tinyML development with TensorFlow Lite for Microcontrollers using CMSIS-NN and Ethos-U55

Fredrik Knutsson, Felix Johnny Thomasmathibaln
June 30, 2020
# AI Virtual Tech Talks Series

<table>
<thead>
<tr>
<th>Date</th>
<th>Title</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>See Arm’s YouTube for recording</td>
<td>Machine learning for embedded systems at the edge</td>
<td>Arm and NXP</td>
</tr>
<tr>
<td>Today</td>
<td>TensorFlow Lite for Microcontrollers using Arm’s CMSIS-NN and Ethos-U55</td>
<td>Arm</td>
</tr>
<tr>
<td>July, 14</td>
<td>Demystify artificial intelligence on Arm MCUs</td>
<td>Cartesiam.ai</td>
</tr>
<tr>
<td>July, 28</td>
<td>Speech recognition on Arm Cortex-M</td>
<td>Fluent.ai</td>
</tr>
<tr>
<td>August, 11</td>
<td>Getting started with Arm Cortex-M software development and Arm Development Studio</td>
<td>Arm</td>
</tr>
<tr>
<td>August, 25</td>
<td>Efficient ML across Arm from Cortex-M to Web Assembly</td>
<td>Edge Impulse</td>
</tr>
</tbody>
</table>

Visit: developer.arm.com/solutions/machine-learning-on-arm/ai-virtual-tech-talks
Today’s speakers

Fredrik Knutsson
ML Software Team Lead

Felix Johnny Thomasmathibalan
ML Engineer
Agenda

• Tensorflow Lite for Microcontrollers (TFLu)
• CMSIS-NN
  • Neural network kernels developed to maximize the performance on Cortex-M CPU
• Ethos-U55
  • A new class of machine learning (ML) processor, called a microNPU, specifically designed to accelerate ML inference in area-constrained embedded and IoT devices.
• Integration: TFLu, Ethos-U55 and CMSIS-NN
  • CMSIS-NN and Ethos-U55 integrated with Tensorflow Lite for microcontrollers
• Demo: CMSIS-NN / TFLu speed-up on Arduino
Tensorflow Lite for Microcontrollers (TFLu)
TensorFlow Lite for Microcontrollers (TFLu)

- Version of TensorFlow Lite designed to execute neural networks on microcontrollers, starting at only a few kB of memory
- Designed to be portable even to 'bare metal' systems
- The core runtime is ~20kB.

Examples/demos
- Micro speech: Detects simple commands such as yes, no and silence.
- Person detection: Detects whether a person is in the room or not.
- Magic wand demo for image recognition etc.

- Generate multiple projects, for example MbedOS and Arduino
- Over 50 operators supported currently. Growing quickly
  - Many integrated operator optimizations
CMSIS-NN
Efficient Neural Network kernels for Arm Cortex-M CPUs via TFLu
CMSIS
Pathway to the Arm ecosystem

- Cortex Microcontroller Software Interface Standard
- Consistent, generic, and standardized software building blocks
- Available for all Cortex-M and Cortex-A5, Cortex-A7 and Cortex-A9 processors
- Open source – public development on GitHub: https://github.com/ARM-software/CMSIS_5
CMSIS-NN
Part of CMSIS that provide optimized ML kernel implementation
8-bit MAC as SIMD operation

Load data -> MAC -> Load data -> MAC -> ..... -> Save data

DSP Extension

- A max capability of 2 MACs/cycle.
- Cortex-M4 processor: 1 MAC/cycle
- Cortex-M7 processor: 2 MAC/cycle (dual issue)

M-profile Vector Extension (Helium tech.)

- Cortex-M55 processor: 8 MAC/cycle
- MAC operands use vector registers (128 bit) and result is stored in a 32 bit GP register.
  \[ y += \sum_{n=1}^{16} (a_n \times b_n), \text{ in two cycles} \]

SIMD – Single Instruction Multiple Data

MAC - Multiply Accumulate

Vector register (128 bit)  
GP register (32 bit)  
\[ a_1 | \ldots | a_{16} \rightarrow y \]  
\[ b_1 | \ldots | b_{16} \]  
\[ \ldots \]  
\[ y \rightarrow \]  
\[ \ldots \]  
\[ \]  
\[ R0 \]  
\[ R1 \]  
\[ R11 \]  
\[ R12 \]
Performance Results - TFLu runtime with CMSIS-NN

On a Cortex-M55 system

- These numbers show current improvements on an FPGA reference system
- Continuously improving performance
Ethos-U55: Accelerating ML Compute further using microNPUs
Ethos-U55: First microNPU for Cortex-M CPUs

• Neural network processor for Cortex-M systems
  • Works alongside Cortex-M55, Cortex-M7, Cortex-M33 and Cortex-M4 processors

• Designed for embedded type systems
  • Fast on-chip SRAM and a slower system flash

• Heavy compute operators for CNN and RNN accelerated in hardware.

