



登临 nnexec 工具使用说明

DL-DG/SW-046A-02

2025-01-07

Copyright©苏州登临科技有限公司，2019 - 2025，版权所有。

未经苏州登临科技有限公司事先书面同意，不得以任何形式或方式复制或传播本文件的任何部分。

商标和许可



和其它苏州登临科技有限公司的其它登临科技的图标为苏州登临科技有限公司的商标。本手册中提及的所有其他商标均为其各自所有者的财产。

通知

所购买的产品、服务和特性由苏州登临科技有限公司与客户签订的合同规定。本文件中描述的所有或部分产品、服务和特性可能不在采购范围或使用范围内。除非合同中另有规定，本文件中的所有声明、信息和建议均按“原样”提供，无任何明示或暗示的保证或陈述。本手册中的信息如有更改，恕不另行通知。本文件在编制过程中已尽一切努力确保内容的准确性，本文件中的所有声明、信息和建议不构成任何明示或暗示的保证。

苏州登临科技有限公司

苏州工业园区扬富路11号南岸新地一期商务楼栋5号1101室，江苏，中国

<http://www.denglin.ai>

email : support@denglin.ai

更新历史

版本	更新描述
02	增加对nnexec工具和 <code>--subGraphsJson</code> 参数的说明。
01	第一次发布。

内容目录

- 1 简介
- 2 参数说明
- 3 日志说明**
- 4 程序exit码说明
- 5 使用示例
 - 5.1 基础功能测试
 - 5.2 性能测试
 - 5.3 模型量化
 - 5.4 结果检查
 - 5.5 动态shape
 - 5.6 Plugin
 - 5.7 序列化和反序列化
 - 5.8 Device设置
- 6 子图切分
- 附录 clusterConfig自动调整规则

1 简介

nnexec工具是登临提供的命令行包装工具，它可以根据随机或者用户输入的数据对网络进行基准测试，并从模型生成序列化引擎和从生成器生成序列化定时缓存。

说明：

nnexec工具需要 source SDK 之后才能使用。

2 参数说明

nnexec工具包含一些必要的参数，本节对这些参数做简要的说明。

1. model

说明：**必传参数**，指定模型路径。

模型类型支持onnx、pb(tensorflow)、rlym。

2. --input

说明：指定模型的输入节点和输入数据（ npy格式 ）。

- 如果未指定输入数据，则由系统随机生成。
- 输入节点之间用','号分割，输入节点与输入数据之间用空格分割。
- 当输入节点不是默认的输入节点时，会切分出子图以满足要求。切分子图功能详见[子图切分](#)章节。
- 如果未指定input节点，则会通过onnx或tensorflow自动查找并设置。
- 如果shape中包含动态shape，则设置动态维度为1并设置 --shape 参数。

示例：

```
--input "input0,input1", --input "input input.npy,input1 input1.npy"
```

3. --output

说明：指定模型的输出节点和输出数据（ npy格式 ）。

- 如果未指定输出数据，并且未设置 --nocheck，则会根据onnxruntime或tensorflow的推理结果作为golden。
- 输出节点之间用','号分割，输出节点与输出数据之间用空格分割。
- 指定输入数据和输出数据（ 设置 --nocheck 选项则无需指定输出数据 ）会跳过onnxruntime或tensorflow推理过程。
- 当输出节点不是默认的输出节点时，会切分出子图以满足要求。切分子图功能详见[子图切分](#)章节。
- 如果未指定output节点，则会通过onnx或tensorflow自动查找并设置。
- 必须通过--input指定输入数据，指定的输出数据才会生效。

示例：

```
--output "output" --output "output output.npy"
```

4. `--shape`

说明：设置动态输入shape。输入节点之间用','号分割。

示例：

```
--shape "input0:1x3x256x256,input1:1x2"
```

5. `--maxBatch`

说明：指定NNE模型构建过程中的 `max_batch_size`。

默认值：32

示例：

```
--maxBatch 64
```

6. `--clusterConfig`

说明：NNE构建context时的cluster配置，值可以为0|1|2|3|01|23|02|13|03|12|0123，支持多选，用','号分割。

当clusterConfig设置的值高于GPU实际cluster数量时，系统将自动调整，调整规则参见附录 [clusterConfig自动调整规则](#)。

默认值：0123

示例：

```
--clusterConfig "01"
```

7. `--device`

说明：设置NNE推理时的device id。

默认值：0

示例：

```
--device 1
```

8. `--batch`

说明：NNE推理时的batch size，复制input数据。

默认值：1

示例：

```
--batch 2
```

9. `--iterations`

说明：设置推理次数。

默认值：1

示例：

```
--iterations 10
```

10. `--warmUp`

说明：设置预热推理次数。

默认值：0

示例：

```
--warmUp 1
```

11. `--async`

说明：设置NNE采用Enqueue异步接口进行推理。

默认值：false

示例：

```
--async
```

12. `--fp16`

说明：float16 量化。

默认值：false

示例：

```
--fp16
```

13. `--quantize`

说明：int8 量化

默认值：false

```
示例: --quantize
```

14. `--saveEngine`

说明：NNE模型序列化保存文件路径。

示例: `--saveEngine "model.engine"`

15. `--loadEngine`

说明：NNE模型反序列化加载的文件路径，需要注意的是还需指定原始模型。

示例: `--loadEngine "model.engine"`

16. `--pluginNneLib`

说明：自定义plugin nne library文件路径，支持传入多个文件路径，用','号分割。

示例: `--pluginNneLib "libnne_plugin1.so,libnne_plugin2.so"`

17. `--pluginTvmLib`

说明：自定义plugin tvm library文件路径，支持传入多个文件路径，用','号分割。

示例: `--pluginTvmLib "libtvm_plugin1.so,libtvm_plugin2.so,"`

18. `--pluginModule`

说明：自定义plugin python前端文件路径，支持传入多个文件路径，用','号分割。

示例: `--pluginModule " front_end1.py,front_end2.py"`

19. `--convert`

说明：将模型convert到rlym格式后再交由NNE推理。

默认值：`false`

示例: `--convert`

20. `--builderFlags`

说明：指定NNE模型构建过程中的builder flags。

flag有SpmAlloc、UserConst、SizIgnoreWeights、ForceUseModulator、Quantize，指定多个时中间用','号分割。

示例: `--builderFlags "SpmAlloc,UserConst"`

21. `--networkConfig`

说明：NNE模型构建设置network SetConfig接口参数，该参数将直接传给接口。

示例：`--networkConfig "--fast-qconvadd=1 --fast-sigmoid-fp32=3"`

22. `--quantizeRegions`

说明：int8量化时设置regions configure配置文件。

示例：`--quantizeRegions regions.json`

23. `--tvmConvertArgs`

说明：指定调用python3 -m dl convert转化模型时需要提供的额外参数。

可以指定的额外参数不包含：`--input-shapes`，`--input-min-shapes`，`--input-opt-shapes`，`--input-max-shapes`，`--data-depend-ops-min-shapes`，`--data-depend-ops-opt-shapes`，`--data-depend-ops-max-shapes`，`--data-dep-ops-json`，`--output-model`

示例：`--tvmConvertArgs "--disabled-pass DISABLED_PASS --pass-config name=value --optimize"`

24. `--tvmQuantizeArgs`

说明：指定调用python3 -m dl quantize量化模型时需要提供的额外参数。

可以指定的额外参数不包含：`--library`，`--module`，`--input-data-dir`，`--downcast`，`--output-model`，`-arch`，`--regions-configure`，`--output-model`

示例：`--tvmQuantizeArgs "--fake-quant --check-acc --dump-regions --quantize_add --cache_size CACHE_SIZE --calibrate-mode disable"`

