## 1. Test environment

| RISC-V information | Version | Arm information | Version |

|---------- |---------------------- |---------- |---------------------- |

| Hardware Information | SG2042 | Hardware Information | Kunpeng 920 |

| Architecture | RISC-V64 | Architecture | AARCH64 |

| Operating System | openEuler 24.03 (LTS) | Operating System | openEuler 23.09 |

| python version | 3.7.12 | python version | 3.7.12 |

| gcc version | 7.3.0 | gcc version | 7.3.0 |

| mpi version | 1.10.3 | mpi version | 1.10.3 |

|protobuf version| 3.1.0 |protobuf version| 3.1.0 |

| libzip version | 1.1.2 | libzip version | 1.1.2 |

| boost version | 1.85.0 | boost version | 1.85.0 |

| openblas version | v0.3.27-261-ge1eef56e0 | openblas version | v0.3.27-261-ge1eef56e0 |

| opencv version | 3.1.0 | opencv version | 3.1.0 |

| swig version | 3.0.10 | swig version | 3.0.10 |

| cntk version | v2.6-167-g0c5189882 | cntk version | v2.6-167-g0c5189882 |

## 2. Test preparation

### Prepare with script

Put everything in the ./temp folder, and the script will put the corresponding files in the corresponding places

```

cd $CNTK\_HOME/../

mkdir -p temp

cp -r xxx/2.patch/\* ./temp

cp -r xxx/3.script/\* ./temp

cp -r xxx/4.dataset/\* ./temp

sudo chmod +x ./temp/\*.sh

#Put the perf script in the corresponding directory

sudo cp ./temp/perf\_exam.sh ./

sudo cp ./temp/performance\_counter.sh ./

sudo cp ./temp/test\_all.sh ./

./temp/test\_prepare.sh

```

### Manual installation and build (with debug log)

#### 1. Test script preparation

Put everything in the ./temp folder

```

cd $CNTK\_HOME/../

mkdir -p temp

cp -r xxx/2.patch/\* ./temp

cp -r xxx/3.script/\* ./temp

cp -r xxx/4.dataset/\* ./temp

```

(1) perf\_exam.sh

Stored in ./2.test/3.script, it is a further encapsulation of performance\_counter.sh, which specifies where the perf results are stored and creates the logs file output by TestDriver.py.

```

cp ./temp/perf\_exam.sh ./

sudo chmod +x ./perf\_exam.sh

cp ./temp/test\_all.sh ./

sudo chmod +x ./test\_all.sh

```

Below is the perf\_exam.sh code

``` perf\_exam.sh

#!/bin/bash

# Get the current system architecture

ARCH=$(uname -m)

# Determine the architecture type

if [[ "$ARCH" == "aarch64" || "$ARCH" == "armv7l" ]]; then

echo "The current system is ARM architecture"

elif [[ "$ARCH" == "riscv64" ]]; then

echo "The current system is RISC-V architecture"

else

echo "The current system is not ARM or RISC-V architecture, but $ARCH"

exit 1

fi

#Store error information of running code

mkdir -p ./logs

#Store perf results

mkdir -p ./results-$ARCH

sudo ./performance\_counter.sh "sudo $1" "./results-$ARCH"

```

(2) performance\_counter.sh

```

cp ./temp/performance\_counter.sh ./

sudo chmod +x ./performance\_counter.sh

```

#### 2. Test set preparation

(1) Additional data sets required by some programs

```

cd temp

tar -xvf dataset.tar

cd dataset

echo "export DATASET\_HOME=$(pwd)" >> ~/.bashrc

source ~/.bashrc

cd ..

#Data set required by the original program

tar -xzf cifar-10-batches-py.tar.gz

cp -r ./cifar-10-batches-py $CNTK\_HOME/Tutorials/ImageHandsOn/

cd ..

```

(2) Dataset required by the original program

```

The same data set, just placed in a specific place and with a different directory structure

cp -r $DATASET\_HOME/Image/CIFAR/v0/tutorial201 $CNTK\_HOME/Examples/Image/DataSets/CIFAR-10

cp -r $DATASET\_HOME/Image/MNIST/v0 $CNTK\_HOME/Examples/Image/DataSets/MNIST

```

#### 3. Modify the test script

```

cp 2.patch/TestDriver.py $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py

```

#### 4. Make some modifications to the original script

\* In the CNTK directory

```

cd $CNTK\_HOME

```

\* Solve the problem of mpi error under root permission

```

find ./ -type f -name "\*run-test" -exec sed -i 's/run "\$MPI\_BINARY" -n/run "\$MPI\_BINARY" --allow-run-as-root -n/g' {} +

find ./ -type f -name "\*run-test" -exec sed -i 's/cntkmpirun "-n/cntkmpirun "--allow-run-as-root -n/g' {} +

```

\* Solve the problem that HTKMLFReader cannot be used

```

git apply --reject ../temp/cntk\_test.patch

find $CNTK\_HOME/Tests/EndToEndTests -type f -exec sed -i '/readerType = "HTKMLFReader"/,+1 s/readMethod = "blockRandomize"/readMethod = "none"/; s/readerType = "HTKMLFReader"/readerType = "HTKDeserializers"/' {} +

find $CNTK\_HOME/Tests/EndToEndTests -type f -name "\*.cntk" -exec sed -i 's/readerType = "HTKMLFReader"/readerType = "HTKDeserializers"/g' {} +

find $CNTK\_HOME/Tests/EndToEndTests -type f -name "\*.cntk" -exec sed -i 's/readerType = HTKMLFReader/readerType = HTKDeserializers/g' {} +

```

\* Solve the segmentation problem

```

find ./ -type f -name "\*run-test" -exec sed -i 's/Instances=2/Instances=3/g' {} +

```

\* Solve the problem that the diff display fails due to blank characters in Linux and Windows

```

find $CNTK\_HOME/Tests/EndToEndTests/Speech/DNN -type f -name "\*run-test" -exec sed -i 's/diff /diff -b /g' {} +

```

#### 5. Python dependency preparation

```

$PY37\_PATH -m pip install PyYAML pytest pillow==4.0.0

```

\*\*The following part may not be used. riscv's python3.7 cannot install its corresponding highest version of pandas\==1.3.5. If necessary, a higher version of python may be required\*\*

```

$PY37\_PATH -m pip install nbformat pybind11 notebook pandas requests matplotlib sphinx\_rtd\_theme google google-api-python-client protobuf==3.20.0

```

#### 6. Call TestDriver.py to see which test sets can be measured (no need to do it)

To view the code that the cpu can run, you must first specify a tag. If there is no tag, all will be matched. The tag that can be used by the cpu is basically bvt, so

```

$PY37\_PATH TestDriver.py list -d cpu -t bvt

$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py list -d cpu -t bvt

```

## 3. Test

Run in the \<workspace\> directory (usually $CNTK\_HOME/../)

### Automatic testing

```

sudo ./test\_all.sh

```

### Manual Testing

\*\*1.file/test.xlsx\*\* stores the commands and precautions for each test sample. Samples that cannot be tested are also explained in the document.

Each sample has its own baseline. If it is inconsistent with the baseline, an error message "failed" will be displayed. Check the output.txt file of the corresponding test in the logs folder to see the cause of the error.

Modified 3. The sample that solves the segmentation problem (related to the number of mpi instances) will show failed. It can run but the result does not match the baseline.

\*\*When running the sample for the second time, you need to delete the sample folder under logs. The code will determine that it has been trained and directly output that it does not need to train, resulting in inconsistency with the baseline and displaying failed\*\*

```

cd $CNTK\_HOME/../logs

rm -rf ...

```

\*\*Test command explanation:

\* perf\_exam.sh: test script

\* $PY37\_PATH: Python env path

\* $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py: Call the test script that comes with cntk

\* -d cpu -s cpu -f release: Run on cpu

\* -r $CNTK\_HOME/../logs: output logs to the corresponding path

\* Text/SLU: test sample name

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Text/SLU"

```

[[CNTK test#4. Test each sample]]

### Problems encountered during testing

#### 1. MPI error reporting issue under root privileges

```

Reason: Running mpi with root privileges will result in an error. It is not recommended to run with root privileges.

Solution:

1. Replace all run-test files in the cntk directory

find $CNTK\_HOME -type f -name "\*run-test" -exec sed -i 's/run "\$MPI\_BINARY" -n/run "\$MPI\_BINARY" --allow-run-as-root -n/g' {} +

find $CNTK\_HOME -type f -name "\*run-test" -exec sed -i 's/cntkmpirun "-n/cntkmpirun "--allow-run-as-root -n/g' {} +

2. Add mpi's PATH to TestDriver.py and specify a specific mpi

os.environ["MPI\_BINARY"] = "/usr/local/mpi/bin/mpiexec"

current\_path = os.environ.get('PATH')

new\_path = '/usr/local/mpi/bin'

os.environ['PATH'] = new\_path + ':' + current\_path if current\_path else new\_path

```

#### 2. HTKMLFReader cannot be used and reports segments fault

```

mistake:

b'MPI Rank 0: htkmlfreader: reading MLF file /home/jingpeng/package/cntk/Tests/EndToEndTests/Speech/Data/glob\_0000.mlf ...[kunpeng2:1755432] \*\*\* Process received signal \*\*\*\n'

b'MPI Rank 0: [kunpeng2:1755432] Signal: Segmentation fault (11)\n'

b'MPI Rank 0: [kunpeng2:1755432] Signal code: Address not mapped (1)\n'

b'MPI Rank 0: [kunpeng2:1755432] Failing at address: (nil)\n'

b'MPI Rank 0: [kunpeng2:1755432] \*\*\* End of error message \*\*\*\n'

reason:

Compare the cntk that can run and the cntk that cannot run to find the difference

Can't run

readerType = "HTKMLFReader"

Able to run

readerType = "HTKDeserializers"

Read the official website documentation, that is, to use HTKDeserializers to deserialize mlf files

https://learn.microsoft.com/en-us/previous-versions/cognitive-toolkit/understanding-readers

| | | |

|---|---|---|

|HTKMLFDeserializer|HTKDeserializers|Deserializers for HTK MLF files|

solve:

find $CNTK\_HOME/Tests/EndToEndTests -type f -name "\*.cntk" -exec sed -i '/readerType = "HTKMLFReader"/,+1 s/readMethod = "blockRandomize"/readerMethod = "none"/; s/readerType = "HTKMLFReader"/readerType = "HTKDeserializers"/' {} +

find $CNTK\_HOME/Tests/EndToEndTests -type f -name "\*.cntk" -exec sed -i 's/readerType = "HTKMLFReader"/readerType = "HTKDeserializers"/g' {} +

find $CNTK\_HOME/Tests/EndToEndTests -type f -name "\*.cntk" -exec sed -i 's/readerType = HTKMLFReader/readerType = HTKDeserializers/g' {} +

```

Note: The reader of speechTrain is to read data, so HTKMLFReader should be changed to HTKDeserializers, and the reader in write is to write data, so when changing HTKMLFReader to HTKDeserializers, readMethod should be changed to none.

```

speechTrain = [

...

reader = [

readerType = "HTKDeserializers"

readMethod = "blockRandomize"

miniBatchMode = "partial"

randomize = "auto"

verbosity = 0

useMersenneTwisterRand=true

features = [

dim = 363

type = "real"

scpFile = "glob\_0000.scp"

]

labels = [

mlfFile = "$DataDir$/glob\_0000.mlf"

labelMappingFile = "$DataDir$/state.list"

labelDim = 132

labelType = "category"

]

]

]

write = [

action = write

modelPath = "$RunDir$/models/cntkSpeech.dnn"

outputNodeNames=ScaledLogLikelihood

deviceId = $DeviceId$

traceLevel = 1

printValues=true

reader = [

readerType = "HTKDeserializers"

readMethod = "none"

miniBatchMode = "partial"

randomize = "auto"

verbosity = 0

features = [

dim = 363

type = "real"

scpFile = "glob\_0000.write.scp"

]

]

```

#### 3. Solve the segmentation problem (related to the number of mpi instances)

```

Error: Signal: Segmentation fault (11)\n'

reason:

When the number of instances is 2, segementation cannot be run.

Comparing the running codes, we found that the instances are all 3.

But the reported baseline is inconsistent.

Among the many baseline inconsistencies, the error reported in the Speech/DNN/ParallelCrossValidation test sample is as follows

Running test Speech/DNN/ParallelCrossValidation (release cpu) - [FAILED] 268.47 sec

[FAILED] DataParallelSGD training parameters must match for each MPI Rank

6 expected lines weren't observed in the output.

First unmatched: MPI Rank 0: 12/12/2017 15:24:05: Starting minibatch loop, DataParallelSGD training (myRank = 0, numNodes = 2, numGradientBits = 64), distributed reading is ENABLED.

See log file for details: /root/jingpeng/package/cntk/../logs/Speech/DNN\_ParallelCrossValidation@release\_cpu/output.txt

0/1 tests passed, 1 failed

The comparison shows that the calculation results are consistent, except that numNodes changes from 1 to 2 due to the change in the number of mpi banks, and there are also 6 rows in the result.

So it can be considered that this one can run.

The reason why Instances=2 cannot run was not found.

solve:

Increase the number of instances from 2 to 3

find $CNTK\_HOME -type f -name "\*run-test" -exec sed -i 's/Instances=2/Instances=3/g' {} +

```

#### 4. Solve the problem that the difference in blank characters between Linux and Windows causes diff to display failed

```

Error: $CNTK\_HOME/Tests/EndToEndTests/Speech/DNN This test example failed, but no error was found when compared with the baseline

reason:

The baseline is for Windows platform, and the diff error occurs because of inconsistent characters.

solve:

Change to diff -b

find $CNTK\_HOME/Tests/EndToEndTests/Speech/DNN -type f -name "\*run-test" -exec sed -i 's/diff /diff -b /g' {} +

```

#### 5. Missing external dataset

```

solve:

Modify TestDriver.py and add the corresponding code

Line 289

os.environ["CNTK\_EXTERNAL\_TESTDATA\_SOURCE\_DIRECTORY"] = args.external if args.external else "

And add in line 860

runSubparser.add\_argument("-e", "--external", help="external dataset")

```

#### 6. Missing libboost\_atomic.so

```

solve:

Modify TestDriver.py and add the corresponding code

current\_ld\_path = os.environ.get('LD\_LIBRARY\_PATH')

new\_ld\_path = '/usr/local/boost-1.85.0/lib'

os.environ['LD\_LIBRARY\_PATH'] = new\_ld\_path + ':' + current\_ld\_path if current\_ld\_path else new\_ld\_path

os.environ['LD\_PRELOAD']='/usr/local/boost-1.85.0/lib/libboost\_atomic.so.1.85.0'

```

#### 7. Under sudo perf, the code needs to be run under sudo

```

solve:

Modify cmdLine in TestDriver.py

cmdLine = ["sudo","-E","bash", "-c", self.testDir + "/run-test 2>&1"]

```

## 4. Test each sample

#### 1. CNTKv2CSharp/ExampleTests/LSTMSequenceClassifierTest

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs CNTKv2CSharp/ExampleTests/LSTMSequenceClassifierTest"

```

\* Test results

```

Running test CNTKv2CSharp/ExampleTests/LSTMSequenceClassifierTest (release cpu) - [FAILED] 0.04 sec

[FAILED] Test run must produce matching results

7 expected lines weren't observed in the output.

First unmatched: Minibatch: 0 CrossEntropyLoss = 1.597417, EvaluationCriterion = 0.5

[FAILED] Test run must be completed

1 expected lines weren't observed in the output.

First unmatched: ======== Train completes. ========

See log file for details: /root/jingpeng/package/cntk/../logs/CNTKv2CSharp/ExampleTests\_LSTMSequenceClassifierTest@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

\* Cause of error:

Requires NET Framework

#### 2. CNTKv2CSharp/ExampleTests/LogisticRegressionTest

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs CNTKv2CSharp/ExampleTests/LogisticRegressionTest"

```

\* Test results

```

Running test CNTKv2CSharp/ExampleTests/LogisticRegressionTest (release cpu) - [FAILED] 0.04 sec

[FAILED] Test run must be completed

1 expected lines weren't observed in the output.

First unmatched: ======== Train completes. ========

See log file for details: /root/jingpeng/package/cntk/../logs/CNTKv2CSharp/ExampleTests\_LogisticRegressionTest@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

\* Cause of error:

Requires NET Framework

#### 3. CNTKv2CSharp/ExampleTests/MNISTClassifierTest

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs CNTKv2CSharp/ExampleTests/MNISTClassifierTest"

```

\* Test results

```

Running test CNTKv2CSharp/ExampleTests/MNISTClassifierTest (release cpu) - [FAILED] 0.04 sec

[FAILED] Test run must produce matching results

4 expected lines weren't observed in the output.

First unmatched: Minibatch: 0 CrossEntropyLoss = 2.307698, EvaluationCriterion = 0.875

[FAILED] Test run must be completed

1 expected lines weren't observed in the output.

First unmatched: ======== Train completes. ========

See log file for details: /root/jingpeng/package/cntk/../logs/CNTKv2CSharp/ExampleTests\_MNISTClassifierTest@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

\* Cause of error:

Requires NET Framework

#### 4. CNTKv2CSharp/ExampleTests/SimpleFeedForwardClassifierTest

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs CNTKv2CSharp/ExampleTests/SimpleFeedForwardClassifierTest"

```

\* Test results

```

Running test CNTKv2CSharp/ExampleTests/SimpleFeedForwardClassifierTest (release cpu) - [FAILED] 0.05 sec

[FAILED] Test run must produce matching results

20 expected lines weren't observed in the output.

First unmatched: Minibatch: 0 CrossEntropyLoss = 0.694379, EvaluationCriterion = 0.5

[FAILED] Test run must be completed

1 expected lines weren't observed in the output.

First unmatched: ======== Train completes. ========

See log file for details: /root/jingpeng/package/cntk/../logs/CNTKv2CSharp/ExampleTests\_SimpleFeedForwardClassifierTest@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

\* Cause of error:

Requires NET Framework

#### 5. CNTKv2Library/Distribution

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME CNTKv2Library/Distribution"

```

\* Test results

```

Running test CNTKv2Library/Distribution(release cpu) - [OK] 452.55sec

1/1 tests passed, 0 failed

```

#### 6. CNTKv2Library/LSTMSequenceClassifier

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME CNTKv2Library/LSTMSequenceClassifier"

```

\* Test results

```

Running test CNTKv2Library/LSTMSequenceClassifier(release cpu) - [OK] 3.89sec

1/1 tests passed, 0 failed

```

#### 7. CNTKv2Library/MNISTClassifier

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME CNTKv2Library/MNISTClassifier"

```

\* Test results

```

Running test CNTKv2Library/MNISTClassifier(release cpu) - [OK] 13.28sec

1/1 tests passed, 0 failed

```

#### 8. CNTKv2Library/SequenceToSequenceTranslator

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME CNTKv2Library/SequenceToSequenceTranslator"

```

\* Test results

```

Running test CNTKv2Library/SequenceToSequenceTranslator(release cpu) - [OK] 2182.59sec

1/1 tests passed, 0 failed

```

#### 9. CNTKv2Library/TruncatedLSTMAcousticModel

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME CNTKv2Library/TruncatedLSTMAcousticModel"

```

\* Test results

```

Running test CNTKv2Library/TruncatedLSTMAcousticModel(release cpu) - [OK] 8822.69sec

1/1 tests passed, 0 failed

```

#### 10. EvalClientTests/CNTKLibraryCPPEvalExamplesTest

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME EvalClientTests/CNTKLibraryCPPEvalExamplesTest"

```

\* Test results

```

Running test EvalClientTests/CNTKLibraryCPPEvalExamplesTest (release cpu) - [FAILED] 0.04 sec

[FAILED] Exit code must be 0

==> got exit code 1 when running: sudo -E bash -c /root/jingpeng/package/cntk/Tests/EndToEndTests/EvalClientTests/CNTKLibraryCPPEvalExamplesTest/run-test 2>&1

See log file for details: /root/jingpeng/package/cntk/../logs/EvalClientTests\_CNTKLibraryCPPEvalExamplesTest@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

\* Cause of error:

[[CNTK test #5, missing external dataset]]

There is a lack of pre-trained models and the corresponding dataset is not found.

#### 11. Examples/Image/Regression/RegrSimple

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Image/Regression/RegrSimple"

```

\* Test results

```

Running test Examples/Image/Regression/RegrSimple(release cpu) - [OK] 10.27sec

1/1 tests passed, 0 failed

```

#### 12.Examples/Speech/AN4/FeedForward

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Speech/AN4/FeedForward"

```

\* Test results

```

Running test Examples/Speech/AN4/FeedForward(release cpu) - [OK] 44.34sec

1/1 tests passed, 0 failed

```

#### 13. Examples/Speech/AN4/LSTM

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Speech/AN4/LSTM"

```

\* Test results

```

Running test Examples/Speech/AN4/LSTM(release cpu) - [OK] 1055.46sec

1/1 tests passed, 0 failed

```

#### 14. Examples/Text/G2P

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Text/G2P"

```

\* Test results

```

Running test Examples/Text/G2P(release cpu) - [OK] 149.2sec

1/1 tests passed, 0 failed

```

#### 15.Examples/Text/Miscellaneous/SLU

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Text/Miscellaneous/SLU"

```

\* Test results

```

Running test Examples/Text/Miscellaneous/SLU(release cpu) - [OK] 477.53sec

1/1 tests passed, 0 failed

```

#### 16.Examples/Text/PennTreebank/RNN

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Text/PennTreebank/RNN"

```

\* Test results

```

Running test Examples/Text/PennTreebank/RNN (release cpu) - [FAILED] 33.82 sec

[FAILED] Epochs (with low train loss) must be finished with expected results

...(omitted)

```

\* Cause of error:

I didn't see any errors in the log, it just showed that the result was wrong.

#### 17. Simple2d/MultiGpu

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Simple2d/MultiGpu"

```

\* Test results

```

Running test Simple2d/MultiGpu(release cpu) - [OK] 35.55sec

1/1 tests passed, 0 failed

```

#### 18. Simple2d/OneHidden

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Simple2d/OneHidden"

```

\* Test results

```

Running test Simple2d/OneHidden(release cpu) - [OK] 38.12sec

1/1 tests passed, 0 failed

```

#### 19. Speech/HTKDeserializers/LSTM/FullUtterance

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/HTKDeserializers/LSTM/FullUtterance"

```

\* Test results

```

Running test Speech/HTKDeserializers/LSTM/FullUtterance(release cpu) - [OK] 813.35sec

1/1 tests passed, 0 failed

```

#### 20. Speech/HTKDeserializers/LSTM/FullUtteranceBinaryMLF

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/HTKDeserializers/LSTM/FullUtteranceBinaryMLF"

```

\* Test results

```

Running test Speech/HTKDeserializers/LSTM/FullUtteranceBinaryMLF(release cpu) - [OK] 822.07sec

1/1 tests passed, 0 failed

```

#### 21. Speech/LSTM/FullUtterance

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/LSTM/FullUtterance"

```

\* Test results

```

Running test Speech/LSTM/FullUtterance (release cpu) - [FAILED] 0.14 sec

[FAILED] Must train epochs in exactly same order and parameters

2 expected lines weren't observed in the output.

First unmatched: 12/15/2016 08:46:19: Starting Epoch 1: learning rate per sample = 0.025000 effective momentum = 0.000000 momentum as time constant = 0.0 samples

[FAILED] Epochs must be finished with expected results

2 expected lines weren't observed in the output.

First unmatched: 12/15/2016 08:46:33: Finished Epoch[ 1 of 2]: [Training] ce = 4.60596459 \* 2584; err = 0.90866873 \* 2584; totalSamplesSeen = 2584; learningRatePerSample = 0.025; epochTime=13.9556s

[FAILED] Per-minibatch training results must match

8 expected lines weren't observed in the output.

First unmatched: 12/15/2016 08:46:23: Epoch[ 1 of 2]-Minibatch[ 1- 1, 0.78%]: ce = 4.88277172 \* 886; err = 0.99548533 \* 886; time = 4.1250s; samplesPerSecond = 214.8

See log file for details: /root/jingpeng/package/cntk/../logs/Speech/LSTM\_FullUtterance@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

#### 22. UnitTests/CNTKv2Library

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME UnitTests/CNTKv2Library"

```

\* Test results

```

Running test UnitTests/CNTKv2Library (release cpu) - [FAILED] 299.31 sec

[FAILED] Exit code must be 0

==> got exit code 201 when running: sudo -E bash -c /root/jingpeng/package/cntk/Tests/EndToEndTests/UnitTests/CNTKv2Library/run-test 2>&1

See log file for details: /root/jingpeng/package/cntk/../logs/UnitTests\_CNTKv2Library@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

\* Cause of error:

One of the test samples failed. After checking run-test, I found that it was because this test sample could not be run.

```

$TEST\_BIN\_DIR/$TEST\_BINARY --report\_level=detailed --run\_test=SerializationSuite/LearnerSerializationBackcompat

```

It will read the data in ./cntk/Tests/UnitTests/V2LibraryTests/data/learner.checkpoint.backcompat.bin, and the error will occur if the data is not read correctly.

#### 23. CNTKv2Python/Doc

\* Test command

```

./perf\_exam.sh "PYTHONPATH=${CNTK\_HOME}/bindings/python $PY37/bin/pytest --verbose --deviceid cpu $CNTK\_HOME/bindings/python/doc"

```

\* Test results

```

platform linux -- Python 3.7.12, pytest-7.4.4, pluggy-1.2.0 -- /root/jingpeng/package/cntk2.6\_py3.7/bin/python3.7

cachedir: .pytest\_cache

rootdir: /root/jingpeng/package/cntk/bindings/python/doc

configfile: pytest.ini

plugins: anyio-3.7.1

collected 4 items

cntk/bindings/python/doc/concepts.rst::concepts.rst PASSED [25%]

cntk/bindings/python/doc/gettingstarted.rst::gettingstarted.rst PASSED [50%]

cntk/bindings/python/doc/test/simplenet\_test.py::test\_ffnet\_error[-1] PASSED [ 75%]

cntk/bindings/python/doc/test/simplernn\_test.py::test\_rnn\_error[-1] PASSED [100%]

================================== warnings summary ===================================

cntk/bindings/python/cntk/cntk\_py\_init.py:56

/root/jingpeng/package/cntk/bindings/python/cntk/cntk\_py\_init.py:56: UserWarning: Unsupported Linux distribution ("openeuler"-0;31). CNTK supports Ubuntu 16.04 and above, only.

warnings.warn('Unsupported Linux distribution (%s-%s). CNTK supports Ubuntu 16.04 and above, only.' % (\_\_my\_distro\_\_, \_\_my\_distro\_ver\_\_))

cntk/bindings/python/cntk/cntk\_py\_init.py:90

/root/jingpeng/package/cntk/bindings/python/cntk/cntk\_py\_init.py:90: UserWarning:

############################################ Missing optional dependency ( MKL ) ############################################

CNTK may crash if the component that depends on those dependencies is loaded.

Visit https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#mkl for more information.

############################################################################################################################################

warnings.warn(WARNING\_MSG % (' MKL ', 'https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#mkl'))

cntk/bindings/python/cntk/cntk\_py\_init.py:98

/root/jingpeng/package/cntk/bindings/python/cntk/cntk\_py\_init.py:98: UserWarning:

############################################# Missing optional dependency (GPU-Specific) #############################################

CNTK may crash if the component that depends on those dependencies is loaded.

Visit https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#optional-gpu-specific-packages for more information.

############################################################################################################################################

If you intend to use CNTK without GPU support, you can ignore the (likely) GPU-specific warning!

############################################################################################################################################

warnings.warn(WARNING\_MSG\_GPU\_ONLY % ('GPU-Specific', 'https://docs.microsoft.com/en-us/cognitive-toolkit/Setup-Linux-Python#optional-gpu-specific-packages'))

cntk/bindings/python/cntk/default\_options.py:89: 14 warnings

concepts.rst: 9 warnings

gettingstarted.rst: 5 warnings

test/simplenet\_test.py: 14 warnings

test/simplernn\_test.py: 27 warnings

/root/jingpeng/package/cntk/bindings/python/cntk/default\_options.py:89: DeprecationWarning: inspect.getargspec() is deprecated since Python 3.0, use inspect.signature() or inspect.getfullargspec()

args, \_, \_, \_ = getargspec(function\_or\_class) if isfunction(function\_or\_class) else getargspec(function\_or\_class.\_\_init\_\_)

concepts.rst::concepts.rst

/root/jingpeng/package/cntk/bindings/python/cntk/internal/sanitize.py:286: DeprecationWarning: Using or importing the ABCs from 'collections' instead of from 'collections.abc' is deprecated since Python 3.3, and in 3.9 it will stop working

if isinstance(arg, collections.Iterable):

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html

```

\* Cause of error:

```

arm can run

The riscv server is because pandas cannot be installed

```

#### 24. CNTKv2Python/ModuleTests

\* Test command

```

source cntk2.6\_py3.7/bin/activate

export PYTHONPATH=$CNTK\_HOME/bindings/python:$PYTHONPATH

export LD\_LIBRARY\_PATH=$CNTK\_HOME/bindings/python/cntk/libs:$LD\_LIBRARY\_PATH

./perf\_exam.sh "PYTHONPATH=${CNTK\_HOME}/bindings/python $PY37/bin/pytest --verbose --deviceid cpu $(python -c 'import cntk, os, sys; sys.stdout.write(os.path.dirname(os.path.abspath(cntk.\_\_file\_\_))))')"

```

\* Test results

```

cntk/bindings/python/cntk/tests/persist\_test.py::test\_load\_save\_inputs PASSED [94%]

cntk/bindings/python/cntk/tests/persist\_test.py::test\_load\_save\_unique\_input PASSED [94%]

cntk/bindings/python/cntk/tests/persist\_test.py::test\_large\_model\_serialization\_float sudo: Killed

```

\* Cause of error:

```

The Arm server did not have enough memory (only 16GB) and was killed by the system.

The riscv server is because pandas cannot be installed

```

#### 25. EvalClientTests/CNTKLibraryCSEvalExamplesTest

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs EvalClientTests/CNTKLibraryCSEvalExamplesTest"

```

\* Test result (actually failed)

```

Running test EvalClientTests/CNTKLibraryCSEvalExamplesTest (release cpu py) - [OK] 0.04 sec

1/1 tests passed, 0 failed

```

\* Cause of error:

```

It can only run in python 3.5. The correct display is because the python version is wrong and it returns directly

```

#### 26. EvalClientTests/JavaEvalTest

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME EvalClientTests/JavaEvalTest"

```

\* Test results

```

Running test EvalClientTests/JavaEvalTest (release cpu) - [FAILED] 0.04 sec

[FAILED] Exit code must be 0

==> got exit code 1 when running: sudo -E bash -c /root/jingpeng/package/cntk/Tests/EndToEndTests/EvalClientTests/JavaEvalTest/run-test 2>&1

See log file for details: /root/jingpeng/package/cntk/../logs/EvalClientTests\_JavaEvalTest@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

\* Cause of error:

```

JAVA test sample, java is not installed

```

#### 27. Speech/LSTM/Truncated

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/LSTM/Truncated"

```

\* Test results

```

Running test Speech/LSTM/Truncated(release cpu) - [OK] 12021.59sec

1/1 tests passed, 0 failed

```

#### 28. ParallelTraining/NoQuantization/DoublePrecision

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs ParallelTraining/NoQuantization/DoublePrecision"

```

\* Test results

```

Running test ParallelTraining/NoQuantization/DoublePrecision(release cpu) - [OK] 28.91sec

1/1 tests passed, 0 failed

```

#### 29. ParallelTraining/NoQuantization/SinglePrecision

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs ParallelTraining/NoQuantization/SinglePrecision"

```

\* Test results

```

Running test ParallelTraining/NoQuantization/SinglePrecision(release cpu) - [OK] 27.32sec

1/1 tests passed, 0 failed

```

#### 30. Speech/DNN/Dropout

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/DNN/Dropout"

```

\* Test results

```

Running test Speech/DNN/Dropout (release cpu) - [FAILED] 939.22 sec

[FAILED] Epochs must be finished with expected results for each MPI Rank

```

\* Cause of error:

[[CNTK test #3, solving the segmentation problem (related to the number of mpi instances)]]

#### 31. Speech/DNN/ParallelCrossValidation

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/DNN/ParallelCrossValidation"

```

\* Test results

```

Running test Speech/DNN/ParallelCrossValidation (release cpu) - [FAILED] 268.47 sec

[FAILED] DataParallelSGD training parameters must match for each MPI Rank

6 expected lines weren't observed in the output.

First unmatched: MPI Rank 0: 12/12/2017 15:24:05: Starting minibatch loop, DataParallelSGD training (myRank = 0, numNodes = 2, numGradientBits = 64), distributed reading is ENABLED.

See log file for details: /root/jingpeng/package/cntk/../logs/Speech/DNN\_ParallelCrossValidation@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

\* Cause of error:

[[CNTK test #3, solving the segmentation problem (related to the number of mpi instances)]]

#### 32. Speech/DNN/ParallelNoQuantization

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/DNN/ParallelNoQuantization"

```

\* Test results

```

Running test Speech/DNN/ParallelNoQuantization(release cpu) - [OK] 295.99sec

1/1 tests passed, 0 failed

```

#### 33. Speech/DNN/ParallelNoQuantizationBufferedAsyncGradientAggregation

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/DNN/ParallelNoQuantizationBufferedAsyncGradientAggregation"

```

\* Test results

```

Running test Speech/DNN/ParallelNoQuantizationBufferedAsyncGradientAggregation(release cpu) - [OK] 378.81sec

1/1 tests passed, 0 failed

```

#### 34. Speech/DNN/SaveBestModelPerCriterion

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/DNN/SaveBestModelPerCriterion"

```

\* Test results

```

Running test Speech/DNN/SaveBestModelPerCriterion (release cpu) - [FAILED] 42.43 sec

[FAILED] Must train epochs in exactly same order and parameters for each MPI Rank

...(omitted)

```

\* Cause of error:

[[CNTK test #3, solving the segmentation problem (related to the number of mpi instances)]]

#### 35. Speech/HTKDeserializers/DNN/Dropout

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/HTKDeserializers/DNN/Dropout"

```

\* Test results

```

Running test Speech/HTKDeserializers/DNN/Dropout (release cpu) - [FAILED] 92.86 sec

[FAILED] Must train epochs in exactly same order and parameters for each MPI Rank

...(omitted)

```

\* Cause of error:

[[CNTK test #3, solving the segmentation problem (related to the number of mpi instances)]]

#### 36. Text/SparseDSSM

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME Text/SparseDSSM"

```

\* Test results

```

Running test Text/SparseDSSM (release cpu) - [FAILED] 0.10 sec

[FAILED] Exit code must be 0

==> got exit code 1 when running: sudo -E bash -c /root/jingpeng/package/cntk/Tests/EndToEndTests/Text/SparseDSSM/run-test 2>&1

See log file for details: /root/jingpeng/package/cntk/../logs/Text\_SparseDSSM@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

\* Cause of error:

Requires NET Framework

#### 37. UnitTests/MultiversoTests

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs UnitTests/MultiversoTests"

```

\* Test results

```

Running test UnitTests/MultiversoTests(release cpu) - [OK] 0.15sec

1/1 tests passed, 0 failed

```

#### 38. Speech/DNN/Parallel1BitQuantization

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/DNN/Parallel1BitQuantization"

```

\* Test results

```

Running test Speech/DNN/Parallel1BitQuantization(release cpu) - [OK] 345.59sec

1/1 tests passed, 0 failed

```

#### 39. Speech/DNN/ParallelBM

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/DNN/ParallelBM"

```

\* Test results

```

Running test Speech/DNN/ParallelBM (release cpu) - [FAILED] 287.98 sec

[FAILED] Epochs must be finished with expected results for each MPI Rank

...(omitted)

```

\* Cause of error:

[[CNTK test #3, solving the segmentation problem (related to the number of mpi instances)]]

#### 40. Speech/DNN/ParallelBufferedAsyncGradientAggregation

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/DNN/ParallelBufferedAsyncGradientAggregation"

```

\* Test results

```

Running test Speech/DNN/ParallelBufferedAsyncGradientAggregation(release cpu) - [OK] 284.14 sec

1/1 tests passed, 0 failed

```

#### 41. Speech/DNN/PlotDNN

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/DNN/PlotDNN"

```

\* Test results

```

Running test Speech/DNN/PlotDNN(release cpu) - [OK] 47.21sec

1/1 tests passed, 0 failed

```

#### 42. Speech/DNN/WriteCommand

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/DNN/WriteCommand"

```

\* Test results

```

Running test Speech/DNN/WriteCommand (release cpu) - [OK] 3274.46 sec

1/1 tests passed, 0 failed

```

\* Notice:

[[CNTK test #2, HTKMLFReader cannot be used and reports segments fault]]

#### 43. Speech/HTKDeserializers/DNN/ParallelBM

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/HTKDeserializers/DNN/ParallelBM"

```

\* Test results

```

Running test Speech/HTKDeserializers/DNN/ParallelBM (release cpu) - [FAILED] 289.92 sec

[FAILED] Epochs must be finished with expected results for each MPI Rank

...(omitted)

```

\* Cause of error:

[[CNTK test #3, solving the segmentation problem (related to the number of mpi instances)]]

#### 44. Speech/HTKDeserializers/DNN/ParallelBMWithAdjustLR

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/HTKDeserializers/DNN/ParallelBMWithAdjustLR"

```

\* Test results

```

Running test Speech/HTKDeserializers/DNN/ParallelBMWithAdjustLR (release cpu) - [FAILED] 283.51 sec

[FAILED] Epochs must be finished with expected results for each MPI Rank

...(omitted)

```

\* Cause of error:

[[CNTK test #3, solving the segmentation problem (related to the number of mpi instances)]]

#### 45. Speech/HTKDeserializers/DNN/WriteCommand

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/HTKDeserializers/DNN/WriteCommand"

```

\* Test results

```

Running test Speech/HTKDeserializers/DNN/WriteCommand (release cpu) - [OK] 386.60 sec

1/1 tests passed, 0 failed

```

#### 46. Speech/LSTM\_CTC

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/LSTM\_CTC"

```

\* Test results

```

Running test Speech/LSTM\_CTC(release cpu) - [OK] 1.89sec

1/1 tests passed, 0 failed

```

#### 47. Speech/LSTM\_CTC\_MLF

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/LSTM\_CTC\_MLF"

```

\* Test results

```

Running test Speech/LSTM\_CTC\_MLF(release cpu) - [OK] 4.33sec

1/1 tests passed, 0 failed

```

#### 48. Speech/QuickE2E

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/QuickE2E"

```

\* Test results

```

Running test Speech/QuickE2E(release cpu) - [OK] 491.55sec

1/1 tests passed, 0 failed

```

#### 49. Speech/SVD

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/SVD"

```

\* Test results

```

Running test Speech/SVD(release cpu) - [OK] 495.75sec

1/1 tests passed, 0 failed

```

#### 50. Speech/Simple

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Speech/Simple"

```

\* Test results

```

Running test Speech/Simple(release cpu) - [OK] 1.19sec

1/1 tests passed, 0 failed

```

#### 51. Text/IRMetric

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME Text/IRMetric"

```

\* Test results

```

Running test Text/IRMetric (release cpu) - [FAILED] 0.26 sec

[FAILED] Exit code must be 0

==> got exit code 1 when running: sudo -E bash -c /root/jingpeng/package/cntk/Tests/EndToEndTests/Text/IRMetric/run-test 2>&1

See log file for details: /root/jingpeng/package/cntk/../logs/Text\_IRMetric@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

\* Cause of error:

[[CNTK test #5, missing external dataset]]

There is a lack of pre-trained models and the corresponding dataset is not found.

#### 52. Text/SLU

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Text/SLU"

```

\* Test results

```

Running test Text/SLU(release cpu) - [OK] 24.38sec

1/1 tests passed, 0 failed

```

#### 53. Text/SequenceClassification

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Text/SequenceClassification"

```

\* Test results

```

Running test Text/SequenceClassification(release cpu) - [OK] 41.33sec

1/1 tests passed, 0 failed

```

#### 54. EvalClientTests/CPPEvalClientTest

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME EvalClientTests/CPPEvalClientTest"

```

\* Test results

```

Running test EvalClientTests/CPPEvalClientTest(release cpu) - [OK] 259.71sec

1/1 tests passed, 0 failed

```

#### 55. EvalClientTests/CPPEvalExtendedClientTest

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs -e $DATASET\_HOME EvalClientTests/CPPEvalExtendedClientTest"

```

\* Test results (ARM)

```

Running test EvalClientTests/CPPEvalExtendedClientTest(release cpu) - [OK] 1134.2sec

1/1 tests passed, 0 failed

```

\* Test results (riscv)

```

Running test EvalClientTests/CPPEvalExtendedClientTest (release cpu) - [FAILED] 10772.38 sec

[FAILED] Exit code must be 0

==> got exit code 1 when running: sudo -E bash -c /home/openeuler/jingpeng/package/CNTK/Ts/EndToEndTests/EvalClientTests/CPPEvalExtendedClientTest/run-test 2>&1

See log file for details: /home/openeuler/jingpeng/package/CNTK/../logs/EvalClientTests\_CPPlExtendedClientTest@release\_cpu/output.txt

0/1 tests passed, 1 failed

View logs

'Evaluation failed. EXCEPTION occurred: word not found in source vocab\n'

This means that a certain English word is no longer in the vocab. The reason for this problem has not been found and it only occurs on riscv.

```

#### 56. UnitTests/EvalTests

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs UnitTests/EvalTests"

```

\* Test results

```

Running test UnitTests/EvalTests(release cpu) - [OK] 0.29sec

1/1 tests passed, 0 failed

```

#### 57. UnitTests/V2LibraryCSTests

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs UnitTests/V2LibraryCSTests"

```

\* Test results

```

Running test UnitTests/V2LibraryCSTests (release cpu) - [FAILED] 0.04 sec

[FAILED] Exit code must be 0

==> got exit code 127 when running: sudo -E bash -c /root/jingpeng/package/cntk/Tests/EndToEndTests/UnitTests/V2LibraryCSTests/run-test 2>&1

See log file for details: /root/jingpeng/package/cntk/../logs/UnitTests\_V2LibraryCSTests@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

#### 58. Tutorials/HelloWorld-LogisticRegression

\* Test command

```

cd $CNTK\_HOME/Tutorials/HelloWorld-LogisticRegression

sudo bash $CNTK\_HOME/../performance\_counter.sh "$CNTK\_HOME/build/cpu/release/bin/cntk configFile=lr\_bs.cntk makeMode=false" "$CNTK\_HOME/../results-$(uname -m)"

cd $CNTK\_HOME/..

```

\* Test results

```

parameter1=/root/jingpeng/package/cntk/build/cpu/release/bin/cntk configFile=lr\_bs.cntk makeMode=false

file name: cntkconfigFile=lr\_bs.cntkmakeMode=false

performance.txt has been deleted

CNTK CNTK\_VERSION\_BANNER (HEAD 10a8ff, Jul 24 2024 05:17:14) at 2024/08/26 03:01:53

/root/jingpeng/package/cntk/build/cpu/release/bin/cntk configFile=lr\_bs.cntk makeMode=false

-------------------------------------------------------------------

Build info:

Built time: Jul 24 2024 05:04:18

Last modified date: Thu Jul 18 03:16:56 2024

Build type: release

Build target: CPU-only

With ASGD: yes

Math lib: openblas

Build Branch: HEAD

Build SHA1: 10a8ffcf50d7b9225f3236ffcfdc422b2014fb92 (modified)

MPI distribution: Open MPI

MPI version: 1.10.3

-------------------------------------------------------------------

##############################################################################

# #

# Train command (train action) #

# #

##############################################################################

Model has 9 nodes. Using CPU.

Training criterion: lr = Logistic

Evaluation criterion: err = SquareError

Training 3 parameters in 2 parameter tensors.

Finished Epoch[ 1 of 50]: [Training] lr = 0.31759290 \* 1000; err = 0.09908523 \* 1000; totalSamplesSeen = 1000; learningRatePerSample = 0.039999999; epochTime=1.25066s

Finished Epoch[ 2 of 50]: [Training] lr = 0.11039351 \* 1000; err = 0.02357974 \* 1000; totalSamplesSeen = 2000; learningRatePerSample = 0.039999999; epochTime=1.20542s

Finished Epoch[ 3 of 50]: [Training] lr = 0.08720607 \* 1000; err = 0.01866767 \* 1000; totalSamplesSeen = 3000; learningRatePerSample = 0.039999999; epochTime=1.24132s

Finished Epoch[ 4 of 50]: [Training] lr = 0.07586163 \* 1000; err = 0.01674400 \* 1000; totalSamplesSeen = 4000; learningRatePerSample = 0.039999999; epochTime=1.22528s

Finished Epoch[ 5 of 50]: [Training] lr = 0.06810059 \* 1000; err = 0.01533284 \* 1000; totalSamplesSeen = 5000; learningRatePerSample = 0.039999999; epochTime=1.19227s

Finished Epoch[ 6 of 50]: [Training] lr = 0.06305315 \* 1000; err = 0.01422650 \* 1000; totalSamplesSeen = 6000; learningRatePerSample = 0.039999999; epochTime=1.23501s

Finished Epoch[ 7 of 50]: [Training] lr = 0.06117695 \* 1000; err = 0.01446881 \* 1000; totalSamplesSeen = 7000; learningRatePerSample = 0.039999999; epochTime=1.22961s

Finished Epoch[ 8 of 50]: [Training] lr = 0.05866559 \* 1000; err = 0.01376450 \* 1000; totalSamplesSeen = 8000; learningRatePerSample = 0.039999999; epochTime=1.26519s

Finished Epoch[ 9 of 50]: [Training] lr = 0.05701302 \* 1000; err = 0.01382473 \* 1000; totalSamplesSeen = 9000; learningRatePerSample = 0.039999999; epochTime=1.20164s

Finished Epoch[10 of 50]: [Training] lr = 0.05424744 \* 1000; err = 0.01259778 \* 1000; totalSamplesSeen = 10000; learningRatePerSample = 0.039999999; epochTime=1.21685s

Finished Epoch[11 of 50]: [Training] lr = 0.05344507 \* 1000; err = 0.01313297 \* 1000; totalSamplesSeen = 11000; learningRatePerSample = 0.039999999; epochTime=1.22768s

Finished Epoch[12 of 50]: [Training] lr = 0.05381976 \* 1000; err = 0.01307356 \* 1000; totalSamplesSeen = 12000; learningRatePerSample = 0.039999999; epochTime=1.24027s

Finished Epoch[13 of 50]: [Training] lr = 0.05037591 \* 1000; err = 0.01223938 \* 1000; totalSamplesSeen = 13000; learningRatePerSample = 0.039999999; epochTime=1.23706s

Finished Epoch[14 of 50]: [Training] lr = 0.04970043 \* 1000; err = 0.01228315 \* 1000; totalSamplesSeen = 14000; learningRatePerSample = 0.039999999; epochTime=1.25965s

Finished Epoch[15 of 50]: [Training] lr = 0.04974930 \* 1000; err = 0.01263379 \* 1000; totalSamplesSeen = 15000; learningRatePerSample = 0.039999999; epochTime=1.27026s

Finished Epoch[16 of 50]: [Training] lr = 0.04806321 \* 1000; err = 0.01187134 \* 1000; totalSamplesSeen = 16000; learningRatePerSample = 0.039999999; epochTime=1.2452s

Finished Epoch[17 of 50]: [Training] lr = 0.04796240 \* 1000; err = 0.01172756 \* 1000; totalSamplesSeen = 17000; learningRatePerSample = 0.039999999; epochTime=1.22796s

Finished Epoch[18 of 50]: [Training] lr = 0.04780054 \* 1000; err = 0.01239561 \* 1000; totalSamplesSeen = 18000; learningRatePerSample = 0.039999999; epochTime=1.22069s

Finished Epoch[19 of 50]: [Training] lr = 0.04736344 \* 1000; err = 0.01195484 \* 1000; totalSamplesSeen = 19000; learningRatePerSample = 0.039999999; epochTime=1.20685s

Finished Epoch[20 of 50]: [Training] lr = 0.04651536 \* 1000; err = 0.01168596 \* 1000; totalSamplesSeen = 20000; learningRatePerSample = 0.039999999; epochTime=1.22688s

Finished Epoch[21 of 50]: [Training] lr = 0.04548656 \* 1000; err = 0.01136616 \* 1000; totalSamplesSeen = 21000; learningRatePerSample = 0.039999999; epochTime=1.20188s

Finished Epoch[22 of 50]: [Training] lr = 0.04701764 \* 1000; err = 0.01205853 \* 1000; totalSamplesSeen = 22000; learningRatePerSample = 0.039999999; epochTime=1.22364s

Finished Epoch[23 of 50]: [Training] lr = 0.04807642 \* 1000; err = 0.01247087 \* 1000; totalSamplesSeen = 23000; learningRatePerSample = 0.039999999; epochTime=1.25515s

Finished Epoch[24 of 50]: [Training] lr = 0.04654221 \* 1000; err = 0.01212249 \* 1000; totalSamplesSeen = 24000; learningRatePerSample = 0.039999999; epochTime=1.18288s

Finished Epoch[25 of 50]: [Training] lr = 0.04519672 \* 1000; err = 0.01188752 \* 1000; totalSamplesSeen = 25000; learningRatePerSample = 0.039999999; epochTime=1.18765s

Finished Epoch[26 of 50]: [Training] lr = 0.04554772 \* 1000; err = 0.01212742 \* 1000; totalSamplesSeen = 26000; learningRatePerSample = 0.039999999; epochTime=1.23903s

Finished Epoch[27 of 50]: [Training] lr = 0.04717724 \* 1000; err = 0.01232804 \* 1000; totalSamplesSeen = 27000; learningRatePerSample = 0.039999999; epochTime=1.22222s

Finished Epoch[28 of 50]: [Training] lr = 0.04635520 \* 1000; err = 0.01231099 \* 1000; totalSamplesSeen = 28000; learningRatePerSample = 0.039999999; epochTime=1.22707s

Finished Epoch[29 of 50]: [Training] lr = 0.04359039 \* 1000; err = 0.01143182 \* 1000; totalSamplesSeen = 29000; learningRatePerSample = 0.039999999; epochTime=1.20733s

Finished Epoch[30 of 50]: [Training] lr = 0.04405872 \* 1000; err = 0.01164983 \* 1000; totalSamplesSeen = 30000; learningRatePerSample = 0.039999999; epochTime=1.2291s

Finished Epoch[31 of 50]: [Training] lr = 0.04420972 \* 1000; err = 0.01164209 \* 1000; totalSamplesSeen = 31000; learningRatePerSample = 0.039999999; epochTime=1.23021s

Finished Epoch[32 of 50]: [Training] lr = 0.04337909 \* 1000; err = 0.01130067 \* 1000; totalSamplesSeen = 32000; learningRatePerSample = 0.039999999; epochTime=1.22856s

Finished Epoch[33 of 50]: [Training] lr = 0.04398178 \* 1000; err = 0.01223733 \* 1000; totalSamplesSeen = 33000; learningRatePerSample = 0.039999999; epochTime=1.22886s

Finished Epoch[34 of 50]: [Training] lr = 0.04342689 \* 1000; err = 0.01140238 \* 1000; totalSamplesSeen = 34000; learningRatePerSample = 0.039999999; epochTime=1.25596s

Finished Epoch[35 of 50]: [Training] lr = 0.04300383 \* 1000; err = 0.01094254 \* 1000; totalSamplesSeen = 35000; learningRatePerSample = 0.039999999; epochTime=1.22165s

Finished Epoch[36 of 50]: [Training] lr = 0.04331202 \* 1000; err = 0.01136943 \* 1000; totalSamplesSeen = 36000; learningRatePerSample = 0.039999999; epochTime=1.21545s

Finished Epoch[37 of 50]: [Training] lr = 0.04345496 \* 1000; err = 0.01147922 \* 1000; totalSamplesSeen = 37000; learningRatePerSample = 0.039999999; epochTime=1.19191s

Finished Epoch[38 of 50]: [Training] lr = 0.04424128 \* 1000; err = 0.01172341 \* 1000; totalSamplesSeen = 38000; learningRatePerSample = 0.039999999; epochTime=1.24145s

Finished Epoch[39 of 50]: [Training] lr = 0.04669956 \* 1000; err = 0.01262951 \* 1000; totalSamplesSeen = 39000; learningRatePerSample = 0.039999999; epochTime=1.22085s

Finished Epoch[40 of 50]: [Training] lr = 0.04297209 \* 1000; err = 0.01148758 \* 1000; totalSamplesSeen = 40000; learningRatePerSample = 0.039999999; epochTime=1.21496s

Finished Epoch[41 of 50]: [Training] lr = 0.04553094 \* 1000; err = 0.01266350 \* 1000; totalSamplesSeen = 41000; learningRatePerSample = 0.039999999; epochTime=1.199s

Finished Epoch[42 of 50]: [Training] lr = 0.04287576 \* 1000; err = 0.01152806 \* 1000; totalSamplesSeen = 42000; learningRatePerSample = 0.039999999; epochTime=1.22974s

Finished Epoch[43 of 50]: [Training] lr = 0.04388394 \* 1000; err = 0.01206369 \* 1000; totalSamplesSeen = 43000; learningRatePerSample = 0.039999999; epochTime=1.22539s

Finished Epoch[44 of 50]: [Training] lr = 0.04223350 \* 1000; err = 0.01105061 \* 1000; totalSamplesSeen = 44000; learningRatePerSample = 0.039999999; epochTime=1.23751s

Finished Epoch[45 of 50]: [Training] lr = 0.04207988 \* 1000; err = 0.01140505 \* 1000; totalSamplesSeen = 45000; learningRatePerSample = 0.039999999; epochTime=1.2222s

Finished Epoch[46 of 50]: [Training] lr = 0.04261599 \* 1000; err = 0.01158317 \* 1000; totalSamplesSeen = 46000; learningRatePerSample = 0.039999999; epochTime=1.22364s

Finished Epoch[47 of 50]: [Training] lr = 0.04326449 \* 1000; err = 0.01164270 \* 1000; totalSamplesSeen = 47000; learningRatePerSample = 0.039999999; epochTime=1.23767s

Finished Epoch[48 of 50]: [Training] lr = 0.04225181 \* 1000; err = 0.01148765 \* 1000; totalSamplesSeen = 48000; learningRatePerSample = 0.039999999; epochTime=1.22329s

Finished Epoch[49 of 50]: [Training] lr = 0.04173198 \* 1000; err = 0.01124937 \* 1000; totalSamplesSeen = 49000; learningRatePerSample = 0.039999999; epochTime=1.20538s

Finished Epoch[50 of 50]: [Training] lr = 0.04399340 \* 1000; err = 0.01202173 \* 1000; totalSamplesSeen = 50000; learningRatePerSample = 0.039999999; epochTime=1.23312s

##############################################################################

# #

# Output command (write action) #

# #

##############################################################################

Minibatch[0]: ActualMBSize = 500

Written to LR.txt\*

Total Samples Evaluated = 500

##############################################################################

# #

# DumpNodeInfo command (dumpNode action) #

# #

##############################################################################

Warning: node name '\_\_AllNodes\_\_' does not exist in the network. dumping all nodes instead.

##############################################################################

# #

# Test command (test action) #

# #

##############################################################################

evalNodeNames are not specified, using all the default evalnodes and training criterion nodes.

Final Results: Minibatch[1-1]: err = 0.00685278 \* 500; lr = 0.02953914 \* 500

COMPLETED.

