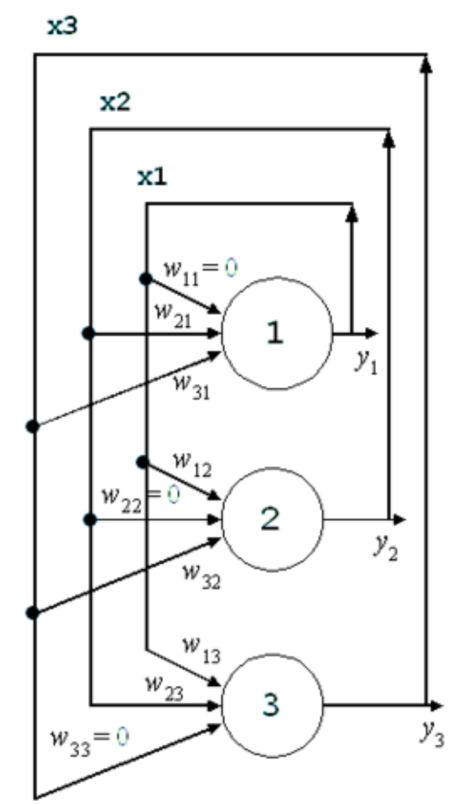
CS 7140:
ADVANCED MACHINE LEARNING

Recap:Binary Hopfield Network

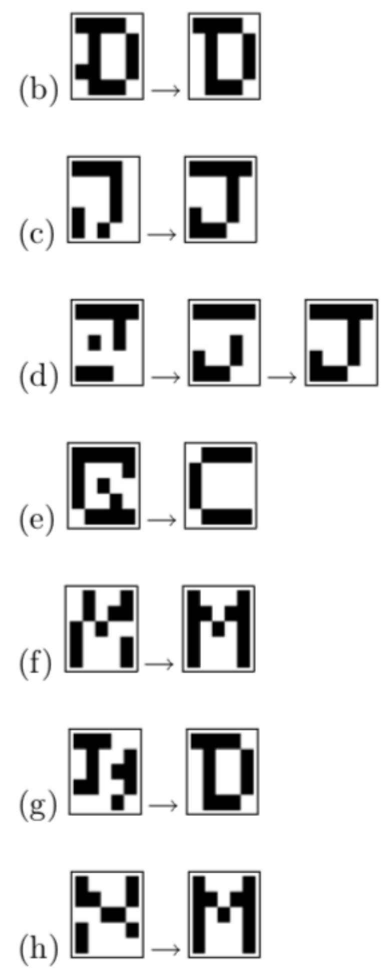
- **Weights** w_{ij} denotes the connection from neuron i to neuron j
- **Architecture** symmetric, bidirectional connections $w_{ij} = w_{ji}$, no self-connections $w_{ii} = 0$
- **Activity rule** single neuron update
$$x(a) = \begin{cases} 1 & a \geq 0 \\ -1 & a < 0 \end{cases}$$
- **Updates** activations $a_i = \sum_j w_{ij} x_j$

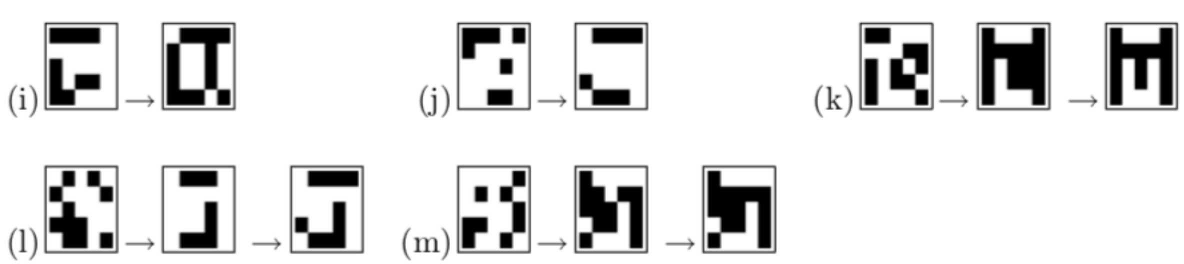


- **Learning rule** make a set of desired memory $\{x^{(n)}\}$ be stable states, set weights $w_{ij} = \eta \sum_{ij} x_i^{(n)} x_j^{(n)}$

Recap:Associative Memory

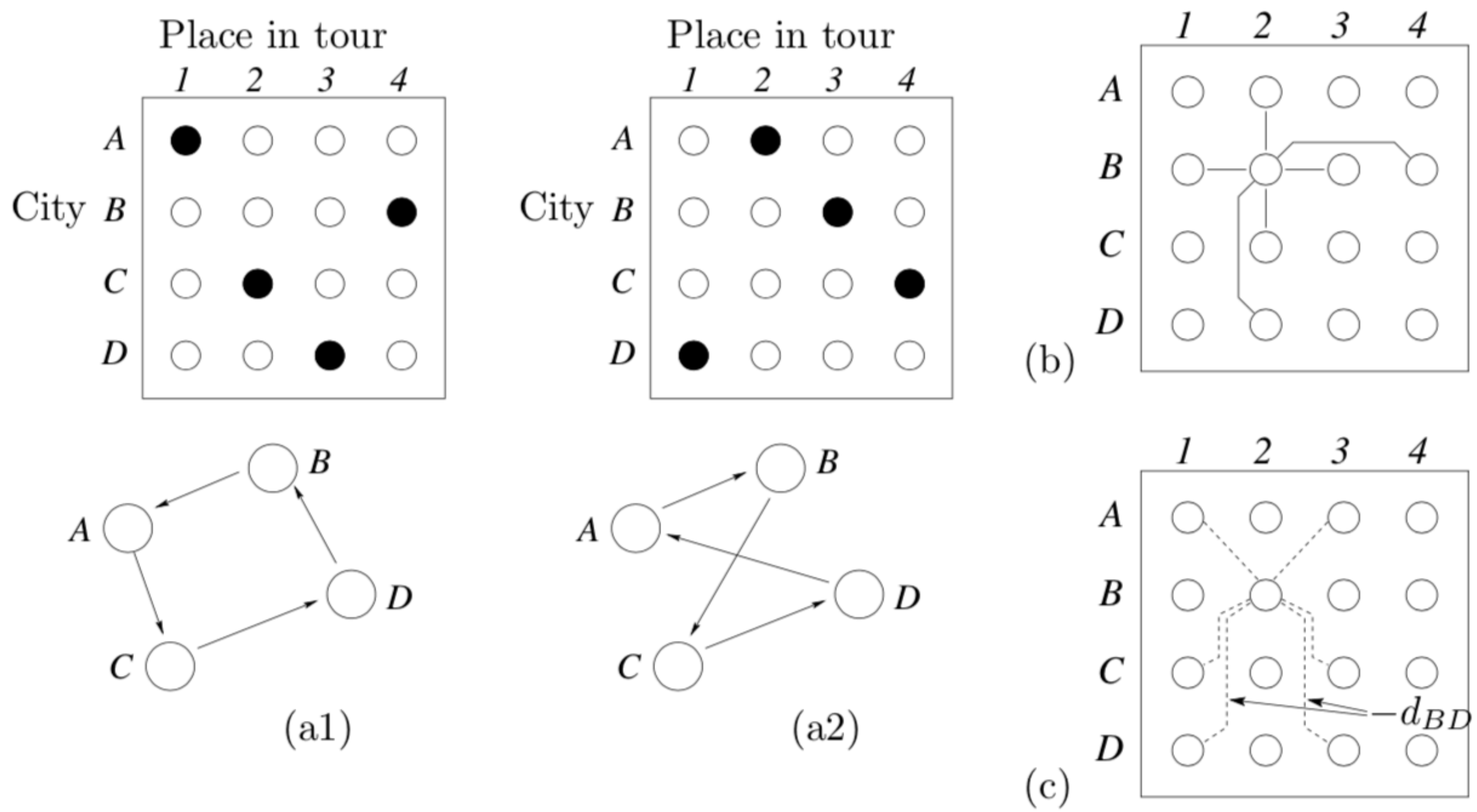
.	0	0	0	0	-2	2	-2	2	2	-2	0	0	0	2	0	0	-2	0	2	2	0	0	-2	-2
0	.	4	4	0	-2	-2	-2	-2	-2	-2	0	-4	0	-2	0	0	-2	0	-2	-2	4	4	2	-2
0	4	.	4	0	-2	-2	-2	-2	-2	-2	0	-4	0	-2	0	0	-2	0	-2	-2	4	4	2	-2
0	4	4	.	0	-2	-2	-2	-2	-2	-2	0	-4	0	-2	0	0	-2	0	-2	-2	4	4	2	-2
0	0	0	0	.	2	-2	-2	2	-2	2	-4	0	0	-2	4	-4	-2	0	-2	2	0	0	-2	2
-2	-2	-2	-2	2	.	0	0	0	0	4	-2	2	-2	0	2	-2	0	-2	0	0	-2	-2	0	4
2	-2	-2	-2	-2	0	.	0	0	4	0	2	2	-2	4	-2	2	0	-2	4	0	-2	-2	0	0
-2	-2	-2	-2	-2	0	0	.	0	0	0	2	2	2	0	-2	2	4	2	0	0	-2	-2	0	0
2	-2	-2	-2	2	0	0	0	.	0	0	-2	2	2	0	2	-2	0	2	0	4	-2	-2	-4	0
2	-2	-2	-2	-2	0	4	0	0	.	0	2	2	-2	4	-2	2	0	-2	4	0	-2	-2	0	0
-2	-2	-2	-2	2	4	0	0	0	0	.	-2	2	-2	0	2	-2	0	-2	0	0	-2	-2	0	4
0	0	0	0	-4	-2	2	2	-2	2	-2	.	0	0	2	-4	4	2	0	2	-2	0	0	2	-2
0	-4	-4	-4	0	2	2	2	2	2	2	0	.	0	2	0	0	2	0	2	2	-4	-4	-2	2
0	0	0	0	0	-2	-2	2	2	-2	-2	0	0	.	-2	0	0	2	4	-2	2	0	0	-2	-2
2	-2	-2	-2	-2	0	4	0	0	4	0	2	2	-2	.	-2	2	0	-2	4	0	-2	-2	0	0
0	0	0	0	4	2	-2	-2	2	-2	2	-4	0	0	-2	.	-4	-2	0	-2	2	0	0	-2	2
0	0	0	0	-4	-2	2	2	-2	2	-2	4	0	0	2	-4	.	2	0	2	-2	0	0	2	-2
-2	-2	-2	-2	-2	0	0	4	0	0	0	2	2	2	0	-2	2	.	2	0	0	-2	-2	0	0
0	0	0	0	0	-2	-2	2	2	-2	-2	0	0	4	-2	0	0	2	.	-2	2	0	0	-2	-2
2	-2	-2	-2	-2	0	4	0	0	4	0	2	2	-2	4	-2	2	0	-2	.	0	-2	-2	0	0
2	-2	-2	-2	2	0	0	0	4	0	0	-2	2	2	0	2	-2	0	2	0	.	-2	-2	-4	0
0	4	4	4	0	-2	-2	-2	-2	-2	-2	0	-4	0	-2	0	0	-2	0	-2	-2	.	4	2	-2
0	4	4	4	0	-2	-2	-2	-2	-2	-2	0	-4	0	-2	0	0	-2	0	-2	-2	4	.	2	-2
-2	2	2	2	-2	0	0	0	-4	0	0	2	-2	-2	0	-2	2	0	-2	0	-4	2	2	.	0
-2	-2	-2	-2	2	4	0	0	0	0	4	-2	2	-2	0	2	-2	0	-2	0	0	-2	-2	0	.