25. `--callBack`

说明：NNE模型构建时设置callback，只支持MergeNodes和MergeAllNodes两种模式。

- MergeNodes模式：

示例：`1:CU6,CU24,CU25,empty----;2,4:CU266,CU267,CU272,empty----`

其中，':'号前的1/2/4指定不同cluster的数量使用该callBack，不同cluster之间用逗号分割。不指定cluster则应用到全部类型，不同类型配置间用';'号分割。

- MergeAllNodes模式：`"1:@"`，cluster的设置与MergeNodes相同，节点设置为'@'符号即可。

示例：`--callBack "1:@;2,4:CU266,CU267,CU272,empty----"`

26. `--callBackFile`

说明：指定NNE模型构建时的callback。文件中每一行为一个list，list中的节点名称用','号分割。

示例：`--callBackFile callback.txt`

27. `--subGraphsJson`

说明：指定NNE模型解析时的sub-graphs json文件，指定该文件将采用nne parser ParseV2接口解析模型（仅支持V2）。

示例：`--subGraphsJson subgraphs.json`

说明：

Hamming™ V1 SDK Linux 版本暂不支持 `--subGraphsJson`。

28. `--dumpNodeList`

说明：dump nne callback提供的所有node，每个node占一行。该选项只有在未设置callBack和callBackFile选项时生效。

示例：`--dumpNodeList callback.txt`

29. `--dumpOutput`

说明：开启NNE输出dump。

- 输出结果会保存在 output0.npy、output1.npy等文件中，文件名由"output" + 输出id + ".npy"组成。
- 校验golden会保存在 golden0.npy、golden1.npy等文件中，文件名由"golden" + 输出id + ".npy"组成。
- 日志中会打印节点和输出文件的对应关系，日志示例：`@@ [16:53:38][INFO]: user_input -> input0.npy`

默认值：`false`

示例：`--dumpOutput`

30. `--dumpInput`

说明：开启NNE输入dump。

- 输出结果会保存在 input0.npy、input1.npy等文件中，文件名由"input" + 输出id + ".npy"组成。
- 日志中会打印节点和输出文件的对应关系，日志示例：`@@ [16:55:12][INFO]: Sigmoid -> output0.npy`

