Adversarial Examples from Computational Constraints

Sébastien Bubeck [1] **Yin Tat Lee** [1 2] **Eric Price** [3] **Ilya Razenshteyn** [1]

Abstract

Why are classifiers in high dimension vulnerable to "adversarial" perturbations? We show that it is likely not due to information theoretic limitations, but rather it could be due to computational constraints. First we prove that, for a broad set of classification tasks, the mere existence of a robust classifier implies that it can be found by a possibly exponential-time algorithm with relatively few training examples. Then we give two particular classification tasks where learning a robust classifier is computationally intractable. More precisely we construct two binary classifications task in high dimensional space which are (i) information theoretically easy to learn robustly for large perturbations, (ii) efficiently learnable (non-robustly) by a simple linear separator, (iii) yet are not efficiently robustly learnable, even for small perturbations. Specifically, for the first task hardness holds for any efficient algorithm in the statistical query (SQ) model, while for the second task we rule out *any* efficient algorithm under a cryptographic assumption. These examples give an exponential separation between classical learning and robust learning in the statistical query model or under a cryptographic assumption. It suggests that adversarial examples may be an unavoidable byproduct of computational limitations of learning algorithms.

1. Introduction

The most basic task in learning theory is to learn from a data set $(X_i, f(X_i))_{i\in[n]}$ a good approximation to the unknown input-output function f. One is typically interested in finding a hypothesis function h with small out of sample probability of error. That is, assuming the X_i's are i.i.d. from some distribution D, one wishes to approximately minimize $\mathbb{P}_{X\sim D}(h(X) \neq f(X))$. A more challenging task is to learn a *robust* hypothesis, that is, one that would minimize the probability of error against *adversarially corrupted examples*. More precisely, assume that the input space is endowed with a norm $\|\cdot\|$ and let $\varepsilon > 0$ be a fixed robustness parameter. In robust learning the goal is to find h to minimize:

$$\mathbb{P}_{X\sim D}(\exists\, z \text{ such that } \|z\| \leq \varepsilon, \text{ and } h(X+z) \neq f(X+z))\,. \tag{1}$$

Such an input $X + z$ in the above event is colloquially referred to as an *adversarial example*[1].

Following Szegedy et al. (2013) there is a rapidly expanding literature exploring the vulnerability of neural networks to adversarially chosen perturbations. The surprising observation is that, say in vision applications, for most images $X \sim D$ the perturbation can be chosen in a way that is imperceptible to a human yet dramatically changes the output of state-of-the-art neural networks. This is a particularly important issue as these neural networks are currently being deployed in real-world situations. Naturally there is by now a large literature (in fact going back at least to (Dalvi et al., 2004; Globerson and Roweis, 2006)) on *attacks* (finding adversarial perturbations) and *defenses* (making classifiers robust against certain type of attacks).

While we have a sophisticated theory for the classical goal of minimizing the non-robust probability of error, our understanding of the robust scenario is still very rudimentary. At the moment, the "attackers" seem to be winning the arms race against the "defenders", see e.g., (Athalye et al., 2018). We identify four mutually exclusive possibilities for why all known classification algorithms are vulnerable to adversarial examples:

1. No robust classifier exists.
2. Identifying a robust classifier requires too much training data.
3. Identifying a robust classifier from limited training data

[*]Equal contribution [1]Microsoft Research, Redmond, Washington, USA [2]University of Washington, Seattle, Washington, USA [3]University of Texas, Austin, Texas, USA. Correspondence to: Ilya Razenshteyn <ilyaraz@microsoft.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

[1]In the literature one sometimes uses a more stringent definition of adversarial examples, where X and z are in addition required to satisfy $f(X + z) = f(X)$. We ignore this requirement here.

is information theoretically possible but computationally intractable.

4. We just have not found the right algorithm yet.

The goal of this paper is to provide two pieces of evidence, one in favor of hypothesis 3 and one against hypothesis 2. Our primary result is that hypothesis 3 is indeed possible: there exist robust classification tasks that are information theoretically easy but computationally intractable under a powerful model of computation (namely the statistical query model, see below) or for unrestricted efficient algorithms but under a cryptographic hardness assumption. Our secondary result is evidence against hypothesis 2, showing that if a robust classifier exists then it can be found with relatively few training examples under a standard assumption on the data distribution (for example, that the distribution within each label is close to a Lipschitz generative model, or is drawn from a finite set of exponential size).

In Section 1.1 we discuss related work on adversarial examples in light of those four hypotheses. In Section 1.2 we introduce the model of computation under which we will prove intractability. We conclude the introduction with Section 1.4 where we give a brief proof overview for our primary and secondary result. These results are discussed in greater depth respectively in Sections 4,5 and Section 3.

1.1. Related work on adversarial examples

To the best of our knowledge, previous works have not linked computational constraints to adversarial examples, but instead have focused on the other three hypotheses.

Supporting hypothesis 1 is the work of Fawzi et al. (2018). Here the authors consider a generative model for the features, namely $X = g(r)$ where $r \in \mathbb{R}^d$ is sampled from an isotropic Gaussian (in particular it is typically of Euclidean norm roughly $\sqrt{d}$). The observation is that, due to Gaussian isoperimetry, no classifier is robust to perturbations in r of Euclidean norm $O(1)$. If g is L-Lipschitz, this corresponds to perturbations of the image X of at most $O(L)$. On the other hand, evidence against hypothesis 1 is the fact that humans seem to be robust classifiers with low error rate (albeit nonzero error rate, as shown by examples in (Elsayed et al., 2018)). This suggests that, to fit real distributions on images, the Lipschitz parameter L in the data model assumed in (Fawzi et al., 2018) may be prohibitively large.

