受限玻尔兹曼机(Restricted Boltzmann Machine)

作者：凯鲁嘎吉 - 博客园 http://www.cnblogs.com/kailugaji/

受限玻尔兹曼机（Restricted Boltzmann Machine，RBM）是一个二分图结构的无向图模型，如图12.3所示。受限玻尔兹曼机中的变量也分为隐藏变量和可观测变量。我们分别用可观测层和隐藏层来表示这两组变量。同一层中的节点之间没有连接，而不同层一个层中的节点与另一层中的所有节点连接，这和两层的全连接神经网络的结构相同。

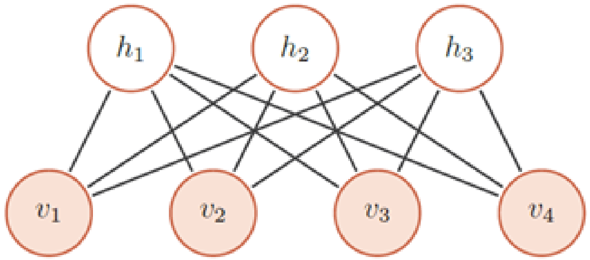


图 12.3 一个有 7 个变量的受限玻尔兹曼机

一个受限玻尔兹曼机由 m_1 个可观测变量和 m_2 个隐变量组成，其定义如下：

- 可观测的随机向量 $\mathbf{v} = [v_1, \cdots, v_{m_1}]^{\mathrm{T}}$；

- 隐藏的随机向量 $\mathbf{h} = [h_1, \cdots, h_{m_2}]^{\mathrm{T}}$；
- 权重矩阵 $W \in \mathbb{R}^{m_1 \times m_2}$，其中每个元素 w_{ij} 为可观测变量 v_i 和隐变量 h_j 之间边的权重；
- 偏置 $\mathbf{a} \in \mathbb{R}^{m_1}$ 和 $\mathbf{b} \in \mathbb{R}^{m_2}$，其中 a_i 为每个可观测的变量 v_i 的偏置，b_j 为每个隐变量 h_j 的偏置。

受限玻尔兹曼机的能量函数定义为

$$\begin{aligned} E(\mathbf{v}, \mathbf{h}) &= -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{ij} h_j && (12.28) \\ &= -\mathbf{a}^{\mathrm{T}}\mathbf{v} - \mathbf{b}^{\mathrm{T}}\mathbf{h} - \mathbf{v}^{\mathrm{T}} W \mathbf{h}, && (12.29) \end{aligned}$$

受限玻尔兹曼机的联合概率分布 $p(\mathbf{v}, \mathbf{h})$ 定义为

$$\begin{aligned} p(\mathbf{v}, \mathbf{h}) &= \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h})) && (12.30) \\ &= \frac{1}{Z} \exp(\mathbf{a}^{\mathrm{T}}\mathbf{v}) \exp(\mathbf{b}^{\mathrm{T}}\mathbf{h}) \exp(\mathbf{v}^{\mathrm{T}} W \mathbf{h}), && (12.31) \end{aligned}$$

其中 $Z = \sum_{\mathbf{v},\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))$ 为配分函数。

1. 生成模型

在给定受限玻尔兹曼机的联合概率分布 $p(\mathbf{h},\mathbf{v})$ 后，可以通过吉布斯采样方法生成一组服从 $p(\mathbf{h},\mathbf{v})$ 分布的样本。

