机器学习基础

一、基本概念

1. 可以通过无监督学习来求解监督学习问题 $p(y \mid \vec{\mathbf{x}})$：
   - 首先求解无监督学习问题来学习联合概率分布 $p(\vec{\mathbf{x}}, y)$
   - 然后计算：

$$p(y \mid \vec{\mathbf{x}}) = \frac{p(\vec{\mathbf{x}}, y)}{\sum_{y'} p(\vec{\mathbf{x}}, y')}$$

1.1 泛化能力度量

1. 为了评估机器学习算法的能力，必须给定其性能的衡量指标。
2. 有些情况下，很难决定衡量指标是什么：
   - 如：翻译任务中，应该衡量整个翻译结果的准确率，还是衡量每个单词翻译的准确率？
   - 如：密度估计任务中，很多模型都是隐式地表示概率分布。此时计算样本空间某个点的真实概率是不可行的，因此也就无法判断该点的概率估计的准确率。
3. 通常利用最小化训练误差来训练模型，但是真正关心的是测试误差。因此通过测试误差来评估模型的泛化能力。
4. 统计理论表明：如果训练集和测试集中的样本都是独立同分布产生的，则有 **模型的训练误差的期望等于模型的测试误差的期望**
   - 训练集和测试集共同的、潜在的样本分布称作数据生成分布，记作 p_{data}

1.2 模型容量

1.2.1 过拟合、欠拟合

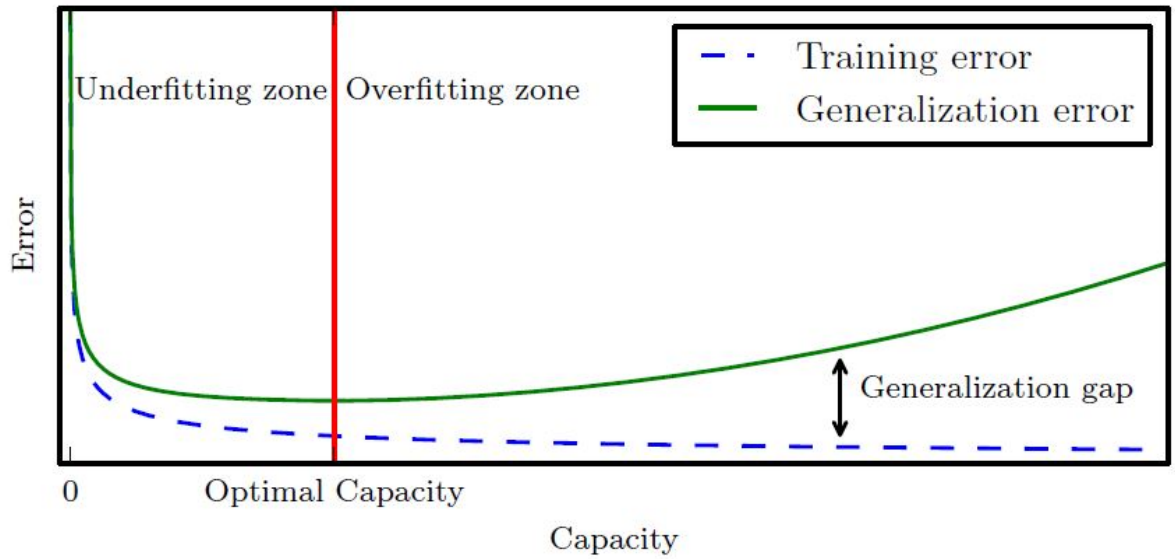
1. 当使用机器学习算法时，决定机器学习算法效果的两个因素：
   - 降低训练误差
   - 缩小训练误差和测试误差的差距
2. 这两个因素对应着机器学习中的两个主要挑战：欠拟合和过拟合。
   - 欠拟合是由于模型不能在训练集上获取足够小的训练误差（即：训练误差较大）
   - 过拟合是由于模型的训练误差和测试误差之间的差距太大

