PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL
Préparée à l'Université Paris-Dauphine

On the impact of randomization on robustness in machine learning

Soutenue par
Rafaël PINOT
Le 02/12/2020

Ecole doctorale n° ED 543
Ecole doctorale SDOSE

Spécialité
Informatique

Composition du jury :

Francis BACH Directeur de recherche, INRIA / ENS-PSL	*Président du jury*
Stéphane CANU Professeur, INSA ROUEN	*Rapporteur*
Panayotis MERTIKOPOULOS Chargé de recherche, CNRS	*Rapporteur*
Sébastien BUBECK Senior researcher, MICROSOFT	*Examinateur*
Cordelia SCHMID Directrice de recherche, INRIA / GOOGLE	*Examinatrice*
Michèle SEBAG Directrice de recherche, CNRS	*Examinatrice*
Jamal ATIF Professeur, PARIS DAUPHINE-PSL	*Directeur de thèse*
Cédric GOUY-PAILLER Ingénieur chercheur, CEA	*Co-encadrant de thèse*

Dauphine | PSL UNIVERSITÉ PARIS

On the impact of randomization on robustness in machine learning

A probabilistic point of view on adversarial examples

Rafaël PINOT

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Université Paris-Dauphine – PSL Research University

in collaboration with

Institut LIST CEA – Université Paris Saclay

under the joint supervision of

Pr. Jamal ATIF, Dr. Cédric GOUY-PAILLER and Dr. Florian YGER

À la glaise, aux étoiles et à mon premier élève.
Tu aimais tellement les regarder ...

Remerciements

En premier lieu, je veux remercier l'ensemble des enseignants qui m'ont inculqué leur amour d'apprendre et de transmettre. Que ce soit au lycée, ou pendant mes études supérieures, j'ai croisé la route de personnes passionnées qui m'ont donné l'envie de faire le métier de chercheur.

Je souhaite exprimer ma profonde reconnaissance envers Stéphane Canu et Panayotis Mertikopoulos pour avoir accepté d'être les rapporteurs de cette thèse, et pour leur lecture attentive du manuscrit ainsi que leurs commentaires détaillés. Je remercie Francis Bach d'avoir présidé mon jury ainsi que Sébastien Bubeck, Cordelia Schmid et Michèle Sébag d'avoir accepté d'en faire partie. C'est un véritable honneur d'avoir pu leur présenter mes travaux.

Je remercie évidemment Jamal Atif, Cédric Gouy-Pailler et Florian Yger pour ces trois années passées à mes côtés et pour la qualité de leur encadrement. Si cette thèse a été pour moi une expérience professionnelle et personnelle aussi réussie, je le leur dois en grande partie. Je remercie Jamal pour sa rigueur scientifique et son exigence intellectuelle. Il m'a appris à prendre du recul et à avoir une vision plus globale des domaines que nous avons abordés. Son investissement scientifique et humain dans mes travaux et mon bien être a été sans faille. J'espère pouvoir être un jour pour un de mes étudiants le directeur de thèse qu'il a été pour moi. Je remercie Cédric pour son pragmatisme et sa sérénité. Il m'a permis de faire avancer mes réflexions dans un environnement calme et bienveillant. Mais surtout, il a su déceler en moi la volonté de diffuser mes connaissances à un public plus large et il m'a offert la chance de me former aux différents aspects de la médiation scientifique. Pour cela, et pour tant d'autres choses, je lui suis éternellement reconnaissant. Je remercie Florian pour sa prévenance et pour m'avoir fait partager son expérience internationale. Il m'a incité en début de thèse à postuler à un programme d'échange avec un pays qu'il connait bien : le Japon. Ce voyage m'a permis de prendre du recul sur le monde de la recherche et a été une expérience de vie extraordinaire. Cet intérêt commun pour l'univers nippon nous a rapprochés et a marqué le début d'une belle relation humaine.

Speaking of my Japanese experience, I would like to thank the JSPS and the CNRS for making this stay possible and Hisashi Kashima for welcoming me in his team at Kyoto University. It was a pleasure to work with him and his collaborators and I hope we will meet again soon.

Pendant ces trois années, j'ai eu l'occasion de collaborer avec des chercheurs formidables que ce soit à Dauphine ou au CEA. Je remercie tout particulièrement Yann Chevaleyre et Benjamin Negrevergne pour leurs conseils et pour tous ces moments où ils m'ont fait partager leur expérience. Je remercie également Alexandre Araujo, Raphael Ettedgui, Arnaud Grivet-Sebert, Geovani Rizk, Laurent Meunier, Anne Morvan, Renaud Sirdey et Martin Zuber pour les collaborations fructueuses que nous avons développées et pour toutes ces heures au tableau à refaire le monde à

notre façon. Je n'oublie pas bien sûr la joyeuse bande de doctorants et d'enseignants-chercheurs avec qui ce fut un plaisir d'échanger et de passer de bons moments autour d'une bière, d'un repas ou d'un café. Plus globalement, et parce qu'il serait trop long de citer tout le monde, je remercie l'ensemble des équipes scientifiques et administratives des deux laboratoires qui m'ont accueilli.

I would like to thank Claudia Hrvatin for accepting to proofread a preliminary version of this manuscript's introduction and for her detailed comments that definitely helped improving it.

Je veux également remercier la famille et les amis. Vous m'avez toujours soutenu et accordé votre confiance, même dans les moments les plus difficiles. Vous avez su vous armer de patience quand parfois j'ai manqué de présence, tout en m'offrant à chacune de nos retrouvailles, des instants de vie réels et profonds. Amytis, pour t'exprimer toute mon affection et ma gratitude, les mots me manquent. Un seul peut-être me vient à l'esprit, tu le connais déjà.

Abstract

This thesis investigates the theory of robust classification under adversarial perturbations – *a.k.a.* adversarial attacks. An adversarial attack refers to a small – humanly imperceptible – change of an input specifically designed to fool a machine learning model. The vulnerability of state-of-the-art classifiers to these attacks has genuine security implications especially for deep neural networks used in AI-driven technologies – *e.g.* for self-driving cars. Besides security issues, this shows how little we know about the worst-case behaviors of models the industry uses daily. Accordingly, it became increasingly important for the machine learning community to understand the nature of this failure mode to mitigate the attacks. One can always build trivial classifiers that will not change decision under adversarial manipulation – *e.g.* constant classifiers– but this comes at odds with standard accuracy of the model. This raises several questions. Among them, we tackle the following one:

Can we build a class of models that ensure both robustness to adversarial attacks and accuracy?

We first provide some intuition on the adversarial classification problem by adopting a game theoretical point of view. We present the problem as an infinite zero-sum game where classical results – *e.g.* Nash or Sion theorems – do not apply. We then demonstrate the non-existence of a Nash equilibrium in this game when the classifier and the adversary both use deterministic strategies. This constitutes a negative answer to the above question in the deterministic regime. Nonetheless, the question remains open in the randomized regime. We tackle this problem by showing that randomized classifiers outperform deterministic ones in term robustness against realistic adversaries. This gives a clear argument for further studying randomized strategies as a defense against adversarial example attacks.

Consequently, we present an analysis of randomized classifiers – *i.e.* classifiers that output random variables – through the lens of statistical learning theory. To do so, we first define a new notion of robustness for randomized classifiers using probability metrics. This definition boils down to forcing the classifier to be locally Lipschitz. We then devise bounds on the generalization gap of any randomized classifier that respects this new notion of robustness. Finally, we upper-bound the adversarial gap – *i.e.* the gap between the risk and the worst-case risk under attack – of these randomized classifiers.

Finally, we highlight some links between our line of research and another emerging topic in machine learning called differential privacy. Both notions build upon the same theoretical ground – *i.e.* stability of probability metrics. Therefore, results from one domain can be transferred to the other. Based on this idea, we use the differential privacy literature to design a simple noise injection method. The scheme allows us to build a class of robust randomized classifiers out of a deterministic hypothesis class, making our previous findings applicable to a wide range of machine learning models.

Open questions and perspectives for future research conclude this work.

Résumé

Cette thèse étudie la théorie de la classification robuste aux attaques adverses. Une attaque adverse est une modification imperceptible de l'entrée d'un algorithme, spécifiquement conçue pour provoquer un dysfonctionnement de celui-ci.

La vulnérabilité des modèles d'intelligence artificielle à ces attaques pose de véritables problèmes en matière de sécurité, notamment en ce qui concerne les réseaux neuronaux profonds utilisés dans les nouvelles technologies, par exemple pour les voitures autonomes. Outre les questions de sécurité, cela montre à quel point nous en savons peu sur le comportement des modèles que l'industrie utilise quotidiennement. Par conséquent, il devient de plus en plus important pour la communauté scientifique de comprendre d'où proviennent ces défaillances. Parmi les nombreuses questions que soulèvent les attaques adverses, nous abordons la suivante :

Pouvons-nous construire une classe de modèles qui garantissent à la fois la robustesse aux attaques adverses et la précision dans des taches classiques?

Nous donnons d'abord quelques intuitions en abordant le problème sous l'angle de la théorie des jeux. Nous formalisons la classification robuste comme un jeu à somme nulle infini et démontrons la non-existence d'un équilibre de Nash dans ce jeu lorsque le modèle et l'adversaire utilisent tous les deux des stratégies déterministes. Ceci constitue une réponse négative à la question ci-dessus dans le cas déterministe. Néanmoins, la question reste ouverte si l'on prend en compte des stratégies aléatoires. Nous abordons ensuite ce problème en montrant que les modèles aléatoires, c'est-à-dire des modèles qui produisent des variables aléatoires, obtiennent de meilleurs résultats que les modèles déterministes en termes de robustesse aux attaques. Cela donne un argument fort en faveur des stratégies aléatoires.

Par conséquent, nous présentons une analyse approfondie des modèles aléatoires. Pour ce faire, nous définissons une nouvelle notion de robustesse à l'aide de métriques/divergences sur les espaces des distributions de probabilité. Ensuite, nous étudions le comportement en terme d'erreur de généralisation de tout modèle aléatoire qui respecte cette nouvelle notion de robustesse. Enfin, nous adaptons notre analyse à la généralisation adverse, c'est-à-dire l'écart entre le risque théorique et le risque adverse de ces modèles.

Enfin, nous mettons en évidence certains liens entre notre champ de recherche et un autre sujet émergent dans le domaine de l'apprentissage automatique, à savoir la protection des données personnelles. Ces deux notions reposent sur le même fondement théorique. Par conséquent, les résultats d'un domaine peuvent être transférés dans l'autre. Sur la base de cet constat, nous utilisons la littérature sur la protection des données personnelles pour concevoir une méthode simple d'injection de bruit. Cette méthode nous permet de construire une classe de modèles aléatoires robustes à partir d'une classe de modèles (déterministes) précis dans des tâches classiques.

Nous concluons se manuscrit par des questions ouvertes et des perspectives de recherche.

Contents

List of Figures and Tables

Notations and Symbols

We use bold lower-case to denote vectors and functions with multidimensional outputs and standard lower-case to denote scalars and real-value functions. Depending on the context, we either use calligraphic font or upper-case to denote ensembles – most of the times calligraphic, sometimes upper-case to denote sub-sets or elements of a set of sets.

Algebra

$\mathbb{R}$	Set of real numbers	
$\mathbb{N}$	Set of natural integers	
$\mathbb{R}^d$	Set of d-dimensional real-valued vectors	
$\mathcal{M}_{d\times d'}(\mathbb{R})$	Set of $d \times d'$ real-valued matrices	
I_d	$d \times d$ identity matrix	
$[a]$	Set of integers between 1 and a	$[a] := \{1, \dots, a\}$
$\Delta(K)$	K dimensional simplex	$\Delta(K) := \{\boldsymbol{z} \in \mathbb{R}^K \textit{ s.t. } \|\boldsymbol{z}\|_1 = 1\}$
$\|\boldsymbol{v}\|_p$	ℓ_p-norm of $\boldsymbol{v} \in \mathbb{R}^d$ for $p \in [1, +\infty)$	$\|\boldsymbol{v}\|_p = \left(\sum_{i=1}^d \|\boldsymbol{v}_i\|^p\right)^{1/p}$
$\|\boldsymbol{v}\|_\infty$	Infinite norm of $\boldsymbol{v} \in \mathbb{R}^d$	$\|\boldsymbol{v}\|_\infty = \max_{i\in[d]}(\|\boldsymbol{v}_i\|)$
$\|\boldsymbol{v}\|_M$	Mahalanobis norm of $\boldsymbol{v} \in \mathbb{R}^d$ with $M \in \mathcal{M}_{d\times d}(\mathbb{R})$	$\|\boldsymbol{v}\|_M = \sqrt{\boldsymbol{v}^\intercal M \boldsymbol{v}}$
$B_p(\boldsymbol{v}, \alpha)$	ℓ_p ball with center $\boldsymbol{v} \in \mathbb{R}^d$ and radius $\alpha \geq 0$	$\{\boldsymbol{u} \textit{ s.t. } \|\boldsymbol{u} - \boldsymbol{v}\|_p \leq \alpha\}$
$B_p(\alpha)$	ℓ_p ball with center 0 and radius $\alpha \geq 0$	$\{\boldsymbol{u} \textit{ s.t. } \|\boldsymbol{u}\|_p \leq \alpha\}$
$\text{Vol}(B)$	Volume of the sub-space $B \subset \mathbb{R}^d$	

Probability

$\mathcal{A}(\mathcal{Z})$	σ-algebra of an arbitrary space $\mathcal{Z}$
$\mathcal{P}(\mathcal{Z})$	Set of probability distribution over $(\mathcal{A}(\mathcal{Z}), \mathcal{Z})$
$\mathcal{F}_{\mathcal{Z}\times\mathcal{Z}'}$	Set of measurable functions from $\mathcal{Z}$ to $\mathcal{Z}'$
$\boldsymbol{\psi}\#\rho$	Push-forward of $\rho \in \mathcal{P}(\mathcal{Z})$ by $\boldsymbol{\psi} \in \mathcal{F}_{\mathcal{Z}\times\mathcal{Z}'}$
$\mathbb{E}[.]$	Expectation of a random event
$\mathbb{P}[.]$	Probability of a random event
$\mathcal{N}(.,.)$	Gaussian distribution
$\text{Lap}(.,.)$	Laplace distribution
Φ	cdf of the standard Gaussian distribution $\mathcal{N}(0,1)$

Classification and Learning theory

$\mathcal{X}$	Input space
d	Dimension of the input space
$\mathcal{Y}$	Output space
K	Number of classes
$\mathcal{D}$	Ground-truth distribution
$\mathcal{S}$	Training sample
$\mathcal{H}$	Hypothesis space
$\mathcal{L}$	Loss function

Functions

$\mathbb{1}\{.\}$	Indicator function of an event	$\mathbb{1}\{A\} = 1$ if A is true, 0 otherwise
$\text{sign}(x)$	Sign function applied on x	$\text{sign}(x) = 1$ if $x > 0$, -1 if $x < 0$ and 0 if $x = 0$

Abbreviations

a.k.a.	**a**lso **k**nown **a**s
cdf	**c**umulative **d**ensity **f**unction
C & W	**C**arlini and **W**agner (attack)
e.g.	***e****xempli* ***g****ratia*
Eq.	**Eq**uation
ERM	**E**mpirical **R**isk **M**inimization
FGM	**F**ast **G**radient **M**ethod (attack)
i.e.	***i****d* ***e****st*
i.i.d.	**i**dentically and **i**ndependently **d**istributed
PGD	**P**rojected **G**radient **D**escent (attack)
resp.	**resp**ectively
s.t.	**s**uch **t**hat
SRM	**S**tructural **R**isk **M**inimization
std	**st**andard **d**eviation
w.r.t.	**w**ith **r**espect **t**o

Foreword

Funding & grants

This thesis was prepared in the MILES team, part of the LAMSADE lab, at Université Paris Dauphine-PSL, and the LI3A lab, at CEA LIST from October 2017 to October 2020. It was funded by a grant from CEA. Rafael Pinot was also awarded with a Summer Program Fellowship from the Japanese Society for the Promotion of Science in 2018 to spend three months in Kyoto university – invited by Professor Hisashi Kashima – as a visiting scholar. Finally, this work was granted acces to OpenPOWER prototype from GENCI-IDRIS under the Preparatory Access AP010610510, HPC resources of IDRIS under the allocation 2020-101141 made by GENCI, and HPC resources of FactoryIA partially funded by région Ile-de-France – projet SESAME 2017.

Manuscript main focus

This thesis work focused on the notions of security and privacy in machine learning. We list below some of the contributions we made. In the main part of this manuscript, we try to give a clear overview of our contributions to the field of robust machine learning. We deliberately focus on some of our most significant contributions to make the manuscript light and easy to follow. This thesis work also took other directions including unsupervised learning under privacy constraints, private deep learning, and cryptography.

Contributions discussed in the main part of the manuscript

- "Theoretical evidence for adversarial robustness through randomization" (chap. 4).
  Journal version, ongoing work 2020.
- "Randomization matters, how to defend against strong adversarial attacks" (chap. 3).
  International Conference on Machine Learning (ICML) 2020.
  R. Pinot, R. Ettedgui, G. Rizk, Y. Chevaleyre, J. Atif.
- " A unified view on differential privacy and robustness to adversarial examples" (chap. 5).
  Workshop on Machine Learning for CyberSecurity (ECML-PKDD) 2019.
  R. Pinot, F. Yger, C. Gouy-Pailler, J. Atif.
- "Theoretical evidence for adversarial robustness through randomization" (chap. 4 & 5).
  Avances in Neural Information Processing (NeurIPS) 2019.
  R. Pinot, L. Meunier, A. Araujo, H. Kashima, F. Yger, C. Gouy-Pailler, J. Atif.

Other contributions

- "SPEED: Secure, PrivatE, and Efficient Deep learning" (app. C).
  preprint, ongoing work 2020.
  A. Grivet Sébert, R. Pinot, M. Zuber, C. Gouy-Pailler, R. Sirdey.
- "Advocating for Multiple Defense Strategies against Adversarial Examples" (app. A).
  Workshop on Machine Learning for CyberSecurity (ECML-PKDD) 2020.
  A. Araujo, L. Meunier, R. Pinot and B. Negrevergne.
- "Graph-based Clustering under Differential Privacy" (app. B).
  Uncertainty in Artificial Intelligence (UAI) 2018.
  R. Pinot, A. Morvan, F. Yger, C. Gouy-Pailler, J. Atif.

Scientific Outreach

Throughout this thesis, we did not only focus on the production of scientific material. We also engaged in scientific outreach through demos and press releases. We believe that it is also the role of scientists, especially in the field of machine learning and artificial intelligence, to explain their work to a wider audience within the scientific community, and to increase public knowledge on the challenges and issues of new technologies. Here are some of our contributions.

- "Attaques adversariales: comprendre pour atténuer les risques" (media article).
  Clef du CEA num 69 2020.
  Contributors: R. Pinot, C. Gouy-Pailler.
- "AI vs Wild. How to strengthen neural networks of AI systems" (demo).
  Consumer Electronic Show Las Vegas 2020.
  Contributors: C. Gouy-Pailler, E. Kawasaki, R. Pinot, F. Valente.
- "Randomization based defenses against adversarial examples" (demo).
  DIGIHALL days Paris Saclay 2019.
  Contributors: R. Pinot, C. Gouy-Pailler.
- "La recherche et les risques inhérents à l'IA" (media article).
  Préventique num 166 2019.
  Contributors: R. Pinot, C. Gouy-Pailler.

Pedagogical responsibilities

Last but not least, teaching is an integral part of a doctoral journey, and the rapid development of machine learning requires the design of new learning materials. During the time of this thesis, I also participated in the development of two new machine learning courses.

- "Mathématiques du machine learning" – Université Paris-Dauphine - PSL.
  Master IDD première année 2019-2020.
  Lecturer: R. Pinot.
- "Trustworthy machine learning in practice" – Université Paris-Dauphine - PSL.
  Executive Master 2019-2020.
  Lecturers: A. Araujo, R. Pinot, G. Rizk.

Reading guide

The organization of the manuscript has been designed to be simple, and the chapter titles should be self-explanatory. We advise the reader to follow a linear reading style – chapter by chapter. However, readers already familiar with the concepts and existing results in learning theory and adversarial classification may skip Chapter 2 and go directly to the technical chapters 3, 4, and 5. Each chapter is divided into four main parts.

1. An introduction to the specific issues and terminology of the chapter – one section.
2. Simple results with a focus on consequences and interpretations – one or two section(s).
3. Extension of previous results to more difficult/technical contexts – one section. This section can be skipped for a first reading as it is not essential to understand the overall message. Nevertheless, we encourage the interested reader to skim over additional results in Chapter 4 to better understand the link between privacy and robustness we establish in Chapter 5.
4. Lessons learned and future works – one section. We summarize our findings and discuss some future research directions.

1 Introduction

Contents

Machine learning models are part of our everyday life and their weaknesses in terms of security or privacy can be used to harm us either directly or indirectly. It is thus crucial to be able to account for, and deal with, any new vulnerabilities. Besides, the legal framework in Europe is evolving, forcing practitioners – from both the private and the public sectors – to adapt quickly to these new concerns. In this chapter, we first present the context in which the idea of this thesis was born and our main motivations in Section 1.1. Then, we present the problem which we have chosen to focus on: *robust classification under adversarial perturbation* in Section 1.2. Finally, we summarize some of our contributions to the domain in Section 1.2.3.

1.1 Context & motivations

In the 1950s, the first artificial intelligence projects were developed[1]. At that time, the ultimate goal was the replication of human intelligence. The proposed approaches consisted of using mathematics to describe the world, model the human perception, and simulate the cerebral mechanisms. Seventy years later, the initial objective of replication of brain's functions has been largely supplanted by technological projects aiming to reproduce human performance in simple cognitive tasks [142]. To this end, deep neural networks achieve state-of-the-art performance in a variety of domains such as natural language processing [132], image recognition [76] and speech recognition [79]. The impressive efficacy of AI-driven technologies has made them omnipresent both in industry and in some public sectors. However, recent studies have identified several major flaws of machine learning and data analysis such as information leakage [120] or vulnerability to adversarial perturbations [20]. These shortcomings raise questions about the legal liability of model providers and cause practitioners to reevaluate the trust they place in the systems they use.

[1] The Dartmouth conference of 1956 is often considered the founding act of the artificial intelligence project. However, it follows several pioneer works on the notion of machine intelligence by Mc Culloch, Pitts and Wiener [105, 157, 165] in cybernetics and by Turing [157] in computer science.

1.1.1 Dealing with privacy issues: the General Data Protection Regulation

Protecting individuals' privacy against information leakage while producing statistical analysis is already an old topic; its foundations were largely established in the 1980s [3, 42, 65]. These concerns were brought back to light notably in 2008, when Narayanan *et al.* [120] demonstrated a robust de-anonymization procedure on the dataset released for the "Netflix Price contest". In 2016, the European Union provided an answer to these concerns from a legal standpoint by publishing the General Data Protection Regulation [126] – GDPR.

1. *Transparency.* Personal data must be processed with transparency, and data owners can ask to be informed at any time on how their data are being used.

---

2. *Limited purposes and retention.* Personal data must be collected for clear, understandable and legitimate purposes, and should not be processed for any other purpose than the initial ones. Moreover, data must be kept for a period that does not exceed the processing of the selected purposes.

---

3. *Data integrity and privacy.* Personal data must be processed in such a way as to ensure appropriate security of such data, including protection against unauthorized or unlawful processing and against accidental loss.

---

4. *Access, rectification, erasing.* Data owners should be granted easy access to their personal data as well as the possibility to rectify of any inaccuracy or to erase any personal data.

---

5. *Privacy by design.* Each new technology or application processing personal data, or making it possible to process it, must ensure, from its design and each time it is used, that it incorporates all the protection principles of the GDPR.

Figure 1.1: Key principles of the GDPR on the development of new machine learning models.

This regulation aims to define the duties of model providers with respect to the personal data they use – see Figure 1.1 for an overview of the key principles[2]. In order to comply with the GDPR, industries and governments are required to design models that preserve privacy. These new obligations, coupled with already existing users' concerns regarding their personal data, have made privacy issues the priority within the computer science community. Accordingly, several definitions have been introduced to characterize privacy preserving algorithms in the context of machine learning and data publishing [57]. Among them, differential privacy [52] has become the dominant standard to provide a formal and adaptive conception of privacy preserving data analysis. The rationale is that one individual's information is protected if *"the outcome of any analysis is*

[2] We do not claim to provide a thorough presentation of this regulation here. To keep the discussion concise, we only highlight some points that we – as computer scientists – believe to be central.

essentially equally likely, independent of whether any individual joins, or refrains from joining, the data set" [53].

More formally, an algorithm is said to be differentially private if, given two similar databases, it produces statistically indistinguishable outputs. This privacy definition has been broadly investigated in numerous frameworks and applications – see [50] for a book of reference. Overall, privacy preserving machine learning is now a well-known and accepted concept. It relies on a proper legal framework, and technical solutions such as differential privacy are consistently being implemented by major companies – *e.g.* Google [54, 166] – and public entities – *e.g.* the U.S. Census Bureau [102]. The GDPR has been a real revolution both from a legal and scientific standpoint. In our point of view, the battle for users' privacy is not over yet, but significant efforts have been made both by practitioners and researchers to meet the privacy requirements of our era.

1.1.2 Beyond privacy: interpretability, trust and and adversarial attacks

Despite focusing on data protection, the GDPR also includes an article – Article 22 – on the right to receive an explanation for an algorithmic decision [126]. This raises a number of questions on both the interpretability of machine learning algorithms and the trust users place in them [68]. While there is no clear consensus yet on the definition of interpretability or trust in machine learning [24], recurring themes such as social bias [4] or vulnerability to perturbations [20, 152] often resurface. These new concerns, along with the privacy issues mentioned above, are sometimes put together under the name *trustworthy machine learning* and have lately attracted a lot of attention[3]. Furthermore, the deployment of machine learning in real-world systems and the recent legal progress on data protection and decision explanation should encourage intensifying the research in this new domain.

In this thesis, our primary focus is the models' vulnerability to adversarial perturbations. The term adversarial perturbation – *a.k.a.* adversarial attack – denotes a carefully chosen and humanly imperceptible perturbation that causes a model to fail. The existence of these vulnerabilities shows how far the deep learning community has drifted from the initial goal of reproducing the human perception. To demonstrate the genuine security issue that adversarial attacks represent, we take the example of self-driving cars. Recently, technology companies have made enormous investments in self-driving cars – *i.e.* autonomous vehicles equipped with a tremendous amount of cameras and sensors that help them move with little to no human input. Much of the information gathered by these cars is processed using in-vehicle machine learning models. In particular, vision tasks process images through deep neural networks. However, recent works [55, 146, 150, 174] have indicated that these very systems can be fooled by real-world adversarial attacks on traffic signs – *e.g.* by adding stickers on the traffic sign.

Figure 1.2 illustrates an attack setting where an adversary added such a sticker on a traffic sign. In the first schematic – at top – the car captures the original version of the traffic sign, recognizes it as a speed limitation, and goes on normally. In the second schematic, at bottom, the red car captures an adversarial version of the traffic sign and recognizes it as a stop sign causing an accident with

[3] Top tier machine learning conferences started launching several workshops on this matter – see *e.g.* https://trustworthyiclr20.github.io/ or https://icml2019workshop.github.io/. Note also that the number of paper on this matter have been growing exponentially in the last few years – see *e.g.* https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html for vulnerability to adversarial perturbations.

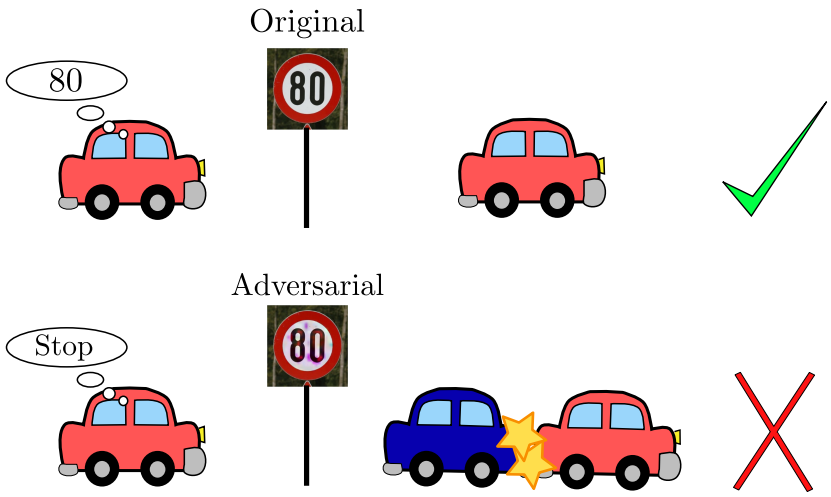


Figure 1.2: Illustration of a self-driving car misclassifying real-world traffic signs with adversarial perturbations. On the top line: scenario without attack. On the bottom line: scenario with attack. Traffic sign images come from a real-life attack paper [150].

the blue car. Note that in this case, no human would have change his/her decision, but the car did. This gap between the human and model responses could lead to various security issues – here for example an accident triggered by an attack on a traffic sign. These technologies are currently being deployed; it is thus crucial to adapt quickly to this new threat both from a technical and legal standpoint. In the sequel, we will use self-driving cars as a running example. Accordingly, we will focus our application setting to deep learning for image classification.

1.2 Problem setting(s)

The vulnerability of machine learning and deep learning models to adversarial attacks is a critical security issue, especially for high-stakes applications such as self-driving cars. It is essential for the community to understand the nature of this phenomenon in order to mitigate the threat. In this section, we start by giving some reminders on the problem of classification in the standard setting – *i.e.* without adversary. Then we present the problem of classification in the adversarial setting and identify the core questions to which we aim to provide some answers. Finally, we outline the main questions we wish to address in this manuscript.

1.2.1 Classification in the standard setting

Let us consider the supervised classification problem with an input space $\mathcal{X}$ – *e.g.* images – and an output space $\mathcal{Y}$ – *e.g.* label describing the images. For simplicity here, we will consider that $\mathcal{Y} = \{1, \dots, K\}$, meaning that each description is characterized by an integer between 1 and K. The goal of a supervised machine learning algorithm is to design an accurate prediction function $c : \mathcal{X} \to \mathcal{Y}$ – *a.k.a.* classifier – that for any image $\boldsymbol{x} \in \mathcal{X}$ matches a label $y \in \mathcal{Y}$ that correctly describes the image. To find c, the learner has access to a set of n input-output pairs

$\mathcal{S} := \{(\boldsymbol{x_1}, y_1), \dots, (\boldsymbol{x_n}, y_n)\}$ – *a.k.a. training sample*. The main assumption behind the theory of classification is that there exists some ground truth distribution $\mathcal{D}$ that describes the connection between the images and the labels and from which the pairs $(\boldsymbol{x_i}, y_i)$ are drawn *i.i.d.*.

To build a classifier, the usual strategy is to build a hypothesis function $\boldsymbol{h} : \mathcal{X} \to \mathbb{R}^K$ that for any $\boldsymbol{x} \in \mathcal{X}$ outputs a set of scores $\boldsymbol{h}(\boldsymbol{x}) := [\boldsymbol{h}_1(\boldsymbol{x}), \dots, \boldsymbol{h}_K(\boldsymbol{x})]^\intercal$ – one for every possible label. Then, the prediction function c outputs the label with the better score for $\boldsymbol{h}$. More formally, c writes $c(\boldsymbol{x}) := \operatorname{argmax}_{k \in [K]} \boldsymbol{h}_k(\boldsymbol{x})$. The problem then amounts to build a function $\boldsymbol{h}$ that describes well the connection between the images and the labels. To do so, the learner aims to select $\boldsymbol{h}^*$ from a predefined set $\mathcal{H}$, called the hypothesis class, that solves – or approximate – the *risk minimization* problem. This optimization problem writes

$$\inf_{\boldsymbol{h} \in \mathcal{H}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}(\boldsymbol{h}(\boldsymbol{x}), y)]\,, \tag{1.1}$$

where $\mathcal{L} : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}$ is some loss function that measures how well $\boldsymbol{h}$ fits the ground-truth distribution. If $\mathcal{L}$ is sufficiently well chosen – typically if it is convex and smooth [9] – and if the hypothesis class $\mathcal{H}$ is rich[4] enough, the classifier c we get will have a small probability to give the wrong label for a new sample $(\boldsymbol{x}, y) \sim \mathcal{D}$.

In practice, the learner does not have access to the ground-truth distribution; hence it cannot estimate the risk $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}(\boldsymbol{h}(\boldsymbol{x}), y)]$. To find an approximate solution for Problem (1.1), a learning algorithm solves the *empirical risk minimization* problem instead. In this case, we simply replace the risk by its empirical counterpart over the training $\mathcal{S} := \{(\boldsymbol{x_1}, y_1), \dots, (\boldsymbol{x_n}, y_n)\}$. It writes

$$\inf_{\boldsymbol{h} \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x_i}), y_i)\,. \tag{1.2}$$

Then, to evaluate how far the selected hypothesis $\boldsymbol{h}_\mathcal{S}$ is from the optimal $\boldsymbol{h}^*$, one wants to upper bound the difference between the risk and the empirical risk of any $\boldsymbol{h} \in \mathcal{H}$. This difference is known as the *generalization gap*. Intuitively, if we can control the difference between the risk and the empirical risk of any function in $\boldsymbol{h} \in \mathcal{H}$, then the risk minimization problem and the empirical risk minimization problem will have similar solutions.

In light of the above, the choice of the hypothesis class $\mathcal{H}$ in supervised classification is critical. On one hand, if it is too large, it will be hard to control the generalization gap of all the elements in the class and the optimization problem is difficult. On the other hand, if it is too small, the generalization gap will be easy to control but the class might not be sufficiently rich to describe the behavior of the ground-truth distribution, which leads to poor prediction functions. Another key component is the size of the training sample. If we have enough training samples, thanks to the uniform law of large numbers, the empirical risk of any hypothesis is a good approximation for its true risk. More precisely, for some well chosen hypothesis classes one can bound the generalization gap of any hypothesis by $O\left(\frac{1}{\sqrt{n}}\right)$. Then, when the sample size n is big enough, it is sufficient to solve the empirical risk minimization – Problem (1.2) – to get good approximation for the risk

[4] The richness of a hypothesis set is a complicated notion. We will later discuss it in more depth. For now, one may think of this notion as the size of the hypothesis class. When the hypothesis class is large enough, it is easy to find at least one $\boldsymbol{h}$ that describes $\mathcal{D}$ well. Conversely, when it is too small it is hard to find a good candidate.

minimization – Problem (1.1). Let us now present the alternative classification setting we will study in this manuscript, namely *Classification under adversarial perturbation*.

Reading note. *The interested reader can find a more thorough introduction to classification and learning theory in Chapter 2.*

1.2.2 Classification under adversarial perturbation

Given a hypothesis $\boldsymbol{h} \in \mathcal{H}$ and an image-label pair $(\boldsymbol{x}, y) \sim \mathcal{D}$, the goal of an adversary is to find a perturbation $\boldsymbol{\tau} \in \mathcal{X}$ such that the following assertions *both* hold.

- The perturbation is imperceptible to humans. Strictly speaking, this means that a human cannot visually distinguish the standard example $\boldsymbol{x}$ from the *adversarial example* $\boldsymbol{x}+\boldsymbol{\tau}$. In a less conservative viewpoint, this could also mean that a human will give the same answer if it is asked to classify $\boldsymbol{x}$ or $\boldsymbol{x} + \boldsymbol{\tau}$. For simplicity, we consider the strict definition here.
- The perturbation modifies $\boldsymbol{x}$ enough to make the classifier misclassify. More formally, the adversary seeks a perturbation $\boldsymbol{\tau} \in \mathcal{X}$ such that $c(\boldsymbol{x} + \boldsymbol{\tau}) \neq y$.

Although the notion of imperceptible modification is very natural for humans, it is genuinely hard to formalize. Despite these difficulties, a sufficient condition to ensure that the attack will remain undetected is to constrain the perturbation $\boldsymbol{\tau}$ to have a small ℓ_p norm. This means that for any $p \in [1, \infty]$, there exists a threshold $\alpha_p > 0$ for which any perturbation $\boldsymbol{\tau}$ is imperceptible as soon as $\|\boldsymbol{\tau}\|_p \leq \alpha_p$. The literature on adversarial attacks for image classification [27, 103] usually uses either an ℓ_∞ or an ℓ_2 norm as a surrogate for imperceptibility [5].

Remark 1. *Note that these norms have very different behaviors in high-dimensional spaces, hence the choice of p has a crucial impact on the answer one provides to **Q1** and **Q2** below. We will further discuss this point in Chapter 2 and Appendix A.*

Adversarial examples represent a serious security threat that machine learning models should deal with. To do so, we need to revisit the standard risk minimization by incorporating the adversary in the problem. The goal becomes to minimize the *worst-case* risk under α_p-bounded manipulations. We call this problem the *adversarial risk minimization*. It writes

$$\inf_{\boldsymbol{h} \in \mathcal{H}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\sup_{\boldsymbol{\tau} \in B_p(\alpha_p)} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x} + \boldsymbol{\tau}), y) \right], \tag{1.3}$$

where $B_p(\alpha_p) := \{\tau \in \mathcal{X} \text{ s.t. } \|\boldsymbol{\tau}\|_p \leq \alpha_p\}$. In this new problem, the adversary focuses on optimizing the inner maximization, while the classifier tries to get the best hypothesis $\boldsymbol{h}^*$ from $\mathcal{H}$ "under attack". In the standard setting, we can most of the time design sufficiently rich hypothesis classes such that the risk minimization problem gives a solution $\boldsymbol{h}^*$ with small risk. But in the adversarial setting, it becomes unclear whether this statement still holds. Hence the following question.

[5] Sometimes, the adversary uses an ℓ_1 norm [33] or an ℓ_0 semi-norm [124].

Q1: *Is there some hypothesis class $\mathcal{H}$ for which the adversarial risk minimization problem has a solution $\boldsymbol{h}^*$ with small adversarial risk?*

At a first glance – looking at the empirical literature on adversarial examples – the answer seems to be no. Indeed, a large body of works has been trying to design new models that would be less vulnerable to the adversarial setting [67, 81, 107, 162, 170] but most of them were proven – in time – to offer only limited protection against more sophisticated attacks [6, 27, 38, 77, 154]. Nevertheless, it is important to investigate this question from a theoretical point of view to provide either definitive negative answers or to design more robust models.

Let us suppose for a moment that **Q1** has a positive answer and that we can design a hypothesis class $\mathcal{H}$ for which the adversarial risk minimization has a solution $\boldsymbol{h}^*$ with small adversarial risk. By analogy with the standard setting, given n training examples $\mathcal{S} := \{(\boldsymbol{x_1}, y_1), \dots, (\boldsymbol{x_n}, y_n)\}$, we want to find an approximate solution to the adversarial risk minimization by studying its empirical counterpart, the *empirical adversarial risk minimization*. This optimization problem writes

$$\inf_{\boldsymbol{h}\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x_i}+\boldsymbol{\tau}), y_i)\,. \tag{1.4}$$

In the presence of an adversary, two major issues appear in the empirical risk minimization. First, as recently pointed out by Madry *et al.* [103], the adversarial generalization gap – *i.e.* the gap between the empirical adversarial risk and the adversarial risk – can be much larger than in the standard setting. Indeed, the adversary makes the problem dependent on the dimension of $\mathcal{X}$. Hence, in high-dimensional spaces – *e.g.* for images – one needs much more samples to classify correctly [141, 148]. Second, finding an approximated solution to the adversarial risk minimization is not always sufficient. Two recent works [156, 180] gave theoretical evidence that training a robust model may lead to an increase of its standard risk. Hence finding a good approximation for the adversarial classification problem – Problem (1.3) – may lead to a poor solution for the standard problem – Problem (1.1). Accordingly, a second question emerges.

Q2: *Can we find a class $\mathcal{H}$ and a hypothesis $\boldsymbol{h}^* \in \mathcal{H}$ that simultaneously has small standard and adversarial risks?*

1.2.3 Main contributions and outline of the thesis

In this thesis, we aim to provide answers to the previously stated problems. Our contribution is threefold. First, we analyze the adversarial classification problem and provide results showing that randomized classifiers – *i.e.* classifiers that output a random variable – are good candidates to give a positive answer to **Q1**. Second, we identify sub-classes of randomized classifiers that provide some positive answers to both **Q1** and **Q2**. Finally, we present simple schemes to build these classes by bridging connections with privacy preserving machine learning.

Analyzing the adversarial classification problem – Q1

Our first contribution consists in building new intuitions on the adversarial classification problem. To do so, we cast the adversarial risk minimization – Problem (1.3) – as an *infinite* zero-sum game between a defender – the learner – and an adversary that produces adversarial examples. In this new setting, we obtainw the following results.

1. We demonstrate the non-existence of a Nash equilibrium in the game when both the defender and the adversary play deterministic strategies. This, coupled with some recent results from related works [18, 131], entails that deterministic hypothesis classes may not be good candidates to provide a positive answer to **Q1**.

2. From a game theoretical standpoint, the natural next step is to investigate randomized strategies. We focus on randomizing the strategies for the defender – leaving the adversary strategies unchanged. In this context, we demonstrate that randomized classifiers can outperform deterministic ones in terms of worst-case theoretical guarantees – Problem (1.3). Therefore, we identify randomized classifiers as good candidates to answer **Q1** positively.

Theoretical properties of randomized classifiers – Q1 & Q2

For our second contribution, we study randomized classifiers through the prism of learning theory and information theory. By analogy with the deterministic case, we define a notion of robustness for randomized classifiers. This definition boils down to forcing the classifier to satisfy local Lipschitzness with respect to the ℓ_p norm on $\mathcal{X}$, and a probability metric on $\mathcal{Y}$. Denoting $\mathcal{H}_{\text{Lip}}$ the class of randomized classifier that respect this Lipschitz condition, we present the following results.

1. We show that for any $\boldsymbol{h} \in \mathcal{H}_{\text{Lip}}$, we can upper-bound the gap between the risk and the adversarial risk of $\boldsymbol{h}$. This result says that any good approximation of the risk minimization problem – Problem (1.1) – on $\mathcal{H}_{\text{Lip}}$ is also a good approximation of the adversarial risk minimization – Problem (1.3). This means that $\mathcal{H}_{\text{Lip}}$ is a good candidate to answer **Q2**.

2. We devise an upper-bound on the generalization gap of any $\boldsymbol{h}$ in $\mathcal{H}_{\text{Lip}}$. This means that, for a sufficiently large training sample, solving the ERM – Problem (1.2) – on $\mathcal{H}_{\text{Lip}}$ can provide a good approximate solution to the risk minimization problem. Since we can also bound the gap between the adversarial and the standard risk, this gives answers to both **Q1** and **Q2**. Note, however, that this result relies on a strong assumption on $\mathcal{X}$ that does not always bypass dimensionality issues. The problem of finding a subclass of $\mathcal{H}$ that provides tighter generalization bounds is an open question.

Practical schemes based on differential privacy literature – Q2

Previous contributions identified a class of randomized hypotheses $\mathcal{H}_{\text{Lip}}$, that answers both **Q1** and **Q2** – at least partially. But they gave no practical way to design this class. Our final contribution tackles this issue by drawing lessons learned from privacy preserving machine learning. More precisely our contribution is as follows.

1. We highlight some links between our definition of robustness and the definition of differential privacy. Both frameworks build upon the same theoretical ground – *i.e.* stability with respect to probability metrics. Therefore, results obtained so far in differential privacy can easily be transferred to design robust randomized classifiers.

2. Based on this idea, we use two famous tools from differential privacy – namely noise injection and post-processing – to design classes of robust randomized classifiers. In particular, we show that our previous findings are applicable to a wide range of machine learning models, provided some minor adaptations. We further corroborate our findings with experimental results using deep neural networks on standard image datasets – namely CIFAR10 and CIFAR100 [93]. These models can simultaneously provide accurate prediction and reasonable robustness, giving practical answers to **Q2**.

Outline of the thesis

The remainder of the manuscript is organized as follows. Chapter 2 presents an overview of the domain of adversarial classification. Then, Chapters 3, 4 and 5 are devoted to the three main contributions we just presented above. Finally, Chapter 6 concludes this work with additional discussions and open problems. Appendices provide a high-level summary of some additional results obtained during this thesis in terms of robustness to adversarial examples – Appendix A, differential privacy – Appendix B, and cryptography for deep learning – Appendix C.

2 Background

Contents

At the beginning of this thesis work – in 2017 – the vulnerability of machine learning models to adversarial examples was not much studied. But over the past three years, we have seen a massive increase in the number of articles published on this topic[1]. In this chapter, we aim to provide an overview of this emerging field. First, we give some background on image classification and learning theory in Section 2.1. We then review in Sections 2.2 and 2.3 the current state-of-the-art in terms of adversarial attacks and defenses. We present in Section 2.4 some recent results studying the adversarial risk minimization through the lens of learning theory and Section 2.5 asks whether adversarial examples are unavoidable. Finally, we discuss in Section 2.6 how our work contributes to the domain.

2.1 An introduction to learning theory and image classification

We first come back to some of the elements we discussed in the introduction in a more precise manner, and present some prerequisites on classification and learning theory.

2.1.1 Formalizing the classification problem

To begin, let us present the supervised learning setting for classification. In this context, the learner – *e.g.* model provider – has access to the following elements.

- An input space $\mathcal{X}$, which is the set of objects the learner wants to classify. Here we consider a setting where $\mathcal{X}$ is a set of images with d pixels and values in $[0, 1]$; hence $\mathcal{X} \subset [0, 1]^d$. Note that in image classification, there is often thousand of pixels in the image, which means that $\mathcal{X}$ is a high dimensional space. As we will discuss later, this characteristic of the input space plays a key role in our understanding of the adversarial setting.
- An output space $\mathcal{Y}$ that denotes the set of possible labels for elements in $\mathcal{X}$. In the image classification setting, a label is a succinct description of the image. For simplicity, we characterize $\mathcal{Y}$ by a set of K integers $\mathcal{Y} = \{1, \dots, K\} := [K]$.
- A training sample $\mathcal{S} := \{(\boldsymbol{x_1}, y_1), \dots, (\boldsymbol{x_n}, y_n)\}$, which is a set of n elements from $\mathcal{X} \times \mathcal{Y}$. In the supervised learning setting, we hypothesize that these input-output pairs are drawn *i.i.d.*from some ground-truth distribution $\mathcal{D}$ the learner does not have access to.

Remark 2. *Below, we define probabilities and expectations over the ground-truth distribution $\mathcal{D}$. Formally, we assume that there exists a σ-algebra $\mathcal{A}(\mathcal{X} \times \mathcal{Y})$ over $\mathcal{X} \times \mathcal{Y}$ and that $\mathcal{D}$ is a probability measure over $(\mathcal{X} \times \mathcal{Y}, \mathcal{A}(\mathcal{X} \times \mathcal{Y}))$. Accordingly, the set we evaluate are assumed to be in $\mathcal{A}(\mathcal{X} \times \mathcal{Y})$ and the functions we define are measurable.*

With these elements at hand, the goal of the learner is to find a prediction function $c : \mathcal{X} \to \mathcal{Y}$ – *a.k.a.* classifier – to predict the label of any new input $\boldsymbol{x} \in \mathcal{X}$. To measure the quality of this prediction we use the notion of *misclassification error*, *i.e.* the probability that c does not predict the correct label for a random sample $(\boldsymbol{x}, y) \sim \mathcal{D}$. This probability writes

$$\mathbb{P}_{(\boldsymbol{x},y)\sim\mathcal{D}}[c(\boldsymbol{x}) \neq y] = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathbb{1}\{c(\boldsymbol{x}) \neq y\}]\,. \tag{2.1}$$

[1] See https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html.

Given any ground-truth probability distribution $\mathcal{D}$, one can easily verify that the optimal prediction function on $\mathcal{X} \times \mathcal{Y}$ writes

$$c^{\text{opt}} : \boldsymbol{x} \mapsto \underset{k \in [K]}{\operatorname{argmax}} \, \mathbb{P}_{y \sim \mathcal{D}}[y = k \mid \boldsymbol{x}]^{2}. \tag{2.2}$$

This function is called the Bayes optimal classifier [44, Chap. 2]. It is optimal in the sense that no other classifier $c : \mathcal{X} \to \mathcal{Y}$ can have a lower probability of misclassification on $\mathcal{D}$.

In practice the learner does not have access to the ground-truth distribution $\mathcal{D}$; hence it cannot know the Bayes optimal classifier. Its objective is then to design a learning procedure that finds a prediction function with misclassification error as close as possible from c^{opt}. To do so, the usual strategy in machine learning is to define a set of functions $\mathcal{H} \subset \{\boldsymbol{h} : \mathcal{X} \to \mathbb{R}^K\}$ that will mimic the behavior of $\mathbb{P}_{y \sim \mathcal{D}}[y \mid \boldsymbol{x}]$. This set is also known as the *hypothesis class*. For any hypothesis function $\boldsymbol{h} \in \mathcal{H}$, by analogy with the Bayes optimal classifier, the prediction function writes

$$c : \boldsymbol{x} \mapsto \underset{k \in [K]}{\operatorname{argmax}} \, \boldsymbol{h}_k(\boldsymbol{x}) \,, \tag{2.3}$$

where $\boldsymbol{h}_k(\boldsymbol{x})$ is the kth element of the vector $\boldsymbol{h}(\boldsymbol{x}) = [\boldsymbol{h}_1(\boldsymbol{x}), \dots, \boldsymbol{h}_K(\boldsymbol{x})]^\intercal$.

To select the best hypothesis out of $\mathcal{H}$, the learner uses a surrogate notion of misclassification error called the *risk* or the *generalization error*. The key component on which the risk relies is $\mathcal{L} : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}$ the loss function. It measures how well $\boldsymbol{h}$ fits the ground-truth distribution on a given sample $(\boldsymbol{x}, y) \sim \mathcal{D}$. Accordingly, the learner's objective is to find the hypothesis $\boldsymbol{h}^* \in \mathcal{H}$ that minimizes the expected value of the loss function over $\mathcal{D}$. The risk minimization problem then writes

$$\inf_{\boldsymbol{h} \in \mathcal{H}} \mathcal{R}(\boldsymbol{h}) \text{ with } \mathcal{R}(\boldsymbol{h}) := \mathbb{E}_{(\boldsymbol{x},y) \sim \mathcal{D}}[\mathcal{L}(\boldsymbol{h}(\boldsymbol{x}), y)] \,. \tag{1.1}$$

If $\mathcal{L}$ and $\mathcal{H}$ are well chosen, a solution to the above optimization problem gives a classifier with small misclassification error. For example, if we use the $0/1$ loss

$$\mathcal{L}_{0/1}(\boldsymbol{h}(\boldsymbol{x}), y) := \mathbb{1}\left\{ \underset{k \in [K]}{\operatorname{argmax}} \, \boldsymbol{h}_k(\boldsymbol{x}) \neq y \right\}, \tag{2.4}$$

then Equation (1.1) directly amounts to seek a hypothesis $\boldsymbol{h}^*$ in $\mathcal{H}$ minimizing the probability of misclassification of c^* – the classifier associated to $\boldsymbol{h}^*$. The $0/1$ loss is mostly used to analyze the problem theoretically. Since the indicator function is not differentiable everywhere, for optimization purposes, the community often studies surrogate loss functions instead [9] – *a.k.a. classification calibrated* losses. Under smoothness and convexity assumptions, we can also prove that Problem (1.1) minimizes the misclassification error of c^*.

Since the loss function is not a problem, the main objective the learner has is to design right class of hypotheses $\mathcal{H}$ to search on. On the one hand, if we take a sufficiently rich set of hypotheses, it could contain the optimal $\boldsymbol{h}^{\text{opt}}$ such that $\operatorname{argmax}_k \boldsymbol{h}_k^{\text{opt}}(\boldsymbol{x}) = c^{\text{opt}}(\boldsymbol{x})$. But when the hypothesis class becomes too complex, the learning process is much more difficult to manage – in terms of

[2] Note that argmax operator might output a set. In this case $c^{\text{opt}}(\boldsymbol{x})$ can be any element this set. Here we suppose that there is a unique maximum for simplicity.

optimization. More generally, the choice of $\mathcal{H}$ is subject to a trade-off between estimation and approximation errors.

An important special case: binary classification

To study classification from a theoretical standpoint, it is often easier to consider the *binary classification* setting – *i.e.* $K = 2$. In this context, it is standard to consider a setup slightly different from the above. We still consider $\mathcal{X} = [0,1]^d$, but the output space is now $\mathcal{Y} = \{-1, 1\}$. Furthermore the hypothesis space only considers real-valued functions $h : \mathcal{X} \to \mathbb{R}$, and we also adjust the definition of the classifier $c(\boldsymbol{x}) := \operatorname{sign}(h(\boldsymbol{x}))$. All other notions adapt accordingly. Our result in terms of learning theory are based on the K-class classification setting; hence we keep presenting results in this setting.

2.1.2 The estimation/approximation trade-off

Let $\boldsymbol{h} \in \mathcal{H}$ be any hypothesis function, the *excess risk* of $\boldsymbol{h}$ is the difference between $\mathcal{R}(\boldsymbol{h})$ and the optimal risk $\mathcal{R}(\boldsymbol{h}^{\text{opt}})$. It can decompose into two error types, namely the *estimation error* and the *approximation error*

$$\mathcal{R}(\boldsymbol{h}) - \mathcal{R}(\boldsymbol{h}^{\text{opt}}) = \underbrace{\mathcal{R}(\boldsymbol{h}) - \inf_{\boldsymbol{h} \in \mathcal{H}} \mathcal{R}(\boldsymbol{h})}_{\text{estimation}} + \underbrace{\inf_{\boldsymbol{h} \in \mathcal{H}} \mathcal{R}(\boldsymbol{h}) - \mathcal{R}(\boldsymbol{h}^{\text{opt}})}_{\text{approximation}}. \tag{2.5}$$

On one hand, the estimation error represents the difference between the minimal error we could get in $\mathcal{H}$ and the actual error we have by using $\boldsymbol{h}$. If the risk minimization problem on $\mathcal{H}$ admits a solution $\boldsymbol{h}^*$, the estimation error measures how well $\boldsymbol{h}$ estimates $\boldsymbol{h}^*$. On the other hand, the approximation error represents the minimal excess risk a hypothesis in $\mathcal{H}$ can achieve. It measures the amount of risk that is solely determined by the choice of the hypothesis class $\mathcal{H}$. This error does not depend on the optimization procedure the learner uses. In that sens, it can be seen as a notion of richness of the hypothesis class. When we enlarge $\mathcal{H}$ the approximation error will drop. Unfortunately, enlarging the hypothesis class will also increase the estimation error.

Figure 2.1 illustrates this phenomenon for two nested hypothesis classes $\mathcal{H}_1 \subset \mathcal{H}_2$. Let us suppose that the there exist $\boldsymbol{h}_1^*$, and $\boldsymbol{h}_2^*$ solutions of the risk minimization problems respectively on $\mathcal{H}_1$ and $\mathcal{H}_2$. If we fix $\boldsymbol{h}$ and make the hypothesis set grow from $\mathcal{H}_1$ to $\mathcal{H}_2$, the estimation error grows but the approximation error diminishes. The field of statistical learning theory studies this trade-off by designing hypothesis classes that have small approximation error – rich enough – while maintaining reasonable estimation error – not too complex. Note that the approximation error is very difficult to evaluate because we do not have access to the ground-truth distribution. Conversely, there is some learning procedure such as the *empirical risk minimization* for which we can estimate the approximation error.

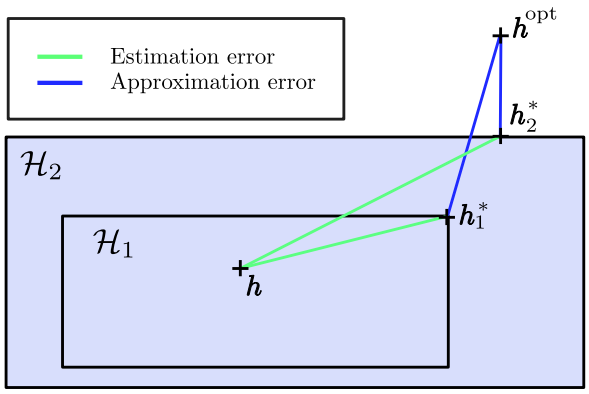


Figure 2.1: Evolution of the approximation and estimation error for a fixed hypothesis $\boldsymbol{h}$ and two nested hypothesis classes $\mathcal{H}_1$ and $\mathcal{H}_2$.

2.1.3 Empirical risk minimization and generalization gap

Empirical risk minimization – ERM – is the most popular learning procedures in machine learning. In a nutshell, the idea is to replace the true risk by the average error over the training sample $\mathcal{S}$ – *a.k.a.* the *empirical risk*. Then, to find an approximate solution for the risk minimization problem – Problem (1.1), a learning algorithm maps the n training examples to a hypothesis by solving the following optimization problem

$$\inf_{\boldsymbol{h}\in\mathcal{H}} \mathcal{R}_{\mathcal{S}}(\boldsymbol{h}) \text{ with } \mathcal{R}_{\mathcal{S}}(\boldsymbol{h}) := \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x_i}), y_i)\,. \tag{2.6}$$

Intuitively, if we have enough training samples[3], the empirical risk of a hypothesis $\mathcal{R}_{\mathcal{S}}(\boldsymbol{h})$ is a good approximation for its true risk $\mathcal{R}(\boldsymbol{h})$. Then, a hypothesis $\boldsymbol{h}_{\mathcal{S}}$ that minimizes the empirical risk also minimizes the risk – or has risk close to the minimum – on $\mathcal{H}$. More formally, we can bound the estimation error of the ERM as follows[4]

$$\begin{aligned}
\mathcal{R}(\boldsymbol{h}_{\mathcal{S}}) - \mathcal{R}(\boldsymbol{h}^*) &= \mathcal{R}(\boldsymbol{h}_{\mathcal{S}}) - \mathcal{R}_{\mathcal{S}}(\boldsymbol{h}_{\mathcal{S}}) + \mathcal{R}_{\mathcal{S}}(\boldsymbol{h}_{\mathcal{S}}) - \mathcal{R}(\boldsymbol{h}^*) && (2.7)\\
&\leq \mathcal{R}(\boldsymbol{h}_{\mathcal{S}}) - \mathcal{R}_{\mathcal{S}}(\boldsymbol{h}_{\mathcal{S}}) + \mathcal{R}_{\mathcal{S}}(\boldsymbol{h}^*) - \mathcal{R}(\boldsymbol{h}^*) && (2.8)\\
&\leq 2\sup_{\boldsymbol{h}\in\mathcal{H}} |\mathcal{R}(\boldsymbol{h}) - \mathcal{R}_{\mathcal{S}}(\boldsymbol{h})|. && (2.9)
\end{aligned}$$

Thanks to the above inequality, we can control the estimation error if we bound the for all $\boldsymbol{h} \in \mathcal{H}$ the difference between the risk and the empirical risk of $\boldsymbol{h}$. This difference is called *generalization gap* and can generally be characterized according to the complexity of $\mathcal{H}$ and the size of the training sample n.

[3] This holds thanks to the uniform law of large numbers.

[4] We suppose that the risk minimization problem has a solution $\boldsymbol{h}^*$ for simplicity. Similar results hold in the general case considering approximate solutions.

In most classical settings, $\mathcal{H}$ is an infinite dimensional space, which makes the complexity analysis difficult. To measure the size of the hypothesis set anyway, the learning theory community [113, 145] uses different complexity notions. Among them, the *empirical Rademacher complexity* is particularly useful to obtain quality bounds for complex classes such as neural networks, conversely to combinatorial notions such as the *VC dimension* [10].

Definition 1 (Rademacher complexity)**.** *For any function class $\mathcal{F} := \{(\boldsymbol{x}, y) \mapsto \mathbb{R}\}$, given a sample $\mathcal{S} = \{(\boldsymbol{x_1}, y_1), \dots, (\boldsymbol{x_n}, y_n)\}$, the* empirical Rademacher complexity *is defined as*

$$\mathfrak{R}_{\mathcal{S}}(\mathcal{F}) := \frac{1}{n}\,\mathbb{E}_{r_i}\left[\sup_{f\in\mathcal{F}} \sum_{i=1}^{n} r_i f(\boldsymbol{x_i}, y_i)\right],$$

where r_i are i.i.d. *drawn from a Rademacher measure* – i.e. $\mathbb{P}(r_i = 1) = \mathbb{P}(r_i = -1) = \frac{1}{2}$.

