http://inst.eecs.berkeley.edu/~ee290-2/sp21/

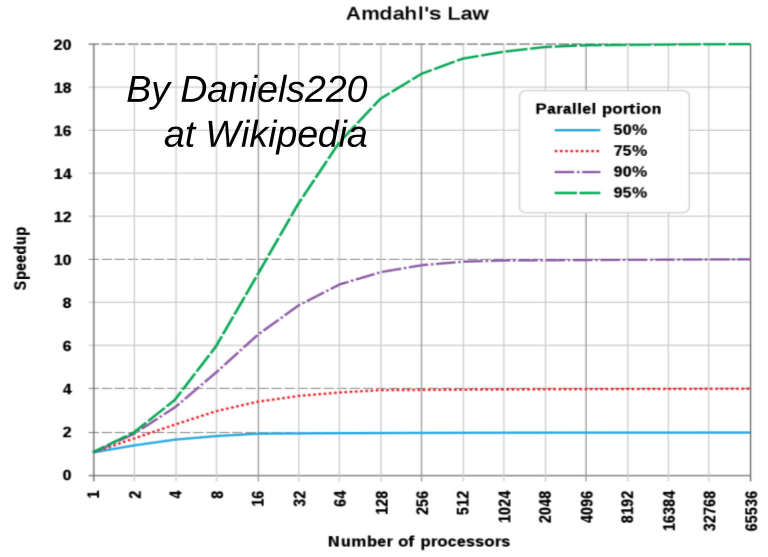
Hardware for Machine Learning

Lecture 24: End-to-End Deployment

Sophia Shao

Amdahl's Law

"The theoretical speedup of the latency of the execution of a program as a function of the number of processors executing it, …, is limited by the serial part of the program."



"In November 1961, IBM formed a group know as the SPREAD committee, which was chartered to set IBM's design goals for the next decade and to create an engineering and marketing plan to unify the efforts of the three IBM computer divisions. … After much engineering effort, on April 7, 1964, IBM announced the 360 architecture… The family name of 360 was chosen by IBM to say that it excelled at all "360 degrees of data processing."

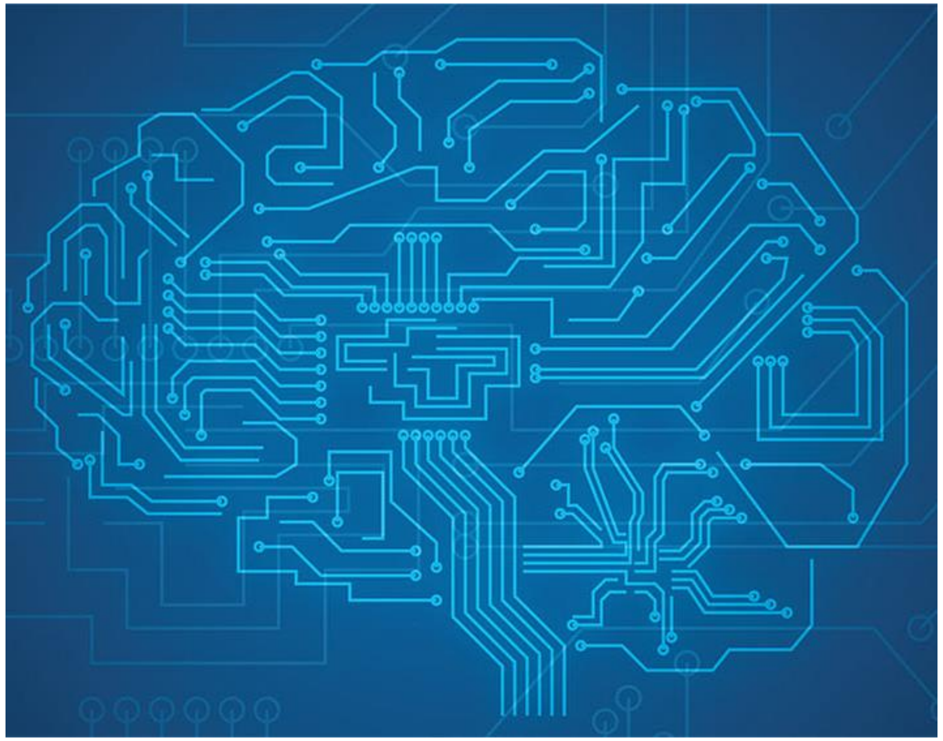
Readings in Computer Architecture

G. M. Amdahl
G. A. Blaauw
F. P. Brooks, Jr.,

Architecture of the IBM System/360

Review

- Core computation in DNN
- Execution order of the core computation
- Hardware realization of the core computation
- Mapping DNNs to hardware
- Data transfer mechanisms across storage hierarchy
- Sparsity in DNNs
- Codesign example
- Other Operators and Near-Data Processing
- Training Kernels
- Advanced Technology
- Accelerator-Level Parallelism
    - X-Level Parallelism
    - ALP in Mobile SoC
    - Open Challenges



