DEEP LEARNING

Assignment-2

Submitted by:

Anshika Soni

Submitted to: Dr. Mahesh Kumar

SESSION: 2020-2024

Department of Computer Science & Engineering

JAYPEE UNIVERSITY OF ENGINEERING &TECHNOLOGY,

AB ROAD, RAGHOGARH, DT. GUNA-473226 MP, INDIA

Program 4:

```
import numpy as np

n=int(input("Enter the number of neurons in the hidden layer: "))

inputs=[]
num=int(input("Number of Inputs: "))
print("Enter the inputs: ")
for i in range(0,num):
    x=int(input())
    inputs.append(x)

def percept(n,inputs):
    Ohp=[]
    c=0
    wts=np.random.randn(n*num)
    print("Weights of Hidden Layer: ",wts)
    for i in range(0,n):
        o=0
        for j in range(0,num):
            o=o+wts[c]*inputs[j]
            c=c+1
        Ohp.append(o)
    print("Outputs of Hidden Layer: ",Ohp)
    y=0
    Wts=np.random.randn(n)
    print("Weights of Output Layer: ",Wts)
    for i in range(0,n):
        y=y+Wts[i]*Ohp[i]

    return y

Y=percept(n,inputs)
print("Output: ",Y)
```

Output:

```
============== RESTART: C:\Users\HP\Documents\Python Scripts\p4.py ========
Enter the number of neurons in the hidden layer: 2
Number of Inputs: 2
Enter the inputs:
2
3
Weights of Hidden Layer:  [-1.34166359 -2.00979179 -1.03087213 -1.42132448]
Outputs of Hidden Layer:  [-8.712702540893343, -6.325717700727962]
Weights of Output Layer:  [ 0.95942349 -0.79693597]
Output:  -3.317979461581248
>>>
```

Program 5:

```
import numpy as np

n=int(input("Enter the number of neurons in the hidden layer: "))

inputs=[]
num=int(input("Number of Inputs: "))
print("Enter the inputs: ")
for i in range(0,num):
    x=int(input())
    inputs.append(x)

def percept(n,inputs):
    Ohp=[]
    c=0
    np.random.seed(52)
    wts=np.random.randn(n*num)
    print("Weights of Hidden Layer: ",wts)
    for i in range(0,n):
        o=0
        for j in range(0,num):
            o=o+wts[c]*inputs[j]
            c=c+1
        Ohp.append(o)
    print("Outputs of Hidden Layer: ",Ohp)
    y=0
    np.random.seed(52)
    Wts=np.random.randn(n)
    print("Weights of Output Layer: ",Wts)
    for i in range(0,n):
        y=y+Wts[i]*Ohp[i]

    return y

Y=percept(n,inputs)
print("Output: ",Y)
```

Output:

```
Enter the number of neurons in the hidden layer: 2
Number of Inputs: 2
Enter the inputs:
2
3
Weights of Hidden Layer:  [ 0.51947584 -1.26875038  0.24042003 -0.80395743]
Outputs of Hidden Layer:  [-2.7672994610363832, -1.9310322430015345]
Weights of Output Layer:  [ 0.51947584 -1.26875038]
Output:  1.0124526798051972
```

Program 6:

```
import numpy as np
import matplotlib.pyplot as plt

n=int(input("Enter the number of neurons in the hidden layer: "))

inputs=[]
num=int(input("Number of Inputs: "))
print("Enter the inputs: ")
for i in range(0,num):
    x=int(input())
    inputs.append(x)

def percept(n,inputs):
    Ohp=[]
    c=0
    np.random.seed(52)
    wts=np.random.randn(n*num)
    print("Weights of Hidden Layer: ",wts)
    for i in range(0,n):
        o=0
        for j in range(0,num):
            o=o+wts[c]*inputs[j]
            c=c+1
        sig1=1/(1+np.exp(-o))
        Ohp.append(sig1)
    print("Outputs of Hidden Layer: ",Ohp)
    y=0
    np.random.seed(52)
    Wts=np.random.randn(n)
    print("Weights of Output Layer: ",Wts)
    for i in range(0,n):
        y=y+Wts[i]*Ohp[i]
    sig2=1/(1+np.exp(-y))

    return sig2

Y=percept(n,inputs)
print("Output: ",Y)
```

Output:

```
========== RESTART: C:\Users\HP\Documents\Python Scripts\p6(sig).py =======
Enter the number of neurons in the hidden layer: 2
Number of Inputs: 2
Enter the inputs:
2
3
Weights of Hidden Layer:  [ 0.51947584 -1.26875038  0.24042003 -0.80395743]
Outputs of Hidden Layer:  [0.0591170447148126, 0.1266363704524397]
Weights of Output Layer:  [ 0.51947584 -1.26875038]
Output:  0.4675556348817955
>>>
```

