Analysis of Neural Network Architectures for Handwritten Digit Classification

Student: Yunzhen Wu

Course: MSDS 422 Practical Machine Learning

Instructor: Dr. Irene Tsapara

Date: Feb. 23, 2025

Introduction

Handwritten digit recognition is a principal task in computer vision with the most referred applications in sorting postal mail, processing bank checks, and automated document scanning. The aim of this research is to carry out the experimentation of different neural network architectures for digit classification using the well-known dataset MNIST. This disparity holds a 2x2 completely crossed experimental design, which puts in comparison the number of layers (2, 5) and 128 or 256 neuron configuration per layer. The models will be employed and evaluated based on accuracy, training period, and generalization. Therefore, the results would suggest a decision about accepting or rejecting trade-offs between the complexity of the model and efficiency in computation.

Data Presentation

The dataset used in this research is the MNIST (Modified National Institute of Standards and Technology) dataset. It consists of grayscale images of handwritten digits for the digits from 0 to 9, each represented in a 28x28 pixel matrix. All pixels have intensity values numbered from 0 to 255. The dataset consists of 42,000 training images and 28,000 test images.

This dataset has been made available under the Creative Commons Attribution-ShareAlike 3.0 (CC BY-SA 3.0) License, which allows redistribution and adaptation as long as appropriate credit is given. That dataset turned out to be cited in numerous studies and benchmark evaluations over the years.

Preprocessing

The dataset contains images of hand-written digits provided with labels. The training set includes labeled observations, while the testing set, has unlabeled images for model evaluation. Each image presents a string of flattened pixel values.

For the purpose of preparing the data for training, pixels were adjusted to the range of [0, 1] to maintain some numeric stability. The training data was then split into two sets: the training set and the validation set for evaluation.

Methodology

This study follows an experimental design approach to systematically compare different neural network architectures. The experimental setup consists of four models:

- Model 1: 2 layers, 128 neurons per layer
- Model 2: 2 layers, 256 neurons per layer
- Model 3: 5 layers, 128 neurons per layer
- Model 4: 5 layers, 256 neurons per layer

Besides being a typical feedforward architecture with hidden layers using ReLU activation and softmax as the output, Batch Normalization and Dropout with a rate of 0.3 were also employed to enhance generalization and retain overfitting. The RMSprop optimizer was trained with a learning rate of 0.001, and the batch size was set to 128. The models were trained for 40 epochs. Additionally, the ReduceLROnPlateau was utilized to ensure a dynamic adjustment of the learning rate when the validation accuracy achieved a plateau.

Results & Evaluations

The performance evaluation of four experimental neural networks with different numbers of layers and neurons shows that all models have strong classification capabilities. Each model is evaluated based on training time, training accuracy, and validation accuracy to measure its efficiency and generalization ability.

To analyze the learning dynamics, we generated loss curves for each model to illustrate how the training and validation losses evolved over 40 epochs. The loss curve for the model with two layers and 128 neurons (Figure 1) shows steady convergence, with validation loss stabilizing after 10 epochs but fluctuating slightly. The confusion matrix for this model (Figure 2) shows that all digits were classified with high accuracy, with only a few misclassifications. The ROC curve (Figure 3) shows a near-perfect area under the curve (AUC), confirming the model's excellent performance, while the precision-recall curve (Figure 4) shows that the confidence in the classifications is high.

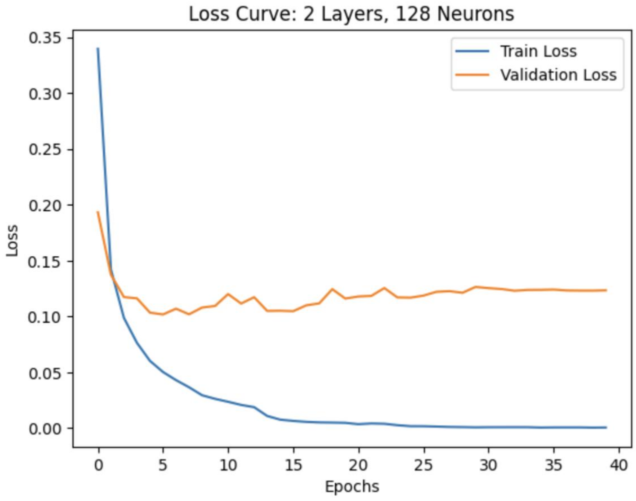
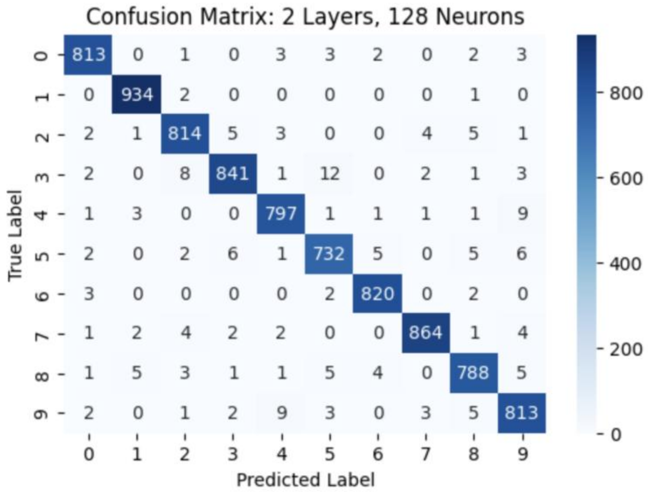


Figure 1 - Loss Curve: 2 Layers, 128 Neurons



True Label \ Predicted Label	0	1	2	3	4	5	6	7	8	9
0	813	0	1	0	3	3	2	0	2	3
1	0	934	2	0	0	0	0	0	1	0
2	2	1	814	5	3	0	0	4	5	1
3	2	0	8	841	1	12	0	2	1	3
4	1	3	0	0	797	1	1	1	1	9
5	2	0	2	6	1	732	5	0	5	6
6	3	0	0	0	0	2	820	0	2	0
7	1	2	4	2	2	0	0	864	1	4
8	1	5	3	1	1	5	4	0	788	5
9	2	0	1	2	9	3	0	3	5	813

Figure 2 - Confusion Matrix: 2 Layers, 128 Neurons

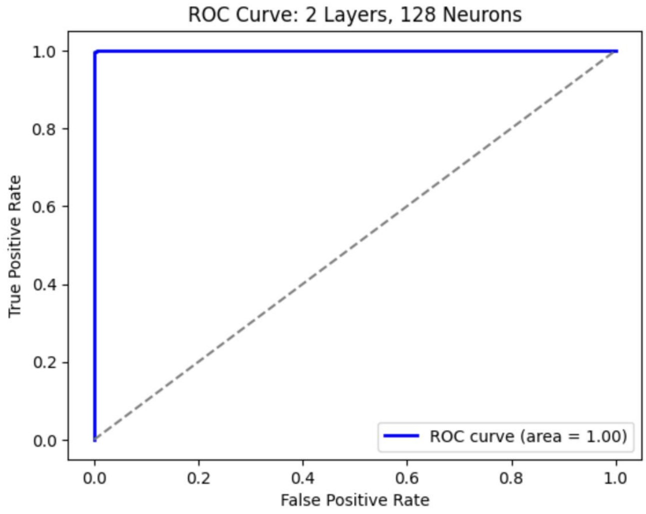


Figure 3 - ROC Curve: 2 Layers, 128 Neurons

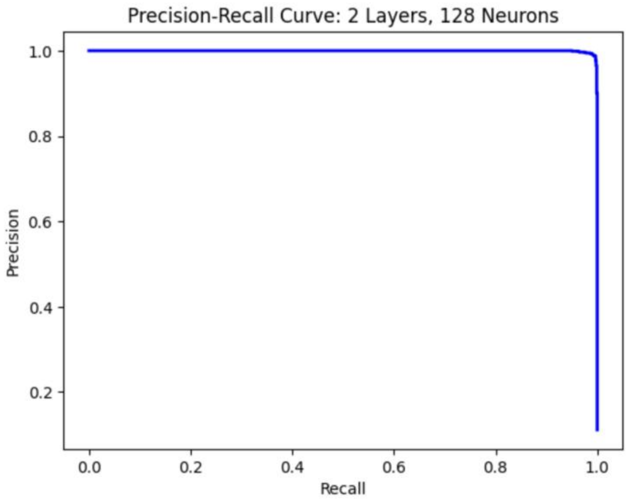


Figure 4 - Precision-Recall Curve: 2 Layers, 128 Neurons

For the model with two layers and 256 neurons, the loss curve (Figure 5) shows improved convergence compared to the model with 128 neurons. The corresponding confusion matrix (Figure 6) shows a further reduction in misclassifications. In addition, the ROC curve (Figure 7) maintains an AUC of 1.00, confirming the model's ability to distinguish between digit categories. Similarly, the precision-recall curve (Figure 8) remains stable, reinforcing the model's strong classification performance.

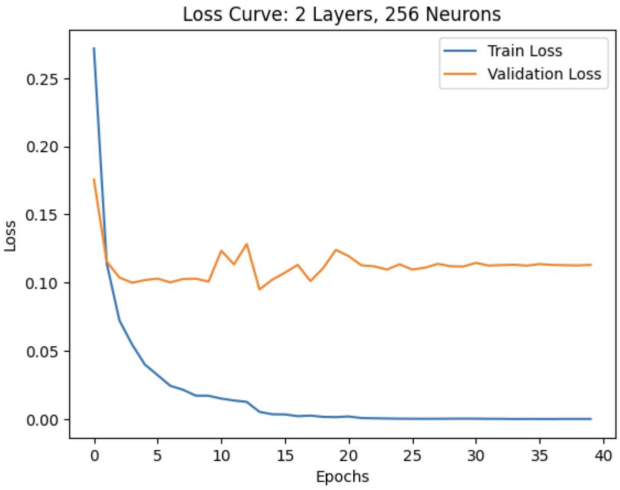
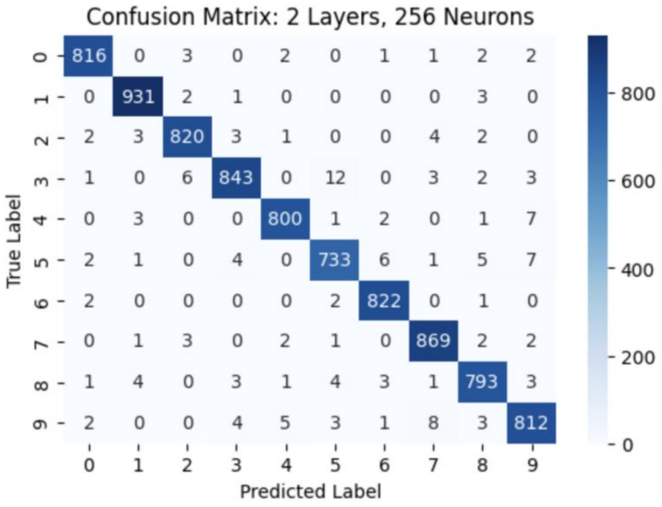


Figure 5 - Loss Curve: 2 Layers, 256 Neurons



	0	1	2	3	4	5	6	7	8	9
0	816	0	3	0	2	0	1	1	2	2
1	0	931	2	1	0	0	0	0	3	0
2	2	3	820	3	1	0	0	4	2	0
3	1	0	6	843	0	12	0	3	2	3
4	0	3	0	0	800	1	2	0	1	7
5	2	1	0	4	0	733	6	1	5	7
6	2	0	0	0	0	2	822	0	1	0
7	0	1	3	0	2	1	0	869	2	2
8	1	4	0	3	1	4	3	1	793	3
9	2	0	0	4	5	3	1	8	3	812

Figure 6 - Confusion Matrix: 2 Layers, 256 Neurons

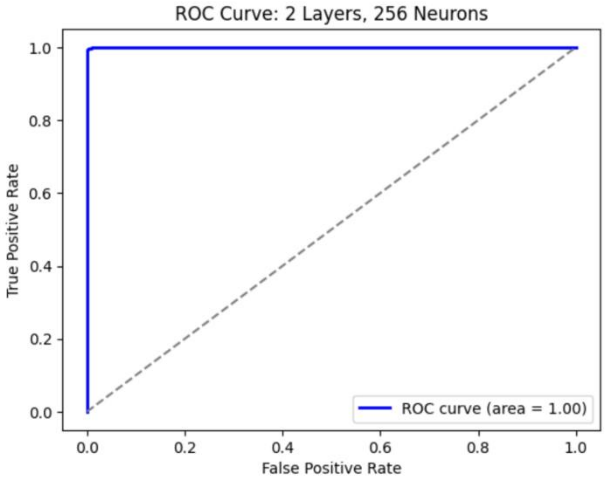


Figure 7 - ROC Curve: 2 Layers, 256 Neurons

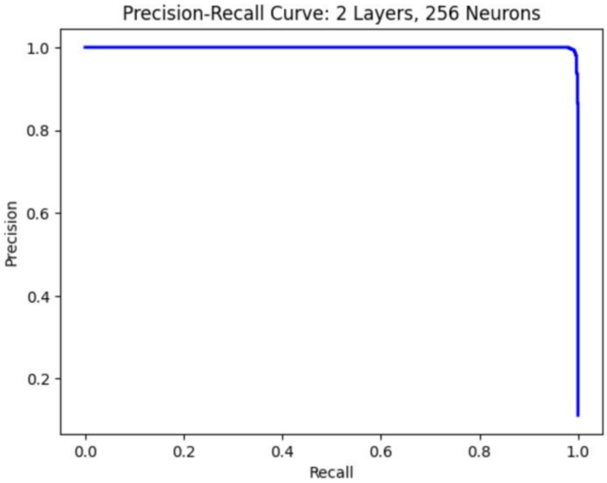


Figure 8 - Precision-Recall Curve: 2 Layers, 256 Neurons

When examining models with deeper architectures, the loss curve for the model with five layers and 128 neurons (Figure 9) shows that convergence takes longer, but validation loss remains stable after several epochs. The confusion matrix (Figure 10) shows that this model performs comparable to the two-layer architecture, though with slightly more misclassifications. The ROC curve (Figure 11) continues to show near-optimal classification performance, and the precision-recall curve (Figure 12) remains consistent with the previous model, reinforcing strong classification capabilities.

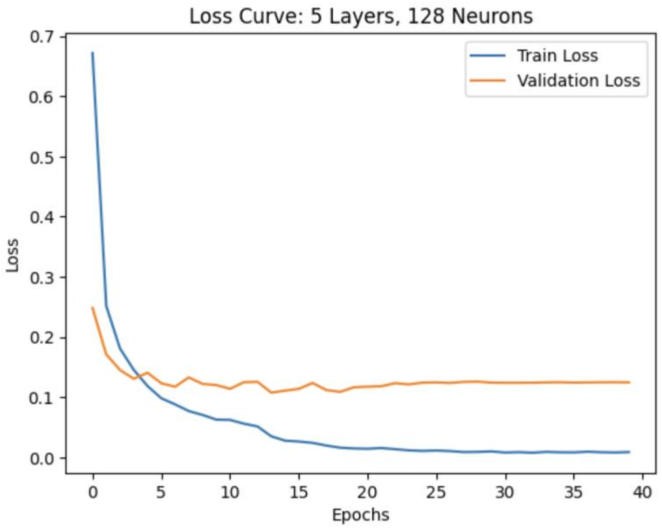


Figure 9 - Loss Curve: 5 Layers, 128 Neurons

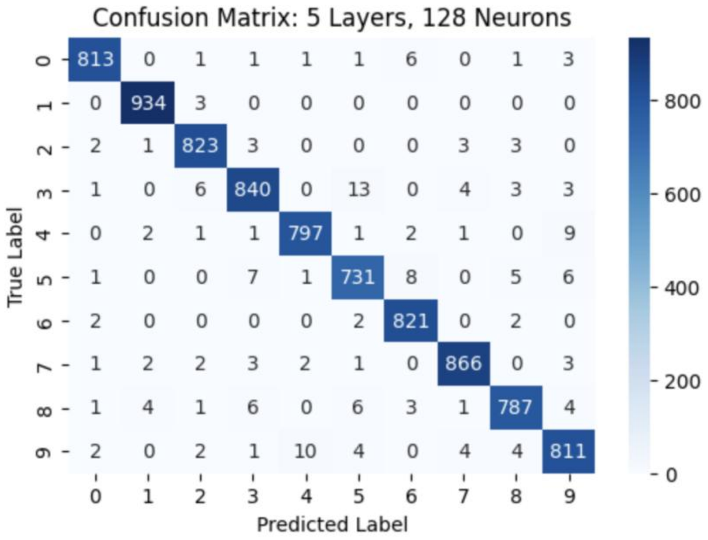


Figure 10 - Confusion Matrix: 5 Layers, 128 Neurons

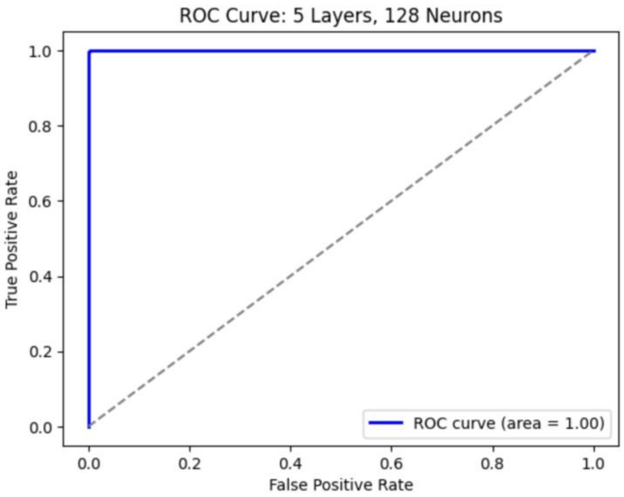


Figure 11 - ROC Curve: 5 Layers, 128 Neurons

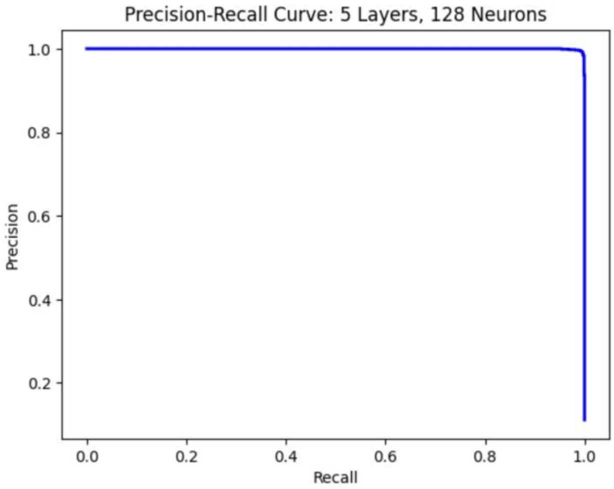


Figure 12 - Precision-Recall Curve: 5 Layers, 128 Neurons

The final model, with five layers and 256 neurons, exhibited the best convergence behavior, as seen in its loss curve (Figure 13). This model had the lowest training loss and maintained stable validation loss throughout the training process. The confusion matrix (Figure 14) indicated the lowest misclassification rate among all tested models. The ROC curve (Figure 15) maintained an AUC of 1.00, further confirming the model's effectiveness. The precision-recall curve (Figure 16) still demonstrates a high level of confidence.

