Approximate Contrastive Free Energies for Learning in Undirected Graphical Models

Max Welling
Gatsby Computational Neuroscience Unit
University College London
17 Queen Square
London WC1N 3AR, UK
welling@gatsby.ucl.ac.uk

Geoffrey E. Hinton
Gatsby Computational Neuroscience Unit
University College London
17 Queen Square
London WC1N 3AR, UK
hinton@gatsby.ucl.ac.uk

Abstract

We present a novel class of learning algorithms for undirected graphical models, based on the *contrastive free energy* (**CF**). In particular we study the naive mean field, TAP and Bethe approximations to the contrastive free energy. The main advantage of **CF**learning is the fact that it eliminates the need to infer equilibrium statistics for which mean field type approximations are particularly unsuitable. Instead, learning decreases the distance between the data distribution and the distribution with one-step reconstructions clamped on the visible nodes. We test the learning algorithm on the classification of digits.

1 Introduction

When learning undirected graphical models from data we change the parameters such that the model distribution is matched with the data distribution. To compute the statistics of the model distribution we need to perform inference in a network with no evidence clamped on any of its nodes. However, for a large class of models inference is intractable and approximate methods need to be employed. A wide variety of approximate inference methods are now available, like variational approximations, Markov Chain Monte Carlo (MCMC) sampling and more recently loopy Belief Propagation. Unfortunately, these methods typically fail when no external evidence is present (and the correlations are not weak), since the distribution is then likely to be highly multimodal. In this regime variational approximations fail to capture the complicated dependencies between the random variables, MCMC methods suffer from extremely slow equilibration and Belief Propagation does not converge or gives poor results. In this paper we argue therefore that instead of trying to (marginally) improve our inference methods it may be more fruitful to look for alternative learning objectives which avoid the need to compute equilibrium statistics. As one such learning objective we advocate the contrastive free energy (**CF**), introduced by (Hinton 2000) in the context of “restricted Boltzmann machines”. In this paper we will extend the use of **CF**for general undirected graphical models in the context of deterministic approximations like the naive mean field (MF), TAP and Bethe approximations.

2 Undirected Graphical Models

Consider an undirected graphical model with visible nodes v_i, hidden nodes h_i and edges e_{ij}. We will assume that each random variable associated with a node in the graph can take values from a discrete alphabet. In the context of contrastive free energies it will be natural to think in terms of the 4 classes of undirected graphical models shown in figure 1 (left). In the "Fully Connected Random Field" (FCRF) all nodes are connected and there are no conditional independence relationships. In the "Product of Experts" (PoE) model (Hinton 2000), the hidden nodes are independent given the observable nodes. The "Hidden Random Field" is the opposite architecture where the visible nodes are independent given the hidden nodes. Finally the "Bipartite Random Field" (BRF), has both the independence properties of the PoE and the HRF.

A natural objective function for learning these graphical models from data is the KL-divergence between the data distribution $P_0(\mathbf{v})$ and the equilibrium distribution $P_\infty(\mathbf{v})$. The subscripts "0" and "∞" will be clarified later, but can be understood by imagining running a Markov chain that is started at the data distribution ($t = 0$) and run until equilibrium ($t = \infty$, see figure 1, right). This KL-divergence can be rewritten as follows,

$$KL[P_0(\mathbf{v})||P_\infty(\mathbf{v})] = \mathbf{CF}_\infty = F_0 - F_\infty \geq 0 \tag{1}$$

where F_0 denotes the free energy of the distribution $P_0(\mathbf{v})P(\mathbf{h}|\mathbf{v})$, while $F_\infty = -\log(Z)$ denotes the free energy of the system at equilibrium. The free energy can be conveniently expressed in terms of the energy and entropy of the system as follows,

$$F_0 = \langle E \rangle_0 - H_0 \qquad F_\infty = \langle E \rangle_\infty - H_\infty \tag{2}$$

where $\langle . \rangle_0$ denotes averaging with respect to the joint $P(\mathbf{h}|\mathbf{v})P_0(\mathbf{v})$ and $\langle . \rangle_\infty$ denotes averaging with respect to the equilibrium distribution $P_\infty(\mathbf{v}, \mathbf{h})$. It is now easy to derive gradient descent update rules for the parameters $\boldsymbol{\theta}$ of the model,

$$\delta\boldsymbol{\theta} \propto \frac{\partial \mathbf{CF}_\infty}{\partial \boldsymbol{\theta}} = \left\langle \frac{\partial E}{\partial \boldsymbol{\theta}} \right\rangle_0 - \left\langle \frac{\partial E}{\partial \boldsymbol{\theta}} \right\rangle_\infty \tag{3}$$

