Adversarial examples and where to find them
Preprint, compiled April 24, 2020

Niklas Risse* Christina Göpfert* Jan Philip Göpfert*
Bielefeld University, Germany

Abstract

Adversarial robustness of trained models has attracted considerable attention over recent years, within and beyond the scientific community. This is not only because of a straight-forward desire to deploy reliable systems, but also because of how adversarial attacks challenge our beliefs about deep neural networks. Demanding more robust models seems to be the obvious solution – however, this requires a rigorous understanding of how one should judge adversarial robustness as a property of a given model. In this work, we analyze where adversarial examples occur, in which ways they are peculiar, and how they are processed by robust models. We use robustness curves to show that \( \ell_\infty \) threat models are surprisingly effective in improving robustness for other \( \ell_p \) norms; we introduce perturbation cost trajectories to provide a broad perspective on how robust and non-robust networks perceive adversarial perturbations as opposed to random perturbations; and we explicitly examine the scale of certain common data sets, showing that robustness thresholds must be adapted to the data set they pertain to. This allows us to provide concrete recommendations for anyone looking to train a robust model or to estimate how much robustness they should require for their operation. The code for all our experiments is available at www.github.com/niklasrisse/adversarial-examples-and-where-to-find-them.

1 Introduction

Adversarial robustness is a property that describes a model’s ability to behave correctly under small input perturbations that are crafted with the intent to mislead the model. In recent years, it has been pointed out that deep neural networks, despite their astonishing success in a wide range of prediction tasks, frequently lack this property [17, 8]. The study of adversarial robustness – with its definitions, their implications, attacks, and defenses – has subsequently attracted considerable research interest. This is due both to the practical importance of trustworthy models and the intellectual interest in the differences between decisions of machine learning models and our human perception.

From a theoretical perspective, this scientific enthusiasm has led to an analysis of different definitions of adversarial robustness, their implications in terms of sample complexity and computational hardness, and the possibility or impossibility of achieving adversarial robustness under certain distributional assumptions [7, 11]. These results on hardness pose an intellectual challenge: for many domains, a robust and efficient classifier does exist in the human brain (indeed, adversarial examples are often motivated by the differences between human and model classification). If we are not yet able to replicate this artificially, is it due to our definitions, our models, our data representation, or our optimization procedures?

From a practical perspective, we have seen the development of a wide array of defenses against adversarial attacks, most of which have quickly prompted the development of new attack methods capable of circumventing them [2, 4, 15, 19, 3]. A helpful overview of methodological foundations for defense mechanisms (and thereby indirectly of attack mechanisms) is given by [3]. Central to the development of an adversarial defense is the specification of a threat model that defines the types of perturbations which, when applied to real data points, should not change the classification behavior of the model. The challenge then lies in finding a tractable optimization procedure that satisfies the threat model, and in providing credible empirical or – ideally – theoretical evidence to show that the resulting model is indeed robust to all perturbations in the threat model. The most commonly used threat models are \( \ell_p \) ball constrained perturbations, i.e. perturbations with \( \ell_p \) norm smaller than a specified threshold. The most commonly used norms in this context are \( \ell_\infty, \ell_2 \) and \( \ell_1 \). Under this type of threat model, by definition, the study of adversarial robustness is closely tied to the distance of examples to decision boundaries. Constructing models where adversarial perturbations are rare, then involves directly or indirectly maximizing margins around decision boundaries.

One popular defense mechanism for indirectly maximizing margins is Adversarial Training with Projected Gradient Descent (AT), which Madry et al. [15] argue provides a natural security guarantee against first-order adversaries motivated by robust optimization. Wong and Kolter [20] introduce a method (\( \text{AT}^\infty \)) with provable robustness against any norm-bounded perturbation on the training data by adapting the training process to minimize an upper bound on the robust loss of the network. Croce et al. [5] and Croce and Hein [6] achieve improved robustness to \( \ell_p \) norm bounded perturbations by adding a regularization term that penalizes closeness to decision boundaries and linear regions of the model (\( \text{MMR} + \text{AT} \) and \( \text{MMR–UNIV} \)).

For this work, we explicitly decided not to focus on developing a new adversarial attack or defense. Instead, we explore and present adversarial robustness from a broader perspective, in order to increase the community’s understanding of how defense methods interact with choice of threat model, properties of the data set, and the way adversaries are processed. We start by

It should – therefore – not surprise us when algorithms, such as the Support Vector Machine, that by construction maximize margins, also lead to adversarially robust models regarding the type of margin they optimize.
describing robustness curves [9] in Section 2 as a tool to evaluate the robustness of specific models and data sets and then study how robustness depends on distance function, data distribution and training method. The main part of the paper is structured as follows:

1. In Section 3, we compare robustness curves for different models and different choices of distance functions. Our goal is to understand how the choice of model and training method impacts robustness, and to what extent robustness transfers between distance functions. For linear classifiers we show theoretically, that the shape of the robustness curve – representing the global robustness behaviour – is the same for all norms, and closeness of the norms can be controlled via sparsity of the weight vector. For non-linear models we empirically find that robustness transfers to $\ell_p$ norms not considered in the threat model, and the $\ell_\infty$ threat models frequently provide better $\ell_2$ robustness than $\ell_2$ threat models.

2. In Section 4, we analyze how adversarial perturbations are perceived by robust and non-robust networks, specifically compared to random perturbations. To gain insight, we introduce perturbation cost trajectories, which offer a broader perspective on how a network perceives an input, and we find that adversarial perturbations propagate through deep networks in peculiar ways.

3. Finally, in Section 5, we analyze how scale depends on the underlying data space, choice of distance function, and distribution. The main results are that the robustness curve for one norm-induced distance places bounds on the robustness curves for all other norm-induced distances and that the separation between curves can be extremely large, depending on the model and underlying data distribution. Based on these findings we suggest that the maximum perturbation size of threat models should be selected carefully in order to be meaningful, depending on characteristics of the data set.

