|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **ANNFS** | **Semester:** | **VI** |
| **Date of Performance:** | **16/02/2022** | **Batch No:** | **B2** |
| **Faculty Name:** | **Dr. Jagannath Nirmal** | **Roll No:** | **1912052** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** |  |

**Experiment No: 4**

**Title: Implement Perceptron Learning Rule in ANN using MATLAB**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| Write a program to implement AND operation using perceptron neural network |

|  |
| --- |
| **COs to be achieved:** |
| **CO1:** Understand the role of neural networks in engineering, artificial intelligence, and cognitive modelling.  **CO3:** Develop concepts and techniques of neural networks through the study of the most important supervised and unsupervised neural network models. |

**Apparatus / Software tools used: VLAB**

|  |
| --- |
| **Theory:** |
| 1. **Perception learning rule:** 2. Link   http://vlabs.iitb.ac.in/vlabs-dev/labs/machine\_learning/labs/exp4/simulation.php  **Algorithm:**  1. Initialize weights and bias.  2. While stopping condition is false do  3. For each training pair do steps 4 to 6 .  4. Set activation of input unit xi = sj.  5. Compute the o/p with response    6. Weight & bias are updated if target o/p is not equal to actual o/p.  7. Test for stopping condition. |

|  |
| --- |
| **Block Diagram/Program flowchart:** |
|  |

|  |
| --- |
| **Algorithm:** |
| 1. Initialize the weights and the threshold. Weights may be initialized to 0 or to a small random value. 2. Initialize *i* as training set *D* 3. Calculate the actual output   )   1. 4.Compare with targeted output if 2. Update the weights:   w i ( t + 1 ) = w i ( t ) + ( d j − y j ( t ) ) x j , i {\displaystyle w\_{i}(t+1)=w\_{i}(t)+(d\_{j}-y\_{j}(t))x\_{j,i}}for all features 0≤ *i* ≤ *n*0 ≤ i ≤ n {\displaystyle 0\leq i\leq n}  Step 3 to 5 may be repeated until the y=t or error is less than user specified  **MATLAB Code:**  clc; clear all; close all;  x1 = [1 1 0 0]; x2 = [1 0 1 0]; t = [1 0 0 0]; alpha = 0.7; w = 1; w1=0.2; w2=0.2; b=0.2; for i = 1:4     yin = x1(i)\*w1 + x2(i)\*w2 + b\*w;     if yin>=0         y=1;     else         y = 0;     end     wnew1 = w1 + alpha\*x1(i)\*(t(i)-y);     wnew2 = w2 + alpha\*x2(i)\*(t(i)-y);     bnew = b + alpha\*w\*(t(i)-y);     w1 = wnew1;     w2 =wnew2;     b=bnew;     end  disp(w1) disp(w2) disp(b) |
|  |

|  |
| --- |
| **Observation:** |
| Attach program and its results after execution  **Command Window:**  -0.5000   0.2000   -0.5000 |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| 1. Flow chart to Implement OR Gate using Perceptron learning algorithm.      1. What is the building block of the perceptron?   A neural network is an interconnected system of perceptrons, so it is safe to say perceptrons are the foundation of any neural network. Perceptrons can be viewed as building blocks in a single layer in a neural network, made up of four different parts:   1. Input Values or One Input Layer 2. Weights and Bias 3. Net sum 4. Activation function |

|  |
| --- |
| **Conclusion:** |
| In this experiment, we have understood the perceptron learning algorithm by writing a program to implement AND operation in MATLAB  The perceptron learning algorithm is similar to the MC Pits model and this algorithm is more efficient which can also be used for non-binary data.  Link of feedback  <http://vlabs.iitb.ac.in/feedback/> |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |