baseline-models

March 3, 2022

[1]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras import models, layers
from fenpreprocessing import fen_to_array
from tensorflow.keras.callbacks import EarlyStopping
from data_generation import position_generator, fix_positions, PosGen

import datetime
%load_ext tensorboard
```

[2]:
```
tf.config.list_physical_devices()
```

[2]:
```
[PhysicalDevice(name='/physical_device:CPU:0', device_type='CPU'),
 PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

[3]:
```
# Setting paramaters on early stopping
earlystop = EarlyStopping(monitor='val_loss',
                          min_delta=0,
                          patience=20,
                          verbose=1,
                          mode='min',
                          restore_best_weights=True)

log_dir = "logs/fit/baseline" + datetime.datetime.now().
 ↪strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir,␣
 ↪histogram_freq=1)
```

[4]:
```
# Memory management, likely not necessary, but used as a safety as per the␣
 ↪documentation recommendations on using GPUS

gpus = tf.config.list_physical_devices('GPU')
if gpus:
  try:
    # Currently, memory growth needs to be the same across GPUs
    for gpu in gpus:
```

```
        tf.config.experimental.set_memory_growth(gpu, True)
    logical_gpus = tf.config.list_logical_devices('GPU')
    print(len(gpus), "Physical GPUs,", len(logical_gpus), "Logical GPUs")
  except RuntimeError as e:
    # Memory growth must be set before GPUs have been initialized
    print(e)
```

```
1 Physical GPUs, 1 Logical GPUs
```

[5]:
```
small_train = pd.read_csv('fens/converted_train_partial.csv')
small_val = pd.read_csv('fens/converted_val_partial.csv')
```

[6]:
```
print(f"Train: {small_train.shape}, Validation: {small_val.shape}")
```

```
Train: (338855, 2), Validation: (19123, 2)
```

[7]:
```
small_val.shape[0] / 32
```

[7]:
```
597.59375
```

[8]:
```
# train_gen = position_generator(small_train)
# val_gen = position_generator(small_val)
train_gen = PosGen(small_train, 'Position', 'Target')
val_gen = PosGen(small_val, 'Position', 'Target')
```

[10]:
```
baseline_model = models.Sequential()
baseline_model.add(layers.Conv2D(64, 4, padding='same', input_shape=(8,8,13),␣
 ↪activation='relu'))
baseline_model.add(layers.MaxPooling2D(2))
baseline_model.add(layers.Conv2D(32, 2, padding='same', activation='relu'))
baseline_model.add(layers.Flatten())
baseline_model.add(layers.Dense(64, activation='relu'))
baseline_model.add(layers.Dense(1, activation='sigmoid'))
baseline_model.compile(optimizer="adam", loss="binary_crossentropy",␣
 ↪metrics=['acc'])
baseline_model.summary()

# Fitting the model
baseline_history = baseline_model.fit(x=train_gen,
                    validation_data=val_gen,
                    # steps_per_epoch=100,
                    epochs=30,
                    callbacks=[earlystop, tensorboard_callback]
                    )
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
```

```
=================================================================
conv2d_2 (Conv2D)            (None, 8, 8, 64)          13376
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 4, 4, 64)          0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 4, 4, 32)          8224
_________________________________________________________________
flatten_1 (Flatten)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                32832
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 65
=================================================================
Total params: 54,497
Trainable params: 54,497
Non-trainable params: 0
_________________________________________________________________
Epoch 1/30
10590/10590 [==============================] - 148s 14ms/step - loss: 0.1348 -
acc: 0.9681 - val_loss: 0.1263 - val_acc: 0.9693
Epoch 2/30
10590/10590 [==============================] - 149s 14ms/step - loss: 0.1247 -
acc: 0.9684 - val_loss: 0.1221 - val_acc: 0.9691
Epoch 3/30
10590/10590 [==============================] - 150s 14ms/step - loss: 0.1192 -
acc: 0.9689 - val_loss: 0.1180 - val_acc: 0.9695
Epoch 4/30
10590/10590 [==============================] - 149s 14ms/step - loss: 0.1157 -
acc: 0.9695 - val_loss: 0.1189 - val_acc: 0.9696
Epoch 5/30
10590/10590 [==============================] - 140s 13ms/step - loss: 0.1129 -
acc: 0.9701 - val_loss: 0.1176 - val_acc: 0.9695
Epoch 6/30
10590/10590 [==============================] - 149s 14ms/step - loss: 0.1105 -
acc: 0.9706 - val_loss: 0.1176 - val_acc: 0.9697
Epoch 7/30
10590/10590 [==============================] - 155s 15ms/step - loss: 0.1079 -
acc: 0.9711 - val_loss: 0.1209 - val_acc: 0.9680
Epoch 8/30
10590/10590 [==============================] - 149s 14ms/step - loss: 0.1056 -
acc: 0.9717 - val_loss: 0.1239 - val_acc: 0.9672
Epoch 9/30
10590/10590 [==============================] - 147s 14ms/step - loss: 0.1033 -
acc: 0.9724 - val_loss: 0.1288 - val_acc: 0.9671
Epoch 10/30
10590/10590 [==============================] - 146s 14ms/step - loss: 0.1012 -
acc: 0.9726 - val_loss: 0.1245 - val_acc: 0.9679
Epoch 11/30
```

```
10590/10590 [==============================] - 148s 14ms/step - loss: 0.0992 -
acc: 0.9732 - val_loss: 0.1262 - val_acc: 0.9686
Epoch 12/30
10590/10590 [==============================] - 148s 14ms/step - loss: 0.0971 -
acc: 0.9737 - val_loss: 0.1322 - val_acc: 0.9655
Epoch 13/30
10590/10590 [==============================] - 149s 14ms/step - loss: 0.0953 -
acc: 0.9743 - val_loss: 0.1306 - val_acc: 0.9678
Epoch 14/30
10590/10590 [==============================] - 145s 14ms/step - loss: 0.0936 -
acc: 0.9746 - val_loss: 0.1326 - val_acc: 0.9652
Epoch 15/30
10590/10590 [==============================] - 147s 14ms/step - loss: 0.0918 -
acc: 0.9750 - val_loss: 0.1365 - val_acc: 0.9647
Epoch 16/30
10590/10590 [==============================] - 147s 14ms/step - loss: 0.0899 -
acc: 0.9754 - val_loss: 0.1345 - val_acc: 0.9657
Epoch 17/30
10590/10590 [==============================] - 146s 14ms/step - loss: 0.0880 -
acc: 0.9758 - val_loss: 0.1414 - val_acc: 0.9646
Epoch 18/30
10590/10590 [==============================] - 145s 14ms/step - loss: 0.0866 -
acc: 0.9763 - val_loss: 0.1427 - val_acc: 0.9645
Epoch 19/30
10590/10590 [==============================] - 145s 14ms/step - loss: 0.0852 -
acc: 0.9765 - val_loss: 0.1461 - val_acc: 0.9635
Epoch 20/30
10590/10590 [==============================] - 146s 14ms/step - loss: 0.0836 -
acc: 0.9771 - val_loss: 0.1484 - val_acc: 0.9647
Epoch 21/30
10590/10590 [==============================] - 145s 14ms/step - loss: 0.0824 -
acc: 0.9773 - val_loss: 0.1479 - val_acc: 0.9646
Epoch 22/30
10590/10590 [==============================] - 145s 14ms/step - loss: 0.0810 -
acc: 0.9775 - val_loss: 0.1626 - val_acc: 0.9614
Epoch 23/30
10590/10590 [==============================] - 145s 14ms/step - loss: 0.0799 -
acc: 0.9778 - val_loss: 0.1533 - val_acc: 0.9626
Epoch 24/30
10590/10590 [==============================] - 145s 14ms/step - loss: 0.0784 -
acc: 0.9783 - val_loss: 0.1617 - val_acc: 0.9607
Epoch 25/30
10590/10590 [==============================] - 143s 14ms/step - loss: 0.0771 -
acc: 0.9785 - val_loss: 0.1632 - val_acc: 0.9610
Restoring model weights from the end of the best epoch.
Epoch 00025: early stopping
```

[11]:
```
baseline_model.save('LongModel-PB')
```

```
INFO:tensorflow:Assets written to: LongModel-PB/assets
```

[12]:
```
baseline_model.save('LongModel.h5')
```

[ ]:
```

```