2 Robustness curves

Let us consider what we would actually hope to see when performing an adversarial attack, instead of the small performance-breaking perturbations one encounters. We would hope to see either a meaningful – but small – alteration of the original input, such as an airplane growing eyes, a beak, and plumage, to be classified as a bird; or a complete dismantling of meaningful features resulting in unrecognizable noise, where we can readily accept pseudo-random class predictions. Semantics are crucial in both scenarios, which leads us to the conviction that our human perception and classification are integral to the nature of adversarial examples. Still, we must work with a more rigorous definition and therefore abstract further, which we do in the following.

An adversarial perturbation for a classifier $f$ and input-output pair $(x, y)$ is a small perturbation $\delta$ with $f(x + \delta) \neq y$. The resulting input $x + \delta$ is called an adversarial example. The set of input-output pairs vulnerable to adversarial perturbations is the set of data points which the classifier can be induced to misclassify using small perturbations – that is the set of points that are either already misclassified, or that lie in some small distance from the decision boundary. But what does small mean? How should distance be measured? And what distinguishes adversarial examples from other data points? These are the questions we aim to answer in the following sections.

One tool for our study of these questions are robustness curves [9], which we shortly explain in this section. A robustness curve is the cumulative distribution function of the distance of points from the decision boundaries of a classifier:

Definition 1. Given an input space $X$ and label space $Y$, distance function $d$ on $X \times X$, and classifier $f : X \rightarrow Y$. Let $|X| = \max_{x, x' \in X} d(x, x')$ and let $P$ denote the data distribution that generates the samples. Then the $d$-robustness curve for $f$ is the graph of the function

$$R'_d : [0, |X|] \rightarrow [0, 1]$$

$$\varepsilon \mapsto P((x, y) \text{ s.t. } \exists x' : d(x, x') \leq \varepsilon \wedge f(x') \neq y)$$

A model’s robustness curve shows how data points are distributed in relation to the decision boundaries of the model. The advantage of this distributional perspective is that it allows a step back from robustness regarding a specific quantification of what it means to be close to the decision boundary, and instead allows us to compare global robustness properties and their dependence on a given classifier, distribution and distance function. One of the main questions driving this work is how robustness – and robust training – are affected by the choice of distance function.

In principle, there exist a wealth of possible distance functions, and the choice of one particular distance function should depend on the downstream application domain of the model. For this work, we will assume that the distance function $d$ is induced by some norm $n$ on $X$, i.e. $d(x, x') = n(x - x')$. As Göpfert et al. [10] show, this is not sufficient to encode dissimilarities of images as perceived by a human, or guarantee that perturbations are imperceptible to the human eye. However, the majority of current work on robustness uses distances induced by $\ell_p$ norms [19]. They are easy to define and well studied, and it is reasonable to assert that perturbations that are small in some $\ell_p$ norm typically should not change the classification of real-world images. In the following, we always assume that the distance we consider is induced by some norm, usually an $\ell_p$ norm, on a finite-dimensional vector space. In our experiments we consider the $\ell_1$, $\ell_2$ and $\ell_\infty$ norms, as these are frequently used prototypical examples.

3 Robustness curves and their dependence on distance functions

Due to strong concerns about trust in and security of machine learning models that are susceptible to adversarial perturbations, the last years have seen significant effort in developing training methods that produce more robust models. These training methods typically require the choice of a distance function $d$ and perturbation threshold $\varepsilon$. The pair $(d, \varepsilon)$ is sometimes called a threat model. The training method will attempt (and possibly guarantee) to make the model robust against attacks in distance
The distortion factor \( c \) is given by \( R^d_p(\epsilon) \). In the following, we analyze to what extent the minimization of \( R^d_p(\epsilon) \) impacts \( R^d_p(\epsilon') \) for other choices \( d' \) and \( \epsilon' \).

For the special case of linear classifiers, we find that the global robustness behavior of a model w. r. t. a data distribution for all \( \ell_p \) norms is fully specified by the model parameters and the robustness curve for a single \( \ell_p \) norm. Thus, we fully understand the behavior of \( R^d_p(\epsilon') \) for any \( \ell_p \) norm induced distance \( d' \) and any \( \epsilon' \) based on knowing \( R^d_p(\cdot) \) for one \( \ell_p \) norm induced distance \( d \). This is an extension of a weaker theorem from [9].

**Theorem 1.** If \( f \) is a linear classifier on \( \mathbb{R}^d \) parameterized by \((\mathbf{w}, b)\), i.e., \( f(x) = \text{sgn}(\mathbf{w}^T x + b) \), then the shape of the robustness curve for \( f \) regarding an \( \ell_p \) norm-induced distance does not depend on the choice of \( p \). There exists a constant \( c > 0 \) such that

\[
R^d_{\|\cdot\|_p}(\epsilon) = R^d_{\|\cdot\|_2}(c \cdot \epsilon) \quad \forall \epsilon.
\]

The distortion factor \( c \) is given by \( \frac{\|\mathbf{w}\|_1}{\|\mathbf{w}\|_2} \), where \( q_1 = \frac{p_1}{p_2-1} \).

We provide a proof in Appendix A. Clearly, the distortion factor \( c \) can be as large as \( m \), where \( m \) is the dimensionality of the input space. See Appendix B.2 for an example where this occurs. One way to minimize this distortion factor is to encourage sparse weight vectors. However, as illustrated by Appendix B.2, this may backfire by worsening robustness for norms the model would otherwise be robust to, instead of improving robustness for norms it would otherwise not be robust to.

For more complex decision boundaries, dependence on \( p \) can be more than a question of scale. The shape of the robustness curve may depend on \( p \), i.e. the distribution of the distances between points and decision boundary may be different for different choices of \( p \). See Figure 1 for an example. In the following, we empirically evaluate robustness curves for different models to investigate how robustification efforts transfer across norms. For complex models, exactly calculating the distance of a point to the closest decision boundary, and thus estimating the robustness curve, is computationally very intensive, if not intractable. Instead, we bound the true robustness curve from below using adversarial attacks. To verify that this approach is justified, we compare a robustness curve estimated using adversarial attacks to one based on exact calculation of distances to the decision boundary on a simple robust model. We observe that the shape and scale of the curves are very similar, especially for small perturbation sizes, which are most relevant in assessing the robustness of the model. See Appendix D.1 for detailed results.

We compare \( \ell_\infty \) and \( \ell_2 \) robustness curves for different robustified models from the literature. The network architecture we use in our analysis is a convolutional network with two convolutional layers, two fully connected layers and ReLu activation functions. We compare the following training methods:

1. Standard training (ST), i.e. training without specific robustness considerations.
2. Adversarial training (AT), see Madry et al. [15].
3. Training with robust loss (KW), see Wong and Kolter [20].
4. Maximum margin regularization for a single \( \ell_p \) norm together with adversarial training (MMR + AT), see Croce et al. [5]
5. Maximum margin regularization simultaneously for \( \ell_\infty \) and \( \ell_1 \) margins (MMR-UNIV), see Croce and Hein [6].

Together with each training method, we state the threat model the model is optimized to guarantee (e.g. Threat Model: \( \ell_\infty(\epsilon = 0.1) \) for robustness guarantees in \( \ell_\infty \) norm for \( \ell_2 = 0.1 \), if available. Unless explicitly stated otherwise, the trained models are those made publicly available\(^2\) by Croce et al. [5] and Croce and Hein [6]. We use six datasets: MNIST, Fashion-MNIST, German Traffic Signs (GTS) [12], CIFAR-10, Tiny-ImageNet-200 (TINY-IMG) [14], and Human Activity Recognition (HAR) [1]. For specifics on model training, refer to Appendix C. Models are generally trained on the full training set for the corresponding data set, and robustness curves evaluated on the full test set, unless stated otherwise. See Appendix D.2 for examples of adversarial perturbations optimized for different norms and different models, which provides intuition on how norm constraints and adversarial defenses affect adversarial perturbations.

First, we compute \( \ell_2 \) and \( \ell_\infty \) robustness curves on MNIST for the five different training methods mentioned above, each for \( \ell_2 \) and \( \ell_\infty \) threat models, except for MMR-UNIV which simultaneously defends against attacks in all \( \ell_p \) norms. The results can be seen in Figure 2. We find that for AT and MMR + AT, the \( \ell_\infty \) threat model leads to better robustness than the \( \ell_2 \) threat model both for \( \ell_1 \) and \( \ell_2 \) robustness curves. In fact, MMR + AT with the \( \ell_\infty \) threat model even leads to better \( \ell_\infty \) and \( \ell_2 \) robustness curves than MMR-UNIV, which is specifically designed to improve robustness for all \( \ell_p \) norms. For KW, interestingly the \( \ell_\infty \) threat model leads to better \( \ell_\infty \) robustness than the \( \ell_2 \) threat model only for perturbation sizes up to around 0.15, and the \( \ell_2 \) threat model only leads to better \( \ell_2 \) robustness than the \( \ell_\infty \) threat model up to perturbation size around 1.2, after which the two curves intersect. For all three training methods under the \( \ell_\infty \) threat model, we observe a noticeable change in slope of the \( \ell_\infty \) robustness curve between perturbation sizes 0.1 and 0.2 (perturbation size in the threat model chosen to be 0.1 in each case). MMR + AT is significant in that the change in slope occurs later than in the other models, even though the target perturbation size is identically 0.1. Each method leads to significantly improved robustness curves as compared to ST, even beyond the perturbation sizes specified by the threat model.

Because perturbations sizes in \( \ell_\infty \) and \( \ell_2 \) norm tend to be very different, it is difficult to compare robustness curves for the two norms in a single plot. In Figure 3, we introduce two methods of rescaling to better compare the shapes of the curves. With method 1, we rescale the \( \ell_\infty \) robustness curve by the mean ratio \( \frac{R_{\|\cdot\|_2}(\epsilon)}{R_{\|\cdot\|_\infty}(\epsilon)} \) between the curves. This method allows us to see even more clearly how the \( \ell_\infty \) robustness curve for MMR + AT with \( \ell_\infty \) threat model changes slope around the target perturbation size, which does not occur in the \( \ell_2 \) robustness curve for any of the training methods. With method 2, we rescale the \( \ell_\infty \) robustness curve by the mean ratio \( \frac{R_{\|\cdot\|_2}(x)}{R_{\|\cdot\|_\infty}(x)} \) for all points \( x \) in the data set. The benefit of this rescaling method is that it does not require \(^2\)www.github.com/fra31/mmr-universal, www.github.com/max-andr/provable-robustness-max-linear-regions
Figure 1: A model with a non-linear decision boundary and three differently shaped robustness curves.

Figure 2: Robustness curves resulting from different training methods (MMR + AT, MMR-UNIV (first column), KW (second column), and AT (third column), always with ST for comparison) and different threat models (indicated by color), measured in the $\ell_\infty$ norm (first row) and the $\ell_2$ norm (second row).

Figure 3: Scaled and unscaled robustness curves for different training methods. Scaling methods are indicated by plot titles. Training methods are indicated by plot titles.
computation of both curves, but, at least empirically, can still provide an estimate of the $\ell_2$ robustness curve or vice versa. It is interesting to see that method 2 overestimates the $\ell_2$ robustness curve for ST, underestimates it for MMR + AT with the $\ell_\infty$ threat model, and almost perfectly estimates it for MMR + AT with the $\ell_2$ threat model. This reinforces that compared to standard training, MMR + AT with both the $\ell_\infty$ and $\ell_2$ threat models improve $\ell_\infty$ robustness more strongly than $\ell_2$ robustness.

Figure 4 shows $\ell_\infty$ robustness curves for MMR + AT with $\ell_\infty$ threat model as provided by Croce et al. [5]. The models trained on MNIST and FMNIST both show the characteristic change in slope. Further, the robustness curves for CIFAR-10 and GTS show that the models for these two data sets, which were trained with much smaller target perturbation sizes, are very non-robust compared to the former two models and do not have a visually detectable change in slope. We will revisit the question of which target perturbation level to choose in dependence on the data set in Section 5.

4 How adversarial perturbations are processed

In the previous section, our focus was on how robust training methods impact the robustness curve of the resulting model. Now, we investigate how they impact the way the neural network “perceives” adversarial perturbations. Instead of only looking at the perturbed input and the resulting (mis-)classification, we follow the effects of a perturbation along every step along its way through the network – i.e. at each layer. As a tool for this, we introduce perturbation cost trajectories.

Definition 2. Let $f : \mathbb{R}^m \to \mathbb{R}^l$ be the function calculated by a neural network, where $m$ is the input dimensionality, $l$ is the cardinality of the label space, and the output of $f$ are the softmax probabilities for each class. Let $k$ be the number of layers in the network. For $x \in \mathbb{R}^m$ and $i \in \{0, \ldots, k\}$ let $f_i(x)$ denote the output of the $i$-th layers of the network, applied to $x$. In particular, $f_0(x) = x$ and $f_k(x) = f(x)$. Let $x$ be an input to the network, let $\Delta \in \mathbb{R}^m$ be a perturbation and let $\|\cdot\|$ denote a norm. Then the perturbation cost of $\Delta$ for $x$ at layer $i \in \{1, \ldots, k-1\}$ is defined as

$$C^i_x(\Delta) := \frac{\|f_i(x + \Delta) - f_i(x)\|}{\|f_i(x)\|}. \quad (2)$$

The perturbation cost of $\Delta$ at layer $k$ is $C^k_x(\Delta) = 1_{f(x+\Delta) \neq f(x)}$. The perturbation cost trajectory of $\Delta$ at $x$ is the sequence $(C^i_x(\Delta))_{i=1}^k$.

Adversarial versus random perturbations

In the following, we present and discuss several perturbation cost trajectories for a simple architecture trained with either ST or MMR + AT. We perturb 1000 data points from the MNIST test set and feed them through the network – then, for comparison, we sample random perturbations of the exact same size and trace those through the network. For each trained model, this yields two pairs of curves – one for the $\ell_\infty$ norm and one for the $\ell_2$ norm. Perturbation cost trajectories are computed w.r.t. the norm that the perturbation is optimized for. This allows a number of observations (cf. Figure 5):

1. For all three models, data points perturbed with random noise are almost never misclassified, compared to (by construction) all data points with adversarial perturbations.

2. Only perturbation costs of $\ell_2$ scaled perturbations decrease monotonously with each additional layer.

3. In the non-robust network, perturbation costs for adversarial perturbations are amplified by each layer.

4. Perturbation cost trajectories for adversarial perturbations are similar for $\ell_2$-optimized adversarial perturbations and $\ell_\infty$-optimized adversarial perturbations.

5. Perturbation costs for random perturbations in $\ell_\infty$ norm decrease more slowly than those for $\ell_2$ norm. For the model robustified to $\ell_\infty < 0.1$, the random perturbation cost trajectory mirrors that of the adversarial perturbations, but leads to misclassification in only approximately 10% of cases.

By construction, adversarial examples are close to a decision boundary, but what about the rest of their surroundings? In Figure 6, we present perturbation cost trajectories for random perturbations added to adversarial perturbations. We observe that, although each adversarial example is misclassified (as that is how it was constructed), only a small fraction of points in its immediate vicinity are also misclassified. However, contrary to what we see for random perturbations added to the original inputs (cf. Figure 5), perturbation costs for random perturbations on top of adversarial perturbations do not decrease layer by layer – instead, the average perturbation cost trajectory for the adversarial perturbation with random noise follows the average perturbation cost trajectory for the adversarial perturbations. The two trajectories only separate in the argmax layer. This shows that perturbations in the general vicinity of the adversarial perturbation, are not abstracted away by the neural network processing, although most of them do not affect the final classification. We also observe that a much larger proportion of points in the vicinity of the adversarial examples for the $\ell_2$-robustified model are misclassified than for the $\ell_\infty$-robustified model.

Since $\ell_\infty$ robustification also improves $\ell_2$ robustness and vice versa (see Section 3), one might expect that adversarial examples themselves transfer between the models robustified for different norms. To explore this, we find adversarial perturbations for each trained model and calculate the respective trajectories not only for said model, but for all three models – see Figure 7. The adversarial perturbations are only successful against the models upon which they were created. In fact, both robustified models process “foreign” adversarial perturbations similarly to $\ell_\infty$-scaled random perturbations. Perhaps surprisingly, the non-robust ST model manages to correctly classify almost all “foreign” adversarial examples, even though the perturbations are relatively large.

All in all, we observe that both models trained with MMR + AT have less variation in perturbation costs across layers than the non-robust ST model. We are surprised by how strongly the perturbation cost trajectory decreases for $\ell_2$-scaled random per-
Figure 4: Approximated $l_\infty$ robustness curves for multiple data sets. Each curve is calculated for a different model and a different test data set. The data sets are indicated by the labels. The models are trained with MMR + AT, Threat Models: MMRIST: $\ell_\infty(e = 0.1)$, FMNIST: $\ell_\infty(e = 0.1)$, GTS: $\ell_\infty(e = 4/255)$, CIFAR-10: $\ell_\infty(e = 2/255)$.

Figure 5: Averaged perturbation cost trajectories for adversarial and random perturbations for ST and AT. The labels of the curves show the perturbation type of the individual curves. The training methods are indicated by the subplot titles. In Appendix D.3 we present an additional experiment to explore why the $\ell_\infty$-robustified model misclassifies a number of random $\ell_\infty$ perturbations (middle).

Figure 6: Perturbation cost trajectories for adversarial perturbations plus random perturbations of different sizes for different training methods. The labels of the curves show the perturbation type and random perturbation size of the individual curves. The training methods are indicated by the subplots titles.

Figure 7: Perturbation cost trajectories for $\ell_\infty$ adversarial perturbations (optimized for different threat models) for models trained with different training methods. The labels of the curves show the perturbation type and the model, for which the perturbation was optimized. The training methods are indicated by the column titles.
turbations, for all three models. Our finding that “foreign” adversarial examples are repelled is at odds with the observation that for black-box attacks, transfer of adversarial examples between different models seems to be crucial.

5 Scale of robustness curves

The dependence of scale on the norm

In Section 3, we showed examples of how different the scale of robustness curves (as characterized, for example, by the average distance of a sample from a decision boundary) can be, both based on the distance function chosen as well as the data set under consideration. In the following, we present some considerations on how heavily scale may vary, and end with some practical recommendations for practitioners working with popular data sets. We begin by observing that on finite-dimensional vector spaces, one robustness curve places natural bounds on all others that represent distances induced by norms:

**Proposition 1.** Let \( X \) be a finite-dimensional vector space. Let \( d \) and \( d' \) be two distance measures on \( X \) that are induced by norms. Then there exist constants \( c_1, c_2 > 0 \) such that for every classifier \( f \),

\[
R^d(f, \epsilon) \leq R^{d'}(f, \epsilon) \leq R^d(f, c_2 \epsilon) \quad \forall \epsilon.
\]

See Appendix B.1 for the proof. Even restricting ourselves to \( \ell_p \) norm-induced distances, it is clear that the bounds provided by Proposition 1 are not necessarily informative. As noted in the discussion for Theorem 1, even for this reduced class of distances, the constants \( c_1, c_2 \) can lead to a very broad range. For example, we find

\[
R^{\ell_p}(f, \epsilon) \leq R^{\ell'_p}(f, \epsilon) \leq R^{\ell_p}(f, m \epsilon)
\]

where \( m \) is the data dimensionality, and these inequalities are tight. See Appendix B.2 for an example of a data distribution where we construct two classifiers with identical \( \ell_p \) robustness, but \( \ell_1 \) robustness that differs by a factor of \( m \). Consequently, at least in high dimensions, the scale of a robustness curve can, but doesn’t necessarily depend strongly on the choice of norm.

Scale for specific data sets

Robustness curves allow us to analyze robustness globally, without focusing too much on (more or less) arbitrary thresholds. Nonetheless, the question of scale remains. Which perturbations should one consider “large” or “small”? One of the underlying assumptions in the definition of adversarial examples is that when \( f(x') \neq y \), \( x' \) is being incorrectly classified, i.e., \( f(x') = y \) would be the correct classification choice. The question of scale therefore cannot be answered independently of the data distribution. In order to understand how to interpret different perturbation sizes, it can be helpful to understand how strongly the data point would need to be perturbed to actually change the correct classification. In this section, we analyze this question for several popular data sets.

Recall that for two differently labeled points \((x_1, y_1)\) and \((x_2, y_2)\), with \( x_1 \neq x_2 \) and \( y_1 \neq y_2 \), the perturbation \( x_2 - x_1 \) when applied to \( x_1 \) hopefully leads to a change in prediction, but it can hardly be considered adversarial. Instead, it can provide some sense of scale and serve as an upper bound on reasonable robustness thresholds: when a perturbation is large enough to change an input such that the result should *correctly* be assigned a different label, it might not be sensible to expect a model to be robust against it. Similarly, when a perturbation does not even “reach” the closest point of the same label, it probably should not be able to influence a model’s prediction. Of course, this is subject to the concrete distribution in question and depends on the specific sampled data that is available.

When we look at the respective numbers for the MNIST test set in the \( \ell_\infty \), \( \ell_2 \), and \( \ell_1 \) norms, we can make several observations. See Figure 8 for the distributions of distances between points and their closest neighbors from the same class and different classes. Because the smallest inter-class distance in the \( \ell_\infty \) norm is around 0.9, we can see that transforming an input from one class to one from a different class almost always requires completely flipping at least one pixel from almost-black to almost-white or vice versa. Transforming within a given class frequently requires only a smaller perturbation. For the \( \ell_2 \) and \( \ell_1 \) norms, the inter-class distance distributions are more spread out than the \( \ell_\infty \) inter-class distance distribution. We observe that with \( \ell_2 \) perturbations of size \( \geq 3 \), it becomes possible to transform samples from different classes into each other, so starting from this threshold, any classifier must necessarily trade off between accuracy and robustness.

Now that we have developed a certain sense of scale w. r. t. MNIST, let us compare it to two further data sets, namely GTS and TINY-IMG. We present the inter- and intra-class distance distributions in the \( \ell_\infty \) norm (where all distances lie in the interval \([0, 1]\)) in Figure 9. The shapes of the curves differ strongly between data sets, and – most notably – for TINY-IMG, the distance of each point to the nearest neighbor from a different class is smaller than the distance to the nearest neighbor within the same class. Presumably, this is due to high data dimensionality, a large number of classes, and relatively large variation within a given class, when compared to MNIST – refer to Table 1 for exact values.

Finally, we compare the inter-class distance distributions in \( \ell_\infty \), \( \ell_2 \), and \( \ell_1 \) norm for all data sets considered in this work – see Figure 10. We observe that for the \( \ell_1 \) norm, the shape of the curves is similar across data sets, but their extent is determined by the dimensionality. In the \( \ell_\infty \) norm, vastly different curves emerge for the different data sets. We hypothesize that, because the inter-class distance distributions vary more strongly for \( \ell_\infty \) distances than for \( \ell_1 \) distances, the results of robustifying a model w. r. t. \( \ell_\infty \) distances may depend more strongly on the underlying data distribution than the results of robustifying w. r. t. \( \ell_1 \) distances. This is an interesting avenue for future work. In any case, it is safe to say that, when judging the robustness of a model by a certain threshold, that number must be set with respect to the scale of the distribution the model operates on.

In Table 1, we summarize the smallest and largest inter-class distances in different norms together with additional information about the size, number of classes, and dimensionality of the all the data sets we consider in this work. The values correspond directly to Figure 10, but even in this simplified view, we can quickly make out key differences between the data sets. Compare, for example, MNIST and GTS: While it appears entirely reasonable to expect \( \ell_\infty \), robustness of 0.5 for MNIST, the same
Figure 8: Minimum inter- and intra-class distances in different $\ell_p$ norms. Red curves: Sorted closest distances of each point to the closest point of the same class. Blue curves: Sorted closest distances of each point to the closest point of a different class. Left: Distance Measured in $\ell_\infty$. Middle: Distance measured in $\ell_2$. Right: Distance measured in $\ell_1$. Data set is the full MNIST test set (10 000 points).

