LOW CURVATURE ACTIVATIONS REDUCE OVERFITTING IN ADVERSARIAL TRAINING

Vasu Singla, Sahil Singla, David Jacobs, Soheil Feizi
University of Maryland
{vsingla, ssingla, djacobs, sfeizi}@cs.umd.edu

ABSTRACT

Adversarial training is one of the most effective defenses against adversarial attacks. Previous works suggest that overfitting is a dominant phenomenon in adversarial training leading to a large generalization gap between test and train accuracy in neural networks. In this work, we show that the observed generalization gap is closely related to the choice of the activation function. In particular, we show that using activation functions with low (exact or approximate) curvature values has a regularization effect that significantly reduces both the standard and robust generalization gaps in adversarial training. We observe this effect for *both* differentiable/smooth activations such as Swish as well as non-differentiable/non-smooth activations such as LeakyReLU. In the latter case, the "approximate" curvature of the activation is low. Finally, we show that for activation functions with low curvature, the double descent phenomenon for adversarially trained models does not occur.

1 INTRODUCTION

Deep Neural Networks can be readily fooled by adversarial examples, which are computed by adding small perturbations to clean inputs (Szegedy et al., 2014). Adversarial attacks have been well studied in the machine learning community in recent years (Carlini & Wagner, 2017; Madry et al., 2018; Goodfellow et al., 2015). There have been several empirical and certified defenses proposed against adversarial attacks (Papernot et al., 2016; Song et al., 2018; Singla & Feizi, 2020). We focus on adversarial training (Madry et al., 2018), one of the most effective empirical defenses proposed in the literature. It has been shown that networks produced through vanilla adversarial training do not robustly generalize well (Schmidt et al., 2018; Rice et al., 2020; Farnia et al., 2018) and the gap between robust train and test accuracy i.e. the *robust generalization gap* can be far greater than the generalization gap that occurs during standard empirical risk minimization. Rice et al. (2020) showed that while traditional overfitting approaches such as l_1, l_2 regularization can reduce the robust generalization gap, no approach works better than simple early stopping.

The key observation of our work is that for smooth activation functions, the maximum of the second derivative or the maximum curvature has a significant impact on generalization. Specifically by using activations with low curvature, both *the robust and standard generalization gap* can be reduced, whereas with high curvature the gap increases. We also show similar observations for non-smooth activations, with low "approximate" curvature. Our results therefore show that the robust overfitting phenomenon can be mitigated with a properly chosen activation function without the need for early stopping (Rice et al., 2020). Lastly, we study the phenomenon of double descent for adversarially trained models (Nakkiran et al., 2019). In this phenomenon, when increasing model size, test accuracy first increases and then starts decreasing. However, upon reaching a critical point in model size known as the interpolation threshold, the test accuracy again starts increasing as model size increases. We show that double descent curves reported by Rice et al. (2020) for robust overfitting using ReLU do not hold for activation functions with low curvature such as Swish.

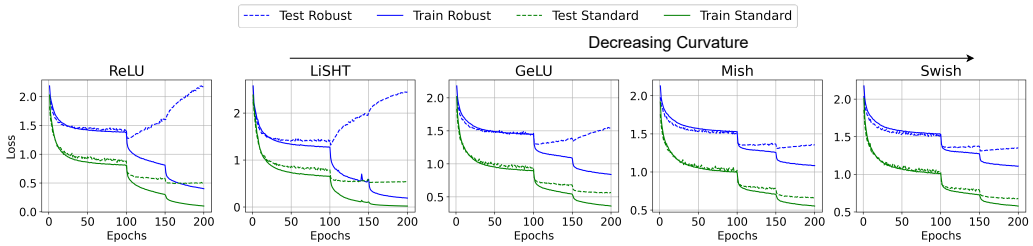


Figure 1: Learning curves for adversarially trained Resnet-18 models. ReLU activation is non-smooth and included as a baseline, all the other activations are ordered by decreasing curvature from left to right.

2 Impact of Activation Curvature on Adversarial Training

In this section we consider the effects of the curvature of smooth activation functions on standard and robust generalization gaps. We consider the smooth activation functions shown in Figure 2a. We define curvature as **the maximum of the second derivative** i.e $\max_x f''(x)$, and rank the activations by their curvature i.e LiSHT $>$ GeLU $>$ Mish $>$ Swish. More details regarding the activations are provided in Appendix B. We also conduct experiments for non-smooth ReLU activation as a baseline.

2.1 Smooth Activation Functions and Generalization Gap

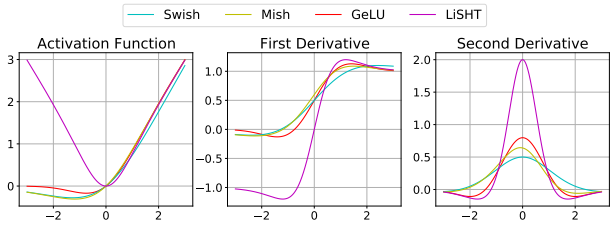
We show our results on the CIFAR-10 dataset (Krizhevsky et al.) with Resnet-18 (He et al., 2015) architecture for an l_∞ threat model with $\epsilon = 8/255$. We also systematically evaluate this for another family of activation functions with different curvatures in Appendix C.2. More experimental details are discussed in C.1.

In Figure 1 we show the learning curves for different activations, and reproduce the robust overfitting phenomenon for all activations (Rice et al., 2020). The robust training loss keeps decreasing, however robust test loss rises shortly after the first learning rate drop. For standard training and standard test loss however, both keep decreasing throughout training. This phenomenon shows the best performance for robust test accuracy is **not** achieved by training till convergence, unlike standard training. The best standard accuracy however, is still reached by training till convergence. In contrast to Xie et al. (2020) we also show that, LiSHT a smooth activation function performs worse than ReLU and shows a larger robust generalization gap. We also note that for activations that display a large robust generalization gap, the standard generalization gap is also higher. Finally, the *curvature of activation function has a direct impact on both the robust and standard generalization gaps*, as shown in the learning curves. Activations with high curvature such as LiSHT and GeLU have large generalization gap and activations with low curvature such as Mish and Swish have smaller generalization gap.

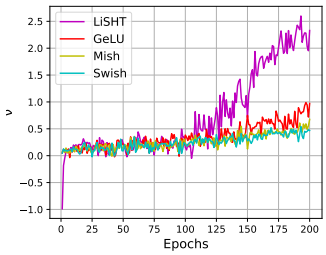
In Table 1 we show the robust and standard accuracies for the models. To show the gap due to robust overfitting, we also show the best robust accuracy using early stopping with validation set in the "Best Val" column. We also report the corresponding standard accuracy for the **best robust accuracy checkpoint** (not the best standard accuracy checkpoint). The robust generalization gap falls from 44.83% to 6.93% and standard generalization gap falls from 17.37 to 5.24%. This indicates the large impact of activation curvature on adversarial training. The decrease in robust performance caused by longer training (i.e best vs final checkpoint performance) also decreases for activations with smaller curvature. For example, the overfitting gap falls from 3.09% for LiSHT to 0.61% for Swish. Standard accuracy however, either remains the same or improves by training longer (compared to the best checkpoint).

2.2 Analyzing the influence of activations on robustness

In this section, we analyze the relationship between curvature of the activation function and adversarial robustness. We consider a simple two-layer neural network performing binary classification, represented as $f(x) = w_2^T \sigma(w_1 x)$ where $\sigma(\cdot)$ is a twice differentiable activation function and $\sigma''(\cdot)$



(a) Activation functions along with their first and second derivatives.



(b) Maximum eigenvalues for a batch of test examples.

	Robust Accuracy				Standard Accuracy			
Activation	Final Train	Final Test	Best Val	Diff.	Final Train	Final Test	Best Val	Diff.
LiSHT	92.27	47.44	50.53	**44.83**	99.90	82.53	82.44	**17.37**
ReLU.	82.46	49.77	51.61	**32.69**	98.9	83.73	81.62	**15.17**
GeLU	65.45	49.63	50.40	**15.82**	92.41	82.81	79.25	**09.60**
Mish	57.00	49.38	49.87	**07.62**	86.48	80.05	79.96	**06.43**
Swish	56.15	49.22	49.83	**06.93**	85.79	80.55	80.57	**05.24**

Table 1: Results for different activations on CIFAR-10 with ResNet-18. We use the best checkpoint based on **best robust accuracy** on the validation set shown in "Best Val" column. The generalization gap, i.e diff. between final train and test accuracy is shown in "Diff." column. Generalization gap for both standard and robust accuracy decreases for activations with decreasing curvature.

denotes its second derivative. Assume the final layer of the network outputs a single logit on which the sigmoid function is applied, given as $p(x) = \sigma(f(x))$. Assuming a sample is classified into class 1 if $p(x) < 0.5$, then a sample x is classified into class 1 iff $f(x) < 0$ and class 0 otherwise. We assume that the neural network can be locally well approximated using the second order Taylor expansion. Let x belong to class 1, then for $x+\delta$ to be classified as class 0, the minimal l_2 perturbation that fools the classifier can be written as:

$$\delta^* = \arg\min_{\delta} \|\delta\| \text{ s.t. } f(x) + \nabla_x f(x)^T \delta + \frac{1}{2}\delta^T \nabla_x^2 f(x) \delta \geq 0$$

It can be shown under these assumptions the magnitude of δ^* can be upper and lower bounded with respect to input curvature. We use the following lemma from Moosavi-Dezfooli et al. (2018) -

Lemma 1. *Let x be such that $c = -f(x) \geq 0$, and let $g = \nabla_x f(x)$. Let $\nu = \lambda_{max}\Big(\nabla_x^2 f(x)\Big) \geq 0$, denote the largest eigenvalue u be the eigenvector corresponding to ν. Then,*

$$\frac{\|g\|}{\nu}\left(\sqrt{1+\frac{2\nu c}{\|g\|^2}} - 1\right) \leq \|\delta^*\| \leq \frac{\|g^T u\|}{\nu}\left(\sqrt{1+\frac{2\nu c}{(g^T u)^2}} - 1\right) \tag{1}$$

The lemma shows that upper and lower bounds on the magnitude of δ^* increase, as v decreases, as shown in Moosavi-Dezfooli et al. (2018). An increase in $\|\delta^*\|$ therefore increases the minimum l_2 ball required to find an adversarial example for input x, leading to increased robustness. Therefore, a low maximum eigenvalue of the input Hessian leads to higher adversarial robustness assuming all other factors are kept constant.

We now show the relation between activation functions and input curvature. For the considered two layer neural network, the Hessian with respect to the input x is given as:

$$\nabla_x^2 f(x) = w_1^T diag\Big(\sigma''(w_1 x) \odot w_2\Big) w_1 \tag{2}$$

where $\odot$ denotes the Hadamard product between two vectors. Equation 2 shows that the Hessian of the input directly depends on $\sigma''(.)$, which suggests that an increase in the curvature of the activation function leads to an increase in the norm of the input Hessian. We observe this for adversarially trained Resnet-18 models in Fig. 2b; maximum eigenvalue is larger for activation with large curvature. Although we assume our activation to be smooth, we expect similar results for non-smooth activations. This result combined with our previous observation therefore suggests high activation curvature indeed leads to lower robustness.

k	Robust Accuracy			Standard Accuracy		
	Train	Test	Diff.	Train	Test	Diff.
0.5	52.7	49.0	**3.7**	82.9	79.5	**3.4**
0.3	63.0	50.1	**12.9**	92.0	83.5	**8.4**
0.2	69.6	49.6	**19.9**	95.3	84.2	**11.1**
0	82.4	49.7	**32.6**	98.9	83.7	**15.1**
-0.2	85.8	48.8	**37.0**	99.4	83.0	**16.4**

Table 2: Results for LeakyReLU activation function. Standard and robust generalization gap increases with increasing k.

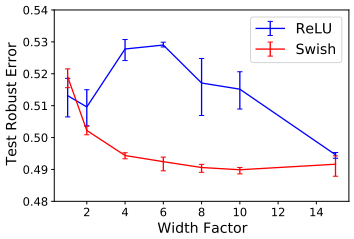


Figure 3: Generalization curves showing the double descent phenomenon occurs for ReLU but not Swish activation.

2.3 Does smoothness matter?

Xie et al. (2020) posit that smooth activations improve gradients achieving superior performance. In contrast, we show that the relation of the generalization gap to activations can be observed for non-smooth activations as well. We use the non-smooth LeakyReLU activation function defined as follows:

$$\text{LeakyReLU}(k, x) = \begin{cases} x & \text{if } x \geq 0 \\ kx & \text{if } x < 0 \end{cases}$$

where k is a hyper-parameter that can be tuned. For non-smooth LeakyReLU, we use the difference of slopes, i.e $\|1 - k\|$ as the "approximate" curvature of the function. Hence, for $k \leq 1$ the approximate curvature decreases with increasing value of k. See Figure 4b in Appendix for visualization of LeakyReLU. We observe behavior similar to smooth activations for LeakyReLU as shown in Table 2. For $k = 0.5$, the approximate curvature is low, and both robust and standard generalization gap, 3.74 and 3.43 respectively is much smaller than $k = -0.2$, for which robust and standard generalization gap are 37.07 and 16.46. We therefore hypothesize for non-smooth activations, the "approximate" curvature of the activation function has impact on the generalization gap.

3 Double descent curves

Generalization in deep learning typically has shown improved performance for increased model complexity beyond data interpolation point, known as *double descent* phenomenon (Belkin et al., 2019). Although model size and training time can both be viewed as increasing model complexity, double descent is observed with increasing model size, and training longer causes overfitting. They therefore posit that training longer and increasing architecture size have separate effects on robust generalization.

In Figure 3, we show the results for ReLU and the Swish[1] activation function using Wide Resnet-28 with different width factors. While the double descent phenomenon is observed for ReLU activation, robust test performance continues to decrease for the Swish activation function. We observe Swish with width 4 is able to match performance of ReLU with width 15 when trained till convergence. The results therefore suggest that activations with low curvature can act as a regularizer to mitigate the double descent phenomenon. Also robust test error for width 15 is equivalent for ReLU and Swish, suggesting that low curvature activations may not be useful for models with very large width.

4 Conclusion

In this work, we analyze the regularization effect of curvature of activation functions on adversarial training and show this extends to non-smooth activations as well. Our experiments also show that double descent, another phenomenon that has a significant impact on robust generalization can be mitigated using activations with low curvature. Since robust overfitting is common in adversarial training, the properties of activation functions that we bring to light in this work can be useful for state of the art robust models.

[1]Experiments with other activations could not be conducted due to the expensive training of Wide Resnets, so we use Swish activation function, because of its lowest curvature among all the activations considered.

REFERENCES

Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation, 2019.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. volume 80 of *Proceedings of Machine Learning Research*, pp. 274–283, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/athalye18a.html.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine learning practice and the bias-variance trade-off, 2019.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. 2018. URL https://openreview.net/pdf?id=S18Su--CW.

Rudy Bunel, Ilker Turkaslan, Philip H. S. Torr, Pushmeet Kohli, and M. Pawan Kumar. A unified view of piecewise linear neural network verification, 2018.

N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods, 2017.

Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing, 2019.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense, 2018.

Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks, 2017.

Farzan Farnia, Jesse M. Zhang, and David Tse. Generalizable adversarial training via spectral normalization, 2018.

Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. Detecting adversarial samples from artifacts, 2017.

Matteo Fischetti and Jason Jo. Deep neural networks as 0-1 mixed integer linear programs: A feasibility study, 2017.

T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2018. doi: 10.1109/SP.2018.00058.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J. Cree. Regularisation of neural networks by enforcing lipschitz continuity, 2020.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models, 2019.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2020.

Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks, 2017.

Guy Katz, Clark Barrett, David Dill, Kyle Julian, and Mykel Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks, 2017.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale, 2017.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy, 2019.

Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble, 2018.

Alessio Lomuscio and Lalit Maganti. An approach to reachability analysis for feed-forward relu neural networks, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations, 2017.

Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. volume 80 of *Proceedings of Machine Learning Research*, pp. 3578–3586, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/mirman18b.html.

Diganta Misra. Mish: A self regularized non-monotonic activation function, 2020.

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

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt, 2019.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training, 2020a.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Hang Su, and Jun Zhu. Boosting adversarial training with hypersphere embedding, 2020b.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks, 2016.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017.

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning, 2020.

Swalpa Kumar Roy, Suvojit Manna, Shiv Ram Dubey, and Bidyut Baran Chaudhuri. Lisht: Non-parametric linearly scaled hyperbolic tangent activation function for neural networks, 2020.

Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. A convex relaxation barrier to tight robustness verification of neural networks, 2020.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data, 2018.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free!, 2019.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3(POPL), January 2019. doi: 10.1145/3290354. URL https://doi.org/10.1145/3290354.

Sahil Singla and Soheil Feizi. Second-order provable defenses against adversarial attacks, 2020.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming, 2019.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses, 2020.

Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope, 2018.

Eric Wong, Frank R. Schmidt, Jan Hendrik Metzen, and J. Zico Kolter. Scaling provable adversarial defenses, 2018.

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

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization, 2018.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness, 2019.

Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V. Le. Smooth adversarial training, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle, 2019a.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482, Long Beach, California, USA, 09–15 Jun 2019b. PMLR. URL http://proceedings.mlr.press/v97/zhang19p.html.

Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger, 2020.

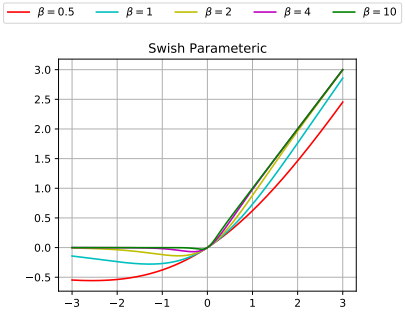
A Related Works

Goodfellow et al. (2015) provided one of the first approaches for adversarial training based on generating adversarial examples through the fast sign gradient method (FGSM). Building on this, a stronger adversary known as basic iterative method Kurakin et al. (2017) was proposed in subsequent work, using multiple smaller steps for generating adversarial examples. Madry et al. (2018) extended this adversary with multiple random restarts to train models on adversarial data, referred to as projected gradient descent (PGD) adversarial training. Further works have focused on improving the performance of the adversarial training procedure with methods such as feature denoising (Xie et al., 2019), hypersphere embedding (Pang et al., 2020b) and using friendly adversarial data (Zhang et al., 2020). The TRADES objective (Zhang et al., 2019b) balances standard and robust error achieving state of the art performance for adversarial training. However Rice et al. (2020) showed that the performance of TRADES can be matched using simple early-stopping. Another recent work challenges this study and shows that with modifications to the training framework such as weight decay and batch-normalization (Pang et al., 2020a), TRADES again achieves state of art performance. A separate line of works has focused on speeding up adversarial training due to its increased time complexity, by reducing attack iterations and computational complexity for calculating gradients (Zhang et al., 2019a; Shafahi et al., 2019; Wong et al., 2020).

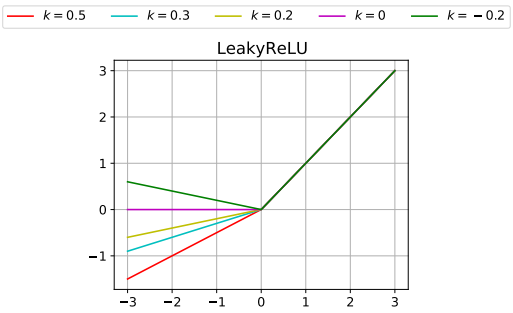
Besides adversarial training, several other defenses have been proposed such as defensive distillation (Papernot et al., 2016), preprocessing techniques (Guo et al., 2018; Song et al., 2018; Buckman et al., 2018) and randomized transformations (Xie et al., 2018; Dhillon et al., 2018; Liu et al., 2018) or detection of adversarial examples (Metzen et al., 2017; Feinman et al., 2017). However these methods were later broken by stronger adversaries (Athalye et al., 2018; Tramer et al., 2020; Carlini & Wagner, 2017). These defense methods were shown to rely on obfuscated gradients (gradient masking), which provided a false sense of security. Due to the bitter history of gradient masking as a defense, Xie et al. (2020) proposed use of smooth activations with a single step PGD attack to improve adversarial robustness, reaching state of the art robust performance on ImageNet (Deng et al., 2009). Xie et al. (2020) hypothesize that using smooth activations provides networks with better gradient updates and allows adversaries to find harder examples.

Since many defenses proposed in the literature have been broken, another separate line of work has focused on certified defenses, which can guarantee robustness against adversarial attacks for different norms such as l_2 or l_∞. Some of these techniques however are not scalable to large neural networks. The various different methods proposed in the literature use techniques such as mixed-integer programming methods (Tjeng et al., 2019; Lomuscio & Maganti, 2017; Fischetti & Jo, 2017; Bunel et al., 2018) and satisfiability modulo theories (Katz et al., 2017; Ehlers, 2017; Huang et al., 2017). Some certification methods bound the global Lipschitz constant of the network. Such bounds are usually loose for large neural networks with multiple layers (Anil et al., 2019; Gouk et al., 2020). Another line of work has focused on providing loose certificates, which leverage techniques such as randomized smoothing (Cohen et al., 2019; Lecuyer et al., 2019), abstract representations (Gehr et al., 2018; Mirman et al., 2018; Singh et al., 2019), interval bound propagation (Gowal et al., 2019) and duality and linear programs (Salman et al., 2020; Wong & Kolter, 2018; Wong et al., 2018).

Lack of overfitting in overparameterized deep learning models is an intriguing phenomenon for deep learning (Zhang et al., 2017). These models can be trained to effectively zero training error, without having impact on test time performance. Hence, it is now standard practice in deep learning to train longer and use large overparameterized models, since test accuracy generally improves past an interpolation point also known as double descent generalization (Belkin et al., 2019; Nakkiran et al., 2019). Schmidt et al. (2018) however have shown that sample complexity required for adversarially robust generalization is significantly higher than sample complexity for standard generalization. In a recent work, Rice et al. (2020) have shown the overfitting phenomenon to be dominant in adversarial training. In their work, they show after training for a certain period, the model starts to overfit and robustness decreases on the test set, and even double descent generalization curves seemed to hold. Rice et al. (2020) also tried various regularization techniques to prevent robust overfitting, among which early-stopping was the most effective solution.



(a) PSwish with different β values.



(b) LeakyReLU with different k values.

β	Robust Accuracy			Standard Accuracy		
	Train	Test	Diff.	Train	Test	Diff.
0.5	47.00	45.86	**1.14**	75.39	73.57	**1.82**
1	56.15	49.22	**6.93**	85.79	80.55	**5.24**
2	69.65	49.96	**19.69**	94.57	83.39	**11.18**
4	83	50.11	**32.89**	98.82	84.48	**14.34**
10	89.2	50.91	**38.29**	99.7	83.57	**16.13**

Table 3: Performance of PSwish with different β values, higher β value indicates higher curvature. Results are shown for final checkpoint and show that for activations with high curvature, standard and robust generalization gap increases.

B Activation Functions

We use the activation functions defined below, ranked by their curvature:

1. **Linearly Scaled Hyperbolic Tangent** (LiSHT) (Roy et al., 2020): $f(x) = x * \tanh(x)$, this function has highest curvature among activations considered.
2. **Gaussian Error Linear Unit** (GeLU) (Hendrycks & Gimpel, 2020): $f(x) = x * \Phi(x)$, where $\Phi(x)$ is gaussian cummulative distribution function.
3. **Mish** (Misra, 2020): $f(x) = x * \tanh(ln(1 + \exp(x)))$ is a smooth continuous function similar to Swish.
4. **Swish** (Ramachandran et al., 2017): $f(x) = x * sigmoid(x)$ is a smooth approximation to ReLU with a non-monotonic "bump" for $x < 0$.

C More Experiments

C.1 Experimental Details

Experimental Settings - For comparison with best early-stop checkpoint Rice et al. (2020), we randomly split the original set into training and validation set with 90% and 10% images respectively. We consider the l_∞ threat model and use PGD-10 step attack with $\epsilon = 8/255$ and $\alpha = 2/255$ for reporting the train and test accuracy. We use the ResNet-18 He et al. (2015) architecture for all our experiments except for experiments with double descent curves where we use Wide ResNet-28 Zagoruyko & Komodakis (2017). We use the same training setup as Rice et al. (2020) throughout the paper, an SGD optimizer with momentum of 0.9, weight decay 5×10^{-4} for 200 epochs with batch size of 128.

C.2 Analyzing curvature effects with Parameteric Swish

To further understand the impact of activation curvature on standard and robust generalization gap, we conduct analysis with *Parameteric Swish (PSwish)*, defined as follows:

$$f(x) = x \cdot sigmoid(\beta x)$$

The Swish function defined previously is a special case of PSwish, when $\beta = 1$. PSwish transitions from the identity function for $\beta = 0$, to ReLU for $\beta \to \infty$. The curvature of PSwish increases as β increases. Figure 4a shows the PSwish activation function for different values of β.

We show the results with the CIFAR-10 dataset, for final checkpoints for training and testing set in Table 3. Interestingly, we observe that both the standard and robust generalization gap are extremely dependent on the choice of β. The robust generalization gap increases from 1.14 to 38.29 and the standard generalization gap increases from 1.82 to 16.13 for $\beta = 0.5$ and $\beta = 10$ respectively. We also observe that robust test accuracy for the final checkpoint increases from 45.86 to 50.91 for the same β values. For larger values of β i.e $\beta \to \infty$, PSwish behaves like ReLU and standard and robust final test accuracy start decreasing. The results are consistent with our previous experiments and show that the standard and robust generalization gap increases for activations with high curvature. Further using the early stopping checkpoint with the validation set, PSwish with $\beta = 10$ outperforms ReLU baseline by 0.7% on robust accuracy and 1.24% on standard accuracy, highlighting that the choice of activation function can improve standard and robust performance for adversarially trained models.