Lecture 6: Machine Learning theory

Data Science and Information Theory, ED127 course (2025)

Florent Leclercq

www.florent-leclercq.eu

Institut d'Astrophysique de Paris
CNRS & Sorbonne Université

9 April 2025

Lake Kluane, Yukon, Canada

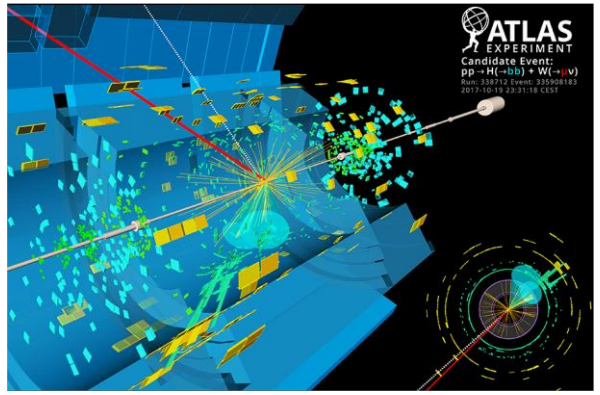
06

WHAT IS MACHINE LEARNING?

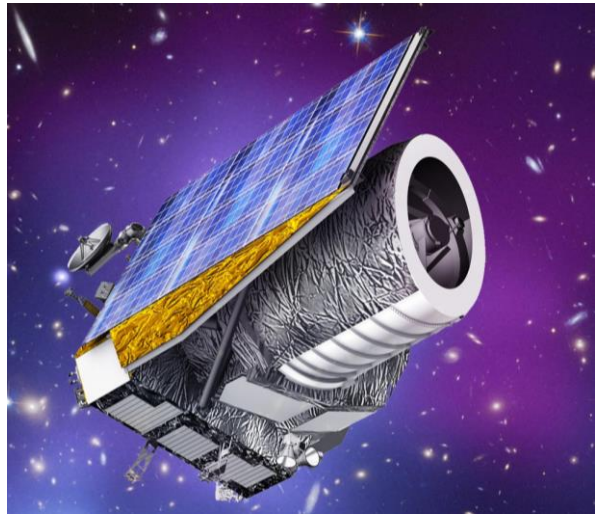
INTRODUCTION

Modern science is data-driven

- Big datasets have become the hallmark of modern science.
  - ATLAS: 3200 TB = 3.2 PB
  - Euclid: 30 PB
  - Vera Rubin Observatory (LSST): 20 TB per night
- The large-scale analysis and interpretation of complex datasets exceeds our human capabilities.
- Automating intricate and repetitive data analysis tasks enables efficient decision making (e.g., triggering systems for particle detection).



The ATLAS experiment

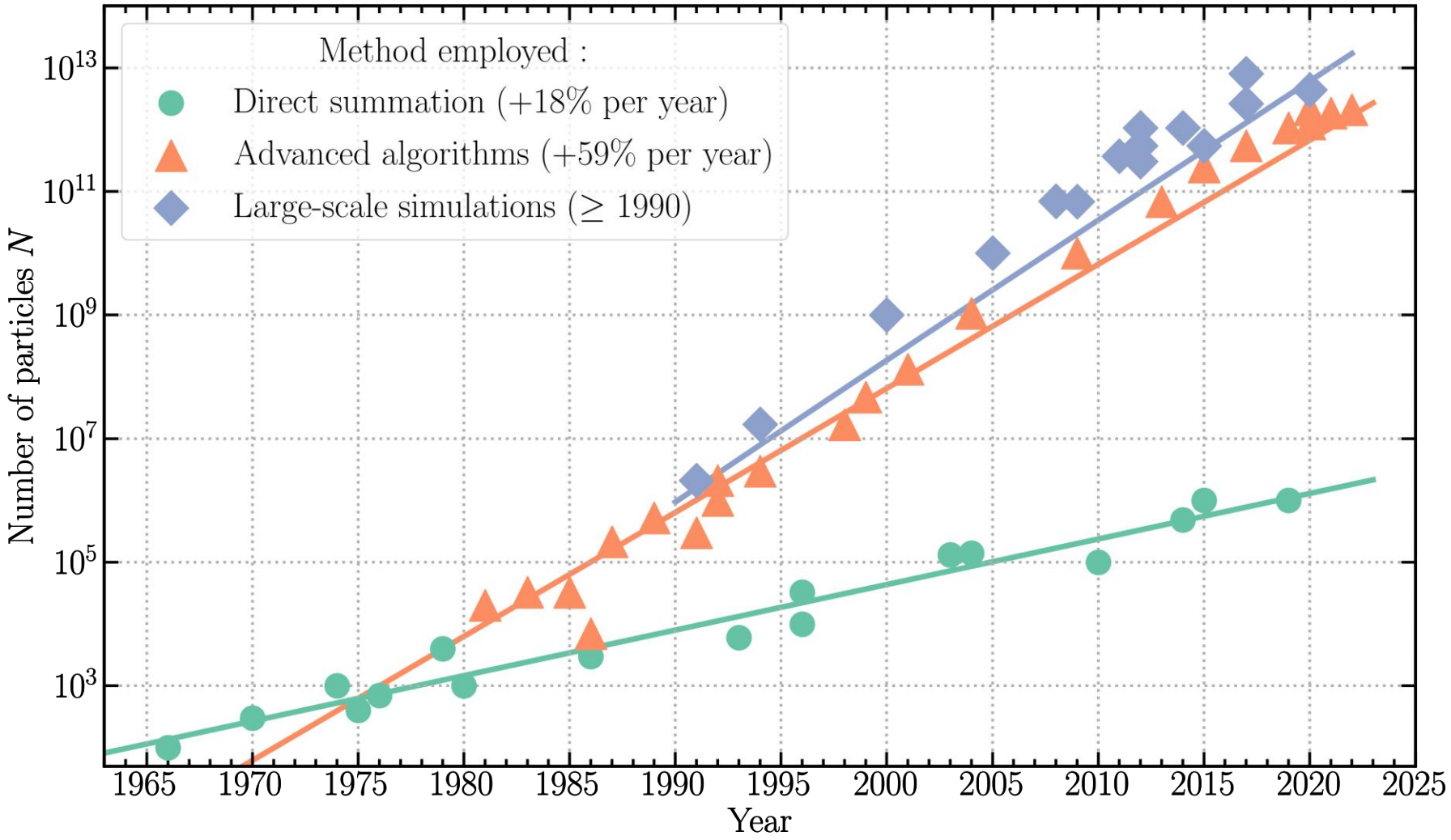


The Euclid satellite

The Vera Rubin Observatory

The growth of models

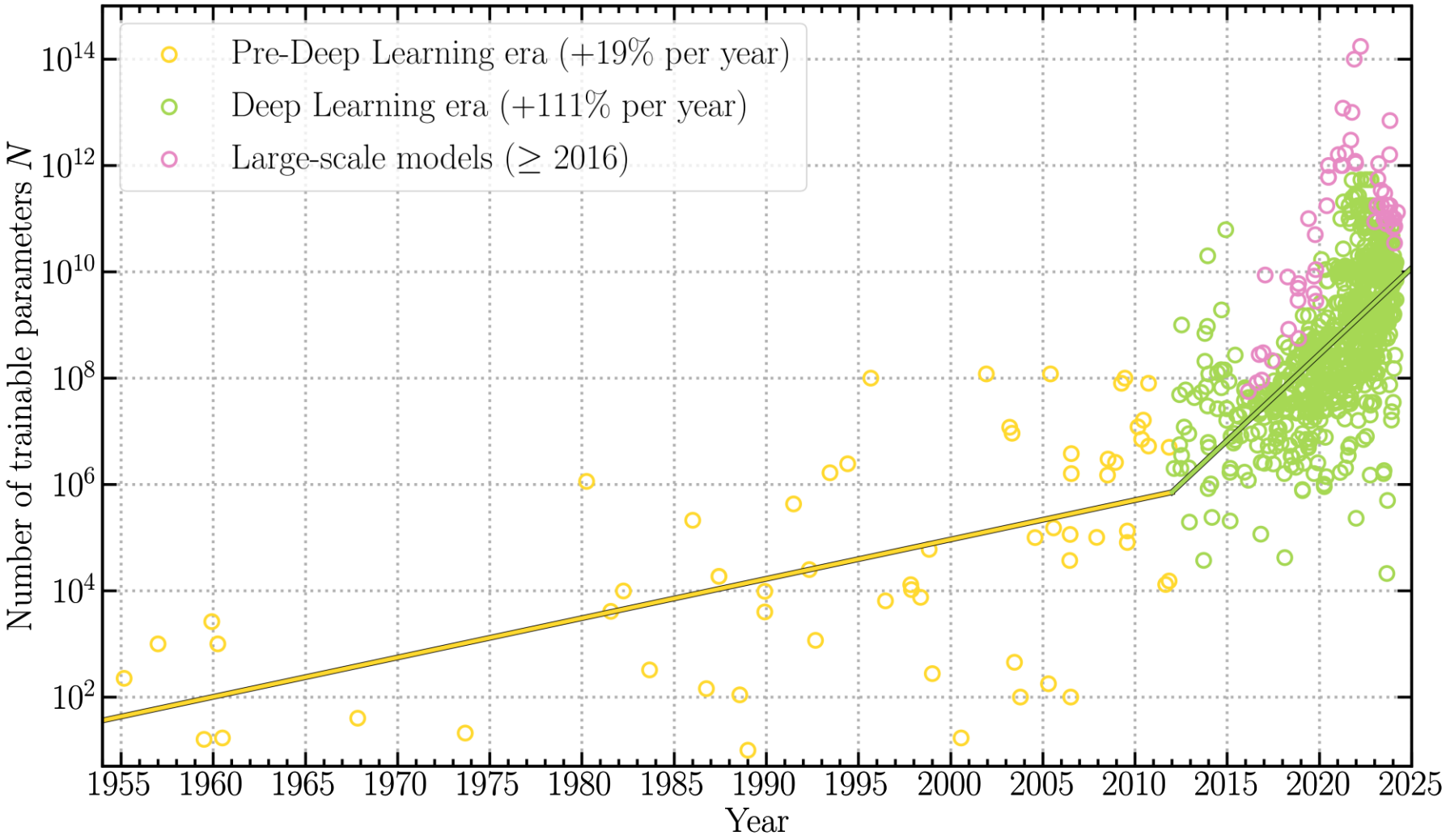
- Numerical simulations are the new way to express theoretical models.



Cosmological simulations: Github:florent-leclercq/Moore_law_cosmosims

The growth of methods

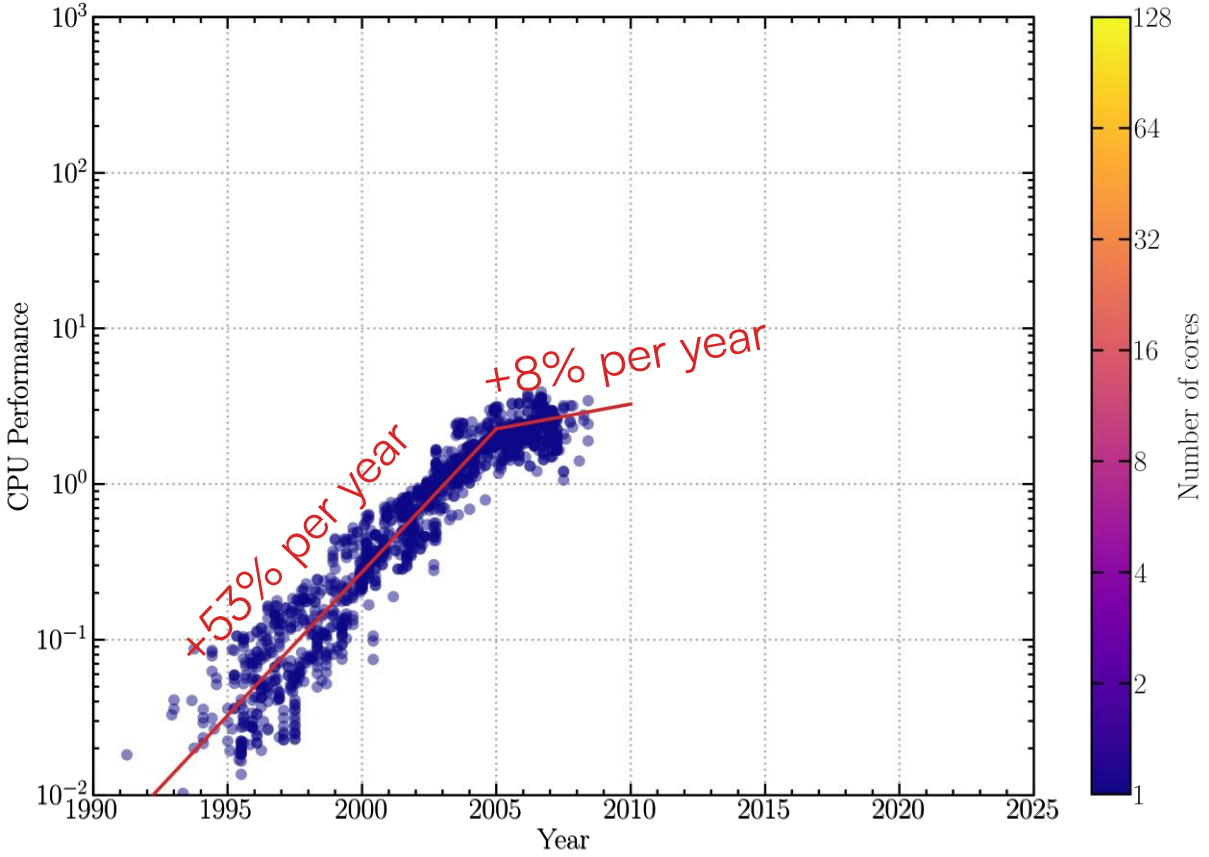
- ML methods are characterised by the number of trainable parameters.



IA models: data from epochai.org

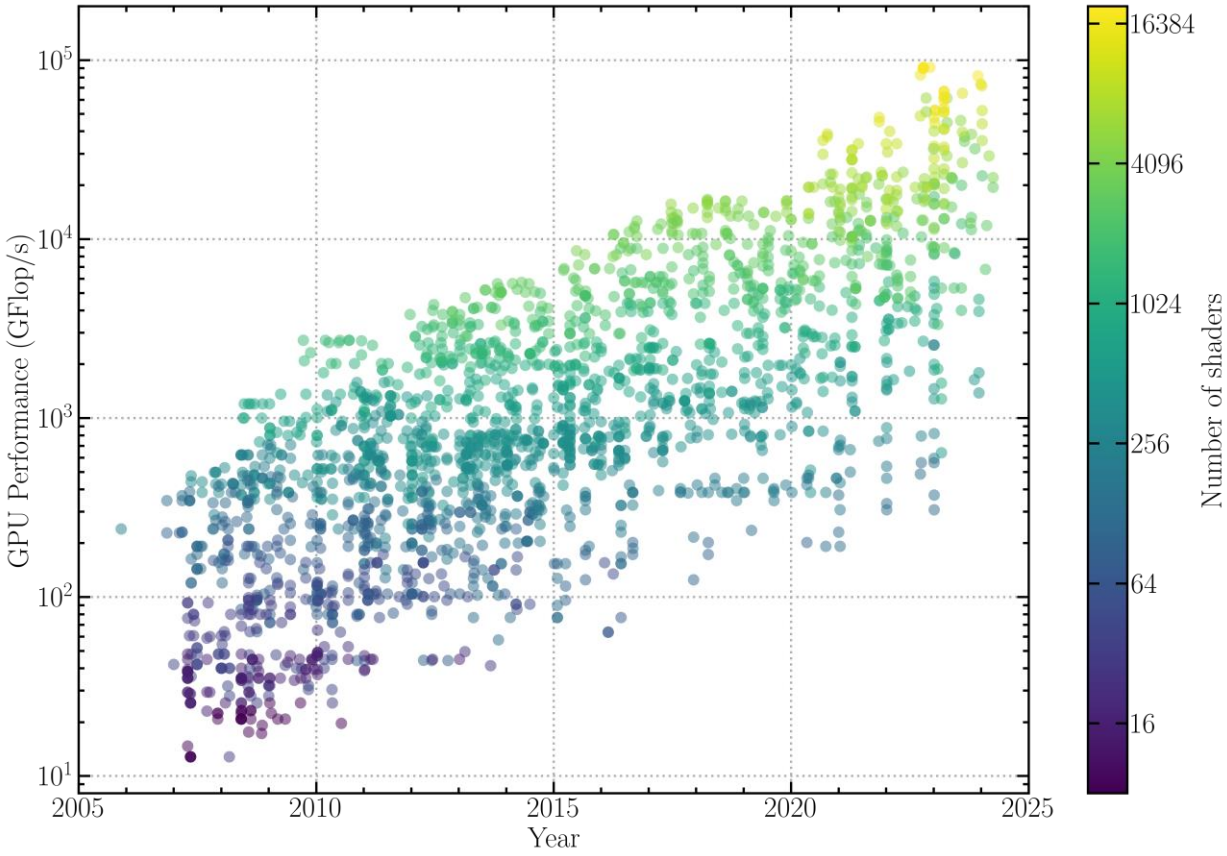
The growth of computers

- Traditional hardware architectures are reaching their physical limit: per-core compute performance is slowing down.



CPU Performance: based on adjusted SPECfp® results

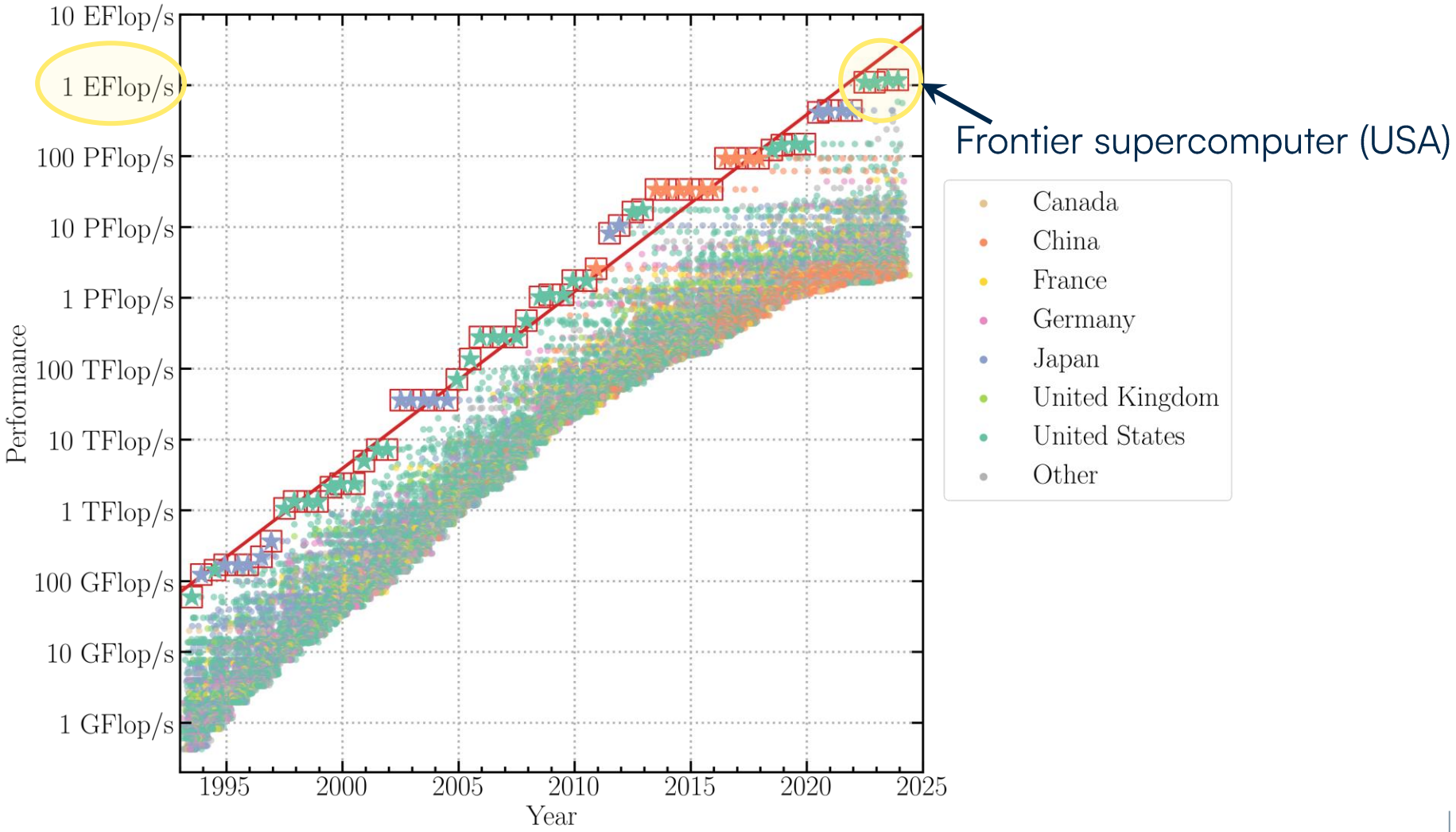
- Modern architectures are hybrid: cores + hardware accelerators: GPUs, reconfigurable or dedicated chips (FPGAs/ASICs).



GPU Performance: based on data from techpowerup.com

The growth of computers

- We have just entered the era of exascale computing.



