网络结构与网络参数：谁动了人工神经网络模型的奶酪？

Neural Architecture or Network Parameters: who moved the cheese of artificial neural networks?

苏江　Jiang Su

高性能AI研究组 HiPerAIR
暗物智能科技 DMAI Inc.
2020.08.29

以强认知AI平台，提升人类福祉

主讲人

苏江　博士

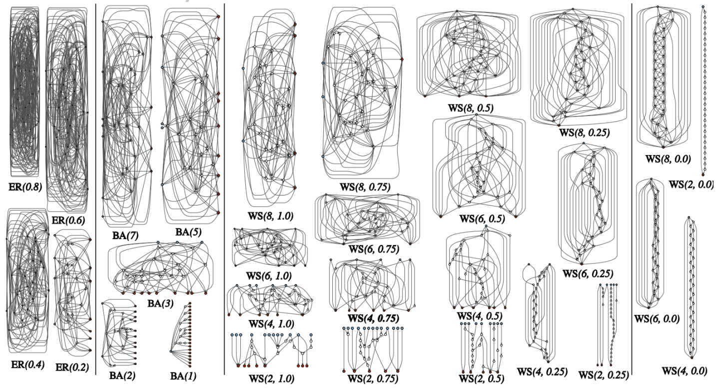
sujiang@dm-ai.cn

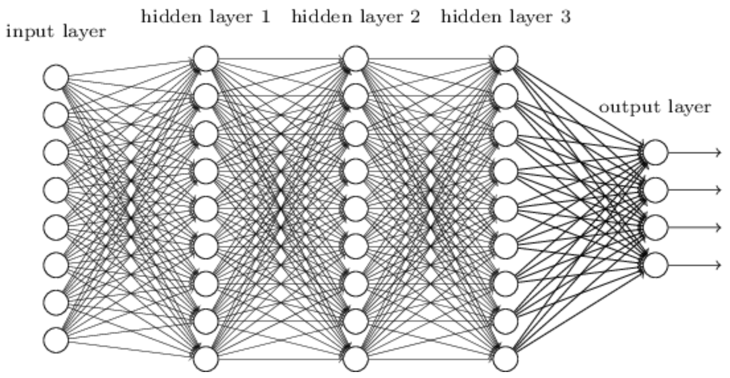
- 英国帝国理工学院 电子电气工程系 博士
- 剑桥大学 计算机学院 助理研究员
- DMAI 高性能AI与硬件技术部 研发副总监

