digit-classification-multi-class

June 27, 2023

[2]:

```
import tensorflow
from tensorflow import keras
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense,Flatten
```

[3]:

```
(X_train, y_train),(X_test, y_test) = keras.datasets.mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-
datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

[9]:

```
X_train[0].shape
```

[9]:

```
(28, 28)
```

[10]:

```
X_test.shape
```

[10]:

```
(10000, 28, 28)
```

[11]:

```
y_train
```

[11]:

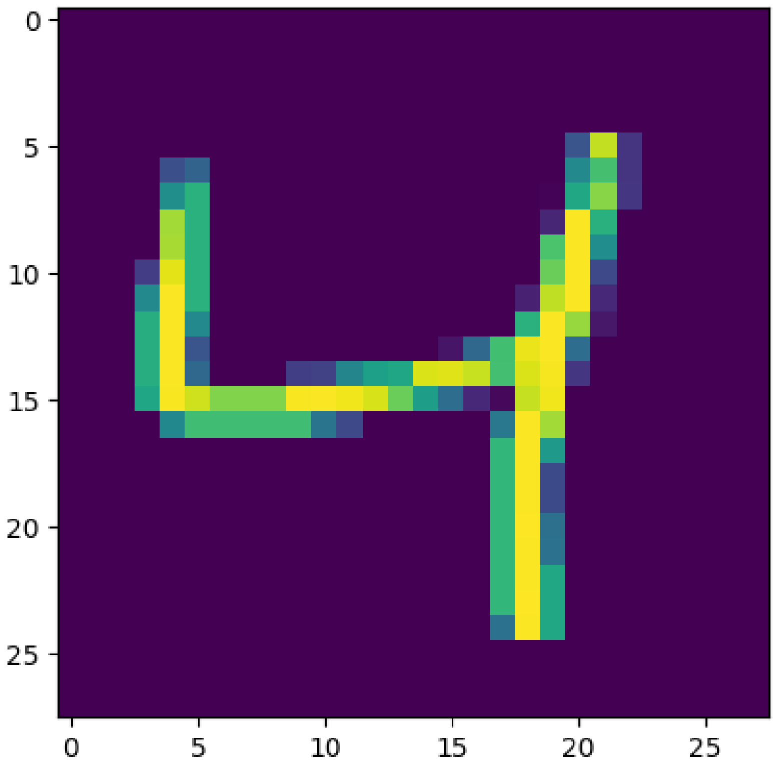
```
array([5, 0, 4, …, 5, 6, 8], dtype=uint8)
```

[14]:

```
import matplotlib.pyplot as plt
plt.imshow(X_train[2])
```

[14]:

```
<matplotlib.image.AxesImage at 0x7f258ad504f0>
```



0.1 Now divide each pixel with 255 to get accurate and fast results of weights

[15]:
```
X_train = X_train/255
X_test = X_test/255
```

[17]:
```
X_train[3]
```

[17]:
```
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.        ,
        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.        , 0.48627451, 0.99215686,
        1.        , 0.24705882, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.37647059, 0.95686275, 0.98431373,
        0.99215686, 0.24313725, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.49803922, 0.98431373, 0.98431373,
        0.99215686, 0.24313725, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.26666667, 0.9254902 , 0.98431373, 0.82745098,
        0.12156863, 0.03137255, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.23529412, 0.89411765, 0.98431373, 0.98431373, 0.36862745,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.60784314, 0.99215686, 0.99215686, 0.74117647, 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.07843137,
        0.99215686, 0.98431373, 0.92156863, 0.25882353, 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.1254902 , 0.80392157,
        0.99215686, 0.98431373, 0.49411765, 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.40784314, 0.98431373,
        0.99215686, 0.72156863, 0.05882353, 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.31372549, 0.94117647, 0.98431373,
        0.75686275, 0.09019608, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.1254902 , 0.99215686, 0.99215686, 0.99215686,
        0.62352941, 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.59215686, 0.98431373, 0.98431373, 0.98431373,
        0.15294118, 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.18823529, 0.86666667, 0.98431373, 0.98431373, 0.6745098 ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.91764706, 0.98431373, 0.98431373, 0.76862745, 0.04705882,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.99215686, 0.98431373, 0.98431373, 0.34901961, 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.62352941,
        1.        , 0.99215686, 0.99215686, 0.12156863, 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.18823529, 0.89411765,
        0.99215686, 0.96862745, 0.54901961, 0.03137255, 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.25098039, 0.98431373,
        0.99215686, 0.8627451 , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.25098039, 0.98431373,
        0.99215686, 0.8627451 , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.09411765, 0.75686275,
        0.99215686, 0.8627451 , 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.        ]])
```

- The given input is `2d` we have to flatten in to convert it into `1d`
- The `Flatten` method in tenseorflow is used to bring rows side by side.
- ’Flatten`converts higher dimensional data into`1d‘.

Eg.

Given input is 28*28

```
          - 28 rows and 28 columns
```

When we use `Flatten` it converts rows and columns into single value i.e `28*28 = 784`

[30]:
```
model = Sequential()

model.add(Flatten(input_shape = (28,28)))
model.add(Dense(128,activation = 'relu'))
model.add(Dense(32,activation = 'relu'))
model.add(Dense(10,activation = 'softmax'))
```

[31]:
```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_1 (Flatten)         (None, 784)               0

 dense_2 (Dense)             (None, 128)               100480

 dense_3 (Dense)             (None, 32)                4128

 dense_4 (Dense)             (None, 10)                330

=================================================================
Total params: 104,938
Trainable params: 104,938
```

```
Non-trainable params: 0
_________________________________________________________________
```

[40]:

```
model.compile(loss = 'sparse_categorical_crossentropy', optimizer = 'Adam')
```

[41]:

```
history = model.fit(X_train,y_train,epochs = 25,validation_split = 0.2)
```

```
Epoch 1/25
1500/1500 [==============================] - 10s 6ms/step - loss: 0.0096 -
val_loss: 0.1757
Epoch 2/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0073 -
val_loss: 0.1725
Epoch 3/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0086 -
val_loss: 0.1682
Epoch 4/25
1500/1500 [==============================] - 8s 6ms/step - loss: 0.0061 -
val_loss: 0.1892
Epoch 5/25
1500/1500 [==============================] - 7s 5ms/step - loss: 0.0117 -
val_loss: 0.1812
Epoch 6/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0073 -
val_loss: 0.1712
Epoch 7/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0066 -
val_loss: 0.1795
Epoch 8/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0077 -
val_loss: 0.1999
Epoch 9/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0059 -
val_loss: 0.2070
Epoch 10/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0072 -
val_loss: 0.1825
Epoch 11/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0075 -
val_loss: 0.1887
Epoch 12/25
1500/1500 [==============================] - 10s 6ms/step - loss: 0.0062 -
val_loss: 0.1809
Epoch 13/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0078 -
val_loss: 0.1745
Epoch 14/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0067 -
```

```
val_loss: 0.1920
Epoch 15/25
1500/1500 [==============================] - 11s 7ms/step - loss: 0.0075 -
val_loss: 0.1979
Epoch 16/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0062 -
val_loss: 0.1981
Epoch 17/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0070 -
val_loss: 0.1986
Epoch 18/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0050 -
val_loss: 0.2230
Epoch 19/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0058 -
val_loss: 0.2275
Epoch 20/25
1500/1500 [==============================] - 7s 5ms/step - loss: 0.0066 -
val_loss: 0.2033
Epoch 21/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0053 -
val_loss: 0.2129
Epoch 22/25
1500/1500 [==============================] - 8s 6ms/step - loss: 0.0067 -
val_loss: 0.1999
Epoch 23/25
1500/1500 [==============================] - 8s 5ms/step - loss: 0.0049 -
val_loss: 0.2121
Epoch 24/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0064 -
val_loss: 0.2176
Epoch 25/25
1500/1500 [==============================] - 9s 6ms/step - loss: 0.0036 -
val_loss: 0.2021
```

[42]:

```
y_prob = model.predict(X_test)
```

```
313/313 [==============================] - 1s 2ms/step
```

[43]:

```
y_pred = y_prob.argmax(axis = 1)
```

[44]:

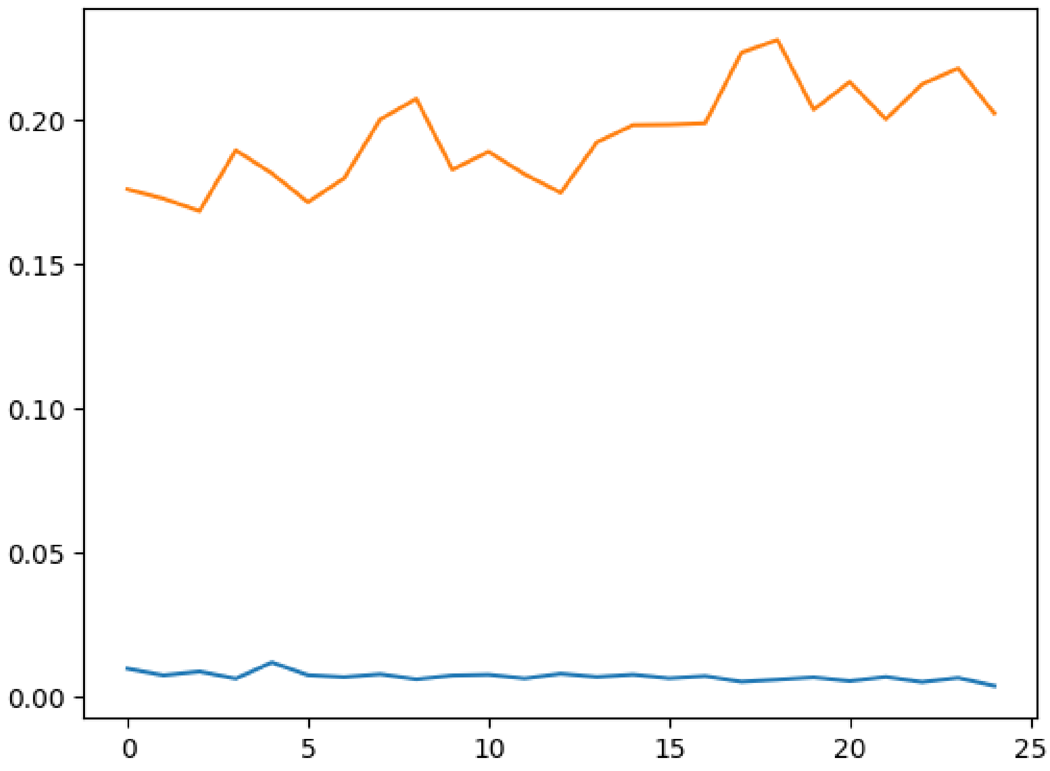
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

[44]: 0.9809

[48]:

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

