# 机器学习上机实验报告

17030140014 张笑天

目录

[机器学习上机实验报告 1](#_Toc44275922)

[1 实验C2-1 3](#_Toc44275923)

[a) 实验内容 3](#_Toc44275924)

[b) 实验步骤 3](#_Toc44275925)

[c) 实验结果 4](#_Toc44275926)

[d) 实验感想 5](#_Toc44275927)

[2 C3-1 6](#_Toc44275928)

[a) 实验内容 6](#_Toc44275929)

[b) 实验步骤 6](#_Toc44275930)

[c) 实验结果 7](#_Toc44275931)

[d) 实验感想 8](#_Toc44275932)

[3 C4-1 8](#_Toc44275933)

[a) 实验内容 8](#_Toc44275934)

[b) 实验步骤 8](#_Toc44275935)

[c) 实验结果 11](#_Toc44275936)

[d) 实验感想 12](#_Toc44275937)

[4 C4-2 12](#_Toc44275938)

[a) 实验内容 12](#_Toc44275939)

[b) 实验步骤 12](#_Toc44275940)

[c) 实验结果 13](#_Toc44275941)

[d) 实验感想 13](#_Toc44275942)

[5 C4-3 13](#_Toc44275943)

[a) 实验内容 14](#_Toc44275944)

[b) 实验步骤 14](#_Toc44275945)

[c) 实验结果 14](#_Toc44275946)

[d) 实验感想 15](#_Toc44275947)

[6 C5-1 15](#_Toc44275948)

[a) 实验内容 15](#_Toc44275949)

[b) 实验步骤 15](#_Toc44275950)

[c) 实验结果 17](#_Toc44275951)

[d) 实验感想 21](#_Toc44275952)

[7 C6-1 21](#_Toc44275953)

[a) 实验内容 21](#_Toc44275954)

[b) 实验过程 21](#_Toc44275955)

[c) 实验结果 24](#_Toc44275956)

[d) 实验感想 26](#_Toc44275957)

[8 C6-2 26](#_Toc44275958)

[a) 实验内容 26](#_Toc44275959)

[b) 实验步骤 26](#_Toc44275960)

[c) 实验结果 27](#_Toc44275961)

[d) 实验感想 29](#_Toc44275962)

[9 C7-1 29](#_Toc44275963)

[a) 实验内容 29](#_Toc44275964)

[b) 实验步骤 29](#_Toc44275965)

[c) 实验结果： 33](#_Toc44275966)

[d) 实验感想 39](#_Toc44275967)

[10 C8-1 39](#_Toc44275968)

[a) 实验内容 39](#_Toc44275969)

[b) 实验步骤 39](#_Toc44275970)

[c) 实验结果 41](#_Toc44275971)

[d) 实验感想 48](#_Toc44275972)

## 实验C2-1

### 实验内容

构建一个二分类问题：逻辑与(AND)的一般问题。

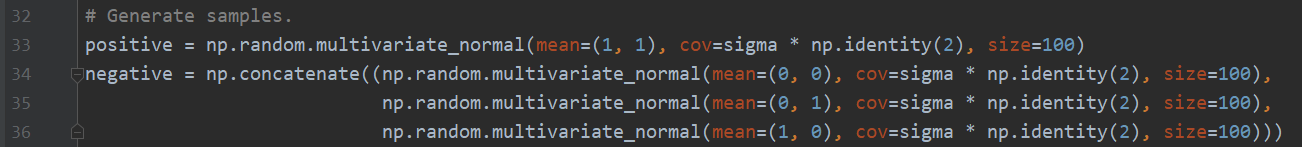
生成数据：正样本样本服从高斯分布，均值为[1,1]，协方差矩阵为单位矩阵；负样本服从三个高斯分布的混合高斯分布，这三个高斯分布的均值分别为[0,0], [0,1], [1,0]，每个高斯分布的协方差矩阵均为(sigma\*单位矩阵)。

学习：设sigma=0.01，请依上面的分布生成正负样本各300个，运用perceptron learning algorithm从数据中学习出一个perceptron，实现对正负样本的二分类。

实验与讨论：请通过编程实验，讨论如下问题：a. 学习算法的收敛性与哪些因素存在怎样的关系？b. 讨论当sigma取值不断变大（如取值从0.01-1）情况下，学习算法的收敛性问题，由此可以得出怎样的结论？

### 实验步骤

利用numpy方法生成二维高斯分布：



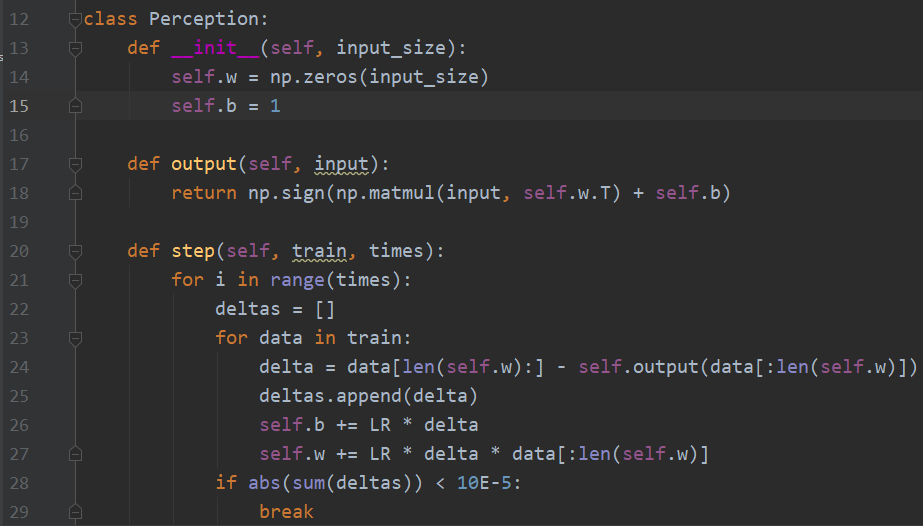
构建感知机类：

要点：

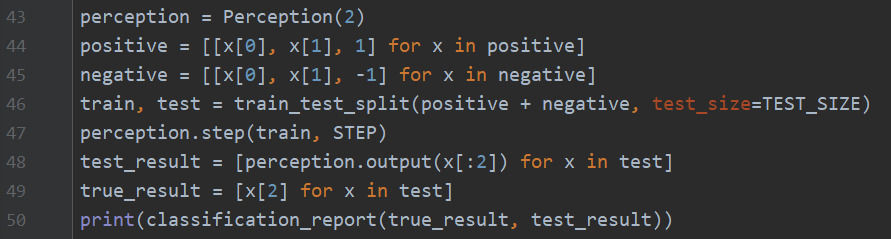
使用bias，否则只有无法得出结果

output使用符号函数

利用损失函数使得训练可以在适当时候停止



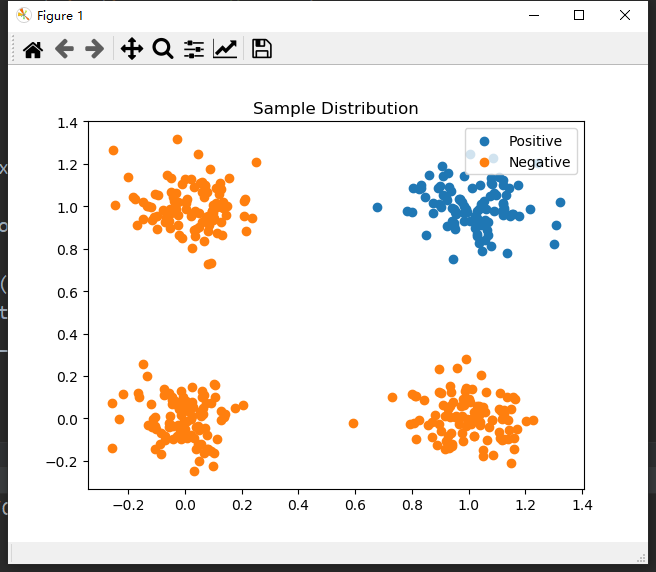
构建训练集、测试集并进行训练，并最后评估测试结果：



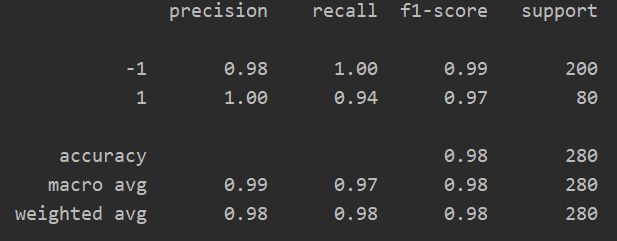
### 实验结果

Sigma=0.01时

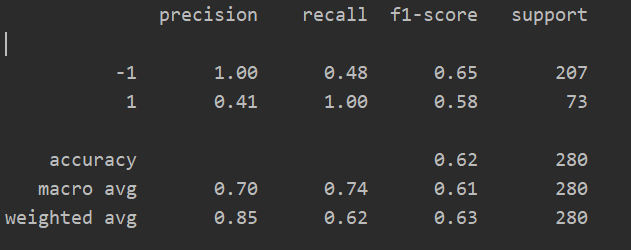
样本：



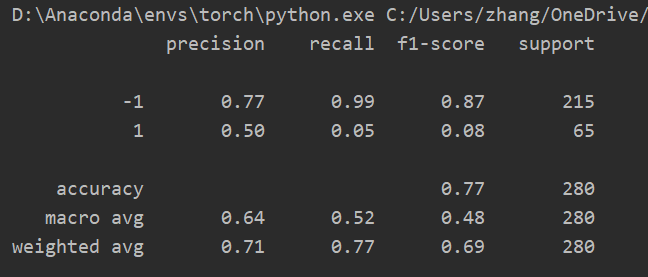
结果：



Sigma=0.1：



Sigma=0.99



结论：

训练结果和权重初始化、bias初始化、感知机结构、训练次数、训练终止方式、学习率等有关系。

Sigma变大，算法收敛性减小，说明学习效果与样本分布有很大关系。当样本无法划分时，学习效果很差。

### 实验感想

学习到了具体的感知器设计，对于包的使用更加熟悉。

## C3-1

### 实验内容

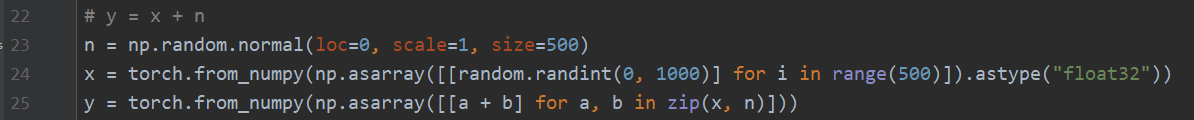
生成500个数据点(x, y), y = x + n, n为均值为0标准差为delta的正态分布。请使用线性回归算法从输入x估计y。

