Restricted Boltzmann Machine: an introduction

A/Prof Richard Yi Da Xu
Yida.Xu@uts.edu.au
http://richardxu.com

University of Technology Sydney (UTS)

June 22, 2018

What is contained in here

- You should read my notes on Neural Networks Basics and Convolution Neural Networks first, then in this notes we have:
- Recurrent Neural Neworks
- Generative Adversarial Networks
- Restrictive Botzmann Machine
- Other fun stuff

Game theory

- **maximin value** of a player is the highest value that a player can be sure to get **without** knowing actions of the other players
- equivalently, it is lowest value the other players can force the player to receive when they know the player's action.

$$\underline{v_i} = \max_{a_i} \min_{a_{-i}} v_i(a_i, a_{-i})$$

- Calculating maximin value of a player is done in a worst-case approach: for each possible action of the player, we check all possible actions of the other players and determine the worst possible combination of actions, the one that gives player *i* smallest value.
  then, we determine which action player i can take in order to make sure that this smallest value is the highest possible

Generative Adversarial Training

- cost for discriminator

$$J^{(D)}(\theta^{(D)}, \theta^{(G)}) = -\frac{1}{2}\mathbb{E}_{\mathbf{x}\sim P_{\text{data}}} \log D(\mathbf{x}) - \frac{1}{2}\mathbb{E}_z \log(1 - D(G(z)))$$

- by training a discriminator, we are able to obtain an estimate of ratio at every $\mathbf{x}$:

$$\frac{P_{\text{data}}(\mathbf{x})}{P_{\text{model}}(\mathbf{x})}$$

- GANs make approximations based on **supervised learning** to estimate ratio of two densities
- This is what happen before training G properly:

$$\left(\text{when } G(z) \text{ does NOT look like data}\right) \implies \left(D(G(z)) \downarrow\right) \implies \left(\log(1 - D(G(z))) \uparrow\right)$$

- So our aim for G is to:

$$\left(\text{make } G(z) \text{ look like data}\right) \implies \left(D(G(z)) \uparrow\right) \implies \left(\log(1 - D(G(z))) \downarrow\right) \implies \min_G$$

Adversarial Training

- a prior on input noise variables $z \sim p_z(z)$,
- G is differentiable function with parameters θ_g it transforms $z \to x$ space.
- $D(x; \theta_d)$ outputs a single scalar. Represents the probability x came from data rather than p_g.
- Simultaneously train both D and G:
  - Train D to maximize the probability of assigning correct label to both training examples and samples from G
  - Train G to minimize $\log(1 - D(G(z)))$

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

- This is what happen before training G properly:

$$\Big(\text{when } G(z) \text{ does NOT look like data}\Big) \implies \Big(D(G(z)) \downarrow\Big) \implies \Big(\log(1 - D(G(z))) \uparrow\Big)$$

- So our aim for G is to:

$$\Big(\text{make } G(z) \text{ look like data}\Big) \implies \Big(D(G(z)) \uparrow\Big) \implies \Big(\log(1 - D(G(z))) \downarrow\Big) \implies \min_G$$

Adversarial Training algorithm

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

for *number of training iterations* **do**
  for *k steps* **do**
    Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_z(z)$;
    Sample minibatch of m examples $\{x^{(1)}, \dots, x^{(m)}\}$ from $p_{\text{data}}(x)$;
    Update the discriminator by ascending its stochastic gradient:;

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D(G(z^{(i)}))\right) \right]$$

  end
  Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_z(z)$;
  Update the generator by descending its stochastic gradient;

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D(G(z^{(i)}))\right)$$

end

Minimizing Negative Log-Likelihood

- Think about the following MLE or Minimizing Negative Log-Likelihood:

$$
\begin{aligned}
p_{\mathbf{x}}(\theta) &= \prod_{i=1}^{N} \frac{1}{Z(\theta)} f_{x_i}(\theta) = \frac{1}{Z(\theta)^n} \prod_{i=1}^{N} f_{x_i}(\theta) \qquad \text{where } Z(\theta) = \int_x f_\theta(x)\mathrm{d}x \\
\log[p_{\mathbf{x}}(\theta)] &= \sum_{i=1}^{N} \log(f_{x_i}(\theta)) - n\log(Z(\theta)) \\
\mathcal{L}(\theta) &= -\log[p_{\mathbf{x}}(\theta)] = \log(Z(\theta)) - \frac{1}{N}\sum_{i=1}^{N} \log(f_{x_i}(\theta))
\end{aligned}
$$

- The problem is that we don't have an analytic form of $Z(\theta)$.

Contrast Divergence (1)

$$\mathcal{L}(\theta) = -\log[p_{\mathbf{X}}(\theta)] = \log(Z(\theta)) - \frac{1}{N}\sum_{i=1}^{N}\log(f_{x_i}(\theta))$$

$$\begin{aligned}
\implies \frac{\partial \mathcal{L}(\theta)}{\theta} &= \frac{\partial \log(Z(\theta))}{\partial\theta} - \frac{1}{N}\sum_{i=1}^{N}\frac{\partial \log(f_{x_i}(\theta))}{\partial\theta} \\
&= \frac{1}{Z(\theta)}\frac{\partial Z(\theta)}{\partial\theta} - \frac{1}{N}\sum_{i=1}^{N}\frac{\partial \log(f_{x_i}(\theta))}{\partial\theta} \\
&= \frac{1}{Z(\theta)}\frac{\partial}{\partial\theta}\int_x f_x(\theta)\mathrm{d}x - \frac{1}{N}\sum_{i=1}^{N}\frac{\partial \log(f_{x_i}(\theta))}{\partial\theta} \\
&= \frac{1}{Z(\theta)}\int_x \frac{\partial f_x(\theta)}{\partial\theta}\mathrm{d}x - \frac{1}{N}\sum_{i=1}^{N}\frac{\partial \log(f_{x_i}(\theta))}{\partial\theta}
\end{aligned}$$

Here comes the trick:

$$f_x(\theta)\frac{\partial \log(f_x(\theta))}{\partial\theta} = f_x(\theta)\frac{1}{f_x(\theta)}\frac{\partial f_x(\theta)}{\partial\theta} = \frac{\partial f_x(\theta)}{\partial\theta}$$

substitute into, one get:

$$\begin{aligned}
\frac{\partial\mathcal{L}(\theta)}{\theta} \propto \frac{\partial - \log[p_{\mathbf{x}}(\theta)]}{\theta} &= \frac{1}{Z(\theta)}\int_x \frac{\partial f_x(\theta)}{\partial\theta}\mathrm{d}x - \frac{1}{N}\sum_{i=1}^{N}\frac{\partial\log(f_{x_i}(\theta))}{\partial\theta} \\
&= \frac{1}{Z(\theta)}\int_x f_x(\theta)\frac{\partial\log(f_x(\theta))}{\partial\theta}\mathrm{d}x - \frac{1}{N}\sum_{i=1}^{N}\frac{\partial\log(f_{x_i}(\theta))}{\partial\theta} \\
&= \underbrace{\int_x \frac{\partial\log(f_x(\theta))}{\partial\theta}p_\theta(x)\mathrm{d}x}_{\text{population mean of } \left\{\frac{\partial\log(f_x(\theta))}{\partial\theta}\right\}} - \underbrace{\frac{1}{N}\sum_{i=1}^{N}\frac{\partial\log(f_{x_i}(\theta))}{\partial\theta}}_{\text{sample mean of } \left\{\frac{\partial\log(f_{x_i}(\theta))}{\partial\theta}\right\}}
\end{aligned}$$

Simple CD example in estimating Gaussian mean μ

$$\frac{\partial \log(f_x(\theta))}{\partial \theta} = \frac{\partial\left(\frac{-\tau}{2}(x-\mu)^2\right)}{\partial \mu} = \tau(x-\mu)$$

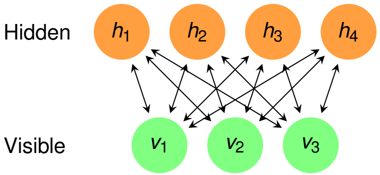
$$= \underbrace{\int_x \frac{\partial \log(f_x(\theta))}{\partial \theta} p_x(\theta)\mathrm{d}x}_{\text{population mean of } \left\{\frac{\partial \log(f_x(\theta))}{\partial \theta}\right\}} - \underbrace{\frac{1}{N}\sum_{i=1}^{N} \frac{\partial \log(f_{x_i}(\theta))}{\partial \theta}}_{\text{sample mean of } \left\{\frac{\partial \log(f_{x_i}(\theta))}{\partial \theta}\right\}}$$

$$= \int_x \tau(x-\mu)p_\theta(x)\mathrm{d}x - \frac{1}{N}\sum_{i=1}^{N}\tau(x_i-\mu)$$

$$= -\frac{1}{N}\sum_{i=1}^{N}\tau(x_i-\mu)$$

$$= \tau\mu - \frac{1}{N}\sum_{i=1}^{N}\tau x_i = \tau\left(\mu - \frac{1}{N}\sum_{i=1}^{N} x_i\right)$$

Restrictive Botzmann Machine



Define:

$$
\begin{aligned}
E(\mathbf{v}, \mathbf{h}) &= -b^\top \mathbf{v} - c^\top \mathbf{h} - \mathbf{v}^\top W \mathbf{h} \\
&= -\sum_j b_j v_j - \sum_i c_i h_i - \sum_i \sum_j v_j W_{ij} h_i
\end{aligned}
$$

$$
p(\mathbf{v}, \mathbf{h}) = \exp(-E(\mathbf{v}, \mathbf{h})) = \exp\left(b^\top \mathbf{v} + c^\top \mathbf{h} + \mathbf{v}^\top W \mathbf{h}\right)
$$

- There are two separate offset parameters: b and c, associated with $\mathbf{v}$ and $\mathbf{h}$ respectively.
- Note that there is no interconnecting terms between elements of $\mathbf{v}$ and $\mathbf{h}$. Otherwise, there will be a term $\mathbf{v}^\top W_v \mathbf{v}$ and $\mathbf{h}^\top W_h \mathbf{h}$
- In this presentation, $\mathbf{v}$ and $\mathbf{h}$ are binary arrays.
- $\mathbf{v}$ and $\mathbf{h}$ can take other values, for example Softmax and Gaussian.

RBM Marginal

$$p(\mathbf{v},\mathbf{h}) = \exp(-E(\mathbf{v},\mathbf{h})) = \exp\left(b^\top \mathbf{v} + c^\top \mathbf{h} + \mathbf{v}^\top W\mathbf{h}\right) = \exp\left(\sum_j b_j v_j + \sum_i c_i h_i + \sum_i \sum_j v_j W_{ij} h_i\right)$$

$$\begin{aligned}
p(\mathbf{v}) &= \frac{1}{Z}\sum_{\mathbf{h}} p(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\sum_{\mathbf{h}} \exp\left(b^\top \mathbf{v} + c^\top \mathbf{h} + \mathbf{v}^\top W\mathbf{h}\right) \\
&= \frac{1}{Z}\exp(b^\top \mathbf{v}) \sum_{\mathbf{h}} \exp\left(c^\top \mathbf{h} + \mathbf{v}^\top W\mathbf{h}\right) \\
&= \frac{1}{Z}\exp(b^\top \mathbf{v}) \sum_{h_1}\sum_{h_2}\cdots\sum_{h_N} \exp^{\sum_i c_i \underbrace{h_i} + \sum_i \sum_j v_j W_{ij} \underbrace{h_i}} \\
&= \frac{1}{Z}\exp(b^\top \mathbf{v}) \sum_{h_1}\sum_{h_2}\cdots\sum_{h_N} \exp^{\sum_i \underbrace{h_i} \left(c_i + \sum_j v_j W_{ij}\right)} \\
&= \frac{1}{Z}\exp(b^\top \mathbf{v}) \sum_{h_1} \exp^{h_1\left(c_1 + \sum_j w_{1j} v_j\right)} \sum_{h_2} \exp^{h_2\left(c_i + \sum_j w_{2j} v_j\right)} \dots \sum_{h_N} \exp^{h_N\left(c_N + \sum_j w_{Nj} v_j\right)} \\
&= \frac{1}{Z}\exp^{\sum_j b_j v_j} \prod_{i=1}^{N} \sum_{h_i} \exp^{h_i\left(c_i + \sum_j w_{ij} v_j\right)} \\
&= \frac{1}{Z}\prod_j \exp^{b_j v_j} \prod_{i=1}^{N} \left(1 + \exp^{c_i + \sum_j w_{ij} v_j}\right)
\end{aligned}$$