- 25 -unit binary Hopfield network
- (a) four patterns as 5x5 binary images
- (b)- (m) evolution of state of the network

Recap:TSP on Hopfield Network



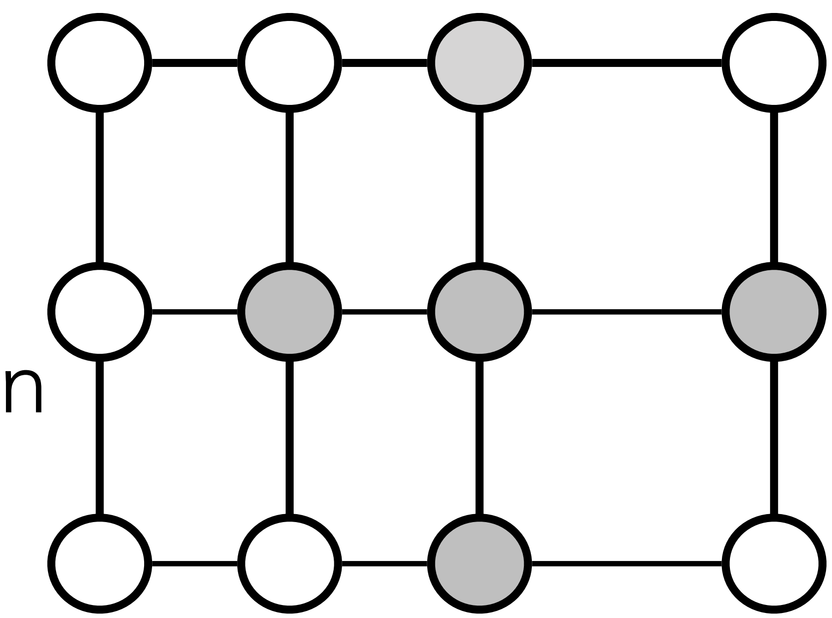
- **States have exactly one `1` in every row/column**: putting large negative weights between any pair of neurons in the same row/column
- **Weights encode the total distance**: putting negative weights proportional to the distance between the nodes