1.2.2 模型容量

1. 通过调整模型的容量 `capacity` 可以缓解这欠拟合和过拟合
2. 模型的容量是指其拟合各种函数的能力。
   - 容量低的模型容易发生欠拟合，模型拟合能力太弱
   - 容量高的模型容易发生过拟合，模型拟合能力太强
3. 通过选择不同的假设空间可以改变模型的容量

    - 模型的假设空间：即代表模型的函数集合（这也称作模型的表示容量`representational capacity`）。
- 通常在这些函数中挑选出最佳的函数是非常困难的优化问题，实际应用中只是挑选一个使得训练误差足够低的函数即可。

4. 由于额外的限制因素（比如优化算法的不完善），模型的有效容量`effective capacity`一般会小于模型的表示容量。
5. 统计学习理论提供了量化模型容量的方法，其中最出名的是`VC`维理论：**训练误差与泛化误差之间差异的上界随着模型容量增长而增长，随着训练样本增多而下降**。
6. 虽然`VC`维理论对于机器学习算法有很好的指导作用，但是深度学习很难应用。原因有二：
    - 边界太宽泛
    - 难以确定深度学习的容量。由于深度学习模型的有效容量受限于优化算法，因此确定深度学习模型的容量特别困难。
7. 通常泛化误差是关于模型容量的`U`形函数。随着模型容量增大：
    - 训练误差会下降直到逼近其最小值
    - 泛化误差先减小后增大
    - 泛化误差与训练误差的差值会增大。



1.3 没有免费午餐定理

1. 机器学习的“没有免费的午餐定理”表明：在所有可能的数据生成分布上，没有一个机器学习算法总是比其他的要好。
    - 该结论仅在考虑所有可能的数据分布时才成立。
    - 现实中，特定任务的数据分布往往满足某类假设，从而可以设计在这类分布上效果更好的学习算法。
    - 这意味着机器学习并不需要寻找一个通用的学习算法，而是寻找一个在关心的数据分布上效果最好的算法。
2. 正则化是对学习算法做的一个修改，这种修改趋向于降低泛化误差（而不是降低训练误差）
    - 正则化是机器学习领域的中心问题之一
    - 没有免费的午餐定理说明了没有最优的学习算法，因此也没有最优的正则化形式

1.4 验证集

1. 大多数机器学习算法具有超参数，超参数的值无法通过学习算法拟合出来（比如正则化项的系数、控制模型容量的参数 ）。
2. 为了解决这个问题，可以引入验证集。
    - 将训练数据分成两个不相交的子集：训练集用于学习模型，验证集用于更新超参数。
    - 验证集通常会低估泛化误差。因此当超参数优化完成后，需要通过测试集来估计泛化误差。

二、点估计、偏差方差

2.1 点估计

1. 点估计：对参数 θ 的一个预测，记作 $\hat{\theta}$。

   假设 $\{x_1,x_2,\cdots,x_m\}$ 为独立同分布的数据点，该分布由参数 θ 决定。则参数 θ 的点估计为某个函数：

   $$\hat{\theta}_m=g(x_1,x_2,\cdots,x_m)$$

   注意：点估计的定义并不要求 g 返回一个接近真实值 θ 。
2. 根据频率学派的观点：
    - 真实参值 θ 是固定的，但是未知的。
    - $\hat{\theta}_m$ 是数据点的函数。
    - 由于数据是随机采样的，因此 $\hat{\theta}_m$ 是个随机变量。

2.2 偏差

1. 偏差定义为： $bias(\hat{\theta}_m)=\mathbb{E}(\hat{\theta}_m)-\theta$，期望作用在所有数据上。
    - 如果 $bias(\hat{\theta}_m)=0$ ，则称估计量 $\hat{\theta}_m$ 是无偏的。
    - 如果 $\lim_{m\to\infty}bias(\hat{\theta}_m)=0$，则称估计量 $\hat{\theta}_m$ 是渐近无偏的。
2. 无偏估计并不一定是最好的估计。
3. 偏差的例子：
    - 一组服从均值为 θ 的伯努利分布的独立同分布样本 $\{x_1,x_2,\cdots,x_m\}$ ， $\hat{\theta}_m=\frac{1}{m}\sum_{i=1}^m x_i$ 为 θ 的无偏估计。
    - 一组服从均值为 μ，方差为 σ^2 的高斯分布的独立同分布样本 $\{x_1,x_2,\cdots,x_m\}$：
        - $\hat{\mu}_m=\frac{1}{m}\sum_{i=1}^m x_i$ 为 μ 的无偏估计。
        - $\hat{\sigma}_m^2=\frac{1}{m}\sum_{i=1}^m(x_i-\hat{\mu}_m)^2$ 为 σ^2 的有偏估计。因为 $\mathbb{E}[\hat{\sigma}_m^2]=\frac{m-1}{m}\sigma^2$
        - $\tilde{\sigma}_m^2=\frac{1}{m-1}\sum_{i=1}^m(x_i-\hat{\mu}_m)^2$ 为 σ^2 的无偏估计。

2.3 方差

1. 估计量的方差记作 $Var(\hat{\theta})$，标准差记作 $SE(\hat{\theta})$ 。
    - 它们刻画的是：从潜在的数据分布中独立的获取样本集时，估计量的变化程度。
2. 例：一组服从均值为 θ 的伯努利分布的独立同分布样本 $\{x_1,x_2,\cdots,x_m\}$
    - $\hat{\theta}_m=\frac{1}{m}\sum_{i=1}^m x_i$ 为 θ 的无偏估计
    - $Var(\hat{\theta}_m)=\frac{1}{m}\theta(1-\theta)$ 。表明估计量的方差随 m 增加而下降。
3. 估计量的方差随着样本数量的增加而下降，这是所有估计量的共性。
4. 例：均值估计 $\hat{\mu}_m=\frac{1}{m}\sum_{i=1}^m x_i$ ，其标准差为：

$$SE(\hat{\mu}_m) = \sqrt{Var\left[\frac{1}{m}\sum_{i=1}^{m} x_i\right]} = \frac{\sigma}{\sqrt{m}}$$

其中 σ 是样本 x_i 的真实标准差，但是这个量难以估计。

> $\sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \hat{\mu}_m)^2}$ 和 $\sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(x_i - \hat{\mu}_m)^2}$ 都不是真实标准差 σ 的无偏估计
>
> - 这两种方法都倾向于低估真实的标准差
> - 实际应用中，$\sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(x_i - \hat{\mu}_m)^2}$ 是一种比较合理的近似估计，尤其是当 m 较大的时候。

2.4 偏差方差分解

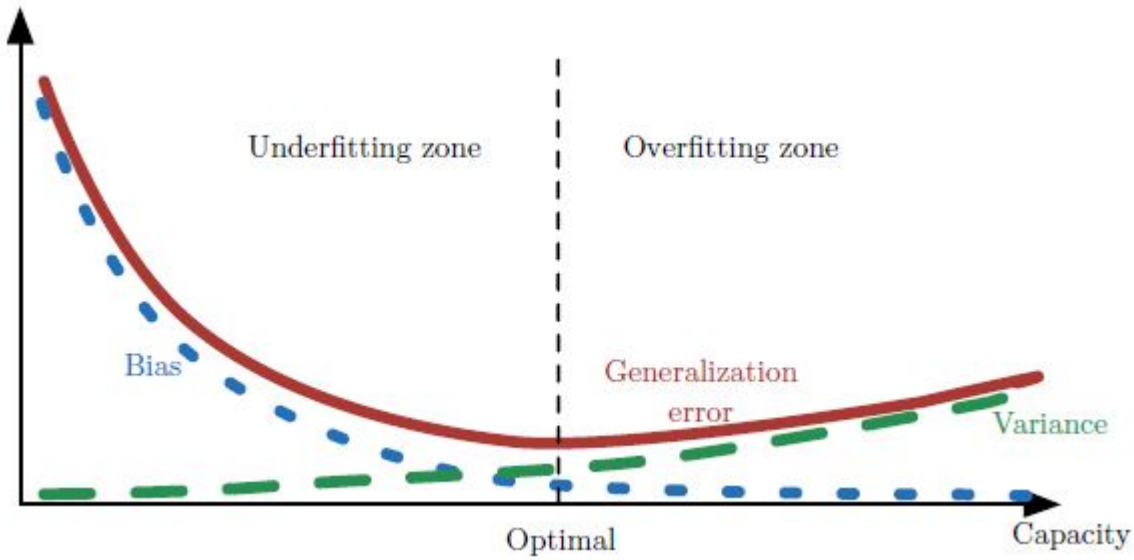
1. 通常希望的是：
   - 估计量的偏差比较小，即：估计量的期望值接近真实值
   - 估计量的方差比较小，即：估计量的波动比较小
2. 偏差和方差衡量的是估计量的两个不同误差来源：
   - 偏差衡量的是偏离真实值的误差的期望
   - 方差衡量的是由于数据采样的随机性可能导致的估计值的波动
3. 考虑均方误差

   $$MSE = \mathbb{E}[(\hat{\theta}_m - \theta)^2]$$

   根据：

   $$\begin{aligned} bias(\hat{\theta}_m)^2 + Var(\hat{\theta}_m) &= [\mathbb{E}(\hat{\theta}_m) - \theta]^2 + \mathbb{E}[(\hat{\theta}_m - \mathbb{E}(\hat{\theta}_m))^2] \\ &= \mathbb{E}(\hat{\theta}_m)^2 + \theta^2 - 2\theta\mathbb{E}(\hat{\theta}_m) + \mathbb{E}[\hat{\theta}_m^2] - \mathbb{E}(\hat{\theta}_m)^2 \\ &= \mathbb{E}[\hat{\theta}_m^2] - 2\theta\mathbb{E}(\hat{\theta}_m) + \theta^2 \\ &= \mathbb{E}[(\hat{\theta}_m - \theta)^2] = MSE \end{aligned}$$

   即：`MSE` 由偏差和方差组成。
4. 偏差、方差与模型容量有关。用`MSE`衡量泛化误差时，增加容量会增加方差、降低偏差
   - 偏差降低，是因为随着容量的增大，模型的拟合能力越强：对给定的训练数据，它拟合的越准确。
   - 方差增加，是因为随着容量的增大，模型的随机性越强：对不同的训练集，它学得的模型可能差距较大。



2.5 一致性

1. 通常希望当数据集的大小 m 增加时，点估计会收敛到对应参数的真实值。即：

$$\text{plim}_{m\to\infty}\hat{\theta}_m = \theta$$

   - plim 表示依概率收敛。即对于任意的 $\epsilon > 0$，当 $m \to \infty$ 时，有：$P(|\hat{\theta}_m - \theta|) > \epsilon \to 0$
2. 上述条件也称做一致性。它保证了估计偏差会随着样本数量的增加而减少。
3. 渐近无偏不一定意味着一致性。

   如：在正态分布产生的数据集中，可以用 $\hat{\mu}_m = x_1$ 作为 μ 的一个估计。

   - 它是无偏的，因为 $\mathbb{E}[x_1] = \mu$，所以不论观测到多少个数据点，该估计都是无偏的
   - 但它不是一致的，因为他不满足 $\text{plim}_{m\to\infty}\hat{\mu}_m = \mu$

三、最大似然估计

1. 假设数据集 $\mathbf{X} = \{\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \cdots, \vec{\mathbf{x}}_m\}$ 中的样本独立同分布地由 $p_{data}(\vec{\mathbf{x}})$ 产生，但是该分布是未知的。

