Composing Monotonic Neural Networks to Train and Verify Complex Networks

Background

Neural networks are the building blocks of deep learning systems. They are useful for solving numerous complex tasks like pattern recognition [3], natural language processing [5] and medical diagnosis [2]. In recent years, several papers have outlined the importance of formal verification of neural networks in systems like autonomous driving, fraud detection [4] and visual recognition [9]. For safety reasons in real world applications, when models are used to inform decisions that affect human actors, robustness to small perturbations is a desired property [7].

Monotonic neural networks are especially important in achieving highly accurate, transparent and interpretable models [11]. These monotonic neural networks make use of linear functions and are mostly used for smaller datasets. Generally, monotonic models can be split into two major categories: built-in and constrained monotonic architectures [7] where the monotonicity is guaranteed, and regularized architectures where monotonicity is enforced during training [11]. Complex networks on the other hand rely on non-linear functions, are more flexible and are not limited by monotone constraints. As the name suggests, these networks are suitable for complex tasks like audio or image pattern recognition [8, 6]. While smaller neural networks have already been successfully verified, complex networks still lack certification.

The simple and linear properties of monotonic networks make them more predictable, and easier to verify than the complex, unconstrained neural networks. For monotonic networks, efficient mathematical tools like SMT solvers can be used for verification [10], whereas the unpredictable nonlinearity of complex networks is much harder to analyze. This research project combines the properties of monotonic neural networks and non-linear networks to make the analysis of complex networks less abstract.

Description

The aim of this thesis is to extend the CORA neural network verification procedure [1] by combining various monotonic neural networks in order to create big and complex networks to study, train and verify non-monotone functions. We can exploit the partial monotonicity characteristic of these networks to determine exact output constraints for the formal verification while keeping them interpretable. These big networks will be trained and verified with CORA, namely with open-loop and closed-loop verification. We want to evaluate these big networks and compare their efficiency to neural networks without partial monotonicity.

Tasks

- Familiarize with the toolbox CORA [1].
- Literature research on monotonic neural networks and compositional neural network architectures.
- Combine monotonic neural networks to analyze non-monotonic functions.
- Develop efficient algorithms for the verification which exploit the monotonicity.
- Evaluate efficiency of combined neural networks for training and verification.
- How many monotonic neural networks do we have to combine for good results?
- How big can each monotonic neural network be?

References

[1] Matthias Althoff. An introduction to cora 2015. In *ARCH@ CPSWeek*, pages 120–151, 2015.

Technical University of Munich

Department of Informatics

Chair of Robotics, Artificial Intelligence and Real-time Systems

Supervisor:
Prof. Dr.-Ing. Matthias Althoff

Advisor:
Lukas Koller, M.Sc.

Research project:
DFG–SPP2422

Type:
BT

Research area:
Formal verification, neural networks

Programming language:
MATLAB

Required skills:
Knowledge in formal methods and machine learning, good mathematical background

Language:
English

Date of submission:
28. Juli 2025

For more information please contact us:

Phone: +49 (89) 289 - 18140

E-Mail: lukas.koller@tum.de

Website: ce.cit.tum.de/cps/

[2] Filippo Amato, Alberto López, Eladia María Peña-Méndez, Petr Vaňhara, Aleš Hampl, and Josef Havel. Artificial neural networks in medical diagnosis, 2013.

[3] Jayanta Kumar Basu, Debnath Bhattacharyya, and Tai-hoon Kim. Use of artificial neural network in pattern recognition. *International journal of software engineering and its applications*, 4(2), 2010.

[4] Kang Fu, Dawei Cheng, Yi Tu, and Liqing Zhang. Credit card fraud detection using convolutional neural networks. In *International conference on neural information processing*, pages 483–490. Springer, 2016.

[5] Yoav Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420, 2016.

[6] Samer Hijazi, Rishi Kumar, Chris Rowen, et al. Using convolutional neural networks for image recognition. *Cadence Design Systems Inc.: San Jose, CA, USA*, 9(1), 2015.

[7] Ouail Kitouni, Niklas Nolte, and Michael Williams. Expressive monotonic neural networks. *arXiv preprint arXiv:2307.07512*, 2023.

[8] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

[9] Claus Nebauer. Evaluation of convolutional neural networks for visual recognition. *IEEE transactions on neural networks*, 9(4):685–696, 1998.

[10] Luca Pulina and Armando Tacchella. Checking safety of neural networks with smt solvers: a comparative evaluation. In *Congress of the Italian Association for Artificial Intelligence*, pages 127–138. Springer, 2011.

[11] Davor Runje and Sharath M Shankaranarayana. Constrained monotonic neural networks. In *International Conference on Machine Learning*, pages 29338–29353. PMLR, 2023.

Technical University of Munich

Department of Informatics

Chair of Robotics, Artificial Intelligence and Real-time Systems