(Deep) Learning Theory.

Instructor: Andrea Agazzi
andrea.agazzi@unipi.it

Motivation: neural networks have emerged as a powerful tool to solve complex problems, and many of the successes of the ongoing AI revolution would be unthinkable without such tool.

Examples include:
- Deepfakes
- AlphaGO
- Snapchat filters
- Tiktok / Chatbots

But also:
- Medical diagnosis (data)
- Self-driving cars / robotics
- Advertising (data)
- Translation
- Autopilots

List of links to videos / tutorials available on class website.

$\Rightarrow$ aim to develop a mathematical theory explaining this success.

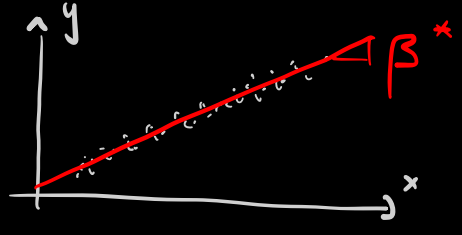
Mathematical preliminaries: statistical learning theory

Example (linear regression) assume we have measured the price of texas crude oil and the price of diesel, giving rise to a database

$$\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^{n}$$

(x_i: crude \$, y_i: diesel \$)

We can plot the dataset and plot the linear regression



Then, for a new day, measuring the price of crude x_{new} we can predict

$$y_{new} \approx \beta^* x_{new}$$

Question: what can I say about my prediction? is it going to be accurate?

We now introduce a setting where this question can be formulated

Classical Supervised Learning problem:

Given: a dataset/training set $\mathcal{D}_n$ (P) that we aim to "learn" (the world). $(x_i, y_i) \in X \times Y$ $i \in \{1, \dots, n\}$

(P: Peter *)

where :

- $x_i \in \mathcal{X}$ are the inputs / features / covariates / dependent variables / predictors
  - what you have
  - examples: - cat pics - text - car comands - code - music
- $y_i \in \mathcal{Y}$ are the outputs / labels / responses / independent variables / outcomes
  - what you predict

throughout this class we will assume that

$$(x_i, y_i) \overset{iid}{\sim} P^*(dx, dy) \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y})$$

P^* unknown !

and disintegrate $P(dx, dy) = P_{y|x}(x, dy) \cdot P_x(dx)$

Note : We distinguish between : • $\mathcal{Y} \subseteq \mathbb{R}^{d'}$ → regression

• $\mathcal{Y} \subseteq \mathbb{Z}$ → classification

Example (simple linear regression) : $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}$

P_x unknown, $P_{y|x} = \mathcal{N}(\beta^* x, 1)$

in other words $y_i = \beta^* x_i + \varepsilon_i$ $\quad \varepsilon \overset{iid}{\sim} \mathcal{N}(0,1)$

Example (Classification) : $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, 1\}$

P_x unknown, $P_{y|x}(x,y) = (1 - \pi(x))^{\frac{1-y}{2}} \pi(x)^{\frac{1+y}{2}}$ $\quad \pi : \mathcal{X} \to [0,1]$

$\pi(x) = P(y = 1 \mid x)$

Decision theory (what the engineer decides).

Find $f: X \to Y$ such that for a new $(x_{new}, y_{new}) \sim P$

$$f(x_{new}) \approx y_{new} \Longrightarrow f(x) \text{ approximates well } y?$$

To perform / evaluate a prediction, we have to fix:

- a measure of performance (loss function).

$$\ell : Y \times Y \longrightarrow \mathbb{R}$$

returning the cost $\ell(y, y')$ of predicting $y' \in Y$ when the label is $y \in Y$. This induces

The expected / population Risk (generalization error) — expected loss

$$R(f) = \int_{X \times Y} \ell(y_{new}, f(x_{new}))\, p(dx_{new}, dy_{new})$$

a proxy for this quantity is the empirical risk (training error)

$$\hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i))$$

Example (regression): $\ell(y, y') = \|y' - y\|^2$ (MSE).

$$\Longrightarrow \quad \mathcal{R}(f) = \int_{X\times Y} \|f(x) - y\|^2 P(dx, dy)$$

$$\hat{\mathcal{R}}_n(f) = \frac{1}{n}\sum \|f(x_i) - y_i\|^2 \qquad (x_i, y_i) \overset{iid}{\sim} P$$

Example (linear regression): for a given $f_\beta : x \mapsto \beta \cdot x$ we have:

$$\mathcal{R}(\beta) = \int_{\mathbb{R}^2} (\beta \cdot x - y)^2 \cdot e^{-\frac{(y-\beta^* x)^2}{2}} \cdot \frac{1}{\sqrt{2\pi}} \cdot P_x(dx)\, dy$$

exercise ⟶ $= \int [(\beta - \beta^*)x]^2 P_x(dx) + 1$

Example (classification, 0-1) $\ell(y, y') = \mathbb{1}(y \neq y')$ $\quad \mathbb{1}(yy' \leq 0)$

$$\mathcal{R}(f) = \int \mathbb{1}(f(x) \neq y)\, P(dx\, dy) = \mathbb{P}(f(X) \neq y).$$

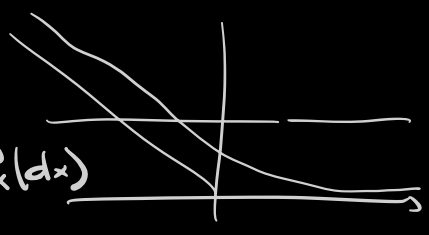
Example (classification, logistic) $\ell(y, y') = \log(1 + \exp(-yy'))$

$$\mathcal{R}(f) = \sum_{y=-1,1} \int \log\left(1 + \exp(-y f(x))\right) \underbrace{P_y(x, y)}_{\text{Bernoulli}} P_x(dx)$$

note that sometimes in classification tasks we take $f : X \to \mathbb{R}$ and predict with $\operatorname{sign} f(x)$. (the larger $\|f(x)\|$ the more "sure")

Exercise: setting $\hat{p}(x, 1) = \dfrac{1}{1 + e^{-f(x)}}$ then

$$\mathcal{R}(f) = \int \sum_y P_{y|x}(x, y) \log\left(\hat{p}(x, y)\right) P_x(dx)$$



- a model or hypothesis class

$$\mathcal{F} \subseteq \{f : \mathcal{X} \to \mathcal{Y}\}$$

the class of functions where we search for the "optimal" approximator.

Usually we work with parametric models, for a param. space Θ:

$$\mathcal{F}_\Theta = \{f_\vartheta : \mathcal{X} \to \mathcal{Y}, \ \vartheta \in \Theta\}.$$

→ classes of functions indexed by a parameter ϑ.

Example (linear regression) $\Theta = \mathbb{R}^d$, $\mathcal{F}_\Theta = \{x \mapsto \beta \cdot x, \ \beta \in \Theta\}$

Example (logistic regression) $\Theta = \mathbb{R}^d$, $\mathcal{F}_\Theta = \{(1 + e^{-\beta \cdot x}), \ \beta \in \Theta\}$

Example (neural networks): (next class).

- an optimization algorithm.

$$\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \longrightarrow \mathcal{F} \qquad \mathcal{D}_n(P) \mapsto \mathcal{A}(\mathcal{D}_n)$$

often, such algorithms aim to minimize $\hat{R}$ on $\mathcal{F}_\Theta$

This strategy is called Empirical Risk Minimization (ERM).

Example (linear regression):

Writing $\hat{R}(\beta) = \frac{1}{n} \sum_{j=1}^{n} (\beta \cdot x_j - y_j)^2 =$

Notation:

$X = \begin{pmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & & \vdots \\ x_{n,1} & \cdots & x_{n,d} \end{pmatrix} \in \mathbb{R}^{n \times d}$ $\quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n$

$$= \frac{1}{n} \| X \cdot \vec{\beta} - \vec{y} \|_2^2$$

we find a minimizer by setting

$$D_\beta \hat{R}(\beta) = \frac{1}{n} X^T (X \cdot \vec{\beta} - \vec{y}) = 0$$

$$\Longrightarrow \mathrm{ft}((X, \vec{y})) = (X^T X)^{-1} X^T \vec{y}$$

.

Summary: for a given $P, \mathcal{C}, \mathcal{F}_0, \mathrm{ft}$ we study

$$\Delta(\mathcal{D}_n) = \mathcal{R}(\mathrm{ft}(\mathcal{D}_n)) - \inf_{f: X \to Y} \mathcal{R}(f)$$

for instance: $\mathbb{E}_{\mathcal{D}_n}(\Delta(\mathcal{D}_n)) \xrightarrow{n \to \infty} 0$?

