Perceptron Learning Algorithm: NAND & XOR Visualization

1. Objective of the Assignment (30 words)

To implement and visualize the Perceptron Learning Algorithm on NAND and XOR datasets, demonstrating linear separability, decision boundary evolution, and limitations of single-layer neural networks through epoch-wise graphical analysis clearly.

---

2. Description of the Assignment

This assignment focuses on implementing a single-layer perceptron from scratch using Python.
The perceptron is trained on two logical datasets—**NAND** and **XOR**—to study its learning behavior.

During training, the decision boundary is visualized and saved as images after each epoch.
This helps observe how the perceptron converges for linearly separable data (NAND) and fails for non-linearly separable data (XOR).

---

3. Explanation of the Assignment and Code

a) Libraries Used

- **NumPy**
  Used for numerical computations such as vector dot products and weight updates.
- **Matplotlib**
  Used for plotting input data points and the perceptron's decision boundary.
- **Seaborn**
  Used only for improving plot aesthetics and readability.
- **os**
  Used to create directories for saving epoch-wise images.

---

b) Code Structure

The program is written in a **single Python file** and consists of:

1. **Perceptron Class**
   - Initializes weights, bias, and learning rate
   - Implements step activation function
   - Contains training logic and visualization
2. **Training Section**
   - NAND dataset training
   - XOR dataset training
3. **Output Generation**
   - Saves one image per epoch showing the decision boundary

c) Core Logic of the Perceptron

- The perceptron computes a weighted sum of inputs and bias.
- A **step activation function** converts the output into binary class labels.
- Weights are updated using the **error-correction learning rule**:

  $$w = w + \eta \cdot error \cdot x$$

- Training continues until:
  - All samples are classified correctly, or
  - The maximum number of epochs is reached.

---

d) Visualization Logic

- At every epoch:
  - Input points are plotted
  - The current decision boundary is drawn
  - The plot is saved as an image
- For **NAND**, the boundary stabilizes.
- For **XOR**, the boundary keeps shifting and never converges.

---

4. What I Learned from This Assignment

- How a single-layer perceptron works internally
- Why linear separability is crucial for perceptron convergence
- Why XOR cannot be solved by a single perceptron
- How learning rate and weight updates affect decision boundaries
- How to visualize and interpret learning behavior graphically
- The practical limitations of simple neural network models

Conclusion:

This assignment provided a strong conceptual and practical understanding of perceptron's, their strengths, and their fundamental limitations in neural network learning.