An Introduction to deep learning

Ard Louis

UNIVERSITY OF OXFORD

Learning machines?

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's ? If this were then subjected to an appropriate course of education one would obtain the adult brain …..

We have thus divided our problem into two parts. The child-programme and the education process.

Alan Turing, *Computing Machinery and Intelligence,* Mind **59**, 433 (1950)

History of modern AI: Hype and AI winters

1950 – Turing test

First automated translators

1960

1957 Rosenblatt invents perceptron

The Navy revealed the embryo of an electronic computer that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence. NYT 1957

1970

1969 Perceptrons book by Minsky and Papert – connectionism takes a big hit

1st AI winter

1973 -- Lighthill report – combinatorial explosion will make AI only suitable for toy problems
No major UK investment till 1983

1980

1980's – Expert systems, XCON, LISP based companies etc…

1990

2nd AI winter

[Investors] were put off by the term 'voice recognition' which, like 'artificial intelligence', is associated with systems that have all too often failed to live up to their promises, Economist 2007

2000

2010

2012 AlexNet wins Imagenet 2012 competition, deep learning era begins

2020

AI is one of the most profound things we're working on as humanity. It's more profound than fire or electricity.

Google CEO Sundar Pichai
At World Economic Forum in Davos, 2020

Image from https://www.zdnet.com/article/why-is-a-i-reporting-so-bad/

March 2016 – Alpha Go beats Lee Sedol, 18 times world champion at Go

Dec, 2017 Alpha Go Zero beats Alpha Go, but playing only against itself.
It can also beat top chess computers and "learns" the game from "scratch".

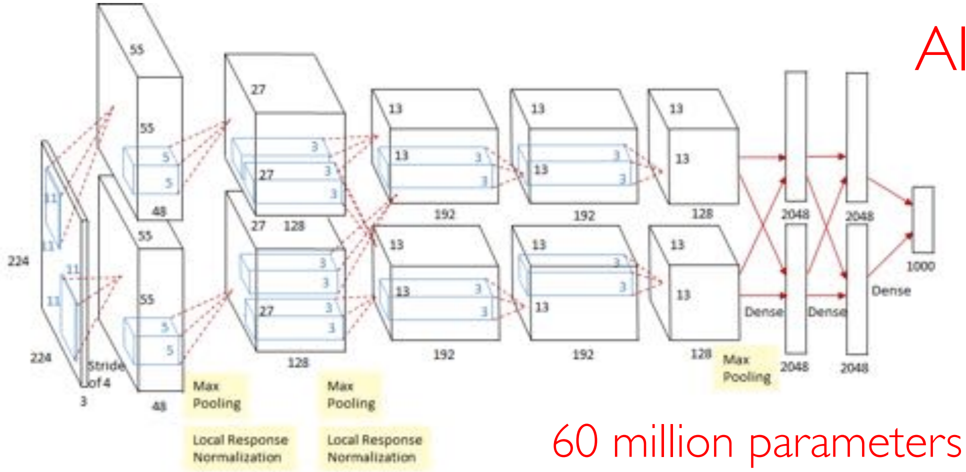
2012 – start of the deep-learning era

14 million images
20,000 categories

Annual competition

Fei-Fei Li

Alex Net

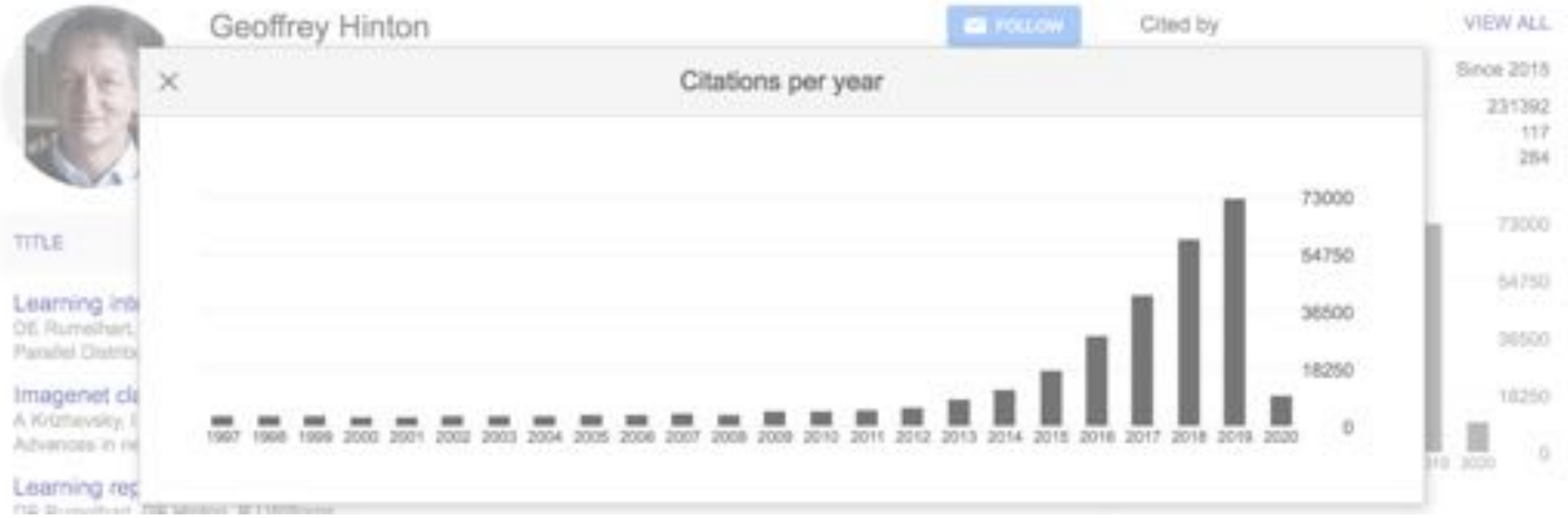


60 million parameters

2012 -- a team from U of Toronto used a deep neural network (Alex Net) to beat all competitors with 40% lower error.

Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, Imagenet classification with deep convolutional neural networks
Advances in neural information processing systems, 1097 (2012)

Growth and growth of deep learning research



Top 3 of the 5 most cited Nature papers in 2019 are on deep learning

Deep learning has revolutionized artificial intelligence

2019 Turing Award (highest prize in computer science)
Yann LeCun, Geoffrey Hinton and Yoshua Bengio,

For many years these pioneers worked without much recognition:
Hinton on the referee report for an AI conference submission "*It said, Hinton's been working on this idea for seven years and nobody's interested, it's time to move on,*"

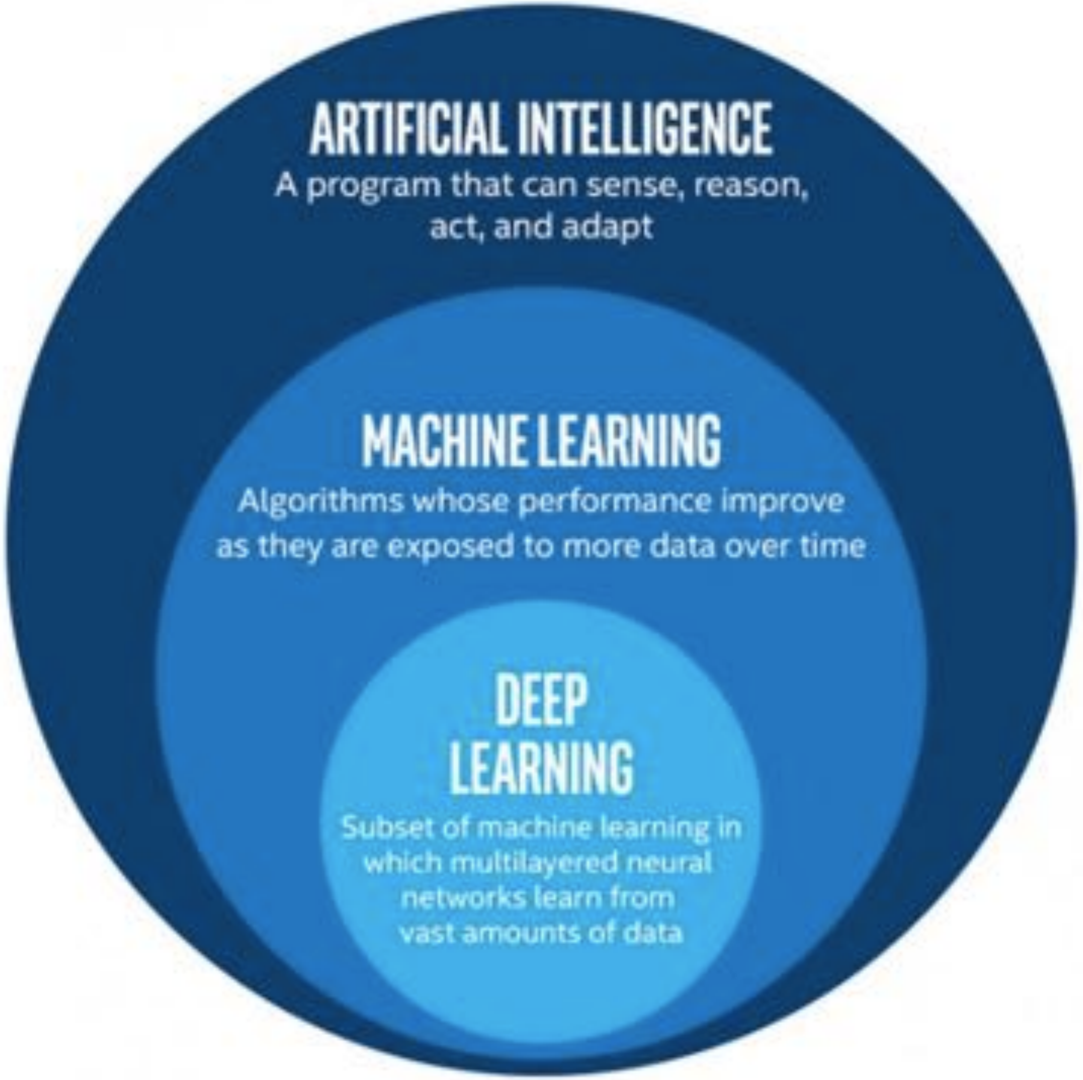
Will machine learning revolutionise Physics?

https://physicsworld.com/a/a-machine-learning-revolution/ (March 2019)

-- many applications, for example

- Data analysis (long standing, e.g. in particle physics)
- Image analysis
  - E.g. biological physics, astrophysics, etc…
  - Analysis of quantum states in experiment (see e.g. Nature 570, 484 (2019))
- Approximating quantum many-body wave function
- Finding new materials
- Control experiments
- Much more (see next two talks for some cool examples)

Basics



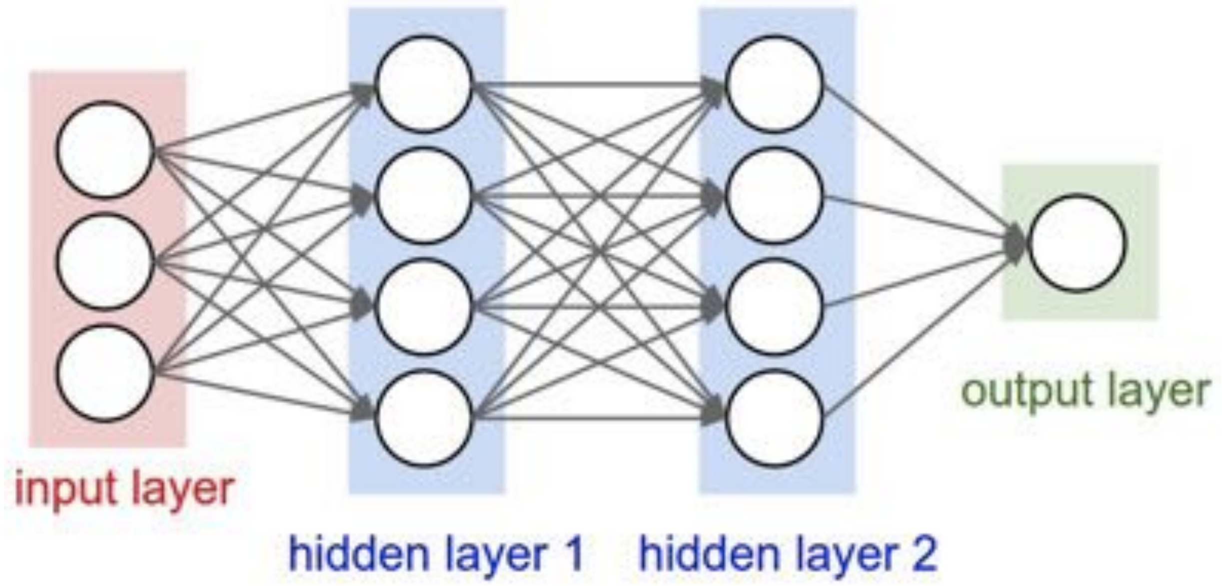
https://interestingengineering.com/whats-the-difference-between-machine-learning-and-ai

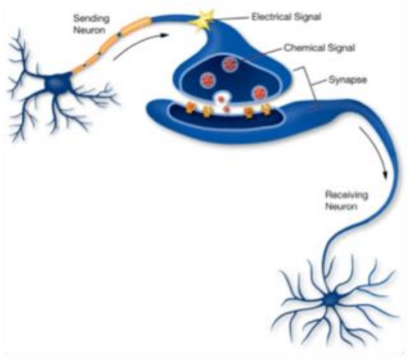
Basics

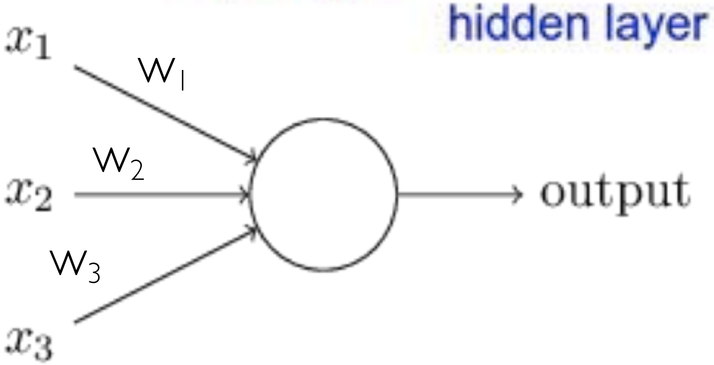
We have thus divided our problem into two parts. The child-programme and the education process.

- A Turing (1950)

Child-programme: Neural Network







$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases}$$

Basics

We have thus divided our problem into two parts. The child-programme and the education process.

- A Turing (1950)

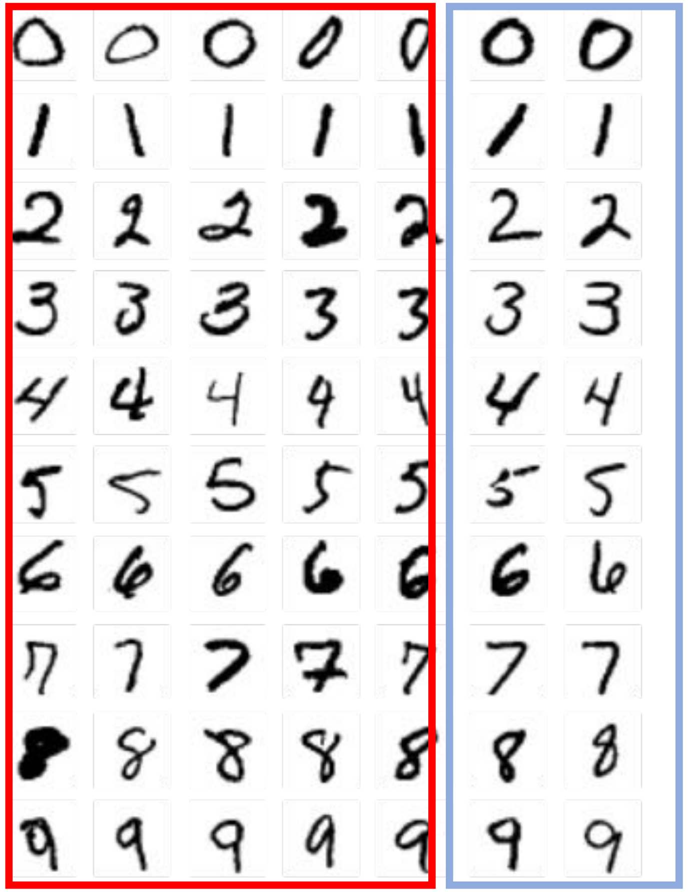
Education process:

1) Supervised learning

First: pick a training set *to find parameters*

Next: apply network to a test set *of unseen data*

How well you do on unseen data Is called ***generalization***



Basics

We have thus divided our problem into two parts. The child-programme and the education process.

- A Turing (1950)

Education process:

1) Supervised learning
2) Reinforcement learning
   - Parameters are updated with some kind of cumulative reward. AlphaZero is a reinforcement learning system.

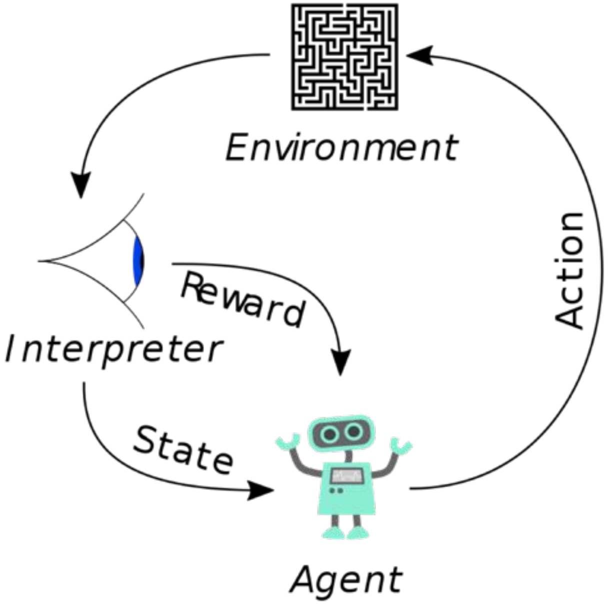


Image: wikipedia

Basics

We have thus divided our problem into two parts. The child-programme and the education process.

- A Turing (1950)

Education process:

1) Supervised learning
2) Reinforcement learning
3) Unsupervised learning
   Patterns are learned from unlabeled data

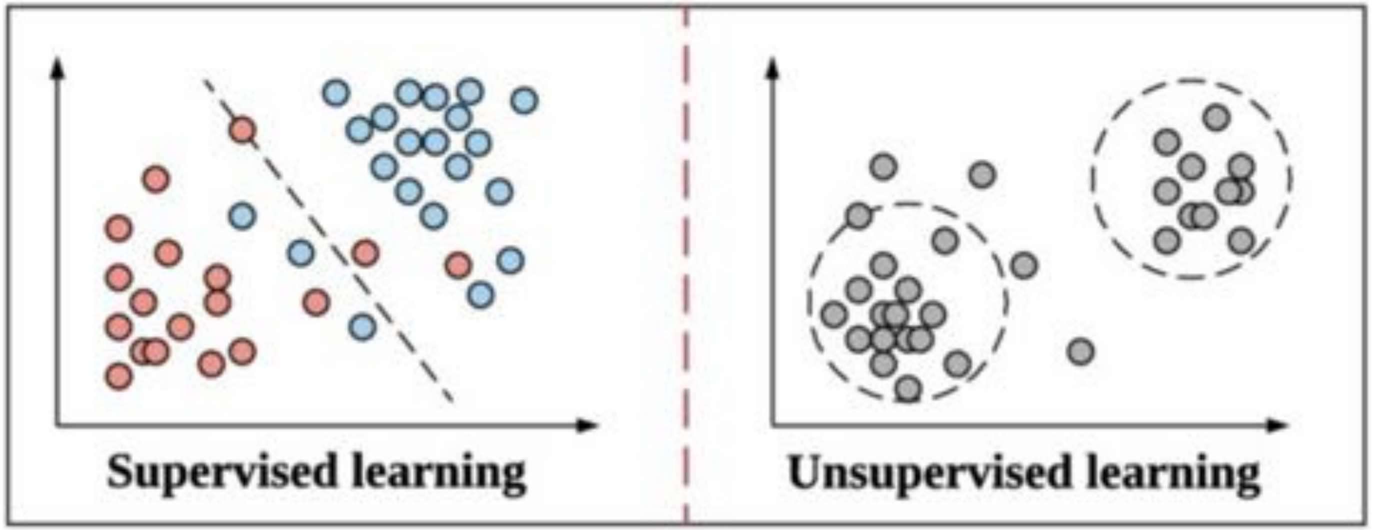


Figure: B. Qian arxiv: 1910.05433

Why do DNNs work so well?

Universal approximation theorem for NN

Neural networks are fundamentally function approximators. The following theorem holds:

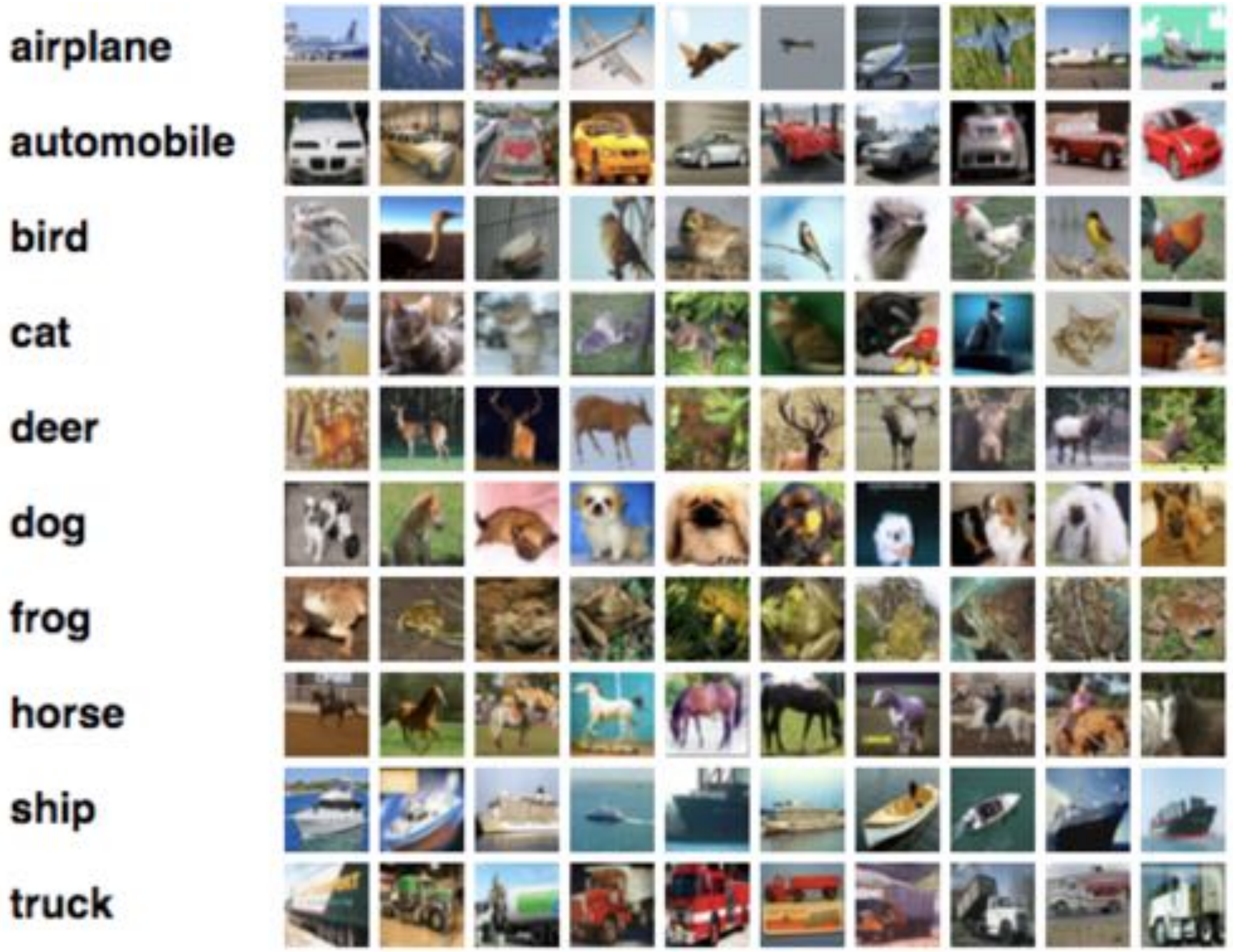
For any Lebesgue-integrable function $f : \mathbb{R}^n \to \mathbb{R}$ and any $\epsilon > 0$, there exists a fully-connected ReLU network $\mathcal{A}$ with width $d_m \leq n + 4$, such that the function $F_{\mathcal{A}}$ represented by this network satisfies

$$\int_{\mathbb{R}^n} |f(x) - F_{\mathcal{A}}(x)| \, \mathrm{d}x < \epsilon$$

Neural networks are highly expressive -

B. Hanin Approximating Continuous Functions by ReLU Nets of Minimal Width. arXiv preprint arXiv:1710.11278.

Conundrum: if DNNs are highly expressive, why do they pick functions that generalize so well?



CIFAR-10 dataset

C. Zhang et al., Understanding deep learning requires rethinking generalization.
arXiv:1611.03530 (2016)
Showed that you could randomise the labels, and still easily train to zero training error.

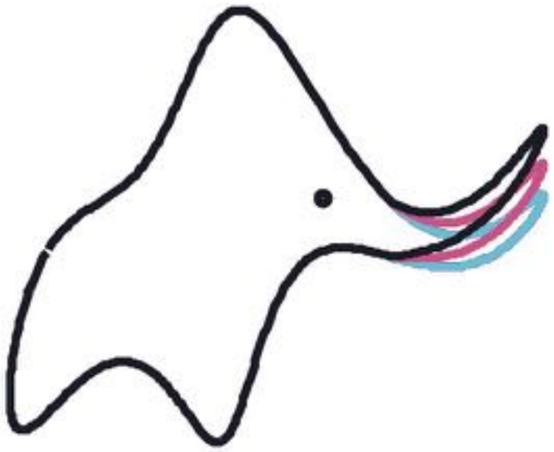
If a DNN can "memorize" a dataset, why does it pick functions that generalise so well?

Neural networks are typically highly over-parameterized: number of parameters >> number of data points

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk
-- John von Neuman (according to Fermi)

F. Dyson, A meeting with Enrico Fermi.
Nature. 427, 287 (2004)

4 parameters

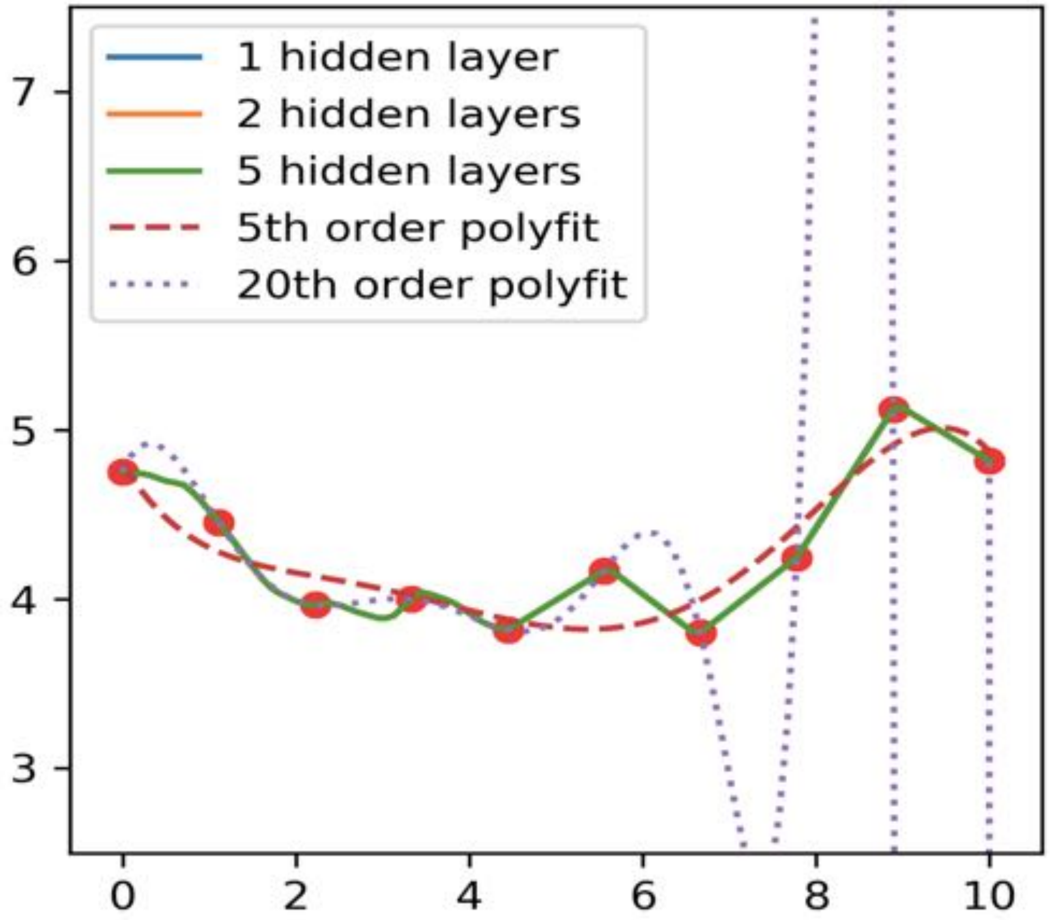


5 parameters

AI researchers allege that machine learning is alchemy
M Hutson - Science, 2018

Drawing an elephant with four complex parameters
Jürgen Mayer; Khaled Khairy; Jonathon Howard; American Journal of Physics 78, 648-649 (2010)

Neural networks are typically highly over-parameterized: number of parameters >> number of data points



Why do the DNNs not over-fit?

