The Model-Efficiency Pipeline
Enabling deep learning inference at the edge

Bert Moons,
Engineer, Senior
Qualcomm Technologies Netherlands B.V.
Qualcomm AI Research
Advancing research to make AI ubiquitous

We are creating platform innovations to scale AI across the industry
Agenda

- Energy-Efficient machine learning and the computational budget gap
- The Model-Efficiency Pipeline reduces the cost of on-device inference

Qualcomm Innovation Center, Inc. open sources through AI Model Efficiency Toolkit (AIMET)

- What’s next in energy-efficient AI
AI is being used all around us
increasing productivity, enhancing collaboration, and transforming industries

AI video analysis is on the rise
Trend toward more cameras, higher resolution, and increased frame rate across devices
Deep neural networks are energy hungry and growing fast

AI is being powered by the explosive growth of deep neural networks

2025: N = 100T = 10^{14}

2021: Extremely large neural networks (N=1.6T)

2017: Very large neural networks (N=137B)

2013: Google/Y! (N=+/− 1B)

2009: Hinton’s Deep Belief Net (+/− N=10M)

1943: First NN (+/− N=10)

1988: NetTalk (+/− N=20K)


10^0 10^2 10^4 10^6 10^8 10^{10} 10^{12} 10^{14}

Deep neural networks are energy hungry and growing fast

AI is being powered by the explosive growth of deep neural networks

2025: Increasingly large and complex neural networks for Natural Language Processing, Image and Video Processing

Source: Welling
Power and thermal efficiency are essential for on-device AI.

The challenge of AI workloads:
- Very compute intensive
- Large, complicated neural network models
- Complex concurrencies
- Real-time
- Always-on

Constrained mobile environment:
- Must be thermally efficient for sleek, ultra-light designs
- Requires long battery life for all-day use
- Storage/memory bandwidth limitations
Trend 1: Increasingly complex Neural Networks:
Image, NLP, video, ensembles, higher resolution, ...

The Deep Learning Budget Gap
The Deep Learning Budget Gap

Trend 1: Increasingly complex Neural Networks: Image, NLP, video, ensembles, higher resolution, ...

Trend 2: Faster, more efficient hardware platforms close the Budget Gap

Mobile AIP in 1W range

Budget Gap

Neural Network Applications

Computational Budget [ops/s]

2018 2020 2022

248 FPS\textsuperscript{1*}

902 FPS\textsuperscript{2*}

1: Qualcomm® Hexagon™ 698 DSP in the Qualcomm® Snapdragon™ 865 running on the ASUS ROG Phone 3
2: Qualcomm® Hexagon™ 780 DSP in the Qualcomm® Snapdragon™ 780 running on the OnePlus 9 Pro

Qualcomm Hexagon and Qualcomm Snapdragon are products of Qualcomm Technologies, Inc. and/or its subsidiaries.

The Deep Learning Budget Gap

**Trend 1:**
Increasingly complex Neural Networks: Image, NLP, video, ensembles, higher resolution, ...

**Trend 2:**
Faster, more efficient hardware platforms close the Budget Gap

Tiny AIP in 10mW range

Neural Network Applications

Computational Budget [ops/s]
The Deep Learning Budget Gap

**Trend 1:** Increasingly complex Neural Networks: Image, NLP, video, ensembles, higher resolution, …

**Trend 2:** Faster, more efficient hardware platforms close the Budget Gap

**Trend 3:** Faster, optimized Neural Networks and Applications close the Budget Gap

---

**Neural Network Applications**

**Efficient Neural Networks**

**Mobile AIP in 1W range**

2018

2020

2022

248 FPS\(^1\)*

902 FPS\(^2\)*

---

1: Qualcomm® Hexagon™ 698 DSP in the Qualcomm® Snapdragon™ 865 running on the ASUS ROG Phone 3
2: Qualcomm® Hexagon™ 780 DSP in the Qualcomm® Snapdragon™ 780 running on the Oneplus 9 Pro

Qualcomm Hexagon and Qualcomm Snapdragon are products of Qualcomm Technologies, Inc. and/or its subsidiaries.

