Deep Learning: The Rest of Topics

A/Prof Richard Yi Da Xu
Yida.Xu@uts.edu.au
Wechat: aubedata
https://github.com/roboticcam/machine-learning-notes

University of Technology Sydney (UTS)

February 18, 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

Recurrent Neural Networks

$$h_t = \tanh(\underbrace{Ux_t + Wh_{t-1}}_{z_t}) \qquad \hat{y}_t = \text{softmax}(Vh_t)$$

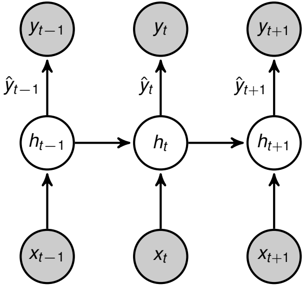
- The overall loss can be defined as cross entropy:

$$\mathcal{C}(y, \hat{y}) = \sum_t \mathcal{C}_t(y_t, \hat{y}_t) = -\sum_t \sum_{i \in \mathbb{S}} y_t \log \hat{y}_t$$

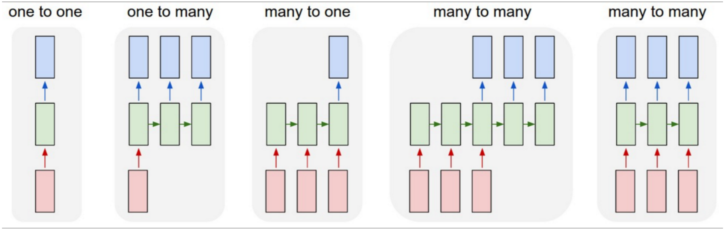
- The overall loss can also be defined as sum of square error:

$$\mathcal{C}(y, \hat{y}) = \sum_t \mathcal{C}_t(y_t, \hat{y}_t) = \sum_t \sum_{\mathbb{S}} (y_{t,i} - \hat{y}_{t,i})^2$$

- It has t individual cost functions as oppose to just a single one in the standard neural network.



How may we labelled a RNN?



- Each configuration serves a different purpose
- Think about the scenarios for their use!

Back propagation for $\frac{\partial \mathcal{C}_t}{\partial V}$

$$h_t = \tanh(\underbrace{Ux_t + Wh_{t-1}}_{z_t}) \qquad \hat{y}_t = \mathrm{softmax}(\underbrace{Vh_t}_{b_t})$$

$$\mathcal{C}(y, \hat{y}) = \sum_t \mathcal{C}_t(y_t, \hat{y}_t) = -\sum_t \sum_{\mathbb{S}} y_t \log \hat{y}_t$$

where $\mathbb{S}$ is the output space, e.g., all the words we try to predict.

$$\begin{aligned}
\frac{\partial \mathcal{C}_t(y_t, \hat{y}_t)}{\partial V} &= \frac{\partial \mathcal{C}_t(y_t, \hat{y}_t)}{\partial b_t} \frac{\partial b_t}{\partial V} \\
&= \frac{\partial \left(-\sum_{\mathbb{S}} y_t \log \hat{y}_t\right)}{\partial b_t} \times \underbrace{\frac{\partial b_t}{\partial V}}_{\text{a vector}} \\
&= (\hat{y}_t - y_t) h_t^\top
\end{aligned}$$

Back propagation for $\frac{\partial \mathcal{C}_t}{\partial W}$

$$h_t = \tanh(\underbrace{Ux_t + Wh_{t-1}}_{z_t}) \qquad \hat{y}_t = \text{softmax}(\underbrace{Vh_t}_{b_t})$$

$$\mathcal{C}(y, \hat{y}) = \sum_t \mathcal{C}_t(y_t, \hat{y}_t) = -\sum_t \sum_{\mathbb{S}} y_t \log \hat{y}_t$$

- Looking at **individual** cost term $\mathcal{C}_t$:

$$\frac{\partial \mathcal{C}_t}{\partial W} = \left(\frac{\partial \mathcal{C}_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial h_t}\right)\frac{\partial h_t}{\partial W} = \left(\frac{\partial \mathcal{C}_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial h_t}\right)\sum_{k=0}^{t}\frac{\partial h_t}{\partial h_k}\frac{\partial h_k}{\partial W}$$

