Exercise 3

In the videos you looked at how you would improve Fashion MNIST using Convolutions. For your exercise see if you can improve MNIST to 99.8% accuracy or more using only a single convolutional layer and a single MaxPooling 2D. You should stop training once the accuracy goes above this amount. It should happen in less than 20 epochs, so it's ok to hard code the number of epochs for training, but your training must end once it hits the above metric. If it doesn't, then you'll need to redesign your layers.

I've started the code for you -- you need to finish it!

When 99.8% accuracy has been hit, you should print out the string "Reached 99.8% accuracy so cancelling training!"

In [16]:

```
import tensorflow as tf
from os import path, getcwd, chdir

# DO NOT CHANGE THE LINE BELOW. If you are developing in a local
# environment, then grab mnist.npz from the Coursera Jupyter Notebook
# and place it inside a local folder and edit the path to that location
path = f"{getcwd()}/../tmp2/mnist.npz"
```

In [17]:

```
config = tf.ConfigProto()
config.gpu_options.allow_growth = True
sess = tf.Session(config=config)
```

In [20]:

```
# GRADED FUNCTION: train_mnist_conv
def train_mnist_conv():
    # Please write your code only where you are indicated.
    # please do not remove model fitting inline comments.

    # YOUR CODE STARTS HERE
    class myCallback(tf.keras.callbacks.Callback):
        def on_epoch_end(self, epoch, logs={}):
            if(logs.get('acc')>0.998):
                print("\nReached 99.8% accuracy so cancelling training!")
                self.model.stop_training = True
    # YOUR CODE ENDS HERE

    mnist = tf.keras.datasets.mnist
    (training_images, training_labels), (test_images, test_labels) = mnist.load_dat
    training_images=training_images.reshape(60000, 28, 28, 1)
    training_images=training_images / 255.0
    test_images = test_images.reshape(10000, 28, 28, 1)
    test_images=test_images/255.0

    callbacks = myCallback()
    # YOUR CODE ENDS HERE

    model = tf.keras.models.Sequential([
            # YOUR CODE STARTS HERE
            tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28, 2
            tf.keras.layers.MaxPooling2D(2, 2),
            tf.keras.layers.Flatten(),
            tf.keras.layers.Dense(128, activation='relu'),
            tf.keras.layers.Dense(10, activation='softmax')
            # YOUR CODE ENDS HERE
    ])

    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics
    # model fitting
    history = model.fit(
        # YOUR CODE STARTS HERE
        training_images, training_labels, epochs=20, callbacks=[callbacks]
        # YOUR CODE ENDS HERE
    )
    # model fitting
    return history.epoch, history.history['acc'][-1]
```

In [21]:

```
_, _ = train_mnist_conv()
```

```
Epoch 1/20
60000/60000 [==============================] - 21s 358us/sample - los
s: 0.1384 - acc: 0.9591
Epoch 2/20
60000/60000 [==============================] - 21s 352us/sample - los
s: 0.0476 - acc: 0.9858
Epoch 3/20
60000/60000 [==============================] - 21s 356us/sample - los
s: 0.0297 - acc: 0.9908
Epoch 4/20
60000/60000 [==============================] - 21s 355us/sample - los
s: 0.0190 - acc: 0.9941
Epoch 5/20
60000/60000 [==============================] - 21s 355us/sample - los
s: 0.0121 - acc: 0.9958
Epoch 6/20
60000/60000 [==============================] - 21s 355us/sample - los
s: 0.0098 - acc: 0.9967
Epoch 7/20
60000/60000 [==============================] - 21s 357us/sample - los
s: 0.0077 - acc: 0.9976
Epoch 8/20
59712/60000 [============================>.] - ETA: 0s - loss: 0.0049
- acc: 0.9982
Reached 99.8% accuracy so cancelling training!
60000/60000 [==============================] - 21s 356us/sample - los
s: 0.0049 - acc: 0.9983
```

In [ ]:

```
# Now click the 'Submit Assignment' button above.
# Once that is complete, please run the following two cells to save your work and c
```

In [ ]:

```
%%javascript
<!-- Save the notebook -->
IPython.notebook.save_checkpoint();
```

In [ ]:

```
%%javascript
<!-- Shutdown and close the notebook -->
window.onbeforeunload = null
window.close();
IPython.notebook.session.delete();
```