

tinyML[®] EMEA

Enabling Ultra-low Power Machine Learning at the Edge

June 26 - 28, 2023



www.tinyML.org



arm



Creating end-to-end tinyML applications for the Arm Ethos-U in the cloud

George Gekov
27/06/2023

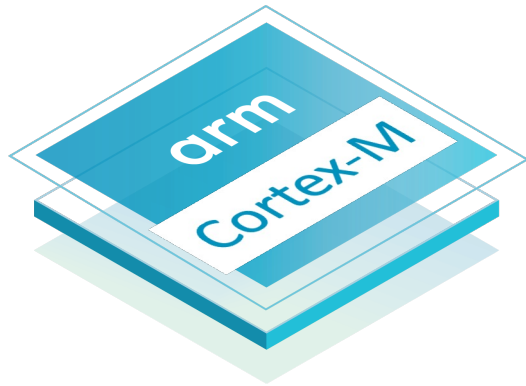
Creating TinyML applications is difficult

1. Explain the key principles in firmware development for TinyML applications
2. Common pitfalls to avoid in the NN design phase

Arm Ethos-U microNPUs for Endpoint & Embedded Solutions

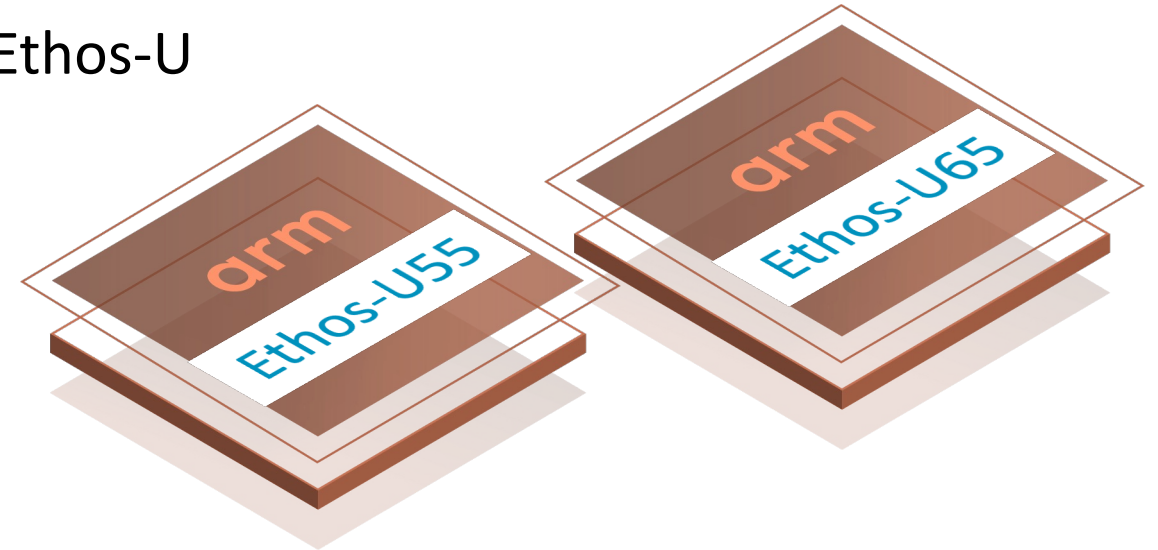
Providing NN acceleration in highly constrained environments

Traditional Cortex-M



- + int8 quantisation
- + Capable to run ML workloads

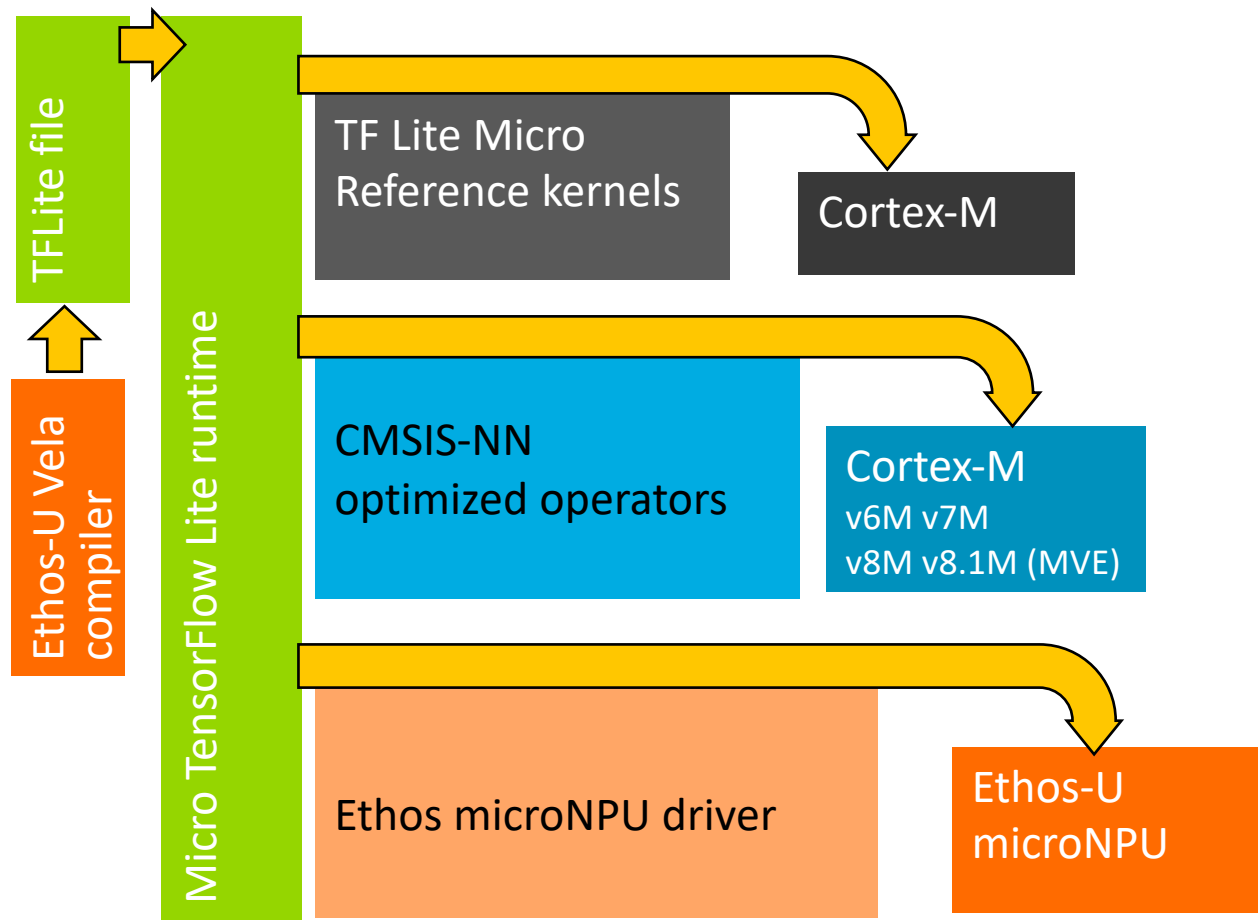
Ethos-U



- + Hardware acceleration for NN
- + **800x improvement in performance**

Main software stack to run ML on Cortex-M today

Cortex-M is robust and flexible, Ethos-U is dedicated ML accelerator



Hardware supported operators

Abs, Add, Average_Pool_2D, Concatenation, Conv_2D, Depthwise_Conv_2D, Fully_Connected, Leaky_ReLu, Logistic, Maximum, Max_Pool_2D, Minimum, Mul, Pack, Quantize, ReLu, ReLu6, ReLu_N1_to_1, Reshape, Resize_Bilinear, Slice, SoftMax, Split, Split_V, Squeeze, Strided_Slice, Sub, TanH, Transpose_Conv, Unpack

Key steps to run an inference on Cortex-M

Pre-processing and post-processing is specific to a model

+ Map the model C byte array

```
model = tflite::GetModel(model_C_array);
```

+ Pull in the TF Lite Micro kernels required for your model

```
static tflite::MicroMutableOpResolver<1> micro_op_resolver(error_reporter);  
if (micro_op_resolver.AddConv2D() != kTfLiteOk) {  
    return;  
}
```

+ Build an interpreter

```
static tflite::MicroInterpreter static_interpreter(  
    model, micro_op_resolver, tensor_arena, kTensorArenaSize, error_reporter);
```

+ Allocate memory

```
TfLiteStatus allocate_status = interpreter->AllocateTensors();
```

+ Run an inference

- TfLiteStatus invoke_status = interpreter->Invoke();

What changes if you are to use an Ethos-U accelerator?

Minimal change in the embedded code

+ Pull in the TF Lite Micro Ethos-U kernel

```
static tflite::MicroMutableOpResolver<1> micro_op_resolver(error_reporter);  
if (micro_op_resolver.AddEthosU() != kTfLiteOk) {  
    return;  
}
```

+ All other steps are unchanged

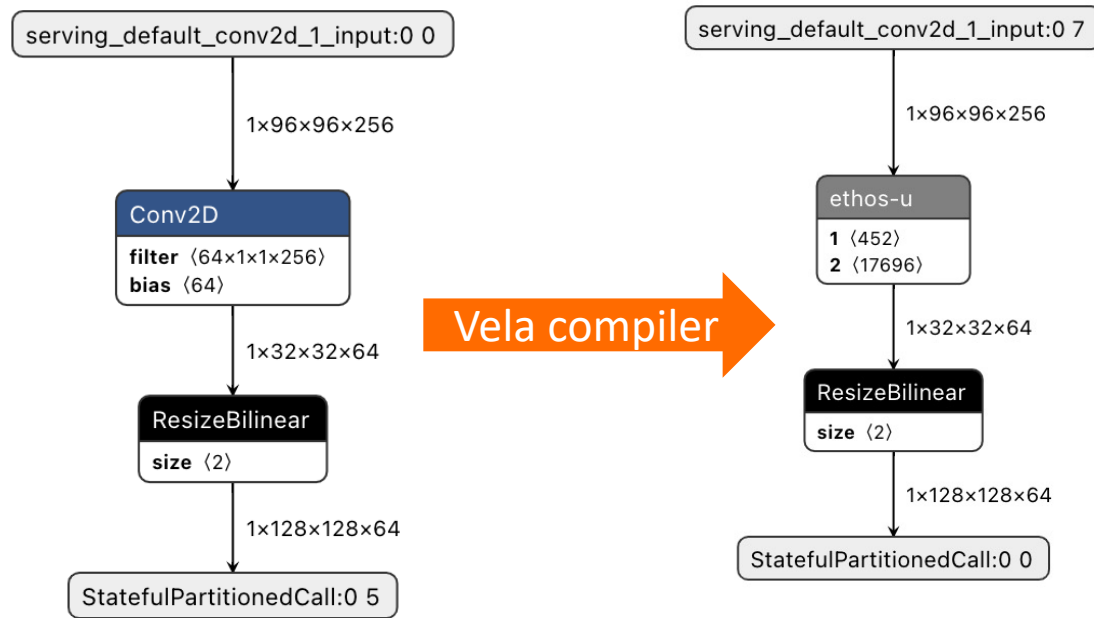
arm

Pitfalls to avoid in the NN
design phase if you want
to benefit fully from
silicon acceleration on
Arm Ethos-U

Hardware supported vs non-supported operator in the NN

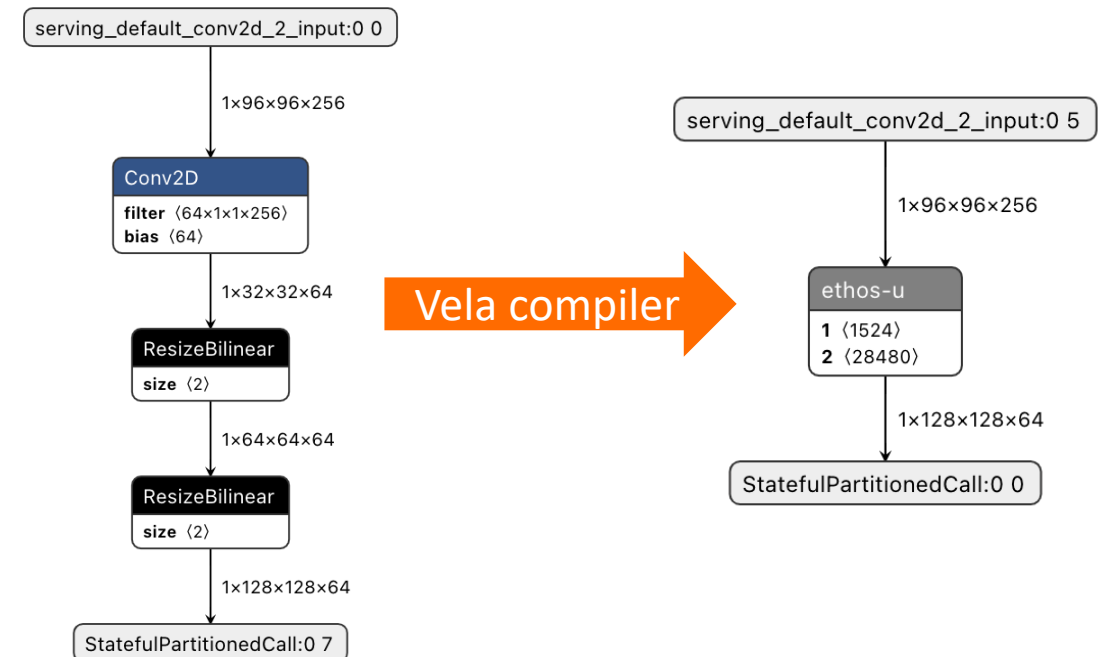
Example of the benefit of using hardware supported operators on Ethos-U

