第一章　受限波尔兹曼机(RBM,Restricted Boltzmann Machines)浅介

本文是我在阅读[1]之后做的一个读书笔记，所以这里的内容几乎也是翻译外加一些自己的理解，希望对读者有益。

概括地说，RBM根据MLE原理来估计预定义分布中的参数，以便预定义分布能尽可能地逼近产生观测数据的未知分布。由多个RBM分层堆叠而成的DBN(deep belief networks)构成深度学习的主要框架。

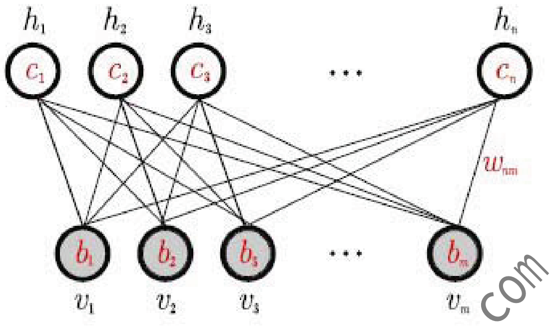

图 1.1: RBM示意图

RBM是一个随机无向图模型(图.一)，分为可见层$(V_1, V_2, \ldots, V_m)$和隐层$(H_1, H_2, \ldots, H_n)$。可见层中单元的个数与观测数据的维数相同，用于输入观测数据，隐层则用于刻画观测数据每个维度之间的依赖关系。每个可见层单元和所有隐层单元都有一个偏置参数b_j, c_i，并且每个隐层单元H_i和所有的可见层单元都有一个权重为$w_{ij}, j = 1, 2, \cdots, m$ 的无向连接边，这些都是RBM需要学习的参数，记为$\boldsymbol{\theta}$。另外，每个可见层单元和每个隐层单元的取值是一个0-1随机变量.RBM可以用于提取特征，每当向可见层输入一个观测值时，每个隐层单元取值为1和取值为0的概率便可以确定，此时我们可以将每个隐层单元的期望作为特征输出。

RBM假设观测数据的概率分布为:

$$p(\mathbf{v}^{(0)}) = \sum_{\mathbf{h}} p(\mathbf{v}^{(0)}, \mathbf{h}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v}^{(0)},\mathbf{h})}}{\sum_{\mathbf{v}} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}} \tag{1.1}$$

其中$p(\mathbf{v}, \mathbf{h})$是可见层和隐层的联合概率分布，$E(\mathbf{v}, \mathbf{h})$称作能量函数，由可见层取值$\mathbf{v}$，隐层取值$\mathbf{h}$以及RBM的参数确定。可以看出在所有参数确定以后，任给一个观测值$\mathbf{v}^{(0)}$，RBM都可以计算出一个对应的估计概率值。为了使得估计概率值能很接近真实的概率值，RBM需要从观测数据中学习适当的参数。RBM 采用的MLE 来学习参数，先假设只要一个观测数据$\mathbf{v}^{(0)}$，那么相应的似

然函数为，

$$ln\mathcal{L}(\boldsymbol{\theta}|\mathbf{v}^{(\mathbf{0})}) = lnp(\mathbf{v}^{(\mathbf{0})}|\boldsymbol{\theta}) = ln\sum_{\mathbf{h}} e^{-E(\mathbf{v}^{(\mathbf{0})},\mathbf{h})} - ln\sum_{\mathbf{v}}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{1.2}$$

为了求得参数的最大似然估计，我们对似然函数进行求导，

$$\begin{aligned}
\frac{\partial ln\mathcal{L}(\boldsymbol{\theta}|\mathbf{v}^{(\mathbf{0})})}{\partial\boldsymbol{\theta}} &= \frac{\partial}{\partial\boldsymbol{\theta}}\left(ln\sum_{\mathbf{h}} e^{-E(\mathbf{v}^{(\mathbf{0})},\mathbf{h})}\right) - \frac{\partial}{\partial\boldsymbol{\theta}}\left(ln\sum_{\mathbf{v}}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}\right) \\
&= -\sum_{\mathbf{h}} \frac{e^{-E(\mathbf{v}^{(\mathbf{0})},\mathbf{h})}}{\sum_{\mathbf{h}} e^{-E(\mathbf{v}^{(\mathbf{0})},\mathbf{h})}} \frac{\partial E(\mathbf{v}^{(\mathbf{0})},\mathbf{h})}{\partial\boldsymbol{\theta}} + \sum_{\mathbf{v}}\sum_{\mathbf{h}} \frac{e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{v}}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}} \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\boldsymbol{\theta}} \\
&= -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}^{(\mathbf{0})}) \frac{\partial E(\mathbf{v}^{(\mathbf{0})},\mathbf{h})}{\partial\boldsymbol{\theta}} + \sum_{\mathbf{v}}\sum_{\mathbf{h}} p(\mathbf{v},\mathbf{h}) \frac{\partial E(\mathbf{v},\mathbf{h})}{\partial\boldsymbol{\theta}}
\end{aligned} \tag{1.3}$$

注意

$$p(\mathbf{h}|\mathbf{v}) = \frac{p(\mathbf{v},\mathbf{h})}{p(\mathbf{v})} = \frac{\frac{e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{v}}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}}{\frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{v}}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}} = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}} \tag{1.4}$$

如果，给定一个训练样本可以快速计算得到似然函数的梯度，那么我们就可以采用梯度上升算法学习参数，然而似然函数的梯度直接计算起来却非常耗时，因为似然函数梯度的第二项是对$\mathbf{v},\mathbf{h}$所有可能的取值全部进行累加。下面我们就来探讨如何快速准确地计算似然函数的梯度。

标准的RBM中的能量函数为：

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_iv_j - \sum_{j=1}^{m} b_jv_j - \sum_{i=1}^{n} c_ih_i \tag{1.5}$$

另外，RBM有一个重要的条件概率独立假设，这个假设可以大大简化模型的计算，注意这是一个预定义的模型假设，而非由其他模型中的定义经过数学推导得到的。

$$P(\mathbf{v}|\mathbf{h}) = \prod_{j=1}^{m} P(v_j|\mathbf{h}), P(\mathbf{h}|\mathbf{v}) = \prod_{i=1}^{n} P(h_i|\mathbf{v}) \tag{1.6}$$

通过已有的模型相关定义和假设，我们可以证明如下两个结论，

$$P(v_j = 1|\mathbf{h}) = \sigma\left(\sum_{i=1}^{n} w_{ij}h_i + b_j\right), P(h_i = 1|\mathbf{v}) = \sigma\left(\sum_{j=1}^{m} w_{ij}v_j + c_i\right) \tag{1.7}$$

其中$\sigma(x)=\frac{1}{1+e^{-x}}$，上式的具体证明可以参考[1]中的第27式。给定能量函数和条件概率独立假设，并令$\mathbf{h}_{-i}$表示多维随机变量$\mathbf{h}$去掉第i维分量后形成的随机变量，那么

$$
\begin{aligned}
&-\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}^{(\mathbf{0})})\frac{\partial E(\mathbf{v}^{(\mathbf{0})},\mathbf{h})}{\partial w_{ij}} = \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}^{(\mathbf{0})})h_i v_j^{(0)} \\
&= \sum_{h_i}\sum_{\mathbf{h}_{-i}} P(h_i|\mathbf{v}^{(\mathbf{0})})P(\mathbf{h}_{-i}|\mathbf{v}^{(\mathbf{0})})h_i v_j^{(0)} \qquad \text{(加法结合律，条件概率独立)} \\
&= \sum_{h_i} P(h_i|\mathbf{v}^{(\mathbf{0})})h_i v_j^{(0)} \sum_{\mathbf{h}_{-i}} P(\mathbf{h}_{-i}|\mathbf{v}^{(\mathbf{0})}) \qquad \text{(加法交换律)} \\
&= \sum_{h_i} P(h_i|\mathbf{v}^{(\mathbf{0})})h_i v_j^{(0)} \qquad \text{(任何概率分布在整个取值空间中的累加和为1)} \\
&= P(h_i=1|\mathbf{v}^{(\mathbf{0})})v_j^{(0)} \qquad (\text{代入}h_i=1,h_i=0)
\end{aligned}
\tag{1.8}
$$

如此一来，我们就可以很容易地计算似然函数梯度的第一项，而第二项我们可以重写为：

$$\sum_{\mathbf{v}}\sum_{\mathbf{h}} p(\mathbf{v},\mathbf{h})\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial w_{ij}} = \sum_{\mathbf{v}} p(\mathbf{v})\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v})h_i v_j$$

我们可以通过采样从分布$p(\mathbf{v},\mathbf{h})$获得一些样本值$(\mathbf{v}^{(s)},\mathbf{h})^{(s)}, s=1,2,\cdots$，然后代入进去近似地计算第二项，如果采样方法恰当，那么这样的近似也将很好。问题是分布$p(\mathbf{v},\mathbf{h})$的计算很费时，不过好在我们知道$P(\mathbf{v}|\mathbf{h}), P(\mathbf{h}|\mathbf{v})$,而且这两个分布也容易计算，那么那么我们可以通过Gibbs采样获得样本。

Algorithm 1 $p(\mathbf{v},\mathbf{h})$的Gibbs采样算法

```
输入一个观测值v^(0)。
for t = 0, 1, 2, ··· do
    采样，h^{t+1} ~ P(h|v^t)
    采样，v^{t+1} ~ P(v|h^{t+1})
end for
```

相对于其他参数的梯度可以类似地计算。然而，现在的问题是Gibbs采样算法的原理是基于Markov链的，往往Markov链需要做很多次转移(也即是Gibbs采样算法中的t要变得很大)才能到达稳态分布，而只有到达稳态分布才

能得到真正来自$p(\mathbf{v},\mathbf{h})$的采样值。另外，不容易确定究竟什么时候才能到达稳态分布。所以提出了对比散度(CD，Contrastive Divergence) 算法．CD算法只进行k步采样，便将得到的$\mathbf{v}^k$带入计算，也就是似然函数的梯度按如下方式计算，

$$\mathrm{CD}_k(\theta, \boldsymbol{v}^{(0)}) = -\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^{(0)}) \frac{\partial E(\boldsymbol{v}^{(0)}, \boldsymbol{h})}{\partial \theta} + \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^{(k)}) \frac{\partial E(\boldsymbol{v}^{(k)}, \boldsymbol{h})}{\partial \theta}.$$

CD算法的详细步骤如下，实验中一般$k=1$。

Algorithm 1. k-step contrastive divergence

Input: RBM $(V_1,\dots,V_m,H_1,\dots,H_n)$, training batch S
Output: gradient approximation Δw_{ij}, Δb_j and Δc_i for $i=1,\dots,n$, $j=1,\dots,m$

init $\Delta w_{ij}=\Delta b_j=\Delta c_i=0$ for $i=1,\dots,n$, $j=1,\dots,m$
forall the $v\in S$ **do**
  $v^{(0)} \leftarrow v$
  for $t=0,\dots,k-1$ **do**
    for $i=1,\dots,n$ **do** sample $h_i^{(t)} \sim p(h_i \mid v^{(t)})$
    for $j=1,\dots,m$ **do** sample $v_j^{(t+1)} \sim p(v_j \mid h^{(t)})$
  for $i=1,\dots,n$, $j=1,\dots,m$ **do**
    $\Delta w_{ij} \leftarrow \Delta w_{ij} + p(H_i=1 \mid v^{(0)})\cdot v_j^{(0)} - p(H_i=1 \mid v^{(k)})\cdot v_j^{(k)}$
    $\Delta b_j \leftarrow \Delta b_j + v_j^{(0)} - v_j^{(k)}$
    $\Delta c_i \leftarrow \Delta c_i + p(H_i=1 \mid v^{(0)}) - p(H_i=1 \mid v^{(k)})$

我们现在想知道为什么CD算法可行。其实有下面两个定理保证了CD算法合理性。这两个定理的意思就是，CD算法中对似然函数梯度的近似误差的期望随着会随着k的增大会快速趋近于0.

Theorem 1 **(Bengio and Delalleau [3]).** *For a converging Gibbs chain*

$$\boldsymbol{v}^{(0)} \Rightarrow \boldsymbol{h}^{(0)} \Rightarrow \boldsymbol{v}^{(1)} \Rightarrow \boldsymbol{h}^{(1)} \ldots$$

starting at data point $v^{(0)}$, *the log-likelihood gradient can be written as*

$$\frac{\partial}{\partial\theta}\ln p(\boldsymbol{v}^{(0)}) = -\sum_{\boldsymbol{h}} p(h|\boldsymbol{v}^{(0)})\frac{\partial E(\boldsymbol{v}^{(0)},\boldsymbol{h})}{\partial\theta}$$
$$+ E_{p(\boldsymbol{v}^{(k)}|\boldsymbol{v}^{(0)})}\left[\sum_{\boldsymbol{h}} p(h|\boldsymbol{v}^{(k)})\frac{\partial E(\boldsymbol{v}^{(k)},\boldsymbol{h})}{\partial\theta}\right] + E_{p(\boldsymbol{v}^{(k)}|\boldsymbol{v}^{(0)})}\left[\frac{\partial \ln p(\boldsymbol{v}^{(k)})}{\partial\theta}\right]$$

and the final term converges to zero as k goes to infinity.

Theorem 2 **(Fischer and Igel [12]).** *Let p denote the marginal distribution of the visible units of an RBM and let q be the empirical distribution defined by a set of samples* $v_1, \ldots, v_\ell$. *Then an upper bound on the expectation of the error of the CD-k approximation of the log-likelihood derivative w.r.t some RBM parameter* θ_a *is given by*

$$\left| E_{q(\boldsymbol{v}^{(0)})}\left[E_{p(\boldsymbol{v}^{(k)}|\boldsymbol{v}^{(0)})}\left[\frac{\partial \ln p(\boldsymbol{v}^{(k)})}{\partial\theta_a}\right]\right]\right| \leq \frac{1}{2}\|q-p\|\left(1-e^{-(m+n)\Delta}\right)^k \tag{37}$$

with

$$\Delta = \max\left\{\max_{l\in\{1,\ldots,m\}}\vartheta_l, \max_{l\in\{1,\ldots,n\}}\xi_l\right\} \ ,$$

where

$$\vartheta_l = \max\left\{\left|\sum_{i=1}^{n} I_{\{w_{il}>0\}}w_{il}+b_l\right|, \left|\sum_{i=1}^{n} I_{\{w_{il}<0\}}w_{il}+b_l\right|\right\}$$

and

$$\xi_l = \max\left\{\left|\sum_{j=1}^{m} I_{\{w_{lj}>0\}}w_{lj}+c_l\right|, \left|\sum_{j=1}^{m} I_{\{w_{lj}<0\}}w_{lj}+c_l\right|\right\} \ .$$

参考文献

[1] Asja Fischer and Christian Igel. An introduction to restricted boltzmann machines. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 7441, pages 14–36. Springer Berlin Heidelberg, 2012.