InfoLab

Deep Learning Seminar
CH 18. Confronting the Partition Function

2018-01-31

Eunjeong Yi

DGIST 대구경북과학기술원
Daegu Gyeongbuk Institute of Science & Technology

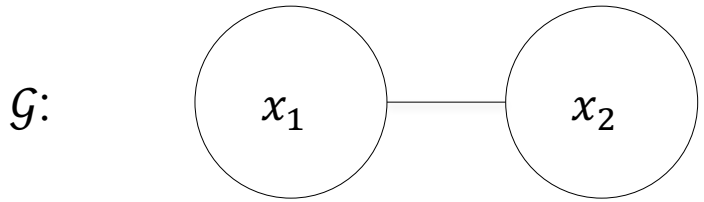
Chapter 18. Confronting the Partition Function

Content

- **Undirected Models**
- **Partition Function**
- **Log-likelihood Gradient**
- **Restricted Boltzmann Machine**
- **Stochastic Maximum Likelihood**
  - Naïve MCMC for Maximizing Log-likelihood
  - Contrastive Divergence
- **Estimating the Partition Function**
  - Importance Sampling

Undirected Models

- **Describe probability distribution of random variables which have no intrinsic direction or operate both directions**
  - ➢ there is no conditional probability distribution between random variables



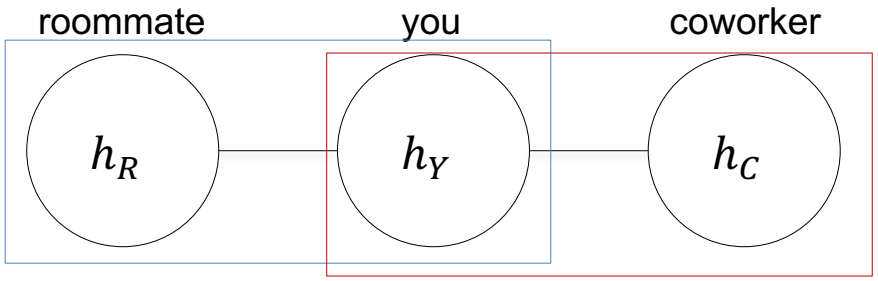
- **Defined by an unnormalized probability distribution**

$$\tilde{p}(\boldsymbol{x}) = \prod_{\mathcal{C} \in \mathcal{G}} \phi(\mathcal{C})$$

  - ➢ a factor $\phi(\mathcal{C})$ measures the affinity of the variables in $\mathcal{C}$ (non-negative)
    - $\mathcal{G}$: an undirected graph
    - $\mathcal{C}$: a clique in $\mathcal{G}$
    - ϕ: a factor (or clique potential)

Health State Example

- **Modeling the states of health of you, roommate, and coworker**
  - h_i: a state of health (1: good health, 0: with a cold)



$\phi(h_R, h_Y)$

c_1	$h_Y = 0$	$h_Y = 1$
$h_R = 0$	4	1
$h_R = 1$	1	10

$\phi(h_Y, h_C)$

c_2	$h_Y = 0$	$h_Y = 1$
$h_C = 0$	2	1
$h_C = 1$	1	10

- **Unnormalized probability distributions of health state**

$$\tilde{p}(\boldsymbol{x}) = \phi(h_R, h_Y)\phi(h_Y, h_C)$$

Partition Function

- **Normalize probability distribution which is not guaranteed to sum or integrate to 1**
- **Partition function Z**

$$Z = \int \tilde{p}(\boldsymbol{x})\, d\boldsymbol{x}$$

or

$$= \boldsymbol{\Sigma}_{\boldsymbol{x}} \tilde{p}(\boldsymbol{x})$$

- **Normalized probability distribution $\boldsymbol{p}(\mathbf{x}; \boldsymbol{\theta})$**

$$p(\mathbf{x}; \theta) = \frac{1}{Z(\theta)} \tilde{p}(\mathbf{x}; \boldsymbol{\theta})$$

x: sampled data
***θ*:** parameter

Probabilities for Health State

- $Z =$

x			$\tilde{p}(x)$
h_R	h_Y	h_C	$\phi(C_1)\phi(C_2)$
0	0	0	8
0	0	1	4
0	1	0	1
0	1	1	10
1	0	0	2
1	0	1	1
1	1	0	10
1	1	1	100
Z			136

$\phi(h_R, h_Y)$

C_1	$h_Y = 0$	$h_Y = 1$
$h_R = 0$	4	1
$h_R = 1$	1	10

$\phi(h_Y, h_C)$

C_2	$h_Y = 0$	$h_Y = 1$
$h_C = 0$	2	1
$h_C = 1$	1	10

- **All three people are in good health**

$$p(1,1,1) = \frac{1}{136} \times 10 \times 10 = 0.7353$$

- **Roommate and you are coughing but coworker is fine**

$$p(0,0,1) = \frac{1}{136} \times 4 \times 1 = 0.0294$$

Log-likelihood Gradient

- **Normalized probability distribution $p(\mathbf{x}; \boldsymbol{\theta})$**

$$p(\mathbf{x}; \theta) = \frac{1}{Z(\theta)} \tilde{p}(\mathbf{x}; \boldsymbol{\theta})$$

- **Log-likelihood**

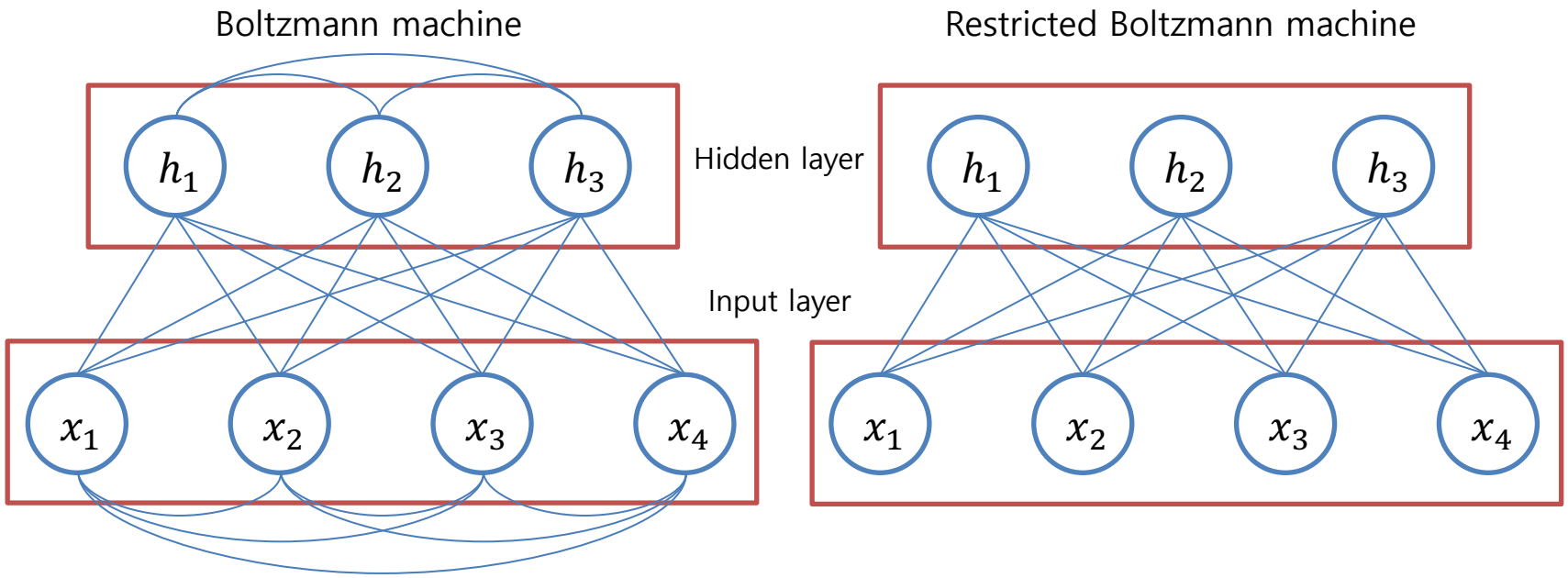
$$\nabla_\theta \log p(\mathbf{x}; \theta) = \nabla_\theta \log \tilde{p}(\mathbf{x}; \theta) - \nabla_\theta \log Z(\theta)$$

- **Gradient of $\log Z(\boldsymbol{\theta})$**

$$\nabla_\theta \log Z(\theta) = \mathbb{E}_{\mathrm{x} \sim p(\mathrm{x})} \nabla_\theta \log \tilde{p}(\mathrm{x}; \theta)$$

Restricted Boltzmann Machine (RBM)

- **Quintessential example of how graphical models are used for deep learning**
- **Connection between visible units and hidden units**

