How to Make Machine Learning Robust Against Adversarial Inputs

Gerardo Muela[1], Christian Servin[2], and Vladik Kreinovich[1]
[1]Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, TX 79989, USA
gdmuela@miners.utep.edu, vladik@utep.edu
[2]Computer Science and Information Technology
Systems Department
El Paso Community College
919 Hunter
El Paso, TX 79915, USA
cservin@gmail.com

Abstract

It has been recently shown that it is possible to "cheat" many machine learning algorithms – i.e., to perform minor modifications of the inputs that would lead to a wrong classification. This feature can be used by adversaries to avoid spam detection, to create a wrong identification allowing access to classified information, etc. In this paper, we propose a solution to this problem: namely, instead of applying the original machine learning algorithm to the original inputs, we should first perform a random modification of these inputs. Since machine learning algorithms perform well on random data, such a random modification ensures us that the algorithm will, with a high probability, work correctly on the modified inputs. An additional advantage of this idea is that it also provides an additional privacy protection.

1 Adversarial Inputs to Machine Learning Algorithms: Formulation of the Problem

Machine learning algorithms have been very successful. Machine learning algorithms allow us, based on the known examples of different phenomena, to develop a general algorithm for detecting this phenomenon; see, e.g., [1]. For example, when presented with data from different patients with different diagnoses, machine learning can help diagnose new patients. When presented with

examples of spam and non-spam emails, machine learning algorithms can then determine, with high reliability, whether a new email is a spam or not.

In many practical applications, machine learning algorithms have been very successful: after the training stage, in the vast majority of case, they correctly classify the inputs.

Possibility of adversarial inputs. On random inputs, the machine learning algorithms work really well. They are not perfect: there is usually a small percentage of cases when these algorithm err, but in most cases, they classify the inputs correctly.

The problem is that in some practical situations, the inputs are not random: e.g., the spammers can use the machine learning algorithm's imperfection and, on purpose, modify the inputs so that the algorithm will erroneously classify them as non-spams. It has been shown that there is indeed a possibility of such adversarial small modifications of the original input; see, e.g., [3] and references therein.

This possibility seems to defeat the purpose of the machine learning algorithm: e.g., in the spam example, we wanted to separate spam from non-spam, and with a clever adversary, we are unable to do it.

How do we deal with such adversarial inputs? What can we do? In the ideal world, we should come up with better machine learning algorithms, algorithms which are not so easy to cheat. However, this is not easy: adversarial inputs use the fact that the machine learning algorithms are not 100% perfect, and the progress of machine learning, while showing a steady decrease in errors, seems to indicate that it is impossible to completely get rid of such errors.

So what can we do?

2 How to Defeat Adversarial Inputs: An Idea

Main idea. For the exact adversarial input, the machine learning algorithm provides a wrong result.

However, as we have mentioned, for a random inputs, the machine learning algorithm usually works well. Thus, if we add a random modification to the adversarial input, with high probability, the machine learning algorithm will provide a correct classification of this input – and we can further decrease the probability of error if we apply several different random modifications and apply the machine learning algorithm to all these modifications. This leads us to the following solution to the above problem.

Resulting solution to the above problem. Instead of applying the machine learning algorithm to the original inputs, we should:

- first, apply a random modification to the input (e.g., add random values), and then
- apply the original machine learning algorithm to the modified input.

Main pros and cons of this solution. The main disadvantage of the proposed solution is that since we are adding noise to the original input, we thus slightly decrease the efficiency of the machine learning algorithm.

The main advantage is that the adversaries can no longer disrupt the algorithm. From the viewpoint of such problems as detecting spam or checking whether a person is authorize to access a certain data, this advantage clearly outdoes the above minor disadvantage.

An additional advantage of the proposed solution. Adding random modifications to the inputs is one of the known ways of preserving data privacy; see, e.g., [2, 4]. Thus, the proposed solution has an additional advantage – it enhances the privacy protection.

Acknowledgments

This work was supported by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award "UTEP and Prudential Actuarial Science Academy and Pipeline Initiative" from Prudential Foundation.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.

[2] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques", *Proceedings of the Third IEEE International Conference on Data Mining ICDM'03*, Melbourne, Florida, USA, November 19–22, 2003, pp. 99–106.

[3] E. Klarreich, "Learning securely: because it is easy to fool, machine learning can be taught hot to handle adversarial inputs", *Communications of the ACM*, 2016, Vol. 59, No. 11, pp. 12–14.

[4] H. T. Nguyen, V. Kreinovich, B. Wu, and G. Xiang, *Computing Statistics under Interval and Fuzzy Uncertainty*, Springer Verlag, Berlin, Heidelberg, 2012.