



深度学习框架Tensorflow学习与应用 第4课

二次代价函数(quadratic cost)

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2$$

- 其中，C表示代价函数，x表示样本，y表示实际值，a表示输出值，n表示样本的总数。为简单起见，同样一个样本为例进行说明，此时二次代价函数为：

- $a = \sigma(z)$, $z = \sum W_j * X_j + b$

- $\sigma()$ 是激活函数

$$C = \frac{(y - a)^2}{2}$$

二次代价函数(quadratic cost)

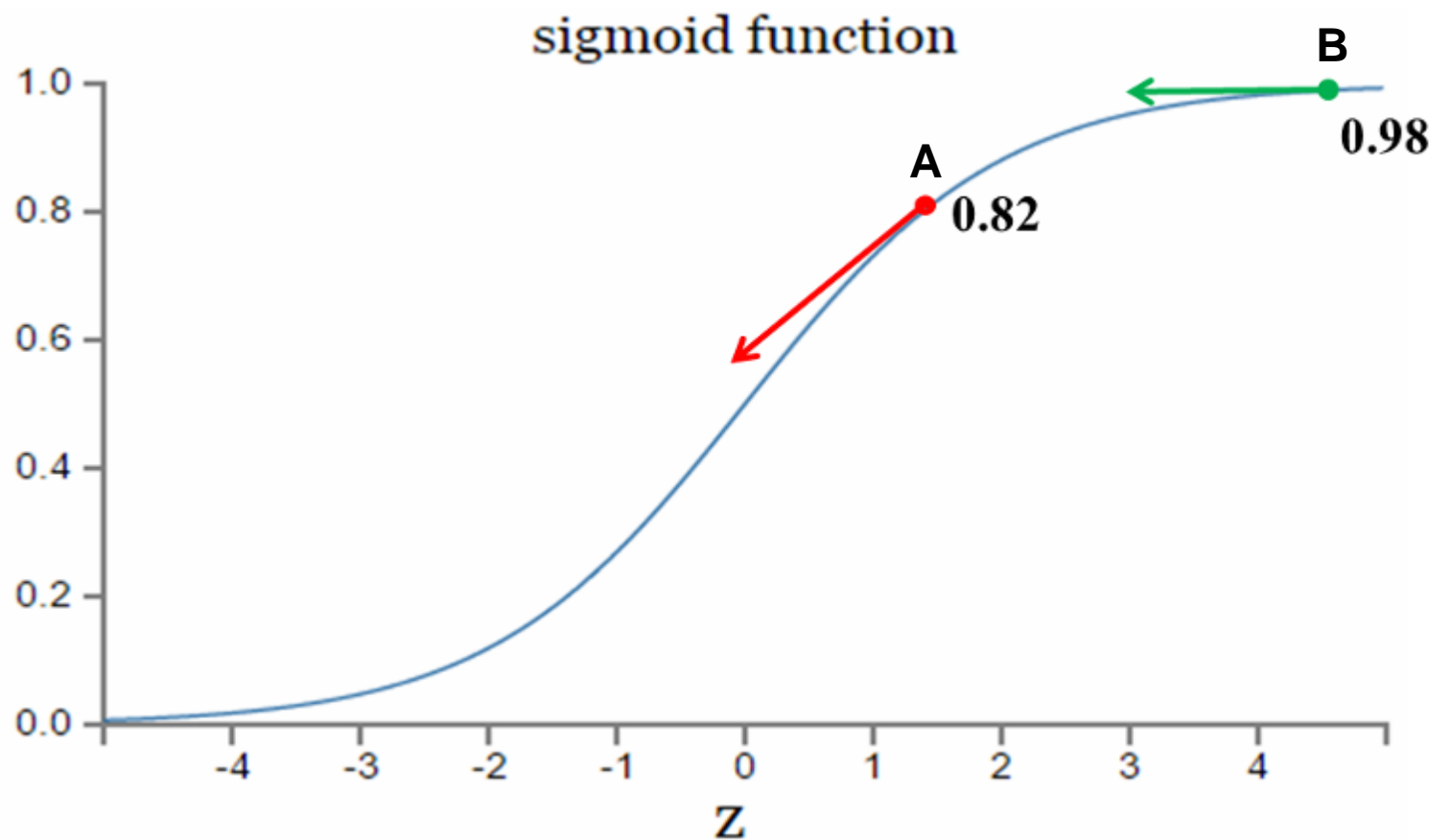
- 假如我们使用梯度下降法(Gradient descent)来调整权值参数的大小，权值w和偏置b的梯度推导如下：

$$\frac{\partial C}{\partial w} = (a - y)\sigma'(z)x$$
$$\frac{\partial C}{\partial b} = (a - y)\sigma'(z)$$

- 其中，z表示神经元的输入， σ 表示激活函数。w和b的梯度跟激活函数的梯度成正比，激活函数的梯度越大，w和b的大小调整得越快，训练收敛得就越快。

二次代价函数(quadratic cost)

- 假设我们的激活函数是sigmoid函数：



交叉熵代价函数(cross-entropy)：

- 换一个思路，我们不改变激活函数，而是改变代价函数，改用交叉熵代价函数：

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)]$$

- 其中，C表示代价函数，x表示样本，y表示实际值，a表示输出值，n表示样本的总数。

$$a = \sigma(z), \quad z = \sum W_j * X_j + b$$

$$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$

$$\begin{aligned} \frac{\partial C}{\partial w_j} &= -\frac{1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{(1-y)}{1-\sigma(z)} \right) \frac{\partial \sigma}{\partial w_j} \\ &= -\frac{1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{(1-y)}{1-\sigma(z)} \right) \sigma'(z) x_j \\ &= \frac{1}{n} \sum_x \frac{\sigma'(z) x_j}{\sigma(z)(1-\sigma(z))} (\sigma(z) - y) \\ &= \frac{1}{n} \sum_x x_j (\sigma(z) - y) \end{aligned}$$

$$\frac{\partial C}{\partial b} = \frac{1}{n} \sum_x (\sigma(z) - y)$$

交叉熵代价函数(cross-entropy) :

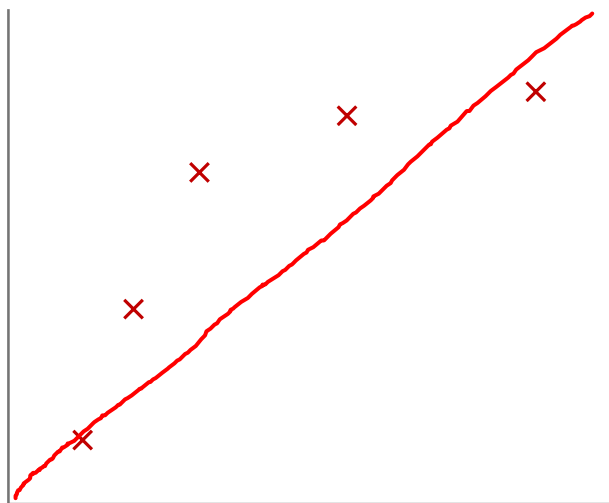
$$\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum_x x_j (\sigma(z) - y)$$

$$\frac{\partial C}{\partial b} = \frac{1}{n} \sum_x (\sigma(z) - y)$$

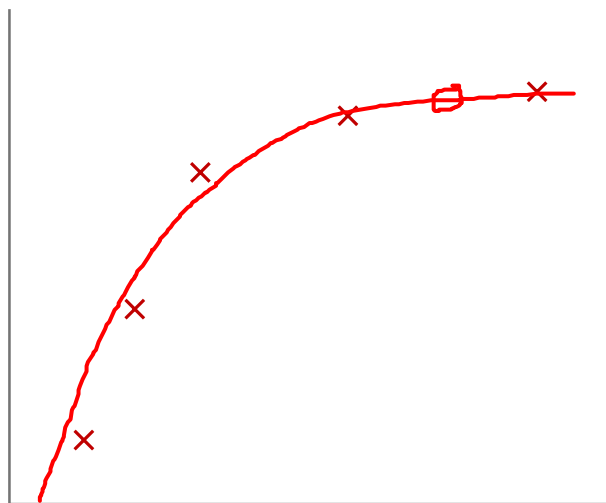
- 权值和偏置值的调整与 $\sigma'(z)$ 无关，另外，梯度公式中的 $\sigma(z) - y$ 表示输出值与实际值的误差。所以当误差越大时，梯度就越大，参数w和b的调整就越快，训练的速度也就越快。
- 如果输出神经元是线性的，那么二次代价函数就是一种合适的选择。如果输出神经元是S型函数，那么比较适合用交叉熵代价函数。

