In [1]:

**import** tensorflow **as** tf

In [3]:

**from** tensorflow **import** keras

In [4]:

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

In [6]:

(x\_train, y\_train), (x\_test, y\_test)**=**keras**.**datasets**.**fashion\_mnist**.**load\_data()

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/tr ain-labels-idx1-ubyte.gz

32768/29515 [=================================] - 0s 1us/step

40960/29515 [=========================================] - 0s 1us/step

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/tr ain-images-idx3-ubyte.gz

26427392/26421880 [==============================] - 16s 1us/step

26435584/26421880 [==============================] - 16s 1us/step

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t1 0k-labels-idx1-ubyte.gz

16384/5148 [========================================================================

=======================] - 0s 0s/step

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t1 0k-images-idx3-ubyte.gz

4423680/4422102 [==============================] - 3s 1us/step

4431872/4422102 [==============================] - 3s 1us/step

In [20]:

x\_train**=**x\_train**.**astype('float32')**/**255.0 x\_test**=**x\_test**.**astype('float32')**/**255.0

In [21]:

x\_train**=**x\_train**.**reshape(**-**1,28,28,1) x\_test**=**x\_test**.**reshape(**-**1,28,28,1)

In [22]:

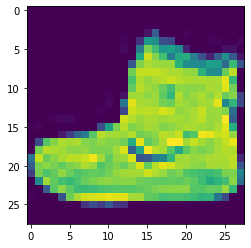
plt**.**imshow(x\_train[0])

Out[22]:

In [23]:

plt**.**imshow(x\_train[10])

<matplotlib.image.AxesImage at 0x1e481ad72e0>



Out[23]:

In [24]:

x\_train**.**shape

<matplotlib.image.AxesImage at 0x1e481b18d60>



Out[24]:

In [25]:

cnn**=**tf**.**keras**.**Sequential()

(60000, 28, 28, 1)

In [26]:

cnn**.**add(tf**.**keras**.**layers**.**Conv2D(32,(3,3),activation**=**'relu',input\_shape**=**(28,28,1))) cnn**.**add(tf**.**keras**.**layers**.**Conv2D(64,(3,3),activation**=**'relu'))

cnn**.**add(tf**.**keras**.**layers**.**Dropout(0.2))

cnn**.**add(tf**.**keras**.**layers**.**Conv2D(128,(3,3),activation**=**'relu')) cnn**.**add(tf**.**keras**.**layers**.**Flatten())

cnn**.**add(tf**.**keras**.**layers**.**Dense(128,activation**=**'relu'))

cnn**.**add(tf**.**keras**.**layers**.**Dropout(0.2))

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

In [29]:

cnn**.**compile(optimizer**=**'Adam',loss**=**'sparse\_categorical\_crossentropy',metrics**=**['accura

In [31]:

cnn**.**summary()

Model: "sequential\_1"

Layer (type) Output Shape Param #

=================================================================

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| conv2d\_3 (Conv2D) | (None, | 26, | 26, | 32) | 320 |
| conv2d\_4 (Conv2D) | (None, | 24, | 24, | 64) | 18496 |
| dropout\_2 (Dropout) | (None, | 24, | 24, | 64) | 0 |
| conv2d\_5 (Conv2D) | (None, | 22, | 22, | 128) | 73856 |

flatten\_1 (Flatten) (None, 61952) 0

|  |  |  |  |
| --- | --- | --- | --- |
| dense\_2 (Dense) | (None, | 128) | 7929984 |
| dropout\_3 (Dropout) | (None, | 128) | 0 |
| dense\_3 (Dense) | (None, | 10) | 1290 |

=================================================================

Total params: 8,023,946

Trainable params: 8,023,946

Non-trainable params: 0

In [30]:

cnn**.**fit(x\_train,y\_train,batch\_size**=**10,validation\_data**=**(x\_test,y\_test))

Out[30]:

In [35]:

6000/6000 [==============================] - 501s 83ms/step - loss: 2.3029 - accurac

y: 0.1005 - val\_loss: 2.3028 - val\_accuracy: 0.1000

<keras.callbacks.History at 0x1e481bc5fd0>

313/313 [==============================] - 14s 46ms/step - loss: 2.3028 - accuracy:

a,accuracy**=**cnn**.**evaluate(x\_test,y\_test)

0.1000

In [36]:

print("Test Accuracy :- ",accuracy)

Test Accuracy :- 0.10000000149011612

In [ ]: