Technology briefing:
The future of deep learning is 8-bit

Author: Paul Foster, 5th July, 2019

Summary:

Over the last two years research has proven that the transformation of ML models to fixed-point integer arithmetic can have little significant impact on the accuracy of a model [2][4][12]. This has inspired a quantization process of ML models to 8-bit integers which can reduce model sizes by 75%. The fixed-point integer arithmetic used in the inference engine provides significant performance improvement. These two techniques have allowed ML models to be successfully run on constrained microcontrollers, modern implementations of which are low cost and have low power consumption. We are observing a significant development toward achieving ambient intelligence in small devices.

The technology is maturing quickly, with several approaches being readily available today, to the point that hobbyists can now build key word spotting audio ML solutions for Arduino devices [10].

It is clear that the future of deep learning is 8-bit, with a robust capability to move ML models to very small edge devices for entirely local data analysis and minimal dependency on cloud connectivity.

Azure Sphere provides a uniquely secure, multi-core system-on-a-chip capable of delivering a leading ML multi-model fusion evaluation application architecture across dedicated cores.

Industry review:

ARM:

ARM have built an optimised application library for their Cortex-M microcontroller range – CMSIS. The CMSIS-NN [6] functions provide optimised arithmetic functions for neural network implementation on constrained MCU. Provided Python tooling automatically generates code to implement the NN and to quantize the model data. This functionality forms the basis for many NN implementations on MCU. ARM have recently announced a collaboration joining ARM microTensor with TensorFlowLite [8].

Google:

March 2019, Pete Warden demonstrated TensorFlowLite for Microcontrollers – an inference only, interpreter that runs on Cortex-M3/M4 and larger microcontrollers [1]. It was demonstrated running a key word spotting model on an Ambiq Micro Apollo 3 MCU Cortex-M4 with dedicated BLE core which has <6uA/MHz power consumption allowing it to run on a CR2032 coin cell battery for days. The demonstration board, from Sparkfun – the Sparkfun Edge – is publicly available for $15 [9].

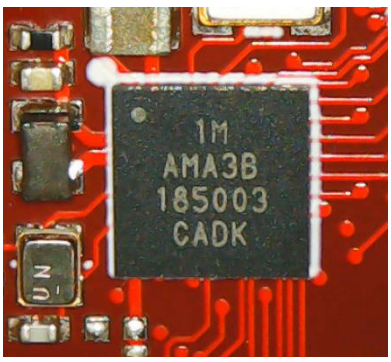

Figure 1: Ambiq Apollo 3 MCU with integrated BLE is 4mm² and runs a keyword spotting ML model

TensorFlowLite for MCU provides a hardware optimised code assembler/generator where handwritten optimisations for hardware platforms are automatically pulled together to compile an application specific TensorFlow interpreter deployed in the solution's application code. Models are then provided as data sets which simplifies their updating over time.

Microsoft Research:

Microsoft Research has delivered several new algorithms and tooling to enable ML on constrained devices [5], which can consume Azure ML produced models. Current work includes the open source Embedded Learning Library (ELL) [3] which uses LLVM to abstract model implementations from target hardware. As LLVM expands, new platforms instantly become available. ELL is described as an ML compiler – compared to the TensorFlowLite interpreter. ELL produces a compiled code implementation of the model to link into a project. TensorFlowLite optimises the functions available to the interpreter to make its size and function specific to the task, the models are then stored as data files. In theory a new project for TensorFlowLite could just be a data update, whereas ELL would be a new

firmware. Given the security requirements of Azure Sphere, as an example of a modern secure MCU, new firmware is the only mechanism to update persistent storage. Other mechanisms would involve data being held in volatile RAM downloaded from the cloud as required or less secure external flash storage.

New algorithms have been published (ProtoNN, Bonsai (not Bons.ai) and FastGRNN) [5] which significantly improve solution implementations on constrained devices. Together with domain specific language tooling to perform the quantization process and automatically generate model implementation code for MCU and for low cost FPGA [4].

Microsoft:

The focus of Microsoft ML tooling is currently in the training of models, with model execution in the cloud or in containerised solutions like IoT Edge. IoT Edge depends on hardware at least two factors greater than the current industry ML on MCU trend. The Azure ML pipeline capably produces models that can be optimised to run on MCU [13].

The Azure Sphere chip, with potentially multiple processing cores, provides an industry leading secure solution to the execution of ML on MCU solutions. The current MT3620AN provides three cores capable of running customer code. Two of these cores are modern Cortex-M4F cores and directly align to supporting the ML on MCU trend.

Microsoft Commercial Software Engineering (CSE):

CSE has been engaged in Azure Sphere development since the start of FY19. Work underway in close collaboration with the Azure Sphere and C++ product groups, is demonstrating the capabilities for ML on MCU devops and multiple module execution on multi-core SoCs [11].

Using the current MT3620 commercial modules, a test harness PCB for automated AzDO CI/CD build pipeline has been designed and implemented with an AzDO self-hosted build agent to enable automated testing on native Azure Sphere hardware.

Industry examples of ML on MCU have been ported to the MT3620 Cortex-M4 cores. Microsoft Research algorithms and tooling is being ported to provide demonstrations on the Azure Sphere.

Using the three customer programmable Azure Sphere cores [of the currently MT3620 Azure Sphere chip], two different ML models can be executed, each on a dedicated Cortex-M4 core, against the same (or different) data to predict an event which the supervising third Cortex-A core code can evaluate to increase insight accuracy through the fusion of the models results.

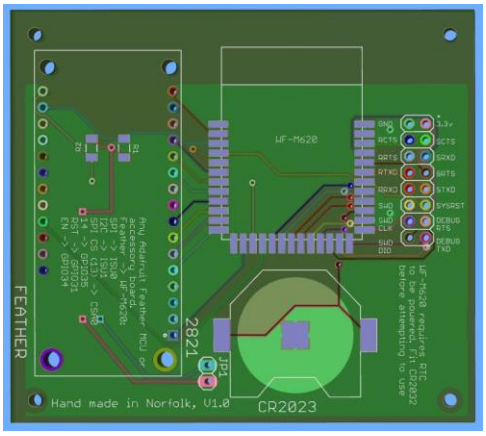

Figure 2: Custom designed CI/CD Azure Sphere testing board

Conclusion:

Size of model is equally important as arithmetic performance for ML on MCU. Research will continue to explore optimal Real number representations for deep learning models [12], with smaller than 8-bit solutions successful for some models. While different Real number representations can improve inference performance, currently only quantization of models achieves both performance improvement and the reduction in model size required by MCU.

The unique high security Azure Sphere solution provides a capable platform for customers wishing to explore multi-model MCU solutions, with centralised, secure application deployment, and multiple MCU cores individually targetable for model and supervisor implementations.

CSE can enable Microsoft's customers:

1. By building the expertise to prove and demonstrate Azure Sphere as the secure MCU for ML
    a. Implement common intelligent edge models identified by CSE ML experts
    b. Build a reference implementation for multi-model fusion evaluation utilising dedicated cores for each model and supervisor application. MT3620 = 2 x ML Model, 1 x Supervisor cores.
    c. Design and build demonstrable PoC sensors for key customer and model scenarios
2. By defining and building the end to end dev ops project pipeline from ML training to MCU testing
    a. Complete AzDO CI/CD pipeline implementation and testing to publish as documented best practice.

ML on secure MCU is a new industry market space for Microsoft to address, which needs the Azure Sphere solution. A new product, in a new space, backed by Azure ML and Azure DevOps and Visual Studio is compelling.

Further reading:

[1] Pete Warden's Why are Eight Bits Enough for Deep Neural Networks? https://petewarden.com/2015/05/23/why-are-eight-bits-enough-for-deep-neural-networks/

[2] Pete Warden's Scaling machine learning models to embedded devices. https://petewarden.com/2019/03/27/scaling-machine-learning-models-to-embedded-devices/

[3] Microsoft Research Embedded Learning Library: https://microsoft.github.io/ELL/

[4] Microsoft Research EdgeML SeeDot quantization tool: https://github.com/microsoft/EdgeML/tree/master/Tools/SeeDot

[5] Microsoft Research resource-efficient ML for Edge and Endpoint IoT devices publications: https://www.microsoft.com/en-us/research/project/resource-efficient-ml-for-the-edge-and-endpoint-iot-devices/#!publications

[6] ARM Helium technology: https://www.arm.com/why-arm/technologies/helium

[7] ARM Image recognition on ARM Cortex-M with CMSIS-NN: https://developer.arm.com/solutions/machine-learning-on-arm/developer-material/how-to-guides/image-recognition-on-arm-cortex-m-with-cmsis-nn/single-page

[8] ARM uTensor and Tensor Flow Announcement: https://os.mbed.com/blog/entry/uTensor-and-Tensor-Flow-Announcement/

[9] Sparkfun Edge: https://www.sparkfun.com/products/15170

[10] Adafruit TFLite Micro Speech (Arduino compatible port of Pete Warden demo): https://github.com/adafruit/Adafruit_TFLite_Micro_Speech

[11] CSE public Azure Sphere Cookbook : https://github.com/PaulDFoster/AzureSphereCookBook

[12] POSIT number system vs Fixed point : https://arxiv.org/pdf/1805.08624.pdf

[13] Training Audio Models using Azure ML, Chris Lovett: https://github.com/microsoft/ELL/wiki/Training-Audio-Models-using-Azure-ML