Karlsruher Institut für Technologie

Application-oriented Formal Verification Research Group
Institute of Information Security and Dependability (KASTEL)
Am Fasanengarten 5
76131 Karlsruhe
https://formal.kastel.kit.edu

Proof Certificates for Neural Network Retraining

Topic for ”Praxis der Forschung”

Philipp Kern, Samuel Teuber

Background I: Neural Network Verification

- **Question:** Given NN g and property ψ, are there x, y s.t. $(g(x) = y) \wedge \psi(y)$ holds?
- Verification is **NP-complete** even for piece-wise linear feed-forward NNs
- Many current approaches are based on overapproximation as **Linear Programs** (LPs) and branch-and-bound.

Background II: Proof Certificates

- Solvers are complex software and might have bugs. With proof certificates their results can be checked independently.
- SAT instances: For input x easy to check if $g(x)$ satisfies ψ.
- UNSAT instances: For conjunction of linear constraints we can use **Farkas' Lemma**.

Idea: Proof Certificates after NN Retraining

Given a certificate $\mathcal{C}$ showing that property ψ is UNSAT for NN g. When new data is available, g may need to be retrained.

- $\mathcal{C}$ might not be a valid certificate for the retrained NN g'.
  Proof repair: Change $\mathcal{C}$, s.t. we obtain valid certificate for g'?
  Partial proofs: Reuse analysis of ”broken” NN?
- **Regularization**: Can we use $\mathcal{C}$ to regularize the retraining, s.t. $\mathcal{C}$ is still valid for g'?

Relevant Literature:

- NN verification [3, 1]
- Proof certificates for NN Verification [2]

Example: Farkas' Lemma

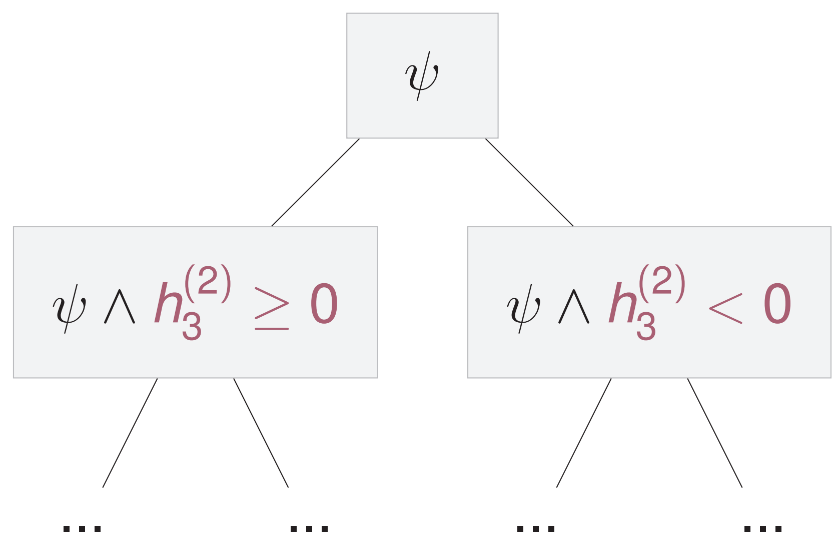
How to prove unsatisfiability of the following set of linear constraints?

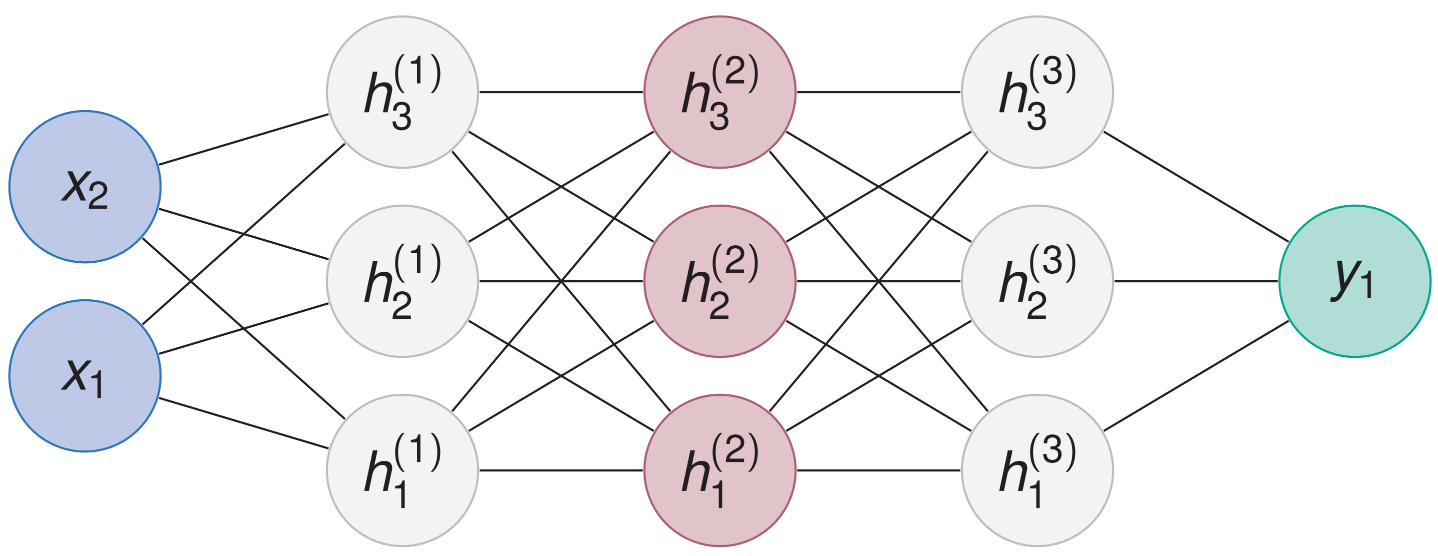
$$2x_1 + 3x_2 - 4x_3 = 5 \quad (1)$$
$$-x_1 - 2x_2 + 5x_3 = -6 \quad (2)$$
$$x_1, x_2, x_3 \geq 0 \quad (3)$$

For $\lambda_1 = \lambda_2 = 1$, we obtain

$$\begin{aligned} &\lambda_1 \cdot (2x_1 + 3x_2 - 4x_3) \\ &+ \lambda_2 \cdot (-x_1 - 2x_2 + 5x_3) \\ =&\lambda_1 \cdot 5 + \lambda_2 \cdot (-6) \\ \Leftrightarrow& x_1 + x_2 + x_3 = -1 \end{aligned}$$

which is clearly unsatisfiable for $x_1, x_2, x_3 \geq 0$.
So $\vec{\lambda} = (1, 1)^T$ is a certificate for the unsatisfiability of these constraints.





[1] Rüdiger Ehlers. “Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks”. In: *Automated Technology for Verification and Analysis*. 2017.

[2] Omri Isac et al. “Neural Network Verification with Proof Production”. In: *2022 Formal Methods in Computer-Aided Design (FMCAD)* (2022), pp. 38–48.

[3] Guy Katz et al. “Reluplex: a calculus for reasoning about deep neural networks”. In: *Formal Methods in System Design* (2021), pp. 1–30.