机器学习
第9节

涉及知识点：
无监督学习简介、k-means聚类、主成分分析、混合高斯模型的EM算法、一般情况下的EM算法、生成对抗网络、限制玻尔兹曼机简介、限制玻尔兹曼机学习算法、深度信念网络、自编码器

无监督学习

张伟楠 – 上海交通大学

无监督学习简介

张伟楠 – 上海交通大学

数据科学

数学上

- 找到联合数据分布$p(x)$
- 找到条件分布$p(x_2|x_1)$

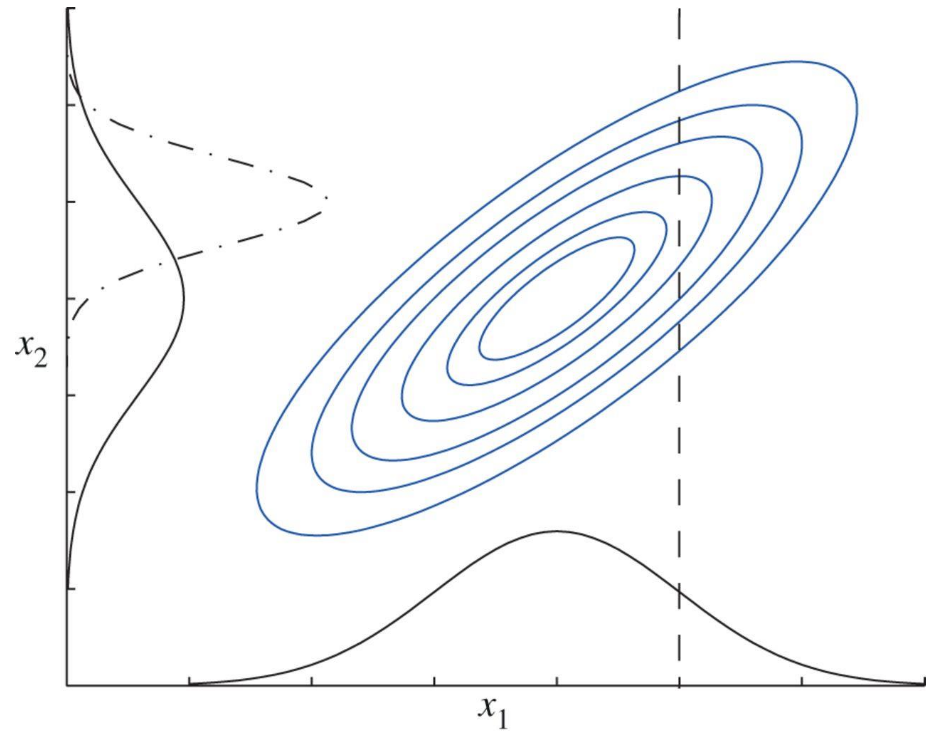
高斯分布

- 多元分布

$$p(x) = \frac{e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}}{\sqrt{|2\pi\Sigma|}}$$

- 一元分布

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



一个用户行为建模的简单例子

Interest	Gender	Age	BBC Sports	PubMed	Bloomberg Business	Spotify
Finance	Male	29	Yes	No	Yes	No
Sports	Male	21	Yes	No	No	Yes
Medicine	Female	32	No	Yes	No	No
Music	Female	25	No	No	No	Yes
Medicine	Male	40	Yes	Yes	Yes	No

- 联合数据分布

  p(Interest=Finance, Gender=Male, Age=29, Browsing=BBC Sports, Bloomberg Business)

- 条件数据分布

  p(Interest=Finance | Browsing=BBC Sports,Bloomberg Business)

  p(Gender=Male | Browsing=BBC Sports,Bloomberg Business)

问题设置

- 首先构建和学习$p(x)$，然后推断条件依赖$p(x_t|x_i)$
  - 无监督学习
  - x的每个维度都是平等对待的
- 直接学习条件依赖$p(x_t|x_i)$
  - 监督学习
  - x_t是要预测的标签

无监督学习（Unsupervised Learning）

定义

- 给定训练数据集

$$D = \{x_i\}_{i=1,2,\dots,N}$$

- 机器学习数据潜在的模式
  - 隐变量

$$z \rightarrow x$$

  - 概率密度函数估计

$$p(x)$$

  - 好的数据表示（用于分类）

$$\phi(x)$$

无监督学习的应用

- 数据结构发现，数据科学
- 数据压缩
- 异常值检测
- 新数据生成
- 学习输入到监督学习/强化学习算法的表示（使原因可能更简单地与输出或奖励相关）
- 生物学习和感知理论

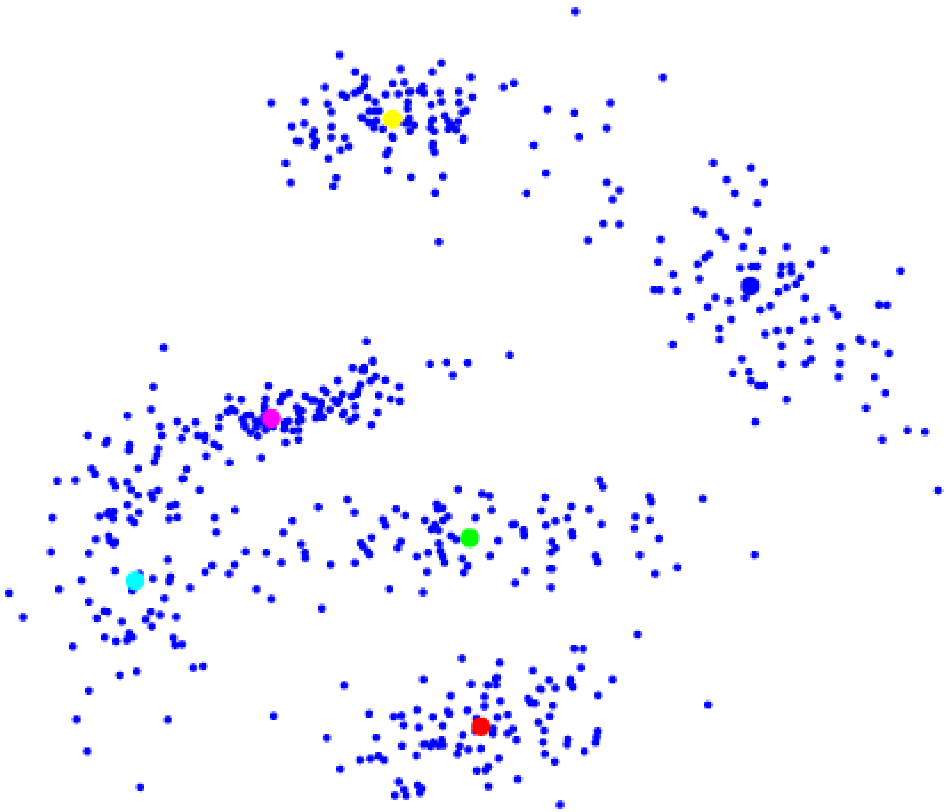
无监督学习的内容

- 无监督学习的基本原理
  - k-means聚类
  - 主成分分析
- 概率无监督学习
  - 混合高斯模型
  - EM算法
- 深度无监督学习
  - 自编码器
  - 生成对抗网络

k-means聚类

张伟楠 – 上海交通大学

k-means聚类（K-means Clustering）



k-means聚类（K-means Clustering）

- 提供所需簇（cluster）的数量k
- 随机选择k个实例作为种子节点，即作为每个簇的质心（centroid）
- 迭代以下步骤
  - 将每个实例分配给最近质心相关联的簇
  - 重新估计每个簇的质心
- 当聚类收敛时停止
  - 或者在经过固定次数的迭代之后

Slide credit: Ray Mooney

k-means聚类：质心

- 假设实例是实值向量

$$x \in \mathbb{R}^d$$

- 簇基于质心，重心或簇C_k中的点的平均值

$$\mu^k = \frac{1}{C_k} \sum_{x \in C_k} x$$

Slide credit: Ray Mooney

k-means聚类：距离

- 到质心的距离$L(x, \mu^k)$
- 欧几里得距离（L2范数）

$$L_2(x, \mu^k) = \| x - \mu^k \| = \sqrt{\sum_{m=1}^{d} (x_m - \mu_m^k)^2}$$

- 欧几里得距离（L1范数）

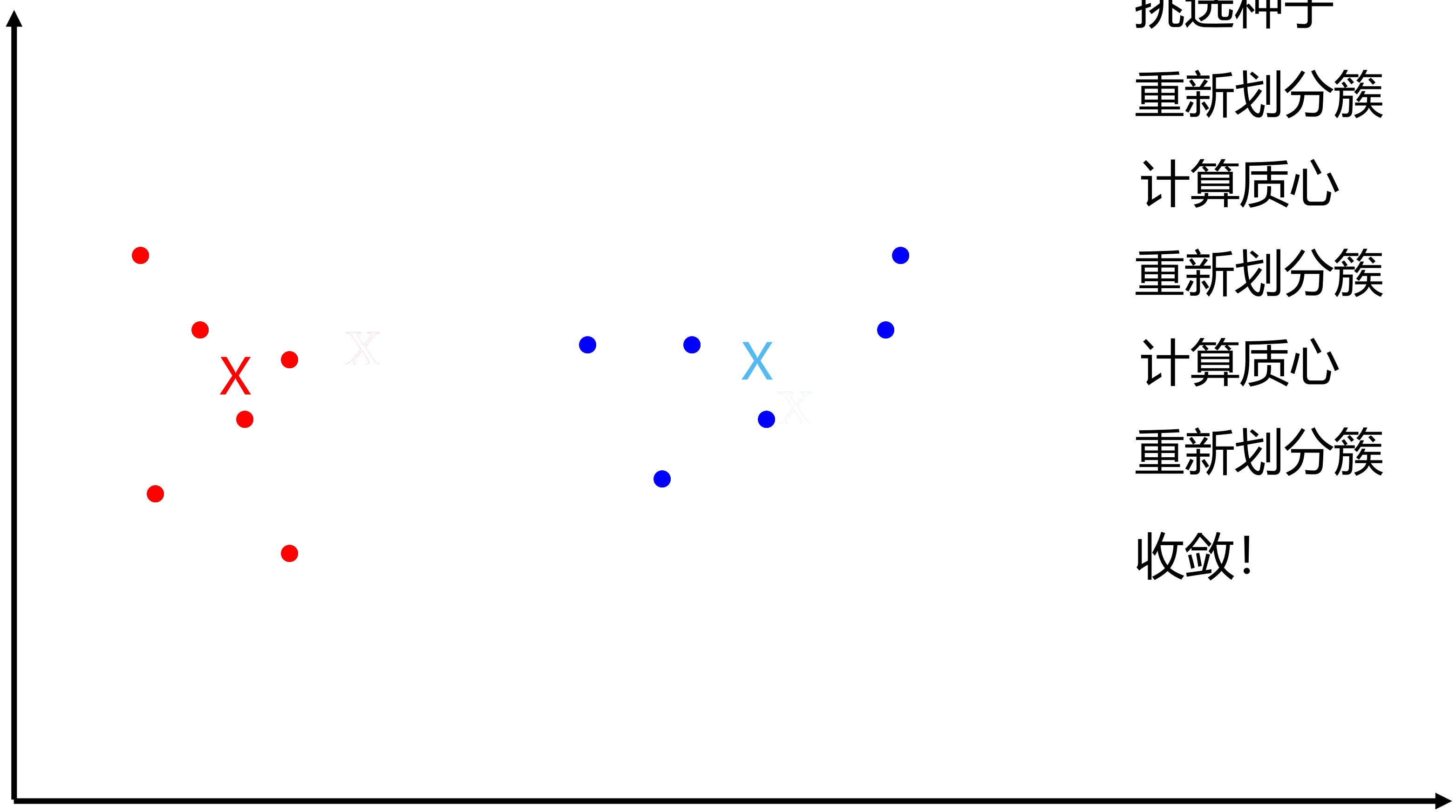
$$L_1(x, \mu^k) = |x - \mu^k| = \sum_{m=1}^{d} |x_m - \mu_m^k|$$

- 余弦距离

$$L_{\cos}(x, \mu^k) = 1 - \frac{x^\top \mu^k}{|x| \cdot |\mu^k|}$$

Slide credit: Ray Mooney

k-means聚类例子



挑选种子

重新划分簇

计算质心

重新划分簇

计算质心

重新划分簇

收敛!

k-means聚类时间复杂度

- 假设两个实例之间的计算距离是$O(d)$，其中d是向量的维数
- 重新分配簇：$O(knd)$距离计算
- 计算质心：每个实例向量添加到某个质心一次：$O(nd)$
- 假设这两个步骤分别迭代l次：$O(lknd)$

k-means聚类目标

- 最小化每个点与其对应的聚类质心的平方距离的总和

$$\min_{\{\mu^k\}_{k=1}^{K}} \sum_{k=1}^{K} \sum_{x \in C_k} L(x - \mu^k) \qquad \mu^k = \frac{1}{C_k} \sum_{x \in C_k} x$$

- 找到全局最优是NP-hard的
- K-means算法保证收敛到局部最优

选择种子

- 结果可以根据随机种子选择而变化
- 一些种子可能导致比较差的收敛速度，或者收敛到次优簇
- 使用启发式或其他方法的结果选择好的种子

聚类应用

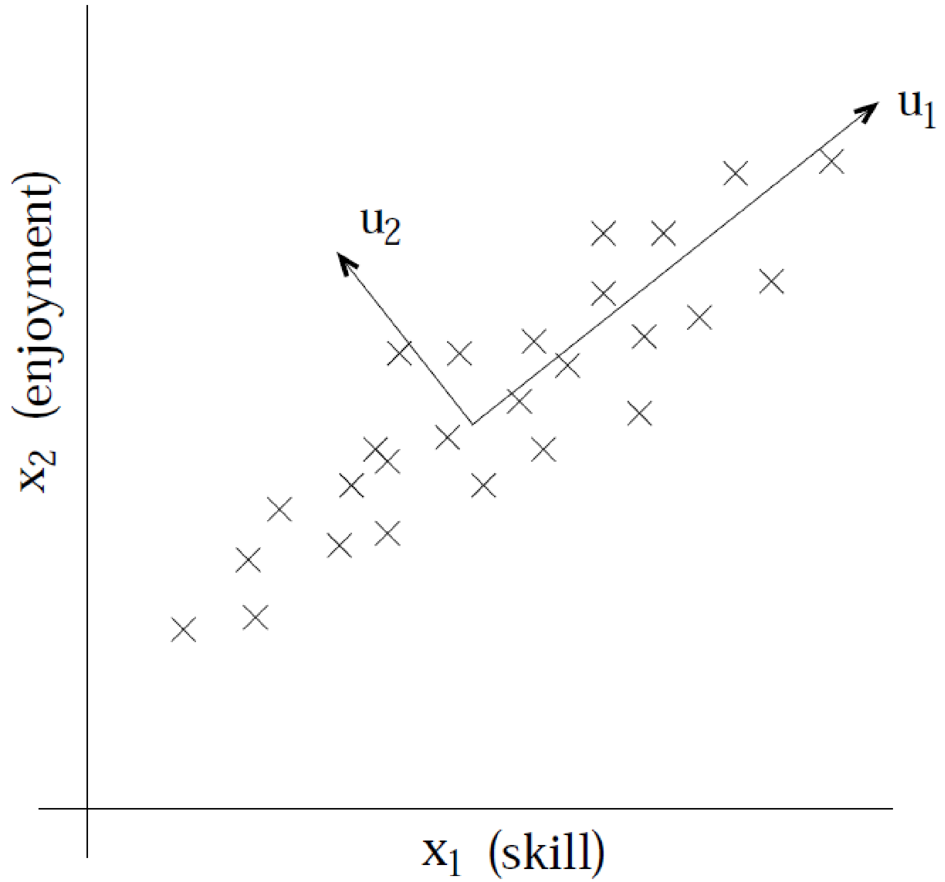
- 文本挖掘
  - 用于相关搜索的文档簇
  - 查询建议的词簇
- 推荐系统和广告
  - 用户簇的物品/广告推荐
  - 相关物品建议的项目簇
- 图片搜索
  - 用于类似图像搜索和重复检测的图像簇
- 语音识别或分离
  - 语音簇功能

主成分分析

张伟楠 – 上海交通大学

主成分分析（Principal Component Analysis）

- 二维数据的示例
  - x_1：飞行员的驾驶技巧
  - x_2：他/她有多喜欢飞行
- 主成分
  - u_1：一个人作为飞行员的内在相关因素
  - u_2：一些噪音



