Verification and Repair of Neural Networks

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

Abstract

Neural Networks (NNs) are popular machine learning models which have found successful application in many different domains across computer science. However it is hard to provide any formal guarantee on the behaviour of neural networks and therefore their reliability is still in doubt, especially concerning their deployment in safety and security-critical applications. Verification emerged as a promising solution to address some of these problems. In the following I will present some of my recent efforts in verifying NNs.

Introduction

Formal verification aims to guarantee that neural networks satisfy stated input-output relations. Several verification methodologies have been proposed for different specifications and architectures (Katz et al. 2019; Balunovic et al. 2019; Tjeng, Xiao, and Tedrake 2019). Despite this impressive progress, verifying neural networks is computationally intensive and remains challenging for non-trivial architectures. This problem is one of the main focuses of my PhD research, as I will describe later in this document. While the main aim of verification is to prove that a neural network is compliant with a certain property or to provide a counter-example which shows that such property is not satisfied, my research focuses also on another problem which appears after verification. How to repair a network that has been proven to be faulty? A straightforward repair methodology is to re-train the network augmenting the dataset with counterexamples obtained by verification (Pulina and Tacchella 2010; Dreossi et al. 2018). Verification and re-training can be iterated until the network of interest is proved to behave correctly, however, since verification may be computationally expensive, this methodology can work only on small networks. Although augmentation-based re-training can improve the accuracy of the model on some points of interest, it must be noted that re-training results in a network different from the original one, thus possibly leaving way to new unwanted behaviours. More recently analytical approaches to repair that do not require re-training have been proposed (Goldberger et al. 2020; Papusha et al. 2020). These approaches use Satisfiability Modulo Theory (SMT) or Mixed Integer Linear Programming (MILP) to perform minimum modification to the weights of the networks in such a way that the property of interest is guaranteed to be satisfied. Repair is the second main focus of my PhD research. In the following, I will present some results obtained in these first two years and, after that, I will outline the activities I plan to accomplish to complete my PhD studies.

Current Results

During the first two years of my PhD, I investigated how to enhance the scalability of existing verification methodologies and how to repair NNs which do not satisfy certain constraints so that they become compliant with the input-output specifications of interest. I focused on leveraging machine learning techniques to reduce NNs so that they are easier to verify. In particular, I investigated how pruning could be used to produce networks whose robustness and performances are comparable with the original ones but which are faster to verify. Pruning consists in removing or blending components of a NN to reduce its complexity: its main application until now has been to reduce the dimension of NNs so that they can be deployed on hardware with low memory resources. In (Guidotti et al. 2020) I considered different pruning methodologies and complete verification techniques. Our results were consistent in showing that, while having comparable accuracy and robustness with respect to the original ones, pruned NNs were much easier to verify for the verification tools. Concerning verification methodologies, I also showed that pruning by removing whole neurons seems more effective than pruning by removing the parameters. The reason for this phenomenon is that removing whole neurons removes non-linearities from the NNs, therefore removing whole constraints, whereas removing the parameters only reduces the arity of such constraints. Regarding the repair of NNs, I investigated how to directly modify the parameters of a NN to make it more robust to a set of adversarial samples. Starting from techniques I developed during my master thesis (Guidotti et al. 2019c,a), in (Guidotti et al. 2019b) I considered convolutional NNs trained on the popular datasets MNIST and CIFAR10 and I considered the Fast Sign Gradient Method (FSGM) (Goodfellow, Shlens, and Szegedy 2015) for the generation of adversarial attacks. The idea was to solve a MILP problem with the parameters of the NNs as variables and with constraints defined to

guarantee the robustness of the resulting network for a set of adversarial example. However, the number of parameters of the convolutional and linear layers was too high to make the approach feasible in practice. Therefore, using ideas from transfer learning, I kept the convolutional layer as a feature extractor and substituted the linear layers with regression-based on support vectors. Then I attempted repair of the resulting hybrid network considering only the parameters of the regression. The resulting hybrid networks were robust to both the adversarial examples considered in the MILP problem and the ones generated before the repair but were not robust to the adversarial attack in general.

Planned Contribution and Future Research

In the following, I will outline three research direction I intend to pursue in the next year.

