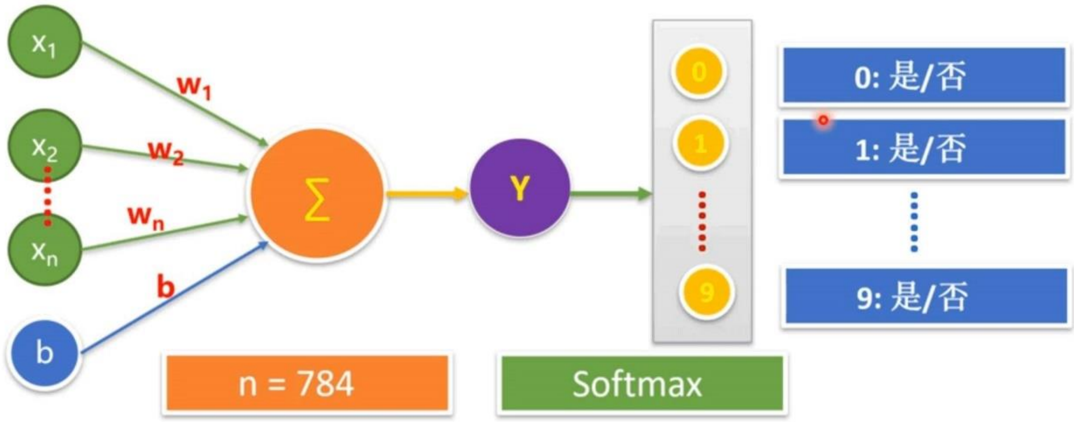
逻辑回归



逻辑回归（Logistic Regression），简称LR。它的特点是能够是我们的特征输入集合转化为0和1这两类的概率。一般来说，回归不用在分类问题上，因为回归是连续型模型，而且受噪声影响比较大。如果非要应用进入，可以使用逻辑回归。了解过线性回归之后再来看逻辑回归可以更好的理解。

优点：计算代价不高，易于理解和实现缺点：容易欠拟合，分类精度不高

适用数据：数值型和标称型

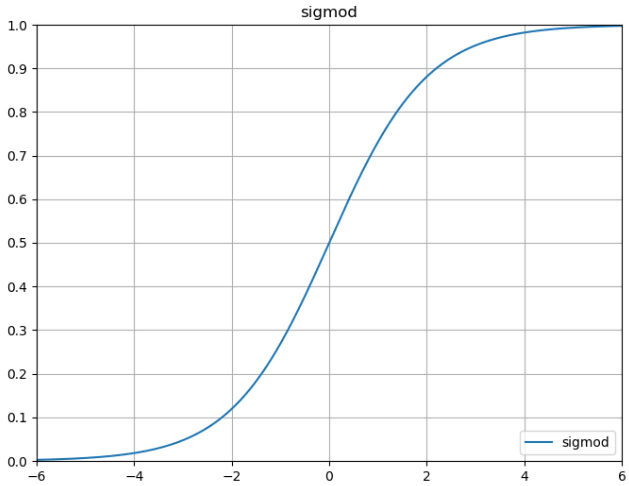
对于回归问题后面会介绍，Logistic回归本质上是线性回归，只是在特征到结果的映射中加入了一层函数映射，即先把特征线性求和，然后使用函数g(z)将最为假设函数来预测。g(z)可以将连续值映射到0和1上。Logistic回归用来分类0/1问题，也就是预测结果属于0或者1的二值分类问题映射函数为：

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

其中

$$z=\theta0+\theta1x1+\theta2x_2+\ldots$$

映射出来的效果如下如：



示例代码:

01-log.py,02-mlog.py

import sklearn.linear_model as lm

逻辑回归分类器 = lm.LogisticRegression(solver='lbfgs')

优化算法选择参数:

solver:

a)liblinear,坐标轴下降法迭代优化损失函数，是开源的 liblinear 库来实现

b)lbfgs，拟牛顿法，利用损失函数二阶导数矩阵，海森矩阵来迭代优化损失函数。

c)newton-cg,也是牛顿法家族中一种，利用损失函数二阶导数矩阵，海森矩阵来迭代优化损失函数

d)sag,随机平均梯度下降，是梯度下降的变种，区别在于仅仅用一部分的样本计算梯度，适合于样本数量

较多的时候。

结论：如果不是大样本，可以选择 liblinear，但 libnear 只支持 OvR，不支持 MvM,即多元逻辑回归时，

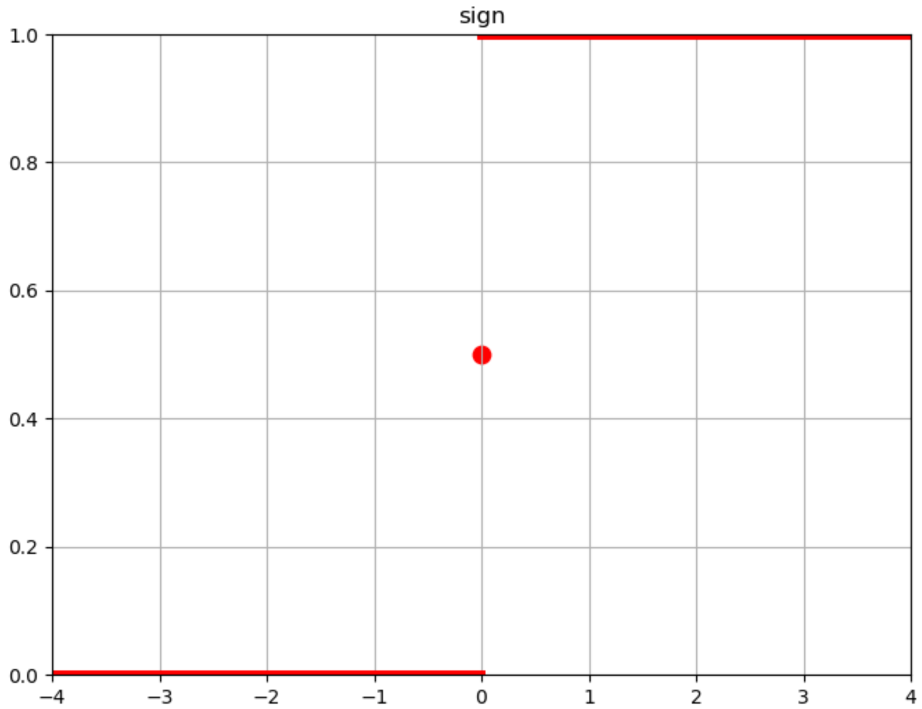
不能选择 liblinear。所以一般采用 lbfgs。样本少的时候不要选择 sag。

1、映射函数

1）sign 函数：跃迁函数，不是连续的

```
          / 0    if z<0
f(z) = | 0.5 if z=0
          \ 1    if z>0
```

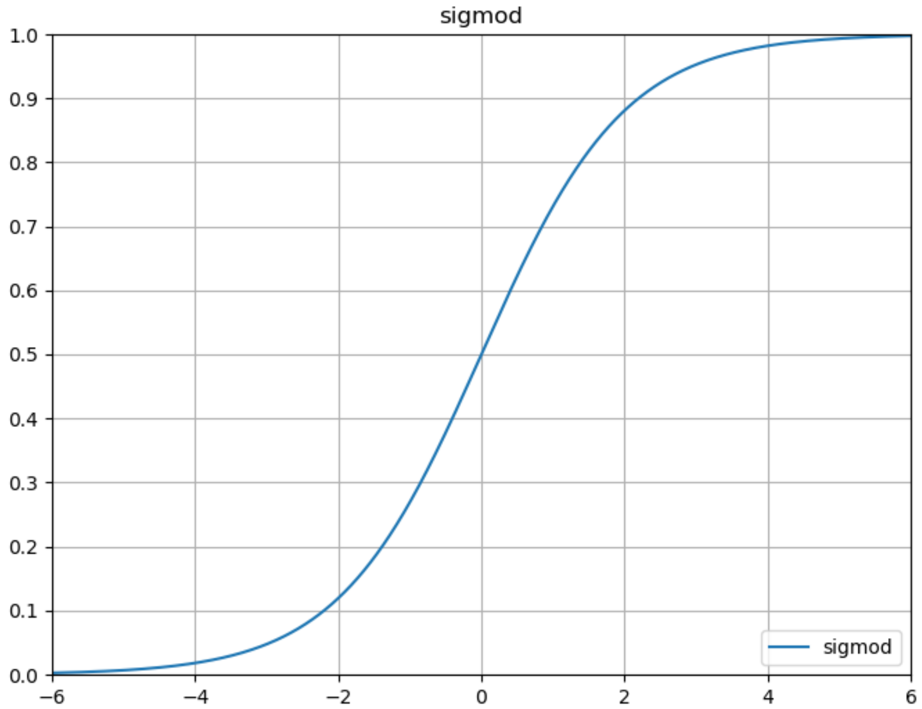
z = w0*x0 + w1*x1 +...+wn*xn



2）sigmoid 函数:　　　　（sign 函数的高阶版，把结果映射到 0,1 之间，但是连续可导的）

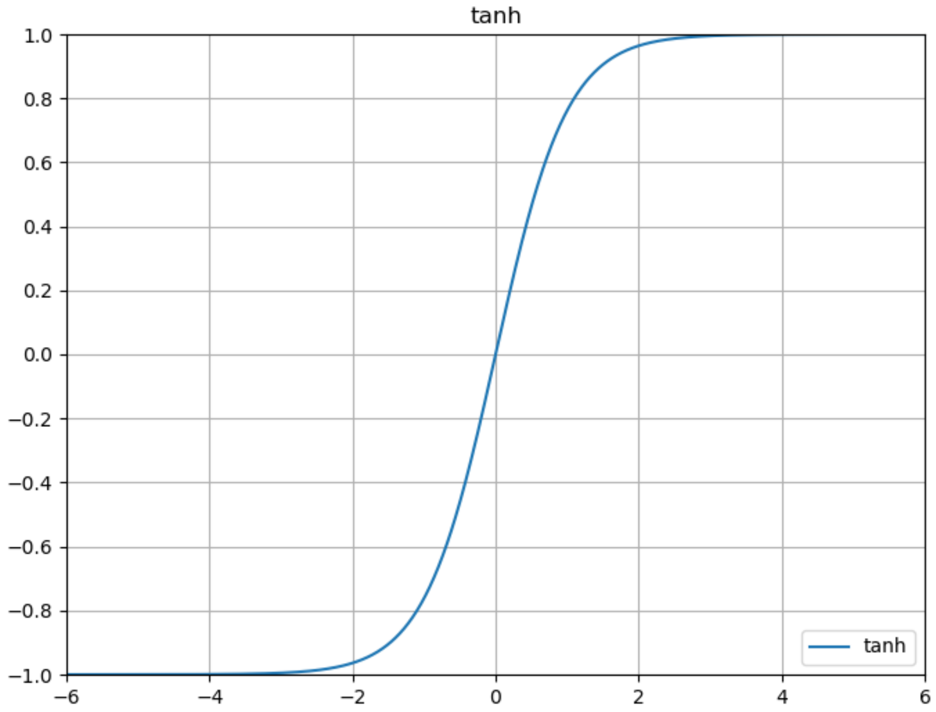
$$\text{sigmoid}(Z) = \frac{1}{1+e^{-z}}$$

z = w0*x0 + w1*x1 +...+wn*xn



3）Tahn 函数 （sign 函数的高阶版，把结果映射到-1,1 之间，但是连续可导的）

$$\mathrm{Tanh}(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$$



2、凸函数

凸函数：sigmoid 函数（逻辑回归），最小二乘法（回归），softmax（神经网络）。
交叉熵误差函数，均方误差函数。

1、softmax

e**Vi

S =　　-----------------------

e**V1+e**V2+...+ e**Vn

直白来说就是通过 softmax 可以将原来的输出映射为（0,1）的值。而这些值的累和为 1（满足概率的性质），可以

把其理解为概率，在最后选取输出结点的时候，我们可以选取概率最大（值对应最大）结点。

一般用在神经网络中。

v1　3----> e**v1 e^3:20 -------->e**vi/e**V1+e**V2+...+ e**Vn : 20/(20+2.7+0.05)=0.88

v2　1----> e**v2 e^1:2.7 ------->e**vi/e**V1+e**V2+...+ e**Vn : 2.7/(20+2.7+0.05)=0.12

v1　-3----> e**v3 e^-3:0.05 ---->e**vi/e**V1+e**V2+...+ e**Vn : 0.05/(20+2.7+0.05)=0

2、交叉熵损失函数

交叉熵损失函数 L = -[y1*logp1 +y2*logp2 +...+ynlogpn]

模型 1 输出结果：正确率 1/3

	模型输出 y'	真实结果 y	预测是否正确
样本 1	0.3 0.3 0.4	0 0 1（男）	正确
样本 2	0.3 0.4 0.3	0 1 0（女）	正确
样本 3	0.1 0.2 0.7	1 0 0（中）	错误

模型 2 输出结果：正确率 1/3

	模型输出 y'	真实结果 y	预测是否正确
样本 1	0.1 0.2 0.7	0 0 1（男）	正确
样本 2	0.1 0.7 0.2	0 1 0（女）	正确
样本 3	0.3 0.4 0.3	1 0 0（中）	错误

1）通过均方误差（MSE）来评价两个模型

模型 1：

样本 1 误差=（0.3-0)^2+(0.3-0)^2+(0.4-1)^2 = 0.18

样本 2 误差=（0.3-0)^2+(0.4-1)^2+(0.3-0)^2 = 0.18

样本 3 误差=（0.1-1)^2+(0.2-0)^2+(0.7-0)^2 = 0.44

样本的平均误差为：（0.18+0.18+0.44）/3 = 0.26

模型 2：

样本 1 误差=（0.1-0)^2+(0.2-0)^2+(0.7-1)^2 = 0.046

样本 2 误差=（0.1-0)^2+(0.7-1)^2+(0.2-0)^2 = 0.046

样本 3 误差=（0.3-1)^2+(0.4-0)^2+(0.3-0)^2 = 0.240

样本的平均误差为：（0.046+0.046+0.240）/3 = 0.11

结论：通过均方误差评价出模型 2 更优。但是为什么不采用均方误差函数呢？

因为逻辑回归配合均方误差函数，当使用梯度下降时，会出现训练时学习速率非常慢的情况。

所以均方误差作为分类问题的损失函数不是最佳的。我们应该选择用交叉熵函数。

2)通过交叉熵损失函数来评价两个模型

交叉熵损失函数 L = -[y1*logp1 +y2*logp2 +...+ynlogpn]

其中：y---表示样本的标签 label

p---表示样本预测为正的概率

模型 1：

样本 1 误差=（log0.3*0)+(log0.3*0)+(log0.4*1) = 0.91

样本 1 误差=（log0.3*0)+(log0.4*1)+(log0.3*0) = 0.91

样本 1 误差=（log0.1*1)+(log0.2*0)+(log0.7*0) = 2.30

样本的平均误差为：（0.91+0.91+0.2.30）/3 = 1.37

模型 2：

样本 1 误差=（log0.1*0)+(log0.2*0)+(log0.7*1) = 0.35

样本 1 误差=（log0.1*0)+(log0.7*1)+(log0.2*0) = 0.35

样本 1 误差=（log0.3*1)+(log0.4*0)+(log0.3*0) = 1.20

样本的平均误差为：（0.35+0.35+1.20）/3 = 0.63
结论：交叉熵损失函数可以得出模型 2 优于模型 1。