Example credit: Andrew Ng

主成分分析（PCA）

- PCA试图识别数据所在的子空间

- PCA使用正交变换将可能相关的变量的观测值转换为称为主成分的线性不相关变量的一组值
  - 主成分的数量小于或等于原始变量的数量/观察值数量中的较小者

$$\mathbb{R}^d \rightarrow \mathbb{R}^k \qquad k \ll d$$

PCA数据预处理

- 给定数据集

$$D = \{x^{(i)}\}_{i=1}^{m}$$

- 通常我们先预处理数据，使其均值和方差规范化

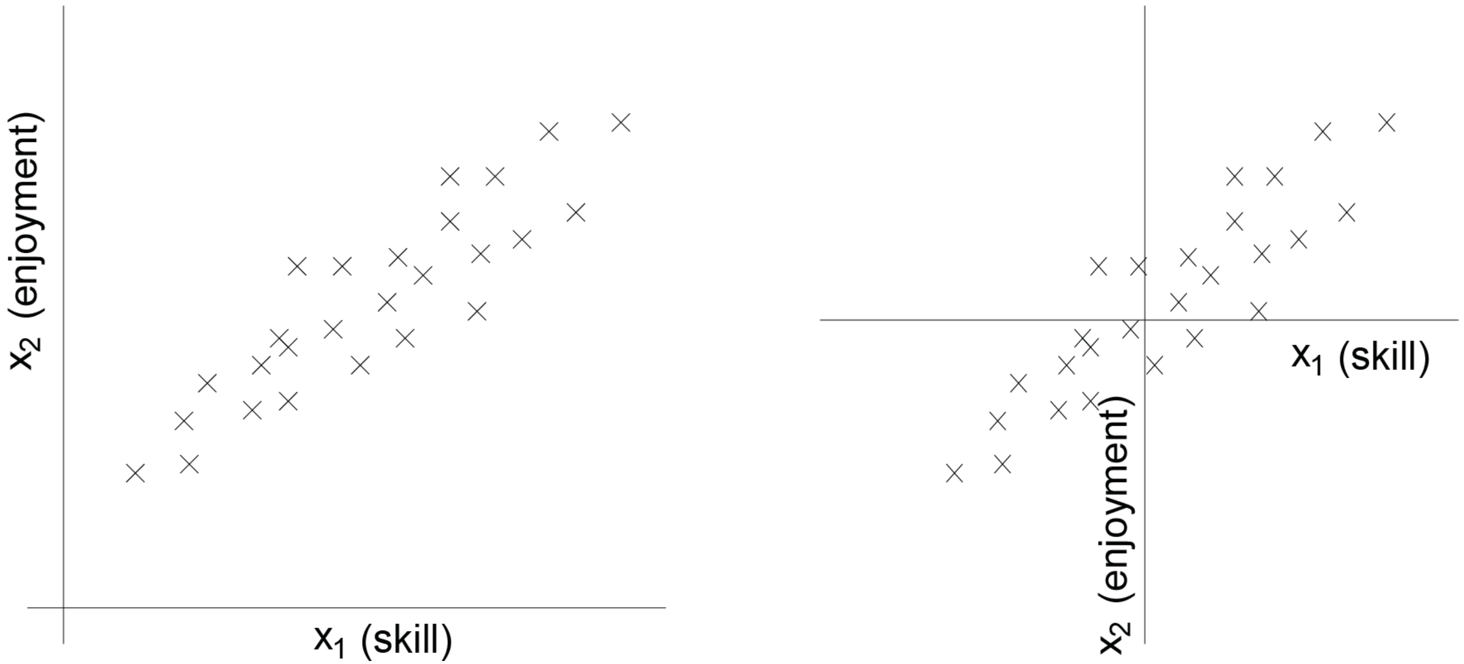
1. 将数据集的中心点移动到 0

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \qquad x^{(i)} \leftarrow x^{(i)} - \mu$$

2. 统一每个变量的方差（可选）

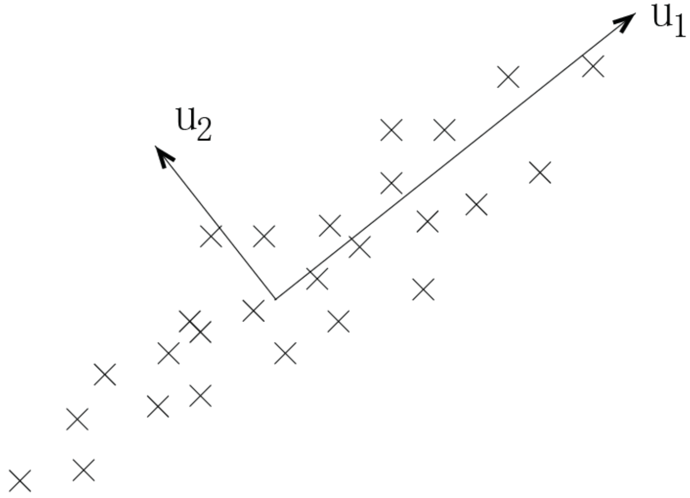
$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)}\right)^2 \qquad x_j^{(i)} \leftarrow x_j^{(i)}/\sigma_j$$

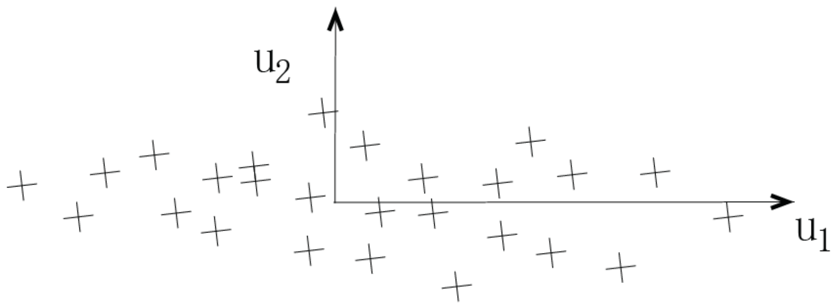
PCA数据预处理



- 将数据的均值归零
- 将每个坐标重新缩放使其具有单位方差，从而确保在相同的“比例”上处理不同的属性（可选）

PCA解决方案





- PCA找到具有最大变量方差的方向
  - 其对应于具有最大特征值的矩阵X^TX的特征向量

PCA解决方案：数据投影

- 每个点$x^{(i)}$到方向u的投影($\| u \|= 1$)

$$x^{(i)^\top} u$$

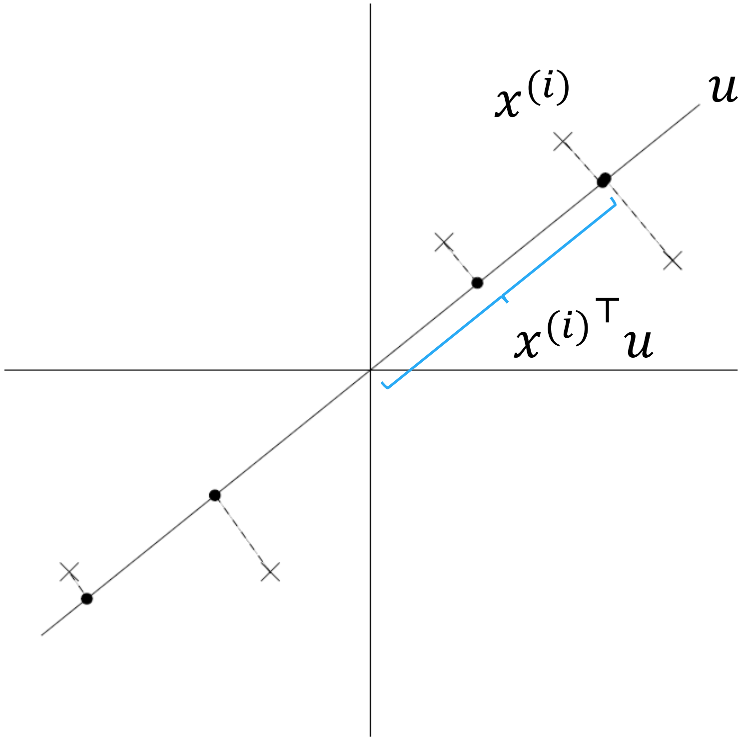
- 投影的方差

$$\frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)^\top} u\right)^2 = \frac{1}{m}\sum_{i=1}^{m} u^\top x^{(i)} x^{(i)^\top} u$$

$$= u^\top \left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)} x^{(i)^\top}\right) u$$

$$\equiv u^\top \Sigma\, u$$

Σ：Covariance matrix

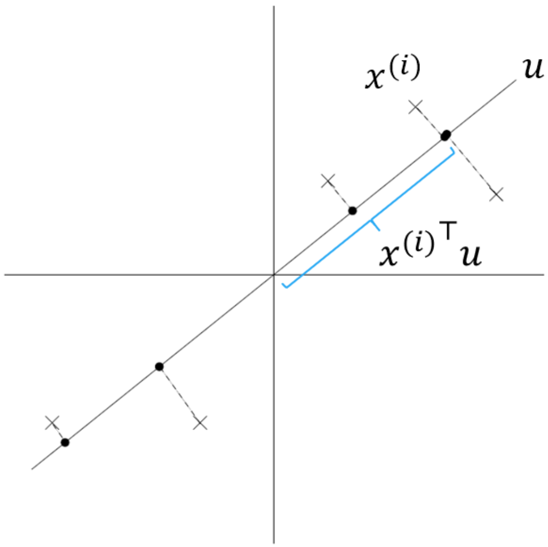


PCA解决方案：最大特征值

$$\max_{u} \quad u^\top \Sigma u$$
$$\text{s.t.} \quad \| u \| = 1$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} x^{(i)\top}$$

- 找到数据的k个主成分就是找到Σ的k个主特征向量
  - 即具有最大特征值的前k个特征向量



- $x^{(i)}$的投影向量

$$y^{(i)} = \begin{bmatrix} u_1^\top x^{(i)} \\ u_2^\top x^{(i)} \\ \vdots \\ u_k^\top x^{(i)} \end{bmatrix} \in \mathbb{R}^k$$

特征分解回顾

- 对于半正定方阵$\Sigma_{d\times d}$
  - 假设u是其特征向量($\| u \|= 1$)
  - 特征值w $\quad\quad \Sigma u = wu$
  - 有d个特征向量-特征值对(u_i, w_i)
  - d个特征向量是正交的，因此它们组成标准正交基$\sum_{i=1}^{d} u_i u_i^\top = I$
  - 因此任何向量v都可以写成

$$v = (\sum_{i=1}^{d} u_i u_i^\top) v = \sum_{i=1}^{d} (u_i^\top v) u_i = \sum_{i=1}^{d} v_{(i)} u_i$$

  - $\Sigma_{d\times d}$可以写成

$$U = [u_1, u_2, \dots, u_d]$$

$$\Sigma = \sum_{i=1}^{d} u_i u_i^\top \Sigma = \sum_{i=1}^{d} w_i u_i u_i^\top = UWU^\top \qquad W = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & w_d \end{bmatrix}$$

特征分解回顾

- 给定数据 $X=\begin{bmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_n^\top \end{bmatrix}$ 及其协方差矩阵$\Sigma = X^\top X$　（简单起见省去m）
- 方向u_i上的方差为

$$\| Xu_i \|^2 = u_i^\top X^\top X u_i = u_i^\top \Sigma u_i = u_i^\top w_i u_i = w_i$$

- 方向v的方差为

$$\| Xv \|^2 \quad = \| X(\sum_{i=1}^{d} v_{(i)} u_i) \|^2 = \sum_{ij} v_{(i)} u_i^\top \Sigma u_j v_{(j)} = \sum_{i=1}^{d} v_{(i)}^2 w_i$$

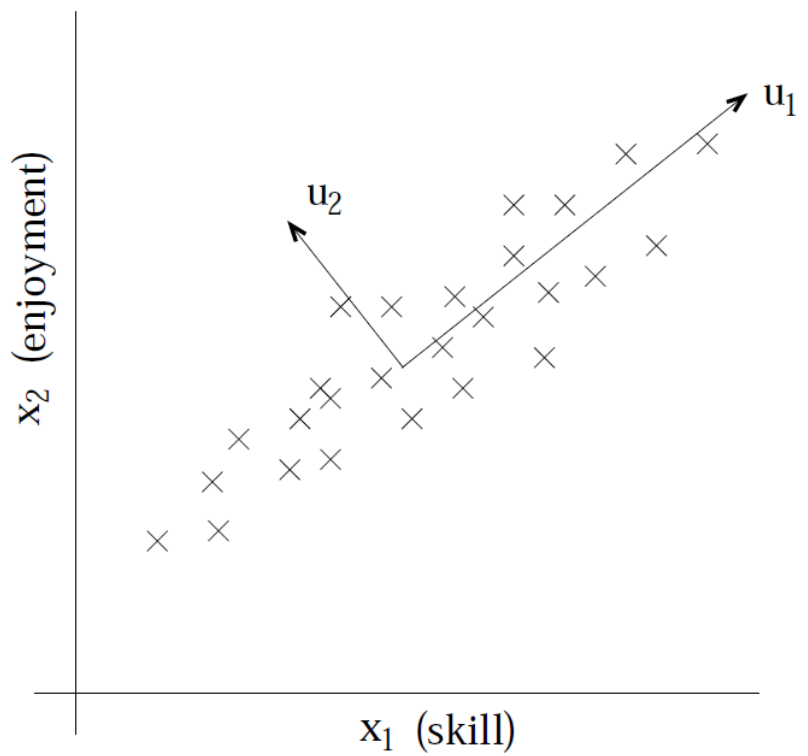
其中$v_{(i)}$是v在u_i上的投影长度

- 如果$v^\top v = 1$，那么$\arg\max_{\|v\|=1} \| Xv \|^2 = u_{(\max)}$

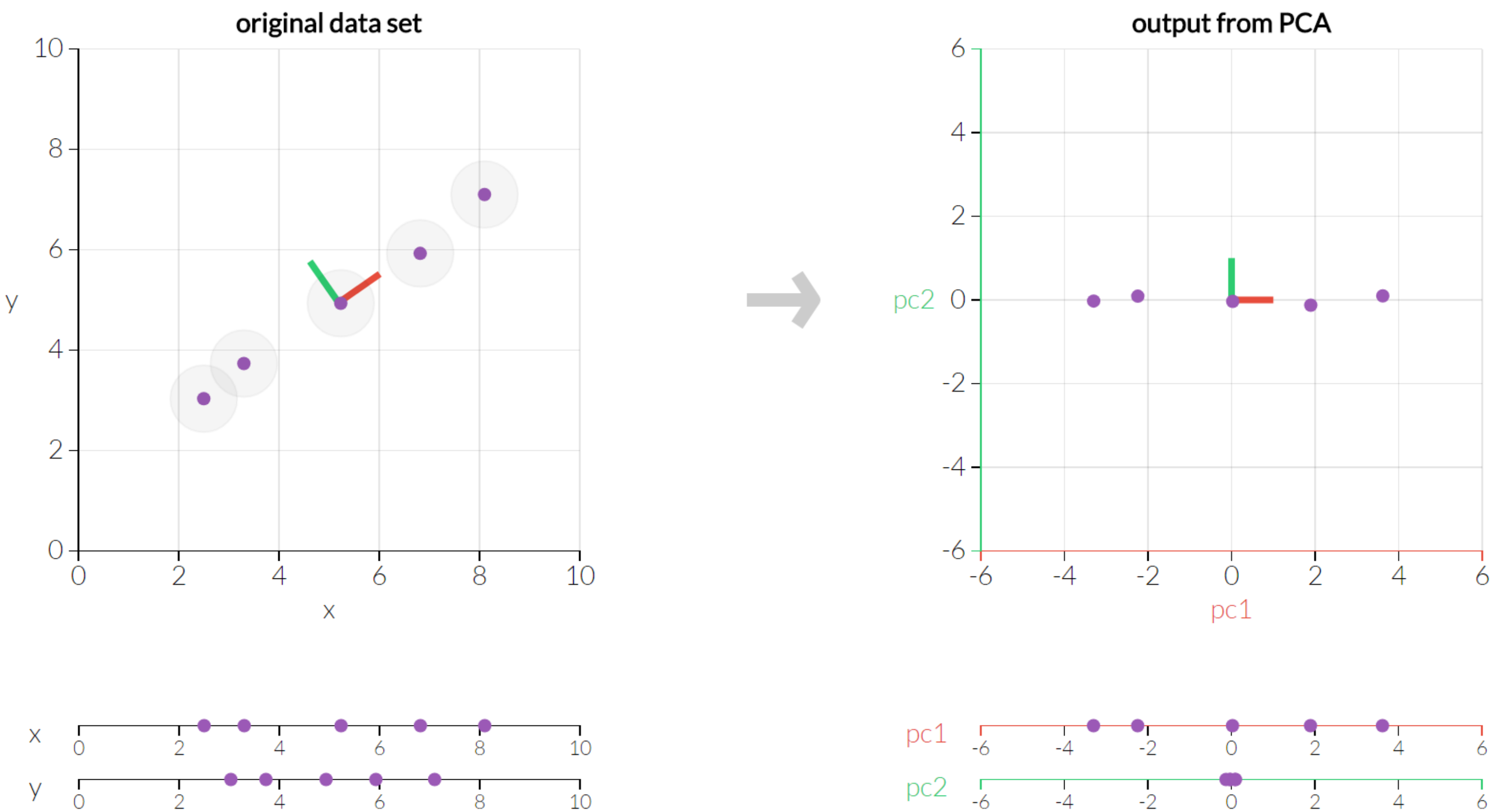
最大方差的方向是具有最大特征值的特征向量

PCA讨论

- PCA还可以通过选择能够最小化将数据投影到由k维子空间所产生的近似误差的基来得到

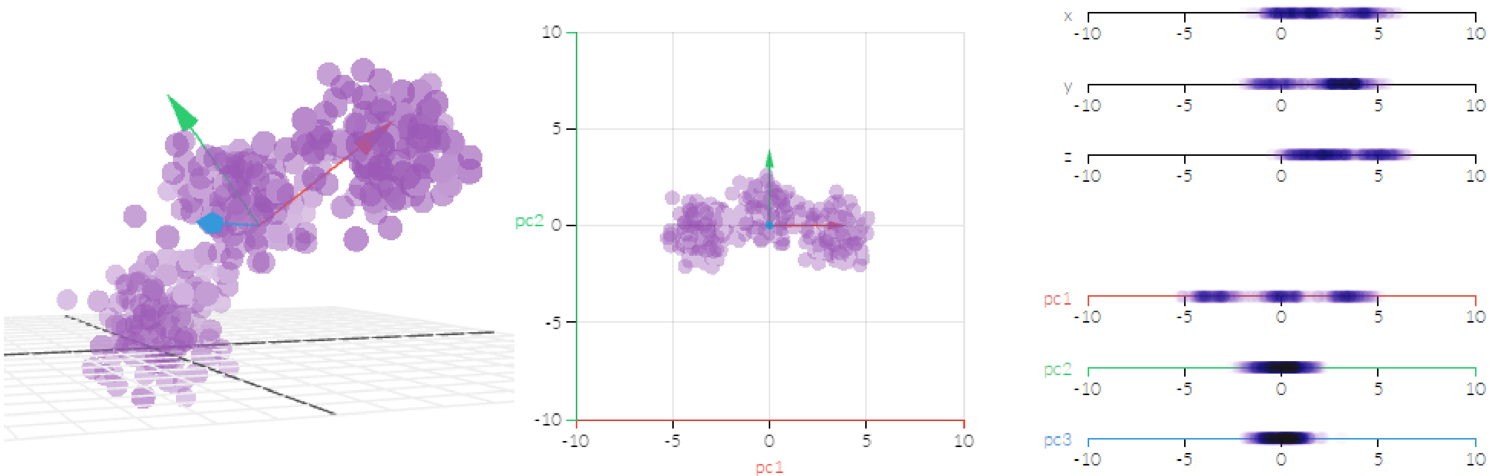


PCA可视化



http://setosa.io/ev/principal-component-analysis/

PCA可视化

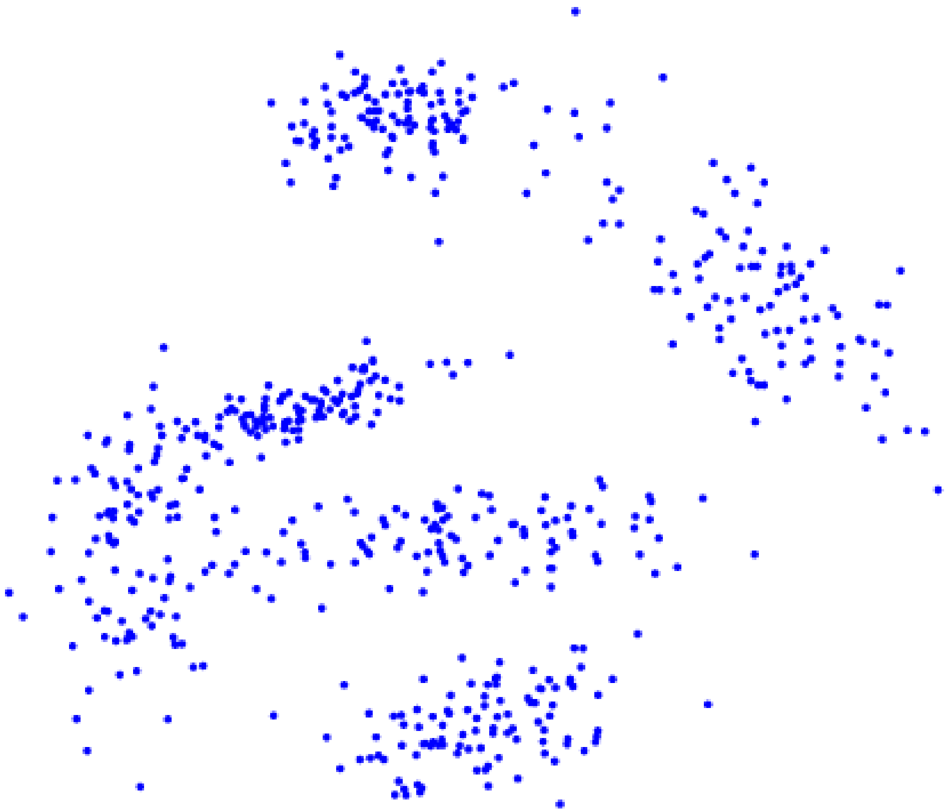


http://setosa.io/ev/principal-component-analysis/

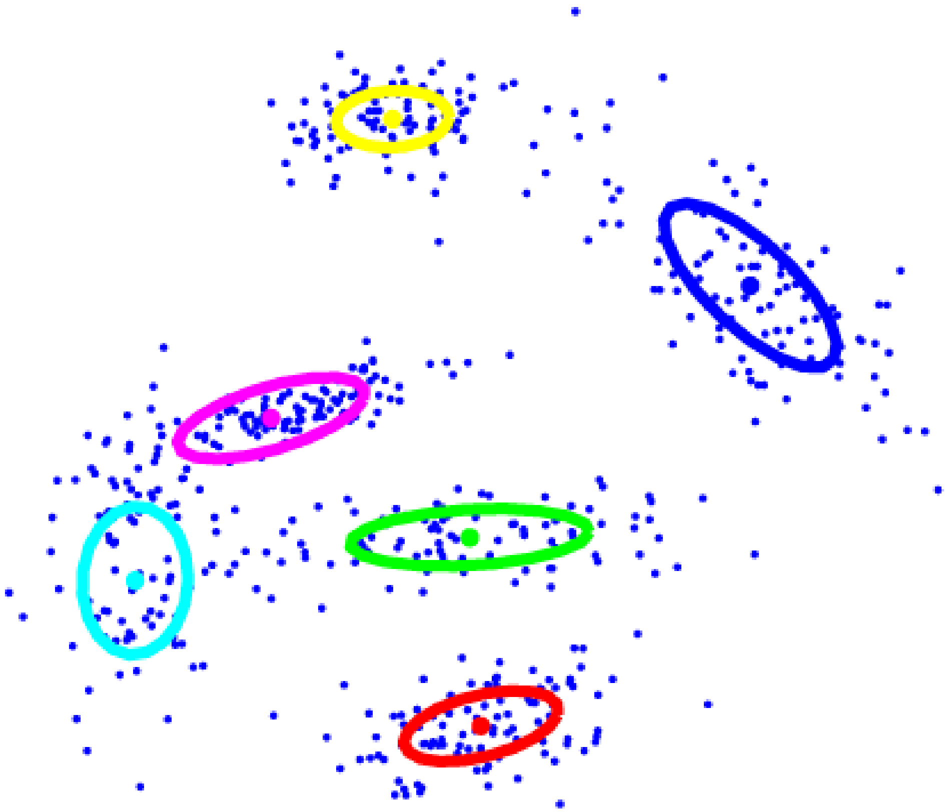
混合高斯模型的EM算法

张伟楠 – 上海交通大学

混合高斯



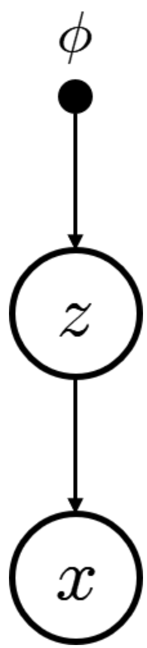
混合高斯



混合高斯图模型

- 给定训练集$\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$
- 通过具体化数据的联合分布对数据进行建模

$$p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)})$$



隐变量分布的参数

$z^{(i)} \sim \mathrm{Multinomial}(\phi)$

$p\left(z^{(i)} = j\right) = \phi_j$

隐变量：高斯聚类的ID
表明每个x来自哪个高斯分布

$x^{(i)} \sim \mathcal{N}(\mu_j, \Sigma_j)$

观测到的数据点

数据似然性

- 我们希望最大化

$$
\begin{aligned}
l(\phi,\mu,\Sigma) &= \sum_{i=1}^{m} \log p\left(x^{(i)};\phi,\mu,\Sigma\right) \\
&= \sum_{i=1}^{m} \log \sum_{z^{(i)}=1}^{k} p\left(x^{(i)} \middle| z^{(i)};\mu,\Sigma\right) p\left(z^{(i)};\phi\right) \\
&= \sum_{i=1}^{m} \log \sum_{j=1}^{k} \mathcal{N}\left(x^{(i)} \middle| \mu_j,\Sigma_j\right)\phi_j
\end{aligned}
$$

- 如果只令

$$
\frac{\partial l(\phi,\mu,\Sigma)}{\partial \phi} = 0 \qquad \frac{\partial l(\phi,\mu,\Sigma)}{\partial \mu} = 0 \qquad \frac{\partial l(\phi,\mu,\Sigma)}{\partial \Sigma} = 0
$$

求不到显式解

最大化数据似然性

- 对每个数据点$x^{(i)}$，隐变量$z^{(i)}$表明它来自哪个高斯分布
- 如果我们知道$z^{(i)}$，那么数据似然可以被可化为

$$l(\phi,\mu,\Sigma)=\sum_{i=1}^{m}\log p\left(x^{(i)},z^{(i)};\phi,\mu,\Sigma\right)$$

$$=\sum_{i=1}^{m}\log p\left(x^{(i)}\middle|z^{(i)};\mu,\Sigma\right)p\left(z^{(i)};\phi\right)$$

$$=\sum_{i=1}^{m}\log\mathcal{N}\left(x^{(i)}\middle|\mu_{z^{(i)}},\Sigma_{z^{(i)}}\right)+\log p\left(z^{(i)};\phi\right)$$

最大化数据似然性

- 给定$z^{(i)}$，最大化数据似然

$$\max_{\phi,\mu,\Sigma} l(\phi,\mu,\Sigma)=\max_{\phi,\mu,\Sigma}\sum_{i=1}^{m}\log\mathcal{N}\left(x^{(i)}\middle|\mu_{z^{(i)}},\Sigma_{z^{(i)}}\right)\ +\log p\left(z^{(i)};\phi\right)$$

- 很容易求得解

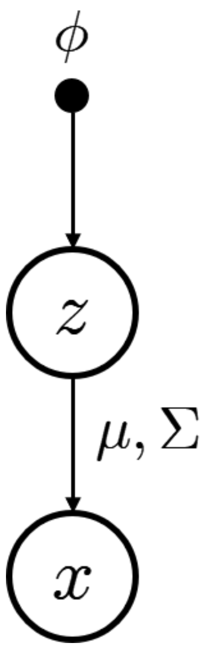
$$\phi_j=\frac{1}{m}\sum_{i=1}^{m}1\left\{z^{(i)}=j\right\}$$

$$\mu_j=\frac{\sum_{i=1}^{m}1\left\{z^{(i)}=j\right\}x^{(i)}}{\sum_{i=1}^{m}1\left\{z^{(i)}=j\right\}}$$

$$\Sigma_j=\frac{\sum_{i=1}^{m}1\left\{z^{(i)}=j\right\}\left(x^{(i)}-\mu_j\right)\left(x^{(i)}-\mu_j\right)^{\mathsf{T}}}{\sum_{i=1}^{m}1\left\{z^{(i)}=j\right\}}$$

隐变量推断

- 给定参数μ, Σ, ϕ，不难推断出每个实例的隐变量$z^{(i)}$的后验分布



$$p\left(z^{(i)}=j \middle| x^{(i)}; \phi, \mu, \Sigma\right)=\frac{p\left(z^{(i)}=j, x^{(i)}; \phi, \mu, \Sigma\right)}{p\left(x^{(i)}; \phi, \mu, \Sigma\right)}$$

$$=\frac{p\left(x^{(i)} \middle| z^{(i)}=j; \mu, \Sigma\right) p\left(z^{(i)}=j; \phi\right)}{\sum_{l=1}^{k} p\left(x^{(i)} \middle| z^{(i)}=l; \mu, \Sigma\right) p\left(z^{(i)}=l; \phi\right)}$$

其中

- $z^{(i)}$的先验为$p(z^{(i)}=j; \phi)$
- 似然性为$p(x^{(i)} | z^{(i)}=j; \mu, \Sigma)$

- 然后基于我们对$z^{(i)}$的猜测更新参数μ, Σ, ϕ

最大期望算法（Expectation Maximization）

- E-步骤：在给定模型参数的情况下推断隐变量的后验分布
- M-步骤：调整参数使在给定隐变量后验分布的情况下最大化数据似然

- EM算法
  - 迭代地执行E-步骤和M-步骤直到收敛

混合高斯的EM算法

- 混合高斯例子

重复直到收敛 {

{E-step} 对每个i、j，设置$w_j^{(i)} = p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma)$

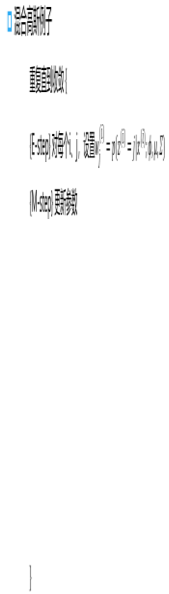
{M-step} 更新参数

$$\phi_j = \frac{1}{m} \sum_{i=1}^{m} w_j^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{m} w_j^{(i)} \left(x^{(i)} - \mu_j\right)\left(x^{(i)} - \mu_j\right)^{\mathrm{T}}}{\sum_{i=1}^{m} w_j^{(i)}}$$

}



一般情况下的EM算法

张伟楠 – 上海交通大学

一般情况下的EM算法

- EM算法性质：
  - 在每个EM步骤之后，数据的似然是不会降低的
  - EM算法寻找隐变量模型似然的一个（局部）最大值

- 现在让我们讨论一般情况下的EM算法，并且验证其提高数据似然及其收敛的有效性

延森不等式

定理：设f为凸函数，X为随机变量。那么

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$$

- 如果f是严格凸的，那么要满足

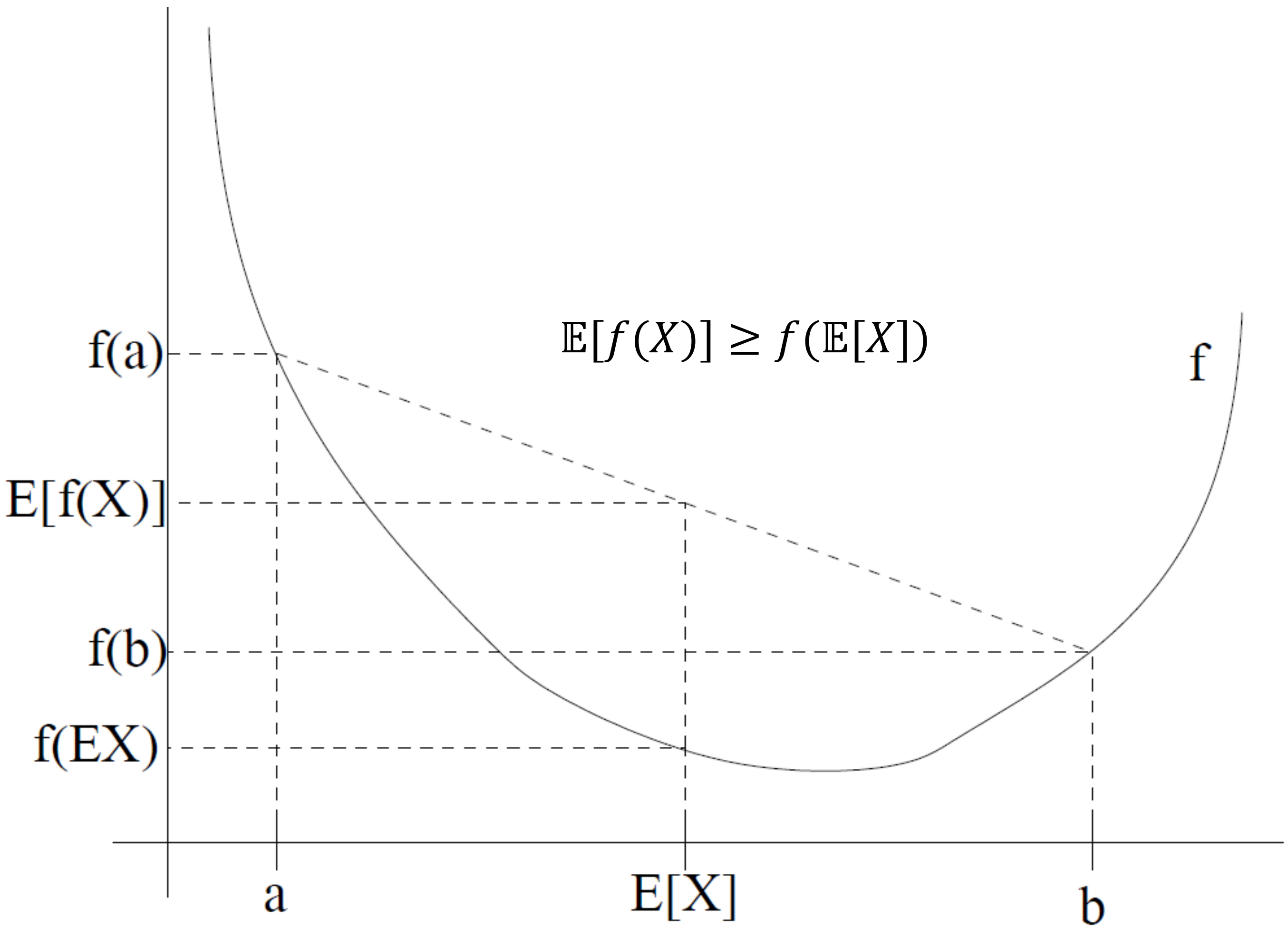
$$\mathbb{E}[f(X)] = f(\mathbb{E}[X])$$

当且仅当以概率1满足

$$X = \mathbb{E}[X]$$

（即，如果X是常量）

延森不等式



延森不等式

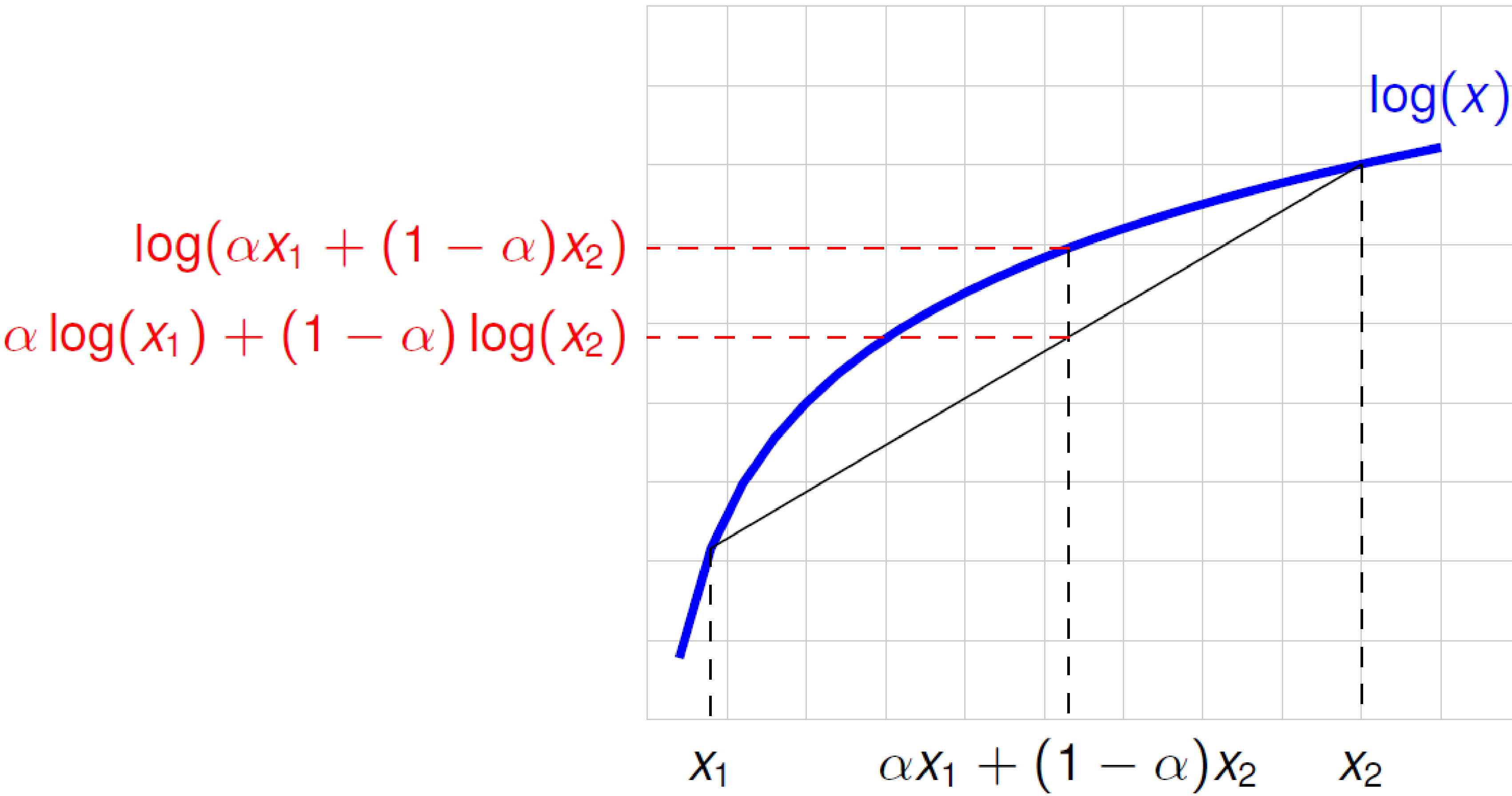


Figure credit: Maneesh Sahani

一般情况下的EM算法：问题

- 给定训练数据集$D=\{x_i\}_{i=1,2,\ldots,N}$，让机器学习数据的潜在模式
- 假定隐变量

$$z \rightarrow x$$

- 我们希望为数据拟合模型$p(x,z)$的参数，其中对数似然为

$$\begin{aligned} l(\theta) &= \sum_{i=1}^{N} \log p(x;\theta) \\ &= \sum_{i=1}^{N} \log \sum_{z} p(x,z;\theta) \end{aligned}$$

一般情况下的EM算法：问题

- EM算法解决如下问题
  - 显式找到最大似然估计（MLE）很困难

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{N} \log \sum_{z^{(i)}} p\left(x^{(i)}, z^{(i)}; \theta\right)$$

  - 但是如果$z^{(i)}$可以观测，MLE很简单

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(x^{(i)} | z^{(i)}; \theta)$$

- EM算法为MLE提供了一个有效的解决方案，迭代执行以下两步

E-步骤：构造对数似然的（良好）下界

M-步骤：优化下界

一般情况下的EM算法：下界

- 对每个实例i，让q_i变成$z^{(i)}$

$$\sum_z q_i(z) = 1, q_i(z) \geq 0$$

- 数据对数似然

$$\begin{aligned}
l(\theta) = \sum_{i=1}^{N} \log p\left(x^{(i)}; \theta\right) &= \sum_{i=1}^{N} \log \sum_{z^{(i)}} p\left(x^{(i)}, z^{(i)}; \theta\right) \\
&= \sum_{i=1}^{N} \log \sum_{z^{(i)}} q_i(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta)}{q_i(z^{(i)})} \\
&\geq \sum_{i=1}^{N} \sum_{z^{(i)}} q_i\left(z^{(i)}\right) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{q_i(z^{(i)})}
\end{aligned}$$

$l(\theta)$的下界

延森不等式 $-\log(x)$是凸函数

一般情况下的EM算法：下界

$$l(\theta)=\sum_{i=1}^{N}\log p\left(x^{(i)};\theta\right)\geq\sum_{i=1}^{N}\sum_{z^{(i)}}q_i\left(z^{(i)}\right)\log\frac{p\left(x^{(i)},z^{(i)};\theta\right)}{q_i\left(z^{(i)}\right)}$$

- 我们应该选择什么样的$q_i(z)$?
- 为了使上述不等式满足紧约束（等号成立），需要使其满足

$$p(x^{(i)},z^{(i)};\theta)=q_i(z^{(i)})\cdot c$$

- 我们可以推出

$$\log p\left(x^{(i)};\theta\right)=\log\sum_{z^{(i)}}p\left(x^{(i)},z^{(i)};\theta\right)=\log\sum_{z^{(i)}}q_i\left(z^{(i)}\right)c=\sum_{z^{(i)}}q_i(z^{(i)})\log\frac{p(x^{(i)},z^{(i)};\theta)}{q_i(z^{(i)})}$$

- 因此，$q_i(z)$可以写成后验分布

$$q_i(z^{(i)})=\frac{p(x^{(i)},z^{(i)};\theta)}{\sum_z p(x^{(i)},z;\theta)}=\frac{p(x^{(i)},z^{(i)};\theta)}{p(x^{(i)};\theta)}=p(z^{(i)}|x^{(i)};\theta)$$

一般情况下的EM算法

重复直到收敛 {

{E-step} 对每个i，设置

$$q_i(z^{(i)}) = p(z^{(i)}|x^{(i)};\theta)$$

{M-step} 更新参数

$$\theta = \arg\max_{\theta} \sum_{i=1}^{N} \sum_{z^{(i)}} q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)};\theta)}{q_i(z^{(i)})}$$

}

EM的收敛性

- 用$\theta^{(t)}$和$\theta^{(t+1)}$表示EM算法两次连续迭代的参数，我们需要证明

$$l(\theta^{(t)}) \leq l(\theta^{(t+1)})$$

这表明EM算法使得对数似然始终是单调递增的，从而保证EM算法至少会收敛到局部最优解

EM收敛性证明

- 从$\theta^{(t)}$开始，我们选择隐变量的后验分布

$$q_i^{(t)}\left(z^{(i)}\right) = p\left(z^{(i)} \middle| x^{(i)}; \theta^{(t)}\right)$$

- 这种选择保证了延森不等式取等号

$$l\left(\theta^{(t)}\right) = \sum_{i=1}^{N} \log \sum_{z^{(i)}} q_i^{(t)}(z^{(i)}) \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{q_i^{(t)}(z^{(i)})} = \sum_{i=1}^{N} \sum_{z^{(i)}} q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{q_i^{(t)}(z^{(i)})}$$

- 然后参数$\theta^{(t+1)}$通过最大化上述方程的右侧获得

- 因此

$$\begin{aligned} l\left(\theta^{(t+1)}\right) &\geq \sum_{i=1}^{N} \sum_{z^{(i)}} q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t+1)})}{q_i^{(t)}(z^{(i)})} && \text{[下界]} \\ &\geq \sum_{i=1}^{N} \sum_{z^{(i)}} q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{q_i^{(t)}(z^{(i)})} && \text{[参数优化]} \\ &= l\left(\theta^{(t)}\right) \end{aligned}$$

EM收敛性备注

- 如果我们定义

$$J(q,\theta)=\sum_{i=1}^{N}\sum_{z^{(i)}}q_i(z^{(i)})\log\frac{p(x^{(i)},z^{(i)};\theta)}{q_i(z^{(i)})}$$

那么

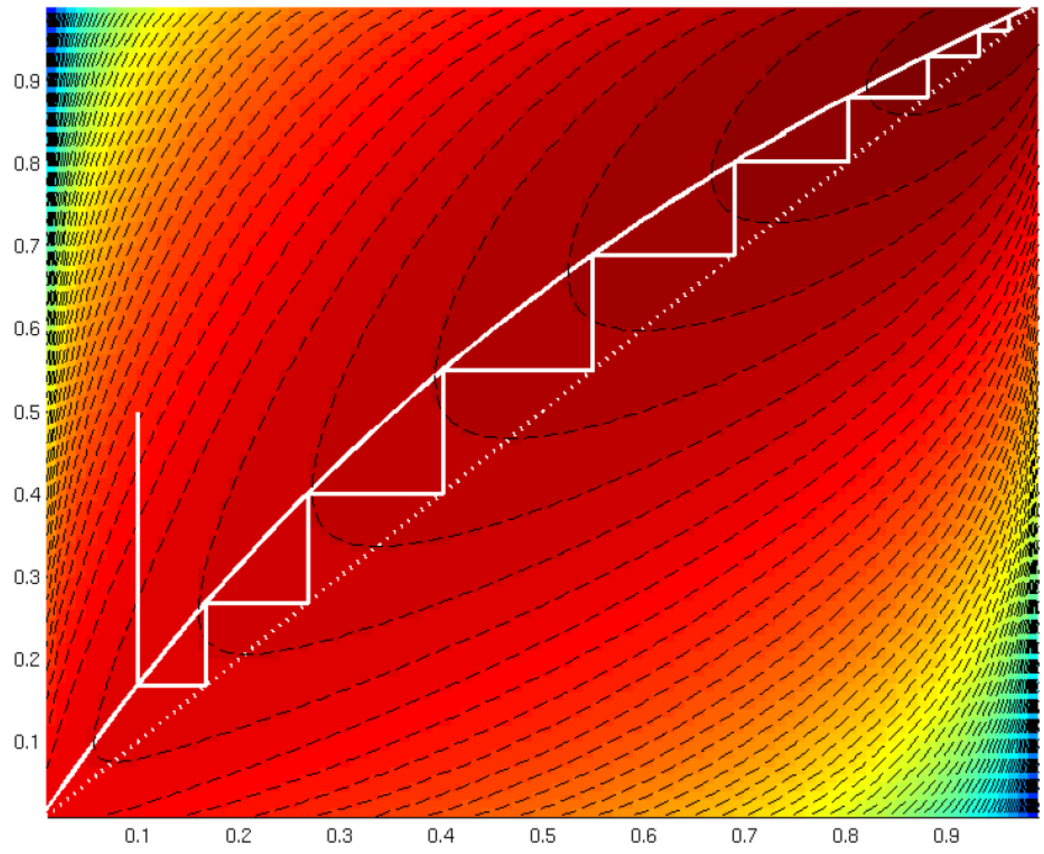
$$l(\theta)\geq J(q,\theta)$$

- EM算法在 J 上坐标上升（coordinate ascent）
  - E-步骤：调整q来最大化J的值（寻找后验分布）
  - M-步骤：调整θ最大化J的值（直接优化）

EM收敛性备注

E-Step

$$q_i^{(t)}(z^{(i)}) = p\left(z^{(i)} \middle| x^{(i)}; \theta^{(t)}\right)$$



$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \sum_{z^{(i)}} q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{q_i^{(t)}(z^{(i)})}$$

M-Step

Figure credit: Maneesh Sahani

总结（传统）无监督学习

- 无监督学习关注从数据中提取结构信息，搭建数据分布，进而对数据的整体概览和变量之间的相互关系达到深入地认知

- 聚类是无监督学习中最常见的任务，其本质就是通过两两数据之间的相似度来建立局部数据之间聚集的模式

- 降维是无监督学习的另一个重要的任务，对高维度数据做出信息损失最小化的变换，将数据的维度从高维降至低维，一方面方便数据可视化，让观察者洞悉数据的分布，另一方面提取了关键的隐空间信息，对数据的预测更加准确

- 隐变量模型通过定义和推断观测数据背后的隐变量分布，以数据生成的方式对数据的结构建模，并以此推断数据点所属的隐层结构信息

- 相比有监督学习，无监督学习任务对数据有更加全盘通透的理解，建模整体数据分布，推断任何标量之间的条件概率分布

生成对抗网络

张伟楠 – 上海交通大学

问题定义：数据生成

- 给定数据集 $D=\{x\}$，搭建数据分布模型 $q_\theta(x)$，使其可以拟合到真实数据分布 $p(x)$

- 传统目标：最大似然估计（Maximum Likelihood Estimation, MLE）

$$\max_\theta \frac{1}{|D|}\sum_{x\in D}\log q_\theta(x) \simeq \max_\theta \mathbb{E}_{x\sim p(x)}[\log q_\theta(x)]$$

  - 检验真实数据是否在学到的模型上具有高质量密度

Goodfellow, Ian, et al. "Generative adversarial nets." In NeuralPS, pp. 2672-2680. 2014.

评估与使用的不一致性

- 给定一个具有一定泛化能力的生成器 q

$$\max_{\theta} \mathbb{E}_{x \sim p(x)}[\log q_{\theta}(x)]$$

训练 / 评估

- 检验真实数据是否在学到的模型上具有高质量密度
- 近似为

$$\max_{\theta} \frac{1}{|D|} \sum_{x \in D} \log q_{\theta}(x)$$

$$\max_{\theta} \mathbb{E}_{x \sim q_{\theta}(x)}[\log p(x)]$$

使用

- 检验模型生成的数据是否尽可能的与真实数据相似
- 更直接，但是直接计算 $p(x)$ 不太现实

生成对抗网络（Generative Adversarial Nets, GANs)

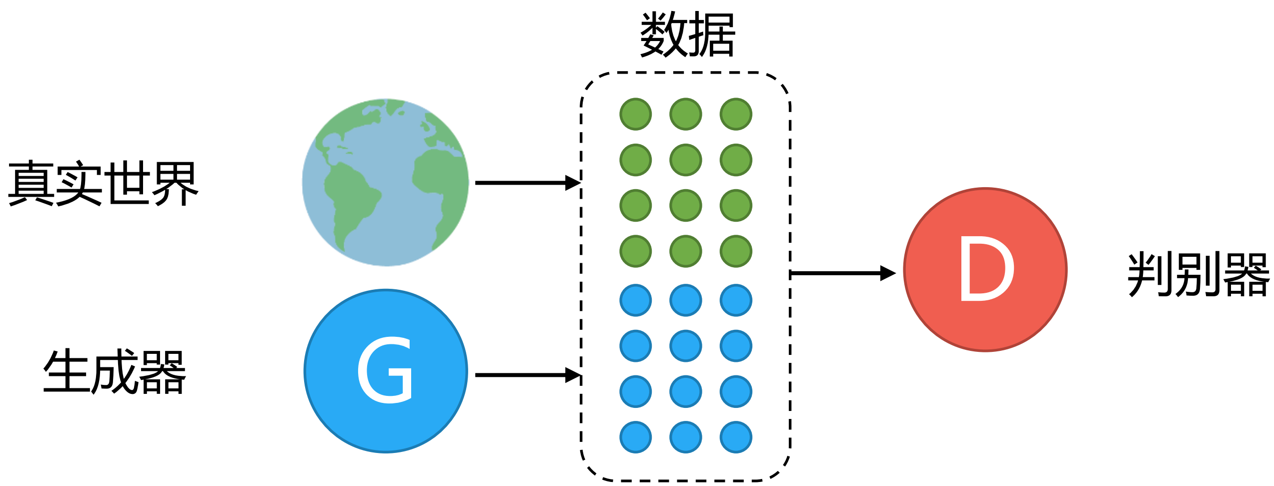
- 对于生成任务，我们真正需要的目标是

$$\max_{\theta} \mathbb{E}_{x\sim q_\theta(x)}[\log p(x)]$$

- 但是我们不能直接计算 $p(x)$

- Idea: 我们可否直接搭建一个判别器来判断一个数据实例是真是的还是假的（智能生成的）？
  - 利用基于深度学习的判别模型的强大能力

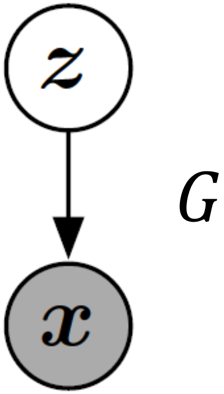
生成对抗网络（Generative Adversarial Nets, GANs）



- 判别器尝试正确的区分真实数据和模型生成的假数据
- 生成器尝试生成高质量的数据来迷惑判别器
- G & D 都可以用神经网络来实现
- 理想状态下，当 D 不能区分真实数据和生成的数据的时候，G 可以很好的拟合真实数据分布

生成器网络（Generator Network）

$$\boldsymbol{x} = G(\boldsymbol{z}; \boldsymbol{\theta})$$

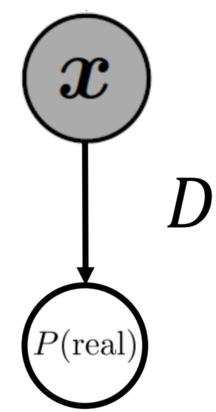


- 从一个随机噪声 $\boldsymbol{z}$ 映射到一个数据实例 $\boldsymbol{x}$
- 必须是可求导（可微分）的
- 没有可逆性要求
- 适用于任何大小的 $\boldsymbol{z}$
- 可以使 $\boldsymbol{x}$ 在给定 $\boldsymbol{z}$ 的时候满足高斯分布，但是不是一定要这么做
  - 例如，变分自编码器（Variational Auto-Encoder，VAE）
- 常见实现方式：多层感知机（MLP）

判别器网络（Discriminator Network）

$$P(real \mid \boldsymbol{x}) = D(\boldsymbol{x}; \boldsymbol{\phi})$$

- 可以由任何能预测概率的神经网络实现
- 例如：
  - Logistic 输出的多层感知机
  - AlexNet 等

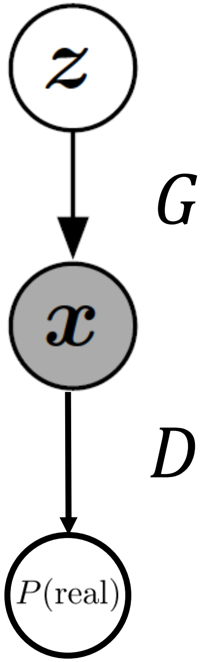


生成器和判别器网络

生成器网络

$$x = G(z; \theta)$$

- 必须是可求导（可微分）的
- 没有可逆性要求
- 常见实现方式：多层感知机（MLP）

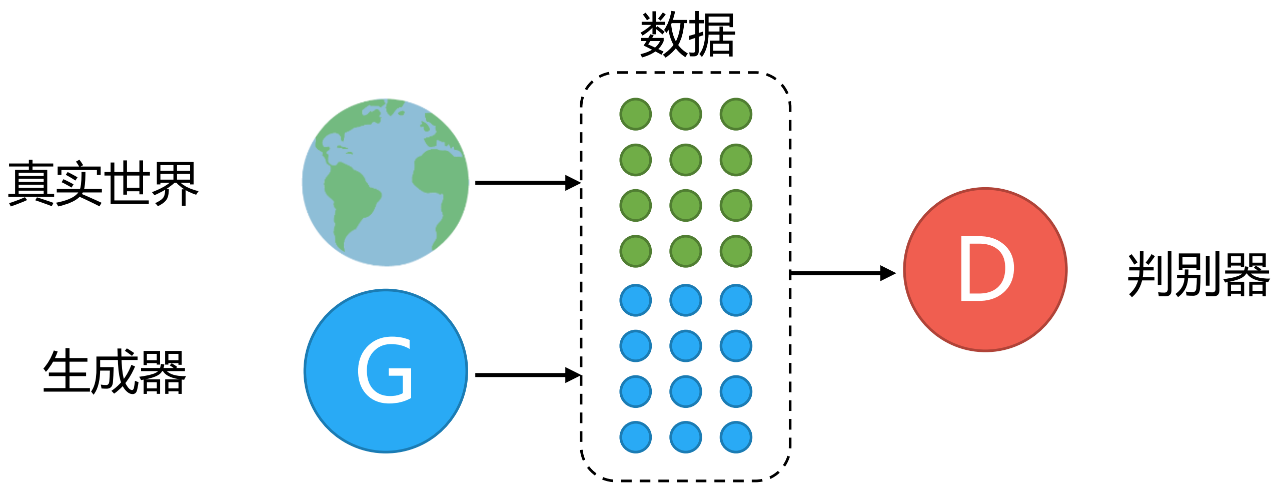


判别器网络

$$P(real \mid x) = D(x; \phi)$$

- 可以由任何能预测概率的神经网络实现
- 例如：
  - Logistic 输出的多层感知机
  - AlexNet 等

生成对抗网络（Generative Adversarial Nets, GANs)



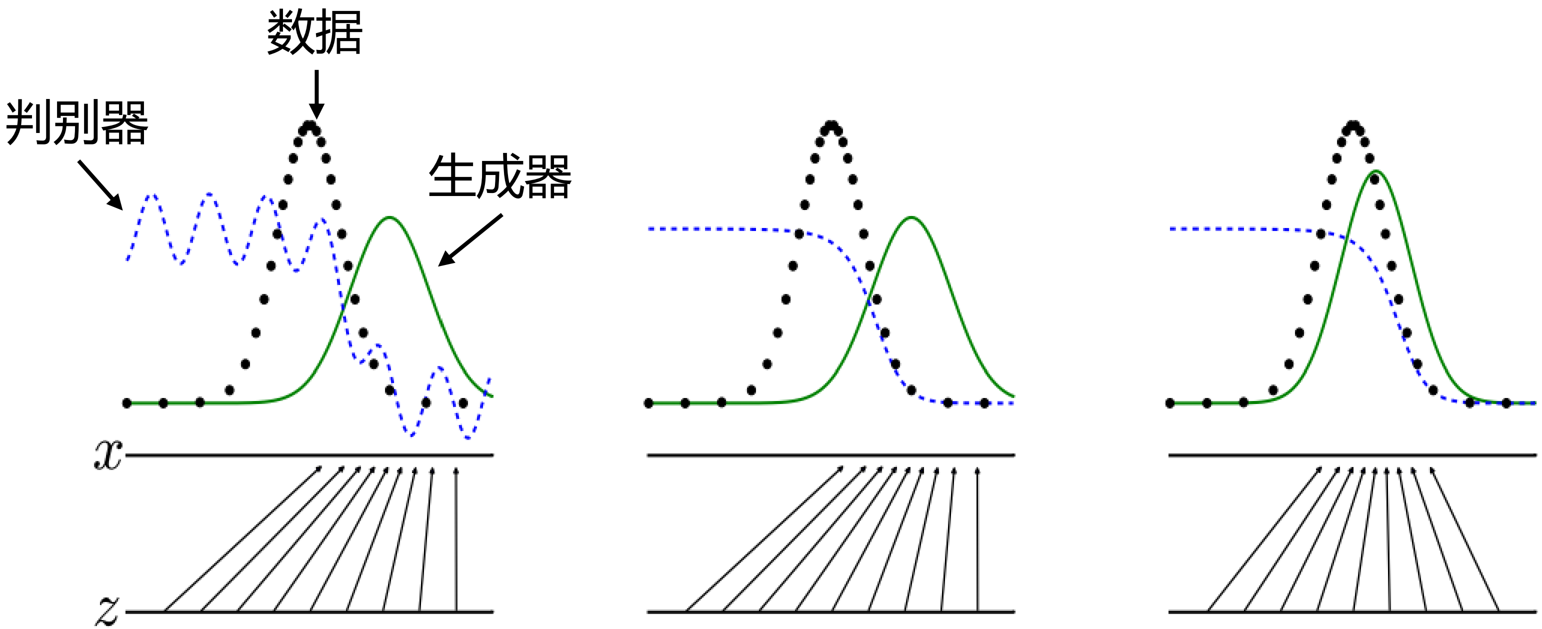
$$\min_{G} \max_{D} J(G, D)$$

$$\max_{D} J(G, D)$$

- 联合目标函数

$$J(G, D) = \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

GAN的图示

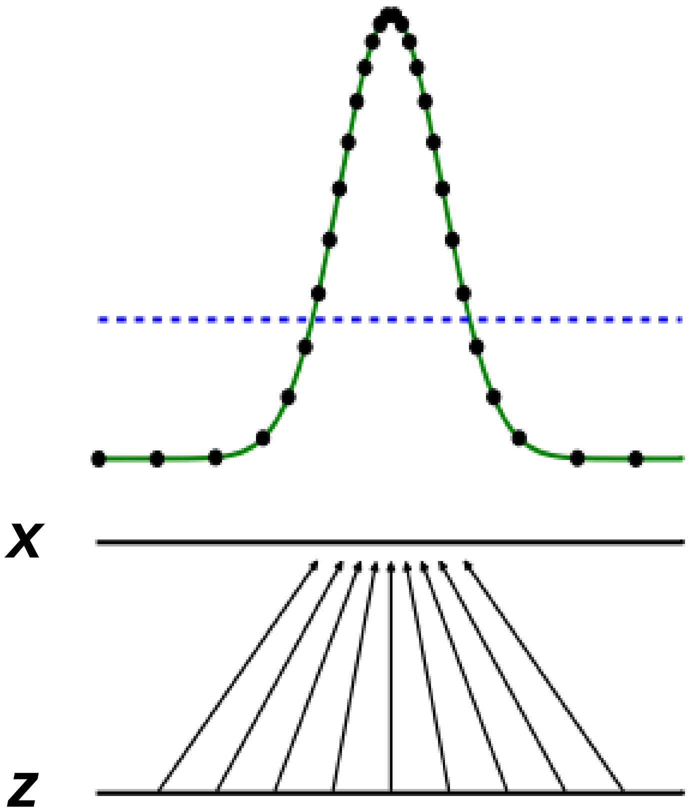


生成器 $\min_G \max_D J^{(D)}$ 判别器 $\max_D J^{(D)}$

$$J(G,D) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

理想最终均衡

- 生成器生成完美的数据分布
- 判别器不能分辨真实的和生成的数据



训练GAN

训练判别器

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

训练GAN

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

训练生成器

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

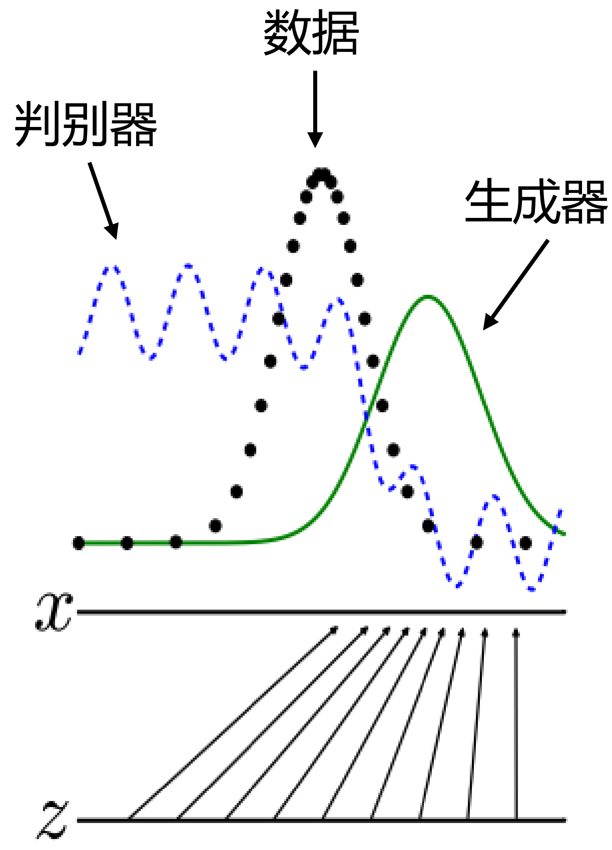
end for

判别器的最优策略

- 对 $p_{data}(\boldsymbol{x})$ 和 $p_G(\boldsymbol{x})$ 来说 $D(\boldsymbol{x})$ 满足

$$D(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_G(\boldsymbol{x})}$$

- 如果可以达到这个最优态，意味着就能达到GAN的理想均衡



MiniMax Game 的均衡

G: $\min_G \max_D J(G,D)$ D: $\max_D J(G,D)$

$$
\begin{aligned}
J(G,D) &= \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(x)] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[\log(1-D(G(z)))\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(x)] + \mathbb{E}_{\boldsymbol{x}\sim p_G(\boldsymbol{x})}\left[\log(1-D(x))\right] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_G(\boldsymbol{x})}] \\
&\quad +\mathbb{E}_{\boldsymbol{x}\sim p_G(\boldsymbol{x})}[\log \frac{p_G(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_G(\boldsymbol{x})}] \\
&= -\log(4) + \underbrace{KL(p_{data} \parallel \frac{p_{data}+p_G}{2})}_{\geq 0} + \underbrace{KL(p_G \parallel \frac{p_{data}+p_G}{2})}_{\geq 0}
\end{aligned}
$$

- 均衡： $p_G(\boldsymbol{x}) = p_{data}(\boldsymbol{x})$ 以及 $D(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x})+p_G(\boldsymbol{x})} = 0.5$

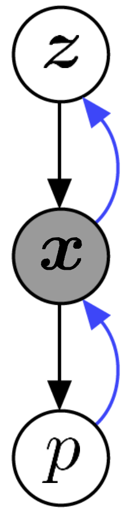
连续型数据上的GAN

1. 生成

$x = G(z;\theta)$

2. 判别

$P(real \mid x) = D(x;\phi)$



4. 求生成器参数上的梯度

$$\frac{\partial J(G,D)}{\partial x}\frac{\partial x}{\partial \theta}$$

3. 求生成数据上的梯度

$$\frac{\partial J(G,D)}{\partial x}$$

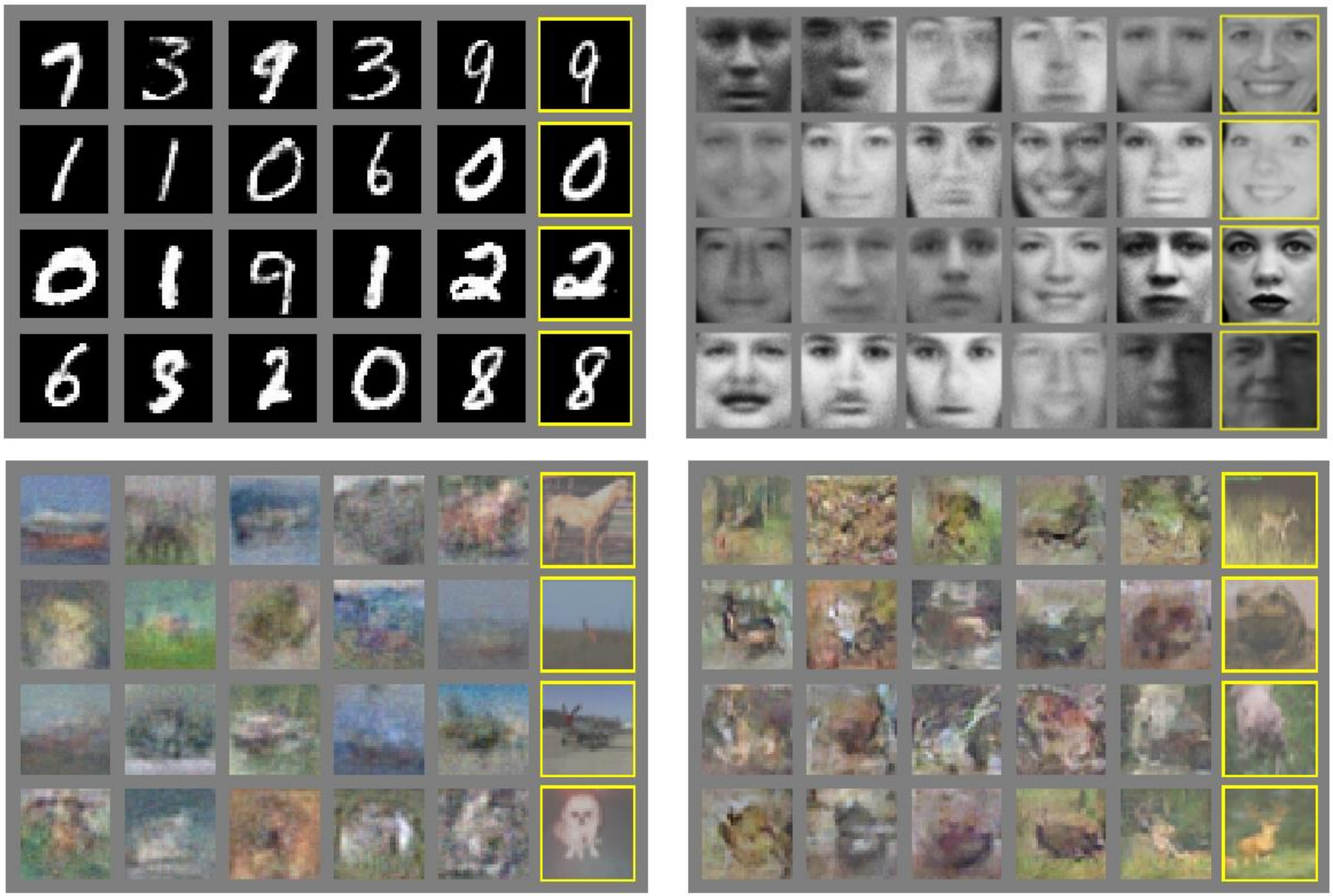
- 为了使用生成器参数的梯度，x 必须是连续的

$$J(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

生成器： $\min\limits_{G}\max\limits_{D} J(G,D)$

判别器： $\max\limits_{D} J(G,D)$

案例分析：用于图像生成GAN



Goodfellow, Ian, et al. "Generative adversarial nets." In NeuralPS, pp. 2672-2680. 2014.

