CS 6423
Scalable Computing for Data Analytics

Lecture 11: Google Cloud Computing

Prof. Gregory Provan
Department of Computer Science
University College Cork

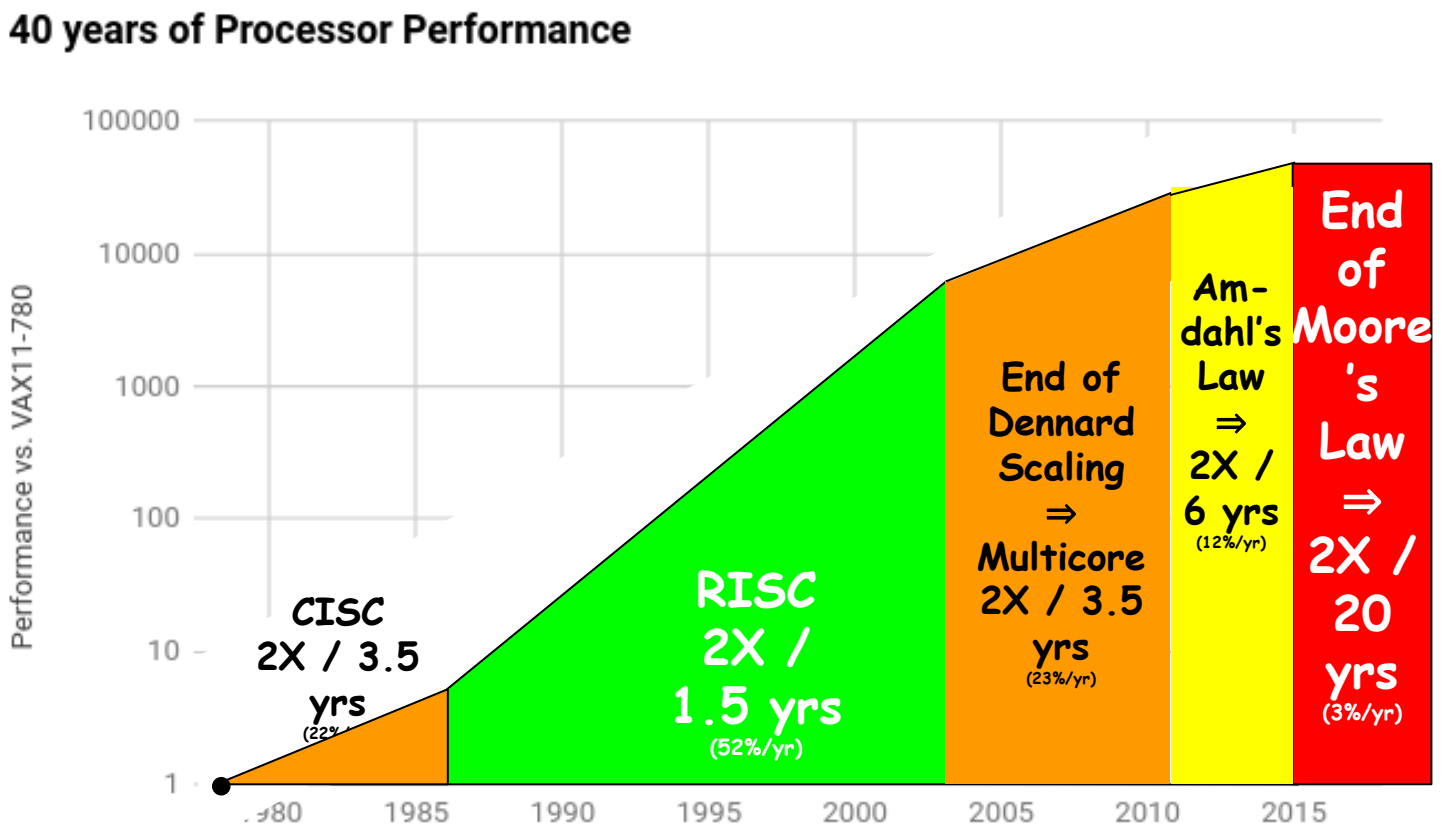
Based on: David Patterson (UC Berkeley) lecture material

Lecture Objectives

Describe Google Cloud TPU
Distributed Algorithms and Architectures

- Google Cloud TPU
  - New cloud architecture
  - Best for machine learning applications
- Secret internal details
- We focus on the programming methodology
  - Computation graphs

Motivation:
End of Growth of Performance?



Based on SPECintCPU. Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e. 2018

A New Cloud Architecture Needed

Since

- Transistors not getting much better
- Power budget not getting much higher
- Already switched from 1 inefficient processor/chip to N efficient processors/chip

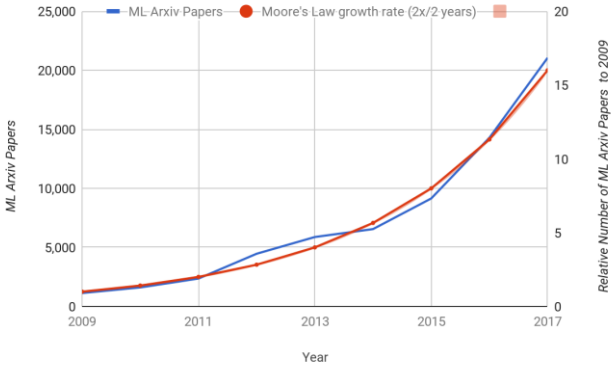
Approach: *Domain Specific Architectures*

- Just do a few tasks, but extremely well

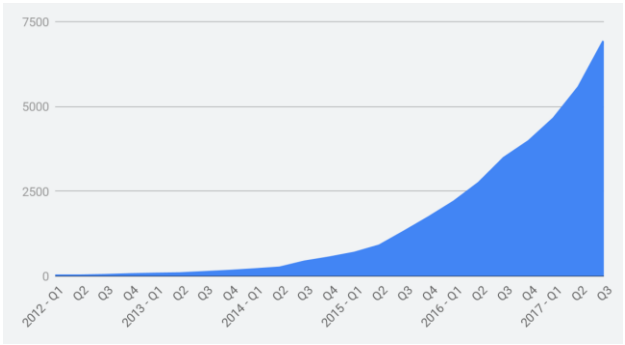
TPU Cloud Programming

- Google Cloud Platform
  - Based on Application Specific Integrated Circuit (ASIC) architecture
- Tensor Flow Research cloud
  - Deep learning
  - Application example: driverless car initiatives that require instant inferences/responses with hard deadlines?
- Focused on growth of deep learning

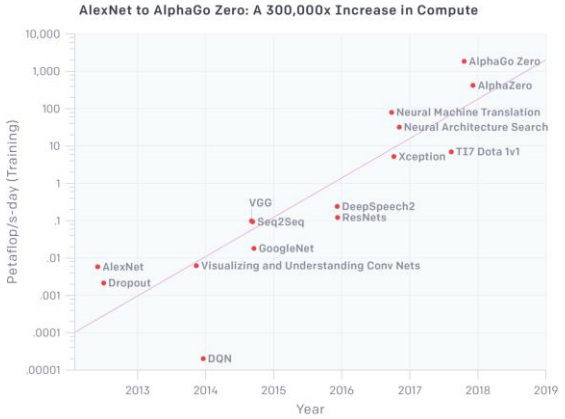
Exponential Growth in Deep Learning



ArXiv papers about ML
~18 months



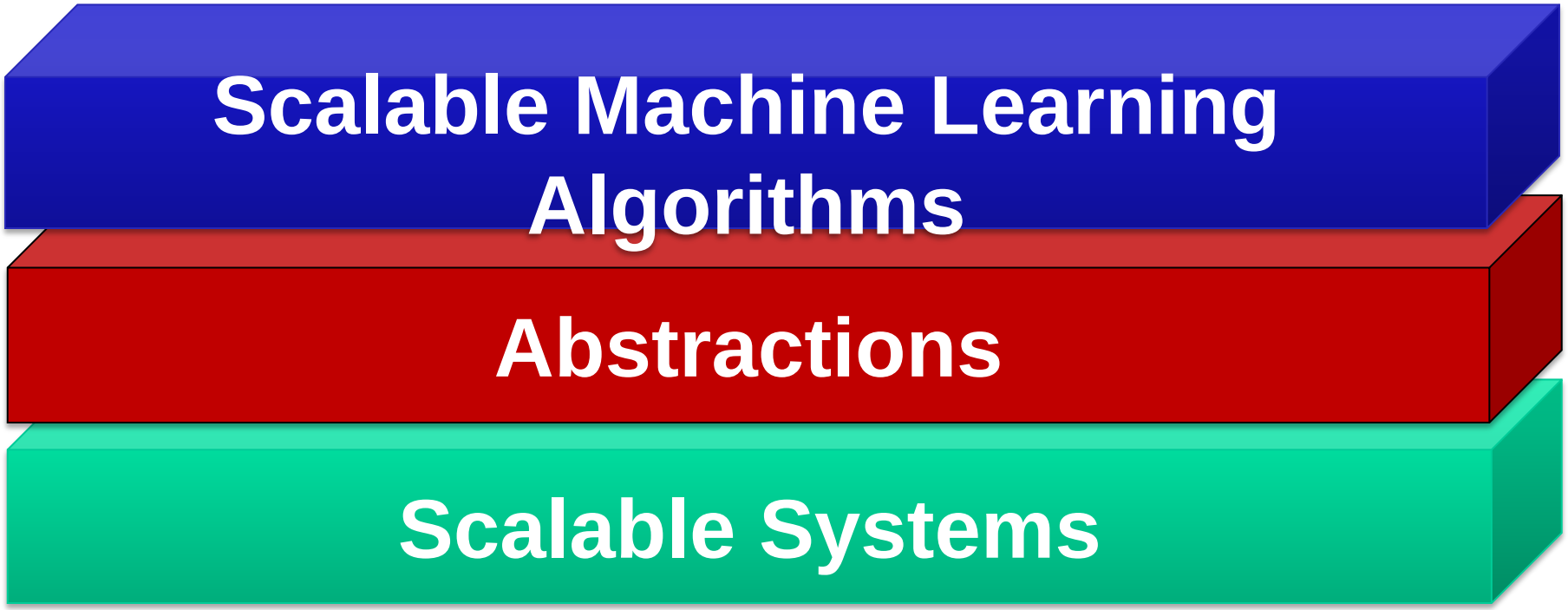
Google project
directories
~18 months



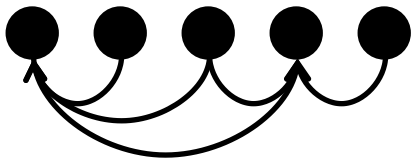
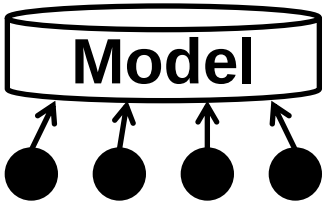
FLOPs to train a model
~3.5 months
(~10X per year)

Why all this growth? Because Deep Learning works.

ML Systems


Scalable Machine Learning Algorithms
Abstractions
Scalable Systems

ML Systems Landscape

Dataflow Systems	Graph Systems	Shared Memory Systems
		
Naïve Bayes, Rocchio	Graph Algorithms, Graphical Models, Computation Graph	SGD, Sampling
Hadoop, Spark	GraphLab, Tensorflow	Bosen, DMTK, ParameterServer.org

MapReduce and Learning

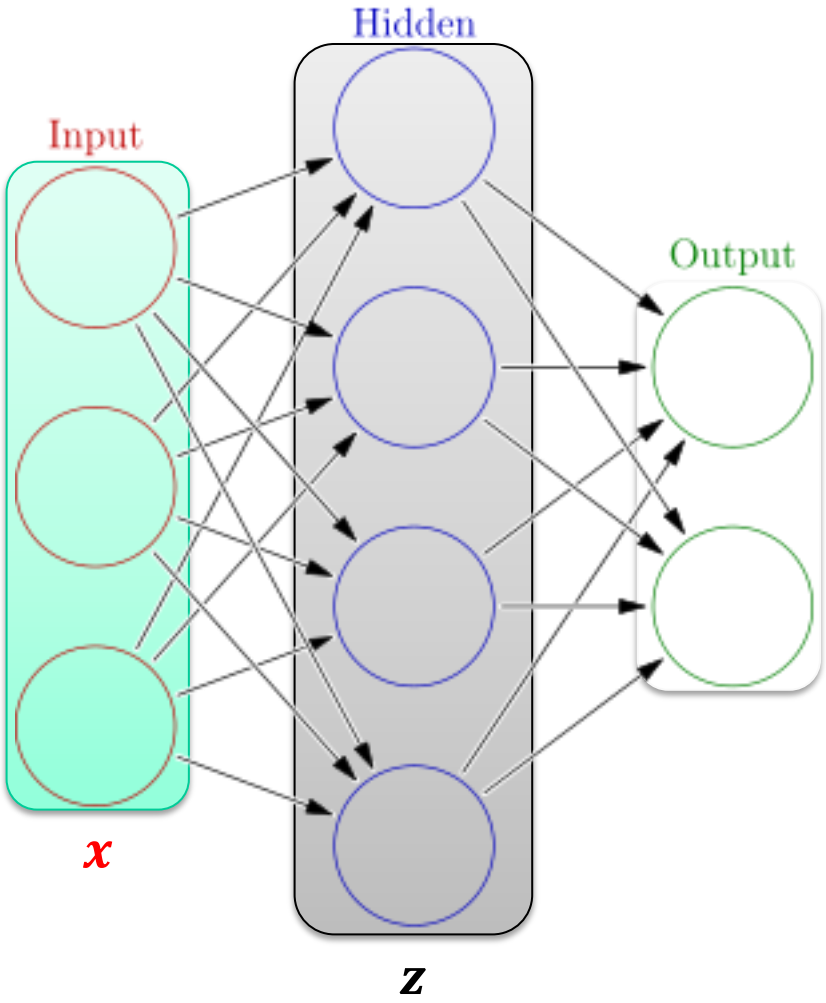
- MapReduce
  - Requires inference at every Map/Reduce node to be the same
- Machine learning
  - Does this paradigm work in general?