Trend 3: The Model-Efficiency Pipeline
The Model-Efficiency Pipeline

Multiple axes to shrink AI models and run them efficiently on hardware

- Neural architecture search
- Pruning and Model Compression
- Accurate Quantization
Neural Architecture Search: automated design of on-device optimal networks

Training networks from scratch is expensive!

>2 GPU months to train a single SotA network on ImageNet
~4k USD per network using commercial cloud services
Training networks from scratch is expensive!

Neural Architecture Search: automated design of on-device optimal networks

>2 GPU months to train a single SotA network on ImageNet
~4k USD per network using commercial cloud services

Manual network design requires training many networks from scratch for every device

Spec A, Platform A
Spec B, Platform B
Spec C, Platform C
Training networks from scratch is expensive!

Neural Architecture Search: automated design of on-device optimal networks

>2 GPU months to train a single SotA network on ImageNet

~4k USD per network using commercial cloud services

Manual network design requires training many networks from scratch for every device

Solution

Cheap, scalable Neural Architecture Search reduces design and training costs of networks optimized for specific devices
Existing NAS solutions do not address all the challenges

- **Lack diverse search**
  Hard to search in diverse spaces, with different block-types, attention, and activations
  Repeated training for every new scenario

- **High cost**
  Brute force search is expensive
  >40,000 epochs per platform

- **Do not scale**
  Repeated training for every device
  >40,000 epochs per platform

- **Unreliable hardware models**
  Requires differentiable cost-functions
  Repeated training phase for every new device
Introducing new AI research

DONNA
Distilling Optimal Neural Network Architectures

Efficient NAS with hardware-aware optimization

Finds pareto-optimal architectures in terms of accuracy-latency at low cost

Diverse search to find the best models
Supports diverse spaces with different cell-types, attention, and activation functions (ReLU, Swish, etc.)

Low cost
Low start-up cost equivalent to training 2-10 networks from scratch

Scalable
Scales to many hardware devices at minimal cost

Reliable hardware measurements
Uses direct hardware measurements instead of a potentially inaccurate hardware model

Distilling Optimal Neural Networks: Rapid Search in Diverse Spaces (Moons, Bert, et al., arXiv 2020)
**DONNA 4-step process**

**Objective:** Build accuracy model of search space once, then deploy to many scenarios

*Bert Moons et al, “Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20*

---

**Define reference and search space once**

**Define backbone:**
- Fixed channels
- Head and Stem

**Varying parameters:**
- Kernel Size
- Expansion Factors
- Network depth
- Network width
- Attention/activation
- Different efficient layer types
Define reference architecture and search-space once
A diverse search space is essential for finding optimal architectures with higher accuracy

Select reference architecture
The largest model in the search-space

Chop the NN into blocks
Fix the STEM, HEAD, # blocks, strides, # channels at block-edge

Choose search space
Diverse factorized hierarchical search space, including variable cell-types, kernel-size, expansion-rate, depth, # channels, activation, attention

Choose diverse search space
- **Kernel**: 3, 5, 7
- **Expand**: 2, 3, 4, 6
- **Depth**: 1, 2, 3, 4
- **Activation**: ReLU/Swish
- **Cell type**: grouped, DW, ...
- **Width scale**: 0.5x, 1.0x
- **Attention**: SE, no SE
- **Conv 3x3s2**: ch=32
- **Conv 1x1**: ch=1536
Define reference architecture and search-space once

Some example blocks in the shared search space: BasicBlocks, ShiftNets, MobileConv, Squeeze-and-Excitation

Choose search space

Examples of variable cell types that can be combined in a single search space


Define reference architecture and search-space once
Some example blocks in the shared search space: BasicBlocks, ShiftNets, MobileConv, Squeeze-and-Excitation

Choose search space
Two example models, pareto-optimal on a desktop GPU

Model A, @73% ImageNet top-1

Model B, @79.5% ImageNet top-1
DONNA 4-step process

Objective: Build accuracy model of search space once, then deploy to many scenarios

Bert Moons et al, “Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20
Build accuracy predictor via Blockwise Knowledge Distillation once

Low-cost hardware-agnostic training phase

**Block library**
- Pretrain all blocks in search-space through blockwise knowledge distillation

**Architecture library**
- Quickly finetune a representative set of architectures

**Accuracy predictor**
- Fit linear regression model

**Block pretrained weights**
- Fast block training
- Trivial parallelized training
- Broad search space

**Block quality metrics**
- Finetune sampled networks
- Fast network training
- Only 20-30 NN required

**Finetuned architectures**
- Linear Regression Model
- Accurate predictions
- Up to 10x improved ranking vs DARTS
Build accuracy predictor via BKD once
Low-cost hardware-agnostic training phase

State-of-the-art references achieve up to 0.65 KT ranking*

DONNA achieves up to 0.91 KT on basic test sets and reliably extends to test sets with previously unseen cell-types: 0.8KT

DONNA = Bert Moons et al, "Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20
DONNA 4-step process

Objective: Build accuracy model of search space once, then deploy to many scenarios

Bert Moons et al, “Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20

A Define reference and search space once

Define backbone:
- Fixed channels
- Head and Stem

Varying parameters:
- Kernel Size
- Expansion Factors
- Network depth
- Network width
- Attention/activation
- Different efficient layer types

B Build accuracy model via Knowledge Distillation (KD) once

Approximate ideal projections of a reference model through KD

Use quality of blockwise approximations to build accuracy model

C Evolutionary search in 24h

Predicted accuracy vs. HW latency

Different compiler versions, different image sizes

Scenario-specific search

DONNA 4-step process

Objective: Build accuracy model of search space once, then deploy to many scenarios

Bert Moons et al, “Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20
Evolutionary search with real hardware measurements

Scenario-specific search allows users to select optimal architectures for real-life deployments.

Quick turnaround time
Results in +/- 1 day using one measurement device.

Accurate scenario-specific search
Captures all intricacies of the hardware platform and software – e.g. run-time version or devices.

NSGA-II
evolutionary sampling algorithm

Task accuracy predictor

Target HW

Measured latency on device

Predicted task accuracy

End-to-end model

NSGA: Non-dominated Sorting Genetic Algorithm
**DONNA 4-step process**

**Objective:** Build accuracy model of search space once, then deploy to many scenarios

*Bert Moons et al, “Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20*

**A** Define reference and search space once

- Define backbone:
  - Fixed channels
  - Head and Stem

- Varying parameters:
  - Kernel Size
  - Expansion Factors
  - Network depth
  - Network width
  - Attention/activation
  - Different efficient layer types

**B** Build accuracy model via Knowledge Distillation (KD) once

Approximate ideal projections of a reference model through KD

Use quality of blockwise approximations to build accuracy model

**C** Evolutionary search in 24h

Evolutionary specific search

Use KD-initialized blocks from step B to finetune any network in the search space in 15-50 epochs instead of 450

**D** Sample and finetune

MSE 1 → MSE 2 → MSE 3 → MSE 4 → MSE 5

Use KD-initialized blocks from step B to finetune any network in the search space in 15-50 epochs instead of 450.
DONNA finds state-of-the-art networks for on-device scenarios

Quickly optimize and make tradeoffs in model accuracy with respect to the deployment conditions that matter

Qualcomm Adreno 660 GPU in the Snapdragon 888 running on the Samsung Galaxy S21.

Qualcomm Hexagon 780 Processor in the Snapdragon 888 running on the Samsung Galaxy S21.

Qualcomm Adreno is a product of Qualcomm Technologies, Inc. and/or its subsidiaries.
DONNA efficiently finds optimal models over diverse scenarios

Cost of training is a handful of architectures*

<table>
<thead>
<tr>
<th>Method</th>
<th>Granularity</th>
<th>Macro-diversity</th>
<th>Search-cost / scenario 1 scenario, 10 models/scenario [FS(_e)]</th>
<th>Search-cost / scenario ∞ scenarios, 10 models/scenario [FS(_e)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFA</td>
<td>Layer-level</td>
<td>Fixed</td>
<td>2.7(10^x)[0.05-0.15]</td>
<td>0.5 - 1.5</td>
</tr>
<tr>
<td>DNA</td>
<td>Layer-level</td>
<td>Fixed</td>
<td>1.5(10^x)1</td>
<td>10</td>
</tr>
<tr>
<td>M NasNet</td>
<td>Block-level</td>
<td>Variable</td>
<td>90(10^x)1</td>
<td>100</td>
</tr>
<tr>
<td>This Work</td>
<td>Block-level</td>
<td>Variable</td>
<td>9(10^x)0.1</td>
<td>1</td>
</tr>
</tbody>
</table>

Good
OK
Not good

DONNA = MnasNet-level diversity at 100x lower cost

*Training 1 model from scratch = 450 epochs

OFA = Han Cai, et al, "Once For All: Train One Network and Specialize it for Efficient deployment", ICLR2020
DNA = Changlin Li, "Blockwisely Supervised Neural Architecture Search with Knowledge Distillation", CVPR20
This work = Bert Moons et al, "Distilling Optimal Neural Networks: rapid search in diverse spaces, Arxiv20
DONNA applies directly to downstream tasks and non-CNN neural architectures without conceptual code changes.
The Model-Efficiency Pipeline

Multiple axes to shrink AI models and run them efficiently on hardware

- Neural architecture search
- Accurate Quantization
- Pruning and Model Compression
Unstructured Pruning of Neural Networks

Pruning removes unnecessary connections in the neural network. Unstructured pruning is non-trivial to accelerate on parallel hardware.

Song Han, et al, “Deep compression: compressing deep Neural Networks with Pruning, Trained Quantization and Huffman Coding”, NIPS2015
Structured compression through low rank approximations

Structured mathematical decompositions (SVD, CP, Tucker-II, Tensor-train,...) are easier to accelerate on parallel hardware

Structured compression through low rank approximations

- **(Structured) Channel Pruning** and **Spatial-SVD** typically work best.
- 50% compression @0.3% accuracy loss for ResNet-50

Andrey Kuzmin, et al, "Taxonomy and Evaluation of Structured Compression of Convolutional Neural Networks", Arxiv 2019
The Model-Efficiency Pipeline

Multiple axes to shrink AI models and run them efficiently on hardware

- Neural architecture search
- Pruning and Model Compression
- Accurate Quantization
What is neural network quantization?

For any given trained neural network:
• Store weights in n bits
• Compute calculations in n bits

Benefits
• Reduced memory usage
• Reduced energy usage
• Lower latency

Quantization example

24 bits per pixel
### Data-free quantization

Baseline training-free method with equalization and bias correction

- No training
- Data free

### AdaRound

Outperform rounding to nearest

- No training
- Minimal unlabeled data

### Bayesian bits

Automated mixed-precision

- Training required
- Training data required
- Jointly learns bit-width precision and pruning

---

**SOTA 8-bit results**

- Accuracy drop for MobileNet V2 against FP32 model: <1%

  Data-Free Quantization Through Weight Equalization and Bias Correction (Nagel, van Baalen, et al., ICCV 2019)

**SOTA 4-bit weight results**

- Accuracy drop for MobileNet V2 against FP32 model: <2.5%

  Up or Down? Adaptive Rounding for Post-Training Quantization (Nagel, Amjad, et al., ICML 2020)

**SOTA mixed-precision results**

- Accuracy drop for MobileNet V2 against FP32 model for mixed precision model with computational complexity equivalent to a 4-bit weight model: <1%

  Bayesian Bits: Unifying Quantization and Pruning (van Baalen, Louizos, et al., NeurIPS 2020)
Adaround  Up or Down? Adaptive Rounding for Post-Training Quantization (Nagel, Amjad, et al., ICML 2020)

• Traditional post-training weight quantization uses rounding to nearest:

• However, rounding-to-nearest is not optimal

<table>
<thead>
<tr>
<th>Rounding Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest</td>
<td>52.29</td>
</tr>
<tr>
<td>Floor / Ceil</td>
<td>00.10</td>
</tr>
<tr>
<td>Stochastic</td>
<td>52.06±5.52</td>
</tr>
<tr>
<td>Stochastic (best)</td>
<td>63.06</td>
</tr>
</tbody>
</table>

4-bit weight quantization of 1st layer of Resnet18, tested on ImageNet.
Up or Down?
How can we systematically find the best rounding choice?
AdaRound: learning to round

- Minimize per-layer L2 loss of output features
  \[ \arg \min_V \| Wx - \tilde{W}x \|_F^2 + \lambda f_{reg}(V) \]
  \[ \tilde{W} = s \cdot \text{clip} \left( \frac{W}{s} + h(V), n, p \right) \]
  round down + learned value between [0,1]

- Regularization:
  \[ f_{reg}(V) = \sum_{i,j} 1 - |2h(V_{i,j}) - 1|^\beta \]
Comparison to literature

Setting a new SOTA for 4-bit post-training weight quantization

<table>
<thead>
<tr>
<th>Optimization</th>
<th>#bits W/A</th>
<th>Resnet18</th>
<th>Resnet50</th>
<th>InceptionV3</th>
<th>MobilenetV2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full precision</td>
<td>32/32</td>
<td>69.68</td>
<td>76.07</td>
<td>77.40</td>
<td>71.72</td>
</tr>
<tr>
<td>DFQ (Nagel et al., 2019)</td>
<td>8/8</td>
<td>69.7</td>
<td>-</td>
<td>-</td>
<td>71.2</td>
</tr>
<tr>
<td>DFQ (our impl.)</td>
<td>4/8</td>
<td>38.98</td>
<td>52.84</td>
<td>-</td>
<td>46.57</td>
</tr>
<tr>
<td>Bias corr (Banner et al., 2019)</td>
<td>4*/8</td>
<td>67.4</td>
<td>74.8</td>
<td>59.5</td>
<td>-</td>
</tr>
<tr>
<td>AdaRound w/ act quant</td>
<td>4/8</td>
<td><strong>68.55±0.01</strong></td>
<td><strong>75.01±0.05</strong></td>
<td><strong>75.72±0.09</strong></td>
<td><strong>69.25±0.06†</strong></td>
</tr>
</tbody>
</table>

Table 7. Comparison among different post-training quantization strategies in the literature. We report results for various models in terms of ImageNet validation accuracy (%). *Uses per channel quantization. †Using CLE (Nagel et al., 2019) as preprocessing.
Tools are open-sourced through AIMET

github.com/quic/aimet

github.com/quic/aimet-model-zoo
AIMET
State-of-the-art quantization and compression techniques

AIMET Model Zoo
Accurate pre-trained 8-bit quantized models

github.com/quic/aimet

github.com/quic/aimet-model-zoo

Join our open-source projects
AIMET Model Zoo includes popular quantized AI models
Accuracy is maintained for INT8 models – less than 1% loss*

Tensorflow

<table>
<thead>
<tr>
<th>Model</th>
<th>FP32 Accuracy</th>
<th>INT8 Accuracy</th>
<th>Loss in accuracy*</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50 (v1)</td>
<td>75.21%</td>
<td>74.96%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>MobileNet-v2-1.4</td>
<td>75%</td>
<td>74.21%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>EfficientNet Lite</td>
<td>74.93%</td>
<td>74.99%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>SSD MobileNet-v2</td>
<td>0.2469</td>
<td>0.2456</td>
<td>&lt;1%</td>
</tr>
</tbody>
</table>

Pytorch

<table>
<thead>
<tr>
<th>Model</th>
<th>FP32 Accuracy</th>
<th>INT8 Accuracy</th>
<th>Loss in accuracy*</th>
</tr>
</thead>
<tbody>
<tr>
<td>MobileNetV2</td>
<td>7167%</td>
<td>71.14%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>EfficientNet-lite0</td>
<td>75.42%</td>
<td>74.44%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>DeepLabV3+</td>
<td>72.62%</td>
<td>72.22%</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>MobileNetV2-SSD-Lite</td>
<td>68.7%</td>
<td>68.6%</td>
<td>&lt;1%</td>
</tr>
</tbody>
</table>

*: Comparison between FP32 model and INT8 model quantized with AIMET.
For further details, check out: https://github.com/quic/aimet-model-zoo/
What’s next in efficient on-device AI
Ultimately limited gains from NAS, compression, uniform quantization

Current tools optimize existing architectures, leading to 1-3x gains over standard networks on device.

NAS

Compression

4-8b established

Mobile SoC throughput\(^1\)

(Adreno 660 GPU)

<table>
<thead>
<tr>
<th>Mobile SoC throughput(^1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>224x224 images</td>
</tr>
</tbody>
</table>

8-bit Integer

up to 16X

4-bit Integer

up to 64X

1 Qualcomm Adreno 660 GPU in the Snapdragon 888 running on the Samsung Galaxy S21

What’s next in efficient AI models?

Baseline Models

8bit Baseline Pareto Front

HW-Aware 8bit NAS, Compressed Pareto Front

1.2-3x

Task Quality

Throughput

What’s next?
Mixed-Precision Quantized NAS
Mixed Precision outperforms uniform quantization
Bayesian Bits: Neural Networks can be optimized for mixed-precision.

During training, the network automatically finds the optimal trade-off between network complexity and accuracy

Many Ways to gain from mixed-precision
An academic system level example

• DVAFS: DVAS + subword parallelism  op/J $10x$ @ 2b vs 8b
Mixed Precision Quantized NAS

- APQ builds on top of OFA
- +/- 1%, or 2.2x BOPS gains expected through joint NAS and Quantization

What’s next in efficient AI models?

Task Quality or Accuracy

Throughput

8bit Baseline Pareto Front

8bit NAS, Compressed

<8bit mixed-precision NAS

1.2-3x

1-2x
Conditional networks
Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

Classification: some samples are easier than others

Copyright Pixel Addict and Doyle (CC BY-ND 2.0), no changes made
found through Huang, G, et al, "Multi-Scale Dense Networks for Resource Efficient Image Classification", ICLR2018
Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

Early exiting, some samples are easier than others

Huang, G, et al, "Multi-Scale Dense Networks for Resource Efficient Image Classification", ICLR2018
Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

Segmentation: **backgrounds** are **abundant** and **easy** to recognize

Detection: **Objects of interest** are relatively **rare**
Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

Dynamic Convolutions: exploiting spatial sparsity

Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

**Video:** Subsequent frames are correlated

Frame 1

Frame 2

Artur Andrzej, https://commons.wikimedia.org/wiki/File:Gda%C5%82sk_skrzy%C4%99zowanie_ulic_Grunwaldzkiej_i_Słowackiego.jpg

(Creative Commons CC0 1.0 Universal Public Domain Dedication), no changes made
Conditional computing as a complementary technique to NAS
Input-dependent network architectures spend less time on easier samples

Temporal Skip-Convolutions in video segmentation/detection

Conditional computing as a complementary technique to NAS

Input-dependent network architectures spend less time on easier samples

Conditional Early exiting in video/action recognition

Figure 1: Efficient video recognition by early exiting.

Figure 6: Accuracy vs. efficiency curves on ActivityNet.

What’s next in efficient AI models?

- **8bit Baseline Pareto Front**
- **8bit NAS, Compressed**
- **<8bit mixed-precision NAS**
- **Conditional computing**

Throughput vs. Task Quality or Accuracy

- 1.2-3x
- 1-2x
- 1-2x
What’s next in efficient AI models?

8bit Baseline Pareto Front

Diverse <8bit Mixed-Precision NAS + Conditional computing

10x
Overview

- Energy-Efficient machine learning and the computational budget gap

- The Model-Efficiency Pipeline reduces the cost of on-device inference

Qualcomm Innovation Center, Inc. open sources through AI Model Efficiency Toolkit (AIMET)

- What’s next in energy-efficient AI
Questions?

Connect with Us

www.qualcomm.com/ai

www.qualcomm.com/news/ong

@QCOMResearch

https://www.youtube.com/qualcomm?

http://www.slideshare.net/qualcommwirelessevolution
Thank you

Follow us on: f /twitter /linkedin /instagram
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.

©2018–2021 Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved.