12.2.1.1　全条件概率

吉布斯采样需要计算每个变量 V_i 和 H_j 的全条件概率。受限玻尔兹曼机中同层的变量之间没有连接。从无向图的性质可知，在给定可观测变量时，隐变量之间互相条件独立。同样在给定隐变量时，可观测变量之间也互相条件独立。即有

$$p(v_i|\mathbf{v}_{\setminus i},\mathbf{h}) = p(v_i|\mathbf{h}), \tag{12.32}$$

$$p(h_j|\mathbf{v},\mathbf{h}_{\setminus j}) = p(h_j|\mathbf{v}), \tag{12.33}$$

其中 $\mathbf{v}_{\setminus i}$ 为除变量 V_i 外其它可观测变量的取值，$\mathbf{h}_{\setminus j}$ 为除变量 H_j 外其它隐变量的取值。因此，V_i 的全条件概率只需要计算 $p(v_i|\mathbf{h})$，而 H_j 的全条件概率只需要计算 $p(h_j|\mathbf{v})$。

定理 12.2 – 受限玻尔兹曼机中变量的条件概率： 在受限玻尔兹曼

机中，每个可观测变量和隐变量的条件概率为

$$p(v_i = 1|\mathbf{h}) = \sigma\Big(a_i + \sum_j w_{ij} h_j\Big), \tag{12.34}$$

$$p(h_j = 1|\mathbf{v}) = \sigma\Big(b_j + \sum_i w_{ij} v_i)\Big), \tag{12.35}$$

其中 σ 为 logistic sigmoid 函数。

证明. (1) 我们先计算 $p(h_j = 1|\mathbf{v})$。可观测层变量 $\mathbf{v}$ 的边际概率为

$$P(\mathbf{v}) = \sum_{\mathbf{h}} P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h})) \tag{12.36}$$

$$= \frac{1}{Z} \sum_{\mathbf{h}} \exp\left(\mathbf{a}^{\mathrm{T}}\mathbf{v} + \sum_j b_j h_j + \sum_i \sum_j v_i w_{ij} h_j\right) \tag{12.37}$$

$$= \frac{\exp(\mathbf{a}^{\mathrm{T}}\mathbf{v})}{Z} \sum_{\mathbf{h}} \exp\left(\sum_j h_j (b_j + \sum_i w_{ij} v_i)\right) \tag{12.38}$$

$$= \frac{\exp(\mathbf{a}^{\mathrm{T}}\mathbf{v})}{Z} \sum_{\mathbf{h}} \prod_j \exp\left(h_j (b_j + \sum_i w_{ij} v_i)\right) \tag{12.39}$$

$$= \frac{\exp(\mathbf{a}^{\mathrm{T}}\mathbf{v})}{Z} \sum_{h_1} \sum_{h_2} \cdots \sum_{h_n} \prod_j \exp\left(h_j (b_j + \sum_i w_{ij} v_i)\right) \tag{12.40}$$

$$= \frac{\exp(\mathbf{a}^{\mathrm{T}}\mathbf{v})}{Z} \prod_j \sum_{h_j} \exp\left(h_j(b_j + \sum_i w_{ij}v_i)\right) \tag{12.41}$$

$$= \frac{\exp(\mathbf{a}^{\mathrm{T}}\mathbf{v})}{Z} \prod_j \left(1 + \exp(b_j + \sum_i w_{ij}v_i)\right). \tag{12.42}$$

固定 $h_j = 1$ 时，$p(h_j = 1, \mathbf{v})$ 的边际概率为

$$p(h_j = 1, \mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}, h_j = 1} \exp\left(-E(\mathbf{v}, \mathbf{h})\right) \tag{12.43}$$

$$= \frac{\exp(\mathbf{a}^{\mathrm{T}}\mathbf{v})}{Z} \prod_{k, k \neq j} \left(1 + \exp(b_k + \sum_i w_{ik}v_i)\right) \exp(b_j + \sum_i w_{ij}v_i). \tag{12.44}$$

由公式 (12.42) 和 (12.44)，可以计算隐藏单元 h_j 的条件概率为：

$$p(h_j = 1|\mathbf{v}) = \frac{p(h_i = 1, \mathbf{v})}{p(\mathbf{v})} \tag{12.45}$$

$$= \frac{\exp(b_j + \sum_i w_{ij}v_i)}{1 + \exp(b_j + \sum_i w_{ij}v_i)} \tag{12.46}$$

$$= \sigma\left(b_j + \sum_i w_{ij} v_i\right). \tag{12.47}$$

（2）同理，条件概率 $p(v_i = 1|\mathbf{h})$ 为

$$p(v_i = 1|\mathbf{h}) = \sigma\left(a_i + \sum_j w_{ij} h_j\right). \tag{12.48}$$

