In [1]:
```
'''Trains a simple deep NN on the MNIST dataset.

Gets to 98.40% test accuracy after 20 epochs
(there is *a lot* of margin for parameter tuning).
2 seconds per epoch on a K520 GPU.
'''

from tensorflow import keras
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import RMSprop

batch_size = 128
num_classes = 10
epochs = 20

# the data, split between train and test sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()

x_train = x_train.reshape(60000, 784)
x_test = x_test.reshape(10000, 784)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')

# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

model = Sequential()
model.add(Dense(512, activation='relu', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))

model.summary()

model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(),
              metrics=['accuracy'])

history = model.fit(x_train, y_train,
                    batch_size=batch_size,
                    epochs=epochs,
                    verbose=1,
                    validation_data=(x_test, y_test))
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-data
sets/mnist.npz
11493376/11490434 [==============================] - 2s 0us/step
60000 train samples
10000 test samples
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 512)               401920
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                5130
=================================================================
Total params: 669,706
Trainable params: 669,706
Non-trainable params: 0
_________________________________________________________________
Epoch 1/20
469/469 [==============================] - 5s 10ms/step - loss: 0.2441 - accu
racy: 0.9248 - val_loss: 0.1169 - val_accuracy: 0.9643
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1021 - accu
racy: 0.9683 - val_loss: 0.0749 - val_accuracy: 0.9777
Epoch 3/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0763 - accu
racy: 0.9766 - val_loss: 0.0758 - val_accuracy: 0.9774
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0612 - accu
racy: 0.9824 - val_loss: 0.0820 - val_accuracy: 0.9767
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0507 - accu
racy: 0.9846 - val_loss: 0.0758 - val_accuracy: 0.9813
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0440 - accu
racy: 0.9867 - val_loss: 0.0755 - val_accuracy: 0.9819
Epoch 7/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0389 - accu
racy: 0.9885 - val_loss: 0.0780 - val_accuracy: 0.9824
Epoch 8/20
469/469 [==============================] - 4s 10ms/step - loss: 0.0347 - accu
racy: 0.9899 - val_loss: 0.0749 - val_accuracy: 0.9817
Epoch 9/20
469/469 [==============================] - 4s 10ms/step - loss: 0.0318 - accu
racy: 0.9911 - val_loss: 0.0903 - val_accuracy: 0.9799
Epoch 10/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0299 - accur
acy: 0.9914 - val_loss: 0.0809 - val_accuracy: 0.9846
Epoch 11/20
469/469 [==============================] - 4s 10ms/step - loss: 0.0275 - accu
racy: 0.9918 - val_loss: 0.0920 - val_accuracy: 0.9831
Epoch 12/20
```

```
469/469 [==============================] - 4s 9ms/step - loss: 0.0244 - accur
acy: 0.9929 - val_loss: 0.0916 - val_accuracy: 0.9831
Epoch 13/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0242 - accu
racy: 0.9930 - val_loss: 0.0997 - val_accuracy: 0.9839
Epoch 14/20
469/469 [==============================] - 4s 10ms/step - loss: 0.0223 - accu
racy: 0.9939 - val_loss: 0.1106 - val_accuracy: 0.9823
Epoch 15/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0229 - accur
acy: 0.9936 - val_loss: 0.1082 - val_accuracy: 0.9832
Epoch 16/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0206 - accur
acy: 0.9941 - val_loss: 0.1120 - val_accuracy: 0.9836
Epoch 17/20
469/469 [==============================] - 4s 10ms/step - loss: 0.0196 - accu
racy: 0.9949 - val_loss: 0.1067 - val_accuracy: 0.9834
Epoch 18/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0180 - accur
acy: 0.9952 - val_loss: 0.1121 - val_accuracy: 0.9841
Epoch 19/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0186 - accur
acy: 0.9955 - val_loss: 0.1160 - val_accuracy: 0.9833
Epoch 20/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0172 - accur
acy: 0.9956 - val_loss: 0.1231 - val_accuracy: 0.9845
Test loss: 0.12314877659082413
Test accuracy: 0.984499990940094
```