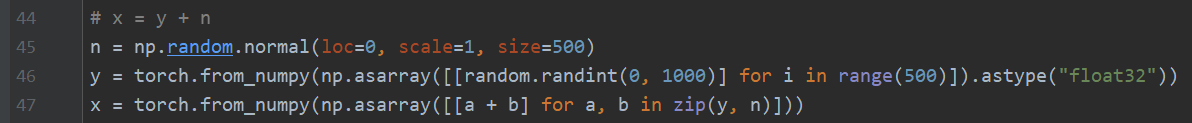
对于x = y + n进行同上述操作，但是依旧从x估计y。

对于上述题目的回归曲线进行比较。

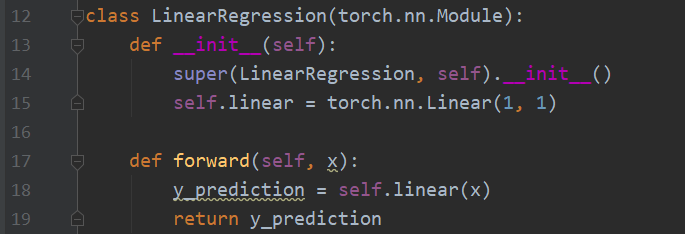
### 实验步骤

构建数据：

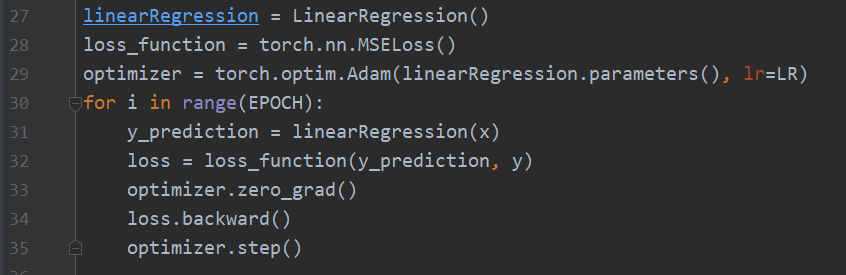




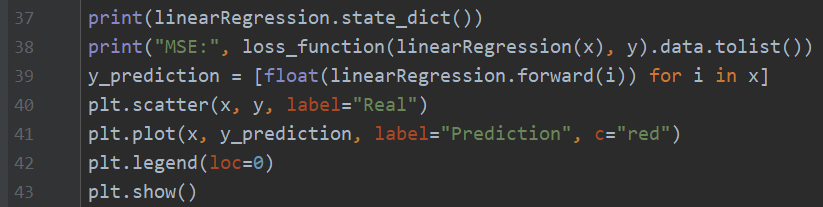
采用最简单的1-1网络：



训练：

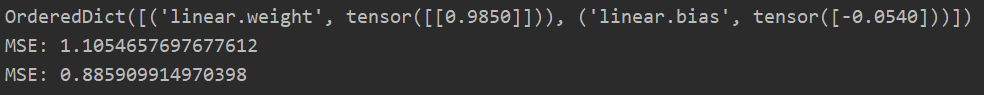


结果显示：



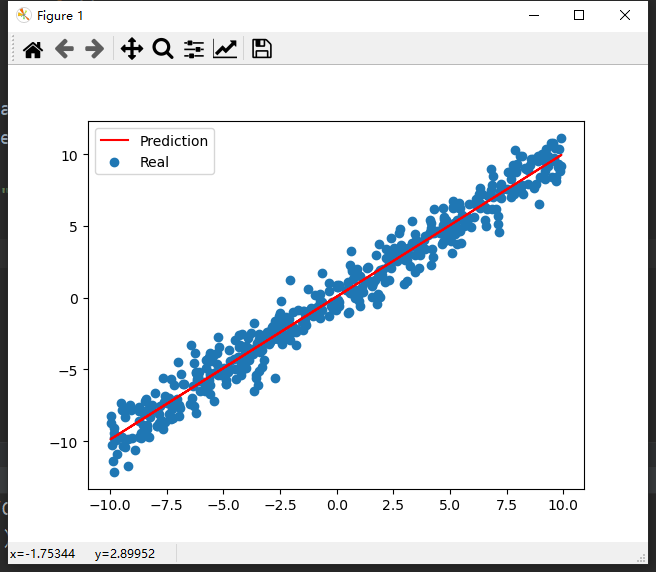
### 实验结果

参数和MSE：

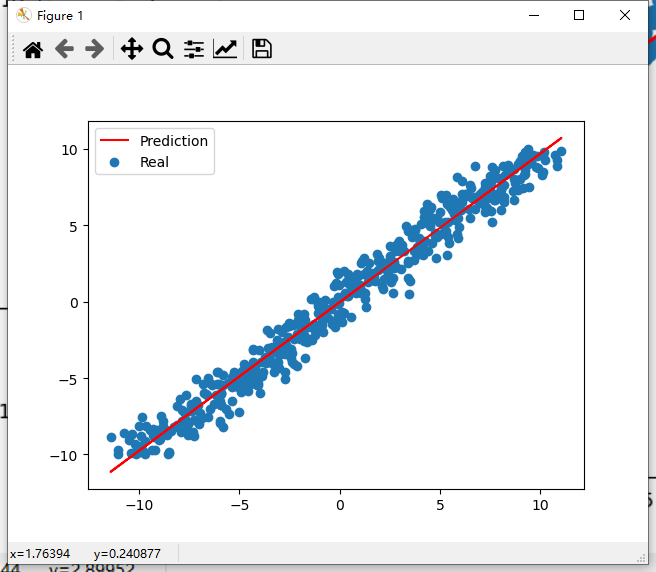


回归曲线：

第一个：



第二个：



结论：

可能题意理解错误，未看出差别

### 实验感想

可能本题题意理解有错误，或者使用了pytorch结果过好。

## C4-1

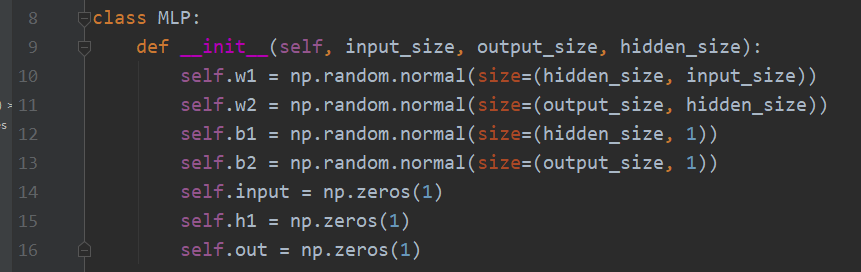
### 实验内容

写一个带有bias的基础的3-3-1网络反向传播程序，来解决三比特奇偶校验问题。偶数个1bit返回+1，反之返回-1。展示输入-隐层的参数并且分析每个隐层神经元的函数。

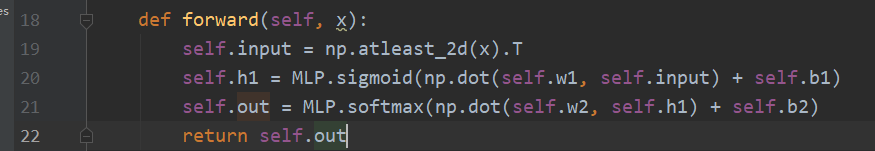
### 实验步骤

自己写的BP，使用softmax分类，也可以使用sigmoid求值：

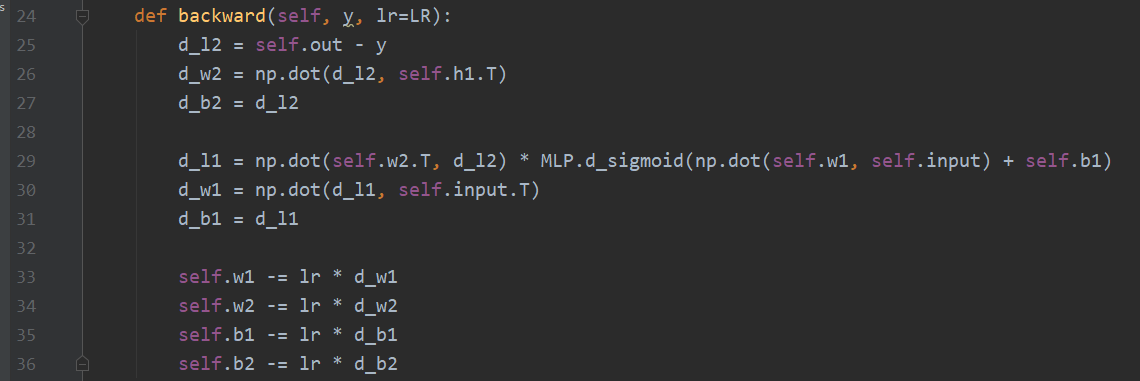
初始化



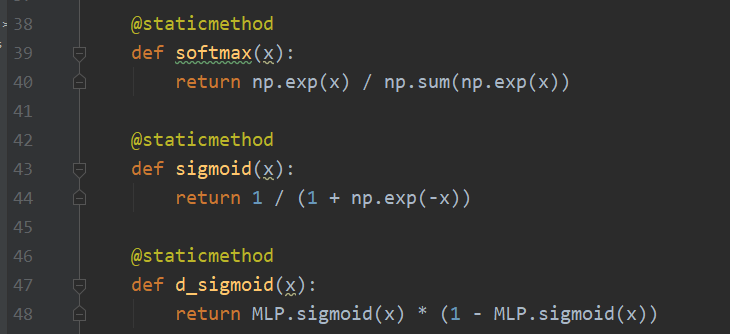
前向传播：



反向传播：

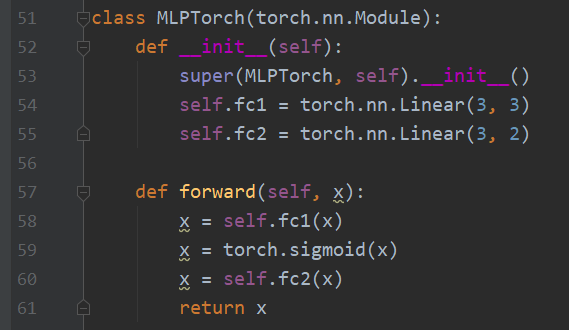


功能函数：



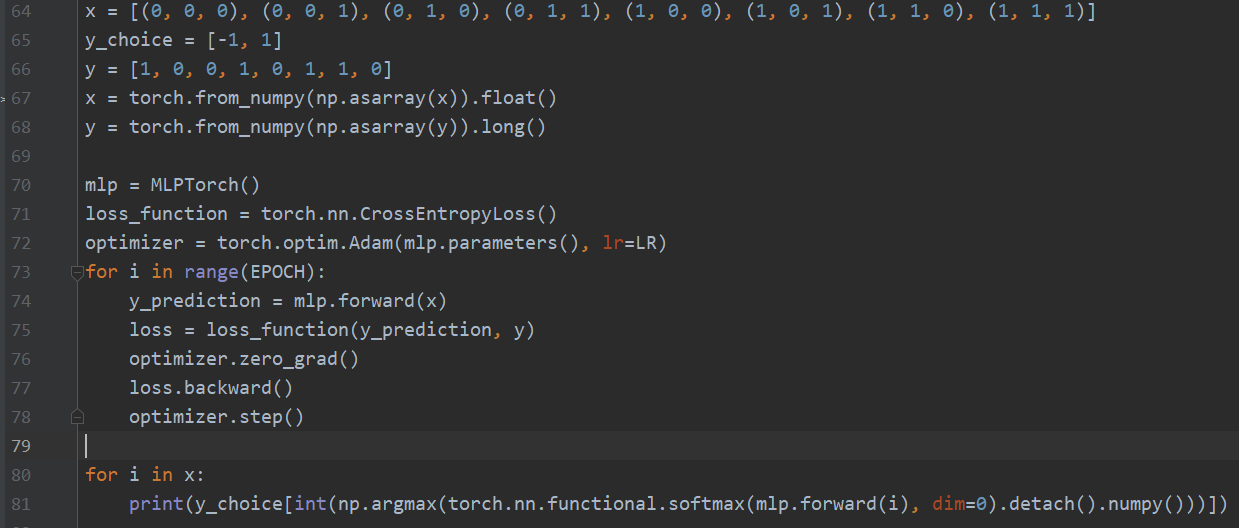
Pytorch版本：

定义MLP，forward最后不进行softmax因为torch的交叉熵自带

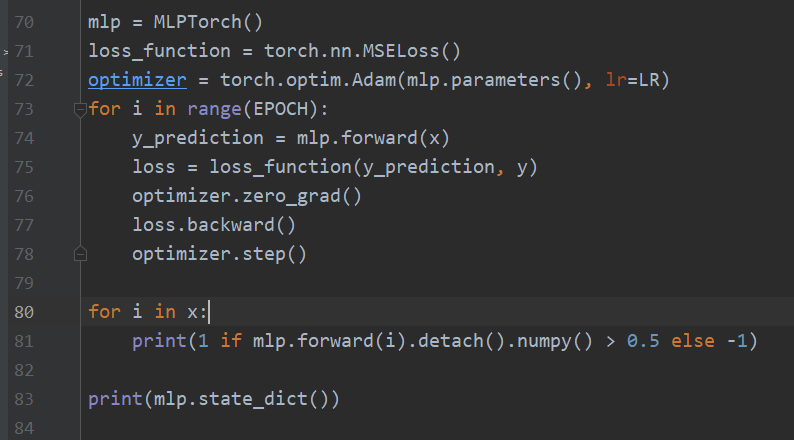


训练：

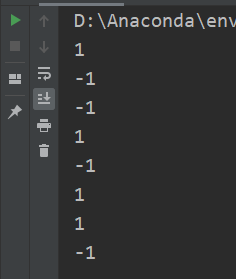
二分类：



直接求值：



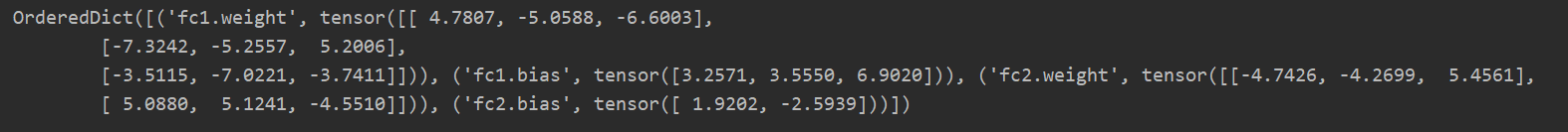
### 实验结果



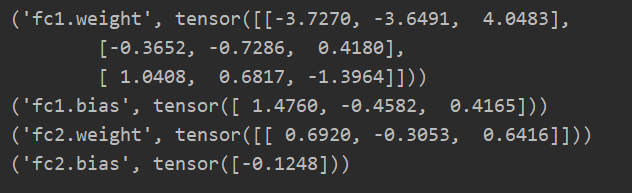
与预估相同

参数如下：

二分类：



Sigmoid：



对于两个输入求异或，之后对于结果和另一个输入求异或

参考：

<https://thalis.math.upatras.gr/~vrahatis/papers/proceedings/PlagianakosV99c_Proc_GECCO-1999_July_13-17_1999_Orlando_USA_vol_1_pp910-915_1999_CameraReady.pdf>