Another work arguing the inevitability of adversarial examples is Gilmer et al. (2018). There the authors propose a simple classification task, namely distinguishing between samples on the unit sphere in high dimension and samples on a sphere of radius R bounded away from 1. They show experimentally that even in such a simple setup, state-of-the-art neural networks have adversarial examples at most points. We note however that this example only applies to specific classifiers, since it is easy to construct an efficient robust classifier for the given example (e.g., just use a linear model on the norm of the features); thus the "hardness" here only appears for a given network structure.

Supporting hypothesis 2 is the work of Schmidt et al. (2018). Here the authors consider a mixture of two separated Gaussians (isotropic, with means at distance $\Theta(\sqrt{d})$). With such a separation a single sample is sufficient to learn non-robustly; but to learn a classifier that is robust to $O(1)$-size perturbations in ℓ_∞-norm one needs $\Omega(\sqrt{d})$ samples. This polynomial separation suggests that avoiding adversarial examples in high dimension requires a lot more samples than mere learning—but only up to $\sqrt{d}$ samples. In fact, since their hard instance is essentially a set of 2^d possible distributions, our secondary result gives a black-box algorithm that would produce a robust classifier with $O(d)$ samples.

Finally the large body of work on "adversarial defense" can be viewed as investigating hypothesis 4. We note that, at the time of writing, the state of the art defense Madry et al. (2018) (according to (Athalye et al., 2018)) is still far from being robust. Indeed on the CIFAR-10 dataset its accuracy is below 50% even with very small perturbations (of order 10^{-2} in ℓ_∞-norm), while state of the art non-robust accuracy is higher than 95%.

Update. Since the first version of the paper appeared (in May 2018), all the directions discussed above have seen lots of progress (for a small and by no means representative sample, see (Garg et al., 2018; Yin et al., 2018; Zhang et al., 2019)). However, the status quo does not seem to change significantly. Namely, the defense from (Madry et al., 2018) is still the state of the art for the CIFAR dataset, and there are no attacks that perform drastically better than simple projected gradient descent (PGD). Perhaps the most significant developments have been related to training models with *provable* robustness guarantees (Dvijotham et al., 2018; Wong and Kolter, 2018; Weng et al., 2018; Xiao et al., 2018), however, currently all the methods for such training are either extremely slow or the certified bound is much weaker than the bound achieved by the PGD attack.

1.2. The SQ model

Proving computational hardness is a notoriously difficult problem. To circumvent this difficulty one usually either (i) reduces the problem at hand to a well-established computational hardness conjecture (e.g., proving NP-hardness), or (ii) proves an unconditional hardness within a limited computational framework (such as the oracle lower bounds in convex optimization, (Nesterov, 2004)). Our task here is further complicated by the *average-case* nature of the problem (the datasets are i.i.d. from some fixed distribution).

Fortunately there is a growing set of results on computational hardness in learning theory that we can leverage. The statistical query (SQ) model of computation from Kearns (1998) is a particularly successful instance of approach (ii) for learning theory: (a) most known learning algorithms fall in the framework, including in particular logistic regression, SVM, stochastic gradient descent, etc; and (b) SQ-hardness has been proved for many interesting problems that are believed to be computationally hard, such as learning parity with noise (Kearns, 1998), learning intersection of halfspaces (Klivans and Sherstov, 2007), the planted clique problem (Feldman et al., 2013), robust estimation of high-dimensional Gaussians (Diakonikolas et al., 2017), or learning a function computable by a small neural network (Song et al., 2017). Thus we naturally use this model to prove our main result on the computational hardness of robust learning. We now recall the definition of the SQ model and state informally our main result.

As Kearns put it in his original paper, the SQ model considers "learning algorithms that construct a hypothesis based on statistical properties of large samples rather than on the idiosyncrasies of a particular sample". More precisely, rather than having access to a data set $(X_i, f(X_i))$, in the SQ model one must make queries to a τ-SQ oracle which operates as follows: given a $[0,1]$-valued function ψ defined on input/output pairs, the SQ oracle returns a value $\mathbb{E}_{X\sim\mathcal{D}}\,\psi(X, f(X)) + \xi$ where $|\xi| \leq \tau$. We refer to τ as the *precision* of the oracle. Obviously, an algorithm using T queries to an oracle with precision τ can be simulated using a data set of size roughly T/τ^2. In our main result we consider an oracle with *exponential* precision. More concretely we take τ of order $\exp(-Cd^c)$ where d is the dimension of the problem and $c, C > 0$ are some numerical constants. Observe that such a high precision oracle cannot be simulated with a polynomial (in d) number of samples. Yet we show that even with such a high precision one needs an exponential number of queries to achieve robust learning for a certain task which on the other hand is easy to learn, and information theoretically learnable robustly:

Theorem 1.1 (informal)**.** *For any $\varepsilon > 0$, there exists a classification task in $\mathbb{R}^d$ which is*

- *learnable in $\mathrm{poly}(d)$ time and $\mathrm{poly}(d)$ samples;*
- *robustly learnable in $\mathrm{poly}(d)$ samples with ℓ_2-robustness parameter $\log^{0.49} d$ (while with high probability all samples have ℓ_2-norm $O(\sqrt{d})$);*
- *not efficiently and robustly learnable in the statistical query model, in the sense that even with an exponential (in d) precision statistical query oracle one needs an exponential (in d) number of queries in order to robustly learn with robustness parameter ε.*

The same result holds using the ℓ_∞ norm instead of ℓ_2, except with diameter $O(\sqrt{d \log d})$.

Of course, a number of natural machine learning algorithms such as nearest neighbor are not based on statistical queries. Although we cannot prove it, we believe that our input distributions are computationally hard in general. For the case of nearest neighbor, the distance to points of each class have very similar distributions—indeed, the two distributions match on polynomially many moments. This suggests that exponentially many samples are necessary for nearest neighbor. For more information about nearest neighbor classifiers in the context of adversarial examples, see (Wang et al., 2017).

Moreover, there are very few problems in any domain with exponential SQ hardness for which polynomial time algorithms are known; in fact, the only such problems involve solving systems of linear equations over finite fields (Feldman, 2017). Since Theorem 1.1 involves a real-valued problem, finding a polynomial time algorithm that avoids the SQ lower bound would be a remarkable breakthrough in SQ theory.

1.3. Cryptographic hardness

We complement Theorem 1.1 with an alternative construction, which has qualitatively similar properties. However, there are two important differences. First, instead of $\log^{0.49} d$-robust classifier, we can guarantee the existence of a $\Omega(\sqrt{d})$-robust one (which is the best possible, since the diameter of the dataset is $O(\sqrt{d})$). Second, instead of ruling out efficient SQ algorithms, we can rule out *all* the efficient algorithms. However, this is of course not an unconditional result, and we show it under a cryptographic assumption.

More specifically, we build a classification task out of a pseudo-random generator from (Blum et al., 1986) and hardness of robust learning follows from computational indistinguishability of the output of the generator and the uniform distribution.

1.4. Overview of proofs

Our secondary result, on the information theoretic achievability of robustness, is proved via simple arguments reminiscent of PAC-learning theory. Namely, if a classifier is not good enough for a given pair of distributions, we can rule it out with high confidence by looking at not too many samples. Then, we use a union bound to claim the result for a family of pairs that is either at most exponentially large, or is at least covered by a net of at most exponential size (the only subtlety is in the proper definition of a net in this robust context).

Our primarily result, on the hardness of robustness, is technically much more challenging. The central object in the proof

is a natural high-dimensional generalization of a construction from Diakonikolas et al. (2017). Roughly speaking, a hard pair of distributions is obtained by taking a standard multivariate Gaussian, choosing a random k-dimensional subspace and planting there two well-separated distributions that match many moments of a Gaussian (in (Diakonikolas et al., 2017) only the case $k = 1$ is considered). To show an SQ lower bound, we use – as in (Diakonikolas et al., 2017) – the framework of (Blum et al., 1994; Feldman et al., 2013) to reduce the question to computing a certain non-standard notion of correlation between the distributions. To bound said correlation, we deviate from (Diakonikolas et al., 2017) significantly, since their argument is tailored crucially to the case $k = 1$. Our argument is less precise, but allows $k \gg 1$ which is necessary to obtain a large separation between the distributions (which in turn controls the parameter M in Theorem 1.1).

For the cryptographic hardness, we, roughly speaking, require a classifier to distinguish the uniform distribution from the output of the pseudo-random generator (PRG) on a uniformly random seed. Because the seed is much shorter than the output, extremely robust classifiers exist (since the image of the generator is tiny). In order to provide an example where an efficient robust classifier *exists* and is *information-theoretically easy* but *computationally intractable* to learn from data, we use a "trapdoor" PRG. The construction from (Blum et al., 1986) gives a trapdoor PRG under standard cryptographic assumptions (Vazirani and Vazirani, 1983).

2. Definitions

Throughout we restrict ourselves to binary classifiers, $\mathbb{R}^d$-feature space, as well as to balanced classes. We fix some norm $\|\cdot\|$ in $\mathbb{R}^d$, and we denote $B(\varepsilon) = \{z \in \mathbb{R}^d : \|z\| \leq \varepsilon\}$.

Definition 2.1. *The ε-robust zero-one loss (with respect to $\|\cdot\|$) is defined as follows, for $f : \mathbb{R}^d \to \{0,1\}$ and $(x,i) \in \mathbb{R}^d \times \{0,1\}$,*

$$\ell_\varepsilon(f,x,i) = \mathbb{1}\{\exists\, z \in B(\varepsilon) : f(x+z) \neq i\}\,.$$

Definition 2.2. *A binary classifier $f : \mathbb{R}^d \to \{0,1\}$ is (ε,δ)-robust for a pair of distributions (D_0, D_1) on $\mathcal{X}$ if for any $i \in \{0,1\}$,*

$$\mathop{\mathbb{E}}_{X\sim D_i}[\ell_\varepsilon(f,X,i)] \leq \delta\,.$$

Definition 2.3. *A (binary)* classification task *is given by a family $\mathcal{D}$ of pairs of distributions $D = (D_0, D_1)$ over a domain $\mathcal{X}$. The goal is to map datasets $\underline{X}_0, \underline{X}_1$ consisting of n i.i.d. samples from D_0 and D_1 respectively into a classifier $f : \mathbb{R}^d \to \{0,1\}$.*

We say that $\mathcal{D}$ is (ε,δ)-robustly learnable with n samples if there is a classification mapping such that, for every $D \in \mathcal{D}$, with probability at least $2/3$ over $\underline{X}_0$ and $\underline{X}_1$, the resulting classifier f is (ε,δ)-robust for D.

Remark 2.4. *The success probability $2/3$ is an arbitrary constant larger than $1/2$. It is easy to see that, for any $\eta > 0$, by using $O(n\log(1/\eta))$ samples one can obtain a success probability of $1-\eta$.*

We also note that the classical (ε',δ')-PAC learning scenario, with $\delta' = 1/3$, corresponds to our definition of (ε,δ)-robust classification with parameters $\varepsilon = 0$ and $\delta = \varepsilon'$. Slightly more precisely, a concept class $\mathcal{F} \subset \{0,1\}^{\mathbb{R}^d}$ for PAC-learning corresponds to the family $\mathcal{D}$ of all pairs of distribution supported respectively on $f^{-1}(0)$ and $f^{-1}(1)$ for some $f \in \mathcal{F}$.

Definition 2.5. *We say that $\mathcal{D}$ is (ε,δ)-robustly feasible if every $D \in \mathcal{D}$ admits an (ε,δ)-robust classifier. When it exists we denote f_D for such a classifier (chosen arbitrarily among all robust classifiers for D), and $\mathcal{F}_\mathcal{D} = \{f_D, D \in \mathcal{D}\}$.*

3. Robust learning with few samples

Obviously robust feasibility is a necessary condition for robust learnability. We show that it is in fact sufficient, even for *sample efficient* robust learnability. We first do so when a finite set of classifiers $\mathcal{F}_\mathcal{D}$ suffices for robust feasibility.

3.1. Robust empirical risk minimization

Theorem 3.1. *Assume that $\mathcal{D}$ is (ε,δ)-robustly feasible. Then it is $(\varepsilon,\delta+\delta')$-robustly learnable with $n = \Omega\left(\frac{\delta+\delta'}{\delta'^2}\log(|\mathcal{F}_\mathcal{D}|)\right)$.*

Proof. Let $\hat{D}_i = \frac{1}{n}\sum_{j=1}^n \delta_{\underline{X}_i(j)}$ be the empirical measure corresponding to the dataset $\underline{X}_i$. We will show that ERM on the ε-robust loss gives the claimed sample complexity. More precisely we consider the following classifier:

$$\hat{f} = \mathop{\arg\min}_{f\in\mathcal{F}_\mathcal{D}} \max_{i\in\{0,1\}} \mathop{\mathbb{E}}_{X\sim\hat{D}_i} \ell_\varepsilon(f,X,i)\,.$$

For shorthand notation we write $p_f = \max_{i\in\{0,1\}} \mathbb{E}_{X\sim D_i}\ell_\varepsilon(f,X,i)$ and $\hat{p}_f = \max_{i\in\{0,1\}} \mathbb{E}_{X\sim \hat{D}_i}\ell_\varepsilon(f,X,i)$. In particular we simply want to prove that $p_{\hat{f}} \leq \delta+\delta'$. Note that by definition $p_{f_D} \leq \delta$. A standard Chernoff bound gives that, with probability at least $2/3$, one has for *every* $f \in \mathcal{F}_\mathcal{D}$,

$$|p_f - \hat{p}_f| = O(\sqrt{p_f\log(|\mathcal{F}_\mathcal{D}|)/n})\,.$$

Now observe that for $n \geq 4\frac{\delta+\delta'}{\delta'^2}\log(|\mathcal{F}_\mathcal{D}|)$ one can has $\sqrt{p_{f_D}\log(|\mathcal{F}_\mathcal{D}|)/n} \leq \delta'/2$, and thus we obtain with $n = \Omega\left(\frac{\delta+\delta'}{\delta'^2}\log(|\mathcal{F}_\mathcal{D}|)\right)$,

$$p_{\hat{f}} - \frac{\delta'}{2}\sqrt{\frac{p_{\hat{f}}}{\delta+\delta'}} \leq \hat{p}_{\hat{f}} \leq \hat{p}_{f_D} \leq \delta+\delta'\,.$$

It now suffices to observe that $s \geq \delta + \delta'$ implies $s - \frac{\delta'}{2}\sqrt{\frac{s}{\delta+\delta'}} > \delta + \frac{\delta'}{2}$. □

3.2. Robust covering number

In many natural situations the classification task is specified by a continuous set of distributions. For example one might have a set of the form $\mathcal{D} = \{(g_0(w_0), g_1(w_1)), (w_0, w_1) \in \Omega\}$ where g_0 and g_1 are Lipschitz functions and Ω is some compact subset of $\mathbb{R}^{d'}$. In this case Theorem 3.1 does not apply, although one would like to say that "essentially" $\mathcal{D}$ is of log-size roughly d'. The classical solution to this difficulty is with covering numbers:

Definition 3.2. *For a metric space $(\mathcal{X}, \mathrm{dist})$ we write*

$$\mathcal{N}_{\mathrm{dist}}(\mathcal{X}, \varepsilon) = \inf\{|X| \text{ s.t. } X \subset \mathcal{X} \text{ and } \mathcal{X} \subset \bigcup_{x \in X} \{y : \mathrm{dist}(x, y) \leq \varepsilon\}\}.$$

With a slight abuse of notation we also extend the distance to the Cartesian product $\mathcal{X} \times \mathcal{X}$ by $\mathrm{dist}((x, x'), (y, y')) = \max(\mathrm{dist}(x, x'), \mathrm{dist}(y, y'))$.

With the above definitions one can obtain the following result as a straightforward corollary of Theorem 3.1 and the definition of total variation distance.

Theorem 3.3. *Assume that $\mathcal{D}$ is (ε, δ)-robustly feasible. Then $\mathcal{D}$ is $(\varepsilon, \delta + 2\delta')$-robustly learnable with $n = \Omega\left(\frac{\delta+\delta'}{\delta'^2} \log(\mathcal{N}_{\mathrm{TV}}(\mathcal{D}, \delta'))\right)$.*

In fact, if one is willing to lose a little bit of robustness, one can use a significantly weaker notion of "distance" than total variation. Indeed we can consider a broader class of modifications to a distribution that preserves the robustness of a classifier: in Theorem 3.3 we used that we can move arbitrarily a small amount of mass, but in fact we can also move *a little* an arbitrary amount of mass. While the former type of movement corresponds to total variation distance, the latter corresponds to the (infinity) Wasserstein distance. We denote $W_\infty(D, D')$ for the infimum of $\sup_{(x,x') \in \mathrm{supp}(\mu)} \|x - x'\|$ over all measures $\mu(x, x')$ with marginal over x (respectively x') equal to D (respectively D'). Next we introduce a slightly non-standard notion of covering with respect to a pair of distances

Definition 3.4. *For a metric space $\mathcal{X}$ equipped with two distances* dist *and* dist' *we define an (ε, δ) neighborhood by*[2]*:*

$$U_{\varepsilon,\delta}(x) = \left\{y : \exists z \text{ s.t. } \mathrm{dist}'(x, z) \leq \delta \text{ and } \mathrm{dist}(z, y) \leq \varepsilon\right\}.$$

[2]The choice of first moving with dist' and then with dist will fit our application. In general a more natural definition would be:

$$U_{\varepsilon,\delta}(x) = \{y : \exists x = z_1, z'_1, \ldots, z_n, z'_n = y \text{ s.t. } \sum_{i=1}^{n} \mathrm{dist}(z_i, z'_i) \leq \varepsilon \text{ and } \sum_{i=1}^{n-1} \mathrm{dist}'(z'_i, z_{i+1}) \leq \delta\}.$$

The corresponding covering number is:

$$\mathcal{N}_{\mathrm{dist},\mathrm{dist}'}(\mathcal{X}, \varepsilon, \delta) = \inf\{|X| \text{ s.t. } X \subset \mathcal{X} \text{ and } \mathcal{X} \subset \bigcup_{x \in X} U_{\varepsilon,\delta}(x)\}.$$

It is now easy to prove the following strengthening of Theorem 3.3:

Theorem 3.5. *Assume that $\mathcal{D}$ is (ε, δ)-robustly feasible. Then $\mathcal{D}$ is $(\varepsilon - \varepsilon', \delta + 2\delta')$-robustly learnable with $n = \Omega\left(\frac{\delta+\delta'}{\delta'^2} \log(\mathcal{N}_{W_\infty,\mathrm{TV}}(\mathcal{D}, \varepsilon', \delta'))\right)$.*

Proof. Let A be the set realizing the infimum in the definition of $\mathcal{N}_{W_\infty,\mathrm{TV}}(\mathcal{D}, \varepsilon', \delta')$. Observe that $\mathcal{D}$ is $(\varepsilon - \varepsilon', \delta + \delta')$-robustly feasible with classifiers from $\mathcal{F}_A$, and apply Theorem 3.1. □

3.3. Covering number bound from generative models

We now show that distributions approximated by generative models have bounded covering numbers (in terms of Definition 3.4), so Theorem 3.5 gives a good sample complexity for such distributions. The proof is deferred to Appendix C in the supplementary material.

Definition 3.6. *A* generative model $g_w : \mathbb{R}^k \to \mathbb{R}^d$ *is a neural network indexed by weights $w \in \mathbb{R}^m$. The* generated distribution $D(g_w)$ *is the distribution given by $g_w(x)$ for $x \sim N(0, I_k)$.*

Lemma 3.7. *Let g_w be an ℓ-layer neural network architecture with at most d activations in each layer and Lipschitz nonlinearities such as ReLUs. Consider any family of distribution pairs $\mathcal{D}$ such that for each $D \in \mathcal{D}$, and each $i \in \{0, 1\}$, there exists some $w \in [-B, B]^m$ with $W_\infty(D_i, D(g_w)) \leq \varepsilon$. Then*

$$\log\left(\mathcal{N}_{W_\infty,\mathrm{TV}}(\mathcal{D}, \varepsilon + \delta, \delta)\right) \leq O(m\ell \log(dB/\delta)).$$

4. Lower bound for the SQ model

Let D_0 and D_1 be two distributions over a set $\mathcal{X}$, for which we would like to solve a (binary) classification task. The SQ model, introduced in (Kearns, 1998), is defined as follows. An algorithm is allowed to access D_0 and D_1 through *queries* of the following kind. A query is specified by a function $h\colon \mathcal{X} \to [0, 1]$, and the response is two

numbers $u, v \in \mathbb{R}$ such that $u \in \mathbb{E}_{x\sim D_0}[h(x)] \pm \tau$ and $v \in \mathbb{E}_{x\sim D_1}[h(x)] \pm \tau$. Here $\tau > 0$ is a positive parameter called *precision*. After asking a number of such queries, the algorithm must output a required (robust or non-robust) classifier for D_0 and D_1.

Our main result is as follows:

Theorem 4.1. *For every sufficiently small $\rho, \gamma > 0$ the following holds. There exists a family of $2^{d^{O(1)}}$ pairs of distributions $(\widetilde{D}_0, \widetilde{D}_1)$ over $\mathbb{R}^d$ such that:*

- *Almost all the mass of $\widetilde{D}_0$ and $\widetilde{D}_1$ is supported in an ℓ_2-ball of radius $O(\sqrt{d})$;*
- *The distributions $\widetilde{D}_0$ and $\widetilde{D}_1$ admits a $(\Omega(\sqrt{1/\gamma}), 2^{-d^{\Omega(\gamma)}})$-robust classifier; moreover, a $\Omega(\sqrt{1/\gamma}), 0.01)$-robust classifier can be learned from $O(d)$ samples from D_0 and D_1;*
- *For $\widetilde{D_0}$ and $\widetilde{D_1}$, there exists a linear (non-robust) classifier, which can be learned in polynomial time;*
- *For every $\varepsilon > \rho$, in order to learn a $(\varepsilon, 0.01)$-robust classifier for $\widetilde{D}_0$ and $\widetilde{D}_1$, one needs at least $2^{d^{\Omega(1)}}$ statistical queries with accuracy as good as $2^{-d^{\Omega(\gamma)}}$.*

For instance, if γ is a small constant we get the existence of a C-robust classifier, where C is a large constant. One could push C as high as $\Omega(\log^{1/2-\varepsilon} d)$ at a cost of the lower bound being against SQ queries with somewhat worse accuracy ($2^{-2^{\log^{\Omega(\varepsilon)} d}}$ instead of $2^{-d^{\Omega(1)}}$).

We first show a family of pairs (D_0, D_1) that admit a robust classifier, yet it is hard (in the SQ model) to learn *any* (non-robust) classifier. Later, in Section 4.3, we show a simple modification of this family to obtain the main result.

