Generative Models: Fundamentals and Applications

Lecture 5: Energy-based models

Shuigeng Zhou, Wenhui Liu
College of CSAI

October 20, 2026

目录

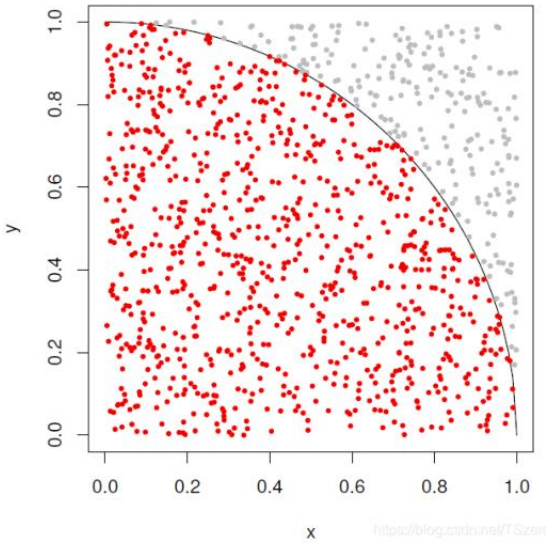
- 马尔可夫链蒙特卡罗法（MCMC）
- 能量模型
- 受限玻尔兹曼机（RBM）
- 深度信念网络（DBN）
- 深度玻尔兹曼机（DBM）

回顾：蒙特卡罗法(Monte Carlo)

- 思想：用独立同分布随机采样近似积分或期望

$$\mathbb{E}_{p(x)}[f(x)] = \int f(x)p(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) \qquad x_i \sim p(x)$$

  - 例如：蒲丰投针
- 前提：能从 $p(x)$ 随机采样
  - 很多时候无法直接计算概率密度，例如：
    - 贝叶斯推断 $p(x) = \int p(x|\theta)p(\theta)d\theta$
    - 玻尔兹曼机 $p(x) = \frac{1}{Z}e^{-E(x)}$



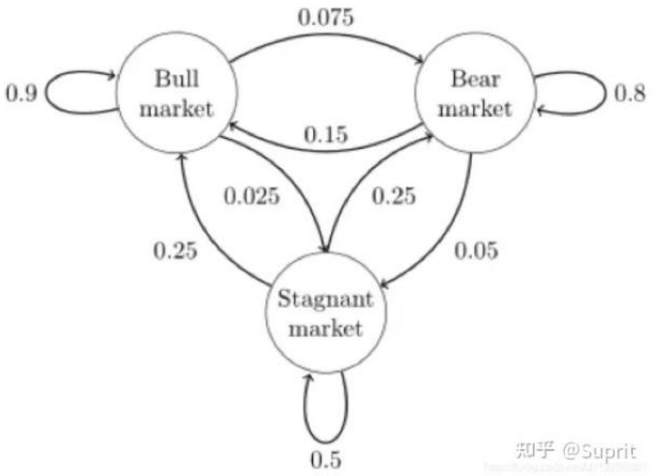
马尔可夫链蒙特卡罗法

- 马尔可夫链蒙特卡罗法 (Markov Chain Monte Carlo，MCMC)
  - 以马尔可夫链为概率模型的随机抽样近似计算方法
  - 设：
    - 多元随机变量 $x \in \mathcal{X}$，其概率密度函数为 $p(x)$
    - $f(x)$ 为定义在 $x \in \mathcal{X}$ 上的函数
  - 目标：
    - 获得概率分布 $p(x)$ 的样本集合
    - 求 $f(x)$ 的数学期望 $\mathbb{E}_{p(x)} f(x)$
  - 适用于多元随机变量、非标准形式密度函数、随机变量各分量不独立等概率密度函数 $p(x)$ 复杂的情况，无法直接采样

马尔可夫链蒙特卡罗法

- 基本思想
  - 在随机变量x的状态空间 S 上定义一个满足遍历定理的马尔可夫链 $x_1, x_2, \ldots, x_t, \ldots$
  - 在该马尔可夫链上随机游走，每个时刻得到一个样本
  - 根据遍历定理，当时间趋于无穷时，样本的分布趋近平稳分布，样本的函数均值趋近函数的数学期望

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



马尔可夫链蒙特卡罗法

- 基本思想
  - 当链的运行时间足够长（大于充分大整数 m）时

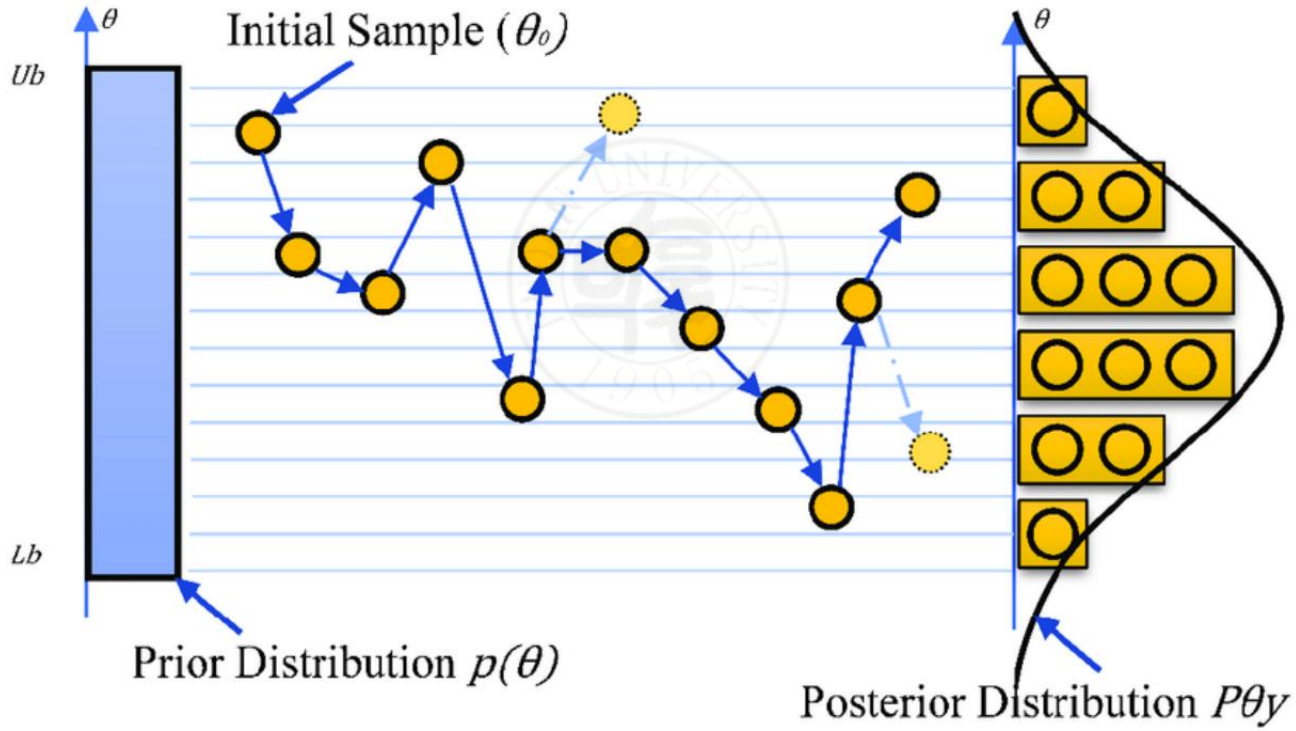
$$x_{m+1}, x_{m+2}, \cdots, x_n \sim p(x)$$

$$\hat{E}f = \frac{1}{n-m} \sum_{i=m+1}^{n} f(x_i)$$

  - 到时刻m为止的时间段称为燃烧期
  - 随机游走的起始点并不影响得到的结果：
    - 从不同的起始点出发，都会收敛到同一平稳分布

马尔可夫链蒙特卡罗法

- 可视化例子



马尔可夫链蒙特卡罗法

- 构造马尔可夫链
  - 定义状态转移概率
    - 如果x是连续变量：转移核函数p(x, x')
    - 如果x是离散变量：转移矩阵P
  - 目标：使马尔可夫链的平稳分布为目标分布$p(x)$
    - 最基本的MCMC：Metropolis-Hastings (MH) 算法
    - 更简单、使用更广泛的MCMC：Gibbs 抽样

Metropolis-Hastings算法

- 基本思想：逐步探索目标分布
  - 在时刻 $t-1$，假设处于状态 x
    - 用建议分布 (proposal distribution) $q(x,x') \triangleq q(x'|x)$ 抽样一个候选状态 x'
    - 用接受概率 (acceptance probability) $\alpha(x,x')$ 评估候选状态 x'
  - 若候选状态 x'
    - 比当前状态更接近目标分布：在时刻 t 转移到 x'
    - 否则：在时刻 t 停留在原状态 x

Metropolis-Hastings算法

- 转移核函数 $p(x, x')$

$$p(x, x') = q(x, x')\alpha(x, x')$$

  - 建议分布 $q(x, x')$
    - 对称分布：如多元正态分布 $\mathcal{N}(x'|x, \Sigma)$
    - 独立抽样：假设 $q(x, x')$ 与当前状态 x 无关：$q(x, x') = q(x')$
  - 接受概率 $\alpha(x, x')$

$$\alpha(x, x') = \min\left\{1, \frac{p(x')q(x', x)}{p(x)q(x, x')}\right\}$$

公式里 p 只以比值 p(x')/p(x) 出现，所以归一化常数 Z 完全不需要知道

  - 满足细致平衡条件：$p(x)p(x, x') = p(x')p(x', x)$

细致平衡是"p 是这条链平稳分布"的充分条件

Metropolis-Hastings算法

算法 19.2（Metropolis-Hastings 算法）

输入：抽样的目标分布的密度函数 $p(x)$，函数 $f(x)$；
输出：$p(x)$ 的随机样本 $x_{m+1}, x_{m+2}, \cdots, x_n$，函数样本均值 f_{mn}；

参数：收敛步数 m，迭代步数 n。

（1）任意选择一个初始值 x_0

（2）对 $i = 1, 2, \cdots, n$ 循环执行

（a）设状态 $x_{i-1} = x$，按照建议分布 $q(x, x')$ 随机抽取一个候选状态 x'

（b）计算接受概率

$$\alpha(x, x') = \min\left\{1, \frac{p(x')q(x', x)}{p(x)q(x, x')}\right\}$$

(c)从区间(0,1)中按照均匀分布随机抽取一个数u。
若u< $\alpha(x, x')$，则状态$x_i = x'$；否则，状态$x_i = x$。

（3）得到样本集合 $\{x_{m+1}, x_{m+2}, \cdots, x_n\}$

计算

$$f_{mn} = \frac{1}{n-m}\sum_{i=m+1}^{n} f(x_i)$$

直觉

当提议对称 $q(x^*|x) = q(x|x^*)$，比值直接简化成密度之比：

$$r = \frac{\pi(x^*)}{\pi(x)}$$

- 如果候选点密度更高：$r > 1, \alpha = 1 \rightarrow$ **无条件接受，放心往高概率区域走**；
- 如果候选点密度更低：$r < 1, \alpha = r \rightarrow$ **有概率放行。**

Gibbs抽样

- MH算法的难点：选择合理的建议分布 $q(x, x')$
- Gibbs抽样：MH算法的「控制变量」版本
  - 可行性假设：能写出联合分布 $p(x_1, \ldots, x_d)$ 的条件分布 $p(x_i|x_{-i})$
  - 一次只更新一个维度

$$q(\mathbf{x}'|\mathbf{x}) = p(x'_i|\mathbf{x}_{-i})\mathbb{I}(\mathbf{x}'_{-i} = \mathbf{x}_{-i})$$

  - 该更新采样自真实条件概率：Gibbs抽样接受每次抽样结果

Gibbs抽样

算法 19.3 (吉布斯抽样)

