Sept, 24 2019

Linaro Connect, San Diego

Qualcomm

IoT Benchmarks

Mark Charlebois

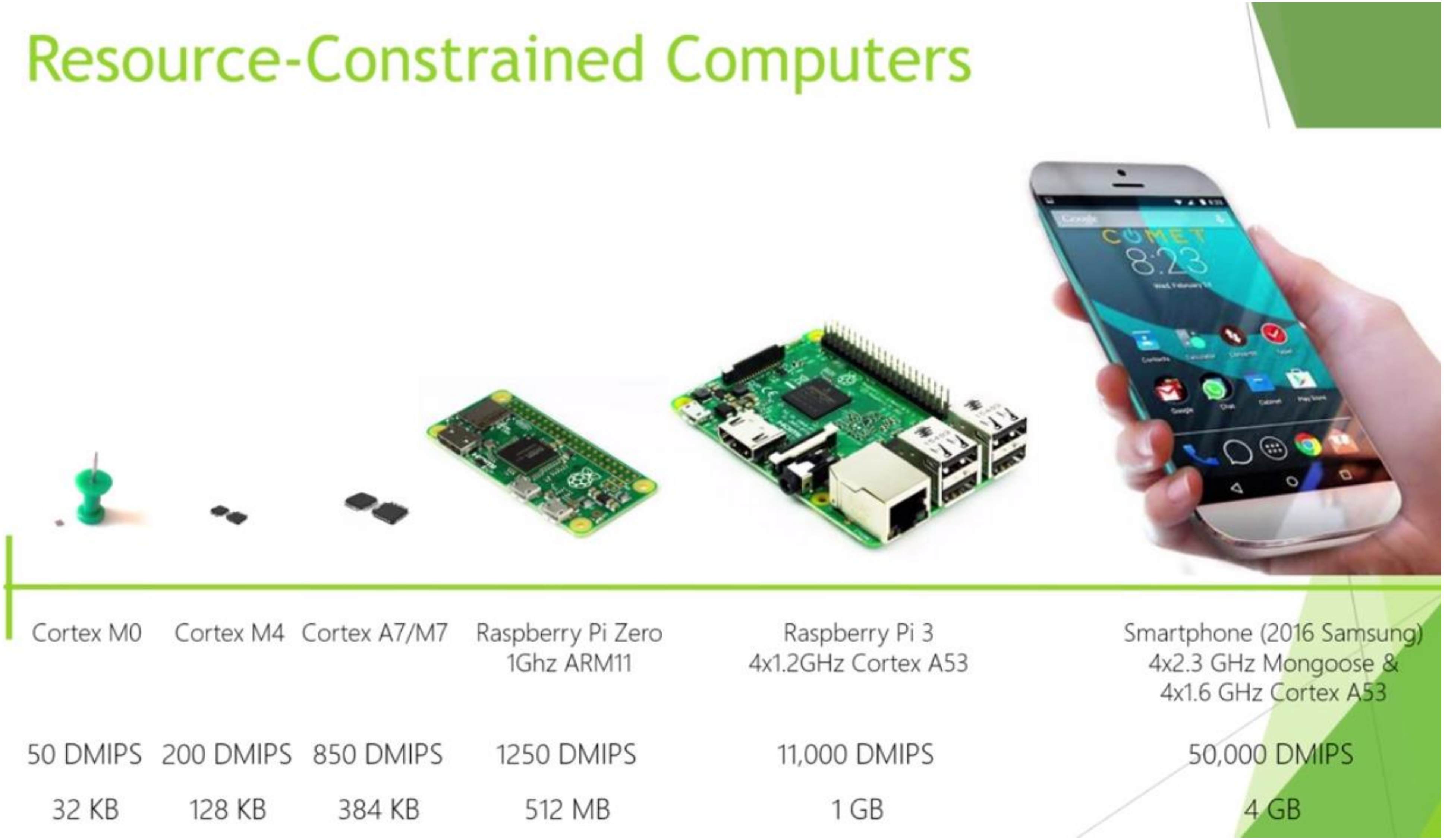
Director Engineering
Qualcomm Technologies Inc

Overview

- IoT Devices
- AI for IoT
- AI On Microcontrollers
- Benchmarks
- Open Issues
- Summary

IoT Devices

https://channel9.msdn.com/Shows/Internet-of-Things-Show/Running-AI-on-IoT-microcontroller-devices-with-ELL



Device Categories

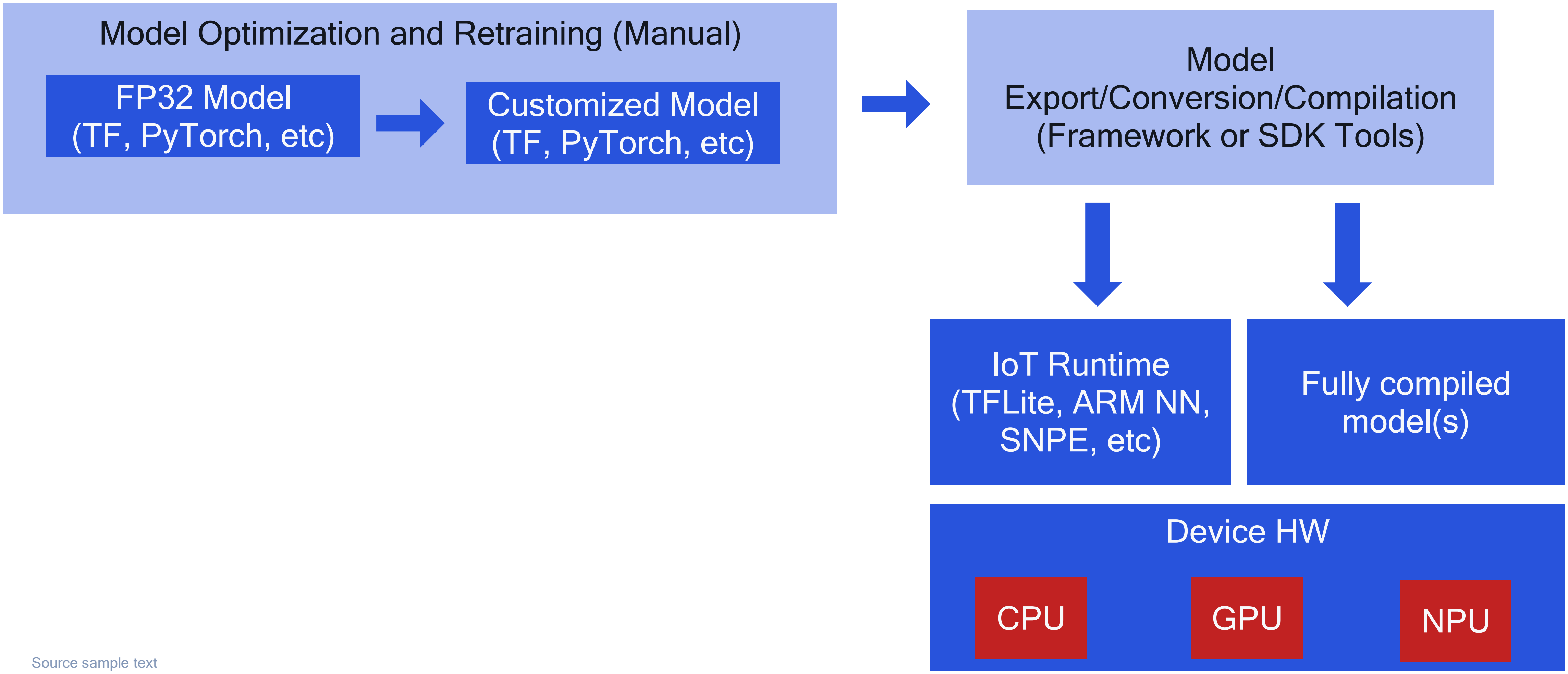
EEMBC Device Categories

- Ultra-Low Power and Internet of Things
- Heterogeneous Compute
- Single-core Processor Performance
- Multi-core Processor Performance
- Phone and Tablet

