Chapter 11.
Stochastic Methods Rooted in Statistical Mechanics

Neural Networks and Learning Machines (Haykin)

Lecture Notes on

Self-learning Neural Algorithms

Byoung-Tak Zhang
School of Computer Science and Engineering
Seoul National University

Version: 20170926 ➔ 20170928➔20171011

Contents

11.1 Introduction 3
11.2 Statistical Mechanics 4
11.3 Markov Chains 6
11.4 Metropolis Algorithm 16
11.5 Simulated Annealing 19
11.6 Gibbs Sampling 22
11.7 Boltzmann Machine 24
11.8 Logistic Belief Nets 29
11.9 Deep Belief Nets 30
11.10 Deterministic Annealing (DA) 34
11.11 Analogy of DA with EM 39
Summary and Discussion 41

11.1 Introduction

- Statistical mechanics as a source of ideas for unsupervised (self-organized) learning systems
- Statistical mechanics
  - ✓ The formal study of macroscopic equilibrium properties of large systems of elements that are subject to the microscopic laws of mechanics.
  - ✓ The number of degrees of freedom is enormous, making the use of probabilistic methods mandatory.
  - ✓ The concept of entropy plays a vital role in statistical mechanics, as with the Shannon's information theory.
  - ✓ The more ordered the system, or the more concentrated the underlying probability distribution, the smaller the entropy will be.
- Statistical mechanics for the study of neural networks
  - ✓ Cragg and Temperley (1954) and Cowan (1968)
  - ✓ Boltzmann machine (Hinton and Sejnowsky, 1983, 1986; Ackley et al., 1985)

11.2 Statistical Mechanics (1/2)

p_i : probability of occurrence of state i of a stochastic system

$$p_i \geq 0 \text{ (for all } i\text{) and } \sum_i p_i = 1$$

E_i : energy of the system when it is in state i

In thermal equilibrium, the probability of state i is

(Canonical distribution / Gibbs distribution)

$$p_i = \frac{1}{Z}\exp\left(-\frac{E_i}{k_B T}\right)$$

$$Z = \sum_i \exp\left(-\frac{E_i}{k_B T}\right)$$

$\exp\left(-E / k_B T\right)$: Boltzmann factor

Z : sum over states (partition function)

1. States of low energy have a higher probability of occurrence than the states of high energy.
2. As the temperature T is reduced, the probability is concentrated on a smaller subset of low-energy states.

We set $k_B = 1$ and view $-\log p_i$ as "energy"

11.2 Statistical Mechanics (2/2)

Helmholtz free energy

$$F = -T \log Z$$

$$< E > \; = \sum_i p_i E_i \qquad \text{(avergage energy)}$$

$$< E > - F = -T \sum_i p_i \log p_i$$

$$H = -\sum_i p_i \log p_i \qquad \text{(entropy)}$$

Thus, we have

$$< E > - F = TH$$

$$F = \; < E > - TH$$

The free energy of the system, F, tends to decrease and become a minimum in an equilibrium situation.
The resulting probability distribution is defined by Gibbs distribution (**The Principle of Minimum Free Energy**).

Consider two systems A and A' in thermal contact.
ΔH and $\Delta H'$: entropy changes of A and A'
The total entropy tends to increase with

$$\Delta H + \Delta H' \geq 0$$

Nature likes to find a physical system with minimum free energy.

11.3 Markov Chains (1/9)

Markov property

$$P(X_{n+1} = x_{n+1} \mid X_n = x_n, \ldots, X_1 = x_1) = P(X_{n+1} = x_{n+1} \mid X_n = x_n)$$

Transition probability from state i at time n to j at time $n+1$

$$p_{ij} = P(X_{n+1} = j \mid X_n = i)$$

$$(p_{ij} \geq 0 \quad \forall i, j \text{ and } \sum_j p_{ij} = 1 \quad \forall i)$$

If the transition probabilities are fixed, the Markov chain is homogeneous.

In case of a system with a finite number of possible states K, the transition probabilities constitute a K-by-K matrix (stochastic matrix):

$$\mathbf{P} = \begin{pmatrix} p_{11} & \cdots & p_{1K} \\ \vdots & \ddots & \vdots \\ p_{K1} & \cdots & p_{KK} \end{pmatrix}$$

11.3 Markov Chains (2/9)

Generalization to *m*-step transition probability

$$p_{ij}^{(m)} = P(X_{n+m} = x_j \mid X_n = x_i), \quad m = 1,2,\ldots$$

$$\lim_{k\to\infty} v_i(k) = \pi_i \quad i = 1,2,\ldots,K$$

$$p_{ij}^{(m+1)} = \sum_k p_{ik}^{(m)} p_{kj}, \quad m = 1,2,\ldots, \quad p_{ik}^{(1)} = p_{ik}$$

We can further generalize to (Chapman-Kolmogorov identity)

$$p_{ij}^{(m+n)} = \sum_k p_{ik}^{(m)} p_{kj}^{(n)}, \quad m,n = 1,2,\ldots$$

11.3 Markov Chains (3/9)

Properties of markov chains

Recurrent

$$p_i = P(\text{every returning to state } i)$$

Transient

$$p_i < 1$$

Periodic

$$\text{If } i \in S_k \text{ and } p_i > 0, \text{ then} \begin{cases} j \in S_{k+1}, & \text{for } k = 1,\ldots,d-1 \\ j \in S_k, & \text{for } k = 1,\ldots,d \end{cases}$$

Aperiodic

Accessable: Accessable from i if there is a finite sequence of transition from i to j

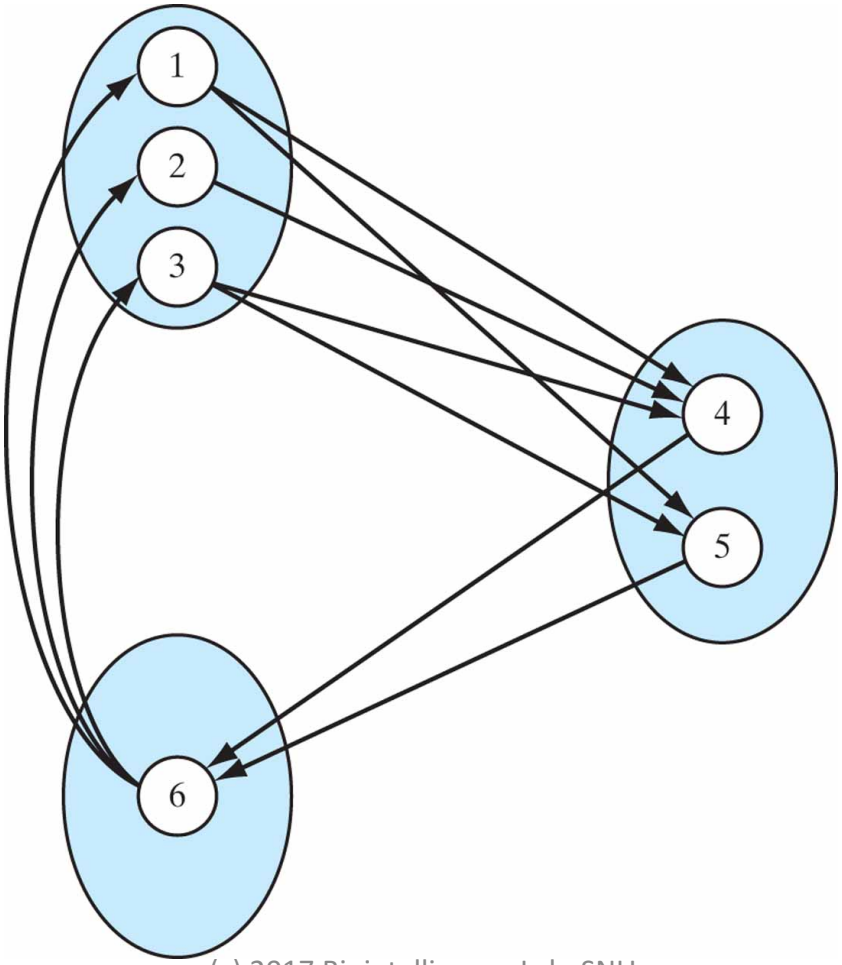
Communicate: If the states i and j are accessible to each other

If two states communicate each other, they belong to the same class.

If all the states consists of a single class, the Markov chain is indecomposible or **irreducible**.

11.3 Markov Chains (4/9)

Figure 11.1: A periodic recurrent Markov chain with d = 3.



11.3 Markov Chains (5/9)

Ergodic Markov chains

Ergodicity: time average = ensemble average

i.e. long-term proportion of time spent by the chain in state i corresponds to the steady-state probability π_i

$v_i(k)$: Proportion of time spent in state i after k returns

$$v_i(k) = \frac{k}{\sum_{\ell=1}^{k} T_i(\ell)}$$

$$\lim_{k \to \infty} v_i(k) = \pi_i \qquad i = 1, 2, \ldots, K$$

11.3 Markov Chains (6/9)

Convergence to Stationary Distributions

Consider an ergodic Markov chain with a stochastic matrix $\mathbf{P}$

$\pi^{(n-1)}$: state transition vector of the chain at time n - 1

State transition vector at time n is

$$\pi^{(n)} = \pi^{(n-1)}\mathbf{P}$$

By iteration we obtain

$$\pi^{(n)} = \pi^{(n-1)}\mathbf{P} = \pi^{(n-2)}\mathbf{P}^2 = \pi^{(n-3)}\mathbf{P}^3 = \cdots$$

$$\pi^{(n)} = \pi^{(0)}\mathbf{P}^n$$

$\pi^{(0)}$: initial value

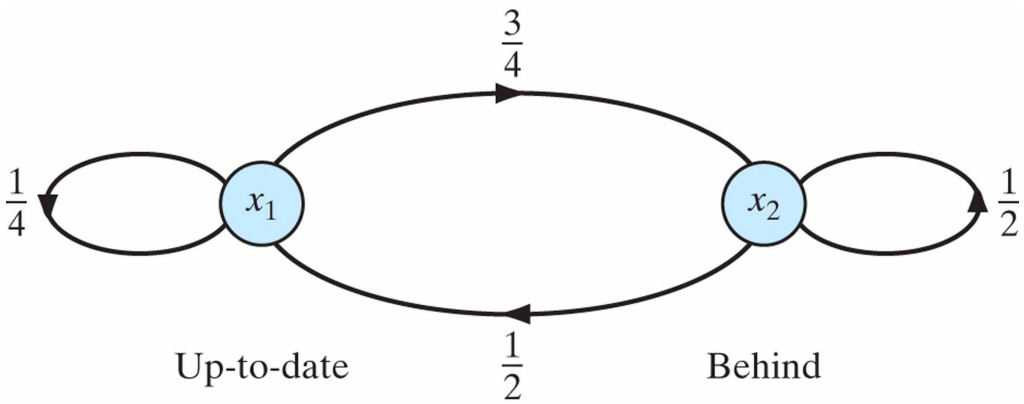
$$\lim_{n\to\infty}\mathbf{P}^n = \begin{pmatrix} \pi_1 & \cdots & \pi_K \\ \vdots & \ddots & \vdots \\ \pi_1 & \cdots & \pi_K \end{pmatrix} = \begin{pmatrix} \pi \\ \vdots \\ \pi \end{pmatrix}$$

Ergodic theorem

1. $\lim_{n\to\infty} p_{ij}^{(n)} = \pi_j \quad \forall i$
2. $\pi_j > 0 \quad \forall j$
3. $\sum_{j=1}^{K} \pi_j = 1$
4. $\pi_j = \sum_{i=1}^{K} \pi_i p_{ij} \quad \text{for } j = 1,2,\ldots,K$

11.3 Markov Chains (6/9)

Figure 11.2: State-transition diagram of Markov chain for Example 1:
The states x1 and x2 and may be identified as up-to-date behind, respectively.



$$\pi^{(0)} = \begin{bmatrix} \frac{1}{6} & \frac{5}{6} \end{bmatrix}$$

$$\pi^{(1)} = \pi^{(1)}\mathbf{P}$$

$$= \begin{bmatrix} \frac{1}{6} & \frac{5}{6} \end{bmatrix} \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{11}{24} & \frac{13}{24} \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

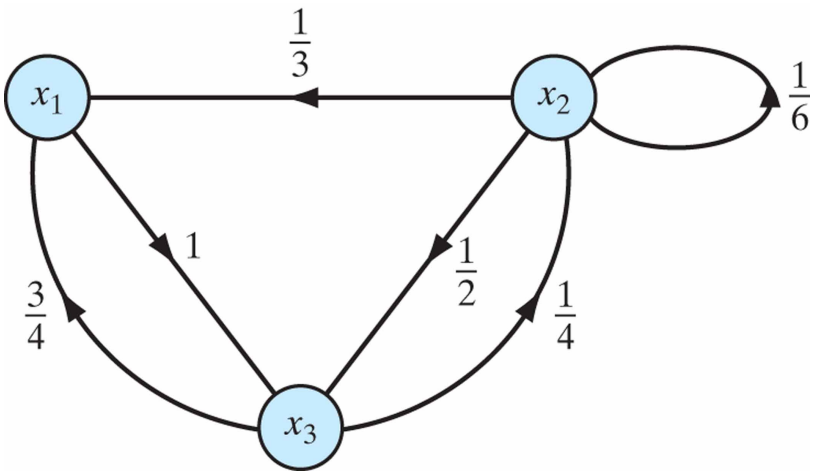
$$\mathbf{P}^{(2)} = \begin{bmatrix} 0.4375 & 0.5625 \\ 0.3750 & 0.6250 \end{bmatrix}$$

$$\mathbf{P}^{(3)} = \begin{bmatrix} 0.4001 & 0.5999 \\ 0.3999 & 0.6001 \end{bmatrix}$$

$$\mathbf{P}^{(4)} = \begin{bmatrix} 0.4000 & 0.6000 \\ 0.4000 & 0.6000 \end{bmatrix}$$

11.3 Markov Chains (7/9)

Figure 11.3: State-transition diagram of Markov chain for Example 2.



$$\pi_j = \sum_{i=1}^{K} \pi_i p_{ij}$$

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{2} \\ \frac{3}{4} & \frac{1}{4} & 0 \end{bmatrix}$$

