Is Smoothness All You Need? Examining the Role of Smooth Activations in Efficient Robustness Trainings

Candidate Number: 1099950

Abstract

Robustness training is notorious for its computational cost. To address this, one-step training methods like Input Gradient Regularization (IGR) and Fast Gradient Sign Method (FGSM) have been revisited as alternatives, but their effectiveness has been inferior to multi-step methods, or even significantly compromised when enduring stronger attacks. In this project, we investigate the mathematical background of IGR and FGSM and how smooth activations could mitigate their limitations.

1 Introduction

Deep neural networks are very brittle to adversarial attacks; imperceptible perturbations to inputs can cause catastrophic misclassification [20]. Extensive research has been devoted to build robustness against adversarial attacks, with Adversarial Training via Projected Gradient Descent (PGD) [11] 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, one-step methods such as Input Gradient Regularization (IGR) and the Fast Gradient Sign Method (FGSM) have been revisited as efficient alternatives.

Although both methods fundamentally rely on a first-order approximation of the robust optimization framework, they have historically been viewed as distinct optimization strategies. In particular, FGSM has been considered data augmentation, whereas IGR as a regularization technique. Consequently, they are often analyzed separately with distinct failure modes. IGR often leads to inferior performance [17, 18, 13], while FGSM suffers from catastrophic overfitting [6], where the model becomes robust to the FGSM attack but performs disastrously (often 0% robustness) against other types of attacks.

Recent work by Rodríguez-Muñoz et al. [16] and Xie et al. [22] demonstrated respectively that IGR and FGSM can yield competitive results when using smooth activation functions like GeLU. However, their studies were mostly empirical and limited to IGR and activations with similar characteristics (GeLU and SiLU), leaving the broader role of smoothness in efficient training unexplored and the theoretical justification for their success unclear.

In this project, we propose a unified theoretical view of IGR and FGSM based on the local robustness bound established in [3] and explain how the smoothness of activations impact the performance of both methods. By extending the analysis to Softplus and PReLU, we confirm that smoothness is the key to success and show that it effectively mitigates catastrophic overfitting in FGSM, allowing it to achieve robustness comparable to IGR.

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 also investigate the role of smoothness in the performance of both methods in this theoretical framework.

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

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. [11] 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. [4] 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 drastically far from the actual loss landscape and leads to the aforementioned catastrophic overfitting [6]. 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 that directly penalizes the input gradient norm [10, 17, 5]. 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 [19] 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. We note that due to the gradient norm term, IGR requires double backpropagation, involving two passes of automatic differentiation: first to compute the gradient with respect to the inputs x, and second to differentiate this penalty term with respect to the model parameters θ [3].

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 [3], 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 [3])**.** *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 [3].

2.4 Smoothness is What You Need

Historically, IGR has struggled to achieve competitive robustness compared to PGD adversarial training [18]. 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. For non-smooth networks, the $\omega(\epsilon)$ can be large even for small ϵ as the local curvature may be unbounded, contradicting the requirement for robustness given by the robustness bound (6). This mathematically explains the phenomenon of gradient masking [13], 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 [3] 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. [16] and Xie et al. [22], 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. Xie et al. described this as a better gradient quality, but we may further elaborate this success as smooth activations bounding the local 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. demonstrated empirical evidence for the effectiveness of smooth activations, they only examined two smooth activation functions of extremely similar characteristics (GeLU and SiLU). We will experiment with other activation functions of different characteristics (e.g. Softplus, PReLU) to further isolate the effect of smoothness on IGR's performance and 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 [21, 1, 7, 15]. 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 [14, 15], and CURE [12]. While these methods are effective, they treat the FGSM-based training and IGR as distinct optimization goals. Simon-Gabriel et al. [19] 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 $\delta = \epsilon \cdot \operatorname{sign}(\nabla_x \mathcal{L}(x))$, which alters the training loss function to

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

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

$$\tilde{\mathcal{L}}_\epsilon(x, y; \theta) = \mathcal{L}(x, y; \theta) + \epsilon \|\nabla_x \mathcal{L}(x)\|_1 + R(x, \delta). \tag{8}$$

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 by minimizing $R(x, \delta)$ to be large and negative, while leaving the gradient norm $\|\nabla_x \mathcal{L}(x)\|_1$ large, contradicting 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 a 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 and the loss function (8) 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 achieve comparable robustness to IGR. This theoretical view is consistent with the experimental results in [22], 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. We train PreActResNet18 models on the CIFAR-10 dataset for IGR and FGSM with attack strength $\epsilon = 8/255$. For IGR training, we follow the setup and adapt the codes from [16], which uses the following objective:

