|  |  |
| --- | --- |
| Subject: Machine Learning Lab | Course ID: CSL-604 |
| Semester: VI | Course: AI & DS |
| Laboratory: 406-B | Name of teacher: Prof. Seema Pawar |
| Name of Student: Ayush Gupta | Roll No: VU2S2324001 |

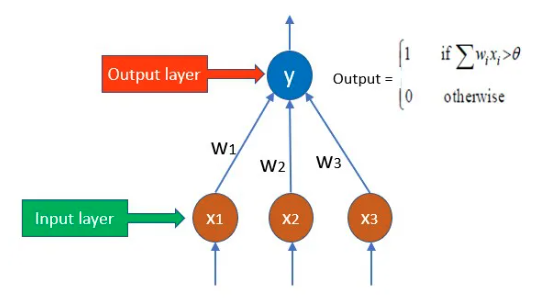
**EXPERIMENT NO. 7**

**Aim:**

To implement the Single Layer Perceptron Learning Algorithm.

**Theory:**

A Single Layer Perceptron (SLP) is one of the simplest models of an Artificial Neural Network (ANN). It is a type of feed-forward neural network where input values are weighted and summed before being passed through an activation function.



**Key Characteristics of Single Layer Perceptron:**

* It consists of only one layer of neurons (also called perceptron’s).
* It works well only for linearly separable problems.
* It does not use backpropagation for learning.
* It adjusts weights using a simple weight update rule.
* It uses an activation function such as Step Function or Sign Function to determine the output.

**Limitations of SLP:**

* It cannot solve problems that are not linearly separable, such as the XOR problem.
* It is limited to binary classification tasks.
* For complex problems, a Multi-Layer Perceptron (MLP) is required.

**Algorithm:**

1. Initialize weights randomly since there is no prior knowledge.
2. Initialize a threshold value randomly.
3. Compute the weighted sum of input features.
4. Apply the activation function to determine the output:
   * If the weighted sum is above the threshold, the perceptron is activated (output = 1).
   * Otherwise, it remains inactive (output = 0).
5. Compare the predicted output with the desired output:
   * If the output matches the target, no changes are made.
   * If not, adjust the weights using the weight update rule:

Perceptron Weight Adjustment Rule:

Δw = η \* d \* x

where:

* d = (Predicted Output - Desired Output)
* η = Learning Rate (a small value, usually < 1)
* x = Input Data

1. Repeat the process until the model converges or a stopping criterion is met.

**Learning Objectives:**

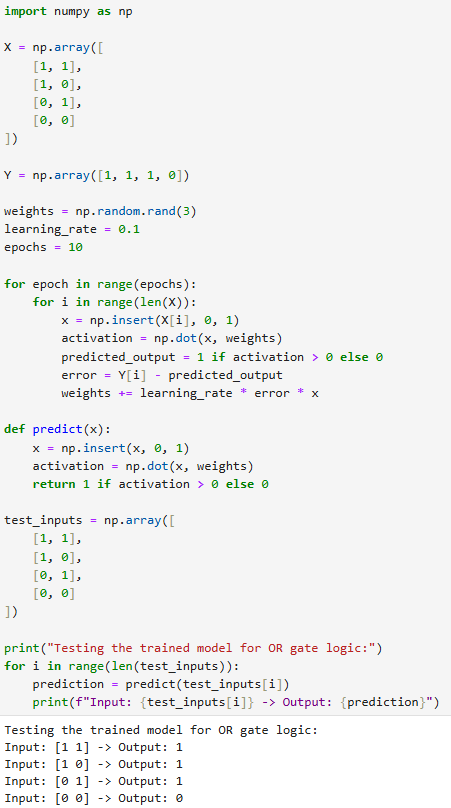
* To understand and implement the Single Layer Perceptron Learning Algorithm.
* To apply Perceptron Weight Adjustment for learning.
* To develop a simple logic gate classifier using SLP.

**Conclusion:**

The Single Layer Perceptron Learning Algorithm was successfully implemented to classify an OR gate. The perceptron was trained using a weight update rule based on the difference between predicted and actual values. After training, the model correctly classified the OR logic function. Since the OR gate is a linearly separable problem, the single-layer perceptron worked efficiently. However, the perceptron model is limited to linearly separable problems and cannot handle complex decision boundaries, such as XOR, which requires a multi-layer perceptron with backpropagation. Nonetheless, this experiment provided a foundational understanding of perceptron learning and its applications in basic classification tasks.



**Program and Output:**

****