Rapid Introduction to Machine Learning/ Deep Learning

Hyeong In Choi

Seoul National University

Lecture 6a
Learning in Restricted Boltzmann machine

November 24, 2015

Table of contents

1. 1. Objectives of Lecture 6a
2. 2. Block Gibbs Sampling for RBM
   - 2.1. Recall
   - 2.2. Sampling
   - 2.3. Block Gibbs sampling
3. 3. Training RBM
   - 3.1. What to train?
   - 3.2. Contrastive Divergence (CD-k)
   - 3.3. Learning rule
   - 3.4. Persistence CD (PCD)
4. 4. Modeling real-valued input data for RBM

1. Objectives of Lecture 6a

Objective 1

Learn minimal Bayesian network formalism that is necessary for the understanding of the deep belief network

Objective 2

Learn how dependency and independency are encoded in the directed acyclic graph structure

Objective 3

Learn the D-separation theorem

Objective 4

Understand to view Markov chain and hidden Markov in the BN framework

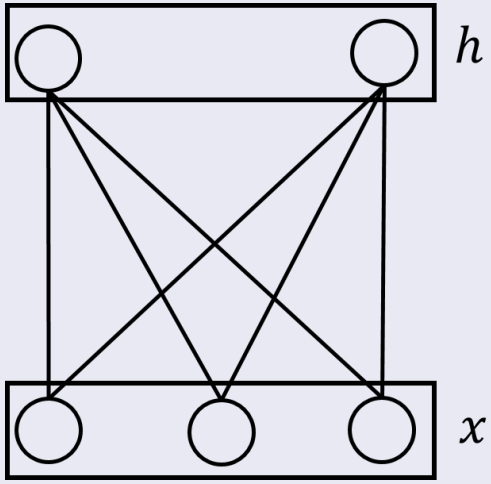
Objective 5

Present a very simple example as a preview of how deep belief network works

2. Block Gibbs Sampling for RBM

2.1. Recall

RBM



$x_j, h_i \in \{0, 1\}$

Probability $P(x, h) = \frac{1}{Z} \exp(-E(x, h))$,

where $E(x, h) = -\sum_{i,j} w_{ij} h_i x_j - \sum_j b_j x_j - \sum_i c_i h_i$

Conditional probability

$$P(h|x) = \prod_i P(h_i|x)$$

$$\begin{aligned} P(h_i = 1|x) &= \mathrm{sigm}(W_{i\bullet}x + c_i) \\ P(x_j = 1|h) &= \mathrm{sigm}(W_{\bullet j}^T h + b_j) \end{aligned}$$

$$P(x) = \frac{1}{Z}\exp\left[b^T x + \sum_i \mathrm{softplus}(W_{i\bullet}x + c_i)\right]$$

note:

- $P(h_i = 1|x)$ and $P(x_j = 1|h)$ are easy to compute
- $P(x, h)$ and $P(x)$ are hard to compute due to the presence of Z

2.2. Sampling

What to a sample $(x, h) \sim P(x, h)$

- But since $P(x, h)$ is hard to compute

Digression: Clifford-Hammersley

- Suppose $P(x_1, \cdots, x_N)$ is a joint distribution of random variables $X_1, \cdots, X_N$. The essence of Clifford-Hammersley theorem is that if we know all of conditional distribution $P(x_i \,|\, x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_N)$ for all $i, \cdots, N$, then we can recover $P(x_1, \cdots, x_N)$

- Example(2 variable)

$$
\begin{aligned}
P(x,y) &= P(x|y)P(y) \\
P(x|y) &= \frac{P(x,y)}{P(y)} \\
P(y|x) &= \frac{P(x,y)}{P(x)}
\end{aligned}
$$

Thus

$$\frac{1}{P(y)} = \sum_x \frac{P(x)}{p(y)} = \sum_x \frac{P(x|y)}{P(y|x)}$$

Therefore

$$P(x,y) = \frac{P(x|y)}{1\P(y)} = \frac{P(x|y)}{\sum_x \frac{P(x|y)}{P(y|x)}}$$

Digression: Gibbs sampling

- Given $P(x_1, \cdots, x_N)$ via $P(x_i \,|\, x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_N)$, can construct a sampling sequence $\sim P(x_1, \cdots, x_N)$
- Start with $X^{(0)} = (a_1, \cdots, a_N) \rightarrow X^{(1)} = (a_1', a_2, \cdots, a_N)$
  where a_1' : sample from $P(x_1 \,|\, x_2 = a_2, \cdots, x_N = a_N)$
  $\rightarrow X^{(2)} = (a_1', a_2', a_3, \cdots, a_N)$
  where a_2' : sample from $P(x_2 \,|\, x_1 = a_1', x_3 = a_3, \cdots, x_N = a_N)$
  Thus way, cycle through each variable, one at a time, fixing the rest, and repeat
- Get a sequence