Naïve Bayes and MapReduce

- Structure of NB model is fixed a priori
- Can transform NB learning to a counting problem
  - Count attributes of training cases
  - Shuffle to appropriate conditional probability tables

- Is MapReduce suitable for other learning models?

Neural Network: Definition



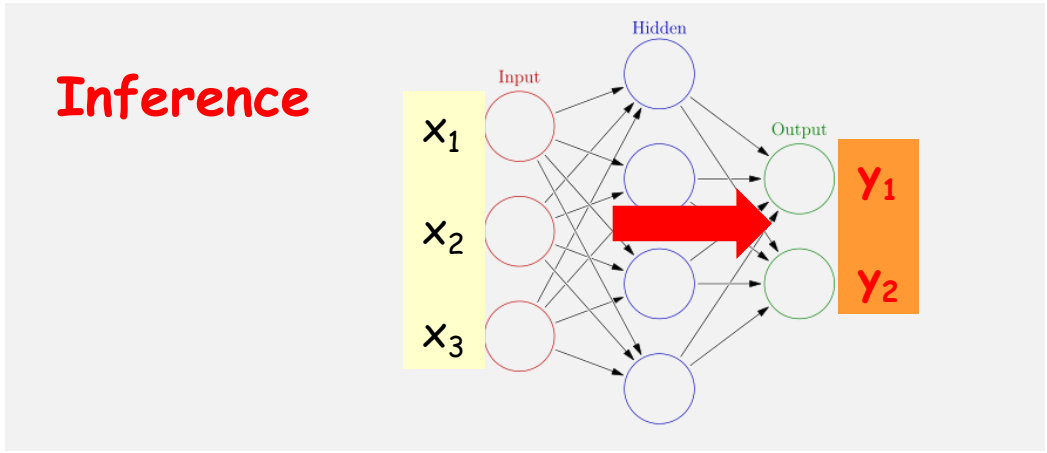
Weights

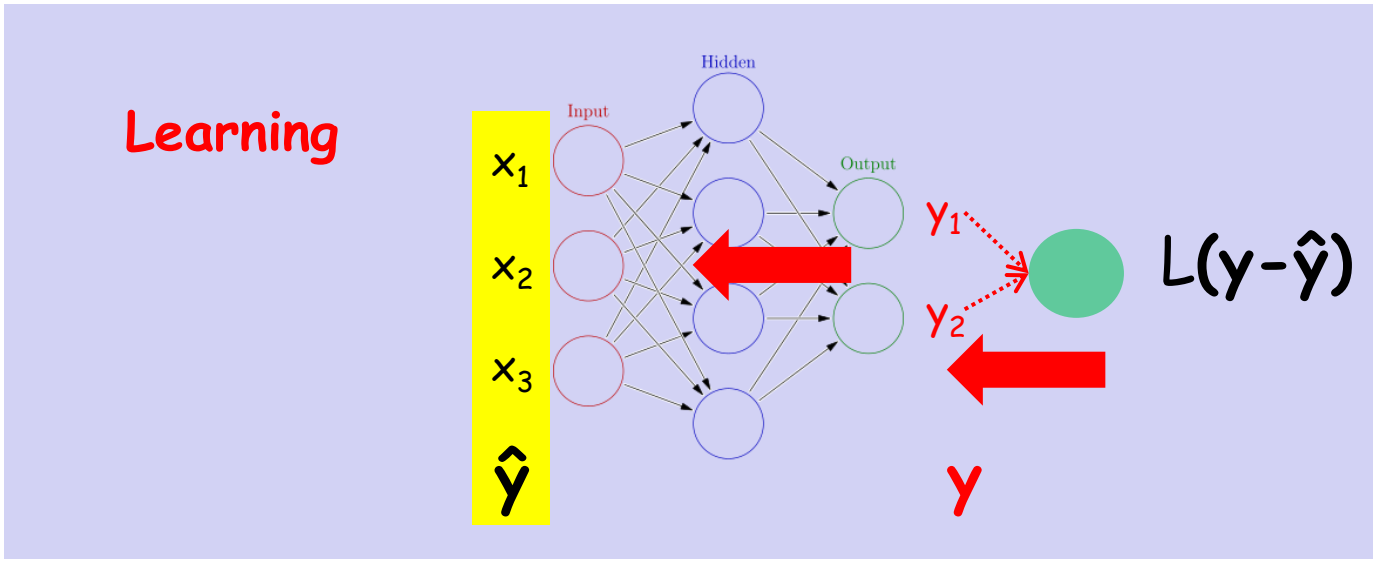
$$z = f(\mathbf{W}_1 x + b_1)$$

$$y = g(W_2 z + b_2)$$

Activation functions

Neural Network Operations



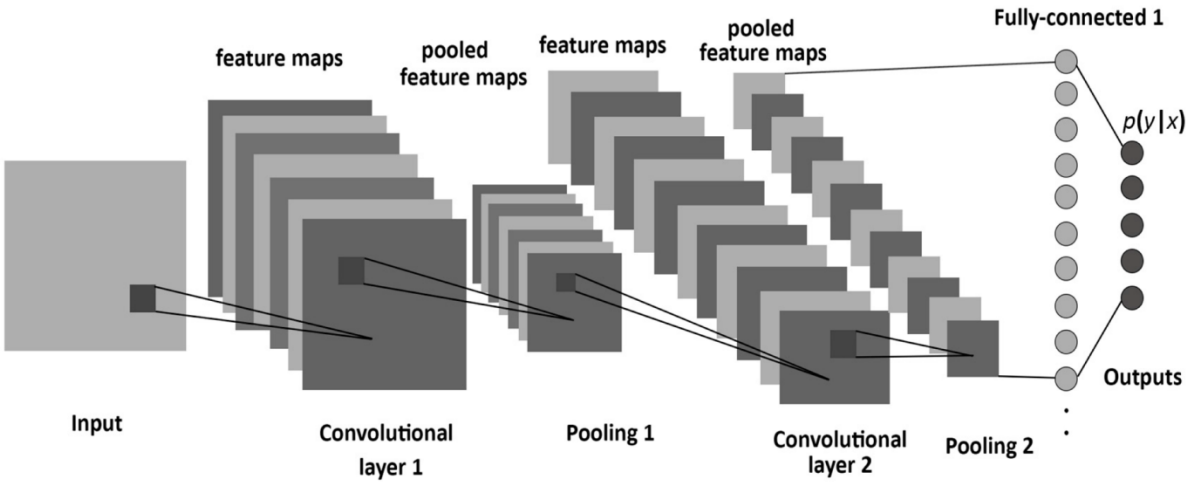


NN Operations

- Inference (classification)
  - Forward direction
  - data
    - $z=Wx+b$; $y=\sigma(z)$
- Training (learning updated weights)
  - Reverse direction
  - Data
    - From $L(y-\hat{y})$: $\delta L/\delta y$
    - Want to compute $\delta W/\delta L$

