344.063/163 KV Special Topic:
Natural Language Processing with Deep Learning
Advanced Topics on Language Processing

Navid Rekab-Saz

navid.rekabsaz@jku.at

Institute of Computational Perception

JOHANNES KEPLER UNIVERSITY LINZ

Institute of Computational Perception

Agenda

- Model size and compression

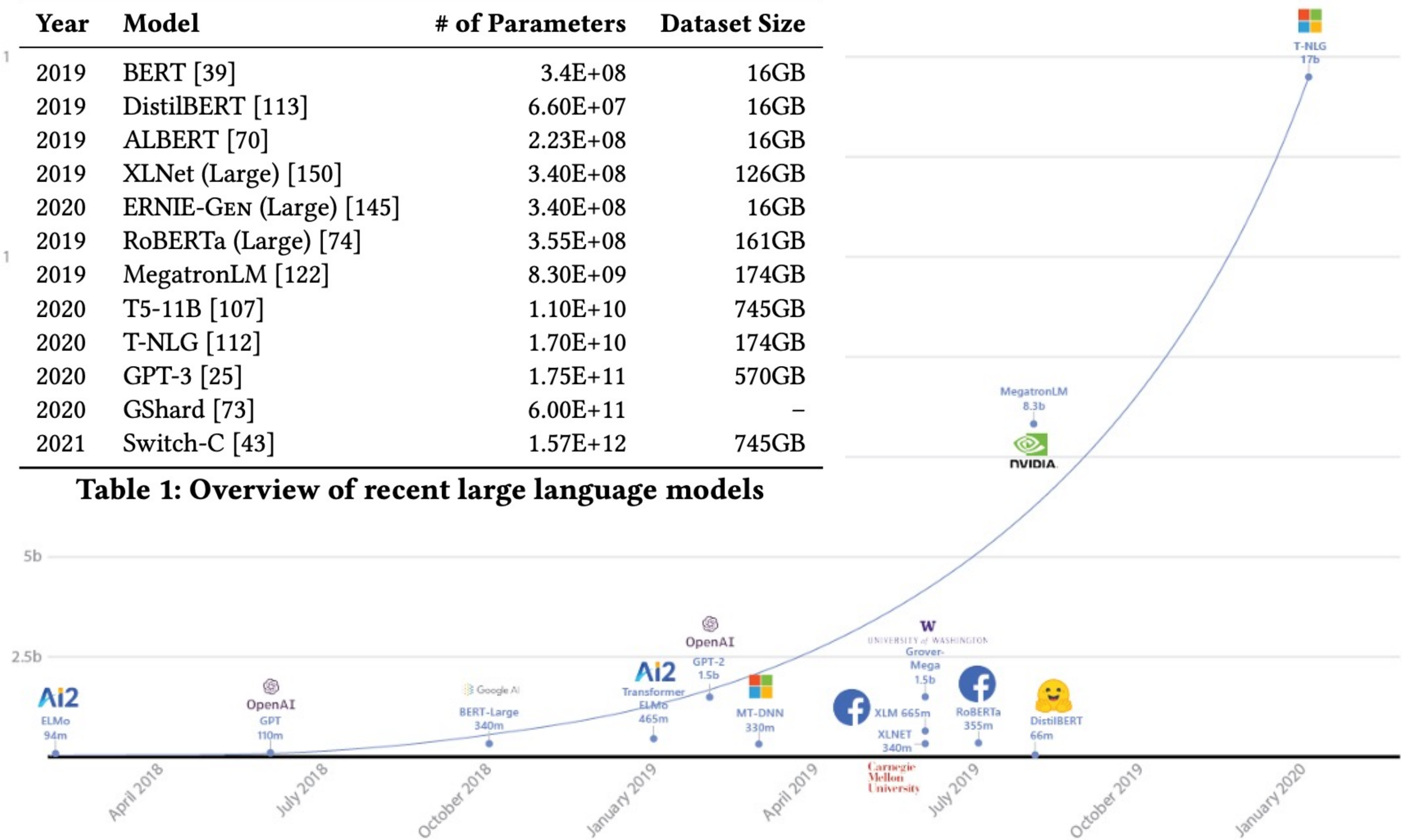
Agenda

- **Model size and compression**

Trend!

