A MEAN FIELD LEARNING ALGORITHM FOR UNSUPERVISED NEURAL NETWORKS

LAWRENCE SAUL
AT&T Labs – Research
180 Park Ave D-130
Florham Park, NJ 07932

AND

MICHAEL JORDAN
Massachusetts Institute of Technology
Center for Biological and Computational Learning
79 Amherst Street, E10-034D
Cambridge, MA 02139

Abstract. We introduce a learning algorithm for unsupervised neural networks based on ideas from statistical mechanics. The algorithm is derived from a mean field approximation for *large, layered* sigmoid belief networks. We show how to (approximately) infer the statistics of these networks without resort to sampling. This is done by solving the mean field equations, which relate the statistics of each unit to those of its Markov blanket. Using these statistics as target values, the weights in the network are adapted by a local delta rule. We evaluate the strengths and weaknesses of these networks for problems in statistical pattern recognition.

1. Introduction

Multilayer neural networks trained by backpropagation provide a versatile framework for statistical pattern recognition. They are popular for many reasons, including the simplicity of the learning rule and the potential for discovering hidden, distributed representations of the problem space. Nevertheless, there are many issues that are difficult to address in this framework. These include the handling of missing data, the statistical interpretation of hidden units, and the problem of unsupervised learning, where there are no explicit error signals.

One way to handle these problems is to view these networks as probabilistic models. This leads one to consider the units in the network as random variables, whose statistics are encoded in a joint probability distribution. The learning problem, originally one of function approximation, now becomes one of density estimation under a latent variable model; the objective function is the log-likelihood of the training data. The probabilistic semantics in these networks allow one to infer target values for hidden units, even in an unsupervised setting.

The Boltzmann machine[1] was the first neural network to be endowed with probabilistic semantics. It has a simple Hebb-like learning rule and a fully probabilistic interpretation as a Markov random field. A serious problem for Boltzmann machines, however, is computing the statistics that appear in the learning rule. In general, one has to rely on approximate methods, such as Gibbs sampling or mean field theory[2], to estimate these statistics; exact calculations are not tractable for layered networks. Experience has shown, however, that sampling methods are too slow, and mean field approximations too impoverished[3], to be used in this way.

A different approach has been to recast neural networks as layered belief networks[4]. These networks have a fully probabilistic interpretation as directed graphical models[5, 6]. They can also be viewed as *top-down* generative models for the data that is encoded by the units in the bottom layer[7, 8, 9]. Though it remains difficult to compute the statistics of the hidden units, the directionality of belief networks confers an important advantage. In these networks one can derive a simple lower bound on the likelihood and develop learning rules based on maximizing this lower bound.

The Helmholtz machine[7, 8] was the first neural network to put this idea into practice. It uses a fast, bottom-up recognition model to compute the statistics of the hidden units and a simple stochastic learning rule, known as wake-sleep, to adapt the weights. The tradeoff for this simplicity is that the recognition model cannot handle missing data or support certain types of reasoning, such as *explaining away*[5], that rely on top-down and bottom-up processing.

In this paper we consider an algorithm based on ideas from statistical mechanics. Our lower bound is derived from a mean field approximation for sigmoid belief networks[4]. The original derivation[10] of this approximation made no restrictions on the network architecture or the location of visible units. The purpose of the current paper is to tailor the approximation to networks that represent hierarchical generative models. These are multilayer networks whose visible units occur in the bottom layer and whose topmost layers contain large numbers of hidden units.

The mean field approximation that emerges from this specialization is interesting in its own right. The *mean field equations*, derived by maximizing the lower bound on the log-likelihood, relate the statistics of each unit to those of its Markov blanket. Once estimated, these statistics are used to fill in target values for hidden units. The learning algorithm adapts the weights in the network by a local delta rule. Compact and intelligible, the approximation provides an attractive computational framework for probabilistic modeling in layered belief networks. It also represents a viable alternative to sampling, which has been the dominant paradigm for inference and learning in large belief networks.

While this paper builds on previous work, we have tried to keep it self-contained. The organization of the paper is as follows. In section 2, we examine the modeling problem for unsupervised networks and give a succinct statement of the learning algorithm. (A full derivation of the mean field approximation is given in the appendix.) In section 3, we assess the strengths and weaknesses of these networks based on experiments with handwritten digits. Finally, in section 4, we present our conclusions, as well as some directions for future research.

