PRACTICAL-9

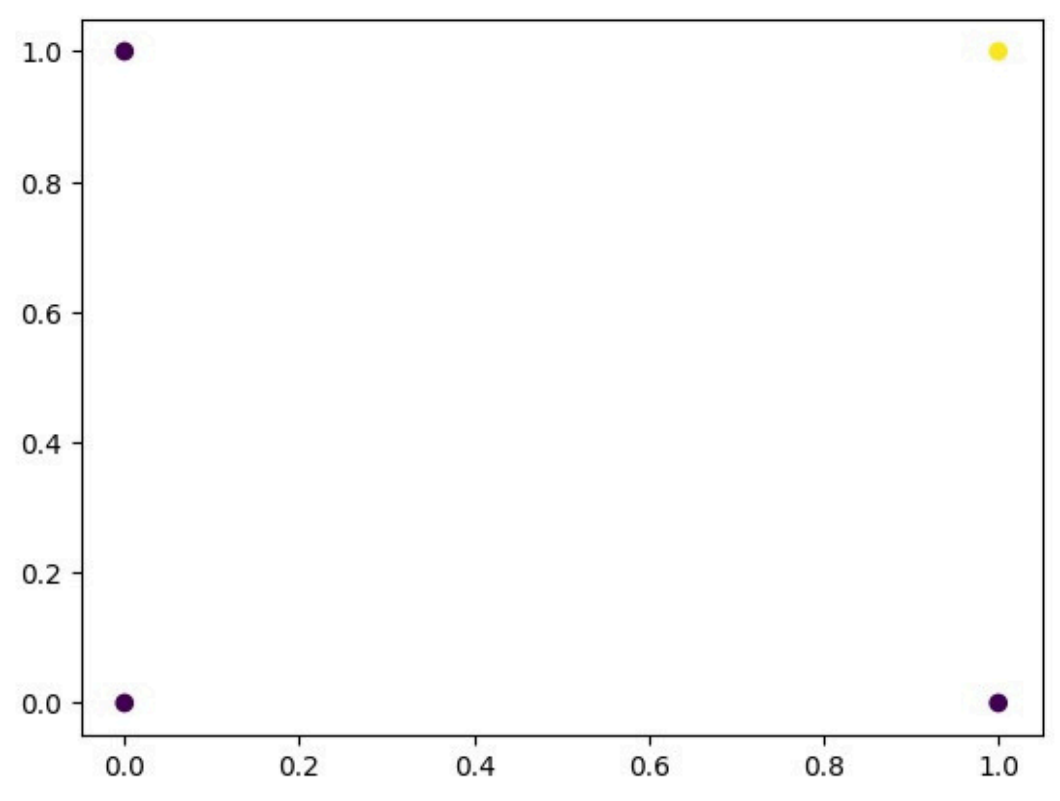
Name:Priyansh Panchal

Roll No: 24BEE106

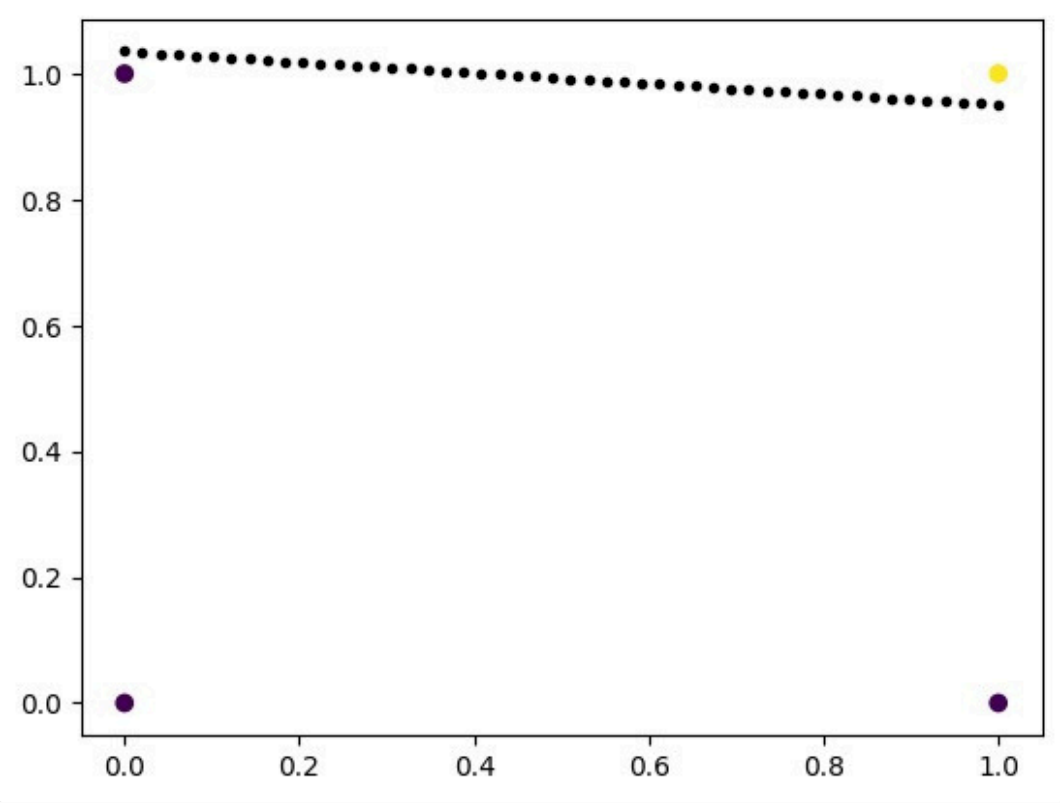
Aim: Write a program that implements AND gate using perceptron learning algorithm.

```
# AND GATE import numpy as np from matplotlib import pyplot as plt
#XOR truth table
X=np.array([[0,0],[0,1],[1,0],[1,1]])
Y=np.array([0,0,0,1])
#scatter plot. Notice data points are not linearly separable
#A network without a hidden layer can't learn to separate them
plt.scatter(x=X[:,0],y=X[:,1],c=Y)
plt.show()
#We will try to demonstrate the XOR learning
n_samples=X.shape[0]
n_features=X.shape[1]
#Initial weights and bias are random
w=np.random.uniform(0,1,size=n_features)
b=np.random.uniform(0,1,1)
#Scan number of epochs
n_epoch=int(input("Enter the number of epochs: "))
#learning rate
lr=0.01
for e in range(n_epoch):

    for s in range(n_samples):
    net=np.dot(X[s,:],w)+b
    if net>=0:
            a=1
        else:
            a=0
        error=Y[s]-a
        w=w+lr*error*X[s,:]
        b=b+lr*error

#calculate slope and intercept
m=-w[0]/w[1]
```

```
c=-b/w[1]

#function to plot decision boundary along with training points
#notice that the line is not able to separate data points
def plot_decision_boundary(X):
    for x in np.linspace(np.min(X[:,0]), np.max(X[:,0])):
        y = m*x + c
        plt.plot(x,y,linestyle='-',color='k',marker='.')
    plt.scatter(X[:,0],X[:,1],c=Y)
    plt.show()
plot_decision_boundary(X)
```




```
Enter the number of epochs: 100
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0