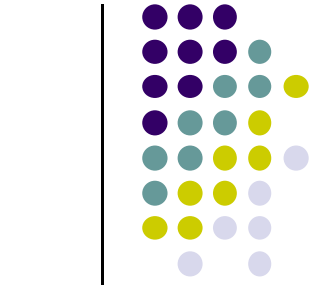
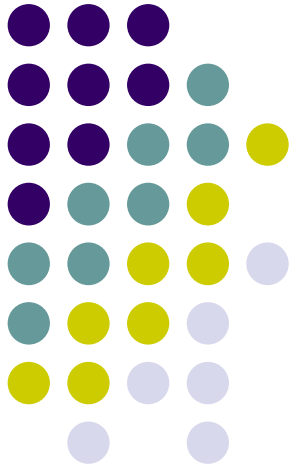


网络优化

哈尔滨工业大学计算学部 刘远超



网络优化

本章内容简介

- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

网络优化

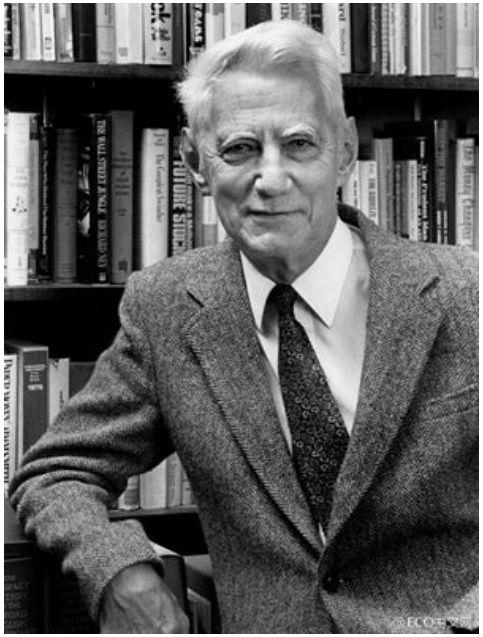
本章内容简介

- **信息论中的熵；**
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

信息论中的熵

信息熵

- 热力学中的熵: 是表示分子状态混乱程度的物理量
- 信息论中的熵： **用来描述信源的不确定性的大小**
- 经常使用的熵概念有下列几种:
  - 信息熵
  - 交叉熵
  - 相对熵
  - 条件熵
  - 互信息



克劳德·艾尔伍德·香农（Claude Elwood Shannon，1916年4月30日—2001年2月24日）是美国数学家、信息论的创始人。1936年获得密歇根大学学士学位 。1940年在麻省理工学院获得硕士和博士学位，1941年进入贝尔实验室工作。香农提出了信息熵的概念，为信息论和数字通信奠定了基础。

信息论中的熵

信息熵

- **信源信息的不确定性函数 f 通常满足两个条件：**

  1)是概率 p 的单调递减函数。

  2)两个独立符号所产生的不确定性应等于各自不确定性之和，即 $f(p_1,p_2)=f(p_1)+f(p_2)$。

- 对数函数同时满足这两个条件：$f(p)=log\frac{1}{p}=-log\,p$

- **信息熵：**要考虑信源所有可能发生情况的平均不确定性。若信源符号有n种取值: $U_1,\dots,U_i,\dots,U_n$，对应概率为$p_1,\dots,p_i,\dots,p_n$，且各种出现彼此独立。此时**信源的平均不确定性应当为单个符号不确定性**$-\log p_i$**的统计平均值(E)，称为信息熵**，即

$$H(U)=E[-\log p_i]=-\sum_{i=1}^{n}p_i log p_i=\sum_{i=1}^{n}\boldsymbol{p_i log(\frac{1}{p_i})}$$

信息论中的熵

交叉熵(cross entropy)

- **定义：**交叉熵是信息论中一个重要的概念,用于表征两 个变量概率分布P, Q（假设P表示真实分布, Q为模型预测的分布）的差异性 。交叉熵越大 , 两个变量差异程度越大。
- 交叉熵公式：

$$H(P,Q) = -\sum_{x \in X} P(x) log Q(x) = \sum_{x \in X} P(x) log \frac{1}{Q(x)}$$

信息论中的熵

相对熵(relative entropy)

- 也称为KL散度(Kullback–Leibler divergence，简称KLD)、信息散度(information divergence)、信息增益(information gain)。
- **相对熵的定义：**是交叉熵与信息熵的差值。表示用分布Q模拟真实分布P，所需的额外信息。
- 计算公式为

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) log \frac{1}{Q(x)} - \sum_{x \in X} P(x) log \frac{1}{P(x)} = \sum_{x \in X} P(x) log \frac{P(x)}{Q(x)}$$

交叉熵　　信息熵

信息论中的熵

相对熵(relative entropy)举例

- 举例：假设某字符发射器随机发出0和1两种字符。且其真实发出概率分布为A。现在有两人的观察概率分布B与C。各个分布如下：

  A(0)=1/2，A(1)=1/2

  B(0)=1/4，B(1)=3/4

  C(0)=1/8，C(1)=7/8

  则B和C哪个更接近实际分布A？

- 求解过程：

  用公式 $D_{KL}(P||Q) = \sum_{x \in X} P(x) log \frac{P(x)}{Q(x)}$，则

  - $D_{KL}(A||B) = \frac{1}{2}\log\left(\frac{1/2}{1/4}\right) + \frac{1}{2}\log(\frac{1/2}{3/4})$
  - $D_{KL}(A||C) = \frac{1}{2}\log\left(\frac{1/2}{1/8}\right) + \frac{1}{2}\log(\frac{1/2}{7/8})$

  结果：

  - $D_{KL}(A||B)$ =0.14,
  - $D_{KL}(A||C) = 0.41$

信息论中的熵

相对熵的性质

- 相对熵（KL散度）有两个主要的性质：
  - 相对熵（KL散度）不具有对称性，即$D_{KL}(P||Q) \neq D_{KL}(Q||P)$。

    例如 $D_{KL}(A||B) = \frac{1}{2}\log\left(\frac{1/2}{1/4}\right) + \frac{1}{2}\log(\frac{1/2}{3/4})$ =**0.1438,**

    $D_{KL}(B||A) = \frac{1}{4}\log\left(\frac{1/4}{1/2}\right) + \frac{3}{4}\log(\frac{3/4}{1/2})$ =**0.1308**

    即$D_{KL}(A||B) \neq D_{KL}(B||A)$

  - 相对熵具有非负性。 即$D_{KL}(P||Q) \geq 0$

信息论中的熵

JS散度

- **JS散度(Jensen–Shannon divergence)具有对称性：**

  由于KL散度不具对称性，因此JS散度在KL散度的基础上进行了改进。

  现有两个分布p1和p2，其JS散度公式为：

$$JS(P_1||P_2) = \frac{1}{2}KL(P_1||\frac{P_1+P_2}{2}) + \frac{1}{2}KL(P_2||\frac{P_1+P_2}{2})$$

信息论中的熵

联合熵

- 联合熵 (复合熵，Joint Entropy)：
  - 用H(X, Y)表示
  - 两个随机变量X，Y的联合分布的熵, 形成**联合熵**

信息论中的熵

条件熵

- 条件熵（ the conditional entropy ）：**H(X|Y)**表示在已知随机变量Y的条件下随机变量X的不确定性。
- **H(X|Y)= H(X, Y) - H(Y)**，表示(X, Y)的联合熵，减去Y单独发生包含的熵。

