|  |  |  |  |
| --- | --- | --- | --- |
| In | [ | ]: | import tensorflow as tf |
|  |  |  |  |
| In | [ | ]: | from tensorflow import keras |
|  |  |  |  |
| In | [ | ]: | import pandas as pd import numpy as np  import matplotlib.pyplot as plt |
|  |  |  | import random |
|  | | | %matplotlib inline |

|  |  |  |  |
| --- | --- | --- | --- |
| In | [ | ]: | mnist=tf.keras.datasets.mnist (x\_train,y\_train),(x\_test,y\_test)=mnist.load\_data() |
|  |  |  |  |
| In | [ | ]: | plt.matshow(x\_train[0]) |
|  |  |  |  |
| In | [ | ]: | x\_train=x\_train/255 |
|  |  |  | x\_test=x\_test/255 |
|  |  |  |  |
| In | [ | ]: | x\_train[0] |
|  |  |  |  |
| In | [ | ]: | model=keras.Sequential([ |
|  | | | keras.layers.Flatten(input\_shape=(28,28)),  keras.layers.Dense(128,activation='relu'), keras.layers.Dense(10,activation='softmax') |

In [ ]:

model.summary()

In [ ]:

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

In [ ]:

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

In [ ]:

test\_loss,test\_acc=model.evaluate(x\_test,y\_test) print(”Loss=%.3f" %test\_loss) print(”Accuracy=%.3f" %test\_acc)

In [ ]:

n=random.randint(0,9999) plt.imshow(x\_test[n]) plt.show

In [ ]:

test\_predict=model.predict(x\_test) test\_predict\_labels=np.argmax(test\_predict,axis=1)

confusion\_matrix=tf.math.confusion\_matrix(labels=y\_test,predictions=test\_predict\_labels) print('confusion matrix of the test set :\n', confusion\_matrix)

In [ ]: