CS294:
Deep Learning Frameworks

Joey Gonzalez and Ion Stoica

February 4, 2019

History: single machine

2007: Sci-kit learn:

- Public release: **2010**
- Machine learning (ML) library for Python
- Large number of “classic” ML algorithms, e.g.,
  - Linear and logistic regression, SVM, random forests, k-means, gradient-boosting
- Highly successful to this day

History: big data ML

Lots of more data available, so people developed distributed algorithms:

- Still “classic” ML, e.g., logistic regression, collaborative filtering
- 2009: Mahout: ML library on top of Apache Hadoop
  - Slow. Each iteration reads/writes data on disk
- 2011: MLlib: ML library for Apache Spark
  - Developed at AMPLab, Berkeley
  - Much faster than Mahout: no reads/writes to the disk
  - Still the library of choice for distributed “classic” ML algorithms

Neural Networks: Single machine libraries

2007: Theano

theano

- Developed by Montreal Institute for Learning Algorithms (MILA)
- Initially no support for GPUs

Why support for GPU important?

- NN requires basically matrix multiplication
  - Space complexity: $O(N^2)$
  - Computation complexity: $O(N^3)$
- Thus, computation complexity super-linear in the input

Neural Networks: Single machine libraries

2007: Theano

- Developed by Montreal Institute for Learning Algorithms (MILA)
- Initially no support for GPUs

2014: Caffee

- Developed by Berkeley Vision and Learning Center
- Support for GPUs, some popular neural networks, e.g., AlexNet

2016: PyTorch

- Developed by Facebook
  - Loosely based on Torch (started in 2002, but no longer sctive)
- Initially single machine, recently distributed

Neural Networks: Distributed systems

2015: Tensorflow

- Developed by Google Brain
- The most popular ML library today

2015: MXNet

- Initially, by UW and others; now by AWS

Systems for data-parallel training leveraging single-machine Tensorflow and PyTorch

- Horovod, RLlib (Ray), …

Computation model

Dataflow graph, e.g.,

- MLlib (Spark), Tensorflow, MXNet, PyTorch

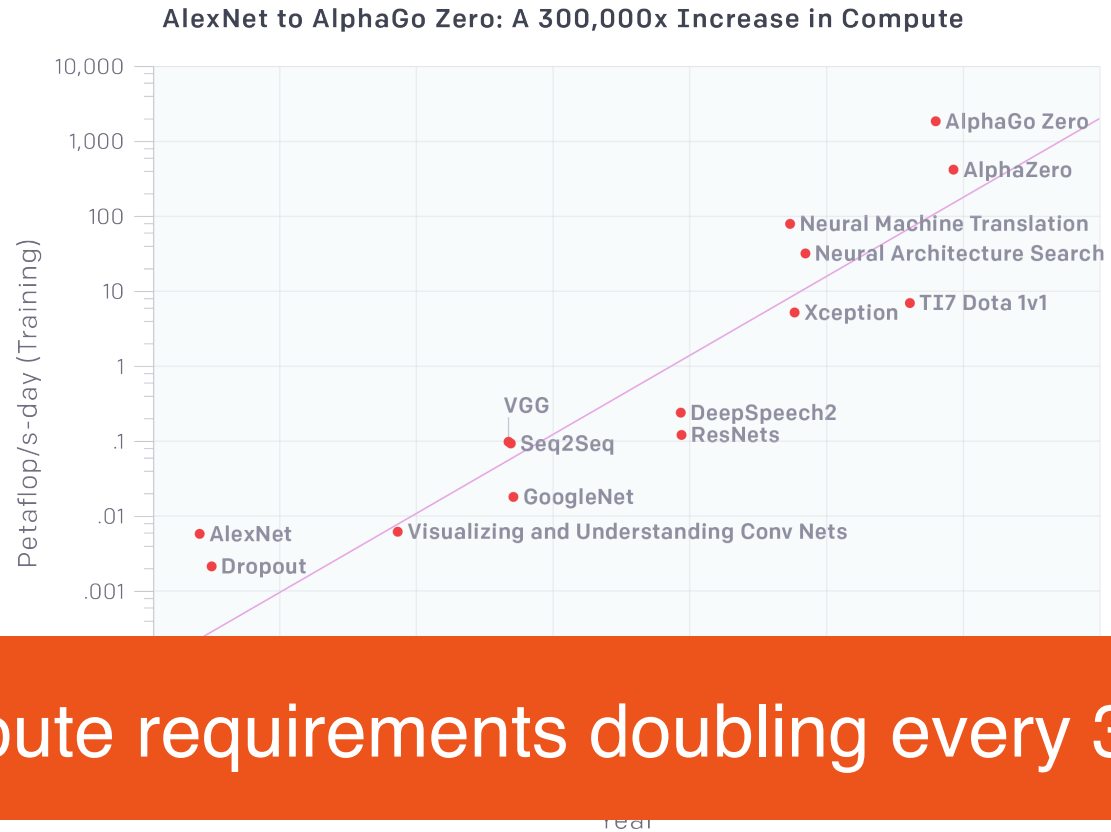
Evaluation:

- Lazy: MLlib (Spark), MXNet, Tensorflow (originally)
  - Enable better optimizations
- Eager: PyTorch
  - Easier to debug

Data:

- Immutable (e.g., Mllib): easy provide fault tolerance
- Mutable (e.g., Tensorflow, MXNet) : more efficient

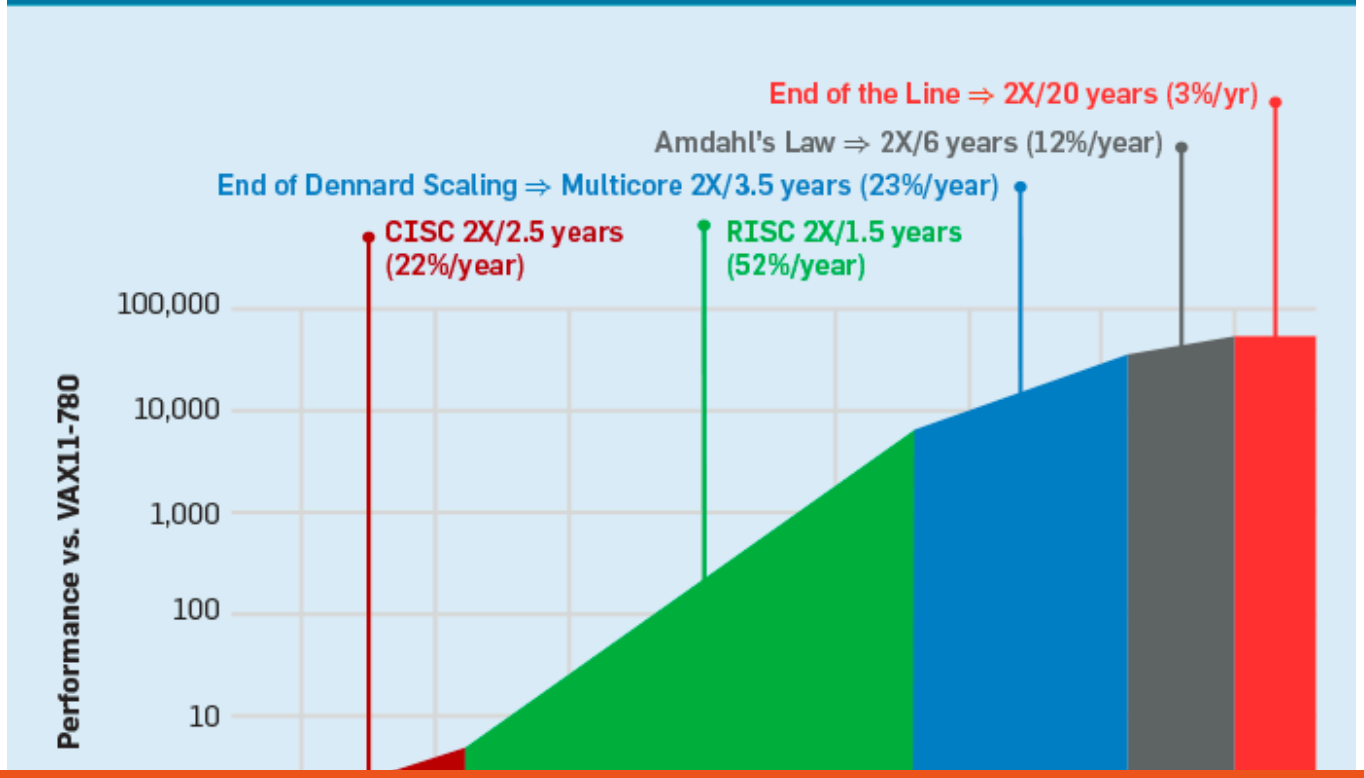
Compute system requirements



Compute requirements doubling every 3 months!

