tinyML® Summit

Enabling Ultra-low Power Machine Learning at the Edge

February 12-13, 2020

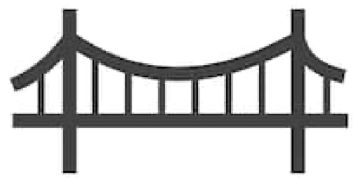
Burlingame, California

www.tinyML.org

Benchmarking Ultra-low Power Machine Learning Systems

Prof. Vijay Janapa Reddi
Harvard University
MLPerf Inference Chair
*(representing the work of **many** people!)*

TINY ML



MLPerf

What is MLPerf ?

A Community-driven
ML Benchmark Suite

Harvard University | Stanford University | University of Arkansas, Littlerock | University of California, Berkeley | University of Illinois, Urbana Champaign | University of Minnesota | University of Texas, Austin

University of Toronto

AI Labs.tw | Alibaba | AMD | Andes Technology | Arm | Baidu | Cadence | Intel | In-Q-Tel | Lenovo | MediaTek | Mentor Graphics | Microsoft | Myrtle

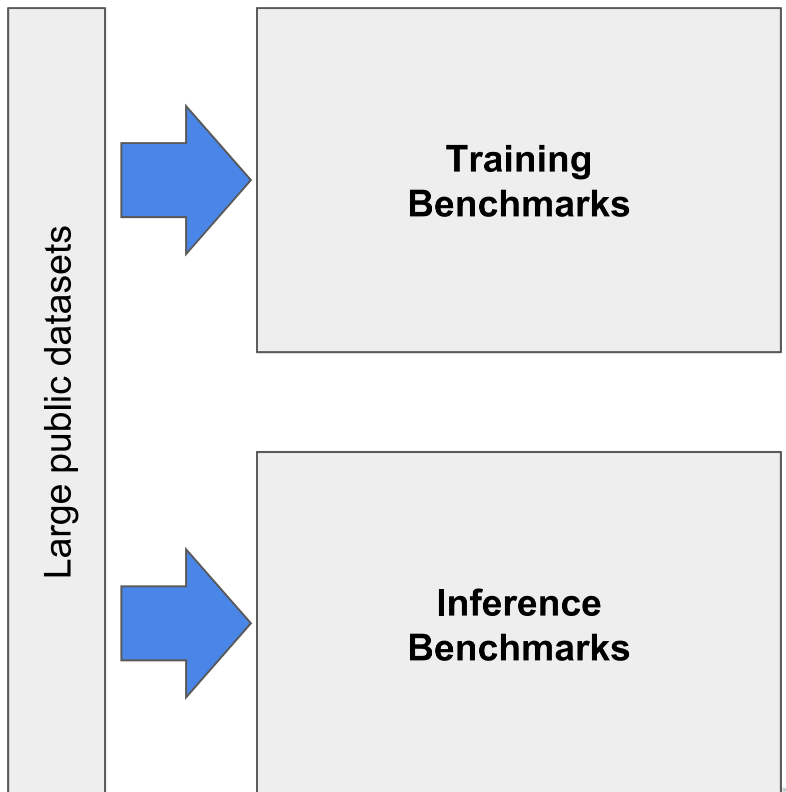
Calypso AI | Centaur Technology | Cerebras | Ceva | Cisco | Cray | CTuning Foundation | Mythic | NetApp | NVIDIA | One Convergence | PathPartner Technology | Qualcomm | Rpa2ai

Dividiti | DDN Storage | Edgify | Enflame Tech | Esperanto | Facebook | Google | Sambanova | Samsung S.LSI | Sigopt | Synopsys | Tencent | Tensyr | Teradyne

Groq | Habana | Hop Labs | Horizon Robotics | Iluvatar | Inspur | Inzone AI | Transpire Ventures | VMware | Volley | Wave Computing | Wiwynn | WekaIO | Xilinx

1,000+ members, 50+ organizations, 8+ universities

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 Goals

- **Enforce 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 play)

MLPerf **Inference** Benchmarks 0.5v

Area	Benchmark	Model	Dataset
Vision	Image Classification	MobileNet v1 ResNet50	ImageNet (224x224) ImageNet (224x224)
	Object Detection	SSD-MobileNet v1 SSD-ResNet34	MS-COCO (300x300) MS-COCO (1200x1200)
Language	Translation	Google NMT	WMT Eng-Germ

Inference v0.5 Results

MLPerf Inference v0.5 Results

November 6th, 2019

Any use of the MLPerf results and site must comply with the **MLPerf Terms of Use**.

