USN NUMBER: 1RVU22CSE071
NAME: Joselyn Riana Manoj

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

Objective:

To implement a 2-class classification neural network with a single hidden layer, utilizing units with a non-linear activation function (such as tanh). The implementation will involve computing the cross entropy loss and developing both forward and backward propagation mechanisms to optimize the model.

Descriptions:

A 2-class classification neural network is a type of machine learning model designed to categorize elements into one of two groups. In this scenario, the model classifies data into two distinct categories, often referred to as binary classification.The neural network we are building includes a single hidden layer and uses a non-linear activation function, such as tanh, to introduce non-linearity into the decision-making process. This allows the model to better capture complex patterns within the data.

We will be working with a "flower" 2-class dataset, which will be loaded into variables X (features) and Y (labels). The dataset contains images of flowers, and the goal is to classify them into one of two categories, such as classifying between two different flower species.

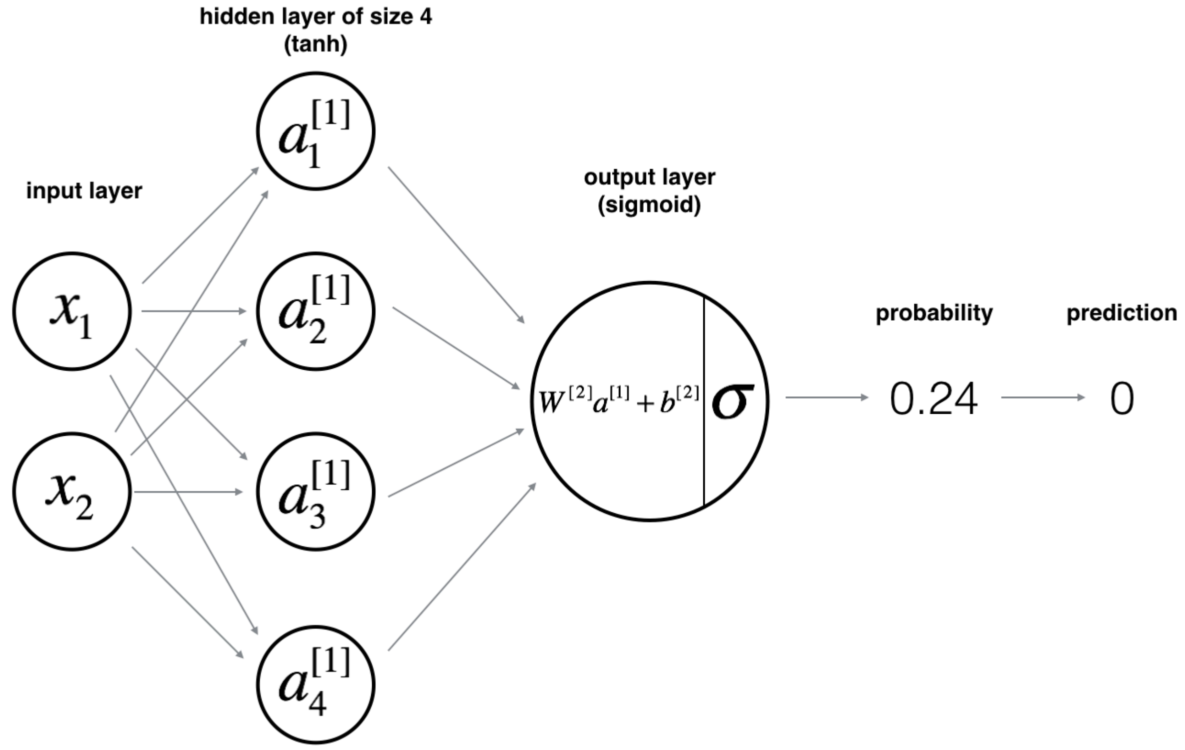
Unlike logistic regression, which is a simpler binary classification model that directly computes the output, our neural network introduces an additional hidden layer. This hidden layer improves the model's ability to generalize and detect more intricate relationships between input features. Each neuron in the hidden layer applies the tanh activation function, allowing the network to make better predictions by capturing non-linear relationships.

The model is trained using a loss function, specifically the cross-entropy loss. Cross-entropy loss measures the difference between the predicted probability distribution of the classes and the true distribution, helping the network learn the correct classification over multiple iterations.

Forward propagation is used to compute the predicted output by passing the input through the network layers, while backward propagation is employed to adjust the network weights. Backward propagation computes gradients with respect to the loss function and updates the weights to minimize the cross-entropy loss, effectively improving the model's performance in each iteration.

USN NUMBER: 1RVU22CSE071
NAME: Joselyn Riana Manoj

Model:



Building the parts of algorithm

The main steps for building a 2-class classification neural network:

1. Define the model structure (specify sizes input layer (n_x), hidden layer (n_h), and output layer (n_y)
2. Initialize the model's parameters
3. Loop:
    - Forward Propagation - using tanh function for the hidden layer and sigmoid for outer layer
    - Compute Cost
    - Backward Propagation - using chain rule and derivatives of the activation functions

- Update Parameters

- Repeat - Iterate through the forward propagation, cost computation, backward propagation, and parameter update steps for a specified number of epochs or until convergence.

This sequence covers: building a complete neural network with a hidden layer, making good use of a non-linear unit, implementing forward propagation and backpropagation, training a neural network, and observing the impact of varying the hidden layer size, including overfitting.

GitHub Link:
https://github.com/joselynrianaaa/DeepLearning_Labs/tree/main/Lab2-Planar