```
[48]: [<matplotlib.lines.Line2D at 0x7f2564288f70>]
```



[47]:

```
history.history
```

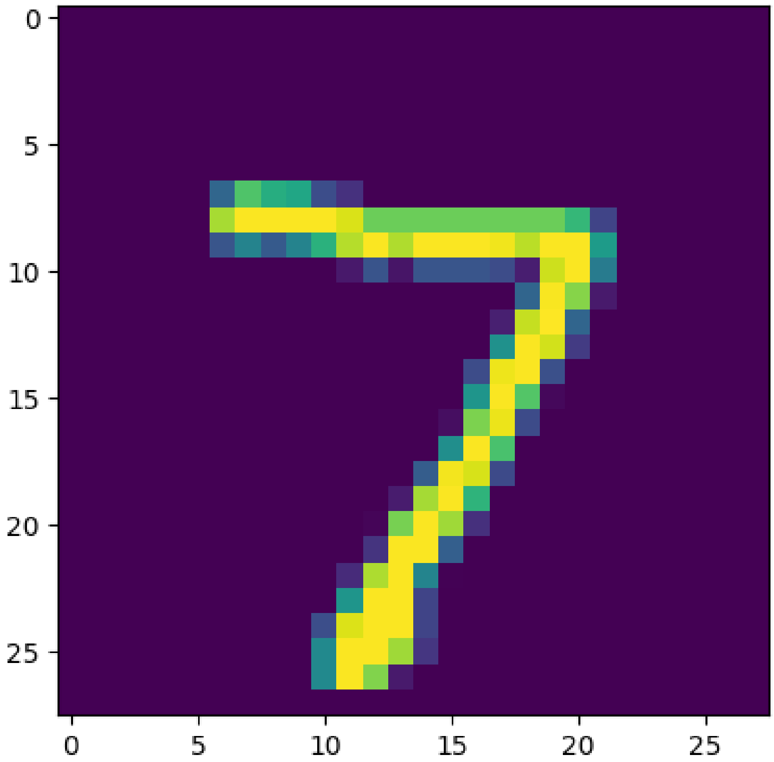
```
[47]: {'loss': [0.009598016738891602,
        0.007257132790982723,
        0.008580993860960007,
        0.006104324944317341,
        0.011731773614883423,
        0.007329855114221573,
        0.006635897792875767,
        0.007652199361473322,
        0.005880948156118393,
        0.007205863483250141,
        0.007479384075850248,
        0.006156647577881813,
        0.007820403203368187,
        0.006691309157758951,
        0.007504675537347794,
        0.006216663401573896,
        0.006981916259974241,
        0.005026218947023153,
        0.0057679121382534 5,
        0.006565769668668508 5,
        0.005262769293040037,
        0.006713254377245903,
        0.004923920147120953,
        0.006385253742337227,
        0.003608071943745017],
       'val_loss': [0.17568786442279816,
        0.1724545657634735,
        0.16816911101341248,
        0.18922746181488037,
        0.18122217059135437,
        0.17122788727283478,
        0.17948123812675476,
        0.19989928603172302,
        0.20703282952308655,
        0.1825072169303894,
        0.18871034681797028,
        0.18089522421360016,
        0.17447751760482788,
        0.1920027881860733,
        0.19793404638767242,
        0.1980905830860138,
        0.19857531785964966,
        0.22302475571632385,
        0.22745320200920105,
        0.20326794683933258,
        0.21290333569049835,
        0.1999477595090866,
        0.21208538115024567,
        0.21761031448841095,
        0.20206433534622192]}
```

[49]:

```
plt.imshow(X_test[0])
```

[49]:

```
<matplotlib.image.AxesImage at 0x7f25640dad30>
```



```
[53]: model.predict(X_test[0].reshape(1,28,28)).argmax(axis = 1)
```

```
1/1 [==============================] - 0s 87ms/step
```

```
[53]: array([7])
```

```
[54]: model.predict(X_test[3].reshape(1,28,28)).argmax(axis = 1)
```

```
1/1 [==============================] - 0s 23ms/step
```

```
[54]: array([0])
```