□

公式 (12.47) 和 (12.48) 也可以写为向量形式。

$$p(\mathbf{h} = \mathbf{1}|\mathbf{v}) = \sigma\left(W^{\mathrm{T}}\mathbf{v} + \mathbf{b}\right) \tag{12.49}$$

$$p(\mathbf{v} = \mathbf{1}|\mathbf{h}) = \sigma\left(W\mathbf{h} + \mathbf{a}\right). \tag{12.50}$$

在受限玻尔兹曼机的全条件概率中，可观测变量之间互相条件独立，隐变量之间也互相条件独立。因此，受限玻尔兹曼机可以并行地对所有的可观测变量（或所有的隐变量）同时进行采样，从而可以更快地达到热平衡状态。

12.2.1.2 吉布斯采样

受限玻尔兹曼机的采样过程如下：

- （给定）或随机初始化一个可观测的向量 $\mathbf{v}_0$，计算隐变量的概率，并从中采样一个隐向量 $\mathbf{h}_0$；
- 基于 $\mathbf{h}_0$，计算可观测变量的概率，并从中采样一个个可观测的向量 $\mathbf{v}_1$；
- 重复 t 次后，获得 $(\mathbf{v}_t, \mathbf{h}_t)$；
- 当 $t \to \infty$ 时，$(\mathbf{v}_t, \mathbf{h}_t)$ 的采样服从 $p(\mathbf{v}, \mathbf{h})$ 分布。

图12.4也给出了上述过程的示例。

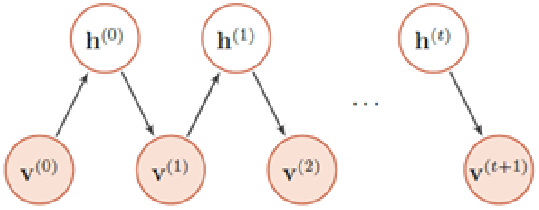


图 12.4 受限玻尔兹曼机的采样过程

2. 参数学习

和玻尔兹曼机一样，受限玻尔兹曼机通过最大化似然函数来找到最优的参数 $W, \mathbf{a}, \mathbf{b}$。给定一组训练样本 $\mathcal{D} = \{\hat{\mathbf{v}}^{(1)}, \hat{\mathbf{v}}^{(2)}, \cdots, \hat{\mathbf{v}}^{(N)}\}$，其对数似然函数为

$$\mathcal{L}(\mathcal{D}; W, \mathbf{a}, \mathbf{b}) = \frac{1}{N}\sum_{n=1}^{N} \log p(\hat{\mathbf{v}}^{(n)}; W, \mathbf{a}, \mathbf{b}). \tag{12.51}$$

和玻尔兹曼机类似，在受限玻尔兹曼机中，对数似然函数 $\mathcal{L}(\mathcal{D}; W, \mathbf{b})$ 对参数 w_{ij}, a_i, b_j 的偏导数为

$$\frac{\partial \mathcal{L}(\mathcal{D}; W, \mathbf{a}, \mathbf{b})}{\partial w_{ij}} = \mathbb{E}_{\hat{p}(\mathbf{v})}\mathbb{E}_{p(\mathbf{h}|\mathbf{v})}[v_i h_j] - \mathbb{E}_{p(\mathbf{v},\mathbf{h})}[v_i h_j], \tag{12.52}$$

$$\frac{\partial \mathcal{L}(\mathcal{D}; W, \mathbf{a}, \mathbf{b})}{\partial a_i} = \mathbb{E}_{\hat{p}(\mathbf{v})}\mathbb{E}_{p(\mathbf{h}|\mathbf{v})}[v_i] - \mathbb{E}_{p(\mathbf{v},\mathbf{h})}[v_i], \tag{12.53}$$

$$\frac{\partial \mathcal{L}(\mathcal{D}; W, \mathbf{a}, \mathbf{b})}{\partial b_j} = \mathbb{E}_{\hat{p}(\mathbf{v})}\mathbb{E}_{p(\mathbf{h}|\mathbf{v})}[h_j] - \mathbb{E}_{p(\mathbf{v},\mathbf{h})}[h_j], \tag{12.54}$$

其中 $\hat{p}(\mathbf{v})$ 为训练数据集上 $\mathbf{v}$ 的实际分布。

公式(12.52)、(12.53)和(12.54)中都需要计算配分函数 Z 以及两个期望 $\mathbb{E}_{p(\mathbf{h}|\mathbf{v})}$ 和 $\mathbb{E}_{p(\mathbf{h},\mathbf{v})}$，因此很难计算，一般需要通过 MCMC 方法来近似计算。

首先，将可观测向量 $\mathbf{v}$ 设为训练样本中的值并固定，然后根据条件概率对隐向量 $\mathbf{h}$ 进行采样，这时受限玻尔兹曼机的值记为 $\langle\cdot\rangle_{\texttt{data}}$。然后再不固定可观测向量 $\mathbf{v}$，通过吉布斯采样来轮流更新 $\mathbf{v}$ 和 $\mathbf{h}$。当达到热平衡状态时，采集 $\mathbf{v}$ 和 $\mathbf{h}$ 的值，记为 $\langle\cdot\rangle_{\texttt{model}}$。

采用梯度上升方法时，参数 $W, \mathbf{a}, \mathbf{b}$ 可以用下面公式近似地更新

$$w_{ij} \leftarrow w_{ij} + \alpha\Big(\langle v_i h_j\rangle_{\texttt{data}} - \langle v_i h_j\rangle_{\texttt{model}}\Big), \tag{12.55}$$

$$a_i \leftarrow a_i + \alpha\Big(\langle v_i\rangle_{\texttt{data}} - \langle v_i\rangle_{\texttt{model}}\Big), \tag{12.56}$$

$$b_j \leftarrow b_j + \alpha\Big(\langle h_j\rangle_{\texttt{data}} - \langle h_j\rangle_{\texttt{model}}\Big), \tag{12.57}$$

其中 $\alpha > 0$ 为学习率。

根据受限玻尔兹曼机的条件独立性，可以对可观测变量和隐变量进行分组轮流采样，如图12.4中所示。这样受限玻尔兹曼机的采样效率会比一般的玻尔兹曼机有很大提高，但一般还是需要通过很多步采样才可以采集到符合真实分布的样本。

3．对比散度学习算法

由于受限玻尔兹曼机的特殊结构,因此可以使用一种比吉布斯采样更有效 的学习算法,即对比散度(Contrastive Divergence)对比散度算法仅需k步吉布斯采样。为了提高效率,对比散度算法用一个训练样本作为可观测向量的初始值。然后,交替对可观测向量和隐藏向量进行吉布斯采样,不需要等到收敛,只需要k步就足够了。这就是CD-k 算法。通常,k = 1就可以学得很好。对比散度的流程如算法12.1所示。

算法 12.1: 单步对比散度算法

输入: 训练集: $\hat{\mathbf{v}}^{(n)}, n = 1, \cdots, N$;

学习率: α

初始化: $W \leftarrow 0$, $\mathbf{a} \leftarrow 0$, $\mathbf{b} \leftarrow 0$;

for $t = 1 \cdots T$ **do**

  for $n = 1 \cdots N$ **do**

    选取一个样本 $\hat{\mathbf{v}}^{(n)}$, 用公式 (12.47) 计算 $p(\mathbf{h} = \mathbf{1}|\hat{\mathbf{v}}^{(n)})$, 并根据这个分布采集一个隐向量 $\mathbf{h}$;

