|  |  |
| --- | --- |
| **Ex No: 2**  **Date: 14-08-2024** | **Gradient descent implementation** |

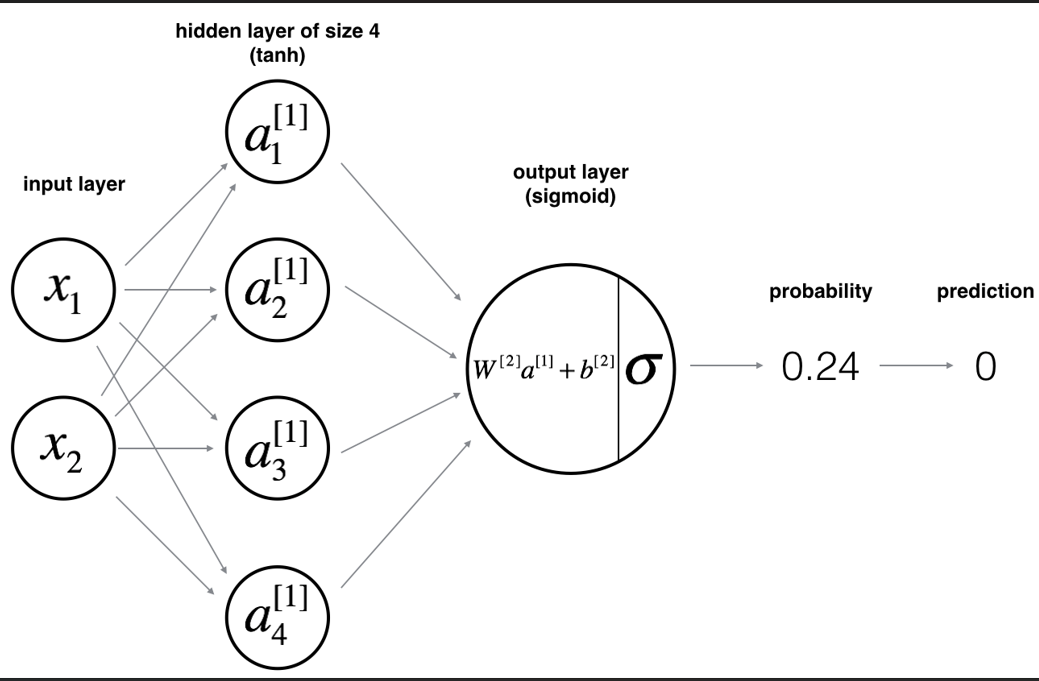
**Objective:**

Learn to build a 2-class classification neural network with a single hidden layer, using tanh activation. Implement forward and backward propagation and compute cross-entropy loss.

**Descriptions:**

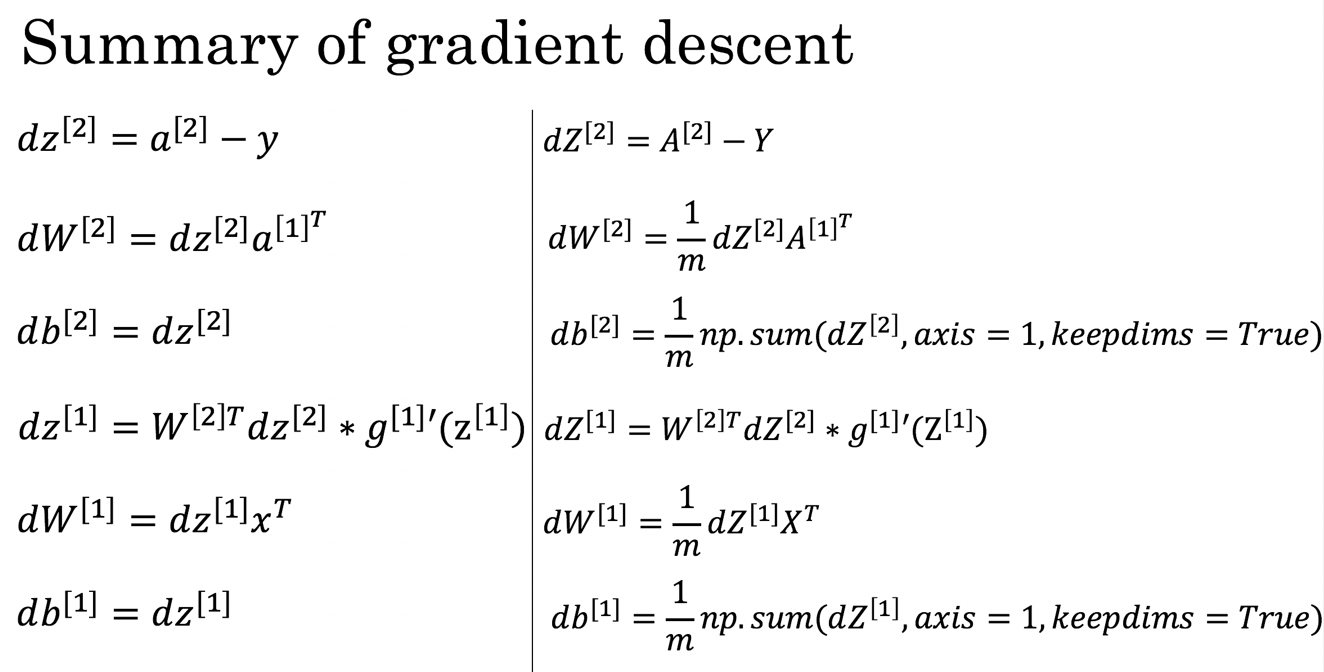
A neural network with one hidden layer is utilized for binary classification tasks using the load\_planar\_dataset() dataset, which features planar data points for class distinction. The network comprises an input layer, a hidden layer with nonlinear activation, and an output layer that produces probability scores. Forward propagation is used to generate predictions, while backward propagation refines the parameters through gradient descent to reduce classification errors. In contrast to logistic regression, which only creates linear decision boundaries, this neural network captures more intricate patterns via its hidden layer, thereby improving its accuracy in classifying data points. The model iteratively learns to differentiate between the two classes, enhancing classification performance over time.

**Model:**

****

**Building the parts of algorithm**

* Get Data: Load your dataset using load\_planar\_dataset().
* Set Up Parameters: Initialize weights and biases using initialize\_parameters().
* Train: Run training by predicting, calculating errors, and updating the model with nn\_model().
* Classify: Use predict() to make predictions with the trained network.

****

**GitHub Link:**

**https://github.com/amruthaa-m/DL-Lab1/tree/main/Unit-1/Lab2**