Verification of Neural Networks for Safety and Security-critical Domains

Dario Guidotti[1,*]

[1]University of Sassari, Piazza Università 21, Sassari, 07100, Italy

Abstract

In recent times, machine learning has gained incredible traction in the artificial intelligence community, and neural networks in particular have been leveraged in many successful applications originating from various domains. However, it is hard to provide any formal guarantee on the behavior of this kind of models, and therefore their reliability is still in doubt, especially concerning their deployment in safety and security-critical applications. In this work, we will present our contributions on the topic of formal verification, which recently emerged as a promising solution to address some of these problems. We will also present two novel use cases originating from real-world applications we are working on and the related challenges and perspectives.

Keywords

Trustworthy AI, Neural Networks, Formal Verification

1. Introduction

In the last few decades, artificial intelligence (AI) has become increasingly popular as it has been employed in many different applications with a great degree of success [1, 2]. Among those domains, safety and security-critical ones are often of particular interest both to the research and industry communities. As an example, the automotive domain has seen increasingly substantial investments in financial and time resources from both. However, industrial applications in this kind of domain require formal guarantees on the behavior of the algorithms and models used, since thorough regulations have been established by national and international authorities regarding the employment of AI in these areas. Unfortunately, the currently most popular AI technologies, that is, machine learning (ML) and neural networks, provide only statistical guarantees on their behavior which, while may be enough for many applications of interests, still fall short when human lives, or large amounts of money, are at stake. Furthermore, neural networks have been proven to be subject to reliability issues like adversarial attacks, which are small variations of the inputs which cause unforeseeable changes in their behavior. It is easy to see how this kind of vulnerability can be both a safety and a security issue with a couple of examples from the automotive domain: burned pixels in the cameras may cause the misclassification of a curve as a straight road [3], or small graffiti written by a malevolent actor on

IPS-RiCeRcA-SPIRIT 2022: 10th Italian Workshop on Planning and Scheduling, RiCeRcA Italian Workshop, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy.

*Corresponding author.

dguidotti@uniss.it (D. Guidotti)

0000-0001-8284-5266 (D. Guidotti)

Figure 1: Example of an adversarial attack in the automotive domain from [3]: a few burned pixels in a camera can easily mislead a neural network to misclassify a left turn as a right one.

a stop sign may make the model recognize it as a speed limit [4]. As a consequence, part of the AI research community focused on developing methodologies to formally evaluate the correctness of the behavior of neural networks and, as witnessed in [5] automated formal verification seems to provide a path to their adoption in safety and security-critical applications [6]. In this paper, we will present some of our contributions to this last topic, their prospective applications, and the open challenges we are facing in the scope of some new use cases originating from realistic domains. Section 2 present the relevant background and state of the art for verification of neural networks. In Section 3 we briefly explain our contribution in the domain of verification of neural networks, whereas in Section 4 we present the new use cases which are guiding our recent research efforts. Finally, in Section 5 we summarize the challenges we are facing with the new use cases and our perspectives on how to overcome them.

2. Background

2.1. Neural Networks

Neural networks are machine learning models inspired by the structure of biological neural networks: more specifically they can be seen as directed graphs composed of interconnected computing units called neurons. In a feed-forward network, the neurons are arranged in disjointed layers and each layer is connected only with the following ones forming a direct acyclic graph (DAG). Every layer of a neural network performs specific computations on the inputs it receives from the previous layers or, for the first layer of the network, from the outside. The computational complexity of a layer is directly correlated with both the number of neurons in such layer and the specific operation carried out by them. It clearly follows that the same quantities can be used to estimate the complexity of a whole network and, indeed, they are often used to formally define such complexity.

Some examples of particularly important layers are the *linear* and *activation* layers, which are normally the building blocks used to assemble even the most basic neural networks. In particular, linear layers apply an affine transformation to their input tensor, whereas activation layers apply a specific (usually non-linear) function element-wise. Even with only these two kinds of layers, it is possible to build neural networks which can, in principle, approximate any

continuous function for inputs within a specific range [7].

