Enhancing Neural Networks through Formal Verification

Dario Guidotti[0000−0001−8284−5266]

University of Genoa - DIBRIS, 16126 Genoa, Italy
dario.guidotti@edu.unige.it

Abstract. In this work we present an overview of our current research activities. Our work lies at the intersection between Formal Verification and Neural Networks.

Keywords: Neural Networks · Formal Verification.

1 Introduction and State of the Art

From face recognition to automated financial crimes detection and cancer diagnosis, the domains in which successful applications for Neural Networks (NNs) have been found are many [1]. NNs do not provide any formal guarantee on their behaviour, therefore their adoption in safety and security-critical domain is still somehow limited, specifically in industrial applications, where hard certifications are at least desired and may be mandatory, *e.g.*, automotive domain. Moreover, in the last few years, the concerns about the robustness of NNs turned out to be legitimate: since the discovery of the vulnerability to adversarial perturbations [6] the research community realised that NNs may not be reliable. In the past years, more and more examples of this weakness have been discovered [3]. The machine learning community usually considers the robustness of NNs concerning adversarial samples: the broadest definition of adversarial sample is a perturbed input which brings the NN to an incorrect behaviour. In the boundaries of this definition, many kinds of adversarial samples can be found: for a systematic study on adversarial samples, we refer to [5]. This increased awareness of the limited reliability of NNs in the research community led to an increased interest in their verification. In [11] more than 170 papers about NNs verification were surveyed, most of them published between 2017 and 2018. To verify properties of NNs many different kinds of verification techniques have been developed in the last few years: we classify them following [11], which divides them with respect to the type of guarantees they can provide. Such guarantees can be *exact deterministic*, *one-sided*, with *converging bounds* or *statistical*. *Deterministic guarantees* are proved by transforming the verification problem in a set of constraints which are then solved using a constraint solver. *One-sided guarantees* consider the computation of a lower (or by duality, an upper) bound, and can claim the sufficiency of achieving properties. *Converging bounds guarantees* consider the computation of converging upper and lower bounds and therefore they can work with both output reachability property and interval property. *Statistical guarantees* are achieved using approaches which claim that the property of interest is respected with a certain probability. For a more in-depth survey on formal verification applied to NNs we refer to [11] and [16]. In spite of the extensive research done on NNs verification the state-of-the-art methods and tools are still far from being able to successfully verify the corresponding state-of-the-art NNs: even when a fairly large network is successfully verified, the results are often not relevant for real-world applications (*e.g.*, the property verified has been simplified too much to be relevant). Some examples of state-of-the-art tools are Marabou [13], which leverages a Satisfiability Modulo Theories (SMT) solver to deal with both fully connected and convolutional networks, ERAN [4], which is based on abstract interpretation and also deals with both kind of networks and MIPVerify [17] which leverage a Mixed Integer Programming solver to verify both typologies networks. The remainder of this document is organized as follows. Section 2 introduces the research questions of interest. Section 3 defines the goals which will be pursued during the Ph.D., while Section 4 briefly reports on preliminary results obtained.

2 Problem Definition

At the best of our knowledge, most of the general-purpose methods and tools for the verification of NNs are not scalable and they usually need massive computational resources. This introduces the first and arguably the most important research question of interest in our work:

Q1*: How can we make verification techniques scalable enough to verify current state-of-the-art NNs?*

Scalability is only one of many problems which limit the application of verification to real-world/industrial NNs: many of the current state-of-the-art tools work only for specific NNs architectures and activation functions. Moreover they usually accept models generated with specific learning frameworks and saved with a specific format. The second research question of interest is thus:

Q2*: How can we provide a general-purpose verification tool?*

Another limit we have noticed in the current state-of-the-art literature about NNs verification is that, at the best of our knowledge, most of the tools and methods focus on leveraging specific architectural properties of NNs [13] or of the input space [18] but none of them leverages the research on pruning and quantization recently done by the machine learning community. Consequently, our third research question is:

Q3*: How can we leverage methods from the machine learning community to enhance the verification methods of interest?*

A further limitation of most current tools is that they do not go beyond verification. Once a NN is found to violate some kind of property it would be useful to repair the model at hand, *i.e.*, to modify it until it complies to the desired properties. In the machine learning community this problem has been tackled with data augmentation techniques using, *e.g.*, generative adversarial networks [2]. We believe it could be interesting to investigate formal methods for NNs repair concerning the properties of choice. At the best of our knowledge, the only contributions in the literature to these last two methodologies are, respectively, [10] and [8], even if the first contribution is not about the verification of NNs but of Kernel Ridge Regression. This brings us to our fourth research question:

Q4*: How can we automatically repair NNs which do not respect the formal specifications of interest?*

We argue that these questions can be addressed partly by combining known formal methods and techniques and partly by providing new theoretical and experimental results. In particular, a comprehensive tool for formal verification applied to NNs is sought. Its expected capabilities are identified by our research questions.

3 Research Goals and Methodologies

The research goal of this proposal can be summarized as follows:

Design and implement a new comprehensive tool for the verification of Neural Networks: it is required to be learning framework agnostic and it needs to provide capabilities for the training, pruning, quantization, verification and repair of NNs models.

The importance of this goal has been argued in Section 2, but we find important to remark that, although in the last few years many different tools and methods for NNs verification have been developed, they present wildly different requirements for their use and in general they do not reach a level of scalability high enough to verify state-of-the-art models. In this work, we will try to tackle the above-mentioned problem and the research questions presented in Section 2, in particular we intend to follow this research plan:

- **Workpackage 1**: Investigation of the correct design and standards for our tool in order to be able to manage models generated using different learning frameworks and to provide the user with an easy to use interface for training, pruning, quantization and verification of generic NNs.
  Milestone 1: Complete design of our tool and its interfaces.

- **Workpackage 2**: Investigation of techniques to simplify state-of-the-art NNs preserving their accuracy and their robustness properties: this will involve testing different state-of-the-art procedures for quantization and pruning on different NNs models. Afterwards, it will be necessary to test the modified models with respect to different verification techniques in order to assess how their robustness has been changed by simplification techniques. We do not exclude the possibility of designing a verification oriented procedure for pruning and/or quantization.
  Milestone 2: Working pruning and quantization capabilities.

- **Workpackage 3**: Investigation of the state of the art concerning verification techniques and their enhancement. If necessary, the development of new more scalable ones.
  Milestone 3: Working verification capabilities.

- **Workpackage 4**: Combination of formal verification and machine learning techniques to develop novel methods for the repair of NNs: in particular the idea is to take a model which does not respect some kind of desired properties and to use an automated procedure to repair it, transforming it in a model respecting such properties.
  Milestone 4: First stable version of the tool with all its capabilities, repair included.

It will be also necessary to validate the capabilities of our tool: to do so we will need a set of standard benchmark, *i.e.,* a stable set of networks and related properties of interest. Currently the only standard benchmark for neural networks verification is the ACAS XU benchmark presented in [12]. However this benchmark consider only small (*i.e.*, with less then 1000 neurons) fully connected networks, therefore a new, closer to the current state of the art, set of benchmarks is needed. We propose to contribute to the community effort to establish these new benchmarks and a standard format for the sharing of neural networks and their properties of interests in the related VNN-LIB project [1].

4 Preliminary Results

The research program stated before has already been started and produced some preliminary results. In particular, we have investigated how to verify and repair machine-learned controllers (even if not neural networks based) using both convex optimization and retraining in [8] and [10]. We have then tried to extend the results obtained in [8] to neural networks: in particular we have considered two different convolutional neural networks trained on the datasets MNIST [15] and CIFAR10 [14] and we have tried to repair them using convex optimization techniques and transfer learning in order to make them more robust with respect

[1] http://www.vnnlib.org/

to adversarial examples computed using an off-the-shelf tool. The networks considered had their last few fully connected layers replaced with a linear support vector machine. The results of these works can be found in [7, 9].

We have investigated on a common format for the NNs models to manage them regardless of the learning framework and, in this regard, we have identified the ONNX format[2]. This format, developed and supported by many important industrial partners (e.g. AWS, IntelAI, AMD, NVIDIA etc.), supports most learning frameworks like PyTorch, Caffe, Microsoft Cognitive Toolkit and others. Moreover, it provides converters from and to other important frameworks like Tensorflow, Keras, Sci-kit Learns and others. Given its characteristic and the industrial support we believe that ONNX is a good choice for a common format for our tool.

We have investigated the state of the art of pruning and quantization techniques for NNs and we have pinpointed some methods we are interested in implementing in our tool: we have realized that this kind of methods usually manages NNs in various ways and therefore we have studied how to design a common interface for this kind of methods. The idea is to provide to the user of our tool with a portfolio of pruning and quantization methods which can be directly applied to their model without manually converting or modifying them.

In order to enhance the state-of-the-art verification techniques we have studied them and, as a first step, we are investigating whether they can benefit from pruning and quantization techniques for NNs: the application of this kind of methodologies before verification could be useful to indirectly enhance the scalability of the latter. We are investigating different techniques to develop our own verification procedure: in particular, we are studying how to leverage knowledge representation, layer-by-layer analysis and transfer learning to enhance the scalability of the procedure. We are interested in leveraging the characteristic of the input domain of the networks application to reduce the complexity of the verification problem in a similar way to what we have done in [10]. We are also investigating how methodologies traditionally used by the complex networks community can be applied to the verification of NNs: seeing NNs as a particular kind of complex networks it is possible to use methodologies like topological data analysis to understand their properties.

Acknowledgement

The research activities presented are carried out under the supervision of professor Armando Tacchella (University of Genoa)

[2] https://onnx.ai/

References

1. Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A., Arshad, H.: State-of-the-art in artificial neural network applications: A survey. Heliyon **4**(11) (2018)
2. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. arXiv:1711.04340 (2017)
3. Carlini, N., Wagner, D.: Audio adversarial examples: Targeted attacks on speech-to-text. In: IEEE SPW. pp. 1–7 (2018)
4. Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., Vechev, M.: Ai2: Safety and robustness certification of neural networks with abstract interpretation. In: IEEE SP. pp. 3–18 (2018)
5. Gilmer, J., Adams, R.P., Goodfellow, I., Andersen, D., Dahl, G.E.: Motivating the rules of the game for adversarial example research. arXiv:1807.06732 (2018)
6. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv:1412.6572 (2014)
7. Guidotti, D., Leofante, F.: Verification and repair of neural networks: a progress report on convolutional models. In: Cyber-Physical Systems PhD Workshop. pp. 18–28 (2019)
8. Guidotti, D., Leofante, F., Castellini, C., Tacchella, A.: Repairing learned controllers with convex optimization: A case study. In: CPAIOR. pp. 364–373 (2019)
9. Guidotti, D., Leofante, F., Pulina, L., Tacchella, A.: Verification and repair of neural networks: a progress report on convolutional models. In: AI*IA (to appear)
10. Guidotti, D., Leofante, F., Tacchella, A., Castellini, C.: Improving reliability of myocontrol using formal verification. IEEE Transactions on Neural Systems and Rehabilitation Engineering **27**(4), 564–571 (2019)
11. Huang, X., Kroening, D., Kwiatkowska, M., Ruan, W., Sun, Y., Thamo, E., Wu, M., Yi, X.: Safety and trustworthiness of deep neural networks: A survey. arXiv:1812.08342 (2018)
12. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient smt solver for verifying deep neural networks. In: CAV. pp. 97–117 (2017)
13. Katz, G., Huang, D.A., Ibeling, D., Julian, K., Lazarus, C., Lim, R., Shah, P., Thakoor, S., Wu, H., Zeljić, A., Dill, D.L., Kochenderfer, M.J., Barrett, C.: The marabou framework for verification and analysis of deep neural networks. In: Dillig, I., Tasiran, S. (eds.) CAV. pp. 443–452 (2019)
14. Krizhevsky, A., Nair, V., Hinton, G.: The cifar-10 dataset. online: http://www. cs. toronto. edu/kriz/cifar. html **55** (2014)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
16. Leofante, F., Narodytska, N., Pulina, L., Tacchella, A.: Automated verification of neural networks: Advances, challenges and perspectives. arXiv:1805.09938 (2018)
17. Tjeng, V., Tedrake, R.: Verifying neural networks with mixed integer programming. arXiv 1711.07356 (2017)
18. Wicker, M., Huang, X., Kwiatkowska, M.: Feature-guided black-box safety testing of deep neural networks. In: TACAS. pp. 408–426 (2018)