本文件夹包括distinguish.py, w.npy, b.npy, requirements.txt。

1. 数据收集与预处理

1从百度图片爬取颐和园五处景点图片各约30张。其中每处景点选20张作为训练集，10张作为测试集。

2 所有图片均压缩为统一尺寸1024\*680像素。

3 所有图片转成1024\*680\*3的矩阵，进而转成 (1024\*680\*3,1)，即(2088960,1)的矩阵。

4 将训练集和测试集的矩阵分别拼接，训练集x为(2088960,100)的矩阵，测试集x为(2088960,64)的矩阵。再设置相应景点的y为

(1,0,0,0,0),(0,1,0,0,0),(0,0,1,0,0),(0,0,0,1,0),(0,0,0,0,1)。

二，模型训练

该算法为全连接神经网络算法，层数为1.

经过一层反向传播，迭代100次最佳，学习速率为0.01，激活函数为sigmoid。

将训练好的w(权重)b(偏移量)分别保存在相应的npy文件里，随用随取。

三，函数调用

定义distinguish函数，传入一张图片，将其压缩为指定尺寸，转为指定尺寸的矩阵，加载npy文件，得到w,b，代入公式(x.T\*w+b),得到一个(5,1)矩阵,按照最大值的下标，返回这个列表

['万寿山', '东宫门', '佛香阁', '排云殿', '昆明湖'] 对应的景点。

总结：由于电脑硬件限制，以及我还没有掌握分布式运行的方法，所以对于数据集的数量一再削减，本来爬取12个景点的图片，最终只用了5个景点，每次训练时间差不多3分钟。

网络层数只有一层，虽然简陋，思想是对的。期间拼接矩阵，矩阵维度配对，梯度爆炸，内存不够等，遇到了不少困难，总算算法整体实现了。

钟德昭