Radboud Repository

Radboud University Nijmegen

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

Efficient learning in Boltzmann Machines using linear response theory*

H.J. Kappen[†]and F. B. Rodríguez[‡]

October 29, 1997

Abstract

The learning process in Boltzmann Machines is computationally very expensive. The computational complexity of the exact algorithm is exponential in the number of neurons. We present a new approximate learning algorithm for Boltzmann Machines, which is based on mean field theory and the linear response theorem. The computational complexity of the algorithm is cubic in the number of neurons.

In the absence of hidden units, we show how the weights can be directly computed from the fixed point equation of the learning rules. Thus, in this case we do not need to use a gradient descent procedure for the learning process. We show that the solutions of this method are close to the optimal solutions and give a significant improvement when correlations play a significant role. Finally, we apply the method to a pattern completion task and show good performance for networks up to 100 neurons.

1 Introduction

Boltzmann Machines (BMs) (Ackley et al., 1985), are networks of binary neurons with a stochastic neuron dynamics, known as Glauber dynamics. Assuming symmetric connections between neurons, the probability distribution over neuron states $\vec{s}$ will become stationary and will be given by the Boltzmann-Gibbs distribution $P(\vec{s})$. The Boltzmann distribution is a known function of the weights and thresholds of the network. However, exact computation of $P(\vec{s})$ or any statistics involving $P(\vec{s})$, such as mean firing rates or correlations, requires exponential time in the number of neurons. This is due to the fact that $P(\vec{s})$ contains a normalization term Z, which involves a sum over all states in the network, of which there are exponentially many. This problem is particularly important for BM learning. This is because the BM learning rule requires the computation of correlations between neurons. Thus, learning in BMs requires exponential time.

For specific architectures, learning can be dramatically accelerated. For instance (Saul and Jordan, 1994) discuss how learning times become linear in the number of neurons for tree-like architectures. (Kappen, 1995) show how strong inhibition between hidden neurons reduces the computation time to polynomial in the number of neurons.

A well-known approximate method to compute correlations is the Monte Carlo method (Itzykson and Drouffe, 1989), which is a stochastic sampling of the state space. Glauber dynamics is an example of such a method. The terms in the sum over states are proportional to a 'Boltzmann factor' $\exp(-E)$. Monte Carlo methods can be more effective than the summation of all terms because the sampling is biased towards states with lower E. These terms will give the dominant contribution to the sum over states. This is the approach chosen for learning in the original Boltzmann Machine (Ackley et al., 1985). Practical use requires that the Markov process converges sufficiently fast, i.e. in polynomial time, to the equilibrium distribution. This property is known as rapid mixing and does probably not hold in general for Glauber dynamics (Sinclair, 1993). Useful results can be obtained with Glauber dynamics when the network is not too large and has small weights.

In (Peterson and Anderson, 1987), an acceleration method for learning in BMs is proposed. They suggest to replace the correlations in the BM learning rule by the naive mean field approximation: $\langle s_i s_j \rangle = m_i m_j$,

---

*To appear in Neural Computation

[†]RWCP SNN Laboratory, Department of Biophysics, University of Nijmegen, Geert Grooteplein 21, NL 6525 EZ Nijmegen, The Netherlands

[‡]Instituto de Ingeniería del Conocimiento & Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, Canto Blanco,28049 Madrid, Spain.

1

where m_i is the mean field activity of neuron i. The mean fields are given by the solution of a set of n coupled mean field equations, with n the number of neurons. The solution can be efficiently obtained by fixed point iteration. The method was further elaborated in (Hinton, 1989). In this paper, we will show that the naive mean field approximation of the learning rules does not converge in general and explain why.

Another way to speed-up learning is to observe that the Kullback-Leibler divergence is bounded from above by an effective free energy expression using Jensen's inequality. Such an approach can be applied to architectures whose probability distribution does not contain a sum over all states for normalization, such as the Helmholz Machine (Dayan et al., 1995) and the sigmoid belief network (Saul et al., 1996). The application of such an approach to Boltzmann Machines is not as simple because it requires in addition an upper bound on Z, which is computationally more complex (Jaakkola and Jordan, 1996).

We will argue, that in the correct treatment of mean field theory for BMs, the correlations can be computed using the linear response theorem (Parisi, 1988). In the context of neural networks this approach was first introduced by (Ginzburg and Sompolinsky, 1994) for the computation of time-delayed correlations and later by (Kappen, 1997) for the computation of stimulus dependent correlations. We will show, that this approximation can be used succesfully to approximate the gradients in the Boltzmann Machine.

This paper is organized as follows. In Section 2, we introduce learning in Boltzmann Machines and show why the naive mean field approximation of the gradients does not work. In Section 3, we derive the mean field approximation for the correlations based on the linear response theory. We argue that an effective self-coupling term can be included to obtain better results. In the absence of hidden units, the fixed point equations for the learning rules can be solved directly in terms of the weights and thresholds of the network. In Section 4, we show results of simulations. We compare our methods with the exact computation of the optimal weights and with a factorized probability model that assumes absence of correlations. We use the Kullback-Leibler divergence as a criterion for comparison on small networks. However for large networks, this criterion can no longer be computed, because it requires exponential time. We propose an approximate criterion for comparison on large networks and show that it correlates well with the Kullback-Leibler divergence for small problems. Subsequently we show good performance of our method for increasing problem size.

2 Boltzmann Machine learning

2.1 General Dynamics of Boltzmann Machines

The Boltzmann Machine is defined as follows. The possible configurations of the network can be characterized by a vector $\vec{s} = (s_1, .., s_i, .., s_n)$, where s_i is the state of the neuron i, and n the total number of the neurons. Each neuron can be in two states $(s_i = \pm 1)$ and its dynamics is governed by the following stochastic rule. At each time step, a neuron is selected ad random. Its new value is determined as:

$$s_i = \begin{cases} +1 & \text{with probability } g(h_i) \\ -1 & \text{with probability } 1 - g(h_i) \end{cases} \tag{1}$$

with $g(h_i)$ and h_i (local field) defined by

$$g(h_i) = \frac{1}{1 + \exp\{-2\beta h_i)\}}, \quad h_i = \sum_{j \neq i} w_{ij} s_j + \theta_i. \tag{2}$$

The magnitude w_{ij} (weight) refers to the connection strength between the neuron i and neuron j, and θ_i is the threshold for neuron i. The weights are chosen symmetric, $w_{ij} = w_{ji}$. The parameter β controls the noise in the neuron dynamics. β is often interpreted as $\beta = \frac{1}{T}$, where T acts like the temperature of a physical system. Since β is just a scaling of the weights and the thresholds, and the latter are optimized through learning, we can set $\beta = 1$ without loss of generality.