$$X^{(0)}, X^{(1)}, \ldots, X^{(t)}, X^{(t+1)}, \ldots$$

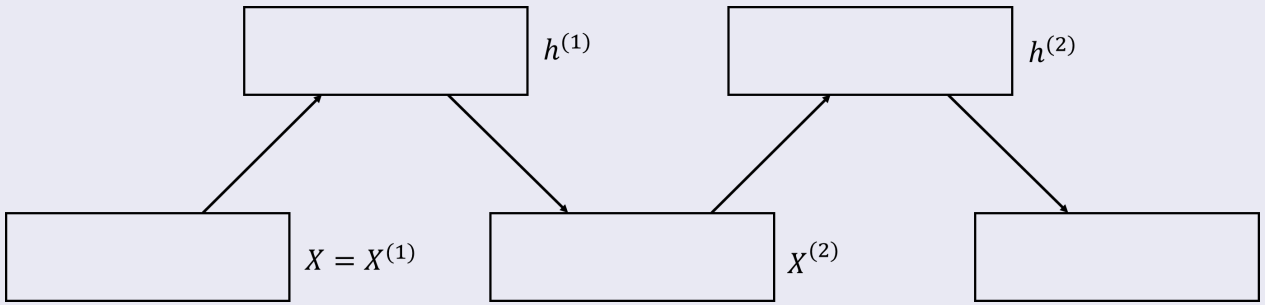
- Throw away first few terms (burn-in)

2.3. Block Gibbs sampling

- RBM has the property that

$$h_i \perp h_j \quad | \quad x$$
$$x_i \perp x_j \quad | \quad h$$

- Thus once x is given, the Gibbs sampling step for $h_1, \cdots, h_m$ can be done simultaneously
- Similarly given $h, X_1, \cdots, X_d$ can be updated simultaneously
-



3. Training RBM
3.1. What to train?

Generative model

- RBM is a generative model, meaning it modes the full doint probability $P(x, h)$
- Contrast it with the discriminative model that is primarily concerned with the conditional probability $P(h|x)$ or $P(y|x)$
- The equilibrium (Gibbs) samples are what we ate mostly interested in, and we ant them to have the resemblance with the data
- Given data $\mathcal{D} = \{X^{(t)}\}_{t=1}^{T}$, we want to maximize the probability

$$\frac{1}{T}\sum_{t=1}^{T}\log P(X^{(t)})$$

- Look at one data $X = x$, and try to maximize $\log P(x)$ by adjusting weights w_{ij} and thresholds b_j and c_i

3.1.2

- $P(x) = \sum_h P(x,h) = \dfrac{\sum_h e^{-E(x,h)}}{\sum_{x,h} e^{-E(x,h)}}$

$$\log P(x) = \log \sum_h e^{-E(x,h)} - \log \sum_{x,h} e^{-E(x,h)}$$

- Let θ be any of w_{ij}, b_j, c_i

$$
\begin{aligned}
\frac{\partial}{\partial\theta}\log P(x) &= \frac{-\sum_h\left\{-E(x,h)\frac{\partial E(x,h)}{\partial\theta}\right\}}{\sum_h e^{-E(x,h)}} \\
&= -\frac{-\sum_{x,h}-E(x,h)\frac{\partial E(x,h)}{\partial\theta}}{\sum_{x,h} e^{-E(x,h)}} \\
&= -\sum_h P(h|x)\frac{\partial E(x,h)}{\partial\theta} + \sum_{x,h} P(x,h)\frac{\partial E(x,h)}{\partial\theta} \\
&= -E_h\left[\frac{\partial E}{\partial\theta}(x,h)\,|\,x\right] + E_{x,h}\left[\frac{\partial E(x,h)}{\partial\theta}\right]
\end{aligned}
$$

- Let $\theta = w_{ij}$

$$\frac{\partial E(x,h)}{\partial\theta} = -h_i x_j$$

Thus

$$\begin{aligned}\frac{\partial}{\partial w_{ij}} \log P(x) &= \sum_h P(h|x) h_i x_j - \sum_{x,h} P(h,x) h_i x_j \\ &= < h_i x_j >_{data} - < h_i x_j >_{model}\end{aligned}$$

Computation of $< h_i x_j >_{data}$: Positive phase

$$\begin{aligned}< h_i x_j >_{data} &= \sum_h P(h|x) h_i x_j \\ &= \sum_{h_1} \cdots \sum_{h_m} \prod_k P(h_k|x) h_i x_j \\ &= \sum_{\substack{h \\ h \neq h_i}} \sum_{h_i} \prod_k P(h_k|x) h_i x_j\end{aligned}$$

Now

$$\begin{aligned}
\sum_{h_i}\prod_k P(h_k|x)h_ix_j &= \sum_{h_i}\left(\prod_k P(h_k|x)\right)P(h_i|x)h_ix_j \\
&= \prod_{k\neq i} P(h_k|x)\sum_{h_i} P(h_i|x)h_ix_j \\
&= \prod_{k\neq i} P(h_k|x)P(h_i=1|x)x_j
\end{aligned}$$

Thus

$$\begin{aligned}
\sum_h P(h|x)h_ix_j &= \sum_{\substack{h\\ h\neq h_i}}\prod_{k\neq i} P(h_k|x)P(h_i=1|x)x_j \\
&= P(h_i=1|x)x_j \qquad \left(\because \sum_{h_k} P(h_k|x)=1\right)
\end{aligned}$$

Thus

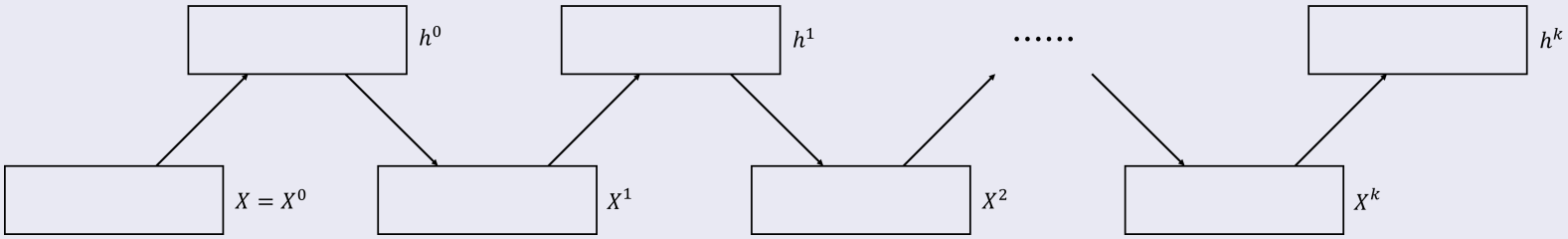
$$
\begin{aligned}
< h_i x_j >_{data} &= P(h_i = 1 | x) x_j \\
&= \mathrm{sigm}(W_i x + c_i) x_j
\end{aligned}
$$

Computation of $< h_i x_j >_{model}$: Negative phase

- $$
\begin{aligned}
< h_i x_j >_{model} &= \sum_{x,h} P(x,h) h_i x_j \\
&= P(x_j = 1, h_i = 1)
\end{aligned}
$$

This is almost impossible to compute analytically ($\because$ due to the presence of Z)

- Block Gibbs sampling
  If h_i and x_j are chosen from $P(x, h)$ then
  $h_i x_j \sim P(x_j = 1, h_i = 1)$
- To get equilibrium sample, are the h lock Gibbs sampling



Then by the convergence property of Markov chain, (x^∞, h^∞) is a sample $\sim P(x, h)$ i.e.

$$\begin{aligned} P(x_j = 1, h_i = 1) &= h_i^\infty x_j^\infty \\ &\approx h_i^k x_j^k \end{aligned}$$

for large k

Thus we take $h_i^k x_j^k$ for $< h_i x_j >_{model}$

3.1.5

- Check by

$$\frac{\partial}{\partial \theta} \log P(x) = -\sum_h P(h|x) \frac{\partial E(x,h)}{\partial \theta} + \sum_{x,h} P(x,h) \frac{\partial E(x,h)}{\partial \theta}$$

that

$$\frac{\partial}{\partial b_j} \log P(x) = -\sum_h P(h|x) \frac{\partial E(x,h)}{\partial b_j} + \sum_{x,h} P(x,h) \frac{\partial E(x,h)}{\partial b_j}$$

