Remarks on Deep Learning (I)

Alberto TONDA, Ph.D. (Senior permanent researcher, DR)

UMR 518 MIA-PS, INRAE, AgroParisTech, Université Paris-Saclay
UAR 3611, Institut des Systèmes Complexes de Paris Île-de-France

Outline

- Neural Networks vs Deep Learning
- Why are Neural Networks winning?
- Overparametrization
- Practical obstacles

Neural Networks vs Deep Learning

- What is the difference?

Neural Networks vs Deep Learning

- (1) Considerable **improvements** over classic Neural Networks
  - New (more effective!) **algorithms to optimize** parameters
  - New architectures to deal with **relational data**
  - Better **software engineering**
  - More **computing power** available, better results
- (2) Rebranding
  - Interest for Neural Networks declined in the 1990s
  - At the time, considered less effective than other methods
- Any network with more than one hidden layer is “deep”

Neural Networks vs Deep Learning

- (1) Considerable **improvements** over classic Neural Networks
  - New (more effective!) **algorithms to optimize** parameters
  - New architectures to deal with **relational data**
  - Better **software engineering**
  - More **computing power** available, better results
- (2) Rebranding
  - Interest for Neural Networks declined in the 1990s
  - At the time, considered less effective than other metho
- Any network with more th

But sometimes, “Deep Learning” with just **one layer**! Inclusive term :-D

Why are neural networks winning?

- Lots of competitors! Why is Deep Learning so dominant?

Why are neural networks winning?

- On **tabular data**, they are *not* winning
  - Current consensus is that ensembles of boosted trees are better
  - See Grinsztajn et al. (2022) *Why do tree-based models still outperform deep learning on tabular data?*
  - This could change with pretrained models like TabPFN

- On **relational data**, they are *massively* dominant
  - Remove the need for feature construction (and maybe selection)
  - Any type of structure: sequences, graphs, images, videos, …

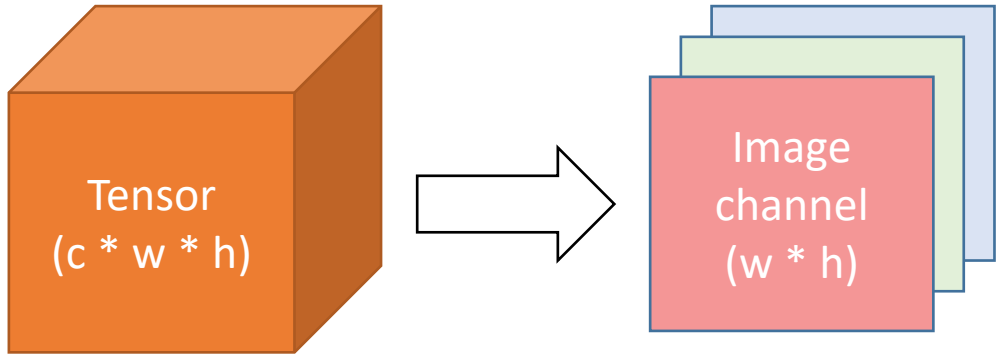
Why are neural networks winning?

- Generative models: how can a NNs output an *image*?

Why are neural networks winning?

- Output tensors interpreted as (approximately) *anything*!

