Critically Assessing the State of the Art in Neural Network Verification

Matthias König, Annelot W. Bosman, Holger H. Hoos, Jan N. van Rijn

ADA research group

LIACS	AI Center
Universiteit Leiden	RWTH Aachen University
The Netherlands	Germany

NeurIPS 2024
Vancouver

The AI revolution:

manually constructed algorithms

$\rightsquigarrow$

automatic adaptation to given set / distribution of inputs

$=$ automation of programming

Key idea:

automate the analysis and design of algorithms
using methods from machine learning, statistics, optimisation

$\rightsquigarrow$ empirical performance models,
automated algorithm selection & configuration, ...

Important special case:

AutoML
– automated analysis & design of ML algorithms

Neural network verification

- neural networks tend to be sensitive to input perturbations
  $\rightsquigarrow$ lack of robustness, vulnerability to adversarial attacks

horn

hot dog

Source: https://kennysong.github.io/adversarial.js/

Neural network verification

- neural networks tend to be sensitive to input perturbations
  $\rightsquigarrow$ lack of robustness, vulnerability to adversarial attacks

Stop

120 km/h

Source: `https://kennysong.github.io/adversarial.js/`

Neural network verification

- neural networks tend to be sensitive to input perturbations
  $\rightsquigarrow$ lack of robustness, vulnerability to adversarial attacks

- use formal reasoning techniques for robustness verification
  (learning + reasoning)

Local robustness in classifiers

(see, e.g., Liu *et al.*, 2021)

Key idea: ensure all x close to given input x_0
are classified with same (correct) label.

$$\forall \mathbf{x} : \|\mathbf{x} - \mathbf{x}_0\|_\infty \leq \epsilon \Rightarrow f(\mathbf{x}) = f(\mathbf{x_0})$$

Neural network verification: Challenges

- diverse network architectures: layer operations, activation functions, ...
- diverse verification approaches & algorithms: MIP-based, SMT-based, ...
- computational complexity

What is the SOTA in NN robustness verification?

- verification methods typically evaluated on
  small # of benchmarks, against different/ill-specified baselines
- VNN Competition (since 2020):
  seeks to determine "winner" based on performance ranking

Step 1: New benchmark

- large, diverse set of benchmarks (79 image classifiers)
  & verifiers (8 CPU- & GPU-based)

Step 2: In-depth empirical evaluation

What is the SOTA in NN robustness verification?

(CPU-based methods, ReLU networks, CIFAR)

Verifier	**#Solved** (*n*=1 500)
BaBSB (Bunel et al., 2018)	307
Marabou (Katz et al., 2018)	400
Neurify (Wang et al., 2018)	915
nnenum (Bak et al., 2020)	76
Verinet (Henriksen & Lomuscio, 2020)	841

What is the SOTA in NN robustness verification?

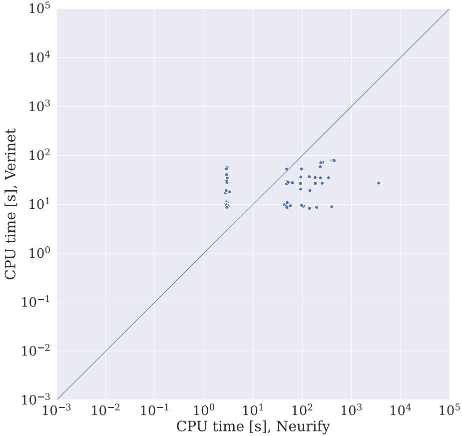
(CPU-based methods, ReLU networks, CIFAR)

Verifier	#Solved (n=1 500)
BaBSB (Bunel et al., 2018)	307
Marabou (Katz et al., 2018)	400
Neurify (Wang et al., 2018) 1	915
nnenum (Bak et al., 2020)	76
Verinet (Henriksen & Lomuscio, 2020) 2	841

What is the SOTA in NN robustness verification?

(CPU-based methods, ReLU networks, CIFAR)

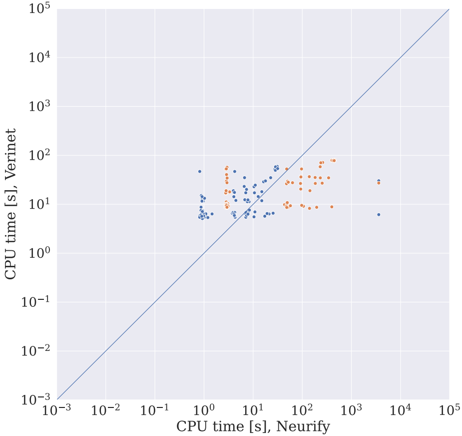
Verifier	#Solved (n=1 500)
BaBSB (Bunel et al., 2018)	307
Marabou (Katz et al., 2018)	400
Neurify (Wang et al., 2018)	915
nnenum (Bak et al., 2020)	76
Verinet (Henriksen & Lomuscio, 2020)	841



What is the SOTA in NN robustness verification?

(CPU-based methods, ReLU networks, CIFAR)

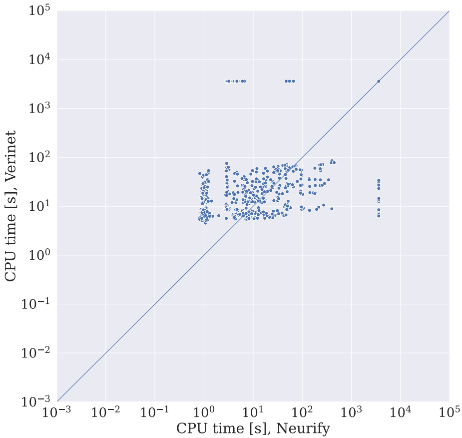
Verifier	#Solved (n=1 500)
BaBSB (Bunel et al., 2018)	307
Marabou (Katz et al., 2018)	400
Neurify (Wang et al., 2018)	1 915
nnenum (Bak et al., 2020)	76
Verinet (Henriksen & Lomuscio, 2020)	2 841



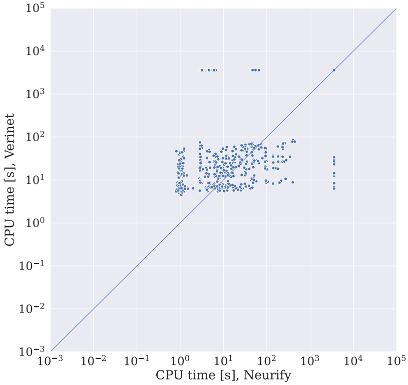
What is the SOTA in NN robustness verification?

(CPU-based methods, ReLU networks, CIFAR)

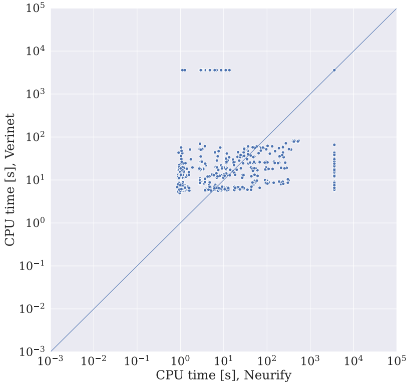
Verifier	#Solved (n=1 500)
BaBSB (Bunel et al., 2018)	307
Marabou (Katz et al., 2018)	400
Neurify (Wang et al., 2018)	915
nnenum (Bak et al., 2020)	76
Verinet (Henriksen & Lomuscio, 2020)	841



Similar results for other ϵ ...

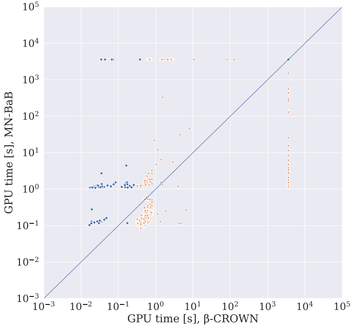


$\epsilon = 0.005$

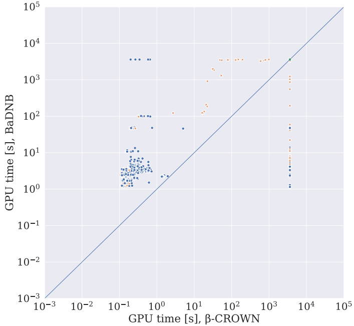


$\epsilon = 0.025$

... and for GPU-based methods



MNIST



*CIFAR*10

What is the SOTA in NN robustness verification?

- no single best verifier, performance complemantarity,
  for CPU- and GPU-based methods,
  different networks, data sets, ϵ

- not all verifiers work on all network types

- major potential for parallel portfolios, algorithm selection

$\rightsquigarrow$ König, Bosman, Hoos, van Rijn, AAAI SafeAI 2023 Workshop (best paper award); extended version published at JMLR 2024.

Conclusions

- AI revolution: explicit $\rightsquigarrow$ automated programming
- AutoML: automated analysis & design of ML algorithms
- robustness verification requires advanced reasoning techniques, adaptation to diverse network architectures, use cases
- automated configuration, selection, portfolio construction are key to next-generation NN robustness verification

Our other work on Neural Network Verification

- König, M., Hoos, H. H., Rijn, J. N. V. (2022). *Speeding up neural network robustness verification via algorithm configuration and an optimised mixed integer linear programming solver portfolio*. Machine Learning, 111(12), 4565-4584.
- Bosman, A. W., Hoos, H. H., van Rijn, J. N. (2023). *A preliminary study of critical robustness distributions in neural network verification*. In Proceedings of the 6th workshop on formal methods for ML-enabled autonomous systems.
- König, M., Zhang, X., Hoos, H. H., Kwiatkowska, M., van Rijn, J. N. (2024, August). *Automated Design of Linear Bounding Functions for Sigmoidal Nonlinearities in Neural Networks.* In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 383-398). Cham: Springer Nature Switzerland.
- Bosman, A. W., Münz, A. L., Hoos, H. H., van Rijn, J. N. (2024, July). *A Preliminary Study to Examining Per-class Performance Bias via Robustness Distributions.* In International Symposium on AI Verification (pp. 116-133). Cham: Springer Nature Switzerland.