Attractor dynamics in feedforward neural networks

Lawrence K. Saul
AT&T Labs – Research
180 Park Ave E-171
Florham Park, NJ 07932
lsaul@research.att.com

Michael I. Jordan
University of California
739 Soda Hall
Berkeley, CA 94720
jordan@cs.berkeley.edu

Abstract

We study the probabilistic generative models parameterized by feedforward neural networks. An attractor dynamics for probabilistic inference in these models is derived from a mean field approximation for large, layered sigmoidal networks. Fixed points of the dynamics correspond to solutions of the mean field equations, which relate the statistics of each unit to those of its Markov blanket. We establish global convergence of the dynamics by providing a Lyapunov function and show that the dynamics generate the signals required for unsupervised learning. Our results for feedforward networks provide a counterpart to those of Cohen-Grossberg and Hopfield for symmetric networks.

1 Introduction

Attractor neural networks lend a computational purpose to continuous dynamical systems. Celebrated uses of these networks include the storage of associative memories (Amit, 1989), the reconstruction of noisy images (Koch et al, 1986), and the search for shortest paths in the traveling salesman problem (Hopfield & Tank, 1986). In all these examples, a distributed computation is performed by an attractor dynamics and its flow to stable fixed points. These examples can also be formulated as problems in probabilistic reasoning, and indeed it is well known that symmetric neural networks can be analyzed as statistical mechanical ensembles or Markov random fields (MRFs).

Attractor neural networks and Markov random fields are connected by the idea of an energy surface. This connection has led to new algorithms for probabilistic inference in symmetric networks, a problem traditionally addressed via stochastic sampling procedures (Metropolis et al, 1953; Geman & Geman, 1984). For example, in one of the first unsupervised learning algorithms for neural networks, Ackley et al (1985) applied Gibbs sampling to estimate the statistics of binary MRFs. Known as Boltzmann machines, these networks relied on time-consuming Monte Carlo simulation and simulated annealing as an inner loop of their learning procedure. Subsequently, Peterson and Anderson (1987) introduced a faster deterministic method for probabilistic inference. Their method, based on the so-called *mean field approximation* from statistical mechanics, transformed the binary-valued MRF into a network with continuous-valued units. The continuous network, endowed with dynamics given by the mean field equations, is itself an attractor network; in particular, it possesses a Lyapunov function (Cohen & Grossberg, 1983; Hopfield, 1984). Thus one can perform approximate probabilistic inference in a binary MRF by relaxing a deterministic, continuous network.

In this paper, we show that this linkage of attractor dynamics and probabilistic inference is not limited to symmetric networks or (equivalently) to models represented as undirected graphs. We investigate an attractor dynamics for feedforward networks, or directed acyclic graphs (DAGs); these are networks with directed edges, but no directed loops. The probabilistic models represented by DAGs are known as Bayesian networks, and together with MRFs, comprise the class of probabilistic models known as *graphical models* (Lauritzen, 1996). Like their undirected counterparts, Bayesian networks have been proposed as models of both artificial and biological intelligence (Pearl, 1988).

The units in Bayesian networks represent random variables, while the links represent assertions of conditional independence. These independence relations endow DAGs with a precise probabilistic semantics. Any joint distribution over a fixed, finite set of random variables can be represented by a Bayesian network, just as it can be represented by an MRF. What is compactly represented by one type of graphical model, however, may be quite clumsily represented by the other. MRFs arise naturally in statistical mechanics, where they describe the Gibbs distributions for systems in thermal equilibrium. Bayesian networks, on the other hand, are designed to model causal or generative processes; hidden Markov models, Kalman filters, soft-split decision trees—these are all examples of Bayesian networks.

The connection between Bayesian networks and neural network models of learning was pointed out by Neal (1992). Neal showed that feedforward neural networks with sigmoid squashing functions have a natural expression as Bayesian networks with binary-valued units. He also showed that these probabilistic networks have gradient-based learning rules that depend only on locally available information (Buntine, 1994; Binder et al, 1997). These observations led Hinton et al (1995) to propose the Helmholtz machine—a multilayered probabilistic network that learns hierarchical generative models of sensory inputs. Helmholtz machines were conceived not only as tools for statistical pattern recognition, but also as abstract models of top-down and bottom-up processing in the brain.

Following Hinton et al (1995), a number of researchers began to investigate unsupervised learning in large layered Bayesian networks (Dayan et al, 1996; Lewicki & Sejnowski, 1996). As in undirected MRFs, probabilistic inference in these networks is generally intractable, and approximations are required. Saul et al (1996) proposed a mean field approximation for these networks, analogous to the existing one for Boltzmann machines. Their approximation transformed the binary-valued network into a continuous-valued network whose statistics were described by a set of mean field equations. These equations related the statistics of each unit to a weighted sum of statistics from its Markov blanket (Pearl, 1988)—a natural generalization of the notion of neighborhood in undirected MRFs. This earlier work did not, however, exhibit the solutions of these equations as fixed points of a simple continuous dynamical system. In particular, Saul et al (1996) did not provide an attractor dynamics nor a Lyapunov function for their mean field equations.

In this paper, we bring this sequence of ideas full circle by forging a link between attractor dynamics and probabilistic inference for directed networks. The link is achieved via mean field theory, just as in the undirected case. In particular, we describe an attractor dynamics whose stable fixed points correspond to solutions of the mean field equations. We also establish global convergence of these dynamics by providing a Lyapunov function. Our results thus provide an understanding of feedforward (Bayesian) networks that parallels the usual understanding of symmetric (MRF) networks. In both cases, we have a satisfying semantics for the set of allowed probability distributions; in both cases, we have a mean field theory that sidesteps the intractability of exact probabilistic inference; and in both cases, we have an attractor dynamics that transforms a discrete-valued network into a continuous dynamical system.

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 introduce the probabilistic models represented by DAGs and review the problems of inference and learning. In section 3, we present the mean field theory for these networks, including the mean field equations, the attractor dynamics, and the learning rule. In section 4, we describe some experiments on a database of handwritten digits and compare our results to known benchmarks. Finally, in section 5, we present our conclusions, as well as directions for future research.

2 Probabilistic DAGs

Consider a feedforward network—or equivalently, a directed acyclic graph—in which each unit represents a binary random variable $S_i \in \{0, 1\}$ and each link corresponds to a non-zero, real-valued weight, W_{ij}. Thus W_{ij} is a *lower-diagonal* weight matrix whose zeros indicate missing links in the underlying DAG.

