

Write a program to demonstrate the working of CNN architecture to classify images

1. Library Imports

The script begins by importing TensorFlow, Keras, NumPy, and Matplotlib. TensorFlow/Keras is used for building and training the model, while NumPy and Matplotlib handle numerical operations and image visualization.

2. Dataset Loading

The MNIST dataset is loaded, which contains 60,000 training and 10,000 testing grayscale images of handwritten digits (0–9), each of size 28x28 pixels.

3. Data Preprocessing

The image data is reshaped to include a channel dimension suitable for CNN input (height, width, channels). The pixel values are then normalized to the range [0, 1] by dividing by 255. Labels are converted to one-hot encoded vectors to match the output format of the model.

4. Model Architecture

A sequential model is defined with:

- a. A convolutional layer with 32 filters and ReLU activation.
- b. A max pooling layer to reduce spatial dimensions.
- c. A flatten layer to convert 2D data into a 1D vector.
- d. A dense (fully connected) layer with 64 units and ReLU activation.
- e. An output layer with 10 units (for digits 0–9) and softmax activation for classification.

5. Model Compilation

The model is compiled using categorical crossentropy loss, Adam optimizer, and accuracy as the performance metric.

6. Training the Model

The model is trained for 10 epochs with a batch size of 128, using 10% of the training data for validation.

7. **Evaluation**

The model's performance is evaluated on the test set, and the test accuracy is printed.

8. **Prediction and Visualization**

The model predicts the label of the first test image. The predicted label and the true label are displayed on the image using Matplotlib.