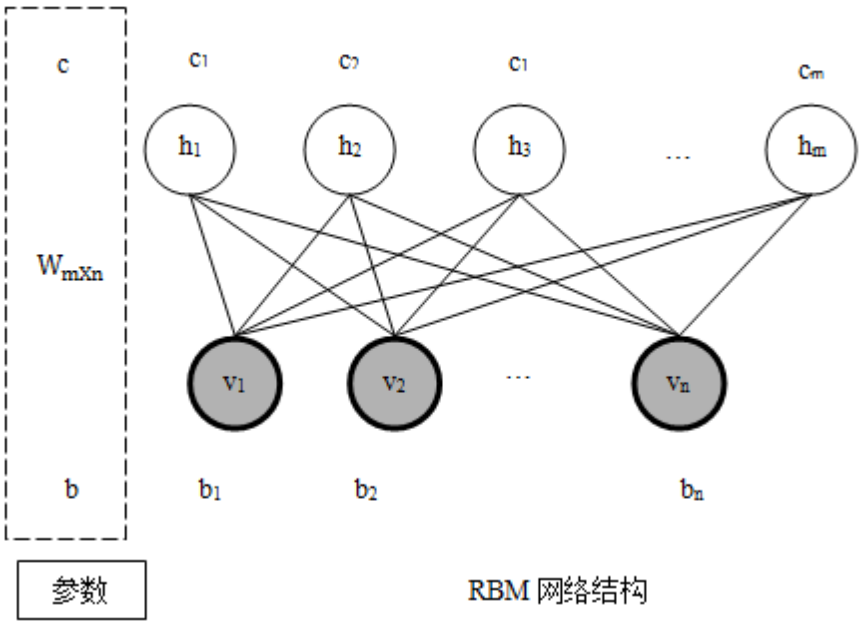
深度学习读书笔记

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 n 个可视节点和 m 个隐藏节点，其中每个可视节点只和 m 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 m 个隐藏节点的影响，对于每个隐藏节点也是，只受 n 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$W_{m\times n}$，一个是可视节点的偏移量$\mathrm{b}=(\mathrm{b1},\mathrm{b2}\cdots\mathrm{bn})$，一个是隐藏节点的偏移量$\mathrm{c}=(\mathrm{c1},\mathrm{c2}\cdots\mathrm{cm})$，这几个参数决定了 RBM 网络将一个 n 维的样本编码成一个什么样的 m 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$\forall \mathrm{i},\mathrm{j},v_i\in\{0,1\},h_j\in\{0,1\}$。

一个训练样本x过来了取值为$\mathrm{x}=(\mathrm{x1},\mathrm{x2}\cdots\mathrm{xn})$，根据 RBM 网络，可以得到这个样本的 m 维的编码后的样本$\mathrm{y}=(\mathrm{y1},\mathrm{y2}\cdots\mathrm{ym})$，这 m 维的编码也可以认为是抽取了 m 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$\mathrm{x}=(\mathrm{x1},\mathrm{x2}\cdots\mathrm{xn})$，隐藏节点的第 j 个特征的取值为 1 的概率为$\mathrm{p}(\mathrm{hj}=1|\mathrm{v})=\sigma(\sum_{i=1}^{n}wji\times\mathrm{vi}+\mathrm{cj})$，其中的 v 取

值就是 x，hj 的取值就是 yj，也就是说，编码后的样本 y 的第 j 个位置的取值为 1 的概率是 $p(hj=1|v)$。所以，生成 yj 的过程就是：

i）先利用公式$p(hj=1|v)=\sigma(\sum_{i=1}^{n} wji \times vi + cj)$，根据 x 的值计算概率$p(hj=1|v)$，其中 vi 的取值就是 xi 的值。

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

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$p(vi=1|h)=\sigma(\sum_{j=1}^{m} wji \times hj + bi)$，根据 y 的值计算概率$p(hj=1|v)$，其中 hj 的取值就是 yj 的值。

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

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 能量模型来源

在说 RBM 之前，先来说点其他的，就是能量模型。因为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。

在统计力学中，基于能量函数的模型(Engery based model)称为能量模型。能量方法来源于热动力学，分子在高温中运动剧烈，能够克服局部约束（分子之间的一些物理约束，比如键值吸引力等），在逐步降到低温时，分子最终会排列出有规律的结构，此时也是低能量状态。受此启发，早期的模拟退火算法就是在高温中试图跳出局部最小。随机场作为物理模型之一，也引入了此方法。

3.2.2 能量模型作用和定义

为什么要弄这个能量模型呢？，因为能凑出个问题来求解。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。

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，其实实际中也可以不用概率表示，比如立体匹配中直接用两个像素点的像素差作为能量，所有像素对之间的能量和最小时的配置即为目标解。

RBM 作为一种概率图模型，引入概率就是为了方便采样，因为在 CD（contrastive divergence）算法中采样部分扮演着模拟求解梯度的角色。

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)，然后分别代人上面的公式，就能得到这个连接结构之间的能

量。

下面就开始想办法求解这个能量模型了。但是要解一个问题得有一个东西，就是是目标函数（也就是全局解的度量），我们对目标函数求个偏导，就可以用梯度法迭代地去解这个问题了。

