


```

from keras import layers, models
from tensorflow import keras

mnist = keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()

 Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [=====] - 0s 0us/step

x_train, x_test = x_train / 255.0, x_test / 255.0

x_train_flattened = x_train.reshape((x_train.shape[0], -1))

x_test_flattened = x_test.reshape((x_test.shape[0], -1))

model = models.Sequential([
    layers.Dense(5, activation='relu', input_shape=(784,)),
    layers.Dense(5, activation='relu'),
    layers.Dense(5, activation='relu'),
    layers.Dense(5, activation='relu'),
    layers.Dense(5, activation='relu'),
    layers.Dense(10, activation='softmax')
])

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

history = model.fit(x_train_flattened, y_train, epochs=10, validation_data=(x_test_flattened, y_test))

Epoch 1/10
1875/1875 [=====] - 9s 3ms/step - loss: 1.4121 - accuracy: 0.4984 - val_loss: 0.9452 - val_accuracy: 0.7116
Epoch 2/10
1875/1875 [=====] - 6s 3ms/step - loss: 0.8063 - accuracy: 0.7638 - val_loss: 0.6990 - val_accuracy: 0.8074
Epoch 3/10
1875/1875 [=====] - 5s 3ms/step - loss: 0.6764 - accuracy: 0.8069 - val_loss: 0.6429 - val_accuracy: 0.8225
Epoch 4/10
1875/1875 [=====] - 5s 3ms/step - loss: 0.6305 - accuracy: 0.8207 - val_loss: 0.6153 - val_accuracy: 0.8337
Epoch 5/10
1875/1875 [=====] - 6s 3ms/step - loss: 0.6021 - accuracy: 0.8283 - val_loss: 0.6079 - val_accuracy: 0.8324
Epoch 6/10
1875/1875 [=====] - 5s 3ms/step - loss: 0.5821 - accuracy: 0.8330 - val_loss: 0.5830 - val_accuracy: 0.8375
Epoch 7/10
1875/1875 [=====] - 6s 3ms/step - loss: 0.5706 - accuracy: 0.8359 - val_loss: 0.5796 - val_accuracy: 0.8371
Epoch 8/10
1875/1875 [=====] - 5s 3ms/step - loss: 0.5587 - accuracy: 0.8389 - val_loss: 0.5879 - val_accuracy: 0.8365
Epoch 9/10
1875/1875 [=====] - 6s 3ms/step - loss: 0.5495 - accuracy: 0.8411 - val_loss: 0.5642 - val_accuracy: 0.8454
Epoch 10/10
1875/1875 [=====] - 6s 3ms/step - loss: 0.5423 - accuracy: 0.8436 - val_loss: 0.5570 - val_accuracy: 0.8468

test_loss, test_acc = model.evaluate(x_test_flattened, y_test)

313/313 [=====] - 1s 4ms/step - loss: 0.5570 - accuracy: 0.8468

print(f'Test Accuracy: {test_acc * 100:.2f}%')

Test Accuracy: 84.68%

```