    计算**正向梯度** $\hat{\mathbf{v}}^{(n)}\mathbf{h}^{\mathrm{T}}$;

    根据 $\mathbf{h}$, 用公式 (12.48) 计算 $p(\mathbf{v} = \mathbf{1}|\mathbf{h})$, 并根据这个分布采集重构的可见变量 $\mathbf{v}'$;

    根据 $\mathbf{v}'$, 重新计算 $p(\mathbf{h} = \mathbf{1}|\mathbf{v}')$ 并采样一个 $\mathbf{h}'$;

    计算**反向梯度** $\mathbf{v}'\mathbf{h}'^{\mathrm{T}}$;

    更新参数:

    $W \leftarrow W + \alpha(\hat{\mathbf{v}}^{(n)}\mathbf{h}^{\mathrm{T}} - \mathbf{v}'\mathbf{h}'^{\mathrm{T}})$;

    $\mathbf{a} \leftarrow \mathbf{a} + \alpha(\hat{\mathbf{v}}^{(n)} - \mathbf{v}')$;

    $\mathbf{b} \leftarrow \mathbf{b} + \alpha(\mathbf{h} - \mathbf{h}')$;

  end

end

输出: $W, \mathbf{a}, \mathbf{b}$

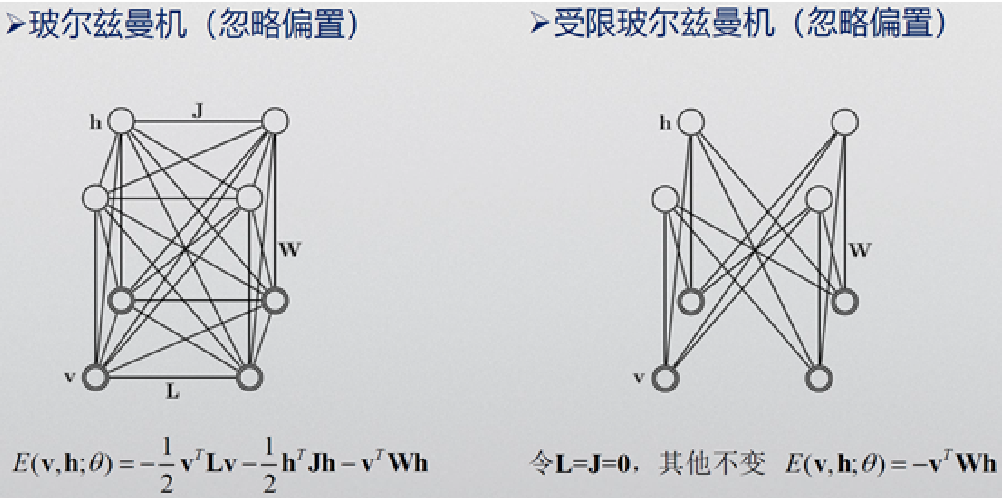
4．MATLAB程序解读

```
% maxepoch  -- 最大迭代次数maximum number of epochs
% numhid    -- 隐含层神经元数number of hidden units
% batchdata -- 分批后的训练数据集the data that is divided into batches (numcases numdims numbatches)
```

```
% restart   -- 如果从第1层开始学习，就置restart为1set to 1 if learning starts from beginning

%作用：训练RBM，利用1步CD算法 直接调用权值迭代公式不使用反向传播
%可见的、二元的、随机的像素通过对称加权连接连接到隐藏的、二元的、随机的特征检测器
epsilonw      = 0.1;   % Learning rate for weights 权重学习率 alpha
epsilonvb     = 0.1;   % Learning rate for biases of visible units 可视层偏置学习率 alpha
epsilonhb     = 0.1;   % Learning rate for biases of hidden units 隐藏层偏置学习率 alpha
weightcost  = 0.0002;   %权衰减，用于防止出现过拟合
initialmomentum  = 0.5; %动量项学习率，用于克服收敛速度和算法的不稳定性之间的矛盾
finalmomentum    = 0.9;

[numcases numdims numbatches]=size(batchdata);%[numcases numdims numbatches]＝[每批中的样本数 每个样本的维数 训练样本批数]

if restart ==1  %是否为重新开始即从头训练
  restart=0;
  epoch=1;

% Initializing symmetric weights and biases. 初始化权重和两层偏置
  vishid     = 0.1*randn(numdims, numhid);% 连接权值Wij 784*1000
  hidbiases  = zeros(1,numhid);% 隐含层偏置项bi
  visbiases  = zeros(1,numdims);% 可视化层偏置项aj

  poshidprobs = zeros(numcases,numhid); %样本数*隐藏层NN数，隐藏层输出p(h1|v0)对应每个样本有一个输出 100*1000
  neghidprobs = zeros(numcases,numhid);  %重构数据驱动的隐藏层
  posprods    = zeros(numdims,numhid);  % 表示p(h1|v0)*v0，用于更新Wij即<vihj>data 784*1000
  negprods    = zeros(numdims,numhid);  %<vihj>recon
  vishidinc  = zeros(numdims,numhid);  % 权值更新的增量 ΔW
  hidbiasinc = zeros(1,numhid);  % 隐含层偏置项更新的增量 1*1000 Δb
  visbiasinc = zeros(1,numdims);  % 可视化层偏置项更新的增量 1*784 Δa
  batchposhidprobs=zeros(numcases,numhid,numbatches);  % 整个数据隐含层的输出  每批样本数*隐含层维度*批数
end

for epoch = epoch:maxepoch  %每个迭代周期
 fprintf(1,'epoch %d\r',epoch);
 errsum=0;
 for batch = 1:numbatches  %每一批样本
 fprintf(1,'epoch %d batch %d\r',epoch,batch);
%%CD-1
%%%%%%%%% START POSITIVE PHASE 正向梯度%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  data = batchdata(:,:,batch);  %data里是100个图片数据
  poshidprobs = 1./(1 + exp(-data*vishid - repmat(hidbiases,numcases,1)));  %隐藏层输出p(h=1|v0)=sigmod函数=1/(1+exp(-wx-b)) 根据这个分布采集一个隐变量h
  batchposhidprobs(:,:,batch)=poshidprobs;  %将输出存入一个三位数组
  posprods    = data' * poshidprobs;  %p(h|v0)*v0 更新权重时会使用到 计算正向梯度vh'
  poshidact   = sum(poshidprobs);  %隐藏层中神经元概率和，在更新隐藏层偏置时会使用到
  posvisact = sum(data);  %可视层中神经元概率和，在更新可视层偏置时会使用到
%%%%%%%%% END OF POSITIVE PHASE  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%gibbs采样
  poshidstates = poshidprobs > rand(numcases,numhid);  %将隐藏层输出01化表示，大于随机概率的置1，小于随机概率的置0，gibbs抽样,设定状态

%%%%%%%%% START NEGATIVE PHASE 反向梯度%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  negdata = 1./(1 + exp(-poshidstates*vishid' - repmat(visbiases,numcases,1)));  %01化表示之后算vt=p(vt|ht-1)重构的数据 p(v=1|h)=sigmod(W*h+a)  采集重构的可见变量v'
  neghidprobs = 1./(1 + exp(-negdata*vishid - repmat(hidbiases,numcases,1)));    %ht=p(h|vt)使用重构数据隐藏层的输出 p(h=1|v)=sigmod(W'*v+b) 采样一个h'
```

```
  negprods  = negdata'*neghidprobs;  %计算反向梯度v'h';
  neghidact = sum(neghidprobs);
  negvisact = sum(negdata);
%%%%%%%%% END OF NEGATIVE PHASE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%更新参数
  err= sum(sum( (data-negdata).^2 ));  %整批数据的误差 ||v-v'||^2
  errsum = err + errsum;

   if epoch>5  %迭代次数不同调整冲量
     momentum=finalmomentum;
   else
     momentum=initialmomentum;
   end

%%%%%%%%% UPDATE WEIGHTS AND BIASES 更新权重和偏置%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    vishidinc = momentum*vishidinc + ...
                epsilonw*( (posprods-negprods)/numcases - weightcost*vishid); %权重的增量 ΔW=alpha*(vh'-v'h')
    visbiasinc = momentum*visbiasinc + (epsilonvb/numcases)*(posvisact-negvisact);  %可视层增量 Δa=alpha*(v-v')
    hidbiasinc = momentum*hidbiasinc + (epsilonhb/numcases)*(poshidact-neghidact);  %隐含层增量 Δb=alpha*(h-h')

    vishid = vishid + vishidinc; %a=a+Δa
    visbiases = visbiases + visbiasinc; %W=W+ΔW
    hidbiases = hidbiases + hidbiasinc;  %b=b+Δb
%%%%%%%%%%%%%%%% END OF UPDATES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  end
  fprintf(1, 'epoch %4i error %6.1f  \n', epoch, errsum);
end
```

