Machine Learning

Digits Data Set – MLP Classifier (Neural Networks with multiple layers)

Here we need to use MLP for classification.

(a) Use the digits data used by you in assignment 5. Train an MLP each with 1,2, 3 and 4 hidden layers using Backpropagation to classify patterns of all the 10 classes.
(b) Take 80%. for training and the rest for testing. Compute the classification accuracy on the test patterns.
(c) Repeat step (b) with different train test size. Pick the best MLP.

SOLUTION:

CODE:

Please find the code Submitted for MLP Classifier as *NN_MLP_Classifier_DigitsDataSet_Imp.py*

- The digit's dataset is obtained and normalised so that the data frame has values 0 and 1.
- Various MLP classifiers from sklearn are used by varying number of hidden layers and train_test_size.
- Accuracy is computed by applying permutations of various hidden layers along with train_test_size

RESULT:

Accuracy for MLP Classifier with 1 Layers and test_size= 0.1 ======> 0.992209
Accuracy for MLP Classifier with 1 Layers and test_size= 0.2 ======> 0.981080
Accuracy for MLP Classifier with 1 Layers and test_size= 0.3 ======> 0.980523
Accuracy for MLP Classifier with 1 Layers and test_size= 0.4 ======> 0.967168

Accuracy for MLP Classifier with 2 Layers and test_size= 0.1 ======> 0.992209
Accuracy for MLP Classifier with 2 Layers and test_size= 0.2 ======> 0.984975
Accuracy for MLP Classifier with 2 Layers and test_size= 0.3 ======> 0.977741
Accuracy for MLP Classifier with 2 Layers and test_size= 0.4 ======> 0.963829

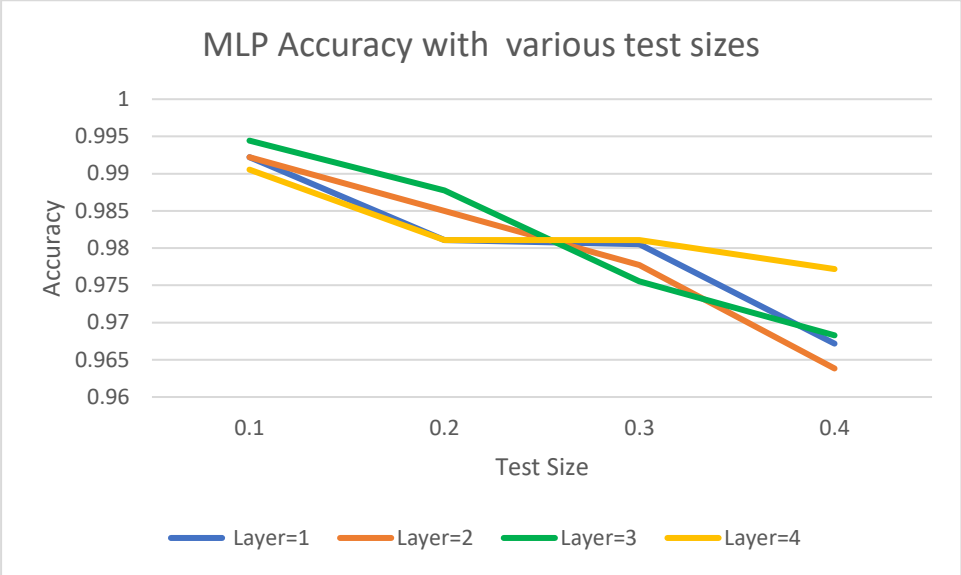
Accuracy for MLP Classifier with 3 Layers and test_size= 0.1 ======> 0.994435
Accuracy for MLP Classifier with 3 Layers and test_size= 0.2 ======> 0.987757
Accuracy for MLP Classifier with 3 Layers and test_size= 0.3 ======> 0.975515
Accuracy for MLP Classifier with 3 Layers and test_size= 0.4 ======> 0.968280

Accuracy for MLP Classifier with 4 Layers and test_size= 0.1 ======> 0.990540
Accuracy for MLP Classifier with 4 Layers and test_size= 0.2 ======> 0.981080
Accuracy for MLP Classifier with 4 Layers and test_size= 0.3 ======> 0.981080
Accuracy for MLP Classifier with 4 Layers and test_size= 0.4 ======> 0.977184

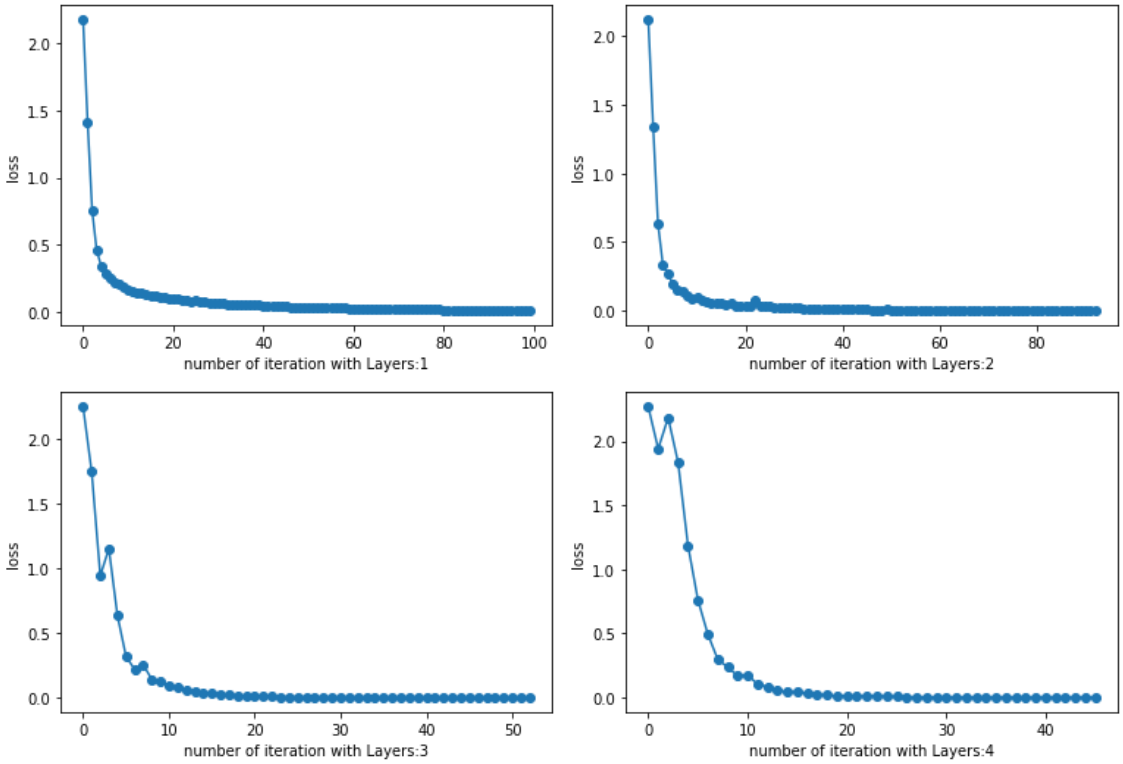
From above results we are seeing that the best MLP accuracy is obtained with 3 hidden Layers and test_size= 0.1

PLOTS:

Below is the plot of accuracy obtained with our MLP Classifiers



Below are various plots of loss curve with our MLP classifier



INFERENCE/ANALYSIS:

- **MLP Classifier** works well and provides a high accuracy of ~98% for the digit's dataset.
- **In digit's dataset** there are 1997 samples with 64 features which forms X and the class variable y represents one of the 10 digit classes.
- From accuracy plot, we can see that the increment in test size results in lesser accuracy.
- From Accuracy plot we can also see that higher number of hidden layers give better accuracy but that also saturated out at Layer=3.
- From Accuracy plot we can also see that, the rate of drop in performance/accuracy reduces with more layers. i.e., with layer=4 when test size is increasing the drop in accuracy is not as steep as others
- From Loss curve we can observe that with higher number of iterations , the MLP Classifier converges to lesser loss values and loss decrement appears to be Hyperbolic in nature

When MLP Classifier is applied over digit's dataset completely , the average accuracy of classification is very high. Additionally, when MLP regressor is applied on the Boston Housing Data, Low RMSE results were seen when we used data pre-processing and main contributing features for training our ML Models. Through the exercise of generating many permutations by varying key parameters, we can clearly see that tanh/relu are better performing activation functions for these data sets and with increasing number of hidden layers the performance is getting better.

RESOURCES USED FOR THE ASSIGNMENT:

• **Environment:** Anaconda, Jupyter notebook
• **Software :** Python **Python libraries/modules:** Pandas, Numpy, SkLearn, Scipy, matplotlib, etc