

xp855v3uu

March 31, 2025

0.1 Practical 5

```
[ ]: import numpy as np

def train_bam(X, Y):
    weight_matrix = np.zeros((X.shape[1], Y.shape[1]))
    for x, y in zip(X, Y):
        weight_matrix += np.outer(x, y)
    return weight_matrix

def recall_bam(weight_matrix, input_vector, direction="X_to_Y"):
    if direction == "X_to_Y":
        return np.sign(np.dot(input_vector, weight_matrix))
    elif direction == "Y_to_X":
        return np.sign(np.dot(input_vector, weight_matrix.T))
    else:
        raise ValueError("Invalid direction. Choose 'X_to_Y' or 'Y_to_X'")

# Define input-output pairs
X = np.array([[1, -1, 1], [-1, 1, -1]])
Y = np.array([[1, 1, -1], [-1, -1, 1]])

# Train BAM
weight_matrix = train_bam(X, Y)

# Test recall
test_X = np.array([1, -1, 1])
output_Y = recall_bam(weight_matrix, test_X, direction="X_to_Y")
print("Recalled Y from X:", output_Y)

test_Y = np.array([1, 1, -1])
output_X = recall_bam(weight_matrix, test_Y, direction="Y_to_X")
print("Recalled X from Y:", output_X)
```