****

# **(B. Tech.) Semester-VII AY 2023-24**

**DL Lab Assignment No. 04**

|  |  |
| --- | --- |
| **Student Name: Rohit Saini\_\_\_\_\_\_\_\_\_\_\_\_** | **PRN No.:1032200897\_** |
| **Date: 06-10-2023** | **Faculty: Prof. Anita Gunjal** |

**Problem Statement:** To study and implement the simple Neural Network for AND logic gate with Binary Input.

**Objectives:**

1. To understand the AND logic gate.
2. To study & implement different activation functions.
3. To implement the simple Neural Network.

**Theory:** (describe the following)

**Logic Gates (AND, OR, XOR):**

* **AND Gate:** Takes two binary inputs and outputs 1 (true) if both inputs are 1, otherwise outputs 0 (false).
* **OR Gate:** Takes two binary inputs and outputs 1 if at least one input is 1, otherwise outputs 0.
* **XOR Gate** (Exclusive OR): Takes two binary inputs and outputs 1 if the inputs are different, otherwise outputs 0.

**Simple Artificial Neural Network (ANN):**

* A simple ANN consists of interconnected artificial neurons or perceptrons.
* It typically includes an input layer, one or more hidden layers, and an output layer.
* Neurons in each layer receive input, apply a weighted sum, and pass it through an activation function.
* The network learns by adjusting the weights during training to minimize prediction errors.

Activation Functions

* **Sigmoid Function:**

S-shaped curve, maps inputs to values between 0 and 1.

Often used in the output layer of binary classification models.

* **ReLU (Rectified Linear Unit):**

Linear for positive inputs (output = input), zero for negative inputs.

Widely used in hidden layers of deep neural networks due to its simplicity and effectiveness.

* **Tanh (Hyperbolic Tangent):**

S-shaped curve, similar to the sigmoid but maps inputs to values between -1 and 1.

Used in some neural network architectures, especially when inputs are centered around zero.

* **Leaky ReLU:**

Similar to ReLU, but allows a small, non-zero gradient for negative inputs.

Addresses the "dying ReLU" problem and can lead to faster convergence.

* **Softmax:**

Maps a vector of inputs to a probability distribution over multiple classes.

Commonly used in the output layer of multi-class classification models.

**Operations to be performed:**

1. Import the required Python libraries
2. Define Activation Functions and plot its graphs: Step, Sigmoid, Tanh, ReLU, Softmax, etc Function
3. Initialize neural network parameters (weights, bias) and define model hyperparameters (number of iterations, learning rate)
4. Perform Forward Propagation
5. Perform Backward Propagation
6. Update weight and bias parameters
7. Train the learning model
8. Plot Loss value vs Epoch
9. Test the model performance

**Program code: (paste your program code)**

**Output: (paste output screen & graphs plotted)**

**FAQs:**

1. Define an Activation Function. Explain different activation functions with its mathematical importance and graphical representation.
2. State the significance of updating the weights during back propagation.
3. Explain the terms with the help of examples:
   1. neural network parameters (weights, bias)
   2. model hyperparameters (number of iterations, learning rate)
   3. gradient

**Conclusion:**

The features of gates were studied and the implementation of Simple ANN was performed successfully.

  