第 12 章　深度信念网络

本章介绍两种特殊的概率图模型：玻尔兹曼机（包括受限玻尔兹曼机）和深度信念网络，这两种模型和神经网络有很强的对应关系，在一定程度上也称为随机神经网络（Stochastic Neural Network）。

作为一种概率图模型，玻尔兹曼机和深度信念网络的共同问题是推断和学习问题。因为这两种模型都比较复杂，并且都包含隐变量，它们的推断和学习一般通过MCMC方法来进行近似估计。

玻尔兹曼机和深度信念网络都是生成模型，使用隐变量来描述数据分布，其参数学习是一种无监督学习，不需要数据的标签信息。

12.1　玻尔兹曼机

动态系统是数学上的一个概念，用一个函数来描述一个空间中所有点随时间的变化情况，比如钟摆晃动、水的流动等。

玻尔兹曼机（Boltzmann Machine）可以看做是一个随机动力系统（Stochastic Dynamical System），每个变量的状态都以一定的概率受到其它变量的影响。玻尔兹曼机可以用概率无向图模型来描述。一个具有 K 个节点（变量）的玻尔兹曼机满足以下三个性质：

1. 每个随机变量是二值的，所有随机变量可以用一个二值的随机向量 $\mathbf{X} \in \{0,1\}^K$ 来表示，其中可观测变量表示为 $\mathbf{V}$，隐变量表示为 $\mathbf{H}$；
2. 所有节点之间是全连接的。每个变量 X_i 的取值依赖于所有其它变量 $\mathbf{X}_{\setminus i}$；
3. 每两个变量之间的相互影响（$X_i \to X_j$ 和 $X_j \to X_i$）是对称的。

图12.1给出了一个包含 3 个可观测变量和 3 个隐变量的玻尔兹曼机。

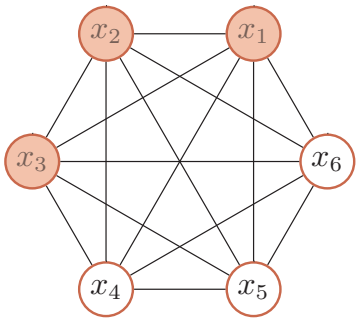


图 12.1 一个有六个变量的玻尔兹曼机

变量 $\mathbf{X}$ 的联合概率由玻尔兹曼分布得到，即

$$p(\mathbf{x}) = \frac{1}{Z} \exp\left(\frac{-E(\mathbf{x})}{T}\right), \tag{12.1}$$

这也是玻尔兹曼机名称的由来。为了简单起见，这里我们把玻尔兹曼常数 k 吸收到温度 T 中。

其中 Z 为配分函数，能量函数 $E(\mathbf{x})$ 的定义为

$$\begin{aligned} E(\mathbf{x}) &\triangleq E(\mathbf{X} = \mathbf{x}) \\ &= -\left(\sum_{i<j} w_{ij}\, x_i\, x_j + \sum_i b_i\, x_i\right), \end{aligned} \tag{12.2}$$

其中 w_{ij} 是两个变量 x_i 和 x_j 之间的连接权重，$x_i \in \{0, 1\}$ 表示状态，b_i 是变量 x_i 的偏置。

如果两个变量 X_i 和 X_j 的取值都为 1 时，一个正的权重 $w_{ij} > 0$ 会使得玻尔兹曼机的能量下降，发生的概率变大；相反，一个负的权重会使得能量上升，发生的概率变小。因此，如果令玻尔兹曼机中的每个变量 X_i 代表一个基本假设，其取值为 1 或 0 分别表示模型接受或拒绝该假设，那么变量之间连接的权重为可正可负的实数，代表了两个假设之间的弱约束关系 [Ackley et al., 1985]。一个正的权重表示两个假设可以相互支持。也就是说，如果一个假设被接受，另一个也很可能被接受。相反，一个负的权重表示两个假设不能同时被接受。

德维希·玻尔兹曼（Ludwig Boltzmann，1844 - 1906），奥地利物理学家、哲学家。主要贡献为分子动力学。

玻尔兹曼机可以用来解决两类问题。一类是搜索问题。当给定变量之间的连接权重，需要找到一组二值向量，使得整个网络的能量最低。另一类是学习问题。当给一组定部分变量的观测值时，计算一组最优的权重。

数学小知识 | 玻尔兹曼分布

在统计力学中，**玻尔兹曼分布**（Boltzmann Distribution）是描述粒子处于特定状态下的概率，是关于状态能量与系统温度的函数。玻尔兹曼分布取自奥地利物理学家路德维希·玻尔兹曼（Ludwig Boltzmann），他在 1868 年研究热平衡气体的统计力学时首次提出了这一分布。一个粒子处于为状态 α 的概率 p_α 是关于状态能量与系统温度的函数，

$$p_\alpha = \frac{1}{Z}\exp\left(\frac{-E_\alpha}{kT}\right), \tag{12.3}$$

其中 E_α 为状态 α 的能量，k 为玻尔兹曼常量，T 为系统温度，$\exp(\frac{-E_\alpha}{kT})$ 称为**玻尔兹曼因子**（Boltzmann Factor），是没有归一化的概率。Z 为归一化因子，通常称为为**配分函数**（Partition Function），是对系统所有状态进行总和，

$$Z = \sum_\alpha \exp\left(\frac{-E_\alpha}{kT}\right). \tag{12.4}$$

玻尔兹曼分布的一个性质是两个状态的概率比仅仅依赖于两个状态能量的差值。

$$\frac{p_\alpha}{p_\beta} = \exp\left(\frac{E_\beta - E_\alpha}{kT}\right). \tag{12.5}$$

12.1.1 生成模型

在玻尔兹曼机中，配分函数 Z 通常难以计算，因此，联合概率分布 $p(\mathbf{x})$ 一般通过 MCMC 方法来近似，生成一组服从 $p(\mathbf{x})$ 分布的样本。本节介绍基于吉布斯采样的样本生成方法。

吉布斯采样参见第11.3.4.3节。

12.1.1.1 全条件概率

吉布斯采样需要计算每个变量 X_i 的全条件概率 $p(x_i|\mathbf{x}_{\setminus i})$，其中 $\mathbf{x}_{\setminus i}$ 表示除变量 X_i 外其它变量的取值。

定理 12.1 – 玻尔兹曼机中变量的全条件概率： 对于玻尔兹曼机中的一个变量 X_i，当给定其它变量 $\mathbf{x}_{\setminus i}$ 时，全条件概率 $p(x_i|\mathbf{x}_{\setminus i})$ 为

$$p(x_i = 1|\mathbf{x}_{\setminus i}) = \sigma\left(\frac{\sum_j w_{ij}\, x_j + b_i}{T}\right), \tag{12.6}$$

$$p(x_i = 0|\mathbf{x}_{\setminus i}) = 1 - p(x_i = 1|\mathbf{x}_{\setminus i}), \tag{12.7}$$

其中 σ 为 logistic sigmoid 函数。

证明. 首先，保持其它变量 $\mathbf{x}_{\setminus i}$ 不变，改变变量 X_i 的状态，从 0（关闭）和 1（打开）之间的能量差异（Energy Gap）为

$$\Delta E_i(\mathbf{x}_{\setminus i}) = E(x_i = 0, \mathbf{x}_{\setminus i}) - E(x_i = 1, \mathbf{x}_{\setminus i}) \tag{12.8}$$

$$= \sum_j w_{ij}\, x_j + b_i, \tag{12.9}$$

其中 $w_{ii} = 0$。

又根据玻尔兹曼机的定义可得

$$E(\mathbf{x}) = -T \log p(\mathbf{x}) - T \log Z, \tag{12.10}$$

因此有

$$\Delta E_i(\mathbf{x}_{\setminus i}) = -T \ln p(x_i = 0, \mathbf{x}_{\setminus i}) - (-T \ln p(x_i = 1, \mathbf{x}_{\setminus i})) \tag{12.11}$$

$$= T \ln \frac{p(x_i = 1, \mathbf{x}_{\setminus i})}{p(x_i = 0, \mathbf{x}_{\setminus i})} \tag{12.12}$$

$$= T \ln \frac{p(x_i = 1|\mathbf{x}_{\setminus i})}{p(x_i = 0|\mathbf{x}_{\setminus i})} \tag{12.13}$$

$$= T \ln \frac{p(x_i = 1, |\mathbf{x}_{\setminus i})}{1 - p(x_i = 1|\mathbf{x}_{\setminus i})}, \tag{12.14}$$

结合公式 (12.14) 和 (12.14)，得到

$$p(x_i = 1|\mathbf{x}_{\setminus i}) = \frac{1}{1 + \exp\left(-\frac{\Delta E_i(\mathbf{x}_{\setminus i})}{T}\right)} \tag{12.15}$$

$$= \sigma\left(\frac{\sum_j w_{ij}\, x_j + b_i}{T}\right). \tag{12.16}$$

□

12.1.1.2 吉布斯采样

玻尔兹曼机的吉布斯采样过程为：随机选择一个变量 X_i，然后根据其全条件概率 $p(x_i|\mathbf{x}_{\setminus i})$ 来设置其状态，即以 $p(x_i = 1|\mathbf{x}_{\setminus i})$ 的概率将变量 X_i 设为1，否则为0。在固定温度 T 的情况下，在运行足够时间之后，玻尔兹曼机会达到热平衡。此时，任何全局状态的概率服从玻尔兹曼分布 $p(\mathbf{x})$，只与系统的能量有关，与初始状态无关。

当玻尔兹曼机达到热平衡时，并不意味其能量最低。热平衡依然是在所有状态上的一个分布。

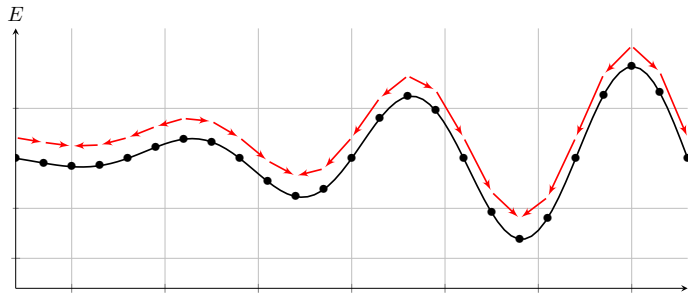
要使得玻尔兹曼机达到热平衡，其收敛速度和温度 T 相关。当系统温度非常高 $T \to \infty$ 时，$p(x_i = 1|\mathbf{x}_{\setminus i}) \to 0.5$，即每个变量状态的改变十分容易，每一种系统状态都是一样的，而从很快可以达到热平衡。当系统温度非常低 $T \to 0$ 时，如果 $\Delta E_i(\mathbf{x}_{\setminus i}) > 0$ 则 $p(x_i = 1|\mathbf{x}_{\setminus i}) \to 1$，如果 $\Delta E_i(\mathbf{x}_{\setminus i}) < 0$ 则 $p(x_i = 1|\mathbf{x}_{\setminus i}) \to 0$，即

$$x_i = \begin{cases} 1 & \text{if } \sum_j w_{ij} x_j + b_i \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{12.17}$$

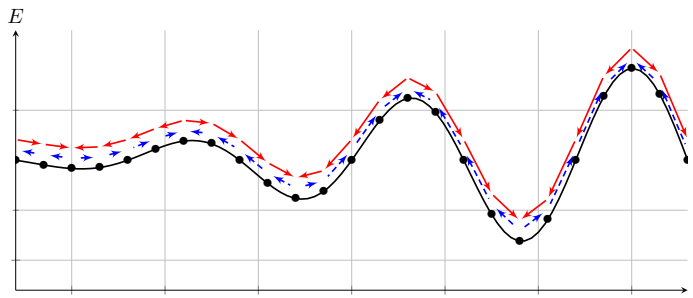
因此，当 $T \to 0$ 时，随机性方法变成了确定性方法。这时，玻尔兹曼机退化为一个 Hopfield 网络。

Hopfield 网络参见第8.2.3.1节。

Hopfield 网络是一种确定性的动态系统，而玻尔兹曼机是一种随机性的动态系统。Hopfield 网络的每次的状态更新都会使得系统的能力降低，而玻尔兹曼机则以一定的概率使得系统的能量上升。图12.2给出了 Hopfield 网络和玻尔兹曼机在运行时系统能量变化的对比。



(a) Hopfield 网络



(b) 玻尔兹曼机

图 12.2 Hopfield 网络和玻尔兹曼机运行时，系统能量变化对比

12.1.2 能量最小化与模拟退火

在一个动态系统中，找到一个状态使得系统能量最小是一个十分重要的优化问题。如果这个动态系统是确定性的，比如 Hopfield 网络，一个简单（但是低效）的能量最小化方法是随机选择一个变量，在其它变量保持不变的情况下，将这个变量设为会导致整个网络能量更低的状态。当每个变量 X_i 取值为 $\{0, 1\}$ 时，如果能量差异 $\Delta E_i(\mathbf{x}_{\setminus i})$ 大于 0，就设 $X_i = 1$，否则就设 $X_i = 0$。

特别地，离散状态的能量最小化是一个组合优化问题。

这种简单、确定性的方法在运行一定时间之后总是可以收敛到一个解。但是这个解是局部最优的，不是全局最优。为了跳出局部最优，就必须允许“偶尔”可以将一个变量设置为使得能量变高的状态。这样，我们就需要引入一定的随机性，我们以 $\sigma\left(\frac{\Delta E_i(\mathbf{x}_{\setminus i})}{T}\right)$ 的概率将变量 X_i 设为 1，否则设为 0。这个过程

局部最优在 Hopfield 网络中不是一个缺点。相反，Hopfield 网络是通过利用局部最优点来存储信息。

和玻尔兹曼机的吉布斯采样过程十分类似。

要使得动态系统达到热平衡，温度 T 的选择十分关键。一个比较好的折中方法是让系统刚开始在一个比较高的温度下运行达到热平衡，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。这样我们就能够得到一个能量全局最小的分布。这个过程被称为模拟退火（Simulated Annealing）[Kirkpatrick et al., 1983]。

模拟退火是一种寻找全局最优的近似方法，其名字来自冶金学的专有名词“退火”，即将材料加热后再以一定的速度退火冷却，可以减少晶格中的缺陷。固体中的内部粒子会停留在使内能有局部最小值的位置，加热时能量变大，粒子会变得无序并随机移动。退火冷却时速度较慢，使得粒子在每个温度都达到平衡态。最后在常温时，粒子以很大的概率达到内能比原先更低的位置。可以证明，模拟退火算法所得解依概率收敛到全局最优解。

12.1.3 参数学习

不失一般性，假设玻尔兹曼机中的变量分为可观测变量 $\mathbf{v} \in \{0,1\}^m$ 和隐变量 $\mathbf{h} \in \{0,1\}^n$。

给定一组可观测的向量 $\mathcal{D} = \{\hat{\mathbf{v}}^{(1)}, \hat{\mathbf{v}}^{(2)}, \cdots, \hat{\mathbf{v}}^{(N)}\}$ 作为训练集，我们要学习玻尔兹曼机的参数 W 和 $\mathbf{b}$ 使得训练集中所有样本的对数似然函数最大。训练集的对数似然函数定义为

$$\mathcal{L}(\mathcal{D}|W,\mathbf{b}) = \frac{1}{N}\sum_{n=1}^{N}\log p(\hat{\mathbf{v}}^{(n)}|W,b) \tag{12.18}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\log \sum_{\mathbf{h}} p(\hat{\mathbf{v}}^{(n)},\mathbf{h}|W,\mathbf{b}) \tag{12.19}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\log \frac{\sum_{\mathbf{h}}\exp\Big(-E(\hat{\mathbf{v}}^{(n)},\mathbf{h})\Big)}{\sum_{\mathbf{v},\mathbf{h}}\exp\Big(-E(\mathbf{v},\mathbf{h})\Big)}. \tag{12.20}$$

每个训练样本的对数似然 $\log p(\hat{\mathbf{v}}^{(n)})$ 对参数 w_{ij} 的偏导数为

$$\frac{\partial \log p(\hat{\mathbf{v}}^{(n)}|W,\mathbf{b})}{\partial w_{ij}} = \frac{\partial \log \sum_{\mathbf{h}} p(\hat{\mathbf{v}}^{(n)},\mathbf{h}|W,\mathbf{b})}{\partial w_{ij}} \tag{12.21}$$

$$= \frac{\partial\Big(\log\sum_{\mathbf{h}}\exp\Big(-E(\hat{\mathbf{v}}^{(n)},\mathbf{h})\Big) - \log\sum_{\mathbf{v},\mathbf{h}}\exp\Big(-E(\mathbf{v},\mathbf{h})\Big)\Big)}{\partial w_{ij}} \tag{12.22}$$

$$= \sum_{\mathbf{h}} \frac{\exp\Big(-E(\hat{\mathbf{v}}^{(n)}, \mathbf{h})\Big)}{\sum_{\mathbf{h}} \exp\Big(-E(\hat{\mathbf{v}}^{(n)}, \mathbf{h})\Big)} x_i x_j - \sum_{\mathbf{v},\mathbf{h}} \frac{\exp\Big(-E(\mathbf{v}, \mathbf{h})\Big)}{\sum_{\mathbf{v},\mathbf{h}} \exp\Big(-E(\mathbf{v}, \mathbf{h})\Big)} x_i x_j \tag{12.23}$$

$$= \sum_{\mathbf{h}} p(\mathbf{h}|\hat{\mathbf{v}}^{(n)}) x_i x_j - \sum_{\mathbf{v},\mathbf{h}} p(\mathbf{v}, \mathbf{h}) x_i x_j \tag{12.24}$$

$$= \mathbb{E}_{p(\mathbf{h}|\hat{\mathbf{v}}^{(n)})}\Big[x_i x_j\Big] - \mathbb{E}_{p(\mathbf{x})}\Big[x_i x_j\Big], \tag{12.25}$$

其中 $p(\mathbf{h}|\hat{\mathbf{v}}^{(n)})$ 和 $p(\mathbf{x}) = p(\mathbf{v}, \mathbf{h})$ 在当前参数 $W, \mathbf{b}$ 条件下计算的条件概率和联合概率。

采用梯度上升法，参数 w_{ij} 的更新公式为：

$$w_{ij} \leftarrow w_{ij} + \alpha \left(\frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{p(\mathbf{h}|\hat{\mathbf{v}}^{(n)})}[x_i x_j] - \mathbb{E}_{p(\mathbf{x})}[x_i x_j] \right), \tag{12.26}$$

其中 $\alpha > 0$ 为学习率。但因为涉及到计算配分函数，这个梯度很难精确计算。对于一个 D 维的随机向量 $\mathbf{X}$，其取值空间大小为 2^D。当 D 比较大时，这两项的计算会十分耗时。因此，只能通过一些采样方法（如吉布斯采样）来进行近似求解。

公式 (12.25) 中第一项为 $x_i x_j$ 在给定 $\mathbf{V} = \hat{\mathbf{v}}^{(n)}$ 时的期望。我们可以将可观测变量固定，只对 $\mathbf{h}$ 进行吉布斯采样，使得网络达到热平衡状态。对于每一个连接，采样 $x_i x_j$ 的值。在训练集上所有的训练样本上重复此过程，得到 $x_i x_j$ 的近似期望 $\langle x_i x_j \rangle_{\texttt{data}}$。公式 (12.25) 中的第二项为没有任何限制时的期望。我们可以对所有变量进行吉布斯采样，使得网络达到热平衡状态。对于每一个连接，采样 $x_i x_j$ 的值，得到 $x_i x_j$ 的近似期望 $\langle x_i x_j \rangle_{\texttt{model}}$。

这样，权重 w_{ij} 可以用下面公式近似地更新

$$w_{ij} \leftarrow w_{ij} + \alpha \left(\langle x_i x_j \rangle_{\texttt{data}} - \langle x_i x_j \rangle_{\texttt{model}} \right). \tag{12.27}$$

这个更新方法的一个特点是仅仅使用了局部信息。也就是说，虽然我们优化目标是整个网络的能量最低，但是每个权重的更新只依赖于它连接的相关变量的状态。这种学习方式和人脑神经网络的学习方式（赫布规则）十分类似。

玻尔兹曼机可以用在监督学习和无监督学习中。在监督学习中，可观测的变量 $\mathbf{v}$ 又进一步可以分为输入和输出变量，隐变量则隐式地描述了输入和输出变量之间复杂的约束关系。在无监督学习中，隐变量可以看做是可观测变量的内部特征表示。玻尔兹曼机也可以看做是一种随机型的神经网络，是 Hopfield

神经网络的扩展，并且可以生成的相应的 Hopfield 神经网络。在没有时间限制时，玻尔兹曼机还可以用来解决复杂的组合优化问题。

12.2 受限玻尔兹曼机

全连接的玻尔兹曼机虽然只在理论上十分有趣，但是由于其复杂性，目前为止并没有被广泛使用。虽然基于采样的方法在很大程度提高了学习效率，但是每更新一次权重，就需要网络重新达到热平衡状态，这个过程依然比较低效，需要很长时间。在实际应用中，使用比较广泛的一种带限制的版本，也就是受限玻尔兹曼机。

受限玻尔兹曼机因其结构最初称为**簧风琴模型**，2000 年后受限玻兹曼机的名称才变得流行。

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。我们分别用可见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。

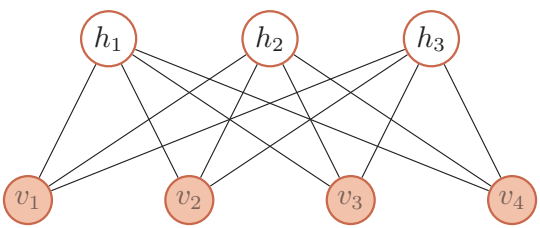


图 12.3 一个有 7 个变量的受限玻尔兹曼机

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

$$E(\mathbf{v},\mathbf{h}) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{ij} h_j \tag{12.28}$$

$$= -\mathbf{a}^{\mathrm{T}}\mathbf{v} - \mathbf{b}^{\mathrm{T}}\mathbf{h} - \mathbf{v}^{\mathrm{T}} W \mathbf{h}, \tag{12.29}$$

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

$$p(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h})) \tag{12.30}$$

$$= \frac{1}{Z}\exp(\mathbf{a}^{\mathrm{T}}\mathbf{v})\exp(\mathbf{b}^{\mathrm{T}}\mathbf{h})\exp(\mathbf{v}^{\mathrm{T}} W \mathbf{h}), \tag{12.31}$$

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

12.2.1 生成模型

受限玻尔兹曼机的联合概率分布 $p(\mathbf{h},\mathbf{v})$ 一般也通过吉布斯采样的方法来近似，生成一组服从 $p(\mathbf{h},\mathbf{v})$ 分布的样本。

吉布斯采样参见第11.3.4.3节。

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，隐变量之间相互条件独立。同样在给定隐变量时，可观测变量之间也相互条件独立。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）我们先证明 $p(v_i=1|\mathbf{h})$。

可见层变量 $\mathbf{v}$ 的边际概率为

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

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

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

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

利用分配律

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

将 h_j 为 0 或 1 的取值代入计算

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

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

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

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

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

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。

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

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

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，而从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。

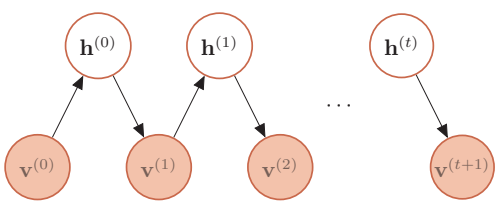


图 12.4 受限玻尔兹曼机的采样过程

12.2.2 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

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

若采用随机梯度上升法，对于每个样本$\hat{\mathbf{v}}^{(n)}$，其对数似然函数关于参数w_{ij}的偏导数为

$$\frac{\partial \log p(\hat{\mathbf{v}}^{(n)}|W, \mathbf{a}, \mathbf{b})}{\partial w_{ij}} = \frac{\partial \log \sum_{\mathbf{h}} p(\hat{\mathbf{v}}^{(n)}, \mathbf{h}|W, \mathbf{a}, \mathbf{b})}{\partial w_{ij}} \tag{12.52}$$

$$= \frac{\partial\Big(\log \sum_{\mathbf{h}} \exp\Big(-E(\hat{\mathbf{v}}^{(n)}, \mathbf{h})\Big) - \log \sum_{\mathbf{v},\mathbf{h}} \exp\Big(-E(\mathbf{v}, \mathbf{h})\Big)\Big)}{\partial w_{ij}} \tag{12.53}$$

$$= \sum_{h_j} p(h_j|\hat{\mathbf{v}}^{(n)})\hat{v}_i^{(n)}h_j - \sum_{\mathbf{h},\mathbf{v}} p(\mathbf{v}, \mathbf{h})v_i h_j \tag{12.54}$$

$$= \sum_{h_j} p(h_j|\hat{\mathbf{v}}^{(n)})\hat{v}_i^{(n)}h_j - \sum_{\mathbf{v}} p(\mathbf{v}) \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v})v_i h_j \tag{12.55}$$

$$= \sum_{h_j} p(h_j|\hat{\mathbf{v}}^{(n)})\hat{v}_i^{(n)}h_j - \sum_{\mathbf{v}} p(\mathbf{v}) \sum_{h_j} p(h_j|\mathbf{v})v_i h_j, \tag{12.56}$$

将$h_j = 0$或1的值代入上式，得到

$$\frac{\partial \log p(\hat{\mathbf{v}}^{(n)}|W, \mathbf{a}, \mathbf{b})}{\partial w_{ij}} = p(h_j = 1|\hat{\mathbf{v}}^{(n)})\hat{v}_i^{(n)} - \sum_{\mathbf{v}} p(\mathbf{v})p(h_j = 1|\mathbf{v})v_i, \tag{12.57}$$

同理，

$$\frac{\partial \log p(\hat{\mathbf{v}}^{(n)}|W, \mathbf{a}, \mathbf{b})}{\partial a_i} = \hat{v}_i^{(n)} - \sum_{\mathbf{v}} p(\mathbf{v})v_i, \tag{12.58}$$

$$\frac{\partial \log p(\hat{\mathbf{v}}^{(n)}|W, \mathbf{a}, \mathbf{b})}{\partial b_j} = p(h_j = 1|\hat{\mathbf{v}}^{(n)}) - \sum_{\mathbf{v}} p(\mathbf{v})p(h_j = 1|\mathbf{v}), \tag{12.59}$$

公式(12.56)、(12.58)和(12.59)中都需要计算边际分布$p(\mathbf{v})$。因为配分函数Z很难计算，因此也需要通过MCMC方法来近似计算。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。

12.2.2.1 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）[Hinton, 2002]。

与吉布斯采样不同，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变量的初始值。然后，轮流进行吉布斯采样，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。

因为目标是 $p(v) \approx p_{train}(v)$

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 $p(\mathbf{h}=\underline{1}|\hat{\mathbf{v}}^{(n)})$，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 $p(\mathbf{v}=\underline{1}|\mathbf{h})$，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 $p(\mathbf{h}=\underline{1}|\mathbf{v}')$ 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: $W, \mathbf{a}, \mathbf{b}$

12.2.3 受限玻尔兹曼机的类型

在具体的不同任务中，需要处理的数据类型不一定都是二值的，也可能是连续值。为了能够处理这些数据，就需要根据输入或输出的数据类型来设计新的能量函数。

一般来说，常见的受限玻尔兹曼机有以下三种：

“伯努利-伯努利”受限玻尔兹曼机 “伯努利-伯努利”受限玻尔兹曼机（Bernoulli-Bernoulli RBM, BB-RBM）就是上面介绍的可观测变量和隐变量都为二值类型的受限玻尔兹曼机。

