Rigorous Verification and Explanation of ML Models

A. Ignatiev, J. Marques-Silva, K. Meel & N. Narodytska

Monash Univ, NU Singapore, VMWare Research & ANITI@Univ. Toulouse

February 08, 2020 | AAAI Tutorial SP1

Many ML successes

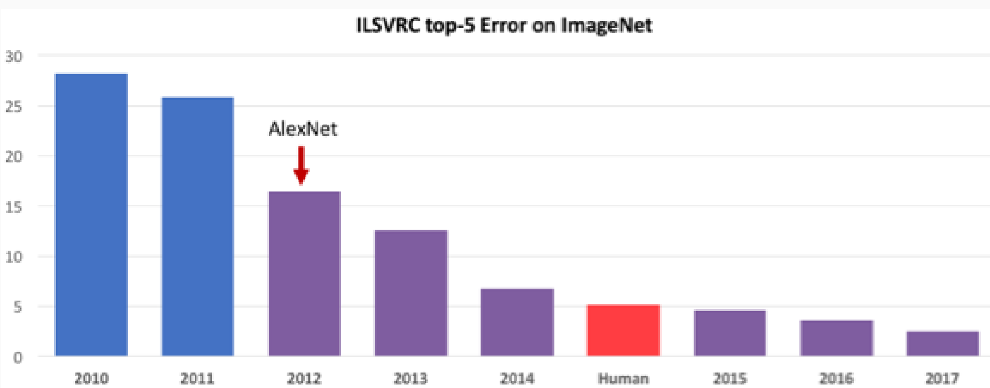
https://en.wikipedia.org/wiki/Waymo

DeepMind

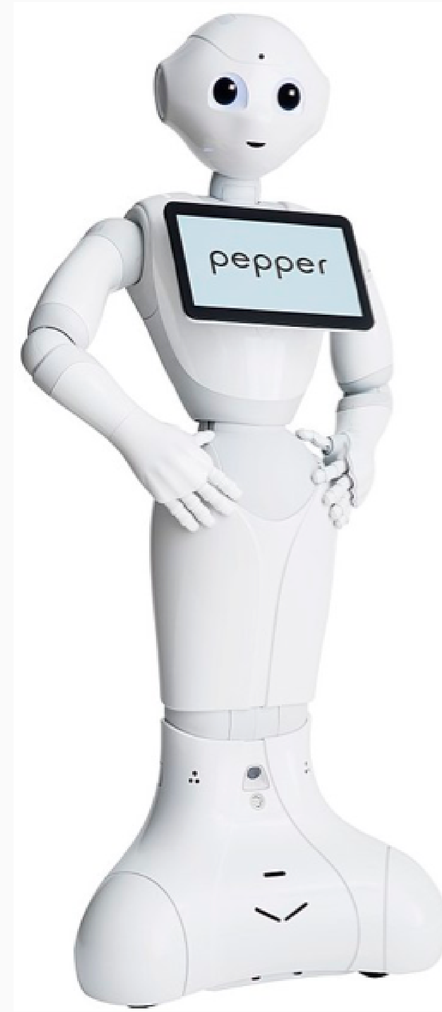
AlphaGo

AlphaGo Zero & **Alpha Zero**

Image & Speech Recognition

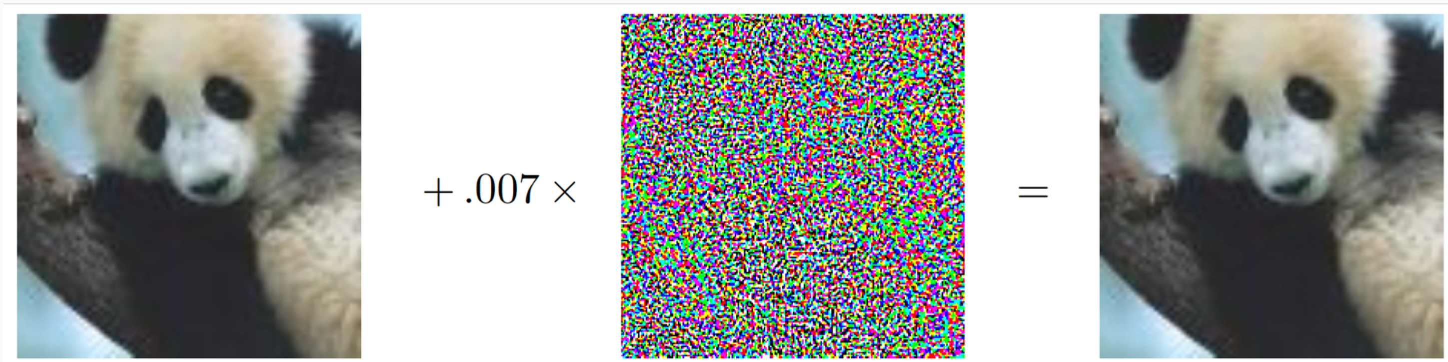


http://gradientscience.org/intro_adversarial/



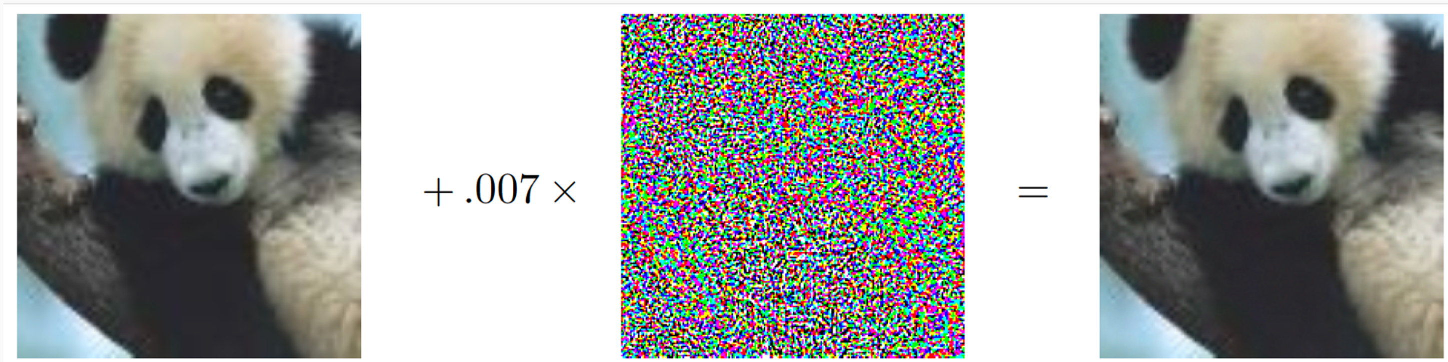
https://fr.wikipedia.org/wiki/Pepper_(robot)

Problem: ML models are brittle



Goodfellow et al., ICLR'15

Problem: ML models are brittle

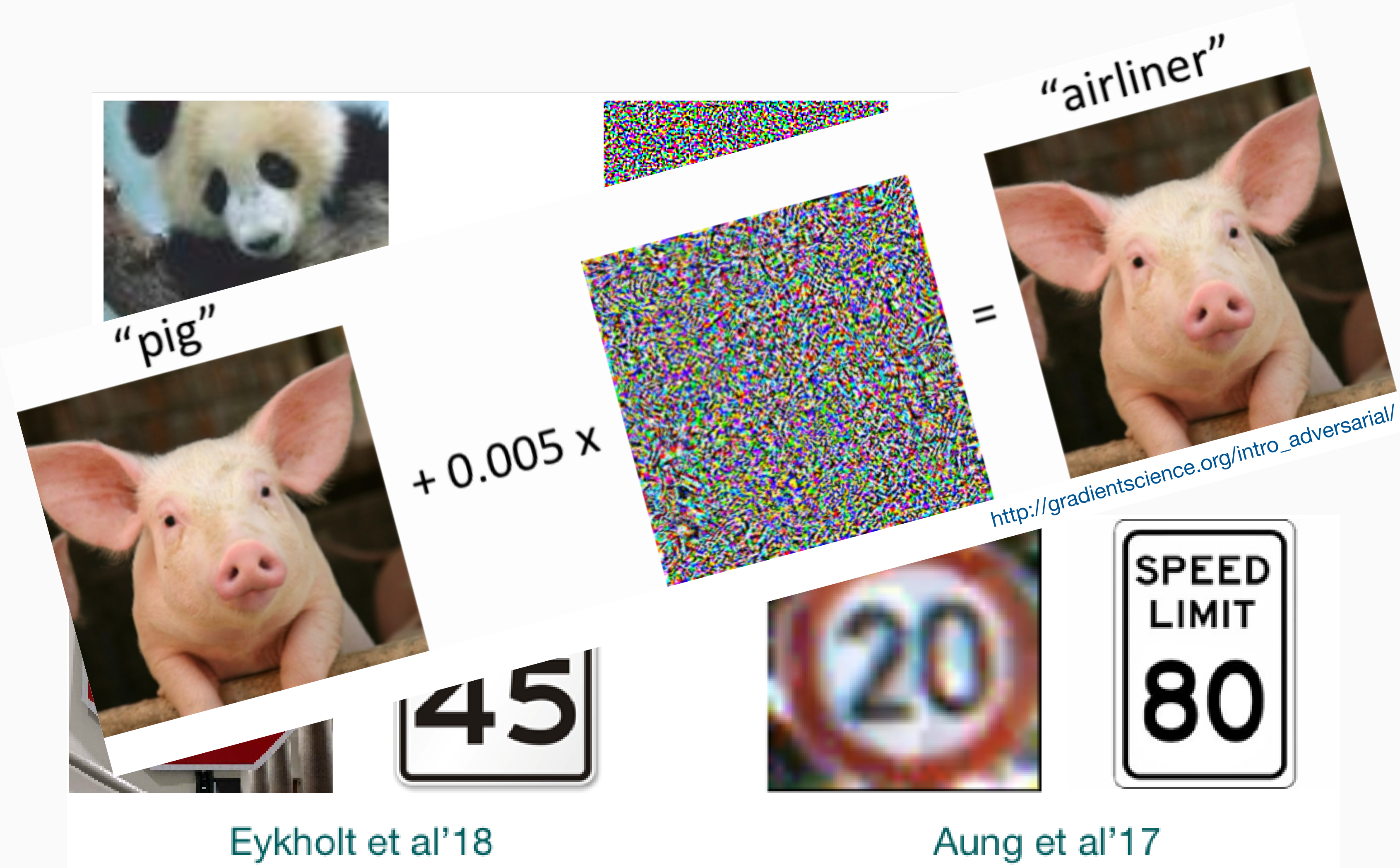


Goodfellow et al., ICLR'15

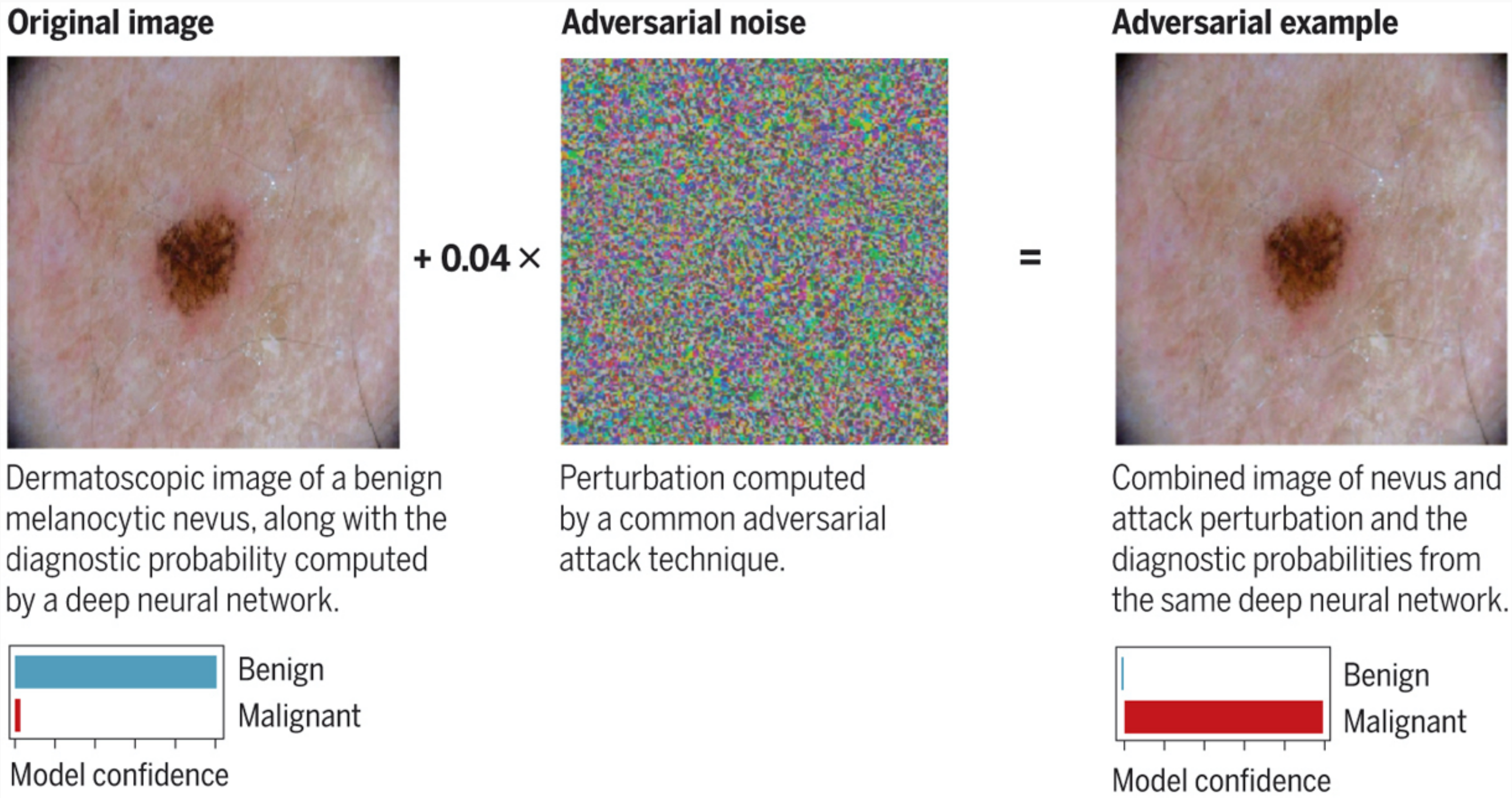
Eykholt et al'18

Aung et al'17

Problem: ML models are brittle

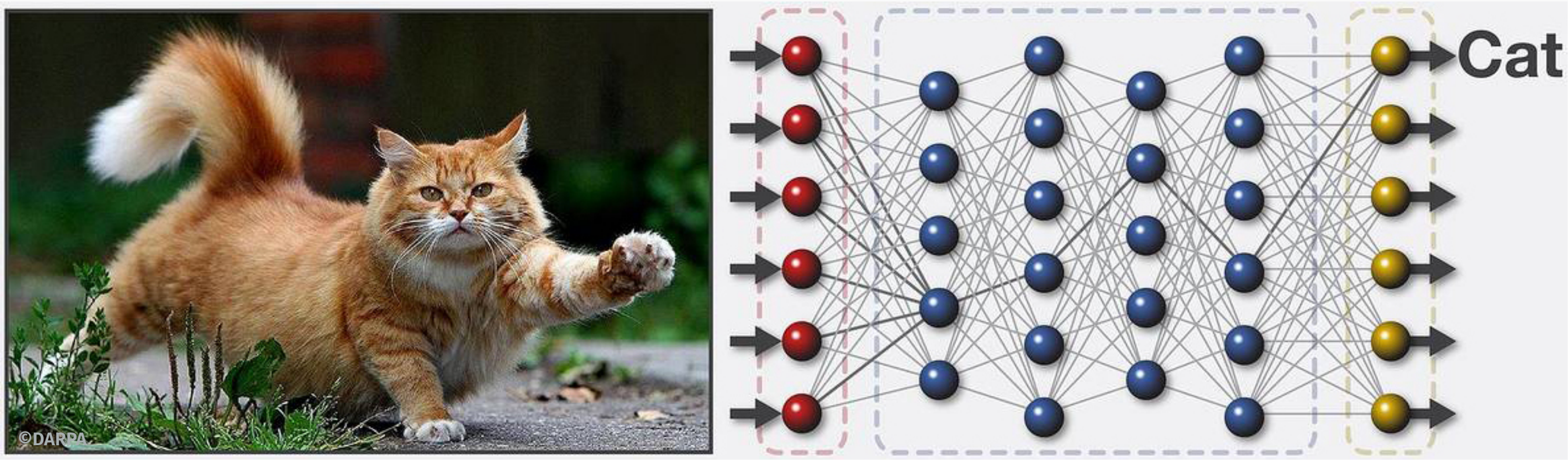


Adversarial examples can be very unsettling

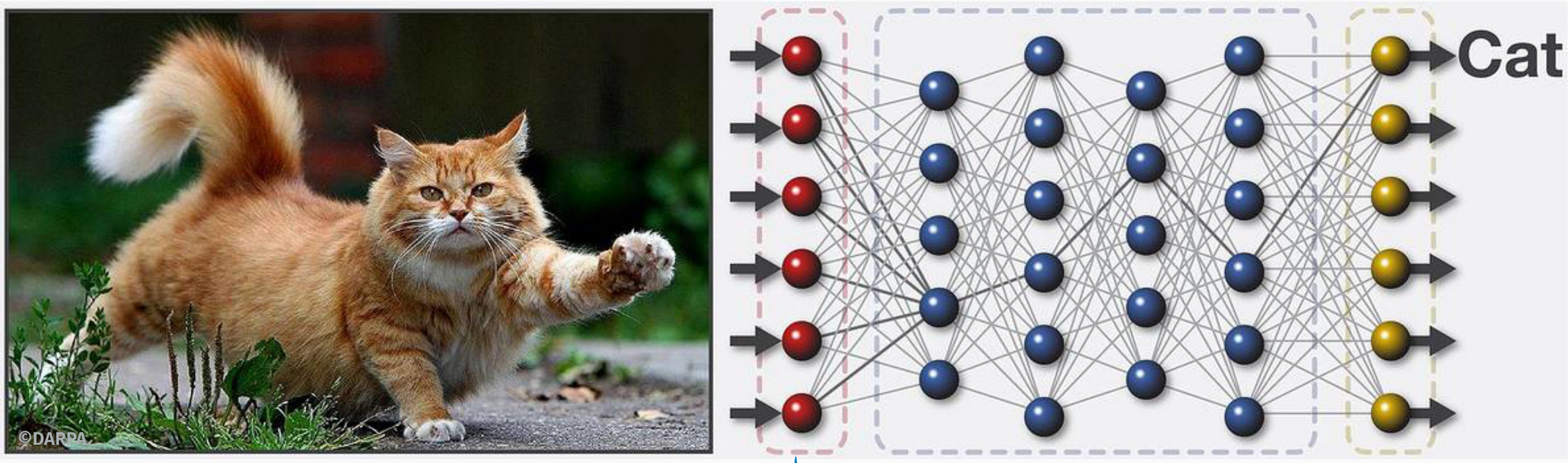


Finlayson et al., Nature 2019

Problem: ML models are opaque

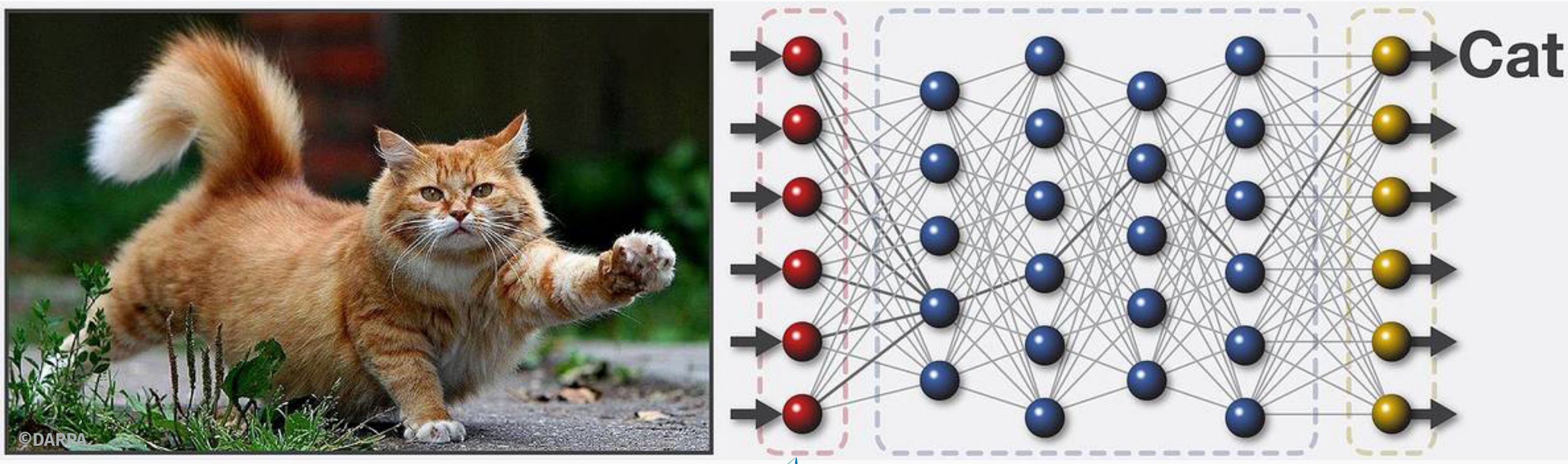


Problem: ML models are opaque



Why does the NN predict a cat?

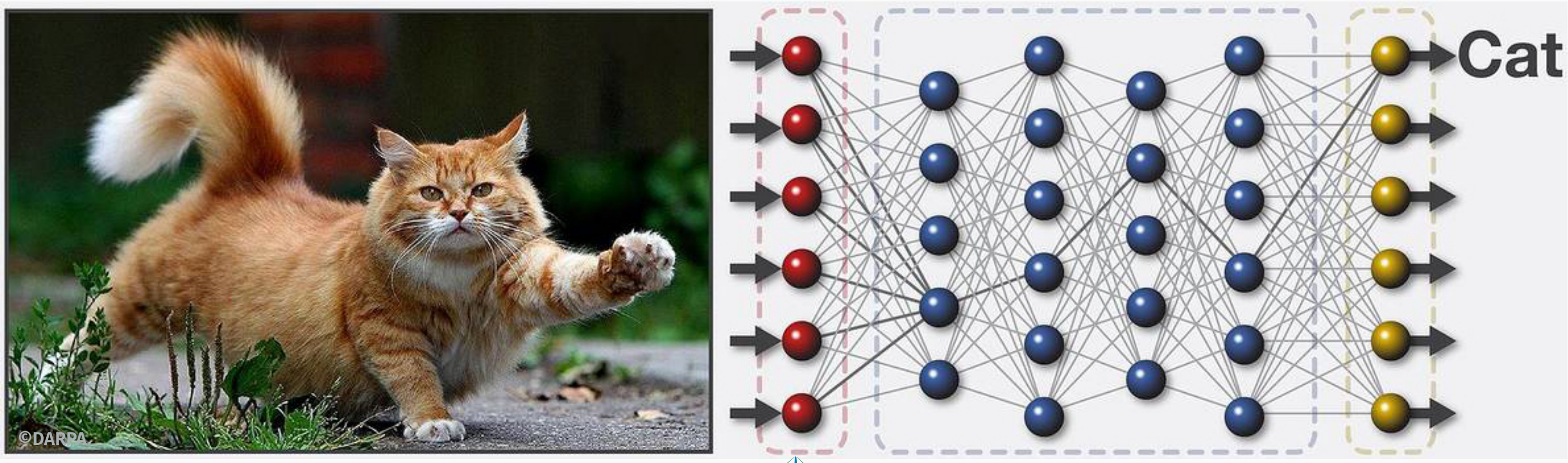
Problem: ML models are opaque



Why does the NN predict a cat?

Which features matter?

Problem: ML models are opaque



Why does the NN predict a cat?

Which features matter?

Are there general explanations??

Why XAI?

REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 27 April 2016

on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

(Text with EEA relevance)

Why XAI?

REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 27 April 2016

on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

(Text with EEA relevance)

European Union regulations on algorithmic decision-making and a "right to explanation"

Bryce Goodman,[1*] Seth Flaxman,[2]

Why XAI?

REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 27 April 2016

on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

(Text with EEA relevance)

European Union regulations on algorithmic decision-making and a "right to explanation"

Bryce Goodman,[1*] Seth Flaxman,[2]

■ *We summarize the potential impact that the European Union's new General Data Protection Regulation will have on the routine use of machine-learning algorithms. Slated to take effect as law across the European Union in 2018, it will place restrictions on automated individual decision making (that is, algorithms that make decisions based on user-level predictors) that "significantly affect" users. When put into practice, the law may also effectively create a right to explanation, whereby a user can ask for an explanation of an algorithmic decision that significantly affects them. We argue that while this law may pose large challenges for industry, it highlights opportunities for computer scientists to take the lead in designing algorithms and evaluation frameworks that avoid discrimination and enable explanation.*

Why XAI?

REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 27 April 2016

on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

(Text with EEA relevance)

European Union regulations on algorithmic decision-making and a "right to explanation"

Bryce Goodman,[1*] Seth Flaxman,[2]

POLICY \ US & WORLD \ TECH

TheVerge.com

A new bill would force companies to check their algorithms for bias

By Adi Robertson | @thedextriarchy | Apr 10, 2019, 3:52pm EDT

Algorithmic Accountability Act

We summarize the potential impact that the European Union's new General Data Protection Regulation will have on the routine use of machine-learning algorithms. Slated to take effect as law across the European Union in 2018, it will place restrictions on automated individual decision making (that is, algorithms that make decisions based on user-level predictors) that "significantly affect" users. When put into practice, the law may also effectively create a right to explanation, whereby a user can ask for an explanation of an algorithmic decision that significantly affects them. We argue that while this law may pose large challenges for industry, it highlights opportunities for computer scientists to take the lead in designing algorithms and evaluation frameworks that avoid discrimination and enable explanation.

Explainable Artificial Intelligence (XAI)

XAI

EXPLAINABLE ARTIFICIAL INTELLIGENCE

FY17 FY18 FY19 FY20 FY21

David Gunning
DARPA/I2O
Program Update November 2017

DARPA

Why XAI?

REGULATION (EU) 2016/67… THE COUNCIL

on the p… …ata and on the free
movem… …tion Regulation)

European Union regulation …
and a "righ…

Bryce Goodma…

…mpanies to check their

TheVerge.com

Algorithmic Accountability Act

■ *We summarize the potential im… that the European Union's new Gene… Data Protection Regulation will have … the routine use of machine-learni… algorithms. Slated to take effect as la… across the European Union in 2018, … will place restrictions on automate… individual decision making (that is, algorithms that make decisions based on user-level predictors) that "significantly affect" users. When put into practice, the law may also effectively create a right to explanation, whereby a user can ask for an explanation of an algorithmic decision that significantly affects them. We argue that while this law may pose large challenges for industry, it highlights opportunities for computer scientists to take the lead in designing algorithms and evaluation frameworks that avoid discrimination and enable explanation.*

In order to trust deployed AI systems, we must not only improve their robustness,[5] but also develop ways to make their reasoning intelligible. Intelligibility will help us spot AI that makes mistakes due to distributional drift or incomplete representations of goals and features. Intelligibility will also facilitate control by humans in increasingly common collaborative human/AI teams. Furthermore, intelligibility will help humans learn from AI. Finally, there are legal reasons to want intelligible AI, including the European GDPR and a growing need to assign liability when AI errs.

Weld & Bansal, CACM, Jun'19

…ence (XAI)

…ber 2017

DARPA

©DARPA

XAI & EU guidelines

Search

European Commission › Strategy › Digital Single Market › Reports and studies ›

Digital Single Market

REPORT / STUDY | 8 April 2019

Ethics guidelines for trustworthy AI

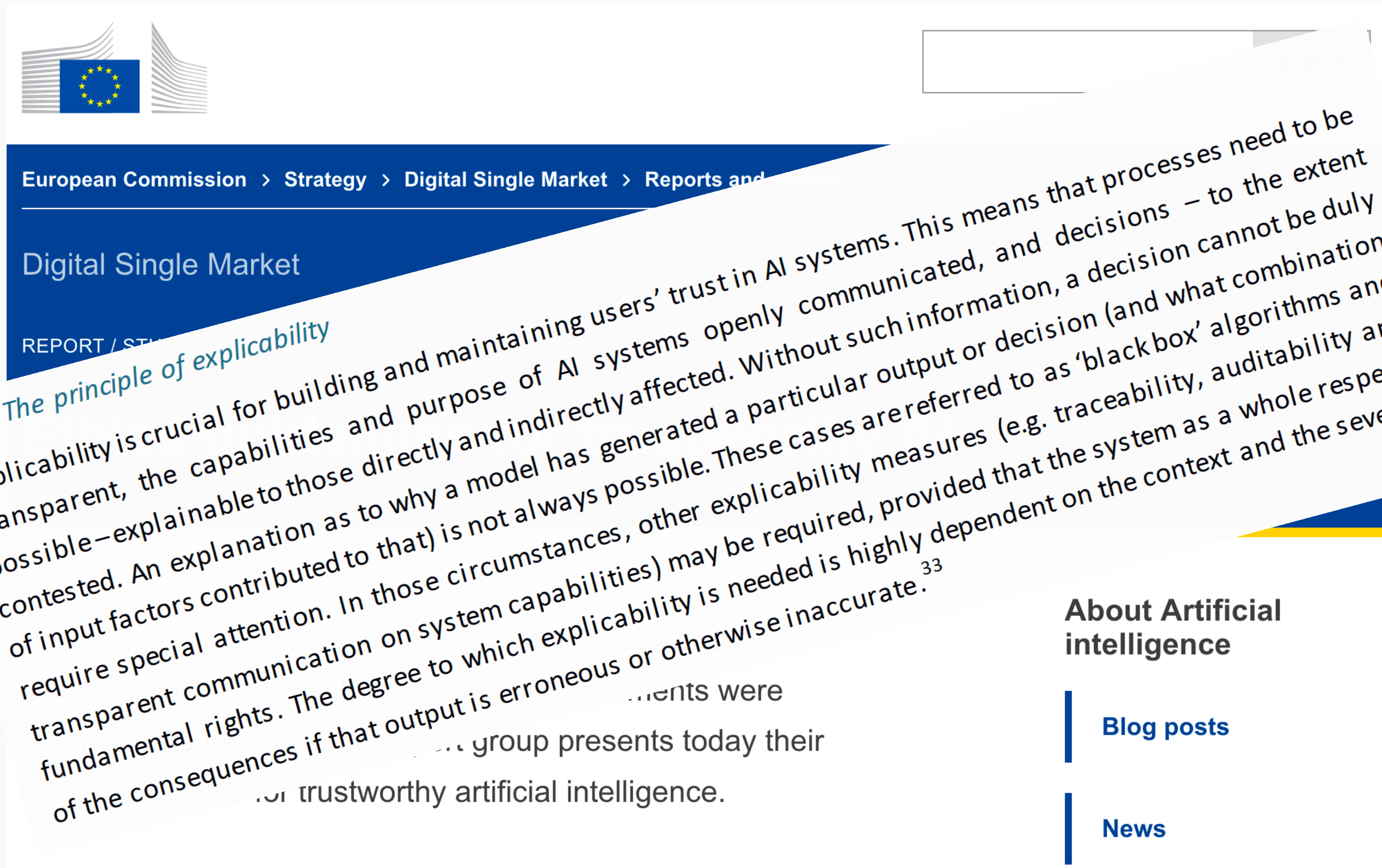
Following the publication of the draft ethics guidelines in December 2018 to which more than 500 comments were received, the independent expert group presents today their ethics guidelines for trustworthy artificial intelligence.

About Artificial intelligence

Blog posts

News

XAI & the principle of explicability



- *The principle of explicability*
  Explicability is crucial for building and maintaining users' trust in AI systems. This means that processes need to be transparent, the capabilities and purpose of AI systems openly communicated, and decisions – to the extent possible – explainable to those directly and indirectly affected. Without such information, a decision cannot be duly contested. An explanation as to why a model has generated a particular output or decision (and what combination of input factors contributed to that) is not always possible. These cases are referred to as 'black box' algorithms and require special attention. In those circumstances, other explicability measures (e.g. traceability, auditability and transparent communication on system capabilities) may be required, provided that the system as a whole respects fundamental rights. The degree to which explicability is needed is highly dependent on the context and the severity of the consequences if that output is erroneous or otherwise inaccurate.[33]

Heuristic explanation approaches unsettling

Dataset	(# unique)	Explanations								
		incorrect			redundant			minimal		
		LIME	Anchor	SHAP	LIME	Anchor	SHAP	LIME	Anchor	SHAP
adult	(5579)	61.3%	80.5%	70.7%	7.9%	1.6%	10.2%	30.8%	17.9%	19.1%
lending	(4414)	24.0%	3.0%	17.0%	0.4%	0.0%	2.5%	75.6%	97.0%	80.5%
rcdv	(3696)	94.1%	99.4%	85.9%	4.6%	0.4%	7.9%	1.3%	0.2%	6.2%
compas	(778)	71.9%	84.4%	60.4%	20.6%	1.7%	27.8%	7.5%	13.9%	11.8%
german	(1000)	85.3%	99.7%	63.0%	14.6%	0.2%	37.0%	0.1%	0.1%	0.0%

Heuristic explanation approaches unsettling

Dataset	(# unique)	Explanations: incorrect LIME	incorrect Anchor	incorrect SHAP	redundant LIME	redundant Anchor	redundant SHAP	minimal LIME	minimal Anchor	minimal SHAP
adult	(5579)	61.3%	80.5%	70.7%	7.9%	1.6%	10.2%	30.8%	17.9%	19.1%
lending	(4414)	24.0%	3.0%	17.0%	0.4%	0.0%	2.5%	75.6%	97.0%	80.5%
rcdv	(3696)	94.1%	99.4%	85.9%	4.6%	0.4%	7.9%	1.3%	0.2%	6.2%
compas	(778)	71.9%	84.4%	60.4%	20.6%	1.7%	27.8%	7.5%	13.9%	11.8%
german	(1000)	85.3%	99.7%	63.0%	14.6%	0.2%	37.0%	0.1%	0.1%	0.0%

Similar results for Google's XAI service??

Solutions to problems?

- Assess robustness
- Learn interpretable models
- Explain black-box models
- How about heuristic approaches?

Solutions to problems?

- Assess robustness
  - How easy it is to fool and ML model?

- Learn interpretable models

- Explain black-box models

- How about heuristic approaches?

Solutions to problems?

- Assess robustness
  - How easy it is to fool and ML model?
- Learn interpretable models
  - Decision trees; decision sets; decision lists; etc.
- Explain black-box models
- How about heuristic approaches?

Solutions to problems?

- Assess robustness
  - How easy it is to fool and ML model?
- Learn interpretable models
  - Decision trees; decision sets; decision lists; etc.
- Explain black-box models
  - By using some accepted definition of explanation
- How about heuristic approaches?

Solutions to problems?

- Assess robustness
  - How easy it is to fool and ML model?
- Learn interpretable models
  - Decision trees; decision sets; decision lists; etc.
- Explain black-box models
  - By using some accepted definition of explanation
- How about heuristic approaches?
  - **No** formal guarantees provided

How/Why to reason about ML models, with formal guarantees?

- Problem complexity **not** necessarily an hopeless obstacle

How/Why to reason about ML models, with formal guarantees?

- Problem complexity **not** necessarily an hopeless obstacle
- Efficient reasoners
  - SAT, SMT, CP, ILP, etc.

How/Why to reason about ML models, with formal guarantees?

- Problem complexity **not** necessarily an hopeless obstacle

- Efficient reasoners
  - SAT, SMT, CP, ILP, etc.

- Effective problem encodings

How/Why to reason about ML models, with formal guarantees?

- Problem complexity **not** necessarily an hopeless obstacle
- Efficient reasoners
  - SAT, SMT, CP, ILP, etc.
- Effective problem encodings
- Exploit known solutions (more latter)
  - Exploit reasoners for efficient problem solving

This tutorial – formal reasoning in ML

- Part 01: first contact with formal reasoning tools — Joao
- Part 02: learning interpretable models — Kuldeep
- Part 03: assessing robustness of ML models — Nina
- Part 04: rigorous explanations of ML models — Alexey
- Part 05: duality in explanations & wrap-up — Joao

1 Preliminaries

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$
- **Features**: $\mathcal{F} = \{f_1, \ldots, f_K\}$
  - For binary features: f_r and $\neg f_r$
  - Otherwise, literals $f_r = v_{ri}$

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$
- **Features**: $\mathcal{F} = \{f_1, \ldots, f_K\}$
  - For binary features: f_r and $\neg f_r$
  - Otherwise, literals $f_r = v_{ri}$
- **Feature space**: $\mathcal{U} \triangleq \prod_{r=1}^{K} \{f_r = v_{r1}, \ldots, f_r = v_{rk_r}\}$

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$
- **Features**: $\mathcal{F} = \{f_1, \ldots, f_K\}$
  - For binary features: f_r and $\neg f_r$
  - Otherwise, literals $f_r = v_{ri}$
- **Feature space**: $\mathcal{U} \triangleq \prod_{r=1}^{K} \{f_r = v_{r1}, \ldots, f_r = v_{rk_r}\}$
- **Classification**: $\mathcal{C} = \{c_0, c_1, \ldots, c_M\}$

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$
- **Features**: $\mathcal{F} = \{f_1, \ldots, f_K\}$
  - For binary features: f_r and $\neg f_r$
  - Otherwise, literals $f_r = v_{ri}$
- **Feature space**: $\mathcal{U} \triangleq \prod_{r=1}^{K} \{f_r = v_{r1}, \ldots, f_r = v_{rk_r}\}$
- **Classification**: $\mathcal{C} = \{c_0, c_1, \ldots, c_M\}$
- ML models: NNs, BTs, DTs, DSs, etc.

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: $($, $)$
  - Often restricted to Conjunctive Normal Form (CNF)

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: $($, $)$
  - Often restricted to Conjunctive Normal Form (CNF)
  - Goal:
    Decide whether formula has a satisfying assignment

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: $($, $)$
  - Often restricted to Conjunctive Normal Form (CNF)
  - Goal:
    Decide whether formula has a satisfying assignment

- Example:

$$\mathcal{F} \triangleq (r) \wedge (\bar{r} \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

  - Example models:

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: $($, $)$
  - Often restricted to Conjunctive Normal Form (CNF)
  - Goal:
    Decide whether formula has a satisfying assignment

- Example:

$$\mathcal{F} \triangleq (r) \wedge (\bar{r} \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

  - Example models:
    - $\{r, s, a, b, c, d\}$

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: (,)
  - Often restricted to Conjunctive Normal Form (CNF)
  - Goal:
    Decide whether formula has a satisfying assignment

- Example:

$$\mathcal{F} \triangleq (r) \wedge (\bar{r} \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

  - Example models:
    - $\{r, s, a, b, c, d\}$
    - $\{r, s, \neg x, y, \neg w, z, \neg a, b, c, d\}$

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: (,)
  - Often restricted to Conjunctive Normal Form (CNF)
  - Goal:
    Decide whether formula has a satisfying assignment

- Example:

$$\mathcal{F} \triangleq (r) \wedge (\bar{r} \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

  - Example models:
    - $\{r, s, a, b, c, d\}$
    - $\{r, s, \neg x, y, \neg w, z, \neg a, b, c, d\}$

- SAT is **NP-complete** [Coo71]

The CDCL SAT disruption

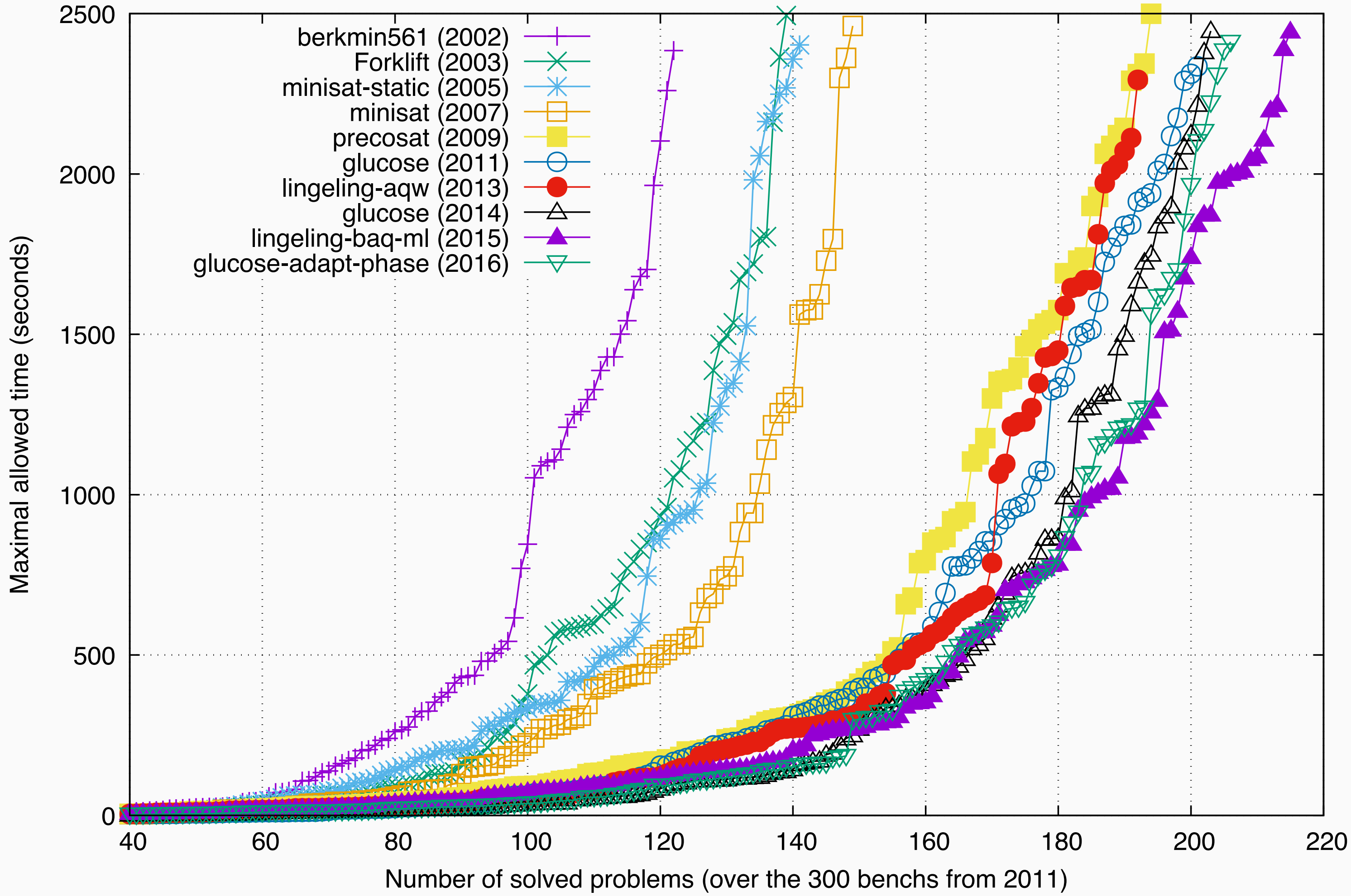
- CDCL SAT solving is a success story of Computer Science

The CDCL SAT disruption

- CDCL SAT solving is a success story of Computer Science
  - Conflict-Driven Clause Learning (CDCL)
  - (CDCL) SAT has impacted many different fields
  - Hundreds (thousands?) of practical applications

CDCL SAT solver (continued) improvement

[Source: Simon 2015]



How good are SAT solvers? – an example

- Cooperative pathfinding (CPF)
  - N agents on some grid/graph
  - Start positions
  - Goal positions
  - Minimize makespan
  - Restricted planning problem

How good are SAT solvers? – an example

- Cooperative pathfinding (CPF)
  - N agents on some grid/graph
  - Start positions
  - Goal positions
  - Minimize makespan
  - Restricted planning problem
- Concrete example
  - Gaming grid
  - 1039 vertices
  - 1928 edges
  - 100 agents

How good are SAT solvers? – an example

- Cooperative pathfinding (CPF)
  - N agents on some grid/graph
  - Start positions
  - Goal positions
  - Minimize makespan
  - Restricted planning problem
- Concrete example
  - Gaming grid
  - 1039 vertices
  - 1928 edges
  - 100 agents

```
*** tracker: a pathfinding tool ***

Initialization ... CPU Time: 0.004711
Number of variables: 113315
Tentative makespan 1
Number of variables: 226630
Number of assumptions: 1
c Running SAT solver ... CPU Time: 0.718112
c Done running SAT solver ... CPU Time: 0.830099
No solution for makespan 1
Elapsed CPU Time: 0.830112
Tentative makespan 2
Number of variables: 339945
Number of assumptions: 1
c Running SAT solver ... CPU Time: 1.27113
c Done running SAT solver ... CPU Time: 1.27114
No solution for makespan 2
Elapsed CPU Time: 1.27114
...
...
Tentative makespan 24
Number of variables: 2832875
Number of assumptions: 1
c Running SAT solver ... CPU Time: 11.8653
c Done running SAT solver ... CPU Time: 11.8653
No solution for makespan 24
Elapsed CPU Time: 11.8653
Tentative makespan 25
Number of variables: 2946190
Number of assumptions: 1
c Running SAT solver ... CPU Time: 12.3491
c Done running SAT solver ... CPU Time: 16.6882
Solution found for makespan 25
Elapsed CPU Time: 16.6995
```

How good are SAT solvers? – an example

- Cooperative pathfinding (CPF)
  - N agents on some grid/graph
  - Start positions
  - Goal positions
  - Minimize makespan
  - Restricted planning problem
- Concrete example
  - Gaming grid
  - 1039 vertices
  - 1928 edges
  - 100 agents
  - Formula w/ 2832875 variables!
  - Formula w/ 2946190 variables!

```
*** tracker: a pathfinding tool ***

Initialization ... CPU Time: 0.004711
Number of variables: 113315
Tentative makespan 1
Number of variables: 226630
Number of assumptions: 1
c Running SAT solver ... CPU Time: 0.718112
c Done running SAT solver ... CPU Time: 0.830099
No solution for makespan 1
Elapsed CPU Time: 0.830112
Tentative makespan 2
Number of variables: 339945
Number of assumptions: 1
c Running SAT solver ... CPU Time: 1.27113
c Done running SAT solver ... CPU Time: 1.27114
No solution for makespan 2
Elapsed CPU Time: 1.27114
...
...
Tentative makespan 24
Number of variables: 2832875
Number of assumptions: 1
c Running SAT solver ... CPU Time: 11.8653
c Done running SAT solver ... CPU Time: 11.8653
No solution for makespan 24
Elapsed CPU Time: 11.8653
Tentative makespan 25
Number of variables: 2946190
Number of assumptions: 1
c Running SAT solver ... CPU Time: 12.3491
c Done running SAT solver ... CPU Time: 16.6882
Solution found for makespan 25
Elapsed CPU Time: 16.6995
```

How good are SAT solvers? – an example

- Cooperative pathfinding (CPF)
  - N agents on some grid/graph
  - Start positions
  - Goal positions
  - Minimize makespan
  - Restricted planning problem
- Concrete example
  - Gaming grid
  - 1039 vertices
  - 1928 edges
  - 100 agents
  - Formula w/ 2832875 variables!
  - Formula w/ 2946190 variables!
- **Note:** In the early 90s, SAT solvers could solve formulas with a few hundred variables!

```
*** tracker: a pathfinding tool ***

Initialization ... CPU Time: 0.004711
Number of variables: 113315
Tentative makespan 1
Number of variables: 226630
Number of assumptions: 1
c Running SAT solver ... CPU Time: 0.718112
c Done running SAT solver ... CPU Time: 0.830099
No solution for makespan 1
Elapsed CPU Time: 0.830112
Tentative makespan 2
Number of variables: 339945
Number of assumptions: 1
c Running SAT solver ... CPU Time: 1.27113
c Done running SAT solver ... CPU Time: 1.27114
No solution for makespan 2
Elapsed CPU Time: 1.27114
...
...
Tentative makespan 24
Number of variables: 2832875
Number of assumptions: 1
c Running SAT solver ... CPU Time: 11.8653
c Done running SAT solver ... CPU Time: 11.8653
No solution for makespan 24
Elapsed CPU Time: 11.8653
Tentative makespan 25
Number of variables: 2946190
Number of assumptions: 1
c Running SAT solver ... CPU Time: 12.3491
c Done running SAT solver ... CPU Time: 16.6882
Solution found for makespan 25
Elapsed CPU Time: 16.6995
```

Grasping the search space ...

- Number of seconds since the Big Bang: $\approx 10^{17}$

Grasping the search space ...

- Number of seconds since the Big Bang: $\approx 10^{17}$
- Number of fundamental particles in observable universe: $\approx 10^{80}$ (or $\approx 10^{85}$)

Grasping the search space ...

- Number of seconds since the Big Bang: $\approx 10^{17}$

- Number of fundamental particles in observable universe: $\approx 10^{80}$ (or $\approx 10^{85}$)

- Search space with 2832875 propositional variables (worst case):

Grasping the search space ...

- Number of seconds since the Big Bang: $\approx 10^{17}$

- Number of fundamental particles in observable universe: $\approx 10^{80}$ (or $\approx 10^{85}$)

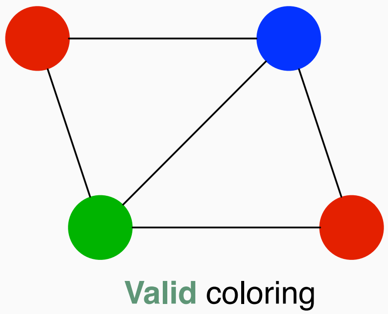
- Search space with 2832875 propositional variables (worst case):
  - # of assignments to $> 2.8 \times 10^6$ variables: $\gg 10^{840000}$ ‼
  - **Obs:** SAT solvers at present (but formula dependent)

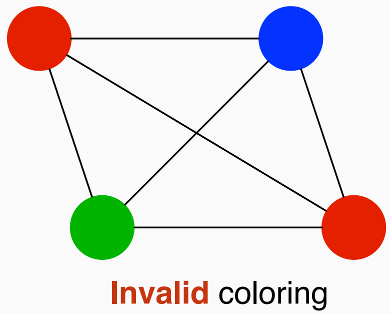
Simple modeling example – graph coloring

- Given undirected graph $G = (V, E)$ and k colors:
  - Can we assign colors to vertices of G s.t. any pair of adjacent vertices are assigned different colors?

Simple modeling example – graph coloring

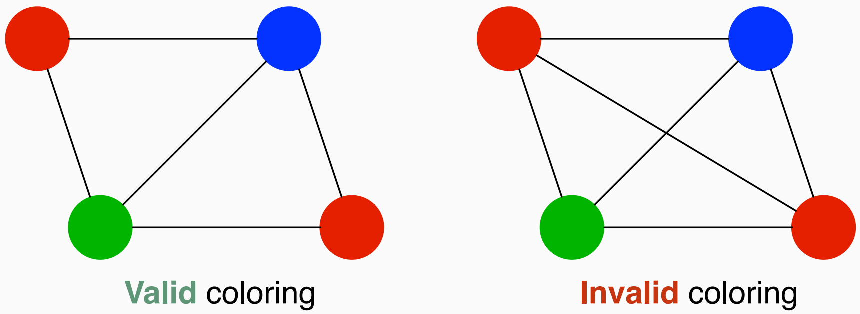
- Given undirected graph $G = (V, E)$ and k colors:
  - Can we assign colors to vertices of G s.t. any pair of adjacent vertices are assigned different colors?





Simple modeling example – graph coloring

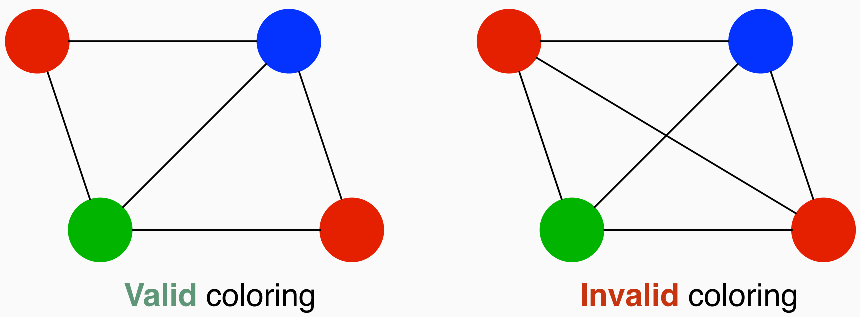
- Given undirected graph $G = (V, E)$ and k colors:
    - Can we assign colors to vertices of G s.t. any pair of adjacent vertices are assigned different colors?



- How to model color assignments to vertices?

Simple modeling example – graph coloring

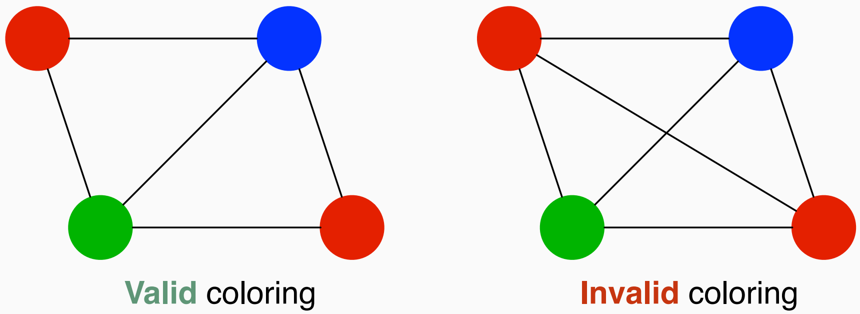
- Given undirected graph $G = (V, E)$ and k colors:
  - Can we assign colors to vertices of G s.t. any pair of adjacent vertices are assigned different colors?



- How to model color assignments to vertices?
  - $x_{i,j} = 1$ **iff vertex** $v_i \in V$ **is assigned color** $j \in \{1, \ldots, k\}$

Simple modeling example – graph coloring

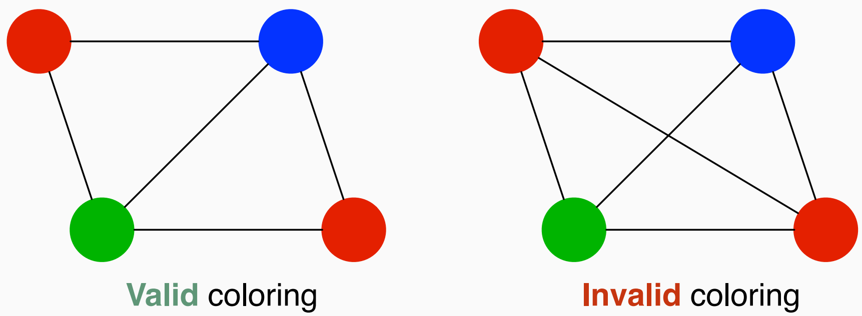
- Given undirected graph $G = (V, E)$ and k colors:
  - Can we assign colors to vertices of G s.t. any pair of adjacent vertices are assigned different colors?



- How to model color assignments to vertices?
  - $x_{i,j} = 1$ **iff vertex** $v_i \in V$ **is assigned color** $j \in \{1, \ldots, k\}$
- How to model adjacent vertices with different colors?

Simple modeling example – graph coloring

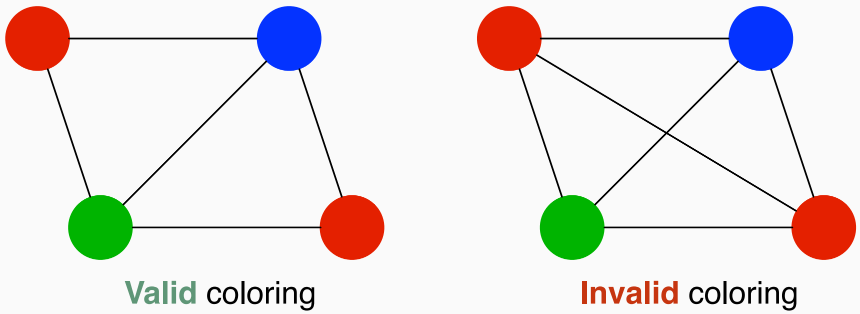
- Given undirected graph $G = (V, E)$ and k colors:
  - Can we assign colors to vertices of G s.t. any pair of adjacent vertices are assigned different colors?



- How to model color assignments to vertices?
  - $x_{i,j} = 1$ **iff vertex** $v_i \in V$ **is assigned color** $j \in \{1, \ldots, k\}$
- How to model adjacent vertices with different colors?
  - $(\neg x_{i,j} \vee \neg x_{l,j})$ **if** $(v_i, v_l) \in E$**, with** $j \in \{1, \ldots, k\}$

Simple modeling example – graph coloring

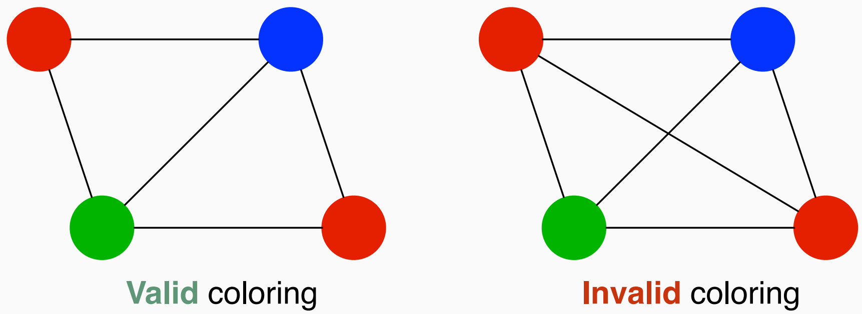
- Given undirected graph $G = (V, E)$ and k colors:
  - Can we assign colors to vertices of G s.t. any pair of adjacent vertices are assigned different colors?



- How to model color assignments to vertices?
  - $x_{i,j} = 1$ **iff vertex** $v_i \in V$ **is assigned color** $j \in \{1, \ldots, k\}$
- How to model adjacent vertices with different colors?
  - $(\neg x_{i,j} \vee \neg x_{l,j})$ **if** $(v_i, v_l) \in E$**, with** $j \in \{1, \ldots, k\}$
- How to model vertices get some color?

Simple modeling example – graph coloring

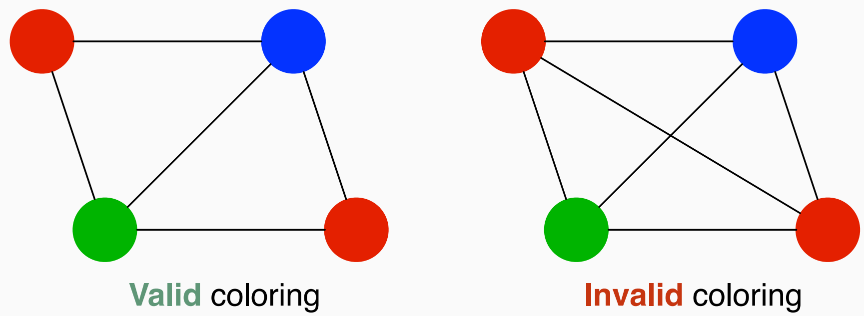
- Given undirected graph $G = (V, E)$ and k colors:
  - Can we assign colors to vertices of G s.t. any pair of adjacent vertices are assigned different colors?



- How to model color assignments to vertices?
  - $x_{i,j} = 1$ **iff vertex** $v_i \in V$ **is assigned color** $j \in \{1, \ldots, k\}$
- How to model adjacent vertices with different colors?
  - $(\neg x_{i,j} \vee \neg x_{l,j})$ **if** $(v_i, v_l) \in E$**, with** $j \in \{1, \ldots, k\}$
- How to model vertices get some color?
  - $\sum_{j\in\{1,\ldots,k\}} x_{i,j} = 1$**, for** $v_i \in V$

Simple modeling example – graph coloring

- Given undirected graph $G = (V, E)$ and k colors:
  - Can we assign colors to vertices of G s.t. any pair of adjacent vertices are assigned different colors?



- How to model color assignments to vertices?
  - $x_{i,j} = 1$ **iff vertex** $v_i \in V$ **is assigned color** $j \in \{1, \ldots, k\}$
- How to model adjacent vertices with different colors?
  - $(\neg x_{i,j} \vee \neg x_{l,j})$ **if** $(v_i, v_l) \in E$**, with** $j \in \{1, \ldots, k\}$
- How to model vertices get some color?
  - $\sum_{j\in\{1,\ldots,k\}} x_{i,j} = 1$**, for** $v_i \in V$
  - **Note: it suffices to use** $\left(\bigvee_{j\in\{1,\ldots,k\}} x_{i,j}\right)$

Optimization with maximum satisfiability (MaxSAT)

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\neg x_1$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$x_7 \vee x_5$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

- Unsatisfiable formula

Optimization with maximum satisfiability (MaxSAT)

$\boxed{x_6 \vee x_2}$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\neg x_1$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\boxed{\neg x_4 \vee x_5}$
$\boxed{x_7 \vee x_5}$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

- Unsatisfiable formula
- Find largest subset of clauses that is satisfiable

Optimization with maximum satisfiability (MaxSAT)

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\boxed{\neg x_1}$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$\boxed{x_7 \vee x_5}$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

- Unsatisfiable formula
- Find largest subset of clauses that is satisfiable
- A Minimal Correction Subset (**MCS**) is an irreducible relaxation of the formula

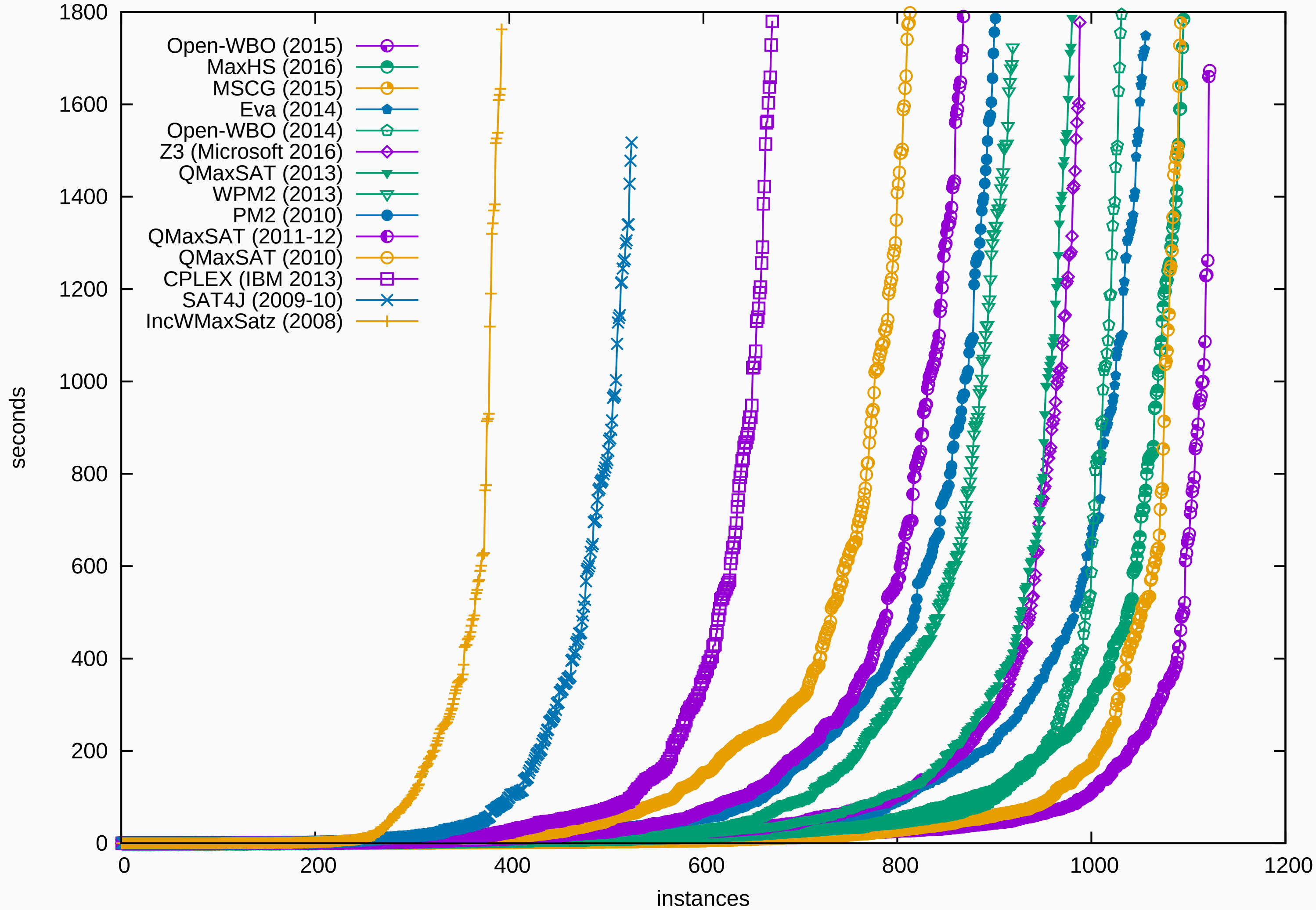
Optimization with maximum satisfiability (MaxSAT)

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\boxed{\neg x_1}$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$\boxed{x_7 \vee x_5}$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

- Unsatisfiable formula
- Find largest subset of clauses that is satisfiable
- A Minimal Correction Subset (**MCS**) is an irreducible relaxation of the formula
- The MaxSAT solution is one of the smallest (cost) MCSes

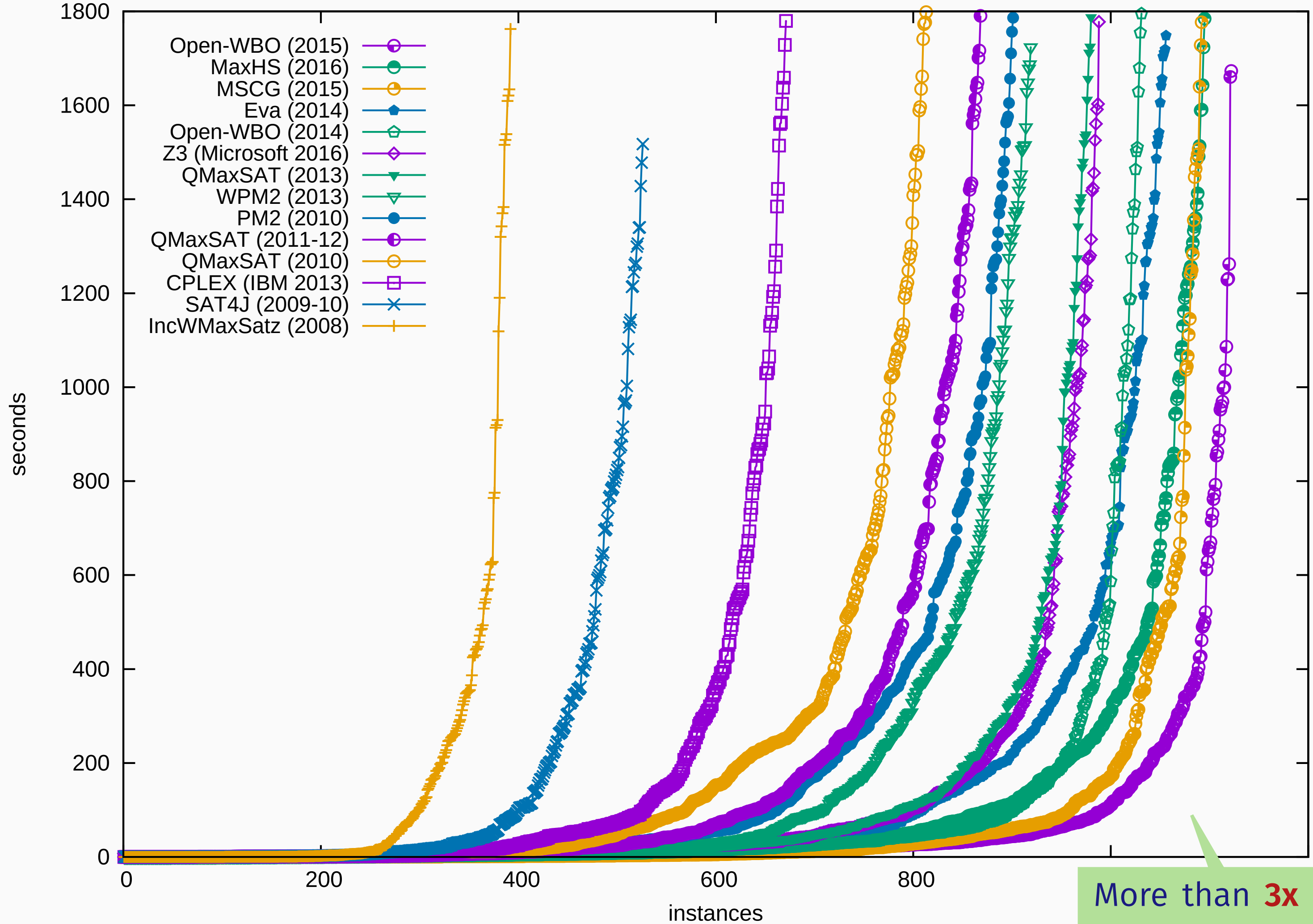
The MaxSAT (r)evolution

The MaxSAT (r)evolution – partial MaxSAT



Source: [2018 MaxSAT Eval. organizers]

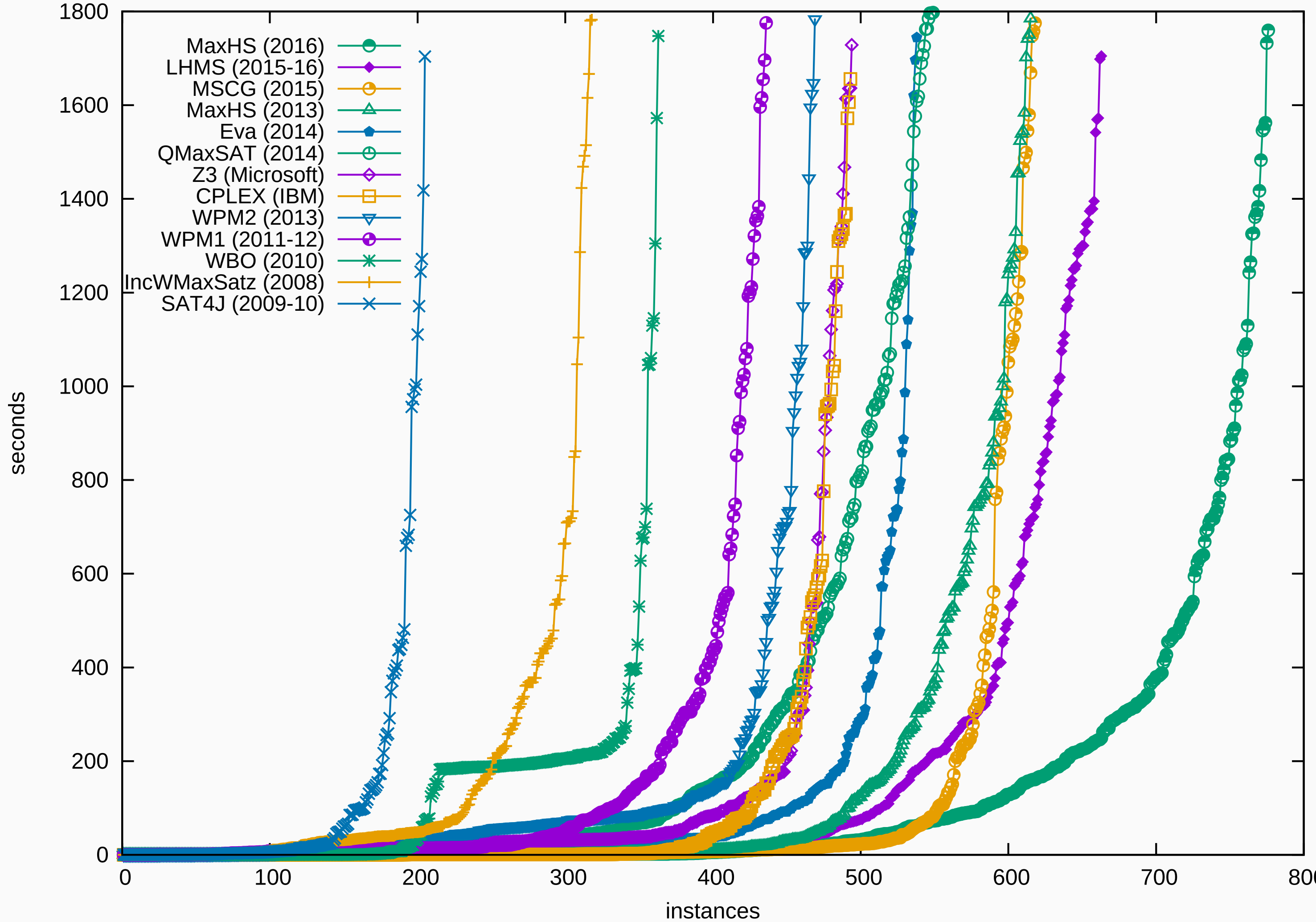
The MaxSAT (r)evolution – partial MaxSAT



More than **3x** more instances solved ‼

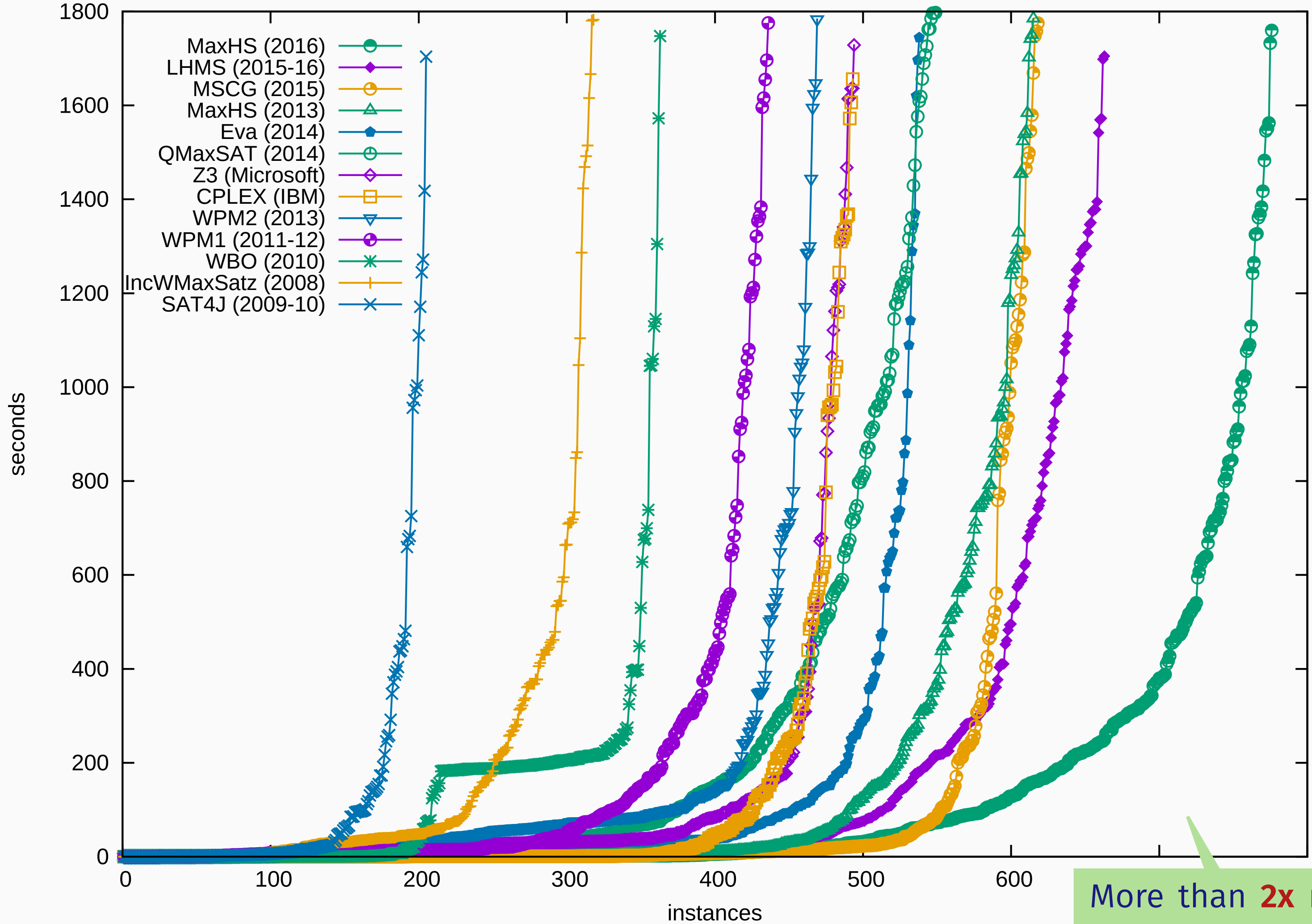
Source: [2018 MaxSAT Eval. organizers]

The MaxSAT (r)evolution – weighted MaxSAT



Source: [2018 MaxSAT Eval. organizers]

The MaxSAT (r)evolution – weighted MaxSAT

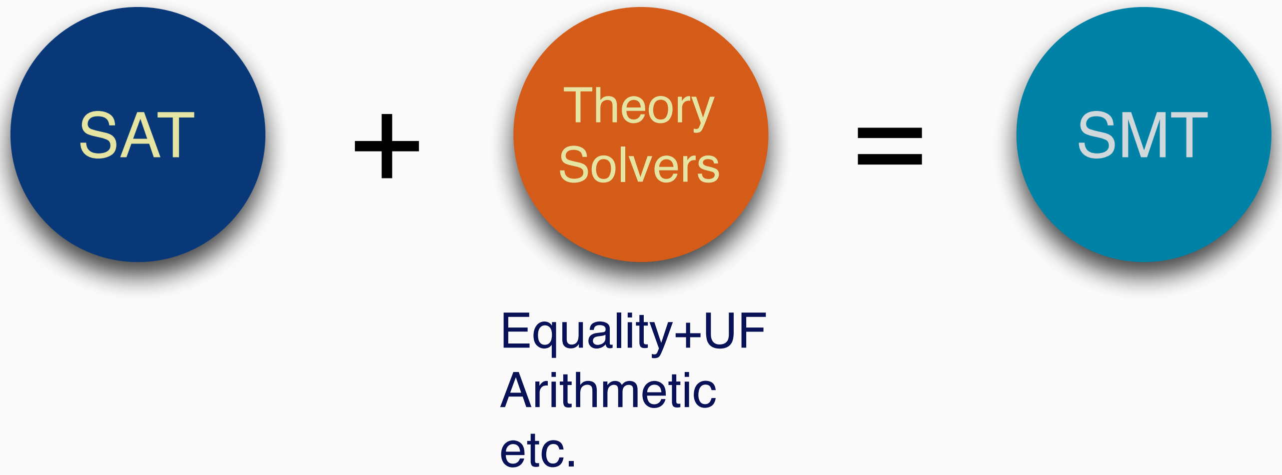


More than **2x** more instances solved ‼

Source: [2018 MaxSAT Eval. organizers]

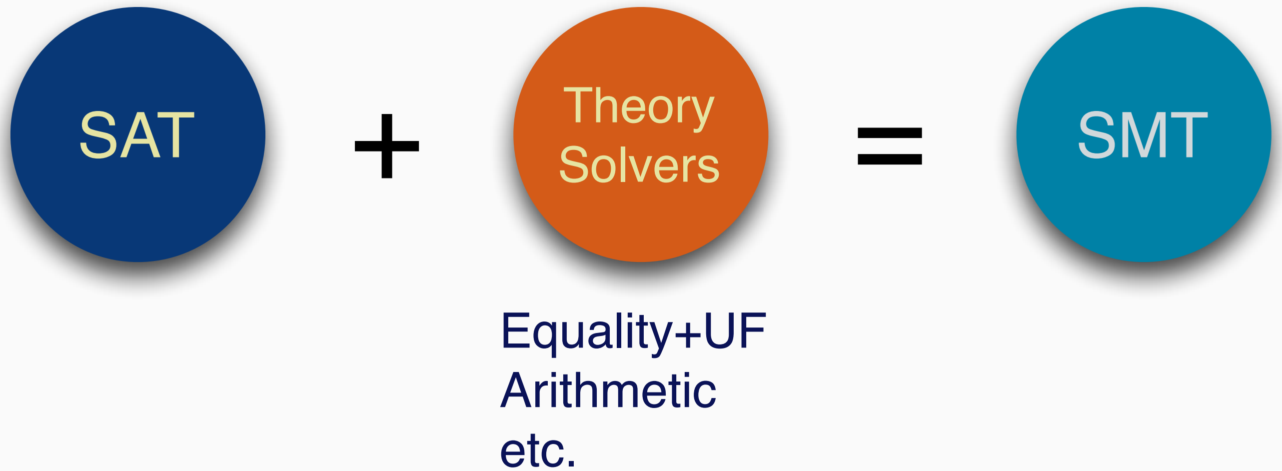
Satisfiability Modulo Theories (SMT)

- Automate reasoning in (fragments of) first-order logic (FOL)



Satisfiability Modulo Theories (SMT)

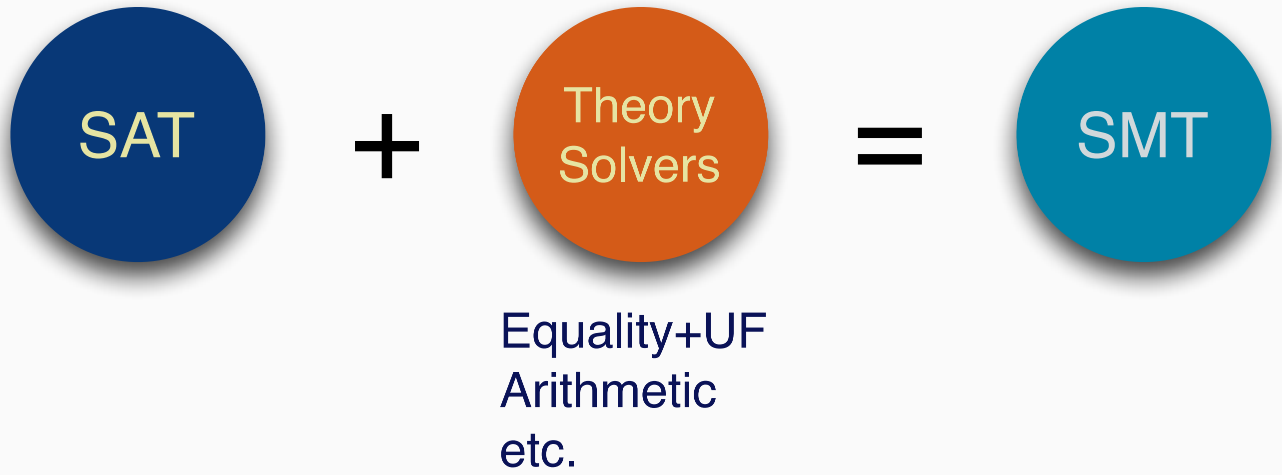
- Automate reasoning in (fragments of) first-order logic (FOL)



- Problem representation in propositional logic (PL):
    - Positive: Efficient (in practice) SAT algorithms
    - Negative: Expresiveness via CNF encodings

Satisfiability Modulo Theories (SMT)

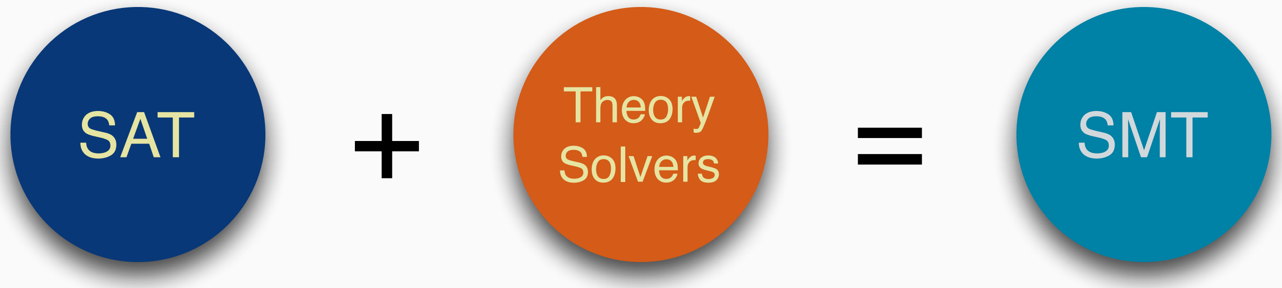
- Automate reasoning in (fragments of) first-order logic (FOL)



- Problem representation in propositional logic (PL):
    - Positive: Efficient (in practice) SAT algorithms
    - Negative: Expresiveness via CNF encodings
- PL + domain-specific reasoning
    - Positive: Improved expressiveness
    - Negative: Can be (far) less efficient than SAT

Satisfiability Modulo Theories (SMT)

- Automate reasoning in (fragments of) first-order logic (FOL)



Equality+UF
Arithmetic
etc.

- Problem representation in propositional logic (PL):
  - Positive: Efficient (in practice) SAT algorithms
  - Negative: Expresiveness via CNF encodings
- PL + domain-specific reasoning
  - Positive: Improved expressiveness
  - Negative: Can be (far) less efficient than SAT
- **Note:** Standard definitions of FOL apply

An example

- All x_i variables integer

An example

- All x_i variables integer
- Solve:

$$
\begin{aligned}
&((x_4 - x_2 \leq 3) \vee (x_4 - x_3 \geq 5)) \wedge (x_4 - x_3 \leq 6) \wedge \\
&(x_1 - x_2 \leq -1) \wedge (x_1 - x_3 \leq -2) \wedge (x_1 - x_4 \leq -1) \wedge (x_2 - x_1 \leq 2) \wedge \\
&(x_3 - x_2 \leq -1) \wedge ((x_3 - x_4 \leq -2) \vee (x_4 - x_3 \geq 2))
\end{aligned}
$$

An example

- All x_i variables integer
- Solve:

$$
\begin{aligned}
&((x_4 - x_2 \leq 3) \vee (x_4 - x_3 \geq 5)) \wedge (x_4 - x_3 \leq 6) \wedge \\
&(x_1 - x_2 \leq -1) \wedge (x_1 - x_3 \leq -2) \wedge (x_1 - x_4 \leq -1) \wedge (x_2 - x_1 \leq 2) \wedge \\
&(x_3 - x_2 \leq -1) \wedge ((x_3 - x_4 \leq -2) \vee (x_4 - x_3 \geq 2))
\end{aligned}
$$

- Integer difference logic (with Boolean structure)

An example

- All x_i variables integer
- Solve:

$$
\begin{aligned}
&((x_4 - x_2 \le 3) \vee (x_4 - x_3 \ge 5)) \wedge (x_4 - x_3 \le 6) \wedge \\
&(x_1 - x_2 \le -1) \wedge (x_1 - x_3 \le -2) \wedge (x_1 - x_4 \le -1) \wedge (x_2 - x_1 \le 2) \wedge \\
&(x_3 - x_2 \le -1) \wedge ((x_3 - x_4 \le -2) \vee (x_4 - x_3 \ge 2))
\end{aligned}
$$

- Integer difference logic (with Boolean structure)
- Unsatisfiable (**Why?**)

Another example

- All $t_{i,j}$ variables integer

Another example

- All $t_{i,j}$ variables integer
- Solve:

$$
\begin{aligned}
&(t_{1,1} \geq 0) \wedge (t_{1,2} \geq t_{1,1} + 2) \wedge (t_{1,2} + 1 \leq 8) \wedge \\
&(t_{2,1} \geq 0) \wedge (t_{2,2} \geq t_{1,1} + 3) \wedge (t_{2,2} + 1 \leq 8) \wedge \\
&(t_{3,1} \geq 0) \wedge (t_{3,2} \geq t_{1,1} + 2) \wedge (t_{3,2} + 3 \leq 8) \wedge \\
&((t_{1,1} \geq t_{2,1} + 3) \vee (t_{2,1} \geq t_{1,1} + 2)) \wedge \\
&((t_{1,1} \geq t_{3,1} + 2) \vee (t_{3,1} \geq t_{1,1} + 2)) \wedge \\
&((t_{2,1} \geq t_{3,1} + 2) \vee (t_{3,1} \geq t_{2,1} + 3)) \wedge \\
&((t_{1,2} \geq t_{2,2} + 1) \vee (t_{2,2} \geq t_{1,2} + 1)) \wedge \\
&((t_{1,2} \geq t_{3,2} + 3) \vee (t_{3,2} \geq t_{1,2} + 1)) \wedge \\
&((t_{2,2} \geq t_{3,2} + 3) \vee (t_{3,2} \geq t_{2,2} + 1))
\end{aligned}
$$

Another example

- All $t_{i,j}$ variables integer
- Solve:

$$
\begin{aligned}
&(t_{1,1} \geq 0) \wedge (t_{1,2} \geq t_{1,1} + 2) \wedge (t_{1,2} + 1 \leq 8) \wedge \\
&(t_{2,1} \geq 0) \wedge (t_{2,2} \geq t_{1,1} + 3) \wedge (t_{2,2} + 1 \leq 8) \wedge \\
&(t_{3,1} \geq 0) \wedge (t_{3,2} \geq t_{1,1} + 2) \wedge (t_{3,2} + 3 \leq 8) \wedge \\
&((t_{1,1} \geq t_{2,1} + 3) \vee (t_{2,1} \geq t_{1,1} + 2)) \wedge \\
&((t_{1,1} \geq t_{3,1} + 2) \vee (t_{3,1} \geq t_{1,1} + 2)) \wedge \\
&((t_{2,1} \geq t_{3,1} + 2) \vee (t_{3,1} \geq t_{2,1} + 3)) \wedge \\
&((t_{1,2} \geq t_{2,2} + 1) \vee (t_{2,2} \geq t_{1,2} + 1)) \wedge \\
&((t_{1,2} \geq t_{3,2} + 3) \vee (t_{3,2} \geq t_{1,2} + 1)) \wedge \\
&((t_{2,2} \geq t_{3,2} + 3) \vee (t_{3,2} \geq t_{2,2} + 1))
\end{aligned}
$$

- Another example of integer difference logic (with Boolean structure)

Another example

- All $t_{i,j}$ variables integer
- Solve:

$$
\begin{aligned}
&(t_{1,1} \geq 0) \wedge (t_{1,2} \geq t_{1,1} + 2) \wedge (t_{1,2} + 1 \leq 8) \wedge \\
&(t_{2,1} \geq 0) \wedge (t_{2,2} \geq t_{1,1} + 3) \wedge (t_{2,2} + 1 \leq 8) \wedge \\
&(t_{3,1} \geq 0) \wedge (t_{3,2} \geq t_{1,1} + 2) \wedge (t_{3,2} + 3 \leq 8) \wedge \\
&((t_{1,1} \geq t_{2,1} + 3) \vee (t_{2,1} \geq t_{1,1} + 2)) \wedge \\
&((t_{1,1} \geq t_{3,1} + 2) \vee (t_{3,1} \geq t_{1,1} + 2)) \wedge \\
&((t_{2,1} \geq t_{3,1} + 2) \vee (t_{3,1} \geq t_{2,1} + 3)) \wedge \\
&((t_{1,2} \geq t_{2,2} + 1) \vee (t_{2,2} \geq t_{1,2} + 1)) \wedge \\
&((t_{1,2} \geq t_{3,2} + 3) \vee (t_{3,2} \geq t_{1,2} + 1)) \wedge \\
&((t_{2,2} \geq t_{3,2} + 3) \vee (t_{3,2} \geq t_{2,2} + 1))
\end{aligned}
$$

- Another example of integer difference logic (with Boolean structure)
- Satisfiable, with model: $t_{1,1} = 5; t_{1,2} = 7; t_{2,1} = 2; t_{2,2} = 6; t_{3,1} = 0; t_{3,2} = 7;$

Additional formalisms & reasoners

- (Mixed) integer linear programming (MILP)

$$\begin{array}{lll} \text{min} & \sum c_j x_j & \\ st & \sum a_{ij} x_j \leq b_i & i = 1, \ldots, M \\ & x_j \in \mathbb{Z} & j = 1, \ldots, N \end{array}$$

  - Significant performance gains since the 90s

Additional formalisms & reasoners

- (Mixed) integer linear programming (MILP)

$$
\begin{array}{lll}
\text{min} & \sum c_j x_j & \\
st & \sum a_{ij} x_j \leq b_i & i = 1, \ldots, M \\
 & x_j \in \mathbb{Z} & j = 1, \ldots, N
\end{array}
$$

  - Significant performance gains since the 90s
- Constraint programming (CP)
  - Significant performance gains since the late 90s
- Answer set programming (ASP)
  - Significant performance gains since the late 90s

Additional formalisms & reasoners

- (Mixed) integer linear programming (MILP)

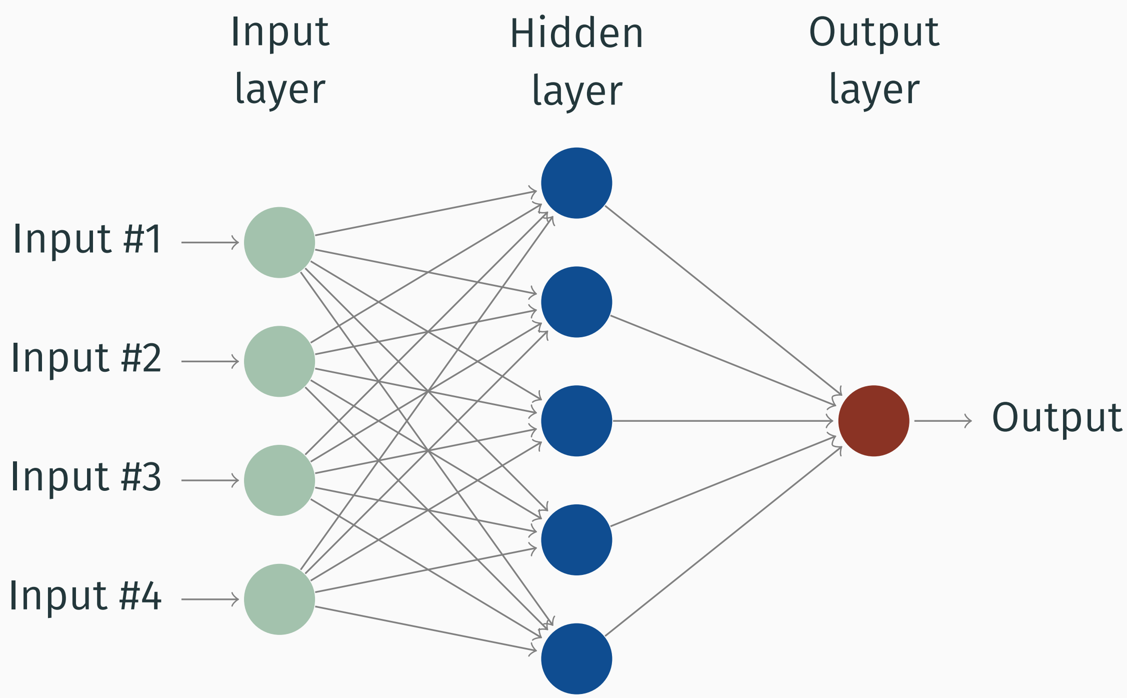
$$\begin{array}{lll} \text{min} & \sum c_j x_j & \\ st & \sum a_{ij} x_j \leq b_i & i = 1, \ldots, M \\ & x_j \in \mathbb{Z} & j = 1, \ldots, N \end{array}$$

  - Significant performance gains since the 90s
- Constraint programming (CP)
  - Significant performance gains since the late 90s
- Answer set programming (ASP)
  - Significant performance gains since the late 90s
- Quantified boolean formulas (QBF)
  - Significant performance gains over the last decade

2 Modeling Examples

How to encode a neural network?

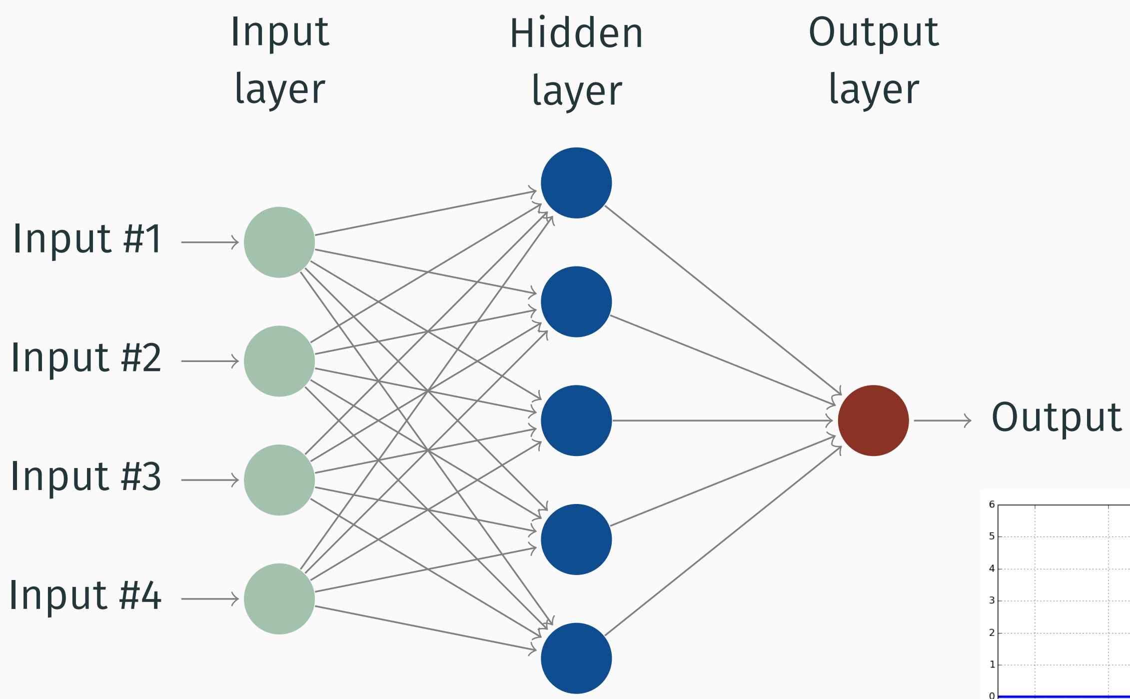
How to encode a neural network, with SMT/(M)ILP?