The empirical Rademacher complexity measures the uniform convergence rate of the empirical risk toward the risk on the class of function $\mathcal{F}$. Thanks to this notion of complexity – under regularity assumption on the loss function – we can bound with high probability the generalization gap of any hypothesis $\boldsymbol{h}$ in a class $\mathcal{H}$.

Theorem 1 ([10, 113])**.** *Let $\mathcal{H}$ be a hypothesis class and $\mathcal{L} : \mathbb{R}^K \times \mathcal{Y} \to [0, L]$. We denote $\mathcal{L}_{\mathcal{H}} := \{(\boldsymbol{x}, y) \mapsto \mathcal{L}(\boldsymbol{h}(\boldsymbol{x}), y)$ s.t. $\boldsymbol{h} \in \mathcal{H}\}$ the set of functions that compose the loss function with a hypothesis. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following holds for any $\boldsymbol{h} \in \mathcal{H}$,*

$$\mathcal{R}(\boldsymbol{h}) - \mathcal{R}_{\mathcal{S}}(\boldsymbol{h}) \leq 2L\,\mathfrak{R}_{\mathcal{S}}(\mathcal{L}_{\mathcal{H}}) + 3L\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

In particular, when $\mathcal{H}$ admits a reasonable Rademacher complexity one can bound the generalization gap of any $\boldsymbol{h} \in \mathcal{H}$ by $O\left(\frac{1}{\sqrt{n}}\right)$ with high probability. This means that, when the training sample is sufficiently large, the ERM gives a solution with risk close to the optimal on $\mathcal{H}$. Note, however, that the ERM will only work if the class is already well chosen. In fact, if $\mathcal{H}$ is not complex enough, the approximation error can be very large. Conversely if $\mathcal{H}$ is too large, the limit of the estimation error becomes loose. But since the approximation error can not be evaluated, how can we select a good $\mathcal{H}$?

2.1.4 Structural risk minimization

One way to look at the hypothesis class selection problem is through the structural risk minimization – SRM. Let us start by taking a hypothesis class $\mathcal{H}$ with very small – or no – approximation error. $\mathcal{H}$ will surely be too rich for the above generalization bounds to make sens. But the rational behind the SRM is to decompose $\mathcal{H}$ as the union of an increasing – in the sens of the inclusion – sequence of subclasses $\mathcal{H} = \bigcup_{m\geq 1} \mathcal{H}_m$. In theory, the problem then consists of selecting the parameter m^* that offers the best trade-off between estimation and approximation errors. Since this quantities are not know, we keep track of the trade-off with an upper bound on the excess risk – *e.g.* by using the generalization gap of the elements in $\mathcal{H}_m$. Figure 2.2 summarizes the evolution of the two error types according to the growing complexity of the hypothesis class – characterized

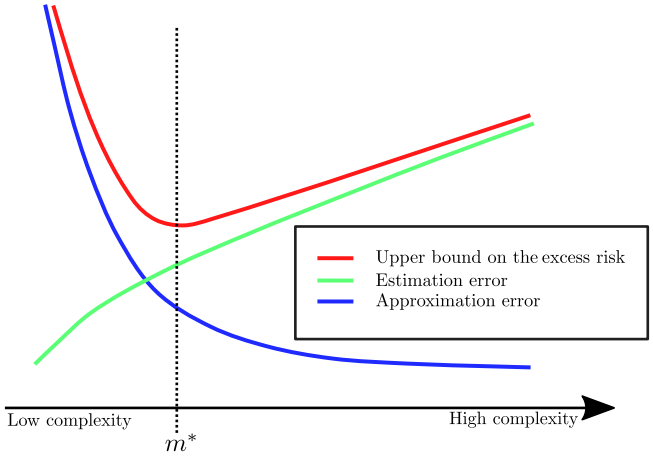


Figure 2.2: Trade-off between the approximation and estimation errors according to the complexity of the hypothesis class.

by m. When the hypothesis class is small it leads to good estimation but high approximation error. Furthermore, enlarging this class may decrease the approximation but also increase the estimation error. m^* represents the best trade-off we found by using the upper bound on the excess risk. In general, we write the structural risk minimization as follows

$$\inf_{m \geq 1} \inf_{\boldsymbol{h} \in \mathcal{H}_m} \mathcal{R}_{\mathcal{S}}(\boldsymbol{h}) + \Omega(\mathcal{H}_m), \tag{2.10}$$

where Ω is a penalty term on the size of the class $\mathcal{H}_m$. This reformulation of the problem allows to revisit the approximation/estimation trade-off using the generalization error and the empirical error. Figure 2.3 illustrates the evolution of the generalization error and the empirical rik, with

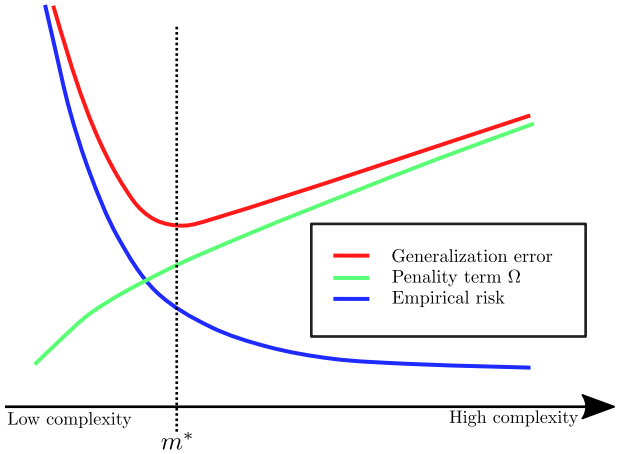


Figure 2.3: Reinterpretation of the approximation/estimation trade-off using the generalization error and the empirical risk – for the SRM.

respect to the complexity of the hypothesis class and the penalization term for the SRM. When

the complexity of the model – m – increases, the training error decreases while the penalty term increases. The generalization error follows the same kind of behavior as the upper bound for the excess risk in Figure 2.3. Therefore, the SRM selects the model that minimizes the generalization error. The SRM provides valuable insights on the links between complexity of the model and generalization bounds. Note however that – in general – the SRM is computationally intractable. In fact, in most hypothesis classes, finding the ERM is already hard and the SRM demands to compute the ERM over a large number of different hypothesis sets. Nevertheless, there exists several workarounds to perform model selection at a lower cost such as cross-validation, or regularization based algorithms [113, Chap. 4]. In this thesis, we focus on analyzing fixed hypothesis classes in the context of adversarial classification; hence we do not discuss model selection. The hypothesis classes we study are however quite general; hence we believe it is safe to assume they have small approximation error. Studying adversarial classification through the lens of the structural risk minimization would be an interesting follow up to our work.

Let us end this section with some more practical considerations by presenting the hypothesis classes we will consider in practice, and some benchmark datasets.

2.1.5 Some more practical considerations: hypothesis classes and datasets

Some remarkable hypothesis classes

One of the first hypothesis classes one should think of when considering a classification problem is the class of linear hypotheses. It writes

$$\mathcal{H} := \{\boldsymbol{x} \mapsto \boldsymbol{h}(\boldsymbol{x}) := \boldsymbol{\theta}^\intercal \boldsymbol{x} \; s.t. \; \boldsymbol{\theta} \in \Theta \subset \mathcal{M}_{d\times K}(\mathbb{R})\}\,. \tag{2.11}$$

The machine learning community often uses this hypothesis class on simple datasets, or to build intuitions on the difficulty of the task. However, for involved applications such as image classification, linear classifiers are too simple to correctly capture the ground-truth distribution. Therefore, one usually uses neural networks instead. A typical class of neural networks is a composition of N – usually non-linear – parametric functions $\boldsymbol{h}^{(i)}_{\boldsymbol{\theta_i}}$ with respective parameter dimensions d_i

$$\mathcal{H} := \{\boldsymbol{x} \mapsto \boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}) := \boldsymbol{h}^{(N)}_{\boldsymbol{\theta_N}} \circ \cdots \circ \boldsymbol{h}^{(1)}_{\boldsymbol{\theta_1}}(\boldsymbol{x}) \; s.t. \; \forall i \in [N],\, \boldsymbol{\theta_i} \in \Theta_i \subset \mathbb{R}^{d_i}\}. \tag{2.12}$$

These classes are characterized by two features, namely their *architecture* and *parameter sets*.

- *The architecture of the model.* The *architecture* is the pre-defined structure materialized by the set of parametric functions $\left\{\boldsymbol{h}^{(1)}_{\boldsymbol{\theta_1}}, \ldots, \boldsymbol{h}^{(N)}_{\boldsymbol{\theta_N}}\right\}$ – *a.k.a.* the *layers* of the network. Depending on the architecture, a neural network sometimes output a vector in the simplex $\Delta(K) := \{\boldsymbol{z} \in \mathbb{R}^K \; s.t. \; \sum_{k=1}^K \boldsymbol{z}_k = 1\}$ – called *probit* vector– or in $\mathbb{R}^K$ without further assumptions – called *logit* vector. In the following, unless stated otherwise we always assume that a neural network gives arbitrary outputs in $\mathbb{R}^K$, *i.e.* logits.

- *The parameter sets.* The parameters materialized by real valued sets $\Theta := \{\Theta_1, \ldots, \Theta_N\}$ on which the learner optimizes to select a hypothesis.

To select a hypothesis from these classes – be it a linear classifier or a neural network, we still need to solve the ERM. Since the classes are parametrized, it simplifies as follows

$$\inf_{\boldsymbol{\theta}\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\boldsymbol{h_\theta}(\boldsymbol{x_i}), y_i)\,, \tag{2.13}$$

where Θ is the set of parameters at hand. For this problem, the loss function $\mathcal{L}$ we select as well as the optimization procedure we use are called the *hyper-parameter* of the model. In the remaining, unless stated otherwise – *e.g.* when we look at the $0/1$ loss, we always consider the mapping $(\boldsymbol{\theta}, \boldsymbol{x}, y) \mapsto \mathcal{L}(\boldsymbol{h_\theta}(\boldsymbol{x}), y)$ to be differentiable with respect to the parameters $\boldsymbol{\theta}$ and the input $\boldsymbol{x}$ – which is a standard assumption. Furthermore, in all our experiments we will use the cross entropy loss defined as follows

$$\mathcal{L} : (\boldsymbol{z}, y) \mapsto -\sum_{k=1}^{K} \mathbb{1}\{y = k\} \log\left(\frac{\exp(\boldsymbol{z}_k)}{\sum_{j=1}^{K}\exp(\boldsymbol{z}_j)}\right). \tag{2.14}$$

Finally, for a well chosen loss such as the cross-entropy, a simple optimization algorithm – *e.g.* a stochastic gradient descent combined with a back-propagation scheme – is sufficient to obtain a good approximate solution to the empirical risk minimization.

Reading note. *Here we only give some quick notions to fix the terminology. The main purpose of this manuscript is not to discuss loss functions or optimization processes. We rather design new hypothesis classes on which we can use well known optimization schemes. We refer the interested reader to [74] or [145, Part II] for a more complete overview on machine learning and deep learning in practice.*

Image datasets and evaluation procedure

Provided with a dataset, we divide it into a training and a test samples – *a.k.a.* train and test *sets*. We use the training sample $\mathcal{S}$ to select $c_\mathcal{S}$ a candidate classifier – model – and evaluate the performance of $c_\mathcal{S}$ on unseen input-output pairs from the test sample. Naturally, the quality of the model depends on the error it gets on the test set – not the train set, but the difference we observe between the quality of the prediction on the test and the training sample can be considered as an empirical evaluation of the generalization error. When we evaluate the performance of a classifier – be it on the train or the test set, we sometimes use the term *accuracy* instead of error. The accuracy of a classifier is simply the average number of good classifications it makes. Accordingly, the notion of *test-time accuracy* – resp. *train-time accuracy* – denotes the accuracy of the model on the test set – resp. the train set. Let us now present the datasets we will most often refer to in this manuscript.

CIFAR-10 / CIFAR-100 We refer to CIFAR-10 or CIFAR-100 datasets [93] to present numerical intuitions and evaluations. The CIFAR-10 dataset is one of the most used benchmarks to evaluate vision tasks in machine learning and current state-of-the-art models achieve over 0.99 test-time accuracy on this dataset. It consists of 60000 color images of size 32×32 divided into 10 classes – 6000 images per class. There are 50000 training images and 10000 test images. CIFAR-100 is just like CIFAR-10 with 100 classes and only 600 images per class. Accordingly, CIFAR-100

is harder to classify, and current state-of-the-art models achieve around 0.93 test-time accuracy on this dataset. Figure 2.4 presents a sample of images from CIFAR datasets.

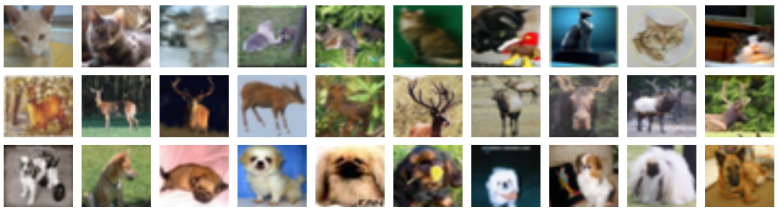


Figure 2.4: Sample of images from CIFAR datasets [93].

Remark 3. *The setting we consider defines the image space $\mathcal{X}$ as a subspace of the hyper-cube $[0,1]^d$. In an image from CIFAR-10/CIFAR-100, each image has 32×32 pixels and 3 channels; hence if we normalize each channel to be within $[0,1]$ – instead of $[0,255]$ – the classification problem for CIFAR-10/CIFAR-100 has dimension $d = 3072$.*

ImageNet To a lesser extent, we sometimes refer to Imagenet dataset [41] to present visual representations. ImageNet is an ongoing project[5] that contributes to building one of the biggest high-quality images database the machine learning community has open access to. It gathers more than 14 million images and over 20000 classes – several hundred images per classes. Dealing with this database calls for huge computational and energy resources. This is why, ImageNet should not be used as a benchmark for new models, but rather to test the scaling of methods that already – provably – work. Figure 2.5 presents a sample of images from ImageNet dataset.

Figure 2.5: Sample of images from Imagenet datasets [41].

2.2 Adversarial attacks, an overview

Adversarial attacks have recently come to light thanks to works studying deep neural networks [20, 67, 152], although it was an existing topic in spam filter analysis [40, 64, 101]. Here we present an overview of the domain in the context of image classification with deep neural networks.

[5] See the project webpage: http://www.image-net.org/

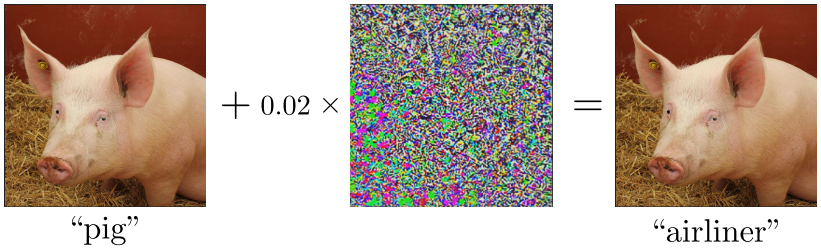


Figure 2.6: Adversarial perturbation of a pig from ImageNet.

2.2.1 A first example

Let us first give a simple example of what an attack looks like. Figure 2.6 illustrates how to design an adversarial example[6] on an image from ImageNet dataset. The original image is a pig – on the left – and a state-of-the-art deep neural network trained on ImageNet recognizes it as such. Then, if we compute a perturbation of the image that forces the network make a mistake, we find the mask in the middle of the figure. To the human eye this mask looks a lot like noise, but it is carefully computed according to the model. If we then multiply this structured perturbation by a small factor and add it to the original pig, we get an image that a human cannot distinguish from the original. That little change is, however, sufficient for the network to classify the new image – on the right – as an airliner. This phenomenon is drawing a lot of attention, and many articles have been published on the vulnerability of neural networks to adversarial attacks [28, 37, 103, 114]. But it is important to note that these vulnerabilities are not restricted to neural networks. Indeed, they apply to essentially any machine learning algorithm [20, 67, 152].

2.2.2 Threat models

We define the adversaries regarding the information they have on the training set, the model architecture, and the parameters. Accordingly, the two main threat models considered in the literature – see *e.g.* [26] – are the following.

- *White box adversary.* In this scenario, the adversary has the same knowledge as the model provider. This means that it has access to the training samples, the architecture and the parameters of the model. Furthermore, the adversary also knows if the model is defended by any pre-or-post processing.
- *Black box adversary.* In this scenario, the adversary has no knowledge about the model, and has only limited oracle access to it – *e.g.* limited queries with access only to the hypothesis outputs' or the predicted classes.

In this work, we only consider the more powerful model threat – *i.e.* white box adversaries. As pointed out by Carlini *et al.* [26], it is not reasonable to assume that the defense algorithm can be held secrets in practice. This concept, called the Kerckhoffs' principle [91], is very common in

[6] To reproduce this example, the interested reader can follow this tutorial: https://adversarial-ml-tutorial.org/introduction/.

the cryptography community. In a nutshell, it says that the only secret on which a cryptography system should hold is the secret of the encryption key. In this work, we apply the same principle and consider that the only secret is the random state of the algorithm – *i.e.* the pseudo-random number generator is unknown from the adversary. Finally, note that considering the white box setting is fully general, since a black box adversary can only be – by definition – less effective than a white box adversary.

2.2.3 On the notions of imperceptibility in high dimension

As we mentioned in Chapter 1, evaluating the imperceptibility of an adversarial example is hard. In practice, we use an ℓ_p norm with $p \in [1, +\infty]$ and a threshold α_p to evaluate an acceptable variation. Accordingly, the set of allowed perturbations for a standard image $\boldsymbol{x} \in \mathcal{X}$ is an ℓ_p ball $B_p(\boldsymbol{x}, \alpha_p) := \{\boldsymbol{x} + \boldsymbol{\tau} \in \mathcal{X} \; s.t. \; \|\boldsymbol{\tau}\|_p \leq \alpha_p\}$. Note that the threshold α_p does not only depend on the norm, it also scales according to the dimension of the problem d. Indeed, if the image has a low resolution, the human eye can easily distinguish the pixels from each other; hence it will be much easier to see changes in this context than in a high quality image. Then the following question arises: given some input space $\mathcal{X}$ with dimension d and $p \in [1, +\infty]$, how should we select α_p for the attacks to remain undetected?

First, one can give an empirical answer to this question for α_∞ [67, 95] – *i.e.* the pixel-wise maximal perturbation that does not change the human perception. Then, to build adversaries with comparable strength, but also because it matches empirical observations[7], we select α_p such that $B_p(\boldsymbol{x}, \alpha_p)$ and $B_\infty(\boldsymbol{x}, \alpha_\infty)$ have equivalent volumes[8]. Typically, on CIFAR datasets, a perturbation $\boldsymbol{\tau}$ is considered imperceptible if $\|\boldsymbol{\tau}\|_\infty \leq 0.031$ or $\|\boldsymbol{\tau}\|_2 \leq 0.8$.

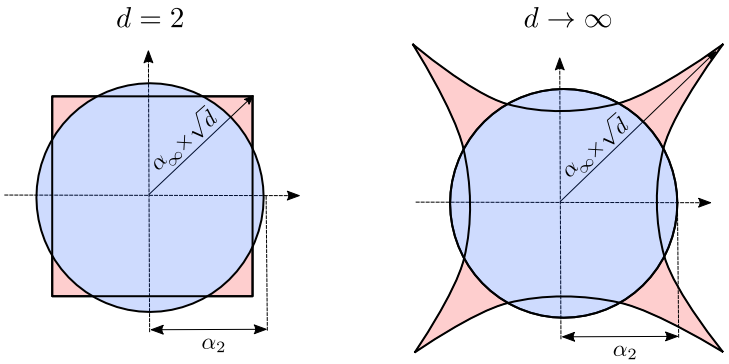


Figure 2.7: Comparison of an ℓ_2 and an ℓ_∞ ball of similar volumes. On the left: $d = 2$. On the right: $d \to \infty$.

Even if the ℓ_p balls have similar volumes – when $\mathcal{X}$ is high dimensional, they will only overlap on a negligible region of the space. Figure 2.7 illustrates this and compares an ℓ_2 and an ℓ_∞ ball of similar volumes when the dimension of the problem – d – increases. In a 2 dimensional space

[7] See *e.g.* https://adversarial-ml-tutorial.org/adversarial_examples/

[8] Simon-Gabriel *et al.* [148] recently hypothesized that we should set $\alpha_p = \alpha_\infty \times d^{1/p}$ instead, but this formula does not match empirical observations as well as equalizing the volumes.

– on the left – the balls overlap on more than 98 percent of their respective volumes, but when the dimension grows – on the right – most of the mass for the ℓ_∞ ball moves toward the corners, leaving only a negligible mass in the intersection. From these different behaviors of the balls in high dimensions, we can draw two conclusions.

- *Low-dimensional intuition can be misleading.* In this work, we will sometimes illustrate our findings with figures. By essence these figures cannot tell the whole story because they fail to render the high-dimensional nature of the problem.

- *Being robust to one adversary does not say much about the others.* Let us suppose that we can design a classifier c that is robust to any ℓ_2 perturbations with maximal radius α_2. Then without a finer grain analysis, c can only ensure protection against ℓ_∞ perturbations of size $\alpha_2/\sqrt{d}$. Therefore, when $\mathcal{X}$ is high dimensional, c does not ensure protection against realistic ℓ_∞ adversaries. Generally, guarantees devised for one adversary will not transfer to other ones. Hence we need to clearly state what adversary we study. In this work, we mainly present some general results, but we focus our empirical analysis on ℓ_2 and ℓ_∞ adversaries.

Reading note. *The interested reader can refer to Appendix A for more detailed discussions on the impossibility to transfer defense strategies for one adversary to another.*

The threat models we just discussed consider that the adversary is constrained to ℓ_p-bounded perturbations – which is the most standard threat model in the literature. Note that these models – based on sufficient conditions for imperceptibility – are too narrow to match real-world threats [26, 62]. Nevertheless, they are mathematically well defined, which facilitates principled analysis and assessments. Furthermore, while ℓ_p threats are not sufficiently realistic, they are part of any more general – realistic – threat model. Thus, building models robust to ℓ_p adversaries – which is still an open question – would allow the community to make a step toward a more general notion of robustness. Hence, in this work, we keep studying ℓ_p-bounded perturbations.

2.2.4 How to build an attack?

Recall the K-class adversarial classification setting with $p \in [1, +\infty]$ and $\alpha_p \geq 0$. Given a hypothesis $\boldsymbol{h} \in \mathcal{H}$ and an input-output pair $(\boldsymbol{x}, y) \sim \mathcal{D}$, the adversary aims to find a solution to the following maximization problem

$$\sup_{\boldsymbol{\tau} \in B_p(\alpha_p)} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x} + \boldsymbol{\tau}), y)\,. \tag{2.15}$$

Two of the most common ways to do so are 1) to try directly solving Problem (2.15) with a projected gradient descent, or 2) to solve a Lagrangian relaxation of the problem.

Remark 4. *Note that – in general – Problem* (2.15) *might not have realizable solutions. However, finding an approximate solution is most of the time sufficient for* c *to misclassify* – i.e. $c(\boldsymbol{x}+\boldsymbol{\tau}) \neq y$. *The attacks we present below are sufficiently strong to make the test-time accuracy of any classical deep neural network drop to* 0 *– on either CIFAR or ImageNet datasets.*

Solving directly with projected gradient descent

In one of the first attack papers, Goodfellow *et al.* [67] presented a simple attack scheme based on the idea that $\boldsymbol{h}$ has a linear behavior. This method is called fast gradient method – FGM – and relies on the idea that a single gradient step – scaled to have an ℓ_p norm smaller than α_p – is sufficient to fool most models. This technique was quickly extended to consider multiple gradient steps [95, 103], and is now known as the projected gradient descent scheme – *a.k.a.* PGD attack. Given a image $\boldsymbol{x}$ to attack, a threshold α_p and a maximal number of steps t_{max}, PGD recursively computes

$$\boldsymbol{x}^{t+1} = \text{proj}_{B_p(\boldsymbol{x},\alpha_p)}\left(\boldsymbol{x}^t + s \underset{v \text{ s.t. } ||v||_p \leq 1}{\text{argmax}} \nabla_{\boldsymbol{x}^t}\mathcal{L}\big(\boldsymbol{h}\big(\boldsymbol{x}^t\big), y\big)^\intercal v\right) \tag{2.16}$$

where, $\nabla_{\boldsymbol{x}^t}$ denotes the gradient with respect to the entry $\boldsymbol{x}^t$, s is a gradient step size, and $\text{proj}_{B_p(\boldsymbol{x},\alpha_p)}$ is the projection operator on $B_p(\boldsymbol{x}, \alpha_p)$[9]. PGD attack has been implemented for several reference norms such as ℓ_∞ or ℓ_2 and is widely used as a state-of-the-art benchmark to evaluate the efficacy of defense strategies [38, 154].

Remark 5. *During training, we usually evaluate the gradient according to the parameters of the model. But one can use the back-propagation algorithm to compute the gradient on the input as well.*

Solving the Lagrangian relaxation

The second procedure searches for the perturbation that has the minimal norm, under the constraint that $\mathcal{L}(\boldsymbol{h}(\boldsymbol{x}+\boldsymbol{\tau}), y)$ is bigger than a parameter κ – typically chosen depending on the loss function $\mathcal{L}$. The associated optimization problem is as follows,

$$\inf_{\boldsymbol{\tau} \text{ s.t. } \mathcal{L}(\boldsymbol{h}(\boldsymbol{x}+\boldsymbol{\tau}),y)\geq\kappa} \|\boldsymbol{\tau}\|_p. \tag{2.17}$$

Problem (2.17) has been studied extensively by Carlini *et al.* [28], resulting in a method called C&W attack. It aims at solving the following Lagrangian relaxation of the problem

$$\inf_{\boldsymbol{\tau}} \|\boldsymbol{\tau}\|_p + \lambda \times g(\boldsymbol{x}+\boldsymbol{\tau}) \tag{2.18}$$

where $g(\boldsymbol{x}+\boldsymbol{\tau}) < 0$ if and only if $\mathcal{L}(\boldsymbol{h}(\boldsymbol{x}+\boldsymbol{\tau}), y) \geq \kappa$. According to the loss function, Carlini *et. al.* use a binary search to optimize the constant κ and a stochastic gradient descent to compute an approximate solution of the problem[10]. The C&W attack is well defined for both $p = 2$ and $p = \infty$. However, empirical observations show a clear gap of efficacy for the ℓ_2-based attack. Accordingly, for this work, we only consider C&W as an ℓ_2 attack.

Remark 6. *Note that when we solve the Lagrangian relaxation, we have no guarantee that the approximate solution will have an α_p bounded norm. To ensure imperceptibility in practice, at the end of the procedure, we force the solution to be in the ℓ_p ball with a projection operator – as in the*

[9] If the projection operator does not exist, any operator that brings $\boldsymbol{x}^{t+1}$ back into the ball can work.
[10] The authors also use a change of variable to ensure that $\boldsymbol{x} + \boldsymbol{\tau} \in \mathcal{X}$

PGD attack. Nevertheless, for a sufficient number of gradient steps, since the goal is to minimize the norm, the solution will usually already be in the appropriate ℓ_p ball.

2.2.5 Discussion on the attack strategies

The literature on attacks is rich, and this section does not provide an exhaustive list of the methods developed so far. We present only two of the most popular attack frameworks. Given an optimization problem, one can use a number of possible algorithms to get an approximate solution to it, which makes the attack literature flourishing. But, at this stage, most attacks are based on solving either one or the other optimization problems we have just introduced. Thus, we believe that the above methods are sufficient to provide a general understanding of how to construct an adversarial example. For a more complete overview of the attack methods one can – for example – refer to [167].

2.3 State-of-the-art on defense strategies

At the moment, most the works that aim to provide robust classifiers do not offer any provable protection against adversarial attacks, as the community has demonstrated on many occasions [6, 27, 38, 154]. However, among the defense strategies, two are susceptible to pass the test of time, namely adversarial training and provable robustness.

2.3.1 Adversarial training

Let us suppose for a moment that we can solve the maximization Problem (2.15). This would for example be the case if Danskin Theorem [17] holds. Then, given a classifier $\boldsymbol{h}$ and a sample $(\boldsymbol{x}, y) \sim \mathcal{D}$, a well-calibrated stochastic gradient descent would find

$$\boldsymbol{\tau}^* = \underset{\boldsymbol{\tau} \text{ s.t. } \|\boldsymbol{\tau}\|_p \leq \alpha}{\operatorname{argmax}} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x} + \boldsymbol{\tau}), y)\,. \tag{2.19}$$

Then – intuitively – a standard training procedure on $\boldsymbol{x} + \boldsymbol{\tau}^*$ instead of $\boldsymbol{x}$ would converge to a robust classifier if it exists. Even-though Danskin Theorem does not hold in practice[11], several works [67, 95, 103] presented a learning procedure called *adversarial training* based on this reasoning. In a nutshell, adversarial training seeks a solution to the empirical adversarial risk minimization – Problem (1.4) – by taking successive stochastic gradient steps on an approximated worst-case perturbation of the clean input. To simulate the worst-case perturbation, the procedure uses an attack method – usually PGD[12]. This solution – inspired by the literature on robust optimization [16] – is intuitive and provides state-of-the-art experimental robustness against the strongest ℓ_∞ attack methods proposed so far [38]. Typically on CIFAR-10, the latest improvement of adversarial training [180] obtains 0.53 test-time accuracy under ℓ_∞ perturbations of size 0.031. However, the main weakness of adversarial training is its lack of formal guarantees. Despite some recent works [149, 180] providing valuable insights, the worst-case adversarial risk of

[11] Danskin Theorem may not hold *e.g.* because given $(\boldsymbol{x}, y)$, $\boldsymbol{\tau} \mapsto \mathcal{L}(\boldsymbol{h}(\boldsymbol{x} + \boldsymbol{\tau}), y)$ will usually not be convex in $\boldsymbol{\tau}$.

[12] Note that most of the time, adversarial training builds attacks by using PGD with reference norm ℓ_∞, even we it wants to defend against other types of attacks [103], with good empirical results.

this method is still unknown. Provable defenses attempt to address this concern by providing an in-depth mathematical analysis with the methods they present.

2.3.2 Provable robustness

The main objective of the literature on provable robustness is to upper-bound the adversary's optimization problem – Problem (2.15). This allows gives worst-case accuracy results, even though $\boldsymbol{h}$ is a complex, non-linear classifier. The two most common methods to obtain provable defenses are 1) to analyze a convex relaxation of the problem and 2) to use randomized smoothing to build more robust classifiers.

Analyzing a convex relaxation of the problem

Given some $\boldsymbol{x} \in \mathcal{X}$, the idea of is to build a convex relaxation of the ball of authorized modifications $B_p(\boldsymbol{x}, \alpha_p)$. To find a good approximation to the inner maximization problem, we should characterize the image of $B_p(\boldsymbol{x}, \alpha_p)$ through $\boldsymbol{h}$ – *i.e.* $\boldsymbol{h}(B_p(\boldsymbol{x}, \alpha_p))$. To simplify the problem, recent works [48, 134, 168, 169] performed a convex relaxation over the image set in the context of neural networks with ReLU non-linearity, and performed robust optimization over this new region.

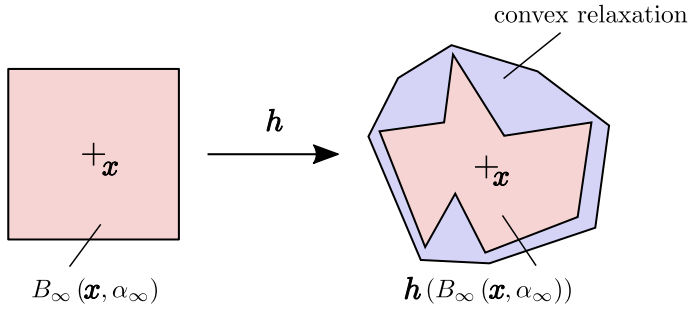


Figure 2.8: Illustration of the convex relaxation technique from [168].

Figure 2.8 illustrates this simplification for an ℓ_∞ adversary. Before applying the hypothesis function $\boldsymbol{h}$ – on the left – the ℓ_∞ ball is convex and easy to study. After applying $\boldsymbol{h}$ – on the right – $\boldsymbol{h}(B_\infty(\boldsymbol{x}, \alpha_\infty))$ is highly non convex. Therefore, to simplify the analysis, one should study the convex relaxation of $\boldsymbol{h}(B_\infty(\boldsymbol{x}, \alpha_\infty))$ instead. The resulting problem is a linear program. By dualizing, we obtain an optimization problem similar to back-propagation and we can draw guarantees for the network. However, this technique involves a linear program and is therefore difficult to apply to high-dimensional datasets; hence hardening its application to image classification.

Randomized Smoothing

Randomized smoothing defenses are randomization based defenses. The idea of provable defense through randomization was first introduced in [98] and refined in [34, 100, 139]. The rationale behind this idea is very simple: take a hypothesis with *probit* outputs $\boldsymbol{h} : \mathcal{X} \to \Delta(K)$, and smooth

it *after training* by convolution with a Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$. Then the robust classifier writes

$$c_{rob} : \boldsymbol{x} \mapsto \underset{k\in[K]}{\operatorname{argmax}} \big(\boldsymbol{h}_k * \mathcal{N}(0, \sigma^2 I)\big)(\boldsymbol{x}) := \underset{k\in[K]}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(0,\sigma^2 I)}[\boldsymbol{h}_k(\boldsymbol{x}+\boldsymbol{z})]\,. \quad (2.20)$$

If we denote Φ the cumulative density function of the standard Gaussian distribution, we can show that $\Phi^{-1}\big(\boldsymbol{h} * \mathcal{N}(0, \sigma^2 I)\big)$ is 1-Lipschitz [139]. Therefore, c_{rob} is robust to adversarial examples that are close enough to the unperturbed input $\boldsymbol{x}$. More precisely, for any point $\boldsymbol{x} \in \mathcal{X}$, we can build a radius around $\boldsymbol{x}$ for which no ℓ_2 adversary can change the decision of c_{rob}. Furthermore, the radius depends on the difference between the two biggest probits of $\hat{\boldsymbol{h}} := \boldsymbol{h} * \mathcal{N}(0, \sigma^2 I)$. Formally, if for any $\boldsymbol{x} \in \mathcal{X}$ we denote $\hat{\boldsymbol{h}}(\boldsymbol{x})_{(1)} \geq \hat{\boldsymbol{h}}(\boldsymbol{x})_{(2)} \geq \cdots \geq \hat{\boldsymbol{h}}(\boldsymbol{x})_{(K)}$ the values of the vector $\hat{\boldsymbol{h}}(\boldsymbol{x})$ in decreasing order, the following hold.

Theorem 2 ([34, 139])**.** *For any $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{\tau} \in \mathcal{X}$ the following hold.*

$$\textit{If } \|\boldsymbol{\tau}\|_2 \leq \frac{1}{2}\Big(\Phi^{-1}\Big(\hat{\boldsymbol{h}}(\boldsymbol{x})_{(1)}\Big) - \Phi^{-1}\Big(\hat{\boldsymbol{h}}(\boldsymbol{x})_{(2)}\Big)\Big), \textit{ then } c_{rob}(\boldsymbol{x}) = c_{rob}(\boldsymbol{x}+\boldsymbol{\tau}).$$

This theorems says that the more separated the probits of the hypothesis, the more robust the classifier is to adversarial perturbations. Then, the model provider can evaluate its worst-case accuracy under attack according to its standard accuracy and the confidence the network has in it predictions. This technique gives provable defense against adversarial examples on *a given dataset*. Table 2.9 presents the current state-of-the-art results in terms of certified accuracy – *i.e.* accuracy that cannot be diminished by an adversary – of randomized smoothing for ℓ_2 based adversaries with different thresholds on CIFAR-10. Note that for a reasonable threshold of 0.75 one gets 0.52 certified accuracy. Following the works investigating Gaussian distributions against ℓ_2 adversaries,

Table 2.9: Certified accuracy of randomized smoothing model [139] on the CIFAR-10 dataset.

ℓ_2 norm of the attack	0.25	0.5	**0.75**	1.0	1.25
Randomized smoothing [139]	0.81	0.63	**0.52**	0.37	0.33

several extensions obtained similar results for other ℓ_p norms [99, 173], or discuss how this method relates to the dimension of the problem [94]. Overall, randomized smoothing presents principled advantages over most previous methods. It is simple to implement and to interpret, computationally efficient and provides state-of-the-art provable robustness for benchmark datasets.

2.3.3 Discussion on the current defense strategies

Over the last few years, there has been significant advances on the robustness of machine learning models to adversarial attacks. However, in terms of the quality of defenses, both provable robustness and adversarial training call for improvements. Indeed, the accuracy under attack of these methods is hardly above 0.5 against imperceptible perturbations on CIFAR-10. These results are not sufficient to consider deploying image recognition systems in real-world applications.

Furthermore, this literature focuses on minimizing the empirical adversarial risk and do not present any generalization guarantee. This question is critical, especially for classification under perturbation. Indeed, Madry *et al.* [103] noticed that the initial version of adversarial training achieves 0.96 train-time adversarial accuracy against 0.47 test-time adversarial accuracy. This gap between train and test performances is significantly larger than what models usually achieve in the standard setting. Hence it is crucial to study generalization guarantees in the adversarial setting to control the generalization gap between training and test errors.

2.4 Adversarial classification through the lens of statistical learning theory

Notations. *By analogy with the standard setting, we denote* $\mathcal{R}^{\text{adv}}(\boldsymbol{h}; \alpha_p)$ *and* $\mathcal{R}^{\text{adv}}_{\mathcal{S}}(\boldsymbol{h}; \alpha_p)$ *the* adversarial risk *and* empirical adversarial risk *of* $\boldsymbol{h}$ *under* α_p*-bounded perturbations*

$$\mathcal{R}^{\text{adv}}(\boldsymbol{h}; \alpha_p) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x}+\boldsymbol{\tau}), y)\right], \tag{2.21}$$

$$\mathcal{R}^{\text{adv}}_{\mathcal{S}}(\boldsymbol{h}; \alpha_p) := \frac{1}{n}\sum_{i=1}^{n} \sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x_i}+\boldsymbol{\tau}), y_i), \tag{2.22}$$

where $B_p(\alpha_p) := \{\tau \in \mathcal{X} \text{ s.t. } \|\boldsymbol{\tau}\|_p \leq \alpha_p\}$.

Unlike other notions such as training set corruptions– *a.k.a.* poisoning attacks [89, 90], the theoretical aspects of adversarial robustness are not widely studied. For now, empirical observations tend to show that 1) adversarial examples on state-of-the-art models are hard to mitigate and 2) robust training methods give poor generalization performances. Some recent works study the problem through the lens of learning theory either to understand the links between robustness and accuracy or to provide bounds on the generalization gap of current learning procedures in the adversarial setting.

2.4.1 Is robustness antagonist with accuracy?

A first line of research [83, 151, 156, 180] suggests that designing robust models might be at odds with standard accuracy. These works study different experimental and theoretical toy settings. Among them, let us start with the toy binary classification task from [156].

Toy Example. *Given any* $(\boldsymbol{x}, y) \sim \mathcal{D}$, $q \in [0, 1)$ *and* $\eta > 0$, *the following holds.*

1. y *is uniformly distributed at random on* $\{-1, 1\}$.
2. *Given* y, $\boldsymbol{x}_1$ *takes value* y *with probability* q *and* $-y$ *otherwise.*
3. *All other elements* $\boldsymbol{x}_2, ..., \boldsymbol{x}_d$ *of the vector* $\boldsymbol{x}$ *are drawn* i.i.d. *from a Gaussian* $\mathcal{N}(\eta y, 1)$.

According to the above distribution, when $\mathcal{X}$ is high dimensional, one can build a simple linear classifier $h(\boldsymbol{x}) = \frac{1}{d-1}\sum_{i=2}^{d} \boldsymbol{x}_i$ that will have arbitrary high test-time accuracy. Indeed – thanks

to the central limit theorem – when $d \to \infty$ we get $h(\boldsymbol{x}) = \frac{1}{d-1}\sum_{i=2}^{d} \boldsymbol{x}_i \to \eta y$, meaning that $\text{sign}(h(\boldsymbol{x})) = y$ with arbitrarily high probability. Nevertheless, an ℓ_∞ adversary that can shift all features by at most $\alpha_\infty = 2\eta$ will be able to make the test-time accuracy of h drop to 0. More generally, on this toy example, Tsipras *et al.* [156] presented the following result.

Theorem 3 ([156])**.** *Any classifier that attains at least* $1 - r$ *standard test-time accuracy on* $\mathcal{D}$ *has robust test-time accuracy at most* $\frac{q}{1-q}r$ *against an* ℓ_∞*-bounded adversary with* $\alpha_\infty \geq 2\eta$*.*

This result proves first-hand that robustness can be at odds with precision. But it is not general enough to draw conclusions – since it is based on a very simple toy distribution. A subsequent work by Zhang *et al.* [180] observed – for the binary classification setting with $0/1$ loss – that the adversarial risk of any hypothesis h straightforwardly decomposes as follows,

$$\mathcal{R}^{\text{adv}}(h; \alpha_p) = \mathcal{R}(h) + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathbb{1}\{c(\boldsymbol{x}) = y \textit{ and } \exists \boldsymbol{\tau} \in B_p(\alpha_p) \textit{ s.t. } c(\boldsymbol{x}+\boldsymbol{\tau}) \neq y\}], \tag{2.23}$$

where $c(\boldsymbol{x}) := \text{sign}(h(\boldsymbol{x}))$. Looking at Equation (2.23), we realize that minimizing the adversarial risk is not enough to ensure good standard accuracy – as one could only optimize over the second term. This indicates that adversarial risk minimization – Problem (1.3) – is harder than standard risk minimization – Problem (1.1). Note, however, that Equation (2.23) does not highlight a fundamental trade-off between robustness and accuracy. Finding such a relation in the general case remains an open question.

2.4.2 Studying adversarial generalization

To further compare the difficulty of the two problems, a recent line of research began to explore the notion of the adversarial generalization gap. In this line, Schmidt *et al.* [141] presented first intuitions by studying a simplified binary classification framework where $\mathcal{D}$ is a mixture of spherical Gaussian distributions. In this framework, the authors show that we only need $O(1)$ training samples to have a small generalization error. But against an ℓ_∞ adversary, we need $O(\sqrt{d})$ training samples instead. In the discussion of their work, the authors present the problem of obtaining similar results without making assumptions about the distribution as an open problem.

This issue was first tackled by Cullina *et al.* [39] by using the VC-dimension. Their analysis shows that for linear classifiers, the VC dimension of the hypothesis class does not change under attack. This work indicates that – with respect to the VC dimension – classification under perturbation is not more difficult than standard classification which does not correspond to the empirical observations and initial intuitions provided earlier [103, 141]. However, as previously mentioned, the Rademacher complexity generally allows for tighter generalization bounds than the VC dimension [8]. Accordingly, further works studied the same problem, using Rademacher complexity and presented the following results relating the adversarial generalization error of linear classifiers[13] with the dimension of the problem[14].

[13] These works also investigate neural networks with one hidden layer – we refer the interested reader to the original papers for more details.

[14] [92] and [175] only present bounds for ℓ_∞ adversaries. [7] extended the results to any ℓ_p attack.

Theorem 4 ([7]). *Let* $\mathcal{H}_q := \left\{ \boldsymbol{x} \mapsto \boldsymbol{\theta}^\intercal \boldsymbol{x} \text{ s.t. } \|\boldsymbol{\theta}\|_q \leq M \right\}$ *and let us suppose that* $\mathcal{L}$ *is* L*-Lipschitz. Then with probability at least* $1-\delta$*, the following holds for any* $h \in \mathcal{H}_q$*,*

$$\mathcal{R}^{\text{adv}}(\boldsymbol{h}; \alpha_p) \leq \mathcal{R}^{\text{adv}}_{\mathcal{S}}(\boldsymbol{h}; \alpha_p) + 2L\,\mathfrak{R}_{\mathcal{S}}(\mathcal{H}_q) + \alpha_p \frac{M}{\sqrt{n}} \max\left(d^{1-\frac{1}{p}-\frac{1}{q}}, 1\right) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

Note that the main difference between this result and the one we presented in the standard setting – Theorem 1 – is the additive factor

$$\alpha_p \frac{M}{\sqrt{n}} \max\left(d^{1-\frac{1}{p}-\frac{1}{q}}, 1\right) = O\left(\frac{d^{1-\frac{1}{p}-\frac{1}{q}}}{\sqrt{n}}\right). \tag{2.24}$$

Therefore, by analyzing the problem using Rademacher complexity, we can show that the adversarial generalization does depend on the dimension of the problem. Hence, in terms of sample complexity, adversarial classification is more difficult than standard classification. However, when facing an ℓ_p adversary, one can always select a class of linear classifiers for which the dimension term disappears. Indeed, if we select q to be the Holder conjugate of p – *i.e.* $\frac{1}{p} + \frac{1}{q} = 1$ – the additive term becomes $O\left(\frac{1}{\sqrt{n}}\right)$. Therefore, we can build strong intuitions for an adversary, but the generalization bounds are not transferable to another ℓ_p adversary.

Reading note. *At first a glance, the difficulty of adversarial generalization seems to contradict previous conclusions on the link between robustness and generalization [171]. But as we will see in Chapter 4, these results are based on very specific assumptions that may not hold in high dimensions.*

2.4.3 Discussion on the learning theory literature

Some compelling insights were presented on whether robustness standard accuracy are in conflict. However, in more general configurations, the question remains open. Moreover, from the different results on the adversarial generalization gap, we can draw two – somewhat contradictory – conclusions. First, learning under perturbation is indeed much more difficult than standard learning and the difficulty increases with the dimension of the problem. Second, for an ℓ_p adversary – fixed p – robustness might be achievable.

Going further, it should be noted that the generalization gap measures only the difference between empirical and theoretical risk. In practice, the empirical adversarial risk is never really estimated – since we cannot compute the exact solution to the inner maximization problem. The following question therefore remains open: even if we can set up a learning procedure with a small generalization gap, will the adversarial risk be low? To answer this question, we need to study the adversarial risk minimization problem – Problem (1.3).

Remark 7. *Another line of research within the learning community studies the problem from a computational viewpoint. This was recently addressed by Bubeck* et al. *[25] who argued that the problem of adversarial classification is not the sample size, but the computational hardness. Thus, even with a reasonable sample size for both problems, we can present a set of learning problems where standard non-robust learning can be performed efficiently, but is difficult to compute in an adversarial setting.*

2.5 Is classification under perturbation feasible?

From preliminary intuitions to advanced mathematical analyses, some works are studying the fundamental properties of classification under perturbation – Problem (1.3). Specifically, the community wonders why adversarial examples exist and whether we can mitigate them. These questions are far from settled, but most works indicate that sensitivity to adversarial perturbations is inevitable. Besides, a very recent line of research began to investigate the worst case adversarial risk of any hypothesis class; thus assessing whether Problem (1.3) is even worth solving.

2.5.1 Initial hypotheses on the existence of adversarial examples

When Szegedy *et al.* [152] first noticed the vulnerability of deep neural networks to small perturbations, they hypothesized that this phenomenon was a consequence of the model's over-fitting. The community uses complex and powerful neural networks that can sometimes be over-parameterized for the task. Thus, even with enough training samples, the network learns structures that are too complicated to describe only the dataset distribution. As a result, it makes random mistakes in low probability regions of the image space that an adversary can exploit.

Figure 2.10 – on the left – illustrates this hypothesis on a training set of three blue crosses and three red circles. It is always possible to build a complex classifier that easily adapts to the training points. But since it has much more parameters than it needs, it also creates small classification areas in low probability regions – somewhat randomly. One can then easily see that a small shift of a point in a well-chosen direction causes an error in the classifier.

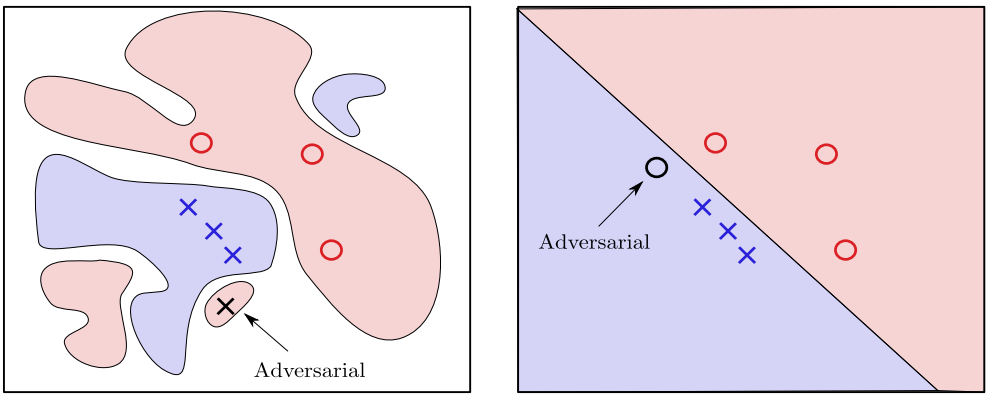


Figure 2.10: On the left: adversarial examples for a complicated over-fitting network. On the right: adversarial examples for a linear under-fitting classifier.

This theory was then invalidated by Goodfellow *et al.* [67] with the following argument. If over-fitting was the main reason, then adversarial examples would be more or less artifacts of the learning procedure, and should be unique to the classifier. Therefore, if we fit the model again, or if we fit a slightly different model, we should get different adversarial examples. But the authors found that different models misclassify the same examples; thus invalidating the initial theory [15].

[15] This phenomenon was later called attack transferability [155]

Goodfellow *et al.* [67] further hypothesized that adversarial examples are more a matter of under-fitting than over-fitting. Specifically, the authors argued that deep neural networks – despite the use of non-linear transformations – have a linear behavior that make them vulnerable to attacks in high-dimensional spaces.

To better understand how under-fitting can lead to a vulnerability, let us go back to Figure 2.10 – on the right. If we fit a linear model to the previous training set, we get a hyperplane lying between the two sets of points. However, this hyperplane does not account for the distribution of the dataset. The circles could be arranged in a C-shape, so moving a point along this shape causes the classifier to make an error. Some follow-up work kept linking the vulnerability of the models to the shape of their decision boundary. For example, Moosavi *et al.* [115] related the vulnerability of a classifier to the curvature of its decision boundary. As such, the shape of the decision boundary is not sufficient to explain the whole phenomenon, but it seems to play an important role as some very efficient attack methods extensively use this hypothesis [67, 114, 115].

2.5.2 Are adversarial examples inevitable?

To further investigate whether adversarial examples are inevitable, subsequent works [63, 144] has focused their analysis on the task – rather than the classifier itself. Consider for example K-class classification on the unit sphere – *i.e.* $\mathcal{X} = \mathbb{S}^{n-1} := \{\boldsymbol{x} \in \mathbb{R}^d \text{ s.t. } \|\boldsymbol{x}\|_2 = 1\}$. In this context, Shafahi *et al.* [144] used isoperimetric inequalities [22] to argue that adversarial examples are inevitable. The authors show that – under assumptions on the concentration of the ground-truth distribution – for any classifier on the unit sphere, there is at least one class $k^* \in \mathcal{Y}$ for which adversarial examples exist with high probability.

Theorem 5 ([144])**.** *Let ν defines the probability distribution for y, μ_k is the conditional distribution for $\boldsymbol{x}$ knowing $y = k$ and g_k its probability density function. Let us also consider c a classifier over the unit sphere $\mathcal{X}$ and α_p a perception threshold for an ℓ_p adversary. Then there exists $k^* \in \mathcal{Y}$ such that for any $\boldsymbol{x} \sim \mu_{k^*}$, with probability at least*

$$1 - V_{k^*}\left(\frac{\pi}{8}\right)^{1/2} \exp\left(-\frac{d-1}{2}(\alpha_p)^2\right),$$

there exist $\boldsymbol{\tau} \in B_p(\boldsymbol{x}, \alpha_p)$ such that $c(\boldsymbol{x} + \boldsymbol{\tau}) \neq k^*$. Where $V_{k^*} := \sup_{\boldsymbol{x} \in \mathcal{X}} g_{k^*}(\boldsymbol{x}) \times s_{n-1}$.

This result means that when the conditional distribution μ_{k^*} has limited concentration, no classifier can be robust to ℓ_p[16] adversaries targeting samples from class k^*. Gilmer *et al.* [63] presented observations of the same nature by studying a toy dataset consisting of two concentric spheres. Their main result relates the errors in the standard and the adversarial setting by saying that even a small standard error on their toy dataset translates to a large adversarial error. These results were later presented in a more general way by Dohmatob [46], but for simplicity we discussed here the initial contributions.

Remark 8. *Note that Shafahi* et al. *[144] tried to extend their conclusions to image classification –* i.e. *when $\mathcal{X} = [0, 1]^d$ and d is large. However, in this context, the probability is high only when the*

[16] The initial result in [144] uses the geodesic distance. Hence the result holds at least for ℓ_2 and ℓ_∞ adversaries.

perturbation threshold α_p is large – hence losing the imperceptibility of the attack. Dohmatob [46] has conducted complementary experiments on – small-scale – image datasets with similar results; the number of adversarial examples is not prohibitive as long as α_p is small. Therefore, the above results still need to be verified on large-scale image classification.

This literature suggests that adversarial examples are inevitable, which means that Problem (1.3) can have a large value. In the following section, we present some works that attempt to assess whether solving Problem (1.3) is even worth trying by estimating this value.

2.5.3 Finding worst case lower bounds on the adversarial risk minimization

Two recent works [18, 131] studied adversarial risk minimization by using arguments from optimal transport [163]. They show how to characterize the adversarial risk for binary classification by an optimal transport cost between the conditional probability distributions of the two classes μ_1 and μ_{-1}. Let us consider the adversary's problem from a distributional point of view. Instead of attacking every point, it directly moves the distributions μ_1 and μ_{-1} to maximize the risk with respect to $\mathcal{D}$. In this context, we can evaluate the worst-case – non-normalized – accuracy under attack of any classifier by the minimal number of points that are not susceptible to be switched from one conditional distribution to the other

$$D_{\alpha_p}(\mu_1, \mu_{-1}) = \inf_{\pi \in \Pi(\mu_1, \mu_{-1})} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{x}') \sim \pi} \Big[\mathbb{1}\{ \big\| \boldsymbol{x} - \boldsymbol{x}' \big\|_p > \alpha_p \} \Big], \tag{2.25}$$

where $\Pi(\mu_1, \mu_{-1})$ is the set of all joint probability measures on $\mathcal{X} \times \mathcal{X}$ with marginals μ_1 and μ_{-1}. Finally, we can define the best-case adversarial risk according to the non-normalized worst-case accuracy under attack $D_{\alpha_p}(\mu_1, \mu_{-1})$ and the classes' distribution ν.

Theorem 6 ([18, 131])**.** *Let ν consider the probability distribution for y with $\nu(1) = \nu(-1) = 1/2$. Then the following holds,*

$$\inf_{h \in \mathcal{H}} \mathcal{R}^{\text{adv}}(h; \alpha_p) = \frac{1}{2}\big(1 - D_{\alpha_p}(\mu_1, \mu_{-1})\big).$$

This result indicates that if the conditional distributions are close enough – according to the above notion of distance – then the adversarial risk will be high, regardless of the classifier. Note, however, that this is already the case without adversaries. Indeed, the risk of the Bayes optimal classifier h^{opt} – *a.k.a.* the Bayes optimal risk – is as follows

$$\mathcal{R}(h^{\text{opt}}) = \frac{1}{2}(1 - D_{TV}(\mu_1, \mu_{-1})), \tag{2.26}$$

where $D_{TV}(\mu_1, \mu_{-1})$ is the total variation distance between the conditional distributions. Hence remains the question: how fast does $D_{\alpha_p}(\mu_1, \mu_{-1})$ grow – compared to $D_{TV}(\mu_1, \mu_{-1})$ – according to the distribution? To answer this, Pydi *et al.* [131] evaluated their bounds on – a smoothed version – of CIFAR-10 dataset. Their preliminary results indicate that best-case adversarial risk – for $\alpha_2 \leq 0.8$ – can be 0.05 bigger than the Bayes optimal risk, which is not prohibitive but still represents an important loss of accuracy. For the gap to be more important – as we already

pointed out in the previous section – we should select a larger α_p; hence the imperceptibility of the attack becomes questionable.

Remark 9. *Given an n points dataset, to build the smooth version of it, the authors use a mixture of n Gaussian distributions, where each mean is placed on a data point. This smoothness assumption seems reasonable, since machine learning practitioners often use Gaussian data augmentation to improve the performances of their models [66, 128].*

2.5.4 Discussion on the feasibility of classification under perturbation

The results we just presented seem to show that adversarial examples are – to some extent – inevitable and that accuracy under attack of a classifier will always be significantly smaller than its standard counterpart. However, these results – even if the authors sometimes claim the opposite – do not prove that no satisfactory solution can be found to the adversarial risk minimization – Problem (1.3). For example, on CIFAR-10, diminishing the current state-of-the-art by 0.05 still gives a good classifier with over 0.9 test-time accuracy[17]. Furthermore, as recently pointed out by Dohmatob [46], the strength of these impossibility results might mainly come from the fact that the adversary we consider is unrealistically strong. Thus, rethinking the constraints for the adversary could question the ongoing consensus on whether we can build models that are both robust and accurate.

2.6 Our positioning with regard to prior art

In this thesis, we aim to bring a new understanding on adversarial examples and contribute to the development of new technical tools. Our main accomplishments are the followings.

Bringing a new point of view on the adversarial classification problem

We begin by revisiting the problem of adversarial risk minimization by regularizing the adversary objective function. We study this new problem through the lens of game theory by casting it as an infinite zero sum game. Our conclusions highlight a very interesting property of the adversarial classification problem, which is its instability – *i.e.* the nature of the game between the adversary and the classifier changes completely when we add a small regularization term. This leads us to question current theses on adversarial classification and to ask whether existing conclusions would still hold if we limit the adversary's strength. The game theoretical point of view we develop also leads us to study randomized classifiers – *i.e.* classifiers that produce random variables. In particular, we show that they have principled advantages over deterministic classifiers in terms of robustness to adversarial perturbations.

Some works have tackled the problem of adversarial examples as a two player game before [127, 138], but they consider restricted versions of the game – *e.g.* when the players only have a finite set of possible strategies. We study a more general setting which allows us to build strong insights on the fundamental nature of the game between the classifier and the adversary.

[17] Still, it remains to find this optimal classifier, which can be hard as discussed in Section 2.4

Studying adversarial defense from a probabilistic point of view

Based on the new insights we develop on adversarial classification, we present a theoretical analysis of randomized classifiers. To do so, we first define a new notion of robustness for these classifiers using probability metrics. Then, we show that under robustness assumptions, we can limit the difference between the standard risk and the adversarial risk of a randomized classifier. This result is important for the community because it shows that a well-chosen class of hypotheses can give both reasonable robustness and accuracy – thus mitigating the previous results on deterministic classifiers. We then devise bounds on the standard generalization gap of this new hypothesis class. This result encompasses existing works on the link between robustness and accuracy for deterministic classifiers. Finally we analyze the stability of the classifier's mode allowing us to present a probabilistic point of view on randomized smoothing techniques. Our point of view on randomization as a defense strategy could pave the way to further investigating randomized smoothing from a theoretical perspective.

Building robust classifiers

Finally, we link our definition of robustness to the notion of differential privacy. Thanks to this connection, we bring to the community a new set of technical tools. As a consequence, we design new noise injection schemes to build robust classes of randomized classifiers. These schemes prove that the theoretical analysis we have previously built is applicable to state-of-the-art image classification models.

Injecting noise into algorithms to improve train time robustness has been used for ages in detection and signal processing tasks [29, 71, 111, 181]. It has also been extensively studied in several machine learning and optimization fields – *e.g.* robust optimization [16] and data augmentation techniques [128]. Concurrently to our work, noise injection techniques have been adopted by the adversarial defense community [45, 170]. In particular, Lecuyer *et al.* [98] first developed randomized smoothing, by using theoretical results from differential privacy. Our work belongs to the same line of research, but the nature of our results is different. While randomized smoothing focuses on the construction of certified defenses, depending on the dataset and the classifier used, we study randomized mechanisms from the perspective of information theory and learning theory. Our analysis presents the fundamental properties of randomized defenses, including – but not limited to – randomized smoothing.

