# School of Computer Science Engineering and Technology

| | |
|---|---|
| **Course-**B. Tech | **Type-** General Elective |
| **Course Code-** CSET-335 | **Course Name-** Deep Leaning |
| **Year-** 2025 | **Semester-** Even |
| **Date-** 20/01/2025 | **Year-** 2024-2025 |

**CO-Mapping**

| Part-1 | CO1 | CO2 | CO3 |
|---|---|---|---|
| **Q1** | √ | | |
| **Q2** | √ | | |
| **Q3** | √ | | |
| **Q4** | √ | | |
| **Q5** | √ | | |
| **Q6** | √ | | |
| **Q7** | √ | | |
| **Q8** | √ | | |
| **Q9** | √ | | |
| **Q10** | √ | | |
| **Part-2** | √ | | |

## 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-1 (Wk1 and WK2)

**Goal(s): Part1: To implement Feed Forward Network (FFN) using Scikit-learn to classify images in MNIST dataset of handwritten digits.**

**https://www.geeksforgeeks.org/mnist-dataset/**
**https://git-disl.github.io/GTDLBench/datasets/mnist_datasets/**

**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 **80% for training** and rest **20% for testing**

- (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 **64 neurons.**
**B.** Set the **max_iter** to a very low value such as **10.**

(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)

Q9. Compare and analyse the **test accuracy** for different train-test splits of data such as 60-40, 75-25, 80-20 and 90-10 with the help of **suitable graphs**.

Q10**. Playing with the model**: Increase the number of iterations to larger values such as 20, 50, 100, 150 and 200 to see the variations in accuracy.

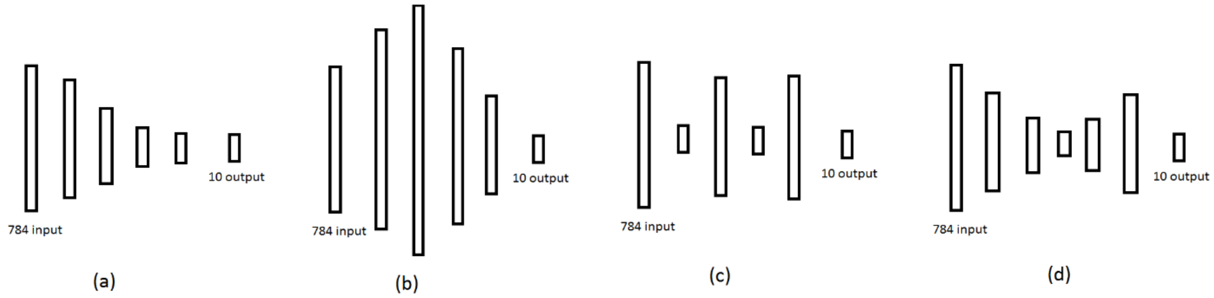
## Part-2: Exploring different Neural Network design choices for Digit classification using MNIST dataset with the help of Keras library.

1. **Number of Nodes:** Run neural network with single hidden layer, 128 nodes (with any activation function and any optimizer) for 10 epochs. Change number of nodes as 4, 32, 64, 512, 2056. What is the training and testing accuracies? Print the number of parameters of the model and training time for each of these configurations.

2. **Number of Layers:** Run neural network with 5 hidden layers, 64 nodes each (with any activation function and any optimizer) for 10 epochs. Change the number of layers to 4, 6, 8, 16. What are the training and testing accuracies? Print the number of parameters of the model and training time for each of these configurations. Run the same models for 30 epochs. Any changes?

3. **Layer-node combinations:** Run different models mimicking the following structures. Which one gives best accuracy? Print the number of parameters of the model and training time for each of these configurations. Here, large sized bar means a layer with larger number of neurons in it. You can choose the number of neurons such as 256, 128, 64, 32 etc.

4. **Input Size:** Run neural network with 4 hidden layers, 256 nodes each (with any optimizer) with ReLU activation function for all layers, for 10 epochs. Print accuracies change?



5. **Dataset Split:** Instead of 60k images for training and 10k images for testing, run different models with different training and testing sizes, and see the changes in the accuracies.

6. **Activation function**: Run neural network with 4 hidden layers, 64 nodes each (with any optimizer) with sigmoid activation function for all layers, for 10 epochs. Change only the activation function as tanh, relu (for all 4 layers) etc. What is the training and testing accuracies? Run the same models for 30 epochs. Any changes?

7. **Activation function combinations:** For the three layers use different combinations of activation functions such as layer 1: sigmoid, layer 2: relu, layer 3: tanh. There can be lot of combinations like this. Which one is the best for 3 layers each with 32 node architectures?