A Brief Introduction to Adversarial Examples

Aleksander Mądry, Ludwig Schmidt · *Jul 6, 2018*

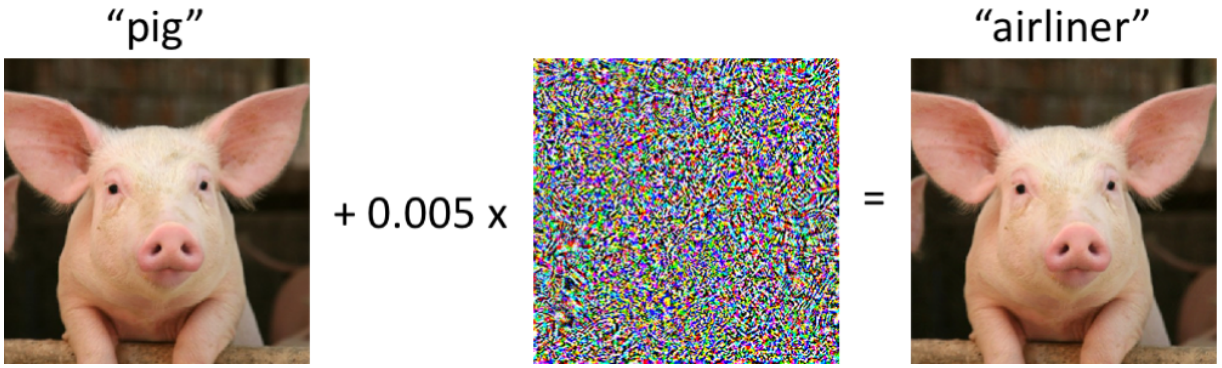
9 minute read

Over the past few years, adversarial examples have received a significant amount of attention in the deep learning community. In this blog post, we want to share our high-level perspective on this phenomenon and how it fits into a larger question of robustness in machine learning. In subsequent posts, we plan to delve deeper into the topics that we will only briefly touch on today.

Adversarial Examples: An Intriguing Phenomenon

To set the stage for our discussion, let us briefly introduce adversarial examples. By now, most researchers in ML have probably seen a picture like the following:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



On the left, we have an image of a pig that is correctly classified as such by a state-of-the-art convolutional neural network. After perturbing the image slightly (every pixel is in the range [0, 1] and changed by at most 0.005), the network now returns class “airliner” with high confidence. Such attacks on trained classifiers have been studied since at least 2004 (link), and there has already been work on adversarial examples for image classification in 2006 (link). This phenomenon has then received considerably more attention starting in 2013 when it was pointed out that neural networks are also vulnerable to such attacks (see here and here). Since then, many researchers have proposed ways to construct adversarial examples, as well as methods to make classifiers more robust against adversarial perturbations. It is important to keep in mind though that we do not need to go to neural networks to observe adversarial examples.

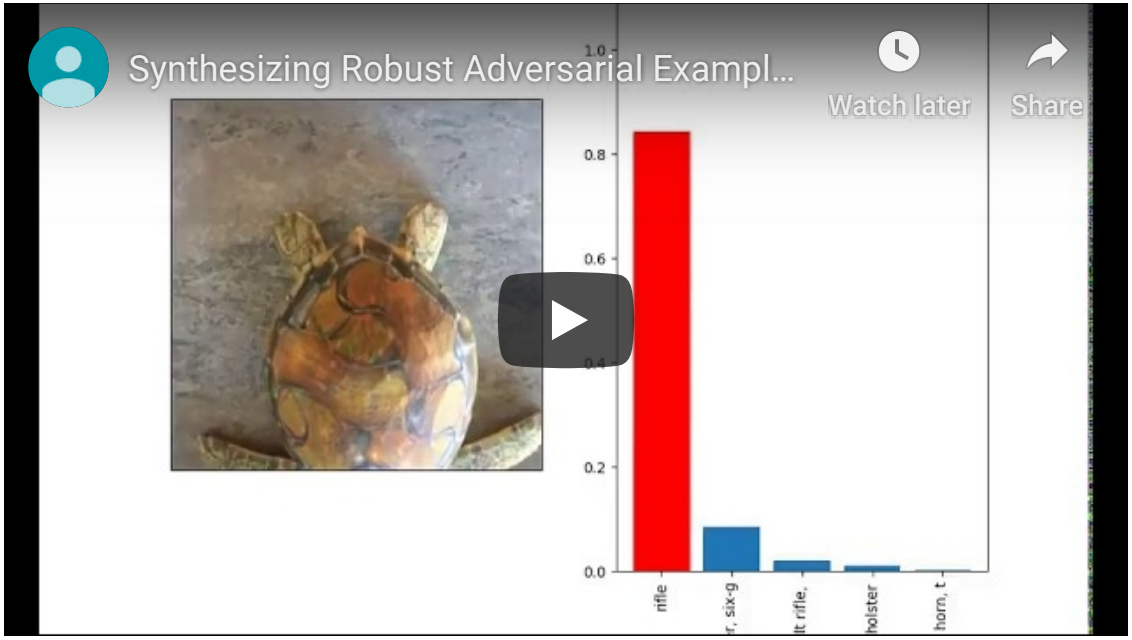
How Robust Are Adversarial Examples?

Seeing the airliner-pig above might be a bit disturbing at first. However, one should note that the underlying classifier (an Inception-v3 network) is not as fragile as it might seem. While the network misclassifies the perturbed pig with high confidence,

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

this occurs only for specifically crafted perturbations - the network is significantly more robust to random perturbations of similar magnitude. So a natural question is whether it is actually the adversarial perturbations that are fragile. If they crucially rely on precise control over all input pixels, adversarial examples become less of a concern for classifying images in real-world settings.

Recent work shows that this is not the case: the perturbations can be made robust to various channel effects in concrete physical scenarios. For instance, you can print adversarial examples with a standard office printer so that pictures of them taken with a smartphone camera are still misclassified. It is also possible to create stickers that cause neural networks to misclassify various real-world scenes (for instance, see link1, link2, and link3). Finally, researchers recently 3D-printed a turtle so that a standard Inception network misclassifies it as a rifle from almost all viewpoints:



Create PDF in your applications with the Pdfcrowd HTML to PDF API
PDFCROWD

Constructing Misclassification Attacks

How do you construct such adversarial perturbations? While there is now a wide range of approaches, optimization offers a unifying view on these different methods. As we all know, training a classifier is often formulated as finding model parameters θ that minimize an empirical loss function for a given set of samples $x_1, \ldots, x_n$:

$$\min_{\theta} \sum_{x} \text{loss}(x, \theta).$$

So to cause a misclassification for a fixed model θ and "benign" input x, a natural approach is to find a bounded perturbation δ such that the loss on $x + \delta$ is as large as possible:

$$\max_{\delta} \text{loss}(x + \delta, \theta).$$

Starting from this formulation, many methods for crafting adversarial examples can be seen as different optimization algorithms (single gradient steps, projected gradient descent, etc.) for different constraint sets (small ℓ_∞-norm perturbation, few pixel changes, etc.). You can find a few examples in the following papers: link1, link2, link3, link4, and link5.

