深度学习读书笔记

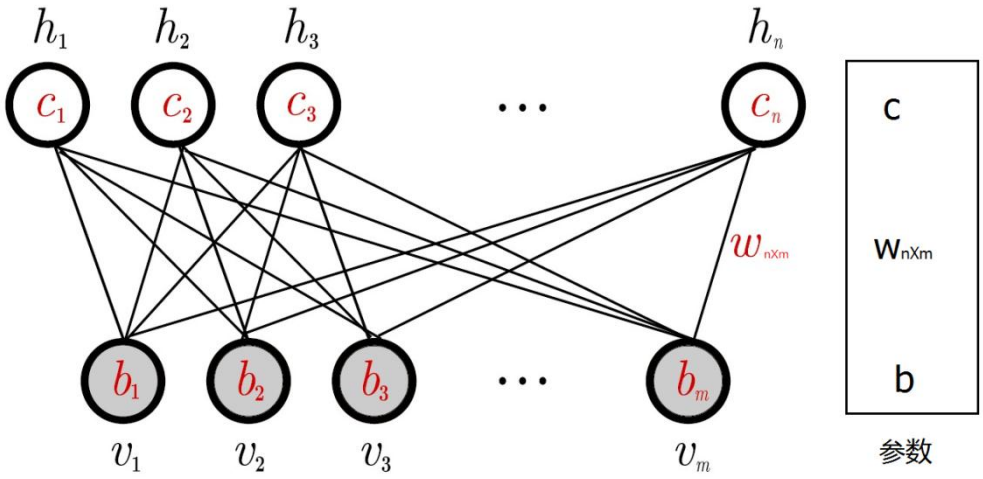
by 北流浪子
博客地址：http://blog.csdn.net/mytestmy/article/details/9150213

三．限制波尔兹曼机

3.1 限制波尔兹曼机（RBM）使用方法

3.1.1 RBM 的使用说明

一个普通的 RBM 网络结构如下。



以上的 RBM 网络结构有 m 个可视节点和 n 个隐藏节点，其中每个可视节点只和 n 个隐藏节点相关，和其他可视节点是独立的，就是这个可视节点的状态只受 n 个隐藏节点的影响，对于每个隐藏节点也是，只受 m 个可视节点的影响，这个特点使得 RBM 的训练变得容易了。

RBM 网络有几个参数，一个是可视层与隐藏层之间的权重矩阵$W_{n\times m}$，一个是可视节点的偏移量$\mathrm{b} = (\mathrm{b1}, \mathrm{b2} \cdots \mathrm{bm})$，一个是隐藏节点的偏移量$\mathrm{c} = (\mathrm{c1}, \mathrm{c2} \cdots \mathrm{cn})$，这几个参数决定了 RBM 网络将一个 m 维的样本编码成一个什么样的 n 维的样本。

RBM 网络的功能有下面的几种，就简单地先描述一下。

首先为了描述容易，先假设每个节点取值都在集合{0,1}中，即$\forall \mathrm{i}, \mathrm{j}, v_j \in \{0,1\}, h_i \in \{0,1\}$。

一个训练样本x过来了取值为$\mathrm{x} = (\mathrm{x1}, \mathrm{x2} \cdots \mathrm{xm})$，根据 RBM 网络，可以得到这个样本的 m 维的编码后的样本$\mathrm{y} = (\mathrm{y1}, \mathrm{y2} \cdots \mathrm{yn})$，这 n 维的编码也可以认为是抽取了 n 个特征的样本。而这个 m 维的编码后的样本是按照下面的规则生成的：对于给定的$\mathrm{x} = (\mathrm{x1}, \mathrm{x2} \cdots \mathrm{xm})$，隐藏层的第 i 个节点的取值为 1（编码后的样本的第 i 个特征的取值为 1）的概率为$\mathrm{p}(\mathrm{hi} = 1|\mathrm{v}) = \sigma(\sum_{j=1}^{m} wij \times \mathrm{vj} + \mathrm{ci})$，其中的 v 取值就是 x，hi 的取值就是 yi，其中$\sigma(\mathrm{x}) = 1/(1 + e^{-x})$，是 sigmoid 函数。也就是说，编码后的样本 y 的第 i 个位置的取值为 1 的概率是$\mathrm{p}(\mathrm{hi} = 1|\mathrm{v})$。所

以，生成 yi 的过程就是：

i）先利用公式$p(hi = 1|v) = \sigma(\sum_{j=1}^{m} wij \times vj + ci)$，根据 x 的值计算概率$p(hi = 1|v)$，其中 vj 的取值就是 xj 的值。

ii）然后产生一个 0 到 1 之间的随机数，如果它小于$p(hi = 1|v)$，yi 的取值就是 1，否则就是 0（假如 p(hi=1|v)=0.6，这里就是因为 yi 的取值就是 1 的概率是,0.6，而这个随机数小于 0.6 的概率也是 0.6；如果这个随机数小于 0.6，就是这个事件发生了，那就可以认为 yi 的取值是 1 这个事件发生了，所以把 yi 取值为 1）。