Let us define the energy of the system for a certain configuration $\vec{s}$ as

$$-E(\vec{s}) = \sum_{i<j} w_{ij} s_i s_j + \sum_i s_i \theta_i. \tag{3}$$

After long times, the probability to find the network in a state $\vec{s}$ becomes independent of time (thermal equilibrium) and is given by the Boltzmann distribution

$$p(\vec{s}) = \frac{1}{Z} \exp\{-E(\vec{s})\}. \tag{4}$$

$Z = \sum_{\vec{s}} \exp\{-E(\vec{s})\}$ is the partition function which normalizes the probability distribution.

2.2 Slow learning in Boltzmann Machines

A learning rule for Boltzmann Machines was introduced by Ackley, Hinton and Sejnowski (Ackley et al., 1985). Let us partition the neurons in a set of n_v visible units and n_h hidden units ($n_v + n_h = n$). Let α and β label the 2^{n_v} visible and 2^{n_h} hidden states of the network, respectively. Thus, every state $\vec{s}$ is uniquely described by a tuple $\alpha\beta$. Learning consists of adjusting the weighs and thresholds in such a way that the Boltzmann distribution on the visible units $p_\alpha = \sum_\beta p_{\alpha\beta}$ approximates a target distribution q_α as closely as possible.

A suitable measure for the difference between the distributions p_α and q_α is the Kullback divergence (Kullback, 1959)

$$K = \sum_\alpha q_\alpha \log \frac{q_\alpha}{p_\alpha}. \tag{5}$$

It is easy to show that $K \geq 0$ for all distributions p_α and $K = 0$ iff $p_\alpha = q_\alpha$ for all α.

Therefore, learning consists of minimizing K using gradient descent and the learning rules are given by (Ackley et al., 1985; Hertz et al., 1991)

$$\Delta\theta_i = \eta\left(\langle s_i\rangle_c - \langle s_i\rangle\right), \quad \Delta w_{ij} = \eta\left(\langle s_i s_j\rangle_c - \langle s_i s_j\rangle\right) \ i \neq j. \tag{6}$$

The parameter η is the learning rate. The brackets $\langle \cdot \rangle$ and $\langle \cdot \rangle_c$ denote the 'free' and 'clamped' expectation values, respectively. The 'free' expectation values are defined as usual:

$$\begin{aligned}
\langle s_i \rangle &= \sum_{\alpha\beta} s_i^{\alpha\beta} p_{\alpha\beta} \\
\langle s_i s_j \rangle &= \sum_{\vec{s}} s_i^{\alpha\beta} s_j^{\alpha\beta} p_{\alpha\beta}.
\end{aligned} \tag{7}$$

The 'clamped' expectation values are obtained by clamping the visible units in a state α and taking the expectation value with respect to q_α:

$$\begin{aligned}
\langle s_i \rangle_c &= \sum_{\alpha\beta} s_i^{\alpha\beta} q_\alpha p_{\beta|\alpha} \\
\langle s_i s_j \rangle_c &= \sum_{\alpha\beta} s_i^{\alpha\beta} s_j^{\alpha\beta} q_\alpha p_{\beta|\alpha}
\end{aligned} \tag{8}$$

$s_i^{\alpha\beta}$ is the value of neuron i when the network is in state $\alpha\beta$. $p_{\beta|\alpha}$ is the conditional probability to observe hidden state β given that the visible state is α. Note that in Eqs. 6–8, i and j run over both visible and hidden units.

Thus, the BM learning rules contain clamped and free expectation values of the Boltzmann distribution. The computation of the free expectation values is intractible, because the sums in Eqs. 7 consist of 2^n terms. If q_α is given in the form of a training set of p patterns, the computation of the clamped expectation values, Eqs. 8, contains $p2^{n_h}$ terms. This is intractible as well, but usually less expensive than the free expectation values. As a result, the BM learning algorithm can not be applied to practical problems.

2.3 The naive mean field approximation

Peterson and Anderson (Peterson and Anderson, 1987) proposed an approximation to calculate the expectation values based on mean field theory. In their approach, the free and clamped expectation values in Eq. 6 are approximated by their mean field values

$$\langle s_i \rangle \approx m_i, \quad \langle s_i s_j \rangle \approx m_i m_j, \ i \neq j, \tag{9}$$

where m_i is the solution to the set of coupled mean field equations

$$m_i = \tanh\left(\sum_{j \neq i} w_{ij} m_j + \theta_i\right). \tag{10}$$

We will refer to this method as the naive mean field approximation. In each step of the gradient descent procedure, one must solve the mean field equations given by Eq. 10. This can be done quite easily using fixed point iteration. In Section 3, we will give more details about mean field theory.

3

Peterson and Anderson found that this method was 10 to 30 times faster than the Monte Carlo method. However, there are many data sets for which the naive mean field approximation does not work. Here, we show the consequences of their approach in the case that there are no hidden units.

Consider a network with only two visible neurons and no hidded neurons. We want to learn the probability distribution given by two patterns $(1,1)$ and $(-1,-1)$ with equal probability. Thus, $\langle s_1 \rangle_c = \langle s_2 \rangle_c = 0$ and $\langle s_1 s_2 \rangle_c = 1$.

On this particular problem, the gradient descent procedure combined with the naive mean field computation does not converge. The reason is very simple. If we assume that the learning process converges to fixed point ($\Delta w_{ij} = 0$ and $\Delta \theta_i = 0$) then we obtain from Eqs. 6 and 9

$$\langle s_i \rangle_c = m_i, \quad \langle s_i s_j \rangle_c = m_i m_j \ i \neq j.$$

Thus, the fixed point equations of the learning process combined with the naive mean field approximation imply that the data set has no non-trivial correlations. In our example, this condition is clearly violated, since $0 = \langle s_1 \rangle_c \langle s_2 \rangle_c \neq \langle s_1 s_2 \rangle_c = 1$.

