**实 验 报 告**

**名称：­ 机器学习2021（春）**

**实验项目名称： 最小解发现**

**报告人： 摸鱼小队-Snakebin**

**实验时间： 2021/8/16**

**实验报告提交时间： 2021/8/17**

|  |  |
| --- | --- |
| 实验目的  1. 理解等高线的几何含义、如何发现一个函数的最小解； 2. 掌握一门绘制函数图形的编程工具； | |
| 实验环境  1. Python或Java或Matlab或Mathematica 2. 如用Python，绘图工具可采用matplotlib或者plotly等。 | |
| 实验内容 给定下述Rosenbrock函数，，其中，。试编写程序完成下述工作：   1. 为不同的a,b取值，绘制该函数的3D表面。请问 a,b取值对该表面形状有大的影响吗？，所谓大影响就是形状不再相似。对a,b的取值区间，能否大致给出一个分类，像下面这样给出一张表：  |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | ① |  |  |  | |  | ② | |  |  |  1. 编写一个算法来找到它的全局最小值及相应的最小解，并在3D图中标出。分析一下你的算法时空效率、给出运行时间。   显然由上图表可知只有当图形为①②时有全局最小值  代码如下 |

|  |
| --- |
| import matplotlib.pyplot as plt  from mpl\_toolkits.mplot3d import Axes3D  import numpy as np    def Fun(x,y,a=1,b=1):#原函数      return (a-x)\*\*2 + b\*(y-x\*\*2)\*\*2    def PxFun(x,y,a=1,b=1):#偏x导      return (2\*b+2)\*x-2\*b\*y\*\*2-2\*a    def PyFun(x,y,a=1,b=1):#偏y导      return 4\*b\*y\*\*3-4\*b\*x\*y    #初始化  a = float(input('请输入a:'))  b = float(input("请输入b:"))  fig=plt.figure()#figure对象  ax=Axes3D(fig)#Axes3D对象  X,Y=np.mgrid[-2:2:100j,-2:2:100j]#取样并作满射联合  Z=Fun(X,Y,a,b)#取样点Z坐标打表  ax.plot\_surface(X,Y,Z)  ax.set\_xlabel('x')  ax.set\_ylabel('y')  ax.set\_zlabel('z')    #梯度下降  step=0.0008#下降系数  x=-2  y=-2#初始选取一个点  tag\_x=[x]  tag\_y=[y]  tag\_z=[Fun(x,y,a,b)]#三个坐标分别打入表中，该表用于绘制点  new\_x=x  new\_y=y  Over=False  while Over==False:      new\_x-=step\*PxFun(x,y,a,b)      new\_y-=step\*PyFun(x,y,a,b)#分别作梯度下降      if Fun(x,y,a,b)-Fun(new\_x,new\_y,a,b)<7e-9:#精度          Over=True      x=new\_x      y=new\_y#更新旧点      tag\_x.append(x)      tag\_y.append(y)      tag\_z.append(Fun(x,y,a,b))#新点三个坐标打入表中    #绘制点/输出坐标  ax.plot(tag\_x,tag\_y,tag\_z,'ro')  plt.title(f'(x,y)~({round(x,2)},{round(y,2)})')  plt.show() 以a=1,b=1为例，成功求出最小值点 |