Mini Deep Learning Project Report:
MNIST Digit Classification

Introduction

The MNIST dataset is a well-known collection of handwritten digits, consisting of 70,000 images (60,000 for training and 10,000 for testing). Each image is a grayscale representation of a single digit (0-9) in a 28x28 pixel format. This dataset serves as a benchmark for evaluating machine learning algorithms, particularly in image classification tasks. The relevance of the MNIST dataset lies in its simplicity and the ability to test various machine learning techniques, making it an ideal starting point for beginners in deep learning.

Model Selection

For this project, a **Feedforward Neural Network** was selected due to its effectiveness in handling simple classification tasks like digit recognition. The Sequential model from TensorFlow's Keras API was chosen for its straightforward implementation and ease of use. This model allows stacking layers linearly and is suitable for problems where input and output are directly connected.

Model Architecture:

- **Input Layer:** Flatten layer to convert the 28x28 images into a 784-dimensional vector.
- **Hidden Layer:** A dense layer with 128 neurons using the ReLU activation function to introduce non-linearity.
- **Output Layer:** A dense layer with 10 neurons (one for each digit) using softmax activation to produce class probabilities.

Methodology

Data Preprocessing

1. **Loading the Dataset:** The MNIST dataset was loaded using TensorFlow’s built-in functions.

2. **Normalization:** The pixel values of the images were normalized from a range of [0, 255] to [0, 1] to improve convergence during training.

Model Training

The model was compiled with the Adam optimizer and sparse categorical cross-entropy loss function. It was trained over 5 epochs using the training dataset.

```
history = model.fit(x_train, y_train, epochs=5)
```

Model Evaluation

After training, the model's performance was evaluated on the test dataset using accuracy and loss metrics:

```
test_loss, test_acc = model.evaluate(x_test, y_test)
```

Results

The performance metrics obtained from evaluating the model are as follows:

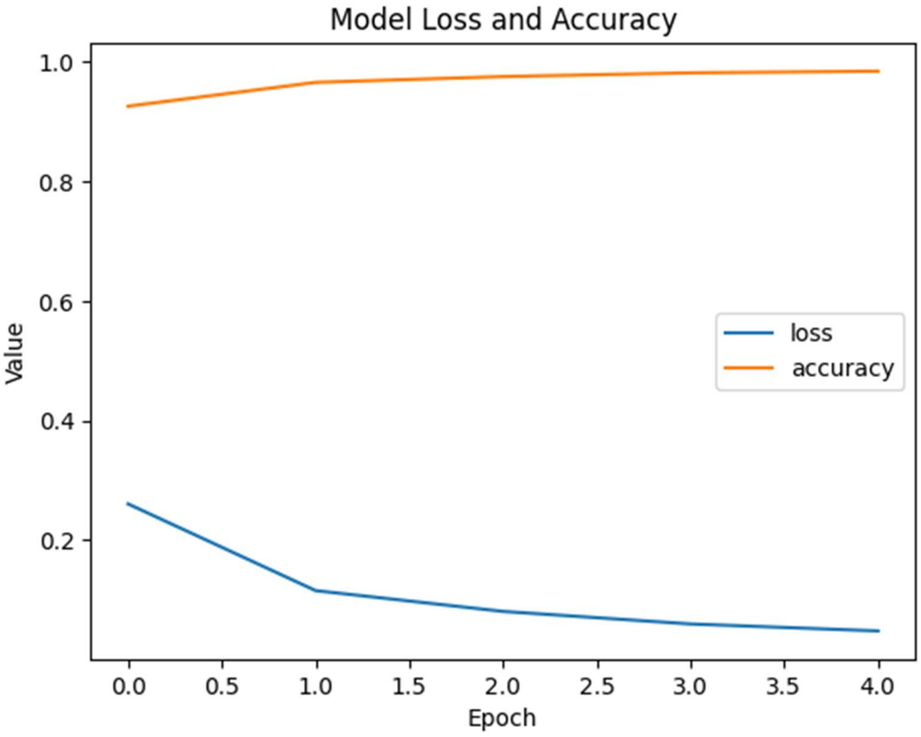
- **Test Accuracy:** 0.976 (or 97.6%)

- **Test Loss:** 0.0745

These results indicate that the model achieved a high level of accuracy in classifying handwritten digits, demonstrating its effectiveness in recognizing patterns within the data.

Loss & Accuracy Curves

The training process was visualized using Matplotlib to plot loss and accuracy curves over epochs. The attached PDF file contains these visualizations.



Conclusion

In conclusion, this mini deep learning project successfully implemented a feedforward neural network to classify handwritten digits from the MNIST dataset. The model achieved an impressive accuracy of 97.6% on unseen test data, indicating its robustness in recognizing patterns in images.

Potential Improvements:

1. **Increase Epochs:** Training for more epochs could further enhance model performance.

2. **Hyperparameter Tuning:** Experimenting with different architectures (e.g., adding more layers or neurons) or optimizers could yield better results.

3. **Regularization Techniques:** Implementing techniques like dropout could help prevent overfitting.

This project serves as a foundation for exploring more complex deep learning models and datasets.