Theory of Statistical Learning

Damien Garreau

Université Côte d'Azur

January 20, 2021

Outline

1. General presentation

2. Introduction to Statistical Learning
   - General introduction
   - First concepts
   - Empirical risk minimization
   - Overfitting
   - PAC learning
   - Uniform convergence

1. General presentation

Who am I?

- maître de conférence (= assistant professor) in LJAD (Laboratoire Jean Dieudonné)
- before that: postdoctoral researcher (Max Planck Institute, Tübingen, Germany)
- even before: PhD in Inria Paris
- teaching ($\approx$ 200 hours per year)
- **Rest of the time?** research!
- **Goal:** think about open problems whose solution could benefit society, solve them, publish papers with the answer
- examples of topics that interest me at the moment:
  - interpretability of machine learning algorithms
  - statistical tools for the study of deep neural networks

Who are you?

- online teaching is suboptimal :(
- speaking to a black screen is weird
- **Please introduce yourselves!** (with camera on if possible)
- I will call your name then you can briefly introduce yourself

Goal of the course

- **Goal i):** understand the *maths* behind the algorithms that you learn
- this can save a lot of time!
- one often works with limited resources
- **Goal ii):** learn about theoretical guarantees on existing algorithms
- a way to be reassured: under some assumptions, my method works
- see more clearly the *limitations* of the methods: if some assumption is not satisfied, we can prove that it will fail

Organization of the course

- all the information, documents $\rightarrow$ Slack
- (provisional) calendar:
  1. January 20, (today), 9am-12am
  2. January 27, 9am-12am
  3. February 3, 9am-12am
  4. February 10, 9am-12am
  5. February 17, 9am-12am
  6. February 24, 9am-12am (midterm)
  7. March 10, 9am-12am
  8. March 17, 9am-12am
  9. March 24, 9am-12am
  10. March 31, 9am-12am (exam)
- **Disclaimer:** midterm and exam may be online depending on the situation in the coming weeks
- final grade $=$ (midterm $+$ final)$/2$

Requirements

- **Elementary real analysis:** functions of a real variable, usual functions, continuity, Lipschitz continuity
- **Calculus:** derivative, partial derivatives, gradient, Taylor series
- **Basic probability theory:** measurable space, probability measure, random variable, expectation, conditional expectation, probability density function, cumulative density functions
- **Limit theorems:** law of large numbers, central limit theorem
- **Linear algebra:** vector space, matrix, norms, diagonalization of a matrix, singular value decomposition

If you feel like you are not up to date on one of these points, write me and I will point you towards some good books.

Useful resources

- **Main reference:** Shalev-Schwartz, Ben-David, *Understanding Machine Learning: from Theory to Algorithms*, Cambridge University Press, 2014
- **Also a good read:** Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, 2001 (second edition: 2009)
- **Wikipedia:** as good as ever.
- **Wolfram alpha:** if you have computations to make and you do not know want to use a proper language: `https://www.wolframalpha.com/`
- **Google scholar:** use it!

2. Introduction to Statistical Learning

2.1. General introduction

The goal of statistical learning

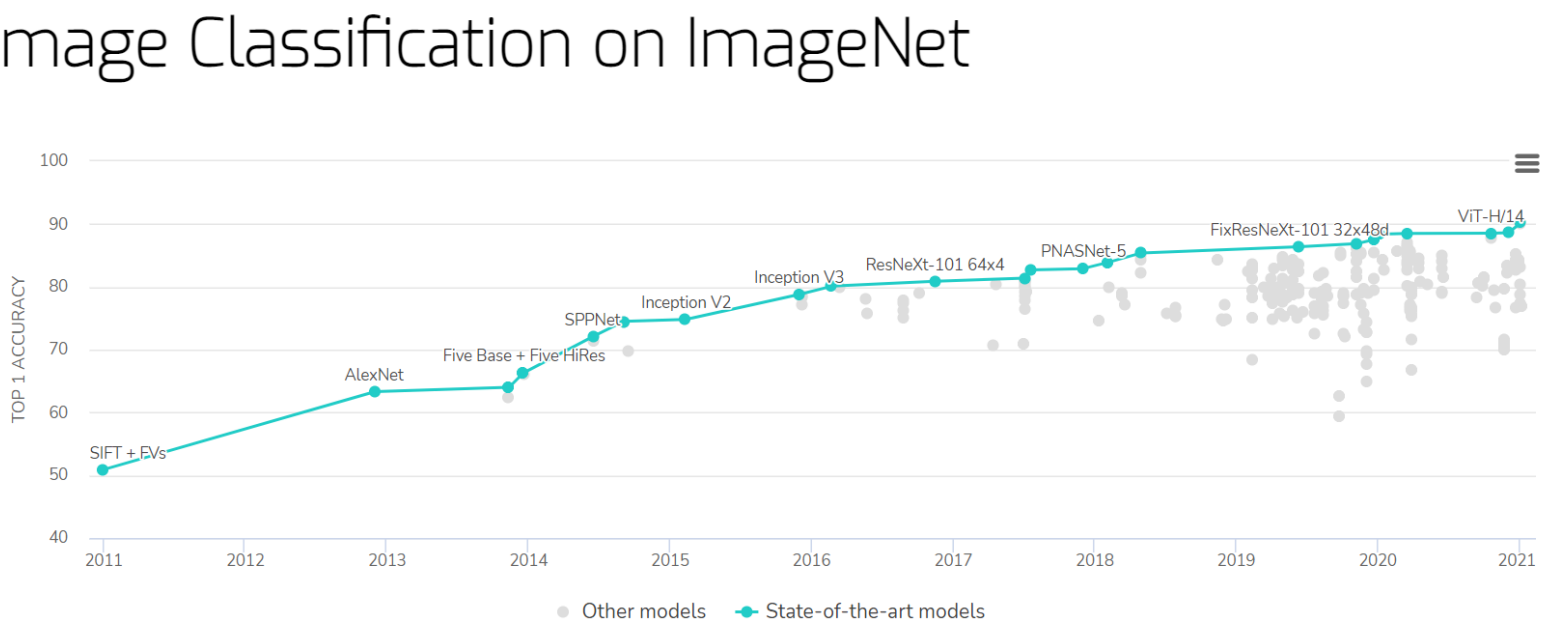
- **Fundamental example:** image classification
- **Goal:** given any image x, we want to predict which object / animal y is in the image

$\longmapsto$ "lion"

- **Main idea:** instead of defining the function f ourselves, we are going to *learn it* from data
- **Why?** no clear definition of a "lion"
- **Motivation:** industry (advertisement), healthcare (automated patient triage), military (automated defense systems)

How is this even possible?

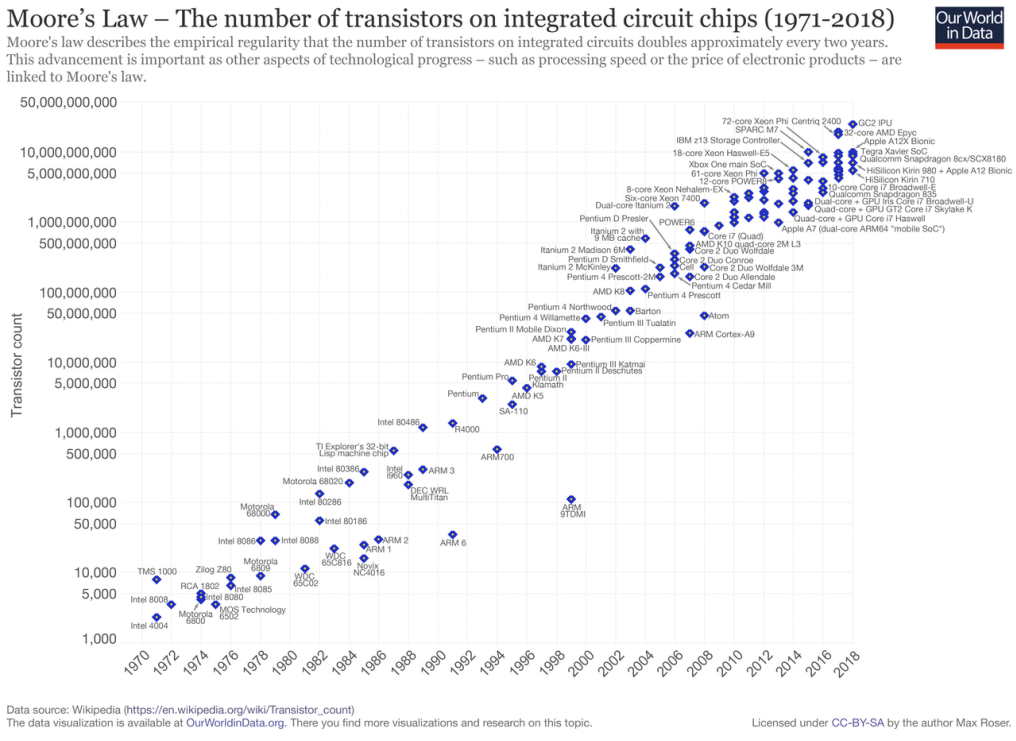
- four ingredients made statistical learning a viable paradigm:
- **Ingredient (i):** data to feed to the models
- previous example from ImageNet[1]: roughly 1 million images for training (150GB of data)



[1] Deng et al., *ImageNet: A large hierarchical image database*, CVPR, 2009

How is this even possible?

- **Ingredient (ii):** computing power
- we have the processing power to deal with these data

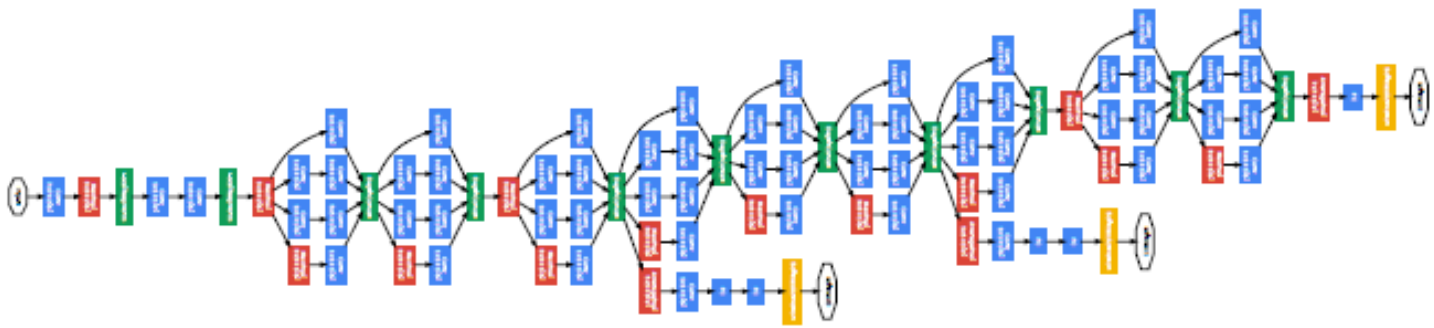


Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Roser.

How is this even possible?