5. 玻尔兹曼机与受限玻尔兹曼机



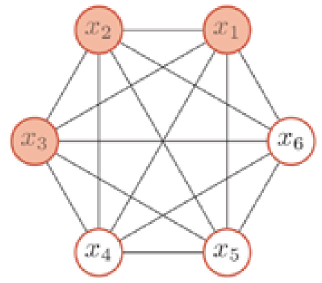
玻尔兹曼机（Boltzmann machine）

- 玻尔兹曼机是一个特殊的概率无向图模型。
  - 每个随机变量是二值的
  - 所有变量之间是全连接的
  - 整个能量函数定义为

$$E(\mathbf{x}) \triangleq E(\mathbf{X}=\mathbf{x}) = -\left(\sum_{i<j} w_{ij}\, x_i\, x_j + \sum_i b_i\, x_i\right)$$

  - P(X) 为玻尔兹曼分布

$$P(\mathbf{X}=\mathbf{x}) = \frac{1}{Z}\exp\left(\frac{-E(\mathbf{x})}{T}\right)$$



一个有六个变量的玻尔兹曼机

两个基本问题：
1. 推断p(h|v)
2. 参数学习W

玻尔兹曼机的推断

- 近似采样--Gibbs采样

$$P(X_i=1|\mathbf{x}_{\setminus i}) = \sigma\left(\frac{\Delta E_i(\mathbf{x}_{\setminus i})}{T}\right)$$

- 模拟退火
  - 让系统刚开始在一个比较高的温度下运行，然后逐渐降低，直到系统在一个比较低的温度下达到热平衡。
  - 当系统温度非常高$T \to \infty$时，$p_i \to 0.5$，即每个变量状态的改变十分容易，每一种网络状态都是一样的，而从很快可以达到热平衡。
  - 当系统温度非常低$T \to 0$时，如果$\Delta E_i(x_{\setminus i}) > 0$则$p_i \to 1$，如果$\Delta E_i(x_{\setminus i}) < 0$则$p_i \to 0$。
    - 随机性方法变成确定性方法

玻尔兹曼机的参数学习

- 最大似然估计

$$\mathcal{LL}(\mathcal{W}) = \frac{1}{N}\sum_{i=1}^{N}\log p(\hat{\mathbf{v}}^{(n)})$$

