Interplay of Adversarial Robustness and Generalization in Deep Convolutional Models

Ivan Grubišić
Department of Electronics, Microelectronics, Computer and Intelligent Systems
Faculty of Electrical Engineering and Computing, University of Zagreb
Zagreb, Croatia
ivan.grubisic@fer.hr

Abstract—**Although common state-of-the-art machine learning algorithms achieve human-level performance in many tasks, when given inputs that are corrupted or domain-shifted, their performance suddenly drops. Moreover, they are extremely sensitive to certain small perturbations, indicating that they do not understand data (e.g. images) in a robust way similar to how humans do. With the goal of better understanding adversarial examples and improving robustness and generalization, a brief overview of research on adversarial examples, their properties, their nature, algorithms for finding them, proposed methods for achieving robustness, and relationship between adversarial robustness and generalization is given.**

1. Introduction

Although common machine learning algorithms achieve human-level performance in many tasks, when given out-of-distribution, domain-shifted, corrupted or slightly modified examples from the training data distribution, they can often make overconfident and incorrect predictions [1, 2, 3, 4, 5, 6]. Perhaps most surprisingly, by perturbing input examples (e.g. images) even inperceptibly for humans, common state-of-the-art algorithms can be made to significantly change their predictions both for examples from and outside of the training data distribution [6, 7]. Such perturbed examples are called *adversarial examples*. The existence of adversarial examples indicates that common algorithms are probably performing well for somewhat wrong reasons, without actually *understanding data*.

Although adversarial examples exist on other tasks, image classification algorithms with deep, mostly convolutional, models is the problem considered in most research on adversarial examples. In the following text, the problem of image classification will be considered mostly as well.

Some evidence suggests that there is a trade-off between robustness and generalization [8, 9, 10] with current algorithms, which is counter-intuitive because a hypothesis which optimally generalizes would have no adversarial examples. The question remains whether it is feasible or tractable with respect to computation or amount of data to implement such algorithms. A small but interesting step in this direction has been made by [11].

A brief overview over a subset of the research on adversarial examples will be given with the goals of improving intuition and understanding of adversarial examples, understanding how robustness and generalization of deep models relate, and getting closer to knowing whether it is possible to improve both robustness and generalization on real-world datasets.

2. Definitions and notation

To prevent some ambiguities, we define some basic machine learning concepts as they will be used here:

- Hypothesis – a mapping from inputs to predictions. A hypothesis will be denoted $h(\boldsymbol{x})$ or $h(\boldsymbol{x};\boldsymbol{\theta})$, where $\boldsymbol{x} \in \mathbb{X}$ is an input vector and $\boldsymbol{\theta}$ a fixed vector of model parameters. For discriminative models $h(\boldsymbol{x}) = \mathrm{p}(\underline{y} \mid \boldsymbol{x})$, where $\underline{y} \mid \boldsymbol{x}$ is a conditioned random variable, short for $\underline{y} \mid \underline{\boldsymbol{x}} = \boldsymbol{x}$. Random variables are underlined.
- Model – a set of hypotheses $\mathbb{H}$ or a function h defined up to some free parameters $\boldsymbol{\theta}$: $\mathbb{H} = \{\boldsymbol{x} \mapsto h(\boldsymbol{x};\boldsymbol{\theta})\}_{\boldsymbol{\theta}}$.
- Machine learning algorithm (short: algorithm) – usually a model together with inductive bias and an objective function.
- Classifier – a hypothesis that outputs a categorical distribution.
- The true hypothesis – the best hypothesis given knowledge of the true underlying distribution. Usually, it can be known only for synthetic data.
- Risk – the expected error on the data distribution. For discriminative models it can often be expressed as

$$R(h, \mathcal{D}) := \mathop{\mathbf{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} L(y, h(\boldsymbol{x})), \tag{1}$$

  where L is the loss function, which is usually proportional to negative log-likelihood $\mathrm{p}(y \mid \boldsymbol{x}, \boldsymbol{\theta})$.
- Empirical risk – the error on the empirical distribution $p_{\mathbb{D}}$ (a uniform distribution over the set of observed examples $\mathbb{D}$): $R_{\mathrm{E}}(h, \mathbb{D}) := R(h, p_{\mathbb{D}})$. The objective of a machine learning algorithm is to minimize a

combination of empirical risk and structural risk, which usually represents the prior hypothesis probability of the hypothesis (parameters) $\mathrm{p}(\boldsymbol{\theta})$.

Here are some common terms related to adversarial examples defined:

- Attack – an algorithm that generates adversarial examples.
- Defense – an idea used to make an algorithm resistant to adverarial examples.
- Natural example – an example from the distribution that the training data was sampled from.
- Adversarial perturbation – the difference between an adversarial example and the natural example it is based on. An adversarial example is based on a natural example iff the it is found in the neighbourhood of the natural example.
- Targeted attack – an attack with the goal that the hypothesis outputs some desired prediction.
- Non-targeted attack – an attack with the goal that the hypothesis outputs any misprediction.
- Threat model – a set of constraints on attacks. A defense should assume a threat model under which it is meant to work.
- Adversarial robustness (short: robustness) – either the performance of a hypothesis or algorithm under some threat model or how large a perturbations must be in order to turn natural examples into adversarial examples.
- Adversarial training – training that uses adversarial examples generated with an attack besides natural examples.
- Standard training – training that uses original examples from the training set without an attack.

Because an adversarial example can be defined in more ways, depending on the usefulness of the definition, and the definition is important for making precise conclusions, the definitions of an adversarial example will be discussed in section 3.

3. Adversarial example definitions

A broadly accepted definition of an adversarial example is that it is *an input designed to fool a hypothesis into producing a misprediction*. To make it more precise and useful, the "designed" part can be replaced with the constraint that adversarial examples are close to examples with correct predictions, which serves as a distinction between adversarial examples and examples far from decision boundaries with mispredicted labels causing standard generalization error. We get the following definition.

Definition 1 (adversarial example). An adversarial example is an input for which the following holds:

1) It is close to an input with a correct prediction.
2) The hypothesis produces a misprediction.

By this definition, the set of adversarial examples is a function of the hypothesis, the true hypothesis, and a neighbourhood function. A misprediction can be defined via some distance between distributions and a threshold. In classification, a misprediction is usually defined as misclassification, i.e. the class with the highest predicted probability being different from true class.

As will be explained, a more practical, but less consistent definition is the following.

Definition 2 (practical adversarial example). A (practical) adversarial example is an input for which the following holds:

1) It is close to an input with a correct prediction.
2) The hypothesis produces a different prediction than for the input it is based on (the input from point 1).

Again, the difference between predictions can be defined via a distance between distributions. In classification, predictions of a hypothesis being different usually means that the class with the highest probability differs.

The difference from the first definition is that in the second definition knowledge of the true label is substituted with the prediction for the input with the correct prediction. This is useful for finding adversarial examples because all constraints can be known, but it is possible for a pair of natural examples differing in the label to have overlapping neighbourhoods and thus to be inside each other's neighbourhoods and thus be adversarial examples of each other. By the second definition, Even the true hypothesis can have adversarial examples close to decision boundaries. The first definition is more consistent in this sense, but then, knowing whether something is an adversarial example requires knowing the true hypothesis (i.e. it requires having solved the problem we are trying to solve with machine learning). In the following text, we will refer to the first definition as *the consistent definition* and the second one as *the practical definition*.

There are also some different or broader definitions of adversarial examples. Some authors consider out-of-distribution examples producing high-confidence predictions [12] or, most broadly, any inputs that fool the hypothesis [13].

4. The manifold hypothesis

A useful assumption in many machine learning problems is that the distribution of high-dimensional data is approximately concentrated in low-dimensional manifolds [14]. A manifold can be described as a set of points related through neighourhoods that locally resembles a euclidean space. In machine learning, the term tends to be used loosely, usually denoting a connected set of points that can be approximated well with a small number of degrees of freedom, each corresponding to a local direction of variation, embedded in a higher-dimensional space. [14] provide the following evidence. Firstly, the fact that the chance a natural example will be sampled from a simple distribution in a high-dimensional space is negligible indicates that the data distribution is highly concentrated. Secondly, we can imagine transformations that represent connections between similar examples and can be applied to traverse the manifold (e.g. for images). One of the goals of generative models is often to model the data manifold with a lower-dimensional representation. This assumption provides useful intuition for generative models, dimensionality reduction, the understanding of adversarial examples, and some adversarial defenses.

5. Properties of adversarial examples

Here some properties, phenomena and hypotheses regarding adversarial examples are described, some of which are more or less supported by evidence. Besides the hypotheses described here, there is also a quite predictive and well suppoted by evidence – the *non-robust features* hypothesis [15, 8] described in section 8, which might not be in contradiction with most of the hypotheses presented here. A broader overview can be found in e.g. Serban and Poll [16].

Rareness and closeness to natural examples. The existence of adversarial examples shows that near almost every input, there are misclassified inputs nearby. Evidence also shows that they are rare, i.e. they can not be easily found with random search in the neighbourhood of an natural example. This initially lead to the hypothesis that existence of adversarial examples is caused by high nonlinearity of deep models and that they are located in *low-probability pockets* in the data manifold [6].

Local linearity. However, Goodfellow et al. [7] have found that it is usually enough to know the locally linear behaviour (gradient with respect to the input) to reliably generate adversarial examples along directions of increasing the linear approximation. This is visualized in figure 1. They show that the direction of the perturbation matters more than the position in input space, evidencing against the low-probability pockets hypothesis. They suggest the *linearity hypothesis* whereby high local linearity of models accounts for the existence of adversarial examples. Summing small perturbations in many elements of high-dimensional inputs can produce a large change in the prediction. Su et al. [17] have shown that even modifying only 1 pixel can often be enough to cause a misclassification, indicating that models can be highly sensitive to some input elements. Tabacof and Valle [18] have found more evidence that adversarial examples span many-dimensional subsets in the input space.

Transferability. An interesting property is that adversarial examples generated for one model are often misclassified by other models and models trained on different datasets, i.e. they generalize across models and datasets [6]. This property of adversarial examples is called transferability. This suggests that most state-of-the art deep models are similarly biased. More analysis of transferebility can be found in Papernot et al. [19], Liu et al. [20], Tramèr et al. [21].

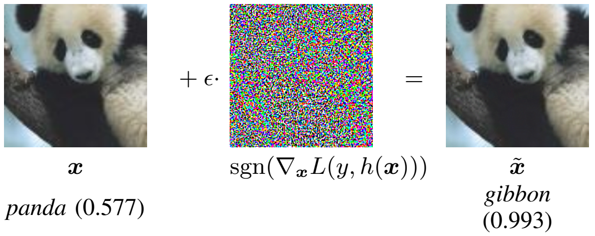

Figure 1: Generation of an adversarial example with FGSM, a single step attack. Italic words and numbers represent classes and confidences. The images are from Goodfellow et al. [7].

Universal perturbations. Moosavi-Dezfooli et al. [22] have observed that there exist perturbations that can reliably produce adversarial examples when added to almost any input and hypothesize that they exploit geometric correlations between different parts of decision boundaries.

Boundary tilting. Tanay and Griffin [23] hypothesize that adversarial examples exist when the decision boundary lies close to the submanifold of sampled data. They suggest that adversarial examples might be occurring along low-variance directions of the data where it is close to the manifold and that robustness could be improved with regularization. This is illustrated in figure 2.

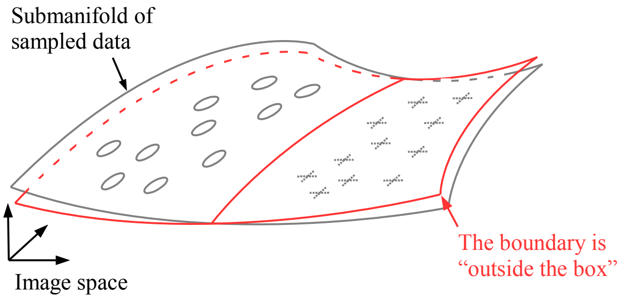

Figure 2: An illustration of boundary tilting from Tanay and Griffin [23].

High-dimensional space manifold geometry. Gilmer et al. [24] hypothesize that the existence of adversarial examples could be a naturally occurring result of the geometry of high-dimensional data manifolds. For a simple dataset with two classes consisting of examples from a pair of high-dimensional concentric spheres, they have observed that most random points in the data distribution are both correctly classified and close to a misclassified point. The authors have also given negative evidence on the hypothesis that adversarial examples are off the data manifold by showing that *on-manifold* adversarial examples can exist as well.

Adversarial examples of generative models. Adversarial examples are not just a phenomenon related to discriminative models. They have been found to exist for some generative models as well [7, 25]. An example is shown in figure 3.

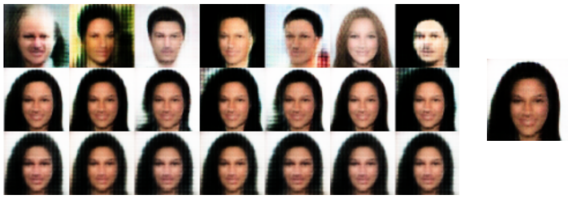

Figure 3: Reconstruction outputs for targeted attacks on a VAE-GAN model from Kos et al. [25]. The rows represent of original image reconstructions (top), reconstructions of adversarial examples generated using an attack in latent space (middle) and a VAE-loss attack (bottom). The target reconstruction is on the right.

True ambiguity of adversarial examples of robust classifers. Qualitative assesment of adversarial examples of robust classifiers suggest that they *understand data* much better. Their adversarial examples really are ambigous, i.e. generated perturbations don't look like noise but are semantically meaningful to humans [8, 26]. This is illustrated in figures 4 and 5. An interesting observation is that adversarially trained discriminative classifiers together with an iterative attack can interpolate between and generate quite realistic examples. Tsipras et al. [8] suggest a connection between the the saddle point problem of adversarial training and GAN training [27].

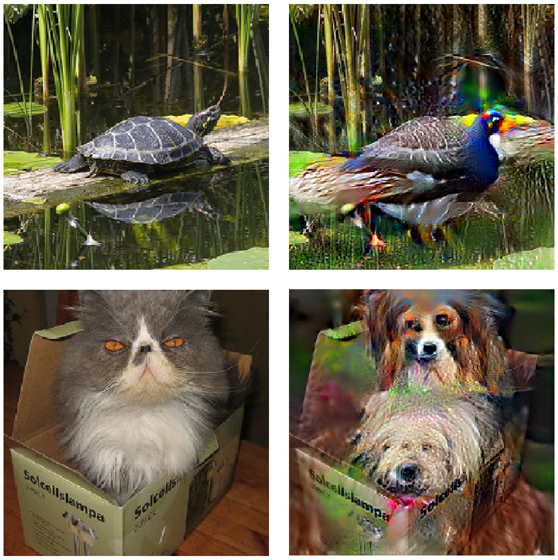

Figure 4: Original images and adversarial examples generated with a large perturbation using an iterative non-targeted attack on an adversarially trained Restricted ImageNet [8] classifier from Tsipras et al. [8].

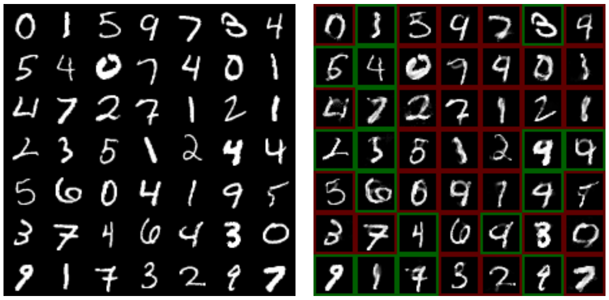

Figure 5: Clean images (left) and adversarial examples generated using an iterative non-targeted attack on a generative MNIST [28] classifier with the factorization $\mathrm{p}(\boldsymbol{z})\,\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{z})\,\mathrm{p}(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{y})$ (right) from Li [26]. The adversarial examples marked in green are successful.

Further interesting phenomena and hypotheses related to the nature of adversarial examples, adversarial robustness and generalization will be discussed in section 8.

6. Finding adversarial examples

Let $\mathbb{X}$ be the input space, and $d \in (\mathbb{X} \times \mathbb{X} \to \mathbb{R}^+)$ a *distance function* for defining similarity between inputs. For each example $\boldsymbol{x}$, we can also define its *neighbourhood* as $B_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}' \colon d(\boldsymbol{x}', \boldsymbol{x}) \leq \epsilon\}$, where ϵ is the maximum distance from the example.

Ideally, the neighbourhood of an example $\boldsymbol{x}$ should be the set of *perceptually similar* examples that all belong to the same class as $\boldsymbol{x}$ (their true class may be at most ambiguous), but it is hard to define such a neighbourhood (as it requires knowing the true model). A practical and common way of defining the neighbourhood function for images is to have d be a L^p distance where p is usually ∞ or 2. Note that, if an example is very near the true class boundary, such a neighbourhood may contain examples belonging to another class.

6.1. Attack objectives

Finding an adversarial example can be defined as a constrained optimization problem of maximizing some loss with respect to the input with the constraint that the input is in the neighbourhood $B_\epsilon(\boldsymbol{x})$:

$$\tilde{\boldsymbol{x}} = \operatorname*{arg\,max}_{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})} L(y, h(\boldsymbol{x}')), \tag{2}$$

where y is the true label. Let $\hat{h}(\boldsymbol{x}) \coloneqq \operatorname{arg\,max}_y h(\boldsymbol{x})_{[y]}$ denote the function that assigns the label with the highest probability to an input. An objective can also be to find the $\tilde{\boldsymbol{x}}$ closest to $\boldsymbol{x}$ such that the classifier misclassifies it [29]:

$$\tilde{\boldsymbol{x}} = \operatorname*{arg\,min}_{\boldsymbol{x}' \colon \boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}) \wedge \hat{h}(\boldsymbol{x}') \neq y} d(\boldsymbol{x}', \boldsymbol{x}). \tag{3}$$

The described objectives, where it only matters that the adversarial example is misclassified, are objectives for *non-targeted adversarial attacks*. There are also *targeted adversarial attacks*, where the objective is to create an adversarial example such that the model classifies it as some desired target. Targeted attack objectives corresponding to equations (2) and (3) are:

$$\tilde{\boldsymbol{x}} = \underset{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})}{\arg\min}\, L(y_\text{a}, h(\boldsymbol{x}')), \tag{4}$$

$$\tilde{\boldsymbol{x}} = \underset{\boldsymbol{x}' :\, \boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}) \wedge \hat{h}(\boldsymbol{x}') = y_\text{a}}{\arg\min}\, d(\boldsymbol{x}', \boldsymbol{x}), \tag{5}$$

where y_a denotes the adversarial target label. The difference in equation (4) is loss minimization and the adversarial target label instead of the true label. The difference in equation (5) is the condition that the predicted labels equals the adversarial target instead of differing from the true label.

Non-targeted adversarial examples can also be generated without knowledge of the true label. Instead of the true label y, the predicted label $\hat{h}(\boldsymbol{x})$ can be used in equations (2) and (3). Such adversarial examples are called *virtual adversarial examples*. Miyato et al. [30], Kurakin et al. [31] propose the following attack objective for use in semi-supervised learning:

$$\tilde{\boldsymbol{x}} = \underset{\boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})}{\arg\min}\, D((\underline{y} \mid \boldsymbol{x}, \boldsymbol{\theta}), (\underline{y} \mid \underline{\boldsymbol{x}} = \boldsymbol{x}', \boldsymbol{\theta})), \tag{6}$$

where D is some non-negative function that represents distance between distributions.

For adversarial training, non-targeted attacks should be preferred due to the *label-leaking* phenomenon [31] where the learned classifier can overfit to adversarial examples and perform better on them than on natural examples, especially with attacks with a small number of iterations. For robustness evaluation with datasets that have many similar classes, non-targeted attacks can too easily fool the classifier and targeted attacks give more meaningful evaluation results [32].

