```
import tensorflow as tf
print("TensorFlow version:", tf.__version__)

     TensorFlow version: 2.13.0
```

```
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
     11490434/11490434 [==============================] - 1s 0us/step
```

```
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)
])
```

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

     array([[-0.7352824 , -0.26631802, -0.36999696, -0.3064358 ,  0.13626602,
             -0.09854539, -0.4259666 , -0.08054319,  0.00721704,  0.04621303]],
           dtype=float32)
```

```
tf.nn.softmax(predictions).numpy()

     array([[0.05737658, 0.09170717, 0.08267536, 0.08810091, 0.13716501,
             0.10845911, 0.07817516, 0.11042929, 0.12055857, 0.12535274]],
           dtype=float32)
```

```
loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
```

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

     2.221382
```

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

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

     Epoch 1/5
     1875/1875 [==============================] - 7s 3ms/step - loss: 0.2990 - accuracy: 0.9130
     Epoch 2/5
     1875/1875 [==============================] - 10s 5ms/step - loss: 0.1446 - accuracy: 0.9574
     Epoch 3/5
     1875/1875 [==============================] - 8s 4ms/step - loss: 0.1095 - accuracy: 0.9668
     Epoch 4/5
     1875/1875 [==============================] - 7s 3ms/step - loss: 0.0889 - accuracy: 0.9728
     Epoch 5/5
     1875/1875 [==============================] - 7s 4ms/step - loss: 0.0773 - accuracy: 0.9758
     <keras.src.callbacks.History at 0x79c69d982260>
```

```
model.evaluate(x_test, y_test, verbose=2)

     313/313 - 1s - loss: 0.0758 - accuracy: 0.9767 - 612ms/epoch - 2ms/step
     [0.07579353451728821, 0.9767000079154968]
```

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

```
probability_model(x_test[:5])

     <tf.Tensor: shape=(5, 10), dtype=float32, numpy=
     array([[4.61580889e-08, 2.92651836e-09, 1.04167123e-06, 3.17492522e-05,
             3.14701487e-10, 2.70651890e-06, 2.29337955e-11, 9.99955893e-01,
             2.56232283e-07, 8.26838914e-06],
            [3.09847953e-10, 1.38110045e-04, 9.99839783e-01, 1.57335344e-05,
             8.72714258e-17, 3.94686549e-06, 1.02908109e-06, 3.72989348e-11,
             1.25277211e-06, 1.02557204e-13],
            [7.39636221e-07, 9.99247789e-01, 7.26030048e-05, 1.01415189e-05,
             4.71060121e-05, 1.77183801e-06, 5.10402015e-06, 4.04067134e-04,
             2.10131853e-04, 5.68810833e-07],
            [9.99897242e-01, 6.31436847e-10, 1.09615639e-05, 2.06993377e-07,
             1.71082618e-07, 6.76040190e-06, 7.78621907e-05, 8.66052858e-07,
             1.13851035e-08, 6.00349176e-06],
            [5.33038519e-06, 2.26009519e-07, 2.57485158e-06, 1.76769777e-06,
             9.76953208e-01, 7.02418720e-06, 7.11563698e-05, 9.11385869e-05,
             1.48836762e-06, 2.28660498e-02]], dtype=float32)>
```