In [1]:
```
import tensorflow as tf

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
```

In [2]:
```
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 [3]:
```
predictions = model(x_train[:1]).numpy()
predictions
```

```
WARNING:tensorflow:Layer flatten is casting an input tensor from dtype float6
4 to the layer's dtype of float32, which is new behavior in TensorFlow 2.  Th
e layer has dtype float32 because it's dtype defaults to floatx.

If you intended to run this layer in float32, you can safely ignore this warn
ing. If in doubt, this warning is likely only an issue if you are porting a T
ensorFlow 1.X model to TensorFlow 2.

To change all layers to have dtype float64 by default, call `tf.keras.backen
d.set_floatx('float64')`. To change just this layer, pass dtype='float64' to
the layer constructor. If you are the author of this layer, you can disable a
utocasting by passing autocast=False to the base Layer constructor.
```

Out[3]:
```
array([[-0.5520137 ,  0.19004199, -0.26991916,  0.00284022, -0.48906517,
         0.06171124, -0.48691612,  0.14503226,  0.00119551,  0.23573586]],
      dtype=float32)
```

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

Out[4]:
```
array([[-0.5520137 ,  0.19004199, -0.26991916,  0.00284022, -0.48906517,
         0.06171124, -0.48691612,  0.14503226,  0.00119551,  0.23573586]],
      dtype=float32)
```

In [5]:
```
loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
```

In [6]:
```
loss_fn(y_train[:1], predictions).numpy()
```

Out[6]:
```
2.1646256
```

In [7]:
```
model.compile(optimizer='adam',
              loss=loss_fn,
              metrics=['accuracy'])
```

In [8]:
```
model.fit(x_train, y_train, epochs=5)
```

```
Train on 60000 samples
Epoch 1/5
60000/60000 [==============================] - 3s 58us/sample - loss: 0.2977
- accuracy: 0.9147
Epoch 2/5
60000/60000 [==============================] - 4s 65us/sample - loss: 0.1443
- accuracy: 0.9571
Epoch 3/5
60000/60000 [==============================] - 3s 52us/sample - loss: 0.1093
- accuracy: 0.9674
Epoch 4/5
60000/60000 [==============================] - 3s 52us/sample - loss: 0.0877
- accuracy: 0.9727
Epoch 5/5
60000/60000 [==============================] - 3s 52us/sample - loss: 0.0765
- accuracy: 0.9759
```

Out[8]:
```
<tensorflow.python.keras.callbacks.History at 0x7ffa928da8d0>
```

In [9]:
```
model.evaluate(x_test,  y_test, verbose=2)
```

```
10000/10000 - 0s - loss: 0.0802 - accuracy: 0.9761
```

Out[9]:
```
[0.0802184832977131, 0.9761]
```

In [10]:
```
probability_model = tf.keras.Sequential([
  model,
  tf.keras.layers.Softmax()
])
```

In [11]:
```
probability_model(x_test[:5])
```

Out[11]:
```
<tf.Tensor: shape=(5, 10), dtype=float32, numpy=
array([[9.5849302e-09, 3.9378549e-11, 6.5275117e-08, 1.3703826e-06,
        1.1944210e-12, 4.1083002e-08, 5.5548193e-16, 9.9999809e-01,
        2.6008884e-10, 4.4550396e-07],
       [1.8828779e-07, 1.2148355e-04, 9.9987233e-01, 5.6443155e-06,
        1.0702423e-14, 4.3269903e-08, 2.1492182e-08, 2.9451448e-12,
        1.8064995e-07, 5.1426874e-14],
       [3.7898255e-07, 9.9705791e-01, 1.7463682e-04, 5.5254237e-05,
        2.8641391e-05, 1.7255533e-06, 5.2309838e-06, 2.2933315e-03,
        3.8145462e-04, 1.5346056e-06],
       [9.9998689e-01, 1.3248206e-11, 5.9398280e-06, 8.1381146e-10,
        1.4066102e-08, 1.2968728e-07, 2.7897138e-06, 3.2611638e-06,
        5.4506916e-10, 1.0591963e-06],
       [7.1158547e-07, 4.4313422e-10, 1.0565856e-05, 5.9094400e-08,
        9.9140549e-01, 1.8611076e-06, 2.4394640e-07, 3.7752758e-05,
        1.4318758e-07, 8.5431812e-03]], dtype=float32)>
```

In [ ]:
```
```