

Class (Section): \_\_\_\_\_

Date: \_\_\_\_\_

Name: \_\_\_\_\_

Deep Learning

Roll No: \_\_\_\_\_

Dr. Zohair Ahmed

### ACTIVITY SHEET 1: Understanding Rosenblatt's Perceptron

*A hands-on worksheet to learn how each step of the perceptron algorithm works*

#### Section 1 — What is a Perceptron? (Review)

Fill in the blanks:

1. A perceptron is a type of \_\_\_\_\_ neural network.
2. It has \_\_\_\_\_ layer(s).
3. The activation function used is called a \_\_\_\_\_ (also known as step function).
4. The perceptron creates a \_\_\_\_\_ boundary to separate classes.
5. The perceptron can learn only \_\_\_\_\_ separable problems.

#### Section 2 — Truth Table (Training Data)

Use **AND gate** as the training dataset.

x1	x2	Target (T)
0	0	_____
0	1	_____
1	0	_____
1	1	_____

#### Section 3 — Initialize Parameters

Fill the empty boxes with **your chosen random values**:

- Weight  $w_1$  = \_\_\_\_\_
- Weight  $w_2$  = \_\_\_\_\_
- Bias  $b$  = \_\_\_\_\_
- Learning rate  $\eta$  = \_\_\_\_\_ (usually 0.1)

#### Section 4 — Perceptron Output Calculation

For each input pattern:

1. Compute weighted sum:

$$v = w_1x_1 + w_2x_2 + b$$

2. Apply hard limiter:

$$O = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases}$$

Fill in the table below:

x1	x2	$v = w_1 \cdot x_1 + w_2 \cdot x_2 + b$	Output O	Target T	Error (T-O)
0	0	_____	_____	_____	_____
0	1	_____	_____	_____	_____
1	0	_____	_____	_____	_____
1	1	_____	_____	_____	_____

#### Section 5 — Weight Update Rule

If **Error**  $\neq 0$ , update weights:

$$w_j \leftarrow w_j + \eta \cdot (\text{Error}) \cdot x_j$$

$$b \leftarrow b + \eta \cdot (\text{Error})$$

Fill in after each training example:

Step	x1	x2	Error	New w1	New w2	New b
1	0	0	_____	_____	_____	_____
2	0	1	_____	_____	_____	_____
3	1	0	_____	_____	_____	_____
4	1	1	_____	_____	_____	_____

---

### Section 6 — Epoch Completion

Answer:

1. Did the perceptron make any mistakes in this epoch?  
**Yes / No** → Circle one
2. If **Yes**, start a new epoch using the **updated weights and bias**.
3. If **No**, the perceptron has **converged** (learned the AND function).

---

### Section 7 — Final Results

Fill in:

- Final w1 = \_\_\_\_\_
- Final w2 = \_\_\_\_\_
- Final b = \_\_\_\_\_
- Number of epochs until convergence = \_\_\_\_\_
- Accuracy = \_\_\_\_\_ %

---

### Section 8 — Reflection Questions

1. Why does the perceptron only learn linearly separable problems?
2. What role does the **bias** play in shifting the decision boundary?
3. Why do some inputs **not** update the weights?
4. Predict: What will happen if we try XOR?