$$\pi_1 = \frac{1}{3}\pi_2 + \frac{3}{4}\pi_3$$

$$\pi_2 = \frac{1}{6}\pi_2 + \frac{1}{4}\pi_3$$

$$\pi_3 = \pi_1 + \frac{1}{2}\pi_2$$

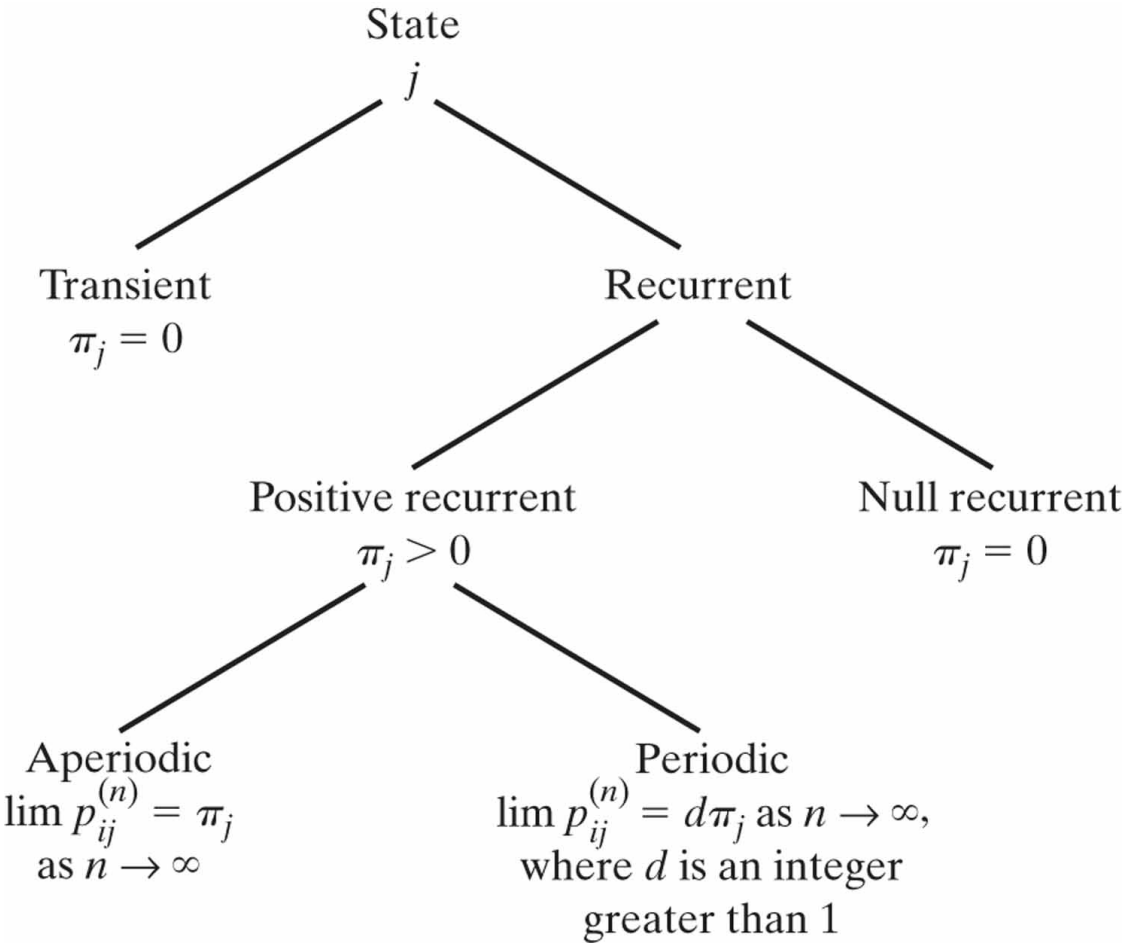
$$\pi_1 = 0.3953$$

$$\pi_2 = 0.1395$$

$$\pi_3 = 0.4652$$

11.3 Markov Chains (8/9)

Figure 11.4: Classification of the states of a Markov chain and their associated long-term behavior.



11.3 Markov Chains (9/9)

Principle of detailed balance

At thermal equilibrium, the rate of occurrence of any transition equals the corresponding rate of occurrence of the inverse transition,

$$\pi_i p_{ij} = \pi_j p_{ji}$$

Application : stationary distribution

$$\sum_{i=1}^{K} \pi_i p_{ij} = \sum_{i=1}^{K} \left(\frac{\pi_i}{\pi_j} p_{ij} \right) \pi_j = \sum_{i=1}^{K} \left(\frac{\pi_j}{\pi_j} p_{ji} \right) \pi_j$$

$$= \sum_{i=1}^{K} \left(p_{ji} \right) \pi_j \qquad (\pi_i p_{ij} = \pi_j p_{ji}, \text{detailed balance})$$

$$= \pi_j \qquad (\text{since } \sum_{i=1}^{K} p_{ji} = 1)$$

11.4 Metropolis Algorithm (1/3)

Metropolis Algorithm

A stochastic algorithm for simulating the evolution of a physical system to thermal equilibrium. A modified Monte Carlo method.

Markov Chain Monte Carlo (MCMC) method

Algorithm Metropolis

1. $X_n = x_i$. Randomly generate a new state x_j.
2. $\Delta E = E(x_j) - E(x_i)$
3. If $\Delta E < 0$, then $X_{n+1} = x_j$

   else if $\Delta E \geq 0$, then

   { Select a random number $\xi \sim \mathrm{U}[0,1]$.

   If $\xi < \exp(-\Delta E / T)$, then $X_{n+1} = x_j$, $\quad$ (*accept*)

   else $X_{n+1} = x_i$. $\quad$ (*reject*)

   }

11.4 Metropolis Algorithm (2/3)

Choice of Transition Probabilities

Proposed set of transition probabilities

1. $\tau_{ij} > 0$ (for all i, j) : Nonnegativity
2. $\sum_j \tau_{ij} = 1$ (for all i) : Normalization
3. $\tau_{ij} = \tau_{ji}$ (for all i, j) : Symmetry

Desired set of transition probabilities

$$p_{ij} = \begin{cases} \tau_{ij}\left(\dfrac{\pi_j}{\pi_i}\right) & \text{for } \dfrac{\pi_j}{\pi_i} < 1 \\ \tau_{ij} & \text{for } \dfrac{\pi_j}{\pi_i} \geq 1 \end{cases}$$

$$p_{ii} = \tau_{ii} + \sum_{j \neq i} \tau_{ij}\left(1 - \frac{\pi_j}{\pi_i}\right) = 1 - \sum_{j \neq i} \alpha_{ij}\tau_{ij}$$

Moving probability

$$\alpha_{ij} = \min\left(1, \frac{\pi_j}{\pi_i}\right)$$

11.4 Metropolis Algorithm (3/3)

How to choose the ratio π_j / π_i?

We choose the probability distribution to which we want the Markov chain to coverge to be a Gibbs distribution

$$\pi_j = \frac{1}{Z}\exp\left(-\frac{E_j}{T}\right)$$

$$\frac{\pi_j}{\pi_i} = \exp\left(-\frac{\Delta E}{T}\right)$$

$$\Delta E = E_j - E_i$$

Proof of detailed balance:

Case 1: $\Delta E < 0$.

$$\pi_i p_{ij} = \pi_i \tau_{ij} = \pi_i \tau_{ji}$$

$$\pi_j p_{ji} = \pi_j \left(\frac{\pi_i}{\pi_j}\tau_{ji}\right) = \pi_i \tau_{ji}$$

Case 2: $\Delta E > 0$.

$$\pi_i p_{ij} = \pi_i \left(\frac{\pi_j}{\pi_i}\tau_{ij}\right) = \pi_j \tau_{ji}$$

$$\pi_j p_{ji} = \pi_i \tau_{ij}$$

11.5 Simulated Annealing (1/3)

Simulated Annealing

- A stochastic relaxation technique for solving optimization problems.
- Improves the computational efficiency of the Metropolis algorithm.
- Makes random moves on the energy surface

$$F = \langle E \rangle - TH, \qquad \lim_{T \to 0} F = \langle E \rangle$$

- Operate a stochastic system at a high temperature (where convergence to equilibrium is fast) and then iteratively lower the temperature (at T=0, the Markov chain collapses on the global minima).

Two ingredients:

1. A schedule that determines the rate at which the temperature is lowered.
2. An algorithm, such as the Metropolis algorithm, that iteratively finds the equilibrium distribution at each new temperature in the schedule by using the final state of the system at the previous temperature as the starting point for the new temperature.

11.5 Simulated Annealing (2/3)

1. Initial Value of the Temperature. The initial value T_0 of the temperature is chosen high enough to ensure that virtually all proposed transitions are accepted by the simulated-annealing algorithm
2. Decrement of the Temperature. Ordinarily, the cooling is performed exponentially, and the changes made in the value of the temperature are small. In particular, the decrement function is defined by

$$T_k = \alpha T_{k-1}, \quad k = 1, 2, \ldots, K$$

   where α is a constant smaller than, but close to, unity. Typical values of α lie between 0.8 and 0.99. At each temperature, enough transitions are attempted so that there are 10 accepted transitions per experiment, on average.
3. Final Value of the Temperature. The system is fixed and annealing stops if the desired number of acceptances is not achieved at three successive temperatures

11.5 Simulated Annealing (3/3)

Simulated Annealing for Combinatorial Optimization

TABLE 11.1 Correspondence between Statistical Physics and Combinatorial Optimization

Statistical physics	Combinatorial optimization
Sample	Problem instance
State (configuration)	Configuration
Energy	Cost function
Temperature	Control parameter
Ground-state energy	Minimal cost
Ground-state configuration	Optimal configuration

11.6 Gibbs Sampling (1/2)

Gibbs sampling

An iterative adaptive scheme that generates a single value for the conditional distribution for each component of the random vector X, rather than all values of the variables at the same time.