反过来，现在知道了一个编码后的样本 y，想要知道原来的样本 x，即解码过程，跟上面也是同理，过程如下：

i）先利用公式$p(vj = 1|h) = \sigma(\sum_{i=1}^{n} wij \times hi + bj)$，根据 y 的值计算概率$p(vj = 1|h)$，其中 hi 的取值就是 yi 的值。

ii）然后产生一个 0 到 1 之间的随机数，如果它小于$p(vj = 1|h)$，vj 的取值就是 1，否则就是 0。

对于（ii）的说明：不说别的——比如吧，你现在出去逛街，走到一个岔路口，你只想随便逛逛，所以你是有 0.5 的概率往左边的路，0.5 的概率往右边的路；但是你不知道怎么选择哪个路，所以你选择了抛硬币，正面朝上你就向左，反面朝上就向右。现在你只抛一次，发现他是正面朝上的，你就向左走了。

——回到上面的问题，某节点 A 取值为 1 的概率是 0.6（假如），也可以看做一个找不均匀的硬币，正面朝上的概率是 0.6，反面朝上的概率是 0.4；现在要给节点 A 取值，就拿这个硬币抛一下，正面朝上就取值 1，反面朝上就取值 0，这个就相当于抛硬币决定走哪个路的那个过程。

——现在假如找不到这样的不均匀的硬币，就拿随机数生成器来代替（生成的数是 0-1 之间的浮点数）；因为随机数生成器取值小于 0.6 的概率也是 0.6，大于 0.6 的概率是 0.4。

3.1.2 RBM 的用途

RBM 的用途主要是两种，一是对数据进行编码，然后交给监督学习方法去进行分类或回归，二是得到了权重矩阵和偏移量，供 BP 神经网络初始化训练。

第一种可以说是把它当做一个降维的方法来使用。

第二种就用途比较奇怪。其中的原因就是神经网络也是要训练一个权重矩阵和偏移量，但是如果直接用 BP 神经网络，初始值选得不好的话，往往会陷入局部极小值。根据实际应用结果表明，直接把 RBM 训练得到的权重矩阵和偏移量作为 BP 神经网络初始值，得到的结果会非常地好。

这就类似爬山，如果一个风景点里面有很多个山峰，如果让你随便选个山就爬，希望你能爬上最高那个山的山顶，但是你的精力是有限的，只能爬一座山，而你也不知道哪座山最高，这样，你就很容易爬到一座不是最高的山上。但是，如果用直升机把你送到最高的那个山上的靠近山顶处，那你就能很容易地爬上最高的那座山。这个时候，RBM 就的角色就是那个直升机。

其实还有两种用途的，下面说说。

第三种，RBM 可以估计联合概率 p(v,h)，如果把 v 当做训练样本，h 当成类别标签（隐藏节点只有一个的情况，能得到一个隐藏节点取值为 1 的概率），就可以利用利用贝叶斯公式求 p(h|v)，然后就可以进行分类，类似朴素贝叶斯、LDA、HMM。说得专业点，RBM 可以作为一个生成模型（Generative model）使用。

第四种，RBM 可以直接计算条件概率 p(h|v)，如果把 v 当做训练样本，h 当成类别标签（隐藏节点只有一个的情况，能得到一个隐藏节点取值为 1 的概率），RBM 就可以用来进行分类。说得专业点，RBM 可以作为一个判别模型（Discriminative model）使用。

3.2 限制波尔兹曼机（RBM）能量模型

3.2.1 能量模型定义

在说 RBM 之前，先来说点其他的，就是能量模型。

能量模型是个什么样的东西呢？直观上的理解就是，把一个表面粗糙又不太圆的小球，放到一个表面也比较粗糙的碗里，就随便往里面一扔，看看小球停在碗的哪个地方。一般来说停在碗底的可能性比较大，停在靠近碗底的其他地方也可能，甚至运气好还会停在碗口附近（这个碗是比较浅的一个碗）；能量模型把小球停在哪个地方定义为一种状态，每种状态都对应着一个能量，这个能量由能量函数来定义，小球处在某种状态的概率（如停在碗底的概率跟停在碗口的概率当然不一样）可以通过这种状态下小球具有的能量来定义（换个说法，如小球停在了碗口附近，这是一种状态，这个状态对应着一个能量 E，而发生“小球停在碗口附近”这种状态的概率 p，可以用 E 来表示，表示成 p=f(E)，其中 f 是能量函数），这就是我认为的能量模型。

这样，就有了能量函数，概率之类的东西。

波尔兹曼网络是一种随机网络。描述一个随机网络，总结起来主要有两点。

第一，概率分布函数。由于网络节点的取值状态是随机的，从贝叶斯网的观点来看，要描述整个网络，需要用三种概率分布来描述系统。即联合概率分布，边缘概率分布和条件概率分布。要搞清楚这三种不同的概率分布,是理解随机网络的关键,这里向大家推荐的书籍是张连文所著的《贝叶斯网引论》。很多文献上说受限波尔兹曼是一个无向图，从贝叶斯网的观点看，受限波尔兹曼网络也可以看作一个双向的有向图，即从输入层节点可以计算隐层节点取某一种状态值的概率，反之亦然。

第二，能量函数。随机神经网络是根植于统计力学的。受统计力学中能量泛函的启发，引入了能量函数。能量函数是描述整个系统状态的一种测度。系统越有序或者概率分布越集中，系统的能量越小。反之，系统越无序或者概率分布越趋于均匀分布，则系统的能量越大。能量函数的最小值，对应于系统的最稳定状态。

3.2.2 能量模型作用

为什么要弄这个能量模型呢？原因有几个。

第一、RBM 网络是一种无监督学习的方法，无监督学习的目的是最大可能的拟合输入数据，所以学习 RBM 网络的目的是让 RBM 网络最大可能地拟合输入数据。

第二、对于一组输入数据来说，现在还不知道它符合那个分布，那是非常难学的。例如，知道它符合高斯分布，那就可以写出似然函数，然后求解，就能求出这个是一个什么样个高

斯分布;但是要是不知道它符合一个什么分布,那可是连似然函数都没法写的,问题都没有,根本就无从下手。好在天无绝人之路——统计力学的结论表明,任何概率分布都可以转变成基于能量的模型,而且很多的分布都可以利用能量模型的特有的性质和学习过程,有些甚至从能量模型中找到了通用的学习方法。有这样一个好东西,当然要用了。

第三、在马尔科夫随机场(MRF)中能量模型主要扮演着两个作用:一、全局解的度量(目标函数);二、能量最小时的解(各种变量对应的配置)为目标解。也就是能量模型能为无监督学习方法提供两个东西:a)目标函数;b)目标解。

换句话说,就是——使用能量模型使得学习一个数据的分布变得容易可行了。

能否把最优解嵌入到能量函数中至关重要,决定着我们具体问题求解的好坏。统计模式识别主要工作之一就是捕获变量之间的相关性,同样能量模型也要捕获变量之间的相关性,变量之间的相关程度决定了能量的高低。把变量的相关关系用图表示出来,并引入概率测度方式就构成了概率图模型的能量模型。

RBM 作为一种概率图模型,引入了概率就可以使用采样技术求解,在 CD(contrastive divergence)算法中采样部分扮演着模拟求解梯度的角色。

能量模型需要一个定义能量函数,RBM 的能量函数的定义如下

$$\mathrm{E}(\mathrm{v},\mathrm{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

这个能量函数的意思就是,每个可视节点和隐藏节点之间的连接结构都有一个能量,通俗来说就是可视节点的每一组取值和隐藏节点的每一组取值都有一个能量,如果可视节点的一组取值(也就是一个训练样本的值)为(1,0,1,0,1,0),隐藏节点的一组取值(也就是这个训练样本编码后的值)为(1,0,1),然后分别代入上面的公式,就能得到这个连接结构之间的能量。

能量函数的意义是有一个解释的,叫做专家乘积系统(POE,product of expert),这个理论也是 hinton 发明的,他把每个隐藏节点看做一个“专家”,每个“专家”都能对可视节点的状态分布产生影响,可能单个“专家”对可视节点的状态分布不够强,但是所有的“专家”的观察结果连乘起来就够强了。具体我也看不太懂,各位有兴趣看 hinton 的论文吧,中文的也有,叫《专家乘积系统的原理及应用,孙征,李宁》。

另外的一个问题是:为什么要搞概率呢?下面就是解释。

能量模型需要两个东西,一个是能量函数,另一个是概率,有了概率才能跟要求解的问题联合起来。

下面就介绍从能量模型到概率吧。

3.3 从能量模型到概率

3.3.1 从能量函数到概率

为了引入概率,需要定义概率分布。根据能量模型,有了能量函数,就可以定义一个可视节点和隐藏节点的联合概率

$$\mathrm{p}(\mathrm{v},\mathrm{h}) = \frac{e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}}$$

也就是一个可视节点的一组取值(一个状态)和一个隐藏节点的一组取值(一个状态)发生的概率 p(v,h)是由能量函数来定义的。

这个概率不是随便定义的，而是有统计热力学的解释的——在统计热力学上，当系统和它周围的环境处于热平衡时，一个基本的结果是状态 i 发生的概率如下面的公式

$$\mathrm{p}_i = \frac{1}{Z} \times e^{-\frac{E_i}{k_b \times T}}$$

其中E_i表示系统在状态 i 时的能量，T 为开尔文绝对温度，k_{B}为 Boltzmann 常数，Z 为与状态无关的常数。

我们这里的E_i变成了 E(v,h)，因为(v,h)也是一个状态，其他的参数 T 和k_{B}由于跟求解无关，就都设置为 1 了，Z 就是我们上面联合概率分布的分母，这个分母是为了让我们的概率的和为 1，这样才能保证 p(v,h)是一个概率。

现在我们得到了一个概率，其实也得到了一个分布，其实这个分布还有一个好听点的名字，可以叫做 Gibbs 分布，当然不是一个标准的 Gibbs 分布，而是一个特殊的 Gibbs 分布，这个分布是有一组参数的，就是能量函数的那几个参数 w，b，c。

有了这个联合概率，就可以得到一些条件概率，是用积分去掉一些不想要的量得到的。

$$P(\mathbf{v}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}},\ P(\mathbf{h}) = \frac{\sum_{\mathbf{v}} e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}$$

