“Cutting the AI Power Cord: Technology to Enable True Edge Inference”
Kristopher Ardis and Robert Muchsel
Maxim Integrated
October 27, 2020
tinyML Talks Sponsors

Additional Sponsorships available – contact Bette@tinyML.org for info
WE USE AI TO MAKE OTHER AI FASTER, SMALLER AND MORE POWER EFFICIENT

Automatically compress SOTA models like MobileNet to <200KB with little to no drop in accuracy for inference on resource-limited MCUs

Reduce model optimization trial & error from weeks to days using Deeplite's design space exploration

Deploy more models to your device without sacrificing performance or battery life with our easy-to-use software

BECOME BETA USER bit.ly/testdeeplite
TinyML for all developers

Get your free account at http://edgeimpulse.com
Health sensors measure PPG and ECG signals critical to understanding vital signs. Signal chain products enable measuring even the most sensitive signals.

The biggest (3MB flash and 1MB SRAM) and the smallest (256KB flash and 96KB SRAM) Cortex M4 microcontrollers enable algorithms and neural networks to run at wearable power levels.

The new MAX78000 implements AI inferences at over 100x lower energy than other embedded options. Now the edge can see and hear like never before.
Qeexo AutoML for Embedded AI
Automated Machine Learning Platform that builds tinyML solutions for the Edge using sensor data

Key Features
- Wide range of ML methods: GBM, XGBoost, Random Forest, Logistic Regression, Decision Tree, SVM, CNN, RNN, CRNN, ANN, Local Outlier Factor, and Isolation Forest
- Easy-to-use interface for labeling, recording, validating, and visualizing time-series sensor data
- On-device inference optimized for low latency, low power consumption, and a small memory footprint
- Supports Arm® Cortex™- M0 to M4 class MCUs
- Automates complex and labor-intensive processes of a typical ML workflow – no coding or ML expertise required!

Target Markets/Applications
- Industrial Predictive Maintenance
- Smart Home
- Wearables
- Automotive
- Mobile
- IoT

QEEXO AUTOML: END-TO-END MACHINE LEARNING PLATFORM

For a limited time, sign up to use Qeexo AutoML at automl.qeexo.com for FREE to bring intelligence to your devices!
Reality AI Tools® software

- Automated Feature Exploration and Model Generation
- Bill-of-Materials Optimization
- Automated Data Assessment
- Edge AI / TinyML code for the smallest MCUs

Reality AI solutions

- Automotive sound recognition & localization
- Indoor/outdoor sound event recognition
- RealityCheck™ voice anti-spoofing

is for building products

https://reality.ai info@reality.ai @SensorAI Reality AI
SynSense (formerly known as aiCTX) builds ultra-low-power (sub-mW) sensing and inference hardware for embedded, mobile and edge devices. We design systems for real-time always-on smart sensing, for audio, vision, bio-signals and more.

https://SynSense.ai
tinyML Strategic Partners

arm

Additional Sponsorships available – contact Bette@tinyML.org for info
Call for Video ‘Posters’
5 minute presentations
Abstracts due October 31

Full program at www.tinyml.org/asia2020
Live online program daily at 9 am China time

Free event courtesy of our sponsors – register today!

Premier Sponsor

Additional sponsorships available - contact Bette@tinyml.org for info
## Next tinyML Talks

<table>
<thead>
<tr>
<th>Date</th>
<th>Presenter</th>
<th>Topic / Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuesday, November 10</td>
<td><strong>Ehsan Saboori</strong> Co-founder and CTO, Deeplite</td>
<td>Networks within Networks: Novel CNN design space exploration for resource limited devices</td>
</tr>
<tr>
<td></td>
<td><strong>Alexander Samuelsson</strong> CTO and co-founder, Imagimob</td>
<td>How to build advanced hand-gestures using radar and tinyML</td>
</tr>
</tbody>
</table>

Webcast start time is 8 am Pacific time  
Each presentation is approximately 30 minutes in length

Please contact talks@tinyml.org if you are interested in presenting
Reminders

Slides & Videos will be posted tomorrow

tinyml.org/forums     youtube.com/tinyml

Please use the Q&A window for your questions
Kristopher Ardis

**Kris Ardis** is an Executive Director in the Micros, Security & Software Business Unit at Maxim Integrated. He began his career with Maxim in 1997 as a software engineer and holds two U.S. patents. In his current role, Ardis is responsible for Edge Artificial Intelligence accelerators, Secure and Low Power Microcontrollers, and Software Algorithms. He has a B.S. in Computer Science from the University of Texas at Austin.
Robert Muchsel

Robert Muchsel is the System Architect for Maxim’s new Embedded Machine Learning Accelerators. He has been with Maxim Integrated in Dallas, Texas since 2001.

With a degree in computer engineering from the Swiss Federal Institute of Technology in Zurich, Switzerland, Robert has worked on countless embedded applications and holds a variety of patents.
Cutting the AI Power Cord: Technology to Enable True Edge Inference
Kris Ardis, Robert Muchsel
An AI revolution is underway...

...allowing machines to see, hear and sense the world around them.
Gap Between Big Machines and Little Machines

“Unconstrained”
Today: self-driving

Cost, size, and power constraints
Today: simple wake words
Machine Learning Algorithms are Slow and Power Hungry on Existing Hardware

- If self-learning is not required, an algorithm can be trained on different hardware or offline with different energy and time budgets
  - Ignore learning for this discussion
- To apply the algorithm, computational effort is in **forward propagation**
  - On classic hardware, almost all spent in a triple nested (matrix) multiplication loop
  - Uses floating point – \( O(n^3) \) to \( O(n^{2.8}) \)
- Very energy intensive even with fast matrix multiply using integer math on DSP or GPU – large number of memory accesses

```c
// Main loop
for (l = 1; l < NLAYERS-1; l++) {
    // Compute z = w * a
    matrix_mul(&w[l-1], &a[l-1], &z[l]);

    // Add the bias values : z = w * a + b
    matrix_add(&b[l], &z[l]);

    // Compute a = g(z)
    nn_activate(z[l].elements, a[l].elements, lw[l]);
}

matrix_mul(matrix_f32_t *a, matrix_f32_t *b, matrix_f32_t *c) {
    uint32_t m = a->nrows;
    uint32_t n = a->ncols;
    uint32_t p = b->ncols;
    c->nrows = m;
    c->ncols = p;

    int i, j, k;
    for (i = 0; i < m; i++) {
        for (j = 0; j < p; j++) {
            f_t sum = 0;
            for (k = 0; k < n; k++) {
                sum += a->elements[i * n + k] * b->elements[k * p + j];
            }
            c->elements[i * p + j] = sum;
        }
    }
}
```

*Strassen's algorithm*
What do Edge Accelerators Need?

**Low Energy + Latency**
100× to 1000× better than standard Arm microcontroller with DSP
Enables AI at the edge

**Support for Any Application**
Fully programmable, application agnostic device
**Introducing the MAX78000**

**Ultra low power micro**
- ARM Cortex-M4F
  - 100 MHz, 16 KB cache
- RISC-V Smart DMA
  - 60 MHz, cache
- 4-ch DMA
- 512 KB Flash
- 128 KB SRAM

**SIMO/DVS**
- 4-ch DMA
- 3×I2S M/S
- 3×I2C (Hi-speed)
- 2×SPI M/S
- 4×Pulse train
- 1-Wire master
- Parallel camera
- SWD
- 8-ch 10-bit ADC
- 4 μPower Comparator

