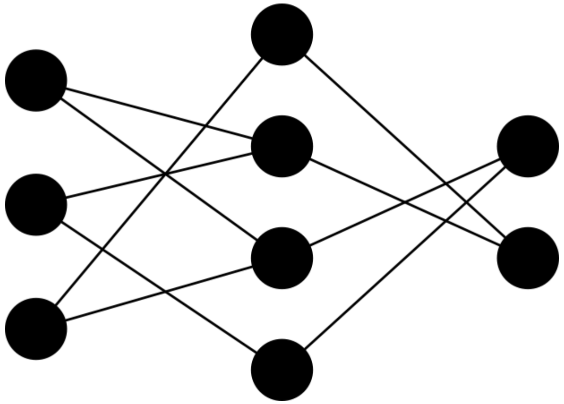


Exploring Complexity Reduction to address AI's Carbon Footprint

Sourya Dey

Research Engineer, Galois

(Some work done as a PhD student at University of Southern California)

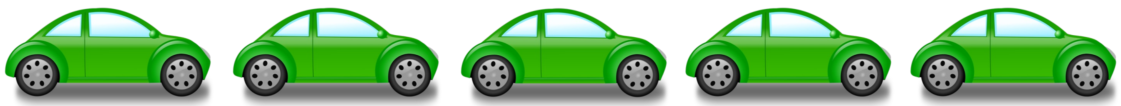
February 9th, 2022
Galois Quotidian Machine Ethics Workshop

Overview

- Machine learning models largely prioritize end-user performance, often neglecting the energy and costs required to make them good.
- Training neural nets in particular requires thousands of GPU hours.

Consumption	**CO_2e (lbs)**
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Strubell et al 2019 – "Energy and Policy Considerations for Deep Learning in NLP"



The Ethical Questions

“it is well documented in the literature on *environmental racism* that the negative effects of climate change are reaching and impacting the world’s most *marginalized communities* first. Is it fair or just to ask, for example, that the residents of the Maldives (likely to be underwater by 2100) or the 800,000 people in Sudan affected by drastic floods pay the environmental price of *training and deploying ever larger English [language models]*, when similar large-scale models aren’t being produced for Dhivehi or Sudanese Arabic?”

Bender, Gebru et al 2021 – “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?”

(events associated with this paper forced Gebru out of Google)

“[this situation] *stifles creativity*. Researchers with good ideas but without access to large-scale compute will simply not be able to execute their ideas”

“[this situation] prohibits certain types of research on the basis of access to financial resources. This even more deeply promotes the already problematic *‘rich get richer’* cycle of research funding”

Strubell et al 2019 – “Energy and Policy Considerations for Deep Learning in NLP”

Suggested solutions in literature

- Quantifying and reporting CO_2 equivalent emissions, energy, and compute efficiency associated with ML models.

ML Emissions Calculator built by *Lacoste et al 2019 – "Quantifying the Carbon Emissions of Machine Learning"*

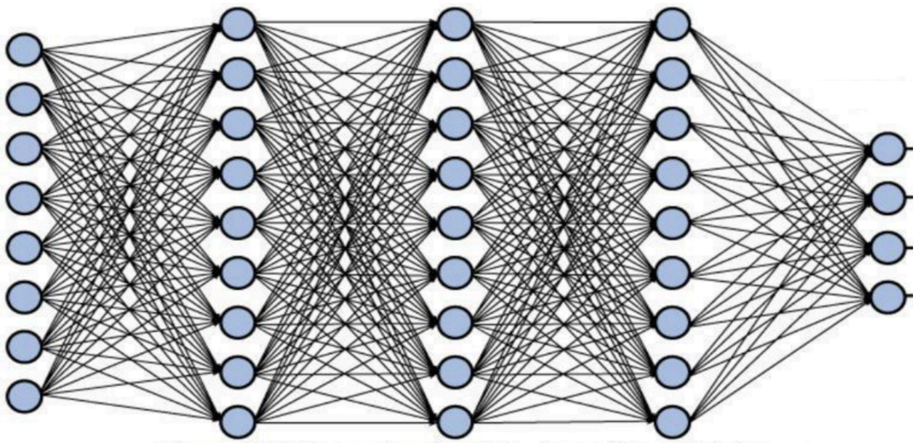
- Choosing more efficient hardware (TPUs > GPUs > CPUs).

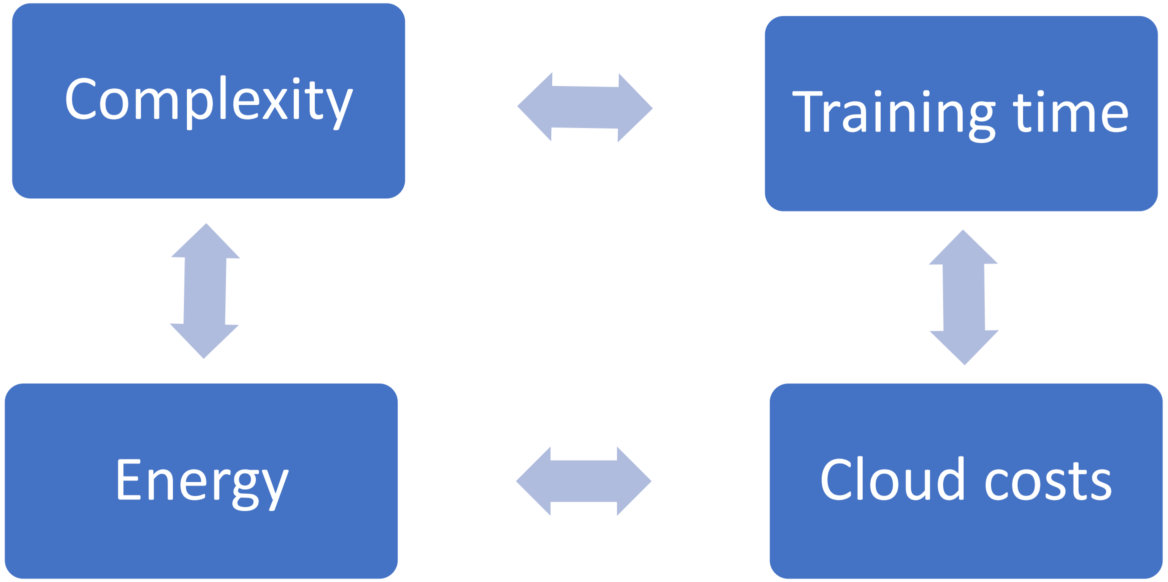
Energy comparison of GPUs vs CPUs done by *Li et al 2016 – "Evaluating the Energy Efficiency of Deep CNNs on CPUs and GPUs"*

- Choosing locations of cloud providers and data centers wisely.

The Complexity Conundrum…

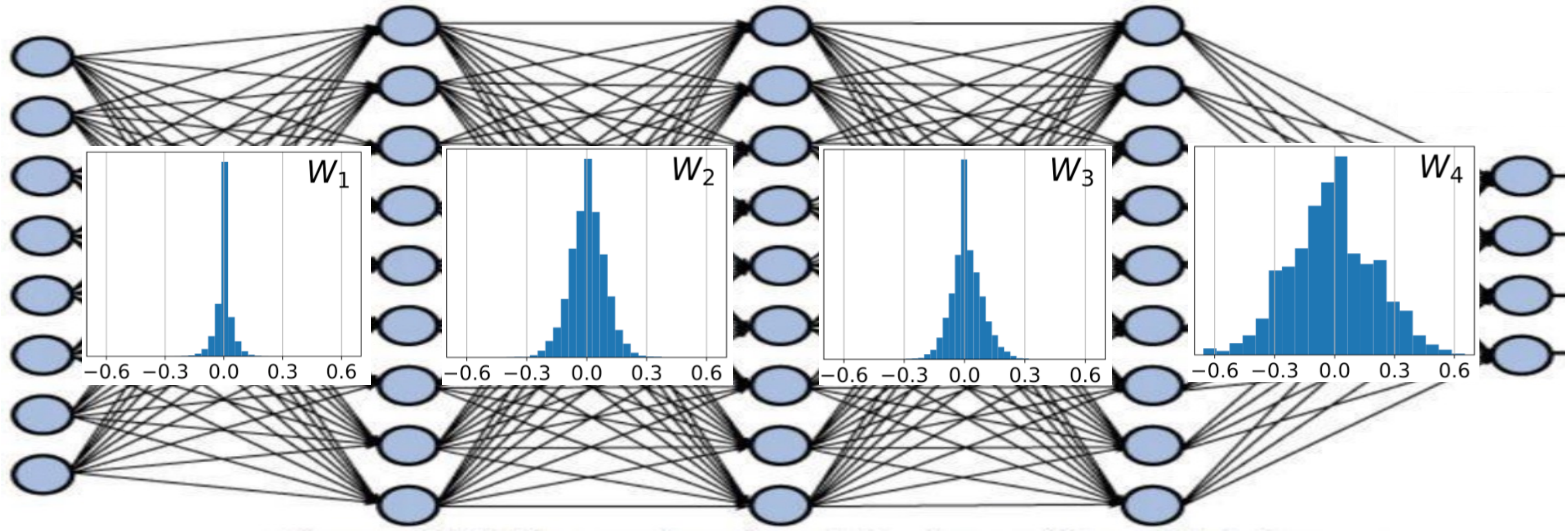
Modern neural networks suffer from parameter explosion





He et al 2016 – "Deep Residual Learning for Image Recognition"

Pre-defined sparsity – Motivation



In a fully connected neural network, most weights are small in magnitude after training

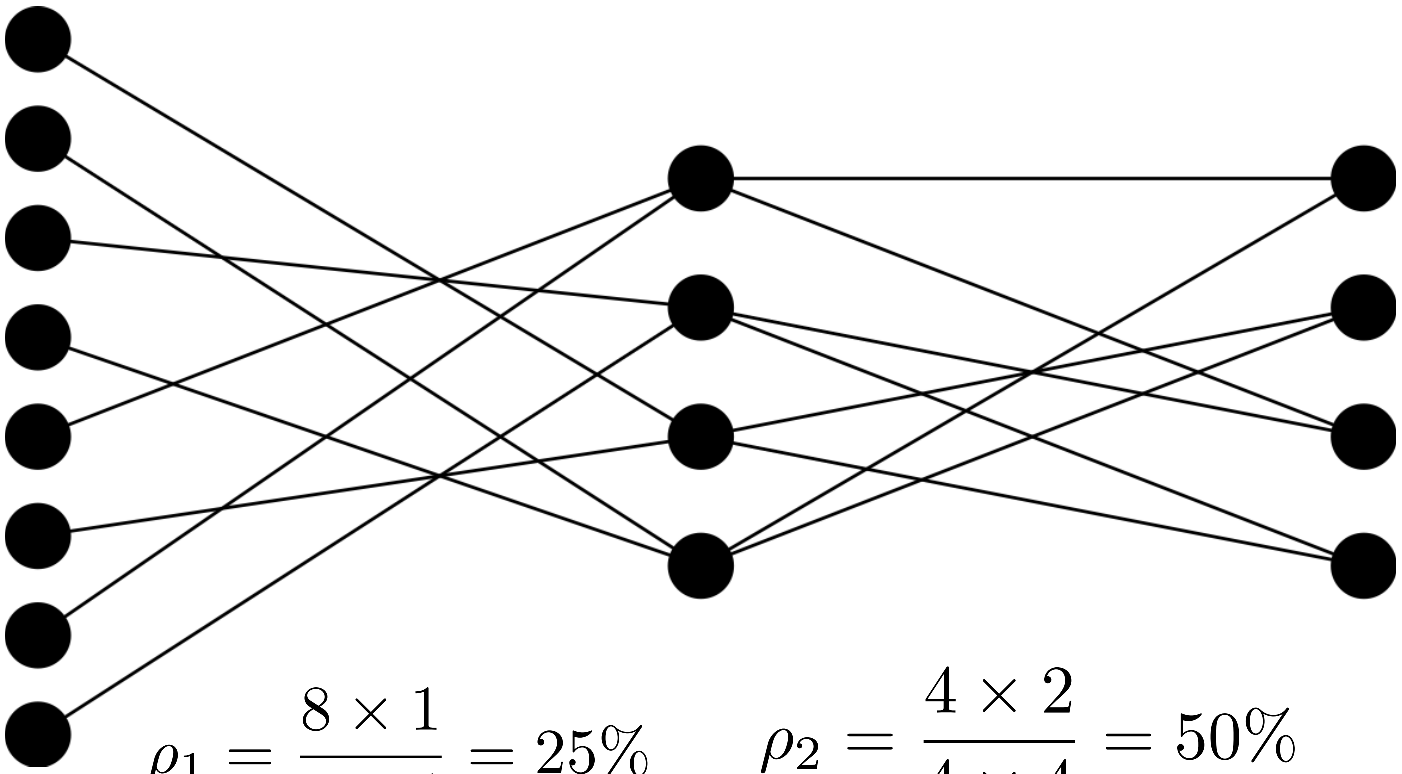
Dey et al 2019 – "Pre-Defined Sparse Neural Networks with Hardware Acceleration"

Pre-defined Sparsity

Pre-define a sparse connection pattern **prior to training**

Use this sparse network for both training and inference

Reduced training *and* inference complexity



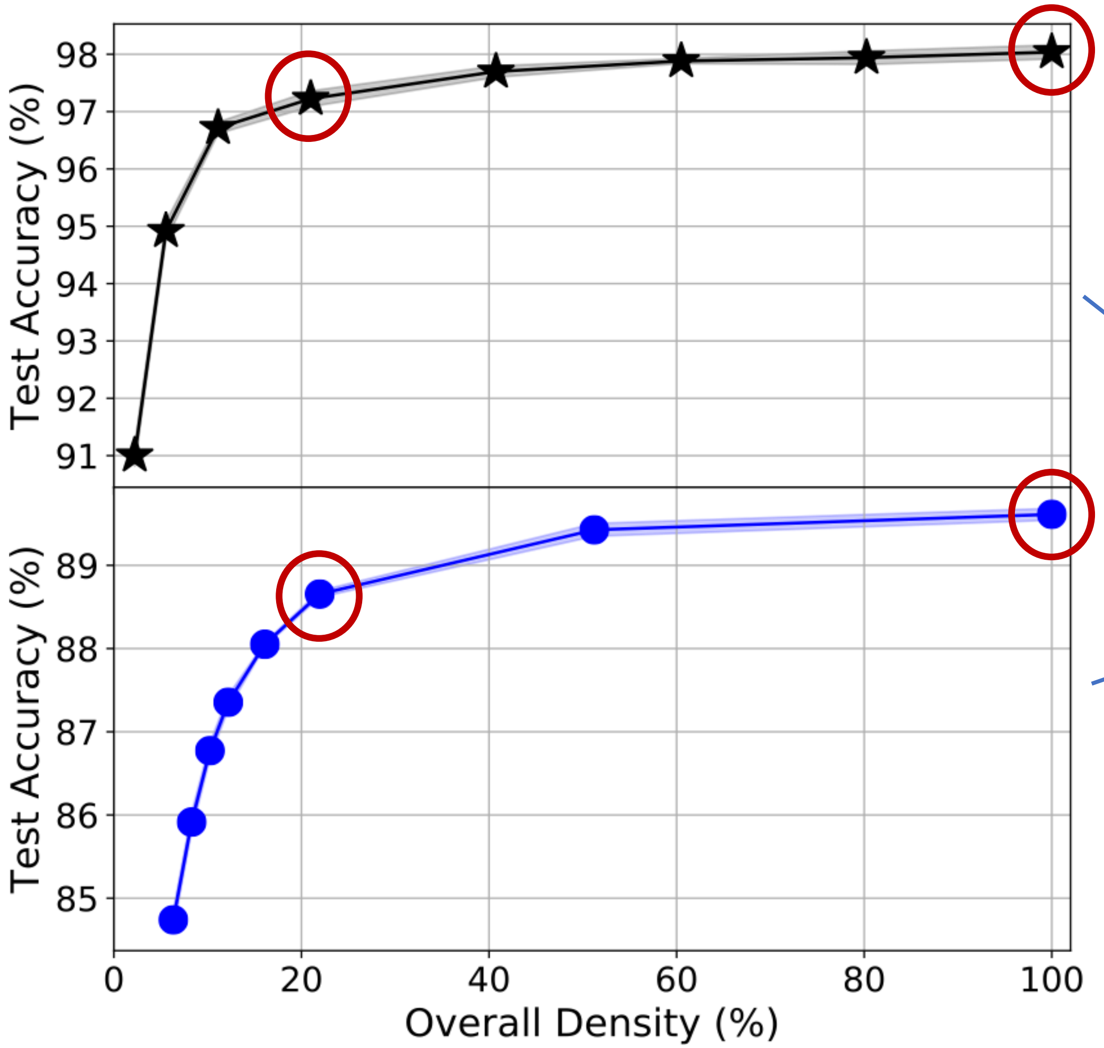
$$\rho_1 = \frac{8 \times 1}{8 \times 4} = 25\%$$

$$\rho_2 = \frac{4 \times 2}{4 \times 4} = 50\%$$

$$\rho_{\mathrm{net}} = \frac{8 + 8}{32 + 16} = 33\%$$

Overall Density compared to fully connected

Pre-defined sparsity – Performance Snapshot



Starting with only 20% of parameters reduces test accuracy by just 1%

2D images: MNIST handwritten digits

Text: Reuters news articles

Speech: TIMIT phonemes

3D images: CIFAR images

1D data: Morse symbols

Deep-n-Cheap

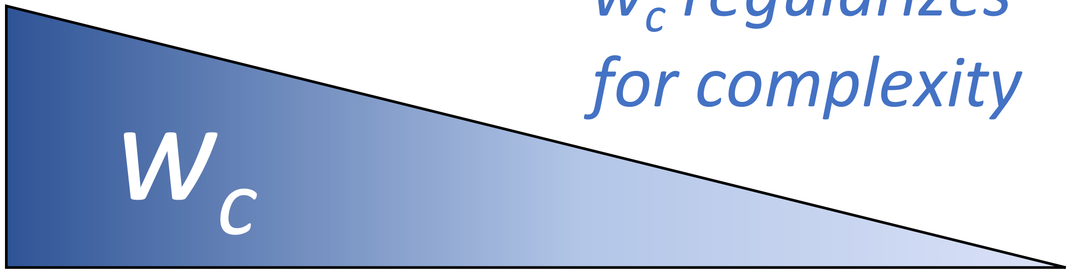
DnC Deep-n-Cheap

Low Complexity AutoML framework

Optimize performance and complexity

Modified loss function: *Original Loss + w_c*Complexity*

w_c regularizes for complexity



Quick to train
Energy-efficient
Sacrifice performance

Good performance
Slow to train
Slow search process

Dey et al 2020 – "Deep-n-Cheap: An Automated Efficient and Extensible Search Framework for Cost-Effective Deep Learning"
https://github.com/souryadey/deep-n-cheap

Deep-n-Cheap – Performance comparison

Table shows CNNs on CIFAR-10 (MLP trends are similar)

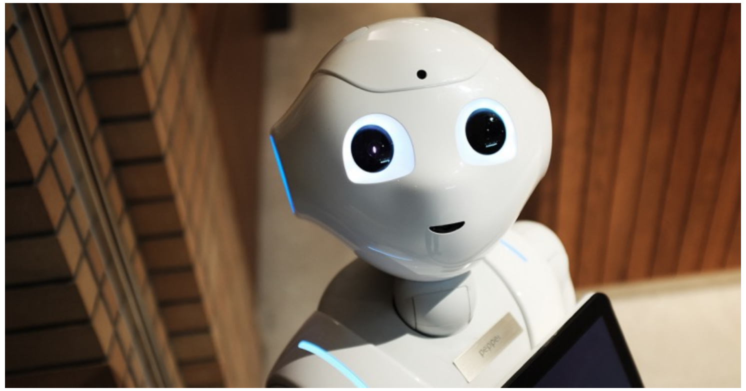
Framework	Additional settings	Search cost (GPU hrs)	Best model found from search			
			Architecture	t_{tr} (sec)	Batch size	Best val acc (%)
Proxyless NAS	Proxyless-G	96	537 conv layers	429	64	93.22
Auto-Keras	Default run	14.33	Resnet-20 v2	33	32	74.89
AutoGluon	Default run	**3**	Resnet-20 v1	37	64	88.6
	Extended run	101	Resnet-56 v1	46	64	91.22
Auto-Pytorch	'tiny cs'	6.17	30 conv layers	39	64	87.81
	'full cs'	6.13	41 conv layers	31	106	86.37
Deep-n-Cheap	$w_c = 0$	29.17	14 conv layers	10	120	**93.74**
	$w_c = 0.1$	19.23	8 conv layers	4	459	91.89
	$w_c = 10$	16.23	4 conv layers	**3**	256	83.82

The non-requirement of deep models is also echoed in the popular Zagoruyko and Komodakis 2016 – *"Wide Residual Networks"*

Takeaways

We may not really need very big / deep ML models.

Efficient models are more desirable
than purely high-performing models.



Thank you !!

Questions ??