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

3.3.2 从概率到极大似然

上面得到了一个样本和其对应编码的联合概率，也就是得到了 RBM 网络的 Gibbs 分布的概率密度函数，引入能量模型的目的是为了方便求解的。

现在回到求解的目标——让 RBM 网络的表示 Gibbs 分布最大可能的拟合输入数据。

其实求解的目标也可以认为是让 RBM 网络表示的 Gibbs 分布与输入样本的分布尽可能地接近。

现在看看“最大可能的拟合输入数据”这怎么定义。

假设Ω表示样本空间，q 是输入样本的分布，即 q(x)表示训练样本 x 的概率， q 其实就是要拟合的那个样本表示分布的概率；再假设 p 是 RBM 网络表示的 Gibbs 分布的边缘分布（只跟可视节点有关，隐藏节点是通过积分去掉了，可以理解为可视节点的各个状态的分布），输入样本的集合是 S，那现在就可以定义样本表示的分布和 RBM 网络表示的边缘分布的 KL 距离

$$\mathrm{KL}(q||p) = \sum_{\boldsymbol{x}\in\Omega} q(\boldsymbol{x}) \ln \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})} = \sum_{\boldsymbol{x}\in\Omega} q(\boldsymbol{x}) \ln q(\boldsymbol{x}) - \sum_{\boldsymbol{x}\in\Omega} q(\boldsymbol{x}) \ln p(\boldsymbol{x})$$

如果输入样本表示的分布与 RBM 表示的 Gibbs 分布完全符合，这个 KL 距离就是 0，否则就是一个大于 0 的数。

第一项其实就是输入样本的熵（熵的定义），输入样本定了熵就定了；第二项没法直接求，但是如果用蒙特卡罗抽样（后面有章节会介绍）来估算这个值，但是，偷懒一下，直接把输入样本当作利用抽样过程抽中的样本（输入样本肯定符合分布 q(x)），第二项可以用

$\frac{1}{l}\sum_{x\in S}\ln p(x)$来估计，其中的 *l* 表示训练样本个数。由于 KL 的值肯定是不小于 0，所以第一项肯定不小于第二项，让第二项取得最大值，就能让 KL 距离最小；最后，还可以发现，最大化$\frac{1}{l}\sum_{x\in S}\ln p(x)$，相当于最大化$\sum_{x\in S}\ln p(x)$，而这就是极大似然估计。

结论就是求解输入样本的极大似然，就能让 RBM 网络表示的 Gibbs 分布和样本本身表示的分布最接近。

这就是为什么 RBM 问题最终可以转化为极大似然来求解。

既然要用极大似然来求解，这个当然是有意义的——当 RBM 网络训练完成后，如果让这个 RBM 网络随机发生若干次状态（当然一个状态是由(v,h)组成的），这若干次状态中，可视节点部分（就是 v）出现训练样本的概率要最大。

这样就保证了，在反编码（从隐藏节点到可视节点的编码过程）过程中，能使训练样本出现的概率最大，也就是使得反编码的误差尽最大的可能最小。

例如一个样本(1,0,1,0,1)编码到(0,1,1)，那么，(0,1,1)从隐藏节点反编码到可视节点的时候也要大概率地编码到(1,0,1,0,1)。

到这，能量模型到极大似然的关系说清楚了，，可能有点不太对，大家看到有什么不对的提一下，我把它改了。

之前想过用似然函数最大的时候，RBM 网络的能量最小这种思路去解释，结果看了很多的资料，都实在没有办法把那两个极值联系起来。这个冤枉路走了好久，希望大家在学习的时候多注意，尽量避免走这个路。

下面就说怎么求解了。求解的意思就是求 RBM 网络里面的几个参数 w，b，c 的值，这个就是解，而似然函数（对数似然函数）是目标函数。最优化问题里面的最优解和目标函数的关系大家务必先弄清楚。

3.4 求解极大似然

既然是求解极大似然，就要对似然函数求最大值，

求解的过程就是对参数就导，然后用梯度上升法不断地把目标函数提升，最终到达停机条件（在这里看不懂的同学就去看参考文献中的《从极大似然到 EM》）

设参数为θ（注意是参数是一组参数），则似然函数可以写为

$$\mathrm{L}(\theta)=\prod_{v}\mathrm{L}(\mathrm{v}|\theta)=\prod_{v}p(v)$$

为了省事$\mathrm{L}(\mathrm{v}|\theta)=p(v|\theta)$写成 p(v)了，然后就可以对它的对数进行求导了（因为直接求一个连乘的导数太复杂，所以变成对数来求）

$$\frac{\partial L(\theta)}{\partial\theta}=\sum_{v}\frac{\partial\ln\mathrm{L}(\mathrm{v}|\theta)}{\partial\theta}=\sum_{v}\frac{\partial\ln p(v)}{\partial\theta}$$

