Robustness Verification of Support Vector Machines

Marco Zanella
mzanella@math.unipd.it

Dipartimento di Matematica, University of Padova, Italy

Research area: machine learning, support vector machines, formal methods, robustness, abstract interpretation

1 Addressed problem and related work

This work addresses the problem of formally verifying robustness of support vector machines (SVMs), a major machine learning model for classification and regression tasks. Robustness properties asserts whether a model produces similar outputs on similar inputs, which is a key concept in adversarial machine learning [7,11,24], an emerging hot topic studying vulnerabilities of machine learning (ML) techniques in adversarial scenarios whose main objective is to design methodologies for making learning tools robust to adversarial attacks. Adversarial examples have been found in diverse application fields of ML such as image classification, speech recognition and malware detection [7]. Current defense techniques include adversarial model training, input validation, testing and automatic verification of learning algorithms [7]. In particular, formal verification of ML classifiers started to be an active field of investigation [1,5,6,8,9,10,14,16,17,19,20,25,26] within the verification and static analysis community. Formal verification of robustness to adversarial inputs has been investigated for neural networks [1,6,16,19,20,26]. A classifier is robust to some perturbation of its input objects representing an adversarial attack when it assigns the same class to all the objects within that perturbation. Thus, slight malicious alterations of input objects should not deceive a robust classifier. Pulina and Tacchella [16] first put forward the idea of a formal robustness verification of neural network classifiers by leveraging interval-based abstract interpretation for designing a sound abstract classifier. This abstraction-based verification approach has been pushed forward by Vechev et al. [6,19,20], who designed a scalable robustness verification technique which relies on abstract interpretation of deep neural networks based on a specifically tailored abstract domain [20].

While all the aforementioned verification techniques consider neural networks, this work focuses on SVMs [4], which are widely applied in different fields where adversarial attacks must be taken into account, notably image classification, malware detection, intrusion detection and spam filtering [2]. Adversarial attacks and robustness issues of SVMs have been defined and studied by some authors [2,3,15,23,27,29,30] investigating robust training and empirical robustness evaluation of SVMs. To the best of our knowledge, no formal and automatic robustness certification technique for SVMs has been studied.

2 Proposed solution

The proposed approach relies on a sound abstract version of a given SVM classifier to be used for checking its robustness. This methodology is parametric on a given numerical abstraction of real values and, analogously to the case of neural networks, needs neither abstract least upper bounds nor widening operators on this abstraction. The standard interval domain provides a simple instantiation of our abstraction technique, which is enhanced with the domain of reduced affine forms, an efficient abstraction of the zonotope abstract domain. This robustness verification technique has been fully implemented in a tool named *SAVer* (**S**VM **A**bstract **Ver**ifier), which is available at [18]. With this tool it is possible to experimentally evaluate robustness of SVMs based on linear and nonlinear (polynomial and radial basis function) kernels, which have been trained on the popular MNIST dataset of images and on the recent and more challenging Fashion-MNIST dataset. The experimental results of our SVM robustness verifier appear to be encouraging: this automated verification is fast, scalable and shows significantly high percentages of provable robustness on the test set of MNIST, in particular compared to the analogous provable robustness of neural networks.

3 Methodology

This work considers a standard per-sample robustness notion in the field of machine learning: a classifier $C : X \to L$ is seen as a function from the input space X to a set of labels L, a perturbation $P : X \to \wp(X)$ is a function mapping a sample to a set of similar samples, a classifier C is said to be robust on a sample $\mathbf{x} \in X$ w.r.t. a perturbation P when every sample in $P(\mathbf{x})$ is classified in the same way as $\mathbf{x}$:

$$\mathit{Robust}(C, \mathbf{x}, P) \Leftrightarrow \forall \mathbf{x}' \in P(\mathbf{x}) \colon C(\mathbf{x}') = C(\mathbf{x})$$

in principle, running this test on every sample in the testing set allows to estimate the probability of a classifier to be robust. However, $P(\mathbf{x})$ is usually either an infinite of unfeasible to compute set of points, making a concrete test impossible.

To overcome this issue, one can abstract the set $P(\mathbf{x})$ with a single abstract value $P^\sharp(\mathbf{x}) \in A$, where A is the abstract domain of choice, such that $P(\mathbf{x}) \subseteq \gamma(P^\sharp(\mathbf{x}))$, then compute a superset of the labels of points in $\gamma(P^\sharp(\mathbf{x}))$ using a sound abstract version of the concrete classifier $C^\sharp : A \to \wp(L)$. By relying on the standard notion of *soundness* in the field of abstract interpretation, it is possible to show that

$$|C^\sharp(P^\sharp(\mathbf{x}))| = 1 \Rightarrow \mathit{Robust}(C, \mathbf{x}, P)$$

as $|C^\sharp(P^\sharp(\mathbf{x}))| = 1$ implies that the superset of the labels of samples in $P(\mathbf{x})$ is a singleton, hence every sample is classified in the same way. This approach has the advantage of being fast and efficient to compute, since an otherwise unfeasible computation is performed symbolically on a single abstract value. On the other hand, the abstract classifier can only compute a superset of the actual labels, hence providing a sufficient but non necessary condition. Whenever $C^\sharp(P^\sharp(\mathbf{x}))$ allows to assert $Robust(C, \mathbf{x}, P)$, that assertion is definitively true, but not vice-versa: it may be the case that a classifier is robust on some input for some perturbation, but the abstract analysis is not able to prove that. This notion is well-known in the field of abstract interpretation, and it is referred to as *incompleteness*.