Thus, we expect that if we use the naive mean field approximation for the computation of the gradients, the resulting learning process will not converge. This is illustrated in Fig. 1. We compare the exact gradient descent method, where the correlations are calculated using Eqs. 7, and gradient descent using the naive mean field approximation. Although the mean field method sometimes reaches close to optimal solutions, the gradients Eqs. 6 are not zero at these points and therefore the solution does not remain there.
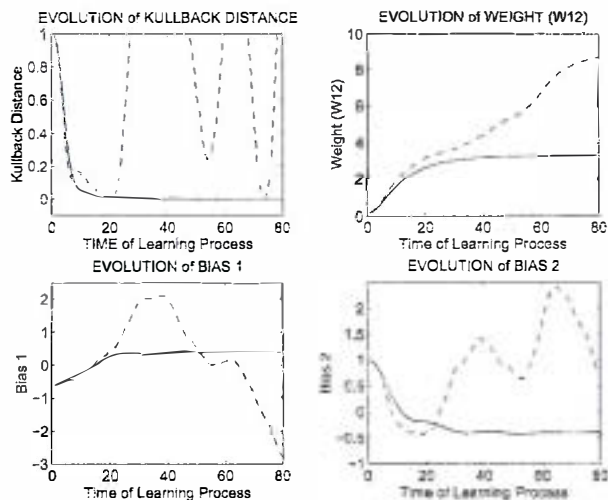

Figure 1: Gradient descent learning. The network consists of 2 visible neurons and no hidden neurons. The target distribution q is given by two patterns $(1,1)$ and $(-1,-1)$ with equal probability. The solid line shows the evolution of the Kullback divergence and the different network parameters when the exact gradient descent method is used. The dotted line shows the evolution of the different network parameters when the naive mean field approximation gradient descent procedure is used. Learning rate $\eta = 0.1$, momentum $\alpha = 0.9$

From this example we conclude that the naive mean field approximation leads to a converging gradient descent algorithm *only* when the data are such that

$$\langle s_i s_j \rangle_c = \langle s_i \rangle_c \langle s_j \rangle_c \ i \neq j. \tag{11}$$

For i and j visible units, this is simply a property of the data. It is equivalent to the statement that the target probability distribution q_α is factorized in all its variables: $q(\vec{s}) = \Pi_i q_i(s_i)$. The quality of the naive mean field approximation will depend on to what extend Eq. 11 is violated. This conclusion holds regardless of whether the network has hidden units or not.

3 The mean field method and the linear response correction

In this Section we introduce an improved method to compute correlations within the mean field framework. We will consider the mean field approximation and its formulation in the first subsection. In the following

4

subsection we will derive our main result based on the linear response theory. In the special case that the network has no hidden units, the optimal weights and thresholds can be computed directly from the fixed point equations, i.e. no gradient procedure needs to be applied.

3.1 Mean field formulation

The basic idea of mean field theory is to replace the quadratic term in the energy, $w_{ij}s_is_j$ in Eq. 3, by a term linear in s_i. Such a linearized form allows for efficient computation of the sum over all states, such as Eqs. 7 and 8 and the partition function Z. We define the mean field energy

$$-E_{mf}(\vec{s}) = \sum_i s_i\{W_i + \theta_i\} \tag{12}$$

where we introduce n mean fields W_i. The mean fields approximate the lateral interaction between neurons. The values of W_i must be chosen such that this approximation is as good as possible. How to do this will be shown below.

We define the mean field probability distribution as

$$p_{mf}(\vec{s}) = \frac{\exp\{-E_{mf}(\vec{s})\}}{Z_{mf}}. \tag{13}$$

with

$$Z_{mf} = \sum_{\vec{s}} \exp\{-E_{mf}(\vec{s})\} = \prod_i 2\cosh(\theta_i + W_i) \tag{14}$$

the mean field partition function.

The expectations values for s_i and s_is_j in the mean field approximation are given by:

$$\langle s_i \rangle_{mf} \equiv \sum_{\vec{s}} s_i p_{mf}(\vec{s}) = \tanh(W_i + \theta_i) \equiv m_i, \tag{15}$$

$$\langle s_is_j \rangle_{mf} \equiv \sum_{\vec{s}} s_is_j p_{mf}(\vec{s}) = m_im_j \quad i \neq j, \tag{16}$$

where we have introduced the parameters m_i, which are still to be fixed because of their dependence on W_i.

The real partition function Z, Eq. 4, can be computed in the mean field approximation (Itzykson and Drouffe, 1989):

$$\begin{aligned} Z &= \sum_{\vec{s}} \exp(-E) = \sum_{\vec{s}} \exp(-E_{mf} + E_{mf} - E) \\ &= Z_{mf} \langle \exp(E_{mf} - E) \rangle_{mf} \approx Z_{mf} \exp(\langle E_{mf} - E \rangle_{mf}) = Z'. \end{aligned} \tag{17}$$

The mean field approximation is in the last step and is related to the convexity of the exponential function $\langle \exp f \rangle \geq \exp \langle f \rangle$ (Itzykson and Drouffe, 1989). Note that $\langle \cdot \rangle_{mf}$ denotes expectation with respect to the mean field distribution Eq. 13 and not with respect to the Boltzmann distribution Eq. 4. Therefore, the free energy in the mean field approximation can be easily computed and is given by

$$-F = \log Z' = \sum_i \log(2\cosh(\theta_i + W_i)) - \sum_i W_im_i + \sum_{i<j} w_{ij}m_im_j \tag{18}$$

We can calculate the mean fields W_i, by minimizing the free energy:

$$\frac{\partial F}{\partial W_i} = (1 - m_i^2)(W_i - \sum_{j \neq i} w_{ij}m_j) = 0. \tag{19}$$

It can be shown, that the solutions $m_i^2 = 1$ maximize F. The required minima are therefore given by $W_i = \sum_{j \neq i} w_{ij}m_j$, which, combined with equation Eq. 15, give the mean field equations Eq. 10. These equations can be solved for m_i in terms of w_{ij} and θ_i using fixed point iteration. The mean fields W_i can then be directly computed using Eq. 19.

