



arm

CMSIS-NN

Felix Thomasmathibalan  
Staff Engineer, Machine Learning Group

# Why Target Arm Cortex-M Processors?



Enable Machine Learning on more than **52 billion\*** shipped units



Availability of exciting new models with **lower memory footprint** and **MACs**



Accelerate deployment of Machine Learning on edge devices



## MobileNet V2

- 3.38 MB parameters
- ~ 307 million MACs



## Person Detect(TFLu)

- 250 kBytes
- ~ 7 million MACs

# Open Source CMSIS-NN Library

Cortex Microcontroller Software Interface Standard – Neural Networks



Optimized software library for key machine learning operators.

Empower and enable Cortex-M processors for tinyML applications

Permissive Apache 2.0 license

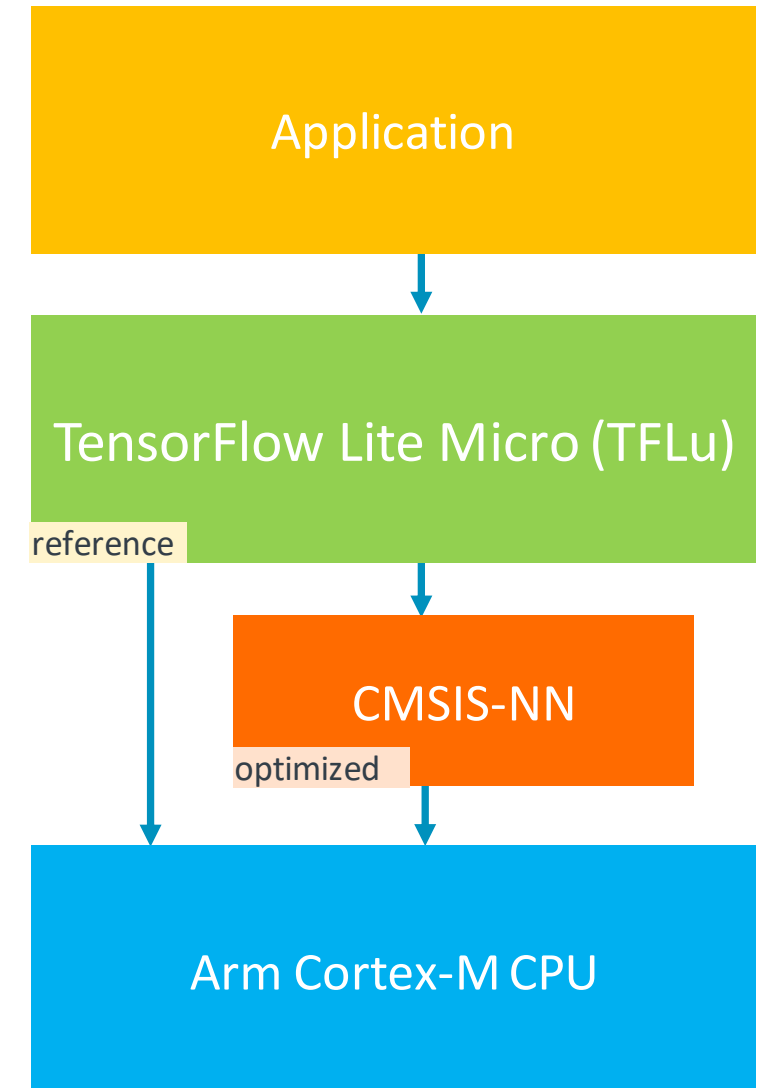
Supports open source framework, TensorFlow Lite for Microcontrollers.

# CMSIS-NN & TensorFlow Lite for Microcontrollers

Access through TensorFlow Lite Micro

Support for int8 quantization specification

Fallback on reference kernels when optimization is not available



# Why Optimize for tinyML ?

## Power constraints

- Extend battery life of edge devices by reducing awake time.



