First-order Adversarial Vulnerability of Neural Networks and Input Dimension

Carl-Johann Simon-Gabriel [1 2] **Yann Ollivier** [2] **Bernhard Schölkopf** [1] **Léon Bottou** [2] **David Lopez-Paz** [2]

Abstract

Over the past few years, neural networks were proven vulnerable to adversarial images: targeted but imperceptible image perturbations lead to drastically different predictions. We show that adversarial vulnerability increases with the gradients of the training objective when viewed as a function of the inputs. Surprisingly, vulnerability does not depend on network topology: for many standard network architectures, we prove that at initialization, the ℓ_1-norm of these gradients grows as the square root of the input dimension, leaving the networks increasingly vulnerable with growing image size. We empirically show that this dimension dependence persists after either usual or robust training, but gets attenuated with higher regularization.

1. Introduction

Following the work of Goodfellow et al. (2015), Convolutional Neural Networks (CNNs) have been found vulnerable to adversarial examples: an adversary can drive the performance of state-of-the art CNNs down to chance level with imperceptible changes to the inputs.

Based on a simple linear model, Goodfellow et al. already noted that adversarial vulnerability should depend on input dimension. Gilmer et al. (2018); Shafahi et al. (2019) later confirmed this, by showing that adversarial robustness is harder to obtain with larger input dimension. However, these results are different in nature from Goodfellow et al.'s original observation: they rely on assumptions on the dataset that amount to a form of uniformity in distribution over the input dimensions (e.g. concentric spheres, or bounded densities with full support). In the end, this analysis tends to incriminate the data: if the data can be anything, and in particular if it can spread homogeneously across many input dimensions, then robust classification gets harder.

Image datasets do not satisfy these assumptions: they do not have full support, and their probability distributions get more and more peaked with larger input dimension (pixel correlation increases). Intuitively, for image classification, higher resolution should help, not hurt. Hence data might be the wrong culprit: if we want to understand the vulnerability of our classifiers, then we should understand what is wrong with our classifiers, not with our images.

We therefore follow Goodfellow et al.'s original approach, which explains adversarial vulnerability by properties of the classifiers. Our main theoretical results start by formally extending their result for a single linear layer to almost all current deep feedforward network architectures. There is a further correction: based on the gradients of a linear layer, Goodfellow et al. predicted a linear increase of adversarial vulnerability with input dimension d. However, they did not take into account that a layer's typical weights decrease like $\sqrt{d}$. Accounting for this, the dependence becomes $\sqrt{d}$ rather than d, which is confirmed by both our theory and experiments.

Our approach relies on evaluating the norm of gradients of the network output with respect to its inputs. At first order, adversarial vulnerability is related to gradient norms. We show that this norm is a function of input dimension only, whatever the network architecture is. The analysis is fully formal at initialization, and experiments show that the predictions remain valid throughout training with very good precision.

Obviously, this approach assumes that the classifier and loss are differentiable. So arguably it is unclear whether it can explain the vulnerability of networks with obfuscated or masked gradients. Still, Athalye et al. (2018) recently showed that masked gradients only give a false sense of security: by reconstructing gradient approximations (using differentiable nets!), the authors circumvented all state-of-the-art masked-gradient defenses. This suggests that explaining the vulnerability of differentiable nets is crucial, even for non-differentiable nets.

Although adversarial vulnerability was known to increase with gradient norms, the exact relation between the two,

[1]Empirical Inference Department, Max Planck Institute for Intelligent Systems, Tübingen, Germany [2]Facebook AI Research, Paris/New York. Correspondence to: Carl-Johann Simon-Gabriel <cjsimon@tue.mpg.de>.

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

and the approximations made, are seldom explained, let alone tested empirically. Section 2 therefore starts with a detailed discussion of the relationship between adversarial vulnerability and gradients of the loss. Precise definitions help with sorting out all approximations used. We also revisit and formally link several old and recent defenses, such as double-backpropagation (Drucker & LeCun, 1991) and FGSM (Goodfellow et al., 2015). Section 3 proceeds with our main theoretical results on the dimension dependence of adversarial damage. Section 4 tests our predictions empirically, as well as the validity of all approximations.

Our contribution can be summarized as follows.

- We show an empirical one-to-one relationship between average gradient norms and adversarial vulnerability. This confirms that an essential part of adversarial vulnerability arises from first-order phenomena.
- We formally prove that, at initialization, the first-order vulnerability of common neural networks increases as $\sqrt{d}$ with input dimension d. Surprisingly, this is almost independent of the architecture. Almost all current architectures are hence, by design, vulnerable at initialization.
- We empirically show that this dimension dependence persists after both usual and robust (PGD) training, but gets dampened and eventually vanishes with higher regularization. Our experiments suggest that PGD-regularization effectively recovers dimension independent accuracy-vulnerability trade-offs.
- We observe that further training after the training loss has reached its minimum can provide improved test accuracy, but severely damages the network's robustness. The last few accuracy points require a considerable increase of network gradients.
- We notice a striking discrepancy between the gradient norms (and therefore the vulnerability) on the training and test sets respectively. It suggests that gradient properties do not generalize well and that, outside the training set, networks may tend to revert to initialization-like gradient properties.

Overall, our results show that, without strong regularization, the gradients and vulnerability of current networks naturally tend to grow with input dimension. This suggests that current networks have too many degrees of 'gradient-freedom'. Gradient regularization can counter-balance this to some extent, but on the long run, our networks may benefit from incorporating more data-specific knowledge. The independence of our results on the network architecture (within the range of currently common architectures) suggests that doing so would require new network modules.