3 A game theoretical point of view on adversarial attacks

Contents

Q1: *Is there some hypothesis class $\mathcal{H}$ for which the adversarial risk minimization problem has a solution $\boldsymbol{h}^*$ with small adversarial risk?*

In this chapter, we aim to answer **Q1** by adopting a game-theoretical point of view. We present in Section 3.1 the adversarial attack and defense problem as an infinite zero-sum game. Then we discuss how unrealistic threat models might impact the analysis of this game and present a simple additional constraint to mitigate the overpower of the adversary. We demonstrate in Section 3.2 that – as long as this small constraint holds – no Pure Nash equilibrium exists in our game. This shows how current impossibility results may provide questionable findings, but this is not sufficient to rehabilitate deterministic hypotheses. Furthermore, we show in Section 3.3 that, in this setting, any deterministic hypothesis can be outperformed by a randomized one. This gives arguments for using randomization, and leads us to a simple method for building randomized classifiers that are robust to state-or-the-art adversarial attacks. In Section 3.4, we validate our

theoretical analysis with empirical results. Finally, we present some additional results, and provide concluding remarks respectively in Sections 3.5, and 3.6.

3.1 Casting the problem as a zero sum game

Notations. *For any set $\mathcal{Z}$ with σ-algebra $\mathcal{A}(\mathcal{Z})$, if there is no ambiguity on the considered σ-algebra, we denote $\mathcal{P}(\mathcal{Z})$ the set of all probability measures over $(\mathcal{Z}, \mathcal{A}(\mathcal{Z}))$. We also denote $\mathcal{F}_{\mathcal{Z}}$ the set of all measurable functions from $(\mathcal{Z}, \mathcal{A}(\mathcal{Z}))$ to $(\mathcal{Z}, \mathcal{A}(\mathcal{Z}))$. For $\mu \in \mathcal{P}(\mathcal{Z})$ and $\boldsymbol{\psi} \in \mathcal{F}_{\mathcal{Z}}$, the* push-forward measure *of μ by $\boldsymbol{\psi}$ is the measure $\boldsymbol{\psi}\#\mu$ such that $\boldsymbol{\psi}\#\mu(B) = \mu(\boldsymbol{\psi}^{-1}(B))$ for any $B \in \mathcal{A}(\mathcal{Z})$. Moreover, for any $B \subset \mathcal{X}$ we denote B^{c} the complement of B in $\mathcal{X}$. Finally, when the probability measure of reference is clear we denote* essup *the essential supremum,* i.e. *the supremum over the non-null sets for this measure.*

3.1.1 Initial problem statement

As this chapter aims at building new intuitions on adversarial classification, we restrict our analysis to the binary classification setting with $0/1$ loss. In the next chapters, we will come back to the more general K-class classification. Let us set $\mathcal{X} \subset [0,1]^d$, $\mathcal{Y} = \{-1, 1\}$ and $\mathcal{H} := \mathcal{C}(\mathcal{X}, \mathbb{R})$ – where $\mathcal{C}(\mathcal{X}, \mathbb{R})$ is the set of functions that are almost everywhere[1] continuous from $\mathcal{X}$ to $\mathbb{R}$. Then, given a distribution $\mathcal{D}$ with full support on $\mathcal{X} \times \mathcal{Y}$, the model provider is looking for a hypothesis $h \in \mathcal{H}$ minimizing the risk of h with respect to $\mathcal{D}$,

$$\begin{aligned} \mathcal{R}(h) &:= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathcal{L}_{0/1}(h(\boldsymbol{x}), y)\right] \\ &= \mathbb{E}_{y\sim\nu}\left[\mathbb{E}_{\boldsymbol{x}\sim\mu_y}\left[\mathcal{L}_{0/1}(h(\boldsymbol{x}), y)\right]\right], \end{aligned} \tag{3.1}$$

where $\nu \in \mathcal{P}(\mathcal{Y})$ is the probability distribution of y, and for any $y \in \mathcal{Y}$, $\mu_y \in \mathcal{P}(\mathcal{X})$ is the conditional law of $\boldsymbol{x}|y$. Given a hypothesis $h \in \mathcal{H}$ and a data sample $(\boldsymbol{x}, y) \sim \mathcal{D}$, the adversary seeks a perturbation $\boldsymbol{\tau} \in \mathcal{X}$ such that $\|\boldsymbol{\tau}\|_p \leq \alpha_p$ and $\mathcal{L}_{0/1}(h(\boldsymbol{x} + \boldsymbol{\tau}), y) = 1$.

From a distributional point of view, this amounts to constructing – for each label $y \in \mathcal{Y}$ – a measurable function $\boldsymbol{\psi}_y$ such that $\boldsymbol{\psi}_y(\boldsymbol{x})$ is the perturbation associated with the labeled example $(\boldsymbol{x}, y)$. This function naturally induces a probability distribution over adversarial examples, which is simply the push-forward measure $\boldsymbol{\psi}_y\#\mu_y$. The goal of the adversary is thus to find $\boldsymbol{\psi} = (\boldsymbol{\psi}_{-1}, \boldsymbol{\psi}_1) \in (\mathcal{F}_{\mathcal{X}|\alpha_p})^2$ that maximizes the adversarial score

$$\text{Score}^{\text{adv}}(h, \boldsymbol{\psi}) := \mathbb{E}_{y\sim\nu}\left[\mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\psi}_y\#\mu_y}\left[\mathcal{L}_{0/1}(h(\boldsymbol{x}), y)\right]\right]. \tag{3.2}$$

Finally, for the attack to remain undetected, we define $\mathcal{F}_{\mathcal{X}|\alpha_p}$ as the set of measurable functions that imperceptibly modifies a distribution on $\mathcal{X}$,

$$\mathcal{F}_{\mathcal{X}|\alpha_p} := \left\{ \boldsymbol{f} \in \mathcal{F}_{\mathcal{X}} \text{ s.t. } \operatorname*{essup}_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{x}\|_p \leq \alpha_p \right\}. \tag{3.3}$$

[1] Here we use almost everywhere with respect to the conditional distributions μ_1 and μ_{-1} defined below.

Within this distributional setting, the adversarial example problem is a two-player zero-sum game, where the defender – model provider – tries to find the best possible hypothesis h, while the adversary is manipulating the dataset distribution. The defender problem then writes as follows.

$$\inf_{h\in\mathcal{H}} \sup_{\boldsymbol{\psi}\in\left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \mathrm{Score}^{\mathrm{adv}}(h,\boldsymbol{\psi}). \tag{3.4}$$

This means that the defender tries to design the hypothesis with the best performance under attack, whereas the adversary will each time design the optimal attack on this hypothesis.

3.1.2 Adversarial attack and defense, a two-player zero-sum game

In game theory, the choice of a hypothesis h – resp. an attack $\boldsymbol{\psi}$ – for the defender – resp. the adversary – is called a *strategy*. Note that the sup-inf and inf-sup problems do not necessarily coincide. In this work, we mainly focus on the defender's point of view which corresponds to the inf-sup problem. We will be interested in understanding how the players behave in this game – *i.e.* the best responses they give to a strategy and whether some equilibria may arise. This motivates the following definitions.

Definition 2 (Best Response)**.** *Let $h \in \mathcal{H}$, and $\boldsymbol{\psi} \in \left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2$.*

- *A* best response *from the defender to $\boldsymbol{\psi}$ is a hypothesis $h^* \in \mathcal{H}$ such that*

$$\mathrm{Score}^{\mathrm{adv}}(h^*,\boldsymbol{\psi}) = \min_{h\in\mathcal{H}} \mathrm{Score}^{\mathrm{adv}}(h,\boldsymbol{\psi}).$$

- *Similarly, a* best response *from the adversary to h is an attack $\boldsymbol{\psi}^* \in \left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2$ such that*

$$\mathrm{Score}^{\mathrm{adv}}(h,\boldsymbol{\psi}^*) = \max_{\boldsymbol{\psi}\in\left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \mathrm{Score}^{\mathrm{adv}}(h,\boldsymbol{\psi}).$$

Remark 10. *Note that the score achieved by a best response from the adversary to h is the adversarial risk of h $\mathrm{Score}^{\mathrm{adv}}(h,\boldsymbol{\psi}^*) = \mathcal{R}^{\mathrm{adv}}(h;\alpha_p)$.*

In the remaining, we denote $\mathrm{BR}(h)$ the set of all best responses of the adversary to a hypothesis h. Similarly $\mathrm{BR}(\boldsymbol{\psi})$ denotes the set of best responses of the defender to an attack $\boldsymbol{\psi}$.

Definition 3 (Pure Nash Equilibrium)**.** *In the zero-sum game from Equation* (3.4)*, a* Pure Nash Equilibrium *is a couple $(h,\boldsymbol{\psi}) \in \mathcal{H} \times \left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2$ such that $h \in \mathrm{BR}(\boldsymbol{\psi})$* and *$\boldsymbol{\psi} \in \mathrm{BR}(h)$.*

When it exists, a Pure Nash Equilibrium is a state of the game in which no player has any incentive to modify its strategy. In our setting, this simultaneously means that no attack could better fool the current hypothesis, and that the hypothesis is optimal for the current attack.

Remark 11. *All the definitions in this section assume a deterministic regime* – i.e. *that neither the defender nor the adversary use randomization – hence the notion of* "Pure" Nash Equilibrium *in the game theory terminology. We discuss extensions to the randomized regime in Section 3.3.*

3.1.3 Trivial solution and regularized adversary

Our current definition of the problem implies that the adversary has perfect information on the dataset distribution and the hypothesis. It also has unlimited computational power and no constraint on the attack except on the size of the perturbations. Thus, it is similar to the adversaries currently studied in the literature – see Section 2.5. However the community starts wondering if this adversary is not too strong to be realistic [46, 62]. Going back to the example of the autonomous car – Chapter 1 – this would mean that the adversary can modify every traffic sign that the camera *may* receive during *any* trip, which is highly unrealistic. The adversary has no downside to attacking, even when the attack is unnecessary – *e.g.* if the attack cannot work or if the point is already misclassified.

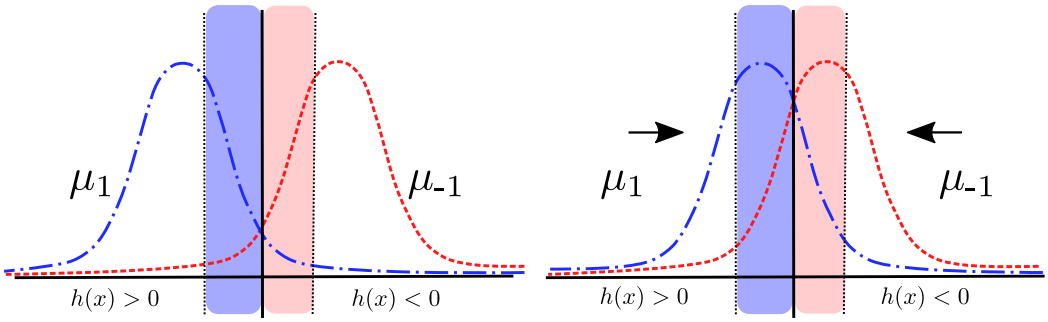


Figure 3.1: Illustration of the conditional distributions μ_{-1} and μ_1. On the left: without attack. On the right: under trivial attack. Blue and red zones are the points that are at distance less than α_p of the boundary.

Figure 3.1 illustrates this phenomenon for the uni-dimensional setting with Gaussian distributions. The adversary moves every point toward the decision boundary[2] – each time saturating the norm constraint – and the defender cannot do much to mitigate the damages. In this case the best classifier remains unchanged, although both curves moved; hence a trivial equilibrium. Furthermore, thanks to Theorem 6, we can evaluate the value of this equilibrium, which can be high – depending on the conditional distributions.

In the remainder of this work, we show that this equilibrium does not hold when we add a small constraint on the adversary's strength – *i.e.* when it is not perfectly indifferent to producing unnecessary perturbations. To formalize the constraint on the adversary, we introduce a penalty term in the initial formulation of the game,

$$\inf_{h\in\mathcal{H}} \sup_{\psi\in\left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \text{Score}^{\text{adv}}_{\Omega}(h,\psi) := \text{Score}^{\text{adv}}(h,\psi) - \lambda\,\Omega(\psi). \tag{3.5}$$

The penalty function Ω represents the limitations on the adversary's budget – be it because of computational resources or to avoid being detected – and $\lambda \in (0,1)$ is some regularization weight.

[2] The decision boundary is the set $\{\boldsymbol{x}\ s.t.\ h(\boldsymbol{x}) = 0\}$

From a computer-security point of view, the first limitation that comes to mind is to limit the number of queries the adversary can send. In our distributional setting, this boils down to penalizing the mass of points that the function $\boldsymbol{\psi}$ moves. Hence we define the penalty as follows[3]

$$\Omega(\boldsymbol{\psi}) := \mathbb{E}_{y\sim\nu}\left[\mathbb{E}_{\boldsymbol{x}\sim\mu_y}[\mathbb{1}\{\boldsymbol{x} \neq \boldsymbol{\psi}_y(\boldsymbol{x})\}]\right]. \tag{3.6}$$

Note that this limitation is also very relevant for the example of the self-driving car example. It forces the adversary to select a few signs that it will attack. In the remaining, we study this regularized game and denote $\mathrm{BR}_\Omega(h)$ the set of all best responses of the adversary to a hypothesis h, under penalty Ω. Since the penalty does not impact the defender's optimization problem, the notation remains unchanged. All above definitions adapt accordingly.

3.2 Instability of the game

Notations. *Let $h \in \mathcal{H}$, we denote $P_h := \{\boldsymbol{x} \in \mathcal{X} \text{ s.t. } h(\boldsymbol{x}) > 0\}$ the set of positive outputs of h and $N_h := \{\boldsymbol{x} \in \mathcal{X} \text{ s.t. } h(\boldsymbol{x}) < 0\}$ the set of negative outputs of h. We also denote $P_h(\alpha_p)$ and $N_h(\alpha_p)$ the set of points on which h can change sign under an α_p-bounded perturbation. $P_h(\alpha_p) := \left\{\boldsymbol{x} \in P_h \text{ s.t. } \exists \boldsymbol{z} \in (P_h)^{\mathsf{c}} \text{ where } \|\boldsymbol{z} - \boldsymbol{x}\|_p \leq \alpha_p\right\}$, and $N_h(\alpha_p)$ likewise.*

3.2.1 Characterizing the best responses

Let us now study how the game behaves when the adversary has been penalized. We show that in this context, no Pure Nash Equilibrium exists. To do so, we characterize the best responses for each player, and show that they can never satisfy Definition 3. We first present the best responses of the penalized adversary.

Lemma 1. *Let $h \in \mathcal{H}$ and $\boldsymbol{\psi} \in \mathrm{BR}_\Omega(h)$. Then the following assertion holds:*

$$\begin{cases} \boldsymbol{\psi}_1(\boldsymbol{x}) \in (P_h)^{\mathsf{c}} & \text{if } \boldsymbol{x} \in P_h(\alpha_p) \\ \boldsymbol{\psi}_1(\boldsymbol{x}) = \boldsymbol{x} & \text{otherwise.} \end{cases}$$

$\boldsymbol{\psi}_{-1}$ is characterized symmetrically.

Proof. Let us first simplify the worst-case adversarial score for h. From the definition of adversarial score we have:

$$\begin{aligned} &\sup_{\boldsymbol{\psi}\in\left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \mathrm{Score}^{\mathrm{adv}}_\Omega(h, \boldsymbol{\psi}) \\ =\ &\sup_{\boldsymbol{\psi}\in\left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \mathbb{E}_{y\sim\nu}\left[\mathbb{E}_{\boldsymbol{x}\sim\mu_y}[\mathbb{1}\{\mathrm{sign}(h(\boldsymbol{\psi}_y(\boldsymbol{x}))) \neq y\} - \lambda\, \mathbb{1}\{\boldsymbol{x} \neq \boldsymbol{\psi}_y(\boldsymbol{x})\}]\right] \end{aligned}$$

[3] We could build a lot of other different penalties. The results would still hold. See *e.g.* Section 3.5 for a penalty on the norm of the perturbation.

$$= \mathbb{E}_{y\sim\nu}\left[\sup_{\boldsymbol{\psi}_y\in\mathcal{F}_{\mathcal{X}|\alpha_p}} \mathbb{E}_{\boldsymbol{x}\sim\mu_y}[\mathbb{1}\{h(\boldsymbol{\psi}_y(\boldsymbol{x}))y\leq 0\} - \lambda\,\mathbb{1}\{\boldsymbol{x}\neq\boldsymbol{\psi}_y(\boldsymbol{x})\}]\right].$$

Finding $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_{-1}$ are two independent optimization problems, hence we focus on characterizing $\boldsymbol{\psi}_1$ – *i.e.* we set $y=1$.

$$\begin{aligned}
&\sup_{\boldsymbol{\psi}_1\in\mathcal{F}_{\mathcal{X}|\alpha_p}} \mathbb{E}_{\boldsymbol{x}\sim\mu_1}[\mathbb{1}\{h(\boldsymbol{\psi}_1(\boldsymbol{x}))\leq 0\} - \lambda\,\mathbb{1}\{\boldsymbol{x}\neq\boldsymbol{\psi}_y(\boldsymbol{x})\}]\\
&= \mathbb{E}_{\boldsymbol{x}\sim\mu_1}\left[\operatorname*{essup}_{\boldsymbol{z}\in B_p(\boldsymbol{x},\alpha_p)} \mathbb{1}\{h(\boldsymbol{z})\leq 0\} - \lambda\,\mathbb{1}\{\boldsymbol{x}\neq\boldsymbol{z}\}\right]\\
&= \int_{\mathcal{X}} \operatorname*{essup}_{\boldsymbol{z}\in B_p(\boldsymbol{x},\alpha_p)} \mathbb{1}\{h(\boldsymbol{z})\leq 0\} - \lambda\,\mathbb{1}\{\boldsymbol{x}\neq\boldsymbol{z}\}\; d\mu_1(\boldsymbol{x}).
\end{aligned}$$

Let us now consider $(H_j)_{j\in J}$ a partition of $\mathcal{X}$, we can write.

$$\begin{aligned}
&\sup_{\boldsymbol{\psi}_1\in\mathcal{F}_{\mathcal{X}|\alpha_p}} \mathbb{E}_{\boldsymbol{x}\sim\mu_1}[\mathbb{1}\{h(\boldsymbol{\psi}_1(\boldsymbol{x}))\leq 0\} - \lambda\,\mathbb{1}\{\boldsymbol{x}\neq\boldsymbol{\psi}_y(\boldsymbol{x})\}]\\
&= \sum_{j\in J}\int_{H_j} \operatorname*{essup}_{\boldsymbol{z}\in B_p(\boldsymbol{x},\alpha_p)} \mathbb{1}\{h(\boldsymbol{z})\leq 0\} - \lambda\,\mathbb{1}\{\boldsymbol{x}\neq\boldsymbol{z}\}\; d\mu_1(\boldsymbol{x}).
\end{aligned}$$

In particular, we can take $H_0 = P_h^{\mathsf{c}}$, $H_1 = P_h \setminus P_h(\alpha_p)$, and $H_2 = P_h(\alpha_p)$. Then we can study the three sets independently.

1. For any $\boldsymbol{x}\in H_0 = P_h^{\mathsf{c}}$, taking $\boldsymbol{z}=\boldsymbol{x}$ we get $\mathbb{1}\{h(\boldsymbol{z})\leq 0\} - \lambda\,\mathbb{1}\{\boldsymbol{x}\neq\boldsymbol{z}\} = 1$. Since for any $\boldsymbol{z}\in\mathcal{X}$ we have $\mathbb{1}\{h(\boldsymbol{z})\leq 0\} - \lambda\,\mathbb{1}\{\boldsymbol{x}\neq\boldsymbol{z}\}\leq 1$, this strategy is optimal. Furthermore, for any other optimal strategy $\boldsymbol{z}'$, we would have $\mathbb{1}\{\boldsymbol{x}\neq\boldsymbol{z}\} = 0$, hence $\boldsymbol{z}'=\boldsymbol{x}$.

2. For any $\boldsymbol{x}\in H_1 = P_h\setminus P_h(\alpha_p)$, we have that $B_p(\boldsymbol{x},\alpha_p)\subset P_h$ by definition of $P_h(\alpha_p)$. Hence, for any $\boldsymbol{z}\in B_p(\boldsymbol{x},\alpha_p)$, one gets $h(\boldsymbol{z})>0$. Then $\mathbb{1}\{h(\boldsymbol{z})\leq 0\} - \lambda\,\mathbb{1}\{\boldsymbol{x}\neq\boldsymbol{z}\}\leq 0$. The only optimal $\boldsymbol{z}$ will thus be $\boldsymbol{z}=\boldsymbol{x}$, giving value 0.

3. Finally, for any $\boldsymbol{x}\in H_2 = P_h(\alpha_p)$, we have that $B_p(\boldsymbol{x},\alpha_p)\cap P_h^{\mathsf{c}}\neq\emptyset$, and for any $\boldsymbol{z}$ in this intersection, one has $h(\boldsymbol{z})\leq 0$ and $\boldsymbol{z}\neq\boldsymbol{x}$. Hence

$$\operatorname*{essup}_{\boldsymbol{z}\in B_p(\boldsymbol{x},\alpha_p)} \mathbb{1}\{h(\boldsymbol{z})\leq 0\} - \lambda\,\mathbb{1}\{\boldsymbol{z}\neq\boldsymbol{x}\} = \max(1-\lambda, 0).$$

   Since $\lambda\in(0,1)$ one has $\mathbb{1}\{h(\boldsymbol{z})\leq 0\} - \lambda\,\mathbb{1}\{\boldsymbol{z}\neq\boldsymbol{x}\} = 1-\lambda$ for any $\boldsymbol{z}\in B_p(\boldsymbol{x},\alpha_p)\cap P_h^{\mathsf{c}}$. Then any function that outputs $\boldsymbol{\psi}_1(\boldsymbol{x})\in B_p(\boldsymbol{x},\alpha_p)\cap P_h^{\mathsf{c}}$ is optimal on H_2.

Since $H_0 \cup H_1 \cup H_2 = \mathcal{X}$, Lemma 1 holds. The proof for $y = -1$ is symmetrical. Furthermore, the value for the optimal score writes

$$\begin{aligned}
&\sup_{\psi \in \left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \mathrm{Score}_\Omega^{\mathrm{adv}}(h, \boldsymbol{\psi}) \\
&= \mathbb{E}_{y\sim\nu}\left[\sum_{j\in J} \int_{H_j} \operatorname*{essup}_{\boldsymbol{z}\in B_p(\boldsymbol{x},\alpha_p)} \mathbb{1}\{h(\boldsymbol{z})y \leq 0\} - \lambda\, \mathbb{1}\{\boldsymbol{x} = \boldsymbol{z}\}\; d\mu_y(\boldsymbol{x})\right] \\
&= \sum_{y=\pm 1} \nu(y) \sum_{j\in J} \int_{H_j} \operatorname*{essup}_{\boldsymbol{z}\in B_p(\boldsymbol{x},\alpha_p)} \mathbb{1}\{h(\boldsymbol{z})y \leq 0\} - \lambda\, \mathbb{1}\{\boldsymbol{x} = \boldsymbol{z}\}\; d\mu_y(\boldsymbol{x})\,.
\end{aligned}$$

Since the value is 0 on $P_h \setminus P_h(\alpha_p)$ for $\boldsymbol{\psi}_1$ – resp. on $N_h \setminus N_h(\alpha_p)$ for $\boldsymbol{\psi}_{-1}$ – one gets

$$= \mathcal{R}(h) + (1-\lambda)\big(\nu(1)\mu_1(P_h(\alpha_p)) + \nu(-1)\mu_{-1}(N_h(\alpha_p))\big), \tag{3.7}$$

where Equation (3.7) holds since $\mathcal{R}(h) = \nu(1)\mu_1(P_h^{\mathsf{c}}) + \nu(-1)\mu_{-1}(N_h^{\mathsf{c}})$. This provides an interesting decomposition of the adversarial risk into the risk without attack and the loss the adversary produces by attacking that recall the decomposition in Chapter 2. □

Note that an optimal attack will only change points that are close enough to the decision boundary. This means that, when the adversary cannot change the hypothesis' decision on a point, it will not attack it. Let us now study what happens for the defender. At a first glance, one would suspect that the best response for the defender ought to be the Bayes optimal classifier for the transported distributions. However, it is only well defined if the conditional distributions admit a probability density function. This might not always hold here for the transported distribution. Nevertheless, we present a property, shared by the Bayes optimal classifier when defined, that always holds for the defender's best response.

Lemma 2. *Let us consider $\boldsymbol{\psi} \in \left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2$. If we take $h \in \mathrm{BR}(\boldsymbol{\psi})$, then for any measurable $B \subset P_h$ one has $\boldsymbol{\psi}_1 \# \mu_1(B) \times \nu(1) \geq \boldsymbol{\psi}_{-1} \# \mu_{-1}(B) \times \nu(1)$. A similar result holds for N_h.*

Proof. We reason ad absurdum with the following assumption:

There exists a measurable set $C \subset P_h$ such that $\nu(-1)\boldsymbol{\psi}_{-1}\#\mu_{-1}(C) > \nu(1)\boldsymbol{\psi}_1\#\mu_1(C)$.

Let us construct $\bar{h}$ as follows:

$$\bar{h}(\boldsymbol{x}) = \begin{cases} h(\boldsymbol{x}) & \text{if } \boldsymbol{x} \notin C \\ -1 & \text{otherwise.} \end{cases}$$

Since h and $\bar{h}$ are identical outside C, the difference between the adversarial scores of h and $\bar{h}$ writes as follows:

$$\mathrm{Score}_\Omega^{\mathrm{adv}}(h, \boldsymbol{\psi}) - \mathrm{Score}_\Omega^{\mathrm{adv}}(\bar{h}, \boldsymbol{\psi})$$

$$= \mathbb{E}_{y\sim\nu}\left[\int_C \mathbb{1}\{h(\boldsymbol{x})y \leq 0\} - \mathbb{1}\{\bar{h}(\boldsymbol{x})y \leq 0\}\ d\boldsymbol{\psi}_y\#\mu_y(\boldsymbol{x})\right]$$
$$= \sum_{y\pm 1}\nu(y)\left[\int_C \mathbb{1}\{h(\boldsymbol{x})y \leq 0\} - \mathbb{1}\{\bar{h}(\boldsymbol{x})y \leq 0\}\ d\boldsymbol{\psi}_y\#\mu_y(\boldsymbol{x})\right].$$

Since – by construction – for any $\boldsymbol{x} \in C$ we have $\bar{h}(\boldsymbol{x}) < 0$ and $h(\boldsymbol{x}) > 0$, we can write

$$\begin{aligned}&\text{Score}^{\text{adv}}_{\Omega}(h, \boldsymbol{\psi}) - \text{Score}^{\text{adv}}_{\Omega}(\bar{h}, \boldsymbol{\psi})\\ &= \nu(-1)\boldsymbol{\psi}_{\text{-}1}\#\mu_{\text{-}1}(C) - \nu(1)\boldsymbol{\psi}_1\#\mu_1(C)\end{aligned}$$

Since we assumed $\nu(-1)\boldsymbol{\psi}_{\text{-}1}\#\mu_{\text{-}1}(C) > \nu(1)\boldsymbol{\psi}_1\#\mu_1(C)$ the difference between the adversarial scores of h and $\bar{h}$ is strictly positive. This means that $\bar{h}$ gives strictly better adversarial score than the best response h, leading to a contradiction. Hence Lemma 2 holds. The proof for N_h is symmetrical. □

In particular, when $\boldsymbol{\psi}_1\#\mu_1$ and $\boldsymbol{\psi}_{\text{-}1}\#\mu_{\text{-}1}$ admit probability density functions, Lemma 2 means that h is the Bayes optimal classifier for the distribution characterized by ν, $\boldsymbol{\psi}_{\text{-}1}\#\mu_{\text{-}1}$ and $\boldsymbol{\psi}_1\#\mu_1$.

3.2.2 No Pure Nash Equilibrium in the game

We can now state our first main result relating the absence of equilibrium in the regularized game.

Theorem 7 (Non-existence of a pure Nash equilibrium)**.** *In the zero-sum game from Equation* (3.5)*, there is no Pure Nash Equilibrium.*

Proof. Let h be a classifier and $\boldsymbol{\psi} \in \text{BR}_{\Omega}(h)$ an optimal attack against h. We will show that $h \notin \text{BR}(\boldsymbol{\psi})$ – *i.e.* that h does not satisfy the condition from Lemma 2. It suffices for Theorem 7 to hold since it implies that there is no $(h, \boldsymbol{\psi}) \in \mathcal{H} \times \left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2$ such that $h \in \text{BR}(\boldsymbol{\psi})$ *and* $\boldsymbol{\psi} \in \text{BR}_{\Omega}(h)$.

According to Lemma 1, we have $\boldsymbol{\psi}_1\#\mu_1(P_h(\alpha_p)) = 0$ – *i.e.* $P_h(\alpha_p)$ is of null measure for the transported distribution conditioned by $y = 1$. Since $\boldsymbol{\psi}_{\text{-}1}$ is the identity on $P_h(\alpha_p)$, and since $\mu_{\text{-}1}$ is of full support on $\mathcal{X}$[a] we have

$$\boldsymbol{\psi}_{\text{-}1}\#\mu_{\text{-}1}(P_h(\alpha_p)) = \mu_{\text{-}1}(P_h(\alpha_p)) > 0. \tag{3.8}$$

Hence we get the following

$$\boldsymbol{\psi}_{\text{-}1}\#\mu_{\text{-}1}(P_h(\alpha_p)) > \boldsymbol{\psi}_1\#\mu_1(P_h(\alpha_p)). \tag{3.9}$$

Since the right side of the inequality is null, for any $\nu(1)$ and $\nu(\text{-}1)$ we get

$$\boldsymbol{\psi}_{\text{-}1}\#\mu_{\text{-}1}(P_h(\alpha_p))\nu(\text{-}1) > \boldsymbol{\psi}_1\#\mu_1(P_h(\alpha_p))\nu(1). \tag{3.10}$$

This inequality is incompatible with the characterization of the best response for the defender of Lemma 2. Hence $h \notin \mathrm{BR}(\psi)$.

□

[a] Note that the full support hypothesis is much stronger than what we actually need. Fundamentally, we only need the null sets for measures μ_1 and μ_{-1} to be sufficiently far one from the other.

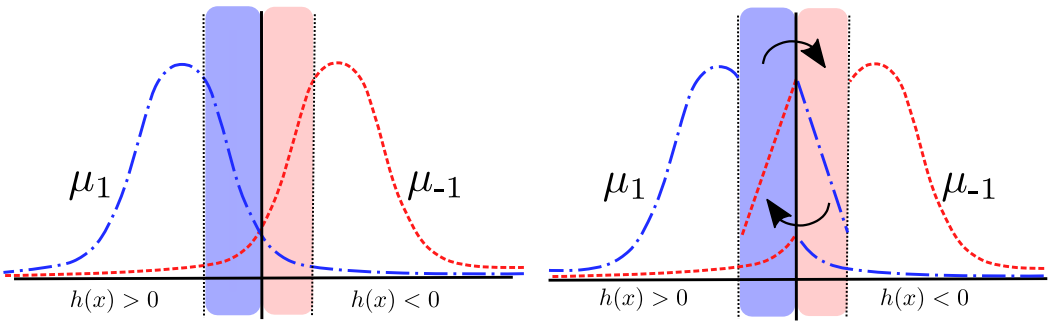


Figure 3.2: Illustration of the conditional distributions μ_{-1} and μ_1. On the left: without attack. On the right: under penalized attack. Blue and red zones are respectively the sets $P_h(\alpha_p)$ and $N_h(\alpha_p)$.

Figure 3.2 illustrates Theorem 7 with two uni-dimensional Gaussian distributions. We see that – one the right – μ_1 is set to 0 in $P_h(\alpha_p)$, and this mass is transferred into $N_h(\alpha_p)$. The symmetric holds for μ_{-1}. After attack, we have $\mu_1(P_h(\alpha_p)) = 0$. Hence, any small amount of mass for μ_{-1} in $P_h(\alpha_p)$ is now sufficient to make it dominant; hence the zone will now be classified -1 by the Bayes optimal classifier. This result has several deep consequences. Among them, we focus on the following two.

Consequence 1: There might be room for robustness after all

The above result shows the fundamental difference between regularized and unregularized games. While in the unregularized setting there may exist a pure – trivial – Nash Equilibrium, our analysis shows that such an equilibrium cannot exist as soon as we add an infinitesimally small regularization. Hence, our result highlights a very interesting property of the unregularized problem, which is its instability. This leads us to the following conclusions.

- We should reconsider the works on the limits of classification under perturbation and verify whether these results still hold – or are diminished – when we add a set of realistic constraints to the adversary, be it the one we just described or more sophisticated ones.

- There might be room for robustness after all. Even if for now, the defense community seems to be losing the race, the game is not over yet. If we design more realistic adversaries, we may be able to understand better the threat and design more robust models.

Consequence 2: No free lunch for transferable examples

To understand this statement, first note that thanks to the weak duality, the following inequality always holds

$$\sup_{\boldsymbol{\psi}\in\left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \inf_{h\in\mathcal{H}} \mathrm{Score}_{\Omega}^{\mathrm{adv}}(h,\boldsymbol{\psi}) \leq \inf_{h\in\mathcal{H}} \sup_{\boldsymbol{\psi}\in\left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \mathrm{Score}_{\Omega}^{\mathrm{adv}}(h,\boldsymbol{\psi}).$$

On the left side problem – sup-inf – the adversary looks for the best strategy $\boldsymbol{\psi}$ against any *unknown* hypothesis. This is closely related to the notion of *transferablility* of the attacks – investigated *e.g.* in [67, 155] – which refers to attacks successful against a wide range of hypotheses. On the right side problem – inf-sup – the defender tries to find the best hypothesis under any possible attack, whereas the adversary plays in second and specifically attacks this hypothesis. As a consequence of Theorem 7, the inequality is always strict:

$$\sup_{\boldsymbol{\psi}\in\left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \inf_{h\in\mathcal{H}} \mathrm{Score}_{\Omega}^{\mathrm{adv}}(h,\boldsymbol{\psi}) < \inf_{h\in\mathcal{H}} \sup_{\boldsymbol{\psi}\in\left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \mathrm{Score}_{\Omega}^{\mathrm{adv}}(h,\boldsymbol{\psi}).$$

This means that the problems are not equivalent. In particular, an attack designed to succeed against *any* hypothesis – *i.e.* a transferable attack – will not be as good as an attack tailored for a given hypothesis. The adversary must therefore make a trade-off between effectiveness and transferability of the attack. This sends a second encouraging message to the defense community.

3.3 Randomization might be the clue

3.3.1 Adaptation of the problem statement

We just found that adversarial defense might be possible. However, both the current literature on adversarial attacks and the instability of the game in the deterministic setting pushes us to widen the class of strategies we consider. A natural extension of the game would be to allow randomization for both players. Now they choose a distribution over pure strategies, leading to the following game

$$\inf_{\eta\in\mathcal{P}(\mathcal{H})} \sup_{\Psi\in\mathcal{P}\left(\left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2\right)} \mathbb{E}_{h\sim\eta,\boldsymbol{\psi}\sim\Psi}\left[\mathrm{Score}_{\Omega}^{\mathrm{adv}}(h,\boldsymbol{\psi})\right]. \tag{3.11}$$

Without making further assumptions – *e.g.* compactness – we cannot apply known results from game theory to prove the existence of an equilibrium. Studying the equilibrium is appealing from a theoretical point of view but would require strong results in the theory of optimal transport; hence we leave it to further investigations. But even without knowing if an equilibrium exists in the randomized setting, we can prove that *randomization matters*. More precisely we show that any deterministic hypothesis can be outperformed by a randomized one in terms of the worst-case adversarial score. To do so we simplify Equation (3.11) in two ways:

- We keep considering deterministic adversaries – *i.e.* we restrict the search space of the adversary to $(\mathcal{F}_{\mathcal{X}|\alpha_p})^2$ instead of $\mathcal{P}\left((\mathcal{F}_{\mathcal{X}|\alpha_p})^2\right)$. This condition corresponds to the current

state-of-the-art in the domain: to the best of our knowledge, no efficient randomized adversarial attack has been designed – and so is used – yet.

- We only consider a subclass of randomized hypotheses, called mixtures, which are discrete probability measures on a finite set of hypotheses. We show that this randomization is enough to outperform any deterministic hypothesis. We will discuss in Chapters 4 and 5 the use of more general randomized hypothesis spaces. Let us now define a mixture.

Definition 4 (Mixture of hypothesis)**.** *Let $m \in \mathbb{N}$, $\boldsymbol{h} = (h_1, ..., h_m) \in \mathcal{H}^m$ a vector of m hypothesis functions and $\boldsymbol{q} = (q_1, \dots, q_m) \in \mathcal{P}(\{1, \dots, m\})$ a probability vector*[4]*. A* mixed hypothesis of $\boldsymbol{h}$ by $\boldsymbol{q}$ *is a mapping $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$ from $\mathcal{X}$ to $\mathcal{P}(\mathbb{R})$ such that for all $\boldsymbol{x} \in \mathcal{X}$, $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}(\boldsymbol{x})$ outputs $h_i(\boldsymbol{x})$ with probability q_i.*

We call such a mixture a *mixed strategy* of the defender. Given some $\boldsymbol{x} \in \mathcal{X}$, this amounts to picking a hypothesis h_i from $\boldsymbol{h}$ at random following the distribution $\boldsymbol{q}$, and use it to output the predicted class for $\boldsymbol{x}$ – *i.e.* $\operatorname{sign}(h_i(\boldsymbol{x}))$. Note that a mixed strategy for the defender is a non-deterministic algorithm, since it depends on the sampling one makes on $\boldsymbol{q}$. Hence, even if the attack space remains unchanged, the adversary now needs to maximize a new objective function which is the expectation of the adversarial score under the distribution $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}(\boldsymbol{x})$. It writes as follows

$$\operatorname{Score}_{\Omega}^{\mathrm{adv}}(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}, \boldsymbol{\psi}) := \mathbb{E}_{y\sim\nu}\left[\mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{\psi}_y\#\mu_y}\left[\mathbb{E}_{i\sim\boldsymbol{q}}\left[\mathcal{L}_{0/1}(h_i(\boldsymbol{x}), y)\right]\right]\right] - \lambda\,\Omega(\boldsymbol{\psi}). \tag{3.12}$$

This notion of score is the natural extension of the deterministic case; hence we keep the notation $\operatorname{Score}_{\Omega}^{\mathrm{adv}}$. In the following, it will be clear from context that the defender uses a mixed strategy.

3.3.2 Randomization matters: how to outperform deterministic hypotheses

Using this new set of hypotheses for the defender, we demonstrate that we can improve deterministic defenses using a simple mixed strategy. This method presents similarities with the notions of fictitious play [23] in game theory, and boosting in machine learning [56]. Given a deterministic hypothesis h_1, we combine it – via randomization – with the best response h_2 to its optimal attack. The rationale behind this idea is that – by construction – efficient attacks on one of these two hypotheses will not work on the other. Mixing h_1 with h_2 has two opposite consequences on the adversarial score. On one hand, where we only had to defend against attacks on h_1, we are now also vulnerable to attacks on h_2, so the total set of possible attacks is now bigger. On the other hand, each attack will only work part of the time, depending on the probability distribution $\boldsymbol{q}$. If we can calibrate the weights so that the new attacks have a low probability of succeeding, then the average risk under attack on the mixture will be low.

Theorem 8 (Randomization matters)**.** *Let us consider $h_1 \in \mathcal{H}$, $\lambda \in (0, 1)$, $\boldsymbol{\psi} \in \mathrm{BR}_{\Omega}(h_1)$ and $h_2 \in \mathrm{BR}(\boldsymbol{\psi})$. Then for any $q_1 \in (\max(\lambda, 1-\lambda), 1)$ and for any $\boldsymbol{\psi}' \in \mathrm{BR}_{\Omega}(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}})$ one has*

$$\operatorname{Score}_{\Omega}^{\mathrm{adv}}(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}, \boldsymbol{\psi}') < \operatorname{Score}_{\Omega}^{\mathrm{adv}}(h_1, \boldsymbol{\psi}).$$

Where $\boldsymbol{h} = (h_1, h_2)$, $\boldsymbol{q} = (q_1, 1-q_1)$, and $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$ is the mixture of $\boldsymbol{h}$ by $\boldsymbol{q}$.

[4] Note that we already defined the simplex $\Delta(m) = \mathcal{P}(\{1, \dots, m\})$, but for consistency of the notations in this chapter we use $\mathcal{P}(\{1, \dots, m\})$.

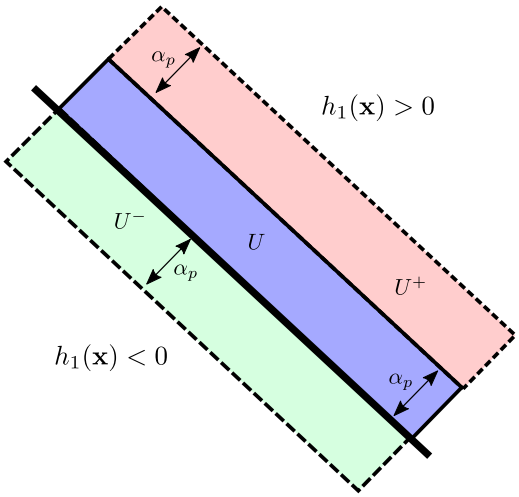


Figure 3.3: Illustration of the notations U, U^+, and U^- for proof of Theorem 8.

Proof. To demonstrate Theorem 8, let us denote $U = P_{h_1}(\alpha_p)$ and define the *α_p-dilation of U* as $U \oplus \alpha_p := \left\{u + v \mid (u, v) \in U \times \mathcal{X} \text{ and } \|v\|_p \leq \alpha_p\right\}$. We can construct h_2 as follows

$$h_2(\boldsymbol{x}) = \begin{cases} -h_1(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in U \\ h_1(\boldsymbol{x}) & \text{otherwise.} \end{cases}$$

This means that h_2 changes the class of all points in U, and do not change the rest, compared to h_1. Then taking $q_1 \in (0, 1)$, we can define $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$, and $\boldsymbol{\psi}' \in \mathrm{BR}_\Omega(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}})$. We aim to find a condition on q_1 so that the score of $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$ is lower than the score of h_1. Finally, let us recall that

$$\begin{aligned}
&\mathrm{Score}_\Omega^{\mathrm{adv}}(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}, \boldsymbol{\psi}') \\
&= \nu(1) \int_{\mathcal{X}} \operatorname*{essup}_{\boldsymbol{z} \in B_p(\boldsymbol{x}, \alpha_p)} q_1 \mathbb{1}\{h_1(\boldsymbol{z}) \leq 0\} + (1 - q_1) \mathbb{1}\{h_2(\boldsymbol{z}) \leq 0\} - \lambda \mathbb{1}\{\boldsymbol{x} \neq \boldsymbol{z}\} \, d\mu_1(\boldsymbol{x}) \\
&+ \nu(-1) \int_{\mathcal{X}} \operatorname*{essup}_{\boldsymbol{z} \in B_p(\boldsymbol{x}, \alpha_p)} q_1 \mathbb{1}\{h_1(\boldsymbol{z}) \geq 0\} + (1 - q_1) \mathbb{1}\{h_2(\boldsymbol{z}) \geq 0\} - \lambda \mathbb{1}\{\boldsymbol{x} \neq \boldsymbol{z}\} \, d\mu_{-1}(\boldsymbol{x}).
\end{aligned}$$

The only terms that may vary between the score of h_1 and the score of $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$ are the integrals on $U, U \oplus \alpha_p \cap P_{h_1}$ and $\boldsymbol{\psi}_{-1}^{-1}(U)$ – inverse image of U by $\boldsymbol{\psi}_{-1}$. These sets represent respectively the points we mix on, the points that may become attacked – when changing from h_1 to $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$ – by moving them on U, and the ones that were – for h_1 – attacked before by moving them on U. Hence, for simplicity, we only write those terms. Furthermore, we denote

$$U^+ := U \oplus \alpha_p \cap P_{h_1} \setminus U, \; U^- := \boldsymbol{\psi}_{-1}^{-1}(U) \text{ and recall } U := P_{h_1}(\alpha_p).$$

One can refer to Figure 3.3 for visual interpretation of this sets. We can now evaluate the worst-case adversarial score for h_1 restricted to the above sets. Thanks to Lemma 1 that characterizes $\boldsymbol{\psi}$, we can write

$$\begin{aligned}
&\mathrm{Score}^{\mathrm{adv}}_{\Omega}(h_1, \boldsymbol{\psi})_{|U,\, U^+,\, U^-} \\
&= (1-\lambda) \times \nu(1)\mu_1(U) + \nu(\text{-}1)\mu_{\text{-}1}(U) \\
&+ 0 \times \nu(1)\mu_1\big(U^+\big) + \nu(\text{-}1)\mu_{\text{-}1}\big(U^+\big) \\
&+ \nu(1)\mu_1\big(U^-\big) + (1-\lambda) \times \nu(\text{-}1)\mu_{\text{-}1}\big(U^-\big).
\end{aligned}$$

Similarly, we can write the worst-case adversarial score of the mixture on the sets we consider. Note that the max operator comes from the fact that the adversary has to make a choice between attacking the zone or just taking advantage of the error due to randomization.

$$\begin{aligned}
&\mathrm{Score}^{\mathrm{adv}}_{\Omega}(\mathfrak{m}^{\boldsymbol{q}}_{\boldsymbol{h}}, \boldsymbol{\psi}')_{|U,\, U^+,\, U^-} \\
&= \max(1-q_1, 1-\lambda) \times \nu(1)\mu_1(U) + \max(q_1, 1-\lambda) \times \nu(\text{-}1)\mu_{\text{-}1}(U) \\
&+ \max(0, 1-q_1-\lambda) \times \nu(1)\mu_1\big(U^+\big) + \nu(\text{-}1)\mu_{\text{-}1}\big(U^+\big) \\
&+ \nu(1)\mu_1\big(U^-\big) + \max(0, q_1-\lambda) \times \nu(\text{-}1)\mu_{\text{-}1}\big(U^-\big).
\end{aligned}$$

Computing the difference between these two terms, we get the following

$$\mathrm{Score}^{\mathrm{adv}}_{\Omega}(h_1, \boldsymbol{\psi}) - \mathrm{Score}^{\mathrm{adv}}_{\Omega}(\mathfrak{m}^{\boldsymbol{q}}_{\boldsymbol{h}}, \boldsymbol{\psi}') \tag{3.13}$$
$$= (1-\lambda-\max(1-q_1, 1-\lambda)) \times \nu(1)\mu_1(U) \tag{3.14}$$
$$+ (1-\max(q_1, 1-\lambda)) \times \nu(\text{-}1)\mu_{\text{-}1}(U) \tag{3.15}$$
$$- \max(0, 1-q_1-\lambda) \times \nu(1)\mu_1\big(U^+\big) \tag{3.16}$$
$$+(1-\lambda-\max(0, q_1-\lambda)) \times \nu(\text{-}1)\mu_{\text{-}1}\big(U^-\big)\,. \tag{3.17}$$

First recall that both μ_1 and $\mu_{\text{-}1}$ have full support. Let us now simplify Equation (3.13) using additional assumptions.

- First, we have that Equation (3.15) is equal to

  $$\min(1-q_1, \lambda)\mu_{\text{-}1}(U)\nu(\text{-}1) > 0.$$

  Thus, a sufficient condition for the difference between the adversarial scores to be positive is to have the other terms greater or equal to 0.

- To have Equation (3.14) ≥ 0 we can always set $\max(1-q_1, 1-\lambda) = 1-\lambda$. This gives us $q_1 \geq \lambda$.

- Also note that to get (3.16) ≥ 0, we can force $\max(1 - q_1 - \lambda, 0) = 0$. This gives us $q_1 \geq 1 - \lambda$.
- Finally, since $q_1 \geq \lambda$, we have that $1 - \lambda - \max(0, q_1 - \lambda) = 1 - q_1$ thus Equations (3.17) > 0.

With the above simplifications, we have (3.13) > 0 for any $q_1 > \max(\lambda, 1 - \lambda)$ which concludes the proof. □

Remark 12. *Note that depending on the initial hypothesis h_1 and the conditional distributions μ_1 and μ_{-1}, the gap between* $\mathrm{Score}_{\Omega}^{\mathrm{adv}}(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}, \boldsymbol{\psi}')$ *and* $\mathrm{Score}_{\Omega}^{\mathrm{adv}}(h_1, \boldsymbol{\psi})$ *could vary. Therefore, with additional conditions on h_1, μ_1 and μ_{-1}, we could make the gap appear more explicitly. We keep the formulation general to emphasize that for* any *h_1, we can build a better $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$.*

Even-though Theorem 8 only applies to mixtures of two classifiers, it directly implies that randomized hypotheses – defined in a broader way – outperform deterministic ones in terms of regularized adversarial score. Based on this finding, we devise a simple procedure called *boosted adversarial training* to construct a robust mixture of two hypotheses. It relies on three core principles: adversarial training, boosting and randomization. The procedure is summarized in Algorithm 1.

Algorithm 1: *boosted adversarial training*

Input : D the training data set and q_1 the probability parameter.

Train h_1 on D with adversarial training
Generate the adversarial data set $\tilde{D}$ against h_1.
Train h_2 on $\tilde{D}$
$\boldsymbol{q} \leftarrow (q_1, 1 - q_1)$
$\boldsymbol{h} \leftarrow (h_1, h_2)$

return $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$

Given a dataset D and a probability parameter $q_1 \in [1/2, 1)$, we construct h_1 the first hypothesis of the mixture by using adversarial training on D. Then, we train the second hypothesis h_2 on a data set $\tilde{D}$ that contains adversarial examples for h_1. At the end of the procedure, we return the mixture constructed with the two hypothesis where the first one has a probability of q_1 and the second $1 - q_1$ accordingly. The parameter q_1 is found by conducting a grid-search.

3.4 Numerical validation: improving adversarial training

To empirically evaluate the above procedure, we run a series of experiments on the CIFAR-10 and CIFAR-100 datasets using deep neural networks. We show that the above simple randomization scheme can improve the robustness of adversarial training. Let us first start by presenting the experimental setup we use. For direct access to the implementation, one can refer to the following Github repository.

https://github.com/MILES-PSL/Randomization-matters-How-to-defend-against-strong-adversarial-attacks

3.4.1 Experimental setup

Architecture and training procedure

All the hypotheses we use in this section are WideResNets – see [177] – with 28 layers, a widen factor of 10, a dropout factor of 0.3 and LeakyRelu activation with a 0.1 slope. To train an undefended classifier we use the following hyper-parameters.

- *Number of Epochs:* 200
- *Batch size:* 128
- *Loss function:* Cross Entropy Loss
- *Optimizer :* Stochastic gradient descent algorithm with momentum 0.9, weight decay of 2×10^{-4} and a learning rate that decreases during the training as follows:

$$lr = \begin{cases} 0.1 & \text{if} \quad 0 \leq \text{epoch} < 60 \\ 0.02 & \text{if} \quad 60 \leq \text{epoch} < 120 \\ 0.004 & \text{if} \quad 120 \leq \text{epoch} < 160 \\ 0.0008 & \text{if} \quad 160 \leq \text{epoch} < 200. \end{cases}$$

Remark 13. *To train a hypothesis with adversarial training we use the same hyper-parameters as above, and generate adversarial examples during training using an ℓ_∞ adversary with 20 iterations. We also use PGD with 20 iterations and $\alpha_\infty = 0.031$ to build $\tilde{D}$.*

Threat models

To compare the empirical performances of our method with adversarial training, we consider two ℓ_p adversaries with thresholds corresponding to CIFAR datasets

- *An ℓ_∞ adversary with perturbation bounded by* 0.031. To model this adversary we use the PGD attack with $t_{max} = 100$ iterations and a step-size $s = 0.008$.
- *An ℓ_2 adversary with perturbation bounded by* 0.8. To model this adversary we use the C&W attack with 100 iterations, a learning rate equal to 0.01, 9 binary search steps, and an initial constant of $\kappa = 0.001$.

Note that, when evaluating a defense against adversarial examples, it is crucial to test the robustness of the method against the best possible attack. Accordingly, the defense method should be evaluated against attacks that were specifically tailored to it – *a.k.a.* adaptive attacks [154]. Specifically, when evaluating randomized algorithms, one should not try to compute the gradient over

the logits/probits directly to avoid gradient masking as pointed out in [6] and [26]. Instead – as we explained in Equation (3.12) – we should provide the expected logits/probits of the mixture to the adversary. Since we assume perfect information for the adversary, it knows the distribution of the mixture; hence it can directly compute the expectation over h_1 and h_2 – without having to go through a Monte Carlo sampling scheme.

3.4.2 Results

In Table 3.1 we present results for $q_1 = 0.8$ and compared with classical adversarial training [103][5]. The accuracy and accuracy under attack presented for the mixture are expectation over h_1 and h_2 with respect to $\boldsymbol{q}$ – as explained above, *i.e.* we give the true expectation.

Table 3.1: Accuracy under attack of a single adversarially trained classifier (AT) and the mixture formed with our method (Ours) on CIFAR-10 and CIFAR-100 datasets.

Dataset	**Method**	**Accuracy without attack**	l_∞**-PGD** $\alpha_\infty = 0.031$	ℓ_2**-C&W** $\alpha_2 = 0.8$
CIFAR-10	Undefended	0.88	0.00	0.00
	AT [103]	0.83	0.42	0.35
	Ours	0.80	**0.55**	**0.53**
CIFAR-100	Undefended	0.62	0.00	0.00
	AT [103]	0.58	0.26	0.22
	Ours	0.56	**0.40**	**0.38**

These results show that for both model threats and on both datasets, the accuracy under attack of our mixture is much higher – 0.10 better against any adversary – than the single classifier with adversarial training. However the standard accuracy of our technique dropped a little bit in the process – minus $0.3/0.2$ compared to adversarial training. A trade-off between robustness and standard accuracy seems to appear. Indeed – be it adversarial training or boosted adversarial training – the better the accuracy under attack, the worse the accuracy without attack. Nevertheless, the trade-off is not linear since boosted adversarial training gains four times more robust accuracy than it loses standard one. This indicates that randomization can improve robustness of deterministic hypotheses. However, one should be careful when analyzing Table 3.1. We should not draw conclusions either on the efficacy of a defense nor on the trade-off between robustness and accuracy only based on empirical evidence – since empirical defenses are often broken sometimes after being designed [6, 38, 154]. Therefore, we need further theoretical and empirical investigations to validate randomization as a proper defense strategy.

[5] Note that we compare here with the vanilla version of adversarial training. Other versions exist with slightly better accuracy under attack. Furthermore, to avoid some computational burden, we did not use data augmentation during the leaning procedure – which explains some differences with the initial paper [103].

3.5 Additional results: another type of penalty

The core arguments we used to demonstrate Lemma 1 and Theorems 8 do not depend on the form of the penalty we consider. Any notion of distance between the perturbation and the initial point would allow to find the same kind of results. To show this, let us consider that $\mathcal{X}$ is a Hilbert space with a dot product $\langle ., . \rangle$ and associated norm $||.|| = \sqrt{\langle ., . \rangle}$. Then we can define the following regularization that penalizes the expected norm under perturbation $\boldsymbol{\psi}$,

$$\Omega(\boldsymbol{\psi}) := \mathbb{E}_{y\sim\nu}\left[\mathbb{E}_{\boldsymbol{x}\sim\mu_y}[\|\boldsymbol{x} - \boldsymbol{\psi}_y(\boldsymbol{x})\|]\right]. \tag{3.18}$$

This regularization could for example materialize an adversary that seeks a solution to the Lagrangian relaxation presented in Section 2.2 – *e.g.* C&W attack [28].

Remark 14. *Note that we only use a dot product for the projection operator to be well defined. But any notion of distance with a well-defined projection works alike.*

In this context, the best responses for the defender remain unchanged; hence we only focus on characterizing the set of best responses for the adversary. The new best response we get for the adversary shares a fundamental similarity with the previous one: the optimal attack will only change points that are close enough to the decision boundary. However, with our new penalty all attacked points are projected on the decision boundary. The proof is very similar to Lemma 1, but we display it below for completeness.

Lemma 2. *Let $h \in \mathcal{H}$ and $\boldsymbol{\psi} \in \mathrm{BR}_\Omega(h)$. Then the following assertion holds:*

$$\boldsymbol{\psi}_1(\boldsymbol{x}) = \begin{cases} \mathrm{proj}(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in P_h(\alpha_p) \\ \boldsymbol{x} & \text{otherwise.} \end{cases}$$

Where proj *is the orthogonal projection on* $(P_h)^{\mathsf{c}}$. $\boldsymbol{\psi}_{-1}$ *is characterized symmetrically.*

Proof. Let us first simplify the worst-case adversarial score for h.

$$\begin{aligned} &\sup_{\boldsymbol{\psi}\in\left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \mathrm{Score}^{\mathrm{adv}}_\Omega(h, \boldsymbol{\psi}) \\ &= \sum_{y=\pm 1} \nu(y) \sup_{\boldsymbol{\psi}_y\in\mathcal{F}_{\mathcal{X}|\alpha_p}} \mathbb{E}_{\boldsymbol{x}\sim\mu_y}[\mathbb{1}\{h(\boldsymbol{\psi}_y(\boldsymbol{x}))y \leq 0\} - \lambda\|\boldsymbol{x} - \boldsymbol{\psi}_y(\boldsymbol{x})\|]\,. \end{aligned}$$

Finding $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_{-1}$ are two independent optimization problems, hence, we focus on characterizing $\boldsymbol{\psi}_1$ – *i.e.* $y = 1$.

$$\begin{aligned} &\sup_{\boldsymbol{\psi}_1\in\mathcal{F}_{\mathcal{X}|\alpha_p}} \mathbb{E}_{\boldsymbol{x}\sim\mu_1}[\mathbb{1}\{h(\boldsymbol{\psi}_1(\boldsymbol{x})) \leq 0\} - \lambda\|\boldsymbol{x} - \boldsymbol{\psi}_1(\boldsymbol{x})\|] \\ &= \int_{\mathcal{X}} \operatorname*{essup}_{\boldsymbol{z}\in B_p(\boldsymbol{x},\alpha_p)} \mathbb{1}\{h(\boldsymbol{z}) \leq 0\} - \lambda\|\boldsymbol{x} - \boldsymbol{z}\|\; d\mu_1(\boldsymbol{x}) \end{aligned}$$

$$= \sum_{j \in J} \int_{H_j} \operatorname*{essup}_{\boldsymbol{z} \in B_p(\boldsymbol{x}, \alpha_p)} \mathbb{1}\{h(\boldsymbol{z}) \leq 0\} - \lambda \|\boldsymbol{x} - \boldsymbol{z}\| \; d\mu_1(\boldsymbol{x}),$$

where $(H_j)_{j \in J}$ is a partition of $\mathcal{X}$. In particular, we take $H_0 = P_h^{\mathsf{c}}$, $H_1 = P_h \setminus P_h(\alpha_p)$, and $H_2 = P_h(\alpha_p)$. Then we can study the three sets independently.

1. For $\boldsymbol{x} \in H_0 = P_h^{\mathsf{c}}$, taking $\boldsymbol{z} = \boldsymbol{x}$ gives us $\mathbb{1}\{h(\boldsymbol{z}) \leq 0\} - \lambda \|\boldsymbol{x} - \boldsymbol{z}\| = 1$. Since for any $\boldsymbol{z} \in \mathcal{X}$ we have $\mathbb{1}\{h(\boldsymbol{z}) \leq 0\} - \lambda \|\boldsymbol{x} - \boldsymbol{z}\| \leq 1$, this strategy is optimal. Furthermore, for any other optimal strategy $\boldsymbol{z}'$, we would have $\|\boldsymbol{x} - \boldsymbol{z}'\| = 0$, hence $\boldsymbol{z}' = \boldsymbol{x}$, and an optimal attack will never move the points of $H_0 = P_h^{\mathsf{c}}$.

2. For $\boldsymbol{x} \in H_1 = P_h \setminus P_h(\alpha_p)$. We have $B_p(\boldsymbol{x}, \alpha_p) \subset P_h$ by definition of $P_h(\alpha_p)$. Hence, for any $\boldsymbol{z} \in B_p(\boldsymbol{x}, \alpha_p)$, one gets $h(\boldsymbol{z}) > 0$. Then $\mathbb{1}\{h(\boldsymbol{z}) \leq 0\} - \lambda \|\boldsymbol{x} - \boldsymbol{z}\| \leq 0$. The only optimal $\boldsymbol{z}$ will thus be $\boldsymbol{z} = \boldsymbol{x}$, giving value 0.

