NAME: VEDANSH MAHESHWARI

|  |  |
| --- | --- |
| **Ex No: 2**  **Date:** 14/08/2024 | **Planar data classification with one hidden layer** |

**Objective:**

Implement a 2-class classification neural network with a single hidden layer, utilizing non-linear activation functions and computing cross-entropy loss through forward and backward propagation..

**Descriptions:**

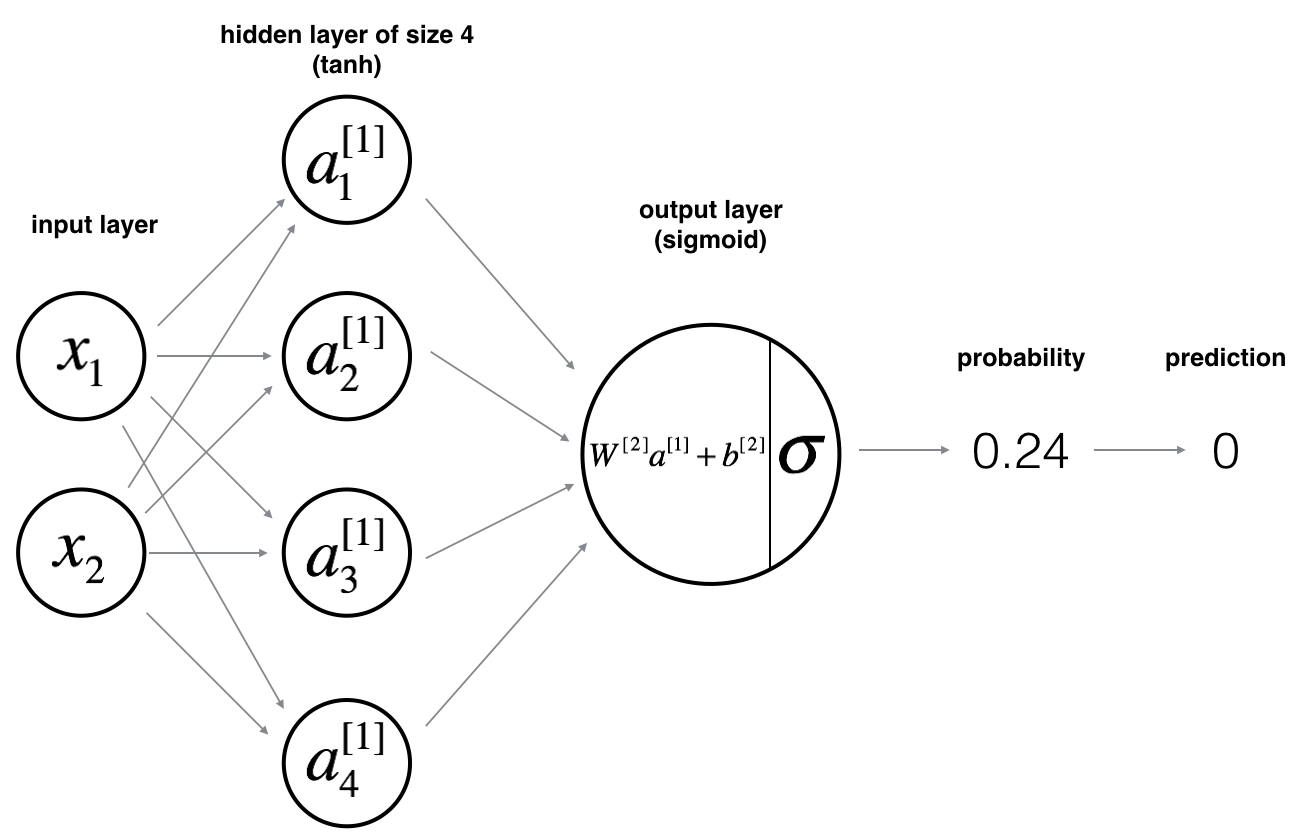
In this assignment, you will build your first neural network from scratch, focusing on a 2-class classification problem. The neural network will consist of an input layer, a hidden layer, and an output layer. The hidden layer will introduce non-linearity into the model using the tanh activation function, which allows the network to learn more complex and nuanced patterns in the data compared to linear models.

You will begin by defining the architecture of the neural network, specifying the number of units in each layer, and initializing the parameters (weights and biases) for the model. The forward propagation step will involve computing the activations of each layer, ultimately producing the output probabilities for the two classes. These probabilities will be used to calculate the cross-entropy loss, which serves as the cost function that the model aims to minimize.

The backward propagation step will then be implemented to compute the gradients of the loss with respect to the model parameters. These gradients are essential for updating the parameters in a way that reduces the loss, using an optimization algorithm like gradient descent.

Throughout this process, you will gain hands-on experience with key neural network concepts, such as activation functions, loss functions, and gradient-based optimization. By the end of the assignment, you will have a fully functional neural network capable of classifying data into two classes, providing a strong foundation for further exploration into deeper and more complex neural network architectures.

**Model:**



**Building the parts of algorithm**

The main steps for building a Neural Network with hidden layer are:

1. **Define the Model Structure**: Determine the architecture by specifying the number of input features, units in the hidden layer, and output layer size. The hidden layer uses the tanh activation function to capture complex patterns.
2. **Initialize Parameters**: Set up the weights and biases for the hidden and output layers, typically with small random values for the weights.
3. **Loop**:
   * **Forward Propagation**: Calculate the activations for each layer, producing output probabilities for the two classes.
   * **Compute Loss**: Use cross-entropy to measure the difference between the predicted and actual class labels.
   * **Backward Propagation**: Compute the gradients of the loss with respect to each parameter.
   * **Update Parameters**: Adjust the weights and biases using gradient descent to minimize the loss.

This process will result in a neural network capable of classifying planar data into two classes, allowing for more complex and accurate decision boundaries than linear models.

NAME: VEDANSH MAHESHWARI

**GitHub Link:**  <https://github.com/VedanshMaheshwari/Deep-Learning/blob/main/Labs/Lab%202/Planar_data%20DISTRI.ipynb>