Does Adversarial Training Hurt or Help Generalization?

Abstract

While adversarial training can improve robust accuracy (against an adversary), it has been observed to hurt standard accuracy (when there is actually no adversary). Previous work has studied this tradeoff between standard and robust accuracy in the infinite data setting, where no classifier performs well on both objectives. In this paper, we construct a problem where a tradeoff manifests in the finite data setting but vanishes in the infinite data setting. Furthermore, the fact that this problem is convex reveals a rather fundamental tension between robustness and generalization.

1. Introduction

It is well known that neural networks trained using standard training have very low accuracy against adversarial examples, where inputs are imperceptibly perturbed (Szegedy et al., 2014). Adversarial training (Goodfellow et al., 2015; Madry et al., 2018) can be effective at improving the accuracy against such examples (*robust accuracy*), but they have been found to hurt accuracy when no adversarial is actually present (*standard accuracy*) (Madry et al., 2018; Zhang et al., 2019). For example, Table 1 shows that on CIFAR-10 , while adversarial training improves robust accuracy from 3.5% to 45.8%, the standard accuracy drops from 95.2% to 87.3%.

This tradeoff has been shown to exist even in simple problems (Tsipras et al., 2019; Zhang et al., 2019), but in all these constructed problems, this tradeoff exists in the *infinite data limit*, where there simply does not exist a classifier that can simultaneously achieve both high standard accuracy and high robust accuracy.

In this paper, we ask the following question: if there is no tradeoff in the infinite data limit, then should adversarial training always help? After all, adversarial training can be viewed as adding additional training data or performing regularization in a way that respects the Bayes optimal classifier.

However, notice that in Table 1, both standard training and adversarial training achieve 100% standard accuracy on the *training set*. The fact that they achieve different test accuracies points at a generalization issue, specifically, a difference in convergence rates to the same optimum. Schmidt et al. (2018) explored generalization in a related but different context: they show that achieving good robust accuracy requires more samples than achieving good standard accuracy, whereas we are interested solely in standard accuracy, but comparing standard and adversarial training (different estimators).

	Standard training	**Adversarial training**
Robust test	3.5%	45.8%
Standard test	95.2%	87.3%
Standard train	100%	100%

Table 1. Train and test accuracies for Wide ResNet 28-10 trained on CIFAR-10 (Krizhevsky, 2009) dataset. Both training algorithms have same (both standard and robust) training accuracy but very different test accuracies. In particular, adversarial training causes worse generalization.

Does adversarial training require more samples because of the different (robust) objective or the difficulty of optimization? In this paper, we show by construction that even a *convex problem* based on regression using splines also exhibits the same phenomeon. Convexity allows us to take optimization concerns out of the picture, and lets us study the accuracy-robustness tradeoff in a clearer light. Our construction is based on the intuition that fitting the standard objective requires fitting global structure, which can be found with a low norm model, but that fitting the adversarial objective requires capturing local structure, which requires high norm, and thus requires more samples to fit accurately.

2. Setup

Prediction task. Consider the prediction task of mapping an $x \in \mathcal{X}$ to output $y \in \mathcal{Y}$. Given training data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \sim p^\star(x, y)$, the goal is to learn a prediction function $f \in \mathcal{F}$. Let $\ell(f,x,y)$ be some non-negative loss function. For a function f, we define the test loss $L_{\text{test}}(f)$ as follows.

$$L_{\text{test}}(f) \stackrel{\text{def}}{=} \mathbb{E}_{(x,y)\sim p^\star}[\ell(f,x,y)]. \tag{1}$$

Perturbations. For an input $x \in \mathcal{X}$, let $T(x)$ denote the set of perturbations of x that we wish to be robust to. For example, for the well-studied ℓ_∞ "adversarial examples", the perturbation set $T(x) = \{\tilde{x} : \|\tilde{x} - x\|_\infty \le \epsilon\}$.

