# kreas-mnist-py

December 6, 2020

[1]:

```
# Name: Pritam shrestha
# Assignment No:1.1
# Format:Pdf
```

[4]:

```
'''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])
```

```
60000 train samples
10000 test samples
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_3 (Dense)              (None, 512)               401920
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_4 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_5 (Dense)              (None, 10)                5130
=================================================================
Total params: 669,706
Trainable params: 669,706
Non-trainable params: 0
_________________________________________________________________
Epoch 1/20
469/469 [==============================] - 4s 10ms/step - loss: 0.2464 -
accuracy: 0.9237 - val_loss: 0.1019 - val_accuracy: 0.9687
Epoch 2/20
469/469 [==============================] - 4s 9ms/step - loss: 0.1013 -
accuracy: 0.9693 - val_loss: 0.0837 - val_accuracy: 0.9756
```

```
Epoch 3/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0739 -
accuracy: 0.9774 - val_loss: 0.0820 - val_accuracy: 0.9753
Epoch 4/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0588 -
accuracy: 0.9824 - val_loss: 0.0991 - val_accuracy: 0.9756
Epoch 5/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0503 -
accuracy: 0.9849 - val_loss: 0.0694 - val_accuracy: 0.9823
Epoch 6/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0452 -
accuracy: 0.9864 - val_loss: 0.0711 - val_accuracy: 0.9823
Epoch 7/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0374 -
accuracy: 0.9890 - val_loss: 0.0741 - val_accuracy: 0.9815
Epoch 8/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0336 -
accuracy: 0.9899 - val_loss: 0.0943 - val_accuracy: 0.9792
Epoch 9/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0312 -
accuracy: 0.9908 - val_loss: 0.0765 - val_accuracy: 0.9848
Epoch 10/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0296 -
accuracy: 0.9914 - val_loss: 0.0814 - val_accuracy: 0.9825
Epoch 11/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0255 -
accuracy: 0.9924 - val_loss: 0.0965 - val_accuracy: 0.9827
Epoch 12/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0245 -
accuracy: 0.9930 - val_loss: 0.0863 - val_accuracy: 0.9847
Epoch 13/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0221 -
accuracy: 0.9937 - val_loss: 0.1004 - val_accuracy: 0.9821
Epoch 14/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0224 -
accuracy: 0.9933 - val_loss: 0.1039 - val_accuracy: 0.9825
Epoch 15/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0226 -
accuracy: 0.9938 - val_loss: 0.1125 - val_accuracy: 0.9828
Epoch 16/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0202 -
accuracy: 0.9947 - val_loss: 0.1220 - val_accuracy: 0.9825
Epoch 17/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0195 -
accuracy: 0.9946 - val_loss: 0.1125 - val_accuracy: 0.9831
Epoch 18/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0194 -
accuracy: 0.9947 - val_loss: 0.1291 - val_accuracy: 0.9829
```

```
Epoch 19/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0186 -
accuracy: 0.9952 - val_loss: 0.1176 - val_accuracy: 0.9835
Epoch 20/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0183 -
accuracy: 0.9951 - val_loss: 0.1178 - val_accuracy: 0.9829
Test loss: 0.11784850805997849
Test accuracy: 0.9829000234603882
```

[ ]:

```
# Thanks!!!
```