DAT340 - Applied Machine Learning
Writing Assignment 2

Breaking things is easy, by Papernot and Goodfellow

Stefano Ribes
Chalmers University of Technology
ribes@chalmers.se

Abstract

This report contains a short discussion on the topic of security and privacy of machine learning. The work summarizes a series of blog posts by Nicolas Pepernot and Ian Goodfellow, in which they provide an overview of adversarial attacks and candidate solutions to the problem. They also describe the current limitations of defensive solutions and verification systems. My personal opinion about their analysis is that they offer an interesting and rich theoretical framework for inspiring more research in the area. They however lack a discussion on some practical aspects which might be relevant when deploying a given machine learning system.

1 Introduction

Designing and implementing secure and reliable machine learning algorithms has become a necessary and challenging requirement to satisfy. In their series of blog posts (bre, a) (bre, b) (bre, c), Nicolas Pepernot and Ian Goodfellow reason and discuss about this topic, trying to formulate a definition of attack and defense strategies in the context of machine learning. In particular, the analyzed research topic is defined as *security and privacy of machine learning*. This report includes a brief summary of their arguments, which include a description of the types of attacks, two candidate defense solutions and a failing one, and the difference between testing and verification. This document also includes extra real-life examples and a short discussion on such themes. Finally, a few personal opinions are drawn and presented.

2 Problem Statement

According to Pepernot and Goodfellow, machine learning algorithms have reached significant performance to justify their adoption in real-life applications, but they still appear vulnerable and easily breakable. In order to "break" a machine learning system, an attacker shall compromise at least one of the following characteristics of the system:

Confidentiality the machine learning system shall not leak sensitive information, *e.g.* training data like medical records in case of a diagnostic system.

Integrity the machine learning system shall not have its predictions altered from the intended ones, *e.g.* attackers can craft specific inputs to fool the machine learning output.

Availability the machine learning system shall be operational and functional, *e.g.* attackers inputs can compromise a system and make it unavailable to the user.

2.1 Type of Attacks

Having set the attackers' targets, the authors also describe three categories of attacks: attacks that can be computed at training time, others applied during inference and ones related to privacy.

Poisoning training sets is one way of making the machine learning system increase its prediction error when deployed. It can be achieved by adding extra malicious training data or by alter existing one.

Using adversarial examples is a way of altering the system's output at inference time and exploits inputs with usually small and undetectable perturbations.

Attacks on privacy are often attacks which aim at recovering part of a training data set or at infer sensitive information from the system's predictions. An attacker can in fact carefully observe and monitor the output of the machine learning model

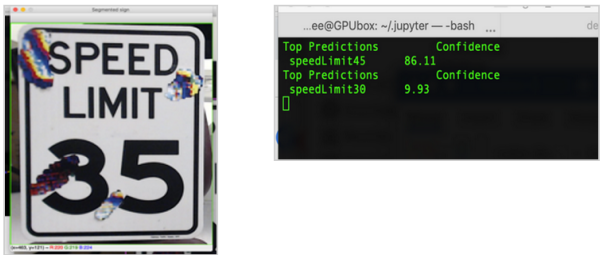

Figure 1: The modified speed limit sign and its detected speed limit.

and retrieve information about how the model was trained.

2.2 Real-life Example of Attacks

In (exa), the authors fool an autonomous driving system image recognition system. As seen in Figure 1, a few stripes of carefully designed tape were able to fool the system into detecting a much higher speed limit. It is clear that such adversarial attack can pose a serious threat to human lives and needs to be taken into consideration by the designers of such systems.

3 Proposed Solutions

The blog posts from Pepernot and Goodfellow continue by describing techniques that aim at making the models more robust against adversarial examples.

Adversarial training consists of generating adversarial training examples such that the model learns to assign the correct labels even on adversarial data. The fast gradient side method enables an efficient generation of such data points.

Defensive distillation starts from a trained model and trains a new model to predict its original output probabilities. The idea is to train the second model on "soft" labels such as probabilities, rather than 100% accurate labels. In this way, the model's decision surface in the adversarial directions is "smoothed" out and cannot be used for attacks.

Failed defense solution: Gradient Masking attempts to hide gradient information which attackers can exploit to generate adversarial examples. For instance, by outputting the predicted class label only, *i.e.* by omitting the output probabilities. This strategy is not however giving the defender any additional defense. Rather, it can slow down the attacker, but doesn't prevent it from training a custom model based on the system's predictions and then identify holes in the original model.

3.1 Limitations

After having listed the aforementioned solutions, the authors give an overview of the reasons why adversarial examples can be so effective against machine learning models. One reason is that the research field lacks of an effective theoretical framework to describe adversarial examples as solutions to non-linear and non-convex optimization problems. Because of that, all possible adversarial example cannot be identified and tackled in advance.

Another reason lies in the vast input space that is used by the majority of machine learning systems. Roughly speaking, it is nearly impossible to determine that the system will produce meaningful output for every possible input.