Standard and robust estimators. In this work, we study two particular estimators: (i) the standard estimator that minimizes the (penalized) empirical loss on the training points

and (ii) the robust (adversarial) estimator that minimizes the (penalized) empirical loss on the training points and the worst-case perturbation of each training point. Formally,

$$\hat{f}_{\text{std}} = \min_{f\in\mathcal{F}} \sum_{i=1}^{n} \ell(f, x_i, y_i) + \lambda \|f\| \tag{2}$$

$$\hat{f}_{\text{rob}} = \min_{f\in\mathcal{F}} \sum_{i=1}^{n} \max_{\tilde{x}_i \in T(x_i)} \ell(f, \tilde{x}_i, y_i) + \lambda \|f\|. \tag{3}$$

3. Simple convex setting

We construct a convex example to study the performance of the two estimators $\hat{f}_{\text{std}}$ (2) and $\hat{f}_{\text{rob}}$ (3). The convexity allows us to disentangle the training objectives from the exact optimization algorithm. Concretely, we show a 1D regression setting where both $\hat{f}_{\text{std}}$ and $\hat{f}_{\text{rob}}$ are consistent, but the convergence behavior of these estimators depends on the interaction between $p^\star$ and $T(x)$. By changing one parameter of the distribution, we can go from a case where robust estimators have higher test loss than standard estimators (i.e. robustness *hurts*) to a case where robust estimators have lower loss and robustness *helps*.

3.1. Consistency of robust and standard solutions

We assume that $f^\star(x) = \mathbb{E}_{p^*}[y \mid x]$ is in the hypothesis class $\mathcal{F}$. We ensure that $\hat{f}_{\text{std}}$ and $\hat{f}_{\text{rob}}$ are consistent and converge to $f^\star$ as $n \to \infty$ by constructing the data distribution $p^\star$ and perturbations sets $T(x)$ such that

$$p^\star(y \mid x) = p^\star(y \mid \tilde{x}) \quad \forall x \in \mathcal{X},\ \tilde{x} \in T(x). \tag{4}$$

for all $x \in \mathcal{X}$ (see Appendix A). In other words, the true underlying function is invariant across the perturbation set. Intuitively, it would seem that enforcing this invariance during estimation should serve as regularizer that reduces standard test error. However, as we show below, this is not always true.

3.2. Description of the regression setting

Perturbation sets. Motivated by the ℓ_∞ setting, we consider perturbation sets of the following form: $T(x) = \{x-\epsilon, x, x+\epsilon\}$. In other words, each input can be increased or decreased by ϵ.

Distribution over $\mathcal{X}$. For convenience, we consider a discrete domain that can be partitioned into a set X_{line} and a set of perturbations of points in X_{line}.

$$\begin{aligned} X_{\text{line}} &= \{0, 1, 2, \ldots s-1\} \\ \bar{T}(X_{\text{line}}) &= T(X_{\text{line}}) \setminus \{x\} \\ \mathcal{X} &= X_{\text{line}} \bigcup \bar{T}(X_{\text{line}}). \end{aligned} \tag{5}$$

The first key insight behind our construction is to keep the probability of $\bar{T}(X_{\text{line}})$ under $p^\star(x)$ smaller than δ for some *small* $\delta > 0$. Any estimator f that has small error only on set X_{line} (but large error on the perturbations $\bar{T}(X_{\text{line}})$) will also have small test error (over the entire $p^\star(x)$), but has large robust error. Such an estimator's performance would mimic standard trained models in practice.

Distribution over $\mathcal{Y}$. We now describe the distribution $p^\star(y \mid x)$. Let $f^\star = \mathbb{E}_{p^*}[y \mid x]$. We assume that $p^\star(y \mid x) = f^\star(x) + \sigma$ where $\sigma \sim \mathcal{N}(0,1)$.

The second key insight behind our construction is that the robust estimator can be forced to be more *complex* than the standard estimator. We let $f^\star(x) = mx\ \forall x \in X_{\text{line}}$. By assumption (4) that guarantees consistency, we have for $\tilde{x} = T(x) \in \bar{T}(X_{\text{line}}), f^\star(\tilde{x}) = mx$. By definition of the data description above, for small number of samples, most of the training points $[x_i]_1^n$ are from X_{line}. The standard estimator $\hat{f}_{\text{std}}$ (with sufficient regularization λ) is a simple linear estimator that fits the training points. The robust estimator $\hat{f}_{\text{rob}}$ however is forced to be invariant on $[T(x_i)]_{i=1}^n$ by construction, and hence is no longer necessarily the simple linear estimator. The slope m controls how complex the robust solution needs to be. Inspired by the shape, we denote this regression setting parameterized by slope m as $\texttt{Stairs}(m)$.

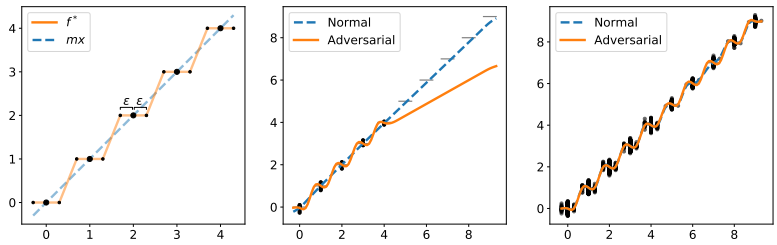
At one extreme, when the slope is 1, we see that any robust estimator that fits $T(x)$ resembles a *staircase* and is more complex. See Figure 1(a). At the other extreme, when the slope $m = 0$, a simple linear estimator is also robust. Figure 2(a) contains an illustration.

As we show in Section 4, when $m = 1$, the complexity of the robust solution results in poor generalization and consequently higher test error—robustness hurts!. However, when $m = 0$, the robust estimator does not suffer from worse generalization. In fact, robust estimator by enforcing invariance on $T(x)$ is less sensitive to noise and has lower test error than the standard estimator—robustness helps!

4. Experiments

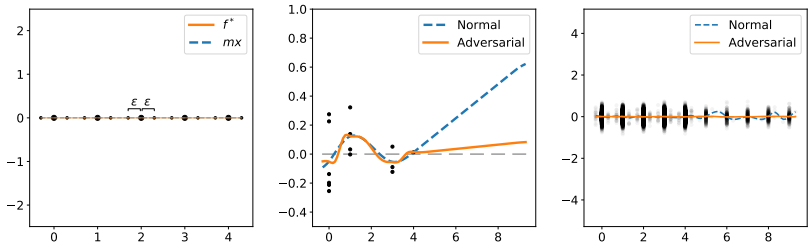
Brief setup. In this section, we experimentally investigate the test errors of the standard estimator (2) and robust estimators (3). We investigate how robustness interacts with generalization and consequently the test loss, as we vary the number of training samples. We report final test losses here; trends in generalization gap are almost identical as the two estimators obtain similar training losses. (See Appendix B.7 for plots.

1. **Convex example from Section 3**: We consider the square loss $\ell(\theta, x, y) = (y - f_\theta(x))^2$. We fit cubic B-splines (Friedman et al., 2001 2001) by minimizing the penalized square loss on the training set for different



(a) Slope $m=1$ (b) Small sample size (c) Large sample size

Figure 1. **Left:** An illustration of $f^\star(x)$ of the data distribution for the convex example $\texttt{Stairs}(m=1)$. The center point in each "stair" is in the set X_{line} and has high probability. The dashed blue line shows a simple solution that fits X_{line} and has low (but not optimal) test error, while the solid orange line shows the complex, optimal solution $f^\star$ that is both robust and accurate. **Middle:** With small sample size, the robust solution has large norm due to fitting the low probability points in $\bar{T}(x)$ thereby generalizing worse. **Right:** The standard and robust solutions have low test error with large number of training samples.



(a) Slope $m=0$ (b) Small sample size (c) Large sample size

Figure 2. (**Left:** An illustration of $f^\star(x)$ of the data distribution for the convex example $\texttt{Stairs}(m=0)$. The optimal robust and accurate solution $f^\star$ is simple. **Middle:** With small samples, the standard solution may overfit to noise, while robust training has a noise cancelling effect. **Right:** With large samples, the the robust and standard solution have similarly low test error.

values of the regularization parameter λ. We compute the standard estimator (2) and robust estimators (3) by optimizing the respective convex objectives. We simulate training data of varying sizes, and report the average and standard deviation over 2000 instances for small training set size and 25 instances for large training set size. See Appendix B.5 for more details.

2. **CIFAR-10 and MNIST.** We train Wide ResNet 40-2 (Zagoruyko and Komodakis, 2016) on CIFAR-10 , and a small convolutional network on MNIST. We report the 0-1 standard test loss for both standard and robust training. For robust training, we consider ℓ_∞ perturbations of varying ϵ and perform *adversarial training* using the gradient based approximation to the worst-case perturbation (Madry et al., 2018). See Appendix **??** for other details and hyperparameters.