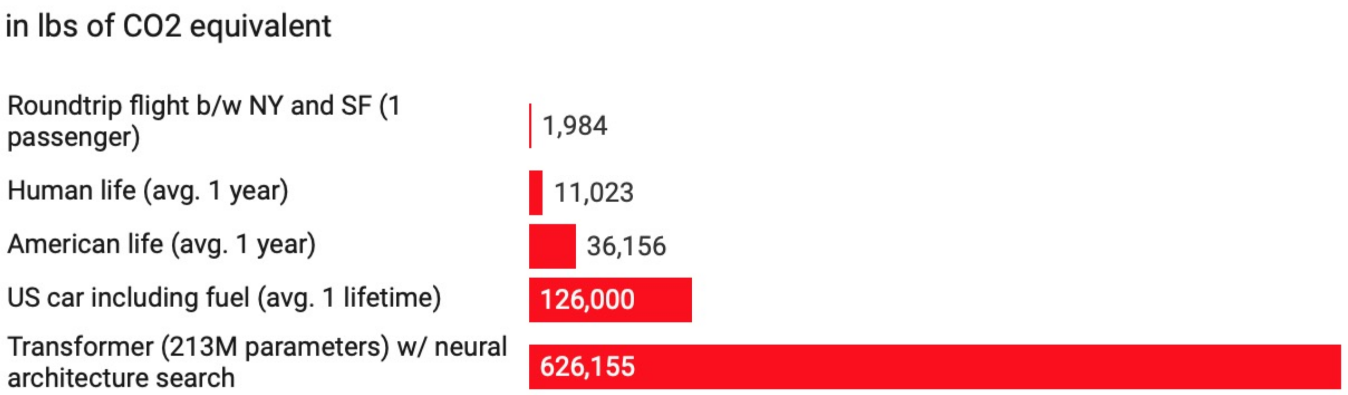
Year	Model	# of Parameters	Dataset Size
2019	BERT [39]	3.4E+08	16GB
2019	DistilBERT [113]	6.60E+07	16GB
2019	ALBERT [70]	2.23E+08	16GB
2019	XLNet (Large) [150]	3.40E+08	126GB
2020	ERNIE-Gen (Large) [145]	3.40E+08	16GB
2019	RoBERTa (Large) [74]	3.55E+08	161GB
2019	MegatronLM [122]	8.30E+09	174GB
2020	T5-11B [107]	1.10E+10	745GB
2020	T-NLG [112]	1.70E+10	174GB
2020	GPT-3 [25]	1.75E+11	570GB
2020	GShard [73]	6.00E+11	–
2021	Switch-C [43]	1.57E+12	745GB

Table 1: Overview of recent large language models



Source: https://www.groundai.com/project/distilbert-a-distilled-version-of-bert-smaller-faster-cheaper-and-lighter/1
Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021, March). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?🦜. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*

Carbon footprint of NLP



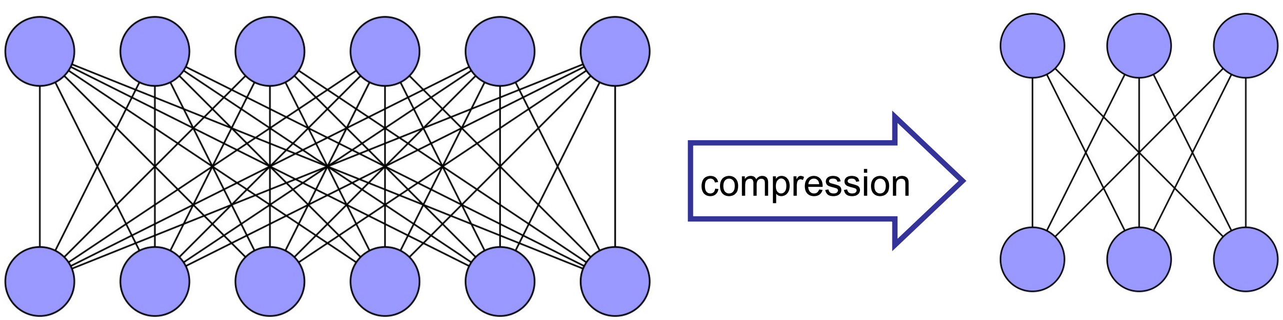
Model	Hardware	Power (W)	Hours	kWh·PUE	CO_2e	Cloud compute cost
$\text{Transformer}_{base}$	P100x8	1415.78	12	27	26	\$41–\$140
Transformer_{big}	P100x8	1515.43	84	201	192	\$289–\$981
ELMo	P100x3	517.66	336	275	262	\$433–\$1472
BERT_{base}	V100x64	12,041.51	79	1507	1438	\$3751–\$12,571
BERT_{base}	TPUv2x16	—	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

