Validation et certification (5/6) - Validation formelle du fonctionnement de réseaux de neurones

DE LA RECHERCHE À L'INDUSTRIE

Augustin Lemesle - 27/01/2023

Reminder

Reminder on formal methods

- **Critical systems**
  - A system whose failure may cause physical harm, economical losses or damage the environement

- **Formal methods**
  - A set of techniques that aims to prove properties on programs

Reminder on formal methods

- **Formal methods were first applied on classical software**
  - Numerous tools and methods available
  - Techniques were developed and refined over time

- **And for AI?**
  - Developments in the past few years
  - New tools and techniques
  - But lots of challenges (input specification, scalability, embedded …)

- **Why?**
  - With the growth of AI it starts to be included in industrial system and possibly critical systems
  - We must bring safety guarantees to it

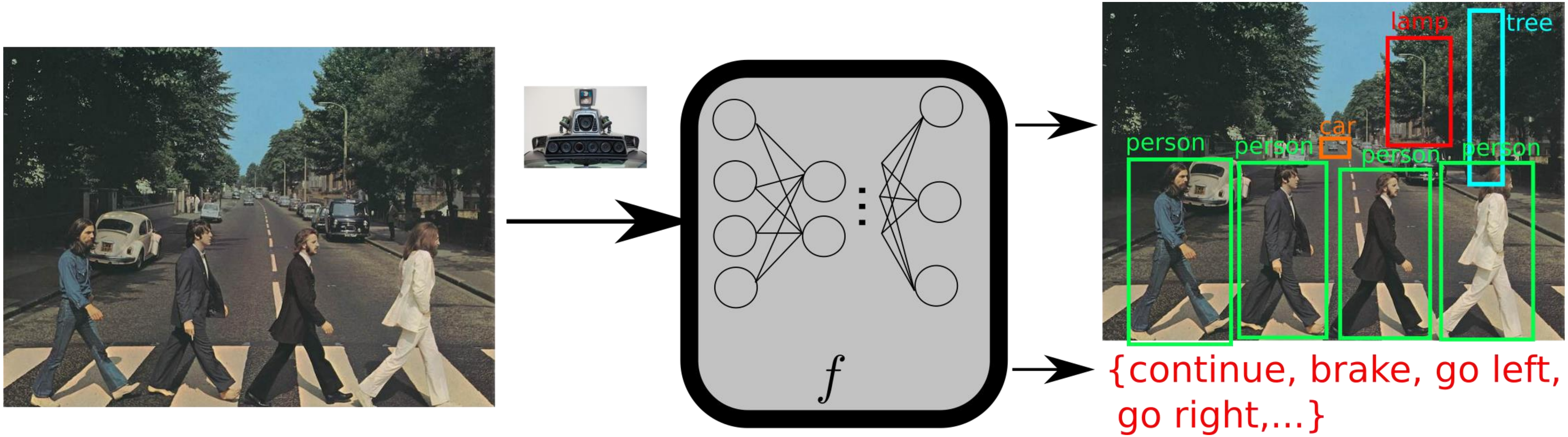
Different tools & techniques

- **Classical:**
  1. SMT Solvers: **Z3**, Alt-Ergo, Colibri, …
  2. Formal methods at large: Apron, Astrée, Frama-C, Fluctuat
  3. Testing utilities

- **For neural network**
  1. **AIMOS**: Testing framework
  2. **Marabou**: Simplex based tool
  3. **PyRAT**: Abstract interpretation tool
  4. Alpha-beta-CROWN: Dual problem based + SMT solver
  5. Nnenum
  6. …

