Sample Complexity Scaling Laws for Adversarial Training

Chris Yuhao Liu
University of California, Santa Cruz
yliu298@ucsc.edu

Abstract

The emergence of gradient-based adversarial training algorithms has made state-of-the-art deep neural networks robust to attacks and achieve far better accuracy than non-adversarially trained ones. Even if models are trained on inputs with "worst-case noise," their performance drop on the original clean test data is trivial. However, it is unknown if adding such noise could affect the sample complexity rate, a scaling law of training data size versus test error. In this work, we focus on examining the effect of FGSM and PGD on the sample complexity scaling law with a variety of model architecures on MNIST and CIFAR-10 datasets. We empirically show that, in almost all cases, both adversarial training methods make the sample complexity rate slower, requiring up to several orders of magnitude more training data to reach a certain level of performance as in standard training. We later identify that such slowness is mainly caused by the perturbation budget, ϵ, which implies a trade-off between sample-efficiency and robustness. We believe our empirical findings will provide further insights to design more sample-efficient adversarial training algorithms.

1 Introduction

In the last few years, many adversarial training techniques have been proposed and proved to be effective in making neural networks more robust in a wide range of domains and downstream tasks. Albeit its success, researchers also identify a trade-off between robustness and both accuracy [10, 17] and sample complexity [12], indicating the former may not be a simple additive (free) property.

In this proposal, we focus on the sample complexity aspect. While [12] and [13] prove that adversarial training has higher sample complexity, the speed with which it scales with adversaries is rarely studied. Therefore, we ask the following questions:

1. Does adversarial training affect the scaling law of sample size vs. standard/robust error?
2. If adversarial training reduces sample efficiency, what is the cause?
3. How much more data points do we need for adversarial training to reach the same robust/standard error as standard training?

To answer these questions, we propose to quantify the trade-off between adversarial robustness and sample complexity by employing the notion of empirical sample complexity rates, a monotonic-decreasing function of generalization error with respect to the clean and adversarial training data size. More specifically, we will use Fast Gradient Sign Method [6] and Projected Gradient Descent [9] as our adversarial training backbone. Our contributions can be summarized as follows:

1. In almost all dataset and model combinations, given attacks during testing, both FGSM and PGD worsen the sample complexity rate. If no attack is used during testing, the sample complexity rate is still slower in some cases.
2. ϵ is the only factor that affects the exponent empirical sample complexity rate. A larger ϵ leads to a slower sample complexity rate, and vice versa. For PGD, the step size and the number of steps only produce shifts in the intercept.
3. To reach the same accuracy as in standard training, we need dataset sizes ranging from a few times to several order of magnitudes larger.

2 Background and Related Work

2.1 PAC learnability and sample complexity rate

Given a finite hypothesis class H and training data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\} \subseteq (\mathcal{X} \times \mathcal{Y})^m$, a standard empirical risk minimization (ERM) algorithm tries to find an $h \in H$ with the smallest empirical error and output learner L [2].

Realizability assumption There exists a learnable target function (i.e., the labeling function of $\mathcal{D}$) $t^* \in H$ such that, for all $i = 1, 2, \cdots, m$, the hypothesis $h = L$ satisfies $h(x_i) = t(x_i)$. If the realizability assumption holds and H is is finite, it is with high probability at least $1 - \delta$ that the learner L achieves some small error ϵ with m examples, where a lower bound of m can be written as follows.

$$m \geq \frac{1}{\epsilon} \ln(\frac{|H|}{\delta}) \tag{1}$$

Realizable case The realizability assumption always holds in the realizable case in that it is assumed that a hypothesis h matches the error on a target hypothesis t. In other words, an optimal hypothesis lies in our hypothesis class. We call such L a learning algorithm for H with confidence $1 - \delta$ and error ϵ, and the sample complexity lower bound is

$$m \geq \frac{1}{\epsilon} \ln(\frac{|H|}{\delta}). \tag{2}$$

Agnostic case It is usually not the case that the labeling function of the dataset $\mathcal{D}$ lies in our predefined hypothesis class H. In other words, it lacks a low-error classifier in the search space of the learning algorithm. In such case, we hope that the learner L can achieve empirical error $\min_{h \in H} \mathcal{E}(h)$ with confidence $1 - \delta$ and error ϵ with sample complexity upper bound

$$m \leq \frac{2}{\epsilon^2} \ln\left(\frac{2|H|}{\delta}\right).$$

The main difference between the two cases is in $1/\epsilon$ and $1/\epsilon^2$, indicating that less data is needed in the realizable case because the hypothesis space is dependent of the data.

Sample Complexity Rate Given a model f_θ trained with n examples using standard training, it can achieve ϵ generalization error, where ϵ is characterized by a power-law function $\mathcal{E}$ of training data size n.

$$\mathcal{E}(n) = \alpha + \beta \cdot n^\gamma \tag{3}$$

Here α, β, and γ are all constants. The exponent γ remains approximately $-1/2$ in most real-world agnostic setups [5, 3], reflecting an asymptotic decay rate of $O(\sqrt{n})$. However, γ can reach as fast as -1 with additional assumptions (e.g., Tsybakov low noise condition [14]), leading to a fast asymptotic decay rate of $O(1/n)$, even if the realizability assumption does not completely.

In this work, we are always in the agnostic case with the slow rate $O(\sqrt{n})$ (or even slower), as are most practical machine learning problems. We want to verify if adversarial training will change the rate, and if it does, to which direction. A theory of the realizable case $O(1/n)$ and agnostic case $O(\sqrt{n})$ is provided in the appendices.

2.2 Adversarial training and sample complexity

Adversarial training is, in some cases, observed to increase the sample complexity. [12] studies if a robust classifier can be trained using only a standard dataset sufficient for training a usually good one. They provide information-theoretic lower bounds on the exact sample complexity required and demonstrate a sample polynomial complexity increase empirically on MNIST and CIFAR-10. They conclude that, for a model to be adversarially robust, more training data are needed. [13] further uses synthetic data to show that adversarial training can only achieve the same accuracy as standard training in the potentially unlimited data regime.

While it still lacks theoretical justification, [8] studies adversarial sample complexity in the context of the multi-source dataset and PAC learning. They show that, in a single-source case, PAC learnability can be easily broken by a strong enough adversary. While they prove that, as long as less than half sources become adversarial, the sample complexity remains upper bounded, a slow convergence rate is almost unavoidable, indicating a necessary growth of per source sample size.

Interestingly, [11] and [4] later find that the training data size requirements relate to the strength of adversaries. More specifically, adding more training data can improve generalization performance in the weak and median adversary regime. However, in the strong regime and later part of the median regime, the performance degrades as more data are added, contradicting [12]. In other words, the data distribution and adversarial strength are the two variables controlling the generalization gap with respect to training data size.

3 Method

In this work, we take an explanatory approach to quantify the trade-off between adversarial robustness (on FGSM and PGD) and the empirical sample complexity rate introduced in 2.1.

3.1 Measuring sample complexity rate

To measure the empirical sample complexity rate of a model on a given dataset, we train the model on randomly sub-sampled training examples of different sizes and then evaluate the model on the full test set for accuracy. For example, the training set of CIFAR-10 consists of 50000 examples. We use randomly sub-sampled dataset sizes $S = \{2500, 5000, ..., 47500, 50000\}$ for training. We randomly initialize a model for each size and repeat the training and evaluation process for T trials to get the averaged test error. In this paper, we use $T = 10$ for all experiments. A larger T could make the curve smoother and easier to fit, and $T = 10$ is good enough in our case. Lastly, we plot the training sample sizes versus their corresponding expected test error.

To measure the exponent of a sample complexity curve, we use non-linear least squares to fit the error function $\mathcal{E}_{adv}(n_{adv}) = \alpha + \beta \cdot (n_{adv})^{\gamma}$, where n_{adv} denotes the sample size and γ is the empirical sample complexity rate we want to measure.

3.2 Adversarial training

We will largely resemble the setting in [10], which is considered a formal and universal adversarial training framework in recent years. More specifically, we will mainly measure the robustness-sample complexity trade-off on two types of adversarial training methods: 1) Fast Gradient Sign Method (FGSM) [6], and 2) Projected Gradient Descent (PGD) [9].

