Task 2: Back-Propagation learning algorithm

Neural networks and deep learning

Task parts:

1- Read data (Number of Hidden Layers, Number of Neurons, Learning Rate, Epochs, Activation Function, Add bias) from Gui.
2- Read data from excel sheet depending on data entered through Gui.
3- Perform preprocessing.
4- Implement Back-propagation Algorithm.
5- Output accuracy and Confusion matrix.

1- GUI:
- Enter Number of Hidden Layers.
- Enter Number of Neurons for each hidden layer.
- Enter parameters needed to be initialized in the algorithm:
  1- Learning Rate.
  2- Epochs.
  3- Bias to be included or not choice.
- Select Activation Function.

2- Read excel sheet:
- Read all columns, rows.

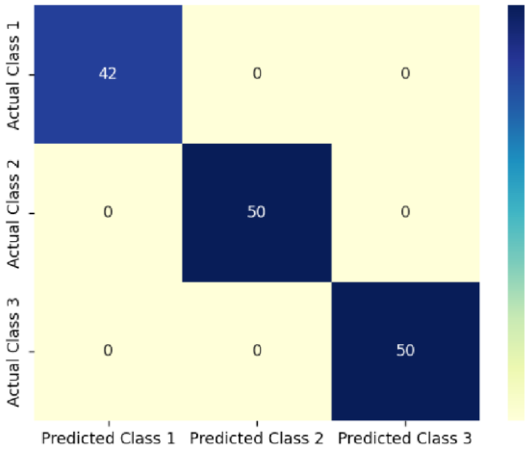
3- Preprocessing:
   - Fill null values in all columns.
   - Convert column Class to hot encoding and split it to 3 columns and drop it.
   - Normalize data with MinMaxScaler.
   - Split data to x_train, x_test, y_train, y_test.

4- Implement Back-propagation Algorithm:
   - Generate Weights and biases: Initialize weights and biases (in case it's decided to be used) with random numbers for input layer to first hidden layer, hidden layers, last hidden layer to output layer.
   - Forward step: loop in each (weights, bias) to calculate net and use activation function for each hidden layer.
   - Backward step:
     - Loop in each weight and calculate the net and use activation function.
     - Calculate error.
     - Calculate delta for output layer = error * derivative of activation function.
     - Loop in each weight and calculate delta for each hidden layer.
     - Loop in each weight and update weights and biases.
   - Classification (Forward step): calculate forward step for x_test and return this prediction.

5- Output accuracy and Confusion matrix:
   - Initialize confusion matrix and loop in each sample to calculate confusion matrix.
   - Print confusion matrix.
   - Calculate accuracy and print it.
   - Convert confusion matrix to data frame and send it to heatmap function and show it.

Results: - Use Activation Functions

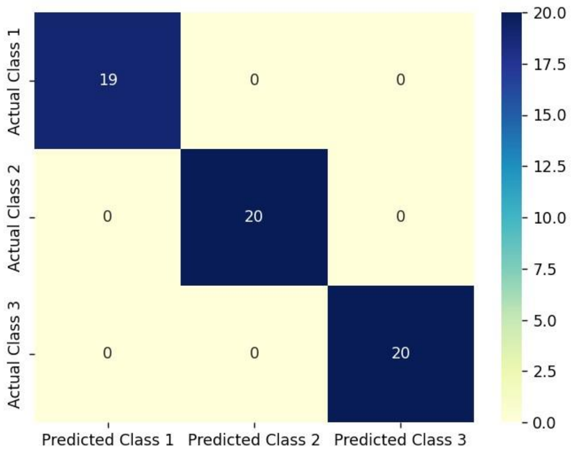
1- Sigmoid Function:
   - Overall accuracy = 94.67 %



	Predicted Class 1	Predicted Class 2	Predicted Class 3
Actual Class 1	42	0	0
Actual Class 2	0	50	0
Actual Class 3	0	0	50

Confusion Matrix

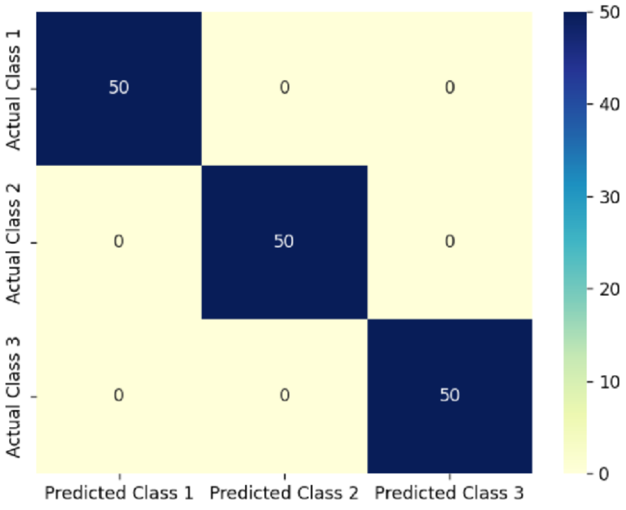
- Highest test accuracy = 98.3 % by - Parameters:
  - Number of Hidden layers = 2,
  - Number of Neurons = [3, 4],
  - Learning Rate = 0.01,
  - Epochs = 1000,
  - Bias = True



Confusion Matrix

2- Hyperbolic Tangent Function: -

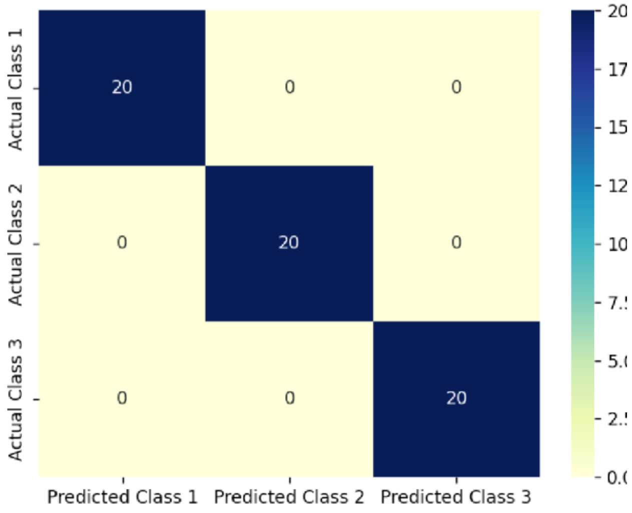
Overall accuracy = 100 %.



Confusion Matrix

- Highest accuracy = 100 % by -
Parameters:

- Number of Hidden layers = 1,
- Number of Neurons = 5,
- Learning Rate = 0.001,
- Epochs = 5000,
- Bias = True,



Confusion Matrix