
RAILENIUM
RAIL RESEARCH & INNOVATION
i-TRANS
INNOVATION CATALYST

Interval weight-based abstraction for neural network verification

Fateh Boudardara, Abderraouf Boussif, Pierre-Jean Meyer, Mohamed Ghazel

IRT Railenium & Univ Gustave Eiffel

RAILENIUM
RAIL RESEARCH & INNOVATION

06/09/2022

Plan

1. Introduction
2. Neural Networks
3. Neural Networks verification: formalization & challenges
4. Interval weight-based abstraction for NN
5. Conclusion

1 Introduction

- Neural networks are broadly applied to solve various problems
- Application of neural networks on Safety-Critical Systems: aviation, autonomous trains and cars
- The need of verifying neural networks models!

Stop vs 45 mph (72km/h)

Source: Robust Physical-World Attacks on Deep Learning Visual Classification, Eykholt et al, 2018

From **Panda** to **Gibbon**

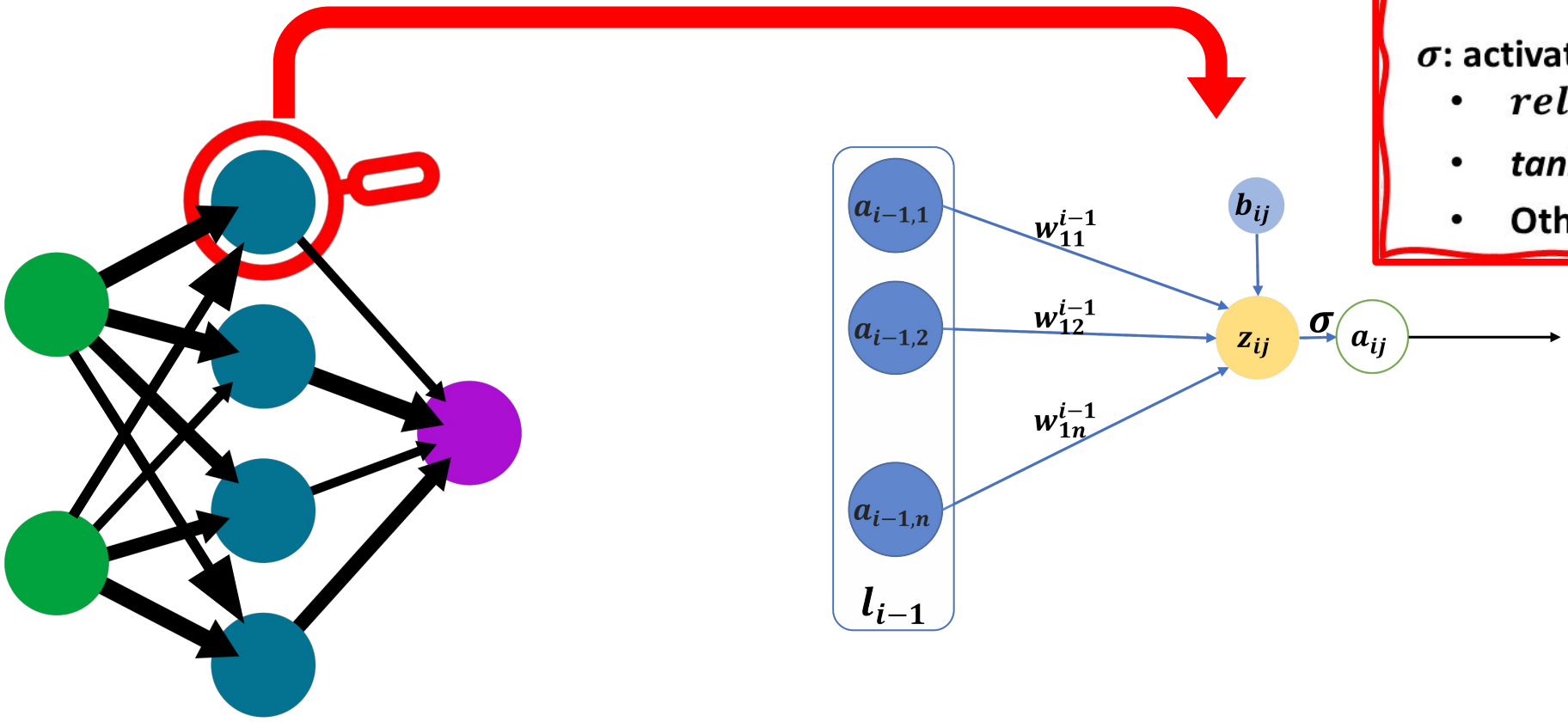
2 Neural Networks

- Connected layers via weighted edges
- Calculation of the output by propagating the values through layers

- $z_{ij} = \sum_k w_{jk}^{i-1} \times a_{i-1,k} + b_{ij}$
- $a_{ij} = \sigma(z_{ij})$

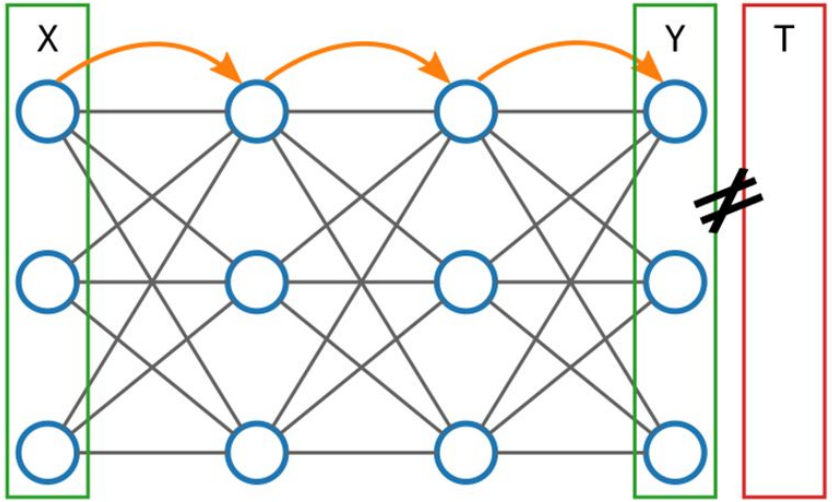
σ: activation function

- $relu(x) = max(x, 0)$
- $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- Others ...



2 Neural Networks

- Connected layers via weighted edges
- Calculation of the output by propagating the values through layers
- The performance of the network is (generally) determine by its accuracy
  - $\mathbf{accuracy} = \frac{number\ correct\ prediction\ (Y\ correct)}{total\ number\ of\ predection}$
  - Based on a test data (test set)
  - Does not cover all cases



Safety-critical systems

- The accuracy is not enough
- Need for verification & certification

Safety-critical systems

3 Neural Networks Verification

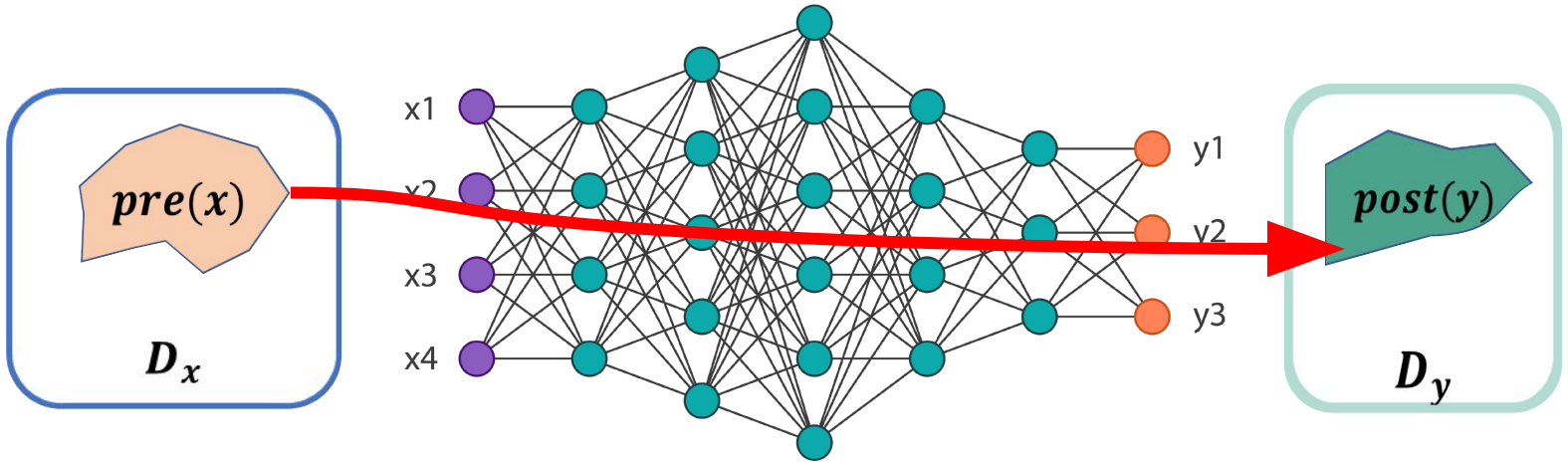
Definition

For a NN $\mathbb{N}$: $\mathbb{R}^n \longrightarrow \mathbb{R}^m$, a general way to define a verification problem of NN is:

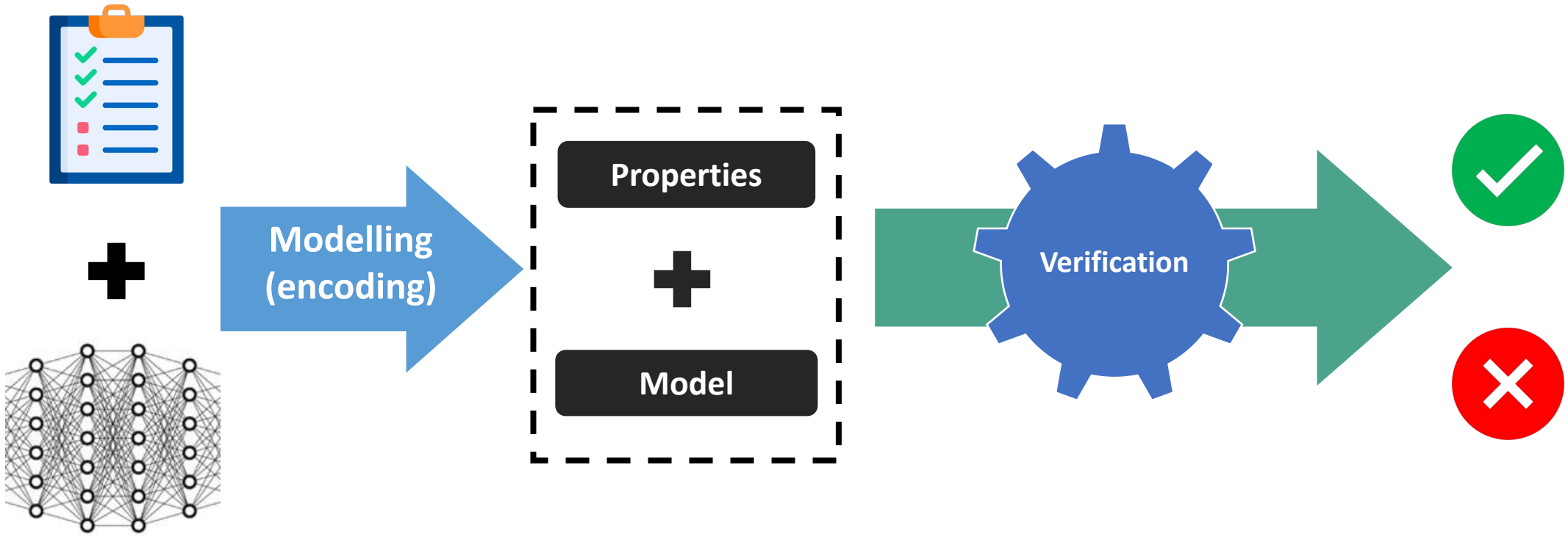
$$\forall \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{pre}(\boldsymbol{x}) \Rightarrow \boldsymbol{post}(\mathbb{N}(\boldsymbol{x}))$$

$\boldsymbol{pre}$: Constraints on the inputs
$\boldsymbol{post}$: Constraints on the outputs

NN-Verification: Under **a set of conditions on the inputs**, we want to check whether **the corresponding outputs** are also **within the safe region** that is defined by the set of constraints $\boldsymbol{post}$



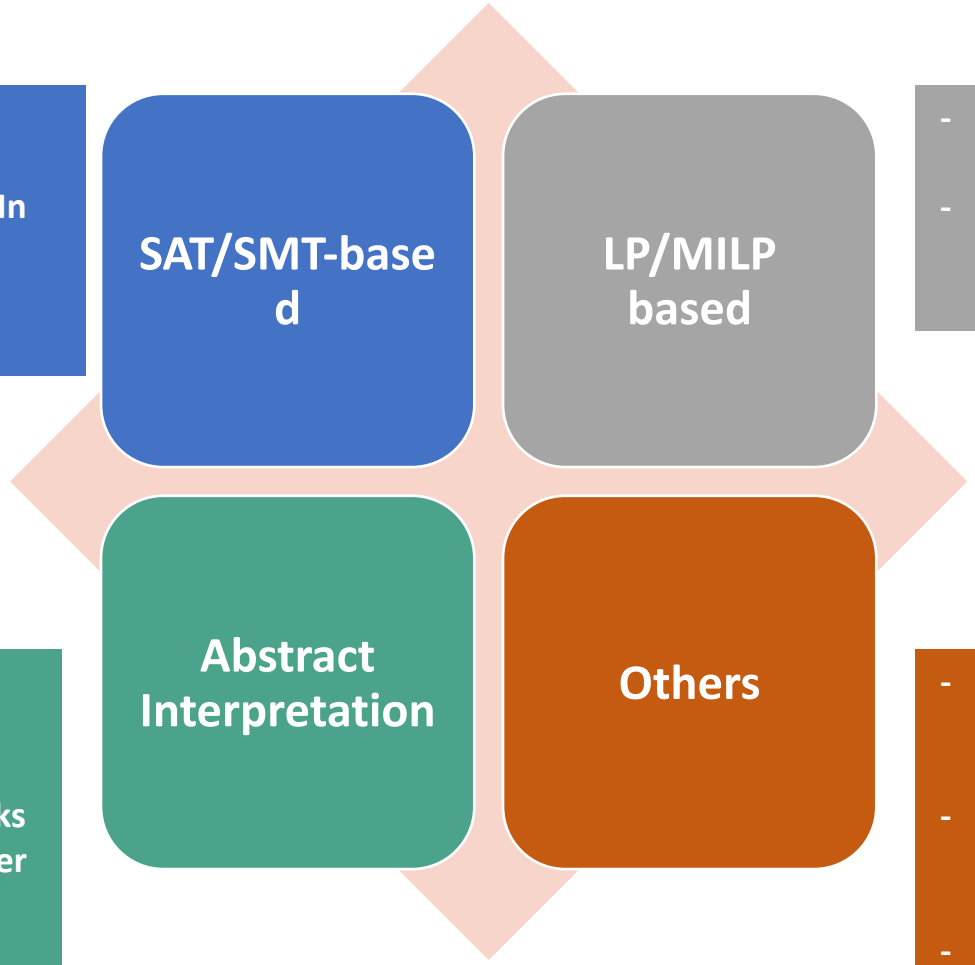
3 Neural Networks Verification



3 Neural Networks Verification

Approaches

- L. Pulina and T. Armando "An abstraction-refinement approach to verification of artificial neural networks". In CAV 2010.
- G. Katz et al. "Reluplex: An efficient SMTsolver for verifying deep neural



- S. Dutta et al. "Output Range Analysis for Deep Feedforward Neural Networks". In NFM 2018.
- V. Tjeng et al. "Evaluating Robustness of Neural Networks with Mixed Integer Programming" In ICLR 2019.

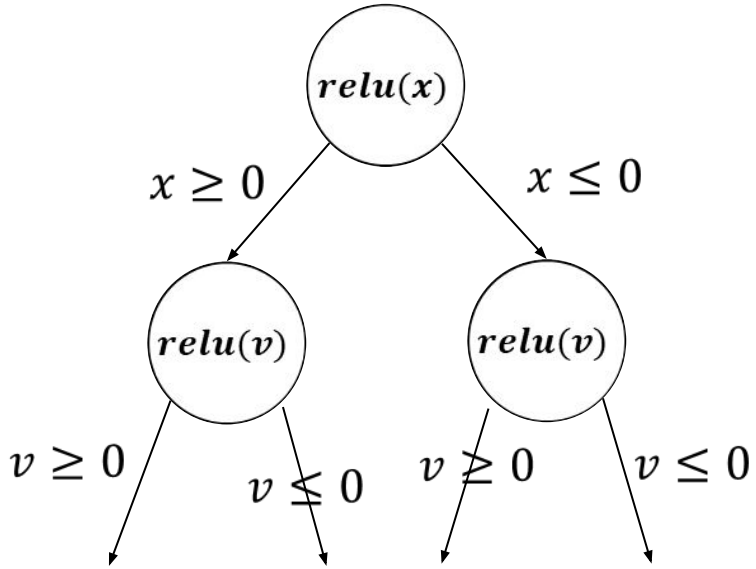
- T. Gehr et al. "AI2: Safety and Robustness Certification of Neural Networks with
- Abstract Interpretation". In S&P 2018.
- J. Li et al. "Analyzing Deep Neural Networks with Symbolic Propagation: Towards Higher Precision and Faster Verification". In SAS 2019.

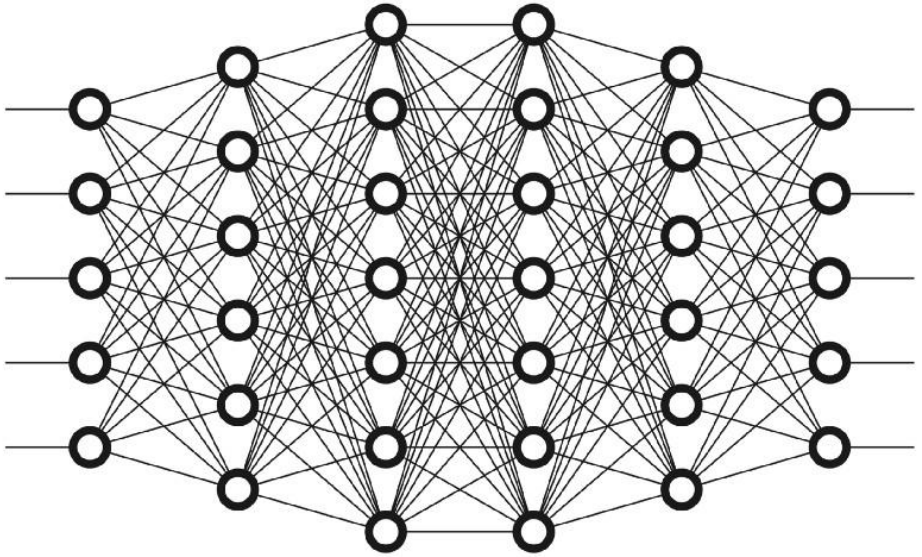
- H.-D. Tran et al. "Star-Based Reachability Analysis of Deep Neural Networks". In FM 2018.
- W. Ruan et al. "Reachability Analysis of Deep Neural Networks with Provable Guarantees". In IJCAI, 2018
- C. Urban and A. Miné. "A review of formal methods applied to machine learning", arXiv 2021

3 Neural Networks Verification

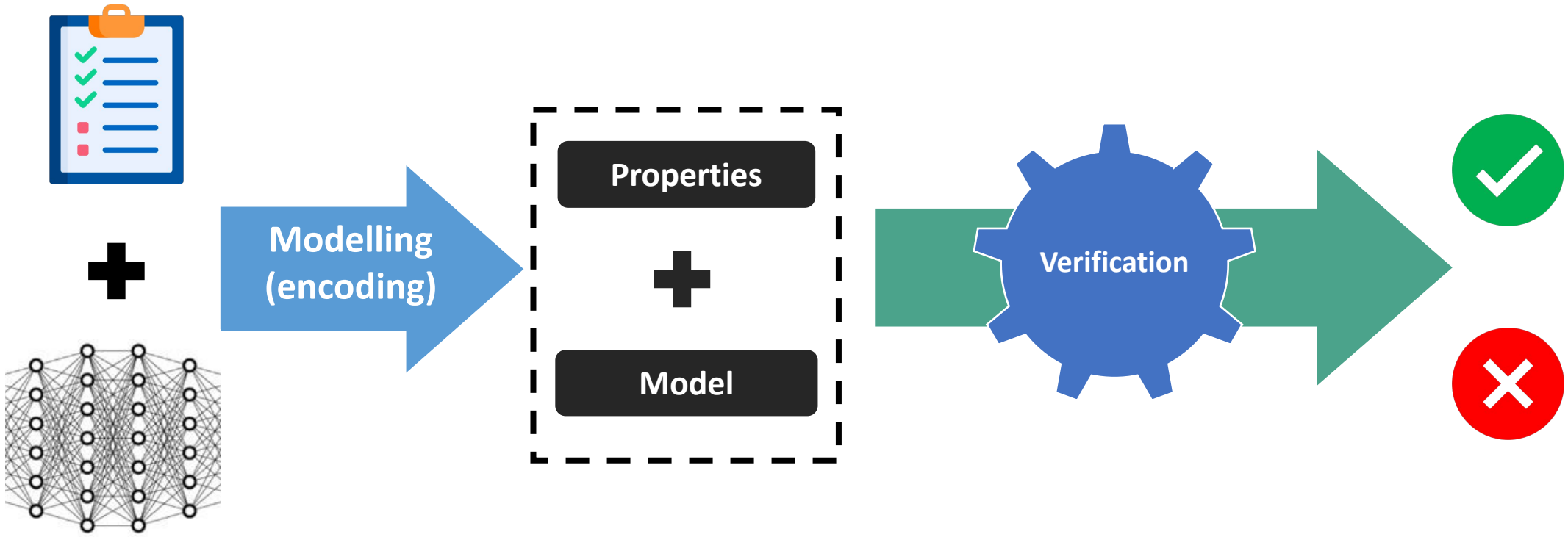
Challenges

- **Formal specification:** there are some NN properties we do not know how to express, formally
- **Scalability: the existing verification methods do not scale to large networks**



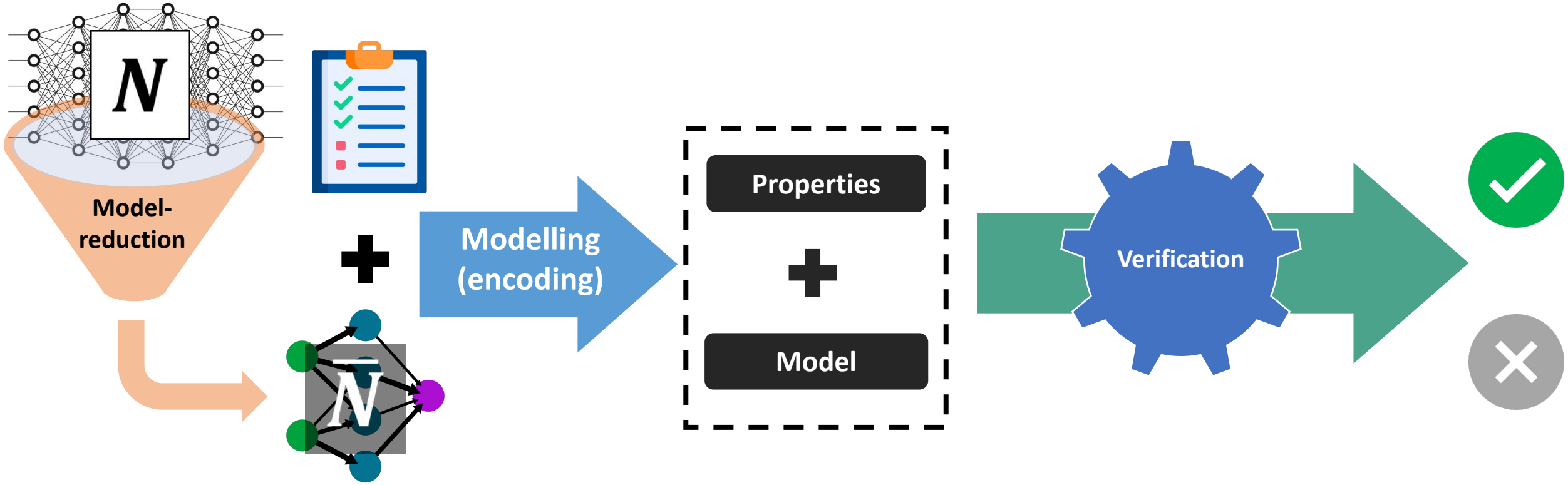


3 Neural Networks Verification



3 Neural Networks Verification

Model reduction



3 Neural Networks Model-Reduction

- For *Relu-NN*
- 4 categories: positive, negative, increasing and deacreasing
- Abstract neurons belonging to the same category

Elboher et al. 2020

Prabhakar and Afzal 2019

- For *Relu-NN*
- Change NN to INN
- Interval hull of the incoming and the outgoing weights

Ashok et al. 2020

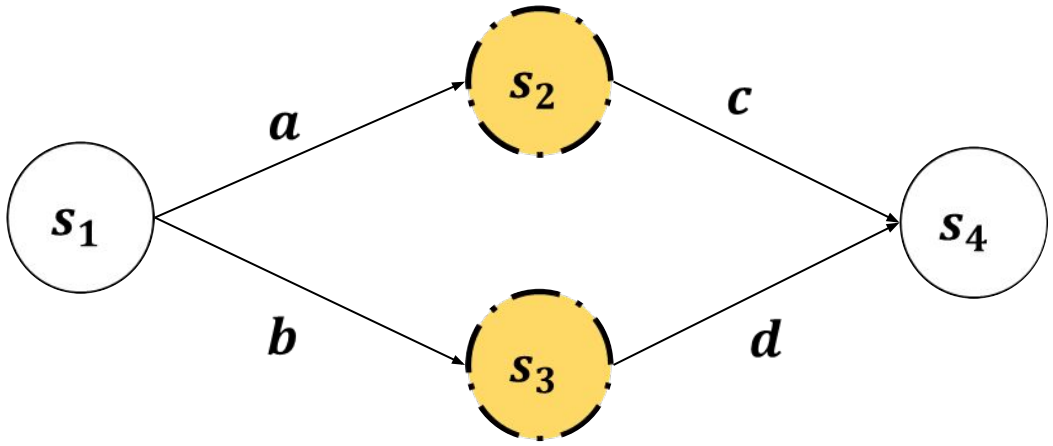
- For *Relu-NN*
- Classification of neurons: k-mean clustering
- Replace each cluster of neurons by its representative

Others

- D. Shriver et al. "Refactoring neural networks for verification“, arXiv 2019.
- M. Sotoudeh and A. V. Thakur. Abstract neural networks. In SAS 2020
- P. Prabhakar. "Bisimulations for Neural Network Reduction“. In VMCAI 2022.

4 Interval weight-based abstraction for NN

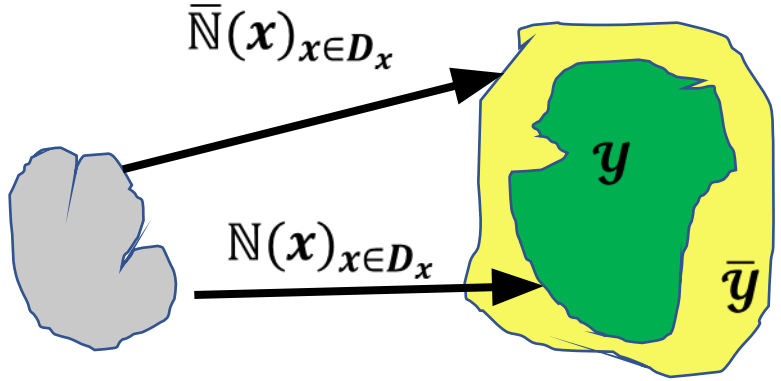
Contribution



4 Interval weight-based abstraction for NN

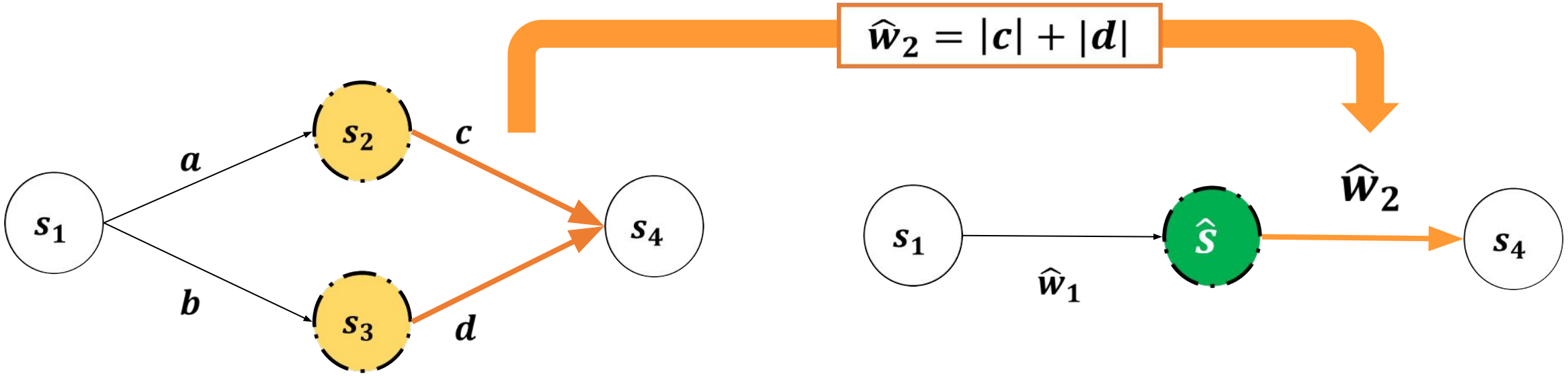
Contribution

- **Model reduction method for *Relu-NN* and *Tanh-NN* :**
  - s_1, s_2, s_3 and s_4 have the same activation function (either ***Relu*** or ***Tanh***)
- **Can abstract all nodes of a hidden layer into one abstract node**
- **Ensure the over-approximation : $N \subseteq \overline{N}$**
  - $\overline{N} \models P \Rightarrow N \models P$



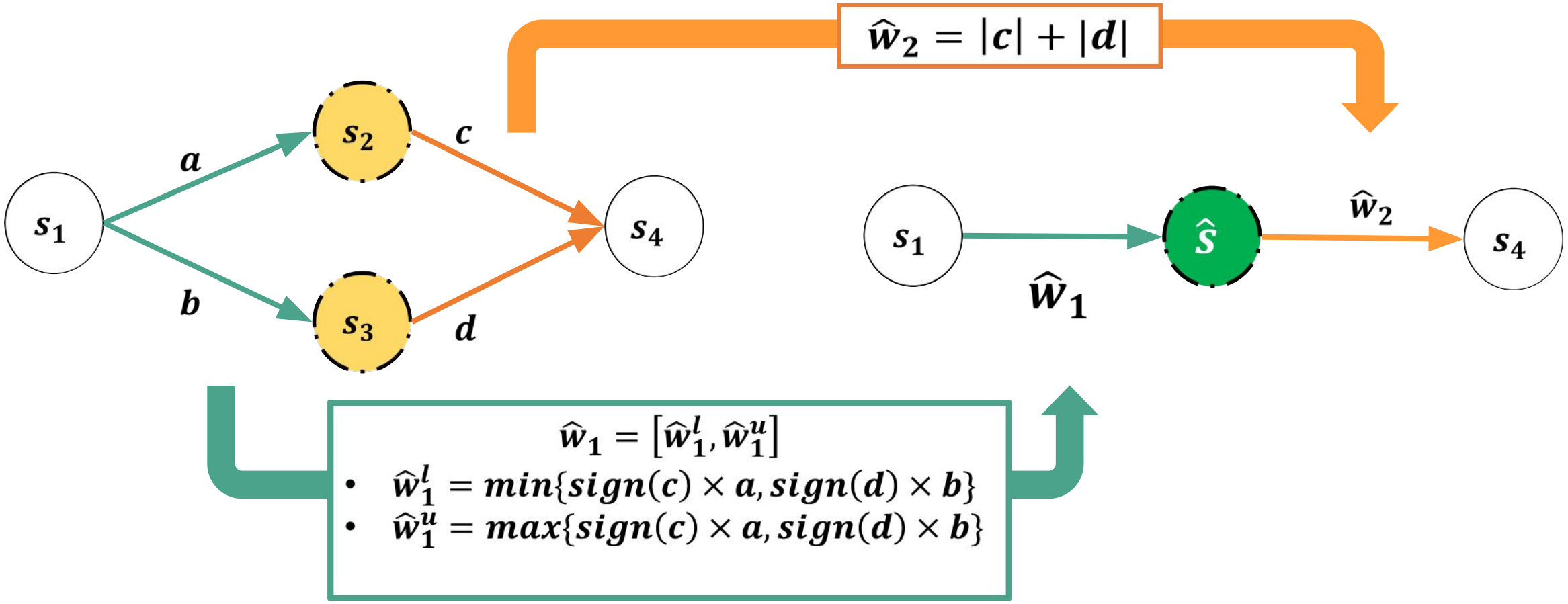
4 Interval weight-based abstraction for NN

Contribution



4 Interval weight-based abstraction for NN

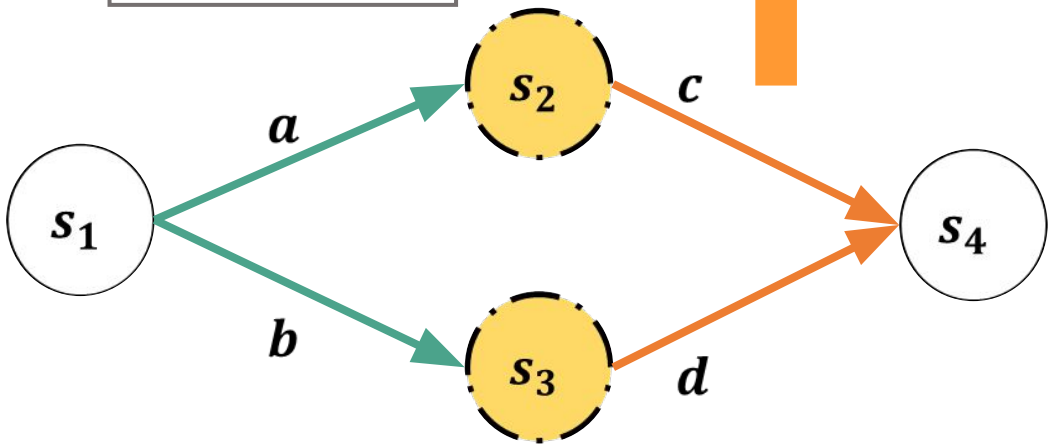
Contribution



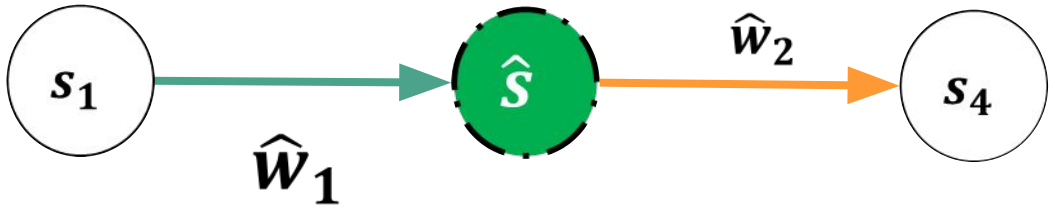
4 Interval weight-based abstraction for NN

Contribution

Relu-NN



$$\hat{w}_2 = |c| + |d|$$



$$\hat{w}_1 = [\hat{w}_1^l, \hat{w}_1^u]$$

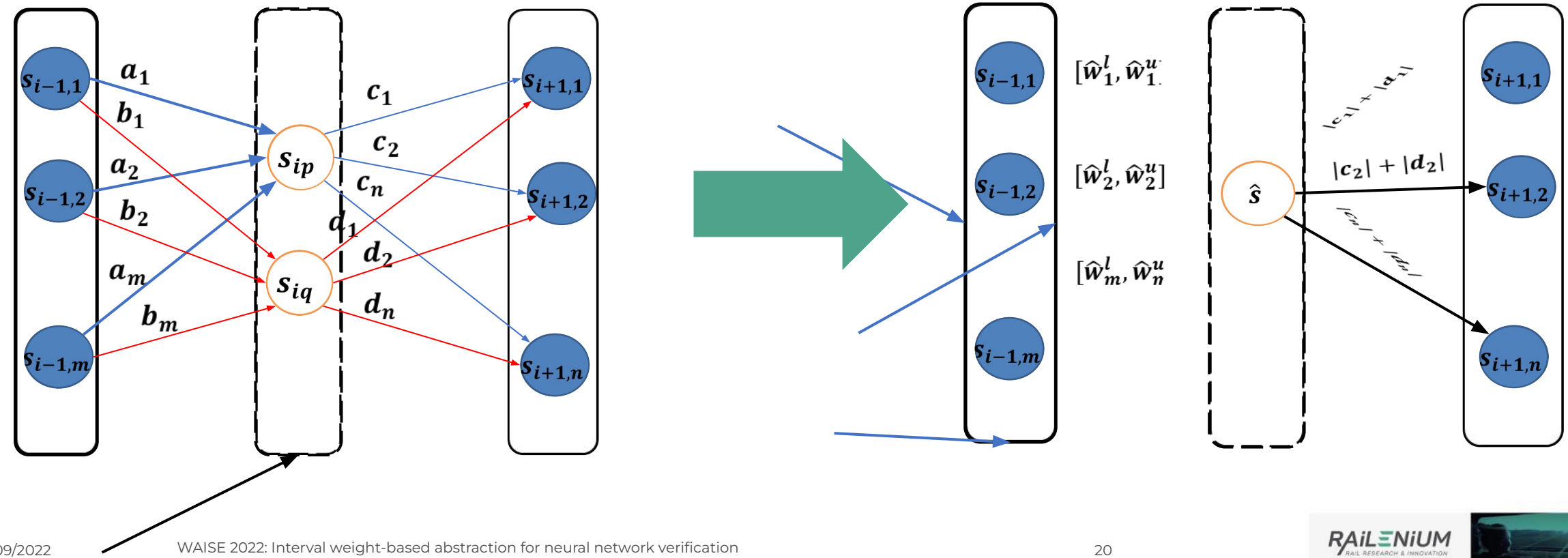
- $\hat{w}_1^u = max\{sign(c) \times a, sign(d) \times b\}$
- $if\ \Big((sign(a) = sign(c) \wedge sign(b) = sign(d))\Big)$:

$$\hat{w}_1^l = min\{a, b\}$$

4 Interval weight-based abstraction for NN

Contribution

$$\hat{w}_k^l = \min_{1 \le j \le n} \{sign(c_j) \times a_k, sign(d_j) \times b_k\}$$
$$\hat{w}_k^u = \max_{1 \le j \le n} \{sign(c_j) \times a_k, sign(d_j) \times b_k\}$$

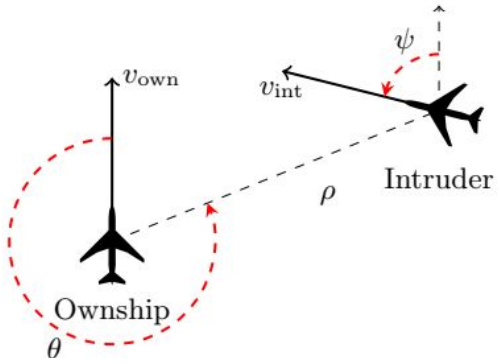


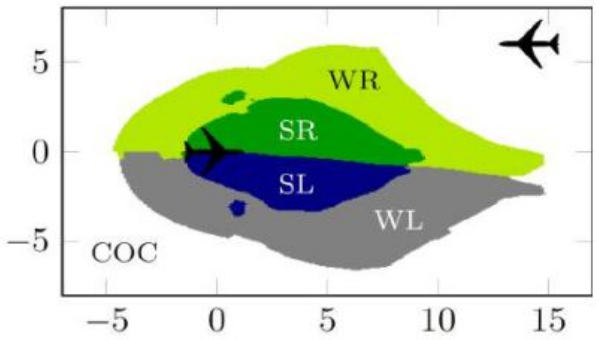
4 Interval weight-based abstraction for NN

Results

ACAS Xu : Airborne Collision Avoidance System for Unmanned Aircraft

- 45 Relu-NNs





❑ 7 Inputs:

- Distance
- Speed of the ownship
- Speed of the intruder
- Angle to intruder relative to ownship headir
- Heading angle to intruder relative to ownshi
- Time until loss of vertical separation
- Previous advisory

- [205, 400]
- ≥ 1000
- [700, 800]
- [0.2, 0.4]
- [-3.14, -3.14 + 0.05]
- COC
- 0

❑ 5 outputs (horizontal advisories):

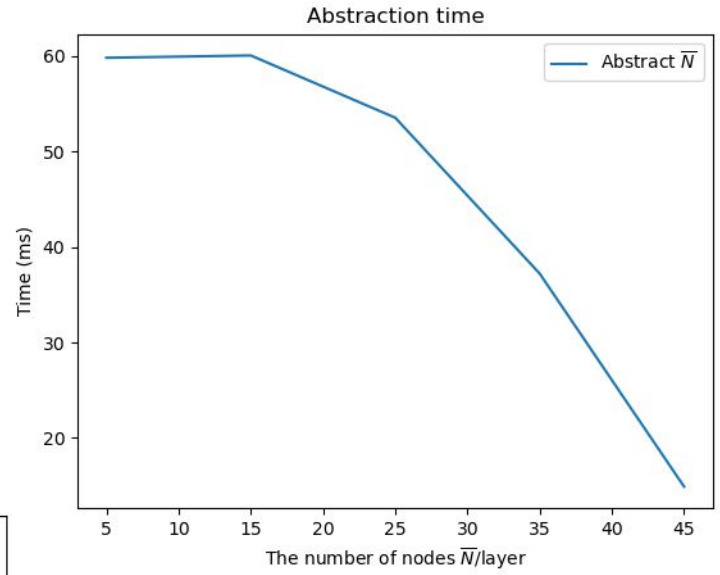
- Weak Right
- Strong Right
- Strong Left
- Weak Left
- Clear of Conflict

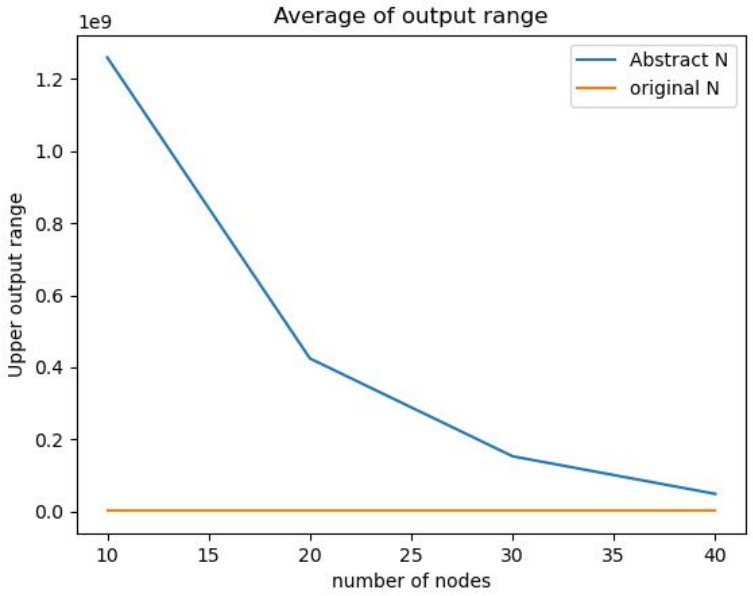
- Strong Right

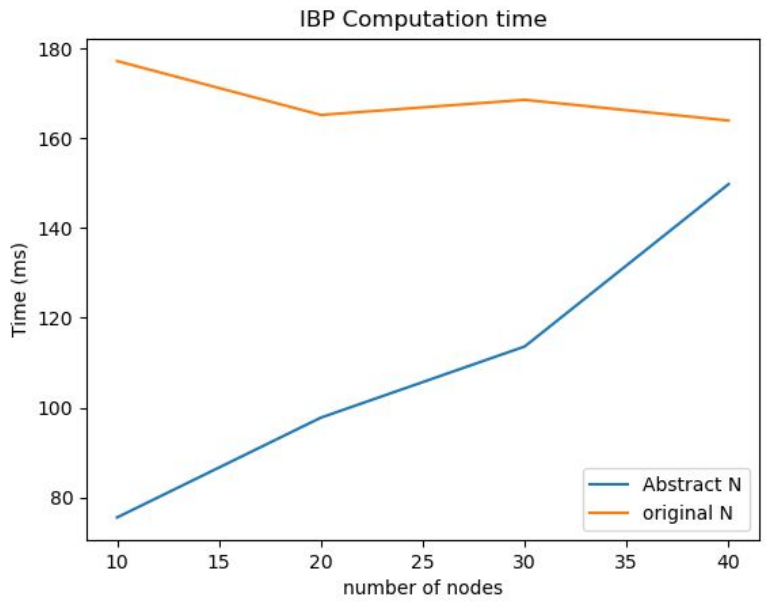
4 Interval weight-based abstraction for NN

Results

- Over 50 runs:
  - The abstraction time
  - The IBP Computation time
  - The average of the outputs







5 Conclusion

- A NN reduction method for the sake of enhancing the efficiency of various analysis operations (verification)
- The proposed method considers feed-forward *Relu-NNs* and *Tanh-NNs*
- Application on the ACAS Xu benchmark

Future works

- Apply the proposed method on other benchmark problems
- Extend the approach to support more activation functions
- Add heuristics for nodes selections

Thank you!

YOUR CONTACT

Fateh Boudardara
PhD student

fateh.boudardara@railenium.eu

RAILENIUM
180 rue Joseph-Louis Lagrange
59308 VALENCIENNES Cedex

www.railenium.eu