Gatsby Computational Neuroscience Unit
University College London
http://www.gatsby.ucl.ac.uk

17 Queen Square, London
WC1N 3AR, United Kingdom
+44 20 7679 1176

Funded in part by the Gatsby Charitable Foundation.

June 5, 2001

GCNU TR 2001–002

A New Learning Algorithm for Mean Field Boltzmann Machines

Max Welling **G.E. Hinton**

Gatsby Unit

Abstract

We present a new learning algorithm for Mean Field Boltzmann Machines based on the *contrastive divergence* optimization criterion. In addition to *minimizing* the divergence between the data distribution and the equilibrium distribution that the network believes in, we *maximize* the divergence between one-step reconstructions of the data and the equilibrium distribution. This eliminates the need to estimate equilibrium statistics, so we do not need to approximate the multimodal probablility distribution of the free network with the unimodal mean field distribution. We test the learning algorithm on the classification of digits.

A New Learning Algorithm for Mean Field Boltzmann Machines

Max Welling **G.E. Hinton**

Gatsby Unit

1 Boltzmann Machines

The stochastic Boltzmann machine (BM) is a probabilistic neural network of symmetrically connected binary units taking values $\{0, 1\}$ (Ackley, Hinton & Sejnowski, 1985). The variant used for unsupervised learning consists of a set of visible units $\mathbf{v}$, which are clamped to the data $\mathbf{v}_{1:N}$, and a set of hidden units $\mathbf{h}$, which allow the modelling of higher order statistics of the data. We may define the energy E of the system at a particular state $\{\mathbf{v}, \mathbf{h}\}$ to be,

$$E(\mathbf{v}, \mathbf{h}) = -(\frac{1}{2}\mathbf{v}^T\mathbf{V}\mathbf{v} + \frac{1}{2}\mathbf{h}^T\mathbf{W}\mathbf{h} + \mathbf{v}^T\mathbf{J}\mathbf{h}) \tag{1}$$

where we have added one unit with value always 1, whose weights to all other units represent the biases. In terms of the energy, the probability distribution of the system can be written as,

$$P(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{Z} \tag{2}$$

where Z denotes the normalization constant[1] (or"partition function" in physics),

$$Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{3}$$

A natural measure to minimize during learning is the Kullback-Liebler divergence between the data distribution $P_0(\mathbf{v})$ and the model distribution for the data $P_\infty(\mathbf{v})$. The notation will become apparent later, but can be understood by imagining running a Markov chain, starting at the data distribution ($t = 0$) until equilibrium ($t = \infty$). This KL-divergence can be rewritten as follows,

$$KL[P_0(\mathbf{v})||P_\infty(\mathbf{v})] = F_0 - F_\infty \tag{4}$$

where F_0 denotes the free energy of the system when we clamp the data distribution on the visible units, while $F_\infty = -\log(Z)$ denotes the free energy of the system when we use the model distribution (i.e. at equilibrium). The free energy can be conveniently expressed in terms of the energy and entropy of the system as follows,

$$F_0 = \langle E\rangle_0 - S_0 \tag{5}$$

$$F_\infty = \langle E\rangle_\infty - S_\infty \tag{6}$$

where $\langle . \rangle_0$ denotes taking the average with respect to the joint $P(\mathbf{h}|\mathbf{v})P_0(\mathbf{v})$ while S_0 is the entropy of that distribution, and $\langle . \rangle_\infty$ denotes taking the average with respect to the equilibrium distribution $P(\mathbf{v}, \mathbf{h})$ while S_∞ is its entropy.

[1] We will set the "temperature" of the system to $T = 1$.

Learning in the BM consists of adjusting the weights such that the probability of the data under the model increases. It is straightforward to take derivatives with respect to the weights and apply gradient descent learning,

$$\Delta \mathbf{W} \propto \langle \mathbf{h}\mathbf{h}^T \rangle_0 - \langle \mathbf{h}\mathbf{h}^T \rangle_\infty \tag{7}$$

$$\Delta \mathbf{V} \propto \langle \mathbf{v}\mathbf{v}^T \rangle_0 - \langle \mathbf{v}\mathbf{v}^T \rangle_\infty \tag{8}$$

$$\Delta \mathbf{J} \propto \langle \mathbf{v}\mathbf{h}^T \rangle_0 - \langle \mathbf{v}\mathbf{h}^T \rangle_\infty \tag{9}$$

In practice, we substitute the *empirical* distribution $\hat{P}_0(\mathbf{v}; \mathbf{v}_{1:N})$ for $P_0(\mathbf{v})$, and adjust the learning rules accordingly.