Related Literature Goodfellow et al. (2015) already noticed the dimension dependence of adversarial vulnerability. As opposed to Amsaleg et al. (2017); Gilmer et al. (2018); Shafahi et al. (2019), their (and our) explanation of the dimension dependence is data-independent. Incidentally, they also link adversarial vulnerability to loss gradients and use it to derive the FGSM adversarial augmentation defense (see Section 2). Ross & Doshi-Velez (2018) propose to robustify networks using the old double-backpropagation, but make no connection to FGSM and adversarial augmentation (see our Prop.3). Lyu et al. (2015) discuss and use the connection between gradient-penalties and adversarial augmentation, but surprisingly never empirically compare both, which we do in Section 4.1. This experiment is crucial to confirm the validity of the first-order approximation made in (2) to link adversarial damage and loss-gradients. Hein & Andriushchenko (2017) derived yet another gradient-based penalty –the *cross-Lipschitz*-penalty– by considering and proving formal guarantees on adversarial vulnerability (see App.D). Penalizing network-gradients is also at the heart of contractive auto-encoders as proposed by Rifai et al. (2011), where it is used to regularize the encoder-features. A gradient regularization of the loss of generative models also appears in Proposition 6 of Ollivier (2014), where it stems from a code-length bound on the data (minimum description length). For further references on adversarial attacks and defenses, see e.g. Yuan et al. (2017).

2. From Adversarial Examples to Large Gradients

Suppose that a given classifier φ classifies an image $\boldsymbol{x}$ as being in category $\varphi(\boldsymbol{x})$. An adversarial image is a small modification of $\boldsymbol{x}$, barely noticeable to the human eye, that suffices to fool the classifier into predicting a class different from $\varphi(\boldsymbol{x})$. It is a *small* perturbation of the inputs, that creates a *large* variation of outputs. Adversarial examples thus seem inherently related to large gradients of the network. A connection that we will now clarify. Note that visible adversarial examples sometimes appear in the literature, but we deliberately focus on imperceptible ones.

Adversarial vulnerability and adversarial damage. In practice, an adversarial image is constructed by adding a perturbation $\boldsymbol{\delta}$ to the original image $\boldsymbol{x}$ such that $\|\boldsymbol{\delta}\| \leq \epsilon$ for some (small) number ϵ and a given norm $\|\cdot\|$ over the input space. We call the perturbed input $\boldsymbol{x} + \boldsymbol{\delta}$ an ϵ-sized $\|\cdot\|$-attack and say that the attack was successful when $\varphi(\boldsymbol{x} + \boldsymbol{\delta}) \neq \varphi(\boldsymbol{x})$. This motivates

Definition 1. Given a distribution P over the input-space, we call *adversarial vulnerability* of a classifier φ to an ϵ-sized $\|\cdot\|$-attack the probability that there exists a perturbation $\boldsymbol{\delta}$ of $\boldsymbol{x}$ such that

$$\|\boldsymbol{\delta}\| \leq \epsilon \quad \text{and} \quad \varphi(\boldsymbol{x}) \neq \varphi(\boldsymbol{x}+\boldsymbol{\delta})\,. \tag{1}$$

We call the average increase-after-attack $\mathbb{E}_{\boldsymbol{x}\sim P}[\Delta\mathcal{L}]$ of a loss $\mathcal{L}$ the *adversarial ($\mathcal{L}$-) damage* (of the classifier φ to an ϵ-sized $\|\cdot\|$-attack).

When $\mathcal{L}$ is the 0-1-loss $\mathcal{L}_{0/1}$, adversarial damage is the accuracy-drop after attack. The 0-1-loss damage is always smaller than adversarial vulnerability, because vulnerability counts all class-changes of $\varphi(\boldsymbol{x})$, whereas some of them may be neutral to adversarial damage (e.g. a change between two wrong classes). The $\mathcal{L}_{0/1}$-adversarial damage thus lower bounds adversarial vulnerability. Both are even equal when the classifier is perfect (before attack), because then every change of label introduces an error. It is hence tempting to evaluate adversarial vulnerability with $\mathcal{L}_{0/1}$-adversarial damage.

From $\Delta\mathcal{L}_{0/1}$ to $\Delta\mathcal{L}$ and to $\partial_{\boldsymbol{x}}\mathcal{L}$. In practice however, we do not train our classifiers with the non-differentiable 0-1-loss but use a smoother surrogate loss $\mathcal{L}$, such as the cross-entropy loss. For similar reasons, we will now investigate the adversarial damage $\mathbb{E}_{\boldsymbol{x}}[\Delta\mathcal{L}(\boldsymbol{x},c)]$ with loss $\mathcal{L}$ rather than $\mathcal{L}_{0/1}$. Like for Goodfellow et al. (2015); Lyu et al. (2015); Sinha et al. (2018) and many others, a classifier φ will hence be robust if, on average over $\boldsymbol{x}$, a small adversarial perturbation $\boldsymbol{\delta}$ of $\boldsymbol{x}$ creates only a small variation $\delta\mathcal{L}$ of the loss. Now, if $\|\boldsymbol{\delta}\| \leq \epsilon$, then a first order Taylor expansion in ϵ shows that

$$\begin{aligned}\delta\mathcal{L} &= \max_{\boldsymbol{\delta}\,:\,\|\boldsymbol{\delta}\|\leq\epsilon} |\mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta},c)-\mathcal{L}(\boldsymbol{x},c)| \\ &\approx \max_{\boldsymbol{\delta}\,:\,\|\boldsymbol{\delta}\|\leq\epsilon} |\partial_{\boldsymbol{x}}\mathcal{L}\cdot\boldsymbol{\delta}| = \epsilon\,|||\partial_{\boldsymbol{x}}\mathcal{L}|||,\end{aligned} \tag{2}$$

where $\partial_{\boldsymbol{x}}\mathcal{L}$ denotes the gradient of $\mathcal{L}$ with respect to $\boldsymbol{x}$, and where the last equality stems from the definition of the dual norm $|||\cdot|||$ of $\|\cdot\|$. Now two remarks. First: the dual norm only kicks in because we let the input noise $\boldsymbol{\delta}$ optimally adjust to the coordinates of $\partial_{\boldsymbol{x}}\mathcal{L}$ within its ϵ-constraint. This is the brand mark of *adversarial* noise: the different coordinates add up, instead of statistically canceling each other out as they would with random noise. For example, if we impose that $\|\boldsymbol{\delta}\|_2 \leq \epsilon$, then $\boldsymbol{\delta}$ will strictly align with $\partial_{\boldsymbol{x}}\mathcal{L}$. If instead $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$, then $\boldsymbol{\delta}$ will align with the sign of the coordinates of $\partial_{\boldsymbol{x}}\mathcal{L}$. Second remark: while the Taylor expansion in (2) becomes exact for infinitesimal perturbations, for finite ones it may actually be dominated by higher-order terms. Our experiments (Figures 4 & 1) however strongly suggest that in practice the first order term dominates the others. Now, remembering that the dual norm of an ℓ_p-norm is the corresponding ℓ_q-norm, and summarizing, we have proven

