

```

from __future__ import print_function
import numpy as np
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers.core import Dense, Activation
#from keras.optimizers import SGD
from tensorflow.keras.optimizers import SGD

from keras.utils import np_utils

np.random.seed(1671) # for reproducibility

# network and training
NB_EPOCH = 200
BATCH_SIZE = 128
VERBOSE = 1
NB_CLASSES = 10 # number of outputs = number of digits
OPTIMIZER = SGD() # SGD optimizer, explained later in this chapter
N_HIDDEN = 128
VALIDATION_SPLIT=0.2 # how much TRAIN is reserved for VALIDATION

# data: shuffled and split between train and test sets
#
(X_train, y_train), (X_test, y_test) = mnist.load_data()

#X_train is 60000 rows of 28x28 values --> reshaped in 60000 x 784
RESHAPED = 784
#
X_train = X_train.reshape(60000, RESHAPED)
X_test = X_test.reshape(10000, RESHAPED)
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')

# normalize
#
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 = np_utils.to_categorical(y_train, NB_CLASSES)
Y_test = np_utils.to_categorical(y_test, NB_CLASSES)

# 10 outputs
# final stage is softmax

model = Sequential()
model.add(Dense(NB_CLASSES, input_shape=(RESHAPED,)))
model.add(Activation('softmax'))

model.summary()

model.compile(loss='categorical_crossentropy',

```

```

optimizer=OPTIMIZER,
metrics=['accuracy'])

history = model.fit(X_train, Y_train,
                    batch_size=BATCH_SIZE, epochs=NB_EPOCH,
                    verbose=VERBOSE, validation_split=VALIDATION_SPLIT)
score = model.evaluate(X_test, Y_test, verbose=VERBOSE)
print("\nTest score:", score[0])
print('Test accuracy:', score[1])

Epoch 15/200
375/375 [=====] - 1s 2ms/step - loss: 0.3945 - accu
Epoch 16/200
375/375 [=====] - 1s 2ms/step - loss: 0.3892 - accu
Epoch 17/200
375/375 [=====] - 1s 3ms/step - loss: 0.3845 - accu
Epoch 18/200
375/375 [=====] - 1s 3ms/step - loss: 0.3802 - accu
Epoch 19/200
375/375 [=====] - 1s 2ms/step - loss: 0.3762 - accu
Epoch 20/200
375/375 [=====] - 1s 2ms/step - loss: 0.3726 - accu
Epoch 21/200
375/375 [=====] - 1s 2ms/step - loss: 0.3693 - accu
Epoch 22/200
375/375 [=====] - 1s 2ms/step - loss: 0.3661 - accu
Epoch 23/200
375/375 [=====] - 1s 3ms/step - loss: 0.3632 - accu
Epoch 24/200
375/375 [=====] - 1s 2ms/step - loss: 0.3605 - accu
Epoch 25/200
375/375 [=====] - 1s 3ms/step - loss: 0.3579 - accu
Epoch 26/200
375/375 [=====] - 1s 2ms/step - loss: 0.3555 - accu
Epoch 27/200
375/375 [=====] - 1s 3ms/step - loss: 0.3532 - accu
Epoch 28/200
375/375 [=====] - 1s 3ms/step - loss: 0.3511 - accu
Epoch 29/200
375/375 [=====] - 1s 3ms/step - loss: 0.3490 - accu
Epoch 30/200
375/375 [=====] - 1s 3ms/step - loss: 0.3471 - accu
Epoch 31/200
375/375 [=====] - 1s 3ms/step - loss: 0.3453 - accu
Epoch 32/200
375/375 [=====] - 1s 3ms/step - loss: 0.3436 - accu
Epoch 33/200
375/375 [=====] - 1s 3ms/step - loss: 0.3419 - accu
Epoch 34/200
375/375 [=====] - 1s 3ms/step - loss: 0.3403 - accu
Epoch 35/200
375/375 [=====] - 1s 3ms/step - loss: 0.3388 - accu
Epoch 36/200
375/375 [=====] - 1s 3ms/step - loss: 0.3373 - accu
Epoch 37/200
375/375 [=====] - 1s 2ms/step - loss: 0.3359 - accu
Epoch 38/200
375/375 [=====] - 1s 3ms/step - loss: 0.3346 - accu
Epoch 39/200
375/375 [=====] - 1s 2ms/step - loss: 0.3332 - accu

```

```
375/375 [=====] - 1s 3ms/step - loss: 0.3333 - accu
Epoch 40/200
375/375 [=====] - 1s 3ms/step - loss: 0.3321 - accu
Epoch 41/200
375/375 [=====] - 1s 2ms/step - loss: 0.3308 - accu
Epoch 42/200
375/375 [=====] - 1s 3ms/step - loss: 0.3297 - accu
Epoch 43/200
375/375 [=====] - 1s 3ms/step - loss: 0.3285 - accu
```

Double-click (or enter) to edit

Double-click (or enter) to edit

✓ 3m 26s completed at 15:13

