

```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

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

```
(xtrain,ytrain),(xtest,ytest)=keras.datasets.fashion_mnist.load_data()
```

```
xtrain.shape
```

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

```
xtrain[0].shape
```

```
(28, 28)
```

```
xtrain[0]
```

```
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, 13, 73,  0,  0,  1,  4,  0,  0,  0,  0,  1,
         1,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0, 36, 136, 127, 62, 54,  0,  0,  0,  1,  3,  4,  0,
         0,  3],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0, 102, 204, 176, 134, 144, 123, 23,  0,  0,  0,  0, 12,
        10,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
         0, 155, 236, 207, 178, 107, 156, 161, 109, 64, 23, 77, 130,
        72, 15],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  0,
        69, 207, 223, 218, 216, 216, 163, 127, 121, 122, 146, 141, 88,
       172, 66],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  1,  1,  0,
       200, 232, 232, 233, 229, 223, 223, 215, 213, 164, 127, 123, 196,
       229,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
       183, 225, 216, 223, 228, 235, 227, 224, 222, 224, 221, 223, 245,
       173,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
       193, 228, 218, 213, 198, 180, 212, 210, 211, 213, 223, 220, 243,
       202,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  1,  3,  0, 12,
       219, 220, 212, 218, 192, 169, 227, 208, 218, 224, 212, 226, 197,
       209, 52],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  6,  0, 99,
       244, 222, 220, 218, 203, 198, 221, 215, 213, 222, 220, 245, 119,
       167, 56],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  4,  0,  0, 55,
       236, 228, 230, 228, 240, 232, 213, 218, 223, 234, 217, 217, 209,
       92,  0],
       [ 0,  0,  1,  4,  6,  7,  2,  0,  0,  0,  0,  0,  0, 237,
       226, 217, 223, 222, 219, 222, 221, 216, 223, 229, 215, 218, 255,
       77,  0],
       [ 0,  3,  0,  0,  0,  0,  0,  0,  0,  0, 62, 145, 204, 228,
       207, 213, 221, 218, 208, 211, 218, 224, 223, 219, 215, 224, 244,
       159,  0],
       [ 0,  0,  0,  0, 18, 44, 82, 107, 189, 228, 220, 222, 217,
       226, 200, 205, 211, 230, 224, 234, 176, 188, 250, 248, 233, 238,
       215,  0],
       [ 0, 57, 187, 208, 224, 221, 224, 208, 204, 214, 208, 209, 200,
       159, 245, 193, 206, 223, 255, 255, 221, 234, 221, 211, 220, 232,
       246,  0],
       [ 3, 202, 228, 224, 221, 211, 211, 214, 205, 205, 205, 220, 240,
       80, 150, 255, 229, 221, 188, 154, 191, 210, 204, 209, 222, 228,
       225,  0],
       [ 98, 233, 198, 210, 222, 229, 229, 234, 249, 220, 194, 215, 217,
```

```
<matplotlib.image.AxesImage at 0x7f72a98f82b0>
```

(60000,)

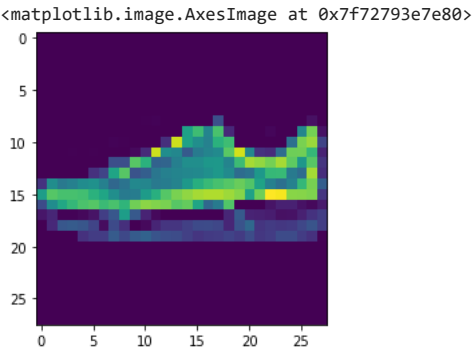
(10000, 28, 28)

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,	0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0],										
[0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
	0,	0,	0,	0,	0,	0,	3,	1,	0,	0,	7,	0,
	0,	0],										37,
[0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
	1,	2,	0,	27,	84,	11,	0,	0,	0,	0,	0,	119,
	0,	0],										
[0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
	1,	0,	0,	88,	143,	110,	0,	0,	0,	0,	22,	93,
	0,	0],										106,
[0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
	4,	0,	53,	129,	120,	147,	175,	157,	166,	135,	154,	168,
	0,	0],										140,
[0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	2,
	0,	11,	137,	130,	128,	160,	176,	159,	167,	178,	149,	151,
	0,	0],										144,
[0,	0,	0,	0,	0,	0,	1,	0,	2,	1,	0,	3,
	0,	115,	114,	106,	137,	168,	153,	156,	165,	167,	143,	157,
	11,	0],										158,
[0,	0,	0,	0,	1,	0,	0,	0,	0,	0,	3,	0,
	89,	139,	90,	94,	153,	149,	131,	151,	169,	172,	143,	159,
	48,	0],										169,
[0,	0,	0,	0,	0,	0,	2,	4,	1,	0,	0,	0,
	136,	110,	109,	110,	162,	135,	144,	149,	159,	167,	144,	158,
	119,	0],										169,
[0,	0,	2,	2,	1,	2,	0,	0,	0,	0,	26,	108,
	99,	111,	117,	136,	156,	134,	154,	154,	156,	160,	141,	147,
	178,	0],										156,
[3,	0,	0,	0,	0,	0,	0,	21,	53,	92,	117,	111,
	115,	129,	134,	143,	154,	165,	170,	154,	151,	154,	143,	138,
	165,	43],										150,
[0,	0,	23,	54,	65,	76,	85,	118,	128,	123,	111,	113,

ytest

```
array([9, 2, 1, ..., 8, 1, 5], dtype=uint8)
```

plt.imshow(xtest[9])



```
xtrain = xtrain/255
xtest  = xtest/255
```

```
# model building

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

# add output layer
model.add(Dense(10, activation = 'softmax'))

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

history = model.fit(xtrain, ytrain, epochs = 4, validation_split = 0.2)
```

```
Epoch 1/4
1500/1500 [=====] - 13s 8ms/step - loss: 0.5165 - accuracy: 0.8150 - val_loss: 0.4215 - val_accuracy: 0.84
Epoch 2/4
1500/1500 [=====] - 8s 5ms/step - loss: 0.3844 - accuracy: 0.8601 - val_loss: 0.3691 - val_accuracy: 0.866
Epoch 3/4
1500/1500 [=====] - 6s 4ms/step - loss: 0.3441 - accuracy: 0.8731 - val_loss: 0.3627 - val_accuracy: 0.866
Epoch 4/4
1500/1500 [=====] - 8s 5ms/step - loss: 0.3179 - accuracy: 0.8816 - val_loss: 0.3363 - val_accuracy: 0.880
```

```
yprob=model.predict(xtest)
yprob[0]
```

```
313/313 [=====] - 1s 2ms/step
array([9.8123848e-05, 1.2610356e-05, 3.0464040e-05, 4.3143787e-06,
       1.1299426e-04, 6.3377484e-03, 7.2911283e-04, 6.7731164e-02,
       2.9188456e-04, 9.2465156e-01], dtype=float32)
```

```
ypred=yprob.argmax(axis=1)
ypred
```

```
array([9, 2, 1, ..., 8, 1, 5])
```

```
from sklearn.metrics import classification_report
print(classification_report(ytest,ypred))
```

	precision	recall	f1-score	support
0	0.81	0.85	0.83	1000
1	0.99	0.96	0.98	1000
2	0.78	0.72	0.75	1000
3	0.89	0.86	0.88	1000
4	0.70	0.87	0.78	1000
5	0.98	0.94	0.96	1000
6	0.74	0.61	0.67	1000
7	0.92	0.96	0.94	1000
8	0.96	0.97	0.97	1000

	9	0.96	0.95	0.95	1000
accuracy				0.87	10000
macro avg		0.87	0.87	0.87	10000
weighted avg		0.87	0.87	0.87	10000

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