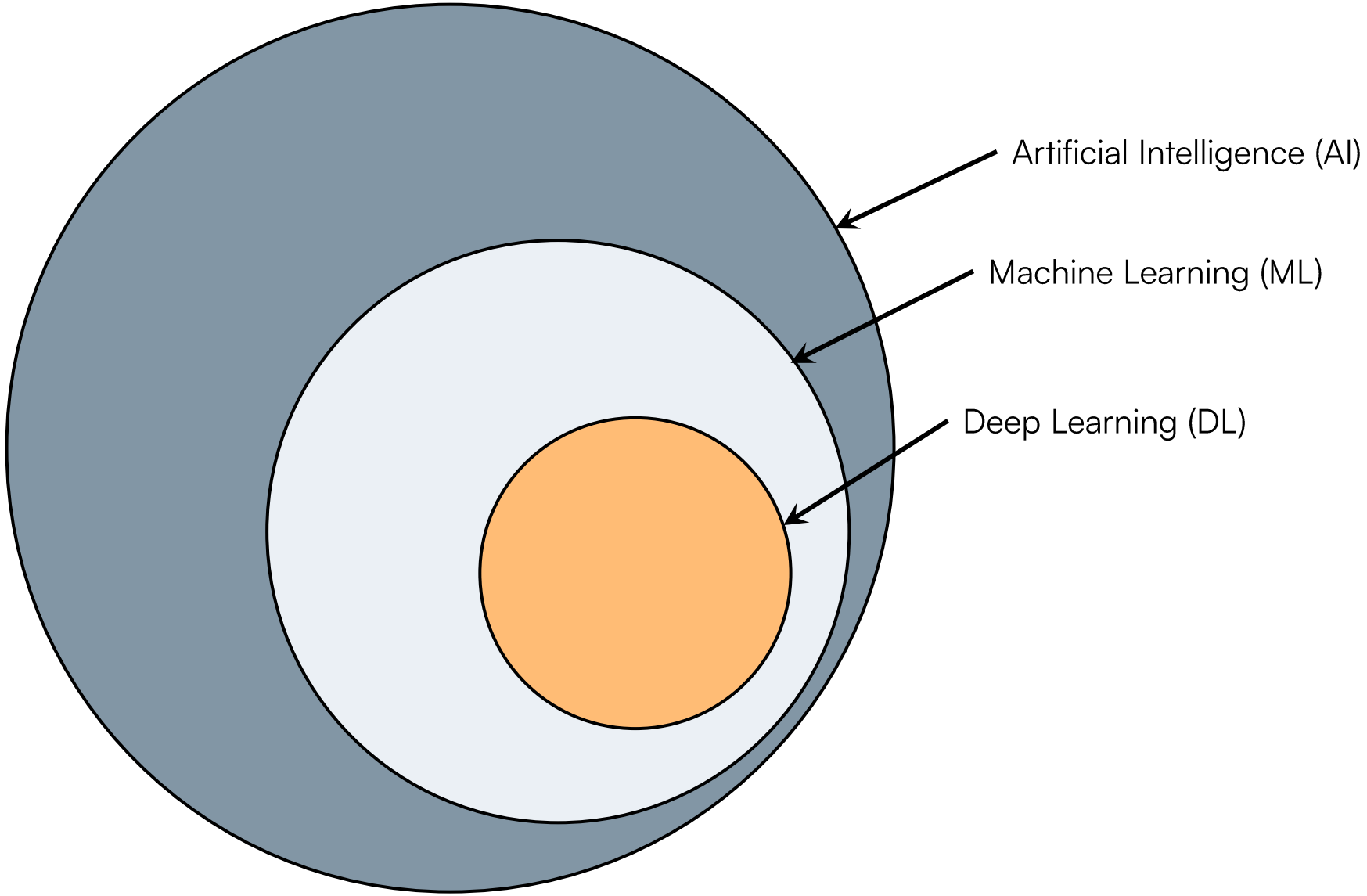
TOP 500 : data from top500.org

DIFFERENT TYPES OF MACHINE LEARNING

What does it mean for a machine to learn something?

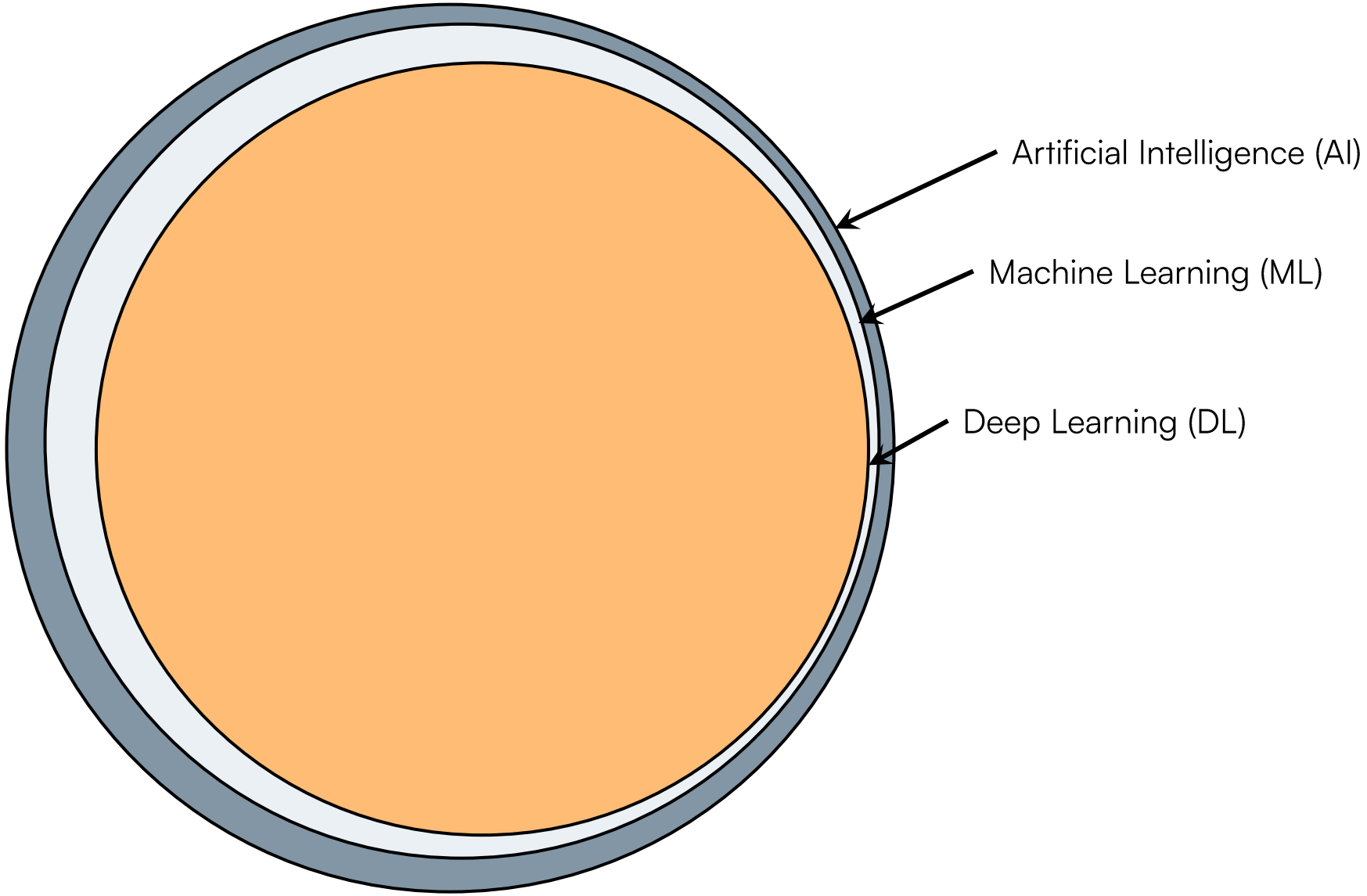
- If I download all of Wikipedia on my laptop, has it learned something? Is it smarter?
- Machine Learning (ML) is the art of programming computers so that they can learn from data.
- ML is described as the "field that gives computers the ability to learn without being explicitly programmed" (Samuel 1959).
- A program is said to learn from experience *E* regarding a set of tasks *T* and measured by a performance metric *P* if its performance on *T*, as evaluated by *P*, improves with more experience *E* (Mitchell 1997).