We can view this network as defining a probabilistic model in which missing links correspond to statements of conditional independence. In particular, suppose that the instantiations of the random variables S_i are generated by a causal process in which each unit is activated or inhibited (i.e., set to one or zero) depending on the values of its parents. This generative process is modeled by the joint distribution:

$$P(S) \;=\; \prod_i P(S_i|S_1, S_2, \ldots, S_{i-1}) \;=\; \prod_i P(S_i|\pi_{S_i}), \tag{1}$$

where π_{S_i} denotes the parents of the ith unit. Eq. (1) states that given the values of its parents, the ith unit is conditionally independent of its other ancestors in the graph. These qualitative statements of conditional independence are encoded by the structure of the graph and hold for arbitrary values of the weights W_{ij} attached to non-missing links.

The quantitative predictions of the model are determined by the conditional distributions, $P(S_i|\pi_{S_i})$, in eq. (1). In this paper, we consider *sigmoid* networks for which

$$P(S_i = 1|\pi_{S_i}) \;=\; \sigma\left(\textstyle\sum_j W_{ij} S_j\right), \tag{2}$$

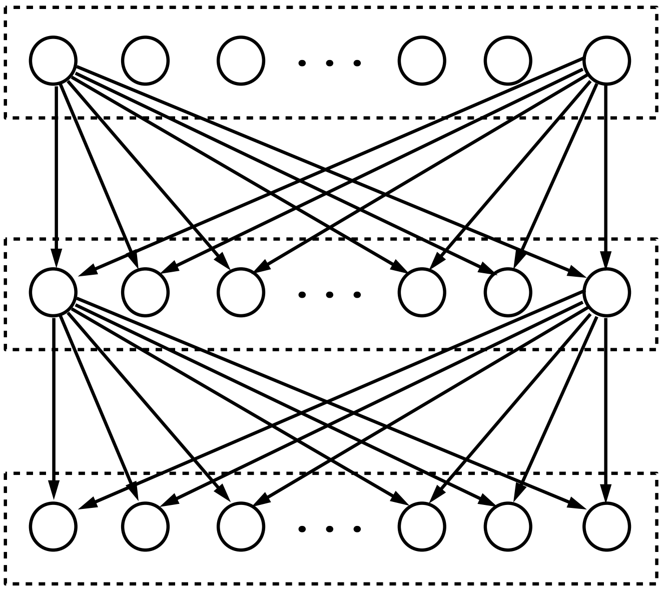

Figure 1: A layered Bayesian network parameterizes a hierarchical generative model for the data encoded by the units in its bottom layer.

where $\sigma(z) = [1 + e^{-z}]^{-1}$; thus the sign of W_{ij}, positive or negative, determines whether unit j excites or inhibits unit i in the generative process. Though we have not done so here, it is straightforward to include a bias term in the argument of the sigmoid function. Note that the sigmoid nonlinearity induces higher-order correlations between the different units in the network. In what follows, we denote by $\sigma_i = \sigma(\sum_j W_{ij} S_j)$ the squashed weighted sum on the right hand side of eq. (2); this *top-down* signal, received by each unit from its parents, can also be regarded as a random variable in its own right.

Layered networks of this form (see figure 1) were introduced as hierarchical generative models by Hinton et al (1995). In typical applications, one imagines the units in the bottom layer to encode sensory data and the units in the top layers to encode different dimensions of variability. Thus, for example, in networks for image recognition, the bottom units might encode pixel values, while the top units encode higher-order features such as orientation and occlusion. The promise of these networks lies in their ability to parameterize hierarchical, non-linear models of multiple interacting causes.

Effective use of these networks requires the ability to make probabilistic inferences. Essentially, these are queries to ascertain likely values for certain units in the network, given values—or *evidence*—for other units. Let V denote the visible units for which values are known, and H the hidden units for which values must be inferred. In principle, inferences can be made from the posterior distribution:

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

where $P(H,V)$ is the joint distribution over hidden and visible units, as given by eq. (1), and $P(V) = \sum_H P(H,V)$ is the marginal distribution obtained by summing over all configurations of hidden units. Exact probabilistic inference, however, is generally intractable in large Bayesian networks (Cooper, 1990). In particular, if there are many hidden units, then the sum to compute $P(V)$ involves an exponentially large number of terms. The same difficulty makes it impossible to compute statistics of the posterior distribution, $P(H|V)$.

Besides the problem of inference, one can also consider the problem of learning, or parameter estimation, in these networks. Unsupervised learning algorithms in probabilistic networks are designed to maximize the log-likelihood[1] of observed data. The likelihood of each data vector is given by the marginal distribution, $P(V) = \sum_H P(H,V)$. Local learning rules are derived by computing the gradients of the log-likelihood, $\ln P(V)$, with respect to the weights of the network (Neal, 1992; Binder et al, 1997). For each data vector, this gives the online update:

$$\Delta W_{ij} \propto \mathrm{E}\left[(S_i - \sigma_i)S_j\right], \tag{4}$$

where $\mathrm{E}[\cdots]$ denotes an expectation with respect to the conditional distribution, $P(H|V)$, and $\sigma_i = \sigma(\sum_j W_{ij}S_j)$ is the top-down signal from the parents of the ith unit. Note that the update takes the form of a delta rule, with the top-down signal σ_i being matched to the target value provided by S_i. Intuitively, the learning rule adjusts the weights bring each unit's expected value in line with an appropriate target value. These target values are specified explicitly by the evidence for the visible units in the network. For the other units in the network—the hidden units—appropriate target values must be computed by running an inference algorithm.

Generally speaking, in large layered networks, we can neither compute the log-likelihood $\ln P(V)$ nor the statistics of $P(H|V)$ that appear in the learning rule, eq. (4). A learning procedure can finesse these problems in two ways: (i) by optimizing the weights with respect to a more tractable cost function, or (ii) by substituting approximate values for the statistics of the hidden units. As we shall see, both strategies are employed in the mean field theory for these networks.

3 Mean field theory

Mean field theory is a general method from statistical mechanics for estimating the statistics of correlated random variables (Parisi, 1988). The name arises from physical models in which weighted sums of random variables, such as $\sum_j W_{ij}S_j$, are interpreted as local magnetic fields. Roughly speaking, under certain conditions, a central limit theorem may be applied to these sums, and a useful approximation is to ignore the fluctuations in these fields and replace them by their mean value—hence the name, "mean field" theory. More sophisticated versions of the approximation also exist, in which one incorporates the leading terms in an expansion about the mean.

The mean field approximation was originally developed for Gibbs distributions, as arise in MRFs. In this paper, we develop a mean field approximation for large layered networks whose probabilistic semantics are given by eqs. (1–2). As in MRFs, our approximation exploits the

[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.

averaging phenomena that occur at units whose conditional distributions, $P(S_i|\pi_{S_i})$, are parameterized in terms of weighted sums, such as $\sum_j W_{ij}S_j$. Addressing the twin issues of inference and learning, the approximation enables one to compute effective substitutes for the log-likelihood, $\ln P(V)$, and the statistics of the posterior distribution, $P(H|V)$.

The organization of this section is as follows. Section 3.1 describes the general approach behind the mean field approximation. Among its many interpretations, mean field theory can be viewed as a principled way of approximating an intractable probabilistic model by a tractable one. This is done via a variational principle that chooses the parameters of the tractable model to minimize an entropic measure of error. The parameters of the tractable model are known as the mean field parameters, and they serve as placeholders for the true statistics of the posterior distribution, $P(H|V)$.

Our most important result for feedforward neural networks is a compact set of equations for determining the mean field parameters. These *mean field equations* relate the statistics of each unit to those of its Markov blanket. Section 3.2 gives a succinct statement of the mean field equations, along with a number of useful intuitions. A more detailed derivation is given in the appendix.

The mean field equations are a coupled set of nonlinear equations, whose solutions cannot be expressed in closed form. Naturally, this raises the following concern: have we merely replaced one intractable problem—that of calculating averages over the posterior distribution, $P(H|V)$—by an equally intractable one—that of solving the mean field equations? In section 3.3, we show how to solve the mean field equations using an attractor dynamics. This makes it quite straightforward to solve the mean field equations, typically at much less computational cost than (say) sampling the statistics of $P(H|V)$.

Finally, in section 3.4, we present a mean field learning algorithm for these networks. Weights are adapted by a regularized delta rule that depends only on locally available information. Interestingly, the attractor dynamics for solving the mean field equations generates precisely those signals required for unsupervised learning.

3.1 A variational principle

We now return to the problem of probabilistic inference in layered feedforward networks. Our goal is to obtain the statistics of the posterior distribution, $P(H|V)$, for some full or partial instantiation V of the units in the network. Since it is generally intractable to compute these statistics exactly, we adopt the following two-step approach: (i) introduce a parameterized family of simpler distributions whose statistics are easily computed; (ii) approximate $P(H|V)$ by the member of this family which is "closest", as determined by some entropic measure of distance.

The starting point of the mean field approximation is to consider the family of *factorial* distributions:

$$Q(H|V) = \prod_{i \in H} \mu_i^{S_i}(1-\mu_i)^{1-S_i}. \tag{5}$$

The parameters μ_i represent the mean values of the hidden units under the factorial distribution, $Q(H|V)$. Note that by design, most statistics of $Q(H|V)$ are easy to compute because the distribution is factorial.

We can measure the distance between the distribution $Q(H|V)$ in eq. (5) and the true posterior distribution $P(H|V)$ by the Kullback-Leibler (KL) divergence:

$$KL(Q||P) = \sum_{H} Q(H|V) \ln \left[\frac{Q(H|V)}{P(H|V)}\right]. \tag{6}$$

The KL divergence is strictly non-negative, vanishing only if $Q(H|V) = P(H|V)$. The idea behind the mean field approximation is to find the parameters $\{\mu_i\}$ that minimize $\mathrm{KL}(Q||P)$ and then to use the statistics of $Q(H|V)$ as a substitute for the statistics of $P(H|V)$. The first step of this calculation is to rewrite the posterior distribution $P(H|V)$ using eq. (3), thus breaking the right hand side of eq. (5) into three terms:

$$KL(Q||P) = \sum_{H} Q(H|V) \ln Q(H|V) - \sum_{H} Q(H|V) \ln P(H,V) + \ln P(V). \tag{7}$$

The first two terms on the right hand side of this equation depend on properties of the approximate distribution, $Q(H|V)$: the first measures the (negative) entropy, while the second term measures the expected value of $\ln P(H,V)$. The last term in eq. (7) is simply the log-likelihood of the evidence, which—importantly—does not depend on the statistics of $Q(H|V)$. Thus this last term can be ignored when we minimize $\mathrm{KL}(Q||P)$ with respect to the parameters $\{\mu_i\}$. It nevertheless has important consequences for learning, a subject to which we return in section 3.4.

3.2 Mean field equations

The first-order statistics of $Q(H|V)$ that minimize $\mathrm{KL}(Q||P)$ naturally depend on the weights of the network, W_{ij}, and the evidence, V. This dependence is captured by the mean field equations, which are derived by evaluating and minimizing the right hand side of eq. (7). The details of this calculation, though interesting in several respects, are presented in the appendix. In what follows, we present the mean field equations as a *fait accompli* so that we may emphasize the main intuitions that emerge from the approximation of $P(H|V)$ by $Q(H|V)$.

For sigmoid DAGs, the mean field approximation works by keeping track of *two* parameters, $\{\mu_i, \xi_i\}$, for each unit in the network. Though only the first of these appears explicitly in eq. (5), it turns out that both are needed to evaluate and minimize the right hand side of eq. (7). Roughly speaking, these parameters are stored as approximations to the statistics of the true posterior distribution: in particular, $\mu_i \approx \mathrm{E}[S_i|V]$ approximates each unit's posterior mean, while $\xi_i \approx \mathrm{E}[\sigma_i|V]$ approximates the expected top-down signal in eq. (2). In some trivial cases, these statistics can be computed exactly: for visible units, $\mathrm{E}[S_i|V]$ is identically zero or one, as determined by the evidence, and for units with no parents, $\mathrm{E}[\sigma_i|V]$ is constant, independent of the evidence. More generally, though, these statistics cannot be exactly computed, and the parameters $\{\mu_i, \xi_i\}$ represent approximate values.

The values of the mean field parameters $\{\mu_i, \xi_i\}$ are computed by solving a coupled set of nonlinear equations. For large, layered networks, these mean field equations are:

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

$$\xi_i = \sigma\left[\sum_j W_{ij}\mu_j + \frac{1}{2}(1-2\xi_i)\sum_j W_{ij}^2 \mu_j(1-\mu_j)\right] \tag{9}$$

In certain cases, these equations may have multiple solutions. Roughly speaking, in these cases, each solution corresponds to the statistics of a different mode (or peak) of the posterior distribution.

The mean field equations couple the parameters of each unit to those of its parents and children. In layered networks, this amounts to a direct coupling between units in adjacent layers. The terms inside the brackets of eqs. (8–9) can be viewed as effective influences on each unit in the network. Let us examine these influences, concentrating for the moment on the leading-order terms linear in the weights, W_{ij}. In eq. (8), we see that the parameter μ_i depends on the statistics of its Markov blanket (Pearl, 1988)—i.e., on its parents through the weighted sum $\sum_j W_{ij}\mu_j$, on its children through the weighted sum $\sum_j W_{ji}\mu_j$, and on the parents of its children through the weighted sum $\sum_j W_{ji}\xi_j$. Note that this last term, $\sum_j W_{ji}\xi_j$, captures the effect of *explaining away* in which units in one layer are coupled by evidence in the layers below. In eq. (9), we see that the parameter ξ_i depends only on the statistics of its parents, with the leading dependence coming through the weighted sum $\sum_j W_{ij}\mu_j$. Thus we can interpret ξ_i as an approximation to the expected top-down signal, $\mathrm{E}[\sigma_i|V]$. The quadratic terms in eqs. (8–9), proportional to W_{ij}^2, capture higher-order corrections to the dependencies already noted. For example, in eq. (9), these terms cause any variance in the parents of unit i to push $\xi_i \approx \mathrm{E}[\sigma_i|V]$ away from the extreme values of zero or one.

The reader will note that these directed probabilistic networks have twice as many mean field parameters as their undirected counterparts. 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.

3.3 Attractor dynamics

The mean field equations provide a self-consistent description of the statistics $\mu_i \approx E[S_i|V]$ and $\xi_i \approx E[\sigma_i|V]$ in terms of the corresponding statistics for the ith unit's Markov blanket. Except in special cases, however, the solutions to these equations cannot be expressed in closed form. Thus, in general, the values for the parameters $\{\mu_i, \xi_i\}$ must be found by *numerically* solving eqs. (8–9). This is greatly facilitated by expressing the solutions to these equations as fixed points of an attractor dynamics: we can then solve the mean field equations by integrating a set of differential equations. To this end, we associate with each unit the conjugate parameters:

$$g_i = \sum_j W_{ij}\mu_j + \sum_j W_{ji}(\mu_j - \xi_j) - \frac{1}{2}(1-2\mu_i)\sum_j W_{ji}^2 \xi_j(1-\xi_j), \tag{10}$$

$$h_i = \sum_j W_{ij}\mu_j + \frac{1}{2}(1-2\xi_i)\sum_j W_{ij}^2\mu_j(1-\mu_j), \tag{11}$$

whose values are simply equal to the arguments of the sigmoid functions in the mean field equations. The variables g_i and h_i summarize the influences of the ith unit's Markov blanket. We consider the dynamics:

$$\tau_\mu \dot{\mu}_i = -\left[\mu_i - \sigma(g_i)\right], \tag{12}$$

$$\tau_h \dot{h}_i = +\left[\xi_i - \sigma(h_i)\right], \tag{13}$$

where τ_μ and τ_h are (positive) time constants and $\dot{\mu}_i$ and $\dot{h}_i$ are the time derivatives of μ_i and h_i. Note that eq. (13) specifies the time derivative of h_i, not ξ_i. As we show below, however, this does not present any difficulty in integrating the dynamics.

By construction, the fixed points of this dynamics correspond to solutions of the mean field equations. To prove the stability of these fixed points, we introduce the Lyapunov function:

$$L = \sum_{ij}\left[-W_{ij}\mu_i\mu_j + \frac{1}{2}W_{ij}^2\xi_i^2\mu_j(1-\mu_j)\right] + \sum_i\left[\int_0^{\mu_i}\sigma^{-1}(\mu)d\mu + \int_{-\infty}^{h_i}\sigma(h)dh\right], \tag{14}$$

where $\sigma^{-1}(\mu)$ is the inverse sigmoid function; i.e., $\sigma^{-1}(\sigma(h)) = h$. The first and third terms in this Lyapunov function are identical to what Hopfield (1984) considered for symmetric networks; the others are peculiar to sigmoid DAGs. Consider the time derivative of this Lyapunov function under the dynamics of eqs. (12–13). Note that this dynamics does not correspond to a strict gradient descent in L, which would trivially give rise to a proof of convergence. With some straightforward algebra, however, one can show that:

$$\dot{L} = -\sum_i\left\{\left[\sigma^{-1}(\mu_i+\tau_\mu\dot{\mu}_i) - \sigma^{-1}(\mu_i)\right]\dot{\mu}_i + \tau_h\dot{h}_i^2\right\} \leq 0, \tag{15}$$

where the inequality follows from the observation that the sigmoid function is monotonically increasing. Thus the function L always decreases under the attractor dynamics. As we discuss in the appendix, the flow to stable fixed points can be viewed as computing the approximate distribution, $Q(H|V)$, that best matches the true posterior distribution, $P(H|V)$.

In practice, one solves the mean field equations by *discretizing* the attractor dynamics in eqs. (12–13). We experimented with two simple schemes to compute updated values $\{\tilde{\mu}_i, \tilde{\xi}_i\}$ at time $t+\Delta t$ based on current values $\{\mu_i, \xi_i\}$ at time t. One of these was a first-order Euler method:

$$\tilde{\mu}_i = \mu_i + \dot{\mu}_i\Delta t, \tag{16}$$

$$\tilde{\xi}_i = \frac{1}{2} - \left[\textstyle\sum_j W_{ij}^2\tilde{\mu}_j(1-\tilde{\mu}_j)\right]^{-1}\left(h_i + \dot{h}_i\Delta t - \textstyle\sum_j W_{ij}\tilde{\mu}_j\right), \tag{17}$$

followed (when necessary) by clipping operations that projected $\{\mu_i, \xi_i\}$ into the interval $[0,1]$. The other scheme we tried was a slight variant that sidestepped the division operation in eq. (17). This was done by making additional use of the sigmoid squashing function,

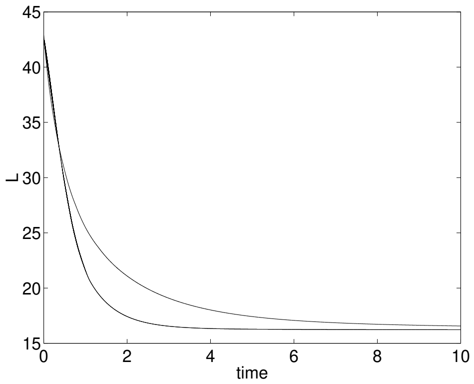

Figure 2: Typical convergence of the Lyapunov function, L, under the discretized attractor dynamics. The top curve shows the trace using eq. (17); the bottom curve, using eq. (18).

replacing eq. (17) by:

$$\tilde{\xi}_i = \sigma(h_i + \dot{h}_i \Delta t). \tag{18}$$

This second method does not strictly reduce to the continuous attractor dynamics in the limit $\Delta t \to 0$; however, empirically it tended to converge more rapidly to solutions of the mean field equations. For this reason, and also because of its naturalness, we favored this method in practice. Figure 2 shows typical traces of L versus time for both methods. The traces were computed from one of the networks learned in section 4.

3.4 Mean field learning

The Lyapunov function in eq. (14) has another interpretation that is important for unsupervised learning. Noting that the KL divergence between two distributions is strictly non-negative, it follows from eq. (7) that:

$$\ln P(V) \geq -\sum_H Q(H|V) \ln \left[\frac{Q(H|V)}{P(H,V)}\right]. \tag{19}$$

Eq. (19) gives a lower bound on the log-likelihood of the evidence, $\ln P(V)$, in terms of an average over the tractable distribution, $Q(H|V)$. In the appendix, we evaluate the right hand side of eq. (19) and show that under certain technical conditions, the Lyapunov function in eq. (14) also provides a lower bound on the log-likelihood; in particular, $\ln P(V) \geq -L$. Thus, in addition to computing approximate statistics of the posterior distribution, $P(H|V)$, the attractor dynamics in eqs. (12–13) also computes an estimate (actually, a lower bound) of the log-likelihood, $\ln P(V)$.

The lower bound on $\ln P(V)$ can be used as an objective function for unsupervised learning in directed probabilistic networks (Hinton et al, 1995). Whereas in tractable networks, one adapts the weights to maximize the log-likelihood $\ln P(V)$, as in eq. (4), in intractable networks, one adapts the weights to maximize its lower bound, $\ln P(V) \geq -L$. Note the dual role of the Lyapunov function in the mean field approximation: the attractor dynamics minimizes L with respect to the mean field parameters $\{\mu_i, \xi_i\}$, while the learning rule minimizes L with respect to the weights W_{ij}. A useful picture is to imagine these two minimizations occurring on vastly different time scales, with the mean field parameters $\{\mu_i, \xi_i\}$ tracking changes in the evidence much more rapidly than the weights, W_{ij}. Put another way, short-term memories are stored by the mean field parameters; long-term memories, by the weights.

We derive a mean field learning rule by computing the gradients of the Lyapunov function L with respect to the weights, W_{ij}. Applying the chain rule gives:

$$\frac{dL}{dW_{ij}} = \frac{\partial L}{\partial W_{ij}} + \sum_k \frac{\partial L}{\partial \mu_k}\frac{\partial \mu_k}{\partial W_{ij}} + \sum_k \frac{\partial L}{\partial \xi_k}\frac{\partial \xi_k}{\partial W_{ij}}, \tag{20}$$

where the last two terms account for the fact the mean field parameters depend implicitly on the weights through eqs. (8–9). (Here we have assumed that the attractor dynamics are allowed to converge fully before adapting the weights to new evidence.) We can simplify this expression by noting that the mean field equations describe fixed points at which $\partial L/\partial \mu_k = \partial L/\partial \xi_k = 0$; thus the last two terms in eq. (20) vanish. Evaluating the first term in eq. (20) gives rise to the online learning rule:

$$\Delta W_{ij} \propto \Big[(\mu_i - \xi_i)\,\mu_j - W_{ij}\xi_i(1-\xi_i)\mu_j(1-\mu_j)\Big]. \tag{21}$$

Comparing this learning rule to eq. (4), we see that the mean field parameters fill in for the statistics of S_i and σ_i. This is, of course, what makes the learning algorithm tractable. Whereas the statistics of $P(H|V)$ cannot be efficiently computed, the parameters $\{\mu_i, \xi_i\}$ are found by solving the mean field equations.

The reader may notice that the rightmost term of eq. (21) has no counterpart in eq. (4). This term, a regularizer induced by the mean field approximation, causes W_{ij} to be decayed according to the mean field statistics of σ_i and S_j. In particular, the weight decay is suppressed if either ξ_i or μ_j is saturated near zero or one; in effect, weights between highly correlated units are *burned in* to their current values.

4 Experimental results

We used a database of handwritten digits to evaluate the computational abilities of unsupervised neural networks represented as DAGs. The database consisted of 11000 examples of handwritten digits compiled by the U.S. Postal Service Office of Advanced Technology. The examples were preprocessed to produce 8x8 binary images, as in figure 3. For each digit, we divided the data into a training set of 700 examples and a test set of 400 examples. The partition of data into training and test sets was the same as used in previous studies (Hinton et al, 1995; Saul et al, 1996).

We used the mean field algorithm from the previous section to learn generative models of each digit class. The generative models were parameterized by three layer networks with $8 \times 24 \times 64$ architectures. In one hundred independent experiments[2], we trained ten networks,

[2]The experimental details were as follows. Each network was trained by five passes through the training

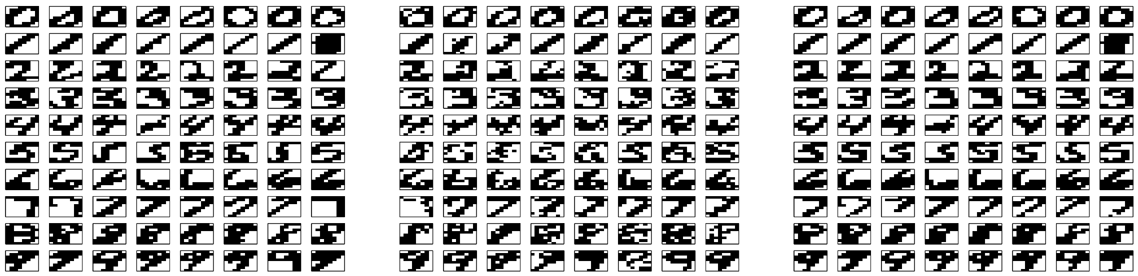

Figure 3: From left to right: actual images from the training set; images sampled from the generative models of trained networks; images whose bottom halves were inferred from their top halves.

one for each digit class, then used these networks to classify the images in the test set. The test images were labeled by whichever network returned the highest value of $-L$, used as a stand-in for the true log-likelihood, $\ln P(V)$. The mean classification error rate in these experiments was 4.4%, with a standard deviation of 0.2%. These results are considerably better than standard benchmarks on this database (Hinton et al, 1995), such as k-nearest neighbors (6.7%) and backpropagation (5.6%). They also improve slightly on results from the wake-sleep learning rule (4.8%) in Hemholtz machines (Hinton et al, 1995) and from an earlier version (4.6%) of the mean field learning rule (Saul et al, 1996).

The classification results show that the mean field networks have learned noisy but essentially accurate models of each digit class. This is confirmed visually by looking at images sampled from the generative model of each network; see figure 3. The three columns in this figure show: (i) actual images from the training set; (ii) *fantasies* sampled from the generative models of trained networks; and (iii) images whose top halves were taken from those in the first column and whose bottom halves were inferred, or *filled in*, by the attractor dynamics. These last images show that probabilistic DAGs can function as associative memories in the same way as symmetric neural networks, such as the Hopfield model (1984).

5 Discussion

In this paper we have extended the attractor paradigm of neural computation to feedforward networks parameterizing probabilistic generative models. The probabilistic semantics of these networks (Lauritzen, 1996; Neal, 1992; Pearl, 1988) differ in useful respects from those of symmetric neural networks, for which the attractor paradigm was first established (Cohen & Grossberg, 1983; Hopfield, 1982; Geman & Geman, 1984; Hinton & Sejnowski, 1986; Peterson & Anderson, 1987). Borrowing ideas from statistical mechanics, we have derived a mean field theory for approximate probabilistic inference. We have also exhibited

examples. The weights were adapted using a fixed learning rate of 0.05. Mean field parameters were computed by 16 iterations of the discretized attractor dynamics, eqs. (16) and (18), with $\tau_\mu = \tau_h = 1$ and a step size of $\Delta t = 0.25$. The mean field parameters were initialized by a top-down pass through the network, setting $\xi_i = \sigma(\sum_j W_{ij}\xi_j)$ and $\mu_i = \xi_i$ for the hidden units. The weights W_{ij} were initialized by random draws from a Gaussian distribution with zero mean and small variance.

an attractor dynamics that converges to solutions of the mean field equations and that generates the signals required for unsupervised learning.

While learning and dynamics have been twin themes of neural network research since its inception, it often appears that the field is divided into two camps—one studying symmetric networks with energy functions, the other studying feedforward networks that do not involve iterative forms of relaxation. In our view, this split has prevented researchers from combining the benefits of both approaches to computation. We note that despite the strong convergence results available for symmetric networks, there have been few applications for these networks involving any significant element of learning. Likewise, despite the powerful learning abilities of feedforward networks, there have been few applications involving more complex forms of inference and decision-making. In the remainder of this section, we discuss the many compelling reasons for combining these two approaches and suggest how this might be done using the ideas in this paper.

Let us begin by considering feedforward networks. Many practical learning algorithms have been developed for feedforward networks, and numerous theoretical results are available to characterize their properties for approximation and estimation. The usual framework for feedforward networks is one of supervised learning, or function approximation. In particular, a network induces a functional relationship between x and y based on a training set consisting of (x, y) pairs. Subsequent x inputs can be used as queries, and the network interpolates or extrapolates to provide a response y.

Although useful and general, this framework also has its limitations. In particular, it is not always the case that the form of future queries is known in advance of training, and indeed, as in the classical setting of associative memory, it can be useful to allow arbitrary components of the joint (x, y) vector to serve as queries. For example, in control and optimization applications, one would like to use y as a query and extract a corresponding x. In missing data problems, one would like to *fill in* components of the x vector given y or given other components of x. In applications involving diagnosis, model critiquing, explanation and sensitivity analysis, one would often like to find values of hidden units that correspond to particular input or output patterns. Finally, in problems with unlabeled examples, one would like to do some form of unsupervised learning. In our view, these manifold problems are best treated as general inference problems on the database of knowledge stored by the network. Moreover, as is suggested by the heuristic iterative techniques that have been employed to "invert" feedforward networks (Hoskins et al, 1992; Jordan & Rumelhart, 1992), we expect issues in dynamical systems to become relevant when inference is performed in an "upstream" direction.

Even in the classical setting where feedforward networks are used for function approximation, an inferential perspective can be useful. Consider two logistic hidden units with strong positive connections to a logistic output unit. If the output unit has a target value of one, then we can exploit the fact that only one hidden unit suffices to activate the output unit. In particular, if we have additional evidence that (say) the first hidden unit is activated, perhaps via its connection to another output unit, then we can infer that the second hidden unit is not required to be activated, and thus can be used for other purposes. This explaining-away phenomenon reflects an induced correlation between the hidden units, and it is natural in many diagnostic settings involving hidden causes (Pearl, 1988). It and other induced correlations between hidden units can be exploited if we augment our view of feedforward network learning to include an "upstream" inferential component.

While classical feedforward networks are powerful learning machines and weak inference engines, the opposite can be said of symmetric neural networks. Properly configured, symmetric networks can perform inferences as complex as solving the traveling salesman problem (Hopfield & Tank, 1986), yet few have emerged in applications involving a significant element of learning. In our view the reasons for this are twofold (Pearl, 1988). First, it is a general fact that the family of undirected graphical models—of which symmetric neural networks, such as the Boltzmann machine, are a special case—are less modular than directed graphical models. In a directed model, nodes that are downstream from the queried and observed nodes can simply be deleted; they have no effect on the query. In undirected networks no such modularity generally exists—nodes are generally coupled via the partition function. Second, in a directed network it is possible to use causal intuitions to understand representation and processing in the model. This can often be an overwhelming advantage. Moreover, if the domain being modeled has a natural causal structure, then it is natural to use a directed model which accords with the observed direction of causality.

We take two lessons from the previous successes of neural computation: (i) from the abilities of symmetric networks, that complex forms of inference require an element of iterative processing; and (ii) from the abilities of feedforward networks, that the capacity to learn is greatly enhanced by the element of directionality. We believe that the formalism in this paper combines the best aspects of symmetric and feedforward neural networks. The models we study are represented by directed acyclic graphs and thus have the natural advantages of modularity and causality that accrue to feedforward networks. Moreover, because they are endowed with probabilistic semantics, they also support complex types of inference and reasoning. This allows them to be applied to a broad range of problems involving diagnosis, explanation, control, optimization, and missing data. Our formalism also reconciles the problems of unsupervised and supervised learning in a manner reminiscent of the Boltzmann machine (Ackley et al, 1985). The supervised case simply emerges as the limiting case in which all of the input and output units are contained in the set of visible units. Finally, as in symmetric neural networks, approximate probabilistic inference is performed by relaxing a continuous dynamical system. Our formalism thus preserves the many compelling features of the attractor paradigm, including the guarantees of stability and convergence, the potential for massive parallelism, and the physical analogy of an energy surface.