Although appealing in theory, these learning rules are not particularly practical, since the number of states we need to sum over in order to compute the averages scales exponentially with the number of units. One solution is to apply Gibbs sampling, which samples one unit (or set of units) according to its posterior distribution, given the current values of all the other units[2]. This strategy can also become computationally demanding since at every iteration of learning, Gibbs sampling must be performed for every datapoint in the "wake" phase (with the visible units clamped) and once more in the sleep phase. Moreover, at every run, we have to wait until the Markov chain has reached equilibrium, and many independent samples are produced.

2 Mean Field Boltzmann Machines

An alternative to the slow Gibbs sampling is to approximate the averages using a fully factorized, mean field (MF) distribution (Peterson & Anderson, 1987).

$$Q(\mathbf{s}) = \prod_i m_i^{s_i} \, (1 - m_i)^{1 - s_i} \tag{10}$$

where the product is taken over all units s_i. This MF distribution has one free parameter per unit, m_i, describing the probability that this unit will be "on". These parameters will be chosen such that the approximating distribution, Q, is as close as possible to the true distribution, P, in the sense of the KL-divergence measure,

$$\mathbf{m}^* = \mathbf{argmin}_{\mathbf{m}} \, KL\left[Q(\mathbf{s}) || P(\mathbf{s})\right] \tag{11}$$

This approximation is applied to $P(\mathbf{h}, \mathbf{v} = \mathbf{v}_n)$ for all datapoints separately in the wake phase, and once more to $P(\mathbf{h}, \mathbf{v})$ in the sleep phase. We have therefore replaced the stochastic Gibbs sampling by a deterministic minimization, which is much faster albeit not as accurate[3]. Using the emprical distribution $\hat{P}_0$, and equating the derivatives with respect to $\mathbf{m}_n$ to zero results in the simple fixed point equations for the wake phase,

$$\mathbf{m}_{h,n} = \sigma(\mathbf{W}\mathbf{m}_{h,n} + \mathbf{J}^T \mathbf{v}_n) \tag{12}$$

where σ denotes the sigmoid function. Similarly in the sleep phase we have,

$$\mathbf{m}_h = \sigma(\mathbf{W}\mathbf{m}_h + \mathbf{J}^T \mathbf{m}_v) \tag{13}$$

$$\mathbf{m}_v = \sigma(\mathbf{V}\mathbf{m}_v + \mathbf{J}\mathbf{m}_h) \tag{14}$$

[2] Also known as Glauber-dynamics.

[3] Note that due to the $\log(Z)$ term, this variational technique does not result in an upper bound to the total free energy.

The solutions of these fixed point equations[4] may then be used to approximate the correlations in the learning rule,

$$\Delta \mathbf{W} \propto \left[\frac{1}{N}\sum_{n=1:N} \mathbf{m}_{h,n}\mathbf{m}_{h,n}^T - \mathbf{m}_h\mathbf{m}_h^T\right] \tag{15}$$

$$\Delta \mathbf{V} \propto \left[\frac{1}{N}\sum_{n=1:N} \mathbf{v}_{n}\mathbf{v}_{n}^T - \mathbf{m}_v\mathbf{m}_v^T\right] \tag{16}$$

$$\Delta \mathbf{J} \propto \left[\frac{1}{N}\sum_{n=1:N} \mathbf{v}_{n}\mathbf{m}_{h,n}^T - \mathbf{m}_v\mathbf{m}_h^T\right] \tag{17}$$

It was shown by Hinton (1989) that these learning rules perform gradient descent on the cost funtion,

$$F^{MF} = F_0^{MF} - F_\infty^{MF} \tag{18}$$

(apart from rare discontinuities) if we choose the stepsizes small enough. Mean field free energies are defined as,

$$F_Q^{MF} = \langle E \rangle_Q - S(Q) \tag{19}$$

Extensions of MFBM learning rules have been put forward, such as linear response corrections (Kappen & Rodriquez, 1998) and TAP corrections (Galland, 1993). These extensions improve on the simple independence assumption of the MF distribution, and capture some of the higher order statistics in estimating the correlations. However, they fail to adress the main drawback of the MFBM, which is that in the sleep phase it uses a unimodal distribution to approximate a distribution with many modes, since there is no data clamped on any of the units. Instead of trying to use better, multimodal approximating distributions in the sleep phase (which will be very difficult), it may prove more fruitful to change the learning rule, such that we only have to deal with inferring posterior distributions, conditioned on data (or reconstructions of data). This is precisely what contrastive divergence learning accomplishes.

3 Contrastive Divergence Learning

In Contrastive Divergence (CD) learning (Hinton, 2000), we replace the correlations computed in the sleep phase of BM learning with the correlations conditioned on one-step reconstructions of the data.

