信号与系统实验报告

名 称：数字信号卷积和的实现及应用

学 院：计算机科学与工程学院

专 业：人工智能

学 号：58119304

姓 名：朱启鹏

日期：2021 年 04 月 04 日

评分：

一、实验目的

1. 掌握 Matlab 中相关函数的使用，程序代码编制与调试的流程。
2. 熟悉卷积和的运算规则及其意义，加深对离散时间信号分析的理解。

二、实验任务

1. 完成实验内容全部题目，分析解决调试代码过程中出现的问题。
2. 认真完成本次实验小结，思考卷积和的应用。

三、主要设备、软件平台

1. 硬件：计算机
2. 软件：Matlab

四、实验内容

1. 输出杨辉三角。
   1) 函数编写
   2) 控制台输出
   3) 循环语句、条件语句
   4) 程序运行、调试

代码：

```python
def generate(numRows):
    """
    :type numRows: int
    :rtype: List[List[int]]
    """
    if numRows == 0:
        return []
    if numRows == 1:
        return [[1]]
    if numRows == 2:
        return [[1], [1, 1]]
    numRows -= 2
    rList = [[1], [1, 1]]
    while numRows > 0:
        newList = [1]
        for i in range(len(rList[-1]) - 1):
            newList.append(rList[-1][i] + rList[-1][i + 1])
        newList.append(1)
        rList.append(newList)
        numRows -= 1
    return rList
```

运行结果：

```
Please input the numRows
[1]
[1, 1]
[1, 2, 1]
[1, 3, 3, 1]
[1, 4, 6, 4, 1]
[1, 5, 10, 10, 5, 1]
[1, 6, 15, 20, 15, 6, 1]
[1, 7, 21, 35, 35, 21, 7, 1]
[1, 8, 28, 56, 70, 56, 28, 8, 1]
[1, 9, 36, 84, 126, 126, 84, 36, 9, 1]
[1, 10, 45, 120, 210, 252, 210, 120, 45, 10, 1]
```

2. 编程实现信号 $x(n)$， $h(n)$间的卷积和运算函数 $my\_\mathrm{cov}(x,h)$，并绘制出下列信号卷积和波形。

1) $x(n)=[1,2,3,...,10]$， $h(n)=[1,1]$
2) $x(n)=[1,2,3,9,4,5,6,0,7,8]$， $h(n)=[-1,2,-1]$
3) $x(n)=[1,2,3,...,10]$， $h(n)=[1,2,3,..,10]$

代码：

```
import matplotlib.pyplot as plt
def draw_scatter(flag,str,path, x):
    plt.title(flag)
    plt.ylim(ymax=max(x)+ 2, ymin=-2)
    plt.xticks(range(0,len(x)+2))
    plt.xlabel("x")
    plt.ylabel(str)
    plt.plot(range(0,len(x)), x, 'ro')
    plt.savefig(path)
    plt.show()
```

```python
def my_cov(x, h):
    list.reverse(h)
    x = [0]*(len(h)-1)+x
    x = x + [0]*(len(h)-1)
    tp = []
    for j in range(len(x)-len(h)+1):
        temp = 0
        for i in range(len(h)):
            temp += h[i] * x[j+i]
        tp.append(temp)
    return tp
```

```python
x = list(range(1,11))
h = [1,1]
y = my_cov(x,h)
print(x,'*',h,'=',y)
draw_scatter("x1_scatter","x1","x1.png",x)
draw_scatter("h1_scatter","h1","h1.png",h)
draw_scatter("x1*h1_scatter","x1*h1","x1convh1.png",y)


x = list(range(1,9))
h = [-1,2,-1]
y = my_cov(x,h)
print(x,'*',h,'=',y)
draw_scatter("x1_scatter","x1","x2.png",x)
draw_scatter("h1_scatter","h1","h2.png",h)
draw_scatter("x1*h1_scatter","x1*h1","x2convh2.png",y)
```

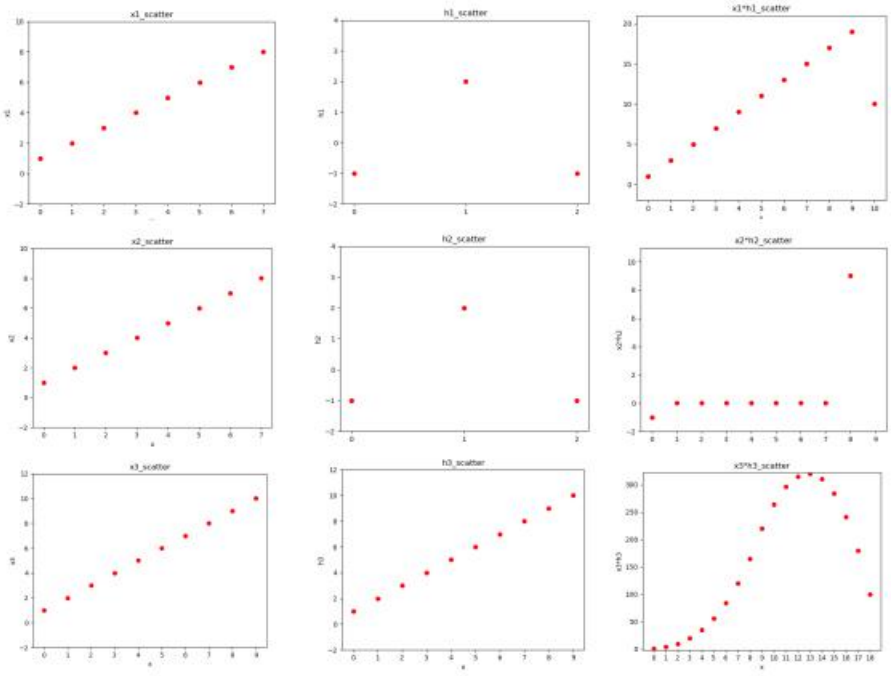
```python
x = list(range(1,11))
h = x[:]
draw_scatter("x1_scatter","x1","x3.png",x)
draw_scatter("h1_scatter","h1","h3.png",h)
y = my_cov(x,h)
print(x,'*',end="")
print(h,end="")
print('=',y)

draw_scatter("x1*h1_scatter","x1*h1","x3convh3.png",y)
```

运行结果：

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10] * [1, 1] = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 10]

[1, 2, 3, 4, 5, 6, 7, 8] * [-1, 2, -1] = [-1, 0, 0, 0, 0, 0, 0, 0, 9, -8]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10] *[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]=
[1, 4, 10, 20, 35, 56, 84, 120, 165, 220, 264, 296, 315, 320, 310, 284, 241, 180, 100]

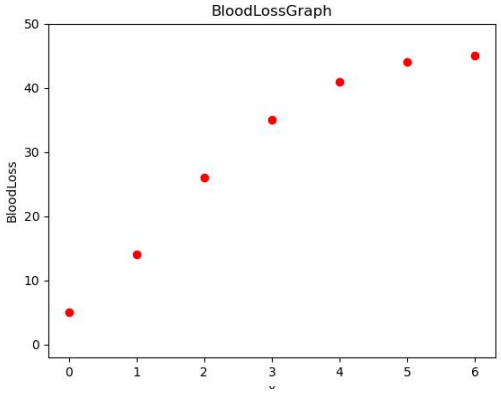


3.设计游戏时，若对小怪使用一次技能的效果是“小怪会在接下来 5 秒内持续掉血，每秒掉血量分别为[5 4 3 2 1]”；如果间隔 1 秒连续发动 3 次技能，请绘制出每次攻击后小怪的累计掉血量情况。

代码：

```python
import matplotlib.pyplot as plt

def affence(i, x, y):
    y = [0]*i + y
    x2 = x + [0]

    return np.sum([y, x2], axis=0).tolist()

def draw_scatter(path, x):
    plt.title("BloodLossGraph")
    plt.ylim(ymax=max(x)+ 5, ymin=-2)
    plt.xticks(range(0,len(x)+3))
    plt.xlabel("x")
    plt.ylabel("BloodLoss")
    plt.plot(range(0,len(x)), x, 'ro')
    plt.savefig(path)
    plt.show()

x = [5, 4, 3, 2, 1]
y = x[:]
x = affence(1, x, y)
x = affence(2, x, y)
for j in range(1, len(x)):
    x[j] += x[j - 1]
print(x)
draw_scatter("BloodLoss.png",x )
```

运行结果：[5, 14, 26, 35, 41, 44, 45]



五、探究拓展

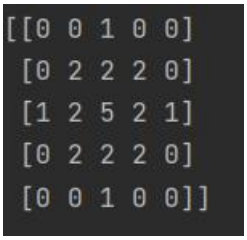
1. 给定一个如下所示的二维矩阵，实现其自身的卷积运算。

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

代码：

```
import numpy as np

f = np.array([[0, 1, 0],
              [1, 1, 1],
              [0, 1, 0]],dtype=np.int_)


def cov2(f):
    f1 = f.T
    inw, inh = f.shape#3,3
    outw = inw  + 4#7
    outh = inh + 4#7
    arr = np.zeros(shape=(outw - 2, outh - 2),dtype=np.int_)
    f2 = np.zeros(shape=(outw, outh),dtype=np.int_)
    for i in range(inw):
        for j in range(inh):
            f2[i+2][j+2] = f[i][j]
    print(f2)

    for j in range(outw - inw + 1):
        for i in range(outw - inw + 1):
            s = 0
            for g in range(inw):
                for k in range(inh):
                    s += f1[g][k] * f2[g+j][k+i]
            arr[j][i] = s
    return arr


print(cov2(f))
```

运行结果：



六、实验小结

经过此次的实验我更加清楚地了解了卷积以及卷积计算方面的原理。