

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers,models
```

```
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()
```

```
x_train, x_test = x_train / 255.0, x_test / 255.0
```

```
model = models.Sequential([
    layers.Flatten(input_shape=(28, 28)),
    layers.Dense(128, activation='relu'),
    layers.Dense(64, activation='relu'),
    layers.Dense(10, activation='softmax') ])
```

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

```
model.fit(x_train, y_train, epochs=5, validation_data=(x_test, y_test))
```

```
Epoch 1/5
1875/1875 [=====] - 11s 5ms/step - loss: 0.2427 - accuracy: 0.9293 - val_loss: 0.1287 - val_accuracy: 0.9612
Epoch 2/5
1875/1875 [=====] - 10s 5ms/step - loss: 0.1018 - accuracy: 0.9690 - val_loss: 0.1021 - val_accuracy: 0.9691
Epoch 3/5
1875/1875 [=====] - 8s 5ms/step - loss: 0.0727 - accuracy: 0.9772 - val_loss: 0.0888 - val_accuracy: 0.9722
Epoch 4/5
1875/1875 [=====] - 10s 5ms/step - loss: 0.0553 - accuracy: 0.9834 - val_loss: 0.0932 - val_accuracy: 0.9719
Epoch 5/5
1875/1875 [=====] - 10s 5ms/step - loss: 0.0424 - accuracy: 0.9868 - val_loss: 0.1106 - val_accuracy: 0.9675
<keras.src.callbacks.History at 0x7f8b9ba46350>
```

```
test_loss, test_accuracy = model.evaluate(x_test, y_test)
print("Test accuracy: ",test_accuracy*100)
```

```
313/313 [=====] - 1s 2ms/step - loss: 0.0839 - accuracy: 0.9747
Test accuracy: 97.46999740600586
```

