TVM at Arm

TVM Summit – Seattle

Ramana Radhakrishnan
u99127/@ramana-arm

5th December 2019
Agenda

• AI / ML in End Points
• Brief overview of Arm’s ML Platform
• Using TVM in Arm
  • Current Areas of Interest
  • Future Areas of Interest
• Challenges / Observations
What is AI Being Used for in Endpoints?

**Vision**
Images and video
Object detection, face unlock, defocus (bokeh), beautification, scaling, etc.

**Voice**
Recognition and creation
Keyword spotting, speech recognition, natural language processing, speech synthesis, etc.

**Vibration**
Any ‘signal’
Accelerometer, pressure, lidar/radar, speed, shock, vibration, pollution, density, viscosity, etc.

*AI performs well with ‘patterns’ of data*
Diversity of AI Requirements in the Market

**Premium**
- Best user experience and responsiveness
- Highest performance in power-efficient design

**Balanced**
- Superior user experience in mid-range designs
- Balance performance with area and power

**Cost Sensitive**
- Delivering advanced user experiences for the most cost-sensitive designs
- Optimized for performance in the smallest area
Introducing Ethos NPUs for Every Market Segment

Performance-critical AI applications delivering premium experiences

Enabling AI applications in mid-range devices balancing performance with cost and battery life constraints

Supporting AI applications in the most cost-sensitive endpoint devices
Ethos NPU Software Stack
Comprehensive ML Platform Makes Developing AI Easy

- **Ecosystem**
  - AI/ML Applications, Algorithms and Frameworks
    - TensorFlow™
    - TensorFlowLite™
    - PyTorch™
    - ONNX™
    - mxnet™
    - Caffe™
    - Caffe2™
    - Android™ NNAPI

- **Software Products**
  - Software Libraries Optimized for Arm Hardware
    - arm NN
    - arm COMPUTE LIBRARY
    - CMSIS-NN

- **Hardware Products**
  - Arm Hardware IP for AI/ML
    - CPU
      - arm CORTEX
      - arm DynamIQ
      - arm NEOVERSE
    - GPU
      - arm MALI
    - NPU
      - arm ETHOS
    - Partner IP
      - DSPs, FPGAs, Accelerators
Ethos Integration into TVM – Compile Time

- DL Framework Frontend
- NPU Graph Partitioning
- Relay Lowering
- Rest of the TVM stack
- Annotated Output
Current Areas of Work with TVM

Arm CPU and GPU

- Support for Mali Bifrost schedules.
  - Improvements by about 20-70%
  - Interested in Arm CPU architecture support
  - Investigating Arm Compute Library integration

General Areas

- Pre-quantized TensorFlow-Lite
  - Some operator support
- Framework versioning.
- Reviewing various bits of Arm architecture support.
- LLDB Pretty Printers
- Investigating μTVM

Ethos NPU

- Graph Partitioning for NPU
- Integrating support for Ethos-N77, Ethos-N57 and Ethos-N37
Future Areas of Interest

General Framework

Graph partitioning and Relay optimizations.

Improvements to code generation and generic optimizations.

Auto-tuning

Common command line utilities for using TVM.

Improvements to Continuous Integration / Testing

μTVM

Arm Architecture Support

Armv8-A architecture

• Scalable Vector Extensions (SVE/SVE2)
• Matrix Multiplication Support
• BFloat16
• Improved Advanced SIMD Support

GPU Support

• Mali Bifrost
• Mali Valhall

Cortex-M Architecture

• DSP Instructions support.
• Support for Helium / MVE.
Challenges / Opportunities

**Deployment**

- **Getting ready for packaging**
  - conda
  - pypi packaging
  - Integration with native packaging

- **Release process**

- **Continuous Integration**
  - Execution tests and performance monitoring.
  - Managing version updates in frameworks

**Scalability**

- **Developmental Practice**
  - Features vs Bug fixes.
  - Isolation of changes.

- **Developer efficiency**
  - Better explanation with changes
  - Debug helpers.
  - Understanding the test infrastructure.

- **Getting Started**
  - Make it easier!
And finally, we are hiring!
tvm-driver

- tvm-driver –help
  - compile
    - --debug-relay-all
    - --debug-tvm-all
    - --debug-all
    - --print-llvm
    - --print-assembler
  - execute
    - --native
    - --remote
  - auto-tune

Motivation

- Ease of use of TVM stack
- Common way of getting hold of outputs.