#Import Packages

import tensorflow as tf

from tensorflow import keras

import matplotlib.pyplot as plt

import random

#load the data #len shape array feature scaling array

mnist=tf.keras.datasets.mnist

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

print(len(x\_train))

print(len(x\_test))

print(x\_train.shape)

print(x\_test.shape)

print(x\_train[0])

plt.matshow(x\_train[0])

#Feature Scaling

x\_train=x\_train/255

x\_test=x\_test/255

print(x\_train[0])

#defining network architecture using keras

model=tf.keras.Sequential([

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

keras.layers.Dense(128,activation='relu'),

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

])

model.summary()

#train the model using sgd

model.compile(optimizer='sgd', loss='sparse\_categorical\_crossentropy',metrics=['accuracy'])

history=model.fit(x\_train, y\_train, validation\_data= (x\_test,y\_test),epochs=6 , verbose=2)

#evaluate the network

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

print("Loss=",test\_loss)

print("Accuracy=",test\_acc)

n=random.randint(0,9999)

plt.imshow(x\_test[n])

#plot the training loss and accuracy

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

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

plt.title('Training Loss and Accuracy')

plt.xlabel("epochs")

plt.legend(["Accuracy","Training Loss"],loc= 'lowerleft')