Smoothness is All You Need: Examining the Role of Smooth Activations in Robustness Training

Candidate Number:

Abstract

Robustness training is notorious for its computational cost. To address this, Input Gradient Regularization (IGR) has been proposed as a more efficient alternative. Before the recent breakthrough of Rodríguez-Muñoz et al. [14], IGR has historically struggled to achieve robustness results comparable to Adversarial Training, either leading to gradient masking or inferior performance when generalizing to modern datasets. In this project, we discuss the mathematical background and motivation behind IGR in robustness training and confirm whether smoothness is a sufficient condition for IGR's success.

1 Introduction

Deep neural networks are very brittle to adversarial attacks; imperceptible perturbations to inputs can cause catastrophic missclassifications [5]. Extensive research has been devoted to build robustness against adversarial attacks, with Adversarial Training via Projected Gradient Descent (PGD) [10] emerging as the most effective defense to date. While effective, Adversarial Training requires significant computational resources, as it involves iteratively generating strong adversarial examples during training, increasing training time by an order of magnitude compared to standard training.

Consequently, Input Gradient Regularization (IGR) has been proposed as a more efficient alternative. First introduced in [3] to improve generalization, this method penalizes sensitivity to input perturbations, which aligns with the necessary requirement for model robustness: a function stable to small perturbations should have a small norm of its derivative. Therefore, IGR serves as a natural defense against adversarial examples by furnishing local smoothness, and its intuition is theoretically grounded in the robust optimization framework established by Madry et al. [10], which formulates robustness training as a minimax optimization problem $\min_\theta \mathbb{E}[\max_\delta \mathcal{L}(x+\delta, y; \theta)]$. Simon-Gabriel et al. showed in [17] that IGR can be viewed as a first-order Taylor approximation of this inner maximization problem, and thus serving as a proxy for Adversarial Training without the cost of iterative backpropagation.

Despite strong theoretical arguments, IGR has historically struggled to achieve competitive robustness comparable results to Adversarial Training, either leading inferior performance when generalizing to modern datasets [15, 16] or gradient masking [12], whereby the model breaks the gradient-based attacks but does not provide robustness against other types of attacks. Recently, however, Rodríguez-Muñoz et al. [14] showed that IGR can yield comparable robustness results to Adversarial Training when using smooth activation functions such as GeLU instead of the standard ReLU. While their paper successfully generalized IGR to modern datasets and achieved competitive robustness results, their experiments only examined two smooth activation functions of extremely similar characteristics (GeLU and SiLU) and did not throughly isolate the effect of smoothness on IGR's performance.

In this project, we discuss the mathematical background and motivation behind IGR in robustness training and confirm whether smoothness is indeed the key to IGR's success.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

2 Mathematical Background

In this section, we discuss the theoretical foundations of robustness training and propose a unified view of two historically distinct methods: FGSM-based training and Input Gradient Regularization (IGR). We begin by defining the robust optimization framework and the PGD attack introduced by Madry et al. [10], as well as its relation to FGSM. We then show how IGR emerges as a first-order Taylor approximation of this framework using the derivations from Simon-Gabriel et al. [17]. Finally, we address the historical ineffectiveness of IGR and FGSM-based training by proposing a novel unified theoretical view of these two methods. In particular, we use the local robustness bound from Finlay and Oberman [4] to decompose the FGSM objective, revealing exactly how it relates to IGR and why non-smooth activations allow models to invalidate the training process.

2.1 The Robust Optimization Framework and Projective Gradient Descent

Standard deep learning training attempts to find the model parameters θ that minimize the risk $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(x, y; \theta)]$ over a data distribution $\mathcal{D}$. However, this approach yields models vulnerable to adversarial examples, where an input x is susceptible to some small perturbation δ such that $x + \delta$ is misclassified. To address this, Madry et al. [10] proposed a general adversarial training objective, which is formulated as a saddle point problem:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{\delta\in\mathcal{S}} \mathcal{L}(x + \delta, y; \theta)]. \tag{1}$$

Here, $\mathcal{S}$ represents the set of allowed perturbations, typically an ℓ_p-norm ball $\mathcal{S} = \{\delta \in \mathbb{R}^d \mid \|\delta\|_p \leq \epsilon\}$. The rest of this project will focus on the standard ℓ_∞-bounded perturbations.

The saddle point problem formulation in (1) can be viewed as a battle between an adversary and a defender, where the defender seeks to find the best model parameters θ that minimizes the the maximum adversarial loss δ. The inner maximization problem is often non-concave and difficult to solve exactly. The famous Fast Gradient Sign Method (FGSM) proposed by Goodfellow et al. [5] can be interpreted as a simple one-step linearization of the inner maximization: By the first-order Taylor expansion,

$$\mathcal{L}(x + \delta, y; \theta) \approx \mathcal{L}(x, y; \theta) + \delta^T \nabla_x \mathcal{L}(x, y; \theta), \tag{2}$$

and so the FGSM attack is given by

$$x^{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x, y; \theta)) = x + \underset{\|\delta\|_\infty \leq \epsilon}{\arg\max}\, \delta^T \nabla_x \mathcal{L}(x, y; \theta). \tag{3}$$

While computationally cheap, FGSM relies on the assumption that the loss surface is locally linear around x, which can be drasticaly far from the actual loss landscape. This often leads to catastrophic overfitting [7], where the model becomes robust to the FGSM attack but performs disastrously (0% robustness) against other types of attacks like PGD.

Madry et al. demonstrated that projected gradient descent (PGD) is a more powerful multi-step variant of FGSM, which iteratively applies FGSM then projects the result back into the allowed perturbation set $\mathcal{S}$:

$$x^{(t+1)} = \Pi_{x+\mathcal{S}}(x^{(t)} + \alpha\, \text{sign}(\nabla_x \mathcal{L}(x, y; \theta))).$$

PGD is then applied to adversarial training framework, which replaces the natural input x with the PGD generated adversarial example x^{adv} during the outer minimization step in (1). While effective, this requires multiple gradient calculations per training step, increasing training time by an order of magnitude compared to standard training.

2.2 Input Gradient Regularization as an Alternative

In contrast to the high training cost of the multi-step Adversarial Training, IGR was proposed as a more efficient alternative [9, 15, 6]. The heuristic motivation is that inducing local smoothness by forcing loss gradients to be small should make the model less sensitive to input perturbations. Consider the inner maximization problem for a perturbation δ bounded by $\|\delta\|_\infty \leq \epsilon$. Assuming the loss function $\mathcal{L}$ is differentiable with respect to the input x, it is shown in [17] that for small ϵ,

$$\max_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}(x + \delta, y; \theta) \approx \mathcal{L}(x, y; \theta) + \epsilon \|\nabla_x \mathcal{L}(x, y; \theta)\|_1.$$

This leads directly to the IGR objective, which effectively replaces the inner maximization problem in (1) with a first-order approximation:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(x, y; \theta) + \lambda \|\nabla_x \mathcal{L}(x, y; \theta)\|_1], \tag{4}$$

where λ is a hyperparameter governing the strength of the regularization.

However, formulating IGR merely as an approximation of the minimax problem ignores the complexity of the loss landscape. If the approximation error is large, minimizing the gradient norm may not result in true robustness. This is the reason why λ is introduced in the IGR objective instead of directly using the attack strength ϵ. To mathematically justify this, we examine the theoretical bound of robustness.

2.3 The Theoretical Bound of Robustness

The validity of IGR is theoretically grounded in [4], where Finlay and Oberman establish a lower bound on the size of an adversarial perturbation δ required to alter the class prediction based on local gradient information. At input x and perturbation δ, denote the first-order approximation error of the loss function as

$$R(x, \delta) = \mathcal{L}(x + \delta) - [\mathcal{L}(x) + \langle \nabla_x \mathcal{L}(x), \delta \rangle]. \tag{5}$$

Then the first-order approximation error is bounded above by

$$\omega(\epsilon) = \sup_{x, \|\delta\| \leq \epsilon} R(x, \delta).$$

We call $\omega(\epsilon)$ the modulus of continuity of the loss function and note that $\omega(\epsilon) \geq R(x, 0) = 0$.

Proposition 2.1 (Finlay and Oberman [4])**.** *Let $\mathcal{L}(x)$ be a loss function and $\mathcal{L}_0$ be such that the model is correct whenever $\mathcal{L}(x) \leq \mathcal{L}_0$. Then the minimum magnitude of perturbation δ necessary to adversarially perturb an input x is bounded below by ϵ if*

$$\frac{\mathcal{L}_0 - \mathcal{L}(x) - \omega(\epsilon)}{\|\nabla_x \mathcal{L}(x)\|_1} \geq \epsilon. \tag{6}$$

This bound yields three sufficient conditions for robustness: (i) large loss gap $\mathcal{L}_0 - \mathcal{L}(x)$, (ii) small first-order approximation error $\omega(\epsilon)$, and (iii) small input gradient norm $\|\nabla_x \mathcal{L}(x)\|_1$. This give a justification for IGR's effectiveness, as its objective function (4) directly targets condition (iii). However, this bound also highlights that minimizing the gradient norm is necessary but not sufficient. If the linearization error $\omega(\epsilon)$ is large, the lower bound on $\|\delta\|$ can remain small despite the gradient norm $\|\nabla_x \mathcal{L}(x)\|_1$ being small. This also justifies the use of λ in (4), as setting $\lambda = \epsilon$ is assuming the loss function is locally linear and ignoring the $\omega(\epsilon)$ term. Thus we scale λ up to absorb the curvature of the loss function [4].

2.4 Smoothness is What You Need

Historically, IGR has struggled to achieve competitive robustness compared to PGD adversarial training [16]. This failure is closely linked to the first-order approximation error term $\omega(\epsilon)$ in (6) and how it is handled in non-smooth networks like ReLU.

It is not explicitly discussed in [4], but for non-smooth networks the first-order approximation error $\omega(\epsilon)$ can be large even for small ϵ, contradicting the requirement for robustness given by the robustness bound (6). This mathematically explains the phenomenon of gradient masking [12], where the model keeps the gradient norm small but does not provide robustness against other types of attacks. Finlay and Oberman did not directly resolve this issue in [4] but offered a workaround using finite differences to estimate the gradient norm and avoided the calculation of second-order derivatives.

In recent work by Rodríguez-Muñoz et al. [14] and Xie et al. [19], it was shown that using smooth activation functions like GeLU and SiLU can significantly improve IGR's robustness. In particular, Rodríguez-Muñoz et al. showed replacing the non-smooth ReLU with the smooth GeLU and SiLU activation function achieves $> 90\%$ of robustness when compared to PGD adversarial training while using merely 63% of the computing cost. While Xie et al. attribute this to better gradient quality, we argue that this success is due to smooth activations bounding the first-order approximation error

$R(x, \delta)$, which in turn bounds the modulus of continuity $\omega(\epsilon)$. This ensures condition (ii) in (6) is satisfied. While Rodríguez-Muñoz et al. only examined two smooth activation functions of extremely similar characteristics (GeLU and SiLU) and did not throughly isolate the effect of smoothness on IGR's performance, we will experiment other activation functions of different characteristics (e.g. Softplus, PReLU) to empirically verify if smoothness is indeed the key to IGR's success.

2.5 IGR vs FGSM

As mentioned in the previous sections, IGR and FGSM are both essentially first-order approximations of the inner maximization problem in (1). So what exactly is the difference between IGR and FGSM-based training? Why is FGSM-based training subjected to catastrophic overfitting but not IGR? In this section, we propose (to the best of our knowledge) a novel unified theoretical view of these two methods using the local robustness bound (6).

The phenomenon of catastrophic overfitting in FGSM-based training has been extensively studied, with numerous works identifying the breakdown of local linearity as the primary cause [18, 1, 8]. To mitigate this, the dominant approach has been to introduce regularization terms that penalize curvature or gradient misalignment, such as GradAlign [1], Local Linearization Regularization [13], and CURE [11]. While these methods are effective, they treat the stability of FGSM and the objective of IGR as largely distinct optimization goals.

Simon-Gabriel et al. [17] noted a link between the two methods by showing that FGSM-based training is a data augmentation technique that accounts for additional points $x + \delta$ perturbed by ϵ-sized FGSM attacks $\delta = \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x))$, which alters the training loss function to

$$\tilde{\mathcal{L}}_\epsilon(x, y; \theta) = \mathcal{L}(x, y; \theta) + \mathcal{L}(x + \delta, y; \theta).$$

On the other hand, IGR is simply adding a regularization term to the loss function to penalize the input gradient norm $\|\nabla_x \mathcal{L}(x)\|_1$. However, this comparison is insufficient to explain why FGSM-based training leads to catastrophic overfitting while IGR remains stable. Here is where we fill in the theoretical gap: if we introduce the approximation error term $R(x, \delta)$ defined in (5), we get

$$\tilde{\mathcal{L}}_\epsilon(x, y; \theta) = \mathcal{L}(x, y; \theta) + \epsilon \|\nabla_x \mathcal{L}(x)\|_1 + R(x, \delta). \tag{7}$$

Notice that this is essentially the same as the IGR loss function (4) when $\lambda = \epsilon$, with the critical difference being the addition of the approximation error term $R(x, \delta)$. This allows the model to "cheat" in FGSM-based training. Instead of minimizing the gradient norm $\|\nabla_x \mathcal{L}(x)\|_1$, the model resorts to minimizing $R(x, \delta)$ to be large and negative. This contradicts the necessary condition (iii) for robustness given by the Finlay-Oberman bound.

By making $R(x, \delta)$ negative and $|R(x, \delta)|$ large, the model essentially invalidates the first-order approximation which FGSM relies on. This gives mathematical explanation for why FGSM-based training leads to catastrophic overfitting. We suspect that this phenomenon is especially pronounced for non-smooth networks like ReLU, where the $R(x, \delta)$ term can be unbounded for the model to exploit. However, if we use smooth activation functions like GeLU, the term $R(x, \delta)$ is bounded by the Lipschitz constant of the gradient, then the loss function (7) resembles the loss function of IGR (4). Additionally, $R(x, \delta)$ being locally bounded suggests that the modulus of continuity $\omega(\epsilon)$ is also constrained, which aligns with a necessary condition for robustness given by (6). Therefore, we suspect that FGSM-based training should be able to achieve comparable, if not better, robustness to IGR. This theoretical view is consistent with the experimental results in [19], and we will also empirically verify FGSM-based training's failure modes and compare its performance with IGR in the next chapter.

3 Experiments

In this section, we provide empirical verification for the our unified theoretical view of IGR and FGSM as well as the effect of smooth activations on both methods.

3.1 Setup

We follow the setup from [14] and trained PreActResNet18 models on the CIFAR-10 dataset using two primary training objectives: (1) Input Gradient Regularization (IGR) with $\lambda = 1.5$ and the attack

Method		Accuracy (%)		
Training	Activation	Clean	PGD-50	AutoAttack-L_∞
IGR	ReLU	61.10	38.80	32.20
	PReLU	58.80	37.60	30.90
	GELU	82.50	44.40	39.30
	Softplus	77.20	44.90	37.10
FGSM	ReLU	85.60	0.00	0.00
	GELU	78.80	47.30	41.90

Table 1: Comparison of Clean and Robust Accuracy across different training methods and activation functions. All models were trained with $\epsilon = 8/255$. Robustness is evaluated using PGD-50 and AutoAttack (L_∞).

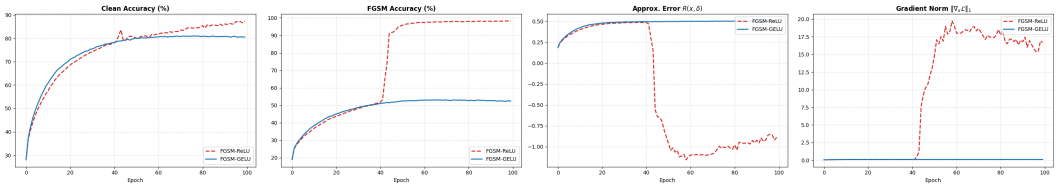

Figure 1: Comparison of first-order approximation error $R(x, \delta)$ and input gradient norm $\|\nabla_x \mathcal{L}(x)\|_1$ during FGSM training for ReLU and GELU activations. The ReLU network exhibits large negative approximation errors and exploding gradient norms after epoch 40, demonstrating catastrophic overfitting, while GELU maintains stable behavior throughout training.

strength $\epsilon = 8/255$, and (2) FGSM adversarial training with attack strength $\epsilon = 8/255$. In particular, we use the same architecture and training hyperparameters as in the original paper, with the exception that all models are trained for 100 epochs instead of 300.

The evaluation metric also follows [14], where we measure the clean accuracy, accuracy against PGD-50, and accuracy against AutoAttack [2], the current state-of-the-art ensemble of parameter-free attacks (APGD-CE, APGD-T, FAB-T, Square) for robustness.

A key novelty of our experimental design is the selection of activation functions. While [14, 19] primarily compared ReLU against GELU or SiLU, which share very similar characteristics, we extend this analysis to include Softplus and PReLU to further isolate the effect of smoothness on IGR's performance. In contrast to GeLU and SiLU, Softplus is monotonic and essentially a smooth approximation of the ReLU, whereas PReLU is a parametric piecewise linear activation that is non-smooth.

We also showcase the effect of smooth activations on FGSM-based training by plotting the approximation error $R(x, \delta)$ and gradient norm $\|\nabla_x \mathcal{L}(x)\|_1$ to epoch graph for ReLU and GeLU. To the best of our knowledge, this visual comparison is also novel.

3.2 Results

3.3 Discussion

4 Conclusion

References

[1] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training, 2020.

[2] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, 2020.

[3] H. Drucker and Y. Le Cun. Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6):991–997, 1992.

[4] Chris Finlay and Adam M Oberman. Scaleable input gradient regularization for adversarial robustness, 2019.

[5] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[6] Alexander G. Ororbia II, C. Lee Giles, and Daniel Kifer. Unifying adversarial training algorithms with flexible deep data gradient regularization, 2016.

[7] Peilin Kang and Seyed-Mohsen Moosavi-Dezfooli. Understanding catastrophic overfitting in adversarial training, 2021.

[8] Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training, 2020.

[9] Chunchuan Lyu, Kaizhu Huang, and Hai-Ning Liang. A unified gradient regularization family for adversarial examples. In *2015 IEEE International Conference on Data Mining*, pages 301–309, 2015.

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[11] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa, 2018.

[12] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning, 2017.

[13] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization, 2019.

[14] Adrián Rodríguez-Muñoz, Tongzhou Wang, and Antonio Torralba. Characterizing model robustness via natural input gradients, 2024.

[15] Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients, 2017.

[16] Ismaïla Seck, Gaëlle Loosli, and Stephane Canu. L 1-norm double backpropagation adversarial defense, 2019.

[17] Carl-Johann Simon-Gabriel, Yann Ollivier, Léon Bottou, Bernhard Schölkopf, and David Lopez-Paz. First-order adversarial vulnerability of neural networks and input dimension, 2019.

[18] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training, 2020.

[19] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V. Le. Smooth adversarial training, 2021.