Strubell, E., Ganesh, A., & McCallum, A.. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of ACL* (2019).
Source: https://www.technologyreview.com/2019/06/06/239031/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/


Home
Shared Task
Organization
Program Comittee
SustaiNLP 2020
First Workshop on Simple and Efficient Natural Language Processing
Workshop at EMNLP 2020

Model compression

- Model compression methods reduce the size of a model
  - applied as a post-processing, but also during training
- A compressed model
  - Efficient in practice:
    - faster inference time
    - better suited to low-resource settings (i.e. on mobile phones)
  - Less energy consumption

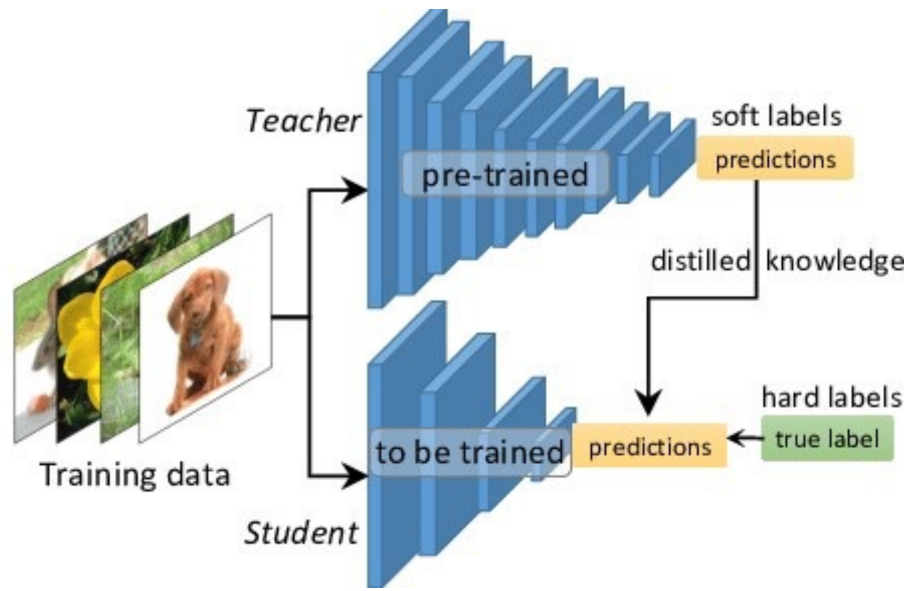


Model compression methods

- Knowledge distillation
  - A smaller model (student) is trained to reproduce the *behavior* of a larger model (teacher)
  - Student *mimics* teachers output or internal representations

Example: DistilBERT

- Distillation loss is defined according to the prediction probabilities of a pre-trained BERT
- Reduce the size to 40% while retaining 97% of performance on GLUE tasks



Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Figure source: https://towardsdatascience.com/knowledge-distillation-simplified-dd4973dbc764

Model compression methods

- Pruning
  - To reduce the extent of a network by removing the superfluous and unnecessary *neurons*, *nodes*, *heads*, *etc.*
  - Pruning can be done after training or during training

A common (post-processing) procedure

1. Train the model
2. Remove the unnecessary units, selected based on their
   - magnitudes, gradients, activations, etc.
3. Fine-tune the pruned network
4. Repeat the last two steps iteratively

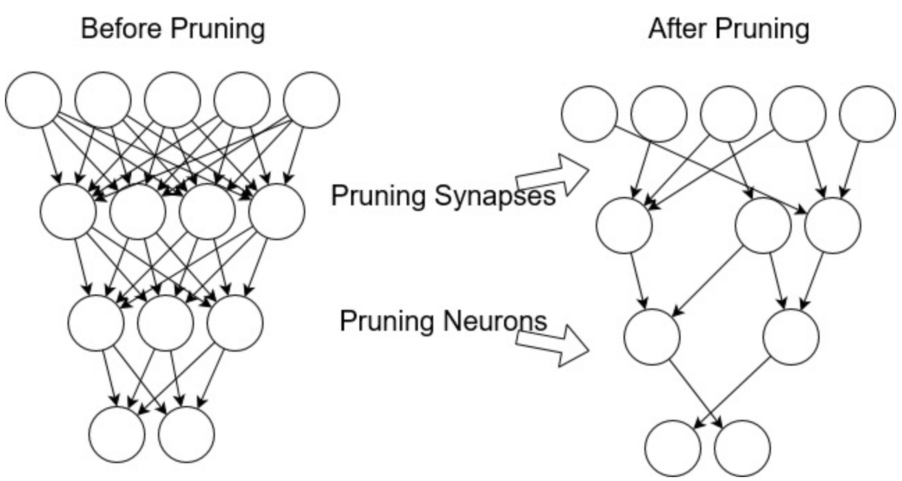
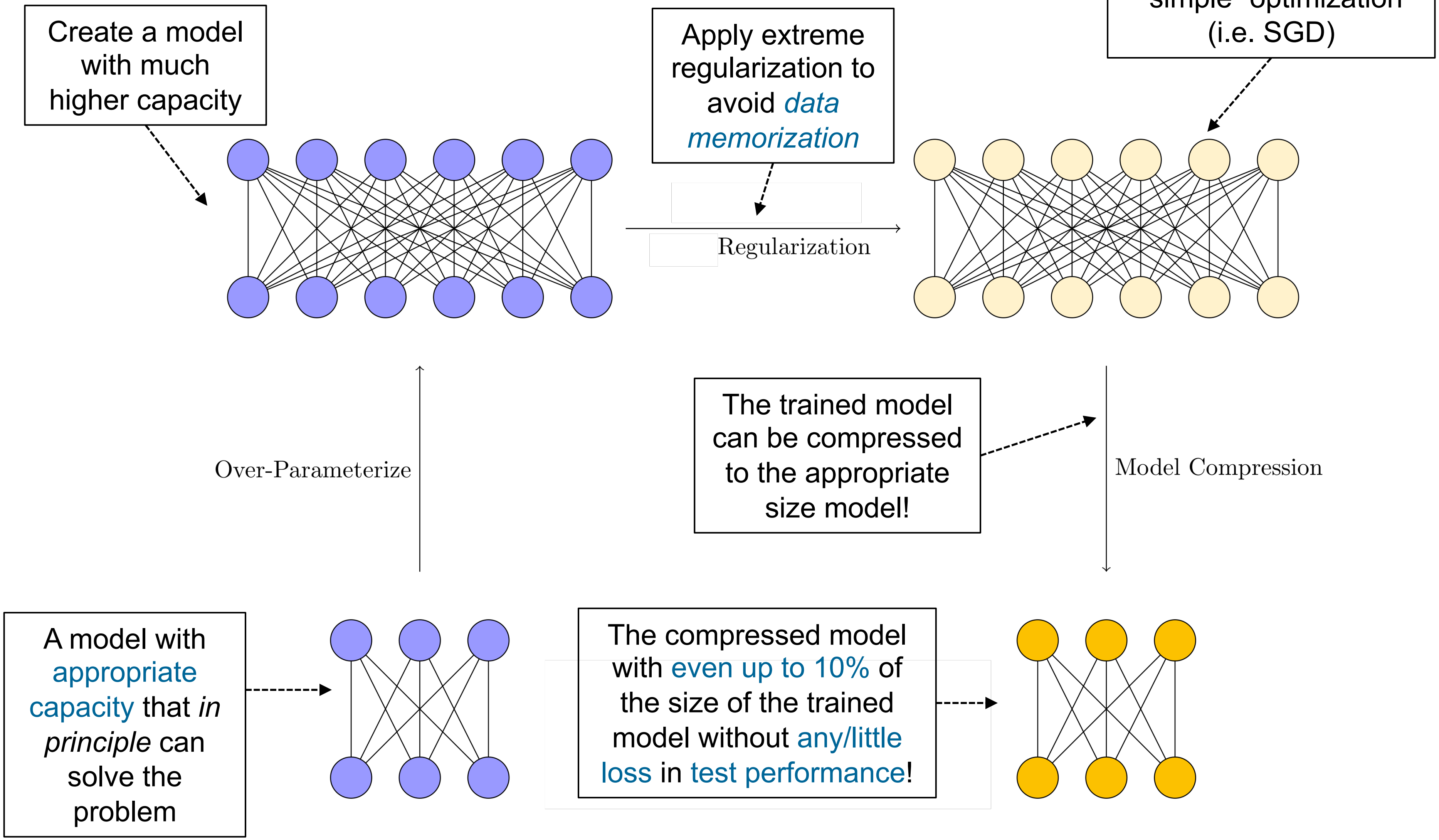


Figure source: https://software.intel.com/content/www/us/en/develop/articles/compression-and-acceleration-of-high-dimensional-neural-networks.html

Model compression methods

- Quantization
  - Quantization methods decrease the numerical precision of model parameters
    - For instance, by turning the 32-bit float parameters of a pre-trained model to 8-bit integers

Over-parameterization – a paradox!



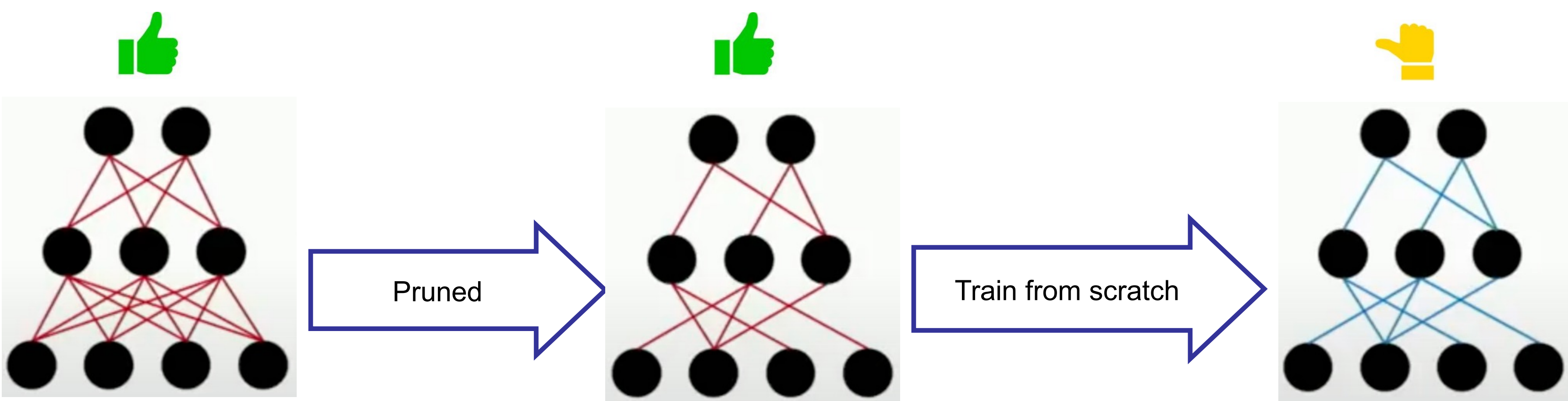
Over-parameterization is in paradox to classical ML doctrine of *"Thou shalt start simple, and then go complex."*

Figure adopted from here (a nice read): http://mitchgordon.me/machine/learning/2020/01/13/do-we-really-need-model-compression.html

Can we then start from small compressed models?!

