

```
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import fashion_mnist
from tensorflow.keras import layers, models
```

```
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 [=====] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 [=====] - 0s 0us/step
```

```
train_images.shape
```

```
(60000, 28, 28)
```

```
len(train_labels)
```

```
60000
```

```
test_images.shape
```

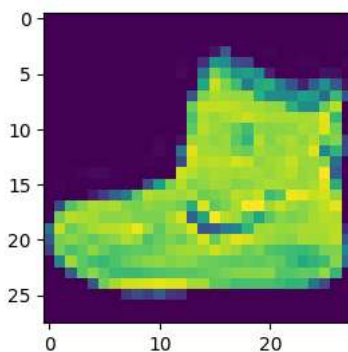
```
(10000, 28, 28)
```

```
len(test_labels)
```

```
10000
```

```
train_images = train_images.reshape(60000, 28, 28, 1)
test_images = test_images.reshape(10000, 28, 28, 1)
```

```
plt.figure(figsize=(3,3))
plt.imshow(train_images[0])
plt.grid=False
plt.show()
```



```
train_images = train_images / 255.0
test_images = test_images / 255.0
```

```
model = models.Sequential()
```

```
model.add(layers.Conv2D(64, (3,3), activation='relu', input_shape=(28,28,1)))
model.add(layers.MaxPooling2D(2,2))
model.add(layers.Conv2D(64, (3,3), activation='relu'))
model.add(layers.MaxPooling2D(2,2))
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))
```

```
model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['Accuracy'])

model.fit(train_images,train_labels,epochs=3,validation_data=(test_images,test_labels),verbose=1)

Epoch 1/3
1875/1875 [=====] - 86s 45ms/step - loss: 0.4417 - Accuracy: 0.8382 - val_loss: 0.3510 - val_Accuracy: 0.8730
Epoch 2/3
1875/1875 [=====] - 80s 43ms/step - loss: 0.2937 - Accuracy: 0.8913 - val_loss: 0.2882 - val_Accuracy: 0.8979
Epoch 3/3
1875/1875 [=====] - 87s 46ms/step - loss: 0.2456 - Accuracy: 0.9090 - val_loss: 0.2665 - val_Accuracy: 0.9020
<keras.src.callbacks.History at 0x7bc801a4f940>

loss1,acc1=model.evaluate(train_images,train_labels)
print("training accuracy :",acc1)
loss2,acc2=model.evaluate(test_images,test_labels)
print("testing accuracy :",acc2)

1875/1875 [=====] - 23s 12ms/step - loss: 0.2062 - Accuracy: 0.9224
training accuracy : 0.9224166870117188
313/313 [=====] - 4s 13ms/step - loss: 0.2665 - Accuracy: 0.9020
testing accuracy : 0.9020000100135803

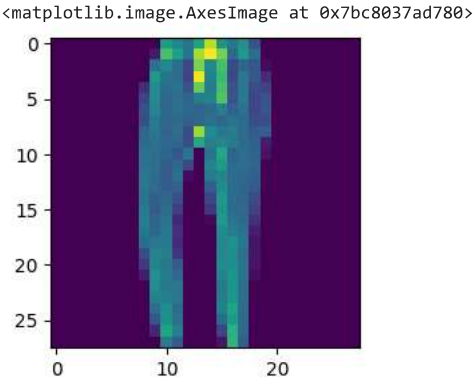
predictions=model.predict(test_images)

313/313 [=====] - 4s 12ms/step

index=80
label=np.argmax(predictions[index])
label

1

plt.figure(figsize=(3,3))
plt.imshow(test_images[index])
```



```
model.summary()

Model: "sequential_1"

Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 26, 26, 64)         640
max_pooling2d (MaxPooling2D) (None, 13, 13, 64)         0
conv2d_1 (Conv2D)             (None, 11, 11, 64)         36928
max_pooling2d_1 (MaxPoolin   (None, 5, 5, 64)           0
g2D)
flatten (Flatten)            (None, 1600)                0
dense_3 (Dense)               (None, 128)                 204928
dense_4 (Dense)               (None, 10)                  1290
=====
Total params: 243786 (952.29 KB)
Trainable params: 243786 (952.29 KB)
```

Non-trainable params: 0 (0.00 Byte)

---

Start coding or [generate](#) with AI.