can be done for HOBM which will increase the accuracy of the estimation. The inversion of equation (28), however, will no longer be as easy as it is now since the derivatives of G will not be linear in the couplings. In this case we propose to use the gradient descent procedure to find a fixed point of the learning rule (5). It is, however, not yet clear if such a fixed point exists nor if it is unique. This gradient descent technique should also be used for learning with hidden neurons, although the couplings between the visible neurons can still be found without iteration.

References

[1] D. Ackley, G. Hinton, T. Sejnowski (1985). A learning algorithm for Boltzmann Machines. *Cognitive Science*, **9**, 147–169

[2] F.X. Albizuri et al (1995) *The High-Order Boltzmann Machine: Learned Distribution and Topology.* IEEE Transactions on Neural Networks 6(3): 767–770

[3] S. Amari, K. Kurata and H. Nagaoka (1992). Information Geometry of Boltzmann Machines. IEEE *Transactions on Neural Networks* **3**, No. 2, March 1992

[4] D. Barber and W. Wiegerinck (1998). Tractable Undirected Approximations for Graphical Models. *In L. Niklasson, M. Bodén and T. Ziemke, editors,* ICANN *98*, **1**, 93—98. ISBN 3 540 76263 9, Springer-Verlag

[5] I.N. Bronshtein and K.A. Semendyayev (1979). *Handbook of Mathematics*, pp. 430 ISBN 0 442 21171 6, Van Nostrand Reinhold Company Inc., New York

[6] G.E. Hinton and T.J. Sejnowski (1986). Learning and relearning in Boltzmann Machines. *Parallel Distributed Processing*, Rumelhart et al, Eds. Cambridge, MA: MIT Press, Vol. 1, Chap. 7

[7] H.J. Kappen and F.B. Rodríguez (1998). Efficient learning in Boltzmann Machines using linear response theory. *Neural Computation* **10(5)**, 1137–1156

[8] H.J. Kappen and F.B. Rodríguez (1997). Boltzmann Machine learning using mean field theory and linear response correction. *Proceedings* NIPS *1998*, 280–286

[9] S. Kullback (1959). Information theory and statistics. Wiley, New York

[10] N.D. Lawrence, C.M. Bishop and M.I. Jordan (1998). Mixture Representations for Inference and Learning in Boltzmann Machines. *Proceedings* NIPS, *1998*,

[11] R.M. Neal (1996). Bayesian Learning for Neural Networks. Springer-Verlag, ISBN 0 387 94724 8

[12] L. Onsager (1936) *Electric Moments of Molecules in Liquid.*

[13] C. Peterson and J. Anderson (1987). A mean field theory learning algorithm for neural networks. *Complex systems* **1**, 995—1019

