tinyML Talks Sponsors

Additional Sponsorships available – contact Bette@tinyML.org for info
Arm: The Software and Hardware Foundation for tinyML

1. Connect to high-level frameworks

2. Supported by end-to-end tooling

3. Connect to Runtime

- Profiling and debugging tooling such as Arm Keil MDK
- Optimized models for embedded
- Optimized low-level NN libraries (i.e. CMSIS-NN)
- RTOS such as Mbed OS
- Arm Cortex-M CPUs and microNPUs

Stay Connected

@ArmSoftwareDevelopers
@ArmSoftwareDev

Resources: developer.arm.com/solutions/machine-learning-on-arm
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
- Automotive
- Smart Home
- Mobile
- Wearables
- IoT

## Qeexo AUTOML: END-TO-END MACHINE LEARNING PLATFORM

For a limited time, sign up to use Qeexo AutoML at [autml.qeexo.com](http://autml.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

https://reality.ai  info@reality.ai  @SensorAI  Reality AI
SynSense 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, IMUs, bio-signals and more.

https://SynSense.ai
<table>
<thead>
<tr>
<th>Date</th>
<th>Presenter</th>
<th>Topic / Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuesday, February 2</td>
<td>Martino Sorbaro</td>
<td>Always-on visual classification below 1 mW with spiking convolutional networks on Dynap™-CNN</td>
</tr>
</tbody>
</table>

Webcast start time is 8 am Pacific time

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

Slides & Videos will be posted tomorrow

Please use the Q&A window for your questions

tinyml.org/forums       youtube.com/tinyml
Lukas Geiger

Lukas Geiger is a deep learning researcher at Plumerai working on new training methods and architectures for improving accuracy and efficiency of Binarized Neural Networks (BNNs). He is the author of the open-source Larq training library and core developer of the Plumerai software stack for deploying BNNs on embedded platforms.
Running Binarized Neural Networks on Microcontrollers

Lukas Geiger - lukas@plumerai.com

TinyML Webinar - 19th January 2021
Why is TinyML not already everywhere?

- Very constrained computational budget
- Accuracy vs latency vs memory footprint trade-off
- Often only used for simple tasks
- Requires many different skill sets in the team
The road ahead

https://openai.com/blog/ai-and-compute/
Efficient ML covers the entire stack

**Hardware**

- Moore’s Law slows down
- Trend to more specialised Hardware (GPUs, TPUs, FPGAs)
- Custom instructions to accelerate ML Algorithms
Efficient ML covers the entire stack

**Hardware**
- Moore's Law slows down
- Trend to more specialised Hardware (GPUs, TPUs, FPGAs)
- Custom instructions to accelerate ML Algorithms

**ML Algorithms**
- Efficient Model Design
- Quantization
- Knowledge Distillation
- Pruning
- …
Efficient ML covers the entire stack

- Hardware
  - Moore's Law slows down
  - Trend to more specialised Hardware (GPUs, TPUs, FPGAs)
  - Custom instructions to accelerate ML Algorithms

- Software / Tools
  - TensorFlow / PyTorch
  - TFLite / TFLite Micro
  - MLIR / TVM / Glow

- ML Algorithms
  - Efficient Model Design
  - Quantization
  - Knowledge Distillation
  - Pruning
  - ...

→ Hardware, Software and Machine Learning Algorithms Co-design required to find the optimal solutions

Hooker, 2020 – The Hardware Lottery
Binarized Neural Networks (BNNs)
Going below 8-bit precision

32-bit

8-bit

4-bit

2-bit

1-bit

common
Memory reduction for different precisions

32-bit

8-bit

4-bit

2-bit

1-bit  →  32x less memory compared to 32-bit
Binarized Convolution

32-bit

\[ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \sum_{i,k} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = -5 \]

1-bit

\( \text{XNOR} \)

\[ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \sum_{i,k} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = -5 \]
Training Neural Networks

Conv. -> BatchNorm

Weight (32-bit)
Training Binarized Neural Networks

- On the forward pass, weights are binarized.
- On the backward pass, a Straight-Through Estimator approximates the gradient.

*Helwegen et al., 2019, NeurIPS – Latent Weights Do Not Exist: Rethinking Binarized Neural Network Optimization*
Open Source BNN Ecosystem

Larq Zoo
Ready to use pre-trained models

Larq
Intuitive and flexible extension of TensorFlow Keras

Larq Compute Engine
Simple deployment and fast inference on ARM Cortex-A

Geiger et al., 2020 – Larq: An Open-Source Library for Training Binarized Neural Networks

larq.dev  github.com/larq
import tensorflow as tf

model = tf.keras.models.Sequential(
    [
        tf.keras.layers.Dense(512, activation="relu"),
        tf.keras.layers.Dense(10, activation="softmax"),
    ]
)
import tensorflow as tf
import larq as lq

model = tf.keras.models.Sequential(
    [
        lq.layers.QuantDense(512, activation="relu"),
        lq.layers.QuantDense(10, activation="softmax"),
    ]
)
```python
import tensorflow as tf
import larq as lq

model = tf.keras.models.Sequential(
    [lq.layers.QuantDense(512,
       kernel_quantizer="ste_sign",
       kernel_constraint="weight_clip"),
     lq.layers.QuantDense(10,
       input_quantizer="ste_sign",
       kernel_quantizer="ste_sign",
       kernel_constraint="weight_clip",
       activation="softmax",)
    ]
)
```
Larq Compute Engine

- Highly optimized open source BNN inference engine based on TFLite
- End to end solution for research and benchmarking of BNNs
- Optimized for ARMv8a Cortex-A (e.g. Android phones, Raspberry PI)
Accelerating BNN research with Larq

Bannink et al., MLSys 2021 – Larq Compute Engine: Design, Benchmark, and Deploy State-of-the-Art Binarized Neural Networks
Running BNNs on Microcontrollers
Person Detection / Visual Wake Words

Chowdhery et al., 2019 – Visual Wake Words Dataset
Person Detection on Cortex-M4

- STMicroelectronics 32L4 R9 development board
- 120 MHz ARM Cortex-M4 core
- OmniVision 9655 1.3 Megapixel camera

- Great for development
- ARM Cortex-M4 widely used
Model Benchmark on Cortex-M4

![Graph showing model benchmark results on Cortex-M4. The x-axis represents latency (ms) with a range from 80 to 280, and the y-axis represents on-device VWM dataset test accuracy ranging from 0.815 to 0.850. There are two models compared: Plumerai BNN and MobileNet v1 0.25. The model size (MiB) is indicated with different shades of gray: 5, 10, 15, 20, and 25.]
Efficient ML covers the entire stack

- Hardware
- Software / Tools
- ML Algorithms
- Data
Model Accuracy ≠ Real World Performance

Important properties beyond accuracy:

- Consistency of prediction
- Generalization to different lighting
- Robustness against perturbations, shifts and artifacts
- Common research datasets do not capture real data distribution
- Models might be sensitive to camera specifics
Unit tests for Deep Learning Applications

1. Analysis and identification of failure cases
2. Collection and labeling of examples
3. Model training
4. Verify against previously collected test cases
Person Detection Networks

Public MobileNet Baseline

- Trained on COCO dataset
- TFLite Micro
- 713 ms latency
- 294 kB model size

Plumerai Binarized Neural Network

- Trained on proprietary data
- Plumerai BNN inference software stack leveraging TFLite Micro
- 858 ms latency
- 115 kB model size
Live Demo
Person Detection using BNNs

- Hardware
- Plumerai Inference Stack
- Plumerai BNNs
- Plumerai Data Pipeline
What’s next?

Hardware
- ARM Cortex-M4
- Plumerai IP-core for BNN inference on low-power FPGAs
- Extension for RISC-V with BNN instructions
- xcore.ai by XMOS

Tasks
- Person detection
- Person localisation with bounding boxes

Plumerai Inference Stack

Plumerai BNNS

Plumerai Data Pipeline
We bring deep learning to all edge devices by using highly accurate and efficient Binarized Neural Networks on cheap, low-power hardware.
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