   $p_{model}(\vec{\mathbf{x}};\theta)$ 是一族由 θ 参数控制的概率分布函数族。

   希望通过 $p_{model}(\vec{\mathbf{x}};\theta)$ 来估计真实的概率分布函数 $p_{data}(\vec{\mathbf{x}})$，也就是要估计 θ 参数。

3.1 准则

1. 最常用的估计准则是：最大似然估计。即：

$$\theta_{ML} = \arg\max_{\theta} p_{model}(\mathbf{X};\theta) = \arg\max_{\theta}\prod_{i=1}^{m} p_{model}(\vec{\mathbf{x}}_i;\theta)$$

2. 由于概率的乘积会因为很多原因不便使用（如容易出现数值下溢出），因此转换为对数的形式：

$$\theta_{ML} = \arg\max_{\theta}\sum_{i=1}^{m}\log p_{model}(\vec{\mathbf{x}}_i;\theta)$$

   因为 m 与 θ 无关，因此它也等价于：

$$\theta_{ML} = \arg\max_{\theta} \sum_{i=1}^{m} \frac{1}{m} \log p_{model}(\vec{\mathbf{x}}_i;\theta)$$

3. 由于数据集的经验分布为：

$$\hat{p}_{data}(\vec{\mathbf{x}}) = \frac{1}{m}\sum_{i=1}^{m}\delta(\vec{\mathbf{x}}-\vec{\mathbf{x}}_i)$$

其中 $\delta(\cdot)$ 为狄拉克函数。因此：

$$\theta_{ML} = \arg\max_{\theta} \mathbb{E}_{\vec{\mathbf{x}}\sim\hat{p}_{data}} \log p_{model}(\vec{\mathbf{x}};\theta)$$

3.2 散度

1. 考虑数据集的经验分布 $\hat{p}_{data}$ 和真实分布函数的估计量 p_{model} 之间的差异，`KL` 散度为：

$$D|_{KL}(\hat{p}_{data}||p_{model};\theta) = \mathbb{E}_{\vec{\mathbf{x}}\sim\hat{p}_{data}}[\log \hat{p}_{data}(\vec{\mathbf{x}}) - \log p_{model}(\vec{\mathbf{x}};\theta)]$$

2. 由于 $\log \hat{p}_{data}(\vec{\mathbf{x}})$ 与 θ 无关，因此要使得 $D|_{KL}(\hat{p}_{data}||p_{model};\theta)$ 最小，则只需要最小化

$$\mathbb{E}_{\vec{\mathbf{x}}\sim\hat{p}_{data}}[-\log p_{model}(\vec{\mathbf{x}};\theta)]$$

也就是最大化

$$\mathbb{E}_{\vec{\mathbf{x}}\sim\hat{p}_{data}} \log p_{model}(\vec{\mathbf{x}};\theta)$$

因此：**最大似然估计就是最小化数据集的经验分布 $\hat{p}_{data}$ 和真实分布函数的估计量 p_{model} 之间的差异**

3.3 条件概率

1. 最大似然估计可以扩展到估计条件概率。

假设数据集 $\mathbf{X} = \{\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \cdots, \vec{\mathbf{x}}_m\}$，对应的观测值为 $\mathbf{Y} = \{y_1, y_2, \cdots, y_m\}$。

则条件概率的最大似然估计为：

$$\theta_{ML} = \arg\max_{\theta} P(\mathbf{Y} \mid \mathbf{X};\theta)$$

2. 如果样本是独立同分布的，则可以分解成：

$$\theta_{ML} = \arg\max_{\theta} \sum_{i=1}^{m} \log P(y_i \mid \vec{\mathbf{x}}_i;\theta)$$

