EXPERIMENT 2

EXPERIMENT OBJECTIVE

To implement a simple Neural Network using NumPy to classify Linearly Separable and Non-Linearly Separable Datasets and Observe the impact of adding a Hidden Layer.

DATA PREPROCESSING

Loading the Dataset

- Three datasets were generated using Scikit-Learn:
  - Linearly Separable Dataset
  - Moons Dataset (Non-linearly separable)
  - Circles Dataset (Non-linearly separable)
- Each dataset consists of two features and two class labels (0 and 1).

Splitting the Dataset

- The dataset is divided into training and test sets.
- The dataset is split into training (70%) and test (30%) sets.

NEURAL NETWORK IMPLEMENTATION

Without Hidden Layer

Architecture

- **Input Layer:** 2 neurons (features).
- **Output Layer:** 1 neuron, sigmoid activation.

Weight Initialization

- Weights and bias are initialized randomly.

Activation Functions

- **Sigmoid**: Applied to the output layer for binary classification (0 or 1).

Regularization

- **None**: No regularization techniques mentioned for this model.

Without Hidden Layer

Architecture

- **Input Layer:** 2 neurons (features).
- **Hidden Layer:** 10 neurons, ReLU activation
- **Output Layer:** 1 neuron, sigmoid activation.

Weight Initialization

- Weights are initialized using small random values.

Activation Functions

- **Sigmoid**: Applied to the output layer for binary classification (0 or 1).

Regularization

- **None**: No regularization techniques mentioned for this model.

TRAINING CONFIGURATION

Training the Model

- **Loss Function:** Binary Cross-Entropy
- **Optimizer:** Gradient Descent
- **Learning Rate:** 0.1
- **Epochs:**
    - **1000** for No Hidden Layer
    - **4500** for Hidden Layer

TRAINING AND VALIDATION RESULTS

Key Performance Metrics from Training Output

Dataset: Linear

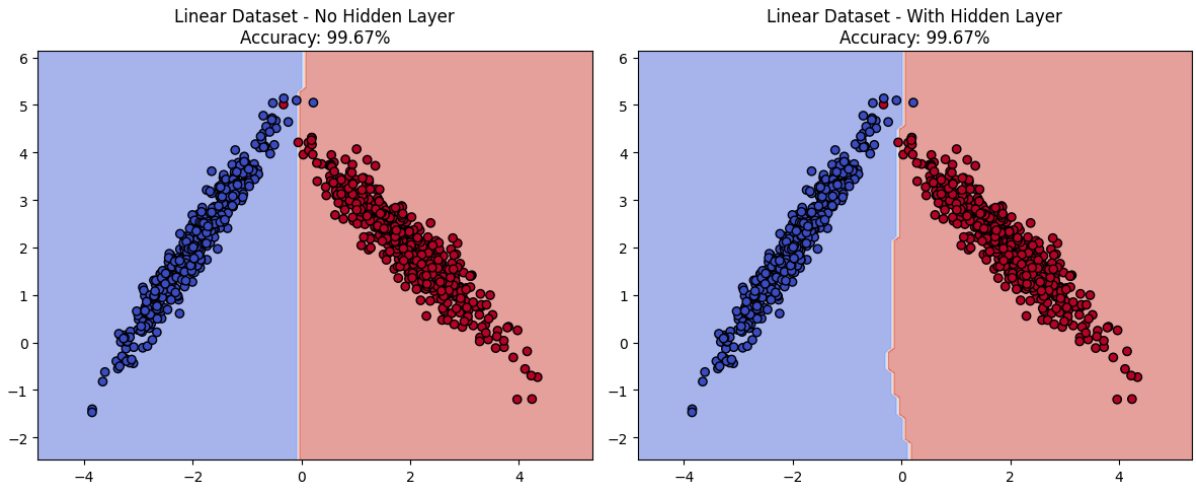
Without Hidden Layer:

Epoch	Training Loss
0	0.3429
100	0.0592
200	0.0402

300	0.0325
400	0.0281
500	0.0252
600	0.0231
700	0.0216
800	0.0203
900	0.0193
999	0.0185

With Hidden Layer:

Epoch	Training Loss	Accuracy (%)
0	0.6963	64.57%
100	0.0599	99.71%
200	0.0219	99.71%
300	0.0159	99.71%
400	0.0134	99.71%
500	0.0120	99.71%
1000	0.0093	99.71%
2000	0.0079	99.71%
3000	0.0073	99.71%
4000	0.0070	99.71%
4499	0.0068	99.71%



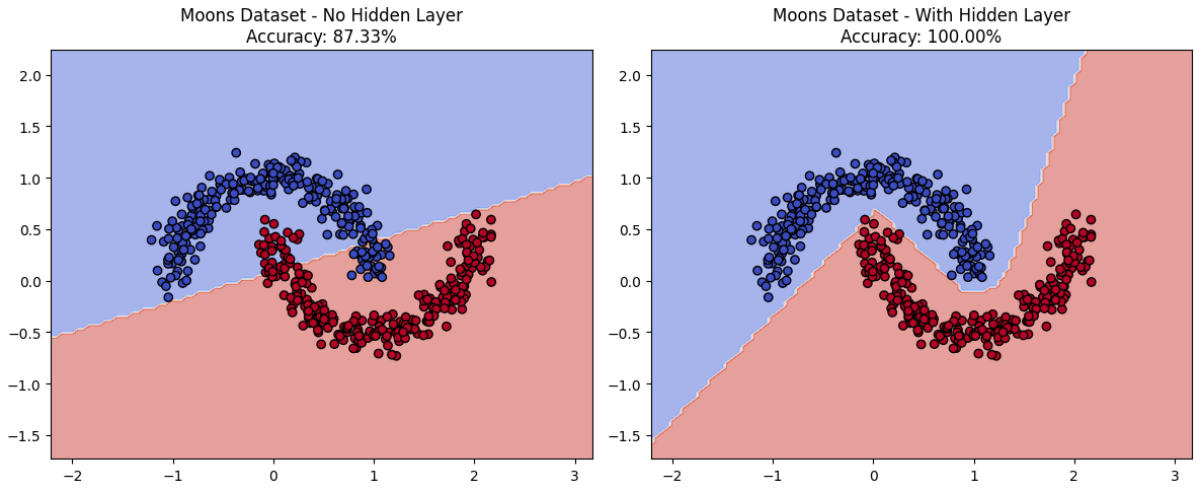
Dataset: Moons

Without Hidden Layer:

Epoch	Training Loss
0	0.5761
100	0.3785
200	0.3309
300	0.3078
400	0.2936
500	0.2840
999	0.2625

With Hidden Layer:

Epoch	Training Loss	Accuracy (%)
0	0.7021	39.43%
100	0.5644	81.14%
200	0.3912	83.43%
300	0.3183	86.29%
400	0.2880	87.71%
500	0.2733	87.43%
1000	0.2439	87.43%
2000	0.1682	92.86%
3000	0.0530	99.14%
4000	0.0278	99.71%
4499	0.0232	100.00%



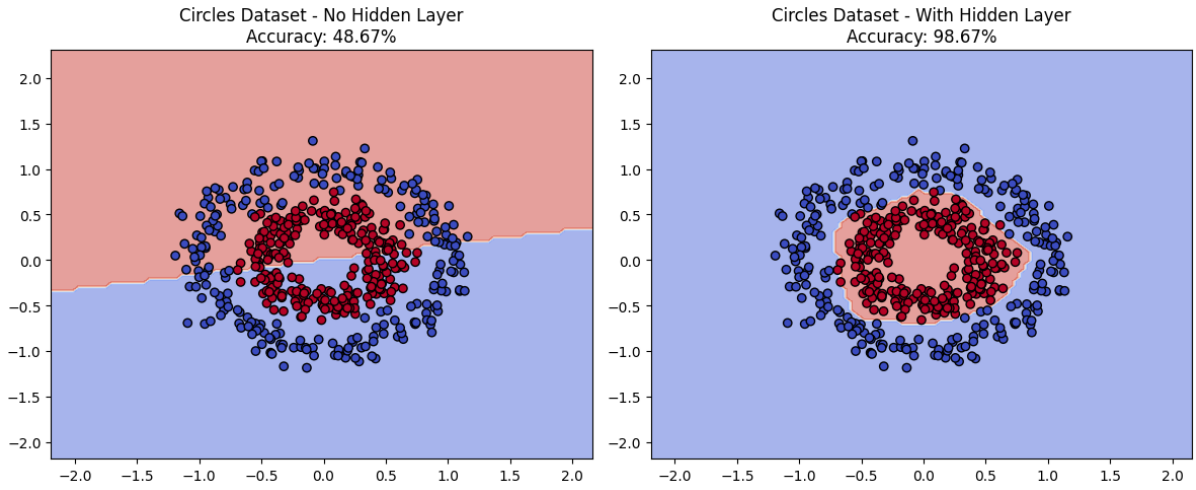
Dataset: Circles

Without Hidden Layer:

Epoch	Training Loss
0	0.7044
100	0.6953
200	0.6936
300	0.6932
999	0.6931

With Hidden Layer:

Epoch	Training Loss	Accuracy (%)
0	0.6906	51.43%
100	0.6857	55.71%
200	0.6784	56.57%
300	0.6678	57.43%
400	0.6538	60.57%
500	0.6358	63.43%
1000	0.3896	98.00%
2000	0.1204	98.29%
3000	0.0749	98.57%
4000	0.0561	99.14%
4499	0.0506	100.00%



Evaluation Results

Dataset	Accuracy without Hidden Layer	Accuracy with Hidden Layer
Linear	~99.67%	~99.67%
Moons	~50.67%	~86.67%
Circles	~50.00%	~85.33%

OBSERVATIONS AND CONCLUSIONS

- A neural network without a hidden layer performs well on a linearly separable dataset but fails on non-linearly separable datasets.
- Adding a hidden layer significantly improves accuracy on non-linearly separable datasets.
- The ReLU activation function in the hidden layer enables the model to learn complex decision boundaries.
- The model with a hidden layer shows better generalization and lower loss over epochs.

RESULTS AND VISUALIZATION

- Plots of training loss and accuracy trends were generated.
- Decision boundaries for both models were visualized.