

Introduction

This project trains three different models to classify handwritten digits from the MNIST dataset:

1. **Single-Layer Perceptron (SLP)**
2. **Multi-Layer Perceptron (MLP)**
3. **Convolutional Neural Network (CNN)**

Each model is trained on the MNIST dataset, and predictions are made for a Kaggle test dataset.

Dataset and Preprocessing

The MNIST dataset contains grayscale images (28x28 pixels) of handwritten digits (0-9). It is split into:

- **Training set:** 60,000 images
- **Test set:** 10,000 images

Steps for preprocessing:

- **Normalization:** Pixel values are scaled between 0 and 1 by dividing by 255.
- **Reshaping:**
 - SLP and MLP: Flattened into 784-dimensional vectors.
 - CNN: Retained as 28x28 images with a single channel.

Model Implementations

1. Single-Layer Perceptron (SLP)

This simple model consists of one dense layer with 10 neurons and a sigmoid activation function. It is trained for 5 epochs.

```
slpmodel = keras.Sequential([  
  
    keras.layers.Dense(10, input_shape=(784,), activation='sigmoid')  
  
])
```

2. Multi-Layer Perceptron (MLP)

This model has:

- Hidden layers: 128 and 64 neurons (ReLU activation).
- Output layer: 10 neurons (softmax activation).

```
mlpmodel = keras.Sequential([  
  
    keras.layers.Dense(128, input_shape=(784,), activation='relu'),  
  
    keras.layers.Dense(64, activation='relu'),  
  
    keras.layers.Dense(10, activation='softmax')  
  
])
```

The model is trained for 5 epochs.

3. Convolutional Neural Network (CNN)

The CNN includes:

- Two convolutional layers (ReLU activation, max pooling).
- A fully connected dense layer with 128 neurons and dropout.
- A softmax output layer.

```
cnn_model = keras.Sequential([  
  
    keras.layers.Conv2D(32, (3,3), activation='relu', padding='same', input_shape=(28, 28, 1)),  
  
    keras.layers.MaxPooling2D((2,2)),  
  
    keras.layers.Conv2D(64, (3,3), activation='relu', padding='same'),  
  
    keras.layers.MaxPooling2D((2,2)),  
  
    keras.layers.Flatten(),  
  
    keras.layers.Dense(128, activation='relu'),  
  
    keras.layers.Dropout(0.5),  
  
    keras.layers.Dense(10, activation='softmax')  
  
])
```

The CNN model is trained for 10 epochs with a batch size of 64.

Making Predictions

Each model generates predictions for the Kaggle test dataset. The results are saved in CSV format using a helper function:

```
def save_predictions(predictions, filename):  
  
    submission = pd.DataFrame({"ImageId": np.arange(1, len(predictions) + 1), "Label":  
predictions})  
  
    submission.to_csv(filename, index=False)  
  
    print(f"Saved {filename}")
```

Conclusion

- The **SLP model** is simple but less effective.
- The **MLP model** improves accuracy with hidden layers.
- The **CNN model** performs best by capturing spatial patterns in images.

The trained models generate predictions for the Kaggle competition, with the CNN model expected to achieve the best results.