Comparative analysis of different model depths on convolutional neural network for handwritten digit recognition

Weitao Deng

School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, United Kingdom

ec21834@qmul.ac.uk

Abstract. In the rapidly advancing field of artificial intelligence, Convolutional Neural Networks (CNNs), as a representative method, have risen to prominence as a pivotal instrument for handling visual data. However, despite their widespread use, the impact of CNN depth on performance remains under-explored. This study delves into this aspect, evaluating the performance of CNN architectures with different depths - two-layer, four-layer, and five-layer - on the MNIST dataset, a version from the National Institute of Standards and Technology, a well-known benchmark dataset for handwritten digit recognition. Experimental results reveal that the four-layer model achieved the highest average accuracy of 99.76%, while the five-layer model, despite its additional complexity, only slightly trailed behind with a 99.73% accuracy rate. However, the five-layer model required a significantly longer training time. In conclusion, while deeper networks can increase accuracy, they can also introduce computational inefficiencies without significant gains in performance. This research provides a better understanding of CNN depth, guiding optimal model selection for image classification tasks.

Keywords: Convolutional Neural Network, MNIST, Handwritten Digit Recognition.

1. Introduction

Empowered by machine learning, artificial intelligence has been revolutionized, enabling computers to derive insights from data and make informed choices. As a representative work, deep learning has garnered significant attention for its ability to model complex patterns in data. Deep learning algorithms, particularly neural networks, have been instrumental in advancing various fields such as healthcare, finance, and autonomous systems. These algorithms have been especially effective in tasks that involve high-dimensional data, including image and speech recognition.

Among the various neural network architectures, Convolutional Neural Networks (CNNs) have established themselves as a formidable instrument for managing visual information [1]. CNNs are engineered to autonomously and dynamically grasp hierarchical structures of features, rendering them exceptionally suitable for activities such as image categorization, object detection, and even medical image analysis. CNNs have also been adapted for sequence data like time-series and natural language, demonstrating their versatility and broad applicability.

However, despite the extensive research and myriad applications, there is a gap in the understanding of how the depth of a CNN affects its performance [2]. While it is generally believed that deeper networks can model more complex features, the computational cost and risk of overfitting also increase with depth. Therefore, it is crucial to understand the trade-offs involved in choosing the depth of a CNN model.

In light of this, this work is conducted for filling the gap by conducting a comprehensive analysis of CNN architectures with varying depths, specifically focusing on two-layer, four-layer, and five-layer models. This work employs the MNIST dataset, a benchmark dataset in the field of machine learning, for evaluation [3]. The MNIST collection contains grayscale pictures of scribbled numbers and is frequently employed to train diverse visual processing platforms. Experiments have been meticulously designed and executed, and this work presents a detailed analysis of the results. One of the configurations achieved a maximum average accuracy of 99.76%, a promising result that warrants further investigation. To provide a more complete picture, this paper also includes line graphs that depict the trends in train and test loss, as well as train and test accuracy over the course of the training process [4].

This research aims to offer valuable insights into the optimal configuration of CNN layers for tasks involving image classification. It is believed that findings will serve as a useful resource for both academia and industry, aiding in the design and implementation of more efficient and effective CNN models for various applications.

2. Method

2.1. Dataset

The MNIST dataset is a collection of handwritten digits, widely recognized and utilized in the machine learning community as a benchmark for evaluating various algorithms. The dataset encompasses 70,000 pictures in total (60,000 for training and 10,000 for testing). Images in this dataset are 28x28 pixels, showcasing grayscale depictions of hand-drawn numbers ranging from 0 through 9. Each image is labeled with its corresponding digit. Representative examples are demonstrated in Figure 1.