Figure 9: Minimum inter- and intra-class $\ell_\infty$ distances for three data sets. Red curves: Sorted closest $\ell_\infty$ distances of each point to the closest point of the same class. Blue curves: Sorted closest distances of each point to the closest point of a different class. Left: 10 000 points from MNIST test data set. Middle: 10 000 points from GTS test data set. Right: 9832 points from TINY-IMAGENET test set.

Figure 10: Minimum inter-class distances of all data sets considered in this work, measured in $\ell_\infty$ (left), $\ell_2$ (middle), and $\ell_1$ (right) norm. See Table 1 for size and dimensionality.
threshold for GTS is not possible. Relating Table 1 and Figure 4, we find entirely plausible the strong robustness results for MNIST, and the small perturbation threshold for GTS. Based on inter-class distances we also expect less $\ell_\infty$ robustness for CIFAR-10 than for FMNIST, but not as seen in Figure 10.

Overall, the strong dependence of robustness curve scaling on the data set and the chosen norm, emphasizes the necessity of informed and conscious decisions regarding robustness thresholds. We hope that Table 1 can provide an easy reference when judging scales in a threat model.

6 Conclusion

We have investigated the relationships between adversarial defenses, the threat models they address, and the robustness of a model w.r.t. small perturbations in different norms, using robustness curves and perturbation cost trajectories. We have found that $\ell_\infty$ threat models are surprisingly effective in improving robustness for other $\ell_p$ norms, and hope that future defenses will be evaluated from this perspective. We have used perturbation cost trajectories to gain a broader view on how robust and non-robust networks perceive adversarial perturbations as opposed to random perturbations. Finally, we have seen how suitable robustness thresholds necessarily depend on the data set under consideration.

It is our hope that practitioners and researchers alike will use the methodology proposed in this work, especially when developing and analyzing adversarial defenses, and carefully motivate any threat models they might choose, taking into account the available context.

References

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and J Reyes-Ortiz. “A Public Domain Dataset for Human Activity Recognition using Smartphones”. In: ESANN. Jan. 2013.

[2] Nicholas Carlini and David A. Wagner. “Towards Evaluating the Robustness of Neural Networks”. In: 2017 IEEE Symposium on Security and Privacy (SP). 2017. arXiv:1608.04644.

[3] Nicholas Carlini et al. On Evaluating Adversarial Robustness. 2019. arXiv:1902.06705.

[4] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples. 2017. arXiv:1709.04114.

[5] Francesco Croce, Maksym Andriushchenko, and Matthias Hein. “Provable Robustness of ReLU networks via Maximization of Linear Regions”. In: Proceedings of Machine Learning Research. 2019. arXiv:1810.07481.

[6] Francesco Croce and Matthias Hein. “Provable robustness against all adversarial $l_p$-perturbations for $p \geq 1$”. In: International Conference on Learning Representations. 2020. arXiv:1905.11213.

[7] Elvis Dohmatob. “Generalized No Free Lunch Theorem for Adversarial Robustness”. In: Proceedings of the 36th International Conference on Machine Learning. 2019. arXiv:1810.04065.

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and Harnessing Adversarial Examples”. In: 3rd International Conference on Learning Representations. 2015. arXiv:1412.6572.

[9] Christina Göpfert, Jan Philip Göpfert, and Barbara Hammer. “Adversarial Robustness Curves”. In: Machine Learning and Knowledge Discovery in Databases. 2020. arXiv:1908.00096.

[10] Jan Philip Göpfert, André Artelt, Heiko Wersing, and Barbara Hammer. “Adversarial attacks hidden in plain sight”. In: ICA. 2020. arXiv:1902.09286.

[11] Pascale Gourdeau, Varun Kanade, Marta Kwiatkowska, and James Worrell. “On the hardness of robust classification”. In: Advances in Neural Information Processing Systems. 2019. arXiv:1909.05822.

[12] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. “Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark”. In: IJCNN. 1288. 2013.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 2014. arXiv:1412.6980.

[14] Fei-Fei Li, Andrej Karpathy, and Justin Johnson. CS231n: Convolutional Neural Networks for Visual Recognition. [Online; accessed March 28, 2020]. 2016. URL: http://cs231n.stanford.edu/2016/project.html.

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. “Towards Deep Learning Models Resistant to Adversarial Attacks”. In: ICLR. 2018. arXiv:1706.06083.
A Proof of Theorem 1

Theorem 1. If \( f \) is a linear classifier on \( \mathbb{R}^m \) parameterized by \((w^T, b)\), i.e. \( f(x) = \text{sgn}(w^T x + b) \), then the shape of the robustness curve for \( f \) regarding an \( \ell_p \) norm-induced distance does not depend on the choice of \( p \). There exists a \( c > 0 \) such that

\[
R_{\|\cdot\|_p}^f (c \cdot \varepsilon) = R_{\|\cdot\|_{p_1}}^f (c \cdot \varepsilon)
\]

(5)

The distortion factor \( c \) is given by \( \|w_i\|_{\|\cdot\|_p} / \|w_i\|_{\|\cdot\|_{p_1}} \), where \( q_i = \frac{p_i}{p_i - 1} \).

Lemma 1. Let \( x \in \mathbb{R}^m \) with \( w^T x + b \neq 0 \). Let \( p \in [1, \infty] \) and \( q \) such that \( \frac{1}{p} + \frac{1}{q} = 1 \), where we take \( \frac{1}{p} = 0 \). Then

\[
\min \{ ||\delta||_p : \text{sgn}(w^T(x + \delta) + b) \neq \text{sgn}(w^T x + b) \} = \frac{||w^T + b||}{||w||_q}
\]

(6)

and the minimum is attained by

\[
\delta = \left( -w^T x - b \right) \text{sgn}(w_j) e_j, j = \arg \max_{j} |w_j|, \quad p = 1
\]

\[
\left( -w^T x - b \left( \text{sgn}(w_j) |w_j|^{\frac{p}{p-1}} \right)^d \right)
\]

\( p \in (1, \infty) \).

Proof. By Hölder’s inequality, for any \( \delta \),

\[
\sum_{i=1}^{m} |w_i \delta_i| \leq ||\delta||_p \|w\|_q.
\]

(8)

For \( \delta \) such that \( \text{sgn}(w^T(x + \delta) + b) \neq \text{sgn}(w^T x + b) \) it follows that

\[
||\delta||_p \geq \frac{\sum_{i=1}^{m} |w_i \delta_i|}{||w||_q} \geq \frac{|w^T x + b|}{||w||_q}. \]

(9)

Using the identity \( q = \frac{p}{p-1} \), it is easy to check that for every \( p \in [1, \infty] \), with \( \delta \) as defined in Equation (7),

1. \( w^T \delta = -w^T x - b \), so that \( w^T(x + \delta) + b = 0 \), and
2. \( ||\delta||_p = \frac{||w^T x + b||}{||w||_q} \).

Item 1 shows that \( \delta \) is a feasible point, while Item 2 in combination with Equation (9) shows that \( ||\delta||_p \) is minimal.

Using Lemma 1, we are ready to prove Theorem 1.

Proof. By definition,

\[
R_{\|\cdot\|_{p_1}}^f (\varepsilon) = P(\{ (x, y) \text{ s.t. } \exists \delta : ||\delta||_{p_1} \leq \varepsilon \land f(x + \delta) \neq y \}) \quad (10)
\]

We can split \( R_{p_1}^f (\varepsilon) \) into the disjoint sets

\[
\{(x, y) : f(x) \neq y \land \exists \delta : ||\delta||_{p_1} \leq \varepsilon \}
\]

(11)

\[
\{(x, y) : f(x) = y \land \exists \delta : ||\delta||_{p_1} \leq \varepsilon \}
\]

(12)

Choose \( q_1, q_2 \) such that \( \frac{1}{p_1} + \frac{1}{q_1} = 1 \). By Lemma 1, and using that \( f(x) = \text{sgn}(w^T x + b) \),

\[
B_{p_1}(\varepsilon) = \{(x, y) : \text{sgn}(w^T x + b) = y \land \frac{|w^T x + b|}{||w||_{q_1}} \leq \varepsilon \}
\]

(14)

\[
= \{(x, y) : \text{sgn}(w^T x + b) = y \land \frac{|w^T x + b|}{||w||_{q_2}} \leq \frac{||w||_{q_1}}{||w||_{q_2}} \varepsilon \}
\]

(15)

\[
= B_{p_2} \left( \frac{||w||_{q_1}}{||w||_{q_2}} \varepsilon \right).
\]

(16)

This shows that

\[
R_{\|\cdot\|_{p_1}}^f (\varepsilon) = P(M) + P(B_{p_1}(\varepsilon)) \quad (17)
\]

\[
= P(M) + P \left( B_{p_2} \left( \frac{||w||_{q_1}}{||w||_{q_2}} \varepsilon \right) \right) \quad (18)
\]

\[
= R_{\|\cdot\|_{p_2}}^f \left( \frac{||w||_{q_1}}{||w||_{q_2}} \varepsilon \right). \quad (19)
\]

\[\Box\]

B Robustness Curve Separation

B.1 Proof of Proposition 1

Proposition 1. Let \( X \) be a finite-dimensional vector space. Let \( d \) and \( d' \) be two distance measures on \( X \) that are induced by norms. Then there exist constants \( c_1, c_2 > 0 \) such that for every classifier \( f \),

\[
R_{d'}^f (c_1 \varepsilon) \leq R_d^f (c_2 \varepsilon) \quad \forall \varepsilon. \quad (20)
\]

Proof. If \( X \) is a finite-dimensional vector space, it is known that any two norms on \( X \) are equivalent, i.e. for any two norms \( n, n' \) on \( X \) there exist constants \( c_1, c_2 > 0 \) such that for all \( x \in X \),

\[
n(x) \leq c_1 n'(x) \text{ and } n'(x) \leq c_2 n(x).
\]

(21)

For a distance function \( d \) and perturbation size \( \varepsilon \), let \( R_d(\varepsilon) := \{ (x, y) \text{ s.t. } \exists x' : f(x') \neq y \land d(x, x') \leq \varepsilon \} \),

(22)

so that \( R_d^f (\varepsilon) = P(R_d(\varepsilon)) \). Let \( d, d' \) be induced by the norms \( n, n' \). If \( (x, y) \in R_d(c_1 \varepsilon) \), it follows that \( (x, y) \in R_d(\varepsilon) \), and if \( (x, y) \in R_d(\varepsilon) \), it follows that \( (x, y) \in R_d(c_2 \varepsilon) \). As a result,

\[
R_{d'}^f (c_1 \varepsilon) \leq R_d^f (\varepsilon) \leq R_{d'}^f (c_2 \varepsilon). \quad (23)
\]

\[\Box\]
A sparse weight vector, the result of a $\ell_1$-regularized linear SVM, with one non-zero entry. Right: a dense weight vector, the result of a $\ell_2$-regularized linear SVM, with no non-zero entries. The vectors are reshaped to $16 \times 16$. Black corresponds to $-1$, beige corresponds to 1, and reddish colors correspond to values 0 or close to 0.

Figure 11: Left two images: the types of images distinguished between in the toy classification problem. Second from the right: a sparse weight vector, the result of a $\ell_1$-regularized linear SVM, with one non-zero entry. Right: a dense weight vector, the result of a $\ell_2$-regularized linear SVM, with no non-zero entries. The vectors are reshaped to $16 \times 16$. Black corresponds to $-1$, beige corresponds to 1, and reddish colors correspond to values 0 or close to 0.

B.2 Example of robustness curve separation

As an illustrative toy example of how strongly the robustness curves for different norms can be separated, consider the following classification task. The goal is to distinguish between two types of $16 \times 16$-pixel images in Figure 11. Running a $\ell_1$-regularized linear SVM on this data leads to the sparse weight vector schematically represented in Figure 11, with just one non-zero entry. Running a $\ell_2$-regularized linear SVM leads to the dense weight vector also schematically represented in Figure 11, with all non-zero entries.

Both classifiers have 100% accuracy, and all samples are at $\ell_\infty$-distance 1 from both decision boundaries, i.e. the $\ell_\infty$ robustness curves of both classifiers are identical. However, the $\ell_1$ robustness curves of the two classifiers are maximally separated: for the sparse classifier, the $\ell_1$ robustness curve is a step function at 1, while for the dense classifier, it is a step function at 256, the dimensionality of the data. Figure 12 shows adversarial examples for the two classifiers. For the sparse classifier, it is sufficient to flip a single pixel, while for the dense classifier, more noticeable changes are necessary.

C Experimental details

C.1 Model training

We use the same model architecture as [5] and [20]. Unless explicitly stated otherwise, the trained models are taken from [5]. The exact architecture of the model is: Convolutional layer (number of filters: 16, size: 4x4, stride: 2), ReLu activation function, fully connected layer (number of units: 100), ReLu activation function, output layer (number of units depends on the number of classes). All models are trained with Adam Optimizer [13] for 100 epochs, with batch size 128 and a default learning rate of 0.001. More information on the training can be found in the experimental details section of the appendix of [5]. All models are publicly available in the GitHub repositories of [5] and [6]: www.github.com/max-andr/provable-robustness-max-linear-regions and www.github.com/fra31/mmr-universal.

C.2 Approximated robustness curves

We use state-of-the-art adversarial attacks to approximate the true minimal distances of input datapoints to the decision boundary of a classifier for our adversarial robustness curves (see Definition 1). We base our selection of attacks on the recommendations of [3]. Specifically, we use the following attacks: For $\ell_1$ robustness curves we use EAD, [4], for $\ell_2$ robustness curves we use [2] and for $\ell_\infty$ robustness curves we use [15]. For all three attacks, we use the implementations of Foolbox [16]. For the $\ell_\infty$ attack, we use the standard hyperparameters of the Foolbox implementation. For the $\ell_1$ and $\ell_2$ attacks we increase the number of binary search steps that are used to find the optimal tradeoff-constant between distance and confidence from 5 to 10, which we found empirically to improve the results. For the rest of the hyperparameters, we again use the standard values of the Foolbox implementation.

C.3 Generating random perturbations

We compare perturbation cost trajectories for adversarial perturbations and random perturbations. In the following, we specify how the random perturbations are generated. Let $\Delta$ be the perturbation that we want to generate. $\Delta$ is a $m$-dimensional vector, where $m$ is defined by the number of input features of the original input. Let $s$ be the target size of the noise.

If the target size is measured in $\ell_\infty$ norm, we sample

$$\Delta \sim s \cdot (\sigma_i)_{i=1}^m$$

(24)

where the $\sigma_i$ are i.i.d. Rademacher variables, i.e. uniformly distributed on $\{\pm 1\}$. This method of sampling is based on the Fast Gradient Sign Method, and means that our random noise is large as possible under the $\ell_\infty$ norm constraints we impose.
If the target size is measured in $\ell_2$ norm, we let

$$\Delta = s \cdot \frac{X}{\|X\|_2}$$

(25)

where $X \sim \mathcal{N}_m(0, I_m)$, i.e. we sample a vector from a Gaussian distribution and scale it to the desired length.

### C.4 Computational architecture

We executed all programs on an architecture with 2 x Intel Xeon(R) CPU E5-2640 v4 @ 2.4 GHz, 2 x Nvidia GeForce GTX 1080 TI 12G and 128 GB RAM.

### D Additional experiments

#### D.1 Justifying approximated robustness curves

We’ve compared approximated and true $\ell_\infty$ curves to justify our use of approximate robustness curves. For computational reasons, this was done only for MNIST, and for a smaller architecture than our other experiments, and not for $\ell_2$ curves. The results can be seen in Figure 13. We observe that the curves track each other reasonably well, especially for small perturbation sizes, which is the most interesting region. Due to computational constraints, we did not perform this comparison for the more complicated architecture our other experiments are based on. The calculation of the exact robustness curve in Figure 13 took 2 days and 22 hours on our computational setup (see Appendix C for details). We started calculation of the exact curves for the architecture used in our other experiments, but had to cancel after 14 days, since not even a single data point had been evaluated.

#### D.2 Adversarial attack intuition

In order to provide intuition on how adversarial examples differ based on the choice of norm and defense, we show adversarial attacks to an image from GTS in Figure 14.

#### D.3 Distinguishing between defended and non-defended adversarial perturbations

Figure 5 gives the impression that the MMR + AT ($\ell_\infty$) model does not reduce perturbation costs of random $\ell_\infty$ perturbations as the other models do (although only a small number of these perturbations lead to misclassification). To verify that this is not simply due to the larger perturbation size made necessary for the robustified model, we calculate perturbation cost trajectories separated by the size of the adversarial perturbation (defended against or part of the percentage not defended against). For both cases, and thus for both perturbation magnitudes, we observe the same effect.

#### D.4 Calculating inter-class distances

In Figure 16, we show minimum inter-class distances of $\ell_\infty$, $\ell_2$ and $\ell_1$ norm for subsets of commonly used data sets. We can observe, that for all norms and all data sets we examined, a small sample (10 % to 25 %) of the data is sufficient to reliably estimate the minimum inter-class distance of the full data set.
Training:

Threat Model:

Figure 14: Adversarial examples for robust/non-robust models and different norms. The adversarial examples are generated with adversarial attacks ($\ell_\infty$: Projected Gradient Descent Attack (PGD) as suggested in [15], $\ell_2$: Carlini Wagner L2 Attack (CW) as suggested in [2], $\ell_1$: Elastic-net attack (EAD) as suggested in [4]). The adversarial perturbations are scaled by factor 10 to increase visibility. We show adversarial examples for three different models. The training methods are indicated by the labels on the y-axes. The original image is taken from GTS.

Figure 15: Perturbation cost trajectories for $\ell_\infty$ adversarial perturbations and random perturbations for a robust model, separated by class (defended against vs. not defended against). The curves show the mean perturbation costs for a subset of 1000 datapoints of the MNIST test set, that falls into the respective class. The labels of the curves show the perturbation type and the class of the individual curves. The model is trained with MMR+AT, Threat Model: $\ell_\infty(\epsilon = 0.1)$.

Figure 16: Minimum inter-class differences for subsets of different data sets, measured in $\ell_\infty$, $\ell_2$ and $\ell_1$ norm.