- when performing $\frac{\partial h_t}{\partial W}$, we need to **sum** over all intermediate latent nodes, i..e.,

$$\left(\frac{\partial h_t}{\partial h_1}\frac{\partial h_1}{\partial W}\right) + \left(\frac{\partial h_t}{\partial h_2}\frac{\partial h_2}{\partial W}\right) + \cdots + \left(\frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial W}\right)$$

- rewrite it to fill in the gap with chain rule:

$$\frac{\partial \mathcal{C}_t}{\partial W} = \sum_{k=0}^{t}\frac{\partial \mathcal{C}_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial h_t}\left(\prod_{j=k+1}^{t}\frac{\partial h_j}{\partial h_{j-1}}\right)\frac{\partial h_k}{\partial W}$$

- we need to sum over all $\mathcal{C}_t$

Back propagation for $\frac{\partial \mathcal{C}_t}{\partial W}$ (1)

$$h_t = \tanh(\underbrace{Ux_t + Wh_{t-1}}_{z_t}) \qquad \hat{y}_t = \text{softmax}(Vh_t)$$

$$\begin{aligned}
\frac{\partial \mathcal{C}_t}{\partial W} &= \sum_{k=0}^{t} \frac{\partial \mathcal{C}_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \left(\prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_k}{\partial W} \\
&= \sum_{k=0}^{t} \frac{\partial \mathcal{C}_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \left(\prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_k}{\partial z_k} \frac{\partial z_k}{\partial W}
\end{aligned}$$

- The following has $t+1$ term, each with varying length due to the product term.
- Derivations can be understood better: $h_2 \left(\underbrace{c_2 + W(h_1(c_1 + W)}_{z_2} \right)$

$$\begin{aligned}
&\frac{\partial h_2 \left(c_2 + W(h_1(c_1 + W)\right)}{\partial W} \\
&= h_2'(c_2 + W(f(c_1 + W)) \frac{\partial (c_1 + W(h_1(c_1 + W))}{\partial W} && \text{using chain rule} \\
&= h_2'(c_2 + W(f(c_1 + W)) \left(h_1(c_1 + W) + W h_1'(c_1 + W) \right) && \text{using product rule} \\
&= h_2'(c_2 + W(f(c_1 + W)) h_1(c_1 + W) + h_2'(c_2 + W(h(c_1 + W)) W h_1'(c_1 + W) \\
&= \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial W} + \frac{\partial h_2}{\partial z_2} \frac{\partial z_2}{\partial h_1} \frac{\partial h_1}{\partial W} = \frac{\partial h_2}{\partial W} + \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W}
\end{aligned}$$

Gradient Vanishing and/or Explosion

$$\frac{\partial \mathcal{C}_t}{\partial W} = \sum_{k=0}^{t} \frac{\partial \mathcal{C}_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \left(\prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_k}{\partial z_k} \frac{\partial z_k}{\partial W}$$

before

$$h_t = \tanh(Ux_t + Wh_{t-1})$$
$$\hat{y}_t = \mathrm{softmax}(Vh_t)$$

hard to analyse $\frac{\partial h_t}{\partial h_k}$

alternative

$$h_t = Ux_t + Wf(h_{t-1})$$
$$\hat{y}_t = Vf(h_t)$$

easier to analyse $\frac{\partial h_t}{\partial h_k}$

In alternative represenation:

$$\frac{\partial h_t}{\partial h_k} = \prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} = \prod_{j=k+1}^{t} W \times \mathrm{diag}[f'(h_{j-1})]$$

This is because:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} W_{1,1} & W_{1,2} \\ W_{2,1} & W_{2,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} w_{1,1}x_1 + w_{1,2}x_2 \\ w_{2,1}x_1 + w_{2,2}x_2 \end{bmatrix} \Longrightarrow \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} = W$$

Gradient vanishing and/or exploding: Matrix norm

Define matrix norm from vector norm:

$$\|A\| = \sup\{ \underbrace{\|Ax\|}_{\text{vector norm}} : x \in \mathbb{R}^n \text{ with } \underbrace{\|x\|}_{\text{vector norm}} = 1\}$$

