现代人工智能 Homework1.2

题目描述：

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

2.What objective does the centroid of the points optimize?

3.Apply gradient descent to find the centroid.

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

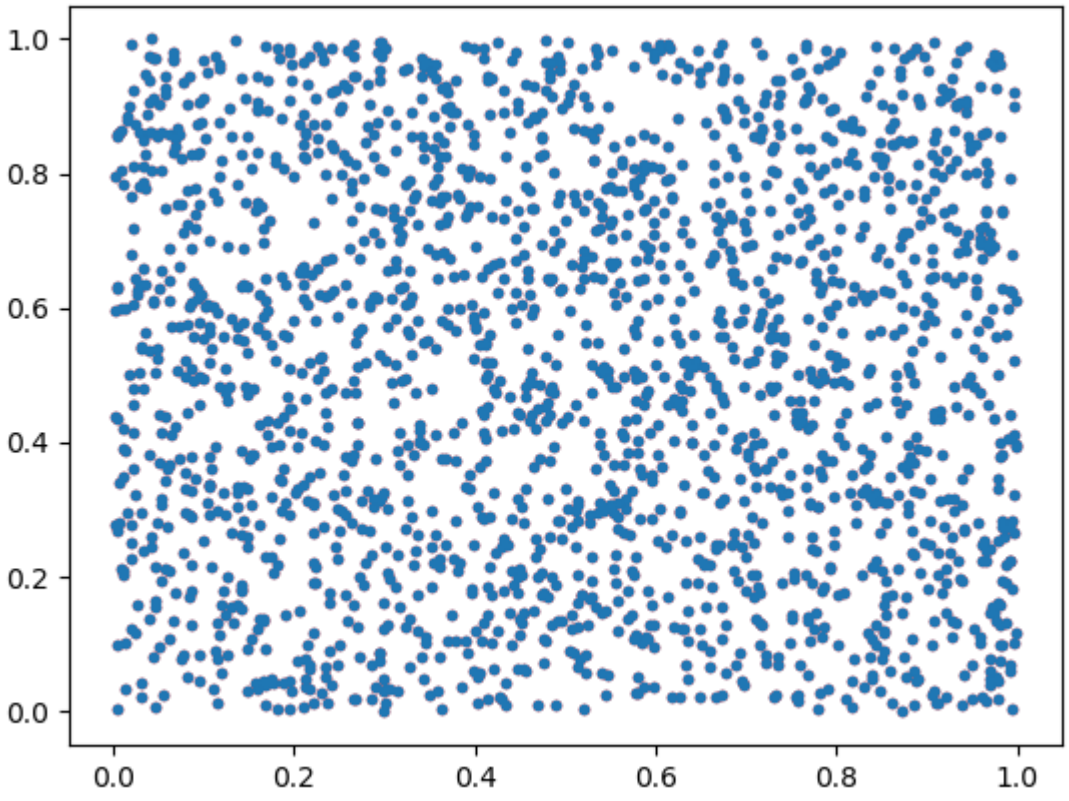
1、所谓质心，就是找到到所有点距离之和最小的点。由于点的生成都是随机的，所以期望的点是：(0.5，0.5)。

生成的散点图代码和图如下：

```
#!/usr/bin/env python
# coding=utf-8
from scipy import *
import matplotlib
import matplotlib.pyplot as plt

points = rand(2000,2)
print(points)

#画出散点图
plt.figure(1)
plt.plot(points[:,0],points[:,1],'.')
plt.show()
```



2.已知质心到其他点的距离和是最短的，即找到这样一个点 C(x,y) 使得 f(x,y)最小即可。所以目标函数可以表示成计算质心到其他点的欧式距离开方，优化目标函数就是 min f(x,y)，目标函数如下：

$$f(x,y) = \sum_{i=0}^{2000} \sqrt{(x-x_i)^2 + (y-y_i)^2}$$

```
def cost(x):
    cost_ = sum(sum((x-points)**2,axis = 1)**0.5)
    return cost_
```