Artificial intelligence (AI), machine learning (ML), deep learning (DL)



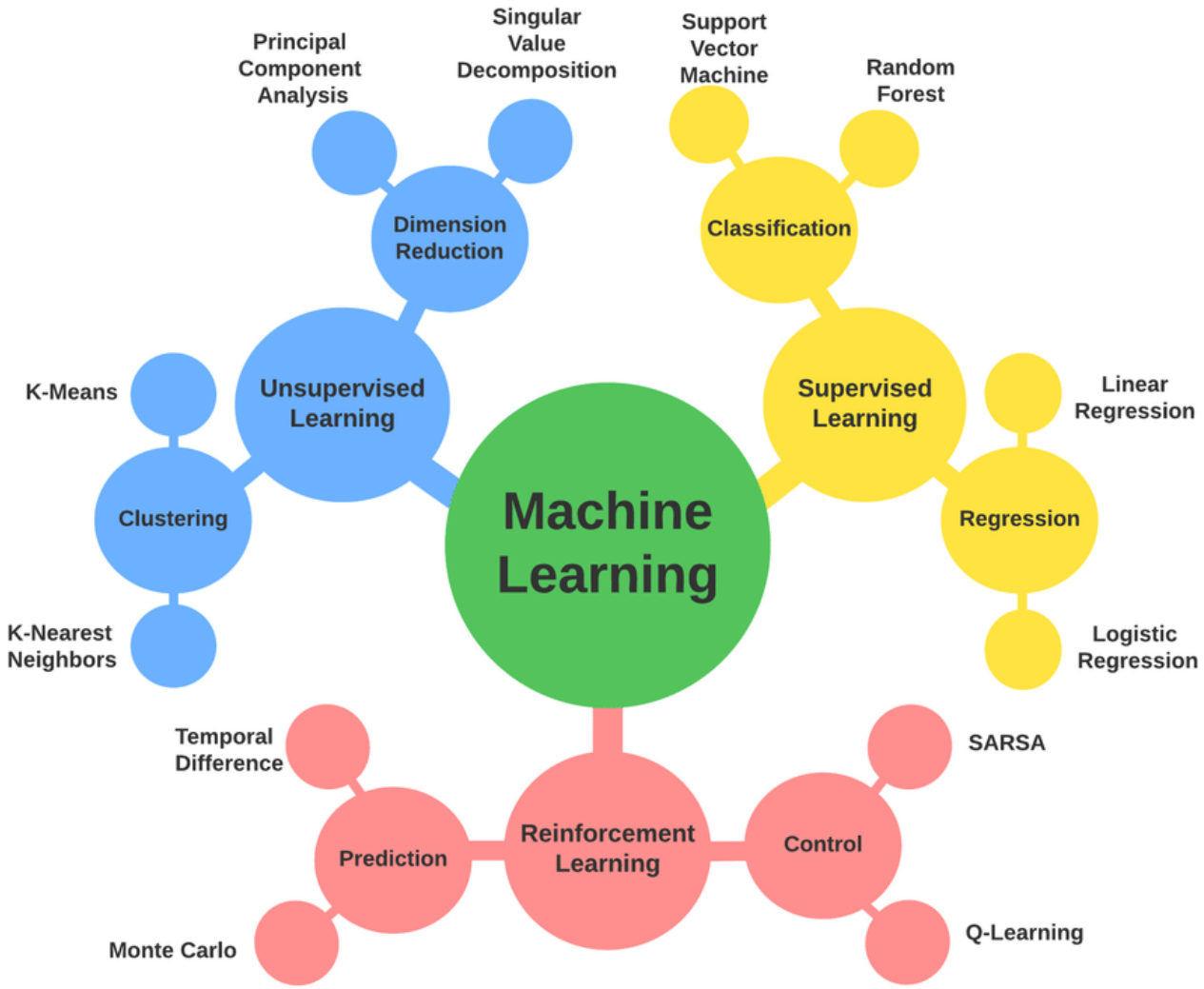
Artificial intelligence (AI), machine learning (ML), deep learning (DL)

Currently...



What are the different types of machine learning approaches?

- There exist 3 major types of ML:
  - Supervised Learning:
    Uses labelled data.
  - Unsupervised Learning:
    Uses unlabelled data
  - Reinforcement Learning:
    Requires no labelled data.
- ML methods differ by their strategy to learn model parameters (for example, weights).



Supervised machine learning example: the Titanic

- How the machine thinks: can we predict the probability of survival?

Passenger Id	Survived	Pclass	Name	Sex	Age	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	A/5 21171	7.25		S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26	STON/O2. 3101282	7.925		S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	113803	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	373450	8.05		S
6	0	3	Moran, Mr. James	male		330877	8.4583		Q
7	0	1	McCarthy, Mr. Timothy J	male	54	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2	349909	21.075		S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	347742	11.1333		S

Supervised learning tasks

- There are two types of supervised learning tasks:
  - Classification: The algorithm aims to categorise each example by selecting from two or more distinct classes. This process is known as binary classification when there are two classes to choose from. When there are more than two classes, it is referred to as multiclass classification.
    Examples:
    - Predicting galaxy morphology (e.g. spiral vs. ellipticals),
    - Detecting Higgs particles (e.g., Yes vs. No),
    - Identifying cats and dogs.
  - Regression: The algorithm produces a continuous numerical value as the output for each example.
    Examples:
    - Predicting the brightness of galaxies,
    - Forecasting the temperature at a given location,
    - Estimating the mass of a particle,
    - Inferring the slope of a linear relation (linear regression).

Unsupervised learning tasks

- Not all machine learning activities involve labelled data!
- What can you do if you only have unlabelled data?
- Unsupervised learning tasks:
  - Clustering enables you to automatically divide the dataset into groups based on similarity.
  - Anomaly detection can automatically identify unusual data points within your dataset.
  - Pattern recognition identifies and classifies patterns in unlabelled data, such as images, sounds, or text.
  - Association mining helps identify sets of items that frequently occur together in your dataset.

Unsupervised learning: clustering

- Clustering groups data by similarity. No supervisor is needed!
- All kinds of clustering algorithms exist, e.g.:
  - k-means clustering,
  - Hierarchical clustering,
  - Principal Component Analysis

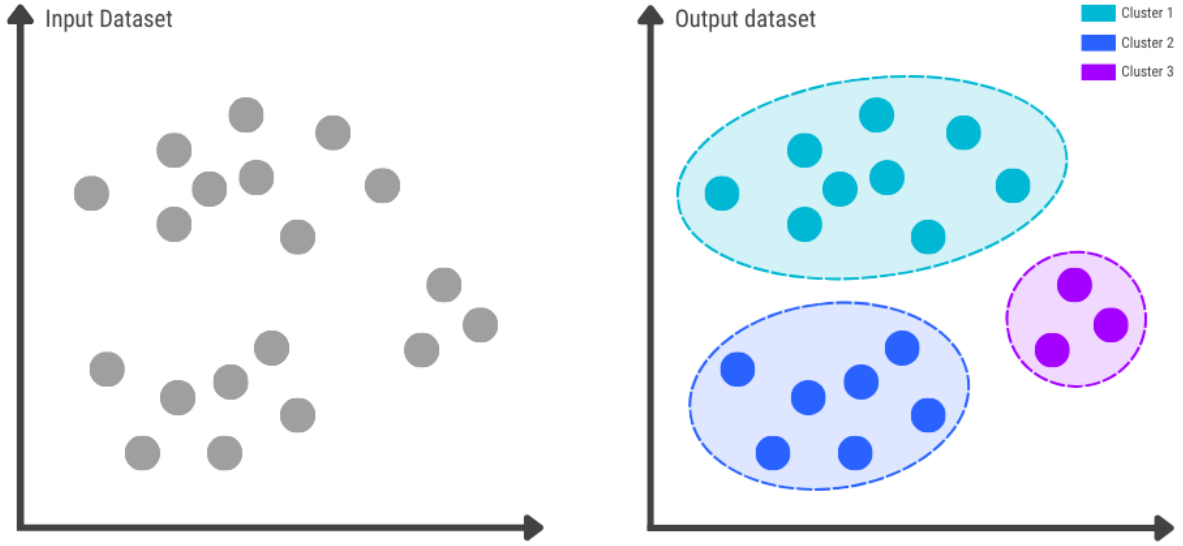


Figure: pi.exchange/blog/clustering-in-machine-learning

Reinforcement learning

- Learning system = Agent.
- The system requires no labelled training data.
- The agent takes actions and receives rewards or penalties as feedback.
- It must then learn what is the best strategy, referred to as a policy, to maximise its rewards.

