INTEL AI DEVCON 2018

Intel® nGraph Compiler

Scott Cyphers and Jayaram Bobba, Intel

- Manually optimizing hardware through a framework is **not efficient**, and prone to **buggy execution**.
- Data scientists shouldn't have to mess with low-level machine code; they should be able to compile and run their data science models on any device.
- Frameworks designed to train large datasets are not inherently optimized for inference, and vice versa. *Adaptability matters in AI.*

nGraph Compiler to the Rescue

Value Proposition: To provide standard and custom DL frameworks with the most developer-friendly library and compiler suite for training and inference models.

Open Source : github.com/NervanaSystems/ngraph

Targeted

Frameworks

- Apache MXNet
- Neon™
- PaddlePaddle*
- TensorFlow*
- ONNX
  - PyTorch*
  - Caffe2*
  - CNTK*

Hardware

- Intel® Xeon®
- Intel® Nervana™ NNP
- GPU
- Inference Engine
  - CPU
  - FPGA
  - GPU

CALL TO ACTION

Help nGraph advance AI and DL application development by using and contributing to our performance-optimizing model compiler for multiple compute devices and deep learning frameworks.

- Python and C++ APIs
- Encouraging performance
- For training and inference
- Dynamic and ahead of time compilation

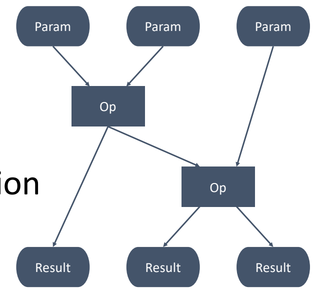
Framework

Bridge

- Backend for frameworks
- Exposes nGraph backends and compilers
- Builds model graph
- Manages tensor allocations
- Invokes computations

nGraph Library core ops

- Graph construction
- Compilation
- Execution
- Graph
  - Describes computation
  - Stateless
  - Strongly-typed



Backend

Transformer

- Optimizes graph
- Plans memory allocation
- Determines tensor layouts
- Generates code
- Transfers data to/from device
- Invokes code