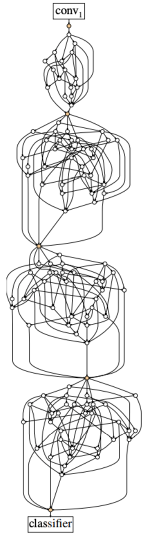
研究兴趣

- AI芯片架构设计
- 深度网络高性能算子
- 深度网络剪枝、量化、压缩

Architecture or Parameters

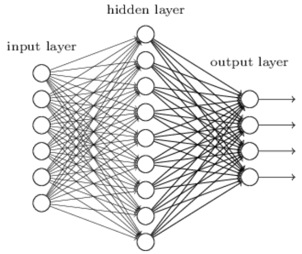






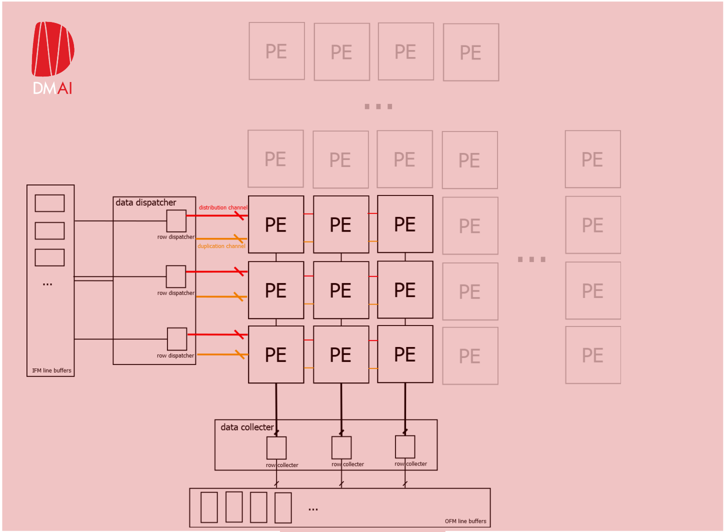
- 自动化结构搜索
- 自动化超参调优
- 模型新结构探索

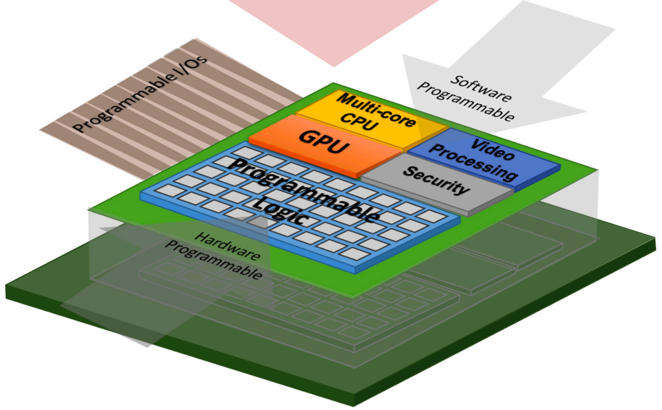
模型精度



- 低精度参数量化
- 冗余神经元剪枝
- 模型存储压缩

执行效率





AI应用

语音识别 | 计算机视觉 | 自然语言处理 | 机器推理

模型优化

AutoML | 量化 | 剪枝 | 压缩

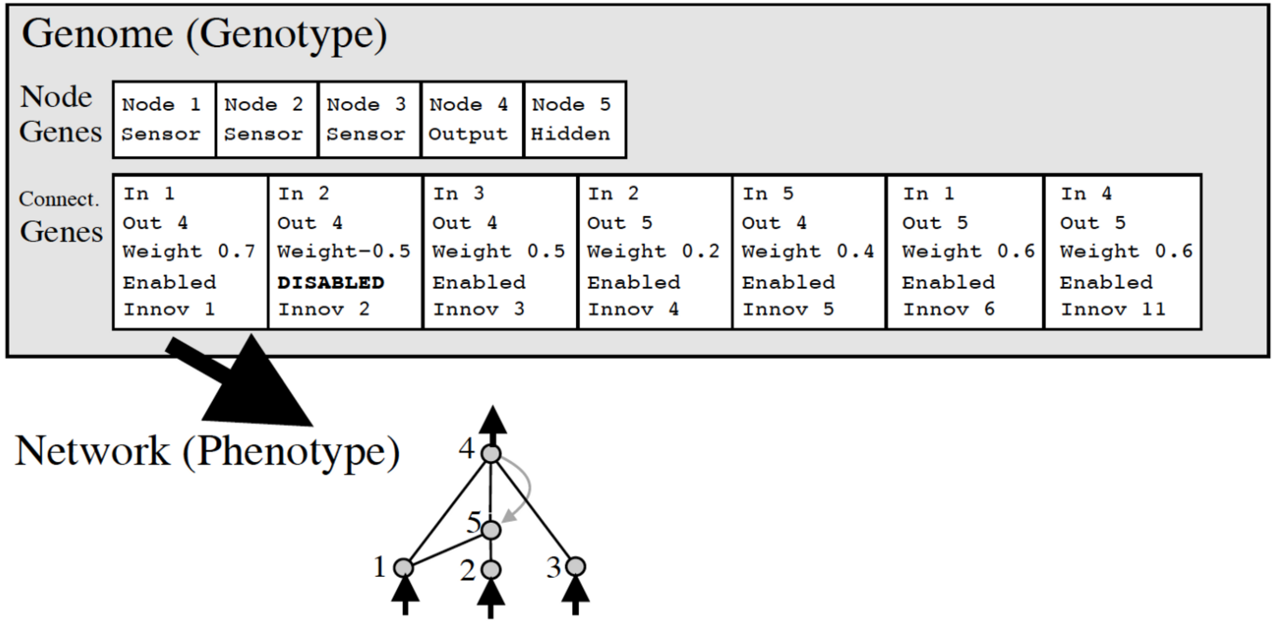
硬件加速平台

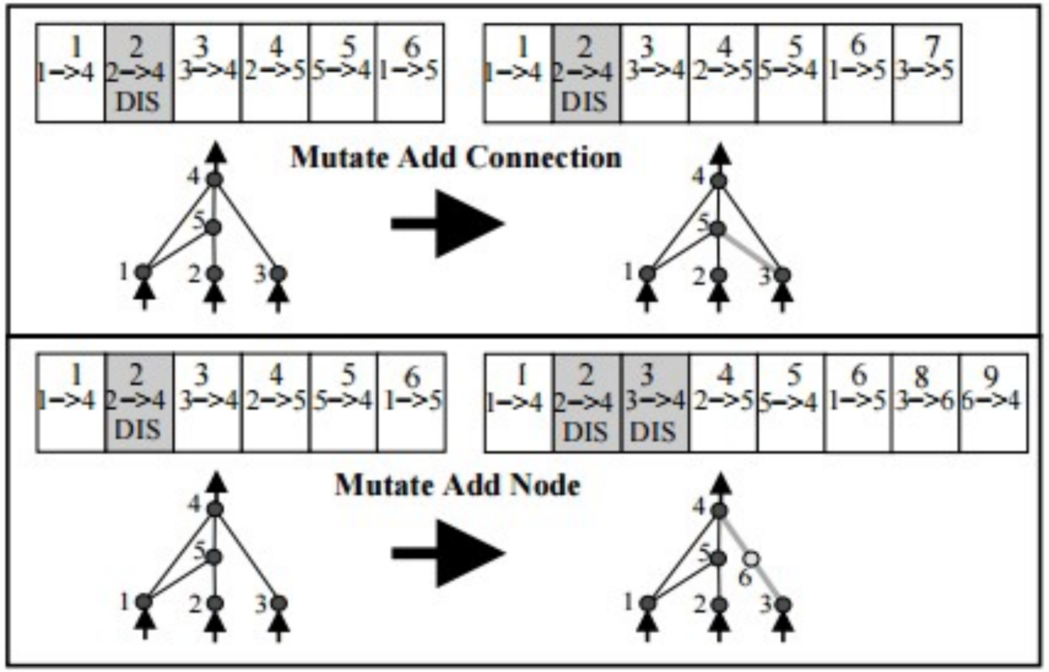
FPGA（端、云） | GPU/CPU（云）

Flexibility ⟷ Performance

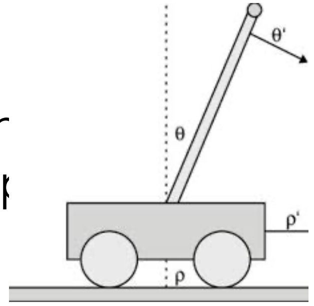
Architecture is more important over parameters

NEAT: EA-based Network Topology Works in Real Problems





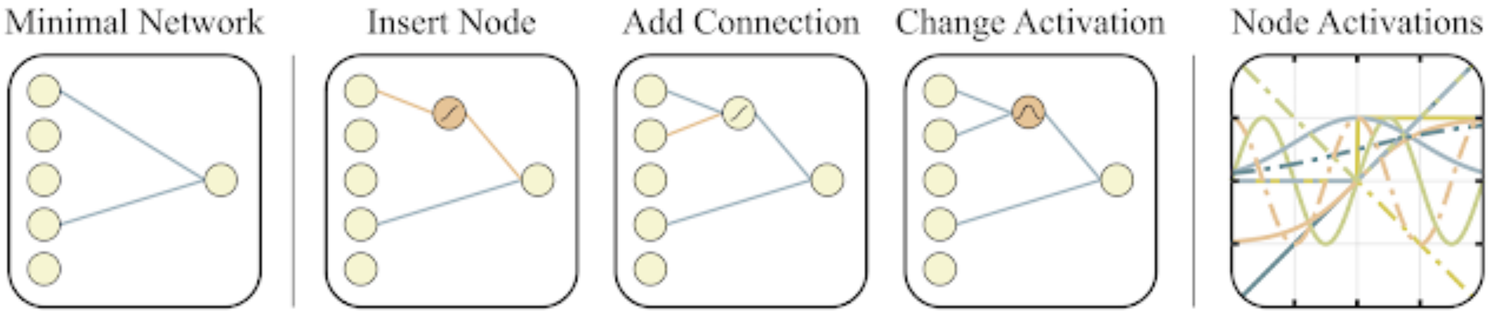
- Instead of train a full net and then de-redundancy, NEAT evolves from minimal baby r
- Weight space explored via crossover or networks weights and mutation of weights/to
- Evolutionary optimization compared to backpropagation

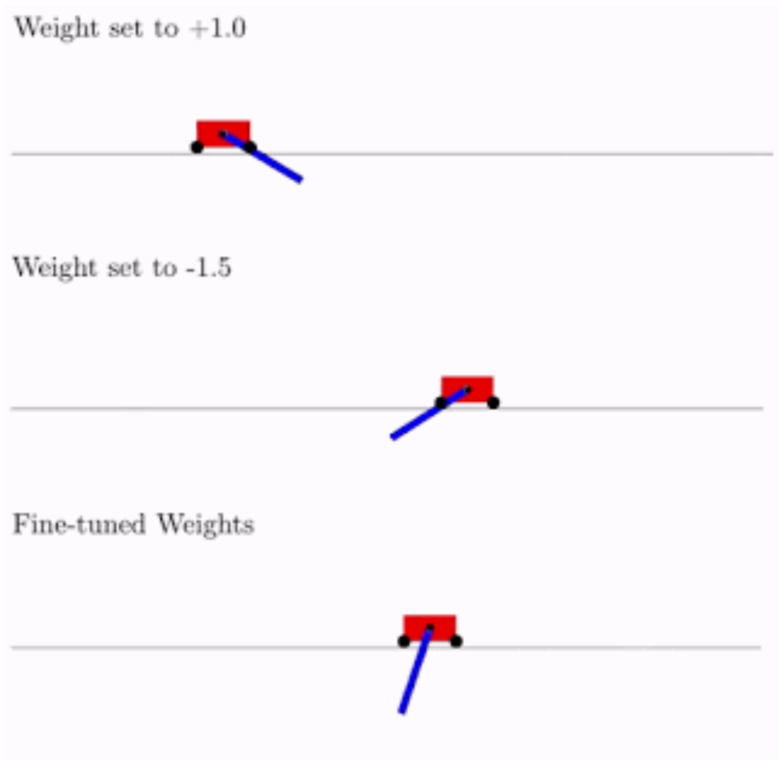


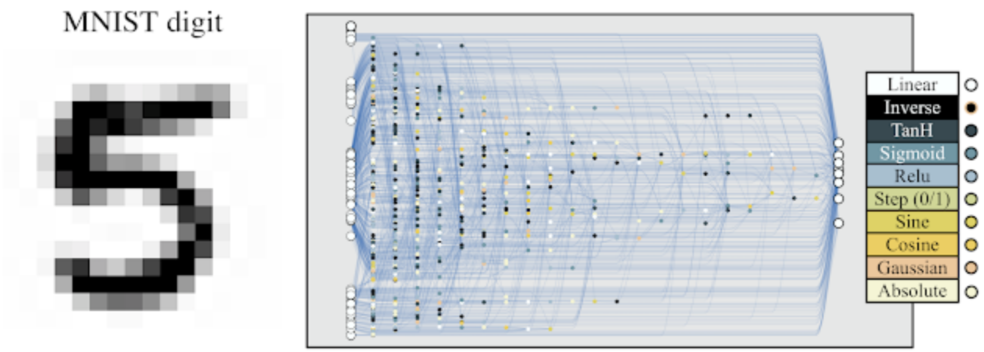
Car Pole Balancing Control Problems

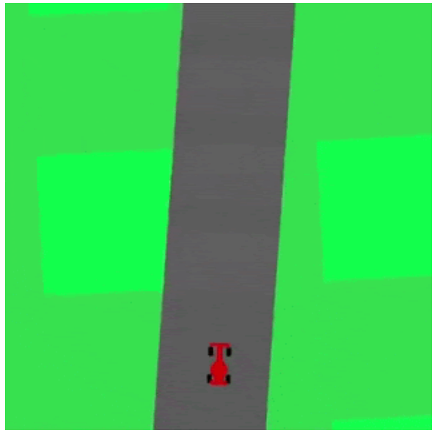
K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," in *Evolutionary Computation*, vol. 10, no. 2, pp. 99-127, June 2002

WANN: NEAT-based Networks Applied in Real Problems without Parameters' Help









Randomly Initialized CNN: ~10% accuracy
WANN + Random weights: >80%
WANN + Shared weights: > 90%

- WANNs can perform its task using range of shared weight parameters
- But the performance is still not comparable to a network that learns weights for each individual connection

A. Gaier and D. Ha, "Weight Agnostic Neural Networks," NeurIPS 2019

Architecture, architecture, architecture…

- Learned "important" weights of the large model are not useful for the small pruned model
- The pruned architecture itself, rather than a set of inherited "important" weights, is more crucial to the efficiency in the final model, which suggests that in some cases pruning can be useful as an architecture search paradigm

-- Z. Liu et. al., Rethinking the Value of Network Pruning, ICLR 2019

- As randomly weighted neural networks with fixed weights grow wider and deeper, an "untrained subnetwork" approaches a network with learned weights in accuracy.

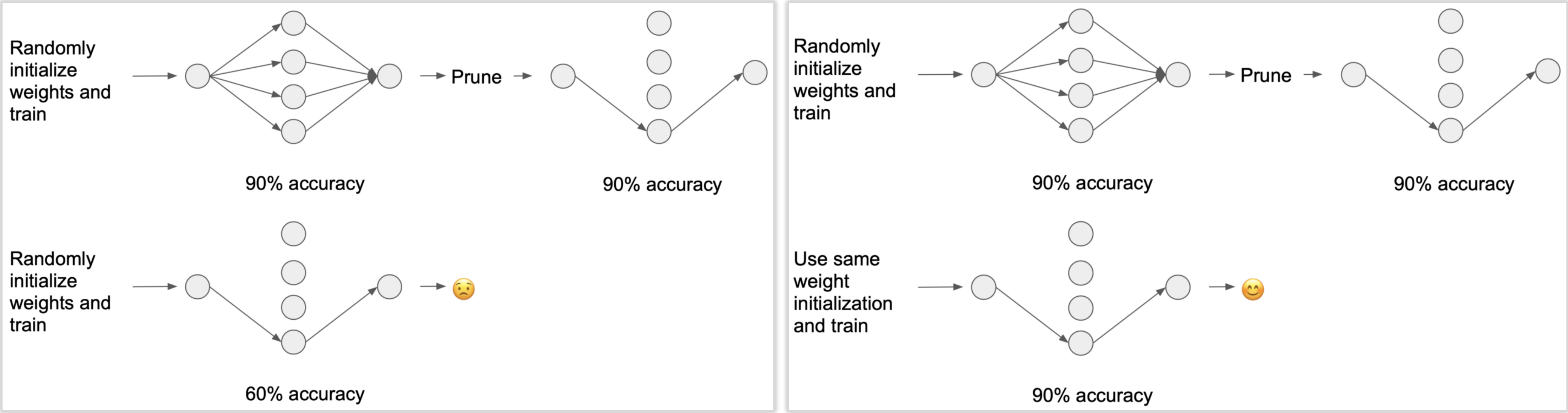
-- V. Ramanujan et. al., What's Hidden in a Randomly Weighted Neural Network?, CVPR 2020

- Networks with randomly generated architectures yield networks with competitive accuracy on ImageNet, the best ones outperform or are comparable to their fully manually designed counterparts and the networks found by various neural architecture search methods

-- S. Xie et. al., Exploring Randomly Wired Neural Networks for Image Recognition, CVPR 2020

But parameters are very very important

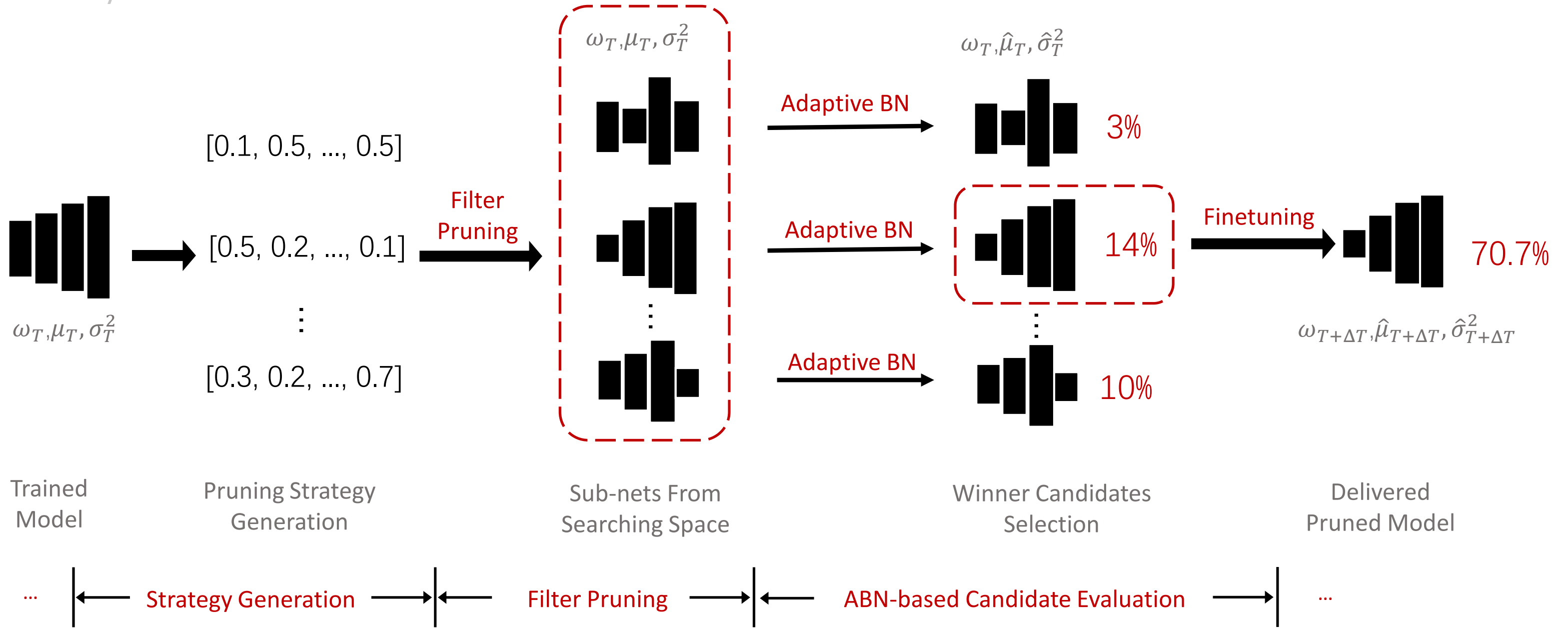
Lottery Ticket Hypothesis: the same architecture + bad initialization weights = NO!



