Deep Learning and Hardware: Matching the Demands from the Machine Learning Community

Ekapol Chuangsuwanich
Department of Computer Engineering, Chulalongkorn University

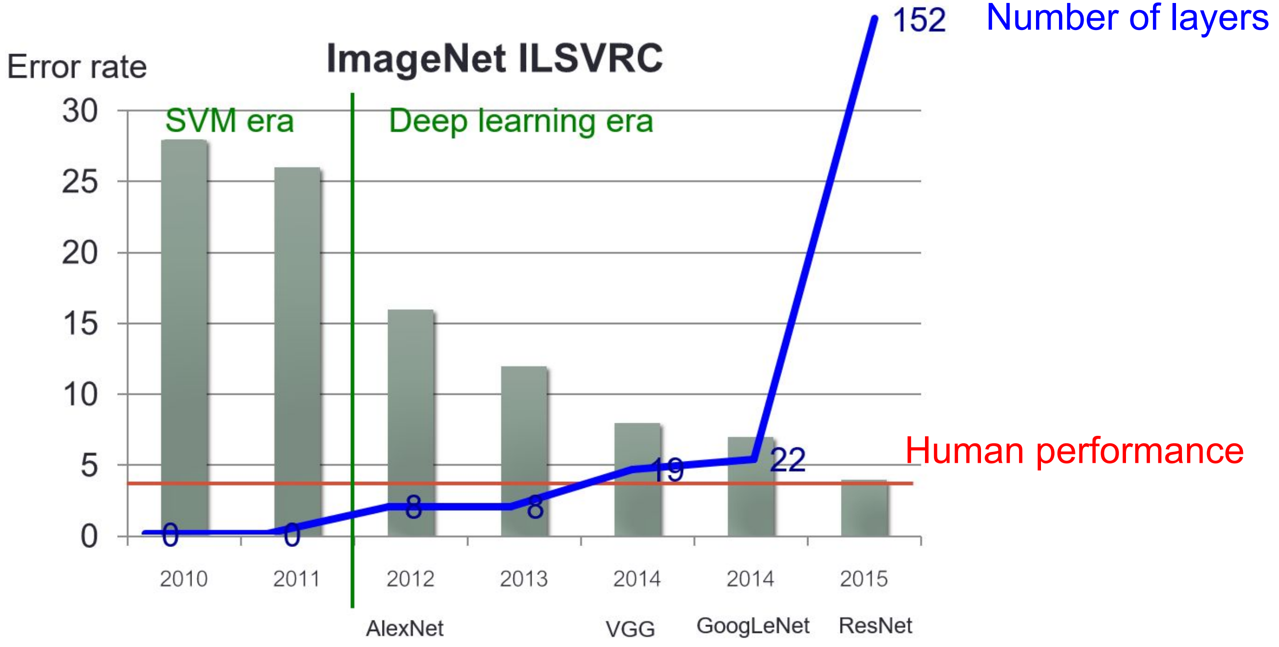
Deep learning

Artificial Neural Networks rebranded

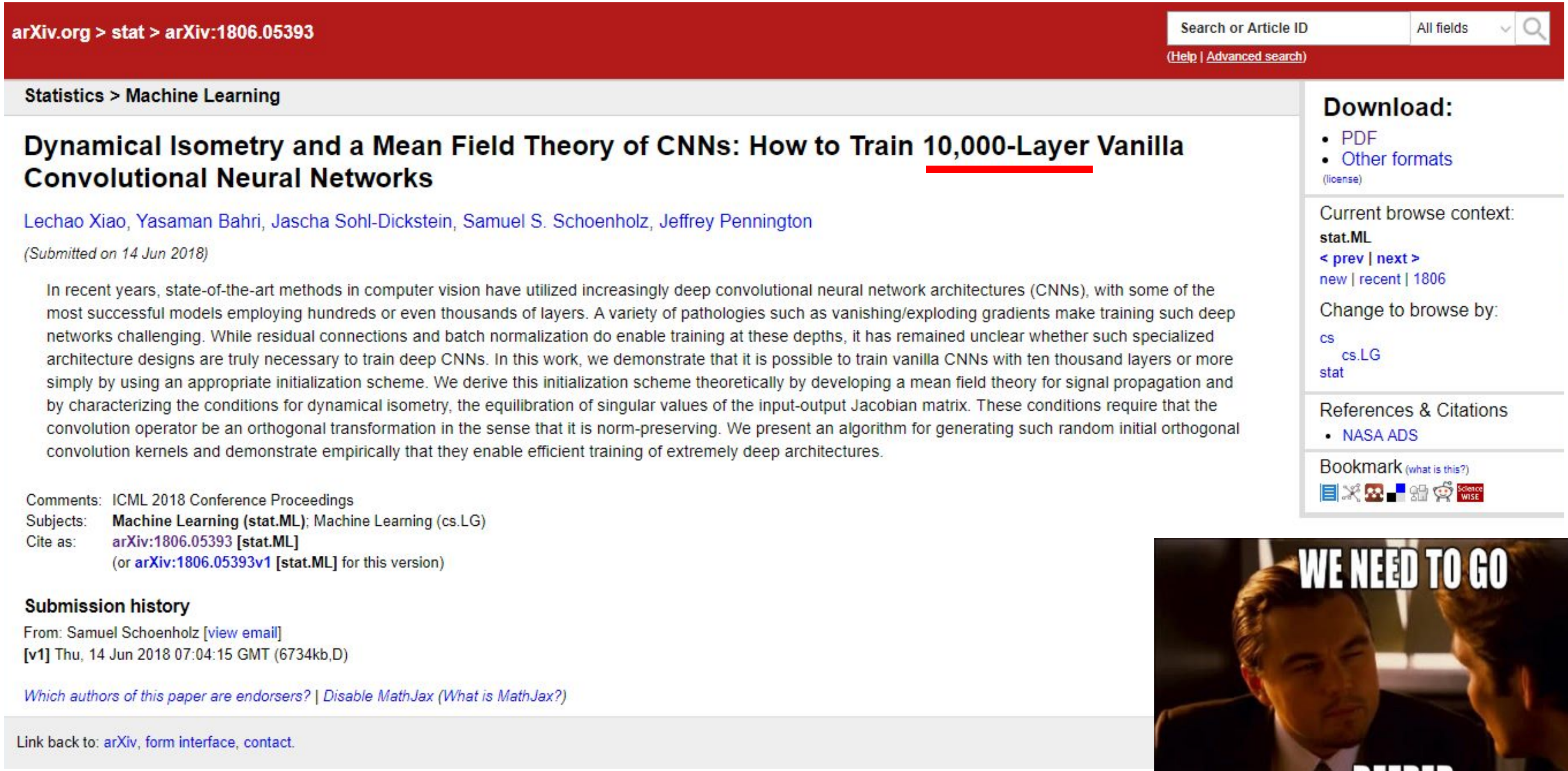
- Deeper models
- Bigger data
- Larger compute