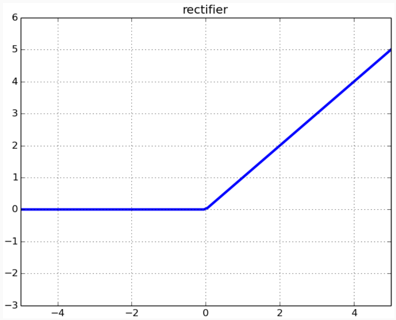


- Each layer (except first) viewed as a **block**
  - Compute $\mathbf{x}'$ given input $\mathbf{x}$, weights matrix $\mathbf{A}$, and bias vector $\mathbf{b}$
  - Compute output $\mathbf{y}$ given $\mathbf{x}'$ and activation function
- Each unit uses a ReLU activation function

[NH10]

How to encode a neural network, with SMT/(M)ILP?





- Each layer (except first) viewed as a **block**
  - Compute $\mathbf{x}'$ given input $\mathbf{x}$, weights matrix $\mathbf{A}$, and bias vector $\mathbf{b}$
  - Compute output $\mathbf{y}$ given $\mathbf{x}'$ and activation function
- Each unit uses a ReLU activation function

[NH10]

How to encode a neural network, with SMT/(M)ILP?

Computation for a NN ReLU **block**:

$$\mathbf{x}' = \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$$
$$\mathbf{y} = \max(\mathbf{x}', \mathbf{0})$$

How to encode a neural network, with SMT/(M)ILP?

Computation for a NN ReLU **block**:

$$\mathbf{x}' = \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$$
$$\mathbf{y} = \max(\mathbf{x}', \mathbf{0})$$

Encoding each **block**: [FJ18]

$$\sum_{j=1}^{n} a_{i,j} x_j + b_i = y_i - s_i$$
$$z_i = 1 \rightarrow y_i \leq 0$$
$$z_i = 0 \rightarrow s_i \leq 0$$
$$y_i \geq 0, s_i \geq 0, z_i \in \{0, 1\}$$

Simpler encodings exist, but **not** as effective [KBD+17]

How to encode cardinality constraints, with prop. logic?

- General form: $\sum_{j=1}^{n} x_j \leq k$

How to encode cardinality constraints, with prop. logic?

- General form: $\sum_{j=1}^{n} x_j \leq k$
- Any combination of $k+1$ true variables is **disallowed**

How to encode cardinality constraints, with prop. logic?

- General form: $\sum_{j=1}^{n} x_j \leq k$
- Any combination of $k + 1$ true variables is **disallowed**
- E.g. encode to propositional logic: $a + b + c + d + e \leq 2$

How to encode cardinality constraints, with prop. logic?

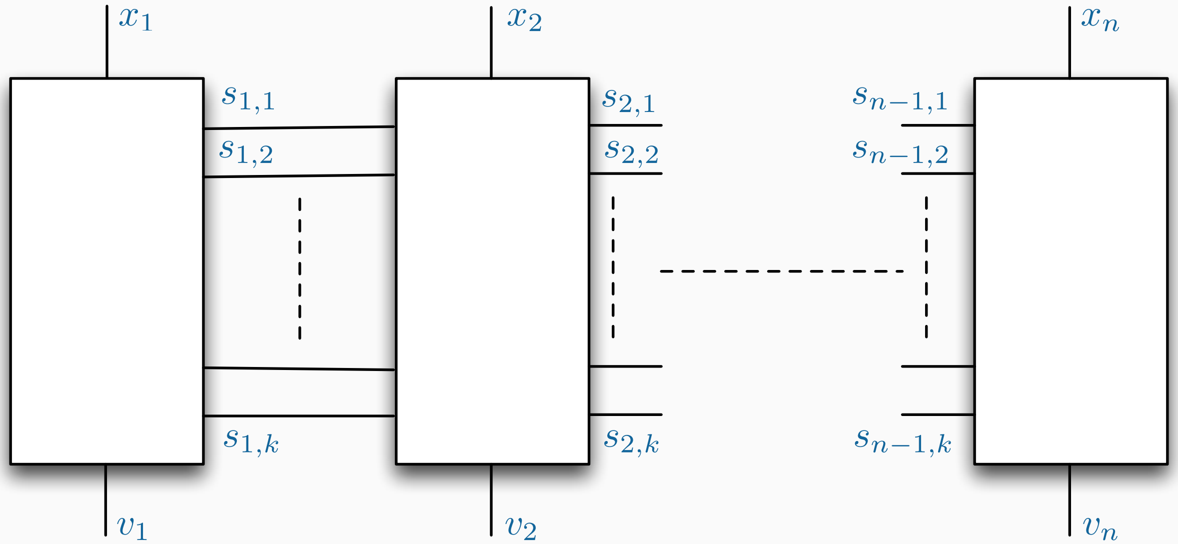
- General form: $\sum_{j=1}^{n} x_j \leq k$
- Any combination of $k+1$ true variables is **disallowed**
- E.g. encode to propositional logic: $a+b+c+d+e \leq 2$
- Resulting constraints:

$$
\begin{aligned}
a \wedge b \rightarrow \bar{c} &\quad\Longrightarrow\quad (\bar{a} \vee \bar{b} \vee \bar{c})\\
a \wedge b \rightarrow \bar{d} &\quad\Longrightarrow\quad (\bar{a} \vee \bar{b} \vee \bar{d})\\
a \wedge b \rightarrow \bar{e} &\quad\Longrightarrow\quad (\bar{a} \vee \bar{b} \vee \bar{e})\\
a \wedge c \rightarrow \bar{d} &\quad\Longrightarrow\quad (\bar{a} \vee \bar{c} \vee \bar{d})\\
a \wedge c \rightarrow \bar{e} &\quad\Longrightarrow\quad (\bar{a} \vee \bar{c} \vee \bar{e})\\
a \wedge d \rightarrow \bar{e} &\quad\Longrightarrow\quad (\bar{a} \vee \bar{d} \vee \bar{e})\\
b \wedge c \rightarrow \bar{d} &\quad\Longrightarrow\quad (\bar{b} \vee \bar{c} \vee \bar{d})\\
b \wedge c \rightarrow \bar{e} &\quad\Longrightarrow\quad (\bar{b} \vee \bar{c} \vee \bar{e})\\
b \wedge d \rightarrow \bar{e} &\quad\Longrightarrow\quad (\bar{b} \vee \bar{d} \vee \bar{e})\\
c \wedge d \rightarrow \bar{e} &\quad\Longrightarrow\quad (\bar{c} \vee \bar{d} \vee \bar{e})
\end{aligned}
$$

- Redundant constraints not shown

In practice, use auxiliary variables, e.g. sequential counter

- Encode $\sum_{j=1}^{n} x_j \leq k$ with sequential counter:



- Equations for each block $1 < i < n\ ,\ 1 < j < k$:

$$s_i = \sum_{j=1}^{i} x_j$$

s_i represented in unary

$$s_{i,1} = s_{i-1,1} \vee x_i$$
$$s_{i,j} = s_{i-1,j} \vee s_{i-1,j-1} \wedge x_i$$
$$v_i = (s_{i-1,k} \wedge x_i) \ = 0$$

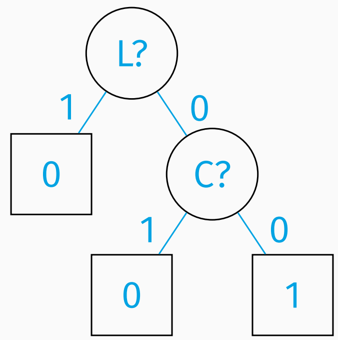
Resulting constraints

- CNF formula for $\sum_{j=1}^{n} x_j \leq k$:
  - Assume: $k > 0 \wedge n > 1$
  - Indeces: $1 < i < n \ , \ 1 < j \leq k$

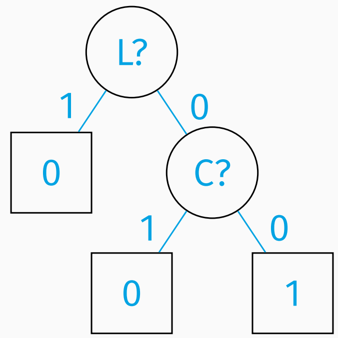
$$
\begin{aligned}
&(\neg x_1 \vee x_{1,1}) \\
&(\neg s_{1,j}) \\
&(\neg x_i \vee s_{i,1}) \\
&(\neg s_{i-1,1} \vee s_{i,1}) \\
&(\neg x_i \vee \neg s_{i-1,j-1} \vee s_{i,j}) \\
&(\neg s_{i-1,j} \vee s_{i,j}) \\
&(\neg x_i \vee \neg s_{i-1,k}) \\
&(\neg x_n \vee \neg s_{n-1,k})
\end{aligned}
$$

- $\mathcal{O}(n\ k)$ clauses & variables
- Many more encodings
- Can do pseudo-boolean constraints

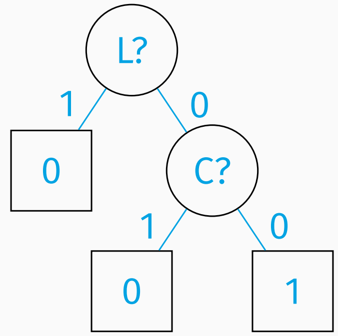
How to guess a decision tree?



How to guess a decision tree, with propositional logic?

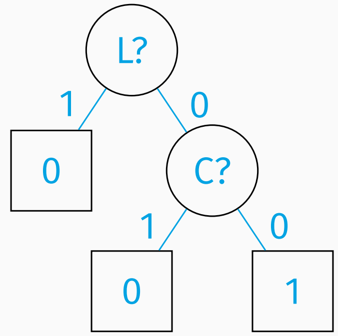


How to guess a decision tree, with propositional logic?



Var	Description
v_i	1 iff node i is a leaf node, $i = 1, \ldots, N$
l_{ij}	1 iff node i has node j as the left child, with $j \in \textsc{lr}(i)$, where $\textsc{lr}(i) = \text{even}([i+1, \min(2i, N-1)])$, $i = 1, \ldots, N$
r_{ij}	1 iff node i has node j as the right child, with $j \in \textsc{rr}(i)$, where $\textsc{rr}(i) = \text{odd}([i+2, \min(2i+1, N)])$, $i = 1, \ldots, N$
p_{ji}	1 iff the parent of node j is node i, $j = 2, \ldots, N$, $i = 1, \ldots, N-1$

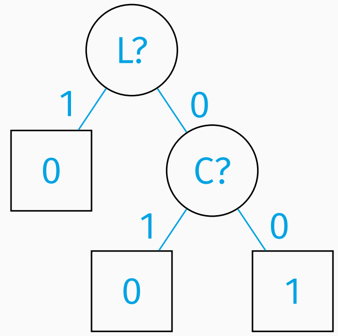
How to guess a decision tree, with propositional logic?



Var	Description
v_i	1 iff node i is a leaf node, $i = 1, \ldots, N$
l_{ij}	1 iff node i has node j as the left child, with $j \in \textsc{lr}(i)$, where $\textsc{lr}(i) = \text{even}([i+1, \min(2i, N-1)])$, $i = 1, \ldots, N$
r_{ij}	1 iff node i has node j as the right child, with $j \in \textsc{rr}(i)$, where $\textsc{rr}(i) = \text{odd}([i+2, \min(2i+1, N)])$, $i = 1, \ldots, N$
p_{ji}	1 iff the parent of node j is node i, $j = 2, \ldots, N$, $i = 1, \ldots, N-1$

Constraints:

How to guess a decision tree, with propositional logic?

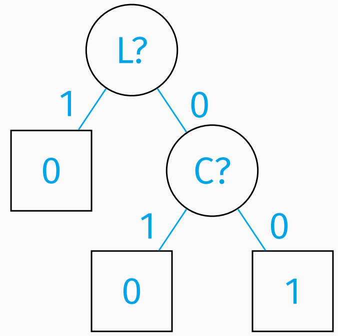


Var	Description
v_i	1 iff node i is a leaf node, $i = 1, \ldots, N$
l_{ij}	1 iff node i has node j as the left child, with $j \in \textsc{lr}(i)$, where $\textsc{lr}(i) = \text{even}([i+1, \text{min}(2i, N-1)])$, $i = 1, \ldots, N$
r_{ij}	1 iff node i has node j as the right child, with $j \in \textsc{rr}(i)$, where $\textsc{rr}(i) = \text{odd}([i+2, \text{min}(2i+1, N)])$, $i = 1, \ldots, N$
p_{ji}	1 iff the parent of node j is node i, $j = 2, \ldots, N$, $i = 1, \ldots, N-1$

Constraints:

$(\neg v_1)$ Root node is not a leaf

How to guess a decision tree, with propositional logic?



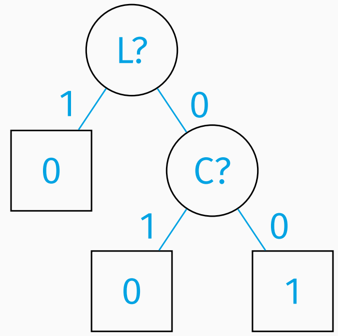
Var	Description
v_i	1 iff node i is a leaf node, $i = 1, \ldots, N$
l_{ij}	1 iff node i has node j as the left child, with $j \in \text{LR}(i)$, where $\text{LR}(i) = \text{even}([i+1, \min(2i, N-1)])$, $i = 1, \ldots, N$
r_{ij}	1 iff node i has node j as the right child, with $j \in \text{RR}(i)$, where $\text{RR}(i) = \text{odd}([i+2, \min(2i+1, N)])$, $i = 1, \ldots, N$
p_{ji}	1 iff the parent of node j is node i, $j = 2, \ldots, N$, $i = 1, \ldots, N-1$

Constraints:

$(\neg v_1)$ Root node is not a leaf

$v_i \rightarrow \neg l_{ij}, \quad j \in \text{LR}(i)$ Leaf nodes have no children

How to guess a decision tree, with propositional logic?

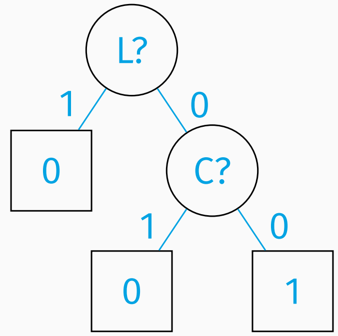


Var	Description
v_i	1 iff node i is a leaf node, $i = 1, \ldots, N$
l_{ij}	1 iff node i has node j as the left child, with $j \in \textsc{lr}(i)$, where $\textsc{lr}(i) = \text{even}([i+1, \min(2i, N-1)])$, $i = 1, \ldots, N$
r_{ij}	1 iff node i has node j as the right child, with $j \in \textsc{rr}(i)$, where $\textsc{rr}(i) = \text{odd}([i+2, \min(2i+1, N)])$, $i = 1, \ldots, N$
p_{ji}	1 iff the parent of node j is node i, $j = 2, \ldots, N$, $i = 1, \ldots, N-1$

Constraints:

$(\neg v_1)$		Root node is not a leaf
$v_i \rightarrow \neg l_{ij},$	$j \in \textsc{lr}(i)$	Leaf nodes have no children
$l_{ij} \leftrightarrow r_{ij+1},$	$j \in \textsc{lr}(i)$	Left and right children of ith node are numbered consecutively

How to guess a decision tree, with propositional logic?



Var	Description
v_i	1 iff node i is a leaf node, $i = 1, \ldots, N$
l_{ij}	1 iff node i has node j as the left child, with $j \in \textsc{lr}(i)$, where $\textsc{lr}(i) = \text{even}([i+1, \min(2i, N-1)])$, $i = 1, \ldots, N$
r_{ij}	1 iff node i has node j as the right child, with $j \in \textsc{rr}(i)$, where $\textsc{rr}(i) = \text{odd}([i+2, \min(2i+1, N)])$, $i = 1, \ldots, N$
p_{ji}	1 iff the parent of node j is node i, $j = 2, \ldots, N$, $i = 1, \ldots, N-1$

Constraints:

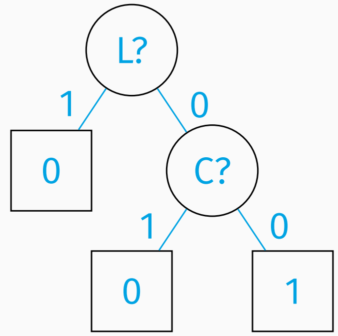
$(\neg v_1)$ — Root node is not a leaf

$v_i \rightarrow \neg l_{ij}, \quad j \in \textsc{lr}(i)$ — Leaf nodes have no children

$l_{ij} \leftrightarrow r_{ij+1}, \quad j \in \textsc{lr}(i)$ — Left and right children of ith node are numbered consecutively

$\neg v_i \rightarrow \left(\sum_{j \in \textsc{lr}(i)} l_{ij} = 1\right)$ — Non-leaf nodes must have a child

How to guess a decision tree, with propositional logic?

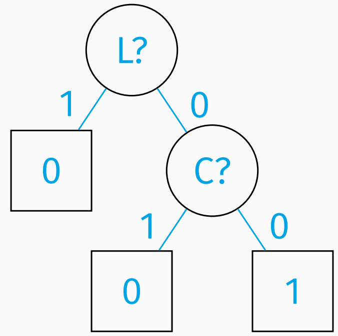


Var	Description
v_i	1 iff node i is a leaf node, $i = 1, \ldots, N$
l_{ij}	1 iff node i has node j as the left child, with $j \in \mathrm{LR}(i)$, where $\mathrm{LR}(i) = \mathrm{even}([i+1, \min(2i, N-1)])$, $i = 1, \ldots, N$
r_{ij}	1 iff node i has node j as the right child, with $j \in \mathrm{RR}(i)$, where $\mathrm{RR}(i) = \mathrm{odd}([i+2, \min(2i+1, N)])$, $i = 1, \ldots, N$
p_{ji}	1 iff the parent of node j is node i, $j = 2, \ldots, N$, $i = 1, \ldots, N-1$

Constraints:

$(\neg v_1)$	Root node is not a leaf
$v_i \rightarrow \neg l_{ij}, \quad j \in \mathrm{LR}(i)$	Leaf nodes have no children
$l_{ij} \leftrightarrow r_{ij+1}, \quad j \in \mathrm{LR}(i)$	Left and right children of ith node are numbered consecutively
$\neg v_i \rightarrow \left(\sum_{j \in \mathrm{LR}(i)} l_{ij} = 1\right)$	Non-leaf nodes must have a child
$p_{ji} \leftrightarrow l_{ij}, \quad j \in \mathrm{LR}(i); \quad p_{ji} \leftrightarrow r_{ij}, \quad j \in \mathrm{RR}(i)$	A parent node ith must have a child

How to guess a decision tree, with propositional logic?



Var	Description
v_i	1 iff node i is a leaf node, $i = 1, \ldots, N$
l_{ij}	1 iff node i has node j as the left child, with $j \in \text{LR}(i)$, where $\text{LR}(i) = \text{even}([i+1, \min(2i, N-1)])$, $i = 1, \ldots, N$
r_{ij}	1 iff node i has node j as the right child, with $j \in \text{RR}(i)$, where $\text{RR}(i) = \text{odd}([i+2, \min(2i+1, N)])$, $i = 1, \ldots, N$
p_{ji}	1 iff the parent of node j is node i, $j = 2, \ldots, N$, $i = 1, \ldots, N-1$

Constraints:

$(\neg v_1)$ — Root node is not a leaf

$v_i \rightarrow \neg l_{ij}, \quad j \in \text{LR}(i)$ — Leaf nodes have no children

$l_{ij} \leftrightarrow r_{ij+1}, \quad j \in \text{LR}(i)$ — Left and right children of ith node are numbered consecutively

$\neg v_i \rightarrow \left(\sum_{j \in \text{LR}(i)} l_{ij} = 1\right)$ — Non-leaf nodes must have a child

$p_{ji} \leftrightarrow l_{ij}, \quad j \in \text{LR}(i); \quad p_{ji} \leftrightarrow r_{ij}, \quad j \in \text{RR}(i)$ — A parent node ith must have a child

$$\left(\sum_{i=\lfloor \frac{j}{2} \rfloor}^{\min(j-1,N)} p_{ji} = 1\right) \quad \text{with } j = 2, \ldots, N$$ — A binary tree must be a tree, i.e. non-root nodes must have a parent

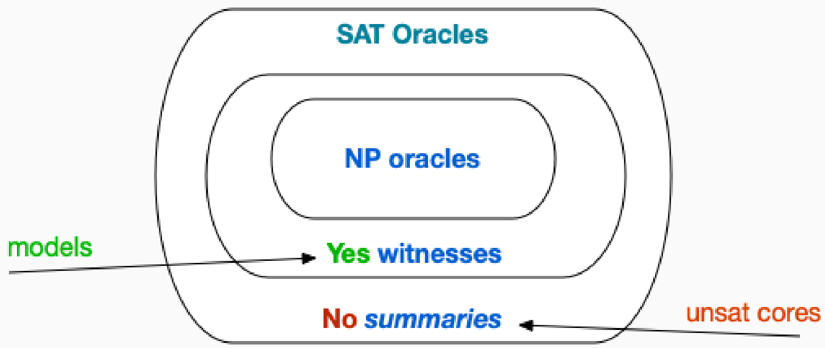
3 Basic Formal Toolbox

Oracle-based problem solving

- Many problems are **not** decision problems

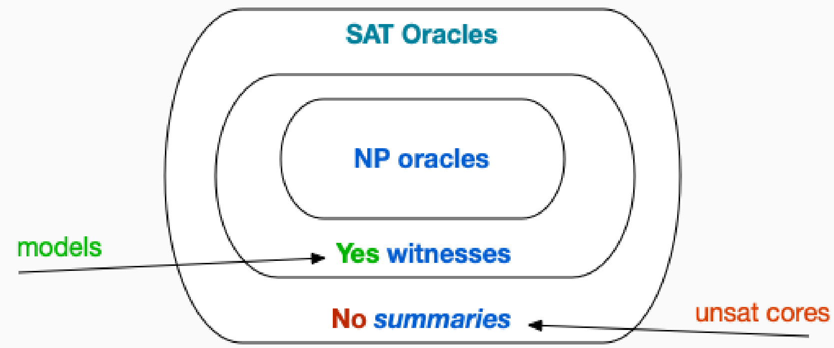
Oracle-based problem solving

- Many problems are **not** decision problems
- Use decision procedures as **oracles** for
  - Optimize some cost function
  - Find one minimal set
  - Enumerate minimal/optimal solutions
  - Other problems



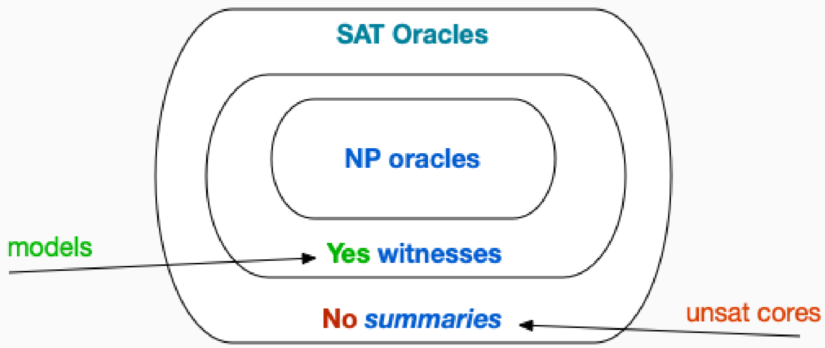
Oracle-based problem solving

- Many problems are **not** decision problems
- Use decision procedures as **oracles** for
  - Optimize some cost function
    - Maximum satisfiability (MaxSAT), pseudo-boolean optimization (PBO)
    - But also MaxSMT, etc.
  - Find one minimal set
  - Enumerate minimal/optimal solutions
  - Other problems



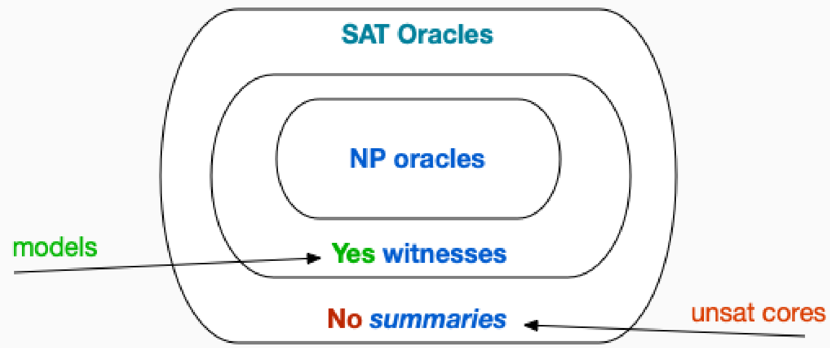
Oracle-based problem solving

- Many problems are **not** decision problems
- Use decision procedures as **oracles** for
  - Optimize some cost function
    - Maximum satisfiability (MaxSAT), pseudo-boolean optimization (PBO)
    - But also MaxSMT, etc.
  - Find one minimal set
    - Reason about inconsistency: MUSes/MCSes
    - Compile knowledge: prime implicants/implicates
  - Enumerate minimal/optimal solutions
  - Other problems



Oracle-based problem solving

- Many problems are **not** decision problems
- Use decision procedures as **oracles** for
  - Optimize some cost function
    - Maximum satisfiability (MaxSAT), pseudo-boolean optimization (PBO)
    - But also MaxSMT, etc.
  - Find one minimal set
    - Reason about inconsistency: MUSes/MCSes
    - Compile knowledge: prime implicants/implicates
  - Enumerate minimal/optimal solutions
    - Enumerate MaxSAT solutions
    - Enumerate primes, MUSes, MCSes
  - Other problems
    - Propositional abduction
    - Etc.



Analyzing inconsistency – timetabling

Subject	Day	Time	Room
Intro Prog	Mon	9:00-10:00	6.2.46
Intro AI	Tue	10:00-11:00	8.2.37
Databases	Tue	11:00-12:00	8.2.37
... (hundreds of consistent constraints)			
Linear Alg	Mon	9:00-10:00	6.2.46
Calculus	Tue	10:00-11:00	8.2.37
Adv Calculus	Mon	9:00-10:00	8.2.06
... (hundreds of consistent constraints)			

- Set of constraints consistent / satisfiable?

Analyzing inconsistency – timetabling

Subject	Day	Time	Room
Intro Prog	Mon	9:00-10:00	6.2.46
Intro AI	Tue	10:00-11:00	8.2.37
Databases	Tue	11:00-12:00	8.2.37
... (hundreds of consistent constraints)			
Linear Alg	Mon	9:00-10:00	6.2.46
Calculus	Tue	10:00-11:00	8.2.37
Adv Calculus	Mon	9:00-10:00	8.2.06
... (hundreds of consistent constraints)			

- Set of constraints consistent / satisfiable? No

Analyzing inconsistency – timetabling

Subject	Day	Time	Room
Intro Prog	Mon	9:00-10:00	6.2.46
Intro AI	Tue	10:00-11:00	8.2.37
Databases	Tue	11:00-12:00	8.2.37
... (hundreds of consistent constraints)			
Linear Alg	Mon	9:00-10:00	6.2.46
Calculus	Tue	10:00-11:00	8.2.37
Adv Calculus	Mon	9:00-10:00	8.2.06
... (hundreds of consistent constraints)			

- Set of constraints consistent / satisfiable? No
- Minimal subset of constraints that is inconsistent / unsatisfiable?

Analyzing inconsistency – timetabling

Subject	Day	Time	Room
Intro Prog	Mon	9:00-10:00	6.2.46
Intro AI	Tue	10:00-11:00	8.2.37
Databases	Tue	11:00-12:00	8.2.37
... (hundreds of consistent constraints)			
Linear Alg	Mon	9:00-10:00	6.2.46
Calculus	Tue	10:00-11:00	8.2.37
Adv Calculus	Mon	9:00-10:00	8.2.06
... (hundreds of consistent constraints)			

- Set of constraints consistent / satisfiable? No
- Minimal subset of constraints that is inconsistent / unsatisfiable?

Analyzing inconsistency – timetabling

Subject	Day	Time	Room
Intro Prog	Mon	9:00-10:00	6.2.46
Intro AI	Tue	10:00-11:00	8.2.37
Databases	Tue	11:00-12:00	8.2.37
… (hundreds of consistent constraints)			
Linear Alg	Mon	9:00-10:00	6.2.46
Calculus	Tue	10:00-11:00	8.2.37
Adv Calculus	Mon	9:00-10:00	8.2.06
… (hundreds of consistent constraints)			

- Set of constraints consistent / satisfiable? No
- Minimal subset of constraints that is inconsistent / unsatisfiable?
- Minimal subset of constraints whose removal makes remaining constraints consistent?

Analyzing inconsistency – timetabling

Subject	Day	Time	Room
Intro Prog	Mon	9:00-10:00	6.2.46
Intro AI	Tue	10:00-11:00	8.2.37
Databases	Tue	11:00-12:00	8.2.37
... (hundreds of consistent constraints)			
Linear Alg	Mon	9:00-10:00	6.2.46
Calculus	Tue	10:00-11:00	8.2.37
Adv Calculus	Mon	9:00-10:00	8.2.06
... (hundreds of consistent constraints)			

- Set of constraints consistent / satisfiable? No
- Minimal subset of constraints that is inconsistent / unsatisfiable?
- Minimal subset of constraints whose removal makes remaining constraints consistent?

Analyzing inconsistency – timetabling

Subject	Day	Time	Room
Intro Prog	Mon	9:00-10:00	6.2.46
Intro AI	Tue	10:00-11:00	8.2.37
Databases	Tue	11:00-12:00	8.2.37
... (hundreds of consistent constraints)			
Linear Alg	Mon	9:00-10:00	6.2.46
Calculus	Tue	10:00-11:00	8.2.37
Adv Calculus	Mon	9:00-10:00	8.2.06
... (hundreds of consistent constraints)			

- Set of constraints consistent / satisfiable? No
- Minimal subset of constraints that is inconsistent / unsatisfiable?
- Minimal subset of constraints whose removal makes remaining constraints consistent?

Minimality matters!

Inconsistent formulas – MUSes & MCSes

- Given $\mathcal{F}$ $(\vDash \bot)$, $\mathcal{M} \subseteq \mathcal{F}$ is a Minimal Unsatisfiable Subset (MUS) iff $\mathcal{M} \vDash \bot$ and $\forall_{\mathcal{M}' \subsetneq \mathcal{M}}, \mathcal{M}' \nvDash \bot$

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

Inconsistent formulas – MUSes & MCSes

- Given $\mathcal{F}$ $(\vDash \bot)$, $\mathcal{M} \subseteq \mathcal{F}$ is a Minimal Unsatisfiable Subset (MUS) iff $\mathcal{M} \vDash \bot$ and $\forall_{\mathcal{M}' \subsetneq \mathcal{M}}, \mathcal{M}' \nvDash \bot$

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

Inconsistent formulas – MUSes & MCSes

- Given $\mathcal{F}$ $(\models \perp)$, $\mathcal{M} \subseteq \mathcal{F}$ is a Minimal Unsatisfiable Subset (MUS) iff $\mathcal{M} \models \perp$ and $\forall_{\mathcal{M}' \subsetneq \mathcal{M}}, \mathcal{M}' \not\models \perp$

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

- Given $\mathcal{F}$ $(\models \perp)$, $\mathcal{C} \subseteq \mathcal{F}$ is a Minimal Correction Subset (MCS) iff $\mathcal{F} \setminus \mathcal{C} \not\models \perp$ and $\forall_{\mathcal{C}' \subsetneq \mathcal{C}}, \mathcal{F} \setminus \mathcal{C}' \models \perp$. $\mathcal{S} = \mathcal{F} \setminus \mathcal{C}$ is MSS

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

Inconsistent formulas – MUSes & MCSes

- Given $\mathcal{F}$ $(\vDash \bot)$, $\mathcal{M} \subseteq \mathcal{F}$ is a Minimal Unsatisfiable Subset (MUS) iff $\mathcal{M} \vDash \bot$ and $\forall_{\mathcal{M}' \subsetneq \mathcal{M}}, \mathcal{M}' \nvDash \bot$

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

- Given $\mathcal{F}$ $(\vDash \bot)$, $\mathcal{C} \subseteq \mathcal{F}$ is a Minimal Correction Subset (MCS) iff $\mathcal{F} \setminus \mathcal{C} \nvDash \bot$ and $\forall_{\mathcal{C}' \subsetneq \mathcal{C}}, \mathcal{F} \setminus \mathcal{C}' \vDash \bot$. $\mathcal{S} = \mathcal{F} \setminus \mathcal{C}$ is MSS

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

Inconsistent formulas – MUSes & MCSes

- Given $\mathcal{F}$ $(\vDash \bot)$, $\mathcal{M} \subseteq \mathcal{F}$ is a Minimal Unsatisfiable Subset (MUS) iff $\mathcal{M} \vDash \bot$ and $\forall_{\mathcal{M}' \subsetneq \mathcal{M}}, \mathcal{M}' \nvDash \bot$

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

- Given $\mathcal{F}$ $(\vDash \bot)$, $\mathcal{C} \subseteq \mathcal{F}$ is a Minimal Correction Subset (MCS) iff $\mathcal{F} \setminus \mathcal{C} \nvDash \bot$ and $\forall_{\mathcal{C}' \subsetneq \mathcal{C}}, \mathcal{F} \setminus \mathcal{C}' \vDash \bot$. $\mathcal{S} = \mathcal{F} \setminus \mathcal{C}$ is MSS

$$(\neg x_1 \vee \neg x_2) \wedge (x_1) \wedge (x_2) \wedge (\neg x_3 \vee \neg x_4) \wedge (x_3) \wedge (x_4) \wedge (x_5 \vee x_6)$$

- MUSes and MCSes are (subset-)**minimal** sets
- MUSes and minimal hitting sets of MCSes and vice-versa

[Rei87, BS05]

Basic MUS extraction

Input : Set $\mathcal{F}$
Output: Minimal subset $\mathcal{M}$
begin
  $\mathcal{M} \leftarrow \mathcal{F}$
  foreach $c \in \mathcal{M}$ **do**
    if $\neg\text{SAT}(\mathcal{M} \setminus \{c\})$ **then**
      $\mathcal{M} \leftarrow \mathcal{M} \setminus \{c\}$ // If $\neg\text{SAT}(\mathcal{M} \setminus \{c\})$, then $c \notin$ MUS
  return $\mathcal{M}$ // Final $\mathcal{M}$ is MUS
end

- Number of oracles calls: $\mathcal{O}(m)$ [CD91, BDTW93]

Basic MUS extraction

Input : Set $\mathcal{F}$
Output: Minimal subset $\mathcal{M}$
begin
  $\mathcal{M} \leftarrow \mathcal{F}$
  foreach $c \in \mathcal{M}$ **do**
    if $\neg\text{SAT}(\mathcal{M} \setminus \{c\})$ **then**
      $\mathcal{M} \leftarrow \mathcal{M} \setminus \{c\}$ // Remove c from $\mathcal{M}$
  return $\mathcal{M}$ // Final $\mathcal{M}$ is MUS
end

Monotonicity implicit & essential!

- Number of oracles calls: $\mathcal{O}(m)$ [CD91, BDTW93]

An example

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_2)	$(\neg x_3 \vee \neg x_4)$	(x_3)	(x_4)	$(x_5 \vee x_6)$

$\mathcal{M}$	$\mathcal{M} \setminus \{c\}$	$\neg\mathrm{SAT}(\mathcal{M} \setminus \{c\})$	Outcome

An example

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_2)	$(\neg x_3 \vee \neg x_4)$	(x_3)	(x_4)	$(x_5 \vee x_6)$

$\mathcal{M}$	$\mathcal{M} \setminus \{c\}$	$\neg$SAT$(\mathcal{M} \setminus \{c\})$	Outcome
$c_1..c_7$	$c_2..c_7$	1	Drop c_1

An example

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_2)	$(\neg x_3 \vee \neg x_4)$	(x_3)	(x_4)	$(x_5 \vee x_6)$

$\mathcal{M}$	$\mathcal{M} \setminus \{c\}$	$\neg\mathsf{SAT}(\mathcal{M} \setminus \{c\})$	Outcome
$c_1..c_7$	$c_2..c_7$	1	Drop c_1
$c_2..c_7$	$c_3..c_7$	1	Drop c_2

An example

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_2)	$(\neg x_3 \vee \neg x_4)$	(x_3)	(x_4)	$(x_5 \vee x_6)$

$\mathcal{M}$	$\mathcal{M} \setminus \{c\}$	$\neg\mathsf{SAT}(\mathcal{M} \setminus \{c\})$	Outcome
$c_1..c_7$	$c_2..c_7$	1	Drop c_1
$c_2..c_7$	$c_3..c_7$	1	Drop c_2
$c_3..c_7$	$c_4..c_7$	1	Drop c_3

An example

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_2)	$(\neg x_3 \vee \neg x_4)$	(x_3)	(x_4)	$(x_5 \vee x_6)$

$\mathcal{M}$	$\mathcal{M} \setminus \{c\}$	$\neg$SAT$(\mathcal{M} \setminus \{c\})$	Outcome
$c_1..c_7$	$c_2..c_7$	1	Drop c_1
$c_2..c_7$	$c_3..c_7$	1	Drop c_2
$c_3..c_7$	$c_4..c_7$	1	Drop c_3
$c_4..c_7$	$c_5..c_7$	0	Keep c_4

An example

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_2)	$(\neg x_3 \vee \neg x_4)$	(x_3)	(x_4)	$(x_5 \vee x_6)$

$\mathcal{M}$	$\mathcal{M} \setminus \{c\}$	$\neg\mathrm{SAT}(\mathcal{M} \setminus \{c\})$	Outcome
$c_1..c_7$	$c_2..c_7$	1	Drop c_1
$c_2..c_7$	$c_3..c_7$	1	Drop c_2
$c_3..c_7$	$c_4..c_7$	1	Drop c_3
$c_4..c_7$	$c_5..c_7$	0	Keep c_4
$c_4..c_7$	$c_4 c_6 c_7$	0	Keep c_5

An example

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_2)	$(\neg x_3 \vee \neg x_4)$	(x_3)	(x_4)	$(x_5 \vee x_6)$

$\mathcal{M}$	$\mathcal{M} \setminus \{c\}$	$\neg\mathrm{SAT}(\mathcal{M} \setminus \{c\})$	Outcome
$c_1..c_7$	$c_2..c_7$	1	Drop c_1
$c_2..c_7$	$c_3..c_7$	1	Drop c_2
$c_3..c_7$	$c_4..c_7$	1	Drop c_3
$c_4..c_7$	$c_5..c_7$	0	Keep c_4
$c_4..c_7$	$c_4 c_6 c_7$	0	Keep c_5
$c_4..c_7$	$c_4 c_5 c_7$	0	Keep c_6

An example

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_2)	$(\neg x_3 \vee \neg x_4)$	(x_3)	(x_4)	$(x_5 \vee x_6)$

$\mathcal{M}$	$\mathcal{M} \setminus \{c\}$	$\neg$SAT$(\mathcal{M} \setminus \{c\})$	Outcome
$c_1..c_7$	$c_2..c_7$	1	Drop c_1
$c_2..c_7$	$c_3..c_7$	1	Drop c_2
$c_3..c_7$	$c_4..c_7$	1	Drop c_3
$c_4..c_7$	$c_5..c_7$	0	Keep c_4
$c_4..c_7$	$c_4c_6c_7$	0	Keep c_5
$c_4..c_7$	$c_4c_5c_7$	0	Keep c_6
$c_4..c_7$	$c_4..c_6$	1	Drop c_7

An example

c_1	c_2	c_3	c_4	c_5	c_6	c_7
$(\neg x_1 \vee \neg x_2)$	(x_1)	(x_2)	$(\neg x_3 \vee \neg x_4)$	(x_3)	(x_4)	$(x_5 \vee x_6)$

$\mathcal{M}$	$\mathcal{M} \setminus \{c\}$	$\neg\text{SAT}(\mathcal{M} \setminus \{c\})$	Outcome
$c_1..c_7$	$c_2..c_7$	1	Drop c_1
$c_2..c_7$	$c_3..c_7$	1	Drop c_2
$c_3..c_7$	$c_4..c_7$	1	Drop c_3
$c_4..c_7$	$c_5..c_7$	0	Keep c_4
$c_4..c_7$	$c_4 c_6 c_7$	0	Keep c_5
$c_4..c_7$	$c_4 c_5 c_7$	0	Keep c_6
$c_4..c_7$	$c_4..c_6$	1	Drop c_7

- MUS: $\{c_4, c_5, c_6\}$

Compilation – extracting & enumerating primes

- Boolean function: $\mathcal{F}$
- Set of literals of $\mathcal{F}$: $\mathbb{L}(\mathcal{F}) \triangleq \{\mathsf{x}, \neg\mathsf{x} | \mathsf{x} \in \mathsf{var}(\mathcal{F})\}$

Compilation – extracting & enumerating primes

- Boolean function: $\mathcal{F}$
- Set of literals of $\mathcal{F}$: $\mathbb{L}(\mathcal{F}) \triangleq \{x, \neg x | x \in \mathrm{var}(\mathcal{F})\}$
- Implicant: $\tau \subseteq \mathbb{L}(\mathcal{F})$ s.t.
  - $\tau \not\models \bot$ $\quad$ τ is consistent
  - $\tau \models \mathcal{F}$ $\quad$ τ entails $\mathcal{F}$

Compilation – extracting & enumerating primes

- Boolean function: $\mathcal{F}$
- Set of literals of $\mathcal{F}$: $\mathbb{L}(\mathcal{F}) \triangleq \{x, \neg x | x \in \text{var}(\mathcal{F})\}$
- Implicant: $\tau \subseteq \mathbb{L}(\mathcal{F})$ s.t.
  - $\tau \nvDash \perp$ $\quad$ τ is consistent
  - $\tau \vDash \mathcal{F}$ $\quad$ τ entails $\mathcal{F}$
- Prime implicant: $\tau \subseteq \mathbb{L}(\mathcal{F})$ s.t.
  - τ is an implicant of $\mathcal{F}$
  - **No** $\tau' \subseteq \tau$ is an implicant of $\mathcal{F}$

Compilation – extracting & enumerating primes

- Boolean function: $\mathcal{F}$
- Set of literals of $\mathcal{F}$: $\mathbb{L}(\mathcal{F}) \triangleq \{x, \neg x | x \in \mathsf{var}(\mathcal{F})\}$
- Implicant: $\tau \subseteq \mathbb{L}(\mathcal{F})$ s.t.
  - $\tau \nvDash \bot$ $\quad$ τ is consistent
  - $\tau \vDash \mathcal{F}$ $\quad$ τ entails $\mathcal{F}$
- Prime implicant: $\tau \subseteq \mathbb{L}(\mathcal{F})$ s.t.
  - τ is an implicant of $\mathcal{F}$
  - **No** $\tau' \subseteq \tau$ is an implicant of $\mathcal{F}$
- To extract a prime implicant τ given some implicant ρ of $\mathcal{F}$:
  - $\rho \wedge \neg\mathcal{F} \vDash \bot$
  - Find minimal subset τ of ρ such that $\tau \wedge \neg\mathcal{F} \vDash \bot$
    - I.e., extract an MUS of $\rho \wedge \neg\mathcal{F}$

Compilation – extracting & enumerating primes

- Boolean function: $\mathcal{F}$
- Set of literals of $\mathcal{F}$: $\mathbb{L}(\mathcal{F}) \triangleq \{\mathsf{x}, \neg\mathsf{x} | \mathsf{x} \in \mathsf{var}(\mathcal{F})\}$
- Implicant: $\tau \subseteq \mathbb{L}(\mathcal{F})$ s.t.
  - $\tau \nvDash \bot$ $\quad$ τ is consistent
  - $\tau \vDash \mathcal{F}$ $\quad$ τ entails $\mathcal{F}$
- Prime implicant: $\tau \subseteq \mathbb{L}(\mathcal{F})$ s.t.
  - τ is an implicant of $\mathcal{F}$
  - **No** $\tau' \subseteq \tau$ is an implicant of $\mathcal{F}$
- To extract a prime implicant τ given some implicant ρ of $\mathcal{F}$:
  - $\rho \wedge \neg\mathcal{F} \vDash \bot$
  - Find minimal subset τ of ρ such that $\tau \wedge \neg\mathcal{F} \vDash \bot$
    - I.e., extract an MUS of $\rho \wedge \neg\mathcal{F}$
- Prime enumeration:
  - Dedicated algorithm [PIMM15]
  - MUS enumerator [LS08, LPMM16]

An example

$$\mathcal{F} \triangleq a \land \neg c \land \neg d \lor \neg a \land b \land \neg d \lor b \land c \land d$$

- Model/implicant: $\rho = \{a, b, \neg c, \neg d\}$
- Extracting a prime implicant:

An example

$$\mathcal{F} \triangleq a \wedge \neg c \wedge \neg d \vee \neg a \wedge b \wedge \neg d \vee b \wedge c \wedge d$$

- Model/implicant: $\rho = \{a, b, \neg c, \neg d\}$
- Extracting a prime implicant:

τ	Literal l	$\tau \setminus \{l\} \models \mathcal{F}$	Action
$\{a, b, \neg c, \neg d\}$	a	Yes	Drop a

An example

$$\mathcal{F} \triangleq a \wedge \neg c \wedge \neg d \vee \neg a \wedge b \wedge \neg d \vee b \wedge c \wedge d$$

- Model/implicant: $\rho = \{a, b, \neg c, \neg d\}$
- Extracting a prime implicant:

τ	Literal l	$\tau \setminus \{l\} \vDash \mathcal{F}$	Action
$\{a, b, \neg c, \neg d\}$	a	Yes	Drop a
$\{b, \neg c, \neg d\}$	b	No	Keep b

An example

$$\mathcal{F} \triangleq a \land \neg c \land \neg d \lor \neg a \land b \land \neg d \lor b \land c \land d$$

- Model/implicant: $\rho = \{a, b, \neg c, \neg d\}$
- Extracting a prime implicant:

τ	Literal l	$\tau \setminus \{l\} \vDash \mathcal{F}$	Action
$\{a, b, \neg c, \neg d\}$	a	Yes	Drop a
$\{b, \neg c, \neg d\}$	b	No	Keep b
$\{b, \neg c, \neg d\}$	$\neg c$	No	Keep $\neg c$

An example

$$\mathcal{F} \triangleq a \land \neg c \land \neg d \lor \neg a \land b \land \neg d \lor b \land c \land d$$

- Model/implicant: $\rho = \{a, b, \neg c, \neg d\}$
- Extracting a prime implicant:

τ	Literal l	$\tau \setminus \{l\} \models \mathcal{F}$	Action
$\{a, b, \neg c, \neg d\}$	a	Yes	Drop a
$\{b, \neg c, \neg d\}$	b	No	Keep b
$\{b, \neg c, \neg d\}$	$\neg c$	No	Keep $\neg c$
$\{b, \neg c, \neg d\}$	$\neg d$	No	Keep $\neg d$

An example

$$\mathcal{F} \triangleq a \wedge \neg c \wedge \neg d \vee \neg a \wedge b \wedge \neg d \vee b \wedge c \wedge d$$

- Model/implicant: $\rho = \{a, b, \neg c, \neg d\}$
- Extracting a prime implicant:

τ	Literal l	$\tau \setminus \{l\} \vDash \mathcal{F}$	Action
$\{a, b, \neg c, \neg d\}$	a	Yes	Drop a
$\{b, \neg c, \neg d\}$	b	No	Keep b
$\{b, \neg c, \neg d\}$	$\neg c$	No	Keep $\neg c$
$\{b, \neg c, \neg d\}$	$\neg d$	No	Keep $\neg d$

  - Prime implicant: $\{b, \neg c, \neg d\}$

An example

$$\mathcal{F} \triangleq a \wedge \neg c \wedge \neg d \vee \neg a \wedge b \wedge \neg d \vee b \wedge c \wedge d$$

Is $\tau \setminus \{l\} \wedge \neg\mathcal{F}$ inconsistent?

- Model/implicant: $\rho = \{a, b, \neg c, \neg d\}$
- Extracting a prime implicant:

τ	Literal l	$\tau \setminus \{l\} \vDash \mathcal{F}$	Action
$\{a, b, \neg c, \neg d\}$	a	Yes	Drop a
$\{b, \neg c, \neg d\}$	b	No	Keep b
$\{b, \neg c, \neg d\}$	$\neg c$	No	Keep $\neg c$
$\{b, \neg c, \neg d\}$	$\neg d$	No	Keep $\neg d$

  - Prime implicant: $\{b, \neg c, \neg d\}$

Propositional abduction – an example

- Example propositional background theory T:

$$T = \{(\neg x_1 \vee x_4), (\neg x_2 \vee \neg x_3 \vee x_4)\}$$

Propositional abduction – an example

- Example propositional background theory T:

$$T = \{(\neg x_1 \vee x_4), (\neg x_2 \vee \neg x_3 \vee x_4)\}$$

A set of manifestations M:

$$M = \{(x_4)\}$$

Propositional abduction – an example

- Example propositional background theory T:

$$T = \{(\neg x_1 \vee x_4), (\neg x_2 \vee \neg x_3 \vee x_4)\}$$

A set of manifestations M:

$$M = \{(x_4)\}$$

A set of hypotheses H that can explain M given T:

$$H = \{(x_1), (x_2), (x_3)\}$$

Propositional abduction – an example

- Example propositional background theory T:

$$T = \{(\neg x_1 \vee x_4), (\neg x_2 \vee \neg x_3 \vee x_4)\}$$

  A set of manifestations M:

$$M = \{(x_4)\}$$

  A set of hypotheses H that can explain M given T:

$$H = \{(x_1), (x_2), (x_3)\}$$

- Find a smallest subset $S \subseteq H$ that together with T explains M, e.g.

Propositional abduction – an example

- Example propositional background theory T:

$$T = \{(\neg x_1 \vee x_4), (\neg x_2 \vee \neg x_3 \vee x_4)\}$$

A set of manifestations M:

$$M = \{(x_4)\}$$

A set of hypotheses H that can explain M given T:

$$H = \{(x_1), (x_2), (x_3)\}$$

- Find a smallest subset $S \subseteq H$ that together with T explains M, e.g.

$$S = \{(x_1)\}$$

Defining propositional abduction

A Propositional Abduction Problem (PAP) is a 5-tuple $P = (V, H, M, T, c)$ where:

- V - finite set of boolean variables
- H - CNF formula representing the set of hypotheses
- M - CNF formula representing the set of manifestations
- T - CNF formula representing the background theory
- $c : H \rightarrow \mathbb{R}^+$ - cost function, associates a cost to each clause in H

Defining propositional abduction

A Propositional Abduction Problem (PAP) is a 5-tuple $P = (V, H, M, T, c)$ where:

- V - finite set of boolean variables
- H - CNF formula representing the set of hypotheses
- M - CNF formula representing the set of manifestations
- T - CNF formula representing the background theory
- $c : H \to \mathbb{R}^+$ - cost function, associates a cost to each clause in H

The set of explanations of a PAP $P = (V, H, M, T, c)$ is:

$$\mathrm{Expl}(P) = \{S \subseteq H \mid T \wedge S \not\models \perp, T \wedge S \models M\}$$

The minimum-cost solutions of P are:

PAP is hard for the Σ_2^{P} !

$$\mathrm{Expl}_c(P) = \mathrm{argmin}_{E \in \mathrm{Expl}(P)}(c(E))$$

Questions?

References i

[BDTW93] R. R. Bakker, F. Dikker, F. Tempelman, and P. M. Wognum.
Diagnosing and solving over-determined constraint satisfaction problems.
In *IJCAI*, pages 276–281, 1993.

[BS05] James Bailey and Peter J. Stuckey.
Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization.
In *PADL*, pages 174–186, 2005.

[CD91] John W. Chinneck and Erik W. Dravnieks.
Locating minimal infeasible constraint sets in linear programs.
INFORMS Journal on Computing, 3(2):157–168, 1991.

[Coo71] Stephen A. Cook.
The complexity of theorem-proving procedures.
In *STOC*, pages 151–158. ACM, 1971.

[FJ18] Matteo Fischetti and Jason Jo.
Deep neural networks and mixed integer linear optimization.
Constraints, 23(3):296–309, 2018.

[KBD+17] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer.
Reluplex: An efficient SMT solver for verifying deep neural networks.
In *CAV*, pages 97–117, 2017.

References ii

[LPMM16] Mark H. Liffiton, Alessandro Previti, Ammar Malik, and Joao Marques-Silva.
Fast, flexible MUS enumeration.
Constraints, 21(2):223–250, 2016.

[LS08] Mark H. Liffiton and Karem A. Sakallah.
Algorithms for computing minimal unsatisfiable subsets of constraints.
J. Autom. Reasoning, 40(1):1–33, 2008.

[NH10] Vinod Nair and Geoffrey E. Hinton.
Rectified linear units improve restricted boltzmann machines.
In *ICML*, pages 807–814, 2010.

[PIMM15] Alessandro Previti, Alexey Ignatiev, António Morgado, and Joao Marques-Silva.
Prime compilation of non-clausal formulae.
In *IJCAI*, pages 1980–1988, 2015.

[Rei87] Raymond Reiter.
A theory of diagnosis from first principles.
Artif. Intell., 32(1):57–95, 1987.

Interpretable Classification

Given training data, learn function that correctly classifies that data, performs suitably well on unseen data, and offers human-interpretable functions for the predictions made

Interpretable Classification

Given training data, learn function that correctly classifies that data, performs suitably well on unseen data, and offers human-interpretable functions for the predictions made

Given training data, learn **decision sets/decision trees** that correctly classify that data, perform suitably well on unseen data, and offer human-interpretable functions for the predictions made

Recipe

Step 1 Discretization of the training and test dataset

Recipe

Step 1 Discretization of the training and test dataset

Step 2 Define the grammar of the classifier

Recipe

Step 1 Discretization of the training and test dataset

Step 2 Define the grammar of the classifier

Step 3 Hard Constraints to capture structure of the rules

Recipe

Step 1 Discretization of the training and test dataset

Step 2 Define the grammar of the classifier

Step 3 Hard Constraints to capture structure of the rules

Step 4 Hard Constraints to capture evaluation of rules: A rule must

- return True on positive example and False on negative example

Recipe

Step 1 Discretization of the training and test dataset

Step 2 Define the grammar of the classifier

Step 3 Hard Constraints to capture structure of the rules

Step 4 Hard Constraints to capture evaluation of rules: A rule must

- return True on positive example and False on negative example

Step 5 Soft Constraints

- Minimize the size of rules

Recipe

Step 1 Discretization of the training and test dataset

Step 2 Define the grammar of the classifier

Step 3 Hard Constraints to capture structure of the rules

Step 4 Hard Constraints to capture evaluation of rules: A rule must

- return True on positive example and False on negative example

Step 5 Soft Constraints

- Minimize the size of rules

Step 6 Rely on progress in SAT and MaxSAT solving over the past decade

Outline

Discretization

Classification via Decision Sets

Decision Sets via MaxSAT

Incremental learning

Discretization

Ex.	Height (H)	Weight (W)	Risk (R)
e_1	160	210	0
e_2	175	210	0
e_3	170	190	1
e_4	166	190	0
e_5	172	170	1

Discretization

Ex.	Height (H)	Weight (W)	Risk (R)
e_1	160	210	0
e_2	175	210	0
e_3	170	190	1
e_4	166	190	0
e_5	172	170	1

- Suppose Height can range between 50 and 250 cm and weight ranges between 100 and 300.
- Do we need variable for every value of H and W?

Discretization

Ex.	Height (H)	Weight (W)	Risk (R)
e_1	160	210	0
e_2	175	210	0
e_3	170	190	1
e_4	166	190	0
e_5	172	170	1

- Suppose Height can range between 50 and 250 cm and weight ranges between 100 and 300.
- Do we need variable for every value of H and W?
- One-hot encoding: Only introduce variables to differentiate two distinct data points.
  - Variables corresponding to $H \geq 170$, $H \geq 165$, $H \geq 172$, $H \geq 175$ suffice
  - Variables corresponding to $W \geq 200$ and $W \geq 180$

Discretization

Ex.	Height (H)	Weight (W)	Risk (R)
e_1	160	210	0
e_2	175	210	0
e_3	170	190	1
e_4	166	190	0
e_5	172	170	1

Ex.	$H \geq 170$	$H \geq 165$	$H \geq 172$	$H \geq 175$	$W > 200$	$W > 180$	Risk (R)
e_1	0	0	0	0	1	0	0
e_2	1	0	1	1	1	0	0
e_3	1	1	0	0	0	1	1
e_4	0	1	0	0	0	1	0
e_5	1	1	1	0	0	0	1

Outline

Discretization

Classification via Decision Sets

Decision Sets via MaxSAT

Incremental learning

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$
- Binary **features**: $\mathcal{F} = \{f_1, \ldots, f_K\}$
  - $f_1 \triangleq \text{V}$, $f_2 \triangleq \text{C}$, $f_3 \triangleq \text{M}$, and $f_4 \triangleq \text{E}$
  - Literals: f_r and $\neg f_r$

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$
- Binary **features**: $\mathcal{F} = \{f_1, \ldots, f_K\}$
  - $f_1 \triangleq \mathrm{V}$, $f_2 \triangleq \mathrm{C}$, $f_3 \triangleq \mathrm{M}$, and $f_4 \triangleq \mathrm{E}$
  - Literals: f_r and $\neg f_r$
- **Feature space**: $\mathcal{U} \triangleq \prod_{r=1}^{K}\{f_r, \neg f_r\}$

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$
- Binary **features**: $\mathcal{F} = \{f_1, \ldots, f_K\}$
  - $f_1 \triangleq$ V, $f_2 \triangleq$ C, $f_3 \triangleq$ M, and $f_4 \triangleq$ E
  - Literals: f_r and $\neg f_r$
- **Feature space**: $\mathcal{U} \triangleq \prod_{r=1}^{K} \{f_r, \neg f_r\}$
- Binary classification: $\mathcal{C} = \{c_0 = 0, c_1 = 1\}$
  - $\mathcal{E}$ partitioned into $\mathcal{E}^-$ and $\mathcal{E}^+$

Example

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Binary features: $\mathcal{F} = \{f_1, f_2, f_3, f_4\}$
  - $f_1 \triangleq \mathrm{V}$, $f_2 \triangleq \mathrm{C}$, $f_3 \triangleq \mathrm{M}$, and $f_4 \triangleq \mathrm{E}$
- e_1 is represented by the 2-tuple (π_1, ς_1),
  - $\pi_1 = (\neg\mathrm{V}, \neg\mathrm{C}, \mathrm{M}, \neg\mathrm{E})$
  - $\varsigma_1 = 0$
- $\mathcal{U} = \{\mathrm{V}, \neg\mathrm{V}\} \times \{\mathrm{C}, \neg\mathrm{C}\} \times \{\mathrm{M}, \neg\mathrm{M}\} \times \{\mathrm{E}, \neg\mathrm{E}\}$

Itemsets & decision sets

- Given $\mathcal{F}$, an **itemset** π is an element of $\mathcal{I} \triangleq \prod_{r=1}^{K} \{f_r, \neg f_r\}$

Itemsets & decision sets

- Given $\mathcal{F}$, an **itemset** π is an element of $\mathcal{I} \triangleq \prod_{r=1}^{K} \{f_r, \neg f_r\}$
- A **rule** is a 2-tuple (π, c), with itemset $\pi \in \mathcal{I}$, and class $c \in \mathcal{C}$
  Rule (π, c) interpreted as:

  IF all specified literals in π are true, **THEN** pick class c

Itemsets & decision sets

- Given $\mathcal{F}$, an **itemset** π is an element of $\mathcal{I} \triangleq \prod_{r=1}^{K}\{f_r, \neg f_r\}$

- A **rule** is a 2-tuple (π, c), with itemset $\pi \in \mathcal{I}$, and class $c \in \mathcal{C}$
  Rule (π, c) interpreted as:

  IF all specified literals in π are true, **THEN** pick class c

- A **decision set** $\mathbb{S}$ is a finite set of rules – **unordered**

Itemsets & decision sets

- Given $\mathcal{F}$, an **itemset** π is an element of $\mathcal{I} \triangleq \prod_{r=1}^{K}\{f_r, \neg f_r\}$

- A **rule** is a 2-tuple (π, c), with itemset $\pi \in \mathcal{I}$, and class $c \in \mathcal{C}$
  Rule (π, c) interpreted as:

  IF all specified literals in π are true, **THEN** pick class c

- A **decision set** $\mathbb{S}$ is a finite set of rules – **unordered**

- A rule of the form $\mathfrak{D} \triangleq (\emptyset, c)$ denotes the **default rule** of a decision set $\mathbb{S}$
  - Default rule is **optional** and used **only** when other rules do not apply on some feature space point
  - In this talk, we will seek to learn

Example

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Rule 1: $((\neg \text{M}, \neg \text{E}), c_1)$
  - Meaning: **if** ¬Meeting and ¬Expo **then** Hike
- Rule 2: $((\text{V}, \neg \text{C}), c_1)$
  - Meaning: **if** Vacation and ¬Concert **then** Hike
- Rule 3: $((\neg \text{V}, \text{M}), c_0)$
  - Meaning: **if** ¬Vacation and Meeting **then** ¬Hike

Example

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Rule 1: $((\neg \mathrm{M}, \neg \mathrm{E}), c_1)$
  - Meaning: **if** ¬Meeting and ¬Expo **then** Hike
- Rule 2: $((\mathrm{V}, \neg \mathrm{C}), c_1)$
  - Meaning: **if** Vacation and ¬Concert **then** Hike
- Rule 3: $((\neg \mathrm{V}, \mathrm{M}), c_0)$
  - Meaning: **if** ¬Vacation and Meeting **then** ¬Hike
- Default rule: $(\emptyset, c_0)$
  - Meaning: if all other rules do not apply, then pick ¬Hike

Succinct explanations

- If a rule fires, the set of literals represents the **explanation** for the predicted class
  - Explanation is **succinct** : **only** the literals in the rule used; independent of example
- For the default class, **must** pick one **falsified** literal in **every** rule that predicts a different class
  - Explanation is **not succinct** : explanation depends on **each** example
- **Obs:** **Uninteresting** to predict c_1 as **negation** of c_0 (and vice-versa)
  - Explanations also **not** succinct

Stating our goals

- Assumptions:
  - Also, let $\mathcal{E}^- \wedge \mathcal{E}^+ \models \bot$

Stating our goals

- Assumptions:
  - Also, let $\mathcal{E}^- \land \mathcal{E}^+ \models \perp$
- DNF functions to compute:
  - F^0 for predicting c_0, while **ensuring** $\mathcal{E}^- \models F^0$
  - F^1 for predicting c_1, while **ensuring** $\mathcal{E}^+ \models F^1$

Different Possibilities

- MinDS$_0$:

  Find the **smallest** DNF formulas F^0 and F^1 such that:

  1. $\mathcal{E}^- \models F^0$
  2. $\mathcal{E}^+ \models F^1$
  3. $F^1 \leftrightarrow F^0 \models \perp$

  – **Obs:** MinDS$_0$ ensures **succinct** explanations
    - Computes F^0 and F^1 (i.e. **no** negation) and **no** default rule

Different Possibilities

- MinDS$_0$:

  Find the **smallest** DNF formulas F^0 and F^1 such that:

  1. $\mathcal{E}^- \vDash F^0$
  2. $\mathcal{E}^+ \vDash F^1$
  3. $F^1 \leftrightarrow F^0 \vDash \bot$

  – **Obs:** MinDS$_0$ ensures **succinct** explanations
    - Computes F^0 and F^1 (i.e. **no** negation) and **no** default rule
- MinDS$_3$: Minimize F^1 such that
  1. $\mathcal{E}^+ \vDash F^1$
  2. $F^1 \wedge \mathcal{E}^- \vDash \bot$

  – No succinct explanations for F^0
- MinDS$_4$: Minimize F^0 such that
  1. $\mathcal{E}^- \vDash F^0$
  2. $F^0 \wedge \mathcal{E}^+ \vDash \bot$

  – No succinct explanations for F^1

Outline

Boolean Formulation of MinDS$_3$

- DNF representation for F^1
- Consider N terms
  - $F^1 := F_1^1 \vee F_2^1 \cdots F_N^1$, where

    $$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots \wedge((b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

    - If $b_{i,1}$ is true, then f_1 is in F_i^1.
    - If $c_{i,1}$ is true, then $\neg f_1$ is in F_i^1.
    - If $d_{i,1}$ is true, then f_1 and $\neg f_1$ do not appear in F_i^1
  - F_i^1 is a DNF term if exactly one of $\{b_{i,r}, c_{i,r}, d_{i,r}\}$ is true for each r.

Boolean Formulation of MinDS$_3$

- DNF representation for F^1

- Consider N terms
  - $F^1 := F_1^1 \vee F_2^1 \cdots F_N^1$, where

$$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots$$
$$\wedge((b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

    - If $b_{i,1}$ is true, then f_1 is in F_i^1.
    - If $c_{i,1}$ is true, then $\neg f_1$ is in F_i^1.
    - If $d_{i,1}$ is true, then f_1 and $\neg f_1$ do not appear in F_i^1
  - F_i^1 is a DNF term if exactly one of $\{b_{i,r}, c_{i,r}, d_{i,r}\}$ is true for each r.

- Goal: Find values of $\{b_{i,j}, c_{i,j}, d_{i,j}\}$

MaxSAT Formulation

- Recall
  - $\sigma(r,q)$: value of feature f_r for e_q

$$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots \wedge((b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

- Structural Constraints: $\bigwedge_{i,r} ExactlyOne(b_{i,r}, c_{i,r}, d_{i,r})$
- $\mathcal{E}^+ \models F^1$: For $e_q \in \mathcal{E}^+$, $F^1[\bigwedge_r f_r \mapsto \sigma(r,q)] = 1$ (Hard)
- $F^1 \wedge \mathcal{E}^- \models \bot$: For $e_q \in \mathcal{E}^-$, $F^1[\bigwedge_r f_r \mapsto \sigma(r,q)] = 0$ (Hard)
- Soft Constraints: $\mathcal{S}_{i,r} := (\neg b_{i,r})c_{i,r})$; $\qquad W(\mathcal{S}_{i,r}) = 1$
  - Minimize the size of each term
  - Can have different objective functions

Example

Ex.	Vacation (V) f_1	Meeting (M) f_2	Expo (E) f_3	Hike (H) Label
e_1	0	1	0	1
e_2	1	0	0	0
e_3	0	1	1	1

Suppose, we want to learn F^1 of one term ,i.e., $N = 1$. Remember,
$F_1^1 = (b_{1,1} \cdot f_1 \vee c_{1,1} \cdot \neg f_1 \vee d_{1,1}) \vee (b_{1,2} \cdot f_2 \vee c_{1,2} \cdot \neg f_2 \vee d_{1,2}) \wedge (b_{1,3} \cdot f_3 \vee c_{1,3} \cdot \neg f_3 \vee d_{1,3})$
$F_2^1 = (b_{2,1} \cdot f_1 \vee c_{2,1} \cdot \neg f_1 \vee d_{2,1}) \vee (b_{2,2} \cdot f_2 \vee c_{2,2} \cdot \neg f_2 \vee d_{2,2}) \vee (b_{2,3} \cdot f_3 \vee c_{2,3} \cdot \neg f_3 \vee d_{2,3})$

1. For e_1, we have $F^1[\bigwedge_r f_r \mapsto \sigma(r, q)] =$ $((c_{1,1} \vee d_{1,1}) \wedge (b_{1,2} \vee d_{1,2}) \wedge (c_{1,3} \vee d_{1,3})) \vee$

Example

Ex.	Vacation (V) f_1	Meeting (M) f_2	Expo (E) f_3	Hike (H) Label
e_1	0	1	0	1
e_2	1	0	0	0
e_3	0	1	1	1

Suppose, we want to learn F^1 of one term ,i.e., $N = 1$. Remember,
$F_1^1 = (b_{1,1} \cdot f_1 \vee c_{1,1} \cdot \neg f_1 \vee d_{1,1}) \vee (b_{1,2} \cdot f_2 \vee c_{1,2} \cdot \neg f_2 \vee d_{1,2}) \wedge (b_{1,3} \cdot f_3 \vee c_{1,3} \cdot \neg f_3 \vee d_{1,3})$
$F_2^1 = (b_{2,1} \cdot f_1 \vee c_{2,1} \cdot \neg f_1 \vee d_{2,1}) \vee (b_{2,2} \cdot f_2 \vee c_{2,2} \cdot \neg f_2 \vee d_{2,2}) \vee (b_{2,3} \cdot f_3 \vee c_{2,3} \cdot \neg f_3 \vee d_{2,3})$

1. For e_1, we have $F^1[\bigwedge_r f_r \mapsto \sigma(r, q)] =$
   $((c_{1,1} \vee d_{1,1}) \wedge (b_{1,2} \vee d_{1,2}) \wedge (c_{1,3} \vee d_{1,3})) \vee$
   $((c_{2,1} \vee d_{2,1}) \wedge (b_{2,2} \vee d_{2,2}) \wedge (c_{2,3} \vee d_{2,3}))$

Example

Ex.	Vacation (V) f_1	Meeting (M) f_2	Expo (E) f_3	Hike (H) Label
e_1	0	1	0	1
e_2	1	0	0	0
e_3	0	1	1	1

Suppose, we want to learn F^1 of one term ,i.e., $N = 1$. Remember,
$F_1^1 = (b_{1,1} \cdot f_1 \vee c_{1,1} \cdot \neg f_1 \vee d_{1,1}) \vee (b_{1,2} \cdot f_2 \vee c_{1,2} \cdot \neg f_2 \vee d_{1,2}) \wedge (b_{1,3} \cdot f_3 \vee c_{1,3} \cdot \neg f_3 \vee d_{1,3})$
$F_2^1 = (b_{2,1} \cdot f_1 \vee c_{2,1} \cdot \neg f_1 \vee d_{2,1}) \vee (b_{2,2} \cdot f_2 \vee c_{2,2} \cdot \neg f_2 \vee d_{2,2}) \vee (b_{2,3} \cdot f_3 \vee c_{2,3} \cdot \neg f_3 \vee d_{2,3})$

1. Suppose, MaxSAT solver returns $b_{1,1} = c_{1,2} = d_{1,3} = d_{2,1} = d_{2,3} = b_{2,3} = 1$; then the rule is

Example

Ex.	Vacation (V) f_1	Meeting (M) f_2	Expo (E) f_3	Hike (H) Label
e_1	0	1	0	1
e_2	1	0	0	0
e_3	0	1	1	1

Suppose, we want to learn F^1 of one term ,i.e., $N = 1$. Remember,
$F^1_1 = (b_{1,1} \cdot f_1 \vee c_{1,1} \cdot \neg f_1 \vee d_{1,1}) \vee (b_{1,2} \cdot f_2 \vee c_{1,2} \cdot \neg f_2 \vee d_{1,2}) \wedge (b_{1,3} \cdot f_3 \vee c_{1,3} \cdot \neg f_3 \vee d_{1,3})$
$F^1_2 = (b_{2,1} \cdot f_1 \vee c_{2,1} \cdot \neg f_1 \vee d_{2,1}) \vee (b_{2,2} \cdot f_2 \vee c_{2,2} \cdot \neg f_2 \vee d_{2,2}) \vee (b_{2,3} \cdot f_3 \vee c_{2,3} \cdot \neg f_3 \vee d_{2,3})$

1. Suppose, MaxSAT solver returns $b_{1,1} = c_{1,2} = d_{1,3} = d_{2,1} = d_{2,3} = b_{2,3} = 1$; then the rule is $F^1 = (f_1 \wedge \neg f_2) \vee (f_2)$

Tools

- The MaxSAT formulation is NP-hard
- Use Local search based approaches [LBS, KDD-16]
  - Local search-based:
    git clone `git@github.com:jirifilip/pyIDS.git`
- Use MaxSAT solvers [IPNM, IJCAR-18]
  - Significant progress in MaxSAT solving over the past decade
  - Usage of symmetry breaking predicates
  - MaxSAT-based Decision sets
    git clone `https://github.com/alexeyignatiev/minds`

Tools

- The MaxSAT formulation is NP-hard
- Use Local search based approaches [LBS, KDD-16]
  - Local search-based:
    git clone `git@github.com:jirifilip/pyIDS.git`
- Use MaxSAT solvers [IPNM, IJCAR-18]
  - Significant progress in MaxSAT solving over the past decade
  - Usage of symmetry breaking predicates
  - MaxSAT-based Decision sets
    git clone `https://github.com/alexeyignatiev/minds`
- Results: Over a set of 49 instances, local-search based approach can handle only 2 instances while MaxSAT based approach can optimal decision sets of 42 instances [IPNM, IJCAR-18]

Looking Beyond: Handling Noise

- Noisy data sets: collection of data, non-existence of perfect rules
  - The optimal decision sets are too large.

Looking Beyond: Handling Noise

- Noisy data sets: collection of data, non-existence of perfect rules
  - The optimal decision sets are too large.
- MinDS$_3$: Minimize F^1 and such that
  1. $\mathcal{E}^+ \models F^1$
  2. $F^1 \wedge \mathcal{E}^- \models \perp$

  - No succinct explanations for F^0
- Noisy MinDS$_3$: Minimize F^1, such that
  1. $\mathbb{1}_q = 1$ if $e_q \not\models F^1$ for $e_q \in \mathcal{E}^+$ or $e_q \models F^1$ for $e_q \in \mathcal{E}^+$
  2. Minimize $|F| + \lambda \sum_q \mathbb{1}_q$

MaxSAT Formulation for Noisy Setting

[MM, CP-18]

$$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots \wedge (b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

- Notations
  - Variables: $\{b_{i,r}, c_{i,r}, d_{i,r}, \eta_q\}$
  - e_q: example q
  - $\sigma(r, q)$: sign of feature f_r for e_q

MaxSAT Formulation for Noisy Setting

[MM, CP-18]

$$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots \wedge (b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

- Notations
  - Variables: $\{b_{i,r}, c_{i,r}, d_{i,r}, \eta_q\}$
  - e_q: example q
  - $\sigma(r,q)$: sign of feature f_r for e_q
- Hard Constraints:
  - Structural Constraints: $\bigwedge_{i,r} ExactlyOne(b_{i,r}, c_{i,r}, d_{i,r})$
  - $\mathcal{E}^+ \models F^1$: For $e_q \in \mathcal{E}^+$, $F^1[\bigwedge_r f_r \mapsto \sigma(r,q)] = 1 \oplus \eta_q$ (Hard)
  - $F^1 \wedge \mathcal{E}^- \models \perp$: For $e_q \in \mathcal{E}^-$, $F^1[\bigwedge_r f_r \mapsto \sigma(r,q)] = 0 \oplus \eta_q$ (Hard)

MaxSAT Formulation for Noisy Setting

[MM, CP-18]

$$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots \wedge (b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

- Notations
  - Variables: $\{b_{i,r}, c_{i,r}, d_{i,r}, \eta_q\}$
  - e_q: example q
  - $\sigma(r,q)$: sign of feature f_r for e_q
- Hard Constraints:
  - Structural Constraints: $\bigwedge_{i,r} ExactlyOne(b_{i,r}, c_{i,r}, d_{i,r})$
  - $\mathcal{E}^+ \models F^1$: For $e_q \in \mathcal{E}^+$, $F^1[\bigwedge_r f_r \mapsto \sigma(r,q)] = 1 \oplus \eta_q$ (Hard)
  - $F^1 \wedge \mathcal{E}^- \models \bot$: For $e_q \in \mathcal{E}^-$, $F^1[\bigwedge_r f_r \mapsto \sigma(r,q)] = 0 \oplus \eta_q$ (Hard)
- Soft Constraints
  - Minimize the size of each term: $\mathcal{S}_{i,r} := (d_{i,r})$; $\quad W(\mathcal{S}_{i,r}) = 1$
  - Minimize mis-classification: $\mathcal{T}_q := (\neg \eta_q)$ $\quad W(\mathcal{T}_q) = 1$

Illustrative Example

- Iris Classification:
- Features: sepal length, sepal width, petal length, and petal width
- MLIC learned $\mathcal{R}$=
    1. (sepal length $\leq$ 6.3 $\wedge$ sepal width $\leq$ 3.0 $\wedge$ petal width $\geq$ 1.5) $\vee$
    2. (sepal width $\geq$ 2.7 $\wedge$ petal length $\leq$ 4.0 $\wedge$ petal width $\leq$ 1.2) $\vee$
    3. (petal length $>$ 5.0)

Accuracy

Dataset	Size	# Features	RIPPER	Log Reg	NN	RF	SVM	MLIC
ionosphere	350	564	0.886 (0.1)	0.909 (0.1)	0.926 (1.2)	0.909 (1.3)	0.886 (0.1)	0.889 (15.04)
parkinsons	190	392	0.868 (0.1)	0.884 (0.1)	0.921 (1.2)	0.895 (1.1)	0.879 (1.6)	0.895 (245)
Trans	740	64	0.78 (0.0)	0.759 (0.0)	0.788 (1.2)	0.788 (1.2)	0.765 (372.3)	0.797 (1177)
WDBC	560	540	0.961 (0.1)	0.936 (0.0)	0.961 (1.3)	0.943 (1.4)	0.955 (3.0)	0.946 (911)

Intepretability

Dataset	Examples	# Features	MLIC
ionosphere	350	564	5.5
parkinsons	190	392	6
Trans	740	64	4
WDBC	560	540	3.5

Scalability

How do we scale to tens of thousands of examples and features?

Primary Bottleneck Size of MaxSAT formula $\mathcal{O}(M \cdot N \cdot K)$ for a formula on M examples, N clauses and K features

Outline

Discretization

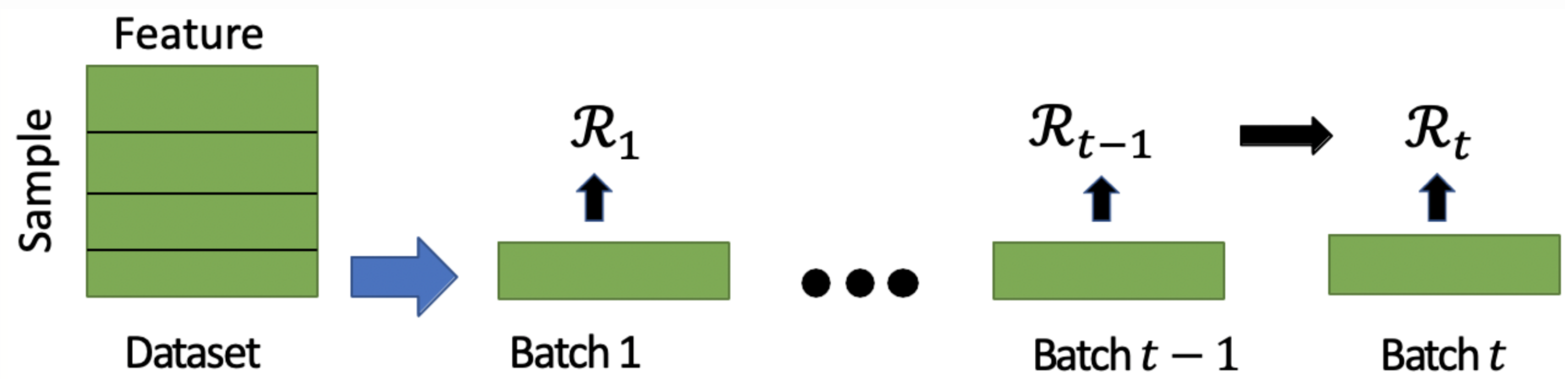
Classification via Decision Sets

Decision Sets via MaxSAT

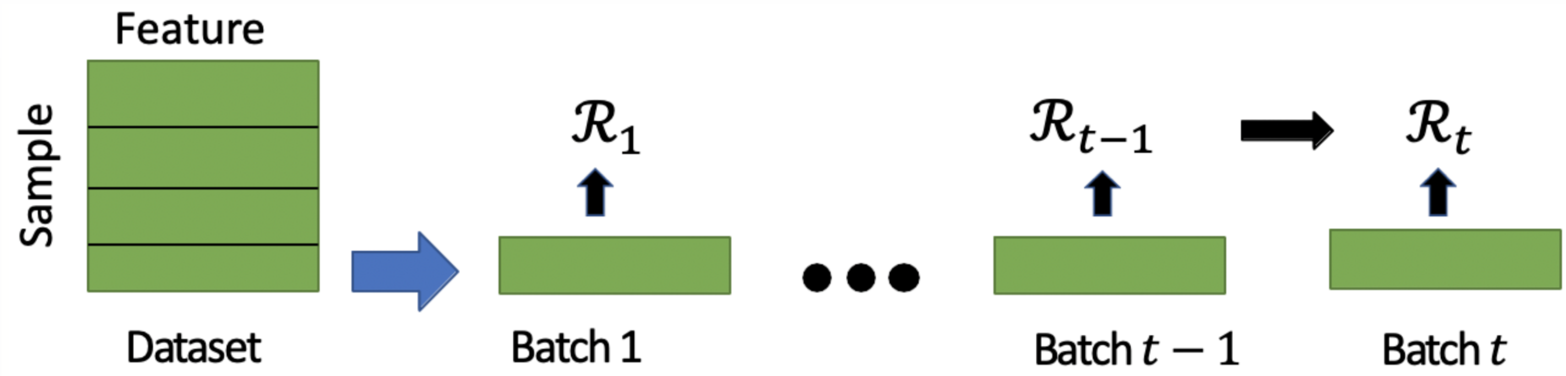
Incremental learning

IMLI: Incremental Rule-learning Approach

- The large formula size of the MaxSAT instance for the poor scalability
- The proposal of mini-batch incremental learning [Ghosh and M., AIES 19]



IMLI: Solution Technique - I



- We propose a mini-batch incremental learning framework with the following objective function on batch t

$$\min \sum_{i,j} (b_{i,j} \cdot I(b_{i,j}) + c_{i,j} \cdot I(c_{i,j}) + d_{i,j} \cdot I(d_{i,j})) + \lambda \sum_{q} \eta_q.$$

where indicator function $I(\cdot)$ is defined as follows.

$$I(b_{i,j}) = \begin{cases} -1 & \text{if } b_{i,j} \in \mathcal{R}_{t-1} \\ 1 & \text{otherwise} \end{cases}$$

Similarly, for $I(c_{i,j})$ and $I(d_{i,j})$

IMLI: Solution Technique - II

$(t-1)$-th batch

we learn assignment

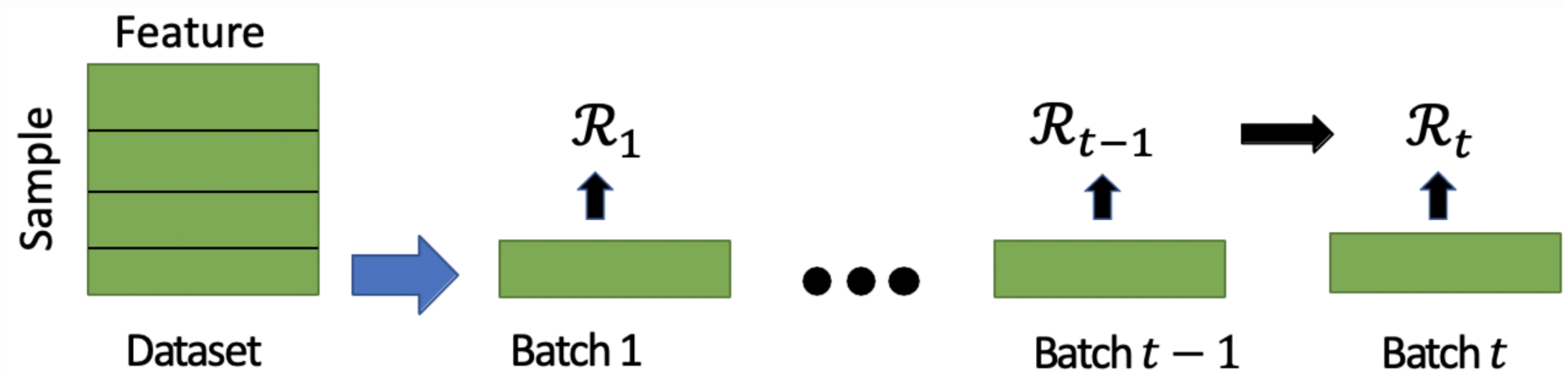
- $b_{1,1} = 0$
- $b_{1,2} = 1$
- $b_{2,1} = 0$
- $b_{2,2} = 1$

t-th batch

we construct soft unit clause

- $\neg b_{1,1}$
- $b_{1,2}$
- $\neg b_{2,1}$
- $b_{2,2}$

IMLI: Solution Technique-III



For M examples, N clauses, and K features,

- The number of clauses for each batch is $\mathcal{O}(\frac{M}{t} \cdot N \cdot K)$
  - Significant reduction from $\mathcal{O}(M \cdot N \cdot K)$

Accuracy and training time of different classifiers

Dataset	Size n	Features m	LR	SVC	RIPPER	IMLI
PIMA	768	134	75.32 (0.3s)	75.32 (0.37s)	75.32 (2.58s)	73.38 (0.74s)
Credit-default	30000	334	80.81 (6.87s)	80.69 (847.93s)	80.97 (20.37s)	79.41 (32.58s)
Twitter	49999	1050	95.67 (3.99s)	Timeout	95.56 (98.21s)	94.69 (59.67s)

Table: Each cell in the last 5 columns refers to test accuracy (%) and training time (s).

MLIC timed out on all the above instances

Size of rules of different rule-based classifiers

Dataset	RIPPER	IMLI
PIMA	8.25	3.5
Twitter	21.6	6
Credit	14.25	3

Table: Average size of the rules of different rule-based models.

IMLI generates shorter rules compared to other rule-based models

Example Rules

Rule for Pima Indians Diabetes Database

Tested positive for diabetes if $:=$

(Plasma glucose concentration > 125 AND Triceps thickness ≤ 35 mm AND Diabetes pedigree function > 0.259 AND Age > 25 years)

Example Rules

Rule for Pima Indians Diabetes Database

Tested positive for diabetes if $:=$
(Plasma glucose concentration $>$ 125 AND Triceps thickness $\leq$ 35 mm AND Diabetes pedigree function $>$ 0.259 AND Age $>$ 25 years)

Rule for Parkinson's Disease Dataset

A person has Parkinson's disease if $:=$
(minimum vocal fundamental frequency $\leq$ 87.57 Hz OR minimum vocal fundamental frequency $>$ 121.38 Hz OR Shimmer:APQ3 $\leq$ 0.01 OR MDVP:APQ $>$ 0.02 OR D2 $\leq$ 1.93 OR NHR $>$ 0.01 OR HNR $>$ 26.5 OR spread2 $>$ 0.3) AND
(Maximum vocal fundamental frequency $\leq$ 200.41 Hz OR HNR $\leq$ 18.8 OR spread2 $>$ 0.18 OR D2 $>$ 2.92)

Recipe so far

- Discretization of the training and test dataset
- Hard Constraints to capture structure of the rules
- Hard Constraints to capture evaluation of rules: A rule must
  - EITHER return True on positive example and False on negative example
  - OR the noise variable is set to True
- Soft Constraints
  - Minimize the size of rules
  - Minimize the number of mis-classifications

From Decisions Sets to Decision Trees

[NIPM, IJCAI-18]

- Hard Constraints to capture structure of the rules
  - A leaf node has no children and is either 0 (False) or 1 (True)
  - A non-leaf node must have a child.
  - If the i-th node is a parent then it must have a child
  - All nodes (except root) must have a parent
  - Left edge corresponding to node with label f_r corresponds to $f_r = 0$
  - Right edge corresponding to node with label f_r corresponds to $f_r = 1$
- Evaluation along a path is just conjunction of edges
- Hard constraints to capture evaluation of rules
  - return True on positive example and False on negative example
- Exploitation of domain specific knowledge to improve encoding

Conclusions & research directions

- SAT/MaxSAT-based solutions for computing (explainable) decision sets
  - Minimize the number of terms
  - Allows several different objective functions
- Far better than local search based approach

Conclusions & research directions

- SAT/MaxSAT-based solutions for computing (explainable) decision sets
  - Minimize the number of terms
  - Allows several different objective functions
- Far better than local search based approach

- Formalizations beyond Decisions sets and Decision Trees
  - Checklists [GMM, ECAI20]
  - The underlying approach can be applied
  - Exploitation of domain specific knowledge
- Scalability and handling very large data sets.

Tools

- Local search-based:
  git clone `git@github.com:jirifilip/pyIDS.git`
- MaxSAT-based Decision sets
  git clone `https://github.com/alexeyignatiev/minds`
- Noisy and Incremental: pip install rulelearning

Questions?

Part 3. Robustness of ML models

Nina Narodytska

Part 3. Robustness of Deep NNs

Nina Narodytska

vmware

Outline

Outline

Motivation

Adversarial attacks

Verification methods

SAT-based verification of Binarized NNs

Outline

Motivation

Why robustness?

Robustness of ML models

Interpretability of ML models

Why robustness?

Robustness of ML models

Interpretability of ML models

Why robustness?

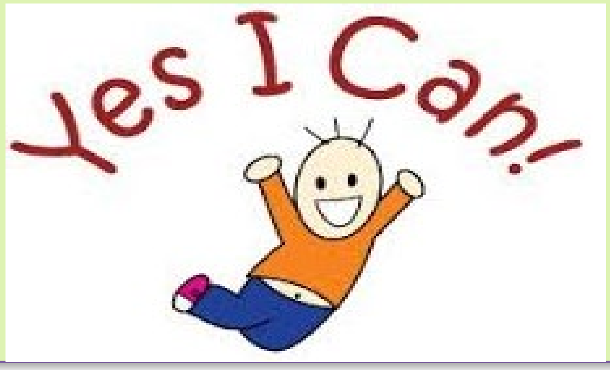
Robustness of ML models

Interpretability of ML models

Why robustness?

Robustness of ML models

Interpretability of ML models



Why robustness?

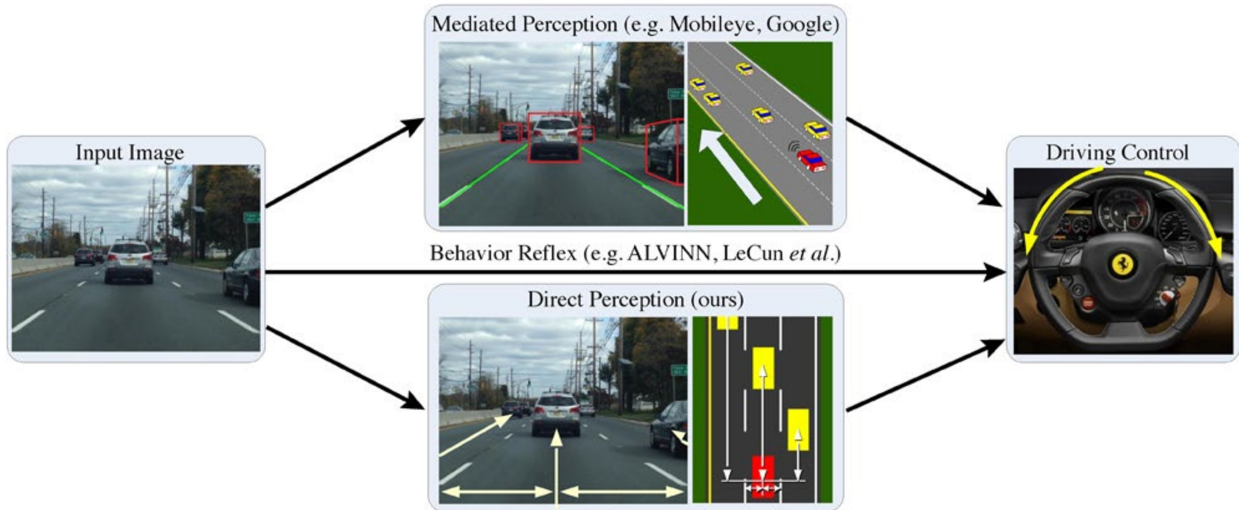
Robustness of ML models

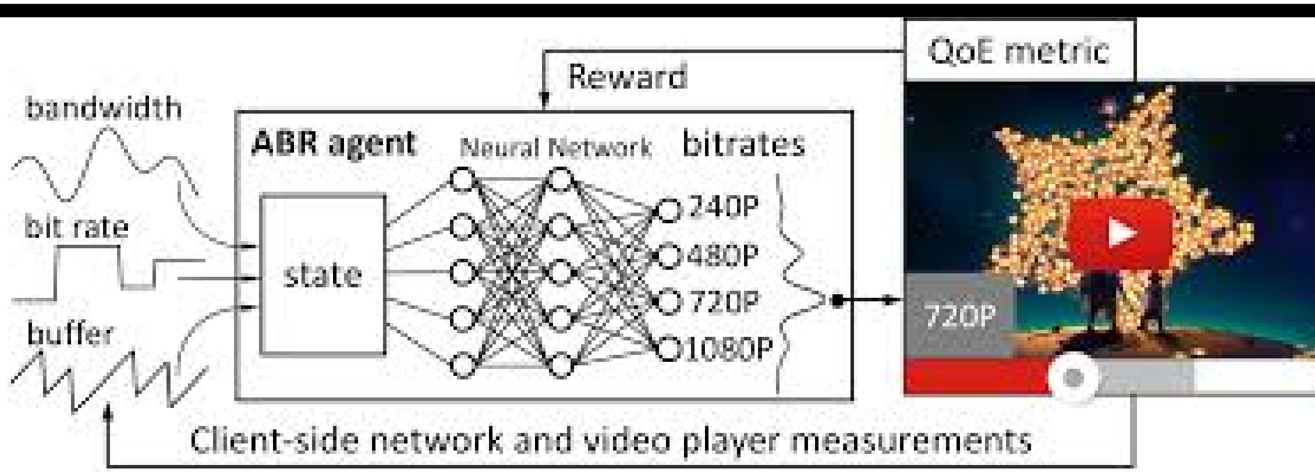
??? Part 5!!!

