**华中科技大学计算机科学与技术学院**

**机器学习报告**



专 业： 计算机科学与技术

班 级： 计算机1807班

学 号： U201814738

姓 名： 黄宗耀

成 绩：

指导教师： 邹复好

**完成日期： 2020年 6 月 30 日**

## 一、实验题目：个人收入预测

## 二、实验要求

**（1）数据集**

给定训练集income.csv，要求根据每个人的属性值来判断此人年收入是否大于50K。

**训练集介绍：**

（1）CSV文件，大小为4000行×59列;

（2）4000行数据对应着4000个人，ID编号从1到4000;

（3）59列数据中，第一列为ID，最后一列label(1或0)表示年收入是否大于50K，中间的57列为57种属性值。

**（2）任务描述**

（1）将数据中前3000项作为训练集，后1000项作为测试集，使用logistic回归进行二分类，实现语言要求为Python；

（2）在使用梯度下降法时，调整学习率的固定值，有能力的同学可以学习并使用动态调整学习率的方法，探究不同学习率的选择对训练误差收敛速度的影响，绘制misclassification rate曲线进行比较并分析。

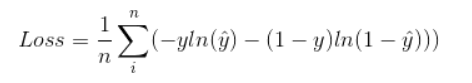
**（3）评测标准**

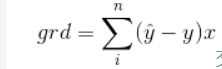
1. 要求计算出准确率。
2. 要求画出训练和测试loss曲线。
3. 要求调整多个学习率和正则化参数后给出上面的结果。

## 三、算法设计

算法1：数据处理，用csv库中的reader读取数据，将数据进行划分后，直接调用sklearn中的linear\_model库，直接进行逻辑回归预测收入。

算法2：数据处理，用csv库中的reader读取数据，将数据进行划分后，通过分步计算的函数，运用mini-batch算法。一步步计算出loss和准确率，绘制不同学习率和正则化参数的曲线。其中



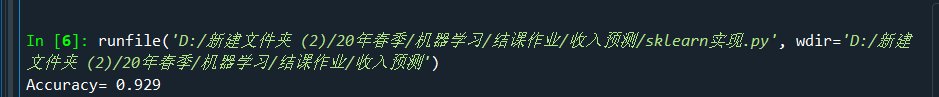


## 四、实验环境与平台

实验环境：Windows10，Spyder，语言 Python语言

## 五、实验结果

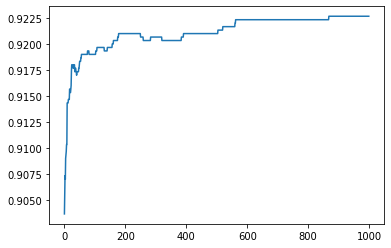
算法1：sklearn直接执行，计算出准确率：

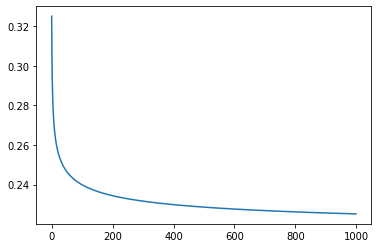
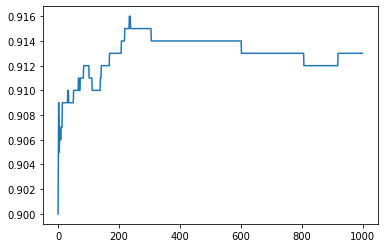


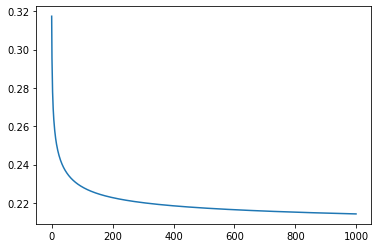
用时为2s。准确率为0.929。

算法2：通过单步执行的算法（mini-batch）来实现：

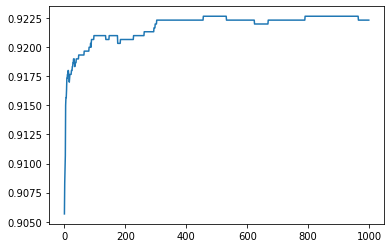
1.正则化系数，每次执行的样本数，学习率分别为0.001 ，20 ，0.2所得的结果如下：（图片纵坐标顺序从上到下为训练准确率、测试准确率、训练loss、测试loss，横坐标为迭代次数）

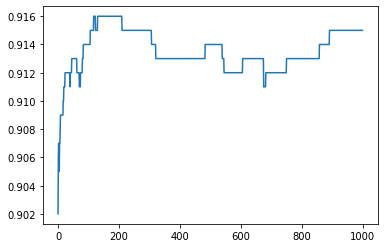


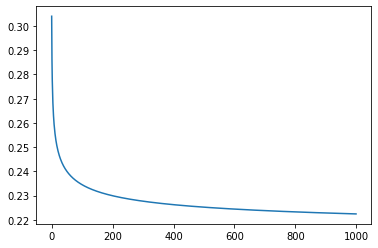


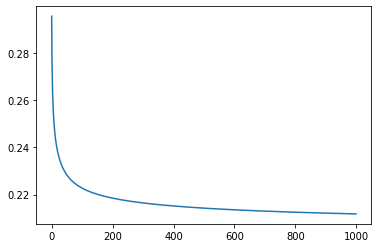
.

2.正则化系数，每次执行的样本数，学习率分别为0.001 ，10，0.2所得的结果如下：（图片纵坐标顺序从上到下为训练准确率、测试准确率、训练loss、测试loss，横坐标为迭代次数）

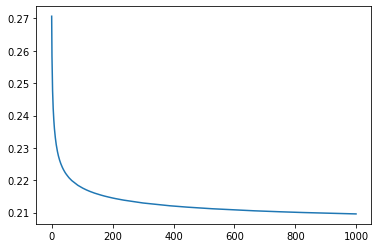
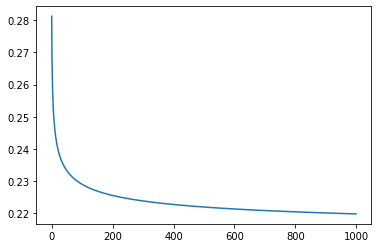
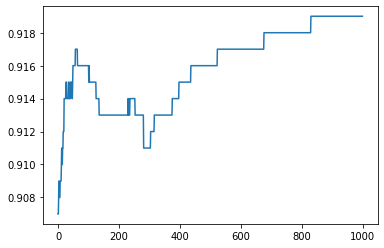
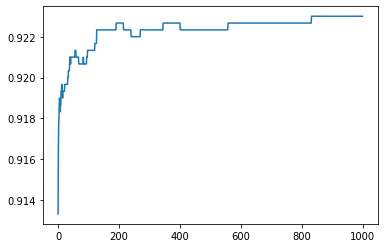




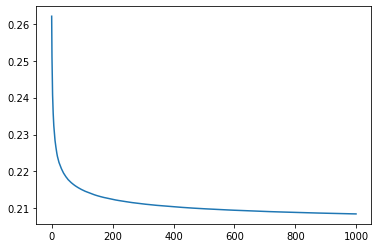
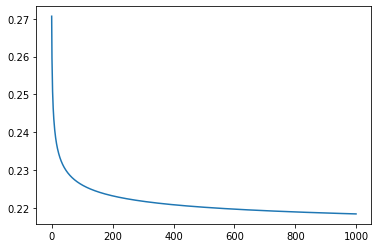
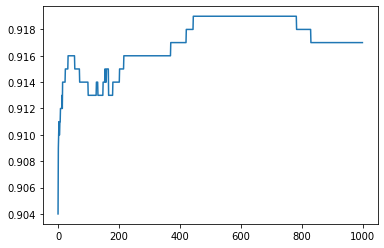
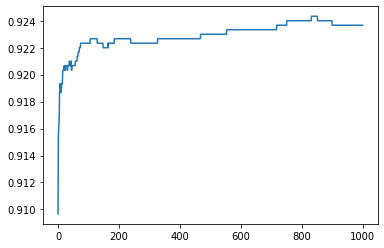




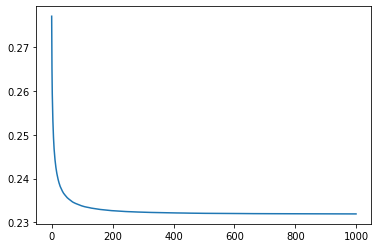
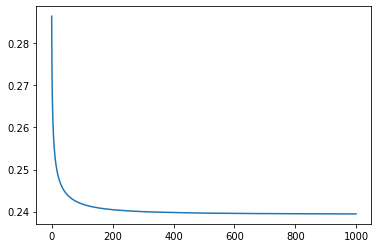
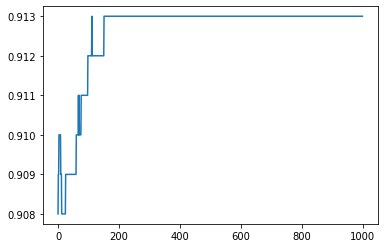
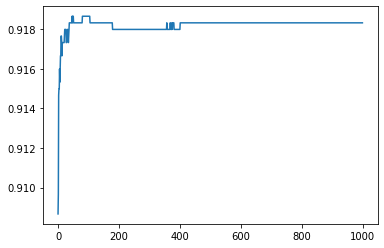
3.正则化系数，每次执行的样本数，学习率分别为0.001 ，10 ，0.3所得的结果如下：（图片纵坐标顺序从上到下为训练准确率、测试准确率、训练loss、测试loss，横坐标为迭代次数）



