1. !pip install tensorflow
2. import tensorflow as tf

mnist = tf.keras.datasets.mnist

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

x\_train,x\_test = x\_train / 255.0, x\_test / 255.0

3.!pip install keras

4. model = tf.keras.models.Sequential([

tf.keras.layers.Flatten(input\_shape=(28,28)),

tf.keras.layers.Dense(512, activation='relu'),

tf.keras.layers.Dense(512, activation='relu'),

tf.keras.layers.Dropout(0.2),

tf.keras.layers.Dense(512, activation='relu'),

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

])

5. model.compile(optimizer='SGD',

loss ='sparse\_categorical\_crossentropy',

metrics=['accuracy'])

tf\_callbacks = tf.keras.callbacks.TensorBoard(log\_dir = "logs/fit" , histogram\_freq = 1)

history=model.fit(x\_train,y\_train,validation\_data=(x\_test,y\_test), epochs=10,callbacks =tf\_callbacks)

6. import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'])

plt.plot(history.history['val\_accuracy'])

plt.title('model accuracy')

plt.ylabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

plt.plot(history.history['loss'])

plt.plot(history.history['val\_loss'])

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()