import tensorflow from tensorflow.keras import Sequential from tensorflow.keras import Sequential from tensorflow.keras.layers import Flatten, Dense     9 : (X_train, y_train) , (X_test, y_test) = keras.datasets.mnist.load_data()     10 : print(X_train)
print(y_test.shape) print(y_test.shape)  [[[0 0 0 0 0 0]
[0 0 0 0 0 0] [0 0 0 0 0 0]
$ar{[} 0 \ 0 \ 0 \ \dots \ 0 \ 0 \ ar{]}$
[0 0 0 0 0 0]  [0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0]]
[[0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0]
[0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0]
[0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0]
[0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 0 0 0] [0 0 0 5 6 8]
[3 0 4 5 0 8] (10000, 28, 28) (10000,)  [11]: X_train[0]  t[11]: array([[ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0
0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
18, 18, 18, 126, 136, 175, 26, 166, 255, 247, 127, 0, 0, 0, 0, 0], 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 30, 36, 94, 154, 170, 253, 253, 253, 253, 253, 225, 172, 253, 242, 195, 64, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 49, 238, 253, 253, 253, 253,
253, 253, 253, 253, 251, 93, 82, 82, 56, 39, 0, 0, 0, 0, 0, 0, 0, 0, 0],  [ 0,  0,  0,  0,  0,  0,  0, 18, 219, 253, 253, 253, 253, 253, 253, 198, 182, 247, 241,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 14, 1, 154, 253,
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11, 190, 253, 70, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
81, 240, 253, 253, 119, 25, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 23, 66, 213, 253, 253, 253, 253, 253, 198, 81, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 18, 171, 219, 253, 253, 253, 253, 195, 80, 9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
<pre>[12]: plt.imshow(X_train[0]) t[12]: <matplotlib.image.axesimage 0x1aa00054be0="" at=""></matplotlib.image.axesimage></pre>
15 -
20 -
0 5 10 15 20 25 [13]: print(y_train[0]) 5
<pre>[14]: X_train= X_train/255 X_test= X_test/255  ain[0] [15]: model = Sequential()</pre>
<pre>[16]: model.add(Flatten(input_shape=(28,28)))     model.add(Dense(128,activation='relu'))     model.add(Dense(10,activation='softmax')) [17]: model.summary()</pre>
Model: "sequential"  Layer (type)
dense (Dense)       (None, 128)       100480         dense_1 (Dense)       (None, 10)       1290         ==================================
Non-trainable params: 0  [18]: model.compile(optimizer= 'Adam', loss = 'sparse_categorical_crossentropy', metrics=['accuracy'])  [19]: history = model.fit(X_train,y_train,batch_size = 64,epochs =10,verbose = 1,validation_split =0.2)
Epoch 1/10 750/750 [====================================
Epoch 4/10 750/750 [====================================
750/750 [====================================
[20]: model.evaluate(X_test,y_test)  313/313 [===================================
[21]: model.predict(X_test)  313/313 [===================================
[9.32085823e-06, 9.94868040e-01, 5.47335017e-04,, 1.15911942e-03, 2.93456414e-03, 2.80791392e-06],, [4.14591647e-13, 3.27928304e-13, 7.02479524e-14,, 9.10616507e-08, 5.82496362e-09, 8.39385393e-05], [3.33104516e-11, 4.66730536e-14, 4.69865954e-13,, 1.31676918e-12, 3.06161587e-06, 9.15509474e-11],
[1.65201874e-09, 6.36100439e-13, 4.10190859e-09,, 3.95666267e-14, 1.74308155e-11, 1.28479297e-10]], dtype=float32)  [29]: model.predict(X_train)  1875/1875 [====================================
5.1008013e-12, 4.4634110e-09], [9.9999809e-01, 4.8815309e-11, 1.8224857e-06,, 2.4087390e-10, 6.2658941e-11, 9.1750479e-08], [2.1431503e-09, 2.1458338e-05, 8.7065469e-05,, 3.3888471e-05, 1.4529827e-05, 9.1147813e-05],, [7.1187756e-10, 1.2044303e-09, 3.9608837e-12,, 3.3250850e-11,
2.9101079e-08, 9.5277596e-07], [1.6898439e-05, 1.6664924e-07, 7.1527014e-05,, 2.9500659e-06, 3.6904200e-07, 3.4044931e-06], [5.4692748e-05, 7.0036293e-08, 2.9540349e-06,, 4.8392812e-06, 9.9960941e-01, 3.2626174e-04]], dtype=float32)
<pre>[25]: plt.imshow(X_test[0]) t[25]: <matplotlib.image.axesimage 0x1aa2d951850="" at="">  0-</matplotlib.image.axesimage></pre>
5 - 10 -
10 -
10 -
10 - 15 - 20 - 25 - 25 - 25 - 25 - 25 - 25 - 2
10 - 20 - 25 - 25 - 25 - 25 - 25 - 25 - 2
10 -
10 - 20 - 25 - 20 - 25 - 20 - 25 - 25 - 2
10 - 25 - 25 - 25 - 25 - 25 - 25 - 25 - 2