对数释然代价函数(log-likelihood cost) :

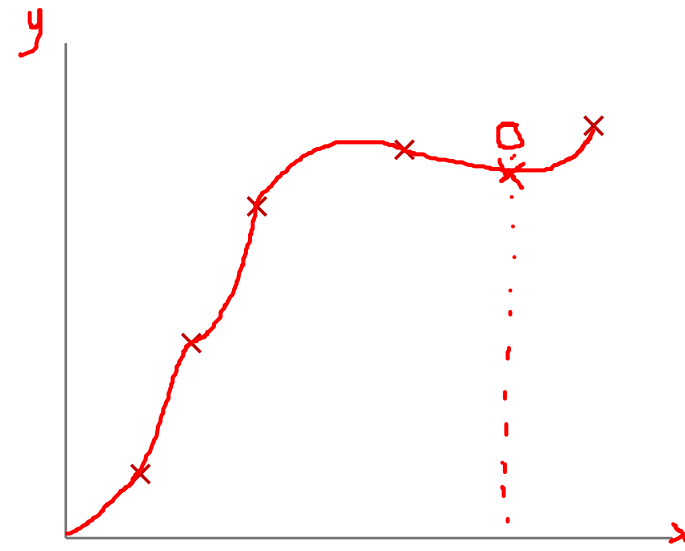
- 对数释然函数常用来作为softmax回归的代价函数，如果输出层神经元是sigmoid函数，可以采用交叉熵代价函数。而深度学习中更普遍的做法是将softmax作为最后一层，此时常用的代价函数是对数释然代价函数。
- 对数似然代价函数与softmax的组合和交叉熵与sigmoid函数的组合非常相似。对数释然代价函数在二分类时可以化简为交叉熵代价函数的形式。
- 在Tensorflow中用：
tf.nn.sigmoid_cross_entropy_with_logits()来表示跟sigmoid搭配使用的交叉熵。
tf.nn.softmax_cross_entropy_with_logits()来表示跟softmax搭配使用的交叉熵。



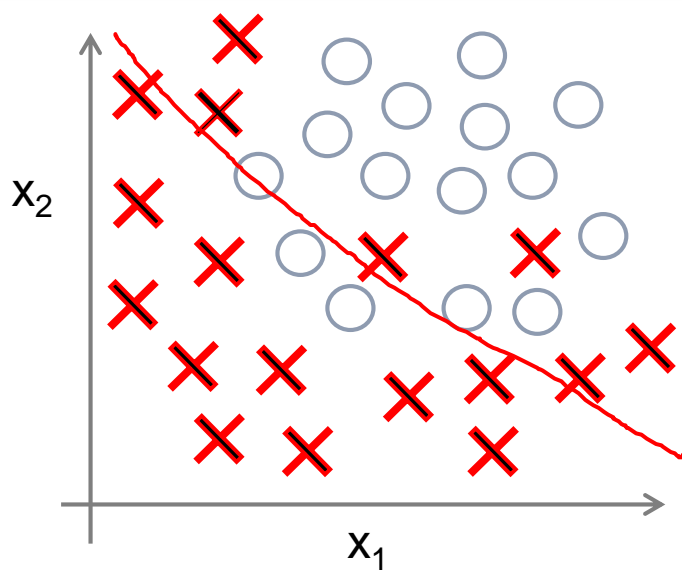
欠拟合(Underfitting)



