

线性模型

笔记本: pytorch

创建时间: 2023/2/14 9:32

更新时间: 2023/3/23 21:50

作者: 22qdnlbn

1.一般而言, 我们需要训练数据、和验证数据集以及测试数据集。

(1)**训练数据集**: 用来训练模型

(2)**开发/验证数据集**: 用来调整模型中的参数等

(3)**测试数据集**: 数据是未知的, 用来最终评判一个模型的好坏。

- 比如一个猫狗分类的模型。

我们只有训练数据集, 测试数据集相当于用户去放上一张图片, 来预测是猫还是狗。但是此时我们只有训练数据集是不够的, 因为我们没有办法评测这个模型的好坏, 比如当训练集都是清晰的图片, 但是用户上传的都是普通手机拍的图片, 就导致它们的分布不同, 那么当该模型去实际运行时, 效果就会不好。

- 解决方法:

我们可以将已有的数据分为训练数据和开发数据集。同时我们也可以通过剪裁或者网络爬虫等方式来获得更多的图片类型。分为训练数据和开发数据可以使得我们来验证模型的好坏。获取更多图片可以使得我们能够有更好的泛化能力。

2、如线性模型 $y^{\wedge}=ax+b$

参数a和b都需要我们通过不断训练来得到一个最好的值。

Training Loss (Error)

$$loss = (\hat{y} - y)^2 = (x * \omega - y)^2$$



Mean Square Error

$$cost = \frac{1}{N} \sum_{n=1}^N (\hat{y}_n - y_n)^2$$

这个损失函数是对应一个样本的，我们应该计算的是一个数据集的损失函数，可以用mse mean square error:

- 通常来说

我们都需要为一个损失函数中的变量初始化，之后通过改变变量参数的值，来得到最好的拟合模型。

3、代码

```
#导入必要的包
import numpy as np
import matplotlib.pyplot as plt
#数据
x_data=[1,2,3]
y_data=[2,4,6]
def forward(x):
    return x*w
def loss(x,y):
    y_pre=forward(x)
    return (y_pre-y)*(y_pre-y)
w_list=[]
mse_list=[]
for w in np.arange(0.0,4.1,0.1):
    print('w=',w)
    l_sum=0
    for x_val,y_val in zip(x_data,y_data):
        y_pre_val=forward(x_val)
        loss_val=loss(x_val,y_val)
        l_sum+=loss_val
    print('\t',x_val,y_val,y_pre_val,loss_val)
    print('mse:',l_sum/3)
    w_list.append(w)
    mse_list.append(l_sum/3)
plt.plot(w_list,mse_list)
plt.ylabel('Loss')
plt.xlabel('w')
```

```
plt.show()
```