- **Ingredient (iii):** models that are complex enough
- sate-of-the-art today: (deep) neural networks (originating from much earlier research[2])
- Inception:[3] 24M parameters, GTP-3:[4] 175B



[2] Rosenblatt, *The perceptron, a perceiving and recognizing automaton*, tech report, 1957

[3] Szegedy et al., *Going deeper with convolutions*, CVPR, 2015

[4] Brown et al., *Language Models are Few-Shot Learners*, tech report, 2020

How is this even possible

SWITCH TRANSFORMERS: SCALING TO TRILLION PARAMETER MODELS WITH SIMPLE AND EFFICIENT SPARSITY

William Fedus*
Google Brain
liamfedus@google.com

Barret Zoph*
Google Brain
barretzoph@google.com

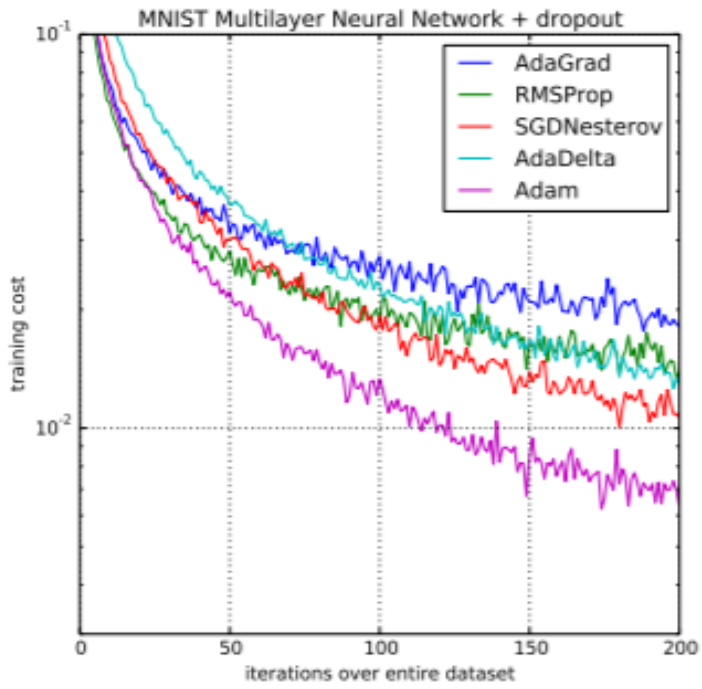
Noam Shazeer
Google Brain
noam@google.com

ABSTRACT

In deep learning, models typically reuse the same parameters for all inputs. Mixture of Experts (MoE) models defy this and instead select *different* parameters for each incoming example. The result is a sparsely-activated model – with an outrageous number of parameters – but a constant computational cost. However, despite several notable successes of MoE, widespread adoption has been hindered by complexity, communication costs, and training instability. We address these with the Switch Transformer. We simplify the MoE routing algorithm and design intuitive improved models with reduced communication and computational costs. Our proposed training techniques mitigate the instabilities, and we show large sparse models may be trained, for the first time, with lower precision (bfloat16) formats. We design models based off T5-Base and T5-Large (Raffel et al., 2019)

How is this even possible?

- **Ingredient (iv):** efficient algorithms to train the models
- gradient descent on steroids[5]
- efficient gradient computations[6]



[5] Kingma, Ba, *ADAM: A method for stochastic optimization*, ICLR, 2015

[6] Rumelhart et al., *Learning representations by back-propagating errors*, Nature, 1986

2.2. First concepts

Input space

- **Input space:** measurable space $\mathcal{X}$ containing all the objects that we want to label
- also called *domain*, or *domain set*
- elements $x \in \mathcal{X}$ are usually described as vectors
- coordinates of the vector $=$ *features*
- **Example:** ImageNet images: RGB images $\rightarrow$ 3 8-bits channels

$$\in [\![0, 255]\!]^{299 \times 299 \times 3}$$

- $\mathcal{X}$ can be very high-dimensional in modern applications (here $299 \times 299 \times 3 = 268,203$)

Labels and training data

- **Label set:** labels belong to a set $\mathcal{Y}$
- **Example:** $\mathcal{Y}$ is the set of names of object and animals of the dataset

```
{0: 'tench, Tinca tinca',
 1: 'goldfish, Carassius auratus',
 2: 'great white shark, white shark, man-eater, man-eating shark, Carcharodon carcharias',
 3: 'tiger shark, Galeocerdo cuvieri',
 4: 'hammerhead, hammerhead shark',
 5: 'electric ray, crampfish, numbfish, torpedo',
 6: 'stingray',
 7: 'cock',
 8: 'hen',
 9: 'ostrich, Struthio camelus',
```

- we restrict ourselves to $\mathcal{Y} = \{0, 1\}$ for the time being, but can be much larger in modern applications ($1,000$ for ImageNet)
- **Training data:** $S = ((x_1, y_1), \ldots, (x_n, y_n))$ *finite* sequence of points of $\mathcal{X} \times \mathcal{Y}$
- also called *training set*

Hypothesis class

- **Hypothesis:** $h : \mathcal{X} \to \mathcal{Y}$ a prediction rule. also called *predictor*, *classifier* (in the context of classification)
- we are looking for a good h
- **Hypothesis class:** $\mathcal{H}$ some space of functions. if no restrictions, set of all measurable functions
- **Example:** linear classifiers:

$$\mathcal{H} = \{h : \text{sign}(x) \mapsto w^\top x + b, w \in \mathbb{R}^d, b \in \mathbb{R}\},$$

  where $w^\top x$ denotes the scalar product between w and x
- given an algorithm A and a dataset S, we will write $h = A(S)$ the output of our algorithm on S

Data generation

- **Data generation:** for now, we assume that there is a true distribution $\mathcal{D}$ of the data on $\mathcal{X}$
- the training examples are i.i.d. samples from $\mathcal{D}$
- i.i.d.: *independent identically distributed*
- **Example:** sample images uniformly at random from a larger set (all the images on the internet)
- hard to satisfy: there is always a bias in the way your dataset is constructed

Assumption (noiseless setting): there exists a function $f : \mathcal{X} \to \mathcal{Y}$ such that $y = f(x)$ for any $x \in \mathcal{X}$.

- **Important:** we know neither $\mathcal{D}$ nor f! we only have access to S

Measure of success

- **Risk of a classifier:** probability that h does not return the correct label on a (new) random sample:

$$\mathcal{R}_{\mathcal{D},f}(h) = \mathbb{P}_{x\sim\mathcal{D}}\left(h(x) \neq f(x)\right) .$$

- **Intuition:** we want to be good, *on average*, for new samples of the same distribution
- subscript often omitted when clear from context
- also called *generalization error*, *true error* (notation L or $\mathcal{E}$)
- **Important:** we want to find h with small generalization error. Ideally,

$$\mathcal{R}_{\mathcal{D},f}(h) = 0 .$$

- **Question:** how to do this?

2.3. Empirical risk minimization

Empirical risk minimization

- as we have seen, what we would *like* to do is find

$$h \in \arg\min_{h \in \mathcal{H}} \mathcal{R}_{\mathcal{D},f}(h) = \arg\min_{h \in \mathcal{H}} \mathbb{P}_{x \sim \mathcal{D}} \left(h(x) \neq f(x) \right) .$$

- **Problem:** we know neither $\mathcal{D}$ nor f...
- ...and even if we did it would still be a very difficult problem (there are *a lot* of measurable functions!)
- **Idea:** replace $\mathcal{R}_{\mathcal{D},f}$ by an *empirical* version
- empirical risk (or training error):

$$\hat{\mathcal{R}}_S(h) = \frac{1}{n} \left| \{ i \in \{1, \ldots, n\} \text{ s.t. } h(x_i) \neq y_i \} \right| ,$$

  where $|E|$ denotes the cardinality of (finite) set E
- minimizing the empirical risk = empirical risk minimization[7] (ERM)

[7] Vapnik, *Principles of risk minimization for learning theory*, NIPS, 1992

Exercise

Exercise: set $h \in \mathcal{H}$. Let n be a fixed integer.

1. Show that
$$\mathbb{E}_S\left[\hat{\mathcal{R}}_S(h)\right] = \mathcal{R}_{\mathcal{D},f}(h)\,,$$
where the expectation is taken with respect to all i.i.d. draws of S.
2. Show that $\hat{\mathcal{R}}_S(h) \xrightarrow{\mathbb{P}} \mathcal{R}_{\mathcal{D},f}(h)$ when $n \to +\infty$.

Solution

1. First, we see that

$$|\{i \in \{1, \dots, n\} \text{ s.t. } h(x_i) \neq y_i\}| = \sum_{i=1}^{n} \mathbb{1}_{h(x_i) \neq y_i} .$$

Then we write

$$\begin{aligned}
\mathbb{E}\left[\hat{\mathcal{R}}_S(h)\right] &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{h(x_i) \neq y_i}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\mathbb{1}_{h(x_i) \neq y_i}\right] && \text{(linearity)} \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{P}\left(h(x_i) \neq y_i\right) && (\mathbb{E}\left[\mathbb{1}_A\right] = \mathbb{P}\left(A\right)) \\
&= \frac{1}{n}\sum_{i=1}^{n} \mathbb{P}\left(h(x_i) \neq f(x_i)\right) && \text{(noiseless assumption)}
\end{aligned}$$

Solution, ctd.

Further, since the x_i are i.i.d., for any $1 \leq i \leq n$,

$$\mathbb{P}\left(h(x_i) \neq f(x_i)\right) = \mathbb{P}\left(h(x) \neq f(x)\right) .$$

We recognize the definition of the true risk. Therefore,

$$\begin{aligned} \mathbb{E}\left[\hat{\mathcal{R}}_S(h)\right] &= \frac{1}{n}\sum_{i=1}^{n} \mathcal{R}_{\mathcal{D},f}(h) \\ &= \mathcal{R}_{\mathcal{D},f}(h) \qquad \text{(does not depend on } i\text{)} \end{aligned}$$

2. Since the x_i are i.i.d. random variables, so are the Z_i defined by

$$Z_i = \mathbb{1}_{h(x_i)\neq f(x_i)} .$$

Solution, ctd.

Moreover, the Z_is are bounded almost surely (by 1). In particular, they are integrable. There fore, we can use the law of large numbers and write

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{h(x_i)\neq y_i} \xrightarrow{\mathbb{P}} \mathbb{E}\left[\mathbb{1}_{h(x_1)\neq f(x_1)}\right] .$$

From question 1., we deduce that

$$\hat{\mathcal{R}}_S(h) \xrightarrow{\mathbb{P}} \mathcal{R}_{\mathcal{D},f}(h) .$$

□

Reminder: the law of large numbers

Theorem (Weak Law of Large Numbers = WLLN): Let $Z_1, Z_2, \ldots$ be a sequence of i.i.d. random variables. Assume that $\mathbb{E}[|Z_1|] < +\infty$ and set $\mu := \mathbb{E}[Z_1]$. Then

$$\frac{Z_1 + \cdots + Z_n}{n} \xrightarrow{\mathbb{P}} \mu\,.$$

- **Intuition:** average of measurements converges towards the true value
- stronger statement is true, *strong* law of large numbers, with almost sure convergence instead of in probability
- multivariate extension: coordinate-wise