BOLTZMANN MACHINES

Stochastic Hopfield Network

- Hopfield Network Minimizes Energy function

$$E(x;W) = -\frac{1}{2}\sum_{m,n} W_{mn}x_m x_n$$



- Approximating probability distribution

$$P(x\,|\,\beta, W) = \frac{1}{Z}\exp[-\beta E(x;J)]$$

- Boltzmann machine is *Stochastic* Hopfield Network

- **E**nergy-**B**ased **M**odel: $P(x) \propto \exp[-\beta E(x)]$

temperature

Boltzmann Machine

- Compute activity $a_i(t) = \sum_j w_{ij} x_j(t)$
- Set $x_i = +1$ with probability $\dfrac{1}{1 + e^{-2a_i}}$

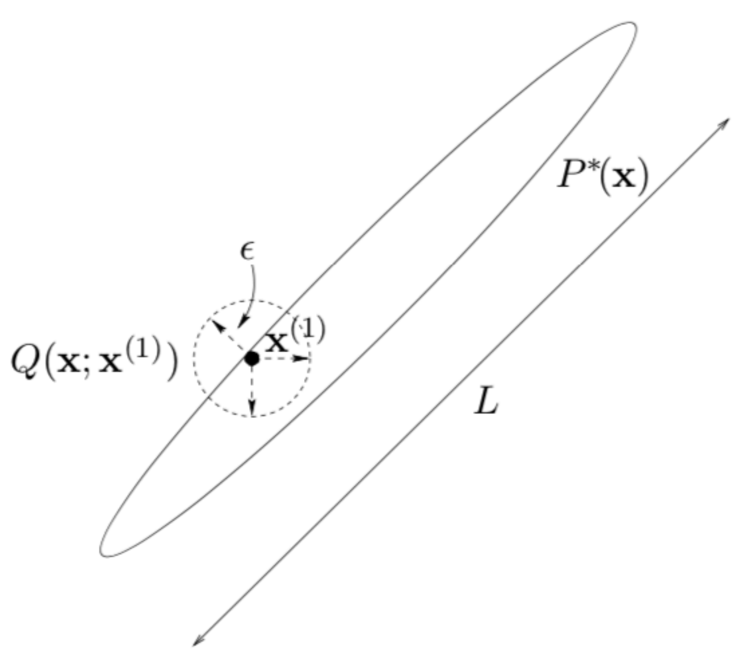
  Else set $x_i = -1$
- Sample from a Boltzmann Machine with Gibbs sampling for probability distribution

  $P(x \mid \beta, W) = \dfrac{1}{Z} \exp[-\beta E(x; W)]$

Markov Chain Monte Carlo

Proposal distribution is state dependent

- For any positive Q , the probability distribution of $x^{(t)}$ tends to $P(x) = P^{\star}(x)/Z$
- Employ a proposal distribution with a small length scale ϵ

