NAME: DIGANTH R GOWDA
USN: 1RUA24CSE0135

Ex No: 2 Date: 2\2\26	**Building and Training a Neural Network for Planar Data Classificationn**

Objective:

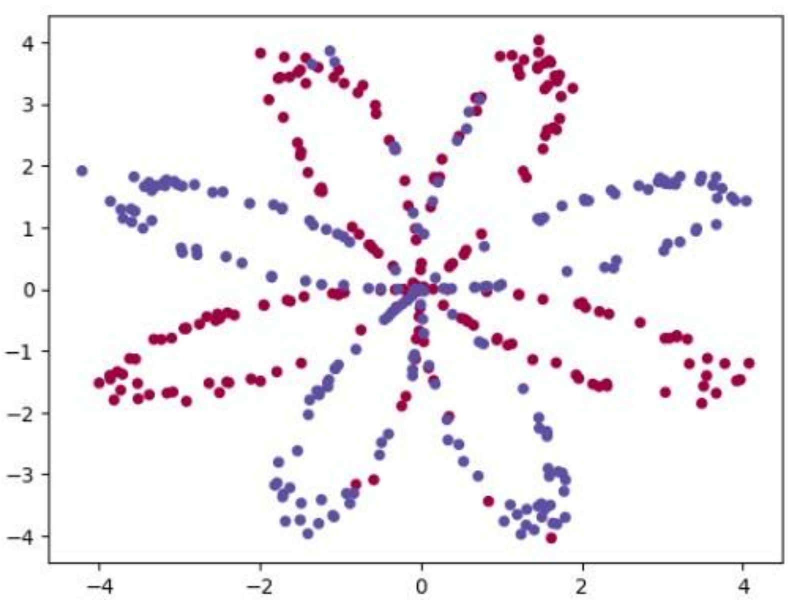
To build and train a **binary classification neural network with one hidden layer** from scratch using **gradient descent**, and to compare its performance with **logistic regression** on a **non-linearly separable planar dataset**.

Descriptions:

Binary classification is the task of classifying data into two distinct classes.
In this experiment, we classify **2D planar data points** into two categories:

- **1 → Class 1 (Blue points)**
- **0 → Class 0 (Red points)**

The dataset used is a **flower-shaped planar dataset**, where the two classes are **not linearly separable**.
This makes it unsuitable for simple logistic regression and ideal for demonstrating the power of **neural networks with hidden layers**.



Two models are implemented and compared:

NAME: DIGANTH R GOWDA
USN: 1RUA24CSE0135

1. Logistic Regression

- **No hidden layers**
- **One output neuron**
- **Linear decision boundary**
- **Uses sigmoid activation**

2. Neural Network (One Hidden Layer)

- **One hidden layer with 4 neurons**
- **Tanh activation in hidden layer**
- **Sigmoid activation in output layer**
- **Non-linear decision boundary**

The neural network is trained using forward propagation, backward propagation, and gradient descent optimization to minimize the cross-entropy loss.

Dataset Description

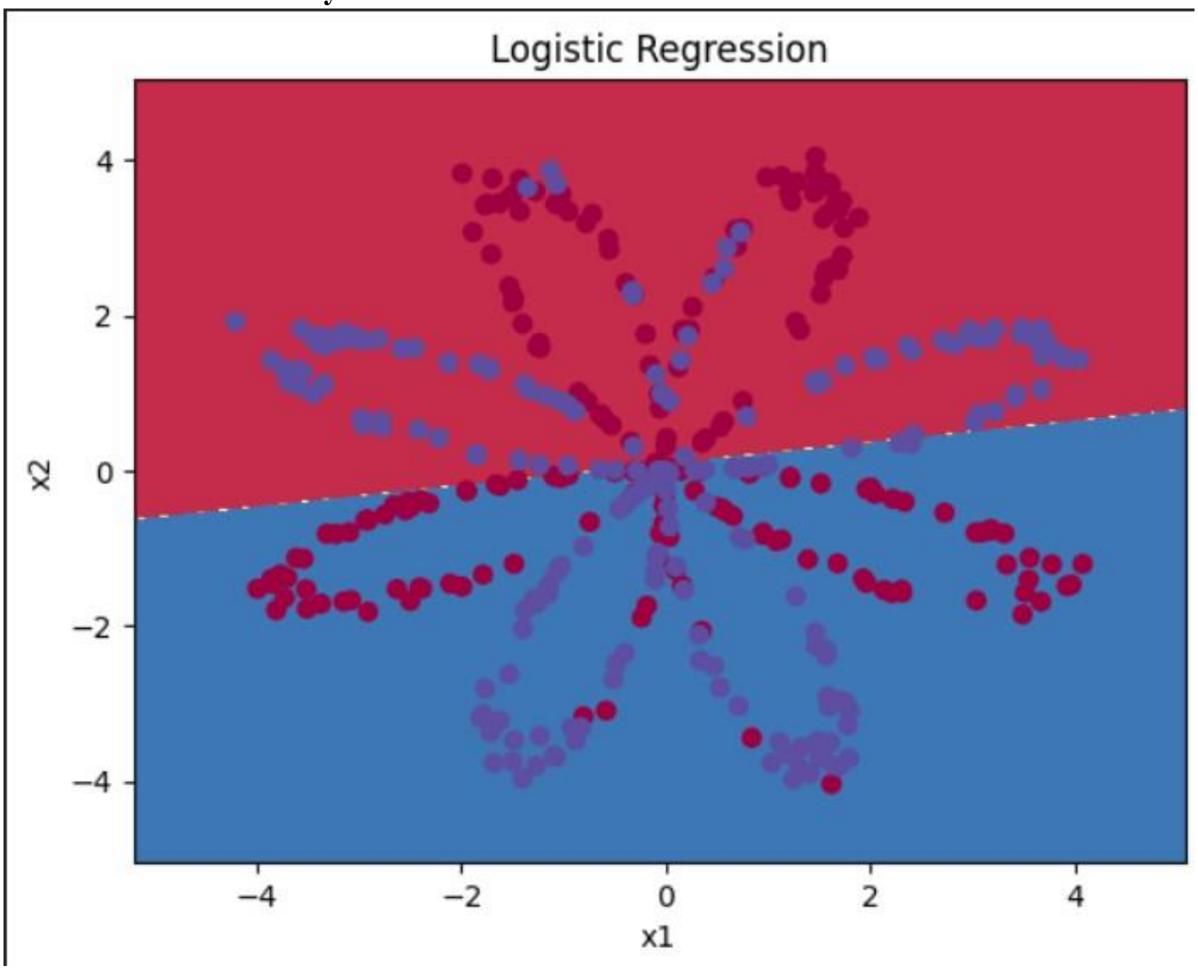
- **The dataset consists of 400 data points**
- **Each data point has 2 input features (x_1, x_2)**
- **Labels are binary (0 or 1)**
- **Data distribution forms a flower-like pattern**
- **Dataset is generated programmatically using** `load_planar_dataset()`

NAME: DIGANTH R GOWDA
USN: 1RUA24CSE0135

Model Architecture

Logistic Regression Model

- **Input Layer: 2 neurons**
- **Output Layer: 1 neuron**
- **Activation Function: Sigmoid**
- **Limitation: Can only learn linear boundaries**



Neural Network Model

- **Input Layer: 2 neurons**
- **Hidden Layer: 4 neurons**

NAME: DIGANTH R GOWDA
USN: 1RUA24CSE0135

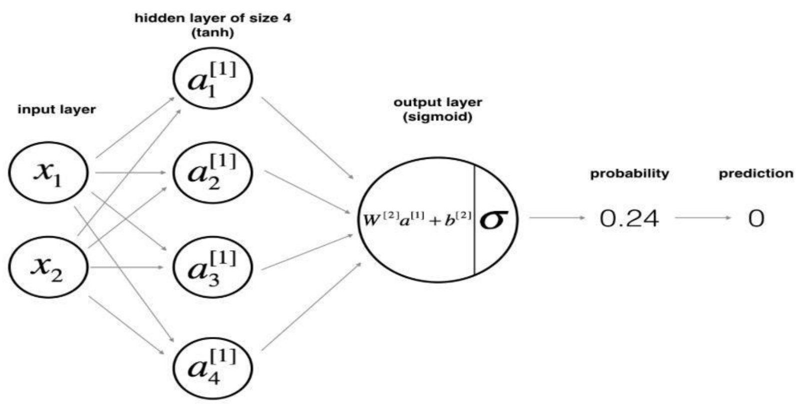
- **Output Layer: 1 neuron**

Activation Functions:

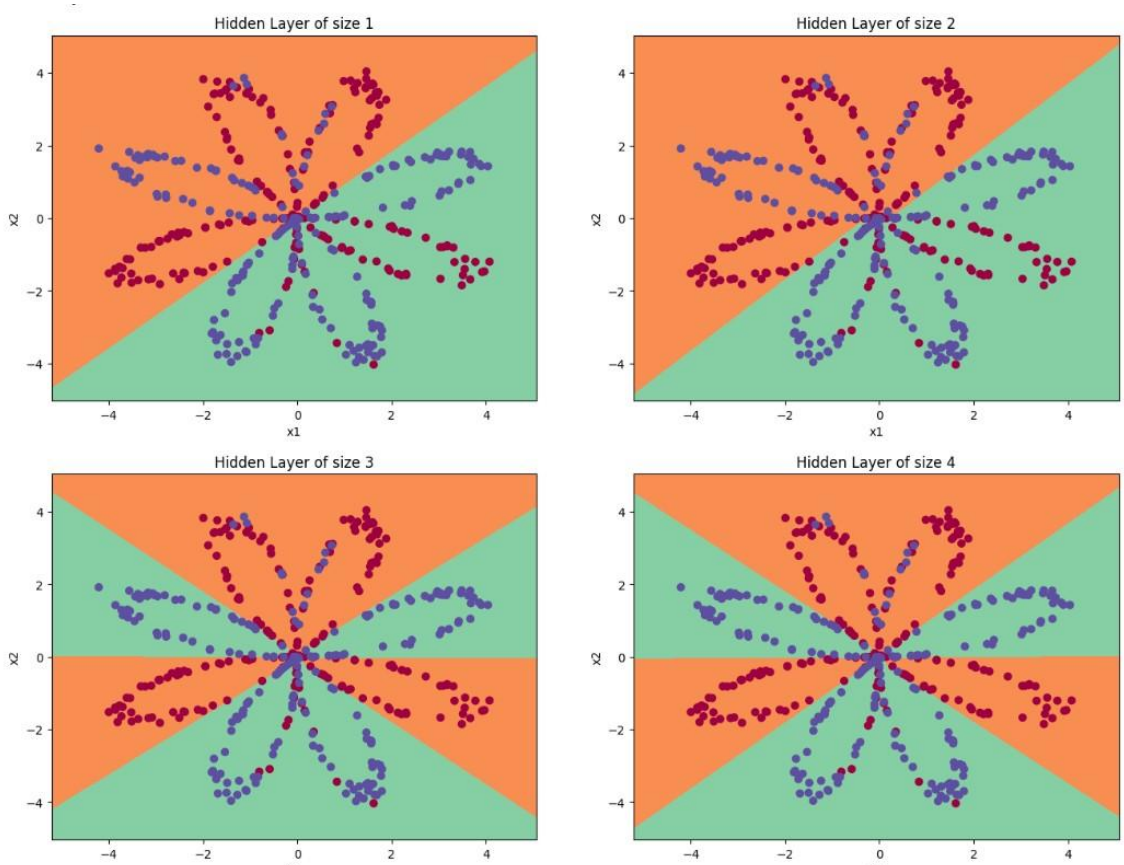
- **Hidden Layer →** tanh
- **Output Layer →** sigmoid

4 - Neural Network model

Logistic regression did not work well on the "flower dataset". You are going to train a Neural Network with a single hidden layer.



Here is our model:



NAME: DIGANTH R GOWDA
USN: 1RUA24CSE0135

Mathematical Model

Forward Propagation

Hidden layer:

$$Z_1 = W_1 X + b_1$$

$$A_1 = \tanh(Z_1)$$

Output layer:

$$Z_2 = W_2 A_1 + b_2$$

$$\hat{y} = \sigma(Z_2)$$

Loss Function (Cross-Entropy Loss)

$$J = -\frac{1}{m} \sum [y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$$

Backward Propagation

$$dZ_2 = A_2 - Y$$

$$dW_2 = \frac{1}{m} dZ_2 A_1^T$$

$$dZ_1 = W_2^T dZ_2 \cdot (1 - A_1^2)$$

$$dW_1 = \frac{1}{m} dZ_1 X^T$$

Algorithm Steps

1. Data Loading

- **Load planar dataset using helper function**
- **Visualize dataset using scatter plot**

NAME: DIGANTH R GOWDA
USN: 1RUA24CSE0135

2. Logistic Regression

- **Train baseline logistic regression model**
- **Plot decision boundary**
- **Compute accuracy**

3. Neural Network Construction

- **Initialize parameters**
- **Perform forward propagation**
- **Compute cost**
- **Apply backward propagation**
- **Update parameters using gradient descent**

4. Training

- **Train for 10,000 iterations**
- **Monitor cost reduction**

5. Evaluation

- **Predict output labels**
- **Calculate accuracy**
- **Plot decision boundaries**
- **Compare results with logistic regression**

NAME: DIGANTH R GOWDA
USN: 1RUA24CSE0135

Results

Logistic Regression

- **Accuracy ≈ 47%**
- **Decision boundary is almost linear**
- **Fails to separate flower-shaped data**

Neural Network

- **Accuracy ≈ 90%**
- **Decision boundary follows complex flower shape**
- **Successfully classifies non-linear data**

Inference

- **Logistic regression is insufficient for non-linearly separable data.**
- **Adding a hidden layer with non-linear activation enables the model to learn complex patterns.**
- **Neural networks significantly outperform logistic regression for complex datasets.**
- **Tanh activation helps in faster convergence and better learning.**

Conclusion

In this experiment, a neural network with one hidden layer was successfully implemented from scratch.
The model achieved high accuracy (~90%) on a complex planar dataset, clearly demonstrating the importance of hidden layers and non-linear activation functions in machine learning.

This experiment highlights how neural networks overcome the limitations of linear models and are effective for solving real-world classification problems.

NAME: DIGANTH R GOWDA
USN: 1RUA24CSE0135

GitHub link
https://github.com/diganthrgowdabtech24-star/mlds-lab-2