LeNet-5 Implementation for MNIST Digit Classification

# 1. Introduction

An implementation of the classic LeNet-5 Convolutional Neural Network (CNN) architecture using TensorFlow and Keras. Originally developed by Yann LeCun et al., LeNet-5 was one of the first CNN architectures and laid the groundwork for modern deep learning approaches in computer vision. It was specifically designed to recognize handwritten digits and is ideal for benchmarking with datasets like MNIST.

# 2. Objective

The primary goals of this project are to:  
- Implement the LeNet-5 architecture using Tensorflow and keras deep learning libraries.  
- Train the network on the MNIST dataset, a standardized dataset of handwritten digits (0–9).  
- Evaluate the trained model’s performance on test data and analyze its generalization capability.

# 3. Dataset Description

The MNIST dataset consists of:  
- 60,000 training images  
- 10,000 test images  
  
Each image is:  
- Grayscale (single channel)  
- Sized 28×28 pixels  
- Represents one digit between 0 and 9

# Step-by-step Code Explanation

## 4.1 Importing Required Libraries

import tensorflow as tf  
from tensorflow.keras import layers, models  
from tensorflow.keras.datasets import mnist  
from tensorflow.keras.utils import to\_categorical

## 4.2 Loading and Preprocessing the Data

(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()  
  
# Normalize pixel values  
x\_train = x\_train.astype('float32') / 255.0  
x\_test = x\_test.astype('float32') / 255.0  
  
# Reshape data to fit CNN input (28x28x1)  
x\_train = x\_train.reshape(-1, 28, 28, 1)  
x\_test = x\_test.reshape(-1, 28, 28, 1)  
  
# One-hot encoding labels  
y\_train = to\_categorical(y\_train, 10)  
y\_test = to\_categorical(y\_test, 10)

## 4.3 Defining the LeNet-5 Architecture

model = tf.keras.Sequential([

tf.keras.Input(shape=(28, 28, 1)),

tf.keras.layers.Conv2D(16, kernel\_size=5, activation='tanh', padding='same'),

tf.keras.layers.AveragePooling2D(pool\_size=(2, 2)),

tf.keras.layers.Conv2D(32, kernel\_size=5, activation='tanh'),

tf.keras.layers.AveragePooling2D(pool\_size=(2, 2)),

tf.keras.layers.Conv2D(64, kernel\_size=5, activation='tanh'),

tf.keras.layers.Flatten(),

tf.keras.layers.Dense(128, activation='tanh'),

tf.keras.layers.Dense(10, activation='softmax')

])

## 4.4 Compiling the Model

model.compile(

optimizer='adam',

loss='sparse\_categorical\_crossentropy',

metrics=['accuracy']

)

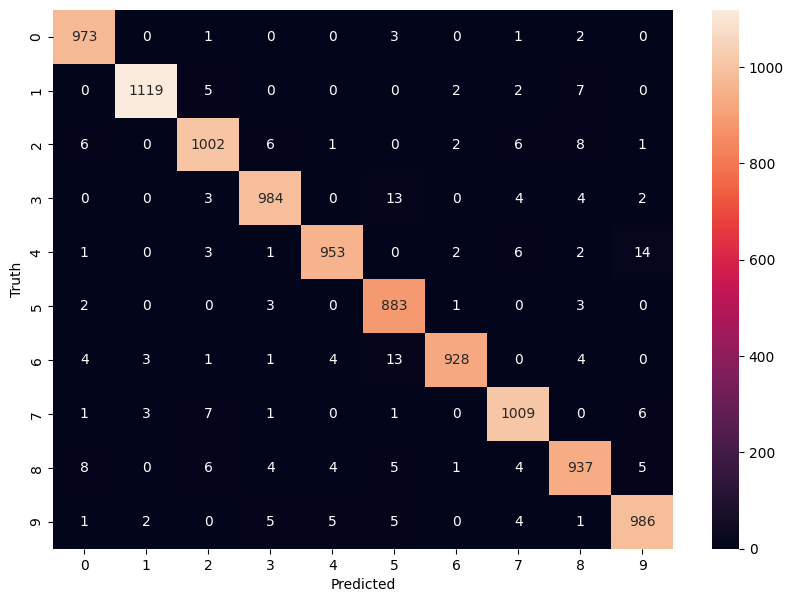
## 4.5 Training the Model

history = model.fit(  
 x\_train, y\_train,  
 epochs=10,  
 batch\_size=64,  
 validation\_split=0.1  
)

# 5. Model Evaluation

test\_loss, test\_acc = model.evaluate(x\_test, y\_test)  
print("Test accuracy:", test\_acc)

# 6. Confussion Matrix



# 7. Conclusion

We have successfully implements the LeNet-5 architecture using TensorFlow/Keras for handwritten digit classification. It demonstrates the effectiveness of this foundational CNN design and reinforces key concepts such as convolutional operations, pooling, and fully connected layers. The model performs well on the MNIST dataset, showcasing the continued relevance of LeNet-5 in educational and benchmarking contexts.