Pay attention to your loss: understanding misconceptions about 1-Lipschitz neural networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Lipschitz constrained networks have gathered considerable attention in deep learning community, with usages ranging from Wasserstein distance estimation to the training of certifiably robust classifiers. However they remain commonly considered as less accurate, and their properties in learning are still not fully understood. In this paper we clarify the matter: when it comes to classification 1-Lipschitz neural networks enjoy several advantages over their unconstrained counterpart. First, we show that these networks are as accurate as classical ones, and can fit arbitrarily difficult boundaries. Then, relying on a robustness metric which reflects operational needs we characterize the most robust classifier: the WGAN discriminator. Next, we show that 1-Lipschitz neural networks generalize well under milder assumptions. Finally, we show that hyper-parameters of the loss are crucial for controlling the accuracy-robustness trade-off. We conclude that they exhibit appealing properties to pave the way toward provably accurate, and provably robust neural networks.

1 Introduction

1-Lipschitz neural networks have drawn a great attention in the last decade, with motivation ranging from adversarial robustness to Wasserstein distance computation. In the following we denote by **LipNet1** the class of 1-Lipschitz neural networks, by **AllNet** the class of neural networks without constraints on their Lipschitz constant (i.e classical neural networks). They are defined more in details in Section 2.

Roughly speaking, the Lipschitz constant of neural networks quantifies how much their outputs can change when inputs are perturbed. When this constant is high, as it is often the case for neural networks of AllNet , they become vulnerable to adversarial attacks (see [1, 2] and references therein): a carefully chosen small noise added to the inputs, usually imperceptible, can change the class prediction. One possible defense against adversarial attacks is to constrain the network to be 1-Lipschitz (in LipNet1) [3], which provides provable robustness guarantees, together with an improvement of generalization [4] and interpretability of the model [5]. LipNet1 networks are also used to estimate Wasserstein distance, thanks to Kantorovich-Rubinstein duality in the seminal work of WGAN [6].

Despite their competitiveness with networks of AllNet on medium scale problems [7, 8], they still suffer from misconceptions. A belief commonly invoked against networks of LipNet1 is that they are less expressive. “Lipschitz-based approaches suffer from some representational limitations that may prevent them from achieving higher levels of performance and being applicable to more complicated problems” [9].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Although this claim seems rational at first glance, the link between Lipschitz constant and expressiveness is not trivial. While there is an obvious lack of expressiveness for regression tasks, this intuition fades when it comes to classification. Indeed, every AllNet network $g : \mathbb{R}^n \to \mathbb{R}^K$ is L-Lipschitz for some (generally unknown) $L > 0$. Then $f = \frac{1}{L} g$ is a 1-Lipschitz neural network with the same decision boundary, since prediction $\arg\max_k g_k$ is invariant by positive rescaling of the logits. In particular, f have the same accuracy and also the same robustness to adversarial attacks as g. We illustrate this empirically by **training a LipNet1 network until it reaches 99.96% accuracy on CIFAR-100 with random labels** (see Appendix I).

We demonstrate that LipNet1 networks are theoretically better grounded than AllNet networks when it comes to classification, through our threefold contribution on Expressiveness (Section 3), Robustness (Section 4) and Generalization (Section 5).

First, in Section 3 we confirm that LipNet1 are as expressive as AllNet networks for classification, and can learn arbitrary complex decision boundary. We show that hyper-parameters of the loss are of crucial importance, and control the ability to fit properly the train set.

Then, in Section 4 we show that accuracy and robustness are often antipodal objectives. We characterize the robustness of the highest accuracy LipNet1 classifier: it is achieved by the Signed Distance Function (Definition 5 in Appendix A). We also characterize the classifier of highest certifiable robustness, and we show it corresponds to the dual potential of Wasserstein-1 distance (i.e the discriminator of a WGAN [6]).

Finally, in Section 5 we show that LipNet1 benefit from several generalization guarantees. They are consistent estimators: contrary to AllNet , we prove that their train loss will converge to test loss as the size of the train set increases. Moreover, we show that LipNet1 classifiers with margin are PAC-learnable [10]: it provides bounds on the number of train examples required to reach a targeted test accuracy. Interestingly, this bound is independent of the architecture size, which allows to train enormous LipNet1 networks without risking overfitting.

2 Notations and experimental setting

The core of the paper mainly deal with binary classification over $\mathbb{R}^n$ with label set $\mathcal{Y} = \{-1, +1\}$. Let (X, Y) be a random variable taking values on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subset \mathbb{R}^n$ is assumed to be a compact set. Such a pair follows the joint distribution $\mathbb{P}_{XY}$, defined on the space of probability measures $\mathcal{P}(\mathcal{X} \times \mathcal{Y})$. The marginal distribution of X is denoted by $\mathbb{P}_X \in \mathcal{P}(\mathcal{X})$ and its support by $\operatorname{supp} \mathbb{P}_X$. We suppose the observation of a sample $(x_1, y_1), \ldots, (x_p, y_p)$ i.i.d. with common law $\mathbb{P}_{XY}$, and the goal is to learn a classifier $c : \mathcal{X} \to \mathcal{Y}$ modeling the optimal Bayes classifier $\arg\max_{y \in \mathcal{Y}} \mathbb{P}_{Y|X}(y|x)$. P (resp. Q) denotes the input distribution of label $+1$ (resp. -1).

The Lipschitz constant $\mathrm{Lip}(f)$ of a function $f : \mathbb{R}^n \to \mathbb{R}^K$ is defined as the smallest $L \geq 0$ such that for all $x, z \in \mathbb{R}^n$ we have $\|f(x) - f(z)\| \leq L\|x - z\|$. In the rest of the paper, we focus on euclidean norm $\|\cdot\|$ for vectors and spectral norm $\|\cdot\|_2$ for matrices. The set of L-Lipschitz functions over $\mathcal{X} \subset \mathbb{R}^n$ with image in $\mathbb{R}^K$ is denoted $\mathrm{Lip}_L(\mathcal{X}, \mathbb{R}^K)$.

Definition 1 (Class of AllNet networks)
AllNet denotes the set of unconstrained neural networks. It includes any feed forward network of fixed depth (without recurrent mechanisms) using affine layers (including convolutions and batch normalization) with weight matrices $W_1, W_2, \ldots W_d$ and activation function σ (such as ReLU, sigmoid, tanh, etc). No constraint is enforced on their Lipschitz constant during training. They benefit from universal approximation theorem in $C(\mathcal{X}, \mathbb{R}^K)$, a classical result of literature [11].

Definition 2 (Class of LipNet1 networks)
LipNet1 denotes the set of feed forward neural networks f defined as in Theorem 3 of Anil et al. [12]: bound on first weight matrix $\|W_1\|_{2\to\infty} \leq 1$ with $\|\cdot\|_{2\to\infty}$ norm (see [13] for definition), bound $\|W_i\|_\infty \leq 1$ on other weight matrices $i \geq 2$, and GroupSort2 activation function. They fulfill $Lip(f) \leq 1$. They also benefit from an universal approximation theorem in $Lip_1(\mathcal{X}, \mathbb{R})$ with respect to uniform convergence. Note that $Lip_L(\mathcal{X}, \mathbb{R}^K) = \{Lf \mid f \in Lip_1(\mathcal{X}, \mathbb{R}^K)\}$ so LipNet1 can be used to approximate functions in $Lip_L(\mathcal{X}, \mathbb{R}^K)$.

In practice authors of [12] noticed that using orthogonal weight matrices (i.e $W_i^T W = I$) yielded the best results. All our experiments use the Deel.Lip[1] library [8], following ideas of [12]. The networks

[1] `https://github.com/deel-ai/deel-lip` distributed under MIT License (MIT).

use 1) orthogonal weight matrices and 2) GroupSort2 activations [12]. Orthogonalization is enforced using Spectral normalization [14] and Björck algorithm [15]. These networks belong to LipNet1 by construction (see Appendix D for our choice of architecture and relevant related work).

LipNet1 networks provide robustness radius certificates against adversarial attacks [16]. Computing these certificates is straightforward and do not increase runtime, contrary to methods based on bounding boxes or abstract interpretation [17, 18, 19, 20]. There is no need for adversarial training [21] that fails to produce guarantees, or for randomized smoothing [22] which is costly.

Confusingly, any network of AllNet has a finite Lipschitz constant, but computing it is NP-hard [23]. Only a loose upper bound can be cheaply estimated: $\text{Lip}(f) \leq \text{Lip}(\sigma)^d \Pi_{i=1}^d \|W_i\|_2$ using the property that $\text{Lip}(f_d \circ f_{d-1} \circ \ldots \circ f_1) \leq \Pi_{i=1}^d \text{Lip}(f_i)$. In practice this bound is often too high to provide meaningful certificates and besides, AllNet networks have usually very small robustness radius [1].

Definition 3 (Adversarial Attack)
For any classifier $c : \mathcal{X} \to \mathcal{Y}$, any $x \in \mathbb{R}^n$, consider the following problem:

$$\epsilon = \min_{\delta \in \mathbb{R}^n} \|\delta\| \text{ such that } c(x+\delta) \neq c(x) \tag{1}$$

δ is an adversarial attack, $x + \delta$ is an adversarial example, and ϵ is the robustness radius of x.

Property 1 (Local Robustness Certificates [16])**.** *For any $f \in$ LipNet1 the robustness radius ϵ of binary classifier sign $\circ f$ at example x verifies $\epsilon \geq |f(x)|$.*

Losses: The Binary Cross Entropy (BCE) loss (also called logloss) is among the most popular choices of loss within the deep learning community. Let $f : \mathbb{R}^n \to \mathbb{R}$ a neural network. For an example $x \in \mathbb{R}^n$ with label $y \in \mathcal{Y}$, and $\sigma(x) = \frac{1}{1+\exp(-x)}$ the logistic function mapping logits to probabilities, the BCE is written $\mathcal{L}_\tau^{bce}(f(x), y) = -\log \sigma(y\tau f(x))$, with temperature scaling parameter[2] $\tau > 0$. Note that $\mathcal{L}_\tau^{bce}(f(x), y) = \mathcal{L}_1^{bce}(\tau f(x), y)$ so we can equivalently tune τ or the Lipschitz constant L. We show in Section 5.1 that **for LipNet1 the temperature τ allow to control the generalization gap**. We also consider the Hinge loss $\mathcal{L}_m^H(f(x), y) = \max(0, m - yf(x))$ with margin $m > 0$, as used in [3] for LipNet1 networks training.

We focus on binary classification for readability and clarity purposes; however we prove in Appendices A.2 and E that **the following theoretical results generalize to the multi-class case**, as done in experiments. The proofs of all propositions can be found in appendix.

3 1-Lipschitz classifiers are expressive

In this section, we show that LipNet1 are as powerful as any other classifier, like their unconstrained counterpart. In particular, when classes are separable they can achieve 100% accuracy.

3.1 Boundary decision fitting

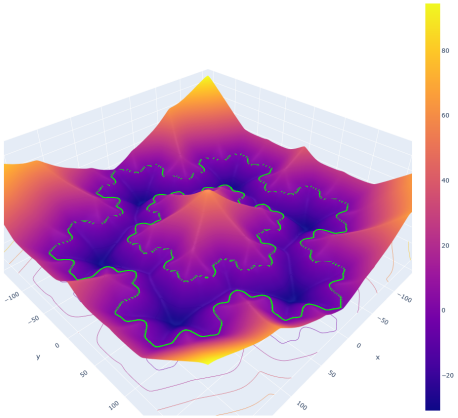
Proposition 1 (**Lipschitz Binary classification**)**.** *For any binary classifier $c : \mathcal{X} \to \mathcal{Y}$ with closed pre-images ($c^{-1}(\{y\})$ is a closed set) there exists a 1-Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$ such that sign$(f(x)) = c(x)$ on $\mathcal{X}$ and such that $\|\nabla_x f\| = 1$ almost everywhere (w.r.t Lebesgue measure).*

The level-sets of a $\text{Lip}_1(\mathcal{X}, \mathbb{R}^K)$ functions (and especially the decision boundary) can be arbitrarily complex: restraining classifiers to $\text{Lip}_1(\mathcal{X}, \mathbb{R})$ does not affect the classification power.

