Robustesse structurelle des architectures d'apprentissage profond

Carlos LASSANCE[1], Vincent GRIPON[1], Jian TANG[2], Antonio ORTEGA[3]

[1]IMT Atlantique et Université de Montréal, Lab-STICC et Mila

[2]HEC Montréal, Mila

[3]University of Southern California

carlos.rosarkoslassance@imt-atlantique.fr, vincent.gripon@imt-atlantique.fr, jian.tang@mila.quebec, antonio.ortega@ee.usc.edu

Résumé – Les réseaux de neurones profonds sont devenus les références dans beaucoup de problèmes d'apprentissage machine. Malheureusement, ils sont sensibles à divers types de bruits ou à des déformations des entrées. Dans ce travail, nous introduisons une nouvelle définition de robustesse caractérisant la constante de Lipschitz de la fonction du réseau dans un sous-ensemble restreint de son domaine de définition. Nous comparons cette définition à celles existantes, et discutons des liens avec différentes méthodes introduites dans la littérature afin accroître la robustesse des réseaux.

Abstract – Deep Networks have been shown to provide state-of-the-art performance in many machine learning challenges. Unfortunately, they are susceptible to various types of noise, including adversarial attacks and corrupted inputs. In this work we introduce a formal definition of robustness which can be viewed as a localized Lipschitz constant of the network function, quantified in the domain of the data to be classified. We compare this notion of robustness to existing ones, and study its connections with methods in the literature. We evaluate this metric by performing experiments on various competitive vision datasets.

1 Introduction

Ces dernières années les réseaux de neurones profonds se sont imposés comme l'état de l'art dans un grand nombre de problèmes de l'apprentissage machine, dans des domaines aussi divers que la vision [4, 6, 9] et le traitement du langage naturel [14, 18]. Une raison de ce succès est la propriété d'approximation universelle [8], leur permettant d'approcher n'importe quelle fonction compatible avec les données d'entraînement. Mais cette propriété est à double tranchant, car la fonction entraînée peut se révéler mal adaptée à des déformations dans le domaine de définition. Les attaques adversaires [5, 17] (i.e., des changements quasi-imperceptibles construits pour tromper la fonction du réseau) illustrent bien ce risque de mauvaise généralisation. Les déformations isotropiques [13] ou le bruit appliqué aux entrées [7] sont aussi susceptibles de produire de mauvaises prédictions. Dans des domaines aussi sensibles que la conduite autonome de véhicules ou la chirurgie assistée par les robots, la robustesse à ces déviations est un problème clé.

Dans la littérature se trouvent plusieurs méthodes permettant d'augmenter la robustesse des fonctions des réseaux. Il est par exemple possible d'augmenter artificiellement l'ensemble des données d'entraînement en appliquant du bruit ou des corruptions [3, 10, 12, 15]. Ainsi, les réseaux obtenus deviennent plus robustes à ces déformations. Il n'y a malheureusement pas de raison de penser qu'augmenter la robustesse à un type de bruit a pour conséquence d'augmenter la robustesse à tout type de déviation [2, 7]. Pour atteindre une robustesse universelle, d'autres approches ont pour objectif des propriétés structurelles de la fonction du réseau, comme par exemple la contraindre à disposer d'une petite constante de Lipschitz [1, 16].

Contraindre la constante de Lipschitz de la fonction d'un réseau peut être problématique. En effet, cette constante caractérise la pente de la fonction *partout dans le domaine de définition*. Toutefois, dans un contexte de classification, il est attendu des transitions nettes à la frontière des classes, alors que la pente devrait être petite autour des exemples fournis pour l'apprentissage. En d'autres mots, la pente de la fonction du réseau devrait dépendre des régions du domaine de définition et une caractérisation globale n'est pas nécessairement souhaitable. Pour illustrer ce point, nous avons tracé dans la figure 1 la proportion de paires de points de classes différentes, pris dans l'ensemble d'entraînement, incompatibles avec une contrainte de Lipschitz donnée, pour la norme $\mathcal{L}_\infty$ et en supposant que la sortie devrait être un vecteur indicateur de la classe correspondante. Cet exemple illustre que la volonté de contraindre la constante de Lipschitz de façon globale est incompatible avec l'objectif d'entraînement.

Dans cet article nous introduisons une nouvelle définition formelle de robustesse, pouvant être vue comme une forme *locale* de la constante de Lispchitz, quantifiée dans le domaine des exemples d'entraînement. Cette définition assure qu'une petite transformation d'un exemple d'entraînement ne peut pas modifier de façon importante la décision du réseau. Nous discutons

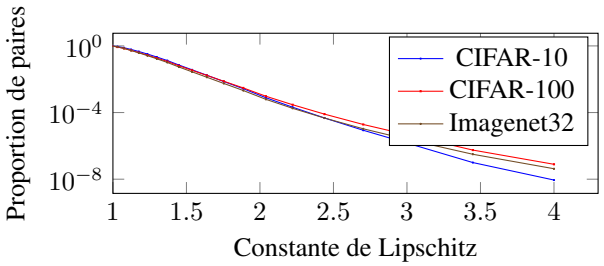

FIGURE 1 – Évolution de la proportion de paires d'exemples d'entraînement de classes différentes incompatibles avec une contrainte de Lipschitz α, pour différentes bases de données et la norme $\mathcal{L}_\infty$.

des relations entre cette définition et des méthodes récemment introduites dans la littérature pour augmenter la robustesse des fonctions de réseaux d'apprentissage profond [1, 11, 12, 16]. Nous réalisons également des expérimentations avec des bases de données de vision par ordinateur afin de mettre en évidence nos arguments.

2 Définition

Notons F la fonction d'un réseau d'apprentissage profond, associant des entrées dans un espace Ω (typiquement un espace tensoriel) avec une décision douce pour classification (typiquement un vecteur dans $\mathbb{R}^C$ où C est le nombre de classes dans le problème de classification traité). Notons $\|\cdot\|$ une norme pouvant s'appliquer sur les entrées ou les sorties de F (typiquement $\mathcal{L}_2$ ou $\mathcal{L}_\infty$).

Nous définissons la robustesse de la fonction F en prenant en compte le domaine R et le rayon r sur lesquels elle s'applique. Plus précisément :

Definition 1. *Une fonction F est dite α-**robuste** sur un domaine R et pour un rayon $r > 0$, noté $F \in \mathrm{Robust}_\alpha(R, r)$, si :*

$$\forall \mathbf{x} \in R, \forall \varepsilon \text{ s.t. } \|\varepsilon\| < r, \|F(\mathbf{x}+\varepsilon) - F(\mathbf{x})\| \leq \alpha\|\varepsilon\| \ . \quad (1)$$

Dans la suite de cet article, nous utiliserons souvent $R = T$, où T sont les entrées de l'ensemble d'entraînement.

Pour un R fixé, nous définissons aussi : $\alpha_{\mathrm{lim}}(F, r) = \inf\{\alpha : F \in \mathrm{Robust}_\alpha(\mathrm{r})\}$. Avec cette notion, il apparaît plus clairement que la robustesse est un compromis entre la pente de la fonction, mesurée localement par α et le rayon r. Considérons par exemple la fonction sigmoide $\sigma : x \mapsto \frac{1}{1+\exp(-x)}$ et $R = \{-10, 10\}$. Sur la figure 2 est représentée l'évolution de $r \mapsto \alpha_{\mathrm{lim}}(\sigma, r)$. Nous observons que la fonction sigmoide dispose d'une constante de Lipschitz presque nulle autour des deux points -10 et 10, ce qui correspondra plus tard à la robustesse de la fonction. En s'éloignant la constante de Lipschitz locale augmente, ce qui correspondra plus tard à la frontière entre les classes.

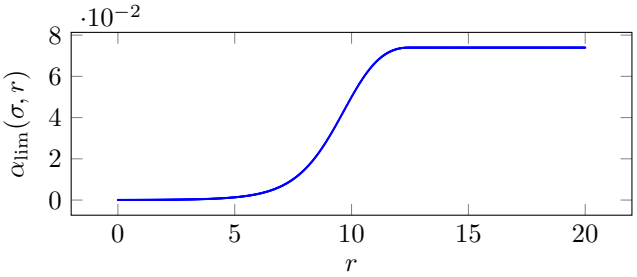

FIGURE 2 – Évolution de $r \mapsto \alpha_{\mathrm{lim}}(\sigma, r)$.

2.1 Relation avec la constante de Lipschitz

Notons qu'il est immédiat par construction que si F est Lipschitz avec une constante α, alors $F \in \mathrm{Robust}_\alpha(+\infty)$. La réciproque n'est en général pas vérifiée. Par exemple considérons un problème de classification où l'on cherche à séparer deux points distincts $\mathbf{x}$ et $\mathbf{x}'$ en utilisant l'hyperplan médiateur. Alors la fonction de décision F n'a pas de constante de Lipschitz (à cause de la pente "infinie" autour de l'hyperplan médiateur) et pourtant $\alpha_{\mathrm{lim}}(F, \|\mathbf{x} - \mathbf{y}\|_2/2) = 0$. Il s'agit là d'un résultat fondamental, car la meilleure constante de Lipschitz atteignable dépend du jeu de données (comme illustré sur la figure 1), alors que la définition proposée permet d'ajuster le compromis entre α et r de sorte que le jeu de données devienne compatible avec la définition.

3 Expérimentations

Nous considérons ici quatre méthodes proposées dans la littérature pour augmenter la robustesse des réseaux profonds : la méthode de Parseval (P) [1], la méthode $\mathcal{L}_2$ non-expensive (L2NN) [16], la méthode des Laplaciens de graphes (L) [11] et une méthode d'apprentissage adversaire (PGD) [12]. La table 1 est un résumé des relations entre chaque méthode et les critères de robustesse proposés.

Méthode	Domaine R	Pente α	Rayon r	Norme
P	Ω	Oui	Non	$\mathcal{L}_2 + \mathcal{L}_\infty$
L2NN	Ω	Oui	Non	$\mathcal{L}_2$
L	T	Oui	Oui	$\mathcal{L}_2$ + cos
PGD	T augmenté	Non	Oui	$\mathcal{L}_\infty$

TABLE 1 – Prise en compte, par diverses méthodes de la littérature, des différents facteurs du critère de robustesse introduit.

Nous réalisons plusieurs expérimentations pour évaluer la pertinence de notre critère de robustesse. Dans les résumés, nous notons V le réseau standard (non robuste). Remarquons que les comparaisons effectuées ne sont pas nécessairement justes, dans la mesure où certaines méthodes introduisent des paramètres supplémentaires pouvant potentiellement affecter la robustesse des architectures. Par exemple, les réseaux entraînés avec PGD sont généralement plus grands car l'ensemble d'apprentissage est enrichi. L2NN utilise quant à lui des réseaux

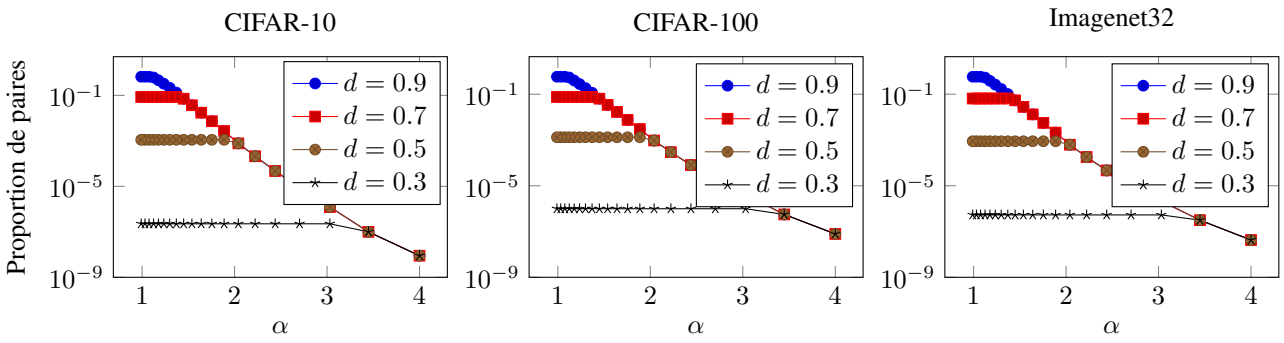

FIGURE 3 – Évolution de la proportion de paires d'exemples de classes différentes incompatibles avec la définition proposée de robustesse, en fonction de α, et pour plusieurs valeurs de d.

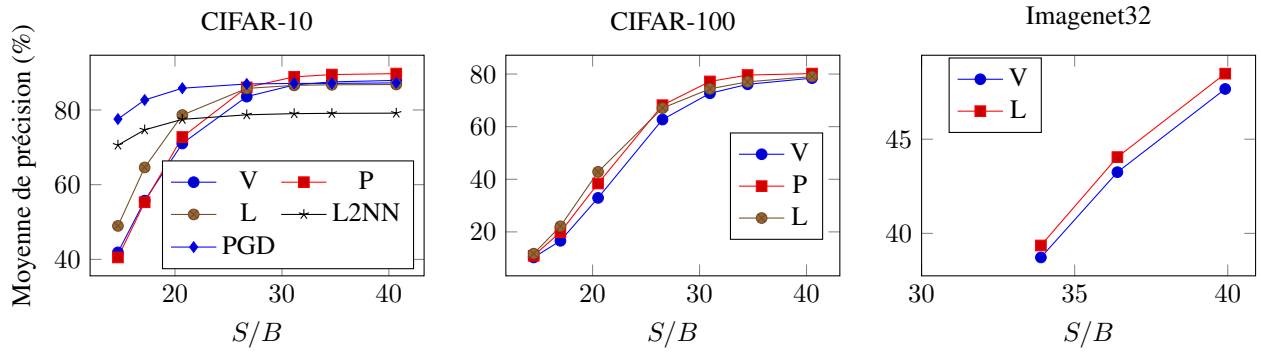

FIGURE 4 – Précision moyenne sur le test lorsque les entrées sont additionnées d'un bruit Gaussien, pour diverses bases de données et méthodes, et en fonction du rapport signal sur bruit (S/B).

sans résidus...

Sur la figure 3, nous montrons le rapport entre le nombre de paires d'exemples de classes différentes dans l'ensemble d'entraînement, à distance au plus d, incompatibles avec le critère α-robuste, et le nombre total de paires d'exemples dans des classes différentes (pour la norme $\mathcal{L}_\infty$). Notons que d devrait être interprété comme environ $2r$ dans notre critère de robustesse. Comme pour la figure 1, nous considérons ici que la fonction du réseau doit produire en sortie des vecteurs indicateurs de la classe de l'exemple considéré. Notons que pour chaque choix de d la courbe est au départ constante, puis elle commence ensuite à décroître. Dans la partie plate, toutes les paires à distance au plus d sont non α-robustes. La compatibilité s'accroît ensuite lorsque α augmente. Il est intéressant de noter que le nombre de paires à distance au plus d s'écroule très vite lorsque d diminue. Pour $d = 0.3$, il devient déjà extrêmement rare de trouver des paires incompatibles avec un objectif de Lipschitz donné, quel qu'il soit. Notons qu'il n'existe à notre connaissance aucune méthode profonde dans la littérature capable d'atteindre une robustesse sur un rayon de 0.15 pour les bases de données considérées.

Sur la figure 5, nous montrons l'évolution de $r \mapsto \alpha_{\text{lim}}(r)$ pour diverses méthodes. Chaque point est une estimation obtenue en générant 1000 bruits additifs pour 100 exemples choisis arbitrairement dans la base de donnée CIFAR-10. Notons que pour toutes les méthodes α augmente en fonction de r jusqu'à atteindre une valeur limite. La méthode non robuste (V) est celle atteignant la saturation au plus vite. Ceci s'explique par deux facteurs : 1) des transitions rapides peuvent arriver très proches des exemples, et 2) le réseau parvient à associer chaque exemple de l'ensemble d'apprentissage avec le vecteur indicateur correspondant, étant donc à une distance de 1 (rappelons que nous utilisons $\mathcal{L}_\infty$) les uns des autres en sortie. Au contraire les autres méthodes atteignent systématiquement une valeur limite plus petite et pour un r plus grand, indiquant que les transitions ne sont pas aussi nettes, notamment car les exemples ne sont pas associés aux vecteurs caractéristiques correspondants (l'apprentissage ne parvient pas à atteindre parfaitement cette association à cause des régularisations utilisées). Le fait que pour P et L la saturation arrive pour des plus grandes valeurs de r suggère que la distance moyenne entre les exemples et la frontière de décision est augmentée par rapport à V. Les réseaux L2NN et PGD atteignent des valeurs limites très faibles. Nous observons une transition pour PGD aux alentours de $r = 0.3$ alors que L2NN reste quasi-constant. Ceci est expliqué par le fait que la méthode L2NN impose une constante de Lipschitz très forte (avec la norme $\mathcal{L}_2$) partout dans le domaine de définition : une conséquence est que la fonction du réseau est quasi-linéaire entre les exemples d'ap-

prentissage. Cette propriété a pour conséquence une grande incompatibilité avec l'ensemble d'apprentissage, comme illustré dans les scores sans déviation donnés dans la table 2.

Nous comparons ensuite les méthodes grâce à un protocole défini récemment [7]. Les résultats, résumés dans la table 2, montrent que PGD atteint les meilleurs compromis entre précision et robustesse, conformément à nos attentes suite à la figure 5. Finalement, nous montrons dans la figure 4 la robustesse des méthodes considérées à un bruit Gaussien, en fonction du rapport signal à bruit. Nous observons la même hiérarchie que dans la table 2 (notons que certains résultats sont manquants car nous ne disposons pas des codes permettant de les faire tourner). Notre lecture des résultats est qu'un comportement sain pour un réseau consiste à disposer d'une pente très faible autour des exemples et plus importantes à la frontière entre les classes, soit un compromis équilibré entre r et α.

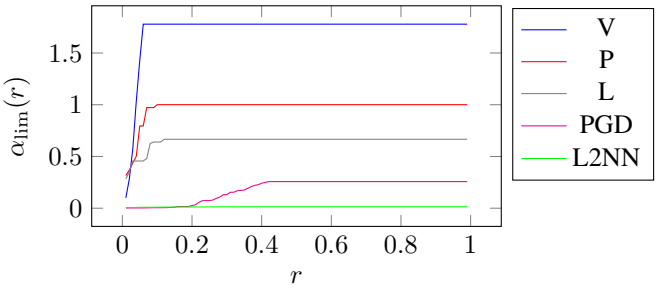

FIGURE 5 – Éstimations de $\alpha_{\text{lim}}(r)$ obtenues pour différents rayons r sur l'ensemble d'apprentissage et pour la norme $\mathcal{L}_\infty$.

Dataset	V	P	L	PGD	L2NN
Référence	11.9%	**10.2%**	13.2%	12.8%	20.9%
Base corrompue	31.6%	30.5%	31.3%	**18.8%**	28.5%
Erreur relative	100	103	92	**30**	39

TABLE 2 – Précision sur la base de donnée CIFAR-10 pour 15 corruptions différentes des entrées [7]. L'erreur relative pour une méthode X est obtenue par la formule $100(Corrompue(X) - Reference(X))/(Corrompue(V) - Reference(V))$.

4 Conclusion

Nous avons introduit une nouvelle définition formelle de robustesse pour une boule de rayon r autour de l'ensemble d'apprentissage. Nous avons discuté des liens entre cette définition et des méthodes existantes. Nos expérimentations semblent indiquer un lien étroit entre cette mesure de robustesse et les performances des méthodes lorsqu'elles sont évaluées sur des entrées corrompues. Dans un avenir proche, nous souhaitons analyser de façon plus précise l'importance de la pente des fonctions à la frontière de décision, et proposer de nouvelles méthodes de régularisation permettant d'accroître la robustesse d'architectures d'apprentissage profond.

Références

[1] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks : Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863, 2017.

[2] L. Engstrom, A. Ilyas, and A. Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv :1807.10272*, 2018.

[3] N. Ford, J. Gilmer, N. Carlini, and D. Cubuk. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv :1901.10513*, 2019.

[4] X. Gastaldi. Shake-shake regularization. *arXiv preprint arXiv :1705.07485*, 2017.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv :1412.6572*, 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[7] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[8] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5) :359–366, 1989.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv :1611.01236*, 2016.

[11] C. E. R. K. Lassance, V. Gripon, and A. Ortega. Laplacian networks : Bounding indicator function smoothness for neural networks robustness. *Open Review*, 2019.

[12] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018.

[13] S. Mallat. Understanding deep convolutional networks. *Phil. Trans. R. Soc. A*, 374(2065) :20150203, 2016.

[14] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[15] M. Pezeshki, L. Fan, P. Brakel, A. Courville, and Y. Bengio. Deconstructing the ladder network architecture. In *International Conference on Machine Learning*, pages 2368–2376, 2016.

[16] H. Qian and M. N. Wegman. L2-nonexpansive neural networks. In *International Conference on Learning Representations*, 2019.

[17] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv :1312.6199*, 2013.

[18] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system : Bridging the gap between human and machine translation. *arXiv preprint arXiv :1609.08144*, 2016.