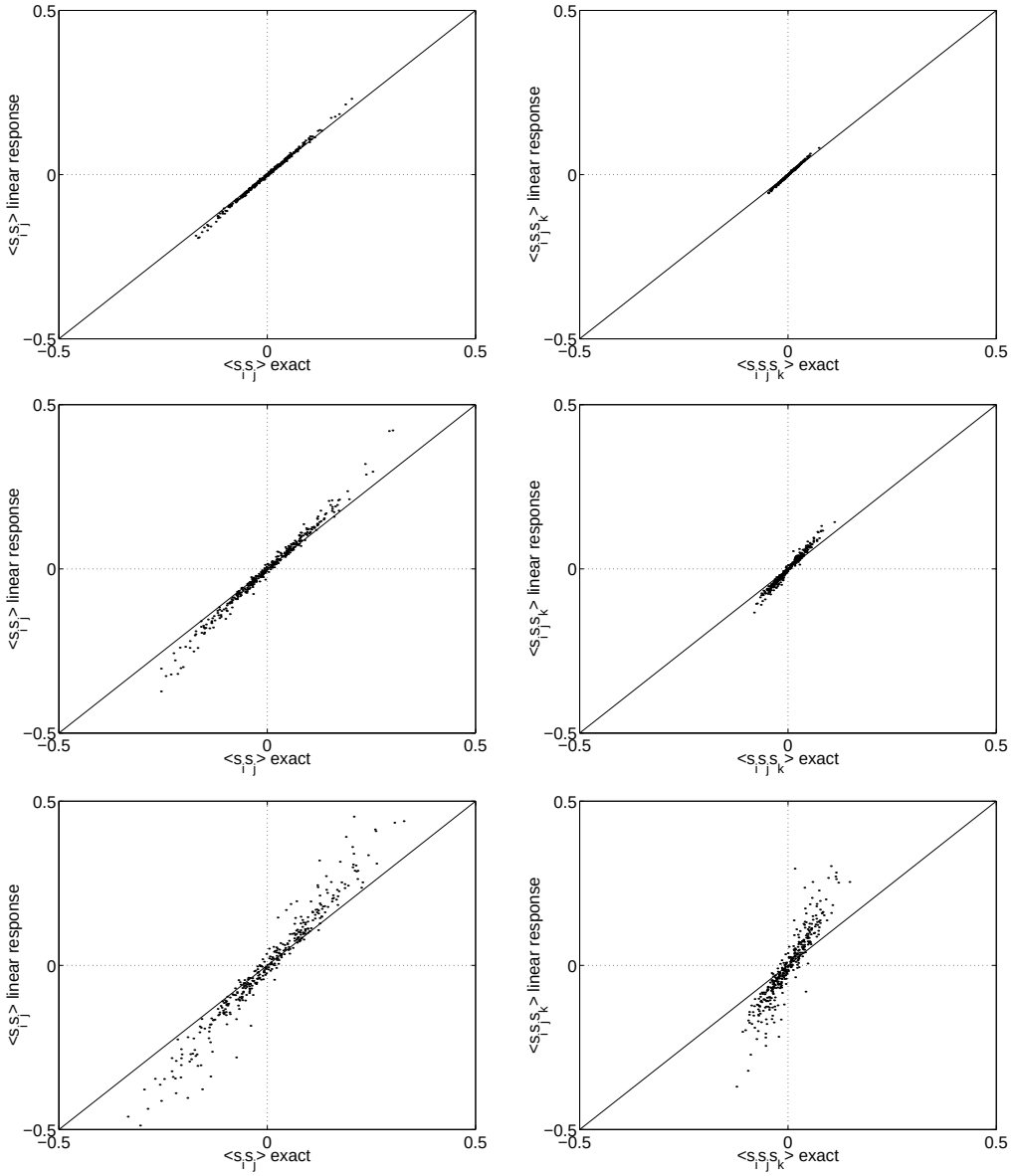
[14] T. Plefka (1982). Convergence condition of the TAP equation for the infinite-ranged Ising spin glass model. *Journal of Physics.A: Math.Gen.* **15**, 1971—1978

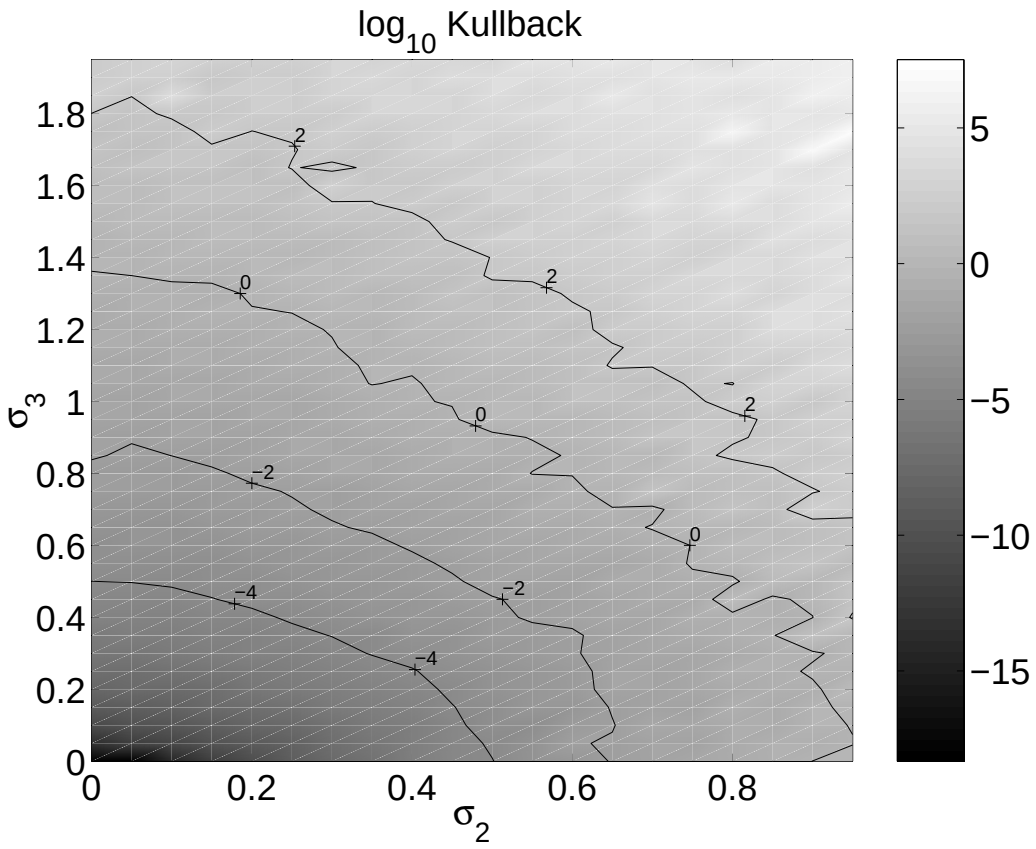
[15] T.J. Sejnowski (1986). Higher-Order Boltzmann Machines. In J.S. Denker (ed.) *AIP Conference Proceedings 151, Neural Networks for Computing*, 398–403, Snowbird, UT

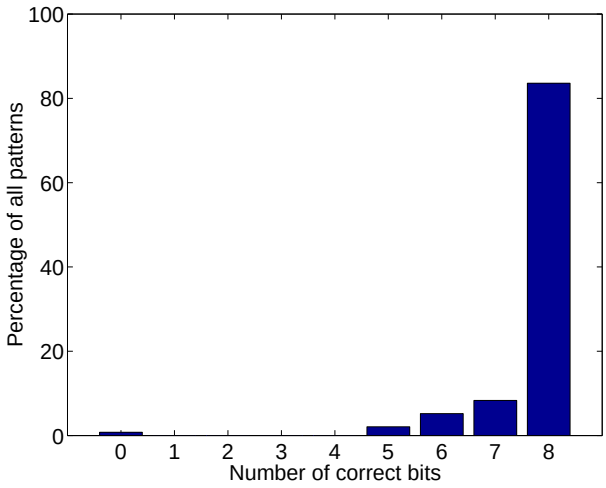
[16] D. Sherrington and S. Kirkpatrick (1975). *Physics Review Letter*, **32** 1792

[17] H. Takayama and K. Nemoto (1990). Spin glass properties of a class of mean-field models. *Journal of Physics: Condensed Matter*, **2**, 1997—2007

[18] D.J. Thouless, P.W. Andersson and R.G. Palmer (1977). Solution of 'Solvable model of a spin glass'. *Philisophical Magazine*, **35**(3): 593—601


0.5
0
−0.5
−0.5
0
0.5
<s_i s_j> linear response
<s_i s_j> exact
<s_i s_j s_k> linear response
<s_i s_j s_k> exact


log10 Kullback
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
σ3
0
0.2
0.4
0.6
0.8
σ2
2
0
−2
−4
5
0
−5
−10
−15


100
80
60
40
20
0
Percentage of all patterns
0
1
2
3
4
5
6
7
8
Number of correct bits

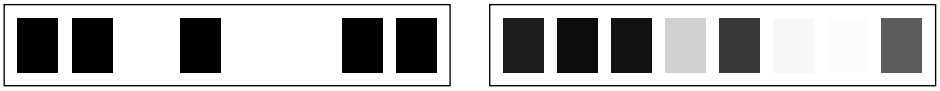


Figure 1: The estimation error of the second order correlations (left) and third order correlations (right) in a network of 10 neurons. The upper graphs have $\sigma = 0.2$, the middle graphs $\sigma = 0.3$ and the lower graphs $\sigma = 0.4$.

Figure 2: The logarithm of the Kullback divergence versus σ_2 and σ_3. Each point was an average over 50 random problems.

Figure 3: The percentage of the total data set versus the number of correct bits.

Figure 4: The input pattern (left) is shifted to the right by the network (right).