Interpretability of ML models

Dialogs/chat bots

Control systems





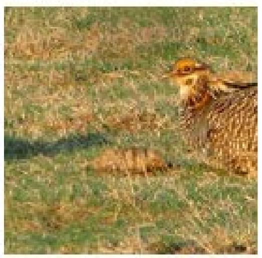
Machine Learning is used on daily basis

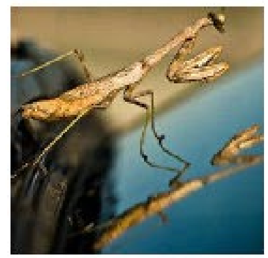
Deep learning-based systems can be fooled

Deep learning-based systems can be fooled

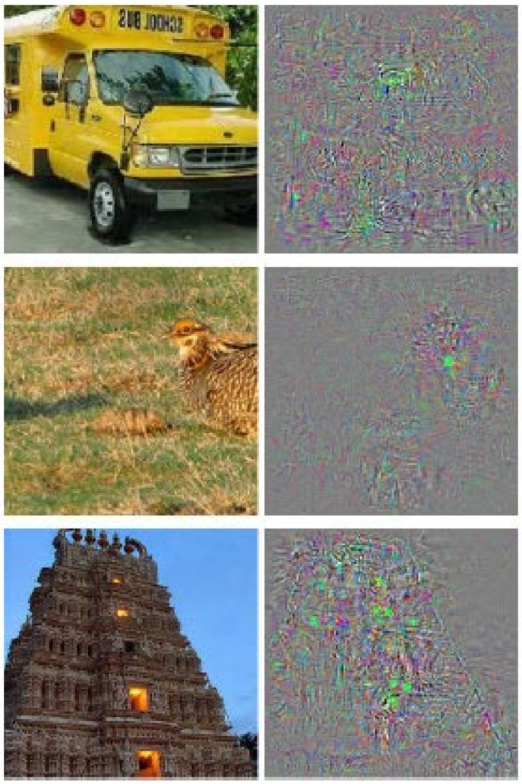
Easily

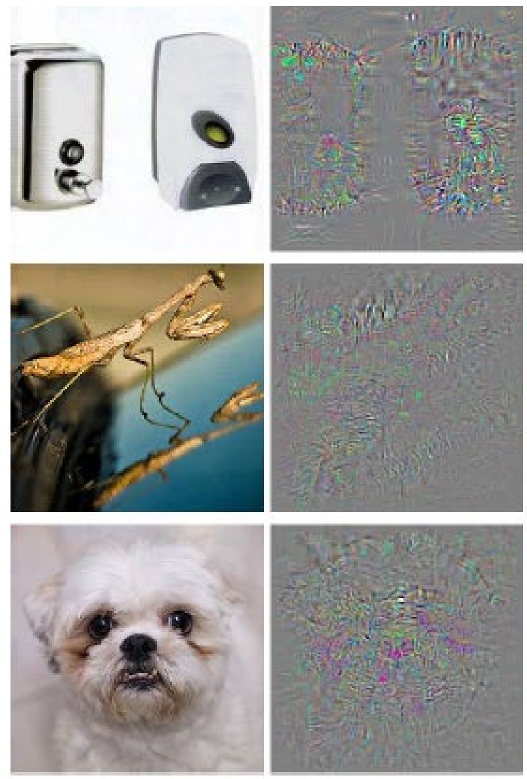
Fooling DL systems



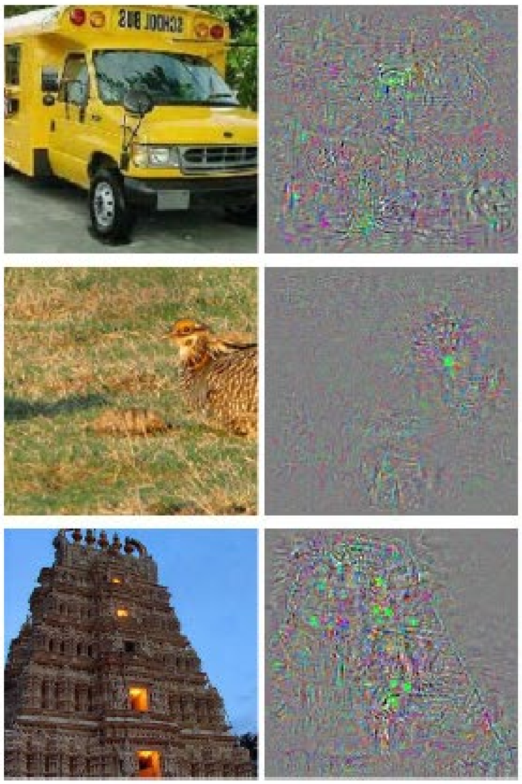


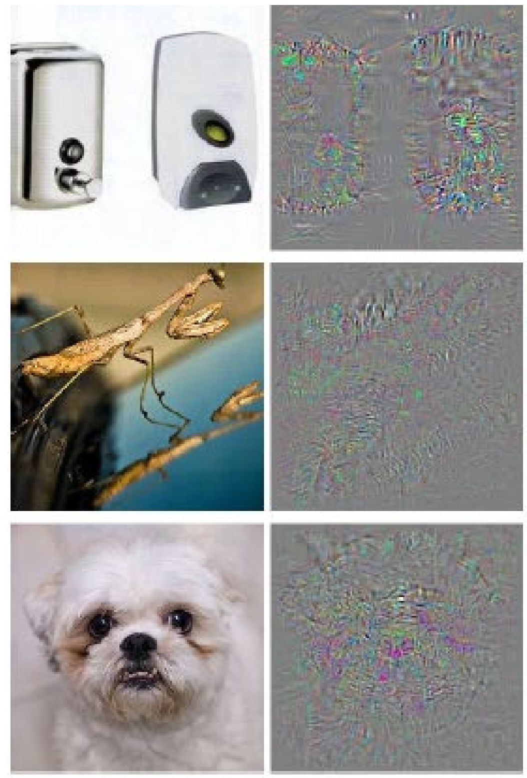
Fooling DL systems



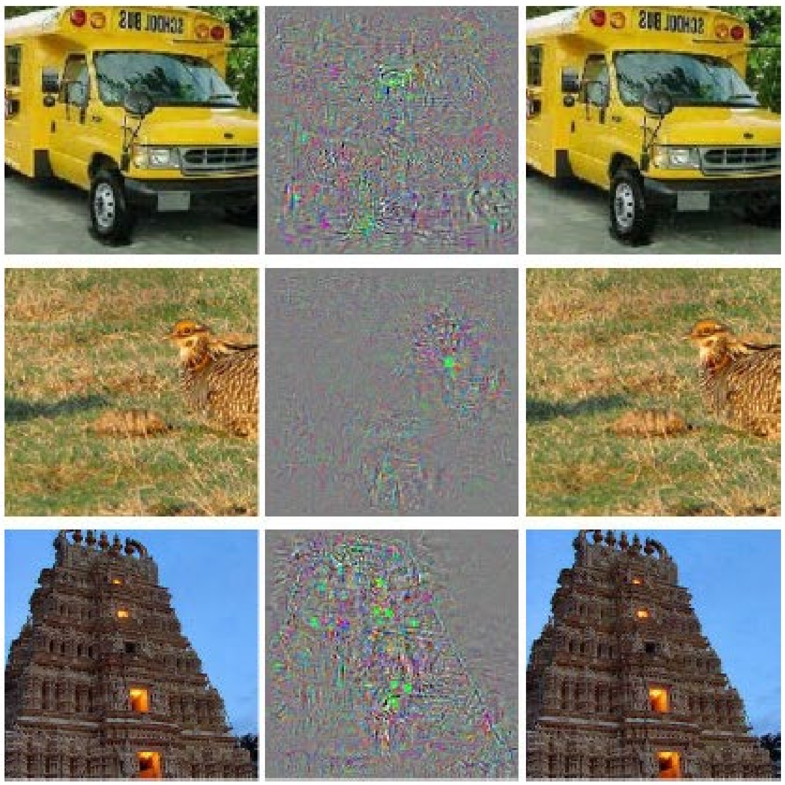


Fooling DL systems

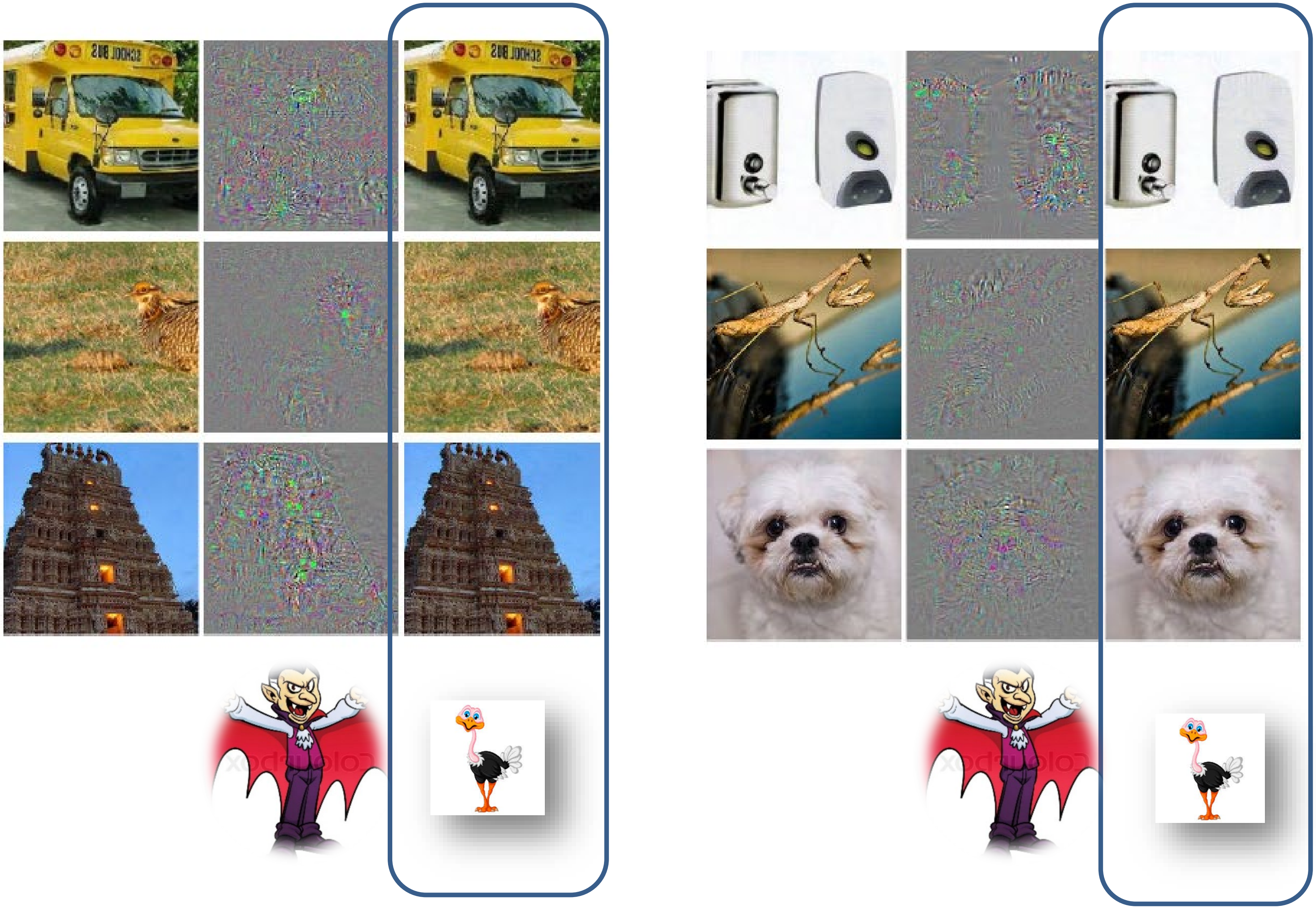




Fooling DL systems



Fooling DL systems



[Szegedy et al.] Intriguing properties of neural networks

Outline

Motivation

Adversarial attacks

Adversarial attacks

Untargeted adversarial examples

Given an input (X, C), an input $X' = X + P$ is an untargeted adversarial example iff NN misclassifies X' and P is small according to some metric.

Untargeted adversarial examples

Given an input $(\mathbf{X}, \mathbf{C})$, an input $X' = X + P$ is an untargeted adversarial example iff NN misclassifies X' and P is small according to some metric.

Untargeted adversarial examples

Given an input $(\mathbf{X}, \mathbf{C})$, an input $\mathbf{X'} = \mathbf{X} + \mathbf{P}$ is an untargeted adversarial example iff NN misclassifies X' and P is small according to some metric.

Untargeted adversarial examples

Given an input $(\mathbf{X}, \mathbf{C})$, an input $\mathbf{X'} = \mathbf{X} + \mathbf{P}$ is an untargeted adversarial example iff *NN misclassifies* X' and P is small according to some metric.

Untargeted adversarial examples

Original image

88% tabby cat

[Szegedy *et al.*] *Intriguing properties of neural networks*
[Athalye et al.]Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples

Untargeted adversarial examples

Original image **+** **P**erturbation

88% tabby cat

[Szegedy *et al.*] *Intriguing properties of neural networks*
[Athalye et al.]Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples

Untargeted adversarial examples

Original image + Perturbation = Perturbed image

88% tabby cat

[Szegedy *et al.*] *Intriguing properties of neural networks*
[Athalye et al.]Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples

Untargeted adversarial examples

Original image + Perturbation = Perturbed image

88% tabby cat

99% guacamole

[Szegedy *et al.*] *Intriguing properties of neural networks*
[Athalye et al.]Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples

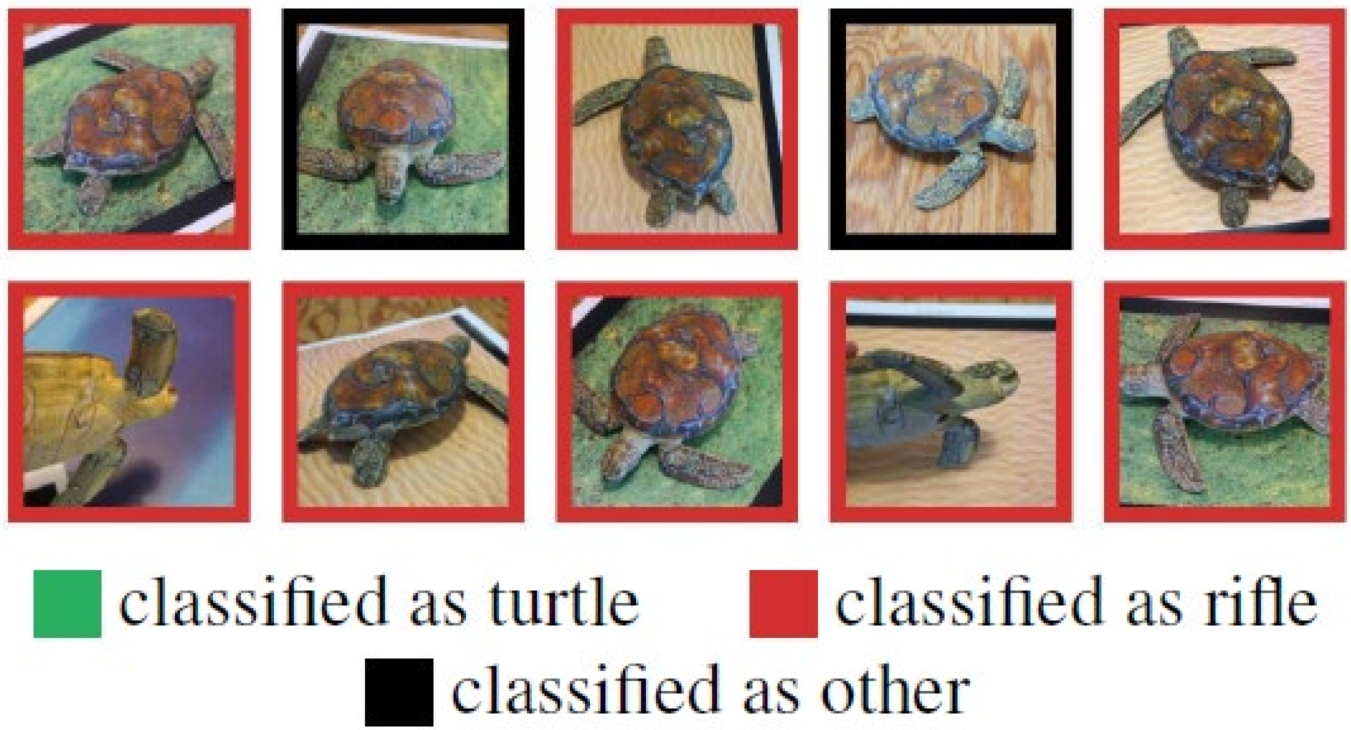
Beyond cats and dogs

[Eykholt at al.] Robust Physical-World Attacks on Deep Learning Visual Classification

Beyond cats and dogs

[Athalye at al.] Synthesizing Robust Adversarial Examples

Beyond cats and dogs

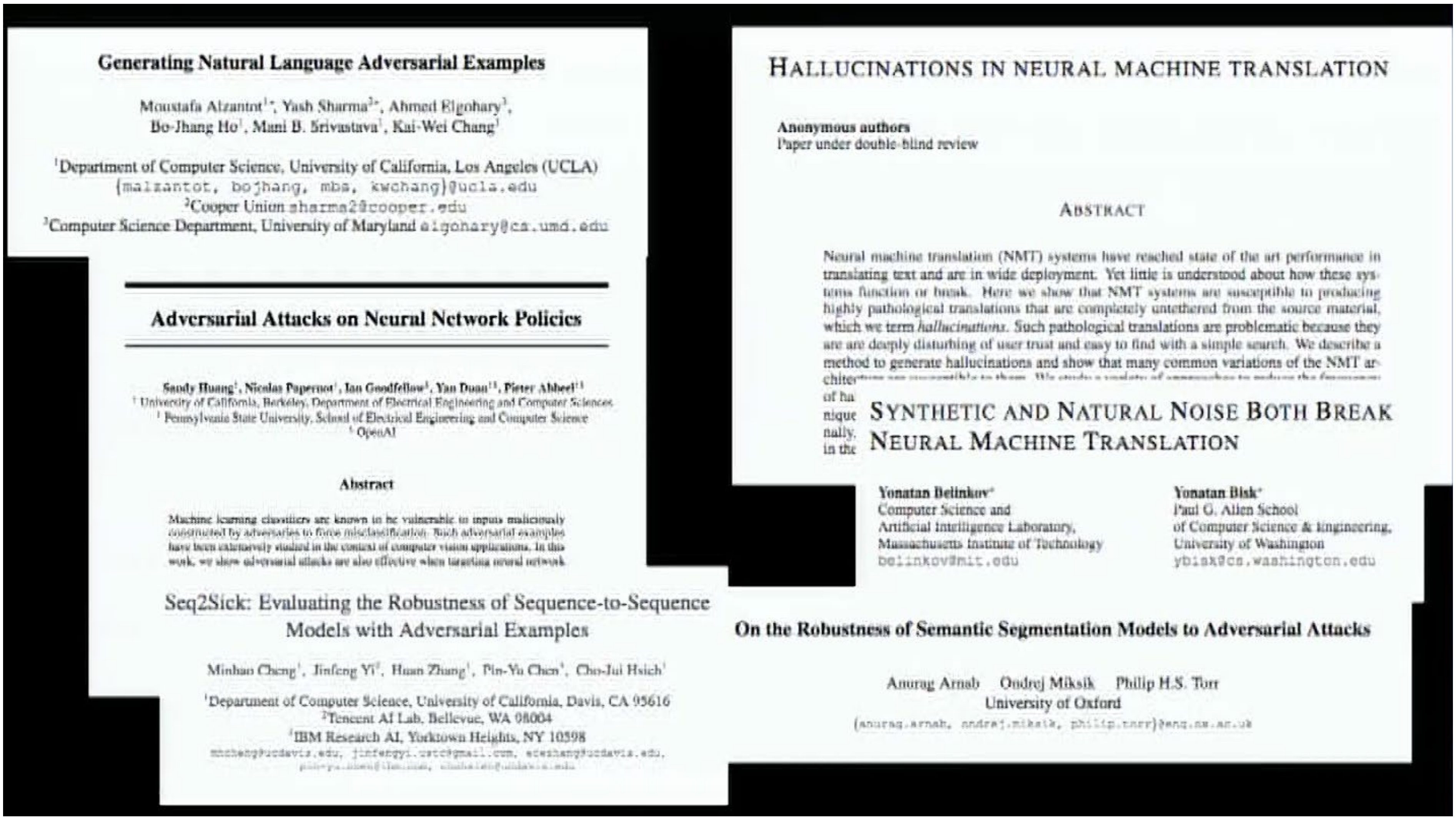


[Athalye at al.] Synthesizing Robust Adversarial Examples

Beyond cats and dogs

[Eykholt at al.] Robust Physical-World Attacks on Deep Learning Visual Classification

Beyond images



Generating Natural Language Adversarial Examples

Moustafa Alzantot[1*], Yash Sharma[2*], Ahmed Elgohary[3], Bo-Jhang Ho[1], Mani B. Srivastava[1], Kai-Wei Chang[1]

[1]Department of Computer Science, University of California, Los Angeles (UCLA)
{malzantot, bojhang, mbs, kwchang}@ucla.edu
[2]Cooper Union sharma2@cooper.edu
[3]Computer Science Department, University of Maryland elgohary@cs.umd.edu

Adversarial Attacks on Neural Network Policies

Sandy Huang[1], Nicolas Papernot[2], Ian Goodfellow[3], Yan Duan[1,3], Pieter Abbeel[1,3]
[1] University of California, Berkeley, Department of Electrical Engineering and Computer Sciences
[2] Pennsylvania State University, School of Electrical Engineering and Computer Science
[3] OpenAI

Abstract

Machine learning classifiers are known to be vulnerable to inputs maliciously constructed by adversaries to force misclassification. Such adversarial examples have been extensively studied in the context of computer vision applications. In this work, we show adversarial attacks are also effective when targeting neural network

Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples

Minhao Cheng[1], Jinfeng Yi[2], Huan Zhang[1], Pin-Yu Chen[3], Cho-Jui Hsieh[1]

[1]Department of Computer Science, University of California, Davis, CA 95616
[2]Tencent AI Lab, Bellevue, WA 98004
[3]IBM Research AI, Yorktown Heights, NY 10598

HALLUCINATIONS IN NEURAL MACHINE TRANSLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural machine translation (NMT) systems have reached state of the art performance in translating text and are in wide deployment. Yet little is understood about how these systems function or break. Here we show that NMT systems are susceptible to producing highly pathological translations that are completely untethered from the source material, which we term *hallucinations*. Such pathological translations are problematic because they are are deeply disturbing of user trust and easy to find with a simple search. We describe a method to generate hallucinations and show that many common variations of the NMT ar-

SYNTHETIC AND NATURAL NOISE BOTH BREAK NEURAL MACHINE TRANSLATION

Yonatan Belinkov*
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology
belinkov@mit.edu

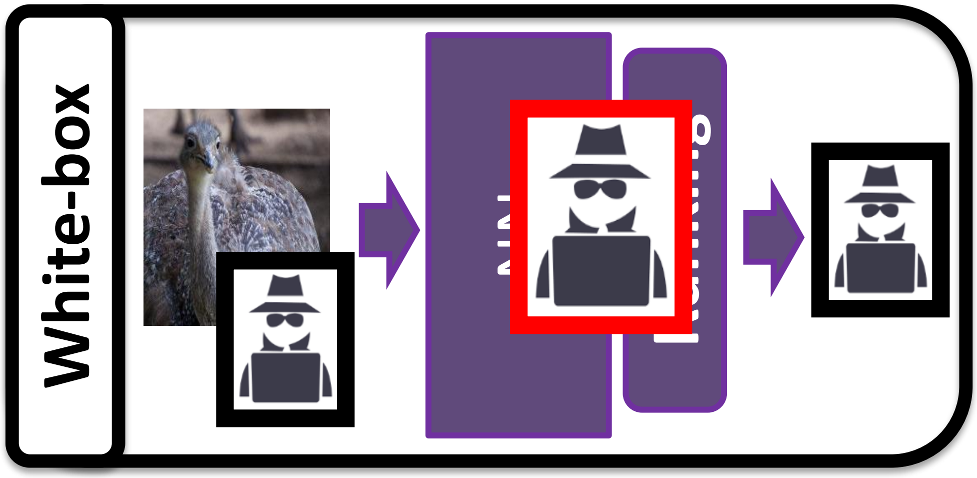
Yonatan Bisk*
Paul G. Allen School of Computer Science & Engineering, University of Washington
ybisk@cs.washington.edu

On the Robustness of Semantic Segmentation Models to Adversarial Attacks

Anurag Arnab Ondrej Miksik Philip H.S. Torr
University of Oxford

[Nicholas Carlini] *On (In-)* security of Deep Learning Models

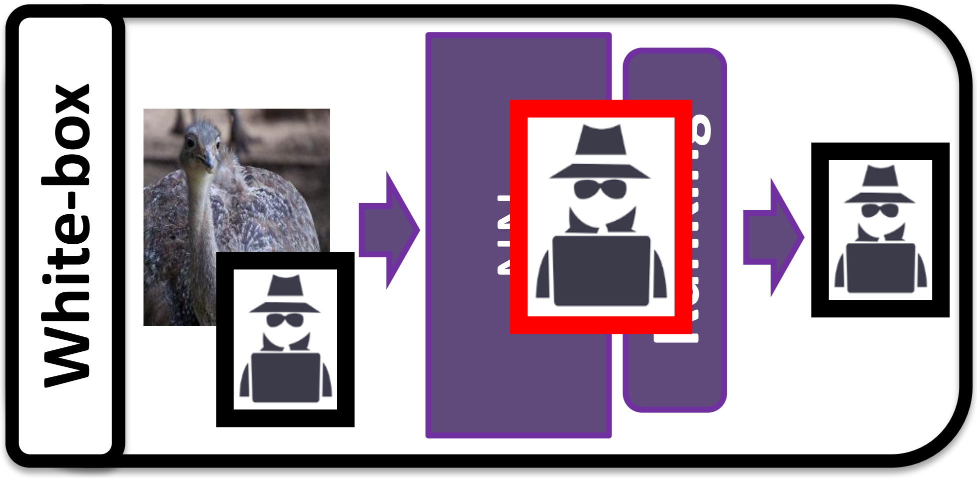
White-box vs Black-box Attacks



[Goodfellow et al., Szegedy et al.]

[Papernot et al., 2016a, 2016b]

White-box vs Black-box Attacks

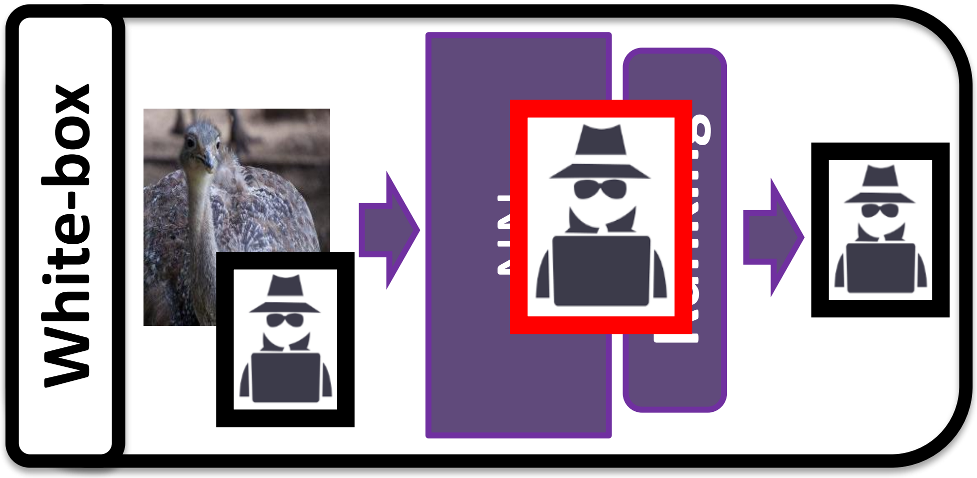


[Goodfellow et al., Szegedy et al.]

[Papernot et al., 2016a, 2016b]

Gradient-based methods that generate adversarial images by perturbing the gradients of the loss function w.r.t. the input image

White-box vs Black-box Attacks

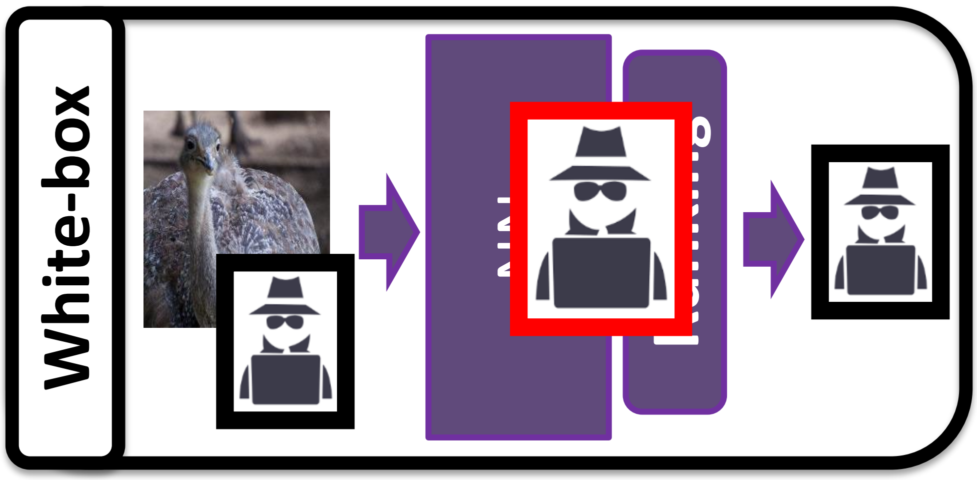


[Goodfellow et al., Szegedy et al.]

[Papernot et al., 2016a, 2016b]

Gradient-based methods that generate adversarial images by perturbing the gradients of the loss function w.r.t. the input image

White-box vs Black-box Attacks



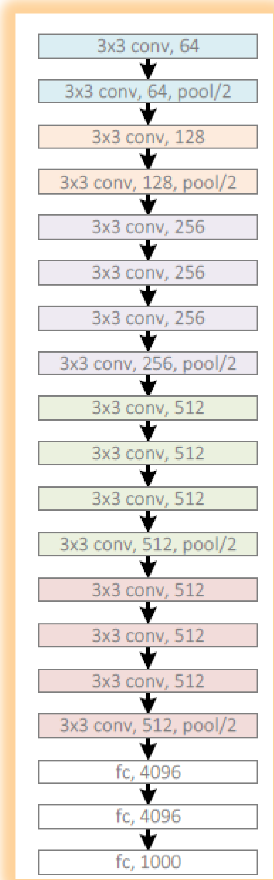
[Goodfellow et al., Szegedy et al.]

