手机屏幕的截图

描述已自动生成

**说明：各个数字的输入顺序为**

-1:346

1:2,5

2:13457

3:13467

4:2346

5:12467

6:124567

7:1236

8:1234567

9:123467

如要显示1这个数字需要二号和五号灯亮，若用1表示led亮，0表示不亮所以输入为0，1,0，0，1,0,0以此类推

输出为四位二进制数字。

-1=000-1 （这个不知道表示对不对）

1=0001

2=0010

3=0011

4=0100

5=0101

6=0110

7=0111

8=1000

9=1001

代码如下：

1. **import** random
2. **import** torch
4. #10个输入,用七位数字代表一个楼层号码
5. X = torch.tensor([[[0.0,0.0,1.0,1.0,0.0,1.0,0.0]],        #这个代表-1楼以此类推
6. [[0.0,1.0,0.0,0.0,1.0,0.0,0.0]],
7. [[1.0,0.0,1.0,1.0,1.0,0.0,1.0]],
8. [[1.0,0.0,1.0,1.0,0.0,1.0,1.0]],
9. [[0.0,1.0,1.0,1.0,0.0,1.0,0.0]],
10. [[1.0,1.0,0.0,1.0,0.0,1.0,1.0]],
11. [[1.0,1.0,0.0,1.0,1.0,1.0,1.0]],
12. [[1.0,1.0,1.0,0.0,0.0,1.0,0.0]],
13. [[1.0,1.0,1.0,1.0,1.0,1.0,1.0]],
14. [[1.0,1.0,1.0,1.0,0.0,1.0,1.0]]])
15. #以下是期望输出值，用二进制数表示
16. S = torch.tensor([[[0.0,0.0,0.0,0.0]],
17. [[0.0,0.0,0.0,1.0]],
18. [[0.0,0.0,1.0,0.0]],
19. [[0.0,0.0,1.0,1.0]],
20. [[0.0,1.0,0.0,0.0]],
21. [[0.0,1.0,0.0,1.0]],
22. [[0.0,1.0,1.0,0.0]],
23. [[0.0,1.0,1.0,1.0]],
24. [[1.0,0.0,0.0,0.0]],
25. [[1.0,0.0,0.0,1.0]]])
27. lr = 0.01
28. W = torch.rand((7,7),requires\_grad=True)   #7：输入数量，第二个7：中间节点的数量
29. V = torch.rand((7,4),requires\_grad=True)   #7：输入数量，9：输出数量
30. epoch = 1000000       #循环遍历一百万次
31. **for** j **in** range(0,epoch):
32. i = random.randint(0,9)     #从0-8索引
34. Y = torch.sigmoid(torch.mm(X[i],W))   #mm是矩阵乘以矩阵 ，m=matrix
35. Z = torch.sigmoid(torch.mm(Y,V))
36. e = (S[i]-Z)\*(S[i]-Z)
37. e = e[0][0]\*4+e[0][1]\*2+e[0][2]\*1 + e[0][3] \* 1.6 #确保误差为标量
38. **if** j%10000==0 **and** i!=0 **or** j>epoch-10:
39. **print**(f'This is the {i}th floor,now error is {e:.3f},the expected output is:{S[i]},the actual output is:{Z}.')
40. **elif** j%10000==0 **and** i==0 **or** j>epoch-10:
41. **print**(f'This is the {i-1}th floor,now error is {e:.3f},the expected output is:{S[i]},the actual output is:{Z}.')
42. e.backward()                             #反向传播
43. W.data = W.data-lr\*W.grad                #更新权重
44. V.data = V.data-lr\*V.grad
45. W.grad.zero\_()                           #梯度清零
46. V.grad.zero\_()

运行结果如下：

文本

中度可信度描述已自动生成