3.2 Derivation of linear response correction

We can go beyond the naive mean field prediction $\langle s_i s_j \rangle_{mf} = m_i m_j$ of Eq. 16 in the following way. First observe that the mean firing rates and correlations are

$$\langle s_i \rangle = \frac{1}{Z}\frac{dZ}{d\theta_j} \approx \frac{1}{Z'}\frac{dZ'}{d\theta_j}, \quad \langle s_i s_j \rangle \approx \frac{1}{Z'}\frac{d^2 Z'}{d\theta_i d\theta_j}. \tag{20}$$

We will compute these quantities using the approximation Eq. 17. While computing $\frac{dZ}{d\theta_j}$, using Eq. 18, we must be aware that the mean fields W_i depend on θ_i through Eq. 10 and Eq. 19:

$$\langle s_i \rangle \approx \frac{d}{d\theta_i}\log Z' = \left(\frac{\partial}{\partial\theta_i} + \sum_j \frac{\partial W_j}{\partial\theta_i}\frac{\partial}{\partial W_j} \right)\log Z' = m_i \tag{21}$$

$$\langle s_i s_j \rangle \approx \frac{1}{Z'}\frac{d}{d\theta_j}\left(Z' m_i \right) = m_i m_j + A_{ij}, \tag{22}$$

with $A_{ij} = \frac{dm_i}{d\theta_j}$. The last step in Eq. 21 follows when we use the mean field equations Eq. 19. Thus, there are no linear response corrections to the mean firing rate. Eq. 22 is known as the linear response theorem (Parisi, 1988). The inverse of the matrix A can be directly obtained by differentiating Eq. 10 with respect to θ_j. The result is:

$$(A^{-1})_{ij} = \frac{\delta_{ij}}{1 - m_i^2} - w_{ij} \tag{23}$$

When the network is divided into visible and hidden units, the above approximation can be directly applied to computation of the free expectation values Eqs. 7.

When the visible units are clamped, the above derivation can be repeated to compute the expectation values for the hidden units. The only difference is that the thresholds θ_i for the hidden units receive an extra contribution from the clamped visible neurons. Let us assume that the visible units are clamped in state α. The mean firing rates of the hidden units are denoted by $\langle s_i \rangle^\alpha = m_i^\alpha, i \in H$ where m_i^α satisfy the mean field equations

$$m_i^\alpha = \tanh(\sum_{j \in H} w_{ij}m_j^\alpha + \sum_{j \in V} w_{ij}s_j^\alpha + \theta_i), i \in H. \tag{24}$$

V and H denote the subsets of visible and hidden units, respectively. Note, that m_i^α depends on the clamped state α. The correlations $\langle s_i s_j \rangle^\alpha$ are given as follows:

$$i, j \in H \quad : \quad \langle s_i s_j \rangle^\alpha = m_i^\alpha m_j^\alpha + A_{ij}^\alpha \tag{25}$$

$$i \in V, j \in H \quad : \quad \langle s_i s_j \rangle^\alpha = s_i^\alpha m_j^\alpha \tag{26}$$

$$i, j \in V \quad : \quad \langle s_i s_j \rangle^\alpha = s_i^\alpha s_j^\alpha \tag{27}$$

$$(A^{\alpha, -1})_{ij} = \frac{\delta_{ij}}{1 - (m_i^\alpha)^2} - w_{ij} \tag{28}$$

Finally, the clamped expectation values are given by taking the expectation value over q_α: $\langle s_i \rangle_c = \sum_\alpha \langle s_i \rangle^\alpha q_\alpha$ and $\langle s_i s_j \rangle_c = \sum_\alpha \langle s_i s_j \rangle^\alpha q_\alpha$.

Thus, our approximation consists of replacing the clamped and free expectation values in Eqs. 6 by their linear response approximations. Eqs. 10, 21-23 and Eqs. 24-28 define the linear response approximations in the free phase and the clamped phase, respectively. The complexity of the method is dominated by the computations in the free phase. The computation of the linear response correlations involves the inversion of the matrix A, which requires $\mathcal{O}(n^3)$ operations. The computation of the mean firing rates through fixed point iteration of Eq. 10 requires $\mathcal{O}(n^2)$ or $\mathcal{O}(n^2 \log n)$ operations, depending on whether fixed precision in the components of m_i or in the vector norm $\sum_i m_i^2$ is required. Thus, the full mean field approximation, including the linear response correction, computes the gradients in $\mathcal{O}(n^3)$ operations.

3.3 TAP correction to the mean field equations

It is well-known that the standard mean field description Eq. 18 is inadequate for the description of frustrated systems. In general, terms involving higher powers of the coupling matrix w_{ij} must be included. For example,

6

for the Sherrington-Kirkpatrick (SK) model the appropriate mean field free energy becomes (Thouless et al., 1977)

$$-F = \sum_i \log(2\cosh(\theta_i + W_i)) - \sum_i W_i m_i + \frac{1}{2}\sum_{i,j} w_{ij} m_i m_j + \frac{1}{4}\sum_{i,j} w_{ij}^2 (1 - m_i^2)(1 - m_j^2), \quad (29)$$

and the corresponding mean field equations become the TAP equations:

$$m_i = \tanh\left(\sum_{j\neq i} w_{ij} m_j + \theta_i - m_i \sum_{j\neq i} w_{ij}^2 (1 - m_j^2)\right). \quad (30)$$

The additional term is called the Onsager reaction term (Onsager, 1936). It describes how the mean firing of neuron i affects the polarization of the surrounding spins and thus affect the local field of spin i. The effect of this additional term, but in the absence of the linear response correction, was studied by (Galland, 1993). In general there is an infinite sum of terms, each involving a higher power of the couplings w_{ij} (Fischer and Hertz, 1991). It is interesting to note that all higher order terms in the fixed point equation are proportional to m_i and thus represent corrections to the self-coupling term. In the case of the SK model, it can be shown that all terms beyond the Onsager term are negligible (Plefka, 1982). (For unfrustrated systems, like the Ising model, the Onsager term itself is negligible).

One can obtain the linear response corrections for TAP and higher order mean field corrections in a similar way as was described above, i.e. by variation around the TAP equations. These extensions will be explored in a future publication. In this paper, we will restrict ourselves to the linear response corrections to the lowest order mean field equations and ignore higher order corrections. However, we will consider the effect of an *effective* self-coupling term $w_{ii} m_i$. The mean field equations Eq. 10 become

$$m_i = \tanh\left(\sum_j w_{ij} m_j + \theta_i\right), \quad (31)$$

where the sum now includes the diagonal term. The derivation of the linear response correction is unaltered, except that w_{ij} has now non-zero diagonal terms (e.g. in Eq. 23). We propose to fix the value of w_{ii} through learning. We will demonstrate that the inclusion of the self-coupling term is 1) beneficial to obtain a closed form solution for the learning problem in the absence of hidden units and 2) gives significantly better results than without the self-coupling term.

