Master's Thesis

Lipschitz Constants of Functions of Neural Networks

Sunjoong Kim

Department of Mathematics

Graduate School

Korea University

February 2024

Lipschitz Constants of Neural Networks

by
Sunjoong Kim

under the supervision of Professor Seungsang Oh

A thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Mathematics

Department of Mathematics

Graduate School

Korea University

October 2023

The thesis of Sunjoong Kim has been approved by the thesis committee in partial fulfillment of the requirements for the degree of Master of Mathematics

December 2023

Committee Chair: Name

Committee Member: Name

Committee Member: Name

Lipschitz Constants of Neural Networks

by Sunjoong Kim

Department of Mathematics

under the supervision of Professor Seungsang Oh

Abstract

In machine learning and deep learning, making an algorithm stable or robust is just as important as making it perform well. One of the indicator that measures the sensitiveness or the robustness of an algorithm is the Lipschitz constant. This paper discusses several ways to calculate the Lipschitz constant of basic machine learning algorithms. One of the main part of the perceptron and MLP is the linear function, whose Lipschitz constant coincide with the operator norm of the corresponding matrix. Unlike the linear function, evaluating the optimal Lipschitz constant of many algorithms is difficult or infeasible even for the two layered MLP. Instead of calculating the exact constant, there is a systematic way called *AutoLip* to find an upper bound for the constant and we apply it to MLP and CNN. For example, the convolution layer can be thought of as a matrix multiplication by a long matrix and it may require several tools to make it simple.

Keywords: Lipschitz constant, Operator norm, Rademacher's theorem, AutoLip, Power method, Rayleigh quotient

인공신경망의 립시츠 상수

김 선 중

수 학 과

지 도 교 수: 오 승 상

국문 초록

머신러닝과 딥러닝에서 안정적인 알고리즘을 만드는 것은 그 알고리즘이 좋은 성능을 내는 것 만큼이나 중요하다. 알고리즘의 안정성과 관련되어 있는 수학적인 개념 중 하나는 립시츠 상수이다. 이 논문에서는 기본적인 머신러닝 알고리즘의 립시츠 상수를 계산하는 다양한 방법에 대해 논의한다. 가장 기본적인 구조인 퍼셉트론은 선형함수의 구조를 가지고 있고, 선형함수의 립시츠 상수는 그 선형함수에 대응되는 행렬에 대한 작용소 놈의 값과 일치한다. 하지만, 선형함수에서 조금만 알고리즘이 복잡해져도 립시츠 상수를 계산하는 것은 어려워진다. 예를들어, 레이어가 두 개인 다층퍼셉트론의 경우만 해도 정확하게 립시츠 상수를 계산하는 것이 거의 불가능하다(NP-hard). 그래서 이 논문에서는, 다양한 머신러닝 알고리즘의 정확한 립시츠 상수 값을 계산하는 대신 립시츠 상수의 상한(upper bound)을 계산하는 체계적인 알고리즘(AutoLip)을 소개하고, 이것을 다층퍼셉트론과 합성곱신경망에 적용해본다. 합성곱 레이어는, 특정한 형태의 행렬의 행렬곱으로 이해할 수 있고, 따라서 작용소 놈을 계산하면 립시츠 상수가 얻어진다. 이때, 효율적인 계산을 위해 누승법이나 레일리몫과 같은 방법이 사용될 수 있다.

중심어 : 립시츠 상수, 작용소 놈, Rademacher 정리, AutoLip, 누승법, 레일리 몫

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

Deep neural networks have made many advances including computer vision, language modeling, machine translation and text and picture generating. Still, one of the difficulties in applying deep learning algorithms to reality is that the algorithm often lacks its robustness. As an example, Szegedy et al. have found that a small perturbation of an input for true image may cause the classifier to misclasssify the image as false image [2].

To overcome the instabiltiy of training and to enhance robustness of generating models such as GANs, M. Arjovsky et al. proposed Wasserstein distance between distributions and restrict their attention to 1-Lipschitz function to the critic [3], [4]. As this example suggest, the Lipschitz constant can be a good metric to access the robustness of the algorithm.

The Lipschitz constant of a function measures the sensitiveness or the robustness, or the rate of changes of the function. In chapter 2, we define the optimal Lipschitz constant of the functions between euclidean spaces and explore the computation of this optimal constant for various functions including linear maps, affine maps and compositions of functions. However, if the function becomes more complex, it is difficult or almost impossible to calculate the optimal constant. So, we propose algorithms to estimate the constant, using the Rademacher's theorem.

Chapter 2. Lipschitz Constants of Functions Between Euclidean Spaces

2.1 Lipschitz Constants

For vectors $x = [x_1 \ \cdots \ x_n]^T$ and $y = [y_1 \ \cdots \ y_m]^T$ in Euclidean spaces $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, $||x||$ and $||y||$ are the usual standard norms of x and y defined by

$$||x|| = \sqrt{{x_1}^2 + \cdots + {x_n}^2}, \quad ||y|| = \sqrt{{y_1}^2 + \cdots + {y_m}^2}. \tag{2.1.1}$$

Definition 1. *A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is called* Lipschitz continuous *if there is a nonnegative real number c such that*

$$||f(x) - f(y)|| \leq c||x - y|| \qquad (x, y \in \mathbb{R}^n) \tag{2.1.2}$$

The infimum of c which satisfies (2.1.2) *is called the* Lipschitz constant *of f and is denoted by $Lip(f)$.*

This definition can also be extended to functions from a metric space to another metric space if we substitute the norms with the distances. It is useful to know that there

Chapter 3. Lipschitz Constants of Neural Networks

Now we turn to the problem of finding the optimal Lischitz constant of various architectures frequently appearing in machine learning and deep learning.

For locally Lipschitz function, it is possible to express the Lipschitz constant of f in its exact form, as the Rademacher Theorem suggests. But, it is not always easy or feasible to find the exact value of Lipschitz constant whenever the function f is given. In fact, although the function is of simple form such as 2-layered MLP, solving the precise value of Lipschitz constant of the function is known to be NP-hard. Instead of struggling to find analytic solutions, we present a systematic algorithm for each architectuare of neural network[1].

3.1 Rademacher Theorem

Here is a theorem that can be applied to all Lipschitz functions between euclidean spaces. The Lipschitz condition that we impose is not the global one ; it only need to be locally Lipschitz. It has two conclusions : the differentiability in almost everywhere sense and the formula for the optimal constant.

Chapter 4. Conclusion

For a function between metric spaces, the Lipschitz constant measures the rate of change of the function. If we conceive a neural network as a function between Euclidean spaces, the Lipschitz constant of the neural network tells us how sensitive the output is, relative to the input. One can relate the Lipschitz constant to the robustness of the algorithm. But the Lipschitz constant is not the same as the robustness of an *architecture*, in that it measures the sensitivity of an *algorithm* when the parameters are fixed. We may further study the robustness by investigating the changes of the Lipschitz constant when updating the parameters of the algorithm.

The Lipscthiz constant for relatively simple functions such as activation functions, linear functions and affine functions can be obtained either by simple calculation or the notion of matrix norm. We can express the Lipschitz constant of *any* locally Lipschitz functions by Rademacher Theorem, but evaluating the optimal constant in its exact value is difficult or not feasible. Rather, we can think of several numerical way to find an upper bound for the optimal constant either by an algorithm called AutoLip or by the Rayleigh quotient and power method. This paper doesn't include the implementation with programming code, such as Python. One may relate the calculations to the experimental results.

Reference

[1] Virmaux, A., & Scaman, K. (2018). Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31.

[2] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. *Proceedings of the International Conference on Learning Representations (ICLR)*

[3] Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In International conference on machine learning (pp. 214-223). PMLR.

[4] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. Advances in neural information processing systems, 30.

[5] Computing The Norm of a matrix, `https://kconrad.math.uconn.edu/blurbs/linmultialg/matrixnorm.pdf`, University of Connecticut.

[6] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[7] Rudin, Walter. *Real and Complex Analysis.* : McGraw-Hill Science/Engineering/Math, 1986.

[8] Herbert Federer. *Geometric measure theory.* Classics in Mathematics. Springer-Verlag Berlin Heidelberg, 1969.

[9] Villani, C. (2009). *Optimal transport: old and new* (Vol. 338, p. 23). Berlin: springer.

[10] Piotr Hajłasz, *Geometric Analysis*, `https://sites.pitt.edu/~hajlasz/Teaching/Math2304Spring2014/Part\%201.pdf`, University of Pittsburgh.

[11] James T. Murphy, *An elementary proof of Rademacher's theorem*, `https://intfxdx.com/downloads/rademacher-thm-2015.pdf` THEOREM

[12] Jordan, M., & Dimakis, A. G. (2020). Exactly computing the local lipschitz constant of relu networks. *Advances in Neural Information Processing Systems*, 33, 7344-7353.

[13] Bhowmick, A., D'Souza, M., & Raghavan, G. S. (2021, September). Lipbab: computing exact lipschitz constant of relu networks. *In International Conference on Artificial Neural Networks* (pp. 151-162). Springer, Cham.

[14] Beer, G., & Hoffman, M. J. (2013). The Lipschitz metric for real-valued continuous functions. *Journal of Mathematical Analysis and Applications*, 406(1), 229-236.

[15] Carlini, N., & Wagner, D. (2017, May). Towards evaluating the robustness of neural networks. In 2017 *ieee symposium on security and privacy (sp)* (pp. 39-57). Ieee.

[16] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.