FGSM FGSM first computes the sign of the gradient with respect to the loss of a model parameterized by θ on the original example (x, y). To get the perturbed input, we perform a point-wise additive operation on the input x and the gradient sign multiplied by some small ϵ, which scales down the perturbation strength.

$$x + \varepsilon \operatorname{sgn}\left(\nabla_x L(\theta, x, y)\right)$$

Projected Gradient Descent The formulation of projected gradient descent introduced in [10] is essentially a k-step FGSM with negative loss. It tries to find some perturbation with a maximized

model loss on a particular example. It is also cast as a constraint optimization problem, which bounds the perturbation strength in some small value ϵ.

$$x^{t+1} = \text{Proj}_{x+\mathcal{S}}\left(x^t + \alpha \operatorname{sgn}\left(\nabla_{x^t} L\left(\theta, x^t, y\right)\right)\right)$$

4 Experiments

In this section, we briefly introduce the experiment setting and proceed to the results.

Models and training Model wise, we consider two simple model architectures, an MLP and a CNN. The MLP has two layers, each with 512 hidden units, whereas the CNN has two convolutional layers of 32 and 64 filters with filter size = 5, stride = 1, and same paddind, followed by a feed forward layer of 512 hidden units. Additionally, we also include three advanced convolutional architectures: ResNet-18 [7], ResNeXt-50 32x4d [15], and Wide ResNet-50 with 2x width [16]. To better study the change in sample complexity rate as we make the models wider and deeper, we add model variants for MLP and CNN by using 2x and 4x the hidden units/filters, and ResNet-34 and ResNet-50.

Standard training With standard training, the models are trained on clean training data and clean test data. For all models and datasets combinations, we use batch size = 128 and learning rate = 1e-3. The maximum number of epochs is 50 with early stopping on validation data. The validation data is a 10% random split of each sampled subset, and the training is stopped when there is no validation accuracy improvement in the previous 10 epochs. For adversarial training experiments, except for the adversarial defense and attack parts, they will follow the exact same setup here.

Adversarial training For FGSM, we use $\epsilon = 0.3$ for MNIST and $\epsilon = 0.03$ for CIFAR-10 and ℓ_∞ norm, following [10]. For PGD, we use the same ϵ's as in FGSM, and step size = 0.1 for 10 steps. We compute the averaged accuracies of both the standard testing (without attack) and robust testing (with the corresponding attack). Additionally, we also tune the three hyperparameters, ϵ, step size, and the number of steps to observe how they affect the scaling law.

5 Results

5.1 Empirical sample complexity rates of adversarial training

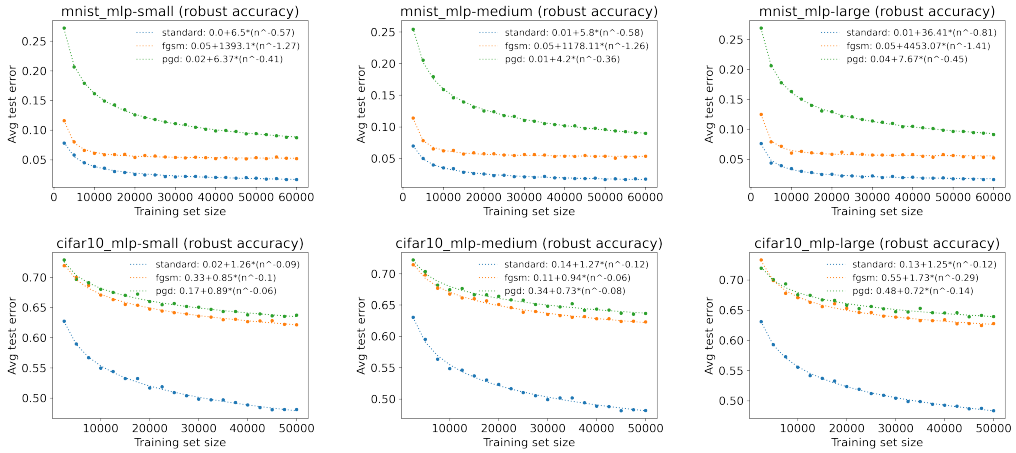

Figure 1: Sample complexity rates of varying-width two-layer MLPs trained on MNIST and CIFAR-10 datasets, including both standard training and adversarial training. "Robust accuracy" indicates that attacks are used at test time.

In Figure 1, for two-layer MLPs with different widths, PGD training almost always results in slower complexity rates on both MNIST and CIFAR-10, with the exception of MLP-Large on CIFAR-10

being slightly faster. However, the behavior for FGSM is different. For MNIST, FGSM training consistently remains fast across three MLP sizes; for CIFAR-10, MLP-Small and MLP-Large also have faster sample complexity rates. Our hypothesis for the later is that MLP may not be sufficient to model CIFAR-10 well due to its simplicity, posing an implicit regularization effect to training. As we'll see in later figures, it is not the case for CNN architectures. For the former case, we still need to investigate more about what exactly causes the fast rates of FGSM training on MNIST.

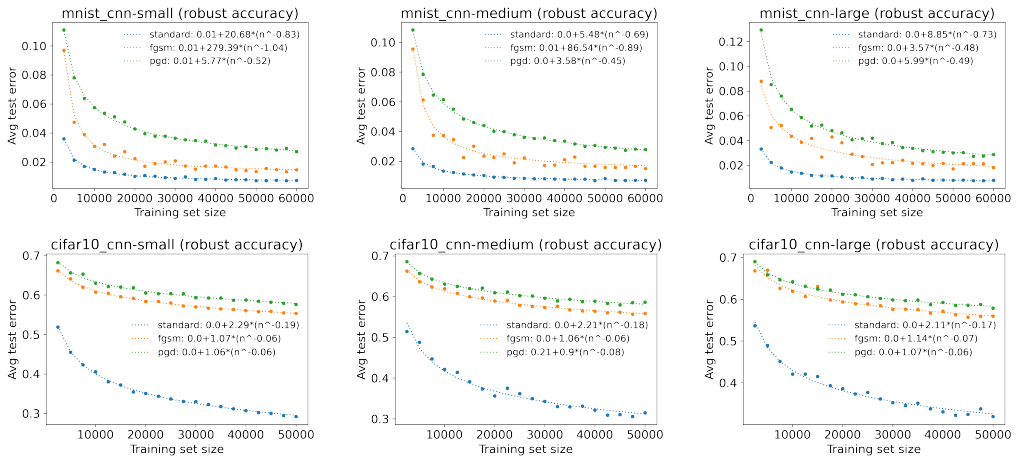

Figure 2: Sample complexity rates of varying-width two-layer CNNs trained on MNIST and CIFAR-10 datasets, including both standard training and adversarial training.

As shown in Figure 2, CNN behave similarly on the MNIST dataset, but the rate of FGSM starts to decrease we gradually increase the model size. This may imply that this phenomenon is related to the implicit regularization imposed by the model itself. The noisiness of the curves might be due to the "catastrophic overfitting" [1], making error rates of some sample sizes significantly higher. On CIFAR-10, adversarially trained CNNs consistently show slower sample complexity rate for both FGSM and PGD than in the standard training setting.

For ResNets models with different depths and their variants, the patterns are more consistent, where adversarially trained counterparts all have a slower sample complexity rate, requiring more data to reach the same level of performance, regardless of the model size and the adversarial training method.

5.2 Empirical sample complexity rates on clean test sets

We further study the sample complexity change when there is no attack at the test time on CIFAR-10. We observe that, on MNIST, the change pattern is not consistent. Albeit trivially, adversarial training usually improve the rate while degrading the accuracy. This behavior is different from what we observed in the attack-in-test scenario. However, with CNN architectures, adversarial training also harm the sample complexity rate even if attacks are absent during testing, but the extent is not as significant as in the previous case. The same observation applies to ResNets, ResNeXt, and WideResNet, but they are not as slow as the case of attack during testing.

5.3 Effects of epsilon, step size, and number of steps

We also demonstrate that ϵ is the only parameter that can make the model require more data to train when using adversarial training. Increasing the step size of the number of steps for PGD trivially affects the sample complexity rate. This partially indicate that it is the data transformation that hinders the rate, but neither the speed of training nor the training time. As the value of ϵ increases, the sample complexity rate generally decreases. This results follows the theory in [14], as adversarially transformed data can be considered as worst-case noisy data.

5.4 How much extra data do we need?

As we shown in the previous sections, while adversarial training offers greater robustness, the side effect is not limited in accuracy but also in the number of data points required to achieve an accuracy as high as in the standard training. To better understand how much more data we need, we provide this statistic in B.

For all tables, we calculate the number of data points required to reach the same level of error in standard training with 100 examples. Because the accuracy of adversarial training is typically lower than that of standard training, we assume that the intercept α and β in the equation $\mathcal{E}_{adv}(n_{adv}) = \alpha + \beta \cdot (n_{adv})^{\gamma}$ can be ignored by using tricks in practice to improve accuracy. The numbers in the FGSM and PGD columns are calculated by solving $100^{\text{standard rate}} = n^{\text{FGSM/PGD rate}}$.

We observe that, for MNIST and CIFAR-10 datasets, both FGSM and PGD make the empirical sample complexity rate slower on a variety of model architectures, requiring up to approximately $10e5$ times more data to achieve a robust accuracy as high as in standard training. For standard accuracy, while the data requirement is much less, it is still tens or hundreds times the original data size.

6 Discussion and Conclusion

Our exploration demonstrates qualitative difference in the empirical sample complexity rate between standard and adversarial training. Almost all evidence support the hypothesis that models trained with FGSM and PGD have slower sample complexity rate. While models are more robust to adversarial attacks, we show that adversarial training do not simply harm the accuracy but, additionally, requires more data to reach the same performance, regardless of model size. Interestingly, this corresponds to the finding that adding more noise could makes the sample complexity rate slower.

We further reveal that ϵ is the only parameter that slows down the sample complexity rate, and the higher the ϵ, the slower the rate. This suggests that robustness can be traded for sample efficiency by using a smaller ϵ. This property may imply that, other than injecting more noisy examples into the dataset, increasing the difficulty of the original examples using adversarial noise also slows down the sample complexity rate.

Lastly, we also show that, to reach the same accuracy as in standard training, we need dataset sizes ranging from a few times to several order of magnitudes larger. We hope to encourage future research on modifying other existing properties of adversarial training to mitigate the issue of requiring more data and the design of new sample-efficient adversarial training methods to obtain better robustness and sample complexity rate at the same time.

References

[1] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *arXiv preprint arXiv:2007.02617*, 2020.

[2] Martin Anthony, Peter L Bartlett, and Peter L Bartlett. *Neural network learning: Theoretical foundations*, volume 9. cambridge university press Cambridge, 1999.

[3] Shai Ben-David and Ruth Urner. The sample complexity of agnostic learning under deterministic labels. In Maria-Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors, *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, volume 35 of *JMLR Workshop and Conference Proceedings*, pages 527–542. JMLR.org, 2014.

[4] Lin Chen, Yifei Min, Mingrui Zhang, and Amin Karbasi. More data can expand the generalization gap between adversarially robust and standard models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1670–1680. PMLR, 2020.

[5] Simon S Du, Yining Wang, Xiyu Zhai, Sivaraman Balakrishnan, Ruslan Salakhutdinov, and Aarti Singh. How many samples are needed to estimate a convolutional or recurrent neural network? *ArXiv preprint*, abs/1805.07883, 2018.

[6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015). *ArXiv preprint*, abs/1512.03385, 2015.

[8] Nikola Konstantinov, Elias Frantar, Dan Alistarh, and Christoph Lampert. On the sample complexity of adversarial multi-source PAC learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5416–5425. PMLR, 2020.

[9] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[10] 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.

[11] Yifei Min, Lin Chen, and Amin Karbasi. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization. *ArXiv preprint*, abs/2002.11080, 2020.

[12] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5019–5031, 2018.

[13] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6976–6987. Computer Vision Foundation / IEEE, 2019.

[14] Alexander B Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

[15] Saining Xie, Ross B Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. corr abs/1611.05431 (2016). *ArXiv preprint*, abs/1611.05431, 2016.

[16] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. BMVA Press, 2016.

[17] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019.