推导过程：

① 假设已知$y = y_j$，则 $H(x|y_j) = -\sum_{i=1}^{n} p(x_i|y_j) log p(x_i|y_j)$

② 对于y的各种可能值，需要根据出现概率做加权平均。即

$$H(x|y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(y_j)p(x_i|y_j) log p(x_i|y_j)$$

$$= -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j) log \frac{p(x_i, y_j)}{p(y_j)}$$

$$= H(x, y) - H(y)$$

信息论中的熵

互信息

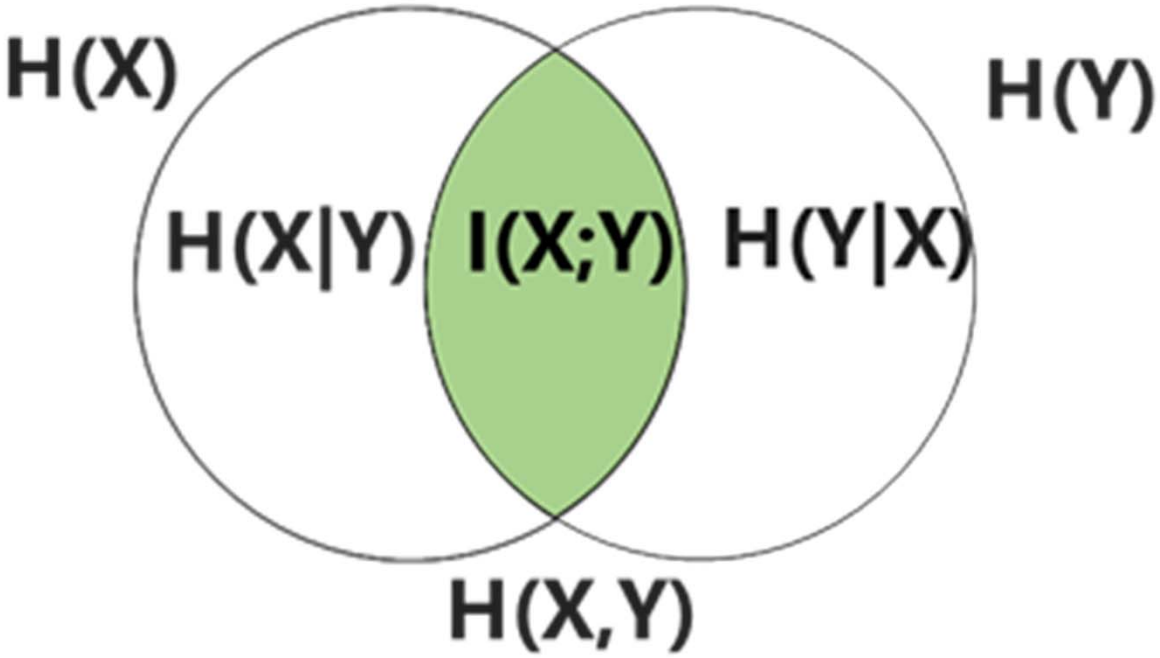
- 互信息(Mutual Information)可以被看成是一个随机变量中包含的关于另一个随机变量的信息量，或者说是一个随机变量由于已知另一个随机变量而减少的不确定性 。

$$I(X;Y) = H(X) - H(X|Y)$$
$$= H(X) + H(Y) - H(X,Y)$$
$$= \sum_{x} p(x) log \frac{1}{p(x)} + \sum_{y} p(y) log \frac{1}{p(y)} - \sum_{x,y} p(x,y) log \frac{1}{p(x,y)}$$
$$= \sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}$$

即互信息$I(X;Y)$是联合分布$p(x,y)$与乘积分布$p(x)p(y)$的相对熵

信息论中的熵

文氏图图解



网络优化

本章内容简介

- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

本章内容简介

- 信息论中的熵；
- **模型优化中的目标函数；**
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

模型优化中的目标函数

通常考虑两个方面，即经验风险和结构风险

- 为了解决机器学习/深度学习中的分类或回归等预测问题，在设计并确定好用于解决问题的模型后，**下一步就是利用训练集中的所有样本数据来确定模型中的参数取值组合**，以便使模型在测试集上取得较好的效果。这就是模型训练和优化的目标。
- 为达成这一目标而定义的被优化的函数称之为**目标函数**。为了达成模型的上述优化目标，通常考虑两个方面，即**经验风险和结构风险**。

模型优化中的目标函数

（一）经验风险（Empirical Risk）

- **什么是经验风险**？假设训练样本集为$T=\{(x_1,y_1),(x_2,y_2),\dots,(x_m,y_m)\}$，则经验风险是假设的模型在训练集上的平均损失：

$$R_{emp}(f)=\frac{1}{m}\sum_{i=1}^{m}L(y_i,f(x_i)) \tag{6-9}$$

其中，$L(y_i,f(x_i))$**定义为每个样本的预测损失函数（loss function），**以衡量样本的预测值$f(x_i)$与真实值y_i之间的差异。

- 与经验风险相对应的是**经验风险最小化策略**，其认为经验风险最小的模型是最优模型，即在此策略下模型参数的求解方法为：

$$\min_{f\in\mathcal{F}}\frac{1}{N}\sum_{i=1}^{N}L(y_i,f(x_i)) \tag{6-10}$$

- **当样本数量足够大时，经验风险最小化策略可以取得较好的效果**。在训练集上的极大似然估计（Maximum Likelihood Estimate，MLE）是经验风险最小化的一个例子。

模型优化中的目标函数

（二）结构风险（structural risk ）

- **结构风险最小化是为了防止过拟合而提出的策略**，其在经验风险最小化的基础上引入参数的正则化（Regularization）来限制模型的复杂度，以使模型避免过度最小化在训练集上的经验风险。
- 因此，结构风险函数形式如下：

$$R_{sr}(f) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f) \qquad (6\text{-}11)$$

其中，**$J(f)$为模型的复杂度**。模型函数f越复杂，$J(f)$的复杂度就越大。

- 与结构风险相对应的是结构风险最小化策略，其认为**最优模型应该是使结构风险而不是使经验风险最小的模型**，即：

$$\min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f) \qquad (6\text{-}12)$$

其中$\lambda \geq 0$是权重因子超参数。可见，结构风险的最小化策略对复杂的模型进行了惩罚，**以便使模型能对训练集和未知的测试样本都进行较好的预测。**

模型优化中常用的经验损失函数

交叉熵损失函数和均方误差损失函数

- 在机器学习/深度学习中，**训练的目标就是通过若干次迭代，最小化或者最大化目标函数，使模型达到收敛状态**。其中，无论是经验风险还是结构风险，损失函数都会在模型的优化过程的参数估计中被用到。其计算往往是在训练集或者验证集上的批量数据进行。
- 神经网络中常用的损失函数包括**交叉熵损失函数（常用于分类问题）**和**均方误差损失函数（常用于回归问题）**等。

模型优化中常用的经验损失函数

（一）交叉熵损失函数

很多神经网络模型，如卷积神经网络、循环神经网络等的分类问题基本预测过程为：

1）**多层变换并最终得到分类得分**。输入样本经过若干中间的隐藏层的变换，最后一个隐藏层L得到的是一个向量h_L，该向量将通过与分类参数W_c点乘得到输入样本在各种可能类别上的得分向量S（scores，logits），即

$$\boldsymbol{S = W_c h_L} \tag{6-13}$$

2）**分类得分的归一化处理**。分类得分向量S将进一步通过softmax函数归一化处理以得到概率分布输出Q，其第i个元素为

$$\boldsymbol{Q_i = Softmax_i = \frac{e^{S_i}}{\sum_j e^{S_j}}}\text{，j=1，2，…，}\boldsymbol{K}\text{（}\boldsymbol{K}\text{为类别总数）} \tag{6-14}$$

因此每个元素取值在0和1之间，各元素的总和为1。

模型优化中常用的经验损失函数

（一）交叉熵损失函数(续)

3）交叉熵损失函数计算。即进行模型预测得到的类别概率输出Q与真实类别的独热编码向量（one-hot vector）P进行交叉熵计算：

$$H(P,Q) = -\sum_{x \in X} P(x) log Q(x) = \sum_{x \in X} P(x) \frac{1}{log Q(x)} \qquad (6\text{-}15)$$

输入样本的真实分类标签P采用的独热编码形式为[0，…，1，…，0]，即只有真实标签的索引（设其为i）为1，其余均为0。因此分布P和Q之间的交叉熵计算结果实际上将只剩下Q中的第i个元素，从而变为如下形式：

$$H(P,Q) = -log Q_i = -log \frac{e^{S_i}}{\sum_j e^{S_j}}, \ j=1, 2, \cdots, K \qquad (6\text{-}16)$$

其中，S_i表示模型对真实类别标签给出的预测分数。

模型优化中常用的经验损失函数

（二）均方误差损失函数

- **均方误差**是指预测值$f(x_i|\theta)$和真实值y_i之间的平方差的均值，是回归问题中最常用的损失函数：

$$MSE(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(f(x_i|\theta) - y_i\right)^2 \qquad (6\text{-}17)$$

- **问题：**我们注意到，这两类问题还有更多的性能度量方法，例如度量分类的准确率、精确率、召回率等，度量回归的平均绝对误差、均方误差等，那么是否可以用性能度量方法代替损失函数？

网络优化

本章内容简介

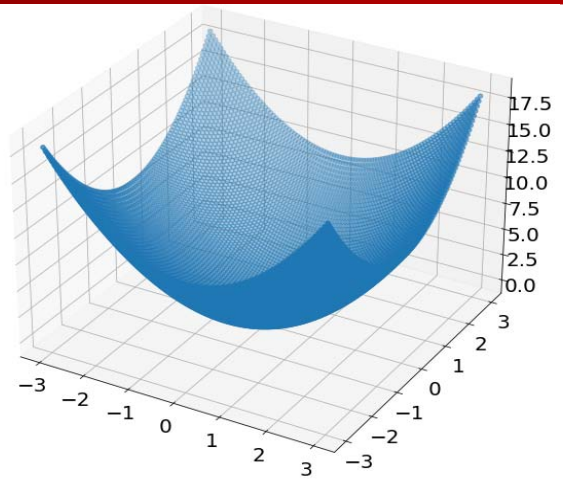
- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

本章内容简介

- 信息论中的熵；
- 模型优化中的目标函数；
- **用梯度下降法求解目标函数的极值；**
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

用梯度下降法求解目标函数的极值

理想的目标函数极值



- 定义了模型的优化目标函数以后，下一步需要找到能使目标函数取极值的参数取值组合。**梯度下降**是利用目标函数一阶梯度信息找到函数局部最优解的一种方法：模型中每个参数θ_j的更新方法为：

$$\theta_j = \theta_j - \eta \cdot \frac{\partial J(\theta)}{\partial \theta_j}$$ ，其中，η为学习速率。

- **三种梯度下降**：批量梯度下降、随机梯度下降、小批量随机梯度下降 ⇨

模型优化中的目标函数

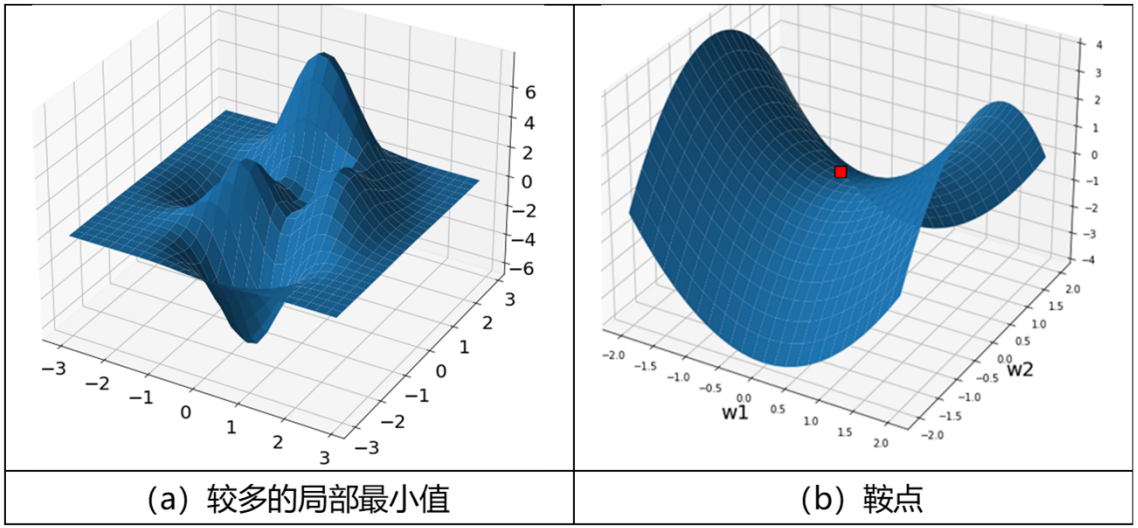
（一）批量梯度下降

- 批量梯度下降法是梯度下降最原始的形式，因而也称为香草（Vanilla）梯度下降。**其特点是每一次迭代使用训练数据集中的全部样本**来计算目标函数相对于参数θ的梯度。
- **批量梯度下降法的优点**：
  - 每次迭代使用的样本数量较多，泛化能力好；
  - 利用矩阵处理并行计算可以弥补由于样本数量多带来的开销问题。

模型优化中的目标函数

（一）批量梯度下降（续）

- 但批量梯度下降法也存在**明显的缺点**：
  - 首先就是**对所有训练样本处理完毕后才只完成一次迭代**，并进行一次参数更新。因而当样本数量很大时效率较为低下、收敛速度也容易受到影响，对内存容量也有一定要求。也不适合模型的在线更新。
  - 除此之外，批量梯度下降法更**容易受到局部最小值和鞍点问题的困扰**：



（a）较多的局部最小值　　（b）鞍点

马鞍形中心位置是参数w_1的局部最小值，但却是参数w_2的局部最大值。

模型优化中的目标函数

（二）随机梯度下降

- 随机梯度下降是与批量梯度下降相区别而言，即每次迭代时不是利用训练集中全部样本来计算梯度，**而是只使用一个训练样本$x^{(i)}$及其标签$y^{(i)}$来更新参数**，即模型中每个参数θ_j的更新方法为

$$\boldsymbol{\theta_j = \theta_j - \eta \cdot \frac{\partial J(\theta; x^{(i)}; y^{(i)})}{\partial \theta_j}} \qquad (6\text{-}19)$$

- 因此，随机梯度下降试图过引入随机性来避开局部最小值和鞍点。

模型优化中的目标函数

（三）小批量随机梯度下降

- 小批量梯度下降方法相当于**介于之前介绍的批量梯度下降与随机梯度下降之间的一种折中策略**。
- 其基本思想是，每次迭代过程只利用训练集中的**某个固定数量的小规模子集**来对参数进行更新。假设小批量中含有n个样本，则模型中每个参数θ_j的更新方法为：

$$\theta_j = \theta_j - \eta \cdot \frac{\partial J(\theta; x^{(i:i+n)}; y^{(i:i+n)})}{\partial \theta_j} \tag{6-20}$$

- 因此小批量随机梯度下降的基本思想在于其试图结合批量梯度下降和随机梯度下降的优点。

网络优化

本章内容简介

- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

网络优化

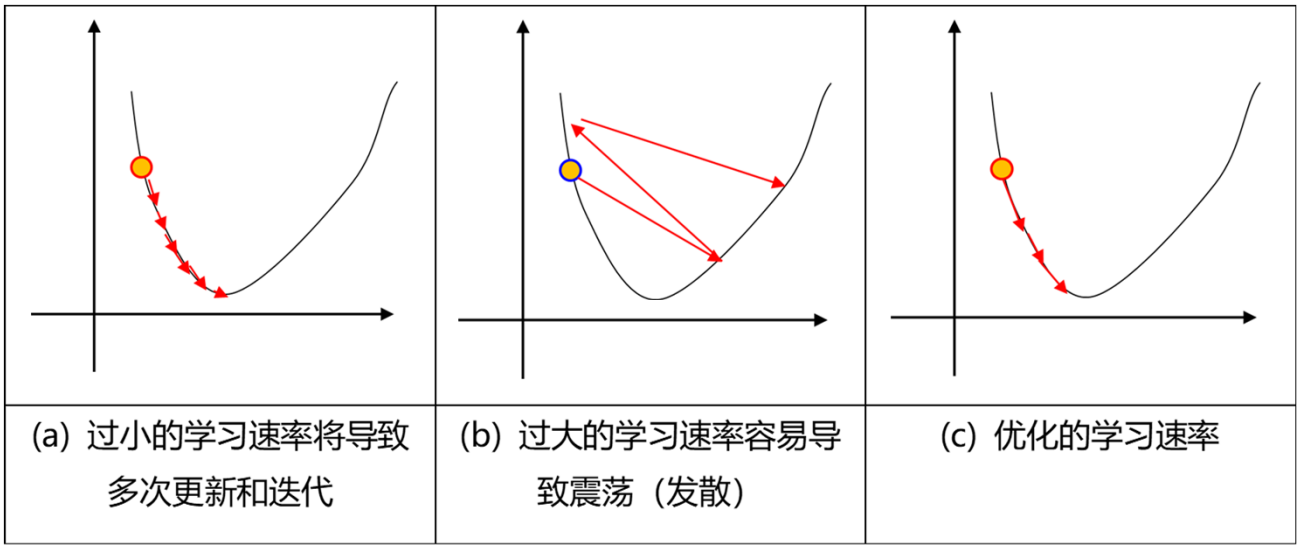
本章内容简介

- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- **梯度下降法中学习率的调节方法；**
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

