Robust Neural Network Optimization and Evaluation for Handwritten Digit Recognition

Veera Harshatheeswar Reddy Beemcharla
veera.beemcharla@city.ac.uk

Abstract

In this paper, the Multilayer Perceptron (MLP) and Convolutional Neural Networks (CNN), two models for image classification tasks, are thoroughly evaluated using the MNIST dataset. Data loading and preprocessing are the first steps in the study, and then noise is added to the data to compare the ability of models to correctly classify the noisy images. The resilience of the models is then evaluated by analysing their performances at different noise levels. Incorporating noise as a part of data augmentation during training to increase the model's robustness to noisy inputs, Grid search is used to adjust hyperparameters once baseline models for CNN and MLP have been built. The outcomes reveal that although both models perform well, the CNN outperforms the MLP in terms of accuracy and resilience, particularly when noise levels rise, making CNN the better option for this classification task

1. Introduction:

Task handwritten digit recognition is one of the fundamental problems in the field of computer vision. The MNIST dataset serves as a benchmark for evaluating the efficiency of many machine learning algorithms, the dataset comprise of 70,000 grayscale images of handwritten digits from 0 to 9 which offers best platform to compare different classification models

The main objective of this paper is to critically assess and compare two prominent neural network architectures— the Feedforward Multilayer Perceptron (MLP) and the Convolutional Neural Network (CNN)—which is used for exploring various configurations and training methodologies, we aim to elucidate the strengths and limitations of each model, particularly in their ability to generalize and maintain robustness against noisy inputs.

1.1. multilayer perceptron(MLP):

MLP is artificial neural network that belongs to the class of feedforward neural networks
Which is widely used in different pattern recognition and image classification tasks,
The learning process in an MLP is primarily governed by the backpropagation algorithm, which is used to minimize the loss function by adjusting the weights of the network through gradient descent [1], One of the key strengths of MLPs is their flexibility and ability to approximate any continuous function given sufficient neurons in the hidden layers, a property known as the universal approximation theorem[2]. But MLPs also have limitations, mainly in scalability and efficiency when dealing with high-dimensional data like images. Due to the fully connected nature of the network, MLPs can become computationally expensive and prone to overfitting, especially when the input data has a large number of features[3].

1.2 Convolutional neural Network(CNN):

cnns are powerful class of deep learning models specially known for processing data with a great structure such as images[4], cnns became cornerstone for modern computer vision task due to their ability to learn spatial hierarchies of features from original data automatically.

The effectiveness of CNNs in image classification and other computer vision tasks is largely due to their ability to learn relevant features directly from pixel data, without the need for manual feature extraction. This characteristic has led to CNNs being widely adopted in various applications, including object detection, facial recognition, and medical image analysis

2. Dataset:

The MNIST dataset, which consists of handwritten digit pictures ranging from 0 to 9, was utilised for research and testing in this project. The dataset is frequently used for training and testing different image classification models, and it is a typical benchmark in the field of machine learning. The 70,000 images in the MNIST dataset are divided into 10,000 images for testing and 60,000 images for training. Every image is a 28x28 pixel greyscale picture that has been normalised to guarantee that pixel values fall between 0 and 1, obviating the need for additional normalisation.

The dataset exhibits a good degree of balance among the 10 digit classes, with around equal numbers of samples representing each digit class (0–9). This balancing makes sure that there isn't a big class imbalance, which makes it possible to train the model simply without using methods like SMOTE to deal with class distribution problems.

3. methodology:

MNIST dataset is one of the most widely used dataset in the field of image classification. It consists of 60,000 training images and 10,000 test images. Each image is grayscale handwritten digit from 0 to 9 and the task is it to properly classify each image into one of the ten digit classes. This study employs two types of neural network models for this task: a Multi-Layer Perceptron (MLP) and a Convolutional Neural Network (CNN). MLP is a fully connected feedforward neural network that is effective for simpler tasks, however, it may struggle with spatial hierarchies within the data this makes it less effective for images. To combat this issue this paper employs CNN which is specifically designed to handle image data by capturing spatial hierarchies through its convolutional layers. It is well suited for datasets like MNIST, since spatial arrangement of pixels is crucial for distinguishing between digits.

3.1. Architecture of MLP:

In this study, the MLP starts with an input layer that takes the 784 features derived from the 28x28 pixel images. This is followed by several hidden layers and each of these layers consists of neurons connected to all neurons in the previous layer. To increase the complexity of the model, the first hidden layer has 256 neurons and the ReLU (Rectified Linear Unit) activation function is implemented to introduce non-linearity. This implementation allows the network to learn complex patterns in the data. To prevent overfitting and enhance generalization, dropout is applied after each hidden layer, this technique forces the network to learn more robust features without having to rely too heavily on any particular neuron. The final layer is the output layer, which consists of 10 neurons corresponding to the 10 digit classes. A softmax activation function is applied to the output layer, to convert the raw output scores into probabilities that sum to one.

3.2 Architecture of CNN:

This study using CNN with a series of convolutional layers that apply filters to the input images, detecting features such as edges, textures, and shapes. Each convolutional layer is followed by a ReLU activation function and a max-pooling layer to downsample the feature maps. This reduces their spatial dimensions while retaining the most important features. In this study, the first convolutional layer applies 32 filters of size 3x3 to the input images, followed by a max-pooling layer that reduces the feature map size by half. The second convolutional layer uses 64 filters, further increasing the complexity of the features captured. Then the network transitions to fully connected layers to classify the input image into one of the 10 digit classes. Dropout is applied to the fully connected layers to prevent overfitting, and the finally the output layer uses a softmax activation function.

3.3. Data Augmentation with Noise and Hyper Parameter optimization:

An essential approach of this study applyiing noise to the dataset and tthen training the models. Data augmentation through the addition of noise is a technique used to artificially expand the training dataset and improve model robustness. Once the random noise is introduced to the input images, the models are forced to learn more general features rather than memorizing specific patterns in the training data. This study takes the approach of applying Gaussian noise to the images in the MNIST dataset. Gaussian noise is characterized by a mean of zero and a standard deviation that controls the amount of noise added. Adding noise is a form of regularization and forces the model to identify the correct digit even when the input image is slightly distorted. This noise augmented dataset is then used to train both the MLP and CNN models and then evaluated.

3.4. Hyper parameter architecture of MLP:

The hyper parameter model of MLP begins with an input layer that corresponds to the flattened 28x28 pixel images from the dataset, resulting in 784 input neurons. What sets this apart is that the hidden layer contains 64, 128, or 256 neurons, with each neuron applying a ReLU (Rectified Linear Unit) activation function. then final layer is a softmax output layer with 10 neurons, corresponding to the 10 possible digit classes. And the learning rate, batch size, and zero grad optimizer is implemented along with cross-entropy is used to calculate the loss with the number of epochs set at 10. The model's performance is evaluated using cross-entropy loss, and various learning rates (0.001, 0.01, and 0.1) are explored to identify the best configuration.

3.5. Hyper parameter architecture of CNN:

The hyper parameter CNN consists of convolutional layers that apply 3x3 kernels to the input images. The first convolutional layer extracts 32 feature maps from the input images. Output is then passed through a ReLU activation function and is followed by a max-pooling layer that reduces the spatial dimensions by half. The second convolutional layer applies 64 filters, followed by another ReLU activation and max-pooling layer. The output of the convolutional and pooling layers is then flattened into a 1D tensor and is provided as input to a fully connected layer with 128 neurons, again using ReLU activation. Finally, the output layer consists of 10 neurons corresponding to the 10 classes.

4. Results findings and evaluation:

4.1 model evaluation:

Both the MLP and CNN models achieves high accuracies in testing and training, but the CNN model appears to have a slightly higher test accuracy when compared to the MLP, which is indicated by the red circle being slightly above the orange cross. This suggests that the CNN model might be better at generalizing to unseen data.

This comparison highlights the effectiveness of CNNs, particularly for image data like MNIST, where spatial hierarchies and local patterns are important. The MLP, while effective, may not capture these patterns as efficiently, which is reflected in its slightly lower test accuracy