Comparison of a polynomial fit to a DNN fit (with thousands of parameters)

AIT Coding theorem for input-output maps

Kamal Dingle

NOTE: upper bound only!

$$\boxed{P(x) \lesssim 2^{-a\tilde{K}(x)-b}}$$

(2 Dphils of work)

Chico Camargo

Kolmogorov complexity K(x)= the length of shortest program that describes x on a UTM

INTUITION: likelihood P(x) for a monkey to type

01010101010101010101010101010101010101001010101010101010101010101010101010100101010101

01010101010101010101010101010101010101001010101010101010101010101010101010100101010101

On a binary keyboard, P(x) = 1/2^100

Into a programming language "print "01" 50 times" , P(x) ~1/2^19

K. Dingle, C. Camargo and A.AL, Nature Communications 9, 761 (2018); K Dingle, G.Valle Perez, and AAL, arxiv:1910.00971.

DNNs as an input-output map

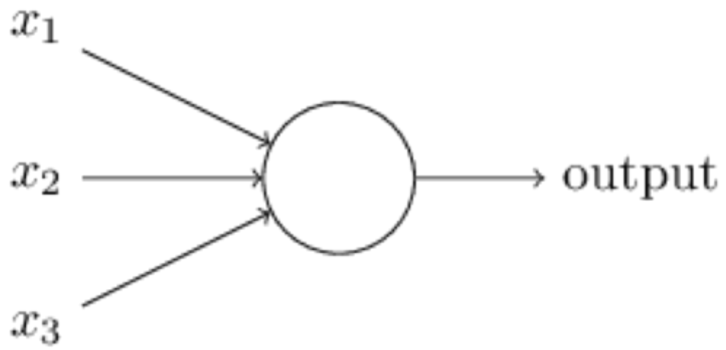
Input = parameters of the DNN
Output = the function it produces

Let the space of functions that the model can express be $\mathcal{F}$. If the model has p real valued parameters, taking values within a set $\Theta \subseteq \mathbb{R}^p$,
the parameter-function map, $\mathcal{M}$, is defined as:

$$\begin{aligned} \mathcal{M} : \Theta &\to \mathcal{F} \\ \theta &\mapsto f_\theta \end{aligned}$$

where f_θ is the function implemented by the model with choice of parameter vector θ.

A-Priori probability: If we randomly sample parameters θ, how likely are we to produce a particular function f?



Chris Mingard

Theorem 4.1. *For a perceptron f_θ with $b = 0$ and weights w sampled from a distribution which is symmetric under reflections along the coordinate axes, the probability measure $P(\theta : \mathcal{T}(f_\theta) = t)$ is given by*

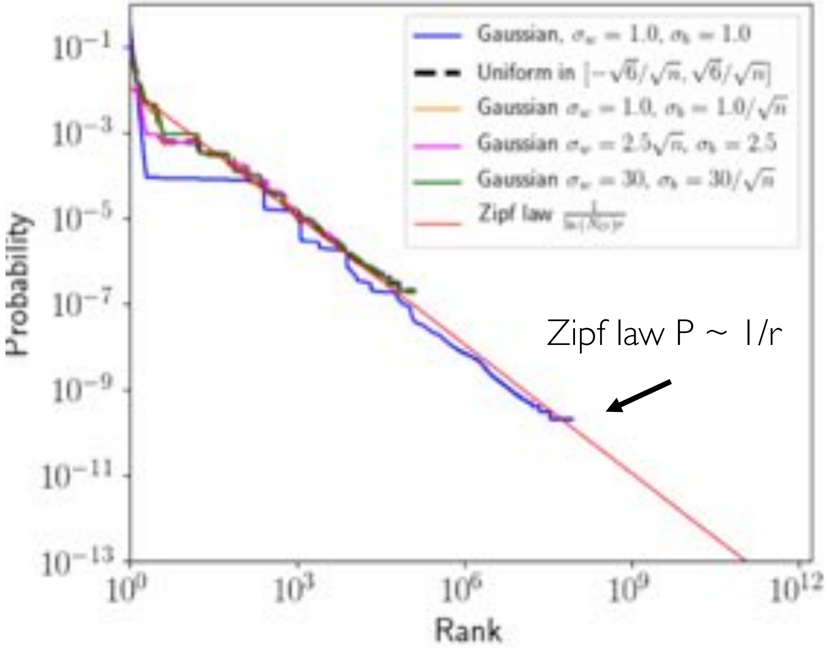
$$P(\theta : \mathcal{T}(f_\theta) = t) = \begin{cases} 2^{-n} & \text{if } 0 \leq t < 2^n \\ 0 & \text{otherwise} \end{cases} .$$

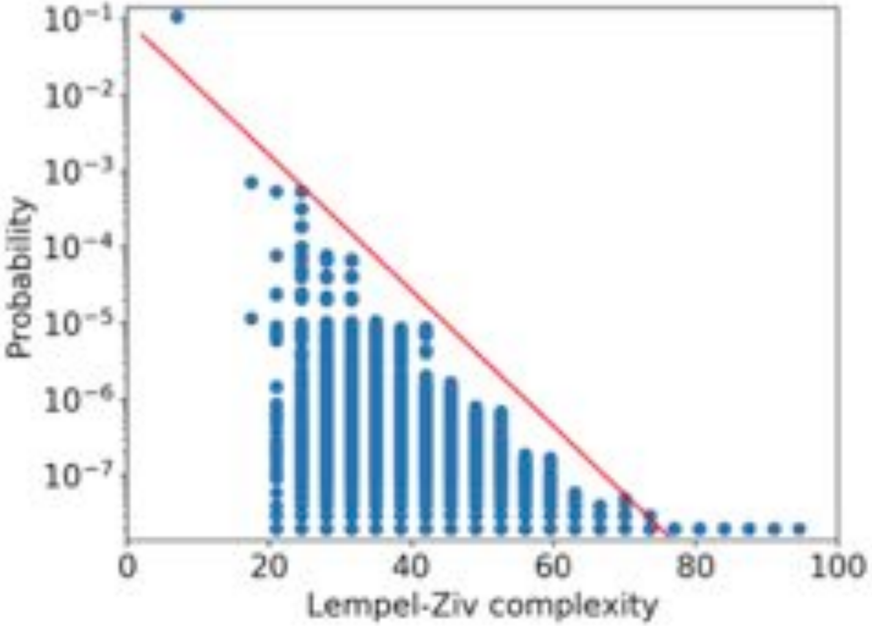
Neural networks are a priori biased towards Boolean functions with low entropy, Chris Mingard, Joar Skalse, Guillermo Valle-Pérez, David Martínez-Rubio, Vladimir Mikulik, Ard A. Louis arxiv:1909.11522

A-Priori probability: If we randomly sample parameters θ, how likely are we to produce a particular function f?

Guillermo
Valle Perez

Model problem for a 7 bit string, study all Boolean functions f.
There are $2^7 = 128$ different strings, and $2^{128} \simeq 10^{38}$ different functions.
You might expect a 10^{-38} chance of finding any function.
Instead, we find strong simplicity bias.

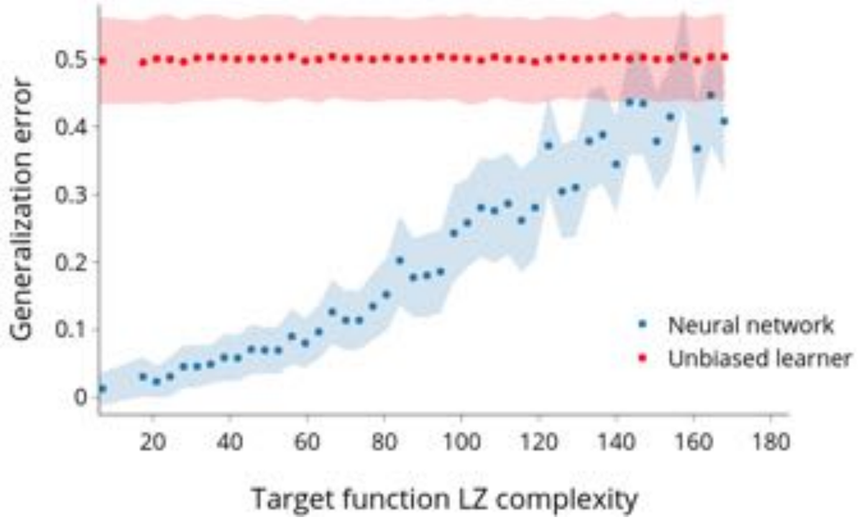




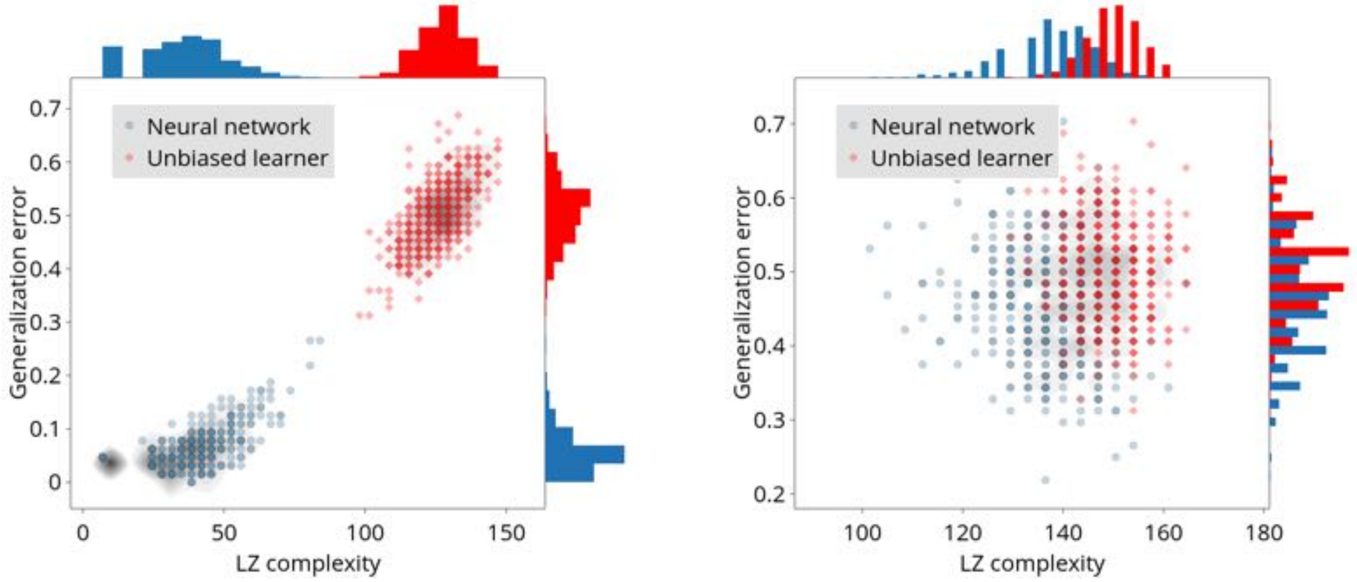
10^8 samples of parameters for (7,40,40,1) vanilla fully connected DNN system.

G. Valle Perez, C. Camargo and A.A. Louis, arxiv:1805.08522 – ICLR 2019

Does simplicity bias help generalisation?



DNN works much better than random learner

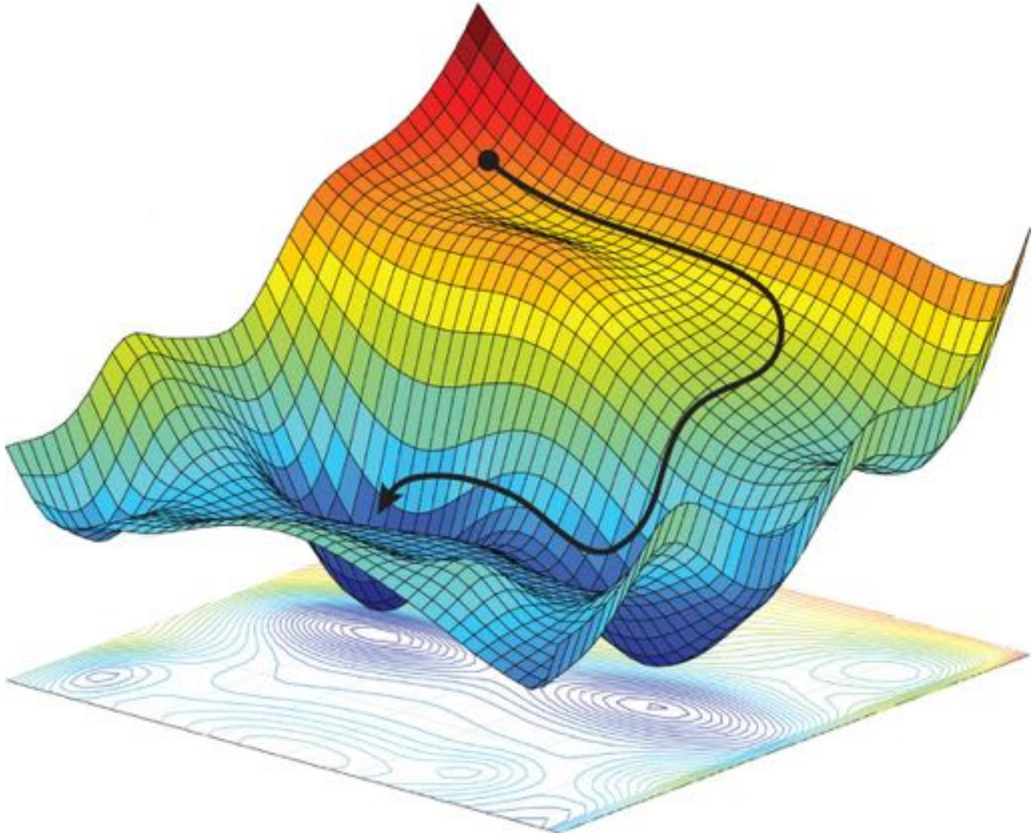


(a) Target function LZ complexity: 38.5

(b) Target function LZ complexity: 164.5

DNN works well on simple functions,
but less well on complex functions

Problem; DNNs are not trained by randomly sampling parameters



DNNs are trained using Stochastic gradient descent (SGD) on a loss function.

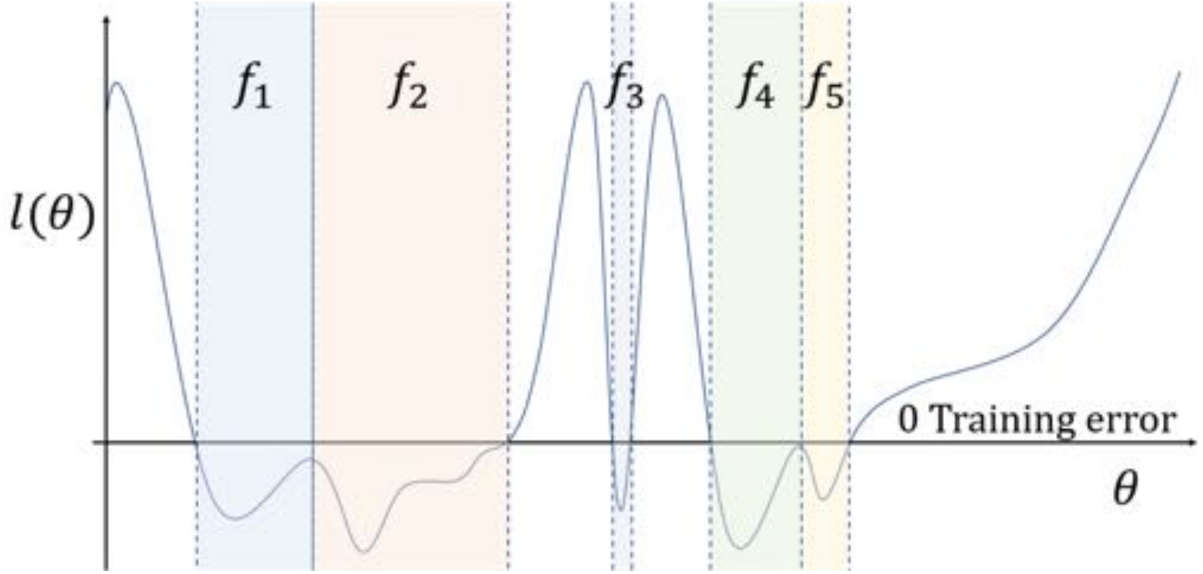
The most common view in the field:

SGD is the cause of the good generalisation.
A-priori P(f) may be irrelevant

Problem; DNNs are not trained by randomly sampling parameters

Chris
Mingard

Intuition: Basin of attraction ~ Basin size (a-priori P(f)

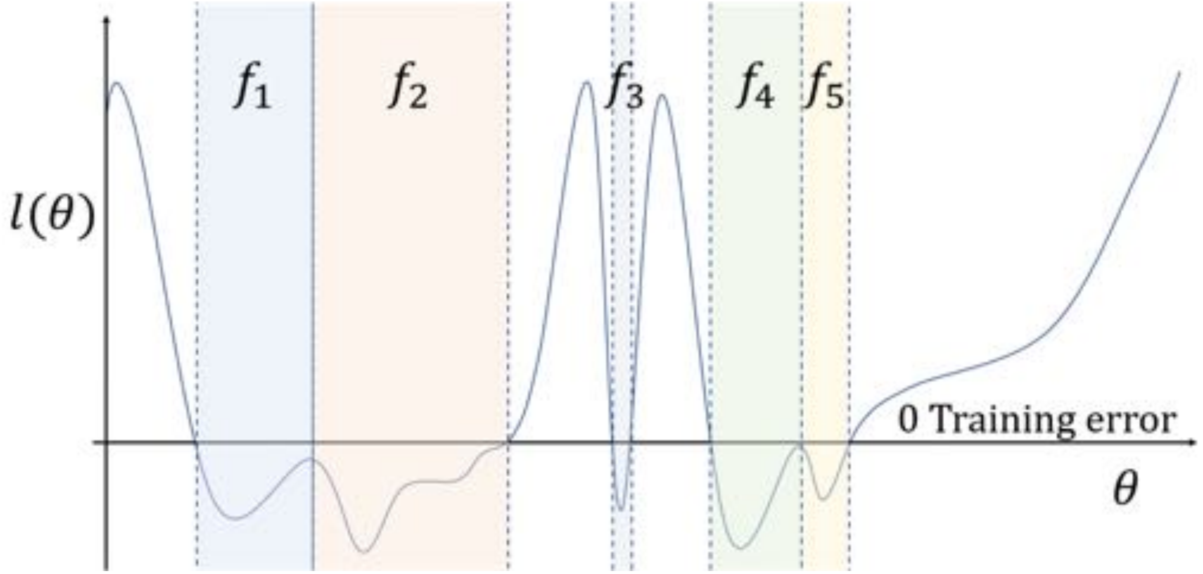


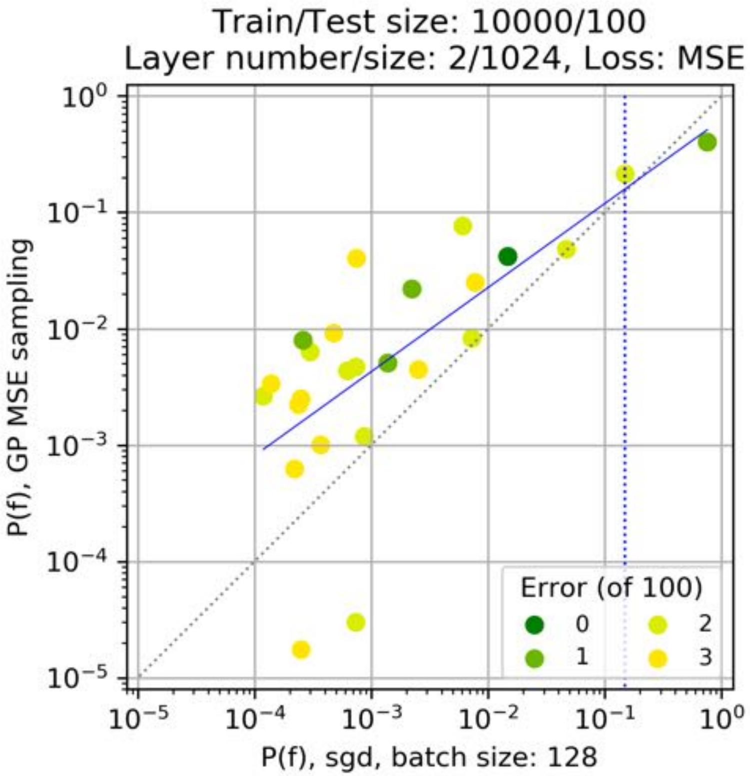
Problem; DNNs are not trained by randomly sampling parameters

Chris Mingard

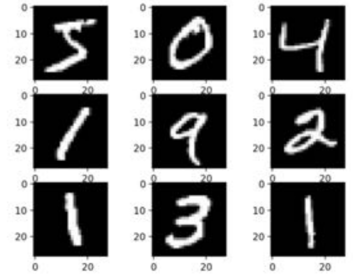
$P_{SGD}(f) \simeq P(f)$

Intuition: Basin of attraction ~ Basin size (a-priori P(f)





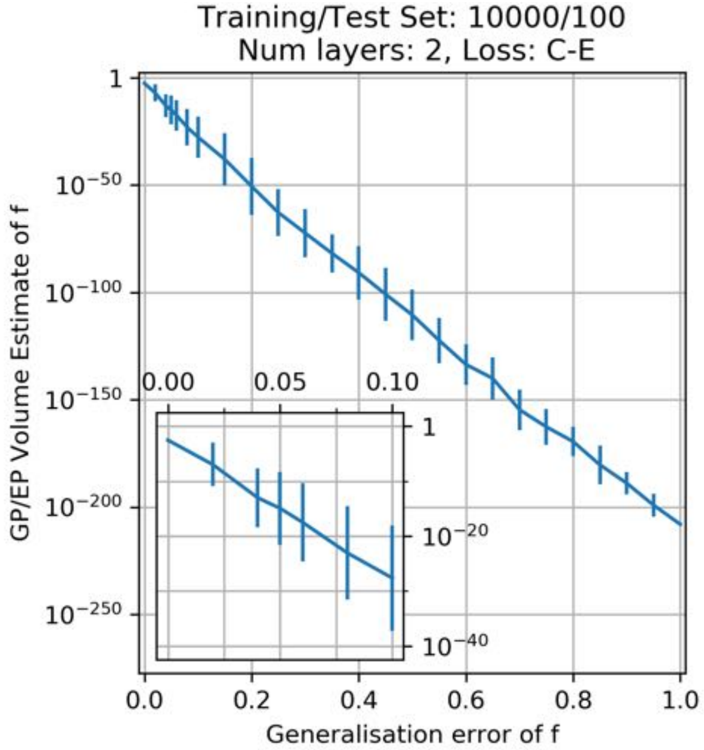
10,000 training set
100 test set on MNIST