Example: image

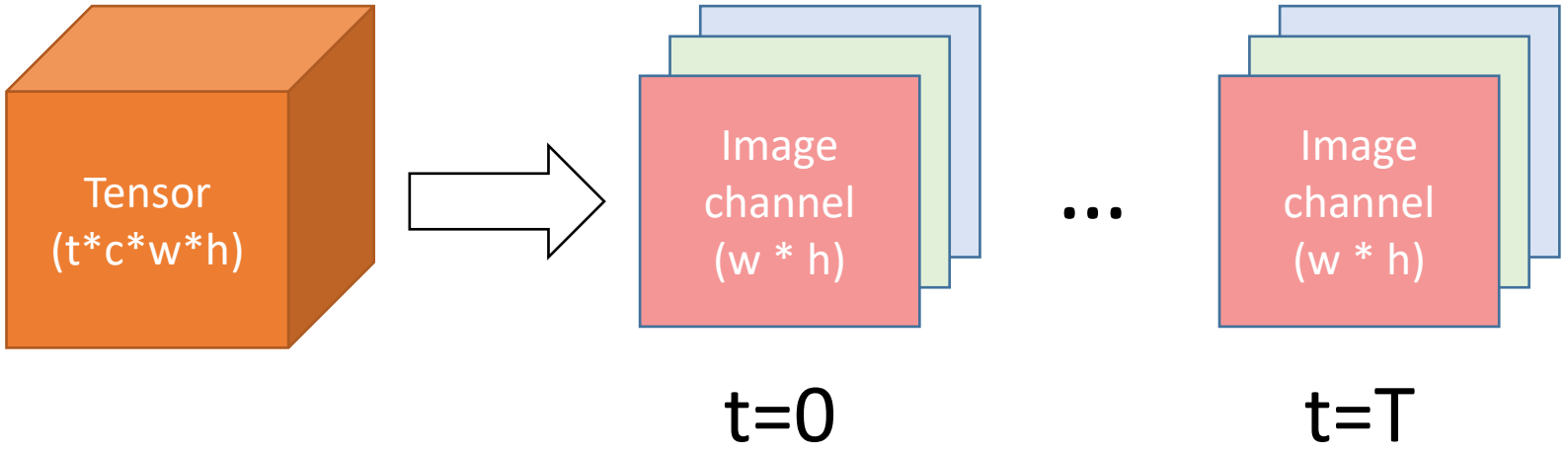


Each value $x_{i,j}$ inside each channel indicates the level or red, blue, or green for pixel $p_{i,j}$ respectively

Why are neural networks winning?

- Output tensors interpreted as (approximately) *anything*!

Example: video



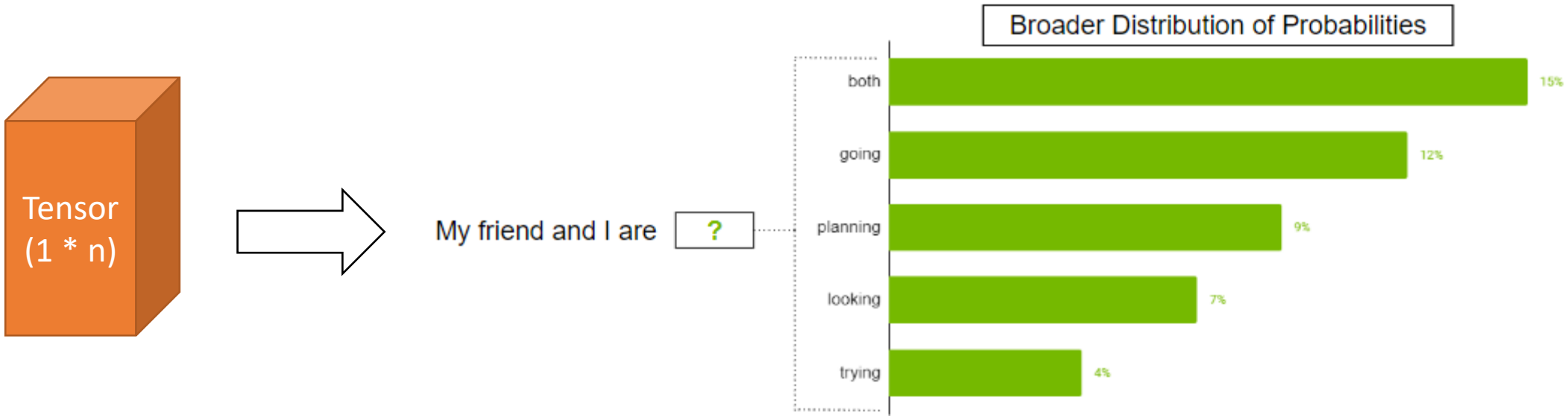
Each value $x_{i,j}^{t}$ inside each channel indicates the level or red, blue, or green for pixel $p_{i,j}$ for the frame at time t

Why are neural networks winning?

- And when we have to choose between *n* discrete values?