输入：目标概率分布的密度函数 $p(x)$，函数 $f(x)$；

输出：$p(x)$ 的随机样本 $x_{m+1}, x_{m+2}, \cdots, x_n$，函数样本均值 f_{mn}；

参数：收敛步数 m，迭代步数 n。

(1) 初始化。给出初始样本 $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \cdots, x_k^{(0)})^{\mathrm{T}}$。

(2) 对 i 循环执行

设第 $(i-1)$ 次迭代结束时的样本为 $x^{(i-1)} = (x_1^{(i-1)}, x_2^{(i-1)}, \cdots, x_k^{(i-1)})^{\mathrm{T}}$，则第 i 次迭代进行如下几步操作：

(1) 由满条件分布 $p(x_1|x_2^{(i-1)}, \cdots, x_k^{(i-1)})$ 抽取 $x_1^{(i)}$

$\vdots$

(j) 由满条件分布 $p(x_j|x_1^{(i)}, \cdots, x_{j-1}^{(i)}, x_{j+1}^{(i-1)}, \cdots, x_k^{(i-1)})$ 抽取 $x_j^{(i)}$

$\vdots$

(k) 由满条件分布 $p(x_k|x_1^{(i)}, \cdots, x_{k-1}^{(i)})$ 抽取 $x_k^{(i)}$

得到第 i 次迭代值 $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \cdots, x_k^{(i)})^{\mathrm{T}}$。

(3) 得到样本集合

$$\{x^{(m+1)}, x^{(m+2)}, \cdots, x^{(n)}\}$$

(4) 计算

$$f_{mn} = \frac{1}{n-m} \sum_{i=m+1}^{n} f(x^{(i)})$$

目录

- 马尔可夫链蒙特卡罗法（MCMC）
- 能量模型
- 受限玻尔兹曼机（RBM）
- 深度信念网络（DBN）
- 深度玻尔兹曼机（DBM）

能量模型



「地形」：能量模型

「方向」：MCMC

Credit: Dall-E 3

能量函数

- 最早来源于热力学：系统倾向于能量最低的稳定状态
- 在机器学习中：
  - 用能量函数 $E(x)$ 来衡量样本状态的好坏：
    - 能量低 → 样本更可能出现
    - 能量高 → 样本较少出现
  - 通过能量，定义分布：

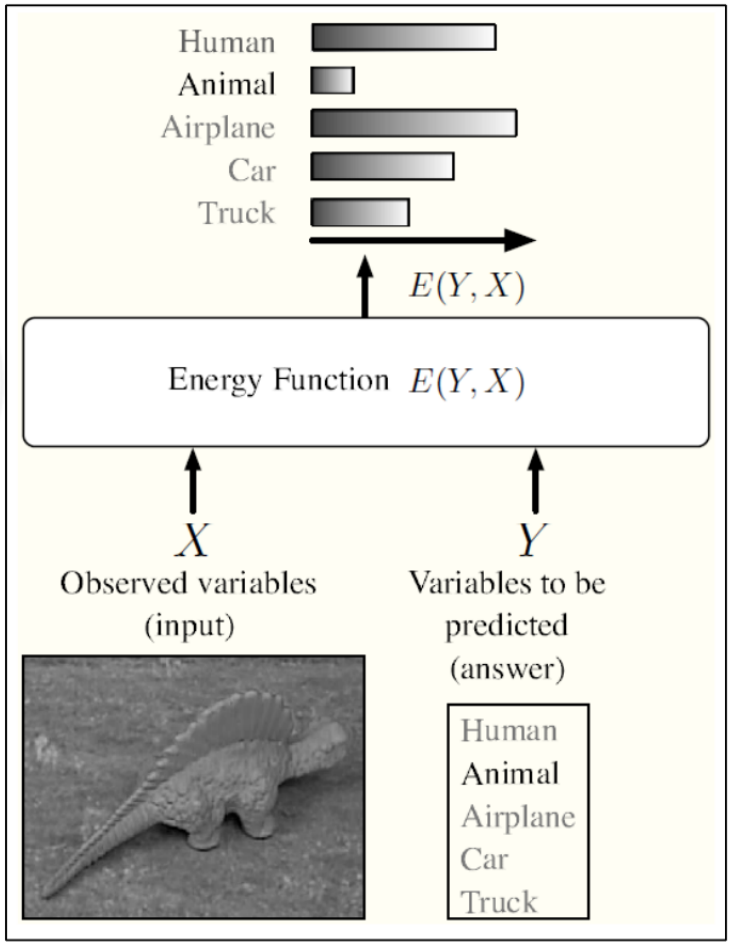
$$p(x) = \frac{1}{Z} e^{-E(x)}$$

    - 系统的「地形」由能量决定，能量谷底对应最稳定、最可能的状态

能量函数

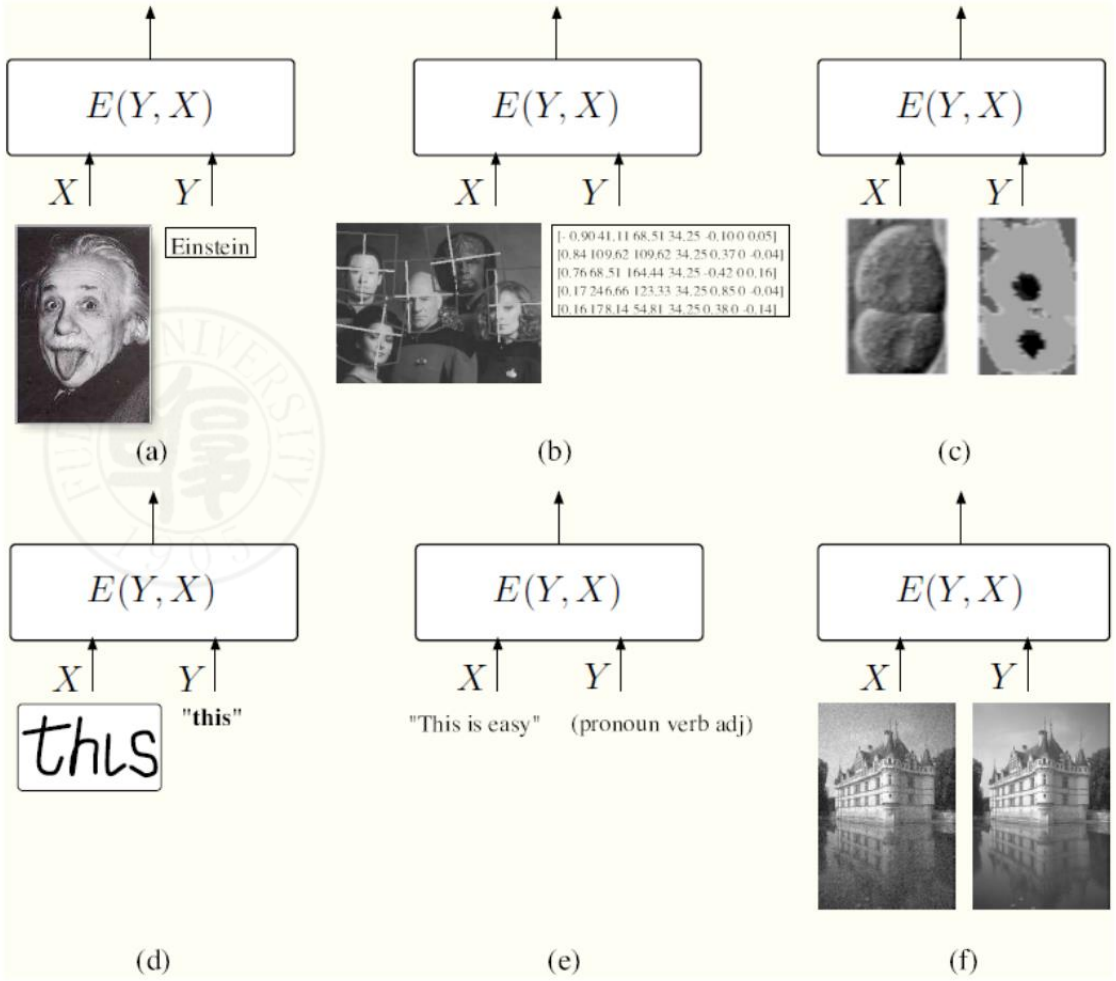
- **能量函数** (Energy Function)：
  - 评价输入 X 与输出 Y 的匹配程度
    - 能量越低 → 组合更合理、概率更高
  - 模型的目标：**寻找能量最小的状态**

$$Y^* = \text{argmin}_{Y \in \mathcal{Y}} E(Y, X)$$



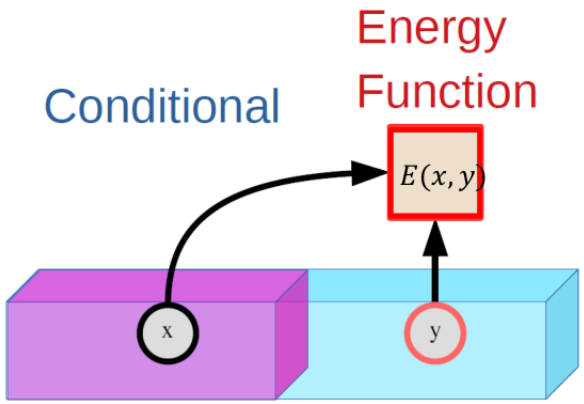
能量函数：实例

- 人脸识别
- 人脸检测
- 图像分割
- 手写字符识别
- 序列标注
- 图像修复

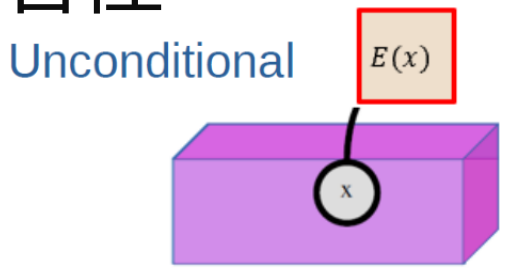


能量函数

- 能量函数 $E(x, y)$：度量 x 和 y 间的相容性
  - 低能量：y 是好的预测结果，$p(y|x)$ 高
    - 例如：$p(\text{"cat"}|$  $)$



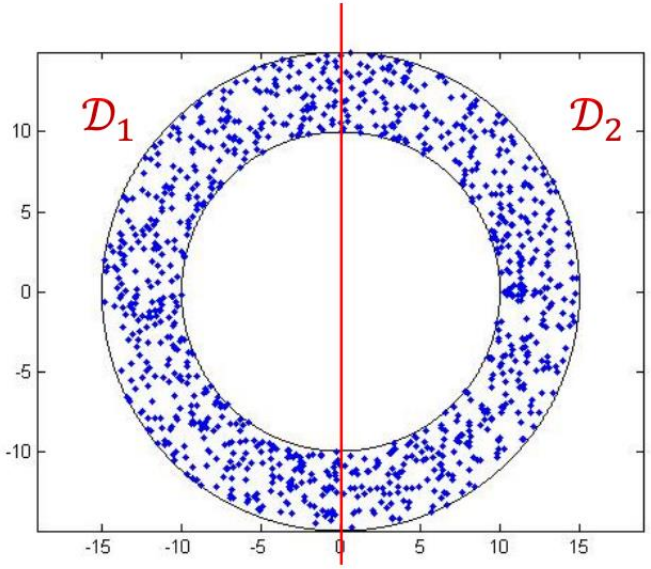
- 无条件能量函数 $E(x)$：度量 x 各分量间的相容性
  - 低能量：$p(x)$ 高
    - 例如：x 更像真实数据



概率分布

- 性质
  - 非负性：$p(x) \geq 0$
  - 归一性：$\sum_x p(x) = 1$ 或 $\int_x p(x)dx = 1$
- 概率空间的总容量固定
  - 例子：均匀分布，求$p(\mathcal{D}_1)$?

$$p(\mathcal{D}_1) = \frac{\text{Volume}(\mathcal{D}_1)}{\text{Volume}(\mathcal{D}_1 + \mathcal{D}_2)}$$



