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 [1]:

```
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 [2]:

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

In [4]:

```
# 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.

    class myCallback(tf.keras.callbacks.Callback):
        def on_epoch_end(self, epoch, logs={}):
            if(logs.get('acc') >= 0.999):
                print("\nReached 95% accuracy so cancelling training!")
                self.model.stop_training = True

    callbacks = myCallback()
    mnist = tf.keras.datasets.mnist
    (training_images, training_labels), (test_images, test_labels) = mnist.load_data(pa
th=path)
    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

    model = tf.keras.models.Sequential([
            tf.keras.layers.Conv2D(64, (3, 3), activation='relu', input_shape=(28, 28,
1)),
            tf.keras.layers.MaxPooling2D(2, 2),
            tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
            tf.keras.layers.MaxPooling2D(2, 2),
            tf.keras.layers.Flatten(),
            tf.keras.layers.Dense(1024, activation='relu'),
            tf.keras.layers.Dense(10, activation='softmax')
    ])

    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['a
ccuracy'])
    # model fitting
    history = model.fit(
        training_images, training_labels, epochs=10, callbacks=[callbacks]
    )
    # model fitting
    return history.epoch, history.history['acc'][-1]
```

In [5]:

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

```
WARNING: Logging before flag parsing goes to stderr.
W0917 14:25:33.687735 139906063697728 deprecation.py:506] From /usr/local/
lib/python3.6/dist-packages/tensorflow/python/ops/init_ops.py:1251: callin
g VarianceScaling.__init__ (from tensorflow.python.ops.init_ops) with dtyp
e is deprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to
the constructor

Epoch 1/10
60000/60000 [==============================] - 17s 288us/sample - loss: 0.
1033 - acc: 0.9674
Epoch 2/10
60000/60000 [==============================] - 14s 228us/sample - loss: 0.
0367 - acc: 0.9884
Epoch 3/10
60000/60000 [==============================] - 14s 228us/sample - loss: 0.
0250 - acc: 0.9918
Epoch 4/10
60000/60000 [==============================] - 12s 204us/sample - loss: 0.
0167 - acc: 0.9947
Epoch 5/10
60000/60000 [==============================] - 12s 205us/sample - loss: 0.
0133 - acc: 0.9958
Epoch 6/10
60000/60000 [==============================] - 13s 225us/sample - loss: 0.
0115 - acc: 0.9962
Epoch 7/10
60000/60000 [==============================] - 14s 226us/sample - loss: 0.
0088 - acc: 0.9971
Epoch 8/10
60000/60000 [==============================] - 13s 218us/sample - loss: 0.
0064 - acc: 0.9980
Epoch 9/10
60000/60000 [==============================] - 14s 238us/sample - loss: 0.
0079 - acc: 0.9975
Epoch 10/10
60000/60000 [==============================] - 15s 250us/sample - loss: 0.
0055 - 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 close
the notebook
```

In [ ]:

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

In [ ]:

```
%%javascript
IPython.notebook.session.delete();
window.onbeforeunload = null
setTimeout(function() { window.close(); }, 1000);
```