Neural Nets Project Proposal: Adversarial Examples

Daniel Donoghue, Nicholas Lines, and Arnaldo Pereira

Abstract. Adversarial approaches and associated model defences are an important area of machine learning research and practice. Neural networks are particularly vulnerable to so-called adversarial examples [**1**], which has led many to experiment with techniques for defeating these attacks and make networks more robust in general.

Our project will explore this issue using the well-known ImageNet dataset to both produce and attempt to defeat adversarial examples created in standard ways, using one or more neural archetectures. This will familiarize the researchers and other course participants with current issues in machine learning security in general and neural network robustness in particular, while providing an opportunity to review and apply architectures discussed or referenced in our course.

The authors are listed alphabetically, and all made equal contributions. This work is performed in association with the Johns Hopkins Engineering for Professionals Program, as a project for EN.625.638.8VL2.FA20 Neural Networks.

1. Introduction and background

As neural networks and other forms of machine learning have become ubiquitous in many areas of industry, academic research, and government use, questions related to machine learning security have come into sharper focus and gathered significant attention. This is evidenced by the sheer volume of papers submitted and accepted at data science and machine learning conferences such as the International Conference on Machine Learning and Neurips.

With this field of research has come a new vocabulary, so let us first establish a handful of useful terms in roughly the vernacular of modern research [**2**]. Adversarial machine learning refers to the misuse of data, models, and algorithms related to machine learning tasks, with the intent to defeat or exploit these elements in a manner unintended by their author[1]. These attacks generally fall into one of the following categories.

(1) **Evasion methods:** Without access to the original training data, an adversary attempts to make data that is misclassified or misunderstood

[1]It is important for those new to this field of study to distinguish it from the unfortunately similarly-named area of adversarial networks such as GANs, which employ two opposing computational entities (i.e. separate networks) to generate data and discriminate the quality of the data, in an effort to match a desired set of features.

by the network. Usually the network is viewed as a black box that the adversary wishes to fool, by guessing features it will care about. As an example, consider pirates who upload copyrighted material that has been inverted to YouTube to try to defeat anti-piracy reviews.

(2) **Data poisoning:** In this scenario the adversary has influence over the training data used to create or refine a model, and they exercise this to introduce data that will produce misclassifications or other incorrect model decisions. The classic example is the introduction of a mislabeled stop sign with a minor alteration in an autonomous vehicle's training data, which may lead the vehicle to interpret a similarly altered stop sign in real-life as a speed limit sign. The adversary may or may not have a transparent view of the model.

(3) **Extraction methods:** This category covers attempts by adversaries to extract from interactions with a model something they weren't intended to have, usually some or all of the training data, or the model itself. An example of this might be a competitor probing an image-recognition application to recreate the model cheaply and reuse it in another illegitimate setting.

Each of these approaches has a role in neural network research, but we will restrict our scope to discussing evasion and poisoning methods used to trick neural networks into making incorrect classification decisions.

This area of work is about 6 years old. In 2014 Szegedy et al discovered what they called "intriguing properties of neural networks," essentially that, despite making classification decisions that were extremely highly correlated with those of human experts, the machine-made and human-made decisions used entirely different justifications to reach the same conclusions. This implies that it is all too easy for an adversary to produce new synthetic data that the human and machine judges will unexpectedly classify entirely differently. The classic example of a panda that GoogLeNet mistakes for a gibbon thanks to added noise that is invisible to humans is shown in Fig. 1. This example was part of Goodfellow et al's first of many papers [**3**] on the subject of creating, detecting, and defeating such adversarial examples.

2. Project description

We propose an exploration of the generation and defeat of adversarial examples applied to an image classification task using the standard ImageNet dataset [**4**]. Our work will be performed in Python using standard libraries where possible, and publicly available under a standard MIT license on GitHub[2]. The goal of this project is to create, apply, and attempt to defeat one or more adversarial examples in the context of one or more neural architectures. In the interest of allowing a naturally expandable scope, we leave open for the moment the question of how many architectures and defeating techniques we will use, stating only that we will use at least one of each, sampled from the following lists.

[2]https://github.com/linesn/adversarial_examples

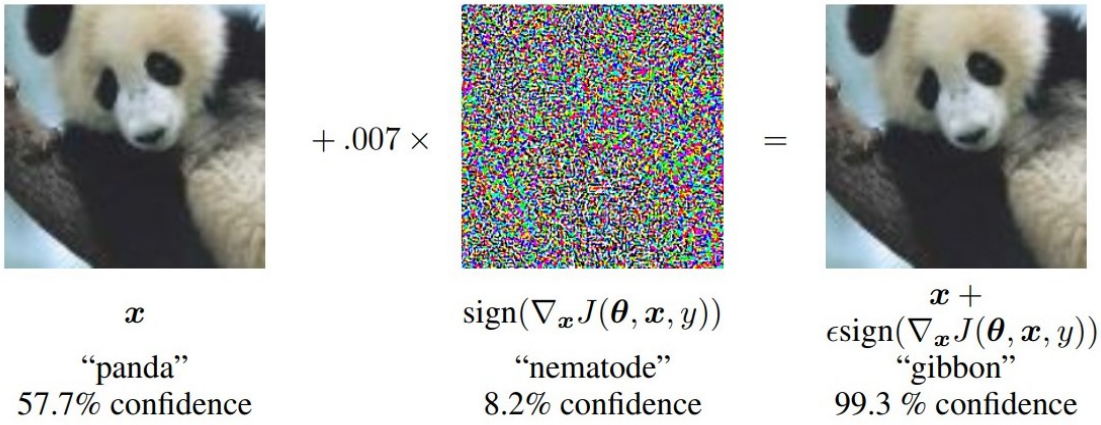

FIGURE 1. A miscategorized panda, reproduced from [**3**].

Network architectures of interest	**Defeat methods**
(Multilayered) Perceptrons	Dropout (regularization)
Convolutional Neural Networks	Pruning
Variations with different learning methods	Training using perturbed training data
	Batch normalization

We will deliver, as a group, at least one successful adversarial example (that is, an attack that worked) against at least one network, attempt to defeat the attack, iterate onward through other defeat methods and architectures as time allows.

This project is not structured as a novel scientific experiment, but rather as a confirmation of scientific theory. While our particular results will never have been shown before, they should resemble expected behavior that has been discussed by neural defence researchers for the past five years. If on the other hand we are unable to produce or defeat these previously discovered attacks, we will analyze why our experiment differed from the accepted model.

3. Timeline

As we have not yet discussed in our course a timeline for the project, we propose the following.

Date	**Deliverable**
11/11/2020	First draft of proposal
12/01/2020	Preliminary results documented in a presentation to give in class
12/12/2020	Project paper turned in

4. Concluding thoughts

We hope this project will not only advance the authors's understanding of adversarial machine learning's intersection with neural networks, but also serve as an introduction to the subject for our classmates. As the world becomes more and more dependent upon real-time processing of big data using artificial intelligence

and other machine-made decisions, gaining an appreciation and intuition for security and reliability of models and algorithms will be of primary importance to machine learning practitioners and researchers alike.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014.

[2] Wikipedia contributors, "Adversarial machine learning — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Adversarial_machine_learning&oldid=987074175, 2020, [Online; accessed 11-November-2020].

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

E-mail address: ddonogh1@jhu.edu

E-mail address: nicholasalines@gmail.com

E-mail address: aepereira@gmail.com