Knn学习过程以及程序结果

我的环境是直接在vscode中嵌入anaconda

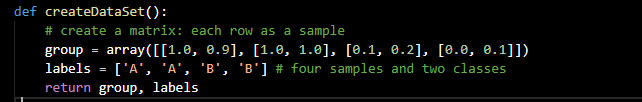
Anaconda版本：



对于knn算法，基本上的概念我上课都听懂了，感觉对于k的取值和计算距离的方法还是需要自己进一步实验才能够理解。

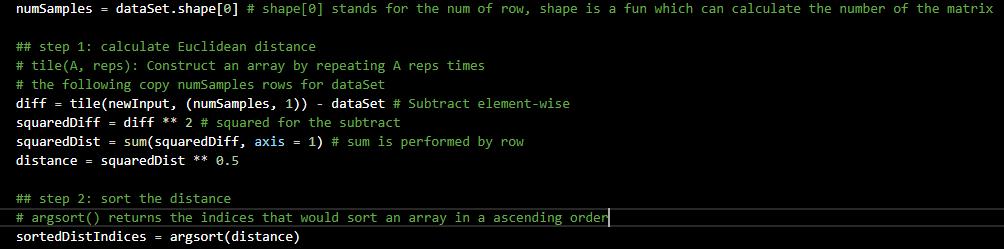
由于之前python不算特别熟，我先去看看网上已经做好的csdn的博客，正好看到了有人专门做的MINIST数字识别的项目，于是我决定学习他的代码，正好加深我对如何实现knn的操作印象。

这个是做的一个最简单的四个点的划分，具体数据以及标签定义如下：



这个函数是可以自己随意设定的一个数据集。

接下来编写分类输入数据的片段：



这里计算的是欧式距离即两点空间距离。

这里我的k取值为3。

我基本了解了knn算法的编写流程，接下来开始分析这次的任务，识别手写数字：

难点就是这次是一个图片数据，而我们这次使用的是MINIST数据集，我上网查找了MINIST数据集的存储方式：

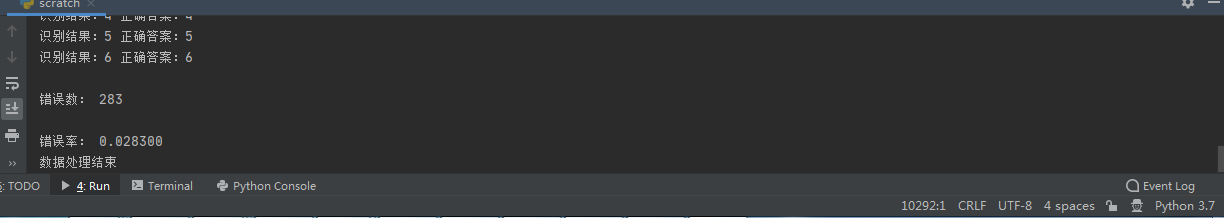
由6万张训练图片和1万张测试图片构成的，每张图片都是28\*28大小（如下图），而且都是黑白色构成（这里的黑色是一个0-1的浮点数，黑色越深表示数值越靠近1），这些图片是采集的不同的人手写从0到9的数字。TensorFlow将这个数据集和相关操作封装到了库中。这些图片会被处理成很简易的二维数组。

发现我的anaconda里面还没装tensorflow，于是我赶紧去装了个tensorflow。

后来我发现tensorflow自带的examples包貌似没更新了，于是我还是自己写转换，把图片转成矩阵。

文件读取：<https://www.cnblogs.com/x1957/archive/2012/06/02/2531503.html>

最后和之前类似knn函数，计算距离，这里我选择k=3，最后得到的结果出来很慢很慢（大概2秒判断1张图片），我决定进行优化。



我最后使用了numpy中的函数进行了加速，优化了分类函数，每秒可以判断一张照片。

我这个程序主要是慢在解码上面，如果有现成的矩阵图片信息会快很多。

此外，我尝试使用了tensorflow库里面自带的mnist数据集，运行起来快了不少。