**VANDANA M**

**1RVU22BSC107**

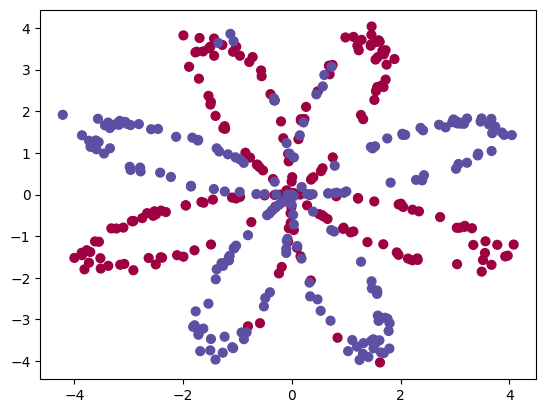
|  |  |  |
| --- | --- | --- |
| **Ex No: 2** | **Date: 14|08|2024** | **Planar Data Classification with a Single Hidden Layer Neural Network** |

**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 dataset consists of points in a two-dimensional space, arranged in a pattern that resembles a "flower." The points are colored to indicate their class labels: red points (label y=0) and blue points (label y=1).



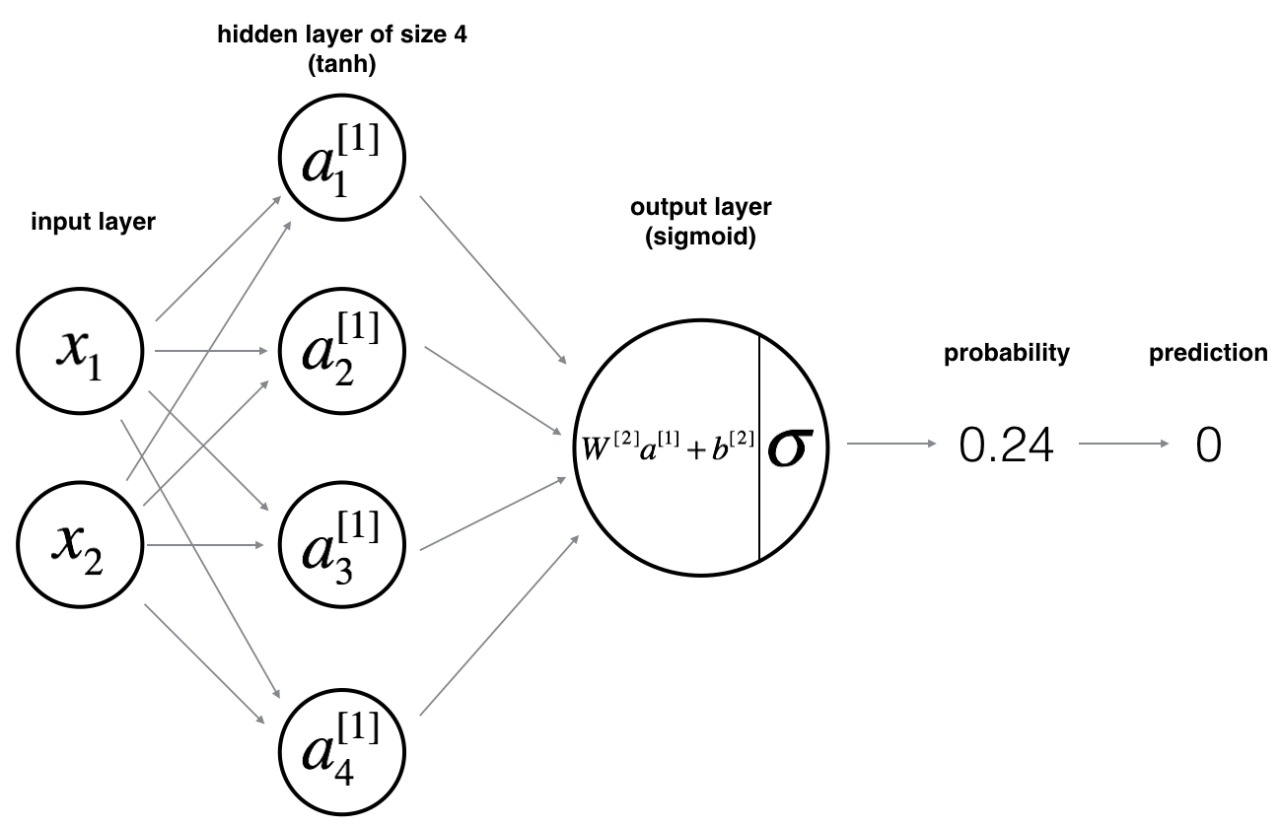
The choice of a neural network with a non-linear activation function, such as tanh, over a linear model like logistic regression is driven by the nature of the dataset. The "flower" pattern is not linearly separable, meaning a straight line cannot effectively separate the red and blue points.

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.

**VANDANA M**

**1RVU22BSC107**

**Model:**



**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)**

**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

1. Performance on other dataset.

**Git-hub link**: https://github.com/VandanaM1008/deeplearning