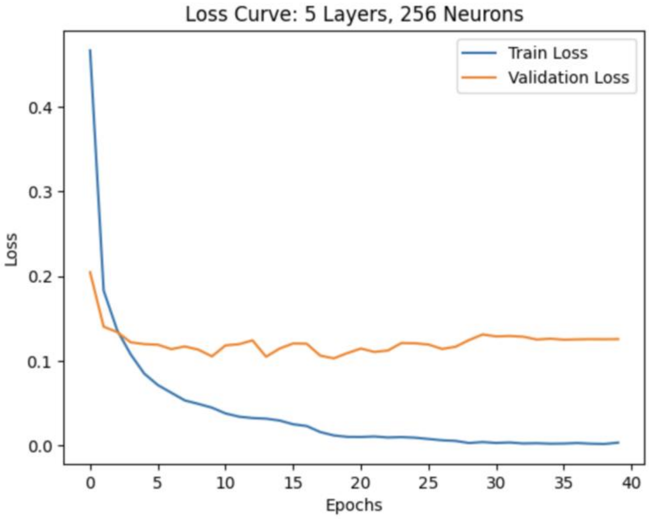
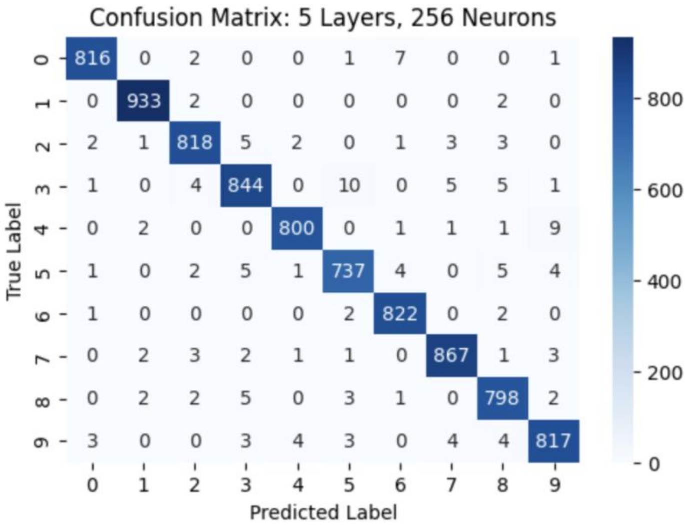


Figure 13 - Loss Curve: 5 Layers, 256 Neurons



	0	1	2	3	4	5	6	7	8	9
0	816	0	2	0	0	1	7	0	0	1
1	0	933	2	0	0	0	0	0	2	0
2	2	1	818	5	2	0	1	3	3	0
3	1	0	4	844	0	10	0	5	5	1
4	0	2	0	0	800	0	1	1	1	9
5	1	0	2	5	1	737	4	0	5	4
6	1	0	0	0	0	2	822	0	2	0
7	0	2	3	2	1	1	0	867	1	3
8	0	2	2	5	0	3	1	0	798	2
9	3	0	0	3	4	3	0	4	4	817

Figure 14 - Confusion Matrix: 5 Layers, 256 Neurons

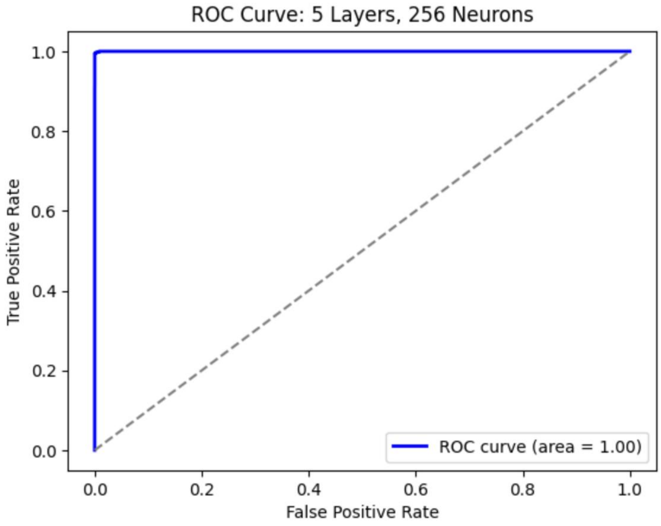


Figure 15 - ROC Curve: 5 Layers, 256 Neurons

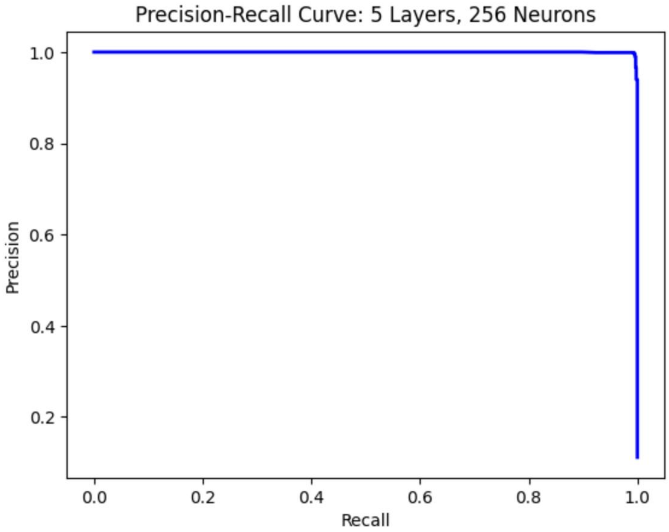
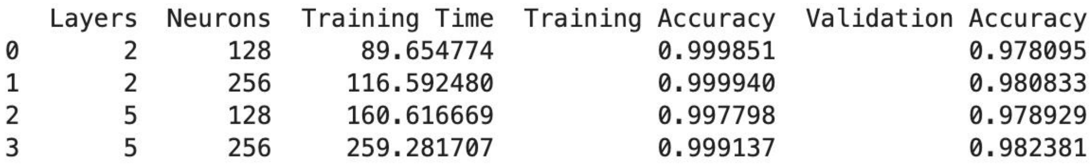


Figure 16 - Precision-Recall Curve: 5 Layers, 256 Neurons

In summary, the results (Table 1) indicate that increasing the number of neurons per layer consistently improves classification accuracy, while increasing the number of layers leads to significantly longer training times without necessarily yielding substantial accuracy gains. The best-performing model, in terms of validation accuracy, was the five-layer, 256-neuron network, achieving a validation accuracy of 0.9824. However, considering computational efficiency, the two-layer, 256-neuron model offered nearly equivalent accuracy with a much lower training time, making it a more practical choice. These findings suggest that a well-balanced neural network architecture, rather than the deepest or most complex model, is often the most effective for achieving high performance while maintaining computational efficiency.



```
   Layers  Neurons  Training Time  Training Accuracy  Validation Accuracy
0       2      128      89.654774           0.999851             0.978095
1       2      256     116.592480           0.999940             0.980833
2       5      128     160.616669           0.997798             0.978929
3       5      256     259.281707           0.999137             0.982381
```

Table 1. Summary of Training Results

Management/Research Question

How can we accurately classify handwritten digits using machine learning models, and which model configurations provide the best performance?

Conclusion

This study examined the impact of network depth, specifically two versus five layers, and neuron count, specifically 128 versus 256 neurons per layer, on the accuracy and efficiency of handwritten digit classification models. The results show that deeper models with five layers generally perform better than shallower models with two layers, but at the expense of increased training time. Batch normalization and dropout were effective in improving generalization and preventing overfitting. ReduceLROnPlateau dynamically adjusted learning rates, leading to stable training and better convergence.

To further optimize performance, future work could implement early stopping to avoid unnecessary training cycles and prevent overfitting. Experimenting with different optimizers such as SGD or Adam may provide insights into convergence speed and accuracy differences. Additionally, convolutional neural networks (CNNs) should be explored for improved feature extraction and classification accuracy.

References

Deng, L. (2012). "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]." *IEEE Signal Processing Magazine, vol. 29, no. 6*, pp. 141-142, Nov. 2012, doi: 10.1109/MSP.2012.2211477. https://ieeexplore.ieee.org/document/6296535

MNIST Dataset License: CC BY-SA 3.0. Retrieved from: https://www.kaggle.com/datasets/hojjatk/mnist-dataset