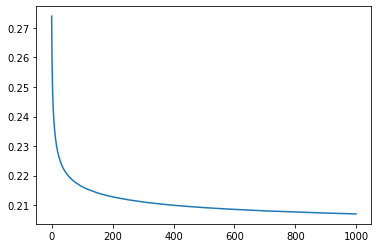
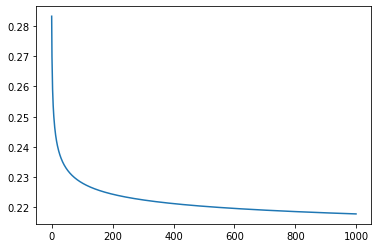
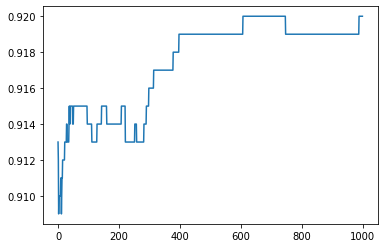
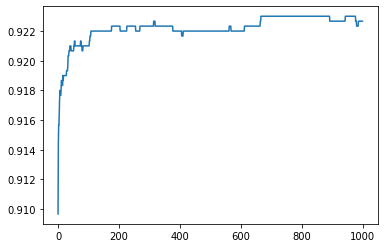
4.正则化系数，每次执行的样本数，学习率分别为0.001 ，10 ，0.4所得的结果如下：（图片纵坐标顺序从上到下为训练准确率、测试准确率、训练loss、测试loss，横坐标为迭代次数）



5.正则化系数，每次执行的样本数，学习率分别为0.01 ，10 ，0.3所得的结果如下：（图片纵坐标顺序从上到下为训练准确率、测试准确率、训练loss、测试loss，横坐标为迭代次数）



6.正则化系数，每次执行的样本数，学习率分别为0.0001 ，10 ，0.3所得的结果如下：（图片纵坐标顺序从上到下为训练准确率、测试准确率、训练loss、测试loss，横坐标为迭代次数）



## 六、结果分析

第二次相比第一次，每次选取的样本数降低。对比第一次实验，明显发现，当每次选取的样本数为10时，loss收敛的更低，为0.22，后续每次选取10个样本进行学习率的对比。

第三次相比第二次，学习率调高为0.3，可以明显看到，第一张图中准确率的变化变得更加明显，这也是学习率提高的改变。而loss相比于0.2的学习率有较小的降低。

第四次相比前两张图，学习率提高为0.4，可以看出，随着学习率的提升，准确率和loss都有着略微的改进，但是影响并不大。

所以后续选择0.3的学习率来进行正则化系数的测试。

第五次相比第三次，正则化系数增大。可以看出，正则化系数提升，对于准确率和loss有着明显的负面影响，后续继续将正则化系数减小进行测试。

第六次相比第三次，正则化系数减小。可以看出，正则化系数在0.0001跟0.001时差距并不大。后续测试发现还是0.001效果最好。

## 七、总结

学习率高一些，准确率可以有部分提高，但是不明显，反而使曲线变得变化很大，可以看出学习率应该在0.3左右比较合适，曲线比较平滑。

正则化系数也是比较小一些更好，经过测试发现，正则化系数在0.001时就比较合适。

与sklearn算法比较：

Sklearn算法非常简单，只需要调用库里面的函数即可，并且准确率达到了0.929，速度也是很快，只需要几秒钟就能够完成分类。

与sklearn比较，首先速度是差了很多的，自己的函数执行一千次迭代需要一分钟左右的时间，而sklearn计算出结果也不过几秒钟的时间。然后在准确率方面，迭代的算法大约在0.922左右，而sklearn也是达到了0.929。所以，虽然通过了很多尝试（不同的正则化、样本数、学习率），准确率得到了提升，但是速度方面还是跟官方算法有很大差距，还需要改进。

收获：

首先是数据的读取，一开始我完全不明白怎么样读取数据，在经过了多方查找资料后，我明白了，通过csv库的reader就可以读取数据，然后还需要把字符数据转化成浮点型才能进行具体操作。不仅如此，图片的绘制一开始我也是一头雾水，但是通过查找资料后，我知道了通过matplotlib库就可以很轻松的绘图了。总的来说，这次实验收获是很大的，加深了我对逻辑回归算法的理解和掌握，明白了不同学习率、正则化系数对算法效率、准确率以及loss的影响。另外，也知道了归一化也是一大影响因素。这次实验还是收获颇丰的。

## 参考文献

[1] Python 机器学习经典实例

https://blog.csdn.net/valada/java/article/details/79910110 [2] [Python机器学习笔记：sklearn库的学习](https://www.cnblogs.com/wj-1314/p/10179741.html)

https://www.cnblogs.com/wj-1314/p/10179741.html

[3] 李宏毅深度学习HW2 收入预测 (logistic regression)

https://blog.csdn.net/null\_zhao/java/article/details/95238412