

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 [3]:

```
# 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') >= .998):
                print("Reached 99.8% accuracy so cancelling training!")
                self.model.stop_training = True

    callbacks = myCallback()
    # YOUR CODE ENDS HERE

    mnist = tf.keras.datasets.mnist
    (training_images, training_labels), (test_images, test_labels) = mnist.load_data(path=path)
    # YOUR CODE STARTS HERE
    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
    # YOUR CODE ENDS HERE
    model = tf.keras.models.Sequential([
        # YOUR CODE STARTS HERE
        tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(28,28,1)),
        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=['accuracy'])
    # 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 [4]:

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

WARNING: Logging before flag parsing goes to stderr.
W1005 23:30:42.830773 140330166441792 deprecation.py:506] From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/init_ops.py:1251: calling VarianceScaling.__init__ (from tensorflow.python.ops.init_ops) with dtype 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 1/20
60000/60000 [=====] - 15s 246us/sample - loss: 0.1344 - acc: 0.9593
Epoch 2/20
60000/60000 [=====] - 12s 193us/sample - loss: 0.0460 - acc: 0.9858
Epoch 3/20
60000/60000 [=====] - 11s 189us/sample - loss: 0.0278 - acc: 0.9912
Epoch 4/20
60000/60000 [=====] - 11s 190us/sample - loss: 0.0173 - acc: 0.9945
Epoch 5/20
60000/60000 [=====] - 11s 188us/sample - loss: 0.0122 - acc: 0.9960
Epoch 6/20
60000/60000 [=====] - 11s 190us/sample - loss: 0.0094 - acc: 0.9970
Epoch 7/20
60000/60000 [=====] - 11s 186us/sample - loss: 0.0067 - acc: 0.9979
Epoch 8/20
59520/60000 [=====>.] - ETA: 0s - loss: 0.0049 - acc: 0.9984Reached 99.8% a
ccuracy so cancelling training!
60000/60000 [=====] - 12s 198us/sample - loss: 0.0049 - acc: 0.9984
```

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 notebok
ok
```

In []:

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

In []:

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