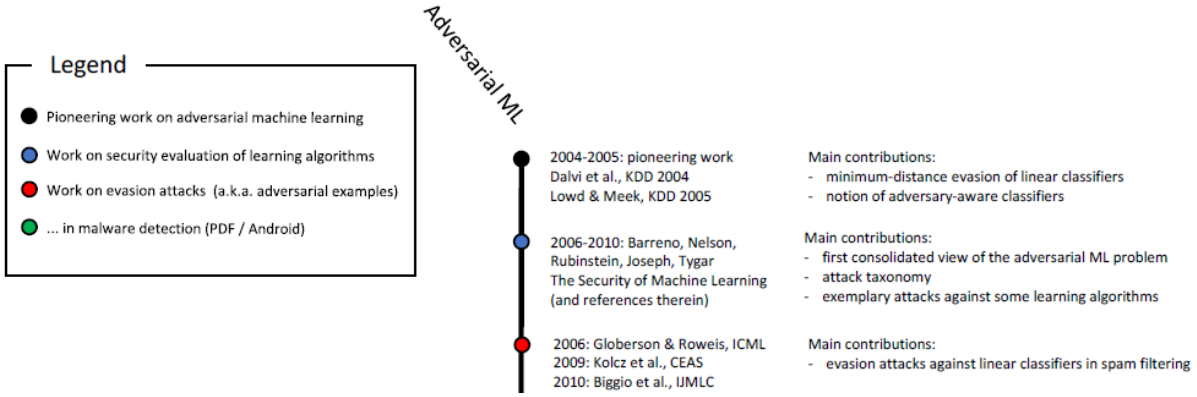
As outlined above, many successful methods for generating adversarial examples work with a fixed target classifier. Hence an important question is whether the perturbations only affect the specific target model. Interestingly, this is not the case. For many perturbation methods, the resulting adversarial examples *transfer* across

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

classifiers trained with different random seeds or different model architectures. Moreover, it is possible to create adversarial examples with only limited access to the target model (sometimes called black-box attacks). For instance, see these five papers: link1, link2, link3, link4, and link5.

Beyond Images

Adversarial examples are not limited to image classification. Similar phenomena occur in speech recognition, question answering systems, reinforcement learning, and other tasks. As mentioned before, studying adversarial examples goes back more than a decade:



Beginning of the timeline in adversarial machine learning. Click on the picture to see the full timeline, or go to Figure 6 in this survey.

In addition, security-related applications are a natural context for studying adversarial aspects of machine learning. If an attacker can fool a classifier to think that its malicious input (e.g., a spam message or a virus) is actually benign, they can render an ML-based spam detector or anti-virus scanner ineffective. It is worth

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

noting that these are not only academic considerations. For instance, the Google Safebrowsing team published a multi-year study about evasions of their malware detection systems already in 2011. Also, see this recent post on adversarial examples in the context of GMail spam filtering.

Beyond Security

The security perspective has clearly dominated the recent work on adversarial examples. Although this is a valid viewpoint, we believe that adversarial examples should be seen in a broader context.

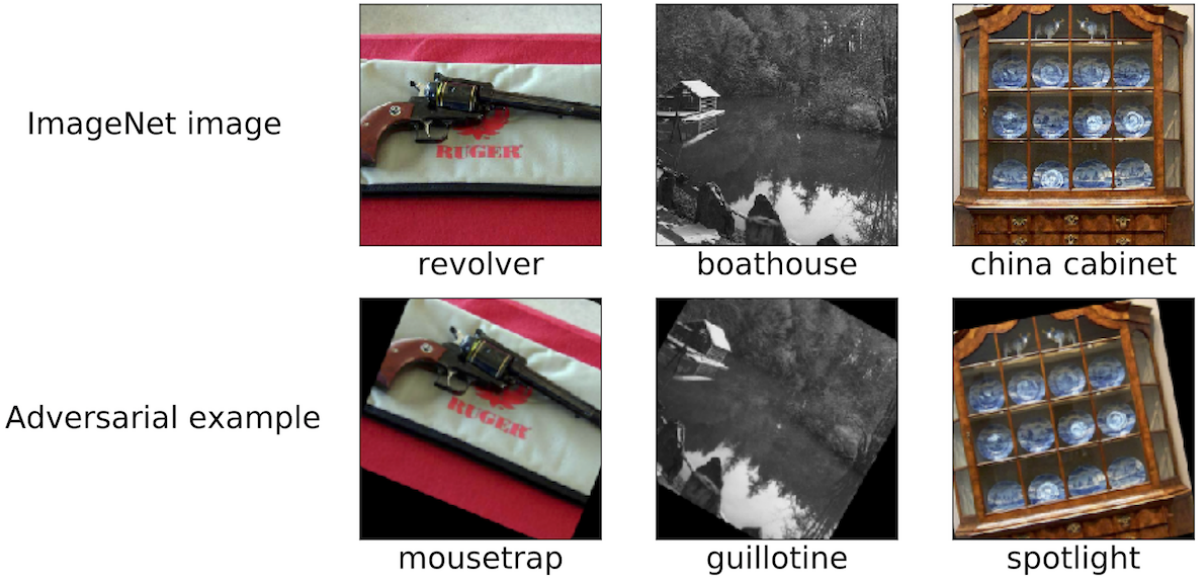
Robustness

First and foremost, adversarial examples are an issue of robustness. Before we can meaningfully discuss the security properties of a classifier, we need to be certain that it achieves good accuracy in a robust way. After all, if we want to deploy our trained models in real-world scenarios, it is crucial that they exhibit a large degree of robustness to changes in the underlying data distribution, regardless of whether these changes correspond to truly malicious tampering or merely natural fluctuations.

In this context, adversarial examples can be a useful diagnostic tool for assessing such robustness of an ML-based system. In particular, the adversarial approach allows us to go beyond the standard evaluation protocol of running a trained classifier on a carefully curated (and usually static) test set.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

This can lead to startling conclusions. For instance, it turns out that we don't actually have to resort to sophisticated optimization methods for constructing adversarial examples. In recent work, we show that state-of-the-art image classifiers are surprisingly susceptible to small, adversarially chosen translations or rotations. (See here and here for other work on this topic.)



So even if we don't worry about say ℓ_∞-perturbations, we often still care about robustness to rotations and translations. More broadly, we need to understand the robustness properties of our classifiers before we can integrate them as truly reliable components into larger systems.

Understanding Classifiers

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In order to understand how a trained classifier works, finding explicit examples of its successes and failures is essential. Here, adversarial examples illustrate that trained neural networks often do not conform to our human intuition of what it means to "learn" a given concept. This is especially relevant in deep learning, where claims of biologically plausible algorithms and human-level performance are frequent (e.g., see here, here, or here). Adversarial examples clearly challenge this view in multiple settings:

- In image classification, changing pixels by a tiny amount or slightly rotating the image hardly impacts a human's ability to determine the correct class. Nevertheless, such changes can completely throw off state-of-the-art classifiers. Putting objects in unusual places (e.g., sheep in a tree), also quickly shows that neural networks interpret scenes differently from humans.
- Inserting the right words in a piece of text can significantly confuse current question answering systems, even if this insertion does not change the meaning of the text to a human.
- This recent article demonstrates the limits of Google Translate via thoughtfully selected pieces of text.

In all three cases, adversarial examples allow us to probe how our current models work and highlight regimes where they behave quite differently from how a human would.

Security

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Finally, adversarial examples are indeed a security concern in domains where machine learning achieves sufficient "benign" accuracy. Up to a few years ago, tasks such as image classification were still quite far from satisfactory performance, and consequently security was only a secondary concern. After all, the security of an ML-based system only matters if it achieves sufficiently high accuracy on benign inputs to begin with. Otherwise, we often cannot rely on its prediction anyway.

Now that classifiers achieve significantly higher accuracy in various domains, deploying them in security-sensitive scenarios is within reach. If we want to do so in a responsible way, it is important to investigate their security properties. But we should maintain a holistic approach to the security question. Some features such as pixels are much easier to tamper with than other sensor modalities or categorical features and meta-data. In the end, the best defense measure might involve relying on features that are hard or even impossible to modify.

