**Task 2 Documentation**

**Team Members:**

|  |  |  |
| --- | --- | --- |
| **رقم السكشن** | **رقم الجلوس** | **الاسم** |
| 4 | 20201700340 | سامي سامر فايز |
| 6 | 20201700570 | فاطمة عوض إسماعيل |
| 2 | 20201700168 | أيمن محمد أحمد |
|  | 20201700994 | يوسف أحمد عبداللطيف |

**Overview**

This Python script implements a neural network from scratch for classification tasks using NumPy and Pandas libraries. The neural network performs classification on a dataset and evaluates its performance using accuracy metrics and a confusion matrix.

**Dependencies**

- NumPy: For numerical operations and handling matrices.

- Pandas: For data manipulation and preprocessing.

- Matplotlib & Seaborn: For data visualization.

- sys: For system-specific parameters and functions.

**Modules and Functions**

1. Data Preprocessing (preprocess)

- Reads the dataset from a CSV file, performs data cleaning by filling missing values and scaling features using Min-Max scaling.

- Splits the data into training and testing sets.

2. Activation Functions

- sigmoidFunc: Implements the sigmoid activation function.

- derivative\_sigmoid: Computes the derivative of the sigmoid function.

- tanh: Implements the hyperbolic tangent (tanh) activation function.

- tanh\_derivative: Computes the derivative of the tanh function.

3.Weight Initialization

- initialize\_weights: Initializes weights for the neural network.

4. Forward Propagation

- activate\_input: Computes neuron activation for input data.

- activate\_hidden: Computes neuron activation for hidden layers.

5. Backpropagation

- backprop\_out: Performs backpropagation for the output layer.

- backprop\_hidden: Performs backpropagation for hidden layers.

6. Weight Update

- update\_weights: Updates weights based on backpropagation and learning rate.

7. Training the Neural Network

- train: Trains the neural network using forward and backward propagation.

8. Model Evaluation

Training Accuracy

- train\_acc: Calculates the training accuracy of the model and the confusion matrix.

**Results:**

The training accuracy = 0.9 (with bias, learning rate = 0.01, and epochs =1000)

The test accuracy =0.7

And it should be noted that the training accuracy =0.38 without adding bias

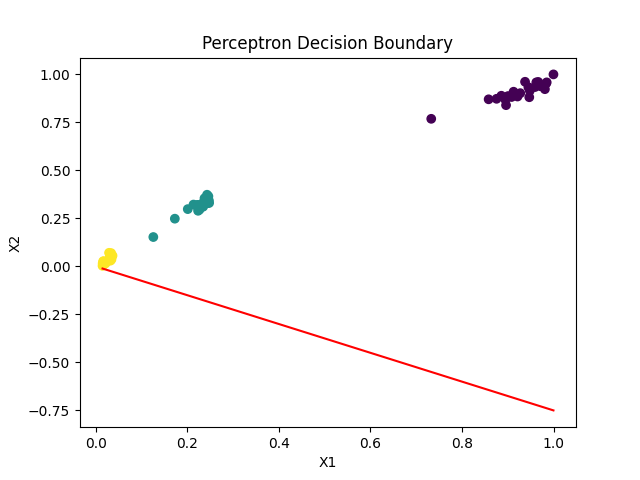
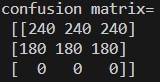
**Usage**

- The code contains functions for data preprocessing, neural network training, and evaluation.

- Users can adjust parameters such as learning rate, epochs, and network architecture for experimentation.

- The ‘testing’ function computes the testing accuracy and generates the confusion matrix.

**Visualization**

 ****