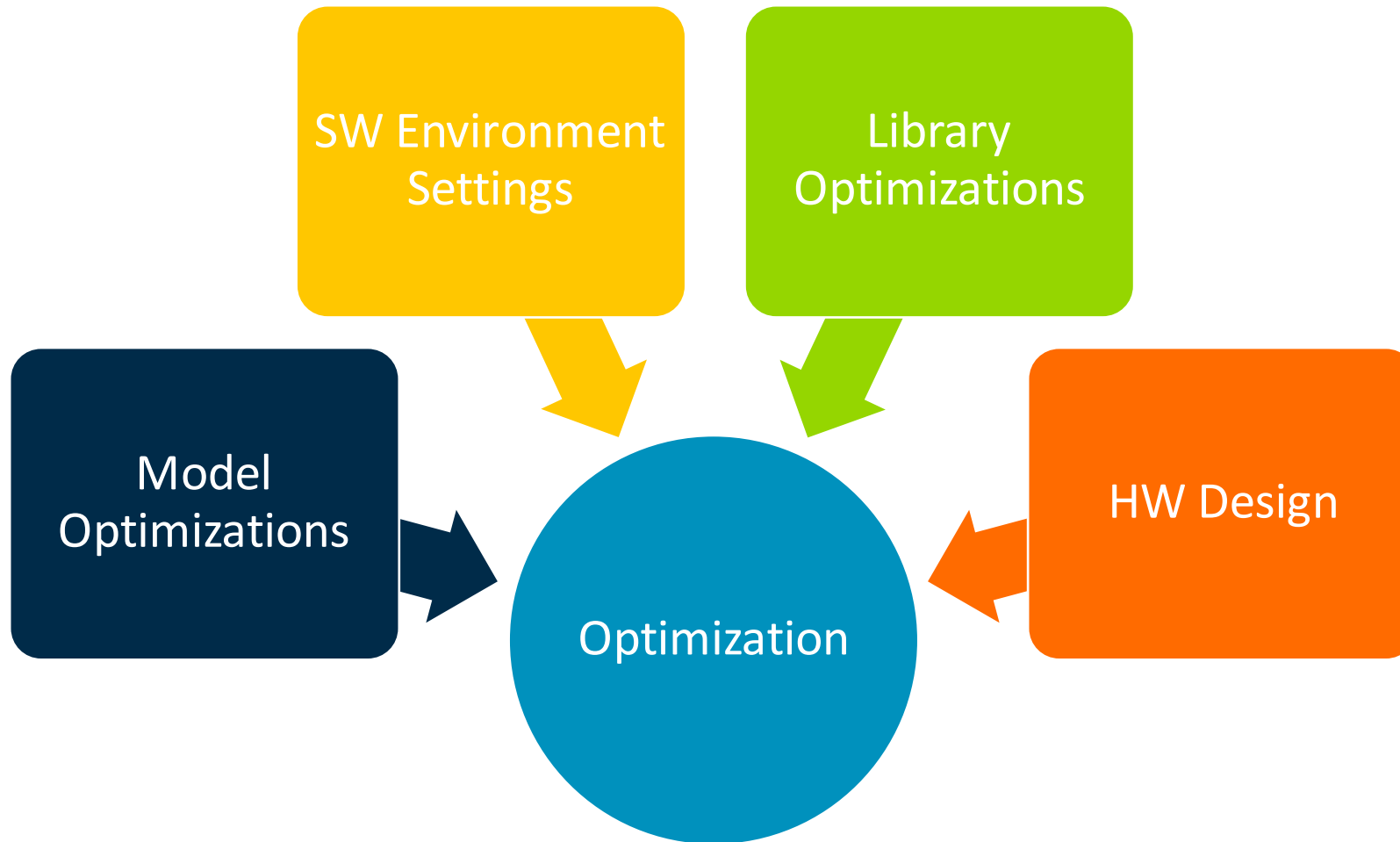
## Cycle constraints

- Enable more complex models to be deployed within the inference budget.
- Meet real time latency constraints.

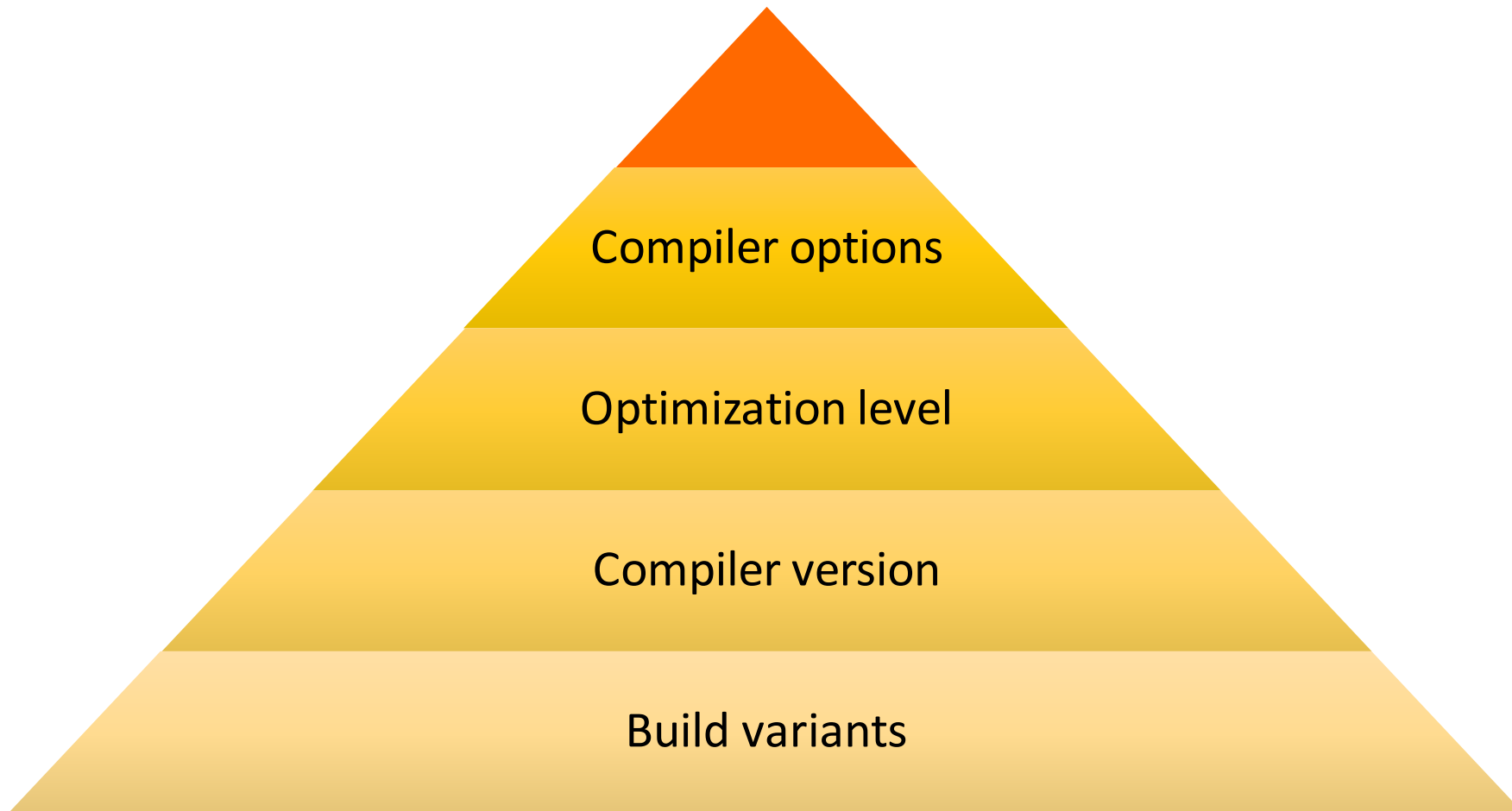


# It Takes a Village to Raise a Child

Everyone has a role to play in optimization, some more important than the other



# Software Environment Foundation



Collect reliable data  
to act upon



# CMSIS-NN Library Optimizations

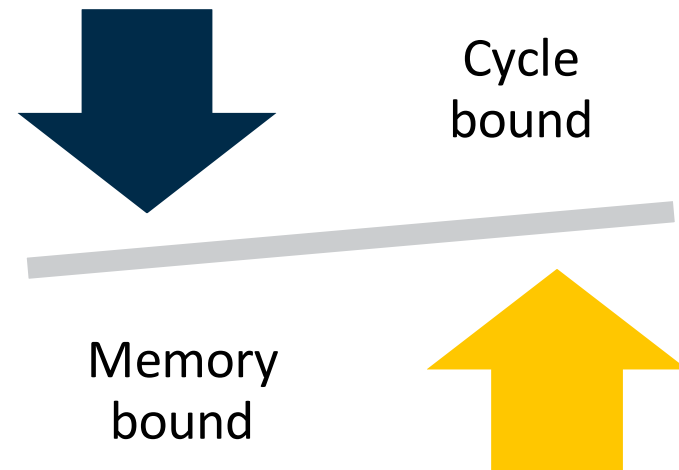
Addresses cycle and memory bound aspects at its core

Cycle bound problem

- Ensure that an algorithm utilizes the processor's capability.

Memory bound problem

- Re-use data for efficient memory access.



# Optimization Categories

The **S**ingle **I**nstruction **M**ultiple **D**ata aspect

## Non-SIMD capable processors



Re-use data to improve memory access efficiency.



Reduce complexity of core processing loops.



E.g., Arm Cortex-M0, Cortex-M3

## SIMD capable processors



Techniques from non-SIMD optimizations.



Utilize processor's capability to process multiple data in a single instruction.

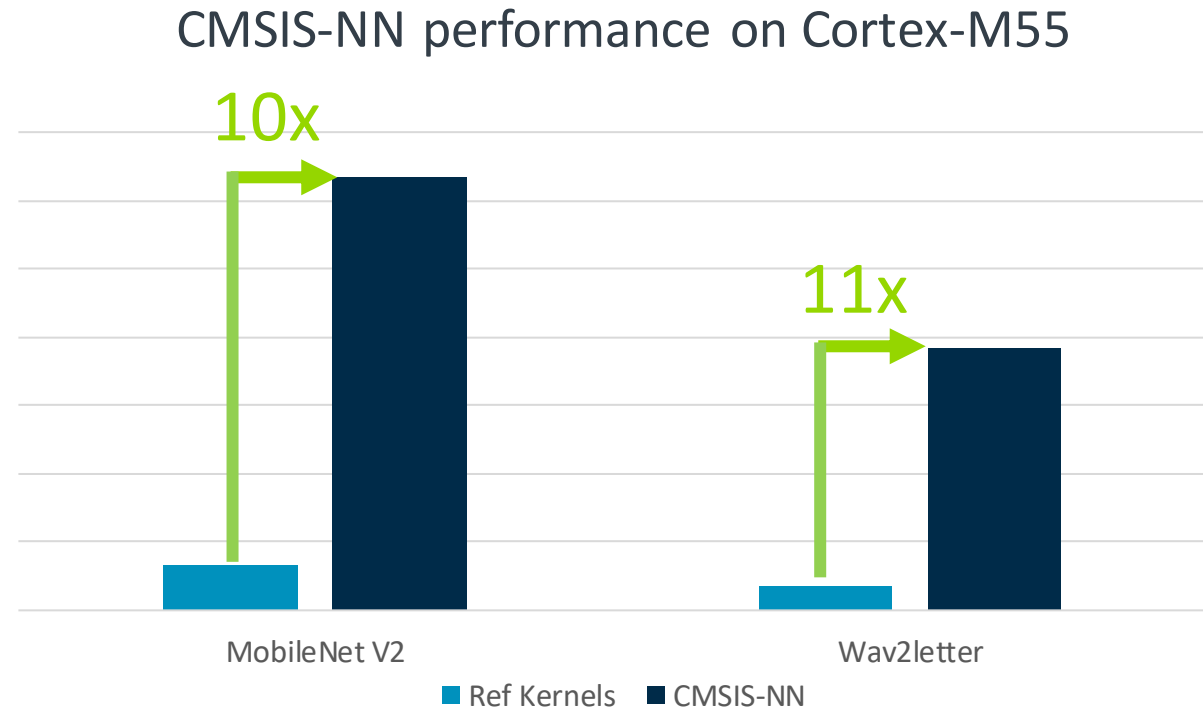


E.g., Cortex-M4, Cortex-M55 with MVE extension.

# Performance Results - TFLu runtime with CMSIS-NN

Cortex-M55 system

- These numbers show current improvements on a FPGA reference system
- Continuously improving performance



# Model Optimizations

Shapes of tensors affect the efficacy of optimized library.

To reduce memory footprint, optimizations for all shapes is not done.

Aligned memory access is recommended for CMSIS-NN and input and output channels that are a multiple of 4 ensures that.

1x1/1xN Convolution and Fully Connected operators can act on input tensors without the need of data rearrangement resulting in better processor utilization.



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](http://www.arm.com/company/policies/trademarks)