

Lecture 01: Revolutions Everywhere

D Donoho/ H Monajemi
Stats 285 Stanford

20170925

Stats 285 Fall 2017



Outline

The Computing Discontinuity

- Mobile is eating the world
- Mobile Drives IT Revolution
- Software is eating the world

The Revolution in Computational Science

Case Study: Deep Learning

- The Sudden Emergence of Deep Learning
- Emergence of Prediction Challenges
- The Slow Emergence of the Common Task Framework
- CTF Goes Mainstream
- Lessons from Case Study
- Framework Wars

Resistance

- Intellectual impoverishment
- Solution: The Great Enrichment

Painless Computational Experiments

Disclaimer

This document contains images obtained by routine Google Images searches. Some of these images may perhaps be copyright. They are included here for educational noncommercial purposes and are considered to be covered by the doctrine of Fair Use. In any event they are easily available from Google Images.

It's not feasible to give full scholarly credit to the creators of these images. We hope they can be satisfied with the positive role they are playing in the educational process.

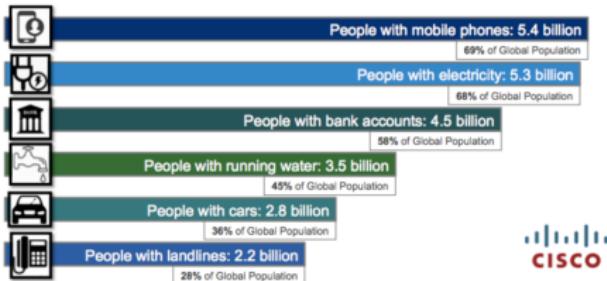
The Mobile Revolution



Mobile Is Spreading Everywhere

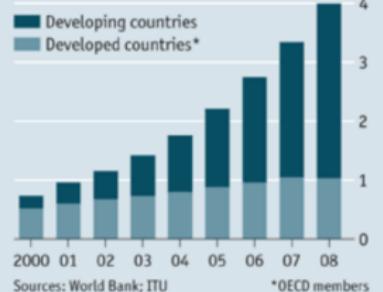
Mobile Growth Continues Through 2020

By 2020, more people will have mobile phones than electricity at home

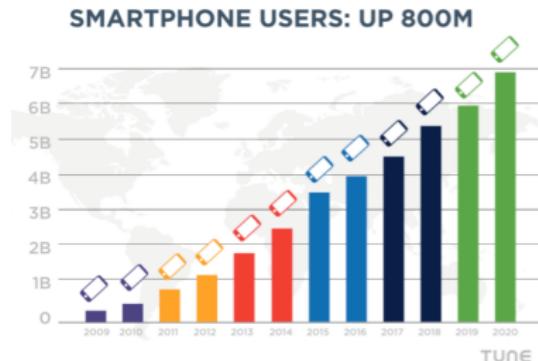


Where to look for growth

Mobile-phone subscriptions, bn

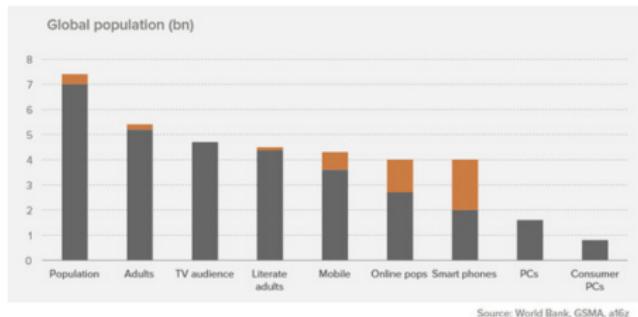


Smartphones are Spreading Everywhere



The world in 2020

By 2020 80% of the adults on earth will have a smartphone



The Computing Discontinuity
The Revolution in Computational Science
Case Study: Deep Learning
Resistance
Painless Computational Experiments

Mobile is eating the world
Mobile Drives IT Revolution
Software is eating the world

Literally Everywhere



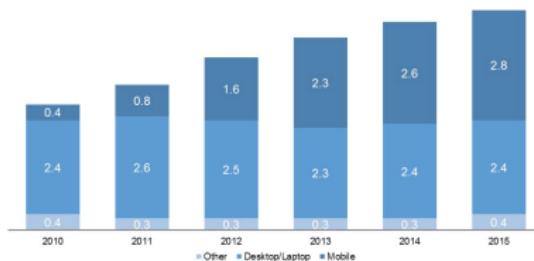
'Scientists in Antarctica are downloading mobile games. Parents in Syria worry about kids using too much tech. Islanders in American Samoa connect with 4G, and sherpas in Nepal chat on their smartphones while lugging 75-pound loads.'