"A randomly-initialized, dense neural network contains a subnetwork that is initialized such that — when trained in isolation — it can match the test accuracy of the original network after training for at most the same number of iterations."

Jonathan Frankle and Michael Carbin
The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks, ICLR 2019

EagleEye: the same architecture + bad initialization weights = NO!



Winning architectures can be very different and unpredictable

Model-FLOPs	Fine-tuning	Train-from-Scratch
MobileNetV1-284M	70.9%	68.7%
ResNet50-3G	77.1%	75.6%
ResNet50-2G	76.4%	74.4%
ResNet50-1G	74.2%	71.7%

Conclusions:
- Prune a trained large model > Train a pruned model
- Fine-tuning > from scratch:
  - Faster(100 epochs VS 180epochs)
  - Better accuracy (left table)
  - Inherit weights from pre-trained on large dataset

Bailin Li, Bowen Wu, Jiang Su, Guangrun Wang, Liang Lin, “EagleEye: Fast Sub-net Evaluation for Efficient Neural Network Pruning”, ECCV 2020

Parameters may work together with architecture to guarantee model accuracy

Parameter Precisions

- Binary / INT2 / INT4 / INT8/ INT16 / FP32

Datasets

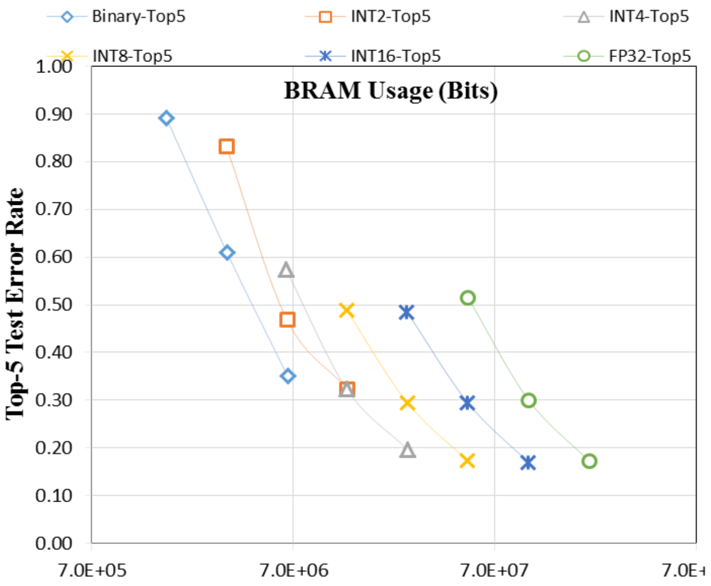
- MNIST / CIFAR-10 / ImageNet

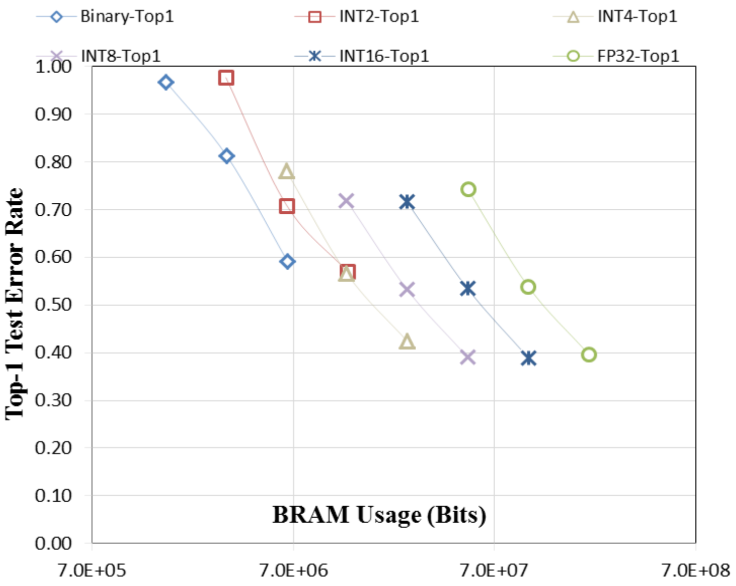
NN models

- FC: 784/4096x3/10
- CNNs: VGGNet (15 CONVs+3FC) and DarkNet (8 CONVs)
- NN arch. scaling factors: 0.03125, 0.0625, 0.125, 0.25, 0.5, 1

Metrics

- BRAM (Bits) is memory footprint on hardware that reflects amount of NN parameters





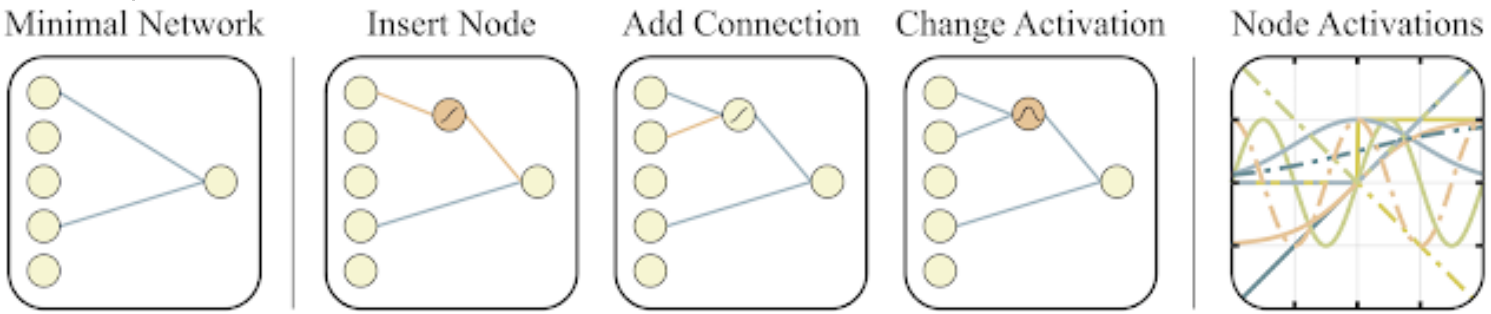
Conclusions:

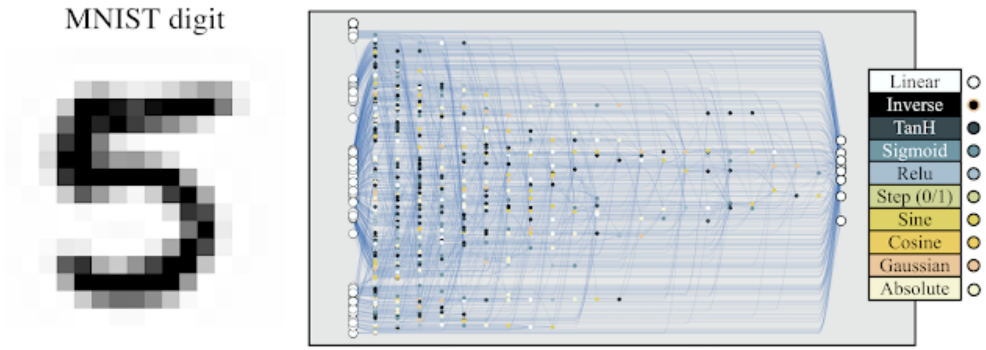
- Model with 2-bit parameters requires ~2X larger architectures than high-precision models to achieve the same accuracy
- INT4 and INT8 are more hardware-efficient than INT2 or Binary networks on ImageNet Tasks

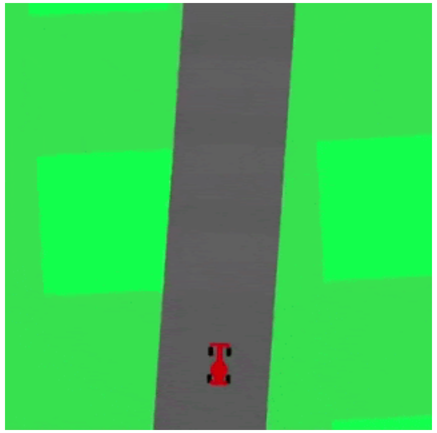
J. Su, N. Fraser, G. Gambardella, M. Blott, G. Durelli, D. B. Thomas, P. Leong and P. Y. K. Cheung, ``Trade-offs Between Accuracy and Throughput for Reduced Precision NNs on Reconfigurable Logic'', Int. Symp. on Applied Reconfig. Comput. , 2018.

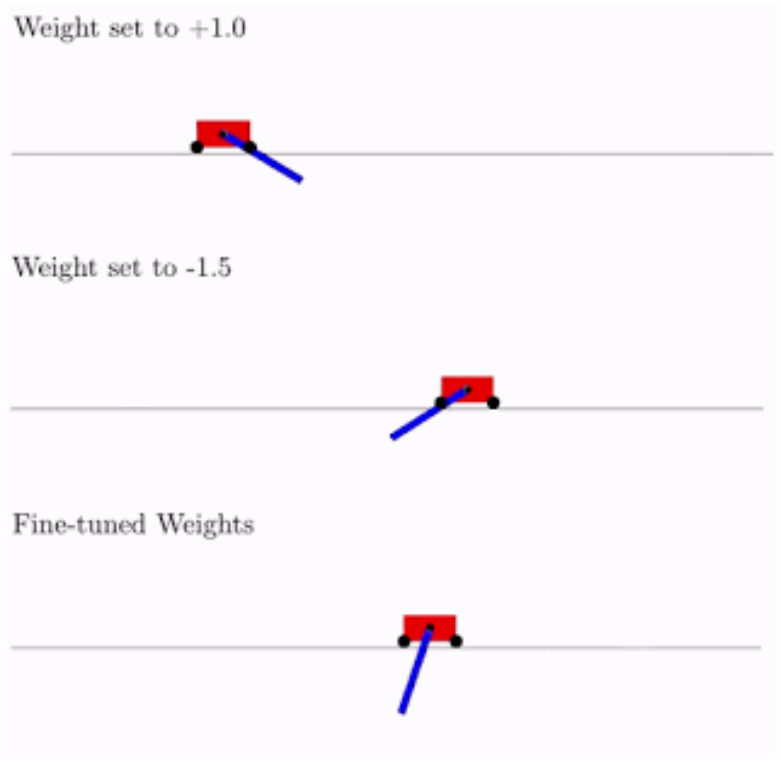
Architecture and parameters are somehow correlated

WANN: NEAT-based Networks Applied in Real Problems without Parameters' Help









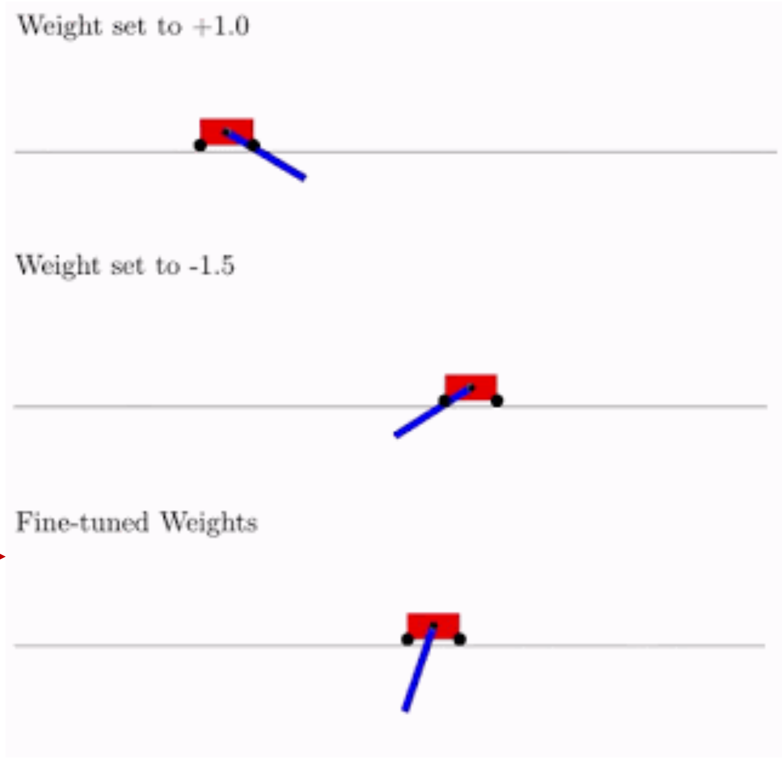
Randomly Initialized CNN: ~10% accuracy
WANN + Random weights: >80%
WANN + Shared weights: > 90%

- WANNs can perform its task using range of shared weight parameters
- But the performance is still not comparable to a network that learns weights for each individual connection

A. Gaier and D. Ha, "Weight Agnostic Neural Networks," NeurIPS 2019

WANN: NAS acts as genome while Parameters act as individual growth

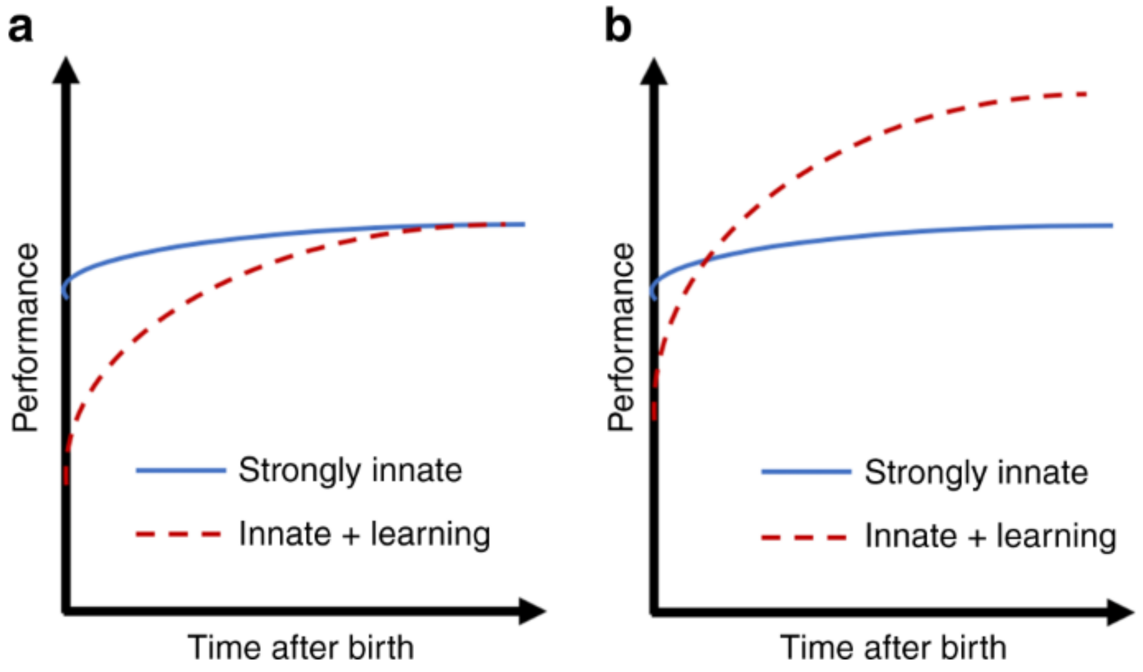
- WANNs can perform its task using range of shared weight parameters
- But the performance is still not comparable to a network that learns weights for each individual connection



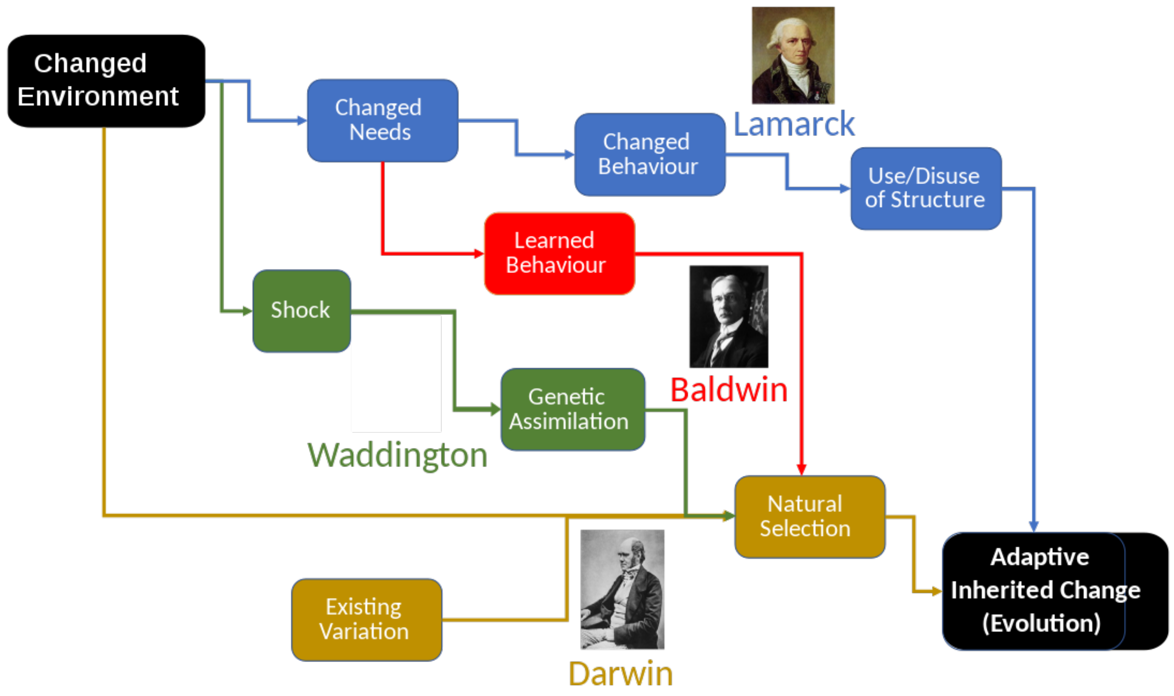
To further improve its performance, we can use the WANN architecture, and the best shared weight as a *starting point* to fine-tune the weights of each individual connection using a learning algorithm

A. Gaier and D. Ha, "Weight Agnostic Neural Networks," NeurIPS 2019

Inspirations from the nature



Baldwin Effect



a Learning makes two different species the same level of fitness

b A species using the mixed strategy may thrive if the environment dramatically changes

Zador, A.M. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nat Commun* **10**, 3770 (2019)

There might be a unified formulation across architecture (α) an d Parameters(β) to describe the black-box of DNNs

$$y = f(x \mid \alpha, \beta)$$

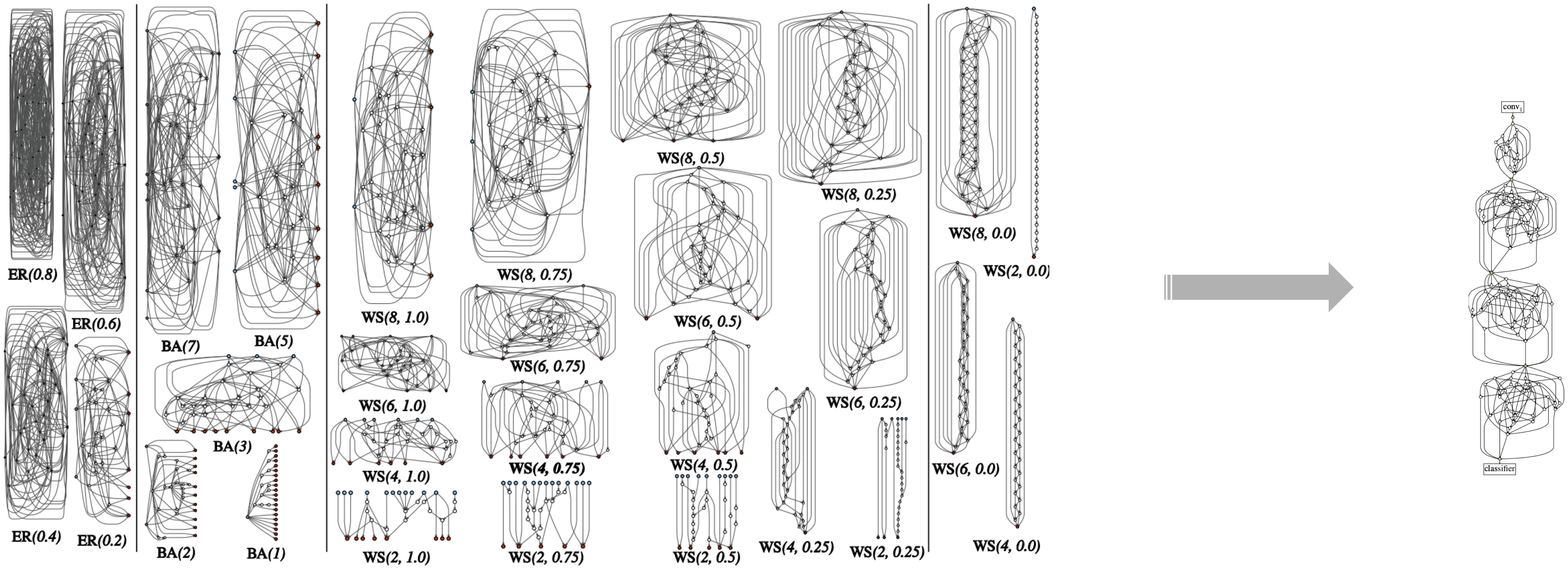
Current NAS:

- Searching encoding of monotonic connections or searching pre-defined super ne twork in a brute-force way (unpredictable).
- More efficient way of evolution needs to be found for complex primitive operators
- Applicability (enormous searching efforts and hardware-friendly issues)

Current pruning methods:

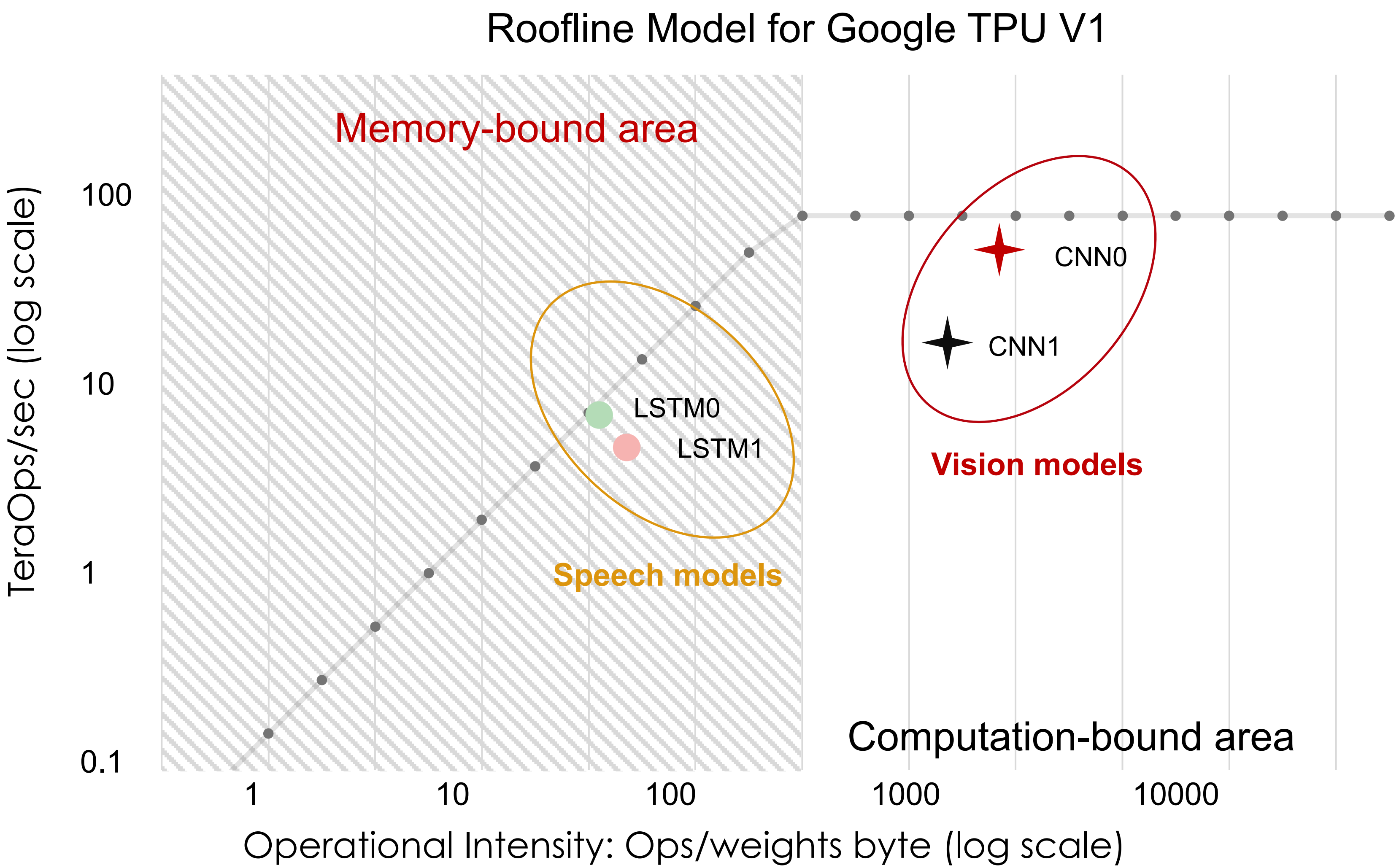
- Do not ignore the power of genome

Deployable NAS: A disaster to computation in both searching and deployment



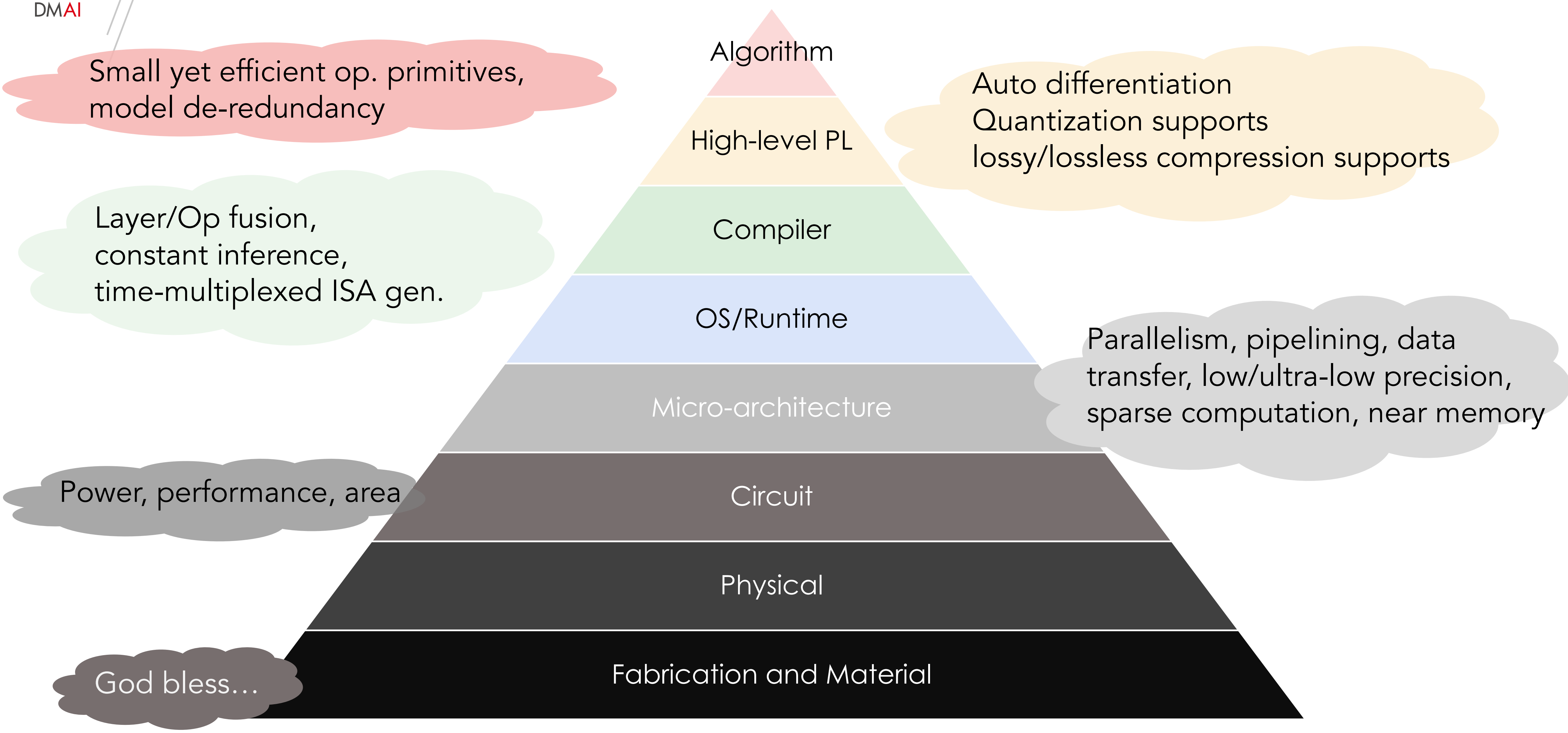
Saining Xie, Alexander Kirillov, Ross Girshick, Kaiming He, "Exploring Randomly Wired Neural Networks for Image Recognition," CVPR 2019

Deployable NAS: Different building blocks = different hardware challenges

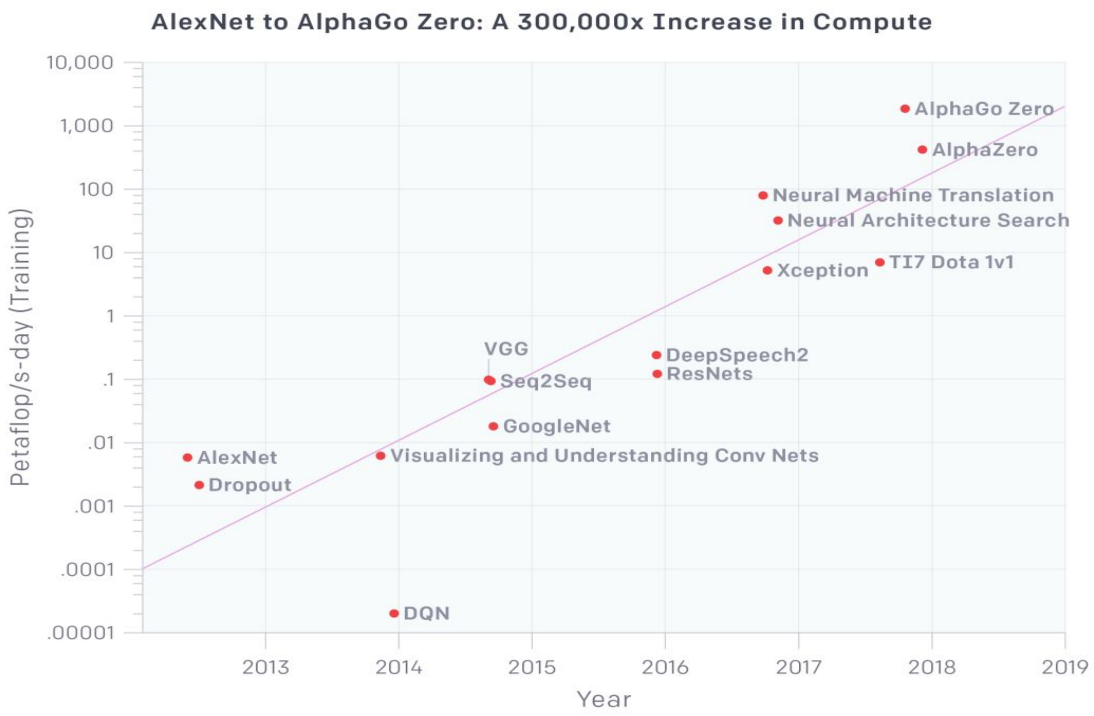


Data source: N. Jouppi *et al.*, "In-Datacenter Performance Analysis of a Tensor Processing Unit", *ACM SIGARCH Computer Architecture News 45(2): 1-12,* June 2017

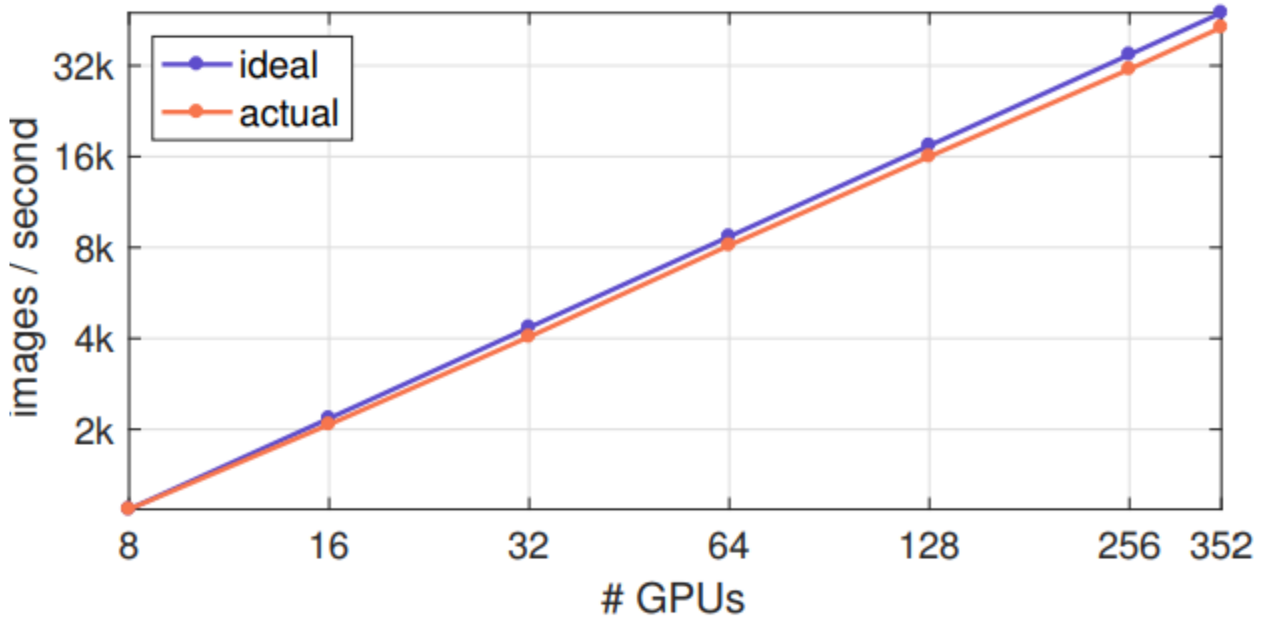
Deployable NAS: Arch.+Param. As in the Software-To-Hardware Full Stack



Deployable NAS/pruning: a way to slow down HPC scalability?



OpenAI, https://www.jiqizhixin.com/articles/051704



P. Goyal *et. al*., "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour", Facebook AI

Can NAS cool down people from the enthusiasm on computational power？

Summary

- Architecture or parameters can be both important and somehow correlated
- Pruning can be a way to conduct deployment-oriented NAS
- Biological analogy: architecture as genome while parameters as individual diff.
- Deployable NAS: a full-stack optimization problem

Thank you

Jiang Su
sujiang@dm-ai.cn
2020-08-29