On the Generalization Properties of Adversarial Training

Yue Xing
Purdue University

Qifan Song
Purdue University

Guang Cheng
Purdue University

Abstract

Modern machine learning and deep learning models are shown to be vulnerable when testing data are slightly perturbed. Existing theoretical studies of adversarial training algorithms mostly focus on either adversarial training losses or local convergence properties. In contrast, this paper studies the generalization performance of a generic adversarial training algorithm. Specifically, we consider linear regression models and two-layer neural networks (with lazy training) using squared loss under low-dimensional and high-dimensional regimes. In the former regime, after overcoming the non-smoothness of adversarial training, the adversarial risk of the trained models can converge to the minimal adversarial risk. In the latter regime, we discover that data interpolation prevents the adversarially robust estimator from being consistent. Therefore, inspired by successes of the least absolute shrinkage and selection operator (LASSO), we incorporate the $\mathcal{L}_1$ penalty in the high dimensional adversarial learning and show that it leads to consistent adversarially robust estimation. A series of numerical studies are conducted to demonstrate how the smoothness and $\mathcal{L}_1$ penalization help improve the adversarial robustness of DNN models.

1 INTRODUCTION

Recent advances in deep learning and machine learning have led to breakthrough performance and are widely applied in practice. However, empirical experiments show that deep learning models can be fragile and vulnerable against adversarial input which is intentionally perturbed (Biggio et al., 2013; Szegedy et al., 2014). For instance, in image recognition, a deep neural network will predict a wrong label when the testing image is slightly altered, while the change is not recognizable by the human eye (Papernot et al., 2016b). To ensure the reliability of machine learning and deep learning when facing real-world inputs, the demand for robustness is increasing. The related research efforts in adversarial learning include designing adversarial attacks in various applications (Papernot et al., 2016b,a; Moosavi-Dezfooli et al., 2016), detecting attacked samples (Tao et al., 2018; Ma and Liu, 2019), and modifications on the training process to obtain adversarially robust models, i.e., adversarial training (Shaham et al., 2015; Madry et al., 2018; Jalal et al., 2017; Balunovic and Vechev, 2020).

To introduce adversarial training, let l denote the loss function and $f_\theta(x) : \mathbb{R}^d \to \mathbb{R}$ be the model with parameter θ. The (population) adversarial loss is defined as $R_f(\theta, \epsilon) := \mathbb{E}\left[l\left(f_\theta[x + A_\epsilon(f_\theta, x, y)], y\right)\right]$, where A_ϵ is an attack of strength $\epsilon > 0$ and intends to deteriorate the loss in the following way

$$A_\epsilon(f_\theta, x, y) := \underset{z \in \mathcal{R}(0,\epsilon)}{\operatorname{argmax}} \{l(f_\theta(x + z), y)\}. \tag{1}$$

In the above, z is subject to the constraint $\mathcal{R}(0, \epsilon)$, i.e. an $\mathcal{L}_2$ ball centered at 0 with radius ϵ.

Given i.i.d. training samples $\{(x_i, y_i)\}_{i=1}^n$, the adversarial training aims to minimize an empirical version of $R_f(\theta, \epsilon)$ w.r.t. θ:

$$\widehat{R}_f(\theta, \epsilon) = \frac{1}{n} \sum_{i=1}^{n} l\left(f_\theta[x_i + A_\epsilon(f_\theta, x_i, y_i)], y_i\right), \tag{2}$$

and $\widehat{\theta} = \operatorname{argmin}_\theta \widehat{R}_f(\theta, \epsilon)$. The minimization in (2) is often implemented through an iterative two-step (min-max) update. In the t-th iteration, we first calculate the adversarial sample $\widetilde{x}_i^{(t)} = x_i + A_\epsilon(f_{\theta^{(t)}}, x_i, y_i)$ based on the current $\theta^{(t)}$, and then update $\theta^{(t+1)}$ based on the gradient of the adversarial training loss while fixing $\widetilde{x}_i^{(t)}$'s; see Algorithm 1. This generic algorithm and its variants have been studied in Shaham et al., 2015; Madry et al., 2018; Jalal et al., 2017; Balunovic and

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

Vechev, 2020; Sinha et al., 2018; Wang et al., 2019a among others. Note that for complex loss function l or model f_θ, there may not be an analytic form for A_ϵ (e.g. deep neural networks). In this case, an additional iterative optimization is needed to approximate A_ϵ at each step; see Wang et al., 2019a.

Algorithm 1 A General Form of Adversarial Training

Input: data (x_1, y_1),..., (x_n, y_n), attack strength ϵ, number of steps T, initialization $\theta^{(0)}$, step size η.
for $t = 1$ **to** T **do**
  for $i = 1$ **to** n **do**
    Calculate the attack for the ith sample and get $\widetilde{x}_i^{(t-1)} = x_i + A_\epsilon(f_{\theta^{(t-1)}}, x_i, y_i)$.
  end for
  Fixing $\widetilde{x}_i^{(t-1)}$'s, update $\theta^{(t)}$ from $\theta^{(t-1)}$ through

$$\begin{aligned} \theta^{(t)} &= \theta^{(t-1)} - \eta \nabla_\theta \widehat{R}_f(\theta^{(t-1)}, \epsilon) \qquad (3) \\ &= \theta^{(t-1)} - \eta \nabla_\theta \left[\frac{1}{n} \sum_{i=1}^{n} l\left(f_{\theta^{(t-1)}}(\widetilde{x}_i^{(t-1)}), y_i \right) \right]. \end{aligned}$$

end for
Output: $\theta^{(T)}$.

In the literature, there are three major strands of theoretical studies related to this work. The first strand focuses on the statistical properties or generalization performance of adversarially robust estimators without taking account of the role of optimization algorithms (Javanmard et al., 2020; Yin et al., 2019; Raghunathan et al., 2019; Schmidt et al., 2018; Najafi et al., 2019; Min et al., 2020; Zhai et al., 2019; Hendrycks et al., 2019; Chen et al., 2020). For instance, Javanmard et al. (2020) studied the statistical properties of $R_f(\widehat{\theta}, \epsilon)$, without specifying how to obtain the exact/approximate global minimizer $f_{\widehat{\theta}}$. The second strand studies the adversarial training loss, i.e., the limiting behaviors of $\widehat{R}_f(\theta^{(t)}, \epsilon)$ as a training algorithm iteration t grows. For instance, Gao et al. (2019); Zhang et al. (2020) showed that for over-parameterized neural networks, the empirical adversarial loss could be arbitrarily close to the minimum value in a local region near initialization. The third strand studies the (local) convergence of adversarial training under certain convexity assumptions (e.g., Sinha et al., 2018; Wang et al., 2019a). Besides these theoretical results, there are a few empirical works as well (e.g., Wong et al., 2020; Wang et al., 2019b; Rice et al., 2020; Lee and Chandrakasan, 2020; Wu et al., 2020; Xie et al., 2020).

In this paper, we investigate the global convergence and the generalization ability (i.e., $R_f(\theta^{(T)}, \epsilon)$) of the adversarial training, for two models f_θ, linear regression and two-layer neural networks, with the squared loss $l(f_\theta(x), y) = (f_\theta(x) - y)^2$ under $\mathcal{L}_2$ and $\mathcal{L}_\infty$ attacks. Our theoretical contributions are summarized as follows.

First, the adversarial loss $R_f(\theta, \epsilon)$ suffers from non-differentiability even if both l and f_θ are smooth, thus the optimization, i.e., Algorithm 1, works poorly. This motivates us to introduce a surrogate attack to overcome the non-smoothness problem. Under low dimensional setup and $\mathcal{L}_2$ attack, we show that under proper conditions, the iterative estimate $\theta_\xi^{(T)}$ trained from the surrogate adversarial loss asymptotically achieves the minimum adversarial risk (Section 2).

Secondly, we observe the "data interpolation" behavior of $\mathcal{L}_2$ adversarial training under high dimensional setup ($d/n \to \infty$). More specifically, the training loss converges to zero, but the population loss $R_f(\theta_\xi^{(T)}, \epsilon)$ converges to a large constant which is the adversarial loss of null model. To remedy the poor generalization, we penalize the adversarial training loss using LASSO. The resulting adversarially robust estimator and adversarial risk are both consistent for under some sparsity assumption (Section 3).

Thirdly, we examine the differences between the $\mathcal{L}_\infty$ and $\mathcal{L}_2$ adversarial training. One similarity with $\mathcal{L}_2$ case is that, when attack strength is small, data interpolation prevents $\mathcal{L}_\infty$ adversarial training from achieving a consistent estimator as well, and the issue can be improved by the use of LASSO penalty. In terms of the differences, in general, it is harder to conduct adversarial training under $\mathcal{L}_\infty$ attack, in the sense that a lower learning rate and more iterations are required to ensure the convergence of $\theta_\xi^{(T)}$ (Section 4).

It is worth mentioning that a recent work by Allen-Zhu and Li (2020) also conducted a similar analysis for the generalization ability of adversarial training for a two-layer ReLU network model. However, the focus of their work is to explain the feature purification effect of adversarial training in neural network and overlook the difficulties of adversarial training when ϵ does not asymptotic goes to 0.

For technical simplicity, throughout the paper, we assume the data are generated from linear regression:

$$y = \theta_0^\top x + \varepsilon, \qquad (4)$$

where $x \in \mathbb{R}^d$ is Gaussian vector with mean 0 and variance Σ (θ_0 is not perpendicular to Σ), and ε is a Gaussian noise (independent of x) with variance $\sigma^2 < \infty$. As d diverges, we assume both maximum and minimum eigenvalues of Σ are finite and bounded away from 0. In addition, $\|\theta_0\|$ and σ^2 are allowed to increase in d, but the signal-to-noise ratio $\|\theta_0\|_\Sigma/\sigma$ is large, say bounded away from zero.

2 LOW DIMENSIONAL ASYMPTOTICS

This section considers linear regression models and two-layer neural network models (training the first layer weights) under the low dimensional scenario. In particular, we examine the landscape of the adversarial loss and further investigate the testing performance of the estimator $\theta_\xi^{(T)}$. In what follows, we rewrite R_f as R_L for linear models and as R_N for two-layer networks.

2.1 Linear regression model

Consider the linear regression model: $f_\theta(x) = \theta^\top x$. By the definition of $A_\epsilon(f_\theta, x, y)$ under the $\mathcal{L}_2$ ball constraint and the fact that x and ε are both Gaussian, R_L has an analytical form as

$$R_L(\theta,\epsilon) = \|\theta-\theta_0\|_\Sigma^2 + \sigma^2 + \epsilon^2\|\theta\|^2 + 2\epsilon c_0\|\theta\|\sqrt{\|\theta-\theta_0\|_\Sigma^2+\sigma^2}, \tag{5}$$

where $\|a\|_\Sigma^2 := a^\top\Sigma a$, $\|a\| := \|a\|_2$ for any vector a, and $c_0 := \sqrt{2/\pi}$. Given any $\epsilon \geq 0$, define

$$\theta^*(\epsilon) := \operatorname*{argmin}_\theta R_L(\theta,\epsilon) \text{ and } R^*(\epsilon) := \min_\theta R_L(\theta,\epsilon).$$

When no confusion arises, we will rewrite $\theta^*(\epsilon)$ and $R^*(\epsilon)$ as θ^* and R^* for simplicity.

We first analyze $R_L(\theta,\epsilon)$ based on the form (5).

Proposition 1. *Assume the attack strength $\epsilon > 0$ and the data generation follows (4). The adversarial loss R_L is differentiable w.r.t. θ if and only if (1) $\sigma^2 > 0$ and $\theta \neq 0$, or (2) $\sigma = 0$, $\theta \neq 0$, and $\theta \neq \theta_0$. The function R_L is convex w.r.t. θ, and there exists some constant c such that when $\epsilon \geq c$, $\theta^* = 0$.*

From Proposition 1, there always exists some θ where $R_L(\theta,\epsilon)$ is not differentiable (e.g., $\theta = 0$), and it is not avoidable when ϵ is large. Even if we smooth the standard loss as in Xie et al. (2020) to improve the quality of the gradients (or use smooth classifiers as in Salman et al., 2019), the non-differentiable issues remains for the adversarial loss. This makes it difficult to track the trajectory of gradient descent algorithms in the sense that the adversarial training loss will fluctuate during training rather than strictly decreases over iterations.

To solve this problem and improve the gradient quality, for both models under consideration, we introduce $A_{\epsilon,\xi}$ that is a surrogate for $\mathcal{L}_2$ attack A_ϵ:

$$A_{\epsilon,\xi}(f_\theta,x,y) = \frac{\|\partial l(f_\theta(x),y)/\partial x\|}{\sqrt{\|\partial l(f_\theta(x),y)/\partial x\|^2+\xi^2}} A_\epsilon(f_\theta,x,y).$$

Lee and Chandrakasan (2020) showed that the instability of adversarial training when $\partial l(f_\theta(x),y)/\partial x$ is closed to zero. Therefore, the design of $A_{\epsilon,\xi}$ aims to impose shrinkage effect on A_ϵ when $\|\partial l(f_\theta(x),y)/\partial x\|$ is small, thus the training process will be more stable.

When $\xi = 0$, $A_{\epsilon,\xi}$ is reduced to A_ϵ. With the surrogate $A_{\epsilon,\xi}$, we define the empirical and population surrogate loss as:

$$\widehat{R}_{L,\xi}(\theta,\epsilon) := \frac{1}{n}\sum_{i=1}^n l(f_\theta(x_i + A_{\epsilon,\xi}(f_\theta,x_i,y_i)),y_i),$$
$$R_{L,\xi}(\theta,\epsilon) := \mathbb{E}l(f_\theta(x_i + A_{\epsilon,\xi}(\theta,x_i,y_i)),y_i).$$

One can show, $R_{L,\xi}(\theta,\epsilon)$ is smooth and convex everywhere. Accordingly, Algorithm 1 is modified by replacing A_ϵ with $A_{\epsilon,\xi}$. Note that a smaller value of ξ, which leads to a closer approximation to $R_{L,0}$, makes $R_{L,\xi}$ less smooth at the origin and thus requires a lower learning rate η and more iteration steps T. Therefore, we require a fine-tuning of ξ in the sense that $\xi/\|\theta_0\|$ slowly decreases to 0. The surrogate loss for two-layer neural networks, i.e., $R_{N,\xi}$ and $\widehat{R}_{N,\xi}$, is defined in the same way.

The use of surrogate attack improve the gradient quality of adversarial training and enforces the smoothness and convexity of surrogate loss $R_{L,\xi}(\theta,\epsilon)$. It facilitates the convergence and generalization analysis of adversarial training. We study the consistency of $\theta_\xi^{(T)}$ toward θ^* and $R_{L,\xi}(\theta_\xi^{(T)},\epsilon)$ toward R^*, under suitable choices of (ξ,η,t) after accounting for the dimensional effect $v^2 := \|\theta_0\|_\Sigma^2 + \sigma^2$ (for simplicity assume v^2 bounded away from zero); see Theorem 2.

Theorem 2. *Assume the data generation follows (4) and let ϵ be a fixed constant. If there exists some constant B_0 such that $\|\theta_\xi^{(0)}\| \leq B_0 v$, and the dimension growth rate satisfies $\log n\sqrt{d^2/n} \to 0$, then with probability tending to 1, the surrogate loss $R_{L,\xi}(\theta_\xi^{(t)},\epsilon)$ decreases in each iteration, and*

$$\frac{R_{L,\xi}(\theta_\xi^{(T)},\epsilon) - R^*}{v^2} \to 0, \text{ and } \frac{\|\theta_\xi^{(T)} - \theta^*\|}{v} \to 0,$$

given $\eta = \xi/(v^2 L)$ for some large constant L, $T = (v^2\log\log n)/\xi$ and $\xi = v^2 d/\sqrt{n}\log n$.

The proof of Theorem 2 is postponed to Appendix C. The results of Theorem 2 actually hold for a wide range of (ξ,η,T) (as elaborated in the Appendix C). In general, the choice of (η,T) can be invariant to v, while a larger v implies a wider range of possible ξ. In terms of tuning $\theta_\xi^{(0)}$ and ξ, one can estimate v^2 via the sample variance of $y_1,...,y_n$. Note that the above discussions apply to Theorems 3, 4, 5, 6, and 7 as well.

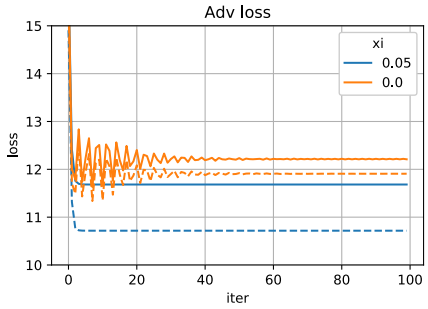

Figure 1: Adversarial training and testing loss in linear regression. Solid line: adversarial testing loss. Dashed line: adversarial training loss. The training process with $\xi = 0$ is more fluctuate.

Theorem 2 also confirms the phenomenon that "adversarial training hurts standard estimation" (Raghunathan et al., 2019). Denote $\widehat{\theta}_{\text{OLS}}$ as the common least square estimator for un-corrupted data (i.e. without attack) and trivially $\widehat{\theta}_{\text{OLS}} \to \theta_0$. By Theorem 2, $\theta_\xi^{(T)} \to \theta^* \neq \theta_0$, thus

$$\frac{R_L(\theta_\xi^{(T)}, 0) - R_L(\widehat{\theta}_{\text{OLS}}, 0)}{v^2} \to \frac{R_L(\theta^*, 0) - R_L(\theta_0, 0)}{v^2} = c(\epsilon) > 0,$$

where the function $c(\epsilon)$ increases in ϵ, and converges to $\|\theta_0\|_\Sigma^2/v^2$ as ϵ diverges.

Remark 1. *The Gaussian assumption of x is used in Theorem 2 for the convenience of showing the analytical form of R_L. All the results in this paper, except for Section 3.2, are still valid as long as the density of (x, y) is finite and $\exp(-t\|x\|^2) < \infty$ for some $t > 0$.*

Numerical experiments

To demonstrate the necessity of smoothing attacks under large ϵ, a simple simulation is conducted. Let $d = 10$, $n = 1000$, $x \in \mathcal{R}^d$ and $\Sigma = I_d$. The response $y = \beta_0^\top x + \varepsilon$ with $\beta_0 = (1, ..., 1)^\top$ and $\sigma^2 = 1$. The level of attack is taken as $\epsilon = 3$. We use the same training data set in the adversarial training for $\xi = 0$ and $\xi = 0.05$. Figure 1 displays the effectiveness of ξ: the loss with $\xi = 0$ always fluctuates, and after we smooth the training process through taking $\xi = 0.05$, (surrogate) training and (non-surrogate) testing loss smoothly decrease in each iteration.

2.2 Two-layer neural networks

We consider the two-layer neural network with an activation function ϕ, say

$$f_\theta(x) = \frac{1}{\sqrt{h}} \sum_{j=1}^{h} \phi(x^\top \theta_j) a_j, \tag{6}$$

where a_j's are known values and $\theta = (\theta_1, ..., \theta_h) \in \mathbb{R}^{d\times h}$ is the parameter to be trained. This "lazy training" setup has been widely used in the literature (e.g. Du et al., 2018, 2019; Arora et al., 2019; Ba et al., 2020; Allen-Zhu and Li, 2020), it eases the theoretical analysis. Additional, we adopt a vanishing initialization scheme (Ba et al., 2020): $\theta_{\xi,j}^{(0)} \sim N(0, I_d/dh^{1+\delta})$ for some $\delta > 0$.

Theorem 3 shows that $R_{N,\xi}(\theta_\xi^{(T)}, \epsilon)$ converges to the same minimal loss R^* as in linear regression under proper choice of T and ξ, as n and h diverge.

Theorem 3. *Under the generative model (4), assume the activation function ϕ in model (6) is twice continuously differentiable, $\phi'(0) \neq 0$, and $\phi(0) = 0$. If a and h satisfy $\|a\|_\infty = O(1)$, $\max|a_j|/(\min|a_j|) = \Theta(1)$, $(d\log n)\|a\|_\infty v/\sqrt{h} \to 0$, and $(d\log n)\|a\|_\infty v\sqrt{h}/\|a\|^2 \to 0$. When $\xi/v^2 \to 0$ where $\xi/v^2 = -\log\log n/\log(\sqrt{d^2\log n/n} \vee (d\log n)\|a\|_\infty/\sqrt{h})$, and $\eta = \xi h/(v^2 L\|a\|^2)$ for some large constant L, with probability tending to 1, for $T = (v^2\log\log n)/\xi$, if $\sqrt{d\log n}(1+v^2\eta\|a\|^3/(h^{3/2}\xi) + L\eta v)^T/h^{\delta/2} \to 0$, then*

$$\frac{R_{N,\xi}(\theta_\xi^{(T)}, \epsilon) - R^*}{v^2} \to 0, \tag{7}$$

where R^* is the exactly the same as Theorem 2.

The detailed proof is postponed to Appendix D. The choice of ξ here depends on the weights a together with the number of hidden nodes h. Note that although both Theorems 2 and 3 establish the convergence of $R_{N,\xi}(\theta, \epsilon)$, Theorems 3 requires that ξ/v converges to zero in a slower speed, leading to a slower convergence rate for $R_{N,\xi}(\theta, \epsilon)$.

Remark 2. *The proof of Theorem 3 is similar to Ba et al. (2020): as the number of hidden nodes h grows, the trajectories of optimization using linear network (with zero initialization) and nonlinear network (with vanishing initialization) are slightly different, while the convergence result of the former one can be simply extended from linear models. Different from Ba et al. (2020), we specify the learning rate as well as the number of iterations as functions of (d, a, n, h), while Ba et al. (2020) utilized gradient flow, which is not applicable in our setup. In addition, compared with Ba et al. (2020), the relationship of (ξ, a, η) is revealed in our result when $\|\theta_0\| \to \infty$.*

Theorem 3 requires a continuous differentiable ϕ, and similar results can be established forReLU activation function as well:

Theorem 4. *Under the generative model (4), assume the activation function ϕ is ReLU function with zero initialization and no bias. Take $\xi/v^2 =$*

$-\log\log n/\log(\sqrt{(d^2\log n)/n})$. *Set* $\eta = \xi h/(v^2 L\|a\|^2)$ *for some constant* L. *Denote* a^+ *as a vector such that* $a_j^+ = a_j 1\{a_j > 0\}$, *and similarly define* a^-. *If a satisfy* $\|a^+\|/\|a^-\| = 1$ *and* $\|a\|_\infty = O(1)$, *with probability tending to 1, for* $T = (v^2\log\log n)/\xi$, *(7) holds.*

Numerical experiments

A series of simulation studies of low dimensional linear regression and two-layer neural network model with lazy training are conducted. Due to page limit, these results are detailed in Appendix A and successfully validate our Theorems 2-4 on the convergence of adversarial training with surrogate loss.

Here we present another experiment that shows the improvement of predicting the performance of adversarial trained estimator via surrogate loss for complicated models beyond our theorems. We fit a ResNet-34 (WideResNet34-1) model for CIFAR-10 dataset. Since the ReLU function is not smooth, we utilize the training technique introduced by Xie et al. (2020): we use ReLU in the forward path and use Softplus($\beta = 10$) in the backward path to improve the gradient quality. The number of epochs is taken as 100. The initial learning rate is 0.1, and at the 75th and 90th epoch, it is multiplied by 0.1. The value of ξ is taken as 0.001 at the initial stage and multiplies 0.1 whenever the learning rate is changed. We repeat this experiment 10 times to obtain the mean and variance and conduct this experiment under various levels of attacks. We use $\mathcal{L}_2$ attack with strength $\epsilon = 3.0$ in this experiment. The results are summarized in Figure 2. It shows that using surrogate loss leads to slightly higher adversarial testing accuracy and much higher standard testing accuracy than the one with $\xi = 0$.

In addition, we conduct two experiments (in case that the adversarial training with $\xi = 0$ does not converges algorithmically in the above experiment): (1) with 200 epochs and (2) with initialization that is obtained by standard training as in Allen-Zhu and Li (2020). The results are postponed to the appendix. In short, a better performance is obtained under surrogate loss, and the initialization from standard training does not improve the performance.

Although our theory only reveals a single non-differentiable point, it is still important to handle this carefully in neural networks. The non-differentiable problem is partially due to that the adversary has no preference in the direction of attack. If we estimate the attack twice (from different initializations) and the difference between the two estimates will be large when the non-differentiable problem is severe. We conduct a small experiment to investigate the attack difference. The details are postponed to the appendix. The results justify the importance of accommodating this non-differentiability issue, especially for large ϵ.

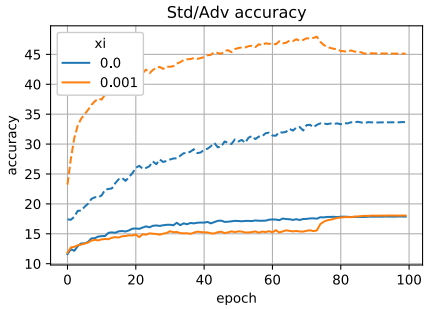

Figure 2: Standard and adversarial testing accuracies for CIFAR-10 with ResNet34. Solid line: adversarial testing accuracy. Dashed line: standard testing accuracy. With $\xi = 0.001$, both adversarial testing accuracy and standard testing accuracy (with standard deviation as 0.458, 0.321) are higher than that for $\xi = 0$ (0.743, 0.219).

3 HIGH DIMENSIONAL ASYMPTOTICS

In this section, we focus on the high dimensional regime where $d/n \to \infty$. It is first revealed that the adversarial training also suffers from the classical interpolation effect, i.e., near-zero (surrogate) adversarial training loss but high generalization error. As a potential remedy, we penalize the adversarial training loss using LASSO and show that the estimate is consistent when θ^* is sparse.

3.1 Effect of interpolation

It is well known that interpolation may occur under high dimensionality. For instance of linear regression, if a gradient descent with zero initialization is applied to minimize the squared loss when $d/n \to \infty$, then the solution converges to

$$\theta(\mathbf{y}) := \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{y},$$

where $\mathbf{X} = (x_1, x_2, ..., x_n)^\top$ and $\mathbf{y} = (y_1, y_2, ..., y_n)^\top$, given a sufficiently small learning rate. This perfectly interpolated estimator $\theta(\mathbf{y})$ is proven to be inconsistent to θ_0 and lead to a large generalization error (e.g., Hastie et al., 2019; Belkin et al., 2019). Note that this over-fitting scenario is different from the one in Rice et al. (2020), which is caused by over-parameterization in deep neural networks rather than high dimensionality of the input.

Our first result shows that $\theta(\mathbf{y})$ also induces the same effect of interpolation in adversarial learning for linear models.

Lemma 1. *Assume data generation follows (4). When $\theta(\boldsymbol{y}) \neq 0$ and $d/n \to \infty$, we have $\|\theta(\boldsymbol{y})\|^2/v^2 = O_p(n/d)$, and with probability tending to 1, for any $\xi \geq 0$, it holds that*

$$\frac{\widehat{R}_{L,\xi}(\theta(\boldsymbol{y}),\epsilon)}{v^2} \to 0 \text{ and } \frac{R_{L,\xi}(\theta(\boldsymbol{y}),\epsilon)}{v^2} \to 1.$$

We next show that $\theta_\xi^{(T)}$ shares the same properties as $\theta(\mathbf{y})$. The core idea is that the training trajectory $\{\theta_\xi^{(t)}\}_{t=1}^T$ can be sufficiently close to that in the standard training, when both are initialized from zero. Since the latter converges to $\theta(\mathbf{y})$, the surrogate adversarial training loss and testing loss of $\theta_\xi^{(T)}$ act in a similar way as those for $\theta(\mathbf{y})$ respectively.

Theorem 5. *Under the same assumptions as in Lemma 1, when $(\log n)\sqrt{n/d} \to 0$, take η small enough such that the largest eigenvalue of $\eta \boldsymbol{X}^\top \boldsymbol{X}$ is smaller than 1. Use zero initialization, and denote $T := \min\{t \in \mathbb{Z}^+ : |\|\boldsymbol{X}\theta_\xi^{(t)} - \boldsymbol{y}\|_2/(v\sqrt{n}) < 1/\sqrt{\log n}\}$, then with probability tending to 1, for any $\xi > 0$, we have $T < \infty$ and*

$$\frac{\widehat{R}_{L,\xi}(\theta_\xi^{(T)},\epsilon)}{v^2} \to 0, \text{ and } \frac{R_{L,\xi}(\theta_\xi^{(T)},\epsilon)}{v^2} \to 1.$$

The proof of Theorem 5 is postponed to Appendix E. Compared with Theorem 2, Theorem 5 no longer requires ξ to be associated with (d, n). A crucial reason for this difference is that under high dimensionality, when $t \leq T$, the smoothness of $\widehat{R}_{L,\xi}$ along the training trajectory (i.e., the gradient of $\widehat{R}_{L,\xi}(\theta_\xi^{(t)},\epsilon)$) is always dominated by a term that is only determined by the eigenvalues of high dimensional design matrix $\mathbf{X}$, regardless of how small ξ is (refer to equation (13) in Appendix E for details). This is contrast to the low dimensional case.

Theorem 5 shows that $R_{L,\xi}(\theta_\xi^{(T)},\epsilon)/v^2$ does not converge to R^*/v^2. Similar results can be established for two-layer neural networks (with lazy training):

Theorem 6. *For the two-layer neural network (6), under the same conditions on ϕ as in Theorem 3, $(\log n)\sqrt{d/n} \to \infty$, take zero/vanishing initialization and $\eta = \eta_{linear} h/\|a\|^2$. Assume $\|a\|_\infty = O(1)$, $(d \log n)\|a\|_\infty v/\sqrt{h} \to 0$, $\max|a_j|/(\min|a_j|) = \Theta(1)$, $\sqrt{dn}(\log n)^2\|a\|_\infty v\sqrt{h}/\|a\|^2 \to 0$, $\sqrt{d \log n}(1 + v^2\eta\|a\|^3/(h^{3/2}\xi) + L\eta v)^T/h^{\delta/2} \to 0$. Denote $T := \min\{t \in \mathbb{Z}^+ : |\|f_{\theta_\xi^{(t)}} - \boldsymbol{y}\|_2/(v\sqrt{n}) < 1/\sqrt{\log n}\}$, then with probability tending to 1, for any $\xi > 0$, we have $T < \infty$ and*

$$\frac{\widehat{R}_{N,\xi}(\theta_\xi^{(T)},\epsilon)}{v^2} \to 0, \text{ and } \frac{R_{N,\xi}(\theta_\xi^{(T)},\epsilon)}{v^2} \to 1. \quad (8)$$

For ReLU network, we have the following result:

Theorem 7. *For the two-layer neural network (6), $(\log n)\sqrt{d/n} \to \infty$, assume the activation function ϕ is ReLU function with zero initialization and no bias. If $\|a^+\|/\|a^-\| = 1$, $\|a\|_\infty = O(1)$. Denote $T := \min\{t \in \mathbb{Z}^+ : |\|f_{\theta_\xi^{(t)}} - \boldsymbol{y}\|_2/(v\sqrt{n}) < 1/\sqrt{\log n}\}$, then for any $\xi > 0$, with probability tending to 1, we have $T < \infty$, and (8) holds.*

Numerical experiment

A simulation is conducted to verify Theorem 5. We choose $n = 20$, $d = 1000$, and $\sigma^2 = 1$. The true underlying model θ_0 is all-zero except for its first 10 elements being 1. The attack intensity is $\epsilon = 0, 0.01, 0.1$. Learning rate is taken as 0.001 with zero initialization and $\xi = 0.5$. The curves in Figure 3 represent means of respective statistics, and the shaded areas represent mean $\pm$ one standard deviation, based on 100 replications. Figure 3 shows that the surrogate adversarial training loss keeps decreasing to around zero for all the choices of ϵ, while the adversarial testing loss converges to some nonzero constant. Note that the three adversarial training loss curves in the left plot of Figure 3 overlap.

More experiments for larger d and $\xi = 0$ (with a change when $\theta = 0$) are postponed to Appendix A. Besides, we also postpone experiments for neural networks to Appendix A.

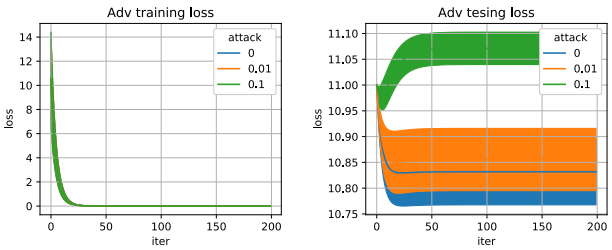

Figure 3: Adversarial training, high-dimension.

3.2 Improving adversarial robustness using LASSO under high dimensionality

In this section, we explore how incorporating LASSO improves adversarial learning under high dimensionality. In particular, we will present some theoretical justifications for linear models and conduct numerical exploration to evaluate the potential of LASSO in neural networks.

Intuitively, a sparse adversarial learning via LASSO makes sense only when the adversarial loss has a sparse global optimization, i.e., θ^* is a sparse vector. Therefore, certain investigation is necessary to understand the sparsity relation between θ_0 and θ^*.

Proposition 8. *Under model (4), the optimal solution of $R_{L,\xi}(\theta,\epsilon)$ is of the form $\theta^* = (\Sigma+\kappa I)^{-1}\Sigma\theta_0$ for some κ as a function of $(\theta_0,\Sigma,\sigma^2,\xi)$. Assuming θ_0 is sparse, then whether the robust coefficient $((\Sigma+\kappa I)^{-1}\Sigma\theta_0)$ is sparse or not depends on Σ.*

The following example illustrates that, based on Proposition 8, when the correlation between active set and inactive set is zero, the adversarially robust model θ^* is sparse as well.

Example 1. *When θ_0 is sparse, and Σ can be represented as $\begin{Bmatrix}\Sigma_1 & 0\\ 0 & \Sigma_2\end{Bmatrix}$, where Σ_1 is the covaraince of the active attributes, and Σ_2 is for the other attributes of x, the model θ^* will be sparse as well.*

To simplify the derivation, we assume $\Sigma = I$ in the following result. Denote S as the active set of θ^*, and $s = |S|$ as the size of S. We consider applying LASSO in the adversarial training loss:

$$\frac{1}{n}\sum_{i=1}^{n} l(f_\theta(x_i + A_{\epsilon,\xi}(f_\theta, x_i, y_i)), y_i) + \lambda\|\theta\|_1.$$

The statistical property of $\widehat{\theta}_\xi$, the minimizer of the above objective function, is as follows:

Theorem 9. *Assume data generation follows (4), θ_0 is sparse and $\Sigma = I$. Take $\xi/v^2 \to 0$, $\lambda/v = o(1)$ and $\lambda/v \geq (c\sqrt{(s\log d)/n}) \vee (\xi a_n/v^2)$ for some large constant c and $a_n \to \infty$. If $\epsilon < \sqrt{\pi}\|\theta_0\|_2/(\sqrt{2}v)$, then $\theta^* \neq 0$, and with probability tending to 1, we have*

$$\frac{R_{L,\xi}(\widehat{\theta}_\xi,\epsilon) - R^*}{v^2} \to 0, \text{ and } \frac{\|\widehat{\theta}_\xi - \theta^*\|_1}{\lambda} < \infty.$$

The proof of Theorem 9 is similar to the traditional LASSO analysis as in Bickel et al. (2009); Belloni and Chernozhukov (2013) but with an important modification. In the literature, the Hessian of the standard training loss, i.e., $\mathbf{X}^\top\mathbf{X}/n$, is usually required to satisfy the so-called restricted eigenvalue condition. However, in adversarial setting, the Hessian changes as θ, so it takes more steps to verify the above condition.

Remark 3. *Theorem 9 shows the effectiveness of LASSO in sparse linear model and it performs better then the case without LASSO. Note that the estimation consistency still holds for low-dimensional dense model, if d and n satisfies $(d\log d)/n \to 0$. But, to ensure that LASSO improves the performance in this case, λ should be carefully tuned.*

We conduct some empirical study to explore the potential applications of LASSO in the adversarial training of neural networks. Similar experiments under large-sample regime can be found for adversarial training Sinha et al. (2018); Wang et al. (2019a); Raghunathan et al. (2019), and pruning in adversarial training Ye et al. (2019); Li et al. (2020).

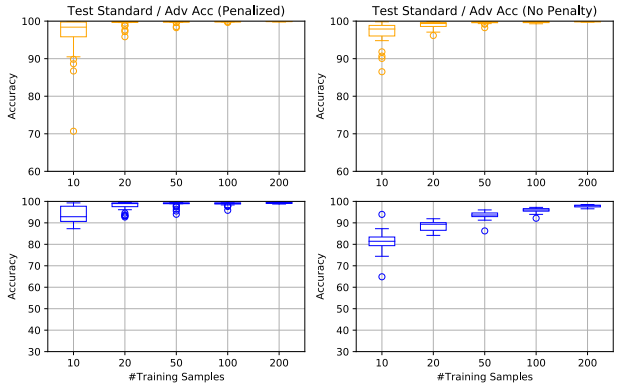

Figure 4: Comparison on standard (upper) / adversarial (lower) test accuracies between training with/without $\mathcal{L}_1$ penalty under $\mathcal{L}_2$ attack with $\xi = 10^{-4}$. Attack strength $\epsilon = 3$.

Numerical experiments

The program was modified from a repository in Github[1] and library `Advertorch`. A simple two-layer neural network is constructed with 1024 hidden nodes and ReLU as activation. We use MNIST dataset to distinguish between digits 0 and 1, and randomly select a small number of samples of 0 and 1 from the training dataset to create a high-dimension scenario. The $\mathcal{L}_2$ attack level is set to be 3. We trained 2000 epochs to ensure the convergence of the algorithms and repeat the experiment for 30 times to draw a boxplot. After training 2000 epochs, for both $\lambda = 0$ (No penalty) and $\lambda = 0.001$ (LASSO), the training accuracies for clean data and adversarial data both reach 100%. The penalty $\lambda = 0.001$ was chosen such that the magnitude of penalty is comparable with loss. The results are summarized in Figure 4.

For adversarial accuracy, as shown in Figure 4, the results with two different λ's are significantly different, where the choice of $\lambda = 0.001$ improves the adversarial accuracy compared with $\lambda = 0$. As a reference, we also plot the standard accuracy (i.e. prediction accuracy for un-corrupted data), even though the objective function minimized is the (penalized) adversarial training loss. Figure 4 shows, it approaches 99% quickly in n for both adversarial training with and without LASSO.

We also observe similar results when $\xi = 0$, and the details are postponed to Appendix A.

We also tried on CIFAR-10 with WideResNet34-10.

[1] `https://github.com/louis2889184/pytorch-adversarial-training`

The λ is chosen to ensure that the magnitude of cross entropy loss and the LASSO penalty are comparable. We use all data in the training dataset to conduct this experiment. The results are summarized in Table 1. Our theorem only concerns the high dimensional case, i.e., small-n-large-d, however, as showed in Table 1, both standard and adversarial testing accuracies are still enhanced when using LASSO, in this large-n application (Refer to Remark 3).

Method	std acc (%)	adv acc (%)
Benchmark	84.346(0.355)	61.760(0.204)
LASSO 10^{-5}	86.568(0.214)	63.072(0.292)

Table 1: Adversarial training in CIFAR-10 using WideResNet34-10 with $\mathcal{L}_2$ attack, $\epsilon = 0.5$.

4 $\mathcal{L}_\infty$ ADVERSARIAL TRAINING

In this section, we discuss the adversarial loss and adversarial training under $\mathcal{L}_\infty$ attack. Similar as stated in Chen et al. (2020), the adversarial risk of the linear model becomes

$$R_L^\infty(\theta, \epsilon) = \|\theta - \theta_0\|_\Sigma^2 + \sigma^2 + \epsilon^2\|\theta\|_1^2 + 2\epsilon c_0 \|\theta\|_1 \sqrt{\|\theta - \theta_0\|_\Sigma^2 + \sigma^2}. \tag{9}$$

Below are discussions w.r.t $\mathcal{L}_\infty$ adversarial training:

Harder to train
From (9), R_L^∞ is not differentiable when some element in θ is zero. Similar as for $\mathcal{L}_2$ attack, we propose to shrink the size of adversarial attack when R_L is not differentiable, while a difference is that the shrinkage is applied on each dimension of x: for $i = 1, ..., d$,

$$[A_{\epsilon,\xi}^\infty(f_\theta, x, y)]_i = \frac{|[\partial l/\partial x]_i|}{|[\partial l/\partial x]_i| + \xi} [A_\epsilon^\infty(f_\theta, x, y)]_i.$$

A major difference between $\mathcal{L}_\infty$ and $\mathcal{L}_2$ attacks is that, $\mathcal{L}_\infty$ attack is more sensitive to ξ. For example, if $\theta = (1/d, ..., 1/d)^\top$ and $\epsilon = 1/\sqrt{d}$, then A_ϵ^∞ becomes $(\epsilon, ..., \epsilon)^\top$ whose $\mathcal{L}_2$ norm is 1, while $A_{\epsilon,\xi}^\infty$ is $(\epsilon/(1 + d\xi), ..., \epsilon/(1 + d\xi))^\top$, whose $\mathcal{L}_2$ norm quickly shrinks to zero if $\xi d \to \infty$. As a result, it is necessary to require that $\xi = o(1/d)$ to avoid overshrinkage of the $\mathcal{L}_\infty$ attack. However, as discussed in previous sections, a smaller ξ requires smaller learning rate and more training iterations, thus training under $\mathcal{L}_\infty$ attack is more difficult.

Effect of interpolation
In high-dimensional case, adversarial training still suffers from data interpolation: when $\epsilon = O(1/\sqrt{d})$ and $d/n \to \infty$, the minimal adversarial training loss converges to zero, while the population adversarial loss converges to v^2 (recall that $v^2 = \|\theta_0\|^2 + \sigma^2$). Similar as for $\mathcal{L}_2$ attack, we add LASSO in the $\mathcal{L}_\infty$ adversarial training in MNIST and CIFAR-10. The results are summarized in Figure 15 and 16 in appendix, as well as Table 2. For both datasets, LASSO improves both standard and adversarial testing accuracies.

Method	std acc (%)	adv acc (%)
Benchmark	82.870(0.131)	50.338(0.315)
LASSO 10^{-5}	84.800(0.282)	54.260(0.376)

Table 2: Adversarial training in CIFAR-10 using WideResNet34-10 with $\mathcal{L}_\infty$ attack, $\epsilon = 8/255$.

Remark 4. *From the aspect of formulation, $\mathcal{L}_\infty$ adversarial loss and LASSO has overlapped effect as both introduce $\mathcal{L}_1$ penalty effect into the loss function, see (9). However, LASSO and $\mathcal{L}_\infty$ are designed for different purposes. From the aspect of loss landscape in deep learning, LASSO does not intend to change the loss landscape near the global minima as the penalty term goes to zero asymptotically, i.e., any global optimum for standard loss are optimum for LASSO problem given infinite training data. On the other hand, for adversarial robustness under $\mathcal{L}_\infty$ attack, it aims to select the certain global minima such that the prediction is robust in the nearby region of training samples, and not all minimizers of standard loss are robust to adversarial attack. From this aspect, they can be applied simultaneously. We refer readers to Guo et al. (2020) for more discussion.*

5 CONCLUSION AND FUTURE WORKS

This paper studies the convergence properties of adversarial training in linear models and two-layer neural networks (with lazy training). In the low-dimensional regime, using adversarial training with surrogate attack, the adversarial risk of the trained model converges to the minimal value. In a high-dimensional regime, data interpolation causes the adversarial training loss close enough to zero, while the generalization is poor. One potential solution is to add $\mathcal{L}_1$ penalty in the adversarial training, which results in both consistent adversarial estimate and risk in high dimensional sparse models.

There are several future directions. First, we may focus on classification tasks as a future work. In regression, the adversarially robust model generally outputs smaller-in-magnitude predictions, which is not practical in classification. One may be interested in how adversarial training works in classification. Second, the scenarios we consider are $d/n \to 0$ and ∞, and one can consider the linear dimensionality case, i.e.

$d/n \to c$, as a future direction. Finally, the non-smoothness issue happens to the adversarial loss and the penalty term (e.g., LASSO, Wang et al., 2019b; Wu et al., 2020), so there is potential to improve further the gradient quality of penalized adversarial training via smoothing the penalty function.

Acknowledgements

Dr. Song's research activities are partially supported by National Science Foundation DMS-1811812.

References

Allen-Zhu, Z. and Li, Y. (2020), "Feature Purification: How Adversarial Training Performs Robust Deep Learning," *arXiv preprint arXiv:2005.10190*.

Arora, S., Du, S. S., Hu, W., Li, Z., and Wang, R. (2019), "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, vol. 97 of *Proceedings of Machine Learning Research*, pp. 322–332.

Ba, J., Erdogdu, M., Suzuki, T., Wu, D., and Zhang, T. (2020), "Generalization of two-layer neural networks: an asymptotic viewpoint," in *8th International Conference on Learning Representations*.

Bai, Z.-D. and Yin, Y.-Q. (2008), "Limit of the smallest eigenvalue of a large dimensional sample covariance matrix," in *Advances In Statistics*, World Scientific, pp. 108–127.

Balunovic, M. and Vechev, M. (2020), "Adversarial training and provable defenses: bridging the gap," in *8th International Conference on Learning Representations*.

Belkin, M., Hsu, D., and Xu, J. (2019), "Two models of double descent for weak features," *arXiv preprint arXiv:1903.07571*.

Belloni, A. and Chernozhukov, V. (2013), "Least squares after model selection in high-dimensional sparse models," *Bernoulli*, 19, 521–547.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. (2009), "Simultaneous analysis of Lasso and Dantzig selector," *The Annals of Statistics*, 37, 1705–1732.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013), "Evasion attacks against machine learning at test time," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, pp. 387–402.

Chen, L., Min, Y., Zhang, M., and Karbasi, A. (2020), "More data can expand the generalization gap between adversarially robust and standard models," *arXiv preprint arXiv:2002.04725*.

Du, S. S., Lee, J. D., Li, H., Wang, L., and Zhai, X. (2019), "Gradient descent finds global minima of deep neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, vol. 97 of *Proceedings of Machine Learning Research*, pp. 1675–1685.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. (2018), "Gradient descent provably optimizes over-parameterized neural networks," *arXiv preprint arXiv:1810.02054*.

Gao, R., Cai, T., Li, H., Hsieh, C., Wang, L., and Lee, J. D. (2019), "Convergence of adversarial training in overparametrized neural networks," in *Advances in Neural Information Processing Systems*, pp. 13009–13020.

Guo, Y., Chen, L., Chen, Y., and Zhang, C. (2020), "On connections between regularizations for improving dnn robustness," *IEEE transactions on pattern analysis and machine intelligence*.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. (2019), "Surprises in high-dimensional ridgeless least squares interpolation," *arXiv preprint arXiv:1903.08560*.

Hendrycks, D., Lee, K., and Mazeika, M. (2019), "Using pre-training can improve model robustness and uncertainty," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97 of *Proceedings of Machine Learning Research*, pp. 2712–2721.

Ing, C.-K. and Lai, T. L. (2011), "A stepwise regression method and consistent model selection for high-dimensional sparse linear models," *Statistica Sinica*, 21, 1473–1513.

Jalal, A., Ilyas, A., Daskalakis, C., and Dimakis, A. G. (2017), "The robust manifold defense: Adversarial training using generative models," .

Javanmard, A., Soltanolkotabi, M., and Hassani, H. (2020), "Precise tradeoffs in adversarial training for linear regression," *arXiv preprint arXiv:2002.10477*.

Lee, K. and Chandrakasan, A. P. (2020), "Rethinking Empirical Evaluation of Adversarial Robustness Using First-Order Attack Methods," *arXiv preprint arXiv:2006.01304*.

Li, B., Wang, S., Jia, Y., Lu, Y., Zhong, Z., Carin, L., and Jana, S. (2020), "Towards Practical Lottery Ticket Hypothesis for Adversarial Training," *arXiv preprint arXiv:2003.05733*.

Ma, S. and Liu, Y. (2019), "Nic: Detecting adversarial samples with neural network invariant checking," in *Proceedings of the 26th Network and Distributed System Security Symposium*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018), "Towards deep learning models resistant to adversarial attacks," in *6th International Conference on Learning Representations*.

Min, Y., Chen, L., and Karbasi, A. (2020), "The curious case of adversarially robust models: More data can help, double descend, or hurt generalization," *arXiv preprint arXiv:2002.11080*.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016), "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582.

Najafi, A., Maeda, S.-i., Koyama, M., and Miyato, T. (2019), "Robustness to adversarial perturbations in learning from incomplete data," in *Advances in Neural Information Processing Systems*, pp. 5542–5552.

Papernot, N., McDaniel, P., Swami, A., and Harang, R. (2016a), "Crafting adversarial input sequences for recurrent neural networks," in *Military Communications Conference, MILCOM 2016-2016 IEEE*, IEEE, pp. 49–54.

Papernot, N., McDaniel, P. D., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016b), "The limitations of deep learning in adversarial settings," in *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, IEEE, pp. 372–387.

Raghunathan, A., Xie, S. M., Yang, F., Duchi, J. C., and Liang, P. (2019), "Adversarial training can hurt generalization," *arXiv preprint arXiv:1906.06032*.

Rice, L., Wong, E., and Kolter, J. Z. (2020), "Overfitting in adversarially robust deep learning," *arXiv preprint arXiv:2002.11569*.

Salman, H., Li, J., Razenshteyn, I., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. (2019), "Provably robust deep learning via adversarially trained smoothed classifiers," in *Advances in Neural Information Processing Systems*, pp. 11292–11303.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. (2018), "Adversarially robust generalization requires more data," in *Advances in Neural Information Processing Systems*, pp. 5014–5026.

Shaham, U., Yamada, Y., and Negahban, S. (2015), "Understanding adversarial training: Increasing local stability of neural nets through robust optimization," *arXiv preprint arXiv:1511.05432*.

Sinha, A., Namkoong, H., and Duchi, J. C. (2018), "Certifiable distributional robustness with principled adversarial training," in *6th International Conference on Learning Representations*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2014), "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations*.

Tao, G., Ma, S., Liu, Y., and Zhang, X. (2018), "Attacks meet interpretability: Attribute-steered detection of adversarial samples," in *Advances in Neural Information Processing Systems*, pp. 7717–7728.

Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., and Gu, Q. (2019a), "On the convergence and robustness of adversarial training," in *International Conference on Machine Learning*, pp. 6586–6595.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. (2019b), "Improving adversarial robustness requires revisiting misclassified examples," in *International Conference on Learning Representations*.

Wong, E., Rice, L., and Kolter, J. Z. (2020), "Fast is better than free: Revisiting adversarial training," *arXiv preprint arXiv:2001.03994*.

Wu, D., Wang, Y., and Xia, S.-t. (2020), "Revisiting Loss Landscape for Adversarial Robustness," *arXiv preprint arXiv:2004.05884*.

Xie, C., Tan, M., Gong, B., Yuille, A., and Le, Q. V. (2020), "Smooth adversarial training," *arXiv preprint arXiv:2006.14536*.

Ye, S., Xu, K., Liu, S., Cheng, H., Lambrechts, J.-H., Zhang, H., Zhou, A., Ma, K., Wang, Y., and Lin, X. (2019), "Adversarial robustness vs. model compression, or both," in *The IEEE International Conference on Computer Vision (ICCV)*, vol. 2.

Yin, D., Ramchandran, K., and Bartlett, P. L. (2019), "Rademacher complexity for adversarially robust generalization," 97, 7085–7094.

Zhai, R., Cai, T., He, D., Dan, C., He, K., Hopcroft, J., and Wang, L. (2019), "Adversarially robust generalization just requires more unlabeled data," *arXiv preprint arXiv:1906.00555*.

Zhang, Y., Plevrakis, O., Du, S. S., Li, X., Song, Z., and Arora, S. (2020), "Over-parameterized Adversarial Training: An Analysis Overcoming the Curse of Dimensionality," *arXiv preprint arXiv:2002.06668*.