Challenges

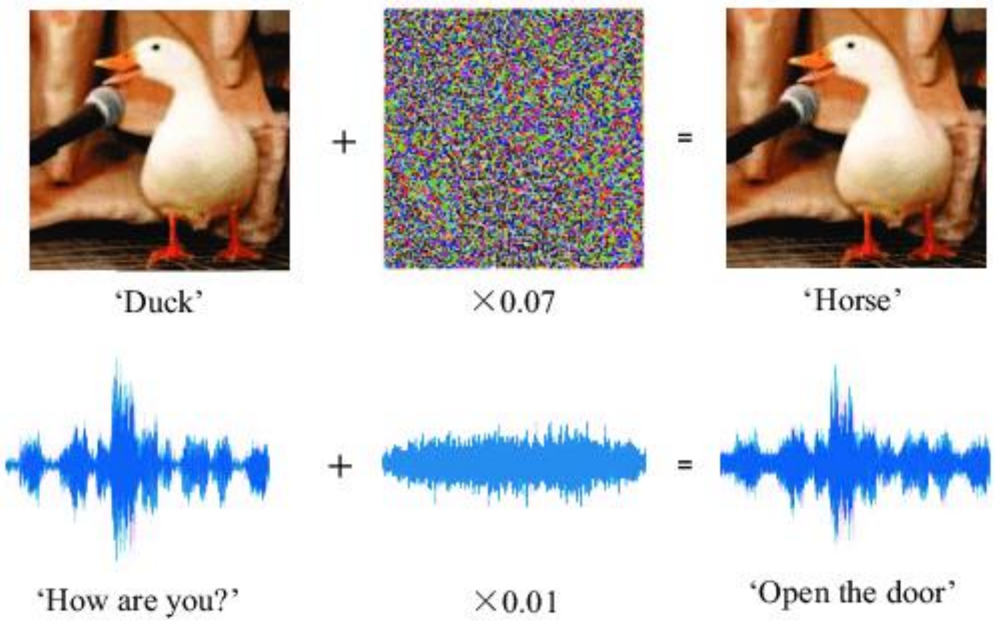
Specifying the property



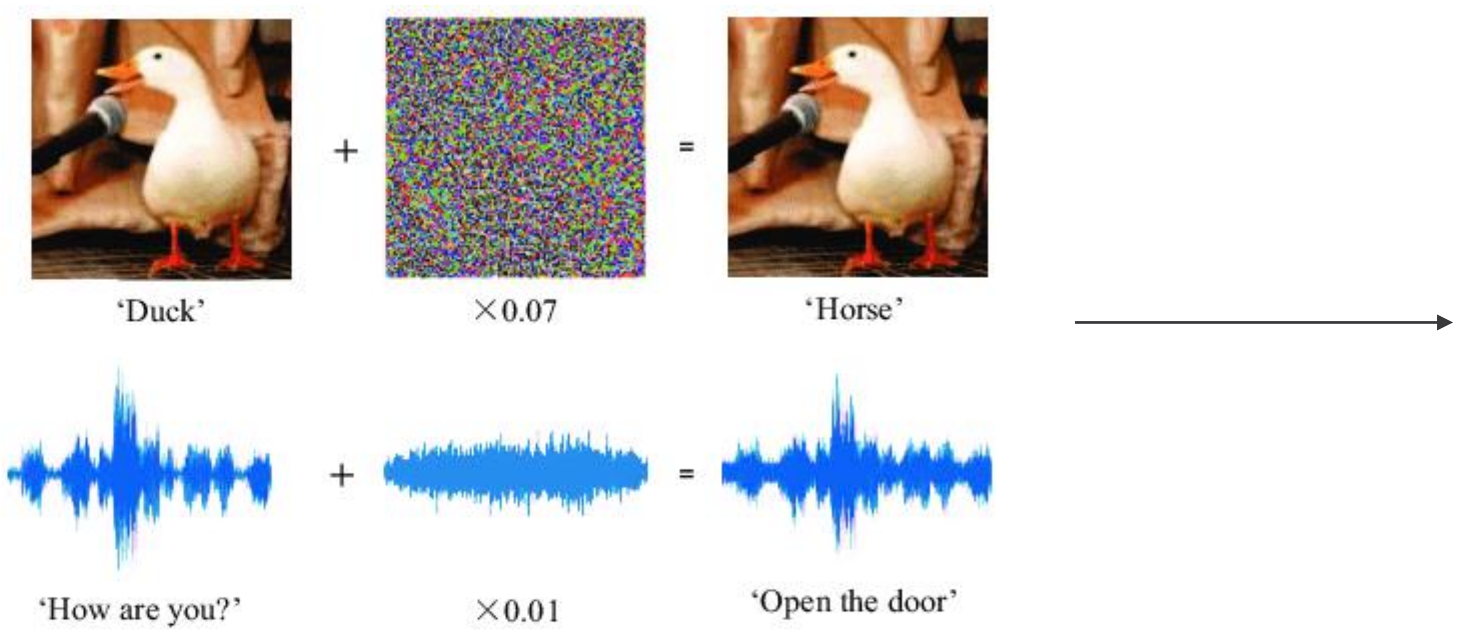
Dream property: « the autonomous car will never run over pedestrians »

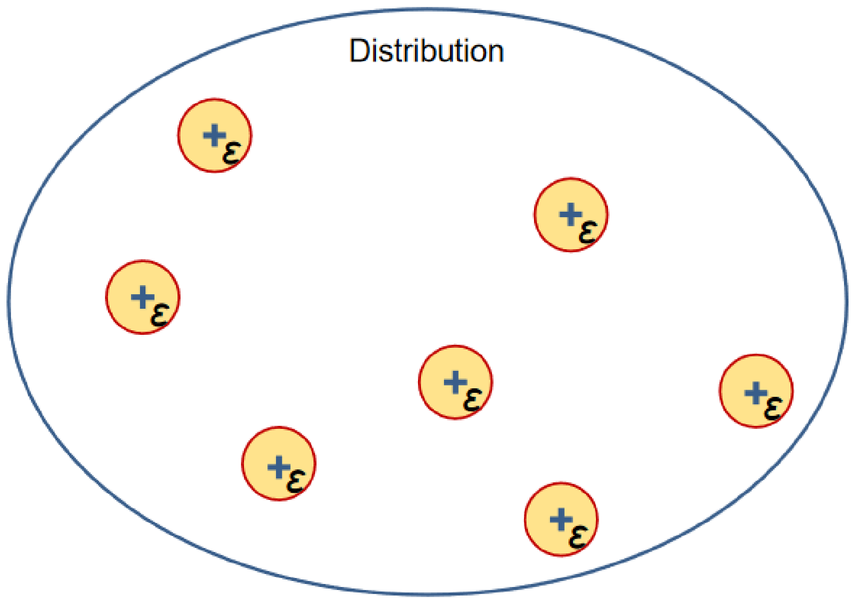
What is a pedestrian?

Adversarial attacks give one possible answer



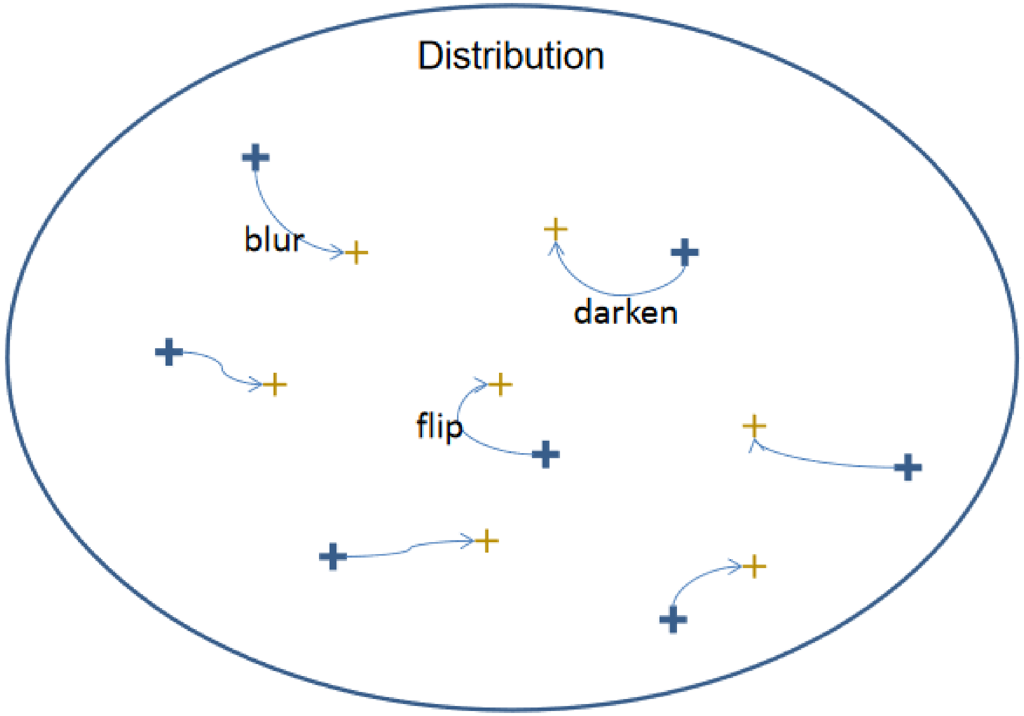
Adversarial attacks give one possible answer





Metamorphism is another

Rotating the images



But other challenges as well

- **Scalability**
- **Embedded systems**
- **…**

A simple example

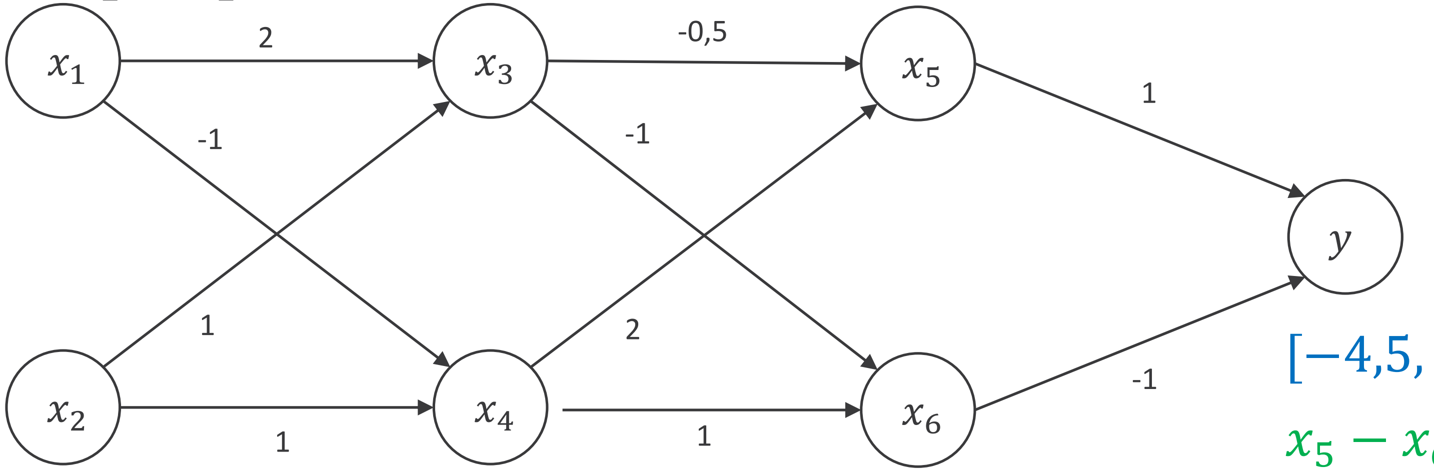
Abstract interpretation

$-0{,}5x_3 + 2\,x_4 = -x_1 \; - 0{,}5x_2 \; - 2x_1 + 2x_2$

$2x_1 + x_2 = [0, 3]$ $\quad -3x_1 + 1{,}5x_2 = [-3, 1{,}5]$

$[0, 3]$ $\quad [-3{,}5, 2]$

$x_1 \in [0, 1]$



$y > 0$?

$[-4{,}5, 6]$

$x_5 - x_6 = 1{,}5x_2$

$[0, 1{,}5]$

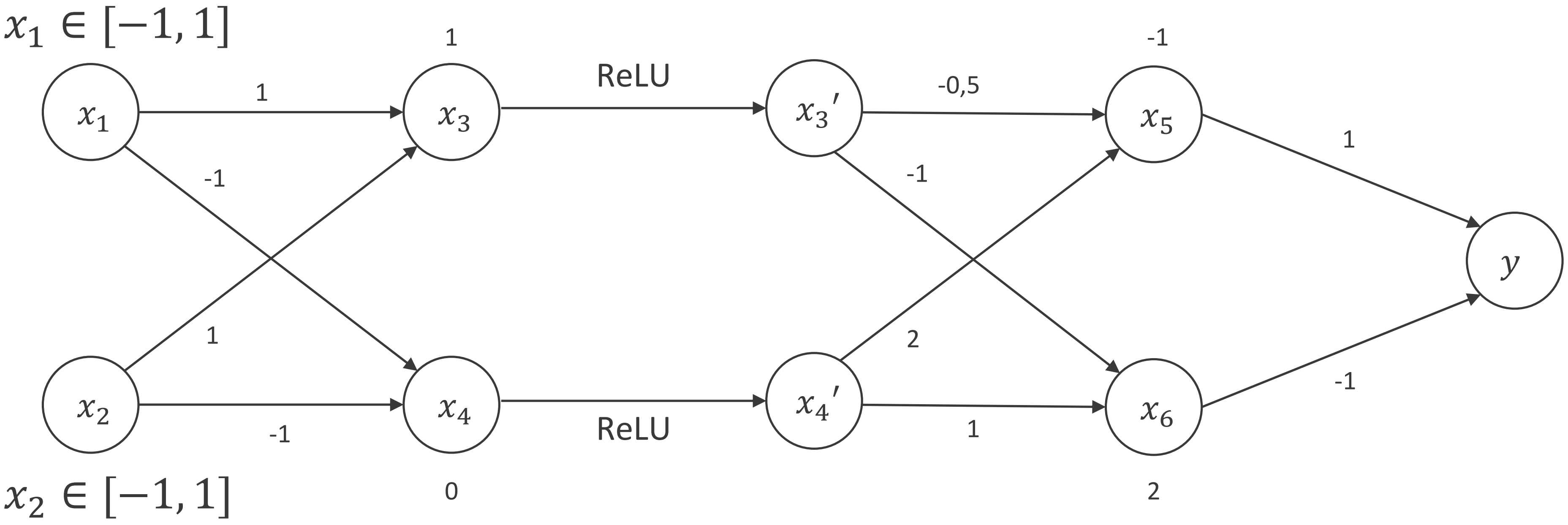
$x_2 \in [0, 1]$

$[-1, 1]$ $\quad [-4, 1]$

$-x_1 + x_2 = [-1, 1]$ $\quad -3x_1 = [-2, 0]$

$-x_3 + x_4 = -2x_1 \; - x_2 \; - x_1 + x_2$

What if ReLU?



$x_1 \in [-1, 1]$
$x_2 \in [-1, 1]$
x_1
x_2
x_3
x_4
x_3'
x_4'
x_5
x_6
y
ReLU
ReLU
1
-1
1
-1
1
0
-0,5
-1
2
1
-1
2
1
-1

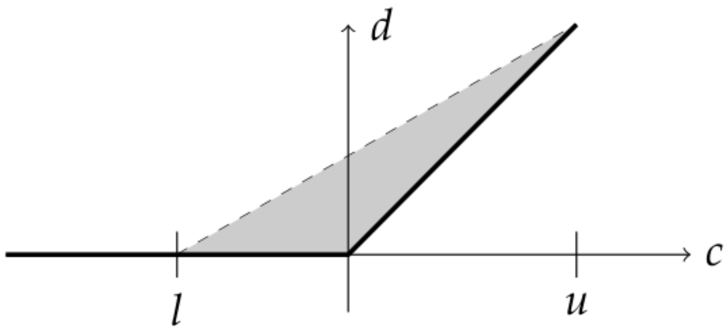
What if ReLU?

On intervals

$$relu([l, u]) = [\max(0, l), \max(0, u)]$$

Simple enough

On symbolic intervals



Abstraction

x < 0

$$relu(x) = 0$$

x >= 0

$$relu(x) = x$$

Case disjunction

Practical session

Practical session

A glimpse of using Formal Verification for NN

- **Tutorial is divided in 4 parts:**
  1. Verification by hand
  2. Small problem verification
  3. Real use case application
  4. Image classification

- **Get all the files from …**

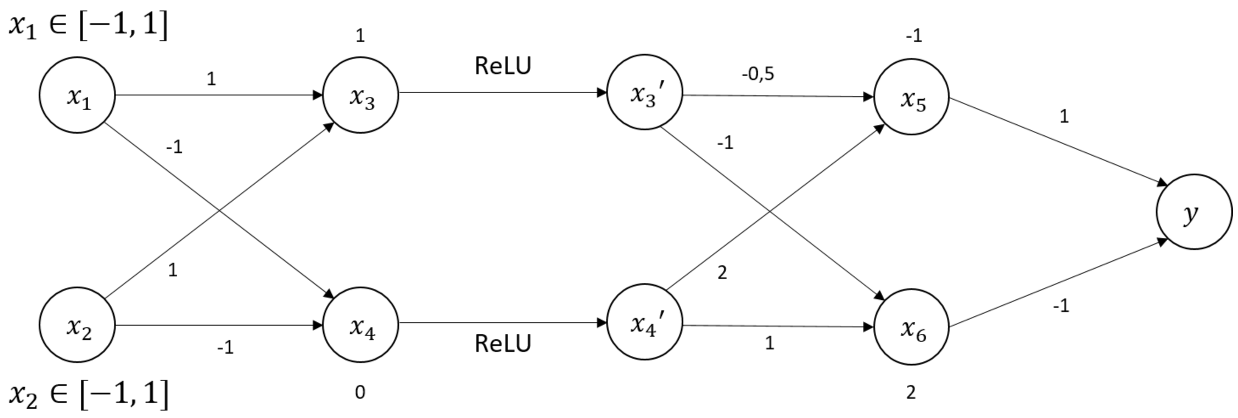
- **Follow the instruction in the README.md file to setup the environment**

- **Run `jupyter notebook tutorial.ipynb` to start the tutorial**

Part 1: Verification by hand

Create your first abstract interpretation based tool

1. Encode the network
2. Create an interval
3. Run the network



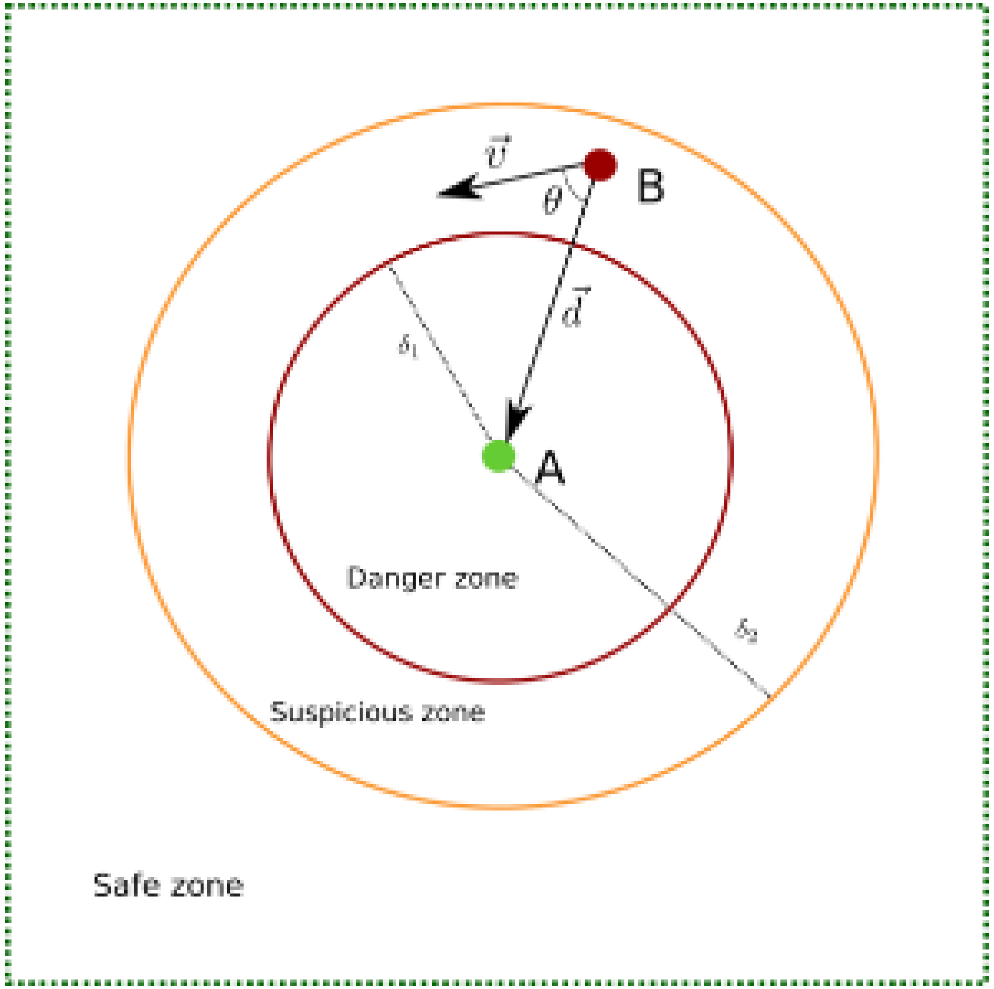
Intervals rules:

- $[l, u] + \lambda = [l + \lambda, u + \lambda]$
- $[l, u] + [l', u'] = [l + l', u + u']$
- $-[l, u] = [-u, -l]$
- $[l, u] - [l', u'] = [l - u', u - l']$
- $[l, u] * \lambda =$
  - si $\lambda >= 0$ -> $[\lambda * l, \lambda * u]$
  - si $\lambda < 0$ -> $[\lambda * u, \lambda * l]$

Part 2: Small problem verification

Small problem

The setting



Let A be a Guardian, and B a Threat. The goal for the Guardian is to send an ALARM when the Threat arrives too close.

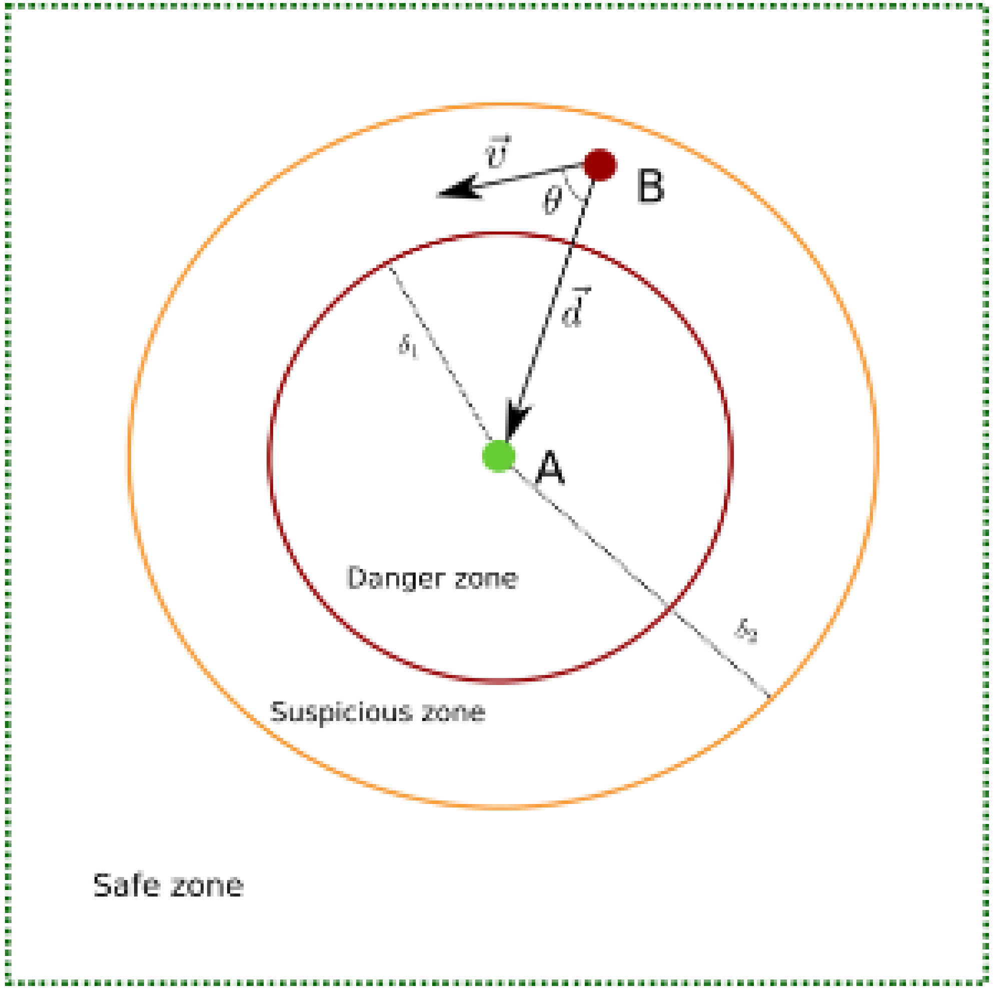
The Guardian has access to the following data:

- The distance from B to A d
- the speed of B v
- the angle θ between d and v

- Network output y
  - $y \geq 0 \Rightarrow ALARM$
  - $y < 0 \Rightarrow SAFE$

Small problem

The properties



We define three zones:

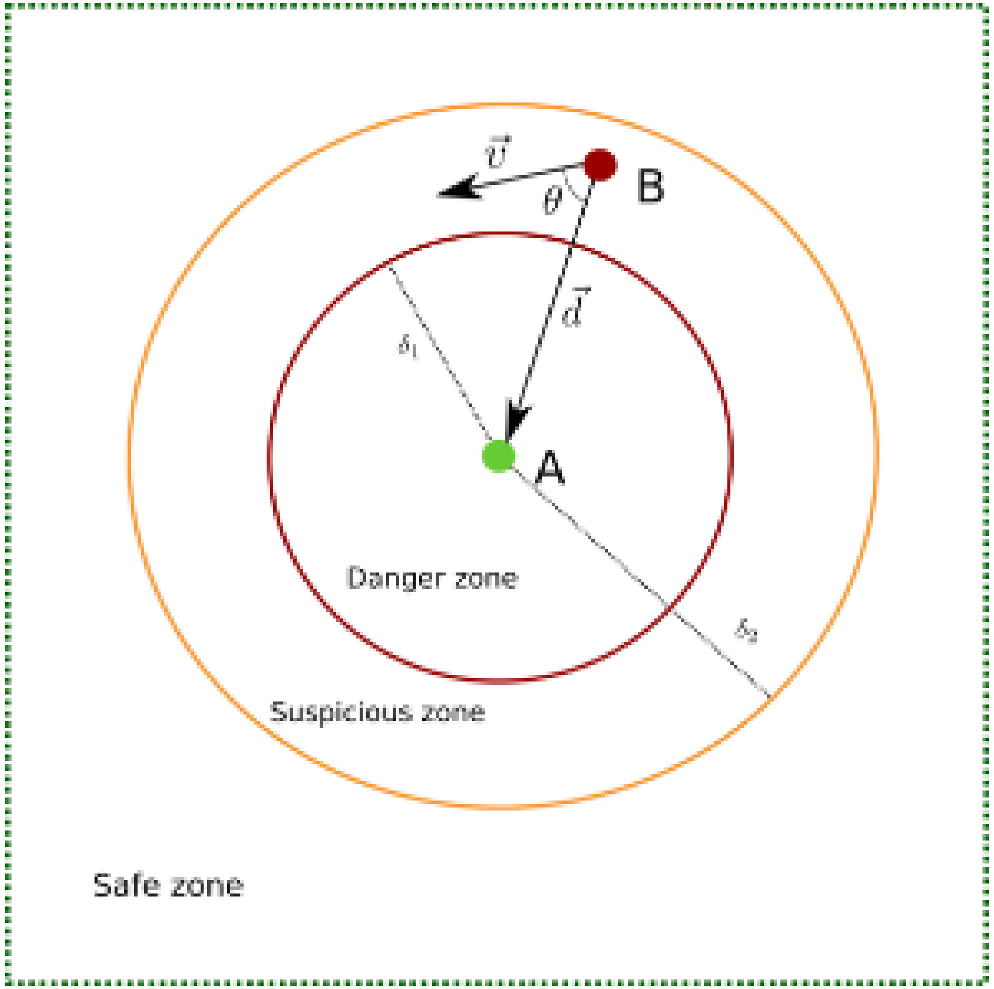
- « **safe** » zone when $d \geq \delta_2$
  - Implies no alarm
- « **suspicious** » zone when $\delta_1 < d < \delta_2$
  - Alarm if $v > \alpha \ and \ \theta < \beta$
- « **danger** » zone when $d \leq \delta_1$
  - Implies alarm

- For the training we had:
  - $\alpha = 0{,}5$
  - $\beta = 0{,}25$
  - $\delta_1 = 0{,}3$
  - $\delta_2 = 0{,}7$

- The main point is to see if the properties hold on the network with these values

Small problem

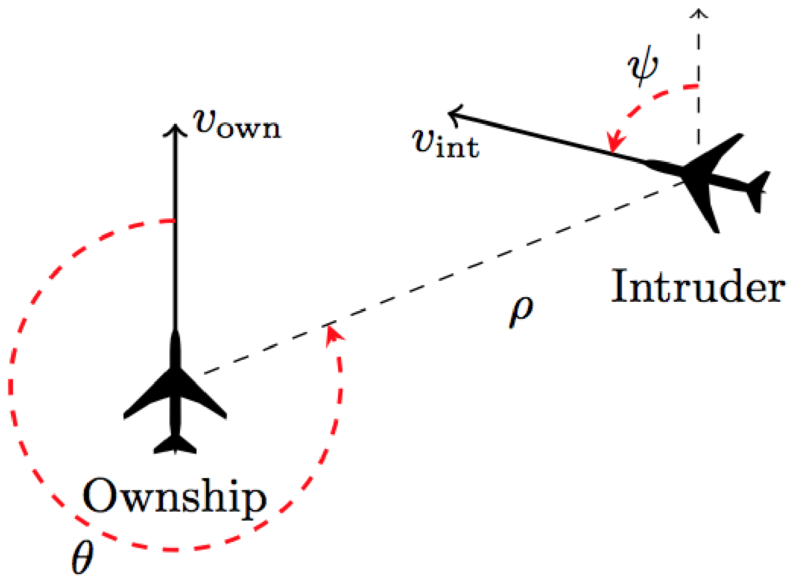
TODO



- Create the safety property
  - Visualise the ouputs
  - Write the property
- Launch solvers to get results
  - **Z3**: the classical SMT solver
  - **Marabou**: the simplex based solver
  - **PyRAT**: with Abstract interpretation
- Moving to bigger properties
  - Try and prove the previous properties