JOHN KOETSIER TUNE 12/29/2015

<https://www.tune.com/blog/global-mobile-why-2016-is-the-global-tipping-point-for-the-mobile-economy/>

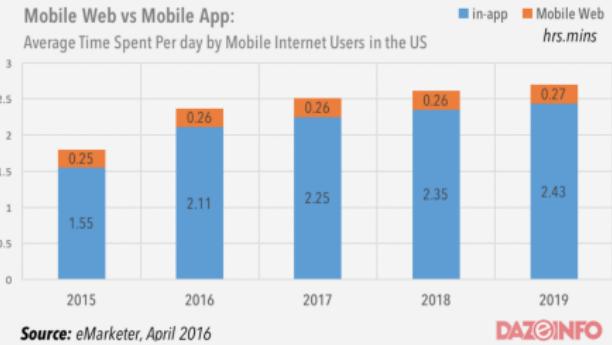
Engagement is Growing

Average Daily Internet Usage by Device (in hours)



Mobile Web vs Mobile App:

Average Time Spent Per day by Mobile Internet Users in the US



Source: eMarketer, April 2016

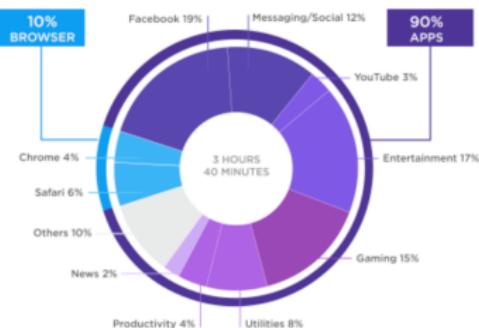
DAZINFO

The Computing Discontinuity
The Revolution in Computational Science
Case Study: Deep Learning
Resistance
Painless Computational Experiments

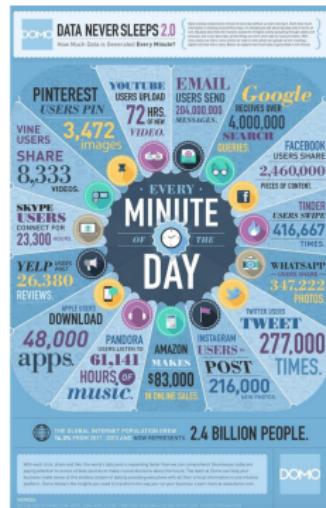
Mobile is eating the world
Mobile Drives IT Revolution
Software is eating the world

App Revolution – 1

90% of Time on Mobile is Spent in Apps



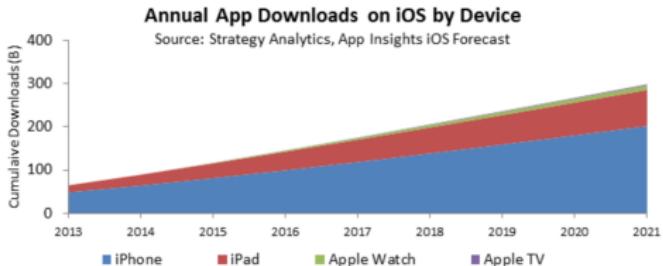
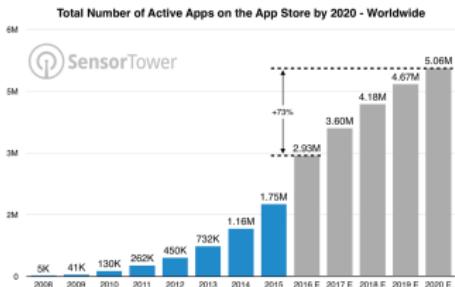
Source: Flurry Analytics, comScore, Pandora, Facebook, NetMarketShare. Note: US Jun 2015



The Computing Discontinuity
The Revolution in Computational Science
Case Study: Deep Learning
Resistance
Painless Computational Experiments

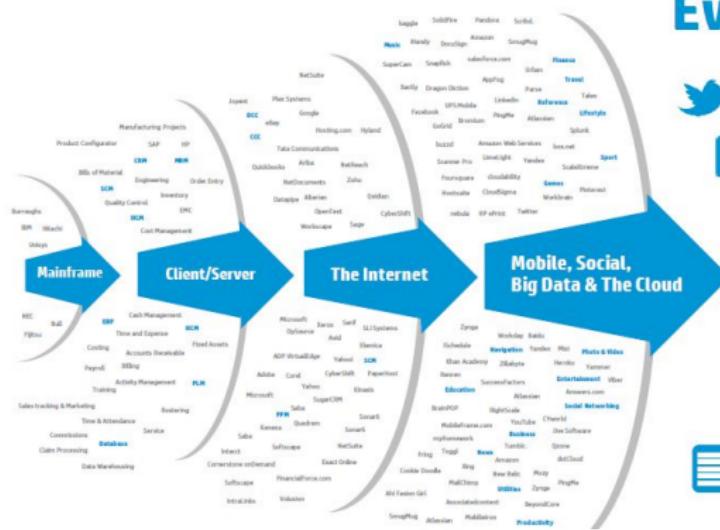
Mobile is eating the world
Mobile Drives IT Revolution
Software is eating the world