Generalisaton: Deep Network

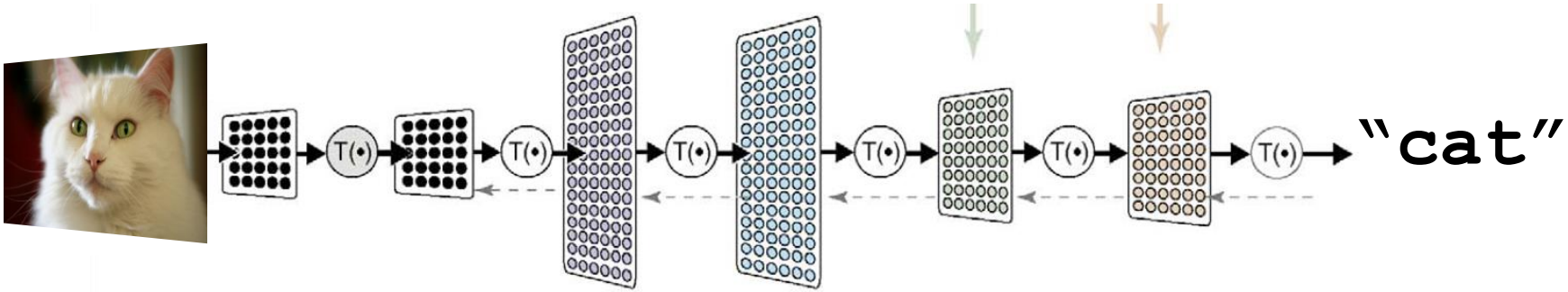
- Structural view of network
  - Different inference at each level



MapReduce Approach

- Does not work for NN operations
- Need a custom-tailored computational framework
- Deep networks
  - Multiple layers
  - Inference different at each level
  - Must allow forward/backward inference
    - Different data being passed

Observation: Training >> Inference



3x the computation: forward propagation, backward propagation, and weight update.

Much longer data storage lifetimes: **memory** capacity and bandwidth.

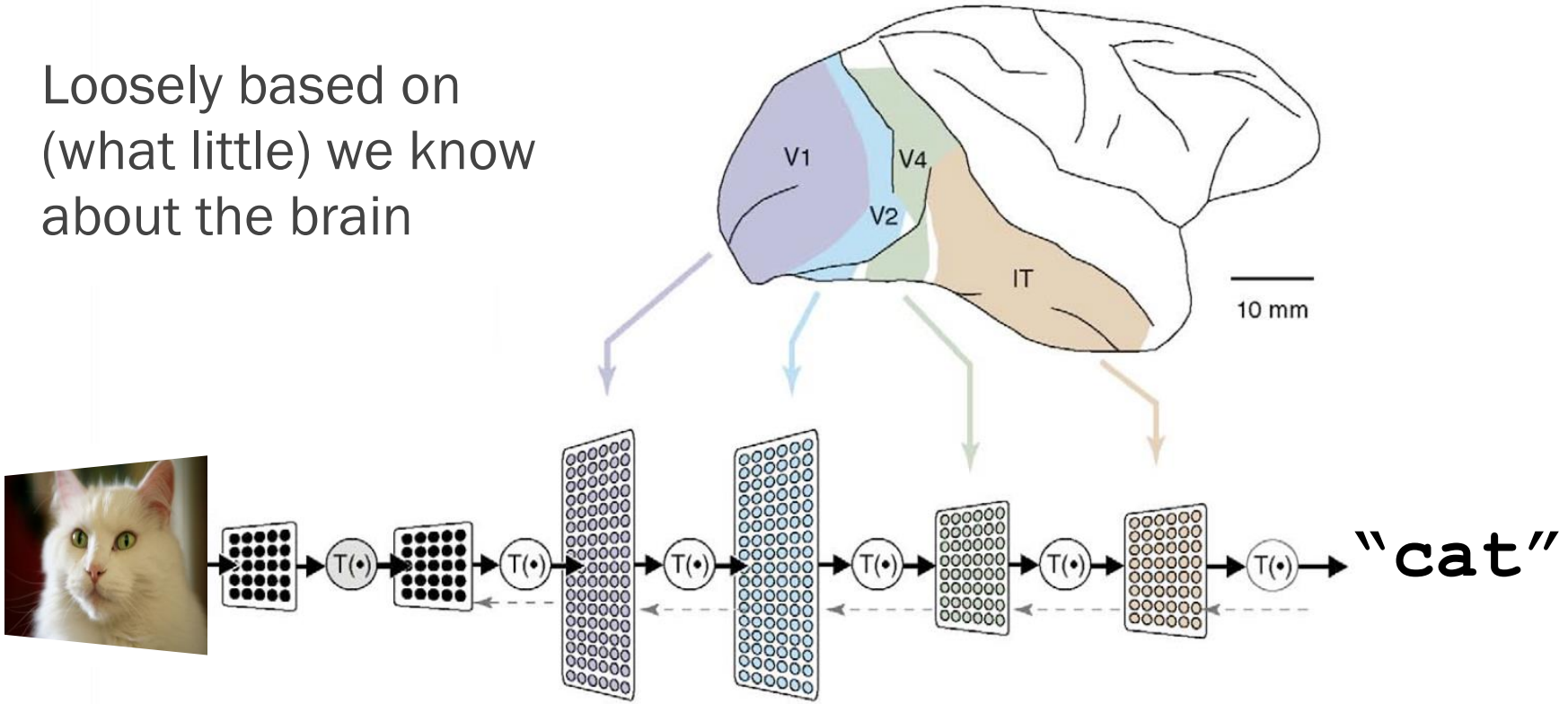
Huge **training datasets** for training, versus scale-out to serve inference.

Changes to algorithms and model structure require more **flexibility**.

Many more potential **Amdahl's Law** bottlenecks.

Motivation: Deep Learning

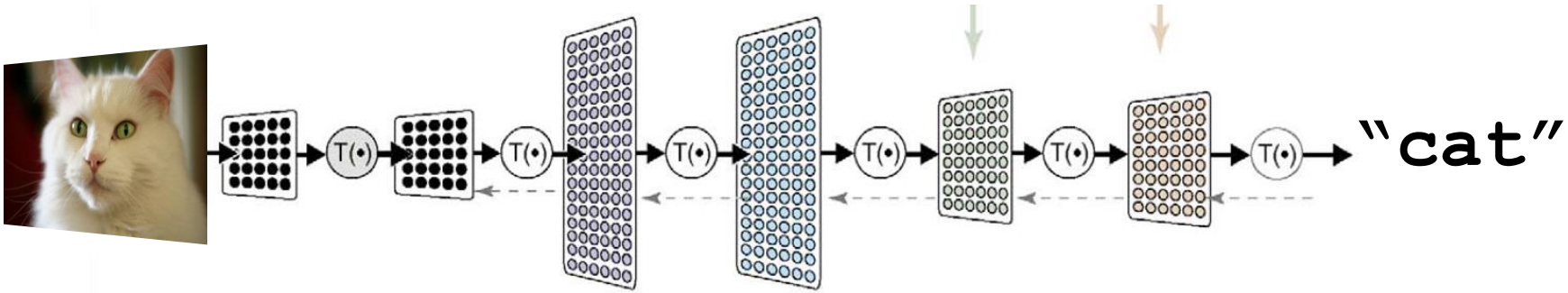
- Loosely based on (what little) we know about the brain