NN with a fallback to Cortex-M55



+ 74M cycles on Ethos-U55 & Cortex-M55

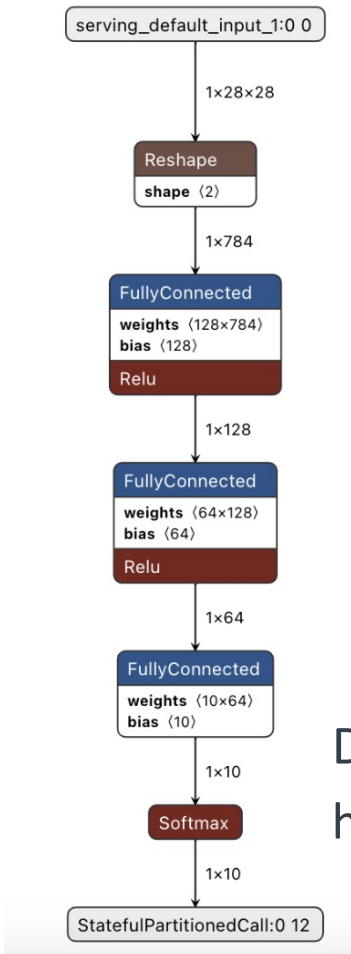
NN fully mapping to the Ethos-U



+ 1.2M cycles on Ethos-U55

Leverage the Weight Compression of the Arm Ethos-U NPU

Pruning & clustering improves performance on memory-bound models



Variant of the model	Ethos-U Active cycles	Accuracy
Baseline	91k	97%
Pruning 80% of weights set to 0	34k	97%
Pruning 80% of weights set to 0 and 32 clusters	26k	97%

Detailed blog:

<https://github.com/ARM-software/ML-examples/tree/main/pruning-clustering-ethos-u>

arm

How do you run end-to-end
tinyML application on
Cortex-M and/or Ethos-U?

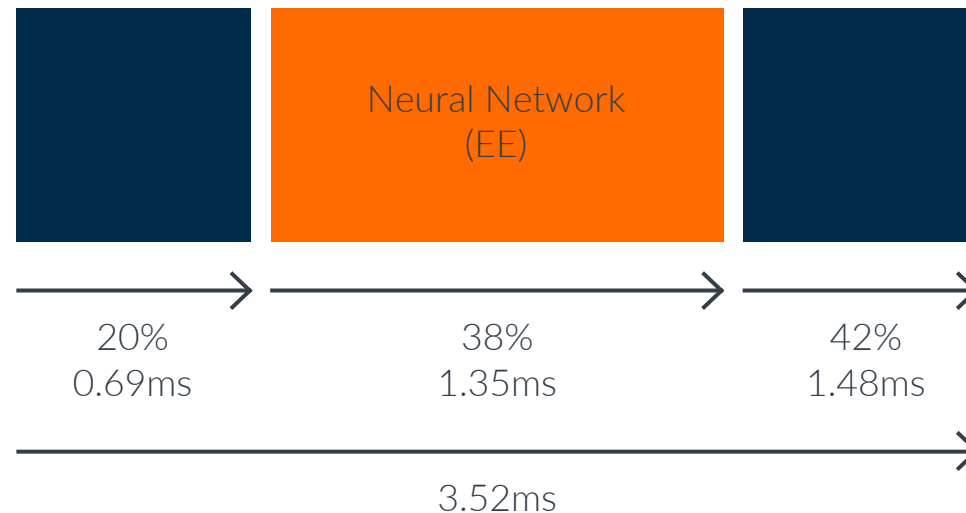
We provide a number of example applications!

+ <https://review.mlplatform.org/plugins/gitiles/ml/ethos-u/ml-embedded-evaluation-kit>

- Open source, Apache 2.0

+ Ready to use end-to-end applications for Arm Ethos-U55/Ethos-U65

- Keyword spotting, speech recognition, noise suppression
- Image classification, object detection, person detection
- Anomaly detection



What can you do with these example applications?

- + Run them on Cortex-M & Ethos-U, in the cloud or locally
- + Evaluate performance of a custom NN on Cortex-M55/85 and/or Ethos-U55/Ethos-U65
- + Adapt them for SoCs with Cortex-M and Ethos-U
 - Need to factor in the specificities of your development board

To wrap up

1. TinyML doesn't have to be difficult – a lot of example applications are already available
2. Ensure the operators in the NN design phase can be accelerated if you aim for optimal performance

arm

Thank You

Danke

Gracias

Grazie

谢谢

ありがとう

Asante

Merci

감사합니다

धन्यवाद

Kiitos

شكرًا

ধন্যবাদ

תודה



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

Copyright Notice



This presentation in this publication was presented as a tinyML[®] EMEA Innovation Forum. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org