**Clocking**
- 100 MHz RO
- 60 MHz RO
- 8-30 kHz RO
- 32 kHz XO (RTC)
- 7.3728 MHz RO
- Ext. Square Wave (up to 80 MHz)

**External interfaces**
- I2S M/S
- 3×I2C (Hi-speed)
- 2×SPI M/S
- 4×Pulse train
- 1-Wire master
- Parallel camera
- SWD
- 3×UART
- 1×LP
- 9×Timers
- 6×32-bit, 2×LP, 2×Watchdog, 1×Wakeup
- 8-ch 10-bit ADC

**CNN accelerator**
- Parallel processors: 64
- Max layers: 32...64
- Max input/output channels in any layer: 1024
- Max weights: 432 KB
  - (up to 3.5 M weights)
- Data memory: 512 KB + 384 KB
- Max. input dimensions:
  - 1023 × 1023 (per channel, streaming)
  - 181 × 181 (per channel, preloaded)

1 – Up to 64 with pooling every other layer, up to 32 with no pooling
2 – Weights can be 1-bit, 2-bit, 4-bit, or 8-bit, selectable per layer

---

**Maxim Integrated**

20 | Maxim Integrated
MAX78000 CNN Accelerator

16 processors per group
× 4 groups
= 64 processors total
CNN Accelerator Data Flow

INPUT CHANNELS

OPTIONAL IN-FLIGHT OPERATIONS

CONVOLUTION

OUTPUT CHANNELS

next layer (up to 32)

unique configuration for each layer
(Some) Hardware Accelerator Features

- Conv2d (1×1, 3×3)
- Conv1d (1 to 9)
- ConvTranspose2d (3×3)
- AvgPool, MaxPool (up to 16×16)
- Flatten, Linear
- Activation: ReLU, Abs, None
- Elementwise (up to 16) add, sub, binary OR, binary XOR
- Sequence Pool / Eltwise / Conv / Act counts as one layer
- 1, 2, 4, and 8-bit weights selectable per layer, 8-bit bias (optional)
- 8-bit data (clipping at activation stage) with optional 32-bit output for last layer
- Output shift (<<, >>) per layer (before clipping)
- Padding 0, 1, or 2
- Stride up to [16, 16]
- RISC-V core as “Smart DMA”
- Streaming mode with FIFOs

Tradeoff: What’s not in hardware will have to run in (slow) software
System Energy: From Traditional Systems to MAX78000

• Accelerator drastically lowers CNN energy
• Input and data manipulation become much larger relative contributors to energy
• MAX78000 improves data loading, better algorithms can help with data manipulation
Introducing the MAX78000

**Ultra low power micro**

- ARM Cortex-M4F 100 MHz
- 16 KB cache
- RISC-V Smart DMA
- 60 MHz, cache
- 4-ch DMA
- 512 KB Flash
- 128 KB SRAM
- I²S M/S
- 3 × I²C (Hi-speed)
- 2 × SPI M/S
- 4 × Pulse train
- 1 × Wire master
- 1 × Power Comparator
- 3 × UART
- 1 × LP
- 9 × Timers
- 6 × Watchdog, 2 × LP, 1 × Wakeup
- 2 × LP
- 2 × Watchdog, 1 × Wakeup
- 32-bit, 2 × LP
- 2 × Wakeup
- 64-bit
- 10-bit ADC
- 3.3 V
- 1.8 V
- 1.5 V
- SWD
- Power
- Comparator

**External interfaces**

- Express loader and optimized bridge
- FIFOs for loading
- Streaming mode – much larger images and video

**Clocking**

- 100 MHz RO
- 60 MHz RO
- 8-30 kHz RO
- 32 kHz RO (RTC)
- 7.3728 MHz RO
- 32 kHz RO
- Ext. Square Wave (up to 80 MHz)

**CNN Accelerator**

- for Speech, Vision, Time Series

**Features**

- 1- to 8-bit weights – more complex networks
- Larger memories – more complex networks
- Streaming mode – much larger images and video
- Features for time series, 1D, upsampling, ...

