MNIST

February 12, 2021

Training of a CNN with the MNIST dataset and visualization with tensorboard
based on the Tutorial in c't Python-Projekte and https://www.tensorflow.org/tensorboard/get_started

[ ]:

```
#preparation for tensorboard
%load_ext tensorboard
import tensorflow as tf
import datetime
```

[ ]:

```
#clear previous logs
import shutil
shutil.rmtree('./logs',ignore_errors=True)
```

[ ]:

```
#load mnist dataset
from tensorflow.keras.datasets import mnist
train_da, test_da = mnist.load_data()
x_train, y_train  = train_da
x_test, y_test    = test_da
```

[ ]:

```
#data preparation/ transformation
import tensorflow.keras.backend as K
from tensorflow.keras.utils import to_categorical
dat_form   = K.image_data_format()
rows, cols = 28, 28
train_size = x_train.shape[0]
test_size  = x_test.shape[0]
if dat_form == 'channels_first':
    x_train     = x_train.reshape(train_size, 1, rows, cols)
    x_test      = x_test.reshape(test_size, 1, rows, cols)
    input_shape = (1, rows, cols)
else:
    x_train     = x_train.reshape(train_size, rows, cols, 1)
    x_test      = x_test.reshape(test_size, rows, cols, 1)
    input_shape = (rows, cols, 1)
# norm data to float in range 0..1
x_train  = x_train.astype('float32')
x_test   = x_test.astype('float32')
x_train /= 255
x_test  /= 255
```

```
# conv class vecs to one hot vec
y_train  = to_categorical(y_train,10)
y_test   = to_categorical(y_test, 10)
```

[ ]:

```
#reduce training data for faster training
reduce   = 1   #set to 1 for training with reduced data set
reduceto = 100 #set to desired amount of data

if reduce==1:
    x_train = x_train[:reduceto]
    y_train = y_train[:reduceto]
```

[ ]:

```
#build network
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
model = Sequential()
model.add(Conv2D(32,
    kernel_size = (3, 3),
    activation  = 'relu',
    input_shape = input_shape))
model.add(Conv2D(64,
    kernel_size = (3, 3),
    activation  = 'relu'))
model.add(MaxPooling2D(
    pool_size   = (2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(200,
    activation  = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(10,
    activation  = 'softmax'))
```

[ ]:

```
from tensorflow.keras.losses import categorical_crossentropy
from tensorflow.keras.optimizers import Adam
model.compile(
loss      = categorical_crossentropy,
optimizer = Adam(),
metrics   = ['accuracy'])
```

[ ]:

```
#log for tensorboard
log_dir              = "logs/fit/" + datetime.datetime.now().
 ↪strftime("%Y%m%d-%H%M%S")
```

```
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir, histogram_freq=1)
```

[ ]:

```
#training (with callback added for tensorboard)
history = model.fit(x_train, y_train,
    batch_size      = 128,
    epochs          = 12,
    verbose         = 1,
    validation_data = (x_test, y_test),
    callbacks       = [tensorboard_callback])
```

[ ]:

```
#visualize training with tensorboard
%tensorboard --logdir logs/fit
```