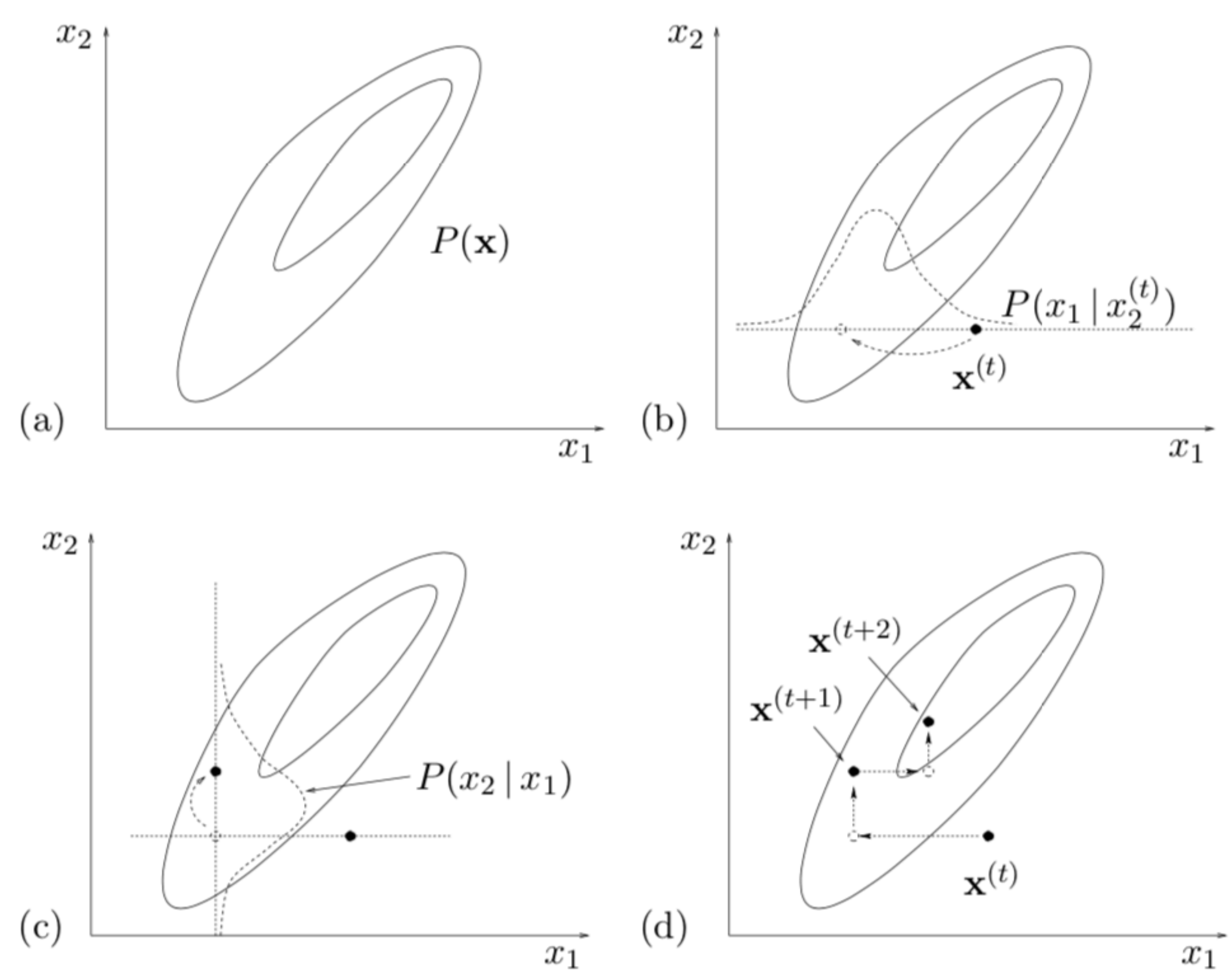


Large scale: low acceptance

Small scale: slow progress

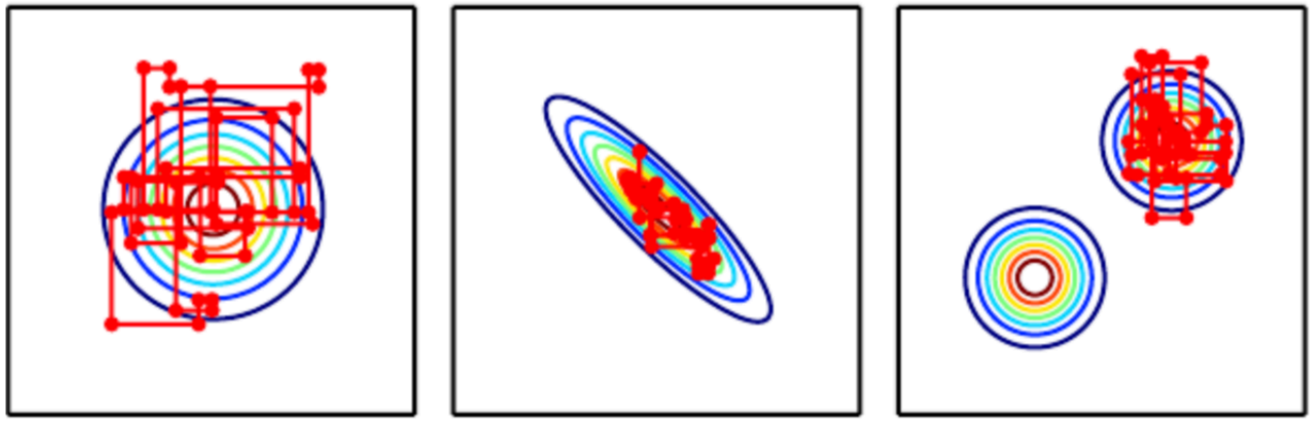
Gibbs Sampling

- Gibbs sampling is a type of MCMC method
- Sample from *joint* distribution $P(\mathbf{x})$ is **hard**, sample from *conditional* distribution $P(x_i \,|\, \{x\}_{j \neq i})$ is **easy**



Sample from Boltzmann Machine

- MCMC has a tendency to mix poorly: stuck in low-energy region



- Two variables example $E(a, b) = -wab$ for large positive w
    - Update b: $P(b = 1 \mid a = 1) = \sigma(w)$ almost surely to be 1
    - Gibbs sampling will very rarely flip the signs

Boltzmann Machine Learning

- Given data examples $\{x^{(n)}\}_{n=1}^{N}$, adjust the weights such that the generative model matches the data
$$P(x \mid \mathbf{W}) = \frac{1}{Z(\mathbf{W})} \exp[-\frac{1}{2}\mathbf{x}^T \mathbf{W} \mathbf{x}]$$

- Maximum Likelihood Estimation:
$$\log \Big[\prod_n P(\mathbf{x}^{(n)} \mid \mathbf{W}) \Big] = \sum_n \Big[\frac{1}{2} \mathbf{x}^{(n)^T} \mathbf{W} \mathbf{x}^{(n)} - \log Z(\mathbf{W}) \Big]$$

- Learning: take the derivative!

Boltzmann Machine Learning

- Gradient of the log likelihood

$$\frac{\partial}{\partial w_{ij}} \log P(\{\mathbf{x}^{(n)}\} \,|\, \mathbf{W}) = \sum_{n}^{N} \left[\langle x_i^{(n)} x_j^{(n)} \rangle - \langle x_i x_j \rangle_{P(\mathbf{x}^{(n)})} \right]$$

data correlation

model correlation

- $\langle x_i x_j \rangle_{\text{Data}} = \frac{1}{N} \sum_{n} x_i^{(n)} x_j^{(n)}$ $\quad \langle x_i x_j \rangle_{P(\mathbf{x}^{(n)})} = \sum_{\mathbf{x}} x_i x_j P(\mathbf{x} \,|\, \mathbf{W})$

- Data correlation is easy but model correlation is hard — estimate by Monte Carlo

Interpretation of BM

- Waking and Sleeping
  $N\left[\langle x_i x_j \rangle_{\text{Data}} - \langle x_i x_j \rangle_{P(\mathbf{x}^{(n)})}\right]$

- Wake: $\langle x_i x_j \rangle_{\text{Data}}$ increase weights in real world

- Sleep: $\langle x_i x_j \rangle_{P(\mathbf{x}^{(n)})}$ decrease weights in the `dream'

- Weights do not change when the two terms balance

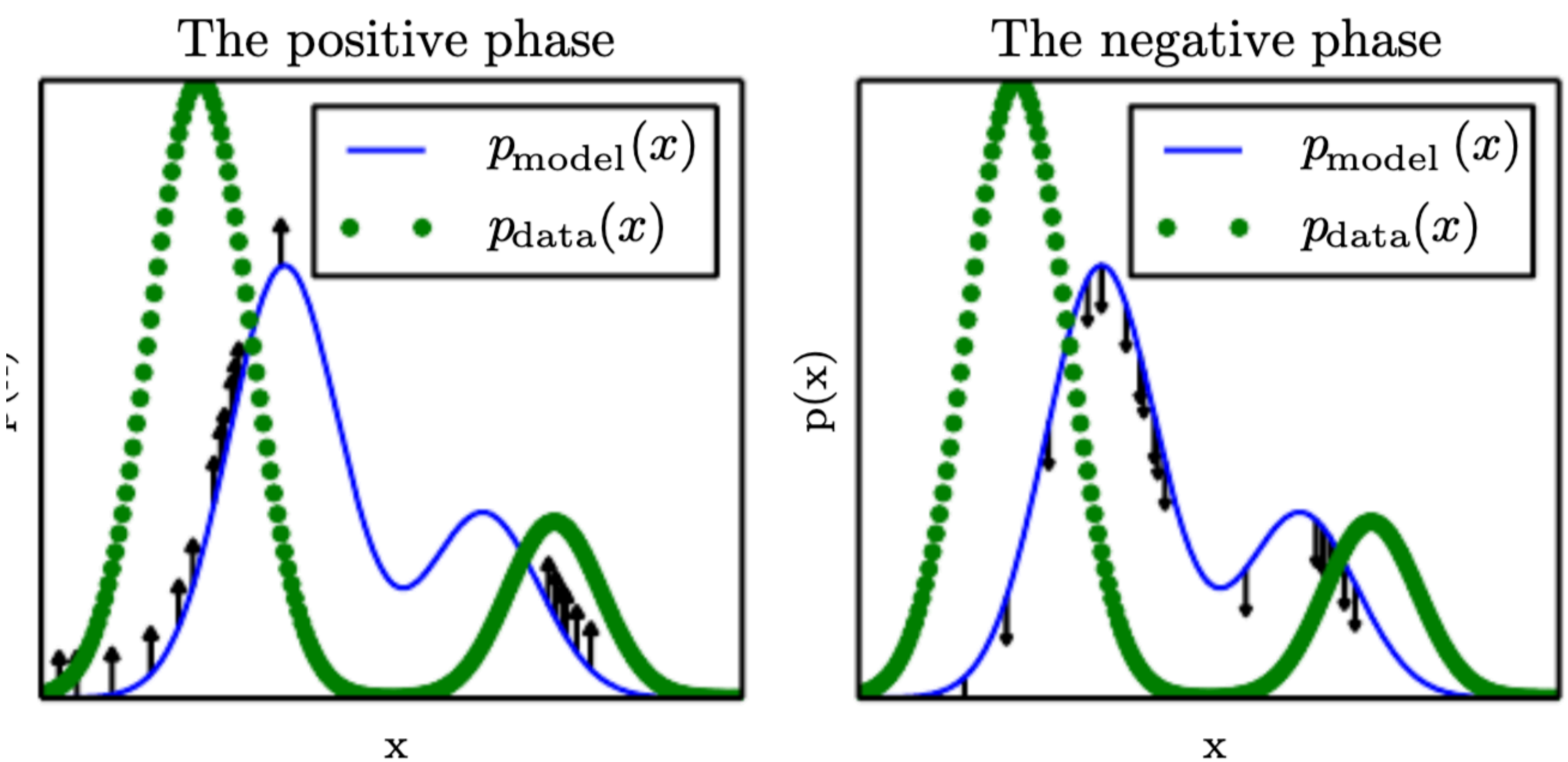
Estimation with MCMC

- In general, taking derivative w.r.t log-likelihood is difficult as the partition function depends on $\mathbf{W}$
- Gradient of $Z(\mathbf{W})$: $\nabla_{\mathbf{W}} \log Z = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \nabla_{\mathbf{W}} \log P^{\star}(\mathbf{x})$
- Monte Carlo methods for approximating maximizing the likelihood of models with intractable partition functions

Naive MCMC

- Sample a batch of data $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(B)}\}$
- Compute gradient $\mathbf{g} = \dfrac{1}{B} \sum_b \nabla_{\mathbf{W}} \log P^\star(\mathbf{x}^{(b)}; \mathbf{W})$
- Random initiate states $\{\tilde{\mathbf{x}}^{(1)}, \cdots, \tilde{\mathbf{x}}^{(b)}\}$
- Update states with Gibbs sampling
- Update $\mathbf{g} = \mathbf{g} - \dfrac{1}{B} \sum_b \nabla_{\mathbf{W}} \log P^\star(\tilde{\mathbf{x}}^{(b)}; \mathbf{W})$

