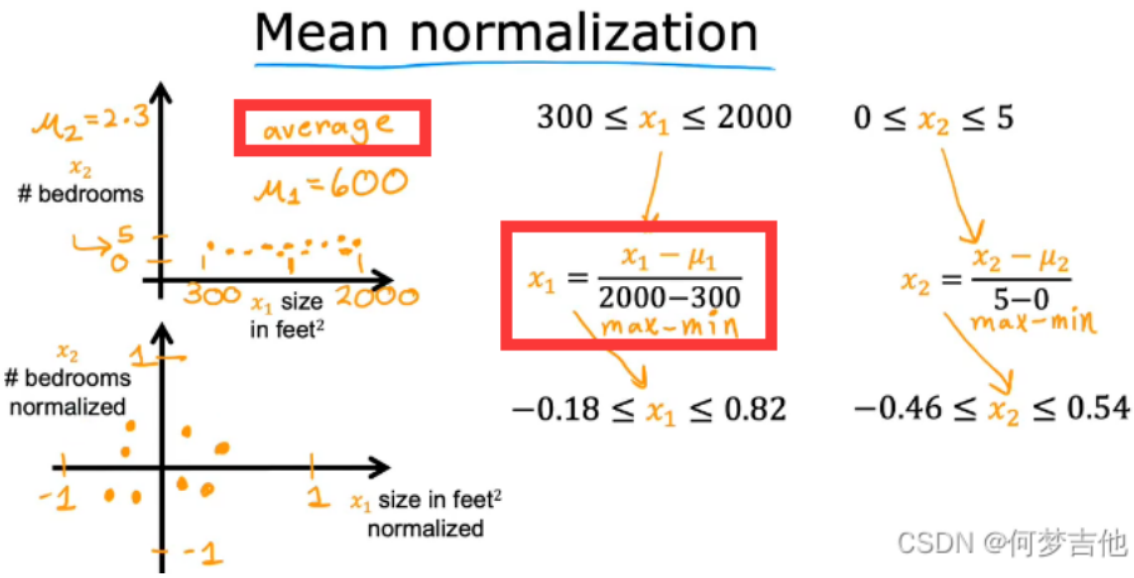
特征归一化

用于数据预处理，使数据处在合理范围内，能使偏离值较为均匀的变化，使得梯度下降更便于找到最好值



逻辑回归

sigmoid 函数

g 代表一个常用的逻辑函数（logistic function）为S形函数（Sigmoid function），公式为：

$$g(z)=\frac{1}{1+e^{-z}}$$

合起来，我们得到逻辑回归模型的假设函数：

$$h_\theta(x)=\frac{1}{1+e^{-\theta^T X}}$$

代价函数

逻辑回归的损失函数Loss 即期望若落空造成的损失

Logistic loss function

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right),y^{(i)}\right)=\begin{cases}-\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)}=1\\ -\log\left(1-f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)}=0\end{cases}$$

简化代价函数

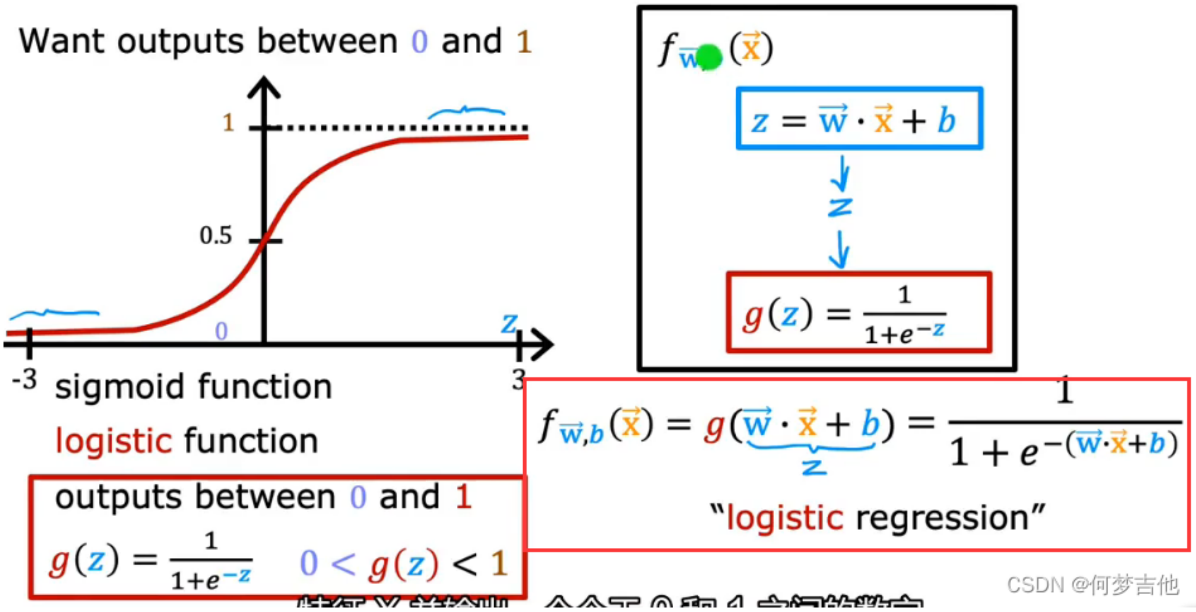
$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right),y^{(i)}\right)=\begin{cases}-\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)}=1\\ -\log\left(1-f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) & \text{if } y^{(i)}=0\end{cases}$$

$$\boxed{L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right),y^{(i)}\right)=-y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)-\left(1-y^{(i)}\right)\log\left(1-f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)}$$

$$J(\vec{w},b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)})\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)\right]$$

这使得Loss也具有了碗的形状，使得逻辑回归也能使用梯度下降算法寻找极值

逻辑回归模型



简化代价函数与梯度下降

Simplified loss function

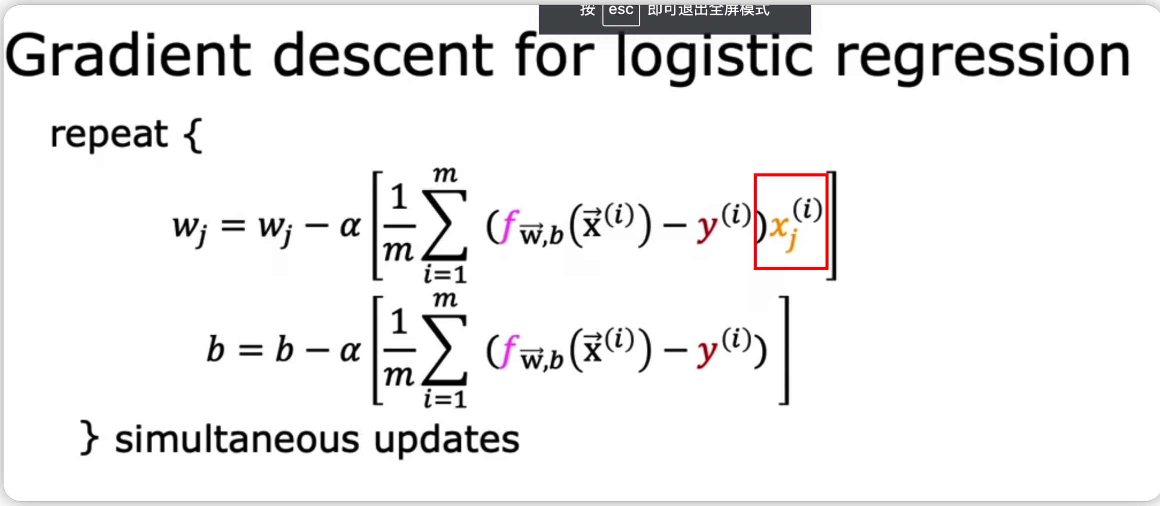
$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right) & \text{if } y^{(i)} = 0 \end{cases}$$

$$\boxed{L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = -y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) - (1-y^{(i)})\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)}$$

进而得到：

$$J(\vec{w},b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)})\log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)\right]$$

梯度下降：



最后发现与正常的线性拟合公式相同

过拟合

当变量过多时，训练出来的假设能很好地拟合训练集，所以代价函数实际上可能非常接近于0，但得到的曲线为了千方百计的拟合数据集，导致它无法泛化到新的样本中，无法预测新样本数据（冗余项）