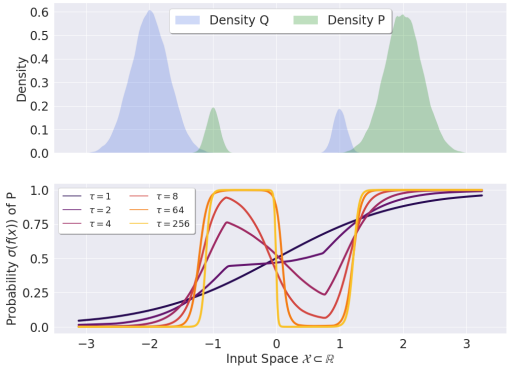
Corollary 1 (**Separable classes implies zero error**)**.** *The distributions P and Q are ϵ-separated if it exists $\epsilon > 0$ such that the distance between supp P and supp Q exceeds ϵ. Then there exists a network $f \in$ LipNet1 such that* ***error*** $E(\text{sign} \circ f) := \mathbb{E}_{(x,y)\sim\mathbb{P}_{XY}}[\mathbb{1}\{\text{sign}(f(x)) \neq y\}] = 0$.

Some empirical studies show that indeed most datasets classes are separable [24] such as CIFAR10 or MNIST. Furthermore, even if the classes are not separable, functions of LipNet1 can nonetheless approximate the optimal Bayes classifier. Lipschitz constraint is not a constraint on the shape of the boundary (Figure 1a), but on the slope of the landscape of f.

[2] This hyper-parameter of the loss defaults to $\tau = 1$ in most frameworks such as Tensorflow or Pytorch.



(a) **Complex Decision Boundary** ∂. We chose ∂ as the fourth iteration of Von Koch Snowflake. We chose P as the interior ring, while the center and the exterior corresponds to Q. We train a LipNet1 network with MSE to fit the SDF (Definition 5 in Appendix) ground truth (160 000 pixels), until MAE is inferior to 1. It proves empirically that LipNet1 networks can handle very sharp (almost fractal) decision boundary.



(b) **Importance of τ in BCE**. We train a LipNet1 network with BCE and different values for τ. We chose a toy example where P and Q are a Gaussian mixtures with two modes of weights 0.9 and 0.1. We highlight the different shapes of the minimizer $\sigma \circ f$ as function of τ. **High values of τ leads to better fitting, whereas for lower τ the small weights Gaussian of the mixture are treated as noise and ignored**.

Figure 1

3.2 Understanding why LipNet1 are often perceived as not expressive

LipNet1 networks cannot reach zero loss with BCE: this may explain why they are perceived as not expressive enough. Yet the minimizer of BCE exists and is well defined.

Proposition 2 (**BCE minimization for L-Lipschitz functions**). *Let $\mathcal{X} \subset \mathbb{R}^n$ be a compact and $\tau > 0$. Then the minimum of Equation 2 is attained.*

$$f^\tau \in \arg \min_{f \in Lip_1(\mathcal{X},\mathbb{R})} \mathbb{E}_{(x,y)\sim\mathbb{P}_{XY}}[\mathcal{L}_\tau^{bce}(f(x), y)] \tag{2}$$

Moreover the LipNet1 networks will not suffer of vanishing gradient of the loss (see Appendix F).

Machine learning practitioners are mostly interested in the minimization of the empirical risk. However one cannot guarantees that the minimizer of BCE will maximize accuracy (see Figure 1b). Besides, the minimum f^τ of Equation 2 for some τ is not scale invariant: $\frac{1}{\tau} f^\tau$ is not is not in general the minimum of Equation 2 for $\tau = 1$. The minimizer of BCE is not necessarily the minimizer of the error. BCE is notoriously a differentiable proxy of the error $E(\text{sign} \circ f)$, and as $\tau \to \infty$ we get asymptotically closer to maximum empirical accuracy. Bigger value for τ might ultimately lead to overfitting, playing the same role as the Lipschitz constant L (see Figure 1b).

The implicit parameter $\tau = 1$ of the loss is partially responsible of the poor accuracy of LipNet1 networks in literature, and not by any means the hypothesis space LipNet1 itself. This can be observed in practice : when temperature τ (resp. margin m) of crossentropy (resp. hinge loss) is correctly adjusted a small LipNet1 CNN can reach a competitive **88.2% validation accuracy on the CIFAR10 dataset** (results synthetized and discussed in Figure 3) *without* batch normalization or dropout. Conversely, AllNet networks are roughly equivalent to learning a LipNet1 network with $\tau \to \infty$: without regularization or data augmentation, such a network can always reach 100% train accuracy without generalization guarantees.

4 1-Lipschitz classifiers are certifiably robust

Is there a trade-off between accuracy and robustness ? Although the existence of a trade-off between accuracy and robustness is commonly admitted, some works argue that "Robustness is not inherently at odds with accuracy"[24]. We propose an unified consideration by stating that for a given train accuracy, robustness can be maximized up to a certain point, but allowing a lower train accuracy

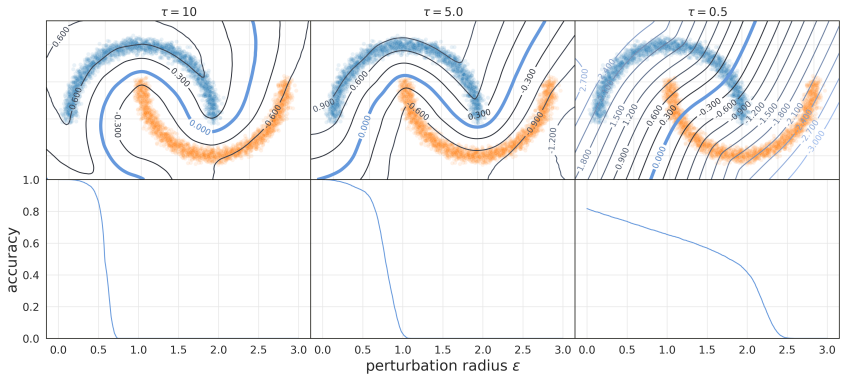


Figure 2: **Accuracy-robustness tradeoff:** Each network is optimal with respect to a certain criterion. The leftmost network is the most accurate at robustness radius $\epsilon \leq 0.3$, the rightmost maximizes the MCR at the cost of low clean accuracy. The center network corresponds to a compromise.

helps achieving a higher robustness. Finally one must keep in mind that this trade-off lives in the shade of generalization (see Section 5).

4.1 Improving the robustness of the maximally accurate classifier

The Signed Distance Function [25] (SDF) (see Definition 5 in Appendix B) associated to the frontier ∂ of Bayes classifier b is the 1-Lipschitz function that provides the largest certificates among the classifiers of maximum accuracy. Moreover those certificates are exactly equal to the distance of adversarial samples. Iterative gradient based attacks (see [26] and references therein) can succeed in one step: far from being a weakness, this may improve interpretability of the model [27, 28, 29].