You may wish to read the **Inference Overview** to better understand the results.

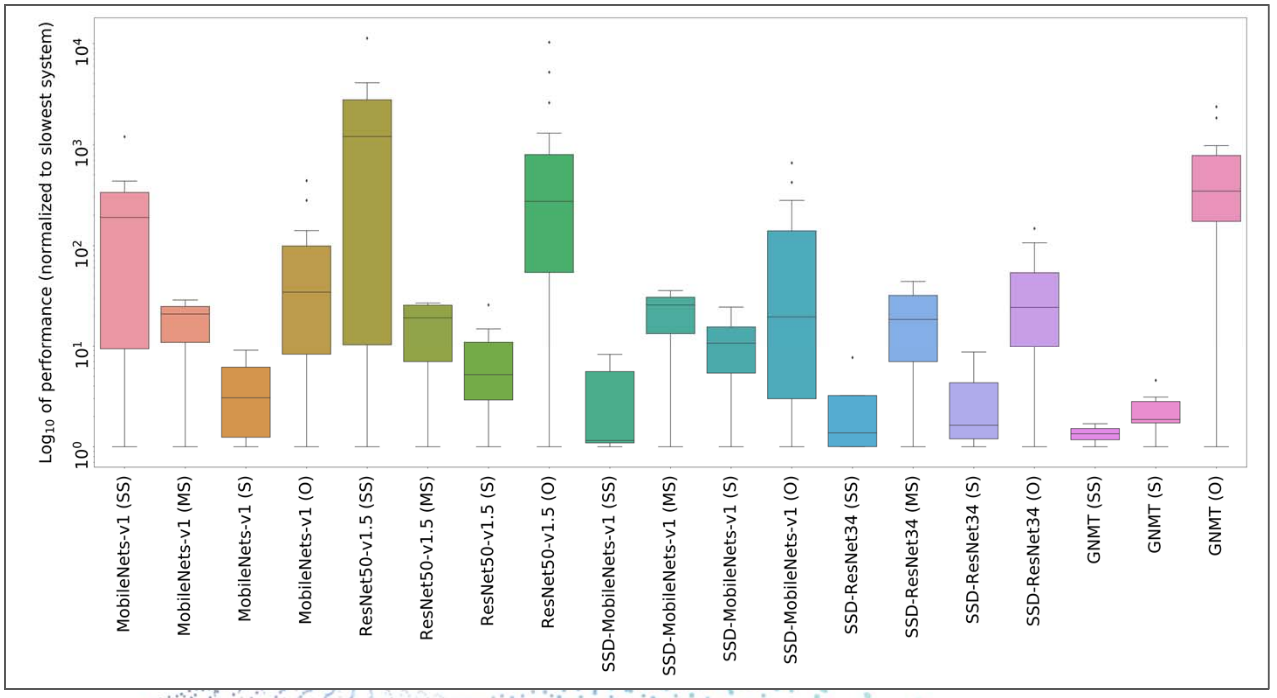
- (•) Closed Division Performance
- ( ) Open Division Performance

Closed Division Times

ID	Submitter	System	Benchmark results (Single Stream in milliseconds, MultiStream in no. streams, Server in QPS, Offline in inputs/second)												
			Image classification								Object detection				
			ImageNet				ImageNet				COCO				COCO
			MobileNet-v1				ResNet-50 v1.5				SSD w/ MobileNet-v1				SSD w/
			Stream	**MultiS**	**Server**	**Offline**	**Stream**	**MultiS**	**Server**	**Offline**	**Stream**	**MultiS**	**Server**	**Offline**	**Stream**
CATEGORY: Available															
Inf-0.5-1	Alibaba Cloud	Alibaba Cloud T4				17,473.60				5,540.10				7,431.20	
Inf-0.5-2	Dell EMC	Dell EMC R740			67,124.18	71,214.50			20,742.83	22,438.00			28,293.31	30,407.90	
Inf-0.5-3	Dell EMC	Dell EMC R740xd with 2nd generation Intel® Xeon® Scalable Processor									1.54			3,744.24	
Inf-0.5-4	Dell EMC	Dell EMC R740xd with 2nd generation Intel® Xeon® Scalable Processor									1.69			4,266.46	
Inf-0.5-5	dividiti	Raspberry Pi 4 (rpi4)	394.34				1,916.65								
Inf-0.5-6	dividiti	Raspberry Pi 4 (rpi4)	103.60				448.31								
Inf-0.5-7	dividiti	Linaro HiKey960 (hikey960)	121.11				518.07								
Inf-0.5-8	dividiti	Linaro HiKey960 (hikey960)	50.77				203.99								
Inf-0.5-9	dividiti	Linaro HiKey960 (hikey960)	143.07				494.90								

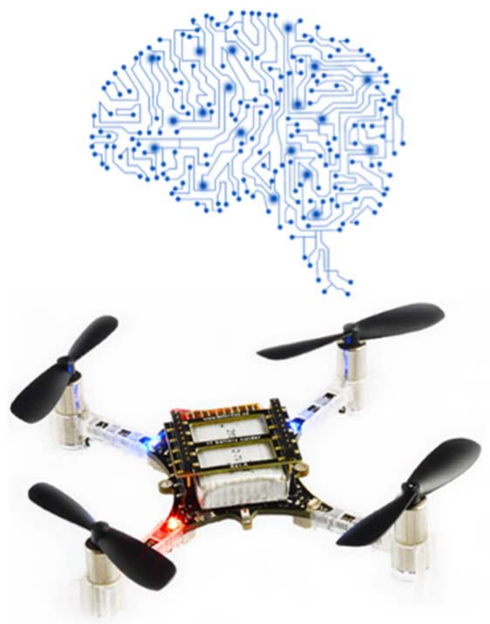
Inference Results

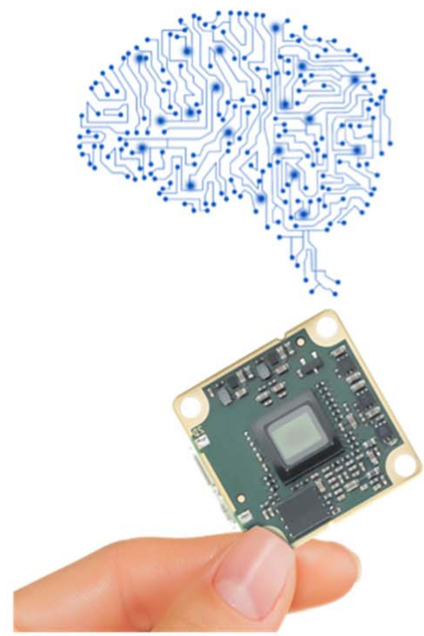
- 600+ inference results
- Over 30 systems submitted
- 10,000x difference in performance



MLPerf

What about TinyML systems?

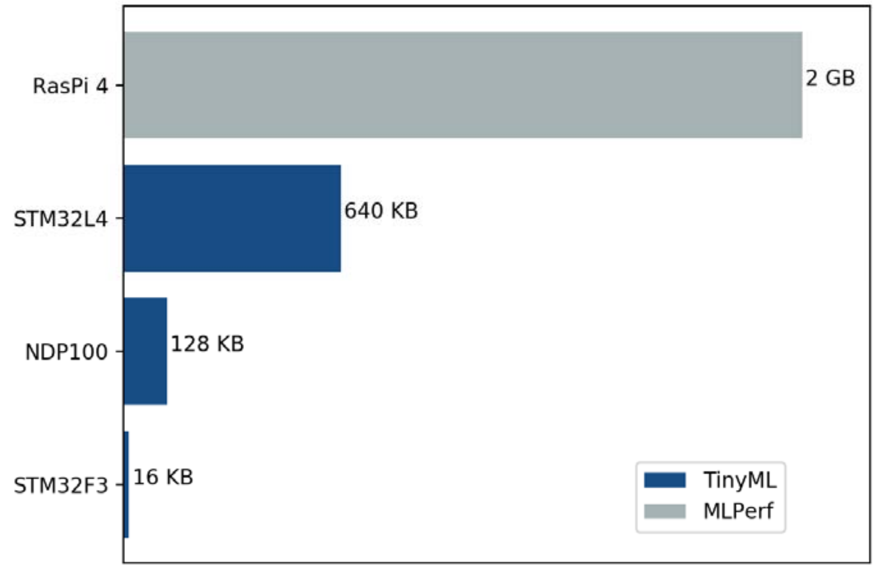




MLPerf

TinyML Challenges for ML Benchmarking

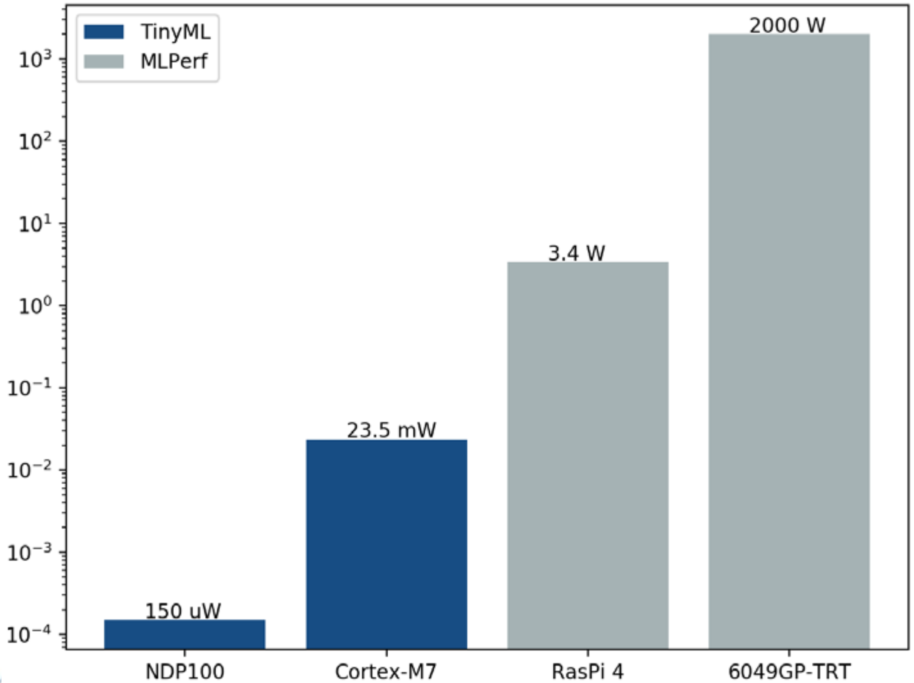
- Resources are extremely condensed in the tinyML devices
- Need to define a methodology to load the SUT for evaluation
- **How can we come up with a methodology that works across many different systems?**



MLPerf

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?**



MLPerf

TinyML Challenges for ML Benchmarking

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

- What are the rules that apply for tinyML?

- Which of the optimizations should be allowed while still enabling fair comparisons?

tinyMLPerf

Working Group Members

RENESAS

HARVARD
John A. Paulson
School of Engineering
and Applied Sciences

GREENWAVES
TECHNOLOGIES

OctoML

intel

Southeast
Missouri State University

W
UNIVERSITY of
WASHINGTON

cisco

Google

ambiq micro

ASU
Arizona State

Microsoft

Cogneefy

KU LEUVEN

arm

Reality AI
Explainable AI at the Edge

ZOOX

cādence

SiMa ai

AONdevices
AI | DSP | ASIC

Red Hat

MEDIATEK

SYNTIANT

SAMSUNG

CIRRUS LOGIC

TinyML Tasks 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

MLPerf

tinyMLPerf Benchmark Design Choices

Big Questions	Inference
1. Benchmark definition	What is the definition of a benchmark task?

MLPerf

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?

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?

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?

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?

MLPerf

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?

MLPerf

ML Benchmark Design Choices: Examples

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

ML Benchmark Design Choices: Examples

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

ML Benchmark Design Choices: Examples

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

ML Benchmark Design Choices: Examples

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.

tinyMLPerf Benchmark Strawperson

Task Category	Use Case	Dataset
Audio	Audio Wake Words	Speech Commands
Visual	Visual Wake Words	Google's VWW dataset
Behavioral	Anomaly Detection	Physionet, HAR, DSA, Opportunity

tinyMLPerf
Benchmarking Resource-constrained Machine Learning Systems

Colby Banbury, Max Lam, Vijay Janapa Reddi, David Kanter, Amin Fazel, Xinyuan Huang, Danilo Pietro Pau
and the tinyMLPerf working group
Harvard University, MLPerf, Samsung Semiconductor, Inc., Cisco Systems, STMicroelectronics

Abstract

Advancements in ultra-low-power machine learning (tinyML) hardware promises to unlock an entirely new class of intelligent applications. However, the complexity and dynamicity of the field obscure the measurement of progress and make application design decisions intractable. In order to enable the continued innovation, a fair, replicable and robust method of comparison is needed. Since progress is often the result of increased hardware capability, a reliable tinyML hardware benchmark is required.

To fulfill the need, we have created a community effort to extend the scope of the existing MLPerf benchmarking suite to include tinyML devices. With the help of over 75 member organizations, this group, dubbed tinyMLPerf, has begun the process of developing a benchmarking suite.

Existing Benchmarks

Existing benchmarks do not represent ML workloads or they are too large to fit on tinyML constrained processors.

Benchmark	ML?	Power?	Tiny?
CoreMark	×	√	√
MLMark	√	×	×
MLPerf Inference	√	√	×
TinyML Requirements	√	√	√

Survey of tinyML Use Cases, Models, and Datasets

The landscape of tinyML use cases is large and wide. We surveyed many state of the art use cases to determine the scope of a representative tinyMLPerf benchmark.

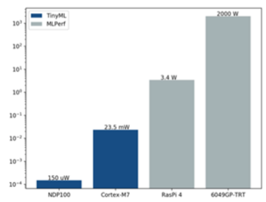
Input Type	Use Case	Model Type	Dataset
Audio	Audio Wake Words Context Recognition Control Words Keyword Detection	DNN CNN RNN LSTM	Speech Commands Audioset ExtraSensory Freesound DCASE
Image	Visual Wake Words Object Detection Gesture Recognition Object Counting Text Recognition	DNN CNN SVM Decision Tree 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

Challenges: Energy

An ideal tinyML benchmark would profile the energy efficiency of each system. Unfortunately, there are many challenges in fairly measuring energy usage:

- Maintaining the accuracy of energy measurement across the diverse range of processors, silicon technologies and memory architectures.
- Determining the scope of the measurement.
  - Memories (RAM, FLASH)? Peripherals? PLL?
  - Pre/Post processing? Interfaces ?
- Measuring energy consumption without significant work or alterations to the SUT.
- Preventing energy measurements from impacting the other metrics

Scope of tinyMLPerf vs. MLPerf Inference: Power Envelope



tinyML Systems consume drastically less power than traditional ML systems, yet still cover a large scope.

Challenges: Model Infancy

Despite the nascency of the field, tinyML systems are already diverse. This poses a number of challenges:

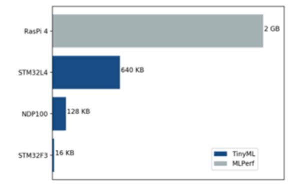
- Lack of standardization makes collecting metrics harder to formalize.
- Novel architectures have drastically different constraints and topologies.
- System requirements vary significantly across use cases.
- Performances are difficult to normalize.
- Different manufacturing technologies jeopardize comparisons with a fairly acceptable methodology.

Challenges: Memory

Memory constraints are one of the primary motivating factors for the creation of a tinyML specific benchmark. However, memory constraints add additional developmental challenges:

- Traditional benchmarks use NN models that are far too large in weights and activations.
- The overhead of the benchmark is more significant factor, pushing the need for non-intrusive inspection of key metric indicators.
- The System Under Test cannot hold the entire testing set, w/out involving host communication.
- Software (e.g. RTOS, drivers, built-in libraries) will require further discrimination.