概率分布

- 构造概率分布：
  - 定义任意非负函数$g_\theta(x)$，概率分布 $p_\theta(x)$ 为其归一化

$$p_\theta(\mathbf{x}) = \frac{1}{Volume(g_\theta)} g_\theta(\mathbf{x}) = \frac{1}{\int g_\theta(\mathbf{x}) d\mathbf{x}} g_\theta(\mathbf{x})$$

  - 例如：
    - 定义 $g_{(\mu,\sigma)}(x) = e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
    - Volume为 $\int e^{-\frac{x-\mu}{2\sigma^2}} dx = \sqrt{2\pi\sigma^2}$
    - $p_\theta(x) = \int \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$，即高斯分布

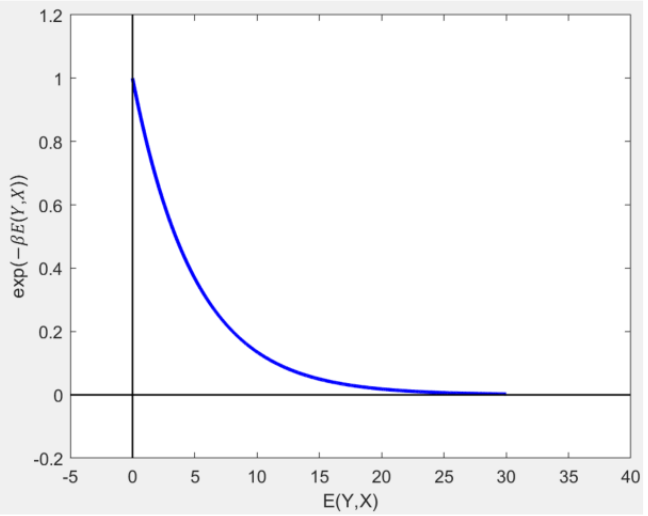
概率分布 vs. 能量函数

- Gibbs分布

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

  - 定义温度 T；配分函数 Z：

$$Z = \int \exp\left(-\frac{E(x)}{T}\right) dx$$



- 概率分布是能量函数的特殊形式
  - 能量函数相当于未归一化的负对数概率

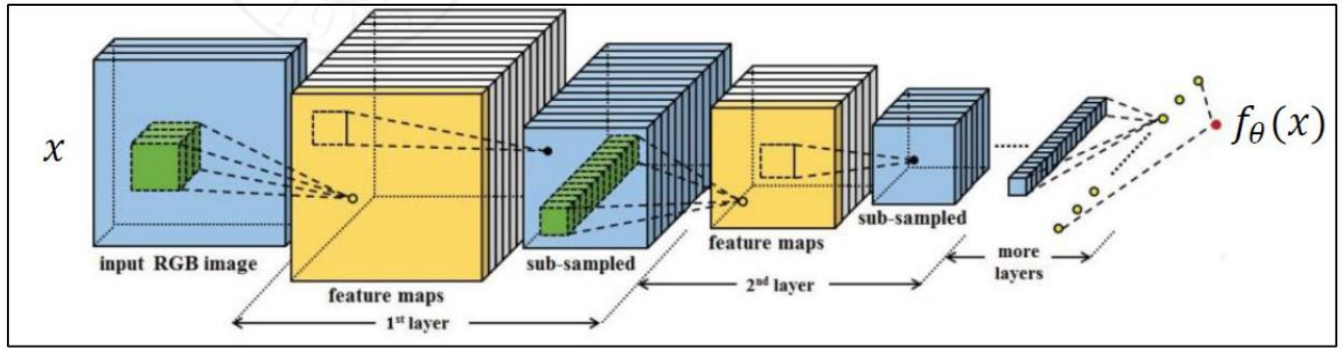
$$E(x) = -\log p(x) - \log Z$$

能量模型

- 能量模型 (Energy-Based Model, EBM)：参数化的能量函数

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp(f_\theta(x)) \qquad Z(\theta) = \int \exp(f_\theta(x))dx$$

  - 其中，$f_\theta(x) = -\frac{E_\theta(x)}{T}$
  - 实现$f_\theta(x)$：
    深度网络，
    例如卷积神经网络



能量模型：讨论

- 能量模型：模型参数 θ 下数据 x 的概率

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp(f_\theta(x)) \qquad Z(\theta) = \int \exp(f_\theta(x)) dx$$

  - 注意到，它类似于 Softmax 概率：

$$\Pr(c) = \frac{1}{Z} \exp(f(c)) = \frac{\exp(f(c))}{\sum_c \exp(f(c))}$$

  - 常关注有约束能量模型 $p_\theta(y|x) = \frac{1}{Z(\theta)} \exp(-\frac{E_\theta(x,y)}{T})$

能量模型：讨论

- 为什么选择指数形式 $\exp\left(f_\theta(x)\right)$?
  - 具有最自然的对数似然形式
    - $\exp(f_\theta(x))$： $\log p_\theta(x) = f_\theta(x) - \log Z(\theta)$
    - 而其它形式，例如 $f_\theta^2(x)$： $\log p_\theta(x) = \log f_\theta^2(x) - \log Z(\theta)$
  - 延续了指数族分布框架
    - 良好的理论基础：充分统计量、最大熵原理、凸优化可解

$$p(x) = h(x)\, e^{\theta^\top T(x) - A(\theta)}$$

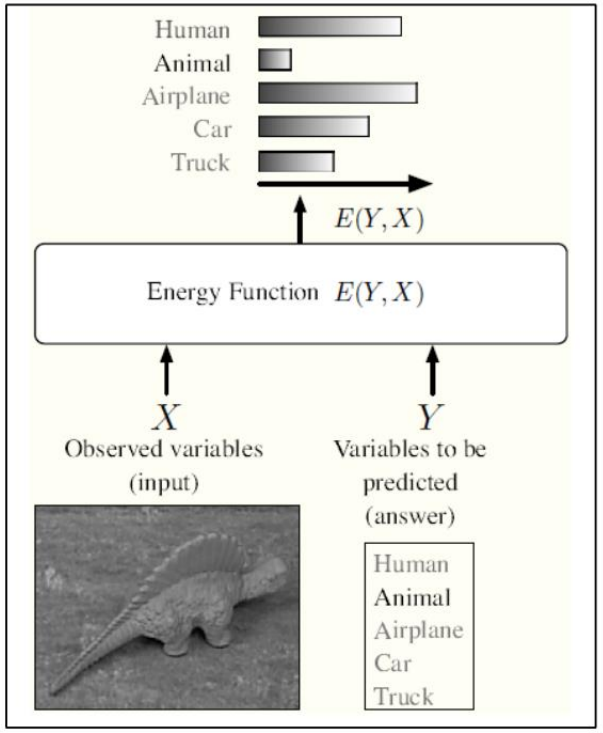
能量模型：讨论

- 计算配分函数 $Z(\theta) = \int \exp(f_\theta(x))\, dx$ 往往很困难
  - 维度灾难，例如：
    - x 是 100^2 的 8-bit 灰度图像；此时图像空间规模为 256^{100^2}
  - 解决方案：MCMC近似
    - 采样$\tilde{n}$个样本$\tilde{x}_i, i = 1, \ldots, \tilde{n}$

$$Z(\theta) = \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \exp(f_\theta(\tilde{x}_i))$$

能量模型

- 实例：分类问题



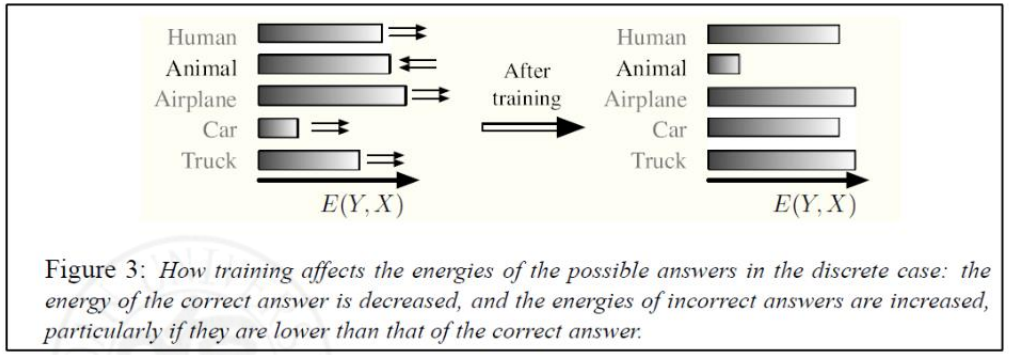


Figure 3: *How training affects the energies of the possible answers in the discrete case: the energy of the correct answer is decreased, and the energies of incorrect answers are increased, particularly if they are lower than that of the correct answer.*

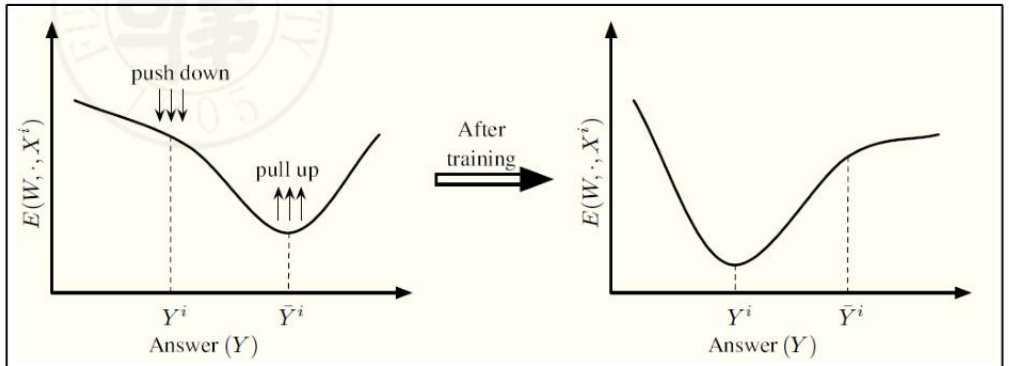


Figure 4: *The effect of training on the energy surface as a function of the answer Y in the continuous case. After training, the energy of the correct answer Y^i is lower than that of incorrect answers.*

能量模型

- 两个核心问题
  - **「学习能量地形」**
    如何估计能量函数 $E(\theta)$ 的模型参数 θ?
    - 极大似然估计
  - **「从能量地形中采样」**
    假设模型参数 θ 已经给定，如何从模型中生成数据?
    - MCMC 近似

KL散度：刻画模型分布

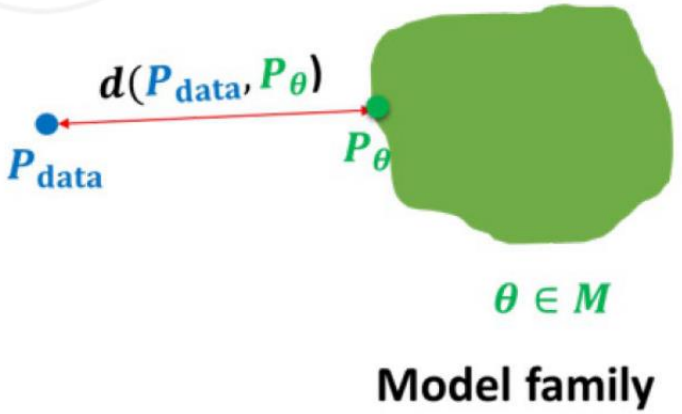
- 从真实分布 $p_{data}(x)$ 独立同分布采样数据

$$x_1, x_2, \ldots, x_n \sim p_{data}(x)$$

- 令模型 $p_\theta(x)$ 近似真实分布
  - KL散度提供「对接近的度量」



$x_i \sim P_{\text{data}}$
$i = 1, 2, \ldots, n$



模型分布：极大似然估计

- 模型分布的对数似然函数

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x_i) \rightarrow \mathbb{E}_{p_{\text{data}}}\left[\log p_\theta(x)\right]$$

  - 当样本量足够多时

$$\begin{aligned} D_{KL}\left(p_{data}(x) \,\|\, p_\theta(x)\right) &= \mathbb{E}_{p_{data}}\left[\log \frac{p_{data}(x)}{p_\theta(x)}\right] \\ &= \mathbb{E}_{p_{data}} \log p_{data}(x) - \mathbb{E}_{p_{data}} \log p_\theta(x) \\ &= \text{constant} - L(\theta) \end{aligned}$$

