|  |  |
| --- | --- |
| **Ex No: 5**  **Date: 11h September 2024** | **Handwritten Digit Recognition using CNN** |

**Objective:** The objective is to build and evaluate a convolution neural network model using TensorFlow and Keras for multi-class classification(10 classes) of handwritten digits (0-9) using MNIST dataset. It focuses on training, validation, and testing while analysing the model's performance by plotting loss over epochs.

**Description:**

CNNs are a class of deep learning models specifically designed for working with grid-like data, such as images. They are excellent at detecting patterns in visual data and have features like Convolutional Operation, Pooling, Multiple Layers, Weight Sharing etc. CNNs are widely used in image recognition, object detection, facial recognition, medical imaging (such as X-ray and MRI analysis), and various other tasks involving visual data. CNN architecture is highly effective at extracting relevant features from images and is widely used in modern computer vision tasks.

**Model Summary:**

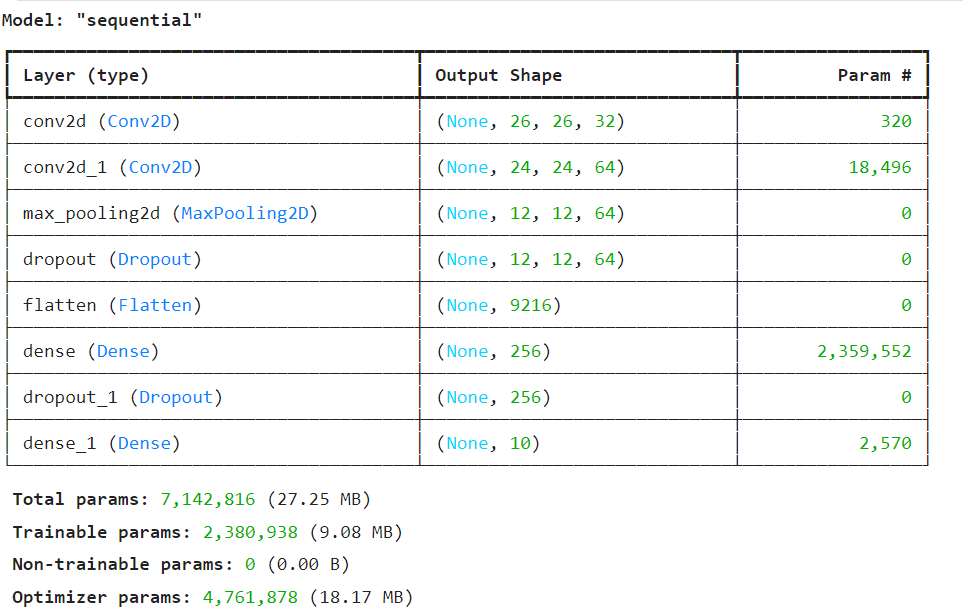


Figure1: A sequential CNN model for Handwritten digit recognition.

**Building the parts of the algorithm**

Here are the steps involved in building each part of the algorithm:

1. **Data Preprocessing**: The dataset is loaded and preprocessed into training and testing sets. The class vectors of the input data is converted to binary class matrices.
2. **Model Architecture**:
   * The neural network model is built using Keras. It contains multiple layers, including input, hidden (dense and dropout), and output layers.
   * ReLU activation functions are used in hidden layers, while softmax is used in the output layer for multi-class classification.
3. **Compilation**:
   * The model is compiled using a loss function (categorical cross-entropy) and an optimizer (Adadelta), with metrics set to monitor accuracy.
4. **Training**:
   * The model is trained over multiple epochs, with validation accuracy and loss tracked during each epoch.
5. **Evaluation**:
   * The final performance is evaluated on a test set which gives the accuracy and loss.
   * Two plots are generated to visualize training and validation loss over time.
   * The model is tested with some test images to understand the prediction and its probability.

**Conclusion:**

The neural network was successfully built and trained on the dataset. The plots of loss indicate how the model performs over the epochs, helping diagnose potential issues like overfitting. The final performance metrics on the test set provide an overall understanding of the model's classification ability. The number of epochs a model is trained plays a key role in predicting the accuracy of the model.

**GitHub Link:** [**https://github.com/tulasigr/DeepLearning**](https://github.com/tulasigr/DeepLearning)