可以看到，是对每个 p(v)求导后再加和，然后就有了下面的推导了。

$$\ln P(\mathbf{v}) = \ln\left(\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}\right) - \ln\left(\sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}\right)$$

$$\frac{\partial \log P(\mathbf{v})}{\partial \boldsymbol{\theta}} = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}\left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}}\right)}{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}} - \frac{\sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}\left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}}\right)}{\sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}$$

$$= \sum_{\mathbf{h}}\left(P(\mathbf{h}|\mathbf{v})\left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}}\right)\right) - \sum_{\mathbf{v},\mathbf{h}}\left(P(\mathbf{v},\mathbf{h})\left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}}\right)\right)$$

$$= \mathbb{E}_{P(\mathbf{h}|\mathbf{v})}\left[-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}}\right] - \mathbb{E}_{P(\mathbf{h},\mathbf{v})}\left[-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \boldsymbol{\theta}}\right]$$

到了这一步的时候，我们又可以发现问题了。我们可以看到的是，对每一个样本，第二个等号后面的两项其实都像是在求一个期望，第一项求的是$-\frac{\partial E(v,h)}{\partial \theta}$这个函数在概率 p(h|v)下的期望，第二项求的是$-\frac{\partial E(v,h)}{\partial \theta}$这个函数在概率 p(v,h)下面的期望。

还有论文是这么解释的，上面的公式，对 w 求偏导，还可以再进一步转化

$$\frac{1}{\ell}\sum_{\boldsymbol{v}\in S}\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta}\mid \boldsymbol{v})}{\partial w_{ij}} = \frac{1}{\ell}\sum_{\boldsymbol{v}\in S}\left[-\mathbb{E}_{p(\boldsymbol{h}\mid \boldsymbol{v})}\left[\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial w_{ij}}\right] + \mathbb{E}_{p(\boldsymbol{h},\boldsymbol{v})}\left[\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial w_{ij}}\right]\right]$$

$$= \frac{1}{\ell}\sum_{\boldsymbol{v}\in S}\left[\mathbb{E}_{p(\boldsymbol{h}\mid \boldsymbol{v})}\left[v_i h_j\right] - \mathbb{E}_{p(\boldsymbol{h},\boldsymbol{v})}\left[v_i h_j\right]\right]$$

$$= \left\langle v_i h_j\right\rangle_{p(\boldsymbol{h}\mid \boldsymbol{v})q(\boldsymbol{v})} - \left\langle v_i h_j\right\rangle_{p(\boldsymbol{h},\boldsymbol{v})}$$

注意：从第二个“=”号到第三个“=”号的时候，第二个“=”号的方括号里面的第一项是把蒙特卡罗抽样的用法反过来了，从抽样回到了积分，所以得到了一个期望；第二项就是因为对每个训练样本，第二项的值算出来都一样（是一个积分的结果，与被积变量无关，是一个标量），所以累加以后的结果除以 l，相当于没有变化，还是那个期望，只是表达的形式换了。

这样的话，这个梯度还可以这么理解，第一项等于输入样本数据的自由能量函数偏导在样本本身的那个分布下的期望值（q(v,h)=p(h|v)q(v)，q 表示输入样本和他们对应的隐藏增状态表示的分布），而第二项是自由能量函数的偏导在 RBM 网络表示的 Gibbs 分布下的期望值。

第一项是可以求的，因为训练样本有了，也就是使用蒙特卡罗估算期望的时候需要的样本已经抽好了，只要求个均值就可以了。

第二项也是可以求的，但是要对 v 和 h 的组合的所有可能的取值都遍历一趟，这就可能没法算了；想偷懒的话，悲剧就在于，现在是没有 RBM 网络表示的 Gibbs 分布下的样本的（当然后面会介绍这些怎么抽）。

为了进行下面的讨论，把这个梯度再进一步化简，看看能得到什么。根据能量函数的公式，是有三个参数的 w，b 和 c，就分别对这三个参数求导

$$\frac{\partial \ln p(v)}{\partial w_{ij}} = \sum_h p(h|v)(-\frac{\partial E(v,h)}{\partial w_{ij}}) - \sum_{v,h} p(v,h)\left(-\frac{\partial E(v,h)}{\partial w_{ij}}\right)$$

$$= \sum_h p(h|v)h_i v_j - \sum_{v,h} p(v,h)h_i v_j = \sum_h p(h|v)h_i v_j - \sum_v p(v)\sum_h p(h|v)h_i v_j$$

$$= p(h_i = 1|v)v_j - \sum_v p(v)p(h_i = 1|v)v_j$$

$$\frac{\partial \ln p(v)}{\partial b_j} = \sum_h p(h|v)(-\frac{\partial E(v,h)}{\partial b_j}) - \sum_{v,h} p(v,h)\left(-\frac{\partial E(v,h)}{\partial b_j}\right) = \sum_h p(h|v)v_j - \sum_{v,h} p(v,h)v_j$$

$$= \sum_h p(h|v)v_j - \sum_v p(v)\sum_h p(h|v)v_j = v_j - \sum_v p(v)v_j$$

$$\frac{\partial \ln p(v)}{\partial c_i} = \sum_h p(h|v)(-\frac{\partial E(v,h)}{\partial c_i}) - \sum_{v,h} p(v,h)\left(-\frac{\partial E(v,h)}{\partial c_i}\right) = \sum_h p(h|v)h_i - \sum_{v,h} p(v,h)h_i$$

$$= \sum_h p(h|v)h_i - \sum_v p(v)\sum_h p(h|v)h_i = p(h_i = 1|v) - \sum_v p(v)p(h_i = 1|v)$$