```

#### 59. Tutorials/ImageHandsOn

\* Test command

```

cd $CNTK\_HOME/Tutorials/ImageHandsOn

sudo bash $CNTK\_HOME/../performance\_counter.sh "$CNTK\_HOME/build/cpu/release/bin/cntk configFile=ImageHandsOn.cntk" "$CNTK\_HOME/../results-$(uname -m)"

cd $CNTK\_HOME/..

```

\* Test results

```

/root/jingpeng/package/cntk/build/cpu/release/bin/cntk configFile=ImageHandsOn.cntk

-------------------------------------------------------------------

Build info:

Built time: Jul 24 2024 05:04:18

Last modified date: Thu Jul 18 03:16:56 2024

Build type: release

Build target: CPU-only

With ASGD: yes

Math lib: openblas

Build Branch: HEAD

Build SHA1: 10a8ffcf50d7b9225f3236ffcfdc422b2014fb92 (modified)

MPI distribution: Open MPI

MPI version: 1.10.3

-------------------------------------------------------------------

##############################################################################

# #

# TrainConvNet command (train action) #

# #

##############################################################################

Node 'z.l1.W' (LearnableParameter operation) operation: Tensor shape was inferred as [5 x 5 x 3 x 32].

Node 'z.l2.W' (LearnableParameter operation) operation: Tensor shape was inferred as [5 x 5 x 32 x 32].

Node 'z.l3.W' (LearnableParameter operation) operation: Tensor shape was inferred as [5 x 5 x 32 x 64].

Node 'z.d1.arrayOfFunctions[0].W' (LearnableParameter operation) operation: Tensor shape was inferred as [64 x 3 x 3 x 64].

Node 'zzW' (LearnableParameter operation) operation: Tensor shape was inferred as [10 x 64].

Model has 33 nodes. Using CPU.

Training criterion: ce = CrossEntropyWithSoftmax

Evaluation criterion: errs = ClassificationError

Training 116906 parameters in 10 parameter tensors.

Finished Epoch[ 1 of 10]: [Training] ce = 1.74353063 \* 50000; errs = 64.146% \* 50000; totalSamplesSeen = 50000; learningRatePerSample = 0.00015625; epochTime=534.199s

Finished Epoch[ 2 of 10]: [Training] ce = 1.36361359 \* 50000; errs = 49.132% \* 50000; totalSamplesSeen = 100000; learningRatePerSample = 0.00015625; epochTime=517.669s

Finished Epoch[ 3 of 10]: [Training] ce = 1.18153727 \* 50000; errs = 41.690% \* 50000; totalSamplesSeen = 150000; learningRatePerSample = 0.00015625; epochTime=517.927s

Finished Epoch[ 4 of 10]: [Training] ce = 1.08160312 \* 50000; errs = 37.522% \* 50000; totalSamplesSeen = 200000; learningRatePerSample = 0.00015625; epochTime=561.894s

Finished Epoch[ 5 of 10]: [Training] ce = 1.01113477 \* 50000; errs = 35.068% \* 50000; totalSamplesSeen = 250000; learningRatePerSample = 0.00015625; epochTime=562.854s

Finished Epoch[ 6 of 10]: [Training] ce = 0.96082023 \* 50000; errs = 33.070% \* 50000; totalSamplesSeen = 300000; learningRatePerSample = 0.00015625; epochTime=522.041s

Finished Epoch[ 7 of 10]: [Training] ce = 0.92518930 \* 50000; errs = 32.188% \* 50000; totalSamplesSeen = 350000; learningRatePerSample = 0.00015625; epochTime=520.014s

Finished Epoch[ 8 of 10]: [Training] ce = 0.78405242 \* 50000; errs = 26.856% \* 50000; totalSamplesSeen = 400000; learningRatePerSample = 4.6875e-05; epochTime=530.321s

Finished Epoch[ 9 of 10]: [Training] ce = 0.75879445 \* 50000; errs = 26.110% \* 50000; totalSamplesSeen = 450000; learningRatePerSample = 4.6875e-05; epochTime=532.165s

Finished Epoch[10 of 10]: [Training] ce = 0.75268398 \* 50000; errs = 25.824% \* 50000; totalSamplesSeen = 500000; learningRatePerSample = 4.6875e-05; epochTime=533.65s

##############################################################################

# #

# Eval command (eval action) #

# #

##############################################################################

Final Results: Minibatch[1-625]: errs = 25.240% \* 10000

COMPLETED.

