E9: 309 Advanced Deep Learning

11-11-2020

http://leap.ee.iisc.ac.in/sriram/teaching/ADL2020/

Housekeeping

- ✴️ 1st mini-project
  - ✓ Deadlines
    - ★ Presentation on Nov19 and Nov20
      - ★ Your date allocation has been finalized
      - ★ Presentation and report template will be sent out this week.
        - ★ Report 1 page + references and tools
        - ★ Slides 4 slides for individual project and 6 slides for 2-member.

Recap of previous class

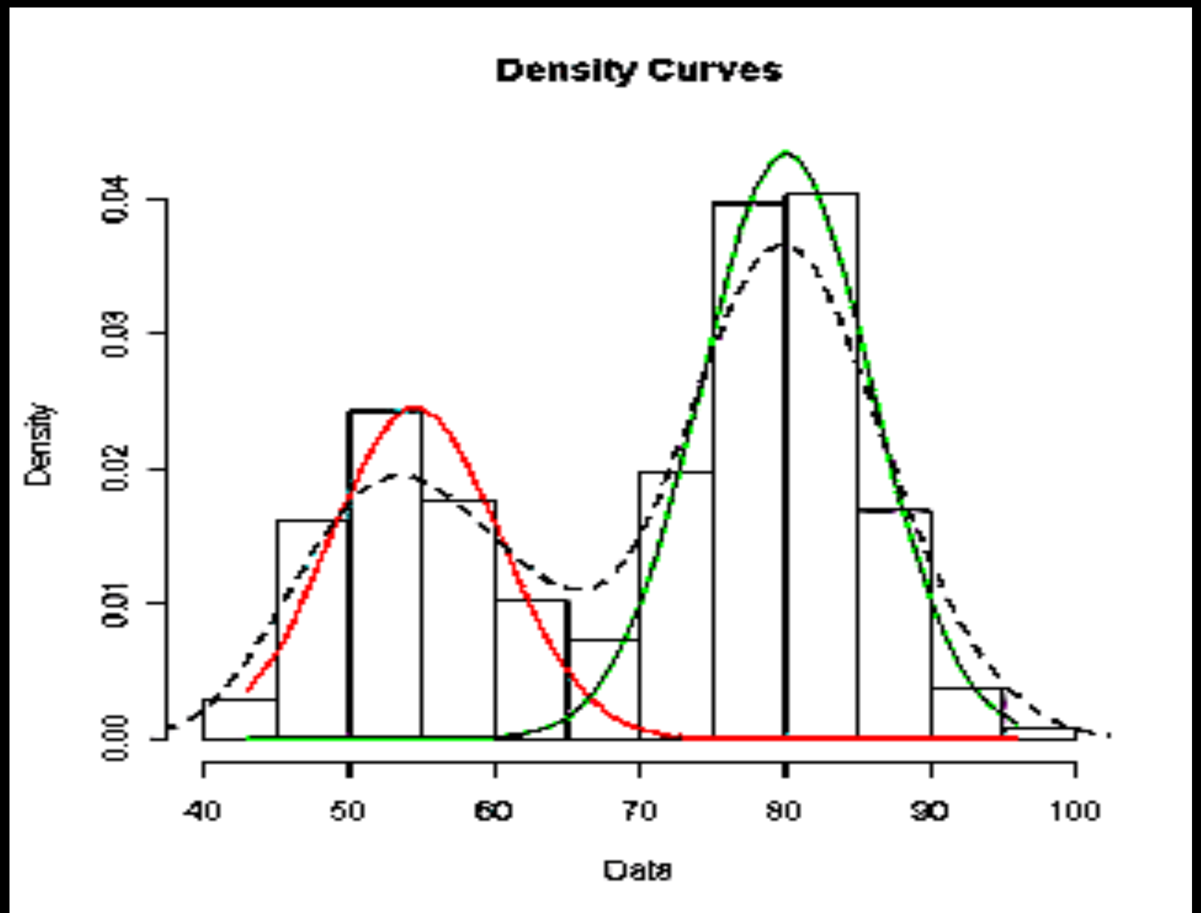
Representation learning/data-visualization

- ✴️ Learning a lower dimensional representation
  - ➡️ Unsupervised dimensionality reductions
    - ✓ Based on neighboorhood preservation
  - ➡️ t-SNE embeddings
    - ✓ visualization of neural network layers.