Corollary 2. *For the SDF(b), the bound of Property 1 is tight: $\epsilon = |f(x)|$. In particular $\delta = -f(x)\nabla_x f(x)$ is guaranteed to be an adversarial attack. The risk is the smallest possible. There is no classifier with the same risk and better certificates. Said otherwise the SDF(b) is the solution to:*

$$\begin{aligned} \max_{f \in Lip_1(\mathbb{R}^n,\mathbb{R})} \min_{x \in \mathcal{X}} \min_{\substack{\delta \in \mathbb{R}^n \\ sign(f(x+\delta)) \neq sign(f(x))}} \|\delta\| \\ \text{such that } f \in \underset{g \in Lip_1(\mathbb{R}^n,\mathbb{R})}{\arg\min} \; E(sign \circ g) \end{aligned} \tag{3}$$

The SDF(b) cannot be explicitly constructed since it relies on the (unknown) optimal Bayes classifier.

4.2 Improving the accuracy of the maximally robust classifier

On the opposite side, we exhibit a family of classifiers with lower accuracy but with higher certifiable robustness. We insist that the quantity of interest is the *certifiable robustness* $|f(x)|$ and not the *true empirical robustness* ϵ (which can be higher). The former is computed exactly and freely, while the later is a difficult problem for which only upper bounds returned by attacks are available. In the literature the robustness is only evaluated on well classified examples. The certificate can be both interpreted as a form of "confidence" of the network, and as the minimal perturbations required to switch the class. Hence, we shall weight negatively this certificate for the examples that are misclassified since confidence in presence of errors is worse. For this reason, we propose in Definition 4 a new metric called the Mean Certifiable Robustness (MCR).

Definition 4 (**Mean Certifiable Robustness – MCR**)
For any function $f : \mathcal{X} \to \mathbb{R} \in$ LipNet1 we define its weighted mean certifiable robustness $\mathcal{R}_{(P,y)}(f)$ on class P with label y as:

$$\mathcal{R}_{(P,y)}(f) = \mathbb{E}_{x\sim P}[\mathbb{1}\{yf(x) > 0\}|f(x)|] + \mathbb{E}_{x\sim P}[-\mathbb{1}\{yf(x) < 0\}|f(x)|] = \mathbb{E}_{x\sim P} yf(x) \tag{4}$$

We can readily see from the definition that the classifier with highest MCR for class P is the constant classifier $f = y \times \infty$. The interest of this notion arises when we consider minimizing the loss function $\mathcal{L}^W(f(x), y) = -\mathcal{R}_{(P,y)}(f) = -yf(x)$, i.e when looking for classifier with the highest MCR.

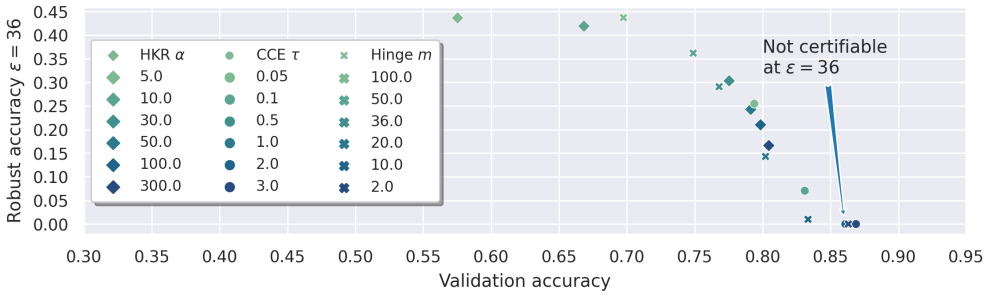


Figure 3: **Accuracy-Robustness trade-off on CIFAR10 with Hinge, HKR and Categorical Cross Entropy (CCE) hyper-parameters.** Overall, for a given network architecture, a Pareto front appears between clean accuracy and robust accuracy. We move along it by tuning the parameters of each loss. We trained small LipNet1 *CNNs* (0.4M params) with basic data augmentation (see appendix K for detailed experimental setting).

Property 2 (Wasserstein classifiers (i.e WGAN discriminators) are optimally robust). *The minimum of $\mathcal{L}^W(f(x), y)$ over P and Q is the Wasserstein-1 distance [30] between P and Q according to the Kantorovich-Rubinstein (KR) duality:*

$$\max_{f \in Lip_1(\mathcal{X},\mathbb{R})} \mathcal{R}_{(P,+1)}(f) + \mathcal{R}_{(Q,-1)}(f) = \min_{f \in Lip_1(\mathbb{R}^n,\mathbb{R})} \mathbb{E}_{\mathbb{P}_{XY}}[\mathcal{L}^W(f(x), y)] = \mathcal{W}_1(P, Q) \tag{5}$$

Even though the minimizer of $\mathcal{L}_W(f(x), y)$ can have low accuracy, it has the highest MCR. Interestingly, the minimizer f^* of 5 is invariant by translation[3]: $f^* - T$ is also a minimum for any $T \in \mathbb{R}$. When $T \to \infty$ (resp. $-\infty$) the classifier has 100% recall on Q (resp. P), and 0% on P (resp. Q). Does it always exist T^* with 100% accuracy overall ? Sadly, even when the P and Q have disjoint support, the answer is no. We precise this empirical observation of [8] in Proposition 3.

Proposition 3 (WGAN discriminators are weak classifiers). *For every $\frac{1}{2} \geq \epsilon > 0$ there exist distributions P and Q with disjoint supports in $\mathbb{R}$ such that for any optimum f of equation 5, the error of classifier sign $\circ f$ is superior to $\frac{1}{2} - \epsilon$.*

4.3 Controlling the accuracy/robustness tradeoff

Interestingly, observe that $\mathcal{L}^{bce}_\tau(f(x), y) = \log 2 - \frac{y\tau f(x)}{2} + \mathcal{O}(\tau^2 f^2(x))$ so when $\tau \to 0$ we get:

$$\min_{f \in \text{Lip}_1(\mathcal{X},\mathbb{R})} \frac{4}{\tau} \left(\mathbb{E}_{(x,y)\sim P_{XY}}[\mathcal{L}^{bce}_\tau(f(x), y)] - \log 2 \right) = -\mathcal{W}_1(P, Q)$$

In the limit of small temperatures, the BCE minimizer essentially behave like the classifier of highest MCR (see Figure 3 and Appendix H). Similarly, the HKR loss $\mathcal{L}^{hkr}$ introduced in [8] for LipNet1 training allows fine grained control of the accuracy-robustness tradeoff:

$$\mathcal{L}^{hkr}_{m,\alpha}(f(x), y) = \mathcal{L}^W(f(x), y) + \alpha \mathcal{L}^H_m(f(x), y) = -yf(x) + \alpha \max(0, m - yf(x)) \tag{6}$$

We recover $\mathcal{W}_1$ behavior for $\alpha = 0$, and hinge $\mathcal{L}^H_m$ behavior for $\alpha \to \infty$, in a fashion that reminds the role of τ for $\mathcal{L}^{bce}$. In conclusion, these last two sections demonstrate that restraining networks to be in LipNet1 does not impact the classification capabilities while providing certificates of robustness; however for these networks the loss parameters play an important role in this trade-off (Figure 3).

