

Softmax

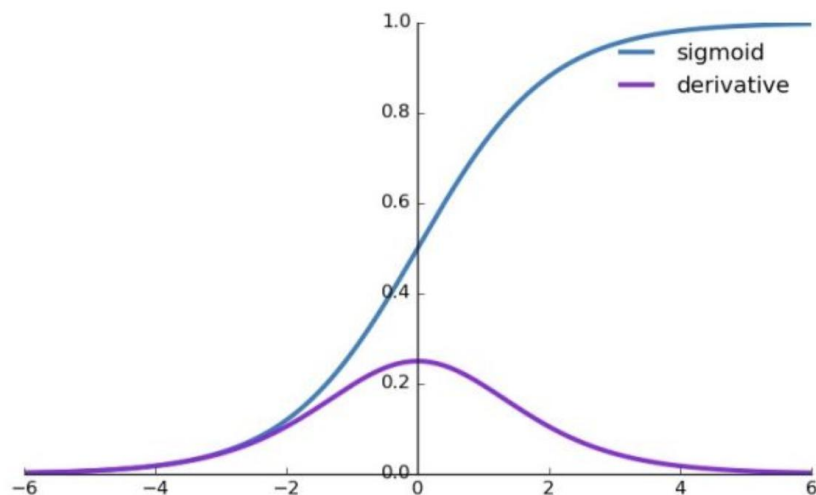
交叉熵损失函数相对于 mse 损失函数的优点

- 1) 收敛速度交叉熵损失函数更快
- 2) mse 求出来的概率值可能在 0-1 之外

缺点

使用MSE的一个缺点就是其偏导值在输出概率值接近0或者接近1的时候非常小，这可能会造成模型刚开始训练时，偏导值几乎消失。

假设我们的MSE损失函数为： $J = \frac{1}{2}(y_i - \hat{y}_i)^2$ ，偏导为： $\frac{dJ}{dW} = (y_i - \hat{y}_i)\sigma'(Wx_i + b)x_i$ ，其中 $\sigma'(Wx_i + b)$ 为 $\sigma(Wx_i + b)(1 - \sigma(Wx_i + b))$ 。可以看出，在 $\sigma(Wx_i + b)$ 值接近0或者1的时候， $\frac{dJ}{dW}$ 的值都会接近于0，其函数图像如下：



这导致模型在一开始学习的时候速率非常慢，而使用交叉熵作为损失函数则不会导致这样的情况发生。

$$\frac{\partial p_j}{\partial y_i} = \frac{\partial \left(\frac{e^{y_j}}{\sum_{j=1}^K e^{y_j}} \right)}{\partial y_i} = \frac{(e^{y_j})' \sum_{j=1}^K e^{y_j} - e^{y_j} \left(\sum_{j=1}^K e^{y_j} \right)'}{\left(\sum_{j=1}^K e^{y_j} \right)^2} = \frac{0 \sum_{j=1}^K e^{y_j} - e^{y_j} e^{y_i}}{\left(\sum_{k=1}^K e^{y_k} \right)^2} = \frac{-e^{y_j} e^{y_i}}{\left(\sum_{j=1}^K e^{y_j} \right)^2} = -S(y_j) S(y_i)$$

- 再看 $\frac{\partial y_i}{\partial w_n}$

$$\frac{\partial y_i}{\partial w_n} = x_n$$

接下来我们只需要把上面的组合起来：

$$\begin{aligned} &= -p(i)(1-p(i)) \sum_{i=1, i=j}^K \frac{y_i}{p_i} - p(i)p(j) \sum_{i=1, i \neq j}^K \frac{y_i}{p_j} \\ \frac{\partial J}{\partial p_j} \cdot \frac{\partial p_j}{\partial y_i} &= -(1-p(i)) \sum_{i=1, i=j}^K y_i - p(i) \sum_{i=1, i \neq j}^K y_i \\ &= - \sum_{i=1, i=j}^K y_i + p(i) \sum_{i=1}^K y_i \end{aligned}$$

最后针对分类问题，给定的 y_i 中只会有一类别是1，其他类别都是0，所以

$$\frac{\partial J}{\partial p_j} \cdot \frac{\partial p_j}{\partial y_i} \cdot \frac{\partial J}{\partial w_n} = \frac{\partial y_i}{\partial w_n} = (p(i) - 1)w_n$$

注意看， $p(i) - 1$ 是啥？是不是**SoftMax层的输出的概率-1**，梯度就是这么容易计算！！！太神奇了？！

就是为什么神经网络分类器要用交叉熵损失函数的原因！

softmax 思想

就是一个向量乘以 w 矩阵，目标为了将向量维度变为和目标类别同样维度，然后计算 $\exp(\text{pi}^{\wedge})$ 即可

交叉熵损失函数

前面提到，在多分类问题中，我们经常使用交叉熵作为损失函数

$$Loss = - \sum_i t_i \ln y_i$$

其中， t_i 表示真实值， y_i 表示求出的softmax值。当预测第 i 个时，可以认为 $t_i = 1$ 。此时损失函数变成了：

$$Loss_i = -\ln y_i$$