In practice, we substitute the *empirical* distribution $\hat{P}_0(\mathbf{v}; \hat{\mathbf{v}}_{1:N})$ 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 nodes. One solution is to apply Gibbs sampling, which samples one node (or set of nodes) according to its posterior distribution, given the current values of all the other nodes. This strategy can also become computationally demanding since at every iteration of learning, Gibbs sampling must be performed for every data vector in the ”clamped” phase (with the data clamped to the visible nodes) and once more in the free phase (with all nodes unclamped). Moreover, at every run, we have to wait until the Markov chain has reached equilibrium, and many independent samples are produced.

3 Learning with Contrastive Free Energies

Recall that the expression for the KL-divergence between the data distribution and the equilibrium distribution (1) can be written as a difference between free energies, and the learning rule (3) as a difference of two averages. 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 nodes, we first sample the hidden nodes, given the data, then sample reconstructions of the data, given the sampled hidden nodes, etc. (see figure 1, middle & right). It is not hard to show that at every step of Gibbs

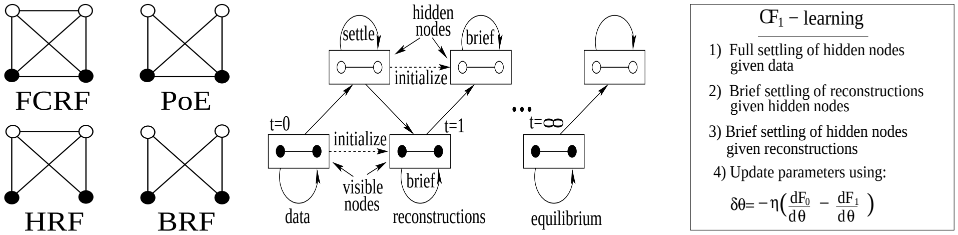
FCRF
PoE
HRF
BRF
hidden
nodes
settle
brief
initialize
t=0
initialize
t=1
visible
nodes
brief
data
reconstructions
...
t=∞
equilibrium
CF1 − learning
1) Full settling of hidden nodes given data
2) Brief settling of reconstructions given hidden nodes
3) Brief settling of hidden nodes given reconstructions
4) Update parameters using:
δθ= −η(dF0/dθ − dF1/dθ)

Figure 1: Left: Four classes of undirected graphical models; Fully Connected Random Fields (FCRF), Products of Experts (PoE), Hidden Random Fields (HRF) and Bipartite Random Fields (BRF). Middle & right: $\mathbf{CF}_1$-learning in pictures and words.

sampling the free energy will decrease, $F_0 \geq F_k \geq F_\infty \quad \forall k$. Moreover, it must therefore be true that if the free energy hasn't changed after k steps of Gibbs sampling (for any k), either $P_0 = P_\infty$ or the Markov chain does not mix. Assuming that the Markov chain mixes properly, the above suggests that we could use the following contrastive free energy,

$$\mathbf{CF}_k = F_0 - F_k = KL\left[P_0(\mathbf{v},\mathbf{h})||P_\infty(\mathbf{v},\mathbf{h})\right] - KL\left[P_k(\mathbf{v},\mathbf{h})||P_\infty(\mathbf{v},\mathbf{h})\right] \geq 0 \qquad (4)$$

as an objective to minimize. 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 may cause slow equilibration.

Learning proceeds by taking derivatives with respect to the parameters and performing gradient descent on $\mathbf{CF}_k$. The derivative is given by,

$$\delta\boldsymbol{\theta} \propto \frac{\partial \mathbf{CF}_k}{\partial \boldsymbol{\theta}} = \left\langle \frac{\partial E}{\partial \boldsymbol{\theta}} \right\rangle_0 - \left\langle \frac{\partial E}{\partial \boldsymbol{\theta}} \right\rangle_k - \frac{\partial F_k}{\partial P_k}\frac{\partial P_k}{\partial \boldsymbol{\theta}} \qquad (5)$$

The last term is hard to evaluate, but small compared with the other two. Hinton (2000) shows that it can be safely ignored.

It is important to note that brief Gibbs sampling of the reconstructions given the hidden nodes is sufficient *if we initialize the reconstutions at the data*. The subsequent sampling of the hidden nodes given the reconstructions may also be brief provided they are initialized at the (sampled) values of the hidden nodes of the clamped distribution. This procedure works since it is guaranteed, even for brief Gibbs sampling, that the free energy with respect to the clamped distribution decreases (if the chain mixes). It is unfortunately necessary to sample the hidden states given the data from the exact posterior distribution.

In practice we will replace the data distribution $P_0(\mathbf{v})$ with the empirical distribution $\hat{P}_0(\mathbf{v};\hat{\mathbf{v}}_{1:N})$ and start a Markov chain on *every data vector* while the two derivatives in the learning rule will be averaged over the data (at $t = 0$) and the reconstructions of those data respectively (at $t = 1$).

Although some progress has been been made, the new learning rules are not very efficient for general graphical models. For instance, for the HRF (and the FCRF) we still need to run Gibbs sampling to equilibrium for the hidden nodes with the data clamped to the visible nodes. The situation is significantly better for the PoE, since only brief Gibbs sampling is required for the reconstructions. For the BRF we can sample both the visible and the hidden nodes independently avoiding the costly Gibbs sampling altogether (Hinton 2000). In the next section we will extend the $\mathbf{CF}$-learning framework to include approximations of the free energy.

4 Approximate Contrastive Free Energies

An alternative to Gibbs sampling is the use of variational approximations to the free energy. A well known example of this is the mean field (MF) approximation (Peterson & Anderson 1987), where the variational parameters are the means $\mathbf{q}_0$ of the approximate (factorized) posterior distribution $Q_0(\mathbf{h}|\mathbf{v})$ and the means $\{\mathbf{q}_\infty, \mathbf{r}_\infty\}$ of the equilibrium distribution. In the following we will always denote variational parameters associated with the hidden nodes with $\mathbf{q}$ and parameters associated with the visible nodes with $\mathbf{r}$. When the parameters define a probability distribution, like in the case of MF, the parameter settings can be computed by minimizing the KL-divergence $KL[Q||P]$ between the variational distribution and the desired distribution. However, this need not be the case in general, since there are many usefull approximations of the free energy which do not come in the form $F^{\mathbf{ap}} = \langle E \rangle_Q - H(Q)$. The TAP and Bethe approximations are two such examples.

An approximate objective function for training undirected graphical models is simply to replace the exact free energies by their approximations, $F_0^{\mathbf{ap}}(\hat{\mathbf{v}}, \mathbf{q}_0) - F_\infty^{\mathbf{ap}}(\mathbf{r}_\infty, \mathbf{q}_\infty)$ where $\hat{\mathbf{v}}$ denotes the data. The key observation about this objective is that F_∞ is *always lower* than F_0 since its value is determined by minimizing over a larger set of parameters. In other words, the equilibrium distribution has more degrees of freedom since there are no data clamped on its visible nodes. Indeed, we can think of computing F_∞ by performing *coordinate descent* in the parameters $\{\mathbf{q}_\infty, \mathbf{r}_\infty\}$, initializing the parameters at their optimal values for the data distribution, so that the initial free energy is simply F_0 (see also Movellan 1991 for a similar idea). Next consider the free energy obtained after k rounds of coordinate descent and call this F_k. Trivially, we now have $F_0^{\mathbf{ap}} \geq F_k^{\mathbf{ap}} \geq F_\infty^{\mathbf{ap}} \;\; \forall k$. As argued before, variational approximations for the equilibrium distribution are unlikely to be accurate since no evidence is clamped on the visible nodes resulting in a highly multimodal distribution. By analogy to the previous section we will now propose to cut the sequence of coordinate descent at "depth k" and define the following contrastive free energy objective to be minimized,

$$\mathbf{CF}_k^{\mathbf{ap}} = F_0^{\mathbf{ap}}(\hat{\mathbf{v}}, \mathbf{q}_0) - F_k^{\mathbf{ap}}(\mathbf{r}_k, \mathbf{q}_k) \tag{6}$$

