

1. 概念

1.1 损失函数的作用

1.2 常用损失函数

1. 概念

“损失”就是所有样本的“误差”的总和，亦即（ m 为样本数）：

$$\text{损失} = \sum_{i=1}^m \text{误差}_i$$

$$J = \sum_{i=1}^m \text{loss}_i$$

在黑盒子的例子中，如果我们把神经网络的参数调整到完全满足独立样本的输出误差为 0，通常会令其它样本的误差变得更大，这样作为误差之和的损失函数值，就会变得更大。所以，我们通常会在根据某个样本的误差调整权重后，计算一下整体样本的损失函数值，来判定网络是不是已经训练到了可接受的状态。

1.1 损失函数的作用

损失函数的作用，就是计算神经网络每次迭代的前向计算结果与真实值的差距，从而指导下一步的训练向正确的方向进行。

步骤：

1. 用随机值初始化前向计算公式的参数；
2. 代入样本，计算输出的预测值；
3. 用损失函数计算预测值和标签值（真实值）的误差；
4. 根据损失函数的导数，沿梯度最小方向将误差回传，修正前向计算公式中的各个权重值；
5. 进入第2步重复，直到损失函数值达到一个满意的值就停止迭代。

1.2 常用损失函数

符号规则： a 是预测值， y 是样本标签值， loss 是损失函数值。

- Gold Standard Loss，又称0-1误差

$$\text{loss} = \begin{cases} 0 & a = y \\ 1 & a \neq y \end{cases}$$

- 绝对值损失函数

$$\text{loss} = |y - a|$$

- Hinge Loss，铰链/折页损失函数或最大边界损失函数，主要用于SVM（支持向量机）中

$$\text{loss} = \max(0, 1 - y \cdot a) \quad y = \pm 1$$

- Log Loss，对数损失函数，又叫交叉熵损失函数(cross entropy error)（二分类）

$$\text{loss} = -[y \cdot \ln(a) + (1 - y) \cdot \ln(1 - a)] \quad y \in 0, 1$$

交叉熵函数常用于逻辑回归(logistic regression)，也就是分类(classification)。

交叉熵可在神经网络中作为损失函数， p 表示真实标记的分布， q 则为训练后的模型的预测标记分布，交叉熵损失函数可以衡量 p 与 q 的相似性。

为什么不能使用均方差做为分类问题的损失函数？

1. 回归问题通常用均方差损失函数，可以保证损失函数是个凸函数，即可以得到最优解。而分类问题如果用均方差的话，损失函数的表现不是凸函数，就很难得到最优解。而交叉熵函数可以保证区间内单调。
 2. 分类问题的最后一层网络，需要分类函数，Sigmoid或者Softmax，如果再接均方差函数的话，其求导结果复杂，运算量比较大。用交叉熵函数的话，可以得到比较简单的计算结果，一个简单的减法就可以得到反向误差
- Squared Loss, 均方差损失函数

$$loss = (a - y)^2$$

，计算预测值和真实值之间的欧式距离。预测值和真实值越接近，两者的均方差就越小。

均方差函数常用于线性回归(linear regression)，即函数拟合(function fitting)。公式如下：

$$loss = \frac{1}{2}(z - y)^2 \quad (\text{单样本})$$

$$J = \frac{1}{2m} \sum_{i=1}^m (z_i - y_i)^2 \quad (\text{多样本})$$

- Exponential Loss, 指数损失函数

$$loss = e^{-(y \cdot a)}$$

##