Lemma 2. *At first order approximation in ϵ, an ϵ-sized adversarial attack generated with norm $\|\cdot\|$ increases the loss $\mathcal{L}$ at point $\boldsymbol{x}$ by $\epsilon\,|||\partial_{\boldsymbol{x}}\mathcal{L}|||$, where $|||\cdot|||$ is the dual norm of $\|\cdot\|$. In particular, an ϵ-sized ℓ_p-attack increases the loss by $\epsilon\,\|\partial_{\boldsymbol{x}}\mathcal{L}\|_q$ where $1 \leq p \leq \infty$ and $\frac{1}{p}+\frac{1}{q}=1$.*

Although the lemma is valid at first order only, it proves that *at least* this kind of first-order vulnerability is present. Moreover, we will see that the first-order predictions closely match the experiments, and that simple gradient regularization helps protecting even against iterative (non-first-order) attack methods (Figure 4).

Calibrating the threshold ϵ to the attack-norm $\|\cdot\|$. Lemma 2 shows that adversarial vulnerability depends on three main factors: (i) $\|\cdot\|$, the norm chosen for the attack (ii) ϵ, the size of the attack, and (iii) $\mathbb{E}_{\boldsymbol{x}}|||\partial_{\boldsymbol{x}}\mathcal{L}|||$, the expected *dual* norm of $\partial_{\boldsymbol{x}}\mathcal{L}$. We could see Point (i) as a measure of our sensibility to image perturbations, (ii) as our sensibility threshold, and (iii) as the classifier's expected marginal sensibility to a unit perturbation. $\mathbb{E}_{\boldsymbol{x}}|||\partial_{\boldsymbol{x}}\mathcal{L}|||$ hence intuitively captures the discrepancy between our perception (as modeled by $\|\cdot\|$) and the classifier's perception for an input-perturbation of small size ϵ. Of course, this viewpoint supposes that we actually found a norm $\|\cdot\|$ (or more generally a metric) that faithfully reflects human perception – a project in its own right, far beyond the scope of this paper. However, it is clear that the threshold ϵ that we choose should depend on the norm $\|\cdot\|$ and hence on the input-dimension d. In particular, for a given pixel-wise order of magnitude of the perturbations $\boldsymbol{\delta}$, the ℓ_p-norm of the perturbation will scale like $d^{1/p}$. This suggests to write the threshold ϵ_p used with ℓ_p-attacks as:

$$\epsilon_p = \epsilon_\infty\, d^{1/p}\,, \tag{3}$$

where ϵ_∞ denotes a dimension independent constant. In Appendix C we show that this scaling also preserves the average signal-to-noise ratio $\|\boldsymbol{x}\|_2 \,/\, \|\boldsymbol{\delta}\|_2$, both across norms and dimensions, so that ϵ_p could correspond to a constant human perception-threshold. With this in mind, the impatient reader may already jump to Section 3, which contains our main contributions: the estimation of $\mathbb{E}_x\|\partial_{\boldsymbol{x}}\mathcal{L}\|_q$ for standard feedforward nets. Meanwhile, the rest of this section shortly discusses two straightforward defenses that we will use later and that further illustrate the role of gradients.

A new old regularizer. Lemma 2 shows that the loss of the network after an $\frac{\epsilon}{2}$-sized $\|\cdot\|$-attack is

$$\mathcal{L}_{\epsilon,|||\cdot|||}(\boldsymbol{x},c) := \mathcal{L}(\boldsymbol{x},c) + \frac{\epsilon}{2}\,|||\partial_{\boldsymbol{x}}\mathcal{L}|||\,. \tag{4}$$

It is thus natural to take this loss-after-attack as a new training objective. Here we introduced a factor 2 for reasons that will become clear in a moment. Incidentally, for

$\|\cdot\| = \|\cdot\|_2$, this new loss reduces to an old regularization-scheme proposed by Drucker & LeCun (1991) called *double-backpropagation*. At the time, the authors argued that slightly decreasing a function's or a classifier's sensitivity to input perturbations should improve generalization. In a sense, this is exactly our motivation when defending against adversarial examples. It is thus not surprising to end up with the same regularization term. Note that our reasoning only shows that training with one specific norm $|||\cdot|||$ in (4) helps to protect against adversarial examples generated from $\|\cdot\|$. A priori, we do not know what will happen for attacks generated with other norms; but our experiments suggest that training with one norm also protects against other attacks (see Figure 1 and Section 4.1).

Link to adversarially augmented training. In (1), ϵ designates an attack-size threshold, while in (4), it is a regularization-strength. Rather than a notation conflict, this reflects an intrinsic duality between two complementary interpretations of ϵ, which we now investigate further. Suppose that, instead of using the loss-after-attack, we augment our training set with ϵ-sized $\|\cdot\|$-attacks $\boldsymbol{x} + \boldsymbol{\delta}$, where for each training point $\boldsymbol{x}$, the perturbation $\boldsymbol{\delta}$ is generated on the fly to locally maximize the loss-increase. Then we are effectively training with

$$\tilde{\mathcal{L}}_{\epsilon,\|\cdot\|}(\boldsymbol{x}, c) := \frac{1}{2}(\mathcal{L}(\boldsymbol{x}, c) + \mathcal{L}(\boldsymbol{x} + \epsilon\,\boldsymbol{\delta}, c))\,, \qquad (5)$$

where by construction $\boldsymbol{\delta}$ satisfies (2). We will refer to this technique as *adversarially augmented training*. It was first introduced by Goodfellow et al. (2015) with $\|\cdot\| = \|\cdot\|_\infty$ under the name of FGSM[1]-augmented training. Using the first order Taylor expansion in ϵ of (2), this 'old-plus-post-attack' loss of (5) simply reduces to our loss-after-attack, which proves

Proposition 3. *Up to first-order approximations in ϵ, $\tilde{\mathcal{L}}_{\epsilon,\|\cdot\|} = \mathcal{L}_{\epsilon,|||\cdot|||}$. Said differently, for small enough ϵ, adversarially augmented training with ϵ-sized $\|\cdot\|$-attacks amounts to penalizing the* dual *norm $|||\cdot|||$ of $\partial_{\boldsymbol{x}}\mathcal{L}$ with weight $\epsilon/2$. In particular, double-backpropagation corresponds to training with ℓ_2-attacks, while FGSM-augmented training corresponds to an ℓ_1-penalty on $\partial_{\boldsymbol{x}}\mathcal{L}$.*

This correspondence between training with perturbations and using a regularizer can be compared to Tikhonov regularization: Tikhonov regularization amounts to training with *random* noise Bishop (1995), while training with *adversarial* noise amounts to penalizing $\partial_{\boldsymbol{x}}\mathcal{L}$. Section 4.1 verifies the correspondence between adversarial augmentation and gradient regularization empirically, which also strongly suggests the empirical validity of the first-order Taylor expansion in (2).