Finally, another significant layer is the convolutional one, which has been especially successful when used in architectures applied to computer vision tasks. The idea behind the convolutional layer is to analyze the input images using a set of perception filters (also known as feature maps) which can be learned from the data.

2.2. Verification

The aim of formal verification is to provide guarantees regarding the behavior of neural networks: more specifically verification methodologies try to prove if specific neural networks satisfy stated input-output relations. For this purpose, in the last decade, several verification methodologies have been proposed for different specifications and architectures [8, 9, 10, 11, 5].

A common distinction between verification methodologies is the one between *complete* and *incomplete* algorithms: complete verification algorithms [12, 13, 14, 15, 16] leverage techniques like Branch and Bound, Satisfiability Modulo Theories (SMT) and Mixed Integer Linear Programming (MILP) to provide a final answer on the compliance of the neural network to the property of interest, at the price of a greatly increased computational complexity. On the other hand, incomplete verification methodologies [17, 18, 19, 20, 21] are typically based on methods like abstract interpretation and bound propagation and, as consequence, they provide an answer subject to a degree of uncertainty. That is, when an incomplete method is unable to certify that a network satisfies the property of interest, it is still unsure if the property is truly violated or if the precision of the approximation used was not good enough. However, when an incomplete algorithm certifies that the network of interest satisfies a certain property, it is certain that the network behavior is compliant with it. While the use of over-approximate methods causes this kind of uncertainty it also produces algorithms whose computational complexity is greatly reduced.

While this categorization is still commonly used, it should be noted that, in recent times, many verification tools began to combine complete and incomplete algorithms to get, as much as possible, the best of both worlds.

3. Contributions

In [22] we investigated if pruning, a methodology developed by the learning community whose main application until now has been to reduce the dimension of neural networks so that they can be deployed on hardware with low memory resources, could be leveraged to produce a training pipeline providing networks easier to verify for the existing verification methodologies. In our experiments, we considered two different pruning methodologies: one based on the reduction of the number of connections between the neurons of a neural network, called weight pruning, and the other based on the removal of whole neurons, called neuron pruning. By leveraging these two pruning algorithms to reduce different networks, we managed to produce models with performances comparable to the original ones but, at the same time, much easier to verify for existing verification tools. In particular, the networks on which neuron pruning was applied were the easiest to verify: we believe this is because the elimination of whole neurons

from the networks eliminates a corresponding number of non-linearities, which are, in general, the main culprit behind the high computational complexity of verification algorithms.

In [17] we developed the first version of our tool pyNeVer which provides capabilities for the training, pruning, and verification of neural networks. The main contribution of this work was a novel algorithm based on over-approximation for the verification of neural networks: in particular, the novelty of this algorithm is in the mechanism to dynamically control the coarseness of the over-approximation down to the single neuron level and a specific eager heuristic for choosing the neurons on which to apply the more precise abstraction. The resulting algorithm resulted to be comparable with the state of the art and even outperforms a similar one on some of our experimental benchmarks. Further experiments were done in [6] on models generated from an automotive case study.

Finally, in [23] we focused on trying to repair neural networks subject to adversarial examples so that they became more robust to particular adversarial examples. To do so we leveraged a MILP solver to find a configuration of the network parameters which made it resistant to the adversarials of interest. However, most of the networks were too complex to be directly modified with such a methodology. Therefore, we selected a specific subset of the network architecture and replaced it with a less complex model which then was repaired. In our experimental evaluation we confirmed that, while this methodology did not manage to make the model resistant to adversarial attack in general, it did make it more robust to specific adversarial examples.

4. Use Cases

Currently, we are working on two new use cases (UCs) from safety and security-critical domains. In the first one, the task is developing reliable neural controllers for self-piloting a drone during the course of different activities. The second one consists in developing reliable neural networks for object detection and recognition tasks in the automotive domain. This second use case is part of the effort to develop reliable neural networks for safety-critical contexts in the scope of the AIDOaRt Project, a 3 years long H2020-ECSEL European project focusing on Artificial Intelligence augmented automation supporting modeling, coding, testing, monitoring, and continuous development of Cyber-Physical Systems.

4.1. Drone Control

Since the scope of the project was to develop various neural controllers for different tasks, we focused on building a modular setup for the training of neural controllers in simulated environments using well-maintained and stable resources. In particular, we leveraged:

- Gym[1]: an open-source python library providing a standard API for communication between reinforcement learning algorithms and environments.
- Stable Baselıne3[2]: an open source training framework providing scripts for training and evaluating RL agents using standard state-of-the-art algorithms.

[1]https://github.com/openai/gym
[2]https://github.com/DLR-RM/stable-baselines3

Figure 2: The Bitcraze Crazyflie 2.1 drone considered in our first use case.

- PyBullet[3]: an open source physics simulator for robotics and reinforcement learning.
- gym-pybullet-drones[4]: an open source Gym-stile environment supporting the definition of various learning tasks on the control of one or more quadcopters.

Using these open-source resources we greatly simplified the complexity of our setup and we were able to directly train the network of interest in the environment corresponding to our case study with the chosen state-of-the-art RL algorithm. In Figure 2 we show the quadcopter model of choice, which was the default one proposed in gym-pybullet-drones and which we intend to use to test our neural controllers in real environments.

Beyond the scope of the project, we also intend to leverage the presented setup to produce novel benchmarks for the verification of neural networks. The motivation for doing so is that, while the verification community has been prolific in developing novel methodologies, very few general benchmarks have been proposed, among which the most popular is still the ACAS XU benchmark [24], released in 2017. Furthermore, drone control is a task relevant to modern applications and, at the same time, the neural networks used in this kind of control task are usually small enough for the existing verification methodologies to be successfully applied.

4.2. Automotive

One of the use cases on which the technologies developed within the AIDOaRt project will be evaluated is the one related to the automotive domain proposed by Abinsula Srl, a company based in Sassari (Italy) that provides innovative ICT solutions worldwide [5].

Given the wealth of interconnected sensors and software supports modern cars can be easily seen as cyber-physical systems. This implies that the various stakeholders in the development

[3]https://pybullet.org/
[4]https://github.com/utiasDSL/gym-pybullet-drones
[5]https://abinsula.com/

process of such systems have to work together to ensure that the corresponding product is safe and reliable. While the methodologies to guarantee the safety and reliability of the hardware and much of the software are quite well established, the same cannot be said for the components based on artificial intelligence and machine learning, whose formal certification is still an open challenge for both the industrial and research communities.

The use case brought by Abinsula aims at enhancing the human interaction and driving experience, proposing an electronic rear-view mirror that gets data from a set of cameras and provides the rear image on a screen. The application of AI and ML to sensor data processing allows for increasing the informative content of the rear environment image providing, for example, alerts or suggestions for a safer and more effective drive. The use case involves the usage of four cameras that capture 1920x1080 images in an up to 60 fps stream and also provide the relevant camera ID. The idea behind the usage of neural networks in this kind of image processing is to reduce the effort the human user needs to apply for recognizing and processing the relevant objects in the image they are seeing. As consequence, a fundamental step to implement a virtual rear mirror able to reliably assist the human user is the formal verification of the adopted neural networks to ensure the predictability of the system.

5. Challenges and Perspectives

Given the scope of the drone control UC we first analyzed the state of the art regarding the learning of neural controllers for the kind of tasks we had in mind and we selected a promising reinforcement learning algorithm, that is, the Soft Actor-Critic. The main advantage of this algorithm is that it uses two different neural networks: the actor one models the controller, whereas the critic is used to provide an estimation of how good the actor is. At high level this allows us to keep down the complexity of the controller (*i.e.*, the actor) that therefore will be easier to verify. Even so, clearly more complex tasks will require more complex architectures and therefore we will need to enhance both the scalability and the generality of pyNeVer. Furthermore, we are interested in developing methodologies which would allow to leverage the results obtained by an unsuccessful verification of the neural controllers to enhance their training process so that they became compliant with the property of interest.

Regarding the Abinsula UC, first we focused on the state of the art regarding neural networks applied in computer vision tasks and we identified an initial set of architectures that could be viable for the applications of interest. In particular, we identified the YOLO [25] network architecture, which is one of the most popular learning models used for object detection. This architecture is able to process videos at 45 frames per second (fps), whereas a more optimized version manages to reach 150 fps. Furthermore, it greatly outperforms the contemporary models leveraging classification and learns better generalizable representations of objects. We also surveyed the state of the art of the verification tools, and we focused on the winner of the 2nd International Verification of Neural Networks Competition [26] (VNN-COMP'21) alpha-beta-crown [27], which is a neural network verifier based on an efficient bound propagation algorithm and a branch and bound methodology. It also leverages dedicated hardware (i.e., GPUs) in order to scale to relatively large convolutional networks and supports a wide range of

architectures. Of course, we also considered the runner-ups of the same competition, which present comparable performances to alpha-beta-crown. Nevertheless, from a first comparison between the benchmarks used during the VNN-COMP'21 (11 convolutional layers) and the YOLO architecture (109 convolutional layers in its last version), it would seem that the current state-of-the-art verification tools are far from reaching the scalability needed to support our preferred architectures. As consequence we focused on identifying various strategies to bridge this gap between our favored architectures and the effective scalability of verification tools. To do so we believe that it could be possible to focus our verification on subsets of the network architecture which are of particular interest for the task at hand and, at the same time, small enough to be feasible to verify, similarly to what was done in [23]. We also intend to investigate pruning and/or quantization as means to produce smaller network models which should enable verification without a significant loss in performances, as we managed to do with less complex models in [22]. Finally, as for the drone control UC, we are evaluating how to enhance existing verification tools and methodologies to support network architectures similar to YOLO.

Acknowledgments

This research work has received funding through the AIDOaRt project from the ECSEL Joint Undertaking (JU) under grant agreement No 101007350. The JU receives support from the European Union's Horizon 2020 research and innovation program and Sweden, Austria, Czech Republic, Finland, France, Italy, and Spain.

The research on reliable drone control has been supported by Fondazione di Sardegna, project "Tecniche e strumenti per la verifica di reti neurali".

References

[1] E. Giunchiglia, A. Nemchenko, M. van der Schaar, RNN-SURV: A deep recurrent model for survival analysis, in: V. Kurková, Y. Manolopoulos, B. Hammer, L. S. Iliadis, I. Maglogiannis (Eds.), Artificial Neural Networks and Machine Learning - ICANN 2018 - 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III, volume 11141 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 23–32.

[2] Y. LeCun, Y. Bengio, G. E. Hinton, Deep learning, Nat. 521 (2015) 436–444.

[3] K. Pei, Y. Cao, J. Yang, S. Jana, Deepxplore: automated whitebox testing of deep learning systems, Commun. ACM 62 (2019) 137–145.

[4] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, D. Song, Robust physical-world attacks on deep learning visual classification, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 1625–1634.

[5] F. Leofante, N. Narodytska, L. Pulina, A. Tacchella, Automated verification of neural networks: Advances, challenges and perspectives, CoRR abs/1805.09938 (2018).

[6] S. Demarchi, D. Guidotti, A. Pitto, A. Tacchella, Formal verification of neural networks: A

case study about adaptive cruise control, in: I. A. Hameed, A. Hasan, S. A. Alaliyat (Eds.), Proceedings of the 36th ECMS International Conference on Modelling and Simulation, ECMS 2022, Ålesund, Norway, May 30 - June 3, 2022, European Council for Modeling and Simulation, 2022, pp. 310–316.

[7] K. Hornik, M. B. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (1989) 359–366.

[8] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, X. Yi, A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability, Comput. Sci. Rev. 37 (2020) 100270.

[9] D. Guidotti, Enhancing neural networks through formal verification, in: M. Alviano, G. Greco, M. Maratea, F. Scarcello (Eds.), Discussion and Doctoral Consortium papers of AI*IA 2019 - 18th International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, November 19-22, 2019, volume 2495 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 107–112.

[10] D. Guidotti, Verification and repair of neural networks, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, 2021, pp. 15714–15715.

[11] D. Guidotti, Safety analysis of deep neural networks, in: Z. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, ijcai.org, 2021, pp. 4887–4888.

[12] G. Katz, D. A. Huang, D. Ibeling, K. Julian, C. Lazarus, R. Lim, P. Shah, S. Thakoor, H. Wu, A. Zeljic, D. L. Dill, M. J. Kochenderfer, C. W. Barrett, The marabou framework for verification and analysis of deep neural networks, in: Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I, volume 11561 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 443–452.

[13] P. Henriksen, A. R. Lomuscio, Efficient neural network verification via adaptive refinement and adversarial search, in: G. D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín, J. Lang (Eds.), ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2020, pp. 2513–2520.

[14] P. Henriksen, A. Lomuscio, DEEPSPLIT: an efficient splitting method for neural network verification via indirect effect analysis, in: Z. Zhou (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, ijcai.org, 2021, pp. 2549–2555.

[15] R. Bunel, J. Lu, I. Turkaslan, P. H. S. Torr, P. Kohli, M. P. Kumar, Branch and bound for piecewise linear neural network verification, J. Mach. Learn. Res. 21 (2020) 42:1–42:39.

[16] A. D. Palma, R. Bunel, A. Desmaison, K. Dvijotham, P. Kohli, P. H. S. Torr, M. P. Kumar, Improved branch and bound for neural network verification via lagrangian decomposition, CoRR abs/2104.06718 (2021).

[17] D. Guidotti, L. Pulina, A. Tacchella, pynever: A framework for learning and verification of neural networks, in: Z. Hou, V. Ganesh (Eds.), Automated Technology for Verification and Analysis - 19th International Symposium, ATVA 2021, Gold Coast, QLD, Australia, October 18-22, 2021, Proceedings, volume 12971 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 357–363.

[18] L. Pulina, A. Tacchella, Never: a tool for artificial neural networks verification, Ann. Math. Artif. Intell. 62 (2011) 403–425.

[19] G. Singh, T. Gehr, M. Püschel, M. T. Vechev, An abstract domain for certifying neural networks, Proc. ACM Program. Lang. 3 (2019) 41:1–41:30.

[20] H. Tran, X. Yang, D. M. Lopez, P. Musau, L. V. Nguyen, W. Xiang, S. Bak, T. T. Johnson, NNV: the neural network verification tool for deep neural networks and learning-enabled cyber-physical systems, in: S. K. Lahiri, C. Wang (Eds.), Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part I, volume 12224 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 3–17.

[21] H. Zhang, T. Weng, P. Chen, C. Hsieh, L. Daniel, Efficient neural network robustness certification with general activation functions, in: S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 2018, pp. 4944–4953.

[22] D. Guidotti, F. Leofante, L. Pulina, A. Tacchella, Verification of neural networks: Enhancing scalability through pruning, in: G. D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín, J. Lang (Eds.), ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2020, pp. 2505–2512.

[23] D. Guidotti, F. Leofante, L. Pulina, A. Tacchella, Verification and repair of neural networks: A progress report on convolutional models, in: M. Alviano, G. Greco, F. Scarcello (Eds.), AI*IA 2019 - Advances in Artificial Intelligence - XVIIIth International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, November 19-22, 2019, Proceedings, volume 11946 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 405–417.

[24] G. Katz, C. W. Barrett, D. L. Dill, K. Julian, M. J. Kochenderfer, Reluplex: An efficient SMT solver for verifying deep neural networks, in: R. Majumdar, V. Kuncak (Eds.), Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I, volume 10426 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 97–117.

[25] J. Redmon, S. K. Divvala, R. B. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, IEEE Computer Society, 2016, pp. 779–788.

[26] S. Bak, C. Liu, T. T. Johnson, The second international verification of neural networks competition (VNN-COMP 2021): Summary and results, CoRR abs/2109.00498 (2021).

[27] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C. Hsieh, J. Z. Kolter, Beta-crown: Efficient

bound propagation with per-neuron split constraints for neural network robustness verification, in: M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, 2021, pp. 29909–29921.