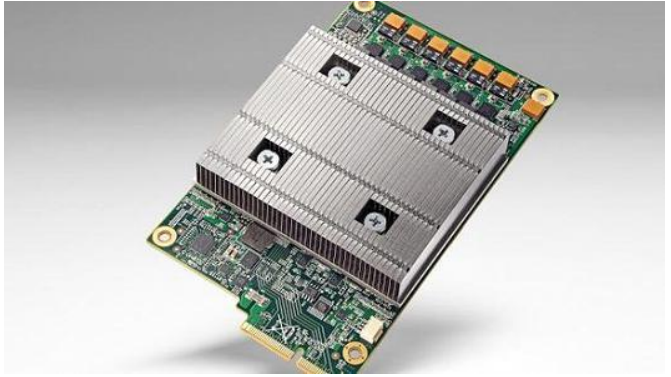
Slide from "Large-Scale Deep Learning with TensorFlow for Building Intelligent Systems," by Jeff Dean, ACM Webinar, 7/7/16

Computation Graph

- Devised for deep learning
- Applicable to many other types of machine learning
- Basic framework
  - Computation graph
  - Captures structure of inference in a deep network



Cloud-TPU Overview



- The Tensor flow processor
  - ASIC (Application Specific Integrated Circuit)
  - optimized for machine learning (Ml) and artificial intelligence (Al) algorithms and operations.
- Four of these TPUs powering a cloud service can deliver 180 Tflops for ML tasks.
- TPU network is the engine/computation power behind the TPU-based tensor flow cloud services for
  - training (data) and testing and
  - inference

Cloud-TPU Approach

- *Objective*: harness the rising interest in ML to drive use of Google's cloud services.
- Google Compute services
  - use the TPU for research and commercial purposes.
  - Provides instant inference and real-time response to events if the analysis is powered by powerful high speed processors.
  - Useful method for processing cloud-based data
- TPU design shrouded in secrecy
- TensorFlow (open-source) framework
  - software interface to the chip
- The Cloud TPU
  - supports floating-point math
  - Google uses for both "training" and "inference" phases of ML.

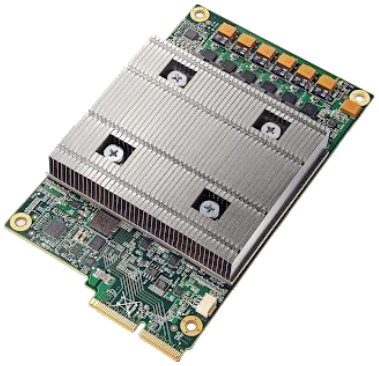
History

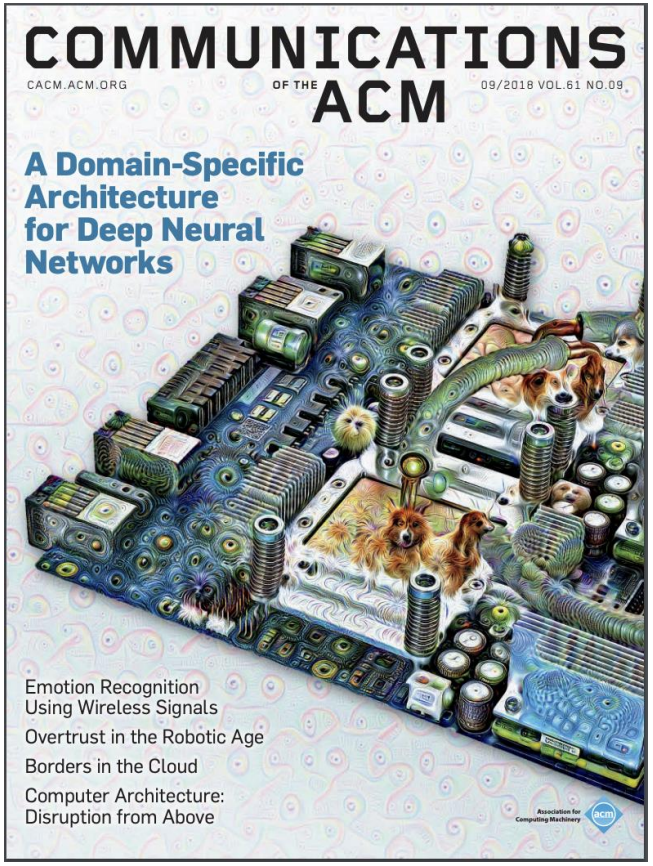
- 2013: Prepare for success-disaster of new DNN apps
  - Scenario with users speaking to phones 3 minutes per day:
    If only CPUs, need 2X-3X times whole fleet
  - Unlike some hardware targets, DNNs applicable to a wide range of problems, so can reuse for solutions in speech, vision, language, translation, search ranking, …
- Custom hardware to reduce the TCO of DNN inference phase by 10X vs. CPUs
  - Must run existing apps developed for CPUs and GPUs
- A very short development cycle
  - Started project 2014, running in datacenter 15 months later:
    Architecture invention, compiler invention, hardware design, build, test, deploy
- Google CEO Sundar Pichai reveals Tensor Processing Unit at Google I/O on May 18, 2016 as "10X performance/Watt"

cloudplatform.googleblog.com/2016/05/Google-supercharges-machine-learning-tasks-with-custom-chip.html

Tensor Processing Unit v1 (deployed 2015)

Google-designed chip for neural net **inference**



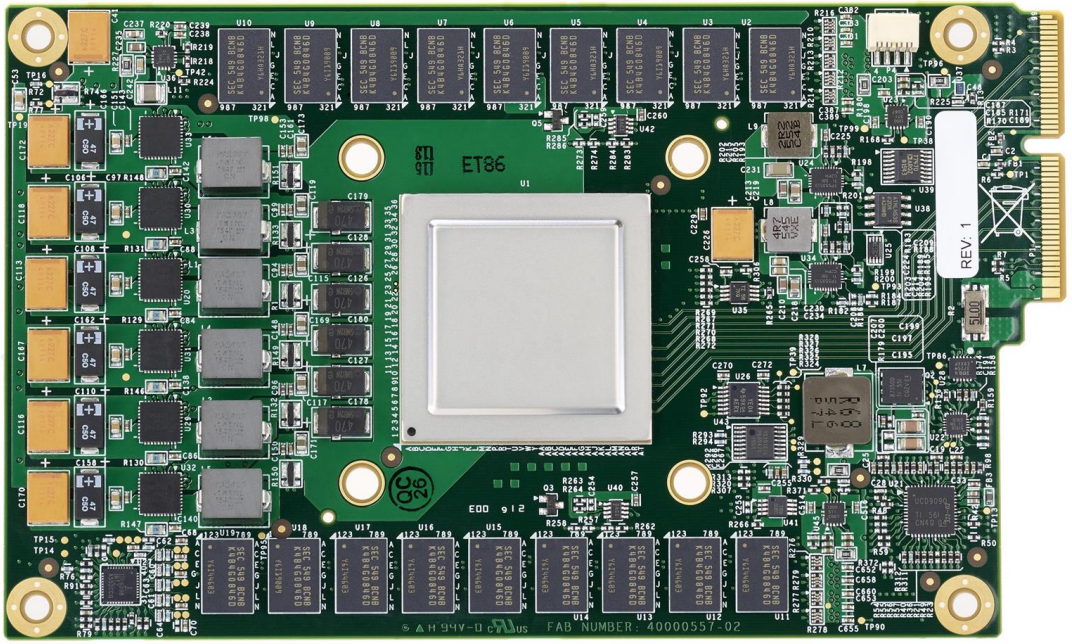


In production use for ≈4 years: used by billions on search queries, for neural machine translation, for AlphaGo match, …

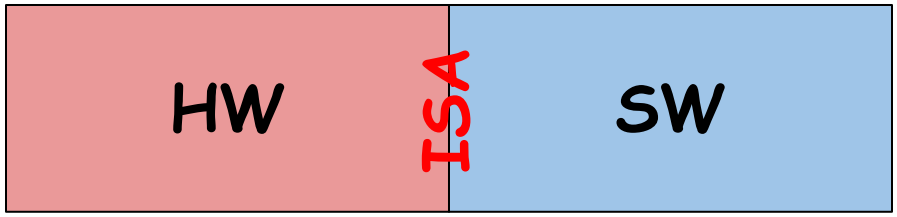
A Domain-Specific Architecture for Deep Neural Networks, Jouppi, **Young, Patil, & Patterson**, *Communications of the ACM*, September 2018

TPUv1 Card & Package

- TPUv1 Card to replace a disk
- Up to 4 cards / server



Classic Codesign at the HW/SW Interface



Definition: design spanning two fields for a common goal.

- Classic version is between architecture and compiler.

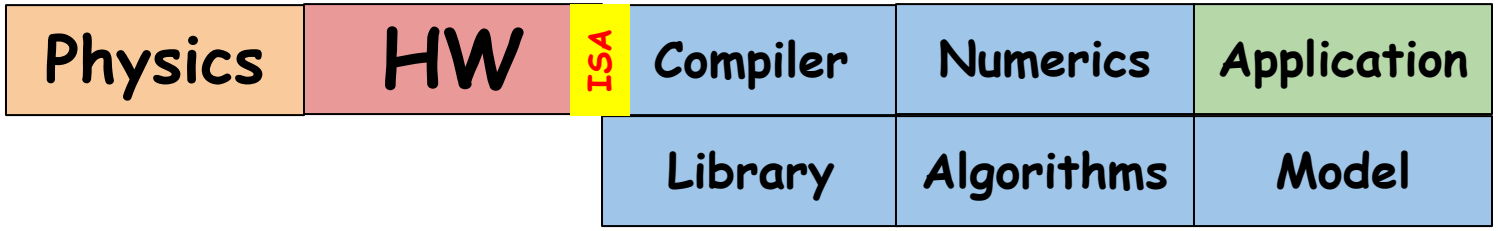
Instruction Set Architecture (ISA) as interface/contract between levels.

Example of pushing things back and forth: instruction scheduling.

- VLIW (static scheduling)
- OoO (dynamic scheduling)
- Answer today=both.

Ultimately ISA is a **single** thin layer between the hardware and software domains.

Codesign for Domain-Specific Architectures



(conceptual, not rigorous diagram)

Now, there are **many** different layers, with **many** different interfaces.

TPUs are still digital (for now).

- Some startups are pushing into physics (NVRAM, Flash, optical).
- Need to do codesign from physics to application: hard!

Fallacy: TPUs are ASICs, so they are not Programmable

ASIC: **A**pplication-**S**pecific **I**ntegrated **C**ircuit

- Means only "build whatever you want into the chip."
- ASICs include general-purpose cores, SoCs, and fixed-function designs.

TPUs are Domain-Specific Architectures (DSAs) for Machine Learning.

- Designed to meet our current and future needs.
- They include the flexibility to handle future models.
- Choosing the **right** amount of flexibility is central to codesign process.

For the technically nitpicky:

- TPUv1 is a coprocessor, controlled by the host.
- TPUv2 and successors are Turing-complete.
- TPUs power both Google **research** and Google **production** applications.

Training: DNN Supercomputer or Cluster of CPUs with DNN Accelerators?

- Single-chip system—built as coprocessor to a CPU like TPUv1—would work fine for inference and standard cluster networks
  - AlphaGo used cluster of 64 TPUv1 chips
- Went instead with large supercomputer because
  - Training takes weeks to months on single chip for our production training runs
  - Deep neural network wisdom was bigger datasets + bigger machines led to breakthroughs
- Build a NN **supercomputer** (TPU v2/v3) vs build a NN **coprocessor chip** (TPU v1)

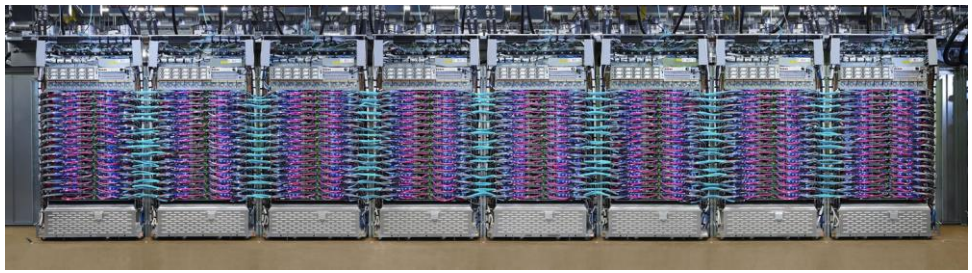
Datacenter-scale Supercomputer

Three ways to spend on computer architectural resources:

- Compute
- Memory
- Interconnect

Cloud TPUs were designed from the beginning to be networked.

- Data parallelism through increased batch size scales seamlessly today.
- Model parallelism is underway.



Actual Deployment

- Architecture
  - 4 TPU on a custom accelerator board and
  - 64 of these boards are packed on a two dimensional torus network in a cluster called a “pod” that is capable of up to 11.5 petaflops.
  - Of course, lots of heat sinks…
- Performance
  - 8 TPUs completed in 6 hours what 32 high GPU took a full day to complete (Training task).
  - In 2015 using Google’s Ml jobs, this TPU ran 15-30 times faster and delivered 30-80 times better performance per watt than Intel’s Haswell server CPU and Nvidia’s K80 GPU.

Cloud TPU Architecture

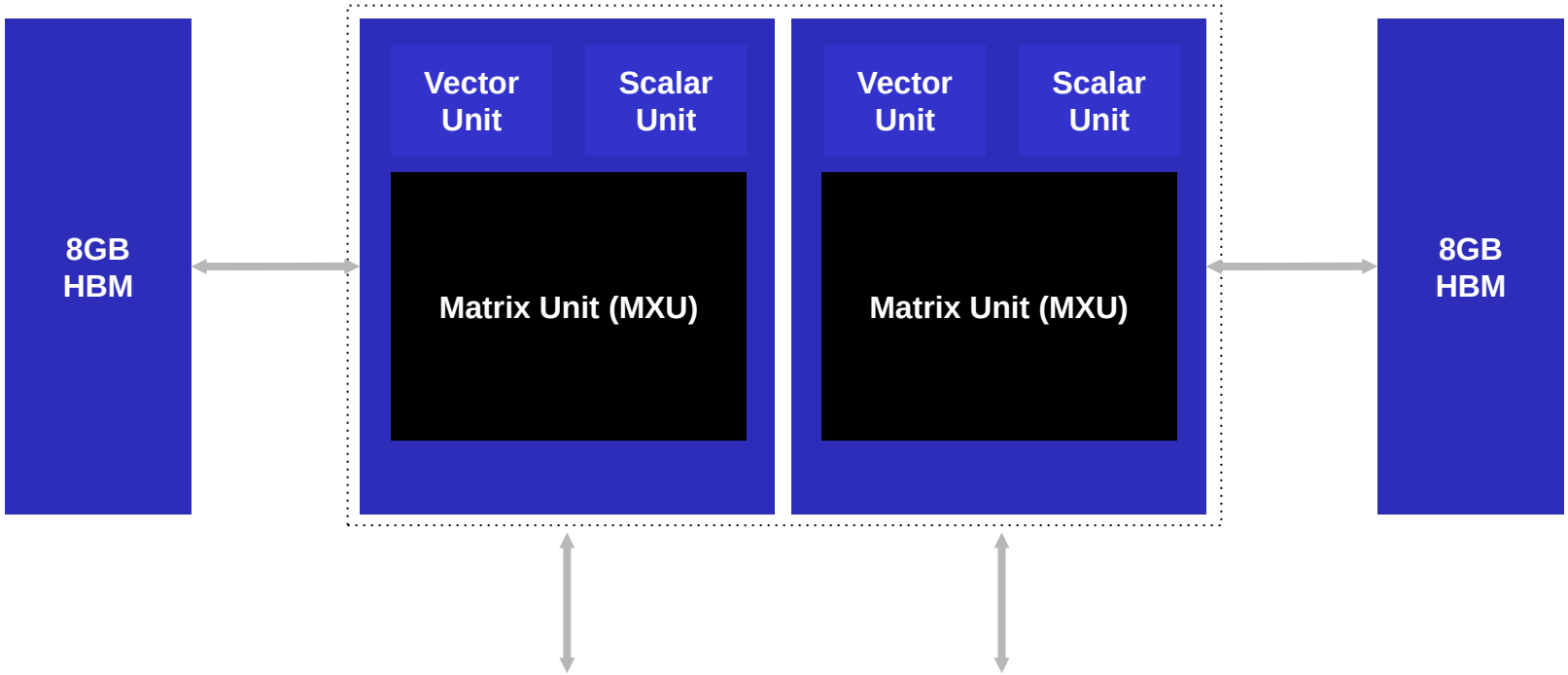
- Google provide very little details of the ASIC and the support systems.
- Earlier version offered only inference: supporting only quantized integer computation.
- Latest version supports training and so requires floating point operations.
- Super-fast memory bandwidth, significant on-chip memory,.
- Things are happening on the board and not on the rack.
- The Cloud TPU supports one large, one medium and 8 small heat sinks.
- Instructions:
  - linear algebra instructions, core matrix multiplication instructions.

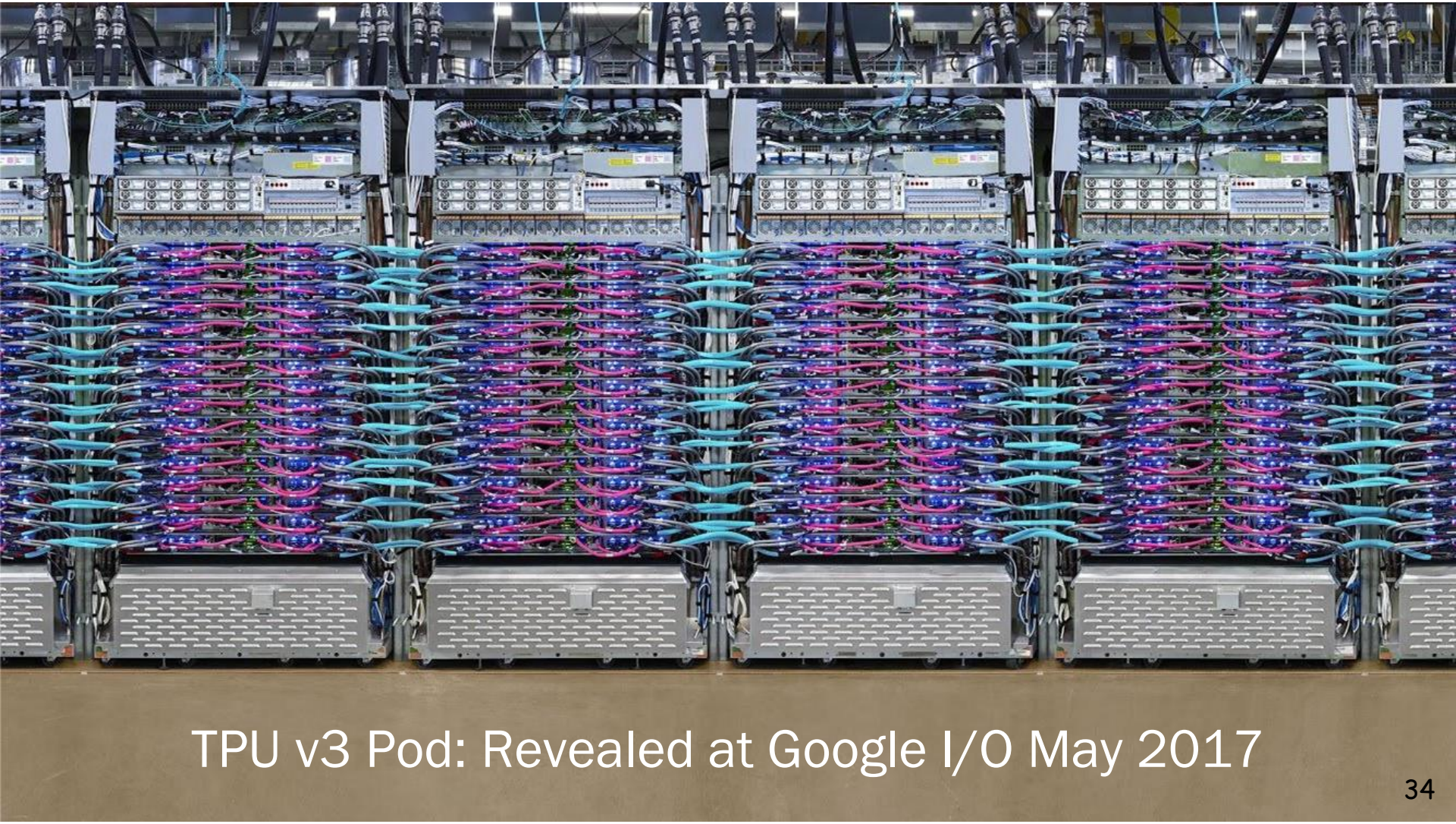
Key Concepts for Hardware

- *Training* or learning (development)
  vs. *Inference* or prediction (production)
- *Batch size*
  - Problem: DNNs have millions of weights that take a long time to load from memory (DRAM)
  - Solution: Large batch ⇒ Amortize weight-fetch time by inferring (or training) many input examples at a time
- Floating-Point vs. Integer ("*Quantization*")
  - Training in Floating Point on GPUs popularized DNNs
  - Inferring in Integers faster, lower energy, smaller

Cloud TPU chip layout

HBM: high bandwidth memory





TPU v3 Pod: Revealed at Google I/O May 2017

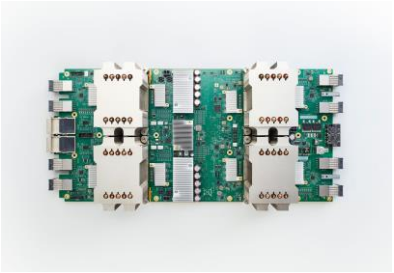
Relentless progress

g.co/cloudtpu

TPU v1
(deployed 2015)

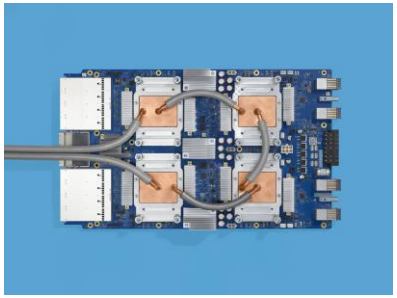
92 teraops
Inference only

Cloud TPUv2



180 teraflops
64 GB HBM
Training and inference
Generally available (GA)

Cloud TPUv3



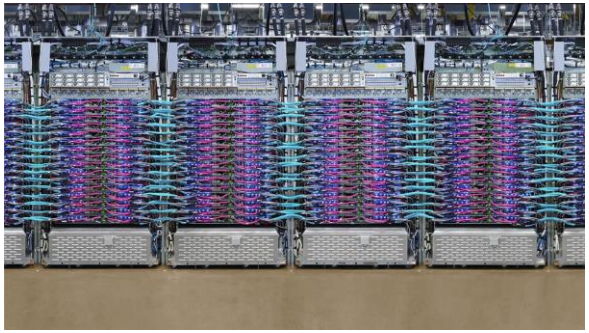
420 teraflops
128 GB HBM
Training and inference
Beta

Relentless progress

g.co/cloudtpu

Cloud TPU Pod (v2, 2017)

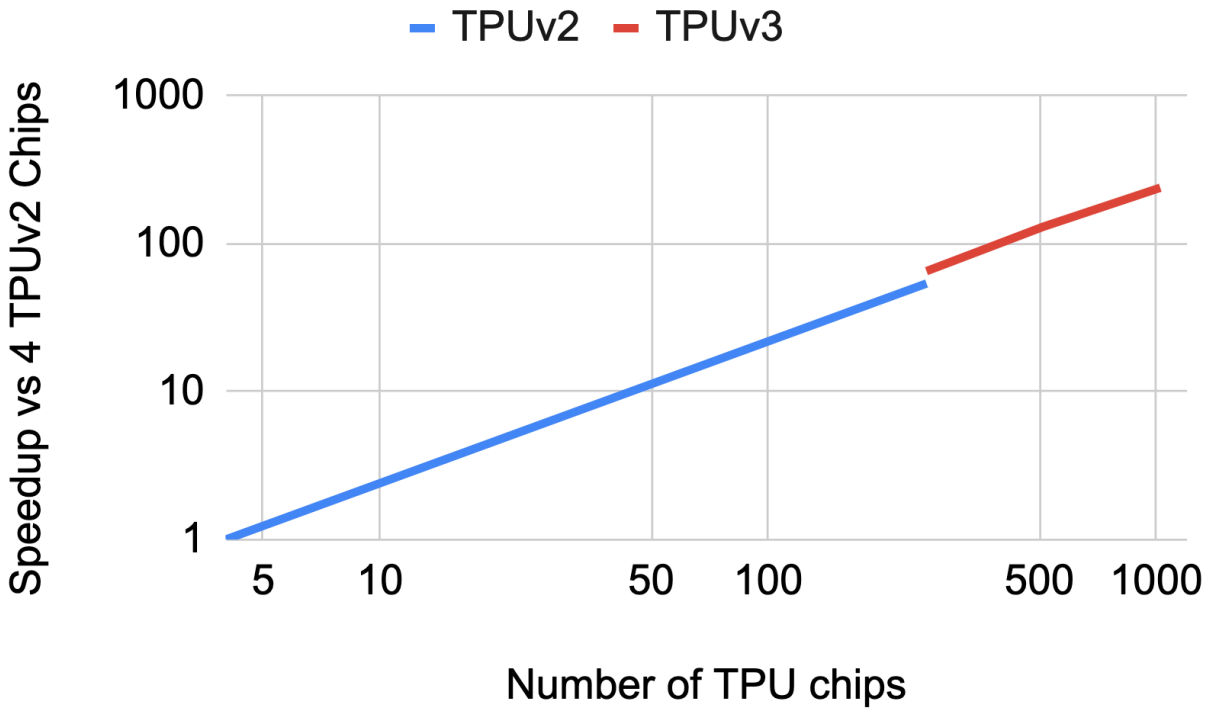
11.5 petaflops
4 TB HBM
2-D toroidal mesh network
Training and inference
Alpha



TPU v3 Pod (2018)

> 100 petaflops!
32 TB HBM
Liquid cooled
New chip architecture + larger-scale system

ResNet-50 Speedup



Ying, C., Kumar, S., Chen, D., Wang, T. and Cheng, Y., 2018. Image Classification at Supercomputer Scale. *arXiv preprint arXiv:1811.06992*.

ML Crisis as both Danger and Opportunity

Danger: the end of Moore's Law, Dennard Scaling, and standard CPU performance

- Limits of CMOS in sight
- Intel 10nm woes, Global Foundries 7nm exit

Opportunity: the revolution in ML

- Economic demand for ML accelerators
- Architectural and codesign experimentation and transformation
- Can we use ML to design better accelerators?

Irony: exponential demand for ML computation, just at the end of Moore's Law

- Efficiency is going to matter a lot
- Huge opportunities for HW/SW codesign in building TPUs and other DSAs

A New Golden Age

Hennessy and Patterson, "A New Golden Age for Computer Architecture," *Communications of the ACM*, February 2019

How do you get started?

- See Google pages on getting started.
- https://www.tensorflow.org/
- https://www.tensorflow.org/get_started/get_started
- https://www.tensorflow.org/tfrc/

References

- https://developers.google.com/appengine/
- Rick Merrit, EE|Times: Google Revs Faster TPU: http://www.eetimes.com/document.asp?doc_id=1331753&_mc=RSS_EET_EDT.