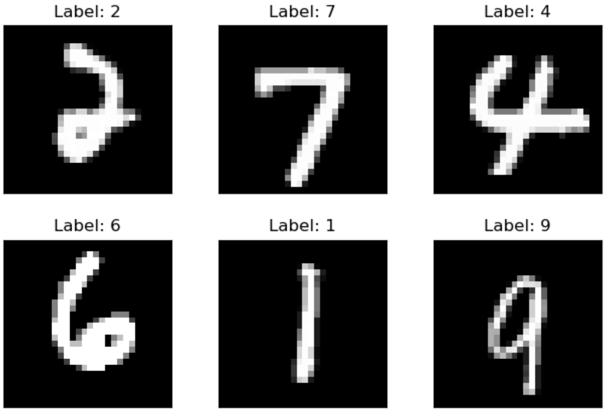

Figure 1. Visualization of representative images in MNIST dataset [3].

2.2. Models

CNN serves as the foundational model for this research. CNNs excel in image classification efforts because of their ability to autonomously and dynamically identify hierarchical attribute structures in the provided visual images.

The central aim of this research was to explore the effects of altering the layer count in the CNN architecture. Three architectures were explored: two-layer, four-layer, and five-layer models.

2.2.1. Two-Layer Model

The two-layer model contains the following components.

1) A convolutional layer (CL) with 32 nodes of 5x5 kernel size, activated by the ReLU and succeeded by batch normalisation.

2) Another CL with 32 nodes of size 5x5, activated by ReLU, and succeeded by batch normalization, max-pooling, and dropout (25%).

3) A dense layer comprising 128 units, succeeded by ReLU activation and a dropout rate of 50%.

4) Another fully connected layer with 10 neurons.

2.2.2. Four-Layer Model

The four-layer model contains the following components.

1) A CL contain 32 filters of 5x5 size, succeeded by the ReLU activation and batch normalisation.

2) A CL contain 32 filters of size 5x5, succeeded by ReLU activation, batch normalization, max-pooling, and dropout (25%).

3) A CL contain 64 filters of size 3x3, succeeded by ReLU activation and batch normalization.

4) Another CL contain 64 filters of size 3x3, succeeded by ReLU activation, batch normalization, max-pooling, and dropout (25%).

5) A fully connected layer contain 256 neurons, followed by ReLU activation and dropout (50%).

6) A fully connected layer contain 10 neurons.

2.2.3. Five-Layer Model

The five-layer model contains the following components.

1) A CL contain 32 filters of 5x5 size, succeeded by the ReLU activation and batch normalisation.

2) A CL contain 32 filters of size 5x5, succeeded byby ReLU activation, batch normalization, max-pooling, and dropout (25%).

3) A CL contain 64 filters of size 3x3, succeeded by ReLU activation and batch normalization.

4) A CL contain 64 filters of size 3x3, succeeded by ReLU activation, batch normalization, max-pooling, and dropout (25%).

5) A CL contain 128 filters of 3x3 size, succeeded by ReLU activation and batch normalization.

6) Fully connected layer contain 256 neurons, followed by ReLU activation and dropout (50%).

7) Fully connected layer contain 10 neurons.

2.2.4. Advantages and Limitations

The advantage of using CNNs for this task lies in their ability to capture local patterns and hierarchies, making them highly effective for image recognition. The exploration of different layer depths aims to identify an optimal model complexity that balances accuracy and computational efficiency. However, increasing the depth might lead to overfitting, especially if not complemented with appropriate regularization techniques.

2.3. Evaluation Metrics

To gauge the model's efficacy, multiple assessment measures were utilized.:

Accuracy: This metric provides a general measure of correctness of prediction.

Precision: This metric evaluates how many positive predictions are actually correct.

Recall: This assesses how many of the true positive instances were correctly identified by the model.

F1-Score: A metric derived from both precision and recall, providing a balanced view of their performance.

The classification report generated post-training provides a detailed breakdown of these metrics for each class, enabling a comprehensive evaluation of the model's performance.

3. Results

3.1. Training Details

The experiments were conducted using a CNN-based architecture with varying depths. The following hyperparameters and settings were employed across all models. All models' weights were initialized using the Kaiming initialization method, which is known for its effectiveness in deep networks. As per the provided code, the learning rate was set to 0.001. Cross-Entropy Loss was employed given its aptness for tasks involving multi-class categorization. The Adam optimizer was employed for its adaptive learning rate properties. A batch size of 128 was used for training and training epoch is 50. The dataset was normalized to have values between 0 and 1. No other significant data augmentation techniques were applied. 80% for the dataset is leveraged for training, 10% for validation, and the rest for testing.