**Energy_reduction**

**SRAM sleep and shutdown**

**Low power DMA**

**Integrated SIMO**
System Improvements

Dual FIFOs

- Single-port memories – accelerator would have to be disabled to write
- Four asynchronous FIFOs
  > Usable from ARM and RISC-V
- One synchronous “fast FIFO”
  > From RISC-V, same clock domain as CNN

Power Saving

“Quadrants” can be individually disabled
## Why is VGA hard?

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Reference</th>
<th>Memory for 128 ch. (max. intermediate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32×32</td>
<td>CIFAR-10</td>
<td>128 KB</td>
</tr>
<tr>
<td>160×160</td>
<td>Intel Movidius</td>
<td>3,200 KB</td>
</tr>
<tr>
<td>224×224</td>
<td>ImageNet</td>
<td>6,272 KB</td>
</tr>
<tr>
<td>320×240</td>
<td>QVGA</td>
<td>9,600 KB</td>
</tr>
<tr>
<td>640×480</td>
<td>VGA</td>
<td>38,400 KB</td>
</tr>
</tbody>
</table>

---

![Diagram showing convolution operations and resolutions](image-url)
MAX78000 Streaming

- Streaming with dynamic feedback allows MAX78000 to process frame sizes that by far exceed internal memory.
- Natural when input data is acquired from an image sensor that scans row-by-row.
- There are limitations for the network (e.g., no upsampling).

feedback between layers
### MAX78000 Performance

#### Inference Time ms

<table>
<thead>
<tr>
<th>Network</th>
<th>MACC</th>
<th>MAX78000</th>
<th>MAX32650</th>
<th>STM32F7</th>
</tr>
</thead>
<tbody>
<tr>
<td>KWS20</td>
<td>13,801,088</td>
<td>2.0ms, 0.14mJ</td>
<td>350ms, 8.37mJ</td>
<td>123ms, 47.5mJ&lt;sup&gt;3&lt;/sup&gt;</td>
</tr>
<tr>
<td>FaceID</td>
<td>55,234,560</td>
<td>13.89ms, 0.40mJ</td>
<td>1760ms, 42.1mJ</td>
<td>754ms, 348 + 116mJ&lt;sup&gt;4&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

#### Inference Energy mJ

1. KWS20: 2.0ms, 0.14mJ, 350ms, 8.37mJ, 123ms, 47.5mJ<sup>3</sup>
2. FaceID: 13.89ms, 0.40mJ, 1760ms, 42.1mJ, 754ms, 348 + 116mJ<sup>4</sup>

---

<sup>1</sup>28 billion operations/second

<sup>2</sup>ARM DSP with CMSIS-NN, running exact same INT8 network as MAX78000

<sup>3</sup>STMF722ZE, internal memories

<sup>4</sup>STMF746NG, external SDRAM at 25% typical active power
Energy efficient AI:
Software optimizations have multiplicative effect

Example:
- Instead of using the DSP to process audio, use Conv1d layers to “learn” the frequency transformation

Any improvement to the model, or to the training procedure will benefit the system!
- Manual model optimization, or
- Neural Architecture Search (NAS)
Development Flow

Machine Learning Experts

- Hardware-aware `nn.Module`
- Training/Test Data Set
- Quant-Aware Training
- Quantization
- Checkpoint or ONNX file
- Model Description ("Sidecar")
- MAX78000 Synthesis
- Embedded C Code

Embedded Engineers

- TensorFlow
- PyTorch Model
- Evaluation
- Input Data Sample
- Embedded C Code
- Checkpoint or ONNX file
- Model Description ("Sidecar")
- Quantization
- Training/Test Data Set
- Hardware-aware `nn.Module`
Thank you!

https://maximintegrated.com/MAX78000
Datasheet, documentation, eval kits...

https://github.com/MaximIntegratedAI
Documentation, tools, examples...

Or email kristopher.ardis@maximintegrated.com
Copyright Notice

This presentation in this publication was presented as a tinyML® Talks webcast. 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