Law of large numbers, in pictures

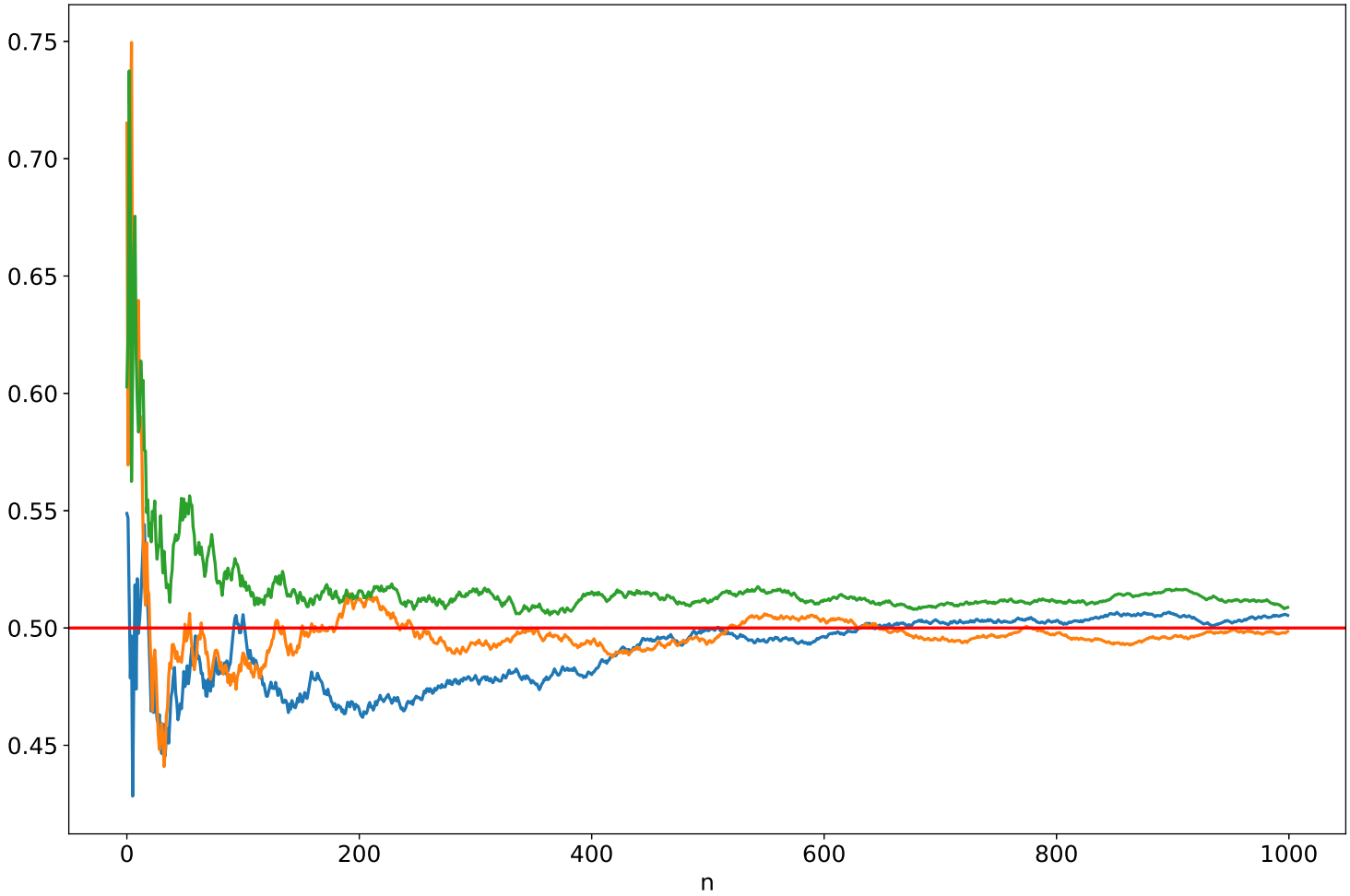


Figure: trajectories of the empirical mean for i.i.d. $\mathcal{B}(1/2)$

2.4. Overfitting

Overfitting

- **Problem:** is the hypotheses class $\mathcal{H}$ is too large, then we can bring the empirical risk to *zero*
- easy when $\mathcal{H}$ is the set of all measurable functions:

$$h(x) = \begin{cases} y_i & \text{if } \exists i \in \{1, \dots, n\} \text{ s.t. } x = x_i \\ 0 & \text{otherwise.} \end{cases}$$

- in particular,

$$\forall 1 \leq i \leq n, \quad h(x_i) = y_i \,.$$

- in that case,

$$\hat{\mathcal{R}}_S(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{h(x_i) \neq y_i} = 0 \,.$$

- our predictor has *memorized* the examples, but cannot *generalize*