# Machine Learning Homework 1.2

专业：软件工程 姓名：沈金龙 学号：18214806

1. **实验题目**

>Generate n = 2,000 points uniformly at random in the two-dimensional unit square. Which point do you expect the centroid to be?

>What objective does the centroid of the points optimize?

>Apply gradient descent (GD) to find the centroid.

>Apply stochastic gradient descent (SGD) to find the centroid. Can you say in simple words, what the algorithm is doing?

注：In mathematics and physics, the **centroid** or geometric center of a plane figure is the arithmetic mean position of all the points in the figure. Informally, it is the point at which a cutout of the shape could be perfectly balanced on the tip of a pin.

1. **实验要求**

1）编程语言不限。

2）作业包含一份报告（word 或pdf格式）及代码加注释，并打包到.zip，其中zip文件的命名格式为学号\_姓名。

3）不允许使用梯度下降相关的库函数。

4） 禁止抄袭。

1. **实验过程及代码**

1）本实验采用python3.6完成，首先使用rand生成2000\*2大小的随机数矩阵。由于质心的定义是所有点的算术平均位置，以此计算出质心的坐标，并在二维单元图中进行标示。

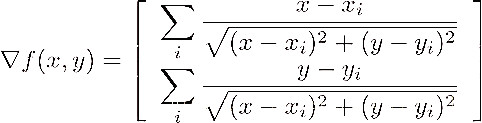
2）点的质心优化的目标是：求质心点到其余点的最短的距离和，即找到一个点C（x，y）使得f（x，y）最小，那么本题的优化目标就是min f（x，y），此f（x，y）即为损失函数cost，其表示如下：

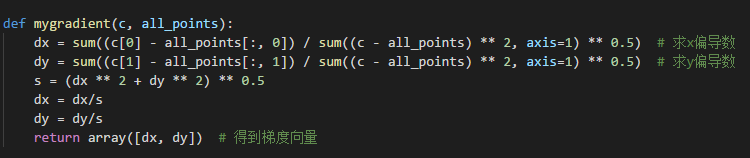
3）使用梯度下降算法来计算质心：

>定义损失函数，优化目标就是最小化损失函数。C表示待求点centroid，all\_points表示生成所有的点。下面的返回值就是表示该点到所有点距离和。

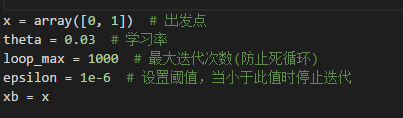


>求梯度向量，根据损失函数求出x,y的偏导数，再求出梯度向量array([dx, dy])





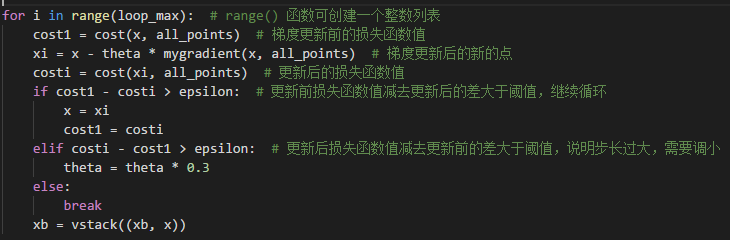
>设置参数值。设置出发点从左上角（0，1）开始，初始化学习率，设置最大迭代次数和阈值。



>开始循环。

cost1：表示梯度更新前的损失函数值，costi： 表示梯度更新后的损失函数值。

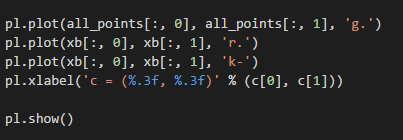
用更新前的损失函数值cost1减去更新后的costi，差值大于阈值说明还没收敛，继续循环。当costi-cost1>阈值，说明步长过长，已经越过最佳值，需要缩小学习率，一直到收敛。



>画图：

‘g.’表示用绿色的点表示2000个随机点

‘r.’表示用红色的点表示每一次迭代过程收敛的点

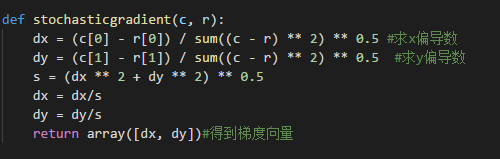


>最终结果：centroid = c = （0.490，0.51）

4）使用随机梯度下降算法来计算质心：

随机梯度下降和梯度下降的区别就是，梯度下降是所有点都参与梯度更新，而随机梯度下降是每次循环随机选取点进行。

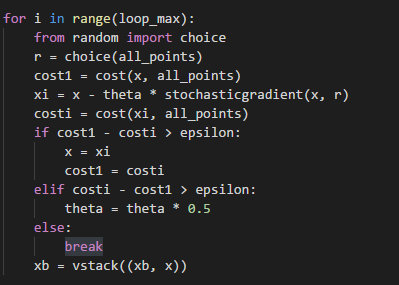
在梯度下降的基础上进行修改：



此处 r 表示在所有点中随机选取的一个点，每次迭代用随机抽取点的方式。

from random import choice

r = choice(all\_points)

