Programming Assignment 2

Name: Mayank Kumar
UCF-ID: 5529148
Email: mayank_kumar@ucf.edu

Convolutional Neural Network (CNN) for Classification

1. **Introduction**
   In this assignment the task is to implement a Convolutional Neural Network based digit classifier and compare its performance with a simple Multi-layer perceptron based classifier. For this, we will implement 5 variants with different architectures.

2. **Dataset**
   For this assignment, we will use the MNIST dataset which is a widely used dataset in the field of machine learning and computer vision. It stands for the "Modified National Institute of Standards and Technology" dataset and is a collection of handwritten digits.

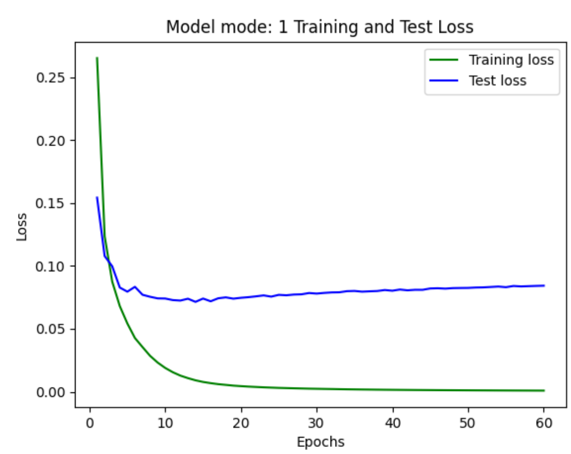
   The MNIST dataset consists of 70,000 grayscale images of handwritten digits, each measuring 28x28 pixels. These images are divided into a training set of 60,000 samples and a test set of 10,000 samples. Each image corresponds to a single digit (0-9) and is labeled with its respective digit. MNIST is a benchmark dataset for digit classification tasks and serves as a fundamental dataset for testing and benchmarking machine learning algorithms.

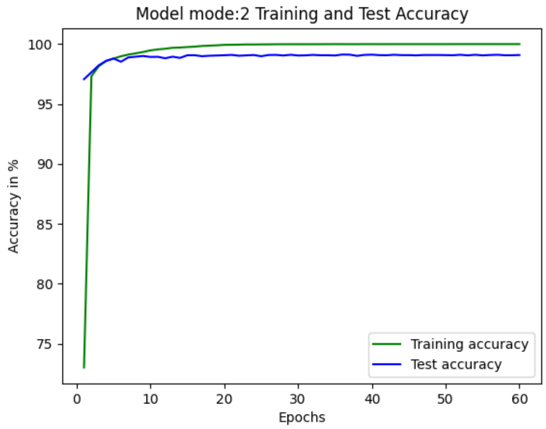
3. **Model Architectures Used**
   The following architectures are used:
   - **Model 1:** One Fully connected (FC) hidden layer (with 100 neurons) with Sigmoid activation function. Trained with SGD optimizer with a learning rate of 0.1 for a total of 60 epochs, a mini-batch size of 10, and no regularization.
   - **Model 2:** Two Convolutional Layers followed by a fully connected layer like in Model 1. Each Conv. layer is followed by a max pooling layer and a sigmoid activation layer. Pooling is done over 2x2 regions, 40 kernels, stride = 1, with kernel size of 5x5. Train on the same setting as Model 1.
   - **Model 3:** Each sigmoid layer in Model 2 is replaced by a ReLU activation function. The new learning rate is 0.03.

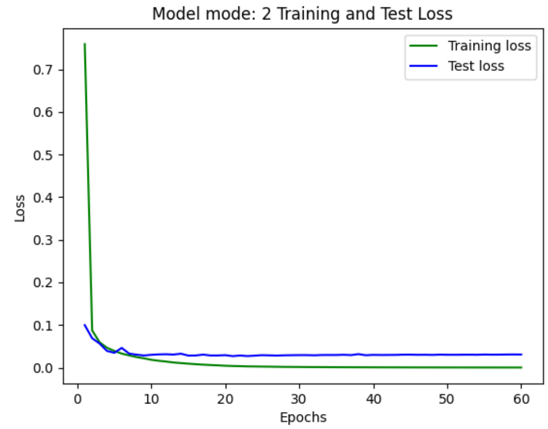
- **Model 4:** Another fully connected layer with 100 neurons is added to Model 3. Training is done with the same parameters as Model 3.
- **Model 5:** The number of neurons in the hidden layers is changed to 1000, and for regularization, Dropout with probability 0.5 is used. The model is trained for 40 epochs. The other parameters are unchanged from Model 4.

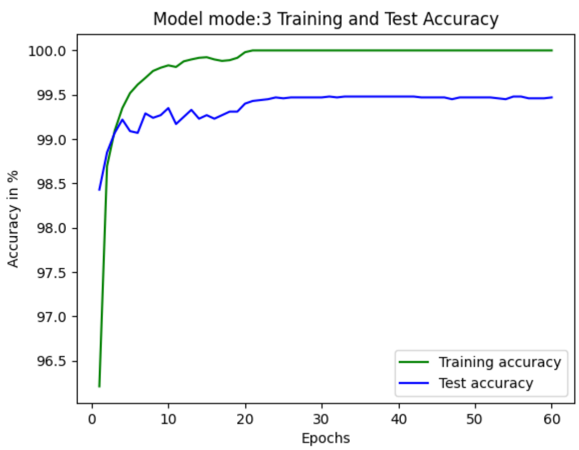
4. Results and Discussion

Model	Accuracy	Convergence Epoch
1	98.01 %	28
2	99.13 %	36
3	99.48 %	31
4	99.35 %	53
5	99.45 %	36



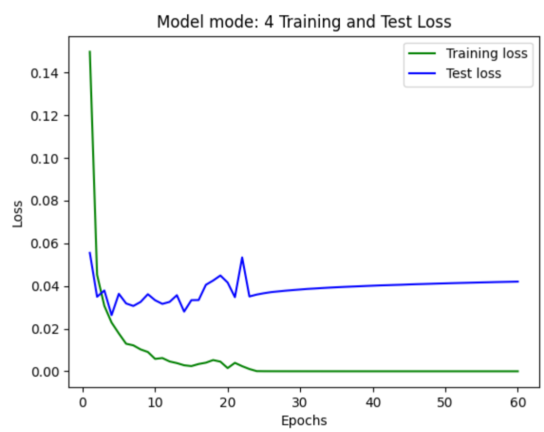

Model mode:2 Training and Test Accuracy
Accuracy in %
Epochs
Training accuracy
Test accuracy

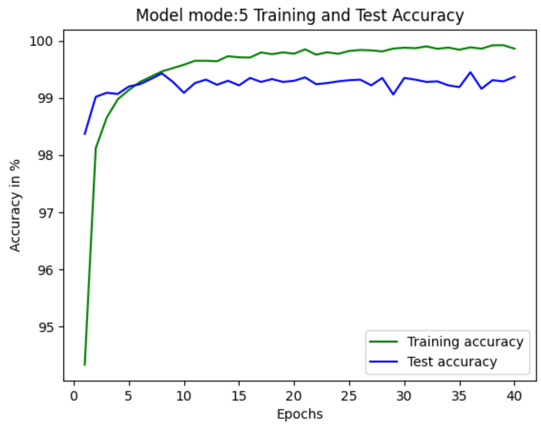

Model mode: 2 Training and Test Loss
Loss
Epochs
Training loss
Test loss

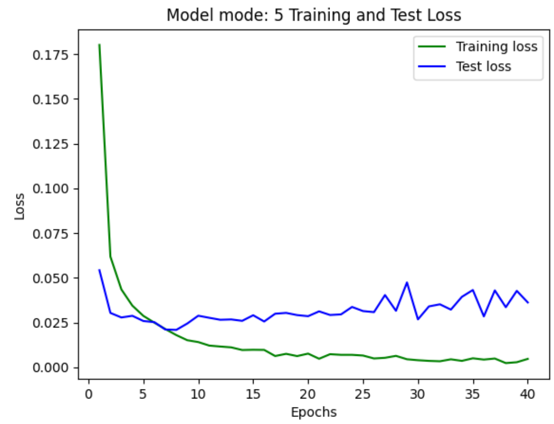

Model mode:3 Training and Test Accuracy
Accuracy in %
Epochs
Training accuracy
Test accuracy


Model mode: 3 Training and Test Loss
Loss
Epochs
Training loss
Test loss


Model mode:4 Training and Test Accuracy
Accuracy in %
Epochs
Training accuracy
Test accuracy


Model mode: 4 Training and Test Loss
Loss
Epochs
Training loss
Test loss





As the complexity of the model architectures increases from model 1 to model 5, we see a general improvement in each subsequent performance. Using CNN in model 2 drastically improved the loss function. As the sigmoid activations in model 2 were replaced by ReLU in model 3, we see a better loss of somewhere less than 0.04. Another fully connected layer actually slightly decreased the performance in model 4. Finally in model 5, due to dropout regularization, the model performs at par with model 3 albeit slightly worse. Depending on the application, these performance differences may be acceptable. Since MNIST is a very simple dataset and model 2 has already achieved near 99% accuracy, the improvement in further models is only fractional. However, in more complex problems, model 5 may be able to generalize well.