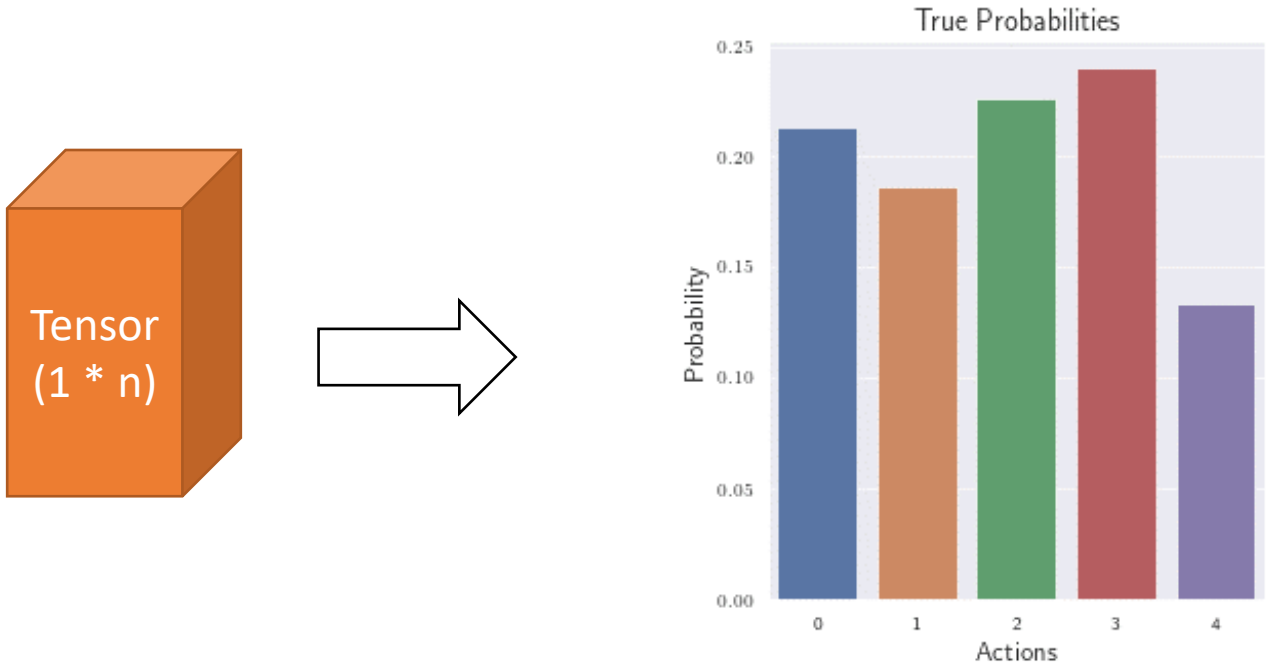
Why are neural networks winning?

- Tensor is interpreted as a *probability distribution*
- Each cell in output tensor is “probability” of picking element



Why are neural networks winning?

- Tensor is interpreted as a *probability distribution*
- Each cell in output tensor is “probability” of picking element



Why are neural networks winning?

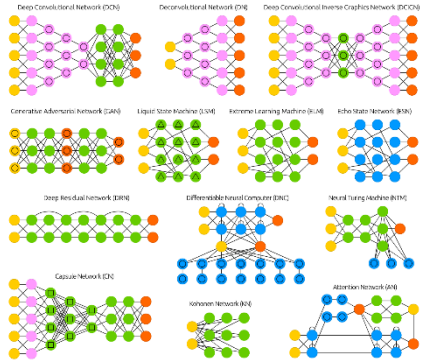
- General approach, works well in practice

Chaotic and dynamic reality

Tensor
(n_1 * n_2 * n_3
…)

Input tensor
(fixed size, meaningful)

Some sort of neural network



Tensor
(d_1 * d_2 * d_3
…)

Output tensor
(fixed size, interpreted)

Why are neural networks winning?

- General approach, works well in practice

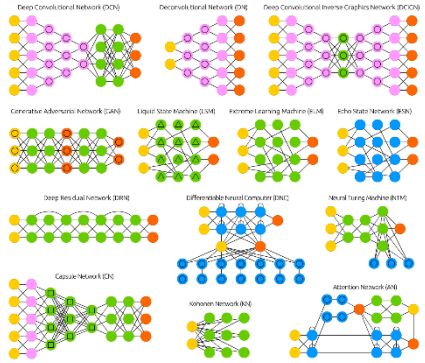
Chaotic and dynamic reality

?

Tensor
(n_1 * n_2 * n_3
…)

Input tensor
(fixed size, meaningful)

Some sort of neural network



Tensor
(d_1 * d_2 * d_3
…)

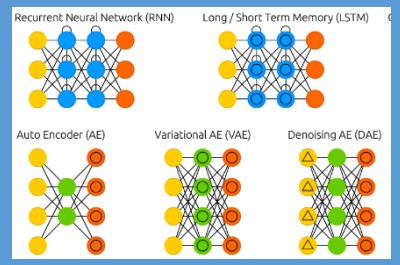
Output tensor
(fixed size, interpreted)

Why are neural networks winning?

- General approach, works well in practice

Chaotic and dynamic reality

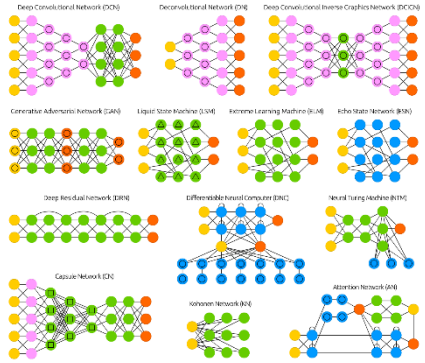
Feature construction



Tensor
$(n_1 * n_2 * n_3 \ldots)$

Input tensor
(fixed size, meaningful)

Some sort of neural network



Tensor
$(d_1 * d_2 * d_3 \ldots)$

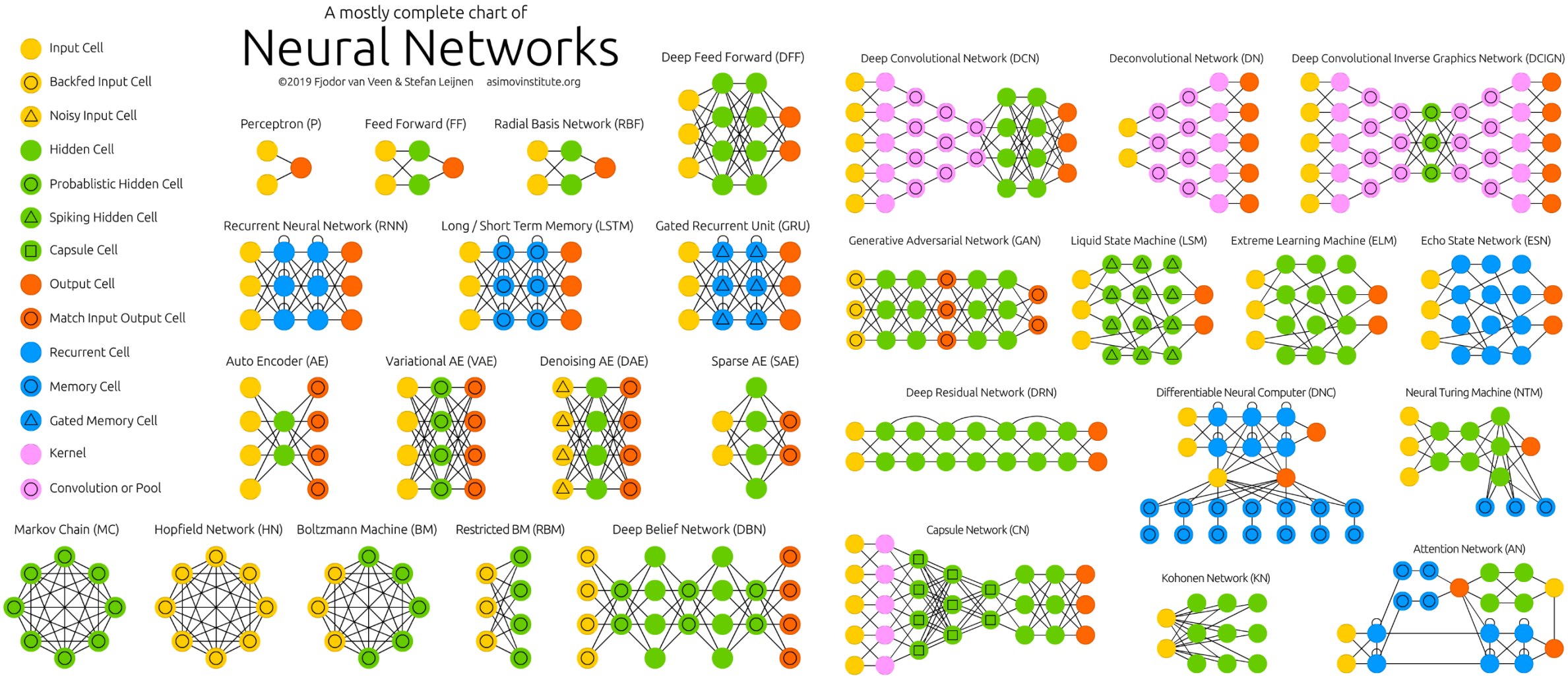
Output tensor
(fixed size, interpreted)

Why are neural networks winning?

- Transfer learning
  - Really started with Deep Learning
  - Using multiple layers creates a “funnel” of knowledge
  - Early layers learn generic problem information, can be **reused**!
  - Checkpoints, restarting training is *easy*

- Not unique to neural networks
  - But it is still an under-researched topic for other ML algorithms
  - It’s probably more straightforward for DL

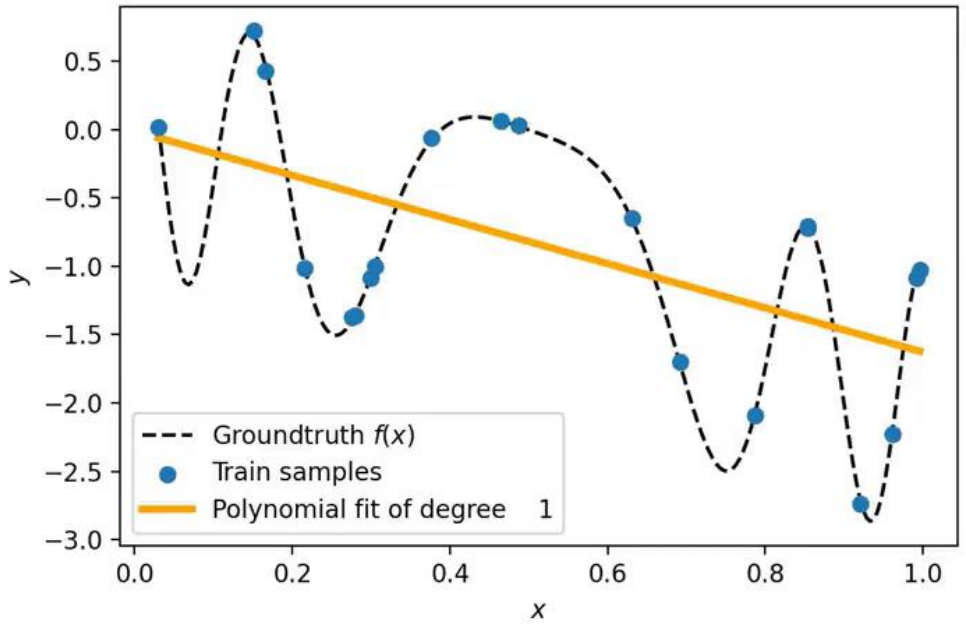
Multiple architectures!