By the end of this talk, I should be able to convince you why all of the big names in Deep learning went to big companies

Wider and deeper models



Olga Russakovsky, ImageNet Large Scale Visual Recognition Challenge, 2014 https://arxiv.org/abs/1409.0575



arXiv.org > stat > arXiv:1806.05393

Search or Article ID | All fields

(Help | Advanced search)

Statistics > Machine Learning

Dynamical Isometry and a Mean Field Theory of CNNs: How to Train 10,000-Layer Vanilla Convolutional Neural Networks

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S. Schoenholz, Jeffrey Pennington

(Submitted on 14 Jun 2018)

In recent years, state-of-the-art methods in computer vision have utilized increasingly deep convolutional neural network architectures (CNNs), with some of the most successful models employing hundreds or even thousands of layers. A variety of pathologies such as vanishing/exploding gradients make training such deep networks challenging. While residual connections and batch normalization do enable training at these depths, it has remained unclear whether such specialized architecture designs are truly necessary to train deep CNNs. In this work, we demonstrate that it is possible to train vanilla CNNs with ten thousand layers or more simply by using an appropriate initialization scheme. We derive this initialization scheme theoretically by developing a mean field theory for signal propagation and by characterizing the conditions for dynamical isometry, the equilibration of singular values of the input-output Jacobian matrix. These conditions require that the convolution operator be an orthogonal transformation in the sense that it is norm-preserving. We present an algorithm for generating such random initial orthogonal convolution kernels and demonstrate empirically that they enable efficient training of extremely deep architectures.

Comments:	ICML 2018 Conference Proceedings
Subjects:	**Machine Learning (stat.ML)**; Machine Learning (cs.LG)
Cite as:	arXiv:1806.05393 **[stat.ML]**
	(or **arXiv:1806.05393v1 [stat.ML]** for this version)

Submission history

From: Samuel Schoenholz [view email]
[v1] Thu, 14 Jun 2018 07:04:15 GMT (6734kb,D)

Which authors of this paper are endorsers? | Disable MathJax (What is MathJax?)

Link back to: arXiv, form interface, contact.

Download:
- PDF
- Other formats

(license)

Current browse context:
stat.ML
< prev | next >
new | recent | 1806

Change to browse by:
cs
cs.LG
stat

References & Citations
- NASA ADS

Bookmark (what is this?)

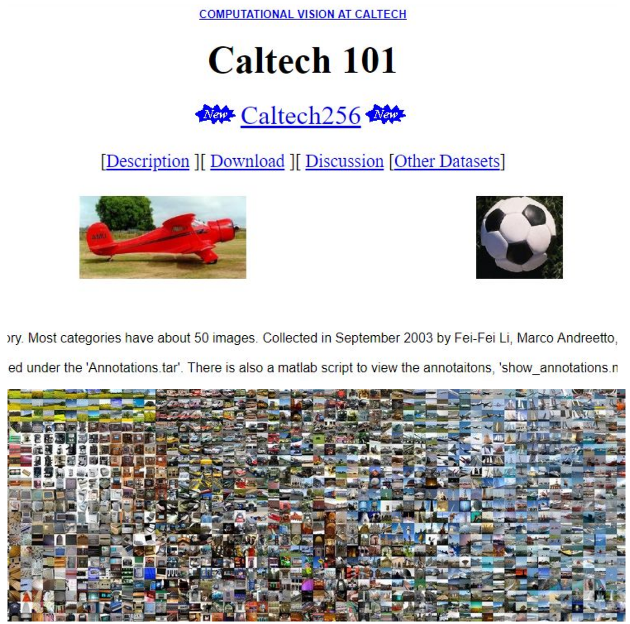
Bigger data

Vision related

Caltech101 (2004) 130 MB

ImageNet Object Class Challenge (2012) 2 GB

BDD100K (2018) 1.8 TB



COMPUTATIONAL VISION AT CALTECH

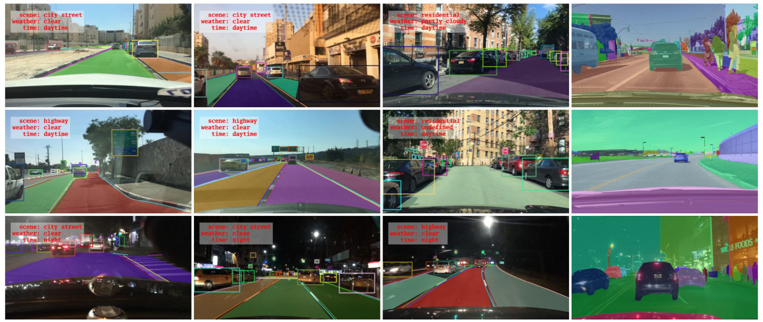
Caltech 101

New Caltech256 New

[Description][Download][Discussion [Other Datasets]

ory. Most categories have about 50 images. Collected in September 2003 by Fei-Fei Li, Marco Andreetto,

ed under the 'Annotations.tar'. There is also a matlab script to view the annotaitons, 'show_annotations.n

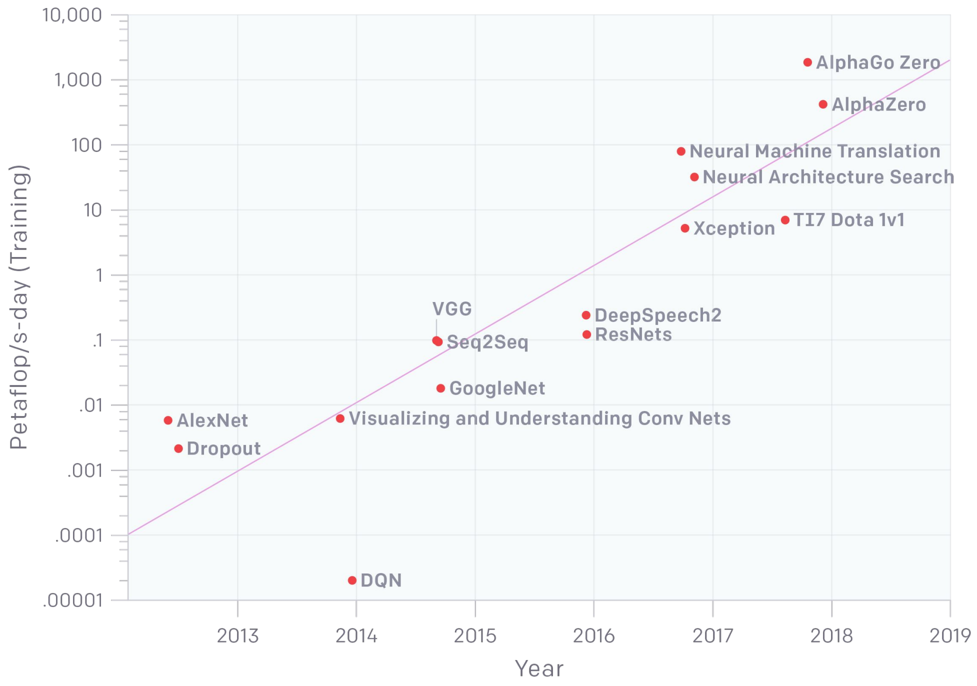


http://www.vision.caltech.edu/Image_Datasets/Caltech101/
http://www.image-net.org/
http://bair.berkeley.edu/blog/2018/05/30/bdd/

Larger Compute

Note that the biggest models are self-taught (RL).

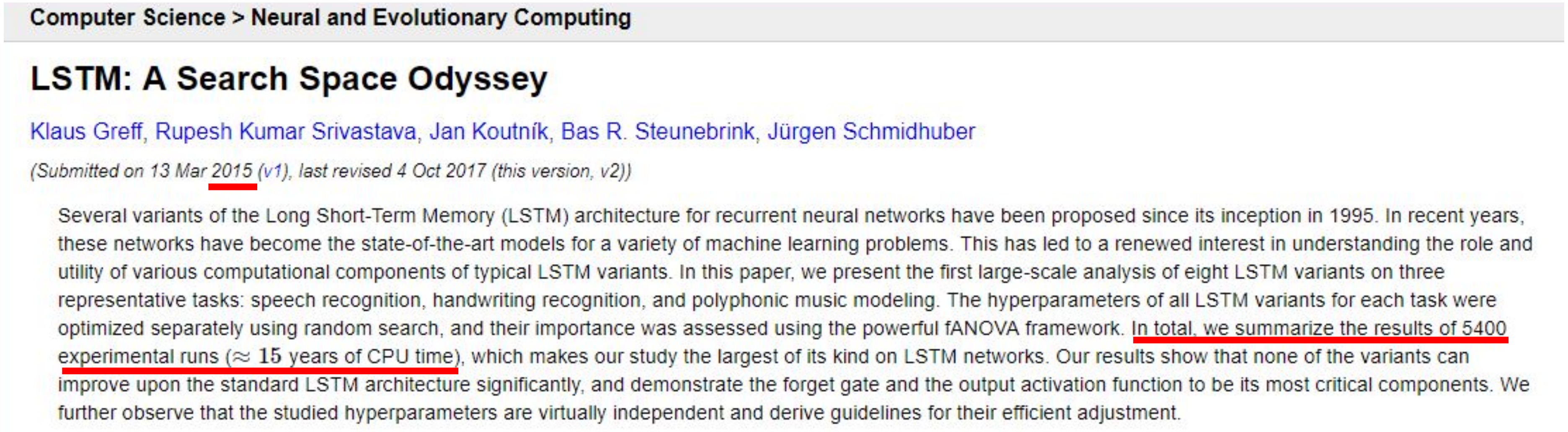
AlexNet to AlphaGo Zero: A 300,000x Increase in Compute



Compute time doubles every ~3 months.

https://blog.openai.com/ai-and-compute/

Deep learning research requires infra



Computer Science > Neural and Evolutionary Computing

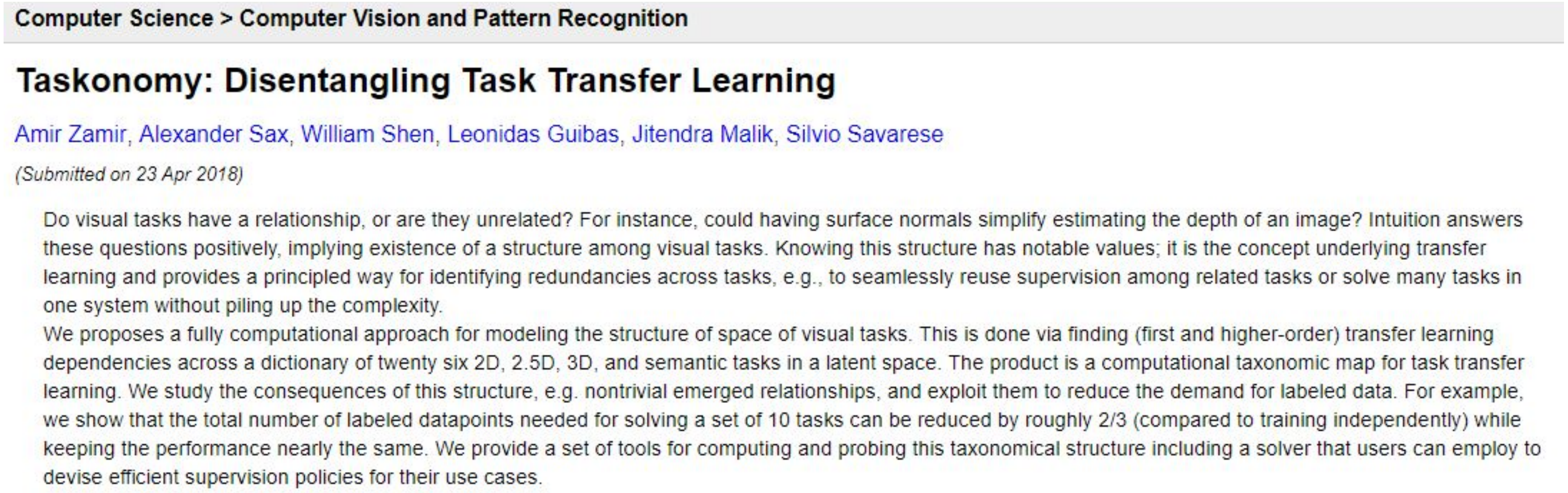
LSTM: A Search Space Odyssey

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, Jürgen Schmidhuber

(Submitted on 13 Mar 2015 (v1), last revised 4 Oct 2017 (this version, v2))

Several variants of the Long Short-Term Memory (LSTM) architecture for recurrent neural networks have been proposed since its inception in 1995. In recent years, these networks have become the state-of-the-art models for a variety of machine learning problems. This has led to a renewed interest in understanding the role and utility of various computational components of typical LSTM variants. In this paper, we present the first large-scale analysis of eight LSTM variants on three representative tasks: speech recognition, handwriting recognition, and polyphonic music modeling. The hyperparameters of all LSTM variants for each task were optimized separately using random search, and their importance was assessed using the powerful fANOVA framework. In total, we summarize the results of 5400 experimental runs (≈ 15 years of CPU time), which makes our study the largest of its kind on LSTM networks. Our results show that none of the variants can improve upon the standard LSTM architecture significantly, and demonstrate the forget gate and the output activation function to be its most critical components. We further observe that the studied hyperparameters are virtually independent and derive guidelines for their efficient adjustment.

Deep learning research requires infra



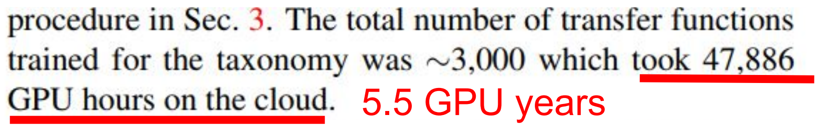
Computer Science > Computer Vision and Pattern Recognition

Taskonomy: Disentangling Task Transfer Learning

Amir Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, Silvio Savarese

(Submitted on 23 Apr 2018)

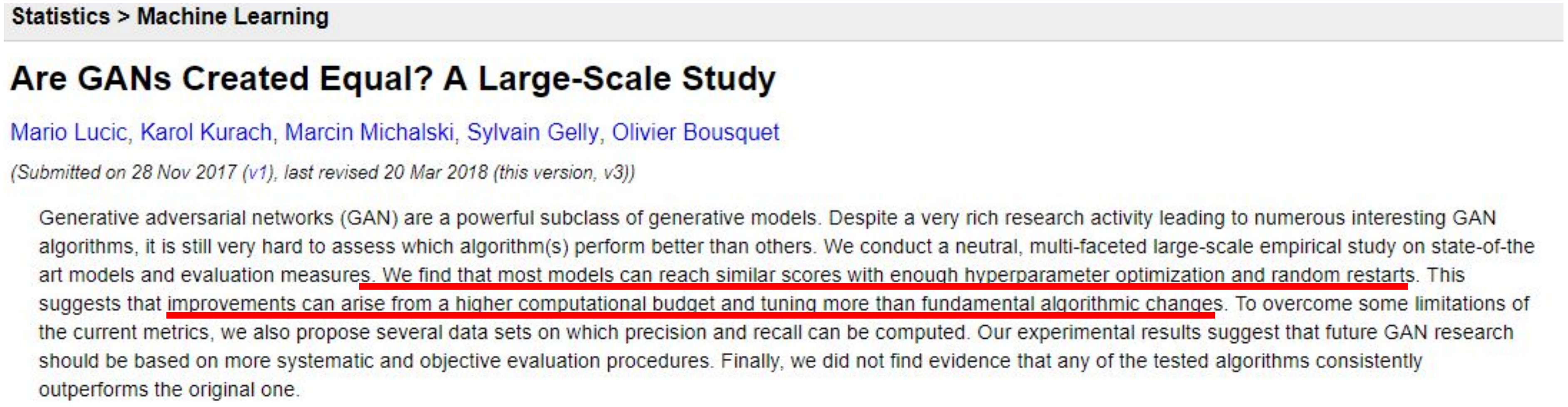
Do visual tasks have a relationship, or are they unrelated? For instance, could having surface normals simplify estimating the depth of an image? Intuition answers these questions positively, implying existence of a structure among visual tasks. Knowing this structure has notable values; it is the concept underlying transfer learning and provides a principled way for identifying redundancies across tasks, e.g., to seamlessly reuse supervision among related tasks or solve many tasks in one system without piling up the complexity.
We proposes a fully computational approach for modeling the structure of space of visual tasks. This is done via finding (first and higher-order) transfer learning dependencies across a dictionary of twenty six 2D, 2.5D, 3D, and semantic tasks in a latent space. The product is a computational taxonomic map for task transfer learning. We study the consequences of this structure, e.g. nontrivial emerged relationships, and exploit them to reduce the demand for labeled data. For example, we show that the total number of labeled datapoints needed for solving a set of 10 tasks can be reduced by roughly 2/3 (compared to training independently) while keeping the performance nearly the same. We provide a set of tools for computing and probing this taxonomical structure including a solver that users can employ to devise efficient supervision policies for their use cases.



procedure in Sec. 3. The total number of transfer functions trained for the taxonomy was ∼3,000 which took 47,886 GPU hours on the cloud.

5.5 GPU years

Deep learning research requires infra



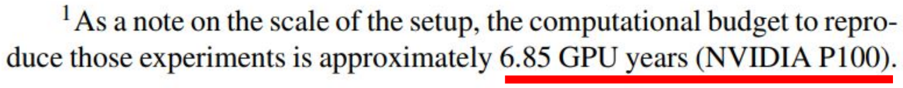
Statistics > Machine Learning

Are GANs Created Equal? A Large-Scale Study

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, Olivier Bousquet

(Submitted on 28 Nov 2017 (v1), last revised 20 Mar 2018 (this version, v3))

Generative adversarial networks (GAN) are a powerful subclass of generative models. Despite a very rich research activity leading to numerous interesting GAN algorithms, it is still very hard to assess which algorithm(s) perform better than others. We conduct a neutral, multi-faceted large-scale empirical study on state-of-the art models and evaluation measures. We find that most models can reach similar scores with enough hyperparameter optimization and random restarts. This suggests that improvements can arise from a higher computational budget and tuning more than fundamental algorithmic changes. To overcome some limitations of the current metrics, we also propose several data sets on which precision and recall can be computed. Our experimental results suggest that future GAN research should be based on more systematic and objective evaluation procedures. Finally, we did not find evidence that any of the tested algorithms consistently outperforms the original one.