While the aforementioned strategies can be applied to any type of classifier, SAVer focuses on SVMs. It turned out that a sound abstract SVM classifier can be built by finding appropriate sound abstract transfer functions for some standard operators (sum, multiplication, sign), for the kernel functions (scalar product, radial basis function, polynomial) and, only in the case of multi-label classification, for the voting mechanism used by the SVM.

While this approach shares some similarities with standard program analysis, there are also some relevant differences. First an foremost, it is possible to rewrite the code of an SVM avoiding branching and loops. This allows to avoid computation of least upperbounds and widening in the abstract classifier, which would cause loss of precision. Moreover, SVMs exhibit patterns which are not common in program analysis, and for which simple abstract domains such as the intervals do not perform well, like expressions $\mathbf{x} - \mathbf{x}$. To overcome this limitation, SAVer deploys an abstract domain based on affine forms [13,22]. This aspect has been further improved by observing that some noise symbols introduced by the transfer functions can be compacted into *reduced* affine forms, as described in [21], saving memory space and computational time.

4 Experimental results

Findings presented in this work has been implemented in a tool called *SAVer* (**S**VM **A**bstract **Ver**ifier), written in C, and made available on GitHub [18]. SAVer has been used to estimate robustness of state-of-the-art classifiers for the popular MNIST [12] image dataset and the recent and more challenging alternative Fashion-MNIST dataset [28]. Both datasets contain gray scale images of 28×28 pixels, represented by normalized vectors of floating-point numbers in $[0, 1]^{784}$. The perturbation models chosen for the tests were L_∞-norm perturbations with increasing (relative) magnitudes. Benchmarks show the percentage of samples of the full test sets for which a SVM is proved to be robust (and, dually, vulnerable) for a given perturbation, the average verification times per sample, and the scalability of the robustness verifier w.r.t. the number of support vectors.

Fig. 1 (left) shows percentage of samples for which SAVer managed to prove robustness w.r.t increasing magnitude of an L_∞ perturbation. Different curves correspond to different kernels, hence different SVM models. Fig. 1 (right) compares provable robustness using the RBF kernel on MNIST against Fashion-MNIST datasets, suggesting that training a robust classifier for the latter is more challenging.

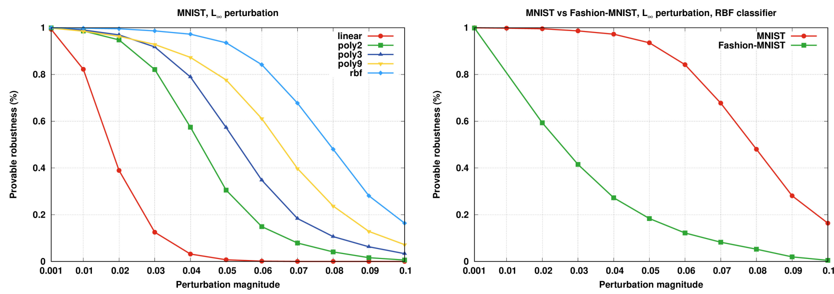

Fig. 1. Robustness under L_∞ perturbations

Tab. 1 reports percentage of provable robustness and execution times for the RBF-based classifier on MNIST, under a L_∞ perturbation with increasing magnitude. As expected, robustness becomes harder to prove (and to achieve) with higher perturbation magnitudes. On the other hand, it turns out that computational cost is not affected by the magnitude, as it is the case for DeepPoly and other similar tools.

Magnitude	Probable robustness (%)	Time per image (ms)
0.01	99.83%	417.18
0.02	99.57%	415.95
0.03	99.19%	417.19
0.04	97.27%	416.98
0.05	93.58%	417.69
0.06	82.21%	417.21
0.07	67.76%	416.93
0.08	48.02%	417.21
0.09	28.10%	417.15
0.10	16.38%	417.97

Table 1. Execution times for an RBF classifier on MNIST, L_∞ perturbation

Results can be compared to those of DeepPoly [20], a robustness verification tool for deep neural networks based on abstract interpretation. As DeepPoly is based on a different model, a strict comparison is not possible. It is however fair to state that SAVer is at least competitive in terms of provable robustness, and clearly outperforms the latter in terms of execution speed, as it can take over 10 seconds to produce an answer ([20]).