Scope of tinyMLPerf vs. MLPerf Inference: Memory Envelope



Limited memory is a significant constraint for tinyML systems and the degree of which can vary widely.

Challenges: Processors Heterogeneity

tinyML is still a new field. It creates an opportunity to foster growth through community efforts (with industry support) but also poses a number of challenges for developing a robust benchmark that features industry acceptance and consensus:

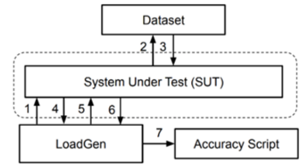
- Widely accepted tinyML neural network models.
- Large open source tinyML datasets.
- Frameworks are still evolving and few de-facto industry standards have become popular therefore model portability/interoperability is evolving:
  - e.g. tensorflow, keras, pytorch, mxnet, caffe etc, associated interoperable file formats (e.g. tflite, keras, onnx, nnef).

Use Cases Selected for v0.1

The criteria for the preliminary selection was to select three use cases that represented the scope of tinyML in terms of input type, size, neural network model type, and maturity. Model selection is still in progress.

Use Case	Dataset
Audio Wake Words	Speech Commands
Visual Wake Words	Google's VWW dataset
Anomaly Detection	Physionet, HAR, DSA, Opportunity

LoadGen and System Under Test



(workflow is subject to change)

Working Group Member Organizations

Join Us!

tinyMLPerf
Help us create a tinyML Benchmark!

Join https://groups.google.com/forum/#!members/mlperf-tiny

Summary

- **Benchmarking ML Systems is hard** due to the fragmented ecosystem
- MLPerf is a **community-driven ML benchmark** for the HW/SW industry
- Lets build a tinyML benchmark suite to **enable tinyML system comparison**
    - Defines Tasks, Scenarios, Datasets, Methods
    - Establish clear set of metrics and divisions
    - Allows for hardware/software flexibility

arXiv:1910.01500v2 [cs.LG] 30 Oct 2019

MLPerf Training Benchmark

Peter Mattson[1] Christine Cheng[2] Cody Coleman[3] Greg Diamos[4] Paulius Micikevicius[5] David Patterson[1,6] Hanlin Tang[2] Gu-Yeon Wei[7] Peter Bailis[3] Victor Bittorf[1] David Brooks[7] Dehao Chen[1] Debojyoti Dutta[8] Udit Gupta[7] Kim Hazelwood[9] Andrew Hock[10] Xinyuan Huang[8] Bill Jia[9] Daniel Kang[3] David Kanter[11] Naveen Kumar[1] Jeffery Liao[12] Guokai Ma[2] Deepak Narayanan[3] Tayo Oguntebi[1] Gennady Pekhimenko[13] Lillian Pentecost[7] Vijay Janapa Reddi[7] Taylor Robie[1] Tom St. John[14] Carole-Jean Wu[9] Lingjie Xu[15] Cliff Young[1] Matei Zaharia[3]

arXiv:submit/2913405 [cs.LG] 4 Nov 2019

MLPerf Inference Benchmark

Vijay Janapa Reddi[1] Christine Cheng[2] David Kanter[3] Peter Mattson[4] Guenther Schmuelling[5] Carole-Jean Wu[6] Brian Anderson[4] Maximilien Breughe[7] Mark Charlebois[8] William Chou[8] Ramesh Chukka[2] Cody Coleman[9] Sam Davis[10] Greg Diamos[11] Jared Duke[4] Dave Fick[12] J. Scott Gardner[13] Itay Hubara[14] Sachin Idgunji[7] Thomas B. Jablin[4] Jeff Jiao[15] Tom St. John[16] Pankaj Kanwar[4] David Lee[17] Jeffery Liao[18] Anton Lokhmotov[19] Francisco Massa[6] Paulius Micikevicius[7] Colin Osborne[20] Gennady Pekhimenko[21] Arun Tejusve Raghunath Rajan[2] Dilip Sequeira[7] Ashish Sirasao[22] Fei Sun[23] Hanlin Tang[2] Michael Thomson[24] Frank Wei[25] Ephrem Wu[22] Lingjie Xu[25] Koichi Yamada[2] Bing Yu[17] George Yuan[7] Aaron Zhong[15] Peizhao Zhang[6] Yuchen Zhou[26]