https://www.eembc.org/products

AI for IoT

Running a Model on IoT Platform



Source sample text

AI on Microcontrollers

TFLite on Microcontrollers

- C++ API, with runtime that fits in 16KB on a Cortex M3
- Uses standard TensorFlow Lite FlatBuffer schema
- Pre-generated project files for popular embedded development platforms, such as Arduino, Keil, and Mbed
- Optimizations for several embedded platforms
- Sample code demonstrating spoken hotword detection
- https://www.tensorflow.org/lite/microcontrollers/overview

ARM NN / CMSIS NN for ARM Microcontrollers

https://github.com/ARM-software/ML-examples

- ARM NN
  - To support the Machine Intelligence Initiative by Linaro, Arm has donated Arm NN, our open-source network machine learning (ML) software.
  - https://www.arm.com/products/silicon-ip-cpu/machine-learning/arm-nn

- CMSIS NN
  - A library of kernels optimized for running neural networks on Cortex-M (and Cortex-A) processor cores.
  - Provides: Convolution, Activation, Fully-connected, Pooling, Softmax, and support Functions
  - https://github.com/ARM-software/CMSIS_5

Benchmarks

Mobile Benchmarks

- Android Benchmarks (Play Store)
  - AIMark (Ludashi)
  - AIBenchmark (Uses NNAPI)
  - AITuTu
  - Neuralscope
  - MLBench
- Consortiums
  - MLPerf
  - AIIA
- Mobile benchmarks use NNAPI, TFLite, and/or Vendor SDK
  - e,g. https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk

Models Used in Mobile Benchmarks

https://neuralscope.org/mobile/index.php?route=product/benchmark

- Object Classification
  - MobileNet-V1 (16.9 MB / 4.3 MB Quantized)
  - MobileNet-V2 (14.0 MB / 3.6 MB Quantized)
  - Inception-V3 (95.3 MB / 24.1 MB Quantized)
  - ResNet-50 (102.2 MB / 25.7 MB Quantized)
  - Resnet34 (AIMark - ~83 MB)
- Object Segmentation
  - DeepLab-V3 (8.5 MB)
- Object Detection
  - MobileNet-SSD (27.3 MB / 6.9 MB Quantized)

MLMark from EMBC

https://github.com/eembc/mlmark

- The targets provided are:
  - Tensorflow (tensorflow) - Only fp32 and concurrency of 1, can use GPU
  - Intel OpenVINO (openvino_ubuntu) - Intel CPUs, GPUs, Movidius Neural Compute Sticks, HDDLr and FPGA.
  - TensorRT Nano (tensorrt) - TensorRT, cuDNN and Cuda for the Jetson Nano platform.
  - ArmNN (armnn_ubuntu) - ArmNN + ACL on CPU (A5 and A7), Mali GPUs. (SSDMobileNet is not supported.)

- The TensorFlow based workloads selected are:
  - ResNet-50 v1.0 (resnet50)
  - MobileNet v1.0 224x224 (mobilenet)
  - SSDMobileNet v1.0 300x300 (ssdmobilenet)

- "Commercial MLMARK License" from EEMBC is required for Licensee to disclose, reference, or publish test results generated by MLMARK … (This does not include academic research.)

MLMark from EMBC

Scoring

- Throughput (fps)
  - throughput = X iterations * Y batch sizes / total time.
- Latency
  - Time it takes to process a single input for a single iteration
  - Use 95th percentile of all iterations in ms
  - latency mode forces a batch size one and a concurrency of one.
- Accuracy
  - For Resnet and MobileNet: Top-1 and Top-N accuracy
  - For SSD its IOU (Intersection over Union) mAP (Mean Average Precision)
  - All floating-point results are reported with three significant figures (not fixed decimal points).

MLPerf

https://github.com/mlperf/inference_policies/blob/master/inference_rules.adoc

- Rules:
  - System and framework must be available
  - Benchmark implementations must be shared (open source)
  - Replicability is mandatory
- Scenarios
  - Single Stream and Multi-stream
- Divisions
  - Open and Closed
- Reporting Framework
  - Must integrate “loadgen” with the runtime
- Audits
  - Still under discussion

MLPerf

Models

- Vision
  - Resnet50-v1.5
  - MobileNets-v1 224
  - SSD-ResNet34
  - SSD-MobileNets-v1
- Language
  - GMNT

- fp32 and some TF quantized models available
- Can used quantization tools on the networks, no re-training
- Quantized Accuracy must be within within 1% of fp32 models

Artificial Intelligence Industry Alliance (AIIA) AIBench

https://github.com/AIIABenchmark/AIIA-DNN-benchmark

- AIBench supports several deep learning frameworks (SNPE, HIAI, TENGINE and TFLite)
- Object_Classification
  - Mobilenetv2 / Resnet101 / VGG16 / Inceptionv3
- Object Detection
  - ssd_mobilenetv1 / ssd_mobilenetv2 / ssd_vgg16
- Image_Super_Resolution
  - Vdsr (Used for enlarging an image)
- Image_Segmentation
  - fcn
- Face_Recognition
  - VGG16 (~528 MB)

Open Issues

Lots of Open Issues

- Active research areas to prune, transform, re-train and optimize models
  - Benchmarks need an apples to apples comparison
  - Are the techniques used open or closed? What is allowed? Auditability and reproducibility
  - Quantized models vs fp models
  - Runtimes vs compiled models
- What is important to measure?
  - TOPS? TOPS/Watt? TOPS/$? TOPS/sqmm?
    - Even if TOPS are high, does YOUR model meet KPIs? FPS/Watt?
  - Accuracy? Accuracy/speed?
  - How do you measure power used?
- Device Categories
  - Many fp32 models (and even many quantized models) are too big for constrained devices

Testing Models vs Use Cases

- Today’s Benchmarks are Model Based
  - Model manipulation techniques must be “available” and useable for benchmarks
  - What manipulations are allowed? Re-training? Preprocessing, post processing?
  - Auditability and reproducibility
  - Typically PyTorch or TensorFlow models…. ONNX?
- Use cases
  - Object detection
  - Face detection
  - Hot word detection
  - How do you handle scoring for accuracy vs speed tradeoffs? Weighting different tests/models?
- Current Approach Ignores Platform Capabilities
  - Pre-processing capabilities of device
  - Concurrency of multiple networks

Summary

- Existing Benchmarks, don’t address the range of IoT devices doing AI
- The models relevant for IoT may be different
- Quantized models are important for IoT, and improving/maintaining quantized accuracy
- TOPS/Watt or FPS/Watt more relevant than TOPS in many cases
- Too much customization of models doesn’t produce meaningful benchmarks
- The device’s concurrent capabilities may outweigh single model performance

Thank you

Follow us on:

For more information, visit us at:

www.qualcomm.com & www.qualcomm.com/blog

Nothing in these materials is an offer to sell any of the components or devices referenced herein.

©2018-2019 Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. Other products and brand names may be trademarks or registered trademarks of their respective owners.

References in this presentation to "Qualcomm" may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable. Qualcomm Incorporated includes Qualcomm's licensing business, QTL, and the vast majority of its patent portfolio. Qualcomm Technologies, Inc., a wholly-owned subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of Qualcomm's engineering, research and development functions, and substantially all of its product and services businesses, including its semiconductor business, QCT.