Since $\frac{\partial E(x,h)}{\partial b_j} = -x$

$$\frac{\partial}{\partial b_j} \log P(x) = \sum_h P(h|x) x_j - \sum_{x,h} P(x,h) x_j = x_j - x_j^\infty$$

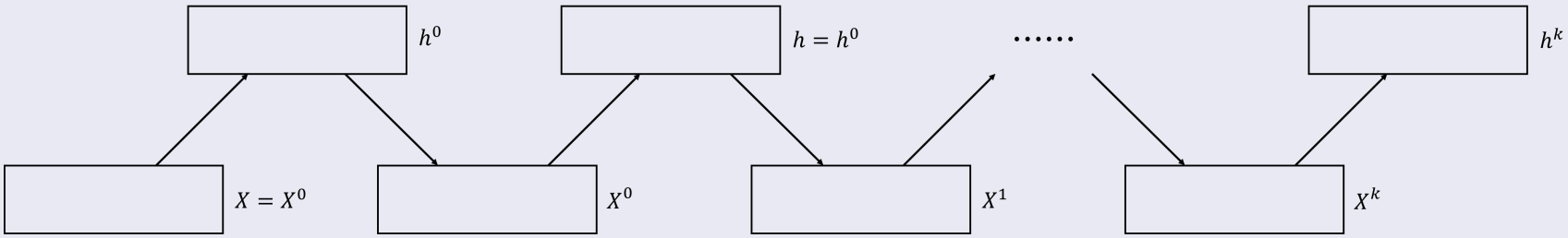
3.1.6

Similarly

$$\begin{aligned} \frac{\partial}{\partial c_i} \log P(x) &= \sum_h P(h|x) h_i - \sum_{x,h} P(x,h) h_i \\ &= P(h_i = 1|x) - \sum_{x,h} P(x,h) h_i \\ &= \mathrm{sigm}(W_i x + c_i) - h_i^\infty \end{aligned}$$

3.2. Contrastive Divergence (CD-k)

Given data $x = x^0$



$$\Delta w_{ij} = \frac{\partial}{\partial w_{ij}} \log P(x) = < h_i x_j >_{data} - < h_i x_j >_{model}$$

$$\approx < h_i x_j >^0 - < h_i x_j >^k,$$

where

$$
\begin{aligned}
< h_i x_j >^0 &= \mathrm{sigm}(W_i x + c_i) x_j \\
< h_i x_j >^k &= h_i^k x_j^k
\end{aligned}
$$

In the equilibrium sample

$$h_i^0 \sim P(h_i^0 = 1 \,|\, x) = \mathrm{sigm}(W_i x + c_i)$$

Thus one may use

$$< h_i x_j >^0 = h_i^0 x_j^0 \qquad (x_j^0 = x_j)$$

3.2.2

Similarly

$$\Delta b_j = \frac{\partial}{\partial b_j} \log P(x) = < x_j >_{data} - < x_j >_{model} \approx x_j^0 - x_j^k,$$

$$\Delta c_i = \frac{\partial}{\partial c_i} \log P(x) = < h_i >_{data} - < h_i >_{model} \approx h_i^0 - h_i^k,$$

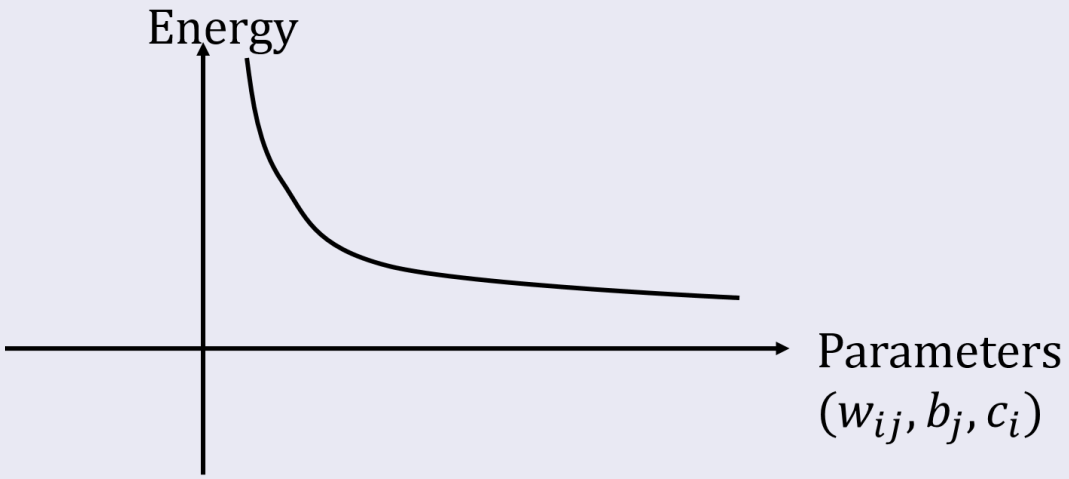
3.3. Learning rule

3.3.1

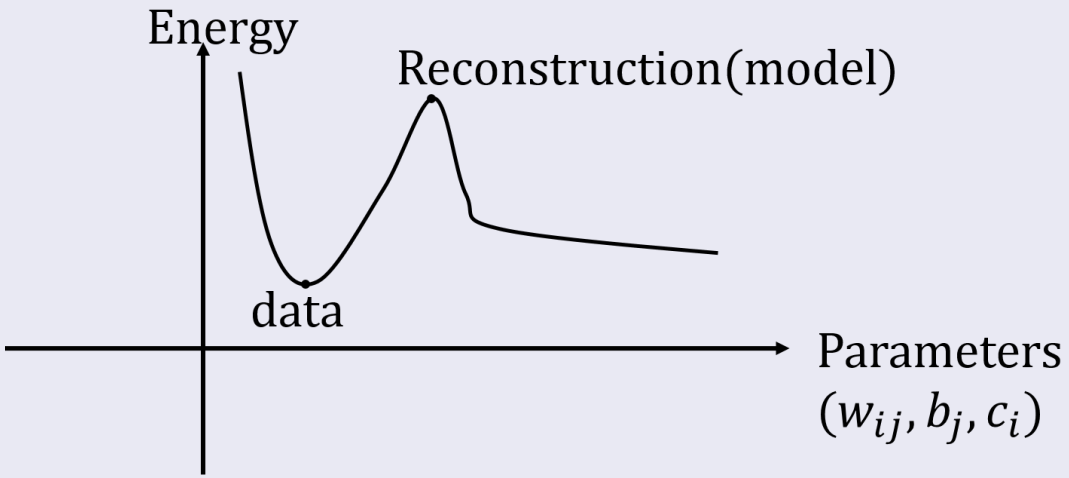
$$
\begin{aligned}
w_{ij}(new) &= w_{ij}(old) + \alpha \Delta w_{ij} \\
b_j(new) &= b_j(old) + \alpha \Delta b_j \\
c_i(new) &= c_i(old) + \alpha \Delta c_i
\end{aligned}
$$

(α: learning rate)

3.3.2



Before learning the data



Mini-batch version

- Mini-batch data $\mathcal{D} = \{x(t)\}_{t=1}^{m}$

$$\begin{aligned}
\Delta w_{ij} &= \frac{1}{m}\sum_{t=1}^{m} h_i^0(t)x_j^0(t) - \frac{1}{m}\sum_{t=1}^{m} h_i^k(t)x_j^k(t) \\
\Delta b_j &= \frac{1}{m}\sum_{t=1}^{m} x_j^0(t) - \frac{1}{m}\sum_{t=1}^{m} x_j^k(t) \\
\Delta c_i &= \frac{1}{m}\sum_{t=1}^{m} h_i^0(t) - \frac{1}{m}\sum_{t=1}^{m} h_i^k(t)
\end{aligned}$$

Changing k in CD-k

- At the beginning, use CD-1 with small weights
- Once the weights grow, the Markov chain mixes more slowly. Use CD-3
- Once the weights have grown more, there are spurious holes to be filled. So far letter mixing, increase k (to, say, 10)

3.4. Persistence CD (PCD)

3.4.1

- In CD, the negative phase was restarted from each data
- It takes long time to get good mixing
- In CD, (x^∞, h^∞) is a well-mixed state. It is called a "fantasy particle"
- It is hard to get such fantasy particle
- One random choose initial samples $\{x^*(t)\}_{t=1}^n$, where n is the number of fantasy particle. n may be different from the number m of mini-batch data, but one frequently set $n = m$
- From each $x^*(t)$, apply CD-k, to get $\{x^{*k}(t), h^{*k}(t)\}$
- Use $h_i^{*k}(t), x_j^{*k}(t)$ for the negative phase of CD-k
- This PCD tends to work better them CD

4. Modeling real-valued input data for RBM