Course No.: CSE 2102
Course Title: Sessional based on CSE 2101
Experiment No. 3
Name of the Experiment: Design and implementation of Multi-layer Neural Networks algorithm (i.e., Back-propagation learning neural networks algorithm).

Course Outcomes: CO1
Learning Domain with Level: Cognitive (Applying, Analyzing, Evaluating & Creating)

Multilayer Perceptron Learning Algorithm

---

1. Initialise weights and thresholds

Set all weights and thresholds to small random values.

2. Present input and desired output

Present input $X_p = x_0, x_1, x_2, \ldots, x_{n-1}$ and target output $T_p = t_0, t_1, \ldots, t_{m-1}$ where n is the number of input nodes and m is the number of output nodes. Set w_0 to be $-\theta$, the bias, and x_0 to be always 1. For pattern association, X_p and T_p represent the patterns to be associated. For classification, T_p is set to zero except for one element set to 1 that corresponds to the class that X_p is in.

3. Calculate actual output

Each layer calculates

$$y_{pj} = f\left[\sum_{i=0}^{n-1} w_i x_i\right]$$

and passes that as input to the next layer. The final layer outputs values o_{pj}.

4. Adapt weights

Start from the output layer, and work backwards.

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_{pj} o_{pj}$$

$w_{ij}(t)$ represents the weights from node i to node j at time t, η is a gain term, and δ_{pj} is an error term for pattern p on node j.

For output units

$$\delta_{pj} = k o_{pj}(1 - o_{pj})(t_{pj} - o_{pj})$$

For hidden units

$$\delta_{pj} = k o_{pj}(1 - o_{pj}) \sum_k \delta_{pk} w_{jk}$$

where the sum is over the k nodes in the layer above node j.

Your task is to:

- **Design** a program for your dataset
- **Analyze** the program
- **Evaluate** the correctness of the program and the accuracy of the algorithm.
- **Apply** Multi-layer Neural Networks algorithm to solve XOR problem.