Formal Verification of Neural Network – Topics

Lukas Koller, Tobias Ladner

Prof. Dr.-Ing. Matthias Althoff
Cyber-Physical Systems Group
Technische Universität München

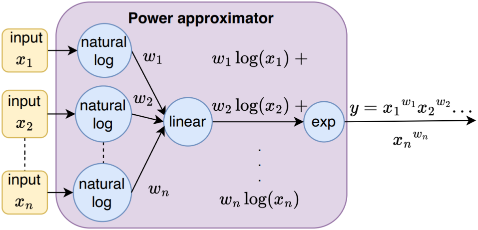
April 19th, 2024

Topic 1 – Learning Laurent Polynomials[1]

Neural Network Architecture

Approach:

- Laurent polynomials can describe many formulas from physics.
- Use a special neural network architecture to learn Laurent polynomials.



Power-term approximator component.

Task: Implement *power-term approximator* and train networks to learn Laurent polynomials.
Research Question: Can we use the learned Laurent polynomials for verification?

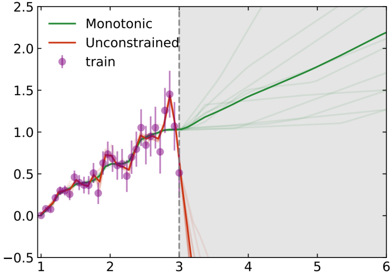
[1] Ranasinghe et al., "GINN-LP: A Growing Interpretable Neural Network for Discovering Multivariate Laurent Polynomial Equations".

Topic 2 – Monotonic Networks[2]

Neural Network Architecture

Approach:

- Use weight constraints and a residual connection to learn monotonic input-output relationships.
- For monotonic networks it suffices to only propagate the input bounds to compute tight output bounds.



Monotonic networks learn monotonic input-output relationships.

Task: Implement and train *monotonic networks*.
Research Question: How expressive are monotonic networks? Can we easily verify monotonic networks?

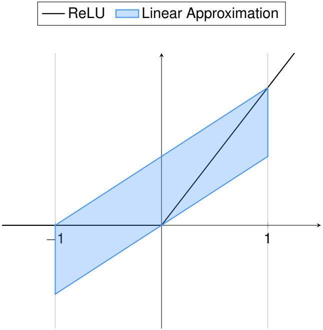
[2] Kitouni et al., “Robust and provably monotonic networks”.

Topic 3 – Complete Verification of ReLU Networks[3]

Neural Network Verification

Approach:

- Neural networks with ReLU activations are piecewise linear.
- By recursively splitting the input the exact output set can be computed.



Linear approximation of ReLU.

Task: Implement complete verification for ReLU networks by recursively splitting the input.

Research Question: What are effective splitting heuristics, e.g. sensitivity of a network?

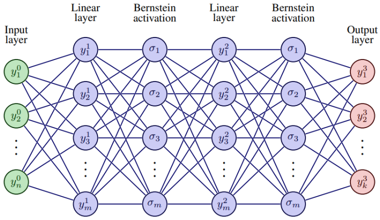
[3] Bunel et al., "Branch and bound for piecewise linear neural network verification".

Topic 4 – Bernstein Networks[4]

Neural Network Architecture

Approach:

- Use Bernstein polynomials to approximate activation functions.
- Bernstein polynomials are a special type of polynomials with unique properties, e.g. easy to bound.



Architecture of Bernstein networks.

Task: Implement and train *Bernstein networks*.

Research Question: How well can we verify Bernstein networks with polynomial Zonotopes?

[4] Fatnassi et al., "Bern-nn: Tight bound propagation for neural networks using bernstein polynomial interval arithmetic".

References

Bunel, Rudy, Ilker Turkaslan, Philip H. S. Torr, M. Pawan Kumar, Jingyue Lu, and Pushmeet Kohli. "Branch and bound for piecewise linear neural network verification". In: 21.1 (Jan. 2020).

Fatnassi, Wael, Haitham Khedr, Valen Yamamoto, and Yasser Shoukry. "Bern-nn: Tight bound propagation for neural networks using bernstein polynomial interval arithmetic". In: *Proc. of the Int. Conf. on Hybrid Systems: Computation and Control (HSCC)*. 2023, pp. 1–11.

Kitouni, Ouail, Niklas Nolte, and Mike Williams. "Robust and provably monotonic networks". In: *Machine Learning: Science and Technology* 4.3 (Aug. 2023).

Ranasinghe, Nisal, Damith Senanayake, Sachith Seneviratne, Malin Premaratne, and Saman Halgamuge. "GINN-LP: A Growing Interpretable Neural Network for Discovering Multivariate Laurent Polynomial Equations". In: *Proc. of the AAAI Conf. on Artificial Intelligence (AAAI)*. 2024.