3. Let us now consider $\boldsymbol{x} \in H_2 = P_h(\alpha_p)$ which is the interesting case where an attack is possible. We know that $B_p(\boldsymbol{x}, \alpha_p) \cap P_h^{\mathsf{c}} \neq \emptyset$, and for any $\boldsymbol{z}$ in this intersection, $\mathbb{1}\{h(\boldsymbol{z}) \leq 0\} = 1$. Hence :

$$\begin{aligned}
&\operatorname*{essup}_{\boldsymbol{z} \in B_p(\boldsymbol{x}, \alpha_p)} \mathbb{1}\{h(\boldsymbol{z}) \leq 0\} - \lambda \|\boldsymbol{x} - \boldsymbol{z}\| \\
&= \max\left(1 - \lambda \operatorname*{essinf}_{\boldsymbol{z} \in B_p(\boldsymbol{x}, \alpha_p) \cap P_h^{\mathsf{c}}} \|\boldsymbol{x} - \boldsymbol{z}\|, 0\right) \\
&= \max\left(1 - \lambda \|\boldsymbol{x} - \operatorname{proj}_{B_p(\boldsymbol{x}, \alpha_p) \cap P_h^{\mathsf{c}}}(\boldsymbol{x})\|, 0\right)
\end{aligned}$$

Where $\operatorname{proj}_{B_p(\boldsymbol{x}, \alpha_p) \cap P_h^{\mathsf{c}}}$ is the projection on the closure of $B_p(\boldsymbol{x}, \alpha_p) \cap P_h^{\mathsf{c}}$[a]. Finally, let us remark that, since $\lambda \in (0, 1)$ and $\alpha_p \leq 1$, one has

$$1 - \lambda \|\boldsymbol{x} - \operatorname{proj}_{B_p(\boldsymbol{x}, \alpha_p) \cap P_h^{\mathsf{c}}}(\boldsymbol{x})\| \geq 0$$

for any $\boldsymbol{x} \in H_2$. Hence, on $P_h(\alpha_p)$, the optimal attack projects all the points on P_h^{c}.

Finally, since $H_0 \cup H_1 \cup H_2 = \mathcal{X}$, Lemma 2 holds. Furthermore, the worst-case adversarial score writes

$$\begin{aligned}
&\sup_{\psi \in \left(\mathcal{F}_{\mathcal{X}|\alpha_p}\right)^2} \operatorname{Score}_{\Omega}^{\mathrm{adv}}(h, \psi) \\
&= \sum_{y = \pm 1} \nu(y) \sum_{j \in J} \int_{H_j} \operatorname*{essup}_{\boldsymbol{z} \in B_p(\boldsymbol{x}, \alpha_p)} \mathbb{1}\{h(\boldsymbol{z}) y \leq 0\} - \lambda \|\boldsymbol{x} - \boldsymbol{z}\| \; d\mu_y(x) \,.
\end{aligned}$$

Since the value is 0 on $P_h \setminus P_h(\alpha_p)$ for ψ_1 – resp. on $N_h \setminus N_h(\alpha_p)$ for ψ_{-1} – one gets

$$= \mathcal{R}(h) + \nu(1) \int_{P_h(\alpha_p)} \big(1 - \lambda \|\boldsymbol{x} - \text{proj}(\boldsymbol{x})\|\big) d\mu_1(\boldsymbol{x})$$
$$+ \nu(-1) \int_{N_h(\alpha_p)} \big(1 - \lambda \|\boldsymbol{x} - \text{proj}(\boldsymbol{x})\|\big) d\mu_{-1}(\boldsymbol{x}) \,.$$

□

[a] Note that $\text{proj}_{B_p(\boldsymbol{x},\alpha_p) \cap P_h^c}$ exists. Indeed h is continuous, so $B_p(\boldsymbol{x}, \alpha_p) \cap P_h^c$ is a closed and bounded set; thus compact – since we are in finite dimension. The projection is however not guaranteed to be unique since we have no evidence on the convexity of the set.

Note that, in practice, it might be computationally hard to generate the exact best response – *i.e.* the projection on P_h^c. That will happen for example if the decision boundary is very complex – *e.g.* highly non-smooth – or when $\mathcal{X}$ is in a high-dimensional space. To keep the attack tractable, the adversary will have to compute an approximate best response by allowing the projection to reach points within a small ball around the boundary. This means that the best responses for the new penalized problem will sometimes match the best response for the previous one.

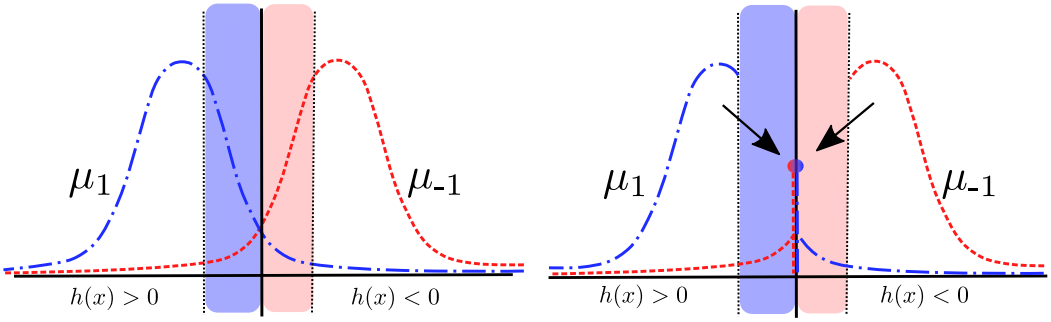


Figure 3.4: Illustration of the conditional distributions μ_{-1} and μ_1. On the left: without attack. On the right: under penalized attack with the new penalty. Blue and red zones are respectively the sets $P_h(\alpha_p)$ and $N_h(\alpha_p)$.

As for the previous penalty, we illustrate in Figure 3.4 the the non-existence of a Pure Nash Equilibrium with two uni-dimensional Gaussian distributions. We can see – on the right – that the mass of μ_1 that was in $P_h(\alpha_p)$ is transported on a Dirac distribution at the decision boundary. Similarly to the previous penalty, the Bayes optimal classifier for the new distribution will predict -1 for the zone $P_h(\alpha_p)$, hence Theorem 7 holds with exactly the same proof as above. Finally, let us present an adaptation of Theorem 8 to our new penalty. The statement is almost the same, with the only difference that we have to interpolate on the bound of the perturbation, getting a new condition: $q_1 > \max(1 - \lambda\delta, \lambda(\alpha_p - \delta))$ with $\delta \in (0, \alpha_p)$. The proof follows the same steps as before but because of the proj operator, some more calculus is needed.

Remark 15. *For the condition on q_1 to make sens, we also need that* $\max(1-\lambda\delta, \lambda(\alpha_p-\delta)) < 1$. *This will hold in particular when $\alpha_p \leq 1$ which is a standard assumption considering the threshold we have discussed so far. In the remaining we will consider that $\alpha_p \leq 1$ accordingly.*

Theorem 9 (Randomization matters bis)**.** *Let us consider $\alpha_p \leq 1$, $h_1 \in \mathcal{H}$, $\lambda \in (0,1)$, $\delta \in (0, \alpha_p)$ and $\boldsymbol{\psi} \in \mathrm{BR}_\Omega(h_1)$. Then there exists h_2 such that, for any $q_1 \in (\max(1-\lambda\delta, \lambda(\alpha_p - \delta)), 1)$ and for any $\boldsymbol{\psi}' \in \mathrm{BR}_\Omega(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}})$ one has*

$$\mathrm{Score}_\Omega^{\mathrm{adv}}(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}, \boldsymbol{\psi}') < \mathrm{Score}_\Omega^{\mathrm{adv}}(h_1, \boldsymbol{\psi}).$$

Where $\boldsymbol{h} = (h_1, h_2)$, $\boldsymbol{q} = (q_1, 1-q_1)$, and $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$ is the mixture of $\boldsymbol{h}$ by $\boldsymbol{q}$.

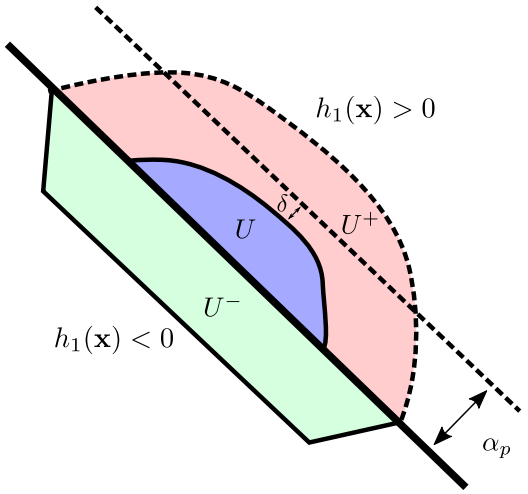


Figure 3.5: Illustration of the notations U, U^+, U^- and δ for proof of Theorem 9.

Proof. Let us take $U \subset P_{h_1}(\alpha_p)$ such that

$$\min_{\boldsymbol{x} \in U} \|\boldsymbol{x} - \mathrm{proj}_{P_h \setminus P_h(\alpha_p)}(\boldsymbol{x})\| = \delta \in (0, \alpha_p).$$

We construct h_2 as follows.

$$h_2(\boldsymbol{x}) = \begin{cases} -h_1(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in U \\ h_1(\boldsymbol{x}) & \text{otherwise.} \end{cases}$$

This means that h_2 changes the class of all points in U, and do not change the rest. Let $q_1 \in (0,1)$, the corresponding mixture $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$, and $\boldsymbol{\psi}' \in \mathrm{BR}_\Omega(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}})$. We will find a condition on q_1 so that the score of $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$ is lower than the score of h_1. Recall that

$$\mathrm{Score}_\Omega^{\mathrm{adv}}(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}, \boldsymbol{\psi}')$$

$$= \nu(1) \int_{\mathcal{X}} \operatorname*{essup}_{\boldsymbol{z} \in B_p(\boldsymbol{x}, \alpha_p)} q_1 \mathbb{1}\{h_1(\boldsymbol{z}) \leq 0\} + (1 - q_1) \mathbb{1}\{h_2(\boldsymbol{z}) \leq 0\} - \lambda \|\boldsymbol{x} - \boldsymbol{z}\| \, d\mu_1(\boldsymbol{x})$$

$$+ \nu(\text{-}1) \int_{\mathcal{X}} \operatorname*{essup}_{\boldsymbol{z} \in B_p(\boldsymbol{x}, \alpha_p)} q_1 \mathbb{1}\{h_1(\boldsymbol{z}) \geq 0\} + (1 - q_1) \mathbb{1}\{h_2(\boldsymbol{z}) \geq 0\} - \lambda \|\boldsymbol{x} - \boldsymbol{z}\| \, d\mu_{\text{-}1}(\boldsymbol{x}) \, .$$

As we discussed in the proof of Theorem 8, the only terms that may vary between the score of h_1 and the score of $\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}$ are the integrals on U, $U \oplus \alpha_p \cap P_{h_1}$ and $\boldsymbol{\psi}_{\text{-}1}^{-1}(U)$. Hence, for simplicity, we only write those terms. Furthermore, we denote

$$U^+ := U \oplus \alpha_p \cap P_{h_1} \setminus U, \; U^- := \boldsymbol{\psi}_{\text{-}1}^{-1}(U) \text{ and } P_{\alpha_p} := P_{h_1}(\alpha_p).$$

One can refer to Figure 3.5 for a visual interpretation of these sets. We can now evaluate the worst-case adversarial score for h_1 restricted to the above sets. Thanks to Lemma 2 that characterizes $\boldsymbol{\psi}$, we can write

$$\begin{aligned}
&\operatorname{Score}_{\Omega}^{\text{adv}}(h_1, \boldsymbol{\psi}) \\
&= \nu(1) \int_{U} \Big(1 - \lambda \|\boldsymbol{x} - \operatorname{proj}_{P_{h_1}^{\text{c}}}(\boldsymbol{x})\|\Big) d\mu_1(\boldsymbol{x}) + \nu(\text{-}1) \mu_{\text{-}1}(U) \\
&+ \nu(1) \int_{U^+ \setminus P_{\alpha_p}} 0 \, d\mu_1(\boldsymbol{x}) + \nu(\text{-}1) \mu_{\text{-}1}\big(U^+ \setminus P_{\alpha_p}\big) \\
&+ \nu(1) \int_{U^+ \cap P_{\alpha_p}} \Big(1 - \lambda \|\boldsymbol{x} - \operatorname{proj}_{P_{h_1}^{\text{c}}}(\boldsymbol{x})\|\Big) d\mu_1(\boldsymbol{x}) + \nu(\text{-}1) \mu_{\text{-}1}\big(U^+ \cap P_{\alpha_p}\big) \\
&+ \nu(1) \mu_1\big(U^-\big) + \nu(\text{-}1) \int_{U^-} \Big(1 - \lambda \|\boldsymbol{x} - \operatorname{proj}_{U}(\boldsymbol{x})\|\Big) d\mu_{\text{-}1}(\boldsymbol{x}).
\end{aligned}$$

Similarly we can evaluate the worst-case adversarial score for the mixture,

$$\begin{aligned}
&\operatorname{Score}_{\Omega}^{\text{adv}}(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}}, \boldsymbol{\psi}') \\
&= \nu(1) \int_{U} \max\Big(1 - q_1, 1 - \lambda \|\boldsymbol{x} - \operatorname{proj}_{P_{h_1}^{\text{c}}}(\boldsymbol{x})\|\Big) \; d\mu_1(\boldsymbol{x}) \\
&+ \nu(\text{-}1) \int_{U} \max(q_1, 1 - \lambda \|\boldsymbol{x} - \operatorname{proj}_{U^+}(\boldsymbol{x})\|) \; d\mu_{\text{-}1}(\boldsymbol{x}) \\
&+ \nu(1) \int_{U^+ \setminus P_{\alpha_p}} \max(0, 1 - q_1 - \lambda \|\boldsymbol{x} - \operatorname{proj}_{U}(\boldsymbol{x})\|) \; d\mu_1(\boldsymbol{x}) + \nu(\text{-}1) \mu_{\text{-}1}\big(U^+ \setminus P_{\alpha_p}\big) \\
&+ \nu(1) \int_{U^+ \cap P_{\alpha_p}} \max\Big(1 - q_1 - \lambda \|\boldsymbol{x} - \operatorname{proj}_{U}(\boldsymbol{x})\|, 1 - \lambda \|\boldsymbol{x} - \operatorname{proj}_{P_{h_1}^{\text{c}}}(\boldsymbol{x})\|\Big) \; d\mu_1(\boldsymbol{x})
\end{aligned}$$

$$+\nu(\text{-}1)\mu_{\text{-}1}\big(U^+\cap P_{\alpha_p}\big)+\nu(1)\mu_1\big(U^-\big)$$

$$+\nu(\text{-}1)\int_{U^-}\max\Big(0,1-\lambda\|\boldsymbol{x}-\mathrm{proj}_{N_{h_1}^{\mathsf{c}}\setminus U}(\boldsymbol{x})\|,q_1-\lambda\|\boldsymbol{x}-\mathrm{proj}_U(\boldsymbol{x})\|\Big)d\mu_{\text{-}1}(\boldsymbol{x}).$$

Note that we need to take into account the special case of the points in the dilation that were already in the attacked zone before, and that can now be attacked in two ways, either by projecting on U – but that works with probability q_1, since the classification on U is now randomized – or by projecting on $P_{h_1}^{\mathsf{c}}$, which works with probability 1 but may use more distance and so pay more penalty. We can now compute the difference between both scores.

$$\mathrm{Score}_\Omega^{\mathrm{adv}}(h_1,\boldsymbol{\psi})-\mathrm{Score}_\Omega^{\mathrm{adv}}(\mathfrak{m}_{\boldsymbol{h}}^{\boldsymbol{q}},\boldsymbol{\psi}') \tag{3.19}$$

$$=\nu(1)\int_U 1-\lambda\|\boldsymbol{x}-\mathrm{proj}_{P_{h_1}^{\mathsf{c}}}(\boldsymbol{x})\|-\max\Big(1-q_1,1-\lambda\|\boldsymbol{x}-\mathrm{proj}_{P_{h_1}^{\mathsf{c}}}(\boldsymbol{x})\|\Big)d\mu_1(\boldsymbol{x}) \tag{3.20}$$

$$+\nu(\text{-}1)\int_U 1-\max(q_1,1-\lambda\|\boldsymbol{x}-\mathrm{proj}_{U^+}(\boldsymbol{x})\|)d\mu_{\text{-}1}(\boldsymbol{x}) \tag{3.21}$$

$$-\nu(1)\int_{U^+\setminus P_{\alpha_p}}\max(1-q_1-\lambda\|\boldsymbol{x}-\mathrm{proj}_U(\boldsymbol{x})\|,0)d\mu_1(\boldsymbol{x}) \tag{3.22}$$

$$+\nu(1)\int_{U^+\cap P_{\alpha_p}}1-\lambda\|\boldsymbol{x}-\mathrm{proj}_{P_{h_1}^{\mathsf{c}}}(\boldsymbol{x})\|$$

$$-\max\Big(1-q_1-\lambda\|\boldsymbol{x}-\mathrm{proj}_U(\boldsymbol{x})\|,1-\lambda\|\boldsymbol{x}-\mathrm{proj}_{P_{h_1}^{\mathsf{c}}}(\boldsymbol{x})\|\Big)d\mu_1(\boldsymbol{x}) \tag{3.23}$$

$$+\nu(\text{-}1)\int_{U^-}1-\lambda\|\boldsymbol{x}-\mathrm{proj}_U(\boldsymbol{x})\|$$

$$-\max\Big(0,1-\lambda\|\boldsymbol{x}-\mathrm{proj}_{N_{h_1}^{\mathsf{c}}\setminus U}(\boldsymbol{x})\|,q_1-\lambda\|\boldsymbol{x}-\mathrm{proj}_U(\boldsymbol{x})\|\Big)d\mu_{\text{-}1}(\boldsymbol{x}). \tag{3.24}$$

First recall that both μ_1 and $\mu_{\text{-}1}$ have full support. Let us simplify Equation (3.19) using using additional assumptions.

- First, note that Equation (3.21)>0. Then a sufficient condition for the difference to be strictly positive is to ensure that other lines are ≥ 0.
- In particular to have (3.20) ≥ 0 it is sufficient to have for all $\boldsymbol{x}\in U$

$$\max\Big(1-q_1,1-\lambda\|\boldsymbol{x}-\mathrm{proj}_{P_{h_1}^{\mathsf{c}}}(\boldsymbol{x})\|\Big)=1-\lambda\|\boldsymbol{x}-\mathrm{proj}_{P_{h_1}^{\mathsf{c}}}(\boldsymbol{x})\|.$$

  This gives us $q_1\geq\lambda(\alpha_p-\delta)\geq\lambda\max_{\boldsymbol{x}\in U}\|\boldsymbol{x}-\mathrm{proj}_{P_{h_1}^{\mathsf{c}}}(\boldsymbol{x})\|$.

- Similarly, to have (3.22) ≥ 0, we should set for all $\boldsymbol{x} \in U^+ \setminus P_{\alpha_p}$

$$q_1 \geq 1 - \lambda \|\boldsymbol{x} - \text{proj}_U(\boldsymbol{x})\|.$$

  Since $\min_{x \in U^+ \setminus P_{\alpha_p}} \|\boldsymbol{x} - \text{proj}_U(\boldsymbol{x})\| = \delta$, we get the condition $q_1 \geq 1 - \lambda\delta$.

- Finally (3.24) ≥ 0, since by definition of U^-, for any $\boldsymbol{x} \in U^-$ we have

$$\|\boldsymbol{x} - \text{proj}_{N_{h_1}^c \setminus U}(\boldsymbol{x})\| \geq \|\boldsymbol{x} - \text{proj}_U(\boldsymbol{x})\|.$$

Finally, by summing all these simplifications, we have (3.19) > 0. Hence the result holds for any $q_1 > \max(1 - \lambda\delta, \lambda(\alpha_p - \delta))$. □

3.6 Lessons learned and future works

In this chapter, we provided a new point of view on the problem of classification under perturbation – Problem (1.3). Based on simple tools from game theory, we demonstrated that adding some regularization can fundamentally modify the nature of the game between the adversary and the defender – model provider. This analysis led us to investigate randomized hypothesis classes. Both our theoretical findings and empirical validation prove the efficacy of this method and thus provide a first answer to **Q1**:

> *There might be a class of* randomized hypotheses $\mathcal{H}$ *for which the adversarial risk minimization problem has a solution* $\boldsymbol{h}^*$ *with small adversarial risk*

In Chapters 4 and 5 we will further investigate some specific classes and show that we can obtain both robustness and accuracy – to some extent. Nevertheless, several questions remain open. We list here some of them that we aim to investigate in the future.

Future work 1: The equilibrium in the randomized regime

There remains to study whether an equilibrium exists in the randomized regime. This question is appealing from a theoretical point of view, and requires to investigate the space of randomized adversaries $\mathcal{P}((\mathcal{F}_{\mathcal{X}|\alpha_p})^2)$ which implies more technicalities. The study of this equilibrium is also tightly related to that of the value of the game, which would be interesting for obtaining min-max bounds on the accuracy under attack.

Future work 2: Study the duality gap

For now, Theorem 7 shows that there is no Pure Nash Equilibrium in the game, meaning that strong duality does not hold. But it does not indicate how distant the values from the $\inf - \sup$ and the $\sup - \inf$ problem are – *a.k.a.* the duality gap. Evaluating this duality gap could help us build a finer analysis on the impact of regularization on the game.

Future work 3: Boosted adversarial training, a certified defense?

Although the experimental results show that our mixture of hypotheses outperforms adversarial training, the algorithm we present do not provide guarantees in terms of certified accuracy. As the literature on adversarial attacks and defenses demonstrated, better attacks always exist. This is why, we need to further study the theoretical aspects of our procedure, to prove the robustness of the mixtures we design.

4 Theoretical analysis of randomized classifiers

Contents

Q1: *Is there some hypothesis class $\mathcal{H}$ for which the adversarial risk minimization problem has a solution $\boldsymbol{h}^*$ with small adversarial risk?*

Q2: *Can we find a class $\mathcal{H}$ and a hypothesis $\boldsymbol{h}^* \in \mathcal{H}$ that simultaneously has small standard and adversarial risks?*

In Chapter 3, we identified randomized hypotheses as good candidates to build more robust classifiers; thus partially answering **Q1**. Here, we keep answering **Q1** and provide preliminary answers to **Q2** by studying randomized classifiers through the prism of learning theory. In Section 4.1, we define this class and adapt the notions of risk and robustness to account for the internal random state of the classifiers. In particular, we use the total variation distance to define

robustness as a local Lipschitz condition from $\mathcal{X}$ to $\mathcal{P}(\mathcal{Y})$. We show in Section 4.2 that under this robustness assumption, we can bound the difference between the standard risk and the adversarial risk of any randomized classifier. This answers **Q2** by evaluating the maximal trade-off between robustness and accuracy of the classifier. Then, we devise bounds on the standard generalization gap of this hypothesis class in Section 4.3 and discuss the consequences and limitations of our result. In Section 4.4 we analyze the stability of randomized classifiers with respect to their mode and the implications it has on our understanding of randomized smoothing. Finally, we extend our results to the Renyi divergence, present some additional results and summarize our findings respectively in Sections 4.5 and 4.6.

4.1 Terminology for randomized classifiers

Notations. *For any set $\mathcal{Z}$ with σ-algebra $\mathcal{A}(\mathcal{Z})$, if there is no ambiguity on the considered σ-algebra, we denote $\mathcal{P}(\mathcal{Z})$ the set of all probability measures over $(\mathcal{Z}, \mathcal{A}(\mathcal{Z}))$. For $\rho \in \mathcal{P}(\mathcal{Y})$, we sometimes denote $\rho = (\rho_1, \dots, \rho_K)$ where for every $i \in \{1, \dots, K\}$, ρ_i is the probability under ρ to get i* – i.e. $\rho(i)$. *Depending on the context, we use one or the other notation without further distinction.*

4.1.1 Definitions on randomized classifiers

This chapter's goal is to analyze how randomized classifiers could constitute good candidates to solve the problem of classification under perturbation. For this, we come back to the general K-class classification setting. We have $\mathcal{X} \subset [0,1]^d$, $\mathcal{Y} = \{1, \dots, K\}$ and $\mathcal{D}$ the ground-truth distribution on $\mathcal{X} \times \mathcal{Y}$. Let us start by defining what we mean by randomized classifiers.

Definition 5 (Probabilistic mapping)**.** *Let $\mathcal{Z}$ and $\mathcal{Z}'$ be two arbitrary spaces. A* probabilistic mapping *from $\mathcal{Z}$ to $\mathcal{Z}'$ is a mapping $\mathfrak{m} : \mathcal{Z} \to \mathcal{P}(\mathcal{Z}')$. When $\mathcal{Z} = \mathcal{X}$ and $\mathcal{Z}' = \mathcal{Y}$, $\mathfrak{m}$ is called a* randomized classifier. *To get a numerical answer out of $\mathfrak{m}$ for an input $\boldsymbol{x}$, we sample $\hat{y} \sim \mathfrak{m}(\boldsymbol{x})$.*

Remark 16. *In Chapter 3, we discussed the properties of mixtures of hypotheses. The above definition generalizes the previous one. Furthermore, we can consider any mapping as a probabilistic mapping whether it explicitly considers randomization or not. In fact, any deterministic classifier can be seen as a randomized one, since we can characterize it with a Dirac measure.*

As we previously did for randomized hypotheses, we can adapt the concepts of risk and adversarial risk for a randomized classifier. The loss function we use is the natural extension of the $0/1$ loss to the randomized regime. Given a randomized classifier $\mathfrak{m}$ and a sample $(\boldsymbol{x}, y) \sim \mathcal{D}$ it writes

$$\mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}), y) := \mathbb{E}_{\hat{y} \sim \mathfrak{m}(\boldsymbol{x})}[\mathbb{1}\{\hat{y} \neq y\}]. \tag{4.1}$$

This loss function evaluates the probability of misclassification of $\mathfrak{m}$ on a data sample $(\boldsymbol{x}, y) \sim \mathcal{D}$. Accordingly, the risk of $\mathfrak{m}$ with respect to $\mathcal{D}$ writes

$$\mathcal{R}(\mathfrak{m}) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big[\mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}), y)\big]. \tag{4.2}$$

Finally, given $\mathfrak{m}$ and $(\boldsymbol{x}, y) \sim \mathcal{D}$, the adversary seeks a perturbation $\boldsymbol{\tau} \in \mathcal{X}$ such that $\|\boldsymbol{\tau}\|_p \leq \alpha_p$ that maximizes the expected error of the classifier on $\boldsymbol{x}$ – *i.e.* $\mathbb{E}_{\hat{y}\sim\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})}[\mathbb{1}\{\hat{y} \neq y\}]$. Therefore, the adversarial risk of $\mathfrak{m}$ under α_p-bounded perturbations writes

$$\mathcal{R}^{\text{adv}}(\mathfrak{m}; \alpha_p) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}), y)\right]. \tag{4.3}$$

4.1.2 Robustness for randomized classifiers

We could define the notion of robustness for a randomized classifier depending on whether it misclassifies any test sample $(\boldsymbol{x}, y) \sim \mathcal{D}$. But in practice, neither the adversary nor the model provider have access to the ground-truth distribution $\mathcal{D}$. Furthermore, in real-world scenarios – *e.g.* the autonomous car – we want to check before its deployment whether the model is robust. Therefore, we want the classifier to be stable on the regions of the space where it already classifies correctly. Formally a – deterministic – classifier $c : \mathcal{X} \to \mathcal{Y}$ is called *robust* if for any $(\boldsymbol{x}, y) \sim \mathcal{D}$ such that $c(\boldsymbol{x}) = y$, and for any $\boldsymbol{\tau} \in \mathcal{X}$ one has

$$\|\boldsymbol{\tau}\|_p \leq \alpha_p \implies c(\boldsymbol{x}) = c(\boldsymbol{x}+\boldsymbol{\tau}). \tag{4.4}$$

By analogy with this notion, we define robustness for a randomized classifier as follows.

Definition 6 (robustness for a randomized classifier)**.** *A randomized classifier* $\mathfrak{m} : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ *is called* (α_p, ϵ)-robust w.r.t. D *if for any* $\boldsymbol{x}, \boldsymbol{\tau} \in \mathcal{X}$*, one has*

$$\|\boldsymbol{\tau}\|_p \leq \alpha_p \implies D(\mathfrak{m}(\boldsymbol{x}), \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})) \leq \epsilon\,.$$

Where D is a metric/divergence between two probability measures. Moreover, given such a metric D, we denote $\mathfrak{M}_D(\alpha_p, \epsilon)$ the set of all randomized classifiers that are (α_p, ϵ)-robust w.r.t. D.

Remark 17. *Note that we did not add the constraint that $\mathfrak{m}$ classifies well on $(\boldsymbol{x}, y) \sim \mathcal{D}$, since it is already encompassed in the probability distribution itself. If the two probabilities $\mathfrak{m}(\boldsymbol{x})$ and $\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})$ are close, and if $\mathfrak{m}(\boldsymbol{x})$ outputs y with high probability, then it will be the same for $\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})$.*

This formulation naturally raises the question of the choice of metric D we should use to defend against adversarial attacks. Any choice of metric/divergence will instantiate a notion of adversarial robustness, and it should be carefully selected. In the present work, we focus our study on the total variation distance and extend our results to the Renyi divergence in Section 2.4. The question whether these metrics/divergences are more appropriate than others remains open but these two divergences are sufficiently general to cover a wide range of other definitions – see Section 2.4 for more details. Furthermore, these notions of distance comply with both a high level – Chapter 4 – and a more practical analysis – Chapter 5.

Let us now recall the definition of total variation distance. Let $\mathcal{Z}$ be an arbitrary space, and ρ, ρ' be two measures in $\mathcal{P}(\mathcal{Z})$. The *total variation distance* between ρ and ρ' is

$$D_{TV}\left(\rho, \rho'\right) := \sup_{Z\subset\mathcal{A}(\mathcal{Z})} |\rho(Z) - \rho'(Z)|\ . \tag{4.5}$$

The total variation distance is one of the most commonly used probability metrics. It admits several very simple interpretations, and is a very useful tool in many mathematical fields such as probability theory, Bayesian statistics or optimal transport [129, 137, 163]. In optimal transport, it can be rewritten as the solution of the Monge-Kantorovich problem with the cost function $cost(\boldsymbol{z}, \boldsymbol{z}') = \mathbb{1}\{\boldsymbol{z} \neq \boldsymbol{z}'\}$,

$$D_{TV}(\rho, \rho') = \inf \int_{\mathcal{Z}^2} \mathbb{1}\{\boldsymbol{z} \neq \boldsymbol{z}'\} d\pi(\boldsymbol{z}, \boldsymbol{z}') \ , \tag{4.6}$$

where the infimum is taken over all joint probability measures π in $\mathcal{P}(\mathcal{Z} \times \mathcal{Z})$ with marginals ρ and ρ'. According to this interpretation, it seems quite standard to consider the total variation distance as a relaxation of the trivial distance on $[0, 1]$ – see *e.g.* [163, chap 1] for details. In the remaining we denote $\mathfrak{M}_{TV}(\alpha_p, \epsilon)$ the set of all (α_p, ϵ) robust classifiers *w.r.t.* D_{TV}.

4.2 Risks' gap for robust randomized classifiers

4.2.1 An additive bound for the risks' gap

As we discussed in Section 2.4, we can always decompose the adversarial risk of a classifier $\mathfrak{m}$ in two terms. First the standard risk of $\mathfrak{m}$ and second the amount of risk the adversary creates with non-zero perturbations

$$\mathcal{R}^{\text{adv}}(\mathfrak{m}; \alpha_p) = \mathcal{R}(\mathfrak{m}) + \mathcal{R}^{\text{adv}}_{>0}(\mathfrak{m}; \alpha_p). \tag{4.7}$$

Hence minimizing $\mathcal{R}(\mathfrak{m})$ can give poor values for $\mathcal{R}^{\text{adv}}(\mathfrak{m}; \alpha_p)$ and vise-versa. In this section, we upper-bound $\mathcal{R}^{\text{adv}}_{>0}(\mathfrak{m}; \alpha_p)$ to simplify the learning procedure. Specifically, let us consider $\mathfrak{m}$ in the class of (α_p, ϵ)-robust classifiers *w.r.t.* D_{TV}. Then we can control the loss of accuracy under attack of this classifier with the robustness parameter ϵ.

Theorem 10 (Risk's gap for TV-robust classifiers)**.** *Let* $\mathfrak{m} \in \mathfrak{M}_{TV}(\alpha_p, \epsilon)$ *. Then we have*

$$\mathcal{R}^{\text{adv}}(\mathfrak{m}; \alpha_p) \leq \mathcal{R}(\mathfrak{m}) + \epsilon \, .$$

Proof. Let $\mathfrak{m}$ be an (α_p, ϵ)-robust classifier *w.r.t.* D_{TV} , $(\boldsymbol{x}, y) \sim \mathcal{D}$ and $\boldsymbol{\tau} \in \mathcal{X}$ such that $\|\boldsymbol{\tau}\|_p \leq \alpha_p$. By definition of the $0/1$ loss we have

$$\mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x} + \boldsymbol{\tau}), y) = \mathbb{E}_{\hat{y} \sim \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})}[\mathbb{1}\{\hat{y} \neq y\}].$$

Furthermore, by definition of the total variation distance we have

$$\mathbb{E}_{\hat{y} \sim \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})}[\mathbb{1}\{\hat{y} \neq y\}] - \mathbb{E}_{\hat{y} \sim \mathfrak{m}(\boldsymbol{x})}[\mathbb{1}\{\hat{y} \neq y\}] \leq D_{TV}(\mathfrak{m}(\boldsymbol{x}), \mathfrak{m}(\boldsymbol{x} + \boldsymbol{\tau})).$$

Since $\mathfrak{m} \in \mathfrak{M}_{TV}(\alpha_p, \epsilon)$, the above amounts to write

$$\mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x} + \boldsymbol{\tau}), y) - \mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}), y) \leq \epsilon.$$

Finally, this holds for any $(\boldsymbol{x}, y) \sim \mathcal{D}$ and any α_p bounded perturbation $\boldsymbol{\tau}$, then we get

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}), y)\right] - \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big[\mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}), y)\big] \leq \epsilon.$$

The above inequality concludes the proof. □

This result means that if we can design a class $\mathfrak{M}_{TV}(\alpha_p, \epsilon)$ with small enough ϵ, then minimizing the risk of $\mathfrak{m} \in \mathfrak{M}_{TV}(\alpha_p, \epsilon)$ is also sufficient to get a good value for the adversarial risk.

4.2.2 Robustness may not be at odds with accuracy.

The above result is relatively easy to obtain, but it has an interesting consequence on the understanding we have of the trade-off between robustness and accuracy. It says that there exists some classes of randomized classifiers for which robustness and standard accuracy may not be at odds, since we can upper-bound the maximal loss of accuracy the model may suffer under attack. This questions previous intuitions developed on deterministic classifiers – see Section 2.4 – and keeps advocating for using randomization schemes as defenses against adversarial attacks. Note, however, that we did not evade the trade-off between robustness and accuracy, we only showed that with certain hypothesis classes it is manageable. Since we can bound the difference between the risk and the adversarial risk of any classifier $\mathfrak{m} \in \mathfrak{M}_{TV}(\alpha_p, \epsilon)$, we now want to minimize the risk over a hypothesis class $\mathfrak{M} \subset \mathfrak{M}_{TV}(\alpha_p, \epsilon)$ to obtain a good approximation for both Problems (1.1) and (1.3). But for this, we still need the empirical risk minimization to converge to a solution with small standard risk. To measure the rate of convergence of the empirical risk toward the risk on $\mathfrak{M}$, we need to upper-bound the Rademacher complexity of $\mathcal{L}_{\mathfrak{M}_{TV}(\alpha_p,\epsilon)}$.

Remark 18. *Remark that this result is not limited to the* $0/1$ *loss. Indeed, any loss function of the form* $(\boldsymbol{x}, y) \mapsto \mathbb{E}_{\hat{y}\sim\mathfrak{m}(\boldsymbol{x})}[\mathcal{L}(\hat{y}, y)]$ *with* $\mathcal{L}$ *non-negative would work alike.*

4.3 Generalization gap for randomized classifiers

4.3.1 Bounding the Rademacher complexity for the total variation

Recall from Chapter 2 that in the supervised learning setting we have access to n *i.i.d.*training examples drawn from $\mathcal{D}$, denoted by $\mathcal{S} := \{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \ldots, (\boldsymbol{x_n}, y_n)\}$. Given a class $\mathfrak{M} \subset \mathfrak{M}_{TV}(\alpha_p, \epsilon)$, we aim to solve the empirical risk minimization problem

$$\inf_{\mathfrak{m}\in\mathfrak{M}} \mathcal{R}_{\mathcal{S}}(\mathfrak{m}) := \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x_i}), y_i)\,. \tag{4.8}$$

Then, to evaluate how far the theoretical risk of the selected classifier $\mathfrak{m}$ is from what we observe on $\mathcal{S}$, we need to upper bound the generalization gap of any $\mathfrak{m} \in \mathfrak{M}_{TV}(\alpha_p, \epsilon)$. To do so, we can to study the empirical Rademacher complexity of

$$\mathcal{L}_{\mathfrak{M}_{TV}(\alpha_p,\epsilon)} := \{(\boldsymbol{x}, y) \mapsto \mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}), y) \text{ s.t. } \mathfrak{m} \in \mathfrak{M}_{TV}(\alpha_p, \epsilon)\}. \tag{4.9}$$

Then, thanks to Theorem 1, for any $\delta \in (0,1)$, and for any $\mathfrak{m} \in \mathfrak{M}_{TV}(\alpha_p, \epsilon)$, the following holds with probability at least $1-\delta$,

$$\mathcal{R}(\mathfrak{m}) \leq \mathcal{R}_{\mathcal{S}}(\mathfrak{m}) + 2\mathfrak{R}_{\mathcal{S}}\left(\mathcal{L}_{\mathfrak{M}_{TV}(\alpha_p,\epsilon)}\right) + 3\sqrt{\frac{\ln(2/\delta)}{2n}} \ . \tag{4.10}$$

Remark 19. *Note that in Theorem 1, there is an additional parameter W where $\|\mathcal{L}\|_\infty \leq W$. Here, by definition of the $0/1$ loss, we have $\left\|\mathcal{L}_{0/1}\right\|_\infty \leq 1$; hence $W = 1$ in Equation (4.10).*

Accordingly, we want to upper bound the empirical Rademacher complexity of $\mathcal{L}_{\mathfrak{M}_{TV}(\alpha_p,\epsilon)}$, which motivates the following definition.

Definition 7 (α-covering and external covering number)**.** *Let us consider $(\mathcal{X}, \|.\|_p)$ a vector space equipped with the ℓ_p norm, $B \subset \mathcal{X}$ and $\alpha \geq 0$. Then*

- *$C = \{\boldsymbol{c_1}, \ldots, \boldsymbol{c_m}\}$ is an α-covering of B for the ℓ_p norm if for any $\boldsymbol{x} \in B$ there exists $\boldsymbol{c_i} \in C$ such that $\|\boldsymbol{x} - \boldsymbol{c_i}\|_p \leq \alpha$.*
- *The external covering number of B writes $N\Big(B, \|.\|_p, \alpha\Big)$. It is the minimal number of points one needs to build an α-covering of B for the ℓ_p norm.*

The covering number is a well-known measure that is often used in statistical learning theory [145] and asymptotic statistics [160] to evaluate the complexity of a set of functions. Here we use it to evaluate the number of ℓ_p balls we need to cover the training samples, which gives us the following bound on the Rademacher complexity of $\mathcal{L}_{\mathfrak{M}_{TV}(\alpha_p,\epsilon)}$.

Theorem 11 (Rademacher complexity TV-robust classifiers)**.** *Let $\mathcal{L}_{\mathfrak{M}_{TV}(\alpha_p,\epsilon)}$ be the loss function class associated with $\mathfrak{M}_{TV}(\alpha_p, \epsilon)$. Then, for any $\mathcal{S} := \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_n}, y_n)\}$, the following holds,*

$$\mathfrak{R}_{\mathcal{S}}\left(\mathcal{L}_{\mathfrak{M}_{TV}(\alpha_p,\epsilon)}\right) \leq \sqrt{\frac{N \times K}{n}} + \epsilon.$$

Where $N = N\Big(\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}, \|.\|_p, \alpha_p\Big)$ is the α_p-external covering number of the inputs $\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}$ for the ℓ_p norm.

Proof. Let us denote $\mathcal{S} := \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_n}, y_n)\}$ and $N = N\Big(\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}, \|.\|_p, \alpha_p\Big)$. By definition of a covering number, there exists $C = \{\boldsymbol{c_1}, \ldots, \boldsymbol{c_N}\}$ an α_p-covering of $\{\boldsymbol{x_1}, \ldots \boldsymbol{x_n}\}$ for the ℓ_p norm. Furthermore, for $j \in \{1, \ldots, N\}$ and $y \in \{1, \ldots, K\}$, we define

$$E_{y,j} = \left\{ i \in \{1, \ldots, n\} \text{ s.t. } y_i = y \text{ and } \underset{l \in \{1,\ldots,N\}}{\arg\min} \|x_i - c_l\| = j \right\}.$$

We also denote $E_j = \bigcup_{y\in[K]} E_{y,j}$. Finally, we denote $\mathcal{L}_{\mathfrak{m}} : (\boldsymbol{x}, y) \mapsto \mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}), y)$. Then, by definition of the empirical Rademacher complexity, we can write

$$\mathfrak{R}_{\mathcal{S}}\left(\mathcal{L}_{\mathfrak{M}_{TV}(\alpha_p,\epsilon)}\right) = \frac{1}{n}\,\mathbb{E}_{r_i}\left[\sup_{\mathfrak{m}\in\mathfrak{M}_{TV}(\alpha_p,\epsilon)} \sum_{i=1}^{n} r_i \mathcal{L}_{\mathfrak{m}}(\boldsymbol{x_i}, y_i)\right].$$

where r_i are *i.i.d.* drawn from a Rademacher distribution – *i.e.* $\mathbb{P}(r_i = 1) = \mathbb{P}(r_i = -1) = \frac{1}{2}$. Then we can use E_j to write

$$\mathfrak{R}_{\mathcal{S}}\left(\mathcal{L}_{\mathfrak{M}_{TV}(\alpha_p,\epsilon)}\right) = \frac{1}{n}\,\mathbb{E}_{r_i}\left[\sup_{\mathfrak{m}\in\mathfrak{M}_{TV}(\alpha_p,\epsilon)} \sum_{j=1}^{N}\sum_{i\in E_j} r_i \mathcal{L}_{\mathfrak{m}}(\boldsymbol{x_i}, y_i)\right].$$

Furthermore for any $\mathfrak{m} \in \mathfrak{M}_{TV}(\alpha_p, \epsilon)$ and $i \in E_j$, there exists $\epsilon_i \in [-\epsilon, \epsilon]$ such that: $\mathcal{L}_{\mathfrak{m}}(\boldsymbol{x_i}, y_i) = \mathcal{L}_{\mathfrak{m}}(\boldsymbol{c_j}, y_i) + \epsilon_i$. Then we have

$$\begin{aligned}\mathfrak{R}_{\mathcal{S}}\left(\mathcal{L}_{\mathfrak{M}_{TV}(\alpha_p,\epsilon)}\right) \leq \frac{1}{n}\,\mathbb{E}_{r_i}&\left[\sup_{\mathfrak{m}\in\mathfrak{M}_{TV}(\alpha_p,\epsilon)} \sum_{j=1}^{N}\sum_{i\in E_j} r_i \mathcal{L}_{\mathfrak{m}}(\boldsymbol{c_j}, y_i)\right] \\ &+ \frac{1}{n}\,\mathbb{E}_{r_i}\left[\sup_{\epsilon_i\in[-\epsilon,\epsilon]} \sum_{j=1}^{N}\sum_{i\in E_j} r_i \epsilon_i\right].\end{aligned}$$

Let us start by studying the second term. We have

$$\frac{1}{n}\,\mathbb{E}_{r_i}\left[\sup_{\epsilon_i\in[-\epsilon,\epsilon]} \sum_{j=1}^{N}\sum_{i\in E_j} r_i \epsilon_i\right] = \frac{1}{n}\,\mathbb{E}_{r_i}\left[\sup_{\epsilon_i\in[-\epsilon,\epsilon]} \sum_{i=1}^{n} r_i \epsilon_i\right] = \frac{1}{n}\sum_{i=1}^{n}\epsilon = \epsilon.$$

Now looking at the first term. Since $\mathcal{L}_{\mathfrak{m}}(\boldsymbol{x}, y) \in [0, 1]$ for all $(\boldsymbol{x}, y)$ we have

$$\begin{aligned}\frac{1}{n}\,\mathbb{E}_{r_i}\left[\sup_{\mathfrak{m}\in\mathfrak{M}_{TV}(\alpha_p,\epsilon)} \sum_{j=1}^{N}\sum_{i\in E_j} r_i \mathcal{L}_{\mathfrak{m}}(\boldsymbol{c_j}, y_i)\right] &= \frac{1}{n}\,\mathbb{E}_{r_i}\left[\sup_{\mathfrak{m}\in\mathfrak{M}_{TV}(\alpha_p,\epsilon)} \sum_{j=1}^{N}\sum_{y=1}^{K} \mathcal{L}_{\mathfrak{m}}(\boldsymbol{c_j}, y) \sum_{i\in E_{y,j}} r_i\right] \\ &\leq \frac{1}{n}\,\mathbb{E}_{r_i}\left[\sum_{j=1}^{N}\sum_{y=1}^{K}\left|\sum_{i\in E_{y,j}} r_i\right|\right].\end{aligned}$$

Finally using the Khintchine inequality and the Cauchy Schartz inequality we get

$$\frac{1}{n}\,\mathbb{E}_{r_i}\left[\sum_{j=1}^{N}\sum_{y=1}^{K}\left|\sum_{i\in E_{y,j}} r_i\right|\right] \leq \frac{1}{n}\sum_{j=1}^{N}\sum_{y=1}^{K}\sqrt{|E_{y,j}|} \quad \text{(Khintchine)}$$

$$\leq \frac{1}{n}\sqrt{N \times K}\sqrt{\sum_{j=1}^{N}\sum_{y=1}^{K}|E_{y,j}|} \quad \text{(Cauchy)}$$
$$= \sqrt{\frac{N \times K}{n}}.$$

By combining the upper-bounds we have for each term, we get the expected result,

$$\mathfrak{R}_{\mathcal{S}}\left(\mathcal{L}_{\mathfrak{M}_{TV}(\alpha_p,\epsilon)}\right) \leq \sqrt{\frac{N \times K}{n}} + \epsilon.$$

□

The above result means that, if we can cover the n training samples with $O(1)$ balls, then we can bound the generalization gap of any randomized classifier $\mathfrak{m} \in \mathfrak{M}_{TV}(\alpha_p, \epsilon)$ by $O\left(\frac{1}{\sqrt{n}}\right) + \epsilon$.

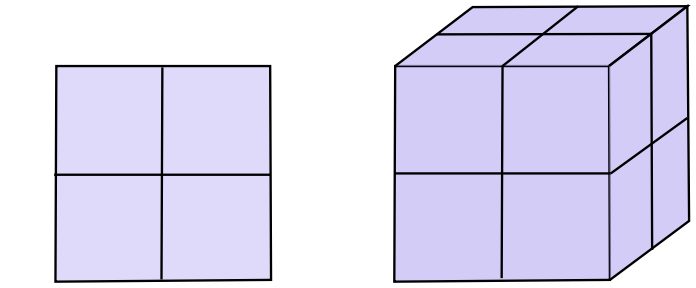
4.3.2 Discussion on the generalization bound

Xu *et al.* [171] previously studied generalization bounds for learning algorithms based on their robustness. Although we use very different techniques of proof, their results and ours are similar. More precisely, both analyses conclude that robust models generalize well if the training samples have a small covering number. Note, however, that we base our formulation on an *adaptive partition* of the samples, while the initial paper only focuses on a fixed partition of the input space. The interested reader can refer to the discussion section in [171] for more details.

These findings seem to contradict the current line of works on the hardness of generalization gap in the adversarial setting – see Section 2.4. In fact, if the ground truth distribution is sufficiently concentrated, a small number of balls can cover $\mathcal{S}$ with high probability; hence $N = O(1)$. This means that we can learn robust classifiers with the same sample complexity as in the standard setting. But if the ground truth distribution is not concentrated enough, the training samples will be far one from another; hence forcing the covering number to be large. In the worse case scenario, we need to cover the whole space $[0,1]^d$ giving a covering number $N = O\left(\frac{1}{(\alpha_p)^d}\right)$ which is exponential in the dimension of the problem. Figure 4.1 illustrates the exponential growth of the covering number of $[0,1]^d$ for the ℓ_∞ norm. When $d = 2$ – on the left – we need $2^2 = 4$ points to cover $[0,1]^d$ with ℓ_∞ balls of radius $\frac{1}{2}$. When $d = 3$ – on the right – we need $2^3 = 8$ points. In general, covering $[0,1]^d$ with ℓ_∞ balls of radius α_∞, requires at least $\frac{1}{(\alpha_\infty)^d}$ elements. Finally, when we change from the ℓ_∞ to any ℓ_p norm, we get a covering number $N\left([0,1]^d, \|.\|_p, \alpha_p\right) = O\left(\frac{1}{(\alpha_p)^d}\right)$.

Therefore, in the worst-case scenario, our bound is in $O\left(\frac{1}{(\alpha_p)^d\sqrt{n}}\right) + \epsilon$. When α_p is small and the dimension of the problem is high, this bound is too large to give any meaningful insight on the generalization gap of the problem. Therefore, we still need to tighten our analysis to show that robust learning for randomized classifiers is possible in high dimensional spaces.

Remark 20. *Note that, we provided a very general result for randomized classifiers under the only assumption that they are robust* w.r.t. *the total variation distance. To build a finer analysis,*



$N([0,1]^2, ||.||_\infty, 1/2) = \left(\frac{1}{1/2}\right)^2 = 4$ $N([0,1]^3, ||.||_\infty, 1/2) = \left(\frac{1}{1/2}\right)^3 = 8$

Figure 4.1: Illustration of a $1/2$-covering for the hyper cube for the ℓ_∞ norm. On the left: $[0,1]^2$. On the right: $[0,1]^3$.

and to evade the dimension dependencies, we should consider designing specific sub-classes $\mathfrak{M} \subset \mathfrak{M}_{TV}(\alpha_p, \epsilon)$ and adapt the techniques of proof to make the term N smaller in the worst-case scenario.

4.4 Mode preservation and randomized smoothing

4.4.1 Mode preservation property for randomized classifiers

Notations. *Let $\rho \in \mathcal{P}(\mathcal{Y})$ be the vector of point-wise probability $\rho = (\rho_1, \dots, \rho_K)$, we denote $(\rho_{(1)}, \dots, \rho_{(K)})$ the probability vector ρ sorted in decreasing order.*

Even though randomized classifiers have some interesting properties regarding generalization gap, we can also study them through the prism of deterministic robustness. Let us for example consider the classifier that outputs the class with the highest probability for $\mathfrak{m}(\boldsymbol{x})$ – *a.k.a.* the mode of $\mathfrak{m}(\boldsymbol{x})$. It writes

$$c : \boldsymbol{x} \mapsto \underset{k \in [K]}{\operatorname{argmax}}\, \mathfrak{m}(\boldsymbol{x})_k \tag{4.11}$$

Then checking whether c is robust boils down to demonstrating that the mode of $\mathfrak{m}(\boldsymbol{x})$ does not change under perturbation. It turns out that D_{TV} robust classifiers have this property. We call it the mode preservation property of $\mathfrak{M}_{TV}(\alpha_p, \epsilon)$.

Proposition 1 (Mode preservation for D_{TV}-robust classifiers)**.** *Let $\mathfrak{m} \in \mathfrak{M}_{TV}(\alpha_p, \epsilon)$ be a robust randomized classifier and $\boldsymbol{x} \in \mathcal{X}$ such that $\mathfrak{m}(\boldsymbol{x})_{(1)} \geq \mathfrak{m}(\boldsymbol{x})_{(2)} + 2\epsilon$. Then, for any $\boldsymbol{\tau} \in \mathcal{X}$, the following holds,*

$$\|\boldsymbol{\tau}\|_p \leq \alpha_p \implies c(\boldsymbol{x}) = c(\boldsymbol{x} + \boldsymbol{\tau}) \ .$$

Proof. Let $\boldsymbol{x}, \boldsymbol{\tau} \in \mathcal{X}$ such that $\|\boldsymbol{\tau}\|_p \leq \alpha_p$ and $\mathfrak{m} \in \mathfrak{M}_{TV}(\alpha_p, \epsilon)$ such that

$$\mathfrak{m}(\boldsymbol{x})_{(1)} \geq \mathfrak{m}(\boldsymbol{x})_{(2)} + 2\epsilon.$$

By definition of $\mathfrak{M}_{TV}(\alpha_p, \epsilon)$, we have that

$$D_{TV}(\mathfrak{m}(\boldsymbol{x}), \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})) \leq \epsilon.$$

Then, for all $k \in \{1, \dots, K\}$ we have

$$\mathfrak{m}(\boldsymbol{x})_k - \epsilon \leq \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_k \leq \mathfrak{m}(\boldsymbol{x})_k + \epsilon\,.$$

Let us denote k^* the index of the biggest value in $\mathfrak{m}(\boldsymbol{x})$ – *i.e.* $\mathfrak{m}(\boldsymbol{x})_{k^*} = \mathfrak{m}(\boldsymbol{x})_{(1)}$. For any $k \in \{1, \dots, K\}$ with $k \neq k^*$, we have $\mathfrak{m}(\boldsymbol{x})_{k^*} \geq \mathfrak{m}(\boldsymbol{x})_k + 2\epsilon$. Finally, for any $k \neq k^*$, we get

$$\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_{k^*} \geq \mathfrak{m}(\boldsymbol{x})_{k^*} - \epsilon \geq \mathfrak{m}(\boldsymbol{x})_k + \epsilon \geq \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_k.$$

Then, $\underset{k\in[K]}{\operatorname{argmax}}\, \mathfrak{m}(\boldsymbol{x})_k = \underset{k\in[K]}{\operatorname{argmax}}\, \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_k$. This concludes the proof. □

Coming back to the decomposition in Equation (4.7), with the above result, we can bound the risk the adversary induces with non-zero perturbations by the mass of points on which the classifier c gives the good response but based on a low probability, *i.e.* with small confidence.

$$\mathcal{R}^{\text{adv}}_{>0}(\mathfrak{m}) \leq \mathbb{P}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big[c(\boldsymbol{x}) = y \textit{ and } \mathfrak{m}(\boldsymbol{x})_{(1)} < \mathfrak{m}(\boldsymbol{x})_{(2)} + 2\epsilon\big]. \tag{4.12}$$

This means that the only points on which the adversary may induce misclassification are the points on which $\mathfrak{m}$ already has a high risk. Once more, this says something fundamental about the behavior of robust randomized classifiers. On undefended models, the adversary could change the decision on any point it wanted; now it is limited to changing points on which the classifier is already bad. This considerably mitigates the threat model we should consider.

Furthermore, for any deterministic classifier designed as in Equation (4.11), we can also bound the maximal loss of accuracy under attack the classifier may suffer. This bound may, however, be harder to evaluate since it now depends on both the classifier and the dataset distribution.

4.4.2 From mode preservation to randomized smoothing

The classifier we define in Equation (4.11) and the mode preservation property of $\mathfrak{m}$ are closely related to provable defenses based on randomized smoothing. Recall that the core idea of randomized smoothing is to take a hypothesis $\boldsymbol{h}$ with probit outputs, and to build a robust classifier that writes

$$c_{rob} : \boldsymbol{x} \mapsto \underset{k\in[K]}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(0,\sigma^2 I)}[\boldsymbol{h}_k(\boldsymbol{x}+\boldsymbol{z})]\,. \tag{4.13}$$

From a probabilistic point of view, for any input $\boldsymbol{x}$, randomized smoothing amounts to output the most probable class of the probability measure

$$\mathfrak{m}(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(0,\sigma^2 I)}\left[[\boldsymbol{h}_1(\boldsymbol{x}+\boldsymbol{z}), \dots, \boldsymbol{h}_K(\boldsymbol{x}+\boldsymbol{z})]^\intercal\right]. \tag{4.14}$$

Hence, randomized smoothing uses the mode preservation property of $\mathfrak{m}$ to build a provably robust – deterministic – classifier. Therefore, the above results – Proposition 1 and Equation 4.14– also hold for provable defenses based on randomized smoothing.

Remark 21. *Note that we can only use the mode preservation property of* $\mathfrak{m}$ *if it is* (α_p, ϵ)*-robust* w.r.t. D_{TV}. *In the next chapter we will demonstrate that from any deterministic hypothesis* $\boldsymbol{h}$*, we can build a robust randomized classifier using noise injection from a Gaussian distribution; hence* $\mathfrak{m}$ *defined in Equation* (4.14) *is robust* w.r.t. D_{TV}.

Studying randomized smoothing from our point of view could give an interesting new perspective on that method. So far no results have been published on the generalisation gap of this defense in the adversarial setting. We could devise generalization bounds by similarity with our analysis. Furthermore, the probabilistic interpretation stresses that randomized smoothing is somewhat restrictive since it only considers probability measures which are the expectation on a simple noise injection scheme. The mode preservation property explains the behavior of randomized smoothing, but also presents fundamental properties of randomized defenses that could be used to construct more general defense schemes.

4.5 Additional results: extension to the Renyi divergence and discussion on probability metrics

4.5.1 Extending previous results to the Renyi divergence

In this section, we extend the previous results to another probability divergence of reference, namely the Renyi divergence. Let $\mathcal{Z}$ be an arbitrary space, and ρ, ρ' be two measures in $\mathcal{P}(\mathcal{Z})$, with probability density functions of g and g' according to a third measure ν. The *Renyi divergence of order* β writes

$$D_\beta(\rho, \rho') := \frac{1}{\beta - 1} \log \int_{\mathcal{Y}} g'(y) \left(\frac{g(y)}{g'(y)} \right)^\beta d\nu(y) \ .$$

The Renyi divergence [136] is a generalized divergence defined for any β on the interval $[1, \infty]$. It equals the Kullback-Leibler divergence when $\beta \to 1$, and the maximum divergence when $\beta \to \infty$. It also has the property of being non-decreasing with respect to β. This divergence is very common in machine learning and Information theory [161], especially in its Kullback-Leibler form as it is widely used as the loss function – cross entropy – of classification algorithms. In the remaining, we denote $\mathfrak{M}_\beta(\alpha_p, \epsilon)$ the set of (α_p, ϵ)-robust classifiers *w.r.t.* D_β.

Let us show that, for any randomized classifier in this class, we can bound the gap between the risk and the adversarial risk of $\mathfrak{m}$. In the context of the Renyi divergence, the factor that controls the classifier's loss of accuracy under attack is multiplicative and depends both on the robustness parameter ϵ and on the divergence parameter β.

Theorem 12 (Multiplicative risks' gap for Renyi-robust classifiers)**.** *Let* $\mathfrak{m} \in \mathfrak{M}_\beta(\alpha_p, \epsilon)$. *Then we have*

$$\mathcal{R}^{\text{adv}}(\mathfrak{m}; \alpha_p) \leq \left(e^\epsilon \, \mathcal{R}(\mathfrak{m}) \right)^{\frac{\beta - 1}{\beta}} .$$

The proof of this theorem mainly relies on a famous property of the Renyi divergence called probability preservation property.

Proposition 2 ([96]). *Let ρ and ρ' be two measures in $\mathcal{P}(\mathcal{Z})$. Then for any $Z \in \mathcal{A}(\mathcal{Z})$, the following holds,*

$$\rho(Z) \leq \left(\exp\left(D_\beta(\rho, \rho')\right)\rho'(Z)\right)^{\frac{\beta-1}{\beta}}.$$

With this proposition at hand, we can now demonstrate how the adversarial risk of a randomized classifier relates to its standard risk, under robustness assumptions.

Proof. Let $\mathfrak{m}$ be an (α_p, ϵ)-robust classifier *w.r.t.* D_β, $(\boldsymbol{x}, y) \sim \mathcal{D}$ and $\boldsymbol{\tau} \in \mathcal{X}$ such that $\|\boldsymbol{\tau}\|_p \leq \alpha_p$. With the same reasoning as above, and with Proposition 2, we get

$$\begin{aligned}
\mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}), y) &= \mathbb{E}_{\hat{y}\sim\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})}[\mathbb{1}\{\hat{y} \neq y\}] \\
&= \mathbb{P}_{\hat{y}\sim\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})}[\hat{y} \neq y] \\
&\leq \left(e^{D_\beta(\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}),\mathfrak{m}(\boldsymbol{x}))}\,\mathbb{P}_{\hat{y}\sim\mathfrak{m}(\boldsymbol{x})}[\hat{y} \neq y]\right)^{\frac{\beta-1}{\beta}} \quad \text{(Prop. 2)} \\
&= \left(e^{D_\beta(\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}),\mathfrak{m}(\boldsymbol{x}))}\,\mathbb{E}_{\hat{y}\sim\mathfrak{m}(\boldsymbol{x})}[\mathbb{1}\{\hat{y} \neq y\}]\right)^{\frac{\beta-1}{\beta}} \\
&\leq \left(e^{\epsilon}\mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}), y)\right)^{\frac{\beta-1}{\beta}}.
\end{aligned}$$

Since this holds for any $(\boldsymbol{x}, y) \sim \mathcal{D}$ and any α_p bounded perturbation $\boldsymbol{\tau}$, we get

$$\begin{aligned}
\mathcal{R}^{\text{adv}}(\mathfrak{m}; \alpha_p) &= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}), y)\right] \\
&\leq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[e^{\frac{\beta-1}{\beta}\epsilon}\mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}), y)^{\frac{\beta-1}{\beta}}\right] \\
&\leq e^{\frac{\beta-1}{\beta}\epsilon}\,\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}), y)^{\frac{\beta-1}{\beta}}\right].
\end{aligned}$$

Finally, using the Jensen inequality, one gets

$$\leq e^{\frac{\beta-1}{\beta}\epsilon}\,\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}), y)\right]^{\frac{\beta-1}{\beta}} = \left(e^{\epsilon}\,\mathcal{R}(\mathfrak{m})\right)^{\frac{\beta-1}{\beta}}.$$

The above inequality concludes the proof. □

This first result gives a multiplicative bound on the gap between the standard and adversarial risks. This means that if we can design a class $\mathfrak{M}_\beta(\alpha_p, \epsilon)$ with small enough ϵ, and big enough β, then minimizing the risk of any $\mathfrak{m} \in \mathfrak{M}_\beta(\alpha_p, \epsilon)$ is sufficient to also minimize the adversarial risk of $\mathfrak{m}$. Nevertheless, multiplicative factors are not easy to analyze. Theorem 13 provides an additive counterpart to Theorem 12. It gives a control on the loss of accuracy under attack with respect to the robustness parameter ϵ and the Shannon entropy of $\mathfrak{m}$.

Theorem 13 (Additive risks' gap for Renyi-robust classifiers). *Let $\mathfrak{m} \in \mathfrak{M}_\beta(\alpha_p, \epsilon)$, then we have*

$$\mathcal{R}^{\text{adv}}(\mathfrak{m}; \alpha_p) - \mathcal{R}(\mathfrak{m}) \leq 1 - e^{-\epsilon}\,\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_{|\mathcal{X}}}\left[e^{-H(\mathfrak{m}(\boldsymbol{x}))}\right]$$

where H is the Shannon entropy – i.e. *for any $\rho \in \mathcal{P}(\mathcal{Y})$, $H(\rho) = -\sum_{k\in\mathcal{Y}} \rho_k \log(\rho_k)$ – and $\mathcal{D}_{|\mathcal{X}}$ is the marginal distribution of $\mathcal{D}$ for $\mathcal{X}$.*

Proof. Let $\mathfrak{m} \in \mathfrak{M}_\beta(\alpha_p, \epsilon)$, then

$$\begin{aligned}
&\mathcal{R}^{\text{adv}}(\mathfrak{m}; \alpha_p) - \mathcal{R}(\mathfrak{m}) \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}), y) - \mathcal{L}_{0/1}(\mathfrak{m}(\boldsymbol{x}), y)\right].
\end{aligned}$$

By definition of the $0/1$ loss, this amounts to write

$$\begin{aligned}
&= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathbb{E}_{\hat{y}_{\text{adv}}\sim\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}),\hat{y}\sim\mathfrak{m}(\boldsymbol{x})}[\mathbb{1}(\hat{y}_{\text{adv}} \neq y) - \mathbb{1}(\hat{y} \neq y)]\right] \\
&\leq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathbb{E}_{\hat{y}_{\text{adv}}\sim\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}),\hat{y}\sim\mathfrak{m}(\boldsymbol{x})}[\mathbb{1}(\hat{y}_{\text{adv}} \neq \hat{y})]\right] \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathbb{P}_{\hat{y}_{\text{adv}}\sim\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}),\hat{y}\sim\mathfrak{m}(\boldsymbol{x})}[\hat{y}_{\text{adv}} \neq \hat{y}]\right] \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} 1 - \mathbb{P}_{\hat{y}_{\text{adv}}\sim\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}),\hat{y}\sim\mathfrak{m}(\boldsymbol{x})}[\hat{y}_{\text{adv}} = \hat{y}]\right] \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} 1 - \sum_{i=1}^{K} \mathfrak{m}(\boldsymbol{x})_i \times \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_i\right].
\end{aligned}$$

Now, note that for any $(\boldsymbol{x}, y) \sim \mathcal{D}$ and $\boldsymbol{\tau} \in \mathcal{X}$, by definition of a probability vector in $\mathcal{P}(\mathcal{Y})$, and thanks to Jensen inequality we can write

$$\sum_{i=1}^{K} \mathfrak{m}(\boldsymbol{x})_i \times \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_i \geq \exp\left(\sum_{i=1}^{K} \mathfrak{m}(\boldsymbol{x})_i \log \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_i\right).$$

Then by definition of the entropy and the Kullback Leibler divergence we have

$$\exp\left(\sum_{i=1}^{K} \mathfrak{m}(\boldsymbol{x})_i \log \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_i\right) = \exp\big(-D_1(\mathfrak{m}(\boldsymbol{x}), \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})) - H(\mathfrak{m}(\boldsymbol{x}))\big).$$

Finally, by combining the above inequalities and since $\mathfrak{m} \in \mathfrak{M}_\beta(\alpha_p, \epsilon)$ we get

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathbb{P}_{\hat{y}_{\text{adv}}\sim\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}),\hat{y}\sim\mathfrak{m}(\boldsymbol{x})}(\hat{y}_{\text{adv}} \neq \hat{y})\right]$$

$$\leq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} 1 - e^{-D_1(\mathfrak{m}(\boldsymbol{x}),\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}))-H(\mathfrak{m}(\boldsymbol{x}))}\right]$$
$$\leq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[1 - e^{-\epsilon-H(\mathfrak{m}(\boldsymbol{x}))}\right] = 1 - e^{-\epsilon}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_{|\mathcal{X}}}\left[e^{-H(\mathfrak{m}(\boldsymbol{x}))}\right].$$

The above inequality concludes the proof. □

This result is interesting because it relates the accuracy of $\mathfrak{m}$ with the bound we obtain. Intuitively, when $\mathfrak{m}(\boldsymbol{x})$ has large entropy – *i.e.* $H(\mathfrak{m}(\boldsymbol{x})) \to \log(K)$ – the output distribution tends towards the uniform distribution; hence $\epsilon \to 0$. This means that the classifier is very robust but also completely inaccurate, since it outputs classes uniformly at random. On the opposite, if $H(\mathfrak{m}(\boldsymbol{x})) \to 0$, then $\epsilon \to \infty$. The classifier may be accurate, but it is not robust anymore – at least according to our definition. Hence we need to find a classifier that has reasonable robustness and good accuracy simultaneously. To evaluate our ability to do so, as a corollary of Theorem 11, we can bound the Rademacher complexity of the class $\mathcal{L}_{\mathfrak{M}_\beta(\alpha_p,\epsilon)}$.

Corollary 1. *Let $\mathcal{L}_{\mathfrak{M}_\beta(\alpha_p,\epsilon)}$ be the loss function class associated with $\mathfrak{M}_\beta(\alpha_p,\epsilon)$. Then, for any $\mathcal{S} := \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_n}, y_n)\}$, the following holds,*

$$\mathfrak{R}_{\mathcal{S}}\left(\mathcal{L}_{\mathfrak{M}_\beta(\alpha_p,\epsilon)}\right) \leq \sqrt{\frac{N\times K}{n}} + \min\left(\frac{3}{2}\left(\sqrt{1+\frac{4\epsilon}{9}}-1\right)^{1/2}, \frac{e^{\epsilon+1}-1}{e^{\epsilon+1}+1}\right).$$

Where $N = N\left(\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}, \|.\|_p, \alpha_p\right)$ is the α_p-external covering number of the inputs $\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}$ for the ℓ_p norm.

To prove Corollary 1, note that thanks to previous works [61, 159], we can always upper-bound the total variation distance by a function of the Renyi divergence.

Proposition 3 (Inequality between total variation and Renyi divergence)**.** *Let ρ and ρ' be two measures in $\mathcal{P}(\mathcal{Z})$, and $\beta \geq 1$. Then the following holds,*

$$D_{TV}(\rho,\rho') \leq \min\left(\frac{3}{2}\left(\sqrt{1+\frac{4D_\beta(\rho,\rho')}{9}}-1\right)^{1/2}, \frac{\exp(D_\beta(\rho,\rho')+1)-1}{\exp(D_\beta(\rho,\rho')+1)+1}\right).$$

Proof. Thanks to [61], one has

$$D_1(\rho,\rho') \geq 2D_{TV}(\rho,\rho')^2 + \frac{4D_{TV}(\rho,\rho')^4}{9}.$$

From which it follows that

$$D_{TV}(\rho,\rho') \leq \frac{3}{2}\left(\sqrt{1+\frac{4D_1(\rho,\rho')}{9}}-1\right)^{1/2}.$$

Moreover, using inequality from [159], one gets

$$D_1(\rho,\rho')+1 \geq \log\left(\frac{1+D_{TV}(\rho,\rho')}{1-D_{TV}(\rho,\rho')}\right).$$

This inequality leads to the following

$$\frac{\exp(D_1(\rho,\rho')+1)-1}{\exp(D_1(\rho,\rho')+1)+1} \geq D_{TV}(\rho,\rho').$$

By combining the above inequalities and by monotony of Renyi divergence regarding β, one obtains the expected result. □

Then to get the proof of Corollary 1, we only need to combine Theorem 11 and Proposition 3. Finally, let us present the mode preservation property for $\mathfrak{m} \in \mathfrak{M}_\beta(\alpha_p,\epsilon)$.

Proposition 4 (Mode preservation for Renyi-robust classifiers)**.** *Let $\mathfrak{m} \in \mathfrak{M}_\beta(\alpha_p,\epsilon)$ be a robust randomized classifier and $\boldsymbol{x} \in \mathcal{X}$ such that $\left(\mathfrak{m}(\boldsymbol{x})_{(1)}\right)^{\frac{\beta}{\beta-1}} \geq \exp\left((2-\frac{1}{\beta})\epsilon\right)\left(\mathfrak{m}(\boldsymbol{x})_{(2)}\right)^{\frac{\beta-1}{\beta}}$. Then, for any $\boldsymbol{\tau} \in \mathcal{X}$, the following holds,*

$$\|\boldsymbol{\tau}\|_p \leq \alpha_p \implies c(\boldsymbol{x}) = c(\boldsymbol{x}+\boldsymbol{\tau}),$$

where $c(\boldsymbol{x}) := \underset{k\in[K]}{\operatorname{argmax}}\, \mathfrak{m}(\boldsymbol{x})_k$.

Proof. Let $\boldsymbol{x},\boldsymbol{\tau} \in \mathcal{X}$ such that $\|\boldsymbol{\tau}\|_p \leq \alpha_p$ and $\mathfrak{m} \in \mathfrak{M}_\beta(\alpha_p,\epsilon)$ such that

$$\left(\mathfrak{m}(\boldsymbol{x})_{(1)}\right)^{\frac{\beta}{\beta-1}} \geq \exp\left((2-\frac{1}{\beta})\epsilon\right)\left(\mathfrak{m}(\boldsymbol{x})_{(2)}\right)^{\frac{\beta-1}{\beta}}.$$

Then by definition of $\mathfrak{M}_\beta(\alpha_p,\epsilon)$, we have

$$D_\beta(\mathfrak{m}(\boldsymbol{x}),\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})) \leq \epsilon.$$

Furthermore, by using Proposition 2, for any $k \in \{1,\dots,K\}$ we have

$$(*)\ \mathfrak{m}(\boldsymbol{x})_k \leq \left(\exp(\epsilon)\,\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_k\right)^{\frac{\beta-1}{\beta}} \text{ and } (**)\ \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_k \leq \left(\exp(\epsilon)\,\mathfrak{m}(\boldsymbol{x})_k\right)^{\frac{\beta-1}{\beta}}.$$

Let us denote k^* the index such that $\mathfrak{m}(\boldsymbol{x})_{k^*} = \mathfrak{m}(\boldsymbol{x})_{(1)}$. Then using $(*)$ we get

$$\mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_{k^*} \geq \exp(-\epsilon)(\mathfrak{m}(\boldsymbol{x})_{k^*})^{\frac{\beta}{\beta-1}}.$$

Furthermore for any $k \in \{1,\dots,K\}$ where $k \neq k^*$, we can use the assumption we made on $\mathfrak{m}$ to get

$$\exp(-\epsilon)(\mathfrak{m}(\boldsymbol{x})_{k^*})^{\frac{\beta}{\beta-1}} \geq \exp\left(\frac{\beta-1}{\beta}\epsilon\right)(\mathfrak{m}(\boldsymbol{x})_k)^{\frac{\beta-1}{\beta}}.$$

Finally, using $(**)$ we have

$$\exp\left(\frac{\beta-1}{\beta}\epsilon\right)(\mathfrak{m}(\boldsymbol{x})_k)^{\frac{\beta-1}{\beta}} \geq \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_k.$$

The above gives us $\underset{k\in[K]}{\operatorname{argmax}}\, \mathfrak{m}(\boldsymbol{x})_k = \underset{k\in[K]}{\operatorname{argmax}}\, \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})_k$. This concludes the proof. □

4.5.2 Discussion on the metric/divergence one should consider

As mentioned earlier in this chapter, the choice of the metric/divergence is crucial as it characterizes the notion of adversarial robustness we are examining. We focus on the total variation distance and Renyi divergence, but the question of whether these metrics/divergences are more appropriate than others remains open. It should be noted, however, that our definition of robustness is monotonous depending on the metric/divergence we use.

Proposition 5 (Monotonicity of the robustness)**.** *Let $\mathfrak{m}$ be a randomized classifier, and let D and D' be two divergences/metrics on $\mathcal{P}(\mathcal{Y})$. If there exists a non decreasing function $f : \mathbb{R} \to \mathbb{R}$ such that $\forall \rho, \rho' \in \mathcal{P}(\mathcal{Y})$, $D(\rho, \rho') \leq f(D'(\rho, \rho'))$, then the following assertion holds.*

$$\mathfrak{m} \textit{ is } (\alpha_p, \epsilon)\text{-robust } \textit{w.r.t. } D' \implies \mathfrak{m} \textit{ is } (\alpha_p, f(\epsilon))\textit{-robust } \text{w.r.t. } D.$$

The proof straightforwardly comes from the definition of robustness.

Proof. Let us consider $\mathfrak{m}$ a randomized classifier (α_p, ϵ)-robust *w.r.t.* D'. Then for any $\boldsymbol{x} \sim \mathcal{D}$, and $\boldsymbol{\tau}$ *s.t.* $\|\boldsymbol{\tau}\|_p \leq \alpha_p$, since f is non decreasing, we have

$$D(\mathfrak{m}(\boldsymbol{x}), \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})) \leq f\big(D'(\mathfrak{m}(\boldsymbol{x}), \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau}))\big) \leq f(\epsilon).$$

Then $\mathfrak{m}$ is $(\alpha_p, f(\epsilon))$-robust *w.r.t.* D which concludes the proof. □

The above result suggests that the different notions of robustness we might conceive are more related than they appear. Here are some of the most classical divergences used in machine learning. Let ρ, ρ', ν three measures in $\mathcal{P}(\mathcal{Y})$. We denotes g and g' the probability density functions of ρ and ρ' with respect to ν. Then we can define the *Wasserstein distance* as follows

$$D_W(\rho, \rho') := \inf \int_{\mathcal{Y}^2} \operatorname{dist}\big(y, y'\big) d\pi(y, y'), \tag{4.15}$$

where dist is some ground distance on $\mathcal{Y}$, and the infimum is taken over all joint distributions π in $\mathcal{P}(\mathcal{Y} \times \mathcal{Y})$ with marginals ρ and ρ'.

Remark 22. *In transportation theory, the Wasserstein distance is solution of the Monge-Kantorovich problem with the cost function $c(y, y') = \operatorname{dist}(y, y')$. Then, the definitions of total variation and Wasserstein distance match when we use the trivial distance* $\operatorname{dist}(y, y') = \mathbb{1}\{y \neq y'\}$.

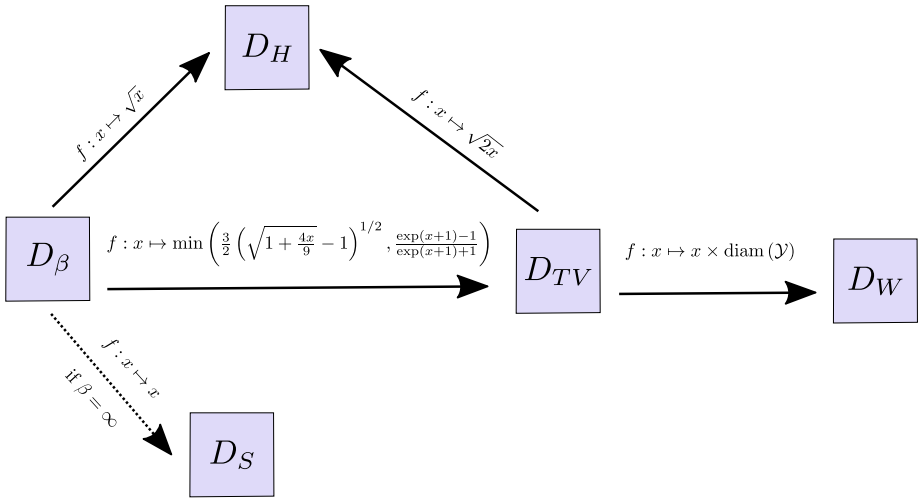


Figure 4.2: Summary of the relations between the different robustness notions from Propositions 6 and 7.

We also define respectively the *Hellinger distance* and the *Separation distance* as follows.

$$D_H(\rho, \rho') := \left[\int_{\mathcal{Y}} \left(\sqrt{g} - \sqrt{g'}\right)^2 d\nu\right]^{1/2}. \tag{4.16}$$

$$D_S(\rho, \rho') := \sup_{y \in \mathcal{Y}} \left(1 - \frac{\rho(y)}{\rho'(y)}\right). \tag{4.17}$$

If we take any of the above metrics/divergences to instantiate a notion of adversarial robustness we might get very different semantics for them. However, we can show that any of these definitions can be covered – with respect to Proposition 5 – either by the Renyi or the total variation robustness. Figure 4.2 summarizes the links we can make between all these different definitions of robustness, and Propositions 6 and 7 present the associated results. We can see that the total variation distance and the Renyi divergence are both central since they can cover any of the other robustness notions. This does not mean that they are more appropriate than the others, but at least they are general enough to cover a wide range of possible definitions.

Proposition 6. *Let* $\mathfrak{m}$ *be a randomized classifier. If* $\mathfrak{m}$ *is* (α_p, ϵ)*-robust* w.r.t. D_{TV} *then the following assertions hold.*

- $\mathfrak{m}$ *is* $(\alpha_p, \epsilon \times \mathrm{diam}(\mathcal{Y}))$*-robust* w.r.t. D_W*, where* $\mathrm{diam}(\mathcal{Y}) := \max_{y,y' \in \mathcal{Y}} \mathrm{dist}(y, y')$.
- $\mathfrak{m}$ *is* $\left(\alpha_p, \sqrt{2\epsilon}\right)$*-robust* w.r.t. D_H.

Proof. Let us consider ρ and $\rho' \in \mathcal{P}(\mathcal{Y})$. Thanks to [59] we have

- $D_W(\rho, \rho') \leq \mathrm{diam}(\mathcal{Y}) D_{TV}(\rho, \rho')$.

- $D_H(\rho, \rho') \leq \sqrt{2D_{TV}(\rho, \rho')}$.

Hence, by using Proposition 5 respectively with $f : x \mapsto \operatorname{diam}(\mathcal{Y})x$ and $f : x \mapsto \sqrt{2x}$ we get the expected results. □

Proposition 7. *Let $\mathfrak{m}$ be a randomized classifier. If $\mathfrak{m}$ is (α_p, ϵ)-robust* w.r.t. *D_β then the following assertions hold.*

- *$\mathfrak{m}$ is (α_p, ϵ')-robust* w.r.t. *D_{TV} with* $\epsilon' = \min\left(\frac{3}{2}\left(\sqrt{1+\frac{4\epsilon}{9}}-1\right)^{1/2}, \frac{\exp(\epsilon+1)-1}{\exp(\epsilon+1)+1}\right)$.
- *$\mathfrak{m}$ is $(\alpha_p, \sqrt{\epsilon})$-robust* w.r.t. *$D_H$*.
- *If $\beta = \infty$, then $\mathfrak{m}$ is (α_p, ϵ) robust* w.r.t. *D_S*.

Proof. 1) First, let us suppose that $\beta \geq 1$. Thanks to Proposition 3 and to [59], for any $\rho, \rho' \in \mathcal{P}(\mathcal{Y})$ we have

- $D_H(\rho, \rho') \leq \sqrt{D_1(\rho, \rho')} \leq \sqrt{D_\beta(\rho, \rho')}$ (see [59]).
- $D_{TV}(\rho, \rho') \leq \min\left(\frac{3}{2}\left(\sqrt{1+\frac{4D_\beta(\rho,\rho')}{9}}-1\right)^{1/2}, \frac{\exp\left(D_\beta(\rho,\rho')+1\right)-1}{\exp\left(D_\beta(\rho,\rho')+1\right)+1}\right)$ (Prop. 3).

Hence, by using Proposition 5, as above, we get the expected results.
2) Now let us suppose that $\beta = \infty$. By definition of the supremum divergence, we have

$$D_\infty(\rho, \rho') = \sup_{B \subset \mathcal{Y}} \left|\ln \frac{\rho(B)}{\rho'(B)}\right|.$$

Furthermore, note that the function $x \mapsto 1 - x - |\ln(x)|$ is negative on $\mathbb{R}$, therefore for any $y \in \mathcal{Y}$ one has

$$1 - \frac{\rho(y)}{\rho'(y)} \leq \left|\ln \frac{\rho(y)}{\rho'(y)}\right|.$$

Since the above inequality is true for any $y \in \mathcal{Y}$, we have

$$D_S\left(\rho, \rho'\right) = \sup_{y \in \mathcal{Y}}\left(1 - \frac{\rho(y)}{\rho'(y)}\right) \leq \sup_{y \in \mathcal{Y}}\left|\ln \frac{\rho(y)}{\rho'(y)}\right| \leq \sup_{B \subset \mathcal{Y}}\left|\ln \frac{\rho(B)}{\rho'(B)}\right| = D_\infty(\rho, \rho').$$

Finally, by using Proposition 5 with $f : x \mapsto x$ we get the expected results. □

4.6 Lessons learned and future works

In this chapter, we examined the theoretical properties of randomized classifiers, both in terms of robustness and accuracy. We first defined a notion of robustness for randomized classifiers using probability metrics/divergences – the total variation distance and the Renyi divergence. We then

demonstrated that when a randomized classifier complies with this definition of robustness, we can estimate the maximum loss of precision it can suffer under attack. This answers questions **Q1** and **Q2** as follows.

> *There exists classes of randomized classifiers for which we can control the gap between the adversarial and the standard risks.*

We then studied the generalization properties of this class of functions and gave results indicating that robust randomized classifiers can generalize. Finally, we showed that randomized classifiers have a mode preservation property. This presents a fundamental property of randomized defenses that can be used to explain randomized smoothing from a probabilistic point of view. Based on the above analysis, we approach randomized classifiers from a more practical point of view in Chapter 5 and demonstrate that we can build such classifiers from state-of-the-art neural network architectures. Our analysis could be refined in several ways. We list some of them here for future considerations.

Future work 1: tighter bounds for the generalization gap

Our results on the standard generalization of randomized classifiers could be improved, especially since they can – in some cases – be very dependent on the dimension of the problem. In future works, we aim to study this property from a different perspective. We could for example study the covering number of the class of functions we consider instead of the covering number of the training samples. To this end, we could use technical tools such as the Massart's lemma or the notion of the shattering dimension to make the bound less dependent on the dimension of the problem.

Future work 2: studying the properties of randomized smoothing

In this chapter, we established some links between the mode preservation property of the randomized classifier, and the provable defense called randomized smoothing. Based on this evidence, we can bound the gap between the standard and adversarial risks for this defense. Another interesting direction would be to show that the classifiers based on randomized smoothing have a generalization gap similar to the classes of randomized classifiers we studied.

Future work 3: study f-divergences and integral probability metrics

For now, we presented results for randomized classifiers that are robust either with respect to the total variation distance or to the Renyi divergence. Both divergences have interesting properties, but we believe that they are a special case of more general classes of divergences for which similar results could be obtained. The study of more general forms of divergences such as f-divergences and integral probability metrics could provide some insights on the generality of the definition of robustness we present in this chapter.

5 A unified view on privacy and robustness to adversarial examples

Contents

Q2: *Can we find a class $\mathcal{H}$ and a hypothesis $\boldsymbol{h}^* \in \mathcal{H}$ that simultaneously has small standard and adversarial risks?*

In chapter 4, we presented two classes of randomized hypotheses that have good properties both in terms of robustness and accuracy; thus answering **Q1** and **Q2** from a theoretical point of view. Here we continue to answer **Q2** but we give a more practical point of view. More precisely, we present simple schemes to build the above mentioned classes and give numerical results demonstrating their accuracy and robustness. In Section 5.1, we highlight the links between differential privacy and Renyi-robustness. By analyzing their definitions, we show that they are based on the same theoretical foundation; therefore, results obtained so far in one domain can be transferred to the other. In Section 5.2, we use tools from the literature on differential privacy to show that Gaussian noise injection can provide principled robustness against ℓ_2 adversarial attacks. Then,

in Section 5.3, we use Gaussian noise injection with advanced neural network architectures to build robust and accurate models. We support our theoretical claims with a series of experiments on CIFAR-10 and CIFAR-100. We achieve both good standard accuracy and state-of-the-art robustness. Finally, we extend our analysis to consider noise injection from exponential families and summarize our results respectively in sections 5.4 and 5.5.

5.1 From differential privacy to Renyi robustness

Notations. *For any set $\mathcal{Z}$ with σ-algebra $\mathcal{A}(\mathcal{Z})$, if there is no ambiguity on the considered σ-algebra, we denote $\mathcal{P}(\mathcal{Z})$ the set of all probability measures over $(\mathcal{Z}, \mathcal{A}(\mathcal{Z}))$. Moreover, $\mathcal{F}_{\mathcal{Z}\times\mathcal{Z}'}$ denotes the set of all measurable functions from $(\mathcal{Z}, \mathcal{A}(\mathcal{Z}))$ to $(\mathcal{Z}', \mathcal{A}(\mathcal{Z}'))$. For $\rho \in \mathcal{P}(\mathcal{Z})$ and $\psi \in \mathcal{F}_{\mathcal{Z}\times\mathcal{Z}'}$, the* pushforward measure *of ρ by ψ is the measure $\psi\#\rho$ such that $\psi\#\rho(B) = \rho(\psi^{-1}(B))$ for any $B \in \mathcal{A}(\mathcal{Z}')$. Finally, let us take $M \in \mathcal{M}_{d\times d}(\mathbb{R})$, we define the Mahalanobis norm with matrix M as the mapping $\boldsymbol{x} \mapsto \|\boldsymbol{x}\|_M = \sqrt{\boldsymbol{x}^\intercal M \boldsymbol{x}}$.*

5.1.1 Introduction to differential privacy

The aim of this chapter is to link two seemingly unrelated notions, namely differential privacy and robustness to adversarial examples. So far, we have discussed robustness thoroughly, but we have only mentioned the central idea behind differential privacy in Chapter 1. So we begin with a brief introduction to this notion before making some links with our domain.

With the large adoption of machine learning techniques, researchers and practitioners are observing growing concerns on the user's privacy of the tools they develop. Beyond primary concerns to guarantee that the private information are not leaked or accidentally disclosed, a crucial issue of machine learning approaches is to ensure that information cannot be recovered or inferred from the sole release of the model – *i.e.* the learning algorithm should be *privacy preserving*. Several definitions have been introduced to characterize these algorithms in the context of machine learning and data publishing. Among them, differential privacy has become the standard by providing a formal and adaptive definition for privacy preserving data analysis. It has been widely studied in many frameworks and applications – see [50] for a complete overview of the field. The central idea of differentiated privacy is to restore the user's trust in their privacy by ensuring that the learning procedure will yield essentially the same results whether or not a person joins the database. Formally, it writes as follows.

Definition 8 (Differential privacy)**.** *Let $\mathfrak{S}_n$ be the space of all data samples of size n, and $\mathcal{H}$ a class of hypotheses. A Learning algorithm $\boldsymbol{A}$ maps a training sample $\mathcal{S} \in \mathfrak{S}_n$ to a hypothesis $\boldsymbol{h} \in \mathcal{H}$. Then $\boldsymbol{A}$ is said to be ϵ-differentially private if for any $\mathcal{S}, \mathcal{S}' \in \mathfrak{S}_n$ that only differ from one input-output pair, and any $\boldsymbol{h} \in \mathcal{H}$, we have*

$$\mathbb{P}[\boldsymbol{A}(\mathcal{S}) = \boldsymbol{h}] \leq \exp(\epsilon)\, \mathbb{P}\big[\boldsymbol{A}\big(\mathcal{S}'\big) = \boldsymbol{h}\big]\,,$$

where the probability is taken over the – possible – random states of the algorithm.

Note that the definition of differential privacy does not make any assumptions about the adversary, it only states a property of the algorithm. In practice, it is difficult to know what kind

of auxiliary information and computing power the adversary may have access to. Therefore, the definition of differential privacy is based on the worst-case scenario where the adversary has access to all training samples and all details of the architecture except those it is trying to deduce. In this sense, the type of adversary we are considering is very close to the threat model we have dealt with until now - white box adversaries - even if they do not have the same objective.

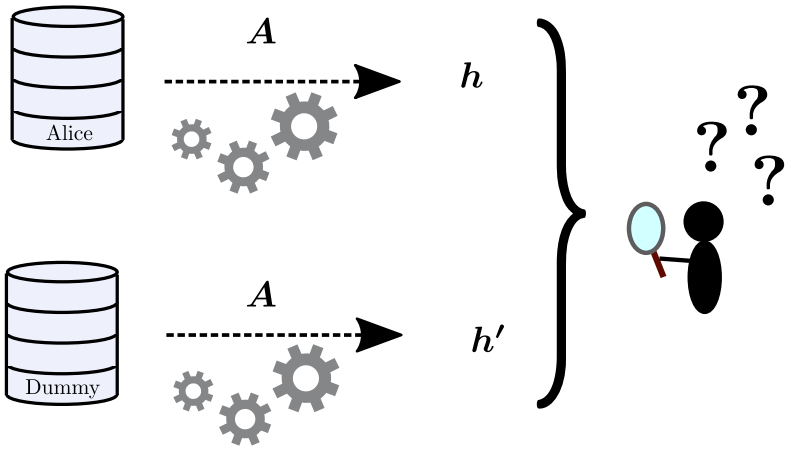


Figure 5.1: Illustration of the typical threat scenario in differential privacy.

To better understand the idea behind differential privacy, let us consider the threat scenario illustrated in Figure 5.1. The model provider has access to a training sample $\mathcal{S}$ in which Alice participated – *e.g.* $(\boldsymbol{x_1}, y_1)$ are Alice's information – and builds a model $\boldsymbol{h}$ with a learning procedure $\boldsymbol{A}$ on $\mathcal{S}$. The adversary wants to infer Alice's information from the knowledge it already has of the learning algorithm and the training sample $\mathcal{S}$. Since we assume the adversary has unlimited knowledge outside Alice's sample, it can retrain the model on another training sample $\mathcal{S}'$ which is $\mathcal{S}$ where Alice has be replaced with a dummy sample. Then the followings can happened.

1. If the training algorithm is not differentially private – for example, if it is deterministic – the adversary can tune the dummy sample so that the training procedure on $\mathcal{S}'$ and $\mathcal{S}$ gives the same results, *i.e.* $h = h'$. The dummy sample is then very likely to contain information about Alice, which is a violation of her privacy.

2. On the contrary, if the algorithm is differentially private, given any dummy sample, the learning procedure produces the same model with a high probability. The adversary cannot tell whether the change in response is due to database change or randomization; therefore, Alice's information is safe.

Remark 23. *Note that in machine learning it is very rare to have a purely deterministic learning procedure – and even more so in deep learning. Indeed, as soon as we use stochastic gradient descent or random initialization, the learning procedure becomes random. In that sense, a learning procedure is a probabilistic mapping from $\mathfrak{S}_n$ to $\mathcal{H}$. This definition of privacy makes intensive use of this property, and designs specific internal randomization procedures to make the algorithms safer.*

5.1.2 Generalization of differential privacy

The notion of differential privacy is strongly correlated with the notion of "closeness", both in the input space $\mathfrak{S}_n$ and in the output space $\mathcal{H}$. Since the original work of Dwork *et al.* [52], many extensions have been introduced to adapt the definition to other possible input or output spaces, depending on the application setting – see [43] for a complete list. Recently, Chatzikokolakis *et al.* [31] introduced a general framework, called "metric-differential privacy", which encompasses many extensions of the initial definition.

Definition 9 (Metric-differential privacy)**.** *Let $\epsilon, \alpha \geq 0$, $(\mathcal{Z}, d_{\mathcal{Z}})$ an arbitrary input metric space, and $\mathcal{Z}'$ an output space. A probabilistic mapping $\mathfrak{m}$ from $\mathcal{Z}$ to $\mathcal{Z}'$ is called (α, ϵ)-metric private if for any $\boldsymbol{z_1}, \boldsymbol{z_2} \in \mathcal{Z}$ such that $d_{\mathcal{Z}}(\boldsymbol{z_1}, \boldsymbol{z_2}) \leq \alpha$ and for any $B \in \mathcal{A}(\mathcal{Z}')$ we have*

$$\mathbb{P}_{\boldsymbol{z}' \sim \mathfrak{m}(\boldsymbol{z_1})}\big[\boldsymbol{z}' \in B\big] \leq \exp(\epsilon)\, \mathbb{P}_{\boldsymbol{z}' \sim \mathfrak{m}(\boldsymbol{z_2})}\big[\boldsymbol{z}' \in B\big].$$

This definition provides a more general view on differential privacy, and adapts to more complex application settings such as geolocation and smart metering [31]. Note that we return to the classical definition of differential privacy when we set $\mathcal{Z} = \mathfrak{S}_n$, $\mathcal{Z}' = \mathcal{H}$, $d_{\mathcal{Z}}$ is the hamming distance, and $\alpha = 1$[1]. It is also worth noting that for any probability measures $\rho, \rho' \in \mathcal{P}(\mathcal{Z}')$, and for any $B \in \mathcal{A}(\mathcal{Z}')$, having $\rho(B)$ and $\rho'(B)$ within a multiplicative factor of $\exp(\epsilon)$ amounts to write

$$D_\infty(\rho, \rho') := \sup_{B \in \mathcal{A}(\mathcal{Z}')} \left| \ln \frac{\rho(B)}{\rho'(B)} \right| \leq \epsilon. \tag{5.1}$$

As a result, the two definitions of differential privacy we just saw simply enforce certain Lipschitz properties on probabilistic mappings based on a Renyi divergence with $\beta = \infty$. As a straightforward relaxation of these measure of privacy, Mironov [110] proposed to use an arbitrary Renyi divergence of order β to obtain a more general definition. This adaptation has principled theoretical advantages over the previous ones, making it the most practical formulation of the differential privacy introduced so far.

Definition 10 (Renyi differential privacy)**.** *Let $\epsilon, \alpha \geq 0$, $(\mathcal{Z}, d_{\mathcal{Z}})$ an arbitrary input metric space, and $\mathcal{Z}'$ an output space. A probabilistic mapping $\mathfrak{m}$ from $\mathcal{Z}$ to $\mathcal{Z}'$ is called* $(\alpha, \epsilon, \beta)$ Renyi-private *if for any $\boldsymbol{z_1}, \boldsymbol{z_2} \in \mathcal{Z}$ one has*

$$d_{\mathcal{Z}}(\boldsymbol{z_1}, \boldsymbol{z_2}) \leq \alpha \implies D_\beta(\mathfrak{m}(\boldsymbol{z_1}), \mathfrak{m}(\boldsymbol{z_2})) \leq \epsilon\,.$$

Reading note. *For the reader who might have skipped the additionnal results in Section 4.5, let us recall the definition of Renyi divergence. Let $\mathcal{Z}$ be an arbitrary space, and ρ, ρ' be two measures in $\mathcal{P}(\mathcal{Z})$, with probability density functions of g and g' according to a third measure ν. The* Renyi divergence of order β *writes*

$$D_\beta(\rho, \rho') := \frac{1}{\beta - 1} \log \int_{\mathcal{Y}} g'(y) \left(\frac{g(y)}{g'(y)} \right)^\beta d\nu(y)\ .$$

[1] The literature often sets $\alpha = 1$, and argue that one can always scale $d_{\mathcal{Z}}$ such that $d_{\mathcal{Z}} \leq 1$ fits the appropriated notion of "closeness". We keep $d_{\mathcal{Z}}$ unchanged and take an arbitrary α instead. Both definitions are equivalent.

The Renyi divergence [136] is a generalized divergence defined for any β on the interval $[1, \infty]$. It equals the Kullback-Leibler divergence when $\beta \to 1$, and the maximum divergence when $\beta \to \infty$. Note also that the previous results we obtained on the risk and robustness using the total variation distance, can be extended with minor variations to the Renyi divergence.

5.1.3 A unified view on privacy and robustness

At a first glance, the link between differential privacy and robustness may not be immediate. First of all, the studied mapping is not the same: in the domain of privacy, we look at the learning algorithms, while in the domain of robustness, we look at the model. Second, adversaries do not have the same objective: in privacy, the adversary wants to infer a subset of the training sample, while in robustness, it wants to force to misclassify. But if robustness and privacy have very different semantics, we can see how the definitions are based on the same mathematical foundation: Lipschitz continuity and information theory. If we consider $\mathcal{Z}$ an arbitrary input space with ℓ_p norm[2] and $\mathcal{Z}'$ an arbitrary output space, then the following holds.

A mapping $\mathfrak{m}$ is (α_p, ϵ)-robust *w.r.t.* D_β if and only if $\mathfrak{m}$ is $(\epsilon, \alpha_p, \beta)$ Renyi-private.

This equivalence – which comes directly from the definitions – is important from a theoretical point of view and has direct practical implications. Let us discuss some of them below.

Consequence 1: Unifying robustness and privacy

At the moment, the privacy and robust machine learning communities do not interact much, and most often use very different mathematical tools. The explicit link we have just established between the two domains shows that they have very similar goals and that technical tools should be shared or transferred from one domain to the other. Moreover, privacy and robustness, although orthogonal in their semantics, may have common application settings. For example, in the case of facial or speech recognition systems, there are real privacy and security issues to be addressed. In this context, instead of considering two separate methodologies, adding unnecessary layers of complexity, we could address both issues simultaneously within a unified framework.

Consequence 2: Using differential privacy as a fast track to robustness

Adversarial examples, in the context of deep learning, have only been studied for a few years now. On the other hand, differential privacy – although still young on the scale of computer science research – is at least 10 years ahead in terms of research accomplishments and societal awareness. It would therefore be interesting to take into account some of the lessons learned from privacy preserving machine learning and apply them to robustness. In particular, the literature on differential privacy makes extensive use of Gaussian noise injection and data processing inequality to

[2] Note that we set the ℓ_p norm for the definition to exactly match. But as we discussed in Chapter 2, we could define a range of other notion for imperceptibility. Thus this equivalence between differential privacy and robustness is much more general.

build private models. In the rest of this chapter, we show how we can easily adapt these two tools to build classes of robust classifiers – both *w.r.t.* D_β and D_{TV}.

5.2 Leveraging tools from differential privacy

5.2.1 Post-processing inequality

We just saw that differential privacy amounts to controlling the Renyi divergence between the outputs of a probabilistic mapping $\mathfrak{m}$. A crucial property of the Renyi divergence is the *Data processing inequality*. It is a well-known result from information theory which states that *"post-processing cannot increase information"* [12, 35]. In the context of privacy preserving data analysis, this means that no adversary – regardless of its computational power – can increase the maximal probability to recover information from the data sample. This is a fundamental property that any good definition of privacy should respect. Indeed, if an adversary can process the output of an algorithm to make the privacy guarrantees of the algorithm drop, then the privacy definition is not reliable. In its general form the data-processing inequality is as follows.

Theorem 14 (Post-processing inequality)**.** *Let us consider two arbitrary spaces* $\mathcal{Z}, \mathcal{Z}'$ *and* $\rho, \rho' \in \mathcal{P}(\mathcal{Z})$*. Then for any* $\psi \in \mathcal{F}_{\mathcal{Z}\times\mathcal{Z}'}$ *we have* $D_\beta(\psi\#\rho, \psi\#\rho') \leq D_\beta(\rho, \rho')$.

The proof of this statement exists in many forms in the information theory literature [12, 35, 161]. But the notations and concepts can sometimes vary from the ones we use in this manuscript. Hence, we recall the proof for readability.

Proof. Let us consider $\rho, \rho' \in \mathcal{P}(\mathcal{Z})$ with probability density functions g and g' with respect to a third measure $\nu \in \mathcal{P}(\mathcal{Z})$. Furthermore, let us denote $\psi\#g$ and $\psi\#g'$ the probability density functions of $\psi\#\rho, \psi\#\rho'$ with respect to $\psi\#\nu$[a]. Then we have

$$\begin{aligned} D_\beta(\psi\#\rho, \psi\#\rho') &= \frac{1}{\beta-1}\log\int_{\mathcal{Z}} (\psi\#g(\boldsymbol{z}))^\beta \big(\psi\#g'(\boldsymbol{z})\big)^{1-\beta} d\psi\#\nu(\boldsymbol{z}) \\ &= \frac{1}{\beta-1}\log\int_{\mathcal{Z}} \left(\frac{\psi\#g(\boldsymbol{z})}{\psi\#g'(\boldsymbol{z})}\right)^\beta d\psi\#\rho'(\boldsymbol{z}). \end{aligned}$$

Using the transfer theorem, we get

$$D_\beta(\psi\#\rho, \psi\#\rho') = \frac{1}{\beta-1}\log\int_{\mathcal{Z}'} \left(\frac{\psi\#g}{\psi\#g'}\circ\psi(\boldsymbol{z})\right)^\beta d\rho'(\boldsymbol{z}).$$

Since $\left(\frac{\psi\#g}{\psi\#g'}\circ\psi(Z)\right) = \mathbb{E}\left(\frac{g}{g'}(Z)\middle|\psi^{-1}(\mathcal{A}(\mathcal{Z}))\right)$ we get the following.

$$\begin{aligned} D_\beta(\psi\#\rho, \psi\#\rho') &= \frac{1}{\beta-1}\log\int_{\mathcal{Z}'} \left(\frac{\psi\#g}{\psi\#g'}\circ\psi(\boldsymbol{z})\right)^\beta d\rho'(\boldsymbol{z}) \\ &= \frac{1}{\beta-1}\log\int_{\mathcal{Z}'} \mathbb{E}\left(\frac{g(\boldsymbol{z})}{g'(\boldsymbol{z})}\middle|\psi^{-1}(\mathcal{A}(\mathcal{Z}))\right)^\beta d\rho'(\boldsymbol{z}). \end{aligned}$$

By using the Jensen inequality, and the property of the conditional expectation, we get

$$D_\beta(\psi\#\rho, \psi\#\rho') \leq \frac{1}{\beta-1}\log\int_{\mathcal{Z}'}\mathbb{E}\left(\frac{g(\boldsymbol{z})}{g'(\boldsymbol{z})}^\beta \middle| \psi^{-1}(\mathcal{A}(\mathcal{Z}))\right)d\rho'(\boldsymbol{z})$$
$$= \frac{1}{\beta-1}\log\int_{\mathcal{Z}'}\frac{g(\boldsymbol{z})}{g'(\boldsymbol{z})}^\beta d\rho'(\boldsymbol{z}) = D_\beta(\rho, \rho').$$

The above inequality concludes the proof. □

[a] $\psi\#g$ and $\psi\#g'$ exist thanks to the Radon-Nykodym Theorem.

Remark 24. *Note that the data-processing inequality is not limited to the Renyi divergence. Looking at the proof above, we see that the main argument is the Jensen inequality. Therefore, any divergence that can be written as the expectation of a convex function would give the same result. In fact, the data-processing inequality holds for any f-divergence[3]; which includes the total variation distance. Therefore the data -processing inequality holds both* w.r.t.D_β *and* D_{TV}.

In the context of robustness to adversarial examples, we want to use the data processing inequality to ease the design of robust randomized classifiers. In particular, let us suppose that we can build a randomized pre-processing $\mathfrak{p} : \mathcal{X} \to \mathcal{P}(\mathcal{X})$ such that for any $\boldsymbol{x} \in \mathcal{X}$ and any α_p-bounded perturbation $\boldsymbol{\tau}$, we have

$$D(\mathfrak{p}(\boldsymbol{x}), \mathfrak{p}(\boldsymbol{x}+\boldsymbol{\tau})) \leq \epsilon, \text{ with } D \in \{D_{TV}, D_\beta\}. \tag{5.2}$$

Then, thanks to the data-processing inequality, we can take any deterministic classifier c to build an (α_p, ϵ) robust classifier *w.r.t.* D defined as $\mathfrak{m} : \boldsymbol{x} \mapsto c\#\mathfrak{p}(\boldsymbol{x})$. This considerably simplifies the problem of building a class of robust models, but we still need to build $\mathfrak{p}$ in the first place. To do so, we keep taking inspiration from the privacy preserving community and study noise injection schemes.

5.2.2 Pre-processing with Gaussian noise injection

We want to build $\mathfrak{p}$ a randomized pre-processing that has a stable Renyi divergence and/or total variation distance. To do this, we analyze the simple procedure of injecting random noise directly on the image before sending it to a classifier. Noise injection is another fundamental tool in the literature of differential privacy. The most common noise choices are the Gaussian distribution and the Laplace distribution. Since the Renyi divergence is particularly well suited to the study of Gaussian distributions, we first use this type of noise injection. More precisely, in this section, we focus on a mapping that writes as follows.

$$\mathfrak{p} : \boldsymbol{x} \mapsto \mathcal{N}(\boldsymbol{x}, \Sigma), \tag{5.3}$$

for some given non-degenerate covariance matrix $\Sigma \in \mathcal{M}_{d\times d}(\mathbb{R})$. We extend our analysis in Section 5.4 to more general classes of noise, namely exponential families – which includes the

[3] see *e.g.* http://www.stat.yale.edu/~yw562/teaching/598/lec04.pdf for a very simple proof in the general case.

Laplace distribution. Let us now evaluate the maximal variation of Gaussian pre-processing $\mathfrak{p}$ when applied to an image $\boldsymbol{x} \in \mathcal{X}$ with and without perturbation.

Lemma 3. *Let $\beta > 1$, $\boldsymbol{x}, \boldsymbol{\tau} \in \mathcal{X}$ and $\Sigma \in \mathcal{M}_{d\times d}(\mathbb{R})$ a non-degenerate covariance matrix. Let $\rho = \mathcal{N}(\boldsymbol{x}, \Sigma)$ and $\rho' = \mathcal{N}(\boldsymbol{x} + \boldsymbol{\tau}, \Sigma)$, then $D_\beta(\rho, \rho') = \frac{\beta}{2}\|\boldsymbol{\tau}\|^2_{\Sigma^{-1}}$.*

Proof. Let $\beta > 1$. Let us denote g and g' respectively the probability density functions of ρ and ρ' with respect to the Lebesgue measure. We also set $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\tau}$ for readability. Then we have

$$
\begin{aligned}
D_\beta(\rho, \rho') &= \frac{1}{\beta - 1} \log \mathbb{E}_{\boldsymbol{z}\sim\rho'} \left[\left(\frac{g(\boldsymbol{z})}{g'(\boldsymbol{z})} \right)^\beta \right] \\
&= \frac{1}{\beta - 1} \log \mathbb{E}_{\boldsymbol{z}\sim\rho'} \left[\exp\left(\frac{\beta}{2} \left((\boldsymbol{z} - \boldsymbol{x}')^\intercal \Sigma^{-1} (\boldsymbol{z} - \boldsymbol{x}') - (\boldsymbol{z} - \boldsymbol{x})^\intercal \Sigma^{-1} (\boldsymbol{z} - \boldsymbol{x}) \right) \right) \right].
\end{aligned}
$$

By change of variable we get

$$
\begin{aligned}
&= \frac{1}{\beta - 1} \log \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(0,\Sigma)} \left[\exp\left(\frac{\beta}{2} \left(\boldsymbol{z}^\intercal \Sigma^{-1} \boldsymbol{z} - (\boldsymbol{z} + \boldsymbol{\tau})^\intercal \Sigma^{-1} (\boldsymbol{z} + \boldsymbol{\tau}) \right) \right) \right] \\
&= \frac{1}{\beta - 1} \log \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(0,\Sigma)} \left[\exp\left(\frac{\beta}{2} \left(-2\boldsymbol{z}^\intercal \Sigma^{-1} \boldsymbol{\tau} - \|\boldsymbol{\tau}\|^2_{\Sigma^{-1}} \right) \right) \right] \\
&= \frac{1}{\beta - 1} \log \int_{\mathbb{R}^d} \frac{\exp\left(-\frac{1}{2} \boldsymbol{z}^\intercal \Sigma^{-1} \boldsymbol{z} - \frac{\beta}{2} 2\boldsymbol{z}^\intercal \Sigma^{-1} \boldsymbol{\tau} - \frac{\beta}{2} \|\boldsymbol{\tau}\|^2_{\Sigma^{-1}} \right)}{(2\pi)^d \det(\Sigma)^{d/2}} d\boldsymbol{z} \,.
\end{aligned}
$$

Furthermore, for any $\boldsymbol{z} \in \mathbb{R}^d$, we have

$$
\begin{aligned}
&- \frac{1}{2} \boldsymbol{z}^\intercal \Sigma^{-1} \boldsymbol{z} - \frac{\beta}{2} 2\boldsymbol{z}^\intercal \Sigma^{-1} \boldsymbol{\tau} - \frac{\beta}{2} \|\boldsymbol{\tau}\|^2_{\Sigma^{-1}} \\
&= - \frac{1}{2} (\boldsymbol{z} + \beta\boldsymbol{\tau})^\intercal \Sigma^{-1} (\boldsymbol{z} + \beta\boldsymbol{\tau}) + \frac{\beta^2 - \beta}{2} \|\boldsymbol{\tau}\|^2_{\Sigma^{-1}} \,.
\end{aligned}
$$

Then we can re-write the Renyi divergence as follows

$$
\begin{aligned}
D_\beta(\rho, \rho') &= \frac{1}{\beta - 1} \log \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(-\beta\boldsymbol{\tau},\Sigma)} \left[\exp\left(\frac{\beta^2 - \beta}{2} \|\boldsymbol{\tau}\|^2_{\Sigma^{-1}} \right) \right] \\
&= \frac{1}{\beta - 1} \log \left(\exp\left(\frac{\beta^2 - \beta}{2} \|\boldsymbol{\tau}\|^2_{\Sigma^{-1}} \right) \right) \\
&= \frac{\beta}{2} \|\boldsymbol{\tau}\|^2_{\Sigma^{-1}} \,.
\end{aligned}
$$

This concludes the proof. □

Thanks to the above lemma, we know how to evaluate the level of Renyi-robustness that a Gaussian noise pre-processing brings to a classifier. Now that we have this result, thanks to Propo-

sition 3, we can also upper-bound the total variation distance between $\mathcal{N}(\boldsymbol{x}, \Sigma)$ and $\mathcal{N}(\boldsymbol{x} + \boldsymbol{\tau}, \Sigma)$. But this bound is not always tight. Besides, we can directly evaluate the total variation distance between two Gaussian distributions as follows.

Lemma 4. *Let $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ and $\Sigma \in \mathcal{M}_{d\times d}(\mathbb{R})$ a non-degenerate covariance matrix. Let $\rho = \mathcal{N}(\boldsymbol{x}, \Sigma)$ and $\rho' = \mathcal{N}(\boldsymbol{x} + \boldsymbol{\tau}, \Sigma)$, then $D_{TV}(\rho, \rho') = 2\Phi(\frac{\|\boldsymbol{\tau}\|_{\Sigma^{-1}}}{2}) - 1$ with Φ the cumulative density function of the standard Gaussian distribution.*

Proof. Let us denote g and g' respectively the probability density functions of ρ and ρ' with respect to the Lebesgue measure. Furthermore, we denote $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\tau}$. Then by definition of the total variation distance, we have $D_{TV}(\rho, \rho) = \rho(Z) - \rho'(Z)$ with $Z = \{\boldsymbol{z} \text{ s.t. } g(\boldsymbol{z}) \geq g'(\boldsymbol{z})\}$. In our case $g(\boldsymbol{z}) \geq g'(\boldsymbol{z})$ is equivalent to

$$(\boldsymbol{z} - \boldsymbol{x}')^\intercal \Sigma^{-1}(\boldsymbol{z} - \boldsymbol{x}') - (\boldsymbol{z} - \boldsymbol{x})^\intercal \Sigma^{-1}(\boldsymbol{z} - \boldsymbol{x}) \geq 0.$$

Then with the same simplification as above, we have

$$\begin{aligned}
\rho(Z) &= \mathbb{P}_{\boldsymbol{z}\sim\mathcal{N}(\boldsymbol{x},\Sigma)}\big((\boldsymbol{z} - \boldsymbol{x}')^\intercal \Sigma^{-1}(\boldsymbol{z} - \boldsymbol{x}') - (\boldsymbol{z} - \boldsymbol{x})^\intercal \Sigma^{-1}(\boldsymbol{z} - \boldsymbol{x}) \geq 0\big) \\
&= \mathbb{P}_{\boldsymbol{z}\sim\mathcal{N}(0,\Sigma)}\big((\boldsymbol{z} - \boldsymbol{\tau})^\intercal \Sigma^{-1}(\boldsymbol{z} - \boldsymbol{\tau}) - \boldsymbol{z}^\intercal \Sigma^{-1}\boldsymbol{z} \geq 0\big) \\
&= \mathbb{P}_{\boldsymbol{z}\sim\mathcal{N}(0,\Sigma)}\big(-2\boldsymbol{z}^\intercal \Sigma^{-1}\boldsymbol{\tau} + \|\boldsymbol{\tau}\|^2_{\Sigma^{-1}} \geq 0\big) \\
&= \mathbb{P}_{\boldsymbol{z}\sim\mathcal{N}(0,I_d)}\left(\boldsymbol{z}^\intercal \Sigma^{-1/2}\boldsymbol{\tau} \leq \frac{1}{2}\|\boldsymbol{\tau}\|^2_{\Sigma^{-1}}\right).
\end{aligned}$$

Furthermore, if $\boldsymbol{z} \sim \mathcal{N}(0, I_d)$ then $\boldsymbol{z}^\intercal \Sigma^{-1/2}\boldsymbol{\tau} \sim \mathcal{N}(0, \|\boldsymbol{\tau}\|^2_{\Sigma^{-1}})$; hence we also have $\frac{\boldsymbol{z}^\intercal \Sigma^{-1/2}\boldsymbol{\tau}}{\|\boldsymbol{\tau}\|_{\Sigma^{-1}}} \sim \mathcal{N}(0, 1)$. Accordingly we get

$$\rho(Z) = \mathbb{P}_{\boldsymbol{z}\sim\mathcal{N}(0,1)}\left(\boldsymbol{z} \leq \frac{1}{2}\|\boldsymbol{\tau}\|_{\Sigma^{-1}}\right) = \Phi\left(\frac{1}{2}\|\boldsymbol{\tau}\|_{\Sigma^{-1}}\right).$$

By symmetry we get that $\rho'(A) = 1 - \rho(A) = 1 - \Phi\big(\frac{1}{2}\|\boldsymbol{\tau}\|_{\Sigma^{-1}}\big)$. We then get

$$D_{TV}(\mu, \nu) = 2\Phi\left(\frac{\|\boldsymbol{\tau}\|_{\Sigma^{-1}}}{2}\right) - 1$$

which concludes the proof. □

Note that both figures increase with the Mahalanobis distance of $\boldsymbol{\tau}$. Furthermore, we see that the greater the entropy of the Gaussian noise we inject, the smaller the distance between distributions. If we simplify the covariance matrix by setting $\Sigma = \sigma^2 I_d$, it means that we can build more or less robust randomized classifiers against ℓ_2 adversaries, depending on σ.

Theorem 15 (Robustness of Gaussian pre-processing)**.** *Let us consider $c : \mathcal{X} \to \mathcal{Y}$ a deterministic classifier, $\sigma > 0$ and $\mathfrak{p} : \boldsymbol{x} \mapsto \mathcal{N}(\boldsymbol{x}, \sigma^2 I_d)$ a pre-processing probabilistic mapping. Then the randomized classifier $\mathfrak{m} := c\#\mathfrak{p}$ is*

- $(\alpha_2, \frac{(\alpha_2)^2\beta}{2\sigma})$-*robust* w.r.t. D_β *against* ℓ_2 *adversaries.*
- $(\alpha_2,\ 2\Phi\left(\frac{\alpha_2}{2\sigma}\right) - 1)$-*robust* w.r.t. D_{TV} *against* ℓ_2 *adversaries.*

Proof. Let $\boldsymbol{x}, \boldsymbol{\tau} \in \mathcal{X}$ such that $\|\boldsymbol{\tau}\|_2 \leq \alpha_2$. Thanks to Lemma 3 we have

$$D_\beta(\mathfrak{p}(\boldsymbol{x}), \mathfrak{p}(\boldsymbol{x}+\boldsymbol{\tau})) = \frac{\beta}{2}\|\boldsymbol{\tau}\|^2_{\Sigma^{-1}} = \frac{\beta}{2\sigma^2}\|\boldsymbol{\tau}\|_2^2 \leq \frac{\beta(\alpha_2)^2}{2\sigma^2}.$$

Similarly, thanks to Lemma 4, we get

$$D_{TV}(\mathfrak{p}(\boldsymbol{x}), \mathfrak{p}(\boldsymbol{x}+\boldsymbol{\tau})) = 2\Phi\left(\frac{\|\boldsymbol{\tau}\|_{\Sigma^{-1}}}{2}\right) - 1 \leq 2\Phi\left(\frac{\alpha_2}{2\sigma}\right) - 1.$$

Finally, from the data-processing inequality – Theorem 14, we get both

$$D_\beta(\mathfrak{m}(\boldsymbol{x}), \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})) \leq \frac{\beta(\alpha_2)^2}{2\sigma^2},$$

and

$$D_{TV}(\mathfrak{m}(\boldsymbol{x}), \mathfrak{m}(\boldsymbol{x}+\boldsymbol{\tau})) \leq 2\Phi\left(\frac{\alpha_2}{2\sigma}\right) - 1.$$

The above inequalities conclude the proof. □

Theorem 15 means that we can build simple noise injection schemes as pre-processing of state-of-the-art image classification models and keep track of the maximal loss of accuracy under attack of the resulting randomized classifier. These results also highlight the profound link between randomized classifiers and randomized smoothing. Even-though our findings are of different nature, both techniques use the same base mechanism – Gaussian noise injection. Therefore, Gaussian pre-processing is a principled defense method that can be analyzed trough several standpoints, including certified robustness and statistical learning theory.

5.3 Numerical validation: the case study of the neural network

To illustrate our theoretical results, we train a randomized neural networks with Gaussian pre-processing during training and inference on CIFAR-10 and CIFAR-100. Based on this randomized classifier, we study the impact of randomization on the standard accuracy of the network, and compare the theoretical trade-off between accuracy and robustness with experimental results against state-of-the-art attacks. Let us first start by presenting the experimental setup we use. For direct access to the implementation, one can refer to the following Github repository.

`https://github.com/MILES-PSL/Adversarial-Robustness-Through-Randomization`

5.3.1 Experimental setup

Important remark on the image space we consider

At the time of these experiments, we were using attack implementations that take an image with pixels scaled between -1 and 1; meaning that $\mathcal{X} = [-1, 1]^d$. All the above results remain valid in this setting, but we have to adapt the perception thresholds – multiply them by 2. This is why we report results with $\alpha_\infty = 0.06$ and $\alpha_2 = 1.6$.

Architecture and training procedure

All the neural networks we use in this section are WideResNets – see [177] – with 28 layers, a widen factor of 10, a dropout factor of 0.3 and LeakyRelu activation with a 0.1 slope. To train an undefended classifier we use the following hyper-parameters.

- *Number of Epochs:* 200
- *Batch size:* 400
- *Loss function:* Cross Entropy Loss
- *Optimizer :* Stochastic gradient descent algorithm with momentum 0.9, weight decay of 2×10^{-4} and a learning rate that decreases during the training as follows:

$$lr = \begin{cases} 0.1 & \text{if} \quad 0 \leq \text{epoch} < 60 \\ 0.02 & \text{if} \quad 60 \leq \text{epoch} < 120 \\ 0.004 & \text{if} \quad 120 \leq \text{epoch} < 160 \\ 0.0008 & \text{if} \quad 160 \leq \text{epoch} < 200. \end{cases}$$

To transform these classical networks into randomized classifiers, we inject noise drawn from Gaussian distributions, each with various standard deviations directly on the image before passing it through the network. Both during training and test, for computational efficiency, we evaluate the performance of the the algorithm over a single run for every images; hence no Monte Carlo estimator is used. However, in practice, the test-time accuracy and accuracy under attack are quite stable when evaluated over the entire test dataset.

Remark 25. *To train a neural network with adversarial training we use the same hyper-parameters as above, and generate adversarial examples during training using an ℓ_∞ adversary with 7 iterations. Furthermore, we want to build state-of-the-art models; hence we use data augmentation during the leaning procedure – which explains some differences with the results from Chapter 3.*

Threat models

To compare the empirical performances of our method with adversarial training, we consider two ℓ_p adversaries with thresholds corresponding to CIFAR datasets

- *An ℓ_∞ adversary with perturbation bounded by* 0.06. To model this adversary we use the PGD attack with $t_{max} = 20$ iterations and a step size $s = 0.006$.

- *An ℓ_2 adversary with perturbation bounded by* 1.6. To model this adversary we use the C&W attack with 60 iterations, a learning rate equal to 0.01, 9 binary search steps, and an initial constant of $\kappa = 0.001$.

As we already mentioned in Chapter 3, when evaluating a defense against adversarial examples, it is crucial to test the robustness of the method against the best possible attack. More precisely, when evaluating randomized algorithms, one must provide the adversary with the expected results from the classifier. Here, the actual distribution of the outputs can be difficult to evaluate since the Gaussian distribution passes through the network. Thus, to build an adversarial example, the adversary has to use a Monte Carlo mean estimator. For each input, we estimate the expected output of the network for 80 different samples of the Gaussian noise.

Remark 26. *In Chapter 3, we knew the exact distribution of the randomized classifier, so we didn't have to use Monte Carlos sampling. Therefore, to keep the calculation tractable, we decrease the number of gradient steps of the attacks compared to the previous experiments.*

5.3.2 Results

Figures 5.2 and 5.3 show the accuracy and the minimum level of accuracy under attack of our randomized neural network for several levels of injected noise. We can see – Figure 5.2 – that the precision decreases as the noise intensity grows. In that sense, the noise must be calibrated to preserve both accuracy and robustness against adversarial attacks – it must be large enough to preserve robustness and small enough to preserve accuracy. This is to be expected, because the greater the entropy of the classifier, the less precise it gets.

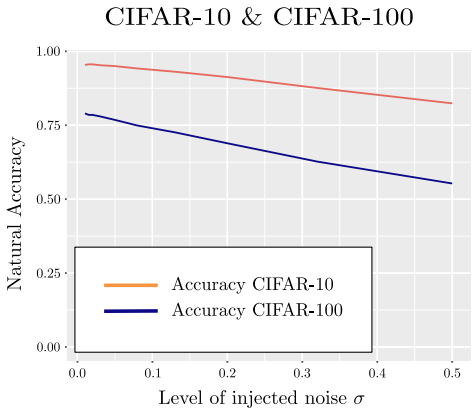


Figure 5.2: Impact of the standard deviation of the Gausian noise on accuracy in a randomized model on CIFAR-10 and CIFAR-100 dataset.

Furthermore, when injecting Gaussian noise as a defense mechanism, the resulting randomized network $\mathfrak{m}$ is both $(\alpha_2, \frac{(\alpha_2)^2}{2\sigma})$-robust *w.r.t.* D_1 and $(\alpha_2, 2\Phi\left(\frac{\alpha_2}{2\sigma}\right) - 1)$-robust *w.r.t.* D_{TV} against ℓ_2 adversaries. Therefore thanks to Theorems 10 and 13 we have that

$$\mathcal{R}^{\text{adv}}(\mathfrak{m}; \alpha_2) - \mathcal{R}(\mathfrak{m}) \leq 2\Phi\left(\frac{\alpha_2}{2\sigma}\right) - 1, \text{ and} \tag{5.4}$$

$$\mathcal{R}^{\text{adv}}(\mathfrak{m}; \alpha_2) - \mathcal{R}(\mathfrak{m}) \leq 1 - e^{-\frac{(\alpha_2)^2}{2\sigma}} \, \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}_{|\mathcal{X}}} \left[e^{-H(\mathfrak{m}(\boldsymbol{x}))} \right]. \tag{5.5}$$

Figure 5.3 illustrates the theoretical lower bound on accuracy under attack – based on the minimum gap between Equations (5.4) and (5.5) – for different standard deviations. The term in entropy has been estimated using a Monte Carlo method with 10^4 simulations. The trade-off between accuracy and robustness appears with respect to the noise intensity. With small noises, the accuracy is high, but the guaranteed accuracy drops fast with respect to the magnitude of the adversarial perturbation. Conversely, with bigger noises, the accuracy is lower but decreases slowly with respect to the magnitude of the adversarial perturbation. Overall, we get strong accuracy guarantees against small adversarial perturbations, but when the perturbation is bigger than 0.5 on CIFAR-10 – resp. 0.3 on CIFAR-100, the guarrantees are still not sufficient.

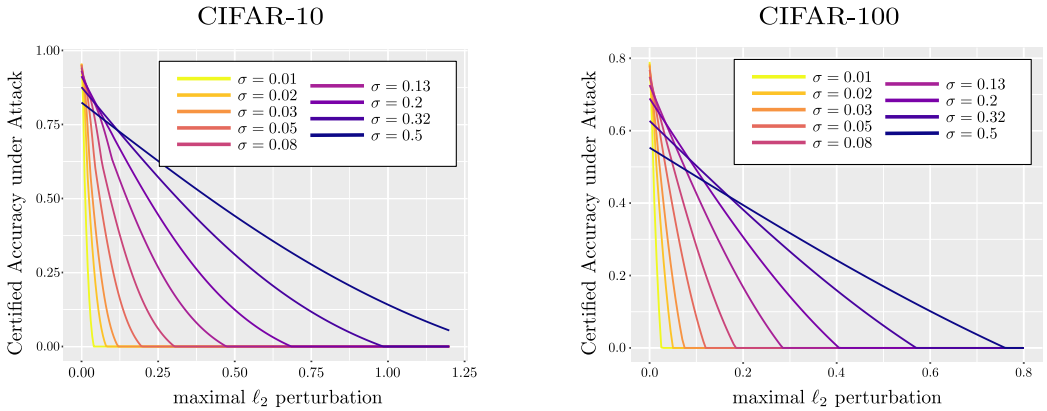


Figure 5.3: Guaranteed accuracy of different randomized models with Gaussian noise given the ℓ_2 norm of the adversarial perturbations.

Remark 27. *Note that the maximal ℓ_2 perturbation considered as imperceptible for CIFAR datasets is $\alpha_2 = 1.6$. Hence our theoretical bounds are still not sufficient to consider worst case threats.*

Table 5.1 shows that in practice, randomized networks reach an accuracy under attack much higher than the theoretical lower bound we obtained, and keep a good accuracy against much larger perturbations. While Figure 5.3 illustrates theoretical robustness against increasing adversarial perturbations, Table 5.1 illustrates this trade-off experimentally. It compares the standard accuracy and accuracy under attack of randomized networks with Gaussian pre-processing for different standard deviations against the adversarial training of Madry *et al.* [103]. We observe that the accuracy on the standard images for the noise injection method is similar to the one we

Table 5.1: Accuracy under attack of randomized neural networks with Gaussian pre-processing for different standard deviations versus adversarial training by Madry *et al.* [103] – AT. The first line refers to the baseline without attack. Results are presented for both CIFAR-10 and CIFAR-100 datasets.

Dataset	**Method**	**Accuracy without attack**	ℓ_∞**-PGD** $\alpha_\infty = 0.06$	ℓ_2**-C&W** $\alpha_2 = 1.6$
CIFAR-10	Undefended	0.96	0.00	0.00
	AT [103]	0.87	0.46	0.47
	Gaussian $\sigma = 0.32$	**0.88**	0.57	**0.51**
	Gaussian $\sigma = 0.5$	0.82	**0.59**	0.49
CIFAR-100	Undefended	0.79	0.00	0.00
	AT [103]	0.58	0.26	0.22
	Gaussian $\sigma = 0.32$	**0.63**	**0.40**	**0.29**
	Gaussian $\sigma = 0.5$	0.56	0.35	0.30

obtain for the adversarial training. Moreover, Gaussian pre-processing is more robust than adversarial training for both PGD and C&W attacks. These experiments show that randomized defenses can be competitive given the intensity of the noise injected into the network.

Remark 28. *Our theoretical findings only hold for ℓ_2 adversaries. Hence we were not guaranteed to have any protection against ℓ_∞-PGD. Nevertheless, our method presents state-of-the-art experimental robustness against this attack as well.*

5.4 Additional results: extension to the exponential family and experiments against ℓ_1 adversaries

5.4.1 Extension to broader classes of noise injection

In the previous section we demonstrated, based on insights from the literature on differential privacy, how Gaussian pre-processing can help build more robust models against adversarial examples. Now, from a differential privacy perspective, the use of Laplace noise is at least as frequent as Gaussian noise. Moreover, the above results only work for ℓ_2 adversaries, so we wonder if we could use other types of noise to defend against other ℓ_p adversaries. In this section, we extend the previous results to a larger family of noises, namely the exponential family.

Definition 11 (Exponential family)**.** *Let $d' \in \mathbb{N}$, Θ be an open convex set of $\mathbb{R}^{d'}$, and $\boldsymbol{\theta} \in \Theta$. Let ρ be a probability measure in $\mathcal{P}\left(\mathbb{R}^d\right)$ that admits a probability density function either with respect*

to the Lebesgue or the counting measure. ρ is said to be part of the exponential family *of parameter $\boldsymbol{\theta}$ if it has the following probability density function*

$$g_{\boldsymbol{\theta}}(\boldsymbol{z}) = \exp(t(\boldsymbol{z})^\intercal \boldsymbol{\theta} - u(\boldsymbol{\theta}) + v(\boldsymbol{z}))$$

where $t(\boldsymbol{z})$ is a sufficient statistics, v a carrier measure (either for a Lebesgue or a counting measure) and $u(\boldsymbol{\theta}) = \log \int_{\mathbb{R}^d} \exp(t(\boldsymbol{z})^\intercal \boldsymbol{\theta} + v(\boldsymbol{z}))d\boldsymbol{z}$. We denote EF($\boldsymbol{\theta}, t, v$) the set of such probability distributions.

To show the robustness of randomized networks with noise injected from an exponential family, we need to define the notion of modulus of continuity.

Definition 12 (Modulus of continuity)**.** *Let us consider $d, d' \in \mathbb{N}$ and an arbitrary function $f : \mathbb{R}^d \to \mathbb{R}^{d'}$. f admits a non-decreasing modulus of continuity with respect to $\|.\|_p$ and $\|.\|_q$ if there exists a non-decreasing function $\omega_f^{p,q} : \mathbb{R}^+ \to \mathbb{R}^+$ such that for all $\boldsymbol{x}, \boldsymbol{x'} \in \mathbb{R}^d$ we have,*

$$\|f(\boldsymbol{x}) - f(\boldsymbol{x})\|_q \leq \omega_f^{p,q}(\left\|\boldsymbol{x} - \boldsymbol{x'}\right\|_p).$$

The definition of modulus of continuity is a simple relaxation of the Lipschitz continuity. Indeed, if $\omega_f^{p,q}$ is linear with 0 intercept and slope W, then f is W-Lipschitz with respect to $\|.\|_p$ and $\|.\|_q$. We can now show how to control the Renyi divergence of a pre-processing based on an exponential family as follows.

Lemma 5. *Let $d' \in \mathbb{N}$, $\beta > 1$ and $\boldsymbol{\theta} \in \mathbb{R}^{d'}$. Let ρ be a probability measure from the exponential family EF(θ, t, v) where t and v have non-decreasing modulus of continuity ω_t and ω_v. Let us define the pre-processing $\mathfrak{p}$ that to any image in $\mathcal{X}$ adds a noise drawn from ρ. Then, for any $\boldsymbol{x} \in \mathcal{X}$ and any α_p-bounded perturbation $\boldsymbol{\tau}$ – for an ℓ_p adversary – we have*

$$D_\beta(\mathfrak{p}(\boldsymbol{x}), \mathfrak{p}(\boldsymbol{x} + \boldsymbol{\tau})) \leq \|\boldsymbol{\theta}\|_2 \omega_t^{p,2}(\alpha_p) + \omega_v^{p,1}(\alpha_p).$$

Proof. Let us denote $g_{\boldsymbol{\theta}}$ the probability density function of ρ and δ_a the Dirac measure mapping any element to 1 if it equals a and to 0 otherwise. Then by definition of the convolution with the Dirac measure, we have that

$$\begin{aligned} & D_\beta(\mathfrak{p}(\boldsymbol{x}), \mathfrak{p}(\boldsymbol{x} + \boldsymbol{\tau})) \\ = \; & D_\beta(\rho * \delta_{\boldsymbol{x}}, \rho * \delta_{\boldsymbol{x}+\boldsymbol{\tau}}). \end{aligned}$$

Since the Renyi divergence is increasing with respect to β, we have

$$\begin{aligned} & \leq D_\infty(\rho * \delta_{\boldsymbol{x}}, \rho * \delta_{\boldsymbol{x}+\boldsymbol{\tau}}) \\ & = \log \sup_{\boldsymbol{z} \in \mathbb{R}^{d'}} \frac{g_{\boldsymbol{\theta}}(\boldsymbol{z} - \boldsymbol{x})}{g_{\boldsymbol{\theta}}(\boldsymbol{z} - (\boldsymbol{x} + \boldsymbol{\tau}))} \\ & = \log \sup_{\boldsymbol{z} \in \mathbb{R}^{d'}} \exp((t(\boldsymbol{z} - \boldsymbol{x}) - t(\boldsymbol{z} - (\boldsymbol{x} + \boldsymbol{\tau})))^\intercal \boldsymbol{\theta} + v(\boldsymbol{z} - \boldsymbol{x}) - v(\boldsymbol{z} - (\boldsymbol{x} + \boldsymbol{\tau}))). \end{aligned}$$

By Cauchy Schwartz inequality, we get

$$\leq \sup_{\boldsymbol{z}\in\mathbb{R}^{d'}} \|\theta\|_2 \|t(\boldsymbol{z}-\boldsymbol{x}) - t(\boldsymbol{z}-(\boldsymbol{x}+\boldsymbol{\tau}))\|_2 + |v(\boldsymbol{z}-\boldsymbol{x}) - v(\boldsymbol{z}-(\boldsymbol{x}+\boldsymbol{\tau}))|.$$

Furthermore, since t and v have modulus of continuity ω_t and ω_v, we get

$$\begin{aligned}
&\leq \|\theta\|_2 \omega_t^{p,2}(\|\boldsymbol{x}+\boldsymbol{\tau},\boldsymbol{x}\|_p) + \omega_v^{p,1}(\|\boldsymbol{x}+\boldsymbol{\tau},\boldsymbol{x}\|_p) \\
&\leq \|\theta\|_2 \omega_t^{p,2}(\alpha_p) + \omega_v^{p,1}(\alpha_p).
\end{aligned}$$

The above inequality concludes the proof. □

Thanks to this result, we identified a range of possible distributions to build robust classifiers. In particular, we can build a pre-processing based on the Laplace distribution to defend against ℓ_1 adversaries. However, the bound can be loose due to the use of the Cauchy Schwartz inequality. The following result uses the same reasoning to get a tighter bound for a pre-processing based on Laplace noise injection.

Theorem 16 (Robustness for Laplace pre-processing)**.** *Let us consider $c : \mathcal{X} \to \mathcal{Y}$ a deterministic classifier, $\sigma > 0$ and a pre-processing $\mathfrak{p}$ that to any image adds noise drawn from $Lap(0, \sigma I_d)$ where $Lap(0, \sigma I_d)$ is the product measure of d uni-variate Laplace distributions with scale parameter σ and mean 0. Then the randomized classifier $\mathfrak{m} := c\#\mathfrak{p}$ is $(\alpha_1, \frac{\alpha_1}{\sigma})$-robust* w.r.t. *$D_\beta$ against ℓ_1 adversaries.*

Proof. First, let us recall that a uni-variate Laplace distribution with scale σ and mean 0 defines an exponential family with $t : x \mapsto |x|$, $v : x \mapsto 0$, and $\theta = -\frac{1}{\sigma}$. Furthermore, the distribution of $\mathrm{Lap}(0, \sigma I_d)$ is defined as the product of d uni-variate Laplace distributions. Then using the same first steps as in Lemma 5, for any $\boldsymbol{x}, \boldsymbol{\tau} \in \mathcal{X}$ such that $\|\boldsymbol{\tau}\|_1 \leq \alpha_1$, we have

$$D_\beta(\mathfrak{p}(\boldsymbol{x}), \mathfrak{p}(\boldsymbol{x}+\boldsymbol{\tau})) \leq \log \sup_{\boldsymbol{z}\in\mathbb{R}^d} \exp\left(-\sum_{i=1}^{d} \frac{|\boldsymbol{z}_i - \boldsymbol{x}_i| - |\boldsymbol{z}_i - (\boldsymbol{x}_i + \boldsymbol{\tau}_i)|}{\sigma}\right).$$

Since $\sum_{i=1}^{d} |\boldsymbol{z}_i - (\boldsymbol{x}_i + \boldsymbol{\tau}_i)| - |\boldsymbol{z}_i - \boldsymbol{x}_i| \leq \|\boldsymbol{\tau}\|_1$ for any $\boldsymbol{z}$, we get

$$D_\beta(\mathfrak{p}(\boldsymbol{x}), \mathfrak{p}(\boldsymbol{x}+\boldsymbol{\tau})) \leq \log\left(\exp\left(\frac{\|\boldsymbol{\tau}\|_1}{\sigma}\right)\right) \leq \frac{\alpha_1}{\sigma}.$$

The above inequality concludes the proof. □

5.4.2 Additional experiments for ℓ_1 adversaries

To illustrate this result, we train a randomized neural network with Laplace pre-processing during training and inference on CIFAR-10 and CIFAR-100. As for the Gaussian case, we study the impact of randomization on the standard accuracy of the network, and on its robustness. We use

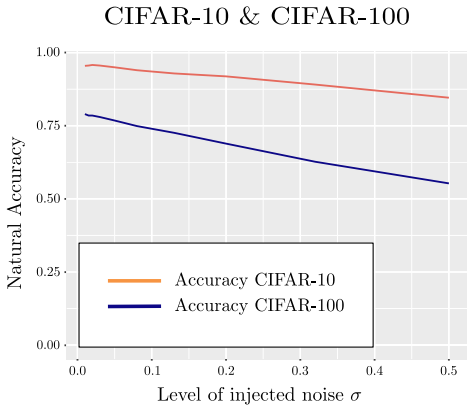


Figure 5.4: Impact of the standard deviation of the Laplace noise on accuracy in a randomized model on CIFAR-10 and CIFAR-100 dataset.

the same experimental protocol as above, but instead of using an ℓ_2 adversary – C&W – we take its adaptation to ℓ_1 attacks called Elastic Net attack – EAD [33]. The optimization algorithm and implementation techniques are the same. The only modification is that we add an ℓ_1 term to the objective function, which gives

$$\kappa_1 \times \|\boldsymbol{\tau}\|_1 + \kappa_2 \times \|\boldsymbol{\tau}\|_2 + g(\boldsymbol{x} + \boldsymbol{\tau}). \tag{5.6}$$

Moreover, we select $\alpha_1 = 7$ for the algorithm, as it corresponds to the volume condition we discussed in Chapter 2. This value may seem important, but the standard perturbations for an ℓ_1 adversary are usually much larger than those for an ℓ_2 or ℓ_∞ adversary – see [33] for more details.

Figures 5.4 and 5.5 show the standard accuracy and the minimum level of accuracy under attack of our new randomized network for multiple levels of injected noise. As in the Gaussian case, the precision decreases as the noise intensity grows. In addition, the theoretical trade-off between precision and robustness appears with respect to the noise intensity. We achieve high precision guarantees against small ℓ_1 perturbations, but when the perturbation is greater than 0.2, the guarantees decrease.

Remark 29. *The above theoretical limits only draw the bounds of Theorem 13. Indeed, we have not demonstrated that Laplace pre-processing gives robustness for the total variation yet. We will have to study it in future works to improve the theoretical worst case accuracy.*

Table 5.2 shows that in practice, Laplace pre-processing achieves higher accuracy under attack than the theoretical bounds against ℓ_1 and ℓ_∞ adversaries. It compares the precision and accuracy under attack of randomized networks with Laplace preprocessing for different standard deviations. As in the Gaussian case, we found out that randomized defenses can be competitive given the intensity of the noise injected in the network.

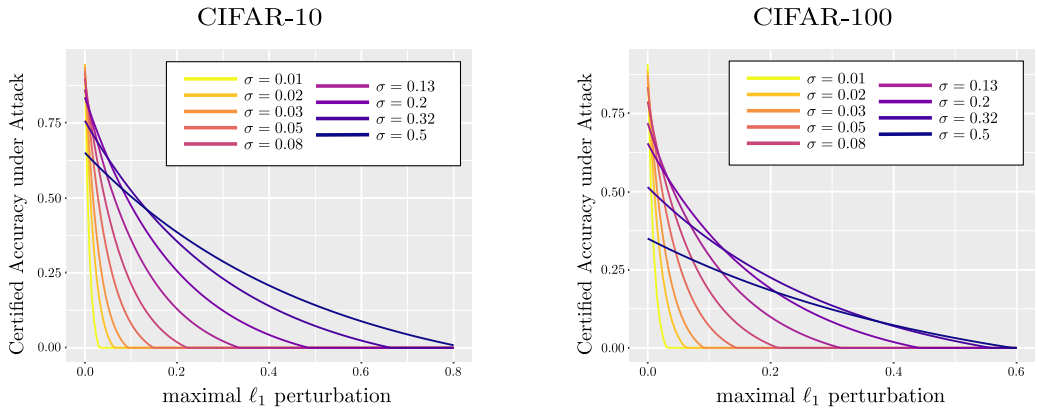


Figure 5.5: Guaranteed accuracy of different randomized models with Laplace noise given the ℓ_1 norm of the adversarial perturbations.

Table 5.2: Accuracy under attack of randomized neural network with Laplace pre-processing for different standard deviations. CIFAR-10 and CIFAR-100 datasets.

Dataset	**Method**	**Accuracy without attack**	ℓ_∞**-PGD** $\alpha_\infty = 0.06$	ℓ_1**-EAD** $\alpha_1 = 7$
CIFAR-10	Laplace $\sigma = 0.32$	0.89	0.58	0.45
	Laplace $\sigma = 0.5$	0.85	0.59	0.47
CIFAR-100	Laplace $\sigma = 0.32$	0.65	0.30	0.36
	Laplace $\sigma = 0.5$	0.67	0.33	0.32

5.5 Lessons learned and future works

In this chapter, we presented simple schemes for building robust randomized classifiers. Based on the links we have established with the privacy preserving literature, we show that Gaussian noise injection can provide principled robustness against ℓ_2 adversarial attacks. Then, we used Gaussian noise injection with advanced neural network architectures to build robust and accurate models and supported our theoretical claims with a set of experiments on CIFAR-10 and CIFAR-100. We achieve both good standard accuracy and state-of-the-art robustness. This responds to **Q1** and **Q2** from a practical point of view.

> We can build randomized classifiers that are robust for both the total variation distance and the Renyi divergence.

Finally, we extended our analysis to take into account the noise injection from an exponential family. This allows us to build a principled defense against ℓ_1 adversaries based on Laplace noise

injection. The practical schemes we developed could be improved in several ways. Among them, we list below some possible approaches.

Future work 1: widen the scope of adversaries

So far, we have identified Gaussian noise injection to defend against ℓ_2 attacks, and we have extended our study to ℓ_1 adversaries. But Lemma 5 is quite general and could lead us to study other noises for different ℓ_p adversaries. The question whether we can build noise injection schemes to defend against ℓ_∞ adversaries also remains open. To this end, we could use the fundamental link that our framework shares with randomized smoothing to study noises that proved useful for this defense. In particular, Yang *et al.* [173] studied new classes of noise by using the Wulff Crystals theory. This could open some interesting leads for more sophisticated noise injection mechanisms.

Future work 2: injecting noise anywhere in the network

In the particular case of neural network, we can decompose the deterministic hypothesis into successive compositions of functions $\boldsymbol{h}(\boldsymbol{x}) := \boldsymbol{h}^{(N)} \circ \cdots \circ \boldsymbol{h}^{(1)}(\boldsymbol{x})$. Thus, from a theoretical point of view, the data-processing inequality allows us to inject noise at any stage $\boldsymbol{h}^{(i)}$ of the network and we would obtain results similar to those of this chapter. Nevertheless, noise injection only works if the maximum perturbation that the adversary can produce is limited, so the network should have a specific design for the scheme to be applicable. The design architectures that allow for noise injection in the network – *e.g.* networks with very small Lipschitz constant – rather than directly on the image would give a very interesting new perspective on the schemes we have just designed.

Future work 3: establishing deeper connections with differential privacy

The link we have established with differential privacy is fundamental, and we are far from having studied all its aspects. For example, we could design much more sophisticated random schemes based on this link, for example by using the exponential mechanism, or differentially private voting procedures[50]. Differential privacy is also known to have very interesting properties for generalization based on stability theory [11]. Thus, we could adapt previous results to improve the analysis we presented in Chapters 4 and 5.

6 Conclusion & open problems

Contents

6.1 Summary of the results

In this thesis, we studied the problem of adversarial classification from different angles, using a series of theoretical and practical tools. We have tried to analyze the problem using both a high level and a more precise analysis. Overall, our work advocates for the development of a probabilistic viewpoint on adversarial examples is a principled way to better understand and to build new useful theories. We can summarize our findings as follows.

1. We first presented the problem as an infinite zero-sum game, and analyzed the fundamental nature of the game under different types of regularization using game theory. This analysis gave us a better understanding of the current analytical framework used in the adversarial examples community, and led us to argue for randomization as a principled defense against adversarial attacks.

2. We then studied in more detail the key properties that random defenses should observe in order to build classifiers that are robust while maintaining high standard accuracy. To this end, we developed new approaches to study the robustness of randomized classifiers using information theory, probability theory and statistical learning theory. We identified sufficient conditions for randomized algorithms to be robust and we studied the generalization property of these classifiers.

3. Finally, we developed practical schemes for designing robust random classifiers using information theory and lessons learned from the privacy literature. This shows that we can build robust random classifiers based on state-of-the-art neural network architectures, and paves the way to exciting future works, both in theory and practice.

We hope that our analysis helped the community move forward and brought new and interesting perspectives on the difficult problem which is adversarial classification. The field is still young and

many research directions are still open. Throughout the manuscript, we have discussed future works that correspond - more or less - to direct improvements of our results. But here, we would like to take the time to present some more challenging open problems that would require more investment in terms of time and resources.

6.2 Open problem 1: Revisiting the adversarial framework

Back to adversarial classification, the community studies the Problem (1.3) defined as

$$\inf_{h\in\mathcal{H}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau} \text{ s.t. } \|\boldsymbol{\tau}\|_p\leq\alpha_p} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x}+\boldsymbol{\tau}),y)\right]. \tag{1.3}$$

As we discussed in this work, from a functional point of view, the same problem writes

$$\inf_{\boldsymbol{h}\in\mathcal{H}} \sup_{\boldsymbol{\psi}\in\mathcal{F}_{\mathcal{X}|\alpha_p}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}(\boldsymbol{h}(\boldsymbol{\psi}(\boldsymbol{x})),y)]\,. \tag{6.1}$$

This is the main focus of the literature, and the current analysis tends to show that the inf problem cannot have a good solution. Empirical evidence support these conclusions [28, 103]. Nevertheless, in chapter 3, we have shown that the conclusions we draw about the nature of adversarial classification can change as soon as we modify the problem, even very slightly. This raises the following question.

Remark 30. *Note that even advanced methods based on randomization such as the one presented in chapter 5, or randomized smoothing achieve an accuracy under attack slightly above* 0.5*, which is not enough to consider classifiers as ultimately robust.*

> Is the problem of classification under adversarial perturbation ill-posed?

By ill-posed, we mean that the problem is not modeling a real threat scenario. We believe that the mathematical convenience of the current formulation has led researchers to study an over-simplified problem in which the opponent is unrealistically strong. This idea is shared by recent literature [46, 62]. We therefore need to rethink the mathematical framework to make it more representative of real threat scenarios.

In Chapter 3, we first pointed out that adversaries who can attack all points in the distribution are unrealistic and presented simple ways to mitigate this concern. In addition, we could also ask whether the better way of evaluating the performance of an attack is by computing the expectation over attacking all points. If we come back to the example of the self-driving car, an adversary who wants to trigger an accident may only want to change the decision on a very limited number of points and not care about the others. Similarly, the classifier may have different priorities on the points it has to defend. If we let $\mathcal{L}^{\text{def}}$ and $\mathcal{L}^{\text{adv}}$ encode the different policies of the defender and the adversary regarding the points, the problem now writes

$$\begin{cases} \sup\limits_{\psi \in \mathcal{F}_{\mathcal{X}|\alpha_p}} & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathcal{L}^{\text{adv}}(\boldsymbol{h}(\boldsymbol{\psi}(\boldsymbol{x})), y)\right], \text{ for a given hypothesis } \boldsymbol{h}. \\ \inf\limits_{\boldsymbol{h}\in\mathcal{H}} & \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathcal{L}^{\text{def}}(\boldsymbol{h}(\boldsymbol{\psi}(\boldsymbol{x})), y)\right], \text{ for a given adversary } \boldsymbol{\psi}. \end{cases} \tag{6.2}$$

From the game theoretical standpoint, we are now studying an asymmetrical and non-zero-sum game, which leads to very interesting and much more sophisticated designs. Wondering whether the model is ill-posed naturally leads to the following question.

> If we change the mathematical framework, would existing findings and insights still hold – at least to some extent?

As we saw in Chapter 3, a simple regularization can fundamentally change the nature of the game. Therefore, most of the previous conclusions may not be valid when we change the whole game design. In any case, studying new frameworks would allow us to assess whether existing conclusions are based solely on the over-strength of the adversary or whether they encompass some form of generality. In both cases, since the modifications will be aimed at limiting the adversary's strength, it would allow us to better understand the actual defense capabilities that we may have when faced with adversarial examples.

6.3 Open problem 2: Rethinking learning theory

Recall that most of the literature on learning theory focuses on demonstrating the convergence of the empirical risk to the theoretical risk using the uniform law of large numbers and some capacity control over the complexity of the hypothesis class. The goal is then to find classes of models that are large enough for the ERM to have a small value, and small enough for us to get a good generalization gap. According to this interpretation – see Figure 6.1 on the left – the common way of thinking is that models with zero training error over-fit samples, leading to poor test time accuracy. However, recent works have questioned the application of this point of view to modern machine learning models such as neural networks. For example, Zangh *et al.* [179] have shown that we can learn a deep network for image classification on CIFAR - 10 which has a training accuracy of 1.0 and still gives more than 0.85 test accuracy. This means that the model may either not over-fit significantly or even not over-fit at all. Returning to the classic form of a bound in learning theory, when the training error is zero, we get

$$\mathcal{R}^{\text{opt}} \leq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}(\boldsymbol{h}(\boldsymbol{x}), y)] \leq O\left(\sqrt{\frac{\mathfrak{C}(n)}{n}}\right), \tag{6.3}$$

where $\mathcal{R}^{\text{opt}}$ is the risk of the Bayes optimal classifier, and $\mathfrak{C}(n)$ is a measure of complexity for the hypothesis class that may or may not depend on n. When $\mathcal{R}^{\text{opt}} = 0$ we can often show that $\sqrt{\frac{\mathfrak{C}(n)}{n}} \underset{n\to\infty}{\longrightarrow} 0$ which makes sense. But, when $\mathcal{R}^{\text{opt}} > 0$, for the right term to explain the error in

a non-trivial way, we need the hidden constants in $O\left(\sqrt{\frac{\mathfrak{C}(n)}{n}}\right)$ to be optimal – which is never the case for the bounds on neural networks. As a result, the ideas of classical learning theory may not apply to deep learning frameworks, meaning that the analysis should not be based on the uniform law of large numbers or capacity control.

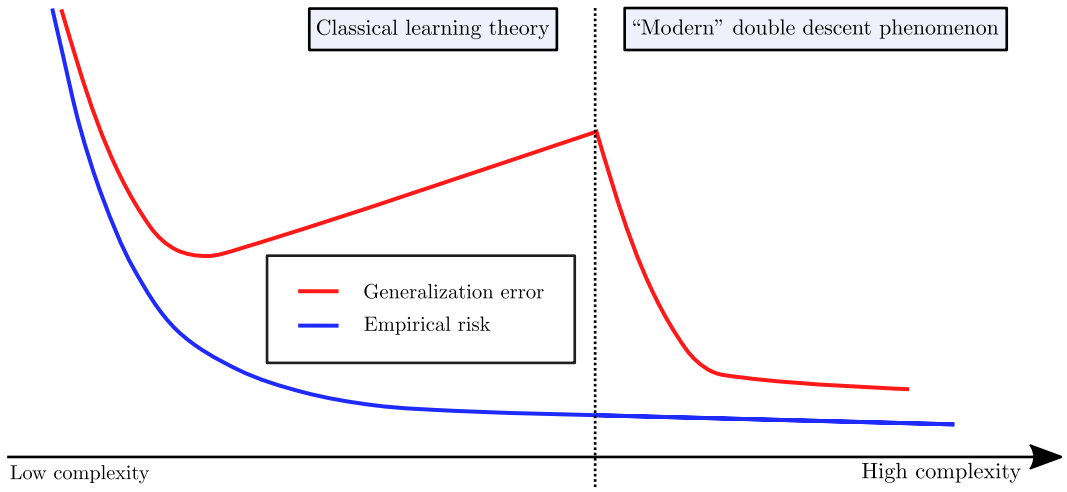


Figure 6.1: Comparison of the classical belief in learning theory with the double descent phenomenon.

Remark 31. *Note that, most of the time, we will have* $\mathcal{R}^{opt} > 0$ *as the support of the conditional distributions* $\{\mu_k\}_{k\in[K]}$ *are not likely to be disjoint.*

We find that the ERM works quite well in practice, which is why researchers have begun to question capacity control. Specifically, the following question arises.

How does the generalization of modern machine learning models depends on their complexity?

A partial answer to this question comes from an observation that is based on the extension of the study beyond the overfitting regime for several machine learning models including neural networks [15, 119]. When we arbitrarily increase the complexity of the model, we find that after overfitting the training samples, the theoretical risk of the model begins to decrease again. This phenomenon is called "double descent". Note that after overfitting, all models have a zero training error, but the larger the model, the lower the theoretical risk. This is a very surprising phenomenon that has been observed on many models of advanced neural networks. From a theoretical point of view, the phenomenon has been identified and analyzed for linear models which indicates that the theory behind neural networks should also be revisited. Coming back to the main objective of this manuscript, we could also ask the following question.

Would this paradigm shift allow us to better understand or avoid adversarial examples?

An interesting way to start answering this question is to examine the k-nearest neighbor model, as suggested by Belkin *et al.* [14]. This is a classical smoothing technique for which we can directly relate the expected loss of the algorithm to the Bayes optimal classifier. For example, Cover and Hard [36] have shown that we can bound the model error as follows.

$$\mathcal{R}^{\text{opt}} \leq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}(\boldsymbol{h}(\boldsymbol{x}), y)] \leq \mathcal{R}^{\text{opt}}\left(2 - \frac{K\mathcal{R}^{\text{opt}}}{K-1}\right). \tag{6.4}$$

Since the guarantees of this technique depend neither on the complexity of the model nor on the uniform law of large numbers, it is a good starting point for the study of a new formalism. Regarding adversarial examples, it should be noted that some classification methods that do not over-fit, such as the k nearest neighbor model, have proven robust against some forms of adversarial examples [164]. Nevertheless, recent results [14] have also shown that if we force a k-nearest neighbor model to over-fit the training samples – *a.k.a.* by interpolating nearest neighbor in [14], then adversarial examples become unavoidable, as it seems to be the case for neural networks. This suggests that the adversarial example phenomenon is closely related to the double-descent regime of deep neural networks and thus to over-fitting.

Remark 32. *As we already discussed in Chapter 2 Goodfellow* et al. *[67] disproved the hypothesis of the model over-fitting the dataset by presenting transferable attacks. But here by over-fitting we mean a much more profound and convoluted phenomenon that occurs when the complexity of the model goes to infinity.*

6.4 Open problem 3: Unifying trustworthy machine learning

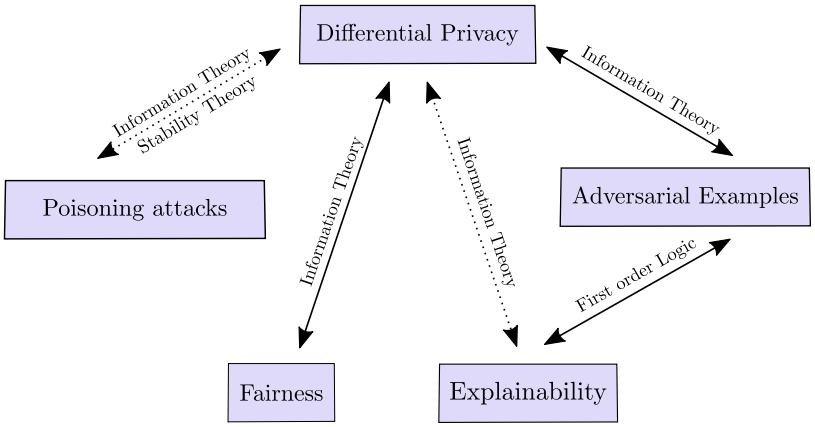


Figure 6.2: Summary of the links and expected links between several areas within the trustworthy machine learning community.

In this manuscript, we have highlighted the theoretical links between differential privacy and robustness to adversarial examples using information theory. But differential privacy also has a formal framework that can be linked to other areas such as fairness in machine learning, as pointed out by Dwork *et al.* [51]. Furthermore, robustness to adversarial examples has been linked to explainability in machine learning by Ignatiev *et al.* [82] using a formal logic viewpoint. Hence, by transitivity, explainability and differential privacy are related notions. It would be interesting to study whether we could establish a direct line between these two concepts using information theory. Differential privacy seems to be a key concept that the community should continue using to build bridges with other areas of trustworthy machine learning.

In particular, the robustness to train time adversaries – *a.k.a.* poisoning attacks [89, 90] – could also be connected to differential privacy since both notions are based on stability theory [11]. Figure 6.2 summarizes the links between the areas we have just discussed – plain lines – and the links that we believe could be useful to the community – dotted lines. We strongly believe that differential privacy and information theory can play a key role in building a unified view on the different domains of trustworthy machine learning.

A Defending against multiple ℓ_p adversarial attacks simultaneously

This appendix is the result of work carried out in collaboration with Alexandre Araujo, Laurent Meunier and Benjamin Negrevergne. It was recently published as a workshop paper at ECML 2020 under the name "Advocating for Multiple Defense Strategies against Adversarial Examples". We also refer the interested reader to the arXiv version of the paper called "Robust Neural Networks using Randomized Adversarial Training".

Contents

As we have discussed several times in this manuscript, the construction of a defense mechanism against an ℓ_p adversary does not guarantee protection against any other type of attack. Furthermore, no unified framework allows to simultaneous protect against multiple threats yet. In this appendix, we refine the geometric analysis presented in Chapter 2 and explain why it is difficult to deal with several threats simultaneously. We also provide a number of empirical insights to illustrate this phenomenon in practice. We then review some of the existing defense mechanisms that attempt to defend against multiple attacks by mixing defense strategies. The rest of this appendix is organized as follows. In Section A.1, we conduct a theoretical analysis to show why ℓ_∞ defense mechanisms cannot be robust against ℓ_2 attacks and vice versa. We then corroborate this analysis in Section A.2 with empirical results using real adversarial attacks and defense mechanisms. In Section A.3, we discuss some recent related works that try to build defenses against multiple adversarial attacks.

A.1 No free lunch for adversarial defenses – a theoretical approach

In this section, we show both theoretically and empirically that defense mechanisms that protect against ℓ_∞ attacks cannot provide adequate protection against ℓ_2 attacks. Our reasoning is perfectly general, so we can demonstrate the reciprocal assertion in the same way, but we focus on this side for the sake of simplicity.

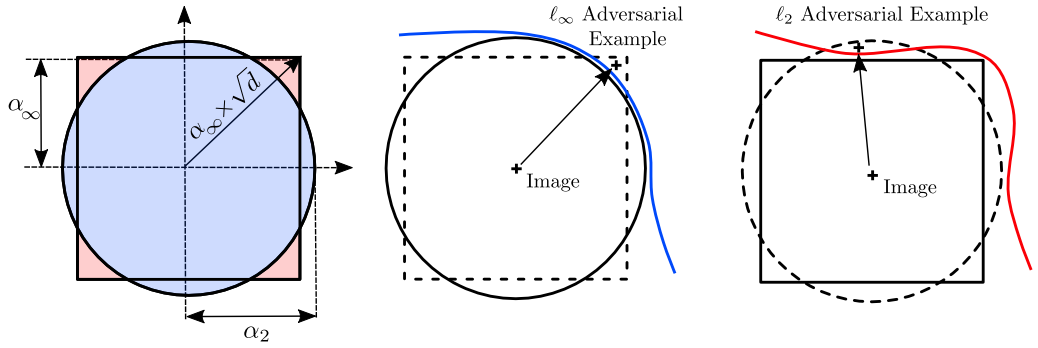


Figure A.1: On the left: 2D representation of the ℓ_∞ and ℓ_2 balls of respective radius α_∞ and α_2. In the middle: a classifier trained with ℓ_∞ adversarial perturbations – red line – remains vulnerable to ℓ_2 attacks. On the right: a classifier trained with ℓ_2 adversarial perturbations – blue line – remains vulnerable to ℓ_∞ attacks.

Let us consider a classifier c_∞ that is robust against adversarial examples with α_∞ bounded perturbation for the ℓ_∞ norm. It guarantees that for any input-output pair $(\boldsymbol{x}, y) \sim \mathcal{D}$ such that $c(\boldsymbol{x}) = y$ and for any α_∞ bounded perturbation $\boldsymbol{\tau}$ we have $c_\infty(\boldsymbol{x} + \boldsymbol{\tau}) = c_\infty(\boldsymbol{x})$. We now wonder what are the performances of this classifier against ℓ_2 adversaries with maximal perturbation α_2. Figure A.1 – on the left, shows the balls with the respective radii α_2 and α_∞. If we build an ℓ_2 adversarial example which is at the intersection of the two balls, it will not work on c_∞. On the other hand, if we manage to select an example outside the ℓ_∞ ball, c_∞ has no more guarantee.

Thus, to characterize the probability that an ℓ_2 perturbation fools an c_∞ in the general case – that is, for any dimension d – we measure the ratio between the volume of the intersection of the ball ℓ_∞ of radius α_∞ and the ball ℓ_2 of radius α_2. As shown in the Theorem 17, this ratio depends on the dimension of the problem d and quickly converges to zero when d increases. It is therefore unlikely that a defense mechanism that protects against ℓ_∞ adversaries will be effective against ℓ_2 attacks.

Theorem 17. *Let us consider $d \in \mathbb{N}$, $\boldsymbol{x} \in \mathbb{R}^d$ and $B_{d|2}(\alpha_2)$ – resp. $B_{d|\infty}(\alpha_\infty)$ – the ℓ_2 ball with radius α_2 – resp. the ℓ_∞ ball with radius α_∞ – with center $\boldsymbol{x}$. If for all d, we select α_2 and α_∞ such that $\mathrm{Vol}\big(B_{d|2}(\alpha_2)\big) = \mathrm{Vol}\big(B_{d|\infty}(\alpha_\infty)\big)$. Then the following holds,*

$$\frac{\mathrm{Vol}\big(B_{d|2}(\alpha_2) \cap B_{d|\infty}(\alpha_\infty)\big)}{\mathrm{Vol}\big(B_{d|\infty}(\alpha_\infty)\big)} \underset{d\to\infty}{\longrightarrow} 0.$$

Proof. Without loss of generality, let us fix $\alpha_\infty = 1$ and $\boldsymbol{x} = \mathbf{0}$. Then we have

$$\mathrm{Vol}\big(B_{d|2}(\alpha_2)\big) = \frac{\big(2\Gamma\big(\frac{1}{2}+1\big)\alpha_2\big)^d}{\Gamma\big(\frac{d}{2}+1\big)} \text{ and } \mathrm{Vol}\big(B_{d|\infty}(1)\big) = 2^d.$$

Then, to have balls with the same volumes, for any d we set

$$\alpha_2 = \frac{2}{\sqrt{\pi}} \Gamma\left(\frac{d}{2} + 1\right)^{1/d}$$

where Γ is the gamma function. Since α_2 is a function of d, in the remaining we use the notation $\alpha_2(d)$. If we denote $\mathcal{U}$, the uniform distribution on $B_{d|\infty}(1)$ we get

$$\begin{aligned}
&\frac{\mathrm{Vol}(B_{d|2}(r_2(d)) \cap B_{d|\infty}(1))}{\mathrm{Vol}(B_{d|\infty}(1))} \\
&= \mathbb{P}_{\boldsymbol{z}\sim\mathcal{U}}\left[\boldsymbol{z} \in B_{d|2}(r_2(d))\right] = \mathbb{P}_{\boldsymbol{z}\sim\mathcal{U}}\left[\sum_{i=1}^{d} \boldsymbol{z}_i^2 \leq \alpha_2(d)^2\right] \\
&= \mathbb{P}_{\boldsymbol{z}\sim\mathcal{U}}\left[\sum_{i=1}^{d} \boldsymbol{z}_i^2 - \mathbb{E}_{\boldsymbol{z}\sim\mathcal{U}}\left[\sum_{i=1}^{d} \boldsymbol{z}_i^2\right] \leq \alpha_2(d)^2 - \mathbb{E}_{\boldsymbol{z}\sim\mathcal{U}}\left[\sum_{i=1}^{d} \boldsymbol{z}_i^2\right]\right].
\end{aligned}$$

Furthermore, when d is sufficiently large we get

$$\alpha_2(d)^2 - \mathbb{E}_{\boldsymbol{z}\sim\mathcal{U}}\left[\sum_{i=1}^{d} \boldsymbol{z}_i^2\right] = \alpha_2(d)^2 - \frac{d}{3} < 0.$$

Using the Hoeffding inequality we get

$$\frac{\mathrm{Vol}(B_{d|2}(r_2(d)) \cap B_{d|\infty}(1))}{\mathrm{Vol}(B_{d|\infty}(1))} \leq \exp\left(-\frac{\left(\alpha_2(d)^2 - \frac{d}{3}\right)^2}{d}\right).$$

Finally, thanks to Stirling's formula, we have

$$\alpha_2(d) \underset{d\to\infty}{\sim} \sqrt{\frac{2}{\pi e}} d^{1/2}.$$

Then the right hand term converges towards 0 when d goes to ∞ which concludes the proof. □

Theorem 17 indicates that, when d is large enough, ℓ_2 based perturbations have a null probability of being also in the ℓ_∞ ball of the same volume. Therefore, when the dimension of the problem is sufficiently large, a defense mechanism offering complete protection against ℓ_∞ adversaries is not guaranteed to offer any protection against ℓ_2 attacks[1]. This result goes against two-dimensional intuition. Indeed, if we consider a two-dimensional problem, the ℓ_∞ and ℓ_2 balls overlap significantly – as shown on the left of figure A.1 – and the probability of sampling at the intersection of the two balls is about 0.98. However, this probability is close to zero for any realistic image setting, even for very simple image data sets such as MNIST [97].

[1] Theorem 17 can easily be extended to any two balls with different norms. But we restrict to the case of ℓ_∞ and ℓ_2 norms as we mainly discussed these norms until now.

Table A.2: Comparison of the bound from Theorem 17 when d varies from $d = 2$ to typical image classification setting – $10^{-0.009} \approx 0.98$.

Dataset	Dimension (d)	Volume of the intersection
–	2	$10^{-0.009}$
MNIST	784	10^{-144}
CIFAR	3072	10^{-578}
ImageNet	150528	10^{-28946}

A.2 No free lunch for adversarial defenses in practice

Our theoretical analysis shows that if adversarial examples were uniformly distributed in a high dimensional space, then any mechanism that only defends perfectly against the ℓ_∞ adversaries has a zero probability to be robust to ℓ_2 adversarial examples. Although existing defense mechanisms do not necessarily assume such a distribution, targeted defenses only work marginally when attacked with different norms. Before analyzing the results, let us briefly present some details of the experimental protocol.

Experimental protocol

All experiments are conducted on CIFAR-10 with the Wide-Resnet 28-10 architecture. We use the training procedure and the hyper-parameters described in the original paper by [177]. To train a neural network with adversarial training, we still use the same hyper-parameters, and generate adversarial examples during training using either PGD-ℓ_∞ or PGD-ℓ_2 adversary with 10 iterations.

To compare the empirical performances of our method with adversarial training, we consider two ℓ_p adversaries with thresholds corresponding to CIFAR-10.

- *An ℓ_∞ adversary with perturbation bounded by* 0.031. To model this adversary we use the PGD attack with $t_{max} = 20$ iterations.
- *An ℓ_2 adversary with perturbation bounded by* 0.8. To model this adversary we use the PGD attack with $t_{max} = 20$ iterations.
- *Another ℓ_2 adversary with perturbation bounded by* 0.8. To model this adversary we use the C&W attack with 60 iterations, a learning rate equal to 0.01, 9 binary search steps, and an initial constant of $\kappa = 0.001$.

Remark 33. *Our analysis is mainly focusing on PGD attacks for both the ℓ_∞ and the ℓ_2 norms. However, these attacks have a very strict geometry*[2]*. This is why, to present a deeper analysis of the behavior of adversarial attacks and defenses, we also present a set of experiments that use C&W attack.*

[2] Due to the projection operator, all PGD attacks saturate the constraint, which makes them all lie in a very small part of the ball.

Results

Table A.3: Average norms of PGD-ℓ_2 and PGD-ℓ_∞ adversarial examples with and without ℓ_∞ adversarial training on CIFAR-10 ($d = 3072$).

Adversary	**Attack PGD-ℓ_2**		**Attack PGD-ℓ_∞**	
Model	Unprotected	AT-ℓ_∞	Unprotected	AT-ℓ_2
Average ℓ_2 norm	0.830	0.830	1.400	1.640
Average ℓ_∞ norm	0.075	0.200	0.031	0.031

To demonstrate that adversarial training is not robust against PGD-ℓ_2 attacks, we measure the evolution of the ℓ_2 norm of the adversarial examples generated by the attack against an unprotected model and a model trained with adversarial training ℓ_∞ – AT-ℓ_∞, where the adversarial examples are generated with the PGD-ℓ_∞. The results are presented in Table A.3. We can see that the average ℓ_∞ norm of an ℓ_2 perturbation more than doubles between an unprotected model and a model trained with an adversarial training. As shown in Figure A.1 – on the right – the attack generates an adversarial example in the cap of the ℓ_2 ball, thus increasing the ℓ_∞ norm while maintaining the same ℓ_2 norm. The same phenomenon can be observed with the AT-ℓ_2 against the PGD-ℓ_∞ attack – see Figure A.1 in the middle and Table A.3. The PGD-ℓ_∞ attack increases the ℓ_2 norm while maintaining the same ℓ_∞ perturbation by generating the perturbation in the corner of the ℓ_∞ ball. As a result, we cannot expect adversarial training ℓ_∞ to guaranty any protection against the ℓ_2 adversarial examples.

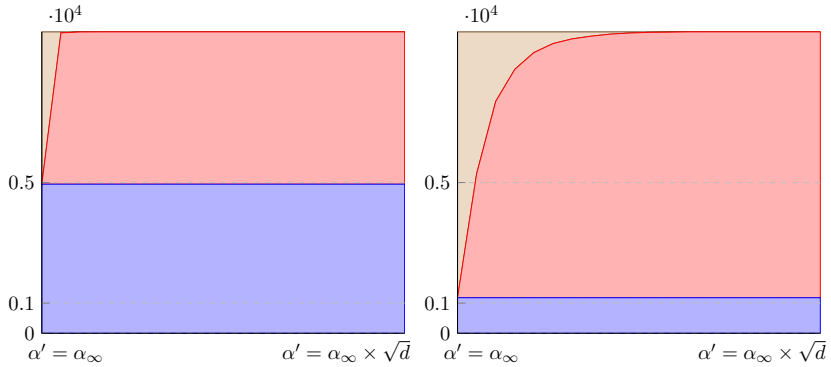


Figure A.4: Comparison of the number of adversarial examples found by C&W, inside the ℓ_∞ ball – lower, blue area, outside the ℓ_∞ ball but inside the ℓ_2 ball – middle, red area – and outside the ℓ_2 ball –upper beige area. α_∞ is set to 0.03 and α' varies along the x-axis. On the left: without adversarial training. On the right: with adversarial training.

To better capture the behavior of ℓ_2 adversarial examples, we now study the performance of an ℓ_2 C&W attack with and without AT-ℓ_∞. It allows us to understand in which area C&W discovers adversarial examples and the impact of AT-ℓ_∞. In high dimensions, the red corners – see

Figure A.1 left – are very far away from the ℓ_2 ball. Therefore, we hypothesize that a large proportion of the ℓ_2 adversarial examples will remain unprotected. To validate this assumption, we measure the proportion of adversarial examples inside the ℓ_2 ball before and after ℓ_∞ adversarial training. The results are presented in Figure A.4 – on the left: without adversarial training, right: with adversarial training. The blue area represents the proportion of adversarial examples that are inside the ℓ_∞ ball. The red zone represents the adversarial examples that are outside the ℓ_∞ ball but still inside the ℓ_2 ball, *i.e.* valid adversarial examples for an ℓ_2 adversary but not for ℓ_∞ adversary. Finally, the brown-beige zone represents the adversarial examples that are beyond the ℓ_2 limit[3]. α' is the size of the ℓ_2 ball and varies along the x-axis from α_∞ to $\alpha_\infty\sqrt{d}$. On the left – without adversarial training - most of the adversarial examples generated by C&W are inside the two balls. On the right, most of the adversarial examples have been moved outside the ℓ_∞ ball. This is the expected consequence of ℓ_∞ adversarial training. However, these adversarial examples remain in the ℓ_2 ball — *i.e.* they are in the upper cap of the ℓ_2 ball described on the right of Figure A.1. This means that even after adversarial training, it is still easy to find good ℓ_2 adversarial examples, which makes the robustness of AT-ℓ_∞ almost non-existent.

A.3 Related works and perspective to defend against multiple perturbations

Defenses against adversarial examples, such as adversarial training, are usually tailored to a single type of perturbation. As we have just discussed, defending against a single perturbation offers no guarantee and can even sometimes increase the vulnerability of the model to new attacks. Hence the community is trying to develop effective defenses against multiple perturbations. This question has been studied in several recent works [104, 153].

Tramer *et al.* [153] first proposed to address this problem by mixing adversarial training with attacks for different norms in order to defend against multiple models of perturbation. To design the loss function, the authors proposed two simple aggregation rules. The first averages the attacks and the second selects the perturbation that maximizes the loss over the different threat models. While this approach can achieve varying degrees of robustness for the adversarial perturbation models considered, in practice it is quite difficult to adjust and often achieves varying degrees of robustness to individual perturbations. This results in a sub-optimal worst-case loss when we consider the union of threat models – which is actually our main purpose. To address this shortcoming, Maini [104] has further refined the PGD-based procedure to simultaneously incorporate all model threats into a single attack called MDS – Multi Steepest Descent. This technique achieves about the same level of robustness as adversarial training – against one model threat — when it faces any of the threats it has learned on. Much work remains to be done on defense against multi-model threats, which opens interesting theoretical and practical perspectives. In particular, it would be interesting to study cases where we could design provable defenses against multiple perturbations without compromising the accuracy of the model.

[3] As mentioned in Chapter 2, the C&W attack is not bounded at first since the Lagragian relaxation does not have a strong constraint. Hence, the beige zone represents the set of examples that will be clipped at the end of the procedure.

B Unsupervised learning under differential privacy constraints

This appendix is based on the individual and joint works of Anne Morvan and Rafael Pinot. We refer the interested reader to the following manuscripts for an in-depth analysis of all the concepts and contributions associated with this appendix.

- "Graph-based Clustering under Differential Privacy".
  Uncertainty in Artificial Intelligence (UAI) 2018.
  R. Pinot, A. Morvan, F. Yger, C. Gouy-Pailler, J. Atif.
- "Contributions to unsupervised learning from massive high-dimensional data streams".
  PhD thesis PSL University 2018.
  A. Morvan.
- "Minimum spanning tree release under differential privacy constraints".
  Master thesis Sorbonne University 2017.
  R. Pinot.

Finally, to access implementation details, one can refer to the following Github repositories.

`https://github.com/RPINOT/privateMST` & `https://github.com/annemorvan/DBMSTClu`

Contents

In this appendix, we present an overview of another line of research we investigated in this thesis, namely unsupervised learning under differential privacy constraints. More precisely, we developed a differentially private clustering algorithm for arbitrarily-shaped clustering of nodes in a graph. The results we present here are somewhat orthogonal to the main object of the manuscript, but they represent advances in the field of differential privacy which is closely related to robustness as we discussed in Chapter 5. In Section B.1, we first present the key concepts of node clustering

in a graph and summarize our findings in this domain. We then introduce clustering based on differentially private graphs and present our main contributions in this area in Section B.2. Finally, Section B.3 presents some numerical results.

Reading note. *We only provide a very light introduction to our contributions and we deliberately skip a lot of technical details and state informal results for simplicity. The interested reader will find above references and source code for a more detailed reading.*

B.1 Graph clustering and minimum spanning tree

Notations. *Let $\mathcal{G} = (V, E, w)$ be a simple undirected weighted graph with a vertex set V, an edge set E, and a weight function $w := E \to \mathbb{R}$. We call $G = (V, E)$ the topology of the graph, and $\mathcal{W}_E$ denotes the set of all possible weight functions mapping E to weights in $\mathbb{R}$. Cursive letters are used to represent weighted graphs and straight letters refer to topological arguments. Since graphs are simple, the path $\mathcal{P}_{u-v}$ between two vertices u and v is characterized by the ordered sequence of vertices $\{u, \dots, v\}$ or the corresponding binding edges depending on the context.*

Weighted graphs are known to be a useful representation of data in many areas of computer science, such as bioinformatics or network analysis – be it social, computer or information networks. More generally, a graph can always be thought of as a representation of data dissimilarity where the points in the dataset are the vertices and the weighted edges express the distances between these objects. In both cases, graph clustering [140] is a key tool for understanding the underlying structure of the datasets by locating groups of nodes ruled by a specific similarity. Furthermore, the minimum spanning tree is known to help recognizing clusters with arbitrary shapes in tree-based clustering algorithms. It thus can be used for a wide range of applications [5, 72, 116, 172, 178].

