AI^2: Safety and Robustness Certification of Neural Network with Abstract Interpretation

Gehr, Mirman, Drachsler-Cohen, Tsankov, Vechev, Chaudhuri

2018 IEEE Symposium on Security and Privacy

Verifica del Software

A.A. 2019-2020

Niccolo' Vettorello 1237291

INTRODUCTION

- First *sound* and *scalable* static analyzer for deep neural networks
- It over-approximates the behavior of the network and can automatically prove safety properties on realistic N. N.
- Rethinks the concept of analysis using the well-studied abstract interpretation framework as theoretical base
- Can handle Fully Connected N. N. and Convolutional N. N. that use the ReLU activation function and Max Pooling layers

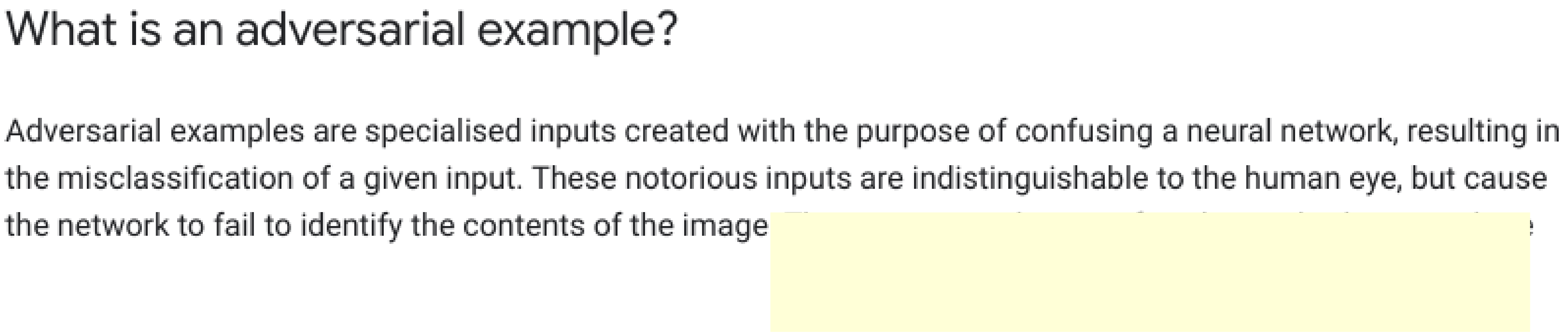
INTRODUCTION

Deep Neural Networks are used in safety-critical applications with an ever-increasing rate:

- Self-driving cars
- Malware detection
- Pedestrian detection

INTRODUCTION

Nonetheless, N. N. are not perfect and can be vulnerable to the so-called adversarial examples



What is an adversarial example?

Adversarial examples are specialised inputs created with the purpose of confusing a neural network, resulting in the misclassification of a given input. These notorious inputs are indistinguishable to the human eye, but cause the network to fail to identify the contents of the image [redacted]

Adversarial examples are effectively obtained by perturbing some sample with respect to a certain feature, giving rise to different type of attack, e.g. Brightness Attack, Fast Gradient Signed Method

---

https://www.tensorflow.org/tutorials/generative/adversarial_fgsm

INTRODUCTION

Local robustness (or **robustness**): *all samples in the neighborhood of a given input are classified with the same output*

- Some people have focused on increasing robustness of networks: ad hoc training techniques have been developed
- Other tried to find method to measure how robust a network was

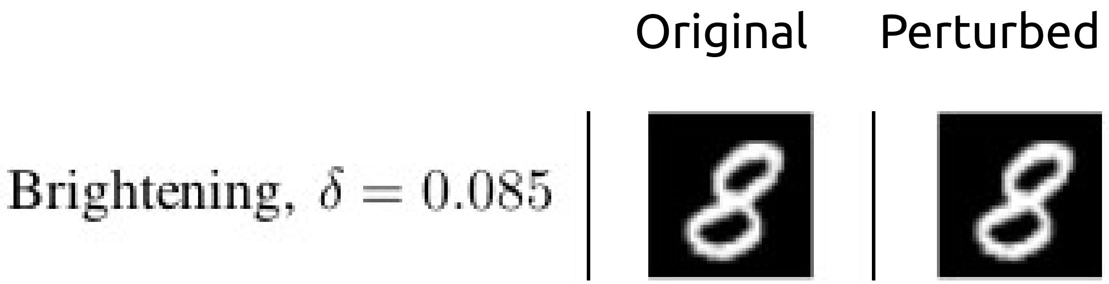
In any case, no existing analyzer deals with CNN, a widely used type of N.N.

Why the progress in this area seems so slow?

INTRODUCTION

PROBLEM

Even with basic samples and simple perturbations there is a combinatorial explosion, and thinking to test every possible input-output couple is absolutely not reasonable (and not tractable)



The brightening attack brings all pixes above the threshold of 1 – δ to max brightness.
For the simple example above there are 10^{1154} possible perturbations.

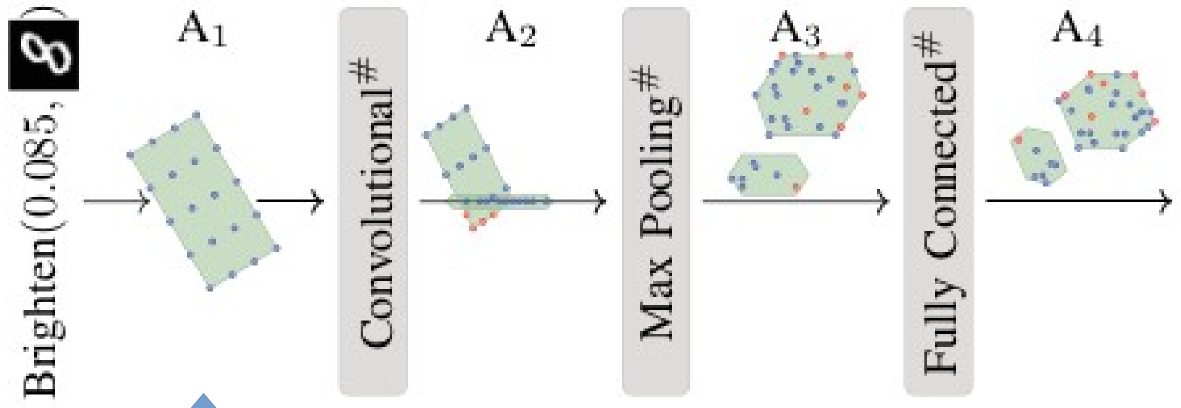
INTRODUCTION

POSSIBLE SOLUTION

Use of Abstract Interpretation

Approximation of all possible outputs

Input and perturbation type



Approximation of all possible perturbations

Abstraction of a layer

INTRODUCTION

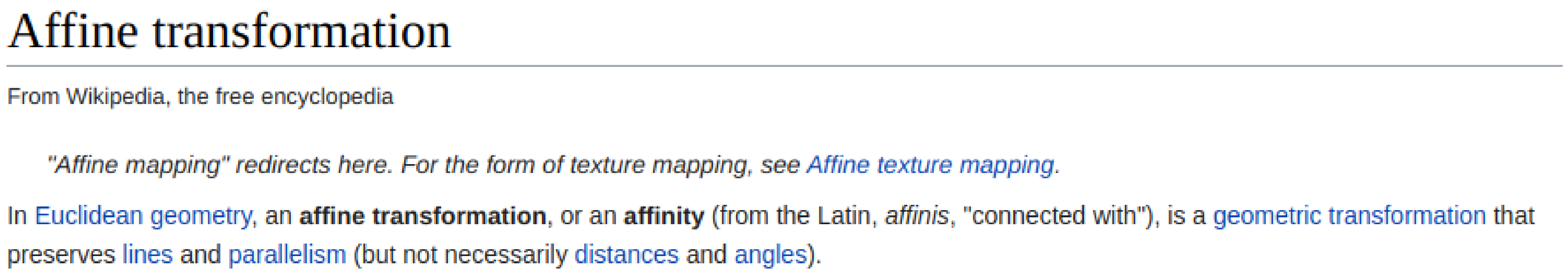
MAIN CONTRIBUTIONS

1) A sound and scalable method for analysis of deep neural networks

2) A prototype of analyzer, tested and evaluated, that is able to handle Feed-Forward and Convolutional N.N.

3) The detection of an application for AI^2: evaluation of neural networks defenses in terms of robustness

REPRESENTING N. N. AS A CAT FUNCTION


Affine transformation

From Wikipedia, the free encyclopedia

"Affine mapping" redirects here. For the form of texture mapping, see Affine texture mapping.

In Euclidean geometry, an **affine transformation**, or an **affinity** (from the Latin, *affinis*, "connected with"), is a geometric transformation that preserves lines and parallelism (but not necessarily distances and angles).

$\downarrow$

Conditional **A**ffine **T**ransformation: an Affine Transformation function guarded by a logical constraint

We'll show how to represent the network as a CAT function. In this way we can use the Abstract Interpretation framework to handle it.

