On Lipschitz Regularization of Convolutional Layers using Toeplitz Matrix Theory

Alexandre Araujo, Benjamin Negrevergne, Yann Chevaleyre, Jamal Atif
PSL, Université Paris-Dauphine,
CNRS, LAMSADE, MILES Team, Paris, France
alexandre.araujo@dauphine.eu

Abstract

This paper tackles the problem of Lipschitz regularization of Convolutional Neural Networks. Lipschitz regularity is now established as a key property of modern deep learning with implications in training stability, generalization, robustness against adversarial examples, etc. However, computing the exact value of the Lipschitz constant of a neural network is known to be NP-hard. Recent attempts from the literature introduce upper-bounds to approximate this constant that are either efficient but loose or accurate but computationally expensive. In this work, by leveraging the theory of Toeplitz matrices, we introduce a new upper-bound for convolutional layers that is both tight and easy to compute. Based on this result we devise an algorithm to train Lipschitz regularized Convolutional Neural Networks.

Introduction

The last few years have witnessed a growing interest in Lipschitz regularization of neural networks, with the aim of improving their generalization (Bartlett, Foster, and Telgarsky 2017), their robustness to adversarial attacks (Tsuzuku, Sato, and Sugiyama 2018; Farnia, Zhang, and Tse 2019), or their generation abilities (*e.g* for GANs: Miyato et al. 2018; Arjovsky, Chintala, and Bottou 2017). Unfortunately computing the exact Lipschitz constant of a neural network is NP-hard (Virmaux and Scaman 2018) and in practice, existing techniques such as Virmaux and Scaman (2018); Fazlyab et al. (2019) or Latorre, Rolland, and Cevher (2020) are difficult to implement for neural networks with more than one or two layers, which hinders their use in deep learning applications.

To overcome this difficulty, most of the work has focused on computing the Lipschitz constant of *individual layers* instead. The product of the Lipschitz constant of each layer is an upper-bound for the Lipschitz constant of the entire network, and it can be used as a surrogate to perform Lipschitz regularization. Since most common activation functions (such as ReLU) have a Lipschitz constant equal to one, the main bottleneck is to compute the Lipschitz constant of the underlying linear application which is equal to its maximal singular value. The work in this line of research mainly relies on the celebrated iterative algorithm by Golub and Van der Vorst (2000) used to approximate the maximum singular value of a linear function. Although generic and accurate, this technique is also computationally expensive, which impedes its usage in large training settings.

In this paper we introduce a new upper-bound on the largest singular value of convolution layers that is both tight and easy to compute. Instead of using the power method to iteratively approximate this value, we rely on Toeplitz matrix theory and its links with Fourier analysis. Our work is based on the result (Gray et al. 2006) that an upper-bound on the singular value of Toeplitz matrices can be computed from the inverse Fourier transform of the characteristic sequence of these matrices. We first extend this result to doubly-block Toeplitz matrices (*i.e.*, block Toeplitz matrices where each block is Toeplitz) and then to convolutional operators, which can be represented as stacked sequences of doubly-block Toeplitz matrices. From our analysis immediately follows an algorithm for bounding the Lipschitz constant of a convolutional layer, and by extension the Lipschitz constant of the whole network. We theoretically study the approximation of this algorithm and show experimentally that it is more efficient and accurate than competing approaches.

Finally, we illustrate our approach on adversarial robustness. Recent work has shown that empirical methods such as adversarial training (AT) offer poor generalization (Schmidt et al. 2018), and can be improved by applying Lipschitz regularization (Farnia, Zhang, and Tse 2019). To illustrate the benefit of our new method, we train a large, state-of-the-art Wide ResNet architecture with Lipschitz regularization and show that it offers a significant improvement over adversarial training alone, and over other methods for Lipschitz regularization. In summary, we make the three following contributions:

1. We devise an upper-bound on the singular values of the operator matrix of convolutional layers by leveraging Toeplitz matrix theory and its links with Fourier analysis.
2. We propose an efficient algorithm to compute this upper-bound which enables its use in the context of Convolutional Neural Networks.
3. We use our method to regularize the Lipschitz constant of neural networks for adversarial robustness and show that it offers a significant improvement over AT alone.

Related Work

A popular technique for approximating the maximal singular value of a matrix is the power method (Golub and Van der Vorst 2000), an iterative algorithm which yields a good approximation of the maximum singular value when the algorithm is able to run for a sufficient number of iterations.

Yoshida and Miyato (2017); Miyato et al. (2018) have used the power method to normalize the spectral norm of each layer of a neural network, and showed that the resulting models offered improved generalization performance and generated better examples when they were used in the context of GANs. Farnia, Zhang, and Tse 2019 built upon the work of Miyato et al. (2018) and proposed a power method specific for convolutional layers that leverages the deconvolution operation and avoids the computation of the gradient. They used it in combination with adversarial training. In the same vein, Gouk et al. (2018) demonstrated that regularized neural networks using the power method also offered improvements over their non-regularized counterparts. Furthermore, Tsuzuku, Sato, and Sugiyama (2018) have shown that a neural network can be more robust to some adversarial attacks, if the prediction margin of the network (*i.e.*, the difference between the first and the second maximum logit) is higher than a minimum threshold that depends on the global Lipschitz constant of the network. Building on this observation, they use the power method to compute an upper-bound on the global Lipschitz constant, and maximize the prediction margin during training. Finally, Virmaux and Scaman (2018) have used automatic differentiation combined with the power method to compute a tighter bound on the global Lipschitz constant of neural networks. Despite a number of interesting results, using the power method is expensive and results in prohibitive training times.