Consider compression with pruning…

- What if we take the pruned model, and re-train it from scratch (with new initializations)?
  - It does not reach the same performance as the compressed model

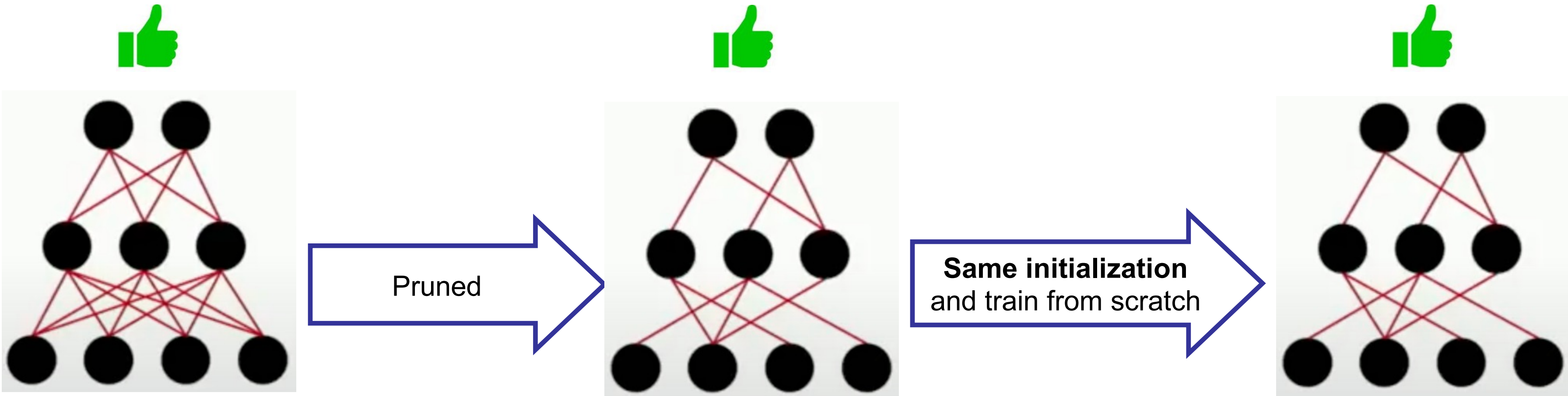


J. Frankle, M. Carbin *The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks.* Best paper of ICLR 2019

Can we then start from small compressed models?!

Consider compression with pruning…

- What if we take the pruned model, and re-train it from scratch (with new initializations)?
  - It does not reach the same performance as the compressed model
- However, if we take the structure of the pruned model, and use the same initialization as the original model …
  - We achieve the same or even better results!



J. Frankle, M. Carbin *The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks.* Best paper of ICLR 2019

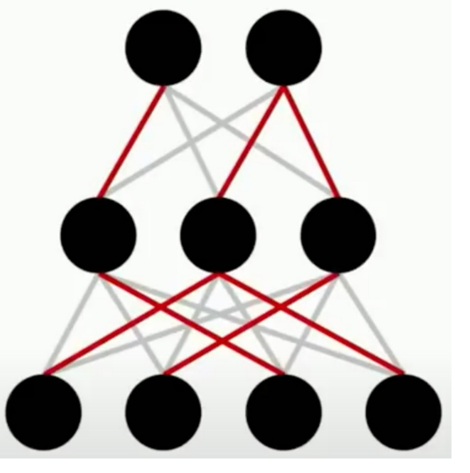
Lottery ticket hypothesis

- The Lottery Ticket Hypothesis (rephrased):

*dense, trainable networks contain sparse trainable subnetworks (i.e., winning tickets) that are equally capable**

** When trained in isolation, they reach test prediction comparable to the original network in a similar number of iterations.*

- Though, we don't know (yet) beforehand how to find these subnetworks
  - What are their structures?
  - What are their initializations?
- But if we do … we can achieve the same results by training much smaller networks



J. Frankle, M. Carbin *The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks.* Best paper of ICLR 2019