Notes on RBM

Shi Feng

September 23, 2022

1 RBM[1]

A RBM is a bipartite binary probabilistic graphical model corresponding to the following distribution,

$$p(v,h) = \frac{1}{Z}\exp[-E(v,h)] \tag{1}$$

which assigns a probability to every possible pair of a visible (v) and a hidden vector (h) via this energy function energy function:

$$E(v,h) = -\sum_{i\in\text{visible}} a_i v_i - \sum_{j\in\text{hidden}} b_j h_j - \sum_{i,j} w_{ij} v_i h_j \tag{2}$$

The probability of v or h is given by a marginalization:

$$p(v) = \frac{1}{Z}\sum_h \exp[-E(v,h)], \quad p(h) = \frac{1}{Z}\sum_v \exp[-E(v,h)] \tag{3}$$

The derivation of the log probability w.r.t. w_{ij} is:

$$\begin{aligned}
\frac{\partial \log p(v)}{\partial w_{ij}} &= \frac{1}{p(v)}\left(-\frac{1}{Z^2}\frac{\partial Z}{\partial w_{ij}}\right)\sum_h e^{-E(v,h)} + \frac{1}{p(v)}\sum_h v_i h_j \frac{e^{-E(v,h)}}{Z} \\
&= -\frac{1}{p(v)}\left(\sum_{h,v} v_i h_j \frac{e^{-E(v,h)}}{Z}\right)\left(\sum_h \frac{e^{-E(v,h)}}{Z}\right) + \sum_h v_i h_j \frac{p(v,h)}{p(v)} \\
&= -\sum_{h,v} v_i h_j p(v,h) + \sum_h v_i h_j p(h|v) \\
&= -\mathbb{E}_{\text{model}}[v_i h_j] + \mathbb{E}_{\text{data}}[v_i h_j]
\end{aligned} \tag{4}$$

this leads to the gradient ascent learning rule of w_{ij}:

$$\delta w_{ij} = \beta(\mathbb{E}_{\text{data}}[v_i h_j] - \mathbb{E}_{\text{model}}[v_i h_j]) \tag{5}$$

and by the same token we can derive the updating process for a_i and b_j:

$$\begin{aligned}
\delta a_i &= \beta(\mathbb{E}_{\text{data}}[v_i] - \mathbb{E}_{\text{model}}[v_i]) \\
\delta b_j &= \beta(\mathbb{E}_{\text{data}}[h_j] - \mathbb{E}_{\text{model}}[h_j])
\end{aligned} \tag{6}$$

where β is the learning rate.

Now we need to figure out how to calculate the relevant expectation values mentioned above. We start with the conditional expectation $\mathbb{E}_{\text{data}}[v_i h_j]$. The key is to sample the probability $p(h|v)$. We can easily write down the conditional probability:

$$p(h|v) = \frac{p(h,v)}{p(v)} = \frac{\frac{1}{Z}e^{-E(v,h)}}{\frac{1}{Z}\sum_h e^{-E(v,h)}} = \frac{e^{-E(v,h)}}{\sum_h e^{-E(v,h)}} \tag{7}$$

and conditional probability for a single hidden node h_j can be derived by marginalization:

$$p(h_j|v) = \sum_{\{h_k\}-h_j} p(\{h_k\}|v) = \frac{\sum_{\{h_k\}-h_j} e^{-E(v,h)}}{\sum_h e^{-E(v,h)}} \tag{8}$$

For convenience we rewrite the energy function in the following form which separates the hidden and the visible nodes:

$$E(v,h) = -\sum_{j\in\text{hidden}}\left[h_j\left(b_j + \sum_{i\in\text{visible}} w_{ij}v_i\right)\right] - \sum_{i\in\text{visible}} a_i v_i \equiv -\sum_j \gamma_j(v)h_j - \sum_i a_i v_i \tag{9}$$

so the the Boltzmann factor in the numerator now takes the form:

