Breaking things is easy

Dec 15, 2016

by Nicolas Papernot and Ian Goodfellow

Until a few years ago, machine learning algorithms simply did not work very well on many meaningful tasks like recognizing objects or translation. Thus, when a machine learning algorithm failed to do the right thing, this was the rule, rather than the exception. Today, machine learning algorithms have advanced to the next stage of development: when presented with *naturally occurring* inputs, they can outperform humans. Machine learning has not yet reached true human-level performance, because when confronted by even a trivial adversary, most machine learning algorithms fail dramatically. In other words, we have reached the point where machine learning works, but may easily be broken.

This blog post serves to introduce our new Clever Hans blog, in which we will discuss all of the many ways an attacker can break a machine learning algorithm. Academically speaking, our topic is the **security and privacy of machine learning** [PMS16]. The blog is jointly written by Ian Goodfellow and Nicolas Papernot. Ian is a research scientist at OpenAI, and Nicolas a Google PhD Fellow in Security at Penn State. We jointly created cleverhans, an open-source library for benchmarking the vulnerability of machine learning models to adversarial examples. The blog gives us a way to informally share ideas about machine learning security and privacy that are not yet concrete enough for traditional academic publishing, and to share news and updates relevant to the cleverhans library.

A **secure** system is one that can be depended upon and is guaranteed to behave as expected [GSS03]. When we attempt to provide guarantees about how the system will behave, we do so with a particular **threat model** in mind. The threat model is a formally defined set of assumptions about the **capabilities** and **goals** of any attacker who may wish the system to misbehave.

So far, most machine learning has been developed with a very weak threat model, in which there is no opponent. The machine learning system is designed to behave correctly when confronted by nature. Today, we are beginning to design machine learning systems that behave correctly even when confronted with a malicious person or a malicious machine learning-based adversary.

For instance, a machine learning system may be targeted by an adversary while the model is being trained (the learning phase) or when the model is making predictions (the inference phase). Adversaries also have varying degrees of capabilities, which may include access to the model internals like its architecture and parameters, or to the model inputs and outputs.

To break a machine learning model, an attacker can compromise its **confidentiality**, **integrity**, or **availability**. Together, these properties form the CIA model of security.

- To provide **confidentiality**, a machine learning system must not leak information to unauthorized users. In practice, confidentiality in machine learning makes more sense when thought about as privacy: the model must not leak sensitive data. For example, suppose that researchers built a machine learning model that could examine a patients' medical records and offer disease diagnoses. Publishing such a model could provide a valuable resource to doctors, but it is important to make sure that a malicious person cannot examine the model and recover private medical data regarding the patients who helped to train the model.
- Adversaries capable of tampering with the model's **integrity** can alter its predictions to differ from the intended ones. For instance, spammers try to design their e-mail messages to be incorrectly recognized as legitimate messages.
- An attacker can also compromise a system's **availability**. For example, a self-driving car might be forced to go into a failsafe mode and pull over if an adversary placed an extremely confusing object next to the road ahead of a car.

Of course, all of this so far is hypothetical. What kinds of attacks have security researchers actually demonstrated so far? Later posts in this blog will give many more examples, but let's start with just three: integrity attacks at training time, integrity attack during inference, and privacy attacks.

Poisoning training sets

The adversary may interfere with the integrity of the training process by making modifications to existing training data or inserting additional data in the existing training set. For example, suppose Professor Moriarty wishes to frame Sherlock Holmes for a crime. He may arrange for an unsuspected accomplice to give Sherlock Holmes a pair of very unique and ornate boots. After Sherlock has worn these boots in the presence of the policemen he routinely assists, the policemen will learn to associate the unique boots with him. Professor Moriarty may then commit a crime while wearing a second copy of the same pair of boots, leaving behind tracks that will cause Holmes to fall under suspicion.

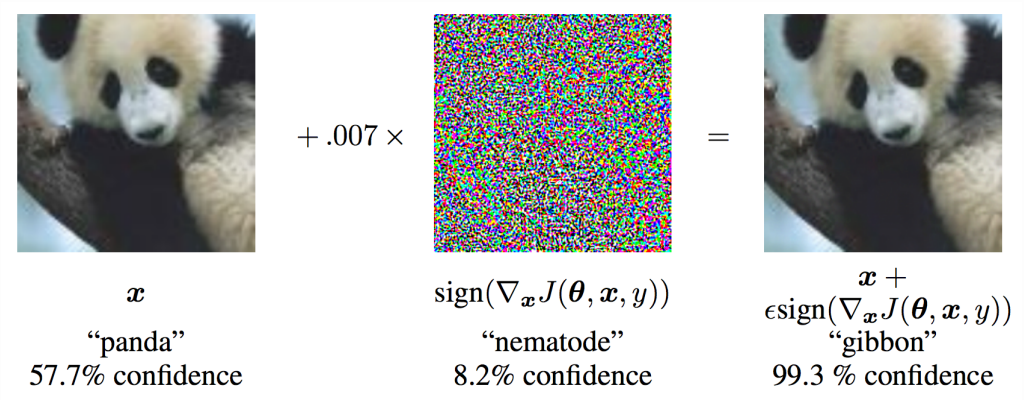
In machine learning, the strategy followed by the adversary is to perturb training points in a way that increases the prediction error of the machine learning when it is used in production. For instance, this method was used to poison the training set of support vector machines: the

convexity of the loss function used to measure their prediction error allows the adversary to find exactly the set of points that when perturbed will most harm the model's performance at inference [BNL12]. The problem of finding effective poisoning points remains open for more complex models, which don't yield convex optimization problems, like deep neural networks.

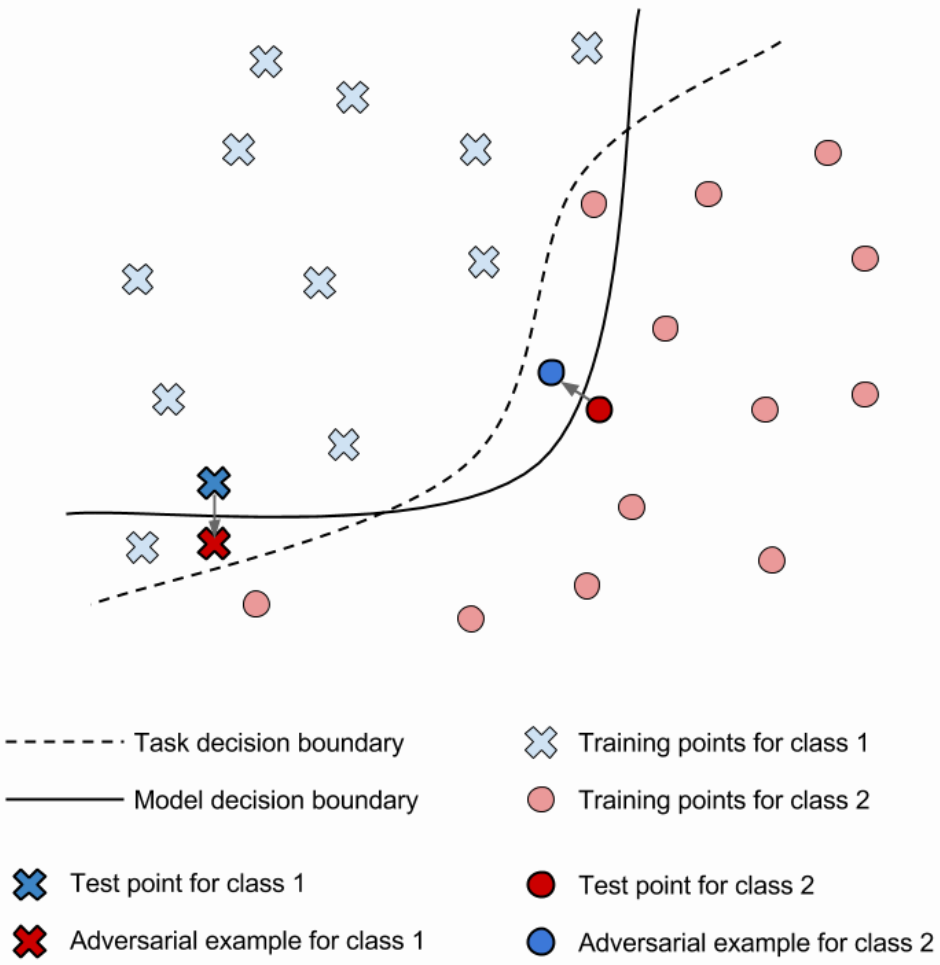
Forcing models to make mistakes with adversarial examples

In fact, breaking things is so easy that adversaries need not poison the training data analyzed to learn the parameters of a machine learning model. Instead, adversaries can force models to make mistakes *instantly* by only perturbing the inputs on which the model makes predictions (after completion of training—during the inference phase).

A common way to find perturbations forcing models to make wrong predictions is to compute **adversarial examples** [SZS13]. They yield perturbations that are very slight and often indistinguishable to humans, yet force machine learning models to produce wrong predictions. For instance, in the illustration below reproduced from [GSS14], the image on the left is correctly classified by a machine learning model as a panda, but adding the noise represented in the middle to that same image results in the image on the right, which is classified as a gibbon by the model.



Note that despite being indistinguishable to the human eye, the perturbation is sufficient to change the model's prediction. Indeed, the perturbation is computed to minimize a specific norm in the input domain while increasing the model's prediction error. This effectively pushes the input, which was originally correctly classified, across the model's decision boundary into a wrong class. This phenomenon is illustrated in the 2D Figure below, for a problem with two output classes.



Many attacks based on adversarial examples require that the adversary have knowledge of the machine learning model parameters to solve the optimization problem for the input perturbation. In constrast, some follow-up work considers the more realistic threat model of an adversary only capable of interacting with the model by observing its predictions on chosen inputs. For instance, this could be used by an attacker to figure out how to design webpages that are well ranked by a machine learning model designed a la PageRank, or how to craft spam that evades detection. In these black-box settings, the machine learning model is said to act as an **oracle**. A strategy is to first query the oracle in order to extract an approximation of its decision boundaries—the substitute model—and then use that extracted model to craft adversarial examples that are misclassified by the oracle [PMG16]. This is one of the attacks that exploit the **transferability** of adversarial examples: they are often misclassified simultaneously across different models solving the same machine learning task, despite the fact that these models differ in their architecture or training data [SZS13].

Privacy issues in machine learning

Privacy issues in machine learning need not necessarily involve adversaries. For instance, an emerging field is addressing concerns related to the lack of fairness and transparency when learning algorithms process the training data. In fact, it has been pointed out recently that social

biases encoded in the training dataset translate to biased model predictions upon completion of learning [BS16]. In the following, we instead focus on scenarios involving adversaries.

The goal of adversaries is often to recover part of the training data used to learn the model, or infer some sensitive properties of users by observing the model's prediction. For instance, smartphone keyboards now learn better predictive completions from the patterns entered by their users. However, typing a certain sequence of letters specific to a user should not lead to its appearance on other phones—unless a sufficiently large pool of users typed that sequence. As such, privacy attacks are primarily meaningful at the inference stage, but mitigating them typically requires some randomization in the learning algorithm [CMS11].

For instance, adversaries may seek to perform **membership inference** queries: know whether a particular training point was used to train the model. A recent paper looks closely at this question in the context of deep neural networks [SSS16]. In a somewhat opposite manner to the gradients used to craft adversarial examples (that change the model's confidence in the right answer), the membership attack searches for points classified with very high confidence in directions identified using gradients. It is also possible to infer more general statistical information about the training data used to learn a deployed model [AMS15].

Conclusion

It is now December 2016. At present, we know many different ways of attacking machine learning models, and very few ways of defending. We hope that by December 2017, we will have more effective defenses. The goal of this blog is to push forward the state of the art in machine learning security and privacy, by documenting advances as they happen, by provoking discussion within the community of researchers involved in these topics, and by inspiring a new generation of researchers to join the community.

Acknowledgements

Thanks to Catherine Olsson, for pointing out that we had "the rule, rather than the exception" backwards in the original post. We have since fixed the mistake.

Thanks to Ryan Sheatsley and Vincent Tjeng for pointing out a typo.

References

[AMS15] Ateniese, G., Mancini, L. V., Spognardi, A., Villani, A., Vitali, D., & Felici, G. (2015). Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. International Journal of Security and Networks, 10(3), 137-150.

[BS16] Barocas, S., & Selbst, A. D. (2016). Big data’s disparate impact. California Law Review, 104.

[BNL12] Biggio, B., Nelson, B., & Laskov, P. (2012). Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389.

[CMS11] Chaudhuri, K., Monteleoni, C., & Sarwate, A. D. (2011). Differentially private empirical risk minimization. Journal of Machine Learning Research, 12(Mar), 1069-1109.

[GSS03] Garfinkel, S., Spafford, G., & Schwartz, A. (2003). Practical UNIX and Internet security. O’Reilly Media, Inc.

[GSS14] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[PMG16] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Berkay Celik, Z., & Swami, A. (2016). Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples. arXiv preprint arXiv:1602.02697.

[PMS16] Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). Towards the Science of Security and Privacy in Machine Learning. arXiv preprint arXiv:1611.03814.

[SSS16] Shokri, R., Stronati, M., & Shmatikov, V. (2016). Membership Inference Attacks against Machine Learning Models. arXiv preprint arXiv:1610.05820.

[SZS13] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.