    最小化 KL 散度 = 最大化对数似然

  - 极大似然估计：等效于让模型分布向真实数据分布靠近

模型分布：变分近似

- 给定能量模型形式的目标分布 p_{target}

$$p_{\text{target}}(x) = \frac{1}{Z}\exp(f(x))$$

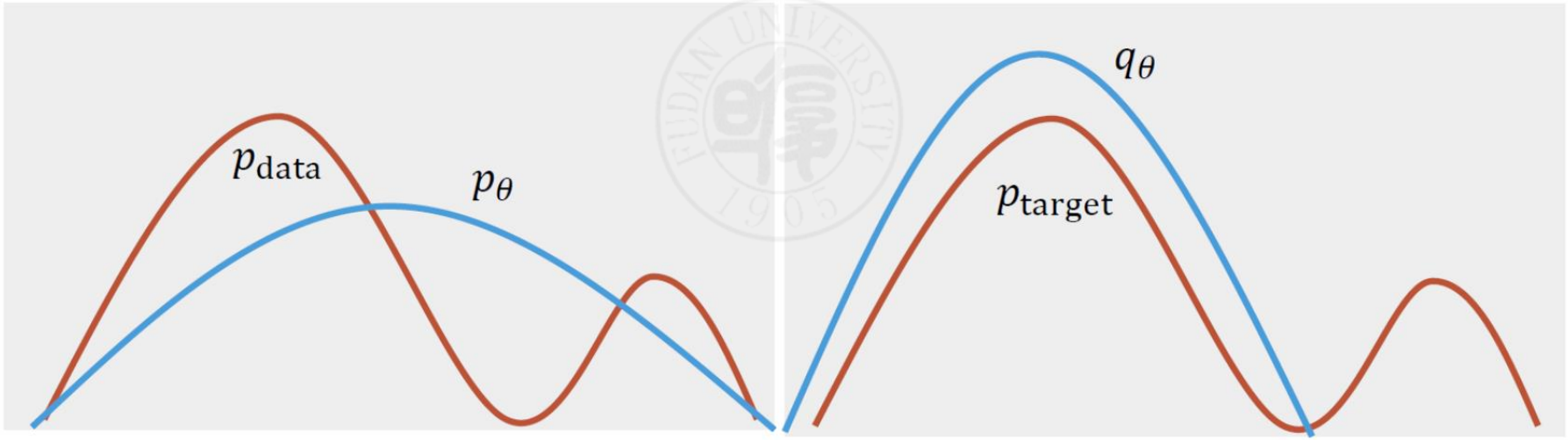
- 无法直接采样 p_{target}
  - 变分近似：希望找到另一容易采样的分布 q_ϕ，并使接近 p_{target}

$$\mathbb{D}_{\text{KL}}\left(q_\phi \| p_{\text{target}}\right) = \mathbb{E}_{q_\phi}\left[\log q_\phi(x)\right] - \mathbb{E}_{q_\phi}\left[f(x)\right] + \log Z$$

模型分布

- 极大似然估计 vs. 变分近似

$$\max L(\theta) = \mathbb{E}_{p_{data}} \log p_\theta(x) \qquad \min \mathbb{E}_{q_\phi}[\log q_\phi(x)] - \mathbb{E}_{q_\phi}[f(x)]$$



极大似然估计：在所有可能的 p_{data} 分布中寻找 $p_\theta(x)$
变分近似： q_ϕ倾向于关注 p_{target} 的主要模式，而忽略次要模式；容易模式坍塌

极大似然估计

- 最大化对数似然函数

$$\max L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x_i)$$

  - 对 $\log p_\theta(x_i)$ 求偏导

$$\nabla_\theta \log p_\theta(x) = \nabla_\theta \log \frac{\exp(f_\theta(x))}{Z(\theta)}$$

$$= \nabla_\theta f_\theta(x) - \nabla_\theta \log Z(\theta)$$

  - 进而

$$\nabla_\theta L(\theta) = \nabla_\theta \frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x_i)$$

$$= \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta f_\theta(x_i) - \nabla_\theta \log Z(\theta)$$

极大似然估计

- 难点：求 $\nabla_\theta \log Z(\theta)$

$$
\begin{aligned}
\nabla_\theta \log Z(\theta) &= \frac{1}{Z(\theta)} \nabla_\theta Z(\theta) \\
&= \frac{1}{Z(\theta)} \nabla_\theta \int \exp(f_\theta(x)) dx \\
&= \frac{1}{Z(\theta)} \int \exp(f_\theta(x)) \nabla_\theta f_\theta(x) dx \\
&= \int \frac{1}{Z(\theta)} \exp(f_\theta(x)) \nabla_\theta f_\theta(x) dx \\
&= \int p_\theta(x) \nabla_\theta f_\theta(x) dx \\
&= \mathbb{E}_{p_\theta(x)} [\nabla_\theta f_\theta(x)]
\end{aligned}
$$

通常是未知或计算困难的

极大似然估计

- 计算 $\mathbb{E}_{p_\theta}[\nabla_\theta f_\theta(x)]$
  - 维度灾难
  - 方法：Monte Carlo 近似
    - 采样 $\tilde{n}$ 个样本 $\tilde{x}_i, i = 1, \ldots, \tilde{n}$

$$\mathbb{E}_{p_\theta}[\nabla_\theta f_\theta(x)] \approx \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \nabla_\theta f_\theta(\tilde{x}_i)$$

极大似然估计

- 对数似然函数求偏导

$$\nabla_\theta L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta f_\theta(x_i) - \nabla_\theta \log Z(\theta)$$

$$= \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta f_\theta(x_i) - \mathbb{E}_{p_\theta}[\nabla_\theta f_\theta(x)]$$

$$\approx \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta f_\theta(x_i) - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \nabla_\theta f_\theta(\tilde{x}_i)$$

观测统计量

合成统计量：用MC近似

极大似然估计

- 迭代算法

  输入：训练样本 $x_1, x_2, \ldots, x_n$，初值 θ_0

  输出：模型参数 θ_T

  For $t = 1$ to T

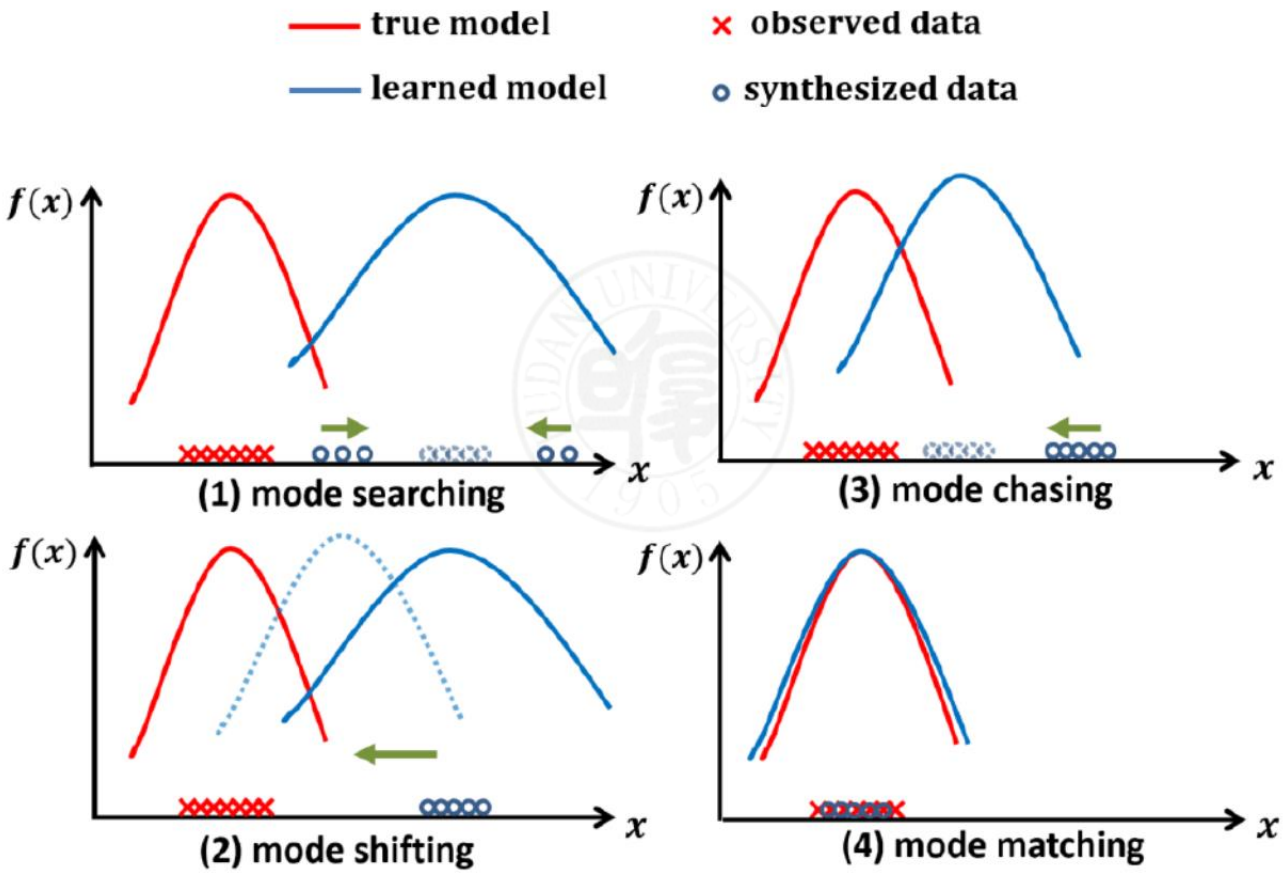
  生成数据 $\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{\tilde{n}}$

  估计模型参数

$$\theta_t = \theta_{t-1} + \eta_t \left[\frac{1}{n} \sum_{i=1}^{n} \nabla_\theta f_\theta(x_i) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \nabla_\theta f_\theta(\tilde{x}_i) \right]$$

极大似然估计

- 例子


true model
observed data
learned model
synthesized data
f(x)
x
(1) mode searching
(3) mode chasing
(2) mode shifting
(4) mode matching

MCMC近似

- 生成数据：
  - 给定模型参数 θ，从分布 $p_\theta(x)$ 中 MCMC 生成数据

    随机初始化 x^0，$t = 1$

    For $t = 1: (m + \tilde{n})$

      令 $x' = x^t + \text{noise}$

      如果 $f_\theta(x') > f_\theta(x^t)$

        令 $x^{t+1} = x'$

      反之

        令 $x^{t+1} = x'$，生成概率为 $\exp(f_\theta(x') - f_\theta(x^t))$

梯度MCMC

- 朗之万动力学 (Langevin Dynamics)

$$x_{t+\Delta t} = x_t + \frac{\Delta t}{2}\nabla_x f_\theta(x_t) + \sqrt{\Delta t}e_t$$

梯度方向　　布朗运动

  - Δt：步长
    - 随着 $\Delta t \to 0$ 和 $t \to \infty$，$p(x_t) \to p_\theta(x)$
    - $e_\tau \sim \mathcal{N}(0, I)$



