

**Assignment 5: Write a program to demonstrate the perceptron learning law with its decision region using python. Give the output in graphical form**

```
Code =>
import numpy as np

import matplotlib.pyplot as plt

# Step Function for Activation
def step_function(x):
    return np.where(x >= 0, 1, 0)

# Perceptron Learning Algorithm
def perceptron_learning(X, y, learning_rate=0.1, epochs=100):
    weights = np.random.rand(X.shape[1])
    bias = np.random.rand()

    for epoch in range(epochs):
        for i in range(len(X)):
            linear_output = np.dot(X[i], weights) + bias
            prediction = step_function(linear_output)
            error = y[i] - prediction
            # Update Rule
            weights += learning_rate * error * X[i]
            bias += learning_rate * error

    return weights, bias

# Generate Data (AND Logic Gate Example)
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
y = np.array([0, 0, 0, 1])
# Train Perceptron
weights, bias = perceptron_learning(X, y)
# Plot Decision Boundary
x_vals = np.linspace(-0.5, 1.5, 100)
y_vals = -(weights[0] * x_vals + bias) / weights[1]
plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c=y, cmap='bwr', edgecolors='k')
plt.plot(x_vals, y_vals, color='black', label='Decision Boundary')
plt.xlim(-0.5, 1.5)
plt.ylim(-0.5, 1.5)
plt.xlabel('X1')
plt.ylabel('X2')
plt.title('Perceptron Decision Boundary for AND Logic Gate')
plt.legend()
plt.grid(True)
plt.show()
```

Output =>

