NOpitz_DSC650_Week1_Assignment1_Tensorflow

March 21, 2021

```
[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))
```

```python
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-datasets/mnist.npz
11493376/11490434 [==============================] - 1s 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 [==============================] - 6s 12ms/step - loss: 0.4352 -
accuracy: 0.8645 - val_loss: 0.1149 - val_accuracy: 0.9646
Epoch 2/20
469/469 [==============================] - 5s 10ms/step - loss: 0.1029 -
accuracy: 0.9683 - val_loss: 0.0947 - val_accuracy: 0.9714
Epoch 3/20
```

```
469/469 [==============================] - 5s 10ms/step - loss: 0.0713 -
accuracy: 0.9786 - val_loss: 0.0712 - val_accuracy: 0.9790
Epoch 4/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0591 -
accuracy: 0.9815 - val_loss: 0.0817 - val_accuracy: 0.9786
Epoch 5/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0448 -
accuracy: 0.9863 - val_loss: 0.0714 - val_accuracy: 0.9811
Epoch 6/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0423 -
accuracy: 0.9867 - val_loss: 0.0761 - val_accuracy: 0.9801
Epoch 7/20
469/469 [==============================] - 4s 10ms/step - loss: 0.0323 -
accuracy: 0.9901 - val_loss: 0.0805 - val_accuracy: 0.9807
Epoch 8/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0326 -
accuracy: 0.9898 - val_loss: 0.0781 - val_accuracy: 0.9830
Epoch 9/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0261 -
accuracy: 0.9918 - val_loss: 0.0863 - val_accuracy: 0.9807
Epoch 10/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0278 -
accuracy: 0.9917 - val_loss: 0.0868 - val_accuracy: 0.9819
Epoch 11/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0246 -
accuracy: 0.9929 - val_loss: 0.0993 - val_accuracy: 0.9819
Epoch 12/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0259 -
accuracy: 0.9926 - val_loss: 0.0915 - val_accuracy: 0.9825
Epoch 13/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0207 -
accuracy: 0.9939 - val_loss: 0.1111 - val_accuracy: 0.9816
Epoch 14/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0197 -
accuracy: 0.9941 - val_loss: 0.1044 - val_accuracy: 0.9830
Epoch 15/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0188 -
accuracy: 0.9944 - val_loss: 0.1143 - val_accuracy: 0.9836
Epoch 16/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0191 -
accuracy: 0.9951 - val_loss: 0.1157 - val_accuracy: 0.9834
Epoch 17/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0168 -
accuracy: 0.9951 - val_loss: 0.1072 - val_accuracy: 0.9842
Epoch 18/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0159 -
accuracy: 0.9950 - val_loss: 0.1252 - val_accuracy: 0.9843
Epoch 19/20
```

```
469/469 [==============================] - 4s 9ms/step - loss: 0.0172 -
accuracy: 0.9953 - val_loss: 0.1088 - val_accuracy: 0.9854
Epoch 20/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0132 -
accuracy: 0.9961 - val_loss: 0.1198 - val_accuracy: 0.9836
Test loss: 0.11977902799844742
Test accuracy: 0.9836000204086304
```