Note that our analysis transforms a feedforward network into a recurrent network that possesses a Lyapunov function. This recurrent network (essentially eqs. (12–13) viewed as a recurrent network) is not a symmetric network, and its Lyapunov function does not follow directly from the theorems of Cohen and Grossberg (1983) and Hopfield (1984). We have derived the attractor dynamics for these networks by combining ideas from statistical mechanics with the probabilistic machinery of directed graphical models. Of course, one can also study recurrent networks that possess a Lyapunov function, independent of any underlying probabilistic formulation. In fact, Seung et al (1997) recently exhibited a Lyapunov function for excitatory-inhibitory neural networks with a mixture of symmetric and antisymmetric interactions. Interestingly, their Lyapunov function has a similar structure to the one in eq. (14).

A general concern with dynamical approaches to computation involves the amount of time required to relax to equilibrium. Although we found empirically that this relaxation time was not long for the problem of recognizing handwritten digits (16 iterations of the discretized differential equations), the issue requires further attention. Beyond general numerical methods for speeding convergence, one obvious approach is to consider methods for providing better initial estimates of the mean field parameters. This general idea is suggestive of the Helmholtz machine of Hinton et al (1995). The Helmholtz machine is a pair of feedforward networks, a top-down generative model that corresponds to the Bayesian network in figure 1, and a bottom-up recognition model that computes the conditional statistics of the hidden units induced by the input vector. This latter network replaces the mean field equations in our approach. The recognition model is itself learned, essentially as a probabilistic inverse to the generative model. This approach obviates the need for the iterative solution of mean field equations. The tradeoff for this simplicity is a lack of theoretical guarantees, and the fact that the recognition model cannot handle missing data or support certain types of reasoning, such as explaining away, that rely on the combination of top-down and bottom-up processing. One attractive idea, however, is to use a bottom-up recognition model to make initial guesses for the mean field parameters, then to use an attractor dynamics to refine these guesses.

Even without such enhancements, however, we feel that the attractor paradigm in directed graphical models is worthy of further investigation. Attractor neural networks have provided a viable approach to probabilistic inference in undirected graphical models (Peterson & Anderson, 1987), particularly when combined with deterministic annealing. We attribute the lack of learning-based applications for symmetric neural networks to their representational limitations for modeling causal processes (Pearl, 1988) and the peculiar instabilities arising from the sleep phase of Boltzmann learning (Neal, 1992). By combining the virtues of attractor dynamics with the probabilistic semantics of feedforward networks, we feel that a more useful and interesting model emerges.

A Details of mean field theory

In this appendix we derive the mean field approximation for large, layered networks whose probabilistic semantics are given by eqs. (1–2). Starting from the factorized distribution for $Q(H|V)$, eq. (5), our goal is to minimize the KL divergence in eq. (7), with respect to the parameters $\{\mu_i\}$. Note that this is equivalent to maximizing the lower bound on $\ln P(V)$, given in eq. (19).

The first term on the right hand side of eq. (7) is simply minus the entropy of the factorial distribution, $Q(H|V)$, or:

$$\sum_H Q(H|V)\ln Q(H|V) = \sum_i \Big[\mu_i \ln \mu_i + (1-\mu_i)\ln(1-\mu_i)\Big]. \tag{22}$$

Here, for notational convenience, we have introduced parameters μ_i for all the units in the network, hidden *and* visible. For the visible units, we use these parameters simply as placeholders for the evidence. Thus the visible units are clamped to either zero or one, and they do not contribute to the entropy in eq. (22).

Evaluating the second term on the right hand side of eq. (7) is not as straightforward as the entropy. In particular, for each unit, let

$$z_i = \sum_j W_{ij} S_j \tag{23}$$

denote its weighted sum of parents, and let $\sigma_i = \sigma(z_i)$ denote its squashed top-down signal. From eqs. (1) and (2), we can write the joint distribution in these networks as:

$$\begin{aligned} \ln P(S) &= \sum_i \left[S_i \ln \sigma_i + (1 - S_i) \ln(1 - \sigma_i)\right] && (24) \\ &= \sum_i \left(S_i z_i - \ln\left[1 + e^{z_i}\right]\right). && (25) \end{aligned}$$

Note that to evaluate the second term in eq. (7), we must average the right hand side of eq. (25) over the factorial distribution, $Q(H|V)$. The logarithm term in eq. (25), however, makes it impossible to compute this average in closed form.

Clearly, another approximation is needed to compute the expected value of $\ln[1 + e^{z_i}]$, averaged over the distribution, $Q(H|V)$. We can make progress by studying the sum of inputs, z_i, as a random variable in its own right. Under the distribution $Q(H|V)$, the right hand side of eq. (23) is a weighted sum of independent random variables with means μ_j and variances $\mu_j(1 - \mu_j)$. The number of terms in this sum is equal to the number of hiddent units in the preceding layer. In large networks, we expect the statistics of this sum—or more precisely, the distribution $Q(z_i^\ell|V)$—to be governed by a central limit theorem. In other words, to a very good approximation, $Q(z_i|V)$ assumes a normal distribution with mean and variance:

$$\begin{aligned} \langle z_i \rangle &= \sum_j W_{ij} \mu_j, && (26) \\ \left\langle \delta z_i^2 \right\rangle &= \sum_j W_{ij}^2 \mu_j (1 - \mu_j), && (27) \end{aligned}$$

where $\langle \cdot \rangle$ is used to denote the expected value. The normality of $Q(z_i|V)$ emerges in the *thermodynamic limit* of large, layered networks where each unit receives an infinite number of inputs from the hidden units in the preceding layer. In particular, suppose that unit i has N_i parents, and that the weights W_{ij} are bounded by $\sqrt{N_i}|W_{ij}| < c$ for some constant c. Then in the limit $N_i \to \infty$, the third and higher order cumulants of $\sum_j W_{ij} S_j$ vanish for any distribution under which S_j are independently distributed binary variables. The assumption that $\sqrt{N_i} W_{ij} < c$ implies that the weights are uniformly small and evenly distributed throughout the network; it is a natural assumption to make for robust, fault-tolerant networks whose computing abilities do not degrade catastrophically with random "lesions" in the weight matrix. Though only an approximation for finite networks, in what follows we make the simplifying assumption that $Q(z_i|V)$ is gaussian. This assumption—specifically tailored to large, layered networks whose evidence arrives in the bottom layer[3]—leads to the simple mean field equations and attractor dynamics in section 3.

The asymptotic form of $Q(z_i|V)$ and the logarithm term in eq. (25) 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, $\langle\ln[1+e^z]\rangle$. Then, for any real number ξ, we have the upper bound (Seung, 1995):

$$\begin{aligned} \langle\ln[1+e^z]\rangle &= \langle\ln[e^{\xi z}e^{-\xi z}(1+e^z)]\rangle, & (28)\\ &= \xi\langle z\rangle + \langle\ln[e^{-\xi z}+e^{(1-\xi)z}]\rangle, & (29)\\ &\leq \xi\langle z\rangle + \ln\langle e^{-\xi z}+e^{(1-\xi)z}\rangle, & (30) \end{aligned}$$

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:

$$\langle\ln[1+e^z]\rangle \leq \frac{1}{2}\xi^2\langle\delta z^2\rangle + \ln\left[1+e^{\langle z\rangle+(1-2\xi)\langle\delta z^2\rangle/2}\right]. \qquad (31)$$

The right hand side of eq. (31) is a convex function of ξ whose minimum occurs in the interval $\xi\in[0,1]$.

We can use this lemma to compute an approximate value for $\langle\ln[1+e^{z_i}]\rangle$, where the average is performed with respect to the distribution, $Q(H|V)$. This is done by introducing an extra parameter, ξ_i, for each unit in the network, then substituting ξ_i and the statistics of z_i into eq. (31). Note that the terms $\ln[1+e^{z_i}]$ appear in eq. (25) 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. (31) translates into a lower bound on $\langle\ln P(S)\rangle$. In particular, from eq. (25), we have:

$$\langle\ln P(S)\rangle \geq \sum_{ij}W_{ij}\mu_i\mu_j - \frac{1}{2}\sum_{ij}W_{ij}^2\xi_i^2\mu_j(1-\mu_j) - \sum_i \ln\left\{1+e^{\sum_j\left[W_{ij}\mu_j+\frac{1}{2}(1-2\xi_i)W_{ij}^2\mu_j(1-\mu_j)\right]}\right\}. \qquad (32)$$

Note that the bound in eq. (32) does not hold rigorously for finite networks; rather, it has been derived in the thermodynamic limit where $Q(z_i|V)$ is described by a normal distribution.

The objective function for the mean field approximation is the difference of eqs. (22) and (32); these expressions correspond to the first two terms of eq. (7). The difference of these two equations is in fact the Lyapunov function, L, from eq. (14). This can be shown by appealing to the definition of h_i in eq. (11) and by noting that

$$\begin{aligned} \int\sigma(h)\,dh &= \ln[1+e^h], & (33)\\ \int\sigma^{-1}(\mu)\,d\mu &= \mu\ln\mu+(1-\mu)\ln(1-\mu), & (34) \end{aligned}$$

[3]One can also proceed without making this assumption, as in Saul et al (1996), to derive approximations for non-layered networks.

where $\sigma^{-1}(\mu) = \ln\left[\frac{\mu}{1-\mu}\right]$ is the inverse sigmoid function. Thus we have derived the Lyapunov function by evaluating the KL divergence in eq. (6). It follows that the Lyapunov function measures the discrepancy between the distributions $Q(H|V)$ and $P(H|V)$ in terms of the mean field parameters, $\{\mu_i, \xi_i\}$. Optimal values for these parameters are found by minimizing L; in particular, computing the gradients $\partial L/\partial\mu_i$ and $\partial L/\partial\xi_i$ and equating them to zero leads to the mean field equations, eqs. (8–9).

Acknowledgements

We are grateful to H. Seung for many useful discussions about attractor dynamics and Lyapunov functions. We also thank Hinton et al (1995) for sharing their preprocessing software for images of handwritten digits.

References

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

Amit, D. 1989. *Modeling Brain Function.* Cambridge University Press, Cambridge.

Binder, J., Koller, D., Russell, S., and Kanazawa, K. 1997. Adaptive probabilistic networks with hidden variables. *Machine Learning* **29**, 213–244.

Buntine, W. 1994. Operations for learning with graphical models. *Journal of Artificial Intelligence Research* **2**, 159-225.

Cohen, M. and Grossberg, S. 1983. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics* **13**, 815–826.

Cooper, G. 1990. Computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence* **42**, 393-405.

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

Frey, B., Hinton, G., and Dayan, P. 1996. Does the wake-sleep algorithm produce good density estimators? In Touretzky, D., Mozer, M., and Hasselmo, M., eds. *Advances in Neural Information Processing Systems* **8**. MIT Press, Cambridge, MA.

Geman, S., and Geman, D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 721–741.

Hinton, G., Dayan, P., Frey, B., and Neal, R. 1995. The wake-sleep algorithm for unsupervised neural networks. *Science* **268**, 1158–1161.

Hopfield, J. 1984. Neurons with graded responses have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences, USA* **81**, 3088–3092.

Hopfield, J., and Tank, D. 1986. Computing with neural circuits: a model. *Science* **233**, 625–633.

Hoskins, D., Hwang, J., and Vagners, J. 1992. Iterative inversion of neural networks and its application to adaptive control, *IEEE Transactions on Neural Networks, 3*, 292–301.

Jordan, M., and Rumelhart, D. 1992. Forward models: Supervised learning with a distal teacher. *Cognitive Science, 16*, 307–354.

Koch, C., Marroquin, J., and Yuille, A. 1986. Analog "neuronal" networks in early vision. *Proceedings of the National Academy of Sciences, USA* **83**, 4263–4267.

Lauritzen, S. 1996. *Graphical Models.* Oxford University Press, Oxford.

Lewicki, M., and Sejnowski, T. 1996. Bayesian unsupervised learning of higher order structure. In Mozer, M., Jordan, M., and Petsche, T., eds. *Advances in Neural Information Processing Systems* **9**. MIT Press, Cambridge, MA.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. 1953. Equation of state calculations for fast computing machines. *Journal of Chemical Physics* **21**, 1087–1092.

Neal, R. 1992. Connectionist learning of belief networks. *Artificial Intelligence* **56**, 71–113.

Parisi, G. 1988. *Statistical Field Theory.* Addison-Wesley, Redwood City, CA.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Mateo, CA.

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

Saul, L., Jaakkola, T., and Jordan, M. 1996. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research* **4**, 61–76.

Seung, H. 1995. Annealed theories of learning. In Oh, J.-H., Kwon, C., and Cho, S., eds. *Neural Networks: The Statistical Mechanics Perspective, Proceedings of the CTP-PRSRI Joint Workshop on Theoretical Physics.* World Scientific, Singapore.

Seung, H., Richardson, T., Lagarias, J., and Hopfield, J. Minimax and Hamiltonian dynamics of excitatory-inhibitory networks. In Jordan, M., Kearns, M., and Solla, S., eds. *Advances in Neural Information Processing Systems* **10**. MIT Press, Cambridge, MA.