6.2. Common attacks

Being that finding adversarial examples is a constrained optimization problem, general gradient-based and black-box optimization algorithms can be used for attacks. Additionally, sometimes techniques specific to some potential defense and machine learning algorithm have to be used.

Some commonly known gradient-based attacks for finding adversarial examples are the following (using the notation from section 6):

- Box-constrained L-BFGS – Szegedy et al. [6] propose to minimize $c\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_2^2 + L(y, h(\tilde{\boldsymbol{x}}))$ with the constraint $\tilde{\boldsymbol{x}} \in [0, 1]$ with L-BFGS, a quasi-Newton optimization method. c is a number obtained via line-search that yields adversarial examples of minimum distance.
- Fast gradient sign method (FGSM) – an attack proposed by Goodfellow et al. [7] that requires a single gradient computation:
$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \epsilon \nabla_{\boldsymbol{x}} L(y, h(\boldsymbol{x})), \tag{7}$$
where ϵ is the L^∞-norm of the perturbation. For L^2-constrined perturbations, Miyato et al. [30] propose L^2 normalization instead of the sign function.
- DeepFool – an iterative non-targeted attack proposed by Moosavi-Dezfooli et al. [29] that in each step finds the optimal solution to a linear approximation of a loss in the L^2 ball $B_\epsilon(\boldsymbol{x})$ using the gradient in the current adversarial input. It is faster and finds smaller perturbations than L-BFGS and stronger than FGSM.
- Projected gradient descent (PGD) [9] or basic iterative method (BIM) [33] – an iterative gradient-based algorithm with random initialization [9] of the perturbation from within the $B_\epsilon(\boldsymbol{x})$ at the start and steps in the direction of the gradient sign:
$$\tilde{\boldsymbol{x}}_i = \mathit{\Pi}_{B_\epsilon(\boldsymbol{x})}\big(\tilde{\boldsymbol{x}}_{i-1} + \alpha \operatorname{sgn}\big(\nabla_{\tilde{\boldsymbol{x}}_{i-1}} L(y, h(\tilde{\boldsymbol{x}}_{i-1}))\big)\big). \tag{8}$$
α is the step size, and $\mathit{\Pi}_{B_\epsilon(\boldsymbol{x})}$ is the projection into the L^p ϵ-ball around $\boldsymbol{x}$.
- Carlini-Wagner (CW) attacks – Carlini and Wagner [34] propose attacks with similar minimal perturbation objectives as Szegedy et al. [6] and Moosavi-Dezfooli et al. [29]. They modify the loss function and, to enable unconstrained optimization, they introduce change of variables $\boldsymbol{\delta} = \frac{1}{2}(\tanh(\boldsymbol{w}) + \boldsymbol{1}) - \boldsymbol{x}$, which limits the perturbation $\boldsymbol{\delta}$ to the interval $[0, 1]$. With the PGD (BIM) attack, it is currently probably one of the 2 strongest attacks.

7. Improving adversarial robustness

There are different approaches (*defenses*) trying to improve adversarial robustness, most of which have been shown to actually be non-robust, but had appeared robust because they intentionally or unintentionally caused attacks that they were evaluated on to be unable to find adversarial examples [35, 32, 36, 34]. Thus, it is important to put as much effort as needed to correctly evaluate robustness, i.e. get an as low as possible upper bound on robustness. [37] is a recent helpful overview on evaluation of robustness.

A broad overview of many defenses can be found in Serban and Poll [16]. To give a few examples, some approaches use generative models to approximately project inputs to a learned data manifold (e.g. Samangouei et al. [38]), some approaches are based on limiting the Lipschitz constant of the model to limit sensitivity to small input perturbations by regularization and model modification (e.g. Qian and Wegman [39]), some research is looking into ways of guaranteeing robustness (e.g. Cohen et al. [40]).

7.1. Adversarial training and empirical adversarial risk

The only defense currently believed to be effective according to Athalye et al. [32] is adversarial training [7] with a strong attack [9], where the model is trained on adversarial examples as well as natural examples.
Madry et al. [9] define what can be called *empirical adversarial risk* by allowing the worst-case attack to modify each the input in the empirical risk expression:

$$R_{\mathrm{EA}}(h, \mathbb{D}) := \mathop{\mathbf{E}}_{(\boldsymbol{x},y)\sim p_{\mathbb{D}}} \left(\max_{\tilde{\boldsymbol{x}} \in B_\epsilon(\boldsymbol{x})} L(y, h(\tilde{\boldsymbol{x}})) \right). \qquad (9)$$

They propose PGD for the attack during training and PGD with as large a number of iterations as necessary to approximate the worst-case adversary, i.e. get a better upper bound on robustness.
Still, adversarially trained models are not robust to attacks with weaker constraints than those used for training [41]. Furthermore, because adversarial examples are generated and robustness is evaluated according to the practical definition of an adversarial example and using L^p distance as a non-ideal approximation of perceptual similarity, performance is affected [9, 8] and there can exist misclassified examples among which are *invariance-based* adversarial examples [42].

8. Adversarial robustness and generalization

Based on the practical definition of an adversarial example or similar definitions, experimental [9, 10] and theoretical evidence [8] suggests that there is a trade-off between robustness and standard generalization for current models. Here are some recent discoveries related to adversarial robustness and generalization presented. Robustness to distributional shift and corruptions also seems to be quite relevant [43, 4], but it will not be discussed here.

8.1. A trade-off between robustness and generalization

Madry et al. [9], Su et al. [17], Tsipras et al. [8] and others have empirically observed that adversarial robustness with current algorithms requires more capacity and negatively affects generalization. Su et al. [17] have observed that older convolutional architectures with no shortcut connections, like AlexNet [44] and VGG [45] seem to be inherently more robust than architectures like ResNet [46], DenseNet [47] and MobileNet [48] and NASNets [49] with standard training. Madry et al. [9] has observed that with adversarial training, more model capacity is required and natural test set performance is reduced. Furthermore, Tsipras et al. [8] have, based on the practical definition of an adversarial example, theoretically demonstrated an aspect of the trade-off. Another reason that affects performance suggested by them is that salient features might be harder to learn and that algorithms rely on highly predictive but *non-robust* features.

8.2. Non-robust features

Based on some ideas from Tsipras et al. [8], Ilyas et al. [15] propose an interesting and experimentally well supported hypothesis on the nature of features that well-generalizing non-robust classifiers learn. They show that existence of adversarial examples can be directly attributed to existence of non-robust features, "features derived from patterns in the data that are highly predictive but brittle and incomprehensible to humans".
They demonstrate the predictiveness of non-robust features by:

1) constructing a dataset $\mathbb{D}_{\mathrm{NR}}$ where approximately the only useful features are non-robust features by turning inputs of the original dataset $\mathbb{D}$ into adversarial examples for a classifier that was trained standardly and relabeling them with the adversarial label,
2) training a new classifier on the non-robust dataset $\mathbb{D}_{\mathrm{NR}}$,
3) testing the new classifier on the original test set where it achieves performance close to the original classifier and lower robustness.

In another experiment, they try to remove non-robust-features from inputs with the help of an adversarially trained classifier. A new classifier trained on the dataset with removed non-robust features achieves a bit lower performance and robustness not as high as the adversarially trained classifier but quite significant compared to the standardly trained classifier. Results of these experiments with the CIFAR-10 dataset [50] are shown in figure 6.

8.3. Training with on-manifold adversarial examples

Stutz et al. [11] challenge the hypothesis that there is a trade-off between robustness and generalization. They hypothesize that most adversarial examples come from directions orthogonal to the learned class manifolds and that training adversarial examples (as per a definition similar to definition 1) limited to the known or learned class manifolds (on-manifold adversarial examples) can improve generalization. They conduct experiments with a synthetic dataset with known class-invariant transformations and datasets with small images that support the hypothesis that generalization can be improved with adversarial training with on-manifold adversarial examples.
Class manifolds to which adversarial examples are restrited and on-manifold and off-manifold adversarial examples are illustrated in figure 7.
In one of the experiments Stutz et al. [11] construct a synthetic dataset with a known manifold (geometric transformations of letters) in order to be able to generate exactly on-manifold adversarial examples by modifying parameters of the geometric transformations. With this

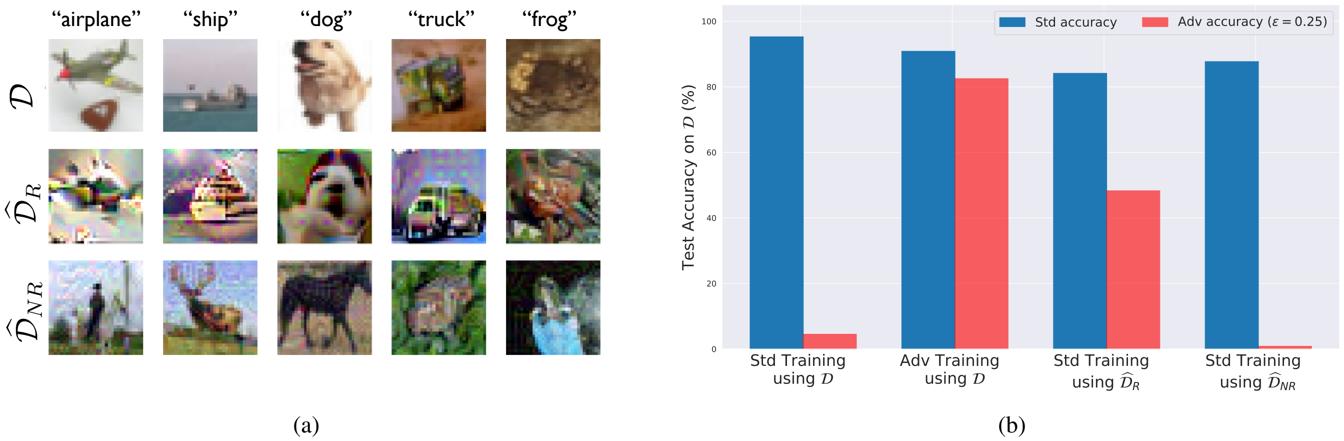

Figure 6: (a) Random samples from variants of the CIFAR-10 training set: the original training set; the *robust training set* $\mathbb{D}_{\mathrm{R}}$, restricted to features used by a robust model; and the *non-robust training set* $\mathbb{D}_{\mathrm{NR}}$, restricted to features relevant to a standard model (labels appear incorrect to humans). (b) Standard and robust accuracy on the CIFAR-10 test set ($\mathbb{D}$) for models trained with standard training, adversarial training, and standard training on datasets $\mathbb{D}_{\mathrm{R}}$ (robust) and $\mathbb{D}_{\mathrm{NR}}$ (non-robust). Adapted from Ilyas et al. [15].

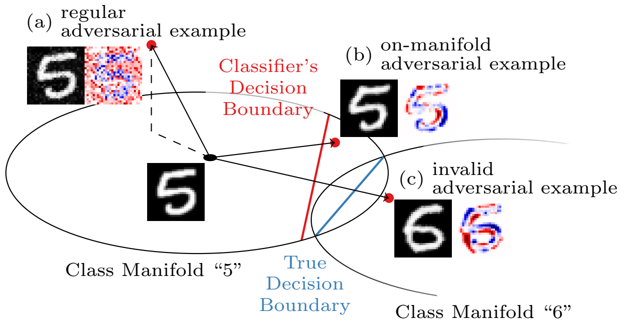
(a) regular adversarial example
Classifier's Decision Boundary
(b) on-manifold adversarial example
(c) invalid adversarial example
Class Manifold "5"
True Decision Boundary
Class Manifold "6"

Figure 7: An illustration by Stutz et al. [11] of class manifolds (classes "5" and "6") with a regular (off-manifold) adversarial example and an on-manifold adversarial example.

dataset, they succeed in improving generalization and on-manifold robustness[1] with adversarial training. In other experiments they use EMNIST [51], FashionMNIST [52] and CelebA [53]. In order to better approximate class manifolds and disable leaving the manifold of a class when an adversarial example is generated for adversarial training, they first train one variational autoencoder (VAE-GAN) per class. They perform training and evaluation analogously to the experiment with synthetic data by allowing the attack to perturb the latent representation of the autoencoder corresponding to the correct class. They measure positive correlation between robustness to on-manifold adversarial examples and generalization. They observe worse quality of on-manifold adversarial examples for the more complex dataset CelebA due to worse approximation quality of their VAE-GAN-s.

9. Conclusion

This paper presents an overview of ideas related to the existence adversarial examples, hypotheses on their existence and some of their properties. It describes general principles regarding adversarial attacks, defenses and robustness evaluation and presents examples of such algorithms. Finally, some recent discoveries and hypotheses regarding the relation between robustness and generalization are explored. Some recent results [11] suggest that finding ways of improving both robustness generalization might be an interesting research direction to explore.

References

[1] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *CoRR*, vol. abs/1610.02136, 2016. [Online]. Available: http://arxiv.org/abs/1610.02136

[2] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 1180–1189. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045244

[3] A. M. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence

[1] By the authors' definition of an adversarial example, which is similar to the consistent definition (definition 1), except for that there is no closeness constraint, making it equivalent to the definition of a misclassified example, on-manifold robustness essentially boils down to generalization.

predictions for unrecognizable images." in *CVPR*. IEEE Computer Society, 2015, pp. 427–436. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2015.html#NguyenYC15

[4] D. Hendrycks and T. G. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *CoRR*, vol. abs/1903.12261, 2019. [Online]. Available: http://arxiv.org/abs/1903.12261

[5] L. Engstrom, D. Tsipras, L. Schmidt, and A. Madry, "A rotation and a translation suffice: Fooling cnns with simple transformations," *CoRR*, vol. abs/1712.02779, 2017. [Online]. Available: http://arxiv.org/abs/1712.02779

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013. [Online]. Available: http://arxiv.org/abs/1312.6199

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014. [Online]. Available: http://arxiv.org/abs/1412.6572

[8] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," 2018, cite arxiv:1805.12152. [Online]. Available: http://arxiv.org/abs/1805.12152

[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *CoRR*, vol. abs/1706.06083, 2017. [Online]. Available: http://arxiv.org/abs/1706.06083

[10] D. Su, H. Zhang, H. Chen, J. Yi, P. Chen, and Y. Gao, "Is robustness the cost of accuracy? - A comprehensive study on the robustness of 18 deep image classification models," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*, 2018, pp. 644–661. [Online]. Available: https://doi.org/10.1007/978-3-030-01258-8_39

[11] D. Stutz, M. Hein, and B. Schiele, "Disentangling adversarial robustness and generalization," *CoRR*, vol. abs/1812.00740, 2018.

[12] Y. Gal and L. Smith, "Sufficient Conditions for Idealised Models to Have No Adversarial Examples: a Theoretical and Empirical Study with Bayesian Neural Networks," *ArXiv e-prints*, Jun. 2018.

[13] T. B. Brown, N. Carlini, C. Zhang, C. Olsson, P. F. Christiano, and I. J. Goodfellow, "Unrestricted adversarial examples," *CoRR*, vol. abs/1809.08352, 2018. [Online]. Available: http://arxiv.org/abs/1809.08352

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[15] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," 2019, cite arxiv:1905.02175. [Online]. Available: http://arxiv.org/abs/1905.02175

[16] A. C. Serban and E. Poll, "Adversarial examples - A complete characterisation of the phenomenon," *CoRR*, vol. abs/1810.01185, 2018. [Online]. Available: http://arxiv.org/abs/1810.01185

[17] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *CoRR*, vol. abs/1710.08864, 2017. [Online]. Available: http://arxiv.org/abs/1710.08864

[18] P. Tabacof and E. Valle, "Exploring the space of adversarial images," in *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*, 2016, pp. 426–433. [Online]. Available: https://doi.org/10.1109/IJCNN.2016.7727230

[19] N. Papernot, P. D. McDaniel, and I. J. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples." *CoRR*, vol. abs/1605.07277, 2016. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1605.html#PapernotMG16

[20] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. [Online]. Available: https://openreview.net/forum?id=Sys6GJqxl

[21] F. Tramèr, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel, "The space of transferable adversarial examples," *CoRR*, vol. abs/1704.03453, 2017. [Online]. Available: http://arxiv.org/abs/1704.03453

[22] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," *CoRR*, vol. abs/1610.08401, 2016. [Online]. Available: http://arxiv.org/abs/1610.08401

[23] T. Tanay and L. D. Griffin, "A boundary tilting persepective on the phenomenon of adversarial examples," *CoRR*, vol. abs/1608.07690, 2016. [Online]. Available: http://arxiv.org/abs/1608.07690

[24] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. J. Goodfellow, "Adversarial spheres," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*, 2018. [Online]. Available: https://openreview.net/forum?id=SkthlLkPf

[25] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," in *2018 IEEE Security and Privacy Workshops (SPW)*, May 2018, pp. 36–42.

[26] Y. Li, "Are generative classifiers more robust to adversarial attacks?" *CoRR*, vol. abs/1802.06552, 2018. [Online]. Available: http://arxiv.org/abs/1802.06552

[27] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in

Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969125

[28] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[29] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *CVPR*. IEEE Computer Society, 2016, pp. 2574–2582.

[30] T. Miyato, S. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *CoRR*, vol. abs/1704.03976, 2017. [Online]. Available: http://arxiv.org/abs/1704.03976

[31] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *CoRR*, vol. abs/1611.01236, 2016. [Online]. Available: http://arxiv.org/abs/1611.01236

[32] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 274–283. [Online]. Available: http://proceedings.mlr.press/v80/athalye18a.html

[33] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *CoRR*, vol. abs/1607.02533, 2016. [Online]. Available: http://arxiv.org/abs/1607.02533

[34] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, 2017, pp. 39–57. [Online]. Available: https://doi.org/10.1109/SP.2017.49

[35] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, ser. AISec '17. New York, NY, USA: ACM, 2017, pp. 3–14. [Online]. Available: http://doi.acm.org/10.1145/3128572.3140444

[36] J. Uesato, B. O'Donoghue, P. Kohli, and A. van den Oord, "Adversarial risk and the dangers of evaluating against weak attacks," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018, pp. 5032–5041. [Online]. Available: http://proceedings.mlr.press/v80/uesato18a.html

[37] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. J. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," *CoRR*, vol. abs/1902.06705, 2019. [Online]. Available: http://arxiv.org/abs/1902.06705

[38] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-gan: Protecting classifiers against adversarial attacks using generative models," *CoRR*, vol. abs/1805.06605, 2018. [Online]. Available: http://arxiv.org/abs/1805.06605

[39] H. Qian and M. N. Wegman, "L2-nonexpansive neural networks," *CoRR*, vol. abs/1802.07896, 2018. [Online]. Available: http://arxiv.org/abs/1802.07896

[40] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 1310–1320. [Online]. Available: http://proceedings.mlr.press/v97/cohen19c.html

[41] L. Schott, J. Rauber, M. Bethge, and W. Brendel, "Towards the first adversarially robust neural network model on mnist," *CoRR*, vol. abs/1805.09190, 2018. [Online]. Available: http://arxiv.org/abs/1805.09190

[42] J. Jacobsen, J. Behrmann, N. Carlini, F. Tramèr, and N. Papernot, "Exploiting excessive invariance caused by norm-bounded adversarial robustness," *CoRR*, vol. abs/1903.10484, 2019. [Online]. Available: http://arxiv.org/abs/1903.10484

[43] J. Gilmer, N. Ford, N. Carlini, and E. Cubuk, "Adversarial examples are a natural consequence of test error in noise," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 2280–2289. [Online]. Available: http://proceedings.mlr.press/v97/gilmer19a.html

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-netwo
pdf

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[47] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available:

http://arxiv.org/abs/1608.06993

[48] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[49] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 8697–8710. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Zoph_Learning_Transferable_Architectures_CVPR_2018_paper.html

[50] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[51] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "Emnist: an extension of mnist to handwritten letters," *arXiv preprint arXiv:1702.05373*, 2017.

[52] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.

[53] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, 12 2015.