MARCH 2021

Key Concepts in AI Safety: Robustness and Adversarial Examples

CSET Issue Brief

AUTHORS

Tim G. J. Rudner

Helen Toner

This paper is the second installment in a series on "AI safety," an area of machine learning research that aims to identify causes of unintended behavior in machine learning systems and develop tools to ensure these systems work safely and reliably. The first paper in the series, "Key Concepts in AI Safety: An Overview," described three categories of AI safety issues: problems of robustness, assurance, and specification. This paper introduces adversarial examples, a major challenge to robustness in modern machine learning systems.

Introduction

As machine learning becomes more widely used and applied to areas where safety and reliability are critical, the risk of system failures causing significant harm rises. To avoid such failures, machine learning systems will need to be much more reliable than they currently are, operating safely under a wide range of conditions.[1] In this paper, we introduce adversarial examples—a particularly challenging type of input to machine learning systems—and describe an artificial intelligence (AI) safety approach for preventing system failures caused by such inputs.

Machine learning systems are designed to learn patterns and associations from data. Typically, a machine learning method consists of a statistical model of the relationship between inputs and outputs, as well as a learning algorithm. The algorithm specifies how the model should change as it receives more information (in the form of data) about the input–output relationship it is meant to represent. This process of updating the model with more data is called "training."

Once a machine learning model has been trained, it can make predictions (such as whether an image depicts an object or a human), perform actions (such as autonomous navigation), or generate synthetic data (such as images, videos, speech, and text). An important trait in any machine learning system is its ability to work well, not only on the specific inputs it was shown in training, but also on other inputs. For example, many image classification

models are trained using a dataset of millions of images called ImageNet; these models are only useful if they also work well on real-life images outside of the training dataset.

Modern machine learning systems using deep neural networks—a prevalent type of statistical model—are much better in this regard than many other approaches. For example, a deep neural network trained to classify images of cats and dogs in black and white is likely to succeed at classifying similar images of cats and dogs in color. However, even the most sophisticated machine learning systems will fail when given inputs that are *meaningfully* different from the inputs they were trained on. A cat-and-dog classifier, for example, will not be able to classify a fish as such if it has never encountered an image of a fish during training. Furthermore, as the next section explores in detail, humans cannot always intuit which kinds of inputs will appear meaningfully different to the model.

Adversarial Examples

One of the most significant current challenges in AI safety is creating machine learning systems that are robust to adversarial examples. Adversarial examples are model inputs (for example, images) designed to trick machine learning systems into incorrect predictions. In the case of a machine learning system designed to distinguish between cats and dogs, an adversarial example could be an image of a cat modified to appear to the model as a dog. Since machine learning systems process data differently from humans, the cat image could be altered in ways imperceptible to humans but meaningfully different to the machine learning system. The modified image may still resemble a cat to humans, but to a machine learning system, it "looks like" a dog.

Adversarial examples can be generated systematically, either by digitally altering the input to a system or by directly altering the appearance of objects in the physical world. Unlike other adversarial attacks, such as "data poisoning," which seeks to attack the *algorithm* used to train a machine learning model, adversarial examples are designed to attack already *trained models*.

Figures 1 and 2 show systematically generated adversarial examples. Specifically, the adversarial example in Figure 1c digitally modifies the original image by an imperceptibly small amount, whereas the adversarial example in Figure 2b is created by adding patches to the image designed to mimic irregularities found in the physical world (such as graffiti or stickers). Both adversarial examples are generated via so-called white-box attacks, which assume the attacker knows how the trained classification model works and can exploit this knowledge to create adversarial examples that trick the model into making incorrect predictions.

Figure 1. An example of a "white-box" adversarial example from Goodfellow et al. (2015). The original image (a) is classified as "panda" with 57.7 percent probability. After being overlaid with a minimal amount of noise—the adversarial alteration (b) multiplied by a factor of 0.007—the resulting image (c) is classified as "gibbon" with 99.3 percent probability. The difference between (a) and (c) is imperceptible to the human eye.

a) Original image b) Adversarial alteration c) Adversarial example

Source: Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and Harnessing Adversarial Examples," International Conference on Learning Representations, Vol. 3, San Diego, CA, USA, May 7-9, 2015, https://arxiv.org/abs/1412.6572.

Figure 2. An example of a white-box adversarial example designed to generate physical alterations for physical-world objects. The adversarial alteration (b), which is designed to mimic the appearance of graffiti (a), tricks an image classifier into not seeing a stop sign.

a) Graffiti on stop sign b) Adversarial Example

Source: Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, Dawn Song, "Robust Physical-World Attacks on Deep Learning Visual Classification," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, June 18-23, 2018, pp. 1625-1634, https://arxiv.org/abs/1707.08945.

Although modern machine learning systems usually generalize remarkably well to data similar to the data used for training, adversarial examples can be created from surprisingly simple modifications to model inputs. Changes such as blurring or cropping images, or altering the appearance of the physical-world objects shown in an image, can fool an otherwise reliable system. In Figure 3b, an adversarial example is constructed by reducing the resolution of the original image, thereby changing the model's prediction from correct to incorrect. Unlike the adversarial examples in Figures 1 and 2, the adversarial example in Figure 3 was created via a black-box attack—that is, created *without* access to the trained classification model. It is not as subtle as the alteration in Figure 1 and not as targeted as the alteration in Figure 2. However, it demonstrates that modern machine learning systems can be fooled with little effort and no knowledge of the prediction model.

Figure 3. An example of a black-box adversarial example. The original image (a) is classified as “washer” with 53 percent probability. The image is altered by reducing its resolution to create an adversarial example (b), which is classified as “safe” with 37 percent and as “loudspeaker” with 24 percent probability.

a) Original image b) Adversarial example

Source: Alexey Kurakin, Ian Goodfellow, and Samy Bengio, “Adversarial Examples in the Physical World,” arXiv [cs.CV] (February 11, 2017), preprint, https://arxiv.org/abs/1607.02533.

Robustness to Adversarial Examples

Robust machine learning systems need to be able to identify data that is *meaningfully* different from training data and provide a defense against adversarial examples. There are a wide range of different research areas attempting to make progress in this direction. One such research direction aims to incorporate predictive uncertainty estimates into machine learning systems. This way, any prediction from the system would come with an estimate of certainty. If the machine learning system indicates uncertainty about the correctness of its prediction, a human operator can be alerted.

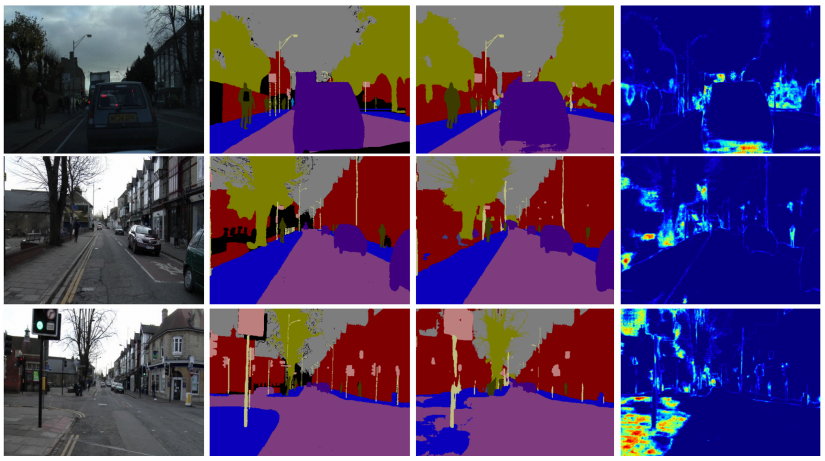
To understand predictive uncertainty estimates and how they can make machine learning systems more robust to adversarial examples, consider the classification “probability scores” given in the descriptions of Figures 1 and 3. In reality, these scores, which express the probability of an input belonging to a certain class (e.g., the class “cat” or “dog”), are misleading. While they do express a

probability, they do not actually express the model's level of *certainty* about the correctness of the predictions.

