Advanced Deep Learning: Properties

Anders Søgaard

coastal

Course outline

Goal 1: Quick tour of recent developments in deep learning

Goal 2: Inspiration for thesis/research projects

Week	Lecturer	Subject	Literature	Assignment
1	Stefan	Introduction to Neural Networks.	d2l 2.1-2.5, 2.7, 11.5.1, **slides**	
2	Stefan	CNNs; FCNs; U-Nets. Data augmentation; invariance; regularization e.g. dropout	d2l 6, 7, 13.9-13.11, **slides**	Assignment 1 (May 10)
3	Anders/Phillip	May 9 (A): RNNs May 11 (P): Transformers	d2l 8 Transformers: d2l 10.5-10.7 + Vaswani et al. (2017)	Assignment 2 (May 20)
4	Phillip/Anders	May 16 (P): Representation and Adversarial Learning May 18 (A): A Learning Framework + Self-supervised Learning + Contrastive Learning	Autoencoders: blog post GANs: Goodfellow (2016) Self-supervised learning: blog post Contrastive learning: Dor et al. (2018) Adversarial examples: Goodfellow et al. (2015)	
5	Anders	May 23: General Properties, e.g., Scaling Laws, Lottery Tickets, Bottleneck Phenomena May 25: Applications of Representation, Adversarial and Contrastive Learning	GANs: Lample et al. (2018) Autoencoders: Chandar et al. (2011) Contrastive learning: Yu et al. (2018) DynaBench: Talk by Douwe Kiela (Facebook, now HuggingFace) Scaling laws: Kaplan et al. (2020)	Assignment 3 *[MC on Representation Learning/1p Report on Lottery Ticket extraction]* (June 3)
6	Anders	May 30: Interpretability, Transparency, and Trustworthiness & Deep Learning for Scientific Discovery June 1: Interpretability (Feature Attribution), including Guest Lecture by Stephanie Brandl	DL for Scientific Discovery: Sullivan (2022) Interpretability/Background: Søgaard (2022)	
7	Anders	June 6: *Off (no teaching)* June 8: Interpretability (Training Data Influence)	Literature: Feng and Boyd-Graber (2018) ; Jiang and Senge (2021)	
8	Anders	June 13-15: Best Practices	Literature: Dodge et al. (2019) and Raji et al. (2021)	Assignment 4 *[MC on Interpretability; 1p Report on Best Practices]* (June 21)

Course outline

Goal 1: Quick tour of recent developments in deep learning

Goal 2: Inspiration for thesis/research projects

Architectures

Framework

Fairness /

Explainable AI

Methodology

Week	Lecturer	Subject	Literature	Assignment
1	Stefan	Introduction to Neural Networks.	d2l 2.1-2.5, 2.7, 11.5.1, **slides**	
2	Stefan	CNNs; FCNs; U-Nets. Data augmentation; invariance; regularization e.g. dropout	d2l 6, 7, 13.9-13.11, **slides**	Assignment 1 (May 10)
3	Anders/Phillip	May 9 (A): RNNs May 11 (P): Transformers	d2l 8 Transformers: d2l 10.5-10.7 + Vaswani et al. (2017)	Assignment 2 (May 20)
		May 16 (P): Representation and Adversarial Learning	Autoencoders: blog post GANs: Goodfellow (2016) Self-supervised learning: blog post Contrastive learning: Dor et al. (2018) Adversarial examples: Goodfellow et al. (2015)	
4	Phillip/Anders	May 18 (A): A Learning Framework + Self-supervised Learning + Contrastive Learning		
5	Anders	May 23: General Properties, e.g., Scaling Laws, Lottery Tickets, Bottleneck Phenomena May 25: Applications of Representation, Adversarial and Contrastive Learning	GANs: Lample et al. (2018) Autoencoders: Chandar et al. (2011) Contrastive learning: Yu et al. (2018) DynaBench: Talk by Douwe Kiela (Facebook, now HuggingFace) Scaling laws: Kaplan et al. (2020)	Assignment 3 *[MC on Representation Learning/1p Report on Lottery Ticket extraction]* (June 3)
6	Anders	May 30: Interpretability, Transparency, and Trustworthiness & Deep Learning for Scientific Discovery June 1: Interpretability (Feature Attribution), including Guest Lecture by Stephanie Brandl	DL for Scientific Discovery: Sullivan (2022) Interpretability/Background: Søgaard (2022)	
7	Anders	June 6: *Off (no teaching)* June 8: Interpretability (Training Data Influence)	Literature: Feng and Boyd-Graber (2018); Jiang and Senge (2021)	
8	Anders	June 13-15: Best Practices	Literature: Dodge et al. (2019) and Raji et al. (2021)	Assignment 4 *[MC on Interpretability; 1p Report on Best Practices]* (June 21)

Today

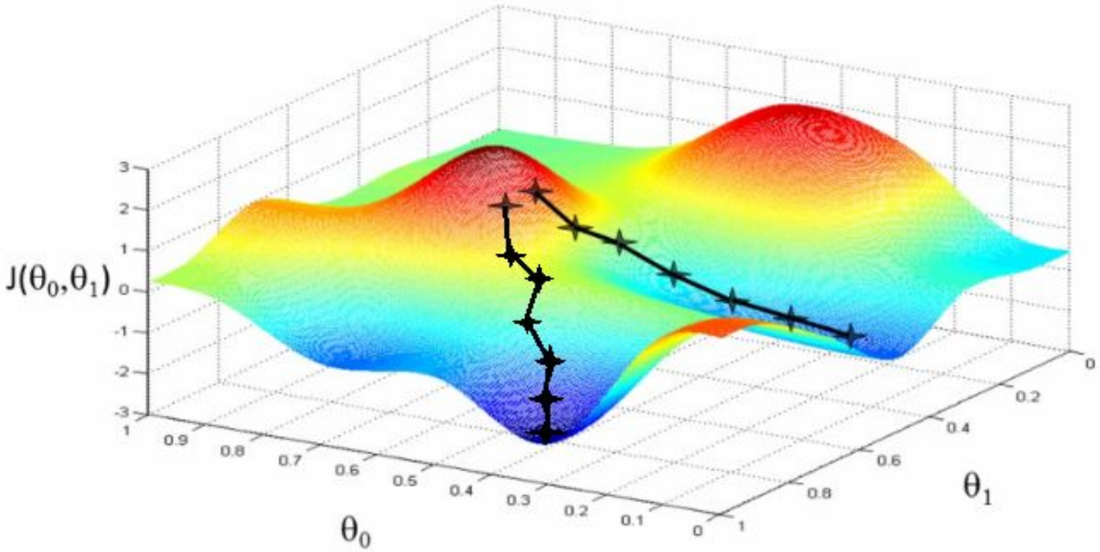
a) Scaling laws
b) Lottery tickets
c) Bottleneck theory
d) Over-parameterization

Background

—

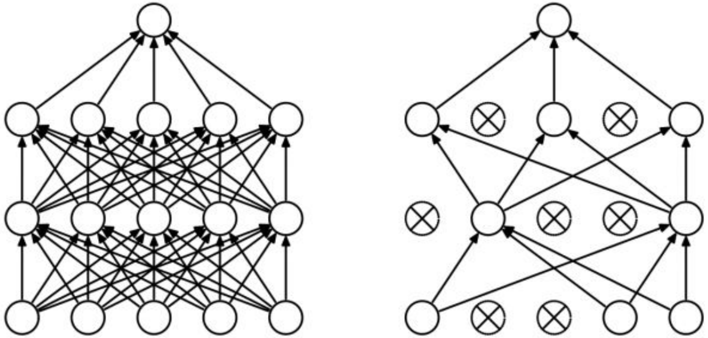
Loss Landscapes

Complex manifolds, riddled with local minima.



Regularization

E.g., drop-out, near-equivalent to L2 regularization. Note that there are **many** forms of regularization, e.g., from averaging, smoothing, multi-task learning.



(a) Standard Neural Net (b) After applying dropout.

Scaling laws

Model	Layers	Parameters	Hidden layer size	Training data	Objective
BERT-base	12	108m	768	16GB	MLM+NSP
BERT-large	24	324m	1024	16GB	MLM+NSP
ALBERT-base	12	12m	768	16GB	MLM+SRO
ALBERT-large	24	18m	1024	16GB	MLM+SRO
RoBERTa-large	24	324m	1024	160GB	MLM
GPT2	48	1542m	1600	40GB	Autoregressive
GPT3	96	170b	12288	570GB	Autoregressive

Other differences, e.g.: RoBERTa used a batch size of 8,000 with 300,000 steps. In comparison, BERT uses a batch size of 256 with 1 million steps

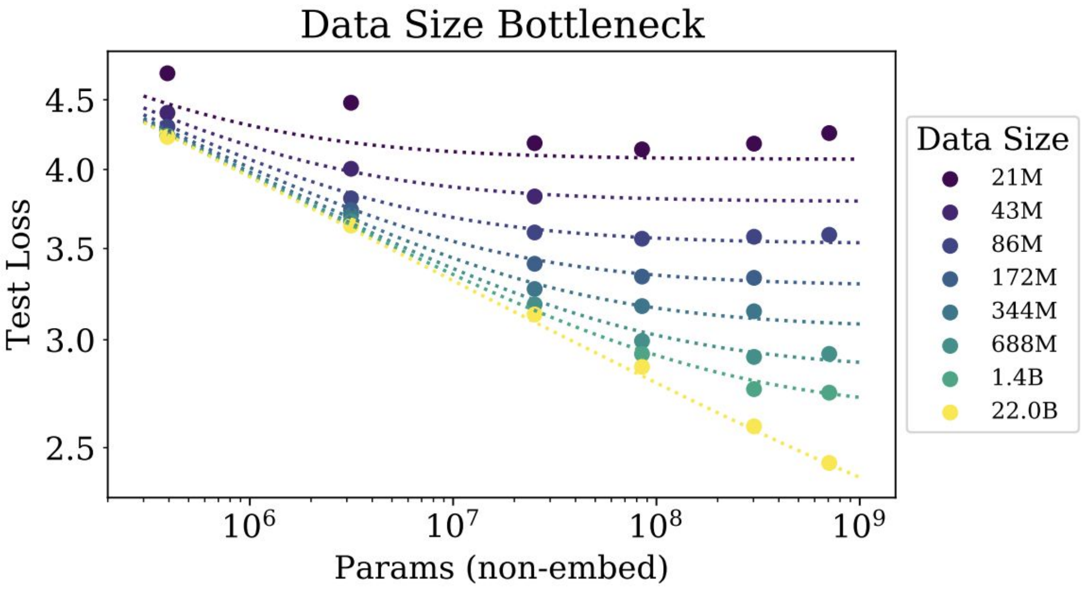
Model	Layers	Parameters	Hidden layer size	Training data	Objective
BERT-base	12	108m	768	16GB	MLM+NSP
BERT-large	24	324m	1024	16GB	MLM+NSP
ALBERT-large	24	18m	1024	16GB	MLM+SRO
RoBERTa-large	24	324m	1024	160GB	MLM
GPT2	48	1542m	1600	40GB	Autoregressive
GPT3	96	170b	12288	570GB	Autoregressive
Chinchilla	80	70b	8192	1.4TB	Autoregressive

Other differences, e.g.: RoBERTa used a batch size of 8,000 with 300,000 steps. In comparison, BERT uses a batch size of 256 with 1 million steps

Scaling laws for NLP

Kaplan et al. (2020) presented the first set of scaling laws, for NLP:

- Your model has to be very big to make use of large volumes of data.
- As model size (N) grows, you data should grow ~$N^{0.74}$.
- Given a 10× increase in budget, you need to increase N by 5.5× and D by 1.8×.

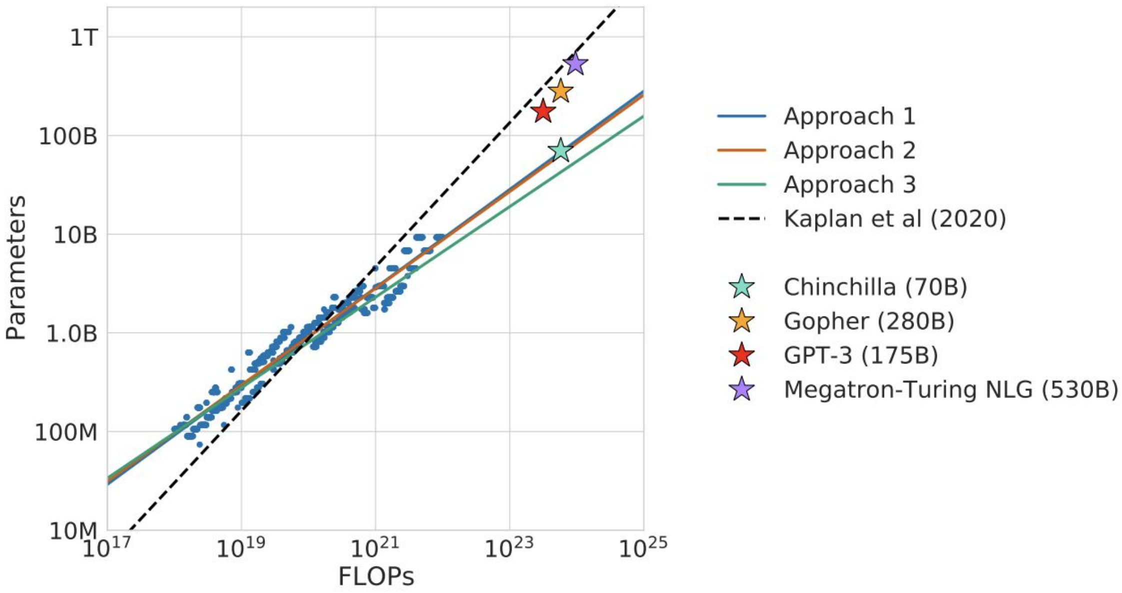


New scaling laws

Hoffmann et al. (2022) update these scaling laws.

- Given an increase in budget, you need to increase N and D by equal factors.

Note: Hoffmann et al. (2022) present three approaches to deriving scaling laws.



Good news

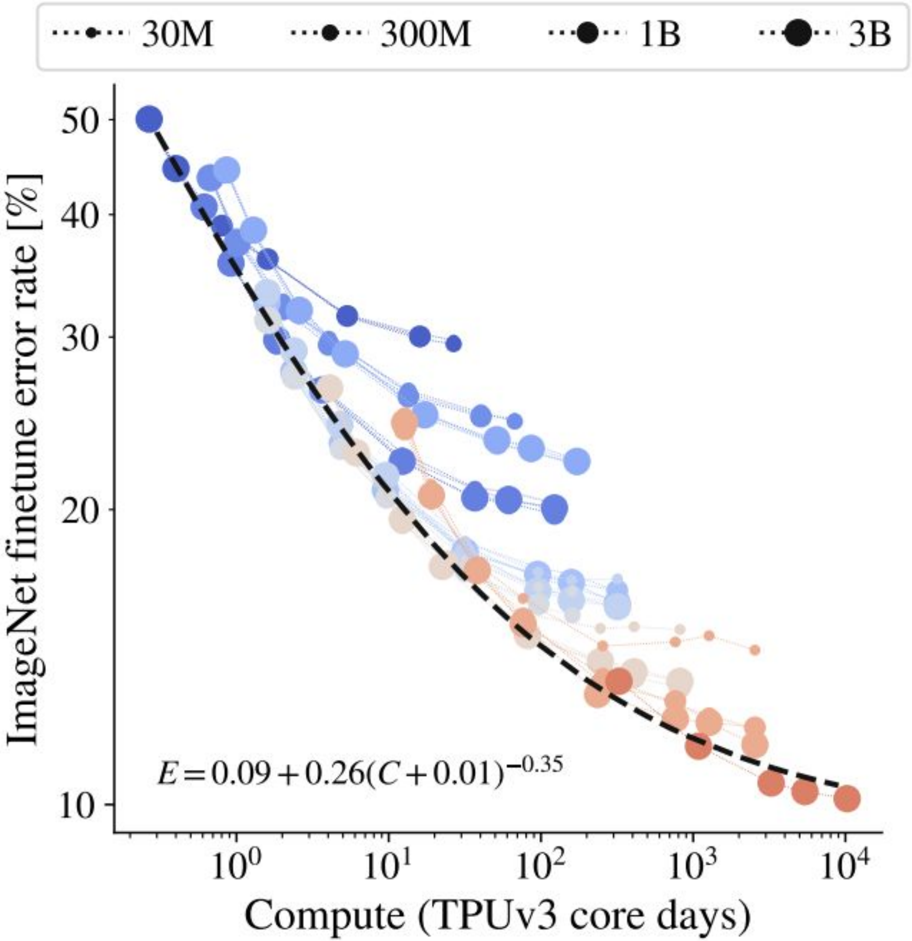
The general take-home message from Hoffmann et al. (2022) is that you don't need your models to be as big as expected (from Kaplan et al., 2020).

Note: It wouldn't make sense to train a 520B parameter model unless you had 60x the compute used for Chinchilla.

Parameters	FLOPs	FLOPs (in *Gopher* unit)	Tokens
400 Million	1.92e+19	1/29, 968	8.0 Billion
1 Billion	1.21e+20	1/4, 761	20.2 Billion
10 Billion	1.23e+22	1/46	205.1 Billion
67 Billion	5.76e+23	1	1.5 Trillion
175 Billion	3.85e+24	6.7	3.7 Trillion
280 Billion	9.90e+24	17.2	5.9 Trillion
520 Billion	3.43e+25	59.5	11.0 Trillion
1 Trillion	1.27e+26	221.3	21.2 Trillion
10 Trillion	1.30e+28	22515.9	216.2 Trillion

Scaling laws in CV

Zhai et al. (2021) present scaling laws for computer vision. **Note:** Small models (blue circles) trained on small data (small circles) fall off the frontier.



Take-home messages

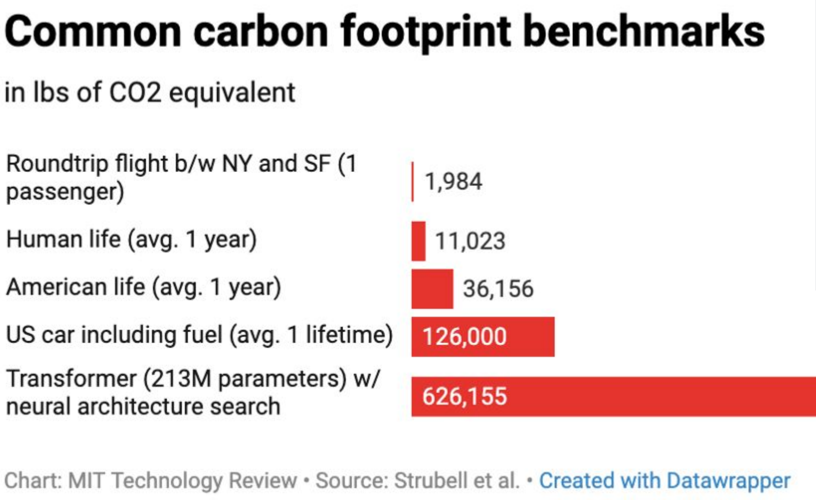
- Power laws can save the world a lot of energy.
- Big models only make sense if you have sufficient data.
- Training for long only makes sense if you have both big models and big data.

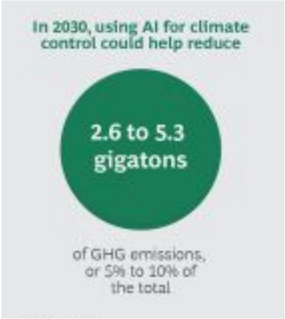
Carbon Emission and DL

DL is used at scale. Even a single company like Meta produces about 5,000 terabytes of data each day, and make trillions of predictions each day. Training even moderate-sized language models, e.g., MegatronLM, requires 3-4 times the energy an average household spends in a year.

Model	Hardware	Power (W)	Hours	kWh·PUE	CO_2e	Cloud compute cost
$\text{Transformer}_{base}$	P100x8	1415.78	12	27	26	\$41–\$140
Transformer_{big}	P100x8	1515.43	84	201	192	\$289–\$981
ELMo	P100x3	517.66	336	275	262	\$433–\$1472
BERT_{base}	V100x64	12,041.51	79	1507	1438	\$3751–\$12,571
BERT_{base}	TPUv2x16	—	96	—	—	\$2074–\$6912
NAS	P100x8	1515.43	274,120	656,347	626,155	\$942,973–\$3,201,722
NAS	TPUv2x1	—	32,623	—	—	\$44,055–\$146,848
GPT-2	TPUv3x32	—	168	—	—	\$12,902–\$43,008

Table 3: Estimated cost of training a model in terms of CO_2 emissions (lbs) and cloud compute cost (USD).[7] Power and carbon footprint are omitted for TPUs due to lack of public information on power draw for this hardware.

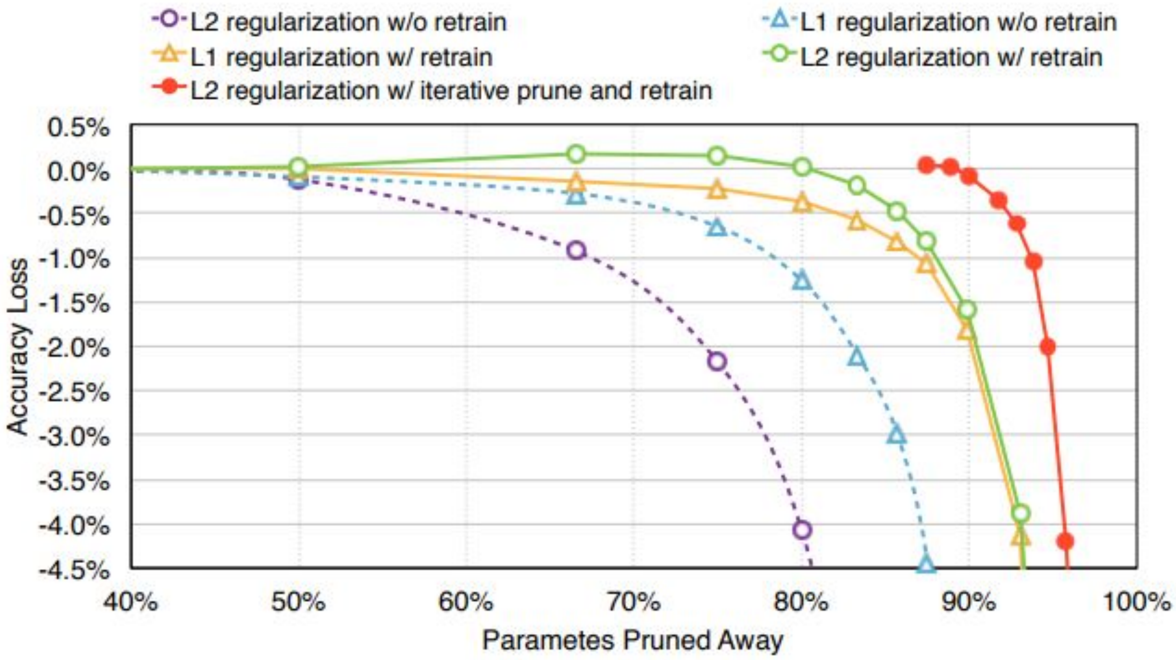




Lottery tickets

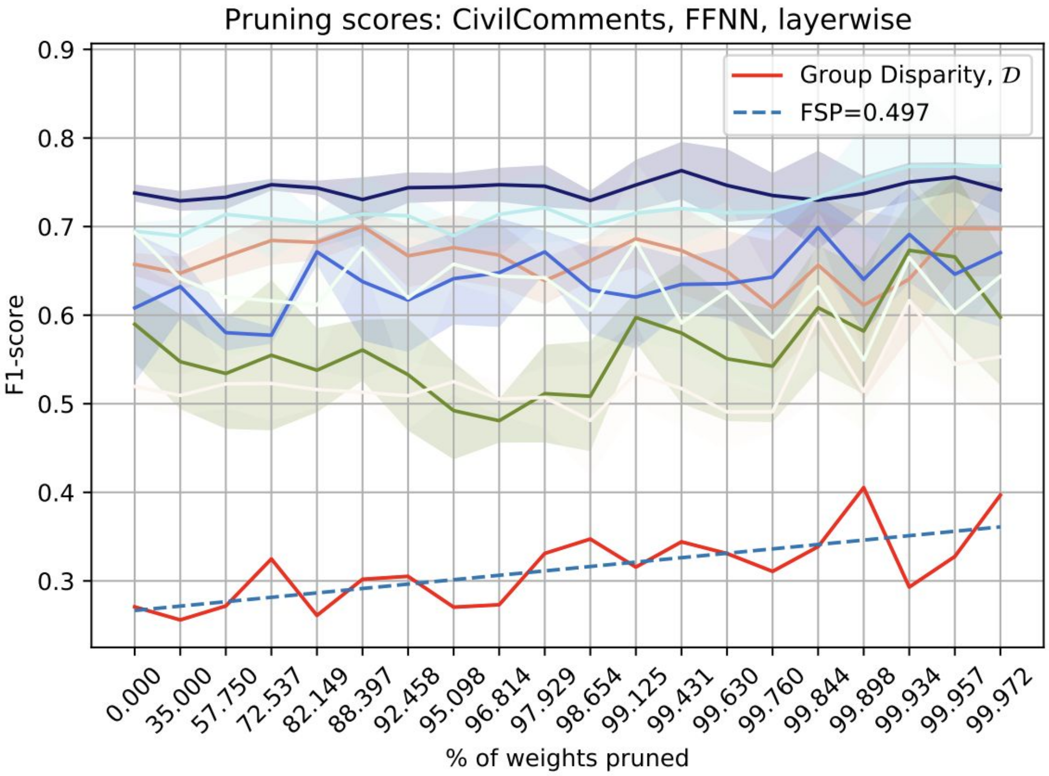
Lottery ticket hypothesis

Hypothesis: Dense, randomly-initialized, feed-forward networks contain subnetworks ("winning tickets") that - when trained in isolation - reach test accuracy comparable to the original network in a similar number of iterations.



Lottery ticket hypothesis

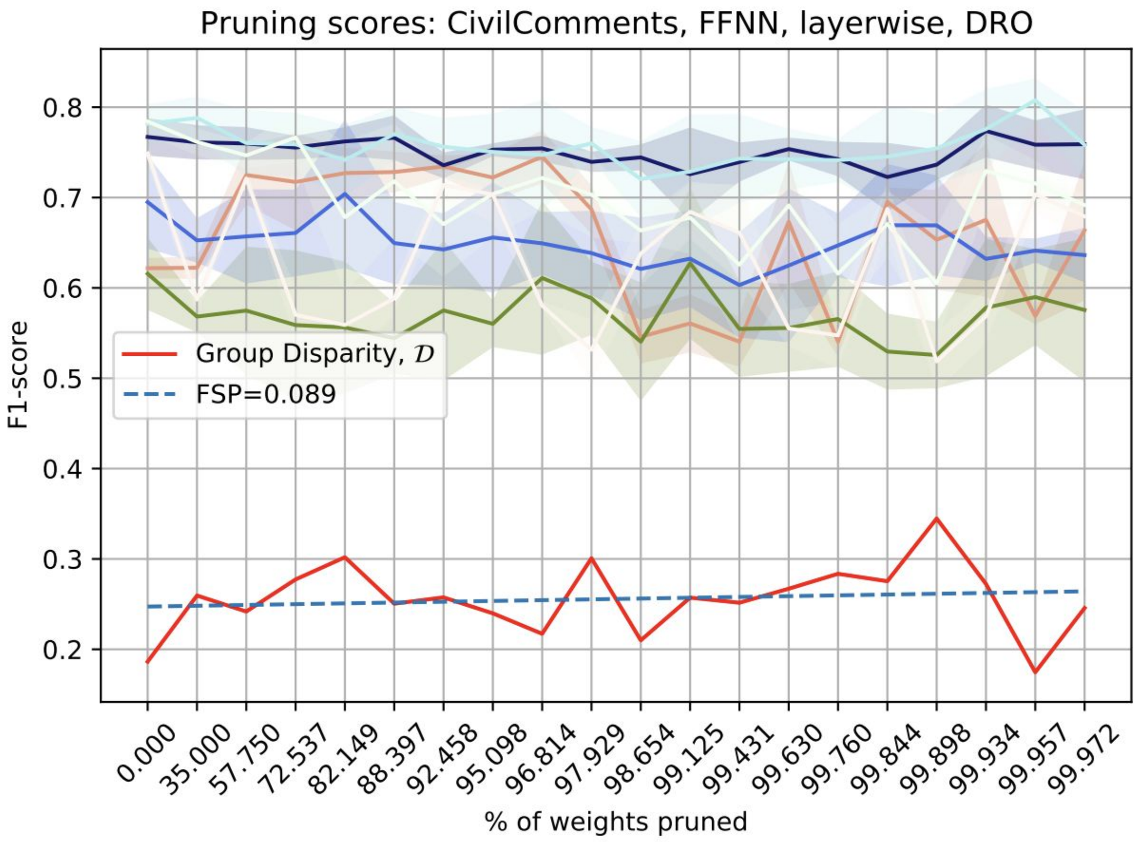
Weight pruning does come at a cost, however: fairness (Hansen and Søgaard, 2021).



Lottery ticket hypothesis

Weight pruning does come at a cost, however: fairness (Hansen and Søgaard, 2021).

Mitigation: Group Distributionally Robust Optimization? See [Sagawa et al. (2019)].

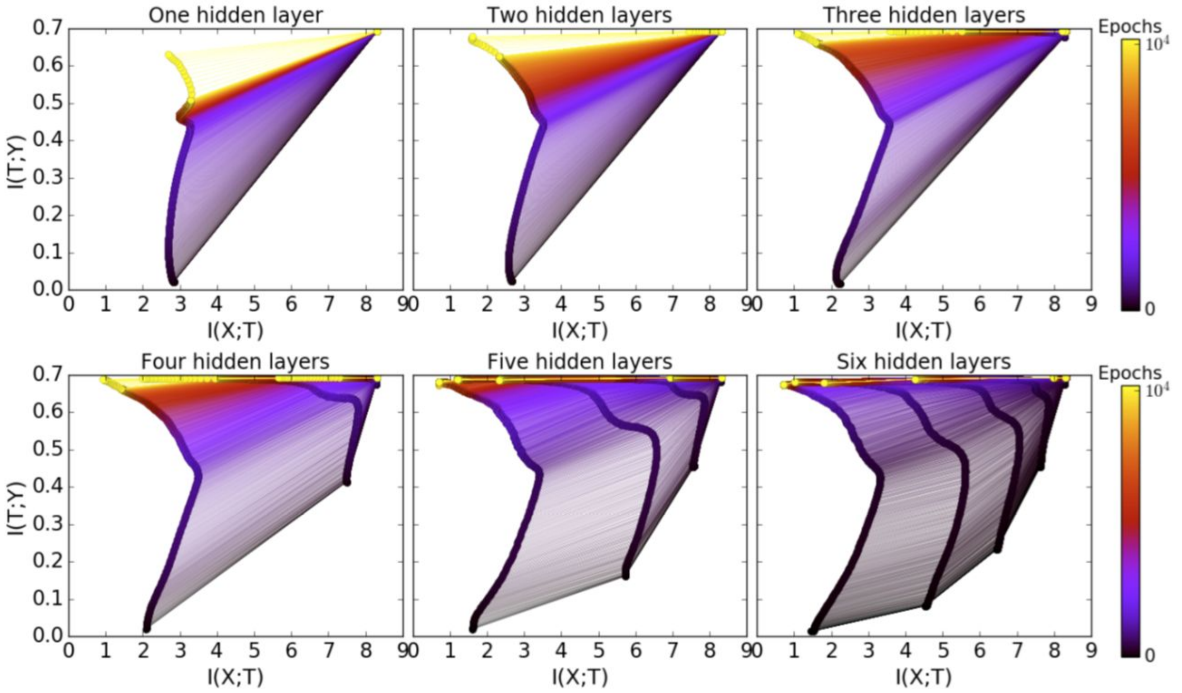


Bottleneck theory

—

Bottleneck theory

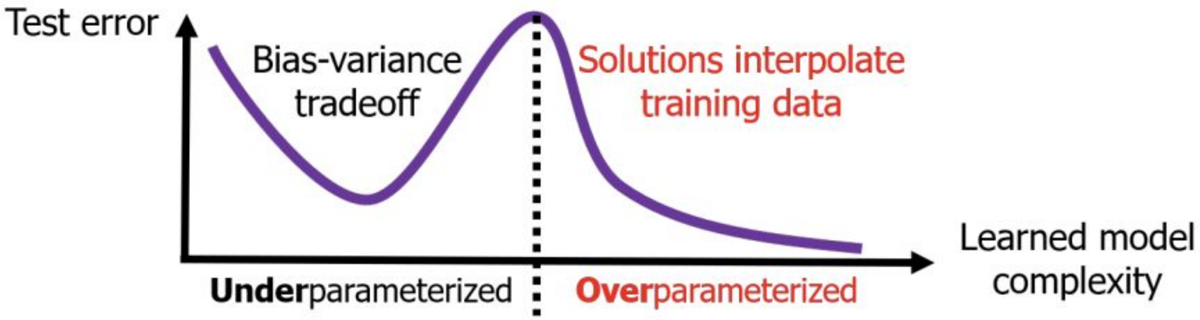
1. DNNs undergo two distinct phases consisting of an initial fitting phase and a subsequent compression phase
2. the compression phase is causally related to generalization performance
3. the compression phase occurs due to the diffusion-like behavior of stochastic gradient descent.

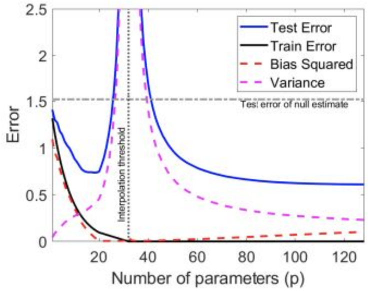


Over-paramerization

Double descent

Our intuitions in the under-parameterized regime are guided by the bias-variance trade-off, but this is less useful, it seems, in the over-parameterized regime.





Take-home messages

- Big models only make sense if you have sufficient data.
- Training for long only makes sense if you have both big models and big data.
- In that case, training for long **does** make sense, however, i.e., it gives better generalization.
- Once you have trained, you can typically distill smaller and more efficient models.