App Revolution– 2



Mobile Creates 24/7 Data Deluge

A new style of IT emerging



Every 60 seconds

98,000+ tweets

695,000 status updates

11million instant messages

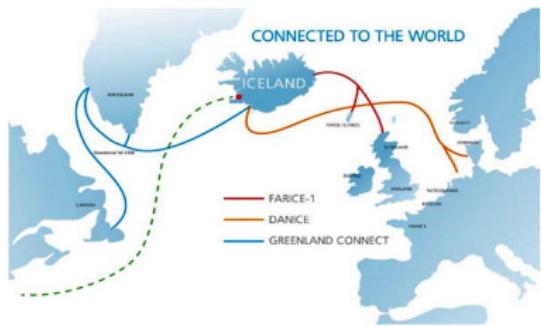
698,445 Google searches

168 million+ emails sent

1,820TB of data created

217 new mobile web users

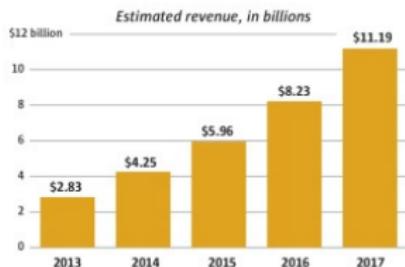
24/7 Deluge Spawns Global Computational Services



Emergence of Cloud Services

Amazon Web Services sales

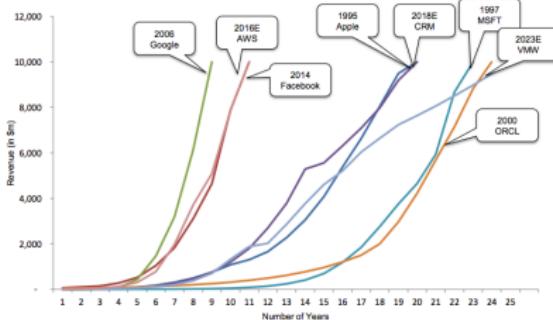
Amazon will break out specific sales data for AWS on Thursday for the first time. Here's Robert W. Baird & Co. analyst Colin Sebastian estimates.



Source: Robert W. Baird & Co.

KELLY SHEA / THE SEATTLE TIMES

Figure 9: AWS is the Fastest-Growing Enterprise Technology Company Ever



Source: Deutsche Bank Estimates, Public Company Filings

Cloud Paradigm

Cloud Paradigm:

- ▶ Billions of smart devices each drive queries to cloud servers
- ▶ Millions of business relying on cloud for all needs

Symbiosis of cloud and economy is *lasting* and *disruptive*.

Software is Eating the World – I

Why Software Is Eating The World

By MARC ANDREESSEN

This week, Hewlett-Packard (where I am on the board) announced that it is exploring jettisoning PC business in favor of investing more heavily in software, where it sees better potential for growth. Google plans to buy up the cellphone handset maker Motorola Mobility. Both moves surprised me. But both moves are also in line with a trend I've observed, one that makes me optimistic about the American and world economies, despite the recent turmoil in the stock market.



In an interview with WSJ's Kevin Delaney, Groupon and venture investor Marc Andreessen made a related point: that companies do not constitute a bubble. He also stressed that both Apple and Google are undervalued and that "the market doesn't like tech."

In short, software is eating the world.

More than 10 years after the peak of the 1990s dot-com bubble, Twitter is sparking controversy in Silicon Valley. Rapidly growing private market valuations, an occasional successful IPO. With scars from the Webvan and Pets.com still fresh in the investment community, many are asking, "Isn't this just a dangerous new bubble?"

I, along with others, have been arguing the opposite case. (I am co-founder and general partner of firm Andreessen-Horowitz, which has invested in

"Six decades into the computer revolution, four decades since the invention of the microprocessor, and two decades into the rise of the modern Internet, all of the technology required to transform industries through software finally works and can be widely delivered at global scale."

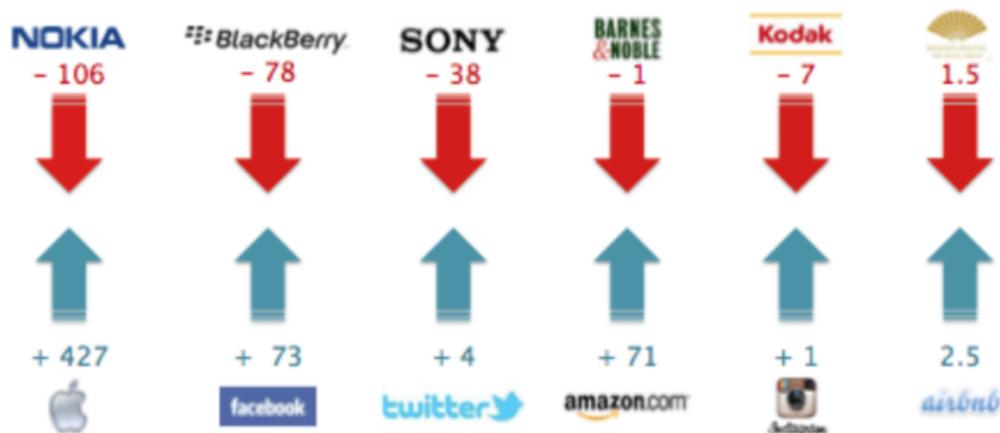
Marc Andreessen - WSJ - 2011

Software is Eating the World – II



Software is Eating the World – III

Software is eating the world



Market Cap Change in 5 Years in US\$ B

Stack Paradigm I

Stack Paradigm:

- ▶ Organizations combine software components from other providers in a stack
- ▶ Massive new capabilities emerge by hybridizing components

Examples:

- ▶ Uber (next slide)
- ▶ Netflix relies on AWS
- ▶ Snap, Dropbox etc. small teams

Stack Paradigm II



Uber doesn't own their cars. They also don't directly employ their own drivers. So, one might ask, what do they own exactly as a core asset? The core application and ecosystem around the Uber experience is their primary asset and differentiator. But to deliver that experience, they apply rigorous focus.

At the practical level, when you look at the technology components of Uber's world-renowned app, they decided to rely on other core platforms and technologies to power many of the key elements.

Jeetu Patel, *Software is still eating the world*, TechCrunch, Jan 2016

Stack Paradigm III

"Six decades into the computer revolution, four decades since the invention of the microprocessor, and two decades into the rise of the modern Internet, all of the technology required to transform industries through software finally works and can be widely delivered at global scale."

Marc Andreessen - WSJ - 2011

Explosion of Computational Resources

Cloud Paradigm:

- ▶ Billions of smart devices each drive queries to cloud servers
- ▶ Millions of business relying on cloud for all needs

Symbiosis of cloud and economy is *lasting* and *disruptive*.

Cloud provides *any user same-day* delivery:

- ▶ Tens to hundreds of thousands of hours of CPU
- ▶ Pennies per CPU hour

Any user can consume *1 Million CPU hours* over a few days for a few \$10K's.

Massive Computational Power Will Transform *Science*

Traditionally:

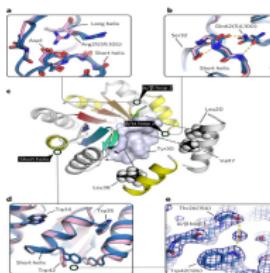
- ▶ Deduction (in math)
- ▶ Induction (in physical sciences)

Emerging new approach:

- ▶ Massive computational experiments

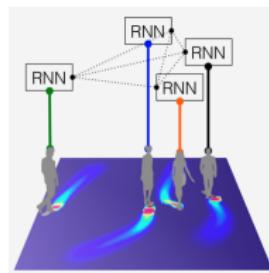
Massive Computations in Science

Traditionally computational fields



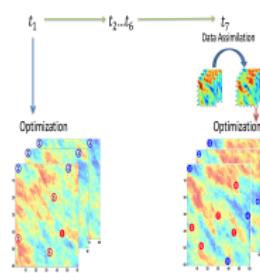
Protein Design

(Huang et al. 2016)



AI

(Alahi et al. 2016)

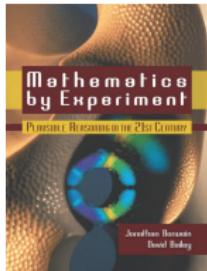


Oil Field Develop.

(Shirangi et al. 2015)

Massive Computations in Science

Traditionally **non-** computational field – Mathematics



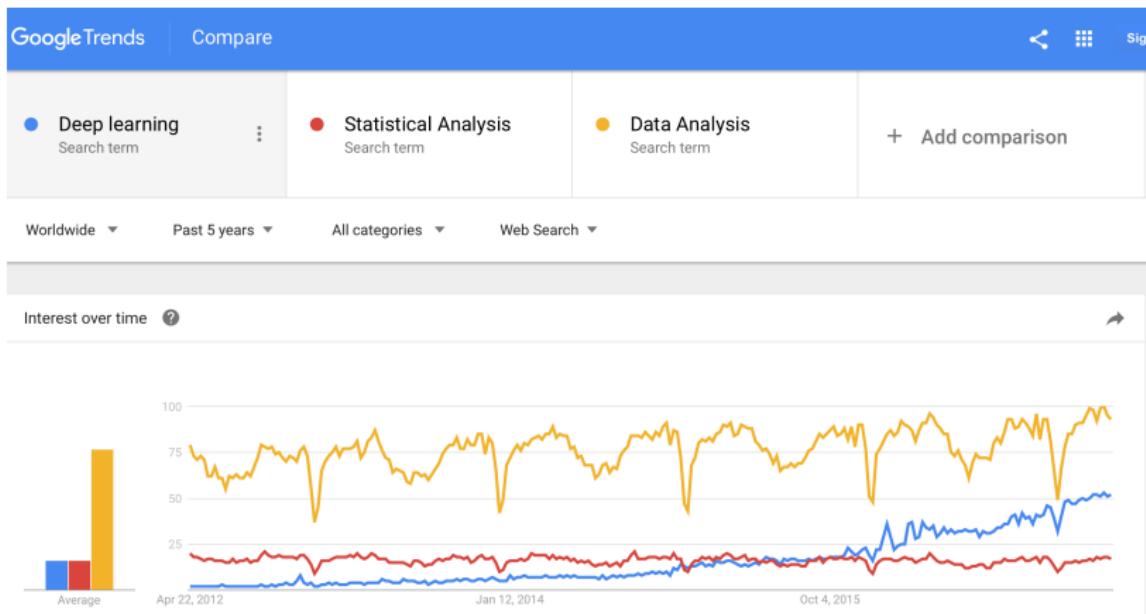
Borwein/Bailey

Borwein/Devlin

Individual Articles

The Computing Discontinuity
The Revolution in Computational Science
Case Study: Deep Learning
Resistance
Painless Computational Experiments