$\Delta(\mathcal{D}_n) \xrightarrow[n \to \infty]{P} 0$?

(...)

Risk decomposition

Often, studying $\Delta(\mathcal{D}_n)$ is too hard (many moving parts) so we decompose it into a sum of terms:

$$\underbrace{R(f_{\hat\vartheta}) - \inf_{f: X\to Y} R(f)}_{\text{population risk}} = \underbrace{R(f_{\hat\vartheta}) - \inf_{\vartheta\in\Theta} R(f_\vartheta)}_{\text{estimation err.}} + \underbrace{\inf_{\vartheta\in\Theta} R(f_\vartheta) - \inf_{f: X\to Y} R(f)}_{\text{approximation err.}}$$

ANALYSIS ($\mathcal{F}$)

typically $|\Theta|\to\infty$, $n\to\infty$ $\to 0$ — typically $|\Theta|\to\infty \to 0$

we can further decompose the estimation error as:

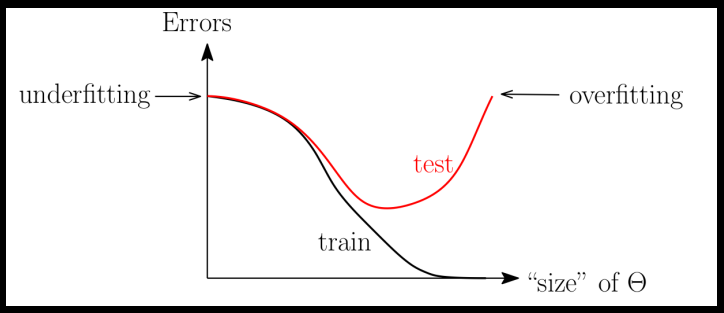
$$\underbrace{R(f_{\hat\vartheta}) - R(f_{\vartheta'})}_{\text{estimation}} = \underbrace{R(f_{\hat\vartheta}) - \hat R(f_{\hat\vartheta})}_{\text{gen / concentr.}} + \underbrace{\hat R(f_{\hat\vartheta}) - \hat R(f_{\vartheta'})}_{\text{empirical optimiz}} + \underbrace{\hat R(f_{\vartheta'}) - R(f_{\vartheta'})}_{\text{gen}}$$

STA ($\mathcal{D}_n$) OPT ($f_t + \mathcal{F}$)

→ each part of this error "competes" a different element and bound

Time permitting. $y_i = \vartheta_* \cdot x_i + \varepsilon_i$ $\quad \mathbb{E}(\varepsilon_i) = 0$ $\quad \mathrm{Var}(\varepsilon_i) = \sigma^2$

$$\Rightarrow R(\vartheta) = \mathbb{E}\left(\|X\vartheta_* + \varepsilon - X\vartheta\|^2\right) = \sigma^2 + \frac{1}{n}(\vartheta-\vartheta_*)X^TX(\vartheta-\vartheta^$$



$$\Rightarrow \mathbb{E}(R(\vartheta)) - R^* = \mathbb{E}\left(\|\hat\vartheta - \mathbb{E}(\hat\vartheta) - \vartheta\|_\Sigma^2\right)$$

$$= \underbrace{\mathbb{E}\left(\|\hat\vartheta - \mathbb{E}(\hat\vartheta)\|_\Sigma^2\right)}_{\text{Variance}} + 0 + \underbrace{\|\mathbb{E}(\hat\vartheta) - \vartheta\|_\Sigma^2}_{\text{Bias}}$$

Program: - General, finite width neural networks

unsatisfactory
- Definition
- Expressivity
- Training
- Generalization bounds when $n \to \infty$

- ∞-width NN 2 regimes: NTK, MF

more satisf. but still not done!
- NTK: • linear behavior of training
  • generalization analysis
- MF: • nonlinear training behavior → optimality

- Differences between two regimes.

- Extensions to RL, RNNs, generative

Logistics: - website contains up-to-date info on class
- notes published on website
- exam: presentation for those who come to class, else oral exam.
- THU class: alternating (not this week).
- discuss open problems.