### 实验感想

深入学习了BP算法。对于奇偶校验有了更多认识。

## C4-2

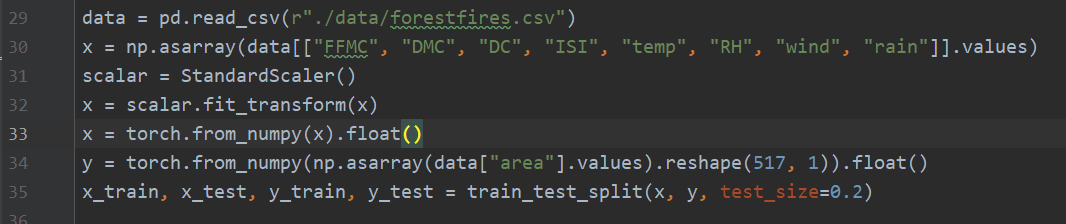
### 实验内容

训练神经网络，计算森林火灾面积

### 实验步骤

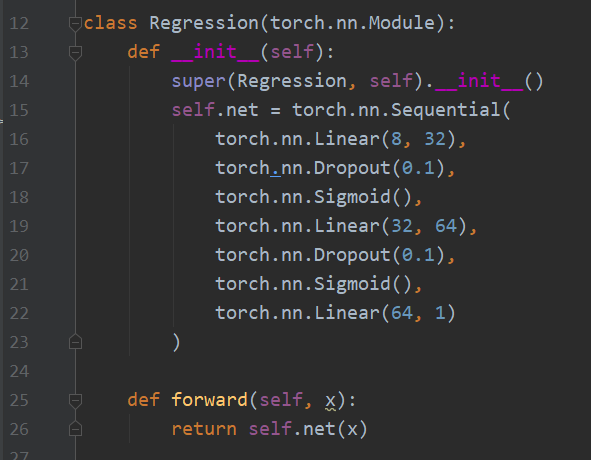
数据读取和预处理：

使用了8个属性

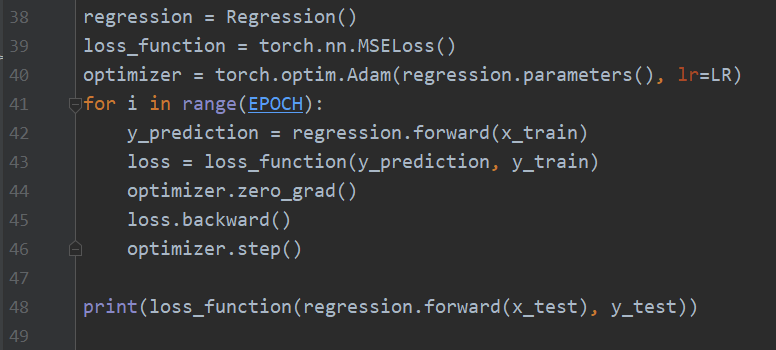


神经网络构造：

使用了dropout防止过拟合



训练与验证结果：



### 实验结果



没有花费很多时间优化参数，所以结果可能较差。

### 实验感想

本次没有很多时间优化，因此结果可能较差。感觉数据量过少，线性回归效果可能不好。

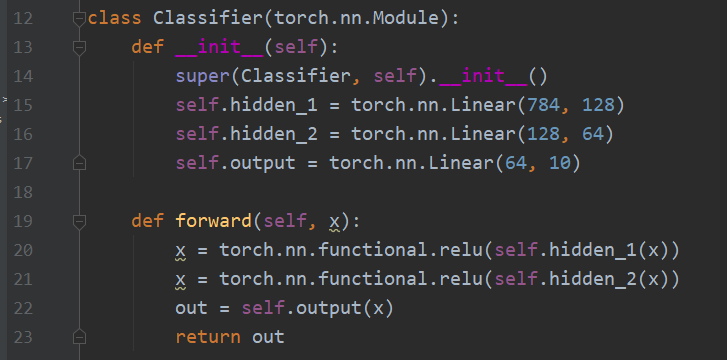
## C4-3

### 实验内容

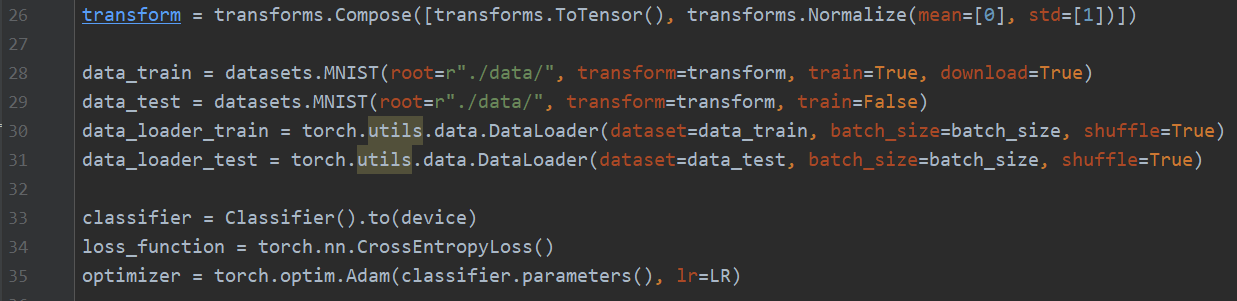
Mnist

### 实验步骤

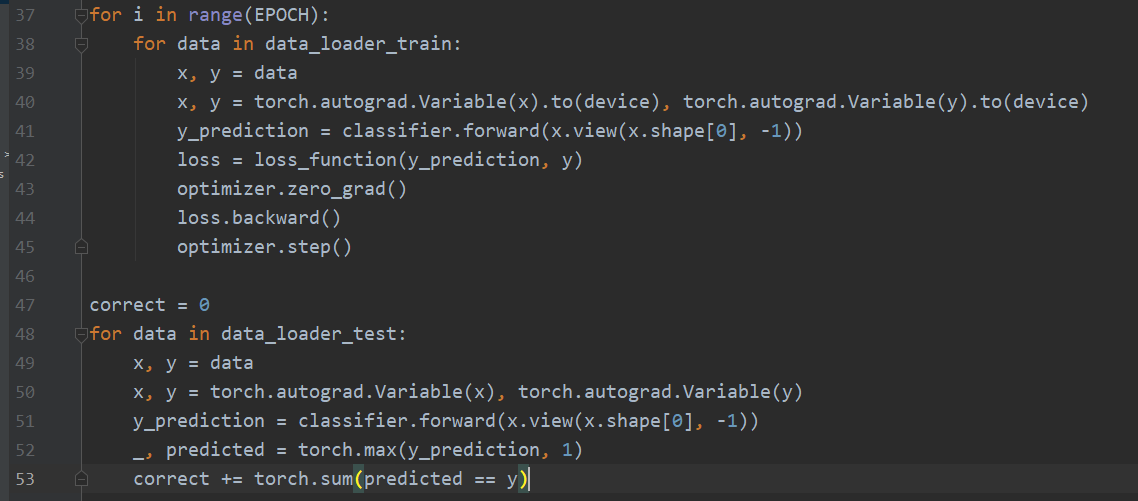
定义模型，未使用CNN：



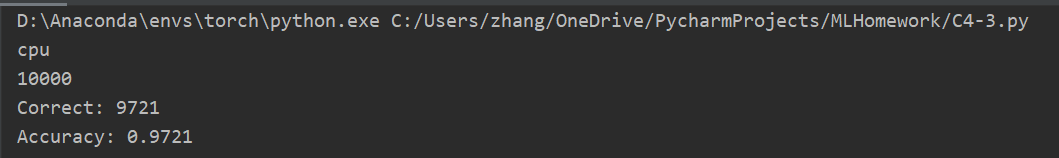
预处理：



训练与测试：



### 实验结果



准确率达97.21%

### 实验感想

因为无法使用GPU所以未使用CNN，Mnist的解决方法已经相当成熟，即使不用CNN也能达到很好结果。

## C5-1

### 实验内容

构建一个类别不平衡的二分类问题。

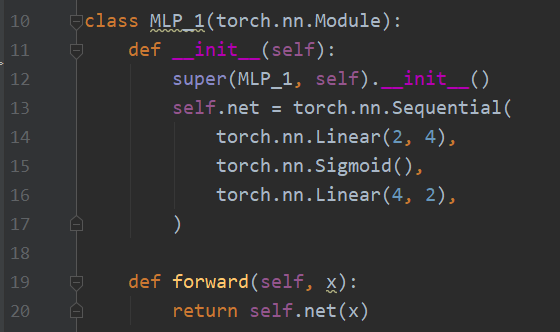
（1）生成数据：正样本样本服从高斯分布，均值为[2,3]，协方差矩阵为单位矩阵；负样本服从高斯分布，均值为[5,6]，协方差矩阵为单位矩阵。

（2）学习：请依上面的分布生成正样本200个，负样本800个，将其划分为包含60%样本的训练集、20%样本的验证集和20%样本的测试集，通过分别构建两个不同的MLP模型实现对正负样本的二分类。其中第一个MLP模型含有一个隐层，第二个MLP模型含有两个隐层。

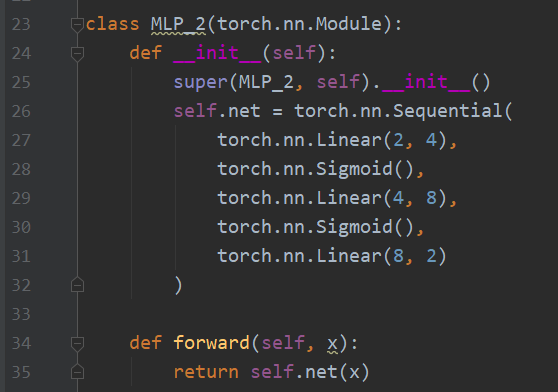
（3）实验与讨论：请通过编程实验，讨论如下问题：a. 若要求MLP模型对于正例样本的预测有更高的查准率和查全率，请考虑在模型选择中采用哪种性能衡量指标；b.通过绘制学习曲线，分析模型的偏差与方差；c.通过绘制ROC曲线，比较两个MLP模型。

### 实验步骤

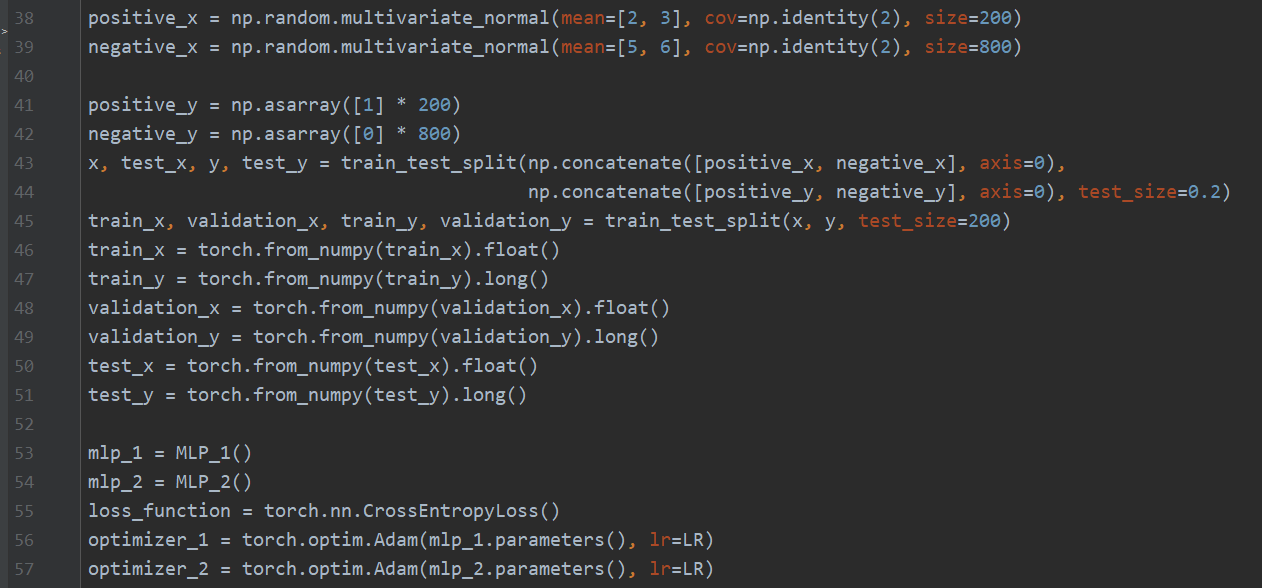
MLP\_1 一个隐层



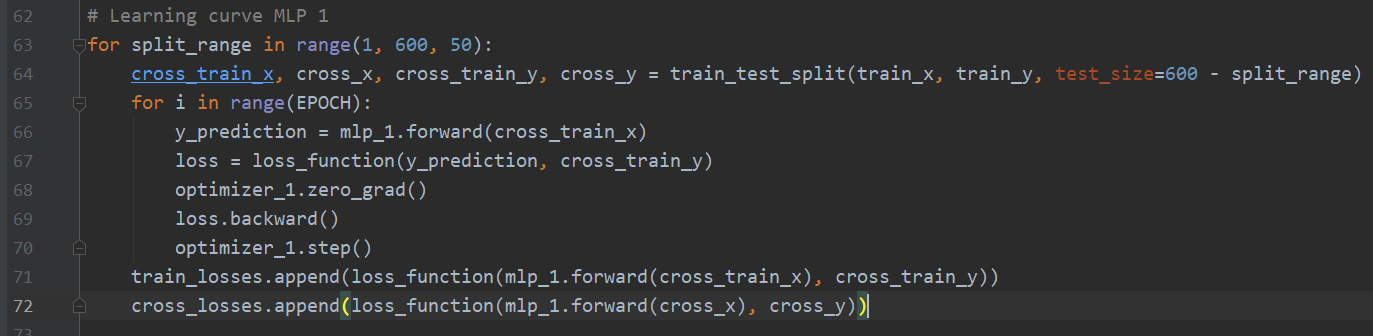
MLP\_2 两个隐层



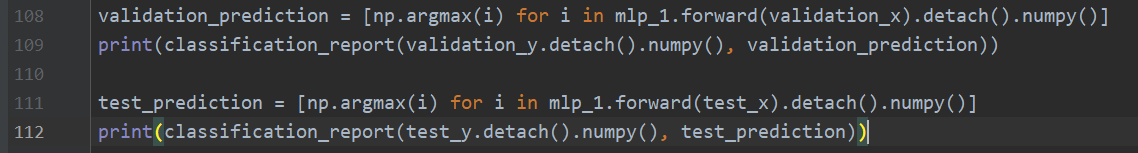
初始化：



交叉验证得到学习曲线：



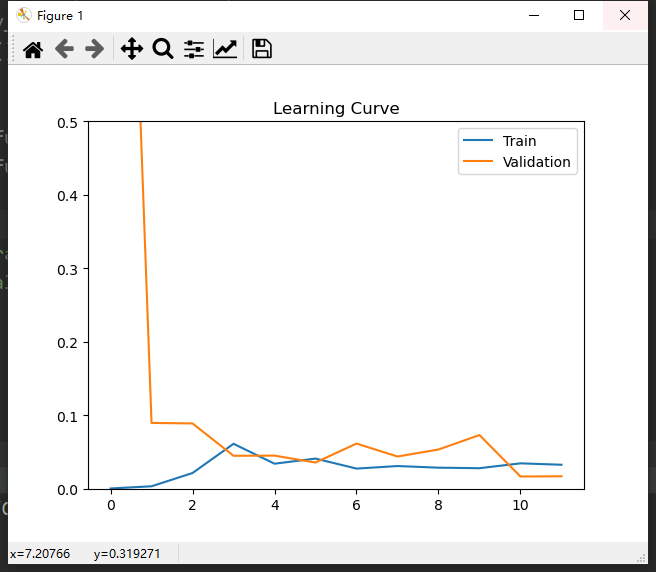
得到性能指标：



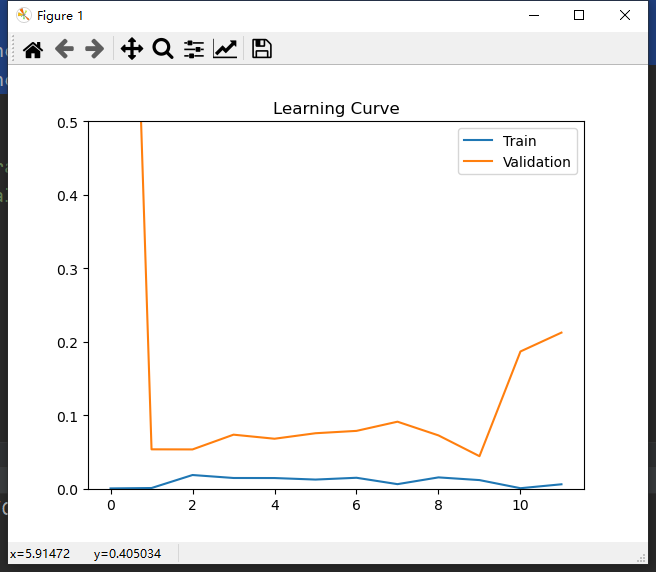
### 实验结果

使用交叉熵作为损失函数

MLP\_1的学习曲线，每次增加50个训练样本：

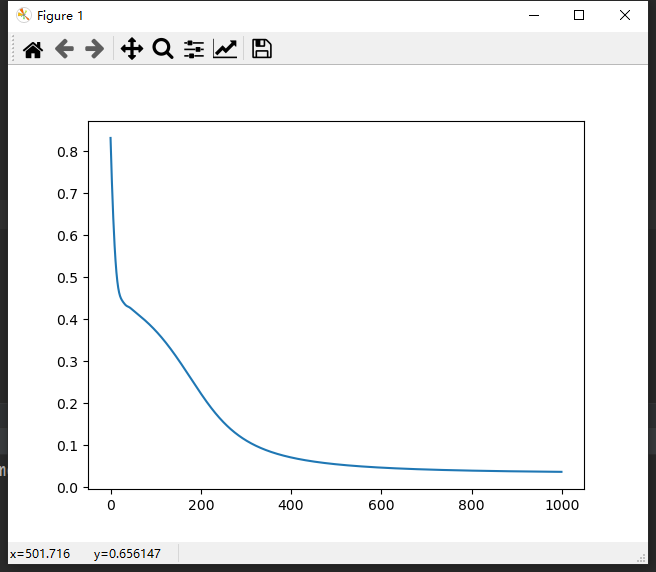


MLP\_2的学习曲线：

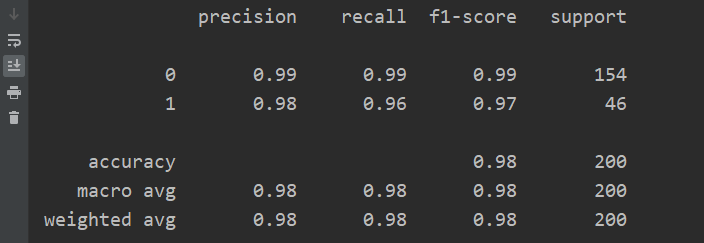


可以看到二者的偏差都较小，但是第二个模型的方差更大，说明出现过拟合

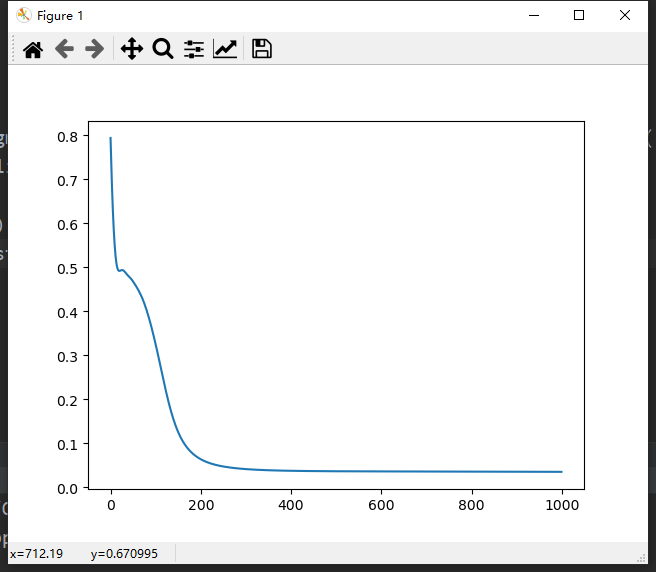
MLP\_1训练时的loss：



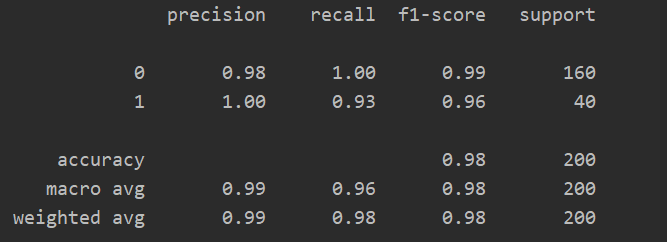
MLP\_1验证集/测试集上的性能指标：



MLP\_2的loss：

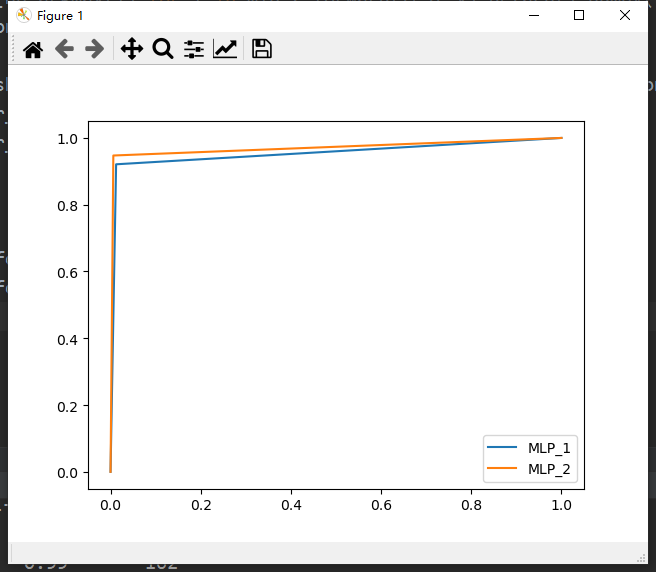


MLP\_2验证集/测试集上的性能指标：

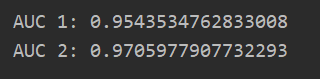


要求对于正率更高准确率和查全率，应该考虑正例的f1性能指标。

ROC曲线：



AUC：



可以看到，二者结果十分相似，其中mlp\_2在正例上可能过拟合更严重一些，但是总的结果更好。

### 实验感想

实际上使用了交叉验证绘制学习曲线，对于验证集并没有过多使用。

## C6-1

### 实验内容

对于Fisheriris数据集

使用一对多SVM针对四个输入数据二分类进行三分类；

使用KNN分类；

使用MLP二分类并最后组合；

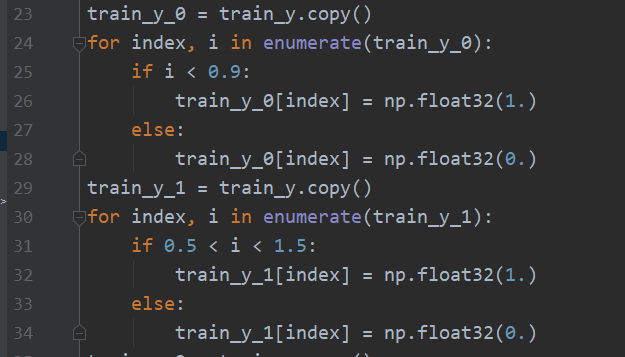
### 实验过程

数据预处理：

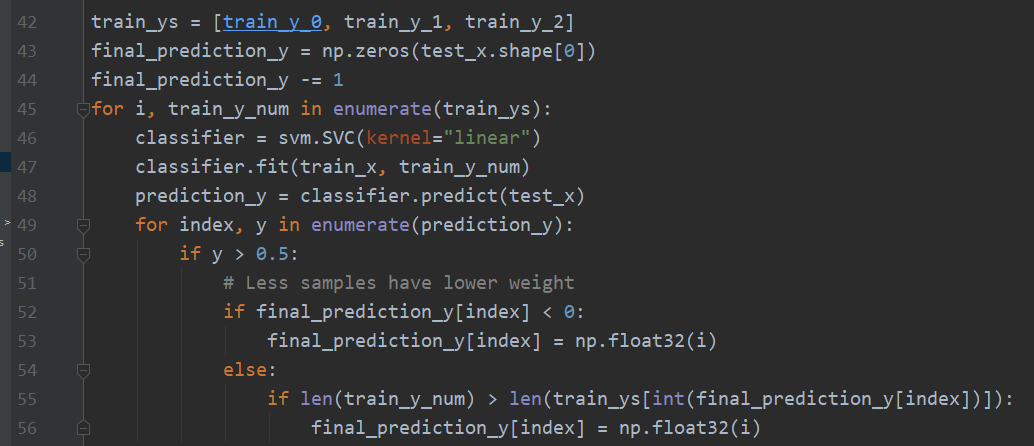


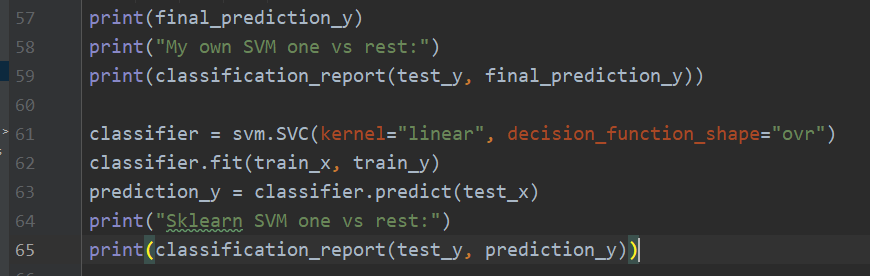
SVM：

将0， 1， 2数据作为正例，其他作为反例，其他两个亦同下图



训练三个分类器，并进行组合

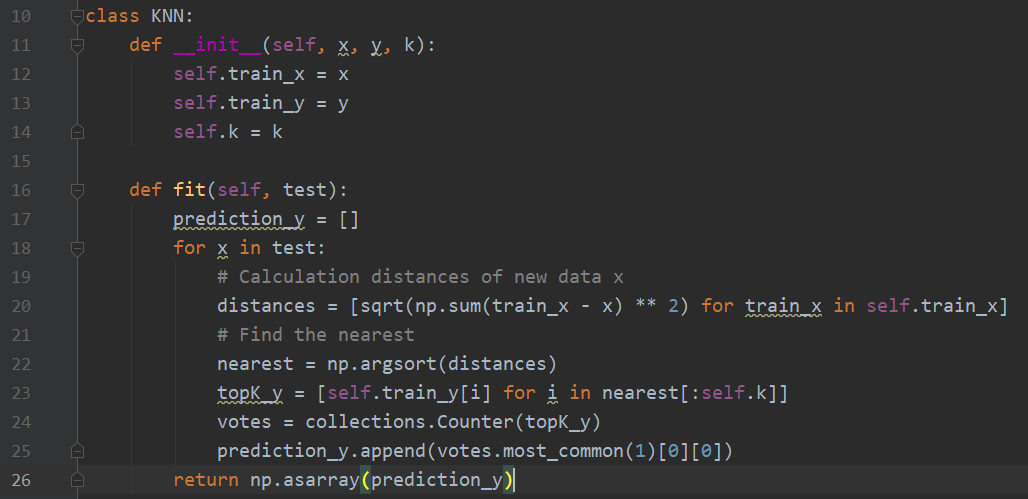




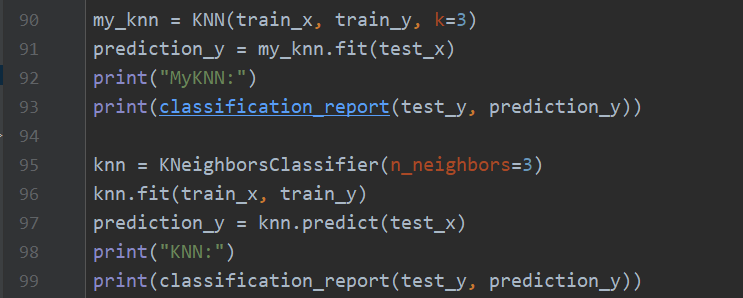
评估结果，分别是自己的一对多和sklearn的一对多

KNN：

自己完成的KNN：



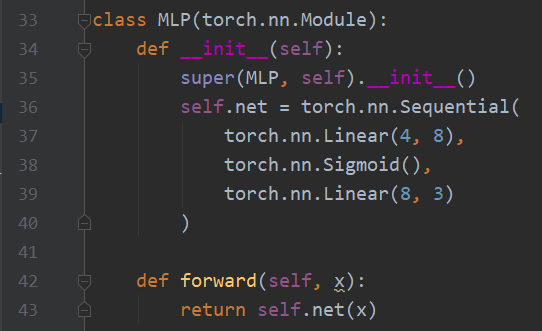
Sklearn的KNN：

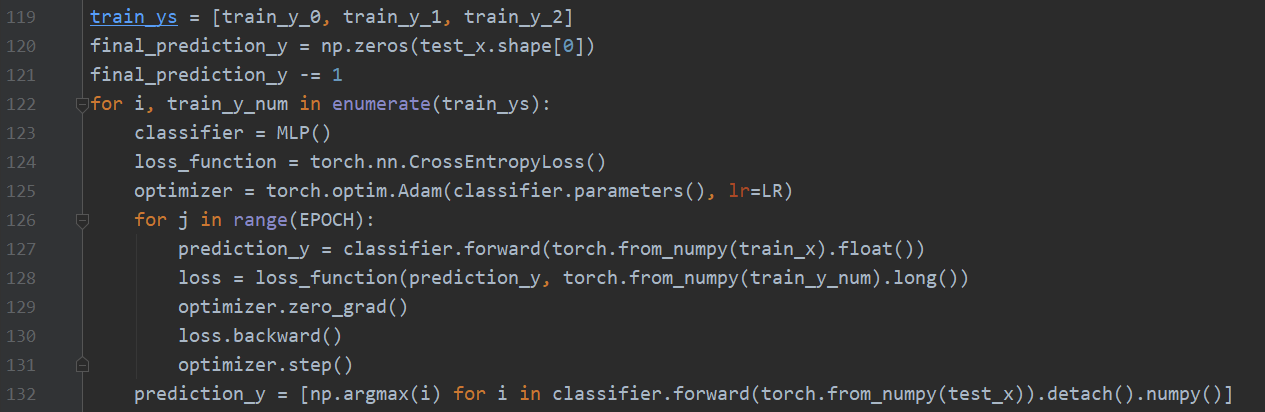


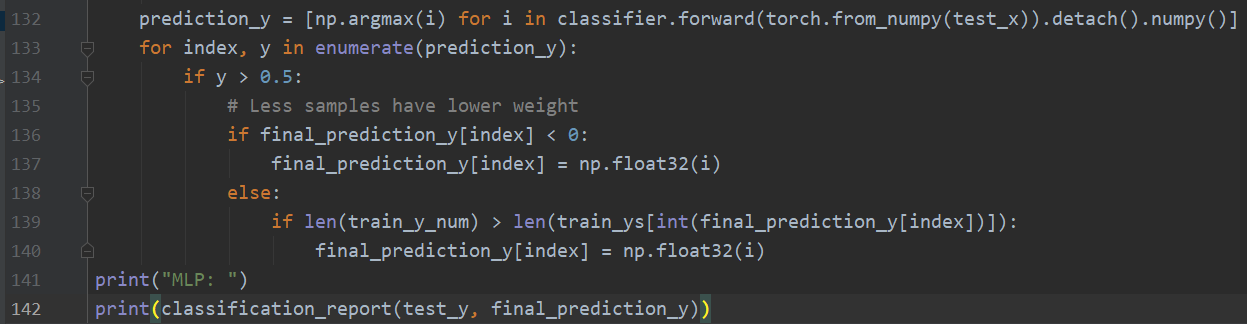
MLP：

实际上过程类似SVM

网络：



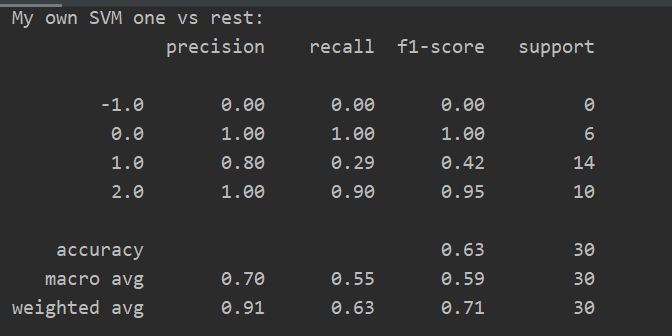




### 实验结果

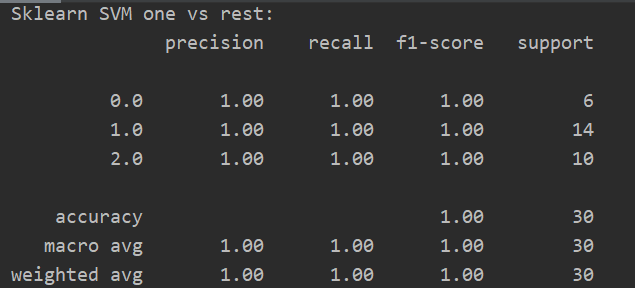
SVM自己完成的一对多

-1为无法分类的数据，无法很好的处理



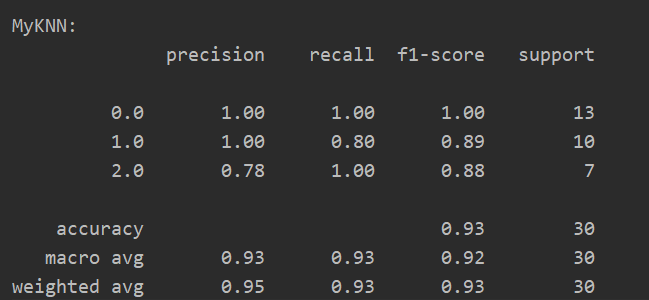
Sklearn的一对多

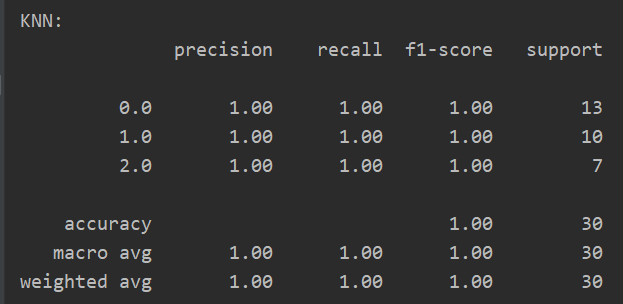
效果非常好



KNN：

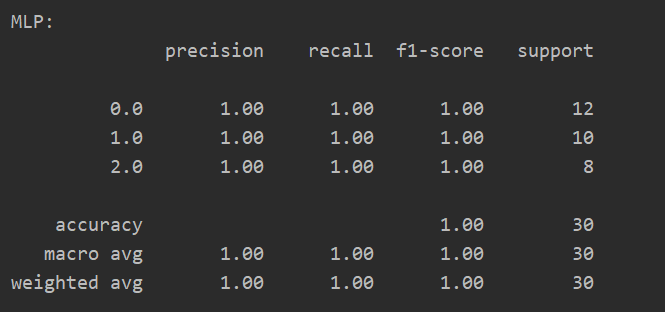
分别是自己完成的knn和sklearn的knn





MLP结果：

效果很好



### 实验感想

使用了多个分类方式，对于分类器有了更多理解。

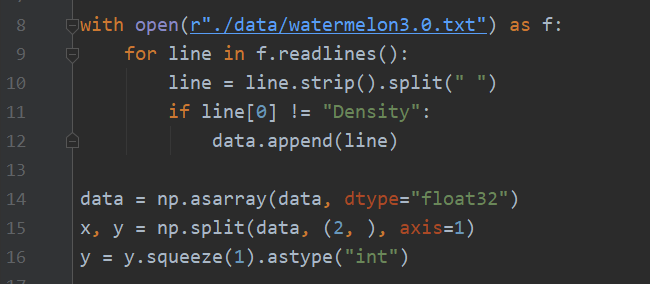
## C6-2

### 实验内容

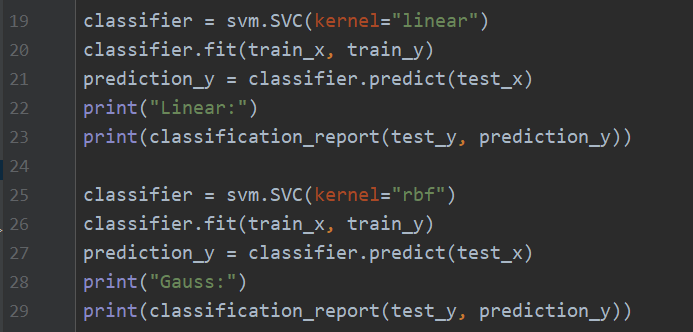
在西瓜数据集3.0上分别用线性核和高斯核训练SVM，并比较其支持向量的差别。

### 实验步骤

数据预处理，不使用test，因为数据量过少：

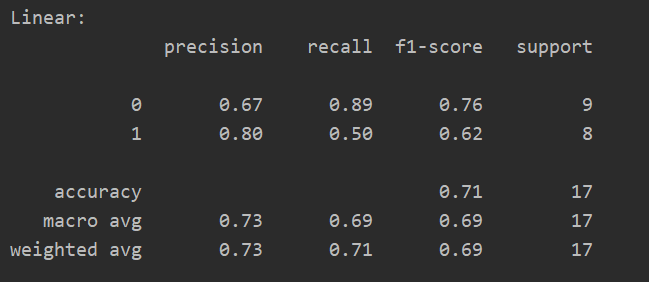


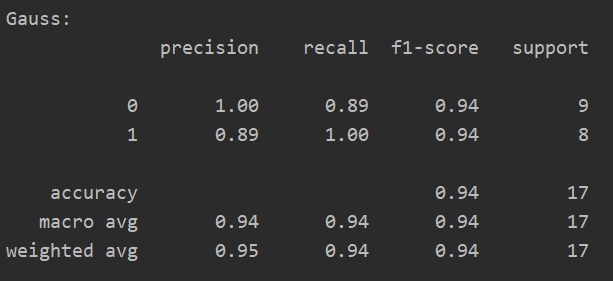
线性核和高斯核：



### 实验结果

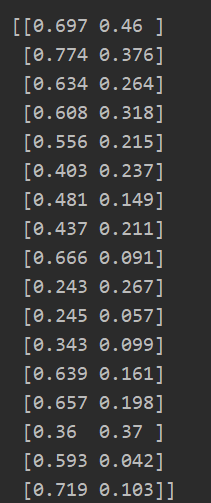
可见，高斯核好于线性核



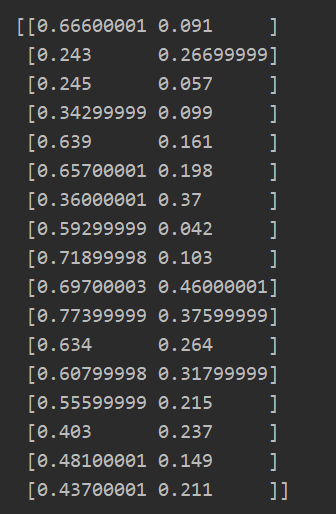


支持向量均为17x2向量

线性核支持向量：



高斯核支持向量：



### 实验感想

对于SVM还是不够了解，第一个实际使用了LinearSVM，和SVM有一些不同。

<https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html?highlight=linearsvc#sklearn.svm.LinearSVC>

