School of Computer Science Engineering and Technology

Course-B. Tech	**Type-** General Elective
Course Code- CSET-335	**Course Name-** Deep Leaning
Year- 2024	**Semester-** Even
Date- 05/02/2024	**Batch-** 2023-2024

CO-Mapping

	CO1	CO2	CO3
Q1	√		
Q2	√		
Q3	√		
Q4	√		
Q5	√		
Q6	√		
Q7	√		
Q8	√		

Objectives:

CO1: To explain the fundamentals of deep learning, Convolution neural network.
CO2: To articulate different problem of classification, detection, segmentation, generation and understand existing solutions/ deep learning architectures.
CO3: To implement a solution for the given problem and improve it using various methods transfer learning, hyperparameter optimization.

Assignment-2

Goal: To implement Feed Forward Network (FFN) using Scikit-learn to classify images in MNIST dataset of handwritten digits.

Q1. 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 the function **fetch_openml("mnist_784")** to directly download.

Q2. 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.

Q3. Display the top ten images using matplotlib. You will be required to reshape the dataset temporarily into (70,000, 28, 28) dimensions. Define **X** matrix (70,000, 784) and **y** vector (target feature).

Q4. Transform: FFN 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.

Q6. Split the dataset into **70% for training** and the rest **30% for testing.**

Hint: sklearn.model_selection.train_test_split function

Q7. Train FFN 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**)

Q8. Use the trained model to **predict** on the **test set** and then:

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)