References

1. G. Anderson, S. Pailoor, I. Dillig, and S. Chaudhuri. Optimization + abstraction: A synergistic approach for analyzing neural network robustness. In *Proc.*

ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI2019). ACM, 2019.
2. B. Biggio, I. Corona, B. Nelson, B. I. P. Rubinstein, D. Maiorca, G. Fumera, G. Giacinto, and F. Roli. Security evaluation of support vector machines in adversarial environments. In Y. Ma and G. Guo, editors, *Support Vector Machines Applications*, pages 105–153. Springer, 2014.
3. B. Biggio, B. Nelson, and P. Laskov. Support vector machines under adversarial label noise. In *Proceedings of the 3rd Asian Conference on Machine Learning (ACML2011)*, pages 97–112, 2011.
4. N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, 2000.
5. R. Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *Proc. 15th Intern. Symp, on Automated Technology for Verification and Analysis (ATVA2017)*, pages 269–286, 2017.
6. T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. T. Vechev. AI2: safety and robustness certification of neural networks with abstract interpretation. In *Proc. 2018 IEEE Symposium on Security and Privacy (SP2018)*, pages 3–18, 2018.
7. I. Goodfellow, P. McDaniel, and N. Papernot. Making machine learning robust against adversarial inputs. *Commun. ACM*, 61(7):56–66, 2018.
8. D. Gopinath, G. Katz, C. S. Pasareanu, and C. Barrett. DeepSafe: A data-driven approach for assessing robustness of neural networks. In *Proceedings of the 16th Int. Symp. on Automated Technology for Verification and Analysis (ATVA2018)*, pages 3–19, 2018.
9. X. Huang, M. Kwiatkowska, S. Wang, and M. Wu. Safety verification of deep neural networks. In R. Majumdar and V. Kunčak, editors, *Proc. Intern. Conf. on Computer Aided Verification (CAV2017)*, pages 3–29. Springer, 2017.
10. G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In R. Majumdar and V. Kunčak, editors, *Proc. Intern. Conf. on Computer Aided Verification (CAV2017)*, pages 97–117. Springer, 2017.
11. A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In *Proceedings of the 5th International Conference on Learning Representations (ICLR2017)*, 2017.
12. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
13. F. Messine. Extentions of affine arithmetic: Application to unconstrained global optimization. *J. Universal Computer Science*, 8(11):992–1015, 2002.
14. M. Mirman, T. Gehr, and M. Vechev. Differentiable abstract interpretation for provably robust neural networks. In *Proc. of the International Conference on Machine Learning (ICML2018)*, pages 3575–3583, 2018.
15. G. P. Nam, B. J. Kang, and K. R. Park. Robustness of face recognition to variations of illumination on mobile devices based on SVM. *KSII Transactions on Internet and Information Systems*, 4(1):25–44, 2010.
16. L. Pulina and A. Tacchella. An abstraction-refinement approach to verification of artificial neural networks. In T. Touili, B. Cook, and P. Jackson, editors, *Proc. of the Intern. Conf. on Computer Aided Verification (CAV2010)*, pages 243–257. Springer, 2010.
17. L. Pulina and A. Tacchella. Challenging SMT solvers to verify neural networks. *AI Commun.*, 25(2):117–135, 2012.

18. F. Ranzato and M. Zanella. SAVer GitHub Repository. https://github.com/svm-abstract-verifier, 2019.
19. G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. T. Vechev. Fast and effective robustness certification. In *Advances in Neural Information Processing Systems 31: Proc. Annual Conference on Neural Information Processing Systems 2018, (NeurIPS2018)*, pages 10825–10836, 2018.
20. G. Singh, T. Gehr, M. Püschel, and M. Vechev. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3(POPL2019):41:1–41:30, Jan. 2019.
21. I. Skalna and M. Hladík. A new algorithm for Chebyshev minimum-error multiplication of reduced affine forms. *Numerical Algorithms*, 76(4):1131–1152, Dec 2017.
22. J. Stolfi and L. H. de Figueiredo. Affine arithmetic: Concepts and applications. *Numerical Algorithms*, 37(1):147–158, Dec 2004.
23. T. B. Trafalis and R. C. Gilbert. Robust support vector machines for classification and computational issues. *Optimisation Methods and Software*, 22(1):187–198, 2007.
24. Y. Vorobeychik and M. Kantarcioglu. Adversarial machine learning. In *Synthesis Lectures on Artificial Intelligence and Machine Learning*, volume 12(3), pages 1–169. Morgan & Claypool Publishers, August 2018.
25. S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana. Formal security analysis of neural networks using symbolic intervals. In *Proceedings of the 27th USENIX Conference on Security Symposium*, (SEC2018), pages 1599–1614. USENIX Association, 2018.
26. T. Weng, H. Zhang, H. Chen, Z. Song, C. Hsieh, L. Daniel, D. S. Boning, and I. S. Dhillon. Towards fast computation of certified robustness for ReLU networks. In *Proceedings of the 35th International Conference on Machine Learning, (ICML2018)*, pages 5273–5282, 2018.
27. H. Xiao, B. Biggio, B. Nelson, H. Xiao, C. Eckert, and F. Roli. Support vector machines under adversarial label contamination. *Neurocomputing*, 160:53–62, 2015.
28. H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.
29. H. Xu, C. Caramanis, and S. Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10:1485–1510, 2009.
30. Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and B. Xi. Adversarial support vector machine learning. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2012)*, pages 1059–1067. ACM, 2012.