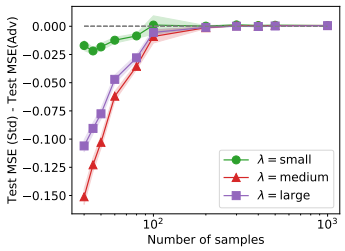
4.1. Cases where robustness hurts

We first consider the case where robustness hurts generalization and we see an increase in the test loss of the robust estimator compared to the standard estimator. Figure 3 shows the difference in test losses of the two estimators for the convex example $\texttt{Stairs}(m=1)$and CIFAR-10 where we observe this phenomenon.

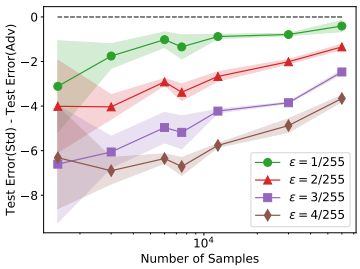
In the convex case, we set the underlying distribution to be $\texttt{Stairs}(m=1)$ where fitting the perturbation set $\bar{T}(X_{\text{line}})$ requires a more complex solution (See Figure 1). With few samples, this robust complex solution generalizes poorly leading to high test loss. In contrast, as the training set mostly consists of points from X_{line} (5), $\hat{f}_{\text{std}}(\lambda)$ is the simple solution that has low test loss. As we increase the number of samples, the robust complex solution $\hat{f}_{\text{rob}}(\lambda)$ also generalizes well, and we see that the test losses of both estimators converges to the same value (Figure 1(c)).

This trend occurs for all range of regularization parameters λ. However, we see the largest difference between standard and robust estimators for intermediate $\lambda_{\texttt{med}}$, where λ is large enough to cause $\hat{f}_{\text{std}}(\lambda_{\texttt{med}})$ to be simple, but $\hat{f}_{\text{rob}}(\lambda_{\texttt{med}})$ is complex.

We see that the above trends from the convex example are mirrored in CIFAR-10 : adversarially trained models (robust estimators) have larger test loss than the standard trained models. However, this difference gets smaller as the number

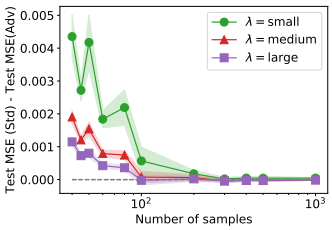


(a) $\texttt{Stairs}(m=1)$

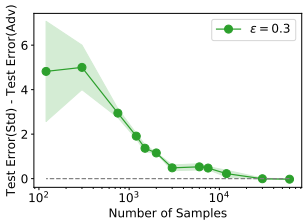


(b) Wide ResNet 40-2 on CIFAR-10

Figure 3. Difference between test losses for the standard estimator and robust estimator as a function of the # of training samples n. (a) For the convex case, we see that for small n, test loss of the standard estimator $L_{\text{test}}(\hat{f}_{\text{std}}(\lambda))$ is larger than $L_{\text{test}}(\hat{f}_{\text{rob}}(\lambda))$. As n increases, the gap closes. (b) On subsampling CIFAR-10 , we see that the gap between test loss of standard training and test loss of adversarially trained model (robust estimator) decreases on increasing the number of samples, just like the convex case in (a). Extrapolating, we see that the gap should close as sample size increases.



(a) $\texttt{Stairs}(m=0)$



(b) Small CNN on MNIST

Figure 4. Difference between test losses for the standard estimator and robust estimator, as a function of the number of training samples n. (a) In the convex case $\texttt{Stairs}(m=1)$, we see that the robust estimator $\hat{f}_{\text{rob}}(\lambda)$ outperforms the standard estimator $\hat{f}_{\text{std}}(\lambda)$ for all values of λ and different number of training samples. (b) On MNIST, we see that the adversarially trained model (robust estimator) has lower loss than the standard trained model. The difference in losses is largest for small number of samples. Both estimators converge to the same test error with increased number of training samples.

of training samples n increases. This leads us to hypothesize that the current *trade-off* observed between robustness and accuracy is due to insufficient samples for complex robust solutions to generalize.

4.2. Cases where robustness helps

We now consider the cases where robustness helps and the robust estimator $\hat{f}_{\text{rob}}$ has smaller test loss. Figure 4 presents the results on the convex example $\texttt{Stairs}(m=0)$ (defined in Section 3) and MNIST . In both cases, we see that the robust estimator has lower test loss than the standard estimator.

In the convex case, the slope of $m=0$ allows the robust estimator to be simple (See Figure 2), unlike the setting with $m=1$. Therefore, $\hat{f}_{\text{rob}}(\lambda)$ does not see an increase in test loss. In fact, because $\hat{f}_{\text{rob}}(\lambda)$ enforces that x and $\bar{T}(x)$ have the same value, by pooling these samples together, $\hat{f}_{\text{rob}}(\lambda)$ is less sensitive to noise (Figure 2). This is reflected in the largest gains of $\hat{f}_{\text{rob}}(\lambda_{\texttt{small}})$ over $\hat{f}_{\text{std}}(\lambda_{\texttt{small}})$ for small number of training samples. Both estimators are consistent and the test losses converge to the same value for very large number of samples.

On MNIST , we observed similar behavior where the difference in the test loss of the adversarially trained model and test loss of the standard trained model decreases with increasing number of samples. As we increase the number of training samples, both standard and adversarially trained models have the same small test error. We remark that Tsipras et al. (2019) report that the test loss of adversarially trained models is *higher* than the test loss of standard training. This is contrary to our results because the adversarially trained models of (Tsipras et al., 2019) haven't been trained to convergence. We discuss this in Section C.

5. Conclusion

In this work, we demonstrated that standard test accuracy performance of the models obtained using robust training is determined by the regime defined by the marginal distribution of the covariates over the perturbation set, sample size and complexity of the true function. We shed some light on the counter-intuitive phenomenon where enforcing invariance respected by the underlying true function can actually hurt performance in some settings.

References

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA: Springer series in statistics New York, NY, USA:, 2001 2001.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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

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

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

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019.

S. Zagoruyko and N. Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.

A. Robust and accurate models

We show that our label distribution invariance condition (restated here) is a sufficient condition for the minimizers of the standard and adversarial objectives to be the same on the support of p^*. Therefore, this joint minimizer is both robust and accurate. The label distribution invariance condition restricts the attack sets T to those where

$$p(y\,|\,x) = p(y\,|\,x') \quad \forall x' \in T(x) \tag{6}$$

for all $x \in \mathcal{X}$. For the classification case, we require only the weaker condition of label invariance, which is that

$$\operatorname*{argmax}_y p(y\,|\,x) = \operatorname*{argmax}_y p(y\,|\,x') \quad \forall x' \in T(x), \tag{7}$$

such that the adversary cannot change the label but can perturb the label distribution. Note that the condition (6) implies (7).

Theorem 1. *(Classification) Consider the minimizer of the standard population 0-1 loss, $f^* = \operatorname{argmin}_f L_{test}(f)$. Assuming (7) holds, we have that for any f, $\mathbb{E}[\max_{x' \in T(x)} \ell(f,x,y)] \geq \mathbb{E}[\max_{x' \in T(x)} \ell(f^*,x,y)]$, such that f^* is also optimal for the adversarial population 0-1 loss.*

Proof. Note that the optimal standard classifier is the Bayes classifier, such that $\operatorname{argmax}_j f^*(x)_j = \operatorname{argmax}_y p(y \mid x)_j$ with probability 1 over p^*. Then by condition (7), $\operatorname{argmax}_j f^*(x') = \operatorname{argmax}_j f^*(x)$ for all $x' \in T(x)$. Thus the adversarial population loss for f^* is

$$\begin{aligned}
\mathbb{E}[\max_{x' \in T(x)} \ell(f^*,x,y)] &= \mathbb{E}\left[\max_{x' \in T(x)} \mathbf{1}\{\operatorname*{argmax}_j f^*(x')_j = y\}\right] \\
&= \mathbb{E}\left[\mathbf{1}\{\operatorname*{argmax}_j f^*(x)_j = y\}\right] \\
&= \mathbb{E}[\ell(f^*,x,y)]
\end{aligned}$$

where we have used in the last line that f^* is the Bayes classifier and the condition (7). Noting that for any classifier f, $\mathbb{E}[\max_{x' \in T(x)} \ell(f,x,y)] \geq \mathbb{E}[\ell(f,x,y)]$, the theorem statement follows. □

In the regression case, we require that the conditional expectation of the label to be invariant, such that

$$\mathbb{E}[y\,|\,x] = \mathbb{E}[y\,|\,x'] \quad \forall x' \in T(x), \tag{8}$$

which again is implied by condition (6).

Theorem 2. *(Regression) Consider the minimizer of the standard population squared loss, $f^* = \operatorname{argmin}_f L_{test}(f)$. Assuming (7) holds, we have that for any f, $\mathbb{E}[\max_{x' \in T(x)} \ell(f,\,x,\,y)] \geq \mathbb{E}[\max_{x' \in T(x)} \ell(f^*,\,x,\,y)]$, such that f^* is also optimal for the adversarial population squared loss.*

Proof. Note that the optimal standard model is the Bayes estimator, such that $f^*(x) = \mathbb{E}[y\,|\,x]$. Then by condition (8), $f^*(x') = \mathbb{E}[y\,|\,x'] = \mathbb{E}[y\,|\,x] = f^*(x)$ for all $x' \in T(x)$. Thus the adversarial population loss for f^* is

$$\begin{aligned}
\mathbb{E}[\max_{x' \in T(x)} \ell(f^*,x,y)] &= \mathbb{E}\left[\max_{x' \in T(x)} (\mathbb{E}[y\,|\,x'] - y)^2\right] \\
&= \mathbb{E}\left[(\mathbb{E}[y\,|\,x] - y)^2\right] \\
&= \mathbb{E}[\ell(f^*,x,y)]
\end{aligned}$$

Then by the same argument as in Theorem 1, the theorem statement follows. □

B. Convex example details

B.1. MNIST

The MNIST dataset consists of 60000 labeled examples of digits. We sub-sample the dataset by factors of $\{1, 2, 5, 8, 10, 20, 40, 50, 80, 200, 500\}$ and report results for a small 3-layer CNN averaged from 2 trials for each sub-sample factor. All models are trained for 200 epochs and achieve 100% standard training accuracy in all cases. The adversarial models achieve $>99\%$ adversarial training accuracy in all cases. We train the adversarial models under the ℓ_∞ attack model with PGD adversarial training and $\epsilon = 0.3$. For computing the max in each training step, we use 40 steps of PGD, with step size 0.01 (the parameters used in (Madry et al., 2018)). The final robust test accuracy when training with the full training set was 91%.

B.2. CIFAR-10

We train Wide ResNet 40-2 models (Zagoruyko and Komodakis, 2016) using standard and adversarial training while varying the number of samples in CIFAR-10 . We sub-sample CIFAR-10 by factors of $\{1,2,5,8,10,20,40\}$, such that sub-sampling by a factor of 2 results in a random half of the dataset. For sub-sample factors 1 to 20, we report results averaged from 2 trials each for standard and adversarial training. For sub-sample factors greater than 20, we average over 5 trials. We train adversarial models under the ℓ_∞ attack model with ℓ_∞-norm constraints of sizes $\epsilon = \{1/255, 2/255, 3/255, 4/255\}$ using PGD adversarial training (Madry et al., 2018). The models are trained for 200 epochs, such that 100% standard training accuracy is achieved for both standard and adversarial models in all cases and $> 98\%$ adversarial training accuracy is achieved by adversarially trained models in most cases.

B.3. Data distribution

We focus on a 1-dimensional regression case. Let $s_0 \in \mathbb{N}$, $s \in \{s : s \geq s_0, s \in \mathbb{N}\}$ be integer parameters and $\delta \in [0,1]$, $\epsilon \in [0, 0.5)$, $\sigma > 0$, $m \in [0,1]$ be real-valued parameters. Define "stair centers" $0 \leq i \leq s-1$ where i is an integer. Let $w \in \Delta_s$ be a distribution over stair centers where Δ_s is the probability simplex of dimension s. We define the data distribution with the following generative process for one sample x. First, sample a stair center i from the categorical distribution described by w, such that $i \sim \text{Categorical}(w)$. Second, sample $x \in \mathbb{R}$ by perturbing i with probability δ such that

$$x = \begin{cases} i & \text{w.p. } 1-\delta \\ i-\epsilon & \text{w.p. } \delta/2 \\ i+\epsilon & \text{w.p. } \delta/2. \end{cases}$$

In particular, we set w such that $w_0, \ldots, w_{s_0}$ have the majority of probability mass. For our examples, we fix $s_0 = 5$, set

$$\hat{w}_j = \begin{cases} 1/s_0 & j < s_0 \\ 0.01 & j \geq s_0 \end{cases}$$

and define $w = \hat{w} / \sum_j \hat{w}$. We define the target distribution as $(Y \mid X = x) \sim \mathcal{N}(m\lfloor x \rceil, \sigma^2)$, where $\lfloor x \rceil$ rounds x to the nearest integer. The attack model is defined as $T(x) = \{\lfloor x \rceil - \epsilon, \lfloor x \rceil, \lfloor x \rceil + \epsilon\}$. See Figure 1 for an illustration.

Note that $T(x)$ is defined such that (4) holds, since for any $x_1, x_2 \in T(x)$, $\lfloor x_1 \rceil = \lfloor x_2 \rceil$ and thus $p(y \mid x_1) = p(y \mid x_2)$, which is defined since $p(x') > 0$ for any $x' \in T(x)$.

B.4. Model

Our hypothesis class is the family of cubic B-splines as defined in (Friedman et al., 2001 2001). Cubic B-splines are piecewise cubic functions and can be viewed as a kernel method with kernel feature map $\Phi : \mathcal{X} \to \mathbb{R}^{3s+2}$. In our example, we fix the knots to be $\tau = [-\epsilon, 0, \epsilon, \ldots, (s-1)-\epsilon, s-1, (s-1)+\epsilon]$. Specifically, for some regularization parameter $\lambda \geq 0$ we optimize the penalized smoothing spline loss function over parameters θ,

$$\ell_{\text{spl}}(f_\theta(x), y) = (y - f_\theta(x))^2 \tag{9}$$

$$+ \lambda \int (f''_\theta(t))^2 dt = (y - \Phi(x)\theta)^2 + \lambda \theta^T \Omega \theta, \tag{10}$$

where $\Omega_{i,j} = \int \Phi''(t)_i \Phi''(t)_j dt)$ measures smoothness in terms of the second derivative. Complexity of these models are measured by the norm $\|f\|_\Omega = \theta^T \Omega \theta$.

B.5. Spline implementation

We implement the spline example using the basis described in (Friedman et al., 2001 2001). We construct a regularization penalty matrix Ω which computes second-order finite differences of the parameters θ. We solve the standard spline objective by computing

$$\theta^*_{\text{std}} = (\Phi(X^n)^T \Phi(X^n) + \lambda\Omega)^{-1} \Phi(X^n)^T \mathbf{y},$$

where the i-th row of $\Phi(X^n) \in \mathbb{R}^{n \times 3s+2}$ is $\Phi(x_i)$ and $\mathbf{y} = [y_1, \ldots, y_n]$ is a vector of labels. We approximate adversarial training by augmenting each sample x in the training data with $\{x - \epsilon, x + \epsilon\}$, producing an augmented dataset X_{aug} with $3n$ examples, and then solving

$$\theta^*_{\text{adv}} = (\Phi(X_{\text{aug}})^T \Phi(X_{\text{aug}}) + \lambda\Omega)^{-1} \Phi(X_{\text{aug}})^T \mathbf{y}.$$

B.6. Role of different parameters

To construct an example of the trade-off, the main parameters needed are that the slope m is significantly nonzero and

that $\delta > 0$. If $m \approx 0$, then the adversarial objective aligns with global generalization and we observe a that robustness helps generalization. If $\delta > 0$, then $p(y \mid x')$ may not be defined for all $x' \in T(x)$. The example where robustness hurts generalization is insensitive to whether there is label noise, meaning that $\sigma = 0$ is sufficient to observe a trade-off.

However, the example when $m = 0$ relies on the fact that there is label noise ($\sigma > 0$). In contrast, the trade-on example is not sensitive to the value of δ. Robustness can help generaliaztion also for small nonzero values of m.

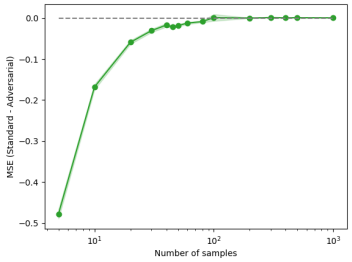
B.7. Plots of other values

We show plots for a variety of quantities for $\lambda = 0.00599$ in the $m = 1$ and $m = 0$ cases, with all the same parameters as before, while adding additional sample sizes $n = s/2$ to $n = 3s$. In both cases, the test MSE and generalization gap (difference between training MSE and test MSE) are almost identical due to low training error. In the $m = 1$ case (Figure 5), the adversarial solution finds higher norm solutions for all sample sizes, while the standard training finds low-norm solutions with good generalization. With enough samples, standard training begins to increase the norm of its solution and the training adversarial MSE starts to drop accordingly.

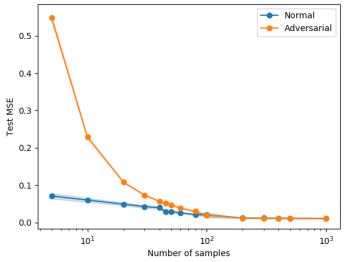
In the $m = 0$ case (Figure 6), the optimal solution is the line $f(x) = 0$, which has 0 norm. The adversarial solution has consistently low norm, while the standard solution starts with a lower norm solution which generalizes badly (wrong slope). With a larger amount of samples, the standard solution actually increases in norm.

C. Initialization and trade-off for MNIST

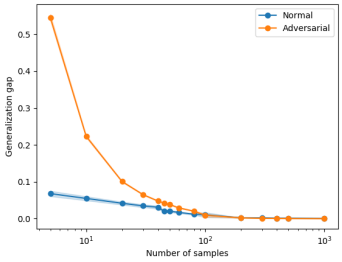
We note here that the tradeoff for adversarial training reported in (Tsipras et al., 2019) is because the adversarially trained model hasn't converged (even after a large number of epochs). On using the Xavier initialization, we get faster convergence with adversarial training and see no drop in clean accuracy at the same level of robust accuracy. Interestingly, standard training is not affected by initialization, while adversarial training is dramatically affected.



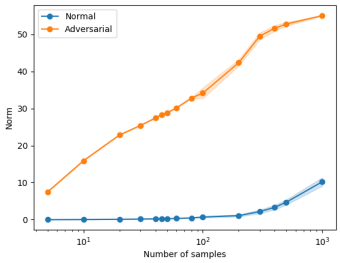
(a) Difference in Test MSE



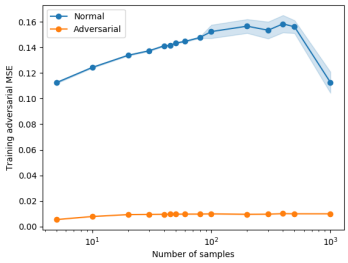
(b) Test MSE



(c) Generalization gap

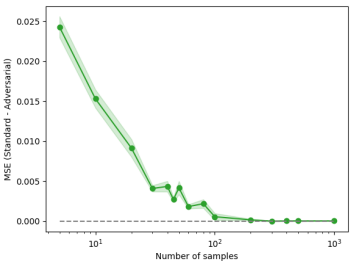


(d) Norm

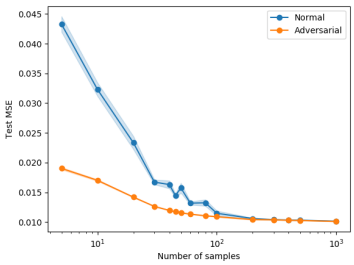


(e) Training adversarial MSE

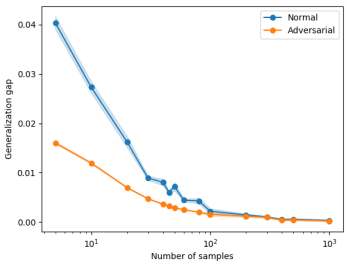
Figure 5. Plots of a variety of values as number of samples varies for $\lambda = 0.00599$ for the trade-off case ($m = 1$). Note that Test MSE is almost identical to generalization gap. Shaded regions are 1 STD.



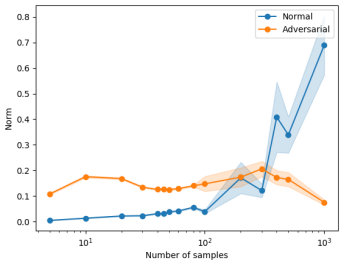
(a) Difference in Test MSE



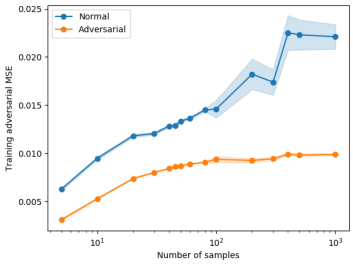
(b) Test MSE



(c) Generalization gap



(d) Norm



(e) Training adversarial MSE

Figure 6. Plots of a variety of values as number of samples varies for $\lambda = 0.00599$ for the trade-on case ($m = 0$). Note that Test MSE is almost identical to generalization gap. Shaded regions are 1 STD.