Other approaches to regularize the Lipschitz constant of neural networks have been proposed by Sedghi, Gupta, and Long (2019) and Singla and Feizi (2019). The method of Sedghi, Gupta, and Long (2019) exploits the properties of circulant matrices to approximate the maximal singular value of a convolutional layer. Although interesting, this method results in a loose approximation of the maximal singular value of a convolutional layer. Furthermore, the complexity of their algorithm is dependent on the convolution input which can be high for large datasets such as ImageNet. More recently, Singla and Feizi (2019) have successfully bounded the operator norm of the Jacobian matrix of a convolution layer by the Frobenius norm of the reshaped kernel. This technique has the advantage to be very fast to compute and to be independent of the input size but it also results in a loose approximation.

To build robust neural networks, Cisse et al. (2017) and Li et al. (2019) have proposed to constrain the Lipschitz constant of neural networks by using orthogonal convolutions. Cisse et al. (2017) use the concept of *parseval tight frames*, to constrain their networks. Li et al. (2019) built upon the work of Cisse et al. (2017) to propose an efficient construction method of orthogonal convolutions. Also, recent work (Fazlyab et al. 2019; Latorre, Rolland, and Cevher 2020) has proposed a tight bound on the Lipschitz constant of the full network with the use of semi-definite programming. These works are theoretically interesting but lack scalability (*i.e.*, the bound can only be computed on small networks).

Finally, in parallel to the development of the results in this paper, we discovered that Yi (2020) have studied the asymptotic distribution of the singular values of convolutional layers by using a related approach. However, this author does not investigate the robustness applications of Lipschitz regularization.

A Primer on Toeplitz and Block Toeplitz Matrices

In order to devise a bound on the Lipschitz constant of a convolution layer as used by the Deep Learning community, we study the properties of doubly-block Toeplitz matrices. In this section, we first introduce the necessary background on Toeplitz and block Toeplitz matrices, and introduce a new result on doubly-block Toeplitz matrices.

Toeplitz matrices and block Toeplitz matrices are well-known types of structured matrices. A Toeplitz matrix (respectively a block Toeplitz matrix) is a matrix in which each scalar (respectively block) is repeated identically along diagonals.

An $n \times n$ Toeplitz matrix $\mathbf{A}$ is fully determined by a two-sided sequence of scalars: $\{a_h\}_{h\in N}$, whereas an $nm \times nm$ block Toeplitz matrix $\mathbf{B}$ is fully determined by a two-sided sequence of blocks $\{\mathbf{B}_h\}_{h\in N}$, where $N = \{-n+1, \dots, n-1\}$ and where each block $\mathbf{B}_h$ is an $m \times m$ matrix.

$$\mathbf{A} = \begin{pmatrix} a_0 & a_{-1} & \cdots & a_{-n+1} \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & \ddots & a_0 & a_{-1} \\ a_{n-1} & \cdots & a_1 & a_0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} \mathbf{B}_0 & \mathbf{B}_{-1} & \cdots & \mathbf{B}_{-n+1} \\ \mathbf{B}_1 & \mathbf{B}_0 & \ddots & \vdots \\ \vdots & \ddots & \mathbf{B}_0 & \mathbf{B}_{-1} \\ \mathbf{B}_{n-1} & \cdots & \mathbf{B}_1 & \mathbf{B}_0 \end{pmatrix}.$$

Finally, a doubly-block Toeplitz matrix is a block Toeplitz matrix in which each block is itself a Toeplitz matrix. In the remainder, we will use the standard notation $(\,\cdot\,)_{i,j\in\{0,\dots,n-1\}}$ to construct (block) matrices. For example, $\mathbf{A} = (a_{j-i})_{i,j\in\{0,\dots,n-1\}}$ and $\mathbf{B} = (\mathbf{B}_{j-i})_{i,j\in\{0,\dots,n-1\}}$.

Bound on the Singular Value of Toeplitz and Block Toeplitz Matrices

A standard tool for manipulating (block) Toeplitz matrices is the use of Fourier analysis. Let $\{a_h\}_{h\in N}$ be the sequence of coefficients of the Toeplitz matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and let $\{\mathbf{B}_h\}_{h\in N}$ be the sequence of $m \times m$ blocks of the block Toeplitz matrix $\mathbf{B}$. The complex-valued function $f(\omega) = \sum_{h\in N} a_h e^{\mathbf{i}h\omega}$ and the matrix-valued function $F(\omega) = \sum_{h\in N} \mathbf{B}_h e^{\mathbf{i}h\omega}$ are the *inverse Fourier transforms* of the sequences $\{a_h\}_{h\in N}$ and $\{\mathbf{B}_h\}_{h\in N}$, with $\omega \in \mathbb{R}$. From these two functions, one can recover these two sequences using the standard Fourier transform:

$$a_h = \frac{1}{2\pi}\int_0^{2\pi} e^{-\mathbf{i}h\omega} f(\omega) d\omega \quad \mathbf{B}_h = \frac{1}{2\pi}\int_0^{2\pi} e^{-\mathbf{i}h\omega} F(\omega) d\omega. \tag{1}$$

From there, similarly to the work done by Gray et al. (2006) and Gutiérrez-Gutiérrez, Crespo et al. (2012), we can define

an operator $\mathbf{T}$ mapping integrable functions to matrices:

$$\mathbf{T}(g) \triangleq \left(\frac{1}{2\pi} \int_0^{2\pi} e^{-\mathbf{i}(i-j)\omega} g(\omega)\, d\omega \right)_{i,j \in \{0,\dots,n-1\}}. \quad (2)$$

Note that if f is the inverse Fourier transform of $\{a_h\}_{h \in N}$, then $\mathbf{T}(f)$ is equal to $\mathbf{A}$. Also, if F is the inverse Fourier transform of $\{\mathbf{B}_h\}_{h \in N}$ as defined above, then the integral in Equation 2 is matrix-valued, and thus $\mathbf{T}(F) \in \mathbb{R}^{mn \times mn}$ is the block matrix $\mathbf{B}$. Now, we can state two known theorems which upper-bound the maximal singular value of Toeplitz and block Toeplitz matrices with respect to their generating functions. In the rest of the paper, we refer to $\sigma_1(\,\cdot\,)$ as the maximal singular value.

Theorem 1 (Bound on the singular values of Toeplitz matrices)**.** *Let $f : \mathbb{R} \to \mathbb{C}$, be continuous and 2π-periodic. Let $\mathbf{T}(f) \in \mathbb{R}^{n \times n}$ be a Toeplitz matrix generated by the function f, then:*

$$\sigma_1\left(\mathbf{T}(f)\right) \leq \sup_{\omega \in [0,2\pi]} |f(\omega)|. \quad (3)$$

Theorem 1 is a direct application of Lemma 4.1 in Gray et al. (2006) for real Toeplitz matrices.

Theorem 2 (Bound on the singular values of Block Toeplitz matrices (Gutiérrez-Gutiérrez, Crespo et al. 2012))**.** *Let $F : \mathbb{R} \to \mathbb{C}^{m \times m}$ be a matrix-valued function which is continuous and 2π-periodic. Let $\mathbf{T}(F) \in \mathbb{R}^{mn \times mn}$ be a block Toeplitz matrix generated by the function F, then:*

$$\sigma_1\left(\mathbf{T}(F)\right) \leq \sup_{\omega \in [0,2\pi]} \sigma_1(F(\omega)). \quad (4)$$

Bound on the Singular Value of Doubly-Block Toeplitz Matrices

We extend the reasoning from Toeplitz and block Toeplitz matrices to doubly-block Toeplitz matrices (*i.e.*, block Toeplitz matrices where each block is also a Toeplitz matrix). A doubly-block Toeplitz matrix can be generated by a function $f : \mathbb{R}^2 \to \mathbb{C}$ using the 2-dimensional inverse Fourier transform. For this purpose, we define an operator $\mathbf{D}$ which maps a function $f : \mathbb{R}^2 \to \mathbb{C}$ to a doubly-block Toeplitz matrix of size $nm \times nm$. For the sake of clarity, the dependence of $\mathbf{D}(f)$ on m and n is omitted. Let $\mathbf{D}(f) = (\mathbf{D}_{i,j}(f))_{i,j \in \{0,\dots,n-1\}}$ where $\mathbf{D}_{i,j}$ is defined as:

$$\mathbf{D}_{i,j}(f) = \left(\frac{1}{4\pi^2} \int_{\Omega^2} e^{-\mathbf{i}\psi} f(\omega_1, \omega_2)\, d(\omega_1, \omega_2) \right)_{k,l \in \{0,\dots,m-1\}} \quad (5)$$

where $\Omega = [0, 2\pi]$ and $\psi = (i-j)\omega_1 + (k-l)\omega_2$.

We are now able to combine Theorem 1 and Theorem 2 to bound the maximal singular value of doubly-block Toeplitz matrices with respect to their generating functions.

Theorem 3 (Bound on the Maximal Singular Value of a Doubly-Block Toeplitz Matrix)**.** *Let $\mathbf{D}(f) \in \mathbb{R}^{nm \times nm}$ be a doubly-block Toeplitz matrix generated by the function f, then:*

$$\sigma_1\left(\mathbf{D}(f)\right) \leq \sup_{\omega_1, \omega_2 \in [0,2\pi]^2} |f(\omega_1, \omega_2)| \quad (6)$$

where the function $f : \mathbb{R}^2 \to \mathbb{C}$, is a multivariate trigonometric polynomial of the form:

$$f(\omega_1, \omega_2) \triangleq \sum_{h_1 \in N} \sum_{h_2 \in M} d_{h_1,h_2} e^{\mathbf{i}(h_1\omega_1 + h_2\omega_2)}, \quad (7)$$

where d_{h_1,h_2} is the $h_2{}^{th}$ scalar of the $h_1{}^{th}$ block of the doubly-Toeplitz matrix $\mathbf{D}(f)$, and where $M = \{-m+1, \dots, m-1\}$.

Bound on the Singular Values of Convolutional Layers

From now on, without loss of generality, we will assume that $n = m$ to simplify notations. It is well known that a discrete convolution operation with a 2d kernel applied on a 2d signal is equivalent to a matrix multiplication with a doubly-block Toeplitz matrix (Jain 1989). However, in practice, the signal is most of the time 3-dimensional (RGB images for instance). We call the channels of a signal *channels in* denoted cin. The input signal is then of size $cin \times n \times n$. Furthermore, we perform multiple convolutions of the same signal which corresponds to the number of channels the output will have after the operation. We call the channels of the output *channels out* denoted $cout$. Therefore, the kernel, which must take into account *channels in* and *channels out*, is defined as a 4-dimensional tensor of size: $cout \times cin \times s \times s$.

The operation performed by a 4-dimensional kernel on a 3d signal can be expressed by the concatenation (horizontally and vertically) of doubly-block Toeplitz matrices. Hereafter, we bound the singular value of multiple vertically stacked doubly-block Toeplitz matrices which corresponds to the operation performed by a 3d kernel on a 3d signal.

Theorem 4 (Bound on the maximal singular value of stacked Doubly-block Toeplitz matrices)**.** *Consider doubly-block Toeplitz matrices $\mathbf{D}(f_1), \dots, \mathbf{D}(f_{cin})$ where $f_i : \mathbb{R}^2 \to \mathbb{C}$ is a generating function. Construct a matrix $\mathbf{M}$ with $cin \times n^2$ rows and n^2 columns, as follows:*

$$\mathbf{M} \triangleq \left(\mathbf{D}^\top(f_1), \dots, \mathbf{D}^\top(f_{cin})\right)^\top. \quad (8)$$

Then, with f_i a multivariate polynomial of the same form as Equation 7, we have:

$$\sigma_1\left(\mathbf{M}\right) \leq \sup_{\omega_1, \omega_2 \in [0,2\pi]^2} \sqrt{\sum_{i=1}^{cin} |f_i(\omega_1, \omega_2)|^2}. \quad (9)$$

In order to prove Theorem 4, we have generalized the famous Widom identity (Widom 1976) expressing the relation between Toeplitz and Hankel matrices to doubly-block Toeplitz matrices.