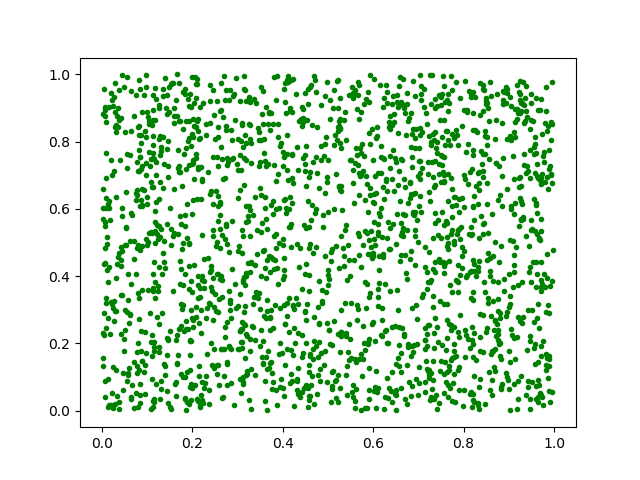


可以看到，由于是随机选取的点，所以在找最优值的过程是曲折的。

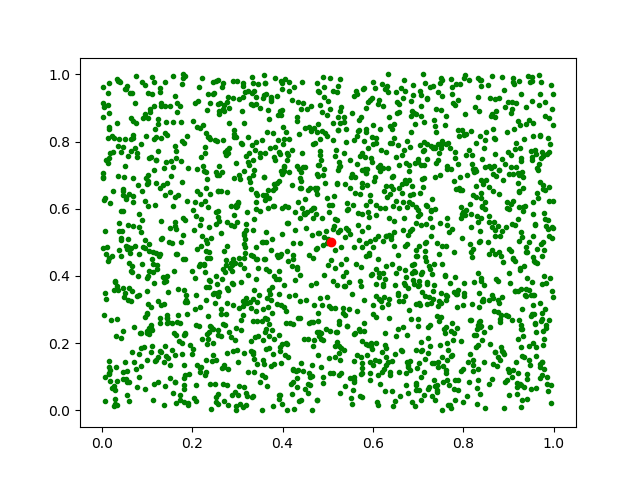
最终收敛：centroid = c = （0.476，0.532）

1. **实验结果与分析**

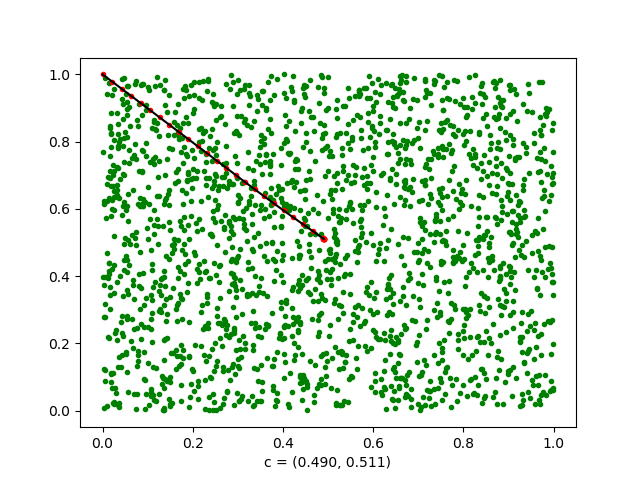
1）在二维单元图中生成2000个点，如下图所示：



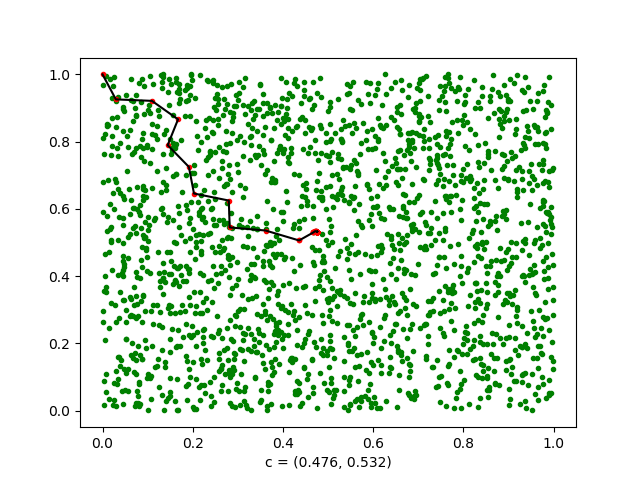
2）在二维单元图中生成2000个点，并按照算术均值的方式求出其质心，如下图所示：



3）按照梯度下降算法求得质心，并绘出寻找路线，如下图所示：



4）按照随机梯度下降算法求得质心，并绘出寻找路线，如下图所示：



1. **总结**

本实验回顾了梯度下降算法以及随机梯度下降算法，并使用python语言进行了实现，从图示中体验两种算法的异同点，同时加深了对损失函数、学习率等相关概念的理解。