[1] *F*ast *G*radient *S*ign *M*ethod

3. Estimating $\|\partial_{\boldsymbol{x}}\mathcal{L}\|_q$ to Evaluate Adversarial Vulnerability

In this section, we evaluate the size of $\|\partial_{\boldsymbol{x}}\mathcal{L}\|_q$ for a very wide class of standard network architectures. We show that, inside this class, the gradient-norms are independent of the network topology and increase with input dimension. We start with an intuitive explanation of these insights (Sec 3.1) before moving to our formal statements (Sec 3.2).

3.1. Core Idea: One Neuron with Many Inputs

This section is for intuition only: no assumption made here is used later. We start by showing how changing q affects the size of $\|\partial_{\boldsymbol{x}}\mathcal{L}\|_q$. Suppose for a moment that the coordinates of $\partial_{\boldsymbol{x}}\mathcal{L}$ have typical magnitude $|\partial_{\boldsymbol{x}}\mathcal{L}|$. Then $\|\partial_{\boldsymbol{x}}\mathcal{L}\|_q$ scales like $d^{1/q}|\partial_{\boldsymbol{x}}\mathcal{L}|$. Consequently

$$\epsilon_p \,\|\partial_{\boldsymbol{x}}\mathcal{L}\|_q \;\propto\; \epsilon_p \, d^{1/q}\, |\partial_{\boldsymbol{x}}\mathcal{L}| \;\propto\; d\, |\partial_{\boldsymbol{x}}\mathcal{L}|\,. \qquad (6)$$

This equation carries two important messages. First, we see how $\|\partial_{\boldsymbol{x}}\mathcal{L}\|_q$ depends on d and q. The dependence seems highest for $q = 1$. But once we account for the varying perceptibility threshold $\epsilon_p \propto d^{1/p}$, we see that adversarial vulnerability scales like $d \cdot |\partial_{\boldsymbol{x}}\mathcal{L}|$, whatever ℓ_p-norm we use. Second, (6) shows that to be robust against any type of ℓ_p-attack at any input-dimension d, the average absolute value of the coefficients of $\partial_{\boldsymbol{x}}\mathcal{L}$ must grow slower than $1/d$. Now, here is the catch, which brings us to our core insight.

In order to preserve the activation variance of the neurons from layer to layer, the neural weights are usually initialized with a variance that is inversely proportional to the number of inputs per neuron. Imagine for a moment that the network consisted only of one output neuron o linearly connected to all input pixels. For the purpose of this example, we assimilate o and $\mathcal{L}$. Because we initialize the weights with a variance of $1/d$, their average absolute value $|\partial_x o| \equiv |\partial_{\boldsymbol{x}}\mathcal{L}|$ grows like $1/\sqrt{d}$, rather than the required $1/d$. By (6), the adversarial vulnerability $\epsilon\,\|\partial_x o\|_q \equiv \epsilon\,\|\partial_{\boldsymbol{x}}\mathcal{L}\|_q$ therefore increases like $d/\sqrt{d} = \sqrt{d}$.

This toy example shows that the standard initialization scheme, which preserves the variance from layer to layer, causes the average coordinate-size $|\partial_{\boldsymbol{x}}\mathcal{L}|$ to grow like $1/\sqrt{d}$ instead of $1/d$. When an ℓ_∞-attack tweaks its ϵ-sized input-perturbations to align with the coordinate-signs of $\partial_{\boldsymbol{x}}\mathcal{L}$, all coordinates of $\partial_{\boldsymbol{x}}\mathcal{L}$ add up in absolute value, resulting in an output-perturbation that scales like $\epsilon\sqrt{d}$ and leaves the network increasingly vulnerable with growing input-dimension.

3.2. Formal Statements for Deep Networks

Our next theorems formalize and generalize the previous toy example to a very wide class of feedforward nets with ReLU activation functions. For illustration purposes, we start with fully connected nets before proceeding with the broader class, which includes any succession of (possibly strided) convolutional layers. In essence, the proofs iterate our insight on one layer over a sequence of layers. They all rely on the following set ($\mathcal{H}$) of hypotheses:

H1 Non-input neurons are followed by a ReLU killing half of its inputs, independently of the weights.
H2 Neurons are partitioned into layers, meaning groups that each path traverses at most once.
H3 All weights have 0 expectation and variance $2/(\text{in-degree})$ ('He-initialization').
H4 The weights from different layers are independent.
H5 Two distinct weights w, w' from a same node satisfy $\mathbb{E}[w\, w'] = 0$.

If we follow common practice and initialize our nets as proposed by He et al. (2015), then H3-H5 are satisfied at initialization by design, while H1 is usually a very good approximation (Balduzzi et al., 2017). Note that such i.i.d. weight assumptions have been widely used to analyze neural nets and are at the heart of very influential and successful prior work (e.g., equivalence between neural nets and Gaussian processes as pioneered by Neal 1996). Nevertheless, they do not hold after training. That is why all our statements in this section are to be understood as *orders of magnitudes* that are very well satisfied at initialization both in theory and practice, and that we will confirm experimentally for trained networks in Section 4. Said differently, while our theorems rely on the statistics of neural nets at initialization, our experiments confirm their conclusions after training.