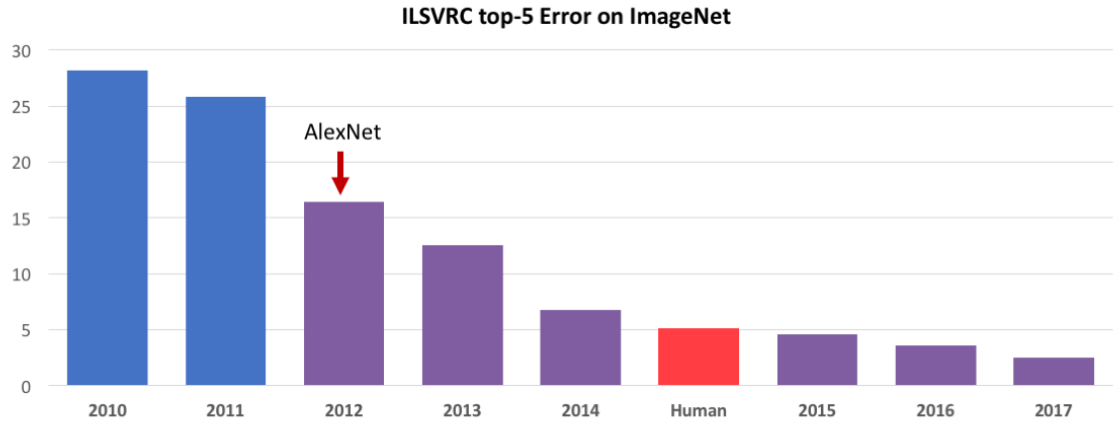
Summary: Not (Quite?) There Yet

While we have seen impressive progress in machine learning over the past few years, we should be cognizant of the limitations our tools still possess. This includes a broad range of issues (e.g., fairness, privacy, or feedback effects), with robustness being one of the key concerns. Human perception and cognition are robust to a vast range of nuisance perturbations in the real world. Yet adversarial examples show that deep networks are currently far from achieving the same level of robustness.

As a concrete example, let's again consider image classification since it has been one of the highlights in recent deep learning. On the one hand, state-of-the-art models

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

now achieve human-level accuracy on challenging tasks such as Imagenet classification. On the other hand, the adversarial perspective shows that good performance on specific tasks does not imply that we have built image classifiers that are as reliable as humans. This gap also occurs in other areas such as speech recognition, where accents or noisy environments are still significant obstacles for deep networks.