3.4 性质

1. 最大似然估计有两个很好的性质：
   - 在某些条件下，最大似然估计具有一致性。

     这意味着当训练样本数量趋向于无穷时，参数的最大似然估计依概率收敛到参数的真实值。

     这些条件为：

        - 真实分布 p_{data} 必须位于分布函数族 $p_{model}(\cdot;\theta)$ 中。否则没有估计量可以表示 p_{data}
        - 真实分布 p_{data} 必须对应一个 θ 值。否则从最大似然估计恢复出真实分布 p_{data} 之后，也不能解出参数 θ
    - 最大似然估计具有很好的统计效率`statistic efficiency`。即只需要较少的样本就能达到一个良好的泛化误差。
2. 最大似然估计通常是机器学习中的首选估计准则。
3. 当样本数量太少导致过拟合时，正则化技巧是最大似然的有偏估计版本。

四、贝叶斯估计

4.1 贝叶斯估计 vs 最大似然估计

1. 在最大似然估计中，频率学派的观点是：真实参数 θ 是未知的固定的值，而点估计 $\hat{\theta}$ 是随机变量（因为数据是随机生成的，所以数据集是随机的）。
2. 在贝叶斯估计中，贝叶斯学派认为：数据集是能够直接观测到的，因此不是随机的。而真实参数 θ 是未知的、不确定的，因此 θ 是随机变量。
    - 对 θ 的已知的知识表示成先验概率分布 $p(\theta)$ ：表示在观测到任何数据之前，对于参数 θ 的可能取值的一个分布。

      在机器学习中，一般会选取一个相当宽泛的（熵比较高）的先验分布，如均匀分布。
    - 假设观测到一组数据 $\mathbf{X}=\{\vec{\mathbf{x}}_1,\vec{\mathbf{x}}_2,\cdots,\vec{\mathbf{x}}_m\}$ ，根据贝叶斯法则，有：

$$p(\theta\mid\mathbf{X})=\frac{p(\mathbf{X}\mid\theta)p(\theta)}{p(\mathbf{X})}$$

3. 贝叶斯估计与最大似然估计有两个重要区别：
    - 贝叶斯估计预测下，一个样本的分布为：

$$p(\vec{\mathbf{x}}_{m+1}\mid\vec{\mathbf{x}}_1,\vec{\mathbf{x}}_2,\cdots,\vec{\mathbf{x}}_m)=\int p(\vec{\mathbf{x}}_{m+1}\mid\theta)p(\theta\mid\vec{\mathbf{x}}_1,\vec{\mathbf{x}}_2,\cdots,\vec{\mathbf{x}}_m)d\theta$$

      而最大似然估计预测下，一个样本的分布为： $p_{model}(\vec{\mathbf{x}};\theta)$
    - 贝叶斯估计会使得概率密度函数向着先验概率分布的区域偏移。
4. 当训练数据有限时，贝叶斯估计通常比最大似然估计泛化性能更好。

   当训练样本数量很大时，贝叶斯估计往往比最大似然估计计算代价较高。

4.2 最大后验估计

1. 有时候希望获取参数 θ 的一个可能的值，而不仅仅是它的一个分布。此时可以通过最大后验估计`MAP`选择后验概率最大的点：

$$\theta_{MAP}=\arg\max_{\theta}p(\theta\mid\mathbf{X})=\arg\max_{\theta}\left[\log p(\mathbf{X}\mid\theta)+\log p(\theta)\right]$$

2. 最大后验估计具有最大似然估计没有的优势：拥有先验知识带来的信息。

   该信息有助于减少估计量的方差，但是增加了偏差。
3. 一些正则化方法可以被解释为最大后验估计，正则化项就是对应于 $\log p(\theta)$ 。
    - 并非所有的正则化方法都对应为某个最大后验估计。

如：有些正则化项依赖于数据，则显然不是一个先验概率分布

4. 最大后验估计估计 `MAP` 提供了一个直观的方法去设计复杂的、可解释的正则化项。
    - 更复杂的正则化项可以通过先验分布为混合高斯分布得到（而不仅仅是一个单独的高斯分布）

五、随机梯度下降

1. 随机梯度下降是梯度下降的一个扩展，几乎所有的深度学习算法都用到了该算法。

5.1 总梯度

1. 机器学习算法中，损失函数通常可以分解成每个样本的损失函数之和。如：

$$J(\theta)=\mathbb{E}_{\vec{\mathbf{x}},y\sim\hat{p}_{data}}L(\vec{\mathbf{x}},y;\theta)=\frac{1}{m}\sum_{i=1}^{m}L(\vec{\mathbf{x}}_i,y_i;\theta)$$

   其中 L 是每个样本的损失函数 $L(\vec{\mathbf{x}},y;\theta)=-\log p(y\mid\vec{\mathbf{x}};\theta)$。

2. 总损失函数的梯度为：

$$\nabla_\theta J(\theta)=\frac{1}{m}\sum_{i=1}^{m}\nabla_\theta L(\vec{\mathbf{x}}_i,y_i;\theta)$$

    - 其时间复杂度为 $O(m)$。
    - 当训练集规模为数十亿时，计算一步梯度将消耗非常长的时间。

5.2 随机梯度

1. 随机梯度下降法的核心是：梯度就是 L 关于 $\hat{p}_{data}$ 在所有样本上的期望；而期望可以用小规模的样本来近似估计。
2. 具体来说：在算法的每一步，从训练集中抽取小批量样本 $\mathbb{B}=\{\vec{\mathbf{x}}_{b_1},\cdots,\vec{\mathbf{x}}_{b_{m'}}\}$
    - 其中 m' 通常是一个相对较小的数（通常从 1 到几百）
    - 不论 m 为多大， m' 通常是固定的（比如在拟合几十亿的样本时，每次更新计算只使用几百个样本）。

   梯度的估计可以表示成：

$$g=\frac{1}{m'}\nabla_\theta\sum_{i=1}^{m'}L(\vec{\mathbf{x}}_i,y_i;\theta)$$

   然后随机梯度下降法的迭代为（ϵ 为学习率）：

$$\theta\leftarrow\theta-\epsilon g$$

七、传统机器学习的挑战

1. 传统机器学习算法的两个困难：
    - 维数灾难
    - 选择性偏好

7.1 维数灾难

1. 当数据的维数很高时，很多机器学习问题变得相当困难。因为许多传统机器学习算法简单地假设：**一个新样本的输出应该大致与最接近的训练样本的输出相同 。**

7.2 选择性偏好

1. 某些算法偏好于选择某类函数。
2. 最广泛的隐式偏好是：要学习的函数是平滑的或者局部不变性的
    - 这个先验知识表明：要学习的函数不会在一个小区域内发生较大的变化。
    - 很多简单算法完全依赖此先验知识来达到良好的泛化

八、低维流形

1. 流形：指的是连接在一起的区域。如位于三维空间中的一个曲面就是一个流形。

8.1 低维流形假设

1. 如果期望机器学习算法处理 $\mathbb{R}^n$ 上的所有输入时，很多机器学习问题看起来都是无解的。

   低维流形假设：$\mathbb{R}^n$ 上的大部分区域都是无效的输入，感兴趣的输入只分布在 $\mathbb{R}^n$ 中的某一组流行中
2. 数据位于低维流形中的假设不一定总是正确的。但是在人工智能某些场景中，如涉及到图像、声音、文本时，流形假设是对的。因为：
    - 现实生活中有意义的图片、文本、声音的概率分布都是高度集中的。
    - 每个样本都被其他高度相似的样本包围，可以通过应用变换来遍历该流形。

      如：一张人脸的图片，逐渐移动或者旋转图中的像素变成另一张人脸图片。

8.2 流形坐标

1. 当数据位于低维流形中时，通常使用流形坐标，而不是 $\mathbb{R}^n$ 中的坐标。
2. 日常生活中，道路就是一维流形。使用道路编号，而不是三维空间的坐标去定位。
3. 提取低维流形中的坐标是非常具有挑战性的