梯度MCMC

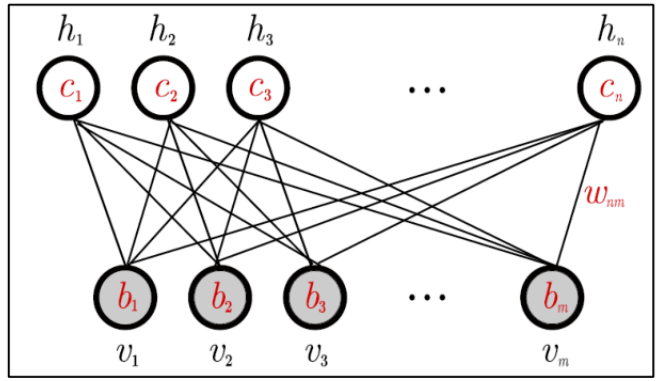
- 朗之万动力学 $$x_{t+\Delta t} = x_t + \frac{\Delta t}{2}\nabla_x f_\theta(x_t) + \sqrt{\Delta t}e_t$$
  - 略作修改：
    - 令初始化 x_0 为纯噪声
    - 明确建模时间步的噪声强度 $\Delta t = \beta_t$
    - 梯度项写成对数概率梯度 $\nabla_x \log p_\theta(x_t)$
  - 结果：Diffusion $$x_{t-1} = x_t - \frac{\beta_t}{2}\nabla_x \log p_\theta(x_t) + \sqrt{\beta_t}\,\epsilon_t$$

目录

- 马尔可夫链蒙特卡罗法（MCMC）
- 能量模型
- 受限玻尔兹曼机（RBM）
- 深度信念网络（DBN）
- 深度玻尔兹曼机（DBM）

受限玻尔兹曼机（RBM）

- 受限玻尔兹曼机 (Restricted Boltzmann Machine)
  - 玻尔兹曼机：无向、成对相互作用的能量网络
    - 一层观测变量 $\boldsymbol{v}$
    - 一层隐变量：学习输入的表示 $\boldsymbol{h}$
    - 变量取值为0/1

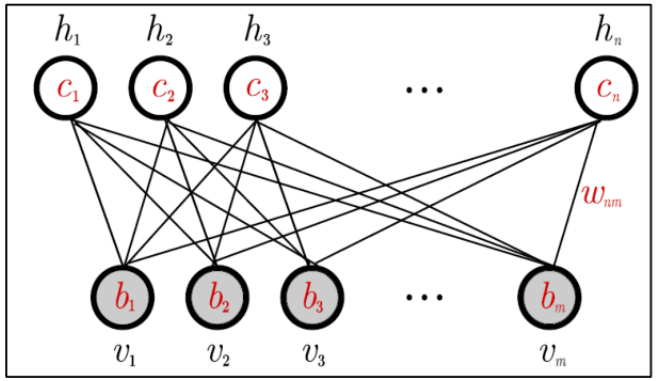


  - 受限：观测变量之间、隐变量之间没有连接，只有v-h之间有连接

$$p(\boldsymbol{v} \mid \boldsymbol{h}) = \prod_{i=1}^{m} p(v_i \mid \boldsymbol{h}) \qquad p(\boldsymbol{h} \mid \boldsymbol{v}) = \prod_{i=1}^{n} p(h_i \mid \boldsymbol{v})$$

受限玻尔兹曼机（RBM）

- 能量函数 $E(v,h) = -b^\top v - c^\top h - v^\top W h$
  - W：模型参数的权重矩阵
  - b, c：偏置向量
- 联合概率分布 $p(v,h) = \dfrac{\exp(-E(v,h))}{Z}$



- 观测变量的边缘分布 $p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$

受限玻尔兹曼机（RBM）

- 观测变量分布

$$
\begin{aligned}
p(\boldsymbol{v}) &= \frac{1}{Z}\sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \\
&= \frac{1}{Z}\sum_{h_1}\sum_{h_2}\cdots\sum_{h_n} e^{\sum_{j=1}^{m} b_j v_j} \prod_{i=1}^{n} e^{h_i\left(c_i+\sum_{j=1}^{m} w_{ij} v_j\right)} \\
&= \frac{1}{Z} e^{\sum_{j=1}^{m} b_j v_j} \sum_{h_1} e^{h_1\left(c_1+\sum_{j=1}^{m} w_{1j} v_j\right)} \sum_{h_2} e^{h_2\left(c_2+\sum_{j=1}^{m} w_{2j} v_j\right)} \cdots \sum_{h_n} e^{h_n\left(c_n+\sum_{j=1}^{m} w_{nj} v_j\right)} \\
&= \frac{1}{Z} e^{\sum_{j=1}^{m} b_j v_j} \prod_{i=1}^{n}\sum_{h_i} e^{h_i\left(c_i+\sum_{j=1}^{m} w_{ij} v_j\right)} \\
&= \frac{1}{Z}\prod_{j=1}^{m} e^{b_j v_j} \prod_{i=1}^{n}\left(1+e^{c_i+\sum_{j=1}^{m} w_{ij} v_j}\right) \qquad (22)
\end{aligned}
$$

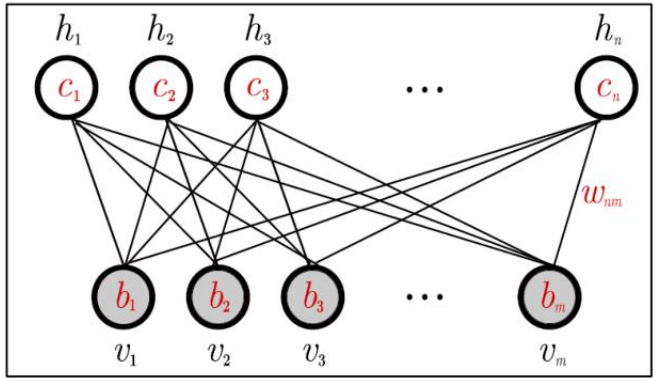
专家模型：神经网络

  - RBM被认为是多专家模型的乘积

受限玻尔兹曼机（RBM）

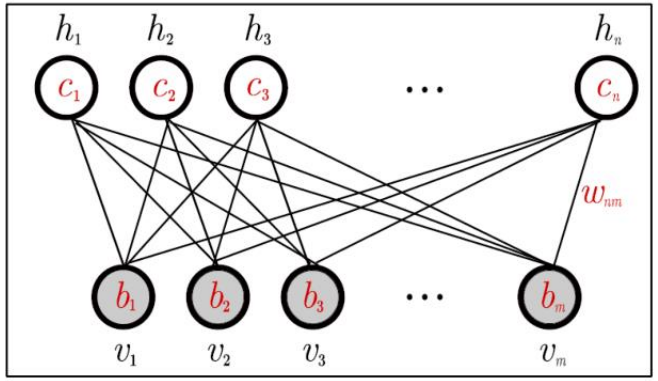
- 条件概率分布

$$
\begin{aligned}
P(\boldsymbol{h} \mid \boldsymbol{v}) &= \frac{P(\boldsymbol{h}, \boldsymbol{v})}{P(\boldsymbol{v})} \\
&= \frac{1}{P(\boldsymbol{v})} \frac{1}{Z} \exp\left\{ \boldsymbol{b}^\top \boldsymbol{v} + \boldsymbol{c}^\top \boldsymbol{h} + \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h} \right\} \\
&= \frac{1}{Z'} \exp\left\{ \boldsymbol{c}^\top \boldsymbol{h} + \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h} \right\} \\
&= \frac{1}{Z'} \exp\left\{ \sum_{j=1}^{n} c_j h_j + \sum_{j=1}^{n} \boldsymbol{v}^\top \boldsymbol{W}_{:,j} h_j \right\} \\
&= \frac{1}{Z'} \prod_{j=1}^{n} \exp\left\{ c_j h_j + \boldsymbol{v}^\top \boldsymbol{W}_{:,j} h_j \right\}.
\end{aligned}
$$



受限玻尔兹曼机（RBM）

- 每个隐含单元的条件概率
  - 观测变量 $v \in \{0,1\}^m$
  - 隐含变量 $h \in \{0,1\}^n$



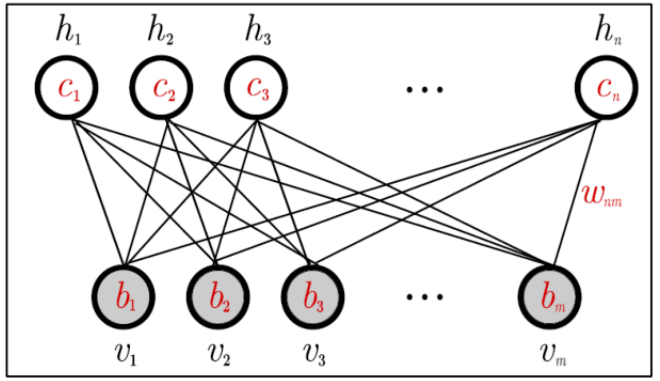
$$
\begin{aligned}
P(h_j = 1 \mid \boldsymbol{v}) &= \frac{\tilde{P}(h_j = 1 \mid \boldsymbol{v})}{\tilde{P}(h_j = 0 \mid \boldsymbol{v}) + \tilde{P}(h_j = 1 \mid \boldsymbol{v})} \\
&= \frac{\exp\left\{c_j + \boldsymbol{v}^\top \boldsymbol{W}_{:,j}\right\}}{\exp\{0\} + \exp\left\{c_j + \boldsymbol{v}^\top \boldsymbol{W}_{:,j}\right\}} \\
&= \sigma\left(c_j + \boldsymbol{v}^\top \boldsymbol{W}_{:,j}\right).
\end{aligned}
$$

  - 其中，$\sigma(x) = \frac{1}{1+\exp(-x)}$ 是 sigmoid 激活函数

受限玻尔兹曼机（RBM）

- 每个隐含单元的条件概率

$$
\begin{aligned}
P(h_j = 0|v) &= \frac{\tilde{P}(h_j = 0|v)}{\tilde{P}(h_j = 0|v)+\tilde{P}(h_j = 1|v)} \\
&= \frac{\exp\{0\}}{\exp\{0\}+\exp\{c_j+v^T \boldsymbol{W}_{:,j}\}} \\
&= \sigma(-c_j - v^T \boldsymbol{W}_{:,j})
\end{aligned}
$$



- 条件概率分布

$$
P(\boldsymbol{h} \mid \boldsymbol{v}) = \prod_{j=1}^{n} \sigma\left((2\boldsymbol{h} - 1) \odot (\boldsymbol{c} + \boldsymbol{W}^\top \boldsymbol{v})\right)_j
$$

$$
P(\boldsymbol{v} \mid \boldsymbol{h}) = \prod_{i=1}^{m} \sigma\left((2\boldsymbol{v} - 1) \odot (\boldsymbol{b} + \boldsymbol{W}\boldsymbol{h})\right)_i
$$

受限玻尔兹曼机（RBM）

- 两个主要问题：
  - 模型参数估计 W, b, c
    - 极大似然估计
  - 从估计分布中采样
    - 块吉布斯采样 (Block Gibbs Sampling)

受限玻尔兹曼机（RBM）

- 极大似然估计
  - 单个观测数据的对数似然函数

$$\ln \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{v}) = \ln p(\boldsymbol{v} \mid \boldsymbol{\theta}) = \ln \frac{1}{Z} \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} = \ln \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} - \ln \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

真实数据区域的能量　　能量地形的整体水平

  - 求偏导

$$\begin{aligned}
\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{v})}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}} \left(\ln \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \right) - \frac{\partial}{\partial \boldsymbol{\theta}} \left(\ln \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \right) \\
&= -\frac{1}{\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}} \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} + \frac{1}{\sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}} \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} \\
&= -\sum_{\boldsymbol{h}} p(\boldsymbol{h} \mid \boldsymbol{v}) \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} + \sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}}
\end{aligned}$$

正项：数据期望　　负项：模型期望

受限玻尔兹曼机（RBM）

- 极大似然估计
  - 单个观测数据的对数似然函数
    - 第一项：数据期望，关于权重 W 求偏导

$$E(v,h) = -b^\top v - c^\top h - v^\top W h$$

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

激活期望 × 当前输入

$$= \sum_{\boldsymbol{h}} \prod_{k=1}^{n} p(h_k\,|\,\boldsymbol{v}) h_i v_j = \sum_{h_i}\sum_{\boldsymbol{h}_{-i}} p(h_i\,|\,\boldsymbol{v}) p(\boldsymbol{h}_{-i}\,|\,\boldsymbol{v}) h_i v_j$$

$$= \sum_{h_i} p(h_i\,|\,\boldsymbol{v}) h_i v_j \underbrace{\sum_{\boldsymbol{h}_{-i}} p(\boldsymbol{h}_{-i}|\boldsymbol{v})}_{=1} = p(H_i = 1|\boldsymbol{v}) v_j = \sigma\left(\sum_{j=1}^{m} w_{ij} v_j + c_i\right) v_j$$