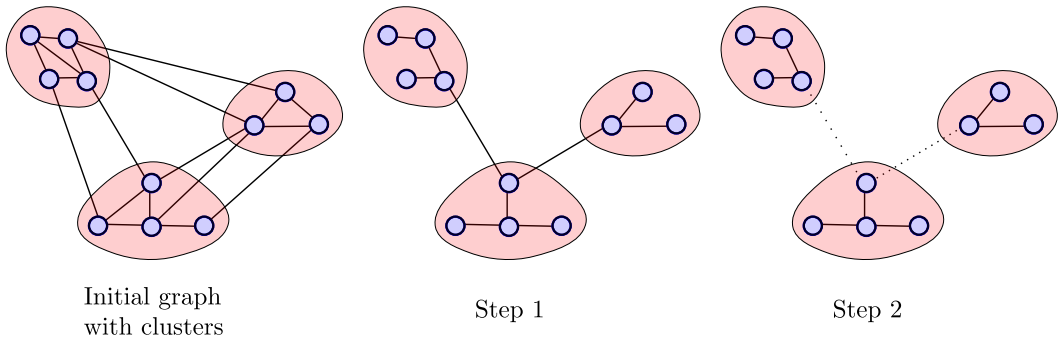


Figure B.1: Summary of the generic procedure for computing an MST-based clustering.

Let us consider $\mathcal{G} = (V, E, w)$ a simple undirected weighted graph with a vertex set V, an edge set E, and a weight function $w := E \to \mathbb{R}$. From $\mathcal{G}$, an MST-based algorithm proceeds as follows – Figure B.1 summarizes the procedure.

1. Compute a connected subset of E that spans V with minimal cumulative weight, *i.e.* a minimum spanning tree of $\mathcal{G}$ – Figure B.1 middle.

2. Prune the tree according to some criteria in order to obtain a forest. Every connected component in this forest characterizes a cluster – Figure B.1 right.

Remark 34. *Note that we will always assume here that the graph is connected, but similar results hold when we have more than one connected component.*

MST-based clustering methods, however effective, generally lack appropriate formal analysis. Our first contribution fills this gap by providing a theoretical framework for MST-based clustering. More precisely, our contribution is twofold: 1) we present a theoretical framework motivating MST-based clustering methods, where the notion of clustering is based on the concept of minimum path distance, 2) we provide theoretical guarantees for DBMSTClu [116] algorithm, an MST-based clustering algorithm we previously worked on.

Analytical framework for MST-based clustering

In order to analyze the effectiveness of an MST-based method, we must first introduce a notion of clustering that we want to comply with. Since the graph is simple, it is possible to define the minimum path distance between two nodes, which makes our definition of clustering more explicit.

Definition 13 (Minimum path distance)**.** *Let us consider $\mathcal{G} = (V, E, w)$ and $u, v \in V$. The minimum path distance between u and v is*

$$\mathrm{dist}(u, v) = \min_{\mathcal{P}_{u-v}} \sum_{e \in \mathcal{P}_{u-v}} w(e),$$

with $\mathcal{P}_{u-v}$ a path from u to v in $\mathcal{G}$ – edges version.

Definition 14 (Cluster extended from [172])**.** *Let us consider $\mathcal{G} = (V, E, w)$,* dist *the minimum path distance defined on $\mathcal{G}$ and $D \subset V$. A vertices set $C \subset D$ is a cluster if and only if $|C| > 2$ and for any partition C_1, C_2 of C we have*

$$\arg\min_{z \in D \setminus C_1} \{ \min_{v \in C_1} \mathrm{dist}(z, v) \} \subset C_2.$$

Remark 35. *Assuming that a cluster is built of at least 3 points makes sense since singletons or groups of 2 nodes can be legitimately considered as noise. For simplicity of the proofs, the following theorems hold in the case where noise is neglected. However, they are still valid in the setting where noise is considered as singletons – with each singleton representing a generalized notion of cluster.*

In particular, this definition states that a cluster C can only be defined if for any vertex $u \in C$, the closest vertex to u in $\mathcal{G}$ must be in C. Figure B.2 illustrates the definition. On the left, we have a valid clustering. On the right, the cluster is obviously non representative of the underlying structure of the graph; hence Definition 14 does not hold. Thanks to the above definition of cluster, we can now motivate the use of MST-based algorithms for node clustering in a graph.

Theorem 18 (Motivation for MST-based clustering - Informal)**.** *Let $\mathcal{G} = (V, E, w)$ be a graph and $\mathcal{T}$ a minimum spanning tree of $\mathcal{G}$. Let also C be a cluster in the sense of Definition 14. Then for any two vertices $u, v \in C$, we have $\mathcal{P}_{u-v} \subset C$, where $\mathcal{P}_{u-v}$ is the path from u to v in $\mathcal{T}$.*

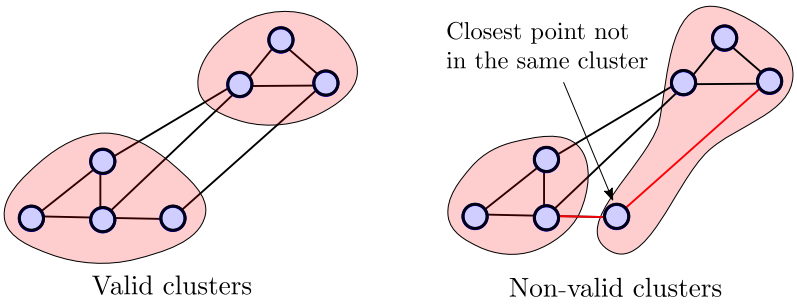


Figure B.2: Illustration of valid and non-valid clusters for Definition 14.

This theorem states that for any two nodes in C, every vertex in the path between them is in C. This means that a cluster can be fully characterized by a subtree of $\mathcal{T}$. It justifies the use of MST-based methods for clustering data or nodes in a graph. All clustering algorithms based on successively cutting the edges of an MST to obtain a forest are sound in the sense of the Theorem 18. In what follows, we focus our analysis on one of the latest MST-based clustering algorithms called DBMSTClu and recently introduced by Morvan *et al.* [116].

MST-based clustering with guarrantees

Let us briefly present this algorithm – refer to Figure B.3 for a basic description. Given an MST $\mathcal{T}$, DBMSTClu consists of successive cuts on $\mathcal{T}$. At each iteration, an edge is cut if a certain criterion, called *Density-Based Validity Index of a Clustering partition* – DBCVI – is improved. This edge is greedily chosen to locally maximize the DBCVI at each step. When no improvement of the DBCVI can be made, the algorithm stops. The notion of DBCVI is constructive. In short, it compares the maximum edge weight inside the cluster with the minimum weight of the edges coming out of the cluster. When this difference is high; the DBCVI is also high and vice versa. For more details on this notion, we refer the reader to [116] and [117].

Then – under mild assumption on the weight function w – we can guarantee that DBMSTClu outputs a set of clusters that comply with Definition 14. In a nutshell, this condition says that the edge separating each cluster must have a weight that is sufficiently distinct from the edges within the cluster. This motivates the following definition.

Definition 15 (Homogeneous separability condition)**.** *Let us consider a graph $\mathcal{G} = (V, E, w)$, $s \in E$ and $\mathcal{T}$ a tree of $\mathcal{G}$ with a set of edges $E(\mathcal{T})$. $\mathcal{T}$ is said to be homogeneously separable by s if the following holds,*

$$\beta_{\mathcal{T}} \max_{e \in E(\mathcal{T})} w(e) < w(s), \text{ with } \beta_{\mathcal{T}} = \frac{\max\limits_{e \in E(\mathcal{T})} w(e)}{\min\limits_{e \in E(\mathcal{T})} w(e)} \geq 1.$$

The latter condition is local and depends on a tree in $\mathcal{G}$. Let us suppose that there exists K clusters in $\mathcal{G}$ – $(C_1, \ldots, C_K)$ – characterized by subtrees $(\mathcal{T}_1, \ldots, \mathcal{T}_K)$. Then if the subtrees are homogeneously separable by a set of edges $(s_1, ..., s_{K-1})$, then the graph is called homogeneous.

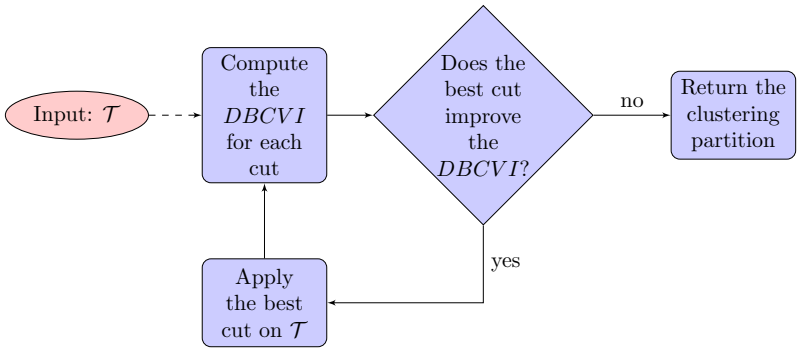


Figure B.3: Diagram summarizing DBMSTCLU algorithm. Figure from [117].

This is simply a condition for clusters to be sufficiently separated one from another in $\mathcal{G}$. When the graph is homogeneous, we can show that DBMSTCLU recovers correctly the K clusters.

Theorem 19 (Efficacy of DBMSTCLU – Informal)**.** *Let $\mathcal{G}$ be a simple weighted graph with K clusters $C_1, \ldots, C_K$. If $\mathcal{G}$ is homogeneous, then DBMSTCLU applied on any MST of $\mathcal{G}$ outputs subtrees that match the clusters $C_1, \ldots, C_K$.*

We now have an algorithm in an appropriate analytical framework, which allows us to evaluate when it will correctly find the underlying structure of the graph. Nevertheless, it is essential that the data representation we use protects the private characteristics contained in the graph. Let us consider an application in which we want to identify groups of web pages that have similar content, *i.e.* web pages with a similar audience. In this case, the vertices represent the web sites. The link between two vertices represents the fact that some users visit both of them. The edge weights reflect the number of common users and therefore carry sensitive information about individuals. When analyzing the graph data, no personal user's browsing behavior should be violated; *i.e.* browsing from one page to another should remain private. As already mentioned in the body of the manuscript, the gold standard for privacy data analysis is differential privacy [50]. In the following, we discuss how differential privacy can be applied to graph based clustering algorithms.

B.2 Differentially private node clustering in a graph

Even though differential privacy has been extensively investigated, learning from graph databases under differential privacy remains challenging. Mir *et al.* [108] as well as Karwa *et al.* [86] formalized the idea of releasing statistics from a graph in a differentially private manner following the seminal work of Nissim *et al.* [122]. Then several definitions of differential privacy on graphs appeared. Among them, the main ones are edge-differential privacy [75], and node-differential privacy [88]. Conceived for the protection of the graph topology, these definitions are not suitable for applications to network analysis where the structure is static and the private information on the

users are carried by the edge weights. Sealfon [143] addressed this issue by providing a new formal framework for the private analysis of weighted graphs where the graph topology $\mathcal{G} = (V, E)$ is public and the private information is contained in the weight function $w : E \to \mathbb{R}$. We recently studied a slightly different version of the definition that relies on the following notion of closeness between weight functions.

Definition 16. *For any edge set E, two weight functions w, w' are called neighbors – denoted $w \sim w'$ – if $||w - w'||_\infty := \max_{e \in E} |w(e) - w'(e)| \leq \alpha$.*

α represents the sensitivity of the weight function and should be chosen according to the application at hand. The definition of weight-differential privacy for a graph algorithm can then write as follows.

Definition 17. *For any graph topology $G = (V, E)$, let $\boldsymbol{A}$ be a randomized algorithm that takes as input a weight function w. $\boldsymbol{A}$ is called ϵ-differentially private on $G = (V, E)$ if for all pairs of neighboring weight functions $w \sim w'$, and for any possible output o, one has*

$$\mathbb{P}[\boldsymbol{A}(w) = o] \leq e^{\epsilon} \mathbb{P}\big[\boldsymbol{A}(w') = o\big].$$

When it is ϵ-differentially private on every graph topology in a class $\mathcal{C}$, $\boldsymbol{A}$ is called ϵ-differentially private on $\mathcal{C}$.

In the remaining, we will be interested in this definition of differential privacy that we call weight-differential privacy.

Node clustering in a graph under differential privacy

Differentially private clustering for unstructured datasets has been first discussed in [122]. This work introduced the first method for differentially private clustering based on the k-means algorithm. Since then most of the work in the field focused on adaptation of this method [21, 49, 106]. The main drawback of this work is that it is not able to deal with arbitrary-shaped clusters. This issue has been recently investigated in [80] and [32]. They proposed two new methods to find arbitrary-shaped clusters in unstructured datasets respectively based on density clustering and wavelet decomposition. Even though both of them produce non-convex clusters, they only deal with unstructured datasets and thus are not applicable to node clustering in a graph. Graph clustering has already been investigated in a topology-based privacy framework [118, 121], however, these works do not consider weight-differential privacy. Hereafter, we present a new generic method for node clustering in a graph under this privacy notion using MST-based clustering methods. Recall that an MST-based clustering algorithm 1) builds a minimum spanning tree and 2) prunes it until getting a forest that represents the clusters. To make this procedure private, we choose to compute step 1) under differential privacy constraint. Then thanks to the post-processing inequality– see Theorem 14, we extend the obtained privacy guarantees to step 2). Hence, we first want to design a differentially private minimum spanning tree algorithm. Sealfon [143] addressed this issue by providing and analyzing an algorithm that releases an approximate minimum spanning tree under weight-differential privacy. The error of this seminal algorithm in

terms of weight approximation is $O(|V|\log|E|)$ [1] for fixed privacy parameters and its time complexity is $O(|E|+|V|\log|V|)$. Accordingly, this method can be highly inaccurate when the graph is large, which is common in machine learning applications.

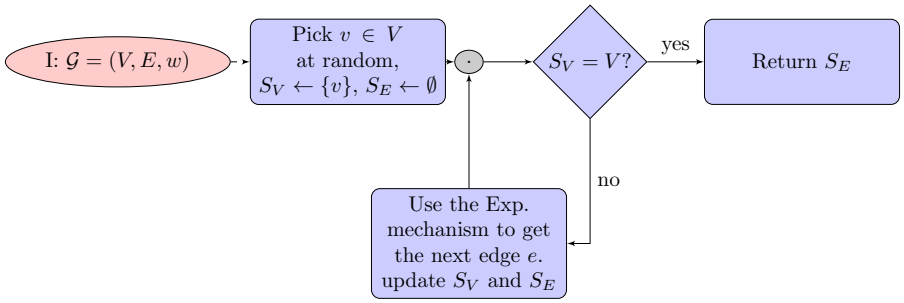


Figure B.4: Diagram summarizing PAMST algorithm. Figure from [117].

A way to improve the algorithm's performances in the context of graph learning under differential privacy is to construct an iterative method which focuses on considering a *local* condition on the weight function in the process. This idea has been first proposed by Gupta *et al.* [73], and extensively investigated in the framework of differentially private submodular optimization by Mitrovic *et al.* in [112]. In the latter, the issue of releasing a minimum spanning tree under weight-differential privacy is not directly investigated. Yet one could derive from the study of monotone submodular maximization a private version of Kruskal algorithm with an improved approximation error of $O\big(|V|^2/|E|\log|V|\big)$. Even though the approximation error is satisfying, Kruskal algorithm in the submodular framework has an algorithmic complexity of $O(|E||V|)$ which is prohibitive when dealing with large and dense graphs.

Under a similar privacy setting we recently produced an algorithm to release the topology of a tree under differential privacy. We designed a Prim-like algorithm[2] – called PAMST – to privately release the topology of an almost minimum spanning tree thanks to an iterative use of the well known scheme from the differential privacy literature called the *Exponential mechanism* [50]. The Exponential mechanism represents a way of privately answering arbitrary range of queries. It is defined according to a utility function which aims at providing some preorder on the possible outputs of the algorithm according to the order in $\mathbb{R}$. In PAMST we use it to select an edge at random from a subset of E. The algorithm is summarized in Figure B.4.

PAMST takes as an input a weighted graph. It outputs the spanning tree which weight is almost minimal, according to the weight function. To do so, the algorithm starts at an arbitrarily chosen vertex and chooses one of its incident edges according to the Exponential mechanism. Then it updates the set of edges S_E and adds the second node belonging to this edge to the current node set S_V. As long as the node set does not contain all the vertices, PAMST continues to choose at each

[1] This error is computed according to the difference between the underlying weight of the tree topology – sum of the edges weights – and the weights of the minimum spanning tree, using the initial weight function.

[2] For more details on Prim algorithm, the interested reader can refer to [130].

step an edge that is incident to S_V to update the edge set, and updates the node set accordingly. Theorem 20 states that using PAMST to get an almost minimal spanning tree topology preserves weight-differential privacy.

Theorem 20 (Privacy for PAMST - Informal)**.** *PAMST is ϵ-differentially private on the set of simple undirected weighted graphs, where ϵ depends on the type of exponential mechanism we use.*

PAMST exhibits a weight approximation error of $O\big(|V|^2/|E|\log|V|\big)$ for fixed privacy parameters, and a time complexity of $O(|V|^2)$. This result, in contrast to previous works, enables to deal with relatively large, and dense graphs, which are frequently met in machine learning applications. For more details on the design of the algorithm and the privacy guarrantees, one can refer to [133]. Thanks to the above and to the data-processing inequality, we get a differentially private clustering algorithm by combining PAMST and DBMSTClu algorithms that we call PTClust.

B.3 Experimental validation

To verify the efficacy of PTClust for varying levels of privacy, we have performed experiments on two classical synthetic graph datasets for clustering with non-convex shapes: two concentric circles and two moons, both in their noisy versions. Before analyzing the results, let us briefly present some details of the experimental protocol.

Experimental protocol

For the readability and visualization purposes, both graphs are embedded into a two dimensional Euclidean space. Each dataset contains 100 vertices represented by a point of two coordinates. Both graphs have been built with respect to the homogeneity condition from Theorem 19. In practice, the complete graph – graph where all vertices are connected – has been trimmed from its irrelevant edges, *i.e.* the edges not respecting the homogeneity condition. Hence, those graphs are not necessarily Euclidean since close nodes in the visual representation may not be connected in the graph. Finally, weights are normalized between 0 and 1, and α is set to 0.1.

Figures B.5 and B.6 show for each dataset (a) the original homogeneous graph $\mathcal{G}$ we built, (b) the clustering partition of DBMSTClu with the underlying MST. We compare this benchmark with the clustering partition for PTClust with different privacy degrees – resp. (c) $\epsilon = 1$, (d) $\epsilon = 0.7$ and (e) $\epsilon = 0.5$[3]. Each experiment is carried out independently and the tree topology obtained by PAMST will be different. This explains why the edge between clusters may not be the same when the experiment is repeated with a different level of privacy. However, this will marginally affect the overall quality of the clustering.

Results

As expected, DBMSTClu – (b) – recovers automatically the right partition and the results are shown here for comparison with PTClust. For PTClust, the true MST is replaced with a private approximate MST obtained using PAMST. When the privacy degree is moderate, *i.e.*

[3] Although we could take any $\epsilon > 0$, it is usually chosen in $(0, 1]$ [50, Chap 1&2].

$\epsilon \in \{1.0, 0.7\}$, it appears that the clustering result is slightly affected. More precisely, in Figures B.5 and B.6 –(c) and (d) – the two main clusters are recovered. Some points however are isolated as singletons. This is due to the randomization involved in determining the edge weights for the topology returned by PAMST. Furthermore, as an expected effect of differential privacy, when ϵ decreases, the clustering quality deteriorates, as DBMSTCLU is sensitive to severe changes in the MST – see Figure B.5 and B.6 (e).

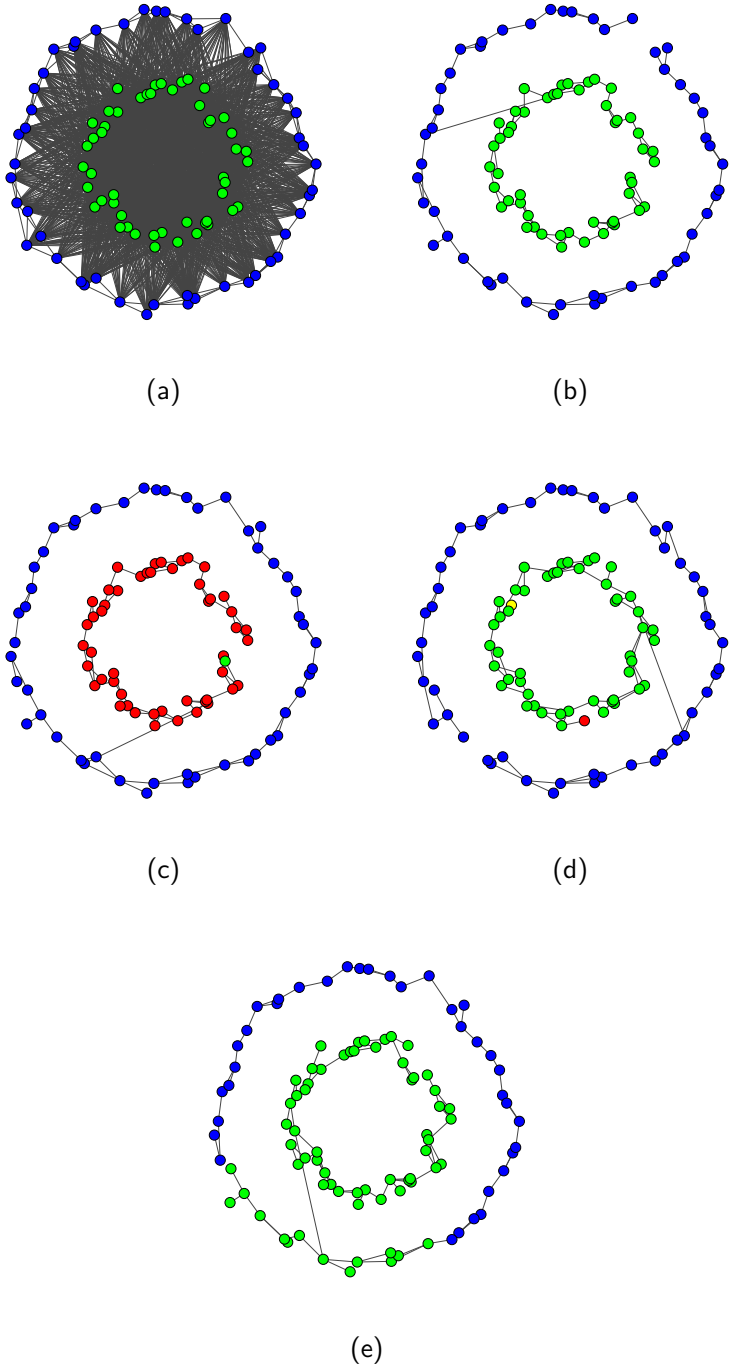


Figure B.5: Differentially private clustering for the Circles dataset of size $n = 100$.

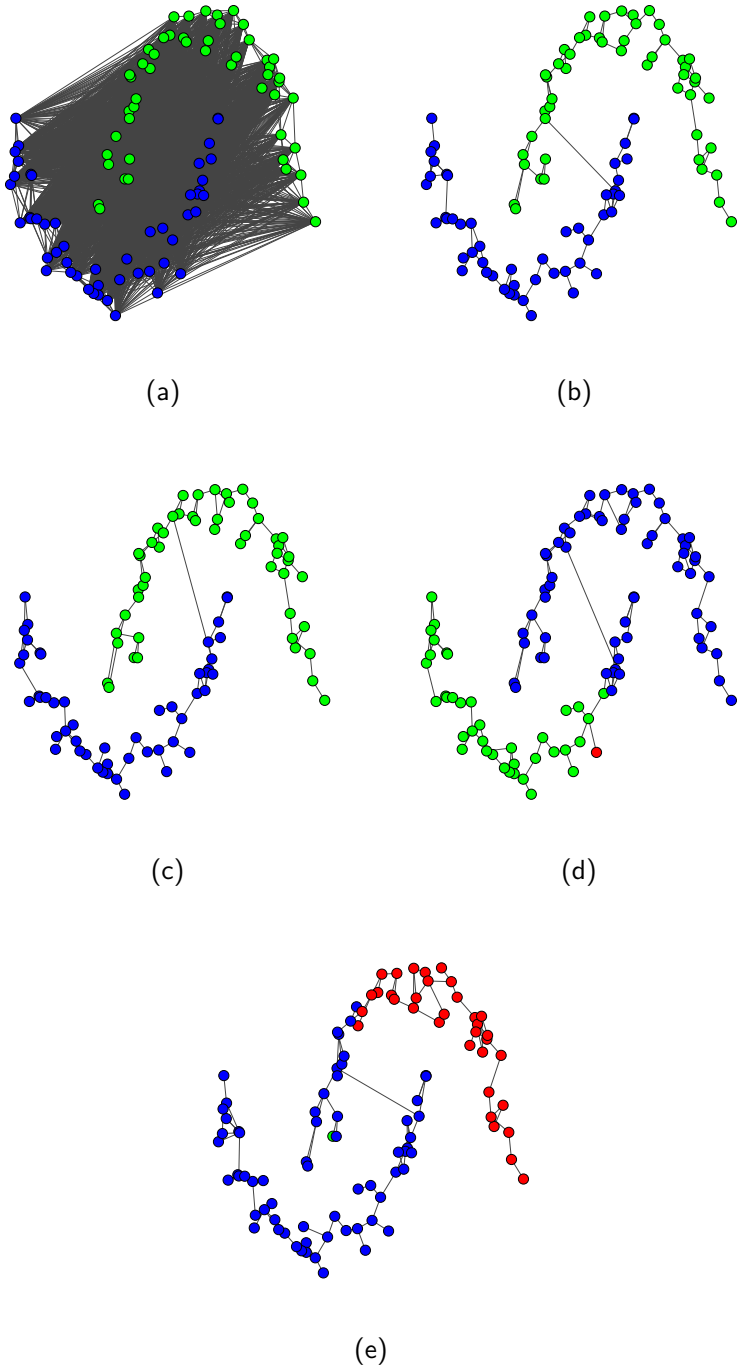


Figure B.6: Differentially private clustering for the Moons dataset of size $n = 100$.

C Secure and private deep learning with encrypted aggregation operator

This appendix is a brief overview of an ongoin work in collaboration with Arnaud Grivet Sebert, Martin Zuber and Renaud Sirdey. We refer the interested reader to an arXiv version of this preparatory work called "SPEED: Secure, PrivatE, and Efficient Deep learning".

Contents

In this appendix, we present some of our most recent work addressing the issue of collaborative deep learning with privacy constraints. In Section C.1 we present our problem setting and the new framework we designed. More precisely we aim to devise a framework that takes into account threat models that are beyond the scope of differential privacy. Then, we present some related works on deep learning with privacy in Section C.2.

C.1 A new framework beyond differential privacy

As we have already discussed, the large adoption of machine learning in several domains, including critical ones, raises a number of concerns on the security and privacy of the tools we develop. For now, we mainly discussed the notion of differential privacy, but here we try to go beyond this definition of privacy preserving machine learning.

Toy Example. *An example of scenario from the field of cybersecurity where we need to consider a more sophisticated threat model is as follows. Several actors hold a database of cybersecurity incident signatures, that have occurred on their customer networks. Each actor can build a malware detection model on its own, but building a model that benefits from a larger set of such signatures would lead to improved detection capabilities. In general, these databases are highly-sensitive and highly-valuable; as such, they cannot be disclosed. In such a setting, the data owners wish to collaboratively transfer the knowledge they have into a global model while preserving the confidentiality of their learning sets.*

In the context of collaborative learning, several recent works [13, 19, 30, 58, 123, 125] focused on using differential privacy to build privacy preserving deep learning models. However, these tech-

niques rely on a "trusted" aggregation server that gathers non-private information before processing some sanitizing scheme. In real-life scenarios the absence of such a server will jeopardize the privacy and security of the overall learning procedure. Hereafter, we present a new approach called SPEED which obtains differential privacy guarantees without the need for a trusted aggregation server. Building upon differentially private decentralized semi-supervised learning [123, 125], we introduce homomorphically encrypted operations to extend the set of threats considered so far. The procedure is summarized in Figure C.1. Our approach is supported by theoretical guarantees

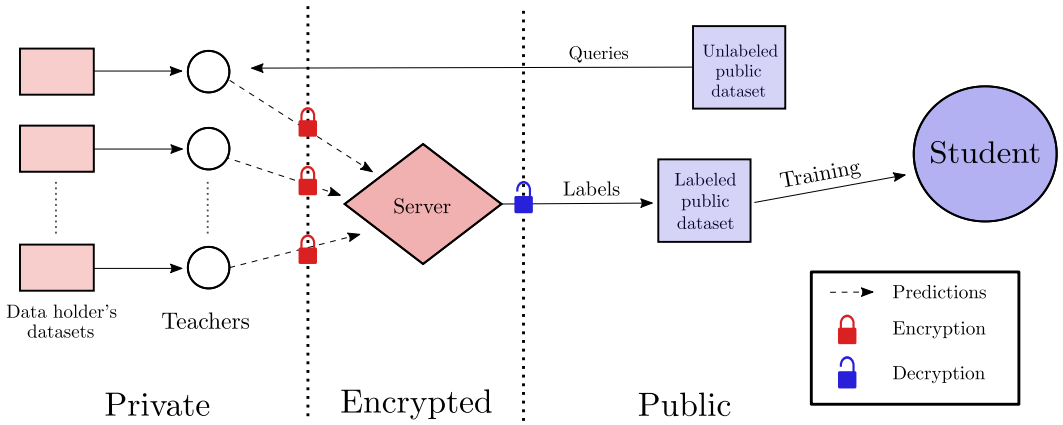


Figure C.1: Diagram illustrating SPEED deep learning framework.

in terms of differential privacy and provably-secure cryptography. In a nutshell, SPEED works as follows.

- First, every data owner builds a local model – *a.k.a.* teacher model – using its own database.
- Then, given a new unlabeled dataset, the teacher models output encrypted predictions and send them to the server which computes a differentially private aggregation *in* the encrypted domain to obtain an encrypted labeled dataset.
- From this new dataset, a collaborative model – *a.k.a.* student model – is learned in a semi-supervised manner.

C.2 Related works on private deep learning

The literature on private training of deep neural networks essentially uses either differential privacy, secure multiparty computing *or* homomorphic encryption – see *e.g.* [109]. Hereafter, we focus on differential privacy and homomorphic encryption.

Differential privacy and deep learning

Through the lens of differential privacy, we can design machine learning algorithms that are protective of the database private attributes. For deep learning tasks, a widely used technique is to use a noisy stochastic gradient descent [1, 176] during the learning procedure, and to keep track of the

privacy budget using the *moments accountant* scheme [1]. Even though some of these models can be satisfying when using a centralized database, none of them meet our problem requirements since it forces all the data owners to merge their databases.

To address these concerns some recent works considered to use differential privacy in decentralized settings close to the one we consider [13, 19, 30, 58, 123, 125]. Among them, the most efficient technique in terms of accuracy and privacy guarantees is Private Aggregation of Teacher Ensembles (PATE) first presented in [123] and refined in [125]. PATE uses private semi-supervised learning to privately transfer to the *student* model the knowledge of the ensemble of *teachers* by using a differentially private aggregation method. This approach considers a setting very close to ours with the notable difference that the aggregator is trusted. Hence applying PATE in our scenario makes the teacher models vulnerable. To tackle this issue, our work builds upon PATE idea, and adds a layer of homomorphic encryption in order for the overall learning framework to be kept secure.

Homomorphic Encryption – HE

HE allows to perform computations over encrypted data. In particular, this can be used so that the model can perform both training and prediction without handling cleartext data. In terms of learning, the naive approach would be to have the training sets homomorphically encrypted, sent to a server for training to be done in the encrypted domain and the resulting – encrypted – model be sent back to the participants for decryption. However, putting aside many subtleties, even by deploying all the arsenal available in the HE practitioner toolbox – batching, transciphering, etc. – this would be impractical as standard learning is both computation and know-how intensive and HE operations are intrinsically costly. As a consequence, there are only very few works that capitalize on HE for private training [70, 78] and inference [60, 84] of machine learning tasks. Moreover, since some attacks can be performed in a black-box setting, the system is still vulnerable to attacks from the end user who has access to the decryption key. In our framework, we do not use HE directly to build the model, we use it as a mean for the aggregation to be kept private. That way, we are protected against potential threats from the aggregator – which does not have the decryption key, and we keep a manageable computational overhead.

Private aggregation

Several approaches have been considered to limit the need for a trusted aggregator when applying differential privacy, for example by considering local differential privacy [47, 85, 87]. In practice it often results in applying too much noise, and maintaining utility can be difficult [87, 158] especially for deep learning applications. In order to recover more accuracy while keeping privacy, some works combined decentralized noise distribution – *a.k.a.* distributed differential privacy [147] – and encryption schemes [2, 69, 135, 147] in the context of aggregation of distributed time-series. Our work contributes to this line of research. Our framework, which combines distributed DP and HE, is the first one to be sufficiently efficient to investigate deep learning applications.

D Résumé en Francais de la Thèse

Cette annexe présente un résumé en français de ma thèse, rédigée en anglais. Elle comporte principalement une traduction du contexte et des motivations de mon travail ainsi que les grandes lignes de chacune de mes contributions.

Contents

Les modèles d'apprentissage automatique font partie de notre vie quotidienne et leurs faiblesses en termes de sécurité peuvent être utilisées pour nous nuire directement ou indirectement. Il est donc crucial de pouvoir prendre en compte et de traiter toute nouvelle vulnérabilité. De plus, le cadre juridique en Europe évolue, ce qui oblige les praticiens – des secteurs public et privé – à s'adapter rapidement à ces nouvelles préoccupations. Dans cette annexe, nous présentons d'abord le contexte dans lequel l'idée de cette thèse est née et nos principales motivations dans la Section D.1. Ensuite, nous présentons l'un des problèmes sur lequel nous nous sommes concentrés pendant ce travail de thèse: *La classification supervisée sous perturbations adverses*, ainsi

que quelques résultats de l'état de l'art dans la Section D.2. Nous résumons certaines de nos contributions au domaine dans la Section D.3. Pour chacune de nos contributions, nous résumons succinctement nos résultats et présentons quelques pistes d'améliorations. Enfin, nous listons nos productions de matériel scientifique et pédagogique dans la Section D.4, et nous présentons un problème ouvert pour la communauté en Section D.5.

D.1 Contexte et motivations

C'est au cours des années 1950 que naît le concept de l'intelligence artificielle. Plus particulièrement, c'est souvent la conférence de Dartmouth de 1956 qui est considérée comme l'acte fondateur du concept [1]. À cette époque, le but était de comprendre et de tenter de reproduire l'intelligence humaine. Les approches proposées consistaient en l'utilisation des mathématiques pour décrire le monde, modéliser la perception humaine et simuler les mécanismes cérébraux. Soixante-dix ans plus tard, l'objectif initial de réplication des fonctions cérébrales a été largement supplanté par des projets technologiques visant à reproduire les performances humaines dans des tâches cognitives simples [142]. À cet effet, les réseaux de neurones profonds atteignent des performances remarquables dans des domaines applicatifs complexes tels que le traitement automatique du langage [132], la reconnaissance d'images [76] ou la reconnaissance de la parole [79].

L'impressionnante efficacité des technologies basées sur ce type de modèles les a rendu omniprésents tant dans l'industrie que dans certains secteurs publics. Cependant, des études récentes ont identifié plusieurs défauts majeurs des algorithmes d'intelligence artificiel tels que la fuite d'informations [120] ou la vulnérabilité aux perturbations adverses [20]. Ces faiblesses soulèvent de nombreuses questions sur la responsabilité juridique des fournisseurs des modèles et amènent les praticiens à réévaluer la confiance qu'ils placent dans les systèmes qu'ils utilisent.

D.1.1 Gestion des questions relatives à la vie privée: le Règlement Général sur la Protection des Données

La protection des données personnelles contre de potentielles fuites pendant un traitement statistique de celles-ci n'est pas exactement une nouvelle préoccupation; les fondements théoriques sur l'analyse de données à caractère sensible ont été largement établis dans les années 1980 [3, 42, 65]. Cependant, ces sujets sont revenus sur le devant de la scène notamment en 2008, lorsque Narayanan *et al.* [120] ont présenté une procédure de désanonymisation très efficace sur la base de données publiée pour le "Netflix Prize". En 2016, l'Union européenne a apporté une réponse à ces préoccupations d'un point de vue juridique en publiant le Règlement Général sur la Protection des Données [126] – RGPD.

Ce règlement vise à définir les obligations des fournisseurs de modèles en ce qui concerne les données à caractère sensible qu'ils utilisent[2]. Afin de se conformer au RGPD, les industries et

[1] Cette conférence est en réalité l'aboutissement de plusieurs travaux pionniers traitant de la notion d'intelligence artificielle par Mc Culloch, Pitts et Wiener [105, 157, 165] par la communauté cybernétique et par Turing [157] pour celle de l'informatique.

[2] Nous ne prétendons pas présenter ici une analyse complète de ce règlement. Pour que la discussion reste concise, nous nous contentons de souligner certains points que nous – en tant qu'informaticiens – jugeons essentiels.

les gouvernements sont tenus de concevoir des modèles pour lesquelles les mécanismes de protection contre la fuite des données soient plus élaborées. Ces nouvelles obligations, associées aux préoccupations déjà existantes des utilisateurs concernant leurs données personnelles, ont fait de la question de protection des données l'une des priorités de la communauté informatique. En conséquence, plusieurs définitions ont été introduites pour caractériser les algorithmes de protection de données dans le contexte de l'apprentissage supervisé et de l'analyse de données [57]. Parmi elles, la confidentialité différentielle [52] est devenue l'un des standards en permettant de fournir une définition forte et pratique de la protection de données. L'idée de cette définition est que les informations d'une personne de la base de données sont protégées si le résultat de toute analyse donne un résultat aussi probable, que la personne fasse ou non parti de l'ensemble des données à disposition de l'algorithme.

Plus formellement, on dit qu'un algorithme est différentiellement confidentiel si, compte tenu de deux bases de données similaires, il produit des résultats statistiquement indissociables. Cette définition de protection de données a été largement étudiée dans de nombreux cadres et applications – voir [50] pour un ouvrage de référence. Dans l'ensemble, l'apprentissage supervisé sous contrainte de protection des données est désormais un concept connu et intégré dans le paysage de la recherche en informatique. Il repose sur un cadre juridique approprié, et des solutions techniques telles que la confidentialité différentielle sont systématiquement mises en œuvre par les grandes entreprises – par exemple Google [54, 166] – et les entités publiques – comme le U.S. Census Bureau [102]. La bataille pour la protection des données utilisateurs n'est pas encore terminée, mais des efforts importants ont été déployés tant par les praticiens que par les chercheurs pour répondre aux exigences de notre époque en matière de protection de la vie privée.

D.1.2 Au-delà de la vie privée : interprétabilité, confiance et attaques adverses

Bien qu'il se concentre sur la protection des données, le RGPD comprend également un article – l'article 22 – sur le droit de recevoir une explication lorsqu'une décision à été prise à l'aide d'un algorithme [126]. Cela soulève un certain nombre de questions tant sur l'interprétabilité des algorithmes d'apprentissage supervisé que sur la confiance que les utilisateurs leur accordent [68]. Bien qu'il n'y ait pas encore de consensus clair sur la définition de l'interprétabilité ou de la confiance dans les modèles d'intelligence artificielle [24], des thèmes récurrents tels que les biais algorithmiques [4] ou la vulnérabilité aux perturbations [20, 152] sont souvent cités en exemple. Ces nouvelles préoccupations, ainsi que les questions de protection de données mentionnées plus tôt, sont parfois regroupées sous le nom d'*Intelligence Artificielle de Confiance*, concept qui a récemment attiré beaucoup d'attention. En outre, le déploiement de modèles d'intelligence artificielle dans les systèmes industriels et commerciaux à fort impact, ainsi que les récents progrès juridiques en matière de droit à la protection et à l'explication encouragent l'intensification de la recherche dans ce nouveau domaine.

Dans cette thèse, nous nous concentrons principalement sur la vulnérabilité des modèles aux perturbations adverses. Le terme de perturbation adverse – ou attaque adverse – désigne une perturbation soigneusement choisie et humainement imperceptible qui déclenche le dysfonctionnement d'un modèle. L'existence de ce type de faiblesse montre à quel point la communauté de l'apprentissage profond s'est éloignée de l'objectif initial de comprendre et reproduire la perception humaine.

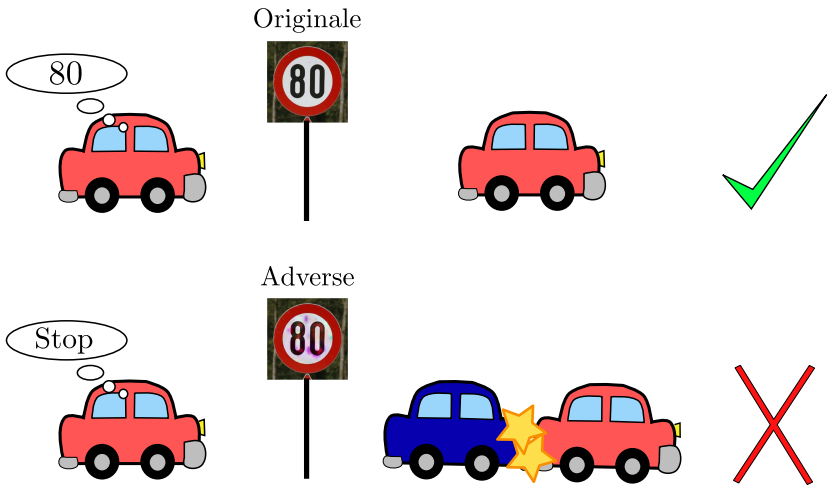


Figure D.1: Illustration d'une voiture autonome, dupée par une modification mineure d'un panneau de signalisation. En première ligne : le scénario sans attaque. En seconde ligne : scénario avec attaque. Les images des panneaux de signalisation proviennent d'une attaque présentée par Sitawarin et ses co-auteurs [150].

Pour mettre en évidence l'enjeu de sécurité que représentent les attaques adverses, nous prenons l'exemple des voitures autonomes. Récemment, les entreprises à la pointe des nouvelles technologies ont fait d'énormes investissements de recherche et de développement dans le domaine des voitures autonomes, c'est-à-dire des véhicules équipés d'un nombre considérable de caméras et de capteurs qui les aident à se déplacer avec peu ou pas d'intervention humaine. Une grande partie des informations recueillies par ces voitures sont traitées à l'aide de modèles d'apprentissage automatique embarqués. En particulier, les tâches de traitement d'images se font par le biais de réseaux de neurones profonds. Cependant, des travaux récents [55, 146, 150, 174] ont démontré que ces mêmes systèmes peuvent être dupés par des modifications marginales de panneaux de signalisation – par exemple en ajoutant des autocollants sur le panneau en question.

La Figure 1.2 illustre un contexte d'attaque où un adversaire a ajouté un tel autocollant sur un panneau de signalisation. Dans le premier schéma – en haut – la voiture analyse la version originale du panneau de signalisation, le reconnaît comme une limitation de vitesse et continue normalement. Dans le second schéma – en bas – la voiture rouge analyse une version modifiée du panneau de signalisation et le reconnaît comme un panneau "Stop" causant un accident avec la voiture bleue. Notez que dans ce cas, aucun humain n'aurait changé sa décision, mais la voiture le fait. Ce décalage manifeste entre la réponse humaine et la réponse du modèle peut conduire à d'innombrables problèmes de sécurité – ici par exemple un accident déclenché par une attaque sur un panneau de signalisation. Ce type de technologies est actuellement en cours de développement; il est donc crucial de s'adapter rapidement à la nouvelle menace que représente les attaques adverses, tant d'un point de vue technique que juridique.

D.2 Formalisation du/des problème(s) de classification

Dans cette section, nous commençons par faire quelques rappels sur le problème de classification dans le cadre standard – c'est-à-dire sans adversaire. Ensuite, nous présentons le problème de classification en présence d'un adversaire et identifions les problèmes fondamentaux auxquels nous souhaitons apporter des réponses.

D.2.1 Classification dans le cadre standard

Considérons le problème de classification supervisée avec un espace d'entrée $\mathcal{X}$ – des images – et un espace de sortie $\mathcal{Y}$ – des étiquettes décrivant les images. Pour simplifier, nous considérerons ici que $\mathcal{Y} = \{1, \dots, K\}$, ce qui signifie que chaque étiquette est caractérisée par un entier compris entre 1 et K. L'objectif d'un algorithme d'apprentissage supervisé est de construire une fonction de prédiction $c : \mathcal{X} \to \mathcal{Y}$ – aussi appelée un classifieur – qui fait correspondre à toute image $\boldsymbol{x} \in \mathcal{X}$ une étiquette $y \in \mathcal{Y}$. Pour trouver c, l'algorithme d'apprentissage a accès à un ensemble $\mathcal{S}$ de n couples entrée-sortie $\mathcal{S} := \{(\boldsymbol{x_1}, y_1), \dots, (\boldsymbol{x_n}, y_n)\}$ – aussi appelé *ensemble d'apprentissage*. L'hypothèse principale qui sous-tend la théorie de la classification est qu'il existe une certaine distribution $\mathcal{D}$ qui décrit le lien entre les images et les étiquettes et dont sont tirées indépendamment les paires $(\boldsymbol{x_i}, y_i)$.

Pour construire un classifieur, on définit en général une fonction $\boldsymbol{h} : \mathcal{X} \to \mathbb{R}^K$ appelée hypothèse, qui pour toute image $\boldsymbol{x} \in \mathcal{X}$ va renvoyer un vecteur de scores $\boldsymbol{h}(\boldsymbol{x}) := [\boldsymbol{h}_1(\boldsymbol{x}), \dots, \boldsymbol{h}_K(\boldsymbol{x})]^\intercal$. Ensuite, la fonction de prédiction c donne l'étiquette avec le meilleur score pour $\boldsymbol{h}$. Plus formellement, c s'écrit

$$c(\boldsymbol{x}) := \underset{k \in [K]}{\operatorname{argmax}}\, \boldsymbol{h}_k(\boldsymbol{x}).$$

Le problème revient donc à construire une fonction $\boldsymbol{h}$ qui décrit bien le lien entre les images et les étiquettes. Pour ce faire, l'algorithme d'apprentissage cherche à sélectionner $\boldsymbol{h}^*$ dans un espace fonctionnel $\mathcal{H}$ – aussi appelé classe d'hypothèses – qui soit solution du problème de *minimisation du risque*. Ce problème d'optimisation s'écrit comme ceci:

$$\inf_{\boldsymbol{h} \in \mathcal{H}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}(\boldsymbol{h}(\boldsymbol{x}), y)]\,, \tag{D.1}$$

où $\mathcal{L} : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}$ est une fonction de coût qui mesure à quel point $\boldsymbol{h}$ correspond à la distribution des données. Si $\mathcal{L}$ est suffisamment bien choisie – typiquement si elle est convexe et suffisamment régulière [9] – et si la classe d'hypothèses $\mathcal{H}$ est suffisamment riche[3], le classifieur c que nous obtenons aura une faible probabilité de donner une mauvaise étiquette pour un nouvel échantillon $(\boldsymbol{x}, y) \sim \mathcal{D}$.

En pratique, l'algorithme d'apprentissage n'a pas accès à la distribution $\mathcal{D}$; il ne peut donc pas estimer le risque $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}(\boldsymbol{h}(\boldsymbol{x}), y)]$. Pour trouver une approximation au Problème (D.1), un

[3] On peut considérer cette notion comme la taille de la classe d'hypothèses. Lorsque la classe d'hypothèses est assez grande, il est facile de trouver au moins une $\boldsymbol{h}$ qui décrive bien $\mathcal{D}$. Inversement, lorsqu'elle est trop petite, il est difficile de trouver un bon candidat. Pour plus de détails, nous incitons le lecteur à se référer à la partie anglaise du manuscrit.

algorithme d'apprentissage résout le problème de *minimisation du risque empirique* à la place. Ce problème s'écrit

$$\inf_{\boldsymbol{h}\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x_i}), y_i)\,. \tag{D.2}$$

Ensuite, pour évaluer la distance entre l'hypothèse sélectionnée $\boldsymbol{h}_{\mathcal{S}}$ et l'hypothèse optimale $\boldsymbol{h}^*$, on cherche à borner supérieurement la différence entre le risque et le risque empirique de toute hypothèse $\boldsymbol{h} \in \mathcal{H}$. Cette différence est connue sous le nom d'*écart de généralisation*. Intuitivement, si nous pouvons contrôler la différence entre le risque et le risque empirique d'une fonction quelconque dans $\mathcal{H}$, alors le problème de minimisation du risque et le problème de minimisation du risque empirique auront des solutions similaires.

Au regard de ce qui précède, le choix de la classe d'hypothèses $\mathcal{H}$ est central pour résoudre un problème de classification. D'une part, si la classe est trop grande, il sera difficile de contrôler l'écart de généralisation pour tous les éléments de la classe et le problème d'optimisation sera difficile. D'autre part, si elle est trop réduite, l'écart de généralisation sera facile à contrôler mais la classe pourrait ne pas être suffisamment riche pour décrire correctement le comportement de la distribution des données, ce qui conduira à sélectionner de mauvaises fonctions de prédiction. Un autre élément clé est la taille de l'ensemble d'apprentissage. Si nous avons suffisamment d'échantillons d'apprentissage, grâce à la loi uniforme des grands nombres, le risque empirique de toute hypothèse est une bonne approximation de son risque théorique.

Plus précisément, pour certaines classes d'hypothèses bien choisies, on peut limiter l'écart de généralisation de toute hypothèse par $O\left(\frac{1}{\sqrt{n}}\right)$. Ensuite, lorsque la taille de l'échantillon n est suffisamment grande, il suffit de résoudre le problème de minimisation du risque empirique – Problème (D.2) – pour obtenir une bonne approximation pour le problème de minimisation du risque – Problème (D.1). Présentons maintenant le cadre de classification alternatif que nous allons étudier dans ce manuscrit, à savoir la *classification sous perturbations adverses*.

D.2.2 Classification sous perturbations adverses

Étant données une hypothèse $\boldsymbol{h} \in \mathcal{H}$ et une paire image-étiquette $(\boldsymbol{x}, y) \sim \mathcal{D}$, le but d'un adversaire est de trouver une perturbation $\boldsymbol{\tau} \in \mathcal{X}$ telle que les affirmations suivantes soient vérifiées.

1. La perturbation doit être imperceptible pour un humain. Cela signifie qu'un humain ne peut pas distinguer visuellement l'image standard $\boldsymbol{x}$ de l'image *adverse* $\boldsymbol{x}+\boldsymbol{\tau}$.
2. La perturbation modifie suffisamment $\boldsymbol{x}$ pour que le classifieur fasse une erreur de classification. Plus formellement, l'adversaire recherche une perturbation $\boldsymbol{\tau} \in \mathcal{X}$ telle que $c(\boldsymbol{x}+\boldsymbol{\tau}) \neq y$.

Bien que la notion de modification imperceptible soit très naturelle pour un humain, elle est véritablement difficile à formaliser. Malgré ces difficultés, une condition suffisante pour garantir que l'attaque restera non détectée est de contraindre la perturbation $\boldsymbol{\tau}$ à avoir une petite norme ℓ_p. Cela signifie que pour tout $p \in [1, \infty]$, il existe un seuil $\alpha_p > 0$ pour lequel une perturbation $\boldsymbol{\tau}$ est imperceptible dès lors que $\|\boldsymbol{\tau}\|_p \leq \alpha_p$. La littérature sur les attaques adverses dans le

cadre de la classification d'images [27, 103] utilise généralement une norme ℓ_∞ ou ℓ_2 pour définir l'imperceptibilité [4].

Les exemples adverses représentent une menace sérieuse pour la sécurité des modèles d'intelligence artificielle. Il est donc primordial de re-formaliser le problème de minimisation du risque standard en intégrant l'adversaire dans le problème. L'objectif devient donc de minimiser le *risque adverse* – aussi appelé risque contradictoire, lorsque les manipulations sont limitées en norme ℓ_p. Nous appelons ce nouveau problème la *minimisation du risque adverse*. Il s'écrit comme suit:

$$\inf_{\boldsymbol{h}\in\mathcal{H}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x}+\boldsymbol{\tau}), y)\right], \tag{D.3}$$

où $B_p(\alpha_p) := \{\tau \in \mathcal{X} \text{ t.q. } \|\boldsymbol{\tau}\|_p \leq \alpha_p\}$. Dans ce nouveau problème, l'adversaire se concentre sur le problème de maximisation intérieur, tandis que l'algorithme d'apprentissage tente d'obtenir l'hypothèse optimale "sous attaque" $\boldsymbol{h}^*$ à partir de $\mathcal{H}$. Dans le cadre standard, nous pouvons la plupart du temps concevoir des classes d'hypothèses suffisamment riches pour que le problème de minimisation du risque donne une solution $\boldsymbol{h}^*$ avec un risque faible. Mais dans le cadre adverse, on ne sait pas si cette affirmation tient toujours. D'où la question suivante.

Q1: *Existe-t-il une classe d'hypothèses $\mathcal{H}$ pour laquelle le problème de minimisation du risque adverse a une solution $\boldsymbol{h}^*$ avec un faible risque adverse?*

À première vue – au regard de la littérature empirique sur les exemples adverses – la réponse semble être non. En effet, un grand nombre de travaux ont tenté de concevoir des modèles qui seraient moins vulnérables aux manipulations [67, 81, 107, 162, 170] mais la plupart d'entre eux se sont avérés – avec le temps – inefficaces contre des attaques plus sophistiquées [6, 27, 38, 77, 154]. Néanmoins, il est important d'étudier cette question d'un point de vue théorique pour apporter des réponses négatives définitives ou pour concevoir des modèles plus robustes.

Supposons un instant que **Q1** ait une réponse positive et que nous puissions concevoir une classe d'hypothèses $\mathcal{H}$ pour laquelle la minimisation du risque adverse a une solution $\boldsymbol{h}^*$ avec un faible risque contradictoire. Par analogie avec le cadre standard, étant donné les n exemples d'apprentissage $\mathcal{S} := \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_n}, y_n)\}$, nous voulons trouver une solution au problème de minimisation du risque adverse en étudiant sa contrepartie empirique, le *risque empirique adverse*. Ce problème d'optimisation s'écrit

$$\inf_{\boldsymbol{h}\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sup_{\boldsymbol{\tau}\in B_p(\alpha_p)} \mathcal{L}(\boldsymbol{h}(\boldsymbol{x_i}+\boldsymbol{\tau}), y_i)\,. \tag{D.4}$$

En présence d'un adversaire, plusieurs problèmes majeurs apparaissent dans la minimisation du risque empirique. Nous présentons ci-dessous quelques pointeurs bibliographiques qui per-

[4] Il arrive aussi parfois que l'on utilise une norme ℓ_1 [33] ou une semi-norme ℓ_0 [124]. Notez que ces normes ont des comportements très différents dans les espaces de grande dimension, d'où l'impact crucial qu'a le choix de p sur la réponse que l'on donne à **Q1** et **Q2** ci-dessous. Pour plus de détails, nous incitons le lecteur à se référer à la partie anglaise du manuscrit.

mettent de mieux comprendre les difficultés et enjeux de la minimisation du risque empirique adverse.

- Comme récemment souligné par Madry *et al.* [103], l'écart de généralisation adverse – c'est-à-dire l'écart entre le risque empirique contradictoire et le risque contradictoire – peut être beaucoup plus important que dans le cadre standard. Plus particulièrement, Madry *et al.* [103] ont remarqué qu'il est possible d'atteindre une précision adverse de $0,96$ pendant l'apprentissage contre $0,47$ pendant l'étape de test. Cet écart entre les performances pendant ces deux étapes est nettement plus important que ce que les modèles atteignent habituellement dans le cadre standard. En effet, l'adversaire rend le problème dépendant de la dimension de $\mathcal{X}$, et donc beaucoup plus difficile.

- Pour mieux comprendre d'où provient la difficulté de résoudre le problème de classification sous perturbations adverses, un certain nombre de travaux théoriques ont été menés. Notamment, Schmidt *et al.* [141] ont montré dans un cadre joué que nous n'avons besoin que de $O(1)$ exemples d'entraînement pour avoir un petit écart de généralisation. Par contre, en présence d'un adversaire ℓ_∞, nous avons besoin d'au moins $O(\sqrt{d})$ échantillons. Cette étude a été suivie de plusieurs avancée majeurs [7, 39, 175] démontrant que la généralisation contradictoire dépend effectivement de la dimension du problème. Ainsi, en termes de taille de l'échantillon, la problème de classification adverse est plus difficile que celui de la classification standard.

- Un autre axe de recherche étudie le problème du point de vue des contraintes de calcul. Bubeck *et al.* [25] se sont récemment penchés sur cette question pour démontrer que même avec un ensemble d'apprentissage suffisamment large, il existe un ensemble de problèmes d'apprentissage pour lesquels l'apprentissage standard non robuste peut être effectué efficacement, mais demande des capacités de calcul considérable dans le cadre adverse.

- Enfin, il ne suffit pas toujours de trouver une solution qui minimise le risque adverse. Certains travaux récents [83, 151, 156, 180] ont apporté des arguments théoriques établissant que construire un modèle avec un faible risque adverse peut conduire à une augmentation de son risque standard. Ainsi, trouver une bonne approximation pour le problème de classification adverse – Problème (D.3) – peut conduire à une mauvaise solution pour le problème standard – Problème (D.1).

Au vu de l'état de l'art que nous venons de discuter, on se pose également la question suivante.

> **Q2:** *Peut-on trouver une classe $\mathcal{H}$ et une hypothèse $\boldsymbol{h}^* \in \mathcal{H}$ qui atteignent simultanément un petit risque standard et contradictoire ?*

D.3 Résumé des contributions de cette thèse

Dans cette thèse, nous cherchons à apporter des réponses aux problèmes énoncés précédemment. Tout d'abord, nous analysons le problème de la classification contradictoire et fournissons des

résultats montrant que les classifieurs randomisés – c'est-à-dire les classifieurs qui renvoient une variable aléatoire – sont de bons candidats pour donner une réponse positive à **Q1**. Ensuite, nous identifions des sous-classes de classifieurs randomisés qui fournissent des réponses positives à la fois à **Q1** et **Q2**. Enfin, nous présentons des méthodes simples pour construire ces classes en établissant des liens avec la littérature sur la protection des données personnelles.

D.3.1 Analyse du problème de la classification contradictoire – Q1

Notre première contribution consiste à construire de nouvelles intuitions sur le problème de la classification adverse. Pour ce faire, nous présentons la minimisation du risque contradictoire – Problème (D.3) – comme un jeu à somme nulle *infini* entre un défenseur – l'algorithme d'apprentissage – et un adversaire qui produit des exemples adverses. Certains travaux récents ont eux aussi abordé le problème des exemples d'adverses comme un jeu à deux joueurs [127, 138], cependant ils considèrent des versions restreintes du jeu – par exemple lorsque les joueurs n'ont qu'un ensemble fini de stratégies possibles. Nous étudions un cadre plus général qui nous permet d'avoir une idée précise de la nature fondamentale du jeu entre le classifieur et l'adversaire. Plus particulièrement, nous obtenons les résultats suivants.

1. Nous démontrons la non existence d'un équilibre de Nash dans le jeu (régularisé) lorsque le défenseur et l'adversaire jouent tous deux des stratégies déterministes. Ceci, associé à certains résultats récents obtenues dans des travaux connexes [18, 131], implique que les classes d'hypothèses déterministes peuvent ne pas être de bons candidats pour fournir une réponse positive à **Q1**. Nos conclusions mettent également en évidence une propriété très intéressante du problème de classification contradictoire : son instabilité. Cela signifie que la nature du jeu entre l'adversaire et le classifieur change complètement lorsque nous ajoutons un petit terme de régularisation. Cela nous amène à remettre en question certaines thèses actuelles sur la classification adverse et à nous demander si les conclusions existantes tiendraient toujours si nous considérons un adversaire réaliste.

2. Du point de vue de la théorie des jeux, l'étape suivante consiste naturellement à étudier des stratégies randomisées. Nous nous concentrons sur la randomisation des stratégies pour le défenseur – en laissant les stratégies de l'adversaire inchangées. Dans ce contexte, nous démontrons que les classifieurs aléatoires peuvent surpasser les classifieurs déterministes en termes de garanties théoriques de robustesse – Problème (D.3). Par conséquent, nous identifions les classifieurs aléatoires comme de bons candidats pour répondre à **Q1** positivement. De plus, ce résultat nous permet de développer une méthode algorithmique que nous nommons Boosted Adversarial Training (BAT). Cette méthode repose sur une construction simple et permet de générer un classifieur randomisé à partir d'un classifieur déterministe. Le classifieur randomisé obtenu donne de meilleurs résultats expérimentaux en terme de précision sous attaques adverses que le classifieur déterministe initial.

