**Objective**

1. Write a Program to implement the Perceptron Learning Algorithm using numpy in Python. Evaluate performance of a single perceptron for NAND and XOR truth tables as input dataset.

**Description of the Perceptron Algorithm**

A perceptron is one of the simplest types of artificial neural networks. It takes multiple inputs, applies weights to them, sums them up, and passes the result through an activation function (usually a step function). Based on this, it decides whether to output a 0 or a 1.

The learning process involves:

1. Initializing the weights and bias randomly.
2. Calculating the output using the step function.
3. Comparing the predicted output with the actual output.
4. Adjusting the weights and bias using the Perceptron Learning Rule: w=w+η⋅(ytrue−ypred)⋅xw = w + \eta \cdot (y\_{true} - y\_{pred}) \cdot xw=w+η⋅(ytrue​−ypred​)⋅x b=b+η⋅(ytrue−ypred)b = b + \eta \cdot (y\_{true} - y\_{pred})b=b+η⋅(ytrue​−ypred​)
5. Repeating this process for multiple iterations (epochs) until the perceptron classifies all inputs correctly (if possible).

**Description of the Code:**

1. **Perceptron Class**:
   * Initializes weights and bias randomly.
   * Implements the activation function (step function).
   * Implements the training function using the perceptron learning rule.
   * Defines a prediction function.
2. **Training and Evaluation for NAND and XOR Gates**:
   1. The NAND truth table (nand\_X) and labels (nand\_y) are defined.
   2. The perceptron is trained using the train() method.
   3. The model accuracy is calculated using the evaluate() function.
   4. The predictions for all inputs in the NAND table are displayed.
   5. The same process is repeated for XOR.
3. **Performance Evaluation**:
   * Since NAND is linearly separable, the perceptron successfully learns it.
   * XOR is not linearly separable, so the perceptron fails to classify XOR correctly.

**Output and Performance Analysis:**

* **For NAND Gate**: The perceptron will successfully classify all inputs and produce correct outputs.
* **For XOR Gate**: The perceptron will fail to learn the correct function since XOR is not linearly separable.

**Personal Comments:**

* The perceptron works well for linearly separable problems. (e.g., NAND).
* For non-linearly separable problems like XOR, a single-layer perceptron is insufficient. A multi-layer perceptron (MLP) with a hidden layer is required to learn XOR.
* This experiment highlights the limitations of simple perceptron and the need for more advanced neural network architectures.