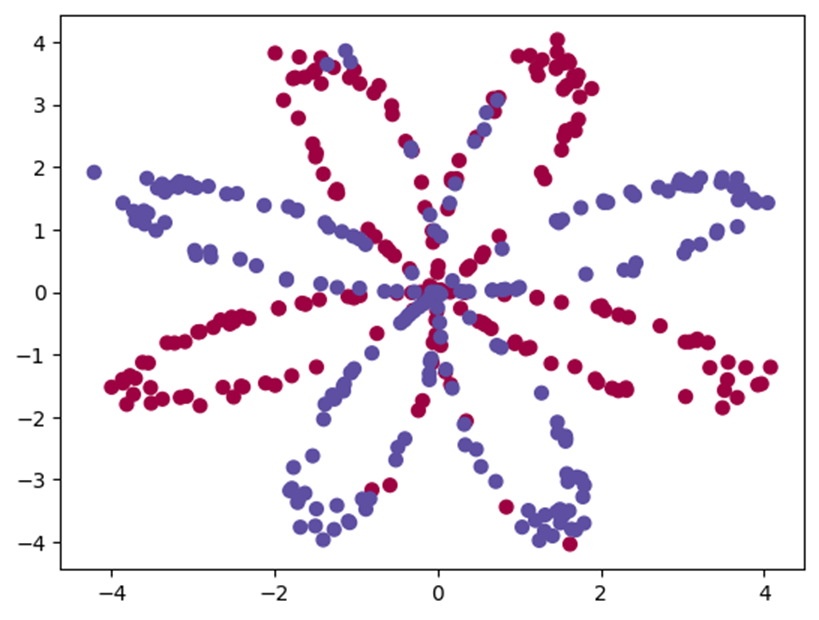
| **Ex No: 2**  **Date: 14-08-24** | **Planar Data classification with a Single Hidden layer Neural Network** |
| --- | --- |

**Objective:**

To implement a 2-class classification neural network with a single hidden layer, and compare its performance to a logistic regression model.

**Descriptions:**

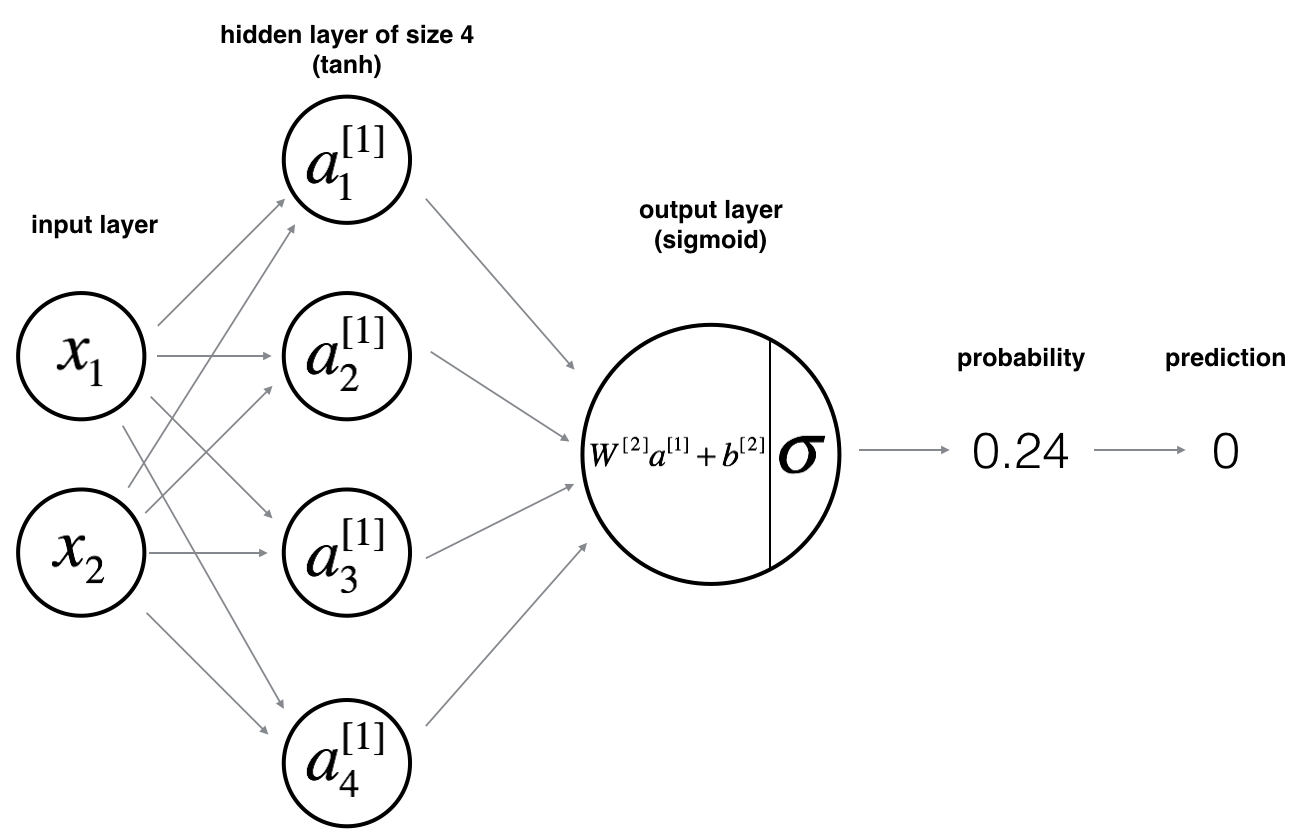
The data looks like a "flower" with some red (label y=0) and some blue (y=1) points. Your goal is to build a model to fit this data.



The choice of a neural network with a non-linear activation function, such as tanh, over a linear model like logistic regression is driven by the nature of the dataset. The "flower" pattern is not linearly separable, meaning a straight line cannot effectively separate the red and blue points.

A neural network with a non-linear activation function offers the flexibility to learn complex, non-linear decision boundaries. This capability allows the network to capture intricate patterns and relationships in the data, leading to improved classification performance. By employing a non-linear model, the network can adjust its decision boundary to fit the curved or complex patterns of the dataset, thereby enhancing its predictive accuracy.

**Model:**

https://piehqfpkotypaenhpraenx.coursera-apps.org/notebooks/Week%202/Logistic%20Regression%20as%20a%20Neural%20Network/images/LogReg_kiank.pnghttps://piehqfpkotypaenhpraenx.coursera-apps.org/notebooks/Week%202/Logistic%20Regression%20as%20a%20Neural%20Network/images/LogReg_kiank.png****

Mathematically:

For one example x(i) :

z[1](i) =W[1]x(i)+b[1](1)

a[1](i)=tanh(z[1](i))(2)

z[2](i)=W[2]a[1](i)+b[2](3)

y^(i)=a[2](i)=σ(z[2](i))(4)

y(i)prediction={10if a[2](i)>0.5otherwise (5)

Given the predictions on all the examples, you can also compute the cost J as follows:

J=−1m∑i=0m(y(i)log(a[2](i))+(1−y(i))log(1−a[2](i)))(6)

**Building the parts of algorithm**

The main steps for building a Neural Network are:

1. Loading the dataset
2. Implementing logistic regression
3. Train Neural network with single hidden layer

· Defining neural network structure

· Initialize model parameter

· Loop – forward propagation

Compute cost

Backward propagation

Update parameter

· Integrate above in nn models

· Predictions

· Tuning hidden layer size

4. Performance on other dataset.

**GitHubLink:** [**https://github.com/chandanab1/Deep\_Learning**](https://github.com/chandanab1/Deep_Learning)