The Adversarial Robustness Toolbox—securing AI against adversarial threats

April 17 2018, by Maria-Irina Nicolae, Mathieu Sinn

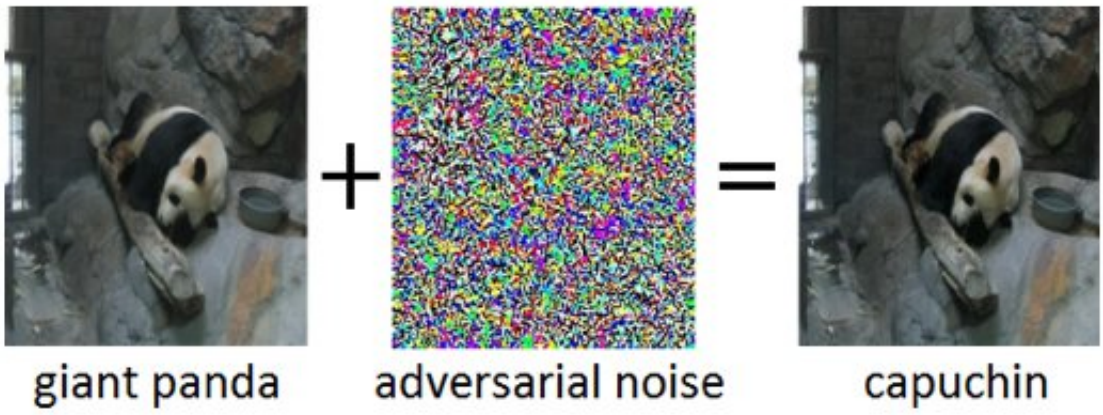

Figure 1: Adversarial example (right) obtained by adding adversarial noise (middle) to a clean input image (left). While the added noise in the adversarial example is imperceptible to a human, it leads the Deep Neural Network to misclassify the image as "capuchin" instead of "giant panda." Credit: IBM Blog Research

Recent years have seen tremendous advances in the development of artificial intelligence (AI). Modern AI systems achieve human-level performance on cognitive tasks such as recognizing objects in images, annotating videos, converting speech to text, or translating between different languages. Many of these breakthrough results are based on Deep Neural Networks (DNNs). DNNs are complex machine learning

models bearing certain similarity with the interconnected neurons in the human brain. DNNs are capable of dealing with high-dimensional inputs (e.g. millions of pixels in high-resolution images), representing patterns in those inputs at various levels of abstraction, and relating those representations to high-level semantic concepts.

An intriguing property of DNNs is that, while they are normally highly accurate, they are vulnerable to so-called adversarial examples. Adversarial examples are inputs (say, images) which have deliberately been modified to produce a desired response by a DNN. An example is shown in Figure 1: here the addition of a small amount of adversarial noise to the image of a giant panda leads the DNN to misclassify this image as a capuchin. Often, the target of adversarial examples is misclassification or a specific incorrect prediction which would benefit an attacker.

Adversarial attacks pose a real threat to the deployment of AI systems in security critical applications. Virtually undetectable alterations of images, video, speech, and other data have been crafted to confuse AI systems. Such alterations can be crafted even if the attacker doesn't have exact knowledge of the architecture of the DNN or access to its parameters. Even more worrisome, adversarial attacks can be launched in the physical world: instead of manipulating the pixels of a digital image, adversaries could evade face recognition systems by wearing specially designed glasses, or defeat visual recognition systems in autonomous vehicles by sticking patches to traffic signs.

IBM Research Ireland is releasing the Adversarial Robustness Toolbox, an open-source software library, to support both researchers and developers in defending DNNs against adversarial attacks and thereby making AI systems more secure. The release will be announced at the RSA conference by Dr. Sridhar Muppidi, IBM Fellow, VP and CTO IBM Security, and Koos Lodewijkx, Vice President and CTO of

Security Operations and Response (SOAR), IBM Security.

The Adversarial Robustness Toolbox is designed to support researchers and developers in creating novel defense techniques, as well as in deploying practical defenses of real-world AI systems. Researchers can use the Adversarial Robustness Toolbox to benchmark novel defenses against the state-of-the-art. For developers, the library provides interfaces which support the composition of comprehensive defense systems using individual methods as building blocks.

The library is written in Python, the most commonly used programming language for developing, testing and deploying DNNs. It comprises state-of-the-art algorithms for creating adversarial examples as well as methods for defending DNNs against those. The approach for defending DNNs is three-fold:

- Measuring model robustness. Firstly, the robustness of a given DNN can be assessed. A straight-forward way for doing this is to record the loss of accuracy on adversarially altered inputs. Other approaches measure how much the internal representations and the output of a DNN vary when small changes are applied to its inputs.
- Model hardening. Secondly, a given DNN can be "hardened" to make it more robust against adversarial inputs. Common approaches are to preprocess the inputs of a DNN, to augment the training data with adversarial examples, or to change the DNN architecture to prevent adversarial signals from propagating through the internal representation layers.
- Runtime detection. Finally, runtime detection methods can be applied to flag any inputs that an adversary might have tempered with. Those methods typically try to exploit abnormal activations in the internal representation layers of a DNN caused by the adversarial inputs.

To get started with the Adversarial Robustness Toolbox, check out the open-source release under github.com/IBM/adversarial-robustness-toolbox . The release includes extensive documentation and tutorials to help researchers and developers get quickly started. A white paper outlining details of the methods implemented in the library is in preparation.

This first release of the Adversarial Robustness Toolbox supports DNNs implemented in the TensorFlow and Keras deep learning frameworks. Future releases will extend the support to other popular frameworks such as PyTorch or MXNet. Currently, the library is primarily intended to improve the adversarial robustness of visual recognition systems, however, we are working on future releases that will comprise adaptations to other data modes such as speech, text or time series.

As an open-source project, the ambition of the Adversarial Robustness Toolbox is to create a vibrant ecosystem of contributors both from industry and academia. The main difference to similar ongoing efforts is the focus on defence methods, and on the composability of practical defence systems. We hope the Adversarial Robustness Toolbox project will stimulate research and development around adversarial robustness of DNNs, and advance the deployment of secure AI in real world applications. Please share with us your experience working with the Adversarial Robustness Toolbox and any suggestions for future enhancements.

More information: Valentina Zantedeschi et al. Efficient Defenses Against Adversarial Attacks, *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security - AISec '17* (2017). DOI: 10.1145/3128572.3140449

Provided by IBM