Drawback of Naive MCMC



- MCMC tries to balance data distribution and model distribution

- Burn-In operations with random initialization take a lot of computation

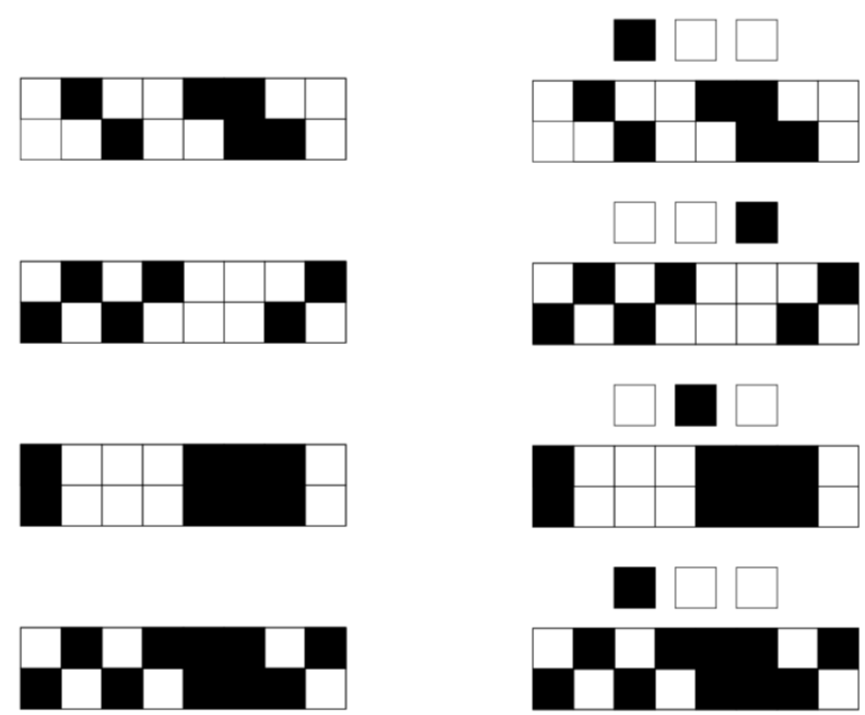
Contrastive Divergence

- Sample a batch of data $\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(b)}\}$
- Compute gradient $\mathbf{g} = \frac{1}{B}\sum_b \nabla_{\mathbf{W}} \log P^\star(\mathbf{x}^{(b)}; \mathbf{W})$
- Initiate states $\{\tilde{\mathbf{x}}^{(1)}, \cdots, \tilde{\mathbf{x}}^{(b)}\}$ with data $\tilde{\mathbf{x}}^b \leftarrow \mathbf{x}^b$
- Update states with Gibbs sampling
- Update $\mathbf{g} = \mathbf{g} - \frac{1}{B}\sum_b \nabla_{\mathbf{W}} \log P^\star(\tilde{\mathbf{x}}^{(b)}; \mathbf{W})$

Criticism

- Second-order statistics of the environment are not enough — need higher-order concepts

- Shift ensemble: not learnable using second-order statistics alone



(a) (b)

Higher-Order Correlations

- Higher-order Boltzmann machines
$$P(\mathbf{x} \,|\, \mathbf{W}, \mathbf{V}, \cdots) = \frac{1}{Z} \exp\Big(\frac{1}{2} \sum_{ij} w_{ij} x_i x_j + \frac{1}{6} \sum_{ij} v_{ijk} x_i x_j x_k + \cdots \Big)$$

- Simulate using stochastic updates, learning is equivalent, but require large number of parameters

- Include **hidden variables**: high-order correlations are described by hidden variables

BM with Hidden Units

- Activity rule, let $\mathbf{y}^{(n)} \equiv (\mathbf{x}^{(n)}, \mathbf{h})$

$$P(\mathbf{x}^{(n)} \,|\, \mathbf{W}) = \sum_{\mathbf{h}} P(\mathbf{x}^{(n)}, \mathbf{h} \,|\, \mathbf{W}) = \frac{1}{Z(\mathbf{W})} \exp[-\frac{1}{2}[\mathbf{y}^{(n)}]^T \mathbf{W} \mathbf{y}^{(n)}]$$

- Differentiating log likelihood

$$\frac{\partial}{\partial w_{ij}} \log P(\{\mathbf{x}^{(n)}\} \,|\, \mathbf{W}) = \sum_n \left[\underbrace{\langle y_i y_j \rangle_{P(\mathbf{h}|\mathbf{x}^{(n)},\mathbf{W})}}_{\boxed{\text{data correlation}}} - \underbrace{\langle y_i y_j \rangle_{P(\mathbf{x},\mathbf{h}|\mathbf{W})}}_{\boxed{\text{model correlation}}} \right]$$

- Gradients hard to compute