

Perceptron Learning Algorithm

Code:

```
# Input Vectors (3 training examples, each of size 4)
X = [
    [1, -2, 0, -1],      # x1
    [0, 1.5, -0.5, -1],  # x2
    [-1, 1, 0.5, -1]     # x3
]

# Desired outputs
D = [-1, -1, 1]

# Initial weights
weights = [1, -1, 0, 0.5]

# Learning rate
alpha = 0.1

# Sign activation function
def sign(value):
    return 1 if value >= 0 else -1

# Training process
print("Initial Weights:", weights)

for i in range(len(X)):
    x = X[i]
    d = D[i]

    # Compute net input
    net = sum(weights[j] * x[j] for j in range(4))
    y = sign(net)

    print(f"\nStep {i+1}:")
    print(f"Input Vector x{i+1} = {x}")
    print(f"Desired Output d{i+1} = {d}")
    print(f"Net = {net}")
    print(f"Predicted Output = {y}")
```

```

# Check if output matches
if y != d:
    print("Mismatch! Updating weights...")
    error = d - y
    for j in range(4):
        weights[j] = weights[j] + alpha * error * x[j]
    print(f"Updated Weights = {weights}")
else:
    print("Match! No update needed.")
    print(f"Weights remain = {weights}")

print("\nFinal Weights after training:", weights)

```

Output:

```

C:\Windows\System32\Windc x + v
[08/19/2025 @ 3:41:46 AM IST][vism0 :: VishalRMahajan][RAM: 7/7GB][100%][0.07s]
D:\SFIT-MU-BE-IT-AllSemesters\Semester 7\AIDS-2 Lab\EXP04[main ?1]
> python .\perceptron.py
Initial Weights: [1, -1, 0, 0.5]

Step 1:
Input Vector x1 = [1, -2, 0, -1]
Desired Output d1 = -1
Net = 2.5
Predicted Output = 1
Mismatch! Updating weights...
Updated Weights = [0.8, -0.6, 0.0, 0.7]

Step 2:
Input Vector x2 = [0, 1.5, -0.5, -1]
Desired Output d2 = -1
Net = -1.5999999999999999
Predicted Output = -1
Match! No update needed.
Weights remain = [0.8, -0.6, 0.0, 0.7]

Step 3:
Input Vector x3 = [-1, 1, 0.5, -1]
Desired Output d3 = 1
Net = -2.1
Predicted Output = -1
Mismatch! Updating weights...
Updated Weights = [0.6000000000000001, -0.3999999999999997, 0.1, 0.4999999999999994]

Final Weights after training: [0.6000000000000001, -0.3999999999999997, 0.1, 0.4999999999999994]

```