Taking derivatives with repsect to parameters $\boldsymbol{\theta}$ we get,

$$\delta\boldsymbol{\theta} \propto \frac{\partial \mathbf{CF}_k^{\mathbf{ap}}}{\partial \boldsymbol{\theta}} = \frac{\partial F_0^{\mathbf{ap}}}{\partial \boldsymbol{\theta}} - \frac{\partial F_k^{\mathbf{ap}}}{\partial \boldsymbol{\theta}} - \frac{\partial F_k^{\mathbf{ap}}}{\partial \mathbf{q}_k}\frac{\partial \mathbf{q}_k}{\partial \boldsymbol{\theta}} - \frac{\partial F_k^{\mathbf{ap}}}{\partial \mathbf{r}_k}\frac{\partial \mathbf{r}_k}{\partial \boldsymbol{\theta}} \tag{7}$$

The last two terms in this expression are difficult to compute, since we don't have explicit expressions for $\mathbf{r}_k$ and $\mathbf{q}_k$ in terms of $\boldsymbol{\theta}$. Fortunately, they are small and rarely in conflict with the other terms in the gradient so they can be safely ignored (Hinton 2000). Notice also that towards the end of learning $\partial F_k / \partial \mathbf{r}_k$ and $\partial F_k / \partial \mathbf{q}_k$ are expected to become vanishingly small.

When the paramers $\mathbf{q}$ and $\mathbf{r}$ define a probablity distribution, we can rewrite $\mathbf{CF}_k^{\mathbf{ap}}$ as

$$\mathbf{CF}_k^{\mathbf{ap}} = KL[Q_0||P_\infty] - KL[Q_k||P_\infty] \tag{8}$$

where $Q_0 = Q(\mathbf{h}|\mathbf{v})P_0(\mathbf{v})$. This is the analogue of expression (4). Similarly, we can write the learning rule analogous to (5) where we replace all occurances of P_k with Q_k and $\langle \cdot \rangle$ denotes averaging over Q. In this case we may also decide to *sample* from the MF distribution after every step of coordinate descent instead of using the mean values directly.

In practice we will start coordinate descent at every data vector seperately, i.e. we assign each data vector a seperate set of variational parameters $\{\mathbf{q}_0^n, \mathbf{q}_k^n, \mathbf{r}_k^n\}$, and average the learning rule over all data vectors. Since for all data vectors the free energy is guaranteed to decrease during settling, so is the average free energy. In our experiments we always used a depth value of $k = 1$, and only a few steps of updates for $\{\mathbf{q}_1, \mathbf{r}_1\}$ in the direction of the negative gradient of F, which increases the efficiently greatly. Note however that the minimization over $\mathbf{q}_0$ has to be run until convergence.

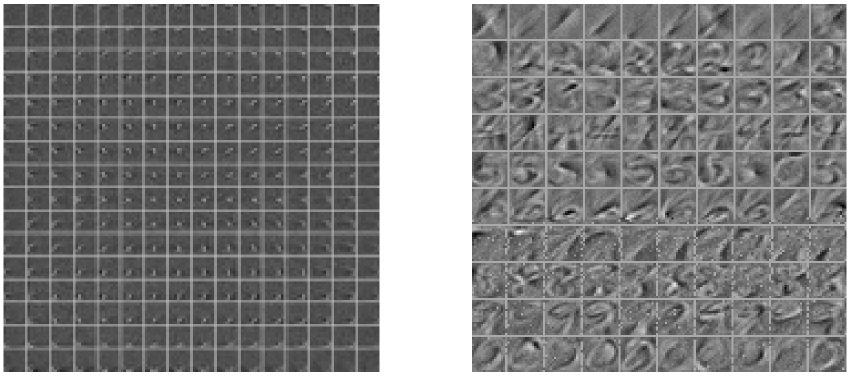

Figure 2: Left: All visible to visible connections for the 16×16 digit “8”. Every patch corresponds to the visible weights for one visible node at the corresponding location in the image (i.e. the top left patch corresponds to the top left pixel). These weights may be interpreted as a local decorrelating filter, removing first and second order statistics from the data. The higher order statistics are modelled by the hidden nodes, whose “projective fields” (hidden-to-visible weights) are shown on the right. Every row depicts the weights from 10 randomly chosen hidden nodes to all visible nodes for one particular digit-model 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).

5 Bolzmann Machines

Boltzmann Machines (Ackley et al. 1985) are binary undirected graphical models with pairwise interactions. We will propose three approximations to be used in conjunction with the CFobjective: MF, TAP and the Bethe approximation. We will call the weights between the hidden nodes $\mathbf{W}$, the weights between the hidden and visible nodes $\mathbf{J}$ and the weights between the visible nodes $\mathbf{V}$. We will also assume that there is one node with value always 1, whose weights to all other nodes represent the biases. The fixed point equations and CF-learning rules (at $k = 1$) for the MF approximation are given below,

$$\mathbf{q}_0^n = \sigma(\mathbf{W}\mathbf{q}_0^n + \mathbf{J}^T\hat{\mathbf{v}}^n) \qquad \Delta\mathbf{W} \propto \frac{1}{N}\sum_{n=1:N}(\mathbf{q}_0^n\mathbf{q}_0^n - \mathbf{q}_1^n\mathbf{q}_1^n) \tag{9}$$

$$\mathbf{r}_1^n = \sigma(\mathbf{V}\mathbf{r}_1^n + \mathbf{J}\mathbf{q}_0^n) \qquad \Delta\mathbf{V} \propto \frac{1}{N}\sum_{n=1:N}(\hat{\mathbf{v}}^n\hat{\mathbf{v}}^n - \mathbf{r}_1^n\mathbf{r}_1^n) \tag{10}$$

$$\mathbf{q}_1^n = \sigma(\mathbf{W}\mathbf{q}_1^n + \mathbf{J}^T\mathbf{r}_1^n) \qquad \Delta\mathbf{J} \propto \frac{1}{N}\sum_{n=1:N}(\hat{\mathbf{v}}^n\mathbf{q}_0^n - \mathbf{r}_1^n\mathbf{q}_1^n) \tag{11}$$

The fixed point equations (left) must be run *sequentially*. The last argument in the sigmoid is fixed and acts as an external evidence (bias) term. Also, damping may be necessary to avoid oscillations.

For the TAP approximation (Galland 1993) additional terms appear which can be implemented by the following substitutions,

$$\mathbf{W}\mathbf{q} \leftarrow \mathbf{W}\mathbf{q} + (\mathbf{1} - \mathbf{q})\sum_j \frac{1}{2}\mathbf{W}^2_{\cdot j}\, q_j(1 - q_j) \qquad q_iq_j \leftarrow q_iq_j + q_i(1 - q_i)\, W_{ij}\, q_j(1 - q_j) \tag{12}$$

and similarly for the other equations (notice that the extra terms disappear on a data vector, which is assumed binary).

	MF_*	MF_∞	MF_1	TAP	$Bethe$
BRF	$8.7 \pm 0.4\%$	$5.5 \pm 0.3\%$	$5.0 \pm 0.3\%$	IDEM MF_1	IDEM MF_1
HRF	$9.6 \pm 0.5\%$	$6.2 \pm 0.2\%$	$5.1 \pm 0.2\%$	$5.1 \pm 0.3\%$	$5.2 \pm 0.3\%$
PoE	$6.1 \pm 0.5\%$	$4.5 \pm 0.2\%$	$4.5 \pm 0.2\%$	$4.4 \pm 0.2\%$	$4.5 \pm 0.2\%$
FCRF	$5.9 \pm 0.3\%$	$4.5 \pm 0.3\%$	$4.4 \pm 0.1\%$	$4.4 \pm 0.2\%$	$4.5 \pm 0.2\%$

Table 1: Classification results for the 8×8 binary digits. In this table we compare 5 approximate methods for learning the Boltzmann machine. MF_* uses a single MF-distribution to approximate the equilibrium distribution. MF_∞ uses a separate MF-distribution for each data vector, initialized at the data and run until convergence. MF_1, TAP and $Bethe$ use CF$_1$-learning (see figure 1). These 5 different methods were compared on 4 architectures with 25 hidden nodes: BRF, PoE, HRF and FCRF. The table shows the mean and standard deviation for 10 runs of the algorithms.

