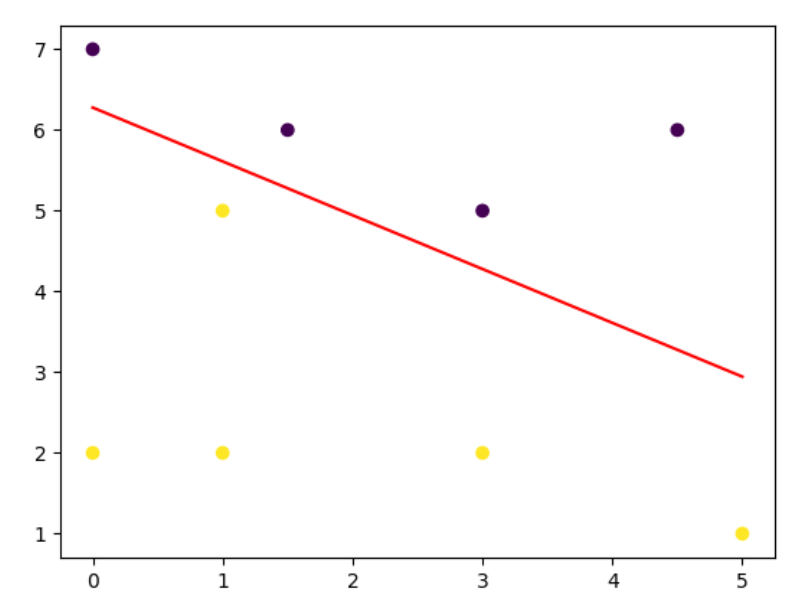
**实验报告**

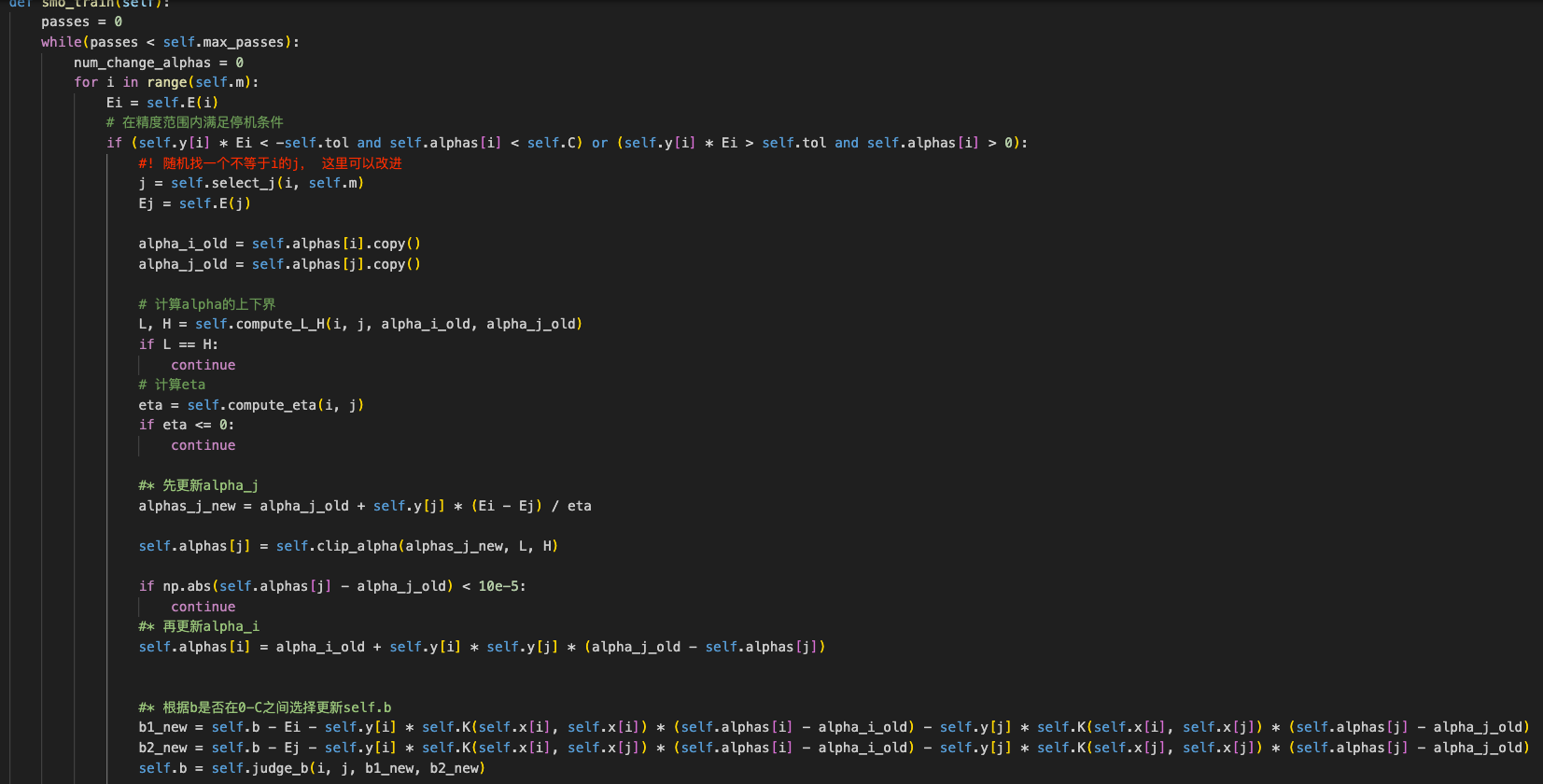
姓名：孙前普 学号：2023214320

1. **SVM的实现**

**先实现和函数采用点乘核函数的方式的线性分类器，经过调优后对标准样本进行分类，分类结果如图所示。**

****

**代码为svm\_single.py，对主要代码进行注释，后面对代码进行讲解。