REPRESENTING N. N. AS A CAT FUNCTION

NOTATION

1) x_i is the ith entry in a vector $\mathbf{x} \in \mathbf{R}^n$

2) For a matrix $W \in \mathbf{R}^{n\text{x}m}$

   a) W_i is the ith row

   b) $W_{i,j}$ is the element at ith row, jth column

REPRESENTING N. N. AS A CAT FUNCTION

CAT FUNCTIONS

Returns $f_i(\mathbf{x})$ for the first E_i satisfied by $\mathbf{x}$

$f(\mathbf{x})$ ::= $W \cdot \mathbf{x} + \mathbf{b}$ |

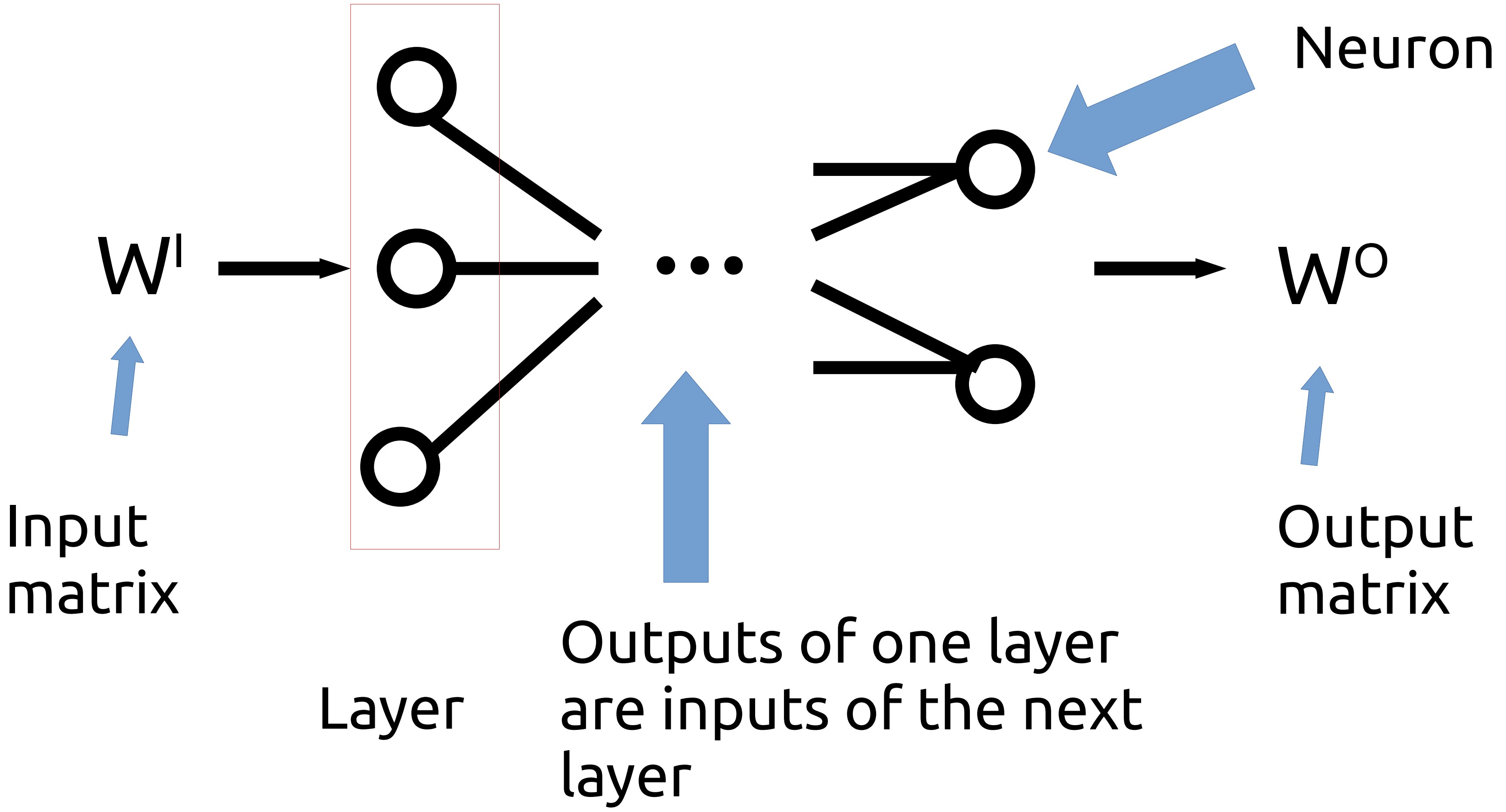
case E_1: $f_1(\mathbf{x})$, . . ., *case* E_k: $f_k(\mathbf{x})$ |

$f(f'(\mathbf{x}))$

E ::= $E \wedge E$ | $x_i >= x_j$ | $x_i >=0$ | $x_i < 0$

REPRESENTING N. N. AS A CAT FUNCTION

GENERAL STRUCTURE OF A NETWORK



REPRESENTING N. N. AS A CAT FUNCTION

RESHAPING OF INPUTS

Inputs are often in the form of a three-dimensional matrix or of a vector.

We can reshape a 3D matrix $\mathbf{W} \in \mathbf{R}^{m x n x r}$ into a vector $\mathbf{x}^v \in \mathbf{R}^{n \cdot m \cdot r}$ by listing elements in the matrix by depth, column and row.

More formally, given **x**

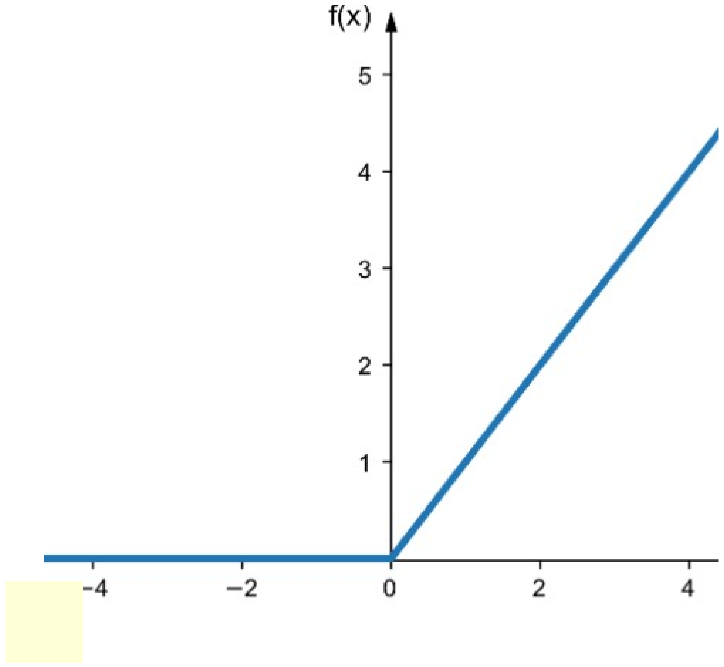
$$\mathbf{x}^v = (x_{1,1,1}, \ldots, x_{1,1,r}, x_{1,2,1}, \ldots, x_{1,n,r}, x_{2,1,1}, \ldots, x_{m,n,r})^T$$

REPRESENTING N. N. AS A CAT FUNCTION

ReLU ACTIVATION FUNCTION

$x \in \mathbf{R}$

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



$\mathbf{x} \in \mathbf{R}^m$

$\mathrm{ReLU}(\mathbf{x}) = (\mathrm{ReLU}(x_1), \ldots, \mathrm{ReLU}(x_m))$

REPRESENTING N. N. AS A CAT FUNCTION

ReLU TO CAT

$$\mathrm{ReLU} = \mathrm{ReLU}_n \circ \ldots \circ \mathrm{ReLU}_1$$

where

$$\mathrm{ReLU}_i(\mathbf{x}) = \textit{case}\ (x_i >= 0){:}\ \mathbf{x},$$
$$\textit{case}\ (x_i < 0){:}\ I_{i\leftarrow 0} \cdot \mathbf{x}$$

Returns the vector with the ith value set to 0

$I_{i\leftarrow 0}$ is the identity matrix with the ith row replaced by zeros

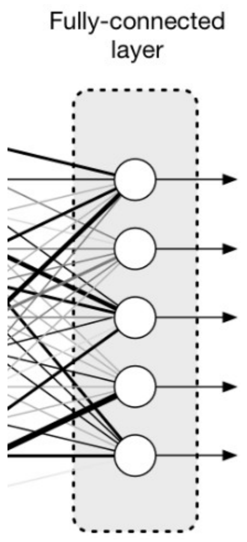
REPRESENTING N. N. AS A CAT FUNCTION

FULLY CONNECTED (FC) LAYER

An FC Layer can be seen as the function

$$FC_{W, \mathbf{b}}: \mathbf{R}^m \rightarrow \mathbf{R}^n$$

$$FC_{W, \mathbf{b}}(\mathbf{x}) = \mathrm{ReLU}(W \cdot \mathbf{x} + \mathbf{b})$$



Takes as input a vector $\mathbf{x} \in \mathbf{R}^m$ and returns a vector $\mathbf{y} \in \mathbf{R}^n$

It's parameterized by a matrix $W \in \mathbf{R}^{n \times m}$ because everyone of the *n* neurons of the layer has a vector of *m* weights, one for every element of the input vector **x**

It's parameterized by a vector $\mathbf{b} \in \mathbf{R}^n$ because every neuron has a bias

REPRESENTING N. N. AS A CAT FUNCTION

CONVOLUTIONAL LAYER (CL)

CONV_F: $\mathbf{R}^{m\mathrm{x}n\mathrm{x}r} \rightarrow \mathbf{R}^{(m-p+1)\ \mathrm{x}\ (n-q+1)\ \mathrm{x}\ t}$

$p,q \in \mathbf{N}$

$p <= m$

$q <= n$

$(F_1^{p,q}, \ldots, F_t^{p,q})$

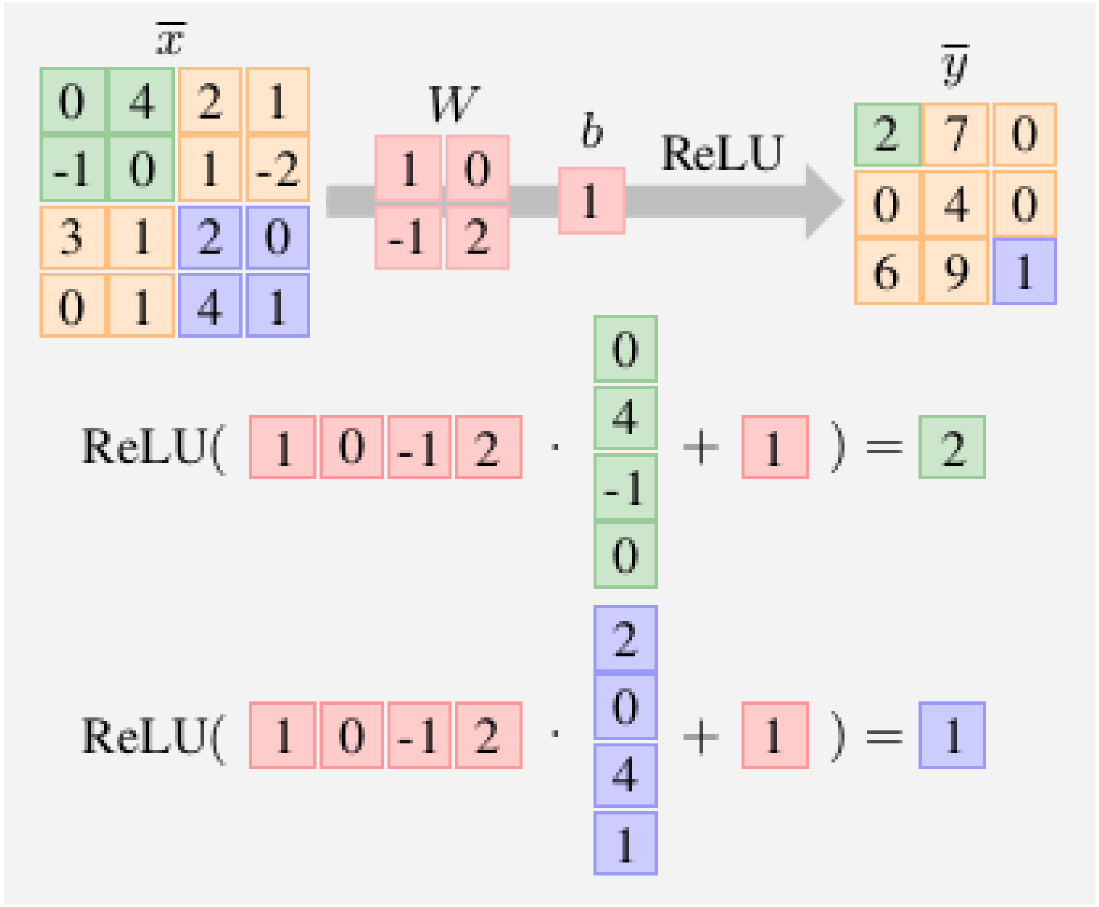
$F_i^{p,q}$: $\mathbf{R}^{m\mathrm{x}n\mathrm{x}r} \rightarrow \mathbf{R}^{(m-p+1)\ \mathrm{x}\ (n-q+1)}$

$b \in \mathbf{R}$, $W \in \mathbf{R}^{p\mathrm{x}q\mathrm{x}r}$

$$y_{i,j} = \mathrm{ReLU}(\sum_{i'=1}^{p} \sum_{j'=1}^{q} \sum_{k'=1}^{r} W_{i',j',k'} \cdot x_{(i+i'-1),(j+j'-1),k'} + b).$$

REPRESENTING N. N. AS A CAT FUNCTION

CONVOLUTIONAL LAYER: EXAMPLE



REPRESENTING N. N. AS A CAT FUNCTION

CONVOLUTIONAL LAYER TO CAT

Can we see a Convolutional Layer as a CAT function?

Yes, we reshape modify the various matrixes to obtain a function in the form:

$$\mathrm{CONV}_{W^F, \mathbf{b}}: \mathbf{R}^{m \cdot n \cdot r} \rightarrow R^{(m-p+1) \cdot (n-q+1) \cdot t}$$

We do this by creating a matrix W^F and a vector **b** of bias that simulate the sliding of the filter

REPRESENTING N. N. AS A CAT FUNCTION

CONVOLUTIONAL LAYER TO CAT: EXAMPLE

$\mathbf{x}^{v}$ = [0, 4, 2, 1, -1, 0, 1, -2, 3, 1, 2, 0, 0, 1, 4, 1]

$$W^F = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

$$\bar{b}^F = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

REPRESENTING N. N. AS A CAT FUNCTION

CONVOLUTIONAL LAYER TO CAT: EXAMPLE

$\mathrm{ReLU}(W^F \cdot \mathbf{x}^v + \mathbf{b}) \quad =$

$[2, 7, 0, 0, 4, 0, 6, 9, 1] \quad =$

$\mathbf{y}^v$

REPRESENTING N. N. AS A CAT FUNCTION

MAX POOLING (MP) LAYER

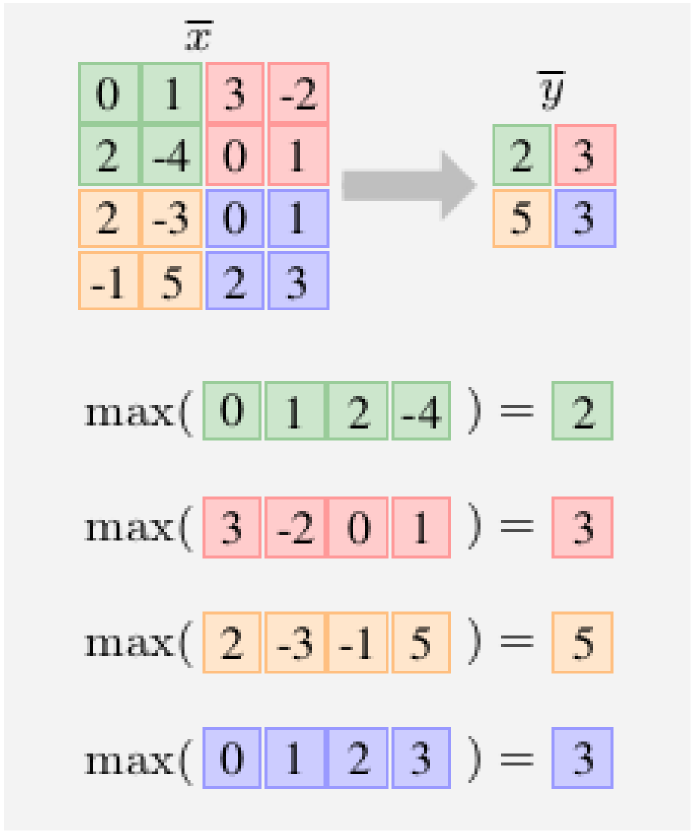
$$\text{MaxPool}_{p,q}: \mathbf{R}^{m \times n \times r} \rightarrow \mathbf{R}^{m/p \times n/q \times r}$$

$$y_{i,j,k} = \max(\{x_{i',j',k} \mid p(i-1) < i' <= p \cdot i,$$
$$q(j-1) < j' <= q \cdot j\})$$

Neurons take disjoint sub-rectangles within the input matrix and return their max value: the original height is then decremented of a factor *p* and the original width is decremented by a factor *q*

REPRESENTING N. N. AS A CAT FUNCTION

MAX POOLING (MP) LAYER: EXAMPLE



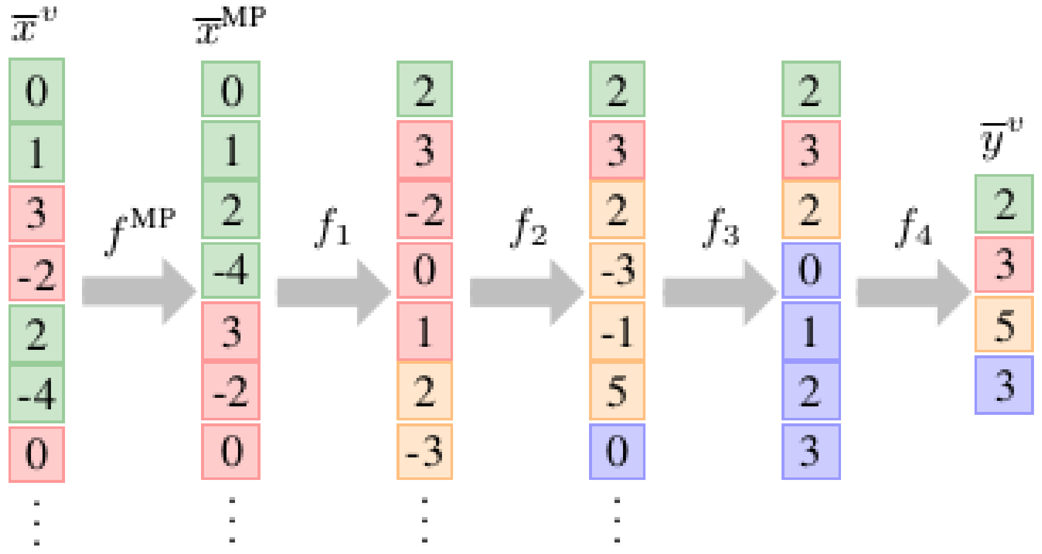
(c) Max pooling layer $\text{MaxPool}_{2,2}$

REPRESENTING N. N. AS A CAT FUNCTION

MAX POOLING (MP) LAYER AS CAT FUNCTION

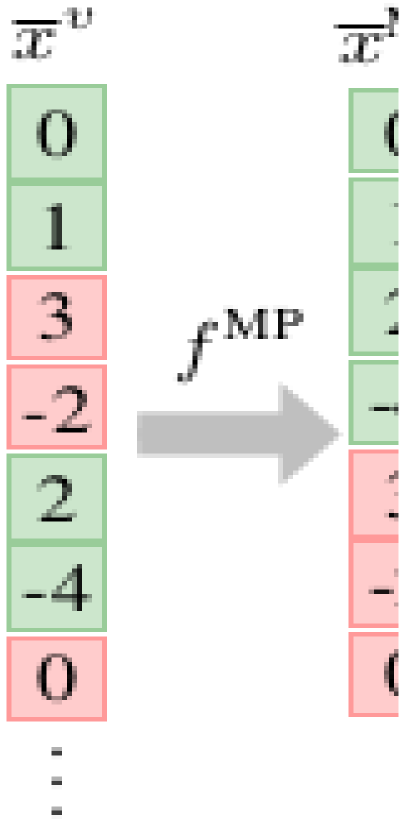
$$\text{MaxPool'}_{p,q}: \mathbf{R}^{m \cdot n \cdot r} \rightarrow \mathbf{R}^{m/p \cdot n/q \cdot r}$$

$$\text{MaxPool'}_{p,q} = f_{m/p \cdot n/q \cdot r} \circ \ldots \circ f_1 \circ f^{MP}$$



REPRESENTING N. N. AS A CAT FUNCTION

MP LAYER AS CAT: DEFINITION AND EXAMPLE



$$f^{MP} = W^{MP} \cdot \mathbf{x^v}$$

16 X 16

$$W^{\mathrm{MP}} = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{pmatrix}$$

REPRESENTING N. N. AS A CAT FUNCTION

MP LAYER AS CAT: DEFINITION AND EXAMPLE

Every f_i handles the index interval [i, i + p·q - 1] (a sub-rectangle). Let *k* be the index associated with the maximum value within the sub-rectangle, then we have

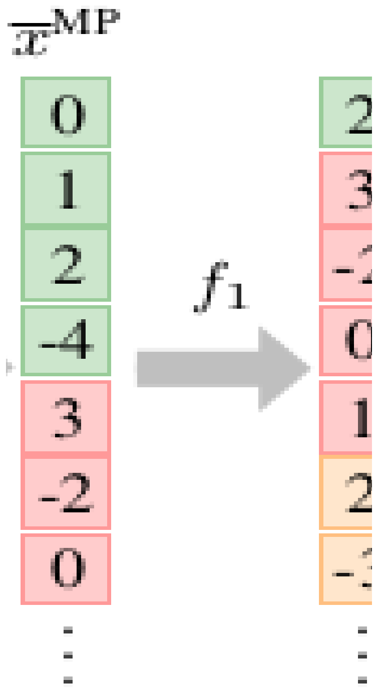
$$f_i = W_{i,k} \cdot \mathbf{x}$$

Where $W^{i,k}$ is given by the basis vectors

$$\mathbf{e}_1, \ldots, \mathbf{e}_{i-1}, \mathbf{e}_k, \mathbf{e}_{i+p \cdot q}, \ldots, \mathbf{e}_{m \cdot n - (p \cdot q - 1) \cdot (i-1)}$$

REPRESENTING N. N. AS A CAT FUNCTION

MP LAYER AS CAT: DEFINITION AND EXAMPLE



$$W^{(1,3)} = \begin{pmatrix}
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{pmatrix}$$

REPRESENTING N. N. AS A CAT FUNCTION

MP LAYER AS CAT: DEFINITION AND EXAMPLE

In general *k* , that is the index of the max value within the sub-rectangle, is not known in advance, so we have to consider p·q possible cases

For example we have

$$f_1(\overline{x}) = \begin{array}{l} \textbf{case}\ (x_1 \geq x_2) \wedge (x_1 \geq x_3) \wedge (x_1 \geq x_4):\ W^{(1,1)} \cdot \overline{x}, \\ \textbf{case}\ (x_2 \geq x_1) \wedge (x_2 \geq x_3) \wedge (x_2 \geq x_4):\ W^{(1,2)} \cdot \overline{x}, \\ \textbf{case}\ (x_3 \geq x_1) \wedge (x_3 \geq x_2) \wedge (x_3 \geq x_4):\ W^{(1,3)} \cdot \overline{x}, \\ \textbf{case}\ (x_4 \geq x_1) \wedge (x_4 \geq x_2) \wedge (x_4 \geq x_3):\ W^{(1,4)} \cdot \overline{x}. \end{array}$$

REPRESENTING N. N. AS A CAT FUNCTION

FULLY CONNECTED FEEDFORWARD ARCHITECTURE (FNN)

FC LAYER + FC LAYER + FC LAYER + …

REPRESENTING N. N. AS A CAT FUNCTION

CONVOLUTIONAL ARCHITECTURE (CNN)

CONVOLUTIONAL LAYER + FC LAYER + MP LAYER + ...

ABSTRACT INTERPRETATION

a) Find suitable abstract domain(s)

b) Define abstract operators that are *sound* and as *precise* as possible

ABSTRACT INTERPRETATION

BOX DOMAIN

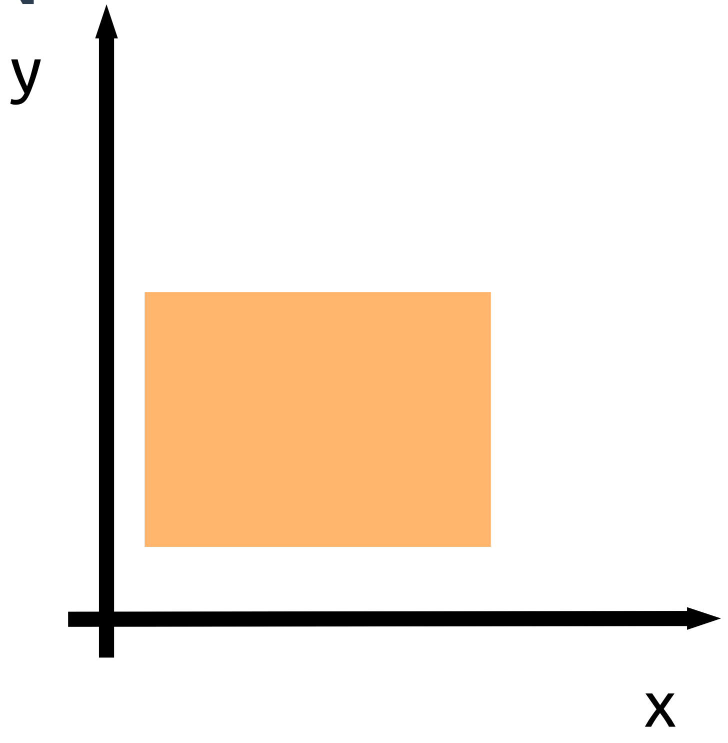
Set of constraints of the form

$$a <= x_i <= b$$

$a <= b$

$a,b \in \mathbf{R} \cup \{-\infty, +\infty\}$

Fast but quite imprecise



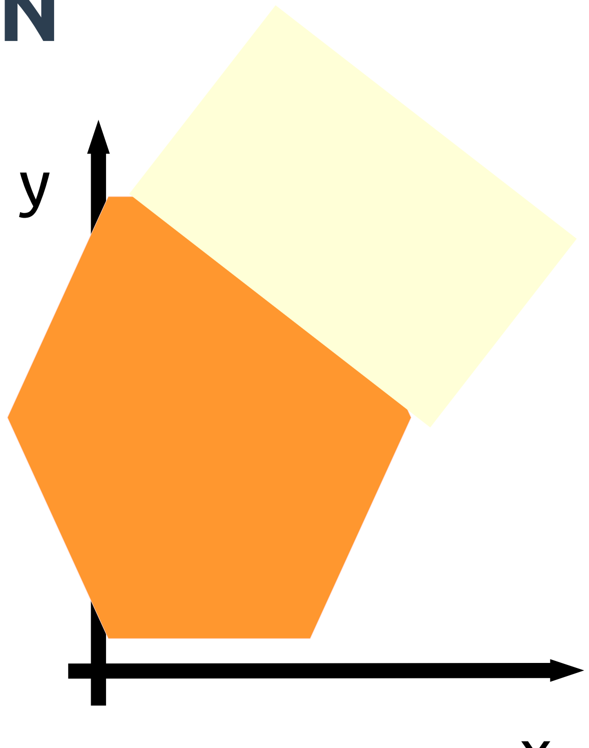
ABSTRACT INTERPRETATION

POLYHEDRA DOMAIN

Set of linear constraints of the form

$$A \cdot \mathbf{x} <= \mathbf{b}$$

For some matrix A and some vector **b**



Precise but computational expensive

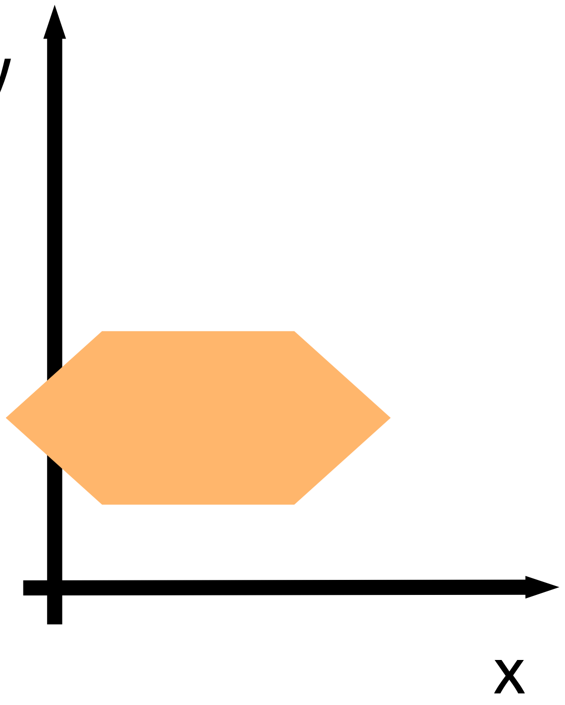
ABSTRACT INTERPRETATION

ZONOTOPES DOMAIN

Center-symmetric, convex, closed polyhedron

$$z: [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_m, b_m] \rightarrow \mathbf{R}^n$$

$$z(\mathbf{e}) = M \cdot \mathbf{e} + \mathbf{b}$$



$e_i \in [a_i, b_i]$

b: bias vector, captures the center of the zonotope

M: matrix that captures the boundaries of the zonotope around the center

ABSTRACT INTERPRETATION

ABSTRACTING CAT FUNCTIONS

$f = f'' \circ f'$

$T_f = T_{f''} \circ T_{f'}$ concrete transformer (collecting version)

$T\#_f = T\#_{f''} \circ T\#_{f'}$ abstract transformer (abstract operator)

ABSTRACT INTERPRETATION

ABSTRACTING CAT FUNCTIONS: COMPOSITION

$a \in A$

$$f(\mathbf{x}) = f_2(f_1(\mathbf{x}))$$

$$T\#_f(a) = T\#_{f2}(T\#_{f1}(a))$$

ABSTRACT INTERPRETATION

ABSTRACTING CAT FUNCTIONS: EXAMPLE

$$f(x) = \mathrm{ReLU2}(\mathrm{ReLU1}(\underbrace{\begin{bmatrix} 2 & -1 \\ 0 & 1 \end{bmatrix}}_{W} \cdot x))$$

$\mathrm{ReLU}_i = \textit{case}\ (x_i >= 0)\text{: } \mathbf{x},$

$\qquad \textit{case}\ (x_i < 0)\text{: } I_{i \leftarrow 0} \cdot \mathbf{x}$

ABSTRACT INTERPRETATION

ABSTRACTING CAT FUNCTIONS: AFFINE TRANSFORMATION

$a \in A$

$$f(\mathbf{x}) = W \cdot \mathbf{x} + \mathbf{b}$$

$$T^{\#}_{f}(a) = \mathrm{Aff}(a, W, \mathbf{b})$$

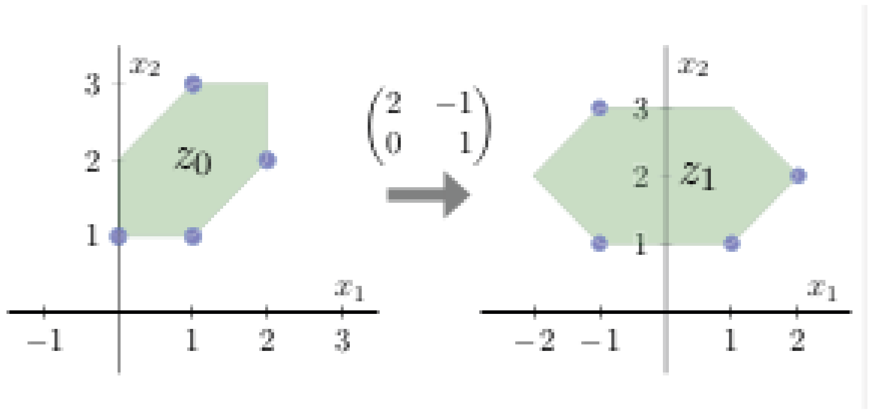
Our abstract domain *must* support affine transformations (Aff operator)

ABSTRACT INTERPRETATION

ABSTRACTING CAT FUNCTIONS: EXAMPLE

$z_0 : [-1, 1]^3 \rightarrow \mathbf{R}^2$

$z_0 (e_1, e_2, e_3) = (1 + e_1/2 + e_2/2, 2 + e_1/2 + e_3/2)$



$\text{Aff}(z_0, W, [0\ 0]^T) =$

$(2 \cdot (1 + e_1/2 + e_2/2) - (2 + e_1/2 + e_3/2), 2 + e_1/2 + e_3/2) =$

$(e_1/2 + e_2 - e_3/2, 2 + e_1/2 + e_3/2)$

ABSTRACTING CAT FUNCTIONS: CASE FUNCTIONS

To abstract case functions our abstract domain must support:

a) *meet operator* $\sqcap$: it's the abstraction of the set intersection. It must be true that:

$$\gamma^n(a) \cap \{\mathbf{x} \in \mathbf{R}^n \mid \mathbf{x} \models E\} \subseteq \gamma^n(\sqcap E)$$

Where:

$a \in A$
E is an inequality expression

γ^n is the concretization function for vectors $\in \mathbf{R}^n$

ABSTRACT INTERPRETATION

ABSTRACTING CAT FUNCTIONS: CASE FUNCTIONS

b) *join operator* $\sqcup$: it's the abstraction of the set union. It must be true that:

$$\gamma^n(a_1) \cup \gamma^n(a_2) \subseteq \gamma^n(a_1 \sqcup a_2)$$

Where:

$a_1, a_2 \in A$

γ^n is the concretization function for vectors $\in \mathbf{R}^n$

ABSTRACT INTERPRETATION

ABSTRACTING CAT FUNCTIONS: CASE FUNCTIONS

c) *bottom element* $\perp$: satisfies:

$\gamma^n(\perp) = \{\}$

$\perp \sqcap E = \perp$

$\perp \sqcup a = a$

$a \in A$

E is an inequality expression

γ^n is the concretization function for vectors $\in \mathbf{R}^n$

ABSTRACT INTERPRETATION

ABSTRACTING CAT FUNCTIONS: CASE FUNCTIONS

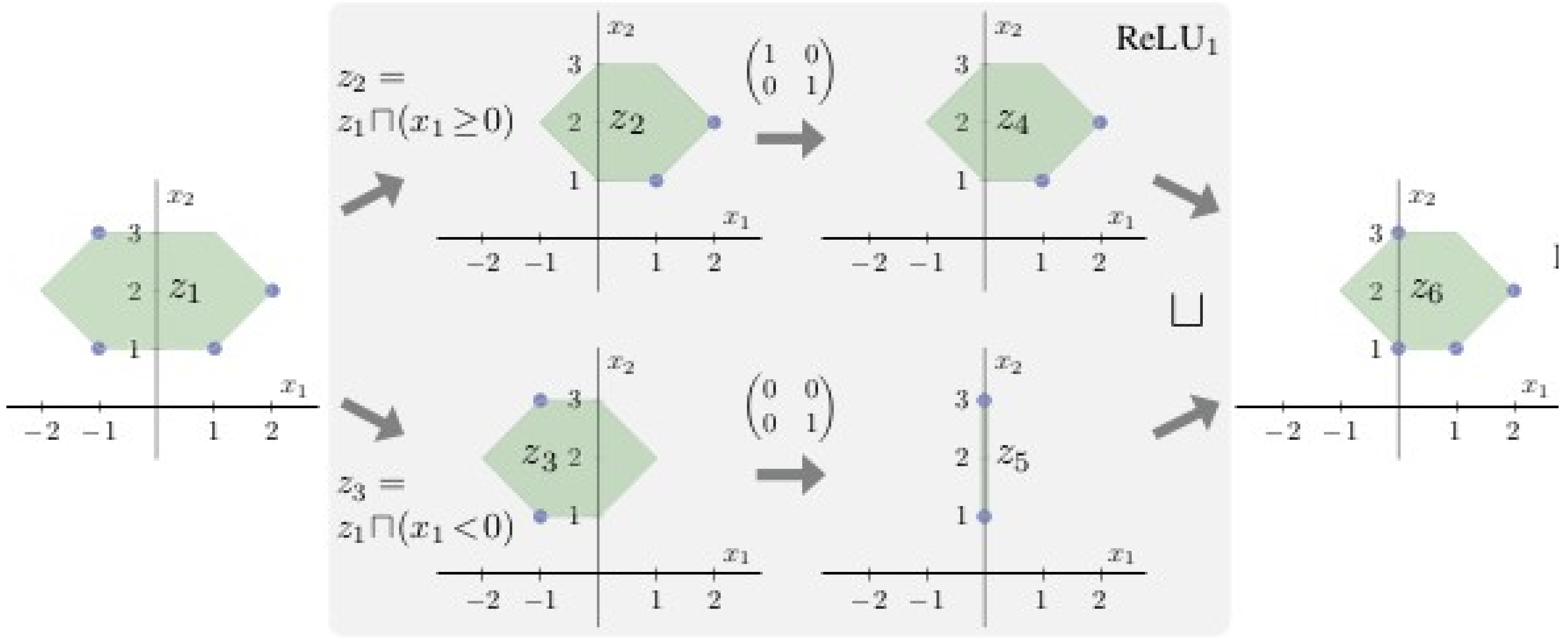
$a \in A$
E is an inequality expression

$$f(\mathbf{x}) = \textit{case}\ E_1: f_1(\mathbf{x}), \ldots, \textit{case}\ E_k: f_k(\mathbf{x})$$

$$T^{\#}_f(a) = \bigsqcup_{1 <= i <= k} f_i^{\#}(a \sqcap E_i)$$

ABSTRACT INTERPRETATION

ABSTRACTING CAT FUNCTIONS: EXAMPLE



$z_2 = z_1 \sqcap (x_1 \geq 0)$
$z_3 = z_1 \sqcap (x_1 < 0)$
$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$
ReLU$_1$
$\sqcup$
z_1
z_2
z_3
z_4
z_5
z_6
x_1
x_2

ABSTRACT INTERPRETATION

SOUNDNESS

It can be proved that:

For any CAT function with concrete transformer T_f: $\mathcal{P}(\mathbf{R}^m) \rightarrow \mathcal{P}(\mathbf{R}^n)$ and for any abstract input $a \in A$

$$T_f\,(\gamma^m(a)) \subseteq \gamma^n\,(T^{\#}_f\,(a)\,)$$

ABSTRACT INTERPRETATION

ROBUSTNESS PROPERTIES

$$N: \mathbf{R}^m \rightarrow \mathbf{R}^n$$

$$(X, C) \in \mathcal{P}(\mathbf{R}^m) \times \mathcal{P}(\mathbf{R}^n)$$

X – robustness region
C – robustness condition

N satisfies robustness property (X,C) if

for all $\mathbf{x} \in X$ we have $N(\mathbf{x}) \in C$

NOTE: Abstract Interpretation is here used to prove robustness properties. However, the framework is completely general and can be used to prove any type of property

ABSTRACT INTERPRETATION

LOCAL ROBUSTNESS

(X, C_L)

$$C_L = \left\{ \overline{y} \in \mathbb{R}^n \;\middle|\; \underset{i \in \{1,\ldots,n\}}{\arg\max}\,(y_i) = L \right\}$$

X – robustness region

C_L – robustness condition, contains all the outputs with the same label

ABSTRACT INTERPRETATION

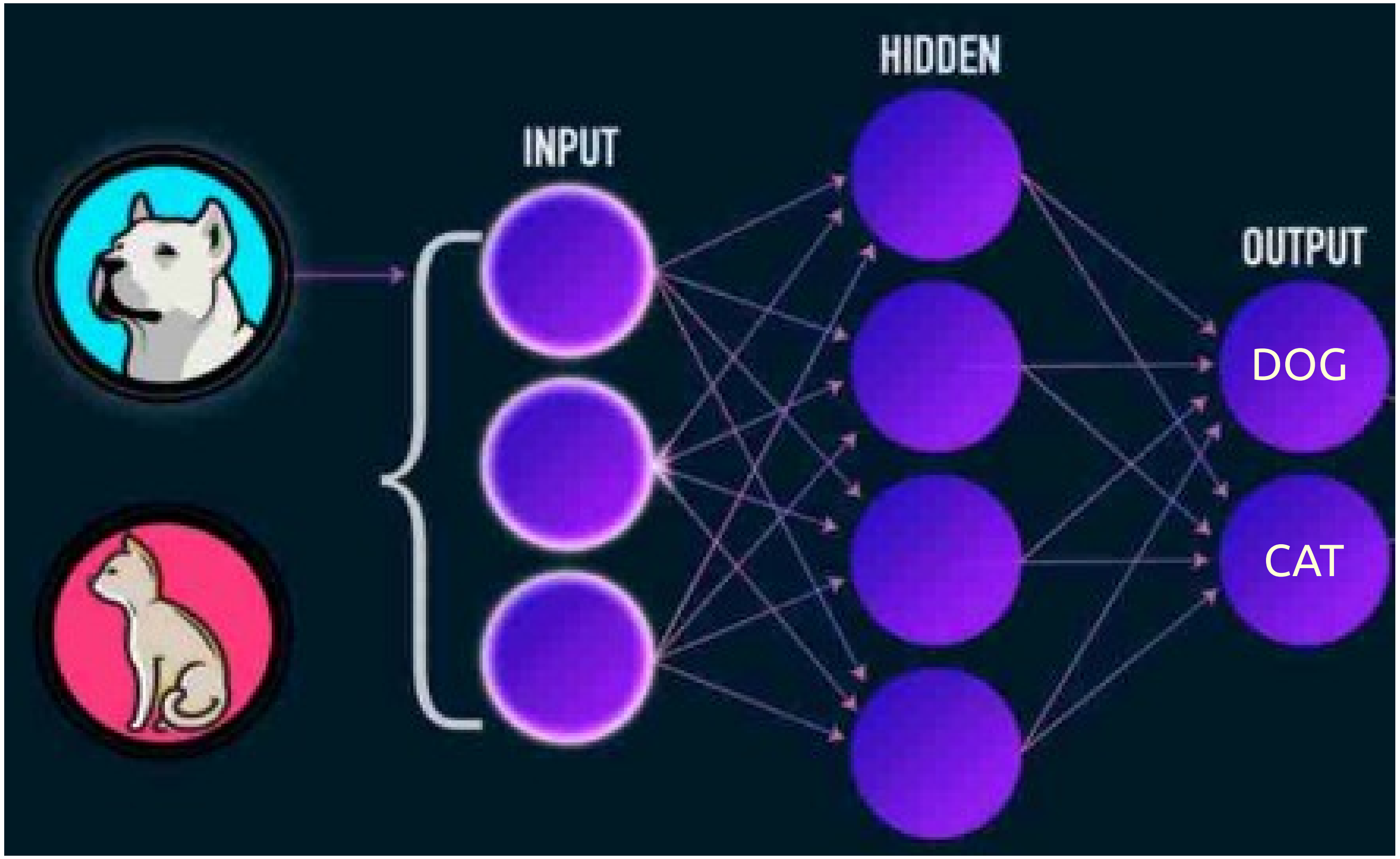
LOCAL ROBUSTNESS

Typically one wants to find if there is some label L for which

(X,C_L) holds

ABSTRACT INTERPRETATION

LOCAL ROBUSTNESS: EXAMPLE



0.7

0.3

ABSTRACT INTERPRETATION

LOCAL ROBUSTNESS

It’s like asking

DOG

“given this particular set of inputs X and this particular set of labels, can we find a label that represents the classification of all the inputs belonging to X?”

CAT

ABSTRACT INTERPRETATION

PROVING PROPERTIES

Let $T^{\#}_{N}$ be the abstract transformer for the neural network N. Then by the properties of α and γ and by soundness we know that to argue that a property (X, C) holds is sufficient to prove that

DOG

CAT

$$\gamma^{n}(T^{\#}_{N}(\alpha^{m}(X))) \subseteq C$$

ABSTRACT INTERPRETATION

PROVING PROPERTIES

NOTE: if C can be expressed as a CNF (Conjuctive Normal Form) there is a general method to prove that

$$\gamma^n(\underbrace{T^\#_N(\alpha^m(X))}_{a}) \subseteq C$$

DOG

CAT

ABSTRACT INTERPRETATION

PROVING PROPERTIES

1) C = $\bigwedge_i \bigvee_j l_{i,j}$

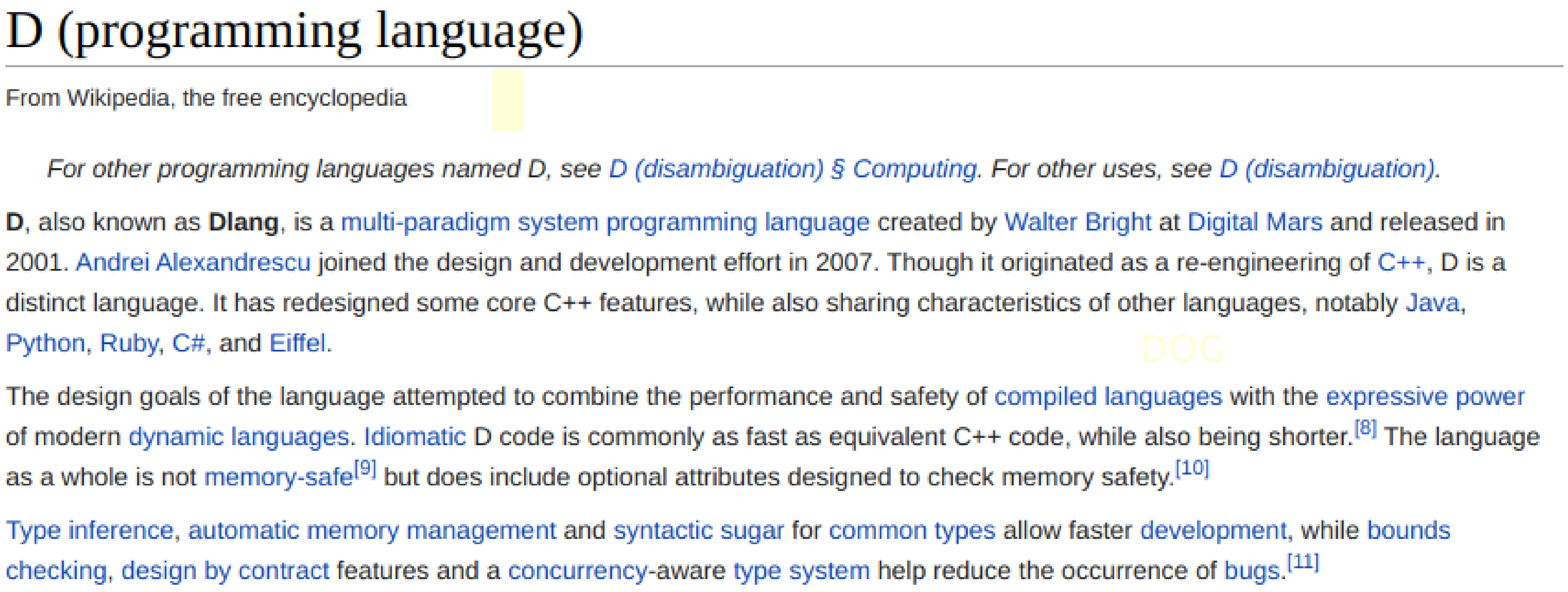
DOG

2) ¬C = $\bigvee_i \bigwedge_j \neg l_{i,j}$

CAT

3) $\gamma^n(a) \subseteq C$ <=> $a \sqcap \left(\bigwedge_j \neg l_{i,j} \right) = \bot$ for all i

IMPLEMENTATION



D (programming language)

From Wikipedia, the free encyclopedia

For other programming languages named D, see D (disambiguation) § Computing. For other uses, see D (disambiguation).

D, also known as **Dlang**, is a multi-paradigm system programming language created by Walter Bright at Digital Mars and released in 2001. Andrei Alexandrescu joined the design and development effort in 2007. Though it originated as a re-engineering of C++, D is a distinct language. It has redesigned some core C++ features, while also sharing characteristics of other languages, notably Java, Python, Ruby, C#, and Eiffel.

DOG

The design goals of the language attempted to combine the performance and safety of compiled languages with the expressive power of modern dynamic languages. Idiomatic D code is commonly as fast as equivalent C++ code, while also being shorter.[8] The language as a whole is not memory-safe[9] but does include optional attributes designed to check memory safety.[10]

Type inference, automatic memory management and syntactic sugar for common types allow faster development, while bounds checking, design by contract features and a concurrency-aware type system help reduce the occurrence of bugs.[11]

IMPLEMENTATION

Supports:

- Convolutional layers

DOG

- Max Pooling layers
- Fully Connected layers

IMPLEMENTATION

PROPERTIES

(X, C)

DOG

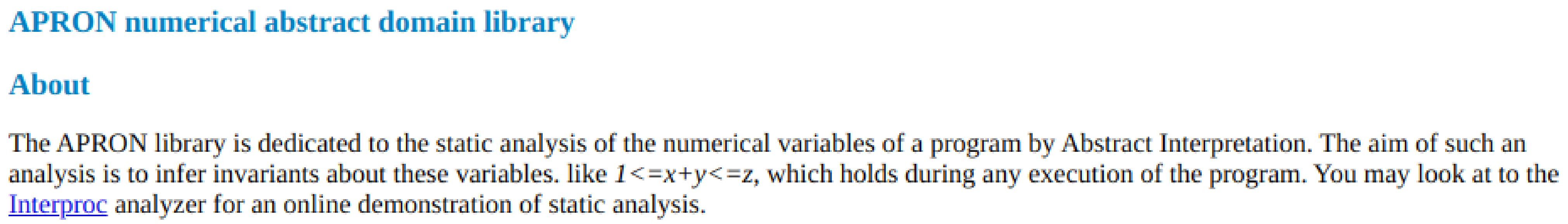
X - robustness region, specified by a zonotope

C - conjunction of linear constraints

Typically X is a box or a line, precisely captured by a zonotope

IMPLEMENTATION

ABSTRACT DOMAINS



APRON numerical abstract domain library

About

The APRON library is dedicated to the static analysis of the numerical variables of a program by Abstract Interpretation. The aim of such an analysis is to infer invariants about these variables. like *1<=x+y<=z*, which holds during any execution of the program. You may look at to the Interproc analyzer for an online demonstration of static analysis.

- Box
- Polyhedra
- Zonotope

http://apron.cri.ensmp.fr/library/

IMPLEMENTATION

BOUNDED POWERSET DOMAIN

P(**A**), where **A** is an abstract domain

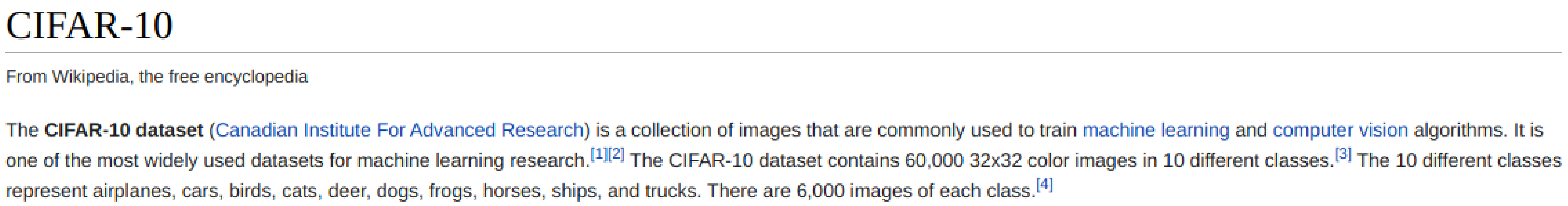
DOG

An element of the bounded powerset domain is a set of max *N* elements of the abstract domain **A**, for some constant *N*

We denote this domain by appending the value N to the abstract domain name, e.g. Zonotope64

EVALUATION

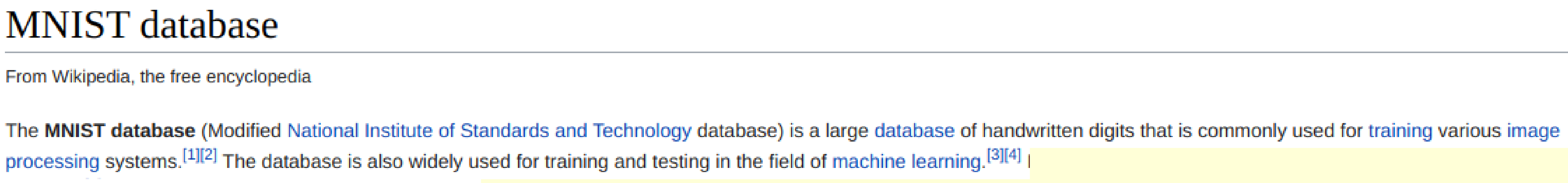
EXPERIMENTAL SETUP: DATASETS



CIFAR-10

From Wikipedia, the free encyclopedia

The **CIFAR-10 dataset** (Canadian Institute For Advanced Research) is a collection of images that are commonly used to train machine learning and computer vision algorithms. It is one of the most widely used datasets for machine learning research.[1][2] The CIFAR-10 dataset contains 60,000 32x32 color images in 10 different classes.[3] The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class.[4]



MNIST database

From Wikipedia, the free encyclopedia

The **MNIST database** (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems.[1][2] The database is also widely used for training and testing in the field of machine learning.[3][4]

EVALUATION

EXPERIMENTAL SETUP: CNN

- Trained on both datasets until test set accuracy was at least 0.9

DOG

- *LeNet* Architecture: 2 CL → 1 MP → 2 CL → 1MP → 3FC

EVALUATION

EXPERIMENTAL SETUP: FNN

- Trained on both datasets until test set accuracy was at least 0.9

- 7 different architectures:
  - 3 x 20
  - 6 x 20
  - 3 x 50
  - 3 x 100
  - 6 x 100
  - 6 x 200
  - 9 x 200

DOG

EVALUATION

EXPERIMENTAL SETUP: PROPERTIES

(X, CL)

where X captures changes in lighting

We define the robustness region as DOG

$$S_{\overline{x},\delta} = \{\overline{x}' \in \mathbb{R}^m \mid \forall i \in [1,m].\, 1-\delta \le x_i \le x'_i \le 1 \vee x'_i = x_i\}$$

$\delta \in \Delta = \{0.001, 0.005, 0.025, 0.045, 0.065, 0.085\}$

AI^2 is used to check whether all input belonging to the robustness region are classified with the same label as **x**

EVALUATION

EXPERIMENTAL SETUP: PROPERTIES

- NOTE 1: as δ increases, the robustness region gets larger

- NOTE 2: the definition of $S_{\mathbf{x},\delta}$ entails that every pixel of the image **x** can be brightened independently: this fact allows to treat cases in which only one part of the image is brightened more easily

- NOTE 3: in general, AI2 can be used to verify every property in which the robustness region can be specified by a zonotope

EVALUATION

EXPERIMENTAL SETUP

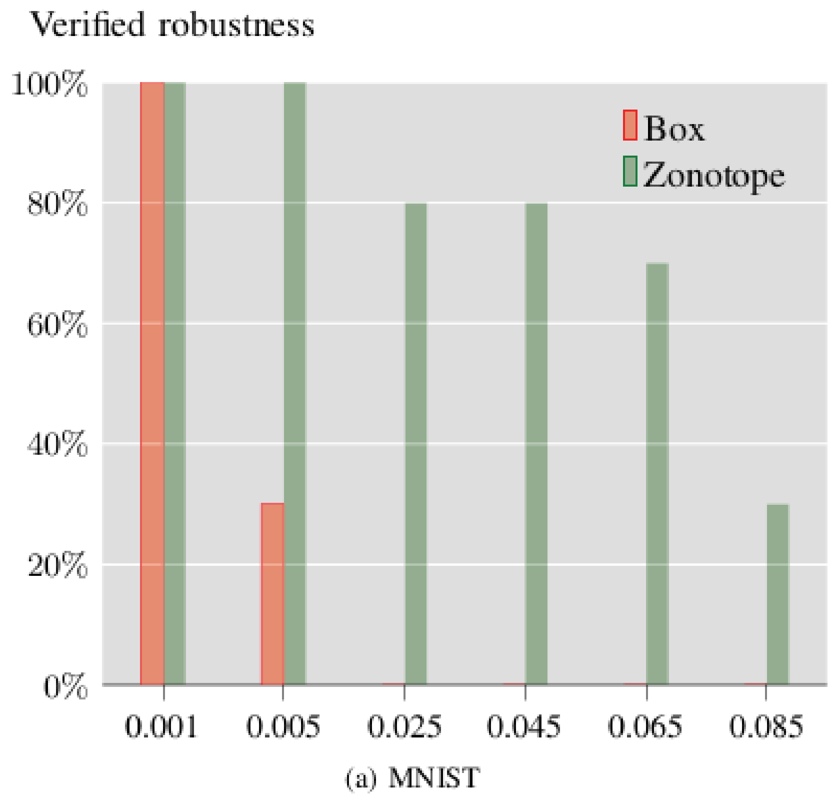
- 10 images for each dataset: thus we have one property for every couple *(image, δ)* = 60 properties per dataset

DOG

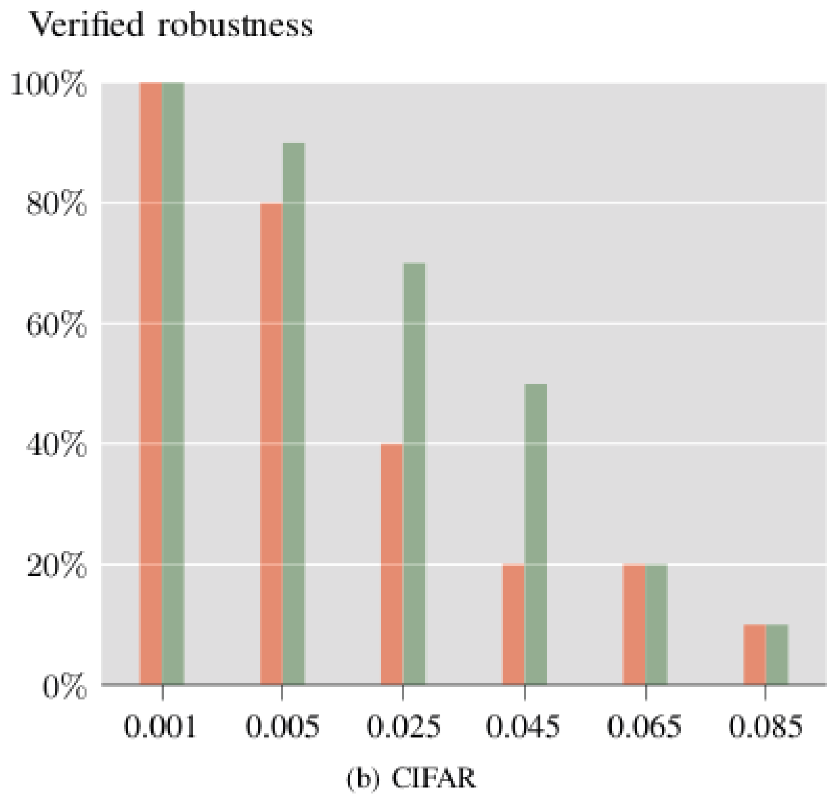
- Box, Zonotope, Zonotope*N* (with *N* between 2 and 128)
- AI^2 is compared to Reluplex on FNN
- Ubuntu 16.04.03 Server, two Intel Xeon E5, 512GB RAM

EVALUATION

ROBUSTNESS OF CNN



(a) MNIST

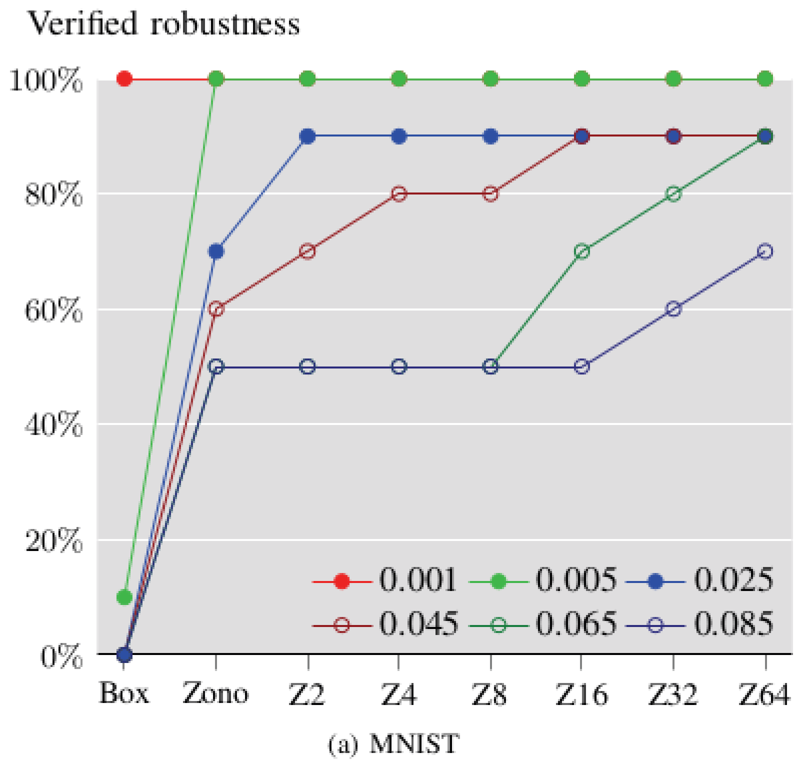


(b) CIFAR

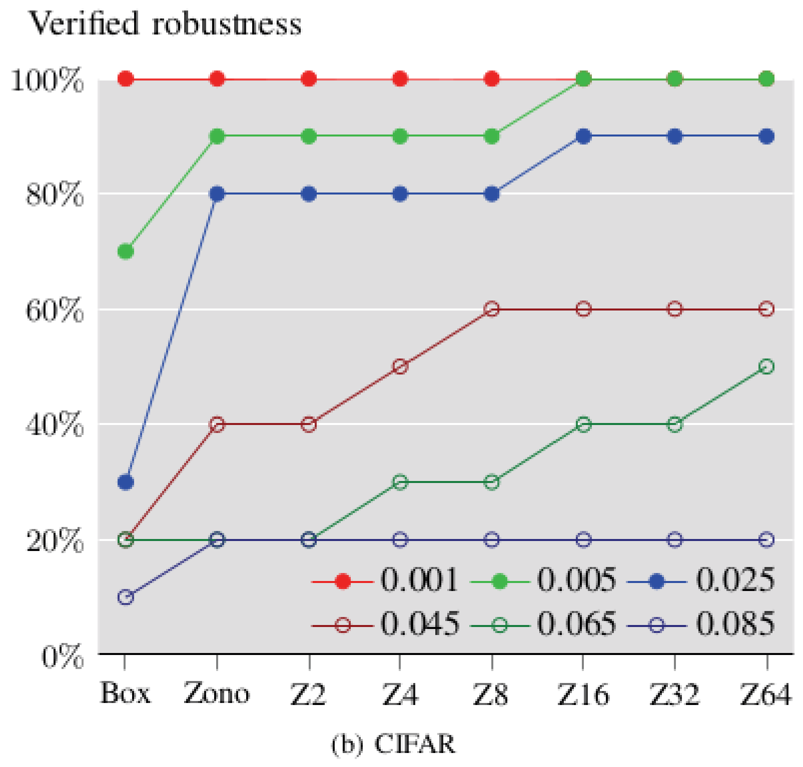
Fig. 9: Verified properties by AI^2 on the MNIST and CIFAR convolutional networks for each bound $\delta \in \Delta$ (x-axis).

EVALUATION

PRECISION OF ABSTRACT DOMAINS



(a) MNIST



(b) CIFAR

Fig. 10: Verified properties as a function of the abstract domain used by AI^2 for the 9×200 network. Each point represents the fraction of robustness properties for a given bound (as specified in the legend) verified by a given abstract domain (x-axis).

EVALUATION

AVERAGE RUNNING TIMES

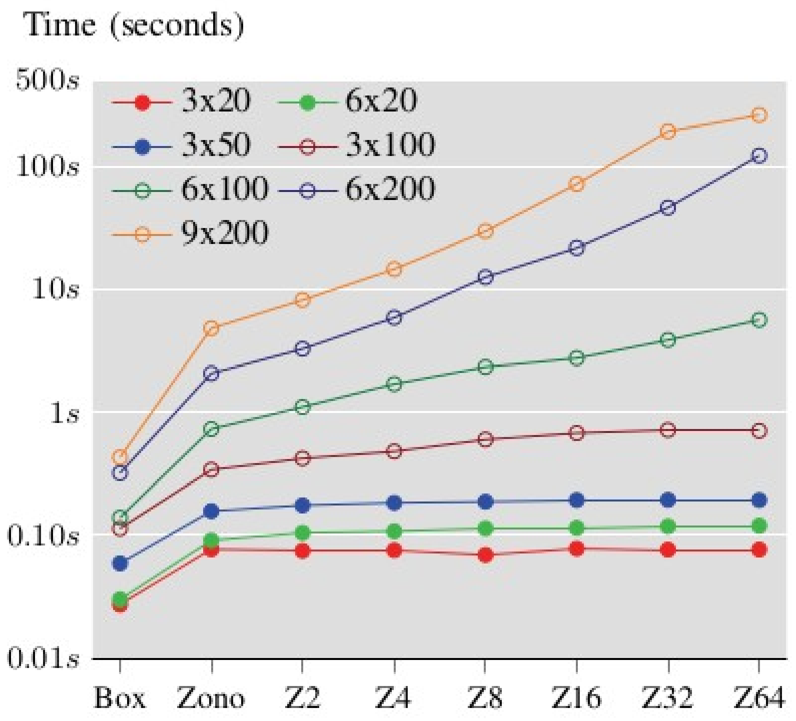
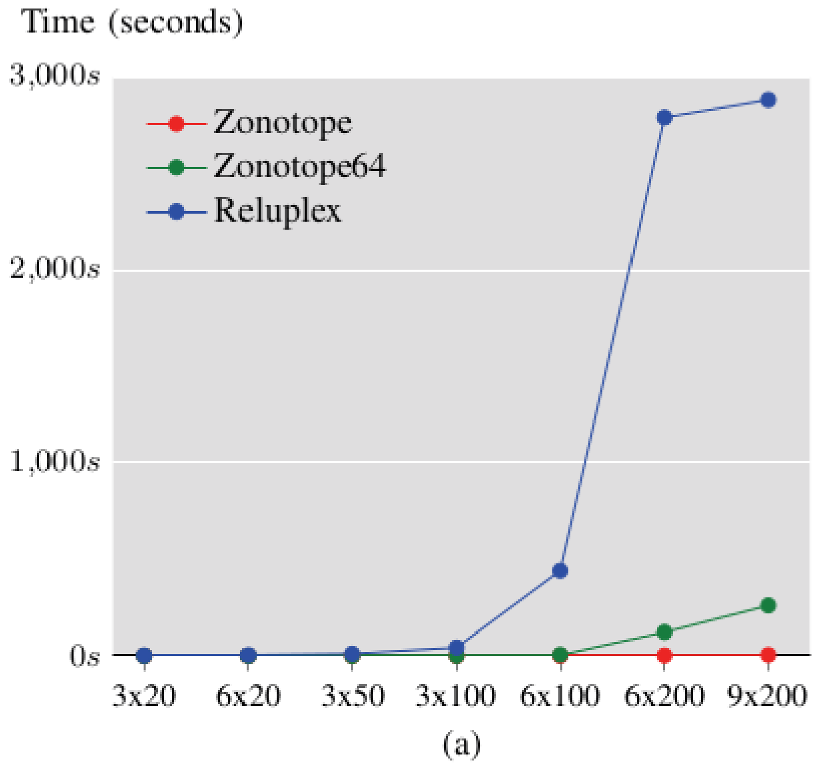


Fig. 11: Average running time of AI^2 when proving robustness properties on MNIST networks as a function of the abstract domain used by AI^2 (x-axis). Axes are scaled logarithmically.

DOG

EVALUATION

COMPARISON TO RELUPLEX



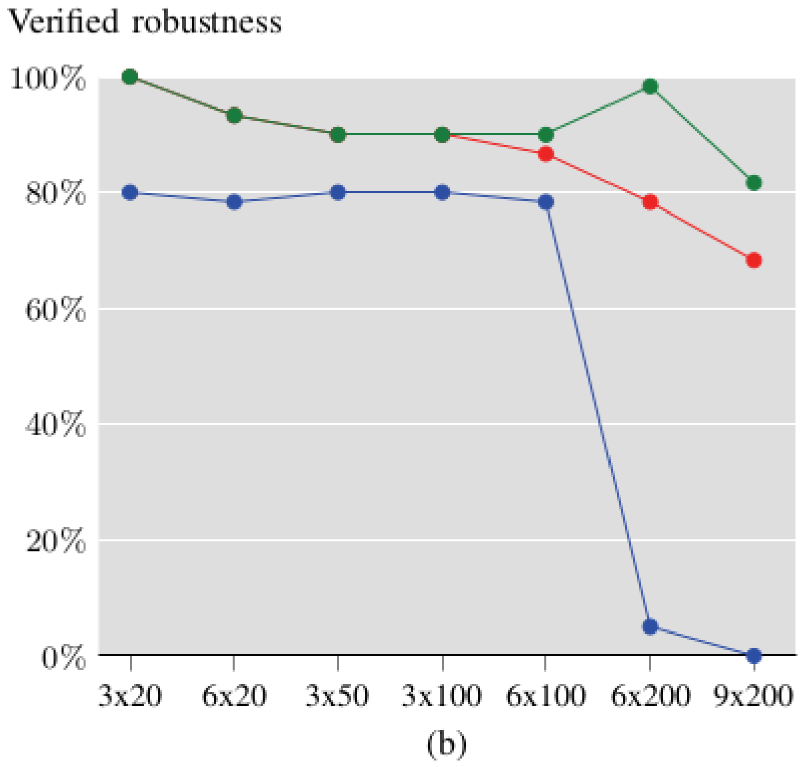


Fig. 12: Comparing the performance of AI^2 to Reluplex. Each point is an average of the results for all 60 robustness properties for the MNIST networks. Each point in (a) represents the average time to completion, regardless of the result of the computation. While not shown, the result of the computation could be a failure to verify, timeout, crash, or discovery of a counterexample. Each point in (b) represents the fraction of the 60 robustness properties that were verified.

COMPARING DEFENSES WITH AI^2

- A defense is “an algorithm whose goal is to reduce the effectiveness of a certain attack against a specific network”
- We can use AI^2 to measure the size of the robustness region
- Intuitively a greater size of the robustness region means a better defense

COMPARING DEFENSES WITH AI²

FGSM

(**F**ast **G**radient **S**ign **M**ethod)

1) Take a network *N* and an image **x**

DOG

2) Compute a vector $\boldsymbol{\rho}_{N,\,\mathbf{x}}$ in the input space along which is very probable to find an adversarial example

3) Generate an adversarial example in the following way:

$$\mathbf{a} = \mathbf{x} + \varepsilon \cdot \boldsymbol{\rho}_{N,\,\mathbf{x}}$$

for some value ε

COMPARING DEFENSES WITH AI^2

ROBUSTNESS REGION

For this experiment the robustness region (here called *Line*) is defined as

DOG

$$L_{N,\overline{x},\delta} = \{\overline{x} + \epsilon \cdot \overline{\rho}_{N,\overline{x}} \mid \epsilon \in [0,\delta]\}.$$

1) This robustness region is a zonotope
2) It captures all points from $\mathbf{x}$ to $\mathbf{x} + \delta \cdot \boldsymbol{\rho}_{N, \mathbf{x}}$ for some bound δ: a large δ implies a large robustness region

COMPARING DEFENSES WITH AI^2

DEFENSES

GSS and **Ensemble**: the common idea behind them is to add a regularization term, that encodes the FGSM attack, to the loss function;

DOG

MMSTV: during the training phase adds a perturbation layer before the input layer. This perturbation layer applies the FGSM^k attack (a multi-step variant of FGSM).

COMPARING DEFENSES WITH AI^2

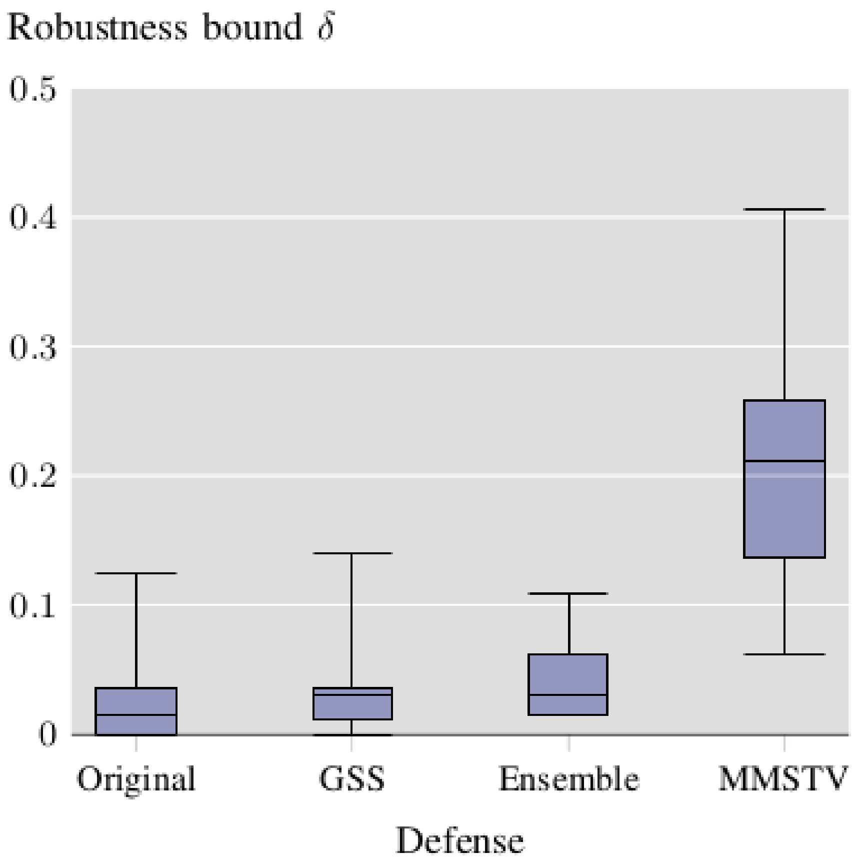


Fig. 13: Box-and-whisker plot of the verified bounds for the Original, GSS, Ensemble, and MMSTV networks. The boxes represent the δ for the middle 50% of the images, whereas the whiskers represent the minimum and maximum δ. The inner-lines are the averages.

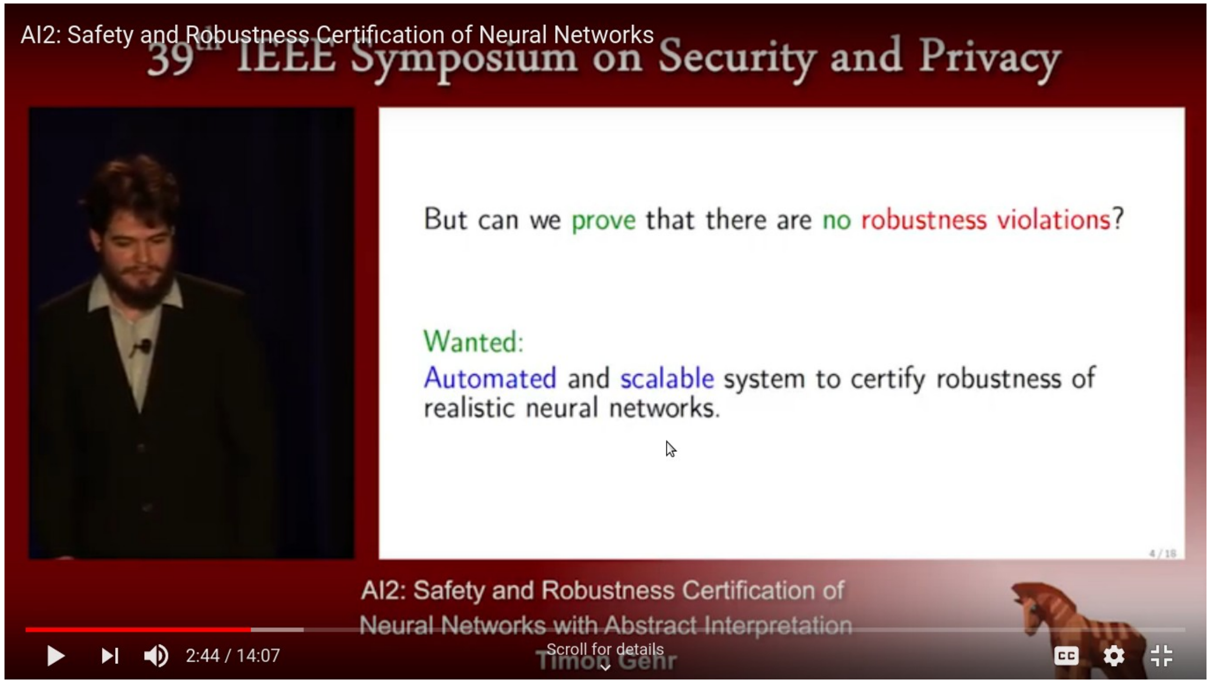
FUTURE WORK

- New abstract transformers to expand supported N. N. features

DOG

- A library to model the most common perturbations

RELATED LINKS


AI2: Safety and Robustness Certification of Neural Networks

39th IEEE Symposium on Security and Privacy

But can we prove that there are no robustness violations?

Wanted:
Automated and scalable system to certify robustness of realistic neural networks.

AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation

Timon Gehr

- https://www.youtube.com/watch?v=LJnjCMV8KzA
- https://www.sri.inf.ethz.ch/publications/gehr2018ai