AI and Compute (https://blog.openai.com/ai-and-compute/)

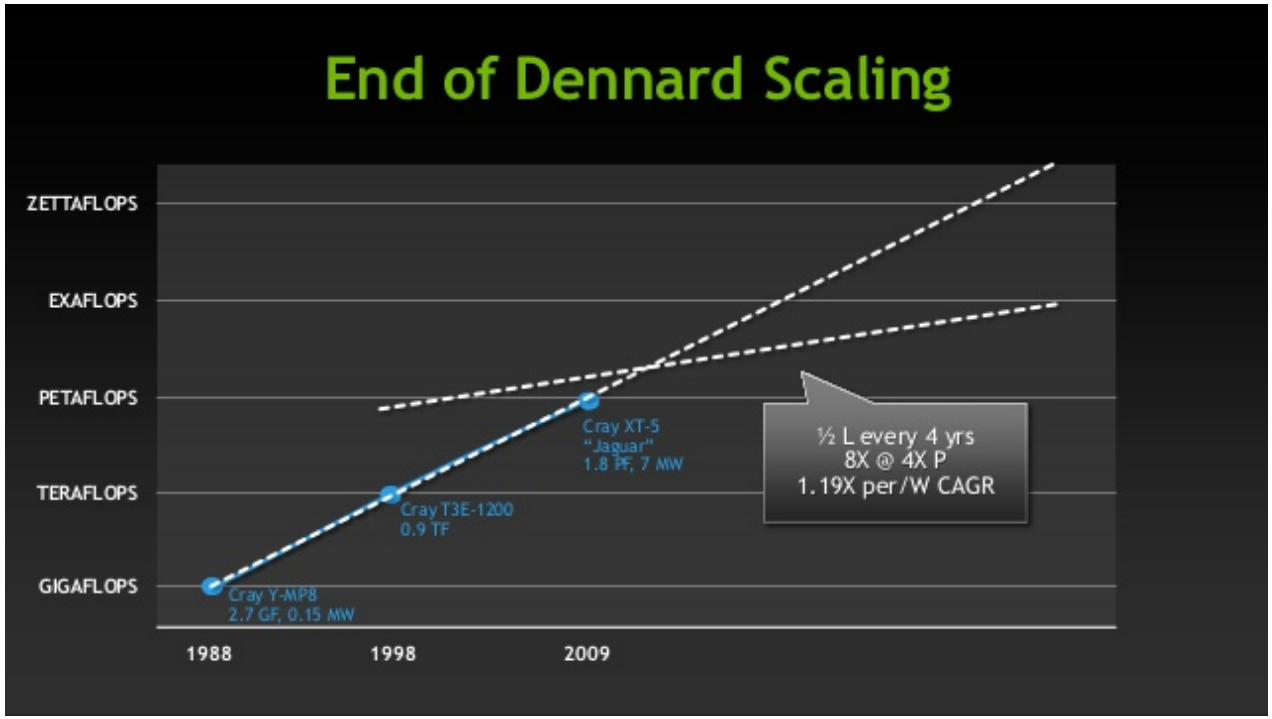
Figure 1. Following Hennessy and Patterson,[17] we plotted highest SPECCPUint performance per year for 32-bit and 64-bit processor cores over the past 40 years; the throughput-oriented SPECCPUint_rate reflects a similar profile, with plateauing delayed a few years.



Moore’s law is dead

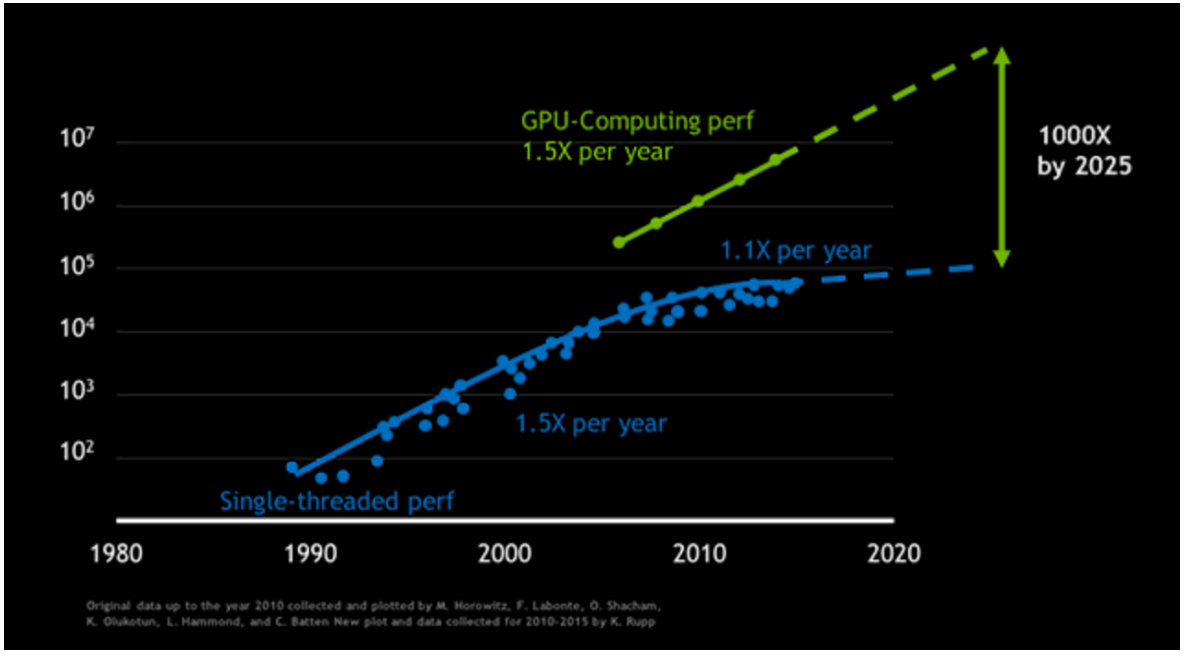
Dennard scaling

- As transistors get smaller, their power density stays constant
- Performance & memory capacity per-watt increase exponentially



In the meantime…

GPU performance increase still follows Moore's law



A plethora of NN accelerators are being developed (e.g. TPU)

So, what does it mean?

1. The computation requirements growing much faster than Moore's law
2. FLOPs still continue to double every 18 months
   - GPUs and hardware accelerators
3. However, RAM capacity growing very slowly
4. Next generation of ML systems
   - Distributed
   - Efficiently use specialized, heterogeneous hardware

Projects

By Wednesday 2/6:

- Check current list of projects: https://tinyurl.com/ycbqz22q
- Add your own project

AI-Sys Spring 2019

- **When:** *Mondays and Wednesdays from 9:30 to 11:00*
- **Where:** *Soda 405*
- **Instructors:** Ion Stoica and Joseph E. Gonzalez
- **Announcements:** Piazza
- **Sign-up to Present:** Google Spreadsheet
- **Project Ideas:** Google Spreadsheet

Projects

By Wednesday 2/6:

- Check current list of projects: https://tinyurl.com/ycbqz22q
- Add your own project

By Friday 2/8:

- Specify your project preference

By Monday 2/11:

- Project matching: at least two, and at most three people per project.