|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Hope Foundation’s**  **Finolex Academy of Management and Technology, Ratnagiri** | | | | | |
| **Department of Computer Science and Engineering (AIML)** | | | | | |
| Subject name: Machine Learning | | | | | | Subject Code: CSL604 | |
| Class | | TE CSE | | | Semester –VI (CBCGS) | Academic year: 2024-25 | |
| Name of Student | |  | | | | **QUIZ Score :6** | |
| Roll No | |  | | Experiment No. | | 06 | |
| Title: **To implement McCulloch-Pitts model.** | | | | | | | |
|  | | | | | | | |
| **1. Lab objectives applicable:**  **LOB1:**To introduce platforms such as Anaconda, COLAB suitable to Machine Learning.  **LOB3:** To develop Neural Network based learning models. | | | | | | | |
| **2. Lab outcomes applicable:**  **LO1:**Implement various Machine learning models.  **LO3:** Implement Neural Network based models. | | | | | | | |
| **3. Learning Objectives:**   1. To implement AND gate using McCulloch-Pitts model. | | | | | | | |
| **4. Practical applications of the assignment/experiment:**  To get informed about how neural network model produces output for the given input. | | | | | | | |
| **5. Prerequisites**:   1. Python language | | | | | | | |
| **6. Minimum Hardware Requirements**:-  I series processor, RAM 4GB,  **7. Software Requirements:-**  Colab or Visual Studio or Jupyter notebook (Anaconda) | | | | | | | |
| **8. Quiz Questions :** [**https://docs.google.com/forms/d/e/1FAIpQLSdLNAkq0xovJMJQOIFIn5OwamznkxQmQLRXIHwE4p\_Pk\_Q4Gw/viewform?usp=dialog**](https://docs.google.com/forms/d/e/1FAIpQLSdLNAkq0xovJMJQOIFIn5OwamznkxQmQLRXIHwE4p_Pk_Q4Gw/viewform?usp=dialog) | | | | | | | |
| **9. Experiment/Assignment Evaluation:** | | | | | | | |
| **Sr. No.** | **Parameters** | | | | | **Marks obtained** | **Out of** |
| **1** | Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used - | | | | | 6 | 6 |
| **2** | Lab Performance | | | | |  | 2 |
| **3** | Punctuality | | | | |  | 2 |
| **Date of performance (DOP)** | | |  | | **Total marks obtained** |  | **10** |

**Signature of Faculty**

**10. Theory:**

**McCulloch-Pitts (M-P) Neuron Model**

The **McCulloch-Pitts neuron**, introduced in **1943** by Warren McCulloch and Walter Pitts, is a **mathematical model of a biological neuron**. It is a fundamental building block in **artificial neural networks (ANNs)**.

**Key Features of M-P Neuron**

1. **Binary Inputs & Outputs**
   * The inputs and outputs are either **0 or 1** (ON/OFF).
   * It mimics a simple neuron firing or not.
2. **Weighted Sum Calculation**
   * Each input is associated with a **weight**
   * The **weighted sum** is calculated as:
3. **Threshold Activation Function**
   * If S **exceeds a threshold value (θ)**, the neuron **fires (1)**; otherwise, it remains **inactive (0)**.
   * The activation function is:

**Gate Parameters:**

* **Weights:** [1, 1]
* **Threshold:** 2

**Iteration Details:**

**Input: (0, 0)**

Weighted Sum = 0 × 1 + 0 × 1 = 0

Comparison: 0 < 2 → Output = 0

**Input: (0, 1)**

Weighted Sum = 0 × 1 + 1 × 1 = 1

Comparison: 1 < 2 → Output = 0

**Input: (1, 0)**

Weighted Sum = 1 × 1 + 0 × 1 = 1

Comparison: 1 < 2 → Output = 0

**Input: (1, 1)**

Weighted Sum = 1 × 1 + 1 × 1 = 2

Comparison: 2 ≥ 2 → Output = 1

**OR Gate**

**Gate Parameters:**

* **Weights:** [1, 1]
* **Threshold:** 1

**Iteration Details:**

**Input: (0, 0)**

Weighted Sum = 0 × 1 + 0 × 1 = 0

Comparison: 0 < 1 → Output = 0

**Input: (0, 1)**

Weighted Sum = 0 × 1 + 1 × 1 = 1

Comparison: 1 ≥ 1 → Output = 1

**Input: (1, 0)**

Weighted Sum = 1 × 1 + 0 × 1 = 1

Comparison: 1 ≥ 1 → Output = 1

**Input: (1, 1)**

Weighted Sum = 1 × 1 + 1 × 1 = 2

Comparison: 2 ≥ 1 → Output = 1

**11. Installation Steps / Performance Steps and Results –**

def mcculloch\_pitts(inputs, weights, threshold):

    weighted\_sum = sum(i \* w for i, w in zip(inputs, weights))

    return 1 if weighted\_sum >= threshold else 0

# Define gates

AND\_GATE = {

    "weights": [1, 1],

    "threshold": 2

}

OR\_GATE = {

    "weights": [1, 1],

    "threshold": 1

}

# Test inputs

test\_cases = [(0, 0), (0, 1), (1, 0), (1, 1)]

print("AND Gate:")

for x1, x2 in test\_cases:

    output = mcculloch\_pitts([x1, x2], AND\_GATE["weights"], AND\_GATE["threshold"])

    print(f"Input: ({x1}, {x2}) -> Output: {output}")

print("\nOR Gate:")

for x1, x2 in test\_cases:

    output = mcculloch\_pitts([x1, x2], OR\_GATE["weights"], OR\_GATE["threshold"])

    print(f"Input: ({x1}, {x2}) -> Output: {output}")

**Output:**

AND Gate:

Input: (0, 0) -> Output: 0

Input: (0, 1) -> Output: 0

Input: (1, 0) -> Output: 0

Input: (1, 1) -> Output: 1

OR Gate:

Input: (0, 0) -> Output: 0

Input: (0, 1) -> Output: 1

Input: (1, 0) -> Output: 1

Input: (1, 1) -> Output: 1

**12. Learning Outcomes Achieved**

1. Students are able to perform implement McCulloch Pitts model.

**13. Conclusion:**

**1. Applications of the Studied Technique in Industry**

Although the MCP model itself is not widely used in modern applications, it serves as the foundation for artificial intelligence (AI) and machine learning (ML).

**2. Engineering Relevance**

The MCP model is crucial in **engineering and computing** as it provides insights into:

* **Neural Network Design**: Understanding how artificial neurons work before diving into deep learning.
* **Mathematical Modeling**: Demonstrates how simple mathematical rules can model complex decision-making.

**3. Skills Developed**

By implementing the MCP model, students and engineers gain programming skills, logical thinking skills, mathematical skills, and understanding of AI basics.

**14. References**:

1. Nathalie Japkowicz & Mohak Shah, ―Evaluating Learning Algorithms: A Classification Perspective‖, Cambridge.
2. Marc Peter Deisenroth, Aldo Faisal, Cheng Soon Ong, ―Mathematics for machine learning‖
3. Samir Roy and Chakraborty, ―Introduction to soft computing‖, Pearson Edition.
4. Ethem Alpaydın, ―Introduction to Machine Learning‖, MIT Press McGraw-Hill Higher

Education

1. Peter Flach, ―Machine Learning‖, Cambridge University Press
2. Tom M. Mitchell, ―Machine Learning‖, McGraw Hill
3. Kevin P. Murphy, ―Machine Learning ― A Probabilistic Perspective‖, MIT Press