## C7-1

### 实验内容

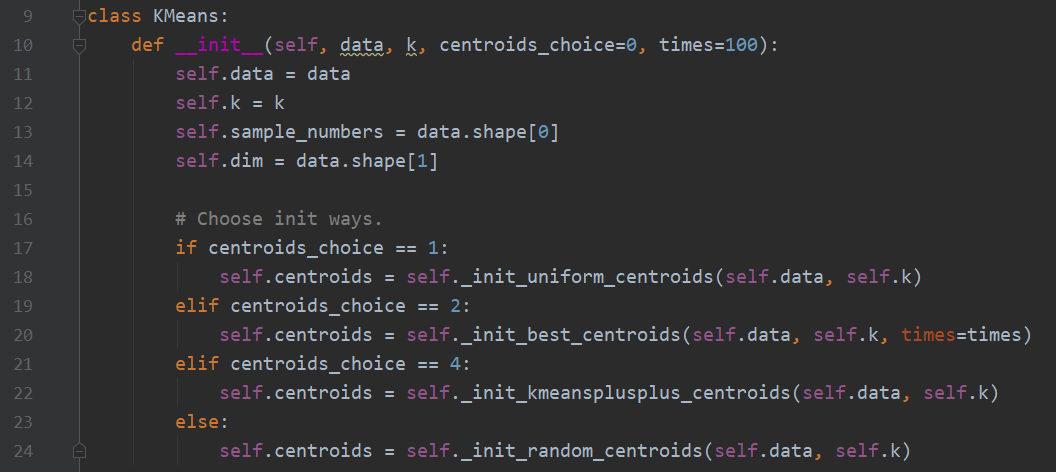
试编程实现K-means算法，设置三组不同的k值、三组不同的初始中心点，在西瓜数据集4.0上进行实验比较，并讨论什么样的初始中心有利于取得好结果。

### 实验步骤

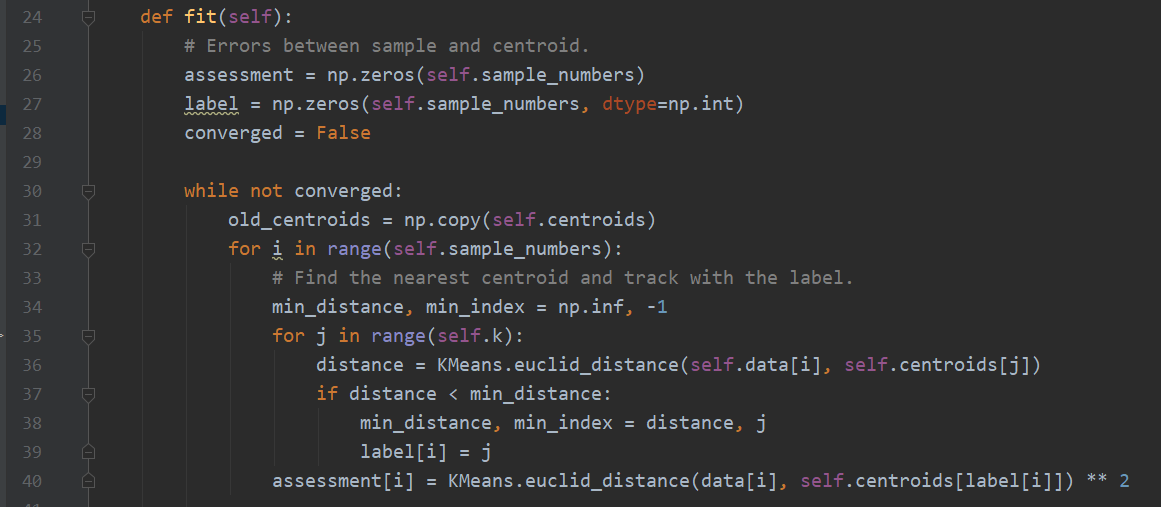
构建自己的KMeans类：

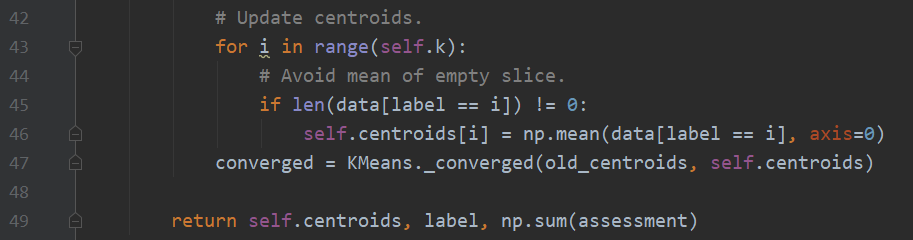
初始化：

利用参数选择质心初始化方式

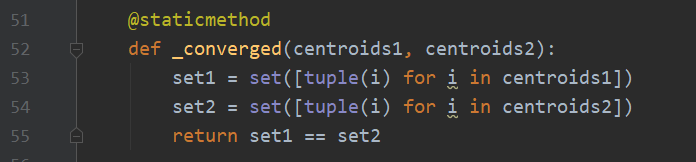


聚类的过程：

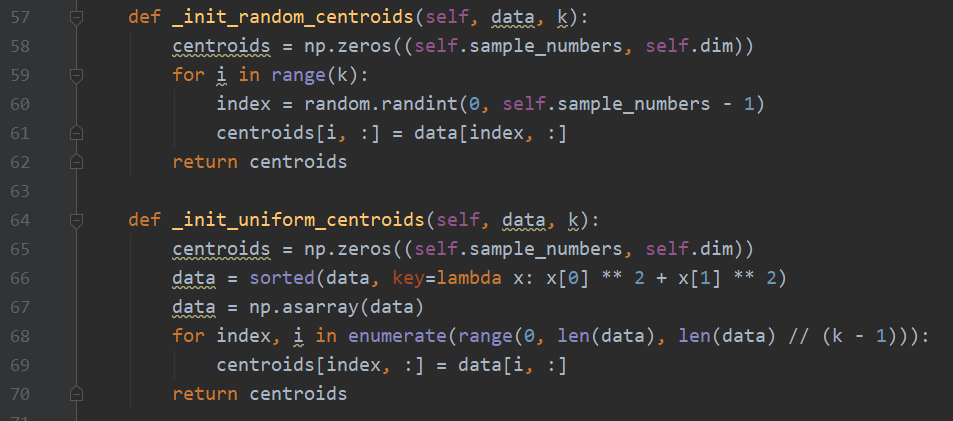


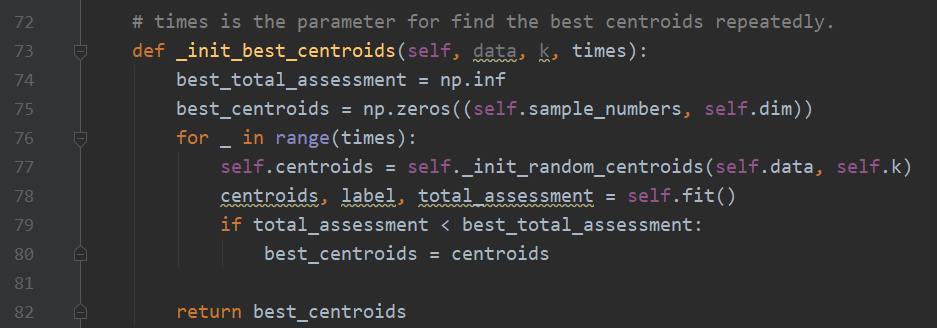


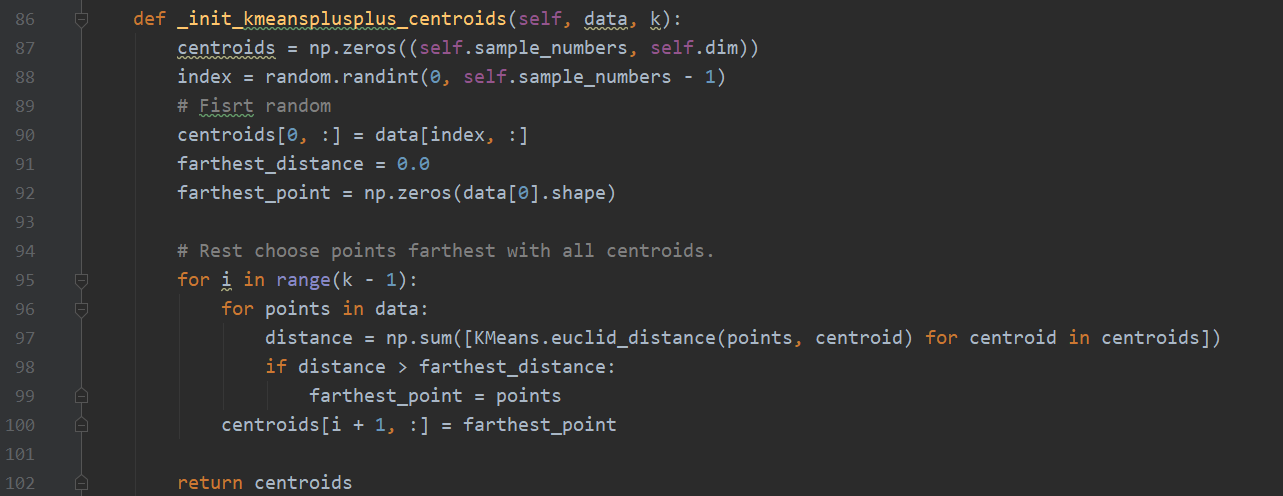
判断聚类结果是否已经不变：



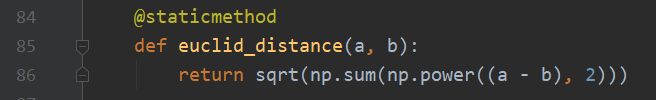
四个选择质心的方法，分别为随机选择，将数据排序后均匀选择，选择最好质心和KMeans++：