那么，我们可以把所有可视节点的取值和隐藏节点的取值的能量累加起来，累加的结果作为 RBM 的目标函数。

然后解起来就麻烦了，对每个样本，都要列举它能对应的所有编码后的样本（隐藏节点取值），这样才能计算能量，那指数级别的计算就难免了，这样，解这个问题恐怖就不实际了，因为用穷举法计算梯度什么的，实在太耗计算资源。当然这个说法可能是我一人的片面看法，大家有更好的解释麻烦提醒一下，我把它修改了。

然后很自然的就会想容易的方法，这里给出来的偷懒方法就是引人概率，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

为了引人概率，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

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

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 从能量最小到极大似然

上面我们得到了一个样本和其对应编码的联合概率，也就是得到了一个 Gibbs 分布，我们引人概率的目的是为了方便求解的。但是我们实际求解的目标是能量最小。

下面来看看怎么从能量最小变成用概率表示。内容是来自《神经网络原理》那本书。

在统计力学上的说法也是——能量低的状态比能量高的状态发生的概率高。

定义一个叫做自由能量的东西，是从统计力学来的概念，

$$\text{FreeEnergy(v)} = -\ln \sum_{h} e^{-E(v,h)}$$

然后 p(v)可以重新写成

$$\text{p(v)} = \frac{e^{-\text{FreeEnergy}(v)}}{Z}$$

其中$\text{Z} = \sum_{v,h} e^{-E(v,h)}$，然后对上面的等式两边取对数，可以得到

$$\text{lnp(v)} = -\text{FreeEnergy(v)} - \text{lnZ}$$

从这里可以看出，自由能量可以用 lnp(v)来度量，当然，是反向的关系，自由能量小时候 p(v)大，两边做个累加

$$\sum_{\text{v}} lnp(v) = -\sum_{v} \text{FreeEnergy(v)} - \sum_{v} lnZ$$

最右边那个是常数，就忽略算了；右边的第一项就是整个网络的自由能量总和的负值，左边可以认为是概率 p(v)的连乘的对数，也就是似然函数。这就得到了一个物理系统（RBM 网络）的自由能量的总和，跟 p(v)的对数和的关系，p(v)的对数和也可以称为对数似然函数。

这样就能得到一个结论了，一个系统的自由能量的总和最小的时候，正是$\prod_{\text{v}} p(v)$最大的时候，也就是说，用极大似然估计去求得的参数，能让 RBM 系统的自由能量的总和最小。

在统计力学上，有一个最小自由能量原则——随机系统变量的自由能量的最小值可以在热平衡是达到，此时系统服从 Gibbs 分布。

自然偏爱具有最小自由能量的物理系统。当 RBM 系统达到自由能量最小是，刚好，似然函数也取得最大值，也就是说，我们可以利用极大似然来求 RBM 系统的参数θ（注意是参数是一组参数）。

就有了下面的说法，由于 RBM 是一种随机机器，所以是依赖于概率论来评价其性能的，概率上的标准就是似然函数，也就是要根据极大似然估计去求参数。极大似然估计，就要求到一组参数，使得训练样本的概率最大，也就是$\prod_{\text{v}} p(v)$最大，$\prod_{\text{v}} p(v)$就是似然函数了，使$\prod_{\text{v}} p(v)$最大就是极大似然估计。

下面说说极大似然的意义。

3.4 极大似然的意义

既然要用极大似然来求解，这个当然是有意义的，只是我一直没有找到这方面的资料，下面还是说说我个人的看法吧。

从上面的讨论可以知道，训练 RBM 的目标，就是要求到一组参数，使得似然函数值最大。

在统计力学上，这个似然函数最大的意义就是使得 RBM 网络在一定条件下（单位温度）的自由能量总和达到最小（也即系统能量达到最小）。上面已经说过，系统越有序或者概率分布越集中，系统的能量越小。其实反过来也可以——系统的能量越小，系统越有序或者概率分布越集中。当 RBM 网络的能量达到最小时，系统最有序，概率分布最集中。

当 RBM 网络的概率分布最集中的时候，每个训练样本经过 RBM 网络编码到隐藏节点的的取值的概率也很集中。例如一个样本(1,0,1,0,1)就能以最大的概率编码到(0,1,1)，编码成其他值的概率变小。反过来，从隐藏节点反编码到可视节点的概率也很集中，如(0,1,1)就能大概率地编码到(1,0,1,0,1)。

另外，极大似然还有另外一个特点，就是使得 RBM 系统最好地拟合数据分布，从而也保证了，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。

到这，能量模型跟极大似然的关系说清楚了，作用也聊了一下，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。

下面就说怎么求解了。

3.5 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，设参数为θ（注意是参数是一组参数），则似然函数可以写为

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

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）

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

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。

我们还可以这么理解，第一项等于输入样本数据的自由能量函数期望值，而第二项是模型产生的样本数据（就是符合我们要求的那个 Gibbs 分布的样本）的自由能量函数期望值。

第一项是可以求的，因为只要对每个训练遍历它可能对应的隐藏节点的值。

第二项也是可以求的，但是要对 v 的所以可能的取值都遍历一趟，这就可能没法算了。

为了进行下面的讨论，我们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，

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

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

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

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

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

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

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

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

$$
\begin{aligned}
p(v_i=1|\mathbf{h},\theta) &= \frac{\sum_{\mathbf{v}_{k\neq i}} p(v_i=1,\mathbf{v}_{k\neq i},\mathbf{h}|\theta)}{\sum_{\mathbf{v}} p(\mathbf{v},\mathbf{h}|\theta)} \\
&= \frac{\sum_{\mathbf{v}_{k\neq i}} \exp\left(W_{ij}h_j + b_i + \sum_{k\neq i,j} W_{kj}h_j v_k + \sum_{k\neq i} v_k b_i + \sum_j c_j h_j\right)}{\sum_{\mathbf{v}} \exp\left(\sum_{ij} W_{ij}h_j v_i + \sum_i v_i b_i + \sum_j c_j h_j\right)} \\
&= \frac{\exp\left(\sum_j W_{ij}h_j + b_i + \sum_j c_j h_j\right)\sum_{\mathbf{v}_{k\neq i}} \exp\left(\sum_{k\neq i,j} W_{kj}h_j v_k + \sum_{k\neq i} v_k b_k\right)}{\sum_{\mathbf{v}_{k\neq i},v_i} \exp\left(\sum_j W_{ij}h_j v_i + v_i b_i + \sum_j c_j h_j\right)\exp\left(\sum_{k\neq i,j} W_{kj}h_j v_k + \sum_{k\neq i} v_k b_k\right)} \\
&= \frac{\exp\left(\sum_j W_{ij}h_j + b_i + \sum_j c_j h_j\right)\sum_{\mathbf{v}_{k\neq i}} \exp\left(\sum_{k\neq i,j} W_{kj}h_j v_k + \sum_{k\neq i} v_k b_k\right)}{\sum_{v_i} \exp\left(\sum_j W_{ij}h_j v_i + v_i b_i + \sum_j c_j h_j\right)\sum_{\mathbf{v}_{k\neq i}} \exp\left(\sum_{k\neq i,j} W_{kj}h_j v_k + \sum_{k\neq i} v_k b_k\right)} \\
&= \frac{\exp\left(\sum_j W_{ij}h_j + b_i\right)}{1+\exp\left(\sum_j W_{ij}h_j + b_i\right)} = \frac{1}{1+\exp\left(-\sum_j W_{ij}h_j - b_i\right)} \qquad (11)
\end{aligned}
$$

$$
\begin{aligned}
p(h_j=1|\mathbf{v},\theta) &= \frac{\sum_{\mathbf{h}_{k\neq j}} p(h_j,\mathbf{h}_{k\neq j},\mathbf{v}|\theta)}{\sum_{\mathbf{h}} p(\mathbf{h},\mathbf{v})} \\
&= \frac{\exp\left(\sum_i v_i b_i + \sum_i W_{ij}h_j v_i + c_j h_j\right)\sum_{\mathbf{h}_{k\neq j}} \exp\left(\sum_{i,k\neq j} W_{ik}h_k v_i + \sum_{k\neq j} h_k c_k\right)}{\sum_{h_j} \exp\left(\sum_i v_i b_i + \sum_i W_{ij}h_j v_i + c_j h_j\right)\sum_{\mathbf{h}_{k\neq j}} \exp\left(\sum_{i,k\neq j} W_{ik}h_k v_i + \sum_{k\neq j} h_k c_k\right)} \\
&= \frac{1}{1+\exp\left(-\sum_i W_{ij}v_i - c_j\right)} = \sigma\left(\sum_i W_{ij}v_i + c_j\right) \qquad (12)
\end{aligned}
$$

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

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，这些样本是符合θ参数确定的 Gibbs 分布的，也就是符合要求的模型样本，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样本（对应的意思就是符合θ参数确定的 Gibbs 分布的），假如叫 y；那么，对于整个的训练集{x1,x2,⋯xn}来说，就得到了一组符合θ参数确定的 Gibbs 分布的样本{y1,y2,⋯,yn}，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

$$
\frac{\partial \ln p(\mathrm{v})}{\partial w_{ij}} = p(h_i=1|v)v_j - \sum_v p(v)p(h_i=1|v)v_j = p(h_i=1|v)v_j - \frac{1}{\mathrm{n}}\sum_{i=1}^{n} p\left(h_i=1\middle|v_y\right)v_{yj}
$$

$$
\frac{\partial \ln p(\mathrm{v})}{\partial b_j} = v_j - \sum_v p(v)v_j = v_j - \frac{1}{\mathrm{n}}\sum_{i=1}^{n} v_{yj}
$$

$$
\frac{\partial \ln p(\mathrm{v})}{\partial c_i} = p(h_i=1|v) - \sum_v p(v)p(h_i=1|v) = p(h_i=1|v) - \frac{1}{\mathrm{n}}\sum_{i=1}^{n} p\left(h_i=1\middle|v_y\right)
$$

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。

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。

3.6 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/n，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。

3.7 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。