$\mathbf{X} = X_1, X_2, ..., X_K$: a random vector of K components

Assume we know $P(X_k \mid \mathbf{X}_{-k})$, where $\mathbf{X}_{-k} = X_1, X_2, ..., X_{k-1} X_{k+1}, ..., X_K$

Gibbs sampling algorithm (Gibbs sampler)

1. Initialize $x_1(0), x_2(0), ..., x_K(0)$.
2. $i \leftarrow 1$

$x_1(1) \sim P(X_1 \mid x_2(0), x_3(0), x_4(0), ..., x_K(0))$

$x_2(1) \sim P(X_2 \mid x_1(1), x_3(0), x_4(0), ..., x_K(0))$

$x_3(1) \sim P(X_3 \mid x_1(1), x_2(1), x_3(0), ..., x_K(0))$

$\cdots$

$x_k(1) \sim P(X_k \mid x_1(1), x_2(1), ..., x_{k-1}(1), x_{k+1}(0), x_K(0))$

$\cdots$

$x_K(1) \sim P(X_K \mid x_1(1), x_2(1), ..., x_{K-1}(1))$

3. If (termination condition not met), then $i \leftarrow i+1$ and go to step 2.

11.6 Gibbs Sampling (2/2)

1. Convergence theorem. The random variable $X_k(n)$ converges in distribution to the true probability distributions of X_k for $k = 1, 2, ..., K$ as n approaches infinity; that is,

$$\lim_{n\to\infty} P(X_k^{(n)} \le x \mid x_k(0)) = P_{X_k}(x) \quad \text{for } k = 1,2,\ldots,K$$

where $P_{X_k}(x)$ is marginal cumulative distribution function of X_k.

2. Rate-of-convergence theorem. The joint cumulative distribution of the random variables $X_1(n), X_2(n), ..., X_K(n)$ converges to the true joint cumulative distribution of $X_1, X_2, ..., X_K$ at a geometric rate in n.

3. Ergodic theorem. For any measurable function g of the random variables $X_1, X_2, ..., X_K$ whose expectation exists, we have

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} g(X_1(i), X_2(i), \ldots, X_K(i)) \to \mathbf{E}[g(X_1, X_2, \ldots X_K)]$$

with probability 1 (i.e., almost surely).

11.7 Boltzmann Machine (1/5)

A stochastic machine consisting of stochastic neurons with symmetric synaptic connections.

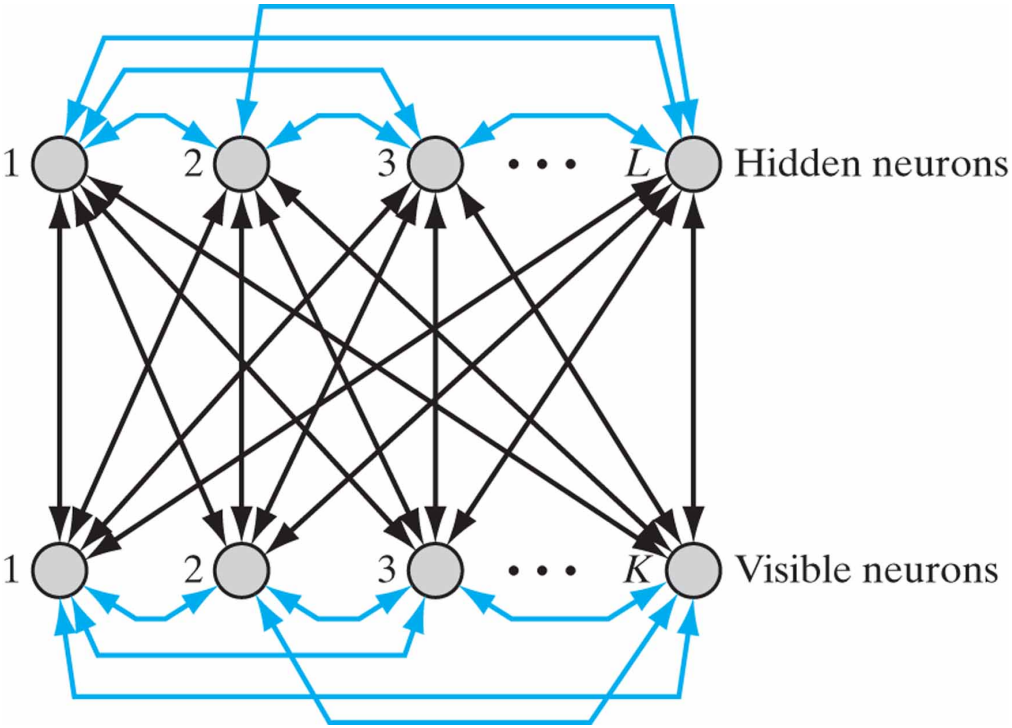


Figure 11.5: Architectural graph of Boltzmann machine; K is the number of visible neurons, and L is the number of hidden neurons. The distinguishing features of the machine are:
1. The connections between the visible and hidden neurons are symmetric.
2. The symmetric connections are extended to the visible and hidden neurons.

Boltzmann machine (BM)

$\mathbf{x}$: state vector of BM

w_{ji} : synaptic connection from i to j

Structure (weights)

$$w_{ji} = w_{ij} \quad \forall i, j$$

$$w_{ii} = 0 \quad \forall i$$

Energy

$$E(\mathbf{x}) = -\frac{1}{2}\sum_i \sum_{j \neq i} w_{ji} x_i x_j$$

Probability

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z}\exp\left(-\frac{E(\mathbf{x})}{T}\right)$$

11.7 Boltzmann Machine (2/5)

Consider three events:

$$A : X_j = x_j$$

$$B : \left\{ X_i = x_i \right\}_{i=1}^{K} \text{ with } i \neq j$$

$$C : \left\{ X_i = x_i \right\}_{i=1}^{K}$$

The joint event B excludes A, and the joint event C includes both A and B.

$$P(C) = P(A, B)$$

$$= \frac{1}{Z} \exp\left(\frac{1}{2T} \sum_i \sum_{j, j \neq i} w_{ji} x_i x_j \right)$$

$$P(B) = \sum_A P(A, B)$$

$$= \frac{1}{Z} \sum_{x_j} \exp\left(\frac{1}{2T} \sum_i \sum_{j, j \neq i} w_{ji} x_i x_j \right)$$

The component involving x_j

$$\frac{x_j}{2T} \sum_{i \neq j} w_{ji} x_i$$

$$P(A \mid B) = \frac{P(A, B)}{P(B)}$$

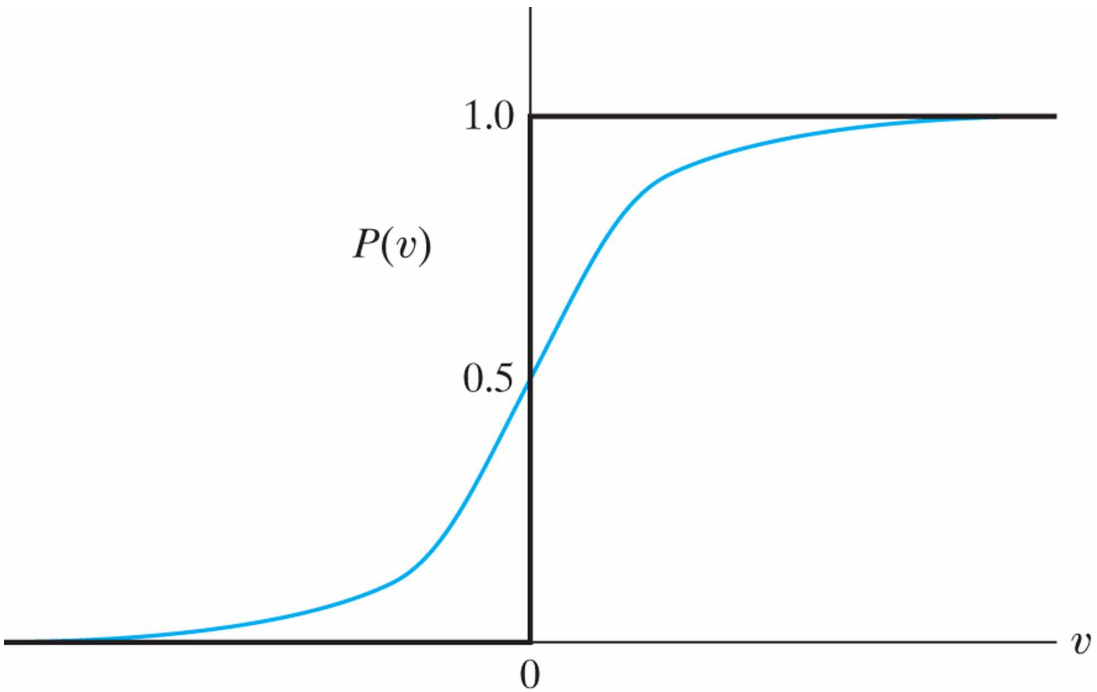
$$= \frac{1}{1 + \exp\left(-\frac{x_j}{T} \sum_{\substack{i \\ i \neq j}} w_{ji} x_i \right)}$$

$$P\left(X_j = x \mid \left\{ X_i = x_i \right\}_{i=1, i \neq j}^{K} \right) = \varphi\left(\frac{x}{T} \sum_{i, i \neq j}^{K} w_{ji} x_i \right)$$

$$\varphi(v) = \frac{1}{1 + \exp(-v)}$$

11.7 Boltzmann Machine (3/5)

Figure 11.6: Sigmoid-shaped function $P(v)$.



1. **Positive phase.** In this phase, the network operates in its clamped condition (i.e.,under the direct influence of the training sample $\mathfrak{I}$).
2. **Negative phase.** In this second phase, the network is allowed to run freely, and therefore with no environmental input.

$$L(\mathbf{w}) = \log \prod_{\mathbf{x}_\alpha \in \mathfrak{I}} P(\mathbf{X}_\alpha = \mathbf{x}_\alpha)$$
$$= \sum_{\mathbf{x}_\alpha \in \mathfrak{I}} \log P(\mathbf{X}_\alpha = \mathbf{x}_\alpha)$$

11.7 Boltzmann Machine (4/5)

$\mathbf{x}_\alpha$: the state of the visible neurons (subset of $\mathbf{x}$)

$\mathbf{x}_\beta$: the state of the hidden neurons (subset of $\mathbf{x}$)

Probability of the visible state

$$P(\mathbf{X}_\alpha = \mathbf{x}_\alpha) = \frac{1}{Z}\sum_{\mathbf{x}_\beta} \exp\left(-\frac{E(\mathbf{x})}{T}\right)$$

$$Z = \sum_{\mathbf{x}} \exp\left(-\frac{E(\mathbf{x})}{T}\right)$$

Log-likelihood function given the training data $\Im$

$$L(\mathbf{w}) = \log P(\mathbf{x} \mid \mathbf{w}) = \sum_{\mathbf{x}_\alpha \in \Im}\left(\log \sum_{\mathbf{x}_\beta} \exp\left(-\frac{E(\mathbf{x})}{T}\right) - \log \sum_{\mathbf{x}} \exp\left(-\frac{E(\mathbf{x})}{T}\right)\right)$$

Derivative of the log-likelihood function

$$\frac{\partial L(\mathbf{w})}{\partial w_{ji}} = \frac{1}{T}\sum_{\mathbf{x}_\alpha \in \Im}\left(\sum_{\mathbf{x}_\beta} P(\mathbf{X}_\beta = \mathbf{x}_\beta \mid \mathbf{X}_\alpha = \mathbf{x}_\alpha) x_j x_i - \sum_{\mathbf{x}} P(\mathbf{X} = \mathbf{x}) x_j x_i\right)$$

11.7 Boltzmann Machine (5/5)

Mean firing rate in the positive phase (clamped)

$$\rho_{ji}^{+} = \langle x_j x_i \rangle^{+} = \sum_{\mathbf{x}_\alpha \in \Im} \sum_{\mathbf{x}_\beta} P(\mathbf{X} = \mathbf{x}_\beta \mid \mathbf{X} = \mathbf{x}_\alpha) x_j x_i$$

Mean firing rate in the negative phase (free-running)

$$\rho_{ji}^{-} = \langle x_j x_i \rangle^{-} = \sum_{\mathbf{x}_\alpha \in \Im} \sum_{\mathbf{x}} P(\mathbf{X} = \mathbf{x}) x_j x_i$$

Thus, we may write

$$\frac{\partial L(\mathbf{w})}{\partial w_{ji}} = \frac{1}{T}(\rho_{ji}^{+} - \rho_{ji}^{-})$$

Gradient ascent to maximize the $L(\mathbf{w})$

$$\Delta w_{ji} = \eta \frac{\partial L(\mathbf{w})}{\partial w_{ji}} = \eta'(\rho_{ji}^{+} - \rho_{ji}^{-})$$

Boltzmann machine learning rule

$$\eta' = \frac{\varepsilon}{T}$$

11.8 Logistic Belief Nets

A stochastic machine consisting of multiple layers of stochastic neurons with directed synaptic connections.

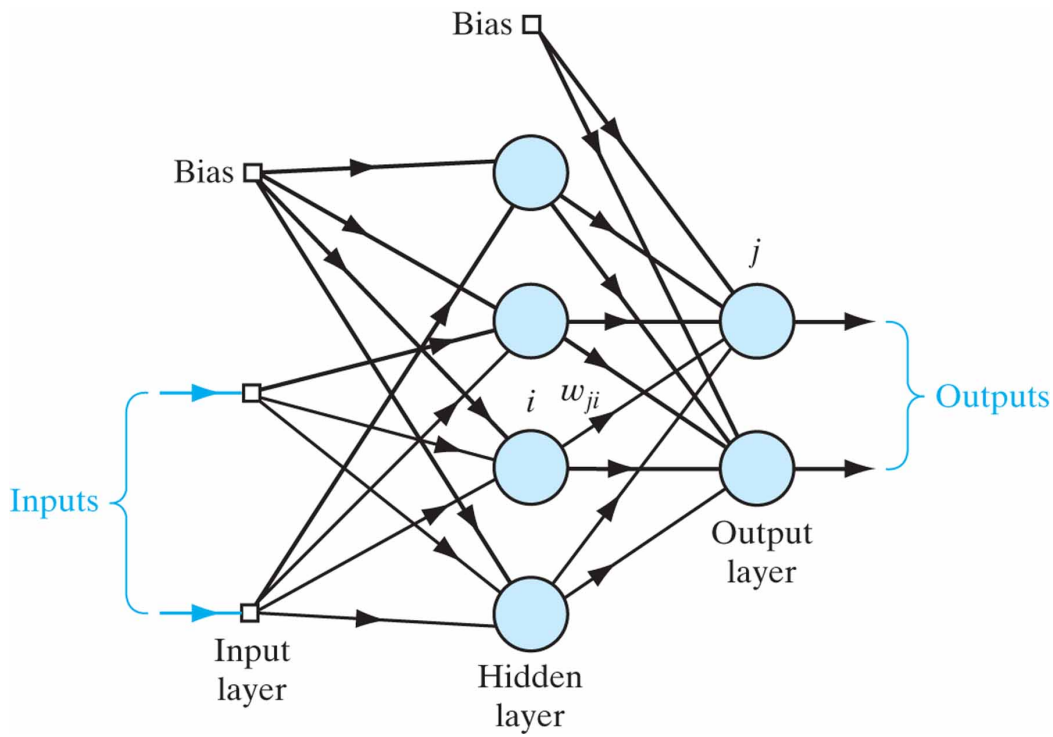


Figure 11.7: Directed (logistic) belief network.

Parents of node j

$$pa(X_j) \subseteq \{X_1, X_2, \ldots, X_{j-1}\}$$

Conditional probability

$$P(X_j = x_j \mid X_1 = x_1, \ldots, X_{j-1} = x_{j-1})$$
$$= P(X_j = x_j \mid pa(X_j))$$

Calculation of conditional probabilities

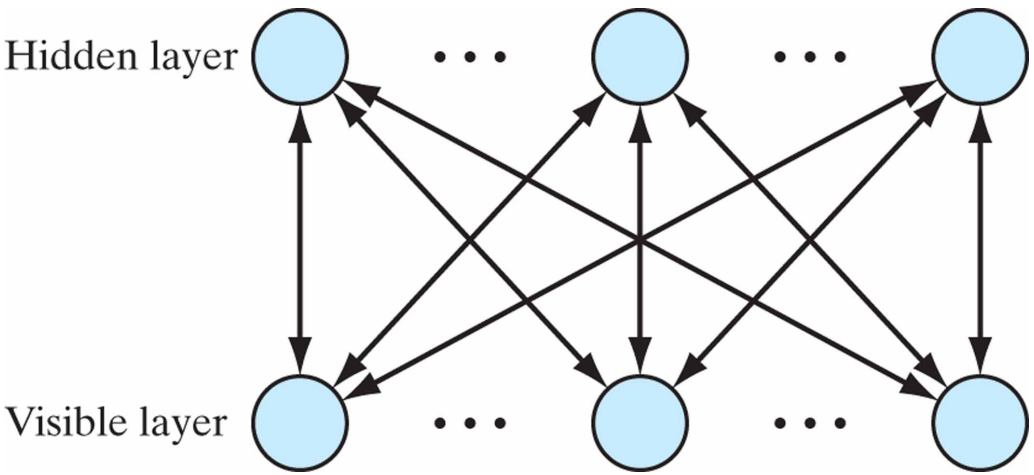
1. $w_{ji} = 0$ for all $X_i \notin pa(X_j)$
2. $w_{ji} = 0$ for $i \geq j$ ($\because$ acyclic)

Weight update rule

$$\Delta w_{ji} = \eta \frac{\partial}{\partial w_{ji}} L(\mathbf{w})$$

11.9 Deep Belief Nets (1/4)

Maximum-Likelihood Learning in a Restricted Boltzmann Machine (RBM)



Sequential pre - training

1. Update the hidden states **h** in parallel, given the visible states **x**.
2. Doing the same, but in reverse: update the visible states **x** in parallel, given the hidden states **h**.

Maximum - likelihood learning

$$\frac{\partial L(\mathbf{w})}{\partial w_{ji}} = \rho_{ji}^{(0)} - \rho_{ji}^{(\infty)}$$

Figure 11.8: Neural structure of restricted Boltzmann machine (RBM). Contrasting this with that of Fig. 11.6, we see that unlike the Boltzmann machine, there are no connections among the visible neurons and the hidden neurons in the RBM.

11.9 Deep Belief Nets (2/4)

Figure 11.9: Top-down learning, using logistic belief network of infinite depth.

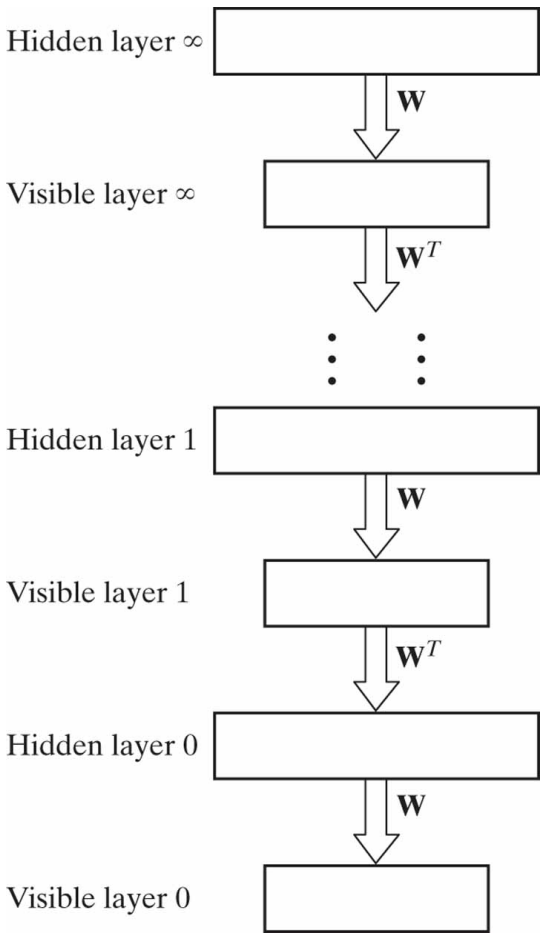
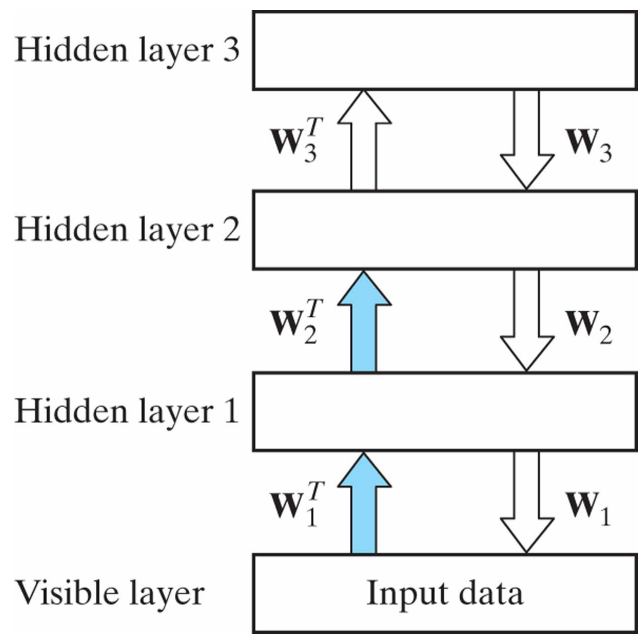
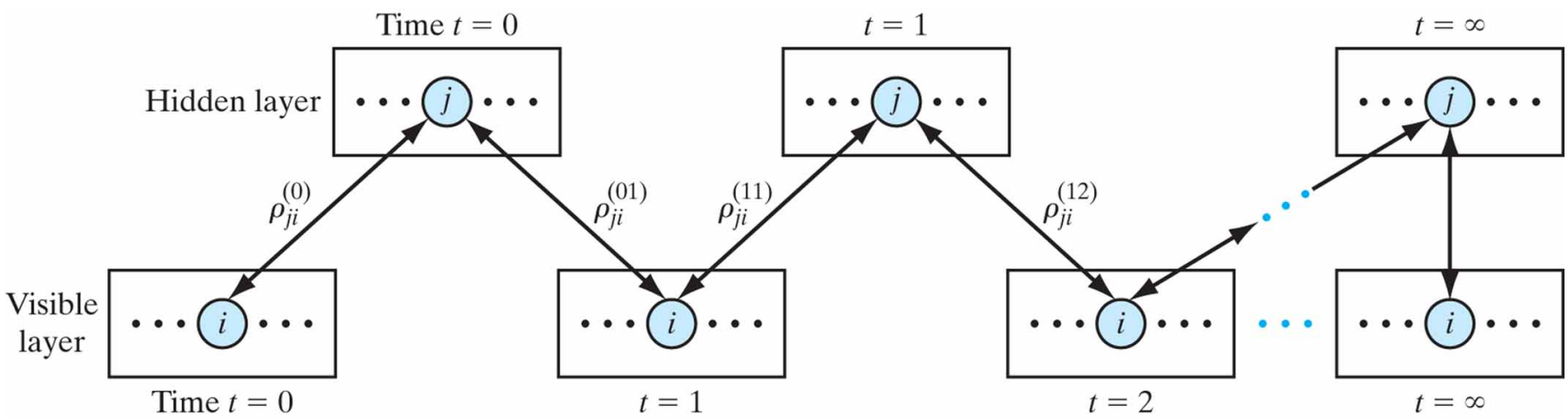


Figure 11.10: A hybrid generative model in which the two top layers form a restricted Boltzmann machine and the lower two layers form a directed model. The weights shown with blue shaded arrows are not part of the generative model; they are used to infer the feature values given to the data, but they are not used for generating data.



