Documentation for Experiment-2

Name-Aditya Prasad Kharwar
Roll no-2K22/CO/32

1. Introduction

This notebook demonstrates the implementation of a simple neural network using TensorFlow and Keras to classify linearly separable and non-linearly separable datasets.

2. Libraries Used

The following Python libraries are used in this project:

- `numpy` – For numerical operations
- `matplotlib.pyplot` – For visualization
- `sklearn.datasets` – To generate synthetic datasets (`make_blobs`, `make_circles`)
- `tensorflow.keras` – To build and train the neural network

3. Dataset Preparation

Two types of datasets are generated:

- **Linearly separable dataset** using `make_blobs`
- **Non-linearly separable dataset** using `make_circles`
  Both datasets are plotted for visualization.

4. Neural Network Model

A simple feedforward neural network is implemented using the **Sequential** model in Keras. It consists of:

- **One dense layer** with a `sigmoid` activation function
- **Binary cross-entropy loss function** for classification
- **Adam optimizer** with a learning rate of 0.01

5. Model Training and Evaluation

- The model is trained for **50 epochs** using a batch size of **10**.
- Performance is evaluated using **accuracy and loss metrics**.

6. Results

- The model successfully classifies the linearly separable dataset.
- For non-linearly separable data, a **more complex architecture** (e.g., adding hidden layers) may be needed for better performance.

7. Conclusion

This experiment shows how a simple neural network can classify data. It highlights the importance of dataset properties and model architecture in machine learning.