3.2. Classification Results

The classification results for the three models with varying depths are as follows:

The two-layer model achieved a maximum average accuracy of 99.59%. The detailed results were shown in Figure 2 and Table 1. The model took approximately 79.0 minutes to train.

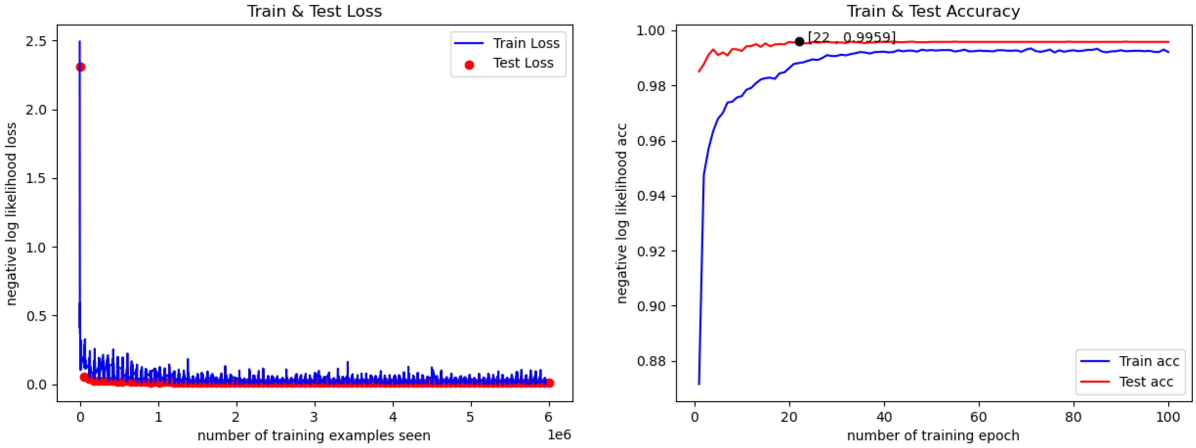

Figure 2. Loss and accuracy curves of the two-layer model (Figure Credits: Original).

Table 1. Evaluation indicators for two-layer model with accuracy at 0.99.

	Precision	Recall	F1 - Score
Number0	1.00	0.99	0.99
Number1	0.96	0.99	0.98
Number2	0.98	0.99	0.99
Number3	0.96	0.99	0.98
Number4	0.99	0.99	0.99
Number5	0.99	0.98	0.99
Number6	1.00	0.98	0.99
Number7	1.00	0.98	0.99
Number8	1.00	0.98	0.99
Number9	1.00	0.98	0.99
Macro Mean	0.99	0.99	0.99
Weighted Mean	0.99	0.99	0.99

The four-layer model attained a maximum average accuracy of 99.76%. Figure 3 and Table 2 demonstrated in results of the training process. The training time for this model was approximately 89.0 minutes.

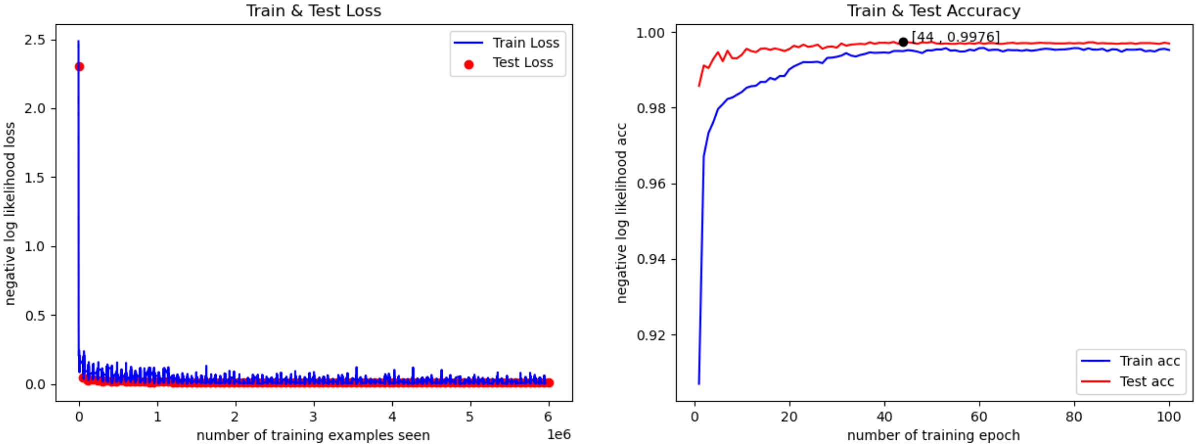

Figure 3. Loss and accuracy curves of the four-layer model (Figure Credits: Original).

Table 2. Evaluation indicators for Four-layer model with accuracy at 0.99.

	Precision	Recall	F1 - Score
Number0	0.99	0.99	0.99
Number1	1.00	0.99	0.99
Number2	1.00	0.99	0.99
Number3	0.99	0.99	0.99
Number4	0.98	0.99	0.99
Number5	1.00	0.98	0.99
Number6	1.00	0.99	0.99
Number7	0.95	0.99	0.97
Number8	1.00	0.99	0.99
Number9	0.98	0.99	0.98
Macro Mean	0.99	0.99	0.99
Weighted Mean	0.99	0.99	0.99

The five-layer model reached a maximum average accuracy of 99.73%. The intermediate results were displayed in Figure 4 and Table 3. This model had a significantly higher training time of 287.0 minutes.

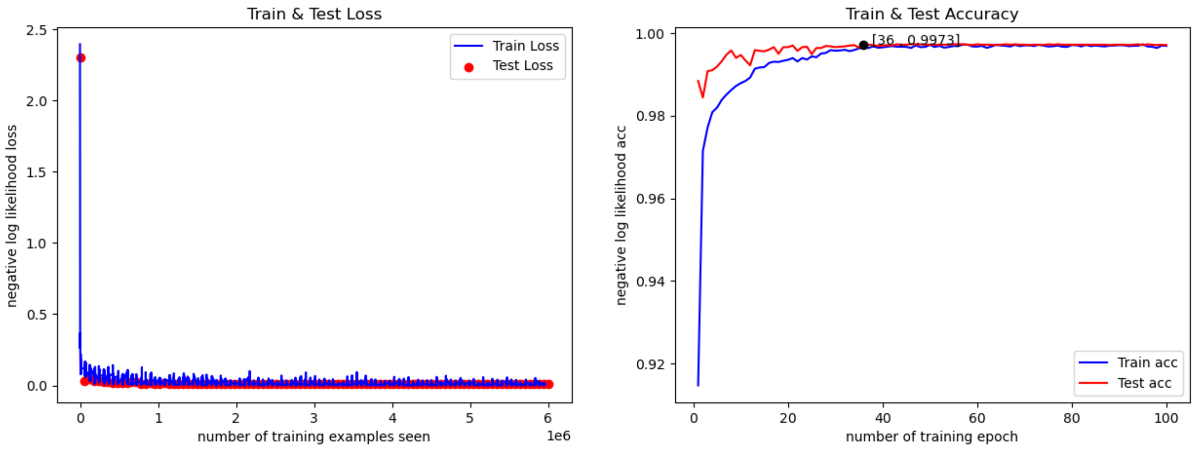

Figure 4. Loss and accuracy curves of the five-layer model (Figure Credits: Original).

Table 3. Evaluation indicators for Five-layer mode with accuracy at 0.99.

	Precision	Recall	F1 - Score
Number0	1.00	0.99	0.99
Number1	0.96	0.99	0.97
Number2	0.95	0.99	0.97
Number3	0.99	0.99	0.99
Number4	1.00	0.99	0.99
Number5	1.00	0.98	0.99
Number6	1.00	0.99	0.99
Number7	1.00	0.98	0.99
Number8	1.00	0.99	0.99
Number9	1.00	0.99	0.99
Macro Mean	0.99	0.99	0.99
Weighted Mean	0.99	0.99	0.99

3.3. Impact of Model Depth on Classification Results

To understand the impact of varying depths on the classification performance, Table 4 summarizes the results.