案例分析：用于图像生成GAN

GAN的逐步发展

由一个随机数生成器生成的两个虚拟人像

Tero Karras et al. Progressive Growing of GANs for Improved Quality, Stability, and Variation. ICLR 2018.

案例分析：用于图像生成GAN

BigGAN（ICLR 2019）

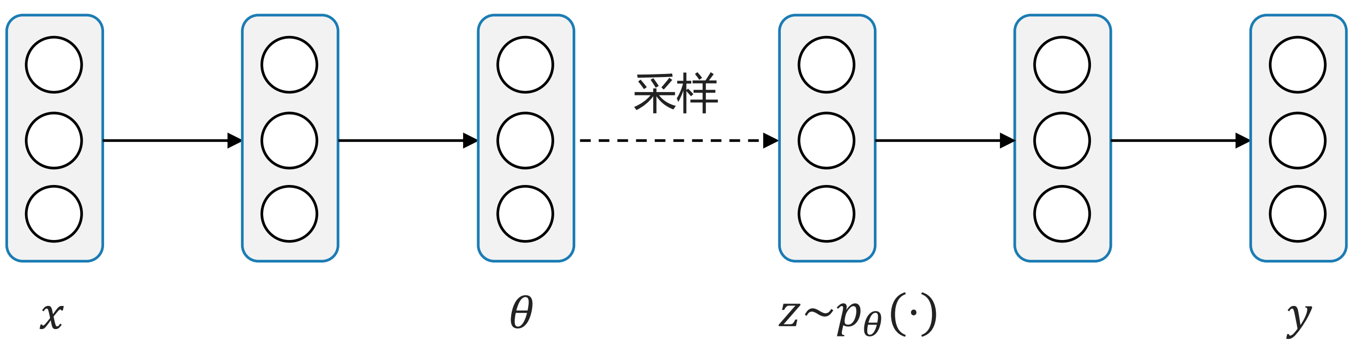
Andrew Brock et al. Large Scale GAN Training for High Fidelity Natural Image Synthesis. ICLR 2019

限制玻尔兹曼机简介

张伟楠 – 上海交通大学

用于无监督学习的随机神经网络

- 随机神经网络是一种将随机变量引入网络而构建的人工神经网络
  - 给出网络的神经元的随机传递函数
  - 给出网络随机权重
- 利用随机神经网络来适应无监督学习的数据分布是很流行的做法



玻尔兹曼机（Boltzmann Machine）

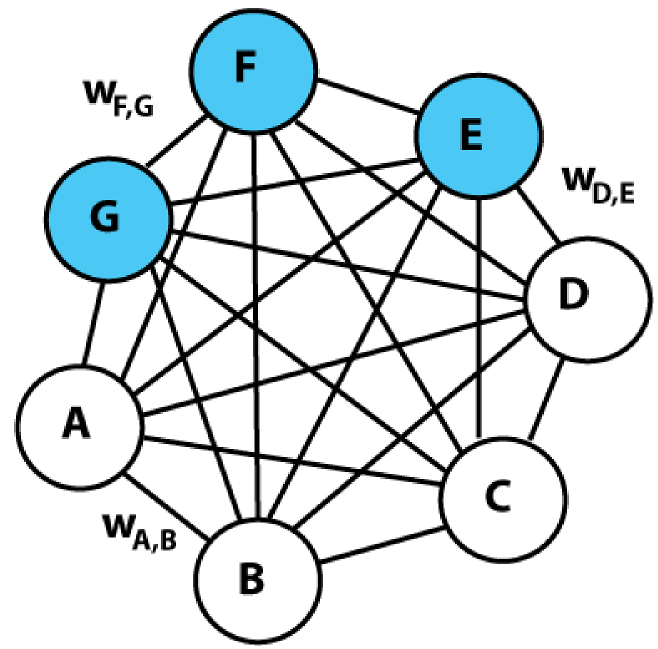
- 玻尔兹曼机是一种特定类型的马尔科夫随机场
  - 无向图网络
  - 玻尔兹曼分布状态

$$p(\boldsymbol{s}) = \frac{e^{-E(\boldsymbol{s})}}{\sum_{\boldsymbol{s}'} e^{-E(\boldsymbol{s}')}} = \frac{1}{Z}\, e^{-E(\boldsymbol{s})}$$

  - 能量函数

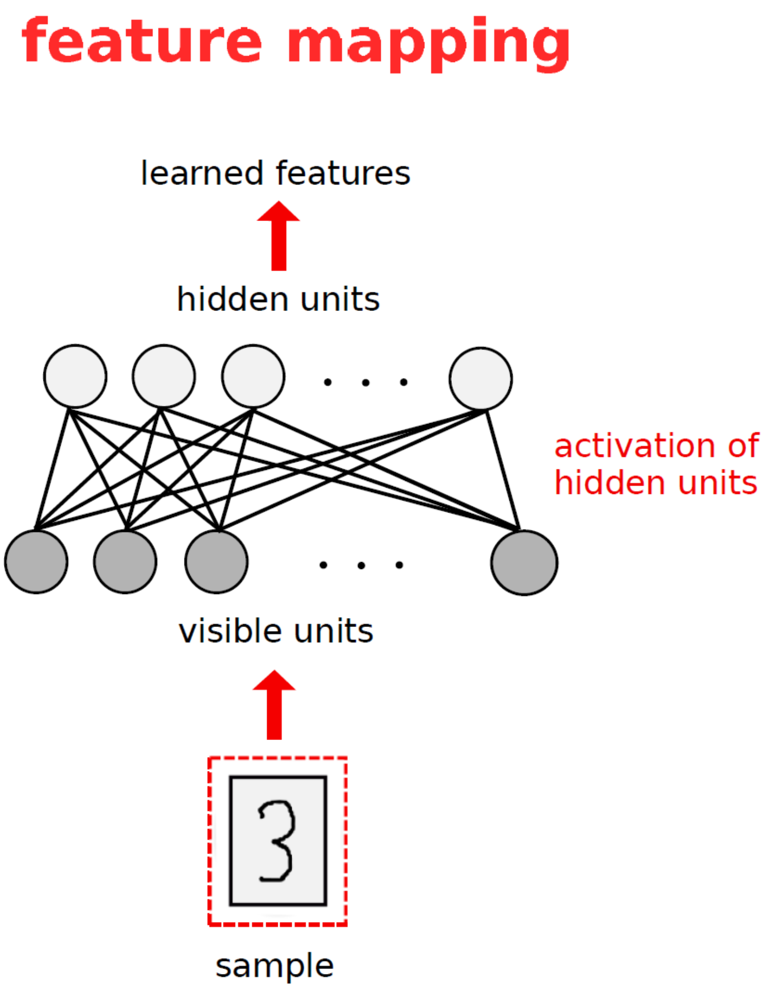
$$E(\boldsymbol{s}) = -\sum_{i<j} w_{ij} s_i s_j - \sum_i \theta_i s_i$$

  - $s_i \in \{0,1\}$是状态单元，w和θ是参数



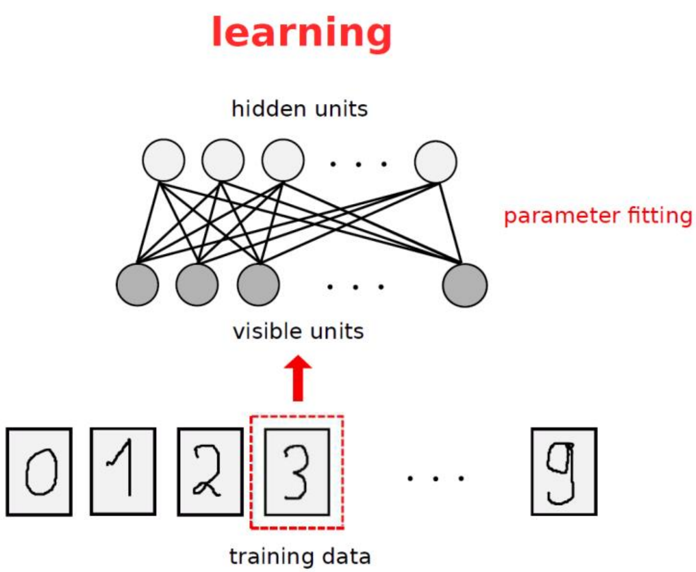
限制玻尔兹曼机 (Restricted Boltzmann Machine)

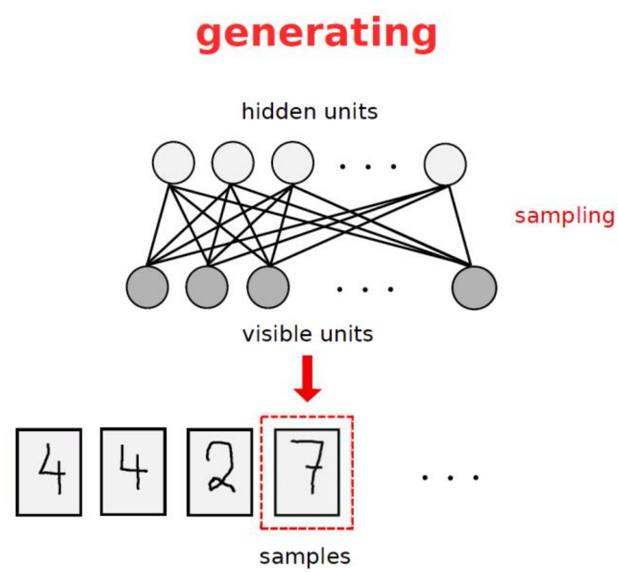
- 限制玻尔兹曼机是一种生成式随机人工神经网络，可以学习其输入集合的概率分布
- 限制：可见（隐藏）单元彼此不连接，像是二分图
- 隐藏单元（随机变量）可以对输入样本的学习特征进行采样



Fischer, Asja, and Christian Igel. "Training restricted Boltzmann machines: An introduction." Pattern Recognition47.1 (2014): 25-39.

限制玻尔兹曼机（RBM）



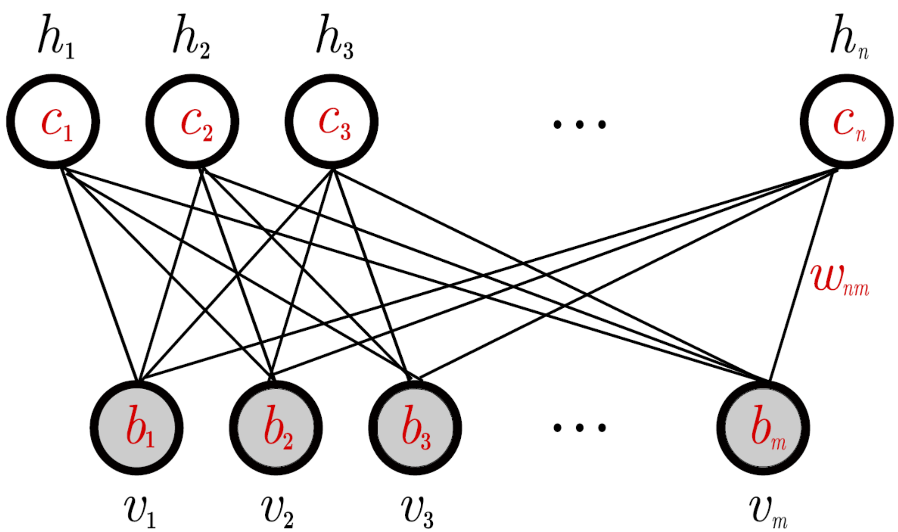


- 学习阶段
  - 拟合参数（连接权重和节点偏差）
  - 以便 RBM 可以恢复输入样本
- 生成阶段
  - 利用拟合参数，隐藏单元的相应边缘分布可用于对隐藏向量进行采样，然后输出相应的样本

Fischer, Asja, and Christian Igel. "Training restricted Boltzmann machines: An introduction." Pattern Recognition47.1 (2014): 25-39.

限制玻尔兹曼机

- 可见单元$\{v_j\}$只连接到隐藏单元$\{h_i\}$



c_i： 隐藏单元偏置项

w_{ij}： 连接权重

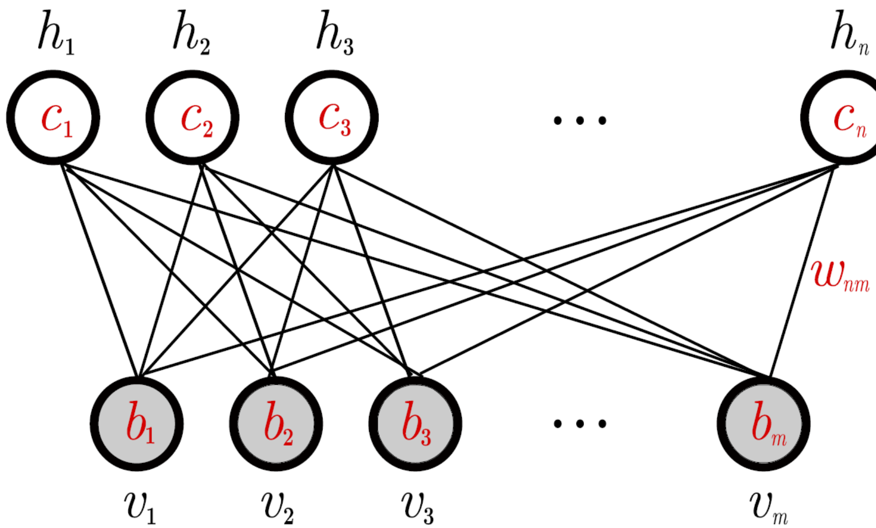
b_j： 可见单元偏置项

- 能量函数

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

限制玻尔兹曼机

- 可见单元$\{v_j\}$只连接到隐藏单元$\{h_i\}$



c_i： 隐藏单元偏置项

w_{ij}： 连接权重

b_j： 可见单元偏置项

- 条件概率

$$p(\boldsymbol{h}|\boldsymbol{v}) = \prod_{i=1}^{n} p(h_i|\boldsymbol{v}) \qquad p(\boldsymbol{v}|\boldsymbol{h}) = \prod_{j=1}^{m} p(v_j|\boldsymbol{h})$$

可见单元的边缘分布

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_iv_j - \sum_{j=1}^{m} b_jv_j - \sum_{i=1}^{n} c_ih_i$$

$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} = \frac{1}{Z}\sum_{h_1}\sum_{h_2}\cdots\sum_{h_n} e^{\sum_{j=1}^{m} b_jv_j}\prod_{i=1}^{n} e^{h_i\left(c_i+\sum_{j=1}^{m} w_{ij}v_j\right)}$$

$$= \frac{1}{Z}e^{\sum_{j=1}^{m} b_jv_j}\sum_{h_1} e^{h_1\left(c_1+\sum_{j=1}^{m} w_{1j}v_j\right)}\sum_{h_2} e^{h_2\left(c_2+\sum_{j=1}^{m} w_{2j}v_j\right)}\cdots\sum_{h_n} e^{h_n\left(c_n+\sum_{j=1}^{m} w_{nj}v_j\right)}$$

$$= \frac{1}{Z}e^{\sum_{j=1}^{m} b_jv_j}\prod_{i=1}^{n}\sum_{h_i} e^{h_i\left(c_i+\sum_{j=1}^{m} w_{ij}v_j\right)} = \frac{1}{Z}\prod_{j=1}^{m} e^{b_jv_j}\prod_{i=1}^{n}\left(1+e^{c_i+\sum_{j=1}^{m} w_{ij}v_j}\right)$$

$h_i \in \{0,1\}$

- $\{0,1\}^m$上的任何分布都可以通过具有m个可见单元和一定数量隐藏单元的限制玻尔兹曼机建模

Fischer, Asja, and Christian Igel. "Training restricted Boltzmann machines: An introduction." Pattern Recognition47.1 (2014): 25-39.

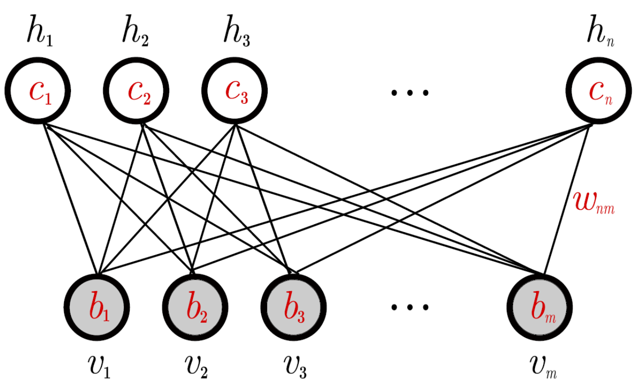
限制玻尔兹曼机的条件概率

- 用sigmoid激活函数的标准实现

$$p(\boldsymbol{h}|\boldsymbol{v}) = \prod_{i=1}^{n} p(h_i|\boldsymbol{v}) \qquad p(h_i = 1|\boldsymbol{v}) = \sigma\left(\sum_{j=1}^{m} w_{ij}v_j + c_i\right)$$

$$p(\boldsymbol{v}|\boldsymbol{h}) = \prod_{j=1}^{m} p(v_j|\boldsymbol{h}) \qquad p(v_j = 1|\boldsymbol{h}) = \sigma\left(\sum_{i=1}^{m} w_{ij}h_i + b_j\right)$$

- 限制玻尔兹曼机可以解释为用sigmoid激活的标准单层前向神经网络



限制玻尔兹曼机学习算法

张伟楠 – 上海交通大学

RBM的对数似然函数

- 可见单元$\boldsymbol{v}$的边缘似然

$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} = \frac{\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{v}',\boldsymbol{h}} e^{-E(\boldsymbol{v}',\boldsymbol{h})}}$$

- 关于RBM的参数$\boldsymbol{\theta}$的对数似然

$$\begin{aligned}\log p(\boldsymbol{v}; \boldsymbol{\theta}) &= \log \frac{1}{Z} \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \\ &= \log \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} - \log \sum_{\boldsymbol{v}',\boldsymbol{h}} e^{-E(\boldsymbol{v}',\boldsymbol{h})}\end{aligned}$$

关于参数的对数似然

- 关于RBM的参数$\boldsymbol{\theta}$的对数似然

$$\log p(\boldsymbol{v};\boldsymbol{\theta}) = \log \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} - \log \sum_{\boldsymbol{v}',\boldsymbol{h}} e^{-E(\boldsymbol{v}',\boldsymbol{h})}$$

- 对于特定的训练数据$\boldsymbol{v}$，关于RBM的参数$\boldsymbol{\theta}$的对数似然的梯度是

$$\begin{aligned}
\frac{\partial \log p(\boldsymbol{v};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}}\left(\log \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}\right) - \frac{\partial}{\partial \boldsymbol{\theta}}\left(\log \sum_{\boldsymbol{v}',\boldsymbol{h}} e^{-E(\boldsymbol{v}',\boldsymbol{h})}\right) \\
&= -\frac{\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}}}{\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}} + \frac{\sum_{\boldsymbol{v}',\boldsymbol{h}} e^{-E(\boldsymbol{v}',\boldsymbol{h})} \frac{\partial E(\boldsymbol{v}',\boldsymbol{h})}{\partial \boldsymbol{\theta}}}{\sum_{\boldsymbol{v}',\boldsymbol{h}} e^{-E(\boldsymbol{v}',\boldsymbol{h})}} \\
&= -\mathbb{E}_{p(\boldsymbol{h}|\boldsymbol{v})}\left[\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}}\right] + \mathbb{E}_{p(\boldsymbol{h}|\boldsymbol{v}')p(\boldsymbol{v}')}\left[\frac{\partial E(\boldsymbol{v}',\boldsymbol{h})}{\partial \boldsymbol{\theta}}\right]
\end{aligned}$$

关于RBM参数的梯度

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_iv_j - \sum_{j=1}^{m} b_jv_j - \sum_{i=1}^{n} c_ih_i$$

$$\frac{\partial \log p(\boldsymbol{v};\boldsymbol{\theta})}{\partial w_{ij}} = \mathbb{E}_{p(\boldsymbol{h}|\boldsymbol{v})}[h_iv_j] - \mathbb{E}_{p(\boldsymbol{h}|\boldsymbol{v}')p(\boldsymbol{v}')}[h_i{v'}_j]$$

$$\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial w_{ij}} = -h_iv_j$$

$$\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial b_j} = -v_j$$

$$\frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial c_i} = -h_i$$

- 对于训练数据集$D = \{\boldsymbol{v}_i\}_{i=1\ldots|D|}$，$w_{ij}$的经验梯度为

$$\frac{1}{|D|}\sum_{v\in D}\frac{\partial \log p(\boldsymbol{v};\boldsymbol{\theta})}{\partial w_{ij}} = \frac{1}{|D|}\sum_{v\in D}\left\{\mathbb{E}_{p(\boldsymbol{h}|\boldsymbol{v})}[h_iv_j]\right\} - \mathbb{E}_{p(\boldsymbol{h}|\boldsymbol{v}')p(\boldsymbol{v}')}[h_iv'_j]$$

$$= \langle h_iv_j\rangle_{p(\boldsymbol{h}|\boldsymbol{v})q(\boldsymbol{v})} - \langle h_iv_j\rangle_{p(\boldsymbol{h},\boldsymbol{v})}$$

$q(\boldsymbol{v})$是经验数据分布

$$= \langle h_iv_j\rangle_{\text{data}} - \langle h_iv_j\rangle_{\text{model}}$$

如何计算梯度

- 对于理论对数似然梯度

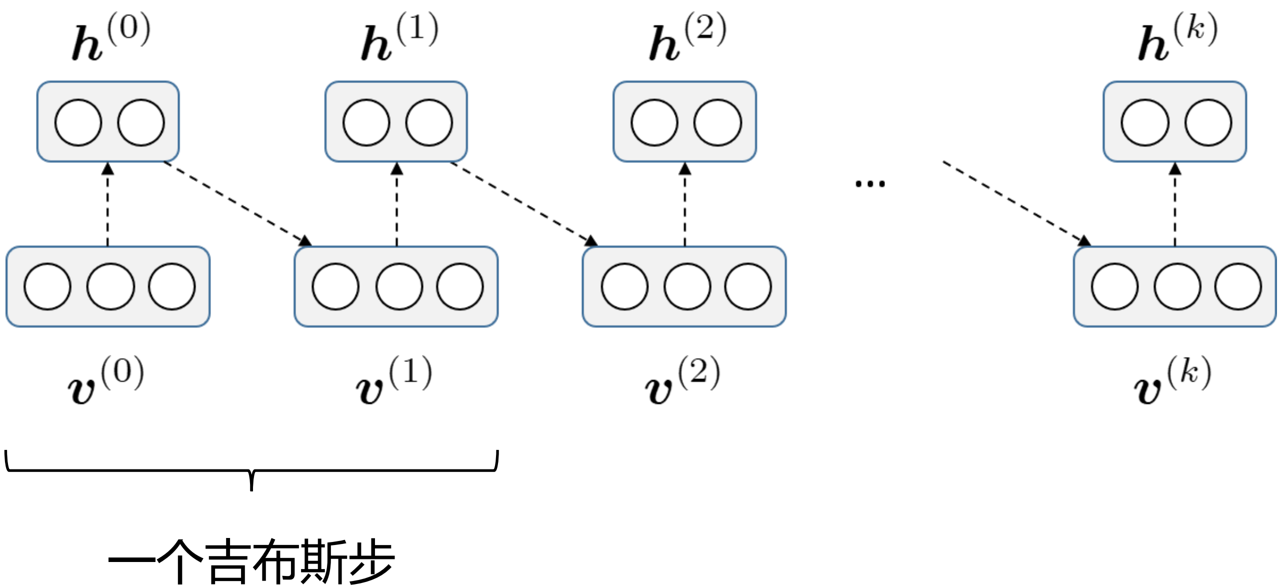
$$\frac{1}{|D|}\sum_{\boldsymbol{v}\in D}\frac{\partial \log p(\boldsymbol{v};\boldsymbol{\theta})}{\partial w_{ij}} = \langle h_i v_j\rangle_{\text{data}} - \langle h_i v_j\rangle_{\text{model}}$$

给定训练数据易于计算　　不可计算

- $\langle h_i v_j\rangle_{\text{model}} = \mathbb{E}_{p(\boldsymbol{h}|\boldsymbol{v}')p(\boldsymbol{v}')}[h_i {v'}_j]$需要从限制玻尔兹曼机的静态分布中采样$v$，这需要无限的MCMC采样步骤
- 对比散度（Contrastive Divergence）的想法：仅通过执行k次吉布斯（Gibbs）采样来逼近它

对比散度（Contrastive Divergence）

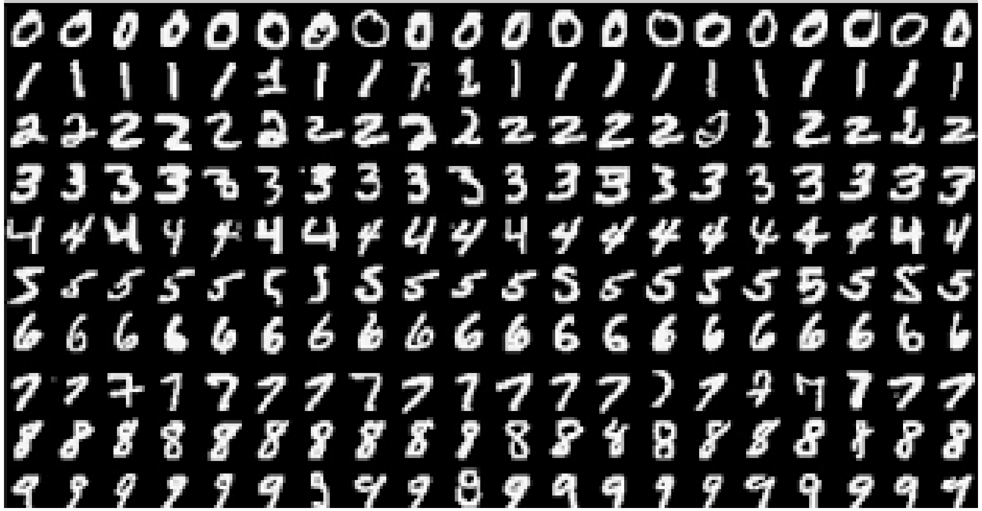
- k-步对比散度学习（CD-k）的想法是仅仅对一个吉布斯链执行k步（通常$k=1$）



$$\mathrm{CD}_k(\boldsymbol{v}^{(0)}) = \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^{(0)}) v_i^{(0)} h_j - \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{v}^{(k)}) v_i^{(k)} h_j$$

RBM实验

MNIST手写数据集



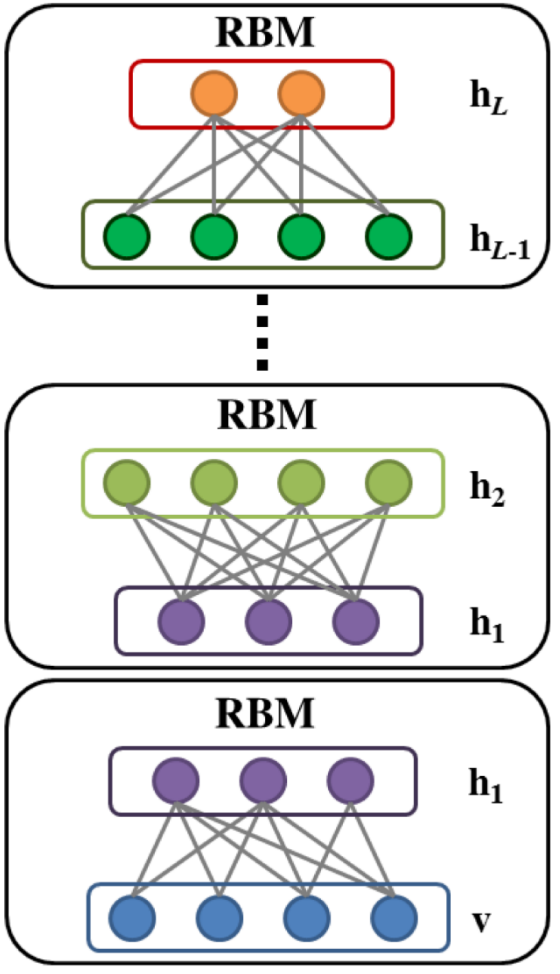
RBM生成的数据示例

深度信念网络

张伟楠 – 上海交通大学

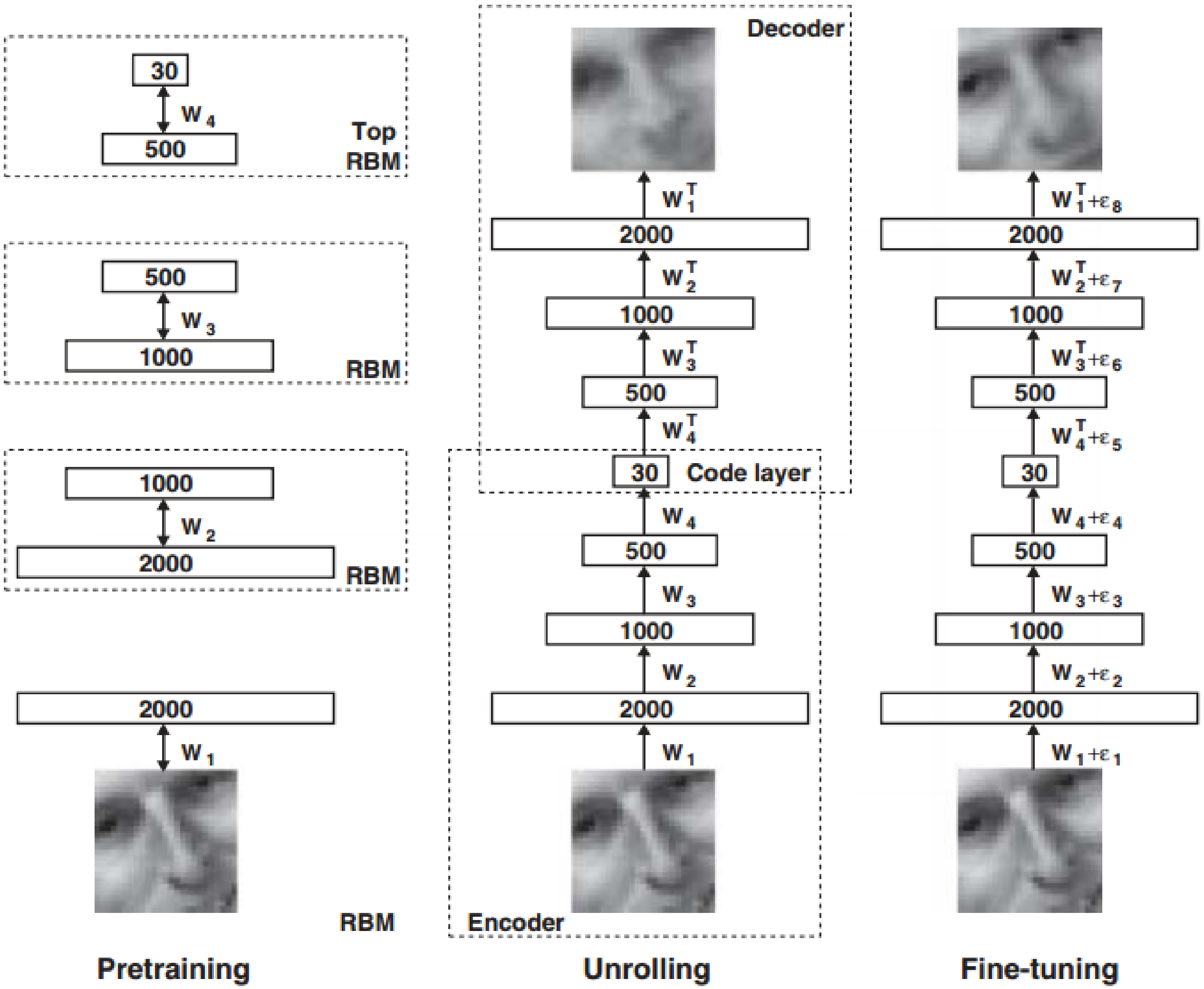
深度信念网络（Deep Belief Network）

- 深度信念网络（DBN）是由多层RBM组成的概率生成模型
  - 逐层训练以恢复数据
  - 通过学习，每一层潜在变量的值可以通过单个自下而上的传递来推断，该传递从底层中的观察数据向量开始并且使用反向的生成权重



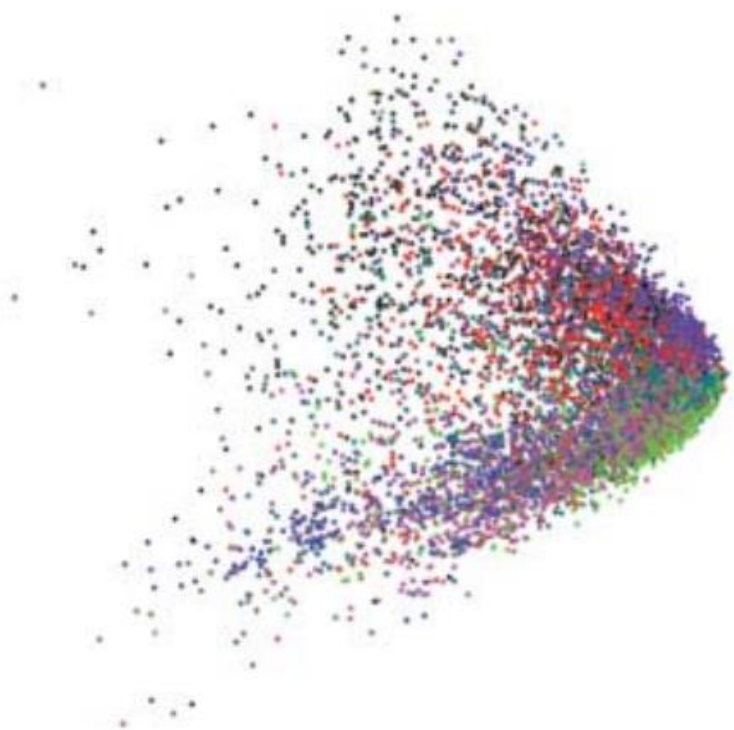
Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313.5786 (2006): 504-507.

深度信念网络（DBN）

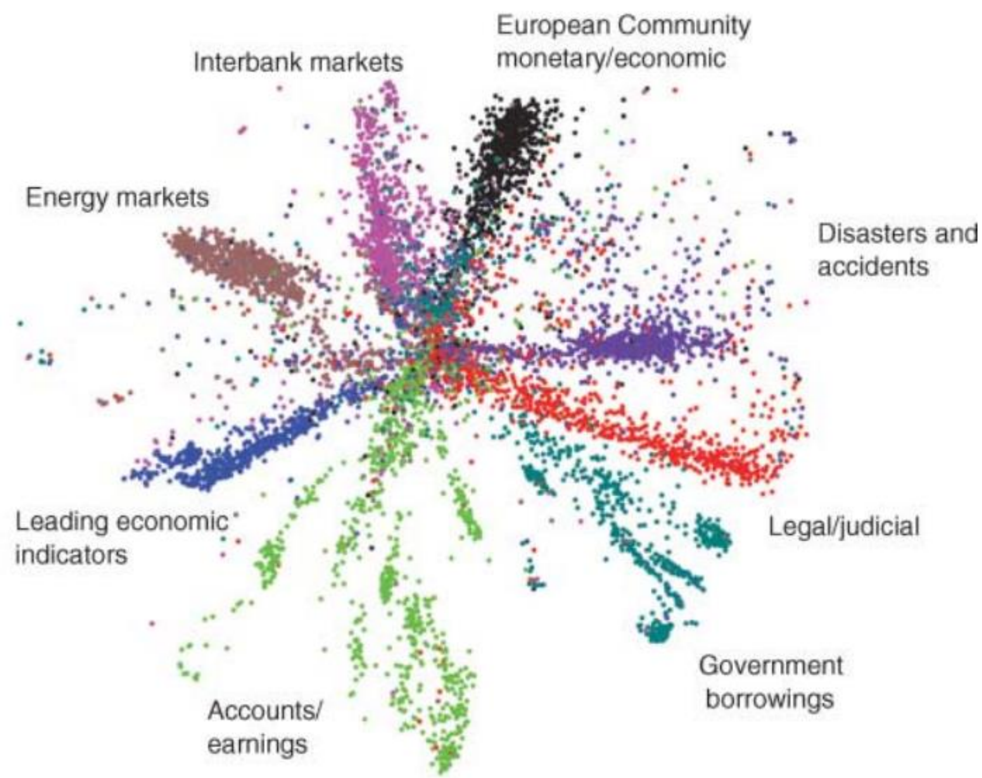


Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313.5786 (2006): 504-507.

潜在因子分析



基于PCA的潜在因子分析



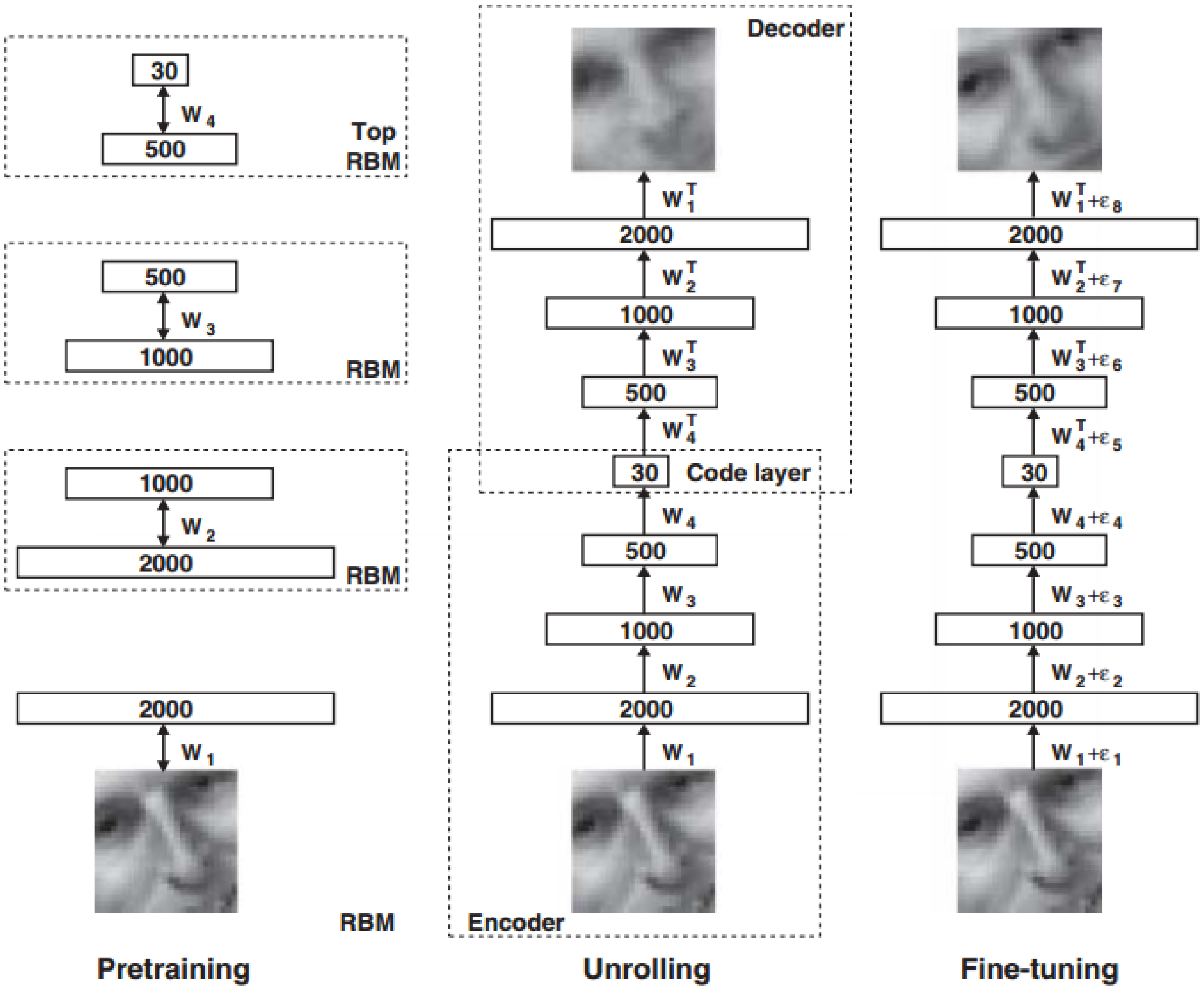
基于一个由DBN训练的2000-500-250-125-2的自编码器的潜在因子分析

Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313.5786 (2006): 504-507.

自编码器

张伟楠 – 上海交通大学

深度信念网络（DBN）



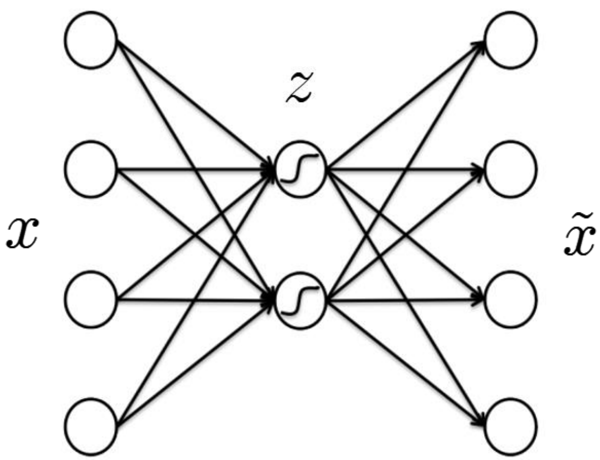
Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313.5786 (2006): 504-507.

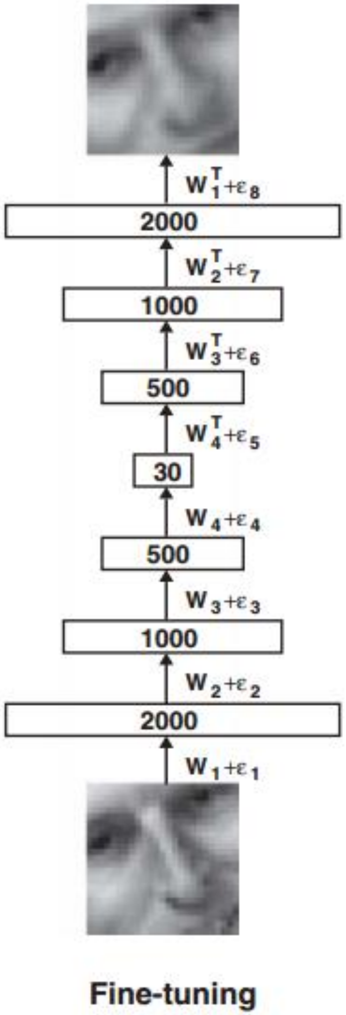
自编码器（Auto-encoder)

- 自编码器是用于无监督学习有效编码的人工神经网络
  - 学习一组数据的表示（编码），通常用于降低维数

$$z = \sigma(W_1 x + b_1)$$
$$\tilde{x} = \sigma(W_2 z + b_2)$$

z被认为是x的低维潜在因子表示





自编码器学习

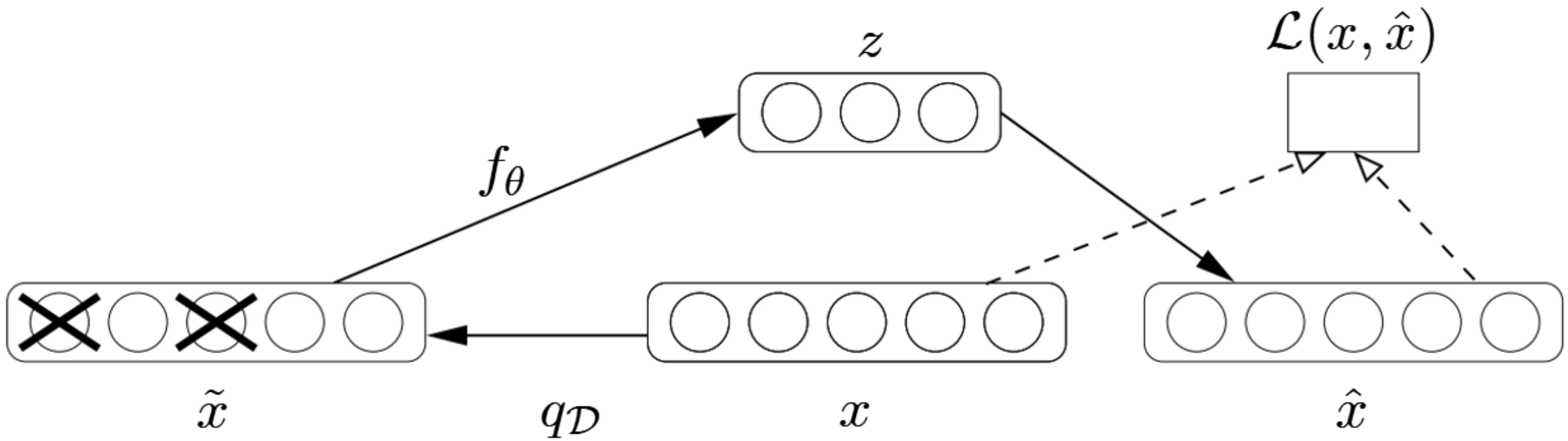
- 目标：x和$\tilde{x}$之间的平方差

$$
\begin{aligned}
J(W_1, b_1, W_2, b_2) &= \sum_{i=1}^{m} \left(\tilde{x}^{(i)} - x^{(i)}\right)^2 \\
&= \sum_{i=1}^{m} \left(W_2 z^{(i)} + b_2 - x^{(i)}\right)^2 \\
&= \sum_{i=1}^{m} \left(W_2 \sigma\left(W_1 x^{(i)} + b_1\right) + b_2 - x^{(i)}\right)^2
\end{aligned}
$$

- 自编码器是一种以监督方式训练的无监督学习模型

$$
\theta \leftarrow \theta - \eta \frac{\partial J}{\partial \theta}
$$

降噪自编码器（Denoising Auto-encoder）



- 原本的输入x部分被破坏，产生损坏的输入

$$\tilde{x} \sim q_\mathcal{D}(\tilde{x}|x)$$

例如：高斯噪声

- 损坏的输入$\tilde{x}$被映射到隐藏表示

$$z = f_\theta(\tilde{x})$$

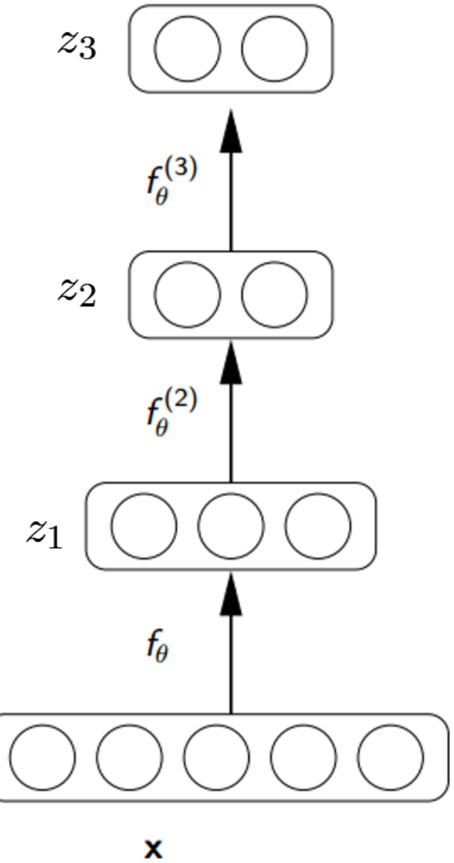
- 从z中还原数据

$$\hat{x} = g_{\theta'}(z)$$

堆叠自编码器（Stacked Auto-encoder)

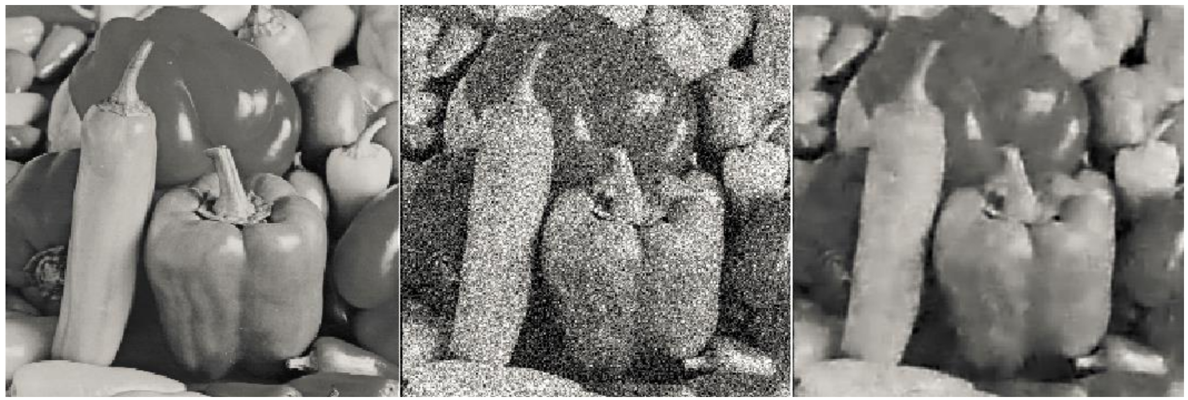
- 逐层训练

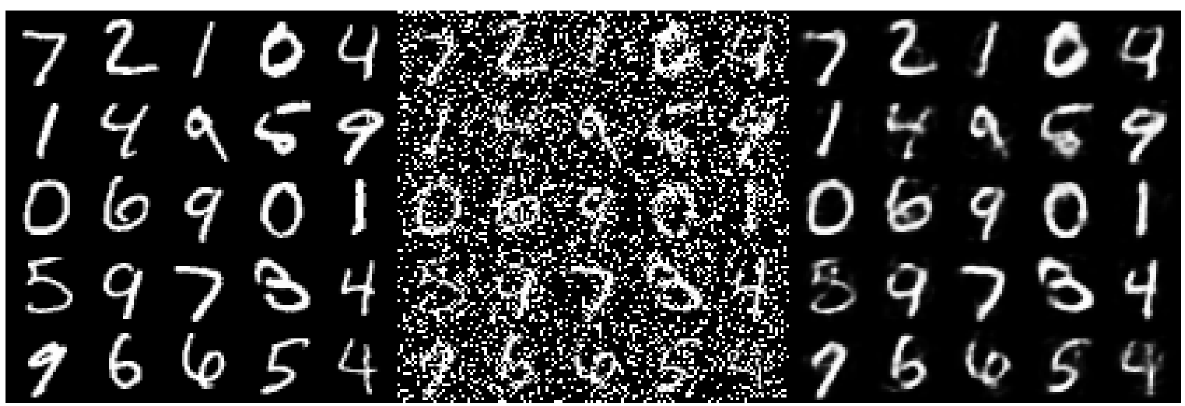
1. 训练第一层使用z_1来恢复x
2. 训练第二层使用z_2恢复z_1
3. 训练第三层使用z_3恢复z_2



一些降噪自编码器的例子

源图片　　损坏后　　恢复





总结深度无监督学习

- 借助深度神经网络的普适逼近性质，深度无监督学习模型可以建模任意数据分布
  - 但需要以黑盒模型建立non-explicit distribution density
- 一旦要给出explicit distribution density，那就得提前定好distribution family（例如高斯）然后用神经网络去拟合分布的参数
  - 这样做很可能拟合并不到位
- GAN、RBM、VAE都是建立non-explicit distribution density
  - 给定中间（噪音）变量z，经过神经网络映射输出数据实例
- 深度无监督学习包含自监督学习（Self-supervised Learning）
  - Word embedding, node embedding, language modeling…
  - 无监督学习模型预训练 + 在具体预测任务上做fine tuning

THANK YOU