Formal Verification by Abstract Interpretation of Functional Correctness Properties of Neural Networks

3-year PhD proposal, 2020

Caterina Urban Antique team, INRIA & DI, École normale supérieure
Antoine Miné APR team, LIP6, Sorbonne Université

Keywords: deep learning, neural networks, software verification, formal methods, static analysis, abstract interpretation, embedded systems, avionics.

Context

Machine learning techniques, notably *deep learning* [3], are enjoying a tremendous success. They are applied to an ever growing set of applications (image classification, speech recognition, prediction, to name a few). They are now also being considered for embedded critical applications, including medical devices, autonomous driving, or avionics. Such applications require a high level of insurance that the system behaves reliably and as expected.

In the context of traditional (non machine learning) software systems, including embedded systems, *formal methods* have been used in order to provide strong, mathematically-grounded proofs of correctness. They are used at an industrial level in avionics [7], which has very stringent verification requirements mandated by international standards (DO-178C). This success is due to the recognition of formal methods by standards and the availability of effective, efficient verification tools. Among formal verification techniques, *static analyzers* based on *abstract interpretation* [1] live in a sweet spot as they are fully automated, efficient, and sound by construction (such as the Astrée analyzer [2] used to ensure the absence of run-time errors in critical avionics C code). By contrast, the use of formal verification techniques in machine learning is extremely limited.

The goal of the PhD is to explore the use of abstract interpretation to verify neural networks, with a focus on avionic applications. The Antique team from INRIA & ENS and the APR team from Sorbonne Université are expert in verification by abstract interpretation and application to avionics [2]. The Antique team also has a growing expertise in the analysis of neural networks [5, 6]. We are collaborating with Airbus to explore the use of embedded neural networks in avionics, define their verification requirements, and obtain case studies.

Problem

The state of the art in the formal verification of neural networks is mostly limited to proving local robustness, which only ensures that a network behaves as expected around a small set of isolated points in the domain space. Several techniques have been applied to this problem (SMT, integer programming, etc.), and abstract interpretation has proven particularly effective [4] thanks to its ability to design abstraction with tunable cost vs. precision tradeoffs. However, global robustness (robustness at all points), which would be more suitable to critical software, seems out of their reach. Another property, fairness, which is of interest for society-related decision making applications, has also been tackled using abstract interpretation [6]. These are all non-functional properties, which are implicitly specified.

Our aim with this PhD is to explore *functional properties* instead. Existing work in this area [8] is even more limited, focusing on linear programming methods, with a single use case and simple, artificial properties. By contrast, we wish to use abstract interpretation to tackle complex, realistic properties, identified through our collaboration with Airbus.

The PhD will thus address two key challenges currently limiting a broader application of formal verification to data science software [5]: the lack of specifications, and the difficulty to achieve both scalability and precision on a wider class of neural networks. We have already identified two kinds of specifications of interest:

- designer-specified input-output assertions stating high-level properties (such as “when input X is between these bounds, then output Y must be greater than Z”);
- for networks approximating (for performance reasons) a function defined analytically or algorithmically, a bound between the exact and approximate result on all inputs.

We believe that more classes will emerge in the course of the PhD project. Concerning scalability and precision, we are confident that the theory of abstract interpretation provides a large enough design space to create abstractions adapted to each class of networks and specifications (as is the case in software verification).

Expected Work

The expected work will consist in several steps from the choice of case studies to designing, implementing and validating experimentally static analyses:

1. The student will select, with the help of our industrial contacts at Airbus, representative examples of neural networks (starting with smaller ones and growing in size and complexity) and properties of interest.
2. Formal specifications for these classes of properties will need to be defined, possibly in a parametric way to enable configuration by the analysis user.
3. The student will define new abstract domains, which are at the core of all abstract interpreters [1]. Both semantic aspects (expressiveness, abstract algebra) and algorithmic aspects (data-structures, algorithms, complexity) must be considered to allow a theoretical analysis and an effective implementation. Many abstract domains (including numeric domains, which are natural candidates to tackle learning methods) have already been proposed, but mainly for software analysis. They will need at least some adaptation, or a complete redesign, to become effective on neural network semantics and properties. Moreover, each different class of networks and properties may require domain-specific abstractions.
4. The abstractions shall be proved formally sound using the abstract interpretation theory.
5. The student will implement the methods and experiment them on the case studies. Because abstract domains embed a precision vs. cost tradeoff, experimental proof is paramount to justify the ability of the method to provide meaningful results in reasonable times or realistic networks. Experiments will provide feedback to hone abstract domains and achieve scalability as larger networks are considered. We expect the student to iterate on steps 3 to 5 in a tight loop.

Notes: The PhD will be take place at both École normale supérieure and Sorbonne Université. We expect frequent meetings with our industrial contacts at Airbus in Toulouse.

Requested Skills: strong theoretical and practical knowledge of formal methods, in particular abstract interpretation; knowledge of deep neural networks; programming skills and the willingness to implement and conduct analysis experiments; ability to read, write, and present in English.

Contacts: Caterina Urban (`caterina.urban@inria.fr`), Antoine Miné (`antoine.mine@lip6.fr`).

References

[1] P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In POPL'77, pages 238-252, 1977.

[2] J. Bertrane, P. Cousot, R. Cousot, J. Feret, L. Mauborgne, A. Miné, and X. Rival. Static analysis and verification of aerospace software by abstract interpretation. In AIAA Infotech@Aerospace, number 2010-3385 in AIAA, pages 1–38. 2010.

[3] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner et al. Gradient-based learning applied to document recognition. In proceedings of the IEEE, 86(11):2278–2324, 1998.

[4] G. Singh, T. Gehr, M. Püschel, M. Vechev. An abstract domain for certifying neural networks. In proceedings of the ACM on Programming Languages, 3(POPL): 41, 2019.

[5] C. Urban. Static analysis of data science software. In proceedings of the 26th Static Analysis Symposium (SAS'19). 2019.

[6] C. Urban, M. Christakis, V, Wüstholz, and F. Zhang. Perfectly parallel fairness certification of neural networks. CoRR abs/1912.02499, 2019.

[7] J. Souyris, V. Wiels, D. Delmas, and H. Delseny. Formal verification of avionics software products. In Ana Cavalcanti and Dennis Dams, editors, FM, volume 5850 of Lecture Notes in Computer Science, pages 532-546. Springer, 2009.

[8] G. Katz, C. Barrett, D. L. Dill, K. Julian, M. J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In Computer Aided Verification, 97–117, 2017, Springer.