Generalized Machine Learning Quantization Implementation for High Level Synthesis Targeting FPGAs

Matthew Trahms - 15th March, 2022

Outline

- The Large Hadron Collider
- Particle Tracking
- hls4ml
- FINN and Brevitas
- Particle Tracking GNN Quantization Aware Training
- FINN Collaboration and QONNX
- QONNX Ingest into hls4ml
- Future Work
- Acknowledgements

The Large Hadron Collider (LHC)

- Large particle accelerator in Europe
  - Accelerate particles near the speed of light
  - Collisions split atoms into subatomic particles
  - Sensors track particle interactions
- Generates large quantities of data
  - 1 petabyte of data / second while operating
    - Data management challenge
    - Set to increase in the future with the high luminosity upgrade
  - Multiple different experiments
    - ATLAS
    - CMS

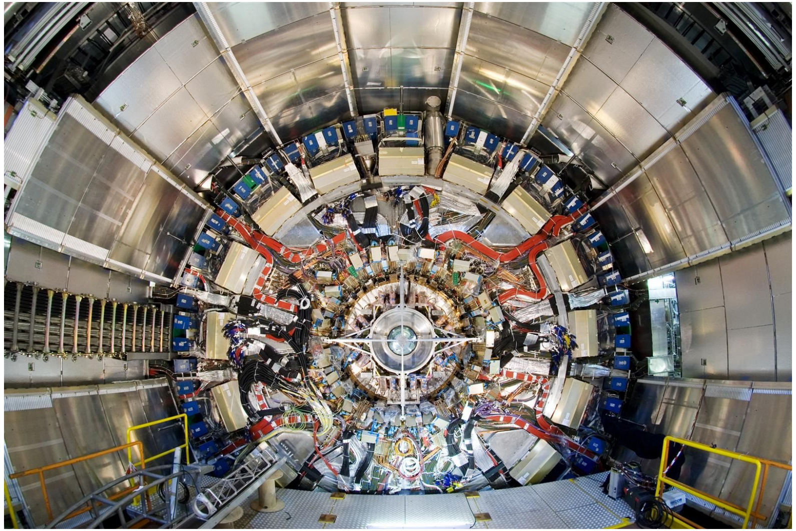

Fig 1. The ATLAS Detector (atlas.cern)

Particle Tracking

- Need to track individual particles from collisions in order to make observations
  - Large number of collisions happening in a small space
    - Need to accurately separate particle paths
  - Non-machine learning algorithm algorithm scales poorly with increasing number of particles
    - Higher luminosity = more particles

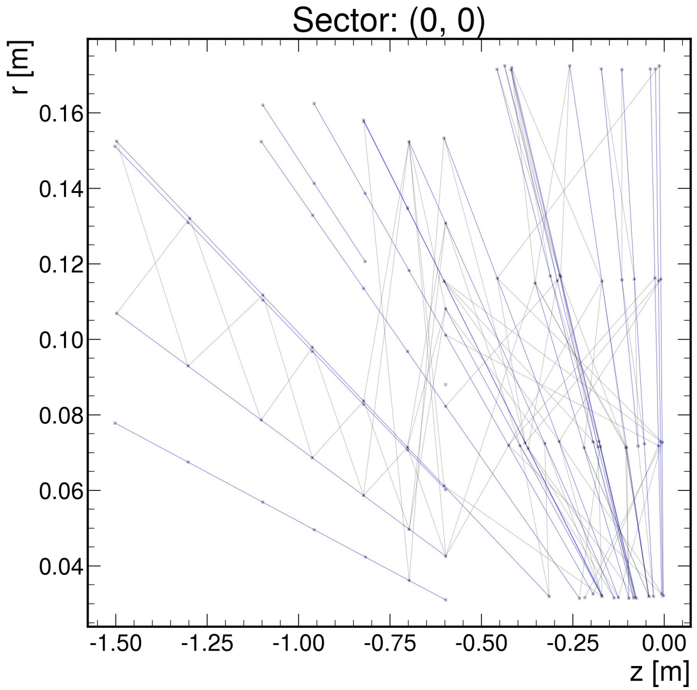

Fig. 2: Possible particle tracks. Blue edges are correct edges while black edges were potential particle paths based on detections. (A. Elabd et al)

