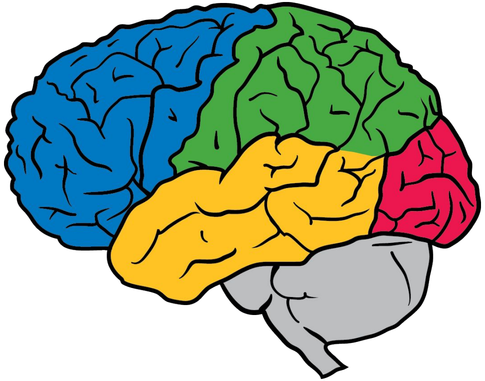


Scaling Machine Learning with TensorFlow

Jeff Dean
Google Brain team
g.co/brain

Presenting the work of **many** people at Google

Our Mission:
Make Machines Intelligent.
Improve People's Lives.

Google Brain Team: Research Impact

- Since 2012, published > 130 papers at top venues in machine learning
- Some highlights:
  - 2012: DistBelief, unsupervised learning to discover cats
  - 2013: opensource of word2vec
  - 2014: sequence to sequence learning, image captioning
  - 2015: Inception, DeepDream, TensorFlow
  - 2016: neural translation, medical imaging, architecture search

Main Research Areas

- General Machine Learning Algorithms and Techniques
- Computer Systems for Machine Learning
- Natural Language Understanding
- Perception
- Healthcare
- Robotics
- Music and Art Generation

Main Research Areas

- General Machine Learning Algorithms and Techniques
- Computer Systems for Machine Learning
- Natural Language Understanding
- Perception
- Healthcare
- Robotics
- Music and Art Generation

Google Research Blog

The latest news from Research at Google

The Google Brain team — Looking Back on 2016

Thursday, January 12, 2017

Posted by Jeff Dean, Google Senior Fellow, on behalf of the entire Google Brain team

The Google Brain team's long-term goal is to create more intelligent software and systems that improve people's lives, which we pursue through both pure and applied research in a variety of different domains. And while this is obviously a long-term goal, we would like to take a step back and look at some of the progress our team has made over the past year, and share what we feel may be in store for 2017.

Research Publications

One important way in which we assess the quality of our research is through publications in top tier international machine learning venues like ICML, NIPS, and ICLR. Last year our team had a total of 27 accepted papers at these venues, covering a wide ranging set of topics including program synthesis, knowledge transfer from one network to another, distributed training of machine learning models, generative models for language, unsupervised learning for robotics, automated theorem proving, better theoretical understanding of neural networks, algorithms for improved reinforcement learning, and many others. We also had numerous other papers accepted at conferences in fields such as natural language processing (ACL, CoNNL), speech (ICASSP), vision (CVPR), robotics (ISER), and computer systems (OSDI). Our group has also submitted 34 papers to the upcoming ICLR 2017, a top venue for cutting-edge deep learning research. You can learn more about our work in our list of papers, here.

Natural Language Understanding

Allowing computers to better understand human language is one key area for our research. In late 2014, three Brain team researchers published a paper on Sequence to Sequence Learning with Neural Networks, and demonstrated that the approach could be used for machine translation. In 2015, we showed that this this approach could also be used for generating captions for images, parsing sentences, and solving computational geometry problems. In 2016, this previous research (plus many enhancements) culminated in Brain team members worked closely with members of the Google Translate team to wholly replace the translation algorithms powering Google Translate with a completely end-to-end learned system (research paper). This new system closed the gap between the old system and human quality translations by up to 85% for some language pairs. A few weeks later, we showed how the system could do "zero-shot translation", learning to translate between languages for which it had never seen example sentence pairs (research paper). This system is now deployed on the production Google Translate service for a growing number of language pairs, giving our users higher quality translations and allowing people to communicate more effectively across language barriers. Gideon Lewis-Kraus documented this translation effort (along with the history of deep learning and the history of the Google Brain team) in "The Great A.I. Awakening", an in-depth article that appeared in *The NY Times Magazine* in December, 2016.

Robotics

Traditional robotics control algorithms are carefully and painstakingly hand-programmed, and therefore embodying robots with new capabilities is often a very laborious process. We believe that having robots automatically learn to acquire new skills through machine learning is a better (research paper). Our robots made about 800,000 grasping attempts during this research. Later in the year, we explored three possible ways for robots to learn new skills, through reinforcement learning, through their own interaction with objects, and through human demonstrations. We're continuing to build on this work in our goals for making robots that are able to flexibly and readily learn new tasks and operate in messy, real-world environments. To help other robotics researchers, we have made multiple robotics datasets publicly available.

Healthcare

We are excited by the potential to use machine learning to augment the abilities of doctors and healthcare practitioners. As just one example of the possibilities, in a paper published in the *Journal of the American Medical Association* (JAMA), we demonstrated that a machine-learning driven system for diagnosing diabetic retinopathy from a retinal image could perform on-par with board-certified ophthalmologists. With more than 400 million people at risk for blindness if early symptoms of diabetic retinopathy go undetected, but too few ophthalmologists to perform the necessary screening in many countries, this technology could help ensure that more people receive the proper screening. We are also doing work in other medical imaging domains, as well as investigating the use of machine learning for other kinds of medical prediction tasks. We believe that machine learning can improve the quality and efficiency of the healthcare experience for doctors and patients, and we'll have more to say about our work in this area in 2017.

Music and Art Generation

Technology has always helped define how people create and share media — consider the printing press, film or the electric guitar. Last year we started a project called Magenta to explore the intersection of art and machine intelligence, and the potential of using machine learning systems to augment human creativity. Starting with music and image generation and moving to areas like text generation and VR, Magenta is advancing the state-of-the-art in generative models for content creation. We've helped to organize a one-day symposium on these topics and supported an art exhibition of machine generated art. We've explored a variety of topics in music generation and artistic style transfer, and our jam session demo won the Best Demo Award at NIPS 2016.

AI Safety and Fairness

As we develop more powerful and sophisticated AI systems and deploy them in a wider variety of real-world settings, we want to ensure that these systems are both safe and fair, and we also want to build tools to help humans better understand the output they produce. In the area of AI safety, in a cross-institutional collaboration with researchers at Stanford, Berkeley, and OpenAI, we published a white paper on Concrete Problems in AI Safety (see the blog post here). The paper outlines some specific problems and areas where we believe there is real and foundational research to be done in the area of AI safety. One aspect of safety on which we are making progress is the protection of the privacy of training data, obtaining differential privacy guarantees, most recently via knowledge transfer techniques. In addition to safety, as we start to rely on AI systems to make more complex and sophisticated decisions, we want to ensure that those decisions are fair. In a paper on equality of opportunity in supervised learning (see the blog post here), we showed how to optimally adjust any trained predictor to prevent one particular formal notion of discrimination, and the paper illustrated this with a case study based on FICO credit scores. To make this work more accessible, we also created a visualization to help illustrate and interactively explore the concepts from the paper.

TensorFlow

In November 2015, we open-sourced an initial version of TensorFlow so that the rest of the machine learning community could benefit from it and we could all collaborate to jointly improve it. In 2016, TensorFlow became the most popular machine learning project on GitHub, with over 10,000 commits by more than 570 people. TensorFlow's repository of models has grown with contributions from the community, and there are also more than 5000 TensorFlow-related repositories listed on GitHub alone! Furthermore, TensorFlow has been widely adopted by well-known research groups and large companies including DeepMind, and applied towards or some unusual applications like finding sea cows Down Under and sorting cucumbers in Japan.

We've made numerous performance improvements, added support for distributed training, brought TensorFlow to iOS, Raspberry Pi and Windows, and integrated TensorFlow with widely-used big data infrastructure. We've extended TensorBoard, TensorFlow's visualization system with improved tools for visualizing computation graphs and embeddings. We've also made TensorFlow accessible from Go, Rust and Haskell, released state-of-the-art image classification models, Wide and Deep and answered thousands of questions on GitHub, StackOverflow and the TensorFlow mailing list along the way. TensorFlow Serving simplifies the process of serving TensorFlow models in production, and for those working in the cloud, Google Cloud Machine Learning offers TensorFlow as a managed service.

Last November, we celebrated TensorFlow's one year anniversary as an open-source project, and presented a paper on the computer systems aspects of TensorFlow at OSDI, one of the premier computer systems research conferences. In collaboration with our colleagues in the compiler team at Google we've also been hard at work on a backend compiler for TensorFlow called XLA, an alpha version of which was recently added to the open-source release.

Machine Learning Community Involvement

We also strive to educate and mentor people in how to do machine learning and how to conduct research in this field. Last January, Vincent Vanhoucke, one of the research leads in the Brain team, developed and worked with Udacity to make available a free online deep learning course (blog announcement). We also put together TensorFlow Playground, a fun and interactive system to help people better understand and visualize how very simple neural networks learn to accomplish tasks.

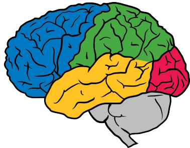
In June we welcomed our first class of 27 Google Brain Residents, selected from more than 2200 applicants, and in seven months they have already conducted significantly original research, helping to author 21 research papers. In August, many Brain team members took part in a Google Brain team Reddit AMA (Ask Me Anything) on r/MachineLearning to answer the community's questions about machine learning and our team. Throughout the year, we also hosted 46 student interns (mostly Ph.D. students) in our group to conduct research and work with our team members.

Spreading Machine Learning within Google

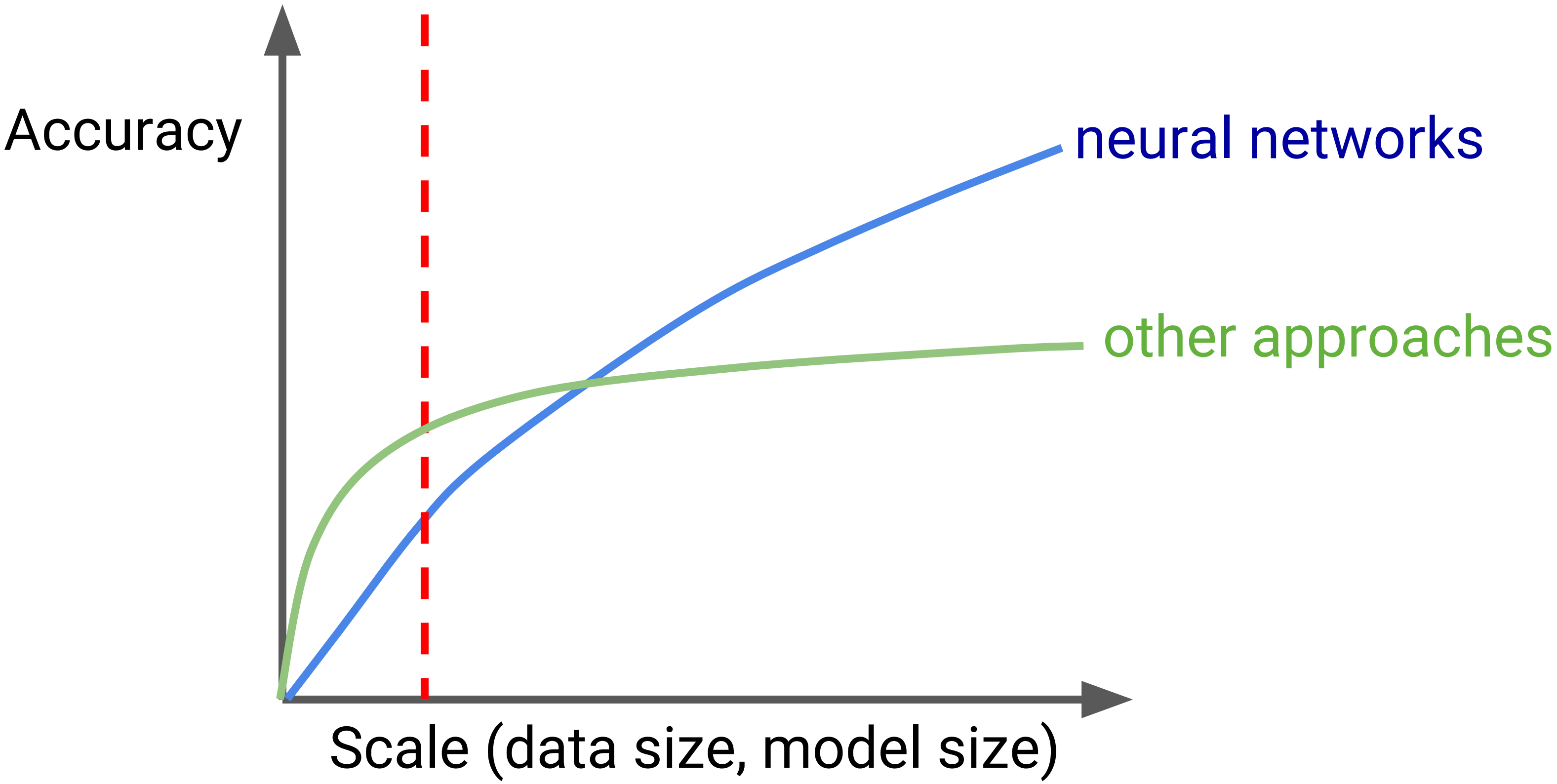
In addition to the public-facing activities outlined above, we have continued to work within Google to spread machine learning expertise and awareness throughout our many product teams, and to ensure that the company as a whole is well positioned to take advantage of any new machine learning research that emerges. As one example, we worked closely with our platforms team to provide specifications and high level goals for Google's Tensor Processing Unit (TPU), a custom machine learning accelerator ASIC that was discussed at Google I/O. This custom chip provides an order of magnitude improvement for machine learning workloads, and is heavily used throughout our products, including for RankBrain, for the recently launched Neural Machine Translation system, and for the AlphaGo match against Lee Sedol in Korea last March.

All in all, 2016 was an exciting year for the Google Brain team and our many collaborators and colleagues both within and outside of Google, and we look forward to our machine learning research having significant impact in 2017!

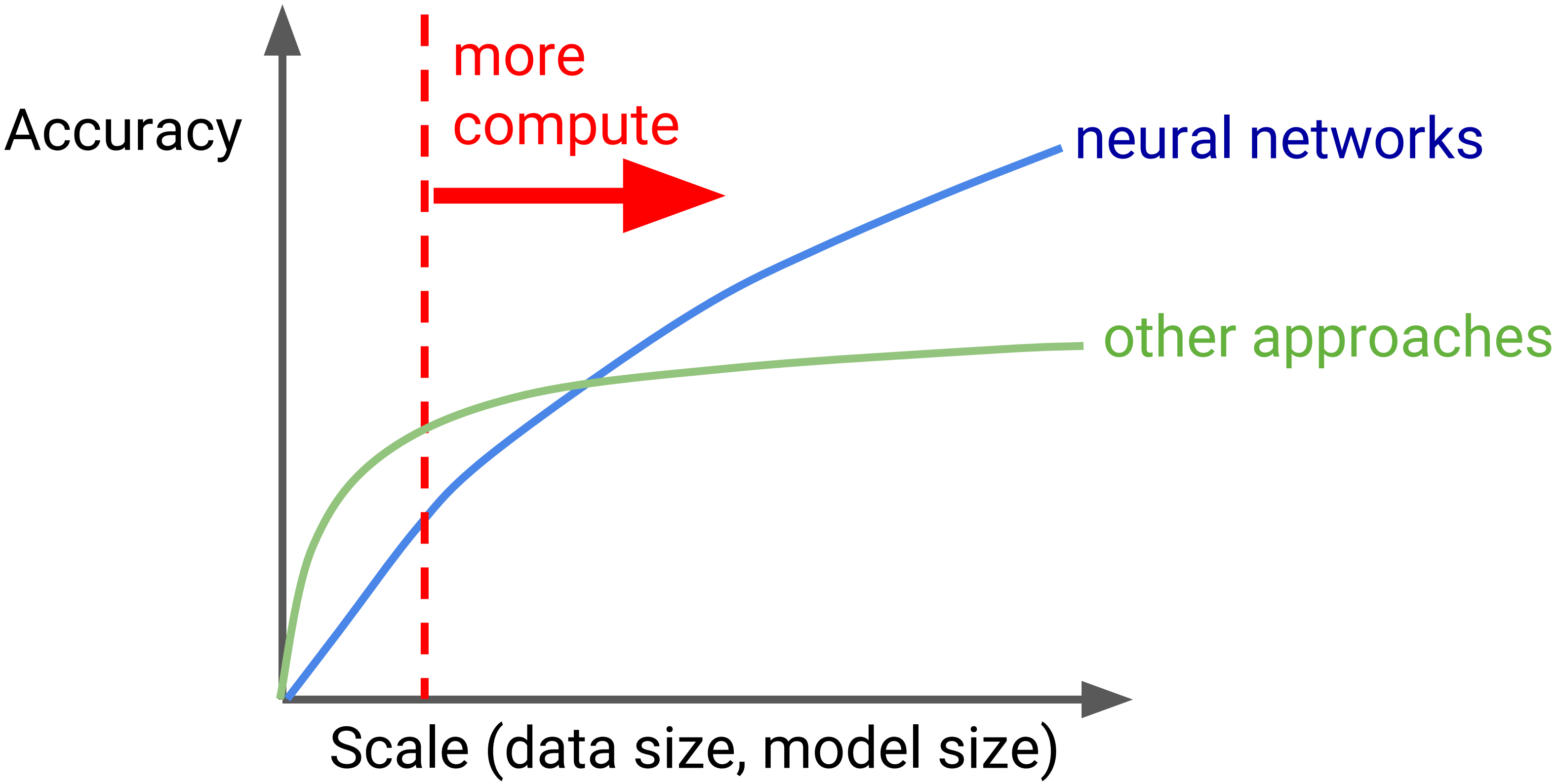
research.googleblog.com/2017/01/the-google-brain-team-looking-back-on.html

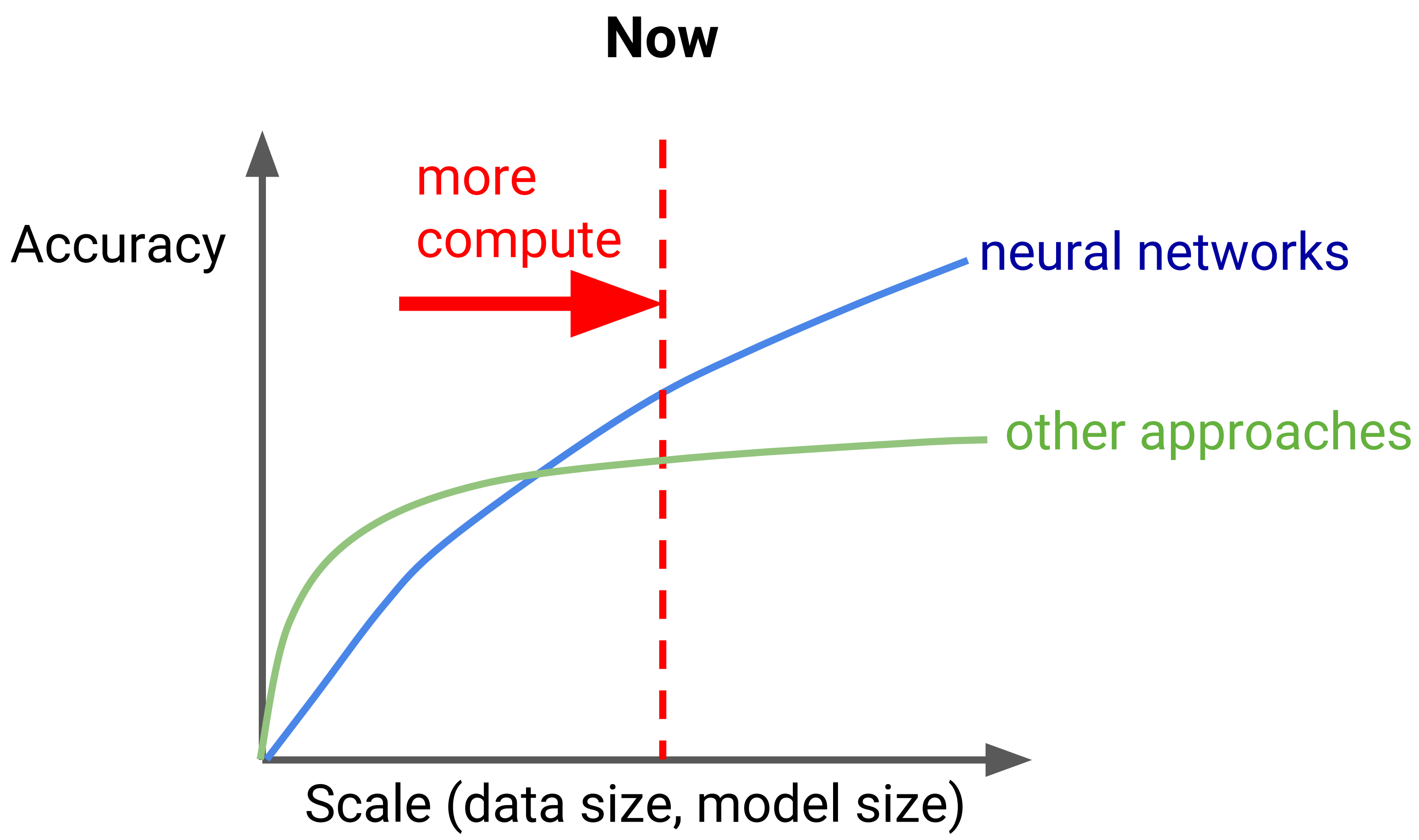


1980s and 1990s



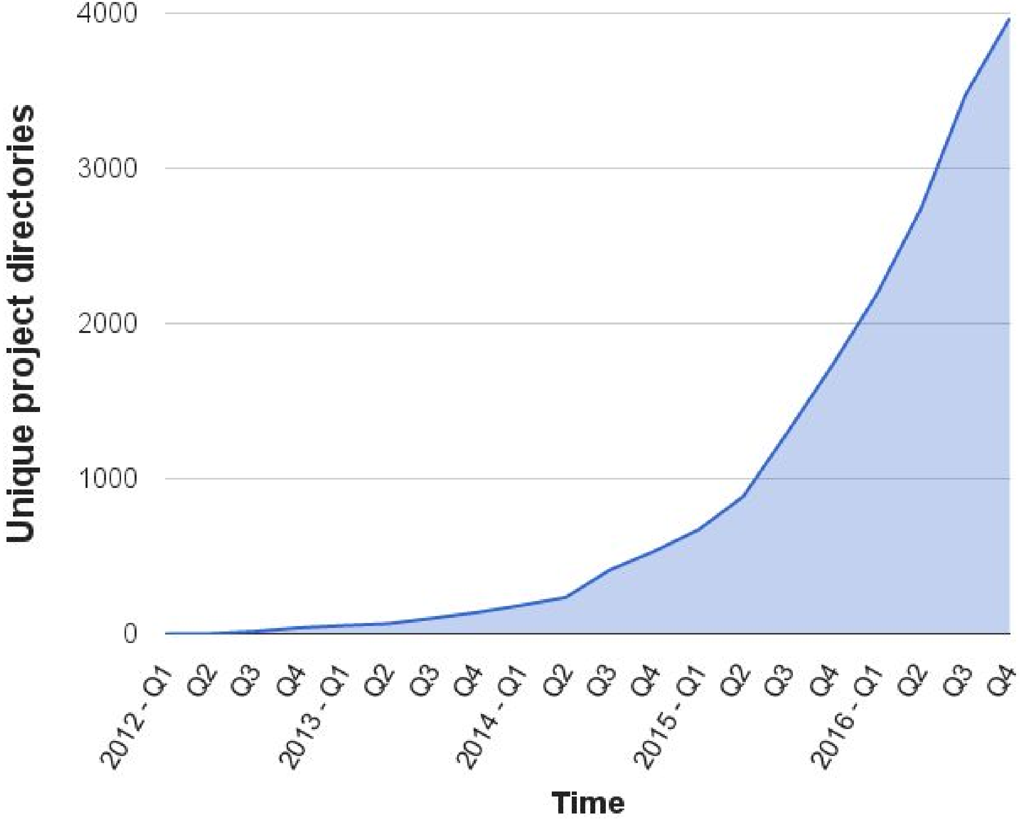
1980s and 1990s




Now
more
compute
Accuracy
neural networks
other approaches
Scale (data size, model size)

Growing Use of Deep Learning at Google

of directories containing model description files



Across many products/areas:

Android
Apps
drug discovery
Gmail
Image understanding
Maps
Natural language understanding
Photos
Robotics research
Speech
Translation
YouTube
… many others ...

Experiment Turnaround Time and Research Productivity

- **Minutes, Hours**:
  - **Interactive research! Instant gratification!**
- **1-4 days**
  - Tolerable
  - Interactivity replaced by running many experiments in parallel
- **1-4 weeks**
  - High value experiments only
  - Progress stalls
- **>1 month**
  - Don’t even try

Build the right tools

TensorFlow

http://tensorflow.org/

and

https://github.com/tensorflow/tensorflow

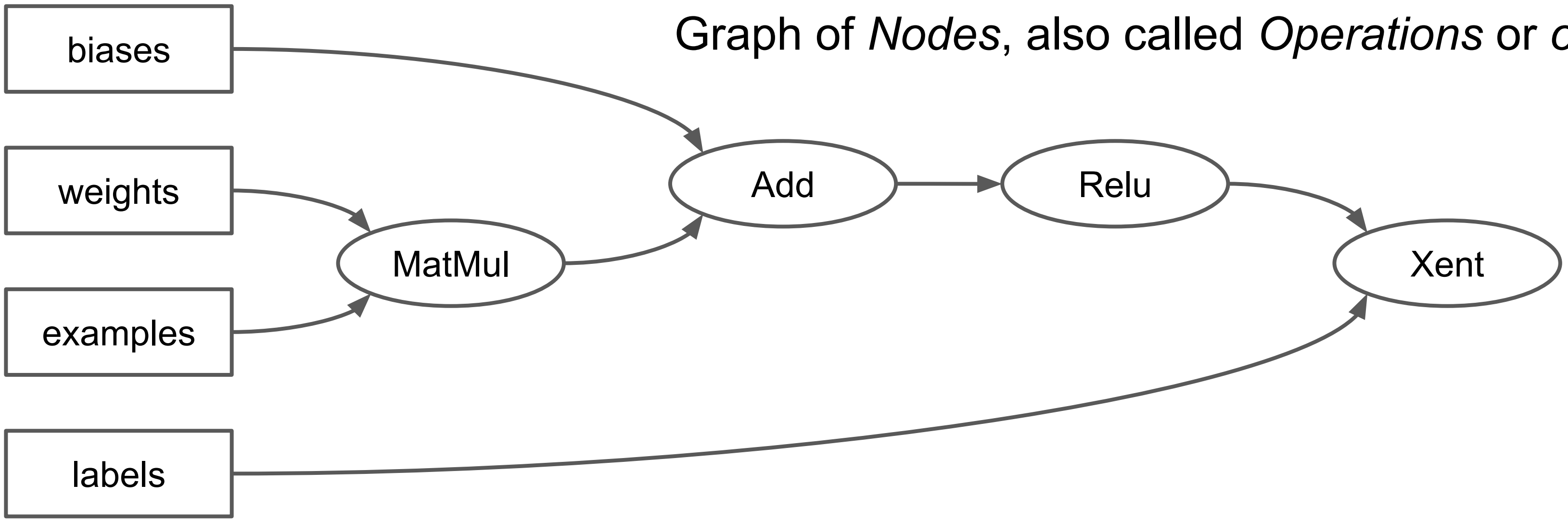
Open, standard software for general machine learning

Great for Deep Learning in particular

First released Nov 2015

Apache 2.0 license

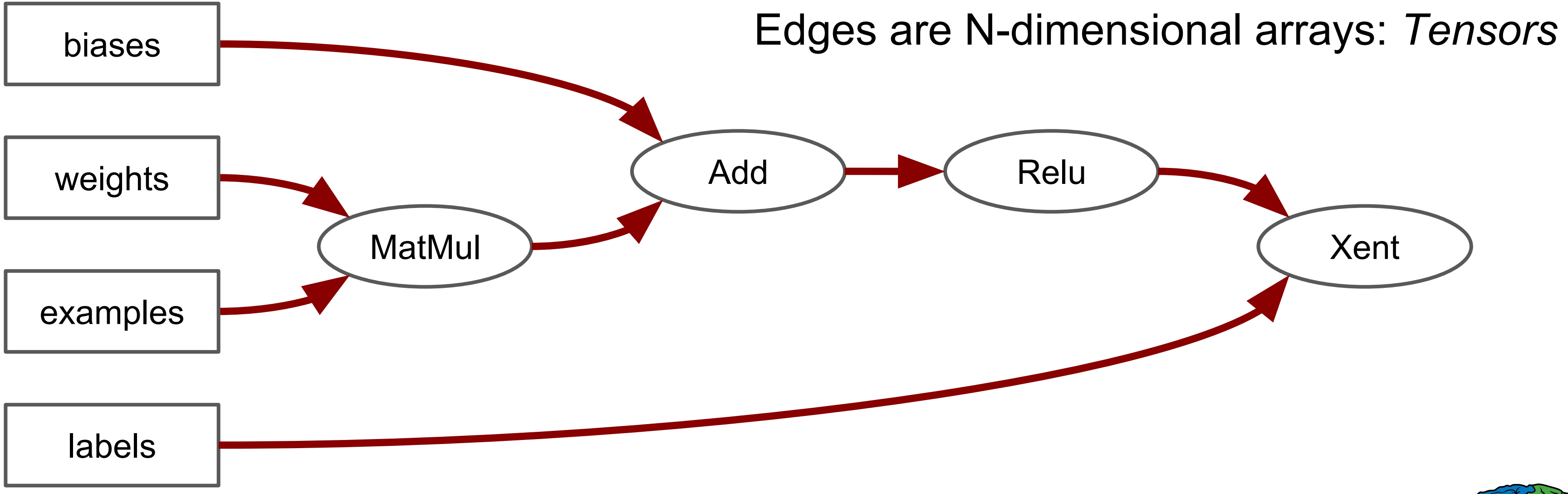
Computation is a dataflow graph



Graph of *Nodes*, also called *Operations* or *ops.*

Computation is a dataflow graph

with tensors



Example TensorFlow fragment

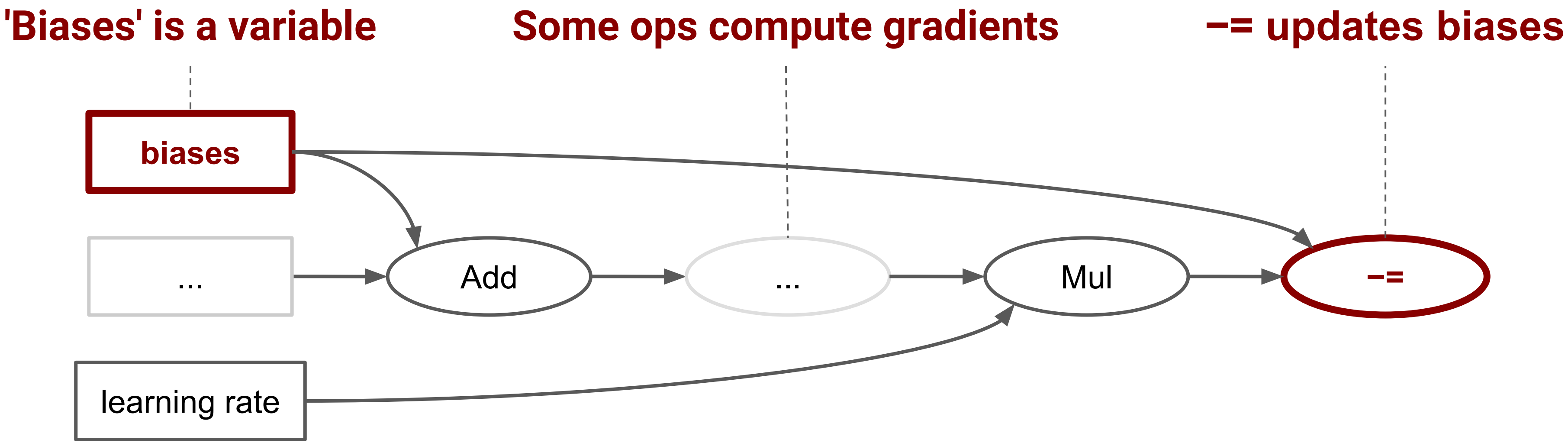
- Build a graph computing a neural net inference.

```
import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data

mnist = input_data.read_data_sets('MNIST_data', one_hot=True)
x = tf.placeholder("float", shape=[None, 784])
W = tf.Variable(tf.zeros([784,10]))
b = tf.Variable(tf.zeros([10]))
y = tf.nn.softmax(tf.matmul(x, W) + b)
```

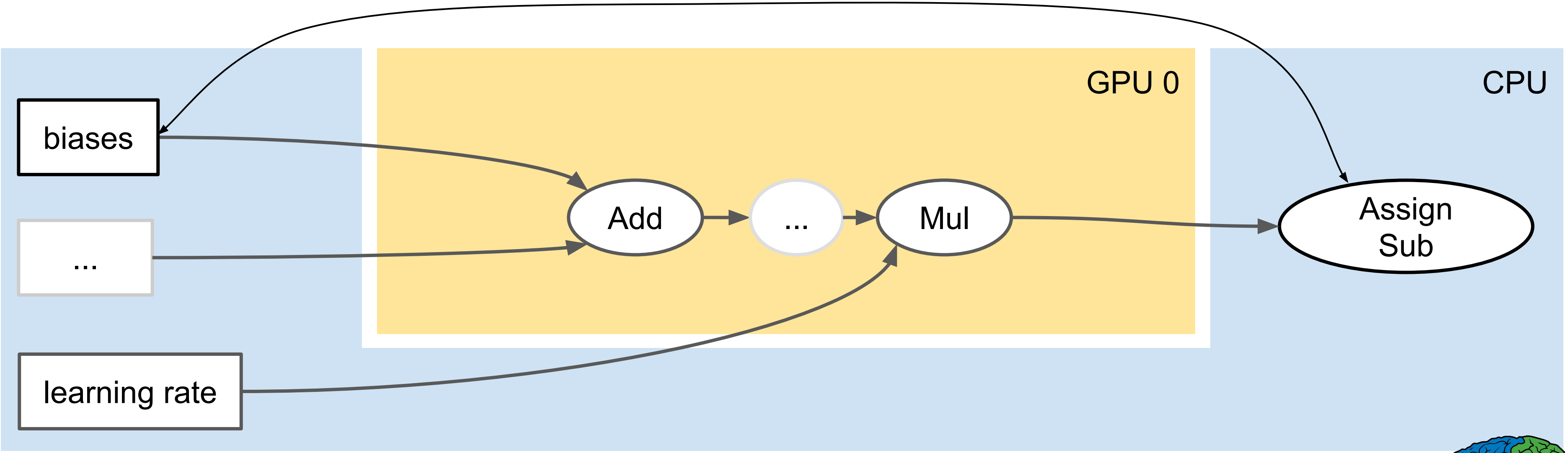
Computation is a dataflow graph

with state



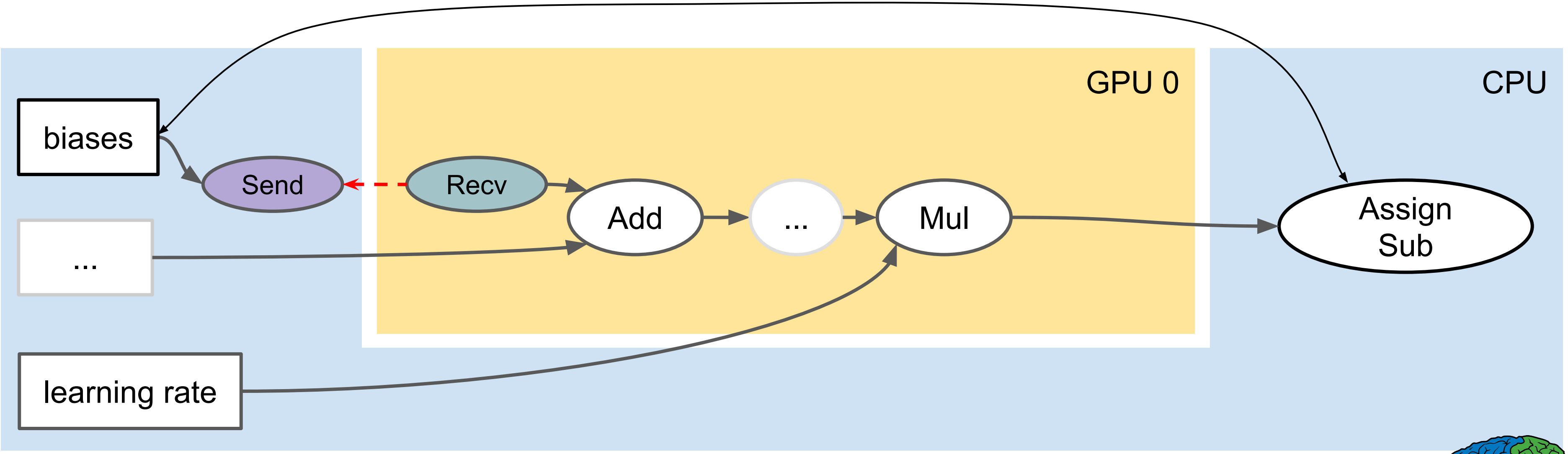
Computation is a dataflow graph

distributed



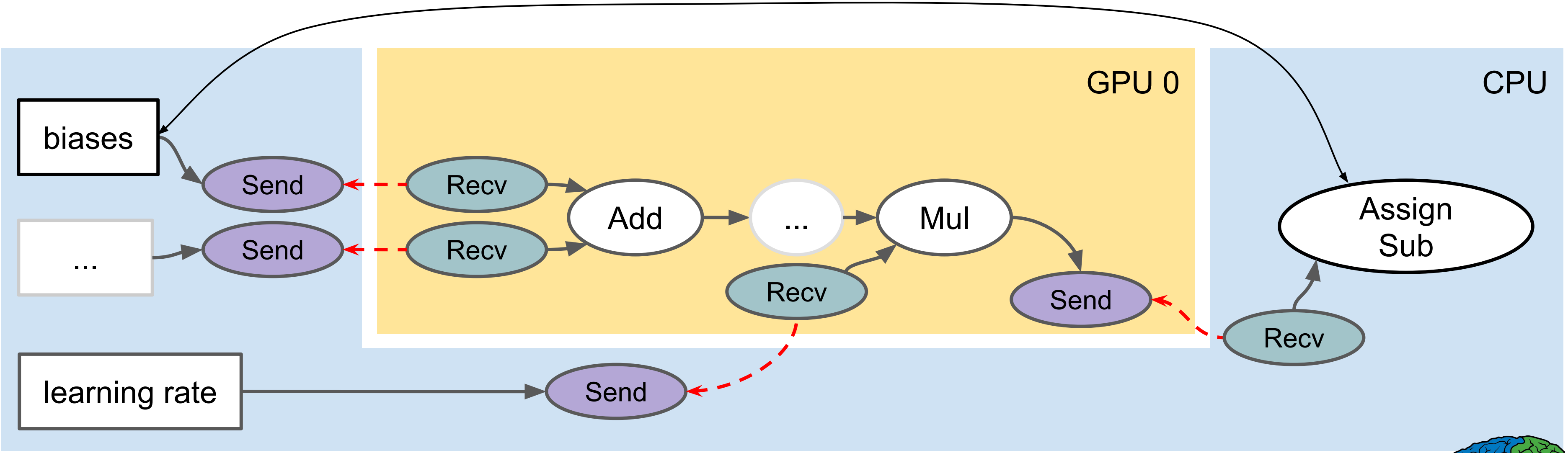
Assign *Devices* to Ops

- TensorFlow inserts *Send/Recv* Ops to transport tensors across devices
- *Recv* ops pull data from *Send* ops



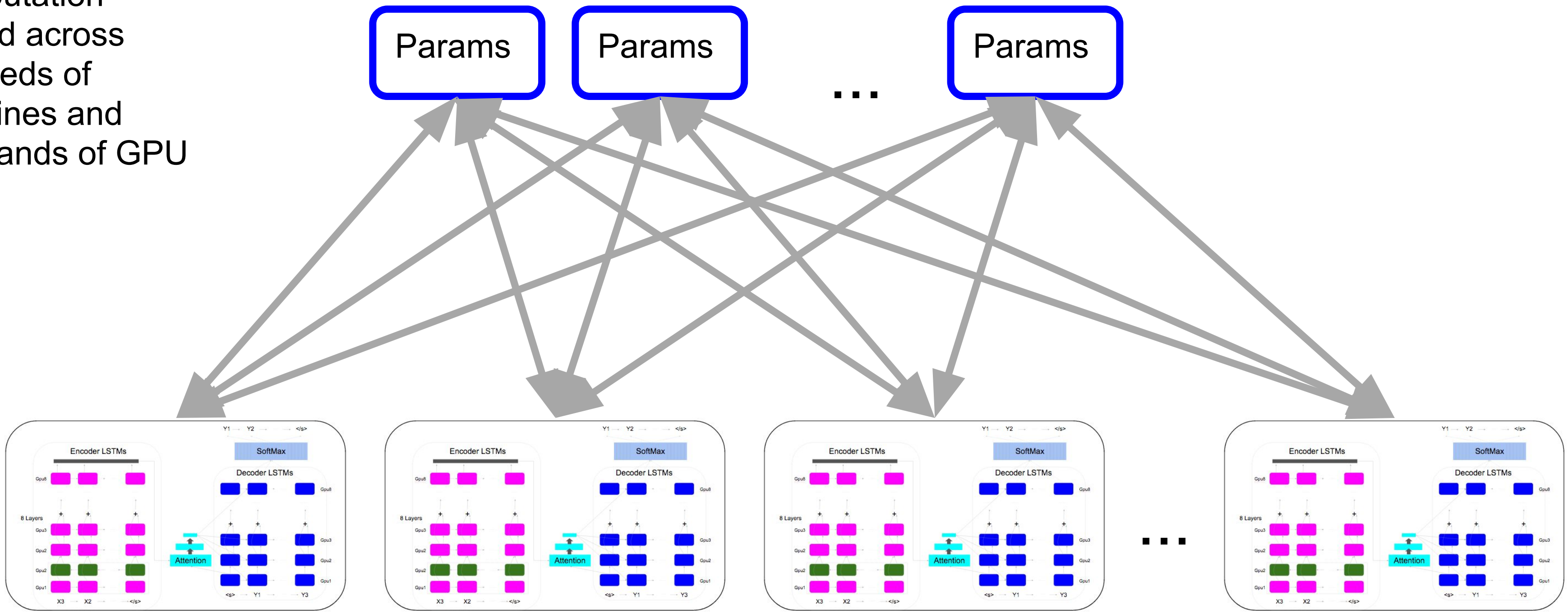
Assign *Devices* to Ops

- TensorFlow inserts *Send/Recv* Ops to transport tensors across devices
- *Recv* ops pull data from *Send* ops



Same mechanism supports large distributed systems

Computation spread across hundreds of machines and thousands of GPU cards



Many replicas

TensorFlow:
Large-Scale Machine Learning on Heterogeneous Distributed Systems
(Preliminary White Paper, November 9, 2015)

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro,
Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow,
Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser,
Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray,
Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar,
Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals,
Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng
Google Research*

http://tensorflow.org/whitepaper2015.pdf

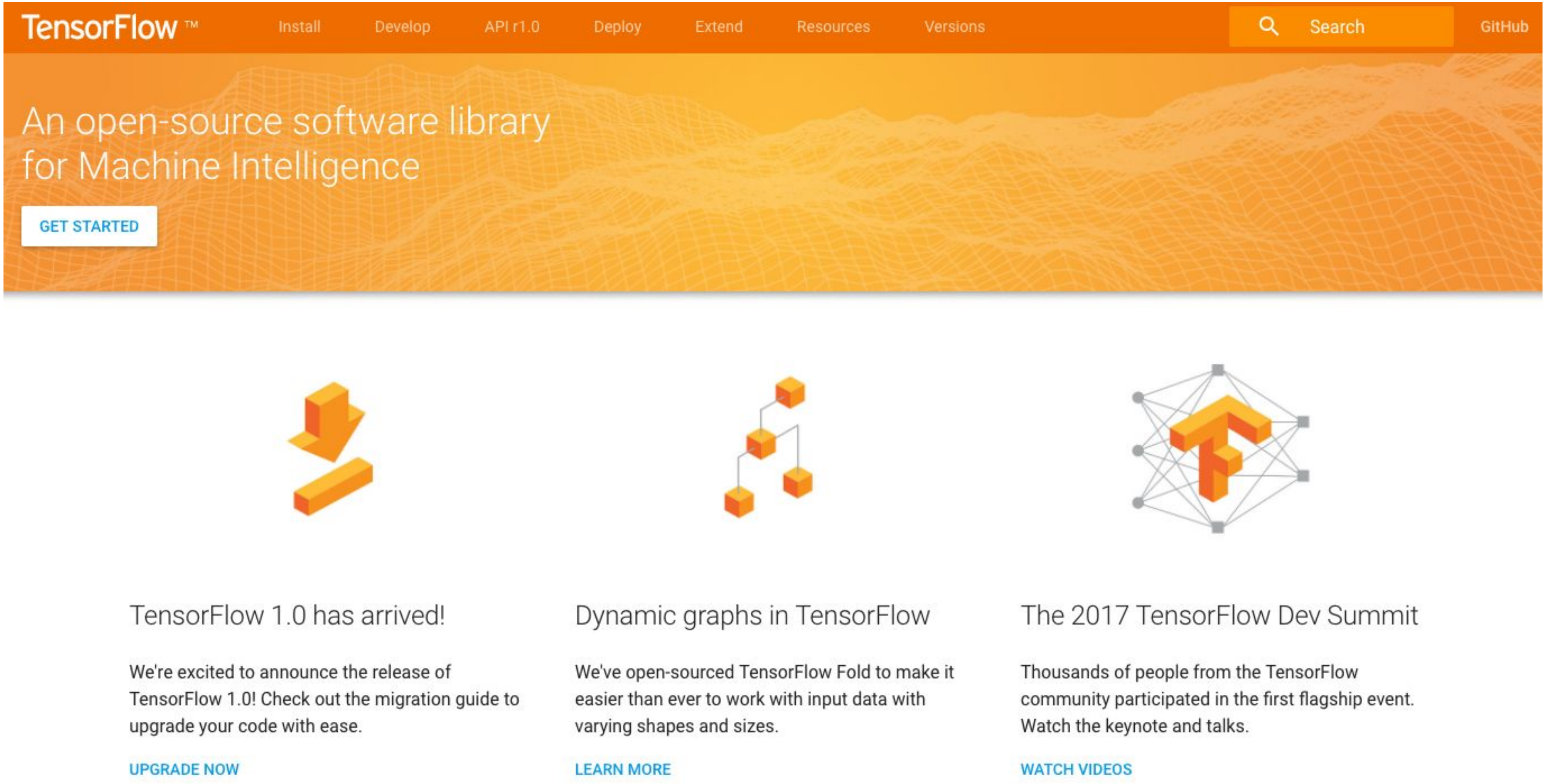
TensorFlow: A system for large-scale machine learning

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean,
Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur,
Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker,
Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng

Google Brain

Paper in OSDI 2016

https://arxiv.org/abs/1605.08695



http://tensorflow.org/

Why Did We Build TensorFlow?

Wanted system that was **flexible**, **scalable**, and **production-ready**

DistBelief, our first system, was good on two of these, but lacked **flexibility**

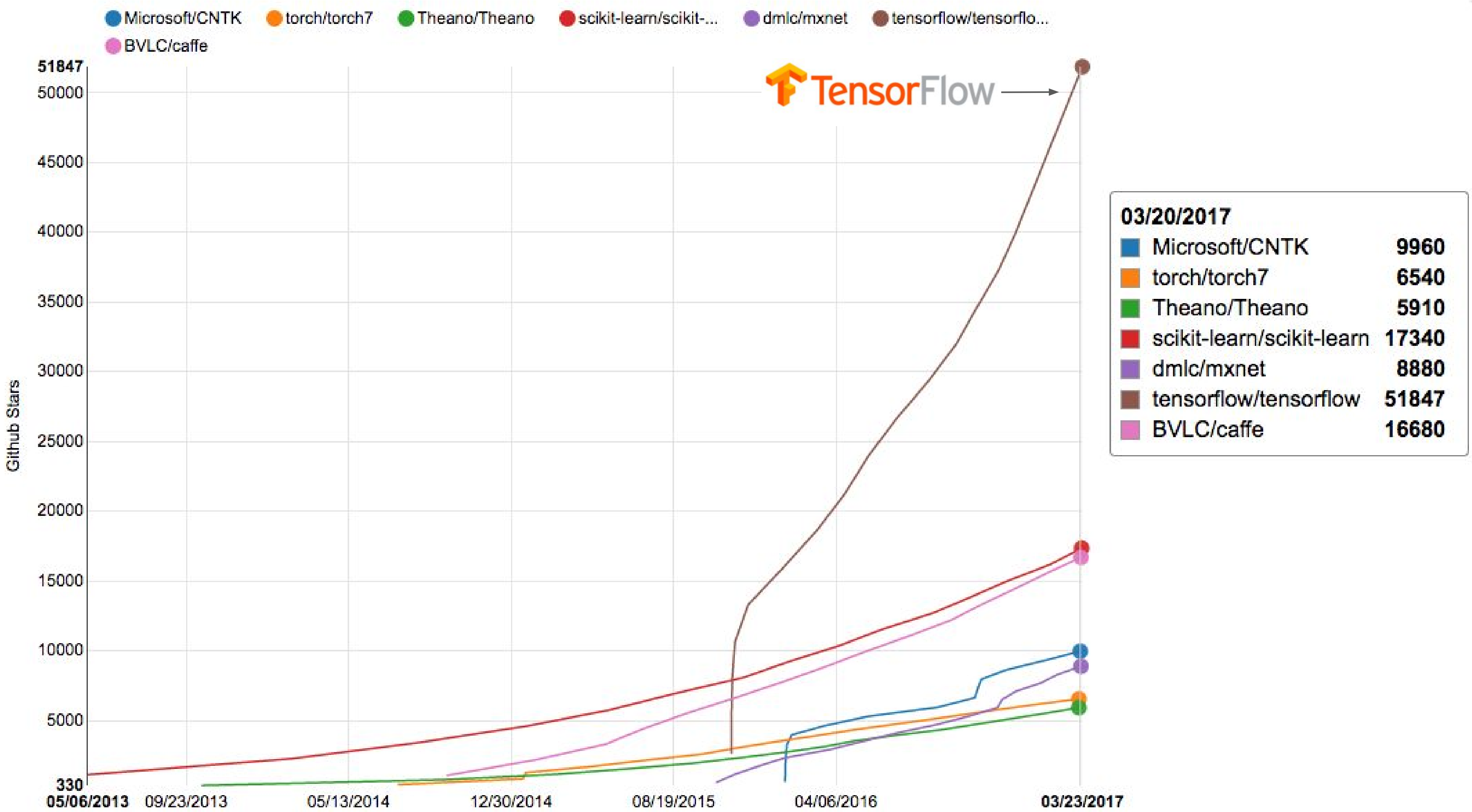
Most existing open-source packages were also good on 2 of 3 but not all 3

TensorFlow Goals

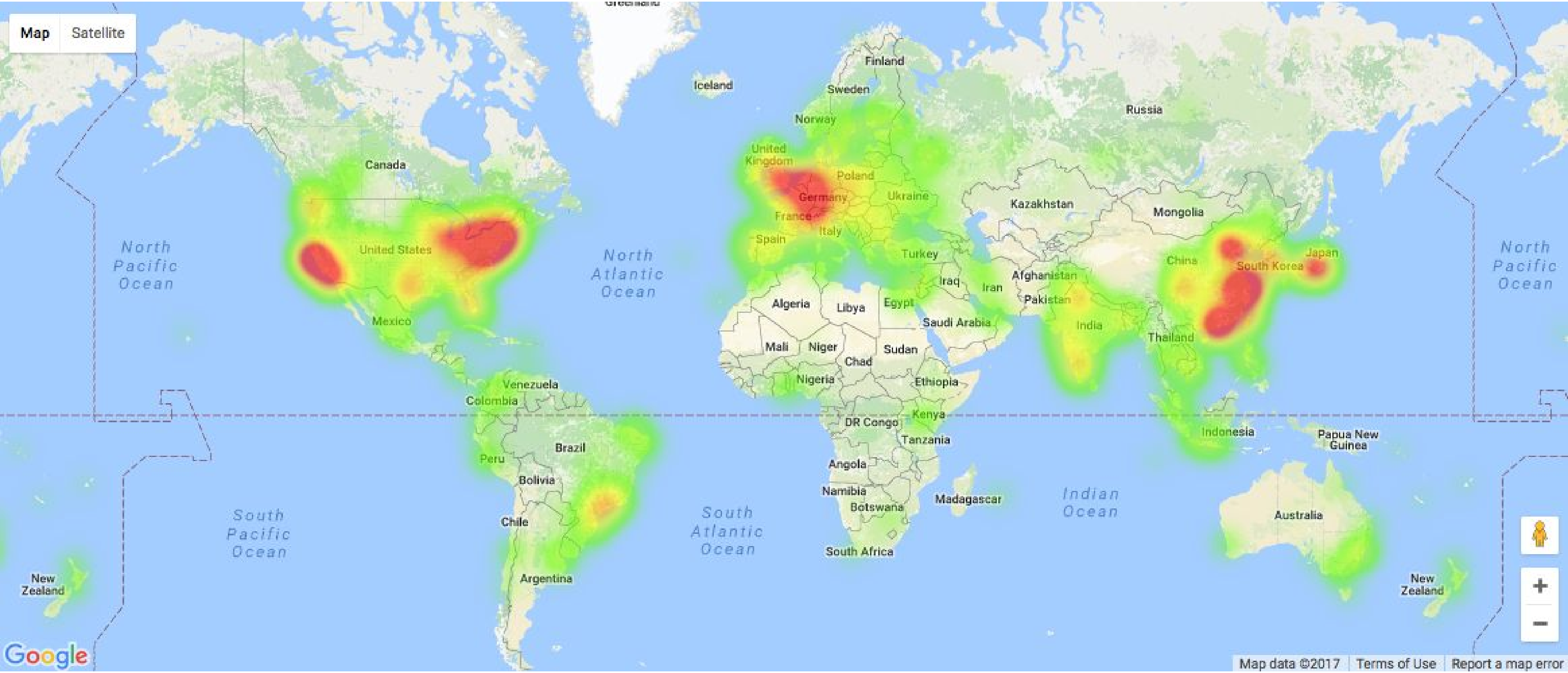
Establish **common platform** for expressing machine learning ideas and systems

Make this platform the **best in the world** for both research and production use

Open source it so that it becomes a **platform for everyone**, not just Google


Microsoft/CNTK
torch/torch7
Theano/Theano
scikit-learn/scikit-...
dmlc/mxnet
tensorflow/tensorflo...
BVLC/caffe
TensorFlow
Github Stars
51847
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
330
05/06/2013
09/23/2013
05/13/2014
12/30/2014
08/19/2015
04/06/2016
03/23/2017
03/20/2017
Microsoft/CNTK 9960
torch/torch7 6540
Theano/Theano 5910
scikit-learn/scikit-learn 17340
dmlc/mxnet 8880
tensorflow/tensorflow 51847
BVLC/caffe 16680

ML is done in many places



TensorFlow GitHub stars by GitHub user profiles w/ public locations
Source: http://jrvis.com/red-dwarf/?user=tensorflow&repo=tensorflow

Progress

v0.5

Initial Release

v0.7

TensorFlow Serving

v0.9

iOS; Mac GPU

v0.11

HDFS; CUDA 8, CuDNN 5

Nov '15 | Dec '15 | Feb '16 | Apr '16 | Jun '16 | Aug '16 | Oct '16 | Nov '16

v0.6

Faster on GPUs; Python 3.3+

v0.8

Distributed TensorFlow

v0.10

Slim

v0.12

Windows 7, 10, and Server 2016; TensorBoard Embedding Visualizer

https://github.com/tensorflow/tensorflow/releases

Progress

Nov '15	v0.5	Initial Release
Dec '15	v0.6	Faster on GPUs; Python 3.3+
Feb '16	v0.7	TensorFlow Serving
Apr '16	v0.8	Distributed TensorFlow
Jun '16	v0.9	iOS; Mac GPU
Aug '16	v0.10	Slim
Oct '16	v0.11	HDFS; CUDA 8, CuDNN 5
Nov '16	v0.12	Windows 7, 10, and Server 2016; TensorBoard Embedding Visualizer

v1.0 released in Feb. 2017

https://github.com/tensorflow/tensorflow/releases

TensorFlow: A Vibrant Open-Source Community

- **Rapid development, many outside contributors**
  - 475+ non-Google contributors to TensorFlow 1.0
  - 15,000+ commits in 15 months
  - Many community created tutorials, models, translations, and projects
    - ~7,000 GitHub repositories with ‘TensorFlow’ in the title

- **Direct engagement between community and TensorFlow team**
  - 5000+ Stack Overflow questions answered
  - 80+ community-submitted GitHub issues responded to weekly

- **Growing use in ML classes: Toronto, Berkeley, Stanford, ...**

TensorFlow ™ Get Started Guides API r0.12 Mobile Resources Versions Search GitHub

Guides

TUTORIALS HOW TO

Tutorials

Basic Neural Networks

The first few Tensorflow tutorials guide you through training and testing a simple neural network to classify handwritten digits from the MNIST database of digit images.

MNIST For ML Beginners

If you're new to machine learning, we recommend starting here. You'll learn about a classic problem, handwritten digit classification (MNIST), and get a gentle introduction to multiclass classification.

View Tutorial

Deep MNIST for Experts

If you're already familiar with other deep learning software packages, and are already familiar with MNIST, this tutorial will give you a very brief primer on TensorFlow.

View Tutorial

TensorFlow Mechanics 101

This is a technical tutorial, where we walk you through the details of using TensorFlow infrastructure to train models at scale. We use MNIST as the example.

View Tutorial

Easy ML with tf.contrib.learn

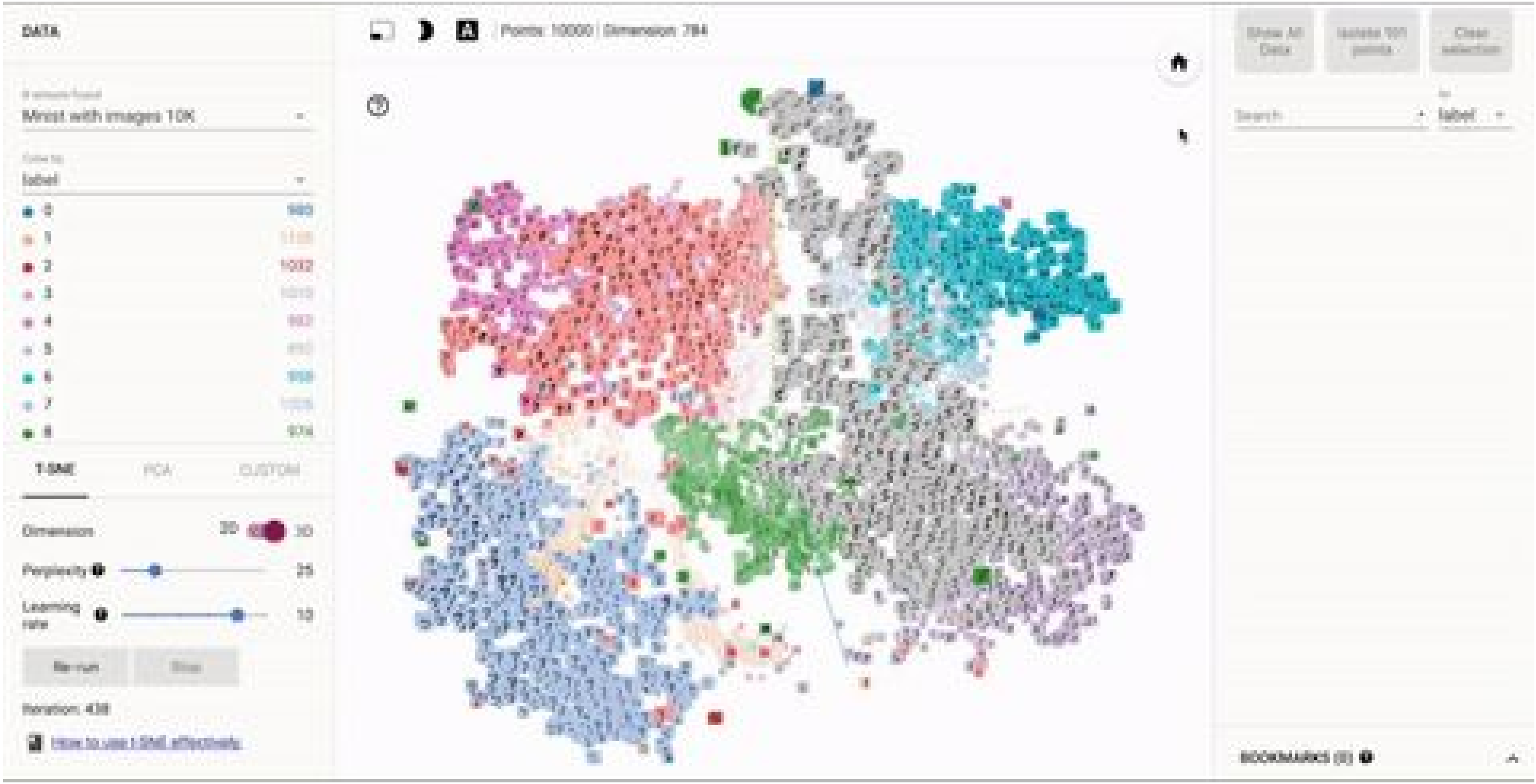
tf.contrib.learn Quickstart

A quick introduction to tf.contrib.learn, a high-level API for TensorFlow. Build, train, and evaluate a neural network with just a few lines of code.

View Tutorial

Contents

tensorflow.org/tutorials


DATA
Points: 10000 | Dimension: 784
Mnist with images 10K
label
T-SNE
PCA
CUSTOM
Dimension
2D
3D
Perplexity
25
Learning rate
10
Re-run
Stop
Iteration: 438
How to use t-SNE effectively
Show All Data
Clear selection
Search
label
BOOKMARKS (0)

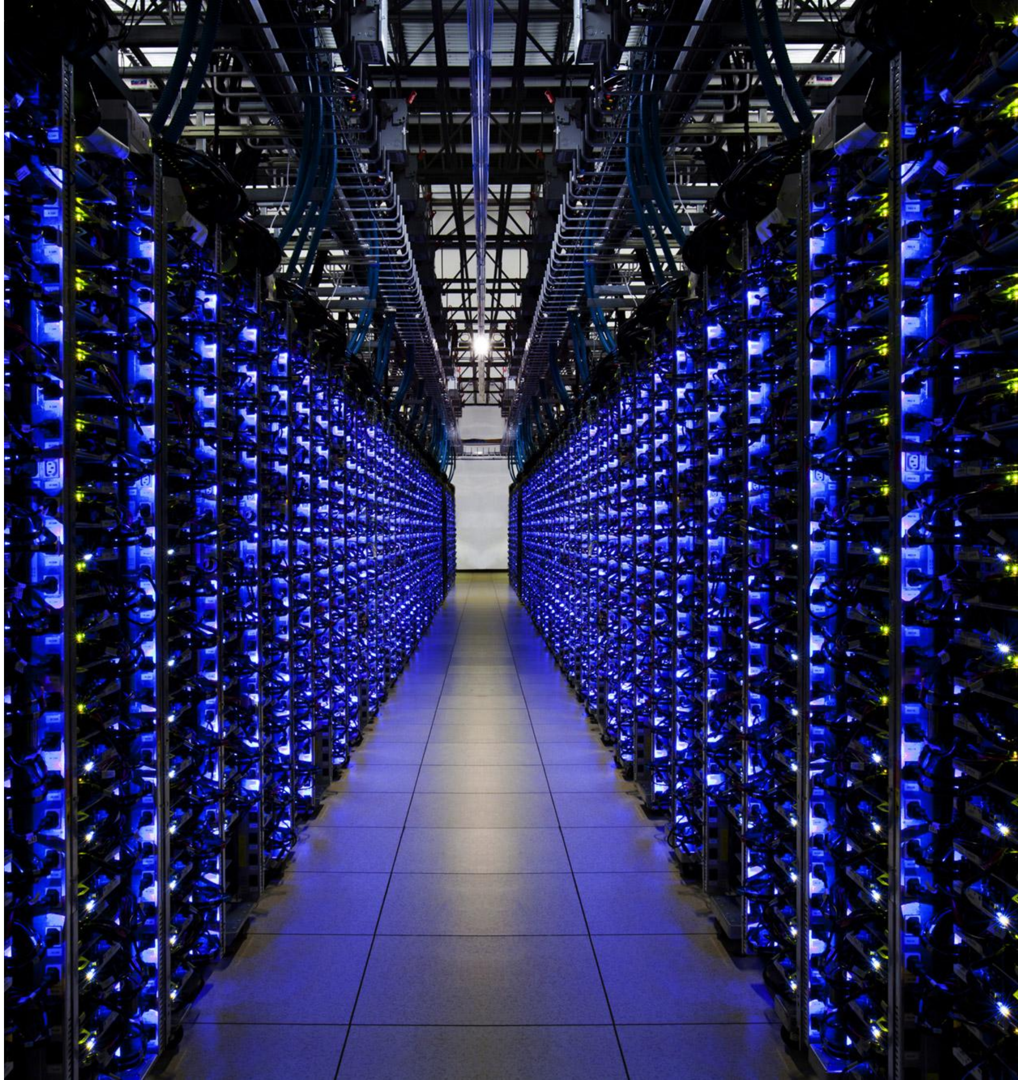
Performance matters

Research

- Iterate quickly
- Train models faster
- Run more experiments in parallel

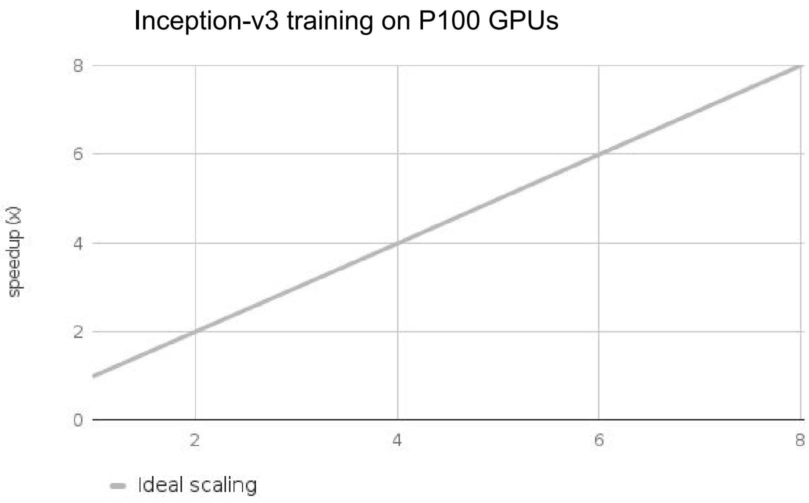
Production

- Server farms and embedded
- CPUs, GPUs, TPUs, and more
- Low-latency serving

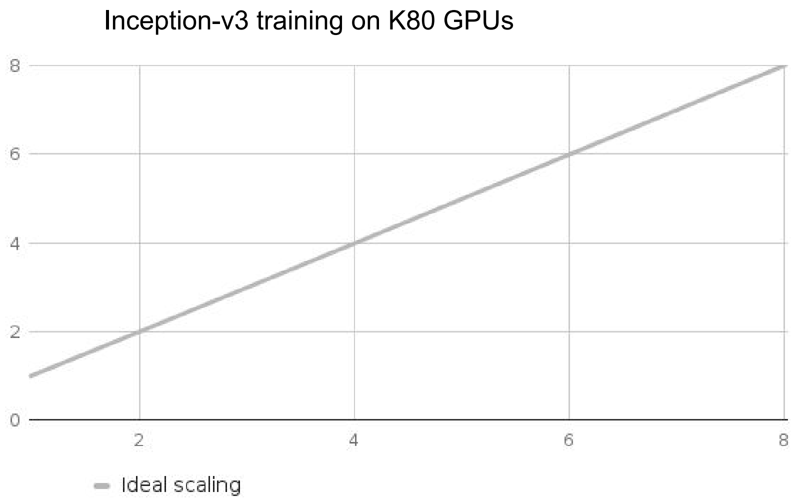


TensorFlow v1.0 Performance
Inception-v3 Training - Ideal Scaling Synthetic Data

DGX-1:



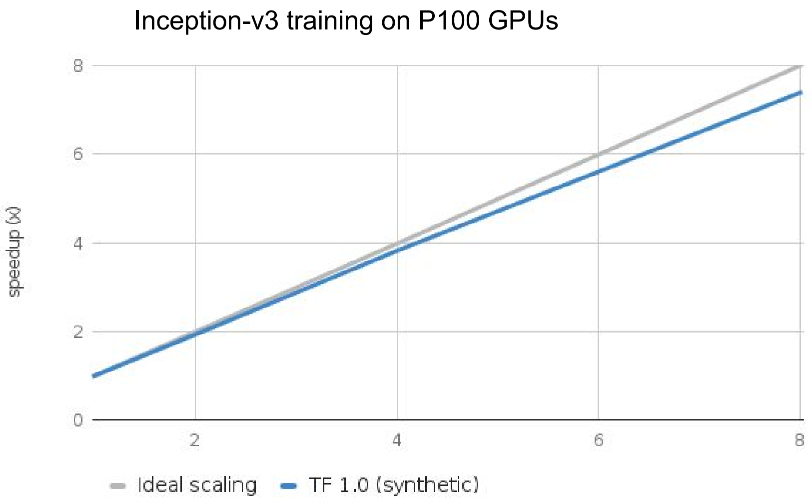
K80:

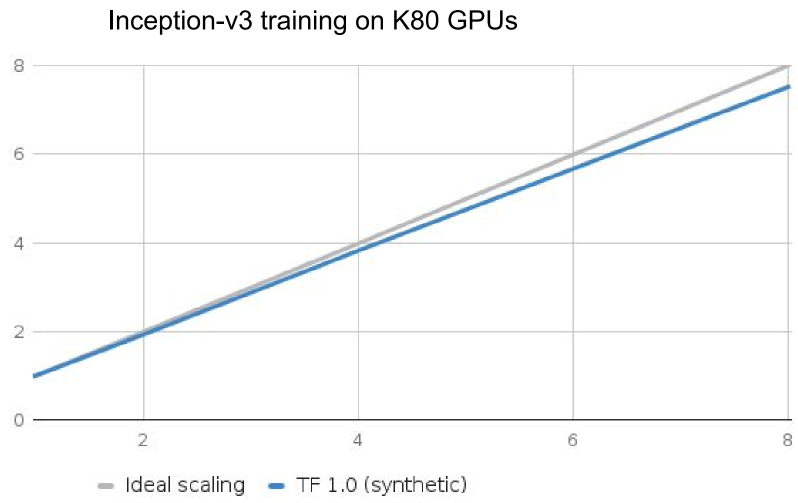


TensorFlow v1.0 Performance
Inception-v3 Training - Synthetic Data

DGX-1: 7.37x speedup at 8 GPUs

K80: 7.5x speedup at 8 GPUs

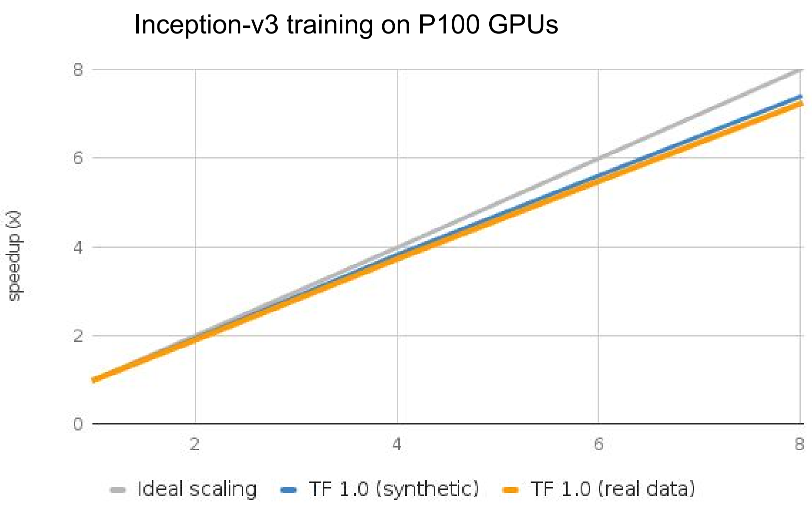


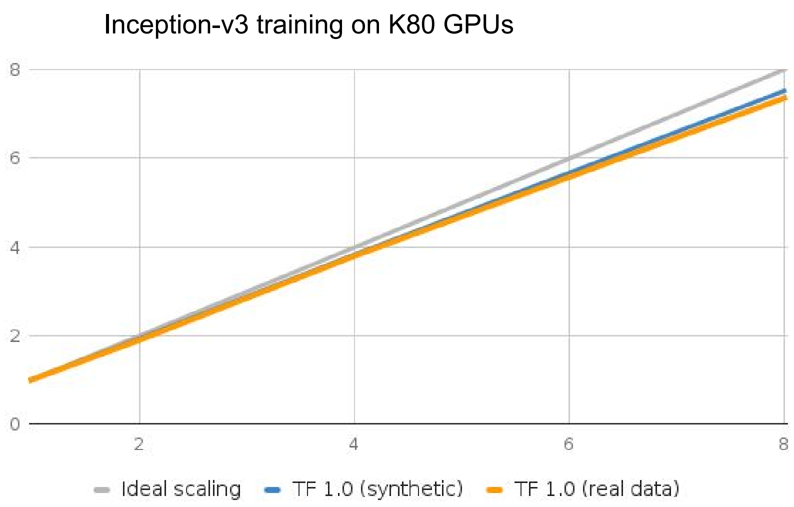


TensorFlow v1.0 Performance
Inception-v3 Training - Real Data

DGX-1: 7.2x speedup at 8 GPUs

K80: 7.3x speedup at 8 GPUs

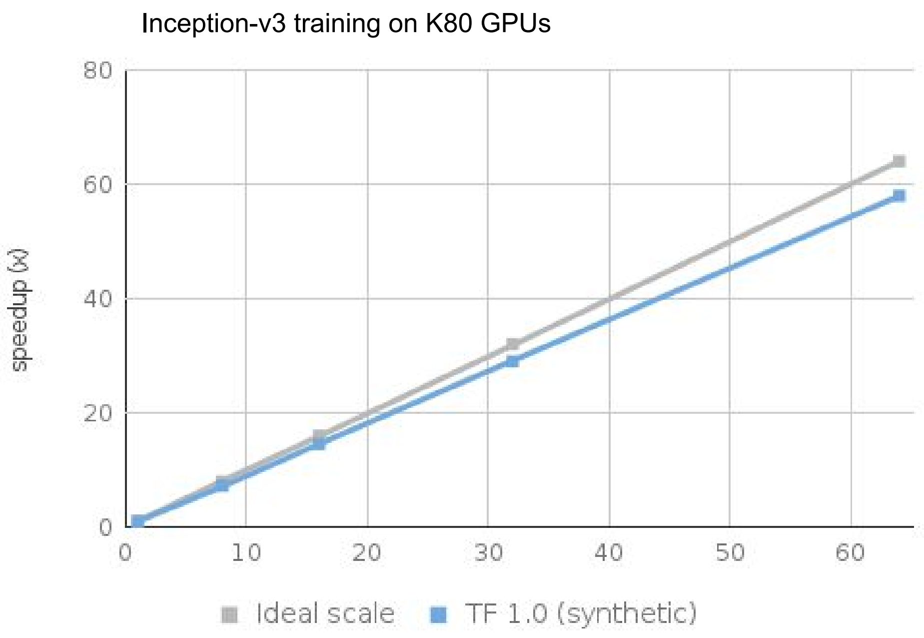




TensorFlow v1.0 Performance
Inception-v3 Distributed Training - Synthetic Data

58x speedup at 64 GPUs (8 Servers / 8 GPUs each)

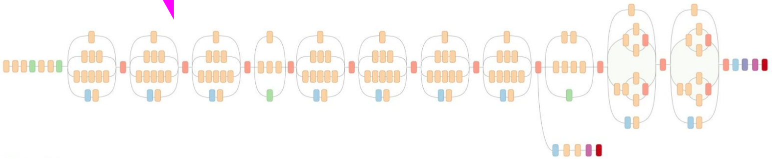
- GPU: K80
- Network: 20 Gb/sec



Just-In-Time Compilation

via XLA, "Accelerated Linear Algebra" compiler

TF graphs go in,

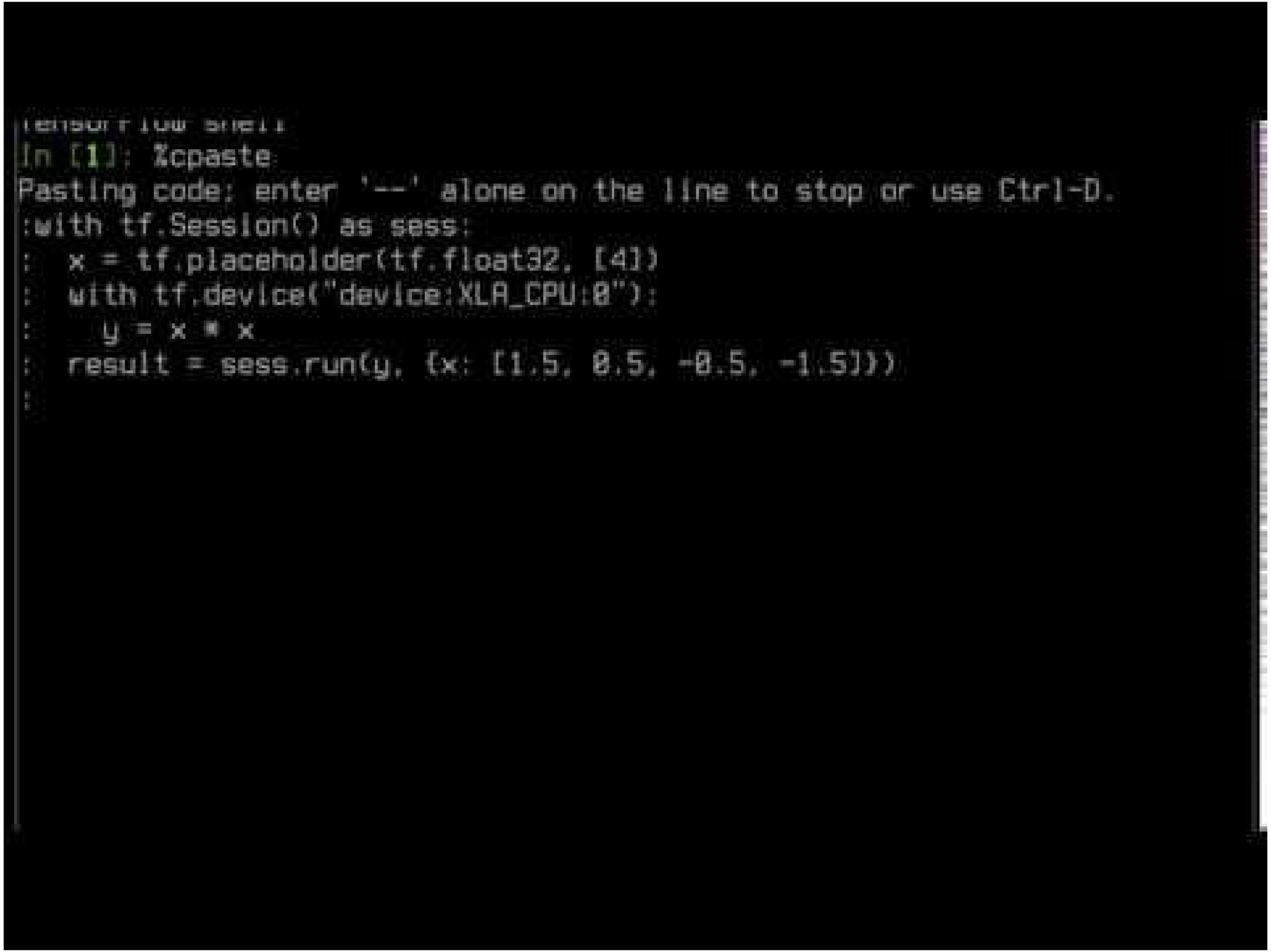


Optimized & specialized assembly comes out.

```
0x00000000      movq    (%rdx), %rax
0x00000003      vmovaps (%rax), %xmm0
0x00000007      vmulps  %xmm0, %xmm0, %xmm0
0x0000000b      vmovaps %xmm0, (%rdi)
                        ...
```

Let's explain that!

Demo:
Inspect **JIT code** in
TensorFlow
iPython shell



XLA:CPU

XLA:GPU

Computers can now see

Large implications for healthcare

MEDICAL IMAGING

Using similar model for detecting diabetic retinopathy in retinal images

JAMA The Journal of the American Medical Association

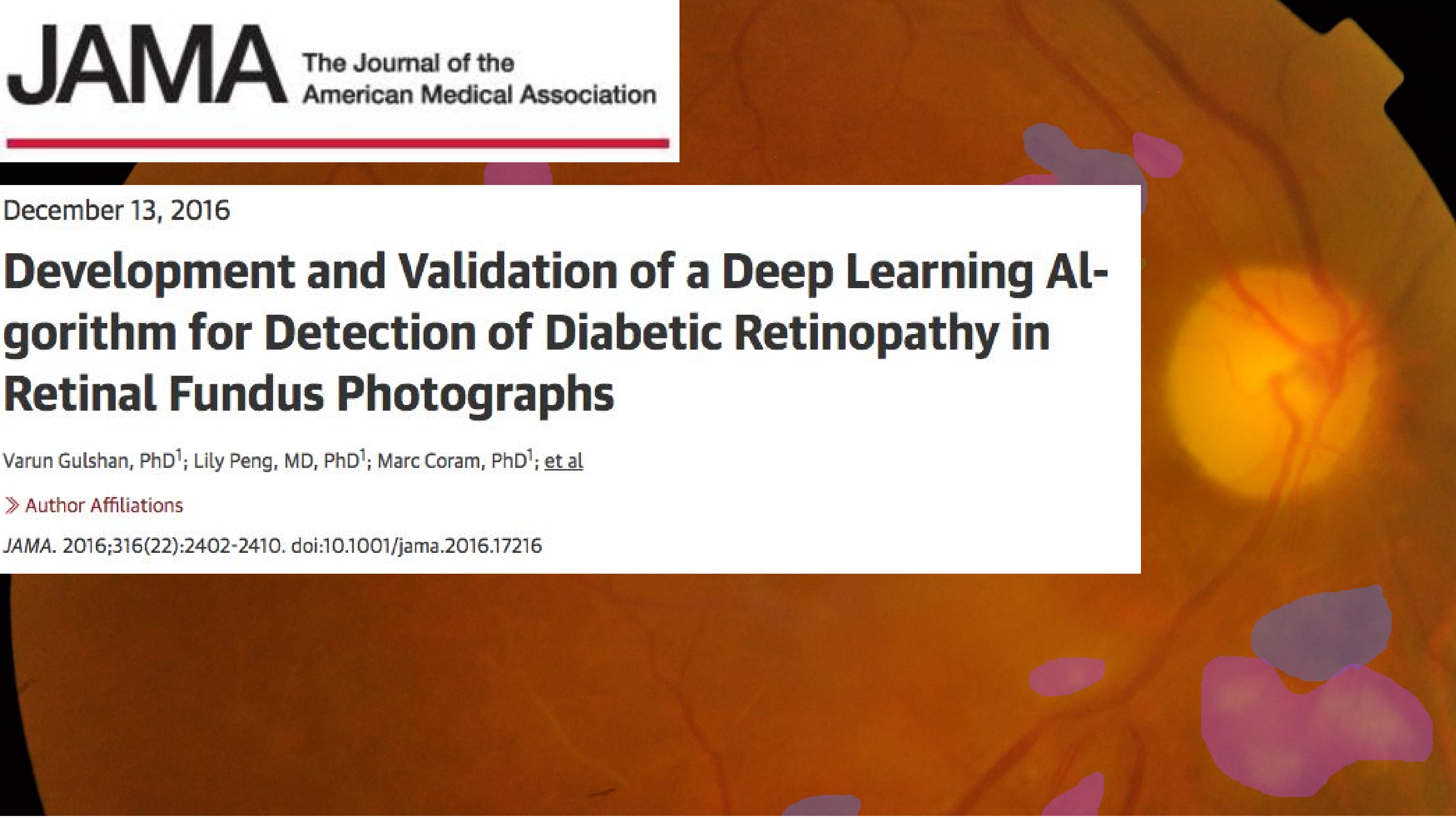
December 13, 2016

Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs

Varun Gulshan, PhD[1]; Lily Peng, MD, PhD[1]; Marc Coram, PhD[1]; et al

Author Affiliations

JAMA. 2016;316(22):2402-2410. doi:10.1001/jama.2016.17216



JAMA The Journal of the American Medical Association

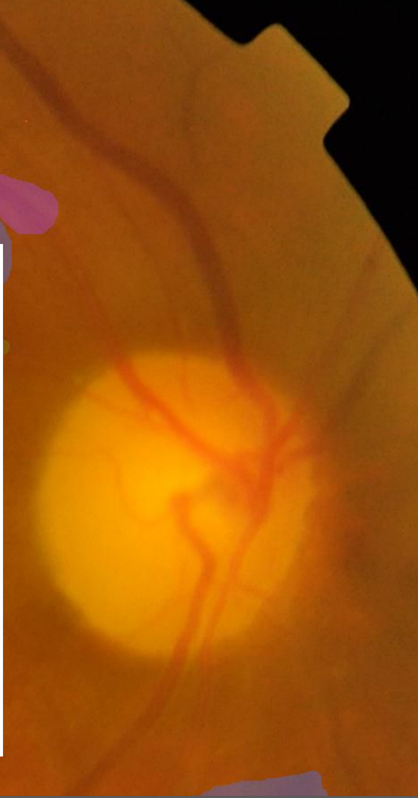
December 13, 2016

Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs

Varun Gulshan, PhD[1]; Lily Peng, MD, PhD[1]; Marc Coram, PhD[1]; et al

» Author Affiliations

JAMA. 2016;316(22):2402-2410. doi:10.1001/jama.2016.17216



Performance **on par or slightly better** than the median of 8 U.S. board-certified ophthalmologists (F-score of 0.95 vs. 0.91).

http://research.googleblog.com/2016/11/deep-learning-for-detection-of-diabetic.html

Detecting Cancer Metastases on Gigapixel Pathology Images

Yun Liu[1*], Krishna Gadepalli[1], Mohammad Norouzi[1], George E. Dahl[1], Timo Kohlberger[1], Aleksey Boyko[1], Subhashini Venugopalan[2**], Aleksei Timofeev[2], Philip Q. Nelson[2], Greg S. Corrado[1], Jason D. Hipp[3], Lily Peng[1], and Martin C. Stumpe[1]

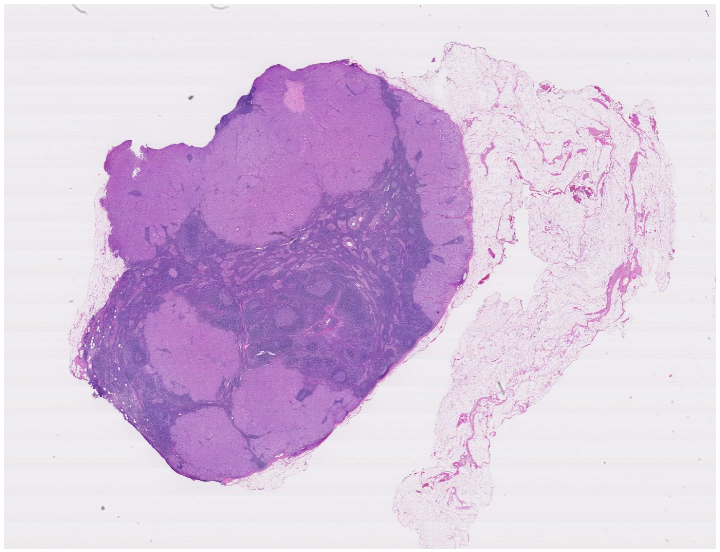
{liuyun,mnorouzi,gdahl,lhpeng,mstumpe}@google.com

[1]Google Brain, [2]Google Inc, [3]Verily Life Sciences, Mountain View, CA, USA

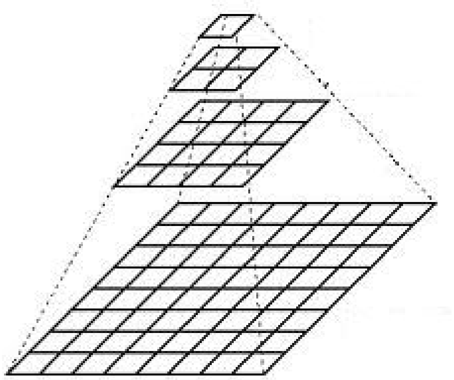
Blog: https://research.googleblog.com/2017/03/assisting-pathologists-in-detecting.html
Paper: https://arxiv.org/abs/1703.02442

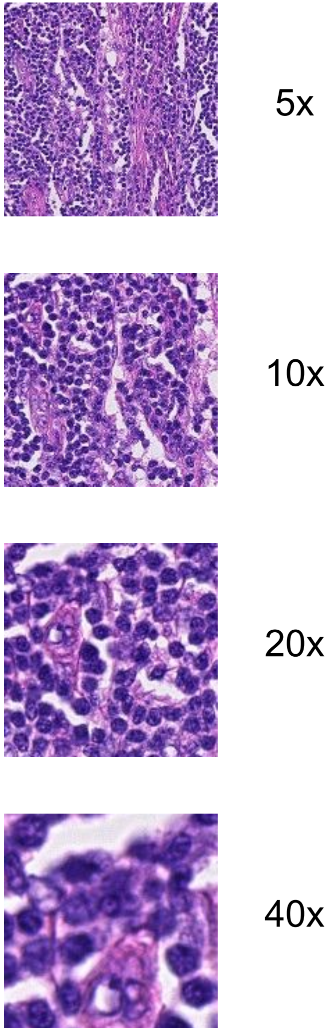
ML Challenges in Pathology

- ❑ Extremely large images (> 100k x 100k pixels)
- ❑ Multiscale problem - need detail as well as context



150k pixels (15 Gigapixel image)



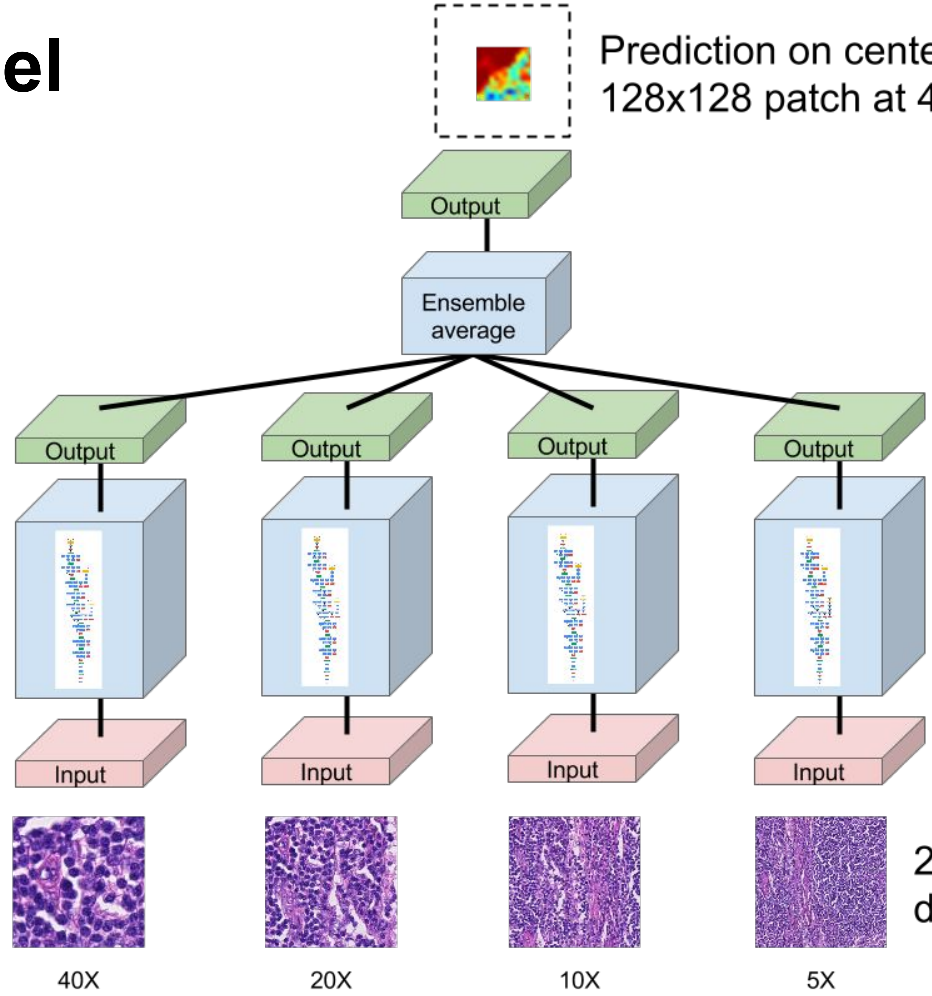


Multiscale model

Multi scale model

detail ←→ context

resembles microscope magnifications



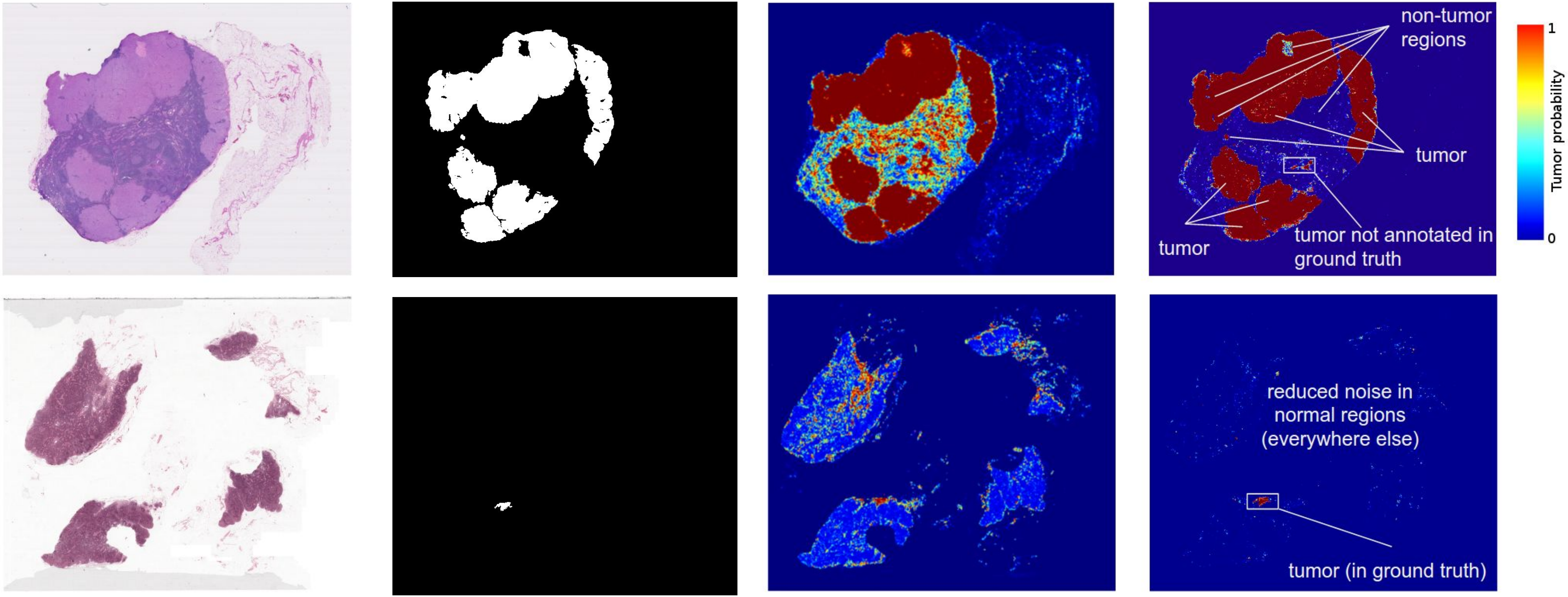
Prediction on center 128x128 patch at 40X

Independent InceptionV3 towers pre-trained on imagenet

299x299 patch at different scales

Detecting breast cancer metastases in lymph nodes

biopsy image | ground truth (from pathologist) | model prediction (early results) | model prediction (current results)



Model performance compared to pathologist

	our model	**pathologist***
Tumor localization score (FROC)	0.89	0.73
Sensitivity at 8 FP	0.92	0.73
Slide classification (AUC)	0.97	0.96

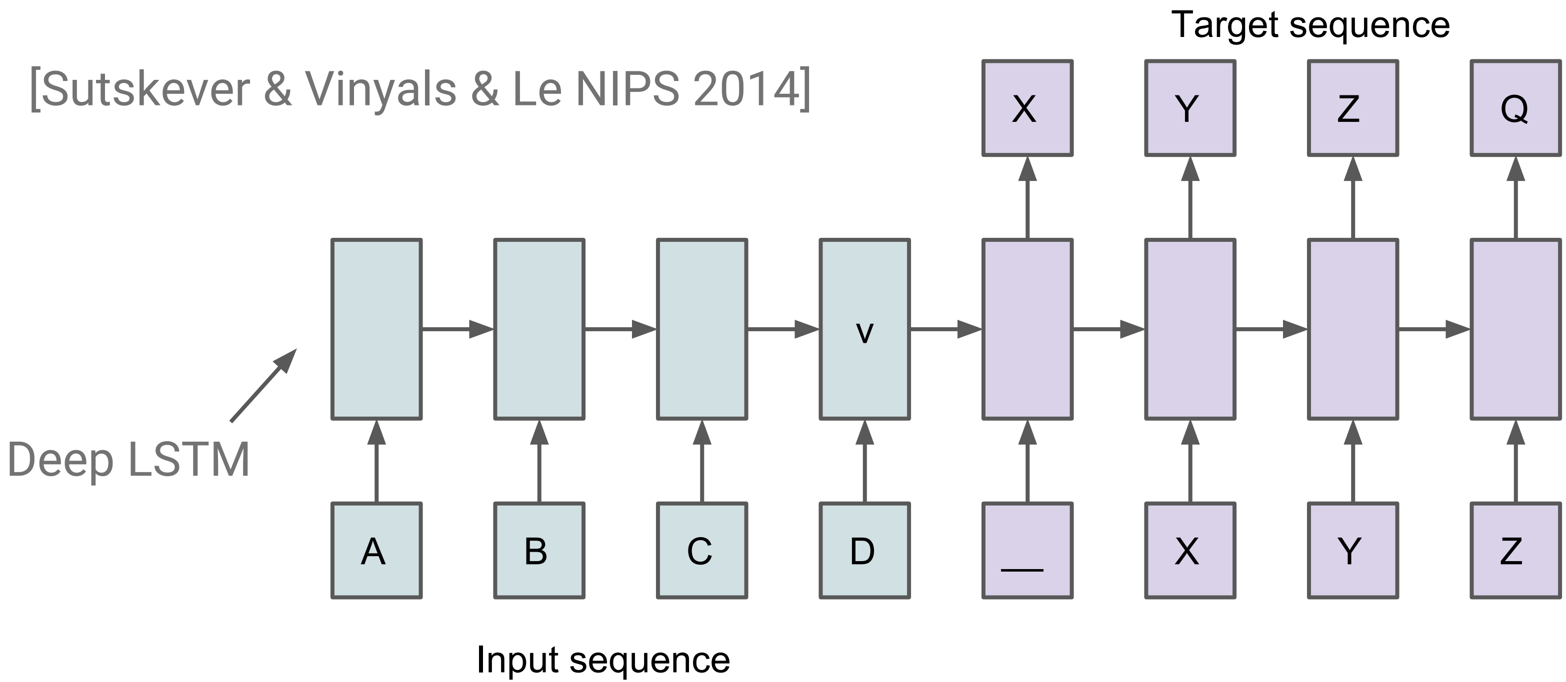
* pathologist given infinite time per image (reaching 0 FPs)

Evaluated using Camelyon16 dataset (**just 270 training examples!**)

Scaling language understanding models

Sequence-to-Sequence Model

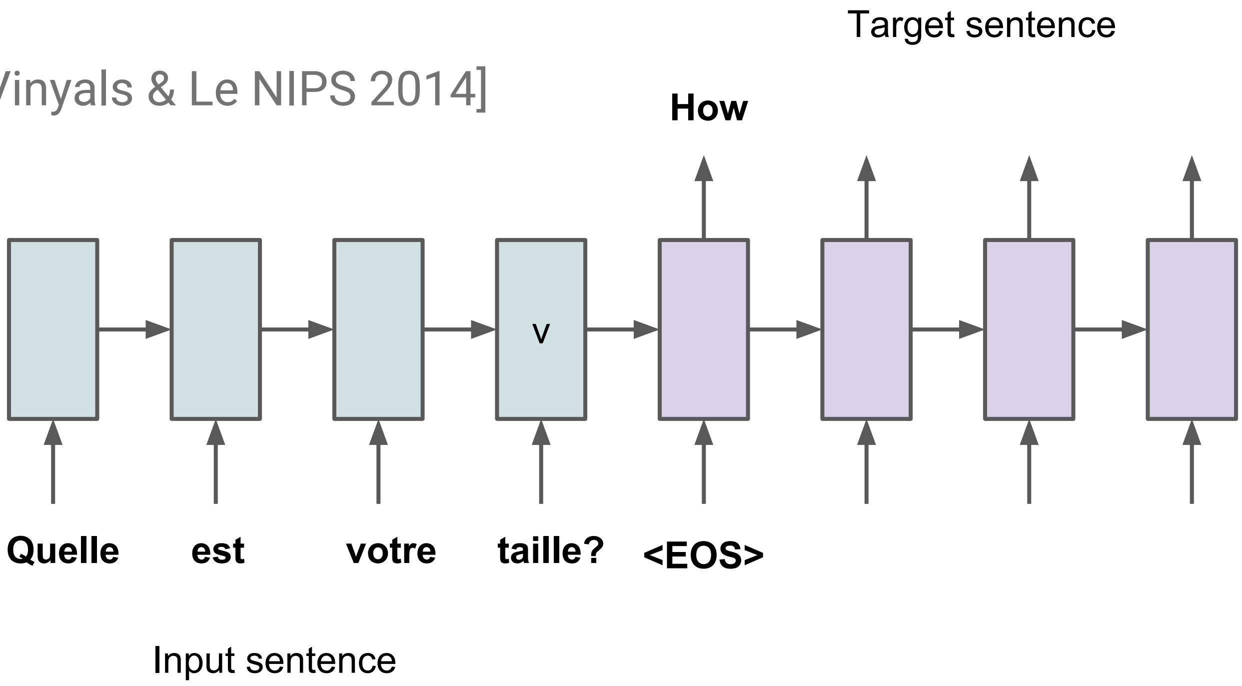
[Sutskever & Vinyals & Le NIPS 2014]



$$P(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1})$$

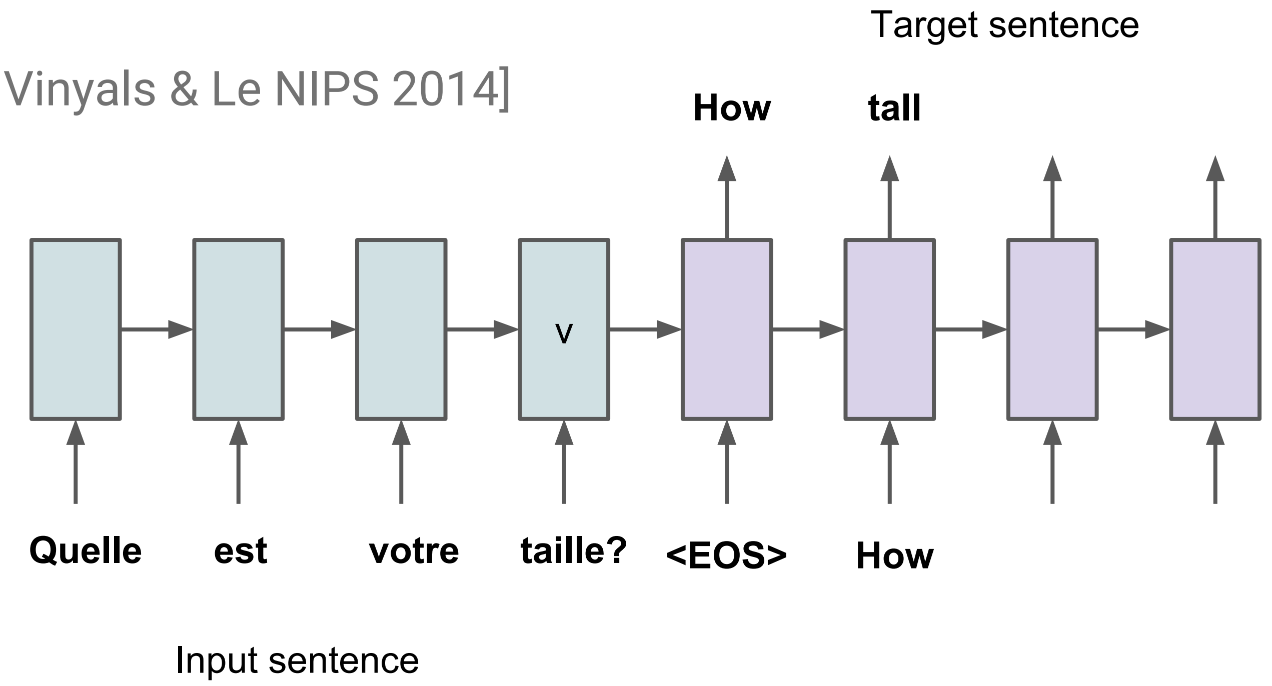
Sequence-to-Sequence Model: Machine Translation

[Sutskever & Vinyals & Le NIPS 2014]



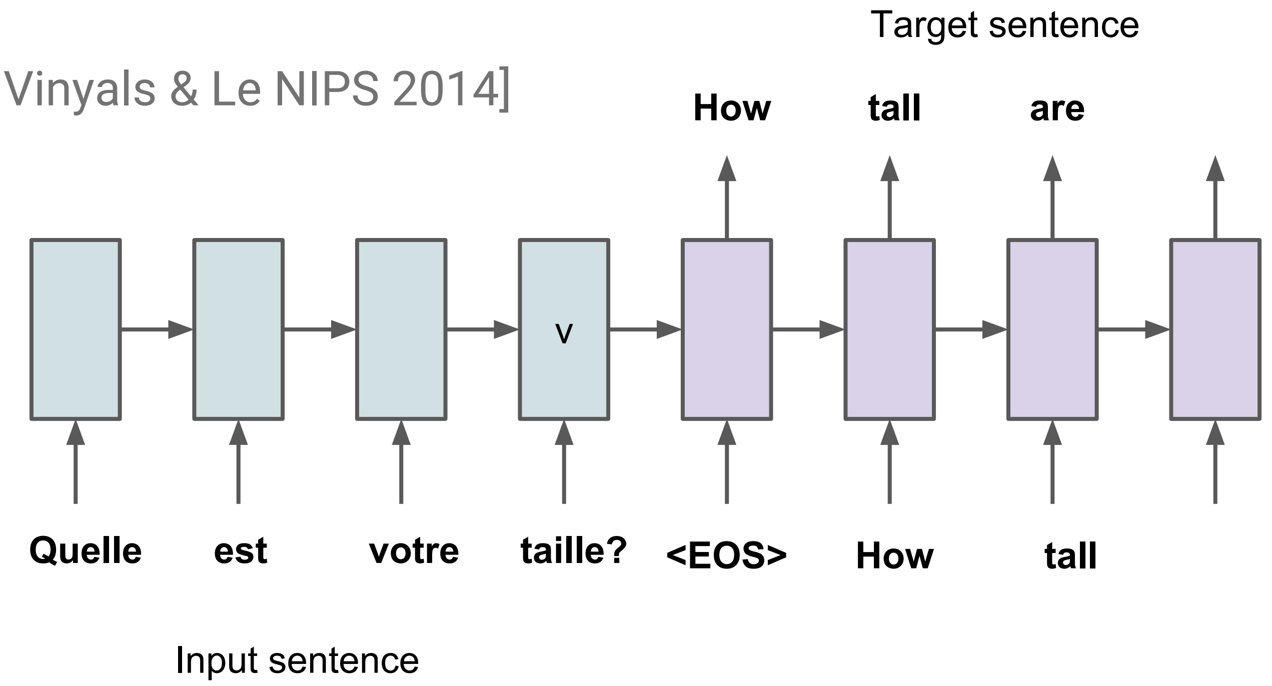
Sequence-to-Sequence Model: Machine Translation

[Sutskever & Vinyals & Le NIPS 2014]



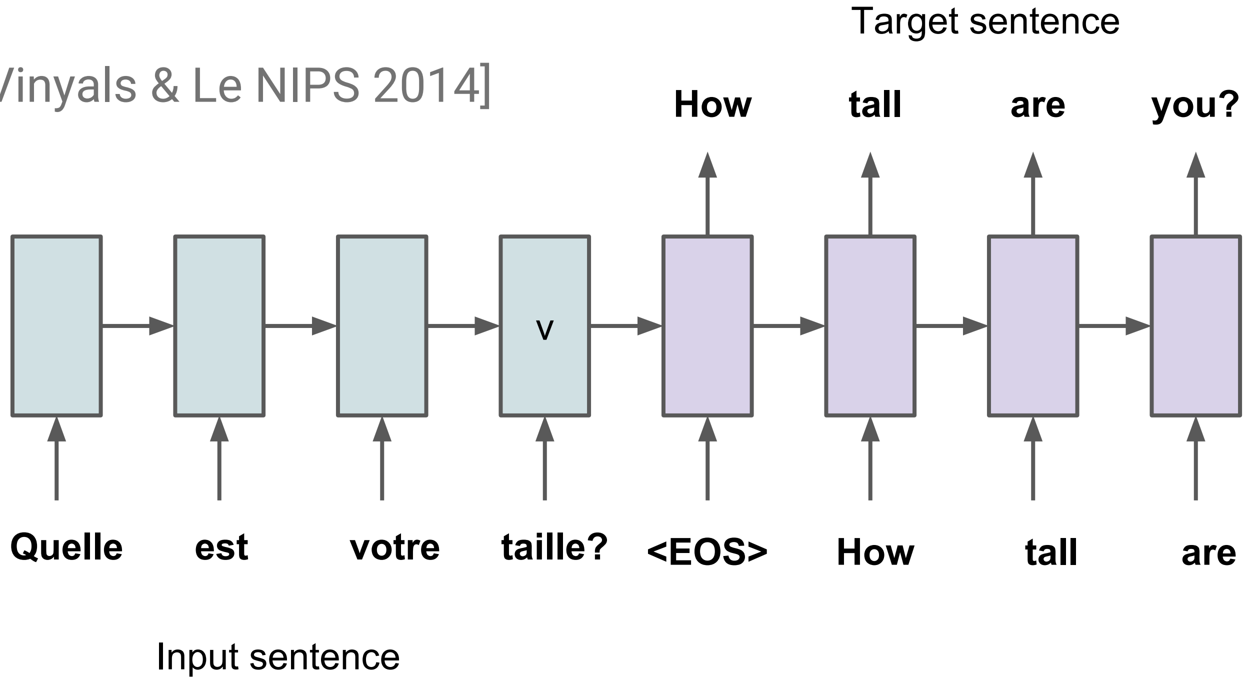
Sequence-to-Sequence Model: Machine Translation

[Sutskever & Vinyals & Le NIPS 2014]



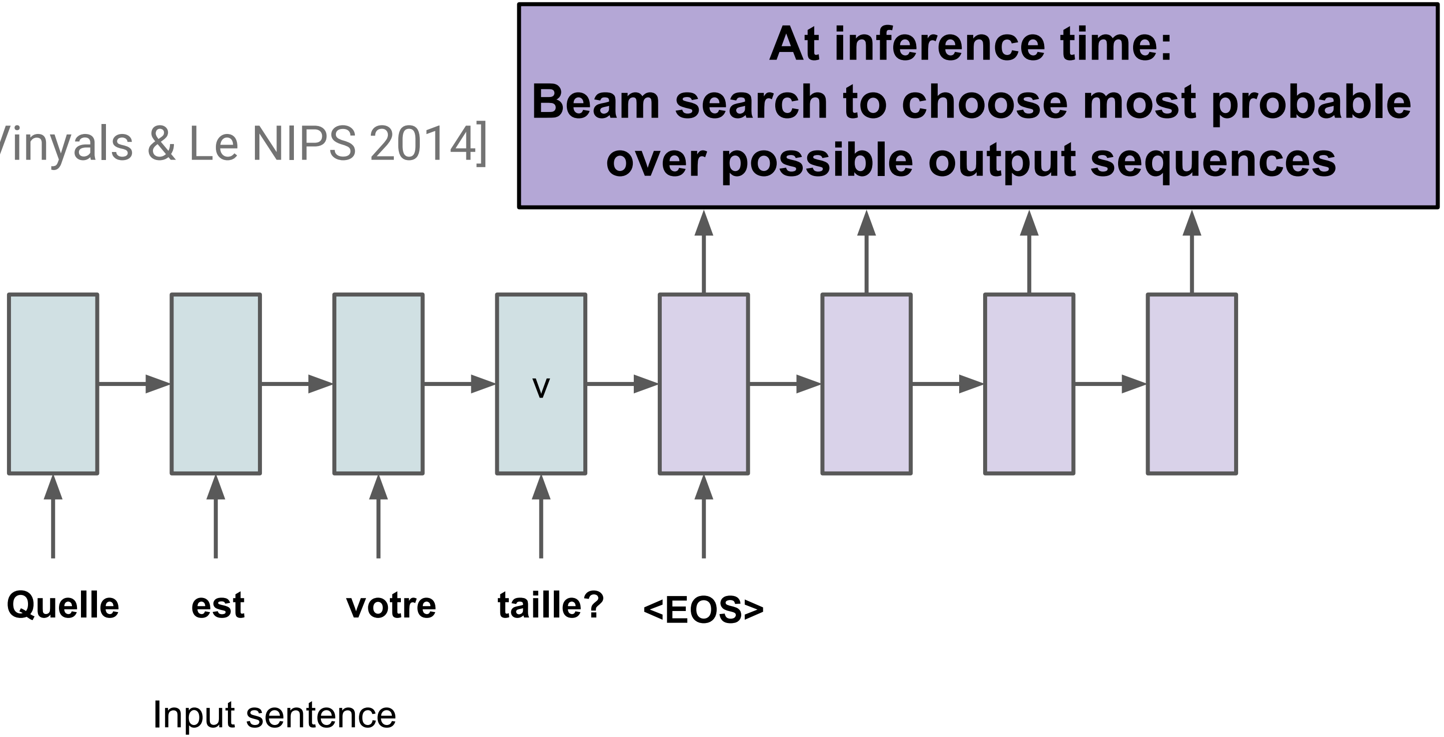
Sequence-to-Sequence Model: Machine Translation

[Sutskever & Vinyals & Le NIPS 2014]



Sequence-to-Sequence Model: Machine Translation

[Sutskever & Vinyals & Le NIPS 2014]



Sequence to Sequence model applied to Google Translate

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

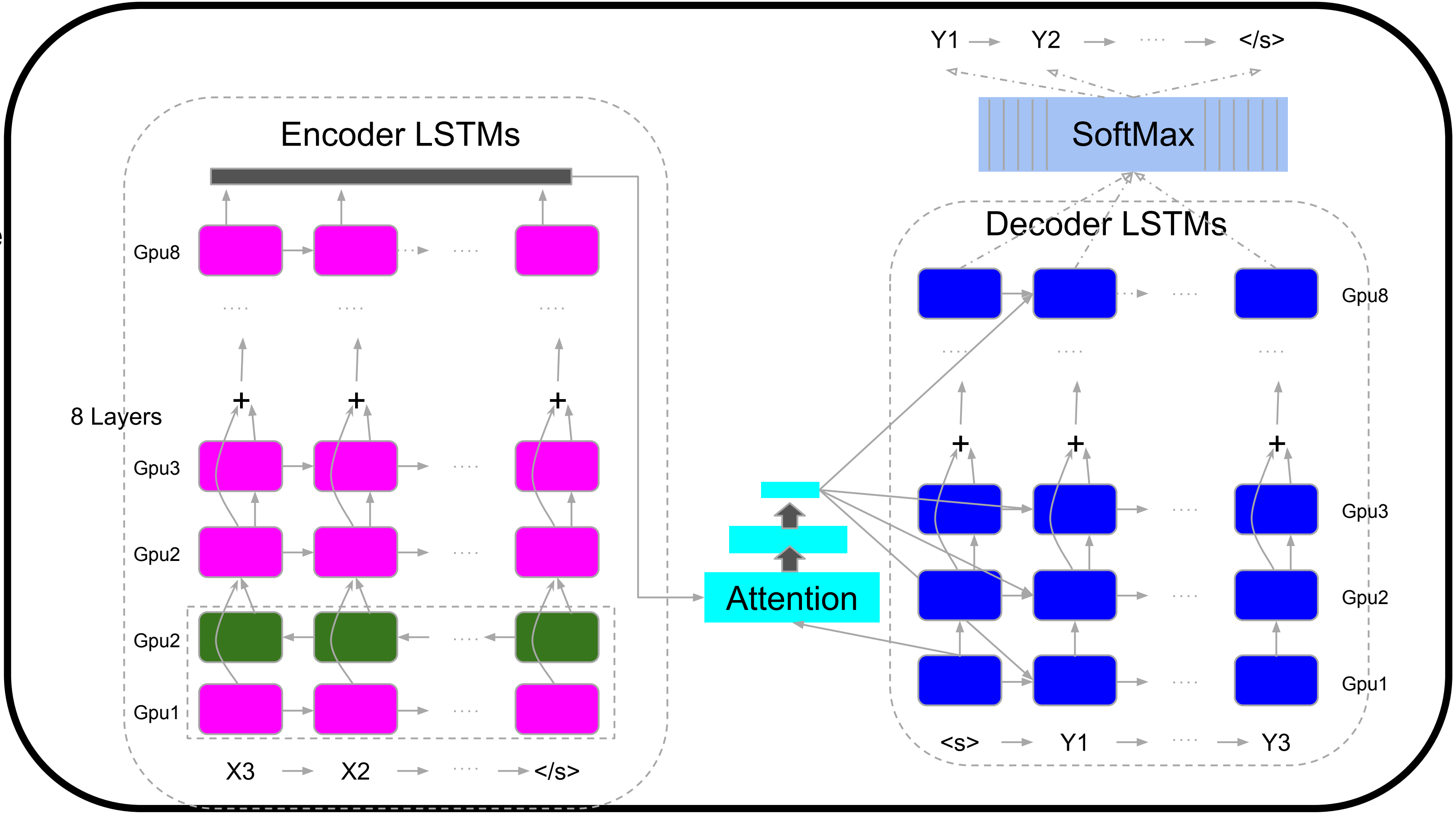
Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

https://arxiv.org/abs/1609.08144

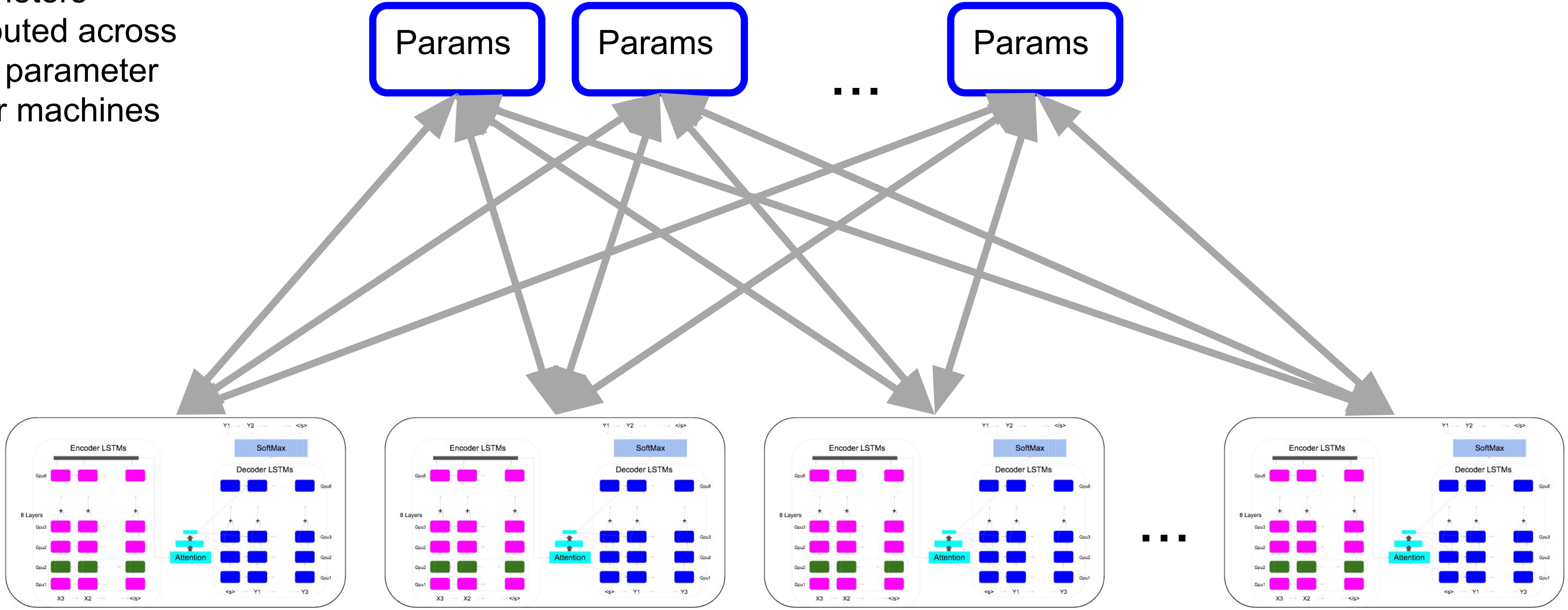
Google Neural Machine Translation Model

One model replica: one machine w/ 8 GPUs



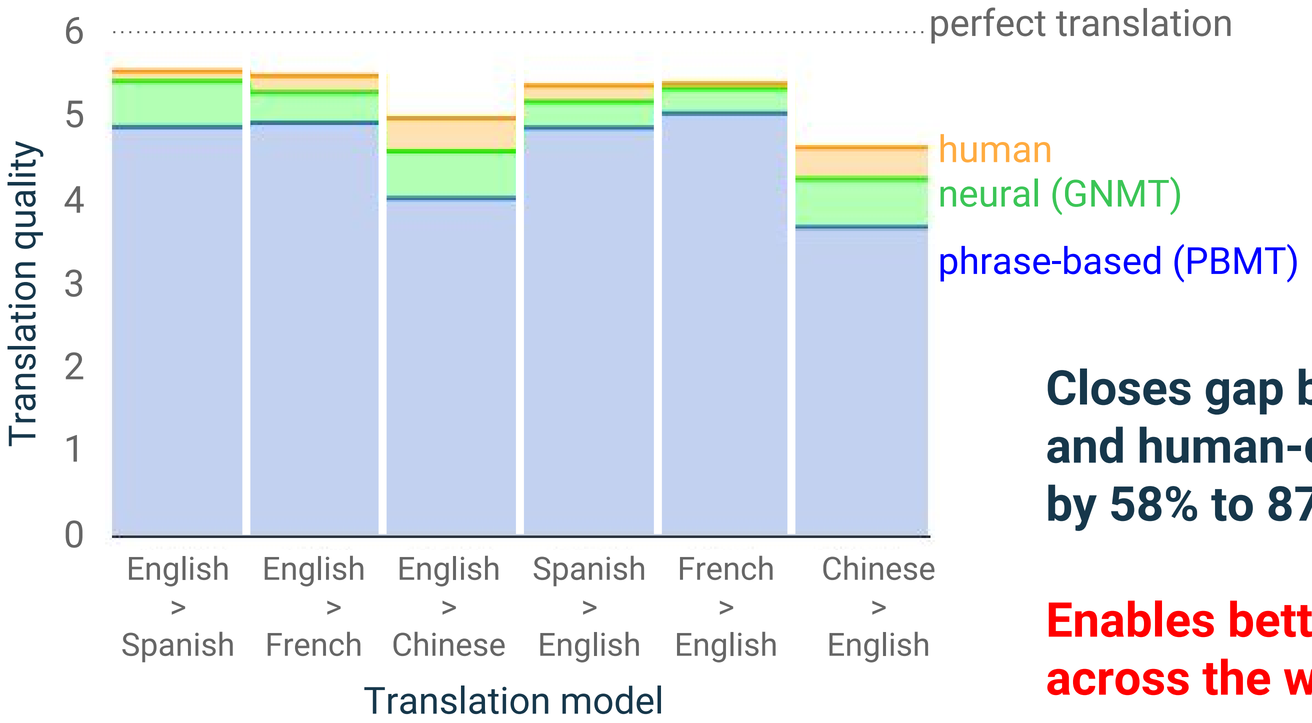
Model + Data Parallelism

Parameters distributed across many parameter server machines



Many replicas

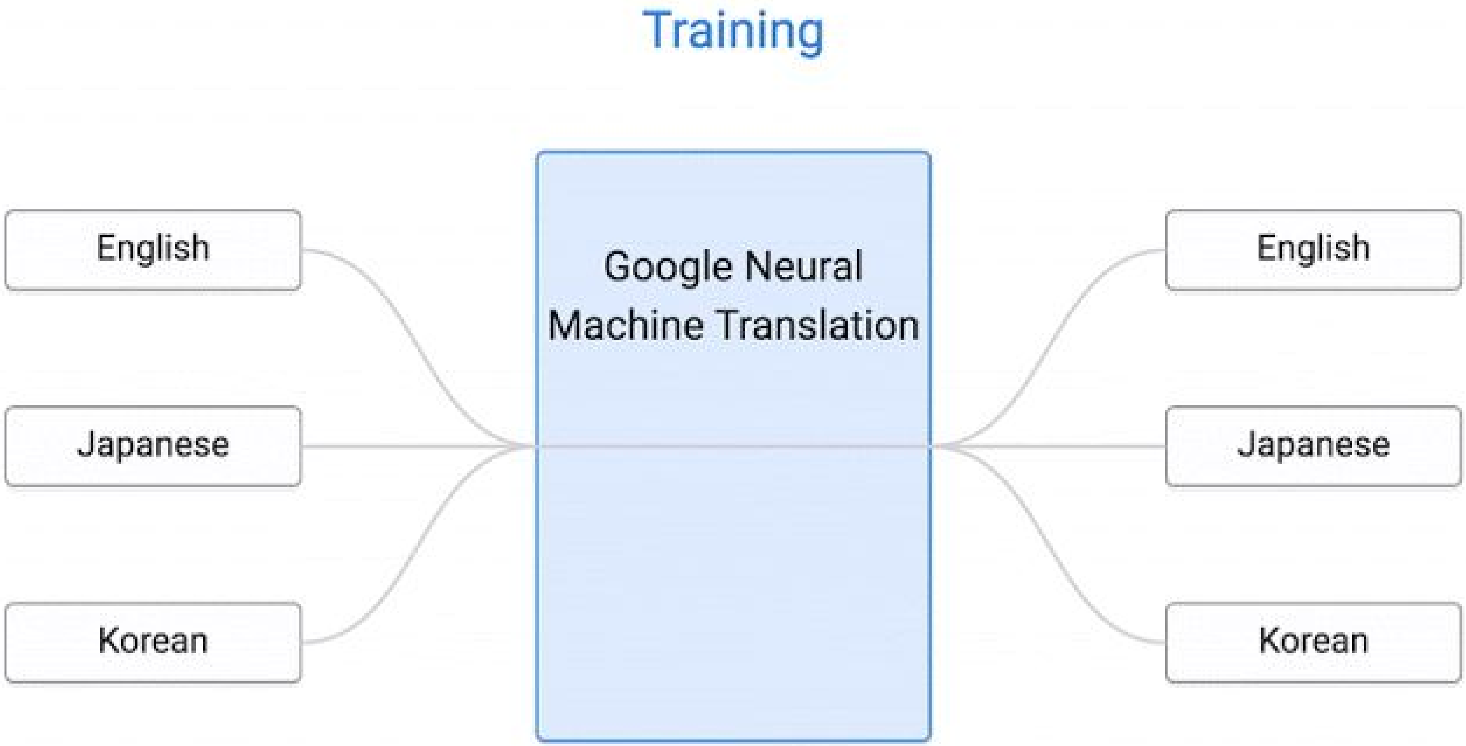
Neural Machine Translation



Closes gap between old system and human-quality translation by 58% to 87%

Enables better communication across the world

research.googleblog.com/2016/09/a-neural-network-for-machine.html



Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation,
Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean
https://arxiv.org/abs/1611.04558

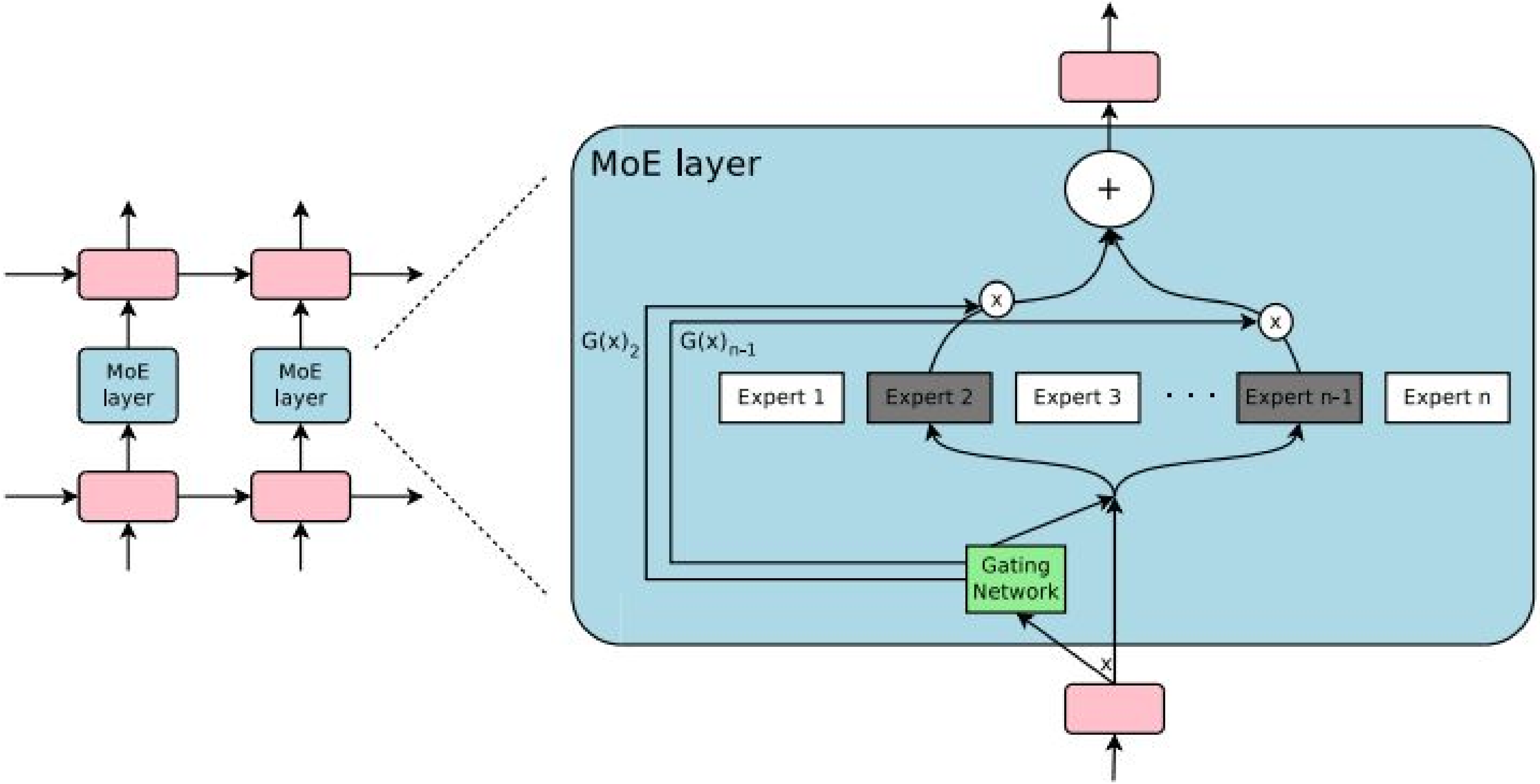
https://research.googleblog.com/2016/11/zero-shot-translation-with-googles.html

Bigger models, but sparsely activated

Bigger models, but sparsely activated

Motivation:
Want huge model capacity for large datasets, but want individual example to only activate tiny fraction of large model

Per-Example Routing



Per-Example Routing

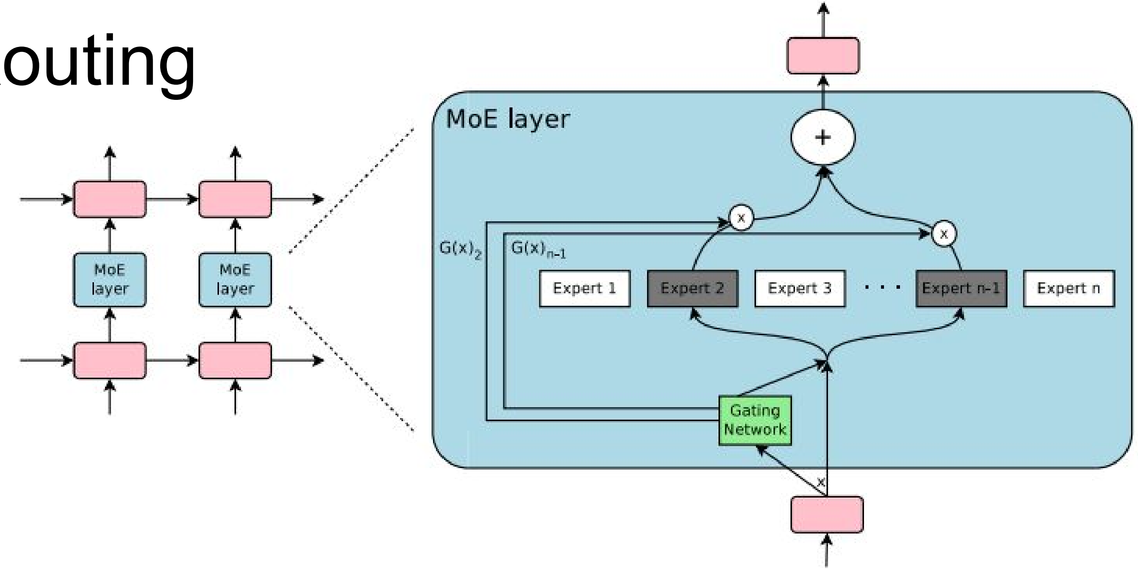


Table 7: Perplexity and BLEU comparison of our method against previous state-of-art methods on the Google Production En→ Fr dataset.

Model	Eval Perplexity	Eval BLEU	Test Perplexity	Test BLEU	Computation per Word	Total #Parameters	Training Time
MoE with 2048 Experts	2.60	37.27	2.69	36.57	100.8M	8.690B	1 day/64 k40s
GNMT (Wu et al., 2016)	2.78	35.80	2.87	35.56	214.2M	246.9M	6 days/96 k80s

Outrageously Large Neural Networks: The Sparsely-gated Mixture-of-Experts Layer, Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le & Jeff Dean
To appear in ICLR 2017, https://openreview.net/pdf?id=B1ckMDqlg

Automated machine learning (“learning to learn”)

Current:

Solution = ML expertise + data + computation

Current:

Solution = ML expertise + data + computation

Can we turn this into:

Solution = data + 100X computation

???

Early encouraging signs

Trying multiple different approaches:

(1) RL-based architecture search
(2) Model architecture evolution

NEURAL ARCHITECTURE SEARCH WITH REINFORCEMENT LEARNING

Barret Zoph*, Quoc V. Le
Google Brain
{barretzoph,qvl}@google.com

To appear in ICLR 2017

Idea: model-generating model trained via RL

(1) Generate ten models
(2) Train them for a few hours
(3) Use loss of the generated models as reinforcement learning signal

arxiv.org/abs/1611.01578

CIFAR-10 Image Recognition Task

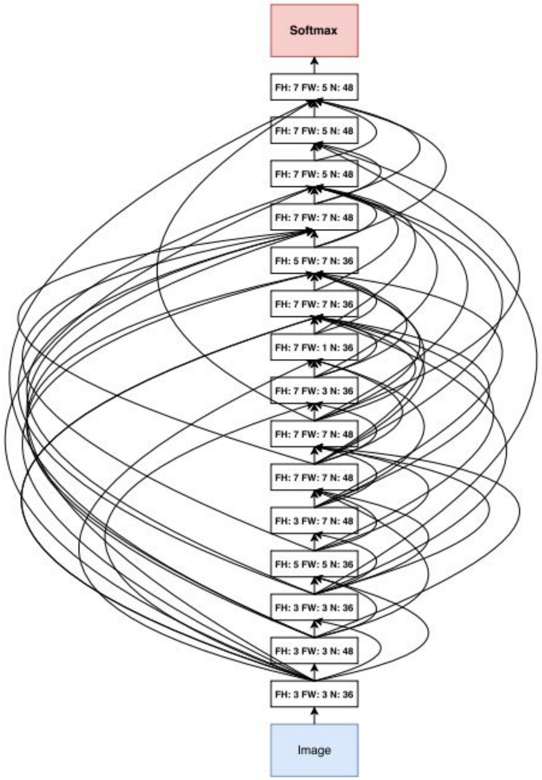


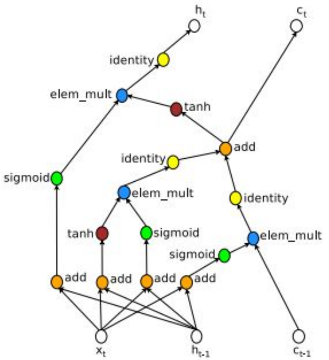
Figure 7: Convolutional architecture discovered by our method, when the search space does not have strides or pooling layers. FH is filter height, FW is filter width and N is number of filters.

Model	Depth	Parameters	Error rate (%)
Network in Network (Lin et al., 2013)	-	-	8.81
All-CNN (Springenberg et al., 2014)	-	-	7.25
Deeply Supervised Net (Lee et al., 2015)	-	-	7.97
Highway Network (Srivastava et al., 2015)	-	-	7.72
Scalable Bayesian Optimization (Snoek et al., 2015)	-	-	6.37
FractalNet (Larsson et al., 2016)	21	38.6M	5.22
with Dropout/Drop-path	21	38.6M	4.60
ResNet (He et al., 2016a)	110	1.7M	6.61
ResNet (reported by Huang et al. (2016b))	110	1.7M	6.41
ResNet with Stochastic Depth (Huang et al., 2016b)	110	1.7M	5.23
	1202	10.2M	4.91
Wide ResNet (Zagoruyko & Komodakis, 2016)	16	11.0M	4.81
	28	36.5M	4.17
ResNet (pre-activation) (He et al., 2016b)	164	1.7M	5.46
	1001	10.2M	4.62
DenseNet ($L = 40, k = 12$) Huang et al. (2016a)	40	1.0M	5.24
DenseNet($L = 100, k = 12$) Huang et al. (2016a)	100	7.0M	4.10
DenseNet ($L = 100, k = 24$) Huang et al. (2016a)	100	27.2M	3.74
Neural Architecture Search v1 no stride or pooling	15	4.2M	5.50
Neural Architecture Search v2 predicting strides	20	2.5M	6.01
Neural Architecture Search v3 max pooling	39	7.1M	4.47
Neural Architecture Search v3 max pooling + more filters	39	32.0M	3.84

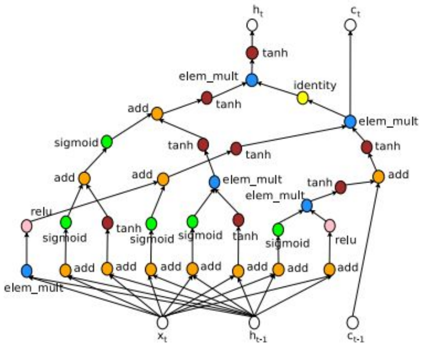
Table 1: Performance of Neural Architecture Search and other state-of-the-art models on CIFAR-10.

Penn Tree Bank Language Modeling Task

“Normal” LSTM cell



Cell discovered by architecture search



Model	Parameters	Test Perplexity
Mikolov & Zweig (2012) - KN-5	2M‡	141.2
Mikolov & Zweig (2012) - KN5 + cache	2M‡	125.7
Mikolov & Zweig (2012) - RNN	6M‡	124.7
Mikolov & Zweig (2012) - RNN-LDA	7M‡	113.7
Mikolov & Zweig (2012) - RNN-LDA + KN-5 + cache	9M‡	92.0
Pascanu et al. (2013) - Deep RNN	6M	107.5
Cheng et al. (2014) - Sum-Prod Net	5M‡	100.0
Zaremba et al. (2014) - LSTM (medium)	20M	82.7
Zaremba et al. (2014) - LSTM (large)	66M	78.4
Gal (2015) - Variational LSTM (medium, untied)	20M	79.7
Gal (2015) - Variational LSTM (medium, untied, MC)	20M	78.6
Gal (2015) - Variational LSTM (large, untied)	66M	75.2
Gal (2015) - Variational LSTM (large, untied, MC)	66M	73.4
Kim et al. (2015) - CharCNN	19M	78.9
Press & Wolf (2016) - Variational LSTM, shared embeddings	24M	73.2
Merity et al. (2016) - Zoneout + Variational LSTM (medium)	20M	80.6
Merity et al. (2016) - Pointer Sentinel-LSTM (medium)	21M	70.9
Zilly et al. (2016) - Variational RHN, shared embeddings	24M	$\boxed{66.0}$
Neural Architecture Search with base 8	32M	67.9
Neural Architecture Search with base 8 and shared embeddings	25M	64.0
Neural Architecture Search with base 8 and shared embeddings	54M	$\boxed{62.4}$

Table 2: Single model perplexity on the test set of the Penn Treebank language modeling task. Parameter numbers with ‡ are estimates with reference to Merity et al. (2016).

Large-Scale Evolution of Image Classifiers

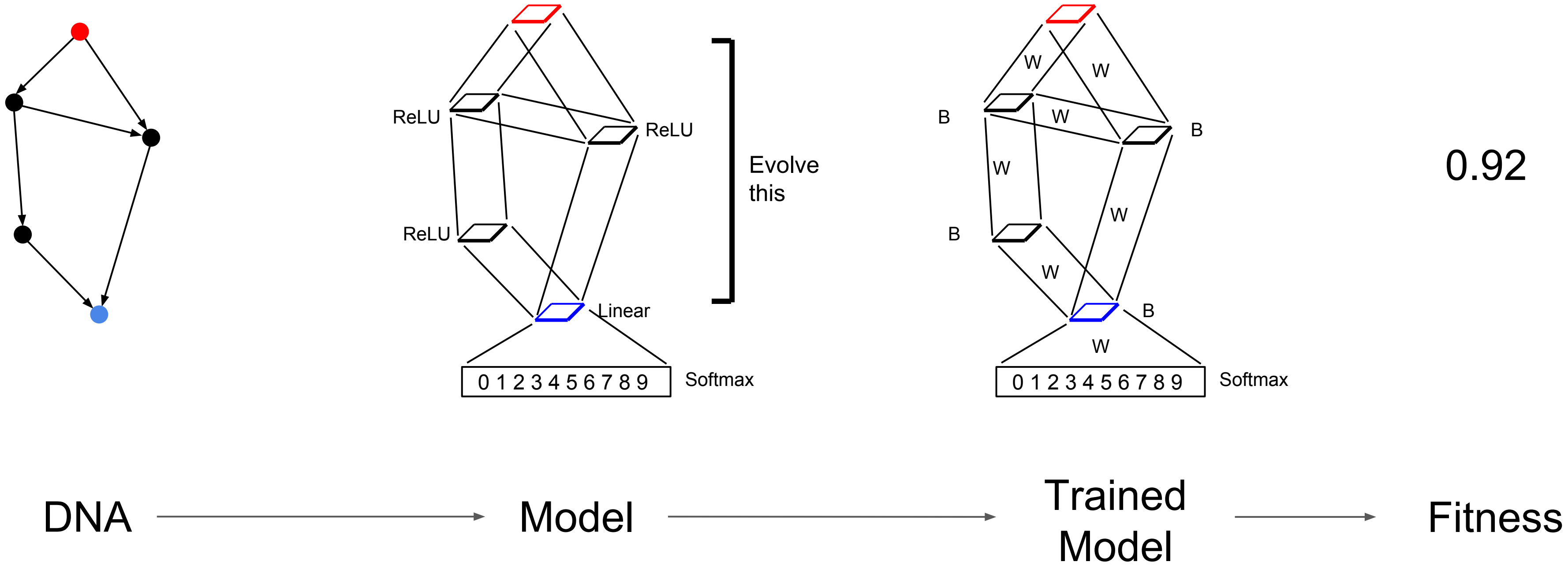
Esteban Real [1] Sherry Moore [1] Andrew Selle [1] Saurabh Saxena [1]
Yutaka Leon Suematsu [2] Quoc Le [1] Alex Kurakin [1]

Idea: evolve models via evolutionary algorithm

https://arxiv.org/abs/1703.01041

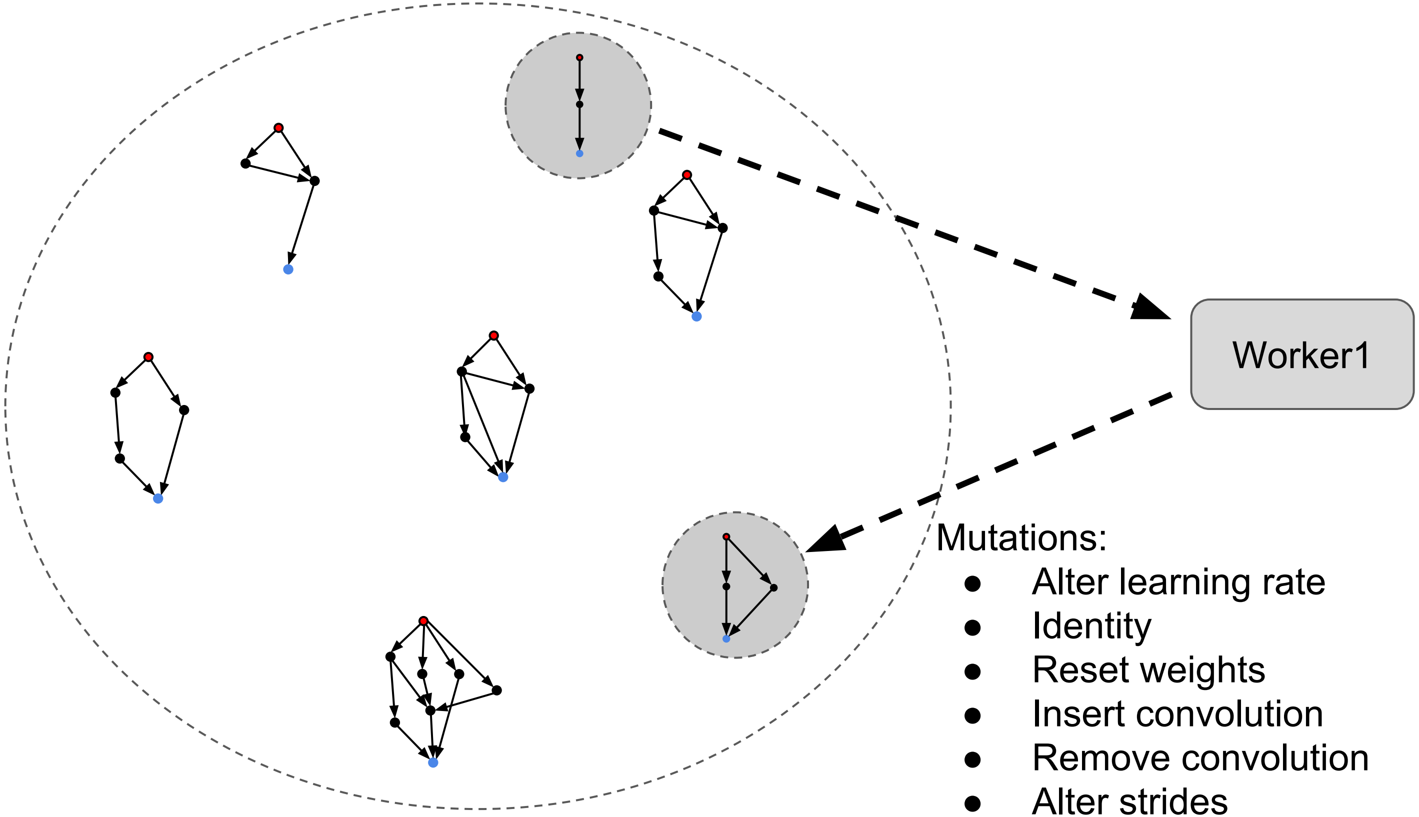
Large-Scale Evolution of Image Classifiers

Esteban Real [1] Sherry Moore [1] Andrew Selle [1] Saurabh Saxena [1]
Yutaka Leon Suematsu [2] Quoc Le [1] Alex Kurakin [1]



https://arxiv.org/abs/1703.01041

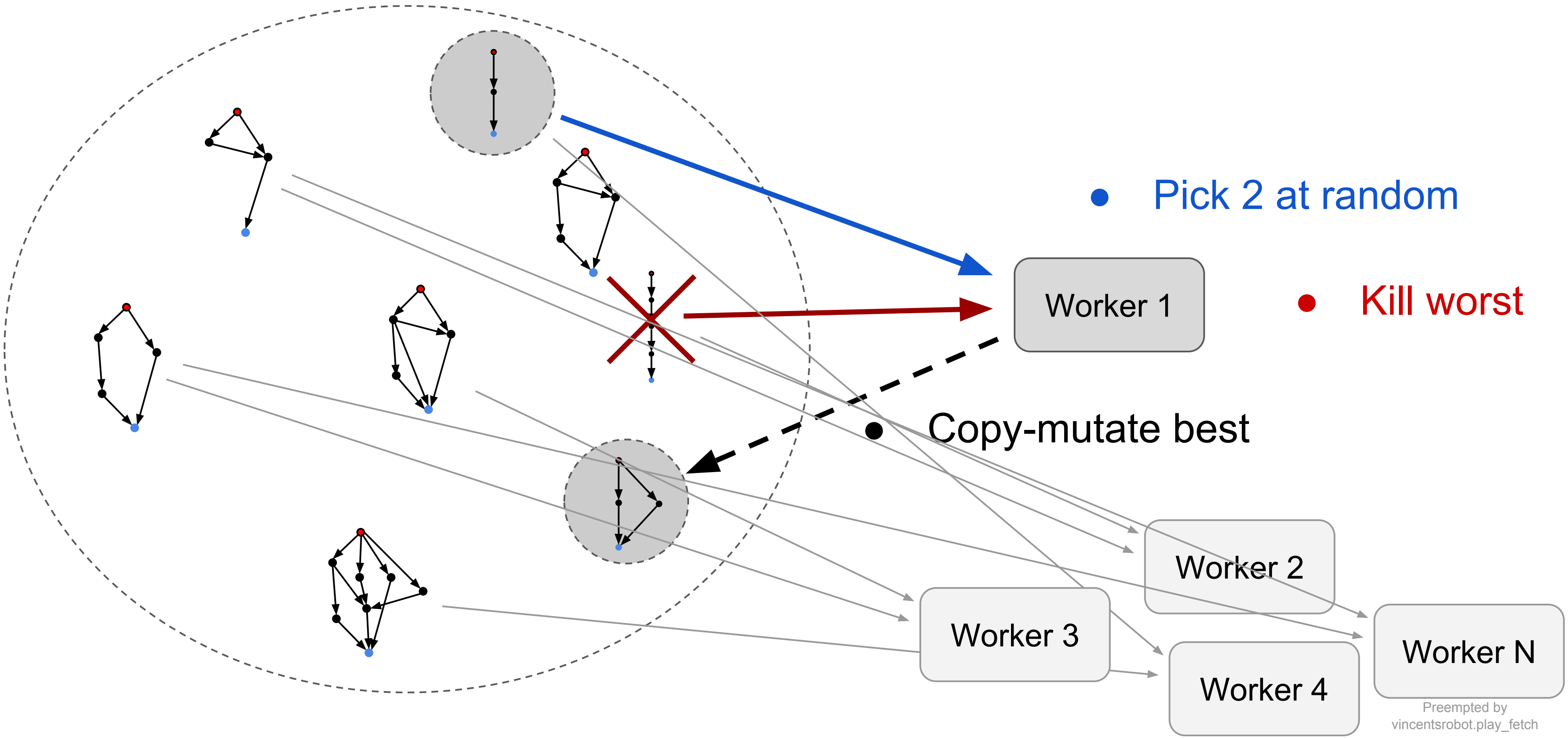
Evolutionary Step



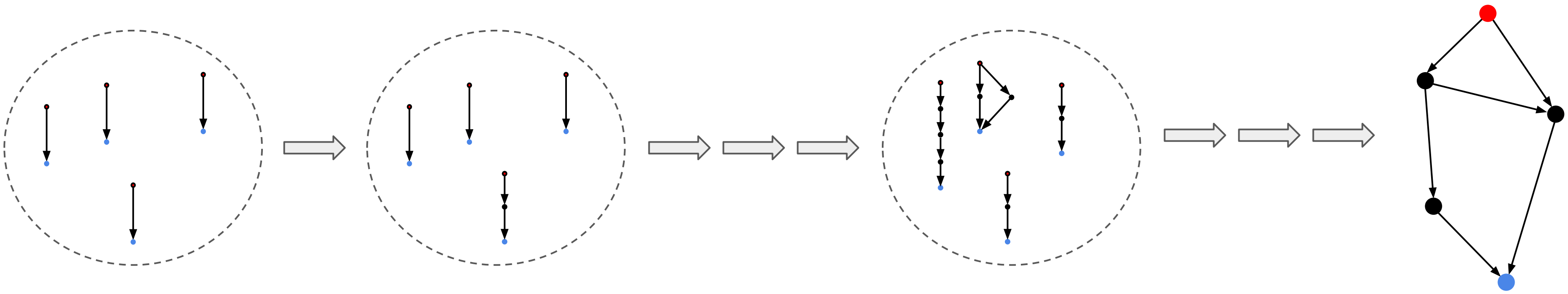
Mutations:

- Alter learning rate
- Identity
- Reset weights
- Insert convolution
- Remove convolution
- Alter strides
- Alter # of channels
- Alter horiz. filter size
- Alter vert. filters size
- Insert nonlinearity
- Remove nonlinearity
- Add-skip
- Remove skip

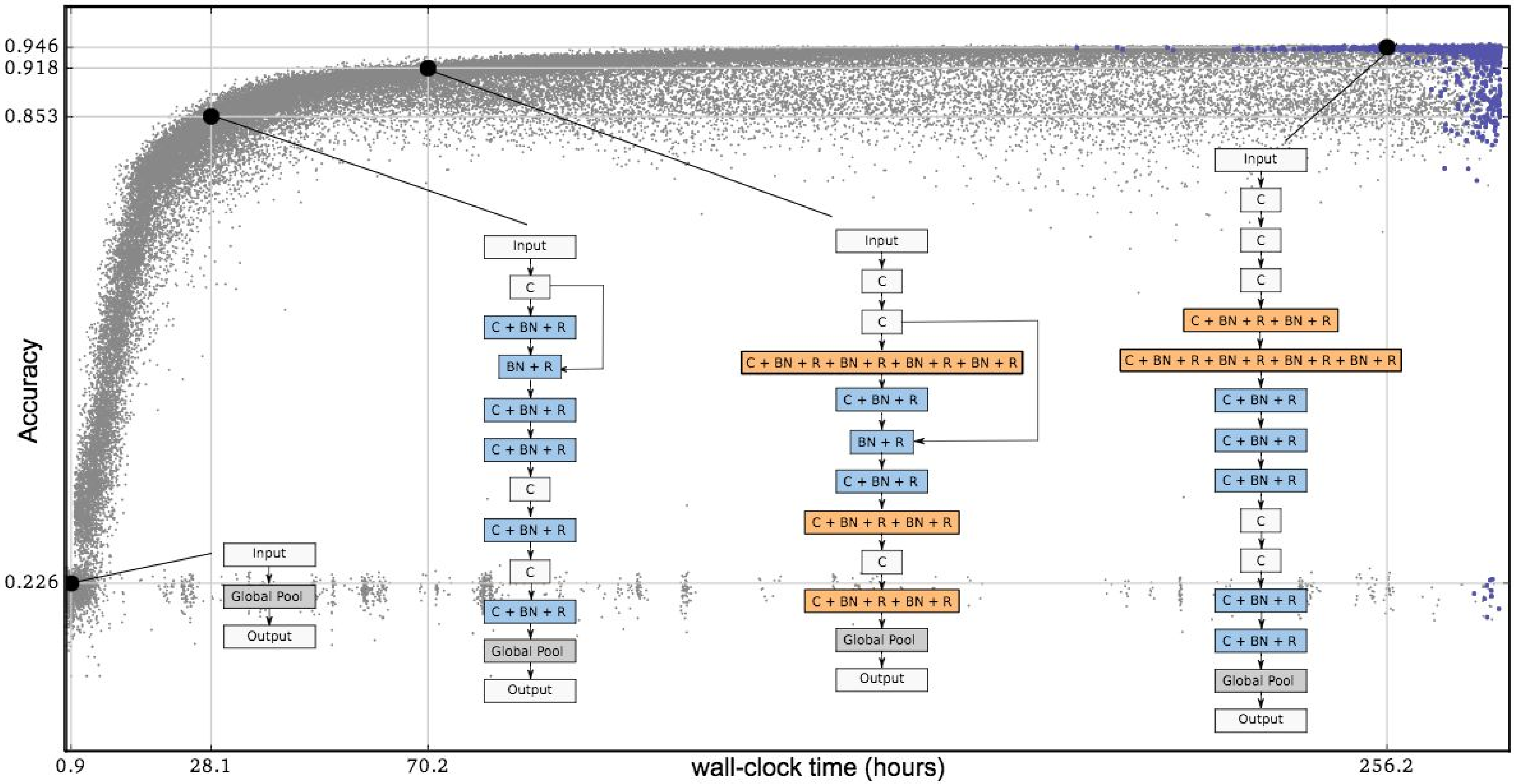
Evolutionary Step



Evolve From Scratch



- Initialize with linear models
- Repeat evolutionary step



0.946
0.918
0.853
0.226
Accuracy
0.9
28.1
70.2
256.2
wall-clock time (hours)
Input
Global Pool
Output
Input
C
C + BN + R
BN + R
C + BN + R
C + BN + R
C
C + BN + R
C
C + BN + R
Global Pool
Output
Input
C
C
C + BN + R + BN + R + BN + R + BN + R
C + BN + R
BN + R
C + BN + R
C + BN + R + BN + R
C
C + BN + R + BN + R
Global Pool
Output
Input
C
C
C
C + BN + R + BN + R
C + BN + R + BN + R + BN + R + BN + R
C + BN + R
C + BN + R
C + BN + R
C
C
C + BN + R
C + BN + R
Global Pool
Output

Study	Params.	C10+	C100+	Within?
Maxout (Goodfellow et al., 2013)	–	90.7%	61.4%	No
Network in Network (Lin et al., 2013)	–	91.2%	–	No
All-CNN (Springenberg et al., 2014)	1.3 M	92.8%	66.3%	Yes
Deeply Supervised (Lee et al., 2015)	–	92.0%	65.4%	No
Highway (Srivastava et al., 2015)	2.3 M	92.3%	67.6%	No
ResNet (He et al., 2016)	1.7 M	93.4%	72.8%†	Yes
Evolution (ours)	5.4 M 40.4 M	94.6%	 76.0%	N/A
Wide ResNet 28-10 (Zagoruyko & Komodakis, 2016)	36.5 M	96.0%	80.0%	Yes
Wide ResNet 40-10+d/o (Zagoruyko & Komodakis, 2016)	50.7 M	96.2%	81.7%	No
DenseNet (Huang et al., 2016a)	25.6 M	96.7%	82.8%	No

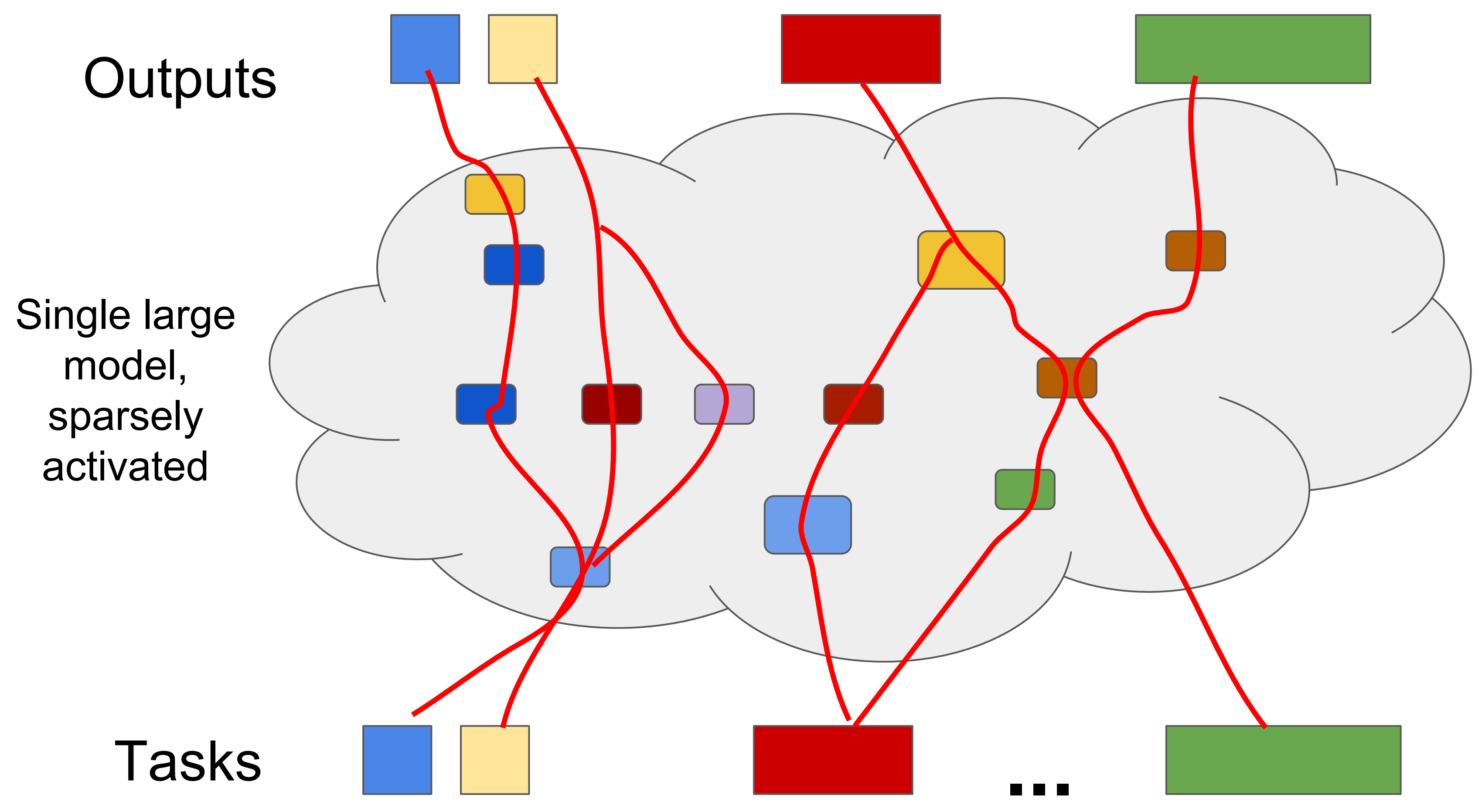
Study	Starting Point	Constraints	Post-Processing	Params.	C10+	C100+
Bayesian (Snoek et al., 2012)	3 layers	fixed architecture, no skips	none	–	90.5%	–
Q-Learning (Baker et al., 2016)	–	discrete params., max. num. layers, no skips	tune, retrain	11.2 M	93.1%	72.9%
RL (Zoph & Le, 2016)	20 layers, 50% skips	discrete params., exactly 20 layers	small grid search, retrain	2.5 M	94.0%	–
RL (Zoph & Le, 2016)	39 layers, 2 pool layers at 13 and 26, 50% skips	discrete params., exactly 39 layers, 2 pool layers at 13 and 26	add more filters, small grid search, retrain	32.0 M	96.2%	–
Evolution (ours)	linear model, zero convs.	power-of-2 strides	None	5.4 M 40.4 M	94.6%	 76.0%

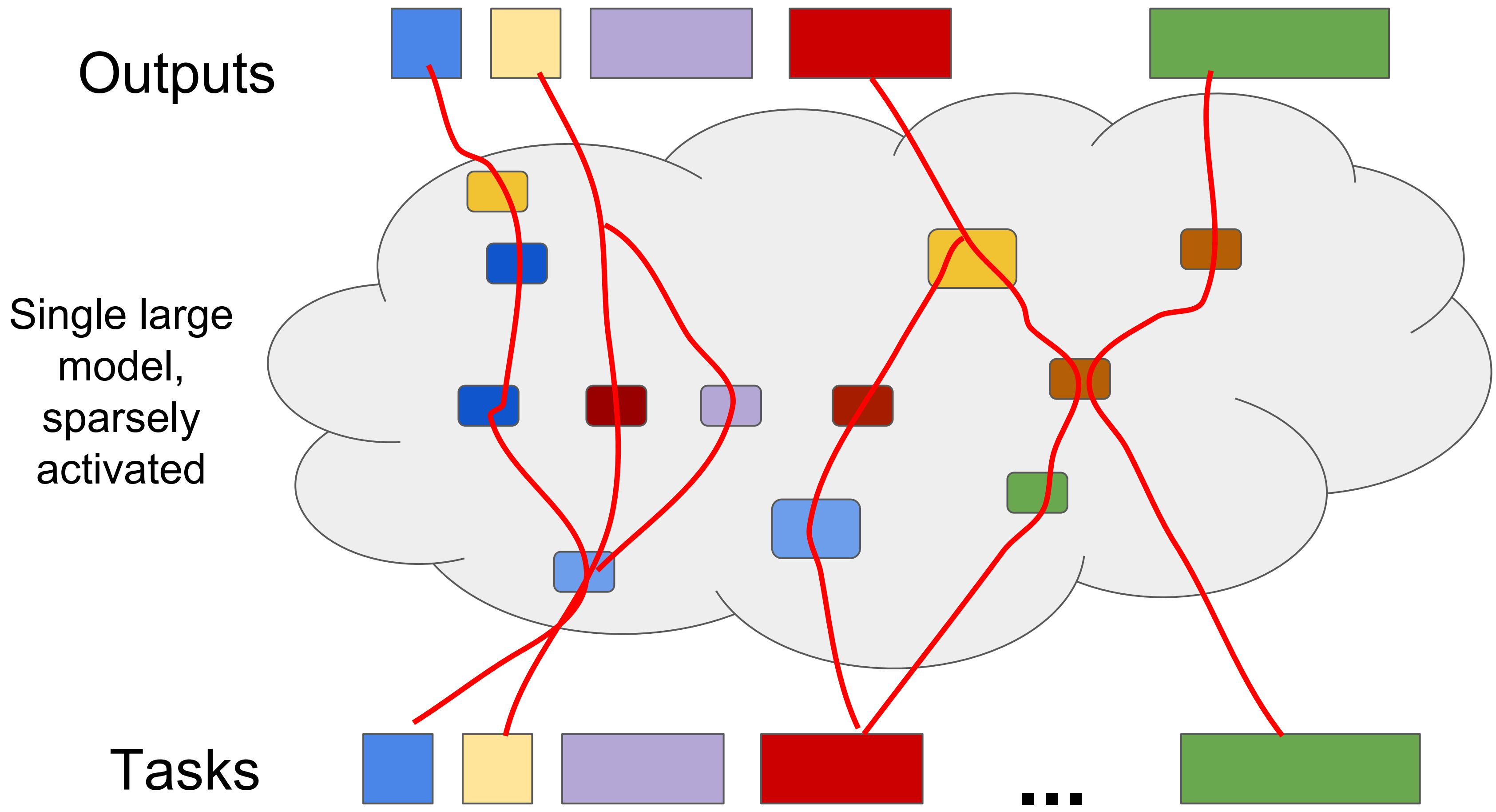
Where are we trying to go?

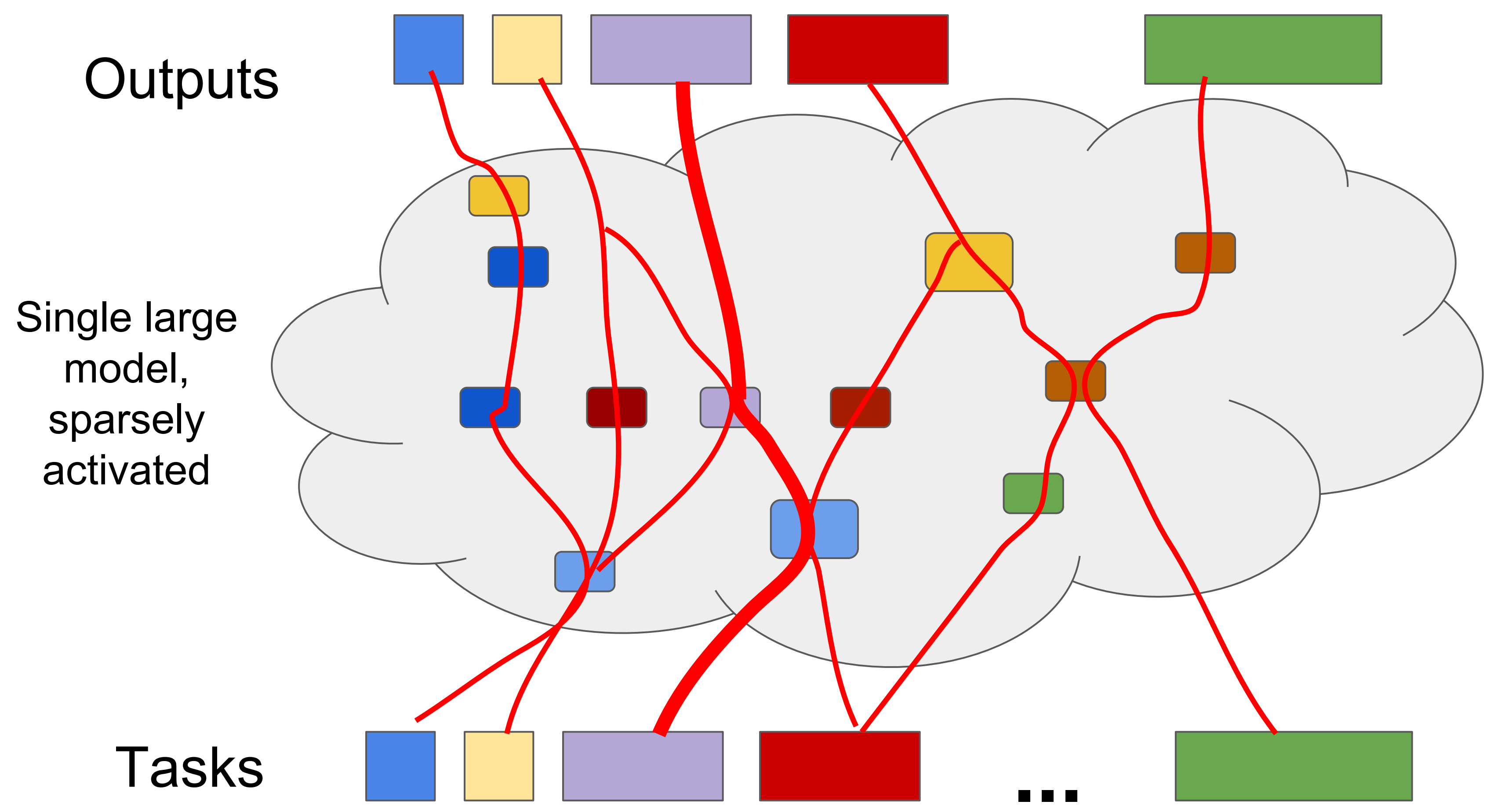
Where are we trying to go?

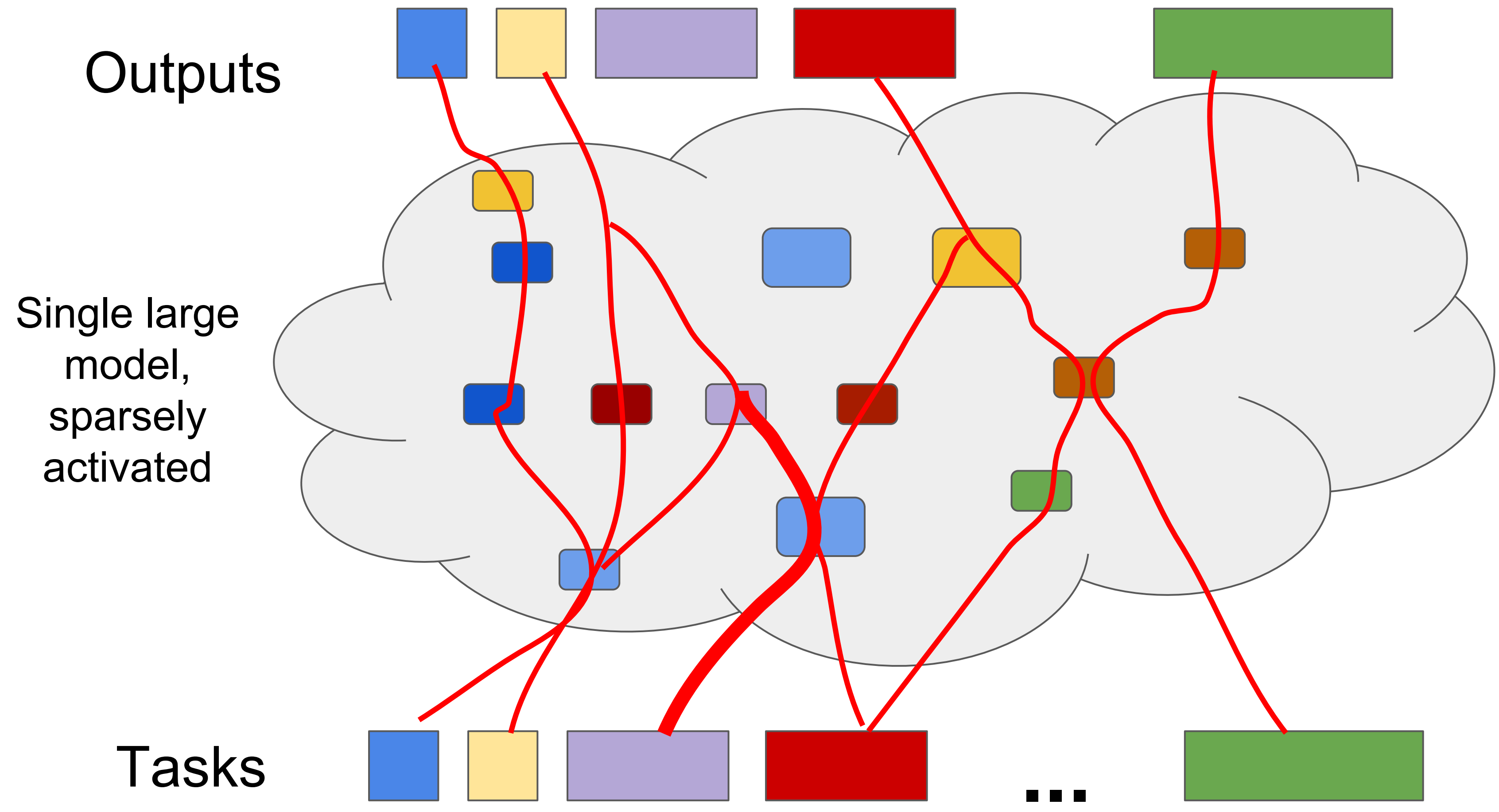
Combine Several of These Ideas:

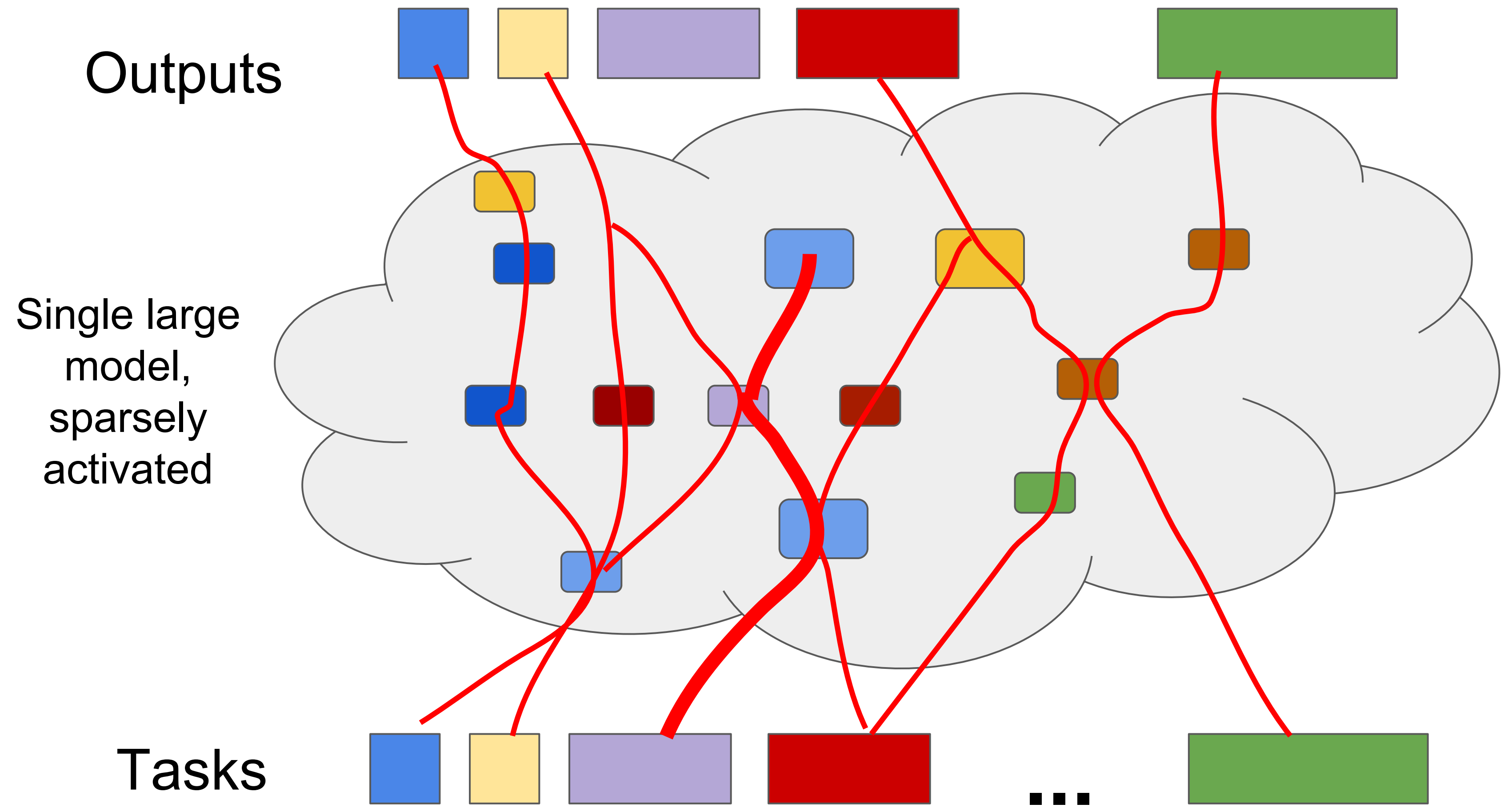
Large model, but **sparsely activated**
Single model to **solve many tasks** (100s to 1Ms)
Dynamically learn and grow pathways through large model

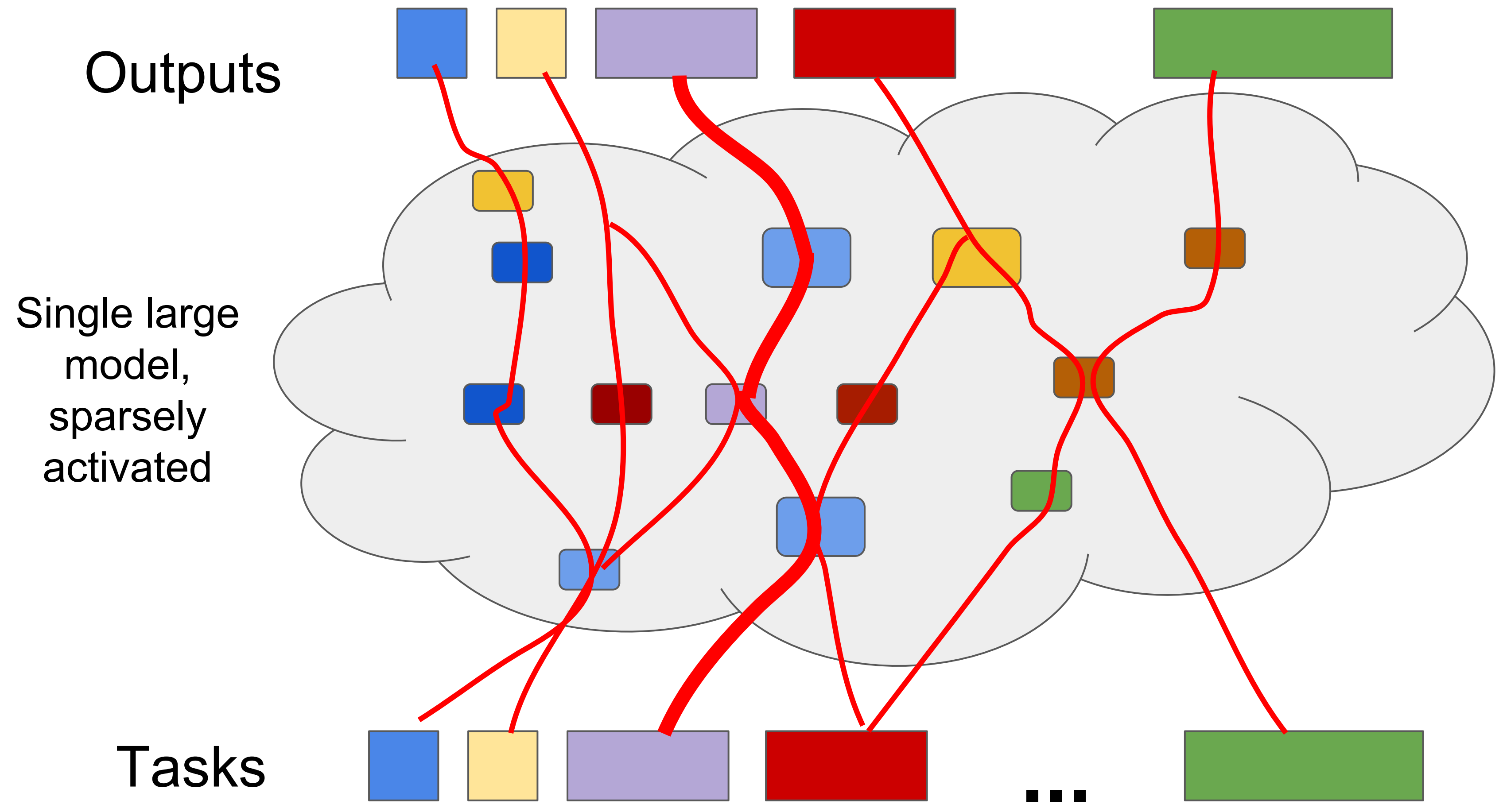

Outputs
Single large model, sparsely activated
Tasks
...


Outputs
Single large model, sparsely activated
Tasks
...


Outputs
Single large model, sparsely activated
Tasks
...


Outputs
Single large model, sparsely activated
Tasks
...


Outputs
Single large model, sparsely activated
Tasks
...



Outputs
Single large model, sparsely activated
Tasks
...

More computational power needed

Deep learning is transforming how we design computers

Special computation properties

reduced
precision
ok

about 1.2
× about 0.6
about 0.7

NOT

~~1.21042~~
~~× 0.61127~~
~~0.73989343~~

Special computation properties

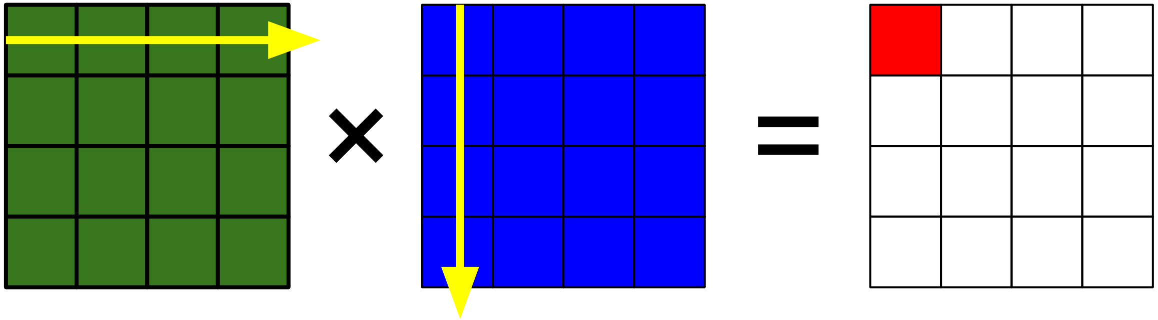
reduced precision ok

about 1.2
× about 0.6
about 0.7

NOT

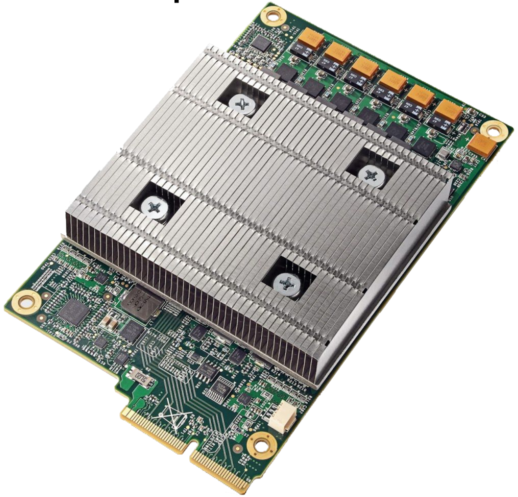
~~1.21042~~
~~× 0.61127~~
~~0.73989343~~

handful of specific operations



Tensor Processing Unit

Custom Google-designed chip for
neural net computations



In production use for >24 months: used on every search query, for neural machine translation, for AlphaGo match, …

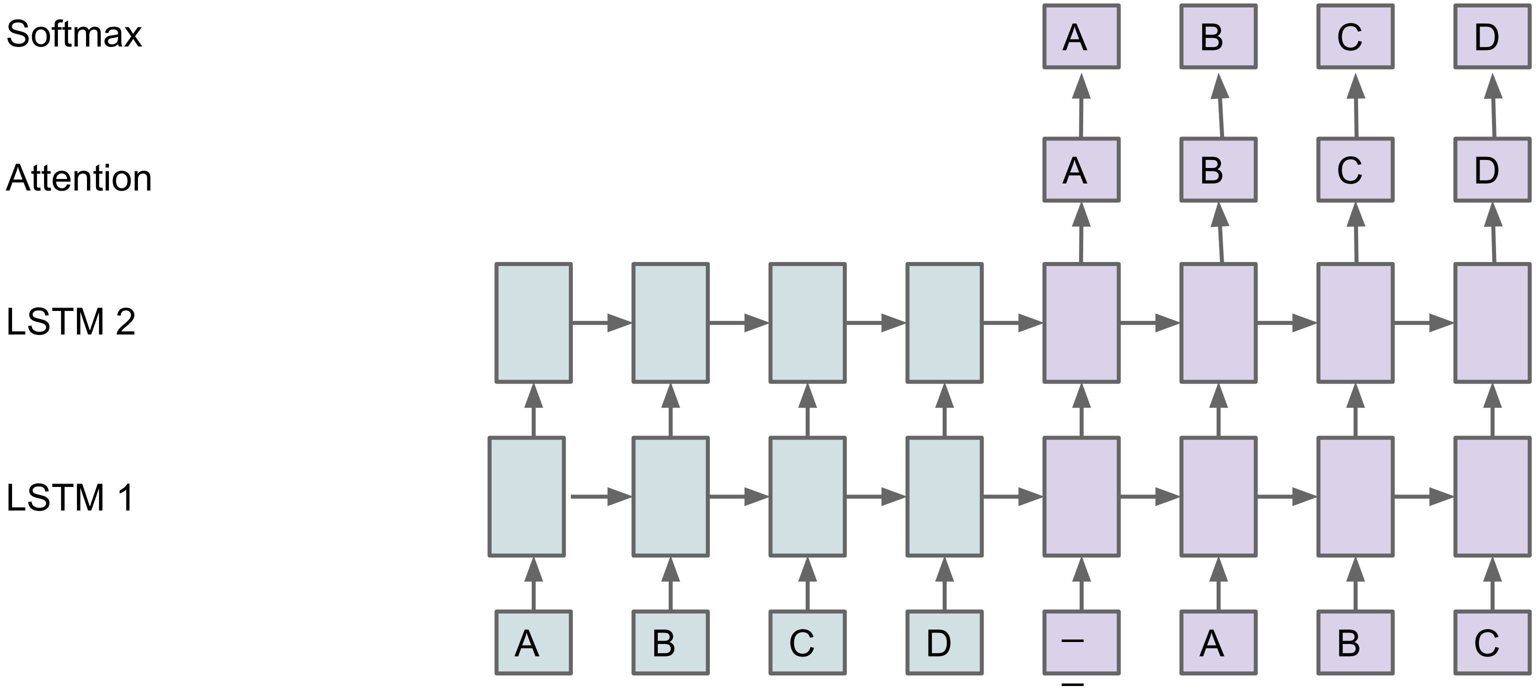
Talk at Computer History Museum on April 5th:
sites.google.com/view/naeregionalsymposium

Machine Learning for
Higher Performance Machine Learning Models

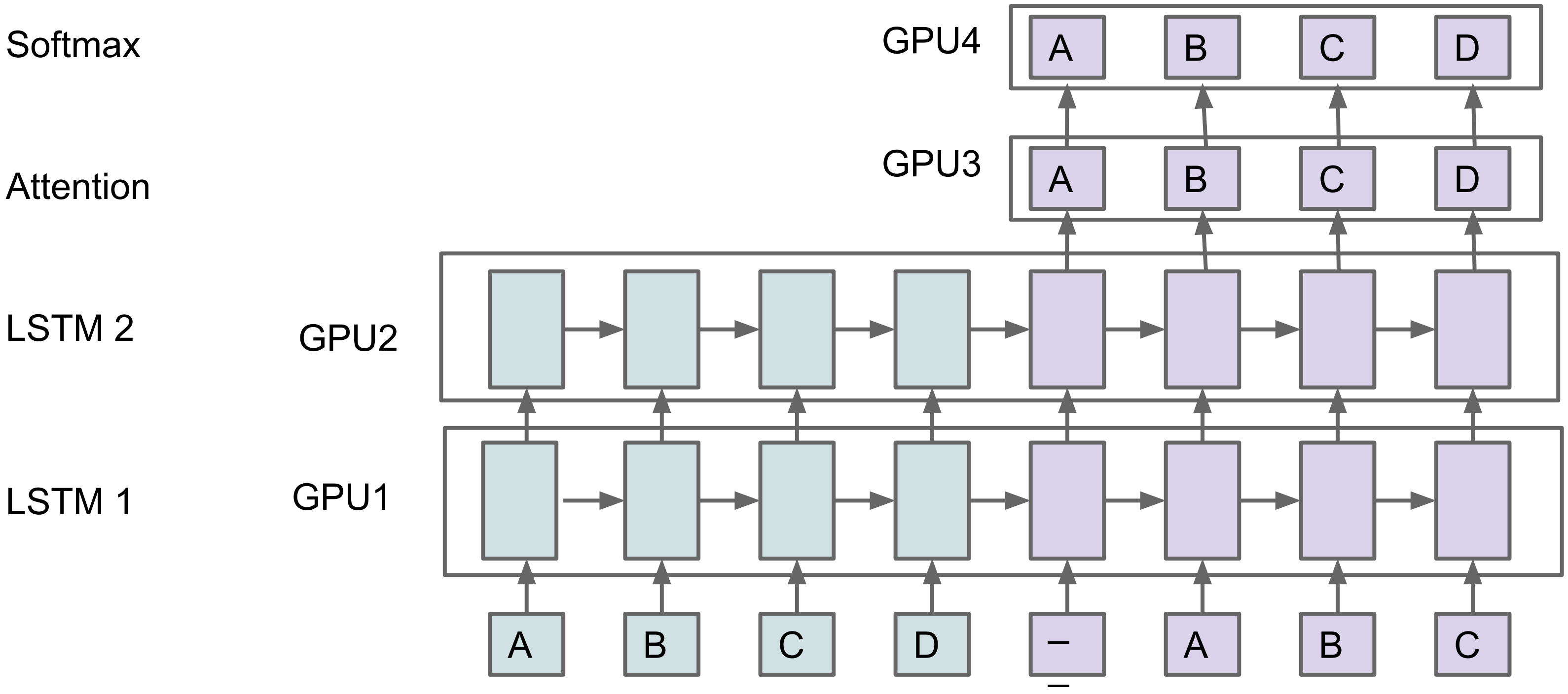
For large models, model parallelism is important

For large models, model parallelism is important

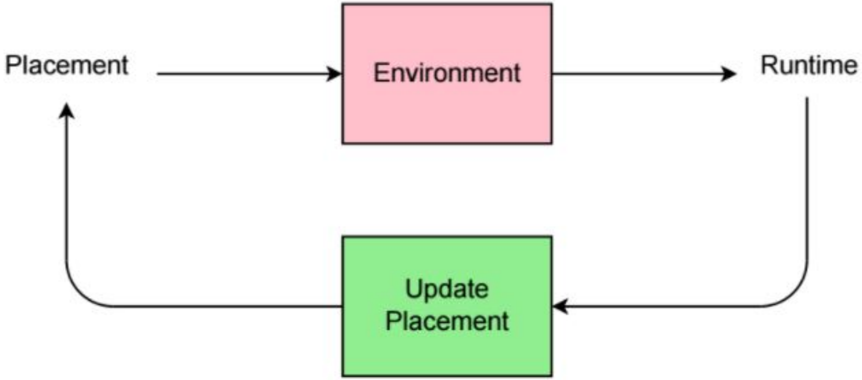
But getting good performance given multiple computing devices is non-trivial and non-obvious



Softmax
A
B
C
D
Attention
A
B
C
D
LSTM 2
LSTM 1
A
B
C
D
_
A
B
C

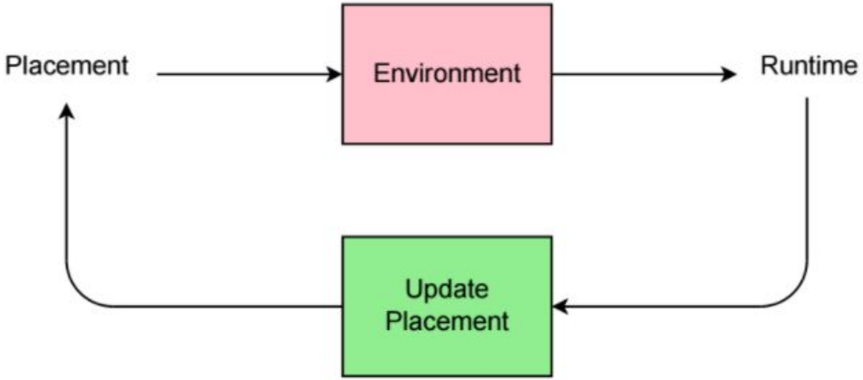

Softmax
Attention
LSTM 2
LSTM 1
GPU4
GPU3
GPU2
GPU1
A
B
C
D
A
B
C
D
A
B
C
D
_
A
B
C

Reinforcement Learning for Higher Performance Machine Learning Models



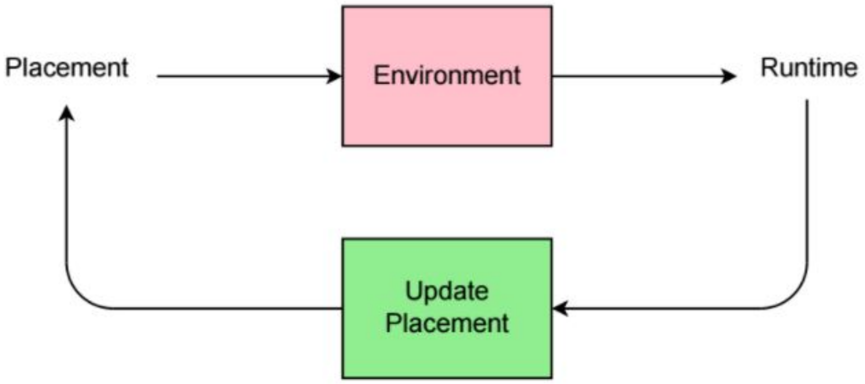
Reinforcement Learning for Higher Performance Machine Learning Models

Placement model (trained via RL) gets graph as input + set of devices, outputs device placement for each graph node



Reinforcement Learning for Higher Performance Machine Learning Models

Placement model (trained via RL) gets graph as input + set of devices, outputs device placement for each graph node



Measured time per step gives RL reward signal

Early results, but it seems to work

Per-step running times (secs)

Model	Hardware	Baseline	RL	Speedup
Neural MT (2 layers) + attention	4 Tesla K80	3.20s	**2.47s**	22.8%
Inception	4 Tesla K80	4.60s	**3.85s**	16.3%

Baselines:

NMT: human expert placement shown on earlier slide

Inception: default placement on GPU/0

Early results, but it seems to work

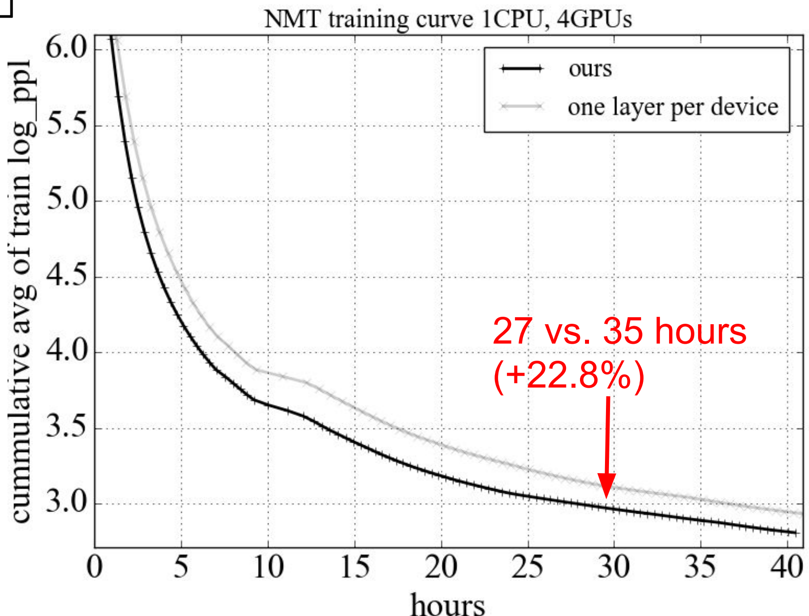
Per-step running times (secs)

Model	Hardware	Baseline	RL	Speedup
Neural MT (2 layers) + attention	4 Tesla K80	3.20s	**2.47s**	22.8%
Inception	4 Tesla K80	4.60s	**3.85s**	16.3%

Baselines:

NMT: human expert placement shown on earlier slide

Inception: default placement on GPU/0



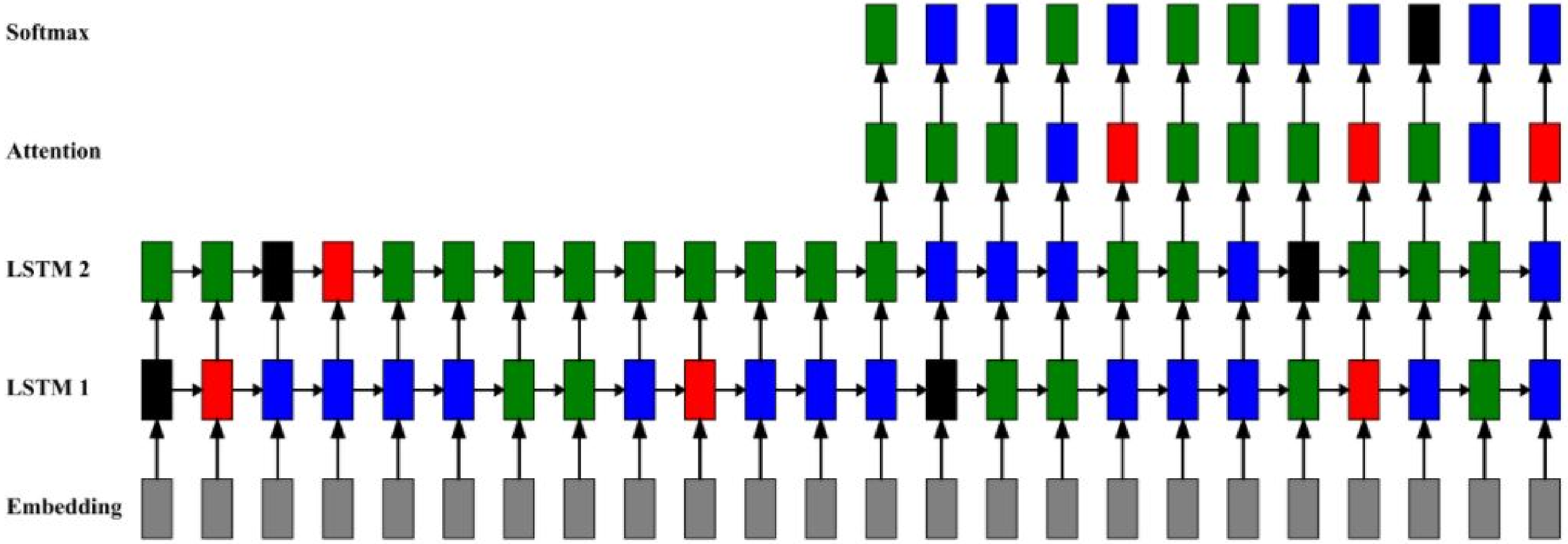
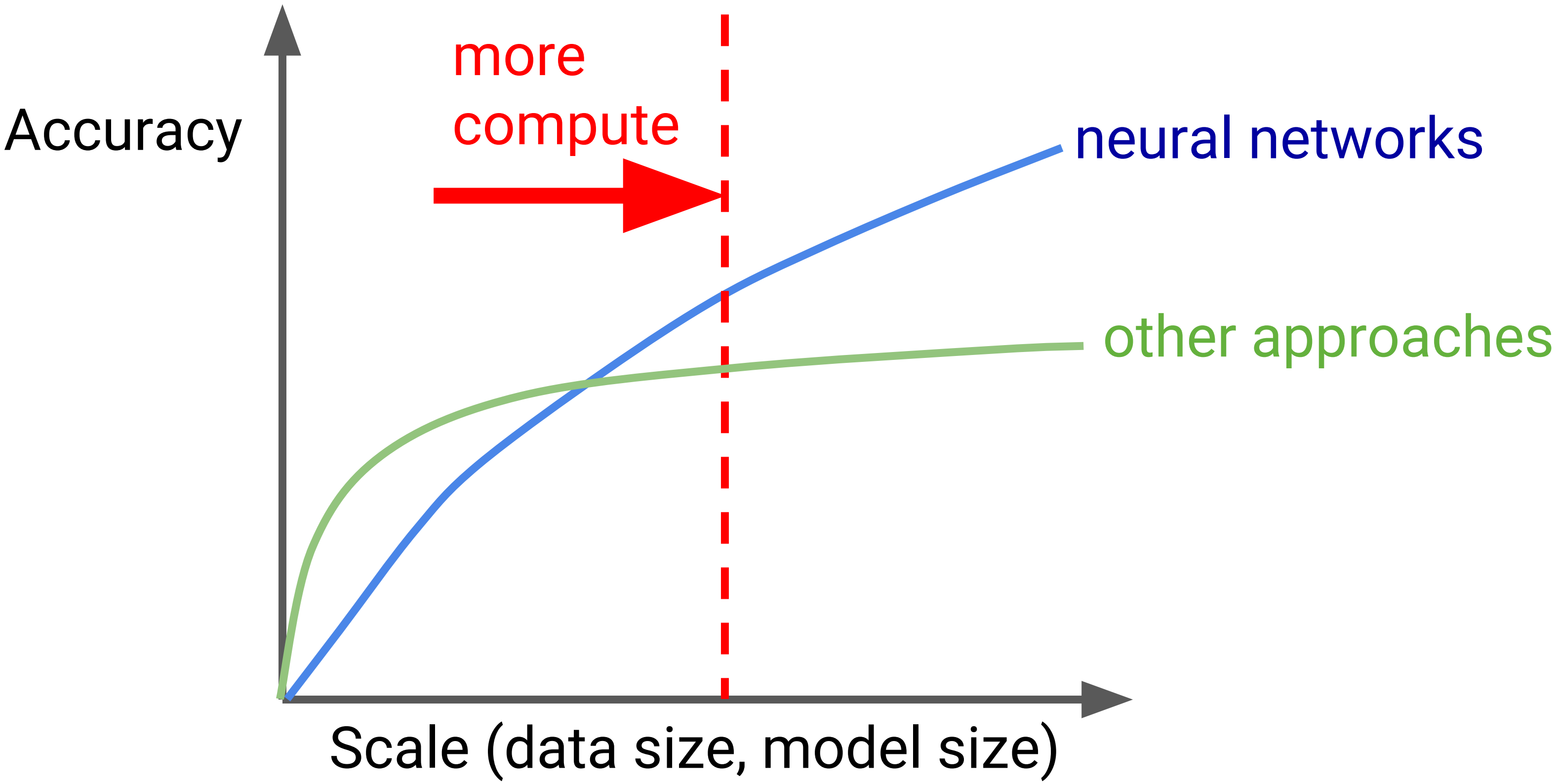
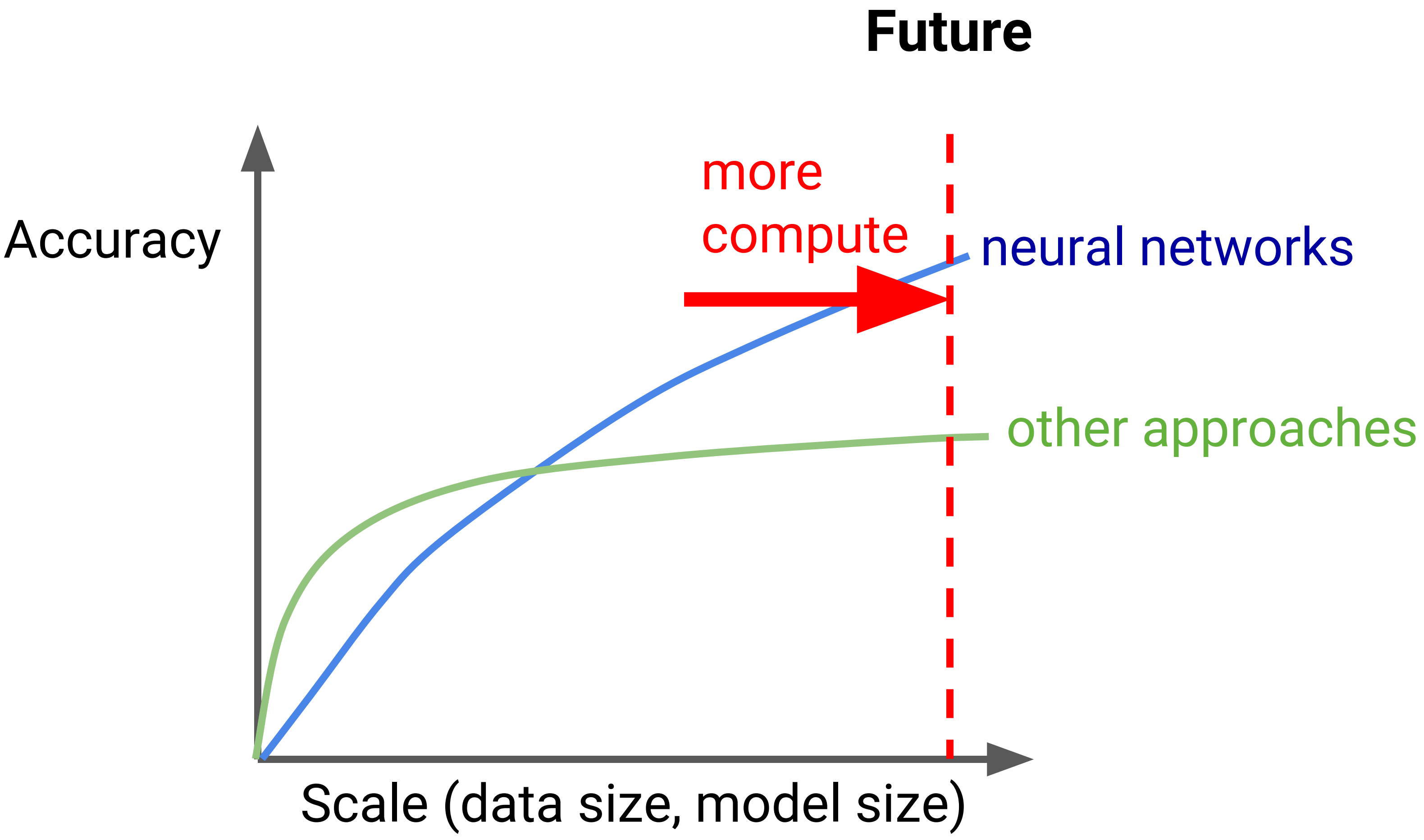


Figure 4: Placement of the NMT graph. Due to space limit, we show only the last 12 steps of the encoder and the first 12 steps of the decoder. Devices are denoted by colors, where gray represents the CPU and each other colors represents a different GPU.

Now


more
compute
Accuracy
neural networks
other approaches
Scale (data size, model size)


Future
more
compute
Accuracy
neural networks
other approaches
Scale (data size, model size)

Example queries of the future

Which of these eye images shows symptoms of diabetic retinopathy?

Describe this video in Spanish

Please fetch me a cup of tea from the kitchen

Find me documents related to reinforcement learning for robotics and summarize them in German

Conclusions

Deep neural networks are making significant strides in speech, vision, language, search, robotics, healthcare, …

If you’re not considering how to use deep neural nets to solve your problems, **you almost certainly should be**

g.co/brain

More info about our work

Main Research Areas

- Machine Learning Algorithms and Techniques
- Healthcare
- Computer Systems for Machine Learning
- Robotics
- Natural Language Understanding
- Music and Art Generation
- Perception

MORE PAPERS

BLOG POSTS

g.co/brain

Join the Team

Full Time Roles

We're looking for talented research scientists and software engineers enthusiastic about deep learning to join us.

VIEW JOBS

Brain Residency

This 12-month program is designed to jumpstart your career in deep learning, working with our scientists and engineers from the Google Brain Team.

VIEW RESIDENCY

Visiting Faculty

Visiting Faculty work closely with our scientists and engineers, and have the opportunity to explore projects at industrial scale with state-of-the-art technology.

VIEW VISITING FACULTY

Interns

Our interns work on projects utilizing the latest techniques in deep learning. In your application, indicate your research interests in the 'Cover letter/other notes' section, so it can be routed to the appropriate recruiter.

VIEW INTERNSHIPS

Thanks!