

In [1]:

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist
```

In [2]:

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

In [3]:

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

In [4]:

```
x_train.shape
```

Out[4]:

```
(60000, 28, 28)
```

In [5]:

```
x_test.shape
```

Out[5]:

```
(10000, 28, 28)
```

In [6]:

```
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Flatten(input_shape=(28, 28)))
model.add(tf.keras.layers.Dense(256, activation=tf.nn.relu))
model.add(tf.keras.layers.Dense(10, activation=tf.nn.softmax))
```

In [7]:

```
model.summary()
```

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 784)	0
dense (Dense)	(None, 256)	200960
dense_1 (Dense)	(None, 10)	2570
Total params: 203,530		
Trainable params: 203,530		
Non-trainable params: 0		

In [8]:

```
784 * 10 + 10
```

Out[8]:

7850

In [9]:

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

In [10]:

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

```
Epoch 1/5  
60000/60000 [=====] - 5s 84us/step - loss: 0.  
2263 - acc: 0.9349  
Epoch 2/5  
60000/60000 [=====] - 5s 76us/step - loss: 0.  
0939 - acc: 0.9715  
Epoch 3/5  
60000/60000 [=====] - 5s 77us/step - loss: 0.  
0615 - acc: 0.9810  
Epoch 4/5  
60000/60000 [=====] - 5s 80us/step - loss: 0.  
0449 - acc: 0.9859  
Epoch 5/5  
60000/60000 [=====] - 5s 82us/step - loss: 0.  
0338 - acc: 0.9894
```

Out[10]:

```
<tensorflow.python.keras.callbacks.History at 0x12ceef4a8>
```

Real World Challenge: Large difference between training and testing set accuracy

Test accuracy

In [11]:

```
model.evaluate(x_test, y_test)
```

```
10000/10000 [=====] - 0s 38us/step
```

Out[11]:

```
[0.07265878109126352, 0.9805]
```

In [12]:

```
model.evaluate(x_test[:2], y_test[:2])
```

2/2 [=====] - 0s 536us/step

Out[12]:

```
[1.12652123789303e-05, 1.0]
```

Training accuracy

In [13]:

```
model.evaluate(x_train, y_train)
```

60000/60000 [=====] - 2s 32us/step

Out[13]:

```
[0.024180645394007053, 0.9924666666666667]
```