We start by recalling that the KL-divergence between the data distribution and the model distribution can be written as a difference between two free energies,

$$KL[P_0(\mathbf{v})||P_\infty(\mathbf{v})] = F_0 - F_\infty \geq 0 \tag{20}$$

To get samples from the equilibrium distribution we imagine running a Markov chain, starting at the data distribution P_0 and eventually reaching equilibrium at $t = \infty$. With hidden units, we would first sample the hidden units, given the data, then sample reconstructions of the data, given the sampled hidden units, etc. It is not hard to show that at every step of Gibbs sampling the free energy has decreased,

$$F_0 \geq F_i \geq F_\infty \quad \forall i \tag{21}$$

Moreover, it must therefore be true that if the free energy hasn't changed after i steps of Gibbs sampling (for any i), either $P_0 = P_\infty$ or the Markov chain does not mix (which we must

[4] To avoid oscillations one may need to "damp" the fixed point equations according to $\mathbf{m}_{\mathbf{new}} = \alpha\mathbf{m}_{\mathbf{old}} + (1-\alpha)\sigma(.)$, $\alpha \in [0, 1)$

therefore avoid). The above suggests that we could use the following contrastive free energy,

$$\begin{aligned} CF_i = F_0 - F_i = & + \; KL\left[P_0(\mathbf{v},\mathbf{h})||P(\mathbf{v},\mathbf{h})\right] \\ & - \; KL\left[P_i(\mathbf{v},\mathbf{h})||P(\mathbf{v},\mathbf{h})\right] \\ & \geq \; 0 \end{aligned} \tag{22}$$

as an objective to minimize[5]. The big advantage is that we do not have to wait for the chain to reach equilibrium. Also, at equilibrium, the distribution has forgotten everything about the data and is therefore expected to be highly multimodal, which is hard to model. In the following we will set $i = 1$.

Learning proceeds by taking derivatives with respect to the parameters and performing gradient descent on CF. The derivative is given by,

$$\frac{\partial CF}{\partial \theta} = \left\langle \frac{\partial E}{\partial \theta} \right\rangle_0 - \left\langle \frac{\partial E}{\partial \theta} \right\rangle_1 - \frac{\partial F_1}{\partial P_1}\frac{\partial P_1}{\partial \theta} \tag{23}$$

The last term is hard to evaluate, but small compared with the other two. Hinton (2000) shows that this ackward term can be safely ignored. For the BM, this results in the following learning rules,

$$\Delta \mathbf{W} \;\propto\; \langle \mathbf{h}\mathbf{h}^T \rangle_0 - \langle \mathbf{h}\mathbf{h}^T \rangle_1 \tag{24}$$

$$\Delta \mathbf{V} \;\propto\; \langle \mathbf{v}\mathbf{v}^T \rangle_0 - \langle \mathbf{v}\mathbf{v}^T \rangle_1 \tag{25}$$

$$\Delta \mathbf{J} \;\propto\; \langle \mathbf{v}\mathbf{h}^T \rangle_0 - \langle \mathbf{v}\mathbf{h}^T \rangle_1 \tag{26}$$

Although some progress has been been made, the new learning objective does not avoid having to run full Gibbs runs on the hidden units given the data, and vice versa for every data point. There is a special type of BM, the restricted BM (RBM), in which there are no hidden-to-hidden or visible-to-visible connections. In a RBM, the hidden units are therefore independent given the visible units and vice versa, so the above learning rules are highly efficient. Unfortunately, for fully-connected BMs this is not the case, and we have to make approximations to make further progress.

4 Contrastive Divergence Mean Field Learning

In this section we formulate the deterministic mean field variant of the contrastive divergence learning strategy.

We will first consider one datapoint $\mathbf{v}_n$, and later generalize to N datapoints. The empirical data distribution for this one point is simply $\hat{P}_{0,n} = \delta(\mathbf{v} - \mathbf{v}_n)$. Next, recall the definition of the mean field free energy,

$$F^{MF} = \langle E \rangle_Q - S(Q) \tag{27}$$

where Q represents the fully factorized mean field distribution (10). It is not difficult to see that the MF free energy at the data distribution is always larger than at equilibrium,

$$F_0^{MF} \geq F_\infty^{MF} \tag{28}$$

We can imagine a deterministic version of the Markov chain from the previous section, where we start at the data distribution and perform *coordinate descent* (which alternates between the coordinates $\mathbf{m}_h$ and $\mathbf{m}_v$), until we reach the minimum of the free energy, F_∞^{MF} (for the time

[5]We could omit the argument $\mathbf{h}$ in (22) since $P_0(\mathbf{v},\mathbf{h})$, $P_i(\mathbf{v},\mathbf{h})$ and $P(\mathbf{v},\mathbf{h})$ have the same posterior $P(\mathbf{h}|\mathbf{v})$, which therefore drops out of the KL-divergence. This is the convention used by Hinton (2000). We include dependence on $\mathbf{h}$, since it is then symmetric with the mean field expression (30)

being we will assume that there are no intermediate local minima). Most importantly, this has the property that at intermediate stages of the minimization the following holds,

$$F_0^{MF} \geq F_i^{MF} \geq F_\infty^{MF} \tag{29}$$

By analogy to the derivation for the stochastic BM, we now define the contrastive free energy,

$$\begin{aligned} CF_i^{MF} = F_0^{MF} - F_i^{MF} = & + \; KL[Q_0(\mathbf{v},\mathbf{h})||P(\mathbf{v},\mathbf{h})] \\ & - \; KL[Q_i(\mathbf{v},\mathbf{h})||P(\mathbf{v},\mathbf{h})] \\ & \geq \; 0 \end{aligned} \tag{30}$$

which is always positive and vanishes when $F_0^{MF} = F_\infty^{MF}$ (or at some intermediate local minimum). This is therefore a sensible cost function to minimize. The most important advantage is that we use Q to approximate P only in the first stages of Gibbs sampling where P is expected to be unimodal (given enough data), in contrast to the equilibrium distribution which has forgotten all about the particular data vector used to initialize the Markov chain.

From (30) we see that the mean field minimization, which replaces the Gibbs sampling, tries to match $Q(\mathbf{v},\mathbf{h})$ with $P(\mathbf{v},\mathbf{h})$. During learning the parameters are adjusted, such that the initial distribution, $Q(\mathbf{h}|\mathbf{v})P_0(\mathbf{v})$ gets closer to the true (equilibrium) distribution $P(\mathbf{h},\mathbf{v})$. When no more improvement is possible, which does not necessarily imply that Q_0 is a perfect match to P, training stops.

In practice, a local minimum may obstruct the path from F_0^{MF} to F_∞^{MF}. But notice that this is not a problem at the intermediate stages of learning, since we are guaranteed that the initial MF distribution has higher free energy than the intermediate MF distribution, so adjusting the weights will improve CF^{MF}, provided the ackward term in (23) is indeed negligible. If all derivatives vanish, while F_0^{MF} is not equal to the lowest free energy state we have landed in a local minimum. Annealing schedules may be used to soften this problem.

It is important to stress that the first minimization over $\mathbf{m}_h$, which is randomly initialized, has to be run until convergence in order to compute the initial free energy F_0^{MF}. The next minimization over $\mathbf{m}_v$, given $\mathbf{m}_h$ must be initialized at the data, but does *not* need to be run until convergence. A few steps in the direction of the negative gradient are guaranteed to lower the free energy, which is all we need. Equivalently, the subsequent minimation over $\mathbf{m}_h$ given the reconstructions of the data $\mathbf{m}_v$, is initialized at the previous attained value of $\mathbf{m}_h$ and may be run for only a few steps in the direction of the negative gradient.

The above is easily generalized to N datapoints. We consider the empirical data distribution $\hat{P}_0 = \frac{1}{N}\sum_{n=1:N}\delta(\mathbf{v}-\mathbf{v}_n)$. We now have N MF distributions Q_n, with separate parameters $\{\mathbf{m}_{h,n},\mathbf{m}_{v,n}\}$ to minimize. But since all data are independent, all minimizations would ideally end up in the same global minimum F_∞^{MF} (ignoring local minima temporarily). We therefore still have,

$$\frac{1}{N}\sum_{n=1:N} F_{0,n}^{MF} \geq \frac{1}{N}\sum_{n=1:N} F_{i,n}^{MF} \geq F_\infty^{MF} \tag{31}$$

which allows us to define the contrastive free energy,

$$CF_i^{MF} = \frac{1}{N}\sum_{n=1:N}\left(F_{0,n}^{MF} - F_{i,n}^{MF}\right) \geq 0 \tag{32}$$

Again, local minima may obstruct the paths to the global minimum. But by the same argument as above, during learning we are always guaranteed to improve CF^{MF}, since the intermediate free energy is always lower than the initial one. However, we are not protected against landing in a local minimum, where all derivatives vanish and learning stops.

To find the learning rules we compute the following derivatives (setting $i=1$),

$$\frac{\partial CF^{MF}}{\partial\theta} = \left\langle\frac{\partial E}{\partial\theta}\right\rangle_{Q_0} - \left\langle\frac{\partial E}{\partial\theta}\right\rangle_{Q_1} - \frac{\partial F_1}{\partial Q_1}\frac{\partial Q_1}{\partial\theta} \tag{33}$$

The last term is again small compared to the others and will be ignored. This results in the simple learning rules,

$$\Delta \mathbf{W} \propto \frac{1}{N} \sum_{n=1:N} \left(\mathbf{m}_{h,n} \mathbf{m}_{h,n}^T - \tilde{\mathbf{m}}_{h,n} \tilde{\mathbf{m}}_{h,n}^T \right) \quad (34)$$

$$\Delta \mathbf{V} \propto \frac{1}{N} \sum_{n=1:N} \left(\mathbf{v}_n \mathbf{v}_n^T - \mathbf{m}_{v,n} \mathbf{m}_{v,n}^T \right) \quad (35)$$

$$\Delta \mathbf{J} \propto \frac{1}{N} \sum_{n=1:N} \left(\mathbf{v}_n \mathbf{m}_{h,n}^T - \mathbf{m}_{v,n} \tilde{\mathbf{m}}_{h,n}^T \right) \quad (36)$$

where $\mathbf{v}_n$ is a data vector, $\mathbf{m}_{h,n}$ are the parameters of $Q(\mathbf{v} = \mathbf{v}_n, \mathbf{h})$ after optimization, $\mathbf{m}_{v,n}$ are the parameters of $Q(\mathbf{v}, \mathbf{h} = \mathbf{m}_{h,n})$ after optimization, and $\tilde{\mathbf{m}}_{h,n}$ are the parameters of $Q(\mathbf{h}, \mathbf{v} = \mathbf{m}_{v,n})$ after optimization. In practice one may want to perform updates on mini-batches when confronted with large redundant datasets. The fixed point equations for the minimization of the free energy are computed by taking derivatives with respect to parameters $\{\mathbf{m}_{v,n}, \mathbf{m}_{h,n}\}$ and equating to zero,

$$\mathbf{m}_{h,n} = \sigma(\mathbf{W}\mathbf{m}_{h,n} + \mathbf{J}^T \mathbf{v}_n) \quad (37)$$

$$\mathbf{m}_{v,n} = \sigma(\mathbf{V}\mathbf{m}_{v,n} + \mathbf{J}\mathbf{m}_{h,n}) \quad (38)$$

$$\tilde{\mathbf{m}}_{h,n} = \sigma(\mathbf{W}\tilde{\mathbf{m}}_{h,n} + \mathbf{J}^T \mathbf{m}_{v,n}) \quad (39)$$

These equations must be run *sequentialy*. The last argument in the sigmoid is fixed and acts as a bias term. Also, damping may be necessary to avoid oscillations.

In rare cases discontinuities can occur in the mean field free energy of a MFBM as a function of the parameters. In those cases, small steps in the negative gradient direction are not guaranteed to lower the objective function.

5 Supervised Learning

The above exposition has focussed on unsupervised learning, but is by no means limited to it. For supervised learning, the visible units are divided into input units $\mathbf{i}$ and output units $\mathbf{o}$. The simplest way in which to extend the general framework is to reconstruct both inputs and outputs to the network, which would not require any change in the learning algorithm (just put $\mathbf{v} = \{\mathbf{i}, \mathbf{o}\}$). When the network is queried with a new input $\mathbf{i}$, it then has to minimize the free energy with respect to hidden units and the output units $\{\mathbf{m}, \mathbf{o}\}$ jointly. The values of the output units are then returned. This is a case of pattern completion.

The downside of this approach is that the algorithm has to spend resources on modelling the input distribution, while we are really only interested in the output distribution given the input. As an alternative, one may therefore decide to only reconstruct the outputs, i.e. the hidden units and output units are free to be updated, but the inputs stay clamped to the data, throughout the one-step minimization of the free energy.

6 Extensions

The MF approximation is limited in the sense that it treats all units as independent during "inference". Extensions to the MFBM have been proposed in the literature to include correlation between the units, using linear response theory (Kappen, 1998) or TAP corrections (Galland, 1993). These improvements can be readily applied to the framework presented in this paper, though it is not yet clear how much they will help.

Table 1: Confusion matrix for digit classification task. Each row shows the classification result for a particular digit out of 400 test cases. For instance, 5 out of 400 "8s" from the test set were classified as a "2s". The last column shows the total number of misclassifications for each digit, while the overall classification error can be found in the lower right corner (99), corresponding to 2.5%.

	1	2	3	4	5	6	7	8	9	0	
1	396	0	0	1	1	1	0	1	0	0	4
2	0	387	2	1	0	1	3	4	0	2	13
3	0	2	390	0	5	0	0	1	1	1	10
4	0	2	0	389	0	2	0	3	4	0	11
5	1	0	7	0	389	0	0	2	0	1	11
6	0	2	0	1	2	392	0	1	0	2	8
7	0	1	1	0	0	0	394	1	3	0	6
8	3	5	2	2	1	2	0	381	3	1	19
9	0	0	0	4	1	0	6	3	386	0	14
0	0	1	0	0	0	1	0	1	0	397	3
ERROR											99

The MFBM does not hinge on a probabilistic interpretation and can be regarded as a deterministic neural net. This idea was taken one step further in (Movellan, 1991) where the entropy term in the free energy was defined for a general (bounded, monotonic, differentiable) nonlinearity,

$$S = \sum_i \int_a^{m_i} f_i^{-1}(m_i) dm_i \tag{40}$$

where $a = f(0)$. Chosing $f(.)$ to be a sigmoid would give the usual expression for the entropy. Interestingly, the learning algorithm can still be used with this more general nonlinearity. The CD learning algorithm can be directly applied to this case as well.

Finally, we are interested in extending the framework to hybrid deterministic and stochastic learning. The restricted BM (RBM) is an example where sampling is fast and exact, but no connections between hidden units or between visible units are allowed (Hinton, 2000). Intermediate types of BM are possible where intra visible connections, or intra hidden units are allowed. Using a combination of Gibbs sampling and MF updates may be a fruitful way to proceed.

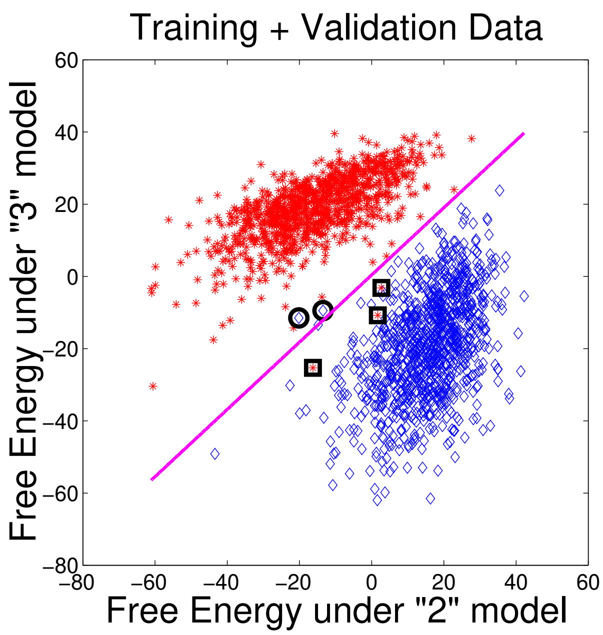
7 Experiments

In the experiments described below we have used 16×16 real valued digits from the "br" set on the CEDAR cdrom # 1. There are 11000 digits available equally divided into 10 classes. The first 7000 were used for training, while we cycled through the last 4000, using 3000 as a validation set and testing on the remaining 1000 digits. The final test-error was averaged over the 4 test-runs. All digits were separately scaled (linearly) between 0 and 1, before presentation to the algorithm.

7.1 Unsupervised Learning of Digit Models

Separate models were trained[6] for each digit, using 700 training examples. Each model was a fully connected MFBM consisting of 50 hidden units. We performed 10 weight updates for

[6] Although the classification task is clearly supervised in the sense that the class labels are assumed known, the digit models are trained to model their input distribution (as oppsosed to an input-output mapping). Hence the



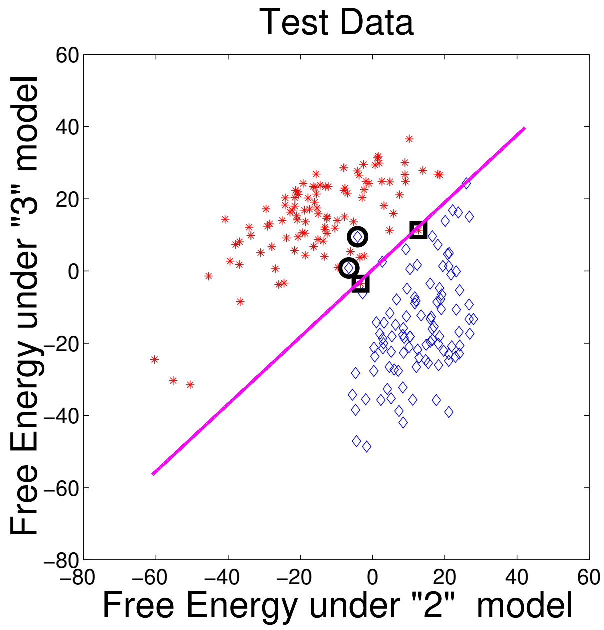


Figure 1: Illustrative example of pairwise classification task on "2" versus "3". For all data, the free energy F_0^{MF} is computed on both models. The values act as features and are plotted as points in a two dimensional space ("$\diamond$" for "3", "$*$" for "2"). We then fit a logistic regression model on the training and validation data (top), which is used for classification on the test data (bottom). In this example, 2 digits "3" are classified as "2" (circles) and 2 digits "2" are classified as "3" (squares) on the test data.