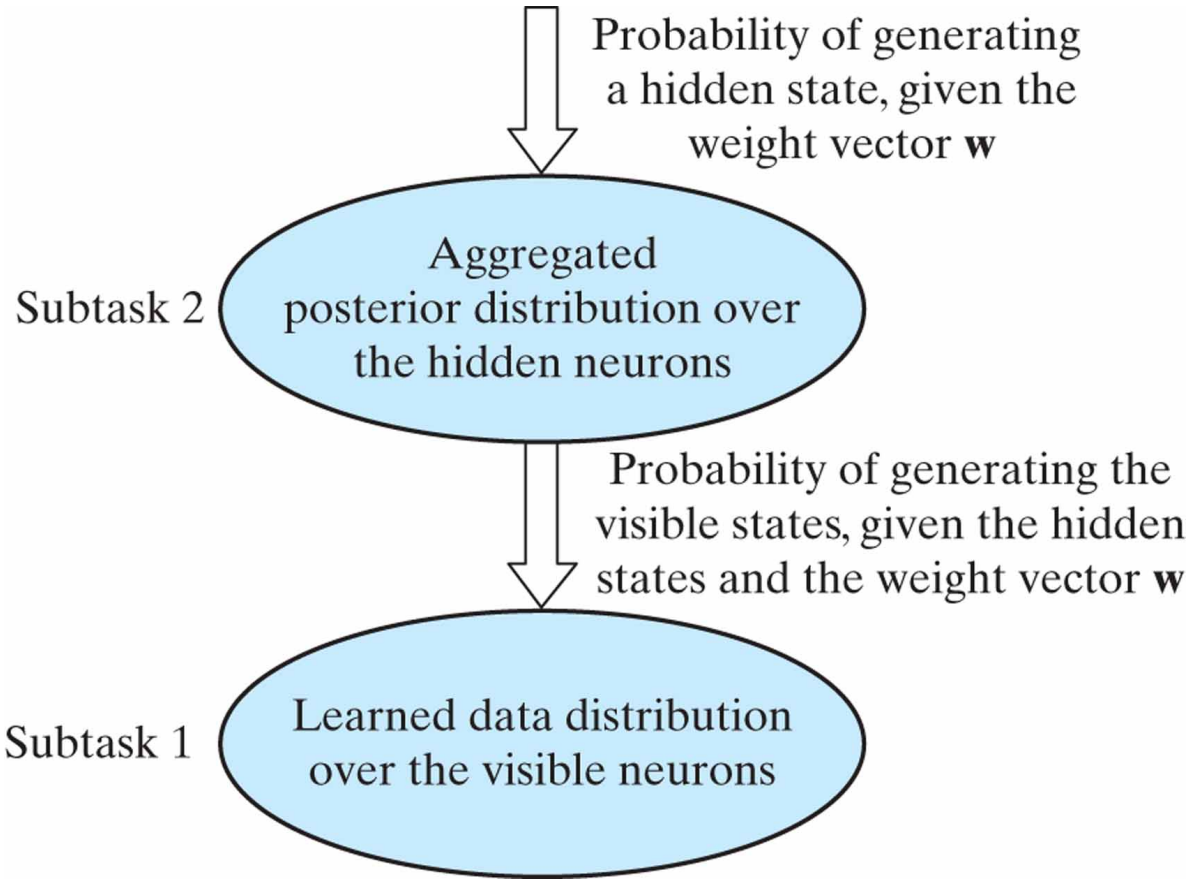
11.9 Deep Belief Nets (3/4)

Figure 11.11: Illustrating the **progression of alternating Gibbs sampling** in an RBM. After sufficiently many steps, the visible and hidden vectors are sampled from the stationary distribution defined by the current parameters of the model.



11.9 Deep Belief Nets (4/4)

Figure 11.12: The task of **modeling the sensory (visible) data** is divided into two subtasks.



11.10 Deterministic Annealing (1/5)

Deterministic Annealing

Incorporates randomness into the energy function, which is then deterministically optimized at a sequence of decreasing temperature (cf. simulated annealing: random moves on the energy surface)

Clustering via Deterministic Annealing

$\mathbf{x}$: source (input) vector

$\mathbf{y}$: reconstruction (output) vector

Distortion measure: $d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^2$

Expected distortion: $D = \sum_{\mathbf{x}} \sum_{\mathbf{y}} P(\mathbf{X}=\mathbf{x}, \mathbf{Y}=\mathbf{y}) d(\mathbf{x},\mathbf{y})$

$$= \sum_{\mathbf{x}} P(\mathbf{X}=\mathbf{x}) \sum_{\mathbf{y}} P(\mathbf{Y}=\mathbf{y} \mid \mathbf{X}=\mathbf{x}) d(\mathbf{x},\mathbf{y})$$

Probability of joint event

$$P(\mathbf{X}=\mathbf{x}, \mathbf{Y}=\mathbf{y}) = \underbrace{P(\mathbf{Y}=\mathbf{y} \mid \mathbf{X}=\mathbf{x})}_{\text{association probability}} P(\mathbf{X}=\mathbf{x})$$

11.10 Deterministic Annealing (2/5)

Table 11.2

TABLE 11.2 Correspondence between Constrained-Clustering Optimization and Statistical Physics

Constrained-clustering optimization	Statistical physics
Lagrangian, F	Free energy, F
Expected distortion, D	Average energy, $<E>$
Joint entropy in the Shannon sense, H	Entropy, H
Lagrange multiplier, T	Temperature, T

Entropy as randomness measure

$$H(\mathbf{X},\mathbf{Y}) = -\sum_{\mathbf{x}}\sum_{\mathbf{y}} P(\mathbf{X}=\mathbf{x},\mathbf{Y}=\mathbf{y})\log P(\mathbf{X}=\mathbf{x},\mathbf{Y}=\mathbf{y})$$

Constrained optimization of D as minimization of the Lagrangean

$$F = D - TH$$

$$H(\mathbf{X},\mathbf{Y}) = \underbrace{H(\mathbf{X})}_{\text{source entropy}} + \underbrace{H(\mathbf{Y}\,|\,\mathbf{X})}_{\text{conditional entropy}}$$

$$H(\mathbf{Y}\,|\,\mathbf{X}) = -\sum_{\mathbf{x}} P(\mathbf{X}=\mathbf{x})\sum_{\mathbf{y}} P(\mathbf{Y}=\mathbf{y}\,|\,\mathbf{X}=\mathbf{x})\log P(\mathbf{Y}=\mathbf{y}\,|\,\mathbf{X}=\mathbf{x})$$

$$P(\mathbf{Y}=\mathbf{y}\,|\,\mathbf{X}=\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}}\exp\left(-\frac{d(\mathbf{x},\mathbf{y})}{T}\right), \qquad Z_{\mathbf{x}} = \sum_{\mathbf{y}}\exp\left(-\frac{d(\mathbf{x},\mathbf{y})}{T}\right)$$

11.10 Deterministic Annealing (3/5)

$$F^* = \min_{P(\mathbf{Y}=\mathbf{y}|\mathbf{X}=\mathbf{x})} F$$
$$= -T\sum_{\mathbf{x}} P(\mathbf{X}=\mathbf{x})\log Z_{\mathbf{x}}$$

Setting

$$\frac{\partial}{\partial \mathbf{y}} F^* = \sum_{\mathbf{x}} P(\mathbf{X}=\mathbf{x},\mathbf{Y}=\mathbf{y})\frac{\partial}{\partial \mathbf{y}} d(\mathbf{x},\mathbf{y}) = 0 \quad \forall \mathbf{y} \in \Upsilon$$

The minimizing condition is

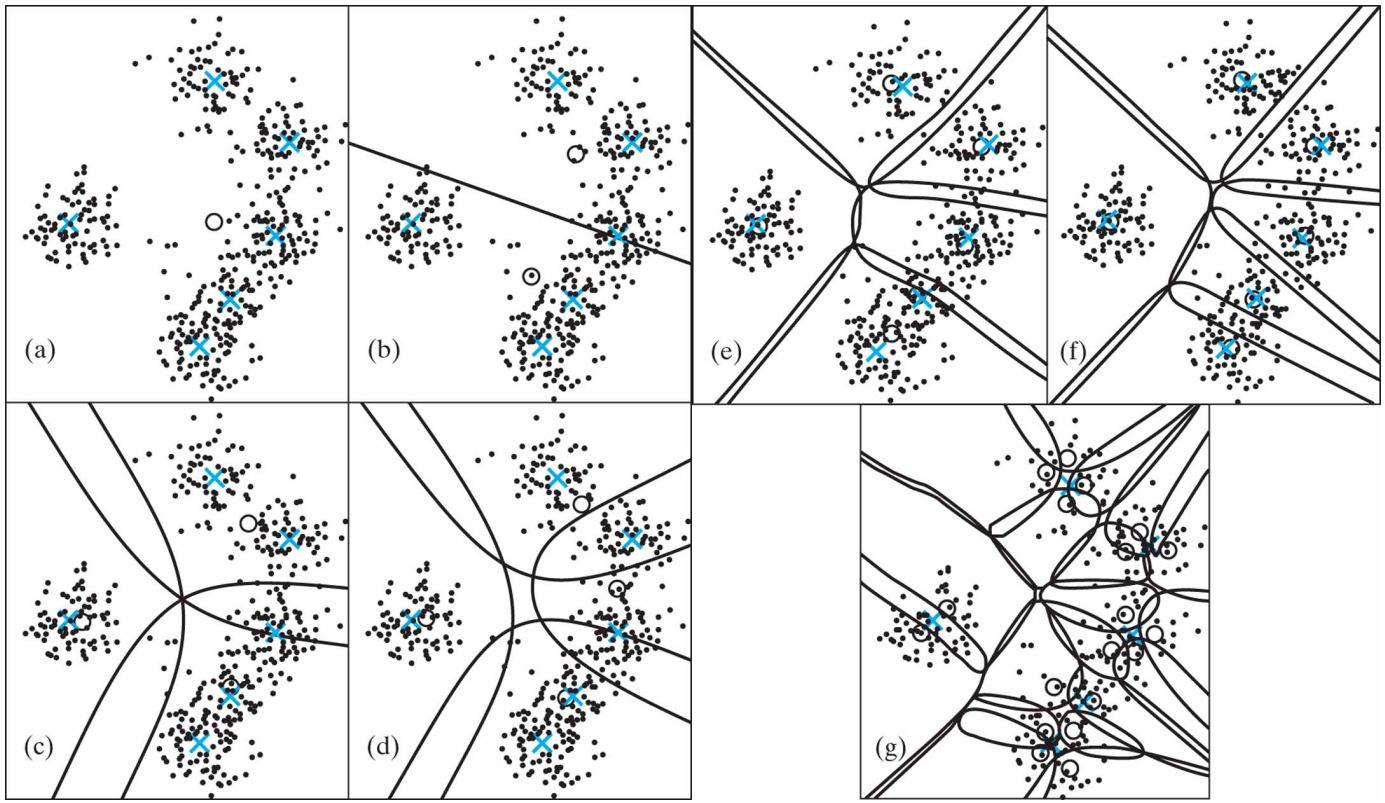
$$\frac{1}{N}\sum_{\mathbf{x}} P(\mathbf{Y}=\mathbf{y}\,|\,\mathbf{X}=\mathbf{x})\frac{\partial}{\partial \mathbf{y}} d(\mathbf{x},\mathbf{y}) = 0 \quad \forall \mathbf{y} \in \Upsilon$$

The deterministic annealing algorithm consists of minimizing the Lagrangian F^* with respect to the code vectors at a high value of temperature T and then tracking the minimum while the temperature T is lowered.

11.10 Deterministic Annealing (4/5)

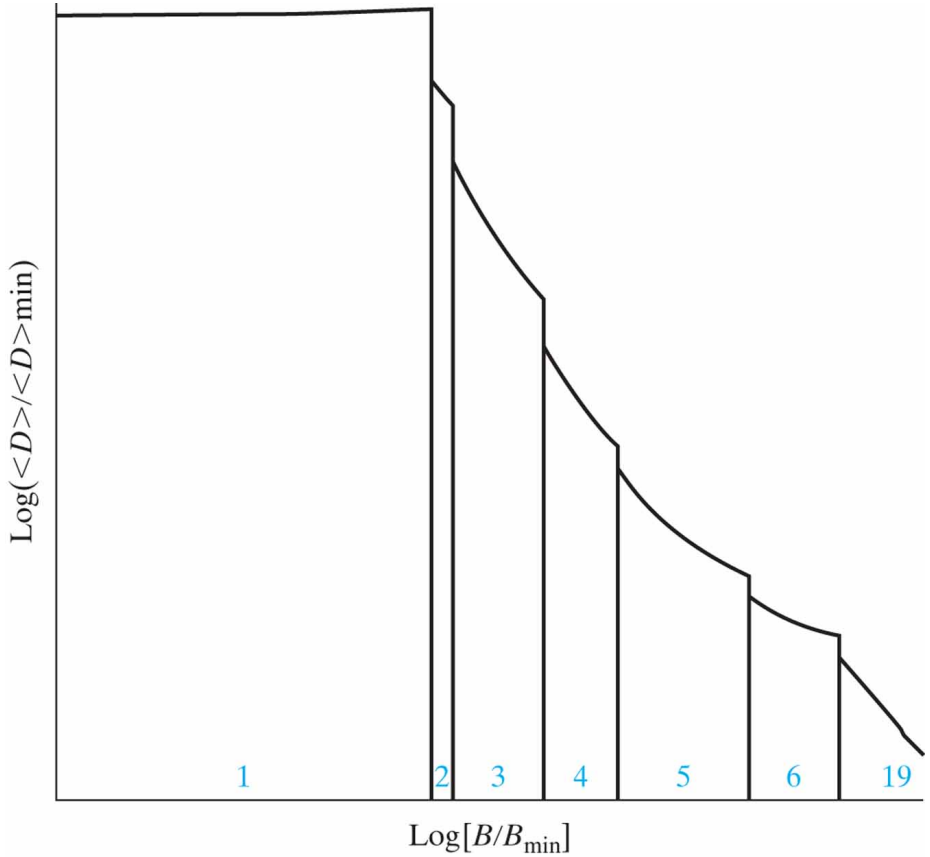
Figure 11.13: Clustering at various phases. The lines are equiprobability contours, p = ½ in (b), and p = ⅓ elsewhere:
(a) 1 cluster (B = 0),
(b) 2 clusters (B = 0.0049),
(c) 3 clusters (B = 0.0056),
(d) 4 clusters (B = 0.0100),
(e) 5 clusters (B = 0.0156),
(f) 6 clusters (B = 0.0347), and
(g) 19 clusters (B = 0.0605).

$$B = \frac{1}{T}$$



11.10 Deterministic Annealing (5/5)

Figure 11.14: Phase diagram for the Case Study in deterministic annealing. The number of effective clusters is shown for each phase.



$$B = \frac{1}{T}$$

11.11 Analogy of DA with EM (1/2)

Suppose we view the association probability $P(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{x})$ as the expected value of a random binary variable $V_{\mathbf{xy}}$ defined as

$$V_{\mathbf{xy}} \begin{cases} 1 & \text{if the source vector } \mathbf{x} \text{ is assigned to code vector } \mathbf{y} \\ 0 & \text{otherwise} \end{cases}$$

Then, the two steps of DA = two steps of EM

1. Step 1 of DA (= E-step of EM)

   Compute the association probabilities $P(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{x})$

2. Step 2 of DA (= M-step of EM)

   Optimize the distortion measure $d(\mathbf{x}, \mathbf{y})$

11.11 Analogy of DA with EM (2/2)

$\mathbf{r}$: complete data including missing data $\mathbf{z}$

$d = d(\mathbf{r})$: incomplete data

Conditional pdf of $\mathbf{r}$ given param vector $\boldsymbol{\theta}$

$$p_D(d \mid \boldsymbol{\theta}) = \int_{\Re(d)} p_c(\mathbf{r} \mid \boldsymbol{\theta})\, d\mathbf{r}$$

$\Re(d)$: subspace of $\Re$ determined by $d = d(\mathbf{r})$

Incomplete log-likelihood function

$$L(\boldsymbol{\theta}) = \log p_D(d \mid \boldsymbol{\theta})$$

Complete-data log-likelihood function

$$L_c(\boldsymbol{\theta}) = \log p_c(\mathbf{r} \mid \boldsymbol{\theta})$$

Expectation - Maximization Algorithm

$\hat{\boldsymbol{\theta}}(n)$: value of $\boldsymbol{\theta}$ at iteration n of EM

1. E-step

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n)) = \mathbf{E}_{\hat{\boldsymbol{\theta}}(n)}\left[L_C(\boldsymbol{\theta})\right]$$

2. M-step

$$\hat{\boldsymbol{\theta}}(n+1) = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}(n))$$

After an interation of the EM algorithm, the incomplete-data log-likelihood function is not decreased:

$$L(\hat{\boldsymbol{\theta}}(n+1) \geq L(\hat{\boldsymbol{\theta}}(n)) \quad \text{for } n = 0,1,2,\ldots,K$$

Summary and Discussion

- Statistical mechanics as mathematical basis for the formulation of stochastic simulation / optimization / learning
  1. Metropolis algorithm
  2. Simulated annealing
  3. Gibbs sampling
- Stochastic learning machines
  1. (Classical) Boltzmann machine
  2. Restricted Boltzmann machine (RBM)
  3. Deep belief nets (DBN)
- Deterministic annealing (DA)
  1. For optimization: Connection to simulated annealing (SA)
  2. For clustering: Connection to expectation-maximization (EM)