Problem; DNNs are not trained by randomly sampling parameters

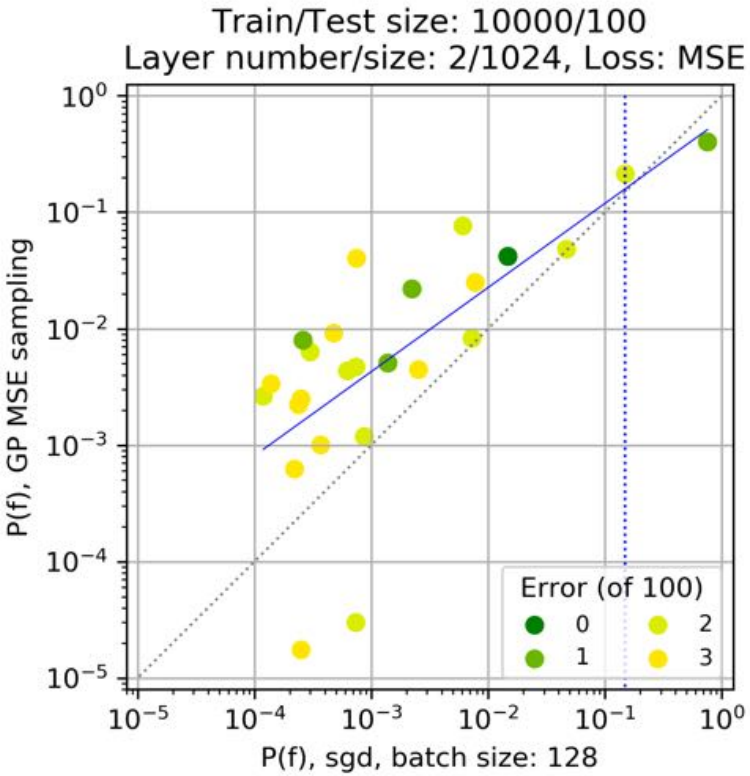
Chris Mingard

P(f) versus generalisation error

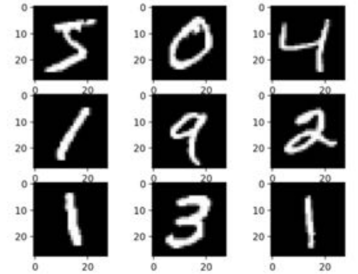


Simplicity bias in MNIST
many orders of magnitude

$P_{SGD}(f) \simeq P(f)$



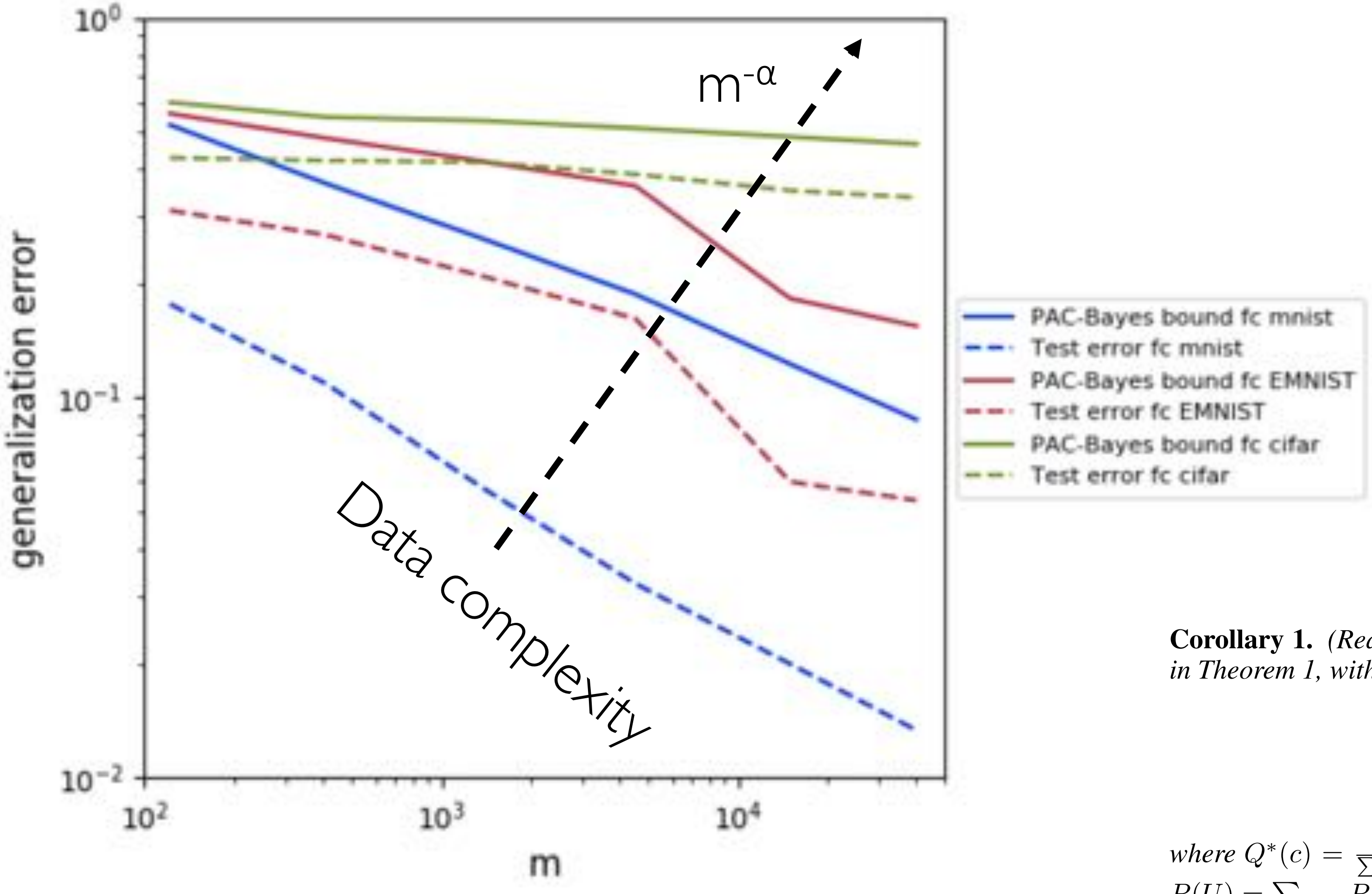
10,000 training set
100 test set on MNIST



$28^2 = 784$ dimensional space, but numbers are typically subspaces of $d \simeq$ 12-16

Scaling of error with training set size m

Guillermo
Valle Perez



Observed: error ~ $m^{-\alpha}$

1) α decreases with data complexity (bad news for machine learning)
2) α appears independent of algorithm
3) We can reproduce this scaling with PAC-Bayes theory approach we have derived.

But, WHY this scaling?

Corollary 1. *(Realizable PAC-Bayes theorem (for Bayesian classifier)) Under the same setting as in Theorem 1, with the extra assumption that $\mathcal{D}$ is realizable, we have:*

$$-\ln\left(1-\epsilon(Q^*)\right) \leq \frac{\ln\frac{1}{P(U)} + \ln\left(\frac{2m}{\delta}\right)}{m-1}$$

where $Q^*(c) = \frac{P(c)}{\sum_{c\in U} P(c)}$, U is the set of concepts in $\mathcal{H}$ consistent with the sample S, and where $P(U) = \sum_{c\in U} P(c)$

Conclusions

- Machine learning is already transforming physics, it is not just hype
- Deep learning may work because they have a natural bias towards simple functions (Occam's razon)

THANK YOU