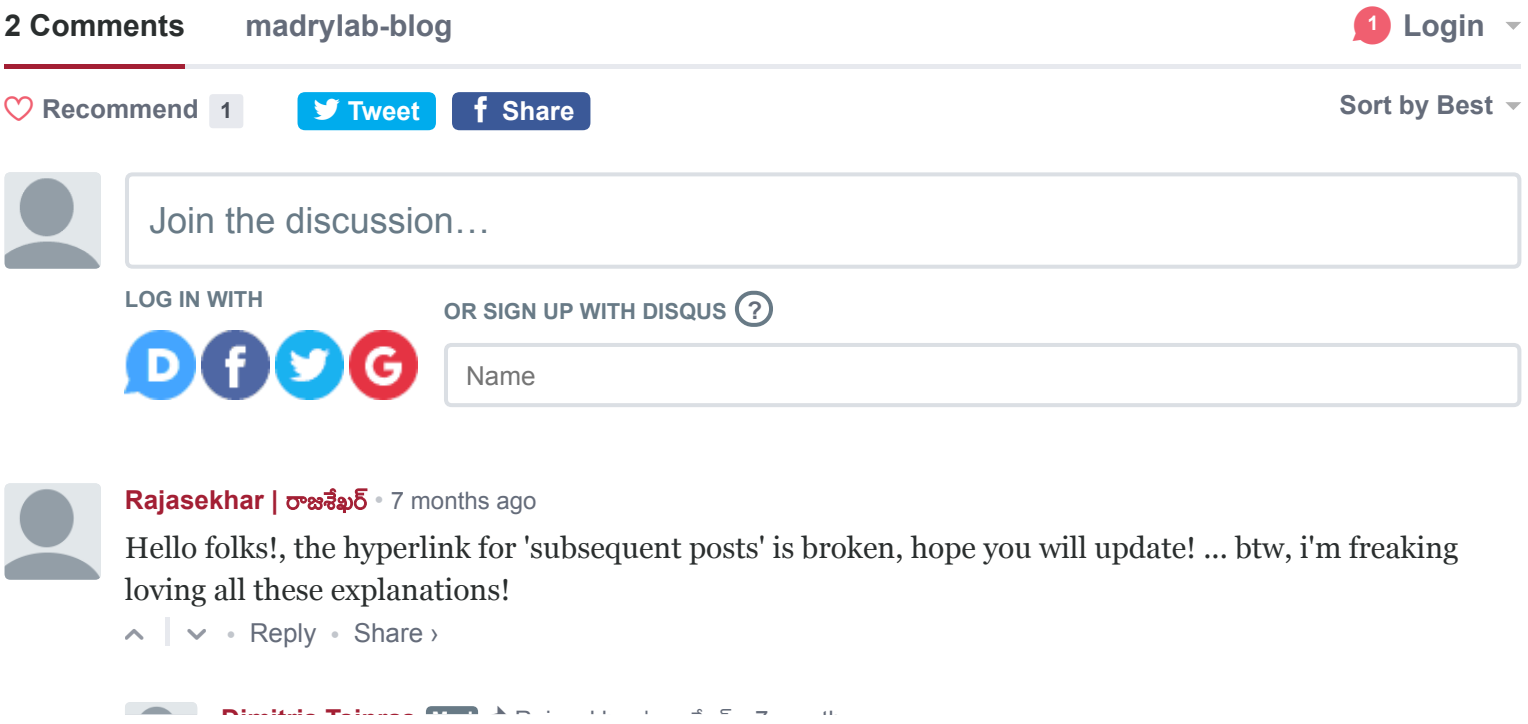
To summarize, we belive that adversarial examples are relevant beyond security aspects of machine learning and also constitute a **diagnostic framework** for evaluating trained models. In contrast to standard evaluation procedures, the adversarial approach goes beyond a static test set and allows us to reveal potentially non-obvious weaknesses. If we want to understand the reliability of current machine learning, it is thus important to explore recent progress also from this adversarial perspective (with appropriately chosen adversaries). As long as our classifiers are susceptible to small changes between train and test distribution, achieving

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

comprehensive robustness guarantees will be out of reach. After all, the goal is to create models that are not only secure, but also agree with our intuition of what it means to "learn" a task so that they are reliable, safe, and easy to deploy in a variety of environments.

In subsequent posts, we will delve into the details and discuss the underpinnings of this issue.

Subscribe to our RSS feed.
Spead the word:



Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Dimitris Tsipras Mod ↷ Rajasekhar | రాజశేఖర్ • 7 months ago

Oh thanks for pointing it out! Fixed. Thank you for your kind words :)

1 ^ | v • Reply • Share ›

ALSO ON **MADRYLAB-BLOG**

A Closer Look at Deep Policy Gradients (Part 2: Gradients and Values)

6 comments • 7 months ago

andyilyas — Thanks for the kind comment!You are right, equation (3) had a small typo, rather than using the current value for the previous state, we

How does Batch Normalization Help Optimization?

3 comments • 8 months ago

Jonathan R. Williford — Thanks for your fast response!

Training Robust Classifiers (Part 2)

7 comments • 7 months ago

Dimitris Tsipras — Exactly.

Training Robust Classifiers (Part 1)

8 comments • a year ago

Dimitris Tsipras — Sure, but this is not an issue if for a given maximizer the function is continuously differentiable, right? This is true for our models as

Subscribe Add Disqus to your site Disqus' Privacy Policy DISQUS

Theme available on GitHub.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD