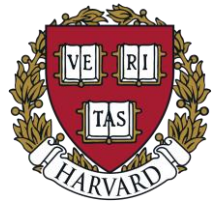
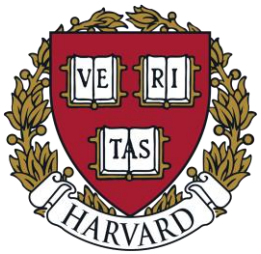
The Future of ML is Tiny and Bright

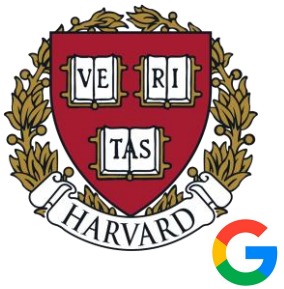
Challenges & Opportunities

Vijay Janapa Reddi
Harvard University

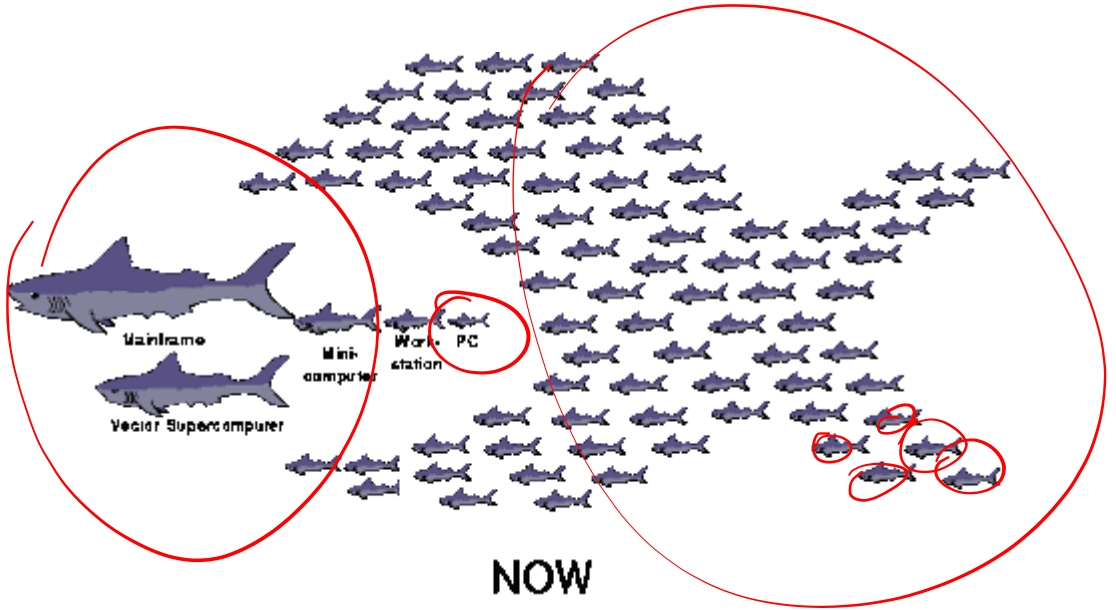


A Little... About Me





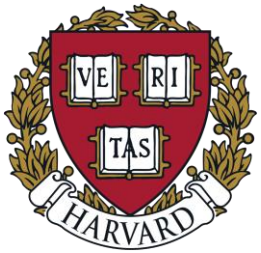
A Little… About Me

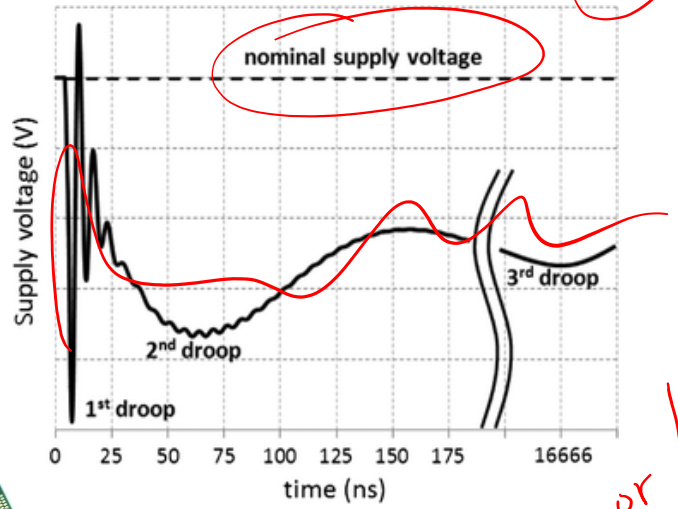


Try

A Little... About Me

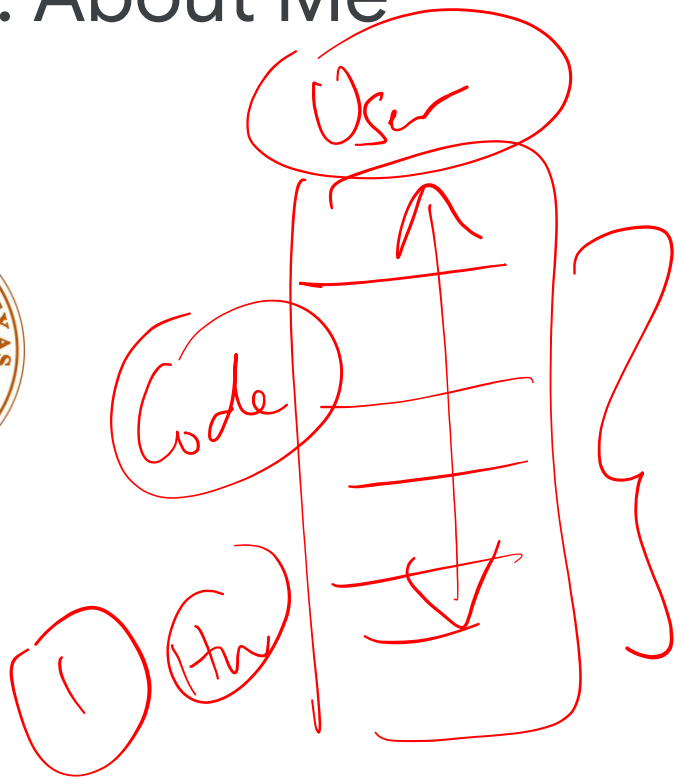
University of Colorado Boulder





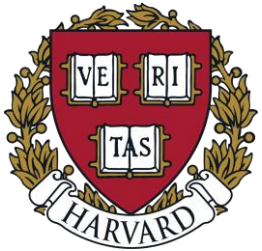
Codesign or body

A Little... About Me



D.C vs. Consulting + Scale

A Little... About Me



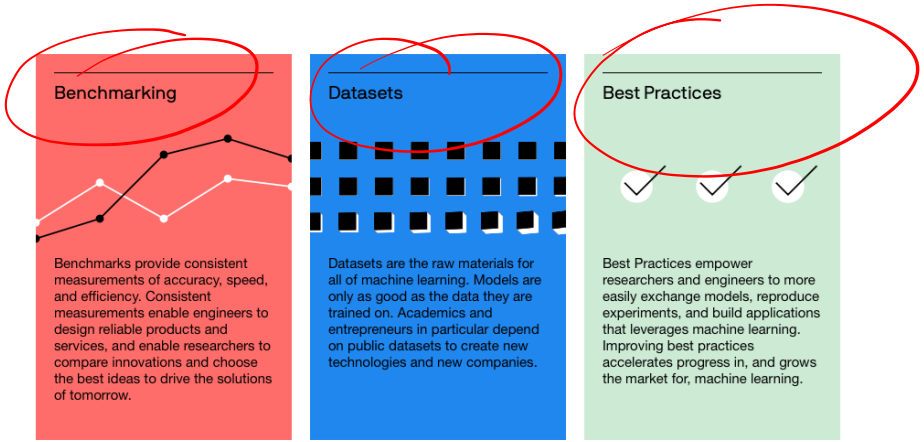
ML Commons


Alibaba Group
阿里巴巴集团
AMD
arm
Baidu 百度
CenTaur technology
cerebras
cisco
DELL EMC
Enflame
FACEBOOK AI
FURIOSA
GIGABYTE
Google
GML
GrAI Matter Labs
GRAPHCORE
地平线
Horizon Robotics
Hewlett Packard Enterprise
inspur
intel AI
KALRAY
LANDING AI
MEDIATEK
Microsoft
myrtle.ai
Nettrix 宁畅
NVIDIA
oppo
Qualcomm
Red Hat
SambaNova SYSTEMS
SAMSUNG Exynos
SYNTIANT
Tenstorrent
vmware
VMind
XILINX

MLCommons aims to accelerate machine learning innovation to benefit everyone.

MLCommons aims to accelerate machine learning innovation to benefit everyone. Machine learning has tremendous potential to save lives in areas like healthcare and automotive safety and to improve information access and understanding through technologies like voice interfaces, automatic translation, and natural language processing. However, machine learning is completely unlike conventional software -- developers train an application rather than program it -- and requires a whole new set of techniques analogous to the breakthroughs in precision measurement, raw materials, and manufacturing that drove the industrial revolution.

MLCommons aims to answer the needs of the nascent machine learning industry through open, collaborative engineering in three areas:



Where is ML going next?

The Future of ML is Tiny and Bright

What is Tiny Machine Learning (**TinyML**)?

What is Tiny Machine Learning (**TinyML**)?

TinyML → Fastest-growing field of **ML** 

What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

On-device sensor analytics

What is Tiny Machine Learning (**TinyML**)?

TinyML

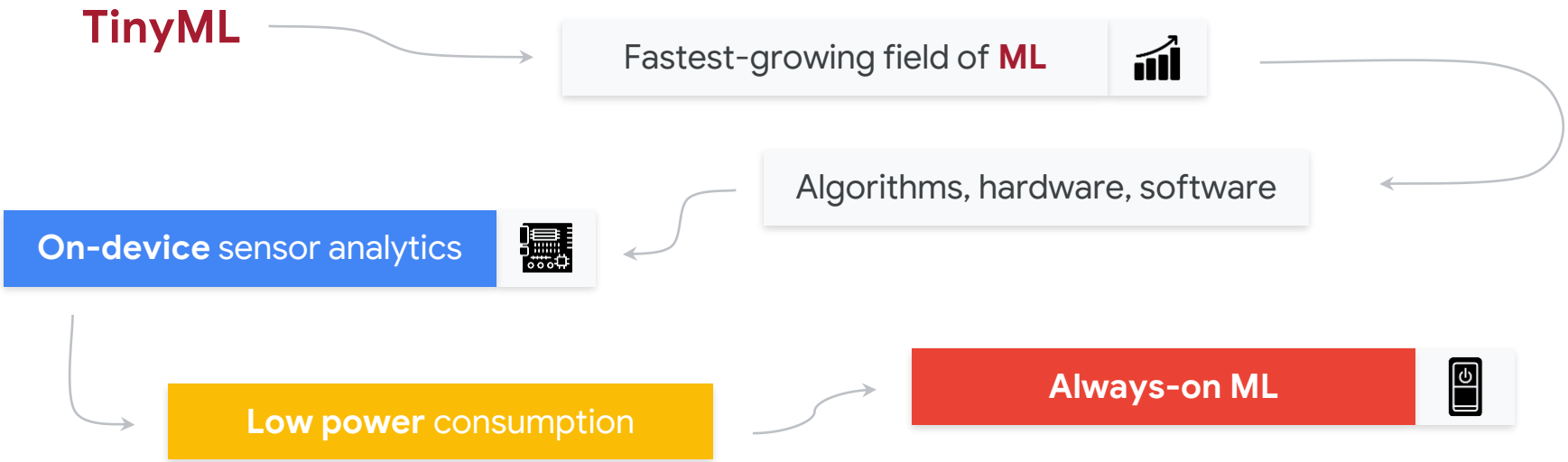
Fastest-growing field of **ML**

Algorithms, hardware, software

On-device sensor analytics

Low power consumption

What is Tiny Machine Learning (**TinyML**)?



What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

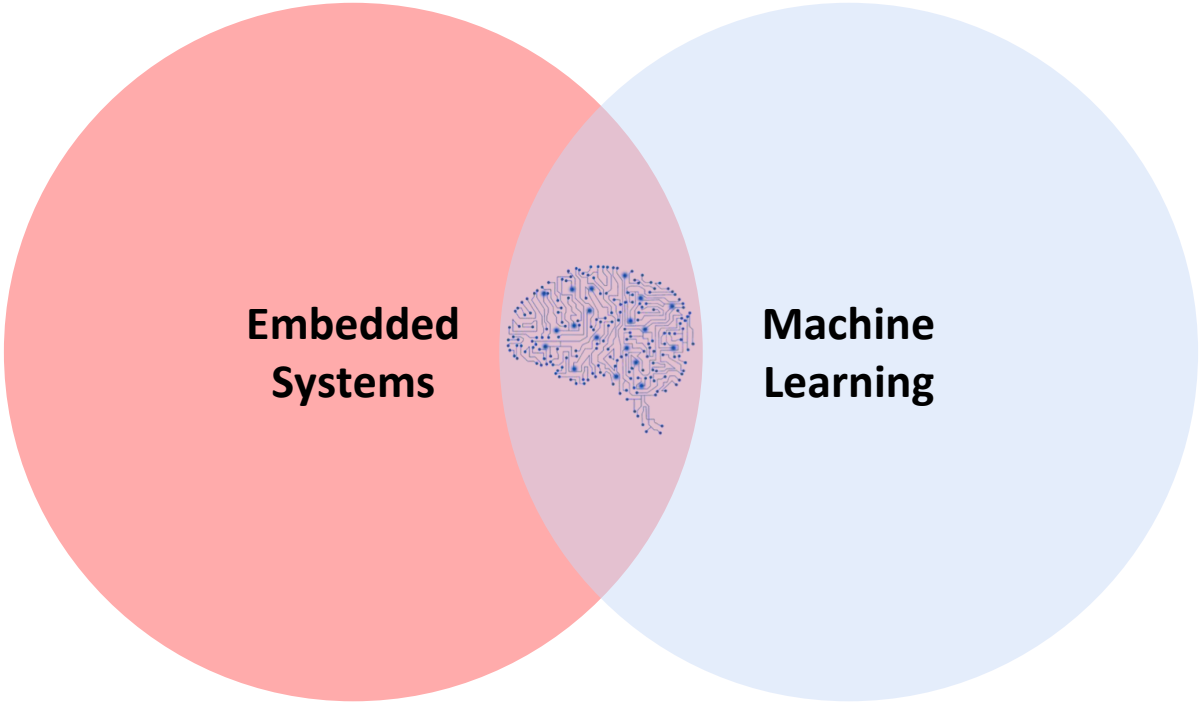
On-device sensor analytics

Low power consumption

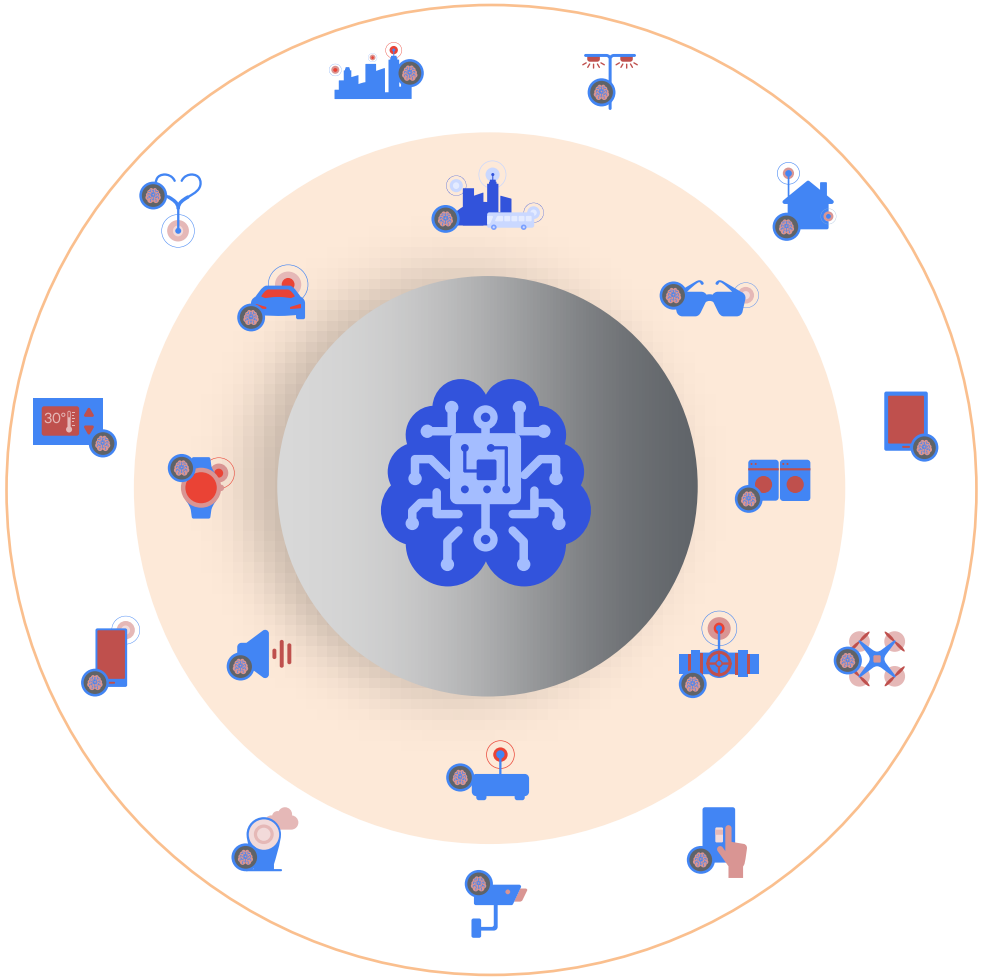
Always-on ML

Battery-operated

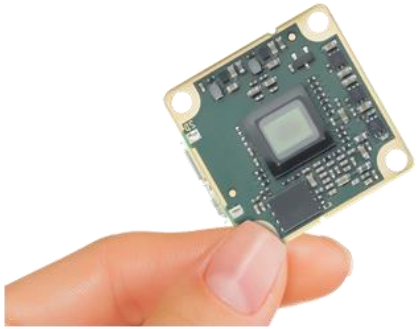

amazon

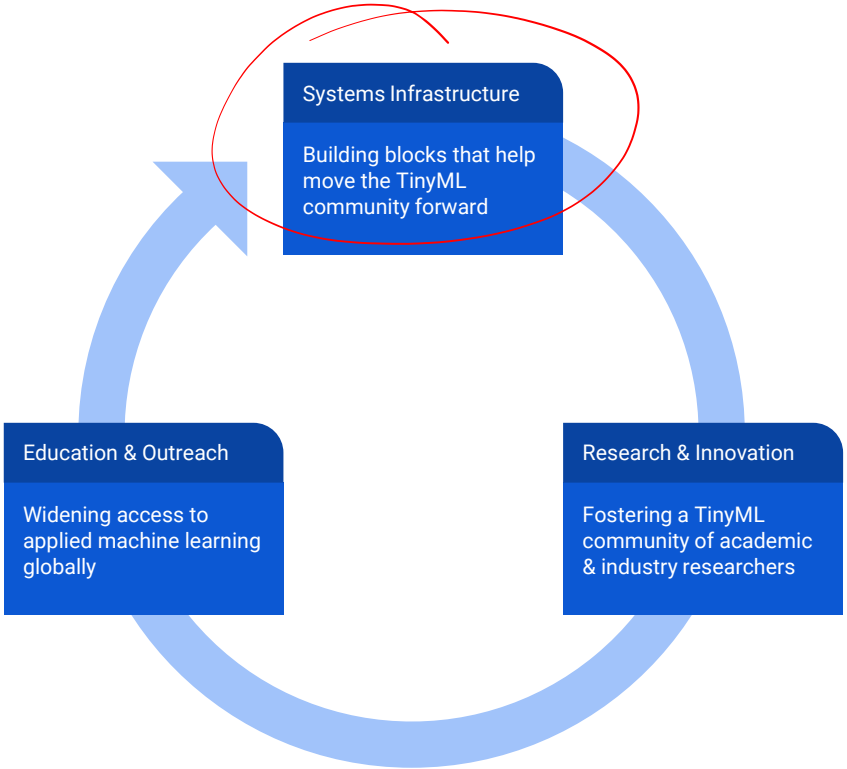

Embedded
Systems
Machine
Learning

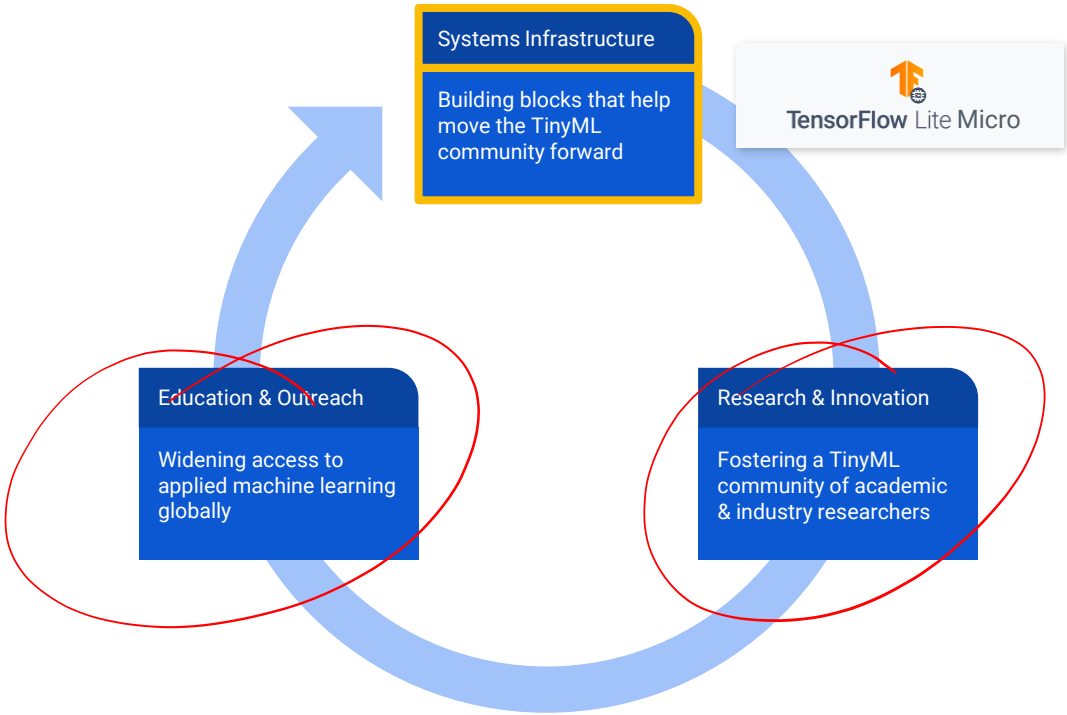
Massive tinyML opportunities in all verticals where machine intelligence meets physical world of billions of sensors



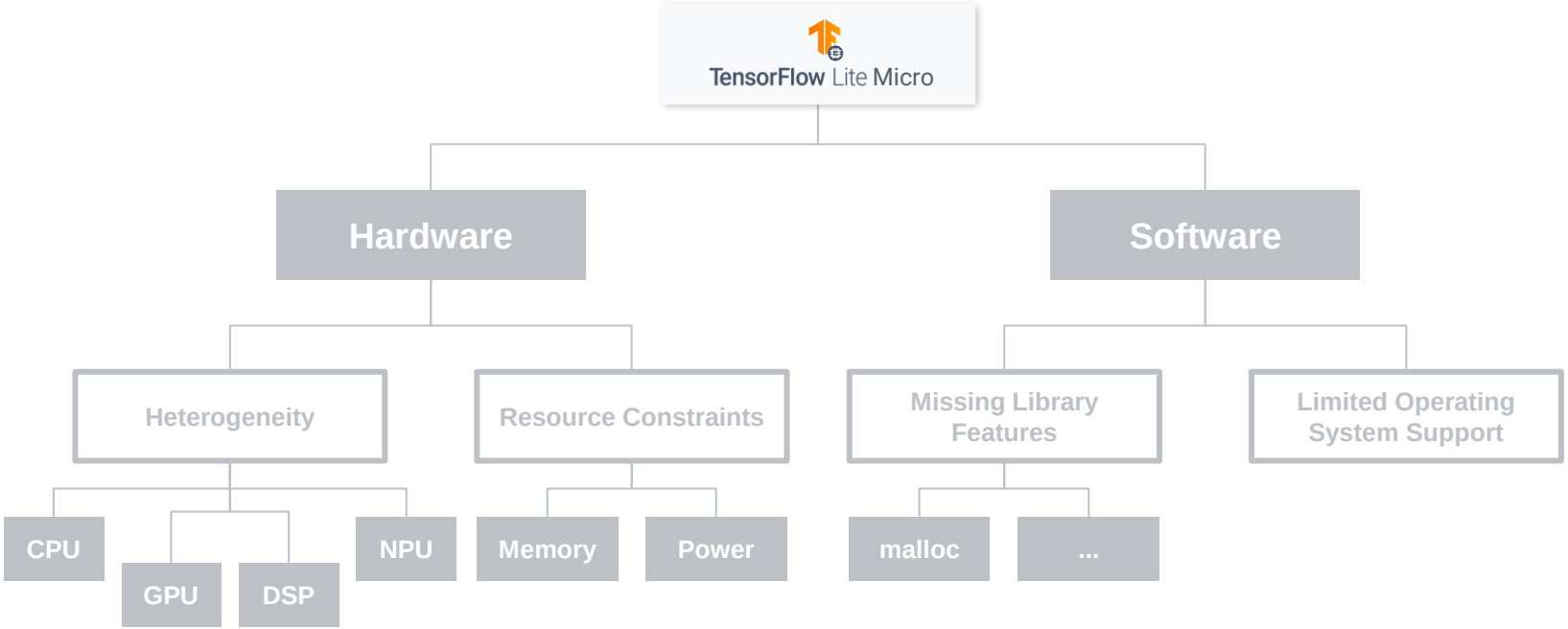
< 1 mW

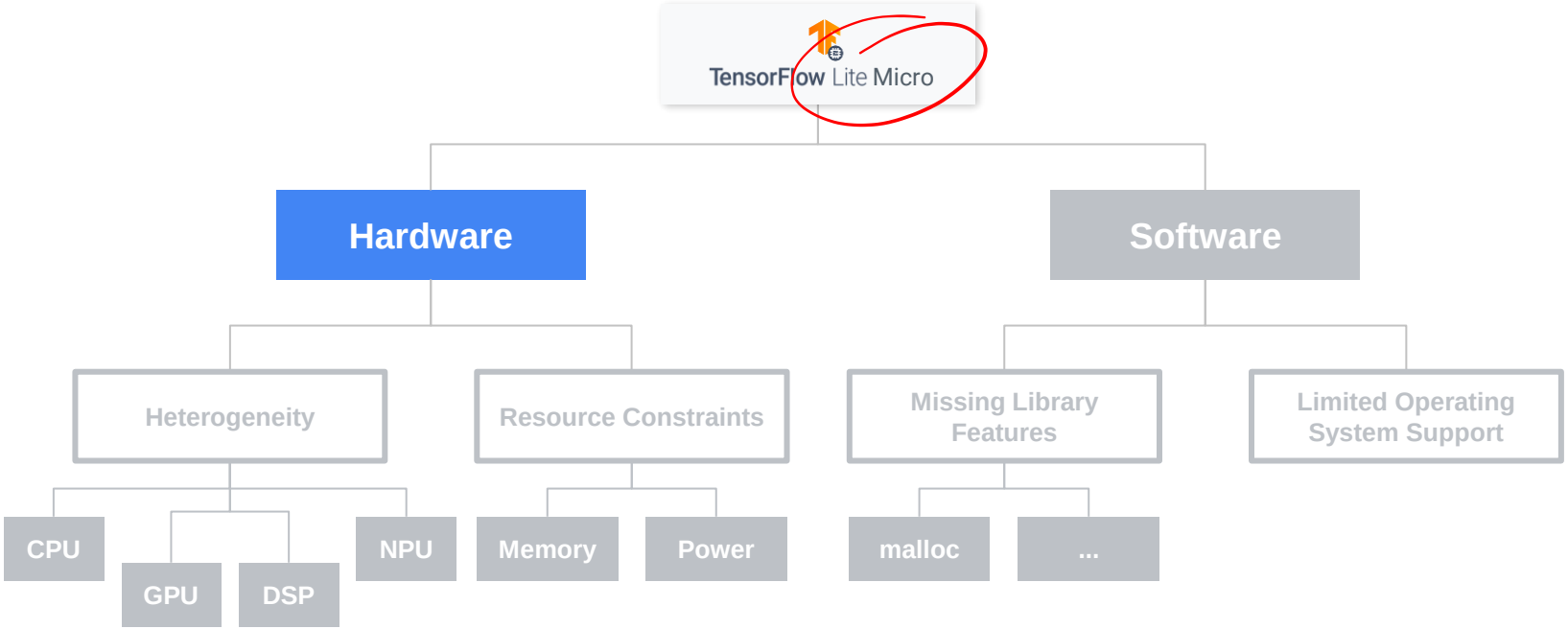


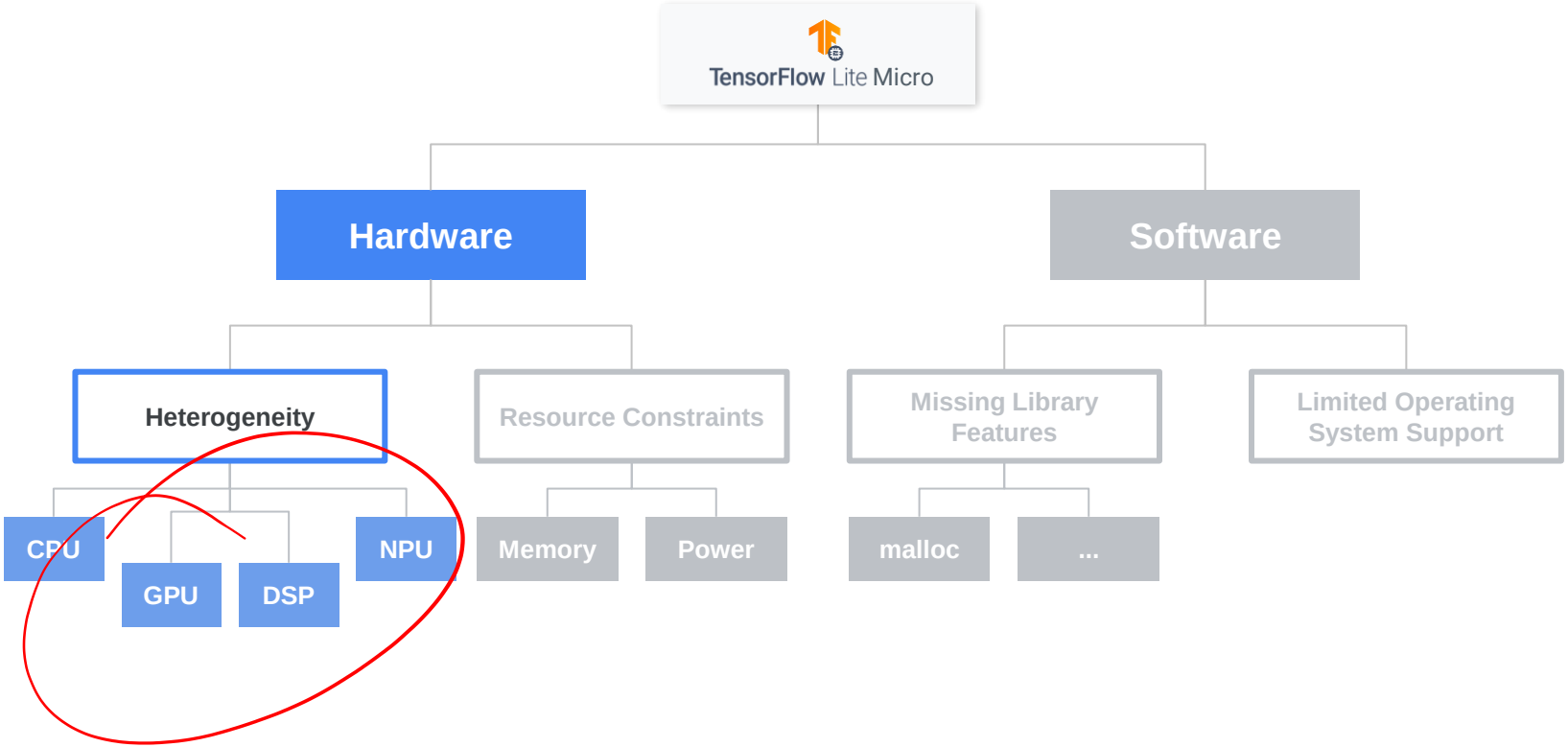

Systems Infrastructure
Building blocks that help move the TinyML community forward
Research & Innovation
Fostering a TinyML community of academic & industry researchers
Education & Outreach
Widening access to applied machine learning globally

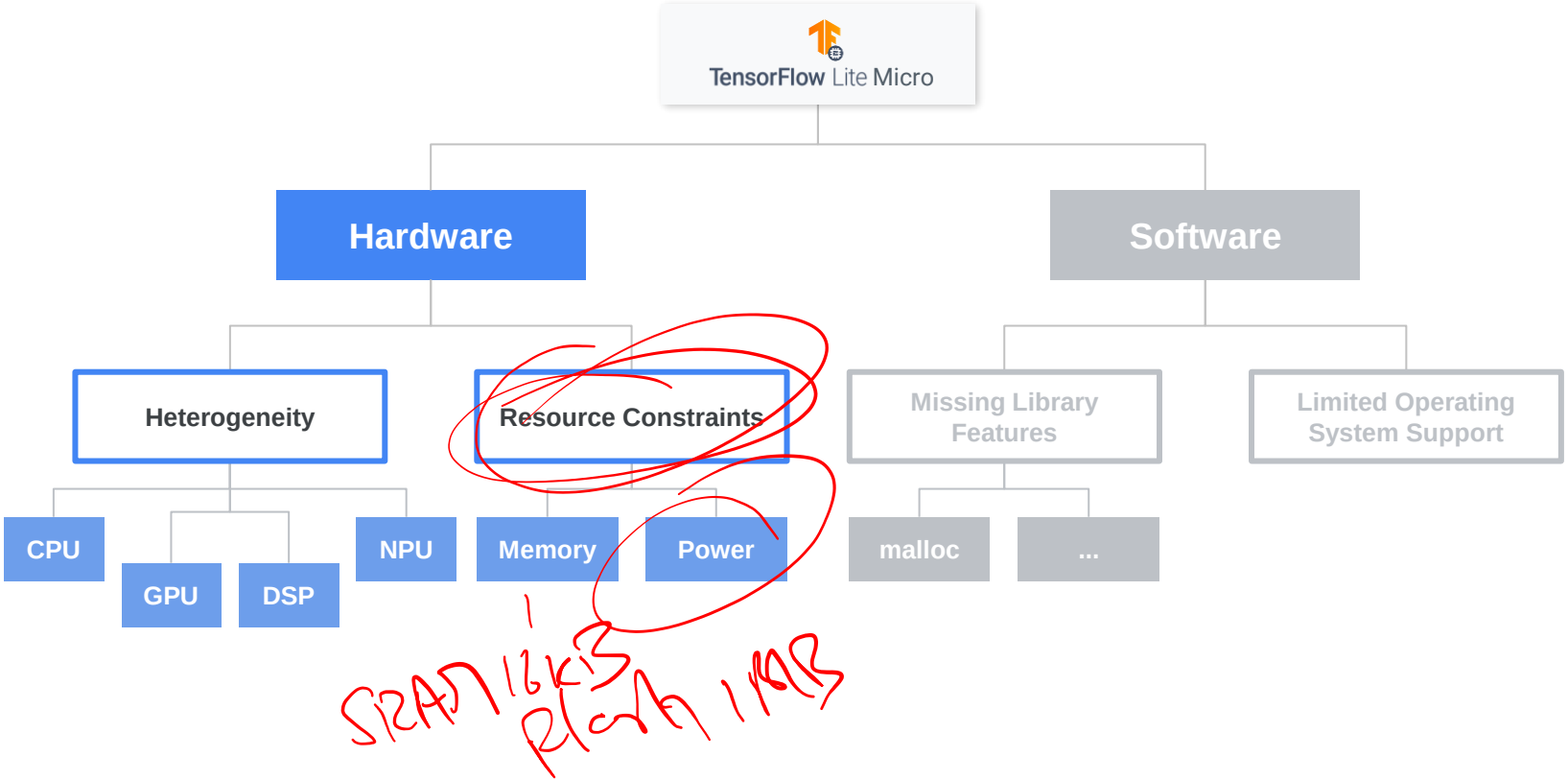

Systems Infrastructure
Building blocks that help move the TinyML community forward
TensorFlow Lite Micro
Education & Outreach
Widening access to applied machine learning globally
Research & Innovation
Fostering a TinyML community of academic & industry researchers

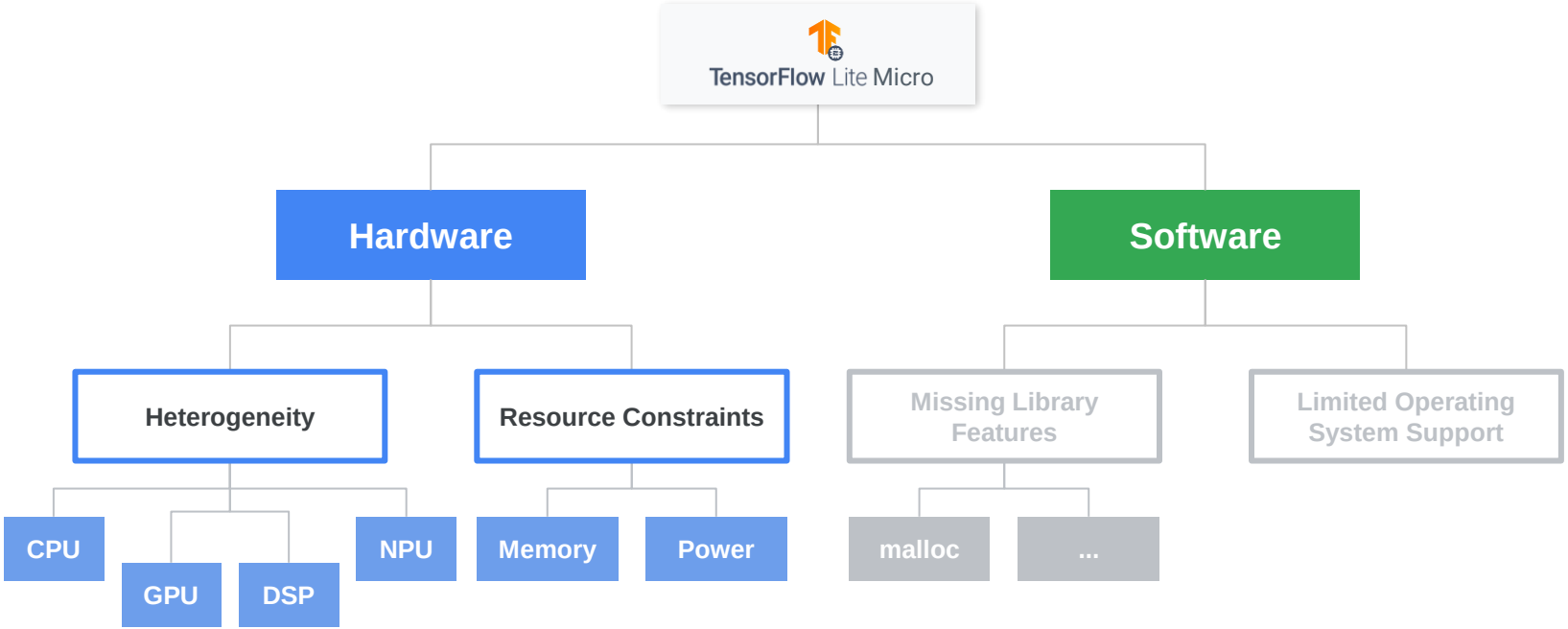

TensorFlow Lite Micro

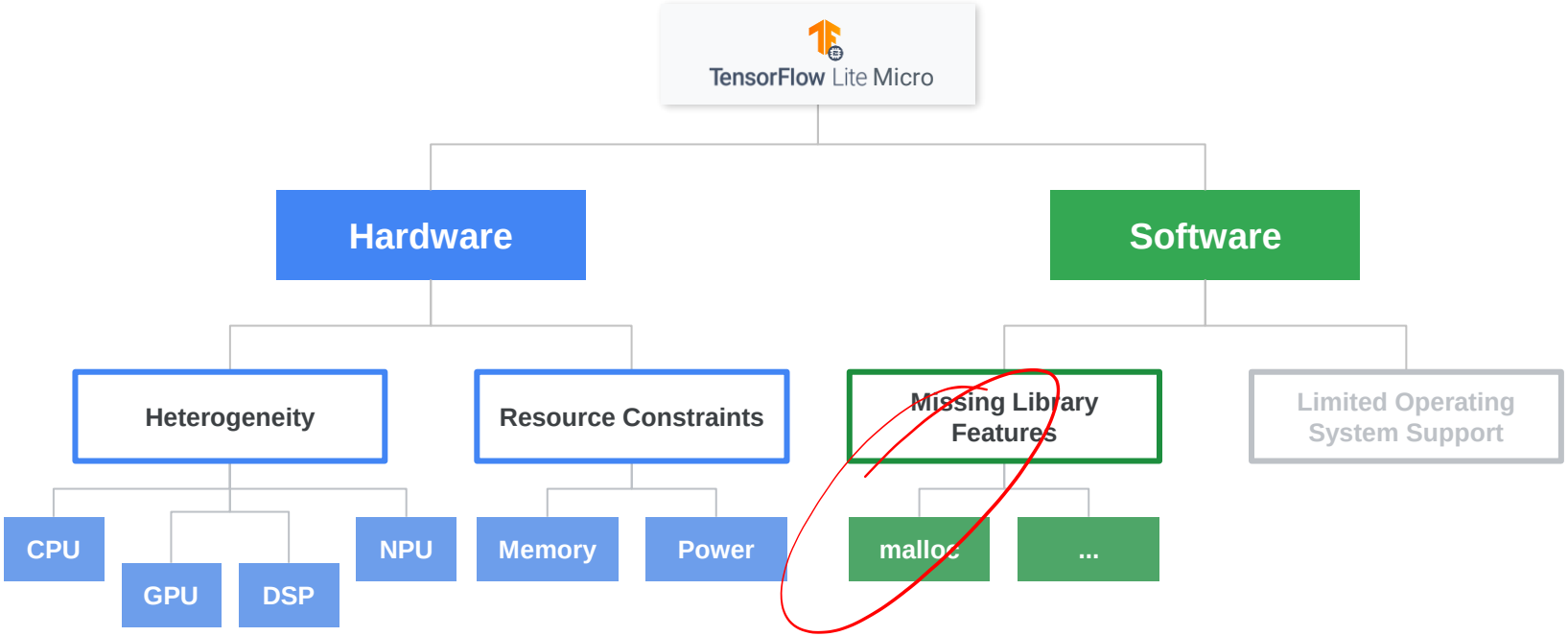

TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...

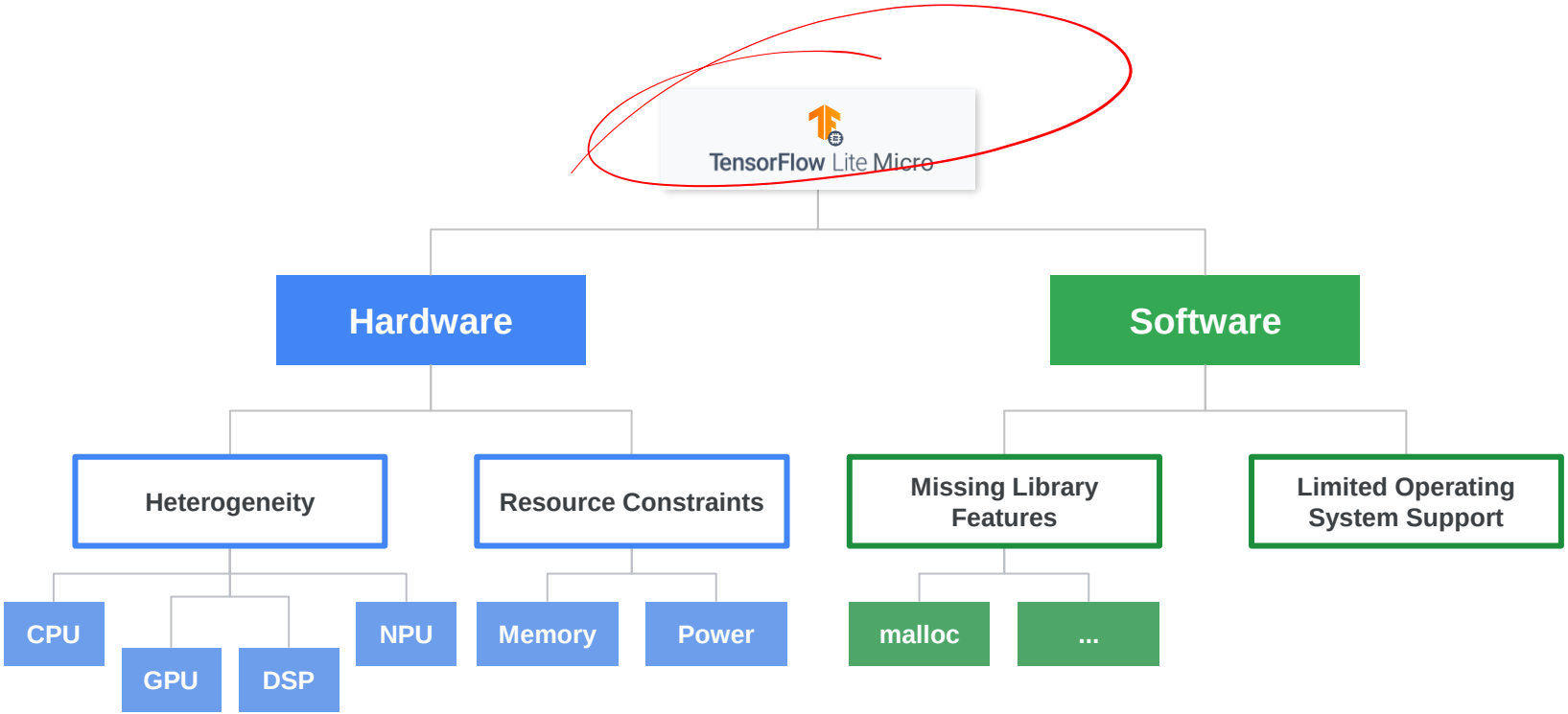

TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...

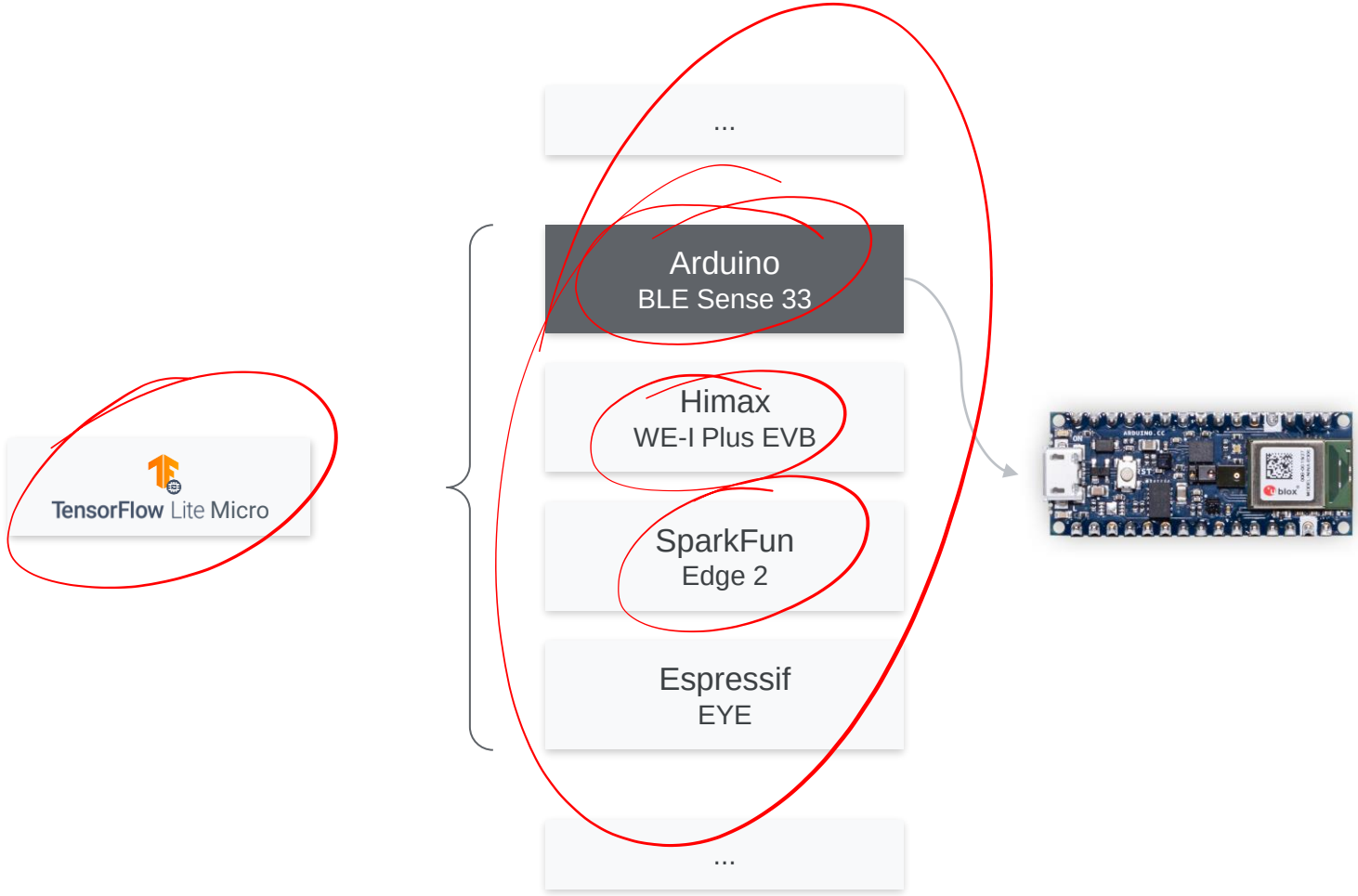

TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...


TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...


TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...


TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...


TensorFlow Lite Micro
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Power
malloc
...


TensorFlow Lite Micro
...
Arduino
BLE Sense 33
Himax
WE-I Plus EVB
SparkFun
Edge 2
Espressif
EYE
...

TensorFlow Lite Micro in a Nutshell

Compatible with the TensorFlow training environment.

Built to fit on **embedded systems**:

- Very **small binary footprint**
- **No** dynamic memory allocation
- **No** dependencies on complex parts of the standard C/C++ libraries
- **No** operating system dependencies, **can run on bare metal**
- Designed to be **portable** across a wide variety of systems

TENSORFLOW LITE MICRO:
EMBEDDED MACHINE LEARNING ON TINYML SYSTEMS

Robert David [1] Jared Duke [1] Advait Jain [1] Vijay Janapa Reddi [1 2]
Nat Jeffries [1] Jian Li [1] Nick Kreeger [1] Ian Nappier [1] Meghna Natraj [1]
Shlomi Regev [1] Rocky Rhodes [1] Tiezhen Wang [1] Pete Warden [1]

ABSTRACT

TensorFlow Lite Micro (TFLM) is an open-source ML inference framework for running deep-learning models on embedded systems. TFLM tackles the efficiency requirements imposed by embedded-system resource constraints and the fragmentation challenges that make cross-platform interoperability nearly impossible. The framework adopts a unique interpreter-based approach that provides flexibility while overcoming these unique challenges. In this paper, we explain the design decisions behind TFLM and describe its implementation. We present an evaluation of TFLM to demonstrate its low resource requirements and minimal run-time performance overheads.

1 INTRODUCTION

Tiny machine learning (TinyML) is a burgeoning field at the intersection of embedded systems and machine learning. The world has over 250 billion microcontrollers (IC Insights, 2020), with strong growth projected over coming years. As such, a new range of embedded applications are emerging for neural networks. Because these models are extremely small (few hundred KBs), running on microcontrollers or DSP-based embedded subsystems, they can operate continuously with minimal impact on device battery life.

The most well-known and widely deployed example of this new TinyML technology is keyword spotting, also called hotword or wakeword detection (Chen et al., 2014; Gruenstein et al., 2017; Zhang et al., 2017). Amazon, Apple, Google, and others use tiny neural networks on billions of devices to run always-on inferences for keyword detection—and this is far from the only TinyML application. Low-latency analysis and modeling of sensor signals from microphones, low-power image sensors, accelerometers, gyros, PPG optical sensors, and other devices enable consumer and industrial applications, including predictive maintenance (Goebel et al., 2020; Susto et al., 2014), acoustic-anomaly detection (Koizumi et al., 2019), visual object detection (Chowdhery et al., 2019), and human-activity recognition (Chavarriaga et al., 2013; Zhang & Sawchuk, 2012).

Unlocking machine learning's potential in embedded de-

vices requires overcoming two crucial challenges. First and foremost, embedded systems have no unified TinyML framework. When engineers have deployed neural networks to such systems, they have built one-off frameworks that require manual optimization for each hardware platform. Such custom frameworks have tended to be narrowly focused, lacking features to support multiple applications and lacking portability across a wide range of hardware. The developer experience has therefore been painful, requiring hand optimization of models to run on a specific device. And altering these models to run on another device necessitated manual porting and repeated optimization effort. An important second-order effect of this situation is that the slow pace and high cost of training and deploying models to embedded hardware prevents developers from easily justifying the investment required to build new features.

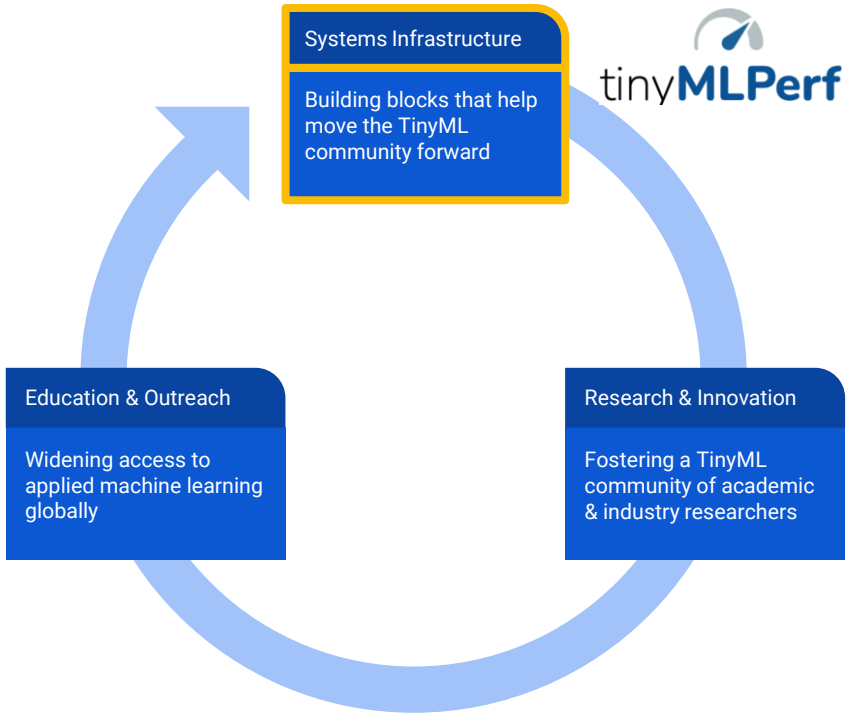
Another challenge limiting TinyML is that hardware vendors have related but separate needs. Without a generic TinyML framework, evaluating hardware performance in a neutral, vendor-agnostic manner has been difficult. Frameworks are tied to specific devices, and it is hard to determine the source of improvements because they can come from hardware, software, or the complete vertically integrated solution.

The lack of a proper framework has been a barrier to accelerating TinyML adoption and application in products. Beyond deploying a model to an embedded target, the framework must also have a means of training a model on a higher-compute platform. TinyML must exploit a broad ecosystem of tools for ML, as well for orchestrating and debugging models, which are beneficial for production devices.

Prior efforts have attempted to bridge this gap. We can distill the major issues facing the frameworks into the following:

[1]Google [2]Harvard University. Correspondence to: Pete Warden <petewarden@google.com>, Vijay Janapa Reddi <vj@eecs.harvard.edu>.

Proceedings of the 4th MLSys Conference, San Jose, CA, USA, 2021.


Systems Infrastructure
Building blocks that help move the TinyML community forward
tinyMLPerf
Research & Innovation
Fostering a TinyML community of academic & industry researchers
Education & Outreach
Widening access to applied machine learning globally

tinyMLPerf

Create a useful benchmark for measuring ML inference performance on ultra low power systems

Comparing ML Systems

GIVEN A TASK AT HAND HOW DO WE KNOW WHICH IS THE **RIGHT SYSTEM**?

HOW DO WE **COMPARE** THE DIFFERENT SOLUTIONS?

Measuring Performance is Hard

- Machine learning system stack is **complicated**
- Many **different** models, datasets, models, frameworks, formats, compilers, libraries, operating systems, targets
- The **cross-product** makes it challenging to decipher performance of any system

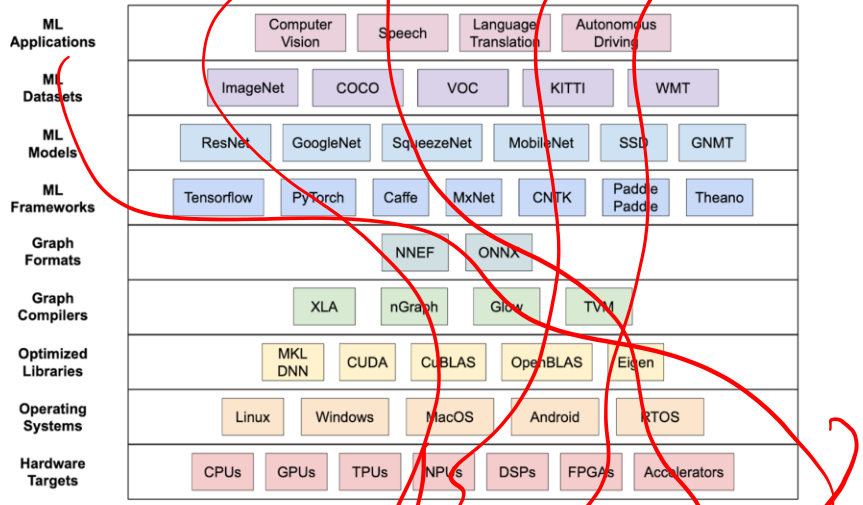
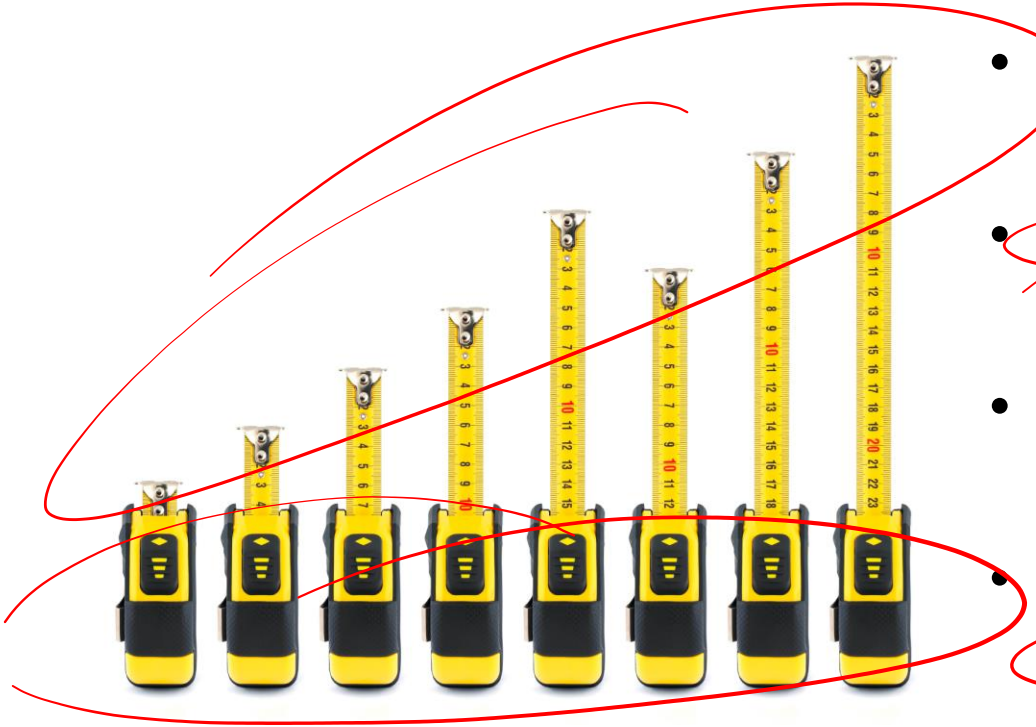


Figure 3. Software and hardware options at every level of the inference stack. The combinations across the layers makes benchmarking ML inference systems a particularly challenging problem.

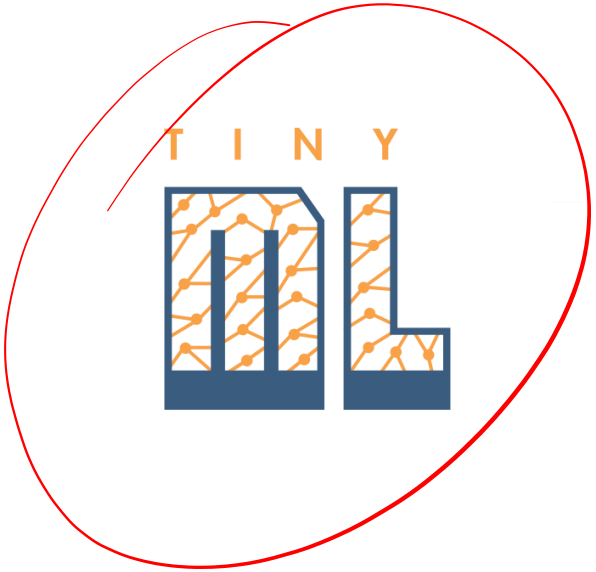
System *X* vs. System *Y*

- What ML task?
- What ML model?
- What ML dataset?
- What batch size?
- What quantization?
- What software libraries?
- …

Need for a tinyMLPerf Benchmark



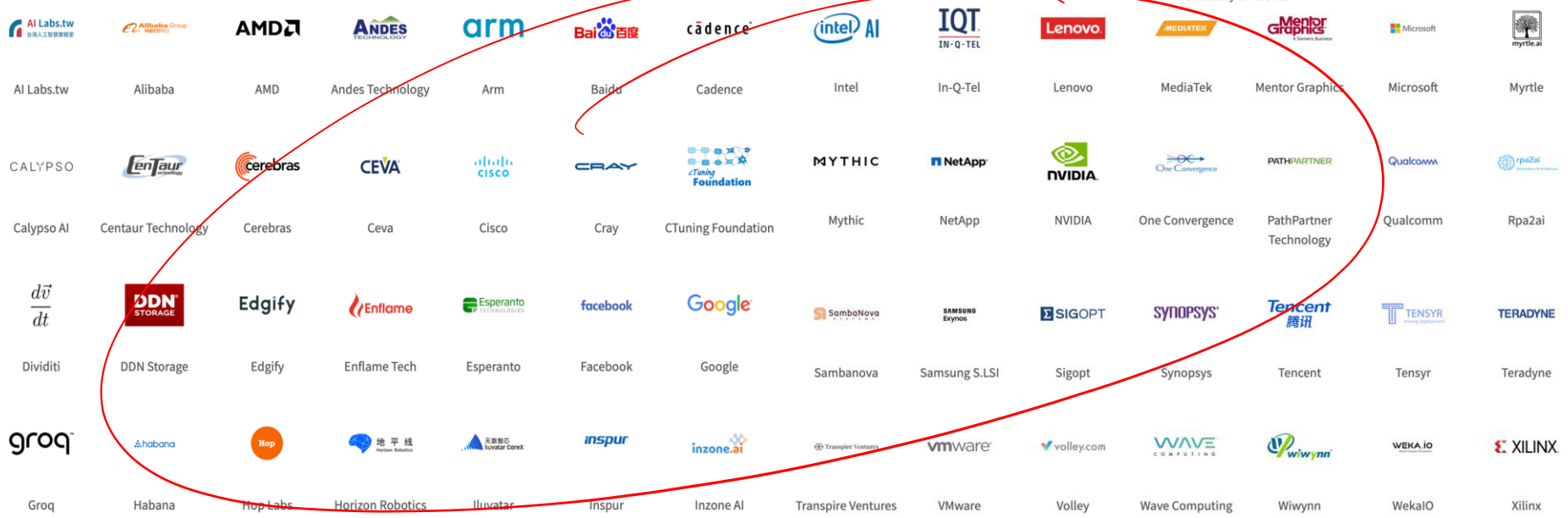
- **Comparability** across hardware and software
- **Standardization** of use cases and workloads
- **Measuring** progress via rigorous methodology
- **Community** building and consensus generation


TINY
ML


MLPerf

What is MLPerf ?

A Community-driven
ML Benchmark Suite



1,000+ members, 50+ organizations, 10+ universities

MLPerf

Goals

Enforce performance result replicability to ensure reliable results

Use **representative workloads**, reflecting production use-cases

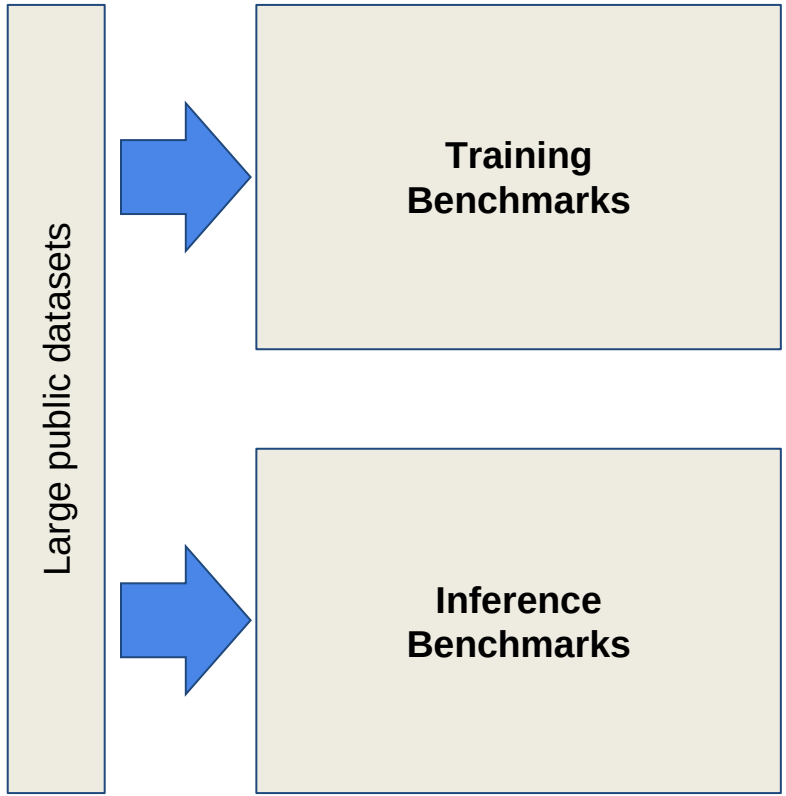
Encourage innovation to improve the state-of-the-art of ML

Accelerate progress in ML via **fair and useful measurement**

Serve both the **commercial and research communities**

Keep **benchmarking affordable** so that all can participate

MLPerf



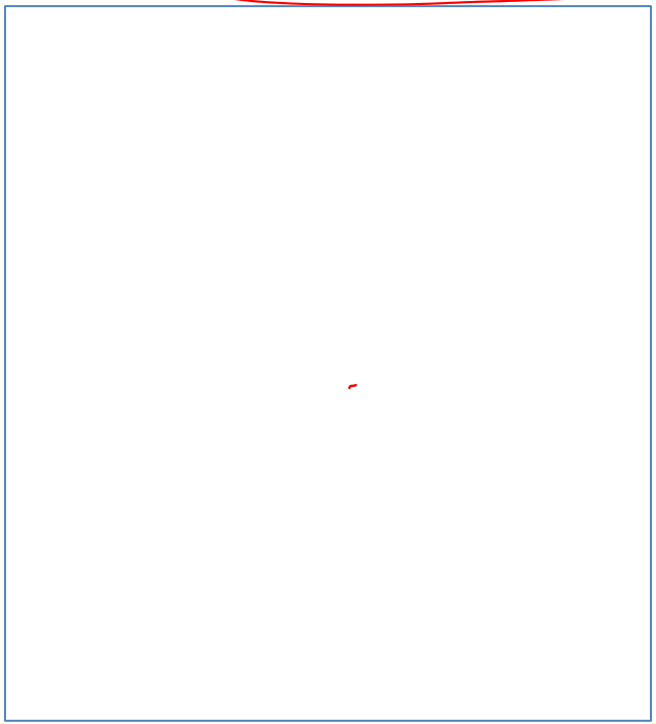
1. Identify a set of **ML tasks** and models
2. Identify real-world **scenarios** to emulate
3. Outline the **rules** for benchmarking
4. Define a clear set of evaluation **metrics**
5. Collect **results** to publish

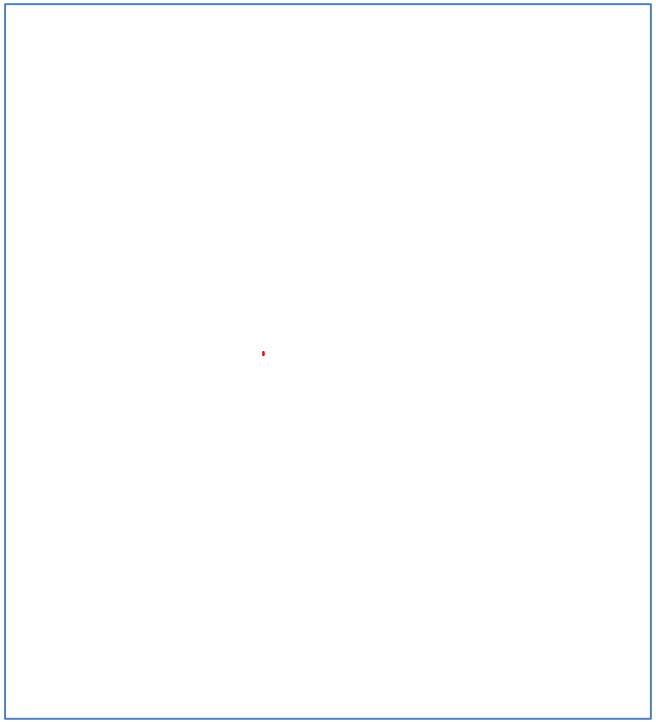
MLPerf Benchmarks

v0.5	v0.7	v1.0	Area	Task	Reference Model	Data Set	Quality Target (Top-1 Accuracy)
✓	✓	✓	Vision	Image classification (heavy)	ResNet-50 v1.5	ImageNet (224x224)	99% of FP32
✓	✓	✓	Vision	Object detection (heavy)	SSD-ResNet34	COCO (1,200x1,200)	99% of FP32
✓	✓	✓	Vision	Object detection (light)	SSD-MobileNet-v1	COCO (300x300)	99% of FP32
✓			Language	Machine translation	GNMT	WMT16 EN-DE	99% of FP32
	✓	✓	Commerce	Recommendation	DLRM	1TB Click Logs	99% of FP32 and 99.9% of FP32
	✓	✓	Language	Language processing	BERT	SQuAD v1.1 (max_seq_len=384)	99% of FP32 and 99.9% of FP32
	✓	✓	Speech	Speech-to-text	RNN-T	Librispeech dev-clean (samples < 15 seconds)	99% of FP32
	✓	✓	Vision	Medical image segmentation	3D U-Net	BraTS 2019 (224x224x160)	99% of FP32 and 99.9% of FP32

MLPerf covers a wide variety of machine learning tasks, models, datasets and prescribes quality targets that have meaningful use to consumers.

Submission & Inference Rules





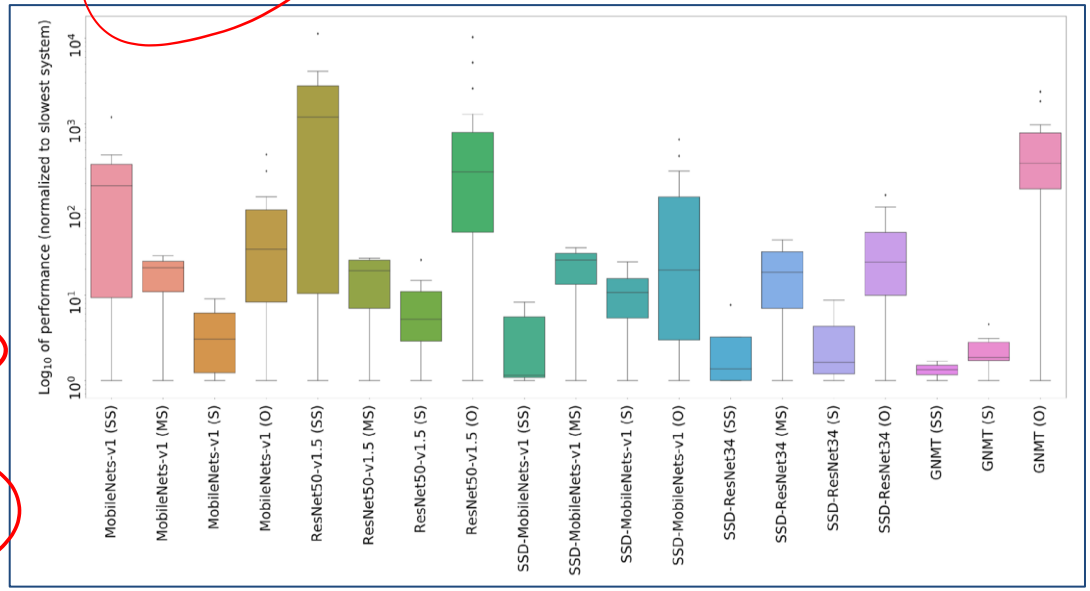
Inference v0.7 Results

Inference Results

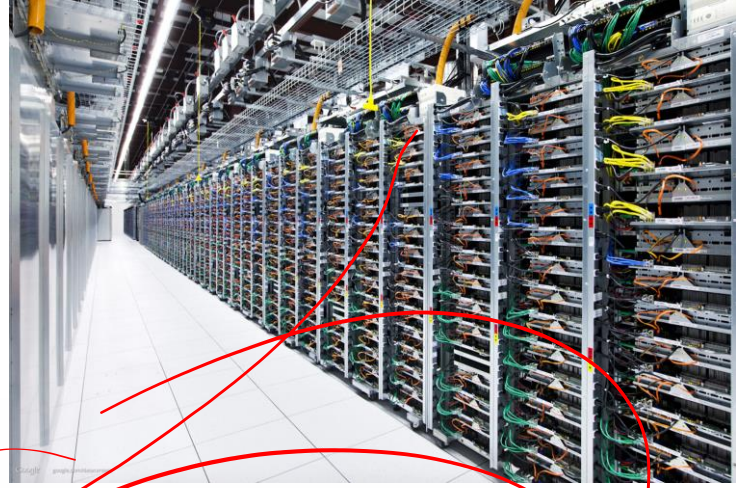
Large repository of over 1800+ reproducible inference results

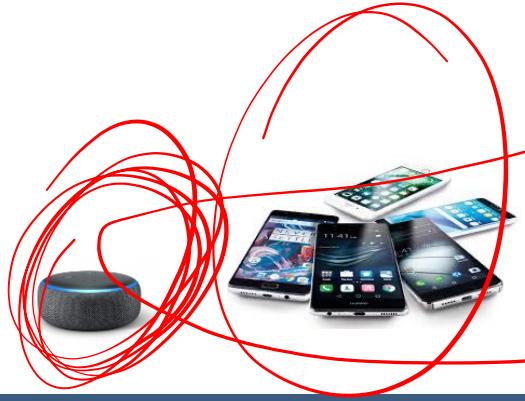
Over 60 different machine learning systems have been submitted

Results capture the vast 10,000x difference in ML system performance

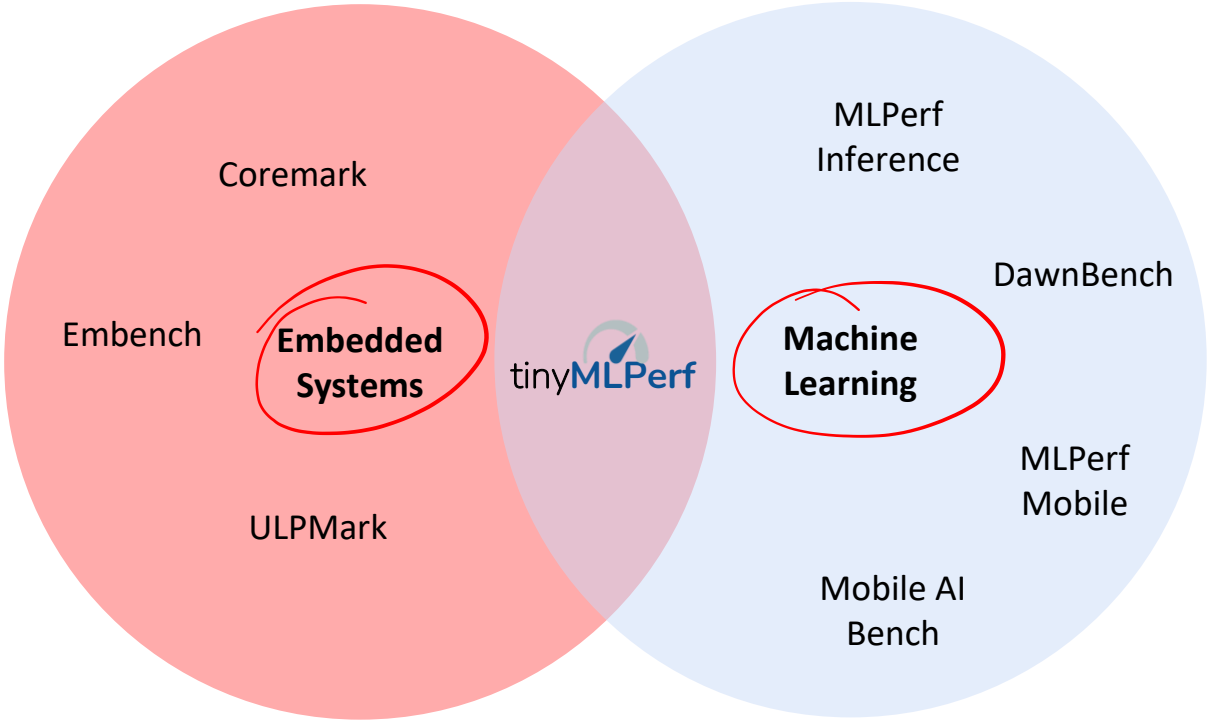


But… the landscape of inference is large and wide





Filling the Void



Create a useful benchmark for measuring ML inference performance on ultra low power systems

ML Benchmark Design Choices

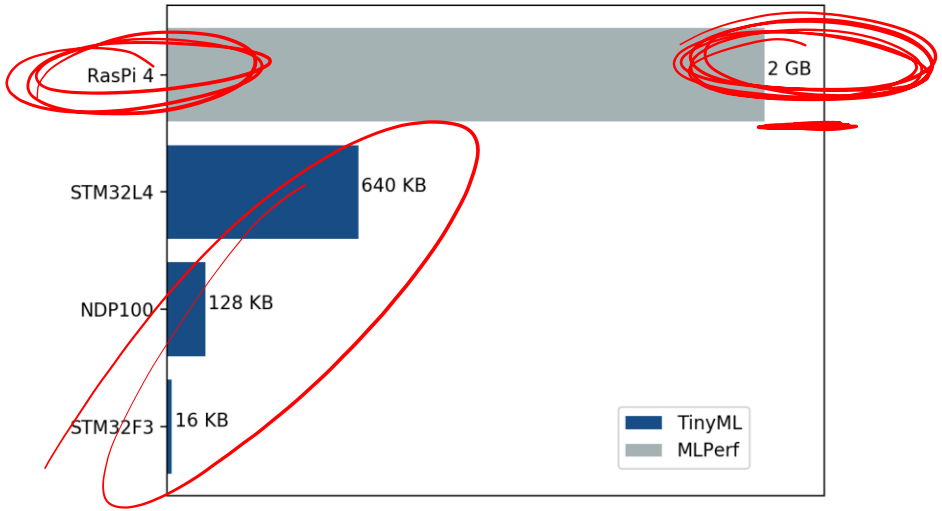
Model Range	Example	Principle
Maturity: Lowest common denominator, most widely used, or most advanced?	**Image recognition:** AlexNet, ResNet, or EfficientNet?	Cutting edge, not bleeding edge
Variety: What broad kind of deep neural network to choose?	**Translation:** GNMT with RNN vs. Transformer with Attention	Try and ensure coverage at a whole suite level
Complexity: Less or more weights?	**Object detection:** SSD vs. Mask R-CNN? Resolution?	Survey and anticipate market demand
Practicality: Availability of datasets?	**Feasibility:** Is there a public dataset?	Good now > perfect.

TinyML Challenges for ML Benchmarking

Task Category	Use Case	Model Type	Datasets
Audio	Audio Wake Words Context Recognition Control Words Keyword Detection	DNN CNN RNN LSTM	Speech Commands Audioset ExtraSensory Freesound DCASE
Image	Visual Wake Words Object Detection Gesture Recognition Object Counting Text Recognition	DNN CNN SVM Decision Tree KNN Linear	Visual Wake Words CIFAR10 MNIST ImageNet DVS128 Gesture
Physiological / Behavioral Metrics	Segmentation Anomaly Detection Forecasting Activity Detection	DNN Decision Tree SVM Linear	Physionet HAR DSA Opportunity
Industry Telemetry	Sensing Predictive Maintenance Motor Control	DNN Decision Tree SVM Linear Naive Bayes	UCI Air Quality UCI Gas UCI EMG NASA's PCoE

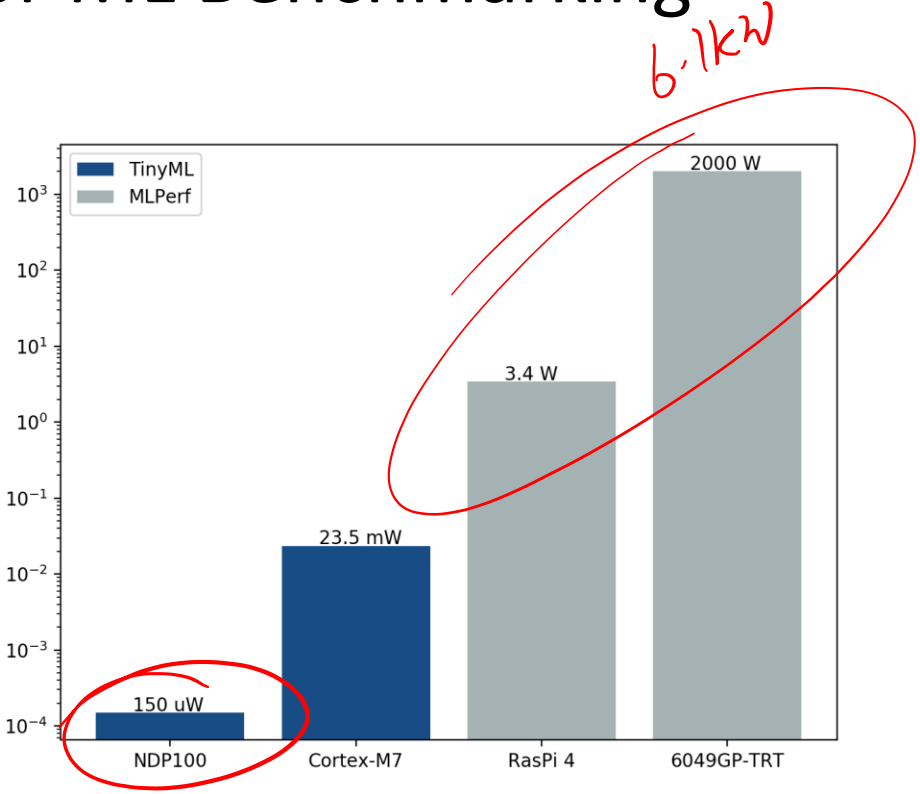
TinyML Challenges for ML Benchmarking

- Resources are extremely condensed in the tinyML devices
- Need to define a methodology to load the SUT for evaluation
- **How can we come up with a methodology that works across many different systems?**



TinyML Challenges for ML Benchmarking

- Power is optional in MLPerf
- MLPerf power working group is trying to develop a specification
- But power is a first-order design constraint in TinyML devices
- **How to define a power spec?**



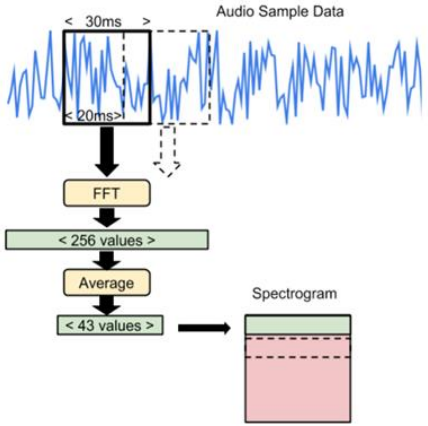
tinyMLPerf Benchmark Design Choices

Big Questions	Inference
1. Benchmark definition	What is the definition of a benchmark task?
2. Benchmark selection	Which benchmark task to select?
3. Metric definition	What is the measure of “performance” in ML systems?
4. Implementation equivalence	How do submitters run on different hardware/software systems?
5. Issues specific to training or inference	Quantization, calibration, and/or retraining?
	Reduce result variance?
6. Results	Do we normalize and/or summarize results?


Driven by Community Consensus
LG
CATAPULT Digital
cisco
OctoML
RENESAS
HARVARD
John A. Paulson School of Engineering and Applied Sciences
Google
intel
Southeast Missouri State University
life.augmented
arm
ZOOX
GREENWAVES TECHNOLOGIES
UNIVERSITY of WASHINGTON
ASU Arizona State University
Microsoft
ambiq micro
AONdevices
AI | DSP | ASIC
KU LEUVEN
SAMSUNG
CIRRUS LOGIC
MEDIATEK
Reality AI
Explainable AI at the Edge
Red Hat
SYNTIANT
SiMa.ai
Cogneefy
NXP
PICOVOICE
cadence

Four Benchmarks

Keyword Spotting



Warden, Pete. "Speech commands: A dataset for limited-vocabulary speech recognition." *arXiv preprint arXiv:1804.03209* (2018).

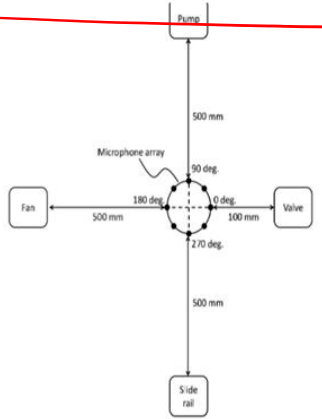
Visual Wake Words

(a) 'Person'

(b) 'Not-person'

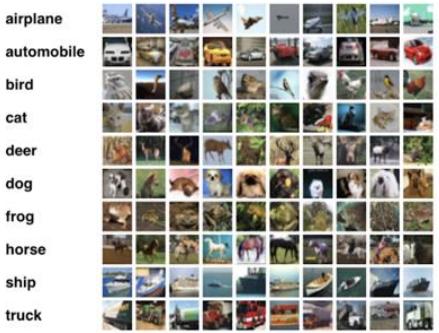
Chowdhery, Aakanksha, et al. "Visual wake words dataset." *arXiv preprint arXiv:1906.05721* (2019).

Anomaly Detection



Purohit, Harsh, et al. "MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection." *arXiv preprint arXiv:1909.09347* (2019).

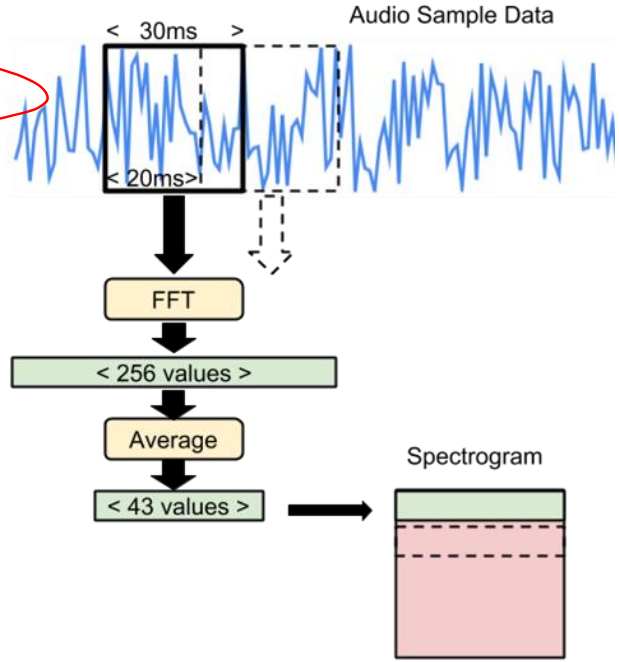
Tiny Image Classification



Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.

Keyword Spotting (KWS)

- Task:
  - Limited vocabulary speech recognition
- Dataset:
  - Google Speech Commands
- Model:
  - DS-CNN
- Benchmark Owner:
  - Syntiant



Warden, Pete. "Speech commands: A dataset for limited-vocabulary speech recognition." *arXiv preprint arXiv:1804.03209* (2018).

Visual Wake Words (VWW)

- Task:
  - Two class image classification
- Dataset:
  - Visual Wake Words Dataset
- Model:
  - MobileNetV1 0.25X
- Benchmark Owner:
  - Google

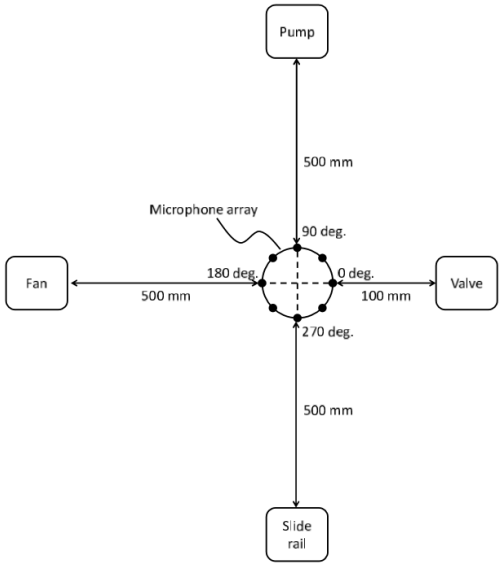
(a) ‘Person’

(b) ‘Not-person’

Chowdhery, Aakanksha, et al. "Visual wake words dataset." *arXiv preprint arXiv:1906.05721* (2019).

Anomaly Detection (AD)

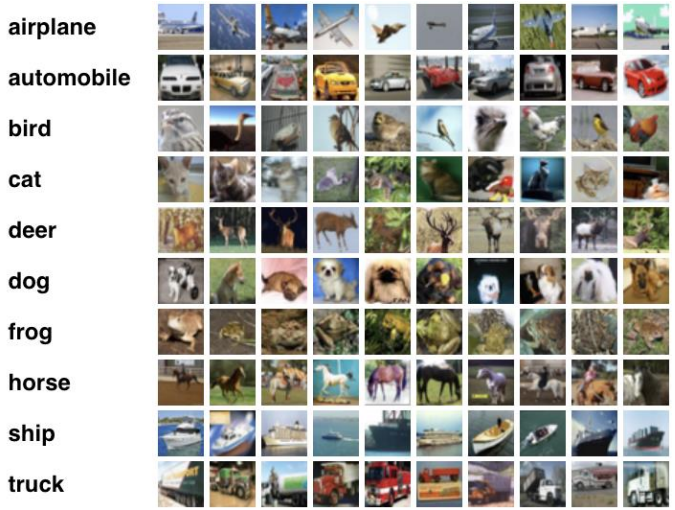
- Task:
  - Audio time series anomaly detection
- Dataset:
  - ToyADMOS
- Model:
  - Deep autoencoder
- Benchmark Owner:
  - Digital Catapult



Purohit, Harsh, et al. "MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection." *arXiv preprint arXiv:1909.09347* (2019).

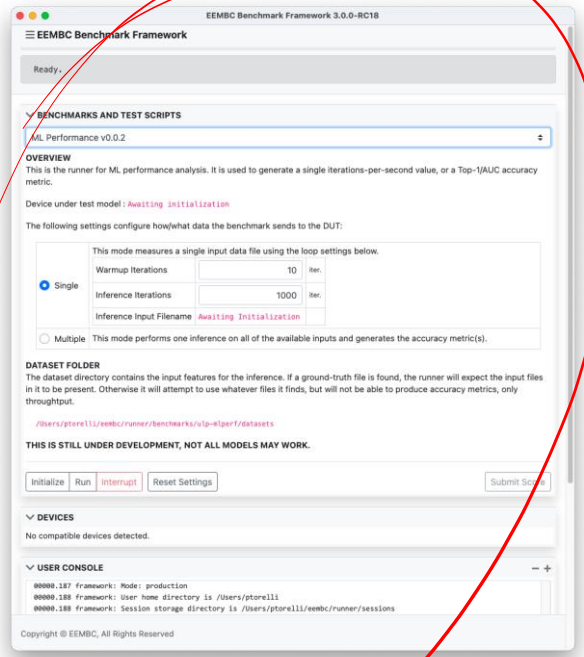
Image Classification (IC)

- Task:
  - Multiclass image classification with small images
- Dataset:
  - CIFAR10
- Model:
  - ResNet8
- Benchmark Owner:
  - SiliconLabs

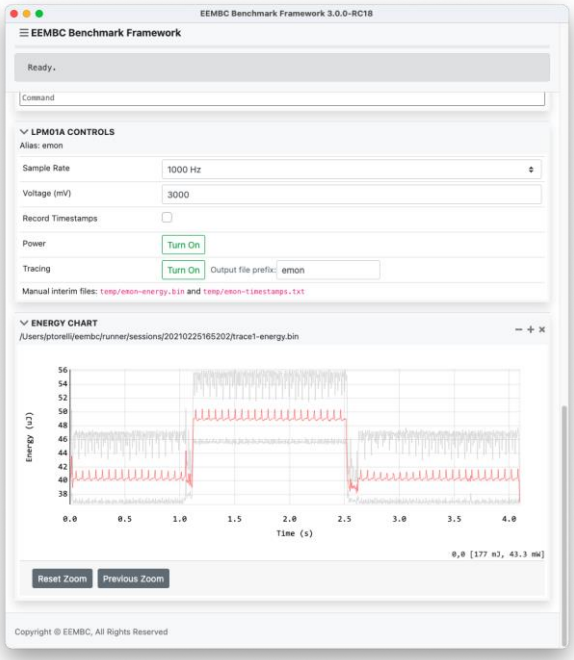


Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.

tinyMLPerf Benchmark Runner



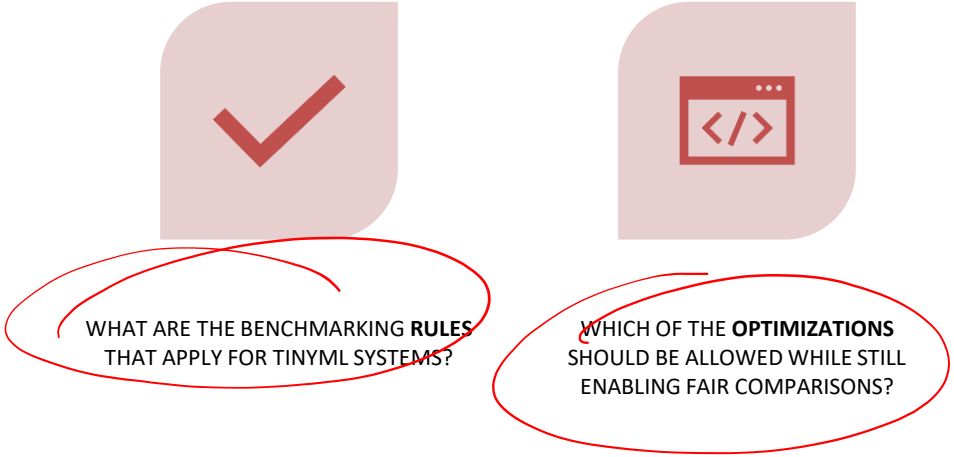
Performance



Power

TinyML Challenges for ML Benchmarking

- Plethora of techniques
  - Quantization
  - Sparsity
  - Pruning
  - Retraining
  - …



Next Steps

Version 0.1 Submissions:
April 30th 2021

Work with member organizations
to submit inference results

Validate submission

Publish results

Version 0.2
Adapt, Adjust, Improve

Adapt the benchmark
based on the feedback from v0.1

Adjust the specifications and requirements

Improve the benchmark

TinyMLPerf

BENCHMARKING TINYML SYSTEMS: CHALLENGES AND DIRECTION

Colby R. Banbury [1] Vijay Janapa Reddi [1] Max Lam [1] William Fu [1] Amin Fazel [2] Jeremy Holleman [3 4] Xinyuan Huang [3] Robert Hurtado [5] David Kanter [7] Anton Lokhmotov [8] David Patterson [9 10] Danilo Pau [11] Jae-sun Seo [12] Jeff Sieracki [13] Urmish Thakker [14] Marian Verhelst [15 16] Poonam Yadav [17]

ABSTRACT

Recent advancements in ultra-low-power machine learning (TinyML) hardware promises to unlock an entirely new class of smart applications. However, continued progress is limited by the lack of a widely accepted benchmark for these systems. Benchmarking allows us to measure and thereby systematically compare, evaluate, and improve the performance of systems. In this position paper, we present the current landscape of TinyML and discuss the challenges and direction towards developing a fair and useful hardware benchmark for TinyML workloads. Our viewpoints reflect the collective thoughts of the TinyMLPerf working group that is comprised of 30 organizations.

1 INTRODUCTION

Machine learning (ML) inference on the edge is an increasingly attractive prospect due to its potential for increasing energy efficiency (Fedorov et al., 2019), privacy, responsiveness (Zhang et al., 2017), and autonomy of edge devices. Thus far, the field edge ML has predominately focused on mobile inference which has led to numerous advancements in machine learning models such as exploiting pruning, sparsity, and quantization. But in recent years, there have major been strides in expanding the scope of edge systems. Interest is brewing in both academia (Fedorov et al., 2019; Zhang et al., 2017) and industry (Flamand et al., 2018; Warden, 2018a) towards expanding the scope of edge ML to microcontroller-class devices.

The goal of "TinyML" (tinyML Foundation, 2019) is to bring ML inference to ultra-low-power devices, typically under a milliWatt, and thereby break the traditional power barrier preventing widely distributed machine intelligence. By performing inference on-device, and near-sensor, TinyML enables greater responsiveness and privacy while avoiding the energy cost associated with wireless communication, which at this scale is far higher than that of compute (Warden, 2018b). Furthermore, the efficiency of TinyML enables a class of smart, battery-powered, always-on applications that can revolutionize the real-time collection and processing of data. This emerging field, which is the culmination of many innovations, is poised only further to accelerate its growth in the coming years.

To unlock the full potential of the field, hardware software co-design is required. Specifically, TinyML models must be small enough to fit within the tight constraints of MCU-class devices (e.g., a few hundred kB of memory and limited onboard compute horsepower in the order of MHz processor clock speed), thus limiting the size of the input and the number of layers (Zhang et al., 2017) or necessitating the use lightweight, non-neural network-based techniques (Kumar et al., 2017). TinyML tools are broadly defined as anything that enables the design, mapping, and deployment of TinyML algorithms including aggressive quantization techniques (Wang et al., 2019), memory aware neural architecture searches (Fedorov et al., 2019), frameworks (TensorFlow), and efficient inference libraries (Lai et al., 2018; Garofalo et al., 2019). Efforts in TinyML hardware include improving inference on the next generation of general-purpose MCUs (arm; Flamand et al., 2018), developing hardware specialized for low power inference, and creating novel architectures intended only as inference engines for specific tasks (Moons et al., 2018).

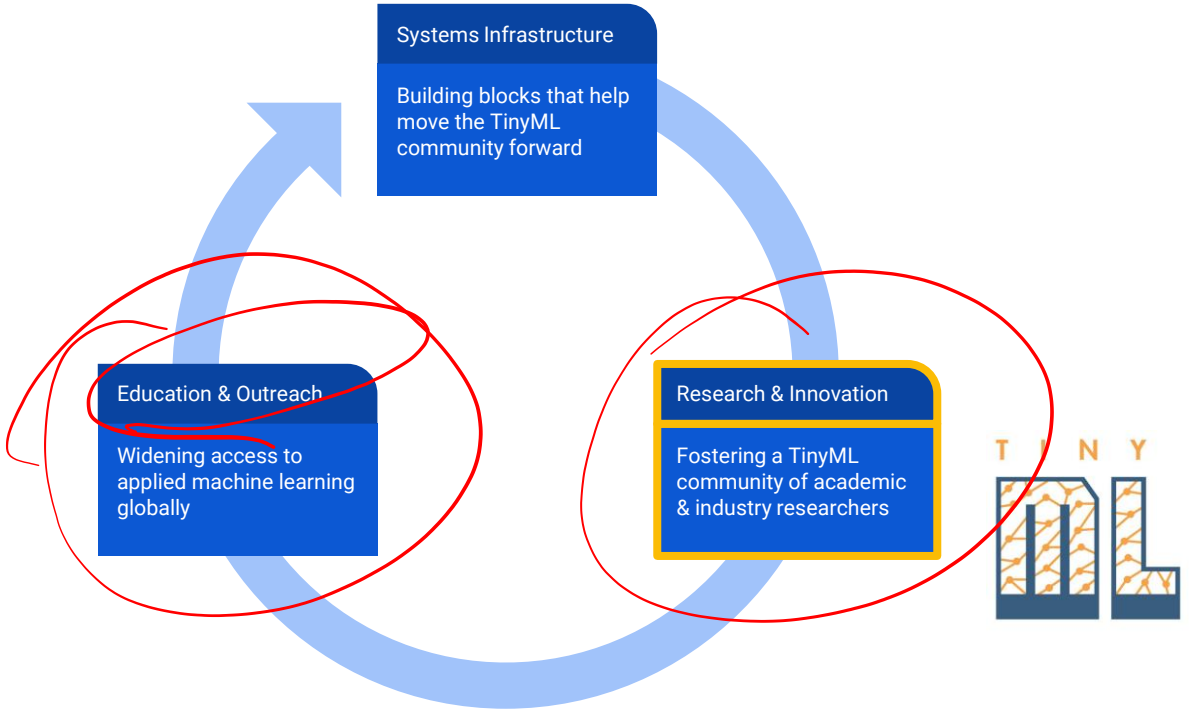
The complexity and dynamicity of the field obscure the measurement of progress and make dynamism design decisions intractable. In order to enable the continued innovation, a fair and reliable method of comparison is needed. Since progress is often the result of increased hardware capability, a reliable TinyML hardware benchmark is required.

In this paper, we discuss the challenges and opportunities associated with the development of a TinyML hardware

[1]Harvard University [2]Samsung Semiconductor, Inc. [3]Syntiant [4]University of North Carolina, Charlotte [7]Cisco Systems [8]California State Polytechnic University, Pomona [5]Real World Insights [9]dividiti [9]University of California, Berkeley [10]Google [11]STMicroelectronics, Italy [12]Arizona State University [13]Reality AI [14]Arm ML Research Lab [15]KU Leuven [16]Interuniversity Microelectronics Centre (IMEC) [17]University of York. Correspondence to: Colby R. Banbury <cbanbury@g.harvard.edu>.

Proceedings of the 3rd MLSys Conference, Austin, TX, USA, 2020.

- **The field of tinyML is growing and needs comparability**
- **Community collaboration is key**
  - Defines Tasks, Scenarios, Datasets, Methods
  - Establish clear set of metrics and divisions
- **Please read the whitepaper for additional details**


Systems Infrastructure
Building blocks that help move the TinyML community forward
Education & Outreach
Widening access to applied machine learning globally
Research & Innovation
Fostering a TinyML community of academic & industry researchers
TINY ML

tinyML Summit 2021

Enabling ultra-low Power Machine Learning at the Edge

First tinyML Research Symposium

The first annual **tinyML research symposium** serves as a **flagship venue for research** at the intersection of machine learning applications, algorithms, software, and hardware in deeply embedded systems. We solicit papers from **academia and industry** combining **cross-layer innovations** across a wide range of topics. Submissions must describe tinyML innovations that **intersect and leverage synergy** between at least two of the following subject areas…

Research Topics

- **Datasets**
  - Public release of new datasets to tinyML
  - Frameworks that automate dataset development
  - Survey and analysis of existing tiny datasets that can be used for research
- **Applications**
  - Novel applications across all fields and emerging use cases
  - Discussions about real-world use cases
  - User behavior and system-user interaction
  - Survey on practical experiences
- **Algorithms**
  - Federated learning or stream-based active learning methods
  - Deep learning and traditional machine learning algorithms
  - Pruning, quantization, optimization methods
  - Security and privacy implications
- **Systems**
  - Profiling tools for measuring and characterizing system performance and power
  - Solutions that involve hardware and software co-design
  - Characterization of tiny real-world embedded systems
  - In-sensor processing, design, and implementation
- **Software**
  - Interpreters and code generator frameworks for tiny systems
  - Optimizations for efficient execution
  - Software memory optimizations
  - Neural architecture search methods
- **Hardware**
  - Power management, reliability, security, performance
  - Circuit and architecture design
  - Ultra-low-power memory system design
  - MCU and accelerator architecture design and evaluation
- **Evaluation**
  - Measurement tools and techniques
  - Benchmark creation, assessment and validation
  - Evaluation and measurement of real production systems

Program Committee

Boris MURMANN
Program Chair
Stanford University

Vijay JANAPA REDDI
Program Chair
Harvard University

Zain ASGAR
Publicity Chair
Stanford University

Edith BEIGNÉ
Facebook

H. T. KUNG
Harvard University

Matthew MATTINA
Arm

Tinoosh MOHSENIN
University of Maryland Baltimore County

Edwin PARK
QUALCOMM Inc

Vikas CHANDRA
Facebook Reality Labs

Yiran CHEN
Duke University

Hiroshi DOYU
Ericsson

Adam FUKS
NXP

Priyanka RAINA
Stanford University

Jae-sun SEO
Arizona State University

Mingoo SEOK
Columbia University

Dennis SYLVESTER
University of Michigan

Hoi-Jun YOO
KAIST

Wolfgang FURTNER
Infineon Technologies

Song HAN
MIT EECS

Prateek JAIN
Microsoft

Kurt KEUTZER
University of California, Berkeley

Jonathan TAPSON
GrAI Matter Labs

Theocharis THEOCHARIDES
Publicity Chair

Marian VERHELST
KU Leuven

Pete WARDEN
Google

TINY ML

tinyML 2021

First International Research Symposium on Tiny Machine Learning (tinyML)

Burlingame, CA Mar 22 2021 https://tinyml.org/home/index.html vj@eecs.harvard.edu

About tinyML Research Symposium

Tiny machine learning (tinyML) is a fast-growing field of machine learning technologies and applications including algorithms, hardware, and software capable of performing on-device sensor (vision, audio, IMU, biomedical, etc.) data analytics at extremely low power, typically in the mW range and below, and hence enabling a variety of always-on use-cases and targeting battery-operated devices. tinyML systems are becoming "good enough" for (i) many commercial applications and new systems on the horizon; (ii) significant progress is being made on algorithms, networks, and models down to 100 kB and below; and (iii) initial low power applications in vision and audio are becoming mainstream and commercially available. There is growing momentum demonstrated by technical progress and ecosystem development.

To nurture innovation and growth in this emerging area within machine learning, the first annual tinyML research symposium serves as a flagship venue for research at the intersection of machine learning applications, algorithms, software, and hardware in deeply embedded machine learning systems. Accepted papers will be published in the form of peer-reviewed online proceedings. An author of an accepted paper will have the opportunity to attend the research symposium to give an oral presentation.

Program Chairs

- Vijay Janapa Reddi, Harvard University
- Boris Murmann, Stanford University

Program Committee

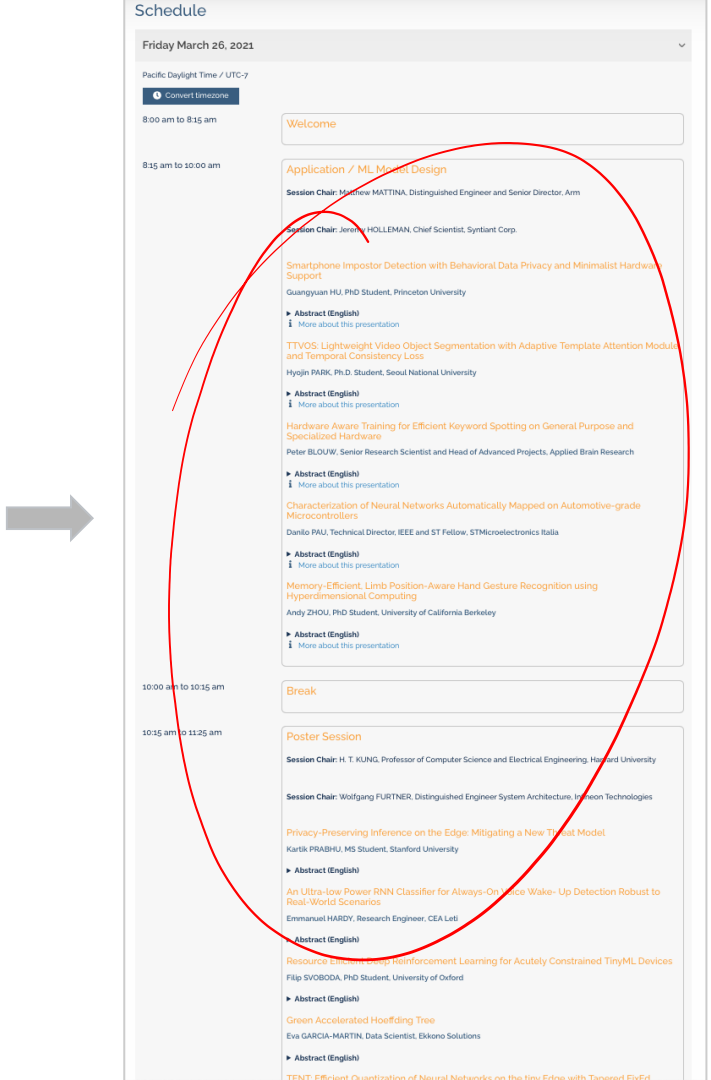
- Edith Beigne, Facebook
- Vikas Chandra, Facebook
- Yiran Chen, Duke Univ.
- Hiroshi Doyu, Ericsson
- Adam Fuks, NXP
- Wolfgang Furtner, Infineon
- Song Han, MIT
- Jeremy Holleman, Syntiant
- Prateek Jain, Microsoft
- Kurt Keutzer, Berkeley
- H. T. Kung, Harvard
- Matthew Mattina, ARM
- Tinoosh Mohsenin, Univ. of Maryland
- Edwin Park, Qualcomm
- Priyanka Raina, Stanford Univ.
- Jae-sun Seo, ASU
- Mingoo Seok, Columbia Univ.
- Dennis Sylvester, Univ. of Michigan
- Jonathan Tapson, GrAI Matter Labs
- Marian Verhelst, KU Leuven
- Pete Warden, Google
- Hoi-Jun Yoo, KAIST

Call for Papers

We solicit papers from academia and industry combining cross-layer innovations across topics. Submissions must describe tinyML innovations that intersect and leverage synergy between at least two of the following subject areas:

== tinyML Datasets ==

- Public release of new datasets to tinyML
- Frameworks that automate dataset development



Schedule

Friday March 26, 2021

Pacific Daylight Time / UTC-7

Convert timezone

8:00 am to 8:15 am — Welcome

8:15 am to 10:00 am — Application / ML Model Design

Session Chair: Matthew MATTINA, Distinguished Engineer and Senior Director, Arm

Session Chair: Jeremy HOLLEMAN, Chief Scientist, Syntiant Corp.

Smartphone Impostor Detection with Behavioral Data Privacy and Minimalist Hardware Support

Guangyuan HU, PhD Student, Princeton University

▸ Abstract (English)

More about this presentation

TTVOS: Lightweight Video Object Segmentation with Adaptive Template Attention Module and Temporal Consistency Loss

Hyojin PARK, Ph.D. Student, Seoul National University

▸ Abstract (English)

More about this presentation

Hardware Aware Training for Efficient Keyword Spotting on General Purpose and Specialized Hardware

Peter BLOUW, Senior Research Scientist and Head of Advanced Projects, Applied Brain Research

▸ Abstract (English)

More about this presentation

Characterization of Neural Networks Automatically Mapped on Automotive-grade Microcontrollers

Danilo PAU, Technical Director, IEEE and ST Fellow, STMicroelectronics Italia

▸ Abstract (English)

More about this presentation

Memory-Efficient, Limb Position-Aware Hand Gesture Recognition using Hyperdimensional Computing

Andy ZHOU, PhD Student, University of California Berkeley

▸ Abstract (English)

More about this presentation

10:00 am to 10:15 am — Break

10:15 am to 11:25 am — Poster Session

Session Chair: H. T. KUNG, Professor of Computer Science and Electrical Engineering, Harvard University

Session Chair: Wolfgang FURTNER, Distinguished Engineer System Architecture, Infineon Technologies

Privacy-Preserving Inference on the Edge: Mitigating a New Threat Model

Kartik PRABHU, MS Student, Stanford University

▸ Abstract (English)

An Ultra-low Power RNN Classifier for Always-On Voice Wake-Up Detection Robust to Real-World Scenarios

Emmanuel HARDY, Research Engineer, CEA Leti

▸ Abstract (English)

Resource Efficient Deep Reinforcement Learning for Acutely Constrained TinyML Devices

Filip SVOBODA, PhD Student, University of Oxford

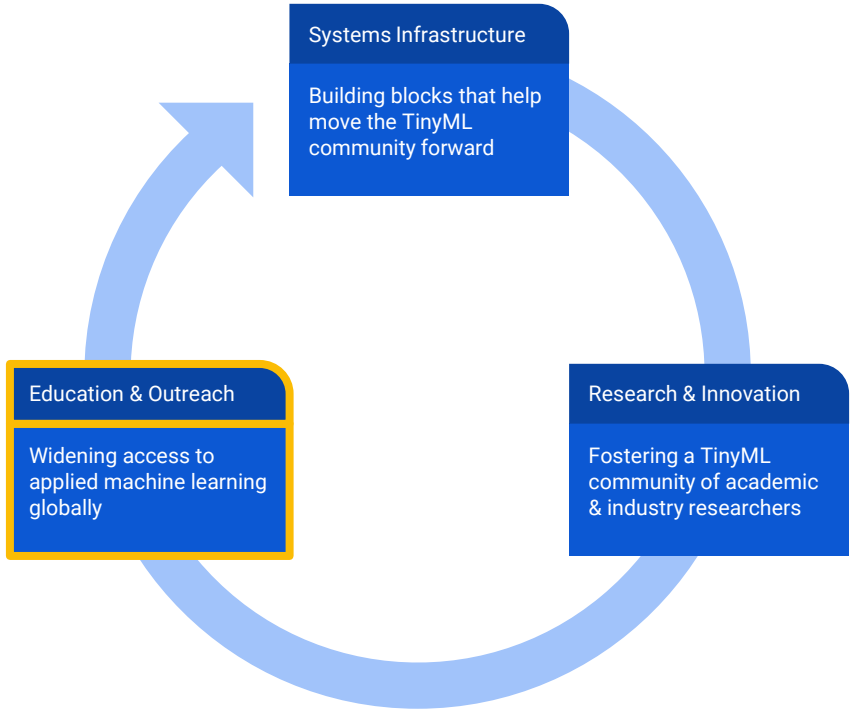
▸ Abstract (English)

Green Accelerated Hoeffding Tree

Eva GARCIA-MARTIN, Data Scientist, Ekkono Solutions

▸ Abstract (English)

TENT: Efficient Quantization of Neural Networks on the tiny Edge with Tapered FixFl


Systems Infrastructure
Building blocks that help move the TinyML community forward
Research & Innovation
Fostering a TinyML community of academic & industry researchers
Education & Outreach
Widening access to applied machine learning globally

Widening Access to Applied ML

- Broaden the reach of applied AI/ML resources globally
- From the Big Tech & Ivory Tower to the Greater Commons
- Focus on end-to-end ML application development

Widening Access to Applied Machine Learning

Vijay Janapa Reddi, Brian Plancher, Susan Kennedy, Laurence Moroney, Pete Warden, Anant Agarwal, Colby Banbury, Massimo Banzi, Benjamin Brown, Sharad Chitlangia, Radhika Ghosal, Rupert Jaeger, Srivatsan Krishnan, Daniel Leiker, Mark Mazumder, Dominic Pajak, Dhilan Ramaprasad, J. Evan Smith, Matthew Stewart, Dustin Tingley

Harvard University
Google

Abstract

Despite the expanding role of machine learning (ML), most ML resources and experts are in just a few countries and organizations. Broadening access to both computational and educational resources is critical to diffusing ML innovation. We suggest that TinyML, which applies ML to resource-constrained embedded devices, is an attractive means to this end. The required computing hardware is low cost and globally accessible, and it naturally encourages self-contained, end-to-end application development. Future ML engineers must have experience with the entire development process from data collection to deployment, and they must understand the ethical implications of their designs before deploying them. To this end, a collaboration between academia (Harvard University) and industry (Google and Arduino) produced a four-part massive online open course (MOOC) that provides application-driven instruction on the development of end-to-end solutions using TinyML. The course is openly available on the edX platform and has no prerequisites, beyond basic programming and was specifically designed for learners from diverse backgrounds. At the time of this writing, 35,000 learners have enrolled on edX. The first two courses progress from an overview of fundamental ML topics to greater detail on TinyML algorithms and applications. The third and fourth courses delve into ML-model deployment and ML-life-cycle management using microcontroller development boards. The courses introduce pupils to real-world applications, ML algorithms, data-set engineering, and the ethical considerations of these technologies through hands-on programming and deployment of TinyML applications in both the cloud and their own microcontrollers. To facilitate continued learning, community building, and collaboration beyond the course, we launched a standalone website, a Discourse forum, and an optional course-project competition. We also released the course materials publicly. Our hope is that these resources inspire and guide the next generation of ML practitioners and educators as well as further broaden access to cutting-edge ML technologies.

1 Introduction

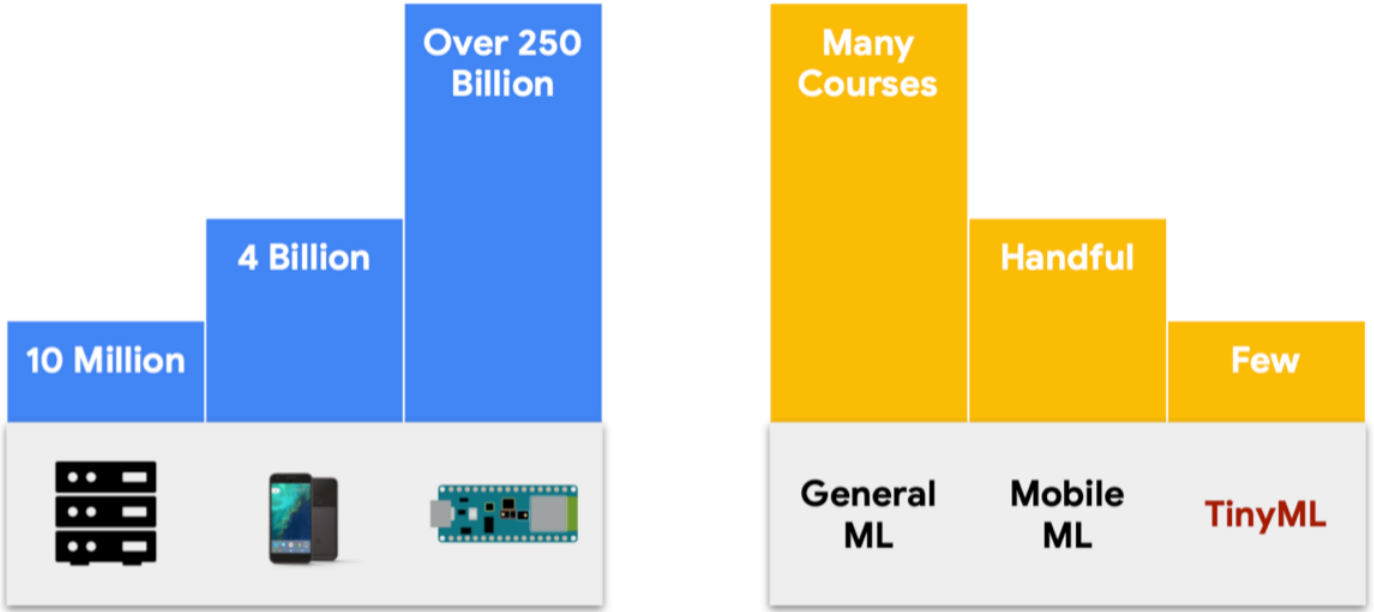
The past two decades have seen dramatic progress in machine learning (ML) from a purely academic discipline to a widespread commercial technology that serves a range of sectors. ML allows developers to improve business processes and human productivity through data-driven automation. Given applied ML's ubiquity and success, its commercial use should only increase. Existing ML applications cover a wide spectrum that includes digital assistants [1, 2], autonomous vehicles [3, 4], robotics [5], health care [6], transportation [7, 8], and security [9], education [10, 11], etc. New use cases are rapidly emerging, every few days there is a new ML use case.