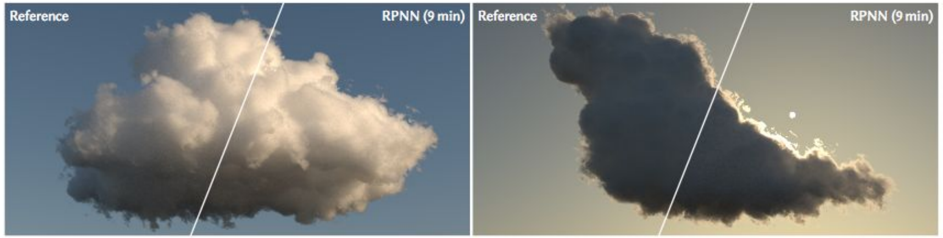
[1] As a note on the scale of the setup, the computational budget to reproduce those experiments is approximately 6.85 GPU years (NVIDIA P100).

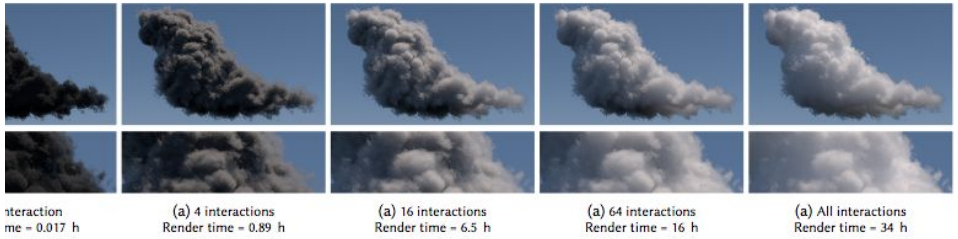
Frontier deep learning research requires

- Clouds
  - Not just any clouds but clouds of GPUs
  - And sometimes traditional CPU clouds too

Frontier deep learning research requires

- Clouds
  - Not just any clouds but clouds of GPUs
  - And sometimes traditional CPU clouds too

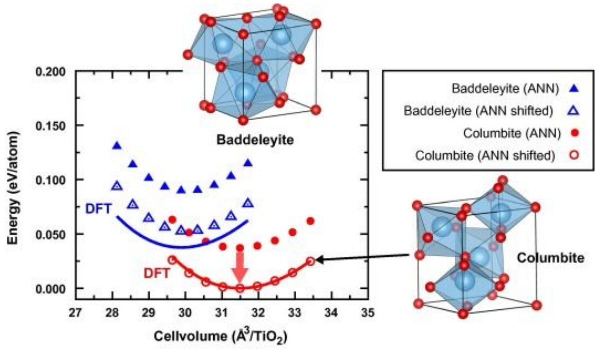




Simon Kallweit, et al. “Deep Scattering: Rendering Atmospheric Clouds with Radiance-Predicting Neural Networks” SIGGRAPH Asia 2018

Frontier deep learning research requires

- Clouds
  - Not just any clouds but clouds of GPUs
  - And sometimes traditional CPU clouds too



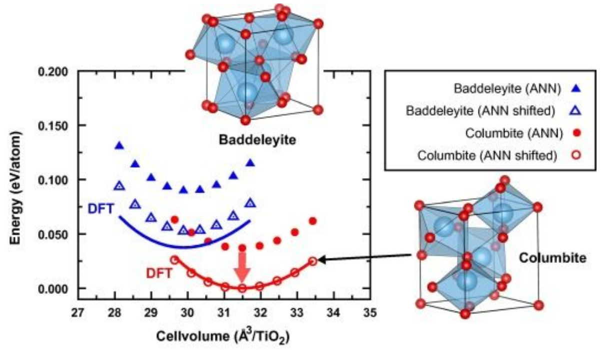
Nongnuch Artrith, et al. “**An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for TiO2**” 2016

Jonathan Tompson, et al. “**Accelerating Eulerian Fluid Simulation With Convolutional Networks**” 2016

Frontier deep learning research requires

- Clouds
  - Not just any clouds but clouds of GPUs
  - And sometimes traditional CPU clouds too

But this is actually the easy part



Nongnuch Artrith, et al. “**An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for TiO2**” 2016

Jonathan Tompson, et al. “**Accelerating Eulerian Fluid Simulation With Convolutional Networks**” 2016

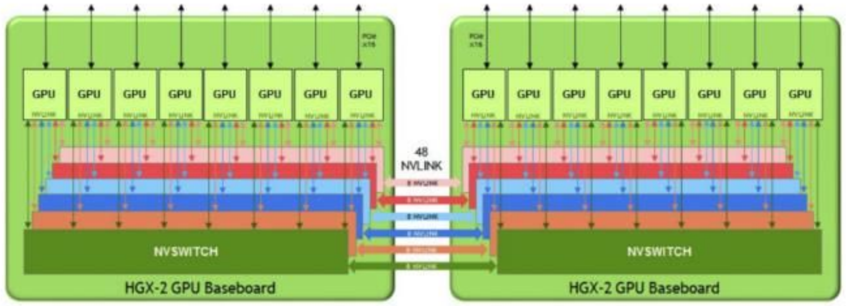
Frontier deep learning research requires

- Clouds
- RAM
  - Big models cannot fit into a single GPU
  - Need ways to split weights into multiple GPUs effectively

https://wccftech.com/nvidia-titan-v-ceo-edition-32-gb-hbm2-ai-graphics-card/

Frontier deep learning research requires

- Clouds
- RAM
- Data transfer
    - Training on multiple GPUs require transfer of weights/feature maps



Frontier deep learning research requires

- Clouds
- RAM
- Data transfer
- Green
  - Low power is prefered even for training
  - Great for inference mode (testing) either on device or in the cloud
  - $$$

Frontier deep learning research requires

- Clouds
- RAM
- Data transfer
- Green

Parallelism

Architecture

Outline

Parallelism

Two main approaches to parallelize deep learning

Data parallel

Model parallel

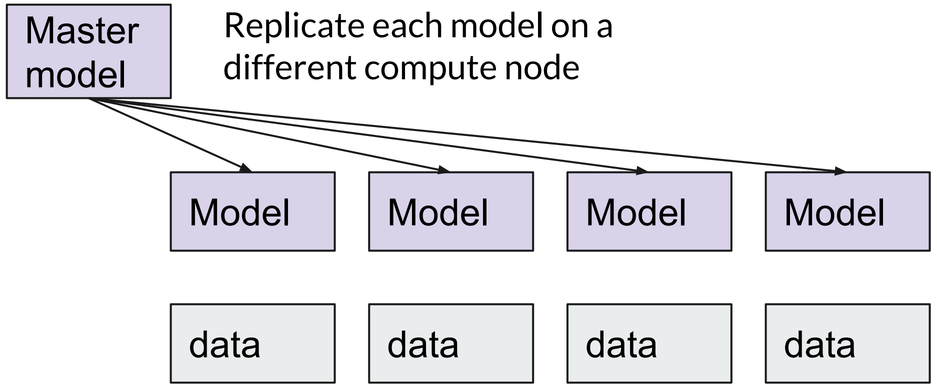
Data parallel

Split the training data into separate batches

Master model
data

Data parallel

Split the training data into separate batches



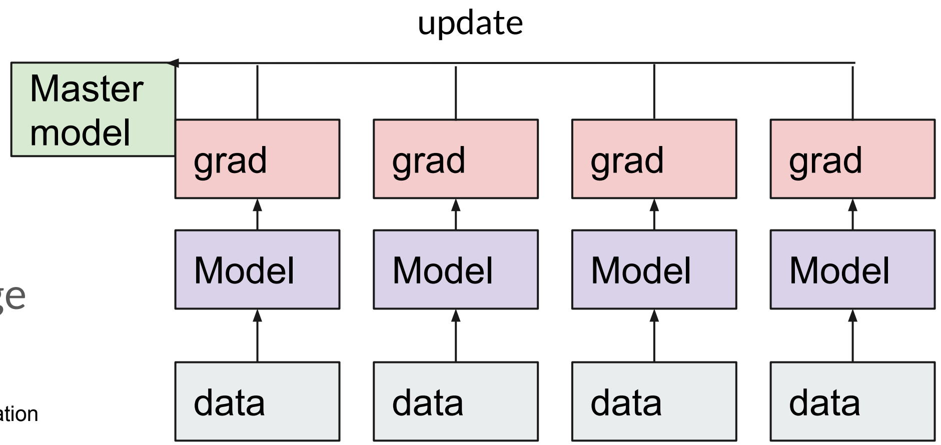
Data parallel

Split the training data into separate batches

Have "merging" step to consolidate

Sends the gradient (better compression/quantization)

Can be considered as a very large mini-batch



Dan Alistarh, et al. "QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding" 2017
Priya Goyal, et al. "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour" 2017

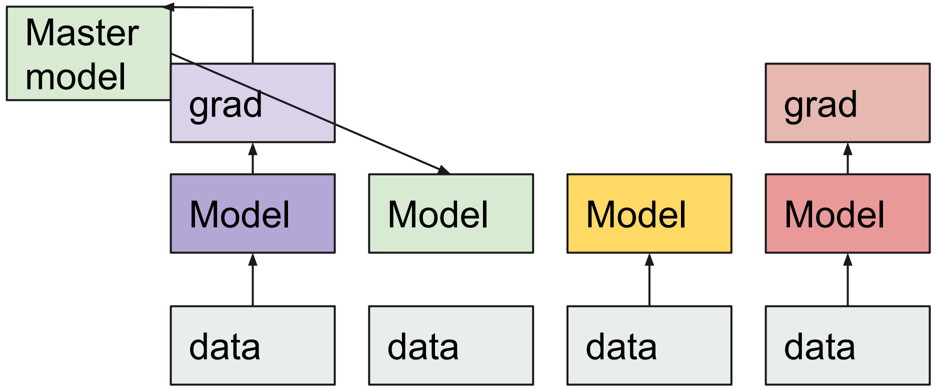
Data parallel

Split the training data into separate batches

Have "merging" step to consolidate

Can be asynchronous

Update and replicate asynchronously



Jeffrey Dean, et al. "Large Scale Distributed Deep Networks" 2012

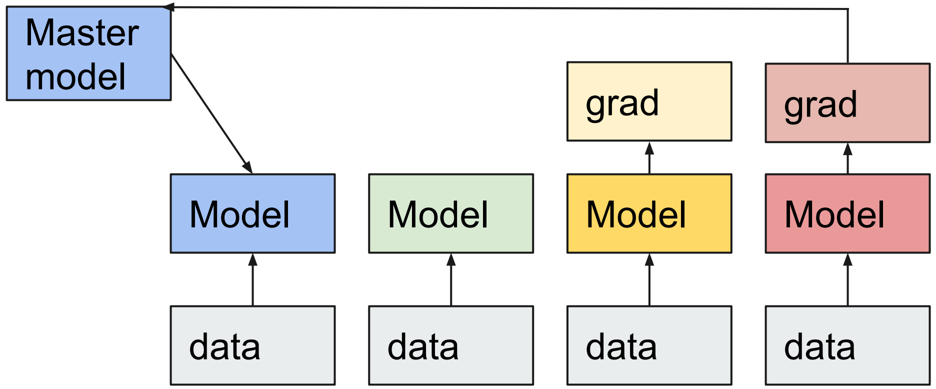
Data parallel

Split the training data into separate batches

Have "merging" step to consolidate

Can be asynchronous

Update and replicate asynchronously



Jeffrey Dean, et al. "Large Scale Distributed Deep Networks" 2012

Data parallel

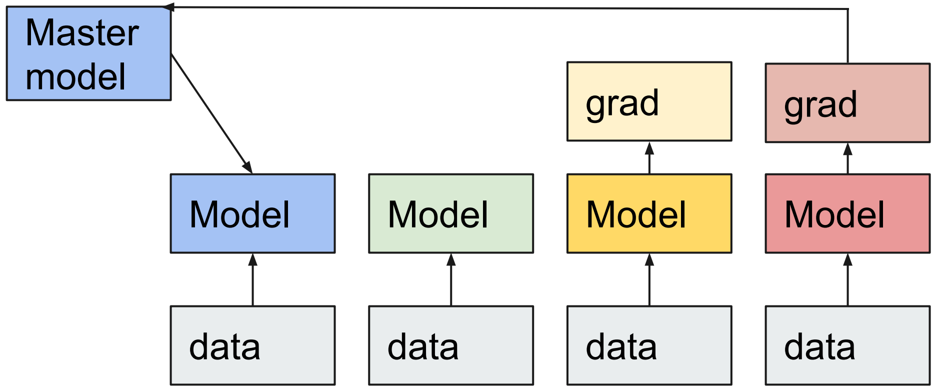
Split the training data into separate batches

Have "merging" step to consolidate

Can be asynchronous

Stale gradient problem

Update and replicate asynchronously



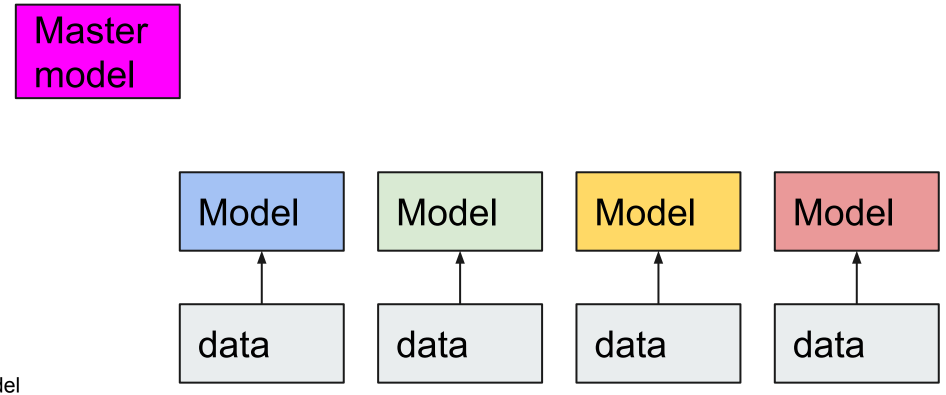
Jeffrey Dean, et al. "Large Scale Distributed Deep Networks" 2012

Data parallel

Some merges at the model level

A form of model averaging/model ensemble

Merge after several steps to reduce transfer overhead



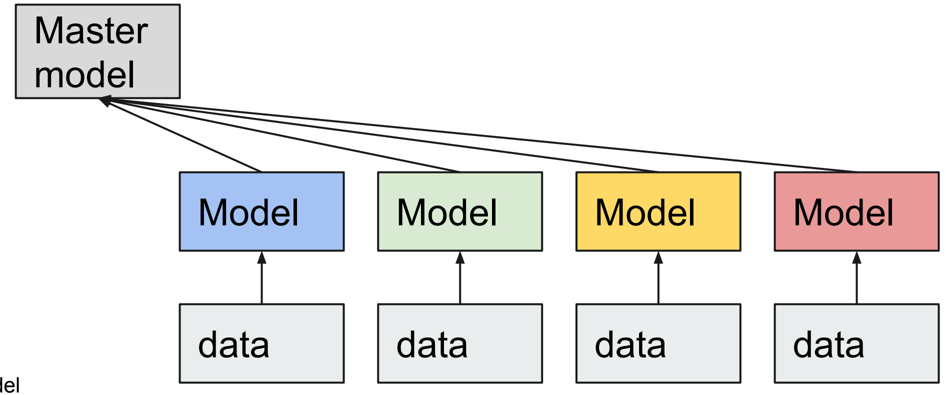
Hang Su, et al. “Experiments on Parallel Training of Deep Neural Network using Model Averaging” 2015

Data parallel

Some merges at the model level

A form of model averaging/model ensemble

Merge after several steps to reduce transfer overhead



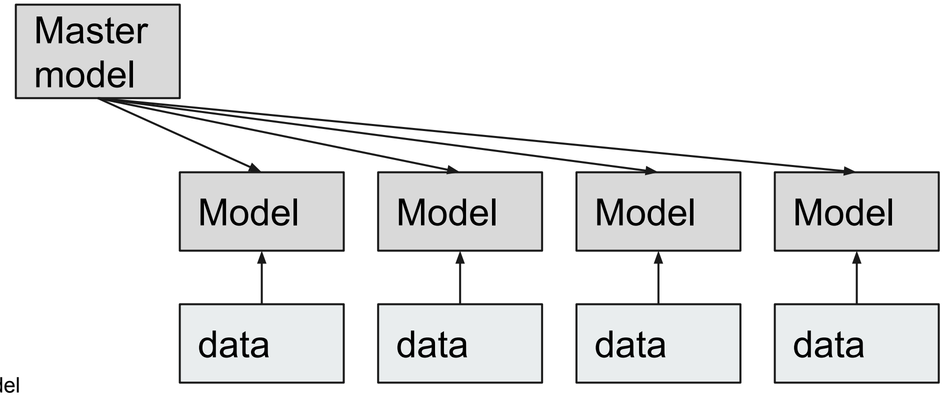
Hang Su, et al. “Experiments on Parallel Training of Deep Neural Network using Model Averaging” 2015

Data parallel

Some merges at the model level

A form of model averaging/model ensemble

Merge after several steps to reduce transfer overhead



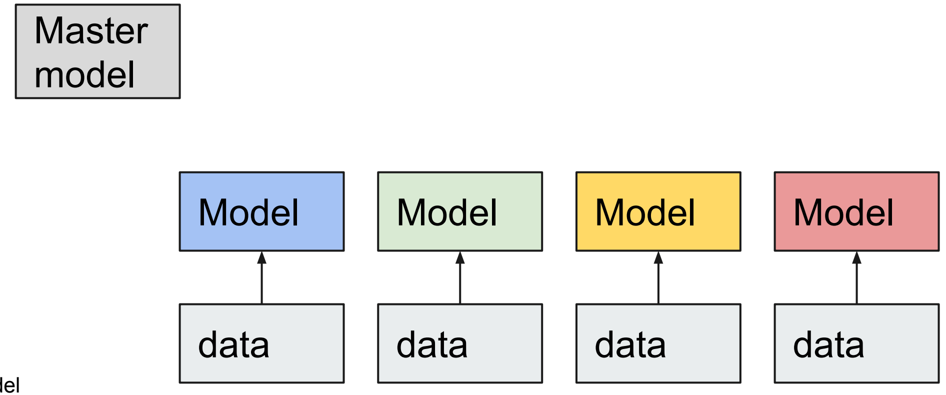
Hang Su, et al. “Experiments on Parallel Training of Deep Neural Network using Model Averaging” 2015

Data parallel

Some merges at the model level

A form of model averaging/model ensemble

Merge after several steps to reduce transfer overhead



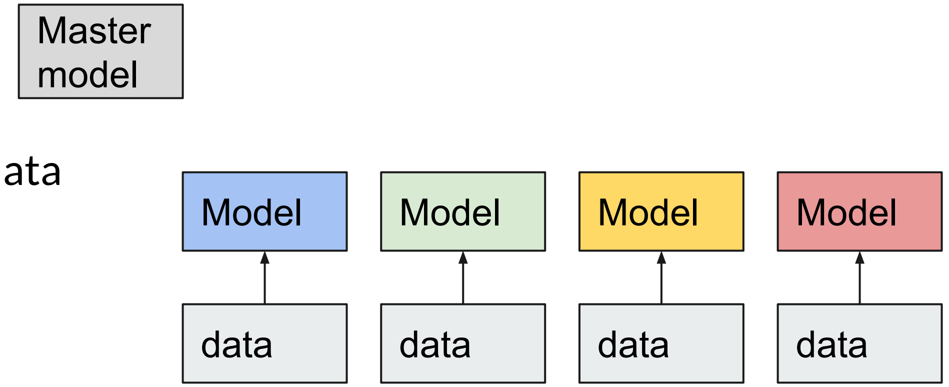
Hang Su, et al. “Experiments on Parallel Training of Deep Neural Network using Model Averaging” 2015

Data parallel: interesting notes

Typically requires tweaking of the original SGD

Final model might actually be better than without parallelization

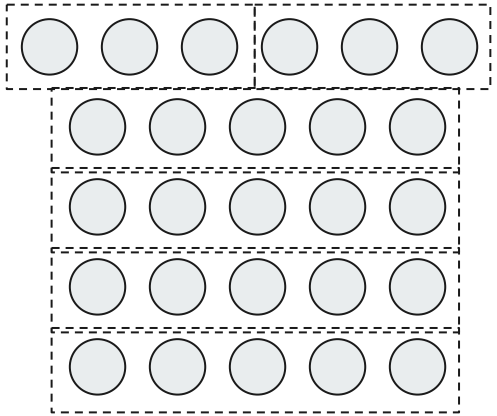
Even with algorithmic optimization data transfer is still the critical path



Model parallel

Split the model into parts each for different compute nodes

Data transfer between nodes is a real concern



Two main approaches to parallelize deep learning

Data parallel

Easy, minimal change in the higher level code

Cannot handle the case when the model is too big to fit on a single GPU

Model parallel

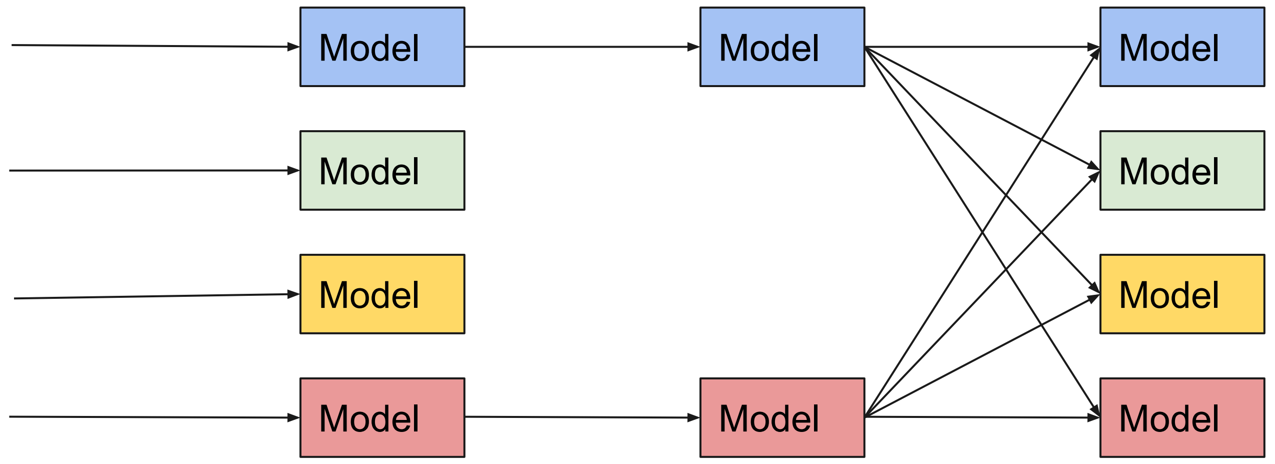
Hard, requires sophisticated changes in both high and low level code

Let’s you fit models bigger than your GPU RAM

People usually use both

Evolutionary algorithms

Embarrassingly parallel
No need for gradient computation
Great fit for RL where gradient is hard to estimate



Randomly initialized models

Evaluate goodness of the models remove the bad ones

Generate a new set of models based on the previous set

Tim Salimans, et al. “Evolution Strategies as a Scalable Alternative to Reinforcement Learning”, 2017

Outline

Re-thinking the architecture

ASICs (TPU)

- Quantization from floating point to fixed-point arithmetic
- Faster than GPU per Watt

Are other numeric representations also possible?

Deep Learning and Logarithmic Number System

In collaboration with Leo Liu, Joe Bates, James Glass, and Singular Computing

Logarithmic Number System

- IEEE floating point format - a real number is represented by the sign, significand, and the exponent

$1.2345 = 12345 * 10^{-4}$

- Logarithmic Number System (LNS) - a real number is represented by its Log value

$\log_2(1.2345) = 0.30392$ (stored as fixed point)

    - Worse precision than IEEE floats

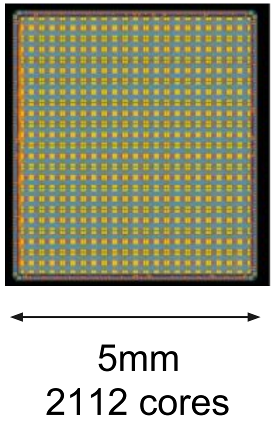
Multiplying/dividing in LNS

Multiplying/dividing in LNS is simply addition/subtraction

$b = \log(B)$, $c = \log(C)$

$\log(B * C) = \log(B) + \log(C) = b + c$

Lots of transistors saved. Smaller and faster per Watt compared to GPUs!



Addition/subtraction in LNS

More complicated

$b = \log(B)$, $c = \log(C)$

$\log(B + C) = \log(B * (1 + C/B)) = \log(B) + \log(1 + C/B) = b + G(c - b)$

$G = \log(1 + 2^x)$ which can be computed efficiently in hardware

Deep learning training with LNS

Simple feed forward network on MNIST

	Validation Error Rate
Normal DNN	2.14%
Matrix multiply with LNS	2.12%
LNS everywhere	3.62%

Smaller weight updates are getting ignored by the low precision

Kahan summation

Weight updates accumulate errors in DNN training

Accumulating the running errors during summation. The total error is added back at the end.

One addition becomes two additions and two substrations with Kahan summation.

With Kahan sum

Simple feed forward network on MNIST

	Validation Error Rate
Normal DNN	2.14%
Matrix multiply with LNS	2.12%
LNS everywhere	3.62%
LNS everywhere with Kahan sum	2.29%

L. Liu, *Acoustic Models for Speech Recognition Using Deep Neural Networks Based on Approximate Math,* M. Thesis, *2015*

Conclusion

Several approaches to make deep learning possible at scale.

To scale deep learning, we need changes to the algorithm, the system, and also the hardware architecture.

Lots of active research from multiple perspectives.