

Name: Nitin Choudhary

Roll no: 50

Class: TY-IT A B3

Single Class Single Layer Perceptron Code:

```
import numpy as np
import matplotlib.pyplot as plt
def signum(net):
    return -1 if net < 0 else 1
d = np.array([1, -1, 1])
Weights = np.array([1, -1, 0, 0.5])
c = 1
x = np.array([
    [1, -2, 1.5, 0],
    [1, -0.5, -2, -1.5],
    [0, 1, -1, 1.5]
])

error_list = []
epochs = 100 # Number of training epochs

for epoch in range(epochs):
    error = 0
    for i in range(3):
        net = np.dot(Weights, x[i])
        o = signum(int(net))
        y = d[i] - o

        for j in range(4):
            if y != 0:
                Weights[j] = Weights[j] + c * y * x[i][j]

        error += 0.5 * (d[i] - o) * (d[i] - o)
    # total_error += error
    print(f"W{i + 1} = {Weights}")
    print(f"Error after processing X{i + 1}: {error}")
    error_list.append(error)

    if error == 0:
        print(f"Training completed in epoch {epoch + 1}/{epochs}, Error is 0")
        break
```

```
# Plotting the error graph
plt.plot(range(1, len(error_list) + 1), error_list, marker='o')
plt.xlabel('Iteration')
plt.ylabel('Error')
plt.title('Error vs. Iteration')
plt.grid(True)
plt.show()
```

Output:

W1 = [1. -1. 0. 0.5]
Error after processing X1: 0.0
W2 = [-1. 0. 4. 3.5]
Error after processing X2: 2.0
W3 = [-1. 0. 4. 3.5]
Error after processing X3: 2.0
W1 = [-1. 0. 4. 3.5]
Error after processing X1: 0.0
W2 = [-1. 0. 4. 3.5]
Error after processing X2: 0.0
W3 = [-1. 0. 4. 3.5]
Error after processing X3: 0.0
Training completed in epoch 2/100, Error is 0

