SAND2017-12451J

Adversarial Issues in Machine Learning

Dr. W. Philip Kegelmeyer, Sandia National Laboratories

Introduction

The US Government makes critical use of machine learning analytics in defense of national security. One of the primary defining characteristics of a "national security" analysis is the existence of adversaries who seek to sap, even suborn, that analysis. Through understanding the machine learning methods in play, they seek to produce data which is evolving, incomplete, deceptive, and otherwise custom-designed to defeat them.

This cannot be easily prevented. Recent work[1, 2] has shown that if a machine learning model is publicly deployed, it can itself be easily modeled, even duplicated, and then studied in private to discover its weaknesses. Even a privately held model might be sufficiently well deduced through reverse engineering, or network compromise. And once a model is understood, there are typically many avenues of attack, as the training data, test data, or both are generally uncontrolled, and can be modified by an adversary.

"Adversarial Machine Learning" addresses these issues, spanning developing attacks against machine learning, assessing defenses, detecting whether an attack is in progress, quantitative assessment of worst case scenarios, considerations around if, when, and how to deploy a machine learning model, and so on. Adversarial machine learning tradecraft is essentially applying vulnerability assessment methods at the *algorithm* level, rather than to software or hardware. The end goal is to harden the machine learning methods in use, and in any case, to regard their outputs with an informed, wary eye. That is, to become the top middle sheep in Figure 1, the one that doesn't quite buy into the "IF (white AND fuzzy) THEN <Harmless>" analytic.

‘¡

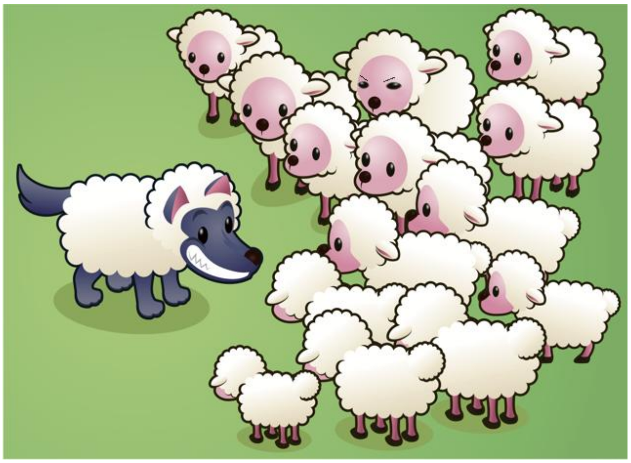


Figure 1: IF (white AND fuzzy) THEN <Harmless>

A Taxonomy of Adversary Goals

Though adversarial aspects of machine learning have been discussed for more than a decade[3], there is no broadly adopted consensus as to how to categorize an adversary's goals. One possibility is to think in terms of “quality”, “confidence”, or “evasion” attacks:

- In a quality attack the adversary’s goal is to drive down the overall effectiveness of machine learning as assessed on the *training* data, regardless of whether test performance is unaffected. The idea might be to convince the defender not to deploy an actually useful analytic, or to cause the defender to waste time attempting to improve it.

- In a confidence attack the adversary’s goal is to drive down the overall effectiveness of machine learning as assessed on the *test* data, without necessarily affecting accuracy on the training data. The idea here is to convince the defender to confidently deploy an ineffective analytic.

- In an evasion attack the adversary’s goal is to engineer a specific desired outcome for a specific future test sample or samples. Thus the idea is to appropriately shape a specific part of the machine learning decision surface, or to understand the existing decision surfaces well enough to be able to move evasively within them.

What Makes Machine Learning Vulnerable?

What might make a machine learning algorithm vulnerable to such attacks? Classic supervised machine learning methods depend on two fundamental assumptions; violating either of them creates exploitable weaknesses.

The first assumption is that the test data is essentially similar to the training data. It has long been well understood that this is often an unreliable expectation. For instance, data often changes slowly and naturally over time. Therefore much research has been focused on building machine learning models robust to dissimilar test data; examples are methods for handling concept drift[4], or for using transfer learning[5] to explicitly extend a machine learning model beyond its original training data.

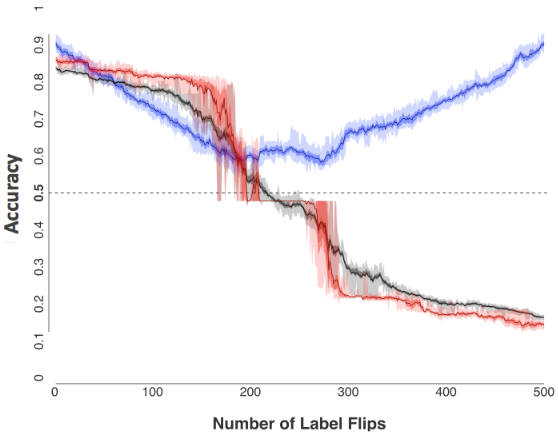
This test set similarity assumption is also the basis for most of the currently popular attacks against deep learning on image data[6]. Deep learning methods typically overfit their training data, generating machine learning models which are indeed very accurate if the test data is similar to the training data, but which are easily led astray by minutely altered test data. This vulnerability has created its own sub-field, Generative Adversarial Networks[7], in which one machine learning model is explicitly trained to generate images designed to fool a competing machine learning model, which is in turn trying to learn how not to be fooled.

The second assumption, perhaps less well appreciated, is that the “groundtruth” labels in the training data used to build the model are accurate. Undermining this assumption by tampering with the labels exposes particularly pernicious algorithmic vulnerabilities.

An Example Label Tampering Vulnerability

As one example, consider ensembles of bagged decision trees[8] as the machine learning method. For machine learning in general it is standard to assume that self-assessment on the training data via cross validation is a useful, if mildly optimistic, estimate of accuracy on an eventual test set. Further, for ensemble methods in particular, it is standard to assume that ensemble accuracy will be higher than the average accuracy of the individual trees. These assumptions have been correct so consistently that they are rarely examined.

With that in mind, consider Figure 2, which depicts the results of tampering with groundtruth labels. The underlying data is a product inspection data set with roughly balanced "Pass/Fail" labels. The curves indicate what happens to three measures of accuracy (on the y-axis) as we flip a certain number of the ground truth labels (the x-axis) before we build the ensemble model. Here the adversary is choosing which labels to flip in a purely random fashion.



(**ensembles training data**, **ensembles test data**, **single tree test data**)

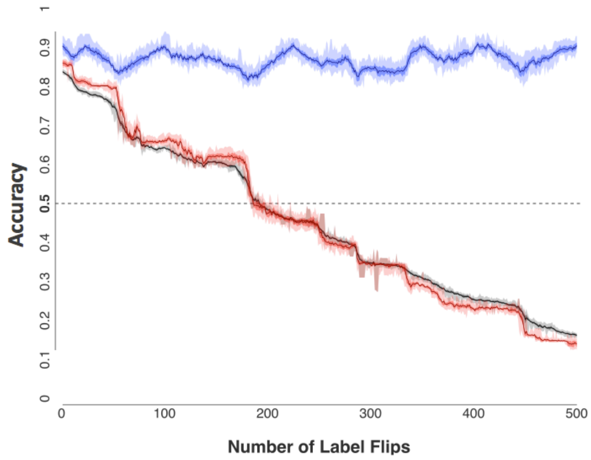
Figure 2: A Random Label Flipping Attack is Effective But Obvious

In that plot, the red curve is the test set accuracy, the accuracy we care about. Happily, nearly two hundred of the five hundred training samples must be corrupted before there is a noted drop in accuracy, which is a reassuring testament to the robustness of ensemble decision tree methods. Also, at first, the ensemble accuracy (in red) out performs the average single tree accuracy (in black), as we would hope.

The blue curve depicts the cross-validated training set accuracy. That does decrease nearly linearly with the amount of tampering (until a full half of the data is flipped), and thereby illustrates a mild example of a "quality" attack. That is, a defender who looked only at training set accuracy might incorrectly conclude that the test set accuracy would not be high enough to be useful.

Unfortunately, Figure 2 is a best-case scenario of a particularly lazy adversarial attack. Now consider Figure 3, which illustrates an effective "confidence" attack. Here the adversary has been slightly smarter, and has clustered all of the training data, randomly ordered the clusters, and then randomly attacked all members of a cluster before going on to the next one. This small change dramatically improves matters for the adversary. Now the test set ensemble accuracy (in red) decreases nearly linearly with the amount of tampering. It is also essentially no better than the average tree accuracy (in black), which means the extra computation required by ensembles is accomplishing nothing.

Most worrisome, however, is the fact that the training set accuracy, in blue, stays relatively flat regardless of the degree of tampering. This means, for instance, that if the



(**ensembles training data**, **ensembles test data**, **single tree test data**)

Figure 3: A Slightly Smarter Label Flipping Attack is Effective But **Not** Obvious

adversary can tamper one hundred of the training points, the actual real-world accuracy will decrease to about 60%, but the defender won't know this! They'll expect the accuracy to be around 90%, because that's what the training set validation indicates.

Conclusion

Adversarial machine learning is a new and rapidly developing field, and so this article was able only to introduce some of its ideas, along with a single example of an unnervingly effective attack.

Still, we can't stop using these methods, so perhaps we can learn to consider them with a useful sense of watchful paranoia. G.K. Chesteron famously said "We must learn to love life without ever quite trusting it" [9]; that seems the right perspective to take with machine learning as well.

Acknowledgments

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

[1] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. *CoRR*, abs/1609.02943, 2016.

[2] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pages 506–519, New York, NY, USA, 2017. ACM.

[3] Amir Globerson and Sam Roweis. Nightmare at test time: robust learning by feature deletion. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 353–360, New York, NY, USA, 2006. ACM Press.

[4] Geoffrey I. Webb, Roy Hyde, Hong Cao, Hai Long Nguyen, and Francois Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994, Jul 2016.

[5] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct 2010.

[6] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on machine learning models. *CoRR*, abs/1707.08945, 2017.

[7] Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017.

[8] Robert E. Banfield, Lawrence O. Hall, Kevin W. Bowyer, and W. Philip Kegelmeyer. A comparison of decision tree ensemble creation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):173–180, January 2007.

[9] G.K. Chesterton. *The Man Who Was Thursday: A Nightmare.* Jovian Press, 1908.