Deep Learning on Arm Cortex-M Microcontrollers

Rod Crawford
Director Software Technologies, Arm
What is Machine Learning (ML)?

Additional terms

• **Location**
  - **Cloud** – processing done in data farms
  - **Edge** – processing done in local devices (growing much faster than Cloud ML)

• **Key components of machine learning**
  - **Model** – a mathematical approximation of a collection of input data
  - **Training** – in deep learning, data-sets are used to create a ‘model’
  - **Inference** – in deep learning, a ‘model’ is used to check against new data
Why is ML Moving to the Edge

Bandwidth  Power  Cost  Latency  Reliability  Security
ML Edge Use cases

Increasing power and cost (silicon)

Increasing performance (ops/second)

Arm Cortex-M systems

~1-10mW

 Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)

ML Edge Use cases

Keyword
detection

Pattern
training

Object
detection

Image
enhancement

Autonomous
drive

~100W

Voice & image
recognition

~10W

Increasing power and cost (silicon)

Arm Cortex-M systems

~1-10mW

Keyword detection

~1W

Object detection

~10W

Pattern training

Image enhancement

~100W

Voice & image recognition

Autonomous drive

~1W

~1-10mW

Increasing power and cost (silicon)
Use cases demand more embedded intelligence

Expanding opportunity for the embedded intelligence market

Arm Cortex-M

More data  More processing  More algorithms  More compute

Sensor fusion  Contextual awareness  Voice activation  Biometrics
Developing NN Solution on Cortex-M MCUs

Problem

Trained NN model

Deployable Solution

Optimized code on h/w

Hardware-constrained NN model search

NN model translation

Optimized NN functions: CMSIS-NN
Keyword Spotting

Listen for certain words / phrases

- Voice activation: “Ok Google”, “Hey Siri”, “Alexa”
- Simple commands: “Play music”, “Pause”, “Set Alarm for 10 am”

Feature extraction

FFT-based mel frequency cepstral coefficients (MFCC) or log filter bank energies (LFBE).

Classification

Neural network based – DNN, CNN, RNN (LSTM/GRU) or a mix of them
Arm Cortex-M based MCU Platforms

- **ST Nucleo-F103RB**
  - Cortex-M3, 72 MHz
  - 20KB RAM, 128KB Flash

- **Nordic nRF52-DK**
  - Cortex-M4, 64 MHz
  - 64KB RAM, 512KB Flash

- **ST Nucleo-F103RB**
  - Cortex-M3, 72 MHz
  - 20KB RAM, 128KB Flash

- **NXP LPC11U24**
  - Cortex-M0, 48 MHz
  - 8KB RAM, 32KB Flash

- **ST Nucleo-F411RE**
  - Cortex-M4, 100 MHz
  - 128KB RAM, 512KB Flash

- **ST Nucleo-F746ZG**
  - Cortex-M7, 216 MHz
  - 320KB RAM, 1MB Flash

- **NXP i.MX RT 1050**
  - Cortex-M7, 600 MHz
  - 512KB RAM

More boards at: https://os.mbed.com/platforms/
Keyword Spotting NN Models: Memory vs. Ops

Need **compact models**: that fit within the Cortex-M system memory.

Need models with **less operations**: to achieve real time performance.

Neural network model search parameters
- NN architecture
- Number of input features
- Number of layers (3-layers, 4-layers, etc.)
- Types of layers (conv, ds-conv, fc, pool, etc.)
- Number of features per layer

**NN Models from literature trained on Google speech commands dataset.**


H/W-Constrained NN Model Search

- **DNN**
  - 10K operations
  - 35K parameters
  - 161K operations
  - 13K parameters

- **Basic LSTM**
  - 1000K operations
  - 109K operations
  - 441K operations
  - 108K parameters

- **LSTM**
  - 100M operations
  - 28K parameters
  - 216K parameters

- **CRNN**
  - 10M operations
  - 216K parameters
Summary of Best NN models

<table>
<thead>
<tr>
<th>NN model</th>
<th>S(80KB, 6MOps)</th>
<th>M(200KB, 20MOps)</th>
<th>L(500KB, 80MOps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN</td>
<td>Acc. 84.6%</td>
<td>Mem. 80.0KB</td>
<td>Ops 158.8K</td>
</tr>
<tr>
<td>CNN</td>
<td>91.6%</td>
<td>79.0KB</td>
<td>5.0M</td>
</tr>
<tr>
<td>Basic LSTM</td>
<td>92.0%</td>
<td>63.3KB</td>
<td>5.9M</td>
</tr>
<tr>
<td>LSTM</td>
<td>92.9%</td>
<td>79.5KB</td>
<td>3.9M</td>
</tr>
<tr>
<td>GRU</td>
<td>93.5%</td>
<td>78.8KB</td>
<td>3.8M</td>
</tr>
<tr>
<td>CRNN</td>
<td>94.0%</td>
<td>79.7KB</td>
<td>3.0M</td>
</tr>
<tr>
<td>DS-CNN</td>
<td>94.4%</td>
<td>38.6KB</td>
<td>5.4M</td>
</tr>
</tbody>
</table>

Depthwise Separable CNN (DS-CNN) achieves highest accuracy
Accuracy asymptotically reaches to 95%.


Training scripts and models are available on github:
https://github.com/ARM-software/ML-KWS-for-MCU
Developing NN Solution on Cortex-M MCUs

Problem

Trained NN model

Deployable Solution

Optimized code on h/w

Hardware-constrained NN model search

NN model translation

Optimized NN functions: CMSIS-NN
NN Model Quantization

Quantization type impacts model size and performance.

• Bit-width: 8-bit or 16-bit
• Symmetric around zero or not
• Quantization range as \([-2^n, 2^n]\) a.k.a. fixed-point quantization.

Steps in model quantization

• Run quantization sweeps to identify optimal quantization strategy.
• Quantize weights: does not need a dataset
• Quantize activations: run the model with dataset to extract ranges.

Neural networks can tolerate quantization.

Minimal loss in accuracy (~0.1%).
May increase accuracy in some cases, because of regularization (or reduce over-fitting).

<table>
<thead>
<tr>
<th>NN model</th>
<th>32-bit floating point model accuracy</th>
<th>8-bit quantized model accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
<td>Val.</td>
</tr>
<tr>
<td>DNN</td>
<td>97.77%</td>
<td>88.04%</td>
</tr>
<tr>
<td>Basic LSTM</td>
<td>98.38%</td>
<td>92.69%</td>
</tr>
<tr>
<td>GRU</td>
<td>99.23%</td>
<td>93.92%</td>
</tr>
<tr>
<td>CRNN</td>
<td>98.34%</td>
<td>93.99%</td>
</tr>
</tbody>
</table>
Model Deployment on Cortex-M MCUs

Running ML framework on Cortex-M systems is impractical.

Need to run bare-metal code to efficiently use the limited resources.

Arm NN translates trained model to the code that runs on Cortex-M cores using CMSIS-NN functions.

**CMSIS-NN**: optimized low-level NN functions for Cortex-M CPUs.

CMSIS-NN APIs may also be directly used in the application code.
Developing NN Solution on Cortex-M MCUs

Problem

Trained NN model

Deployable Solution

Optimized code on h/w

Hardware-constrained NN model search

NN model translation

Optimized NN functions: CMSIS-NN
Cortex Microcontroller Software Interface Standard (CMSIS)

CMSIS: vendor-independent hardware abstraction layer for Cortex-M processor cores. Enable consistent device support, reduce learning curve and time-to-market.

Open-source: https://github.com/ARM-software/CMSIS_5/
CMSIS-NN: collection of optimized neural network functions for Cortex-M cores

Key considerations

- Improve performance using SIMD instructions.
- Minimize memory footprint.
CMSIS-NN – Efficient NN kernels for Cortex-M CPUs

Convolution
- Boost compute density with GEMM based implementation
- Reduce data movement overhead with depth-first data layout
- Interleave data movement and compute to minimize memory footprint

Pooling
- Improve performance by splitting pooling into x-y directions
- Improve memory access and footprint with in-situ updates

Activation
- ReLU: Improve parallelism by branch-free implementation
- Sigmoid/Tanh: fast table-lookup instead of exponent computation

*Baseline uses CMSIS 1D Conv and Caffe-like Pooling/ReLU
CNN on Cortex-M7

- CNN with 8-bit weights and 8-bit activations
- Total memory footprint: 87 kB weights + 40 kB activations + 10 kB buffers (I/O etc.)
- Total operations: 24.7 MOps, run time: 99.1ms
- Example code available in CMSIS-NN github.

OpenMV platform with a Cortex-M7

Video: https://www.youtube.com/watch?v=PdWj_fY9Og
Demo

https://youtu.be/PdWi_fvY9Og?t=1m
Summary

• ML is moving to the IoT Edge and Cortex-M plays a key role in enabling embedded intelligence.

• Case study: keyword spotting on Cortex-M CPU
  • Hardware-constrained neural network architecture exploration
  • Deployment flow of model on Cortex-M CPUs
  • CMSIS-NN: optimized neural network functions
Thank You!
Danke!
Merci!
谢谢!
ありがとうございます!
Gracias!
Kiitos!
감사합니다
धन्यवाद
Project Trillium: Arm ML and OD Processors

Ground-up design for high performance and efficiency

Massive uplift from CPUs, GPUs, DSPs and accelerators

Enabled by open-source software

First-generation ML processor targets Mobile market
Flexible, Scalable ML Solutions

- Only Arm can enable ML everywhere
- 90% of the AI-enabled units shipped today are based on Arm

(source: IDC WW Embedded and Intelligent Systems Forecast, 2017-2022 and Arm forecast)
Project Trillium: Arm's ML Computing Platform

## Ecosystem

### AI/ML Applications, Algorithms and Frameworks

- TensorFlow
- Caffe
- Caffe2
- mxnet

### Android NNAPI

## Software Products

### Software Libraries Optimized for Arm Hardware

- Arm NN
- CMSIS-NN
- Compute Library
- Object Detection Libraries

## Hardware Products

### Arm Hardware IP for AI/ML

- **CPU**
  - Arm Cortex-A
  - Arm NEON
  - Arm Cortex-M
  - Arm Mali
  - Arm DynamIQ
- **GPU**
  - Arm Mali
- **ML and OD Processors**
  - Machine Learning
  - Object Detection (OD)
- **Partner IP**
  - DSPs, FPGAs, Accelerators
Resources

CMSIS-NN paper: https://arxiv.org/abs/1801.06601


CMSIS-NN Github link: https://github.com/ARM-software/CMSIS_5/

KWS (Keyword Spotting) paper: https://arxiv.org/abs/1711.07128

KWS blog: https://community.arm.com/processors/b/blog/posts/high-accuracy-keyword-spotting-on-cortex-m-processors

KWS Github link: https://github.com/ARM-software/ML-KWS-for-MCU/


ArmNN SDK blog: https://community.arm.com/tools/b/blog/posts/arm-nn-sdk
Thank You!
Danke!
Merci!
谢谢!
ありがとうございます!
Gracias!
Kiitos!
감사합니다
धन्यवाद
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