其中第四个等号的原因是$h_i \in \{0,1\}$，第三个等号的原因是$p(v,h) = p(h|v)p(v)$。

到了这一步，我们来分析一下，从上面的联合概率那一堆，我们可以得到下面的

$$p(vj = 1|h) = \frac{\sum_{v_{k\neq j}} p(vj = 1, v_{k\neq j}, h)}{\sum_v p(v,h)}$$

$$= \frac{\sum_{v_{k\neq j}} \exp(\sum_{i=1}^n w_{ij}hi + bj + \sum_{i=1}^n \sum_{k=1,k\neq j}^m w_{ik}h_i v_k + \sum_{k=1,k\neq j}^m b_k v_k + \sum_{i=1}^n c_i h_i)}{\sum_v \exp(\sum_{i=1}^n \sum_{j=1}^m w_{ij}h_i v_j + \sum_{j=1}^m b_j v_j + \sum_{i=1}^n c_i h_i)}$$

$$= \frac{\exp(\sum_{i=1}^n w_{ij}h_i + bj + \sum_{i=1}^n c_i h_i)\sum_{v_{k\neq j}} \exp(\sum_{i=1}^n \sum_{k=1,k\neq j}^m w_{ik}h_i v_k + \sum_{k=1,k\neq j}^m b_k v_k)}{\sum_{v_{k\neq j},v_j} \exp\left(\sum_{i=1}^n w_{ij}h_i v_j + b_j v_j + \sum_{i=1}^n c_i h_i\right)\exp(\sum_{i=1}^n \sum_{k=1,k\neq j}^m w_{ik}h_i v_k + \sum_{k=1,k\neq j}^m b_k v_k)}$$

$$= \frac{\exp(\sum_{i=1}^n w_{ij}h_i + bj + \sum_{i=1}^n c_i h_i)\sum_{v_{k\neq j}} \exp(\sum_{i=1}^n \sum_{k=1,k\neq j}^m w_{ik}h_i v_k + \sum_{k=1,k\neq j}^m b_k v_k)}{\sum_{v_j} \exp(\sum_{i=1}^n w_{ij}h_i v_j + b_j v_j + \sum_{i=1}^n c_i h_i)\sum_{v_{k\neq j}} \exp(\sum_{i=1}^n \sum_{k=1,k\neq j}^m w_{ik}h_i v_k + \sum_{k=1,k\neq j}^m b_k v_k)}$$

$$= \frac{\exp(\sum_{i=1}^n w_{ij}h_i + bj)}{1 + \exp(\sum_{i=1}^n w_{ij}h_i + bj)} = \frac{1}{1 + \exp(-\sum_{i=1}^n w_{ij}h_i - bj)} = \sigma(\sum_{i=1}^n w_{ij}h_i + bj)$$

$$p(h_i = 1|v) = \frac{\sum_{h_{k\neq i}} p(hi = 1, h_{k\neq i}, v)}{\sum_h p(h,v)}$$

$$= \frac{\exp(\sum_{j=1}^m w_{ij}v_j + ci + \sum_{j=1}^m b_j v_j)\sum_{h_{k\neq i}} \exp(\sum_{k=1,k\neq i}^n \sum_{j=1}^m w_{kj}h_k v_j + \sum_{k=1,k\neq i}^n c_k h_k)}{\sum_{h_i} \exp(\sum_{j=1}^m w_{ij}h_i v_j + c_i h_i + \sum_{j=1}^m b_j v_j)\sum_{h_{k\neq i}} \exp(\sum_{k=1,k\neq i}^n \sum_{j=1}^m w_{kj}h_k v_j + \sum_{k=1,k\neq i}^n c_k h_k)}$$

$$= \frac{\exp(\sum_{j=1}^m w_{ij}v_j + ci)}{1 + \exp(\sum_{j=1}^m w_{ij}v_j + ci)} = \frac{1}{1 + \exp(-\sum_{j=1}^m w_{ij}v_j - ci)} = \sigma(\sum_{j=1}^m w_{ij}v_j + ci)$$

也就是说$p(h_i = 1|v)$是可以求的。剩下的麻烦全部集中在第二项了。

第二项就是上面的导数化简后的减号后面的那一项。要求第二项，就要遍历所有可能的 v 的值，然后根据公式去计算几个梯度的值，那样够麻烦的了，还好蒙特卡罗给出了一个偷懒的方法，见后面的章。

只要抽取一堆样本，这些样本符合 RBM 网络表示的 Gibbs 分布的（也就是符合θ参数确定的 Gibbs 分布 p(x)的），就可以把上面的三个偏导数估算出来。

对于上面的情况，是这么处理的，对每个训练样本 x，都用某种抽样方法抽取一个它对应的符合 RBM 网络表示的 Gibbs 分布的样本（对应的意思就是符合θ参数确定的 Gibbs 分布 p(x)的），假如叫 y；那么，对于整个的训练集{x1,x2,⋯xl}来说，就得到了一组符合 RBM 网络表示的 Gibbs 分布的样本{y1,y2,⋯,yl}，然后拿这组样本去估算第二项，那么梯度就可以用下面的公式来近似了

$$\frac{\partial \ln p(v)}{\partial w_{ij}} = p(h_i = 1|v)v_j - \sum_v p(v)p(h_i = 1|v)v_j = p(h_i = 1|v)v_j - \frac{1}{l}\sum_{k=1}^{l} p\left(h_i = 1\middle|v_{yk}\right)v_{ykj}$$

$$\frac{\partial \ln p(v)}{\partial b_j} = v_j - \sum_v p(v)v_j = v_j - \frac{1}{l}\sum_{k=1}^{l} v_{ykj}$$

$$\frac{\partial \ln p(v)}{\partial c_i} = p(h_i = 1|v) - \sum_v p(v)p(h_i = 1|v) = p(h_i = 1|v) - \frac{1}{l}\sum_{k=1}^{l} p\left(h_i = 1\middle|v_{yk}\right)$$

其中 xk 表示第 k 个训练样本，yk 表示第 k 个训练样本对应的 RBM 网络表示的 Gibbs 分布（不妨将这个分布称为 R）的样本（yk 是根据分布 R 抽取的样本，而且这个样本是从 xk 出发，用 Gibbs 抽样抽取出来的，也就是说 yk 服从分布 R，可以用来估算第二项，同时 yk 跟 xk 有关），Vyk 表示可视节点取值为 yk 的时候的状态，那么 Vykj 就表示 yk 的第 j 个特征的取值。

这样，梯度出来了，这个极大似然问题可以解了，最终经过若干论迭代，就能得到那几个参数 w，b，c 的解。

式子中的 v 是指{x1,x2,⋯xl}中的一个样本，因为$\frac{\partial \ln L(\theta|v)}{\partial \theta} = \sum_v \frac{\partial \ln p(v)}{\partial \theta}$，对样本进行累加时，第一项是对所有样本进行累加，而第二项都是一样的，所以累加后 1/l 就没有了，只有对 l 项 y 进行累加，到了下面 CD-k 算法的时候，每次只对一个 x 和一个 y 进行处理，最外层对 x 做 L 次循环后得到的累加结果是一样的

上面提到的“某种抽样方法”，现在一般就用 Gibbs 抽样，然后 hinton 教授还根据这个弄出了一个 CD-k 算法，就是用来解决 RBM 的抽样的。下面一章就介绍这个吧。

3.5 用到的抽样方法

一般来说，在 hinton 教授还没弄出 CD-k 之前，解决 RBM 的抽样问题是用 Gibbs 抽样的。

Gibbs 抽样是一种基于马尔科夫蒙特卡罗（Markov Chain Monte Carlo,MCMC）策略的抽样方法。具体就是，对于一个 d 维的随机向量 x=(x1,x2,⋯xd)，假设我们无法求得 x 的联合概率分布 p(x)，但我们知道给定 x 的其他分量是，其第 i 个分量 xi 的条件分布，即 p(xi|xi-),xi-=(x1,x2, ⋯ xi-1,xi+1 ⋯ xd)。那么，我们可以从 x 的一个任意状态（如(x1(0),x2(0),⋯,xd(0))）开始，利用条件分布 p(xi|xi-)，迭代地对这状态的每个分量进行抽样，随着抽样次数 n 的增加，随机变量(x1(n),x2(n),⋯,xd(n))的概率分布将以 n 的几何级数的速度收敛与 x 的联合概率分布 p(v)。

Gibbs 抽样其实就是可以让我们可以在位置联合概率分布 p(v)的情况下对其进行抽样。

基于 RBM 模型的对称结构，以及其中节点的条件独立行，我们可以使用 Gibbs 抽样方法得到服从 RBM 定义的分布的随机样本。在 RBM 中进行 k 步 Gibbs 抽样的具体算法为：用一个训练样本（或者可视节点的一个随机初始状态）初始化可视节点的状态 v0，交替进行

下面的抽样：

$$\begin{aligned} \mathbf{h}_0 &\sim P(\mathbf{h}|\mathbf{v}_0), \qquad \mathbf{v}_1 \sim P(\mathbf{v}|\mathbf{h}_0), \\ \mathbf{h}_1 &\sim P(\mathbf{h}|\mathbf{v}_1), \qquad \mathbf{v}_2 \sim P(\mathbf{v}|\mathbf{h}_1), \\ &\cdots\cdots, \qquad \mathbf{v}_{k+1} \sim P(\mathbf{v}|\mathbf{h}_k). \end{aligned}$$

在抽样步数 n 足够大的情况下，就可以得到 RBM 所定义的分布的样本（即符合θ参数确定的 Gibbs 分布的样本）了，得到这些样本我们就可以拿去计算梯度的第二项了。

可以看到，上面进行了 k 步的抽样，这个 k 一般还要比较大，所以是比较费时间的，尤其是在训练样本的特征数比较多（可视节点数大）的时候，所以 hinton 教授就弄一个简化的版本，叫做 CD-k，也就对比散度。

对比散度是英文 Contrastive Divergence（CD）的中文翻译。与 Gibbs 抽样不同，hinton 教授指出当使用训练样本初始化 v0 的时候，仅需要较少的抽样步数（一般就一步）就可以得到足够好的近似了。

