CS-723
MLPerf Training Benchmark

Presented by Ayan Chakraborty

PARALLEL SYSTEMS ARCHITECTURE LAB

What are Benchmarks?

- Collection of representative workloads used in a specific field
  - SPEC for Desktop Applications
  - TPC for Databases
  - CloudSuite for Cloud Applications

- Model typical application behaviour in the real world usually at smaller scales

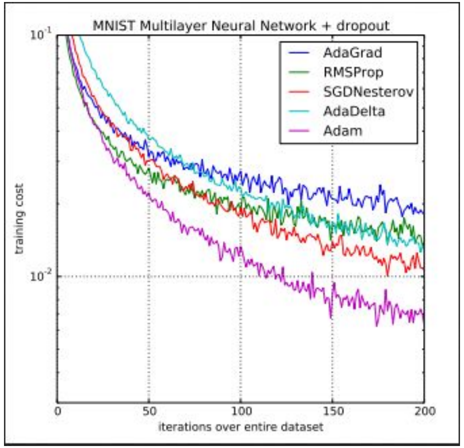
Why are Benchmarks important?

- Enables studying system-level characteristics
  - If benchmarks are representative, then your system behaviour is also representative!
- Exposes bottlenecks in the HW and SW stacks
  - Enables building innovative solutions to solve these bottlenecks
- Sets fair standards for comparing different HW - SW solutions

Q1) What is the problem?

- Lack of a comprehensive benchmark for ML Training!

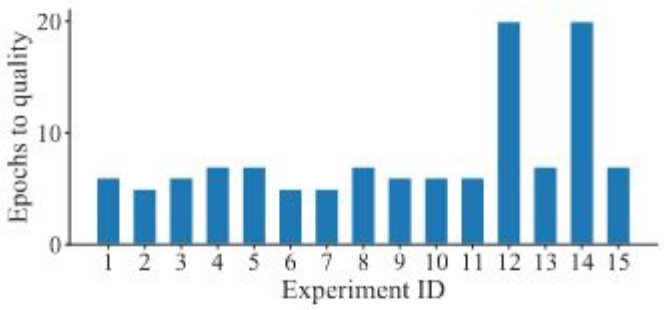
- ML Training is significantly different from traditional applications
  - Optimizations may increase time to reach accuracy target

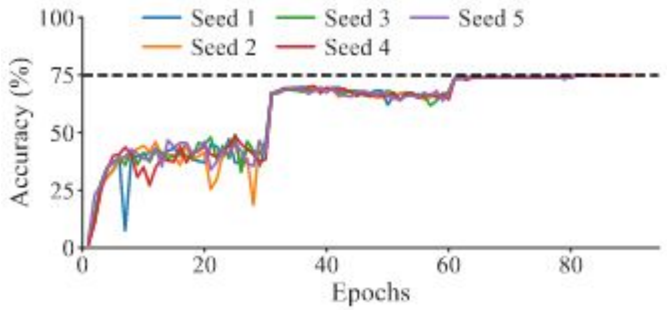


Q1) What is the problem?

- Lack of a comprehensive benchmark for ML Training!

- ML Training is significantly different from traditional applications
  - Stochastic in nature

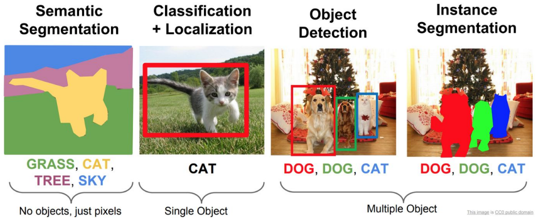


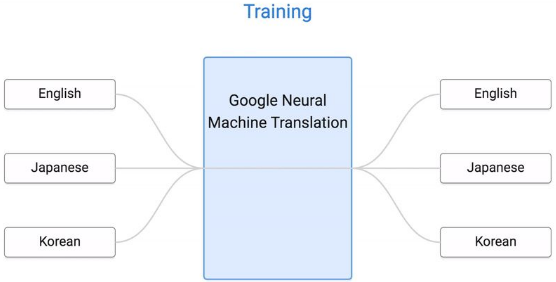


Q1) What is the problem?

- Lack of a comprehensive benchmark for ML Training!