Coordinate descent on the Bethe free energy is more difficult since it is parameterized in terms of both one-node marginals and pairwise marginals which should all be consistent. In the binary case, we can solve the pairwise marginals analytically in terms of the one-node marginals and insert them back into the Bethe free energy. Since the Bethe free energy is now a function of the one-node marginals alone, coordinate descent proceeds similarly as in MF or TAP, by fixing a subset of the one-node marginals and minimizing over the remaining set using gradient descent or fixed point equations. It was shown that these fixed point equations have the same fixed points as loopy BP and reduce to the TAP equations up to second order in the weights and to the MF equations up to first order in the weights. We refer to the paper (Welling & Teh, 2001) for further details of the implementation. For more general undirected graphical models we developed the "Unified Propagation and Scaling" algorithm (UPS), which descends on the Bethe free energy by combining belief propagation and iterative scaling (Teh & Welling 2001). Finally, the learning rule uses the estimates of the pairwise marginals to change the weights.

6 Learning Digit Models with CF

-Binary Digits (8×8)
In this experiment 8×8 real valued digits from the "br" set on the CEDAR cdrom were thresholded to produce binary images. There are 11000 digits available equally divided into 10 classes. The first 6000 were used for training, the next 2000 for validation and the last 3000 for testing. Separate models were trained for each digit, using 600 training examples. A total of 1500 weight updates were performed per digit model on minibatches of 10 data vectors. The updates included a small weight-decay term and a momentum term. When training was completed, we computed the free energy $F_0^{\mathbf{mf}}$ for all data on all models (including validation and test data). Since we do not compute the term $F_\infty^{\mathbf{mf}} = -\log(Z)$ (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 $F_0^{\mathbf{mf}}$ 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. The results of 5 different methods on a variety of architectures is shown in table (1). The results for 1-nearest-neighbour and multinomial logistic regression are 6.3% and 9.2% respectively.

-Real Valued Digits (16×16)
In this experiment we used the 16×16 real valued digits from the USPS Cedar ROM. 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. Each model was a binary FCRF with pairwise interactions

consisting of 50 hidden nodes. The training and classification procedures were similar as for the binary digits. 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 for the digit ”8” and the hidden-to-visible weights for some sampled hidden nodes (see figure caption for explanation).

The conclusion from these experiments is that $\mathbf{CF}$-learning improves considerably on the naive implementation of MF Boltzmann machines, where a single MF distribution is used to model the equilibrium distribution. Moreover, the performance of the $\mathbf{CF}_1$ algorithm seems to be no worse than the $\mathbf{CF}_\infty$ algorithm but much more efficient. Additional improvements over MF, like TAP or Bethe do not seem to further improve performance. The most significant gain was achieved by connecting the visible nodes. The PoE is therefore the preferred model for the digit classification task, since it has good performance (4.5%) and can be implemented efficiently.

7 Conclusion

In this paper we have argued that we can improve the effectivity of approximate inference algorithms by avoiding the need to compute equilibrium statistics. To achieve this we proposed to replace the usual maximum likelihood objective by the contrastive free energy. In experiments we have shown that in combination with the mean field approximation this provides an efficient algorithm to learn the weights (including the lateral weights between the hidden nodes) of a Boltzmann machine.

Acknowledgments

This research was funded by the Gatsby Charitable Foundation.

References

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

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

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

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

[5] Movellan, J.R. (1991) Contrastive Hebbian Learning in the Continuous Hopfield Model. it Connectionist Models, Proc. 1990 Summer School, San Mateo: Morgan Kaufmann, 10–17.

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

[7] Teh, Y.W. & Welling, M. (2000). *Passing and Bouncing Messages for Generalized Inference* (Technical Report GCNU TR 2001-001). Gatsby Computational Neuroscience unit, London, UK.

[8] Welling, M.& Teh, Y.W. (2001). Belief Optimization for Binary Networks: A Stable Alternative to Loopy Belief Propagation. accepted in *17th Conference on Uncertainty in Artificial Intelligence* , Seattle, Washington.

[9] Yedidia, J., Freeman, W. & Weiss Y. (2000). Generalized Belief Propagation. In Leen T.K., Dietterich T.G. & Tresp, V. (eds.), *Advances in Neural Information Processing Systems 13,* pp. 689–695. Cambridge, MA: MIT Press.