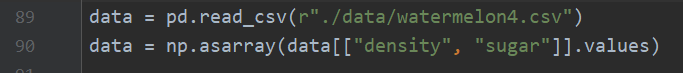




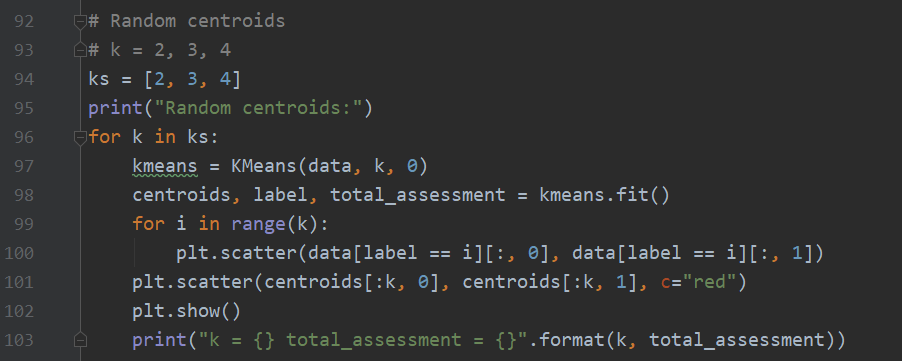
欧氏距离计算：



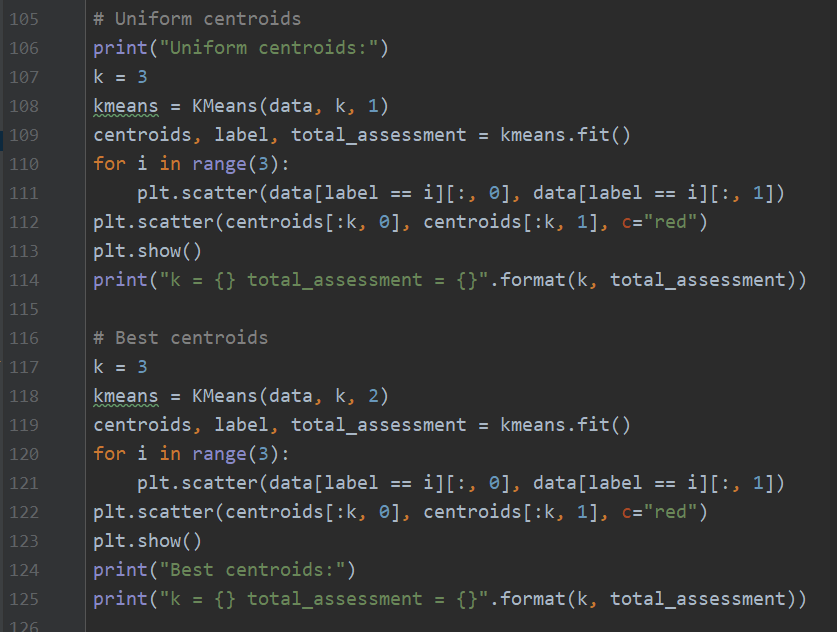
数据预处理：



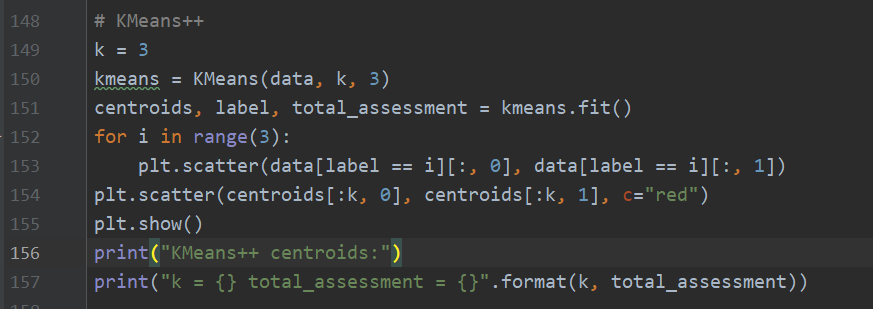
随后分别求k=2，3，4时的随机选择结果：



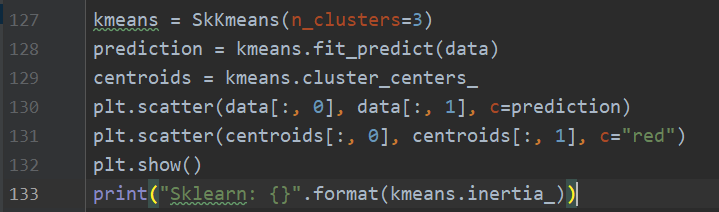
以及k=3时的均匀选择和最好结果：



K=3时 KMeans++：

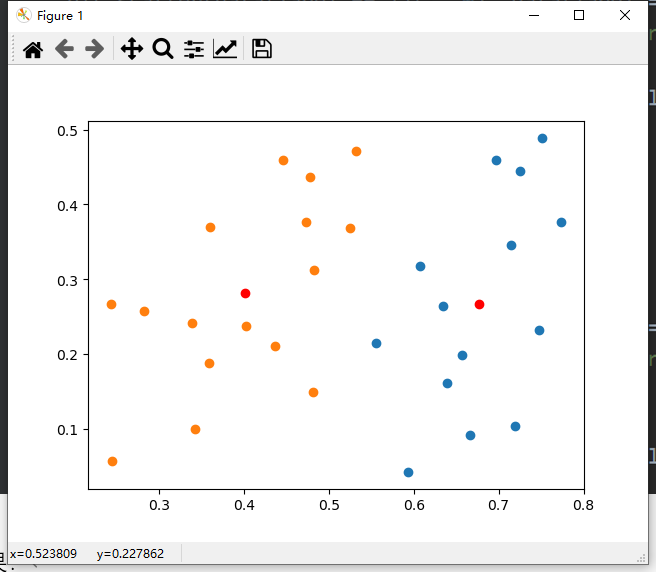


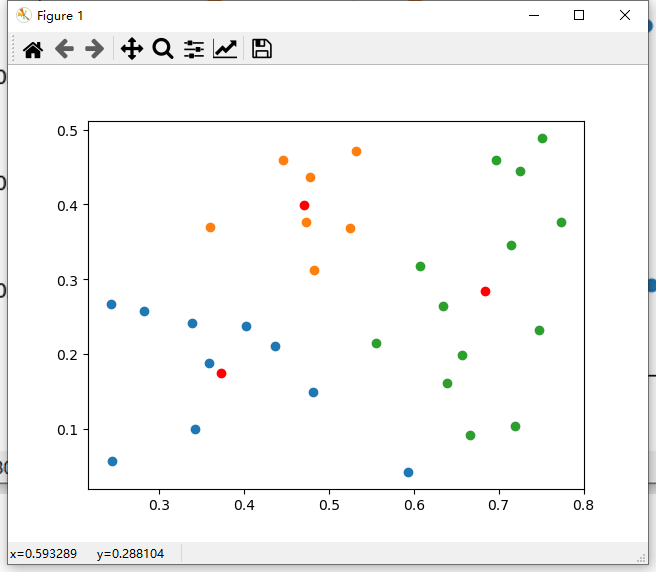
使用Sklearn的Kmeans++选择质心：

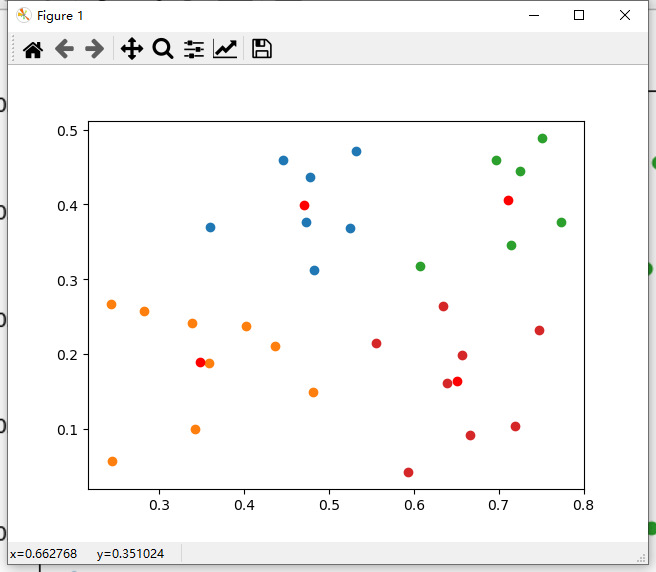


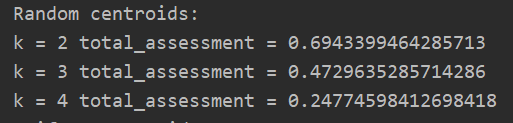
### 实验结果：

以下分别为k=2，3，4时，质心随机选择的聚类图像以及评估距离，质心为红色：

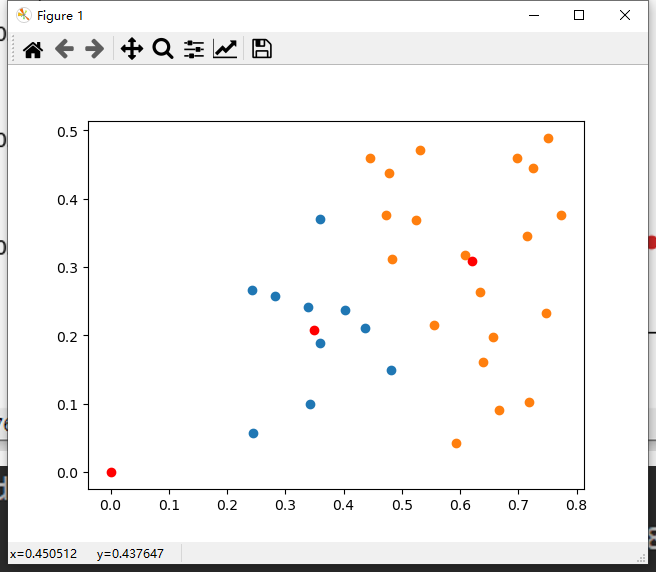






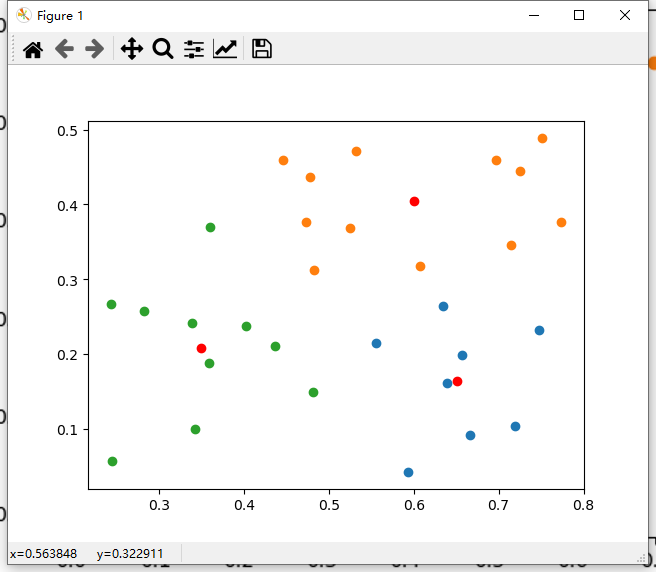


以下为k=3，均匀选择质心结果：



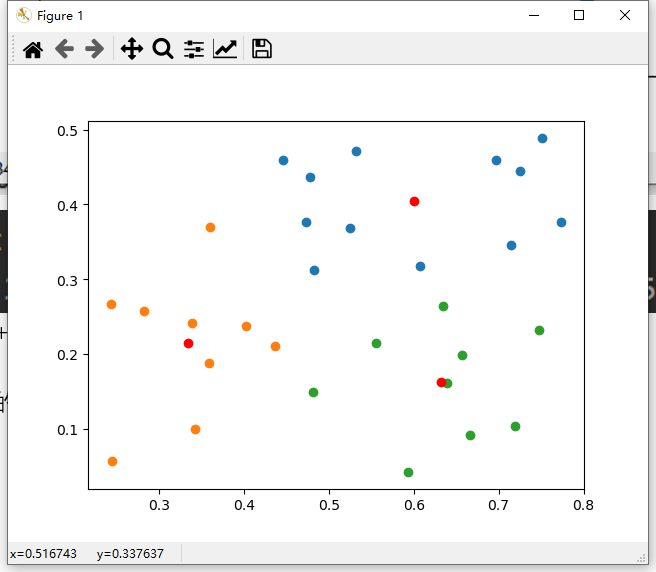


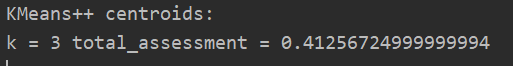
以下为k=3的最好质心：



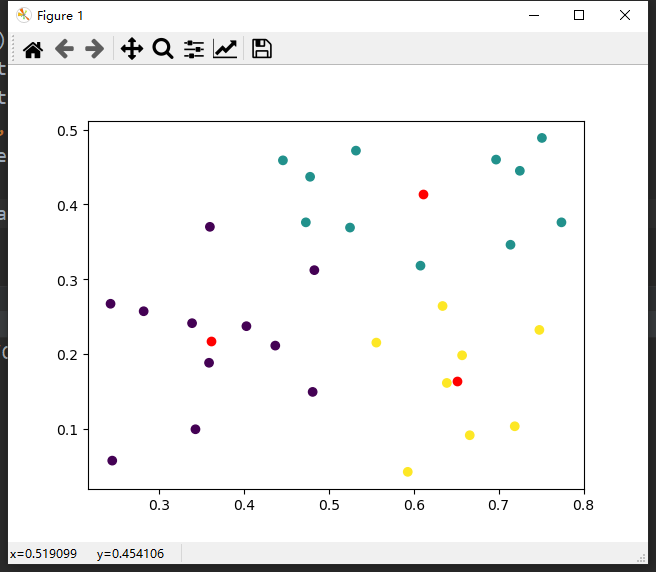


KMeans++的结果：





Sklearn的算法结果：





可以看到，KMeans++可以得到非常优秀的质心选择结果

### 实验感想

自己实现了多种KMeans，很有成就感

## C8-1

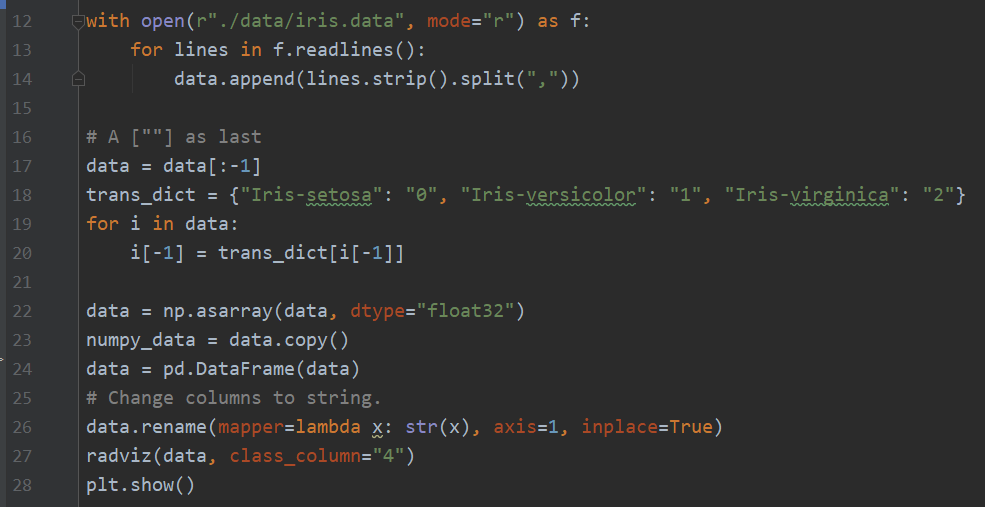
### 实验内容

请通过对Iris data的可视化，比较PCA、LDA和ICA的可视化效果。

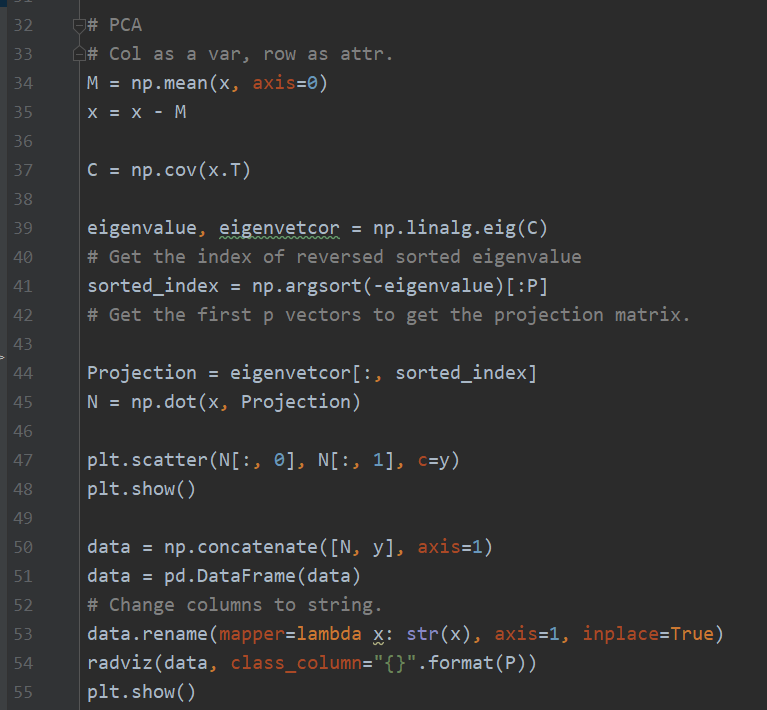
### 实验步骤

自己实现了PCA，其他调包

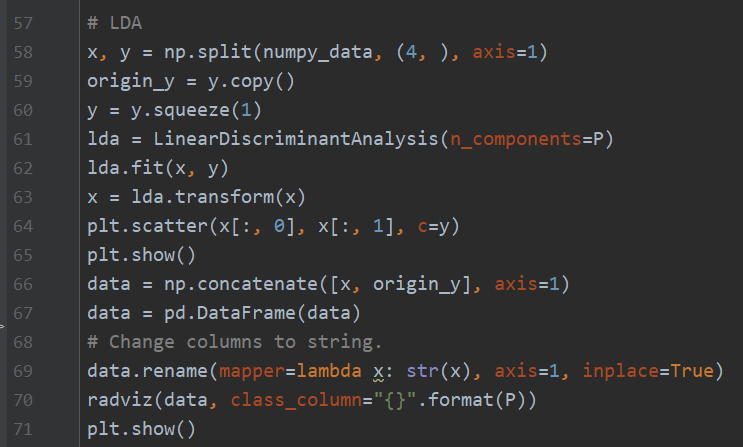
数据预处理：



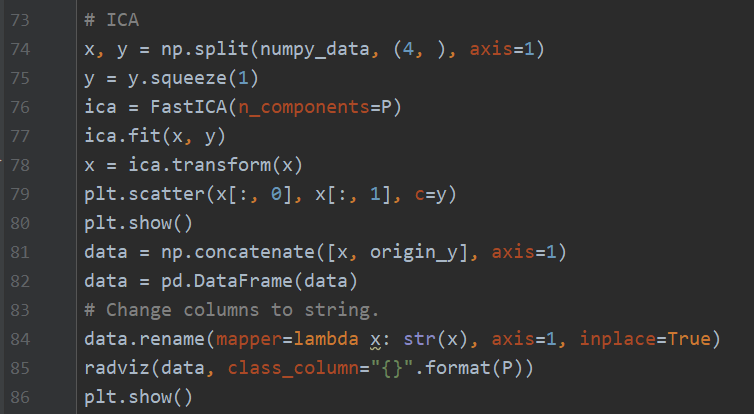
PCA：



LDA：

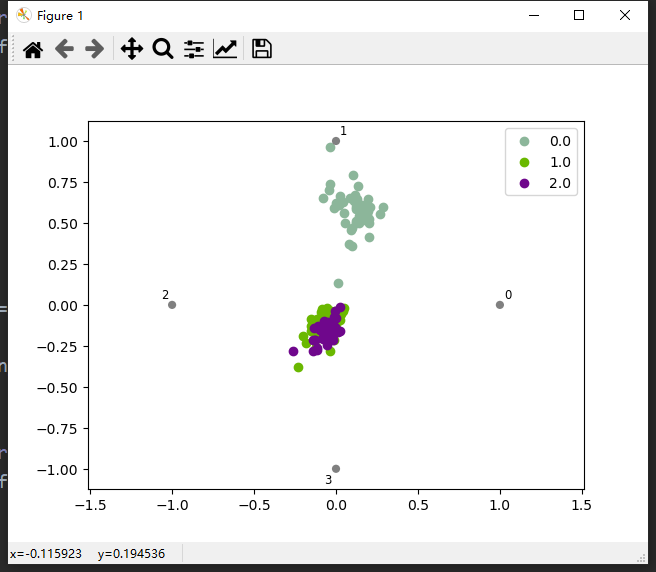


ICA：

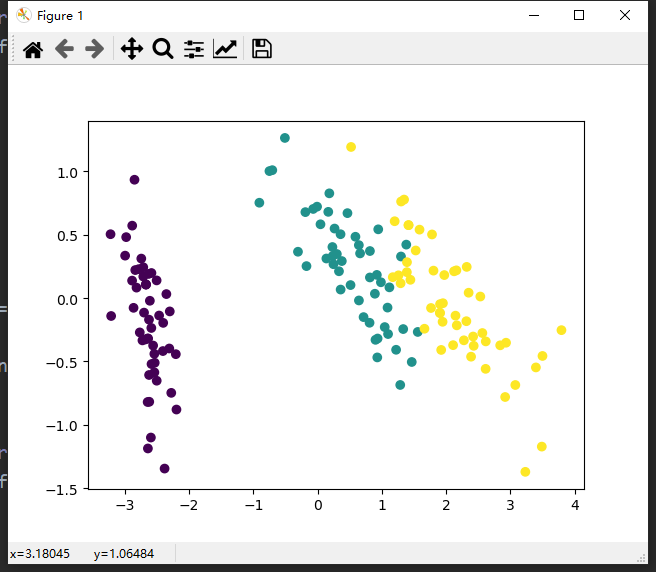


### 实验结果

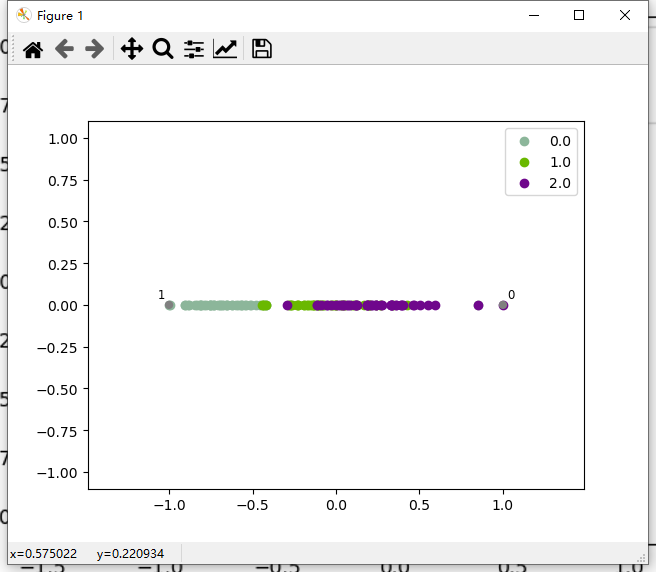
原数据雷达图：



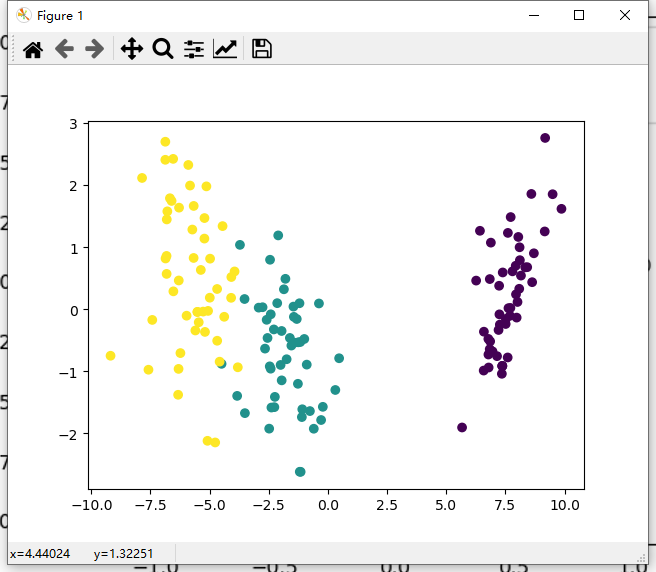
PCA散点图：



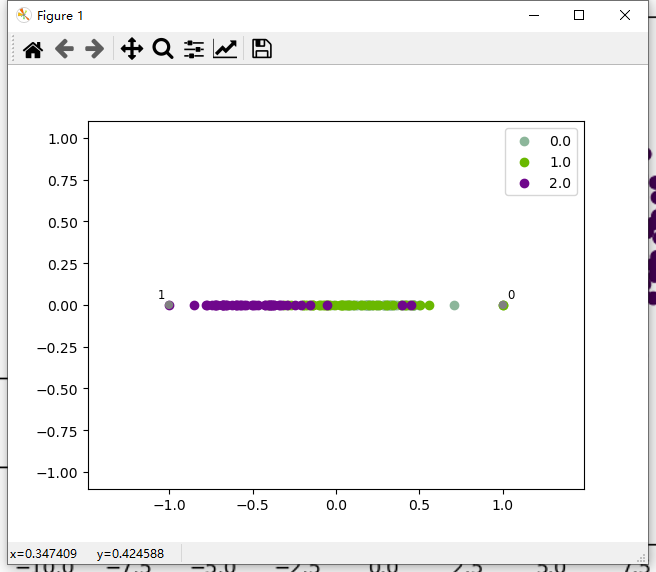
雷达图：



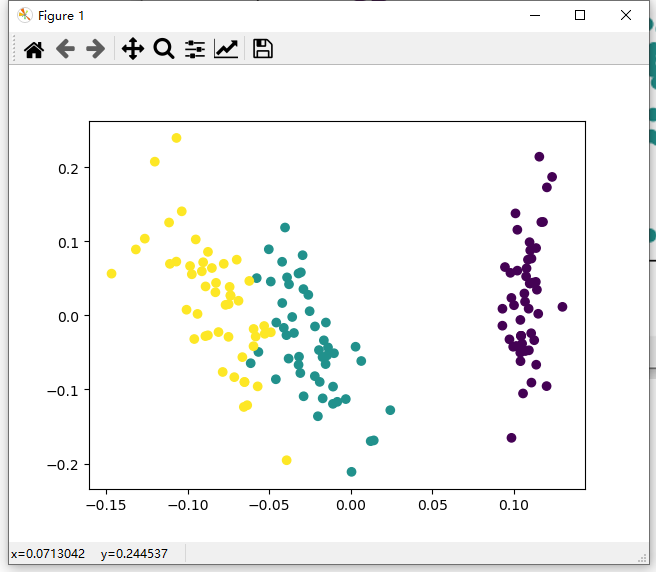
LDA散点图：



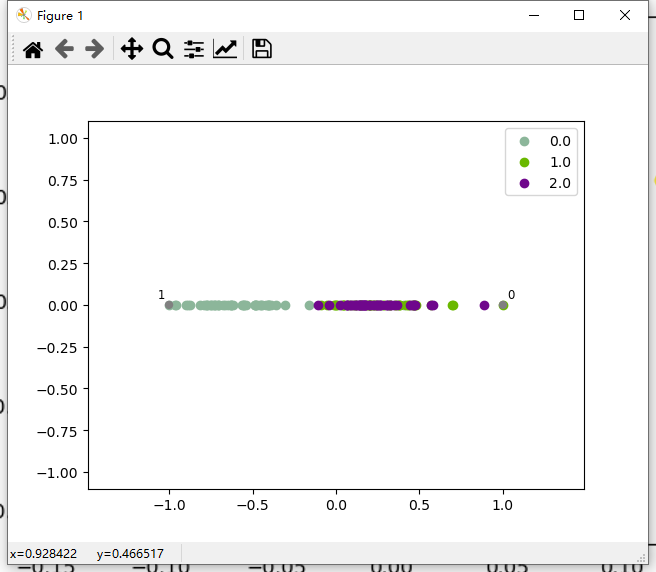
雷达图：



ICA散点图：



雷达图：



结论：

比较效果其实相近。

### 实验感想

学习了可视化方法，对于可视化工具有了新的认识。