Table 4. Performance comparison of various models.

Model Depth	Max Avg Accuracy	Training Time (mins)
Two-layer	99.59%	79.0
Four-layer	99.76%	89.0
Five-layer	99.73%	287.0

From the table, it's evident that while increasing the depth from two to four layers improves the accuracy, further increasing the depth to five layers doesn't provide a significant boost in performance. However, the training time for the five-layer model is considerably higher, indicating potential overfitting or increased computational complexity.

4. Discussion

4.1. Discussion on Performance Discrepancies Among Models

Depth and Model Complexity: The relationship between depth and model complexity is not always linear. While deeper networks can capture more intricate features, they can also introduce redundancy. The marginal improvement in accuracy from the four-layer to the five-layer model suggests that there might be a saturation point beyond which adding more layers doesn't significantly enhance performance [5].

Training Time and Model Depth: As observed, the five-layer model took considerably longer to train compared to its counterparts. This increased training time can be attributed to the added computational complexity introduced by the additional layers [6].

Regularization and Overfitting: The longer training time for the five-layer model might also hint at potential overfitting. While deeper models can fit the training data more closely, they might not generalize well to unseen data. The use of dropout in the models serves as a regularization technique, but its effectiveness can vary with model depth [7].

4.2. Limitations

Lack of Data Augmentation: The models were trained without significant data augmentation. Incorporating data augmentation techniques could potentially improve the generalization capabilities of the models, especially for deeper architectures [8].

Computational Complexity: Deeper models, while potentially more accurate, demand more computational resources. This can be a limitation when deploying models in resource-constrained environments or for real-time applications [9].

4.3. Future Work and Applications

Exploration of Other Regularization Techniques: Future work could delve into other regularization methods like weight decay or early stopping to prevent overfitting, especially in deeper models [10].

Experiments with Larger Datasets: To further validate the findings, experiments can be conducted on larger or more complex datasets. This would provide a clearer picture of how model depth impacts performance in diverse scenarios [11].

Potential Applications of CNNs: The versatility of CNNs extends beyond handwritten digit recognition. They hold promise in fields like medical image analysis, autonomous driving, and security surveillance [12].

Challenges in Real-time Applications: While deeper models might offer better accuracy, their computational demands can pose challenges in real-time applications where quick predictions are essential [13].

5. Conclusion

The exploration of CNN architectures with varying depths on the MNIST dataset has provided valuable insights into the intricate balance between model complexity and performance. While deeper architectures, such as the four-layer model, demonstrated superior performance, the five-layer model's marginal improvement came at a significant computational cost. This research underscores the nuanced relationship between depth and efficiency in CNNs. The findings suggest that blindly increasing model depth might not always yield proportional performance gains and could introduce computational inefficiencies. As the field of deep learning continues to evolve, it becomes imperative for researchers and practitioners to make informed decisions about model architectures, considering both accuracy and computational demands. This study serves as a stepping stone, guiding the community towards optimal model selection for image classification tasks, ensuring both effectiveness and efficiency.

References

[1] Ciregan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In 2012 IEEE conference on computer vision and pattern recognition, 3642-3649.

[2] Abbas, A., Abdelsamea, M. M., & Gaber, M. M. (2021). Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. Applied Intelligence, 51, 854-864.

[3] Yadav, S. S., & Jadhav, S. M. (2019). Deep convolutional neural network based medical image classification for disease diagnosis. Journal of Big data, 6(1), 1-18.

[4] Srinivasu, P. N., SivaSai, J. G., Ijaz, M. F., Bhoi, A. K., Kim, W., & Kang, J. J. (2021). Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. Sensors, 21(8), 2852.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778.

[6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1-9.

[7] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, 448-456.

[8] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 113-123.

[9] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4510-4520.

[10] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

[11] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115, 211-252.

[12] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., et al. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60-88.

[13] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.