RBM conditional

$$p(\mathbf{v}, \mathbf{h}) = \exp(-E(\mathbf{v}, \mathbf{h})) = \exp\left(b^\top \mathbf{v} + c^\top \mathbf{h} + \mathbf{v}^\top W \mathbf{h}\right) = \exp\left(\sum_j b_j v_j + \sum_i c_i h_i + \sum_i \sum_j v_j W_{ij} h_i\right)$$

$$\begin{aligned}
p(V_l = 1|\mathbf{h}) &= \frac{p(V_l = 1, \mathbf{h})}{p(\mathbf{h})} = \frac{p(V_l = 1, \mathbf{h})}{\sum_{v_l} p(V_l = 1, \mathbf{h})} \\
&= \frac{\exp\left(1 \times b_l + \sum_i 1 \times W_{il} h_i\right)}{\sum_{v_l} \exp\left(b_l v_l + \sum_i v_l W_{il} h_i\right)} \qquad \text{reduce } \sum_j \text{ into a single term} \\
&= \frac{\exp\left(b_l + \sum_i W_{il} h_i\right)}{\underbrace{1}_{v_l=0} + \underbrace{\exp\left(b_l + \sum_i W_{il} h_i\right)}_{v_l=1}} \\
&= \sigma\left(b_l + \sum_i W_{il} h_i\right)
\end{aligned}$$

By symmetry,

$$p(H_i = 1|\mathbf{v}) = \sigma\left(c_i + \sum_j v_j W_{ij}\right)$$

The **derivative** of general Markov Random Field Likelihood

- In here, we did NOT use the structure of RBM, i.e., $p(\mathbf{v},\mathbf{h}) = \exp\left(b^\top \mathbf{v} + c^\top \mathbf{h} + \mathbf{v}^\top W\mathbf{h}\right) = \exp\left(\sum_j b_j v_j + \sum_i c_i h_i + \sum_i \sum_j v_j W_{ij} h_i\right)$:

$$
\begin{aligned}
\mathcal{L}_{\mathbf{v}}(\theta) = \log(p(\mathbf{v})) &= \log\left(\sum_{\mathbf{h}} \exp^{-E(\mathbf{v},\mathbf{h})}\right) - \log(Z) \\
&= \log\left(\sum_{\mathbf{h}} \exp^{-E(\mathbf{v},\mathbf{h})}\right) - \log\left(\sum_{\mathbf{h},\mathbf{v}} \exp^{-E(\mathbf{v},\mathbf{h})}\right) \\
\Longrightarrow \frac{\partial \mathcal{L}_{\mathbf{v}}(\theta)}{\partial\theta} &= \frac{1}{\sum_{\mathbf{h}} \exp^{-E(\mathbf{v},\mathbf{h})}} \sum_{\mathbf{h}} \frac{\partial \exp^{-E(\mathbf{v},\mathbf{h})}}{\partial\theta} - \frac{1}{\sum_{\mathbf{h},\mathbf{v}} \exp^{-E(\mathbf{v},\mathbf{h})}} \sum_{\mathbf{h},\mathbf{v}} \frac{\partial \exp^{-E(\mathbf{v},\mathbf{h})}}{\partial\theta} \\
&= -\frac{1}{\sum_{\mathbf{h}} \exp^{-E(\mathbf{v},\mathbf{h})}} \sum_{\mathbf{h}} \exp^{-E(\mathbf{v},\mathbf{h})} \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} + \frac{1}{\sum_{\mathbf{h},\mathbf{v}} \exp^{-E(\mathbf{v},\mathbf{h})}} \sum_{\mathbf{h},\mathbf{v}} \exp^{-E(\mathbf{v},\mathbf{h})} \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} \\
&= -\sum_{\mathbf{h}} \frac{\exp^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{h}} \exp^{-E(\mathbf{v},\mathbf{h})}} \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} + \sum_{\mathbf{h},\mathbf{v}} \frac{\exp^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{h},\mathbf{v}} \exp^{-E(\mathbf{v},\mathbf{h})}} \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} \\
&= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta} + \sum_{\mathbf{h},\mathbf{v}} p(\mathbf{v},\mathbf{h}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\theta}
\end{aligned}
$$

$$
p(\mathbf{h}|\mathbf{v}) = \frac{p(\mathbf{v},\mathbf{h})}{p(\mathbf{v})} = \frac{\frac{1}{Z}\exp^{-E(\mathbf{v},\mathbf{h})}}{\frac{1}{Z}\sum_{\mathbf{h}} \exp^{-E(\mathbf{v},\mathbf{h})}} = \frac{\exp^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{h}} \exp^{-E(\mathbf{v},\mathbf{h})}}
$$

note that the two Z are equal

The **derivative** of RBM Likelihood

$$p(\mathbf{v}, \mathbf{h}) = \exp(-E(\mathbf{v}, \mathbf{h})) = \exp\left(b^\top \mathbf{v} + c^\top \mathbf{h} + \mathbf{v}^\top W\mathbf{h}\right) = \exp\left(\sum_j b_j v_j + \sum_i c_i h_i + \sum_i \sum_j v_j W_{ij} h_i\right)$$

$$E(\mathbf{v}, \mathbf{h}) = -b^\top \mathbf{v} - c^\top \mathbf{h} - \mathbf{v}^\top W\mathbf{h}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}_{\mathbf{v}}(\theta)}{\partial \theta} &= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \theta} + \sum_{\mathbf{h},\mathbf{v}} p(\mathbf{v}, \mathbf{h}) \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial \theta} \\
\implies \frac{\partial \mathcal{L}_{\mathbf{v}}(\theta)}{\partial w_{ij}} &= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial w_{ij}} + \sum_{\mathbf{h},\mathbf{v}} p(\mathbf{v}, \mathbf{h}) \frac{\partial E(\mathbf{v}, \mathbf{h})}{\partial w_{ij}} \\
&= +\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) v_j h_i - \sum_{\mathbf{h},\mathbf{v}} p(\mathbf{v}, \mathbf{h}) v_j h_i \qquad \textbf{note the sign change} \\
&= \underline{\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) v_j h_i} - \sum_{\mathbf{v}} p(\mathbf{v}) \underline{\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) v_j h_i} \\
&= p(H_i = 1|\mathbf{v}) v_j - \sum_{\mathbf{v}} p(\mathbf{v}) p(H_i = 1|\mathbf{v}) v_j
\end{aligned}$$

Because: $\underline{\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) v_j h_i} = \sum_{h_1} \cdots \sum_{h_N} \prod_{k=1}^{N} p(h_k|\mathbf{v}) v_j h_i = \sum_{h_i} p(h_i|\mathbf{v}) v_j h_i \times \underbrace{\sum_{\mathbf{h}_{k\neq i}} \prod_{k\neq i} p(h_k|\mathbf{v})}_{=1}$

$$= \sum_{h_i} p(h_i|\mathbf{v}) v_j h_i = p(H_i = 1|\mathbf{v}) v_j = \sigma\left(c_i + \sum_j v_j W_{ij}\right) v_j$$

Average derivative of RBM Likelihood over data

$$
\begin{aligned}
\frac{\partial \mathcal{L}_{\mathbf{v}}(\theta)}{\partial w_{ij}} &= \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) v_j h_i - \sum_{\mathbf{h},\mathbf{v}} p(\mathbf{v},\mathbf{h}) v_j h_i \\
&= p(H_i = 1|\mathbf{v}) v_j - \sum_{\mathbf{v}} p(\mathbf{v}) p(H_i = 1|\mathbf{v}) v_j
\end{aligned}
$$

- when we are given a set of observed **v**:

$$
\begin{aligned}
\frac{1}{N} \sum_{\mathbf{v} \in S} \frac{\partial \mathcal{L}_{\mathbf{v}}(\theta)}{\partial w_{ij}} &= \frac{1}{N} \sum_{\mathbf{v} \in S} \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}) v_j h_i - \sum_{\mathbf{h},\mathbf{v}} p(\mathbf{v},\mathbf{h}) v_j h_i \\
&= \frac{1}{N} \sum_{\mathbf{v} \in S} \left(\mathbb{E}_{p(\mathbf{h}|\mathbf{v})}[v_j h_i] - \mathbb{E}_{p(\mathbf{h},\mathbf{v})}[v_j h_i] \right) \\
&= \langle v_j h_i \rangle_{p(\mathbf{h}|\mathbf{v}) q(\mathbf{v})} - \langle v_j h_i \rangle_{p(\mathbf{h},\mathbf{v})}
\end{aligned}
$$

where $q(\mathbf{v})$ is the sample distribution

- without going through the normal **contrast divergence equation**, we put RBM in the CD form above:

$$
\frac{\partial - \mathcal{L}_{\mathbf{v}}(\theta)}{\partial w_{ij}} \propto \langle v_j h_i \rangle_{p(\mathbf{h},\mathbf{v})} - \langle v_j h_i \rangle_{p(\mathbf{h}|\mathbf{v}) q(\mathbf{v})}
$$

- **Exercise** how complex is $\langle v_j h_i \rangle_{p(\mathbf{h}|\mathbf{v}) q(\mathbf{v})}$? say **h** and **v** each have 100 nodes?
- **Exercise** how can we deal with such complexity?

RBM LLE via Contrast Divergence

the **answer** is to use Gibbs sampling: In each step of Gradient Descend, one performs the following:

- Let $\mathbf{v}^{(0)} = \mathbf{v}$
- Obtain a new set of Monte-Carlo sampled $\mathbf{v}$ iteratively:
  - sample $h^{(t)} \sim p(h_i | \mathbf{v}^{(t)})$ sample $v_j^{(t+1)} \sim p(v_j | \mathbf{h}^{(t)})$
  - until we obtain $\mathbf{v}^{(k)}$
- Update parameters $\{W_{i,j}\}$, $\{b_j\}$ and $\{c_i\}$ as the gradients:

$$
\begin{aligned}
\frac{\partial \mathcal{L}_{\mathbf{v}}(\theta)}{\partial W_{i,j}} &\approx p(H_i = 1 | \mathbf{v}^{(k)}) v_j^{(k)} - p(H_i = 1 | \mathbf{v}^{(0)}) v_j^{(0)} \\
\frac{\partial \mathcal{L}_{\mathbf{v}}(\theta)}{\partial b_j} &\approx v_j^{(k)} - v_j^{(0)} \\
\frac{\partial \mathcal{L}_{\mathbf{v}}(\theta))}{\partial c_i} &\approx p(H_i = 1 | \mathbf{v}^{(k)}) - p(H_i = 1 | \mathbf{v}^{(0)})
\end{aligned}
$$

RBM Collaborative Filtering

- **each user** can rate one of the m available movies, with a score between $\{1 \dots K\}$
- therefore, **each user** has a V, observed binary indicator matrix sized $K \times m$
- with $v_i^k = 1$ if a user rated movie i as k and 0 otherwise.
- it's a **softmax** function with $\sum_{k=1}^K p(v_i^k = 1|\mathbf{h}) = 1$:

$$p(v_i^k = 1|\mathbf{h}) = \frac{\exp\left(b_i^k + \sum_{j=1}^F h_j W_{ij}^k\right)}{\sum_{k=1}^K \exp\left(b_i^k + \sum_{j=1}^F h_j W_{ij}^k\right)} = \frac{\exp\left(b_i^k + W_{i,:}^k \mathbf{h}\right)}{\sum_{k=1}^K \exp\left(b_i^l + W_{i,:}^k \mathbf{h}\right)}$$

- **each user** has $\mathbf{h} \in \{0,1\}^F$, a binary values of hidden variables
- thought of as representing stochastic binary features that have different values for different users:

$$p(h_j = 1|\mathbf{V}) = \sigma\left(b_j + \sum_{i=1}^m \sum_{k=1}^K v_i^k W_{ij}^k\right) = \sigma\left(b_j + \sum_{k=1}^K (W_{:,j}^k)^\top \mathbf{v}^k\right)$$

Recommendation via RBM

- traditional RBM joint energy

$$E(\mathbf{v},\mathbf{h}) = -\sum_i^m b_i v_i - \sum_j^F b_j h_j - \sum_i^m \sum_j^F v_i W_{ij} h_j$$

- **Exercise** in terms of recommendation engine, how is traditional RBM useful?
- In recommendation setting with a rating range, it has changed to:

$$E(\mathbf{v},\mathbf{h}) - \sum_i^m \sum_{k=1}^K b_i v_i^k - \sum_j^F b_j h_j - \sum_i^m \sum_j^F \sum_{k=1}^K v_i W_{ij}^k h_j v_i^k$$