在 CD 算法一开始，可见单元的状态就被设置为一个训练样本，并用上面的几个条件概率$\mathrm{p}(h_i = 1|\mathrm{v})$来对隐藏节点的每个单元都从{0,1}中抽取到相应的值，然后再利用$\mathrm{p}(\mathrm{v}_j = 1|\mathrm{h})$来对可视节点的每个单元都从{0,1}中抽取相应的值，这样就得到了 v1 了，一般 v1 就够了，就可以拿来估算梯度了。

下面给出 RBM 的基于 CD-k 的快速学习的主要步骤。

Algorithm 1. k-step contrastive divergence

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

```
init Δw_ij = Δb_j = Δc_i = 0 for i = 1,...,n, j = 1,...,m
forall the v ∈ S do
    v^(0) ← v
    for t = 0,...,k−1 do
        for i = 1,...,n do sample h_i^(t) ~ p(h_i | v^(t))
        for j = 1,...,m do sample v_j^(t+1) ~ p(v_j | h^(t))
    for i = 1,...,n, j = 1,...,m do
        Δw_ij ← Δw_ij + p(H_i = 1 | v^(0))·v_j^(0) − p(H_i = 1 | v^(k))·v_j^(k)
        Δb_j ← Δb_j + v_j^(0) − v_j^(k)
        Δc_i ← Δc_i + p(H_i = 1 | v^(0)) − p(H_i = 1 | v^(k))
```

其中，之所以第二项没有了那个 1/l，就是因为这个梯度会对所有样本进行累加（极大似然是多个样本的梯度的和），最终加和的结果跟现在这样算是相等的。

3.6 马尔科夫蒙特卡罗简介

下面简介一下马尔科夫蒙特卡罗（MCMC）方法。

最早的蒙特卡罗方法，是由物理学家发明的，旨在于通过随机化的方法计算积分。假设

给定函数 h(x)，我们想计算积分$\int_a^b h(x)dx$，但是又没有办法对区间内的所有 x 的取值都算一遍，我们可以将 h(x)分解为某个函数 f(x)和一个定义在(a, b)上的概率密度函数 p(x)的乘积。这样整个积分就可以写成：

$$\int_a^b h(x)dx = \int_a^b f(x)p(x)dx = E_{p(x)}[f(x)]$$

这样一来，原积分就等同于 f(x)在 p(x)这个分布上的均值(期望)。这时，如果我们从分布 p(x)上采集大量的样本 x1, x2, ..., xn，这些样本符合分布 p(x)，即$\forall i, x_i / \sum_i x_i \approx p(x_i)$，那么，我们就可以通过这些样本来逼近这个均值：

$$\int_a^b h(x)dx = E_{p(x)}[f(x)] = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

这就是蒙特卡罗方法的基本思想。

然后剩下的就是怎么样能够采样到符合分布 p(x)的样本了，这个简单来说就是一个随机的初始样本，通过马尔科夫链进行多次转移，最终就能得到符合分布 p(x)的样本。上面介绍的 Gibbs 是一种比较常用的抽样算法。

有关 Gibbs 抽样的相关内容可以参看文献【12】。

致谢

Deep Learning 高质量交流群里的多位同学：@ zeonsunlight，@marvin，@tornadomeet，@好久不见，@zouxy09

他们在我写这个笔记的过程中提供了多方面的资料。

尤其是@ zeonsunlight，帮助我弄清楚了很多概念，在此表示特别的感谢。

本人愚钝，多数问题都需要各位同学的多次点拨，在此向他们表示衷心的感谢。

这个笔记的其他方面介绍，我觉得都算清楚，唯独在说能量模型的时候，感觉是没把能量模型介绍清楚，有懂的前辈希望能指导一下。

最后，还是得说，本人才疏学浅，肯定有大量的错误，希望大家能多多包涵并且指出，让我能及时改正。

参考文献

[1] An Introduction to Restricted Boltzmann Machines. Asja Fischer and Christian Igel

[2] 受限波尔兹曼机简介 张春霞，姬楠楠，王冠伟

[3] Learning Deep Architectures for AI Yoshua Bengio

[4] http://blog.csdn.net/cuoqu/article/details/8886971 DeepLearning（深度学习）原理与实现（一）marvin521

[5] http://blog.csdn.net/zouxy09/article/details/8775360 Deep Learning（深度学习）学习笔记整理系列 zouxy09

[6] http://www.cnblogs.com/tornadomeet/archive/2013/03/27/2984725.html Deep learning：十九(RBM 简单理解) tornadomeet

[7] http://blog.csdn.net/celerychen2009/article/details/8984316 受限波尔兹曼机 celerychen2009

[8] http://blog.csdn.net/zouxy09/article/details/8537620 从最大似然到 EM 算法浅解 zouxy09

[9] http://www.sigvc.org/bbs/thread-513-1-3.html 《RBM 和 DBN 的训练》 王颖

[10] 神经网络原理叶[M] 叶世伟，史忠植：机械工业出版社

[11] http://www.sigvc.org/bbs/thread-512-1-1.html 《Restricted Boltzmann Machine (RBM) 推导》朱飞云

[12] http://www.52nlp.cn/lda-math-mcmc-%E5%92%8C-gibbs-sampling1 gibbs 抽样相关