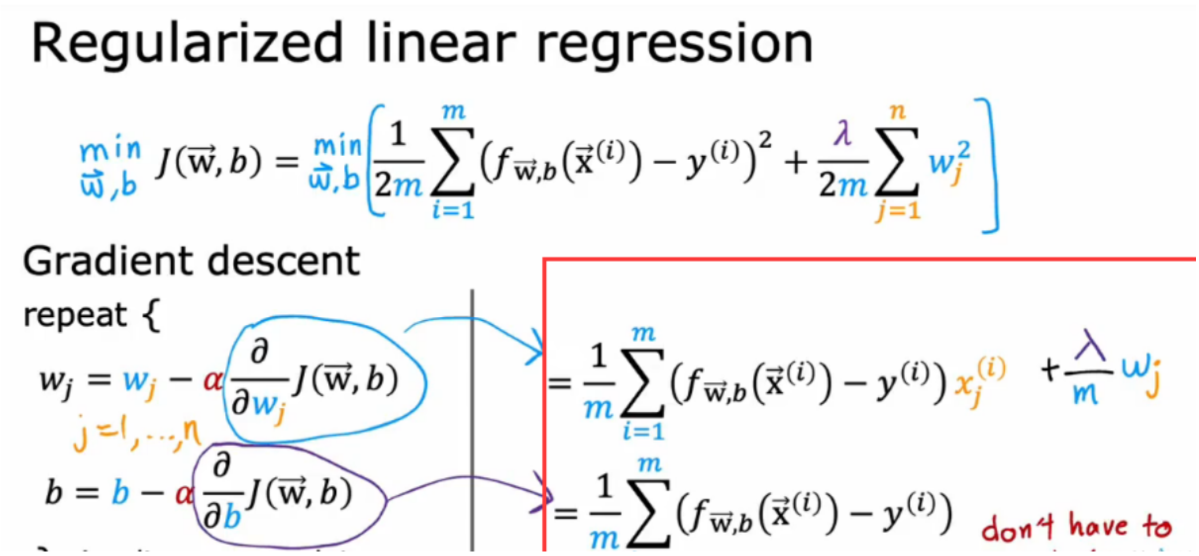
正则化

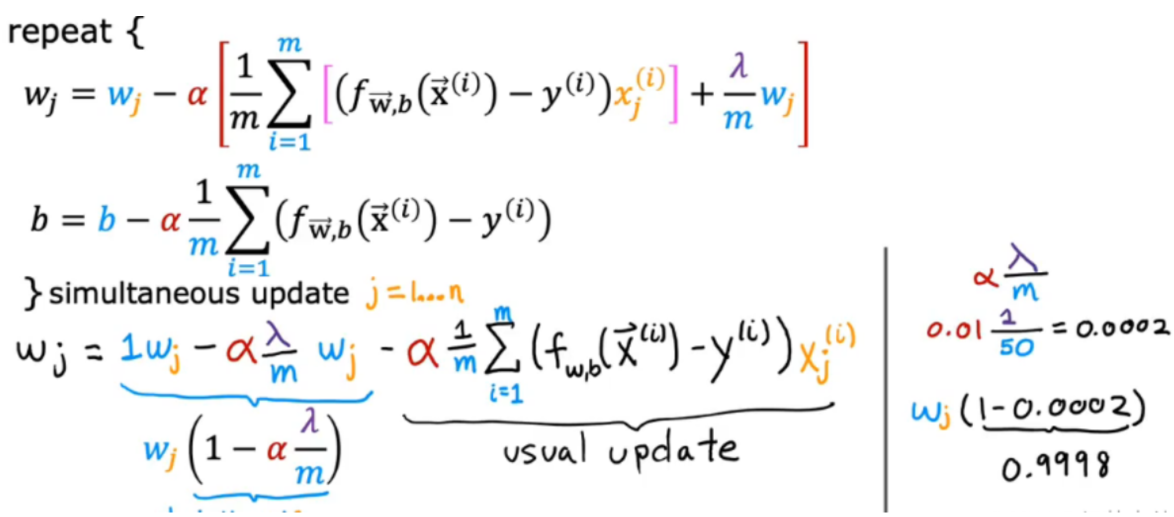
正则化项：

$$\lambda \sum_{j=1}^{n} \theta_j^2$$

如果有很多参数，我们**不清楚哪个参数是高阶项**，即**不知道惩罚哪个能获得更好拟合的结果**，因此**引入正则化项统一惩罚参数**以得到较为简单的函数

正则化后：





得到最终公式

神经网络

输入前一层的激活，给定当前层的参数，它会输出下一层激活值 自定义sequential函数
中间层为网络层、隐藏层，最后输出为激活层

前向传播