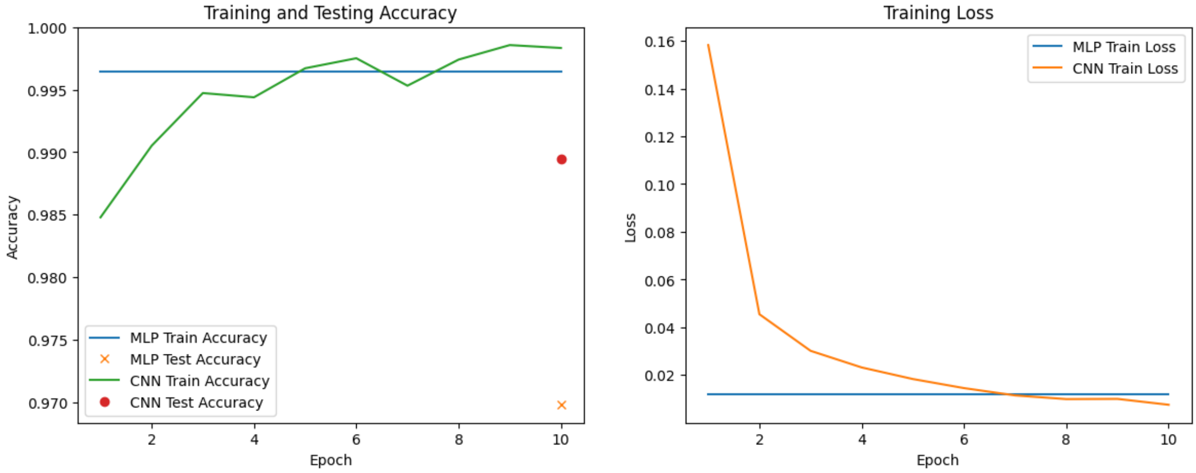


Figure 1: Comparison of training ,testing accuracy and training loss between baseline MLP and CNN model.

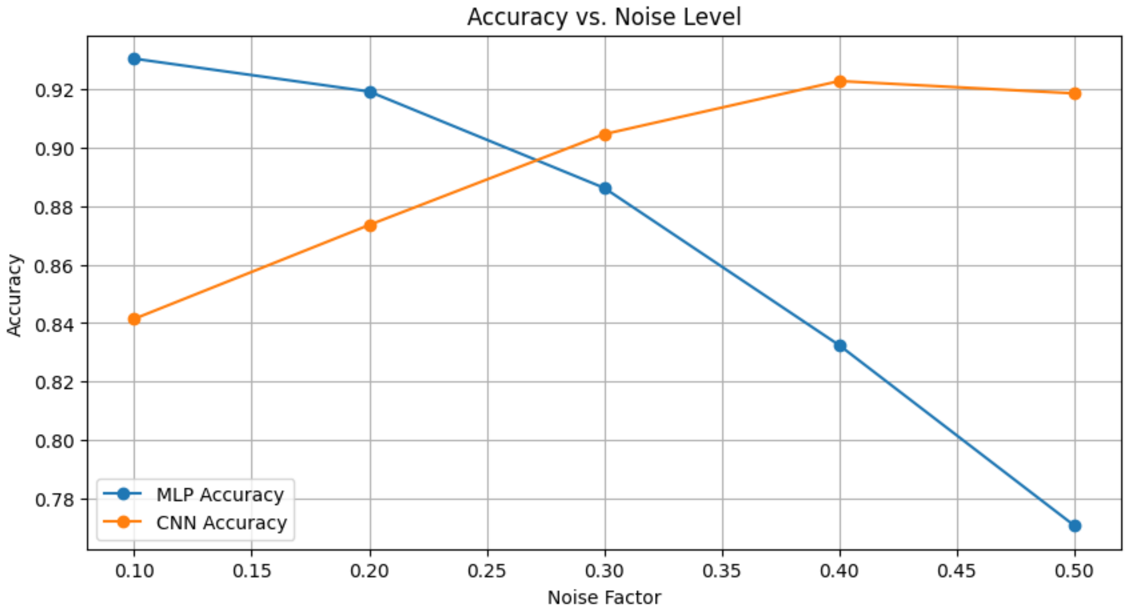
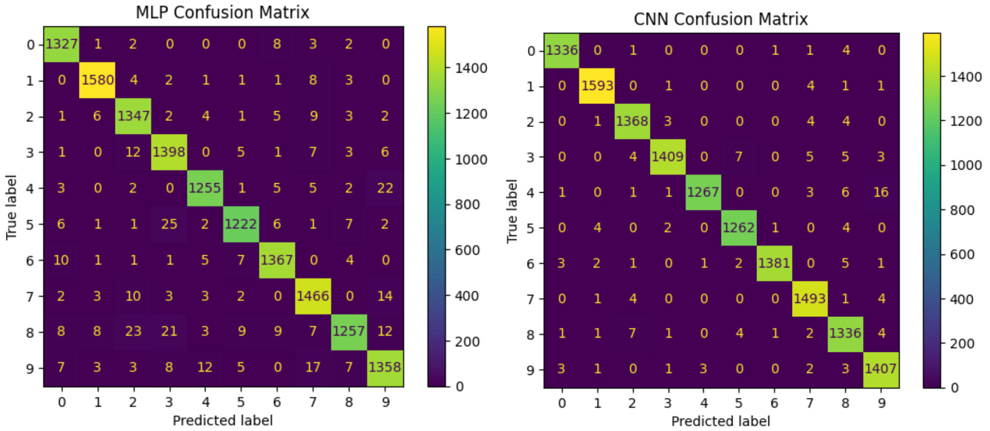


Figure 2. Comparisons of accuracy with different noise levels

The graph and accompanying accuracy values illustrate the performance of two neural network models, a Multi-Layer Perceptron (MLP) and a Convolutional Neural Network (CNN), on the MNIST dataset when subjected to different levels of noise. The noise factor, which ranges from 0.1 to 0.5, indicates the intensity of random noise added to the test images.

At a low noise level (noise factor = 0.1), the MLP model achieves a higher accuracy (92.96%) compared to the CNN (84.06%). This indicates that the MLP is initially more resilient to slight noise in the test images. As the noise factor increases to 0.2, the MLP still outperforms the CNN, though the gap begins to narrow (MLP: 92.11%, CNN: 87.41%).At a noise factor of 0.3, the performance trends begin to shift. The CNN accuracy surpasses the MLP, achieving 90.45% compared to the MLP's 88.59%. This suggests that the CNN is more effective at handling moderate levels of noise. At a noise factor of 0.4, the CNN continues to perform better, with an accuracy of 92.28% compared to the MLP's 83.51%. The CNN's performance not only remains robust but actually improves as noise increases, indicating that it is better at learning invariant features in the presence of noise.



MLP Confusion Matrix

	0	1	2	3	4	5	6	7	8	9
0	1327	1	2	0	0	0	8	3	2	0
1	0	1580	4	2	1	1	1	8	3	0
2	1	6	1347	2	4	1	5	9	3	2
3	1	0	12	1398	0	5	1	7	3	6
4	3	0	2	0	1255	1	5	5	2	22
5	6	1	1	25	2	1222	6	1	7	2
6	10	1	1	1	5	7	1367	0	4	0
7	2	3	10	3	3	2	0	1466	0	14
8	8	8	23	21	3	9	9	7	1257	12
9	7	3	3	8	12	5	0	17	7	1358

CNN Confusion Matrix

	0	1	2	3	4	5	6	7	8	9
0	1336	0	1	0	0	0	1	1	4	0
1	0	1593	0	1	0	0	0	4	1	1
2	0	1	1368	3	0	0	0	4	4	0
3	0	0	4	1409	0	7	0	5	5	3
4	1	0	1	1	1267	0	0	3	6	16
5	0	4	0	2	0	1262	1	0	4	0
6	3	2	1	0	1	2	1381	0	5	1
7	0	1	4	0	0	0	0	1493	1	4
8	1	1	7	1	0	4	1	2	1336	4
9	3	1	0	1	3	0	0	2	3	1407

Figure3: Confusion matrix of CNN and MLP

The CNN model outperforms the MLP in terms of accuracy and robustness, particularly in reducing misclassifications between visually similar digits, such as "3" and "8" or "5" and "3". While the MLP model shows more off-diagonal errors in the confusion matrix, the CNN significantly mitigates these errors, demonstrating its superior ability to capture and utilize spatial features. This results in fewer misclassifications and higher overall accuracy, highlighting the CNN's effectiveness in image recognition tasks.

5. Conclusion:

As evident from above both models performed exceptionally well in the evaluation of MNIST dataset. MLP achieved a high training accuracy of 99.57% which is high but still slightly below the accuracy of 99.87% achieved by the CNN model. This showcases the capability of CNN to process learning image data. The performance of the models with hyperparamers is another testament to CNNs capability of extracting spatial features more effectively then its counterpart in this study. The MLP managed to get a best accuracy of 98.27% with a hidden layer size of 256 neurons and a learning rate of 0.001. On the other hand, CNN required only learning rate tuning, with the optimal rate being 0.001, and managed to achieve an accuracy of 99.24%. During the testing phase of the models, as evident, MLP achieved a strong accuracy of 96.96% but CNN still outclassed it with an accuracy of 98.79%. The classification reports for both models further emphasize CNN's superiority, with higher precision, recall, and F1-scores across nearly all digit classes. A very important aspect of this study is the addition of noisy data with augmentation, and MLP's accuracy dropped from 97.71% on the original test set to 96.74% on the noisy test set. Suggesting that it has some sensitivity to noise. However, CNN, on the other hand, demonstrated greater resilience, with only a slight decrease in accuracy from 98.86% on the original test set to 98.68% on the noisy test set. CNN is specifically designed for image data which allows it achieve better generalization, and greater robustness to noise. The success of the CNN in handling noisy data and its ability to maintain high performance after hyperparameter tuning is what makes it more preferable for tasks involving image recognition. The slight drop in performance in both models underlines the challenges that noise introduces in image classification tasks and also highlights the CNN's superior capacity to manage these challenges mimicking real world scenarios.