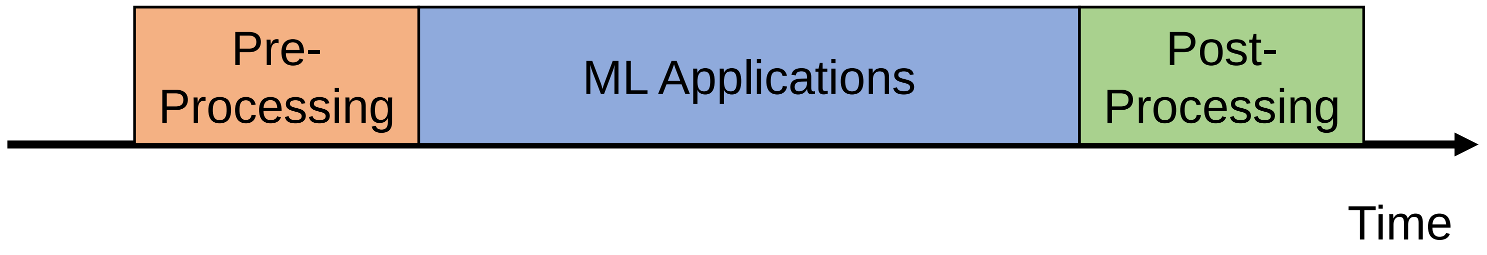
End-to-End Deployment

- AI Processing Pipeline
- Practical Constraints:
  - Thermal
  - Perf. Variability
  - Interference

Holistic View of the ML Processing Pipeline

- End-to-end ML pipeline is more than the ML algorithms.
  - ML algorithms also rely on pre- and post-processing code typically executed on CPUs
- For example:
  - "Data processing machines, aka 'readers', read the data from storage, process and condense them, and then send to the training machines."
  - "Training machines, aka 'trainers', solely focus on executing the training options rapidly and efficiently. "

Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective, HPCA'2018



Data Infeed for TPUs

- Data infeed: the process of preparing and moving input data to the TPU board.

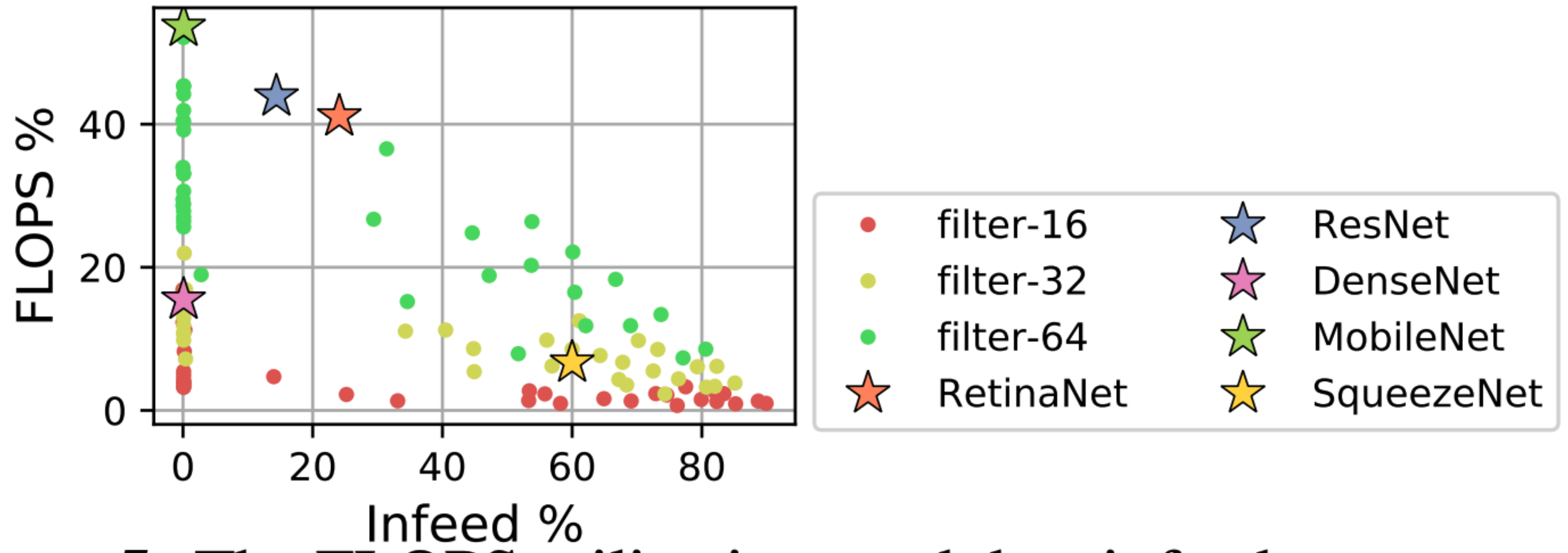
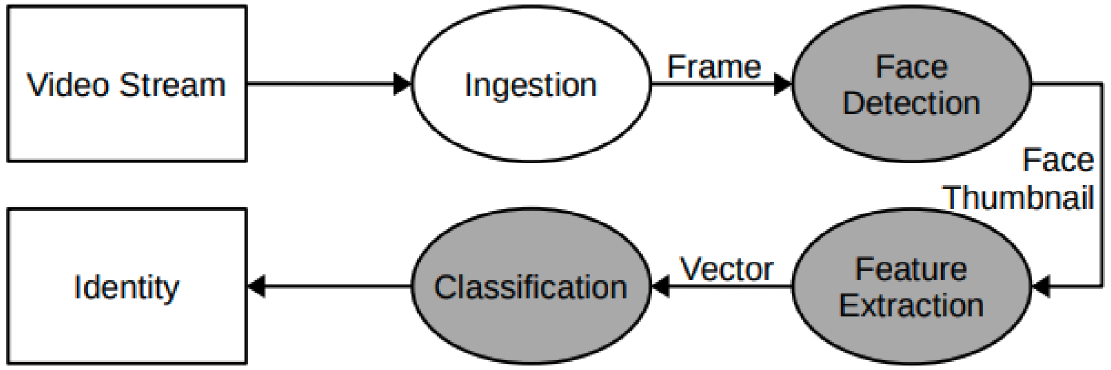


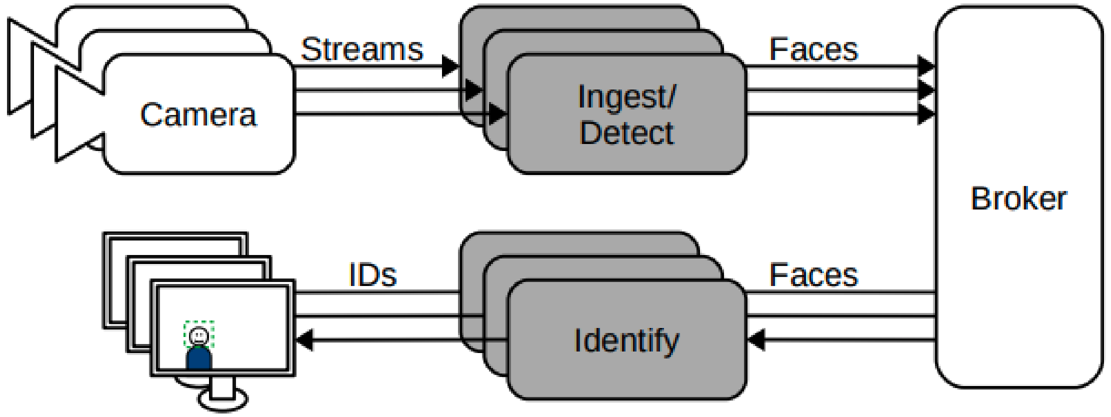
Figure 5: The FLOPS utilizations and data infeed percentages of ParaDnn (dots) and real-world (stars) CNNs.

A Systematic Methodology for Analysis of Deep Learning Hardware and Software Platforms, MLSys'2020

Example: A Face Recognition Pipeline



(a) Algorithmic flow of *Face Recognition*. A video stream enters *ingestion* for separation into individual frames. *Face detection* finds any faces within a frame and produces a thumbnail for each. *Feature extraction* generates identifying features for each face. Finally, *classification* finds a nearest match to known faces to produce an identity.



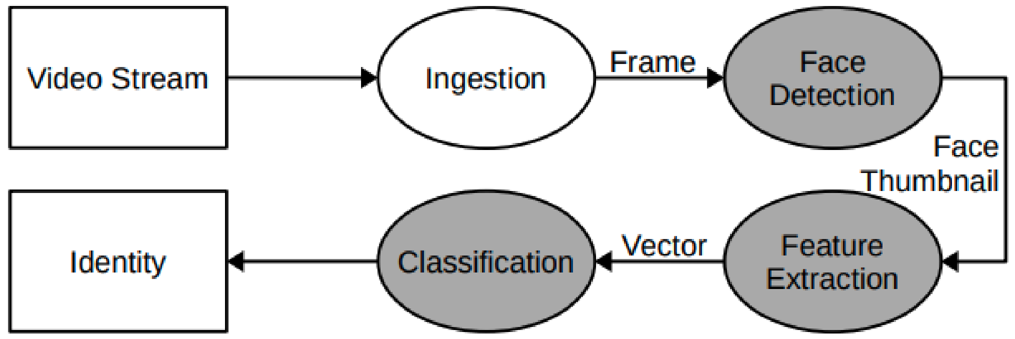
(b) Data center deployment of *Face Recognition*. Algorithms run as standalone processes in lightweight Docker containers deployed on separate nodes throughout the data center. *Ingestion* and *face detection* both run in the *ingest/detect* container while *feature extraction* and *classification* are combined in *identify*. Communication between steps within a container happens internally while communication between containers is coordinated through Apache Kafka *brokers*.

Figure 1: From the conceptual operation of *Face Recognition* to a deployment-ready implementation. Shaded blocks show AI stages. The straightforward application steps are wrapped in containers for deployment in a data center, requiring inputs, outputs, and communication coordination.

Missing the Forest for the Trees: End-to-End AI Application Performance in Edge Data Centers, HPCA'2020

Example: A Face Recognition Pipeline

- **Ingestion**: a pre-processing stage that ingests a video stream and parses it into individual frames.
- **Face detection (ML):** Detect any faces within a frame. It determines bounding boxes for and produces a thumbnail of each face in a frame.
- **Feature extraction (ML):** Produce a 128-byte vector of essential features that describe each face.
- **Classification (ML):** compares the feature vector against known face vectors to find the best match.

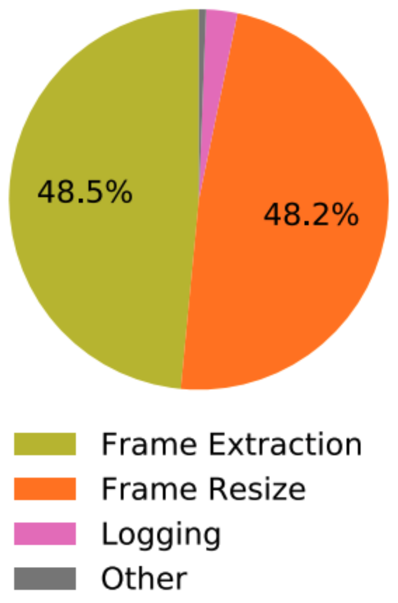


(a) Algorithmic flow of *Face Recognition*. A video stream enters *ingestion* for separation into individual frames. *Face detection* finds any faces within a frame and produces a thumbnail for each. *Feature extraction* generates identifying features for each face. Finally, *classification* finds a nearest match to known faces to produce an identity.

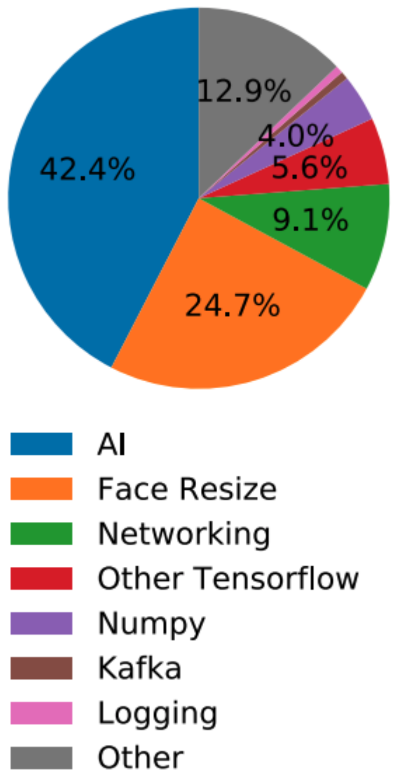
Missing the Forest for the Trees: End-to-End AI Application Performance in Edge Data Centers, HPCA'2020

Example: A Face Recognition Pipeline

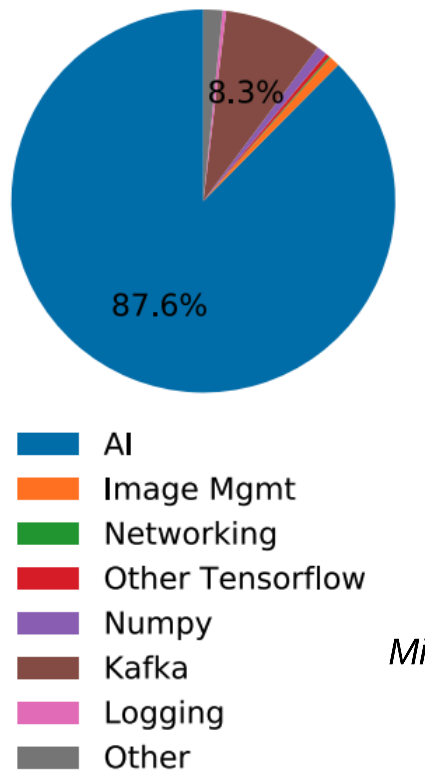
- CPU time breakdown for different components.
- With Amdahl’s law, 8X AI speedup -> 1.59X detection speedup.



(a) Ingestion.

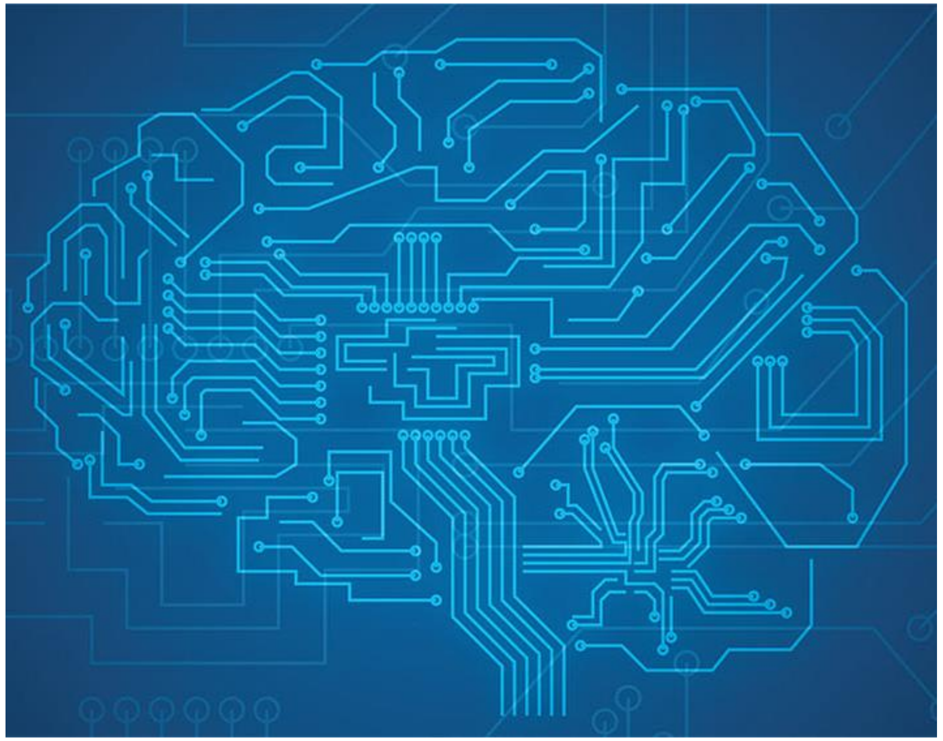


(b) Detection.



(c) Identification.

Missing the Forest for the Trees: End-to-End AI Application Performance in Edge Data Centers, HPCA’2020

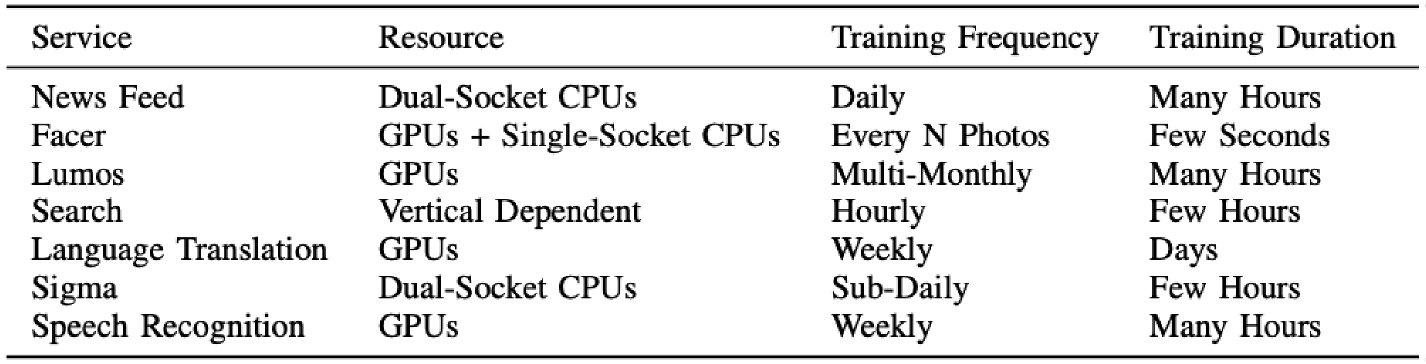


End-to-End Deployment

- AI Processing Pipeline
- Practical Constraints:
  - Thermal
  - Perf. Variability
  - Interference

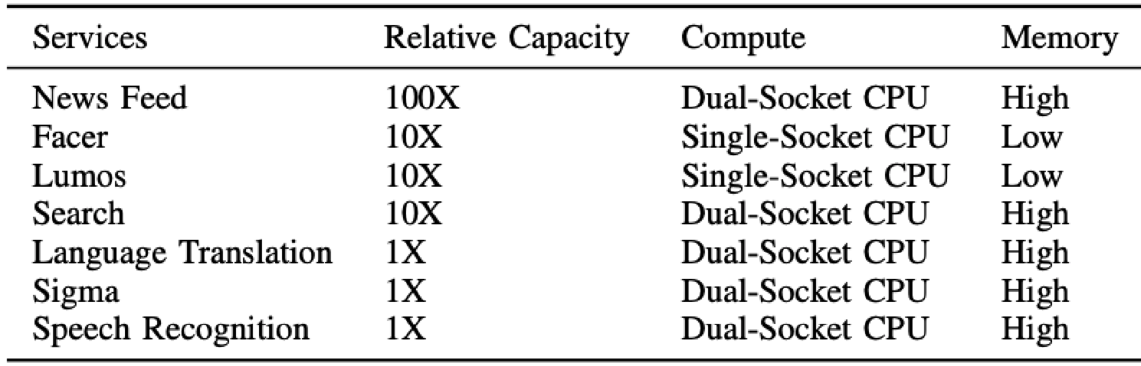
Practical Constraints in Deploying DL

- DL hardware design may seem easy after this course ☺
- Deploying DL hardware at scale requires in-depth understanding of the eco-system.
- CPU is still widely used for DL execution in a range of scenarios.



Service	Resource	Training Frequency	Training Duration
News Feed	Dual-Socket CPUs	Daily	Many Hours
Facer	GPUs + Single-Socket CPUs	Every N Photos	Few Seconds
Lumos	GPUs	Multi-Monthly	Many Hours
Search	Vertical Dependent	Hourly	Few Hours
Language Translation	GPUs	Weekly	Days
Sigma	Dual-Socket CPUs	Sub-Daily	Few Hours
Speech Recognition	GPUs	Weekly	Many Hours

TABLE II
FREQUENCY, DURATION, AND RESOURCES USED BY OFFLINE TRAINING FOR VARIOUS WORKLOADS



Services	Relative Capacity	Compute	Memory
News Feed	100X	Dual-Socket CPU	High
Facer	10X	Single-Socket CPU	Low
Lumos	10X	Single-Socket CPU	Low
Search	10X	Dual-Socket CPU	High
Language Translation	1X	Dual-Socket CPU	High
Sigma	1X	Dual-Socket CPU	High
Speech Recognition	1X	Dual-Socket CPU	High

TABLE III
RESOURCE REQUIREMENTS OF ONLINE INFERENCE WORKLOADS.

Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective, HPCA'2018

Practical Constraints in Deploying DL

- DL hardware design may seem easy after this course ☺
- Deploying DL hardware at scale requires in-depth understanding of the eco-system.
- CPU is still widely used for DL execution in a range of scenarios.

