**实验三 神经网络**

**一、实验目的**

理解反向传播网络的结构和原理，掌握反向传播算法对神经元的训练过程，了解反向传播公式。通过构建BP网络实例，熟悉前馈网络的原理及结构。

**二、实验内容**

1. 通过BP网络各项参数的不同设置，观察BP算法的学习效果。观察比较BP网络 拓朴结构及其它各项参数变化对于训练结果的影响。观察并分析不同训练数据对相同拓朴结构的BP网络建模的影响。
2. 设计简单的感知器，实现简单的逻辑运算（与、或）等，也可做其他更复杂的问题。

**三、实验记录**

**神经网络实验报告表三**

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 |  |  |  |
| 实验目的 | 理解反向传播网络的结构和原理，掌握反向传播算法对神经元的训练过程，了解反向传播公式。通过构建BP网络实例，熟悉前馈网络的原理及结构。 | | |
| 网络  拓朴图 |  |  |  |
| 训练  数据集 | 输入x1~3,输出y  (x1,x2,x3,y)  (0,0,0, 0.0)  (0,0,1, 0.0)  (0,1,0, 0.0)  (1,0,0, 0.0)  (1,1,0, 1.0)  (1,0,1, 1.0)  (0,1,1, 1.0)  (1,1,1, 1.0) | (输入x1,输入x2,输出y)  (0,0,0)  (0,1,1)  (1,1,0)  (1,0,1) | 输入x1,x2，输出与值，或值，非值（x1,x2,与，或，非）  (0,0,0,0,1)  (0,1,0,1,0)  (1,0,0,1,0)  (1,1,1,1,0) |
| 训练误差 | 181 steps 0.02 | 400 steps 0.03 | 250 steps 0.1 |
| 模拟的问题或函数 | 多数赞成表决器 | 异或问题 | 布尔网 |
| 观测结果 | 选择默认参数（已训练），再训练181步后，达到了误差0.0 | 此问题随机生成了参数，经过266步训练后误差达到0.02 | 选择默认参数（未训练），经过250步后误差达到0.099 |
| 学生结论 | 经过多次训练反馈修改权值，最终神经网络能够完成多数赞成表决任务 | 该异或网络存在问题，不可正确求解。因为经过大量学习后误差已经达到目标值，但是输入（1，1）仍然不能达到正确结果（预期是0，实际得到了1） | 经过多次训练反馈修改权值，最终神经网络能够完成布尔网求解任务。此外，初始权值的设定会影响训练的过程和结果，不合适的权值会导致难以训练出结果。 |

**设计感知器求解三个数相乘的结果的奇偶性（正反感知器）**

根据作业中设计到的问题，我们知道此问题求解需要4个输入，1个输入固定为1，初始权值取值(0.3, 0.2 0.5, -0.3)

源码如下：

# 训练一个感知器用以判断输入的三个整数之积是奇数还是偶数

# 需要四个输入端， 取初权值向量为(0.3, 0.2 0.5, -0.3)

# 输入为(1,x,y,z)

#

def is\_odd\_or\_even(x):

    if(x%2==1):

        return -1

    return 1

w0 = [0.3, 0.2, 0.5, -0.3]

new\_w = [0,0,0,0]

def run(w0):

    for i in range(0,10):

        print("No."+ str(i))

        print("w="+ str(w0))

        x = input('input\_x:')

        y = input('input\_y:')

        z = input('input\_z:')

        x = is\_odd\_or\_even(int(x))

        y = is\_odd\_or\_even(int(y))

        z = is\_odd\_or\_even(int(z))

        X = [1,x,y,z]

        # 期望输出

        if(x==-1 and y==-1 and z==-1):

            t = -1

        else:

            t = 1

        # 实际输出

        sum = 0

        for i in range(0,4):

            sum = sum + w0[i]\*X[i]

        if(sum<0):

            o = -1

        else:

            o = 1

        # 更新w0

        for i in range(0,4):

            new\_w[i] = w0[i] + 1\*(t-o)\*X[i]

        w0 = new\_w

    while True:

        print("Now\_w = ",str(w0))

        #检验

        print('测试如下：')

        x = input('input\_x:')

        y = input('input\_y:')

        z = input('input\_z:')

        result = int(x)\*int(y)\*int(z)

        x = is\_odd\_or\_even(int(x))

        y = is\_odd\_or\_even(int(y))

        z = is\_odd\_or\_even(int(z))

        X = [1,x,y,z]

        sum = 0

        for i in range(0,4):

            sum = sum + w0[i]\*X[i]

        if(sum>0):

            print(str(result)+"是偶数")

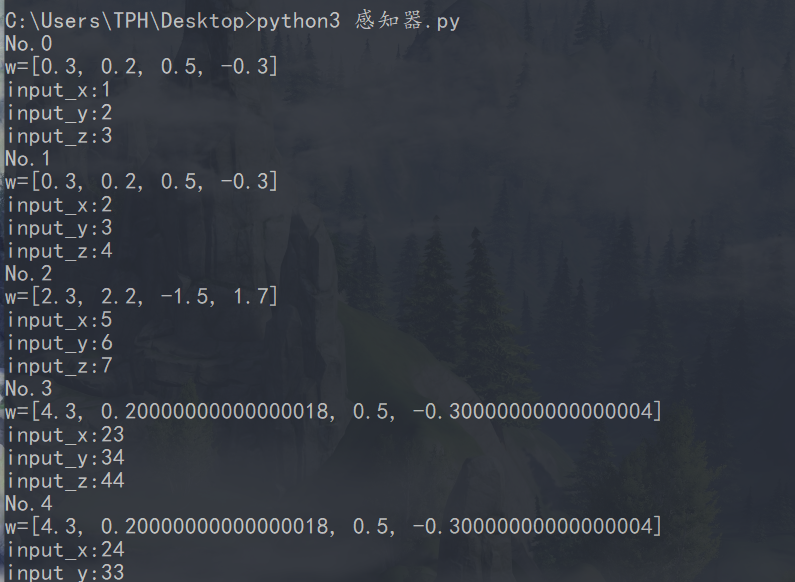
        else:

            print(str(result)+"是奇数")

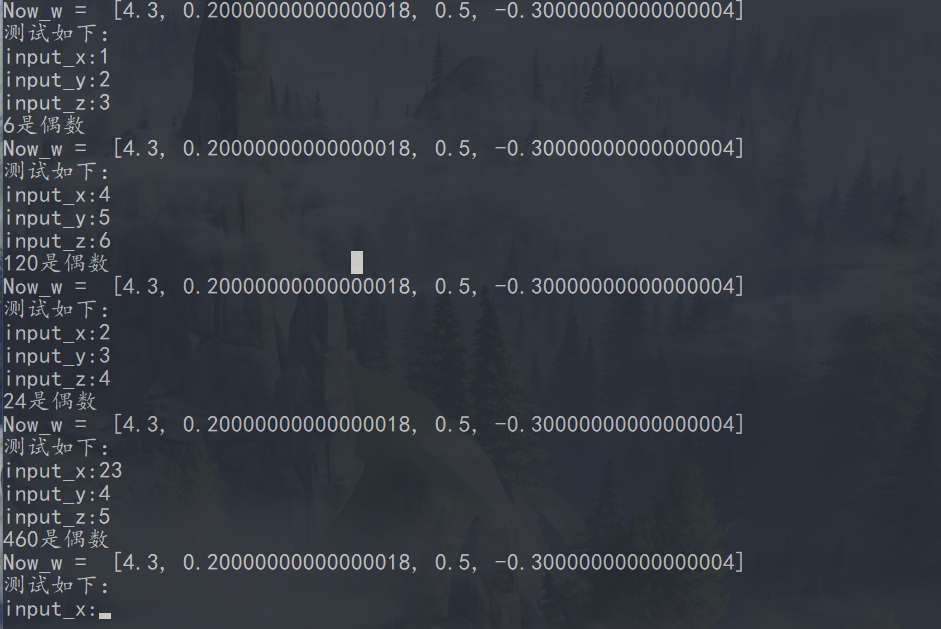
if \_\_name\_\_ == "\_\_main\_\_":

    run(w0)

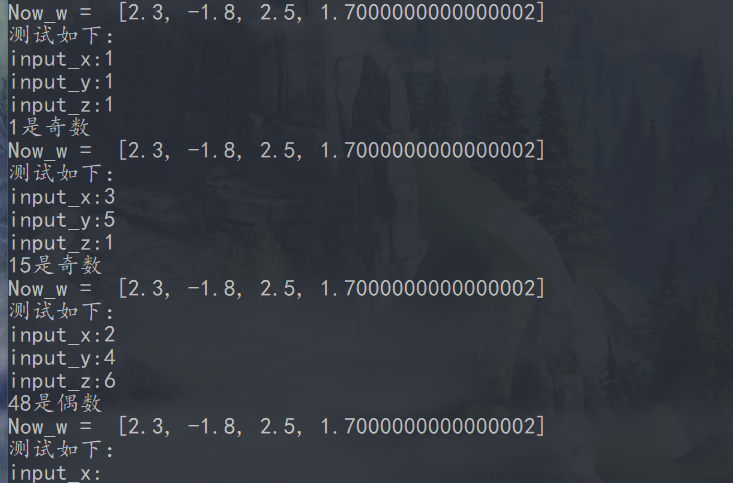
该程序只是简单实现了训练+检验过程，训练样本手动输入10次，其后手动输入测试样例。



**样例（1）- 训练**



**样例（1）- 验证**



**样例（2）- 验证**

过程中遇到的问题：由于初始权值在作业中我们计算过是合理的。在多次的测试中都验证成功。但是在某一次训练及测试中，发现输入1，1，1得到的结果1被判断成了偶数，出现了误差。原因是测试样例过少，程序代码还不够完善，应该写个随机样例生成器进行更多样例的测试。并且将数据用作测试计算误差。