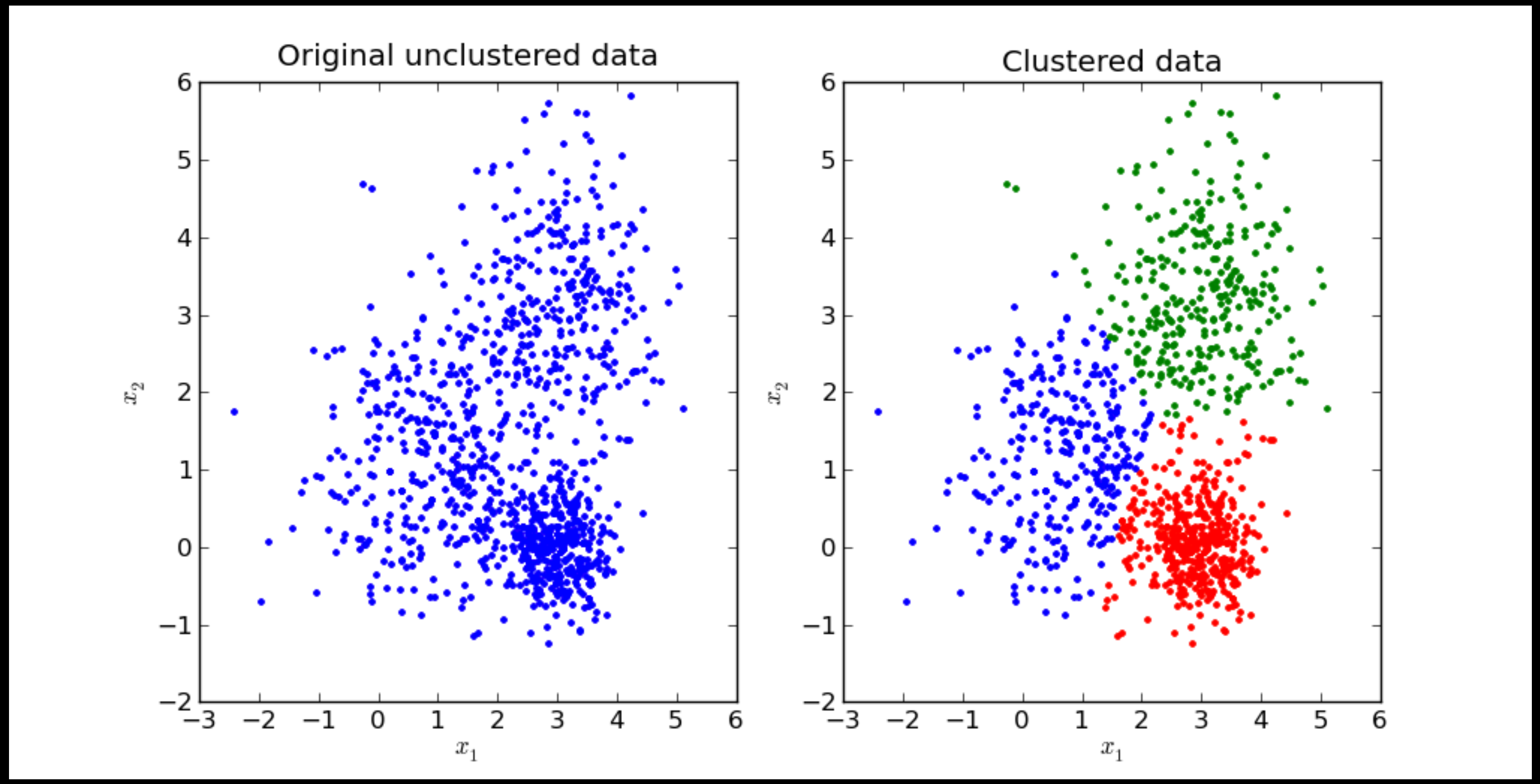
Unsupervised learning

- Developing models that do not need labels
  - May model the generation of data.
  - May allow generation of new data samples
- Broad strategies for unsupervised learning

Based on maximizing likelihood



Based on clustering



Boltzmann machine

[Deep Learning Book]
(Chap 20)