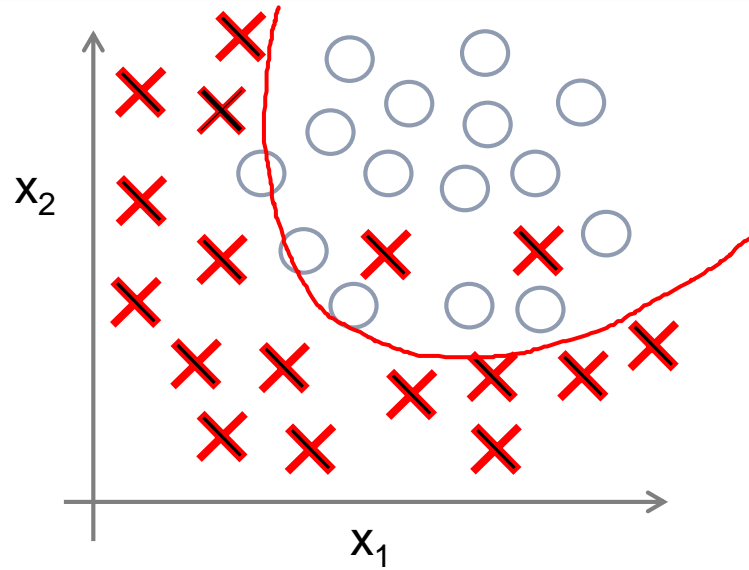
正确拟合(Just right)



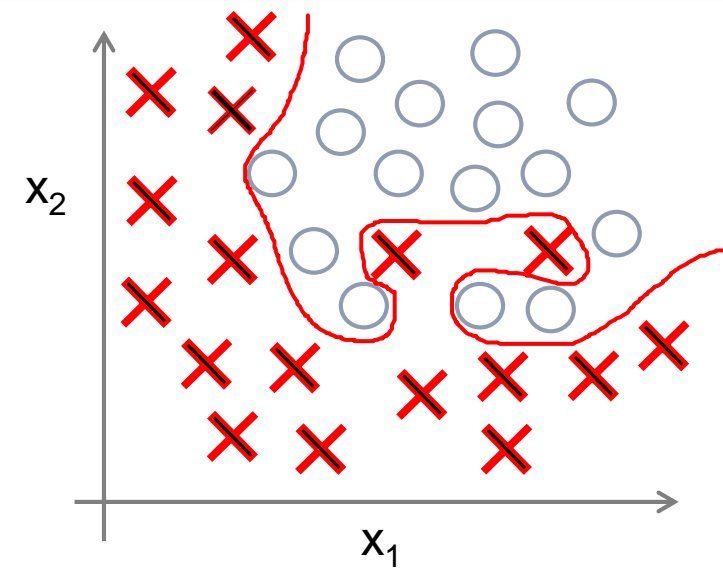
过拟合(Overfitting)



欠拟合(Underfitting)



正确拟合(Just right)



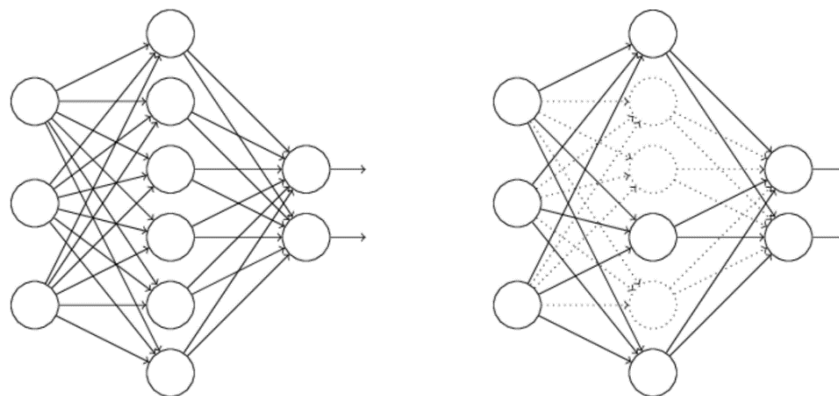
过拟合(Overfitting)

- 增加数据集

- 正则化方法

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2$$

- Dropout



【声明】 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>

- Dataguru (炼数成金) 是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 <http://edu.dataguru.cn>

Thanks

FAQ时间