• Support for efficient weight compression
  • Compression typically offline
  • Decompression on-the-fly

• Configurations 32, 64, 128 or 256 MAC/cc
  • 8-bit activations use 1 cc per MAC
  • 16-bit activations use 2 cc per MAC
Ethos-U55 Optimized Software Flow

- Train network in TensorFlow
- Quantize it to Int8 TFL flatbuffer file (.tflite file)
- Vela compiler identifies graphs to run on Ethos-U55
  - Optimizes, schedules and allocates these graphs
  - Lossless compression, reducing size of tflite file
- Runtime executable file on device
- Accelerates kernels on Ethos-U55. Driver handles the communication
- The remaining layers are executed on Cortex-M
  - CMSIS-NN optimized kernels if available
  - Fallback on the TFLu reference kernels
Vela Compiler
A Python based optimizer executed on your computer

- Reads a tflite file, writes a modified tflite file
- Generates commands for microNPU
- Optimizes scheduling of subgraphs
- Loss-less compression of weights
- Reduces SRAM and Flash footprint
- Enabling networks previously not feasible in embedded systems!
- Open source
Ethos-U55 Performance Results

Using **256 MACs/Cycle** configuration vs. Cortex-M4 using CMSIS-NN optimizations

- **Wav2letter**: 280x
- **MobileNet V2**: 300x
Ethos-U55 & CMSIS-NN: Integration with Tensorflow Lite for Microcontrollers
Software Stack Integration
Add CMSIS-NN and Ethos-U55 under the same stack

- TFLu is built as a lib, then linked with application
- Optimized kernels enabled by using “TAGS” in the TFLu build system
- Software is open source
  - Vela compiler, Ethos-U55 driver, TFLu and CMSIS-NN
Optimize Where it Matters...
...and always have a fallback path

- Reference kernels always a possibility
- For more horsepower - CMSIS-NN
- For most horsepower - Ethos-U55

<table>
<thead>
<tr>
<th>Kernel</th>
<th>TFLu reference implementation</th>
<th>CMSIS-NN (fast)</th>
<th>NPU (faster)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel 1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Kernel 2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Kernel 3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Kernel 4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Kernel 5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Kernel 6</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kernel 7</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Build TFLu with Ethos-U55 and CMSIS-NN
Access to optimized kernels through TFLu, simple example

• Step 1: Clone TensorFlow repository from GitHub

```bash
git clone https://github.com/tensorflow/tensorflow
```

• Step 2: Compile it using TAGS, in prio order.

```bash
make -f tensorflow/lite/micro/tools/make/Makefile TAGS="ethos-u cmsis-nn" TARGET=<your cortex-m plus ethos-u55 board> person_detection_int8
```
Demo:
Person detection with CMSIS-NN and TFLu
The Hardware
Arduino Nano 33 BLE Sense + Arducam Mini 2MP Plus

- Powered by Arm’s Cortex-M4 CPU
- 1 MB flash. 256kB SRAM. 64MHz.
Step-by-step
Utilize CMSIS-NN in TFLu on an Arduino Nano 33 BLE Sense

- **Step 1 (optional):** Clone TensorFlow repository from GitHub
  
git clone https://github.com/tensorflow/tensorflow

- **Step 2 (optional):** Generate an Arduino project
  
make -f tensorflow/lite/micro/tools/make/Makefile TARGET=arduino TAGS=cmsis-nn generate_arduino_zip

- **Step 3 (optional):** Include the generated project into your Arduino libraries folder
  
unzip tensorflow_lite.zip -d ~/Arduino/libraries/

- **Step 4: Compile and flash demo using the Arduino IDE**
  
  - Check “person detection experimental” example in library “Arduino_TensorFlowLite”. A one button install using Arduino IDE library manager.
Useful links

- TFLu + Ethos-U55 instructions: https://github.com/tensorflow/tensorflow/blob/master/tensorflow/lite/micro/kernels/ethos-u/README.md
- CMSIS GitHub: https://github.com/ARM-software/CMSIS_5
- Person Detection Int8 example: https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/micro/examples/person_detection_experimental
- ML platform Ethos-U landing page: https://review.mlplatform.org/plugins/gitiles/ml/ethos-u/ethos-u/+/refs/heads/master/README.md
Contact us!

- Fredrik Knutsson (freddan80 @ Github)
- Felix Johnny Thomasmathibalan (felix-johnny @ Github) and www.instagram.com/photoquiver/
- Jens Elofsson (jenselofsson @ Github)
- Måns Nilsson (mansnils @ Github)
- Patrik Laurell (patriklaurell @ Github)
- Magnus Midholt (mmidholt @ Github)
Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
to"h
Join our next virtual tech talk: Demystify artificial intelligence on Arm MCUs

Tuesday 14 June

Register here: developer.arm.com/solutions/machine-learning-on-arm/ai-virtual-tech-talks
The Arm trademarks featured in this presentation are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

www.arm.com/company/policies/trademarks
Thank You
Danke
Merci
谢谢
ありがとうございます
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה