Analysis of the Effect of MLP Depth on MNIST Classification

Student Name: Vivek Morcha

Git Link: https://github.com/Vivek9049P/MLP.git

Abstract:

This report explores how the depth of a neural network affects its performance on the MNIST handwritten digit classification task. Multi-Layer Perceptron (MLP) models with different numbers of hidden layers specifically 1, 2, and 4 were implemented and trained using the PyTorch framework. All models were trained under the same conditions to ensure a fair comparison. Their performance was evaluated using training loss and test accuracy across several epochs. The results show that increasing network depth improves learning and accuracy up to a certain point, after which the benefits become less noticeable within the limited training time.

Introduction:

Neural networks are one of the most widely used techniques in machine learning, and their performance depends heavily on how they are designed. One important design choice is the depth of the network. Deeper networks are capable of learning more detailed and complex patterns from data, but they can also be more difficult to train and may require careful tuning of parameters. Finding the right balance between simplicity and complexity is especially important when working with basic datasets and baseline models. The MNIST dataset is a popular benchmark for handwritten digit recognition and is commonly used to analyse neural network performance. In this experiment, Multi-Layer Perceptron models with different numbers of hidden layers are trained on MNIST to study how network depth affects learning behaviour and classification accuracy. The goal is to compare shallow and deeper models and understand how depth influences training efficiency and generalization ability.

Dataset Description:

The MNIST dataset is composed of grayscale images representing handwritten digits from 0 to 9, with each image having a resolution of 28×28 pixels. The dataset includes a total of 70,000 images, of which 60,000 are used for training the models and 10,000 are reserved for evaluation. Prior to training, the images are converted into tensor format and normalized so that their pixel values remain within a uniform range. This preprocessing step allows the model to learn more efficiently and contributes to improved training stability and faster convergence.

Model Architecture:

The models used in this experiment are fully connected neural networks, commonly referred to as Multi-Layer Perceptron's (MLPs). Each model takes a 28×28 MNIST image, which is flattened into a vector of 784 input features, and outputs predictions for the 10 possible digit classes.

Three MLP architectures with different depths were evaluated: a model with one hidden layer, another with two hidden layers, and a deeper model with four hidden layers. In all cases, each hidden layer consists of 128 neurons and uses the ReLU activation function to introduce non-linearity. The final output layer applies a linear transformation to generate class scores, which are then used for digit classification.

Training Methodology:

1. **Training Setup**
   To ensure a fair comparison, all models were trained using identical hyperparameters. The Adam optimizer was applied with a learning rate of 0.001, and Cross-Entropy loss was used for optimization. Each network was trained for 5 epochs with a batch size of 64. Training was performed on a GPU when available to improve speed, and on the CPU otherwise.

2. **Training Process**
   During each training epoch, the model processes batches of images from the training dataset and updates its weights based on the calculated loss. Once an epoch is completed, the model is evaluated on the test dataset to measure its performance. Both the training loss and test accuracy are recorded after each epoch, making it easier to compare how models with different depths learn and generalize over time.

Experimental Results:

1. Training Loss Analysis

The training loss curves show a steady decrease for all three models as training progresses, which indicates that each network is learning meaningful features from the data. However, the model with only one hidden layer reduces its loss more slowly compared to the deeper models, suggesting limited learning capacity.

Both the two-layer and four-layer models converge more quickly and achieve lower loss values, showing that deeper networks are better at capturing patterns in the dataset. Among these, the two-layer model displays the most stable and efficient convergence within the limited number of training epochs.

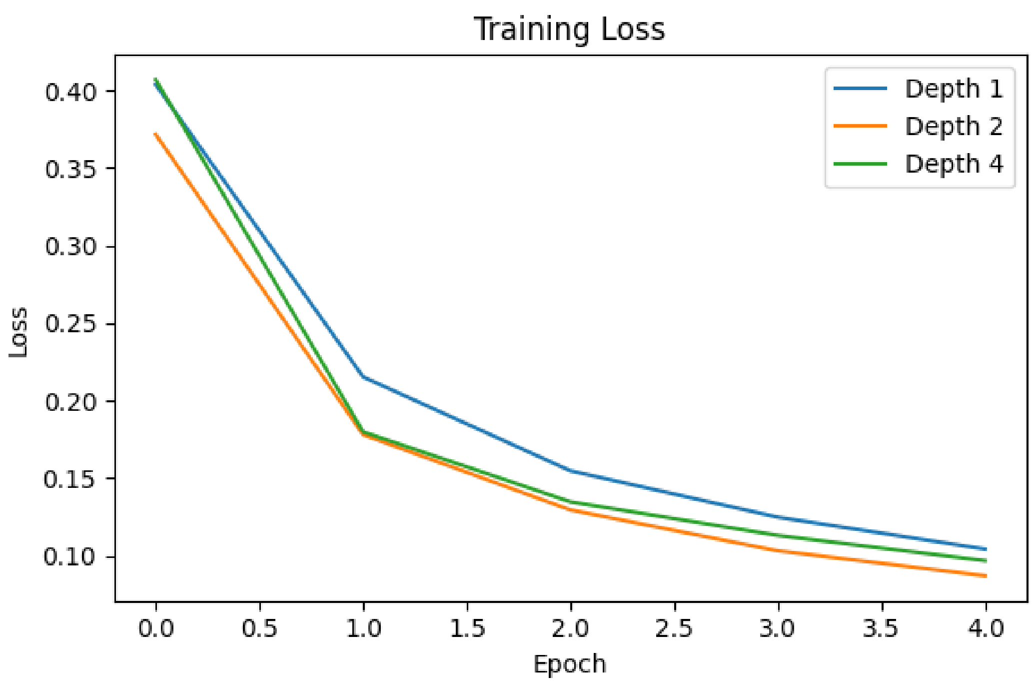

Figure 1: Training loss vs epoch for MLP models with different depths.