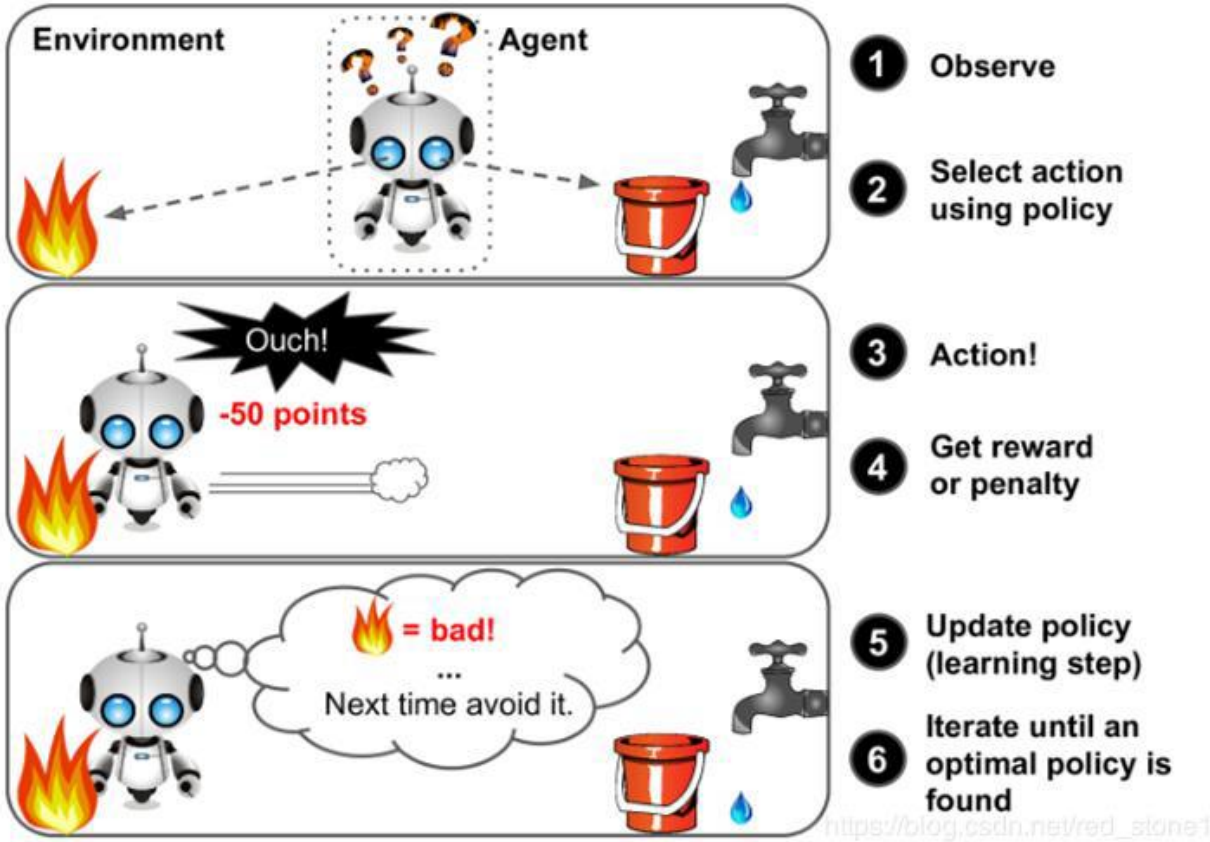


Figure: A. Géron (2017)

06

STATISTICAL LEARNING THEORY

EMPIRICAL RISK MINIMISATION AND PAC LEARNING

Insights into ML from a statistics background

- A data generation model:

$$(x, y) \curvearrowleft p(x, y)$$

features labels

> Nothing is more practical than a good theory.
> Vladimir Vapnik

- Training data are draws from the joint probability of features and labels.
- As usual, we apply the product rule to get:

$$p(x, y) = p(y|x)p(x)$$

- The aim of ML is nothing but to learn an approximation to $p(y|x)$ ("the labelling function"), which is unknown to the learner, from a set of samples ("training data").

Measuring the quality of a predictor

- We seek a function $h(x)$ for predicting y given values of the input x. But how good is $h(x)$ at predicting?
- Let's define a loss function $L(h(x), y)$, which measures the performance of $h(x)$ to predict y. For example:
  - For a classification problem, we could simply count the error rate:

$$L(h(x), y) = \begin{cases} 1 & \text{if } h(x) \neq 1 \\ 0 & \text{otherwise} \end{cases}$$

  - For a regression task with continuous y, we might use a squared loss:

$$L(h(x), y) = [h(x) - y]^2$$

- To assess the reliability of the predictor $h(x)$, we need to evaluate its average performance for any values of x and y. Thus, we need to compute the average loss function:

$$R(h) \equiv \langle L(h(x), y) \rangle_{p(x,y)} = \int L(h(x), y) p(x, y) \, \mathrm{d}x \, \mathrm{d}y$$

- We refer to this average loss function $R(h)$ as the risk of the predictor h. It represents the expected error rate of h averaged over all possible data pairs (x, y).

Finding the optimal predictor using the empirical risk

- Use the rules of probability theory:

$$R(h) = \int p(x) \int L(h(x), y) p(y|x) \, \mathrm{d}y \, \mathrm{d}x$$

- The optimal predictor function $h(x)$ minimises the risk $R(h)$, that is:

$$h_{\mathrm{opt}}(x) = \mathrm{argmin}_h \int p(x) \int L(h(x), y) p(y|x) \, \mathrm{d}y \, \mathrm{d}x$$

- Since $p(x, y)$ is unknown, we use our empirical training set of pairs $\mathcal{S} = \{(x_i, y_i)\}$ to construct a Monte Carlo density estimation of $p(x, y)$:

$$p(x, y) \approx \frac{1}{N} \sum_{i=1}^{N} \delta_{\mathrm{D}}(x - x_i) \delta_{\mathrm{D}}(y - y_i)$$

- Then we get:

$$R(h) \approx R_{\mathcal{S}}(h) = \iint \frac{1}{M} \sum_{i=1}^{N} \delta_{\mathrm{D}}(x - x_i) \delta_{\mathrm{D}}(y - y_i) L(h(x), y) \, \mathrm{d}y \, \mathrm{d}x = \frac{1}{N} \sum_{i=1}^{N} L(h(x_i), y_i)$$

This is the empirical risk function, an approximation to $R(h)$, which we seek to minimise.