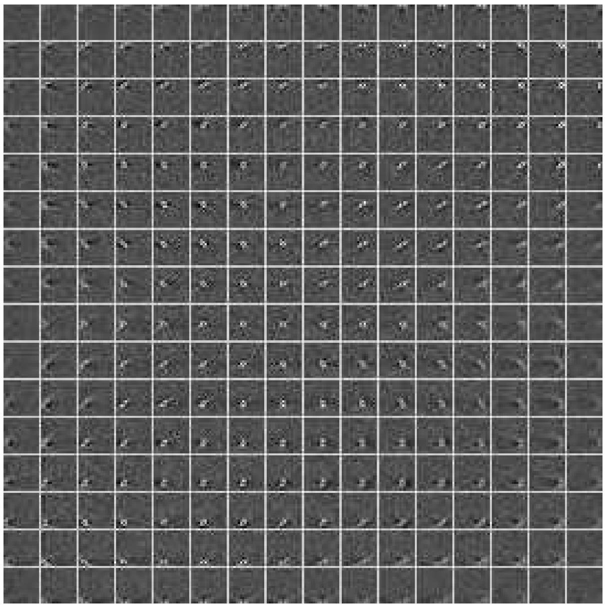


Figure 2: All visible to visible connections for the digit "8". Every patch corresponds to the visible weights for one visible unit at the corresponding location in the image (i.e. the top left patch corresponds to the top left pixel). The visible weights decorrelate the image (using only first and second order statistics). The solution roughly corresponds to ZCA-filtering (see below)

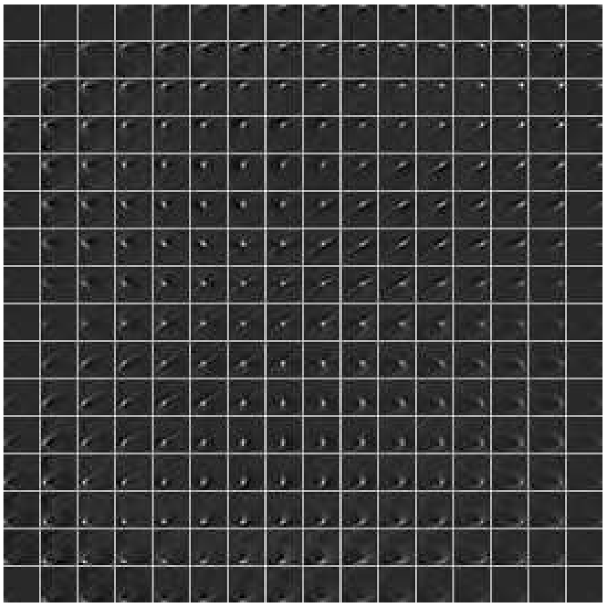


Figure 3: Basis Functions of the ZCA whitening filter, which scales all eigenvalues of the data covariance to one.

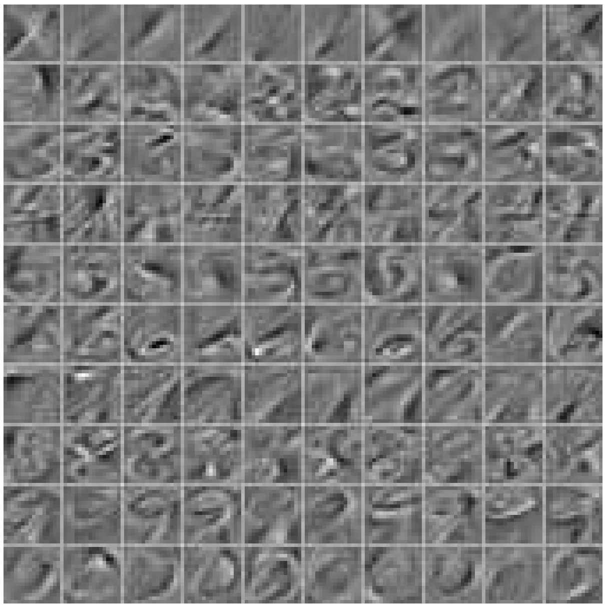


Figure 4: Typical features found by the algorithm on the unsupervised learning task. Every row depicts the weights from 10 randomly chosen hidden units to all visible units for one particular digit-model. The features are global and edge-like. They can be interpreted as small deformations of a digit.

each mini-batch of 100 data vectors, and cycled through those batches 200 times. The updates included a small weight-decay term and a momentum term.

When training was completed, we computed the free energy F_0^{MF} for all data on all models (including validation and test data). Since we do not compute the term $F_\infty^{MF} = -\log(Z)$ (which is much harder), we have no direct access to the log-likelihood. Instead, we fit a multinomial logistic regression model to the training data *plus* the validation data, using the 10 free energies F_0^{MF} for each model as "features". The prediction of this logistic regression model on the test data is finally compared with ground truth, from which a confusion matrix is calculated (table 1). As an example we show in figure (1) the decision boundary computed with binomial logistic regression on the free energies of the digit "2" versus the digit "3".