To fully understand this point, consider a machine learning system trained to distinguish between two classes: cats and dogs. Such a system will by design have two outputs: one for the class "cat" and one for the class "dog." If the model is given an image of a dog, it will output values between zero and one for each class—for instance, 90 percent and 10 percent for the classes "dog" and "cat," respectively, so that the values sum up to 100 percent. However, if given an image of a fish, the model will still make predictions for the two classes on which it was trained, unaware that it is being asked to identify an object it was not trained to recognize. In a best-case scenario, it would give outputs of 50 percent for each class, indicating that the input is equally likely to be either a cat or a dog. In a worst-case scenario, it would give a high probability score for one class, providing a false sense of certainty. But the way most machine learning systems are designed, they cannot give a low score to *both* the "cat" and "dog" labels. As such, these outputs should not be read as the machine learning system's "confidence" in the correctness of its classification.

Predictive uncertainty estimates can fill this gray spot. They complement the regular model outputs by expressing the model's uncertainty about the correctness of its predictions. If a machine learning system has good predictive uncertainty estimates, then the probability scores in Figure 3 would be accompanied by a high uncertainty score, indicating that the model is highly uncertain about the correctness of the predictions. Such uncertainty estimates can help a human operator avoid wrong predictions in safety-critical settings and ensure the system's reliability and safety, as demonstrated in Figure 4.

Figure 4. An example of predictive uncertainty estimates for autonomous vehicles. The first column shows the image fed into the system, the second column shows the ground truth classification of objects in the image (buildings, sky, street, sidewalk, etc.), the third column shows the model's classification, and the rightmost column shows the system's uncertainty about its classification. As can be seen from the image on the bottom right, the system is uncertain about its classification of parts of the sidewalk and could alert the human operator to take over the steering wheel.



Source: Alex Kendall and Yarin Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Vol. 30, Long Beach, CA, USA, December 4-9, 2017, pp. 5574-5584, https://arxiv.org/abs/1703.04977.

Unfortunately, obtaining reliable predictive uncertainty estimates for modern machine learning systems remains an unsolved problem. While several existing methods can generate uncertainty estimates, there are no mathematical guarantees that these uncertainty estimates are actually accurate. Furthermore, while empirical studies demonstrate that certain methods produce good predictive uncertainty estimates in some settings, those results cannot be generalized to any setting. Like other areas of robustness research, developing methods that yield reliably well-calibrated uncertainty estimates for modern machine learning systems is an active and ongoing area of research.

Outlook

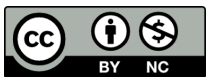
While modern machine learning systems often perform well on narrowly defined tasks, they can fail when presented with tasks meaningfully different from those seen during training. Adversarial attacks exploit this vulnerability by presenting inputs to machine learning systems specifically designed to elicit poor predictions. Adversarially robust machine learning systems seek to fix this vulnerability through mechanisms allowing the system to recognize when an input is meaningfully different from data seen during training, making the system more reliable in practice. Unfortunately, while an active area of research, existing approaches to detecting and defending against adversarial attacks do not yet provide satisfactory solutions, and the timeline to develop and deploy truly robust modern machine learning systems remains uncertain. For now, anyone considering deploying modern machine learning systems in safety-critical settings must therefore grapple with the fact that in doing so, they are introducing safety risks that we do not yet know how to mitigate effectively.[2]

Authors

Tim G. J. Rudner is a non-resident AI/ML fellow with CSET and a PhD candidate in computer science at the University of Oxford. Helen Toner is director of strategy at CSET.

Acknowledgements

For feedback and assistance, we would like to thank Igor Mikolic-Torreira, Shelton Fitch, Alexandra Vreeman, Lynne Weil, Larry Lewis, Jack Clark, Dewey Murdick, and Michael Page.



CSET Product ID #: 20190041

Document Identifier: doi: 10.0.201.137/20190041

Endnotes

[1] Organizational factors will also play an important role in whether the deployment of a given AI system is safe. For more on the importance of robust organizational practices, see Thomas G. Dietterich, "Robust Artificial Intelligence and Robust Human Organizations," arXiv [cs.AI] (November 27, 2018), preprint, https://arxiv.org/abs/1811.10840.

[2] Andrew Lohn, "Hacking AI" (Center for Security and Emerging Technology, December 2020), https://cset.georgetown.edu/research/hacking-ai/.