Binary data

- Energy based distribution of the data

$$p(\mathbf{x}) = \frac{exp(-E(\mathbf{x}))}{Z}$$

$$E(\mathbf{x}) = -\mathbf{x}^T\mathbf{U}\mathbf{x} - \mathbf{b}^T\mathbf{x}$$

$$\mathbf{x} \in \mathcal{B}^{D \times 1}$$

Z - scalar
normalization
consta

- The model parameters consists are learned by maximizing the likelihood.
- The data can be partitioned as visible and hidden units as well

$$\mathbf{x} = [\mathbf{v}^T\ \mathbf{h}^T]^T$$

$$x = \begin{bmatrix} \underline{v} \\ \underline{h} \end{bmatrix}$$

Restricted Boltzmann machine

$x = \begin{bmatrix} v \\ h \end{bmatrix}$

- Removing the self-connections between visible and hidden units.

$$p([\mathbf{v}\ \mathbf{h}]) = \frac{exp(-E([\mathbf{v}\ \mathbf{h}]))}{Z}$$

$$E([\mathbf{v}\ \mathbf{h}]) = -\mathbf{v}^T\mathbf{W}\mathbf{h} - b^T\mathbf{v} - \mathbf{c}^T\mathbf{h}$$

$$Z = \sum_{\mathbf{v}}\sum_{\mathbf{h}} exp(-E([\mathbf{v}\ \mathbf{h}]))$$

$x^T U x$

$U = \begin{bmatrix} U_1 & \frac{1}{2}W \\ \frac{1}{2}W^T & U_2 \end{bmatrix}$

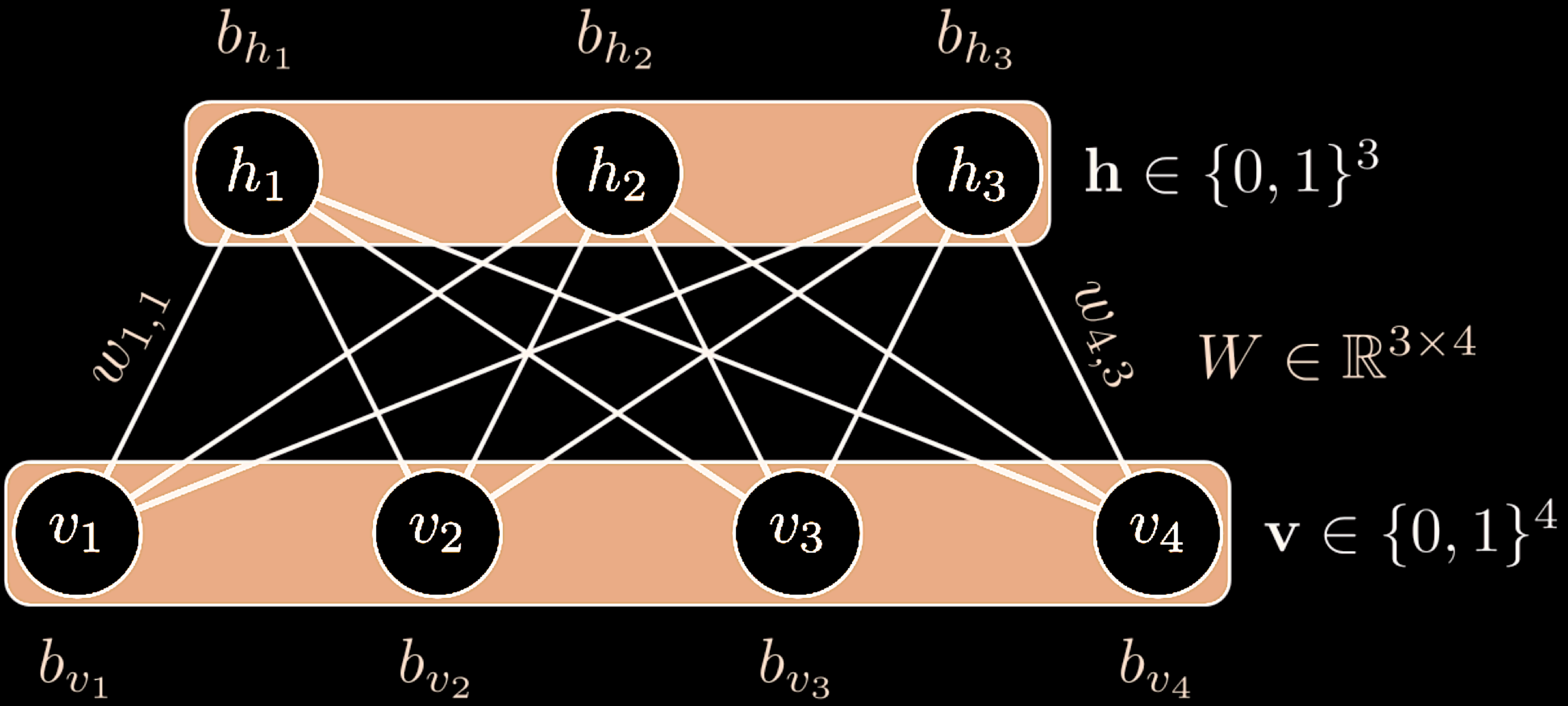
$\cancel{v^T U_1 v}$

$\cancel{h^T U_2 h} + v^T W h$

- The normalizing constant Z is called partition function.
  - The partition function is intractable to compute explicitly.

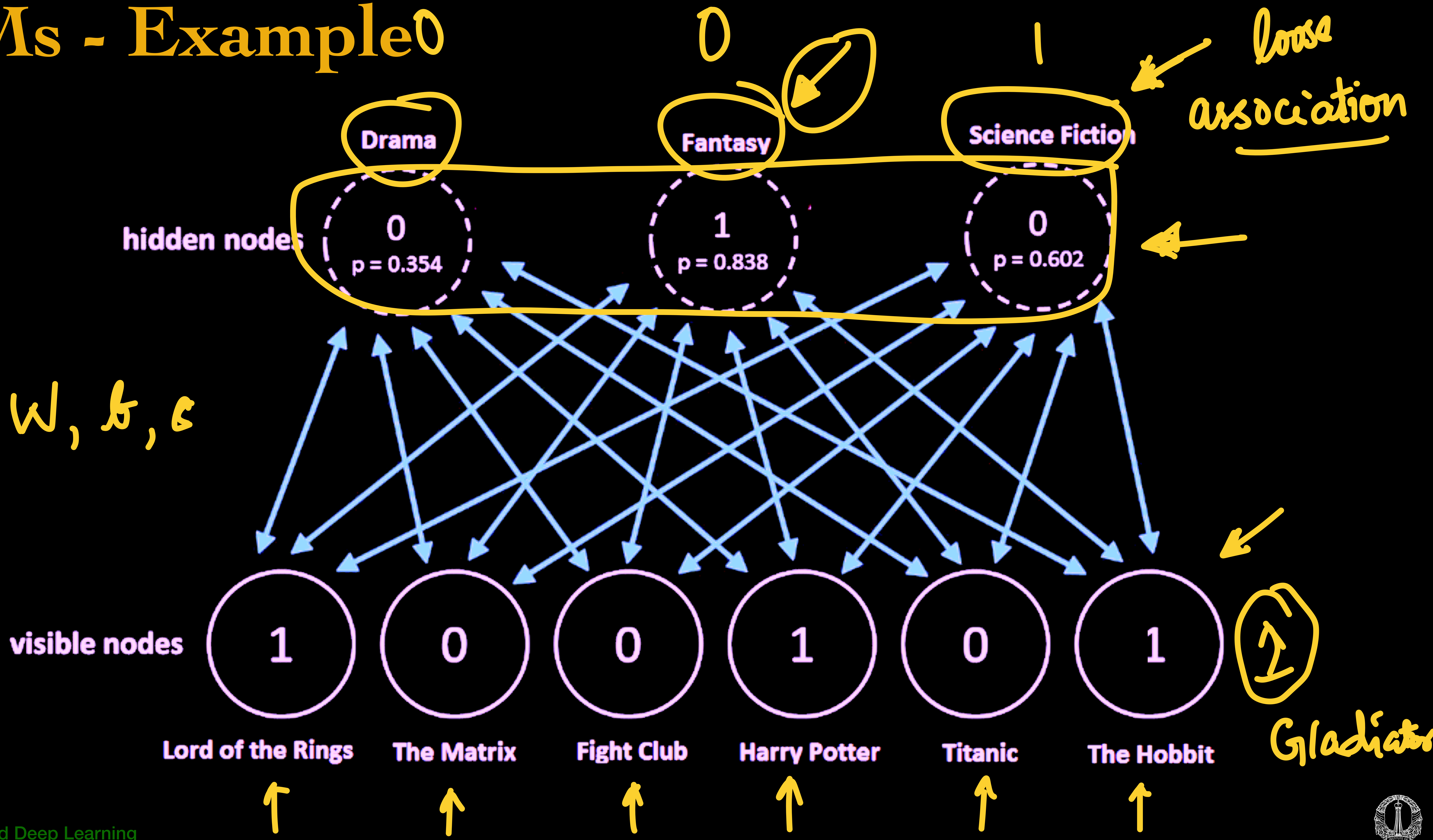
$p(x)$ is not explicitly computed.

RBM - Example



b_{h_1}
b_{h_2}
b_{h_3}
h_1
h_2
h_3
$\mathbf{h} \in \{0,1\}^3$
$w_{1,1}$
$w_{4,3}$
$W \in \mathbb{R}^{3\times 4}$
v_1
v_2
v_3
v_4
$\mathbf{v} \in \{0,1\}^4$
b_{v_1}
b_{v_2}
b_{v_3}
b_{v_4}

RBMs - Example



Conditional independence

- Conditional probability of hidden given visible

$$p(\mathbf{h}|\mathbf{v}) = \frac{p(\mathbf{v},\mathbf{h})}{p(\mathbf{v})}$$

$$= \frac{1}{Z}\frac{exp(\mathbf{v}^T\mathbf{W}\mathbf{h} + \mathbf{b}^T\mathbf{v} + \mathbf{c}^T\mathbf{h})}{p(\mathbf{v})}$$

const.

constant

$$= \frac{1}{Z'} exp(\mathbf{v}^T\mathbf{W}\mathbf{h} + \mathbf{c}^T\mathbf{h})$$

visible - observations

v, h

$h \in \mathbb{B}^{n_h \times 1}$

$H = h$

$V = v$

$$Z' = \frac{exp(b^T v)}{Z \cdot p(v)}$$

$$= \frac{1}{Z'} exp\left(\sum_{j=1}^{n_h} (v^T w_{:j}) h_j + c_j h_j\right)$$

$$= 1/Z' \; exp(v^T w_{:1} h_1 + c_1 h_1) \; exp(v^T w_{:2} h_2 + c_2 h_2) \times \ldots$$

$$p(h/v) = \frac{1}{Z_1'} \exp\{v^T W_1 h_1 + c_1 h_1\} \times$$

$$\frac{1}{Z_2'} \exp\{ \ldots \} \times \ldots$$

$$= \prod_{j=1}^{n_h} \underline{\underline{p(h_j/v)}}$$

$$p(h_j/v) = \frac{\exp(v^T W_{[:,j]} h_j + c_j h_j)}{Z_j'}$$

$$p(h_j = 0 \mid v) = \frac{1}{Z_j'}$$

$$p(h_j = 1 / v) + p(h_j = 0 / v) = 1$$

$$\frac{1}{Z_j'} \exp\{ v^T W_{[:,j]} + c_j \} + \frac{1}{Z_j'} = 1.$$

$$Z_j' = 1 + \exp\{ v^T W_{:,j} + c_j \}$$

$\underline{p(v/h)}$

Sigmoidal activation

- Conditional probability of hidden variable having a value of 1 given visible

$$p(h_j = 1|\mathbf{v})) = \frac{1}{Z'_j} exp(c_j + \mathbf{v}^T \mathbf{W}_{:,j})$$

$$= \frac{exp(c_j + \mathbf{v}^T \mathbf{W}_{:,j})}{1 + exp(c_j + \mathbf{v}^T \mathbf{W}_{:,j})}$$

$$\sigma(x) = \frac{1}{1+e^{-(x)}}$$

$$= \sigma((c_j + \mathbf{v}^T \mathbf{W}_{:,j}))$$

$$p(\mathbf{h}|\mathbf{v})) = \prod_j \sigma(\boxed{(2h_j - 1)}(c_j + \mathbf{v}^T \mathbf{W}_{:,j}))$$

$$\sigma(-z) = 1 - \sigma(z)$$

- Note that these are probabilities.

