full-puzzle-model

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, ChessPositionGen

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=5,
                          verbose=1,
                          mode='min',
                          restore_best_weights=True)

log_dir = "logs/fit/full_puzzle" + 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]:
```
train = pd.read_csv('fens/train.csv')
val = pd.read_csv('fens/val.csv')
```

[6]:
```
train_gen = ChessPositionGen(train, batch_size=512)
val_gen = ChessPositionGen(val, batch_size=512)
```

[7]:
```
full_puzzle_model = models.Sequential()
full_puzzle_model.add(layers.Conv2D(64, 4, padding='same',␣
 ↪input_shape=(8,8,13), activation='relu'))
full_puzzle_model.add(layers.MaxPooling2D(2))
full_puzzle_model.add(layers.Conv2D(32, 2, padding='same', activation='relu'))
full_puzzle_model.add(layers.Flatten())
full_puzzle_model.add(layers.Dense(64, activation='relu'))
full_puzzle_model.add(layers.Dense(1, activation='sigmoid'))
full_puzzle_model.compile(optimizer="adam", loss="binary_crossentropy",␣
 ↪metrics=['acc'])
full_puzzle_model.summary()

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

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 8, 8, 64)          13376
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 4, 4, 64)          0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 4, 4, 32)          8224
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
```

```
dense (Dense)                (None, 64)                32832
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 65
=================================================================
Total params: 54,497
Trainable params: 54,497
Non-trainable params: 0
_________________________________________________________________
Epoch 1/30
```

```
---------------------------------------------------------------------------
InternalError                             Traceback (most recent call last)
<ipython-input-7-49893aa3d3dd> in <module>
     10
     11 # Fitting the model
---> 12 full_puzzle_history = full_puzzle_model.fit(x=train_gen,

     13                       validation_data=val_gen,
     14                       # steps_per_epoch=100,

~/anaconda3/envs/Better-learn/lib/python3.8/site-packages/tensorflow/python/
 ↪keras/engine/training.py in fit(self, x, y, batch_size, epochs, verbose,␣
 ↪callbacks, validation_split, validation_data, shuffle, class_weight,␣
 ↪sample_weight, initial_epoch, steps_per_epoch, validation_steps,␣
 ↪validation_batch_size, validation_freq, max_queue_size, workers,␣
 ↪use_multiprocessing)
   1181                 _r=1):
   1182               callbacks.on_train_batch_begin(step)
-> 1183               tmp_logs = self.train_function(iterator)
   1184               if data_handler.should_sync:
   1185                 context.async_wait()

~/anaconda3/envs/Better-learn/lib/python3.8/site-packages/tensorflow/python/
 ↪eager/def_function.py in __call__(self, *args, **kwds)
    887
    888       with OptionalXlaContext(self._jit_compile):
--> 889         result = self._call(*args, **kwds)
    890
    891       new_tracing_count = self.experimental_get_tracing_count()

~/anaconda3/envs/Better-learn/lib/python3.8/site-packages/tensorflow/python/
 ↪eager/def_function.py in _call(self, *args, **kwds)
    948         # Lifting succeeded, so variables are initialized and we can ru␣
 ↪the
    949         # stateless function.
--> 950         return self._stateless_fn(*args, **kwds)
    951     else:
    952       _, _, _, filtered_flat_args = \
```

```
~/anaconda3/envs/Better-learn/lib/python3.8/site-packages/tensorflow/python/
↪eager/function.py in __call__(self, *args, **kwargs)
   3021       (graph_function,
   3022        filtered_flat_args) = self._maybe_define_function(args, kwargs)
-> 3023     return graph_function._call_flat(
   3024         filtered_flat_args, captured_inputs=graph_function.
↪captured_inputs)  # pylint: disable=protected-access
   3025

~/anaconda3/envs/Better-learn/lib/python3.8/site-packages/tensorflow/python/
↪eager/function.py in _call_flat(self, args, captured_inputs,␣
↪cancellation_manager)
   1958         and executing_eagerly):
   1959       # No tape is watching; skip to running the function.
-> 1960       return self._build_call_outputs(self._inference_function.call(
   1961           ctx, args, cancellation_manager=cancellation_manager))
   1962     forward_backward = self._select_forward_and_backward_functions(

~/anaconda3/envs/Better-learn/lib/python3.8/site-packages/tensorflow/python/
↪eager/function.py in call(self, ctx, args, cancellation_manager)
    589       with _InterpolateFunctionError(self):
    590         if cancellation_manager is None:
--> 591           outputs = execute.execute(
    592               str(self.signature.name),
    593               num_outputs=self._num_outputs,

~/anaconda3/envs/Better-learn/lib/python3.8/site-packages/tensorflow/python/
↪eager/execute.py in quick_execute(op_name, num_outputs, inputs, attrs, ctx,␣
↪name)
     57   try:
     58     ctx.ensure_initialized()
---> 59     tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name,␣
↪op_name,
     60                                         inputs, attrs, num_outputs)
     61   except core._NotOkStatusException as e:

InternalError:  Blas xGEMM launch failed : a.shape=[1,2454,512], b.
↪shape=[1,512,64], m=2454, n=64, k=512
        [[node sequential/dense/MatMul (defined at␣
↪<ipython-input-7-49893aa3d3dd>:12) ]] [Op:__inference_train_function_831]

Function call stack:
train_function
```

[ ]:
```
full_puzzle_model.save('full-puzzle.h5')
```

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

[ ]:
```
# test = pd.read_csv('fens/test.csv')
# test_gen = ChessPositionGen(test)
```

[ ]:
```

```