School of Computer Science Engineering and Technology

Course- BTech	Type- Core
Course Code- 301	Course Name-AIML
Year- 2022	Semester- Even
Date- 06-03-2022	Batch- 4th Sem (SPL)

Lab Assignment No. 8.1

Objective: To implement Multilayer Perceptron (using Scikit-learn) to classify images in MNIST dataset of handwritten digits.

1. **Download** the dataset from https://www.openml.org/d/554 . The MNIST database contains a total of 70000 examples of handwritten digits of size 28x28 pixels, labelled from 0 to 9. You can use function **fetch_openml("mnist_784")** to directly download. (10)

2. **Fetch_openml** function returns a data bunch. Using its attributes print the **shape of the input data and target data.** It should be (70,000, 784) and (70,000,) respectively. (5)

3. **Display** the top ten images using matplotlib. You will be required to reshape the dataset temporarily into (70,000, 28, 28) dimensions.

4. Define **X** matrix (70,000, 784) and **y** vector (target feature). (5)

5. **Transform:** Multi-layer Perceptron is sensitive to feature scaling, so it is highly recommended to scale your data. For example, scale each attribute on the input vector X to [0, 1] or [-1, +1], or standardize it to have mean 0 and variance 1. (15)

6. **Split** the dataset into **80% for training** and rest **20% for testing** (sklearn.model_selection.train_test_split function) (5)

7. **Train** MultilayerPerceptron Model using built-in function on the training set MLPclassifier() constructor with following settings:
   a. only **one hidden layer** consisting of just **32 neurons**
   b. Set the **max_iter** to a very low value such as **5.**

(Use **sklearn.neural_network import MLPClassifier**) (10)

8. Use the trained model to **predict** on the **test set** and then (15)
   a. Print 'Accuracy' obtained on the testing dataset i.e. (sklearn.metrics.accuracy_score function)
   b. Precision, Recall and F1 scores (sklearn.metrics.precision_recall_fscore_support)

9. Compare and analyse the **test accuracy** for different train-test splits of data such as 60-40, 70-30, 80-20 and 90-10 with the help of **suitable graphs**. (15)

10. **Playing with the model**: Increase the number of iterations to larger values such as 50, 100, 150 and 200 to see the variations in accuracy.