2. Test Accuracy Analysis

The test accuracy results highlight clear performance differences between the models. Although all three architectures achieve high accuracy on the MNIST dataset, the deeper models consistently perform better than the shallow model.

The model with two hidden layers achieves the highest test accuracy, exceeding 97%. The four-layer model also shows strong performance but exhibits slight fluctuations across epochs, likely due to increased model complexity and the relatively short training duration. Overall, the results suggest that a moderately deep network provides the best balance between accuracy and training stability.

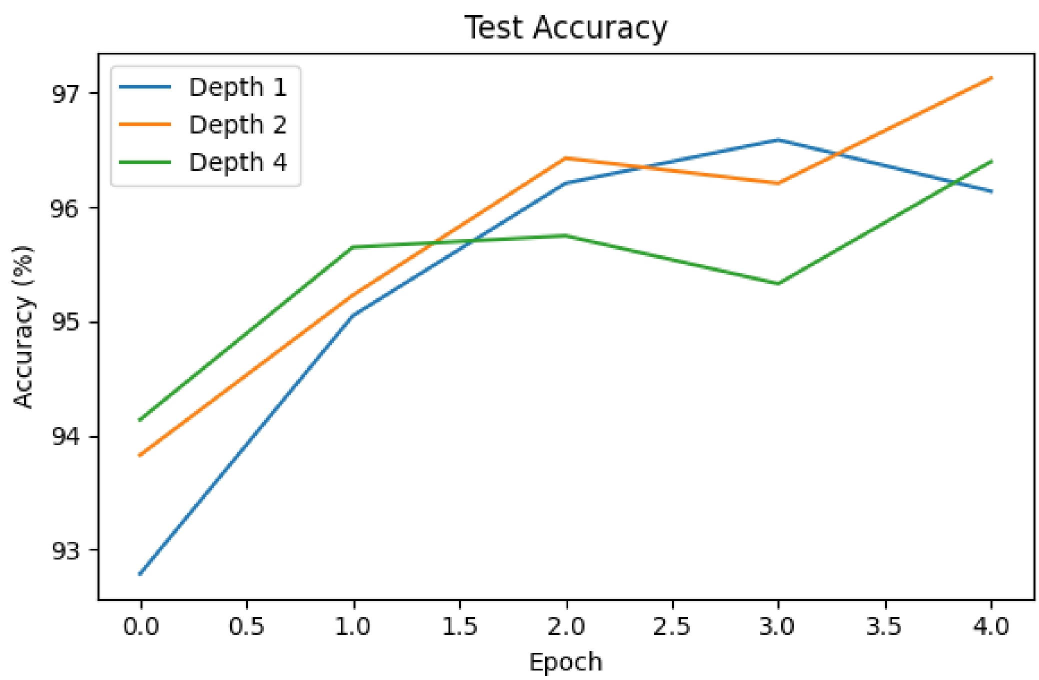

Figure 2: Test accuracy vs epoch for MLP models with different depths.

Limitations:

This study has a few limitations that should be considered. The models were trained for only a small number of epochs, which may not be sufficient for deeper networks to reach their full potential. Additionally, only one hidden layer size was explored, and no regularization methods such as dropout or weight decay were used. Future work could involve experimenting with deeper architectures, longer training durations, and more advanced optimization or regularization techniques to further improve performance and gain deeper insights into the effects of network design.

Discussion:

The results of this experiment clearly show how network depth affects model performance. As the depth of the network increases, the model becomes better at learning more detailed and complex features from the data, which leads to faster learning and improved accuracy. However, simply adding more layers does not always result in better performance, particularly when the training time is limited. The four-layer model appears to need more training epochs or additional techniques such as regularization to make full use of its increased capacity. On the other hand, the two-layer model delivers strong and consistent performance with very little tuning, making it a practical and efficient choice for this classification task.

Conclusion:

This experiment explored how the depth of a Multi-Layer Perceptron influences its performance on the MNIST handwritten digit dataset. Three models with different numbers of hidden layers were trained and evaluated under the same conditions to ensure a fair comparison. The results clearly show that increasing network depth improves the model’s ability to learn meaningful patterns and generalize to unseen data, but only up to a certain point. Among the architectures tested, the model with two hidden layers consistently delivered the best overall performance in terms of accuracy and training stability.

These findings highlight the importance of thoughtful network design when building neural networks. While deeper models can offer greater representational power, they also require careful tuning and sufficient training time to fully utilize their capacity. This experiment demonstrates that a balanced architectural choice often leads to more reliable and efficient results than simply increasing model complexity.