# ONNX Roadmap Discussions

AT THE VERY EDGE

Enabling AI

Martin Croome
GreenWaves

GREENWAVES TECHNOLOGIES

## Background and experience with ONNX

- GreenWaves is a French fabless semiconductor manufacturer.
- We focus on ultra low power real time signal processing and neural networks for highly power constrained devices: sensors with a battery life measured in years, hearables and wearables with tiny batteries.
- Our first product GAP8 is in production.
- We have our own toolchain that processes / quantizes / optimizes / analyses graphs for our SoC.
- We import both TFLITE and ONNX graphs so we are very familiar with both

- ONNX: The good
  - Understandable operator set and structure
  - **GREAT** documentation
  - **GREAT** operator versioning system
- ONNX: To be improved
  - Quantization
  - Fusion friendliness

## Quantization (1)

- Current status in ONNX: Mix of Fake Quantization operators and a few quantized operators
- What is the goal?
    - a) Express a quantized ONNX graph directly?
        - We currently support mixed graphs with ieee16 or bfloat16 or 16- or 8-bit fixed point. We have the ability to handle sub-byte quantization for filters. In fixed point we support scaled symmetric or asymmetric quantization. We have customers who have produced their own kernels doing even more specialized things. We have even more specialized hardware blocks that we have no expectation that ONNX will support.
        - Are you going to provide quantized operators for every scheme with every different quantization technique?
    - b) Provide the information necessary for a backend to quantize a graph or execute a prequantized graph?

## Quantization (2)

- For (b) more is needed
  - Parameter statistics are easy (unless the tensors are prequantized)
  - Activation statistics are not - best place to get them is training environment - all training data has been run forward through the graph - Save them with the graph

- Proposal: Add statistics metadata to every (non-constant but why not all) tensor (TFLITE has this now)

- Absolutely necessary
  - Min/Max/Std/Mean
- Nice to have
  - By channel
  - Outlier statistics
  - More??
- Should be easy to strip if ONNX format graph is used by a runtime.

## Optimised kernel friendliness

- Currently:
  - Some fused operators (GRU support is highly appreciated - not in TFLite)
  - Move towards functions for composed operators
- The problem:
  - Decomposed operators are extremely difficult / impossible to recover with full optimisation versus custom kernels - particularly quantized / particularly with custom hardware.
- The solution:
  - Encourage/**force** exporters to wrap the native high level operators that they are exporting in an ONNX function (a function is just a subgraph).
  - The namespace and function name should match the exporting platform function. i.e. “tf.keras.experimental.PeepholeLSTMCell”
  - Reading the ONNX file you can either select an optimised version of the function (fusion) or run the operators that implement the function (subgraph)

# Thank You

GREENWAVES
TECHNOLOGIES