Università di Genova

Counter-example Guided Abstract Refinement for Verification of Neural Networks

Stefano Demarchi | Dario Guidotti

CPS Workshop
19/09/2022

Agenda

- Context
- Abstraction algorithms
- Refinement / CEGAR
- Evaluation
- Remarks

Context

Neural Networks

Neural Networks (NNs) are widespread and «fashionable»

NNs provide fast classification and regression results

Many success stories in Natural Language Processing, Computer Vision, Control...

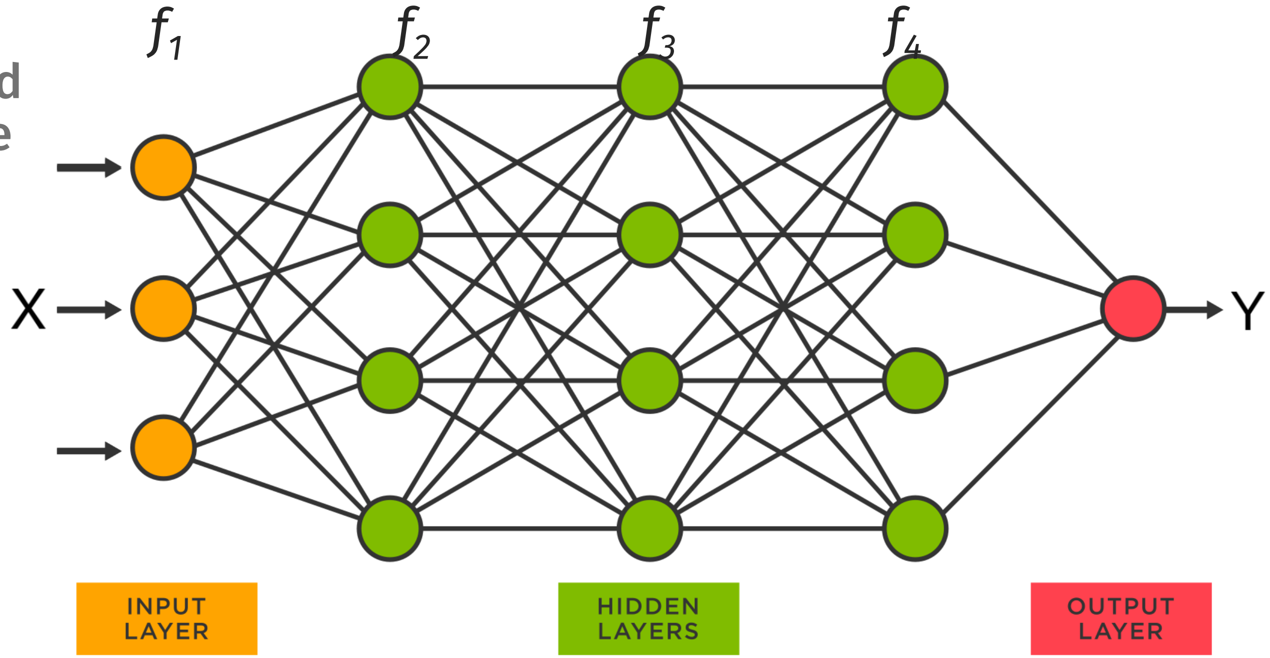
Context

Neural Networks

Nodes are connected through linear affine mappings

(weighted sum)

$y = Wx + b$



Nodes represent non-linear activation functions

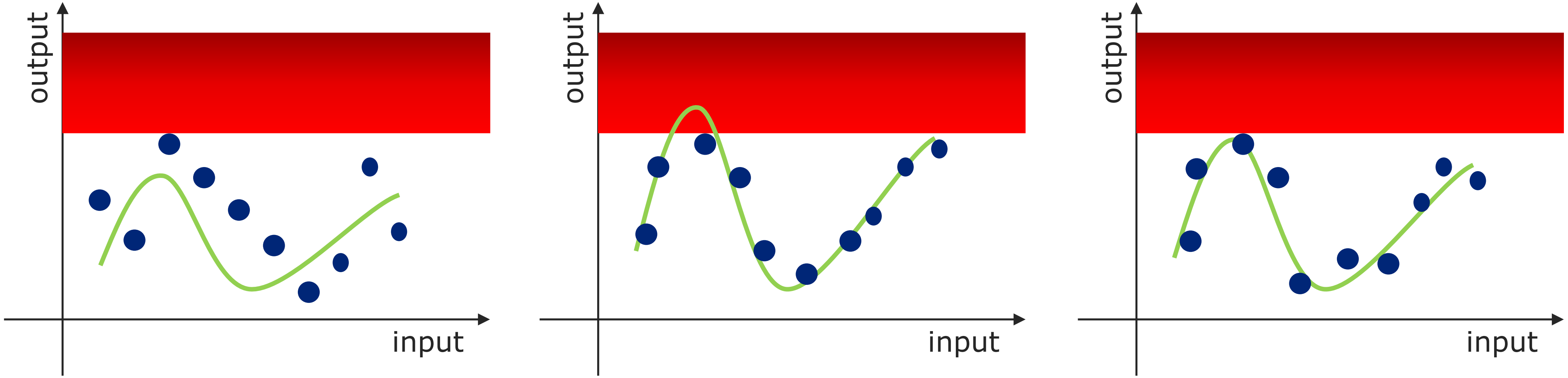
(ReLU, Sigmoid, ...)

$\mathrm{ReLU}(x) = \max(0, x)$

$$Y = f_4(\,f_3(\,f_2(\,f_1(\,X\,))))$$

Context

NN Verification



Safe, but not accurate

Accurate, but not safe

Accurate and safe

= Neural Network

= Training sample

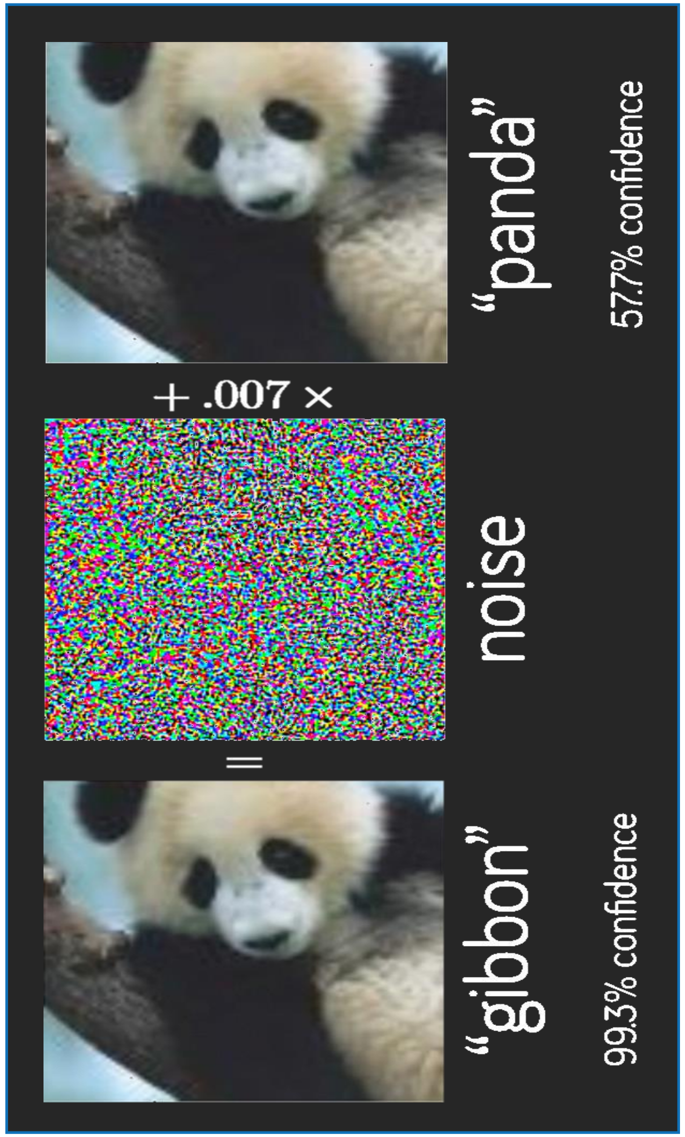
= Unsafe zone

Context

NN Verification

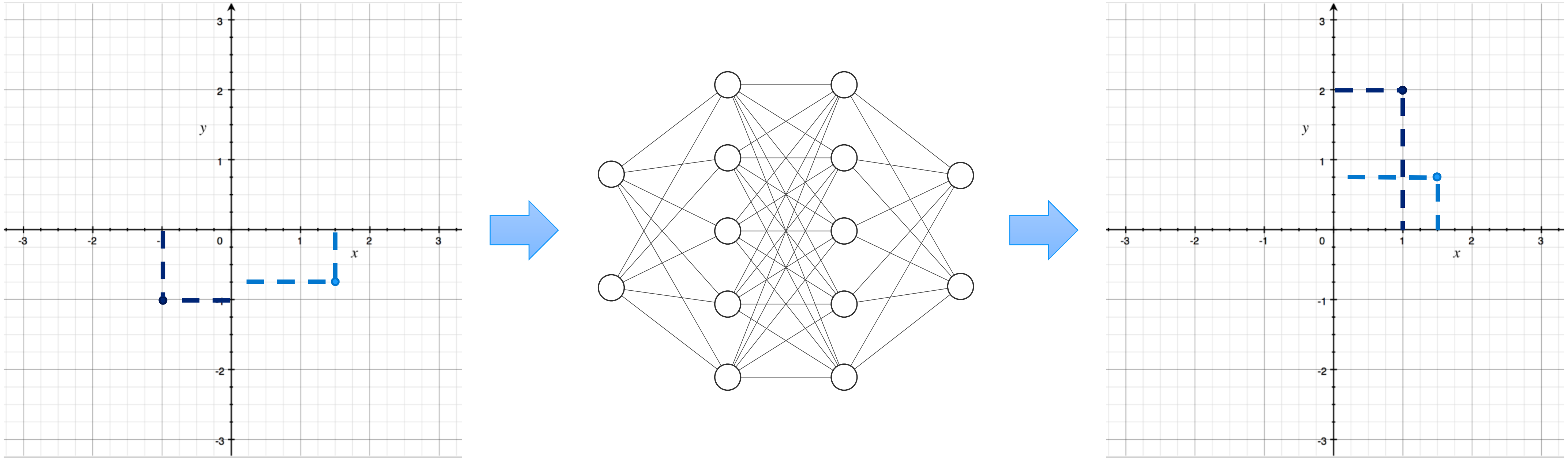
Adversarial perturbation

- Minimal changes impact the classification
- Concerns for safety-critical applications
- Formal Verification for Input/Output specifications



Abstraction algorithms

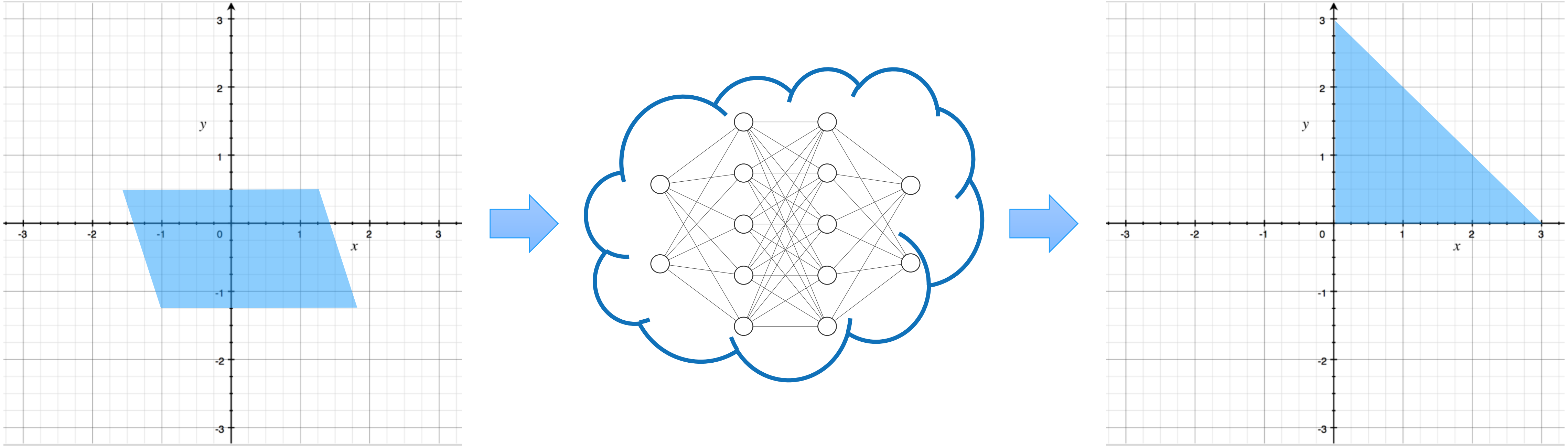
Abstract Interpretation



Infeasible to run the NN on all inputs

Abstraction algorithms

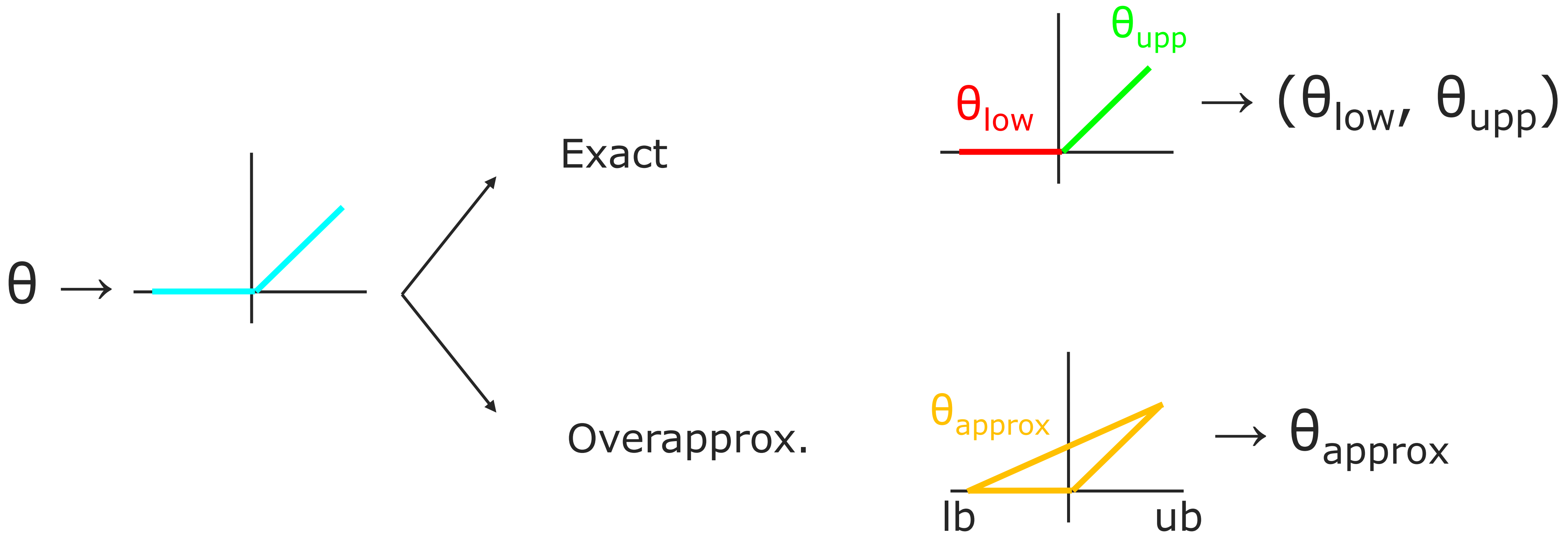
Abstract Interpretation



Abstraction provides finite approximation of (potentially) infinite sets

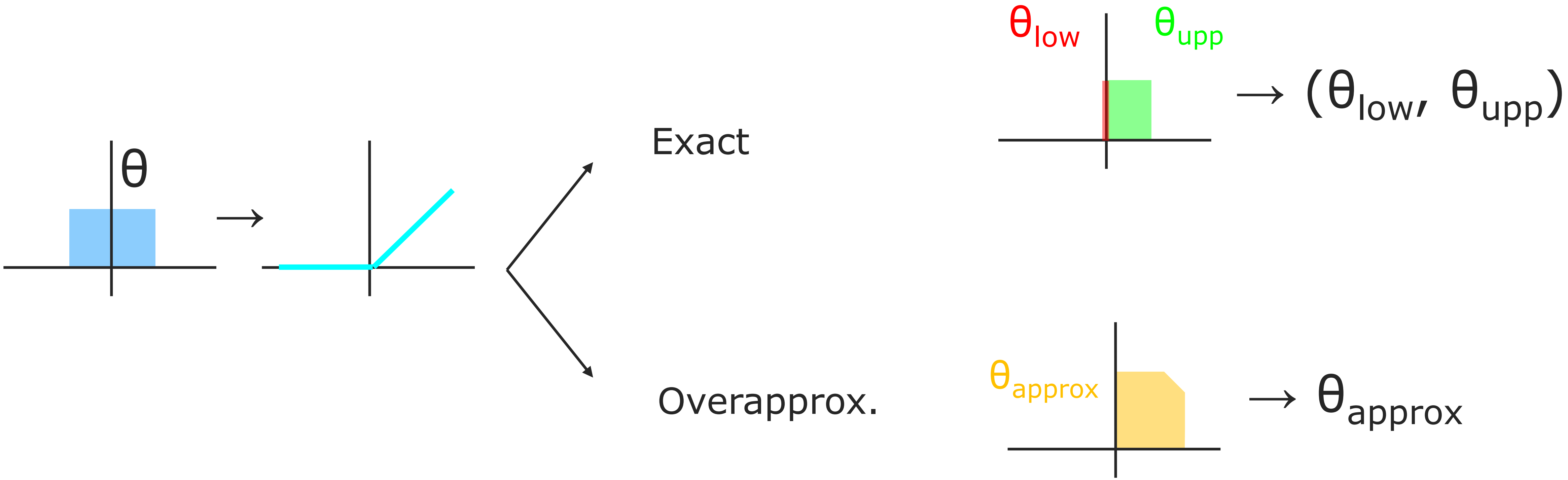
Abstraction algorithms

NN abstraction – ReLU layers



Abstraction algorithms

NN abstraction – ReLU layers



Abstraction algorithms

Complete

Propagates the exact transformation of the input

If the input is unstable then we split

In the worst case the input set grows exponentially with the number of neurons

Over-approximate

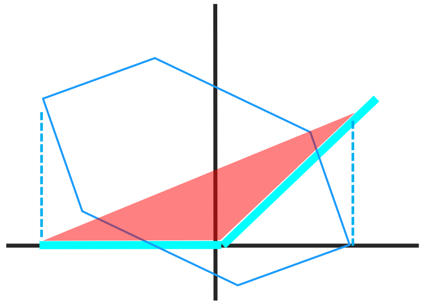
Propagates an approximation of the input

If the input is stable we keep the exact transformation

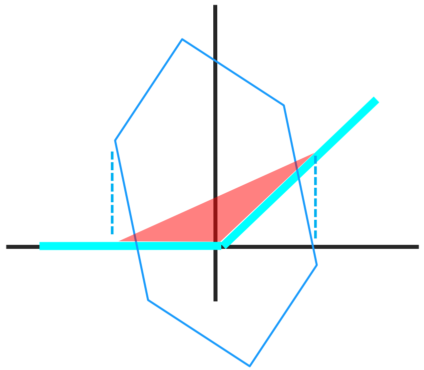
The approximation introduces a new variable and 3 constraints for each neuron

Abstraction algorithms

Mixed algorithm



The over-approximation abstract area depends on the set bounds



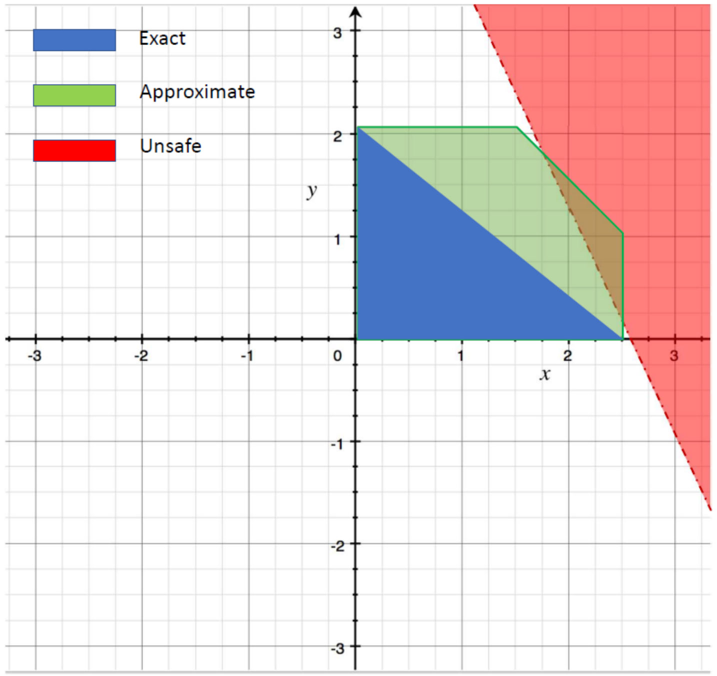
Approximate all neurons **but** the one with the greatest area

Still approximate, but faster and more precise

CEGAR

Approximation refinement

- If the exact output violates the safety property we can identify the unsafe input
- Not the same with the approximate output
- If we find a counterexample we can prove that the property is not verified



CEGAR

Approximation refinement

We enhance the refinement measuring neuron relevance[1]

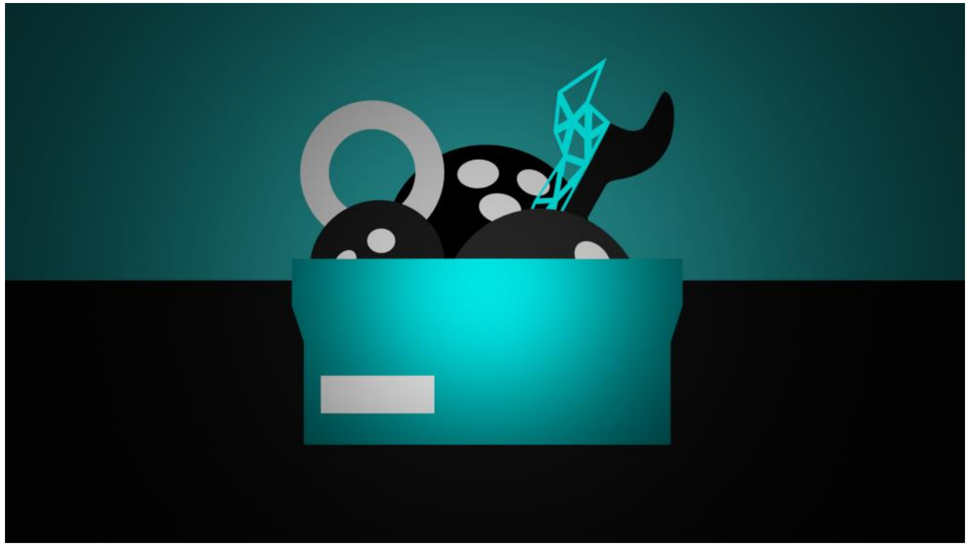
Relevance is computed propagating the prediction backwards on samples from the output set

Measures the neurons contribution to the result

[1] *Montavon, G. et al – Layer-Wise Relevance Propagation: An Overview; Explanable AI, 2019*

NeVer Tools

A suite of tools for the manipulation and verification of NNs



pyNeVer – baseline API

CoCoNet – Tool for NNs manipulation and conversion

NeVer 2 – Tool for NNs learning and verification

neuralverification.org | github.com/NeVerTools

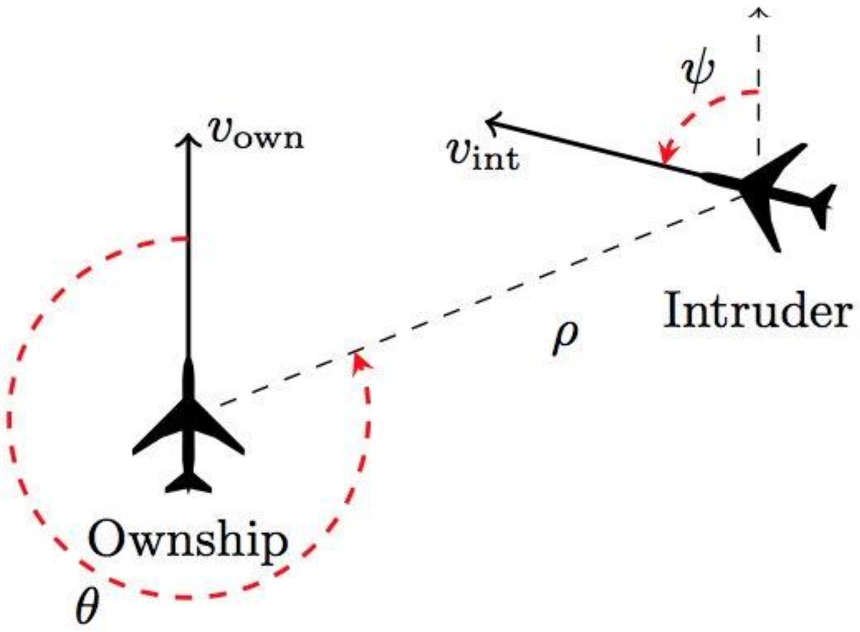
Evaluation

Verification of ACAS-Xu properties

Classic verification benchmark

Properties expressed for never issuing a Clear-Of-Conflict command

All properties are known to be verified



Evaluation

PROPERTY	NETWORK	MIXED		CEGAR-PS		CEGAR-mR	
		TIME	VERIFIED	TIME	VERIFIED	TIME	VERIFIED
# 3	1_1	13	T	10	3/10	9	9/10
	1_3	10	T	14	6/10	10	0/10
	2_3	7	T	10	9/10	7	6/10
	4_3	15	T	17	10/10	14	10/10
	5_1	6	T	11	10/10	9	10/10
# 4	1_1	11	T	10	0/10	9	0/10
	1_3	8	T	16	0/10	11	0/10
	3_2	12	T	12	10/10	12	10/10
	4_2	12	T	11	10/10	12	10/10

Remarks

Explainability insights

We tried to enhance our refinement procedure

No clear improvement in results, but interesting insights

Unpredictability due to sampling

Working on better sampling/counterexample identification

Università
di Genova