The total averaged classification error is 2.5% on this data set, which is a significant improvement over simple classifiers such as a 1-nearest-neighbour (5.5%) and multinomial logistic regression (6.4%).

Figure (2) shows the visible-to-visible weights (see figure caption for explanation) for the digit "8". By comparison with the basis functions of a ZCA-filter (figure (3)), we may interpret the visible layer as a whitening filter, removing first and second order statistics from the data. The higher order statistics are modelled by the hidden units, whose "projective fields" (hidden-to-visible weights) are depicted in figure (4). These features contain edge-like elements and are rather global. From a generative perspective, they can be interpreted as small deformations of one digit into another, just like edges are generators of small translations. From a recognition perspective, these features are sensitive to the boundaries of a digit (or parts of a digit).

name "unsupervised".

Figure 5: Features (weights from one hidden to all input units) for supervised learning task. From left to right the data consisted of the following two digits: $\{(3,8);(7,9);(0,1);(6,9)\}$. Notice that the features are discriminative.

7.2 Supervised Learning

To test whether the new algorithm can be used in supervised learning settings, we trained models on pairs of digits, providing it with the true class label. Learning involved the reconstruction of this one bit of information, while the input pattern (image of the digit) was clamped to the visible units at all times. Since we are not reconstructing the original image, no resources are spent on modelling the input distribution. As expected, the algorithm focusses on discovering features which are good for *discrimination* (as opposed to representation). In figure (5) we see 4 examples of features (weights from one hidden unit to all input units) which focus on discriminating two digit classes. For instance, the first feature will be “on” in case of a 3 and “off” in case of an 8. However, the supervised version of the MFBM does not significantly outperform standard logistic regression on pairwise discrimination.

8 Discussion

In this paper we have shown that efficient *contrastive divergence learning* is not limited to structures like the RBM, where the hidden units are independent given the visible units and vice versa. Although exact sampling is no longer feasible in more general structures, approximate mean field methods can be employed instead. The resulting learning rules are analoguous to the standard MFBM learning rules. The sleep phase has however been replaced with a “one-step-reconstruction” phase, for which the unimodal mean field approximation is expected to be much more appropriate.

The new learning algorithm can be interpreted as a deterministic neural network architecture, without making reference to the underlying stochastic version. However, it is precisely this relationship which highlights the limitations of the MFBM. For instance, the deterministic algorithm assigns real values between 0 and 1 to the units, while the stochastic version uses binary values. This implies that it can transmit much more information between units than just 1 bit per unit, a potential danger for overfitting. Instead of passing around real values, one could sample from the binary hidden units, given the mean field parameters, and use that for reconstructing the data. This limits the information that can be conveyed by the hidden states.

Another limitation is the independence assumption of the units when approximated by the mean field distribution. More involved approximate variational distributions could alleviate this problem at least partially.

The main purpose of this paper was to show that the new learning algorithm works, and can be used for classification problems. A more thorough comparison on well documented datasets (like MNIST) has yet to be performed to assess the real merit of mean field CD learning.

Acknowledgements

This research was funded by the Gatsby Foundation. Thanks to David MacKay, Peter Dayan, Zoubin Ghahramani, Simon Osindero, Andy Brown and Yee-Why Teh and other members of the Gatsby Unit for interesting discussions.

References

Ackley, D., Hinton, G.E. & Sejnowski, T. (1985). A Learning Algorithm for Boltzmann Machines. *Cognitive Science*, *9*, 147–169.

Galland, C. (1993). The Limitations of Deterministic Boltzmann Machine Learning. *Network*, *4*, 355–380.

Hinton, G.E. (1989). Deterministic Boltzmann Learning Performs Steepest Descent in Weight-Space. *Neural Computation*, *1*, 143–150.

Hinton, G.E. (2000). *Training Products of Experts by Minimizing Contrastive Divergence* (Technical Report GCNU TR 2000-004). Gatsby Computational Neuroscience unit, London, UK.

Kappen, H.J. & Rodriquez, F.B. (1998). Efficient Learning in Boltzmann Machines using Linear Response Theory. *Neural Computation*, *10*, 1137–1156.

Movellan, J.R. (1991). Contrastive Hebbian Learning in the Continuous Hopfield Model. *Connectionist Models, Proceedings of the 1990 Summer School* (pp. 10–17). San Mateo: Morgan Kaufmann.

Peterson, C.& Anderson, J. (1987). A Mean Field Theory Learning Algorithm for Neural Networks. *Complex Systems*, *1*, 995–1019.