Formal Verification of Robust Neural Networks using Reachability Analysis

Technical University of Munich

Department of Informatics

Chair of Robotics, Artificial Intelligence and Real-time Systems

Background

Neural networks are powerful machine learning models that can achieve state-of-the-art results, including for safety-critical tasks [1]. However, they are also vulnerable to adversarial examples [2], which are slightly perturbed inputs that cause the neural network to misclassify them. Adversarial examples pose a severe challenge to the security and reliability of neural network applications, especially in safety-critical domains. Therefore, it is essential to study the robustness of neural networks [3], which measures how well they can resist adversarial perturbations.

To improve the robustness of neural networks, various methods have been proposed in the literature. One category of methods is based on adversarial training, which involves generating adversarial examples [2] during the training process. Adversarial training aims to make the neural network more invariant to worst-case perturbations within a given distance metric. For example, ϵ-robustness can be improved by training with projected gradient descent attacks [2]. Another category of methods is based on modifying the network architecture [4] or regularization [5] to enhance the robustness.

Description

The main focus of this work is the formal verification of neural networks trained by these robustness-improving training methods. While the authors of several training methods have shown to empirically improve the robustness of neural networks, they do not give formal guarantees. Formal guarantees can be provided by determining the ϵ-safe radius of an input x_0, i.e. all inputs x' with a perturbation of at most ϵ from x_0 are still classified correctly by the neural network. We use reachability analysis [6, 7] to determine if all perturbed inputs are classified correctly by modeling this problem using sets.

The ϵ-safe radius can be bounded via a binary search algorithm: By applying reachability analysis, we can check whether the output set contains more than one class label. If it does or we find a counterexample, then the ϵ-safe radius is smaller than our current estimate. Otherwise, the ϵ-safe radius is larger than or equal to our current estimate. As the computed ϵ-safe radius only holds a specific input x_0, we envision providing a safety statement over an entire dataset, e.g. by averaging the results. This approach should be thoroughly evaluated on different datasets and robustness-improving training methods.

Tasks

- Literature research on the robustness of neural networks
- Implementation and training of neural networks with robustness properties
- Familiarize with the verification toolbox CORA [8]
- Implementation of an algorithm to determine a lower bound on the ϵ-safe radius using reachability analysis
- Extensive evaluation of the robustness of neural networks

Supervisor:
Prof. Dr.-Ing. Matthias Althoff

Advisor:
Tobias Ladner, M.Sc.

Research project:
-

Type:
Bachelor's Thesis

Research area:
Neural networks, robustness, formal verification

Programming language:
MATLAB, Python

Required skills:
Good programming skills, knowledge of neural networks and formal verification

Language:
English

Date of submission:
May 26, 2023

For more information please contact us:

Phone: +49 (89) 289 - 18140

E-Mail: tobias.ladner@tum.de

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

References

[1] Debasmita Mukherjee, Kashish Gupta, Li Hsin Chang, and Homayoun Najjaran. A survey of robot learning strategies for human-robot collaboration in industrial settings. *Robotics and Computer-Integrated Manufacturing*, 2022.

[2] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017.

[4] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[5] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[6] Niklas Kochdumper, Christian Schilling, Matthias Althoff, and Stanley Bak. Open- and closed-loop neural network verification using polynomial zonotopes. *arXiv preprint arXiv:2207.02715*, 2022.

[7] Tobias Ladner and Matthias Althoff. Automatic abstraction refinement in neural betwork verification using sensitivity analysis. In *Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control*, pages 1–13, 2023.

[8] 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