6. References:

1. Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986). https://doi.org/10.1038/323533a0

2. Kurt Hornik, Maxwell Stinchcombe, Halbert White,Multilayer feedforward networks are universal approximators,Neural Networks,Volume 2, Issue 5,1989,Pages 359-366,ISSN 0893-6080,
https://doi.org/10.1016/0893-6080(89)90020-8.

3. Ian Goodfellow and Yoshua Bengio and Aaron Courville,deep learning,
note={\url{http://www.deeplearningbook.org}},
year={2016}

4. LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. In The handbook of brain theory and neural networks (Vol. 3361, pp. 1995)

5. Tong Yu, Hong Zhu, Hyper-Parameter Optimization: A Review of Algorithms and Applications,
arXiv:2003.05689

6. Yu, T. and Zhu, H., 2020. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*.

7. Glossary:

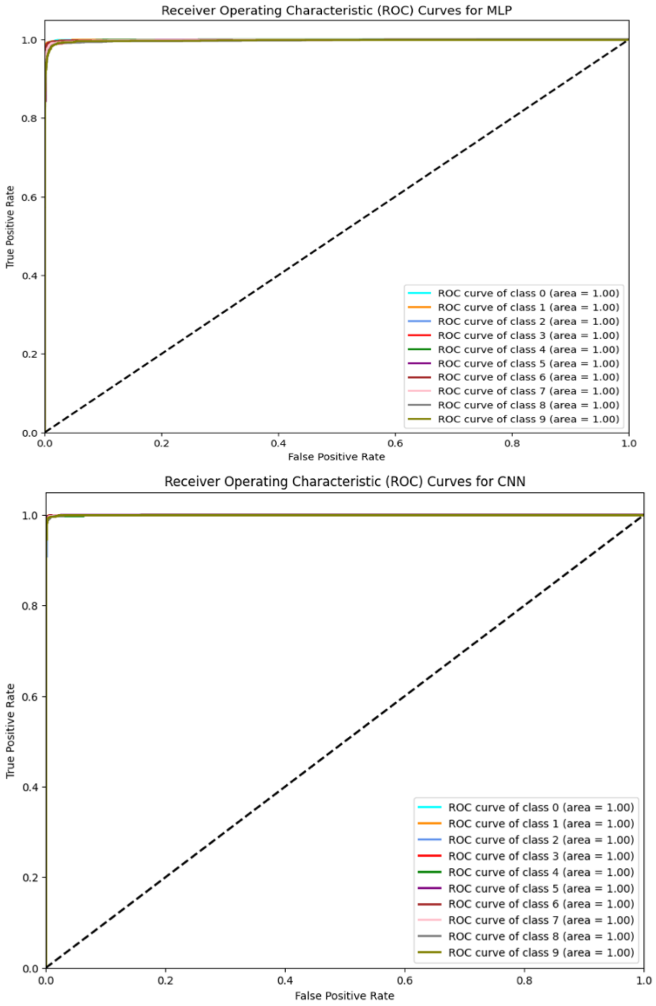


Figure 4: ROC curves for MLP and CNN

While the ROC curves and AUC scores for both MLP and CNN are perfect or near-perfect, indicating exceptional performance in class discrimination, the real difference between these models might lie in other metrics (e.g., precision, recall, and specific error rates) and their performance in noisier or more challenging datasets. The ROC curves show that both models are very effective in handling the MNIST dataset, but this metric alone does not fully capture the models' robustness or ability to handle edge cases.

Reason for the reduction of accuracy after hyperparameter tuning:

data augmentation(noise addition) was used during tuning, the model might have adapted to these augmented examples but performed worse on the original, unaltered test data. The tuned model might be better at handling noisy data but less effective on clean data, leading to a drop in test accuracy.