```

#### 60.Tutorials/NumpyInterop

\* Test command

```

./perf\_exam.sh "PYTHONPATH=${CNTK\_HOME}/bindings/python $PY37\_PATH $CNTK\_HOME/Tutorials/NumpyInterop/FeedForwardNet.py"

```

\* Test results

```

Running test Tutorials/NumpyInterop(release cpu) - [OK] 28.43sec

1/1 tests passed, 0 failed

```

#### 61.Tutorials/SLUHandsOn

\* Test command

```

cd $CNTK\_HOME/Tutorials/SLUHandsOn

sudo bash $CNTK\_HOME/../performance\_counter.sh "$CNTK\_HOME/build/cpu/release/bin/cntk configFile=SLUHandsOn.cntk" "$CNTK\_HOME/../results-$(uname -m)"

cd $CNTK\_HOME/..

```

\* Test results

```

Running test Tutorials/SLUHandsOn(release cpu) - [OK] 11379.65sec

1/1 tests passed, 0 failed

```

#### 62.Examples/Text/PennTreebank/RNN

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Text/PennTreebank/RNN"

```

\* Test results

```

Running test Examples/Text/PennTreebank/RNN (release cpu) - [FAILED] 33.33 sec

[FAILED] Epochs (with low train loss) must be finished with expected results

Baseline: 01/10/2018 02:57:02: Finished Epoch[ 1 of 3]: [Training] TrainNodeClassBasedCrossEntropy = 6.84899452 \* 2087; totalSamplesSeen = 2087; learningRatePerSample = 0.1; epochTime=0.600154s

Output: b'08/15/2024 07:23:33: Finished Epoch[ 1 of 3]: [Training] TrainNodeClassBasedCrossEntropy = 6.75345065 \* 2077; totalSamplesSeen = 2077; learningRatePerSample = 0.1; epochTime=0.925764s\n'

Failed pattern: TrainNodeClassBasedCrossEntropy = {{float,tolerance=0.05}}

Baseline: 01/10/2018 02:57:08: Finished Epoch[ 2 of 3]: [Training] TrainNodeClassBasedCrossEntropy = 6.57702282 \* 2099; totalSamplesSeen = 4186; learningRatePerSample = 0.1; epochTime=0.472393s

Output: b'08/15/2024 07:23:42: Finished Epoch[ 2 of 3]: [Training] TrainNodeClassBasedCrossEntropy = 6.79943284 \* 2080; totalSamplesSeen = 4157; learningRatePerSample = 0.1; epochTime=0.687432s\n'

Failed pattern: TrainNodeClassBasedCrossEntropy = {{float,tolerance=0.05}}

Baseline: 01/10/2018 02:57:12: Finished Epoch[ 3 of 3]: [Training] TrainNodeClassBasedCrossEntropy = 6.55210278 \* 2276; totalSamplesSeen = 6462; learningRatePerSample = 0.1; epochTime=0.392923s

Output: b'08/15/2024 07:23:50: Finished Epoch[ 3 of 3]: [Training] TrainNodeClassBasedCrossEntropy = 6.72811865 \* 2369; totalSamplesSeen = 6526; learningRatePerSample = 0.1; epochTime=0.634608s\n'

Failed pattern: TrainNodeClassBasedCrossEntropy = {{float,tolerance=0.05}}

Baseline: 01/10/2018 02:57:17: Finished Epoch[ 3 of 3]: [Training] TrainNodeClassBasedCrossEntropy = 6.30667152 \* 2276; totalSamplesSeen = 6462; learningRatePerSample = 0.050000001; epochTime=0.375603s

Output: b'08/15/2024 07:23:57: Finished Epoch[ 3 of 3]: [Training] TrainNodeClassBasedCrossEntropy = 6.45229766 \* 2369; totalSamplesSeen = 6526; learningRatePerSample = 0.050000001; epochTime=0.631646s\n'

Failed pattern: TrainNodeClassBasedCrossEntropy = {{float,tolerance=0.05}}

...

See log file for details: /root/jingpeng/package/cntk/../logs/Examples/Text/PennTreebank\_RNN@release\_cpu/output.txt

0/1 tests passed, 1 failed

```

\* Cause of error:

The operation results are inconsistent.

#### 63. Examples/Image/Deprecated/CIFAR-10/01\_Convolution

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Image/Deprecated/CIFAR-10/01\_Convolution"

```

\* Test results

```

Running test Examples/Image/Deprecated/CIFAR-10/01\_Convolution(release cpu) - [OK] 181.96sec

1/1 tests passed, 0 failed

```

#### 64. Examples/Image/Deprecated/MNIST/01\_OneHidden\_ndl

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Image/Deprecated/MNIST/01\_OneHidden\_ndl"

```

\* Test results

```

Running test Examples/Image/Deprecated/MNIST/01\_OneHidden\_ndl(release cpu) - [OK] 1488.99sec

1/1 tests passed, 0 failed

```

#### 65. Examples/Image/GettingStarted/01\_OneHidden

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Image/GettingStarted/01\_OneHidden"

```

\* Test results

```

Running test Examples/Image/GettingStarted/01\_OneHidden(release cpu) - [OK] 1.34sec

1/1 tests passed, 0 failed

```

#### 66. Examples/Image/GettingStarted/02\_OneConv

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Image/GettingStarted/02\_OneConv"

```

\* Test results

```

Running test Examples/Image/GettingStarted/02\_OneConv(release cpu) - [OK] 13.72sec

1/1 tests passed, 0 failed

```

#### 67. Examples/Image/GettingStarted/03\_OneConvDropout

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Image/GettingStarted/03\_OneConvDropout"

```

\* Test results

```

Running test Examples/Image/GettingStarted/03\_OneConvDropout(release cpu) - [OK] 14.23sec

1/1 tests passed, 0 failed

```

#### 68. Examples/Image/GettingStarted/05\_OneConvRegr

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Image/GettingStarted/05\_OneConvRegr"

```

\* Test results

```

Running test Examples/Image/GettingStarted/05\_OneConvRegr(release cpu) - [OK] 13.55sec

1/1 tests passed, 0 failed

```

#### 69. Examples/Image/GettingStarted/06\_OneConvRegrMultiNode

\* Test command

```

./perf\_exam.sh "$PY37\_PATH $CNTK\_HOME/Tests/EndToEndTests/TestDriver.py run -d cpu -s cpu -f release -r $CNTK\_HOME/../logs Examples/Image/GettingStarted/06\_OneConvRegrMultiNode"

```

\* Test results

```

Running test Examples/Image/GettingStarted/06\_OneConvRegrMultiNode(release cpu) - [OK] 13.46sec

1/1 tests passed, 0 failed

```