Theorem 4 (**Vulnerability of Fully Connected Nets**). *Consider a succession of fully connected layers with ReLU activations which takes inputs $\boldsymbol{x}$ of dimension d, satisfies assumptions ($\mathcal{H}$), and outputs logits $f_k(\boldsymbol{x})$ that get fed to a final cross-entropy-loss layer $\mathcal{L}$. Then the coordinates of $\boldsymbol{\partial_x} f_k$ grow like $1/\sqrt{d}$, and*

$$\left\|\boldsymbol{\partial_x}\mathcal{L}\right\|_q \propto d^{\frac{1}{q}-\frac{1}{2}} \quad \text{and} \quad \epsilon_p \left\|\boldsymbol{\partial_x}\mathcal{L}\right\|_q \propto \sqrt{d}\,. \qquad (7)$$

These networks are thus increasingly vulnerable to ℓ_p-attacks with growing input-dimension.

Theorem 4 is a special case of the next theorem, which will show that the previous conclusions are essentially independent of the network-topology. We will use the following symmetry assumption on the neural connections. For a given path $\boldsymbol{p}$, let the *path-degree* $d_{\boldsymbol{p}}$ be the multiset of encountered in-degrees along path $\boldsymbol{p}$. For a fully connected network, this is the unordered sequence of layer-sizes preceding the last path-node, including the input-layer. Now consider the multiset $\{d_{\boldsymbol{p}}\}_{\boldsymbol{p}\in\mathcal{P}(x,o)}$ of all path-degrees when $\boldsymbol{p}$ varies among all paths from input x to output o. The symmetry assumption (relatively to o) is

($\mathcal{S}$) All input nodes x have the same multiset $\{d_{\boldsymbol{p}}\}_{\boldsymbol{p}\in\mathcal{P}(x,o)}$ of path-degrees from x to o.

Intuitively, this means that the statistics of degrees encountered along paths to the output are the same for all input nodes. This symmetry assumption is exactly satisfied by fully connected nets, almost satisfied by CNNs (up to boundary effects, which can be alleviated via periodic or mirror padding) and exactly satisfied by strided layers, if the layer-size is a multiple of the stride.

Theorem 5 (**Vulnerability of Feedforward Nets**). *Consider any feedforward network with linear connections and ReLU activation functions. Assume the net satisfies assumptions ($\mathcal{H}$) and outputs logits $f_k(\boldsymbol{x})$ that get fed to the cross-entropy-loss $\mathcal{L}$. Then $\left\|\boldsymbol{\partial_x} f_k\right\|_2$ is independent of the input dimension d and $\epsilon_2 \left\|\boldsymbol{\partial_x}\mathcal{L}\right\|_2 \propto \sqrt{d}$. Moreover, if the net satisfies the symmetry assumption ($\mathcal{S}$), then $|\partial_x f_k| \propto 1/\sqrt{d}$ and (7) still holds: $\left\|\boldsymbol{\partial_x}\mathcal{L}\right\|_q \propto d^{\frac{1}{q}-\frac{1}{2}}$ and $\epsilon_p \left\|\boldsymbol{\partial_x}\mathcal{L}\right\|_q \propto \sqrt{d}$.*

Theorems 4 and 5 are proven in Appendix A. The main proof idea is that in the gradient norm computation, the He-initialization exactly compensates the combinatorics of the number of paths in the network, so that this norm becomes independent of the network topology. In particular, we get

Corollary 6 (**Vulnerability of CNNs**). *In any succession of convolution and dense layers, strided or not, with ReLU activations, that satisfies assumptions ($\mathcal{H}$) and outputs logits that get fed to the cross-entropy-loss $\mathcal{L}$, the gradient of the logit-coordinates scale like $1/\sqrt{d}$ and (7) is satisfied. It is hence increasingly vulnerable with growing input-resolution to attacks generated with any ℓ_p-norm.*

Remarks.

- Appendix B shows that the network gradients are dampened when replacing strided layers by average poolings, essentially because average-pooling weights do not follow the He-init assumption H3.
- Although the principles of our analysis naturally extend to residual nets, they are not yet covered by our theorems (residual connections do not satisfy H3).
- Current weight initializations (He-, Glorot-, Xavier-) are chosen to preserve the variance from layer to layer, which constrains their scaling to $1/\sqrt{\text{in-degree}}$. This scaling, we show, is incompatible with small gradients. But decreasing gradients simply by reducing the initial weights would kill the output signal and make training impossible for deep nets (He et al., 2015, Sec 2.2). Also note that rescaling all weights by a constant does not change the classification decisions, but it affects cross-entropy and therefore adversarial damage.

4. Empirical Results

Section 4.1 empirically verifies the validity of the first-order Taylor approximation made in (2) and the correspondence between gradient regularization and adversarial augmentation (Fig.1). Section 4.2 analyzes the di-