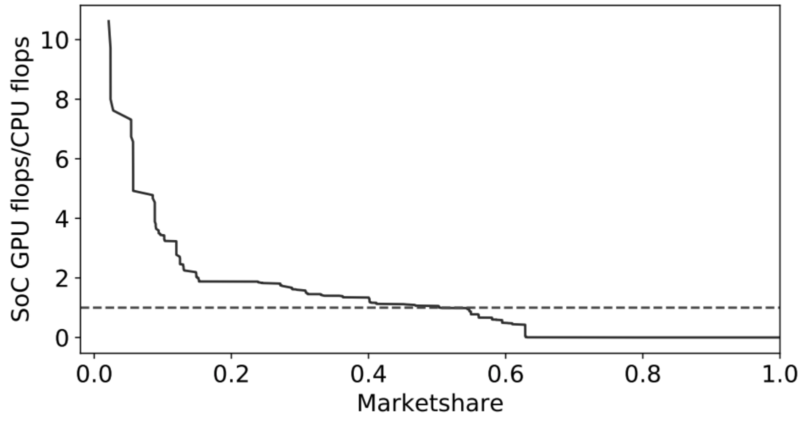
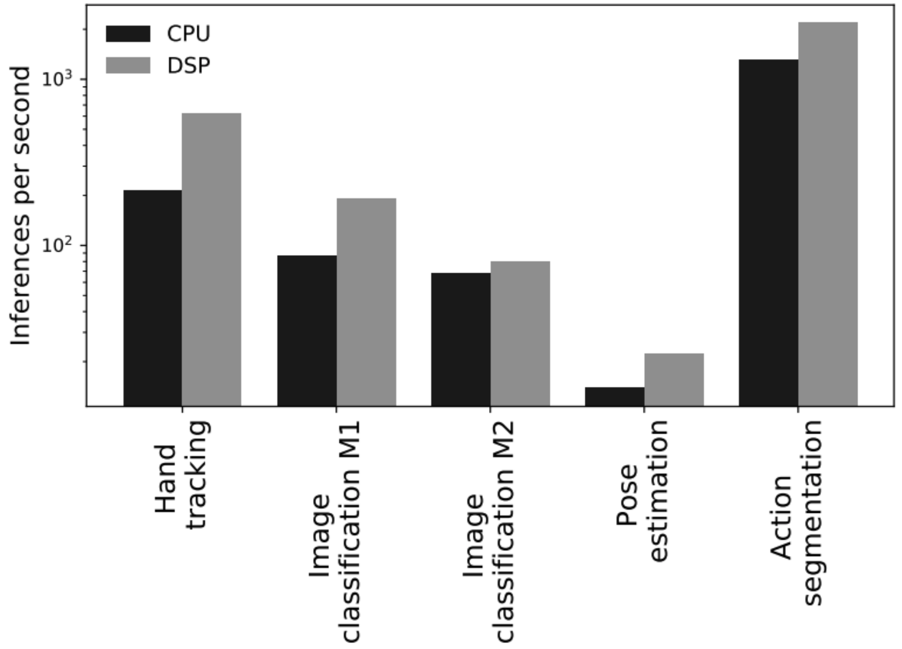


Figure 4: The theoretical peak performance difference between mobile CPUs and GPUs is narrow. In a median Android device, GPU provides only as much performance as its CPU. Only 11% of the smartphones have a GPU that is 3 times more performant than its CPU.



Machine Learning at Facebook: Understanding Inference at the Edge, HPCA'2019

Administrivia

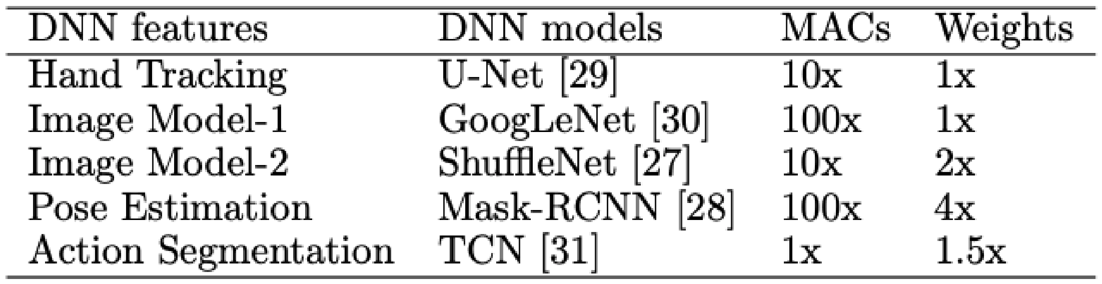
- Course Survey
  - https://course-evaluations.berkeley.edu/berkeley/
  - We VALUE your feedback!
  - Tell us your experience!
  - Tell us what worked and what could be improved!
  - Extra credits:
    - 1pt for everyone if we hit 70% response rate!
    - 1pt if you submit a confirmation screenshot (private post on Piazza)!

Administrivia

- Checkpoint 2 this week:
  - Looking forward to your great progress!
- Project Presentation:
  - Tentative: 5/3
  - 15 mins presentation + 5 mins Q&A
- Project Report:
  - Max 8 pages
  - 2-column format. Use Latex (e.g., with overleaf)
  - Provide a link to your code repository.

Thermal Constraints

- Running multiple DNN inferences for Oculus Virtual Reality.
- High performance requirements:
  - Real-time w/ 30-6- frames per second.
  - Multiple cameras to cover a wide field of view
  - On-device processing
- E.g., Qualcomm Snapdragon SoC, with
  - The CPU model utilizes a big.LITTLE core cluster with 4 Cortex-A73 and 4 Cortex-A53 and a Hexagon DSP.



DNN features	DNN models	MACs	Weights
Hand Tracking	U-Net [29]	10x	1x
Image Model-1	GoogLeNet [30]	100x	1x
Image Model-2	ShuffleNet [27]	10x	2x
Pose Estimation	Mask-RCNN [28]	100x	4x
Action Segmentation	TCN [31]	1x	1.5x

Table 1: DNN-powered features for Oculus.

Machine Learning at Facebook: Understanding Inference at the Edge, HPCA'2019

Thermal Constraints

- The thermal throttling has a significant effect on performance, degrading the FPS performance to 10 frames-per-second.
- Despite higher performance, lower power consumption and operating temperature, the DSP implementation comes with significantly higher programming overhead.

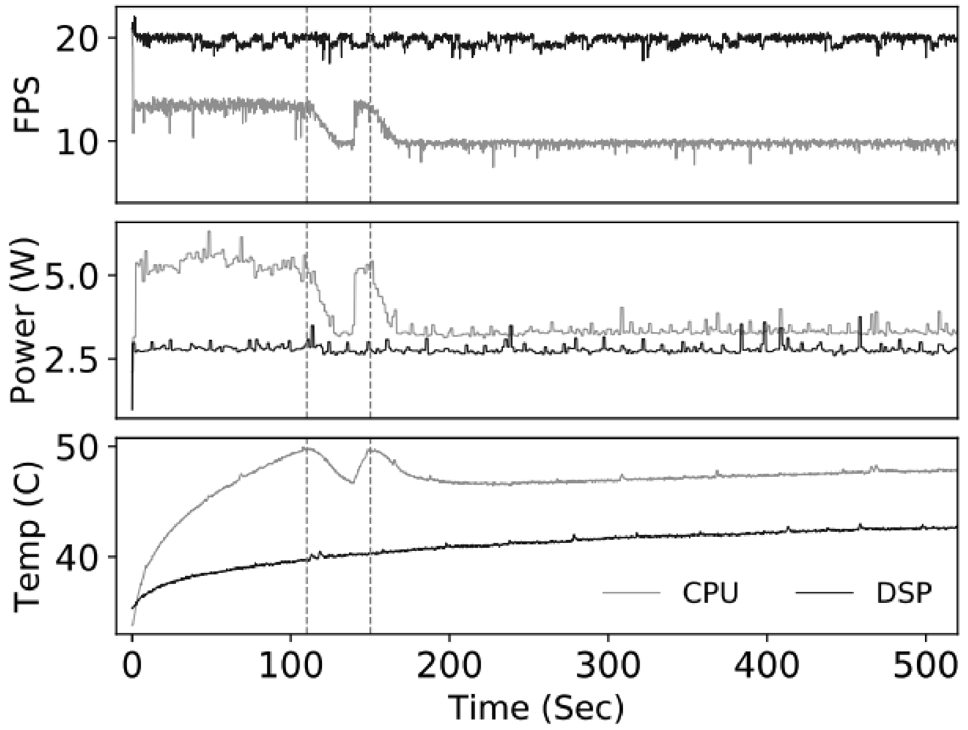


Figure 9: Inference frames-per-second performance, power, and temperature comparison for a vision model for Oculus. Thermal throttling (marked with dotted lines) prevents CPU from operating at an acceptable FPS performance level.

Machine Learning at Facebook: Understanding Inference at the Edge, HPCA'2019

Performance Variability

- Challenging to make guarantees on quality of service.
- Likely due to system activities in deployed smartphones and the environment the smartphones are in.

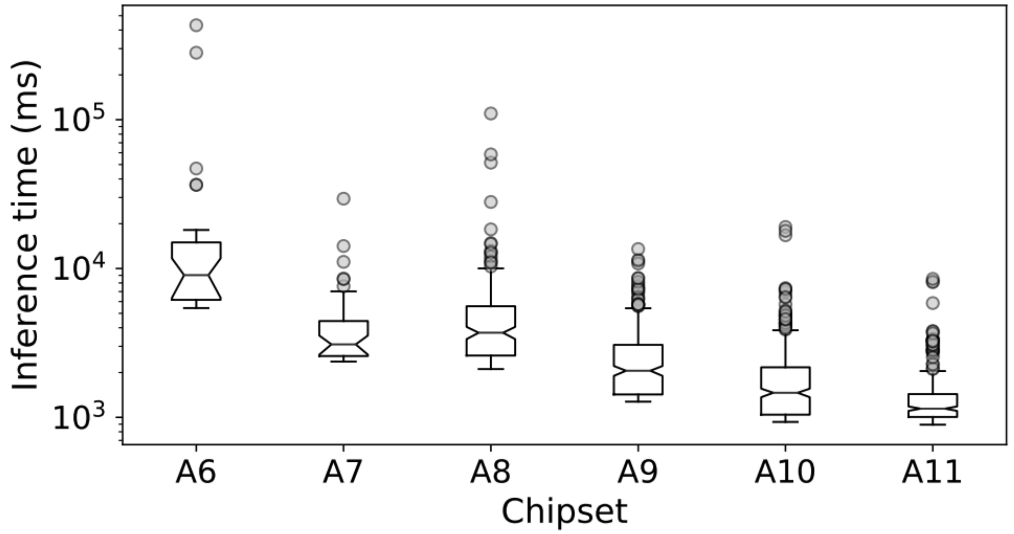


Figure 10: The inference time performance improves over generations of iPhones. However, within each generation, significant inference performance variability is observed with a large number of outliers.

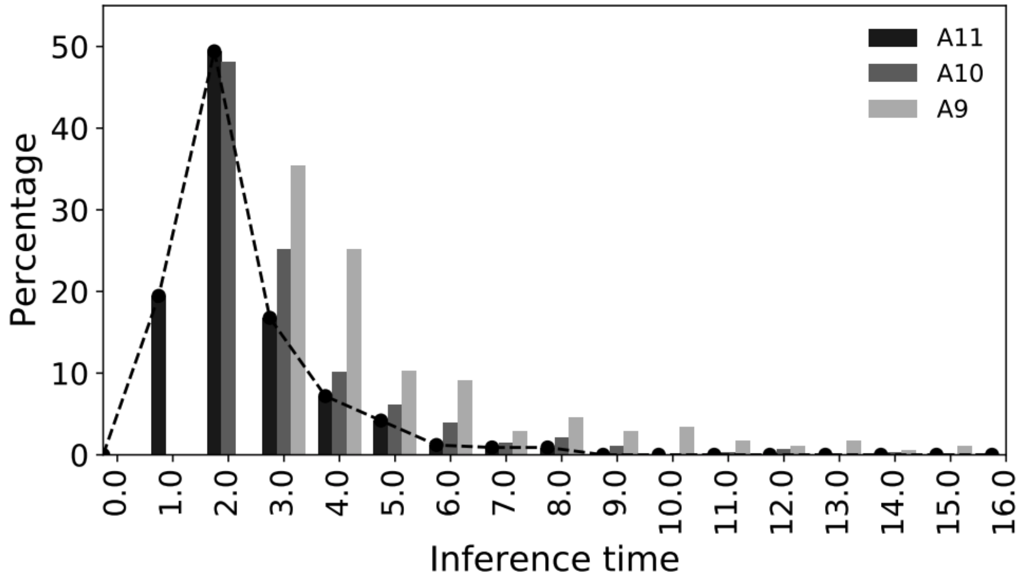


Figure 11: The inference time follows an approximate Gaussian distribution with the mean centered at 2.02ms and the standard deviation of 1.92ms.

Machine Learning at Facebook: Understanding Inference at the Edge, HPCA'2019

Interference

- Contention for shared resources also applies to accelerators, especially in a datacenter environment.

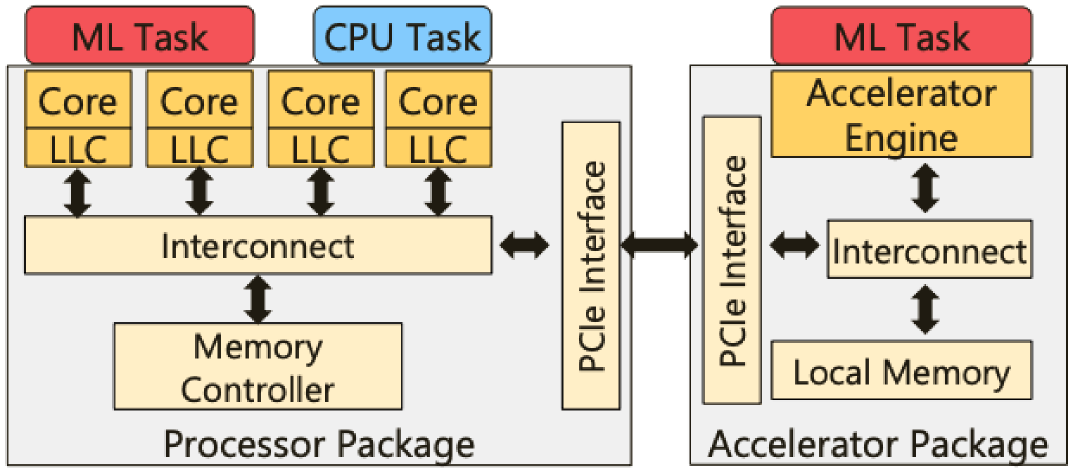


Figure 4: Architecture of an accelerated platform.

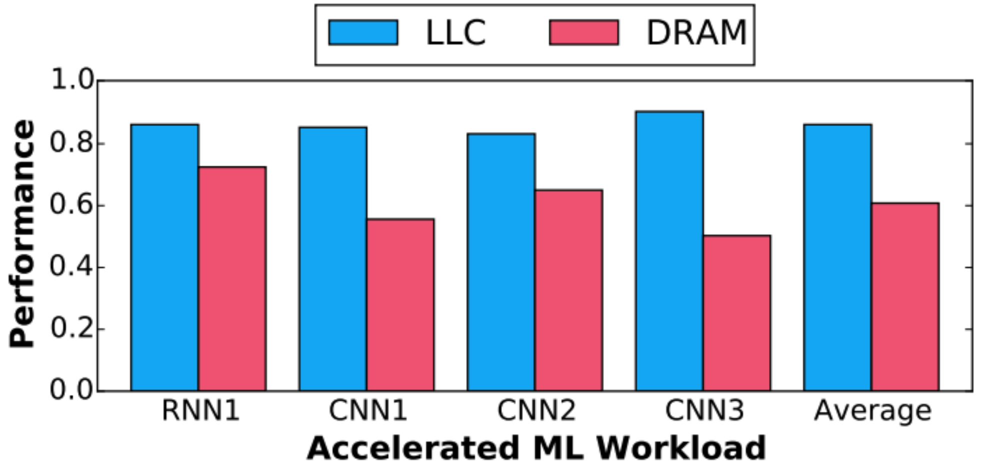


Figure 5: Workload sensitivity to shared resource interference. Performance is normalized to no interference.

Kelp: QoS for Accelerated Machine Learning Systems, HPCA'2019

Review

- Core computation in DNN
- Execution order of the core computation
- Hardware realization of the core computation
- Mapping DNNs to hardware
- Data transfer mechanisms across storage hierarchy
- Sparsity in DNNs
- Codesign example
- Other Operators and Near-Data Processing
- Training Kernels
- Advanced Technology
- Accelerator-Level Parallelism
- This lecture: end-to-end deployment