The Sudden Emergence of Deep Learning
The Slow Emergence of the Common Task Framework
CTF Goes Mainstream
Lessons from Case Study
Framework Wars



The Computing Discontinuity
The Revolution in Computational Science
Case Study: Deep Learning
Resistance
Painless Computational Experiments

The Sudden Emergence of Deep Learning
The Slow Emergence of the Common Task Framework
CTF Goes Mainstream
Lessons from Case Study
Framework Wars

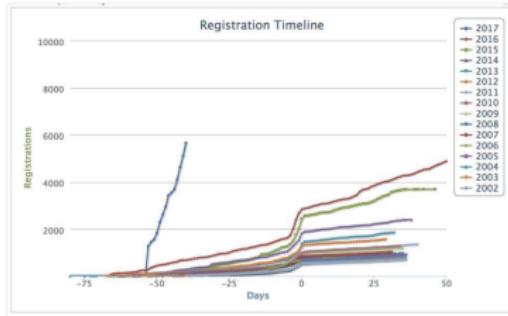


Alex Lebrun
@lxbrun

Follow



Deep learning hype in one picture
(NIPS conference registrations, 2002 through
2017) #nips2017



8:20 AM - 15 Sep 2017

758 Retweets 1,005 Likes



20 758 1.0K

The Computing Discontinuity
The Revolution in Computational Science
Case Study: Deep Learning
Resistance
Painless Computational Experiments

The Sudden Emergence of Deep Learning
The Slow Emergence of the Common Task Framework
CTF Goes Mainstream
Lessons from Case Study
Framework Wars



Andrej Karpathy

@karpathy

Follow

Came to visit first class of [@cs231n](#) at Stanford. 2015: 150 students, 2016: 350, this year: 750. [#aiinterestsingularity](#)



12:11 PM - 4 Apr 2017

155 Retweets 623 Likes



19 155 623



michael_nielsen @michael_nielsen · Apr 4

Replying to [@karpathy @cs231n](#)

Faster than Moore's Law. At this rate - doubling each year - in 24 years everyone on Earth will be enrolled :-)

Synchronies, 1

Over same timeframe – 2010-2014

- ▶ Instagram, Snapchat emerge to global prominence
- ▶ Deep Learning catapults to global attention

Coincides with emergence of

- ▶ Smartphone photography
- ▶ Cloud computing
- ▶ Cloud storage of selfie/smartphone photography

Synchronies, 2

"Six decades into the computer revolution, four decades since the invention of the microprocessor, and two decades into the rise of the modern Internet, all of the technology required to transform industries through software finally works and can be widely delivered at global scale."

Marc Andreessen - WSJ - 2011

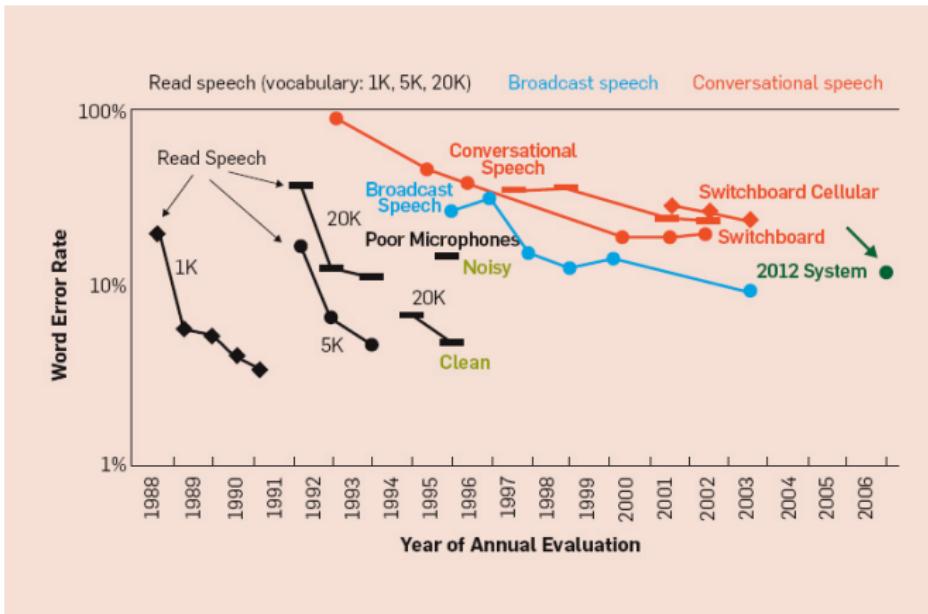
Common Task Framework (1980's)

Under CTF we have the following ingredients

- (a) A **publicly available training dataset** involving, for each observation, a list of (possibly many) feature measurements, and a class label for that observation.
- (b) A set of **enrolled competitors** whose **common task** is to **infer** a class **prediction rule from the training data**.
- (c) A **scoring referee**, to which competitors can submit their prediction rule. The referee runs the prediction rule against a testing dataset which is sequestered behind a Chinese wall. The referee objectively and automatically reports the score achieved by the submitted rule.

See Mark Liberman's description (Liberman, 2009).

CTF *Really* Works!