Qualcomm, Snapdragon, Adreno, and Hexagon are trademarks or registered trademark of Qualcomm Incorporated. 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 our licensing business, QTL, and the vast majority of our patent portfolio. Qualcomm Technologies, Inc., a subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of our engineering, research and development functions, and substantially all of our products and services businesses, including our QCT semiconductor business.
Premier Sponsor
Automated TinyML

Zero-code SaaS solution

Create tiny models, ready for embedding, in just a few clicks!

Compare the benchmarks of our compact models to those of TensorFlow and other leading neural network frameworks.

Executive Sponsors
Optimized models for embedded

Optimized low-level NN libraries (i.e. CMSIS-NN)

RTOS such as Mbed OS

Arm Cortex-M CPUs and microNPUs

Application

Runtime (e.g. TensorFlow Lite Micro)

Profiling and debugging tooling such as Arm Keil MDK

Connect to high-level frameworks

Supported by end-to-end tooling

Connect to Runtime

Stay Connected

@ArmSoftwareDevelopers

@ArmSoftwareDev

Resources: developer.arm.com/solutions/machine-learning-on-arm

Arm: The Software and Hardware Foundation for tinyML
TinyML for all developers

Acquire valuable training data securely

Edge Device
Real sensors in real time
Open source SDK

Embedded and edge compute deployment options

Enrich data and train ML algorithms

Impulse
Test

Test impulse with real-time device data flows

www.edgeimpulse.com
Advancing AI research to make efficient AI ubiquitous

Power efficiency
- Model design, compression, quantization, algorithms, efficient hardware, software tool

Personalization
- Continuous learning, contextual, always-on, privacy-preserved, distributed learning

Efficient learning
- Robust learning through minimal data, unsupervised learning, on-device learning

Perception
- Object detection, speech recognition, contextual fusion

Reasoning
- Scene understanding, language understanding, behavior prediction

Action
- Reinforcement learning for decision making

A platform to scale AI across the industry

Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.
Syntiant Corp. is moving artificial intelligence and machine learning from the cloud to edge devices. Syntiant’s chip solutions merge deep learning with semiconductor design to produce ultra-low-power, high performance, deep neural network processors. These network processors enable always-on applications in battery-powered devices, such as smartphones, smart speakers, earbuds, hearing aids, and laptops. Syntiant's Neural Decision Processors™ offer wake word, command word, and event detection in a chip for always-on voice and sensor applications.

Founded in 2017 and headquartered in Irvine, California, the company is backed by Amazon, Applied Materials, Atlantic Bridge Capital, Bosch, Intel Capital, Microsoft, Motorola, and others. Syntiant was recently named a CES® 2021 Best of Innovation Awards Honoree, shipped over 10M units worldwide, and unveiled the NDP120 part of the NDP10x family of inference engines for low-power applications.

www.syntiant.com  @Syntiantcorp
Platinum Sponsors
Part of your life. Part of tomorrow.

www.infineon.com
Add Advanced Sensing to your Product with Edge AI / TinyML

Pre-built Edge AI sensing modules, plus tools to build your own

**Reality AI solutions**
- Prebuilt sound recognition models for indoor and outdoor use cases
- Solution for industrial anomaly detection
- Pre-built automotive solution that lets cars “see with sound”

**Reality AI Tools® software**
- Build prototypes, then turn them into real products
- Explain ML models and relate the function to the physics
- Optimize the hardware, including sensor selection and placement

https://reality.ai  info@reality.ai  @SensorAI  Reality AI
Gold Sponsors
Adaptive AI for the Intelligent Edge

LatentAI

Latentai.com
Build Smart IoT Sensor Devices From Data

SensiML pioneered TinyML software tools that auto generate AI code for the intelligent edge.

- End-to-end AI workflow
- Multi-user auto-labeling of time-series data
- Code transparency and customization at each step in the pipeline

We enable the creation of production-grade smart sensor devices.

sensiml.com
Silver Sponsors
Copyright Notice

The presentation(s) in this publication comprise the proceedings of tinyML® EMEA Technical Forum 2021. The content reflects the opinion of the authors and their respective companies. This version of the presentation may differ from the version that was presented at tinyML EMEA. 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