Exploration of properties of differentiable logics through mechanisation

Reynald Affeldt, Alessandro Bruni, Ekaterina Komendantskaya, Kathrin Stark, and Natalia Ślusarz

Verification of Neural Networks consists of two parts: 1. verification of a given property *per se*, and 2. training of the neural network, that optimises neural network's parameters towards satisfying the given logical property; cf. e.g. [1]. The latter is usually done via modifying the neural network's loss function. A group of methods that allow to generate loss functions from an arbitrary logical property is known under an umberella term of *Differentiable Logics (DLs)*. For example, well-studied *fuzzy logics* that date back to the works of Łukasiewicz and Gödel can be used as DLs [6]. Recently, both verification and machine-learning communities formulated alternative DLs such as DL2 [7] and STL [4] claimed to be more performant in optimisation tasks.

Our interest in DLs lies in making their embedding into the overall verification cycle more smooth and generic [2]. However, our initial attempt in [3] to express the existing DLs in the same generalised language met some obstacles. Firstly, soundness of some DLs, such as STL [4], remained an open problem and was hard to resolve manually. Pen and paper proofs of some other conjectured soundness results had many tricky cases that were prone to errors. Secondly, Varnai et al. [2] strongly argued for characterisation of DLs in terms of their *geometric* properties that are valuable for optimisation tasks: smoothness, scale-invariance, shadow-lifting. Shadow-lifting was the most original and important of the three, and encapsulated the idea of gradual improvement in property-driven training. However, their published proof sketches were not sufficiently detailed, and—even more importantly—did not immediately generalise to other DLs. Both of these groups of challenges called for rigorous computer formalisation of DLs, together with proofs of their major logical and geometric properties, in an interactive theorem prover based on dependent-type theory.

In this talk, we will present the resulting formalisation in Coq. By exploiting higher-order syntax and dependent types, we formalised and proved properties of all DLs of interest in a generic way. In particular, the *Mathematical Components library*, especially its modules *algebra* and *analysis*, enabled full proofs of geometric properties. This new level of rigour and modularity allowed us to positively solve the open problem of STL soundness; verify other soundness results; and for the first time give proofs of shadow-lifting for all DLs for which they hold. Proofs of other geometric properties are left for future work.

References

[1] Casadio, M., Komendantskaya, E., Daggitt, M., Kokke, W., Katz, G., Amir, G., Refaeli, I.: Neural Network Robustness as a Verification Property: A Principled Case Study. Int. Conf. on Computer Aided Verification: CAV (1) 2022: 219-231

[2] Daggitt, M., Kokke, W., Atkey, R., Slusarz, N., Arnaboldi, L., Komendantskaya, E.: Vehicle: Bridging the Embedding Gap in the Verification of Neuro-Symbolic Programs. arXiv https://arxiv.org/abs/2401.06379

[3] Ślusarz, N., Komendantskaya, E., Daggitt, M. L., Stewart, R., and Stark, K. (2023). Logic of Differentiable Logics: Towards a Uniform Semantics of DL. In Proceedings of 24th International Conference on Logic (Vol. 94, pp. 473-493).

[4] Varnai, P., and Dimarogonas, D. V. (2020, July). On robustness metrics for learning STL tasks. In 2020 American Control Conference (ACC) (pp. 5394-5399). IEEE.

[5] Mathematical Components Team. Mathematical components library, 2007. URL https://github.com/math-comp/math-comp. Last stable version: 2.2 (2024).

[6] van Krieken, E., Acar, E., and van Harmelen, F. (2022). Analyzing differentiable fuzzy logic operators. Artificial Intelligence, 302, 103602.

[7] Fischer, M., Balunovic, M., Drachsler-Cohen, D., Gehr, T., Zhang, C., and Vechev, M. (2019, May). DL2: training and querying neural networks with logic. In International Conference on Machine Learning (pp. 1931-1941). PMLR.