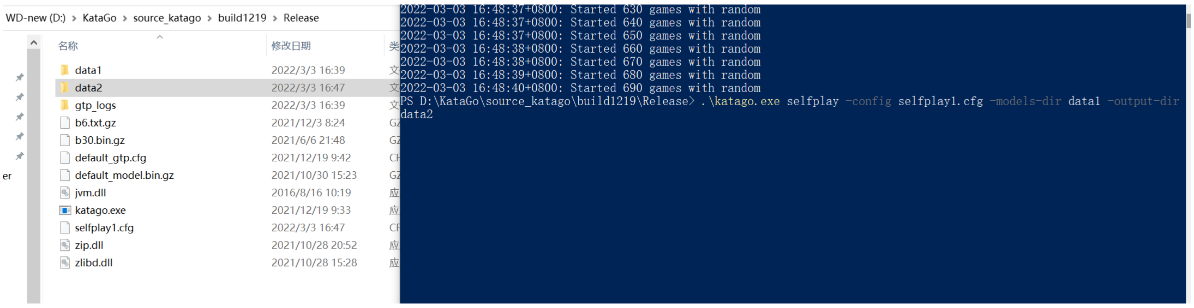
第一步

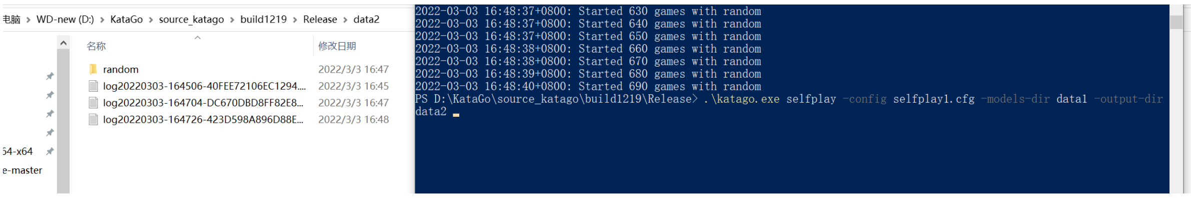
第一步应该是去生成game数据



WD-new (D:) › KataGo › source_katago › build1219 › Release

名称	修改日期
data1	2022/3/3 16:39
data2	2022/3/3 16:47
gtp_logs	2022/3/3 16:39
b6.txt.gz	2021/12/3 8:24
b30.bin.gz	2021/6/6 21:48
default_gtp.cfg	2021/12/19 9:42
default_model.bin.gz	2021/10/30 15:23
jvm.dll	2016/8/16 10:19
katago.exe	2021/12/19 9:33
selfplay1.cfg	2022/3/3 16:47
zip.dll	2021/10/28 20:52
zlibd.dll	2021/10/28 15:28

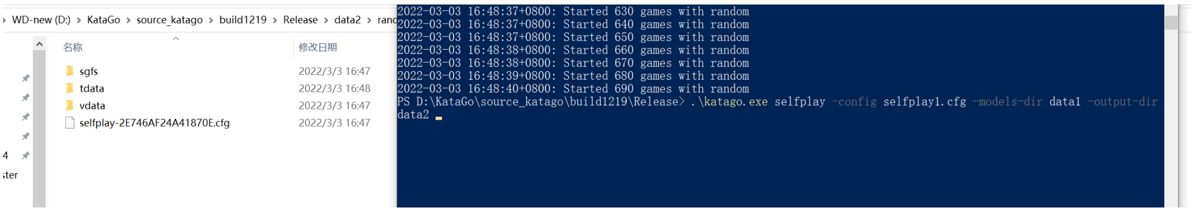
```
2022-03-03 16:48:37+0800: Started 630 games with random
2022-03-03 16:48:37+0800: Started 640 games with random
2022-03-03 16:48:37+0800: Started 650 games with random
2022-03-03 16:48:38+0800: Started 660 games with random
2022-03-03 16:48:38+0800: Started 670 games with random
2022-03-03 16:48:39+0800: Started 680 games with random
2022-03-03 16:48:40+0800: Started 690 games with random
PS D:\KataGo\source_katago\build1219\Release> .\katago.exe selfplay -config selfplay1.cfg -models-dir data1 -output-dir
data2
```



电脑 › WD-new (D:) › KataGo › source_katago › build1219 › Release › data2

名称	修改日期
random	2022/3/3 16:47
log20220303-164506-40FEE72106EC1294....	2022/3/3 16:45
log20220303-164704-DC670DBD8FF82E8...	2022/3/3 16:47
log20220303-164726-423D598A896D88E...	2022/3/3 16:48

```
2022-03-03 16:48:37+0800: Started 630 games with random
2022-03-03 16:48:37+0800: Started 640 games with random
2022-03-03 16:48:37+0800: Started 650 games with random
2022-03-03 16:48:38+0800: Started 660 games with random
2022-03-03 16:48:38+0800: Started 670 games with random
2022-03-03 16:48:39+0800: Started 680 games with random
2022-03-03 16:48:40+0800: Started 690 games with random
PS D:\KataGo\source_katago\build1219\Release> .\katago.exe selfplay -config selfplay1.cfg -models-dir data1 -output-dir
data2
```

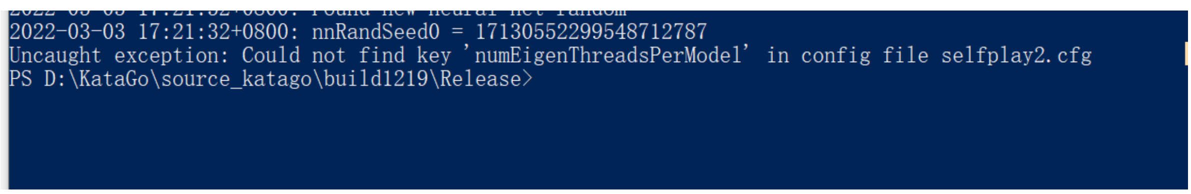


WD-new (D:) › KataGo › source_katago › build1219 › Release › data2 › ranc

名称	修改日期
sgfs	2022/3/3 16:47
tdata	2022/3/3 16:48
vdata	2022/3/3 16:47
selfplay-2E746AF24A41870E.cfg	2022/3/3 16:47

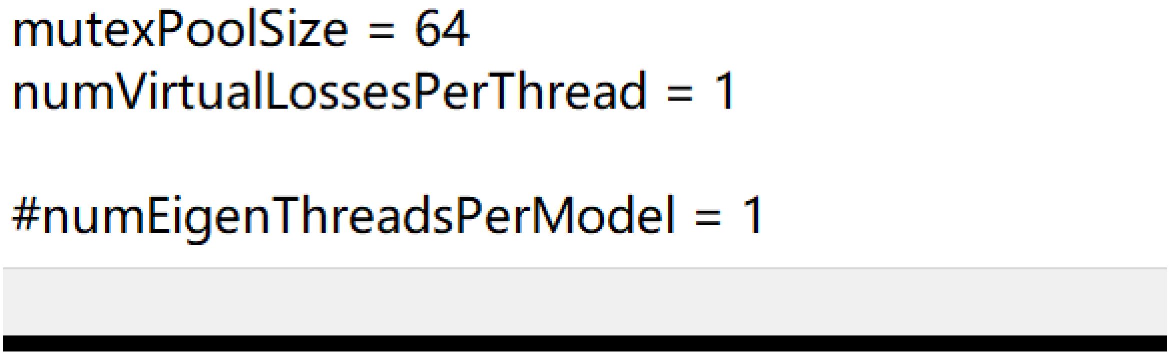
```
2022-03-03 16:48:37+0800: Started 630 games with random
2022-03-03 16:48:37+0800: Started 640 games with random
2022-03-03 16:48:37+0800: Started 650 games with random
2022-03-03 16:48:38+0800: Started 660 games with random
2022-03-03 16:48:38+0800: Started 670 games with random
2022-03-03 16:48:39+0800: Started 680 games with random
2022-03-03 16:48:40+0800: Started 690 games with random
PS D:\KataGo\source_katago\build1219\Release> .\katago.exe selfplay -config selfplay1.cfg -models-dir data1 -output-dir
data2
```

注意



```
2022-03-03 17:21:32+0800: Found new neural net random
2022-03-03 17:21:32+0800: nnRandSeed0 = 17130552299548712787
Uncaught exception: Could not find key 'numEigenThreadsPerModel' in config file selfplay2.cfg
PS D:\KataGo\source_katago\build1219\Release>
```

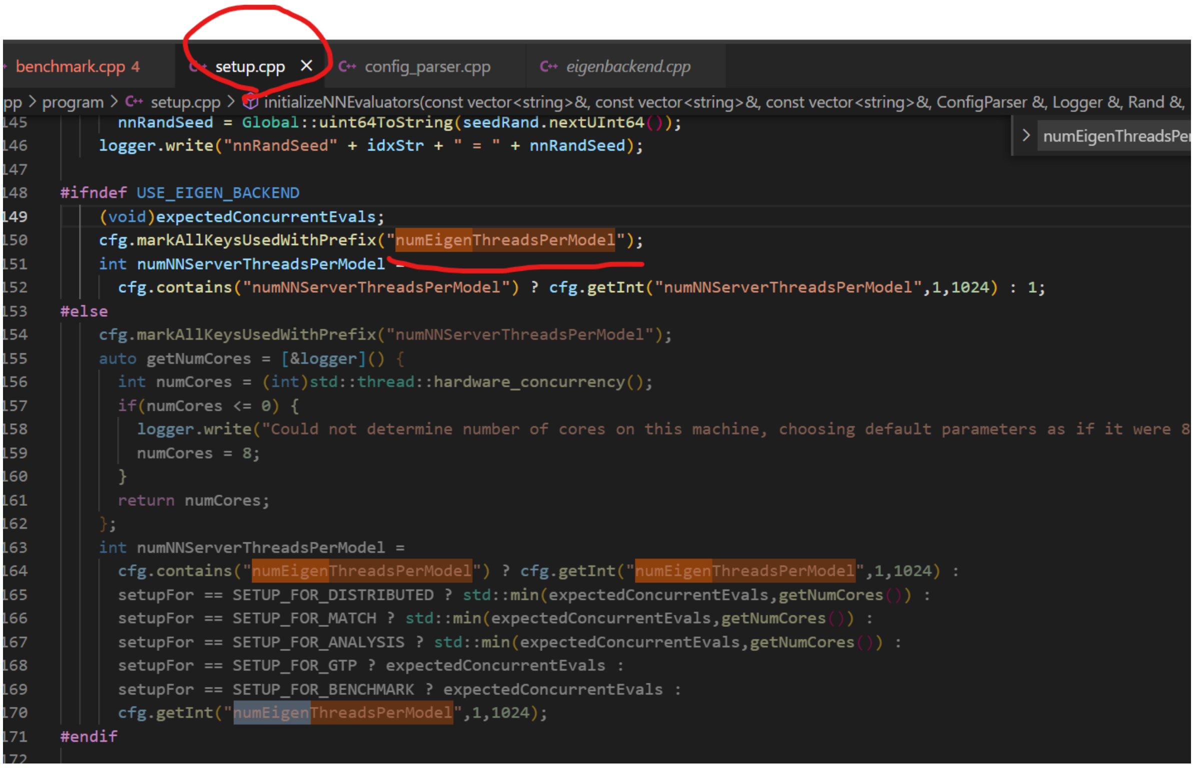
numEigenThreadsPerModel 没有就要自己添加



```
mutexPoolSize = 64
numVirtualLossesPerThread = 1

#numEigenThreadsPerModel = 1
```

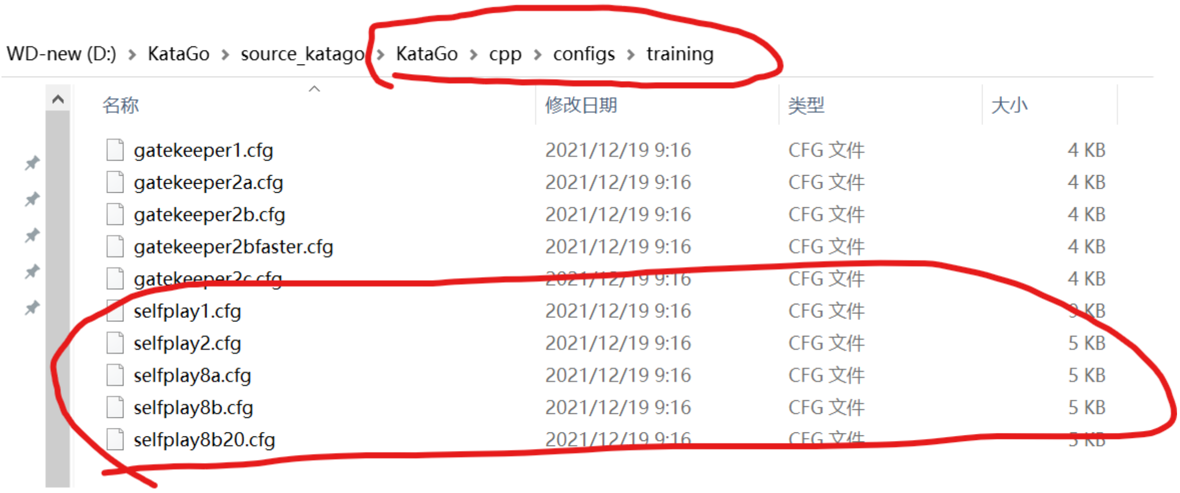
因为开发者自己编译出的katago.exe是使用 cpu做后端，就需要这个关键字

官方的给的cfg中就没有这个关键字，应该是官方用的是别的后端



benchmark.cpp 4
setup.cpp ×
config_parser.cpp
eigenbackend.cpp
pp > program > setup.cpp > initializeNNEvaluators(const vector<string>&, const vector<string>&, const vector<string>&, ConfigParser &, Logger &, Rand &,
numEigenThreadsPer

```
      nnRandSeed = Global::uint64ToString(seedRand.nextUInt64());
    logger.write("nnRandSeed" + idxStr + " = " + nnRandSeed);

#ifndef USE_EIGEN_BACKEND
    (void)expectedConcurrentEvals;
    cfg.markAllKeysUsedWithPrefix("numEigenThreadsPerModel");
    int numNNServerThreadsPerModel =
      cfg.contains("numNNServerThreadsPerModel") ? cfg.getInt("numNNServerThreadsPerModel",1,1024) : 1;
#else
    cfg.markAllKeysUsedWithPrefix("numNNServerThreadsPerModel");
    auto getNumCores = [&logger]() {
      int numCores = (int)std::thread::hardware_concurrency();
      if(numCores <= 0) {
        logger.write("Could not determine number of cores on this machine, choosing default parameters as if it were 8
        numCores = 8;
      }
      return numCores;
    };
    int numNNServerThreadsPerModel =
      cfg.contains("numEigenThreadsPerModel") ? cfg.getInt("numEigenThreadsPerModel",1,1024) :
      setupFor == SETUP_FOR_DISTRIBUTED ? std::min(expectedConcurrentEvals,getNumCores()) :
      setupFor == SETUP_FOR_MATCH ? std::min(expectedConcurrentEvals,getNumCores()) :
      setupFor == SETUP_FOR_ANALYSIS ? std::min(expectedConcurrentEvals,getNumCores()) :
      setupFor == SETUP_FOR_GTP ? expectedConcurrentEvals :
      setupFor == SETUP_FOR_BENCHMARK ? expectedConcurrentEvals :
      cfg.getInt("numEigenThreadsPerModel",1,1024);
#endif
```



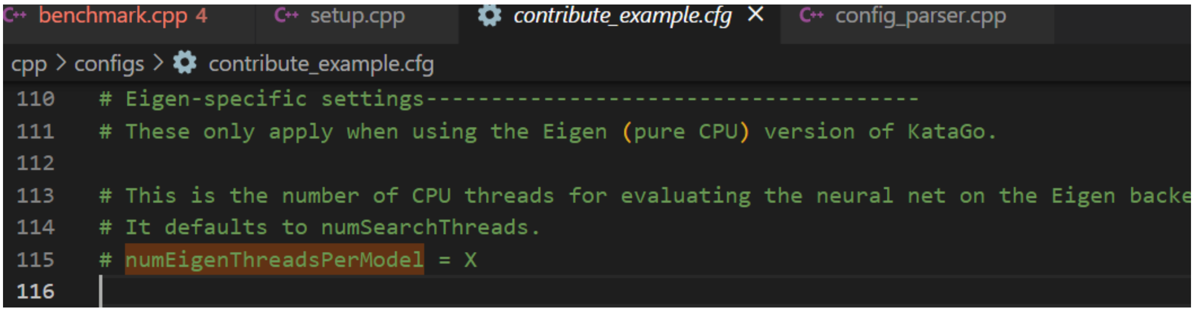
WD-new (D:) > KataGo > source_katago > KataGo > cpp > configs > training

名称	修改日期	类型	大小
gatekeeper1.cfg	2021/12/19 9:16	CFG 文件	4 KB
gatekeeper2a.cfg	2021/12/19 9:16	CFG 文件	4 KB
gatekeeper2b.cfg	2021/12/19 9:16	CFG 文件	4 KB
gatekeeper2bfaster.cfg	2021/12/19 9:16	CFG 文件	4 KB
gatekeeper2c.cfg	2021/12/19 9:16	CFG 文件	4 KB
selfplay1.cfg	2021/12/19 9:16	CFG 文件	9 KB
selfplay2.cfg	2021/12/19 9:16	CFG 文件	5 KB
selfplay8a.cfg	2021/12/19 9:16	CFG 文件	5 KB
selfplay8b.cfg	2021/12/19 9:16	CFG 文件	5 KB
selfplay8b20.cfg	2021/12/19 9:16	CFG 文件	5 KB

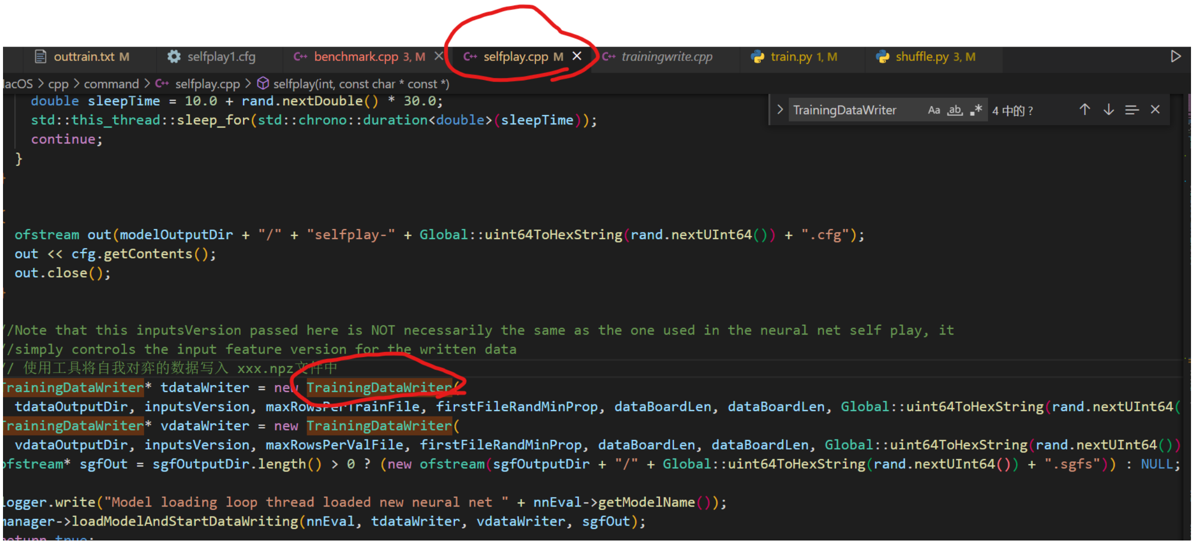
官方解释

这是在Eigen后端评估神经网的CPU线程数。 使用Eigen的katago.exe 进行selfplay 来生成数据就要添加该关键字

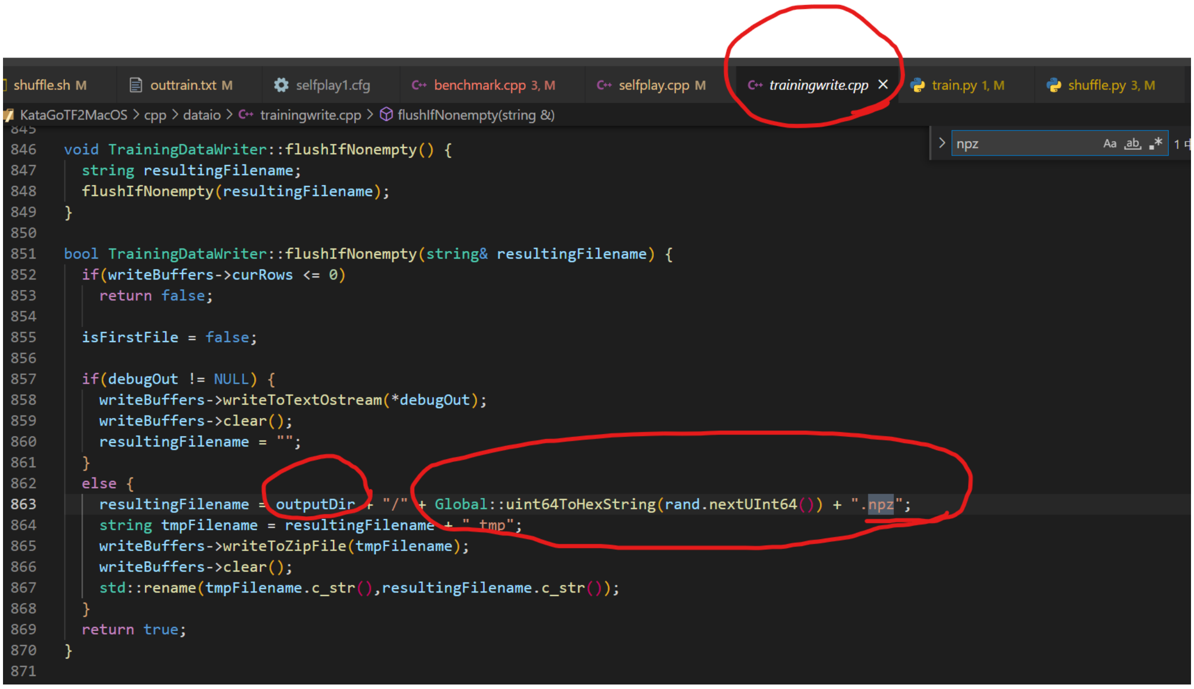
它默认为numSearchThreads。



```
# Eigen-specific settings--------------------------------------
# These only apply when using the Eigen (pure CPU) version of KataGo.

# This is the number of CPU threads for evaluating the neural net on the Eigen backe
# It defaults to numSearchThreads.
# numEigenThreadsPerModel = X

```



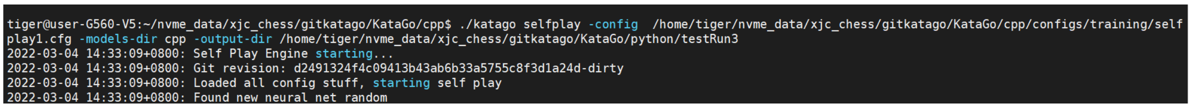
```
   double sleepTime = 10.0 + rand.nextDouble() * 30.0;
   std::this_thread::sleep_for(std::chrono::duration<double>(sleepTime));
   continue;
 }


 ofstream out(modelOutputDir + "/" + "selfplay-" + Global::uint64ToHexString(rand.nextUInt64()) + ".cfg");
 out << cfg.getContents();
 out.close();


/Note that this inputsVersion passed here is NOT necessarily the same as the one used in the neural net self play, it
/simply controls the input feature version for the written data
/ 使用工具将自我对弈的数据写入 xxx.npz文件中
rainingDataWriter* tdataWriter = new TrainingDataWriter(
 tdataOutputDir, inputsVersion, maxRowsPerTrainFile, firstFileRandMinProp, dataBoardLen, dataBoardLen, Global::uint64ToHexString(rand.nextUInt64(
rainingDataWriter* vdataWriter = new TrainingDataWriter(
 vdataOutputDir, inputsVersion, maxRowsPerValFile, firstFileRandMinProp, dataBoardLen, dataBoardLen, Global::uint64ToHexString(rand.nextUInt64())
fstream* sgfOut = sgfOutputDir.length() > 0 ? (new ofstream(sgfOutputDir + "/" + Global::uint64ToHexString(rand.nextUInt64()) + ".sgfs")) : NULL;

ogger.write("Model loading loop thread loaded new neural net " + nnEval->getModelName());
anager->loadModelAndStartDataWriting(nnEval, tdataWriter, vdataWriter, sgfOut);
eturn true;
```



```
void TrainingDataWriter::flushIfNonempty() {
  string resultingFilename;
  flushIfNonempty(resultingFilename);
}

bool TrainingDataWriter::flushIfNonempty(string& resultingFilename) {
  if(writeBuffers->curRows <= 0)
    return false;

  isFirstFile = false;

  if(debugOut != NULL) {
    writeBuffers->writeToTextOstream(*debugOut);
    writeBuffers->clear();
    resultingFilename = "";
  }
  else {
    resultingFilename = outputDir + "/" + Global::uint64ToHexString(rand.nextUInt64()) + ".npz";
    string tmpFilename = resultingFilename + ".tmp";
    writeBuffers->writeToZipFile(tmpFilename);
    writeBuffers->clear();
    std::rename(tmpFilename.c_str(),resultingFilename.c_str());
  }
  return true;
}
```

上面就是katago.exe selfplay 自我博弈生成并保存数据为 npz格式的基本代码路径

Linux下



```
tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo/cpp$ ./katago selfplay -config  /home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/cpp/configs/training/self
play1.cfg -models-dir cpp -output-dir /home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun3
2022-03-04 14:33:09+0800: Self Play Engine starting...
2022-03-04 14:33:09+0800: Git revision: d2491324f4c09413b43ab6b33a5755c8f3d1a24d-dirty
2022-03-04 14:33:09+0800: Loaded all config stuff, starting self play
2022-03-04 14:33:09+0800: Found new neural net random
```

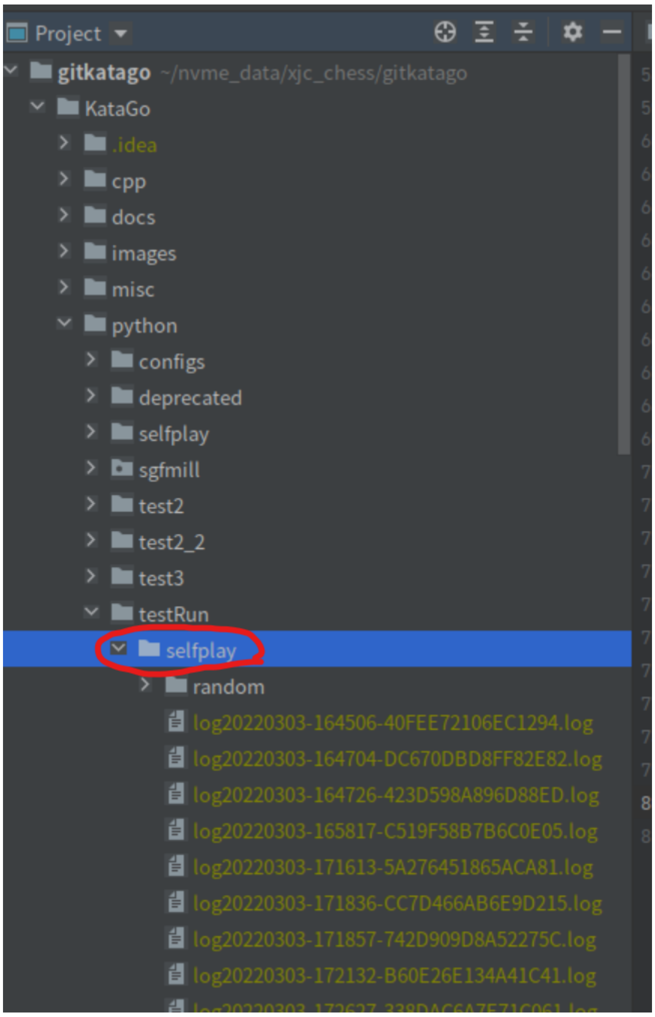
第二步

数据集打乱操作

把第一步生成的数据集拷贝

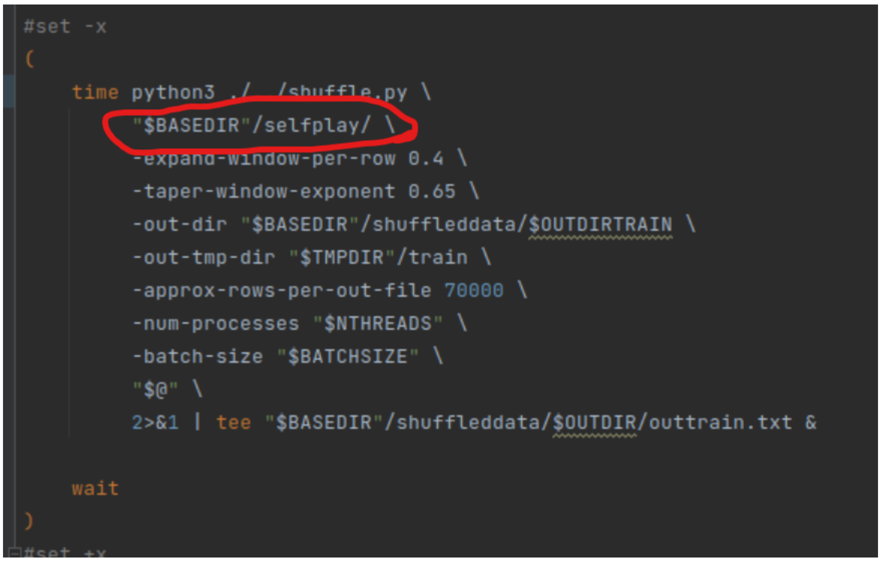
注意，我是在Windows10下生成数据集，在Linux下执行shuffled打乱操作

自己新建一个文件夹，注意这个文件夹位置，因为后续很多操作需要这个文件夹


Project
gitkatago ~/nvme_data/xjc_chess/gitkatago
KataGo
.idea
cpp
docs
images
misc
python
configs
deprecated
selfplay
sgfmill
test2
test2_2
test3
testRun
selfplay
random
log20220303-164506-40FEE72106EC1294.log
log20220303-164704-DC670DBD8FF82E82.log
log20220303-164726-423D598A896D88ED.log
log20220303-165817-C519F58B7B6C0E05.log
log20220303-171613-5A276451865ACA81.log
log20220303-171836-CC7D466AB6E9D215.log
log20220303-171857-742D909D8A52275C.log
log20220303-172132-B60E26E134A41C41.log
log20220303-172627-338DAC6A7E71C061.log

注意

注意，我把第一步数据集的文件夹名字改了，这是必须的，因为>>>>



```
#set -x
(
    time python3 ./../shuffle.py \
        "$BASEDIR"/selfplay/ \
        -expand-window-per-row 0.4 \
        -taper-window-exponent 0.65 \
        -out-dir "$BASEDIR"/shuffleddata/$OUTDIRTRAIN \
        -out-tmp-dir "$TMPDIR"/train \
        -approx-rows-per-out-file 70000 \
        -num-processes "$NTHREADS" \
        -batch-size "$BATCHSIZE" \
        "$@" \
        2>&1 | tee "$BASEDIR"/shuffleddata/$OUTDIR/outtrain.txt &

    wait
)
#set +x
```

必须改，不然不对，要么就是改脚本，避免不必要的麻烦，还是该文件名

$BASEDIR 就是哪个自己创建的testRun ， 在运行脚本是要全路径

根据脚本参数需求输入运行参数

#./shuffle.sh

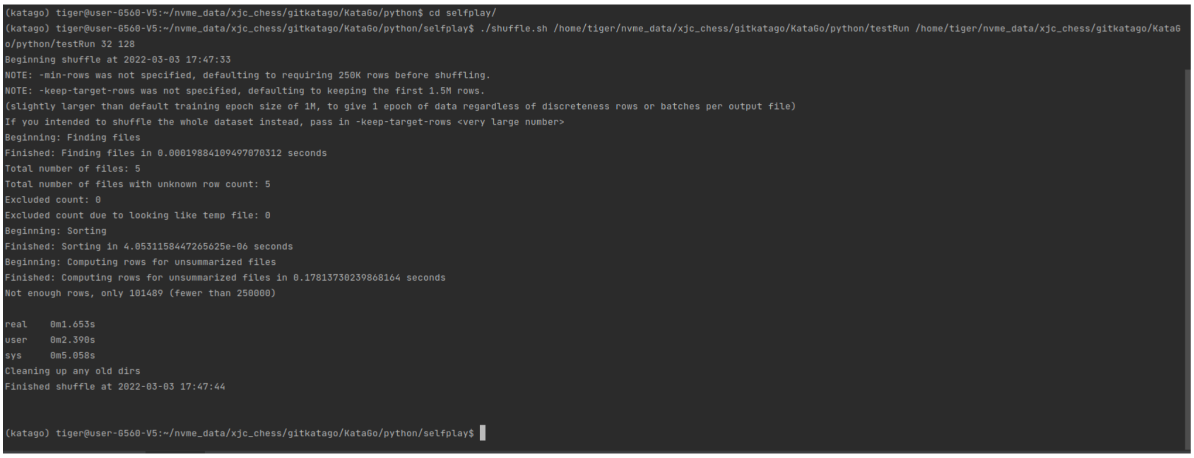
/home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun #就是自己创建的用来存放了game数据的目录

/home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun # 临时目录

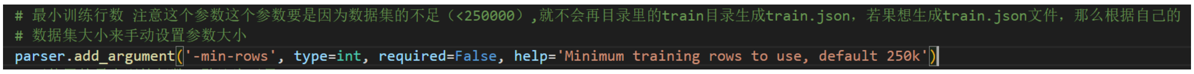
32 # 线程数

128 # batch-size

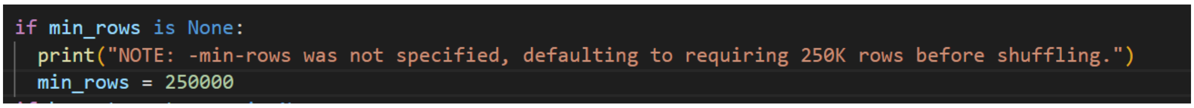
运行状态



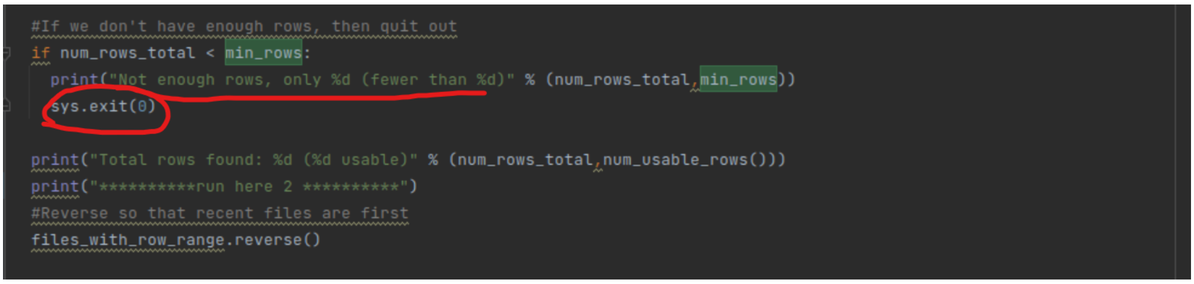
```
(katago) tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo/python$ cd selfplay/
(katago) tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo/python/selfplay$ ./shuffle.sh /home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun /home/tiger/nvme_data/xjc_chess/gitkatago/KataG
o/python/testRun 32 128
Beginning shuffle at 2022-03-03 17:47:33
NOTE: -min-rows was not specified, defaulting to requiring 250K rows before shuffling.
NOTE: -keep-target-rows was not specified, defaulting to keeping the first 1.5M rows.
(slightly larger than default training epoch size of 1M, to give 1 epoch of data regardless of discreteness rows or batches per output file)
If you intended to shuffle the whole dataset instead, pass in -keep-target-rows <very large number>
Beginning: Finding files
Finished: Finding files in 0.00019884109497070312 seconds
Total number of files: 5
Total number of files with unknown row count: 5
Excluded count: 0
Excluded count due to looking like temp file: 0
Beginning: Sorting
Finished: Sorting in 4.0531158447265625e-06 seconds
Beginning: Computing rows for unsummarized files
Finished: Computing rows for unsummarized files in 0.17813730239868164 seconds
Not enough rows, only 101489 (fewer than 250000)

real    0m1.653s
user    0m2.390s
sys     0m5.058s
Cleaning up any old dirs
Finished shuffle at 2022-03-03 17:47:44


(katago) tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo/python/selfplay$ 
```



```
# 最小训练行数 注意这个参数这个参数要是因为数据集的不足（<250000）,就不会再目录里的train目录生成train.json，若果想生成train.json文件，那么根据自己的
# 数据集大小来手动设置参数大小
parser.add_argument('-min-rows', type=int, required=False, help='Minimum training rows to use, default 250k')
```



```
if min_rows is None:
  print("NOTE: -min-rows was not specified, defaulting to requiring 250K rows before shuffling.")
  min_rows = 250000
```



```
#If we don't have enough rows, then quit out
if num_rows_total < min_rows:
  print("Not enough rows, only %d (fewer than %d)" % (num_rows_total,min_rows))
  sys.exit(0)

print("Total rows found: %d (%d usable)" % (num_rows_total,num_usable_rows()))
print("***********run here 2 **********")
#Reverse so that recent files are first
files_with_row_range.reverse()
```

数据量不足就会退出，所以在使用katago.exe selfplay 时最后生成足够多的数据，或者在

./shuffle.sh

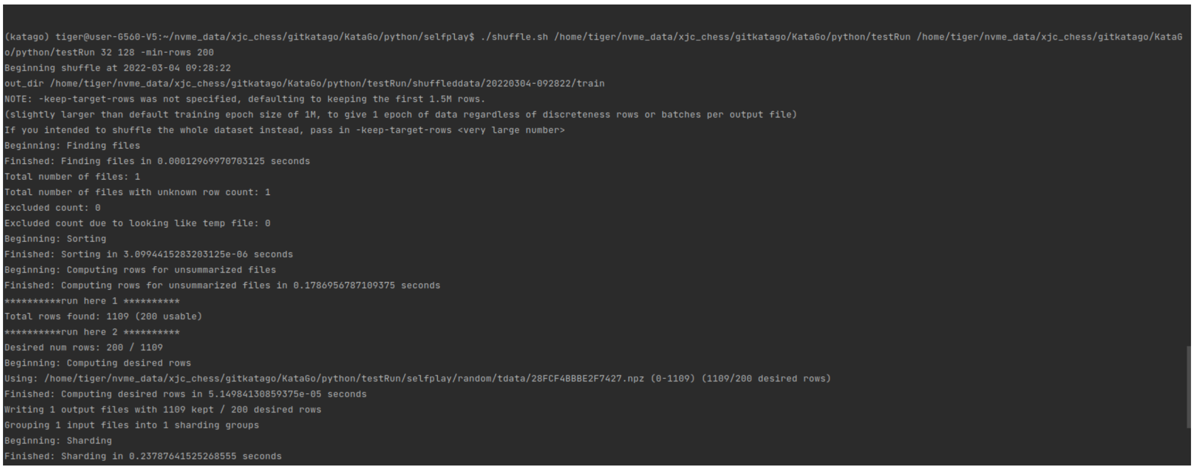
/home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun /home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun

32

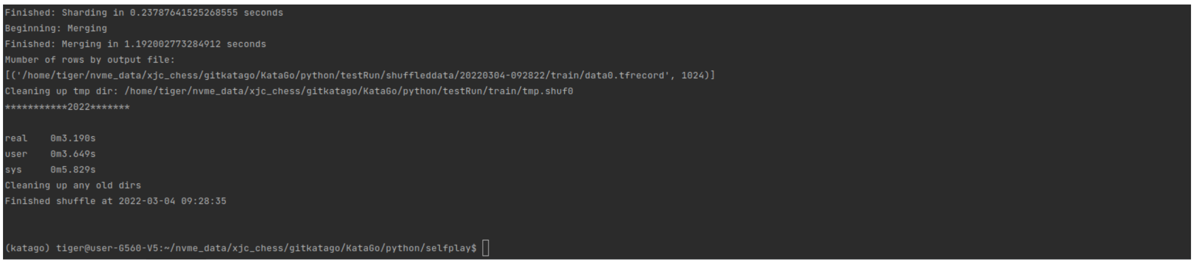
128

-min-rows 200 指定参数大小，默认是250000，数据不足就不会有train.json生成和 xxx.tfrecord等文件输出

正确情况下

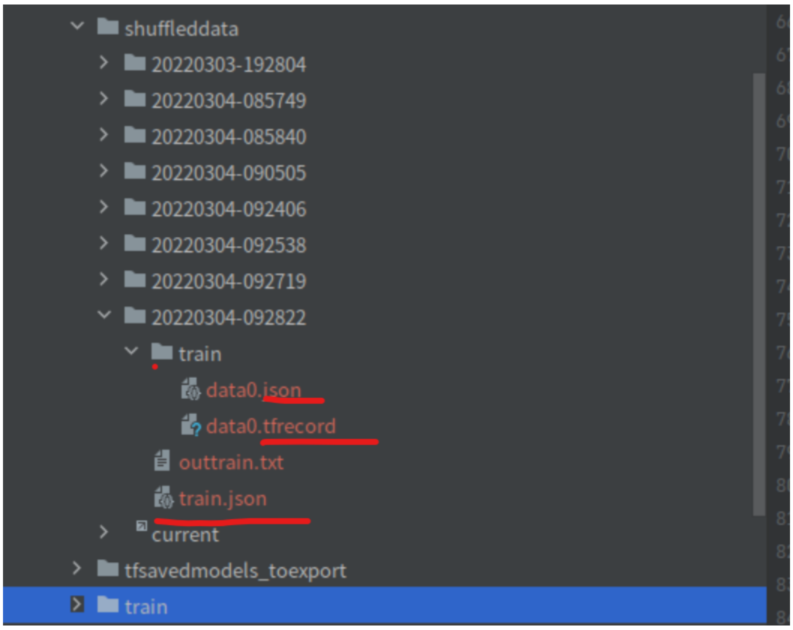


```
(katago) tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo/python/selfplay$ ./shuffle.sh /home/tiger/nvme_data/xjc_chess/gitkatago/KataG
o/python/testRun 32 128 -min-rows 200
Beginning shuffle at 2022-03-04 09:28:22
out_dir /home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun/shuffleddata/20220304-092822/train
NOTE: -keep-target-rows was not specified, defaulting to keeping the first 1.5M rows.
(slightly larger than default training epoch size of 1M, to give 1 epoch of data regardless of discreteness rows or batches per output file)
If you intended to shuffle the whole dataset instead, pass in -keep-target-rows <very large number>
Beginning: Finding files
Finished: Finding files in 0.00012969970703125 seconds
Total number of files: 1
Total number of files with unknown row count: 1
Excluded count: 0
Excluded count due to looking like temp file: 0
Beginning: Sorting
Finished: Sorting in 3.0994415283203125e-06 seconds
Beginning: Computing rows for unsummarized files
Finished: Computing rows for unsummarized files in 0.1786956787109375 seconds
***********run here 1 **********
Total rows found: 1109 (200 usable)
***********run here 2 **********
Desired num rows: 200 / 1109
Beginning: Computing desired rows
Using: /home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun/selfplay/random/tdata/28FCF4BBBE2F7427.npz (0-1109) (1109/200 desired rows)
Finished: Computing desired rows in 5.14984130859375e-05 seconds
Writing 1 output files with 1109 kept / 200 desired rows
Grouping 1 input files into 1 sharding groups
Beginning: Sharding
Finished: Sharding in 0.23787641525268555 seconds
```



```
Finished: Sharding in 0.23787641525268555 seconds
Beginning: Merging
Finished: Merging in 1.192002773284912 seconds
Mumber of rows by output file:
[('/home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun/shuffleddata/20220304-092822/train/data0.tfrecord', 1024)]
Cleaning up tmp dir: /home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun/train/tmp.shuf0
***********2022*******

real    0m3.190s
user    0m3.649s
sys     0m5.829s
Cleaning up any old dirs
Finished shuffle at 2022-03-04 09:28:35


(katago) tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo/python/selfplay$ 
```

正确下的目录及目录下内容



```
shuffleddata
  20220303-192804
  20220304-085749
  20220304-085840
  20220304-090505
  20220304-092406
  20220304-092538
  20220304-092719
  20220304-092822
    train
      data0.json
      data0.tfrecord
    outtrain.txt
    train.json
  current
tfsavedmodels_toexport
train
```

第三步

#./train.sh

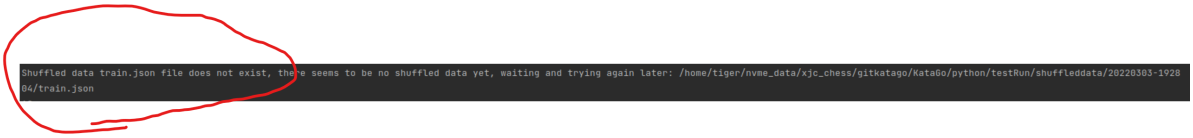
/home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun # 这个就是前面的目录

trainruning # 文件名前缀，目前结果看，就是在testRun目录下创建train目录再在该目录下创建trainruning目录，训练的结果文件就存在这个目录里面

b2c16

128

main



```
Shuffled data train.json file does not exist, there seems to be no shuffled data yet, waiting and trying again later: /home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun/shuffleddata/20220303-1928
04/train.json
```

Shuffled data train.json file does not exist

不知道这个JSON是在前面的步骤中自己生成还是要自己写，先复制一个进入解决眼前的问题

解决：这个是因为第二步shuffled中的min_rows这个参数所导致的，解决了就没有这个问题了

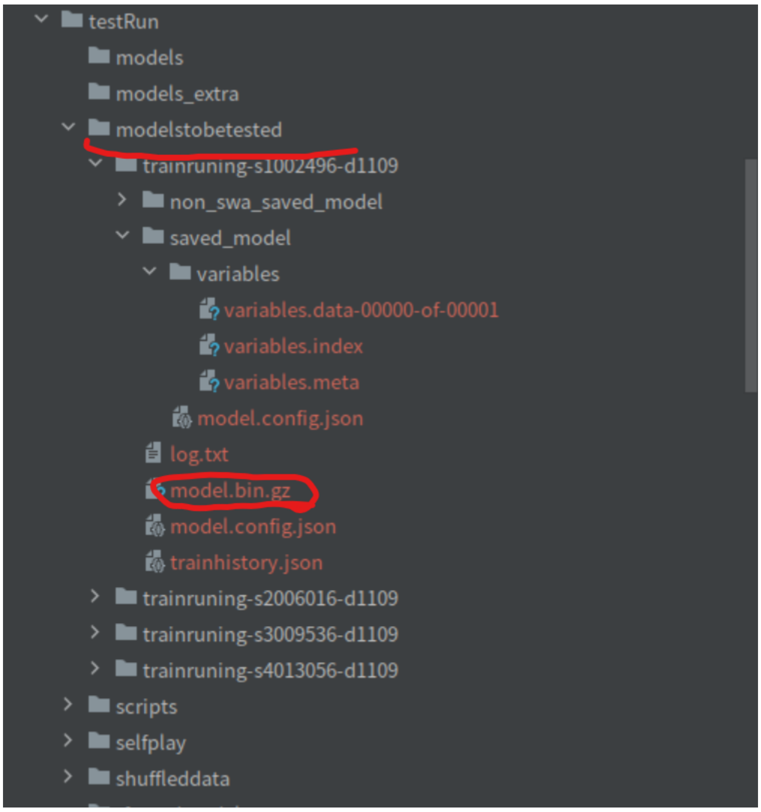
第四步

- Exporter (python - `python/export_model.py`) - scans a directory of saved models and converts from Tensorflow's format to the format that all the C++ uses, exporting to some directory.

扫描一个保存模型的目录，并从Tensorflow的格式转换为所有C++使用的格式，导出到某个目录。

就是将第三步中训练的模型文件转换为c++引擎所需要的，就是在katago.exe 引擎所能解析的网络模型文件格式

./export_model_for_selfplay.sh 20220304 /home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/testRun 1



```
testRun
  models
  models_extra
  modelstobetested
    trainruning-s1002496-d1109
      non_swa_saved_model
      saved_model
        variables
          variables.data-00000-of-00001
          variables.index
          variables.meta
        model.config.json
      log.txt
      model.bin.gz
      model.config.json
      trainhistory.json
    trainruning-s2006016-d1109
    trainruning-s3009536-d1109
    trainruning-s4013056-d1109
  scripts
  selfplay
  shuffleddata
```

第五步

守门员

- Gatekeeper (C++ - `cpp/katago gatekeeper`) - polls a directory of newly exported models, plays games against the latest model in an accepted models directory, and if the new model passes, moves it to the accepted models directory. OPTIONAL, it is also possible to train just accepting every new model.

守门员 (C++ - cpp/katago gatekeeper) - 轮询一个新导出模型的目录，与接受模型目录中的最新模型进行博弈，如果新模型通过，则将其移至接受模型目录中。可选的，也可以只训练接受每个新模型。

意思就是"强者胜出"，将最新的训练模型导出成c++下格式后，使用gatekeeper.cpp 脚本与上一个次新的网络模型进行相互对弈，以获得更强的结果，强的网络就保存，用来和下一个最新网络来对弈，打擂台决出强者

./katago.exe gatekeeper

-test-models-dir   #放置了对弈双方模型文件的目录

-sgf-output-dir # 保存对弈的sgf文件

-accepted-models-dir # 保存对弈双方的强者的网络模型文件路径

-rejected-models-dir # 保存对弈双方的弱者的网络模型文件路径

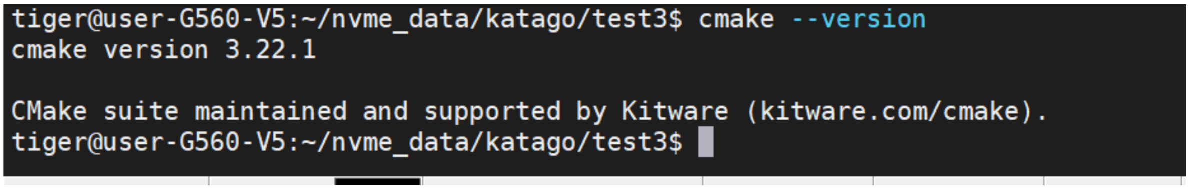
-selfplay-dir # 保存对弈信息，当一个模型选择pass

暂时这么理解

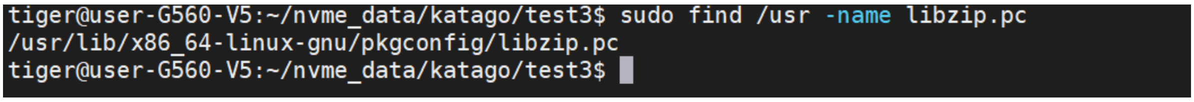
在Ubuntu服务器上编译-运行katago

当然在编译前需要在设备中安装好各个所需的包和环境

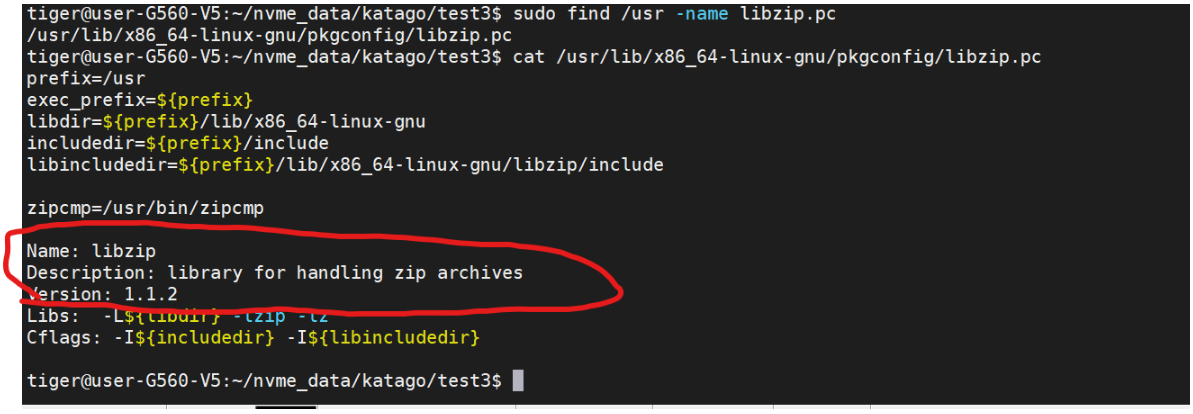
编译



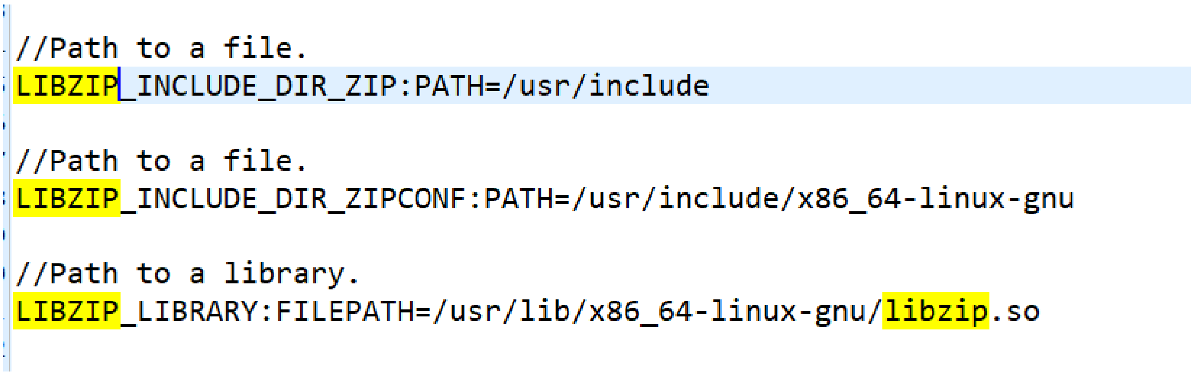
```
tiger@user-G560-V5:~/nvme_data/katago/test3$ cmake --version
cmake version 3.22.1

CMake suite maintained and supported by Kitware (kitware.com/cmake).
tiger@user-G560-V5:~/nvme_data/katago/test3$ 
```



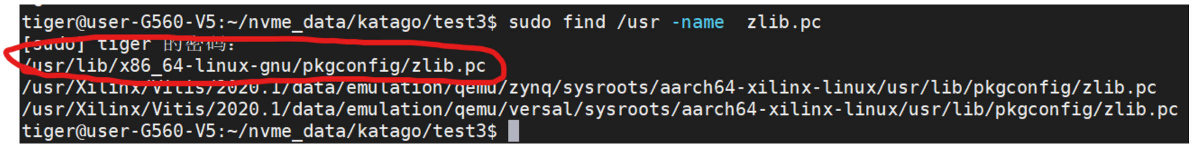
```
tiger@user-G560-V5:~/nvme_data/katago/test3$ sudo find /usr -name libzip.pc
/usr/lib/x86_64-linux-gnu/pkgconfig/libzip.pc
tiger@user-G560-V5:~/nvme_data/katago/test3$ 
```



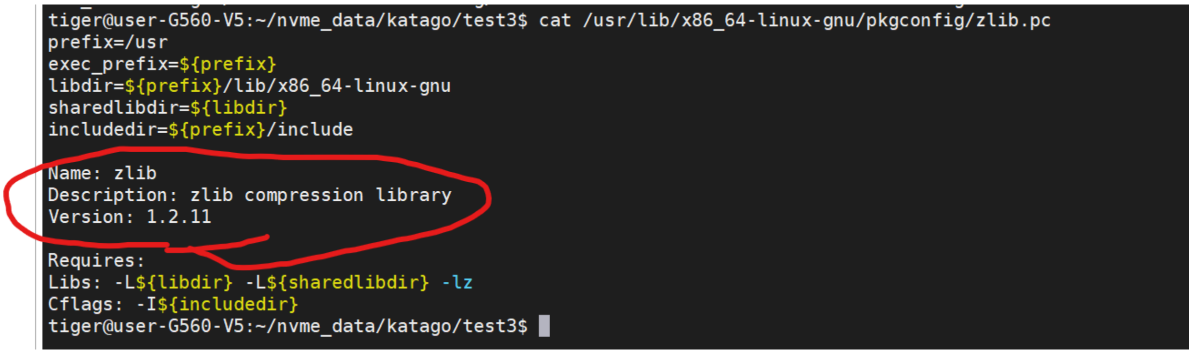
```
tiger@user-G560-V5:~/nvme_data/katago/test3$ sudo find /usr -name libzip.pc
/usr/lib/x86_64-linux-gnu/pkgconfig/libzip.pc
tiger@user-G560-V5:~/nvme_data/katago/test3$ cat /usr/lib/x86_64-linux-gnu/pkgconfig/libzip.pc
prefix=/usr
exec_prefix=${prefix}
libdir=${prefix}/lib/x86_64-linux-gnu
includedir=${prefix}/include
libincludedir=${prefix}/lib/x86_64-linux-gnu/libzip/include

zipcmp=/usr/bin/zipcmp

Name: libzip
Description: library for handling zip archives
Version: 1.1.2
Libs:  -L${libdir} -lzip -lz
Cflags: -I${includedir} -I${libincludedir}

tiger@user-G560-V5:~/nvme_data/katago/test3$
```



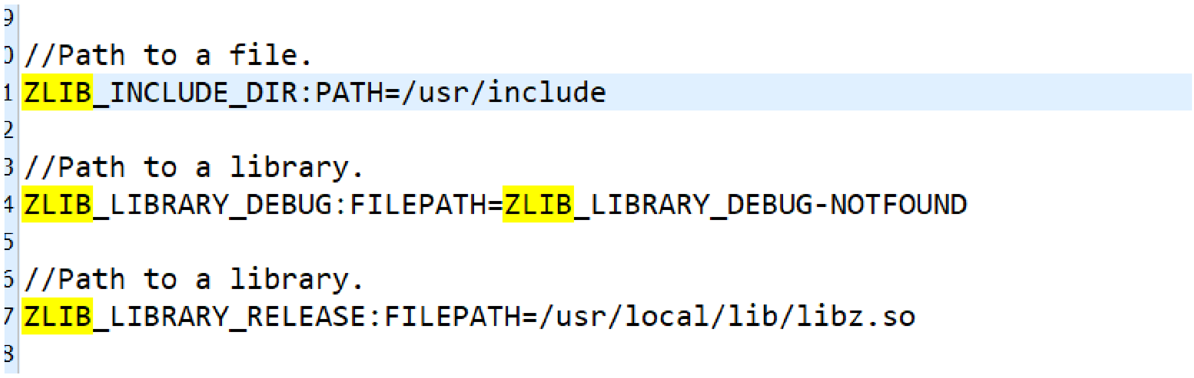
```
//Path to a file.
LIBZIP_INCLUDE_DIR_ZIP:PATH=/usr/include

//Path to a file.
LIBZIP_INCLUDE_DIR_ZIPCONF:PATH=/usr/include/x86_64-linux-gnu

//Path to a library.
LIBZIP_LIBRARY:FILEPATH=/usr/lib/x86_64-linux-gnu/libzip.so
```



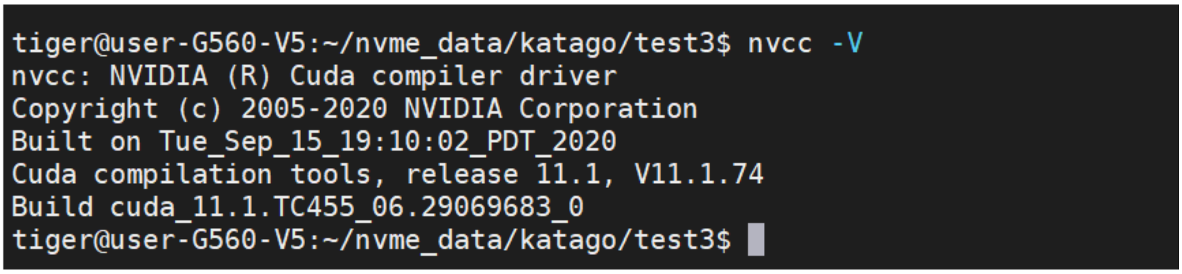
```
tiger@user-G560-V5:~/nvme_data/katago/test3$ sudo find /usr -name  zlib.pc
[sudo] tiger 的密码:
/usr/lib/x86_64-linux-gnu/pkgconfig/zlib.pc
/usr/Xilinx/Vitis/2020.1/data/emulation/qemu/zynq/sysroots/aarch64-xilinx-linux/usr/lib/pkgconfig/zlib.pc
/usr/Xilinx/Vitis/2020.1/data/emulation/qemu/versal/sysroots/aarch64-xilinx-linux/usr/lib/pkgconfig/zlib.pc
tiger@user-G560-V5:~/nvme_data/katago/test3$
```



```
tiger@user-G560-V5:~/nvme_data/katago/test3$ cat /usr/lib/x86_64-linux-gnu/pkgconfig/zlib.pc
prefix=/usr
exec_prefix=${prefix}
libdir=${prefix}/lib/x86_64-linux-gnu
sharedlibdir=${libdir}
includedir=${prefix}/include

Name: zlib
Description: zlib compression library
Version: 1.2.11

Requires:
Libs: -L${libdir} -L${sharedlibdir} -lz
Cflags: -I${includedir}
tiger@user-G560-V5:~/nvme_data/katago/test3$
```



```
//Path to a file.
ZLIB_INCLUDE_DIR:PATH=/usr/include

//Path to a library.
ZLIB_LIBRARY_DEBUG:FILEPATH=ZLIB_LIBRARY_DEBUG-NOTFOUND

//Path to a library.
ZLIB_LIBRARY_RELEASE:FILEPATH=/usr/local/lib/libz.so
```

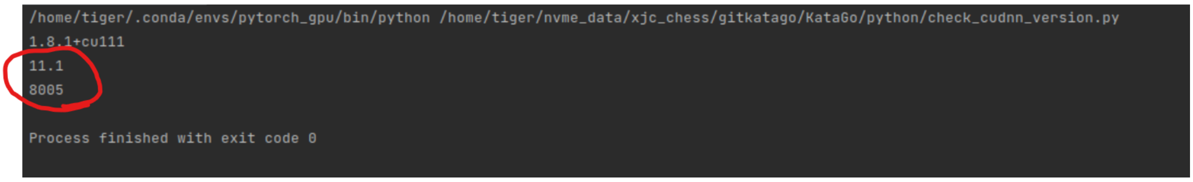


```
tiger@user-G560-V5:~/nvme_data/katago/test3$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Tue_Sep_15_19:10:02_PDT_2020
Cuda compilation tools, release 11.1, V11.1.74
Build cuda_11.1.TC455_06.29069683_0
tiger@user-G560-V5:~/nvme_data/katago/test3$
```

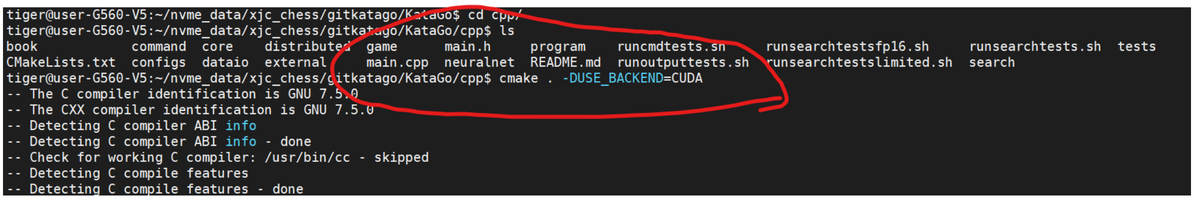
查看cuda与cudnn版本

import torch
print(torch.version)

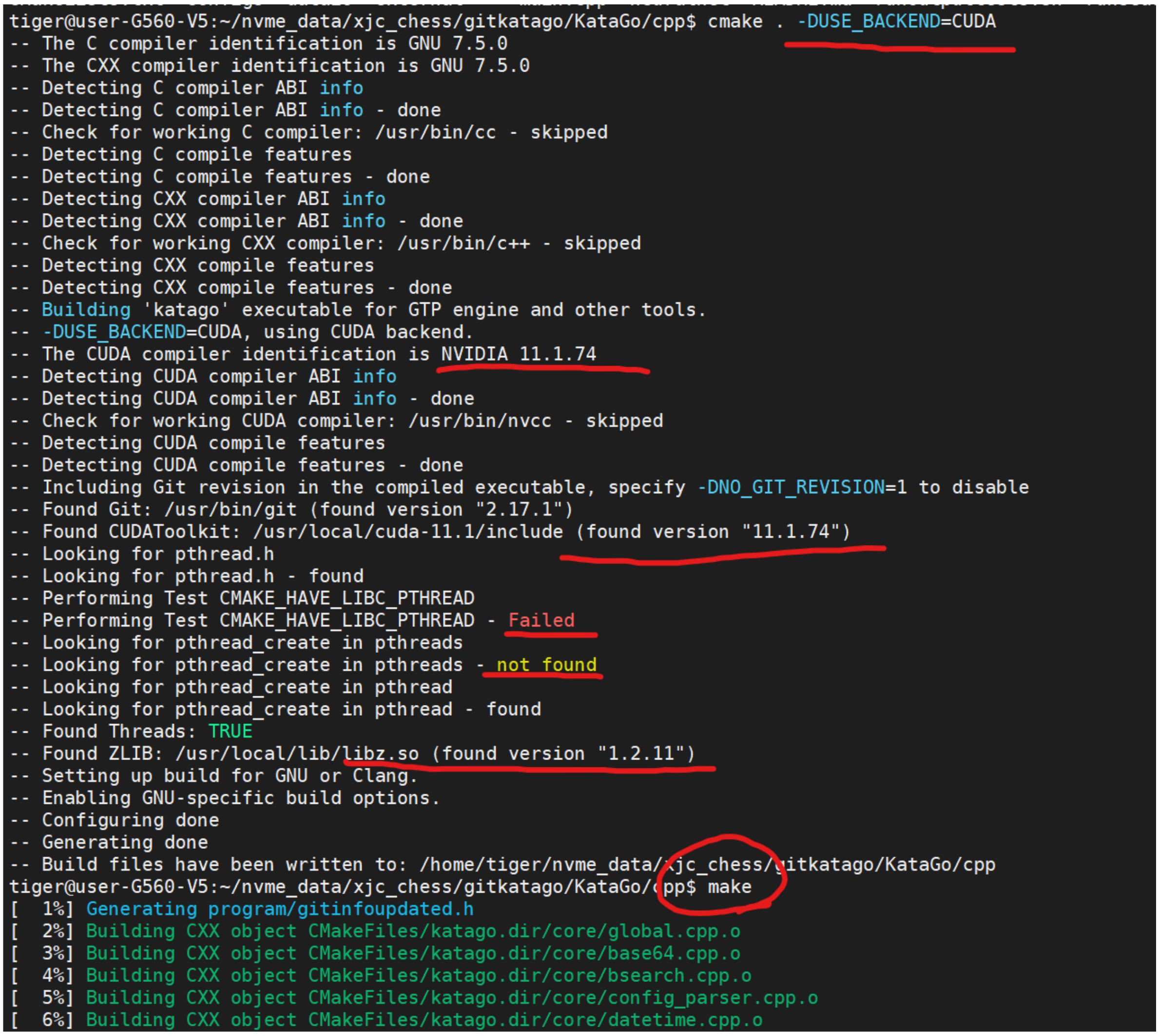
print(torch.version.cuda)
print(torch.backends.cudnn.version())



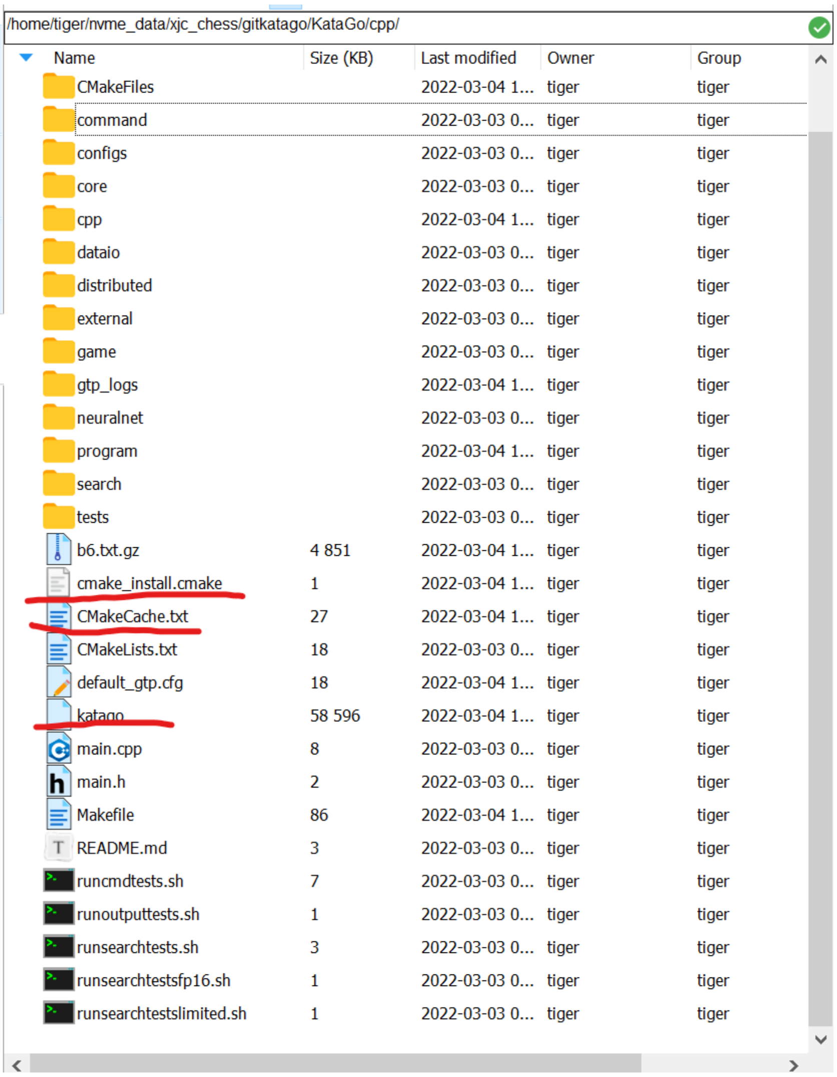
```
/home/tiger/.conda/envs/pytorch_gpu/bin/python /home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/python/check_cudnn_version.py
1.8.1+cu111
11.1
8005

Process finished with exit code 0
```



```
tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo$ cd cpp/
tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo/cpp$ ls
book           command  core    distributed  game      main.h     program    runcmdtests.sh     runsearchtestsfp16.sh     runsearchtests.sh  tests
CMakeLists.txt configs  dataio  external     main.cpp  neuralnet  README.md  runoutputtests.sh  runsearchtestslimited.sh  search
tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo/cpp$ cmake . -DUSE_BACKEND=CUDA
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
```



```
tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo/cpp$ cmake . -DUSE_BACKEND=CUDA
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Building 'katago' executable for GTP engine and other tools.
-- -DUSE_BACKEND=CUDA, using CUDA backend.
-- The CUDA compiler identification is NVIDIA 11.1.74
-- Detecting CUDA compiler ABI info
-- Detecting CUDA compiler ABI info - done
-- Check for working CUDA compiler: /usr/bin/nvcc - skipped
-- Detecting CUDA compile features
-- Detecting CUDA compile features - done
-- Including Git revision in the compiled executable, specify -DNO_GIT_REVISION=1 to disable
-- Found Git: /usr/bin/git (found version "2.17.1")
-- Found CUDAToolkit: /usr/local/cuda-11.1/include (found version "11.1.74")
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD
-- Performing Test CMAKE_HAVE_LIBC_PTHREAD - Failed
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Found Threads: TRUE
-- Found ZLIB: /usr/local/lib/libz.so (found version "1.2.11")
-- Setting up build for GNU or Clang.
-- Enabling GNU-specific build options.
-- Configuring done
-- Generating done
-- Build files have been written to: /home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/cpp
tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo/cpp$ make
[  1%] Generating program/gitinfoupdated.h
[  2%] Building CXX object CMakeFiles/katago.dir/core/global.cpp.o
[  3%] Building CXX object CMakeFiles/katago.dir/core/base64.cpp.o
[  4%] Building CXX object CMakeFiles/katago.dir/core/bsearch.cpp.o
[  5%] Building CXX object CMakeFiles/katago.dir/core/config_parser.cpp.o
[  6%] Building CXX object CMakeFiles/katago.dir/core/datetime.cpp.o
```



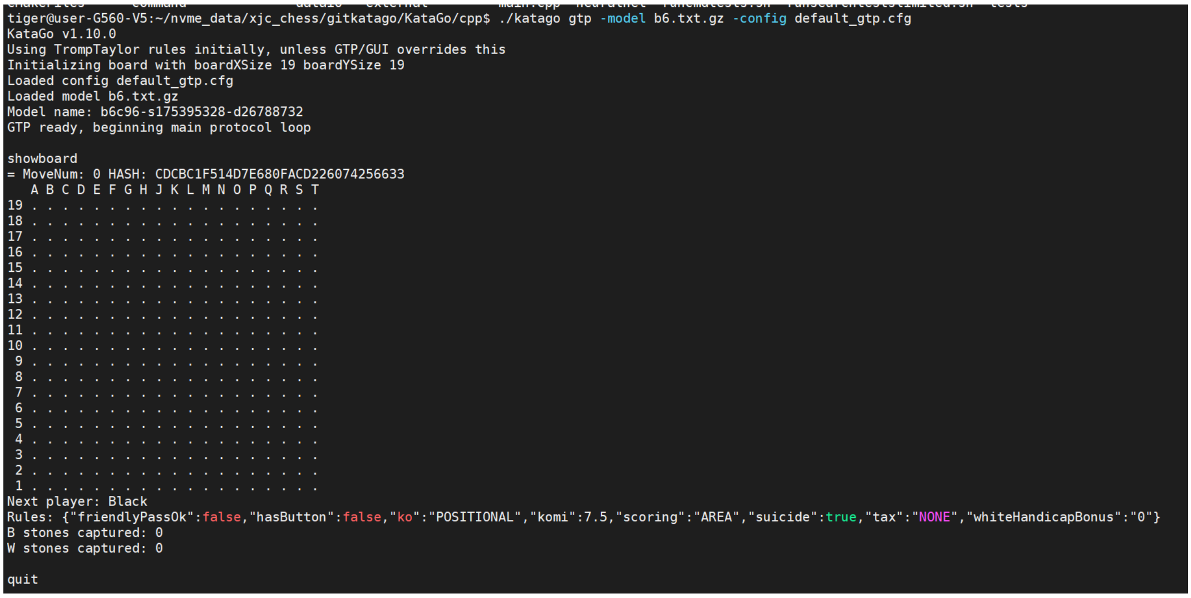
/home/tiger/nvme_data/xjc_chess/gitkatago/KataGo/cpp/

Name	Size (KB)	Last modified	Owner	Group
CMakeFiles		2022-03-04 1...	tiger	tiger
command		2022-03-03 0...	tiger	tiger
configs		2022-03-03 0...	tiger	tiger
core		2022-03-03 0...	tiger	tiger
cpp		2022-03-04 1...	tiger	tiger
dataio		2022-03-03 0...	tiger	tiger
distributed		2022-03-03 0...	tiger	tiger
external		2022-03-03 0...	tiger	tiger
game		2022-03-03 0...	tiger	tiger
gtp_logs		2022-03-04 1...	tiger	tiger
neuralnet		2022-03-03 0...	tiger	tiger
program		2022-03-04 1...	tiger	tiger
search		2022-03-03 0...	tiger	tiger
tests		2022-03-03 0...	tiger	tiger
b6.txt.gz	4 851	2022-03-04 1...	tiger	tiger
cmake_install.cmake	1	2022-03-04 1...	tiger	tiger
CMakeCache.txt	27	2022-03-04 1...	tiger	tiger
CMakeLists.txt	18	2022-03-03 0...	tiger	tiger
default_gtp.cfg	18	2022-03-04 1...	tiger	tiger
katago	58 596	2022-03-04 1...	tiger	tiger
main.cpp	8	2022-03-03 0...	tiger	tiger
main.h	2	2022-03-03 0...	tiger	tiger
Makefile	86	2022-03-04 1...	tiger	tiger
README.md	3	2022-03-03 0...	tiger	tiger
runcmdtests.sh	7	2022-03-03 0...	tiger	tiger
runoutputtests.sh	1	2022-03-03 0...	tiger	tiger
runsearchtests.sh	3	2022-03-03 0...	tiger	tiger
runsearchtestsfp16.sh	1	2022-03-03 0...	tiger	tiger
runsearchtestslimited.sh	1	2022-03-03 0...	tiger	tiger

编译成功就会多三个文件

运行

运行前需要提前把网络模型文件和对应模式下的cfg配置文件准备好



```
tiger@user-G560-V5:~/nvme_data/xjc_chess/gitkatago/KataGo/cpp$ ./katago gtp -model b6.txt.gz -config default_gtp.cfg
KataGo v1.10.0
Using TrompTaylor rules initially, unless GTP/GUI overrides this
Initializing board with boardXSize 19 boardYSize 19
Loaded config default_gtp.cfg
Loaded model b6.txt.gz
Model name: b6c96-s175395328-d26788732
GTP ready, beginning main protocol loop

showboard
= MoveNum: 0 HASH: CDCBC1F514D7E680FACD226074256633
   A B C D E F G H J K L M N O P Q R S T
19 . . . . . . . . . . . . . . . . . . .
18 . . . . . . . . . . . . . . . . . . .
17 . . . . . . . . . . . . . . . . . . .
16 . . . . . . . . . . . . . . . . . . .
15 . . . . . . . . . . . . . . . . . . .
14 . . . . . . . . . . . . . . . . . . .
13 . . . . . . . . . . . . . . . . . . .
12 . . . . . . . . . . . . . . . . . . .
11 . . . . . . . . . . . . . . . . . . .
10 . . . . . . . . . . . . . . . . . . .
 9 . . . . . . . . . . . . . . . . . . .
 8 . . . . . . . . . . . . . . . . . . .
 7 . . . . . . . . . . . . . . . . . . .
 6 . . . . . . . . . . . . . . . . . . .
 5 . . . . . . . . . . . . . . . . . . .
 4 . . . . . . . . . . . . . . . . . . .
 3 . . . . . . . . . . . . . . . . . . .
 2 . . . . . . . . . . . . . . . . . . .
 1 . . . . . . . . . . . . . . . . . . .
Next player: Black
Rules: {"friendlyPassOk":false,"hasButton":false,"ko":"POSITIONAL","komi":7.5,"scoring":"AREA","suicide":true,"tax":"NONE","whiteHandicapBonus":"0"}
B stones captured: 0
W stones captured: 0

quit
```