RBM - Training

visible bias

hidden bias

- Model parameters $\boxed{\Theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}}$ $\qquad \mathbf{x} = [\mathbf{v}^T \ \mathbf{h}^T]^T$
  - ✓ Learnt by maximizing the log-likelihood $log(p(\mathbf{x}; \Theta))$
  - ✓ Non-convex optimization.

$$\Theta^{n+1} = \Theta^{n} + \eta \frac{\partial \log p(x;\Theta)}{\partial \Theta}$$

- Gradient ascent based optimization

$$\frac{\partial log(p(\mathbf{x};\Theta))}{\partial \Theta} = \frac{\partial log(\tilde{p}(\mathbf{x};\Theta))}{\partial \Theta} - \frac{\partial log(Z(\Theta))}{\partial \Theta}$$

$$\frac{\partial log(Z(\Theta))}{\partial \Theta} = \frac{1}{Z}\frac{\partial Z(\Theta)}{\partial \Theta}$$

$$Z(\Theta) = \sum_{\mathbf{x}} \tilde{p}(\mathbf{x};\Theta)$$

$$p(x;\theta) = \frac{\boxed{\tilde{p}(x;\theta)}}{\boxed{Z(\theta)}}$$

$$x = \begin{bmatrix} v \\ h \end{bmatrix}$$

normalized

unnormalized

$$Z(\theta) = \sum_x \tilde{p}(x;\theta)$$

$$\underline{\underline{\exp(-E(x;\theta))}}$$

RBM - Training

$Z(\theta) = \sum_x \tilde{p}(x;\theta)$

- For exponential families

$$\boxed{\tilde{p}(\mathbf{x};\Theta) > 0 \quad \forall\ \mathbf{x}}$$

$$\frac{\partial log(Z(\Theta))}{\partial \Theta} = \frac{1}{Z}\sum_{\mathbf{x}} \frac{\partial \tilde{p}(\mathbf{x};\Theta)}{\partial \Theta}$$

$$\frac{\partial log(Z(\Theta))}{\partial \Theta} = \frac{1}{Z}\sum_{\mathbf{x}} \frac{\partial exp(log(\tilde{p}(\mathbf{x};\Theta)))}{\partial \Theta}$$

$\tilde{p} = exp(log\ \tilde{p})$

$$= \frac{1}{Z}\sum_{\mathbf{x}} \underbrace{exp(log(\tilde{p}(\mathbf{x};\Theta)))}_{\tilde{p}(x;\theta)} \left(\frac{\partial(log(\tilde{p}(\mathbf{x};\Theta)))}{\partial \Theta}\right)$$

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

$$= \sum_{\mathbf{x}} p(\mathbf{x};\Theta) \frac{\partial(log(\tilde{p}(\mathbf{x};\Theta)))}{\partial \Theta}$$

$$\frac{\partial}{\partial\theta}\log Z(\theta)$$

$$\sum_x p(x;\theta)\frac{\partial}{\partial\theta}\log\tilde{p}(x;\theta) = E_{x\sim p(x)}\left(\frac{\partial}{\partial\theta}\log\tilde{p}(x)\right)$$

$$\frac{\partial}{\partial\theta}\log p(x;\theta) = \underbrace{\frac{\partial}{\partial\theta}\log\tilde{p}(x;\theta)}_{\text{positive phase}} - \underbrace{E_{x\sim p(x)}\left[\frac{\partial}{\partial\theta}\log\tilde{p}(x;\theta)\right]}_{\text{negative phase}}$$

positive phase

negative phase

$$p(x;\theta)$$

RBM - Training

$$\frac{\partial}{\partial \theta} \log Z(\theta) = E_{x \sim p(x;\theta)}\left\{ \frac{\partial}{\partial \theta} \log \tilde{p}(x;\theta) \right\}$$

- Intractable to compute the exact gradient
  - Using approximations to expectations (computationally simple)
    - ✓ Based on sampling methods.
      - ★ Monte-carlo Markov Chain (MCMC) based approximation
      - ★ Resorting to Gibbs sampling.

Next lecture

→ Gibbs sampling

→ Gaussian RBM $\quad v \in \mathbb{R}^{n_v \times 1}$

→ Gaussian mixture model

→ Application

→ Deep Belief Network

$$\mathop{E}_{x \sim p(x)} \left[f(x) \right] \approx \frac{1}{n} \sum_{i=1}^{n} f(x_i)$$

$$x_i \sim p(x)$$