3.4   No hidden units

For the special case of a network without hidden units and with the effective self-coupling we can make significant simplifications. In this case, the gradients Eqs. 6 can be set equal to zero and can be solved directly in terms of the weights and thresholds, i.e. no 'gradient based learning' is required. First note, that $\langle s_i \rangle_c$ and $\langle s_i s_j \rangle_c$ can be computed exactly from the data for all i and j. Let us define $C_{ij} = \langle s_i s_j \rangle_c - \langle s_i \rangle_c \langle s_j \rangle_c$.

The fixed point equation for $\Delta\theta_i$ gives

$$\Delta\theta_i = 0 \Leftrightarrow m_i = \langle s_i \rangle_c. \quad (32)$$

The fixed point equation for Δw_{ij}, using Eq. 32, gives

$$\Delta w_{ij} = 0 \Leftrightarrow A_{ij} = C_{ij} \; i \neq j. \quad (33)$$

Because we have introduced n self-coupling parameters, we must specify n additional constraints. An obvious choice is to ensure that $\langle s_i^2 \rangle = 1$ is also true in the linear response approximation: $1 = \langle s_i^2 \rangle_{lr} = m_i^2 + A_{ii} \Leftrightarrow A_{ii} = C_{ii}$. Then, Eq. 33 is equivalent to $(A^{-1})_{ij} = (C^{-1})_{ij}$ if C is invertible. Using Eq. 23 we obtain

$$w_{ij} = \frac{\delta_{ij}}{1 - m_i^2} - (C^{-1})_{ij} \quad (34)$$

In this way we have solved m_i and w_{ij} directly from the fixed point equations. The thresholds θ_i can now be computed from Eq. 10:

$$\theta_i = \tanh^{-1}(m_i) - \sum_j w_{ij} m_j \quad (35)$$

Note, that this method does not require fixed point iterations to obtain mean firing rates m_i in terms of w_{ij} and θ_i. Instead, the 'inverse' computation of θ_i given m_i and w_{ij} is required in Eq. 35. Note also, that the thresholds depend on the diagonal weights. The solution of the example task of two neurons discussed in section 2.3 is computed in the appendix.

Although the above choice of constraint is particular convenient, we should keep in mind that in principle other choices could be made, leading to other solutions. The justification for our choice is that it gives a closed form solution of high quality, as we will show.

4 Results

In this Section we will compare the accuracy of the linear response correction with and without self-coupling with the exact method and with a factorized model that ignores correlations. We restrict ourselves to networks without hidden units. Of course, there are many probability estimation problems, for which the BM without hidden units is a poor model. Our main concern is whether the linear response approximation will give a solution which is sufficiently close to the optimal solution, and not whether the optimal solution is good or bad.

The correct way to compare our method to the exact method is by means of the Kullback divergence. However, this comparison can only be done for small networks. The reason is that the computation of the Kullback divergence requires the computation of the Boltzmann distribution, Eq. 4, which requires exponential time due to the partition function Z. In addition, the exact learning method requires exponential time. The comparison by Kullback divergence on small problems is the subject of Section 4.1.

For networks with a large number of units, we will demonstrate the quality of the linear response method by means of a pattern completion task i.e. the network must be able to generate the rest of a pattern, when part of the pattern is shown. The comparison of pattern completion on larger problems is the subject of Section 4.2.

4.1 Comparison using Kullback divergence

In order to show the performance of the linear response correction, we have compared it with the results obtained with a factorized model and with the exact method.

For the exact method (ex) we have used conjugate gradient. The mean firing rates and correlations are computed using Eqs. 7. For the linear response method without self-coupling term (lr0) we have solved the fixed point Eqs. 33 for $i \neq j$ using least squares and the Levenberg-Marquardt method. The matrix A is given by Eq. 23 with $w_{ii} = 0$. For the linear response method with self-coupling (lr) we obtain the weights and thresholds from Eq. 34 and Eq. 35. This method can be applied when $\det(C) > 0$. When $\det(C)=0$, we have solved the fixed point Eqs. 33 for all i, j using least squares and the Levenberg-Marquardt method. The matrix A is given by Eq. 23 with w_{ii} free parameters.

In the case of the factorized model we assume

$$p_{mf}(\vec{s}) = \prod_i \frac{1}{2}(1 + s_i m_i). \tag{36}$$

The mean firing rates are given by $m_i = \langle s_i \rangle_c$. The four methods are compared by computing the Kullback divergence, using Eq. 5.

In Fig. 2, we present the results for a network of 6 neurons. The number of patterns in the training set is varied from $p = 1$ until $p = 64$. For each p, 5 data sets were randomly generated. Each of the p patterns in the data set is assigned a random probability such that the total probability on the p patterns sums to 1.

The lr method used least squares minimization for $2 \leq p \leq 6$. For the methods lr0 and lr we observed for $2 \leq p \leq 6$ in approximately 10 % of the cases that the fixed point equations could not be solved. This can of course happen because the equations are approximations to the true gradients and therefore do not need to have a fixed point solution. These cases were deleted from the computation of the average Kullbacks in Fig. 2.

We see that the exact method approaches the target distribution ($K = 0$) for very small number of patterns and for $p \rightarrow 2^n$. For $p = 1$, the correlations in the target distribution are absent, and all methods yield Kullback zero. For $p \rightarrow 2^n$ the factorized model approaches the exact model. This is because the target distribution becomes more or less constant over all patterns and correlations are absent in the constant distribution. The most difficult learning tasks are for low and intermediate values of p. The difference between
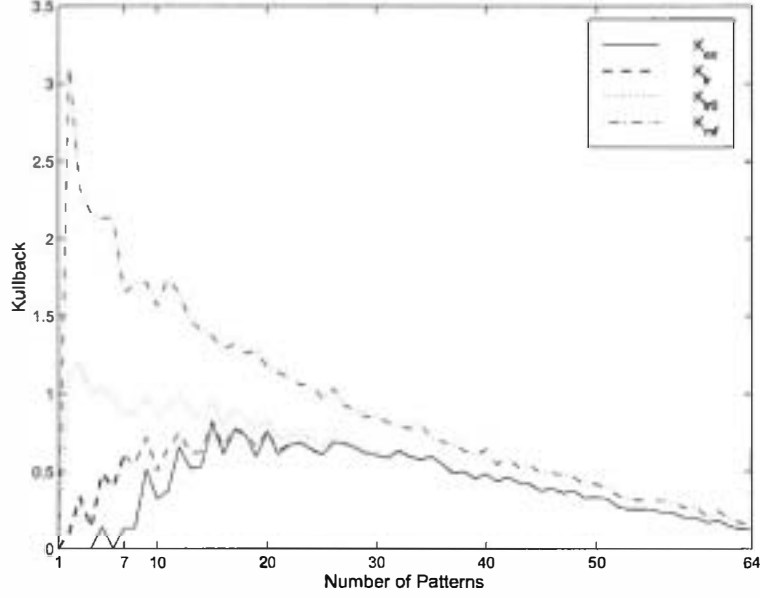
Figure 2: Average Kullback divergence over 5 random training sets as a function of the number of patterns in the trainingset. The network consists of 6 neurons.

K_{mf} and K_{ex} shows that correlations play a significant role. The linear response solutions with and without the self-coupling term give a significant improvement. Linear response with self-coupling term gives the best approximation. In the remaining of the numerical studies we will only consider the linear response method with self-coupling.

We compare the performance of the various methods on networks with 3 to 10 neurons in Fig. 3. For each problem size, training data were randomly generated with $p = 2n$. Each neuron value $s_i^\mu = \pm 1, i = 1, \ldots, n, \mu = 1, \ldots, p$ is generated randomly and independently with equal probability. For each data set we compute $K_{lr} - K_{ex}$ and $K_{mf} - K_{ex}$. In the Figure, we show these values averaged over all data sets, as well as their variances. From the difference between K_{ex} and K_{mf} we see that correlations play an increasingly important role. The linear response approximation is often quite close to the exact result. The quality of the approximation does not deteriorate with increasing problem size.
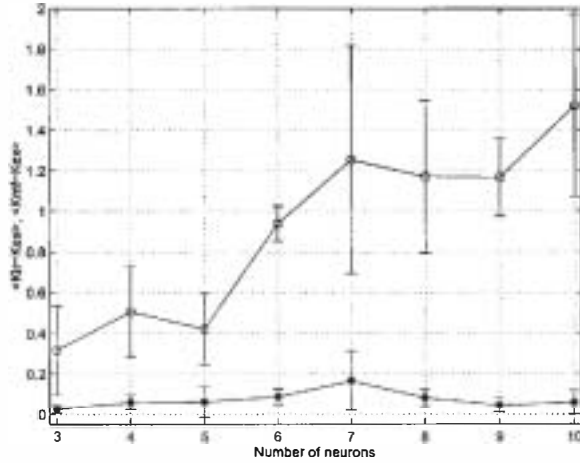

Figure 3: Kullback divergence relative to exact method, for mean field approximation (open circles) and linear response method with self-coupling (closed circles). The number of patterns $p = 2n$. Results are averaged over 4 data sets. The error bars indicate the variance over the data sets.

9

4.2 Comparison on pattern completion

In this subsection, we demonstrate the quality of the linear response method for larger networks. As we mentioned above, this can not be done by comparison of the Kullback divergence. Therefore, we propose to compare the different methods on n pattern completion tasks.

We first train the networks as before as if the problem were a joint probability estimation problem, i.e. with no distinction between 'input' and 'output'. Subsequently, we measure the quality of the different solutions by computing

$$Q = -\frac{1}{np} \sum_{i\mu} \log(p(s_i^\mu | \tilde{s}_i^\mu)), \quad \tilde{s}_i^\mu = (s_1^\mu, \cdots, s_{i-1}^\mu, s_{i+1}^\mu, \cdots, s_n^\mu) \tag{37}$$

The quantity $p(s_i^\mu | \tilde{s}_i^\mu)$ is the conditional probability of finding neuron i in the state s_i^μ, given that the rest of the state is $\tilde{s}_i^\mu$. We can do this for the exact method (for small networks) for the linear response method and for the factorized model. Note, that the computation of Q is fast because it does not require the computation of the partition function.
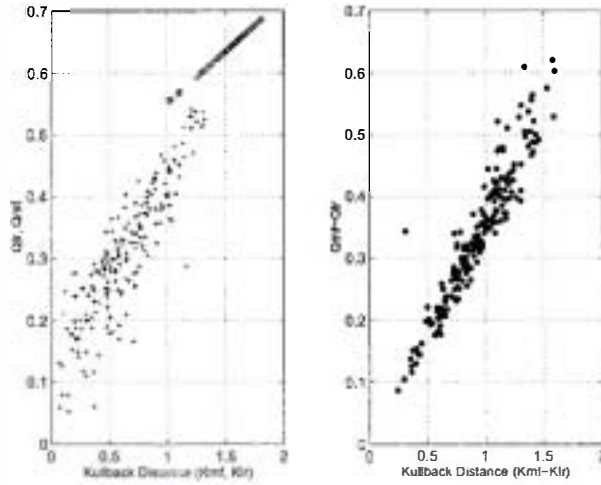

Figure 4: Variation of the pattern completion quality Q with respect to the Kullback divergence K, for 200 data sets on 6 neurons. Each data set consists of 10 patterns. In the left graph, the plus signs represent the linear response method and the open circles represent the factorized model. In the right graph we plot the difference between the two pattern completion qualities $(Q_{mf} - Q_{lr})$ versus the difference of the Kullback divergence $(K_{mf} - K_{lr})$ for the same data sets.

In order to use Q to assess the quality of the various methods, we must establish that low Q implies low Kullback divergence K and vice versa. This is shown in Fig. 4. The left graph shows for the linear response solutions and for the factorized model solutions separately that there is a more or less linear relation between the quality in terms of K and in terms of Q. In the right graph we show for the same data sets the difference in pattern completion quality, $Q_{mf} - Q_{lr}$, versus the difference in Kullback divergence, $K_{mf} - K_{lr}$. From this we see that if one method has a lower Q than another method, we can expect that its Kullback divergence is lower as well.