Formal Study of Pruning

I intend to deepen the analysis of the formal relation between pruned networks and original networks. Indeed in (Guidotti et al. 2020) I have shown that pruned network are more easily verified but I did not define a formal connection between them and the original networks, since I assumed that the pruned ones would be the "deployed" networks. However, I believe it would be significant to provide a formal connection between the two kinds of networks considering pruning as a particular form of abstraction to enable analysis of pruned networks to verify the original ones.

Scalability of Repair Approaches

Another research topic I wish to consider is how to enhance the scalability of repair methodologies for neural networks. As previously mentioned, these methodologies have scalability issues even for small networks and therefore their enhancement is essential to foster pratical application. At present, I am studying how abstract interpretation could be leveraged in this topic. In particular, I am studying how to use abstractions of the input and output domains, as well of the network to design a scalable repair procedure.

Definition of a Standard for NNs Verification

Another relevant issue for the verification of neural networks is the current absence of a standard for verification benchmarks, *i.e.*, networks and properties thereof. I am currently working on this topic with other members of the VNN-LIB [1] initiative, whose aim is to encourage collaboration and facilitate research and development in verification of neural networks. At present we have completed the first draft of a common standard for the sharing of benchmarks and we already provide some networks compliant with it.

NeVer 2.0

As the culmination of my PhD, I aim to implement all the methodologies resulting from my research in the tool *NeVer 2.0* (Guidotti, Pulina, and Tacchella 2020). The complete version of the tool will provide capabilities for the training, verification, pruning and repair of different kind of neural networks. It will also offer a graphical user interface for the costruction of some of the most popular network architectures in verification.

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

References

Balunovic, M.; Baader, M.; Singh, G.; Gehr, T.; and Vechev, M. T. 2019. Certifying Geometric Robustness of Neural Networks. In *Proc. of NeurIPS'19*, 15287–15297.

Dreossi, T.; Ghosh, S.; Yue, X.; Keutzer, K.; Sangiovanni-Vincentelli, A. L.; and Seshia, S. A. 2018. Counterexample-Guided Data Augmentation. In *Proc. of IJCAI'18*, 2071–2078.

Goldberger, B.; Katz, G.; Adi, Y.; and Keshet, J. 2020. Minimal Modifications of Deep Neural Networks using Verification. In *Proc. of LPAR'20*, 260–278.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In Bengio, Y.; and LeCun, Y., eds., *Proc. of ICLR'15*.

Guidotti, D.; Leofante, F.; Castellini, C.; and Tacchella, A. 2019a. Repairing Learned Controllers with Convex Optimization: A Case Study. In *Proc. of CPAIOR'19*, 364–373.

Guidotti, D.; Leofante, F.; Pulina, L.; and Tacchella, A. 2019b. Verification and Repair of Neural Networks: A Progress Report on Convolutional Models. In *Proc. of AI*IA'19*, 405–417.

Guidotti, D.; Leofante, F.; Pulina, L.; and Tacchella, A. 2020. Verification of Neural Networks: Enhancing Scalability Through Pruning. In *Proc. of ECAI'20*, volume 325, 2505–2512.

Guidotti, D.; Leofante, F.; Tacchella, A.; and Castellini, C. 2019c. Improving reliability of myocontrol using formal verification. *IEEE TNSRE* 27(4): 564–571.

Guidotti, D.; Pulina, L.; and Tacchella, A. 2020. NeVer 2.0: Learning, Verification and Repair of Deep Neural Networks. *arXiv preprint arXiv:2011.09933* .

Katz, G.; Huang, D. A.; Ibeling, D.; Julian, K.; Lazarus, C.; Lim, R.; Shah, P.; Thakoor, S.; Wu, H.; Zeljic, A.; Dill, D. L.; Kochenderfer, M. J.; and Barrett, C. W. 2019. The Marabou Framework for Verification and Analysis of Deep Neural Networks. In *Proc. of CAV'19*, 443–452.

Papusha, I.; Wu, R.; Brulé, J.; Kouskoulas, Y.; Genin, D.; and Schmidt, A. 2020. Incorrect by Construction: Fine Tuning Neural Networks for Guaranteed Performance on Finite Sets of Examples. *CoRR* abs/2008.01204.

Pulina, L.; and Tacchella, A. 2010. An Abstraction-Refinement Approach to Verification of Artificial Neural Networks. In *Proc. of CAV'10*, 243–257.

Tjeng, V.; Xiao, K. Y.; and Tedrake, R. 2019. Evaluating Robustness of Neural Networks with Mixed Integer Programming. In *Proc. of ICLR'19*.