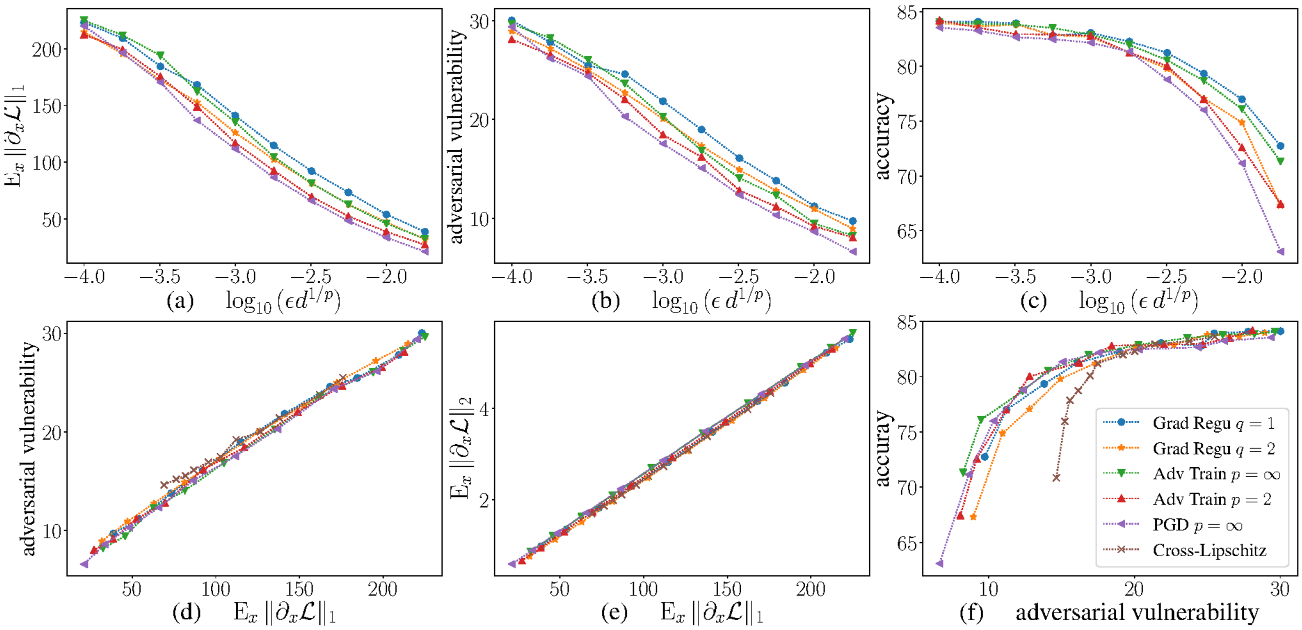

Figure 1. Average norm $\mathbb{E}_{\boldsymbol{x}}\|\partial_{\boldsymbol{x}}\mathcal{L}\|$ of the loss-gradients, adversarial vulnerability and accuracy (before attack) of various networks trained with different adversarial regularization methods and regularization strengths ϵ. Each point represents a trained network, and each curve a training-method. *Upper row*: A priori, the regularization-strengths ϵ have different meanings for each method. The near superposition of all upper-row curves illustrates (i) the duality between adversarial augmentation and gradient regularization (Prop.3) and (ii) confirms the rescaling of ϵ proposed in (3) and (iii) supports the validity of the first-order Taylor expansion (2). (d): near functional relation between adversarial vulnerability and average loss-gradient norms. (e): the near-perfect linear relation between the $\mathbb{E}\|\partial_{\boldsymbol{x}}\mathcal{L}\|_1$ and $\mathbb{E}\|\partial_{\boldsymbol{x}}\mathcal{L}\|_2$ suggests that protecting against a given attack-norm also protects against others. (f): Merging 1b and 1c shows that all adversarial augmentation and gradient regularization methods achieve similar accuracy-vulnerability trade-offs.

mension dependence of the average gradient-norms and adversarial vulnerability after usual and robust training. Section 4.1 uses an attack-threshold $\epsilon_\infty = 0.5\%$ of the pixel-range (invisible to humans), with PGD-attacks from the Foolbox-package (Rauber et al., 2017). Section 4.2 uses self-coded PGD-attacks with random start with $\epsilon_\infty = 0.08\%$. As a safety-check, other attacks were tested as well (see App.4.1 & Fig.4), but results remained essentially unchanged. Note that the ϵ_∞-*thresholds* should not be confused with the *regularization-strengths* ϵ appearing in (4) and (5), which will be varied. The datasets were normalized ($\sigma \approx .2$). All regularization-values ϵ are reported in these normalized units (i.e. multiply by .2 to compare with 0-1 pixel values). Code available at `https://github.com/facebookresearch/AdversarialAndDimensionality`.

4.1. First-Order Approximation, Gradient Penalty, Adversarial Augmentation

We train several CNNs with same architecture to classify CIFAR-10 images (Krizhevsky, 2009). For each net, we use a specific training method with a specific regularization value ϵ. The training methods used were ℓ_1- and ℓ_2-penalization of $\partial_{\boldsymbol{x}}\mathcal{L}$ (Eq. 4), adversarial augmentation with ℓ_∞- and ℓ_2- attacks (Eq. 5), projected gradient descent (PGD) with randomized starts (7 steps per attack with step-size $= .2\,\epsilon_\infty$; see Madry et al. 2018) and the cross-Lipschitz regularizer (Eq. 18 in Appendix D). For this experiment, all networks have 6 'strided convolution $\rightarrow$ batchnorm $\rightarrow$ ReLU' layers with strides [1, 2, 2, 2, 2, 2] respectively and 64 output-channels each, followed by a final fully-connected linear layer. Results are summarized in Figure 1. Each curve represents one training method. Note that our goal here is not to advocate one defense over another, but rather to check the validity of the Taylor expansion, and empirically verify that first order terms (i.e., gradients) suffice to explain much of the observed adversarial vulnerability.

Confirming first order expansion and large first-order vulnerability. The following observations support the validity of the first order Taylor expansion in (2) and suggest that it is a crucial component of adversarial vulnerability: (i) the efficiency of the first-order defense against iterative (non-first-order) attacks (Fig.1&4a); (ii) the striking similarity between the PGD curves (adversarial augmentation with *iterative* attacks) and the other adversarial training training curves (*one-step* attacks/defenses); (iii) the functional-like dependence between any approximation of adversarial vulnerability and $\mathbb{E}_{\boldsymbol{x}}\|\partial_{\boldsymbol{x}}\mathcal{L}\|_1$ (Fig.4b), and its independence on the training method (Fig.1d). (iv) the excellent correspondence between the gradient regularization and adversarial augmentation curves (see next paragraph). Said differently, adversarial examples seem indeed to be primarily caused by large gradients of the classifier as captured via the induced loss.

Gradient regularization matches adversarial augmentation (Prop.3). The upper row of Figure 1 plots $\mathbb{E}_{\boldsymbol{x}}\|\boldsymbol{\partial}_{\boldsymbol{x}}\mathcal{L}_1\|$, adversarial vulnerability and accuracy as a function of $\epsilon\, d^{1/p}$. The excellent match between the adversarial augmentation curve with $p = \infty$ ($p = 2$) and its gradient regularization dual counterpart with $q = 1$ (resp. $q = 2$) illustrates the duality between ϵ as a threshold for adversarially augmented training and as a regularization constant in the regularized loss (Proposition 3). It also supports the validity of the first-order Taylor expansion in (2).

Confirming correspondence of norm-dependent thresholds (Eq.3). Still on the upper row, the curves for $p = \infty, q = 1$ have no reason to match those for $p = q = 2$ when plotted against ϵ, because the ϵ-threshold is relative to a specific attack-norm. However, (3) suggested that the rescaled thresholds $\epsilon d^{1/p}$ may approximately correspond to a same 'threshold-unit' across ℓ_p-norms and across dimension. This is well confirmed by the upper row plots: by rescaling the x-axis, the $p = q = 2$ and $q = 1$, $p = \infty$ curves get almost super-imposed.