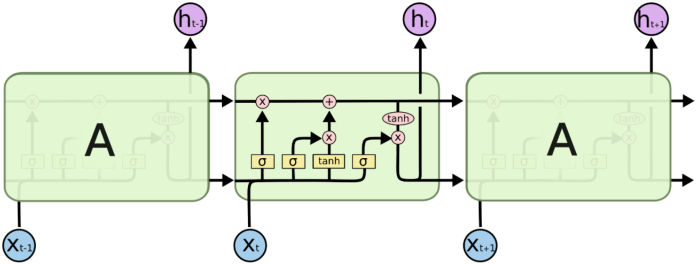
$$\left\| \frac{\partial h_j}{\partial h_{j-1}} \right\| \leq \beta_W \beta_s$$

$$\left\| \frac{\partial h_t}{\partial h_k} \right\| = \left\| \prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \right\| = \left\| \prod_{j=k+1}^{t} W \times \text{diag}[f'(h_{j-1})] \right\| \leq (\beta_W \beta_s)^{t-k}$$

Possible solution:

- Let $f(x) = \max(0, x)$, i.e., another activation function, for example, ReLU helps with gradient.
- Initialise W to be the identity matrix.

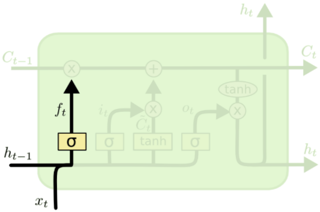
Looking at very complicated structure. But it works!



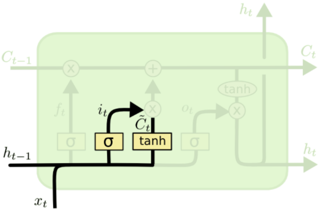
- There is a concept of Cell State $\{C_t\}$ in addition to state $\{h_t\}$.
- http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM): forget and input gate

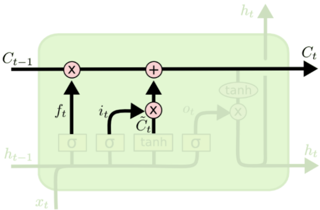
forget gate:
$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$



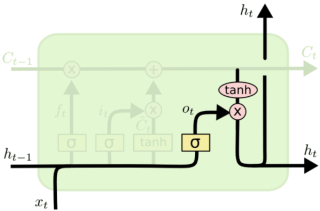
input gate:
$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$
$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C)$



state update:
$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t$



output gate:
$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$
$h_t = o_t \times \tanh(C_t)$



Adversarial Training

- a prior on input noise variables $z \sim p_z(z)$,
- G is differentiable function with parameters θ_g it transforms $z \rightarrow x$ 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:

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

- 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 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 $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_z(z)$;
    Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ 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 $\{z^{(1)}, \ldots, z^{(m)}\}$ 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:

$$p_{\mathbf{X}}(\theta) = \prod_{i=1}^{N} \frac{1}{Z(\theta)} f_{x_i}(\theta) = \frac{1}{Z(\theta)^n} \prod_{i=1}^{N} f_{x_i}(\theta) \qquad \text{where } Z(\theta) = \int_x f_\theta(x)\mathrm{d}x$$

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

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

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

Contrast Divergence (1)

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

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

Contrast Divergence (2)

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

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

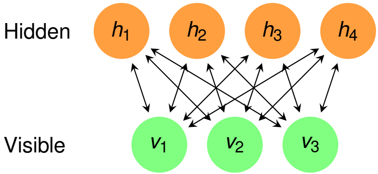
$$= \underbrace{\int_x \frac{\partial \log(f_x(\theta))}{\partial \theta} p_x(\theta) \mathrm{d}x}_{\text{population mean of } \left\{ \frac{\partial \log(f_x(\theta))}{\partial \theta} \right\}} - \underbrace{\frac{1}{N} \sum_{i=1}^{N} \frac{\partial \log(f_{x_i}(\theta))}{\partial \theta}}_{\text{sample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

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

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

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

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

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

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

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

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

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

note that the two Z are equal

The **derivative** of RBM Likelihood

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

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

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

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**:

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

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:

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

- 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:

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