Project Report: Handwriting Recognition using Deep Learning

Overview:

This project focuses on developing a deep learning model for handwriting recognition, specifically targeting the MNIST dataset, which contains images of handwritten digits. The model leverages convolutional neural networks (CNNs) to classify and predict the digits from the input images. The project is implemented using TensorFlow and Keras, popular frameworks for building and training deep learning models.

Dataset:

The dataset used in this project is the MNIST dataset, a well-known benchmark in the field of machine learning and image processing. It consists of 60,000 training images and 10,000 test images of handwritten digits, each of size 28x28 pixels. The dataset is loaded using Keras' built-in functionality:

```
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
```

Data Pre-processing:

Before feeding the data into the neural network, the images are normalized to have pixel values between 0 and 1. This is achieved by dividing the pixel values by 255:

```
x_train = x_train/255
x_test = x_test/255
```

Model Architecture:

The deep learning model is constructed using a sequential model in Keras, which includes the following layers:

1. Convolutional Layers: To extract features from the input images.

2. Max-Pooling Layers: To reduce the dimensionality and computational load.

3. Dense Layers: Fully connected layers to perform classification.

4. Dropout Layers: To prevent over fitting by randomly setting a fraction of input units to 0.

Training the Model

The model is compiled with the Adam optimizer and categorical cross-entropy loss function, which are suitable for multi-class classification problems. The model is then trained on the training dataset, with a portion of it reserved for validation to monitor the model's performance on unseen data:

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

model.compile(
    optimizer = 'adam',
    loss = 'sparse_categorical_crossentropy',
    metrics = ['accuracy']
)

model.fit(x_train_flattened, y_train, epochs = 5)
```

Evaluation:

The trained model is evaluated on the test dataset to determine its accuracy and generalization capabilities. The confusion matrix is used to visualize the performance of the model across different classes:

```
model.evaluate(x_test_flattened, y_test)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.9146 - loss: 0.3021
[0.267755925655365, 0.925000011920929]
```

Results:

The model achieves a high accuracy on the MNIST dataset, demonstrating its effectiveness in recognizing handwritten digits. The confusion matrix provides insights into specific classes where the model performs well and where there might be room for improvement:

```
import seaborn as sns
plt.figure(figsize = (10,7))
sns.heatmap(cm, annot=True, fmt = 'd')
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Conclusion:

This project successfully implements a deep learning model for handwriting recognition using CNNs. The model demonstrates high accuracy and robust performance on the MNIST dataset. Future work can involve experimenting with different architectures, hyperparameters, and advanced techniques like data augmentation to further improve the model's accuracy and generalization capabilities.