4.1. Hard family of distributions

Here we define a hard family of pairs of distributions (D_0, D_1) as discussed above. This section contains the definition and key properties of the family; proofs of those properties appear in Appendix A. This family can be seen to be a high-dimensional generalization and modification of a family considered in (Diakonikolas et al., 2017). The family depends on three parameters: integers $1 \le k \le d$, $m \ge 1$ and a positive real $\varepsilon > 0$.

Fix an integer $m \ge 1$. We introduce two auxiliary distributions over $\mathbb{R}$ that we will use later as building blocks.

Lemma 4.2. *There exist two distributions D_A and D_B over $\mathbb{R}$ with everywhere positive p.d.f.'s $A(t)$ and $B(t)$ respectively such that:*

- *D_A and D_B match $N(0,1)$ in the first m moments;*

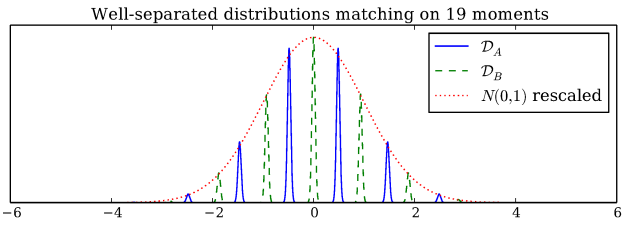

Figure 1. The distributions in Lemma 4.2 are similar to discretized Gaussians, with careful discretization and weighting from Gauss-Hermite quadrature.

- *There exist two subsets $S_A, S_B \subset \mathbb{R}$ such that the distance between S_A and S_B is at least $\Omega(1/\sqrt{m})$, $\mathbb{P}_{x\sim D_A}[x \in S_A] \ge 1 - e^{-\Omega(m)}$, and $\mathbb{P}_{x\sim D_B}[x \in S_B] \ge 1 - e^{-\Omega(m)}$;*
- *$A, B \in C^\infty$, and for every $0 \le l \le m+1$ and t, one has: $|\frac{d^l}{dt^l}\frac{A(t)}{G(t)}|, |\frac{d^l}{dt^l}\frac{B(t)}{G(t)}| \le m^{O(l+1)}$.*

(See Figure 1 for the illustration.)

Next let us fix parameters $1 \le k \le d$ and $\varepsilon > 0$. Let $\mathcal{U} = \{U_i\}$ be a family of k-dimensional subspaces of $\mathbb{R}^d$ with fixed orthonormal bases such that for every $i \ne j$ and $u \in U_i$, one has: $\|\mathrm{proj}_{U_j} u\|_2 \le \varepsilon \cdot \|u\|_2$. Informally speaking, subspaces from $\mathcal{U}$ are pairwise near-orthogonal.

Lemma 4.3. *For every $k \le d^{\Omega(1)}$, there exists such a family $\mathcal{U}$ with $\varepsilon \le d^{-0.49}$ and $|\mathcal{U}| = 2^{d^{\Theta(1)}}$.*

Now we are ready to define our family of hard pairs (D_0, D_1) of distributions over $\mathbb{R}^d$. The family is parameterized by a k-dimensional subspace $U \in \mathcal{U}$ together with an orthonormal basis $u_1, u_2, \ldots, u_k \in U$, where $\mathcal{U}$ is the family of subspaces guaranteed by Lemma 4.3. Let us extend the above basis to a basis for the whole $\mathbb{R}^d$: $u_1, u_2, \ldots, u_d$. Now we define a pair of distributions $D_{U,A}$ and $D_{U,B}$ via their p.d.f.'s $A_U(x)$ and $B_U(x)$ respectively as follows:

$$A_U(x) = \prod_{i=1}^{k} A(\langle x, u_i\rangle) \cdot \prod_{i=k+1}^{d} G(\langle x, u_i\rangle)$$

$$\text{and} \qquad B_U(x) = \prod_{i=1}^{k} B(\langle x, u_i\rangle) \cdot \prod_{i=k+1}^{d} G(\langle x, u_i\rangle),$$

where $A(\cdot)$ and $B(\cdot)$ are densities of distributions D_A and D_B from Lemma 4.2, and $G(t) = \frac{1}{\sqrt{2\pi}} \cdot e^{-t^2/2}$ is the p.d.f. of the standard Gaussian distribution $N(0,1)$. Now we simply take D_0 to be $D_{U,A}$ and D_1 to be $D_{U,B}$.

Lemma 4.4. *There exist two sets $S_{U,A}, S_{U,B} \subset \mathbb{R}^d$ such that the distance between $S_{U,A}$ and $S_{U,B}$ is $\Omega(\sqrt{k/m})$, and for which $\mathbb{P}_{x\sim D_{U,A}}[x \in S_{U,A}] \ge 1 - e^{-\Omega(km)}$ and $\mathbb{P}_{x\sim D_{U,B}}[x \in S_{U,B}] \ge 1 - e^{-\Omega(km)}$.*

As a result, the pair (D_0, D_1) admits a $(\Omega(\sqrt{k/m}), e^{-km^{\Omega(1)}})$-robust classifier. Moreover, since $\log|\mathcal{U}| \leq O(d)$ (which follows from standard bounds on the number of pairwise near-orthogonal unit vectors in $\mathbb{R}^d$), it follows from Theorem 3.1 that one can learn a $(\Omega(\sqrt{k/m}), 0.01)$-robust classifier from merely $O(d)$ samples.

4.2. SQ lower bound for learning a classifier for $D_{U,A}$ and $D_{U,B}$

The heart of the matter is to show that it requires $2^{d^{\Omega(1)}}$ statistical queries with precision $\tau = 2^{-d^{\Theta(\gamma)}}$ to learn a classifier for $D_{U,A}$ and $D_{U,B}$ provided that all the parameters m, k, ε are set correctly. The argument is fairly involved and uses the framework of (Feldman et al., 2013) to reduce the question to that of upper bounding χ-correlation between the distributions. Due to space limitations, we show the argument in Appendix B of the supplementary material.

4.3. Making the distribution easy to learn non-robustly

Let us now show a family of pairs distributions $(\widetilde{D}_0, \widetilde{D}_1)$ over $\mathbb{R}^{d+1}$ such that it is easy to learn a (non-robust) classifier, but hard to learn a robust one. The construction is very simple: we take distributions (D_0, D_1) over $\mathbb{R}^d$ as defined above and define $x \sim \widetilde{D}_0$ to be $x = (0, y_1, y_2, \ldots, y_d)$, where $y \sim D_0$, and, similarly, $x \sim \widetilde{D}_1$ to be $x = (\rho, y_1, y_2, \ldots, y_d)$, where $y \sim D_1$ and $\rho > 0$. These distributions admit a trivial (non-robust) classifier based on the first coordinate. Moreover, since $\widetilde{D}_0$ and $\widetilde{D}_1$ are linearly separable, they can be classified using linear SVM or logistic regression. Information-theoretically, one can learn a $(\sqrt{1/\gamma}, 0.1)$-robust classifier using $O(d)$ samples by ignoring the first coordinate and applying Theorem 3.1. However, for every $\varepsilon > \rho$, one needs $2^{d^{\Omega(1)}}$ SQ queries with accuracy $2^{-d^{\Theta(\gamma)}}$ to learn an $(\varepsilon, 0.1)$-robust separator. This can be shown exactly the same way as for D_0 and D_1 (see Appendix B in the supplementary material).

The above distributions are hard to learn robustly with respect to the ℓ_2 norm. We can switch to ℓ_∞ by replacing x by its Hadamard transform Hx. Since $\|Hx - Hy\|_\infty \geq \|H(x-y)\|_2/\sqrt{d} = \|x-y\|_2$, the robustness parameters in the theorem are unchanged while the diameter becomes $O(\sqrt{d \log d})$.

5. Cryptographic hardness

5.1. Hard-to-compute robust classifiers

We will now exhibit a binary classification task that admits a maximally robust classifier (that is, robust to perturbations comparable to the *diameter of the support*), yet any *efficiently computable classifier* has an accuracy close to random guessing.

Let $G : \{0,1\}^{d/2} \to \{0,1\}^d$ be a cryptographic pseudo-random generator (PRG). Let $\mathcal{D}_0$ be uniform on $\{0,1\}^n$ and $\mathcal{D}_1$ be the distribution of $G(s)$ for s uniform in $\{0,1\}^{d/2}$. Clearly a simple volume argument shows that there exists a classifier A which satisfies (1) for $\varepsilon = \Theta(\sqrt{d})$ (i.e., this problem admits a maximally robust classifier). Yet by definition of a PRG no polynomial time algorithm can have a non-trivial classification accuracy (let alone robust accuracy).

5.2. Adversarial examples and trapdoor PRG

Given Section 5.1, our goal is now to construct a classification task which admits a maximally robust classifier *that is also efficiently computable*, yet one cannot get non-trivial accuracy in polynomial time. The main idea here is to replace the PRG in the construction of Section 5.1 with a *trapdoor PRG*. In a nutshell a trapdoor PRG comes with a *key*, such that knowing the key allows to efficiently distinguish the PRG from a true source of randomness (and thus allows for efficient classification in the construction of Section 5.1). Note also that, by a simple union bound, the sample complexity of such a problem would be of order of the number of bits in the key.

Let us now detail the construction a bit more. For the sake of concreteness, we use a specific trapdoor PRG, namely the Blum–Blum–Shub PRG (Blum et al., 1986) (in its "backward" form). Let p and q be large distinct prime numbers congruent to 3 mod 4, let $N = pq$ and $d = O(\log(N))$. The BBS PRG $G_N : \{0,1\}^d \to \{0,1\}^*$ works as follow. First it maps the seed $s \in \{0,1\}^d$ to $x_0 \in \mathbb{N}$ a quadratic residue mod N in such a way that a uniformly random seed gives a nearly-uniform quadratic residue modulo N. Next it iteratively takes square roots mod N, that is let x_{i+1} be such that $x_i = x_{i+1}^2 \mod N$ and x_{i+1} is a quadratic residue itself (this is well-defined per our assumption on p and q). The i^{th} element of the output of G_N is then simply the parity of x_i.

The key property of the BBS PRG is that, without knowing the factorization $N = pq$, its output is computationally indistinguishable (under the quadratic residuosity assumption) from a true source of randomness (even when the seed is known), while on the other hand knowing the factorization allows for efficient distinguishing. To make this mathematically precise let us recall the notion of *computational* statistical distance for a family of pairs of distribution $\{(D_0(\omega), D_1(\omega)), \omega \in \Omega\}$: it is the supremum over all polynomial-time algorithms of the infimum over $\omega \in \Omega$ of the success probability one can have to identify whether a random sample was generated from $D_0(\omega)$ or generated from $D_1(\omega)$. Let us fix some constant $c > 1$ and denote $\mathcal{D}_0^n = \mathrm{unif}(\{0,1\}^{d^c})$ and $\mathcal{D}_1^n(N)$ the distribution of the

first d^c bits of $s \circ G_N(s)$ where s is a uniformly random element of $\{0,1\}^d$.

Theorem 5.1 ((Blum et al., 1986; Vazirani and Vazirani, 1983)). *Assuming that for infinitely many N the computational statistical distance of $\{(\mathcal{D}_0^d, \mathcal{D}_1^d(pq))\}_{p,q}$ is greater than $1/2+1/\mathrm{poly}(d)$ would refute the* quadratic residuosity assumption.

On the other hand, if p and q are known, then the computational statistical distance of $\{(\mathcal{D}_0^d, \mathcal{D}_1^d(N))\}$ is $1 - o_d(1)$.

From the above discussion we have the following properties for the classification task described by the family $\{(\mathcal{D}_0^d, \mathcal{D}_1^d(pq))\}_{p,q}$:

a. The (robust) sample complexity of this family is $O(d)$.

b. Any task in this family admits a maximally robust classifier (same volume argument as in Section 5.1) that is also efficiently computable (second statement in Theorem 5.1).

c. Under the quadratic residuosity assumption, any polynomial time learning algorithm for this family has an accuracy close to chance on some task in the family (first statement in Theorem 5.1).

We also note that, using the trick of adding a dummy coordinate revealing the label from Section 4.3, one could replace property c by c' and add property d as follows (for any fixed $\varepsilon > 0$):

c'. Under the quadratic residuosity assumption, any polynomial time learning algorithm for this family has a ε-robust accuracy close to chance on some task in the family.

d. One can learn non-robustly in polynomial time (and polynomial sample complexity).

Remark: After the preliminary version of the present paper appeared, we got notified by Degwekar and Vaikuntanathan that there is an issue with the above construction. Namely, it is not clear that the item b holds (the existence of an efficient robust classifier), since one can corrupt seed, which prevents us from (efficiently) distinguishing $\mathcal{D}_0$ and $\mathcal{D}_1$. However, as they explain in their paper (Degwekar and Vaikuntanathan, 2019), this can be remedied by post-composing our construction with a constant-rate linear distance efficient error-correcting code. We refer the reader to (Degwekar and Vaikuntanathan, 2019) for a further discussion.

6. Conclusion and future directions

In this paper we put forward the thesis that adversarial examples might be an unavoidable consequence of computational constraints for learning algorithms. Our main piece of evidence is two classification tasks, for which there exist classifiers robust to large Euclidean perturbations, yet finding *any* non-trivial robust classifier is hard in the statistical query model or under a cryptographic hardness assumption. The most important question for the validity of our thesis is whether one could prove a similar hardness result for *natural* distributions. This is a particularly challenging open problem as the concept of a natural distribution is fuzzy (for instance there is no consensus on what a natural distribution for images should look like).

References

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, ICML '18, 2018. URL https://arxiv.org/abs/1802.00420.

Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 253–262. ACM, 1994.

Lenore Blum, Manuel Blum, and Mike Shub. A simple unpredictable pseudo-random number generator. *SIAM Journal on computing*, 15(2):364–383, 1986.

Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 99–108. ACM, 2004.

Akshay Degwekar and Vinod Vaikuntanathan. Computational limitations in robust classification and win-win results. *arXiv preprint arXiv:1902.01086*, 2019.

Ilias Diakonikolas, Daniel Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *Proceedings of the fifty-eighth Annual Symposium on Foundations of Computer Science*, FOCS '17, 2017. URL https://arxiv.org/pdf/1611.03473.pdf.

Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. *arXiv preprint arXiv:1803.06567*, 2018.

Gamaleldin F Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alex Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial examples that fool both human and computer vision. *arXiv preprint arXiv:1802.08195*, 2018.

Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier, 2018. URL https://arxiv.org/pdf/arXiv:1802.08686.pdf.

Vitaly Feldman. A general characterization of the statistical query complexity. *Proceedings of Machine Learning Research vol*, 65:1–46, 2017.

Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, STOC '13, pages 655–664. ACM, 2013.

Shivam Garg, Vatsal Sharan, Brian Zhang, and Gregory Valiant. A spectral view of adversarially robust features. In *Advances in Neural Information Processing Systems*, pages 10159–10169, 2018.

Amparo Gil, Javier Segura, and Nico M Temme. Asymptotic approximations to the nodes and weights of gauss–hermite and gauss–laguerre quadratures. *Studies in Applied Mathematics*, 140(3):298–332, 2018.

Justin Gilmer, Luke Metz, Fartash Faghri, Sam Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. In *International Conference on Learning Representations Workshop*, 2018. URL https://arxiv.org/pdf/1801.02774.pdf.

Amir Globerson and Sam Roweis. Nightmare at test time: Robust learning by feature deletion. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 353–360. ACM, 2006.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

Adam R. Klivans and Alexander A. Sherstov. Unconditional lower bounds for learning intersections of halfspaces. *Machine Learning*, 69(2):97–114, 2007.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/pdf/1706.06083.pdf.

Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Publishers, 2004.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data, 2018. URL https://arxiv.org/pdf/arXiv:1804.11285.pdf.

Le Song, Santosh Vempala, John Wilmes, and Bo Xie. On the complexity of learning neural networks. In *Advances in Neural Information Processing Systems*, pages 5520–5528, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2013. URL https://arxiv.org/pdf/1312.6199.pdf.

Gabor Szego. *Orthogonal polynomials*, volume 23. American Mathematical Soc., 1939.

Umesh V Vazirani and Vijay V Vazirani. Trapdoor pseudo-random number generators, with applications to protocol design. In *Foundations of Computer Science, 1983., 24th Annual Symposium on*, pages 23–30. IEEE, 1983.

Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. *arXiv preprint arXiv:1706.03922*, 2017.

Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S Dhillon, and Luca Daniel. Towards fast computation of certified robustness for relu networks. *arXiv preprint arXiv:1804.09699*, 2018.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5283–5292, 2018.

Kai Y Xiao, Vincent Tjeng, Nur Muhammad Shafiullah, and Aleksander Madry. Training for faster adversarial robustness verification via inducing relu stability. *arXiv preprint arXiv:1809.03008*, 2018.

Dong Yin, Kannan Ramchandran, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. *arXiv preprint arXiv:1810.11914*, 2018.

Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S Dhillon, and Cho-Jui Hsieh. The limitations of adversarial training and the blind-spot attack. *arXiv preprint arXiv:1901.04684*, 2019.