Particle Tracking Graph Neural Network (GNN)

- Possible particle tracks represented as directional acyclic graphs
  - Nodes = locations where particles were detected
  - Edges = possible particle paths, determined by distance
- Graph Neural Nets (GNNs) transform graphs
  - Interaction networks determine true edges, or the actual path of the particle
  - Able to match performance of non-machine learning algorithm and scales with luminosity

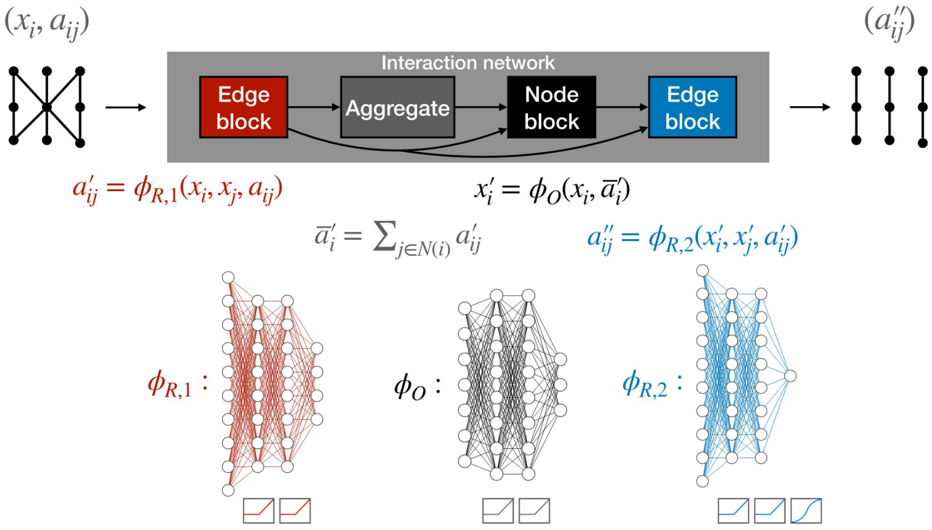

Fig. 3: The structure of an Interaction Network Notice how the graph is transformed at the output (A. Elabd et al)

hls4ml

- Machine learning algorithms work as alternatives for high energy physics applications
  - Need high throughput
  - Field Programmable Gate Array (FPGA): Reprogrammable logic that can be used to implement digital algorithms
    - Generally able to achieve higher throughput than acceleration on CPUs

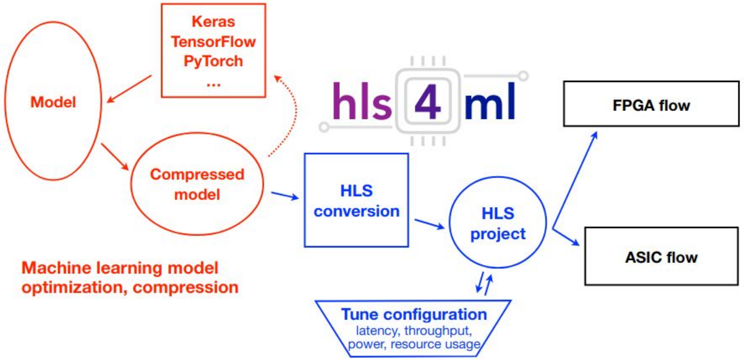

Fig. 4: The hls4ml flow to generate a hardware implementation of a machine learning model (Luca Carloni et al)

FINN and Brevitas

- FINN: hls4ml alternative developed by Xilinx Research
  - Targets extremely low bit width deployment
    - Low latency, high throughput
  - Brevitas: Quantization Aware Training (QAT) library developed for FINN
    - QAT allows for high accuracy at low bit widths compared to post training quantization (PTQ)
    - Based on Pytorch
  - Uses ONNX graphs with custom nodes for internal representation

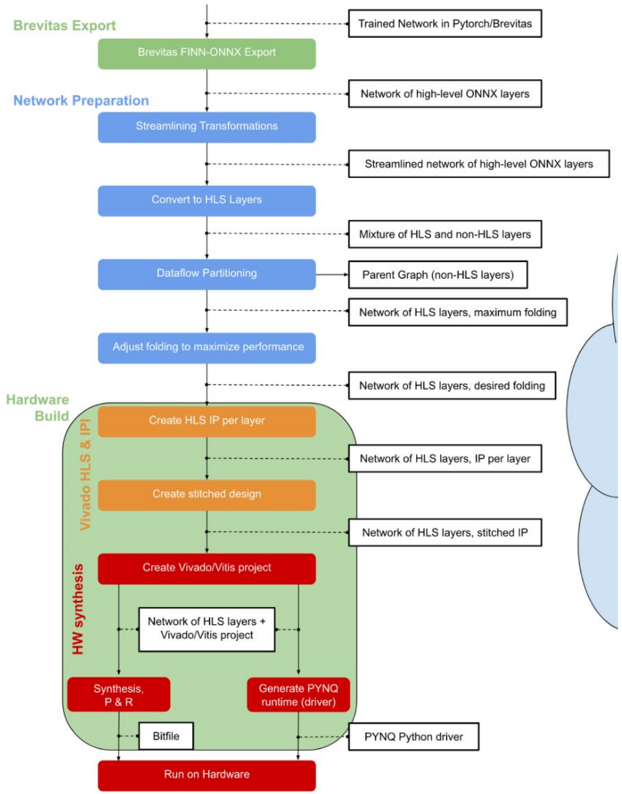

Fig. 5: The FINN Flow, from Brevitas export to deployment (xilinx.github.io)

Tracking GNN Quantization Aware Training

- FPGA acceleration of particle tracking GNN
  - Need for high throughput
  - Network originally implemented in Pytorch
    - Only option for hls4ml is PTQ
    - Loss in accuracy
  - Re-implemented and trained network in Brevitas
    - Layer by layer replacement
    - Retrained on same dataset
  - Achieved near equivalent performance
    - AUC: Area under ROC curve
      - Compares true positive rate and false positive rate

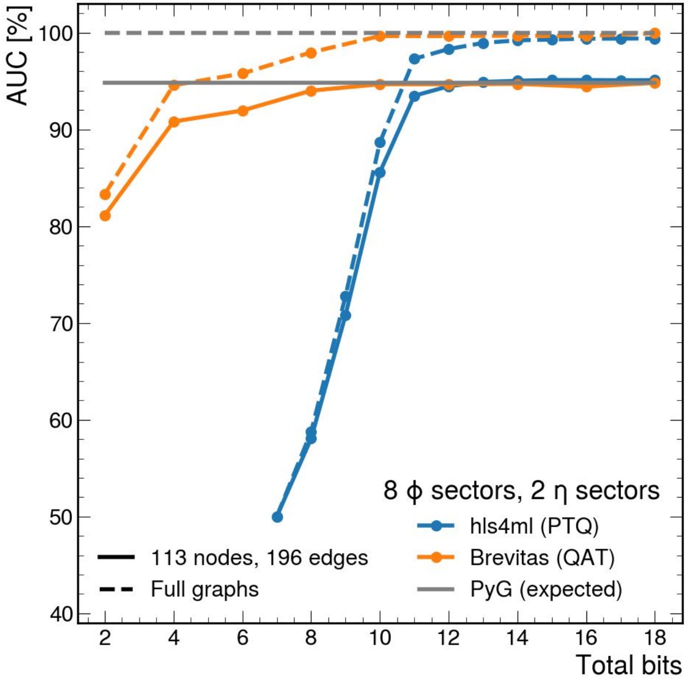

Fig. 6: Tracking GNN QAT Results
(A. Elabd et al)

FINN Collaboration and QONNX

- FINN and hls4ml accomplish similar tasks
  - Cross organizational collaboration - develop a shared model format that can be used by hls4ml and FINN
    - Generalized version of FINN ONNX
    - Extends the ONNX framework to add quant nodes
      - Represents either weight or input quantization
  - Interoperability means that users can choose the solution that works better for their purposes

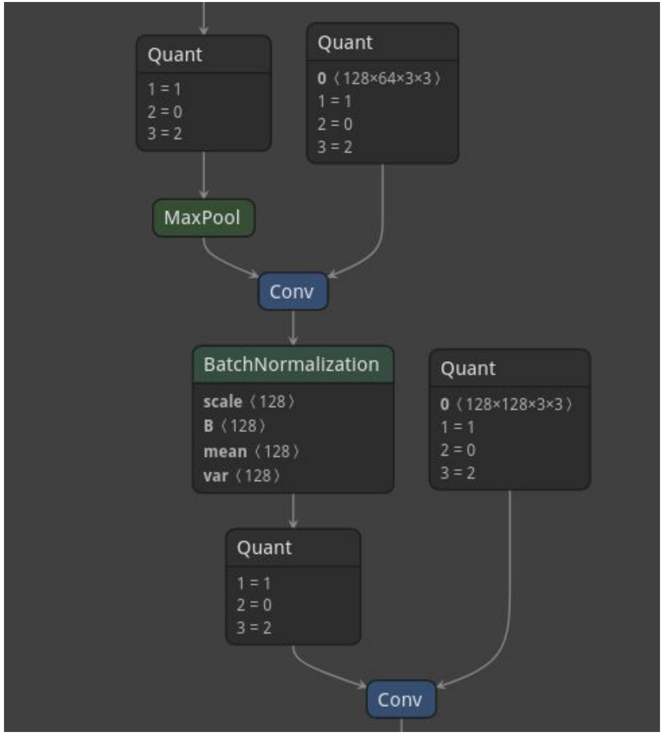

Fig. 7: Visual representation of a QONNX network (netron.app)

QONNX Ingest Into hls4ml

- Need to convert QONNX to HLSModel to synthesize in hls4ml
    - QONNX quant nodes specify quantization
    - HLSModel layers have quantization attributes built into layers
    - Set of transformations
        - Ingest complete structure into HLSModel
        - Incorporate quantizations from Quant nodes into layers
        - Remove Quant nodes

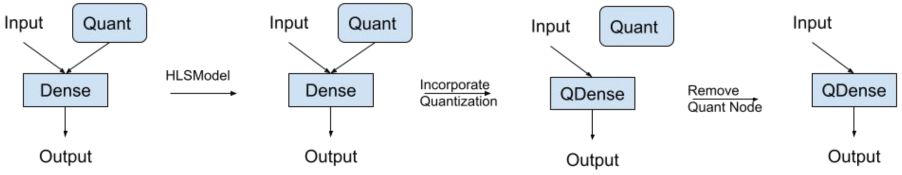

Fig. 8: Process to incorporate QONNX quantizations into HLSModel

Future Work

- QONNX ingest into hls4ml needs further work
  - Bugs with convolutional models and different model architectures
  - Needs to be pulled into the master branch of hls4ml
    - Needs to be converted to the new workflow
  - Need to test latency synthesis implementation
    - Currently testing with resource
  - Need to test with more/different model architectures
- Take a QAT particle tracking GNN through synthesis
  - Only a numerical study currently, need to validate actual performance on an FPGA
- More collaboration between FINN and hls4ml
  - Streamlining in hls4ml
  - MLIR

Acknowledgements

- Dr. Hauck and Dr. Hsu - Thank you for everything, working with you for the past three years has had a drastic impact on my life. I don’t think I would have pursued my graduate degree if I didn’t meet you.
- Dr. Nhan Tran - Thank you for helping me spin up into the greater FastML community to make contributions where I could.
- Dr. Jovan Mitrevski and Dr. Sioni Summers - It has been a pleasure working with you on the QONNX ingest into hls4ml.
- Dr. Yaman Umuroğlu - Thank you for being my point of contact when I was first getting started with FINN.
- Dr. Javier Duarte and Abdelrahman Elabd - Thank you for helping me contribute to the GNN tracking investigation
- My parents, Rob and Lisa Trahms - Thank you for encouraging and supporting me through this process.
- All my friends and labmates in the audience - Thank you for coming here to support me in defense of my thesis.

Questions?