Heaven's Light is Our Guide
Computer Science & Engineering
Rajshahi University of Engineering & Technology

Course No: CSE 4204
Course Name: Sessional based on CSE 4203

Experiment No: 3

Name of the Experiment: Design and implementation of Multi-layer Neural Networks algorithm (i.e., Back-propagation learning neural networks algorithm).

Submitted to

Rizoan Toufiq
Assistant Professor
Dept. of Computer Science & Engineering
Rajshahi University of Engineering & Technology

Submitted by

Mondol Mridul Provakar
Roll: 1803062
Dept. of Computer Science & Engineering
Rajshahi University of Engineering & Technology

Date of Submission: 02/12/2023

Contents

1 Breast Cancer Dataset

Breast cancer is the most common cancer amongst women in the world. It accounts for 25% of all cancer cases, and affected over 2.1 Million people in 2015 alone. It starts when cells in the breast begin to grow out of control. These cells usually form tumors that can be seen via X-ray or felt as lumps in the breast area. The key challenges against it's detection is how to classify tumors into malignant (cancerous) or benign(non cancerous). The dataset has following characteristics:

- **Size:** The dataset consists of a total of 569 rows and 6 columns.
- **Data Types:** All values are numerical.
- **Features:** Mean_radius, Mean_texture, Mean_perimeter, Mean_area, Mean_smoothness.
- **Target Variable:** Diagnosis.

1.1 Exploratory Data Analysis

In a correlation heatmap, the values displayed within the heatmap represent the correlation coefficients between pairs of variables. These correlation coefficients quantify the strength and direction of the linear relationship between two variables. The values in a correlation heatmap typically range between -1 and +1:

- +1: A correlation coefficient of 1 indicates a perfect positive linear relationship between the variables. This means that as one variable increases, the other variable also increases proportionally.
- 0: A correlation coefficient of 0 indicates no linear relationship between the variables. There's no systematic linear pattern in their relationship.
- -1: A correlation coefficient of -1 indicates a perfect negative linear relationship between the variables. This means that as one variable increases, the other variable decreases proportionally.

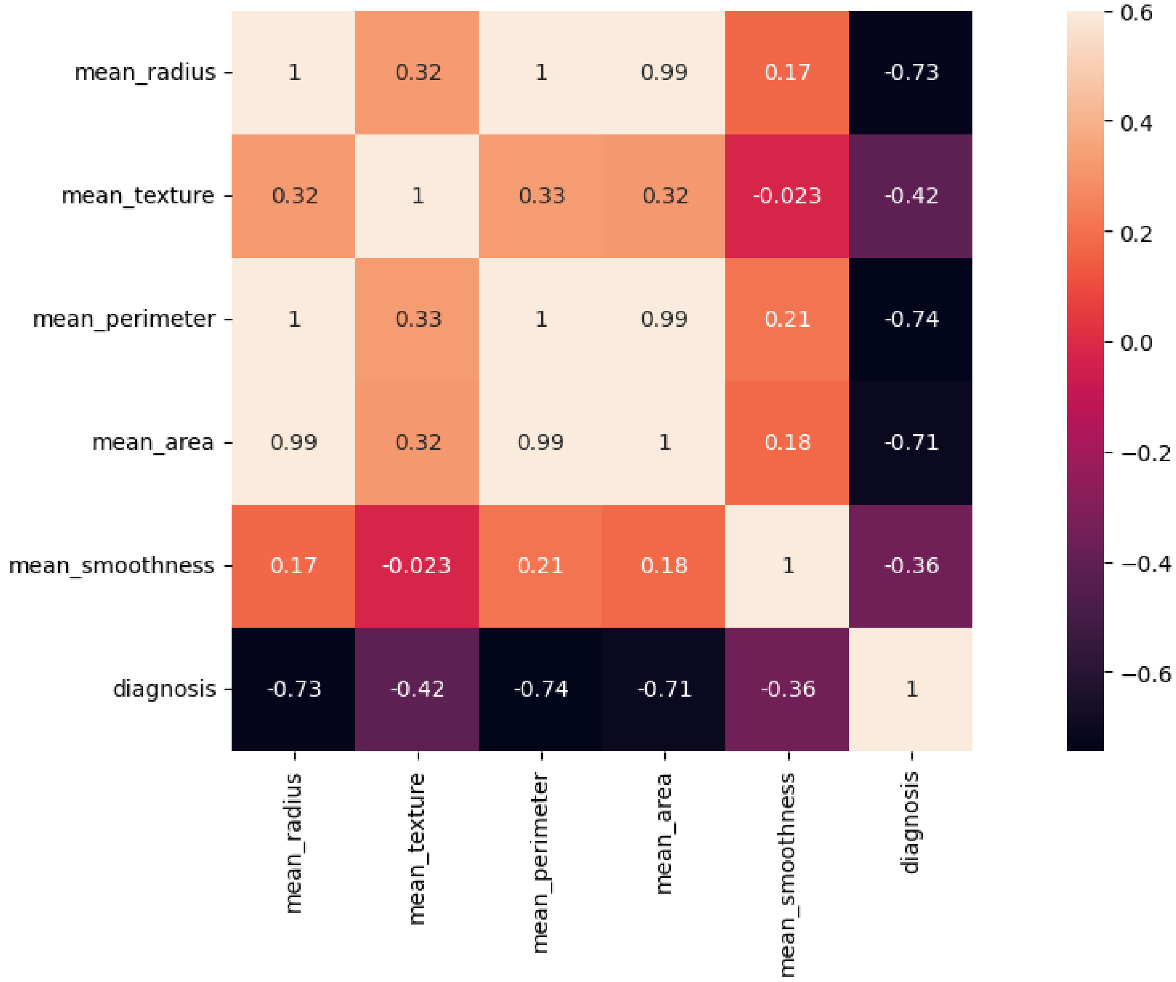

Figure 1: Correlation Heatmap

- **Correlation Heatmap:** The above correlation heatmap was generated to visualize the correlations between continuous attributes. It was found out that there were high correlations among the features: Mean_perimeter and Mean_area. It was seen that Mean_perimeter was correlated with Mean_radius by 1 and Mean_area was correlated with Mean_radius by 0.99. So these two columns were removed from the dataset.

  Final Dataset Size: 569 rows and 4 columns.

1.2 Training and Test Dataset Ratio

In this analysis, the dataset is divided into a training set and a test set with an 80/20 split. This means that 80% of the data is used for training the single layer perceptron algorithm, while the remaining 20% is reserved for testing and evaluating the classifier's performance.
After train & test split:

Training dataset size: 455 rows and 4 columns.
Test dataset Size: 114 rows and 4 columns.

2 Multi-Layer Perceptron Algorithm

2.1 The multi-layer perceptron learning algorithm

1. Initialise weights and threshold
   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 desired 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 t o be always l. 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 t o the next layer. The final layer outputs values o_{pj}

4. Adapt weights

   Start from the output layer, and work backwards.
   $$w_i(t+1) = w_i(t) + \eta \delta_{pj} o_{pj}$$
   $w_{ij}(t)$ 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
   $$\delta_{pj} = k o_{pj}(1 - o_{pj}) \sum_k \delta_{pk} w_{pk}$$

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

2.2 Model Evaluation

The multi-layer perceptron learning algorithm was implemented and the accuracy score & the confusion matrix was calculated.

- **Accuracy Score:** 93.86%
- **Confusion Matrix:**

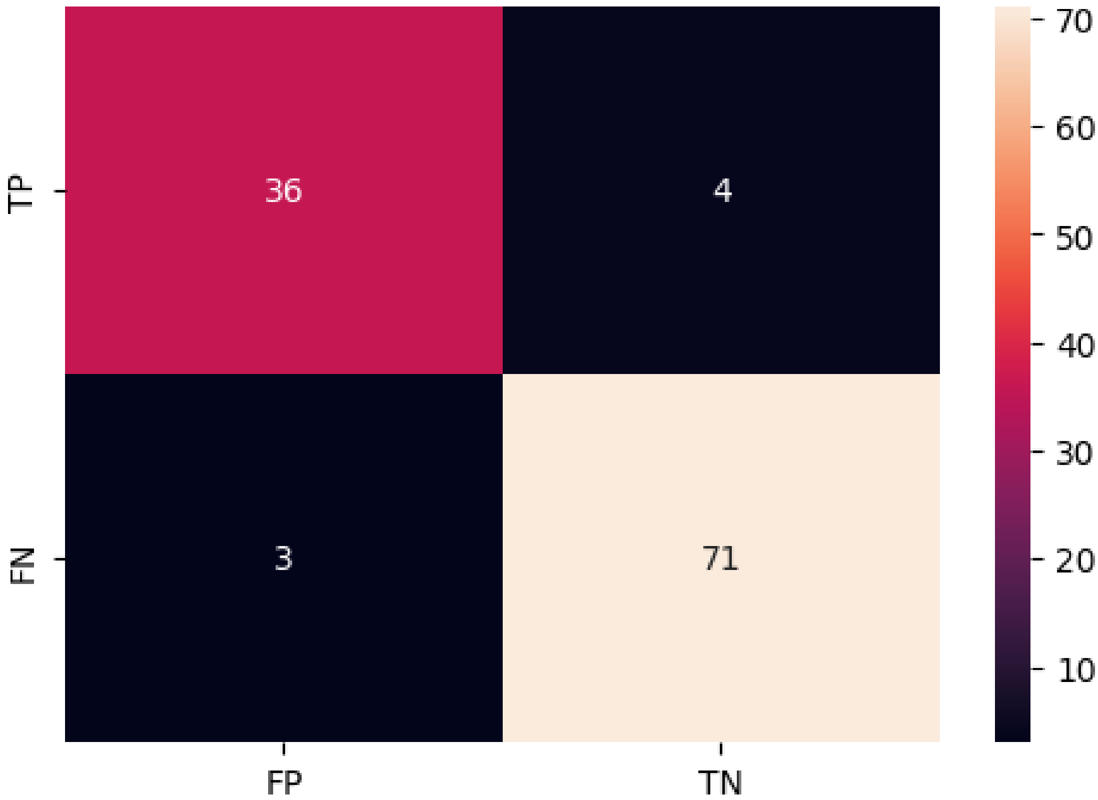

Figure 2: Confusion Matrix

- **Discussion:** The above confusion matrix consists of four different metrics:
- True Positives (TP): The number of samples correctly predicted as positive. Here 36 samples are predicted as true positive. It means they had breast cancer and the model predicted them as positive.
- True Negatives (TN): The number of samples correctly predicted as negative. Here 71 samples are predicted as true negative. It means they did not have breast cancer and the model predicted them as negative.
- False Positives (FP): The number of samples incorrectly predicted as positive. Here 3 samples are predicted as false positive. It means they did not have breast cancer and the model predicted them as positive.
- False Negatives (FN): The number of samples incorrectly predicted as negative. Here 4 samples are predicted as false negative. It means they had breast cancer and the model predicted them as negative.

3 Multi-Layer Perceptron for XOR Problem

The XOR problem is a classic example that showcases the limitations of a single-layer perceptron. A single-layer perceptron cannot learn the XOR function due to its linear nature. To overcome this limitation, multi-layer perceptron can be used. The above MLP model is applied to solve the Xor Problem and got an accuracy of 100%.

3.1 Problem Overview

The XOR problem involves binary inputs (0 or 1) and a binary output. The task is to learn a mapping from the input pairs to the correct output. The XOR function outputs 1 only when the inputs are different ($0 \oplus 1$ or $1 \oplus 0$).

3.2 Truth Table

X_1	X_2	XOR Output
0	0	0
0	1	1
1	0	1
1	1	0

3.3 Output

Accuracy: 100.00%
Input: [0 0] - Predicted Output: 0.0
Input: [0 1] - Predicted Output: 1.0
Input: [1 0] - Predicted Output: 1.0
Input: [1 1] - Predicted Output: 0.0

3.4 Visualizing The Decision Boundary

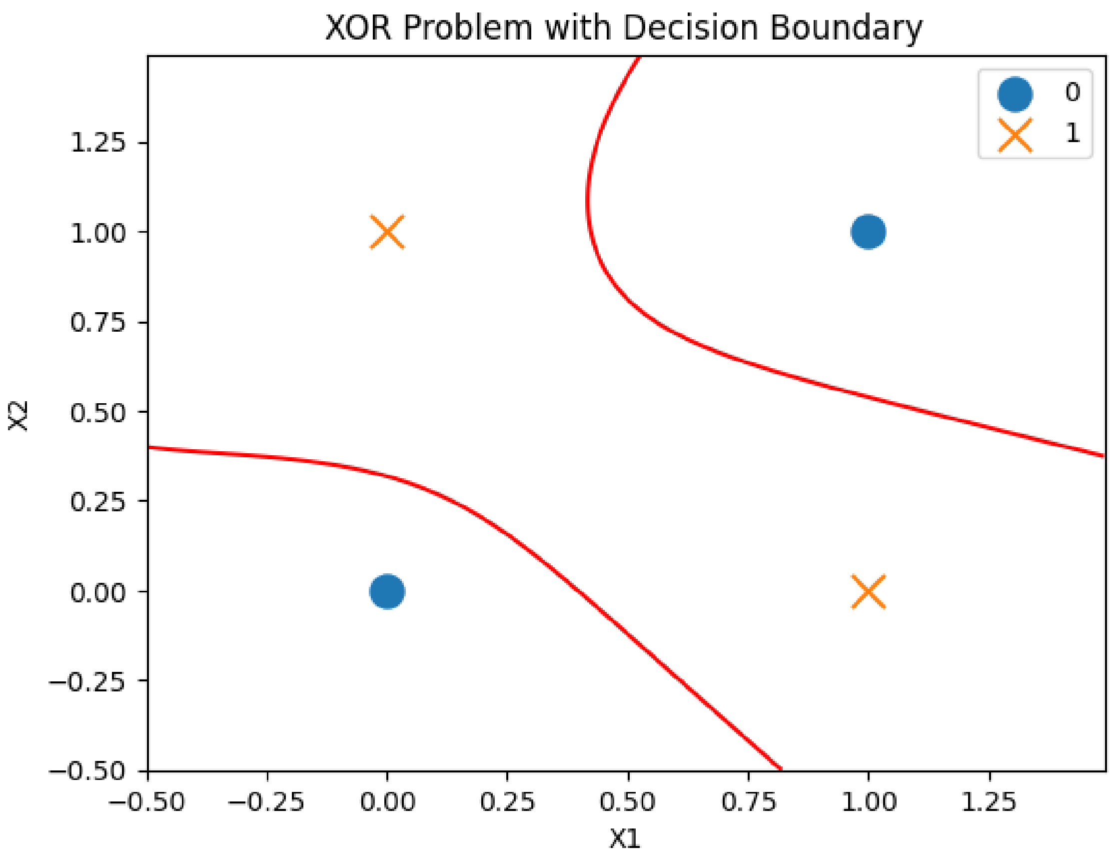

Figure 3: Xor Problem with Decision Boundary

4 Conclusion

- Explored dataset using Exploratory Data Analysis techniques to understand its structure and relationships.
- Developed a Multi-Layer Neural Network (MLP) with appropriate architecture.
- Trained the MLP model using specified parameters (epochs, learning rate) on the dataset. Accuracy was 93.86%.
- Successfully solved the XOR problem, achieving a 100% accuracy rate.

References

[1] R. Beale and T. Jackson, *Neural Computing: An introduction.*, CRC Press, 1990

[2] Breast Cancer Dataset, [Online]. Available at: https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset

[3] Source Code, [Online]. Available at: https://colab.research.google.com/drive/124Lc4-a1_xR1u_KmKho_Ov0nPl0yr_xY#scrollTo=gGee9Lnlic4o