2. Generative models

Suppose we are given a large sample of binary (0/1) vectors, then asked to model the process by which these vectors were generated. A multilayer network (see Figure 1) can be used to parameterize a generative model of the data in the following way. Let S_i^ℓ denote the ith unit in the ℓth layer of the network, h_i^ℓ its bias, and $J_{ij}^{\ell-1}$ the weights that feed into this unit from the layer above. We imagine that each unit represents a binary random variable whose probability of activation, in the data-generating process, is conditioned on the units in the layer above. Thus we have:

$$P(S_i^\ell = 1|S^{\ell-1}) = \sigma\left(\textstyle\sum_j J_{ij}^{\ell-1} S_j^{\ell-1} + h_i^\ell\right), \tag{1}$$

where $\sigma(z) = [1 + e^{-z}]^{-1}$ is the sigmoid function. We denote by σ_i^ℓ the squashed sum of inputs that appears on the right hand side of eq. (1). The joint distribution over all the units in the network is given by:

$$P(S) = \prod_{\ell i} (\sigma_i^\ell)^{S_i^\ell} (1 - \sigma_i^\ell)^{1-S_i^\ell} \tag{2}$$

A neural network, endowed with probabilistic semantics in this way, is known as a sigmoid belief network[4]. Layered belief networks were proposed as hierarchical generative models by Hinton et al[7].

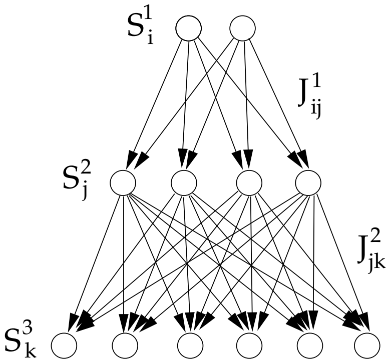

Figure 1. A multilayer sigmoid belief network that parameterizes a generative model for the units in the bottom layer.

The goal of unsupervised learning is to model the data by the units in the bottom layer. We shall refer to these units as visible (V) units, since the data vectors provide them with explicit target values. For the other units in the network—the hidden (H) units—appropriate target values must be inferred from the probabilistic semantics encoded in eq. (2).

2.1. MAXIMUM LIKELIHOOD ESTIMATION

The problem of unsupervised learning, in this framework, is essentially one of density estimation. A network with many hidden units can parameterize a large family of distributions over the visible units. The learning problem is to find the weights J^ℓ_{ij} and biases h^ℓ_i that make the statistics of the visible units match those of the data.

One simple approach to learning is to maximize the log-likelihood[1] of the training sample. The likelihood of each data vector is obtained from its marginal distribution,

$$P(V) = \sum_H P(H,V), \tag{3}$$

where by definition $P(H,V) = P(S)$ is the joint distribution over all units (hidden and visible). We can derive local learning rules by computing the gradients of the log-likelihood, $\ln P(V)$, with respect to the weights and biases of the network. For each data vector, this gives the on-line updates:

$$\Delta J^\ell_{ij} \propto \mathrm{E}\left[(S^{\ell+1}_i - \sigma^{\ell+1}_i)S^\ell_j\right], \tag{4}$$

$$\Delta h^\ell_i \propto \mathrm{E}\left[S^\ell_i - \sigma^\ell_i\right], \tag{5}$$

[1] For simplicity of exposition, we do not consider forms of regularization (e.g., penalized likelihoods, cross-validation) that may be necessary to prevent overfitting.

where E[$\cdots$] denotes an expectation with respect to the conditional distribution, $P(H|V)$. Note that the updates take the form of a delta rule, with unit activations σ_i^ℓ being matched to target values S_i^ℓ. Many authors[4, 18] have noted the associative, error-correcting nature of gradient-based learning rules in belief networks.

2.2. MEAN FIELD LEARNING

In general, it is intractable[11, 12] to calculate the likelihood in eq. (3) or the statistics in eqs. (4–5). It is also time-consuming to estimate them by sampling from $P(H|V)$. One way to proceed is based on the following idea[7]. Suppose we have an approximate distribution, $Q(H|V) \approx P(H|V)$. Using Jensen's inequality, we can form a lower bound on the log-likelihood from:

$$\ln P(V) \geq \sum_H Q(H|V) \ln \left[\frac{P(H,V)}{Q(H|V)}\right]. \tag{6}$$

If this bound is easy to compute, then we can derive learning rules based on maximizing the bound. Though one cannot guarantee that such learning rules always increase the actual likelihood, they provide an efficient alternative to implementing the learning rules in eqs. (4–5).

Our choice of $Q(H|V)$ is motivated by ideas from statistical mechanics. The mean field approximation[13] is a general method for estimating the statistics of large numbers of correlated variables. The starting point of the mean field approximation is to consider factorized distributions of the form:

$$Q(H|V) = \prod_\ell \prod_{i\in H} (\mu_i^\ell)^{S_i^\ell}(1-\mu_i^\ell)^{1-S_i^\ell} \tag{7}$$

The parameters μ_i^ℓ are the mean values of S_i^ℓ under the distribution $Q(H|V)$, and they are chosen to maximize the lower bound in eq. (6). A full derivation of the mean field theory for these networks, starting from eqs. (6) and (7), is given in the appendix. Our goal in this section, however, is to give a succinct statement of the learning algorithm. In what follows, we therefore present only the main results, along with a number of useful intuitions.

For these networks, the mean field approximation works by keeping track of two parameters, $\{\mu_i^\ell, \xi_i^\ell\}$ for each unit in the network. Roughly speaking, these parameters are stored as approximations to the true statistics of the hidden units: $\mu_i^\ell \approx \mathrm{E}[S_i^\ell]$ approximates the mean of S_i^ℓ, while $\xi_i^\ell \approx \mathrm{E}[\sigma_i^\ell]$ approximates the average value of the squashed sum of inputs. Though only the first of these appears explicitly in eq. (7), it turns out that both are needed to compute a lower bound on the log-likelihood. The values of $\{\mu_i^\ell, \xi_i^\ell\}$ depend on the states of the visible units, as well as the weights and biases of the network. They are computed by solving the *mean* *field equations*:

$$\mu_i^\ell = \sigma\left[\sum_j J_{ij}^{\ell-1}\mu_j^{\ell-1} + h_i^\ell + \sum_j J_{ji}^\ell(\mu_j^{\ell+1} - \xi_j^{\ell+1}) - \frac{1}{2}(1-2\mu_i^\ell)\sum_j (J_{ji}^\ell)^2\xi_j^{\ell+1}(1-\xi_j^{\ell+1})\right] \quad (8)$$

$$\xi_i^\ell = \sigma\left[\sum_j J_{ij}^{\ell-1}\mu_j^{\ell-1} + h_i^\ell + \frac{1}{2}(1-2\xi_i^\ell)\sum_j (J_{ij}^{\ell-1})^2\mu_j^{\ell-1}(1-\mu_j^{\ell-1})\right]. \quad (9)$$

These equations couple the parameters of each unit to those in adjacent layers. The terms inside the brackets can be viewed as effective influences (or "mean fields") on each unit in the network. The reader will note that sigmoid belief networks have twice as many mean field parameters as their undirected counterparts[2]. For this we can offer the following intuition. Whereas the parameters μ_i^ℓ are determined by top-down and bottom-up influences, the parameters ξ_i^ℓ are determined only by top-down influences. The distinction—essentially, one between parents and children—is only meaningful for directed graphical models.

The procedure for solving these equations is fairly straightforward. Initial guesses for $\{\mu_i^\ell, \xi_i^\ell\}$ are refined by alternating passes through the network, in which units are updated one layer at a time. We alternate these passes in the bottom-up and top-down directions so that information is propagated from the visible units to the hidden units, and vice versa. The visible units remain clamped to their target values throughout this process. Further details are given in the appendix.

The learning rules for these networks are designed to maximize the bound in eq. (6). An expression for this bound, in terms of the weights and biases of the network, is derived in the appendix; see eq. (24). Gradient ascent in J_{ij}^ℓ and h_i^ℓ leads to the learning rules:

$$\Delta J_{ij}^\ell \propto \left[\left(\mu_i^{\ell+1} - \xi_i^{\ell+1}\right)\mu_j^\ell - J_{ij}^\ell\xi_i^{\ell+1}(1-\xi_i^{\ell+1})\mu_j^\ell(1-\mu_j^\ell)\right], \quad (10)$$

$$\Delta h_i^\ell \propto (\mu_i^\ell - \xi_i^\ell). \quad (11)$$

Comparing these learning rules to eqs. (4–5), we see that the mean field parameters fill in for the statistics of S_i^ℓ and σ_i^ℓ. This is, of course, what makes the learning algorithm tractable. Whereas the statistics of $P(H|V)$ cannot be efficiently computed, the parameters $\{\mu_i^\ell, \xi_i^\ell\}$ can be found by solving the mean field equations. We obtain a simple on-line learning algorithm by solving the mean field equations for each data vector in the training set, then adjusting the weights by the learning rules, eqs. (10) and (11).

The reader may notice that the rightmost term of eq. (10) has no counterpart in eq. (4). This term, a regularizer induced by the mean field approximation, causes J_{ij}^ℓ to be decayed according to the mean-field statistics of $\sigma_i^{\ell+1}$ and S_j^ℓ. In particular, the weight decay is suppressed if either $\xi_i^{\ell+1}$ or μ_j^ℓ is saturated near zero or one; in effect, weights between highly correlated units are *burned in* to their current values.

3. Experiments

We used a large database of handwritten digits to evaluate the strengths and weaknesses of these networks. The database[16] was constructed from NIST Special Databases 1 and 3. The examples in this database were deslanted, downsampled, and thresholded to create 10×10 binary images. There were a total of 60000 examples for training and 10000 for testing; these were divided roughly equally among the ten digits ZERO to NINE. Our experiments had several goals: (i) to evaluate the speed and performance of the mean field learning algorithm; (ii) to assess the quality of multilayer networks as generative models; (iii) to see whether classifiers based on generative models work in high dimensions; and (iv) to test the robustness of these classifiers with respect to missing data.

We used the mean field algorithm from the previous section to learn generative models for each digit. The generative models were parameterized by four-layer networks with $4 \times 12 \times 36 \times 100$ architectures. Each network was trained by nine passes[2] through the training examples. Figure 2 shows a typical plot of how the score computed from eq. (6) increased during training. To evaluate the discriminative capabilities of these models, we trained ten networks, one for each digit, then used these networks to classify the images in the test set. The test images were labeled by whichever network assigned them the highest likelihood score, computed from eq. (6). Each of these experiments required about nineteen CPU hours on an SGI R10000, or roughly 0.12 seconds of processing time per image per network. We conducted five such experiments; the error rates were 4.9%(×2), 5.1%(×2), and 5.2%. By comparison, the error rates[3] of several k-nearest neighbor algoriths were: 6.3% ($k = 1$), 5.8% ($k = 3$), 5.5% ($k = 5$), 5.4% ($k = 7$), and 5.5% ($k = 9$). These results show that the networks have learned noisy but essentially accurate models of each digit class. This is confirmed by looking at images sampled from the generative model of each network; some of these are shown in figure 3.

One advantage of generative models for classification is the seamless handling of missing data. Inference in this case is simply performed on the

[2] The first pass through the training examples was used to initialize the biases of the bottom layer; the rest were used for learning. The learning rate followed a fixed schedule: 0.02 for four epochs and 0.005 for four epochs.

[3] All the error rates in this paper apply to experiments with 10×10 binary images. The best backpropagation networks[16], which exploit prior knowledge and operate on 20×20 greyscale images, can obtain error rates less than one percent.

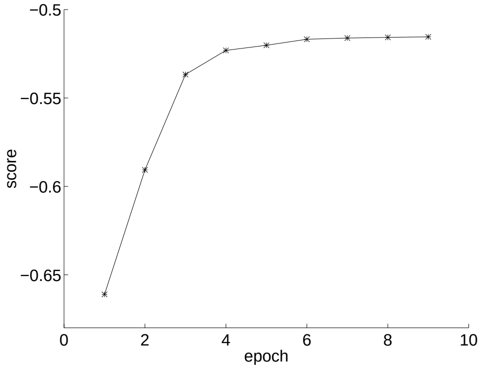

Figure 2. Plot of the lower bound on the log-likelihood, averaged over training patterns, versus the number of epochs, for a $4 \times 12 \times 36 \times 100$ network trained on the digit TWO. The score has been normalized by $100 \times \ln 2$.

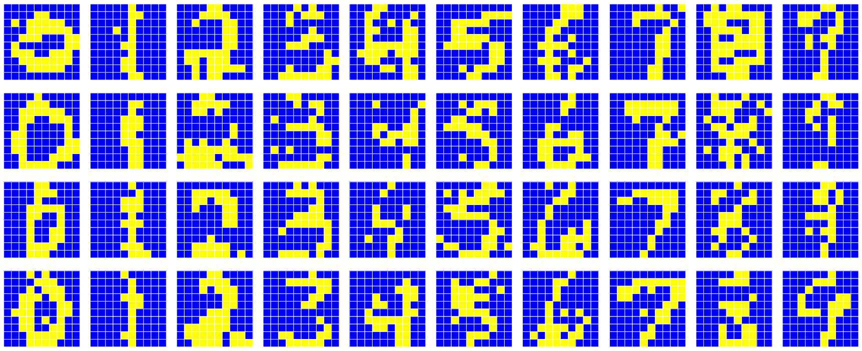

Figure 3. Synthetic images sampled from each digit's generative model.

pruned network in which units corresponding to missing pixels have been removed (i.e., marginalized). We experimented by randomly labeling a certain fraction, f, of pixels in the test set as missing, then measuring the number of classification errors versus f. The solid line in figure 4 shows a plot of this curve for one of the mean field classifiers. The overall performance degrades gradually from 5% error at $f = 0$ to 12% error at $f = 0.5$.

One can also compare the mean field networks to other types of generative models. The simplest of these is a mixture model in which the pixel values (within each mixture component) are conditionally distributed as independent binary random variables. Models of this type can be trained by an Expectation-Maximization (EM) algorithm[19] for maximum likelihood estimation. Classification via mixture models was investigated in a separate set of experiments. Each experiment consisted of training ten mixture models, one for each digit, then using the mixture models to classify the

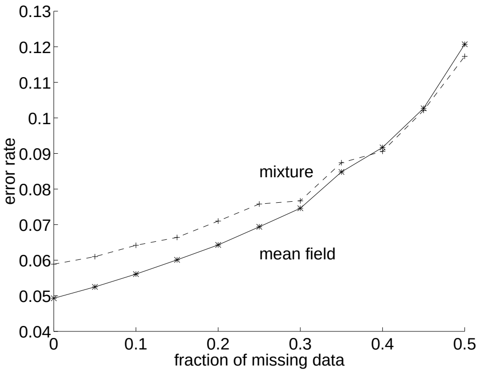

Figure 4. Plot of classification error rate versus fraction of missing pixels in the test set. The solid curve gives the results for the mean field classifier; the dashed curve, for the mixture model classifier.

digits in the test set. The mixture models had forty mixture components and were trained by ten iterations of EM. The classification error rates in five experiments were 5.9%(×3) and 6.2%(×2); the robustness of the best classifier to missing data is shown by the dashed line in figure 4. Note that while the mixture models had roughly the same number of free parameters as the layered networks, the error rates were generally higher. These results suggest that hierarchical generative models, though more difficult to train, may have representational advantages over mixture models.

4. Discussion

The trademarks of neural computation are simple learning rules, local message-passing, and hierarchical distributed representations of the problem space. The backpropagation algorithm for multilayer networks showed that many supervised learning problems were amenable to this style of computation. It remains a challenge to find an unsupervised learning algorithm with the same widespread potential.

In this paper we have developed a mean field algorithm for unsupervised neural networks. The algorithm captures many of the elements of neural computation in a sound, probabilistic framework. Information is propagated by local message-passing, and the learning rule—derived from a lower bound on the log-likelihood—combines delta-rule adaptation with weight decay and burn-in. All these features demonstrate the advantages of tailoring a mean field approximation to the properties of layered networks.

It is worth comparing our approach to methods based on Gibbs sampling[4]. One advantage of the mean field approximation is that it enables one to compute a lower bound on the marginal likelihood, $P(V)$. Estimating these likelihoods by sampling is not so straightforward; indeed, it is considerably harder than estimating the statistics of individual units. In a recent study, Frey et al[15] reported that learning by Gibbs sampling was an extremely slow process for sigmoid belief networks.

Mean field algorithms are evolving. The algorithm in this paper is considerably faster and easier to implement than our previous one[10, 15]. There are several important areas for future research. Currently, the overall computation time is dominated by the iterative solution of the mean field equations. It may be possible to reduce processing times by fine tuning the number of mean field updates or by training a feed-forward network (i.e., a bottom-up recognition model[7, 8]) to initialize the mean field parameters close to solutions of eqs. (8–9). In the current implementation, we found the processing times (per image) to scale linearly with the number of weights in the network. An interesting question is whether mean field algorithms can support massively parallel implementations.

With better algorithms come better architectures. There are many possible elaborations on the use of layered belief networks as hierarchical generative models. Continuous-valued units, as opposed to binary-valued ones, would help to smooth the output of the generative model. We have not exploited any sort of local connectivity between layers, although this structure is known to be helpful in supervised learning[16]. An important consideration is how to incorporate prior knowledge about the data (e.g., translation/rotation invariance[20]) into the network. Finally, the synthetic images in figure 3 reveal an inherent weakness of top-down generative models; while these models require an element of stochasticity to model the variability in the data, they lack a feedback mechanism (i.e., relaxation[21]) to clean up noisy pixels. These extensions and others will be necessary to realize the full potential of unsupervised neural networks.

ACKNOWLEDGEMENTS

The authors acknowledge useful discussions with T. Jaakkola, H. Seung, P. Dayan, G. Hinton, and Y. LeCun and thank the anonymous reviewers for many helpful suggestions.

A. Mean field approximation

In this appendix we derive the mean field approximation for large, layered sigmoid belief networks. Starting from the factorized distribution for $Q(H|V)$, eq. (7), our goal is to maximize the lower bound on the log-likelihood, eq. (6). This bound consists of the difference between two terms:

$$\ln P(V) \geq \left[-\sum_H Q(H|V)\ln Q(H|V)\right] - \left[-\sum_H Q(H|V)\ln P(H,V)\right]. \tag{12}$$

The first term is simply the entropy of the mean field distribution. Because $Q(H|V)$ is fully factorized, the entropy is given by:

$$-\sum_H Q(H|V)\ln Q(H|V) = -\sum_{i\ell\in H}\left[\mu_i^\ell \ln \mu_i^\ell + (1-\mu_i^\ell)\ln(1-\mu_i^\ell)\right] \tag{13}$$

We identify the second term in eq. (12) as (minus) the mean field energy; the name arises from interpreting $P(H,V) = e^{\ln P(H,V)}$ as a Boltzmann distribution. Unlike the entropy, the energy term in eq. (12) is not so straightforward.

The difficulty in evaluating the energy stems from the form of the joint distribution, eq. (2). To see this, let

$$z_i^\ell = \sum_j J_{ij}^{\ell-1} S_j^{\ell-1} + h_i^\ell \tag{14}$$

denote the weighted sum of inputs into unit S_i^ℓ. From eqs. (1) and (2), we can write the joint distribution in sigmoid belief networks as:

$$\begin{aligned}\ln P(S) &= -\sum_{\ell i}\left\{S_i^\ell \ln\left[1+e^{-z_i^\ell}\right] + (1-S_i^\ell)\ln\left[1+e^{z_i^\ell}\right]\right\} && (15)\\ &= \sum_{\ell i}\left\{S_i^\ell z_i^\ell - \ln\left[1+e^{z_i^\ell}\right]\right\}. && (16)\end{aligned}$$

The difficulty in evaluating the mean field energy is the logarithm on the right hand side of eq. (16). This term makes it impossible to perform the averaging of $\ln P(S)$ in closed form, even for the simple distribution, eq. (7).

Clearly, another approximation is needed to evaluate $\langle \ln[1+e^{z_i^\ell}]\rangle$, averaged over the distribution, $Q(H|V)$. We can make progress by studying the sum of inputs, z_i^ℓ, as a random variable in its own right. Under the distribution $Q(H|V)$, the right hand side of eq. (14) is a weighted sum of independent random variables with means $\mu_j^{\ell-1}$ and variances $\mu_j^{\ell-1}(1-\mu_j^{\ell-1})$. The number of terms in this sum is equal to the number of hidden units in the $(\ell-1)$th layer of the network. In large networks, we expect the statistics of this sum—or more precisely, the mean field distribution $Q(z_i^\ell|V)$—to be governed by a central limit theorem. In other words, to a very good approximation, $Q(z_i^\ell|V)$ assumes a normal distribution with mean and variance:

$$\langle z_i^\ell \rangle = \sum_j J_{ij}^{\ell-1}\mu_j^{\ell-1} + h_i^\ell, \tag{17}$$

$$\left\langle(\delta z_i^\ell)^2\right\rangle = \sum_j (J_{ij}^{\ell-1})^2 \mu_j^{\ell-1}(1-\mu_j^{\ell-1}). \qquad (18)$$

In what follows, we will use the approximation that $Q(z_i^\ell|V)$ is Gaussian to simplify the mean field theory for sigmoid belief networks. The approximation is well suited to layered networks where each unit receives a large number of inputs from the (hidden) units in the preceding layer.

The asymptotic form of $Q(z_i^\ell|V)$ and the logarithm term in eq. (16) motivate us to consider the following lemma. Let z denote a Gaussian random variable with mean $\langle z\rangle$ and variance $\langle\delta z^2\rangle$, and consider the expected value, $\langle\ln[1+e^z]\rangle$. For any real number ξ, we can form the upper bound[22]:

$$\begin{aligned}
\langle\ln[1+e^z]\rangle &= \langle\ln[e^{\xi z}e^{-\xi z}(1+e^z)]\rangle, & (19)\\
&= \xi\langle z\rangle + \langle\ln[e^{-\xi z}+e^{(1-\xi)z}]\rangle, & (20)\\
&\leq \xi\langle z\rangle + \ln\langle e^{-\xi z}+e^{(1-\xi)z}\rangle, & (21)
\end{aligned}$$

where the last line follows from Jensen's inequality. Since z is Gaussian, it is straightforward to perform the averages on the right hand side. This gives us an upper bound on $\langle\ln[1+e^z]\rangle$ expressed in terms of the mean and variance:

$$\langle\ln[1+e^z]\rangle \leq \frac{1}{2}\xi^2\langle\delta z^2\rangle + \ln\left[1+e^{\langle z\rangle+(1-2\xi)\langle\delta z^2\rangle/2}\right]. \qquad (22)$$

In what follows, we will use this bound to approximate the exact value of $\langle\ln[1+e^{z_i^\ell}]\rangle$. Recall from eq. (16) that these are the intractable averages that appear in the mean field energy; we are therefore motivated to find the value of ξ that makes the bound as tight as possible. The right hand side of eq. (22) is minimized when:

$$\xi = \sigma\left[\langle z\rangle + \frac{1}{2}(1-2\xi)\langle\delta z^2\rangle\right]. \qquad (23)$$

Eq. (23) has a unique solution in the interval $\xi \in [0,1]$. Given values for $\langle z\rangle$ and $\langle\delta z^2\rangle$, it is easily solved by iteration; in fact, the iteration $\xi \leftarrow \sigma\left[\langle z\rangle + \frac{1}{2}(1-2\xi)\langle\delta z^2\rangle\right]$ is guaranteed to tighten the upper bound in eq. (22). We can understand eq. (23) as a self-consistent approximation for computing $\xi \approx \langle\sigma(z)\rangle$ where z is a Gaussian random variable. To see this, consider the limiting behaviors: $\langle\sigma(z)\rangle \to \sigma(\langle z\rangle)$ as $\langle\delta z^2\rangle \to 0$ and $\langle\sigma(z)\rangle \to \frac{1}{2}$ as $\langle\delta z^2\rangle \to \infty$. Eq. (23) captures both these limits and interpolates smoothly between them for finite $\langle\delta z^2\rangle$.

Equipped with the lemma, eq. (22), we can proceed to deal with the intractable terms in the mean field energy. Unable to compute the average over $Q(H|V)$ exactly, we instead settle for the tightest possible bound. This is done by introducing a new mean field parameter, ξ_i^ℓ, for each unit in the network, then substituting ξ_i^ℓ and the statistics of z_i^ℓ into eq. (22). Note that these terms appear in eq. (6) with an overall minus sign; thus, to the extent that $Q(z_i^l|V)$ is well approximated by a Gaussian distribution, the upper bound in eq. (22) translates[4] into a lower bound on the log likelihood.

Assembling all the terms in eq. (12), we obtain an objective function for the mean field approximation:

$$\begin{aligned}\ln P(V) \;\geq\; & -\sum_{i\ell\in H}\left[\mu_i^\ell\ln\mu_i^\ell+(1-\mu_i^\ell)\ln(1-\mu_i^\ell)\right] \;+\; \sum_{ij\ell}J_{ij}^{\ell-1}\mu_i^\ell\mu_j^{\ell-1} \qquad (24)\\ & +\sum_{i\ell}h_i^\ell\mu_i^\ell \;-\; \frac{1}{2}\sum_{ij\ell}(\xi_i^\ell)^2(J_{ij}^{\ell-1})^2\mu_j^{\ell-1}(1-\mu_j^{\ell-1})\\ & -\sum_{i\ell}\ln\left\{1+e^{h_i^\ell+\sum_j\left[J_{ij}^{\ell-1}\mu_j^{\ell-1}+\frac{1}{2}(1-2\xi_i^\ell)(J_{ij}^{\ell-1})^2\mu_j^{\ell-1}(1-\mu_j^{\ell-1})\right]}\right\}\end{aligned}$$

The mean field parameters are chosen to maximize eq. (24) for different settings of the visible units. Equating their gradients to zero gives the mean field equations, eqs. (8–9). Likewise, computing the gradients for J_{ij}^ℓ and h_i^ℓ gives the learning rules in eqs. (10–11).

The mean field equations are solved by finding a local maximum of eq. (24). This can be done in many ways. The strategy we chose was cyclic stepwise ascent—fixing all the parameters except one, then locating the value of that parameter that maximizes eq. (24). This procedure for solving the mean field equations can be viewed as a sequence of local message-passing operations that "match" the statistics of each hidden unit to those of its Markov blanket[5]. For the parameters ξ_i^ℓ, new values can be found by iterating eq. (9); it is straightforward to show that this iteration always leads to an increase in the objective function. On the other hand, iterating eq. (8) for μ_i^ℓ does *not* always lead to an increase in eq. (24); hence, the optimal values for μ_i^ℓ cannot be found in this way. Instead, for each update, one must search the interval $\mu_i^\ell \in [0,1]$ using some sort of bracketing procedure[17] to find a local maximum in eq. (24). This is necessary to ensure that the mean field parameters converge to a solution of eqs. (8–9).

[4] Our earlier work[10] showed how to obtain a strict lower bound on the log likelihood; i.e., the earlier work made no appeal to a Gaussian approximation for $Q(z_i^\ell|V)$. For the networks considered here, however, we find the difference between the approximate bound and the strict bound to be insignificant in practice. Moreover, the current algorithm has advantages in simplicity and interpretability.

References

1. D. Ackley, G. Hinton, and T. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science* **9**:147–169 (1985).
2. C. Peterson and J. R. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems* **1**:995–1019 (1987).
3. C. Galland. The limitations of deterministic Boltzmann machine learning. *Network* **4**:355–379.
4. R. Neal. Connectionist learning of belief networks. *Artificial Intelligence* **56**:71–113 (1992).
5. J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann: San Mateo, CA (1988).
6. S. Lauritzen. *Graphical Models.* Oxford University Press: Oxford (1996).
7. G. Hinton, P. Dayan, B. Frey, and R. Neal. The wake-sleep algorithm for unsupervised neural networks. *Science* **268**:1158–1161 (1995).
8. P. Dayan, G. Hinton, R. Neal, and R. Zemel. The Helmholtz machine. *Neural Computation* **7**:889–904 (1995).
9. M. Lewicki and T. Sejnowski. Bayesian unsupervised learning of higher order structure. In M. Mozer, M. Jordan, and T. Petsche, eds. *Advances in Neural Information Processing Systems* **9**: . MIT Press: Cambridge (1996).
10. L. Saul, T. Jaakkola, and M. Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research* **4**:61–76 (1996).
11. G. Cooper. Computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence* **42**:393-405 (1990).
12. P. Dagum and M. Luby. Approximately probabilistic reasoning in Bayesian belief networks is NP-hard. *Artificial Intelligence* **60**:141-153 (1993).
13. G. Parisi. *Statistical Field Theory.* Addison-Wesley: Redwood City (1988).
14. J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley: Redwood City (1991).
15. B. Frey, G. Hinton, and P. Dayan. Does the wake-sleep algorithm produce good density estimators? In D. Touretzky, M. Mozer, and M. Hasselmo, eds. *Advances in Neural Information Processing Systems* **8**:661-667. MIT Press: Cambridge, MA (1996).
16. Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. In *Proceedings of ICANN'95.*
17. W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes.* Cambridge University Press: Cambridge (1986).
18. S. Russell, J. Binder, D. Koller, and K. Kanazawa. Local learning in probabilistic networks with hidden variables. In *Proceedings of IJCAI-95.*
19. A. Dempster, N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* B39:1–38.
20. P. Simard, Y. LeCun, and J. Denker. Efficient pattern recognition using a new transformation distance. In S. Hanson, J. Cowan, and C. Giles, eds. *Advances in Neural Information Processing Systems* **5**:50–58. Morgan Kaufmann: San Mateo, CA (1993).
21. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**:721–741 (1984).
22. H. Seung. Annealed theories of learning. In J.-H. Oh, C. Kwon, and S. Cho, eds. *Neural Networks: The Statistical Mechanics Perspective, Proceedings of the CTP-PRSRI Joint Workshop on Theoretical Physics.* World Scientific: Singapore (1995).