“高斯-伯努利”受限玻尔兹曼机 “高斯-伯努利”受限玻尔兹曼机（Gaussian-Bernoulli RBM, GB-RBM）是假设可观测变量为高斯分布，隐变量为伯努利分布，其能量函数定义为

$$E(\mathbf{v},\mathbf{h}) = -\sum_i \frac{(v_i-\mu_i)^2}{2\sigma_i^2} - \sum_j b_j h_j - \sum_i \sum_j \frac{v_i}{\sigma_i} w_{ij} h_j, \tag{12.60}$$

参见习题12-5。

其中每个可观测变量 v_i 服从 (μ_i, σ_i) 的高斯分布。

“伯努利-高斯”受限玻尔兹曼机 “伯努利-高斯”受限玻尔兹曼机（Bernoulli-Gaussian RBM, BG-RBM）是假设可观测变量为伯努利分布，隐变量为高斯分布，其能量函数定义为

$$E(\mathbf{v},\mathbf{h}) = -\sum_i a_i v_i - \sum_j \frac{(h_j-\mu_j)^2}{2\sigma_j^2} - \sum_i \sum_j v_i w_{ij} \frac{h_j}{\sigma_j}, \tag{12.61}$$

其中每个隐变量 h_j 服从 (μ_j, σ_j) 的高斯分布。

12.3 深度信念网络

和全连接的神经网络结构相同。

深度信念网络（Deep Belief Network，DBN）是深度的概率有向图模型，其图结构由多层的节点构成。每层节点的内部没有连接，相邻两层的节点之间为全连接。网络的最底层为可见变量，其它层节点都为隐变量。最顶部的两层间的连接是无向的，其他层之间有连接上下的有向连接。图12.5给出了一个深度信念网络的示例。对一个有 L 层隐变量的深度信念网络，最底层（第 0 层）为可观测的变量 $\mathbf{v} = \mathbf{h}^{(0)}$，其余每层变量为 $\mathbf{h}^{(1)}, \cdots, \mathbf{h}^{(L)}$。除了最顶上两层外，每一层变量 $\mathbf{h}^{(l)}$ 依赖于其上面一层 $\mathbf{h}^{(l+1)}$，即

$$p(\mathbf{h}^{(l)}|\mathbf{h}^{(l+1)}, \cdots, \mathbf{h}^{(L)}) = p(\mathbf{h}^{(l)}|\mathbf{h}^{(l+1)}), \tag{12.62}$$

其中，$l = \{0, \cdots, L-2\}$。

顶部的两层是一个无向图，可以看做是一个受限玻尔兹曼机，用来产生 $p(\mathbf{h}^{(L-1)})$ 的先验分布。

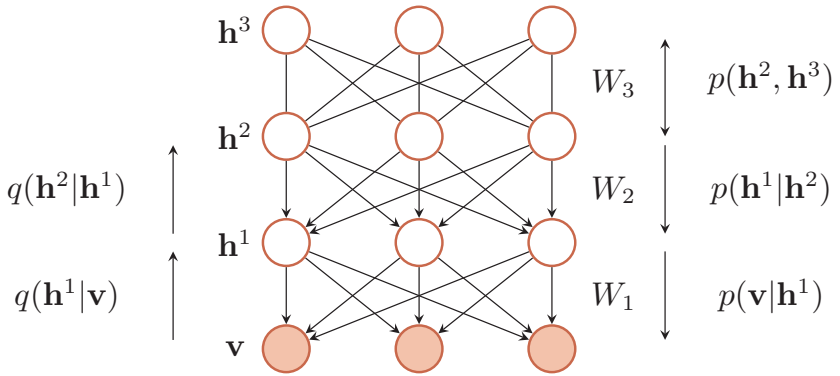


图 12.5 一个有 4 层结构的深度信念网络。

深度信念网络中所有变量的联合概率可以分解为

$$p(\mathbf{v}, \mathbf{h}^{(1)}, \cdots, \mathbf{h}^{(L)}) = p(\mathbf{v}|\mathbf{h}^{(1)}) \left(\prod_{l=1}^{L-2} p(\mathbf{h}^{(l)}|\mathbf{h}^{(l+1)}) \right) p(\mathbf{h}^{(L-1)}, \mathbf{h}^{(L)}) \quad (12.63)$$

$$= \left(\prod_{l=0}^{L-2} p(\mathbf{h}^{(l)}|\mathbf{h}^{(l+1)}) \right) p(\mathbf{h}^{(L-1)}, \mathbf{h}^{(L)}), \quad (12.64)$$

其中，$p(\mathbf{h}^{(l)}|\mathbf{h}^{(l+1)})$ 为 sigmoid 型条件概率分布为

$$p(\mathbf{h}^{(l)}|\mathbf{h}^{(l+1)}) = \sigma\left(\mathbf{b}^{(l+1)} + W^{(l+1)}\mathbf{h}^{(l+1)}\right), \quad (12.65)$$

其中，$\sigma(\cdot)$ 为按位计算的 logistic sigmoid 函数，$\mathbf{b}^{(l+1)}$ 为偏置参数，$W^{(l+1)}$ 为权重参数。

生成模型 深度信念网络是一个生成模型，可以用来生成符合特定分布的样本。隐变量用来描述在可观测变量之间的高阶相关性。假如训练数据服从分布 $p(\mathbf{v})$，通过训练得到一个深度信念网络。在生成样本时，首先在最顶两层进行足够多次的吉布斯采样，生成 $\mathbf{h}^{(L-1)}$，然后依次计算下一层隐变量的分布。因为在给定上一层变量取值时，下一层的变量是条件独立的，因为可以独立采样。这样，我们可以从第 $L-1$ 层开始，自顶向下进行逐层采样，最终得到可见层的样本。

