```
import tensorflow as tf
import keras


mnist = tf.keras.datasets.fashion_mnist


(training_images, training_labels), (test_images,test_labels) =  mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-
32768/29515 [================================] - 0s 0us/step
40960/29515 [========================================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-
26427392/26421880 [==============================] - 0s 0us/step
26435584/26421880 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-l
16384/5148 [================================================================================
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-i
4423680/4422102 [==============================] - 0s 0us/step
4431872/4422102 [==============================] - 0s 0us/step
```

```
training_images = training_images/255.0
test_images = test_images/255.0


model·=·tf.keras.models.Sequential([tf.keras.layers.Flatten(),
····································tf.keras.layers.Dense(512,·activation·=·tf.nn.relu),
····································tf.keras.layers.Dense(10,activation·=·tf.nn.softmax)])


model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy')


model.fit(training_images, training_labels, epochs = 5)
```

```
Epoch 1/5
1875/1875 [==============================] - 12s 6ms/step - loss: 0.8658
Epoch 2/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.5434
Epoch 3/5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.4846
Epoch 4/5
1875/1875 [==============================] - 10s 5ms/step - loss: 0.4538
Epoch 5/5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.4334
<keras.callbacks.History at 0x7f5ccc6f1d90>
```

```
model.fit(training_images, training_labels, epochs = 10)
```

```
Epoch 1/10
1875/1875 [==============================] - 9s 5ms/step - loss: 0.4179
```

```
Epoch 2/10
1875/1875 [==============================] - 8s 5ms/step - loss: 0.4058
Epoch 3/10
1875/1875 [==============================] - 12s 6ms/step - loss: 0.3954
Epoch 4/10
1875/1875 [==============================] - 14s 7ms/step - loss: 0.3861
Epoch 5/10
1875/1875 [==============================] - 12s 7ms/step - loss: 0.3783
Epoch 6/10
1875/1875 [==============================] - 8s 4ms/step - loss: 0.3710
Epoch 7/10
1875/1875 [==============================] - 8s 4ms/step - loss: 0.3634
Epoch 8/10
1875/1875 [==============================] - 8s 4ms/step - loss: 0.3571
Epoch 9/10
1875/1875 [==============================] - 9s 5ms/step - loss: 0.3508
Epoch 10/10
1875/1875 [==============================] - 8s 5ms/step - loss: 0.3453
<keras.callbacks.History at 0x7f5ccc2833d0>
```

```
model.evaluate(test_images, test_labels)
```

```
313/313 [==============================] - 1s 3ms/step - loss: 82.6850
82.6849594116211
```

```
classification = model.predict(test_images)
```

```
print(classification[0])
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
```

```
print(test_labels[0])
```

```
9
```

```
import matplotlib.pyplot as plt
```

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

```
<matplotlib.image.AxesImage at 0x7f5cc4dd6790>
```


```
print(training_labels[0])
```

```
9
```

```
print(training_images[0])
```

```
 0.00000000e+00 2.96808920e-03 3.50634371e-03 3.35255671e-03
 3.27566321e-03 3.04498270e-03 2.76816609e-03 3.26028451e-03
 3.22952710e-03 3.24490581e-03 3.27566321e-03 3.42945021e-03
 3.38331411e-03 3.73702422e-03 3.10649750e-03 0.00000000e+00]
[0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 1.53787005e-05 4.61361015e-05 0.00000000e+00
 1.84544406e-04 3.36793541e-03 3.38331411e-03 3.26028451e-03
 3.35255671e-03 2.95271050e-03 2.59900038e-03 3.49096501e-03
 3.19876970e-03 3.35255671e-03 3.44482891e-03 3.26028451e-03
 3.47558631e-03 3.02960400e-03 3.21414840e-03 7.99692426e-04]
[0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 9.22722030e-05 0.00000000e+00
 1.52249135e-03 3.75240292e-03 3.41407151e-03 3.38331411e-03
 3.35255671e-03 3.12187620e-03 3.04498270e-03 3.39869281e-03
 3.30642061e-03 3.27566321e-03 3.41407151e-03 3.38331411e-03
 3.76778162e-03 1.83006536e-03 2.56824298e-03 8.61207228e-04]
[0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 6.15148020e-05 0.00000000e+00 0.00000000e+00
 8.45828527e-04 3.62937332e-03 3.50634371e-03 3.53710111e-03
 3.50634371e-03 3.69088812e-03 3.56785852e-03 3.27566321e-03
 3.35255671e-03 3.42945021e-03 3.59861592e-03 3.33717801e-03
 3.33717801e-03 3.21414840e-03 1.41484045e-03 0.00000000e+00]
[0.00000000e+00 0.00000000e+00 1.53787005e-05 6.15148020e-05
 9.22722030e-05 1.07650903e-04 3.07574010e-05 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 3.64475202e-03 3.47558631e-03 3.33717801e-03 3.42945021e-03
 3.41407151e-03 3.36793541e-03 3.41407151e-03 3.39869281e-03
 3.32179931e-03 3.42945021e-03 3.52172241e-03 3.30642061e-03
 3.35255671e-03 3.92156863e-03 1.18415994e-03 0.00000000e+00]
[0.00000000e+00 4.61361015e-05 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 9.53479431e-04 2.22991157e-03 3.13725490e-03
 3.50634371e-03 3.18339100e-03 3.27566321e-03 3.39869281e-03
 3.35255671e-03 3.19876970e-03 3.24490581e-03 3.35255671e-03
 3.44482891e-03 3.42945021e-03 3.36793541e-03 3.30642061e-03
 3.44482891e-03 3.75240292e-03 2.44521338e-03 0.00000000e+00]
[0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 2.76816609e-04 6.76662822e-04 1.26105344e-03 1.64552095e-03
 2.90657439e-03 3.50634371e-03 3.38331411e-03 3.41407151e-03
 3.33717801e-03 3.47558631e-03 3.07574010e-03 3.15263360e-03
 3.24490581e-03 3.53710111e-03 3.44482891e-03 3.59861592e-03
```

```
  2.70665129e-03 2.89119569e-03 3.84467512e-03 3.81391772e-03
  3.58323722e-03 3.66013072e-03 3.30642061e-03 0.00000000e+00]
 [0.00000000e+00 8.76585928e-04 2.87581699e-03 3.19876970e-03
  3.44482891e-03 3.39869281e-03 3.44482891e-03 3.19876970e-03
  3.13725490e-03 3.29104191e-03 3.19876970e-03 3.21414840e-03
  3.07574010e-03 2.44521338e-03 3.76778162e-03 2.96808920e-03
  3.16801230e-03 3.42945021e-03 3.92156863e-03 3.92156863e-03
  3.39869281e-03 3.59861592e-03 3.39869281e-03 3.24490581e-03
  3.38331411e-03 3.56785852e-03 3.78316032e-03 0.00000000e+00]
 [4.61361015e-05 3.10649750e-03 3.50634371e-03 3.44482891e-03
  3.39869281e-03 3.24490581e-03 3.24490581e-03 3.29104191e-03
  3.15263360e-03 3.15263360e-03 3.15263360e-03 3.38331411e-03
  3.69088812e-03 1.23029604e-03 2.30680507e-03 3.92156863e-03
  3.52172241e-03 3.39869281e-03 2.89119569e-03 2.36831988e-03
```

```
model = tf.keras.models.Sequential([tf.keras.layers.Flatten(),
                                    tf.keras.layers.Dense(512, activation = tf.nn.relu),
                                    tf.keras.layers.Dense(10,activation = tf.nn.softmax)])
model.compile(optimizer = 'sgd', loss = 'sparse_categorical_crossentropy')


model.fit(training_images, training_labels, epochs = 5)
```

```
Epoch 1/5
1875/1875 [==============================] - 8s 4ms/step - loss: 2.2992
Epoch 2/5
1875/1875 [==============================] - 8s 4ms/step - loss: 2.2980
Epoch 3/5
1875/1875 [==============================] - 9s 5ms/step - loss: 2.2968
Epoch 4/5
1875/1875 [==============================] - 9s 5ms/step - loss: 2.2954
Epoch 5/5
1875/1875 [==============================] - 8s 4ms/step - loss: 2.2938
<keras.callbacks.History at 0x7f5cc5a97290>
```

```
model = tf.keras.models.Sequential([tf.keras.layers.Flatten(),
                                    tf.keras.layers.Dense(512, activation = tf.nn.relu),
                                    tf.keras.layers.Dense(10,activation = tf.nn.softmax)])
model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy')


model.fit(training_images, training_labels, epochs = 5)
```

```
Epoch 1/5
1875/1875 [==============================] - 10s 5ms/step - loss: 0.8764
Epoch 2/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.5512
Epoch 3/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.4916
Epoch 4/5
1875/1875 [==============================] - 9s 5ms/step - loss: 0.4598
Epoch 5/5
1875/1875 [==============================] - 8s 5ms/step - loss: 0.4389
<keras.callbacks.History at 0x7f5cc59d9a50>
```

```
model = tf.keras.models.Sequential([tf.keras.layers.Flatten(),
                                    tf.keras.layers.Dense(512, activation = tf.nn.relu),
                                    tf.keras.layers.Dense(10,activation = tf.nn.softmax)])
model.compile(optimizer = 'rmsprop', loss = 'sparse_categorical_crossentropy')


model.fit(training_images, training_labels, epochs = 5)
```

```
Epoch 1/5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.8864
Epoch 2/5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.5461
Epoch 3/5
1875/1875 [==============================] - 12s 6ms/step - loss: 0.4851
Epoch 4/5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.4522
Epoch 5/5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.4315
<keras.callbacks.History at 0x7f5cc5922410>
```

✓ 56s completed at 8:08 PM