3.最关键的地方就是求梯度，求梯度的代码如下：

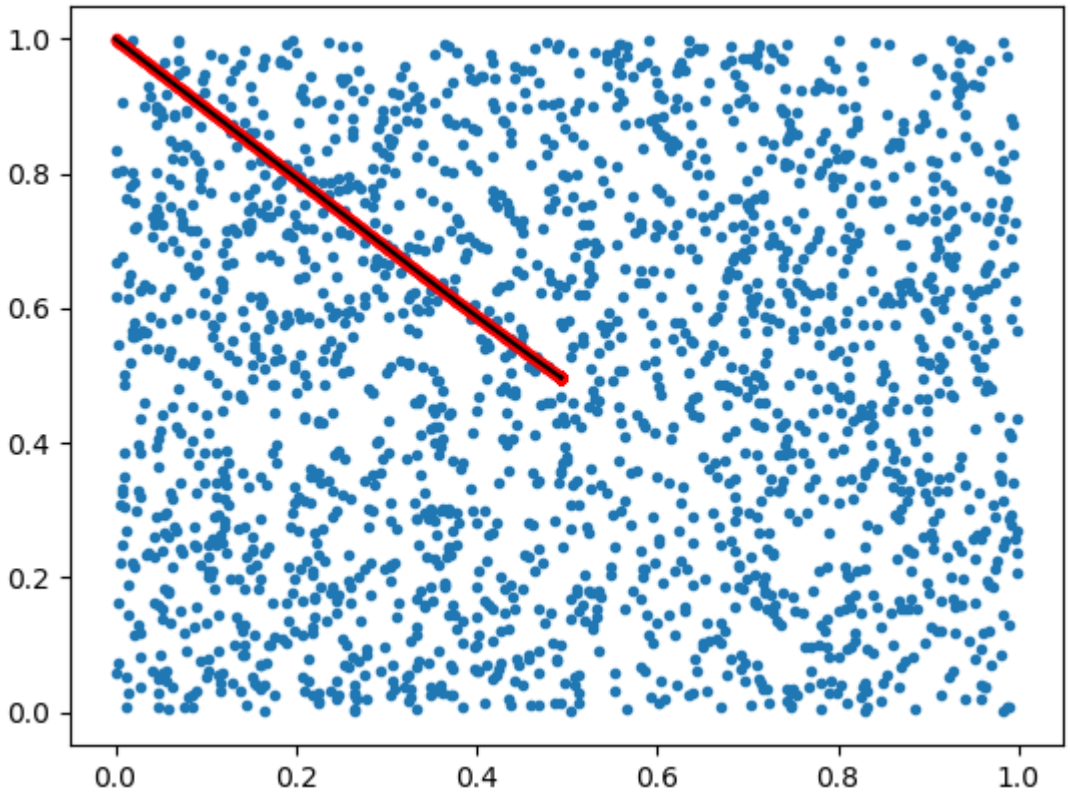
```
#求梯度
def get_gredient(x):
    dx_ = sum((x[0] - points[:,0]) / sum((x - points)**2,axis=1)**0.5)
    dy_ = sum((x[1] - points[:,1]) / sum((x - points)**2,axis=1)**0.5)
    s = (dx_**2+dy_**2)**0.5
    dx = dx_/s
    dy = dy_/s
    return array([dx,dy])
```

设置参数循环梯度下降寻找质点

```
x = array([0,1])
theta = 0.01
max = 5000
#thread = 1e-6
xb = x
for i in range(max):
    cost_ = cost(x)
    x = x - theta*get_gredient(x)
    print(cost_)
    xb = vstack((xb,x))
c = x
print(c)
```

求出的质点结果为：

```
759.7797321421571
759.7072337278952
759.7797321421571
759.7072337278952
759.7797321421571
759.7072337278952
[0.4988894 0.5086899]
```



4.

随机梯度下降和梯度下降的区别：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。这样的话可以解决一次性加载数据量太大内存不够的问题。

跟普通梯度下降不同的是，随机梯度下降采用了随机选点，代码如下。

```
def get_stochastic_gredient(x):
    points_ = choice(points)
    dx_ = (x[0] - points_[0]) / sum((x - points_)**2)**0.5
    dy_ = (x[1] - points_[1]) / sum((x - points_)**2)**0.5
    s = (dx_**2+dy_**2)**0.5
    dx = dx_/s
    dy = dy_/s
    return array([dx,dy])
```

结果如下：

```
774.9549136271424
774.9371444747956
774.9549228098632
774.9375332515817
774.965474745768
[0.49476938 0.50623188]
```

可视化结果如下：