5 1-Lipschitz classifiers have generalization guarantees

In this section we explore the statistical and optimization properties of LipNet1 networks, and we prove the assumption of [31] that "adjusting the Lipschitz constant of a feed-forward neural network controls how well the model will generalise to new data".

[3] It is also invariant by dilatation: any *finite* upper bound L can be chosen (see Appendix G).

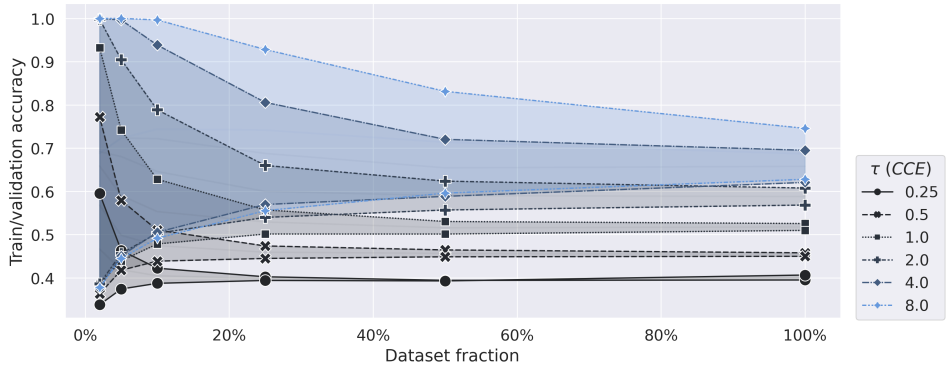


Figure 4: **Link between LipNet1 and generalization gap, dataset size and crossentropy temperature.** We train a CNN on different fractions of the CIFAR10 train set (2%, 5%, 10%, 25%, 50% and 100% on x-axis) with different values of temperature τ (highlighted by different colors). Train (resp. validation) accuracy form the upper (resp. lower) bound of each envelope. As τ increases, more samples are required to reduce the generalization gap. Conversely, training a LipNet1 network with small τ is equivalent to train a Lipschitz network with small L: the network generalizes well but the accuracy reach a plateau (under-fitting). See appendix J for detailed experimental setting.

5.1 Consistency

LipNet1 class enjoys another remarkable property since it is a Glivenko-Cantelli class: minimizers of Lipschitz losses are consistent estimators. In other words, as the size of the training set increases, the training loss becomes a proxy for the test loss: LipNet1 neural networks will not overfit in the limit of (very) large sample sizes.

Proposition 4 (**Train Loss is a proxy of Test Loss**). *Let $\mathbb{P}_{XY}$ a probability measure on $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{X} \subset \mathbb{R}^n$ is a bounded set. Let $(x_i, y_i)_{1 \leq i \leq p}$ be a sample of p iid random variables with law $\mathbb{P}_{XY}$. Let $\mathcal{L}$ be a Lipschitz loss function over $\mathbb{R} \times \mathcal{Y}$. Let:*

$$\mathcal{E}_p(f) = \frac{1}{p} \sum_{i=1}^{p} \mathcal{L}(f(x_i), y_i) \text{ and } \mathcal{E}_\infty(f) = \mathbb{E}_{(x,y) \sim \mathbb{P}_{XY}}[\mathcal{L}(f(x), y)] \tag{7}$$

Then the empirical loss $\mathcal{E}_p(f)$ converges to the test loss $\mathcal{E}_\infty(f)$ (taking the limit $p \to \infty$):

$$\min_{f \in Lip_L(\mathcal{X}, \mathbb{R})} \mathcal{E}_p(f) \xrightarrow{a.s} \min_{f \in Lip_L(\mathcal{X}, \mathbb{R})} \mathcal{E}_\infty(f) \tag{8}$$

It is an other flavor of the bias-variance trade-off in learning. Thanks to Corollary 1 we know the LipNet1 class does not suffer of bias, while the generalization gap (i.e the variance) can be made as small as we want by increasing the size of the training set (see Figure 4). This result may seem obvious[4], but we emphasize **this property is not shared by AllNet networks** (see Proposition 9 in Appendix C.2).

5.2 Understanding why unconstrained networks are prone to overfitting

Surprisingly, on AllNet networks, minimization of BCE leads to uncontrolled growth of Lipschitz constant and saturation of the predicted probabilities. This property is an impediment to generalization results.

Proposition 5 (**Optimizing BCE over AllNet leads to divergence**). *Let f_t be a sequence of neural networks, that minimizes the BCE over a non trivial training set (at least two different examples with different labels) of size p, i.e:*

$$\lim_{t \to \infty} \frac{1}{p} \sum_{i=1}^{p} \mathcal{L}_\tau(f_t(x_i), y_i) = 0 \tag{9}$$

[4]Most practitioners take for granted that bigger train set ensures generalization.

Let L_t the Lipschitz constant of f_t. Then $\lim_{t\to\infty} L_t = +\infty$. There is at least one weight matrix W such that $\lim_{t\to\infty} \|W_t\| = +\infty$. Furthermore the predicted probabilities are saturated:

$$\lim_{t\to\infty} \sigma(f_t(x_i)) \in \{0, 1\} \tag{10}$$

This issue is especially important since Lipschitz constant and adversarial vulnerabilities are related [32]. The predicted probability $\sigma(f(x))$ will either be 0 or 1 (regardless of the train set), which do not carry any useful information on the true confidence of the classifier, especially in the *out-of-distribution* setting.

Example 1. *Consider a classification task on $\mathbb{R}$ with linearly separable inputs $\{-1, 1\}$ and labels $\{-1, 1\}$. We use an affine model $f(x) = Wx + b$ for the logits (with $W \in \mathbb{R}$ and $b \in \mathbb{R}$) (one-layer neural network). It exists $\bar{W}, \bar{b}$ such that f achieves 100% accuracy. However, as noticed in [33] (Section 4.3.2) the BCE loss will not be zero. The minimization occurs only with the diverging sequence of parameters $(\lambda\bar{W}, \lambda\bar{b})$ as $\lambda \to \infty$. It turns out the infimum is not a minimum!*

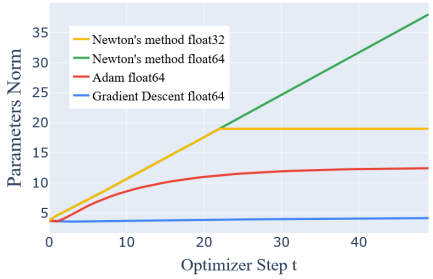


Figure 5

Even on toy example 1 with a trivial model, the minimization problem is ill-defined. Without weight regularization the minimizer can not be attained. Furthermore there is an issue of vanishing gradients with BCE : first order methods struggle to saturate the logits of AllNet networks, whereas second order methods in *float64* diverge as expected. The poor properties of the optimizer, and the rounding errors in 32 bits floating point arithmetic, have greatly contributed to the caveat of BCE minimization remaining mostly unnoticed by the community.

5.3 Lipschitz classifiers are PAC learnable

Hinge loss $\mathcal{L}_m^H$ and HKR loss $\mathcal{L}^{hkr}$ benefit from Proposition 4. The certificate $|f(x)|$ can be understood as confidence. Hence, we are interested in a classifier that makes a decision only if the prediction is above some threshold $m > 0$, while $|f(x)| < m$ can be understood as examples x for which the classifier is unsure: the label may be flipped using attacks of norm $\epsilon \leq m$. In this setting, we fall back to PAC learnability [10]: this theory gives bounds on the number of train samples required to guarantee that the test error will fall below some threshold $0 \leq e < \frac{1}{2}$ with probability at least $1 > \beta \geq 0$, through the use of Vapnik Chervonenkis (VC) dimension bounds [34].

Proposition 6 (1-Lipschitz Functions with margin are PAC learnable). *Assume P and Q have bounded support $\mathcal{X}$. Let $m > 0$ the margin. Let $\mathcal{C}^m(\mathcal{X}) = \{c_f^m : \mathcal{X} \to \{-1, \perp, +1\}, f \in Lip_1(\mathcal{X}, \mathbb{R})\}$ be the hypothesis class defined as follow.*

$$c_f^m(x) = \begin{cases} +1 & \text{if } f(x) \geq m \\ -1 & \text{if } f(x) \leq -m \\ \perp & \text{otherwise, meaning “}f\text{ doesn't feel confident”} \end{cases} \tag{11}$$

Let $\mathfrak{B}$ be the unit ball. Then the VC dimension of $\mathcal{C}^m$ is finite:

$$(\frac{1}{m})^n \frac{vol(\mathcal{X})}{vol(\mathfrak{B})} \leq VC_{dim}(\mathcal{C}^m(\mathcal{X})) \leq (\frac{3}{m})^n \frac{vol(\mathcal{X})}{vol(\mathfrak{B})} \tag{12}$$

Interestingly if the classes are ϵ separable ($\epsilon > 0$), choosing $m = \epsilon$ guarantees that 100% accuracy is reachable. Prior over the separability of the input space is turned into VC bounds over the space of hypothesis. When $m = 0$ the VC dimension of space $\mathcal{C}^m(\mathcal{X})$ becomes infinite and the class is not PAC learnable anymore: the training error will not converge to test error in general, regardless of the size of the training set. It is not a contradiction with Proposition 4: error $E(c_f^m(x))$ lacks continuity w.r.t $f(x)$ so it is not a consistent estimator.

This VC bound is *architecture independent* which contrasts with the rest of litterature on AllNet networks. Practically, it means that the LipNet1 network architecture can be chosen as big as

Properties		AllNet network	LipNet1 network
Fit any boundary		yes [11]	yes (Proposition 1)
Robustness certificates		no	yes (Property 1)
Consistent estimator		no (App C.2)	yes (Proposition 4, Figures 1b, 4)
Gradients		exploding or vanishing	preserved for GNP (App F)
VC dimension bounds		architecture dependent [35]	when $m > 0$ (Proposition 6)
BCE $\mathcal{L}_{\tau}^{bce}$	minimizer	ill-defined $L_t \to \infty$ (Proposition 5)	attained (Proposition 2)
	remark	vanishing gradient (Ex 1)	L or τ must be tuned (Figure 3)
Wasserstein $\mathcal{L}^{W}$	minimizer	ill-defined $L_t \to \infty$	attained, robust (Property 2)
	remark	diverges during training	weak classifier (Proposition 3)
Hinge $\mathcal{L}_{m}^{H}$	minimizer	attained	attained
	remark	no guarantees on margin	m must be tuned
HKR $\mathcal{L}_{m,\alpha}^{hkr}$	minimizer	ill-defined $L_t \to \infty$	accuracy-robustness tradeoff
	remark	diverges during training	α and m must be tuned (Figure 3)

Table 1: Summary of notable results and the contributions.

we want without risking overfitting, as long as the margin m is chosen appropriately. Proposition 7 also provides an architecture dependant bound for LipNet1 networks:

Proposition 7 (**VC dimension of LipNet1 neural networks**). *Let $f_\theta : \mathbb{R}^n \to \mathbb{R}$ a LipNet1 neural network with parameters $\theta \in \Theta$, with **GroupSort2** activation functions, and a total of W neurons. Let $\mathcal{H} = \{sign f_\theta | \theta \in \Theta\}$ the hypothesis class spanned by this architecture. Then:*

$$VC_{dim}(\mathcal{H}) = \mathcal{O}\left((n+1)2^W\right) \tag{13}$$

In literature, tighter VC dimension bounds for neural networks exist, but they assume element-wise activation function [35]. This hypothesis does not apply to GroupSort2 which is known to be more expressive [36], however we believe that this preliminary result can be strengthened.

6 Related work

LipNet1 networks parametrization benefit from a rich literature (see Appendix D) to enforce the Lipschitz constraint in various layers [37, 38, 6, 39, 40, 14, 41, 42, 43, 44] such as activation functions, affine layers, attention layers or recurrent units. Residual connections are also Lipschitz (see Appendix F). **Gradient Norm Preserving networks** avoid the vanishing gradients [3, 45] phenomenon to which the LipNet1 networks are prone, by using orthogonal matrices in affine layers. This justifies the *"orthogonal neural network"* terminology [46, 47, 48]. ReLU based Lipschitz networks suffer from expressiveness issues [12], and activation functions like GroupSort [12, 36] (a special case of Householder reflection [49, 50]) have been proposed in replacement. **Orthogonal kernels** are still an active research area [51, 52, 53, 54, 3, 55, 56, 57]. They are used in normalizing flows [58], ensemble methods [59], reinforcement learning [60] or graph neural networks [61]. The optimization over the group of orthogonal matrices (known as Stiefel manifold) has been extensively studied in [62], and algorithms suitable for deep learning are detailed in [63, 64, 64, 65, 66, 67, 68, 69].

Generalization bounds for general Lipschitz classifiers are given in [70, 71, 72]. Links between adversarial robustness, large margins classifiers and optimization bias are studied in [73, 74, 16, 75]. The importance of the loss in adversarial robustness is studied in [76]. See Appendix C.5.

7 Conclusion

In this paper, we challenged the common belief that constraining Lipschitz constant degrades the classification performance of neural networks. We proved that LipNet1 networks exhibit numerous attractive properties (see Table 1 in summary): they provide robustness radius certificates without restrictions on their expressive power. They benefit from generalization guarantees. We showed that the hidden parameters of the loss allow to control generalization gap and certifiable robustness.

References

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[2] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

[3] Qiyang Li, Saminul Haque, Cem Anil, James Lucas, Roger B Grosse, and Jörn-Henrik Jacobsen. Preventing gradient attenuation in lipschitz constrained convolutional networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, Cambridge, MA, 2019. MIT Press.

[4] J. Sokolic, R. Giryes, G. Sapiro, and M. R. D. Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017.

[5] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

[6] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[7] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863. PMLR, 2017.

[8] Mathieu Serrurier, Franck Mamalet, Alberto González-Sanz, Thibaut Boissin, Jean-Michel Loubes, and Eustasio del Barrio. Achieving robustness in classification using optimal transport with hinge regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[9] Todd Huster, Cho-Yu Jason Chiang, and Ritu Chadha. Limitations of the lipschitz constant as a defense against adversarial examples. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 16–29. Springer, 2018.

[10] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[11] Mohamad H Hassoun et al. *Fundamentals of artificial neural networks*. MIT press, 1995.

[12] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR, 2019.

[13] Joshua Cape, Minh Tang, and Carey E Priebe. The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics. *The Annals of Statistics*, 47(5):2405–2439, 2019.

[14] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[15] Åke Björck and Clazett Bowie. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971.

[16] Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 31, pages 6541–6550. Curran Associates, Inc., 2018.

[17] Fabian Latorre, Paul Rolland, and Volkan Cevher. Lipschitz constant estimation of neural networks via sparse polynomial optimization. In *International Conference on Learning Representations*, 2019.

[18] Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning*, pages 5276–5285. PMLR, 2018.

[19] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *International Conference on Learning Representations*, 2018.

[20] Bohang Zhang, Tianle Cai, Zhou Lu, Di He, and Liwei Wang. Towards certifying l-infinity robustness using neural networks with l-inf-dist neurons. In *International Conference on Machine Learning*, pages 12368–12379. PMLR, 2021.

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[22] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 2019.

[23] Kevin Scaman and Aladin Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3839–3848, 2018.

[24] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *Advances in Neural Information Processing Systems*, 33, 2020.

[25] Mikael Rousson and Nikos Paragios. Shape priors for level set representations. In *European Conference on Computer Vision*, pages 78–92. Springer, 2002.

[26] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

[27] Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017.

[28] Richard Tomsett, Amy Widdicombe, Tianwei Xing, Supriyo Chakraborty, Simon Julier, Prudhvi Gurram, Raghuveer Rao, and Mani Srivastava. Why the failure? how adversarial examples can provide insights for interpretable machine learning. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 838–845. IEEE, 2018.

[29] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[30] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[31] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

[32] Kamil Nar, Orhan Ocal, S Shankar Sastry, and Kannan Ramchandran. Cross-entropy loss and low-rank features have responsibility for adversarial examples. *arXiv preprint arXiv:1901.08360*, 2019.

[33] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[34] VN Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

[35] Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20:63–1, 2019.

[36] Ugo Tanielian and Gerard Biau. Approximating lipschitz continuous functions with groupsort neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 442–450. PMLR, 2021.

[37] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, volume 30, pages 5767–5777. Curran Associates, Inc., 2017.

[38] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

[39] Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NIPS*, 2016.

[40] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *International Conference on Machine Learning*, pages 5562–5571. PMLR, 2021.

[41] Kyle Helfrich, Devin Willmott, and Qiang Ye. Orthogonal recurrent neural networks with scaled cayley transform. In *International Conference on Machine Learning*, pages 1969–1978. PMLR, 2018.

[42] N. Benjamin Erichson, Omri Azencot, Alejandro Queiruga, Liam Hodgkinson, and Michael W. Mahoney. Lipschitz recurrent neural networks. In *International Conference on Learning Representations*, 2021.

[43] Yujia Huang, Huan Zhang, Yuanyuan Shi, J Zico Kolter, and Anima Anandkumar. Training certifiably robust neural networks with efficient local lipschitz bounds. *Advances in Neural Information Processing Systems*, 34, 2021.

[44] Zhenyu Zhu, Fabian Latorre, Grigorios G Chrysos, and Volkan Cevher. Controlling the complexity and lipschitz constant improves polynomial nets. *arXiv preprint arXiv:2202.05068*, 2022.

[45] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? *Advances in Neural Information Processing Systems*, 31, 2018.

[46] Bartłomiej Stasiak and Mykhaylo Yatsymirskyy. Fast orthogonal neural networks. In *International Conference on Artificial Intelligence and Soft Computing*, pages 142–149. Springer, 2006.

[47] Shuai Li, Kui Jia, Yuxin Wen, Tongliang Liu, and Dacheng Tao. Orthogonal deep neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 43(4):1352–1368, 2019.

[48] Jiahao Su, Wonmin Byeon, and Furong Huang. Scaling-up diverse orthogonal convolutional networks with a paraunitary framework. *arXiv preprint arXiv:2106.09121*, 2021.

[49] Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. Efficient orthogonal parametrisation of recurrent neural networks using householder reflections. In *International Conference on Machine Learning*, pages 2401–2409. PMLR, 2017.

[50] Sahil Singla, Surbhi Singla, and Soheil Feizi. Improved deterministic l2 robustness on cifar-10 and cifar-100. In *International Conference on Learning Representations*, 2021.

[51] Alexander V Gayer and Alexander V Sheshkus. Convolutional neural network weights regularization via orthogonalization. In *Twelfth International Conference on Machine Vision (ICMV 2019)*, volume 11433, page 1143326. International Society for Optics and Photonics, 2020.

[52] Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and Stella X Yu. Orthogonal convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11505–11515, 2020.

[53] Sheng Liu, Xiao Li, Yuexiang Zhai, Chong You, Zhihui Zhu, Carlos Fernandez-Granda, and Qing Qu. Convolutional normalization: Improving deep convolutional network robustness and training. *arXiv preprint arXiv:2103.00673*, 2021.

[54] El Mehdi Achour, François Malgouyres, and Franck Mamalet. Existence, stability and scalability of orthogonal convolutional neural networks. *arXiv preprint arXiv:2108.05623*, 2021.

[55] Asher Trockman and J Zico Kolter. Orthogonalizing convolutional layers with the cayley transform. In *International Conference on Learning Representations*, 2021.

[56] Sahil Singla and Soheil Feizi. Skew orthogonal convolutions. In *International Conference on Machine Learning*, pages 9756–9766. PMLR, 2021.

[57] Tan Yu, Jun Li, YUNFENG CAI, and Ping Li. Constructing orthogonal convolutions in an explicit manner. In *International Conference on Learning Representations*, 2021.

[58] Leonard Hasenclever, Jakub M Tomczak, Rianne van den Berg, and Max Welling. Variational inference with orthogonal normalizing flows. In *Bayesian Deep Learning, NIPS 2017 workshop*, 2017.

[59] Peyman Sheikholharam Mashhadi, Sławomir Nowaczyk, and Sepideh Pashami. Parallel orthogonal deep neural network. *Neural Networks*, 140:167–183, 2021.

[60] Florin Gogianu, Tudor Berariu, Mihaela Rosca, Claudia Clopath, Lucian Busoniu, and Razvan Pascanu. Spectral normalization for deep reinforcement learning: an optimisation perspective. In *Proceedings of the International Conference on Machine Learning (ICML)*. JMLR. org, 2021.

[61] George Dasoulas, Kevin Scaman, and Aladin Virmaux. Lipschitz normalization for self-attention layers with application to graph neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, pages 2456–2466. PMLR, 2021.

[62] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[63] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pages 1120–1128. PMLR, 2016.

[64] Stephanie L Hyland and Gunnar Rätsch. Learning unitary operators with help from u (n). In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[65] Mario Lezcano-Casado and David Martınez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *International Conference on Machine Learning*, pages 3794–3803. PMLR, 2019.

[66] Lei Huang, Xianglong Liu, Bo Lang, Adams Yu, Yongliang Wang, and Bo Li. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[67] Pierre Ablin and Gabriel Peyré. Fast and accurate optimization on the orthogonal manifold without retraction. *arXiv preprint arXiv:2102.07432*, 2021.

[68] Iordanis Kerenidis, Jonas Landman, and Natansh Mathur. Classical and quantum algorithms for orthogonal neural networks. *arXiv preprint arXiv:2106.07198*, 2021.

[69] Krzysztof Choromanski, David Cheikhi, Jared Davis, Valerii Likhosherstov, Achille Nazaret, Achraf Bahamou, Xingyou Song, Mrugank Akarte, Jack Parker-Holder, Jacob Bergquist, et al. Stochastic flows and geometric optimization on the orthogonal group. In *International Conference on Machine Learning*, pages 1918–1928. PMLR, 2020.

[70] Ulrike von Luxburg and Olivier Bousquet. Distance-based classification with lipschitz functions. *J. Mach. Learn. Res.*, 5:669–695, 2004.

[71] Lee-Ad Gottlieb, Aryeh Kontorovich, and Robert Krauthgamer. Efficient classification for metric data. *IEEE Transactions on Information Theory*, 60(9):5750–5759, 2014.

[72] Peter L Bartlett, Dylan J Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6241–6250, 2017.

[73] Chris Finlay, Jeff Calder, Bilal Abbasi, and Adam Oberman. Lipschitz regularized deep neural networks generalize and are adversarially robust. *arXiv preprint arXiv:1808.09540*, 2018.

[74] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. *arXiv preprint arXiv:1810.00113*, 2018.

[75] Fartash Faghri, Cristina Vasconcelos, David J. Fleet, Fabian Pedregosa, and Nicolas Le Roux. Bridging the gap between adversarial robustness and optimization bias, 2021.

[76] Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. In *International Conference on Learning Representations*, 2019.

[77] Jon Wellner A.W. van der vaart. *Weak Convergence and Empirical Processes*. Springer-Verlag New York, 1996.

[78] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

[79] Stanisław J Szarek. Metric entropy of homogeneous spaces. *Banach Center Publications*, 43(1):395–410, 1998.

[80] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[81] Emerson León and Günter M Ziegler. Spaces of convex n-partitions. In *New Trends in Intuitive Geometry*, pages 279–306. Springer, 2018.

[82] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. In *Ninth International Conference on Artificial Neural Networks ICANN 99*. IET, 1999.

[83] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[84] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Proofs in appendix, experiments in the paper.
   (b) Did you describe the limitations of your work? [Yes] We talked about the difficulty of training the networks.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Mostly theoretical contribution.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Assumptions are stated in each Proposition or Theorem.
   (b) Did you include complete proofs of all theoretical results? [Yes] Yes (in appendix).
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Experimental setting is given in appendix, and the code is in supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Yes, in the core paper for toy experiments and in appendix K for large scale experiments.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Toy experiments were run only once. CIFAR-10 experiments like in Figure3 and Figure 4 involved multiple runs (with different values for hyper-parameters). All the runs are reported (no cherry-picking), and the results show a coherent trend. Other experiments (see Appendix) showed our experiments are robust with respect to random seed, but discussing stability of the training of LipNet1 is out of the scope of this article.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See appendix L.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Yes for Deel-Lip library we used.
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Our code is available in supplementary.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]