Machine-learning (ML) hardware and software system demand is burgeoning. Driven by ML applications, the number of different ML inference systems has exploded. Over 100 organizations are building ML inference chips, and the systems that incorporate existing models span at least three orders of magnitude in power consumption and four orders of magnitude in performance; they range from embedded devices to data-center solutions. Fueling the hardware are a dozen or more software frameworks and libraries. The myriad combinations of ML hardware and ML software make assessing ML-system performance in an architecture-neutral, representative, and reproducible manner challenging. There is a clear need for industry-wide standard ML benchmarking and evaluation criteria. MLPerf Inference answers that call. Driven by more than 30 organizations as well as more than 200 ML engineers and practitioners, MLPerf implements a set of rules and practices to ensure comparability across systems with wildly differing architectures. In this paper, we present the method and design principles of the initial MLPerf Inference release. The first call for submissions garnered more than 600 inference-performance measurements from 14 organizations, representing 44 systems that show a wide range of capabilities.

1 Introduction

Machine learning (ML) powers a variety of applications from computer vision (He et al., 2016; Goodfellow et al., 2014; Liu et al., 2016; Krizhevsky et al., 2012) and natural-language processing (Vaswani et al., 2017; Devlin et al., 2018) to self-driving cars (Xu et al., 2018; Badrinarayanan et al., 2017) and autonomous robotics (Levine et al., 2018). These applications are deployed at large scale and require substantial investment to optimize inference performance. Although training of ML models has been a development bottleneck and a considerable expense (Amodei & Hernandez, 2018), inference has become a critical workload, since models can serve as many as 200 trillion queries and perform over 6 billion translations a day (Lee et al., 2019b).

[1]Harvard University [2]Intel [3]Real World Insights [4]Google [5]Microsoft [6]Facebook [7]NVIDIA [8]Qualcomm [9]Stanford University [10]Myrtle [11]Landing AI [12]Mythic [13]Advantage Engineering [14]Habana Labs [15]Alibaba T-Head [16]Tesla [17]MediaTek [18]Synopsys [19]dividiti [20]Arm [21]University of Toronto [22]Xilinx [23]Alibaba (was Facebook) [24]Centaur Technology [25]Alibaba Cloud [26]General Motors. MLPerf Inference is the product of (1) individuals that led the benchmarking effort from the various organizations and (2) submitters that produced the first set of benchmark results. It takes both groups to have a successful industry benchmark. We credit the submitters and their organizations in the acknowledgments. Send correspondence email to *vj@eecs.harvard.edu* and see *mlperf.org*.

To address these growing computational demands, hardware, software, and system developers have focused on inference performance for a variety of use cases by designing optimized ML hardware and software systems. Estimates indicate that over 100 companies are producing or are on the verge of producing optimized inference chips. By comparison, only about 20 companies target training.

Each system takes a unique approach to inference and presents a trade-off between latency, throughput, power, and model quality. For example, quantization and reduced precision are powerful techniques for improving inference latency, throughput, and power efficiency at the expense of accuracy (Han et al., 2015; 2016). After training with floating-point numbers, compressing model weights enables better performance by decreasing memory-bandwidth requirements and increasing computational throughput (e.g., by using wider vectors). Similarly, many weights can be removed to boost sparsity, which can reduce the memory footprint and the number of operations (Han et al., 2015; Molchanov et al., 2016; Li et al., 2016). Support for these techniques varies among systems, however, and these optimizations can drastically reduce final model quality. Hence, the field needs an ML inference benchmark that can quantify these trade-offs in an architecturally neutral, representative, and reproducible manner.

Join us!

Google group: https://groups.google.com/forum/#!members/mlperf-tiny

vj@eecs.harvard.edu
cbanbury@g.harvard.edu

Copyright Notice