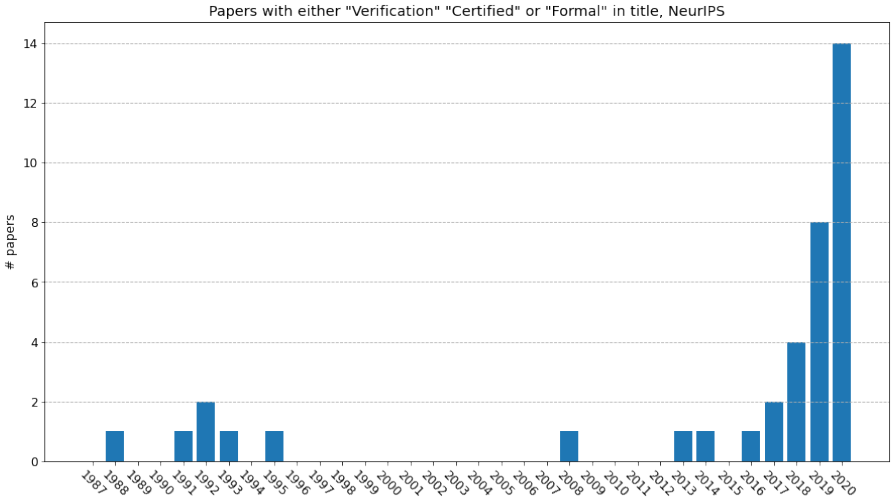
CAISAR: A Platform for Characterizing Artificial Intelligence Safety and Robustness

AISafety 2022

Julien Girard-Satabin (CEA LIST): julien.girard2@cea.fr

Formal verification for machine learning



Adapted from `https://github.com/nemanja-rakicevic/conference_historical_data_analysis`

Non-exhaustive list of tools

Include only the latest "version" (including extensions and rebranding)

- Marabou [Kat+19]
- Neurify
- ERAN [Sin+19; Mül+21]
- $\alpha - \beta$-Crown [Wan+21]
- Nnenum [Bak21]
- NNV (https://github.com/verivital/nnv)
- FaceLattice (https://arxiv.org/abs/2003.01226, https://github.com/verivital/FaceLattice)
- Facet-Vertex incidence (https://github.com/Shaddadi/Facet-Vertex-FFNN)
- Veritex (https://github.com/Shaddadi/veritex)
- Verinet and Venus (https://github.com/vas-group-imperial/VeriNet)

Non-exhaustive list of tools

Include only the latest ”version” (including extensions and rebranding)

- ReluDiff (https://arxiv.org/abs/2001.03662, https://github.com/pauls658/ReluDiff-ICSE2020-Artifact)
- Peregrinn (https://arxiv.org/abs/2006.10864, https://github.com/rcpsl/PeregriNN)
- Oval (https://github.com/oval-group/oval-bab)
- Libra [Urb+19]
- MIPVerify [TXT19]
- Planet [Ehl17]
- Sherlock [Dut+17]
- ZoPE (https://arxiv.org/abs/2106.05325,https://github.com/sisl/NeuralPriorityOptimizer.jl)
- DNNV [SED21]

(credits: Bill Wurtz)

Lots of tools increases burden of choice

Which tool to choose?

How to encode a given problem for multiple tools?

Could we specify a verification problem independently of the tool?

A new, thriving ecosystem

Selective pressure

Short lifetime of tools: Reluplex to Marabou, AI^2 to ERAN, Fast-Lin to $\alpha - \beta$-CROWN

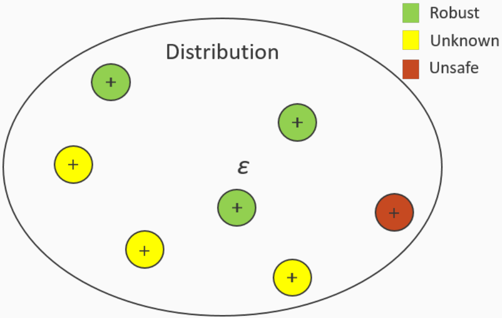
Ecological niches: from fully-connected neural networks to state-of-the-art architectures, by way of SVMs

Collaboration

Collaborative initiatives: VNN-COMP or VNNLib

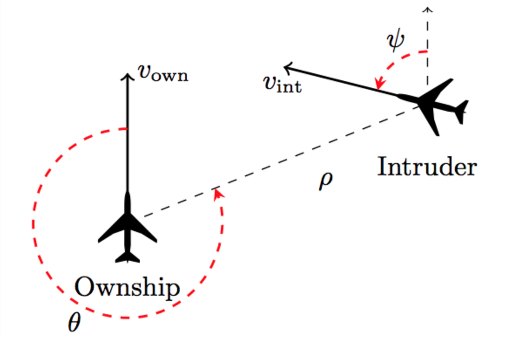
Cross-fertilization of techniques across tools: symbolic propagation [Li+19; Sun+18; HL20; Wan+21], efficient space parititionning [Urb+19; Gir+21], mixing exact solvers and fast bound propagation [Gil+18; Fer+22]

Families of properties according to the literature



local robustness: given x_0

$\forall x, dist(x - x_0) < \varepsilon \implies f(x) = f(x_0)$



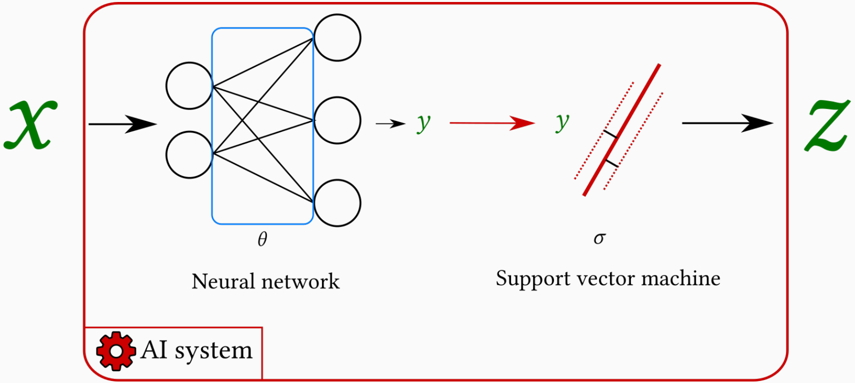
clearly defined semantics *à la* ACAS

global properties on low-dimensional programs

Existing benchmarks: ACAS-like and local adversarial robustness?

What about characterizing privacy? Fairness? Symmetry relations? How to phrase custom properties for provers that are not designed to?

Handling complex systems?



How to compose system components in the analysis?

Three interesting venues

1. tool-independant modelling
2. flexibility in problem statement
3. composition of components

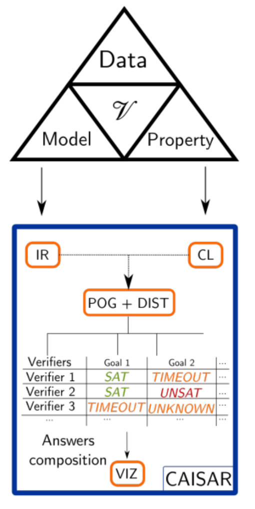
CAISAR

platform for Characterizing Artificial Intelligence Safety And Robustness

A platform building from principled and industrial-tested techniques

Written in OCaml, using Why3 as backend

Overall architecture of CAISAR



Supports SMT and abstract interpretation reasoning (Marabou, PyRAT, SAVer), and soon metamorphic testing (AIMOS)

WhyML language: composition

free component composition

tool independent modelling

```
theory T
 use Net.NNasTuple
 use SVM.SVMAsArray
 use ieee_float.Float64
 use caisar.NN

 goal G: forall x1 x2 x3.
  (0.0:t) .< x1 .< (0.5:t) ->
  let (y1,y2) = Net.net_apply x1 x2 x3 in
  let (z1,z2) = SVM.svm_apply y1 y2 in
    (0.0:t) .< z1 .< (0.5:t)
    /\
    (0.0:t) .< z2 .< (0.5:t)
end
```

WhyML language: expressivity

first-order language with polymorphic types, pattern matching, and inductive properties capabilities
modelization freedom to define a vast set of property

```
predicate dist_linf
  (a: input_type)
  (b: input_type)
  (eps:t)
  (n: int) =
  forall i. 0 <= i < n ->
    .- eps .< a i .- b i .< eps

predicate robust_to
  (model: model)
  (a: input_type)
  (eps: t) =
  forall b. dist_linf a b eps
  model.num_input ->
  model.app a = model.app b
```

Future work

- support more prominent verifiers (among ERAN, nnenum) as well as VNNLib/SMTLIB format
- how to compose several verifiers techniques (metamorphic tests plus formal verification)?
- how to choose the proper prover heuristics? more generally, how to refine and adapt proof strategies according to one's need?

CAISAR

Libre software under LGPLv2 at `https://git.frama-c.com/pub/caisar`

CAISAR maturation is partially funded by the Confiance.IA program

References

Stanley Bak. "Nnenum: Verification of ReLU Neural Networks with Optimized Abstraction Refinement". In: *NASA Formal Methods.* Ed. by Aaron Dutle, Mariano M. Moscato, Laura Titolo, César A. Muñoz, and Ivan Perez. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 19–36. ISBN: 978-3-030-76384-8. DOI: 10.1007/978-3-030-76384-8_2 (cit. on p. 4).

Souradeep Dutta, Susmit Jha, Sriram Sanakaranarayanan, and Ashish Tiwari. "Output Range Analysis for Deep Neural Networks". In: *arXiv:1709.09130 [cs, stat]* (Sept. 2017). arXiv: `1709.09130 [cs, stat]` (cit. on p. 5).

Ruediger Ehlers. "Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks". In: *arXiv:1705.01320 [cs]* (May 2017). arXiv: `1705.01320 [cs]` (cit. on p. 5).

Bibliography iii

Claudio Ferrari, Mark Niklas Mueller, Nikola Jovanović, and Martin Vechev. "Complete Verification Via Multi-neuron Relaxation Guided Branch-and-bound". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=l_amHf1oaK (cit. on p. 8).

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. "Adversarial Spheres". Jan. 2018. URL: http://arxiv.org/abs/1801.02774 (visited on 10/24/2018) (cit. on p. 8).

Bibliography iv

Julien Girard-Satabin, Aymeric Varasse, Marc Schoenauer, Guillaume Charpiat, and Zakaria Chihani. "DISCO: Division of Input Space into COnvex polytopes for neural network verification". In: *JFLA* (2021) (cit. on p. 8).

P Henriksen and A Lomuscio. "Efficient Neural Network Verification via Adaptive Refinement and Adversarial Search". In: *24th European Conference on Artificial Intelligence - ECAI 2020*. Santiago de Compostela, Spain, 2020, p. 8 (cit. on p. 8).

Guy Katz, Derek A. Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, David L. Dill, Mykel J. Kochenderfer, and Clark Barrett. “The Marabou Framework for Verification and Analysis of Deep Neural Networks”. en. In: *Computer Aided Verification*. Ed. by Isil Dillig and Serdar Tasiran. Vol. 11561. Cham: Springer International Publishing, 2019, pp. 443–452. ISBN: 978-3-030-25539-8 978-3-030-25540-4. (Visited on 07/18/2019) (cit. on p. 4).

Bibliography vi

Jianlin Li, Jiangchao Liu, Pengfei Yang, Liqian Chen, Xiaowei Huang, and Lijun Zhang. “Analyzing Deep Neural Networks with Symbolic Propagation: Towards Higher Precision and Faster Verification”. In: *Static Analysis - 26th International Symposium, SAS 2019, Porto, Portugal, October 8-11, 2019, Proceedings*. Ed. by Bor-Yuh Evan Chang. Vol. 11822. Lecture Notes in Computer Science. Springer, 2019, pp. 296–319. DOI: `10.1007/978-3-030-32304-2\_15`. URL: `https://doi.org/10.1007/978-3-030-32304-2%5C_15` (cit. on p. 8).

Bibliography vii

Christoph Müller, François Serre, Gagandeep Singh, Markus Püschel, and Martin Vechev. "Scaling Polyhedral Neural Network Verification on GPUs". In: *Proceedings of Machine Learning and Systems* 3 (2021) (cit. on p. 4).

David Shriver, Sebastian Elbaum, and Matthew B. Dwyer. "DNNV: A Framework for Deep Neural Network Verification". In: *Computer Aided Verification*. Ed. by Alexandra Silva and K. Rustan M. Leino. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 137–150. ISBN: 978-3-030-81685-8. DOI: 10.1007/978-3-030-81685-8_6 (cit. on p. 5).

Gagandeep Singh, Rupanshu Ganvir, Markus Püschel, and Martin Vechev. “Beyond the Single Neuron Convex Barrier for Neural Network Certification”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 15098–15109. URL: `http://papers.nips.cc/paper/9646-beyond-the-single-neuron-convex-barrier-for-neural-network-certification.pdf` (visited on 07/27/2020) (cit. on p. 4).

Youcheng Sun, Min Wu, Wenjie Ruan, Xiaowei Huang, Marta Kwiatkowska, and Daniel Kroening. "Concolic Testing for Deep Neural Networks". In: *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 109–119. ISBN: 9781450359375. URL: `https://doi.org/10.1145/3238147.3238172` (cit. on p. 8).

Bibliography x

Vincent Tjeng, Kai Xiao, and Russ Tedrake. "Evaluating Robustness of Neural Networks with Mixed Integer Programming". In: International Conference on Learning Representations (ICLR). 2019. URL: `https://openreview.net/pdf?id=HyGIdiRqtm` (visited on 06/19/2019) (cit. on p. 5).

Caterina Urban, Maria Christakis, Valentin Wüstholz, and Fuyuan Zhang. "Perfectly Parallel Fairness Certification of Neural Networks". In: *arXiv:1912.02499 [cs]* (Dec. 2019). arXiv: `1912.02499 [cs]` (cit. on pp. 5, 8).

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. *Beta-CROWN: Efficient Bound Propagation with Per-neuron Split Constraints for Complete and Incomplete Neural Network Robustness Verification.* Oct. 31, 2021. arXiv: 2103.06624 [cs, stat]. URL: http://arxiv.org/abs/2103.06624 (visited on 03/04/2022) (cit. on pp. 4, 8).