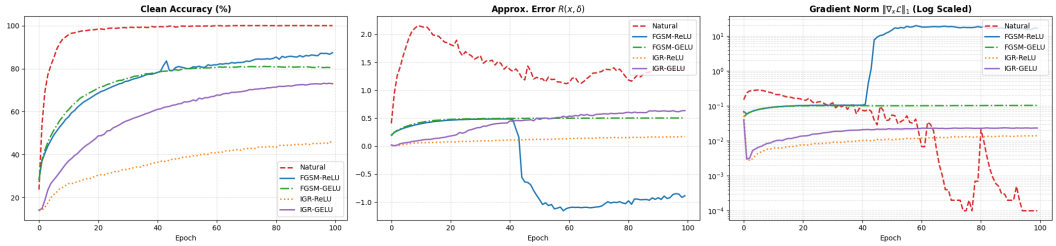
$$\mathcal{L}(x, y; \theta) = \lambda_{\text{CE}} \mathcal{L}_{\text{CE}}(x, y; \theta) + \lambda_{\text{GN}} \frac{\epsilon}{\sigma} \|\nabla_x \mathcal{L}_{\text{CE}}(x, y; \theta)\|_1,$$

where $\mathcal{L}_{\text{CE}}$ is the cross entropy loss, $\lambda_{\text{CE}} = 0.8$ and $\lambda_{\text{GN}} = 1.2$ are weighting hyperparameters, $\epsilon = \frac{8}{255}$ is the adversarial strength, and $\sigma = 0.2023$ is the standard deviation used for normalization on CIFAR-10. In particular, we change the activation function in the original code from GeLU to ReLU, Softplus, and PReLU. For FGSM training, we use the objective defined in (7) and use the cross entropy loss. Both training objectives are trained for 100 epochs (reduced from 300 in [16]). Same as [16], we employ a cosine learning rate scheduler [8] with a 5-epoch warmup and use the AdamW optimizer [9] with a weight decay of 0.05 and gradient clipping of 1.0. The evaluation metric also follows [16], where we measure the clean accuracy, accuracy against PGD-50, and accuracy against AutoAttack [2]. A key novelty of our experimental design is the selection of activation functions. While [16, 22] 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 $R(x, \delta)$ defined in (5) and $\|\nabla_x \mathcal{L}(x)\|_1$ to epoch graphs for ReLU and GeLU. A similar visualization for FGSM with ReLU is done in [15], but (to the best of our knowledge) its comparison to GeLU and IGR is novel.

Table 1: Accuracy results for Natural, FGSM, and IGR training after 100 epochs.

Method		**Accuracy (%)**		
Training	Activation	Clean	PGD-50	AutoAttack-L_∞
Natural	ReLU	95.40	0.00	0.00
FGSM	ReLU	85.60	0.00	0.00
	GELU	78.80	47.40	42.00
IGR	ReLU	60.50	37.10	30.90
	PReLU	58.80	37.60	30.90
	GELU	83.10	41.40	38.20
	Softplus	77.20	44.90	37.10

Figure 1: The plots from left to right are the clean accuracy, approximation error $R(x, \delta)$, and gradient norm $\|\nabla_x \mathcal{L}(x)\|_1$ to epoch graph for different training methods.



It is clear in Table 1 that both methods achieve significant improvements by adopting smooth activation functions. Softplus's close performance to GELU in IGR suggests that monotonicity is not a major factor in IGR's success. PReLU also fails to achieve better result than ReLU despite being a generalization of ReLU. This indicates that that smoothness almost everywhere is insufficient and even a slight discontinuity in the gradient could significantly reduce robustness. Both comparison yields further evidence that smooth activations are crucial for IGR's success. Meanwhile, FGSM achieves comparable robustness to IGR when using GeLU, confirming our guess earlier. One thing to note is the gap in input gradient norm $\|\nabla_x \mathcal{L}(x)\|_1$ between FGSM and IGR when using GeLU, which could be caused by the mismatch between the approximation error term $R(x, \delta)$ in (8) and the gradient norm penalty strength λ in (4). This suggests that the gradient norm penalty strength λ in (4) is not the optimal choice for IGR. We also see different failure modes for FGSM and IGR when using ReLU as activation. While the approximation error and input gradient norm are suppressed for IGR with ReLU in Figure 1, its clean accuracy and robustness against are noticeably worse than IGR with GeLU. This can be explained by the Finlay-Oberman bound (6): low clean accuracy implies small loss gap $\mathcal{L}_0 - \mathcal{L}(x)$, which in turn implies small lower bound for the magnitude of perturbation δ required to adversarially perturb an input x and thus the worsened robustness. On the other hand, our experiment perfectly captures the phenomenon of catastrophic overfitting and how the model "cheats" in FGSM-based training when using ReLU. As shown in Figure 1, $R(x, \delta)$ drops to the negative and the gradient norm $\|\nabla_x \mathcal{L}(x)\|_1$ explodes from near 0 to above 15. Paired with the 0% accuracy against PGD-50 and AutoAttack-L_∞ in Table 1, this confirms our view that the model exploits $R(x, \delta)$ to achieve superficial robustness.

4 Conclusion

Our experiments provide empirical evidence for the theoretical view of IGR and FGSM as well as the effect of smooth activations on both methods. According to our theoretical view and experimental results, FGSM may be a better alternative to IGR for robustness training, as it achieves comparable robustness to IGR while being computationally cheaper (FGSM does not need double backpropagation). However, due to limited computing resources, we are not able perform the experiments on modern large-scale models and datasets (ResNet-50 on ImageNet) as in [16]. Similarly, we are also unable to train models via multi-step adversarial training with smooth activations to give a direct evidence that IGR and FGSM can indeed achieve state-of-the-art robustness to Adversarial Training. It would be interesting to see if smooth activations can also improve multi-step adversarial training.

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

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

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

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

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

[8] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017.

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[10] 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.

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

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

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

[14] 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.

[15] Elias Abad Rocamora, Fanghui Liu, Grigorios G. Chrysos, Pablo M. Olmos, and Volkan Cevher. Efficient local linearity regularization to overcome catastrophic overfitting, 2024.

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

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

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

[19] 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.

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

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

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