$$\exp[-E(v,h)] = \prod_i e^{-a_i v_i} \prod_j e^{-\gamma_j(v)h_j} \tag{10}$$

Therefore the denomiator in Eq.8 can be written as

$$\sum_h e^{-E(v,h)} = \prod_i e^{-a_i v_i} \sum_h \prod_k e^{-\gamma_k(v)h_k} = \left[\prod_i e^{-a_i v_i}\right]\left[\sum_{h_j=\{0,1\}} e^{-\gamma_j(v)h_j}\right]\left[\sum_{\{h_k\}-h_j}\prod_{k\neq j} e^{-\gamma_k(v)h_k}\right]$$

and the numerator:

$$\sum_{\{h_k\}-h_j} e^{-E(v,h)} = e^{-\gamma_j(v)h_j}\left[\prod_i e^{-a_i v_i}\right]\left[\sum_{\{h_k\}-h_j}\prod_{k\neq j} e^{-\gamma_k(v)h_k}\right]$$

hence Eq.8 becomes a Logistic form:

$$p(h_j|v) = \frac{e^{-\gamma_j(v)h_j}}{1+e^{-\gamma_j(v)}} \tag{11}$$

Since each element in h_j is binary, we can readily write down the conditional probability for $h_j = 1, 0$ conditioned on v:

$$p(h_j = 1|v) = \frac{\exp(-b_j - \sum_i w_{ij}v_i)}{1+\exp(-b_j - \sum_i w_{ij}v_i)} = \sigma\left(b_j + \sum_i w_{ij}v_i\right) \tag{12}$$

$$p(h_j = 0|v) = 1 - p(h_j = 1|v) = \frac{1}{1+\exp(-b_j - \sum_i w_{ij}v_i)} \tag{13}$$

By the same token we can show $p(v_i|h)$ is also a similar sigmoid function:

$$p(v_i = 1|h) = \sigma\left(a_i + \sum_j w_{ij}h_j\right) \tag{14}$$

Algorithm 1 Sampling $\mathbb{E}_{\text{data}}[v_i h_j]$

Input: Data batch $(v_1, \cdots, v_N)$ and initial parameters of RBM
Output: $\mathbb{E}_{\text{data}}[v_i h_j]$
1. Initialize the $\mathbf{M} = 0$ matrix
2. For each v_t in data batch:
    Sample $h \sim p(h|v_t) = \sigma(\mathbf{b} + \mathbf{w}^\top v)$
    $\mathbf{M} \leftarrow \mathbf{M} + v_t h^\top$
3. $\mathbb{E}_{\text{data}}[vh^\top] \leftarrow \mathbf{M}/N$

Algorithm 2 Sampling $\mathbb{E}_{\text{model}}[v_i h_j]$

Input: Initial parameters of RBM
Output: $\mathbb{E}_{\text{model}}[v_i h_j]$
1. Initialize the $\mathbf{M} = 0$ matrix
2. Initialize v to be a random vector
3. Repeat N_c times (until convergence):
    Sample $h \sim p(h|v) = \sigma(\mathbf{b} + \mathbf{w}^\top v)$
    Sample $v \sim p(v|h) = \sigma(\mathbf{a} + \mathbf{w}h)$
    $\mathbf{M} \leftarrow \mathbf{M} + vh^\top$
3. $\mathbb{E}_{\text{model}}[vh^\top] \leftarrow \mathbf{M}/N_c$

We are now prepared to sample calculate $\mathbb{E}_{\text{data}}[v_i h_j] = \sum_h v_i h_j p(h|v)$ for every pair of i and j.

Next we need to compute $\mathbb{E}_{\text{model}}[v_i h_j] = \sum_{v,h} v_i h_j$, which is significantly harder since we are drawing correlated samples. Nevertheless, note that elements in v or h are not correlated within the same layer, so, assuming convergence is achievable, we can write down a similar algorithm sampling the hidden and visible layer one after another:

However, this scheme usually converges very slowly since samples of h and v are correlated. This is exactly where the contrastive divergence (CD) has a part to play. This can simply be done by setting $N_c = n$ for CD_n, where n is common chosen to be $n = 1$.

References

[1] Hinton, G. E. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, 599–619 (Springer, 2012).