Program 7:

```
import numpy as np

n=int(input("Enter the number of neurons in the hidden layer: "))

inputs=[]
num=int(input("Number of Inputs: "))
print("Enter the inputs: ")
for i in range(0,num):
    x=int(input())
    inputs.append(x)

def percept(n,inputs):
    Ohp=[]
    c=0
    np.random.seed(52)
    wts=np.random.randn(n*num)
    print("Weights of Hidden Layer: ",wts)
    for i in range(0,n):
        o=0
        for j in range(0,num):
            o=o+wts[c]*inputs[j]
            c=c+1
        tanh1=(2/(1+np.exp(-2*o)))-1
        Ohp.append(tanh1)
    print("Outputs of Hidden Layer: ",Ohp)
    y=0
    np.random.seed(52)
    Wts=np.random.randn(n)
    print("Weights of Output Layer: ",Wts)
    for i in range(0,n):
        y=y+Wts[i]*Ohp[i]
    tanh2=(2/(1+np.exp(-2*y)))-1

    return tanh2

Y=percept(n,inputs)
print("Output: ",Y)
```

Output:

```
========== RESTART: C:\Users\HP\Documents\Python Scripts\p7(tanh).py =====
Enter the number of neurons in the hidden layer: 2
Number of Inputs: 2
Enter the inputs:
2
3
Weights of Hidden Layer:  [ 0.51947584 -1.26875038  0.24042003 -0.80395743]
Outputs of Hidden Layer:  [-0.9921354641679867, -0.9588167594928882]
Weights of Output Layer:  [ 0.51947584 -1.26875038]
Output:  0.6050710568666464
>>>
```

Program 8:

```
import numpy as np

n=int(input("Enter the number of neurons in the hidden layer: "))

inputs=[]
num=int(input("Number of Inputs: "))
print("Enter the inputs: ")
for i in range(0,num):
    x=int(input())
    inputs.append(x)

def percept(n,inputs):
    Ohp=[]
    c=0
    np.random.seed(52)
    wts=np.random.randn(n*num)
    print("Weights of Hidden Layer: ",wts)
    for i in range(0,n):
        o=0
        for j in range(0,num):
            o=o+wts[c]*inputs[j]
            c=c+1
        m1=max(0,o)
        Ohp.append(m1)
    print("Outputs of Hidden Layer: ",Ohp)
    y=0
    np.random.seed(52)
    Wts=np.random.randn(n)
    print("Weights of Output Layer: ",Wts)
    for i in range(0,n):
        y=y+Wts[i]*Ohp[i]
    m2=max(0,y)

    return m2

Y=percept(n,inputs)
print("Output: ",Y)
```

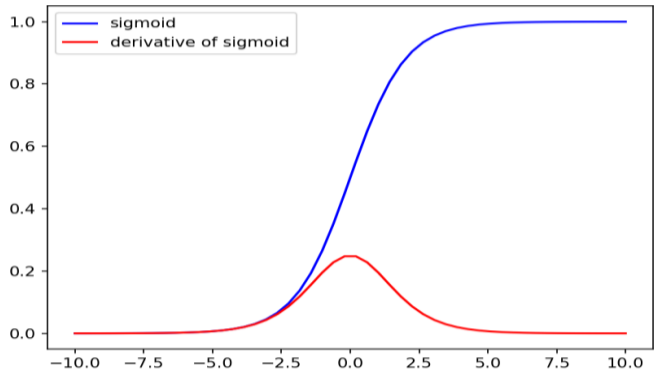
Output:

```
========== RESTART: C:\Users\HP\Documents\Python Scripts\p8(relu).py =======
Enter the number of neurons in the hidden layer: 2
Number of Inputs: 2
Enter the inputs:
2
3
Weights of Hidden Layer:  [ 0.51947584 -1.26875038  0.24042003 -0.80395743]
Outputs of Hidden Layer:  [0, 0]
Weights of Output Layer:  [ 0.51947584 -1.26875038]
Output:  0
>>>
```

Program 9:

```
import numpy as np
import matplotlib.pyplot as plt

def sigmoid(x):
    return 1.0/(1.0+np.exp(-x))

def d_sig(x):
    return (np.exp(-x))/((1+np.exp(-x))*(1+np.exp(-x)))

x=np.linspace(-10,10)
plt.plot(x,sigmoid(x),'blue',label='sigmoid')
plt.plot(x,d_sig(x),'red',label='derivative of sigmoid')
plt.legend()
plt.show()
```

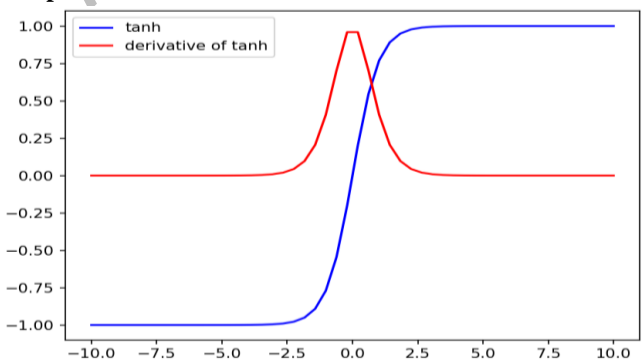
Output:



Program 10:

```
import numpy as np
import matplotlib.pyplot as plt

def tanh(x):
    return (np.exp(x)-np.exp(-x))/(np.exp(x)+np.exp(-x))

def d_tanh(x):
    return (4*np.exp(-x)*np.exp(x))/((np.exp(x)+np.exp(-x))*(np.exp(x)+np.exp(-x)))

x=np.linspace(-10,10)
plt.plot(x,tanh(x),'blue',label='tanh')
plt.plot(x,d_tanh(x),'red',label='derivative of tanh')
plt.legend()
plt.show()
```

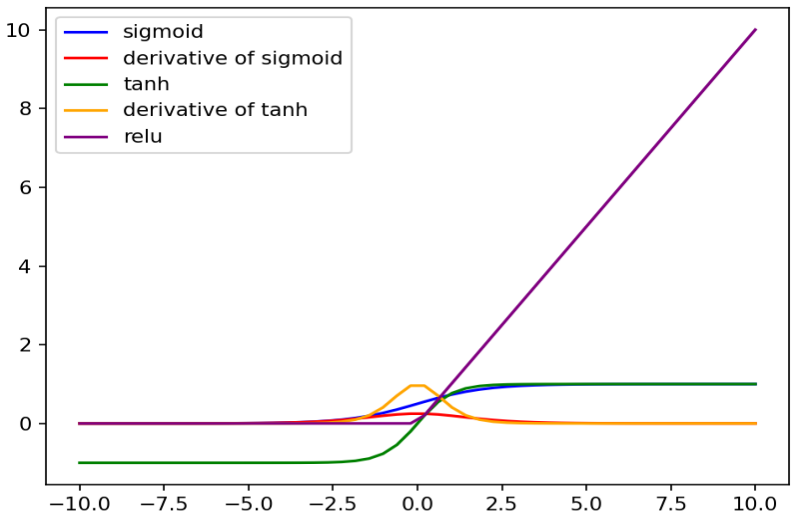
Output:



Program 11:

```
import numpy as np

import matplotlib.pyplot as plt

def sigmoid(x):
    return 1.0/(1.0+np.exp(-x))

def d_sig(x):
    return (np.exp(-x))/((1+np.exp(-x))*(1+np.exp(-x)))

def tanh(x):
    return (np.exp(x)-np.exp(-x))/(np.exp(x)+np.exp(-x))

def d_tanh(x):
    return (4*np.exp(-x)*np.exp(x))/((np.exp(x)+np.exp(-x))*(np.exp(x)+np.exp(-x)))

def relu(x):
    return np.maximum(0,x)

x=np.linspace(-10,10)

plt.plot(x,sigmoid(x),'blue',label='sigmoid')
plt.plot(x,d_sig(x),'red',label='derivative of sigmoid')
plt.plot(x,tanh(x),'green',label='tanh')
plt.plot(x,d_tanh(x),'orange',label='derivative of tanh')
plt.plot(x,relu(x),'purple',label='relu')
plt.legend()
plt.show()
```

Output:



Program 12:

```
import numpy as np

x1=[0,0,1,1]
x2=[0,1,0,1]
y=[0,1,1,1]
b=np.random.randn(1);

while(True):
    flag=True
    for i in range(0,4):
        o=x1[i]+x2[i]-b
        if o>=0:
            Y=1
        else:
            Y=0
        if y[i]!=Y:
            flag=False
            if y[i]>Y:
                b=b-1
                break
            else:
                b=b+1
                break

    if flag==True:
        break

for i in range(0,4):
    print("Input 1: ",x1[i])
    print("Input 2: ",x2[i])
    o=x1[i]+x2[i]-b
    print("Output: ",o)
    if o>=0:
        Y=1
    else:
        Y=0
    print("Actual Output: ",Y)
    print()

print("Final Threshold Value is: ",b)
```

Output:

```
>>>
========= RESTART: C:/Users/HP/Documents/Python Sc
Input 1:  0
Input 2:  0
Output:  [-0.70621328]
Actual Output:  0

Input 1:  0
Input 2:  1
Output:  [0.29378672]
Actual Output:  1

Input 1:  1
Input 2:  0
Output:  [0.29378672]
Actual Output:  1

Input 1:  1
Input 2:  1
Output:  [1.29378672]
Actual Output:  1

Final Threshold Value is:  [0.70621328]
```