Module 3: Introduction
to NN Frameworks - Assignment

Problem Statement:

This assignment challenges students to construct and train a neural network to accurately classify handwritten digits from the MNIST dataset. The goal is to navigate through the preprocessing of image data, select an appropriate neural network architecture, and apply optimization techniques to achieve high classification accuracy.

Introduction:

The MNIST dataset is a large database of handwritten digits commonly used for training various image processing systems. The dataset contains 60,000 training images and 10,000 testing images, each of which is a 28x28 pixel grayscale image. The goal is to build a neural network model that can accurately classify these images into one of the ten digit classes (0 through 9).

Objectives:

Data Preprocessing: Prepare the MNIST dataset for training and testing the neural network model. This includes normalization of image pixel values and converting the output labels to a one-hot encoded format.

Model Development: Create a neural network model capable of classifying 28x28 pixel grayscale images of handwritten digits. The model should have an input layer, one or more hidden layers, and an output layer.

Training: Train the neural network model using the preprocessed training data. Utilize categorical cross-entropy as the loss function, stochastic gradient descent as the optimizer, and track accuracy as a performance metric.

Evaluation: Evaluate the performance of the trained model using the testing dataset, ensuring that the model generalizes well to unseen data.

Prediction: Implement functionality to make predictions on single images or batches of images, returning the digit class with the highest probability.

Analysis: Analyze the results, understand the model's performance, and explore ways to improve it if necessary.

Challenges and Considerations:

Class Imbalance: Ensure that the dataset does not have a significant class imbalance that could negatively affect the model's performance. If imbalance is present, strategies to mitigate its effects should be considered.

Model Complexity: Choose an appropriate model complexity that is capable of learning from the data without overfitting. This may involve experimenting with different numbers of layers and units, as well as regularization techniques.

Activation Functions: Use appropriate activation functions in the hidden layers (ReLU is commonly used for image data) and softmax activation in the output layer to convert logits to probabilities.

Evaluation Metrics: Focus on accuracy as the primary evaluation metric, but also be open to exploring other metrics that could provide additional insights into the model's performance.

Hyperparameter Tuning: Experiment with different hyperparameters such as learning rate, batch size, and number of epochs to optimize the model's performance.

Predictions and Interpretability: Provide clear and interpretable predictions, ensuring that the output probabilities are well-calibrated and meaningful.