Empirical risk minimisation (ERM)

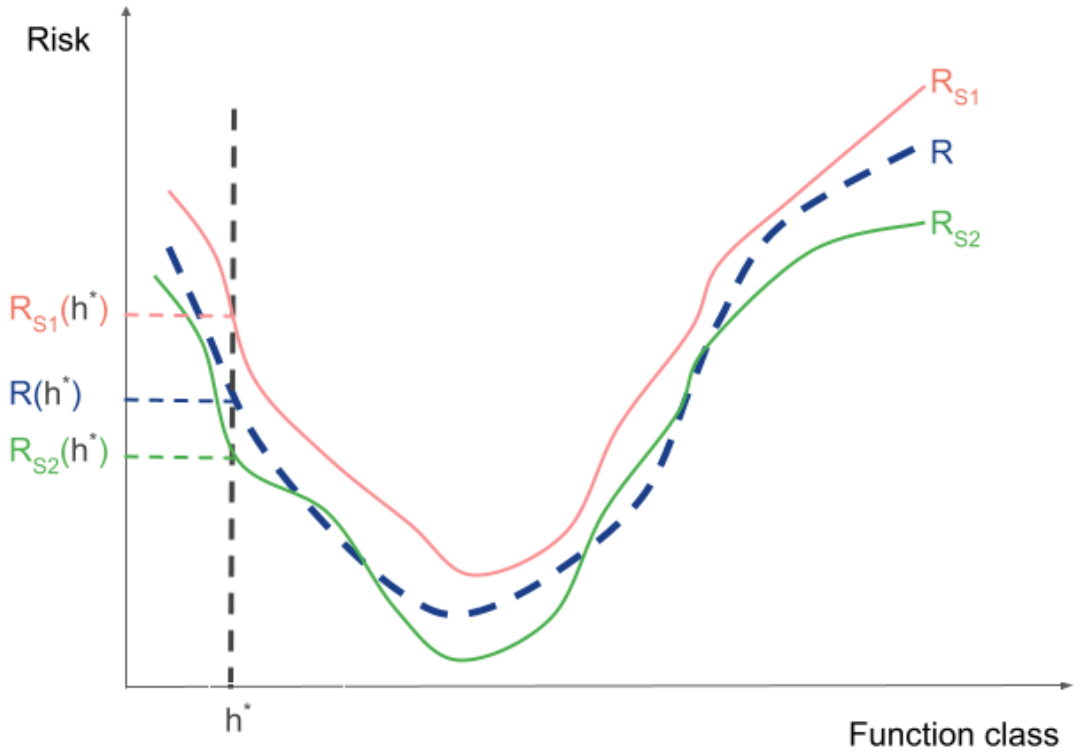
- We can now find optimal predictor(s) $h^{\mathcal{S}}_{\text{opt}}(x)$ by Empirical Risk Minimisation (ERM):

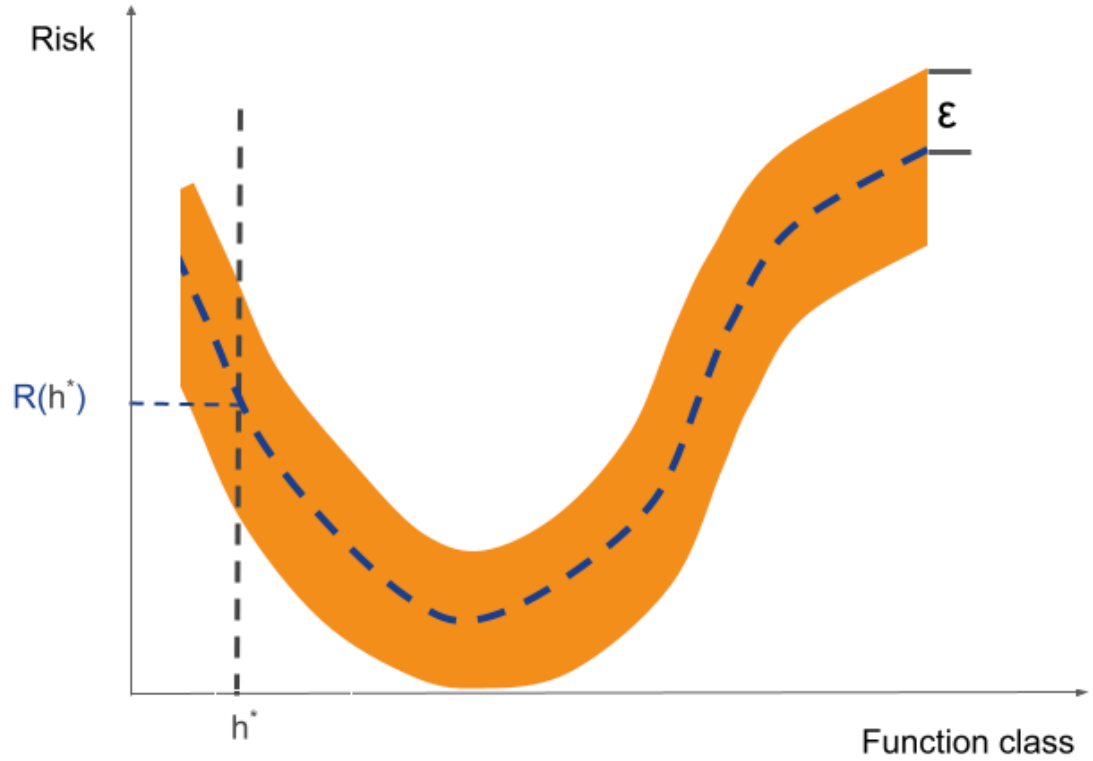
$$h^{\mathcal{S}}_{\text{opt}}(x) = \text{argmin}_h \frac{1}{N} \sum_{i=1}^{N} L(h(x_i), y_i)$$

  - This is at the core of ML.
- Any learning problem can be framed as an optimisation problem by minimising a risk function.
- The optimal function h within the space of all hypotheses $\mathcal{H}$ for the prediction task is the one that yields the lowest empirical risk.
  - Careful consideration is required when selecting the set $\mathcal{H}$ of possible hypotheses h .

Probably Approximately Correct (PAC) learning

- Is learning possible given just finite amounts of training samples $\mathcal{S}$?





- We seek to bound the approximation error of the empirical risk as: $|R(h) - R_{\mathcal{S}}(h)| \leq \epsilon$

Figures: J. Jasche

Probably Approximately Correct (PAC) learning

- The condition $|R(h) - R_{\mathcal{S}}(h)| \leq \epsilon$ is true for independent and identically distributed (i.i.d.) samples, in the large sample limit (cf. Monte Carlo approximation of integrals):

$$|R(h) - R_{\mathcal{S}}(h)| = \left| \int L(h(x), y) p(x, y) \, \mathrm{d}x \, \mathrm{d}y - \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} L(h(x_i), y_i) \right| = 0 \leq \epsilon$$

- Thus the empirical risk is an unbiased estimate of the true risk (asymptotically).
- With a finite number of training samples N, we can only make a probabilistic statement. We seek to bound the probability that the approximation error exceeds ϵ:

$$p(|R(h) - R_{\mathcal{S}}(h)| > \epsilon) < \delta$$

- The PAC criterion: a hypothesis class $\mathcal{H}$ of possible predictors is PAC learnable if there exists a finite number of required training samples N, and if N is a polynomial of $1/\epsilon$ and $1/\delta$.
- Theorem:
  - Finite hypotheses classes are PAC learnable given sufficient data!
  - But infinite hypothesis classes are not PAC learnable!

Fort, Lerasle, Moulines (2020)

WHY MACHINE LEARNING IS DIFFICULT

The no-free lunch theorem

- A typical ML workflow:
  - Given a dataset, randomly select every second data point. Split the dataset into training and validation (test) data.

  - Train on only half of the data, ideally achieving: $R_{\mathcal{S}_{\text{train}}}(h) = 0$.
  - Assuming that every function over unknown validation data is equally likely, each point has an equal probability of being labelled as 0 or 1.
  - In the best scenario, the algorithm will predict with an error rate of 0 on data it has seen but can only guess with an error rate of 0.5 on unseen data. This results in an overall error rate of 0.25.
- The no-free lunch theorem (ML version): there exists no algorithm that performs equally well on all learning tasks! (Except for the trivial but useless one that tests all models.)
- This means that there is no universal learner. Every learner has to be tailored to a specific task using prior information.

Cross-validation and bias-variance trade-off

- Cross-validation:
  - Fitting and predicting are not the same!
  - Every trained model needs to be cross-validated with test data. We define:
    - In-sample error:

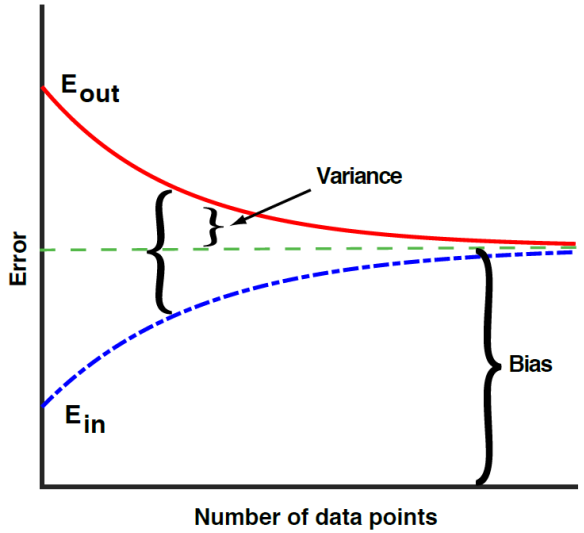
$$E_{\text{in}} = R_{\mathcal{S}_{\text{train}}}(h) = \frac{1}{N_{\text{train}}} \sum_i L(h(x_i^{\text{train}}), y_i^{\text{train}})$$

    - Out-of-sample error:

$$E_{\text{out}} = R_{\mathcal{S}_{\text{test}}}(h) = \frac{1}{N_{\text{test}}} \sum_j L(h(x_j^{\text{test}}), y_j^{\text{test}})$$

  - We have observed that: $E_{\text{out}} \geq E_{\text{in}}$
  - Clearly, the goal is to choose the model with the lowest E_{out}

- Bias-variance trade-off:
  - Suppose that the data has been generated from a sufficiently complex process that cannot be perfectly explained by our hypothesis class.
  - As the sample size increases, $E_{\text{out}} \approx E_{\text{in}}$,



  - The bias represents the best performance our model can achieve.
  - To reduce the residual bias, we need a more complex hypothesis class!

Figure: Mehta *et al.*, 1803.08823

Bias-variance trade-off, error decomposition, underfitting and overfitting

- By model complexity, we usually mean the number of model parameters w (e.g. the number of weights of a neural network). However, we only have a finite amount of training data!
- Even though more complex models reduce bias, there comes a point where the model complexity is too high to be constrained (fitted) by the data, and the variance increases!
- The prediction error can be decomposed as:

$$E_{\text{out}} = \text{Bias}^2 + \text{Variance} + \text{Noise}$$

- Typically, there is a sweet spot at intermediate model complexity that minimises E_{out}. To find this sweet spot, we compare models of different complexity and choose the one with the minimal E_{out}.
  - See also Occam's razor in Bayesian statistics!

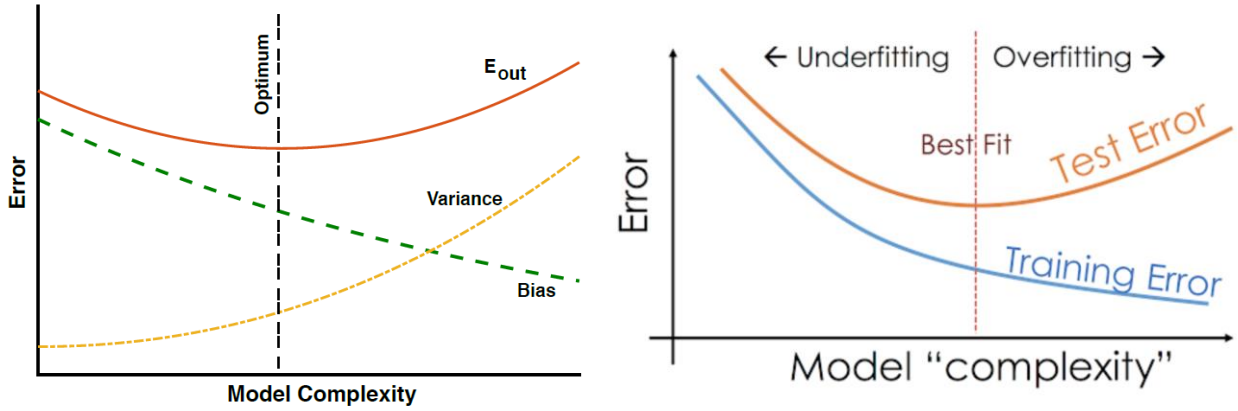


Figure: Mehta *et al.*, 1803.08823

Monitoring underfitting and overfitting

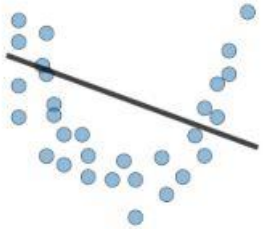
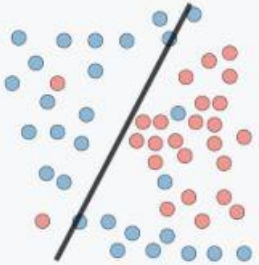
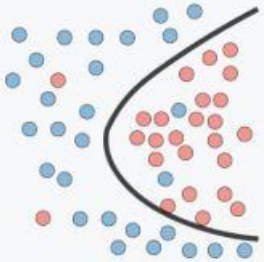
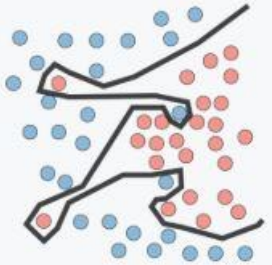
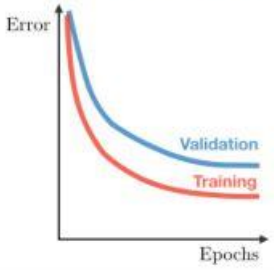
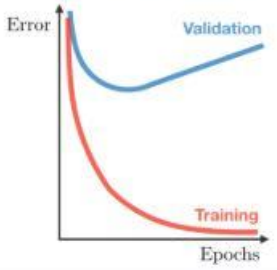
	Underfitting	Just right	Overfitting
Symptoms	• High training error • Training error close to test error • High bias	• Training error slightly lower than test error	• Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	• Complexify model • Add more features • Train longer		• Perform regularization • Get more data

Figure: kaggle.com

References and acknowledgements

References:

- Mehta *et al.* (2019), 1803.08823, *A high-bias, low-variance introduction to Machine Learning for physicists*
- G. Fort, M. Lerasle, E. Moulines (2020), *Statistique et Apprentissage*
- I. Goodfellow, Y. Bengio, A. Courville (2016), *Deep Learning*

- For his lectures, thanks to Jens Jasche

https://florent-leclercq.eu/teaching.php