- ML Training is significantly different from traditional applications
  - Diverse set of models for different application domains





Q1) What is the problem?

- Lack of a comprehensive benchmark for ML Training!

- ML Training is significantly different from traditional applications
  - Diverse set of HW-SW solutions make it hard to benchmark fairly

Q1) What is the problem?

- Lack of a comprehensive benchmark for ML Training!

- ML Training is significantly different from traditional applications
  - Optimizations may increase time to reach accuracy target
  - Stochastic in nature
  - Diverse set of models for different application domains
  - Diverse set of HW-SW solutions make it hard to benchmark fairly

Prior work do not address all challenges together!

Q2) What are the insights?

PARALLEL SYSTEMS ARCHITECTURE LAB

Problem	Insight
Optimizations may increase time to reach accuracy target	Choose the performance metric as the time to train a model to a defined accuracy target

Q2) What are the insights?

Problem	Insight
Stochastic in nature	▪ Create strict timing rules to exclude non relevant operations and consider multiple runs ▪ Choose reasonably accuracy targets to ensure consistency and full duration training runs

Q2) What are the insights?

Problem	Insight
Diverse set of models for different application domains	▪ Use industry feedback to choose representative tasks across major ML areas ▪ Provide fixed reference of small but powerful model architectures to solve these tasks

Q2) What are the insights?

Problem	Insight
Diverse set of HW-SW solutions	Limit the space of modifiable hyper-parameters, and allow hyperparameter borrowing

Q3) What is the solution?

PARALLEL SYSTEMS ARCHITECTURE LAB

Benchmark	Data set	Model	Quality Threshold
Image classification	ImageNet (Deng et al., 2009)	ResNet-50 v1.5 (MLPerf, 2019b)	74.9% Top-1 accuracy
Object detection (lightweight)	COCO 2017 (Lin et al., 2014)	SSD-ResNet-34 (Liu et al., 2016)	21.2 mAP
Instance segmentation and object detection (heavyweight)	COCO 2017 (Lin et al., 2014)	Mask R-CNN (He et al., 2017a)	37.7 Box min AP, 33.9 Mask min AP
Translation (recurrent)	WMT16 EN-DE (WMT, 2016)	GNMT (Wu et al., 2016)	21.8 Sacre BLEU
Translation (nonrecurrent)	WMT17 EN-DE (WMT, 2017)	Transformer (Vaswani et al., 2017)	25.0 BLEU
Recommendation	MovieLens-20M (GroupLens, 2016)	NCF (He et al., 2017b)	0.635 HR@10
Reinforcement learning	Go (9x9 Board)	MiniGo (MLPerf, 2019a)	40.0% Professional move prediction

- 7 workloads, each with its own accuracy target and set of modifiable hyper-parameters
- Submissions are peer-reviewed and checked for reproducibility

Q4) What is the takeaway message?

- Benchmarking for ML training applications is hard!
- Must consider several factors to ensure:
  - Set of representative workloads
  - Score metrics
  - Rules for fair comparisons
- Results show average performance improved between two submission rounds
  - Driving rapid performance and scaling improvement

Q5) Will this paper win the test of time?

- No
- Very little analysis of results
  - Does not even show which platform achieves the best results for each workload!
- Benchmarks contain some models which are not useful anymore
- Transformer models are widespread now

Q6) Why should this paper not have appeared at a top conference?

- Very little analysis of the results
- Does not identify any bottlenecks
- Does not provide explanations behind their observations
  - Why speedup between two different submission rounds?
  - Why did the number of chips necessary to produce the fastest time to solution increase by 5x?

Thank you

CS-723
MLPerf Inference Benchmark

Presented by Ayan Chakraborty

PARALLEL SYSTEMS ARCHITECTURE LAB

Why separate Inference from Training?

PARALLEL SYSTEMS ARCHITECTURE LAB

- Different application and system level requirements
  - Only forward pass with fixed weights
  - Lesser computational and memory footprint

- More aggressive optimizations possible
  - Much more diverse models and platforms
  - More diverse use cases in the real world

Why separate Inference from Training?