梯度下降法的学习率

学习率的动态调整

- 梯度下降法优化目标函数中用到的一个超参数是学习率（learning rate），其控制每次迭代过程中参数值的更新幅度。



(a) 过小的学习速率将导致多次更新和迭代

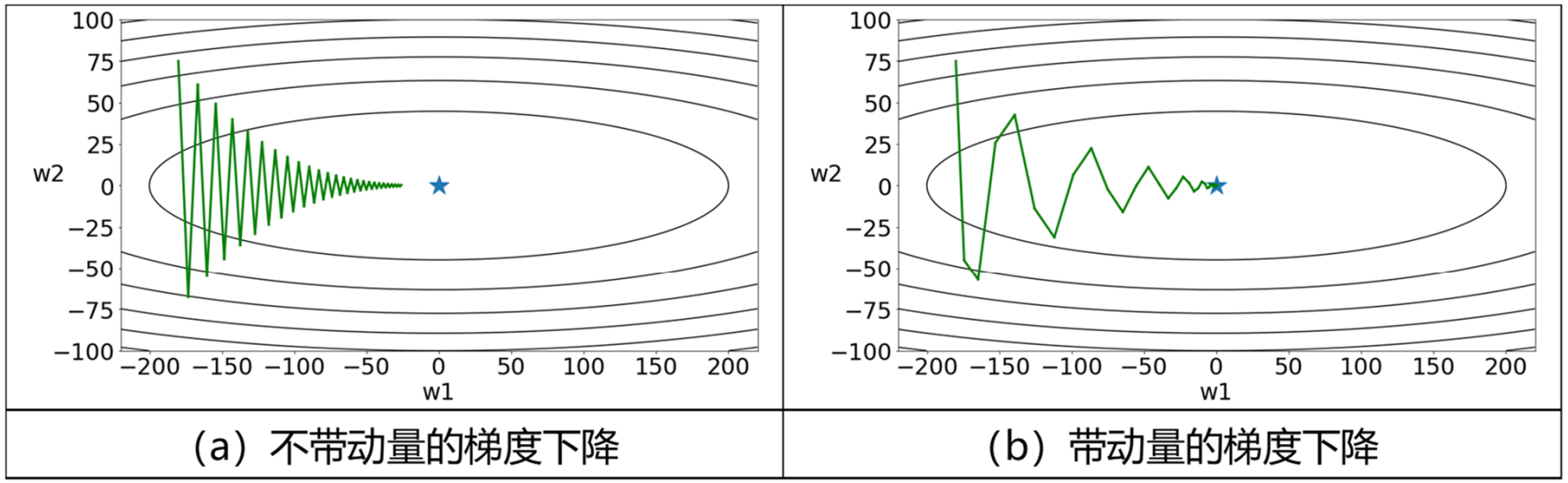
(b) 过大的学习速率容易导致震荡（发散）

(c) 优化的学习速率

- 学习率可以采取如下方式动态调整：1）有序调整，即**根据预定义的计划调整学习率**，如等间隔调整、指数衰减，余弦退火等；2）**基于性能监测的自适应调整**。监测模型预测性能的变化，并据此调整学习率；3）**对不同的参数采取不同的学习率。**
- 下面介绍**几种常见的梯度下降学习率调节方法**，包括动量梯度下降法、Nesterov加速梯度法、自适应梯度算法、均方根传播等。

模型优化中的目标函数

动量梯度下降法（Momentum）



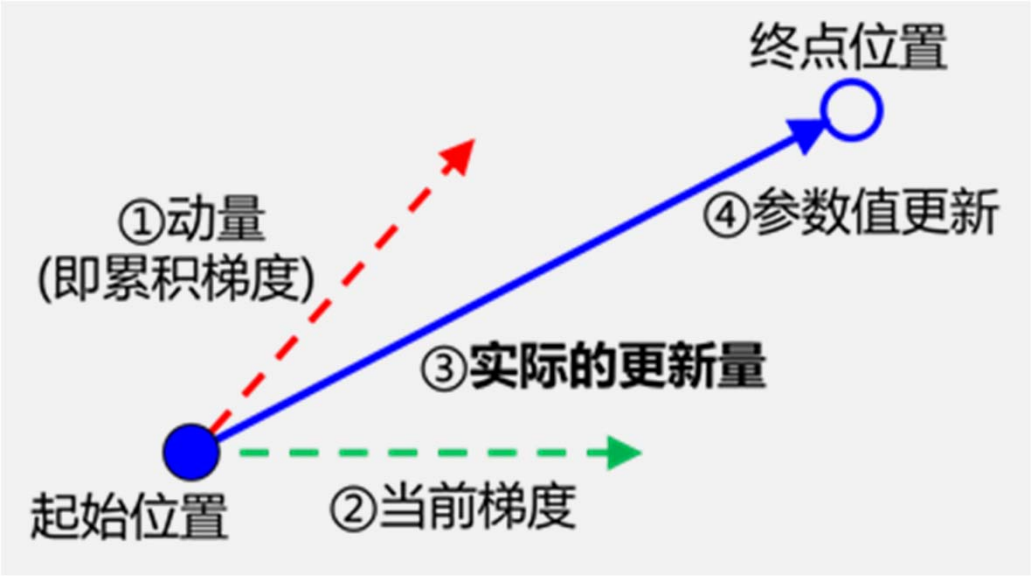
（a）不带动量的梯度下降　　（b）带动量的梯度下降

- 动量梯度下降法（简称动量法）针对的问题：
  - 假设图中的横轴和纵轴分别为损失函数需要优化的两个参数w_1和w_2，不同等高线为不同的目标函数取值，图中中间的“★”点为优化取值点。
  - 则由于纵轴（w_2）方向上的弯曲程度（梯度）远大于横轴（w_1），此时普通的梯度下降法很难快速通过这一陡谷区域到达局部最优点，其表现是目标函数取值将在较陡的纵轴维度（w_2）上发生波动，并在横轴维度（w_1）上向局部最优点方向缓慢前进。

模型优化中的目标函数

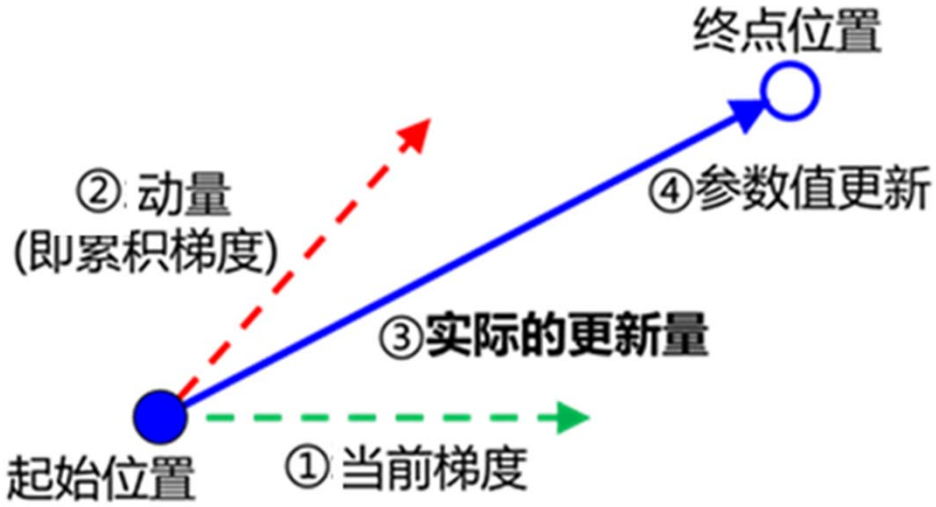
动量梯度下降法（Momentum）

- **动量法的基本原理**：动量法的技术实现借鉴了物理学中的动量概念[Sutton, 1986]：物体的动量是指其在运动方向上保持运动的趋势，其是一种矢量。因此**动量法是指每次迭代获得的参数更新不仅取决于当前位置的梯度，还受到上一次迭代的参数更新的影响。**
- **动量法的形式化描述**：对于某个参数θ_i，其在当前步（第t步）的更新量$v_{t,i}$除了与目标函数对θ_i的梯度有关（体现标准梯度下降法的思想）外，**还与上次（第$t-1$步）的更新量有关（体现引入动量）**，如图6.6所示。



模型优化中的目标函数

动量梯度下降法（Moment



结合图6.6，动量梯度下降法的基本步骤为：

1. **计算当前步的梯度：**

$$g_{t,i} = \frac{\partial J(\theta_{t-1})}{\partial \theta_i} \qquad (6\text{-}21)$$

2. **计算至上一步为止的动量（即累积梯度）：**

$$v_{t-1,i} = g_{t-1,i} + \beta v_{t-2,i} \qquad (6\text{-}22)$$

其中，超参数$\beta \in [0,1]$为动量项系数，表示上一次更新的衰减权重。

3. **计算当前步（第t步）的更新：**

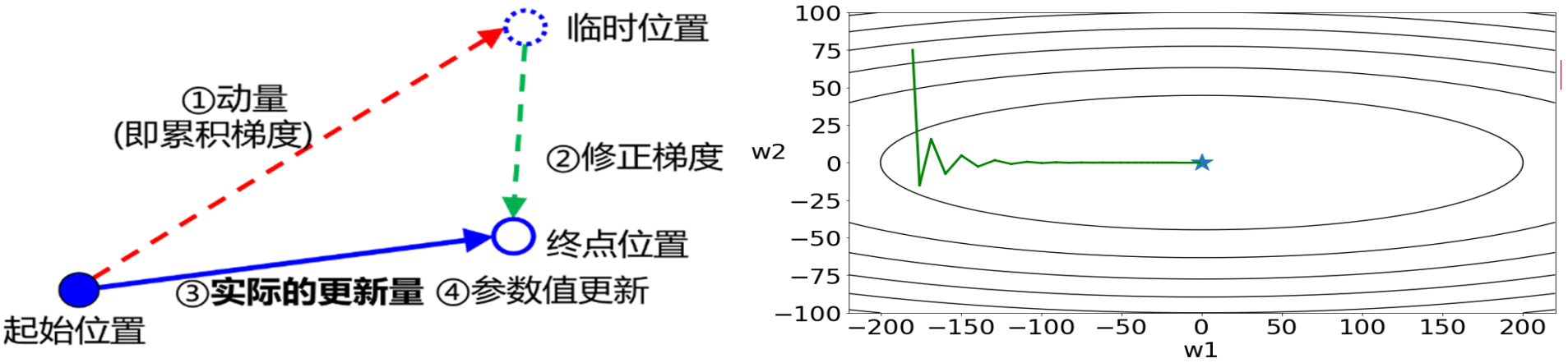
$$v_{t,i} = g_{t,i} + \beta v_{t-1,i} \qquad (6\text{-}23)$$

4. **最后，完成当前迭代中参数θ_i的更新：**

$$\theta_{t,i} = \theta_{t-1,i} - \alpha v_{t,i} \qquad (6\text{-}24)$$

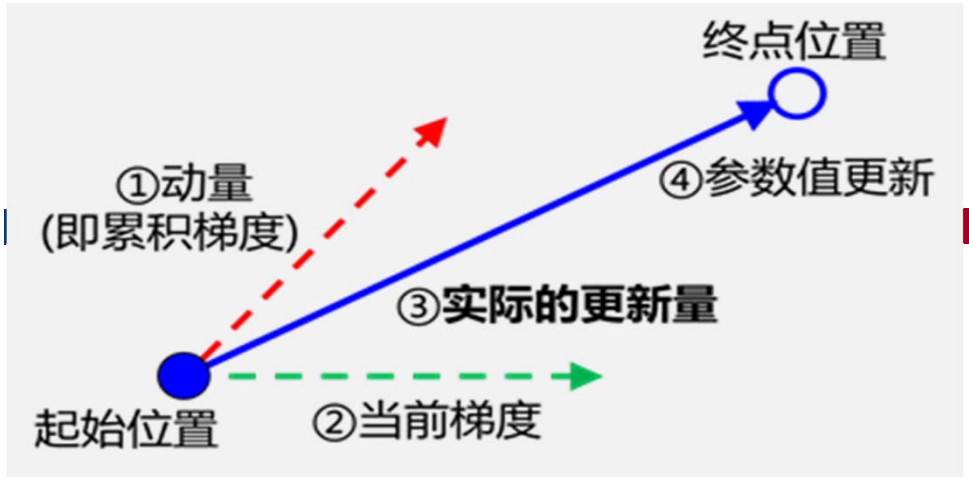
其中，超参数α为学习率。

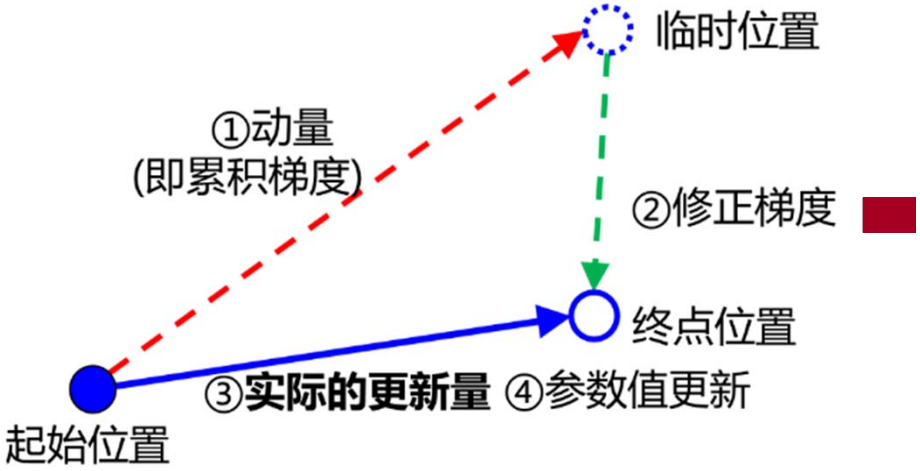
Nesterov加速梯度法(NAG)



- Nesterov加速梯度法是对标准动量梯度下降法的改进，其初衷是**进一步加速对震荡的惩罚，以快速收敛**[Sutskever,2013]。具体做法是，**梯度更新计算不是在当前位置，而是在一个超前动量的临时位置处进行**。图6.7示意了Nesterov动量梯度下降法，可以看出其与标准动量梯度下降法（图6.6）的区别。

- 图6.8 给出了引入Nesterov动量法的梯度下降收敛效果示意图。可见，其相对于图6.5（b），震荡得以快速减弱，收敛过程也得以加速。





- 在**左图所示的标准动量梯度下降法**中，是将当前位置的动量（即累积梯度）与当前位置的梯度进行加权求和获得实际的更新矢量，从而使得参数取值从图中的“起始位置”移动到“终点位置”。
- 而**右图中Nesterov动量梯度下降法**中参数更新量的计算可分解为如下几步：

1. 先在起始位置利用当前动量（即累积梯度）获得临时的参数更新，从而得到一个临时的参数取值θ'（即到达图中的“临时位置”）。
2. **基于θ'计算修正梯度：**
3. 基于动量$v_{t-1,i}$和修正梯度$g'_{t,i}$得到实际的更新量
4. 最后，基于实际更新量$v_{t,i}$使参数θ_i取值从图中的“起始位置”移动到“终点位置”：**因此，经过此步骤，才最终完成实际的参数迭代更新，而第1步中的参数更新并未实际投入使用，只是中间步骤。**

模型优化中的目标函数

自适应梯度算法（AdaGrad）

- AdaGrad**的基本思想是对不同参数使用不同的学习速率**。每个参数的实际学习率在训练刚开始时比较大，以便快速梯度下降；随着优化过程的进行，对于下降幅度较大的参数减慢其实际学习率，而对于下降幅度较小的参数提高其实际学习率。

- **Adagrad的技术原理**是，根据之前时间步的累计梯度，调整每个参数θ_i下一个时间步t的实际学习率，因此$\theta_{t,i}$的更新方法为：

$$\begin{cases} \theta_{t,i} = \theta_{t-1,i} - \dfrac{\alpha}{\sqrt{G_{t-1,ii}+\varepsilon}} \cdot g_{t-1,i} \\ G_{t-1,ii} = \sum_{\tau=1}^{t-1} g_{\tau,i}^2 \\ g_{t-1,i} = \dfrac{\partial J(\theta)}{\partial \theta_{t-1,i}} \end{cases} \tag{6-29}$$

其中，α为全局学习率超参数，$g_{t-1,i}$为参数θ_i在时间步$t-1$处的梯度。$G_{t,ii}$是参数θ_i在之前各个时间步的累积平方梯度，用于产生参数θ_i的学习率调整比例因子，平方再求和的目的是去除梯度符号以防止相互抵消。ε是极小的常数，其作用是防止分母为零。

模型优化中的目标函数

均方根传播（RMSProp）

- RMSProp（Root Mean Square Propagation）即**均方根传播是对AdaGrad算法的一种改进**。其主要针对AdaGrad方法中累积平方梯度值越来越大并导致每个参数的实际学习率过早过量减少的问题。

- **RMSProp的具体技术实现原理**是，使用指数加权移动平均的方法计算累积梯度，以控制之前时间步的累积平方梯度的取值大小。其和AdaGrad唯一的不同是其中的累积平方梯度$G_{t,ii}$的计算改为：

$$G_{t,ii} = \rho \sum_{\tau=1}^{t-1} g_{\tau,i}^2 + (1-\rho) g_{t,i}^2 \tag{6-30}$$

