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

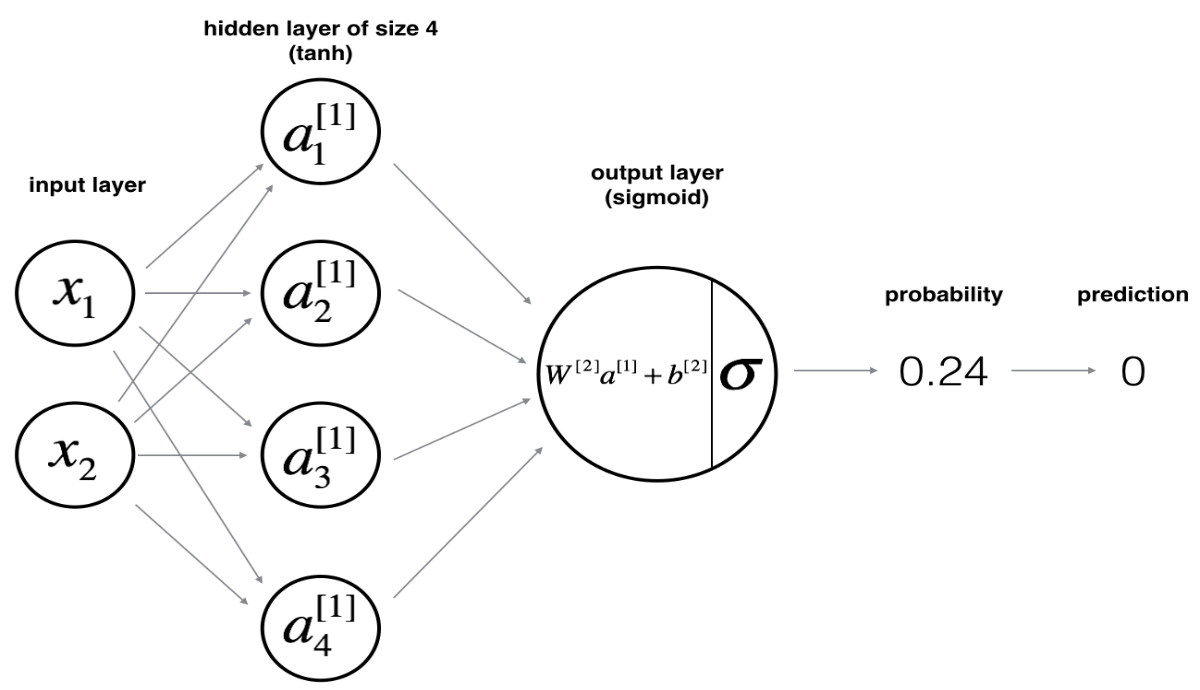
**Objective:**

To build a neural network classifier with one hidden layer to classify planar data, using forward and backward propagation along with gradient descent for parameter optimization.

**Descriptions:**

A neural network with one hidden layer is employed for binary classification tasks using the load\_planar\_dataset() dataset, which consists of planar data points. This dataset involves distinguishing between two classes based on their features. The network architecture includes an input layer, a hidden layer with nonlinear activation, and an output layer that generates probability scores. Forward propagation computes predictions, while backward propagation adjusts the parameters using gradient descent to minimize classification error. Unlike logistic regression, which only models linear boundaries, this neural network captures more complex patterns through its hidden layer, enhancing its ability to classify data points accurately. The model learns to differentiate between the classes in the dataset, improving classification performance through iterative training.

