代码部分：

import numpy as np

#计算λ（学习率）

def compute\_lr(x1,x2):

    a = (16\*x1\*x1 + 4\*x2\*x2) / (64\*x1\*x1 + 8\*x2\*x2)

    return a

#计算梯度

def grad(x1,x2):

    return 4\*x1,2\*x2

if \_\_name\_\_ == "\_\_main\_\_":

    x = [1,1]

    x = np.array(x)

    for i in range(10):

        a,b = grad(x[0], x[1])

        gradient = np.array([a,b])

        lr = compute\_lr(x[0],x[1])

        x = x - lr\*gradient

        print("x%d = "%(i+1),x)

        print("第%d梯度为: "%(i+1),gradient)

输出：

x1 = [-0.11111111 0.44444444]

第1梯度为: [4 2]

x2 = [0.07407407 0.07407407]

第2梯度为: [-0.44444444 0.88888889]

x3 = [-0.00823045 0.03292181]

第3梯度为: [0.2962963 0.14814815]

x4 = [0.00548697 0.00548697]

第4梯度为: [-0.03292181 0.06584362]

x5 = [-0.00060966 0.00243865]

第5梯度为: [0.02194787 0.01097394]

x6 = [0.00040644 0.00040644]

第6梯度为: [-0.00243865 0.00487731]

x7 = [-4.51602342e-05 1.80640937e-04]

第7梯度为: [0.00162577 0.00081288]

x8 = [3.01068228e-05 3.01068228e-05]

第8梯度为: [-0.00018064 0.00036128]

x9 = [-3.34520253e-06 1.33808101e-05]

第9梯度为: [1.20427291e-04 6.02136455e-05]

x10 = [2.23013502e-06 2.23013502e-06]

第10梯度为: [-1.33808101e-05 2.67616202e-05]