损失函数

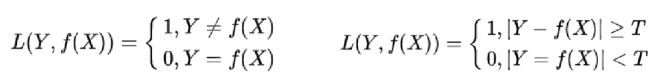
# 什么是损失函数

机器学习，准确地说监督学习的本质是给定一系列训练样本(xi, yi)，尝试学习x -> y 的映射关系，使得给定一个新的x ，即便这个x不在训练样本中，也能够使模型的输出y^，尽量与真实的y接近。

损失函数是用来估量模型的输出y^与真实值y之间的差距，给模型的优化指引方向。

# 常见的损失函数

## 0-1损失函数（Zero-One Loss)



若绝对值在T以下，损失函数为0，否则为1。它是一个非凸函数不太适用

## 对数损失函数

对数损失函数（LogLoss）是经常在离线评估中使用的指标，在一个二分类问题中，LogLoss的定义如下：

其中，为输入实例的真实类别，为预测输入实例是正样本的概率，N是样本总数。

## 交叉熵损失函数

熵(Entropy)源于信息学，用于度量信息的不确定度。熵越大，代表信息的不确定性越大，信息也就越大（一种类比的理解就是，对于十分稳定的物体来说，一直不发生变化是确定的，发生变化所蕴含的信息量就更大）。某个分布P(i)的熵定义为：

实际上，𝐻(𝑃)也可以使用其他底数的log函数计算。举个例子，对于 4 分类问题，如果某个样本的真实标签是第4类，那么标签的One-hot编码为[0,0,0,1]，即这张图片的分类是唯一确定的，它属于第4类的概率𝑃(𝑦为4|𝒙) = 1，不确定性为0，它的熵可以简单的计算为：

也就是说，对于确定的分布，熵为0，不确定性最低。如果它预测的概率分布是[0.1,0.1,0.1,0.7]，它的熵可以计算为：

这种情况比前面确定性类别的例子的确定性要稍微大点。

考虑随机分类器，它每个类别的预测概率是均等的：[0.25,0.25,0.25,0.25]，同样的方法，可以计算它的熵约为2，这种情况的不确定性略大于上面一种情况。

由于𝑃(𝑖) ∈ [0,1], log2 𝑃(𝑖) ≤ 0，因此熵𝐻(𝑃)总是大于等于0。当熵取得最小值0时， 不确定性为0。

交叉熵(Cross Entropy)的定义如下：

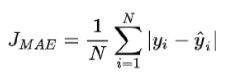
𝐻(𝑝||𝑞) ≜ −∑𝑝(𝑖)log2 𝑞(𝑖)

## 指数损失函数



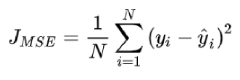
对噪声点，离群点非常敏感。

## 平均绝对误差损失（Mean Absolute Error Loss）



计算预测值和目标值差的绝对值，也称为MAE Loss、L1 Loss

## 平均平方误差损失（Mean Square Error Loss）



计算预测值和目标值差的平方，也称均方误差损失，MSE Loss，L2 Loss

机器学习、深度学习回归任务中最常用的一种损失函数

## 合页损失（Hinge Loss）

其公式如下：

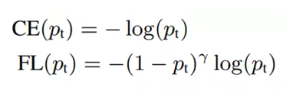


如果被分类正确，损失为0，否则损失就为



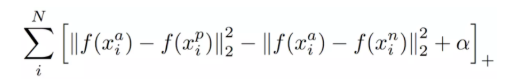
合页损失函数适用于maximum-margin的分类，支持向量机Support Vector Machine (SVM)模型的损失函数，本质上就是Hinge Loss + L2正则化。

## Focal Loss



以交叉熵为基础，使模型更关注较难的样本。

## Triplet Loss



a: anchor，p: positive，n: negative

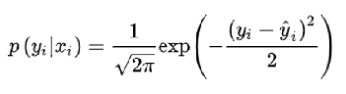
整体优化目标是拉近a, p的距离，拉远a, n的距离，达成分类效果

# 讨论

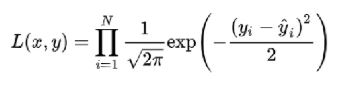
## 为什么回归任务常用均方误差损失

我们可以发现，在深度学习的回归任务中，常常会用到均方误差损失函数，这是为什么呢？

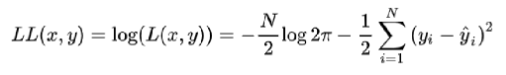
一般在生活中，如果没有系统误差，那么我们对一个数值进行估计或测量，很多情况下估计值和真值之间的误差是服从高斯分布的。我们一个模型对于目标标签的估计也可以这么认为。我们可以假设高斯分布的均值为0，方差为1。第i个样本的输入为xi , 模型的输出为yi^, 而它的标签为yi , 那么我们估计的yi的条件概率密度函数如下图所示。



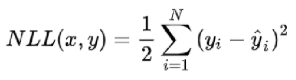
对于所有样本来说，我们可以计算出它的似然函数，我们对模型的训练就是要极大化这个似然函数。



常用的做法是通过log来将比较难处理的连乘形式似然函数转换成连加形式的对数似然函数，其单调性不变。



可以看到对数似然函数是负号的形式，而且第一项是一个常数，所以我们通过进一步的改写将最大化以上的对数似然函数变化为最小化以下的负对数似然函数。

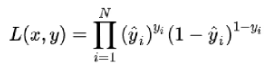


大家应该注意到了，这不就跟均方损失函数的形式几乎一样吗？正是如此，最大化似然函数和最小化均方损失函数是等价的。所以，在回归任务中，估计值和真值误差服从高斯分布的假设下，我们以均方误差作为损失函数来训练模型是合理的。

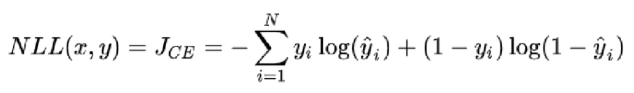
## 为什么分类任务用交叉熵作为损失函数

交叉熵损失函数是我们在分类任务中经常用到的损失函数，我们可以来做类似的分析。我们将要求解的问题简化为二分类，即我们使用一个模型来预测某个样本是属于分类0还是分类1，一般做法我们会去预测一个在0到1内的值yi^=p，如果大于0.5那么就是分类1，小于0.5就是分类0。

那么，在其中我们常常已经做了一个假设，即我们期望模型的输出服从一个伯努利分布。模型的输出就是预测分类为1的概率，即 P(y=1) = p, P(y=0) = 1 - p。我们将左边这个式子改写一下，在输入为x时，P(y|x) = py \* ( 1 - p)(1-y)。对于所有的样本来说，将p替换成模型的预测yi^，那么其似然函数如下：



我们优化模型的过程就是极大化我们的似然函数，让模型的预测值出现的可能性最大。刚刚我们已经介绍过了，我们通过常规操作，可以将最大化似然函数的目标转化为最小化负对数似然函数。



我们又能发现，这显然就是我们的交叉熵损失函数。所以，本质上对分类任务的极大似然估计和最小化交叉熵损失函数是一致的。所以，只要服从伯努利分布的假设，我们使用交叉熵处理分类任务就是很合理的。

参考：

龙书，TensorFlow深度学习。