即RMSProp通过引入衰减因子系数ρ（超参数）来控制之前时间步的累积平方梯度的权重，即用于对历史时间步的梯度累积与当前的梯度累积进行平衡。

网络优化

本章内容简介

- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

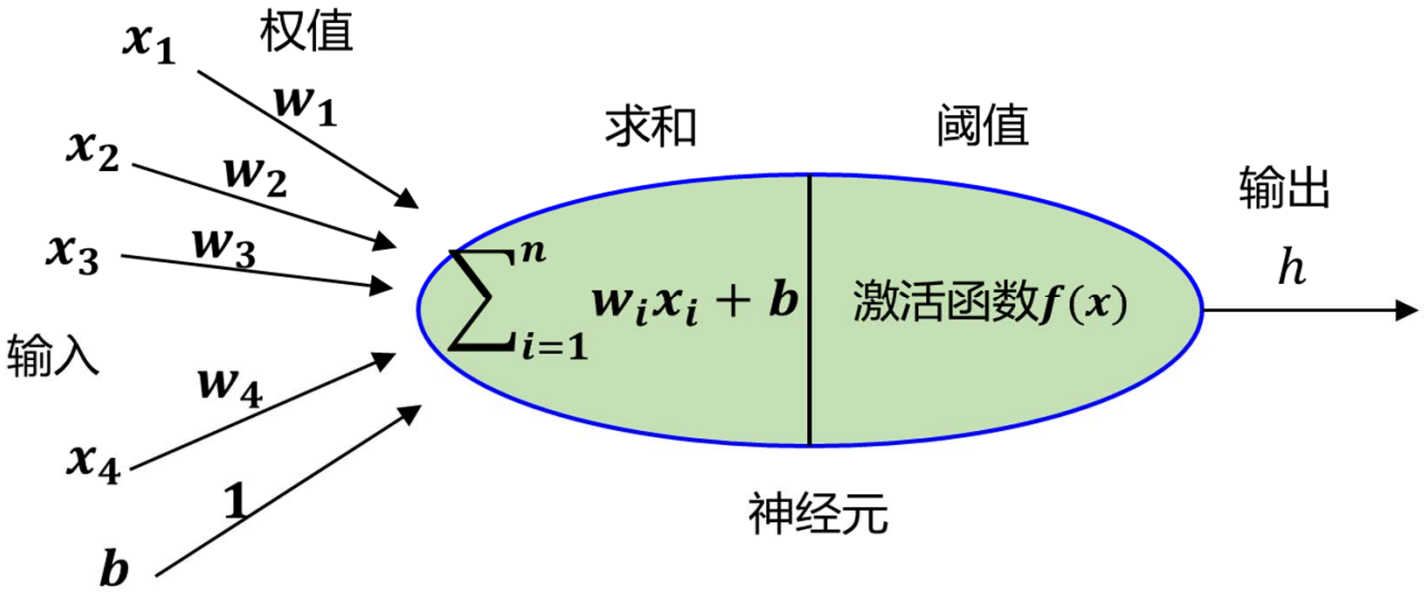
网络优化

本章内容简介

- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- **常见的激活函数；**
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

激活函数

什么是激活函数



- **什么是激活函数。**在多层神经网络中，每层通常是先对前一层的输出进行线性组合和变换，然后再使用一个非线性函数进行处理，并得到输出结果，如图6.9所示。其中的非线性函数就是激活函数（Activation Function）或称激励函数。通过引入激活函数，相当于给神经网络引入了非线性因素，从而理论上可以逼近和模拟任何非线性函数，其表达能力往往强于纯线性模型。

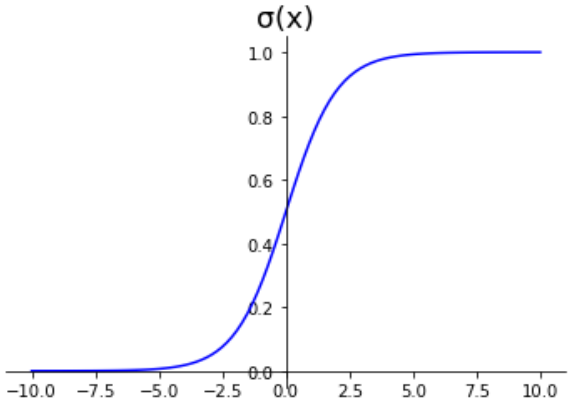
激活函数

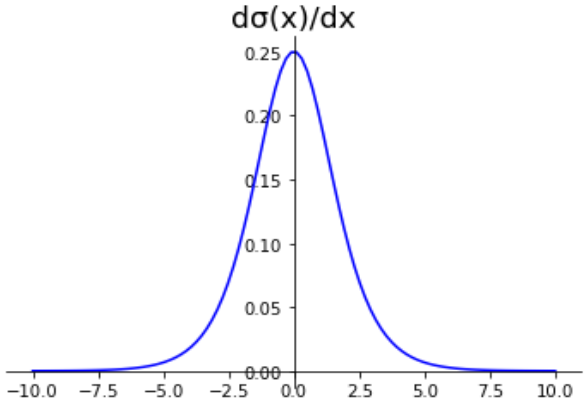
饱和激活函数与非饱和激活函数

- 激活函数也可以分为饱和激活函数与非饱和激活函数。
  - **饱和激活函数。**饱和激活函数分为三种类型：对于激活函数$f(x)$，当x趋近于正无穷，如果$f(x)$的导数趋近于0，即$\lim_{n\to+\infty} f'(x)=0$，则称$f(x)$为**右饱和**；当$x$趋近于负无穷，如果$f(x)$的导数趋近于0，即$\lim_{n\to-\infty} f'(x)=0$，则称$f(x)$为**左饱和**；如果$f(x)$既满足做左饱和又满足右饱和，则称之为**全饱和**。饱和激活函数的例子包括下面即将介绍的Sigmoid、Tanh函数。
  - **非饱和激活函数**是指不满足以上条件的激活函数，例如ReLU及其变体。

激活函数

Sigmoid函数





Sigmoid函数也叫Logistic函数，其主要特点是把输入的连续实值变换为0和1之间的输出，因此也常用来表示概率值。而其输入则通常为线性模型的输出：

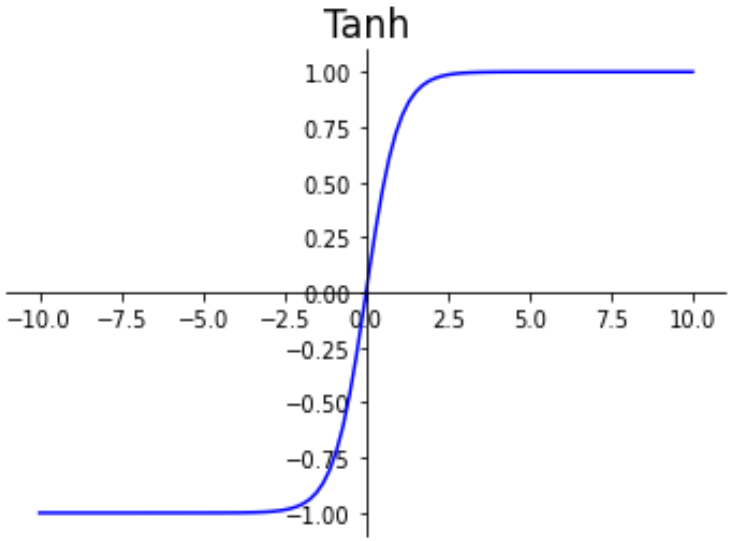
$$\begin{cases} g(z) = \frac{1}{1+e^{-z}} \\ z = h_\theta(x) = \theta^T x \end{cases} \tag{6-31}$$

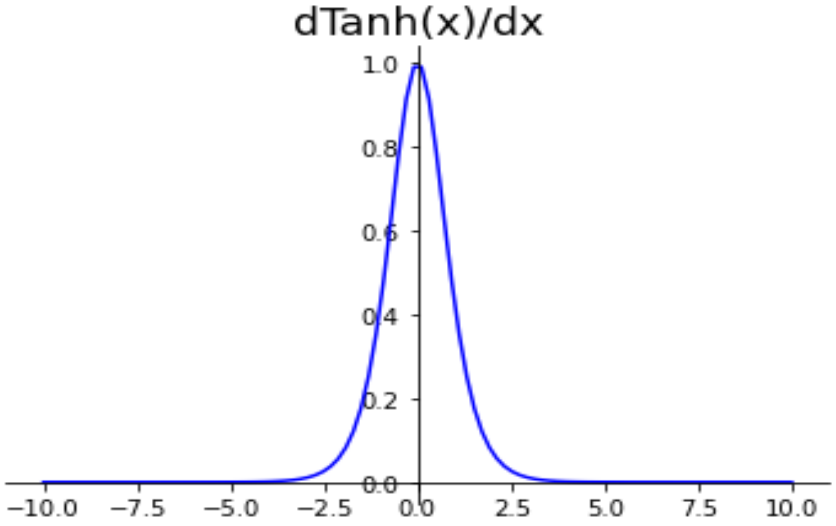
其中$g(z)$就是Sigmoid函数。如图6.10所示，该函数的曲线呈现“S”形，其导数为

$$\frac{d\sigma(z)}{dz} = \frac{e^{-z}}{(1+e^{-z})^2} = \left(\frac{1+e^{-z}-1}{1+e^{-z}}\right)\left(\frac{1}{1+e^{-z}}\right) = (1-\sigma(z))\sigma(z)$$