Accuracy-vulnerability trade-off: confirming large first-order component of vulnerability. Merging Figures 1b and 1c by taking out ϵ, Figure 1f shows that all gradient regularization and adversarial augmentation methods, *including iterative ones (PGD)*, yield equivalent accuracy-vulnerability trade-offs. This suggest that adversarial vulnerability is largely first-order. For higher penalization values, these trade-offs appear to be much better than those given by cross Lipschitz regularization.

The regularization-norm does not matter. We were surprised to see that on Figures 1d and 1f, the $\mathcal{L}_{\epsilon,q}$ curves are almost identical for $q = 1$ and 2. This indicates that both norms can be used interchangeably in (4) (modulo proper rescaling of ϵ via (3)), and suggests that protecting against a specific attack-norm also protects against others. (6) may provide an explanation: if the coordinates of $\boldsymbol{\partial}_{\boldsymbol{x}}\mathcal{L}$ behave like centered, uncorrelated variables with equal variance –which would follow from assumptions ($\mathcal{H}$) –, then the ℓ_1- and ℓ_2-norms of $\boldsymbol{\partial}_{\boldsymbol{x}}\mathcal{L}$ are simply proportional. Plotting $\mathbb{E}_x\|\boldsymbol{\partial}_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x})\|_2$ against $\mathbb{E}_x\|\boldsymbol{\partial}_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x})\|_1$ in Figure 1e confirms this explanation. The slope is independent of the training method. (But Fig 7e shows that it is not independent of the input-dimension.) Therefore, penalizing $\|\boldsymbol{\partial}_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x})\|_1$ during training will not only decrease $\mathbb{E}_{\boldsymbol{x}}\|\boldsymbol{\partial}_{\boldsymbol{x}}\mathcal{L}\|_1$ (as shown in Figure 1a), but also drive down $\mathbb{E}_{\boldsymbol{x}}\|\boldsymbol{\partial}_{\boldsymbol{x}}\mathcal{L}\|_2$ and vice-versa.

4.2. Vulnerability's Dependence on Input Dimension

Theorems 4-5 and Corollary 6 predict a linear growth of the average ℓ_1-norm of $\boldsymbol{\partial}_{\boldsymbol{x}}\mathcal{L}$ with the square root of the input dimension d, and therefore an increased adversarial vulnerability (Lemma 2). To test these predictions, we compare the vulnerability of different PGD-regularized networks when varying the input-dimension. To do so, we resize the original 3x32x32 CIFAR-10 images to 32, 64, 128 and 256 pixels per edge by copying adjacent pixels, and train one CNN for each input-size and regularization strength ϵ. All nets had the same amount of parameters and very similar structure across input-resolutions (see Appendix G.1). All reported values were computed over the last 20 training epochs on the same held-out test-set.

Gradients and vulnerability increase with $\sqrt{d}$. Figures 2a &2b summarize the resulting dimension dependence of gradient-norms and adversarial vulnerability. The dashed-lines follow the medians of the 20 last epochs and the errorbars show their 10^{th} and 90^{th} quantiles. Similar to the predictions of our theorems at initialization, we see that, even after training, $\mathbb{E}_{\boldsymbol{x}}\left[\|\boldsymbol{\partial}_{\boldsymbol{x}}\mathcal{L}\|_1\right]$ grows linearly with $\sqrt{d}$ which yields higher adversarial vulnerability. However, increasing the regularization decreases the slope of this dimension dependence until, eventually, the dependence breaks.

Accuracies are dimension independent. Figure 2c plots accuracy versus regularization strength, with errorbars summarizing the 20 last training epochs.[2] The four curves correspond to the four different input dimensions. They overlap, which confirms that contrary to vulnerability, the accuracies are dimension independent; and that the ℓ_∞-attack thresholds are essentially dimension independent.

PGD effectively recovers original input dimension. Figure 2d plots the accuracy-vulnerability trade-offs achieved by the previous nets over their 20 last training epochs, with a smoothing spline fitted for each input dimension (scipy's UnivariateSpline with s=200). Higher dimensions have a longer plateau to the right, because without regularization, vulnerability increases with input dimension. The curves overlap when moving to the left, meaning that the accuracy-vulnerability trade-offs achieved by PGD are essentially *independent of the actual input dimension.*

PGD training outperforms down-sampling. On artificially upsampled CIFAR-10 images, PGD regularization acts as if it first reduced the images back to their original size before classifying them. Can PGD outperform this strategy when the original image is really high resolution? To test this, we create a 12-class 'Mini-ImageNet' dataset with approximately 80,000 images of size 3x256x256 by merg-

[2]Fig.2c &2c are similar to Figures 1c &1f, but with one curve per input-dimension instead of one per regularization method. See Appendix G for full equivalent of Figure 1.

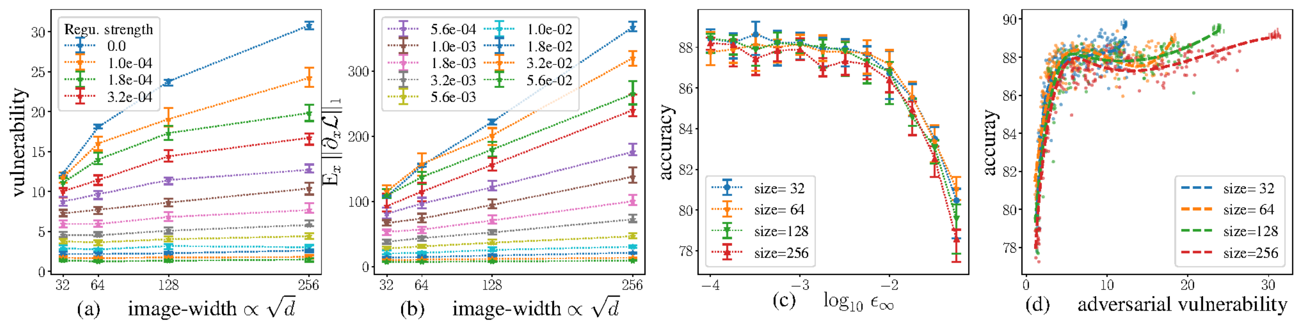

Figure 2. Input-dimension dependence of adversarial vulnerability, gradient norms and accuracy measured on up-sampled CIFAR-10 images. (b) Similar to our theorems' prediction at initialization, average gradient norms increase like $\sqrt{d}$ yielding (a) higher vulnerability. Larger PGD-regularization during training can significantly dampen this dimension dependence with (c) almost no harm to accuracy at first (long plateau on 2c). Accuracy starts getting damaged when the dimension dependence is nearly broken ($\epsilon_\infty \approx .0005$). (d) Whatever the input-dimension, PGD-training achieves similar accuracy-vulnerability trade-offs. (c) & (d) suggest that PGD-training effectively recovers the original image size, 3x32x32.

ing similar ImageNet classes and center-cropping/resizing as needed. We then do the same experiment as with up-sampled CIFAR-10, but using down-sampling instead of up-sampling (Appendix H, Fig. 13). While the dependence of vulnerability to the dimension stays essentially unchanged, PGD training now achieves much better accuracy-vulnerability trade-offs with the original high-dimensional images than with their down-sampled versions.

Insights from figures in Appendix G. Appendix G reproduces many additional figures on this section's experiments. They yield additional insights which we summarize here.

Non-equivalence of loss- and accuracy-damage. Figure 8a&c show that the test-error continues to decrease all over training, while the cross-entropy increases on the test set from epoch ≈ 40 and on. This aligns with the observations and explanations of Soudry et al. (2018). But it also shows that one must be careful when substituting their differentials, loss- and accuracy-damage. (See also Fig.9b.)

Early stopping dampens vulnerability. Fig.8 shows that adversarial damage and vulnerability closely follow the evolution of cross-entropy. Since cross-entropy overfits, early stopping effectively acts as a defense. See Fig.10.

Gradient norms do not generalize well. Figure 12 reveals a strong discrepancy between the average gradient norms on the test and the training data. This discrepancy increases over training (gradient norms decrease on the training data but increase on the test set), and with the input dimension, as $\sqrt{d}$. This dimension dependence might suggest that, outside the training points, the networks tend to recover initial gradient properties. Our observations confirm Schmidt et al.'s (2018) recent finding that PGD-regularization has a hard time generalizing to the test-set. They claim that better generalization requires more data. Alternatively, we could try to rethink our network modules to adapt it to the data, e.g. by decreasing their degrees of 'gradient-freedom'. Evaluating the gradient-sizes at initialization may help doing so.

5. Conclusion

For differentiable classifiers and losses, we showed that adversarial vulnerability increases with the gradients $\partial_x \mathcal{L}$ of the loss. All approximations made are fully specified, and validated by the near-perfect functional relationship between gradient norms and vulnerability (Fig.1d). We evaluated the size of $\|\partial_x \mathcal{L}\|_q$ and showed that, at initialization, many current feedforward nets (convolutional or fully connected) are increasingly vulnerable to ℓ_p-attacks with growing input dimension (image size), independently of their architecture. Our experiments confirm this dimension dependence after usual training, but PGD-regularization dampens it and can effectively counter-balance the effect of artificial input dimension augmentation. Nevertheless, regularizing beyond a certain point yields a rapid decrease in accuracy, even though at that point many adversarial examples are still visually undetectable for humans. Moreover, the gradient norms remain much higher on test than on training examples. This suggests that even with PGD robustification, there are still significant statistical differences between the network's behavior on the training and test sets. Given the generality of our results in terms of architectures, this can perhaps be alleviated only via tailored architectural constraints on the gradients of the network. Based on these theoretical insights, we hypothesize that tweaks on the architecture may not be sufficient, and coping with the phenomenon of adversarial examples will require genuinely new ideas.

Acknowledgements

We thank Martín Arjovsky, Ilya Tolstikhin and Diego Fioravanti for helpful discussions.

References

Amsaleg, L., Bailey, J. E., Barbe, D., Erfani, S., Houle, M. E., Nguyen, V., and Radovanovic, M. The vulnerability of learning to adversarial perturbation increases with intrinsic dimensionality. In *IEEE Workshop on Information Forensics and Security*, 2017.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Balduzzi, D., McWilliams, B., and Butler-Yeoman, T. Neural taylor approximations: Convergence and exploration in rectifier networks. In *ICML*, 2017.

Bishop, C. M. Training with noise is equivalent to Tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

Drucker, H. and LeCun, Y. Double backpropagation increasing generalization performance. In *International Joint Conference on Neural Networks*, 1991.

Gilmer, J., Metz, L., Faghri, F., Schoenholz, S. S., Raghu, M., Wattenberg, M., and Goodfellow, I. Adversarial spheres. In *ICLR Workshop*, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NIPS*, 2017.

Huang, J. *Statistics of Natural Images and Models*. PhD thesis, Brown University, Providence, RI, 2000.

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

Lyu, C., Huang, K., and Liang, H.-N. A unified gradient regularization family for adversarial examples. In *ICDM*, 2015.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. DeepFool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

Neal, R. M. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, 1996.

Ollivier, Y. Auto-encoders: Reconstruction versus compression. arXiv:1403.7752, 2014.

Rauber, J., Brendel, W., and Bethge, M. Foolbox v0.8.0: A Python toolbox to benchmark the robustness of machine learning models. arXiv:1707.04131, 2017.

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. Contractive auto-encoders: Explicit invariance during feature extraction. In *ICML*, 2011.

Ross, A. S. and Doshi-Velez, F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *AAAI*, 2018.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Mądry, A. Adversarially robust generalization requires more data. arXiv:1804.11285, 2018.

Shafahi, A., Huang, W. R., Studer, C., Feizi, S., and Goldstein, T. Are adversarial examples inevitable? In *ICLR*, 2019.

Sinha, A., Namkoong, H., and Duchi, J. Certifiable distributional robustness with principled adversarial training. In *ICLR*, 2018.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *JMLR*, 19(70):1–57, 2018.

Yuan, X., He, P., Zhu, Q., Bhat, R. R., and Li, X. Adversarial examples: Attacks and defenses for deep learning. arXiv:1712.07107, 2017.