CTF Lifestyle – 1

1. Researchers set up local copies of Challenge
 - ▶ Data – Training, Test carved out of public dataset
 - ▶ Scoring – same as challenge scoring rule
2. Researcher's job: '*tuning models*'
 - ▶ Think up a family of model variations – '*tweak's*
 - ▶ Run a full '*experiment*' – suite of tweaks – '*grid*'
 - ▶ Score each tweak
 - ▶ Submit best-scoring result to central authority
3. Successful researchers perpetually motivated by
Game-ification: tweaking, scoring, winning.
4. Researchers who tweak more often, win more often!.
5. If easier to implement tweaks and faster to evaluate them, more likely to win!.

The Computing Discontinuity
The Revolution in Computational Science
Case Study: Deep Learning
Resistance
Painless Computational Experiments

The Sudden Emergence of Deep Learning
The Slow Emergence of the Common Task Framework
CTF Goes Mainstream
Lessons from Case Study
Framework Wars

CTF Lifestyle – 2



Sebastião Salgado Work

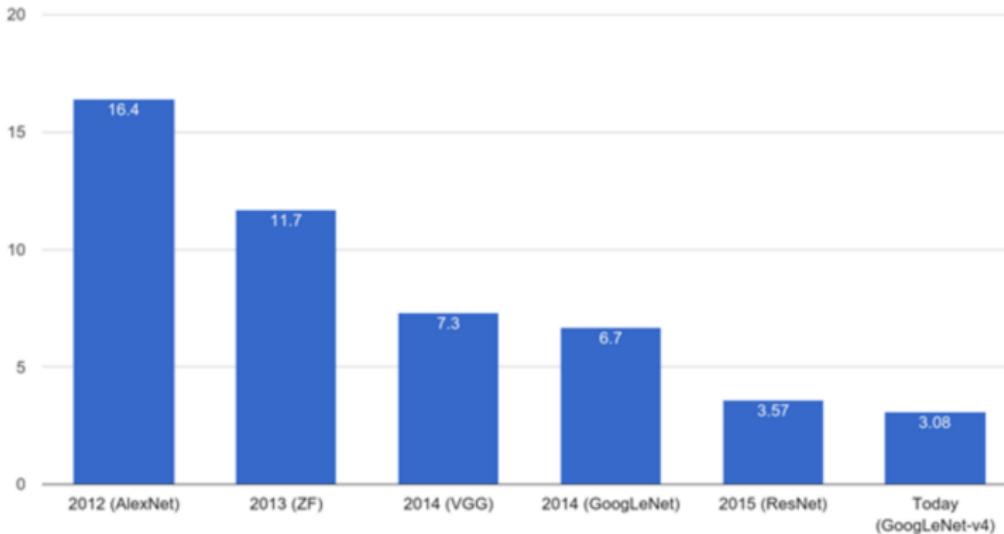
D Donoho / H Monajemi Stats 285 Stanford

Lecture 01: Revolutions Everywhere

CTF Goes Mainstream

1. Netflix Challenge (2009)
\$1 Million Prize
2. Kaggle (2010)
1 Million'th competitor expected Sept. 2017
3. Fei-Fei Li masterminds ImageNet 2008-2010
4. Hinton's Deep Learning Team wins ImageNet 2012

ImageNet Classification Error (Top 5)





Andrej Karpathy

@karpathy

Follow



You can now understand state of the art AI with before high school math. You forward a neural net and repeat guess&check. works well enough.

12:53 PM - 14 Mar 2017

50 Retweets 207 Likes



12

50

207

The Computing Discontinuity
The Revolution in Computational Science
Case Study: Deep Learning
Resistance
Painless Computational Experiments

The Sudden Emergence of Deep Learning
The Slow Emergence of the Common Task Framework
CTF Goes Mainstream
Lessons from Case Study
Framework Wars

Researchers Preparing for NIPS 2017



Lessons from Deep Learning Case Study

1. *Researchers who tweak more often, win more often!*
2. *If easier to implement tweaks and faster to evaluate them, more likely to win!*
3. Successful Research Environment
 - ▶ Easy to tweak models
 - ▶ Easy to score tweaks
 - ▶ Fast to score tweaks
4. Successful researchers perpetually motivated by
Game-ification: tweaking, scoring, winning.
5. Easier to stay motivated when easier and more comfortable to play the game.
 - ▶ Elegant expression of tweaks
 - ▶ Rapid turn-around for scoring

Framework Wars – 1

Influential Frameworks for Deep Learning Influential Frameworks for Deep Learning

- ▶ **Matlab**
 - pre-framework
- ▶ **TensorFlow**
 - open source (Originally by Google Brain)
- ▶ **Torch**
 - scientific computing framework written in Lua
- ▶ **PyTorch**
 - Python package for scientific computing (310 contributors)
- ▶ **Keras**
 - A Python wrapper around TensorFlow, CNTK and

Framework Wars – 2



Andrej Karpathy @karpathy

[Follow](#) ▾

Matlab is so 2012. Caffe is so 2013. Theano is so 2014. Torch is so 2015. TensorFlow is so 2016. :D

12:08 PM - 8 Feb 2017

248 Retweets 618 Likes



47 248 618



Sergio @sguada · Feb 8

Replying to @karpathy

what's your bet for 2017?