Il pourrait donc y avoir une classe d'hypothèses $\mathcal{H}$ aléatoires pour lesquelles le problème de minimisation du risque adverse a une solution $\boldsymbol{h}^*$ avec un faible risque contradictoire

Ce travail en collaboration avec Raphael Ettedgui, Geovani Rizk, Yann Chevaleyre et Jamal Atif a été publié à la Conférence Internationale sur l'Apprentissage Machine (ICML) 2020. Cette publication est accompagnée d'un ensemble de codes hébergé sur Github permettant de reproduire nos résultats expérimentaux.

- "Randomization matters, how to defend against strong adversarial attacks". *International Conference on Machine Learning (ICML)* 2020. R. Pinot, R. Ettedgui, G. Rizk, Y. Chevaleyre, J. Atif.
- https://github.com/MILES-PSL/Randomization-matters-How-to-defend-against-strong-adversarial-attacks

Ce travail ouvre un grand nombre de questions particulièrement intéressantes à la fois sur le plan théorique et pratique. Nous présentons ci-dessous quelques-unes des pistes potentielles.

Travail futur 1 : L'équilibre dans le régime randomisé

Il reste à étudier si un équilibre existe dans le régime randomisé. Cette question est séduisante d'un point de vue théorique, et nécessite d'étudier l'espace des adversaires randomisés ce qui implique plus de technicités. L'étude de cet équilibre est également étroitement liée à celle de la valeur du jeu, ce qui serait intéressant pour obtenir des bornes min-max sur la précision sous attaque des classifieurs randomisés.

Travail futur 2 : Étudier le saut de dualité

Pour le moment, nous avons montré qu'il n'y a pas d'équilibre de Nash Pure dans le jeu. Cela signifie que la dualité forte ne tient pas. Mais cela n'indique pas l'écart entre les valeurs du problème $\inf/\sup$ et du problème $\sup/\inf$ — aussi appelé le saut de dualité. L'évaluation de ce saut de dualité pourrait nous aider à construire une analyse plus fine de l'impact de la régularisation sur le jeu.

Travail futur 3 : Boosted Adversarial Training, une défense certifiée ?

Bien que les résultats expérimentaux montrent que Boosted Adversarial Training est plus performant dans le cadre de la classifiaction sous attaques adverses, l'algorithme que nous présentons ne fournit pas de garanties en termes de précision certifiée. Comme l'a démontré la littérature sur les attaques et les défenses, de meilleures attaques existent toujours. C'est pourquoi, nous devons approfondir les aspects théoriques de notre procédure, afin de prouver la robustesse des classifieurs randomisés que nous concevons.

D.3.2 Propriétés théoriques des classifieurs randomisés – Q1 & Q2

Pour notre deuxième contribution, nous étudions les classifieurs randomisés à travers le prisme de la théorie de l'apprentissage et de la théorie de l'information. Par analogie avec le cas déterministe,

nous définissons une notion de robustesse pour les classifieurs randomisés. Cette définition se résume à vérifier que le classifieur satisfasse une condition de Lipschitzité locale en ce qui concerne la norme ℓ_p sur $\mathcal{X}$, et une métrique de probabilité sur $\mathcal{Y}$. En notant $\mathcal{H}_{\text{Lip}}$ la classe des classifieur randomisés qui respectent cette condition de Lipschitz, nous présentons les résultats suivants.

1. Nous démontrons que pour toute hypothèse $\boldsymbol{h} \in \mathcal{H}_{\text{Lip}}$, il est possible de borner supérieurement l'écart entre le risque et le risque contradictoire de $\boldsymbol{h}$. Ce résultat indique qu'une bonne approximation du problème de minimisation du risque – Problème (D.1) – sur $\mathcal{H}_{\text{Lip}}$ est également une bonne approximation de la minimisation du risque contradictoire – Problème (D.3). Cela signifie que $\mathcal{H}_{\text{Lip}}$ est un bon candidat pour répondre à **Q2**.

2. Nous démontrons ensuite qu'il est possible de borner supérieurement l'écart de généralisation de toute hypothèse $\boldsymbol{h}$ dans $\mathcal{H}_{\text{Lip}}$. Cela signifie que, pour un ensemble d'apprentissage suffisamment important, la résolution du problème de minimisation du risque empirique – Problem (D.2) – sur $\mathcal{H}_{\text{Lip}}$ peut fournir une bonne solution au problème de minimisation du risque théorique. Enfin, nous analysons la stabilité du mode des classifieurs randomisés, ce qui nous permet de présenter un point de vue probabiliste sur un ensemble de techniques existantes regroupées sous l'appellation de lissage randomisé [34, 98, 100, 139]. Notre point de vue sur la randomisation en tant que stratégie de défense pourrait ouvrir la voie à une étude plus approfondie du lissage randomisé d'un point de vue théorique. Ces résultats nous permettent également d'offrir la réponse suivante à **Q1** et à **Q2**[5].

Il existe des catégories de classifieurs aléatoires pour lesquelles nous pouvons contrôler l'écart entre le risque adverse et le risque standard.

Une partie de ce travail en collaboration avec Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cédric Gouy-Pailler et Jamal Atif a été publiée à la Conférence Internationale sur les Systèmes de Traitement de l'Information Neuronale (NeuriPS) 2019. Une version étendue de ce travail est actuellement en cours de préparation dans le but d'une soumission à une revue.

- "Theoretical evidence for adversarial robustness through randomization".
  Version journal, en cours 2020.
- "Theoretical evidence for adversarial robustness through randomization".
  Avances in Neural Information Processing (NeurIPS) 2019.
  R. Pinot, L. Meunier, A. Araujo, H. Kashima, F. Yger, C. Gouy-Pailler, J. Atif.

Notre analyse pourrait être affinée de plusieurs façons. Nous en énumérons ici quelques-unes pour des futurs travaux possibles.

[5] Il convient toutefois de noter que ce résultat repose sur une hypothèse forte concernant l'espace d'entrée qui n'est pas toujours vérifiée. Le problème de trouver une sous-classe de $\mathcal{H}$ qui offre des bornes plus précises sur l'écart de généralisation reste une question ouverte.

Travail futur 1 : des bornes plus précises pour l'écart de généralisation

Nos résultats sur la généralisation standard des classifieurs randomisés pourraient être améliorés. Dans nos travaux futurs, nous visons à étudier ces résultats sous un nouvel angle. A cette fin, nous pourrions utiliser des outils techniques tels que le lemme de Massart ou la notion de dimension d'éclatement pour rendre la borne moins dépendante de la dimension du problème.

Travail futur 2 : étudier les propriétés du lissage randomisé

Nous avons établi des liens entre la propriété de préservation de mode des classifieurs randomisés et la technique de défense appelée lissage randomisé. Sur la base de ces preuves, nous pouvons borner l'écart entre les risques standards et les risques adverses pour cette défense. Une autre direction intéressante serait de montrer que les classifieurs basés sur le lissage randomisé ont un écart de généralisation similaire à celui des classifieurs randomisés que nous avons étudiés.

Travail futur 3 : f-divergences et métriques de probabilité définies par intégrales

Les résultats que nous avons obtenus jusque là reposent sur des propriété fondamentales de la distance de variation totale et de la divergence de Renyi. Ces deux divergences ont des propriétés intéressantes, mais nous pensons qu'elles constituent un cas particulier de classes de divergences plus générales pour lesquelles des résultats similaires pourraient être obtenus. L'étude de formes plus générales de divergences telles que les f-divergences et les métriques de probabilité définies par intégrales pourrait fournir quelques directions sur la généralité de la définition de robustesse que nous présentons dans ce manuscrit.

D.3.3 Méthode simple basée sur l'injection de bruit – Q2

Les contributions précédentes ont identifié une classe d'hypothèses randomisées $\mathcal{H}_{\text{Lip}}$, qui répond à la fois à **Q1** et **Q2** – au moins partiellement. Mais elles ne fournissent aucun moyen pratique pour construire cette classe. Notre dernière contribution aborde cette question en tirant les leçons de la littérature sur la protection des données. Plus précisément, notre contribution est la suivante.

1. Nous mettons en évidence des connections entre notre définition de la robustesse et la définition de la confidentialité différentielle. Les deux notions reposent sur les mêmes fondements théoriques, à savoir la stabilité sur des espaces de mesures. Par conséquent, les résultats obtenus jusqu'à présent en matière de protection des données peuvent facilement être transférés pour construire des classifieurs aléatoires robustes. Sur la base de cette idée, nous utilisons deux outils courants dans la littérature de la confidentialité différentielle – à savoir l'injection de bruit Gaussien et l'inégalité de data-processing [12] – pour concevoir des classes de classifieur aléatoires robustes.

2. L'injection de bruit est une méthode utilisée depuis longtemps dans les tâches d'apprentissage et de traitement du signal [29, 71, 111, 181]. Elle a également été largement étudiée dans plusieurs domaines de l'apprentissage supervisé et de l'optimisation – par exemple en optimisation robuste [16] ou dans les techniques d'augmentation de données [128]. Parallèlement à nos travaux, d'autres techniques d'injection de bruit ont été mises en place par la

communauté de la classification adverse [45, 170]. En particulier, Lecuyer *et al.* [98] ont développé le lissage randomisé, en utilisant des résultats liés à la confidentialité différentielle. Nos travaux s'inscrivent dans le même axe de recherche, cependant la nature de nos résultats est différente. Alors que le lissage randomisé se concentre sur la construction de défenses certifiées, nous étudions les mécanismes randomisés du point de vue de la théorie de l'information et de la théorie de l'apprentissage supervisé. Notre analyse permet de comprendre certaines propriétés fondamentales des défenses randomisées, comprenant – mais ne se limitant pas – au lissage randomisé. Nos résultats sont applicables à un large éventail de modèles d'intelligence artificielle, moyennant quelques adaptations mineures. Nous validons donc nos conclusions par des résultats expérimentaux utilisant des réseaux de neurones profonds et des jeux de données d'images standards – à savoir CIFAR10 et CIFAR100 [93]. Ces modèles peuvent simultanément offrir une prédiction précise et une robustesse raisonnable, donnant des réponses pratiques à **Q1** et **Q2**.

Nous pouvons construire facilement des classifieurs randomisés qui sont robustes aux attaques adverses.

Une partie de ce travail en collaboration avec Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cédric Gouy-Pailler et Jamal Atif a été publiée à la Conférence Internationale sur les Systèmes de Traitement de l'Information Neuronale (NeuriPS) 2019 et dans un Workshop à la Conférence Européenne sur l'Apprentissage Machine (ECML) 2019. De plus, cette publication est accompagné d'un ensemble de codes hébergé sur Github permettant de reproduire nos résultats expérimentaux.

- " A unified view on differential privacy and robustness to adversarial examples". *Workshop on Machine Learning for CyberSecurity (ECML-PKDD)* 2019. R. Pinot, F. Yger, C. Gouy-Pailler, J. Atif.
- "Theoretical evidence for adversarial robustness through randomization". *Avances in Neural Information Processing (NeurIPS)* 2019. R. Pinot, L. Meunier, A. Araujo, H. Kashima, F. Yger, C. Gouy-Pailler, J. Atif.
- `https://github.com/MILES-PSL/Adversarial-Robustness-Through-Randomization`

Les méthodes pratiques que nous avons développées pourraient être améliorées de plusieurs façons. Parmi celles-ci, nous énumérons ci-dessous quelques approches possibles.

Travail futur 1 : élargir le champ des adversaires possibles

Jusqu'à présent, nous avons identifié des mécanismes d'injection de bruit pour se défendre contre les attaques dont l'imperceptibilité est mesurée par une norme ℓ_2. Nous avons étendu notre étude à la norme ℓ_1, mais la question de savoir si nous pouvons construire des systèmes d'injection

de bruit pour nous défendre contre des perturbations en norme ℓ_∞ reste également ouverte. À cette fin, nous pourrions utiliser le lien fondamental que notre cadre partage avec le lissage randomisé pour étudier les bruits qui se sont déjà avérés utiles dans cette littérature. En particulier, Yang *et al.* [173] ont étudié de nouvelles classes de bruit en utilisant la théorie des cristaux de Wulff. Cela pourrait ouvrir des pistes intéressantes pour des mécanismes d'injection de bruit plus sophistiqués.

Travail futur 2 : établir des liens plus profonds avec la confidentialité différentielle

Le lien que nous avons établi avec la confidentialité différentielle est fondamental, et nous sommes loin d'avoir étudié tous ses aspects. Par exemple, nous pourrions concevoir des modèles aléatoires beaucoup plus sophistiqués basés sur ce lien, notamment en utilisant le mécanisme exponentiel, ou des procédures de vote différentiellement confidentiel [50]. La confidentialité différentielle est également connue pour avoir des propriétés très intéressantes pour la généralisation basée sur la théorie de la stabilité [11]. Ainsi, nous pourrions adapter les résultats précédents pour améliorer l'analyse que nous avons présentée sur les propriétés des classifieurs randomisés (au regard de l'écart de généralisation).

D.4 Autres matériels scientifiques et pédagogiques

D.4.1 Publications non évoqués dans les corps du manuscrit

Pendant cette thèse nous avons eu l'occasion de travailler sur différents aspects de la protection des données personnelles et de la robustesse aux exemples adverses. Dans la partie principale de ce manuscrit, nous avons essayé de donner un aperçu clair de nos contributions dans le domaine de l'apprentissage supervisé robuste. Nous nous sommes délibérément concentrés sur certaines de nos contributions les plus significatives afin de rendre le manuscrit léger et facile à suivre. Ce travail de thèse a également pris d'autres directions. Nous avons notamment étudié l'apprentissage non supervisé sous contraintes de confidentialité différentielle, et le développement d'outil cryptographique qui puisse être appliqués au développement de méthodes d'apprentissage profond. Nous listons ci-dessous les contributions qui ne sont pas directement traitées dans le corps du manuscrit.

- "SPEED: Secure, PrivatE, and Efficient Deep learning".
  preprint 2020.
  A. Grivet Sébert, R. Pinot, M. Zuber, C. Gouy-Pailler, R. Sirdey.
- "Advocating for Multiple Defense Strategies against Adversarial Examples".
  Workshop on Machine Learning for CyberSecurity (ECML-PKDD) 2020.
  A. Araujo, L. Meunier, R. Pinot and B. Negrevergne.
- "Graph-based Clustering under Differential Privacy".
  Uncertainty in Artificial Intelligence (UAI) 2018.
  R. Pinot, A. Morvan, F. Yger, C. Gouy-Pailler, J. Atif.

D.4.2 Publications à plus large audience

Tout au long de ce travail de thèse, nous n'avons pas seulement mis l'accent sur la production de publications scientifiques. Nous nous sommes également engagés dans la vulgarisation scientifique par le biais de démonstrations et de communiqués de presse. Nous pensons que c'est aussi le rôle des scientifiques, en particulier dans le domaine de l'intelligence artificielle, d'expliquer leur travail à un public plus large au sein de la communauté scientifique, et d'accroître les connaissances du public sur les défis et les enjeux des nouvelles technologies. Voici quelques-unes de nos contributions.

- "Attaques adversariales: comprendre pour atténuer les risques" (article de presse).
  Clef du CEA num 69 2020.
  Contributeurs: R. Pinot, C. Gouy-Pailler.
- "AI vs Wild. How to strengthen neural networks of AI systems" (démonstration).
  Consumer Electronic Show Las Vegas 2020.
  Contributors: C. Gouy-Pailler, E. Kawasaki, R. Pinot, F. Valente.
- "Randomization based defenses against adversarial examples" (démonstration).
  DIGIHALL days Paris Saclay 2019.
  Contributeurs: R. Pinot, C. Gouy-Pailler.
- "La recherche et les risques inhérents à l'IA" (article de presse).
  Préventique num 166 2019.
  Contributeurs: R. Pinot, C. Gouy-Pailler.

D.4.3 Responsabilités pédagogiques

Enfin, l'enseignement fait partie intégrante du parcours doctoral et le développement rapide de l'intelligence artificielle nécessite la conception de nouveaux supports d'apprentissage. Pendant la durée de cette thèse, j'ai également participé à l'élaboration de deux nouveaux cours d'apprentissage automatique.

- "Mathématiques du machine learning" – Université Paris-Dauphine - PSL.
  Master IDD première année 2019-2020.
  Lecturer: R. Pinot.
- "Trustworthy machine learning in practice" – Université Paris-Dauphine - PSL.
  Executive Master 2019-2020.
  Lecturers: A. Araujo, R. Pinot, G. Rizk.

D.5 Conclusion et problème ouvert pour la communauté

D.5.1 Conclusion

Dans cette thèse, nous avons étudié le problème de la classification contradictoire sous différents angles, en utilisant une série d'outils théoriques et pratiques. Nous avons essayé d'analyser le problème théorique finement pour pouvoir proposer des solutions pratiques et viables. Dans l'ensemble, analyser le problème sous ces différents angles, nous a permis de mieux le comprendre et de construire de nouveaux outils utiles sur le point de vue théorique et pratique. Nous pouvons résumer nos conclusions comme suit.

1. Nous avons d'abord présenté le problème comme un jeu à somme nulle infinie, et analysé les propriétés fondamentales du jeu sous différents types de régularisation. Cette analyse nous a permis de mieux comprendre le formalisme actuellement utilisé dans la communauté des exemples adverses, et nous a amenés à justifier l'utilisation de méthodes randomisées comme défenses contre les attaques adverses.

2. Nous avons ensuite étudié plus en détail les défenses aléatoires. Nous avons notamment développé de nouvelles approches pour étudier la robustesse des classifieurs aléatoires en utilisant la théorie de l'information, la théorie des probabilités et la théorie de l'apprentissage statistique. Cela nous a permis de mettre en évidence les propriétés que devraient respecter les classifieurs randomisés pour être robustes tout en maintenant une bonne précision. Plus particulièrement, nous avons identifié des conditions suffisantes pour que les classifieurs randomisés soient robustes et nous avons étudié la propriété de généralisation de ces classifieurs.

3. Enfin, nous avons élaboré des méthodes simples et pratiques pour concevoir des classifieurs aléatoires robustes en utilisant la théorie de l'information et les leçons que nous avons pu tiré de la littérature sur la confidentialité différentielle. Cela montre que nous pouvons construire des classifieurs aléatoires robustes en nous basant sur des architectures de réseaux de neurones profonds, et ouvre la voie à des travaux futurs passionnants, tant en théorie qu'en pratique.

Nous espérons que notre analyse a aidé la communauté à progresser et a apporté des perspectives nouvelles et intéressantes sur le problème difficile qu'est la classification contradictoire. Le domaine est encore jeune et de nombreuses pistes de recherche sont encore ouvertes. Tout au long du manuscrit, nous avons discuté des travaux futurs qui correspondent – plus ou moins – à des améliorations directes de nos résultats. Mais ici, nous aimerions prendre le temps de présenter un problème ouvert plus large pour l'ensemble de la communauté.

D.5.2 Problème ouvert : Repenser la théorie de l'apprentissage

La majeure partie de la littérature sur la théorie de l'apprentissage s'attache à démontrer la convergence du risque empirique vers le risque théorique en utilisant la loi uniforme des grands nombres et en contrôlant la complexité de la classe d'hypothèses. L'objectif est alors de trouver des classes de

modèles qui soient suffisamment grandes pour que la minimisation du risque empirique ait une petite valeur, et suffisamment petites pour que nous obtenions un faible écart de généralisation. Au regard de cet objectif – voir Figure D.2 à gauche – l'idée la plus courante est qu'un modèle dont l'erreur d'entraînement est nulle aura une mauvaise précision sur l'ensemble de test. Cependant, des travaux récents ont remis en question ce point de vue pour certains modèles tels que les réseaux de neurones. Par exemple, Zangh *et al.* [179] ont montré que nous pouvons apprendre un réseau profond pour la classification d'images sur CIFAR-10 qui a une précision de 1.0 sur les données d'entraînement et qui obtient plus de 0.85 de précision sur les données de test. Cela signifie que le sur-apprentissage du modèle est soit modeste, soit même inexistant. Pour revenir à la forme classique d'une borne en généralisation en théorie de l'apprentissage, lorsque l'erreur de d'apprentissage est nulle, on obtient

$$\mathcal{R}^{\text{opt}} \leq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}(\boldsymbol{h}(\boldsymbol{x}),y)] \leq O\left(\sqrt{\frac{\mathfrak{C}(n)}{n}}\right), \tag{D.5}$$

où $\mathcal{R}^{\text{opt}}$ est le risque du clasiifieur optimal de Bayes, et $\mathfrak{C}(n)$ est une mesure de complexité pour la classe d'hypothèses qui peut ou non dépendre de n. Lorsque $\mathcal{R}^{\text{opt}} = 0$, nous pouvons souvent montrer que $\sqrt{\frac{\mathfrak{C}(n)}{n}} \to 0$, ce qui est logique. Mais, lorsque $\mathcal{R}^{\text{opt}} > 0$[6], pour que le terme de droite permette d'expliquer l'erreur de manière non triviale, nous avons besoin que les constantes cachées dans $O\left(\sqrt{\frac{\mathfrak{C}(n)}{n}}\right)$ soient optimales – ce qui n'est jamais le cas pour des réseaux de neurones. Par conséquent, les idées de la théorie classique de l'apprentissage peuvent ne pas s'appliquer au cadre de l'apprentissage profond, ce qui signifie que l'analyse ne doit pas être uniquement basée sur la loi uniforme des grands nombres ou sur le contrôle de la classe d'hypothèses.

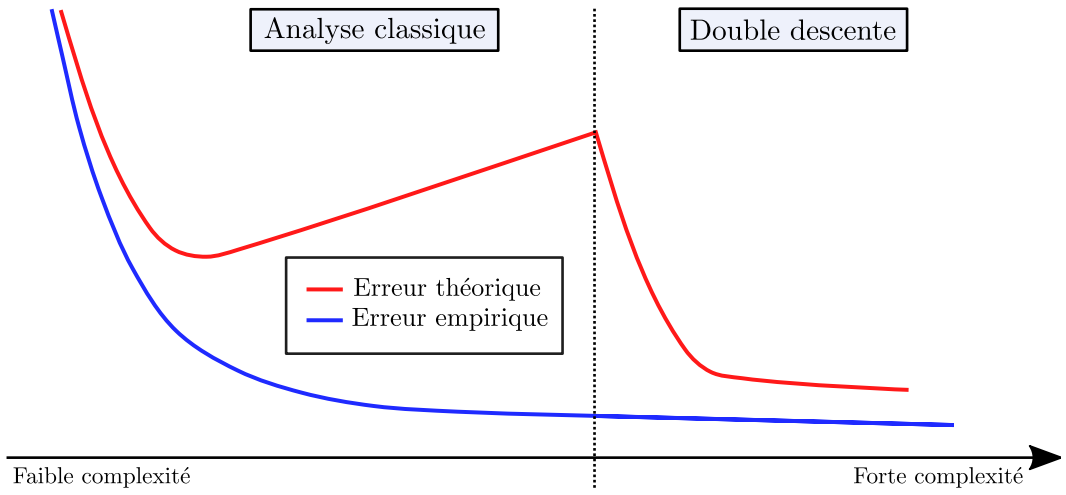


Figure D.2: Illustration du phénomène de la double descente.

[6]Notez que, la plupart du temps, nous aurons $\mathcal{R}^{\text{opt}} > 0$, étant donnée que le support des distributions conditionnelles $\{\mu_k\}_{k\in[K]}$ ne sont pas susceptibles d'être disjoints.

Nous constatons cependant que la minimisation du risque empirique fonctionne assez bien en pratique, c'est pourquoi les chercheurs ont commencé à s'interroger sur le contrôle de la complexité des modèles. Plus précisément, la question suivante se pose.

> En quoi, et à quel point la généralisation des modèles d'apprentissage supervisé dépendent de leur complexité ?

Un premier élément de réponse à cette question provient d'une observation faite en forçant les modèles à sur-apprendre – pour plusieurs modèles d'apprentissage machine, notamment les réseaux de neurones [15, 119]. Lorsque nous augmentons arbitrairement la complexité du modèle, nous constatons qu'après le sur-apprentissage, le risque théorique du modèle recommence à diminuer. Ce phénomène est appelé "double descente". Notez qu'après le sur-apprentissage, tous les modèles ont une erreur d'entraînement nulle, mais plus le modèle est grand, plus le risque théorique est faible. C'est un phénomène très surprenant qui a été observé sur de nombreux modèles de réseaux de neurones avancés. D'un point de vue théorique, le phénomène a été identifié et analysé pour les modèles linéaires. Pour revenir à l'objectif principal de ce manuscrit, nous pourrions également poser la question suivante.

> Ce changement de paradigme nous permettrait-il de mieux comprendre ou d'éviter les exemples adverses ?

Une façon intéressante de commencer à répondre à cette question est d'examiner le modèle des k plus proches voisins, comme le suggère Belkin *et al.* [14]. Il s'agit d'une technique de prédiction classique pour laquelle nous pouvons directement relier l'erreur attendue de l'algorithme au classifieur optimal de Bayes. Par exemple, Cover et Hard [36] ont montré que nous pouvons encadrer l'erreur du modèle comme ceci:

$$\mathcal{R}^{\text{opt}} \leq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}(\boldsymbol{h}(\boldsymbol{x}), y)] \leq \mathcal{R}^{\text{opt}}\left(2 - \frac{K\mathcal{R}^{\text{opt}}}{K-1}\right). \tag{D.6}$$

Comme les garanties de cette technique ne dépendent ni de la complexité du modèle ni de la loi uniforme des grands nombres, elle constitue un bon point de départ pour l'étude d'un nouveau formalisme. En ce qui concerne les exemples adverses, il convient de noter que certaines méthodes de classification, telles que le modèle des k plus proches voisins, se sont révélées robustes face à certaines formes d'exemples adverses [164]. Néanmoins, des résultats récents [14] ont également montré que si nous forçons un tel modèle à sur-apprendre – aussi appelé régime d'interpolation [14], alors les exemples adverses deviennent inévitables, comme cela semble être le cas pour les réseaux de neurones. Cela suggère que le phénomène des exemples adverses est étroitement lié au régime d'interpolation observé dans les réseaux de neurones profonds.

Bibliography

1. M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. "Deep learning with differential privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 308–318 (cited on pages 130, 131).
2. G. Ács and C. Castelluccia. "I have a dream!(differentially private smart metering)". In: *International Workshop on Information Hiding*. Springer. 2011, pp. 118–132 (cited on page 131).
3. N. R. Adam and J. C. Worthmann. "Security-control Methods for Statistical Databases: A Comparative Study". *ACM Comput. Surv.* 21:4, 1989, pp. 515–556. ISSN: 0360-0300 (cited on pages 6, 134).
4. J. Angwin, J. Larson, S. Mattu, and L. Kirchner. "Machine Bias. Thereś software used across the country to predict future criminals. And itś biased against blacks". In: ProPublica. 2017 (cited on pages 7, 135).
5. T. Asano, B. Bhattacharya, M. Keil, and F. Yao. "Clustering Algorithms Based on Minimum and Maximum Spanning Trees". In: *Proceedings of the Fourth Annual Symposium on Computational Geometry*. SCG '88. ACM, Urbana-Champaign, Illinois, USA, 1988, pp. 252–257 (cited on page 118).
6. A. Athalye, N. Carlini, and D. Wagner. "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. 2018 (cited on pages 11, 29, 56, 139).
7. P. Awasthi, N. Frank, and M. Mohri. "Adversarial Learning Guarantees for Linear Hypotheses and Neural Networks". *International Conference on Machine Learning*, 2020 (cited on pages 33, 34, 140).
8. P. L. Bartlett. "For valid generalization the size of the weights is more important than the size of the network". In: *Advances in neural information processing systems*. 1997, pp. 134–140 (cited on page 33).
9. P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. "Convexity, classification, and risk bounds". *Journal of the American Statistical Association* 101:473, 2006, pp. 138–156 (cited on pages 9, 17, 137).
10. P. L. Bartlett and S. Mendelson. "Rademacher and Gaussian complexities: Risk bounds and structural results". *Journal of Machine Learning Research* 3, 2002, pp. 463–482 (cited on page 20).

11. R. Bassily, K. Nissim, A. Smith, T. Steinke, U. Stemmer, and J. Ullman. "Algorithmic stability for adaptive data analysis". In: *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 2016, pp. 1046–1059 (cited on pages 103, 110, 146).
12. N. J. Beaudry and R. Renner. "An Intuitive Proof of the Data Processing Inequality". *Quantum Info. Comput.* 12:5-6, 2012, pp. 432–441. ISSN: 1533-7146 (cited on pages 90, 144).
13. B. K. Beaulieu-Jones, W. Yuan, S. G. Finlayson, and Z. S. Wu. "Privacy-Preserving Distributed Deep Learning for Clinical Data". *arXiv preprint arXiv:1812.01484* abs/1812.01484, 2018 (cited on pages 129, 131).
14. M. Belkin, D. J. Hsu, and P. Mitra. "Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate". In: *Advances in neural information processing systems*. 2018, pp. 2300–2311 (cited on pages 109, 150).
15. M. Belkin, D. Hsu, S. Ma, and S. Mandal. "Reconciling modern machine-learning practice and the classical bias–variance trade-off". *Proceedings of the National Academy of Sciences* 116:32, 2019, pp. 15849–15854 (cited on pages 108, 150).
16. A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Vol. 28. Princeton University Press, 2009 (cited on pages 29, 39, 144).
17. D. P. Bertsekas. "Control of uncertain systems with a set-membership description of the uncertainty." PhD thesis. Massachusetts Institute of Technology, 1971 (cited on page 29).
18. A. N. Bhagoji, D. Cullina, and P. Mittal. "Lower Bounds on Adversarial Robustness from Optimal Transport". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 7496–7508 (cited on pages 12, 37, 141).
19. A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers. "Protection against reconstruction and its applications in private federated learning". *arXiv preprint arXiv:1812.00984*, 2018 (cited on pages 129, 131).
20. B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. "Evasion attacks against machine learning at test time". In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2013, pp. 387–402 (cited on pages 5, 7, 24, 25, 134, 135).
21. A. Blum, K. Ligett, and A. Roth. "A Learning Theory Approach to Non-interactive Database Privacy". In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*. STOC '08. ACM, Victoria, British Columbia, Canada, 2008, pp. 609–618 (cited on page 122).
22. S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013 (cited on page 36).
23. G. W. Brown. "Iterative solution of games by fictitious play". *Activity analysis of production and allocation* 13:1, 1951, pp. 374–376 (cited on page 51).

24. M. Brückner and T. Scheffer. "Stackelberg Games for Adversarial Prediction Problems". In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. Association for Computing Machinery, San Diego, California, USA, 2011, pp. 547–555 (cited on pages 7, 135).

25. S. Bubeck, Y. T. Lee, E. Price, and I. Razenshteyn. "Adversarial examples from computational constraints". In: *International Conference on Machine Learning*. 2019, pp. 831–840 (cited on pages 34, 140).

26. N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, and A. Madry. "On Evaluating Adversarial Robustness". *arXiv preprint arXiv:1902.06705*, 2019 (cited on pages 25, 27, 56).

27. N. Carlini and D. Wagner. "Adversarial examples are not easily detected: Bypassing ten detection methods". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 2017, pp. 3–14 (cited on pages 10, 11, 29, 139).

28. N. Carlini and D. Wagner. "Towards evaluating the robustness of neural networks". In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 39–57 (cited on pages 25, 28, 57, 106).

29. F. Chapeau-Blondeau and D. Rousseau. "Noise-enhanced performance for an optimal Bayesian estimator". *IEEE Transactions on Signal Processing* 52:5, 2004, pp. 1327–1334 (cited on pages 39, 144).

30. M. Chase, R. Gilad-Bachrach, K. Laine, K. E. Lauter, and P. Rindal. "Private Collaborative Neural Network Learning." *IACR Cryptology ePrint Archive* 2017, 2017, p. 762 (cited on pages 129, 131).

31. K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi. "Broadening the Scope of Differential Privacy Using Metrics". In: *Privacy Enhancing Technologies*. Ed. by E. De Cristofaro and M. Wright. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 82–102. ISBN: 978-3-642-39077-7 (cited on page 88).

32. L. Chen, T. Yu, and R. Chirkova. "WaveCluster with Differential Privacy". In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. CIKM '15. ACM, Melbourne, Australia, 2015, pp. 1011–1020 (cited on page 122).

33. P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh. "EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples". In: *AAAI*. 2018 (cited on pages 10, 101, 139).

34. J. M. Cohen, E. Rosenfeld, and J. Z. Kolter. "Certified Adversarial Robustness via Randomized Smoothing". *arXiv preprint arXiv:1902.02918* (cited on pages 30, 31, 143).

35. T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012 (cited on page 90).

36. T. Cover and P. Hart. "Nearest neighbor pattern classification". *IEEE transactions on information theory* 13:1, 1967, pp. 21–27 (cited on pages 109, 150).

37. F. Croce and M. Hein. "Minimally distorted Adversarial Examples with a Fast Adaptive Boundary Attack". *International Conference on Machine Learning*, 2020 (cited on page 25).

38. F. Croce and M. Hein. "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks". In: *International Conference on Machine Learning*. 2020 (cited on pages 11, 28, 29, 56, 139).

39. D. Cullina, A. N. Bhagoji, and P. Mittal. "PAC-learning in the presence of adversaries". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*. Ed. by S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. 2018, pp. 228–239 (cited on pages 33, 140).

40. N. Dalvi, P. Domingos, S. Sanghai, and D. Verma. "Adversarial classification". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, pp. 99–108 (cited on page 24).

41. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*. 2009 (cited on page 24).

42. D. E. Denning. "Secure Statistical Databases with Random Sample Queries". *ACM Trans. Database Syst.* 5:3, 1980, pp. 291–315 (cited on pages 6, 134).

43. D. Desfontaines and B. Pejó. "Sok: Differential privacies". *Proceedings on Privacy Enhancing Technologies* 2020:2, 2020, pp. 288–313 (cited on page 88).

44. L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*. Vol. 31. Springer Science & Business Media, 2013 (cited on page 17).

45. G. S. Dhillon, K. Azizzadenesheli, J. D. Bernstein, J. Kossaifi, A. Khanna, Z. C. Lipton, and A. Anandkumar. "Stochastic activation pruning for robust adversarial defense". In: *International Conference on Learning Representations*. 2018 (cited on pages 39, 145).

46. E. Dohmatob. "Generalized No Free Lunch Theorem for Adversarial Robustness". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Long Beach, California, USA, 2019, pp. 1646–1654 (cited on pages 36, 37, 38, 44, 106).

47. J. C. Duchi, M. I. Jordan, and M. J. Wainwright. "Local privacy and statistical minimax rates". In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE. 2013, pp. 429–438 (cited on page 131).

48. K. Dvijotham, R. Stanforth, S. Gowal, T. A. Mann, and P. Kohli. "A Dual Approach to Scalable Verification of Deep Networks." In: *Uncertainty in Artificial Intelligence*. 2018 (cited on page 30).

49. C. Dwork. "A Firm Foundation for Private Data Analysis". *Commun. ACM* 54:1, 2011, pp. 86–95 (cited on page 122).

50. C. Dwork and A. Roth. "The Algorithmic Foundations of Differential Privacy". *Foundations and Trends® in Theoretical Computer Science* 9:3-4, 2013, pp. 211–407 (cited on pages 7, 86, 103, 121, 123, 124, 135, 146).

51. C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. "Fairness through awareness". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM. 2012, pp. 214–226 (cited on page 110).

52. C. Dwork, F. McSherry, K. Nissim, and A. Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography*. Springer Berlin Heidelberg, 2006, pp. 265–284 (cited on pages 6, 88, 135).

53. C. Dwork and R. Pottenger. "Toward practicing privacy". *Journal of the American Medical Informatics Association* 20:1, 2013, pp. 102–108 (cited on page 7).

54. Ú. Erlingsson, V. Pihur, and A. Korolova. "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS '14. Association for Computing Machinery, Scottsdale, Arizona, USA, 2014, pp. 1054–1067 (cited on pages 7, 135).

55. K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. "Robust physical-world attacks on deep learning visual classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1625–1634 (cited on pages 7, 136).

56. Y. Freund and R. E. Schapire. "A Decision Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Second European Conference on Computational Learning Theory (EuroCOLT-95)*. Ed. by P. M. B. Vitányi. Aix-en-Provence, France, 1995, pp. 23–37 (cited on page 51).

57. B. Fung, K. Wang, R. Cheng, and P. Yu. "Privacy-preserving Data Publishing: A Survey of Recent Developments". *ACM Comput. Surv.* 42:4, 2010, 14:1–14:53 (cited on pages 6, 135).

58. R. C. Geyer, T. Klein, and M. Nabi. "Differentially private federated learning: A client level perspective". *arXiv preprint arXiv:1712.07557*, 2017 (cited on pages 129, 131).

59. A. L. Gibbs and F. E. Su. "On Choosing and Bounding Probability Metrics". *International Statistical Review / Revue Internationale de Statistique* 70:3, 2002, pp. 419–435 (cited on pages 81, 82).

60. R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy". In: *International Conference on Machine Learning*. 2016, pp. 201–210 (cited on page 131).

61. G. L. Gilardoni. "On Pinsker's and Vajda's Type Inequalities for Csiszár's f-Divergences". *IEEE Transactions on Information Theory* 56:11, 2010, pp. 5377–5386 (cited on page 78).

62. J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen, and G. E. Dahl. "Motivating the rules of the game for adversarial example research". *arXiv preprint arXiv:1807.06732*, 2018 (cited on pages 27, 44, 106).

63. J. Gilmer, N. Ford, N. Carlini, and E. Cubuk. "Adversarial examples are a natural consequence of test error in noise". In: *International Conference on Machine Learning*. 2019, pp. 2280–2289 (cited on page 36).

64. A. Globerson and S. Roweis. "Nightmare at test time: robust learning by feature deletion". In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 353–360 (cited on page 24).

65. S. Goldwasser and S. Micali. "Probabilistic encryption". *Journal of Computer and System Sciences* 28:2, 1984, pp. 270–299 (cited on pages 6, 134).

66. I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016 (cited on page 38).

67. I. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations*. 2015 (cited on pages 11, 24, 25, 26, 28, 29, 35, 36, 50, 109, 139).

68. B. Goodman and S. Flaxman. "European Union regulations on algorithmic decision-making and a "right to explanation"". *AI magazine* 38:3, 2017, pp. 50–57 (cited on pages 7, 135).

69. S. Goryczka and L. Xiong. "A comprehensive comparison of multiparty secure additions with differential privacy". *IEEE transactions on dependable and secure computing* 14:5, 2015, pp. 463–477 (cited on page 131).

70. T. Graepel, K. Lauter, and M. Naehrig. "ML confidential: Machine learning on encrypted data". In: *International Conference on Information Security and Cryptology*. Springer. 2012, pp. 1–21 (cited on page 131).

71. Y. Grandvalet, S. Canu, and S. Boucheron. "Noise injection: Theoretical prospects". *Neural Computation* 9:5, 1997, pp. 1093–1108 (cited on pages 39, 144).

72. O. Grygorash, Y. Zhou, and Z. Jorgensen. "Minimum Spanning Tree Based Clustering Algorithms". In: *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*. 2006, pp. 73–81 (cited on page 118).

73. A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. "Differentially Private Combinatorial Optimization". In: *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '10. Society for Industrial and Applied Mathematics, Austin, Texas, 2010, pp. 1106–1125 (cited on page 123).

74. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001 (cited on page 23).

75. M. Hay, C. Li, G. Miklau, and D. Jensen. "Accurate Estimation of the Degree Distribution of Private Networks". In: *2009 Ninth IEEE International Conference on Data Mining*. 2009, pp. 169–178 (cited on page 121).

76. K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cited on pages 5, 134).

77. W. He, J. Wei, X. Chen, N. Carlini, and D. Song. "Adversarial example defense: Ensembles of weak defenses are not strong". In: *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*. 2017 (cited on pages 11, 139).

78. E. Hesamifard, H. Takabi, and M. Ghasemi. "Cryptodl: Deep neural networks over encrypted data". *arXiv preprint arXiv:1711.05189*, 2017 (cited on page 131).

79. G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, et al. "Deep neural networks for acoustic modeling in speech recognition". *IEEE Signal processing magazine* 29, 2012 (cited on pages 5, 134).

80. S.-S. Ho and S. Ruan. "Preserving Privacy for Interesting Location Pattern Mining from Trajectory Data". *Trans. Data Privacy* 6:1, 2013, pp. 87–106 (cited on page 122).

81. S. Hu, T. Yu, C. Guo, W.-L. Chao, and K. Q. Weinberger. "A new defense against adversarial images: Turning a weakness into a strength". In: *Advances in Neural Information Processing Systems*. 2019, pp. 1635–1646 (cited on pages 11, 139).

82. A. Ignatiev, N. Narodytska, and J. Marques-Silva. "On relating explanations and adversarial examples". In: *Advances in Neural Information Processing Systems*. 2019, pp. 15883–15893 (cited on page 110).

83. S. Jetley, N. A. Lord, and P. H. Torr. "With Friends like These, Who Needs Adversaries?" In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Curran Associates Inc., Montréal, Canada, 2018, pp. 10772–10782 (cited on pages 32, 140).

84. C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan. "{GAZELLE}: A low latency framework for secure neural network inference". In: *27th {USENIX} Security Symposium ({USENIX} Security 18)*. 2018, pp. 1651–1669 (cited on page 131).

85. P. Kairouz, S. Oh, and P. Viswanath. "Extremal mechanisms for local differential privacy". *The Journal of Machine Learning Research* 17:1, 2016, pp. 492–542 (cited on page 131).

86. V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev. "Private analysis of graph structure". *Proceedings of the VLDB Endowment* 4:11, 2011, pp. 1146–1157 (cited on page 121).

87. S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. "What can we learn privately?" *SIAM Journal on Computing* 40:3, 2011, pp. 793–826 (cited on page 131).

88. S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith. "Analyzing Graphs with Node Differential Privacy". In: *Proceedings of the 10th Theory of Cryptography Conference on Theory of Cryptography*. TCC'13. Springer-Verlag, Tokyo, Japan, 2013, pp. 457–476 (cited on page 121).

89. M. J. Kearns, R. E. Schapire, and L. M. Sellie. "Toward efficient agnostic learning". *Machine Learning* 17:2-3, 1994, pp. 115–141 (cited on pages 32, 110).

90. M. Kearns and M. Li. "Learning in the presence of malicious errors". *SIAM Journal on Computing* 22:4, 1993, pp. 807–837 (cited on pages 32, 110).

91. A. Kerckhoffs. "La cryptographie militaire". *Journal des sciences militaires*, 1883 (cited on page 25).

92. J. Khim and P.-L. Loh. "Adversarial risk bounds for binary classification via function transformation". *arXiv preprint arXiv:1810.09519* 2, 2018 (cited on page 33).

93. A. Krizhevsky and G. Hinton. *Learning multiple layers of features from tiny images*. Technical report. Citeseer, 2009 (cited on pages 13, 23, 24, 145).

94. A. Kumar, A. Levine, T. Goldstein, and S. Feizi. "Curse of dimensionality on randomized smoothing for certifiable robustness". *International Conference on Machine Learning*, 2020 (cited on page 31).

95. A. Kurakin, I. Goodfellow, and S. Bengio. "Adversarial examples in the physical world". *arXiv preprint arXiv:1607.02533*, 2016 (cited on pages 26, 28, 29).

96. A. Langlois, D. Stehlé, and R. Steinfeld. "GGHLite: More efficient multilinear maps from ideal lattices". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2014, pp. 239–256 (cited on page 76).

97. Y. LeCun and C. Cortes. "MNIST handwritten digit database", 2010 (cited on page 113).

98. M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. "Certified Robustness to Adversarial Examples with Differential Privacy". In: *2019 IEEE Symposium on Security and Privacy (SP)*. 2018, pp. 727–743 (cited on pages 30, 39, 143, 145).

99. G.-H. Lee, Y. Yuan, S. Chang, and T. Jaakkola. "Tight Certificates of Adversarial Robustness for Randomly Smoothed Classifiers". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 4910–4921 (cited on page 31).

100. B. Li, C. Chen, W. Wang, and L. Carin. "Second-Order Adversarial Attack and Certifiable Robustness". *arXiv preprint arXiv:1809.03113*, 2018 (cited on pages 30, 143).

101. D. Lowd and C. Meek. "Adversarial learning". In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 2005, pp. 641–647 (cited on page 24).

102. A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. "Privacy: Theory meets practice on the map". In: *2008 IEEE 24th international conference on data engineering*. IEEE. 2008, pp. 277–286 (cited on pages 7, 135).

103. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations*. 2018 (cited on pages 10, 11, 25, 28, 29, 32, 33, 56, 97, 98, 106, 139, 140).

104. P. Maini, E. Wong, and J. Z. Kolter. "Adversarial robustness against the union of multiple perturbation models". *International Conference on Machine Learning*, 2020 (cited on page 116).

105. W. S. McCulloch and W. Pitts. "A logical calculus of the ideas immanent in nervous activity". *The bulletin of mathematical biophysics* 5:4, 1943, pp. 115–133 (cited on pages 5, 134).

106. F. McSherry. "Privacy Integrated Queries". In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD)*. Association for Computing Machinery, Inc., 2009 (cited on page 122).

107. J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. "On Detecting Adversarial Perturbations". In: *Proceedings of 5th International Conference on Learning Representations (ICLR)*. 2017 (cited on pages 11, 139).

108. D. Mir and R. Wright. "A Differentially Private Graph Estimator". In: *2009 IEEE International Conference on Data Mining Workshops*. 2009, pp. 122–129 (cited on page 121).

109. F. Mireshghallah, M. Taram, P. Vepakomma, A. Singh, R. Raskar, and H. Esmaeilzadeh. *Privacy in Deep Learning: A Survey*. *arXiv preprint arXiv:2004.12254*. 2020 (cited on page 130).

110. I. Mironov. "Rényi Differential Privacy". In: *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*. 2017, pp. 263–275 (cited on page 88).

111. S. Mitaim and B. Kosko. "Adaptive stochastic resonance". *Proceedings of the IEEE* 86:11, 1998, pp. 2152–2183 (cited on pages 39, 144).

112. M. Mitrovic, M. Bun, A. Krause, and A. Karbasi. "Differentially Private Submodular Maximization: Data Summarization in Disguise". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, International Convention Centre, Sydney, Australia, 2017, pp. 2478–2487 (cited on page 123).

113. M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. 2018 (cited on pages 20, 22).

114. S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. "Deepfool: a simple and accurate method to fool deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2574–2582 (cited on pages 25, 36).

115. S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. Frossard. "Robustness via curvature regularization, and vice versa". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9078–9086 (cited on page 36).

116. A. Morvan, K. Choromanski, C. Gouy-Pailler, and J. Atif. "Graph sketching-based Massive Data Clustering". *SIAM Data Mining 2018*, 2018 (cited on pages 118, 119, 120).

117. A. Morvan. "Contributions to unsupervised learning from massive high-dimensional data streams : structuring, hashing and clustering". PhD thesis. PSL University, 2018 (cited on pages 120, 121, 123).

118. Y. Mülle, C. Clifton, and K. Böhm. "Privacy-Integrated Graph Clustering Through Differential Privacy". In: *EDBT/ICDT Workshops*. 2015 (cited on page 122).

119. P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. "Deep double descent: Where bigger models and more data hurt". *International Conference on Learning Representation*, 2020 (cited on pages 108, 150).

120. A. Narayanan and V. Shmatikov. "Robust De-anonymization of Large Sparse Datasets". In: *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 2008 (cited on pages 5, 6, 134).

121. H. H. Nguyen, A. Imine, and M. Rusinowitch. "Detecting Communities Under Differential Privacy". In: *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*. WPES '16. ACM, Vienna, Austria, 2016, pp. 83–93 (cited on page 122).

122. K. Nissim, S. Raskhodnikova, and A. Smith. "Smooth sensitivity and sampling in private data analysis". In: *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing - STOC*. ACM Press, 2007 (cited on pages 121, 122).

123. N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. *Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data*. *arXiv preprint arXiv:1610.05755*. 2016 (cited on pages 129, 130, 131).

124. N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. "Distillation as a defense to adversarial perturbations against deep neural networks". In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2016, pp. 582–597 (cited on pages 10, 139).

125. N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and U. Erlingsson. *Scalable Private Learning with PATE*. *arXiv preprint arXiv:1802.08908*. 2018 (cited on pages 129, 130, 131).

126. E. Parliament and E. Council. *Regulation (EU) 2016/679 of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC*. Technical report. European Parliament and European Council, 2016 (cited on pages 6, 7, 134, 135).

127. J. C. Perdomo and Y. Singer. "Robust Attacks against Multiple Classifiers". *arXiv preprint arXiv:1906.02816*, 2019 (cited on pages 38, 141).

128. L. Perez and J. Wang. "The effectiveness of data augmentation in image classification using deep learning". *arXiv preprint arXiv:1712.04621*, 2017 (cited on pages 38, 39, 144).

129. G. Peyré, M. Cuturi, et al. "Computational Optimal Transport: With Applications to Data Science". *Foundations and Trends® in Machine Learning* 11:5-6, 2019, pp. 355–607 (cited on page 68).

130. R. Prim. "Shortest connection networks and some generalizations". *The Bell System Technical Journal* 36:6, 1957, pp. 1389–1401 (cited on page 123).

131. M. S. Pydi and V. Jog. "Adversarial risk via optimal transport and optimal couplings". In: *International Conference on Machine Learning*. 2020 (cited on pages 12, 37, 141).

132. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. *Language Models are Unsupervised Multitask Learners*. Technical report. OpenAi, 2018 (cited on pages 5, 134).

133. P. Rafael. "Minimum spanning tree release under differential privacy constraints". MA thesis. 2017 (cited on page 124).

134. A. Raghunathan, J. Steinhardt, and P. S. Liang. "Semidefinite relaxations for certifying robustness to adversarial examples". In: *Advances in Neural Information Processing Systems*. 2018, pp. 10877–10887 (cited on page 30).

135. V. Rastogi and S. Nath. "Differentially private aggregation of distributed time-series with transformation and encryption". In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. 2010, pp. 735–746 (cited on page 131).

136. A. Rényi. *On measures of entropy and information*. Technical report. Hungarian Academy of Sciences Budapest Hungary, 1961 (cited on pages 75, 89).

137. C. Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007 (cited on page 68).

138. S. Rota Bulò, B. Biggio, I. Pillai, M. Pelillo, and F. Roli. "Randomized Prediction Games for Adversarial Machine Learning". *IEEE Transactions on Neural Networks and Learning Systems* 28:11, 2017, pp. 2466–2478 (cited on pages 38, 141).

139. H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang. "Provably robust deep learning via adversarially trained smoothed classifiers". In: *Advances in Neural Information Processing Systems*. 2019, pp. 11289–11300 (cited on pages 30, 31, 143).

140. S. E. Schaeffer. "Graph clustering". *Computer Science Review* 1:1, 2007, pp. 27–64 (cited on page 118).

141. L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry. "Adversarially robust generalization requires more data". In: *Advances in Neural Information Processing Systems*. 2018, pp. 5014–5026 (cited on pages 11, 33, 140).

142. E. Schmitt. "Explore, visualise, decide: a methodological paradigm for knowledge production based on Big Data Analysis". PhD Thesis. Université de technologie de Compiègne, 2018 (cited on pages 5, 134).

143. A. Sealfon. "Shortest Paths and Distances with Differential Privacy". In: *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems - PODS*. ACM Press, 2016 (cited on page 122).

144. A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein. "Are adversarial examples inevitable?" *International Conference on Learning Representtation*, 2018 (cited on page 36).

145. S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014 (cited on pages 20, 23, 70).

146. M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition". In: *Proceedings of the 2016 acm sigsac conference on computer and communications security*. 2016, pp. 1528–1540 (cited on pages 7, 136).

147. E. Shi, T. H. Chan, E. Rieffel, R. Chow, and D. Song. "Privacy-preserving aggregation of time-series data". In: *Proc. NDSS*. Vol. 2. Citeseer. 2011, pp. 1–17 (cited on page 131).

148. C.-J. Simon-Gabriel, Y. Ollivier, L. Bottou, B. Schölkopf, and D. Lopez-Paz. "First-order adversarial vulnerability of neural networks and input dimension". In: *International Conference on Machine Learning*. 2019, pp. 5809–5817 (cited on pages 11, 26).

149. A. Sinha, H. Namkoong, and J. Duchi. "Certifying some distributional robustness with principled adversarial training". *arXiv preprint arXiv:1710.10571*, 2017 (cited on page 29).

150. C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal. "Darts: Deceiving autonomous cars with toxic signs". *arXiv preprint arXiv:1802.06430*, 2018 (cited on pages 7, 8, 136).

151. D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao. "Is Robustness the Cost of Accuracy?–A Comprehensive Study on the Robustness of 18 Deep Image Classification Models". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 631–648 (cited on pages 32, 140).

152. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. "Intriguing properties of neural networks". In: *International Conference on Learning Representations*. 2014 (cited on pages 7, 24, 25, 35, 135).

153. F. Tramèr and D. Boneh. "Adversarial training and robustness for multiple perturbations". In: *Advances in Neural Information Processing Systems*. 2019, pp. 5866–5876 (cited on page 116).

154. F. Tramer, N. Carlini, W. Brendel, and A. Madry. *On Adaptive Attacks to Adversarial Example Defenses*. *arXiv preprint arXiv:2002.08347*. 2020 (cited on pages 11, 28, 29, 55, 56, 139).

155. F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. "The Space of Transferable Adversarial Examples". *arXiv*, 2017 (cited on pages 35, 50).

156. D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. "Robustness May Be at Odds with Accuracy". *International Conference on Learning Representation*, 2019 (cited on pages 11, 32, 33, 140).

157. A. Turing. "Computing Machinery and Intelligence". *Mind* 59:236, 1950, pp. 433–460 (cited on pages 5, 134).

158. J. Ullman. "Tight lower bounds for locally differentially private selection". *arXiv preprint arXiv:1802.02638*, 2018 (cited on page 131).

159. I. Vajda. "Note on Discrimination Information and Variation". *IEEE Trans. Inform. Theory* 16:6, 1970, pp. 771–773 (cited on pages 78, 79).

160. A. W. Van der Vaart. *Asymptotic statistics*. Vol. 3. Cambridge university press, 2000 (cited on page 70).

161. T. van Erven and P. Harremos. "Rényi Divergence and Kullback-Leibler Divergence". *IEEE Transactions on Information Theory* 60:7, 2014, pp. 3797–3820 (cited on pages 75, 90).

162. G. Verma and A. Swami. "Error Correcting Output Codes Improve Probability Estimation and Adversarial Robustness of Deep Neural Networks". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8646–8656 (cited on pages 11, 139).

163. C. Villani. *Topics in optimal transportation*. 58. American Mathematical Soc., 2003 (cited on pages 37, 68).

164. Y. Wang, S. Jha, and K. Chaudhuri. "Analyzing the robustness of nearest neighbors to adversarial examples". In: *International Conference on Machine Learning*. 2018, pp. 5133–5142 (cited on pages 109, 150).

165. N. Wiener. "Cybernetics; or control and communication in the animal and the machine.", 1948 (cited on pages 5, 134).

166. R. J. Wilson, C. Y. Zhang, W. Lam, D. Desfontaines, D. Simmons-Marengo, and B. Gipson. "Differentially private sql with bounded user contribution". *Proceedings on Privacy Enhancing Technologies* 2020:2, 2020, pp. 230–250 (cited on pages 7, 135).

167. R. R. Wiyatno, A. Xu, O. Dia, and A. de Berker. *Adversarial Examples in Modern Machine Learning: A Review*. *arXiv preprint arXiv:1911.05268*. 2019 (cited on page 29).

168. E. Wong and Z. Kolter. "Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope". In: *International Conference on Machine Learning*. 2018, pp. 5286–5295 (cited on page 30).

169. E. Wong, F. Schmidt, J. H. Metzen, and J. Z. Kolter. "Scaling provable adversarial defenses". In: *Advances in Neural Information Processing Systems*. 2018, pp. 8400–8409 (cited on page 30).

170. C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. "Mitigating Adversarial Effects Through Randomization". In: *International Conference on Learning Representations*. 2018 (cited on pages 11, 39, 139, 145).

171. H. Xu and S. Mannor. "Robustness and generalization". *Machine learning* 86:3, 2012, pp. 391–423 (cited on pages 34, 72).

172. Y. Xu, V. Olman, and D. Xu. "Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees". 18:4, 2002, pp. 536–545 (cited on pages 118, 119).

173. G. Yang, T. Duan, E. Hu, H. Salman, I. Razenshteyn, and J. Li. "Randomized Smoothing of All Shapes and Sizes". *International Conference on Machine Learning*, 2020 (cited on pages 31, 103, 146).

174. D. Yao, Z. Xi, Z. Tianyi, C. Chen, L. Guannan, and K. Miryung. "An Analysis of Adversarial Attacks and Defenses on Autonomous Driving Models". In: *18th Annual IEEE International Conference on Pervasive Computing and Communications*. IEEE. 2020 (cited on pages 7, 136).

175. D. Yin, R. Kannan, and P. Bartlett. "Rademacher complexity for adversarially robust generalization". In: *International Conference on Machine Learning*. 2019, pp. 7085–7094 (cited on pages 33, 140).

176. L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex. "Differentially private model publishing for deep learning". In: *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2019, pp. 332–349 (cited on page 130).

177. S. Zagoruyko and N. Komodakis. "Wide Residual Networks". In: *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2016, pp. 87.1–87.12. ISBN: 1-901725-59-6 (cited on pages 55, 95, 114).

178. C. T. Zahn. "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters". *IEEE Trans. Comput.* 20:1, 1971, pp. 68–86 (cited on page 118).

179. C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. "Understanding deep learning requires rethinking generalization". *arXiv preprint arXiv:1611.03530*, 2016 (cited on pages 107, 149).

180. H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. "Theoretically principled trade-off between robustness and accuracy". *International conference on Machine Learning*, 2019 (cited on pages 11, 29, 32, 33, 140).

181. S. Zozor and P.-O. Amblard. “Stochastic resonance in discrete time nonlinear AR(1) models”. *IEEE transactions on Signal Processing* 47:1, 1999, pp. 108–122 (cited on pages 39, 144).

RÉSUMÉ

Les modèles d'intelligence artificielle font partie de notre vie quotidienne et leurs faiblesses peuvent être utilisées pour nous nuire directement ou indirectement. Il est donc crucial de pouvoir prendre en compte et traiter toute nouvelle vulnérabilité. Par ailleurs, le cadre juridique en Europe évolue, ce qui oblige les professionnels, tant du secteur privé que du secteur public, à s'adapter rapidement à de nouvelles préoccupations en matière de sécurité et de transparence des algorithmes.

Cette thèse étudie comment construire des modèles plus sûrs. Nous étudions en particulier une nouvelle menace: les attaques adverses. La vulnérabilité des modèles d'intelligence artificielle à ces attaques est un véritable problème de sécurité, en particulier lorsque ceux-ci sont utilisés dans des technologies sensibles telles que les voitures autonomes. Outre les questions de sécurité, ces attaques montrent à quel point nous manquons de recul sur les modèles que l'industrie utilise quotidiennement. Nous fournissons des éléments de réflexion sur les attaques adverses et proposons des méthodes simples pour atténuer leurs effets en utilisant la théorie de l'apprentissage supervisé, de l'information, et des probabilités.

MOTS CLÉS

Théorie de l'apprentissage supervisé - Théorie de l'Information - Intelligence Artificielle de confiance - Exemples adverses

ABSTRACT

Machine learning models are part of our everyday life and their weaknesses in terms of security or privacy can be used to harm us either directly or indirectly. It is thus crucial to be able to account for, and deal with, any new vulnerabilities. Besides, the legal framework in Europe is evolving, forcing practitioners, from both the private and the public sectors, to adapt quickly to these new concerns.

This thesis studies how to build safer machine learning models. In particular, we focus on a new security concern called adversarial attacks. The vulnerability of state-of-the-art models to these attacks has genuine security implications especially when models are used in AI-driven technologies, e.g. for self-driving cars. Besides security issues, these attacks show how little we know about the models used everyday in the industry, and how little control we have over them. We provide some insights explaining how adversarial attacks work, and how to mitigate them by using statistical learning theory as well as probability and information theory.

KEYWORDS

Statistical learning theory - Information theory - Trustworthy machine learning - Adversarial examples