12.3.1 深度信念网络的训练

深度信念网络最直接的训练方式可以通过最大似然方法使得可见层变量 $\mathbf{v}$ 的分布 $p(\mathbf{v})$ 在训练集合上的似然达到最大。但在深度信念网络中，隐变量 $\mathbf{h}$ 之

间的关系十分复杂，由于“贡献度分配问题”，很难直接学习。即使对于简单的单层信念网络 $p(v=1|\mathbf{h})=\sigma(b+\mathbf{w}^{\mathrm{T}}\mathbf{h})$，在已知可观测变量时，其隐变量的联合后验概率 $p(\mathbf{h}|v)$ 不再相互独立，因此很难精确估计所有隐变量的后验概率。早期深度信念网络的后验概率一般通过蒙特卡罗方法或变分方法来近似估计，但是效率比较低，而导致其参数学习比较困难。

为了有效地训练深度信念网络，我们将每一层的 sigmoid 信念网络转换为受限玻尔兹曼机。这样做的好处是隐变量的后验概率是相互独立的，从而可以很容易地进行采样。这样，深度信念网络可以看作是由多个受限玻尔兹曼机从下到上进行堆叠，每一层受限玻尔兹曼机的隐层作为上一层受限玻尔兹曼机的可见层。进一步地，深度信念网络可以采用逐层训练的方式来快速训练，即从最底层开始，每次只训练一层，直到最后一层 [Hinton et al., 2006]。

“逐层训练”是能够有效训练深度模型的最早的方法。

深度信念网络的训练过程可以分为**预训练**和**精调**两个阶段。先通过逐层预训练将模型的参数初始化为较优的值，再通过传统学习方法对参数进行精调。

预训练 在预训练阶段，我们采用逐层训练的方式，将深度信念网络的训练简化为对多个受限玻尔兹曼机的训练。

训练过程：1. 首先充分训练第一个 RBM；2. 固定第一个 RBM 的权重和偏移量，然后使用其隐性神经元的状态，作为第二个 RBM 的输入向量；3. 充分训练第二个 RBM 后，将第二个 RBM 堆叠在第一个 RBM 的上方；4. 重复以上三个步骤任意多次；5. 如果训练集中的数据有标签，那么在顶层的 RBM 训练时，这个 RBM 的显层中除了显性神经元，还需要有代表分类标签的神经元，一起进行训练：a) 假设顶层 RBM 的显层有 500 个显性神经元，训练数据的分类一共分成了 10 类；b) 那么顶层 RBM 的显层有 510 个显性神经元，对每一训练训练数据，相应的标签神经元被打开设为 1，而其他的则被关闭设为 0。6. DBN 被训练好后如下图：(示意)

算法12.2给出一种深度信念网络的逐层预训练方法。大量的实践表明，预训练可以产生非常好的参数初始值，从而极大地降低了模型的学习难度。

精调 经过预训练之后，再结合具体的任务（监督学习或重构），通过传统的全局学习算法对网络进行精调（Fine-Tuning），使模型收敛到更好的局部最优点。

除了顶层的受限玻尔兹曼机，其它层之间的权重被分成向上的*认知权重*（recognition weights）R 和向下的*生成权重*（generative weights）W。认知权重用来进行后验概率计算，而生成权重用来进行定义模型。认知权重的初始值

算法 12.2: 深度信念网络的逐层训练方法

输入: 训练集: $\hat{\mathbf{v}}^{(n)}, n = 1, \cdots, N$;
学习率：α;
深度信念网络层数:L;
第 l 层权重：$W^{(l)}$;
第 l 层偏置 $\mathbf{a}^{(l)}$;
第 l 层偏置 $\mathbf{b}^{(l)}$;

for $l = 1 \cdots L$ **do**
  初始化：$W^{(l)} \leftarrow 0,\ \mathbf{a}^{(l)} \leftarrow 0,\ \mathbf{b}^{(l)} \leftarrow 0$;
  从训练集中采样 $\mathbf{h}^{(0)} = \hat{\mathbf{v}}$;
  for $i = 1 \cdots l-1$ **do**
    根据分布 $q(\mathbf{h}^{(i)}|\mathbf{h}^{(i-1)})$ 采样 $\mathbf{h}^{(i)}$;
  end
  将 $\mathbf{h}^{(l-1)}$ 作为训练样本，充分训练第 l 个受限玻尔兹曼机 $W^{(l)}, \mathbf{a}^{(l)}, \mathbf{b}^{(l)}$;
end

输出: $W_1, \cdots, W_L$

$R^{(l)} = {W^{(l)}}^{\mathrm{T}}$。

salakhutdinov2015learning

DBN 在训练模型的过程中主要分为两步: 第 1 步：分别单独无监督地训练每一层 RBM 网络, 确保特征向量映射到不同特征空间时, 都尽可能多地保留特征信息; 第 2 步：在 DBN 的最后一层设置 BP 网络, 接收 RBM 的输出特征向量作为它的输入特征向量, 有监督地训练实体关系分类器. 而且每一层 RBM 网络只能确保自身层内的权值对该层特征向量映射达到最优, 并不是对整个 DBN 的特征向量映射达到最优, 所以反向传播网络还将错误信息自顶向下传播至每一层 RBM, 微调整个 DBN 网络.RBM 网络训练模型的过程可以看作对一个深层 BP 网络权值参数的初始化, 使DBN 克服了 BP 网络因随机初始化权值参数而容易陷入局部最优和训练时间长的缺点. 上述训练模型中第一步在深度学习的术语叫做预训练，第二步叫做微调。最上面有监督学习的那一层，根据具体的应用领域可以换成任何分类器模型，而不必是BP网络。

深度信念网络一般采用contrastive wake-sleep算法进行精调，其算法过程是：

- 随机初始化权重；
- Wake 阶段：认知过程，通过外界输入（可观测变量）和向上认知权重，计算每一层隐变量的后验概率并采样。然后，修改下行的生成权重使得下一层的变量的后验概率最大。也就是“如果现实跟我想象的不一样，改变我的权重使得我想象的东西就是这样的”；
- Sleep 阶段：生成过程，通过顶层的采样和向下的生成权重，逐层计算每一层的后验概率并采样。然后，修改向上的认知权重使得上一层变量的后验概率最大。也就是“如果梦中的景象不是我脑中的相应概念，改变我的认知权重使得这种景象在我看来就是这个概念”；
- 交替进行 Wake 和 Sleep 过程，直到收敛。

12.3.2 作为深度神经网络的预训练

只需要向上的认知权重。

深度信念网络的一个应用是作为深度神经网络的预训练部分，提供神经网络的初始权重。在深度信念网络的最顶层再增加一层输出层，然后再使用反向传播算法对这些权重进行调优。

特别是在训练数据比较少时，预训练的作用非常大。因为不恰当的初始化权重会显著影响最终模型的性能，而预训练获得的权重在权值空间中比随机权重更接近最优的权重，避免了反向传播算法因随机初始化权值参数而容易陷入局部最优和训练时间长的缺点。这不仅提升了模型的性能，也加快了调优阶段的收敛速度 [Larochelle et al., 2007]。

除了深度信念网络之外，自编码器 [Bengio et al., 2007] 以及它的变体，比如稀疏自编码器 [Ranzato et al., 2006] 和去噪自编码器 [Vincent et al., 2008]，也可以用来作为深度神经网络的初始化。

12.3.3 卷积深度信念网络

12.4 图模型与神经网络的关系

图模型和神经网络有着类似的网络结构，但两者也有很大的不同。图模型的节点是随机变量，其图结构的主要功能是用来描述变量之间的依赖关系，一般是稀疏连接。使用图模型的好处是可以有效进行统计推断。而神经网络中的节点是神经元，是一个计算节点。如果将神经网络中每个神经元看做是一个二值随机变量，那神经网络就变成一个 sigmoid 信念网络。

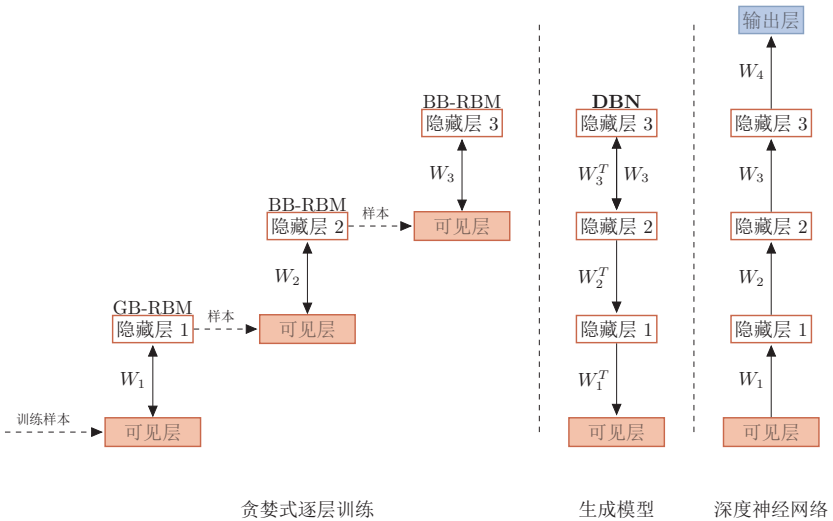


图 12.6 一个有 4 层结构的深度信念网络。

图模型中的每个变量一般有着明确的解释，变量之间依赖关系一般是人工来定义。而神经网络中的神经元则没有直观的解释。

图模型一般是生成模型，可以用生成样本，也可以通过贝叶斯公式用来做分类。而神经网络是判别模型，直接用来分类。

判别模型也可以用图模型来表示。

图模型的参数学习的目标函数为似然函数或条件似然函数，若包含隐变量则通常通过 EM 算法来求解。而神经网络参数学习的目标为交叉熵或平方误差等损失函数。

12.5 总结和深入阅读

12.6 习题

习题 12-1 如果使用 Metropolis 算法对玻尔兹曼机进行采样，给出其提议分布的具体形式。

Metropolis 算法参见第11.3.4.2节。

习题 12-2 在玻尔兹曼机中，求公式 (12.20) 中对数似然函数关于参数 $\mathbf{b}$ 的偏导数。

习题 12-3 在受限玻尔兹曼机中，证明公式 (12.48)。

习题 12-4 在受限玻尔兹曼机中，求公式 (12.20) 中对数似然函数关于参数 $\mathbf{b}$ 的偏导数。

习题 12-5 计算“高斯-伯努利”受限玻尔兹曼机和“伯努利-高斯”受限玻尔兹曼机的条件概率 $p(\mathbf{v}=\mathbf{1}|\mathbf{h})$ 和 $p(\mathbf{h}=\mathbf{1}|\mathbf{v})$。

参考文献

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchi, et al. Optimization by simulated annealing. *science*, 220(4598): 671–680, 1983.

Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pages 473–480. ACM, 2007.

Marc'Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann LeCun. Efficient learning of sparse representations with an energy-based model. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pages 1137–1144. MIT Press, 2006.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the International Conference on Machine Learning*, pages 1096–1103. ACM, 2008.