1 1 1



Andrej Karpathy @karpathy · Feb 8

PyTorch!! But I'm quite sure TensorFlow will do just fine too :)

1 4 35



Yann LeCun
@ylecun

[Follow](#)

Fast.ai has switched from Keras+TensorFlow to PyTorch for their deep learning course. They tell us why in great...
fb.me/17FW41uyw

7:44 PM - 9 Sep 2017

386 Retweets 937 Likes



7 386 937

Framework Wars – 3

Andrej Karpathy  [Follow](#)

The updated ImageNet training example with support for distributed training is a beauty
[github.com/pytorch/exampl...](https://github.com/pytorch/examples) clean 300 lines


pytorch/examples
A set of examples around pytorch in Vision, Text, Reinforcement Learning, etc.
github.com

10:34 AM - 6 Aug 2017

291 Retweets 850 Likes   

4 291 850

hardmaru @hardmaru · Aug 6
Replying to @karpathy
These examples convey concepts more clearly compared to 8-page papers.
Better than reading pseudocode.

1 8 78

Andrej Karpathy  [Follow](#)

Pretty good list. Except the article makes it sound like there's a contest.

Awni Hannun [@awnihannun](#)
PyTorch or TensorFlow? Wrote up some of my thoughts on the question - awni.github.io/pytorch-tensor ...

9:13 PM - 18 Aug 2017

71 Retweets 276 Likes   

1 71 276

Andrej Karpathy  [@karpathy](#) · Aug 18
Replies to @karpathy
:) would add few categories, esp profiling, size/interpretability of lib code base, distributed training, community/support, ...

1 52

Anmol Jawandha  [@anmolsj](#) · Aug 18
Is there any reason one should use Tensorflow over PyTorch for research purposes (given pytorch now supports distributed training)?

Framework Wars - 4

The real action is all in frameworks

1. Dream up, test, and publish better ...
 - ▶ Types of models
 - ▶ Types of tweaks
 - ▶ Properties for evaluation
2. Implement better *frameworks* ...
 - ▶ More elegant expression of models, tweaks
 - ▶ Distributed Learning across clusters
 - ▶ Smoother collection and analysis of results

Resistance – 1

We are at a university!

1. Q: *Where's the intellectual activity in tuning?*
2. Q: *I didn't come here to do hard manual labor!*
3. Q: *I didn't come here to compete as mindless drones!*

Resistance t- 2

We are at a university!

1. Q: *Where's the intellectual activity in tuning?*
2. Q: *I didn't come here to do hard manual labor!*
3. Q: *I didn't come here to compete as mindless drones!*

What we see:



Sebastiao Salgado, *Work*

Resistance 3

We are at a university!

1. Q: *Where's the intellectual activity in tuning?*
2. Q: *I didn't come here to do hard manual labor!*
3. Q: *I didn't come here to compete as mindless drones!*

What we **imagine**:



Metaphor: Computers as Slavery

Traditionally, ‘using computers’ involves interactively running programs (Excel, Point-and-click)

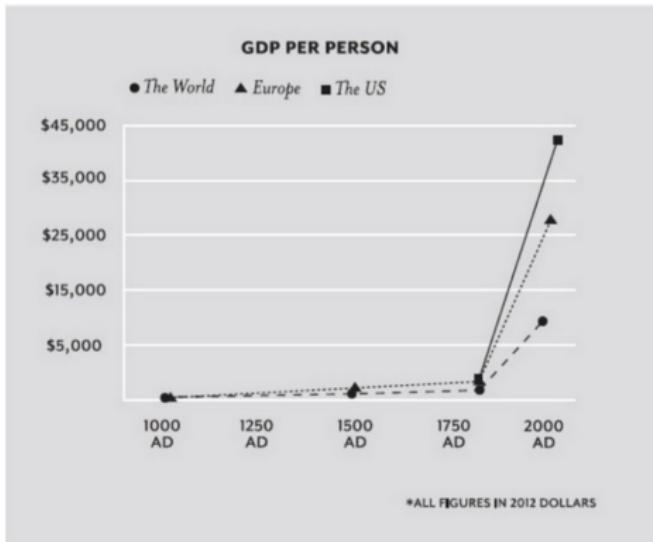
Claerbout’s Dictum: “... dependence on an interactive program can be a form of slavery”

<http://sepwww.stanford.edu/sep/jon/reproducible.html>

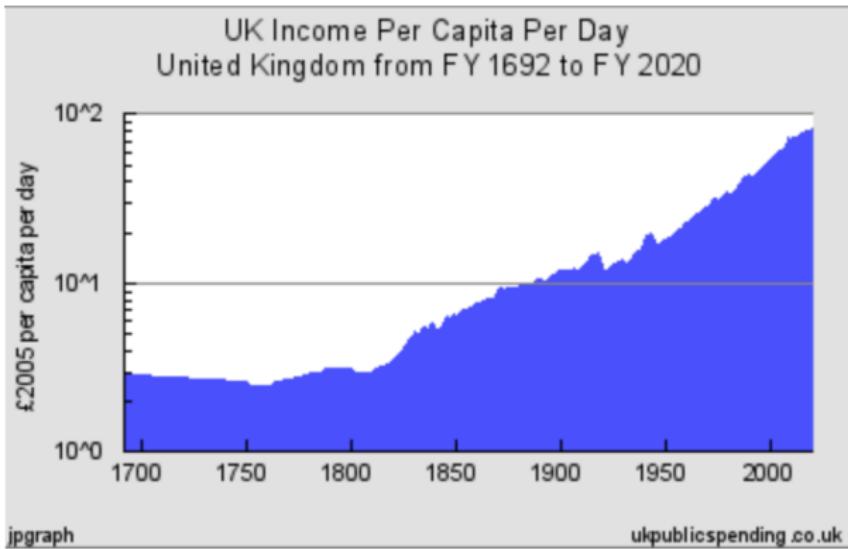


Photo: Jon Claerbout Cartoon: <http://fritsAhlefeldt.com>

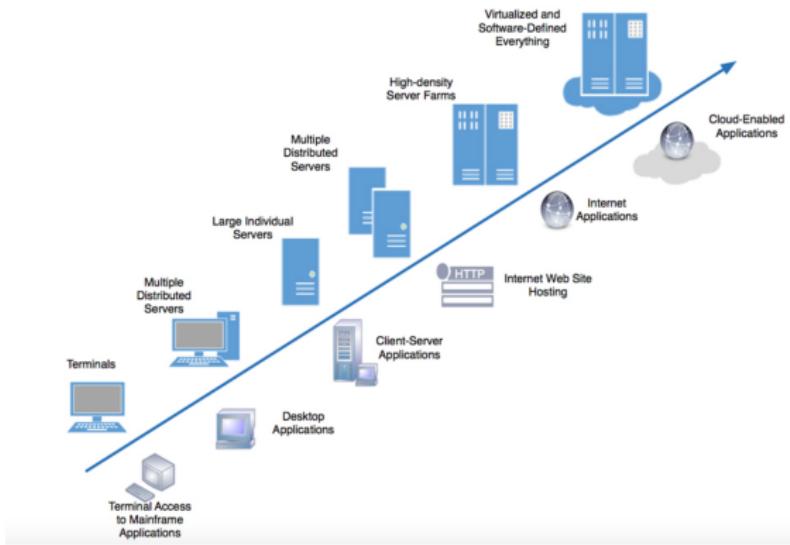
Digression: The Great Enrichment (Deidre McKloskey) 1



Digression: The Great Enrichment (Deidre McKloskey) 2



The Great IT Enrichment – 1



The Great IT Enrichment - 2

Our vision.

The intellectual poverty of the old interactive 'Excel'-era paradigm are real, but they will be transcended.

New and better and more powerful abstractions will lift us out of the mud and out of slavery.

Coming Soon to a Scientific field near you

In the near future,

- ▶ Scientific research will be transformed
 - ▶ *1 million CPU Hours* behind research papers and theses
 - ▶ *Widespread acceptance* of empirical/simulation evidence
- ▶ 1 million-hour hurdle manageable through *new frameworks*.
- ▶ Frameworks offer Convenient and Efficient
 - ▶ ... definition of experiments
 - ▶ ... management of jobs
 - ▶ ... gathering of results
 - ▶ ... analysis and presentation
- ▶ Output:
 - ▶ Better science
 - ▶ Better math

Course Focus: Frameworks for Massive Experiments, 1

- ▶ Traditional issues
 - ▶ Experiments implicitly defined by executing unorganized code
 - ▶ Hard to understand what the baseline is, what variations are
 - ▶ Code dependencies unclear
 - ▶ Ordeal to get all the jobs to run, maybe gave up early
 - ▶ Tedious to harvest all the data, maybe missing some data
 - ▶ Confusing manual compilation and reporting
- ▶ Modern Frameworks
 - ▶ Systematic structure to coding
 - ▶ Base experiment clearly defined
 - ▶ Tweaks clearly defined
 - ▶ Code dependencies explicit
 - ▶ Grid of Jobs run systematically
 - ▶ Automatic transparent access of (cluster, AWS,...)
 - ▶ Data Harvested automatically to central data repository
 - ▶ Data analyzed automatically using defined tools

Course Focus: Frameworks for Massive Experiments, 2

- ▶ Example Frameworks
 - ▶ By individual research teams:
 - ▶ ClusterJob – Hatef Monajemi
 - ▶ CodaLab – Percy Liang
 - ▶ By startups:
 - ▶ Databricks
 - ▶ Civis Analytics
 - ▶ Domino Data Labs

A Look Ahead: <https://stats285.github.io>

Guest Lectures



Monday, 10/9/2017
Eric Jonas
UC Berkeley



Monday, 10/16/2017
Victoria Stodden
iSchool, UIUC



Monday, 10/23/2017
Percy Liang
Stanford



Monday, 10/30/2017
Travis Oliphant
Anaconda



Monday, 11/06/2017
Riccardo Murri
University of Zurich



Monday, 11/13/2017
Andy Konwinski
Databricks

Eric Jonas
Victoria Stodden
Percy Liang
Travis Oliphant
Riccardo Murri
Andy Konwinski

Occupy the Cloud
Enhancing Reproducibility...
CodaLab Worksheets
NumPy
IT for research: a journey ...
Databricks

Global Economy → Computing → Science

