In [19]:

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
import tensorflow_datasets as tfds
```

In [20]:

```
mnist_dataset, mnist_info = tfds.load(name='mnist', with_info=True, as_supervised=Tr

mnist_train, mnist_test = mnist_dataset['train'], mnist_dataset['test']

num_validation_samples = 0.1 * mnist_info.splits['train'].num_examples
num_validation_samples = tf.cast(num_validation_samples,dtype=tf.int64)

num_test_samples = mnist_info.splits['test'].num_examples
num_test_samples = tf.cast(num_test_samples,dtype=tf.int64)
```

In [21]:

```
def scale(image,label):
    image = tf.cast(image,tf.float32)
    image /= 255.
    return image, label

scaled_train_and_validation_data = mnist_train.map(scale)
test_data = mnist_test.map(scale)
```

In [31]:

```
shuffled_train_and_validation_data = scaled_train_and_validation_data.shuffle(buffer

BUFFER_SIZE = 10000

shuffled_train_and_validation_data = scaled_train_and_validation_data.shuffle(BUFFER

validation_data = shuffled_train_and_validation_data.take(num_validation_samples)
train_data = shuffled_train_and_validation_data.skip(num_validation_samples)


BATCH_SIZE = 100
train_data = train_data.batch(BATCH_SIZE)
validation_data = validation_data.batch(num_validation_samples)

test_data = test_data.batch(num_test_samples)

validation_inputs, validation_targets = next(iter(validation_data))
```

In [32]:

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(64, (3,3), activation='elu', input_shape=(28, 28, 1)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='elu'),
  tf.keras.layers.MaxPooling2D(2,2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(100, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

In [33]:

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['ac
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d (Conv2D)              (None, 26, 26, 64)        640       
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 64)        0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 11, 11, 64)        36928     
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 64)          0         
_________________________________________________________________
flatten (Flatten)            (None, 1600)              0         
_________________________________________________________________
dense (Dense)                (None, 100)               160100    
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1010      
=================================================================
Total params: 198,678
Trainable params: 198,678
Non-trainable params: 0
_________________________________________________________________
```

In [34]:

```
NUM_EPOCHS = 10

early_stoping = tf.keras.callbacks.EarlyStopping()

model.fit(train_data,
          epochs = NUM_EPOCHS,
          callbacks = [early_stoping],
          validation_data=(validation_inputs,validation_targets),
          verbose=2)
```

```
Epoch 1/10
540/540 - 49s - loss: 0.1795 - accuracy: 0.9464 - val_loss: 0.0663 - v
al_accuracy: 0.9793
Epoch 2/10
540/540 - 44s - loss: 0.0496 - accuracy: 0.9847 - val_loss: 0.0467 - v
al_accuracy: 0.9845
Epoch 3/10
540/540 - 43s - loss: 0.0335 - accuracy: 0.9895 - val_loss: 0.0352 - v
al_accuracy: 0.9897
Epoch 4/10
540/540 - 45s - loss: 0.0258 - accuracy: 0.9916 - val_loss: 0.0303 - v
al_accuracy: 0.9915
Epoch 5/10
540/540 - 44s - loss: 0.0186 - accuracy: 0.9940 - val_loss: 0.0221 - v
al_accuracy: 0.9928
Epoch 6/10
540/540 - 45s - loss: 0.0154 - accuracy: 0.9949 - val_loss: 0.0190 - v
al_accuracy: 0.9945
Epoch 7/10
540/540 - 45s - loss: 0.0127 - accuracy: 0.9958 - val_loss: 0.0186 - v
al_accuracy: 0.9937
Epoch 8/10
540/540 - 46s - loss: 0.0119 - accuracy: 0.9962 - val_loss: 0.0154 - v
al_accuracy: 0.9948
Epoch 9/10
540/540 - 47s - loss: 0.0099 - accuracy: 0.9966 - val_loss: 0.0133 - v
al_accuracy: 0.9957
Epoch 10/10
540/540 - 54s - loss: 0.0085 - accuracy: 0.9971 - val_loss: 0.0108 - v
al_accuracy: 0.9972
```

Out[34]:

```
<tensorflow.python.keras.callbacks.History at 0x13ffa5880>
```

In [35]:

```
test_data, test_accuracy = model.evaluate(test_data)
print(test_data, test_accuracy)
```

```
1/1 [==============================] - 0s 2ms/step - loss: 0.0388 - ac
curacy: 0.9906
0.03881040801763535 0.9905999898910522
```