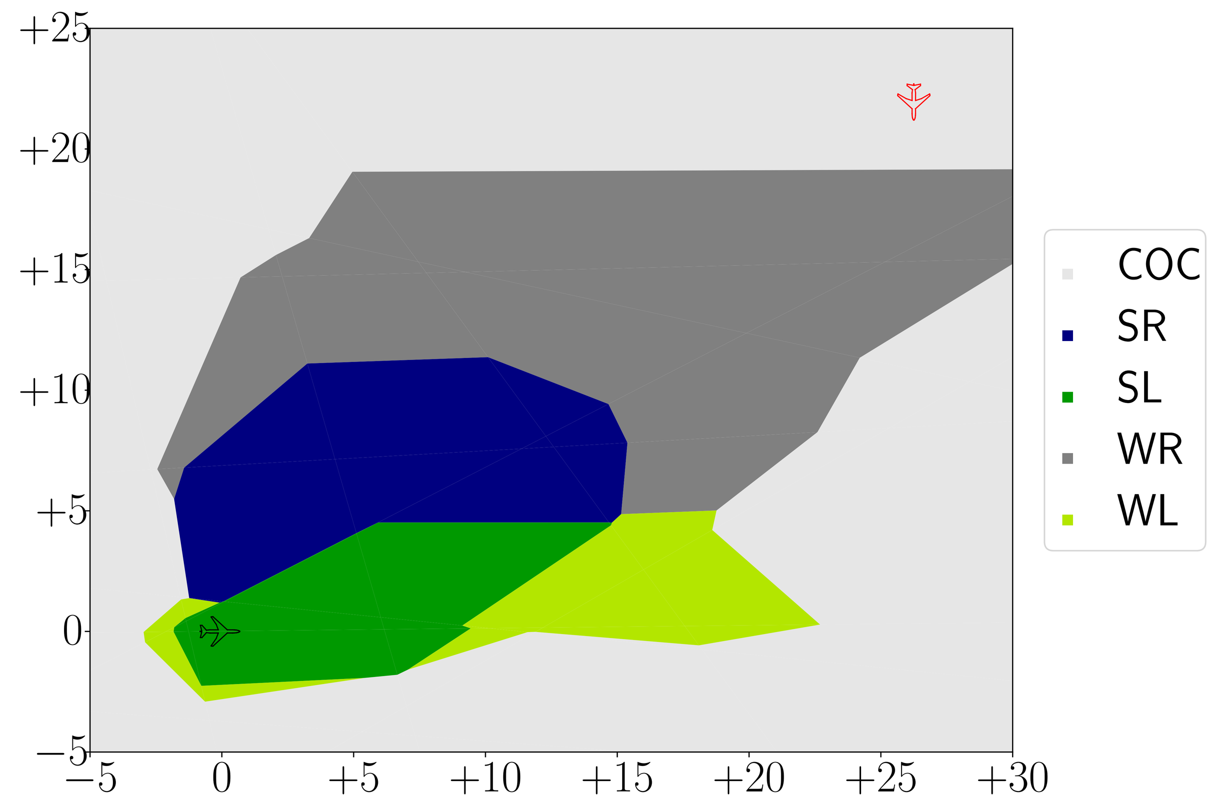
Exact Preimages of Neural Network Aircraft Collision Avoidance Systems

Kyle Matoba[1] and François Fleuret[2]

[1]Idiap and EPFL
[2]University of Geneva

Aircraft Collision Avoidance Systems

ACAS: Navigational aids that use data on positions and velocities to issue guidance on evasive actions to prevent collisions with an intruding aircraft.



- Due to hardware constraints, recent interest in using deep neural networks (DNNs) to compress policies.
- Idea: fit a small DNN that accurately models a given dataset of state (positions, velocities, etc.)-action (evasive action).
- Fundamental problem: very similar inputs can give very different outputs (Szegedy et al. [4]).
- Verification (e.g. Katz et al. [3] and Wong and Kolter [5]) can preclude foreseen property violations.
- Failure modes are hard to anticipate, we would prefer to have a representation that can be easily reasoned about – for example plotted so that a human expert can identify suspect behavior.

Dynamic reachability

Want to answer the question: If we follow an ACAS over time, can a (near) collision occur?

- Fix a model of randomness in state transitions, and worst-case behavior of other aircraft.
- Starting from the boundaries of the domain, ask whether states where two planes are too close can occur under the ACAS policy.
- Julian and Kochenderfer [2] propose: iteratively apply standard verification methods to cubes in the input space known to be reachable, and append any cubes that can be reached.
- Essential problem: volume of discretized decision boundary can be large. $\implies$ Inability to verify truly correct properties.
- If we knew $f^{-1}(\{x : x_i \geq x_j, i \neq j\})$ – the set of all inputs that would be classified as action i – then we could instead iteratively directly apply the transitions to the preimage.
- Exact (no overestimation) and plausibly more efficient.

DNN preimages

How do we compute f^{-1}?

- Write the inference pass of a DNN as $f = f_L \circ f_{L-1} \circ \ldots f_0$, for linear and ReLU layers f_ℓ.
  - Many other "layers" that are off at inference time (e.g. dropout) are absent from the representation.
  - Linear and ReLU can synthesize many common functions: average pooling, maxpooling, convolution, etc.
  - Similar arguments for other piecewise linear layers (e.g. residual blocks).
- The preimage of the composition of functions is the composition of preimages:

$$(f_L \circ f_{L-1} \circ \ldots \circ f_0)^{-1} = f_0^{-1} \circ \ldots \circ f_{L-1}^{-1} \circ f_L^{-1}.$$

- Without loss of generality, assume that f_ℓ operates on flattened tensors. Layer preimage of polytopes:
  - Linear

$$(x \mapsto Wx + a)^{-1}(\{x : b - Ax \geq 0\}) = \{x : (b - Aa) - AWx \geq 0\}.$$

  - ReLU

$$\begin{aligned}&\text{ReLU}^{-1}(\{x : b - Ax \geq 0\}) \\ &= \bigcup_{\nu \in \{0,1\}^n} \{x : b - A\text{diag}(\nu)x \geq 0, -\text{diag}(1-\nu)x \geq 0, \text{diag}(\nu)x \geq 0\}.\end{aligned}$$

- Preimage of union is union of preimages

$$f^{-1}\left(\cup_{i=1}^N S_i\right) = \cup_{i=1}^N f^{-1}(S_i).$$

- So: start at the last layer – $f_L^{-1}(\{x : x_i \geq x_j, i \neq j\})$ is a union of polytopes – and work progressively backwards, at each step computing the preimage of polytopes.
- Like verification, infeasible at scale.

Previous Work

Commonly called	What is computed	Examples
Verification	$(f, X, Y) \mapsto \mathbf{1}_{f^{-1}(Y) \cap X = \emptyset} (= \mathbf{1}_{f(X) \cap Y = \emptyset})$	Wong and Kolter [5]
Reachability	$(f, X) \mapsto f(X)$	Yang et al. [6]
Inversion	$(f, y) \mapsto f^{-1}(\{y\})$	Carlsson et al. [1]
Preimage	$(f, Y) \mapsto f^{-1}(Y)$	Our work

References

[1] Stefan Carlsson, Hossein Azizpour, and Ali Sharif Razavian. "The Preimage of Rectifier Network Activities". In: 2017.

[2] Kyle D. Julian and Mykel J. Kochenderfer. "Guaranteeing safety for neural network-based aircraft collision avoidance systems". In: *IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)* (2019).

[3] Guy Katz et al. "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks". In: *Computer Aided Verification*. Ed. by Rupak Majumdar and Viktor Kunčak. Springer International Publishing, 2017.

[4] Christian Szegedy et al. "Intriguing properties of neural networks". In: (2014). 2nd International Conference on Learning Representations, ICLR 2014 ; Conference date: 14-04-2014 Through 16-04-2014.

[5] Eric Wong and Zico Kolter. "Provable defenses against adversarial examples via the convex outer adversarial polytope". In: *arXiv e-prints* (2017).

[6] Xiaodong Yang et al. "Reachability Analysis for Feed-Forward Neural Networks using Face Lattices". In: *arXiv e-prints*, arXiv:2003.01226 (2020), arXiv:2003.01226.

Acknowledgements

Kyle Matoba was supported by the Swiss National Science Foundation under grant number FNS-188758 "CORTI".