**Model:**

****

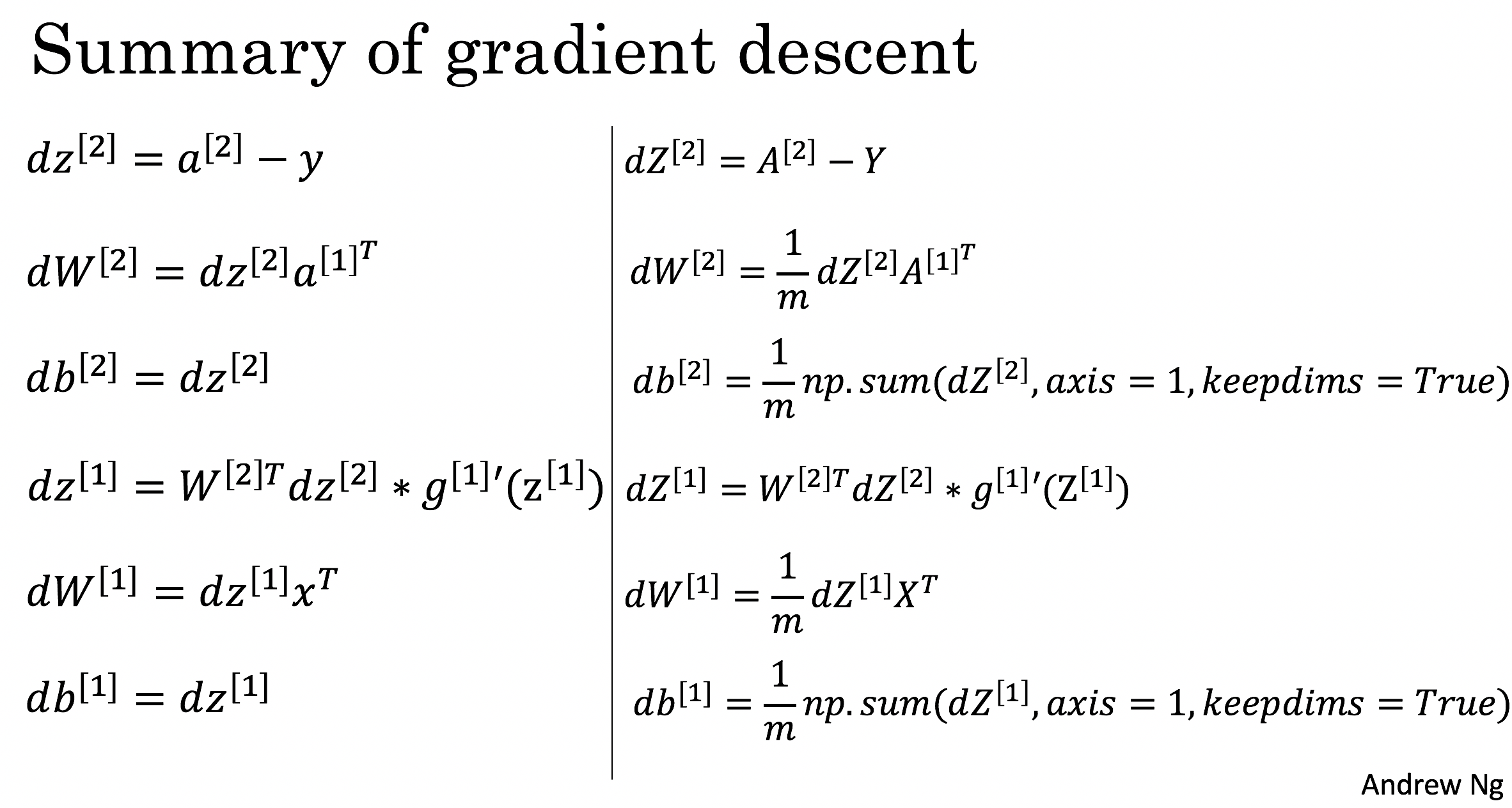
**Building the parts of algorithm**

**Load Data:** Use load\_planar\_dataset() to get your dataset.

**Initialize Parameters:** Set up initial weights and biases with initialize\_parameters().

**Train the Model:** Compute predictions, calculate errors, update parameters with nn\_model().

**Make Predictions:** Use predict() to classify new data based on the trained model.

****

**GitHub Link:**

**https://github.com/Bhargava-Srinivasan-26/Deep\_learning\_elective/tree/main/Unit%201/Lab%202**