

QONNX dev meeting

@FastML for Science Workshop
2023-09-28

Agenda

1. Statistics
2. What's in the repo today?
3. What's new + what's coming
4. What would you like to see in qonnx?
 - a. Poll results, transformation wiki update
5. <insert more items>

Stats as per 2023-09-28

Downloads **71k** from pypi + 5.3k from GitHub clones

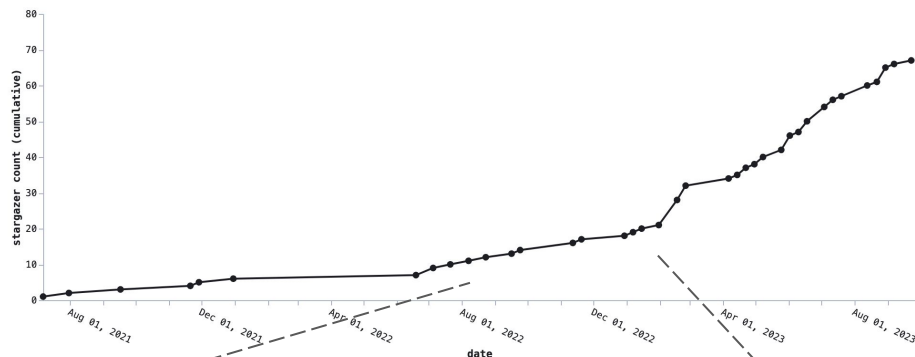
Fork **24**

★ Starred **68**

🔗 3 Open ✓ 55 Closed

Stargazers

Each data point corresponds to at least one stargazer event. The time resolution is one day.



ONNX @onnxai · Aug 15, 2022

Alessandro Pappalardo from @AMD presented QONNX, a proposal for representing arbitrary-precision quantized NNs at the ONNX Community Day:



youtube.com

QONNX: A proposal for representing arbitrary-precision quantized NNs at the ONNX Community Day. We present extensions to the Open Neural Network Exchange (ONNX) intermediate representation ...



2



3



Super PINTO @PINTO03091 · Feb 24

"QONNX: Arbitrary-Precision Quantized Neural Networks in ONNX" Quant, BipolarQuant, Trunc, 1-bit (bipolar) quantization, with scaling and zero-point. ハイボラ量子化? FPGAがターゲットかな。



github.com

GitHub - fastmachinelearning/qonnx: QONNX: Arbitrary-Precision Quantized Neural Networks in ONNX - GitHub - ...



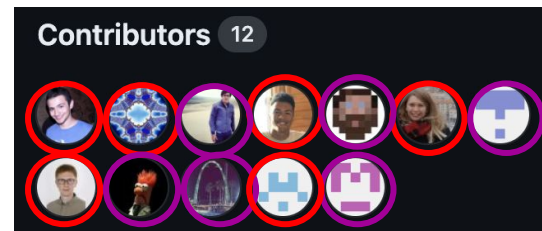
2



7



1,138



8 models in model zoo:
JSC, MNIST, CIFAR-10, ImageNet

What's in the repo today?

```
python.module.for.import  
commandline-util
```

executing QONNX models for (slow) functional verifications	<code>qonnx.core.onnx_exec</code> qonnx-execute
Cleanup: shape inference, constant folding, cleanup and other transformations (~30 different ones)	<code>qonnx.transformation</code> qonnx-cleanup
summarizing the inference cost of a QONNX model in terms of mixed-precision MACs, parameter and activation volume	<code>qonnx.util.inference_cost</code> qonnx-inference-cost
Conversion between different ONNX quantization representations (Quant <-> QCDQ <-> QOp in progress)	<code>qonnx.util.convert</code> qonnx-convert
Python infrastructure for writing transformations and defining executable, shape-inferencable custom ops	<code>qonnx.transformation.base</code> <code>qonnx.custom_op.base</code>

What's new recently merged PRs


Channel pruning utilities ✓

#71 by maltanar was merged last month

Fix (utils): Updating matvec accumulator range calculation ✓

#70 by i-colbert was merged last month

Fix: Adding dropped attributes to transposed convolution op ✓

#69 by i-colbert was merged on Aug 15  2 tasks done


[Convert] simplify zeropoint to scalar in QCDQ2Quant if possible ✓

#68 by maltanar was merged on Aug 3

Feat: Adding quantization support for sub-pixel convolution to deconvolution transformation ✓

#67 by i-colbert was merged on Aug 4


Introduce RemoveUnusedNodes ✓

#66 by maltanar was merged on Jul 19  v0.3.0

Quant/QCDQ converter improvements ✓

#65 by maltanar was merged on Jul 18  v0.3.0

qonnx-exec improvements ✓

#64 by maltanar was merged on Jul 18  v0.3.0

Coming up soon

- QOp conversion (AMD)
- Recurrent nets with Quant (Shashwat/AMD/Trinity College)
- Range analysis (Yaman/AMD)
- Float quantization? (Freddie/Bristol)

What would you like to see in qonnx? #73

maltanar started this conversation in Polls



maltanar 2 weeks ago Maintainer

To help us improve the qonnx repository for the community, please let us know what you'd like to see more of in the feature.

What would you like to see more of in qonnx?

More quantized models in the model zoo 0%

Show-and-tell tutorial examples showing how to implement particular "recipes" 0%

More basic tutorials on how to use ModelWrapper, transformations, etc. 0%

Support for representing other styles of quantization 0%

"Transformation wiki", description of transformations with example pictures of topologies before/after ☺ 100%

QONNX quantization operators as part of the official ONNX standard 0%

4 votes · Hide Results

Vote

↑ 5

Can we do this semi-automatically?

Almost all qonnx transforms have unit tests, they all do something like:

```
new_model = model.transform(BatchNormToAffine())
```

Use [intercepts](#) package to intercept call to transformations of interest in a special script:

```
intercepts.register(ModelWrapper.transform, handler)
handler:
if transformation in [BatchNormToAffine, trafo2...]:
    Save ONNX model before transformation
    Call the transformation
    Save ONNX model after transformation
```

Execute test suite to generate all before/after ONNX pairs + upload to git

Link to before/after files (+Netron?) from wiki, add descriptions manually

Thank you!