Assessment of Neural Networks and Deep Learning Models for MNIST Image Classification

Group 2: David Solero Chicano, Fadi Alkhori

30/12/2024

1 Problem Formulation

The objective of this assignment is to classify images of handwritten digits from the MNIST dataset using neural networks and deep learning algorithms. The dataset consists of 60,000 training examples and 10,000 test samples, each represented as a 28x28 grayscale image. The classification problem involves assigning each image to one of ten classes (digits 0-9).

2 Feature Transformation

To preprocess the data, the following steps were performed:

- **Normalization:** Pixel values were normalized to the range $[0, 1]$ by dividing each pixel by 255.
- **Reshaping:** Images were reshaped to match the input dimensions required by the neural networks. For the CNN model, images were reshaped to $28 \times 28 \times 1$, while for the fully connected network, they were flattened to one-dimensional vectors of size 784.
- **Feature Extraction:** Edge features were extracted using the Sobel operator for the fully connected neural network model.

3 Model Architectures

Two models were implemented to solve the classification task:

3.1 Convolutional Neural Network (CNN)

The CNN architecture consisted of the following layers:

- Two convolutional layers with 32 and 64 filters, respectively, followed by max-pooling layers.
- A flattening layer to convert feature maps into a one-dimensional vector.
- A dense layer with 128 neurons and ReLU activation, with L2 regularization and a dropout rate of 0.5.
- An output layer with 10 neurons and softmax activation.

Hyperparameters: The model was trained with the Adam optimizer, categorical crossentropy loss, a learning rate of 0.001, and a batch size of 32 for 20 epochs.

3.2 Fully Connected Neural Network (FNN)

The FNN architecture included:

- An input layer with 256 neurons and L2 regularization.
- Two hidden layers with 128 and 64 neurons, each followed by a dropout layer (rate 0.3).
- An output layer with 10 neurons and softmax activation.

Hyperparameters: Similar to the CNN, this model was trained with the Adam optimizer, categorical crossentropy loss, and a batch size of 32 for 50 epochs.

4 Evaluation Metrics

The models were evaluated using the following metrics:

- **Accuracy:** The proportion of correctly classified samples.
- **F1 Score:** The harmonic mean of precision and recall, calculated with weighted averaging to account for class imbalances.
- **Confusion Matrix:** A visualization of true versus predicted labels.

5 Results

5.1 Performance Metrics

The table below summarizes the performance of the two models:

Model	Train Accuracy	Test Accuracy	F1 Score
CNN	99.8%	99.1%	0.991
FNN	98.5%	97.2%	0.972

Table 1: Performance metrics of CNN and FNN models.

5.2 Visualization of Results

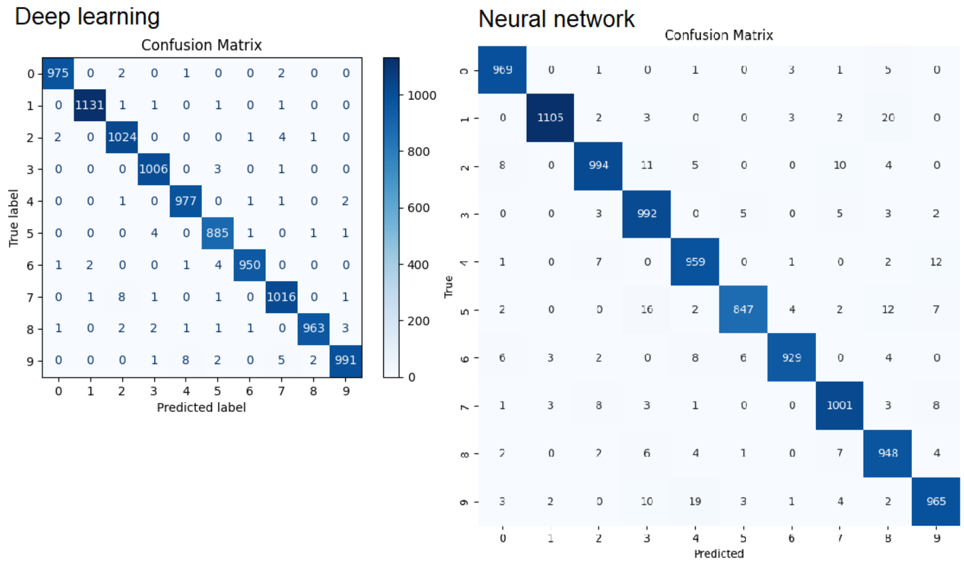

Figure 1: Confusion Matrix for CNN Model.

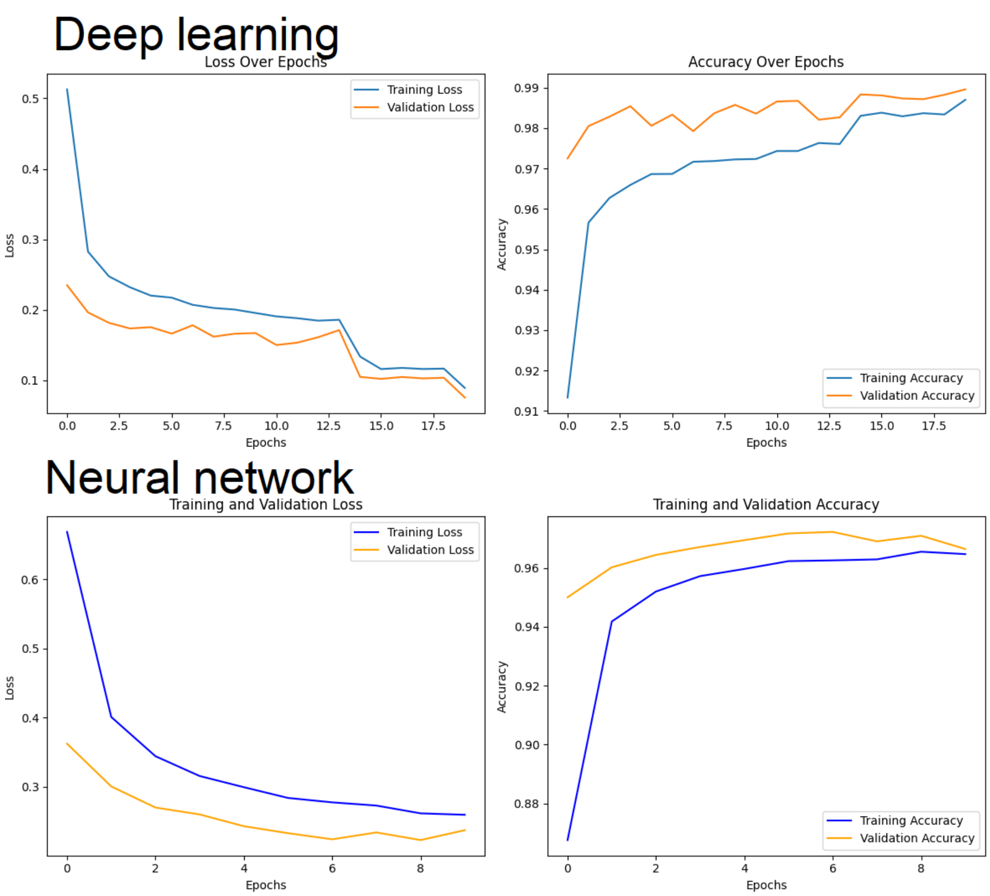

Figure 2: Loss and Accuracy

6 Discussion

The CNN model outperformed the fully connected neural network in terms of both accuracy and F1 score, demonstrating the importance of leveraging spatial information in image classification tasks. However, the performance of both models could be affected by:

- **Hyperparameters:** The choice of regularization and dropout rates played a significant role in preventing overfitting.
- **Data Augmentation:** The inclusion of augmented training data could improve generalization.
- **Computational Resources:** CNNs require more computational power due to their complex architecture.

Future improvements could include exploring advanced architectures (e.g., ResNet) and optimizing hyperparameters through grid search or random search.

7 Conclusion

This assignment demonstrated the application of neural networks and deep learning models to the MNIST classification problem. The CNN model achieved superior performance, validating the effectiveness of convolutional layers for image classification tasks.