Part 3: A real problem

Airborne Collision Avoidance System for Unmanned vehicles



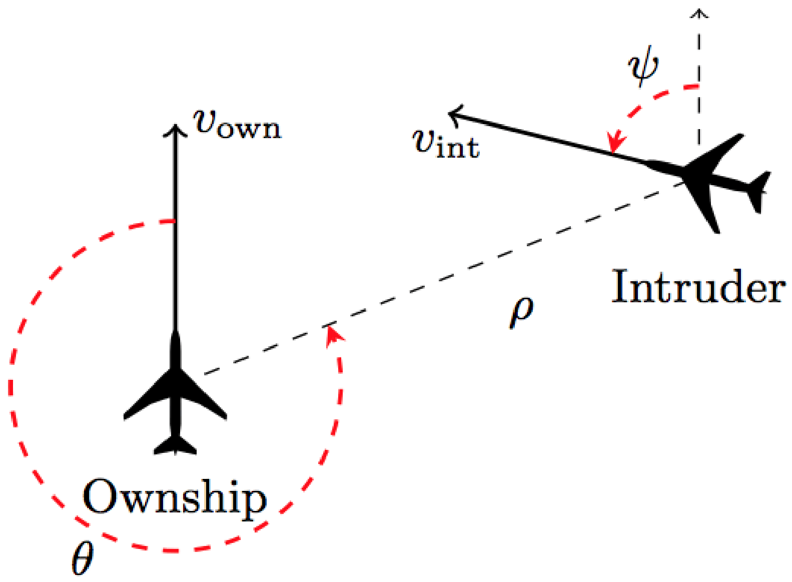
- **Formerly implemented as lookup tables**
  1. From to 2 GB to 3 MB

- **In Reluplex, they provided 45 networks and 10 safety properties on them**
- **This became a common benchmark for safety**

- **The network takes 5 inputs**
  1. ρ (m): Distance from ownship to intruder.
  2. θ (rad): Angle to intruder relative to ownship heading direction.
  3. ψ (rad): Heading angle of intruder relative to ownship heading direction.
  4. v_{own} (m/s): Speed of ownship.
  5. v_{int} (m/s) Speed of intruder.

- **5 outputs a score for Clear of Conflict, Right, Strong Right, Left, Strong Left**

The property



- **Description:** If the intruder is distant and is significantly slower than the ownship, the score of a COC advisory will always be below a certain fixed threshold.
- **Input constraints:** ρ ≥ 55947.691, $v_{own} \geq 1145, v_{int} \leq 60$.
- **Desired output property:** the score for COC is at most 1500.

- **Normalized and translated**
  - $0{,}6 \leq \rho \leq 0{,}6798577687$
  - $-0{,}5 \leq \theta \leq 0{,}5$
  - $-0{,}5 \leq \psi \leq 0{,}5$
  - $0{,}45 \leq v_{own} < 0{,}5$
  - $-0{,}5 \leq v_{int} \leq 0{,}45$

ACAS-Xu

First outputs

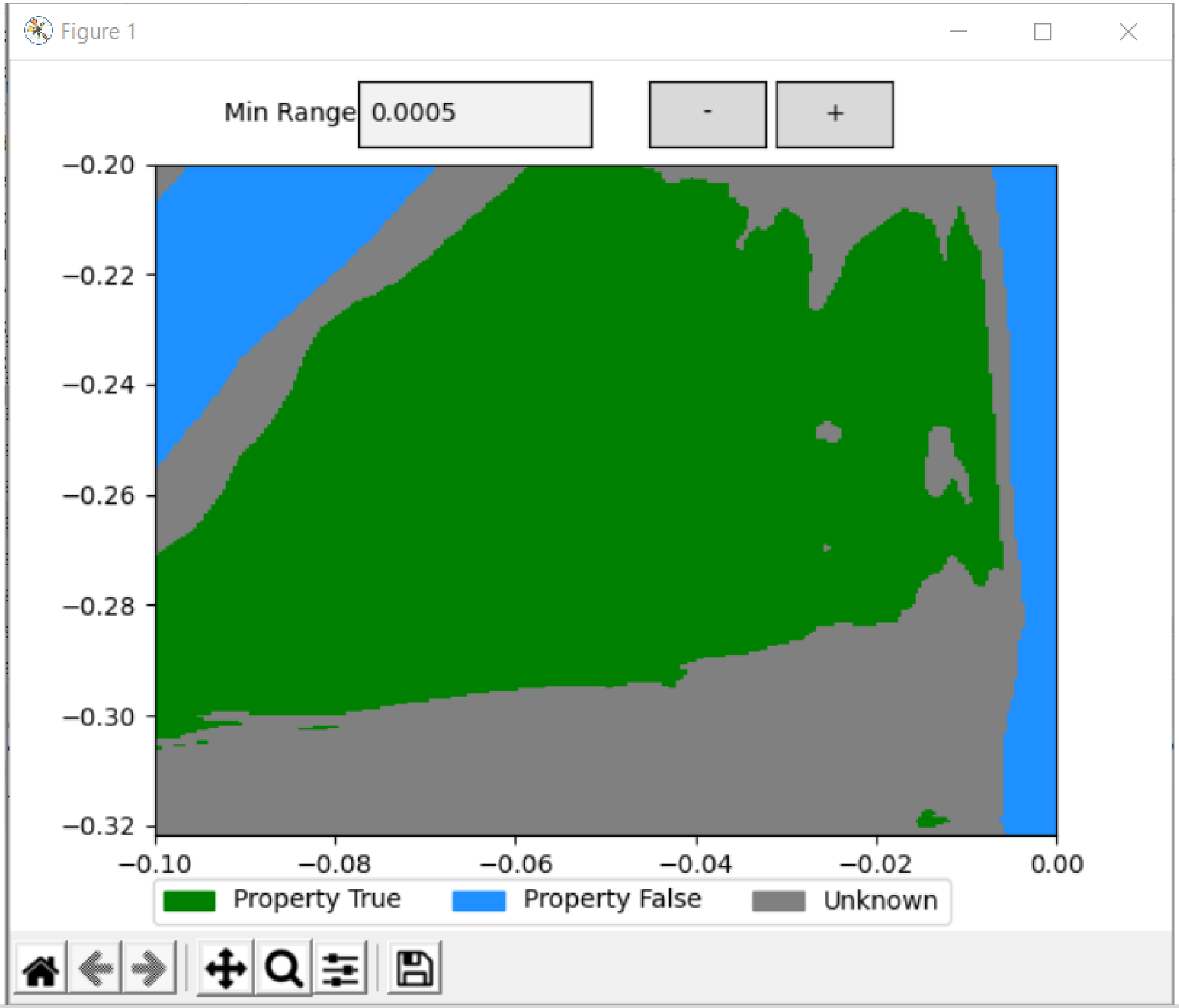
- **PyRAT**

```
Output bounds:
 [-10710.81020574  -9110.01167952 -12047.17947303  -2730.80270361
 -13194.04888676]
[ 5272.51746506  5429.67755395  4817.41682782 10864.69018591
  5844.88081043]
Result = Unknown, Time = 0.01 s
```

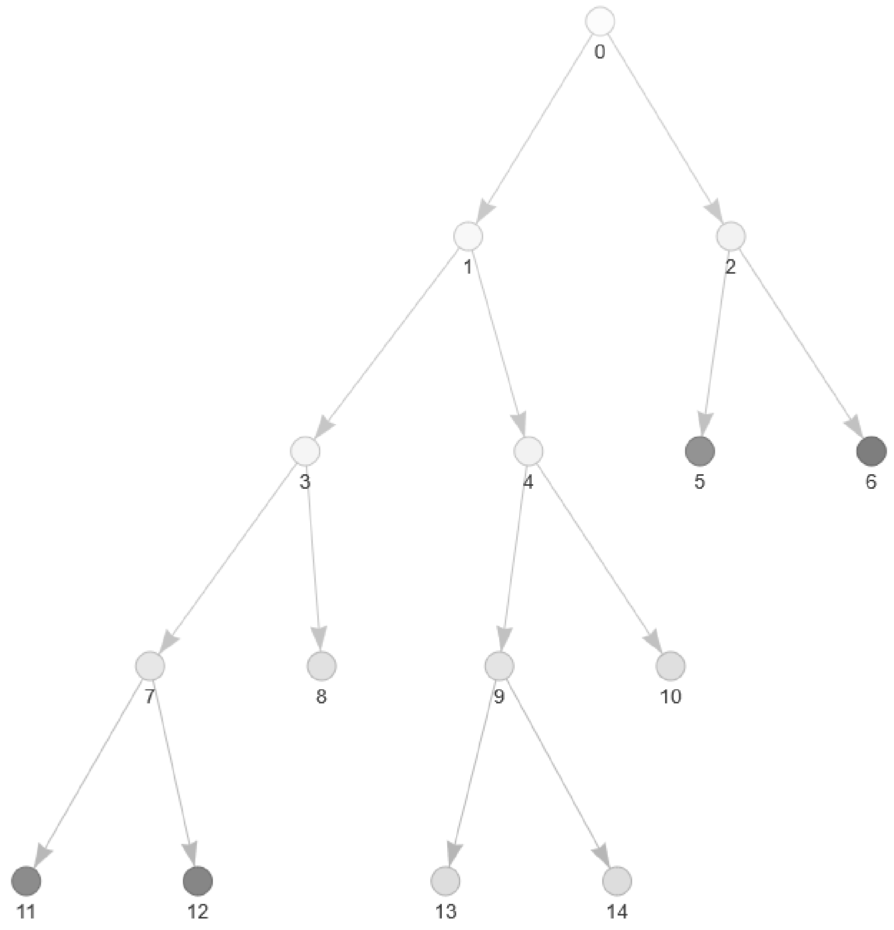
- **Marabou**

  Does not finish with only 1 process

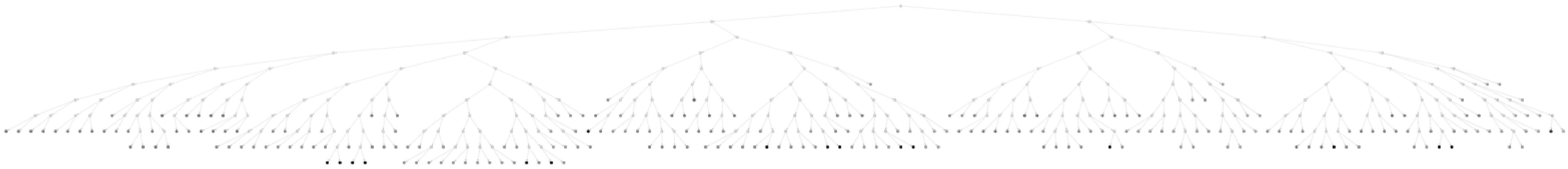
Splitting the input space



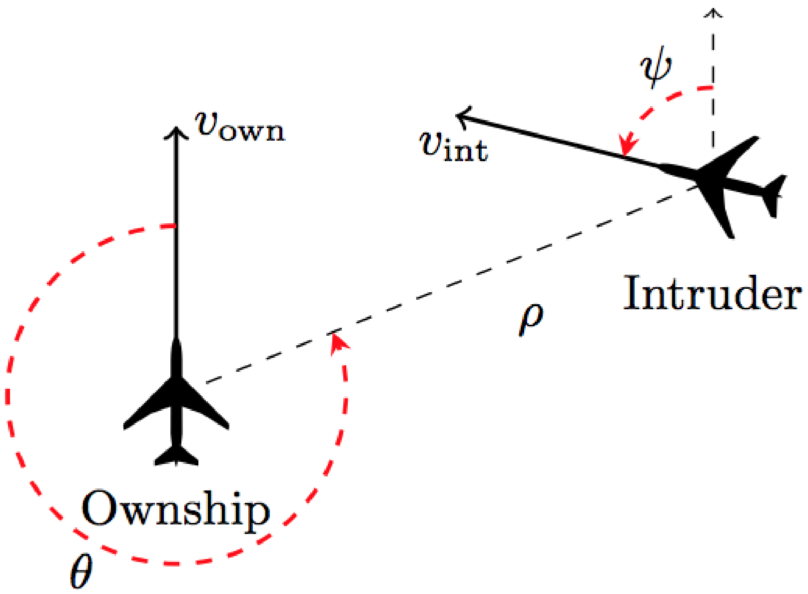
Splitting the input space



Splitting the input space



Your turn to verify the property

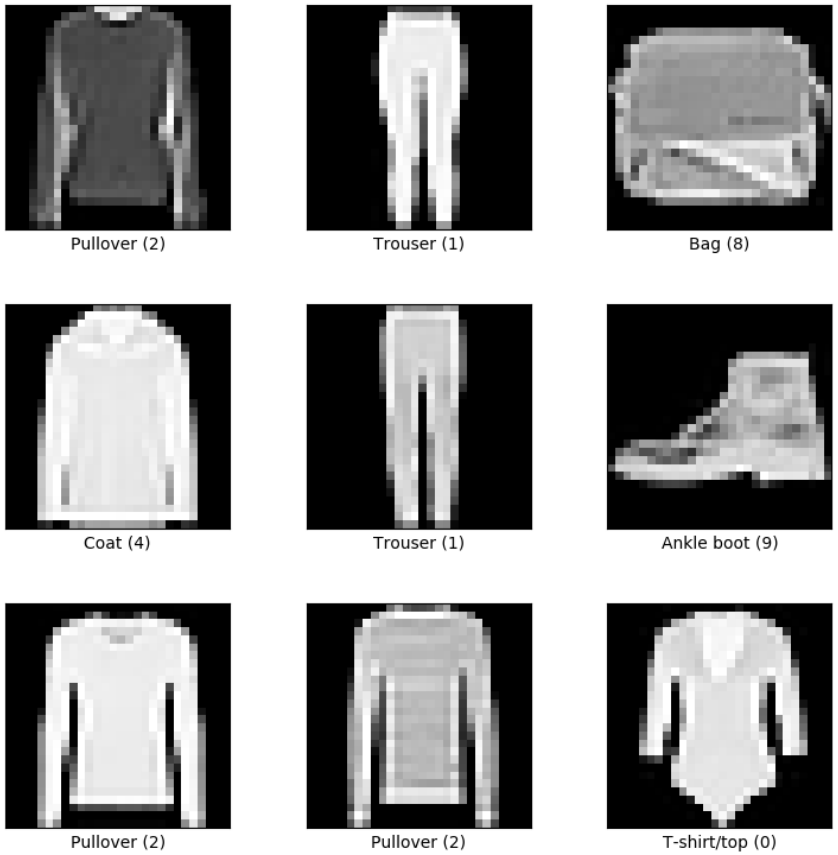


- **Test PyRAT & Marabou**
- **Implement your own splitting approach**
  1. Divide a formula or an Interval
  2. Create the iteration algorithm
- **Can you do better than 281 analysis?**

Part 4: Image classification

Fashion mnist dataset

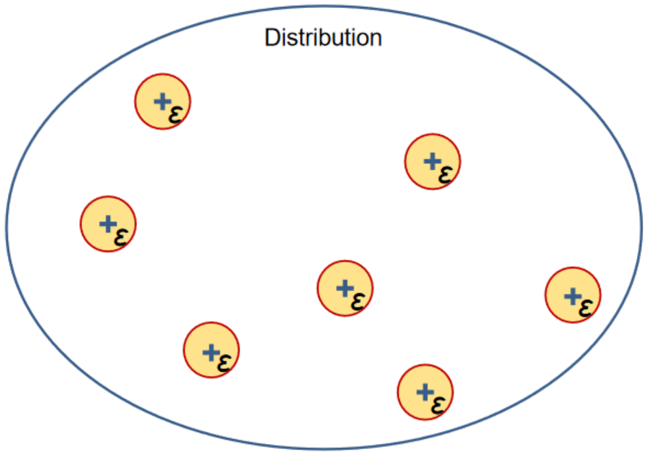
- **Zalando clothes dataset**
- **28x28 grayscale images**
- **10 classes**
  - T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot
- **Subset of 50 images**

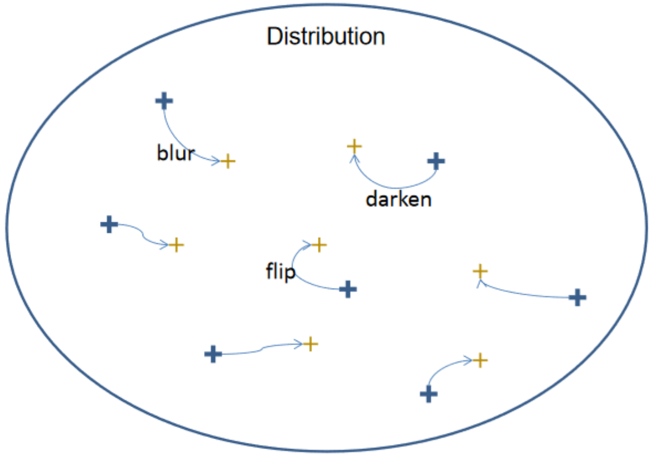


How to specify a safety property?

- Global properties won't work
- Too many input dimension splitting will not work

- Local robustness
- Testing





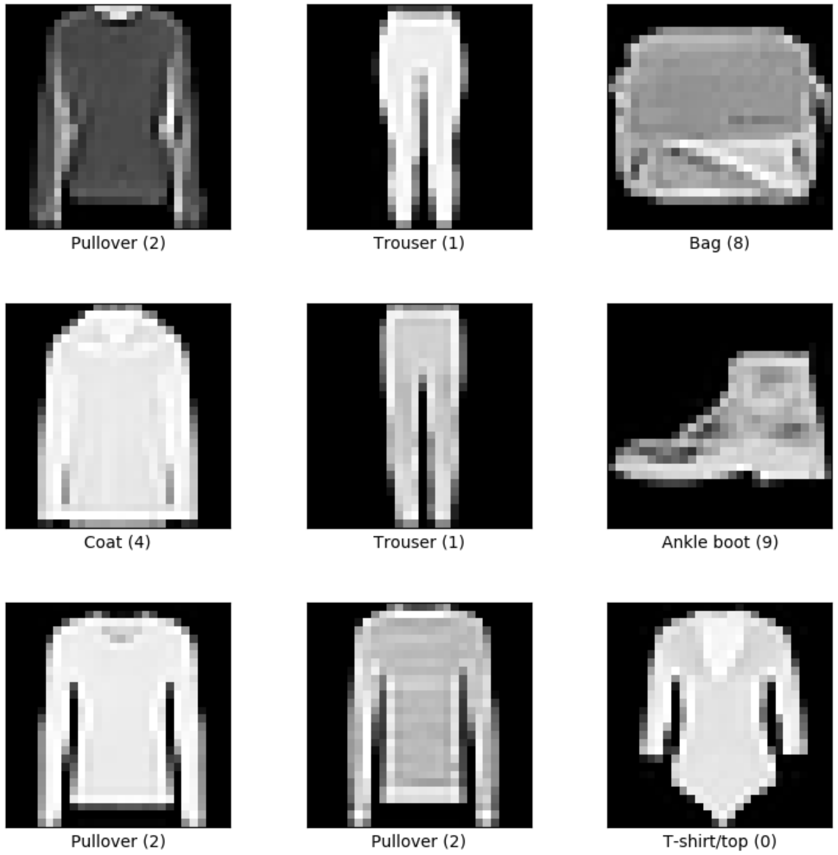


Image classification

Goal: Decide which network is the best suited for our needs

- **For our critical system we developed 5 networks with different methods**
  1. Baseline model, normal training
  2. Adversarial model, adversarial training
  3. Pruned model, normal training + pruning
  4. Certified model, certified training
  5. Pruned certified model, certified training + pruning

- **The accuracy of the model is already calculated on the test set**

Model	Accuracy
Baseline	90.50%
Adversarial	79%
Pruned	89%
Certified	72.30%
Pruned Certified	73.20%

Image classification

TODO

- **Bench for local robustness and plot the results**
    1. From 1/255 to 75/255 of perturbation

- **Test the model against metamorphic properties**
    1. Luminosity of the setting varies from -30 to +30
    2. Angle of the clothes might vary from -15° to + 15°
    3. Picture can be blurry

- **Choose the best model**