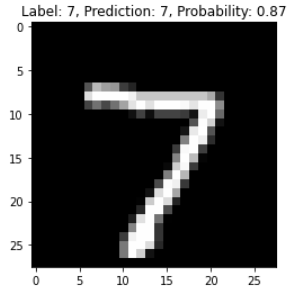
| **Ex No: 4**  **Date: 28-08-24** | **Handwritten Digit recognition using CNN** |
| --- | --- |

**Objective:**

To implement a convolutional neural network (CNN) for classifying handwritten digits from the MNIST dataset and to evaluate its performance.

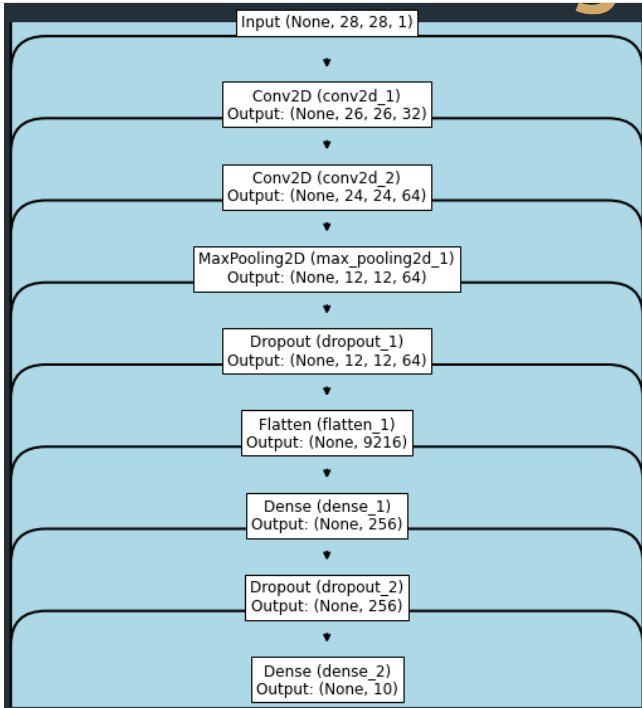
**Descriptions:**

The goal of this experiment is to build a CNN model to classify images of handwritten digits (0-9) from the MNIST dataset. The MNIST dataset consists of 28x28 grayscale images of handwritten digits, and the task is to correctly classify each image into one of the 10 classes (digits 0-9).



A CNN is chosen due to its effectiveness in handling image data, particularly through the use of convolutional layers that automatically learn spatial hierarchies of features from input images. This model architecture is well-suited for image classification tasks as it can capture patterns such as edges, textures, and shapes at different levels of abstraction.

**Model:**

****

### Model Architecture:

1. **Input Layer:**
   * Image size: 28x28 pixels
   * Channels: 1 (grayscale)
2. **Convolutional Layer 1:**
   * Filters: 32
   * Kernel size: 3x3
   * Activation: ReLU
3. **Convolutional Layer 2:**
   * Filters: 64
   * Kernel size: 3x3
   * Activation: ReLU
4. **Pooling Layer:**
   * Pool size: 2x2
5. **Dropout Layer:**
   * Dropout rate: 0.25
6. **Flatten Layer:**
   * Converts 2D feature maps into 1D feature vector
7. **Fully Connected Layer 1:**
   * Units: 128
   * Activation: ReLU
8. **Dropout Layer:**
   * Dropout rate: 0.5
9. **Output Layer:**
   * Units: 10 (corresponding to the 10 classes)
   * Activation: Softmax

### Model Compilation:

* **Loss Function:** Categorical Crossentropy
* **Optimizer:** Adadelta
* **Metrics:** Accuracy

Mathematically:

* Convolution Operation:  
  Z[l]=W[l]∗X[l−1]+b[l]Z^{[l]} = W^{[l]} \ast X^{[l-1]} + b^{[l]}Z[l]=W[l]∗X[l−1]+b[l]  
  where ∗\ast∗ denotes convolution, and lll is the layer index.
* Activation Function:  
  A[l]=ReLU(Z[l])A^{[l]} = ReLU(Z^{[l]})A[l]=ReLU(Z[l])
* Pooling Operation:  
  Max pooling over 2x2 regions.
* Final Prediction:  
  y^=softmax(Z[L])\hat{y} = \text{softmax}(Z^{[L]})y^​=softmax(Z[L])
* Cost Function:  
  J=−1m∑i=1my(i)log⁡(y^(i))J = -\frac{1}{m} \sum\_{i=1}^{m} y^{(i)} \log(\hat{y}^{(i)})J=−m1​∑i=1m​y(i)log(y^​(i))  
  where mmm is the number of examples.

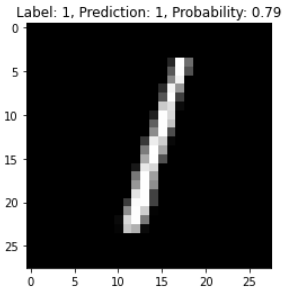
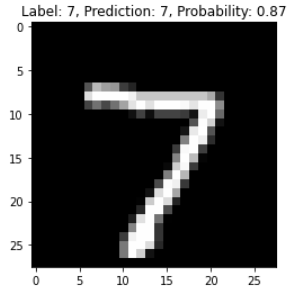
**Building the parts of algorithm**

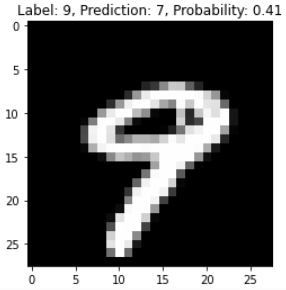
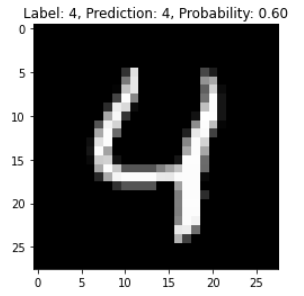
### Implementation Steps:

1. Loading the MNIST Dataset.
2. Data Preprocessing.
3. Building the CNN Model.
4. Model Training (10 epochs).
5. Evaluating the Model on the Test Set.
6. Tuning Hyperparameters for Improved Performance.
7. Visualizing Model Predictions.

### Results and Analysis:

* **Test loss:** 0.6120116114616394
* **Test accuracy:** 0.8532000184059143





**GitHubLink:** [**https://github.com/chandanab1/Deep\_Learning**](https://github.com/chandanab1/Deep_Learning)