**

****

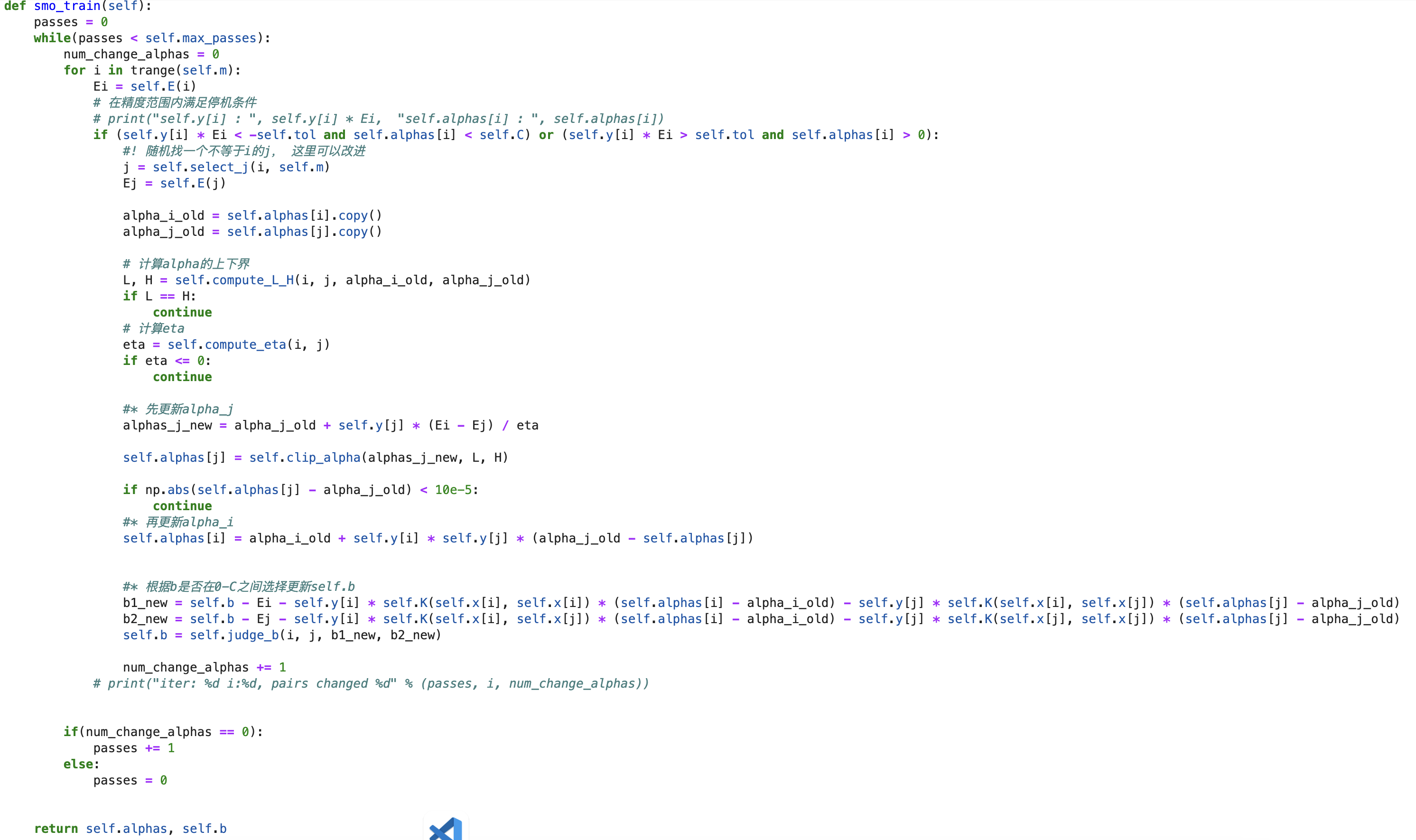
1. **MNIST数据集分类**

2.1先加载数据集MNIST

通过MNIST（）类实现对应数据集加载，得到对应数据和标签，包括one-hot类型编码（把对应0的位置设置为-1），因为svm需要标签设置为-1，1为基础svm的分类标签。最后的决策函数采用sign符号函数。

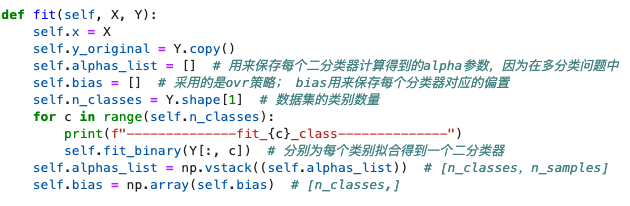
****

**2.2 svm多分类训练主体代码**

****

Passes作为循环次数，每次随机选取两个alphas值，计算对应的E值，判断是否满足停机条件，如果满足停机条件，passes++，连续max\_passes次都满足停机条件的时候，停止循环，得到最后的alpha和b值。

* 1. 拓展到多分类任务，由于svm是二分类任务，所以需要对每个样本的每个类别进行二分类任务,fit函数进行训练，得到最后的alphas，并存放到alphas\_list中， 在预测测试阶段，把每个类别的alphas取出来进行预测。



* 1. svm预测阶段

给定test数据，test数据10000个测试样本，进行测试。

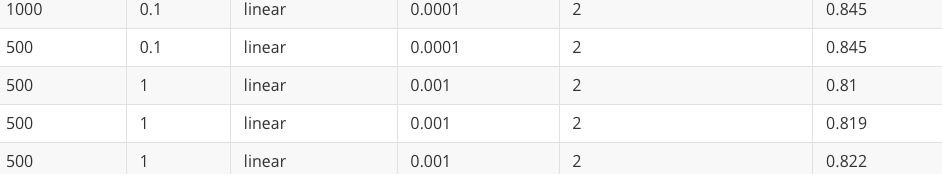
对模型进行预测，预测代码如下所示，预测每个类别的每个样本的y的分数，再进行softmax，得到最终的分类结果



2.5 最终的模型精确度计算使用sklearn库中的工具accuracy\_score

1. **对SVM进行超参数的搜索**

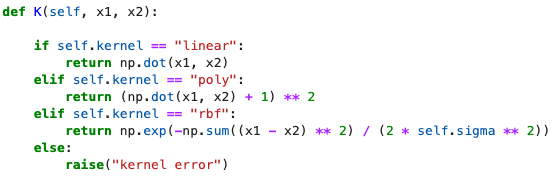
由于样本数量过大，训练起来过与繁琐，所以缩小数据集进行超参搜索。下面是结果，少量数据集上可以达到0.845，参数从左到右分别是，数据集样本数量，惩罚系数C，核函数，精度（表示可以接受最后的停机条件在这个范围内），max\_passes(表示连续几次都可以满足停机条件以后再停机)，最后是acc精度

****

**额外加分项**

1. **构建使用kernel的方法的分类器，并**

本实验直接采用核函数的方法实现，在普通方法中使用linear核函数，即两个向量进行点乘，并加入poly核函数和高斯核函数，下面是对核函数的实现



对使用poly的核函数的svm进行超参数搜索，参数从左到右分别是，数据集样本数量，惩罚系数C，核函数，精度（表示可以接受最后的停机条件在这个范围内），max\_passes(表示连续几次都可以满足停机条件以后再停机)，最后是acc精度



可以看到在全量数据上可以达到0.97的精确度，sklearn中实现的svm实现结果也是0.97

1. **对比不同的方法**

实验时间上来看，linear相比poly和rbf核函数的收敛时间较慢，由于采用的是软间隔距离，当C不是非常非常大的时候，它可以忽略掉一些异常点的影响，得到更好的决策界。甚至当你的数据不是线性可分的时候，支持向量机也可以给出好的结果。

C 较大时，相当于λ较小，可能会导致过拟合，高方差。  
C 较小时，相当于λ较大，可能会导致低拟合，高偏差。

线性核函数不是很容易拟合对应的样本数据，所以需要很小的惩罚系数，但poly可以拟合比较复杂的 函数，收敛时间较快，并且可以使用比较大的惩罚系数进行拟合