**WEEK-10**

**PYTHON LIBRARIES – TENSORFLOW AND KERAS**

**Description:**

This unit introduces two powerful Python libraries for machine learning and deep learning – TensorFlow and Keras. TensorFlow is an open-source platform developed by Google, which is used to perform high-performance numerical computations. It supports building and training neural networks using computational graphs.  
Keras is a high-level API built on top of TensorFlow that allows for fast prototyping of deep learning models. It abstracts much of the complexity in TensorFlow by providing an intuitive interface for creating, training, and evaluating deep learning models.

In this unit, students will learn:

* How to create and manipulate Tensors in TensorFlow.
* Building deep learning models using Keras.
* Defining and customizing layers like Dense, Flatten, and Dropout.
* Training models with various optimizers and loss functions.
* Performing data preprocessing for feeding real-world data into models.

This foundational knowledge is crucial for anyone entering the field of machine learning or AI using Python.

**1.** **Write a program for dataset loading, data preprocessing, defining a model, training, evaluating, saving, and loading the model using TensorFlow and keras libraries.**

**Description:**

In this experiment, we:

1. **Loaded** the MNIST handwritten digit dataset using tensorflow.keras.datasets.
2. **Preprocessed** the data by reshaping and normalizing the pixel values between 0 and 1.
3. **Converted** labels to one-hot encoded vectors using to\_categorical.
4. **Built** a Convolutional Neural Network (CNN) using Keras with layers like Conv2D, MaxPooling2D, Flatten, and Dense.
5. **Compiled and trained** the model using the Adam optimizer and categorical crossentropy loss.
6. **Evaluated** the model's accuracy on the test dataset.
7. **Saved** the trained model to an HDF5 file (.h5) using model.save().
8. **Loaded** the saved model using load\_model() and verified its accuracy again.

This experiment demonstrates the end-to-end process of a deep learning workflow using TensorFlow and Keras.

**Program:**

import tensorflow as tf

from tensorflow.keras import layers, models

from tensorflow.keras.datasets import mnist

from tensorflow.keras.utils import to\_categorical

# 1. Load Dataset

(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data()

# 2. Data Preprocessing

x\_train = x\_train.reshape((x\_train.shape[0], 28, 28, 1)).astype('float32') / 255

x\_test = x\_test.reshape((x\_test.shape[0], 28, 28, 1)).astype('float32') / 255

y\_train = to\_categorical(y\_train)

y\_test = to\_categorical(y\_test)

# 3. Model Definition

model = models.Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(28, 28, 1)))

model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Flatten())

model.add(layers.Dense(64, activation='relu'))

model.add(layers.Dense(10, activation='softmax'))

# 4. Compile Model

model.compile(optimizer='adam',

loss='categorical\_crossentropy',

metrics=['accuracy'])

# 5. Train the Model

model.fit(x\_train, y\_train, epochs=5, batch\_size=64, validation\_split=0.1)

# 6. Evaluate the Model

test\_loss, test\_acc = model.evaluate(x\_test, y\_test)

print(f"Test Accuracy: {test\_acc:.4f}")

# 7. Save the Model

model.save('mnist\_model.h5')

print("Model saved to mnist\_model.h5")

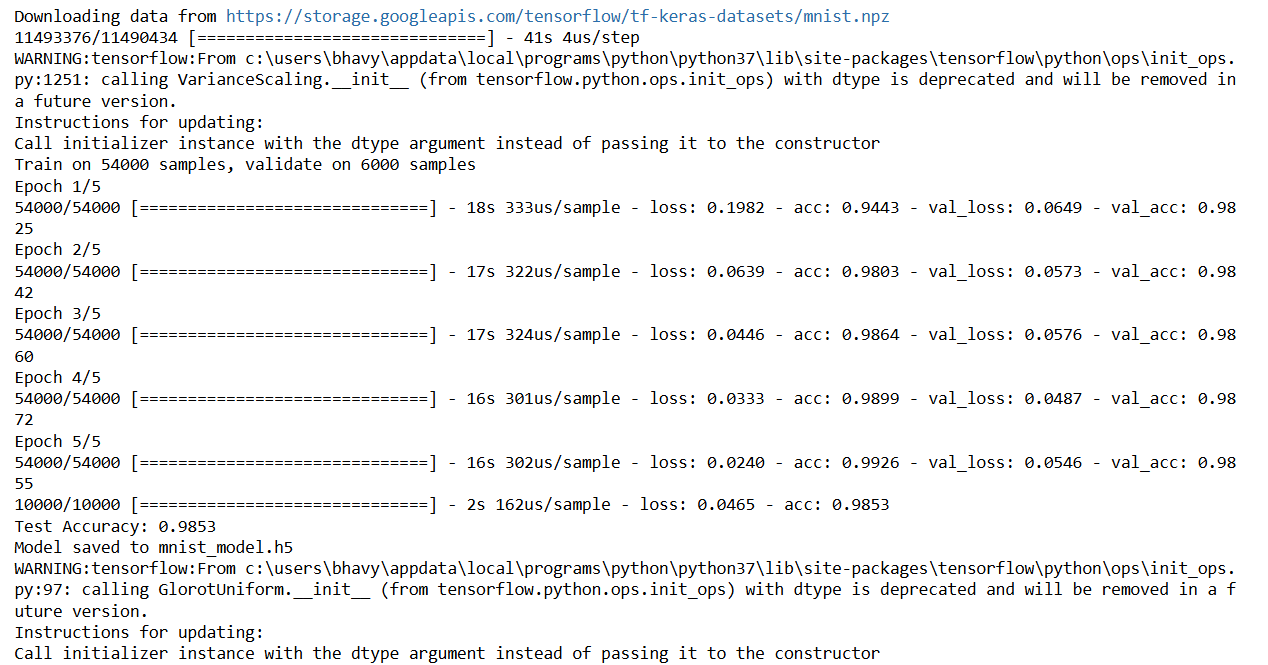
# 8. Load the Model

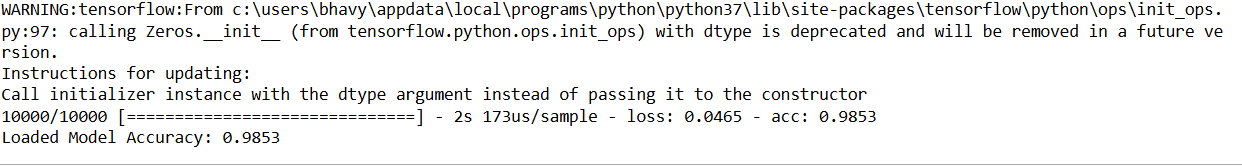
loaded\_model = models.load\_model('mnist\_model.h5')

loss, acc = loaded\_model.evaluate(x\_test, y\_test)

print(f"Loaded Model Accuracy: {acc:.4f}")

**Output :**

****

****

**Viva Questions – Unit 5 :TENSORFLOW AND KERAS**

1. **What is TensorFlow and why is it used?**

TensorFlow is an open-source machine learning library developed by Google. It is used for numerical computation and large-scale machine learning, particularly deep learning using computational graphs.

1. **What is a Tensor in TensorFlow?**

A Tensor is a multi-dimensional array (similar to NumPy arrays) used as the basic data structure in TensorFlow to represent inputs, outputs, and model parameters.

1. **What is the role of Keras in TensorFlow?**

Keras is a high-level API built on top of TensorFlow that simplifies the process of building and training neural networks by providing pre-built layers, models, and utilities.

1. **How do you preprocess data before training a model?**

Preprocessing includes reshaping, normalizing input data, and converting labels to categorical format using functions like reshape(), division by 255, and to\_categorical().

**5.What function is used to save and load a model in Keras?**

To save a model, we use model.save('filename.h5'). To load it back, we use keras.models.load\_model('filename.h5').