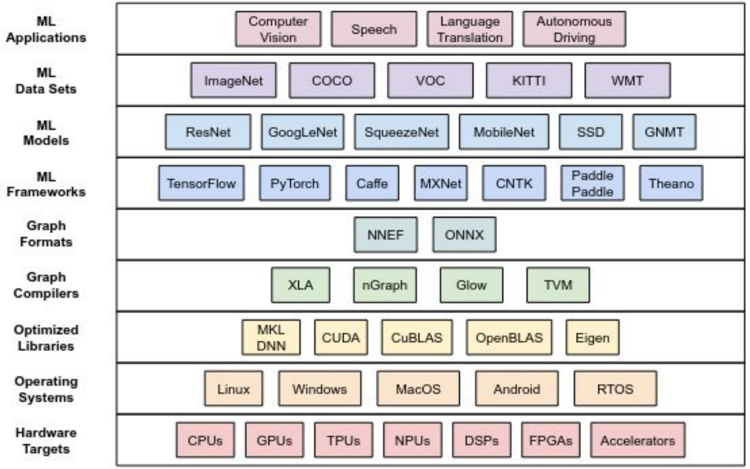
- Inference tasks usually have strict service level objectives (SLO) constraints
  - Each inference request usually has a latency bound
  - 99% of all requests have to be served within the latency bound
  - This latency bound is referred to as the tail latency constraint

- Accuracy loss is acceptable depending on use cases
  - Do not need full accuracy to classify dogs and cats
  - Need full accuracy for autonomous driving!

Q1) What is the problem?

- Lack of a comprehensive benchmark for ML Inference!

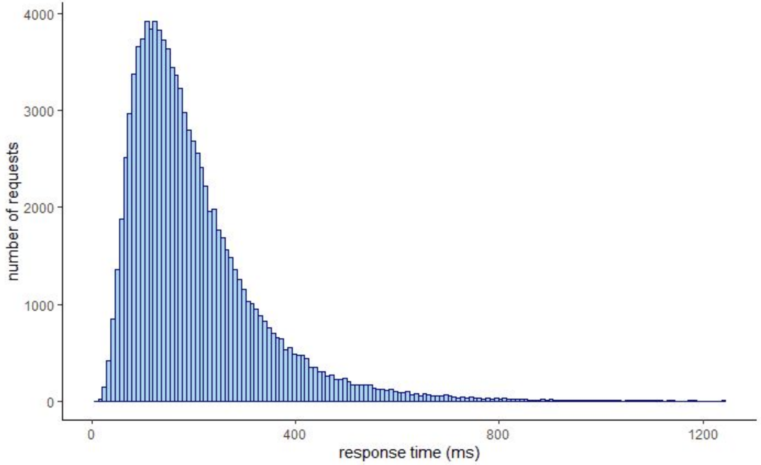
- Much more diverse range of devices and use cases
  - 100 companies targeting inference compared to 20 for training



Q1) What is the problem?

- Strict tail latency constraints depending on use-case

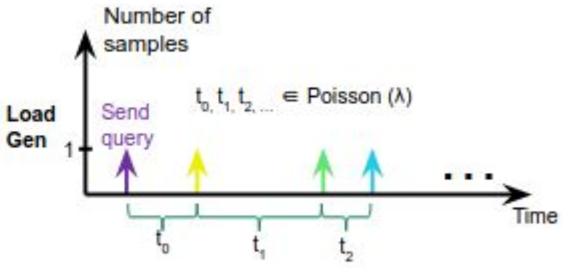
- Can sacrifice model quality to reduce latency, reduce total cost of ownership (TCO), or increase throughput

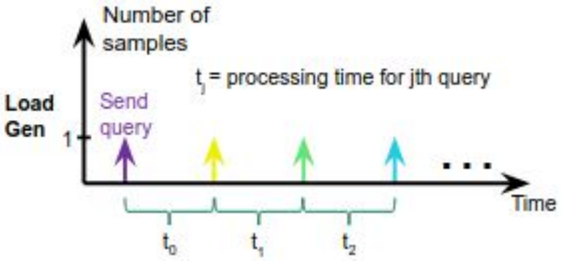


Q1) What is the problem?

- Can be deployed in a wide range of scenarios

- Autonomous cars, Online services, Edge devices
  - Different request stream characteristics
  - Different goals





Q1) What is the problem?

- Lack of a comprehensive benchmark for ML Inference tasks!

- ML Inference is different from ML Training & traditional applications
  - More diverse models and devices
  - Strict tail latency constraints
  - Wide range of deployment scenarios
  - Stochastic in nature

Prior work do not address all challenges together!

Q2) What are the insights?

Problem	Insight
Much more diverse range of devices and use cases	▪ Choose vision and translation as the main two tasks based on industry feedback ▪ Choose both light and heavy models, and provide reference weights, with individual quality targets

Q2) What are the insights?

Problem	Insight
Much more diverse range of devices and use cases	▪ Allow untimed pre-processing, mathematically equivalent deviations and different number formats ▪ Obtain reference weights for light models using quantization aware training

Q2) What are the insights?

PARALLEL SYSTEMS ARCHITECTURE LAB

Problem	Insight
Strict tail latency constraints + Wide range of deployment scenarios	▪ Use four realistic categories: Single Stream, Multi Stream, Server and Offline ▪ Each combination of models and scenarios have different performance metrics and tail latency constraints

Q2) What are the insights?

Problem	Insight
Stochastic in nature	Set different query count requirements for different task and scenario combinations to ensure statistically robust results, and to capture steady state behaviour

Q3) What is the solution?

PARALLEL SYSTEMS ARCHITECTURE LAB

AREA	TASK	REFERENCE MODEL	DATA SET	QUALITY TARGET
VISION	IMAGE CLASSIFICATION (HEAVY)	RESNET-50 V1.5 25.6M PARAMETERS 8.2 GOPS / INPUT	IMAGENET (224X224)	99% OF FP32 (76.456%) TOP-1 ACCURACY
VISION	IMAGE CLASSIFICATION (LIGHT)	MOBILENET-V1 224 4.2M PARAMETERS 1.138 GOPS / INPUT	IMAGENET (224X224)	98% OF FP32 (71.676%) TOP-1 ACCURACY
VISION	OBJECT DETECTION (HEAVY)	SSD-RESNET-34 36.3M PARAMETERS 433 GOPS / INPUT	COCO (1,200X1,200)	99% OF FP32 (0.20 MAP)
VISION	OBJECT DETECTION (LIGHT)	SSD-MOBILENET-V1 6.91M PARAMETERS 2.47 GOPS / INPUT	COCO (300X300)	99% OF FP32 (0.22 MAP)
LANGUAGE	MACHINE TRANSLATION	GNMT 210M PARAMETERS	WMT16 EN-DE	99% OF FP32 (23.9 SACREBLEU)

5 workloads, each with its own accuracy target

Q3) What is the solution?

PARALLEL SYSTEMS ARCHITECTURE LAB

Scenario	Query Generation	Metric	Samples/Query	Examples
Single-stream (SS)	Sequential	90th-percentile latency	1	Typing autocomplete, real-time AR
Multistream (MS)	Arrival interval with dropping	Number of streams subject to latency bound	N	Multicamera driver assistance, large-scale automation
Server (S)	Poisson distribution	Queries per second subject to latency bound	1	Translation website
Offline (O)	Batch	Throughput	At least 24,576	Photo categorization

4 different realistic deployment scenarios

- 20 different combinations of model + scenario
- Each combination has individual tail latency constraint if applicable and request stream characteristics

Q4) What is the takeaway message?

- Benchmarking for ML inference applications is even more hard!
- Apart from representativeness, also need to worry about constraints and goals depending on use case
- Results show that latency constraints result in throughput degradation and under utilization of resources
- Hence, optimizing systems for latency is challenging and underappreciated

Q5) Will this paper win the test of time?

- Good paper, but No
- Does not offer insights into their results
- Benchmark unfortunately does not contain any transformer models which are the current state of the art in most machine learning tasks
- Some application domains are missing such as Speech Recognition

Q6) Why should this paper not have appeared at a top conference?

- Does not analyse their results well
  - Translation task suffers higher throughput degradation compared to vision tasks but no explanation

- Do not analyse why certain systems have a lower throughput drop compared to others

- Although the paper identifies inefficient batching as a bottleneck, it does not propose any solutions to overcome it

Thank you