

```
In [1]: import tensorflow as tf
print("TensorFlow version:", tf.__version__)
```

TensorFlow version: 2.8.0

```
In [2]: mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz> (<https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>)

11493376/11490434 [=====] - 27s 2us/step

11501568/11490434 [=====] - 27s 2us/step

```
In [3]: model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10)
])
```

```
In [4]: predictions = model(x_train[:1]).numpy()
predictions
```

```
Out[4]: array([[ -0.10172732, -0.1430343 ,  0.34567282, -0.14304838, -0.592537  ,
                -0.40135938,  0.03544755, -0.34366286, -0.49084222,  0.63207805]],
              dtype=float32)
```

```
In [5]: tf.nn.softmax(predictions).numpy()
```

```
Out[5]: array([[0.09497306, 0.09112994, 0.14856067, 0.09112865, 0.05813591,
                0.07038366, 0.10893682, 0.074564  , 0.06435911, 0.19782814]],
              dtype=float32)
```

```
In [6]: loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
```

```
In [7]: loss_fn(y_train[:1], predictions).numpy()
```

```
Out[7]: 2.653794
```

```
In [8]: model.compile(optimizer='adam',  
                      loss=loss_fn,  
                      metrics=['accuracy'])
```

```
In [9]: model.fit(x_train, y_train, epochs=5)
```

```
Epoch 1/5  
1875/1875 [=====] - 8s 4ms/step - loss: 0.2970 - accuracy: 0.9135  
Epoch 2/5  
1875/1875 [=====] - 6s 3ms/step - loss: 0.1451 - accuracy: 0.9575  
Epoch 3/5  
1875/1875 [=====] - 5s 3ms/step - loss: 0.1094 - accuracy: 0.9670  
Epoch 4/5  
1875/1875 [=====] - 6s 3ms/step - loss: 0.0895 - accuracy: 0.9725  
Epoch 5/5  
1875/1875 [=====] - 6s 3ms/step - loss: 0.0744 - accuracy: 0.9766
```

```
Out[9]: <keras.callbacks.History at 0x21053c60670>
```

```
In [10]: model.evaluate(x_test, y_test, verbose=2)
```

```
313/313 - 4s - loss: 0.0765 - accuracy: 0.9771 - 4s/epoch - 12ms/step
```

```
Out[10]: [0.07646225392818451, 0.9771000146865845]
```

```
In [11]: probability_model = tf.keras.Sequential([  
        model,  
        tf.keras.layers.Softmax()  
    ])
```

```
In [12]: probability_model(x_test[:5])
```

```
Out[12]: <tf.Tensor: shape=(5, 10), dtype=float32, numpy=
array([[5.02216047e-08, 1.43390799e-09, 1.42837314e-06, 8.00553535e-05,
        3.08454694e-12, 8.18623676e-08, 4.31927850e-15, 9.99915481e-01,
        2.18511673e-07, 2.59140234e-06],
       [9.59392921e-09, 2.27662531e-05, 9.99710143e-01, 2.24533127e-04,
        1.08981195e-17, 1.30096794e-07, 2.23396377e-08, 3.21911348e-14,
        4.24640602e-05, 5.29533015e-13],
       [4.70789587e-07, 9.98622656e-01, 3.28409573e-04, 1.66902973e-05,
        7.75446169e-05, 1.09159919e-05, 3.50928094e-05, 4.01466823e-04,
        5.01114991e-04, 5.58671718e-06],
       [9.99870777e-01, 8.24866930e-09, 2.30555906e-05, 1.22019208e-06,
        1.02376191e-07, 3.71570168e-05, 3.44179011e-06, 1.15974917e-05,
        4.88142234e-07, 5.22540286e-05],
       [2.70520013e-06, 1.17842958e-09, 4.67526343e-06, 3.94038828e-08,
        9.95431662e-01, 1.27763599e-07, 5.56070781e-08, 2.53126855e-05,
        4.42061804e-07, 4.53497050e-03]], dtype=float32)>
```

```
In [ ]:
```