默认值：`false`

示例：`--dumpInput`

31. `--seed`

说明：随机生成input时的随机种子。

默认值：1

示例：`--seed 2`

32. `--setConstInput`

说明：设置生成input时使用常量。常量数值由 `--inputConst` 指定，支持浮点型数据，默认设置为0，并且 `--seed` 选项失效。

默认值：false

示例：`--setConstInput`

33. `--inputConst`

说明：设置生成input时使用的常量数值。只有设置了 `--setConstInput` 选项时该选项才生效，支持浮点型数据。

默认值：0

示例：`--setConstInput --inputConst 1.0`

34. `--nocheck`

说明：跳过结果检查。

默认值：false

示例：`--nocheck`

35. `--minCosine`

说明：设置结果检查时向量间的最小余弦相似度。

- 该选项设置后将开启余弦相似度校验方式，校验结果pass，case才可以pass。
- 该选项会关闭默认的余弦相似度和max diff校验。

默认值：0.90

示例：`--minCosine 0.95`

36. `--maxDiff`

说明：设置结果检查时的最大允许误差。

- 该选项设置后将开启max diff校验方式，校验结果pass，case才可以pass。
- 该选项会关闭默认的余弦相似度和max diff校验。

默认值：0.01

示例：--maxDiff 0.0001

37. --rtol

说明：设置numpy isclose校验相对公差。

- 该选项设置后将开启isclose校验方式，校验结果pass，case才可以pass。
- 该选项会关闭默认的余弦相似度和max diff校验。

默认值：0.001

示例：--rtol 0.0005

38. --atol

说明：设置numpy isclose校验绝对公差。

- 该选项设置后将开启isclose校验方式，校验结果pass，case才可以pass。
- 该选项会关闭默认的余弦相似度和max diff校验。

默认值：0.001

示例：--atol 0.0005

39. --bitAccurate

说明：设置结果检查时使用bit-accurate校验，校验结果pass，case才可以pass。

该选项会关闭默认的余弦相似度和max diff校验。

默认值：false

示例：--bitAccurate

40. -h, --help

说明：打印参数帮助文档。

示例：--help

3 日志说明

对nnexec日志简要说明如下。

日志格式：@@ [时间戳] [日志等级]<日志类型>：其它信息

示例：

```
@@ [18:05:53][INFO]<nne output tensor info>: 206(Tensor<1, 128, 3520>dtype=float32)*

@@ [18:05:53][WARNING]<tensor compare fail>: tensor_name=206 cosine=0.019601
max_diff=16.7327*

@@ [18:05:53][ERROR]<nne output check>: percision check failed!
```

nnexec日志分为 INFO、WARNIGN、ERROR 三个等级。

- INFO：表示关键信息，用来追踪程序的执行流程。
- WARNIGN：表示存在警告性错误，这些错误并不影响程序执行，但行为可能和预期的不一致，需要检查传入参数。
- ERROR：表示存在严重错误，程序无法正常执行。

4 程序exit码说明

0 表示正常结束。

-1 表示出错，如结果检查未通过等。

5 使用示例

5.1 基础功能测试

1. nnexec model.onnx
2. nnexec model.pb
3. nnexec model.onnx --input "input0,input1" --output "output0"
4. nnexec model.pb --input "input0,input1" --output "output0"
5. nnexec model.onnx --input "input0 input0.npy,input1 input1.npy" --output "output0"
6. nnexec model.onnx --input "input0 input0.npy,input1 input1.npy" --output "output0 output0.npy"
7. nnexec model.pb --input "input0,input1" --output "output0" --dumpOutput
8. nnexec model.onnx --input "input0,input1" --output "output0" --batch 4

5.2 性能测试

```
nnexec model.onnx --input "input0,input1" --output "output0" --warmUp 1 --iterations 10
```

5.3 模型量化

1. nnexec model.onnx --input "input0,input1" --output "output0" --fp16
2. nnexec model.onnx --input "input0,input1" --output "output0" --quantize

5.4 结果检查

默认进行结果检查。

如果需要取消默认结果检查，请参照参数说明，加入--nocheck选项。

1. nnexec model.onnx --input "input0,input1" --output "output0" --minCosine 0.99
2. nnexec model.onnx --input "input0,input1" --output "output0" --maxDiff 0.001
3. nnexec model.onnx --input "input0,input1" --output "output0" --nocheck

5.5 动态shape

```
nnexec model.onnx --input "input0,input1" --output "output0" --shape "input0:1x3x256x256,input1:1x2"
```

5.6 Plugin

```
nnexec model.onnx --input "input0 input0.npy,input1 input1.npy" --output "output0 output0.npy" --pluginNneLib  
libnne_plugin.so --pluginTvmLib libtvm_plugin.so --pluginModule front_end.py
```

5.7 序列化和反序列化

1. nnexec model.onnx --input "input0,input1" --output "output0" --saveEngine "model.engine"
2. nnexec model.onnx --input "input0,input1" --output "output0" --loadEngine "model.engine"

5.8 Device设置

```
nnexec model.onnx --input "input0,input1" --output "output0" --device=1
```

6 子图切分

如果onnx或pb模型不包含任何自定义op，可以在原始框架上正常执行，切分子图时只需要指定子图对应的input和output即可，否则需要指定input对应的npz文件。

登临科技保密材料

附录 clusterConfig自动调整规则

clusterConfig自动调整规则如下表所示：

GPU cluster 数量	clusterConfig 设置值	clusterConfig 转换值
1	0123	0
1	01	0
1	23	0
2	0123	01