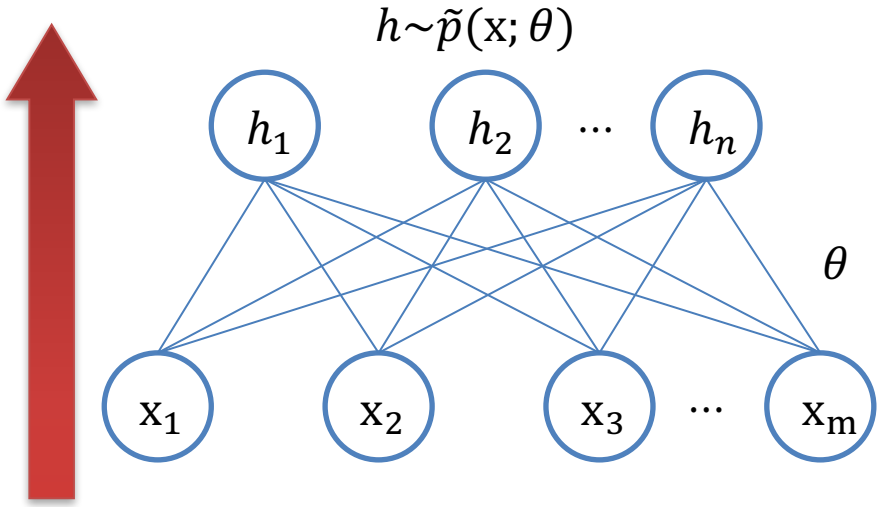


Stochastic Maximum Likelihood

- **Markov chain Monte Carlo (MCMC) method**
  - The method for sampling from target probability distributions using Markov chains
- **Algorithm**
  1) Sample $\{\mathrm{x}^{(1)}, \cdots, \mathrm{x}^{(m)}\}$ from training set
  2) Propagate sample to the hidden unit
  3) Initialize $\{\tilde{\mathrm{x}}^{(1)}, \cdots, \tilde{\mathrm{x}}^{(m)}\}$ to random value
  4) Sample $\{\tilde{\mathrm{x}}^{(1)}, \cdots, \tilde{\mathrm{x}}^{(m)}\}$ from model distribution
  5) Propagate $\{\tilde{\mathrm{x}}^{(1)}, \cdots, \tilde{\mathrm{x}}^{(m)}\}$

Naïve MCMC for Maximizing Log-likelihood

- **Forward step**
  - Sample $\mathrm{X} = \{\mathrm{x}^{(1)}, \cdots, \mathrm{x}^{(m)}\}$ from training set
  - Model distribution $\tilde{p}(\mathrm{x}; \theta)$ from hidden layer
  - Calculate $\frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\mathrm{x}^{(i)}; \theta)$



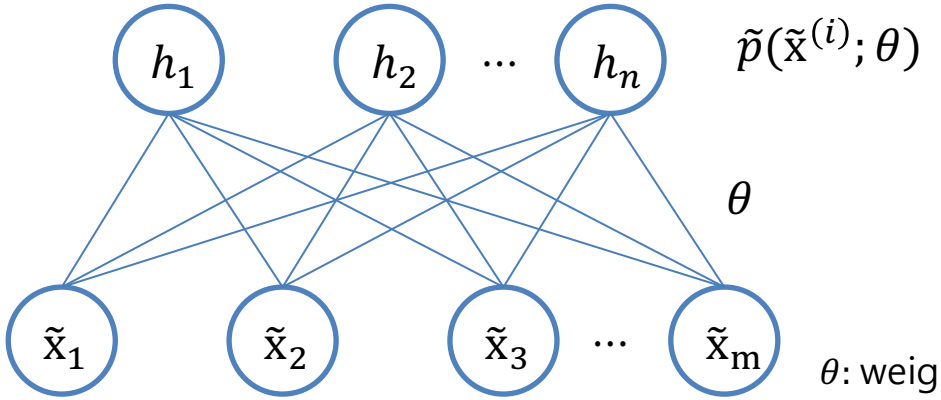
θ: weight

Naïve MCMC for Maximizing Log-likelihood

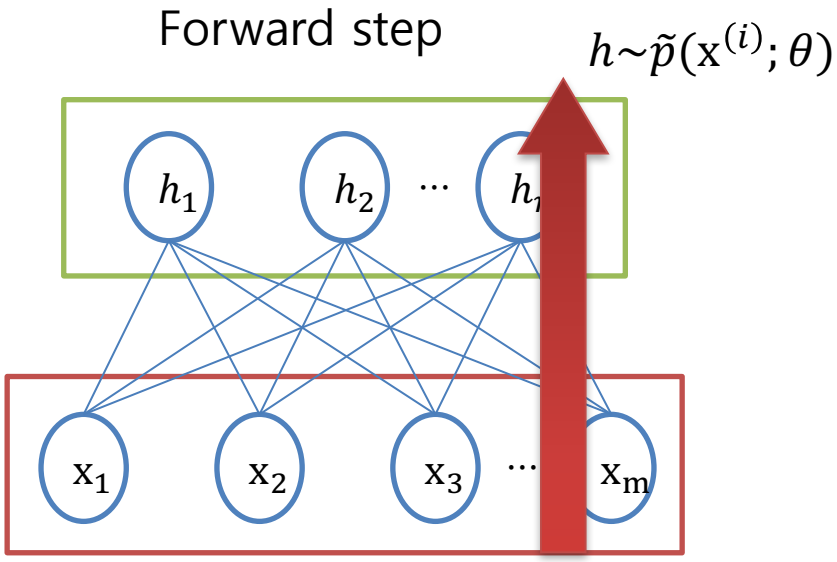
- **Backward step**
  - $\tilde{X} = \{\tilde{\mathrm{x}}^{(1)}, \cdots, \tilde{\mathrm{x}}^{(m)}\}$ initialized to random values
  - $\tilde{X}$ from $\tilde{p}(\mathrm{x}^{(i)}; \theta)$ with Gibbs sampling
  - Calculate $\frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\tilde{\mathrm{x}}^{(i)}; \theta)$
  - Weight update

$$\theta \leftarrow \theta + \epsilon\left(\frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\mathrm{x}^{(i)}; \theta) - \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\tilde{\mathrm{x}}^{(i)}; \theta)\right)$$

- **iteration until converge $\tilde{p}(\tilde{\mathrm{x}}^{(i)}; \theta)$ with $\tilde{p}(\mathrm{x}^{(i)}; \theta)$**

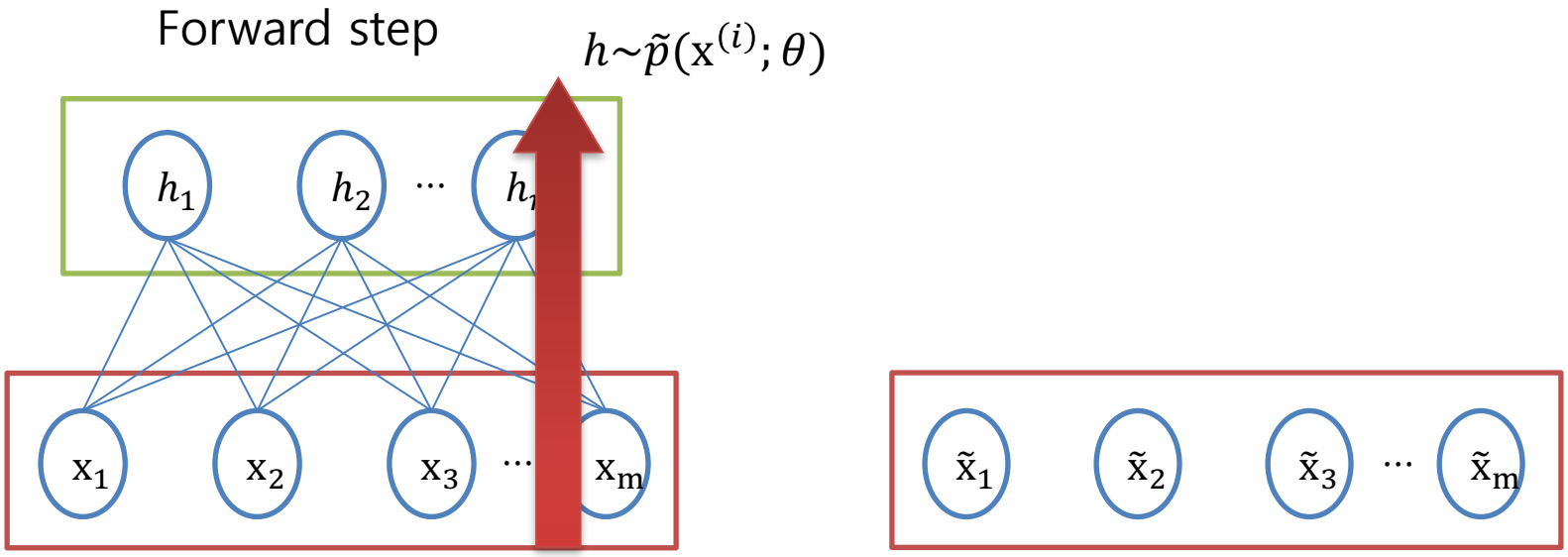


Naïve MCMC for Maximizing Log-likelihood



θ: weight

Naïve MCMC for Maximizing Log-likelihood



$\tilde{x}_1$ initialized to random values

$\tilde{X} = \{\tilde{\mathrm{x}}^{(1)}, \cdots, \tilde{\mathrm{x}}^{(m)}\}$ from $\tilde{p}(\mathrm{x}^{(i)}; \theta)$ Gibbs sampling

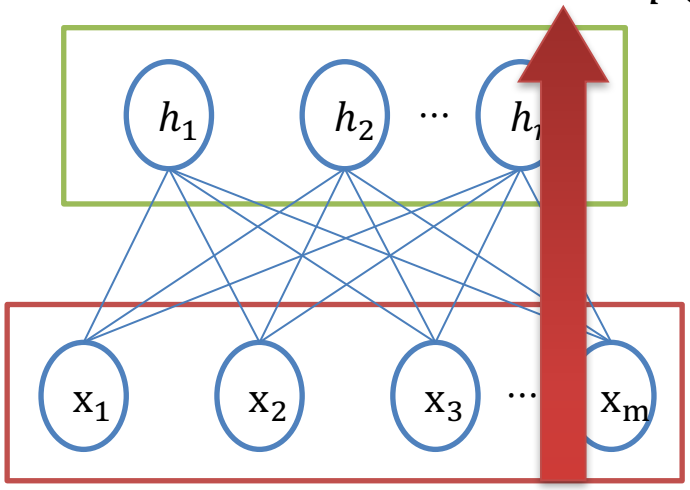
θ: weight

Naïve MCMC for Maximizing Log-likelihood

$$\theta \leftarrow \theta + \epsilon \left(\frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\mathrm{x}^{(i)}; \theta) - \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\tilde{\mathrm{x}}^{(i)}; \theta) \right)$$

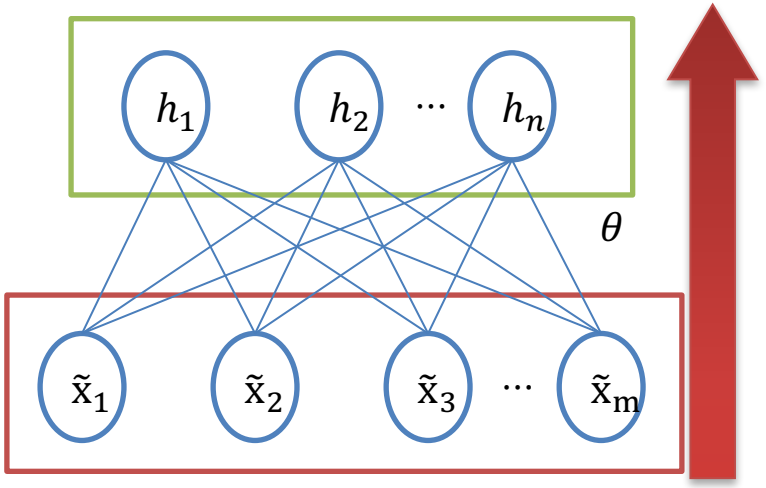
Forward step

$h \sim \tilde{p}(\mathrm{x}^{(i)}; \theta)$



Backward step

$\tilde{p}(\tilde{\mathrm{x}}^{(i)}; \theta)$



$\tilde{x}_1$ initialized to random values

$\tilde{X} = \{\tilde{\mathrm{x}}^{(1)}, \cdots, \tilde{\mathrm{x}}^{(m)}\}$ from $\tilde{p}(\mathrm{x}^{(i)}; \theta)$ Gibbs sampling

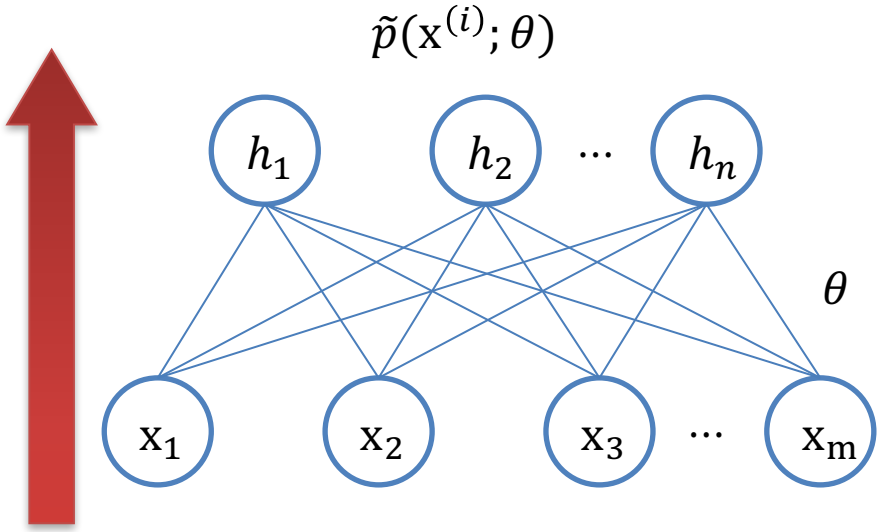
θ: weight

Contrastive Divergence

- **Initialization of sample from the training set instead of the Markov chain**
- **Algorithm**
  1) Sample $\{\mathrm{x}^{(1)}, \cdots, \mathrm{x}^{(m)}\}$ from training set
  2) Propagate sample to the hidden unit
  3) Sample $\{\tilde{\mathrm{x}}^{(1)}, \cdots, \tilde{\mathrm{x}}^{(m)}\}$ from model distribution
  4) Propagate $\{\tilde{\mathrm{x}}^{(1)}, \cdots, \tilde{\mathrm{x}}^{(m)}\}$

CD

- **Forward step**
  - Sample $\mathrm{X} = \{\mathrm{x}^{(1)}, \cdots, \mathrm{x}^{(m)}\}$ from training set
  - Model distribution $\tilde{p}(\mathrm{x}^{(i)}; \theta)$ from hidden layer
  - Calculate $\frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\mathrm{x}^{(i)}; \theta)$



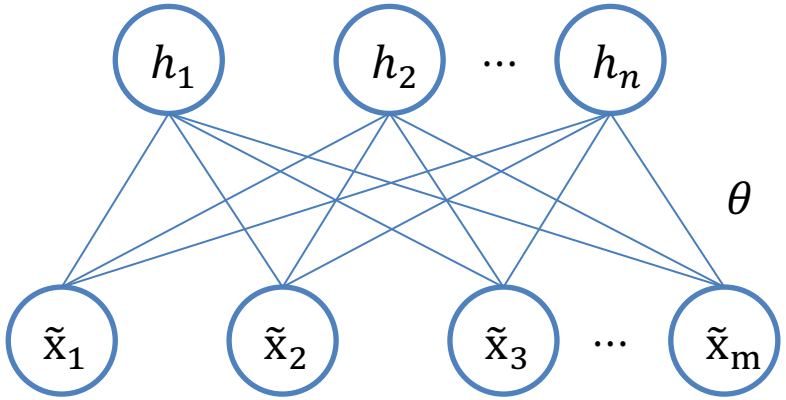
θ: weight

CD

- **Backward step**
  - Sample $\tilde{X} = \{\tilde{\mathrm{x}}^{(1)}, \cdots, \tilde{\mathrm{x}}^{(m)}\} = \{\mathrm{x}^{(1)}, \cdots, \mathrm{x}^{(m)}\}$ with Gibbs sampling
  - Calculate $\frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\tilde{\mathrm{x}}^{(i)}; \theta)$
  - Weight update

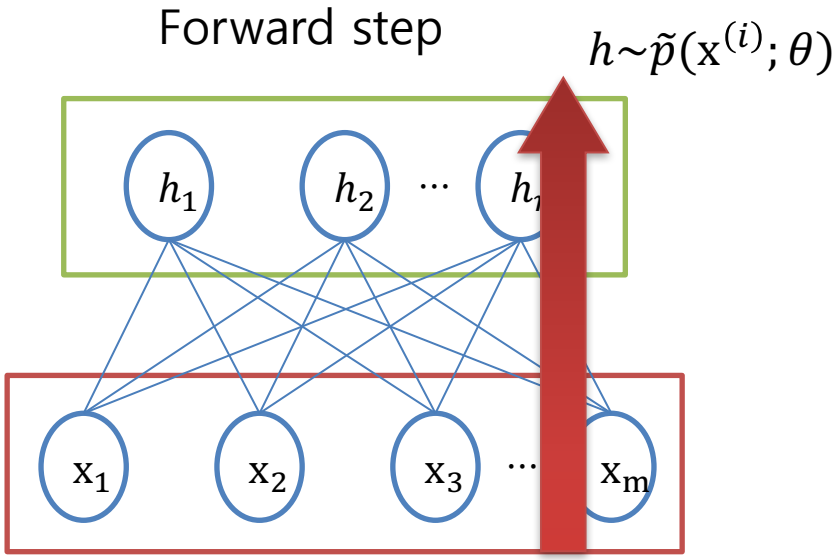
$$\theta \leftarrow \theta + \epsilon\left(\frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\mathrm{x}^{(i)}; \theta) - \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\tilde{\mathrm{x}}^{(i)}; \theta)\right)$$

- **iteration until converge $\tilde{p}(\tilde{\mathrm{x}}^{(i)}; \theta)$ with $\tilde{p}(\mathrm{x}^{(i)}; \theta)$**



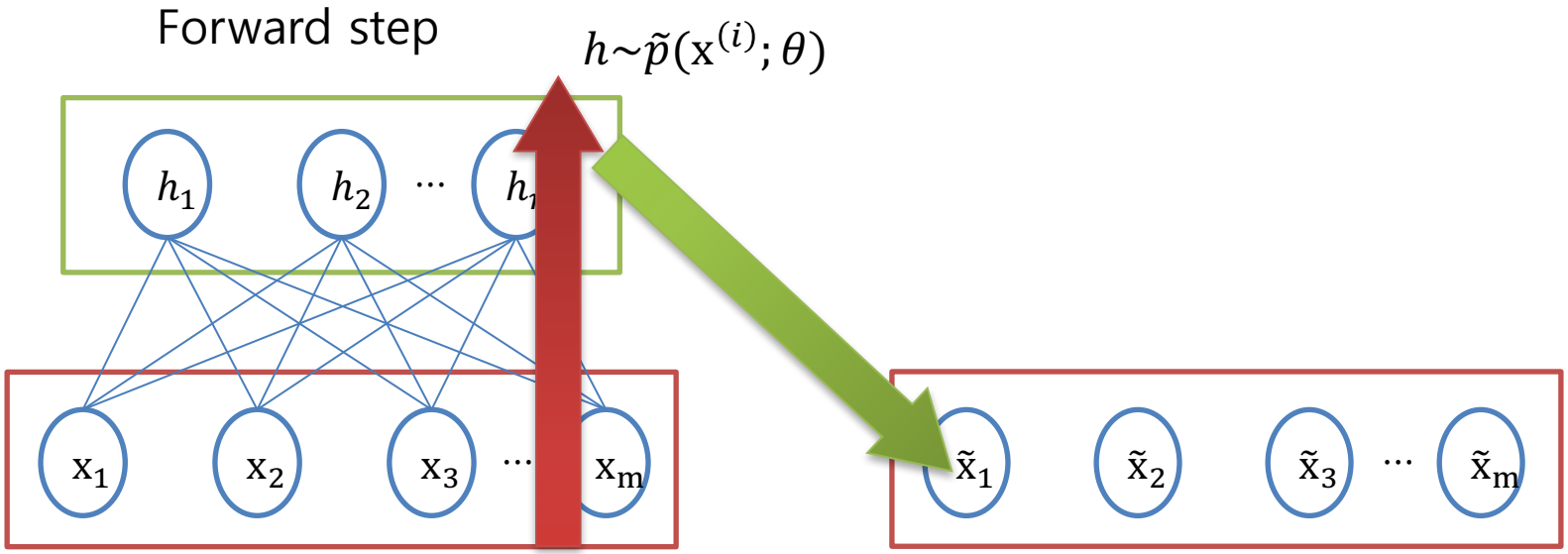
θ: weight

CD



θ: weight

CD



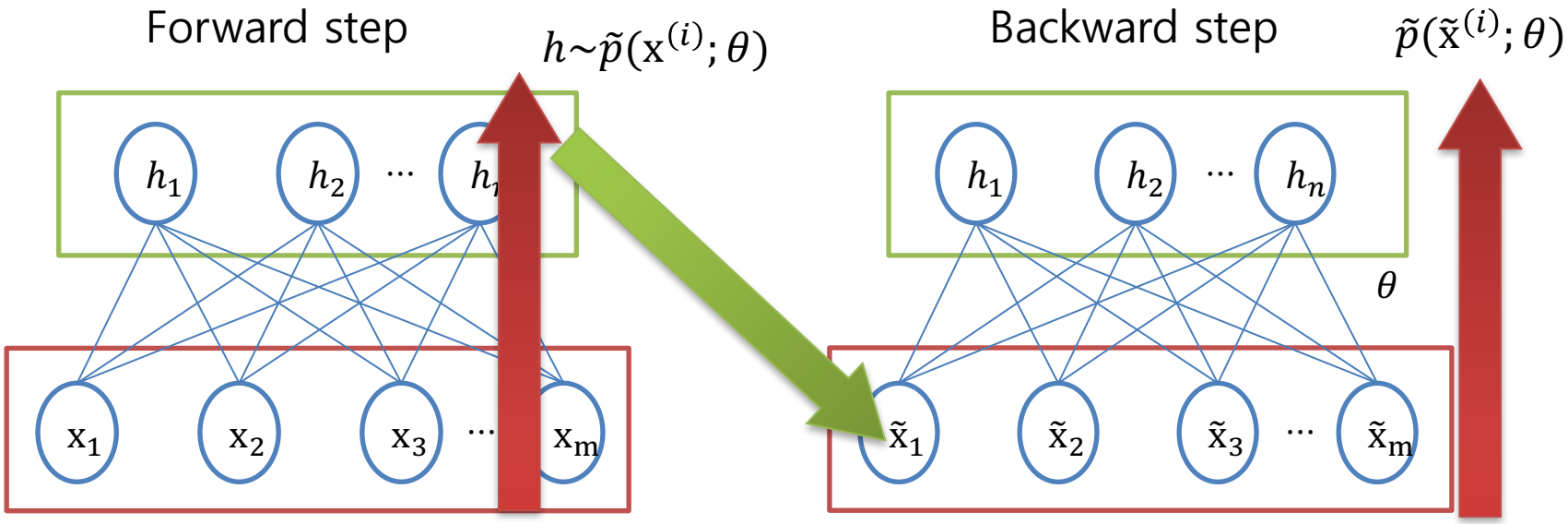
$\tilde{X} = \{\tilde{\mathrm{x}}^{(1)}, \cdots, \tilde{\mathrm{x}}^{(m)}\}$ from $\tilde{p}(\mathrm{x}^{(i)}; \theta)$

Gibbs sampling

θ: weight

CD

$$\theta \leftarrow \theta + \epsilon \left(\frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\mathrm{x}^{(i)}; \theta) - \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log \tilde{p}(\tilde{\mathrm{x}}^{(i)}; \theta) \right)$$



$\tilde{X} = \{\tilde{\mathrm{x}}^{(1)}, \cdots, \tilde{\mathrm{x}}^{(m)}\}$ from $\tilde{p}(\mathrm{x}^{(i)}; \theta)$
Gibbs sampling

θ: weight

Importance sampling

- **A variance reduction technique that can be used in the Monte Carlo method**
- **Estimating properties of a particular distribution using samples generated from a different distribution**
- **Input** $x \sim p(x)$

$$\mathbb{E}_{x\sim p(x)} f(x) = \int p(x) f(x) dx$$

$$p(x)f(x) = q(x)\frac{p(x)f(x)}{q(x)}$$

$$\mathbb{E}_{x\sim p(x)} f(x) = \mathbb{E}_{x\sim q(x)} \frac{p(x)f(x)}{q(x)}$$

Importance sampling

- **Sample** $X = \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(n)}\}, (\mathbf{x}^{(i)} \sim q(\mathbf{x}))$
- **Importance sampling estimator** $\hat{s}_q$

$$\hat{s}_q = \frac{1}{n}\sum_{i=1}^{n}\frac{p(\mathrm{x}^{(i)})f(\mathrm{x}^{(i)})}{q(\mathrm{x}^{(i)})}$$

- **Minimum variance of** $\hat{s}_q$

$$Var(\hat{s}_q) = \frac{1}{n}Var\left(\frac{p(\mathrm{x})f(\mathrm{x})}{q(\mathrm{x})}\right)$$

Estimating the Partition Function

- **Estimate partition function $Z_B(\theta)$ of model M_B**
  - Use different probability distribution $p_A(\mathrm{x};\theta_A) = \frac{1}{Z_A(\theta)}\tilde{p}_A(\mathrm{x};\theta_A)$
  - Estimate the ratio $r = \frac{Z(\theta_B)}{Z(\theta_A)}$ using importance sampling
- **Sample $X = \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(K)}\}$ from $p_A(\mathrm{x};\theta_A)$**

Model M_A

$$p_A(\mathrm{x};\theta_A) = \frac{1}{Z_A}\tilde{p}_A(\mathrm{x};\theta_A)$$

Model M_B

$$p_B(\mathrm{x};\theta_B) = \frac{1}{Z_B}\tilde{p}_B(\mathrm{x};\theta_B)$$

- **Comparing model M_A with M_B**

$$\sum_i \log p_A(x^{(i)};\theta_A) - \sum_i \log p_B(x^{(i)};\theta_B) = \sum_i \log \frac{\tilde{p}_A(x^{(i)};\theta_A)}{\tilde{p}_B(x^{(i)};\theta_B)} - m \log \frac{Z(\theta_A)}{Z(\theta_B)}$$

  - Estimate $\log \frac{Z(\theta_A)}{Z(\theta_B)}$ using importance sampling

Estimating the Partition Function

- **Partition function $Z_B(\theta)$**

$$Z_B(\theta) = \int \tilde{p}_B(\mathrm{x};\theta)d\mathrm{x} = \int \frac{p_A(\mathrm{x};\theta)}{p_A(\mathrm{x};\theta)}\tilde{p}_B(\mathrm{x};\theta)d\mathrm{x} = Z_A(\theta)\int p_A(\mathrm{x};\theta)\frac{\tilde{p}_B(x;\theta)}{\tilde{p}_A(x;\theta)}d\mathrm{x}$$

- **Ratio r**

$$r = \frac{Z_B(\theta)}{Z_A(\theta)} = \int p_A(\mathrm{x};\theta)\frac{\tilde{p}_B(x;\theta)}{\tilde{p}_A(x;\theta)}d\mathrm{x}$$

- **Ratio Estimator $\hat{r}$ using sample $\boldsymbol{X} = \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(K)}\}$**

$$\hat{r} = \frac{1}{K}\sum_{k=1}^{K}\frac{\tilde{p}_B\left(\mathrm{x}^{(k)};\theta\right)}{\tilde{p}_A(\mathrm{x}^{(k)};\theta)}, \qquad \left(\mathrm{x}^{(k)} \sim p_A(\mathrm{x};\theta)\right)$$

Estimating the Partition Function

- **Estimate $Z_B(\theta)$**

$$\hat{Z}_B(\theta) = \frac{Z_A(\theta)}{K} \sum_{k=1}^{K} \frac{\tilde{p}_B\left(\mathrm{x}^{(k)};\theta\right)}{\tilde{p}_A\left(\mathrm{x}^{(k)};\theta\right)}, \qquad \left(\mathrm{x}^{(k)} \sim p_A(\mathrm{x};\theta)\right)$$

- **Minimize variance of $\hat{Z}_B(\theta)$**

$$Var\left(\hat{Z}_B(\theta)\right) = \frac{Z_A(\theta)}{K^2} \sum_{k=1}^{K} \left(\frac{\tilde{p}_B\left(\mathrm{x}^{(k)};\theta\right)}{\tilde{p}_A\left(\mathrm{x}^{(k)};\theta\right)} - \hat{Z}_B \right)^2$$

Thank you