受限玻尔兹曼机（RBM）

- 极大似然估计
  - 单个观测数据的整体对数似然，关于 W 求偏导

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{v})}{\partial w_{ij}} = -\sum_{\boldsymbol{h}} p(\boldsymbol{h} \mid \boldsymbol{v}) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial w_{ij}} + \sum_{\boldsymbol{v}, \boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial w_{ij}}$$

$$= \sum_{\boldsymbol{h}} p(\boldsymbol{h} \mid \boldsymbol{v}) h_i v_j - \sum_{\boldsymbol{v}} p(\boldsymbol{v}) \sum_{\boldsymbol{h}} p(\boldsymbol{h} \mid \boldsymbol{v}) h_i v_j$$

对所有 $\{v, h\}$ 求和是困难的！

$$= p(H_i = 1 \mid \boldsymbol{v}) v_j - \sum_{\boldsymbol{v}} p(\boldsymbol{v}) p(H_i = 1 \mid \boldsymbol{v}) v_j$$

受限玻尔兹曼机（RBM）

- 极大似然估计
  - 整个数据集的对数似然函数，关于 W 求偏导
    - 给定训练集$S = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\ell\}$和$q(\boldsymbol{v})$：经验分布

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

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

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

$$\langle v_i h_j \rangle_{\text{data}} = \mathbb{E}_{v\sim q(v),\, h\sim p(h|v)}[v_i h_j]$$

$$\langle v_i h_j \rangle_{\text{model}} = \mathbb{E}_{(v,h)\sim p(v,h)}[v_i h_j]$$

$$\Rightarrow \quad \sum_{\boldsymbol{v}\in S}\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta}\,|\,\boldsymbol{v})}{\partial w_{ij}} \propto \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}$$

受限玻尔兹曼机（RBM）

- 极大似然估计，同理对其他参数求偏导
  - 单个数据的对数似然，关于 b 求偏导

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{v})}{\partial b_j} = v_j - \sum_{\boldsymbol{v}} p(\boldsymbol{v}) v_j$$

  - 单个数据的对数似然，关于 c 求偏导

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{v})}{\partial c_i} = p(H_i = 1 \mid \boldsymbol{v}) - \sum_{\boldsymbol{v}} p(\boldsymbol{v}) p(H_i = 1 \mid \boldsymbol{v})$$

  - “共同形式”：梯度 = 数据期望 – 模型期望

受限玻尔兹曼机（RBM）

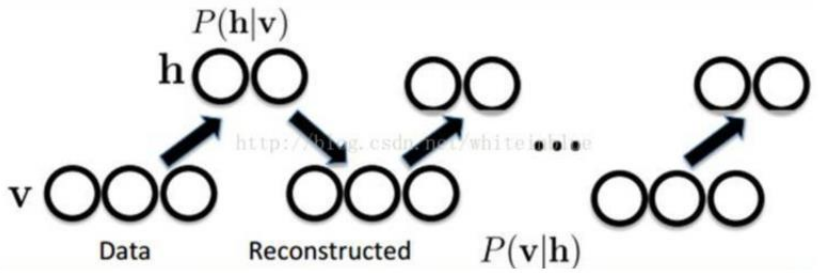
- 对比散度 (Contrastive Divergence, CD)
  - 从真实数据出发，进行有限步 Gibbs 采样

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

真实数据分布　　经过第 k 步 Gibbs 采样的模型近似分布

  - 给定$v^{(t)}$，从分布$p(h|v^{(t)})$采样$h^{(t)}$
  - 给定$h^{(t)}$，从分布$p(v|h^{(t)})$采样$v^{(t+1)}$

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

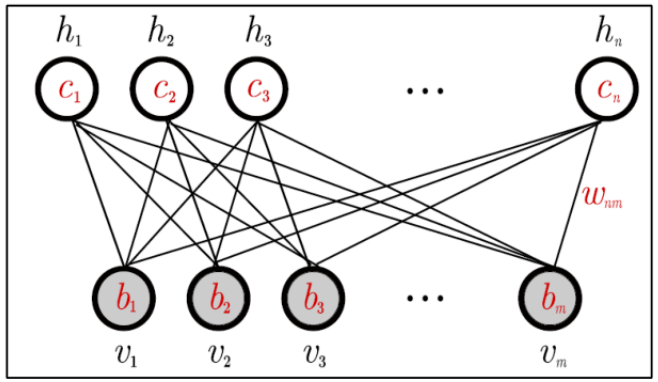


受限玻尔兹曼机（RBM）

- 块吉布斯采样 (Block Gibbs Sampling)
  - 给定所有隐藏单元，所有可见单元可以被并行采样，因为层内无边

$$p(\boldsymbol{v} \mid \boldsymbol{h}) = \prod_{i=1}^{m} p(v_i \mid \boldsymbol{h})$$

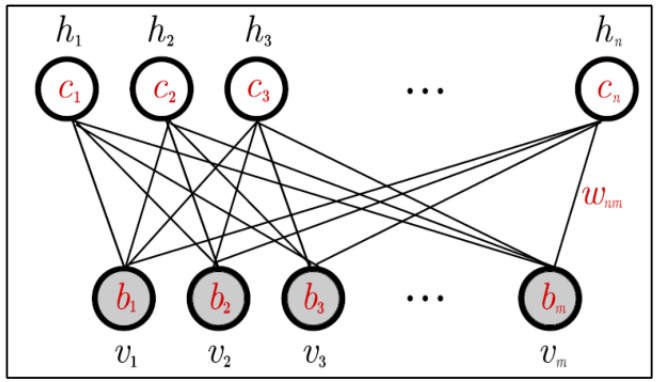
$$\begin{aligned} p(v_l = 1 \mid \boldsymbol{h}) &= \frac{p(v_l = 1, \boldsymbol{h})}{p(v_l = 1, \boldsymbol{h}) + p(v_l = 0, \boldsymbol{h})} \\ &= \frac{\exp\left(b_l + \sum_i W_{il} h_i\right)}{\exp\left(b_l + \sum_i W_{il} h_i\right) + 1} \\ &= \sigma\left(b_l + \sum_i W_{il} h_i\right) \end{aligned}$$



受限玻尔兹曼机（RBM）

- 块吉布斯采样 (Block Gibbs Sampling)
  - 给定所有可见单元，所有隐藏单元可以被并行采样

$$p(\boldsymbol{h} \mid \boldsymbol{v}) = \prod_{i=1}^{n} p(h_i \mid \boldsymbol{v})$$



受限玻尔兹曼机（RBM）

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 $\Delta w_{ij} = \Delta b_j = \Delta c_i = 0$ for $i = 1, \ldots, n$, $j = 1, \ldots, m$

forall the $\boldsymbol{v} \in S$ **do**

  $\boldsymbol{v}^{(0)} \leftarrow \boldsymbol{v}$

  for $t = 0, \ldots, k-1$ **do**

    for $i = 1, \ldots, n$ **do** sample $h_i^{(t)} \sim p(h_i \mid \boldsymbol{v}^{(t)})$

    for $j = 1, \ldots, m$ **do** sample $v_j^{(t+1)} \sim p(v_j \mid \boldsymbol{h}^{(t)})$

块吉布斯采样

  for $i = 1, \ldots, n$, $j = 1, \ldots, m$ **do**

    $\Delta w_{ij} \leftarrow \Delta w_{ij} + p(H_i = 1 \mid \boldsymbol{v}^{(0)}) \cdot v_j^{(0)} - p(H_i = 1 \mid \boldsymbol{v}^{(k)}) \cdot v_j^{(k)}$

    $\Delta b_j \leftarrow \Delta b_j + v_j^{(0)} - v_j^{(k)}$

    $\Delta c_i \leftarrow \Delta c_i + p(H_i = 1 \mid \boldsymbol{v}^{(0)}) - p(H_i = 1 \mid \boldsymbol{v}^{(k)})$

CD-k

受限玻尔兹曼机（RBM）

- 对比散度的理论依据

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 $\boldsymbol{v}^{(0)}$*, the log-likelihood gradient can be written as*

$$\frac{\partial}{\partial \boldsymbol{\theta}} \ln p(\boldsymbol{v}^{(0)}) = -\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^{(0)}) \frac{\partial E(\boldsymbol{v}^{(0)}, \boldsymbol{h})}{\partial \boldsymbol{\theta}}$$

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

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

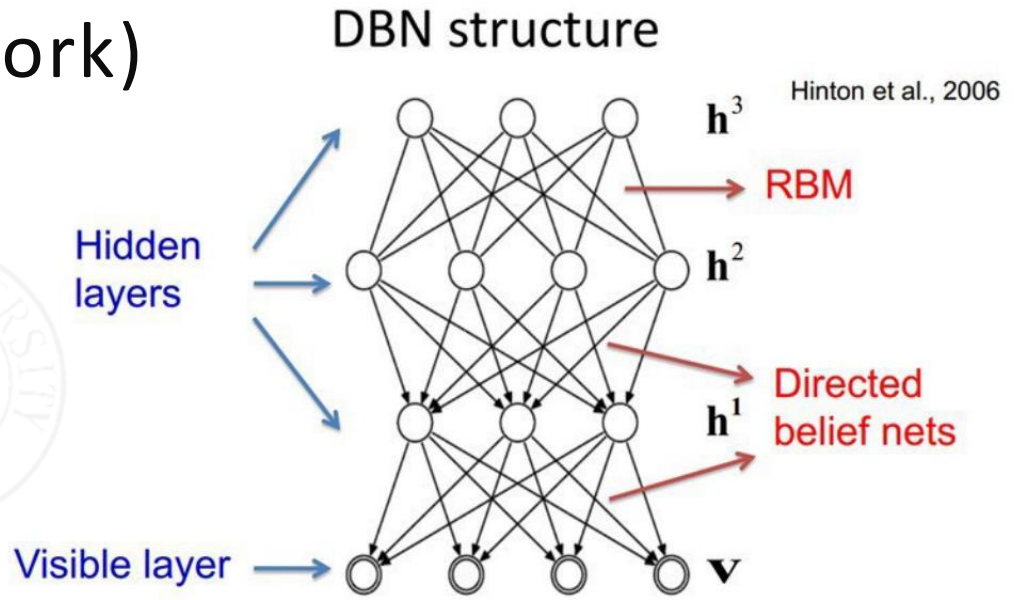
第三项是残差，定理保证它随 k → ∞ 趋于零

目录

- 马尔可夫链蒙特卡罗法（MCMC）
- 能量模型
- 受限玻尔兹曼机（RBM）
- 深度信念网络（DBN）
- 深度玻尔兹曼机（DBM）

深度信念网络（DBN）

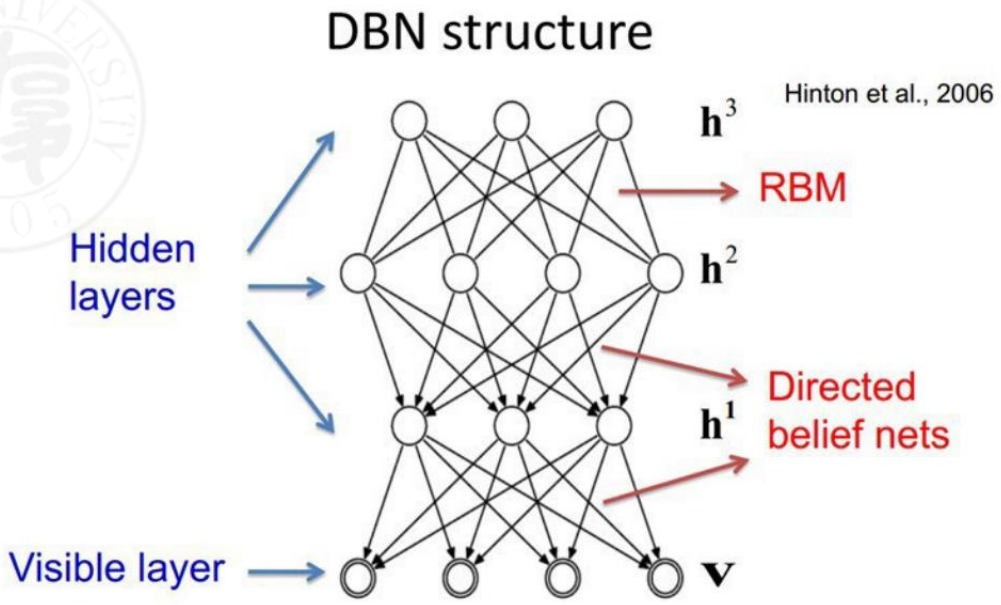
- 深度信念网络 (Deep Belief Network)
  - 是具有若干隐变量层的生成模型
    - 可见单元：二值或实数
    - 隐变量：通常是二值的
  - 是第一批成功应用深度架构训练的非卷积模型之一



深度信念网络（DBN）

- 有向和无向连接的混合图模型
  - 从 v 层出发，以 RBM 建模 $p(v|h^1)$，冻结为有向图
  - 迭代建模 $p(h^1|h^2), p(h^2|h^3)$…

$$p(v,h^1,h^2,h^3) = p(v|h^1,h^2,h^3)p(h^1,h^2,h^3)$$
$$= p(v|h^1)p(h^1|h^2)p(h^2,h^3)$$



深度信念网络（DBN）

- 概率分布：l 个隐藏层
  - l 个权重矩阵 $W^{(1)}, \ldots, W^{(l)}$
  - $l+1$ 个偏置向量 $b^{(0)}, b^{(1)}, \ldots, b^{(l)}$，其中 $b^{(0)}$ 是可见层的偏置向量
- 概率分布

$$P(\boldsymbol{h}^{(l)}, \boldsymbol{h}^{(l-1)}) \propto \exp\left(\boldsymbol{b}^{(l)} \boldsymbol{h}^{(l)} + \boldsymbol{b}^{(l-1)\top} \boldsymbol{h}^{(l-1)} + \boldsymbol{h}^{(l-1)\top} \boldsymbol{W}^{(l)} \boldsymbol{h}^{(l)}\right),$$

无向部分（顶层RBM）

$$P(h_i^{(k)} = 1 \mid \boldsymbol{h}^{(k+1)}) = \sigma\left(b_i^{(k)} + \boldsymbol{W}_{:,i}^{(k+1)\top} \boldsymbol{h}^{(k+1)}\right) \forall i, \forall k \in 1, \ldots, l-2,$$

有向部分

深度信念网络（DBN）

- 概率分布：可见层
  - 如果可见层是二值

$$P(v_i = 1 \mid \boldsymbol{h}^{(1)}) = \sigma\left(b_i^{(0)} + \boldsymbol{W}_{:,i}^{(1)\top} \boldsymbol{h}^{(1)}\right) \forall i$$

  - 如果可见层取实值

$$\mathbf{v} \sim \mathcal{N}\left(\boldsymbol{v}; \boldsymbol{b}^{(0)} + \boldsymbol{W}^{(1)\top} \boldsymbol{h}^{(1)}, \boldsymbol{\beta}^{-1}\right)$$

深度信念网络（DBN）

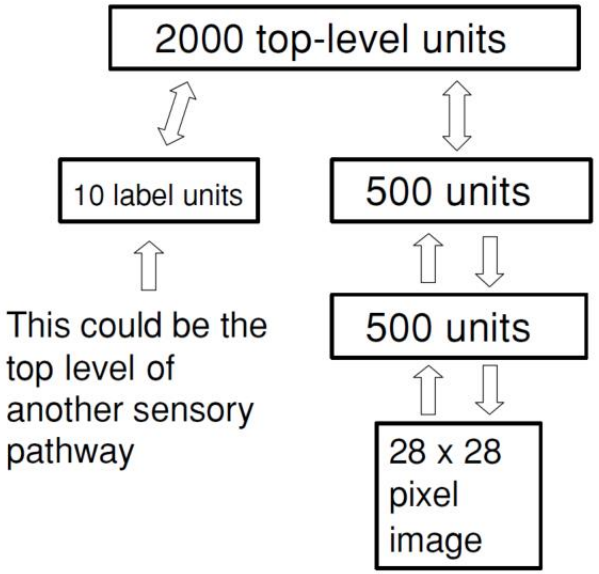
- 训练过程
  - 充分训练第一个 RBM——CD算法
  - 固定第一个 RBM 的 W, b，使用其隐神经元的状态，作为第二个 RBM 的输入向量

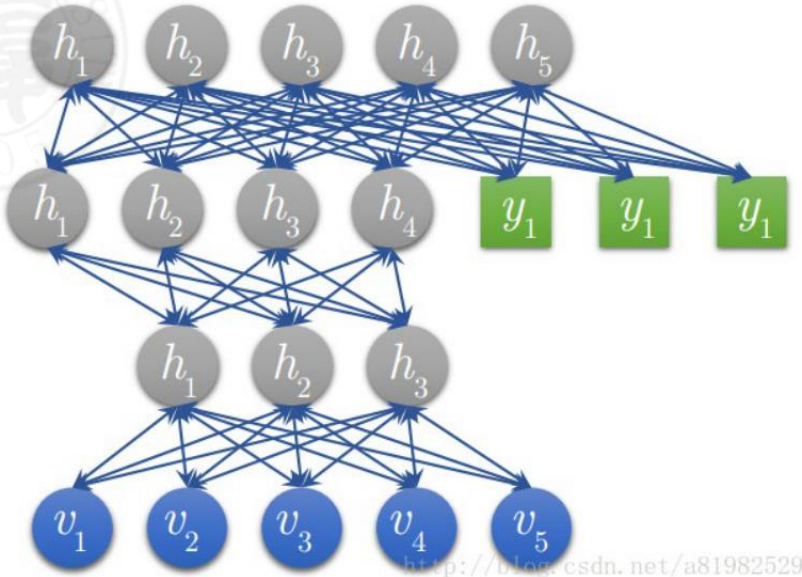
$$\mathrm{E}_{\mathrm{v}\sim p_{\mathrm{data}}} \mathrm{E}_{\mathrm{h}^{(1)}\sim p^{(1)}(\boldsymbol{h}^{(1)}|\boldsymbol{v})} \log p^{(2)}(\boldsymbol{h}^{(1)})$$

  - 充分训练第二个 RBM 后，将第二个 RBM 堆叠在第一个 RBM 的上方
  - 重复以上三个步骤任意多次

深度信念网络（DBN）

- 训练过程
  - 如果训练数据有标签，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

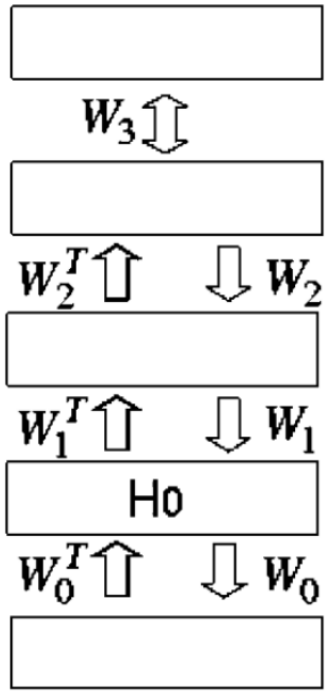




http://blog.csdn.net/a819825294

深度信念网络（DBN）

- 调优过程：Contrastive Wake-Sleep 算法
  - 除了顶层RBM，其他层RBM的权重被分成向上的认知权重和向下的生成权重；
  - Wake阶段：认知过程，通过外界的特征和向上的权重产生每一层的抽象表示 (结点状态)，并且使用梯度下降修改层间的下行权重 (生成权重)；
  - Sleep阶段：生成过程，通过顶层表示 (醒时学得的概念) 和向下权重，生成底层的状态，同时修改层间向上的权重。



目录

- 马尔可夫链蒙特卡罗法（MCMC）
- 能量模型
- 受限玻尔兹曼机（RBM）
- 深度信念网络（DBN）
- 深度玻尔兹曼机（DBM）

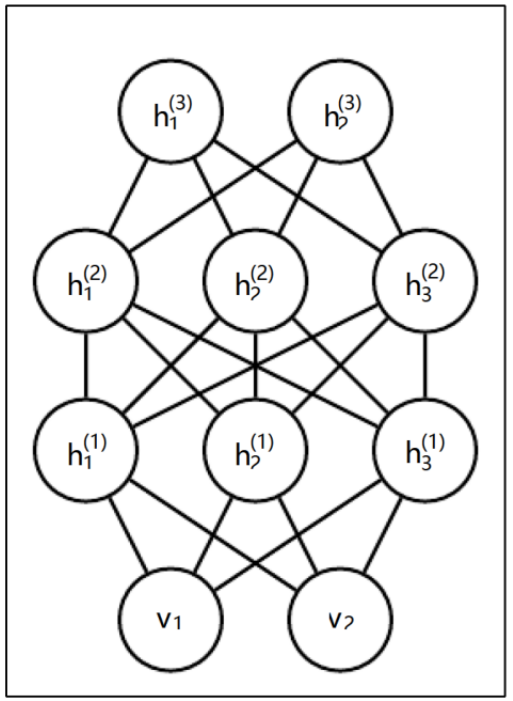
深度玻尔兹曼机（DBM）

- 是具有若干隐变量层的生成模型
  - 可见单元：二值
  - 隐变量：二值
  - 完全无向图
- 能量模型
  - 为简化表示，省略了偏置参数

$$E(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \boldsymbol{h}^{(3)}; \boldsymbol{\theta}) = -\boldsymbol{v}^{\top}\boldsymbol{W}^{(1)}\boldsymbol{h}^{(1)} - \boldsymbol{h}^{(1)\top}\boldsymbol{W}^{(2)}\boldsymbol{h}^{(2)} - \boldsymbol{h}^{(2)\top}\boldsymbol{W}^{(3)}\boldsymbol{h}^{(3)}$$

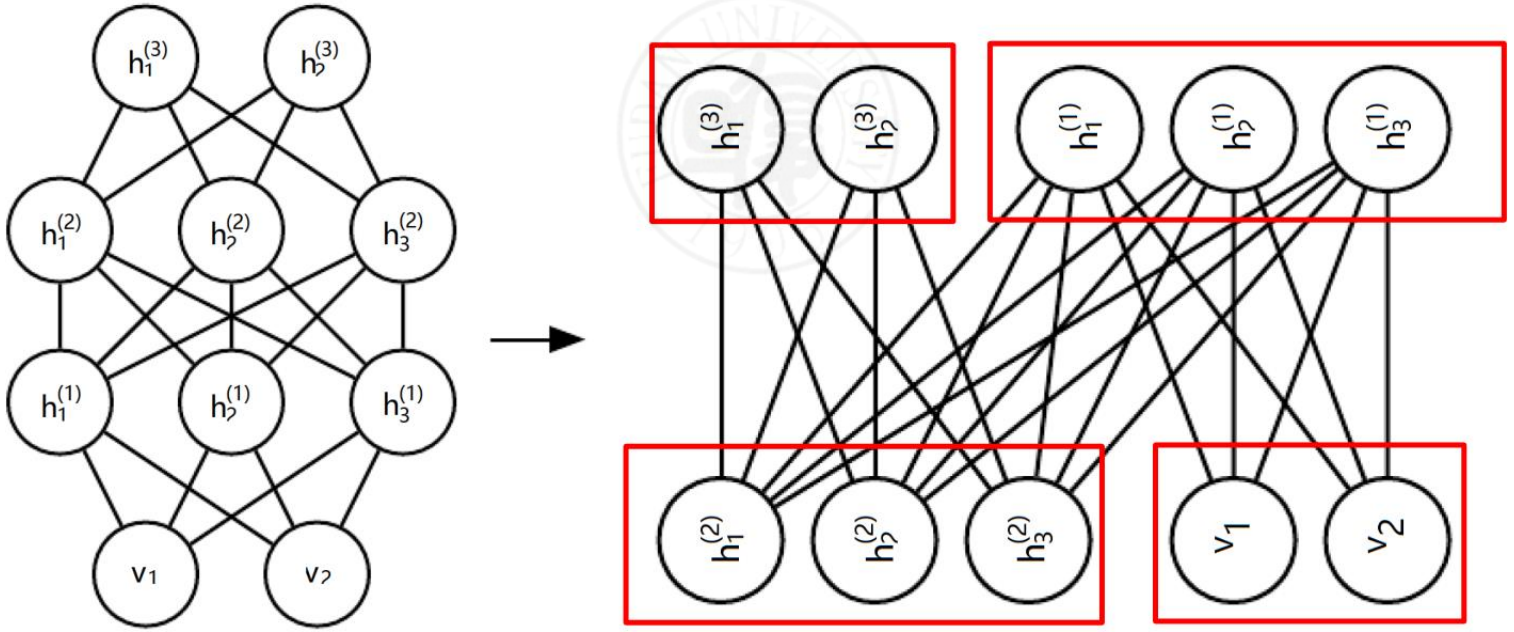
  - 联合概率分布

$$P\left(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \boldsymbol{h}^{(3)}\right) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(-E(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \boldsymbol{h}^{(3)}; \boldsymbol{\theta})\right)$$



深度玻尔兹曼机（DBM）

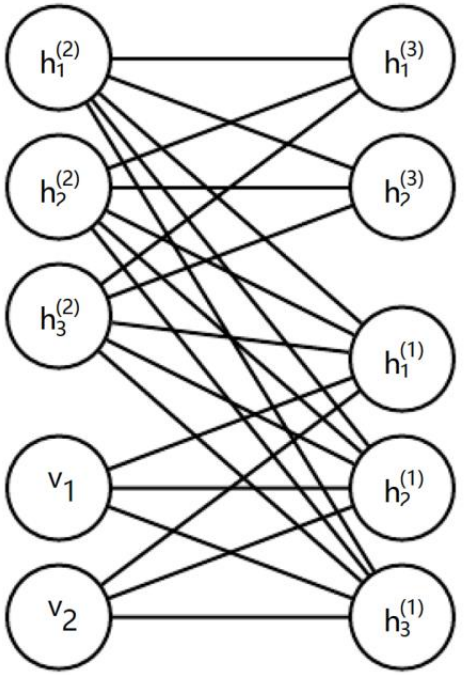
- DBM的层可以组织成一个二分图，其中奇数层在一侧，偶数层在另一侧



深度玻尔兹曼机（DBM）

- 具有 l 个隐藏层的DBM只需要2步更新
  - 给定偶数层，关于奇数层的分布是因子的，因此可以作为块同时且独立地采样
  - 给定奇数层，可以同时且独立地

    将偶数层作为块进行采样



深度玻尔兹曼机（DBM）

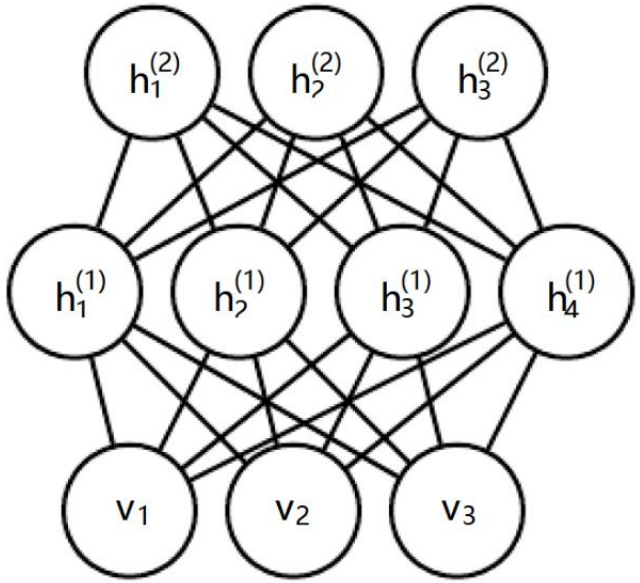
- 训练方法
  - 2009年，提出了深度玻尔兹曼机的思想，并提出逐层预训练法的学习算法
  - 2010年，提出了耦合自适应模拟回火方法
  - 2012年，提出了一种变分思想，通过学习使真实后验分布和平均场变分推断假定因子分布接近， 来估计依赖数据的期望值

Salakhutdinov, R., Hinton, G.E . Deep boltzmann machines. Artificial Intelligence and Statistics (pp. 448-455).
Salakhutdinov, R. Learning deep Boltzmann machines using adaptive MCMC. ICML 2010
Salakhutdinov, R., Hinton, G. E. An efficient learning procedure for deep Boltzmann machines. Neural Computation.24(8)

深度玻尔兹曼机（DBM）

- 难点：$P(h^{(1)}, h^{(2)}|v)$不是因子结构的
  - 给定可见层 v 后，隐层之间依然是相互依赖的

$$P\left(h^{(1)}, h^{(2)}|v\right) = P\left(h^{(1)}|h^{(2)}, v\right)P(h^{(2)}|v)$$



深度玻尔兹曼机（DBM）

- 平均场推断：以两个隐藏层为例
  - 令$Q(h^{(1)}, h^{(2)}|v)$是$P(h^{(1)}, h^{(2)}|v)$的近似

$$Q(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)} \mid \boldsymbol{v}) = \prod_j Q(h_j^{(1)} \mid \boldsymbol{v}) \prod_k Q(h_k^{(2)} \mid \boldsymbol{v})$$

    - 后验分布

$$Q(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)} \mid \boldsymbol{v}) = \prod_j Q(h_j^{(1)} \mid \boldsymbol{v}) \prod_k Q(h_k^{(2)} \mid \boldsymbol{v})$$

$$= \prod_j (\hat{h}_j^{(1)})^{h_j^{(1)}} (1 - \hat{h}_j^{(1)})^{(1-h_j^{(1)})} \times \prod_k (\hat{h}_k^{(2)})^{h_k^{(2)}} (1 - \hat{h}_k^{(2)})^{(1-h_k^{(2)})}$$

$$\hat{h}_j^{(1)} = Q(h_j^{(1)} = 1 \mid \boldsymbol{v}) \qquad \hat{h}_k^{(2)} = Q(h_k^{(2)} = 1 \mid \boldsymbol{v})$$

深度玻尔兹曼机（DBM）

- 平均场推断：以两个隐藏层为例
  - 最小化Q和P之间的距离

$$\mathrm{KL}(Q\|P) = \sum_{\boldsymbol{h}} Q(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)} \mid \boldsymbol{v}) \log \left(\frac{Q(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)} \mid \boldsymbol{v})}{P(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)} \mid \boldsymbol{v})} \right)$$

$$\hat{h}_j^{(1)} = \sigma \left(\sum_i v_i W_{i,j}^{(1)} + \sum_k W_{j,k}^{(2)} \hat{h}_k^{(2)} \right), \quad \forall j,$$

$$\hat{h}_k^{(2)} = \sigma \left(\sum_j W_{j,k}^{(2)} \hat{h}_j^{(1)} \right), \quad \forall k.$$

深度玻尔兹曼机（DBM）

- 平均场推断：以两个隐藏层为例
  - 参数学习：变分推断
    - 最大化证据下界ELBO

$$\mathcal{L}(Q,\boldsymbol{\theta}) = \sum_i \sum_j v_i W^{(1)}_{i,j} \hat{h}^{(1)}_j + \sum_j \sum_k \hat{h}^{(1)}_j W^{(2)}_{j,k} \hat{h}^{(2)}_k - \log Z(\boldsymbol{\theta}) + \mathcal{H}(Q)$$

深度玻尔兹曼机（DBM）

- 算法伪代码

Algorithm 20.1 The variational stochastic maximum likelihood algorithm for training a DBM with two hidden layers

Set ϵ, the step size, to a small positive number
Set k, the number of Gibbs steps, high enough to allow a Markov chain of $p(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}; \boldsymbol{\theta} + \epsilon \Delta_{\boldsymbol{\theta}})$ to burn in, starting from samples from $p(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}; \boldsymbol{\theta})$.

Initialize three matrices, $\tilde{\boldsymbol{V}}$, $\tilde{\boldsymbol{H}}^{(1)}$, and $\tilde{\boldsymbol{H}}^{(2)}$ each with m rows set to random values (e.g., from Bernoulli distributions, possibly with marginals matched to the model's marginals).
while not converged (learning loop) **do**
  Sample a minibatch of m examples from the training data and arrange them as the rows of a design matrix $\boldsymbol{V}$.
  Initialize matrices $\hat{\boldsymbol{H}}^{(1)}$ and $\hat{\boldsymbol{H}}^{(2)}$, possibly to the model's marginals.
  while not converged (mean field inference loop) **do**
    $\hat{\boldsymbol{H}}^{(1)} \leftarrow \sigma\left(\boldsymbol{V}\boldsymbol{W}^{(1)} + \hat{\boldsymbol{H}}^{(2)}\boldsymbol{W}^{(2)\top}\right)$.
    $\hat{\boldsymbol{H}}^{(2)} \leftarrow \sigma\left(\hat{\boldsymbol{H}}^{(1)}\boldsymbol{W}^{(2)}\right)$.
  end while
  $\Delta_{\boldsymbol{W}^{(1)}} \leftarrow \frac{1}{m}\boldsymbol{V}^{\top}\hat{\boldsymbol{H}}^{(1)}$
  $\Delta_{\boldsymbol{W}^{(2)}} \leftarrow \frac{1}{m}\hat{\boldsymbol{H}}^{(1)\top}\hat{\boldsymbol{H}}^{(2)}$
  for $l = 1$ to k (Gibbs sampling) **do**
    Gibbs block 1:
    $\forall i, j, \tilde{V}_{i,j}$ sampled from $P(\tilde{V}_{i,j} = 1) = \sigma\left(\boldsymbol{W}^{(1)}_{j,:}\left(\tilde{\boldsymbol{H}}^{(1)}_{i,:}\right)^{\top}\right)$.
    $\forall i, j, \tilde{H}^{(2)}_{i,j}$ sampled from $P(\tilde{H}^{(2)}_{i,j} = 1) = \sigma\left(\tilde{\boldsymbol{H}}^{(1)}_{i,:}\boldsymbol{W}^{(2)}_{:,j}\right)$.
    Gibbs block 2:
    $\forall i, j, \tilde{H}^{(1)}_{i,j}$ sampled from $P(\tilde{H}^{(1)}_{i,j} = 1) = \sigma\left(\tilde{\boldsymbol{V}}_{i,:}\boldsymbol{W}^{(1)}_{:,j} + \tilde{\boldsymbol{H}}^{(2)}_{i,:}\boldsymbol{W}^{(2)\top}_{j,:}\right)$.
  end for
  $\Delta_{\boldsymbol{W}^{(1)}} \leftarrow \Delta_{\boldsymbol{W}^{(1)}} - \frac{1}{m}\boldsymbol{V}^{\top}\tilde{\boldsymbol{H}}^{(1)}$
  $\Delta_{\boldsymbol{W}^{(2)}} \leftarrow \Delta_{\boldsymbol{W}^{(2)}} - \frac{1}{m}\tilde{\boldsymbol{H}}^{(1)\top}\tilde{\boldsymbol{H}}^{(2)}$
  $\boldsymbol{W}^{(1)} \leftarrow \boldsymbol{W}^{(1)} + \epsilon\Delta_{\boldsymbol{W}^{(1)}}$ (this is a cartoon illustration, in practice use a more effective algorithm, such as momentum with a decaying learning rate)
  $\boldsymbol{W}^{(2)} \leftarrow \boldsymbol{W}^{(2)} + \epsilon\Delta_{\boldsymbol{W}^{(2)}}$
end while

Thanks!

Questions?