Thus, one can use the more or less linear relation between Q and K to test the performance of the linear response method for problems with a large number of neurons. In Fig. 5, we show the pattern completion quality for the different methods as a function of the network size. The exact method was only computed up to 10 neurons, because of the time required. (Depending on the stop criterion, the exact method requires approximately 10-30 minutes on a network of 10 neurons on a SPARC 5). We can see that the linear response method is very close to the exact method. The much higher value of the factorized model indicates the obvious fact that correlations play an important role in this task. Note that the mean field method approaches $q = \log 2$ for large n, which is due to the fact that the mean field method assigns $p(s_i^\mu) \approx \frac{1}{2}$ ($m_i \approx 0$) for all i and μ.
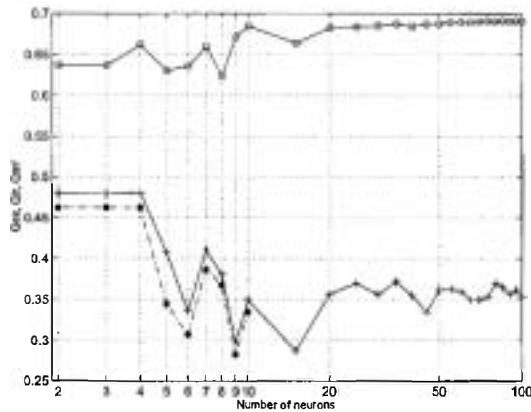
10

Figure 5: Prediction quality for 27 different random problems with different number of neurons. In every problems the number of patterns $p = 2n$. The plus signs represent the linear response correction (Q_{lr}). The open circles represent the factorized model (Q_{mf}). The closed circles represent the exact method (Q_{ex}).

5 Discussion

We have proposed a new efficient method for learning in Boltzmann Machines. The method is generally applicable to networks with or without hidden units. It makes use of the linear response theorem for the computation of the correlations within the mean field framework.

In our numerical experiments we have restricted ourselves to networks without hidden units. We argue that this is sufficient to show the advantage of the method, since the 'free' expectation values are the most time consuming part of the computation.

We have observed numerically that the inclusion of self-coupling is important to get good results. This is probably also true in the presence of hidden units. In that case, a gradient based procedure is required and no closed form solution exists. The presence of self-coupling was motivated from the TAP equations. A full treatment of the linear response correction in this case is the subject of a future publication.

In the presence of hidden units, both the exact method and the linear response method require a gradient descent algorithm. The advantage of our method is that the gradients can be computed in $\mathcal{O}(n^3)$, instead of in $\mathcal{O}(2^n)$, time. The required number of iterations may be somewhat more for the linear response method, because the gradients are only computed approximately.

This brings us to an interesting point, which is the convergence of the gradient descent algorithm in the linear response approximation. Convergence requires the existence of a Lyapunov function. The Kullback divergence is clearly a Lyapunov function for the exact method, but we were not able to find a Lyapunov function for the linear response approximation. In fact, one would like to construct a cost function such that its gradients are equal to the gradients of K in the linear response approximation. Whether such a function exists is unknown to our knowledge.

In addition to probability estimation, Boltzmann Machines have been proposed for combinatoric optimization (Hopfield and Tank, 1985; Durbin and Willshaw, 1987; Yuille and Kosowsky, 1994). For optimization the naive mean field framework can be succesfully applied to combinatoric optimization problems (Yuille et al., 1991; Kosowsky and Yuille, 1994). This method is known as deterministic annealing. Clearly, the situation is different here, since one is mainly concerned with the quality of the solution 'at the end' of the annealing schedule, i.e. when $T \rightarrow 0$. Correlation vanish in this limit in unfrustrated systems but can be quite complex in spin glasses (see for instance (Young, 1983) for numerical results). Whether the linear response correction can improve deterministic annealing is an open question.

As mentioned in the introduction, the naive mean field approach arises as a special case of the variational techniques that have been recently proposed. It should be further investigated whether the linear response correction can be applied in this context as well.

Acknowledgement

11

References

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

Dayan, P., Hinton, G., Neal, R., and Zemel, R. (1995). The Helmholtz Machine. *Neural Computation*, 7:889–904.

Durbin, R. and Willshaw, D. (1987). An analogue approach to the travelling salesman problem using an elastic net method. *Nature*, 326:689–691.

Fischer, K. and Hertz, J. (1991). *Spin glasses*. Cambridge Studies in Magnetism. Cambridge University Press.

Galland, C. (1993). The limitations of deterministic boltzmann machine learning. *Network*, 4:355–380.

Ginzburg, I. and Sompolinsky, H. (1994). Theory of correlations in stochastic neural networks. *Physical Review E*, 50:3171–3191.

Hertz, J., Krogh, A., and Palmer, R. (1991). *Introduction to the theory of neural computation*, volume 1 of *Santa Fe Institute*. Addison-Wesley, Redwood City.

Hinton, G. (1989). Deterministic Boltzmann learning performs steepest descent in weight-space. *Neural Computation*, 1:143–150.

Hopfield, J. and Tank, D. (1985). Neural computation of decision in optimization problems. *Biological Cybernetics*, 52:141–152.

Itzykson, C. and Drouffe, J.-M. (1989). *Statistical Field Theory*. Cambridge monographs on mathematical physics. Cambridge University Press, Cambridge, UK.

Jaakkola, T. and Jordan, M. (1996). Recorsive algorithms for approximating probabilities in graphical models. MIT Computational Cognitive Science Technical Report 9604.

Kappen, H. (1995). Deterministic learning rules for Boltzmann machines. *Neural Networks*, 8:537–548.

Kappen, H. (1997). Stimulus dependent correlations in stochastic networks. *Physical Review E*, 55:5849–5858.

Kosowsky, J. and Yuille, A. (1994). The invisible hand algorithm: solving the assignment problem with statistical physics. *Neural Networks*, 3:477–490.

Kullback, S. (1959). *Information Theory and Statistics*. Wiley, New York.

Onsager, L. (1936). Electric moments of molecules in liquids. *Journal of the American Chemical Society*, 58:1486–1493.

Parisi, G. (1988). *Statistical Field Theory*. Frontiers in Physics. Addison-Wesley.

Peterson, C. and Anderson, J. (1987). A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019.

Plefka, T. (1982). Convergence condition of the TAP equation for the infinite-range Ising spin glass model. *Journal of Physics A*, 24:2173.

Saul, L., Jaakkola, T., and Jordan, M. (1996). Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4:61–76.

Saul, L. and Jordan, M. (1994). Learning in boltzmann trees. *Neural Computation*, 6:1174–1184.

Sinclair, A. (1993). *Algorithms for Random Generation & Counting. A Markov Chain Approach*. Progress in theoretical computer science. Birkhäuser.

Thouless, D., Anderson, P., and Palmer, R. (1977). Solution of 'solvable model of a spin glass'. *Philisophical Magazine*, 35:593–601.

Young, A. (1983). Direct determination of the probability distriubtion for the spin-glass order parameter. *Physical Review Letters*, 51:1206–1209.

Yuille, A., Geiger, D., and Bülthoff, H. (1991). Stereo integration, mean field theory and psychophysics. *Network*, 2:423–442.

Yuille, A. and Kosowsky, J. (1994). Statistical physics algorithms that converge. *Neural Computation*, 6:341–356.

6 Appendix

In this appendix we illustrate the consequences of the linear response method for the simple case of two neurons. This problem was considered numerically in section 2.3.

The general probability distribution in 2 neurons is parametrized by 3 numbers. Consider the symmetric case where $\langle s_1 \rangle = \langle s_2 \rangle$. Then only two parameters are needed, which we choose such that

$$
\begin{aligned}
p(+,+) &= \frac{1}{2}(1+m) - a \\
p(+,-) &= p(-,+) = a \\
p(-,-) &= \frac{1}{2}(1-m) - a.
\end{aligned}
$$

We must require that $0 < a < \frac{1}{2}$ and $2a - 1 < m < 1 - 2a$ to ensure that all probabilities are positive. In this parametrization $\langle s_1 s_2 \rangle = 1 - 4a$ and $\langle s_1 \rangle = \langle s_2 \rangle = m$. The special case of section 2.3 is obtained for $m = a = 0$. The matrix C as defined in section 3.4 is given as

$$
C = \left(\begin{array}{cc} 1 - m^2 & 1 - 4a - m^2 \\ 1 - 4a - m^2 & 1 - m^2 \end{array} \right)
$$

Eq. 34 gives directly

$$
w = \frac{1}{8a} \frac{1 - m^2 - 4a}{1 - m^2 - 2a} \left(\begin{array}{cc} -1 + \frac{4a}{1-m^2} & 1 \\ 1 & -1 + \frac{4a}{1-m^2} \end{array} \right)
$$

and the thresholds are computed using Eq. 35. Note, that the diagonal weights play an important role in the computation of the thresholds.

One can also compute the optimal weigths and thresholds using the exact method. Setting $\Delta w_{ij} = 0$ and $\Delta \theta_i = 0$ in Eq. 6 we obtain

$$
\begin{aligned}
w_{12} &= \log \left(\frac{(1 - 2a)^2 - m^2}{4a^2} \right) \\
\theta_i &= \frac{1}{2} \tanh^{-1} \left(\frac{m}{1 - 2a} \right)
\end{aligned}
$$

The differences are illustrated for $m = 0.1$ and $m = 0.5$ for all allowed values of a in Fig. 6

Note that the linear response approximation is very good in those instances where the optimal weights are small. For larger weights the difference between the two methods increases. However, in the regions of large weights the difference has an exponentially vanishing infuence on the value of the the probability distribution and thus on the quality of the solution as measured by the Kullback diverence. The same statements are more or less true for the thresholds.
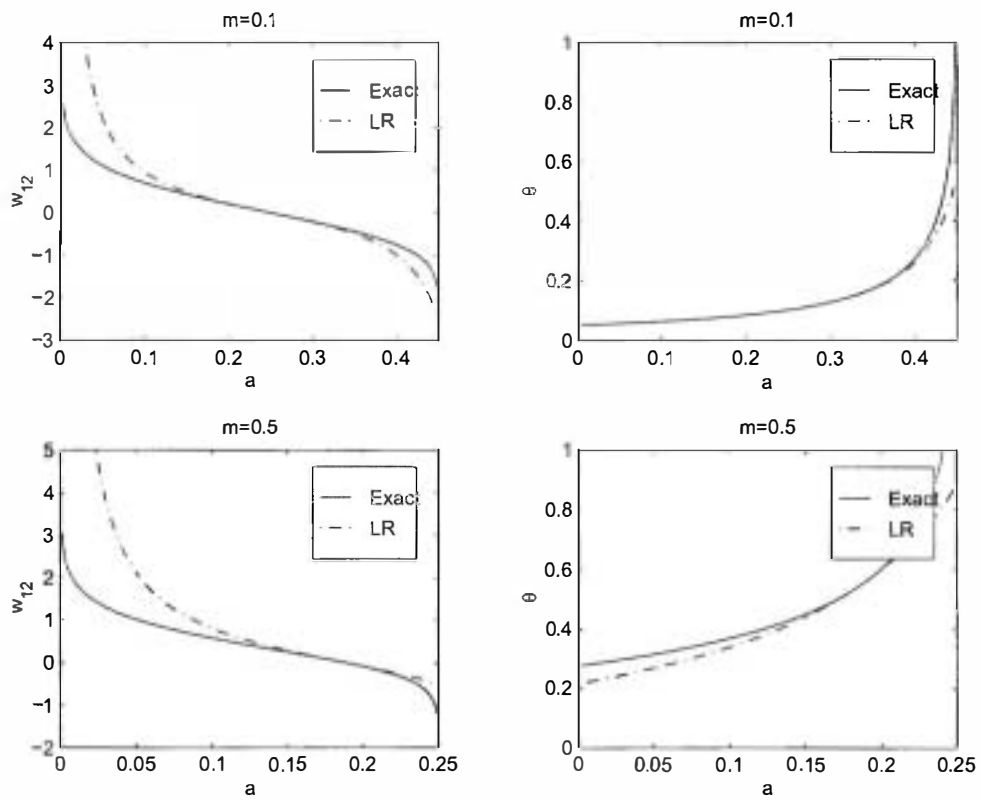
Figure 6: Examples of lateral connection and threshold(s) obtained by exact method and linear response method (LR) for a network of two neurons with $m = 0.1$ and $m = 0.5$