- 采用梯度上升法

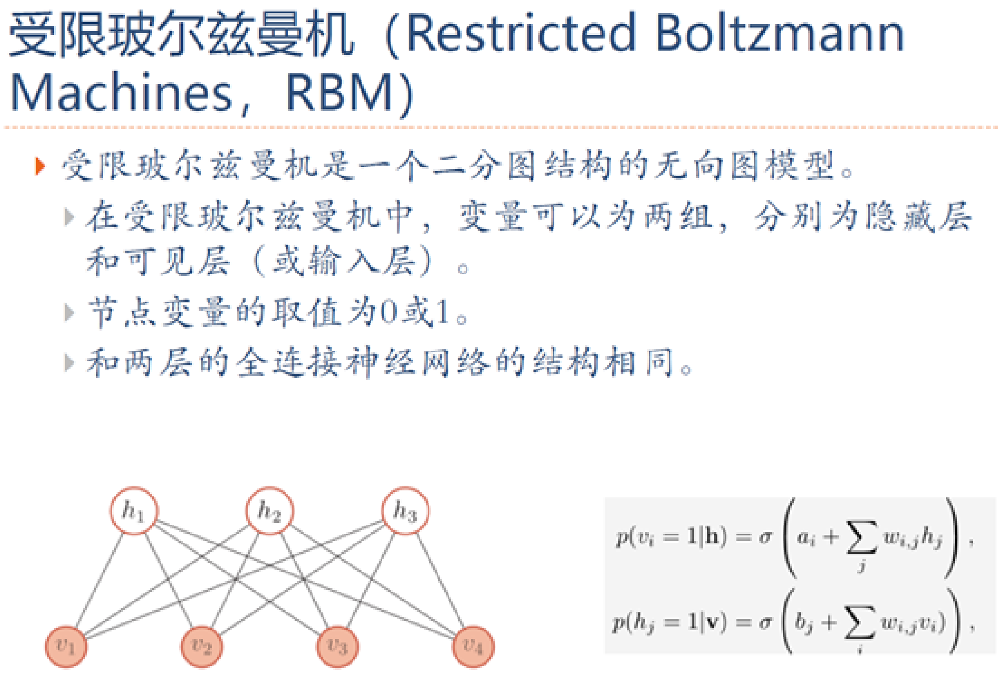
$$w_{ij} \leftarrow w_{ij} + \alpha\left(\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{p(\mathbf{h}|\hat{\mathbf{v}}^{(n)})}[x_i x_j] - \mathbb{E}_{p(\mathbf{x})}[x_i x_j]\right)$$

玻尔兹曼机的参数学习

- 基于Gibbs采样来进行近似求解

$$w_{ij} \leftarrow w_{ij} + \alpha\left(\frac{1}{N}\sum_{n=1}^{N}\underline{\mathbb{E}_{p(\mathbf{h}|\hat{\mathbf{v}}^{(n)})}[x_i x_j]} - \underline{\mathbb{E}_{p(\mathbf{x})}[x_i x_j]}\right)$$

$$w_{ij} \leftarrow w_{ij} + \alpha\left(\underline{\langle x_i x_j\rangle_{\texttt{data}}} - \underline{\langle x_i x_j\rangle_{\texttt{model}}}\right)$$



6．参考文献

[1] 邱锡鹏, 神经网络与深度学习[M]. 2019.

[2] Salakhutdinov R, Hinton G. Deep boltzmann machines[C]//Artificial intelligence and statistics. 2009: 448-455.

[3] Hinton, Training a deep autoencoder or a classifier on MNIST digits. 2006.

[4] Hinton G E. Training products of experts by minimizing contrastive divergence[J]. Neural computation, 2002, 14(8): 1771-1800.

[5] Hinton G E. A practical guide to training restricted Boltzmann machines[M]//Neural networks: Tricks of the trade. Springer, Berlin, Heidelberg, 2012: 599-619.

[6] 深度学习 --- 受限玻尔兹曼机详解(RBM)

[7] 受限玻尔兹曼机（RBM）学习笔记（六）对比散度算法

[8] Deep Learning（深度学习）学习笔记整理系列之（四）

[9] Restricted Boltzmann Machines (RBM)