Customizing Adversarial Machine Learning to test Deep Learning techniques

Temple, Paul; Perrouin, Gilles; Frenay, Benoît; Schobbens, Pierre-Yves

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Customizing Adversarial Machine Learning to Test Deep Learning Techniques

Paul Temple, Gilles Perrouin, Benoît Frénay and Pierre-Yves Schobbens
NADI, PReCISE, Faculty of Computer Science
University of Namur, Belgium

Over the past decade, machine learning (ML) and deep learning (DL) have achieved several breakthroughs, including Google's driverless cars, IBM Watson, deep learning for playing Go, to name just a few. The apparition of ML-based software in various systems also raises concerns on their correct operation. Indeed, a non-perceivable modification to an image can fool an ML algorithm to make incorrect predictions, e.g. recognize a gibbon instead of a panda [1]. Recently, it has been reported that a change in the contrast of the image of a road can lead a DL algorithm to turn right instead of left [2]. These examples, whether they are natural or deliberately engineered, are labeled as adversarial.

These cases have initiated a new trend in the ML community. Adversarial Machine Learning (advML) strives to understand, from security and safety points of view, how ML processes can be biased by the underlying training process or any further manipulations by potentially malicious people. The field is very active and the body of knowledge is growing [2], [3], [4], [5], [6]. Techniques usually focus on fooling ML algorithms by creating small changes with major impact on predictions. To do so, advML techniques usually take the role of attackers trying to pass through the defense that is represented by the ML decision system, trying to filter incoming data. AdvML relies on the design of ML algorithms and automatically targets specific aspects and weaknesses of the algorithm in order to craft new data, so as to make the ML decision system unusable as it will make too many mistakes in its predictions. Adversarial attacks can be used at training time or at exploitation time. Biggio *et al.* [7] have proposed an history of advML techniques which shows how rich and diverse these techniques can be. While first applications were about intrusion detection systems and spams, the field has quickly moved to handwritten digit recognition and are now widely used in DL systems [8].

On the other hand, variability-intensive systems form a vast and heterogeneous class of software systems that encompasses: software product lines (SPLs), configurable systems (operating systems kernels, web development frameworks/stacks, e-commerce configurators, code generators), Systems of Systems, software ecosystems (*e.g.*, Android's "Play Store"), *etc.* All these systems have the ability to be customized to specific needs (each customized system being called a variant) via the use of e.g. configurators. Since DL systems have various parameters, they can be seen as configurable as well: they can be configured in their architecture (with varying number of layers or number of neurons per layer), they can be convolutional or recurrent, activation functions of neurons can be customized, intermediate layers can be defined (using pooling or different kernel sizes in the convolution), *etc.*

If DL based systems can be configured, so are advML techniques. In this talk, we argue that to test DL systems with such diversity, one should be able to configure advML techniques so that they are the most efficient and appropriate for the DL under test. Since each DL algorithm can be confronted to various advML techniques, we might be able to assess the sensitivity of specific DL algorithms to certain advML attacks and thus provide guidelines in the choice of the DL system to use in a certain context which favor one or the other kind of attack. Our goal is to design a framework in which we would be able to combine different advML techniques, in which we can create new, more powerful attacks that can test more efficiently ML and DL algorithms. We think that the expertise developed in the software variability research these last 15 years can be employed to develop (auto)-configurable advML techniques. We will illustrate a few scenarios and exhibit a research agenda that involve both software variability and machine learning communities to work together.

Gilles Perrouin is a research associate at the FNRS. This research was partially funded by the FNRS Grant O05518F-RG03.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples. corr (2015)."
[2] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, "Deeproad: Gan-based metamorphic autonomous driving system testing," *arXiv preprint arXiv:1802.02295*, 2018.
[3] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," *CoRR*, vol. abs/1708.08559, 2017.
[4] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on deep learning models," *arXiv preprint arXiv:1707.08945*, vol. 1, 2017.
[5] G. F. Elsayed, S. Shankar, B. Cheung, N. Papernot, A. Kurakin, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial examples that fool both human and computer vision," *arXiv preprint arXiv:1802.08195*, 2018.
[6] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017.
[7] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *arXiv preprint arXiv:1712.03141*, 2017.
[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.