On top of that, it seems that training models to counter adversarial attacks might affect their accuracy. Not only that, improving the system defenses on "one side", *e.g.* a certain set of adversarial examples, might leave other sides uncovered.

3.2 Testing versus Verification

In their final post, the authors also analyze the different role of performing testing and verification on machine learning systems. Testing can be defined as a limited collection of checks that aim at finding possible errors produced by the model. On the other hand, verification can assure that the system is working as intended for a broad range of circumstances.

The authors believe that, although still in its infancy, verification of machine learning systems should be favoured over testing, as it provides additional guarantees. For example, testing might determine and measure the error rate of a given system for a selected pool of test data, whereas verification can guarantee that the error rate will be below a certain threshold for any possible input. In this scenario, a system which has only been tested and not verified can be affected by attacks involving unseen inputs.

Verification can be performed via verification systems: the idea is to verify that the classifier assigns to the same label all input points which are close to a given input data. However, there are currently two major limitations. One limitation is re-

lated to which set of points the verification system should actually use. The second one refers instead to the neighbourhood of the selected input points: currently small L^p norm balls are utilized, but researcher speculate that such regions might be less straightforward in shape and size.

4 Personal Opinions

Having highlighted the main aspects of the problem and some possible solutions to it, in this final part of this report I include a short personal opinion and discussion on the aforementioned topics.

4.1 Opinions on the Topic

In general, I think that this is an important topic which needs to be addressed when deploying machine learning systems into the real world. We cannot just hope for our system to be resilient and robust, unless we specifically design it with an eye on security. After all, reliability is becoming a more and more important requirement, the same or even more than performance. There exist significant motivations on the practical side that support such argument. In fact, any compromised machine learning system can cause physical and social harm (like in the case of autonomous cars or leaked privacy information), not to mention, it can also pose a legal cost on the company and the designers of such vulnerable systems.

4.2 Opinions on the Problem Statement

Pepernot and Goodfellow categorization of attacks rely on the assumption that a machine learning system is a black box that can be interacted with via its input and output. In my opinion this view, although rich of inspiring points, misses an additional dimension of vulnerability: the machine learning system itself, defined as its parameters. Despite supplying adversarial examples in input is a legitimate way of attacking a model, we should also remember that such systems are eventually deployed in real computational systems. As such, they might be vulnerable to attacks specific to that realm too.

This means that given a classifier parameters, *e.g.* neural network weights, are eventually stored on computer memory and can be altered by malicious attackers. Imagine having some of the parameters numerical values being modified, for instance by random bit flips, how is the output affected by such alterations? Is there a way we can train or build machine learning systems to prevent such errors in the output?

The problem of securing the model's parameters might also be *orthogonal* to the problem of guaranteeing confidentiality, integrity and availability presented by the authors. In fact, nothing stops an attacker from both utilizing adversarial examples and injecting faults into the system parameters. However, it is true that modifying the parameters might affect the effectiveness of the adversarial examples themselves, making their output even more unpredictable and eventually identical to the original expected label, thus nullifying the attack.

4.3 Opinions on the Proposed Solutions

The proposed solutions are in their infancy and unfortunately seem to lack behind the advances of the attacking side. I really appreciated the emphases that the authors put on the matter, because it rises the awareness that this is still an open question and in general an open field of research.

From my understanding, the proposed solutions all tend to find a one-size-fits-all solution through a general purpose theoretical framework. In my opinion, there might be chances that tailored solutions to more practical problems and applications might be more effective at protecting machine learning systems from attacks.

5 Conclusions

In this work, I described some of the theory and applications of the topic of security and privacy of machine learning. To do so, I reported the work of Nicolas Pepernot and Ian Goodfellow presented in a series of online blog posts. In it, they analyze what type of attacks and defense mechanism are currently studied in this research field. They also argue that verification of machine learning systems shall become more popular, as it is more powerful than performing testing. In my opinion, their analysis is really detailed and well motivated, opening interesting research questions and paving the way towards more robust machine learning systems.

References

a. Breaking things is easy. http://www.cleverhans.io/security/privacy/ml/2016/12/16/

breaking-things-is-easy.html. Accessed: 2022-05-23.

b. Is attacking machine learning easier than defending it? http://www.cleverhans.io/security/privacy/ml/2017/02/15/why-attacking-machine-learning-is-easier-than-defending-it.html. Accessed: 2022-05-23.

Model Hacking ADAS to Pave Safer Roads for Autonomous Vehicles. https://www.mcafee.com/blogs/blogs/other-blogs/mcafee-labs/model-hacking-adas-to-pave-safer-roads-for-autonomous-vehicles. Accessed: 2022-05-23.

c. The challenge of verification and testing of machine learning. http://www.cleverhans.io/security/privacy/ml/2017/06/14/verification.html. Accessed: 2022-05-23.