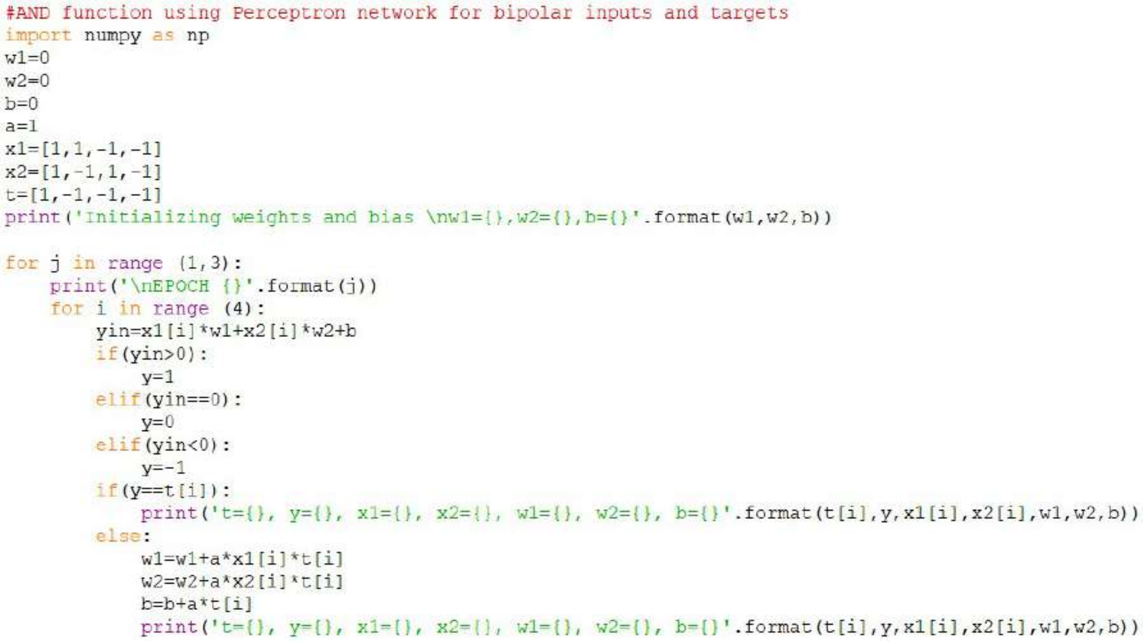
EXPT: Preceptron Network

AND function using Perceptron network for bipolar inputs and targets

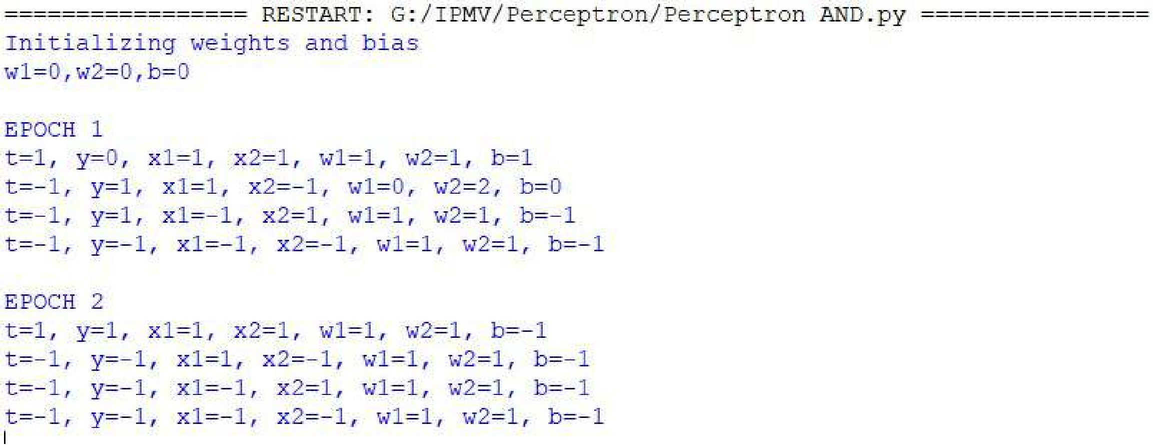
Program:



```
#AND function using Perceptron network for bipolar inputs and targets
import numpy as np
w1=0
w2=0
b=0
a=1
x1=[1,1,-1,-1]
x2=[1,-1,1,-1]
t=[1,-1,-1,-1]
print('Initializing weights and bias \nw1={},w2={},b={}'.format(w1,w2,b))

for j in range (1,3):
    print('\nEPOCH {}'.format(j))
    for i in range (4):
        yin=x1[i]*w1+x2[i]*w2+b
        if(yin>0):
            y=1
        elif(yin==0):
            y=0
        elif(yin<0):
            y=-1
        if(y==t[i]):
            print('t={}, y={}, x1={}, x2={}, w1={}, w2={}, b={}'.format(t[i],y,x1[i],x2[i],w1,w2,b))
        else:
            w1=w1+a*x1[i]*t[i]
            w2=w2+a*x2[i]*t[i]
            b=b+a*t[i]
            print('t={}, y={}, x1={}, x2={}, w1={}, w2={}, b={}'.format(t[i],y,x1[i],x2[i],w1,w2,b))
```

Output:



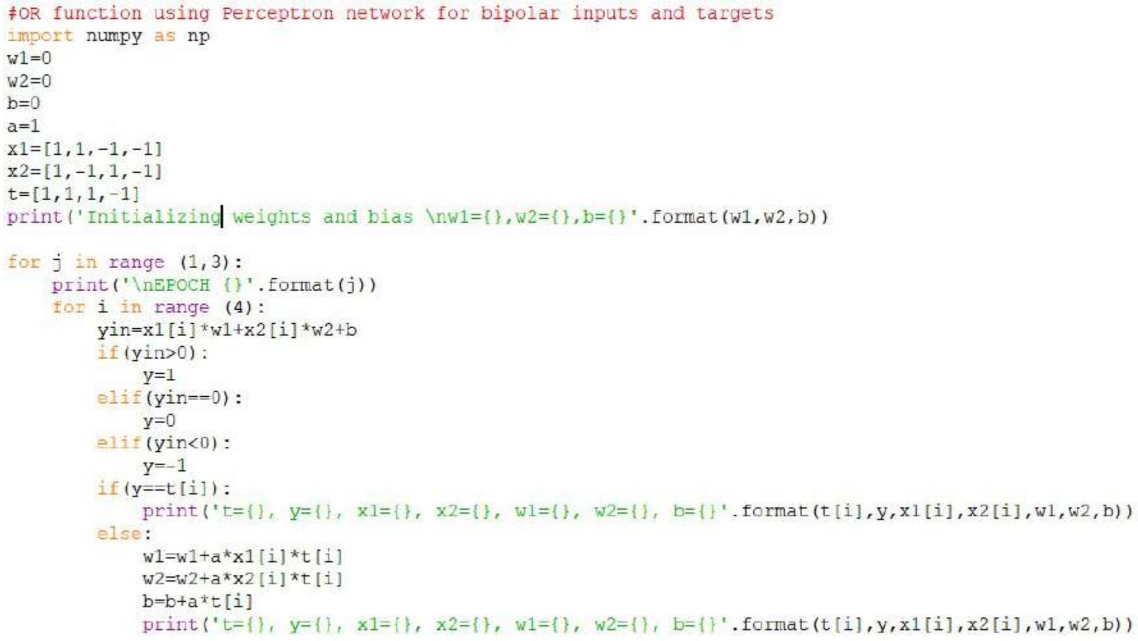
```
================= RESTART: G:/IPMV/Perceptron/Perceptron AND.py ================
Initializing weights and bias
w1=0,w2=0,b=0

EPOCH 1
t=1, y=0, x1=1, x2=1, w1=1, w2=1, b=1
t=-1, y=1, x1=1, x2=-1, w1=0, w2=2, b=0
t=-1, y=1, x1=-1, x2=1, w1=1, w2=1, b=-1
t=-1, y=-1, x1=-1, x2=-1, w1=1, w2=1, b=-1

EPOCH 2
t=1, y=1, x1=1, x2=1, w1=1, w2=1, b=-1
t=-1, y=-1, x1=1, x2=-1, w1=1, w2=1, b=-1
t=-1, y=-1, x1=-1, x2=1, w1=1, w2=1, b=-1
t=-1, y=-1, x1=-1, x2=-1, w1=1, w2=1, b=-1
```

OR function using Perceptron network for bipolar inputs and targets

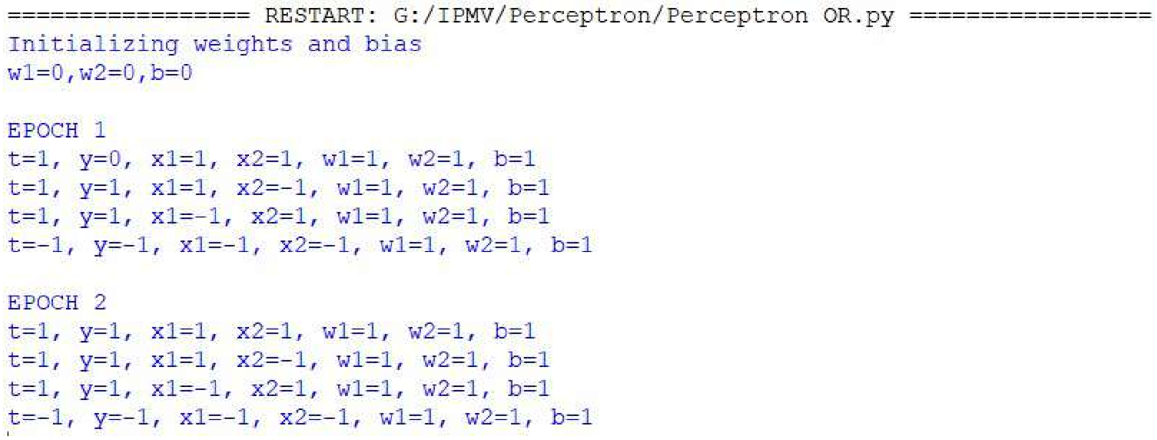
Program:



```
#OR function using Perceptron network for bipolar inputs and targets
import numpy as np
w1=0
w2=0
b=0
a=1
x1=[1,1,-1,-1]
x2=[1,-1,1,-1]
t=[1,1,1,-1]
print('Initializing weights and bias \nw1={},w2={},b={}'.format(w1,w2,b))

for j in range (1,3):
    print('\nEPOCH {}'.format(j))
    for i in range (4):
        yin=x1[i]*w1+x2[i]*w2+b
        if(yin>0):
            y=1
        elif(yin==0):
            y=0
        elif(yin<0):
            y=-1
        if(y==t[i]):
            print('t={}, y={}, x1={}, x2={}, w1={}, w2={}, b={}'.format(t[i],y,x1[i],x2[i],w1,w2,b))
        else:
            w1=w1+a*x1[i]*t[i]
            w2=w2+a*x2[i]*t[i]
            b=b+a*t[i]
            print('t={}, y={}, x1={}, x2={}, w1={}, w2={}, b={}'.format(t[i],y,x1[i],x2[i],w1,w2,b))
```

Output:



```
================= RESTART: G:/IPMV/Perceptron/Perceptron OR.py =================
Initializing weights and bias
w1=0,w2=0,b=0

EPOCH 1
t=1, y=0, x1=1, x2=1, w1=1, w2=1, b=1
t=1, y=1, x1=1, x2=-1, w1=1, w2=1, b=1
t=1, y=1, x1=-1, x2=1, w1=1, w2=1, b=1
t=-1, y=-1, x1=-1, x2=-1, w1=1, w2=1, b=1

EPOCH 2
t=1, y=1, x1=1, x2=1, w1=1, w2=1, b=1
t=1, y=1, x1=1, x2=-1, w1=1, w2=1, b=1
t=1, y=1, x1=-1, x2=1, w1=1, w2=1, b=1
t=-1, y=-1, x1=-1, x2=-1, w1=1, w2=1, b=1
```