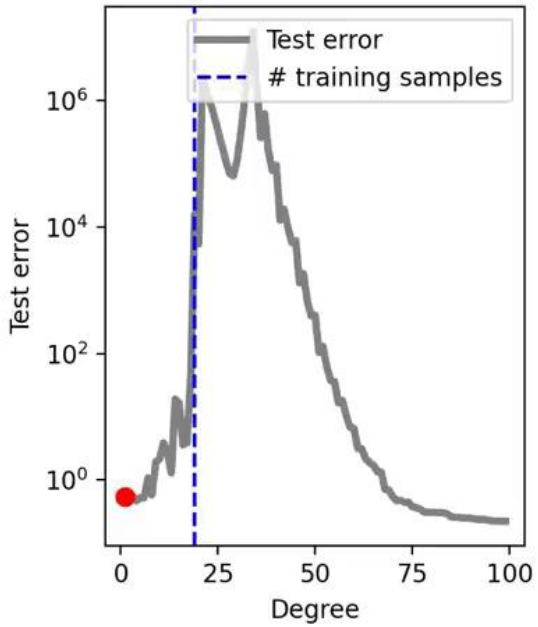


https://www.asimovinstitute.org/neural-network-zoo/

Overparametrization

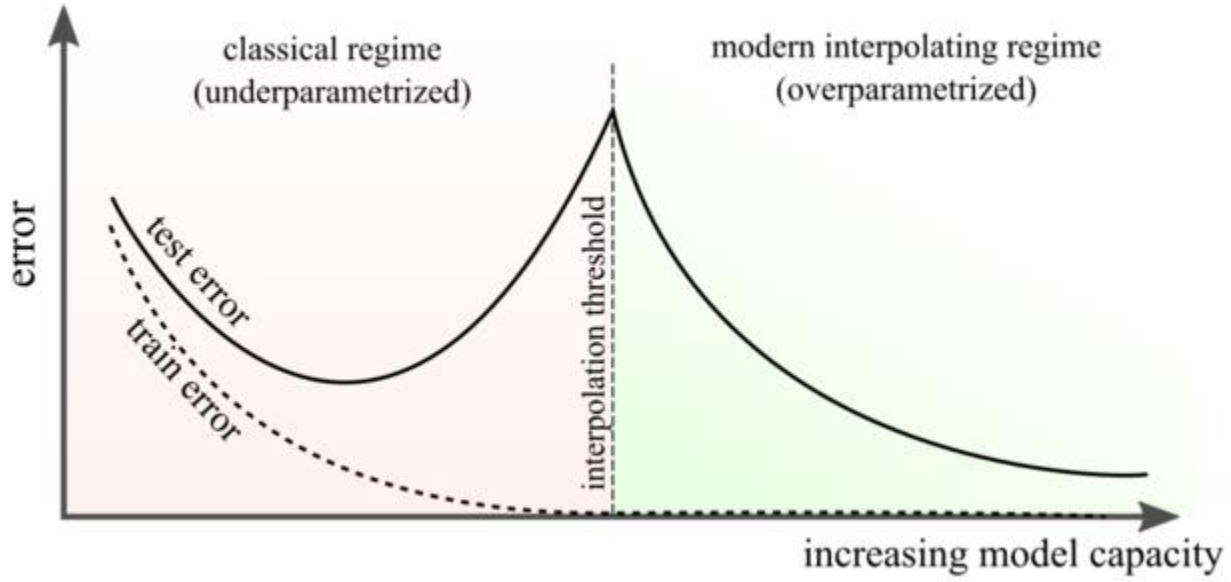
- Wait a second!
  - From ML, we learned that having too many parameters is **bad**!
  - Models with too many parameters tend to **overfit** terribly…right?





Overparametrization

- What is happening here? Well, we don't really know
- Empirical results, **overparametrizing** improves **generalization**
- "Double descent" or "W figure"



Overparametrization

- Research and discussion are still ongoing

A U-turn on Double Descent: Rethinking Parameter Counting in Statistical Learning

Alicia Curth*
University of Cambridge
amc253@cam.ac.uk

Alan Jeffares*
University of Cambridge
aj659@cam.ac.uk

Mihaela van der Schaar
University of Cambridge
mv472@cam.ac.uk

Curth, Jeffares, and van der Schaar. 2023. In: Proceedings of NeurIPS

Worst enemies of DL (in this class, at least)

- (Disk/Memory) Space
  - Modern DL models contain **billions** (10^9) of parameters
  - GPT-4 has ~220 billion parameters
  - If a single parameter is a floating point value on 32 bits…
  - …that is 220 * 10^9 * 32 / 8 = 880 Giga Bytes (!!!)
  - They have to be stored on a hard drive and IN MEMORY!
  - During training, you'll need to also store the gradient IN MEMORY!
- (Training) Time
  - Training takes **a lot** of time, depending on parameter count

Spoiler alert: we are going to have issues

- Downloading a 1GB (small) model from HuggingFace takes quite a long time on Google Colaboratory
  - Maybe we could use downcasting to int8 or int4
- Inference (using the model in prediction) is *slow*
  - Maybe Colaboratory GPUs…?

Questions?

Bibliography

- Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). *Why do tree-based models still outperform deep learning on typical tabular data?*. Advances in Neural Information Processing Systems, 35.
- Curth, A., Jeffares, A., & van der Schaar, M. (2024). *A U-turn on double descent: Rethinking parameter counting in statistical learning*. Advances in Neural Information Processing Systems, 36.