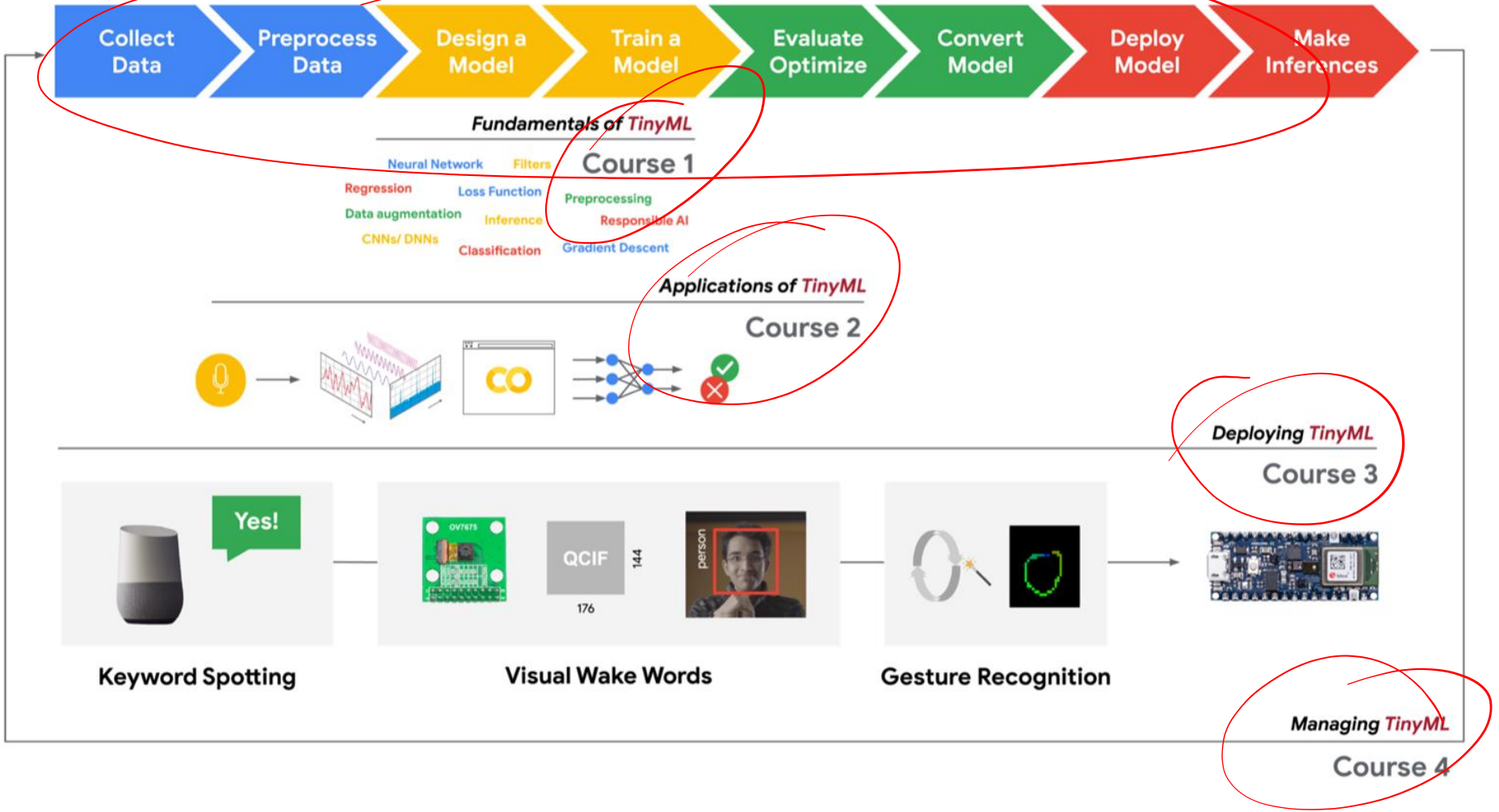
Figure 1: We designed a new applied-ML course motivated by real-world applications, covering not only the software (ML algorithms) and hardware (embedded systems) but also the product life cycle and responsible AI. To make it accessible and scalable, as well as to provide hands-on components, we focused on the emerging TinyML domain and released the course as a MOOC on edX.

The mass proliferation of this technology and associated jobs have great potential to improve society and uncover new opportunities for technological innovation, societal prosperity, and individual growth. But it all rests on the assumption that everyone, globally, has unfettered access to ML technologies, which isn't the case.

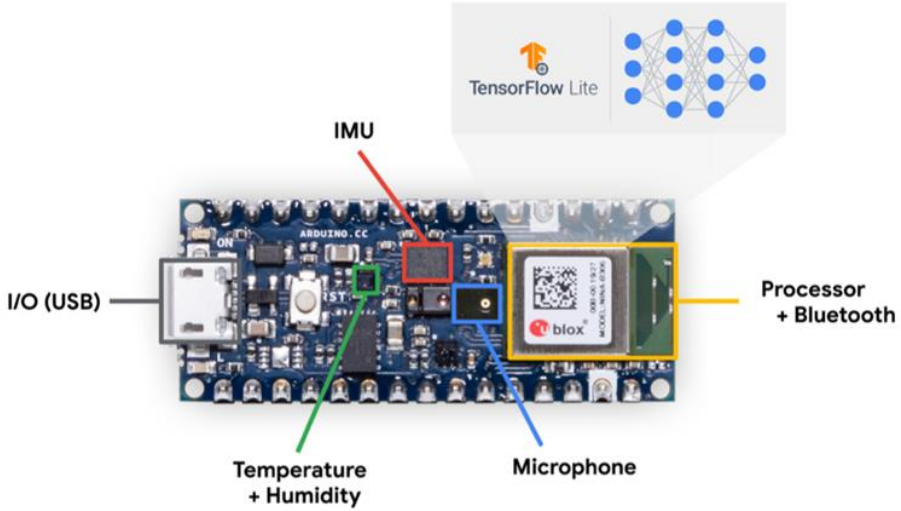
Widening access to applied ML faces three challenges. First is a shortage of ML educators at all levels [12, 13]. Second is insufficient resources to run ML models, especially as data sets continue to balloon. Training and running ML models often requires costly, high-performance hardware. Third is a growing gap between industry and academia, as even the best academic institutions and research labs struggle to keep pace with change. Addressing these critical issues requires innovative education and workforce training to prepare the next generation of applied ML engineers.

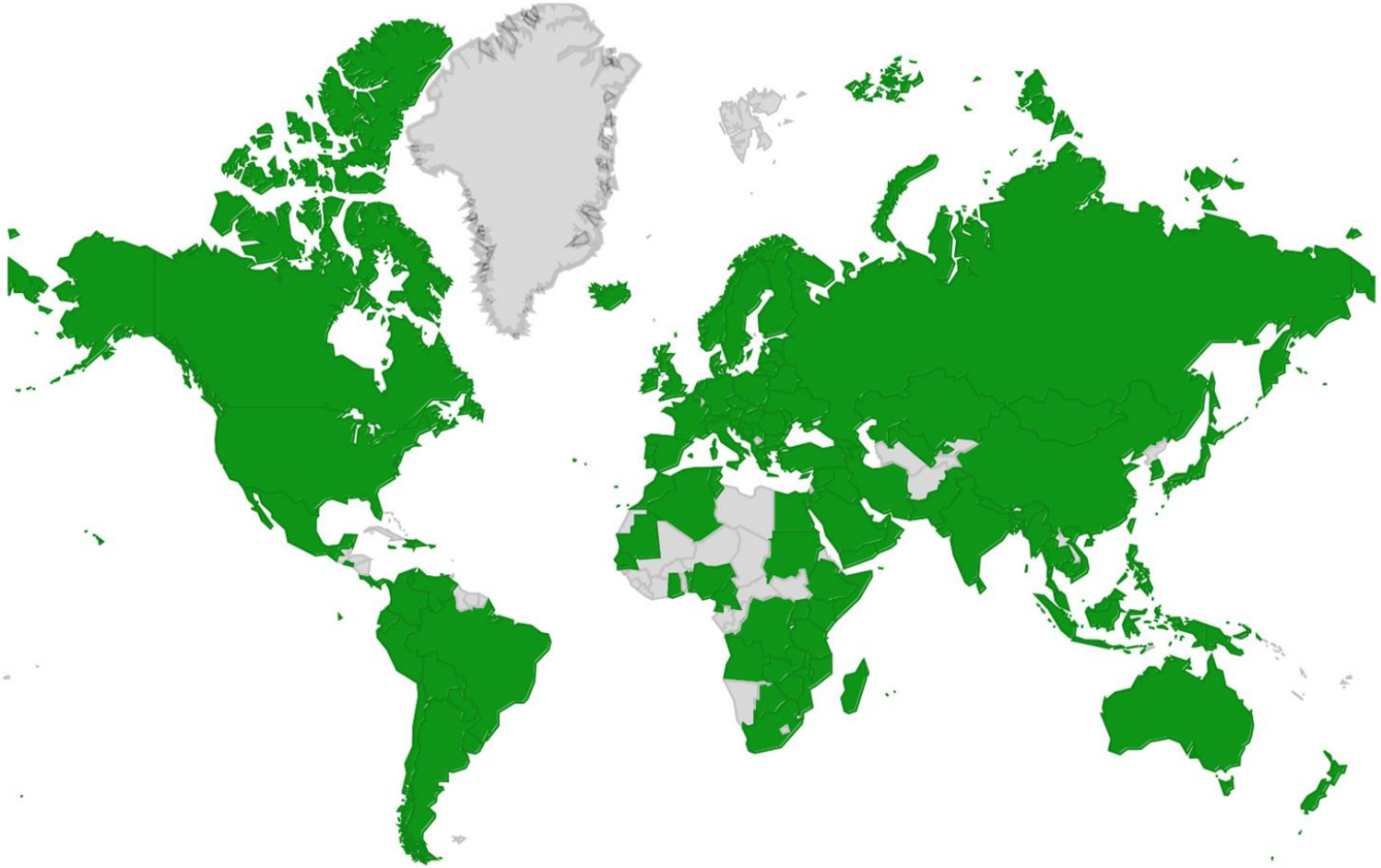
This paper presents a pedagogical approach, developed as an academic/industry collaboration led by Harvard University and Google, to address these challenges and thereby widen access to applied ML. We employ both cloud computing and low-cost hardware. Specifically, we use Google's free, open-source TensorFlow


Over 250 Billion
4 Billion
10 Million
Many Courses
Handful
Few
General ML
Mobile ML
TinyML


Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
Fundamentals of TinyML
Course 1
Neural Network
Filters
Regression
Loss Function
Preprocessing
Data augmentation
Inference
Responsible AI
CNNs/ DNNs
Classification
Gradient Descent
Applications of TinyML
Course 2
Deploying TinyML
Course 3
Yes!
OV7675
QCIF
144
176
person
Keyword Spotting
Visual Wake Words
Gesture Recognition
Managing TinyML
Course 4


ARDUINO
TINY MACHINE
LEARNING KIT
edX
TensorFlow


TensorFlow Lite
IMU
I/O (USB)
Processor
+ Bluetooth
Temperature
+ Humidity
Microphone



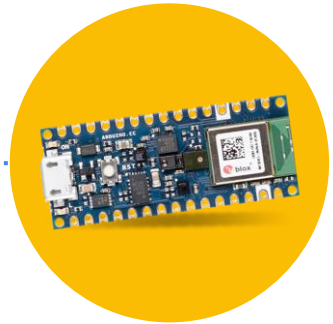
Responsible AI: Human-Centered Design

START → DESIGN → DEVELOPMENT → DEPLOYMENT → END

Course 1
Fundamentals of TinyML

Course 2
Applications of TinyML

Course 3
Deploying TinyML



Responsible AI: Human-Centered Design

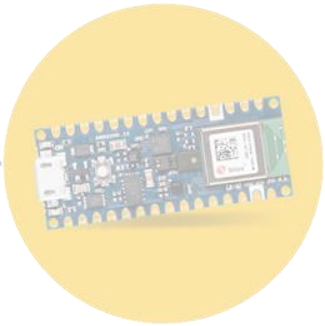
START | **DESIGN** | DEVELOPMENT | DEPLOYMENT | END

Course 1
Fundamentals of TinyML

- **What** am I building?
- **Who** am I building this for?
- What are the **consequences** for the user if it ***fails***?

Course 2
Applications of TinyML

Course 3
Deploying TinyML



Responsible AI: Human-Centered Design

START → DESIGN → DEVELOPMENT → DEPLOYMENT → END

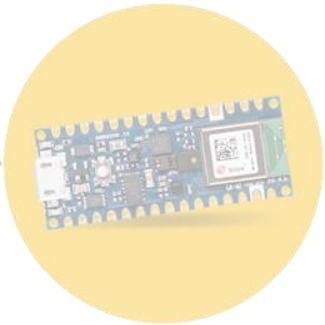
Course 1
Fundamentals of TinyML

- **What** am I building?
- **Who** am I building this for?
- What are the **consequences** for the user if it ***fails***?

Course 2
Applications of TinyML

- **What data** will be collected to train the model?
- Is the dataset **biased**?
- How can we **ensure** the model is **fair**?

Course 3
Deploying TinyML



Responsible AI: Human-Centered Design

Course 1
Fundamentals of TinyML

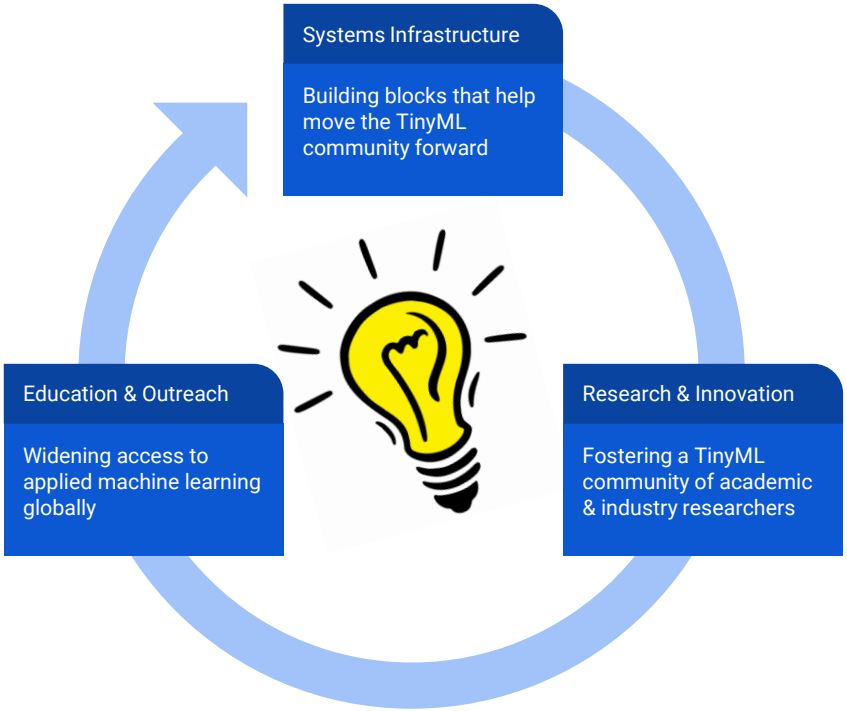
- **What** am I building?
- **Who** am I building this for?
- What are the **consequences** for the user if it ***fails***?

Course 2
Applications of TinyML

- **What data** will be collected to train the model?
- Is the dataset **biased**?
- How can we **ensure** the model is **fair**?

Course 3
Deploying TinyML

- How will **model drift** be monitored?
- How should **security breaches** be addressed?
- How should the user's **privacy** be protected?


Systems Infrastructure
Building blocks that help move the TinyML community forward
Research & Innovation
Fostering a TinyML community of academic & industry researchers
Education & Outreach
Widening access to applied machine learning globally

The Future of ML is Tiny and Bright

Challenges & Opportunities

Vijay Janapa Reddi
Harvard University