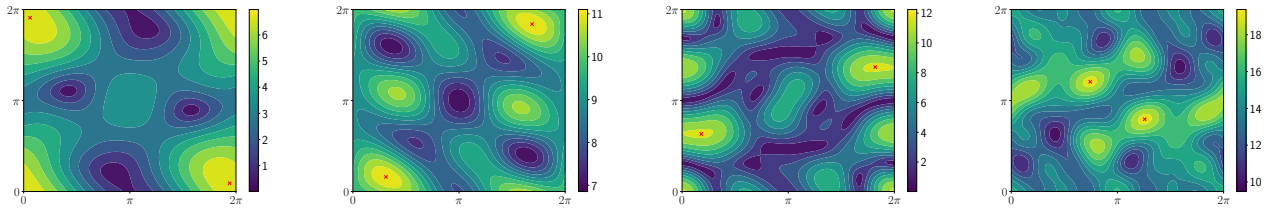
To have a bound on the full convolution operation, we extend Theorem 4 to take into account the number of output channels. The matrix of a full convolution operation is a block matrix where each block is a doubly-block Toeplitz matrices. Therefore, we will need the following lemma which bound the singular values of a matrix constructed from the concatenation of multiple matrix.

Lemma 1. *Let us define matrices $\mathbf{A}_1, \dots, \mathbf{A}_p$ with $\mathbf{A}_i \in \mathbb{R}^{n \times n}$. Let us construct the matrix $\mathbf{M} \in \mathbb{R}^{n \times pn}$ as follows:*

$$\mathbf{M} \triangleq (\mathbf{A}_1, \dots, \mathbf{A}_p) \quad (10)$$

where $(\,\cdot\,)$ define the concatenation operation. Then, we can bound the singular values of the matrix $\mathbf{M}$ as follows:

$$\sigma_1(\mathbf{M}) \leq \sqrt{\sum_{i=1}^{p} \sigma_1(\mathbf{A}_i)^2} \quad (11)$$



kernel $1 \times 3 \times 3$ kernel $9 \times 3 \times 3$ kernel $1 \times 5 \times 5$ kernel $9 \times 5 \times 5$

Figure 1: These figures represent the contour plot of multivariate trigonometric polynomials where the values of the coefficient are the values of a random convolutional kernel. The red dots in the figures represent the maximum modulus of the trigonometric polynomials.

Below, we present our main result:

Theorem 5 (**Main Result**: Bound on the maximal singular value on the convolution operation)**.** *Let us define doubly-block Toeplitz matrices* $\mathbf{D}(f_{11}), \dots, \mathbf{D}(f_{cin \times cout})$ *where* $f_{ij} : \mathbb{R}^2 \to \mathbb{C}$ *is a generating function. Construct a matrix* $\mathbf{M}$ *with* $cin \times n^2$ *rows and* $cout \times n^2$ *columns such as*

$$\mathbf{M} \triangleq \begin{pmatrix} \mathbf{D}(f_{11}) & \cdots & \mathbf{D}(f_{1,cout}) \\ \vdots & & \vdots \\ \mathbf{D}(f_{cin,1}) & \cdots & \mathbf{D}(f_{cin,cout}) \end{pmatrix}. \quad (12)$$

Then, with f_{ij} *a multivariate polynomial of the same form as Equation 7, we have:*

$$\sigma_1(\mathbf{M}) \leq \sqrt{\sum_{i=1}^{cout} \sup_{\omega_1,\omega_2 \in [0,2\pi]^2} \sum_{j=1}^{cin} |f_{ij}(\omega_1, \omega_2)|^2}. \quad (13)$$

We can easily express the bound in Theorem 5 with the values of a 4-dimensional kernel. Let us define a kernel $\mathbf{k} \in \mathbb{R}^{cout \times cin \times s \times s}$, a padding $p \in \mathbb{N}$ and $d = \lfloor s/2 \rfloor$ the degree of the trigonometric polynomial, then:

$$f_{ij}(\omega_1, \omega_2) = \sum_{h_1=-d}^{d} \sum_{h_2=-d}^{d} k_{i,j,h_1,h_2} e^{\mathbf{i}(h_1\omega_1 + h_2\omega_2)}. \quad (14)$$

where $k_{i,j,h_1,h_2} = (\mathbf{k})_{i,j,a,b}$ with $a = s - p - 1 + i$ and $b = s - p - 1 + j$.

In the rest of the paper, we will refer to the bound in Theorem 5 applied to a kernel as LipBound and we denote $\text{LipBound}(\mathbf{k})$ the Lipschitz upper-bound of the convolution performed by the kernel $\mathbf{k}$.

Computation and Performance Analysis of LipBound

This section aims at analyzing the bound on the singular values introduced in Theorem 5. First, we present an algorithm to efficiently compute the bound, we analyze its tightness by comparing it against the true maximal singular value. Finally, we compare the efficiency and the accuracy of our bound against the state-of-the-art.

Computing the Maximum Modulus of a Trigonometric Polynomial

In order to compute LipBound from Theorem 5, we have to compute the maximum modulus of several trigonometric polynomials. However, finding the maximum modulus of a trigonometric polynomial has been known to be NP-hard (Pfister and Bresler 2018), and in practice they exhibit low convexity (see Figure 1). We found that for 2-dimensional kernels, a simple grid search algorithm such as PolyGrid (see Algorithm 1), works better than more sophisticated approximation algorithms (*e.g* Green 1999; De La Chevrotiere 2009). This is because the complexity of the computation depends on the degree of the polynomial which is equal to $\lfloor s/2 \rfloor$ where s is the size of the kernel and is usually small in most practical settings (*e.g* $s = 3$). Furthermore, the grid search algorithm can be parallelized effectively on CPUs or GPUs and runs within less time than alternatives with lower asymptotic complexity.

To fix the number of samples S in the grid search, we rely on the work of Pfister and Bresler (2018), who has analyzed the quality of the approximation depending on S. Following, this work we first define Θ_S, the set of S equidistant sampling points as follows:

$$\Theta_S \triangleq \left\{ \omega \mid \omega = k \cdot \frac{2\pi}{S} \text{ with } k = 0, \dots, S-1 \right\}. \quad (15)$$

Then, for $f : [0, 2\pi]^2 \to \mathbb{C}$, we have:

$$\max_{\omega_1,\omega_2 \in [0,2\pi]^2} |f(\omega_1, \omega_2)| \leq (1-\alpha)^{-1} \max_{\omega'_1,\omega'_2 \in \Theta_S^2} |f(\omega'_1, \omega'_2)|, \quad (16)$$

where d is the degree of the polynomial and $\alpha = 2d/S$. For a 3×3 kernel which gives a trigonometric polynomial of degree 1, we use $S = 10$ which gives $\alpha = 0.2$. Using this result, we can now compute LipBound for a convolution operator with *cout* output channels as per Theorem 4.

Analysis of the Tightness of the Bound

In this section, we study the tightness of the bound with respect to the dimensions of the doubly-block Toeplitz matrices. For each $n \in \mathbb{N}$, we define the matrix $\mathbf{M}^{(n)}$ of size $kn^2 \times n^2$ as follows:

$$\mathbf{M}^{(n)} \triangleq \left(\mathbf{D}^{(n)\top}(f_1), \dots, \mathbf{D}^{(n)\top}(f_k) \right)^\top \quad (17)$$

where the matrices $\mathbf{D}^{(n)}(f_i)$ are of size $n^2 \times n^2$. To analyze the tightness of the bound, we define the function Γ, which

Algorithm 1 PolyGrid

input polynomial f, number of samples S
output approximated maximum modulus of f
$\sigma \leftarrow 0, \omega_1 \leftarrow 0, \epsilon \leftarrow 2\pi/S$
for $i = 0$ **to** $S - 1$ **do**
  $\omega_1 \leftarrow \omega_1 + \epsilon, \omega_2 \leftarrow 0$
  for $j = 0$ **to** $S - 1$ **do**
    $\omega_2 \leftarrow \omega_2 + \epsilon$
    $\sigma \leftarrow \max(\sigma, f(\omega_1, \omega_2))$
  end for
end for
return σ

computes the difference between LipBound and the maximal singular value of the function $\mathbf{M}^{(n)}$:

$$\Gamma(n) = \text{LipBound}(\mathbf{k}_{\mathbf{M}^{(n)}}) - \sigma_1(\mathbf{M}^{(n)}) \tag{18}$$

where $\mathbf{k}_{\mathbf{M}^{(n)}}$ is the convolution kernel of the convolution defined by the matrix $\mathbf{M}^{(n)}$.

To compute the exact largest singular value of $\mathbf{M}^{(n)}$ for a specific n, we use the Implicitly Restarted Arnoldi Method (IRAM) (Lehoucq and Sorensen 1996) available in SciPy. The results of this experiment are presented in Figure 2. We observe that the difference between the bound and the actual value (approximation gap) quickly decreases as the input size increases. For an input size of 50, the approximation gap is as low as 0.012 using a standard $6 \times 3 \times 3$ convolution kernel. For a larger input size such as ImageNet (224), the gap is lower than 4.10^{-4}. Therefore LipBound gives an almost exact value of the maximal singular value of the operator matrix for most realistic settings.

Comparison of LipBound with Other State-of-the-Art Approaches

In this section we compare our PolyGrid algorithm with the values obtained using alternative approaches. We consider the 3 alternative techniques by Sedghi, Gupta, and Long (2019), by Singla and Feizi (2019) and by Farnia, Zhang, and Tse (2019) which have been described in Section .

To compare the different approaches, we extracted 20 kernels from a trained model. For each kernel we construct the corresponding doubly-block Toeplitz matrix and compute its largest singular value. Then, we compute the ratio between the approximation obtained with the approach in consideration and the exact singular value obtained by SVD, and average the ratios over the 20 kernels. Thus good approximations result in approximation ratios that are close to 1. The results of this experiment are presented in Table 1. The comparison has been made on a Tesla V100 GPU. The time was computed with the PyTorch CUDA profiler and we warmed up the GPU before starting the timer.

The method introduced by Sedghi, Gupta, and Long (2019) computes an approximation of the singular values of convolutional layers. We can see in Table 1 that the value is off by an important margin. This technique is also computationally expensive as it requires computing the SVD of n^2 small matrices where n is the size of inputs. Singla and Feizi (2019) have shown that the singular value of the reshape kernel is a bound on the maximal singular value of the convolution layer. Their approach is very efficient but the approximation is loose and overestimate the real value. As said previously, the power method provides a good approximation at the expense of the efficiency. We use the special Convolutional Power Method from (Farnia, Zhang, and Tse 2019) with 10 iterations. The results show that our proposed technique: PolyGrid algorithm can get the best of both worlds. It achieves a near perfect accuracy while being very efficient to compute.

We provide in the supplementary material a benchmark on the efficiency of LipBound on multiple convolutional architectures.

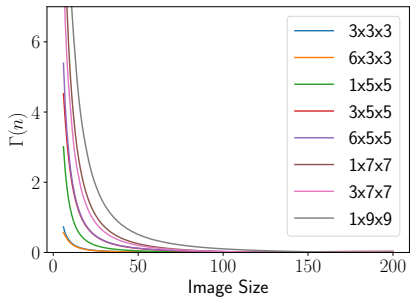


Figure 2: This graph represents the function $\Gamma(n)$ defined in Section for different kernel size.

Application: Lipschitz Regularization for Adversarial Robustness

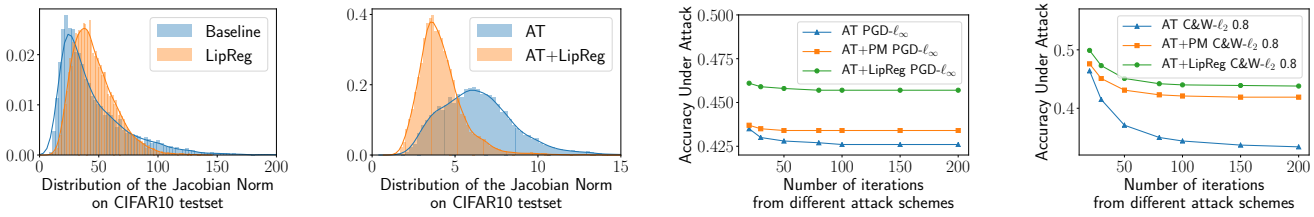
One promising application of Lipschitz regularization is in the area of adversarial robustness. Empirical techniques to improve robustness against adversarial examples such as Adversarial Training only impact the training data, and often show poor generalization capabilities (Schmidt et al. 2018). Farnia, Zhang, and Tse (2019) have shown that the adversarial generalization error depends on the Lipschitz constant of the network, which suggests that the adversarial test error can be improved by applying Lipschitz regularization in addition to adversarial training.

In this section, we illustrate the usefulness of LipBound by training a state-of-the-art Wide ResNet architecture (Zagoruyko and Komodakis 2016) with Lipschitz regularization and adversarial training. Our regularization scheme is inspired by the one used by Yoshida and Miyato (2017) but instead of using the power method, we use our **PloyGrid** algorithm presented in Section which efficiently computes an upper-bound on the maximal singular value of convolutional layers.

We introduce the **AT+LipReg** loss to combine Adversarial Training and our Lipschitz regularization scheme in which layers with a large Lipschitz constant are penalized. We consider a neural network $\mathcal{N}_\theta : \mathcal{X} \to \mathcal{Y}$ with ℓ layers $\phi^{(1)}_{\theta_1}, \ldots, \phi^{(\ell)}_{\theta_\ell}$ where $\theta^{(1)}, \ldots, \theta^{(\ell-1)}$ are the kernels of the first $\ell - 1$ convolutional layers and θ_ℓ is the weight matrix of the last fully-connected layer $\phi^{(\ell)}_{\theta_\ell}$. Given a distribution $\mathcal{D}$

	1x3x3		32x3x3	
	Ratio	**Time (ms)**	**Ratio**	**Time (ms)**
Sedghi, Gupta, and Long	0.431 ± 0.042	1088 ± 251	0.666 ± 0.123	1729 ± 399
Singla and Feizi	1.293 ± 0.126	1.90 ± 0.48	1.441 ± 0.188	1.90 ± 0.46
Farnia, Zhang, and Tse (10 iter)	0.973 ± 0.006	4.30 ± 0.64	0.972 ± 0.004	4.93 ± 0.67
LipBound (Ours)	$\mathbf{0.992} \pm 0.012$	$\mathbf{0.49} \pm 0.05$	$\mathbf{0.984} \pm 0.021$	$\mathbf{0.63} \pm 0.46$

Table 1: This table compares different approaches for computing an approximation of the maximal singular value of a convolutional layer. It shows the ratio between the approximation and the true maximal singular value. The approximation is better for a ratio close to one.



(a) (b) (c) (d)

Figure 3: Figures (a) and (b) show the distribution of the norm of the Jacobian matrix w.r.t the CIFAR10 test set from a Wide Resnet trained with different schemes. Although Lipschitz regularization is not a Jacobian regularization, we can observe a clear shift in the distribution. This suggests that our method does not only work layer-wise, but also at the level of the entire network. Figures (c) and (d) show the Accuracy under attack on CIFAR10 test set with PGD-ℓ_∞ and C&W-ℓ_2 attacks for several classifiers trained with Adversarial Training given the number of iterations.

over $\mathcal{X} \times \mathcal{Y}$, we can train the parameters θ of the network by minimizing the AT+LipReg loss as follows:

$$\min_{\theta} \mathbb{E}_{x,y\sim\mathcal{D}} \left[\max_{\|\tau\|_\infty \leq \epsilon} \mathcal{L}(\mathcal{N}_\theta(x+\tau), y) + \lambda_1 \sum_{i=1}^{\ell} \|\theta_i\|_{\mathrm{F}} + \lambda_2 \sum_{i=1}^{\ell-1} \log\left(\mathrm{LipBound}\left(\theta_i\right)\right) \right] \quad (19)$$

where $\mathcal{L}$ is the cross-entropy loss function, and λ_1, λ_2 are two user-defined hyper-parameters. Note that regularizing the sum of logs is equivalent to regularizing the product of all the LipBound which is an upper-bound on the global Lipschitz constant. In practice, we also include the upper-bound on the Lipschitz of the batch normalization because we can compute it very efficiently (see C.4.1 of Tsuzuku, Sato, and Sugiyama 2018) but we omit the last fully connected layer.

In this section, we compare the robustness of Adversarial Training (Goodfellow, Shlens, and Szegedy 2015; Madry et al. 2018) against the combination of Adversarial Training and Lipschitz regularization. To regularize the Lipschitz constant of the network, we use the objective function defined in Equation 19. We train Lipschitz regularized neural networks with LipBound (Theorem 5) implemented with PolyGrid (Algorithm 1) (AT+LipBound) with $S = 10$ or with the specific power method for convolutions introduced by Farnia, Zhang, and Tse (2019) with 10 iterations (AT+PM).

Table 2 shows the gain in robustness against strong adversarial attacks across different datasets. We can observe that both AT+LipBound and AT+PM offer a better defense against adversarial attacks and that AT+LipBound offers a further improvement over the Power Method. The Figure 3 (c) and (d) shows the Accuracy under attack with different number of iterations. Table 3 presents our results on the ImageNet Dataset. First, we can observe that the networks AT+LipReg offers a better generalization than with standalone Adversarial Training. Secondly, we can observe the gain in robustness against strong adversarial attacks. Network trained with Lipschitz regularization and Adversarial Training offer a consistent increase in robustness across ℓ_∞ and ℓ_2 attacks with different ϵ value. We can also note that increasing the regularization lead to an increase in generalization and robustness.

Finally, we also conducted an experiment to study the impact of the regularization on the gradients of the whole network by measuring the norm of the Jacobian matrix, averaged over the inputs from the test set. The results of this experiment are presented in Figure 3(a) and show more concentrated gradients with Lipschitz regularization, which is the expected effect. This suggests that our method does not only work layer-wise, but also at the level of the entire network. A second experiment, using Adversarial Training, presented in Figure 3(b) demonstrates that the effect is even stronger when the two techniques are combined together. This corroborates the work by Farnia, Zhang, and Tse (2019). It also demonstrates that Lipschitz regularization and Adversarial Training (or other Jacobian regularization techniques) are complementary. Hence they offer an increased robustness to adversarial attacks as demonstrated above.

Dataset	Model	Accuracy	PGD-ℓ_∞	C&W-ℓ_2 0.6	C&W-ℓ_2 0.8
CIFAR10	**Baseline**	$\mathbf{0.953} \pm 0.001$	0.000 ± 0.000	0.002 ± 0.000	0.000 ± 0.000
	AT	0.864 ± 0.001	0.426 ± 0.000	0.477 ± 0.000	0.334 ± 0.000
	AT+PM	0.788 ± 0.010	0.434 ± 0.007	0.521 ± 0.005	0.419 ± 0.003
	AT+LipReg	0.808 ± 0.022	$\mathbf{0.457} \pm 0.002$	$\mathbf{0.547} \pm 0.022$	$\mathbf{0.438} \pm 0.020$
CIFAR100	**Baseline**	$\mathbf{0.792} \pm 0.000$	0.000 ± 0.000	0.001 ± 0.000	0.000 ± 0.000
	AT	0.591 ± 0.000	0.199 ± 0.000	0.263 ± 0.000	0.183 ± 0.000
	AT+LipReg	0.552 ± 0.019	$\mathbf{0.215} \pm 0.004$	$\mathbf{0.294} \pm 0.010$	$\mathbf{0.226} \pm 0.008$

Table 2: This table shows the Accuracy under ℓ_2 and ℓ_∞ attacks of CIFAR10/100 datasets. We compare vanilla Adversarial Training with the combination of Lipschitz regularization and Adversarial Training. We also compare the effectiveness of the power method by Farnia, Zhang, and Tse (2019) and $\mathrm{LipBound}$. The parameters λ_2 (Eq. 19) is equal to 0.008 for AT+PM and AT+LipReg. It has been chosen from a grid search among 10 values. The attacks below are computed with 200 iterations.

Dataset	Model	LipReg λ_2	Natural	PGD-ℓ_∞		C&W-ℓ_2		
				0.02	0.031	1.00	2.00	3.00
ImageNet	**Baseline** (He et al. 2016)	–	**0.782**	0.000	0.000	0.000	0.000	0.000
	AT	–	0.509	0.251	0.118	0.307	0.168	0.099
	AT+LipReg	0.0006	**0.515**	**0.255**	**0.121**	**0.316**	**0.177**	**0.105**
	AT+LipReg	0.0010	**0.519**	**0.259**	**0.123**	**0.338**	**0.204**	**0.129**

Table 3: This table shows the accuracy and accuracy under ℓ_2 and ℓ_∞ attack of ImageNet dataset. We compare Adversarial Training with the combination of Lipschitz regularization and Adversarial Training (Madry et al. 2018).

Experimental Settings CIFAR10/100 Dataset For all our experiments, we use the Wide ResNet architecture introduced by Zagoruyko and Komodakis (2016) to train our classifiers. We use Wide Resnet networks with 28 layers and a width factor of 10. We train our networks for 200 epochs with a batch size of 200. We use Stochastic Gradient Descent with a momentum of 0.9, an initial learning rate of 0.1 with exponential decay of 0.1 (MultiStepLR gamma = 0.1) after the epochs 60, 120 and 160. For Adversarial Training (Madry et al. 2018), we use Projected Gradient Descent with an $\epsilon = 8/255 (\approx 0.031)$, a step size of $\epsilon/5 (\approx 0.0062)$ and 10 iterations, we use a random initialization but run the attack only once. To evaluate the robustness of our classifiers, we rigorously followed the experimental protocol proposed by Tramer et al. (2020) and Carlini et al. (2019). More precisely, as an ℓ_∞ attack, we use PGD with the same parameters ($\epsilon = 8/255$, a step size of $\epsilon/5$) but we increase the number of iterations up to 200 with 10 restarts. For each image, we select the perturbation that maximizes the loss among all the iterations and the 10 restarts. As ℓ_2 attacks, we use a bounded version of the Carlini and Wagner (2017) attack. We choose 0.6 and 0.8 as bounds for the ℓ_2 perturbation. Note that the ℓ_2 ball with a radius of 0.8 has approximately the same volume as the ℓ_∞ ball with a radius of 0.031 for the dimensionality of CIFAR10/100.

Experimental Settings for ImageNet Dataset For all our experiments, we use the Resnet-101 architecture (He et al. 2016). We have used Stochastic Gradient Descent with a momentum of 0.9, a weight decay of 0.0001, label smoothing of 0.1, an initial learning rate of 0.1 with exponential decay of 0.1 (MultiStepLR gamma = 0.1) after the epochs 30 and 60. We have used Exponential Moving Average over the weights with a decay of 0.999. We have trained our networks for 80 epochs with a batch size of 4096. For Adversarial Training, we have used PGD with 5 iterations, $\epsilon = 8/255 (\approx 0.031)$ and a step size of $\epsilon/5 (\approx 0.0062)$. To evaluate the robustness of our classifiers on ImageNet Dataset, we have used an ℓ_∞ and an ℓ_2 attacks. More precisely, as an ℓ_∞ attack, we use PGD with an epsilon of 0.02 and 0.031, a step size of $\epsilon/5$) with a number of iterations to 30 with 5 restarts. For each image, we select the perturbation that maximizes the loss among all the iterations and the 10 restarts. As ℓ_2 attacks, we use a bounded version of the Carlini and Wagner (2017) attack. We have used 1, 2 and 3 as bounds for the ℓ_2 perturbation.

Conclusion

In this paper, we introduced a new bound on the Lipschitz constant of convolutional layers that is both accurate and efficient to compute. We used this bound to regularize the Lipschitz constant of neural networks and demonstrated its computational efficiency in training large neural networks with a regularized Lipschitz constant. As an illustrative example, we combined our bound with adversarial training, and showed that this increases the robustness of the trained networks to adversarial attacks. The scope of our results goes beyond this application and can be used in a wide variety of settings, for example, to stabilize the training of Generative Adversarial Networks (GANs) and invertible networks, or to improve generalization capabilities of classifiers. Our future work will focus on investigating these fields.

Acknowledgements

We would like to thank Rafael Pinot and Geovani Rizk for their valuable insights. This work was granted access to the HPC resources of IDRIS under the allocation 2020-101141 made by GENCI.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* .

Bartlett, P. L.; Foster, D. J.; and Telgarsky, M. J. 2017. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Carlini, N.; Athalye, A.; Papernot, N.; Brendel, W.; Rauber, J.; Tsipras, D.; Goodfellow, I.; and Madry, A. 2019. On Evaluating Adversarial Robustness. *arXiv preprint arXiv:1902.06705* .

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 39–57. IEEE.

Cisse, M.; Bojanowski, P.; Grave, E.; Dauphin, Y.; and Usunier, N. 2017. Parseval Networks: Improving Robustness to Adversarial Examples. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

De La Chevrotiere, G. 2009. Finding the maximum modulus of a polynomial on the polydisk using a generalization of steckins lemma. *SIAM Undergraduate Research Online* .

Farnia, F.; Zhang, J.; and Tse, D. 2019. Generalizable Adversarial Training via Spectral Normalization. In *International Conference on Learning Representations (ICLR)*.

Fazlyab, M.; Robey, A.; Hassani, H.; Morari, M.; and Pappas, G. 2019. Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Golub, G. H.; and Van der Vorst, H. A. 2000. Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics* 123(1-2): 35–65.

Goodfellow, I.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*.

Gouk, H.; Frank, E.; Pfahringer, B.; and Cree, M. 2018. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv preprint arXiv:1804.04368* .

Gray, R. M.; et al. 2006. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory* 2(3): 155–239.

Green, J. 1999. Calculating the maximum modulus of a polynomial using Steckin's lemma. *SIAM journal on numerical analysis* 36(4): 1022–1029.

Gutiérrez-Gutiérrez, J.; Crespo, P. M.; et al. 2012. Block Toeplitz matrices: Asymptotic results and applications. *Foundations and Trends® in Communications and Information Theory* 8(3): 179–257.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jain, A. K. 1989. *Fundamentals of digital image processing*. Englewood Cliffs, NJ: Prentice Hall,.

Latorre, F.; Rolland, P.; and Cevher, V. 2020. Lipschitz constant estimation for Neural Networks via sparse polynomial optimization. In *International Conference on Learning Representations (ICLR)*.

Lehoucq, R. B.; and Sorensen, D. C. 1996. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications* 17(4): 789–821.

Li, Q.; Haque, S.; Anil, C.; Lucas, J.; Grosse, R. B.; and Jacobsen, J.-H. 2019. Preventing Gradient Attenuation in Lipschitz Constrained Convolutional Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957* .

Pfister, L.; and Bresler, Y. 2018. Bounding multivariate trigonometric polynomials with applications to filter bank design. *arXiv preprint arXiv:1802.09588* .

Schmidt, L.; Santurkar, S.; Tsipras, D.; Talwar, K.; and Madry, A. 2018. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Sedghi, H.; Gupta, V.; and Long, P. M. 2019. The Singular Values of Convolutional Layers. In *International Conference on Learning Representations (ICLR)*.

Singla, S.; and Feizi, S. 2019. Bounding Singular Values of Convolution Layers. *arXiv preprint arXiv:1911.10258* .

Tramer, F.; Carlini, N.; Brendel, W.; and Madry, A. 2020. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347* .

Tsuzuku, Y.; Sato, I.; and Sugiyama, M. 2018. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Virmaux, A.; and Scaman, K. 2018. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Widom, H. 1976. Asymptotic behavior of block Toeplitz matrices and determinants. II. *Advances in Mathematics* 21(1): 1–29.

Yi, X. 2020. Asymptotic Singular Value Distribution of Linear Convolutional Layers. *arXiv preprint arXiv:2006.07117* .

Yoshida, Y.; and Miyato, T. 2017. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941* .

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* .