激活函数

Tanh 函数





Tanh（Hyperbolic Tangent）函数即双曲正切函数。其数学表达式为

$$\mathrm{Tanh}(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}=\frac{1-e^{-2x}}{1+e^{-2x}} \qquad (6\text{-}33)$$

Tanh函数的导数为：$\mathrm{Tanh}'(x)=1-\mathrm{Tanh}^{2}(x)$。图6.11给出了Tanh函数及其导数的曲线。

激活函数

ReLU函数

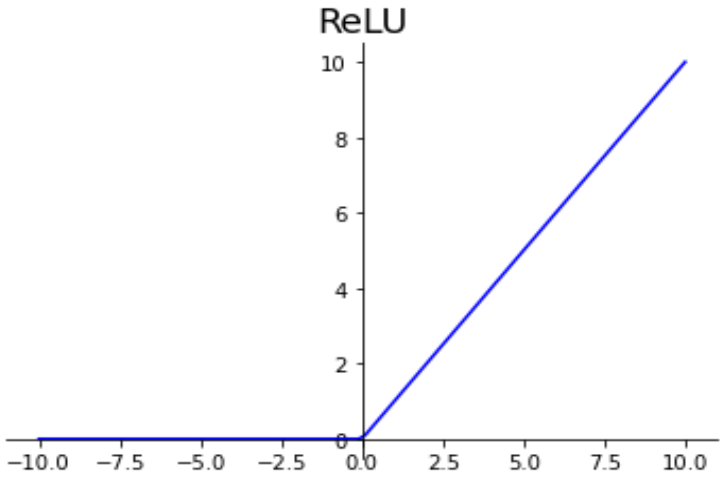


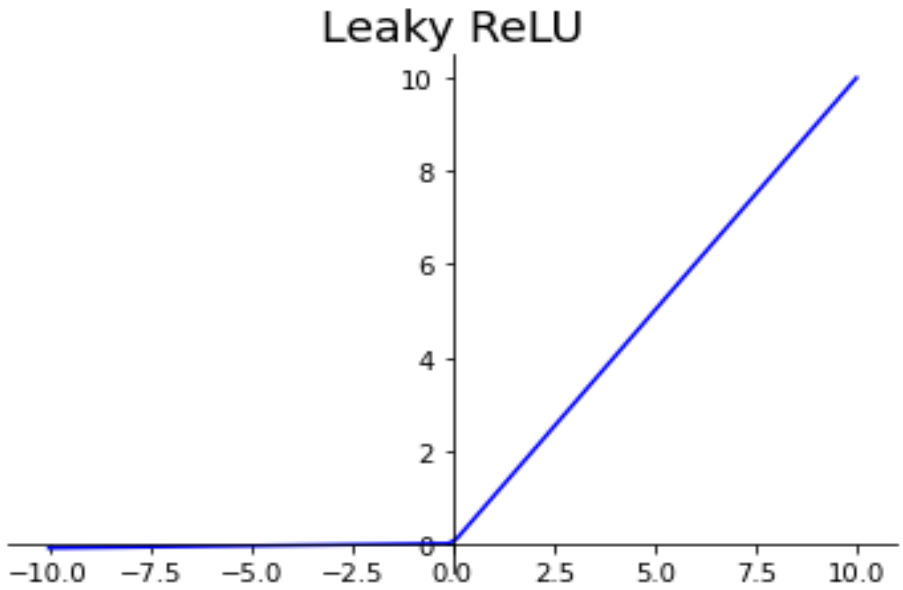
图6.13给出了Relu函数的曲线。可见在神经网络中，Relu从整体上对输入的训练样本空间而言仍然是一种“看似线性”（分段线性）的非线性激活函数。

ReLU（Rectified Linear Unit）是一种修正线性且不饱和的激活函数，其数学表达式为

$$\mathrm{ReLU}(x) = \max(0, x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

激活函数

Leaky ReLU函数

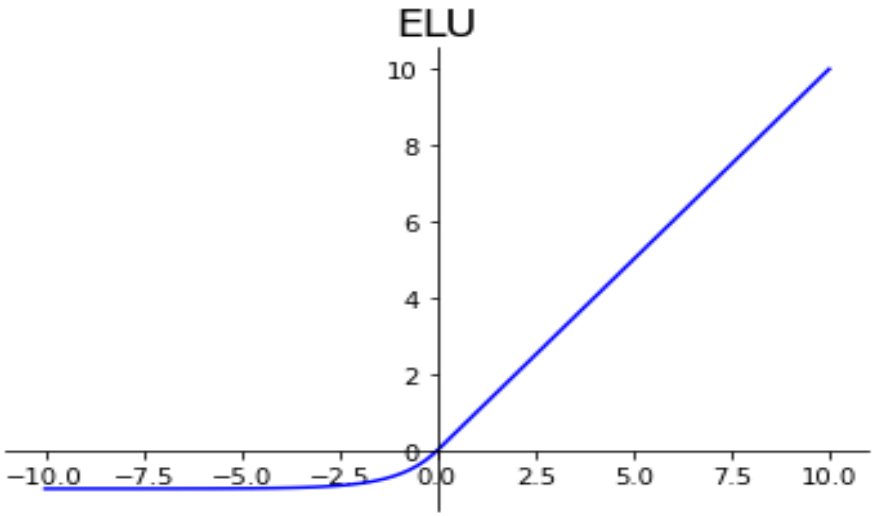


与ReLU相比，Leaky ReLU对负值的输入不是直接赋予0，而是设置了一个非零斜率。图6.14给出了 Leaky ReLU函数曲线，其数学表达式为

$$f(x)=\begin{cases} x & x \geq 0 \\ ax & x < 0 \end{cases} \qquad (6\text{-}35)$$

激活函数

ELU函数



ELU即指数线性单元（Exponential Linear Units），其数学表达式如下所示：

$$f(x)=\begin{cases}x, & x>0\\ a(e^{x}-1), & x\le 0\end{cases} \tag{6-36}$$

其中$a\ge 0$为超参数，用于设置$x\le 0$时的饱和曲线。图6.15 给出了ELU函数曲线。

本章内容简介

- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

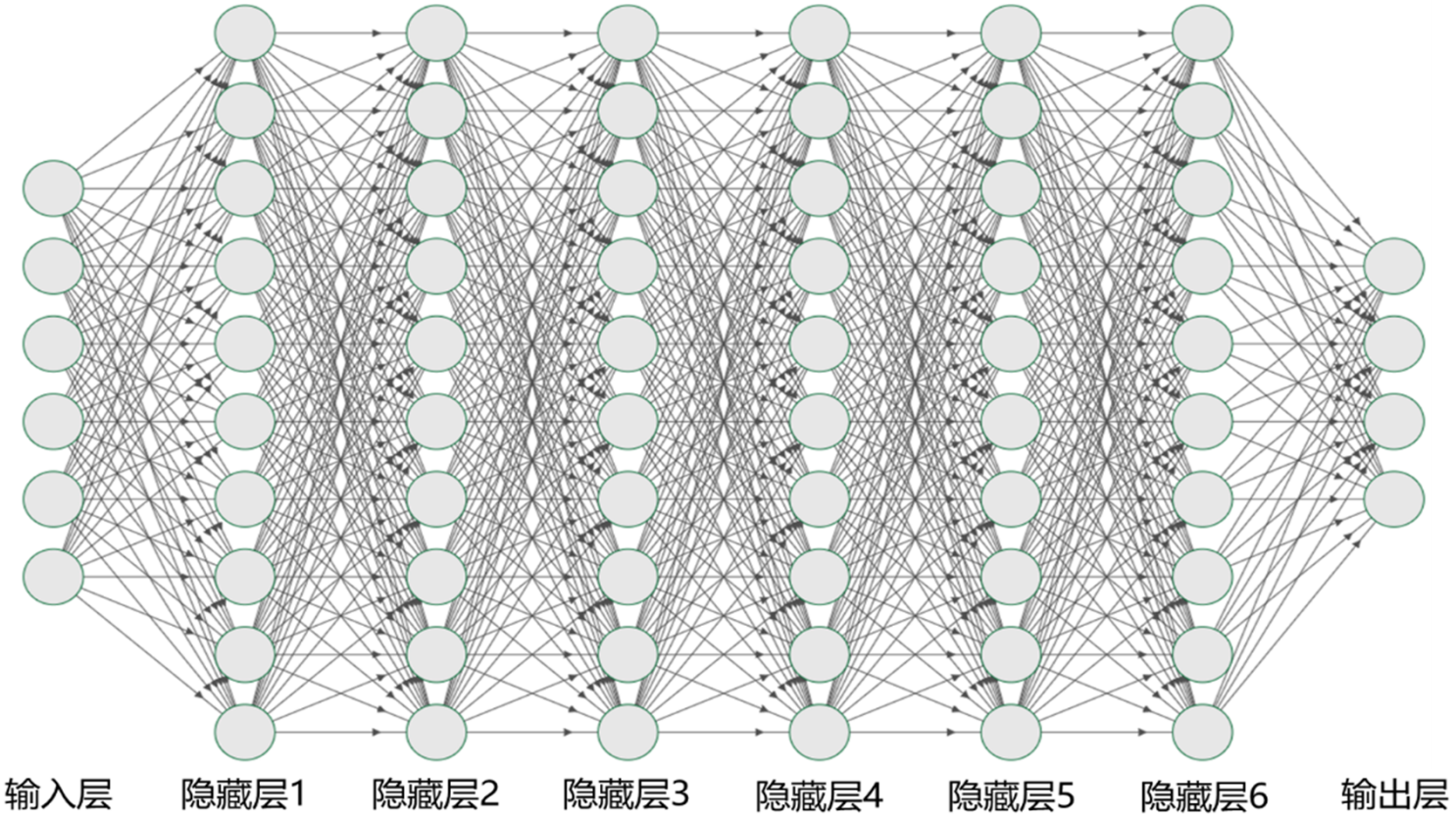
网络优化

本章内容简介

- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- **梯度消失和梯度爆炸问题及应对策略；**
- 欠拟合和过拟合问题及其应对策略。

梯度消失和梯度爆炸问题

梯度消失和梯度爆炸发生的原因



深层神经网络模型（如图6.16所示）在训练时容易出现梯度消失（gradient vanishing problem）或者梯度爆炸问题（gradient exploding problem）。 二者具有共性特征：由于反向传播机制，较小梯度的多次连乘导致梯度消失，较大梯度的多次连乘导致梯度爆炸。

梯度消失和梯度爆炸问题

梯度消失和梯度爆炸问题的解决方法

根据上述梯度消失和梯度爆炸问题产生的原因分析，可以有如下解决方法：

1）**采用合理的激活函数**。目前ReLU激活函数及其变体在隐藏层中较为常用，由于其在正的梯度输入部分的导数为1，则可以消除激活函数偏导数的影响，有助于克服梯度消失和爆炸问题。

2）**梯度剪切**。梯度剪切主要针对梯度爆炸，其基本思想是设置一个梯度阈值，如果更新的梯度超过该阈值，则将其取值强制限制在其范围内，以防止梯度爆炸。

3）**权重正则化**。如果发生梯度爆炸，则将导致权重取值变得非常大，因此通过正则化处理来限制权重的取值，可以在一定程度上防止梯度爆炸的发生。比较常见的权重正则化方法包括L1正则化和L2正则化。

4）**批量标准化（Batch Normalization，BN）**[Ioffe,2015]。批量标准化应用于每层网络的激活函数之前，包括：对向量做标准化（Standardization）处理以使输出满足均值为0，方差为1的分布；然后再进行缩放和平移处理（scale and shift），以提升训练稳定性，解决梯度消失和爆炸问题。

网络优化

本章内容简介

- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。

网络优化

本章内容简介

- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- **欠拟合和过拟合问题及其应对策略。**

欠拟合、过拟合及其应对方法

什么是欠拟合、过拟合

- **从训练数据中学习到模型参数的过程也称为拟合（fit）**。拟合是机器学习的基本问题。机器学习（深度学习）的主要目标是提高模型对训练集以外样本的预测能力，即模型的泛化能力。模型泛化能力不高的两种常见原因包括欠拟合和过拟合。
- **欠拟合是指模型学习能力较弱，无法学习到样本数据隐含的复杂模式**，因而导致模型泛化能力较弱。**欠拟合的一种常见原因是特征数量太少而引起，导致区分标准太粗糙**，例如图像识别中仅仅依靠颜色来区分猫狗。欠拟合的特点是在训练集和测试集上的性能都较差。
- 相反，**过拟合是指模型学习能力过强，以至于从个别样本中学习到了训练集中不属于“一般规律”的噪声，从而导致模型泛化能力下降**。过拟合通常是由于特征参数太多，模型复杂度过高导致。例如猫狗图像分类中如果训练集中个别猫身上有铃铛，则机器学习后认为猫都有铃铛，而没有铃铛的就不是猫。过拟合的特点是在训练集上性能表现非常好，但在测试集上性能表现较差。

梯度消失和梯度爆炸问题

常见的克服欠拟合的方法

1. **增加新特征**。目的是增大假设空间，使区分标准细化；
2. **添加多项式特征**。例如在线性模型中加入二次项、三次项或者更高次项以使模型泛化能力更强；
3. **减小参数的正则化强度**。如果模型发生了欠拟合，则需要减小参数的正则化强度。因为参数正则化的主要目的是用来防止过拟合；
4. **使用非线性模型**。例如引入非线性核函数的支持向量机模型、各种深度学习模型等；
5. **使用集成学习方法**。即将多个弱学习模型组合提升为一个强的学习模型。

梯度消失和梯度爆炸问题

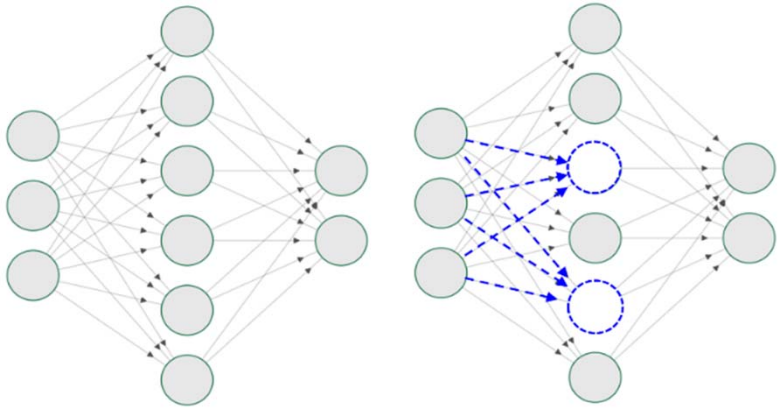
常见的克服过拟合的方法

1. **对模型的复杂度进行约束：**
   - 减少解释变量（特征）的个数。包括人工删选并只保留重要的特征、使用特征选择算法删选特征等；
   - 特征降维。即将原始特征空间进行某种变换或映射，如采用主成分分析方法等；
2. **对模型的参数进行正则化处理。**正则化处理方法仍然保留所有特征，但是通过减少参数的取值来降低模型的复杂度。例如采用常见的L1正则化或L2正则化等；
3. **数据增强（Data augmentation）。**“有时候不是因为算法好赢了，而是因为拥有更多的数据才赢了”。过拟合的一个主要原因是训练样本数据太少，因此可以增加更多的训练样本。数据增强的常见方法有：
   - 从数据来源采集获取更多的样本数据；
   - 对原有样本做某种变换、截取 [Sun, 2014]或加上随机噪声畸变Patrice，例如图像分类中对原始图像做旋转平移等处理并不改变样本的标签。
4. **早停法（Early Stopping）。**早停法是一种使迭代次数截断以防止过拟合的方法。即在训练迭代过程中记录模型在验证集上的表现，当模型在验证集上的性能比上一次迭代的结果差时停止训练；

梯度消失和梯度爆炸问题

常见的克服过拟合的方法

5. **Dropout法**。Dropout法是指在深度学习网络的训练阶段，按照一定的概率将某些神经元暂时屏蔽[Krizhevsky, 2012]，而在模型训练完毕后这些神经元都将重新投入使用。如图6.17所示，被暂时屏蔽的神经元用虚线连接，表示在训练过程中不会被更新。且每次迭代中再次随机选择部分神经元被屏蔽处理。通过这种机制，可以达到类似集成学习的效果；



6. **交叉验证**。其通过对训练集做进一步划分，具体可为5折或者10折交叉验证，并选择在验证集上表现较好的模型；
7. **使用集成学习方法**。即将多个弱学习模型组合提升为一个强的学习模型。例如Boosting、Bagging、随机森林（Random Forest）等技术。

本章小结

- 信息论中的熵；
- 模型优化中的目标函数；
- 用梯度下降法求解目标函数的极值；
- 梯度下降法中学习率的调节方法；
- 常见的激活函数；
- 梯度消失和梯度爆炸问题及应对策略；
- 欠拟合和过拟合问题及其应对策略。