[Papernot et al., 2016a, 2016b]

Gradient-based methods that generate adversarial images by perturbing the gradients of the loss function w.r.t. the input image

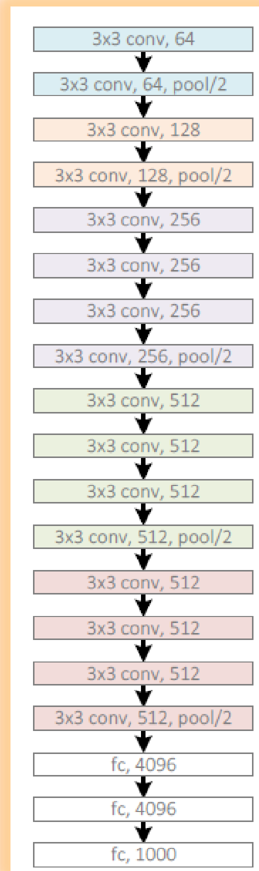
- More realistic and applicable model
- Challenging because of weak adversaries: no knowledge of the network architecture
- Previous attacks require 'transferability' assumption on adversarial examples
- GAN based attacks

New research sub-area



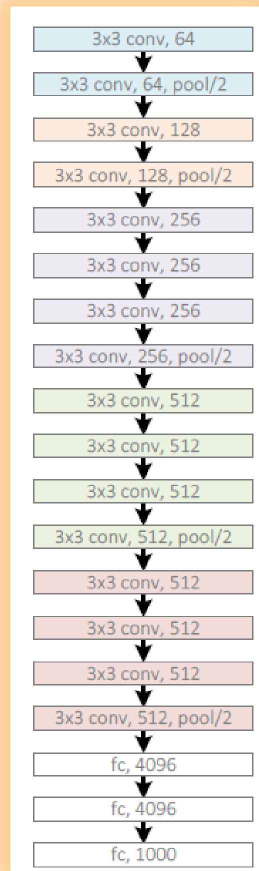
New research sub-area

Attacks



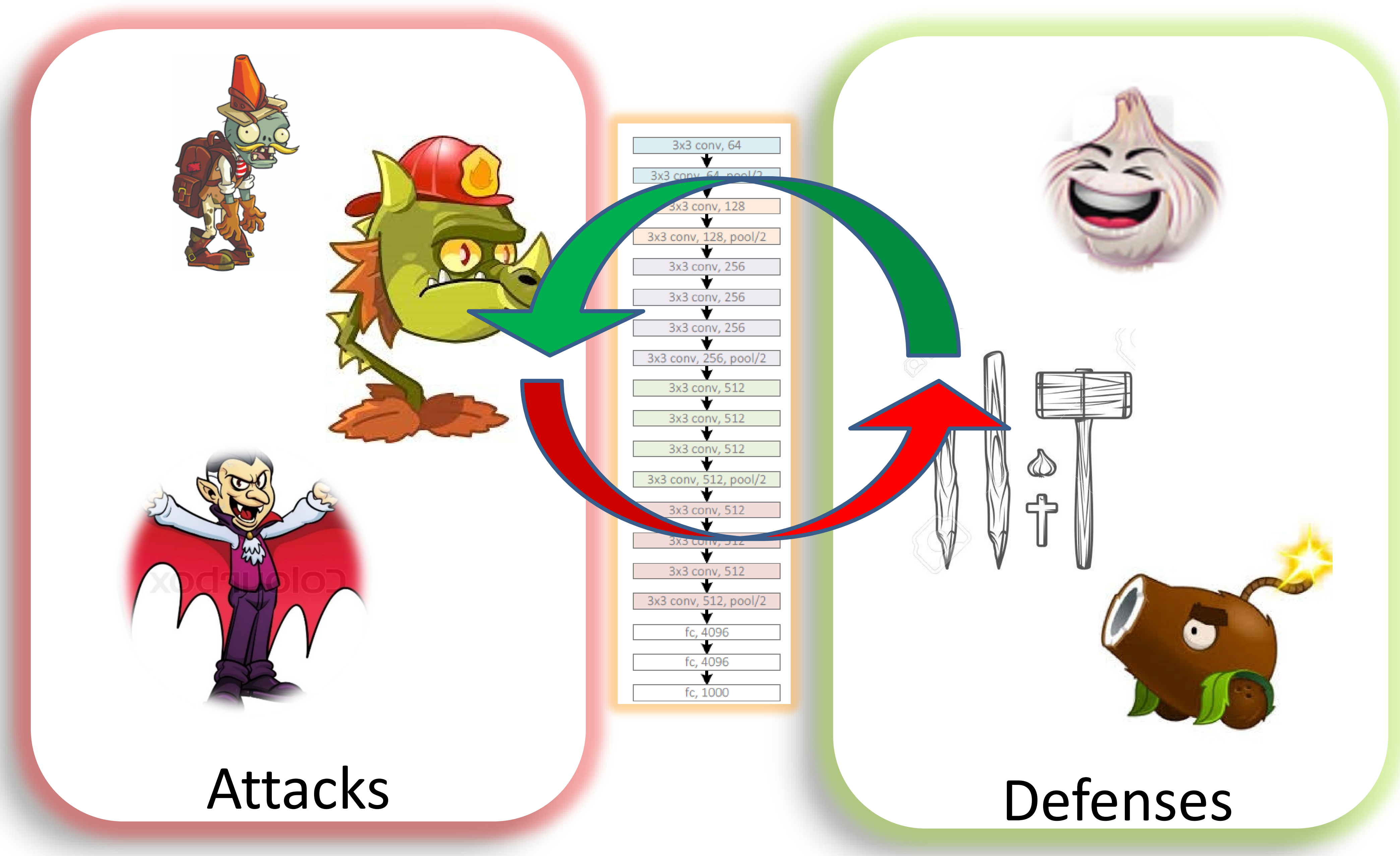
New research sub-area

Attacks

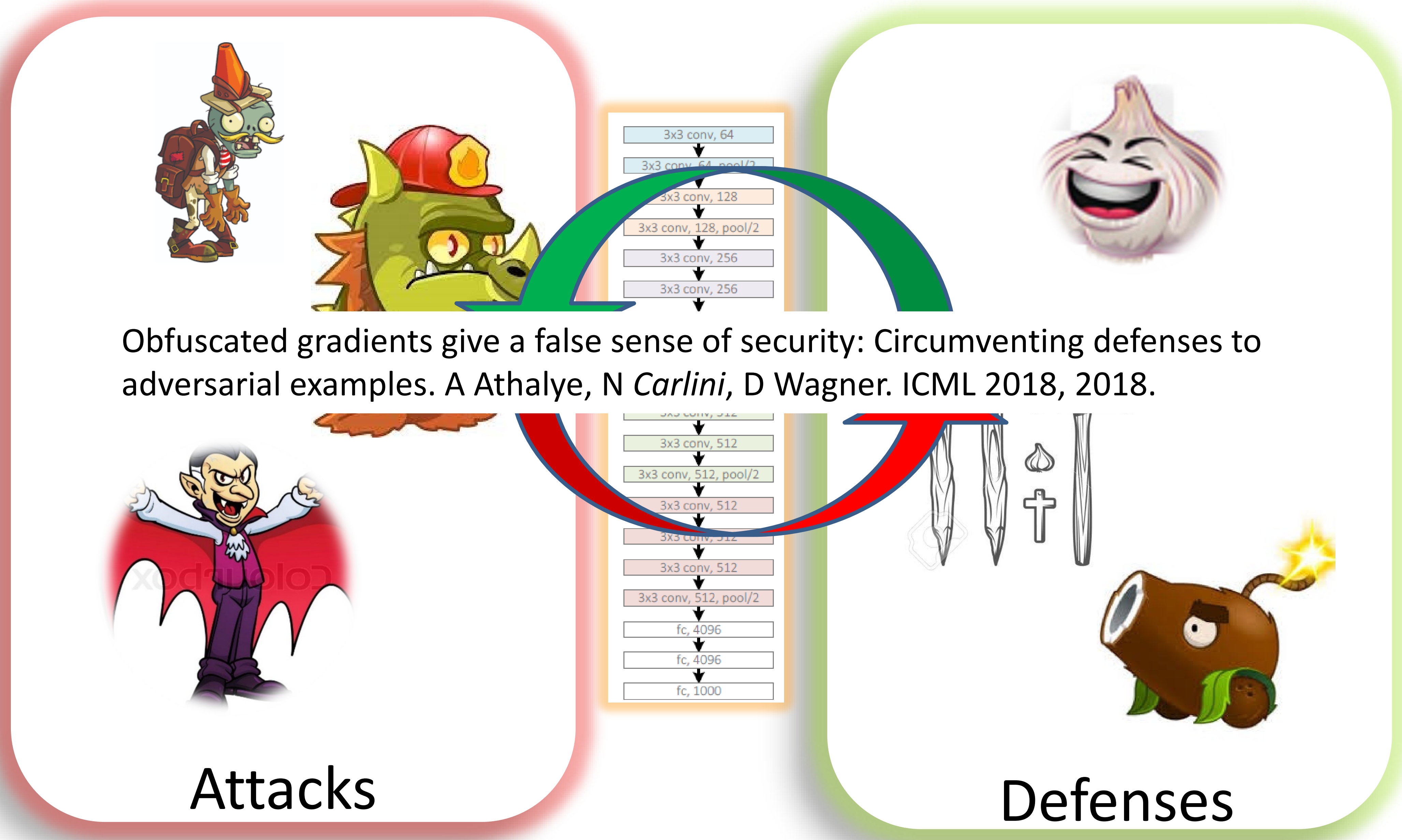


Defenses

New research sub-area

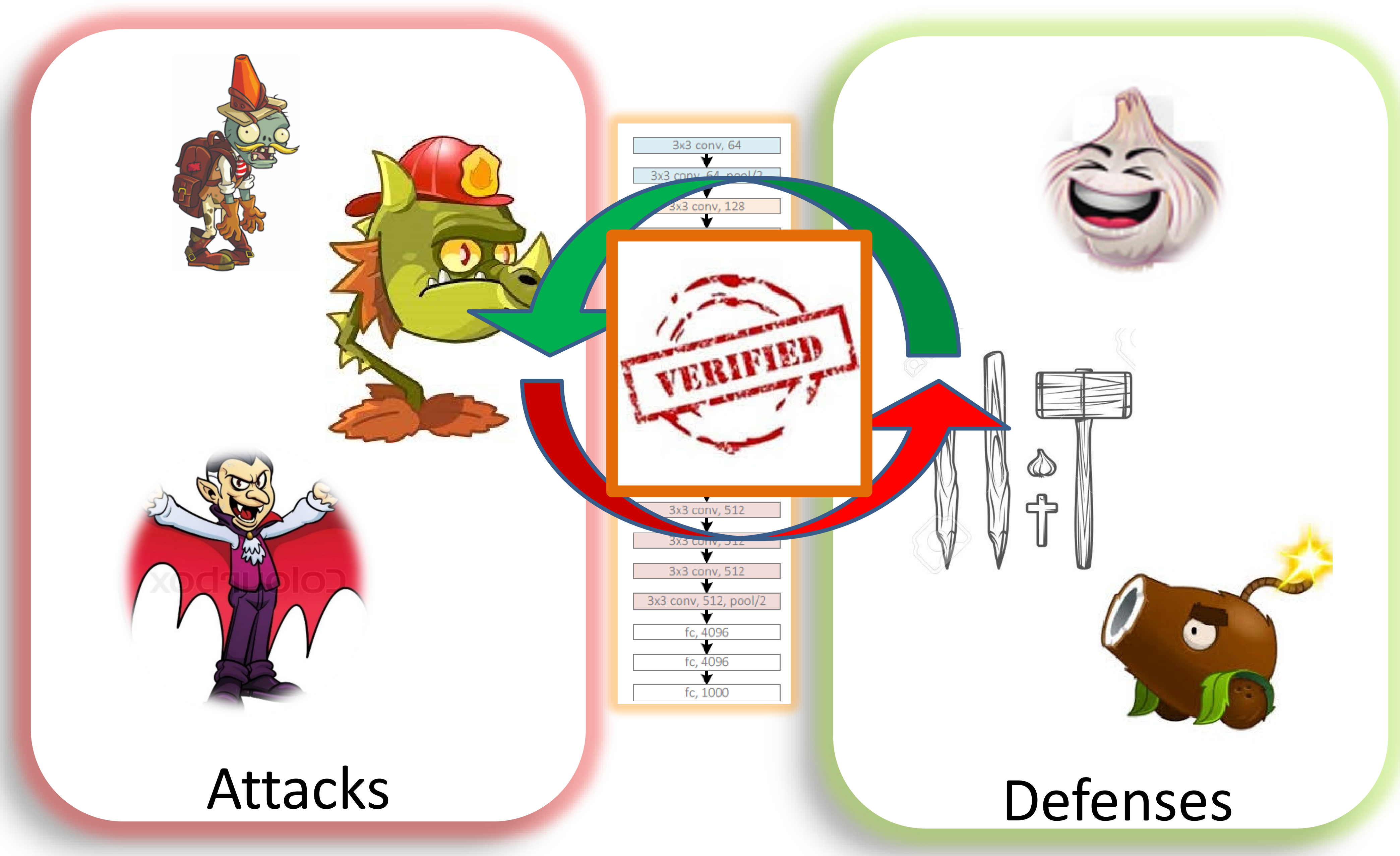


New research sub-area



Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. A Athalye, N *Carlini*, D Wagner. ICML 2018, 2018.

New research sub-area

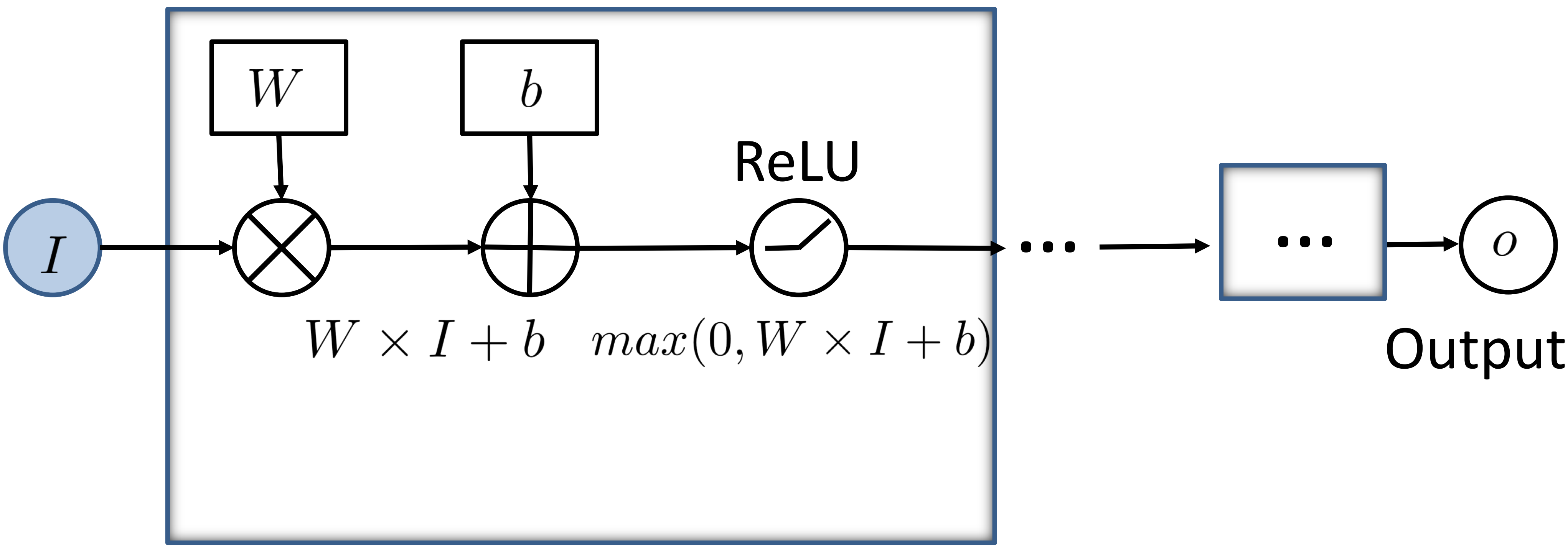


Outline

Motivation
Adversarial attacks
Verification methods

Network verification problem

Network verification problem



Input

- features
- images

Network verification problem

NNs is defined as $I^n \rightarrow O^m$

$pre(x)$ and $post(y)$ are logic formulas

pre defines *preconditions* on the inputs

$post$ defines *postconditions* on the output

Network verification problem

Given conditions *pre* and *post*, a property is:

$$\forall x. \forall y. (pre(x) \wedge y = NN(x)) \implies post(y)$$

Network verification problem

To find a counterexample:

$$pre(x) \wedge (y = NN(x)) \wedge \neg post(y)$$

Network verification problem

Let x' is a given  classified as 'cat'.

$$pre(x) := |x - x'| \leq \epsilon$$

$$post(y) := \text{`cat'}$$

$$\forall x. \forall y. (pre(x) \wedge y = NN(x)) \implies post(y)$$

Verification methods

Verification

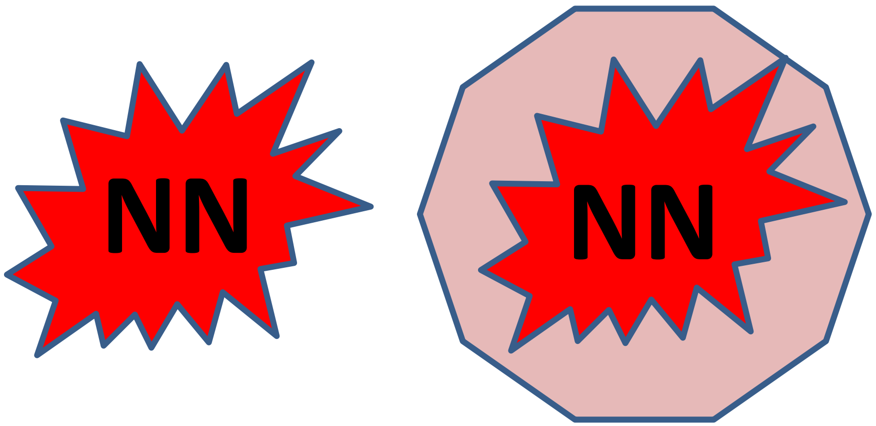
Verification methods

Verification roadmap

Verification roadmap

Exact
Methods

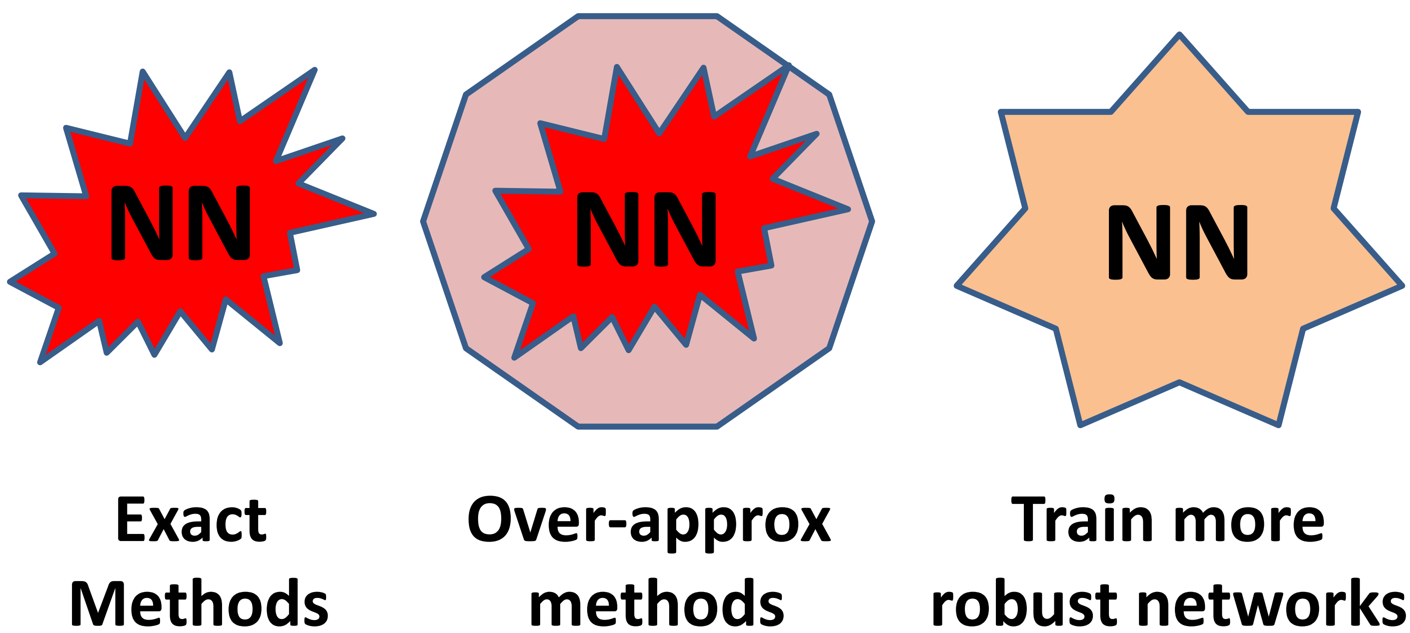
Verification roadmap



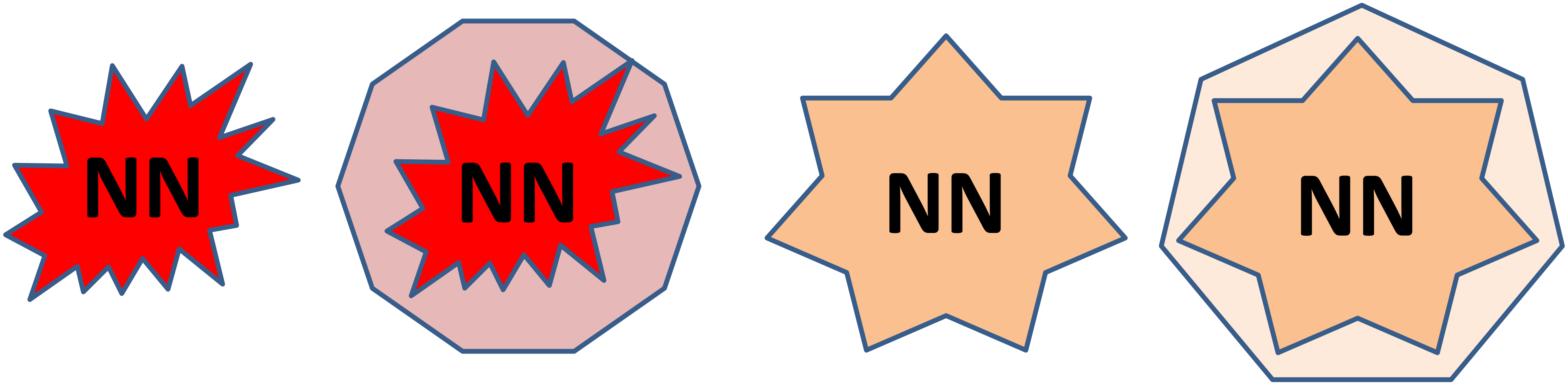
Exact Methods

Over-approx methods

Verification roadmap



Verification roadmap



Exact Methods

Over-approx methods

Train more robust networks

Certified networks

Do we augment training?

Do we augment training?

no	yes

Do we augment training?

no	yes
Sound and complete	
Sound, not complete	

Do we augment training?

no	yes
Sound and complete	Adversarial training
Sound, not complete	Certification of NNs

Do we augment training?

no	yes
Sound and complete	Adversarial training
Sound, not complete	Certification of NNs

Easier-to-verify networks

Do we augment training?

no	yes
Sound and complete	Adversarial training
Sound, not complete	Certification of NNs

Easier-to-verify networks

Sound and complete methods

Strength: Prove whether a property holds

- R. Ehlers. Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks,2017
- R. Bunel, I. Turksaslan, P. Torr, P. Kohli, and P. Kumar. Piecewise Linear Neural Network Verification: A Comparative Study, 2017.
- G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks.2017
- A. Lomuscio and L. Maganti. An Approach to Reachability Analysis for Feed-Forward ReLU Neural Networks, 2017.

Sound and complete methods

$$pre(x) \wedge (y = NN(x)) \wedge \neg post(y)$$

Sound and complete methods

$$pre(x) \wedge (y = NN(x)) \wedge \neg post(y)$$

$$\Downarrow$$

$$\mathrm{SMT}(pre(x)) \wedge \mathrm{SMT}(y = NN(x)) \wedge \mathrm{SMT}(\neg post(y))$$

Sound and complete methods

$$pre(x) \wedge (y = NN(x)) \wedge \neg post(y)$$

$$\Downarrow$$

$$\mathrm{SMT}(pre(x)) \wedge \mathrm{SMT}(y = NN(x)) \wedge \mathrm{SMT}(\neg post(y))$$

$$\Downarrow$$

SMT solver

Sound and complete methods

$$pre(x) \wedge (y = NN(x)) \wedge \neg post(y)$$

(will discuss for BNNs+SAT)

$$\mathrm{SMT}(pre(x)) \wedge \mathrm{SMT}(y = NN(x)) \wedge \mathrm{SMT}(\neg post(y))$$

SMT solver

Sound and complete methods

$$pre(x) \wedge (y = NN(x)) \wedge \neg post(y)$$

$$\mathrm{SMT}(pre(x)) \wedge \mathrm{SMT}(y = NN(x)) \wedge \mathrm{SMT}(\neg post(y))$$

SMT solver (or Reluplex, Planet, etc)

Sound and complete methods

Limitation: scalability (up to 1000 neurons)

Do we augment training?

no	yes
Sound and complete	Adversarial training
Sound, not complete	Certification of NNs

Easier-to-verify networks

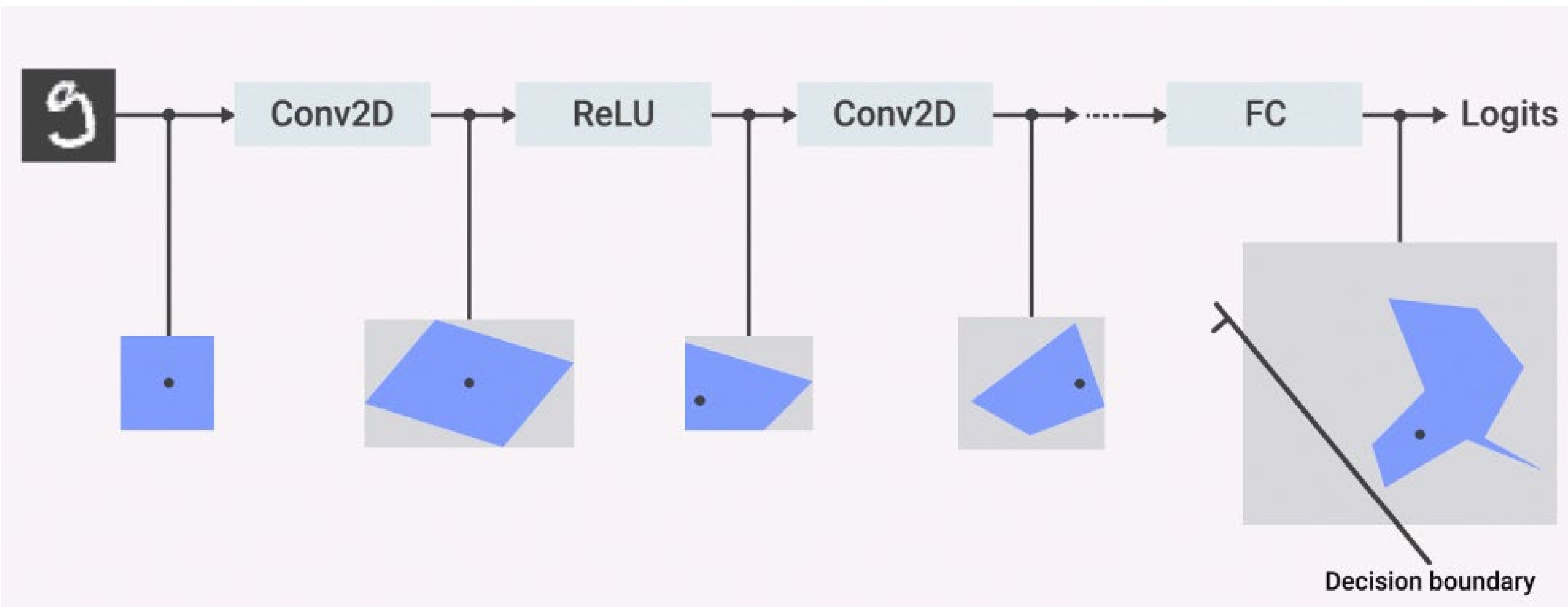
Sound and incomplete methods

Strength: Prove that a property holds
(can return `*do not know*')

- Singh, G., Gehr, T., Mirman, M., Puschel, M., and Vechev, M. T. Fast and effective robustness certification.
- Zhang, H., Weng, T., Chen, P., Hsieh, C., and Daniel, L. Efficient neural network robustness certification with general activation functions.
- Weng, T., Zhang, H., Chen, H., Song, Z., Hsieh, C., Daniel, L., Boning, D. S., and Dhillon, I. S. Towards fast computation of certified robustness for relu networks
- T. Gehr, M. Mirman, D. Drachsler-Cohen, E. Tsankov, S. Chaudhuri, and M. Vechev. AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation.

Sound and incomplete methods

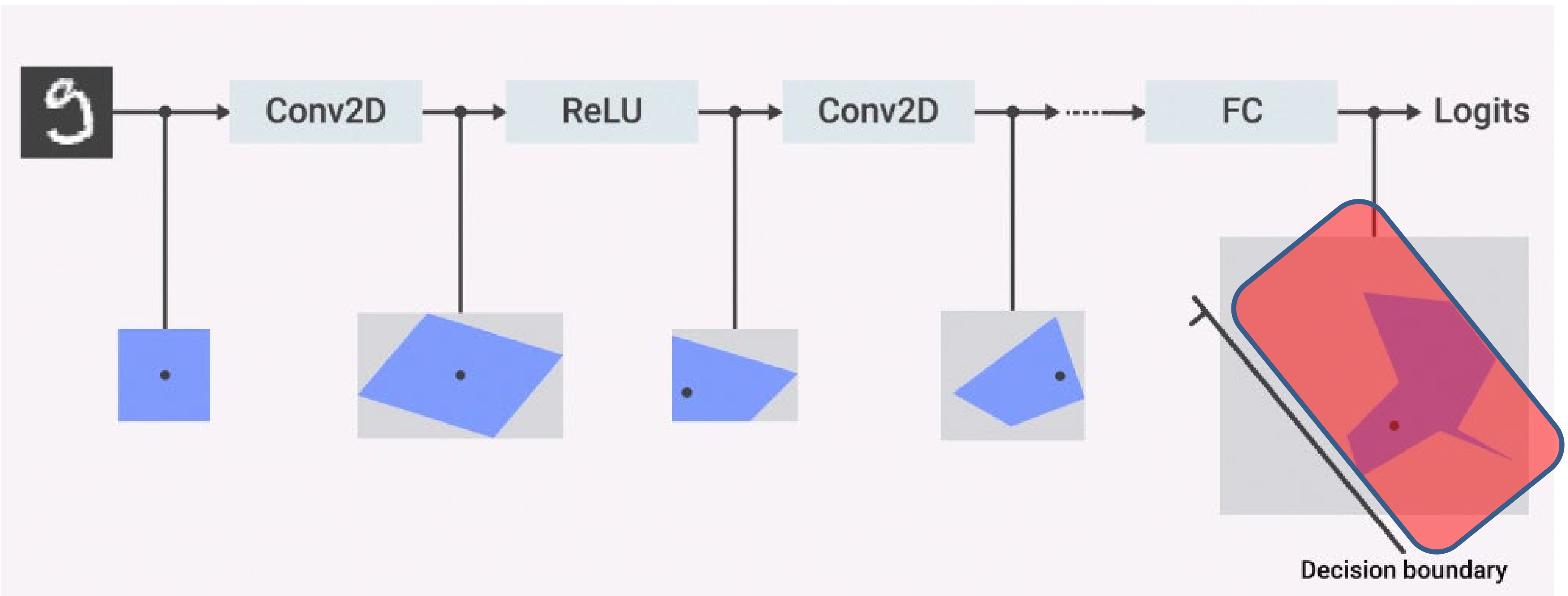
Based on over-approximation of the output space



https://medium.com/@deepmindsafetyresearch/towards-robust-and-verified-ai-specification-testing-robust-training-and-formal-verification-69bd1bc48bda

Sound and incomplete methods

Based on over-approximation of the output space

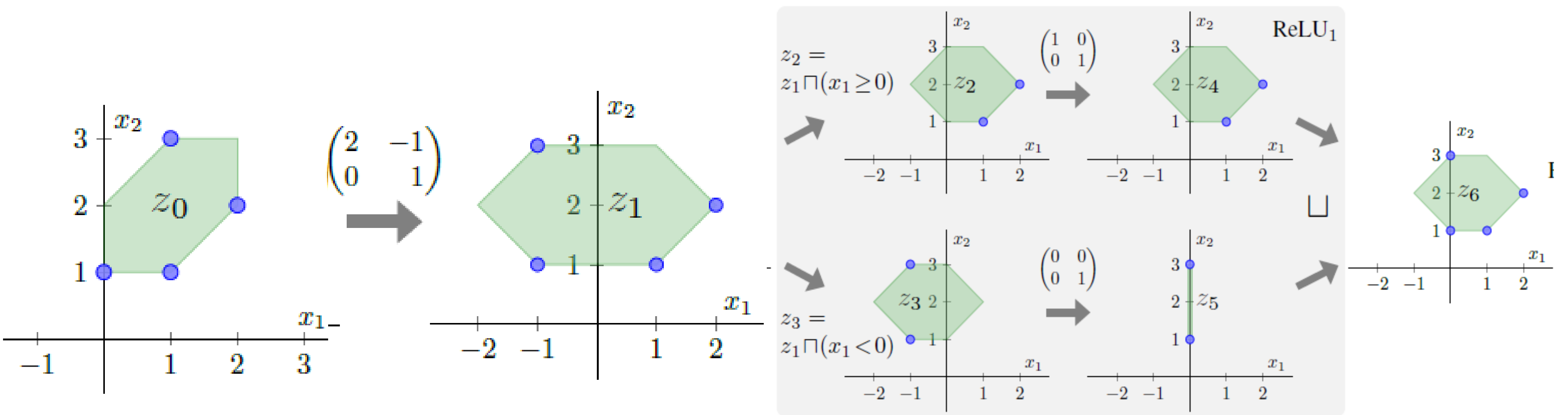


https://medium.com/@deepmindsafetyresearch/towards-robust-and-verified-ai-specification-testing-robust-training-and-formal-verification-69bd1bc48bda

Sound and incomplete methods

Based on over-approximation of the output space

*Input *Linear Transformer *ReLU



[Gehr et al.] AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation

Sound and incomplete methods

Limitation: scalability (up to 10000 neurons)

Do we augment training?

no	yes
Sound and complete	Adversarial training
Sound, not complete	Certification of NNs

Easier-to-verify networks

Adversarial training methods

Strength: (empirically) improve robustness of NNs

- Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale, 2017.
- Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples.2017
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu.Towards deep learning models resistant to adversarial attacks, 2018.

Adversarial training methods

$$\min_{W} \sum_{I,L \in \mathcal{D}} \max_{\delta \in \Delta} Loss(I + \delta, L, W)$$

Adversarial training methods

$$\min_{W} \sum_{I,L \in \mathcal{D}} \max_{\delta \in \Delta} Loss(I + \delta, L, W)$$

Adversarial training methods

$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I + \delta, L, W)}$$

Adversarial training methods

$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I + \delta, L, W)}$$

- Use gradient-based search, e.g. PGD, to solve inner optimization

Adversarial training methods

$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I+\delta, L, W)}$$

1. Select minibatch B
2. For each (I,L) ∈ B compute an adversarial example δ*
3. Update parameters at I+ δ*

Adversarial training methods

Limitation: no guarantees on robustness

Do we augment training?

no	yes
Sound and complete	Adversarial training
Sound, not complete	Certification of NNs

Easier-to-verify networks

Certified training methods

Strength: prove that a property holds
(but can produce false negatives)

- Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope, 2018
- Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. 2018
- Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. 2018

Certification of NNs

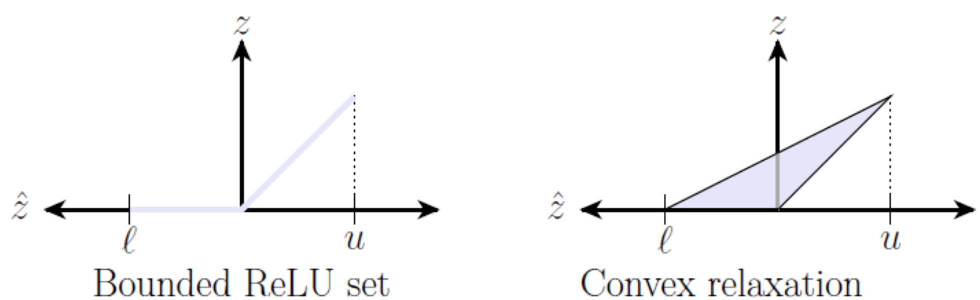
$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I + \delta, L, W)}$$

Certification of NNs

$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I + \delta, L, W)}$$

Certification of NNs

$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I + \delta, L, W)}$$



- Use a convex relaxation inner optimization
- Use gradients of this relaxation in the training procedure

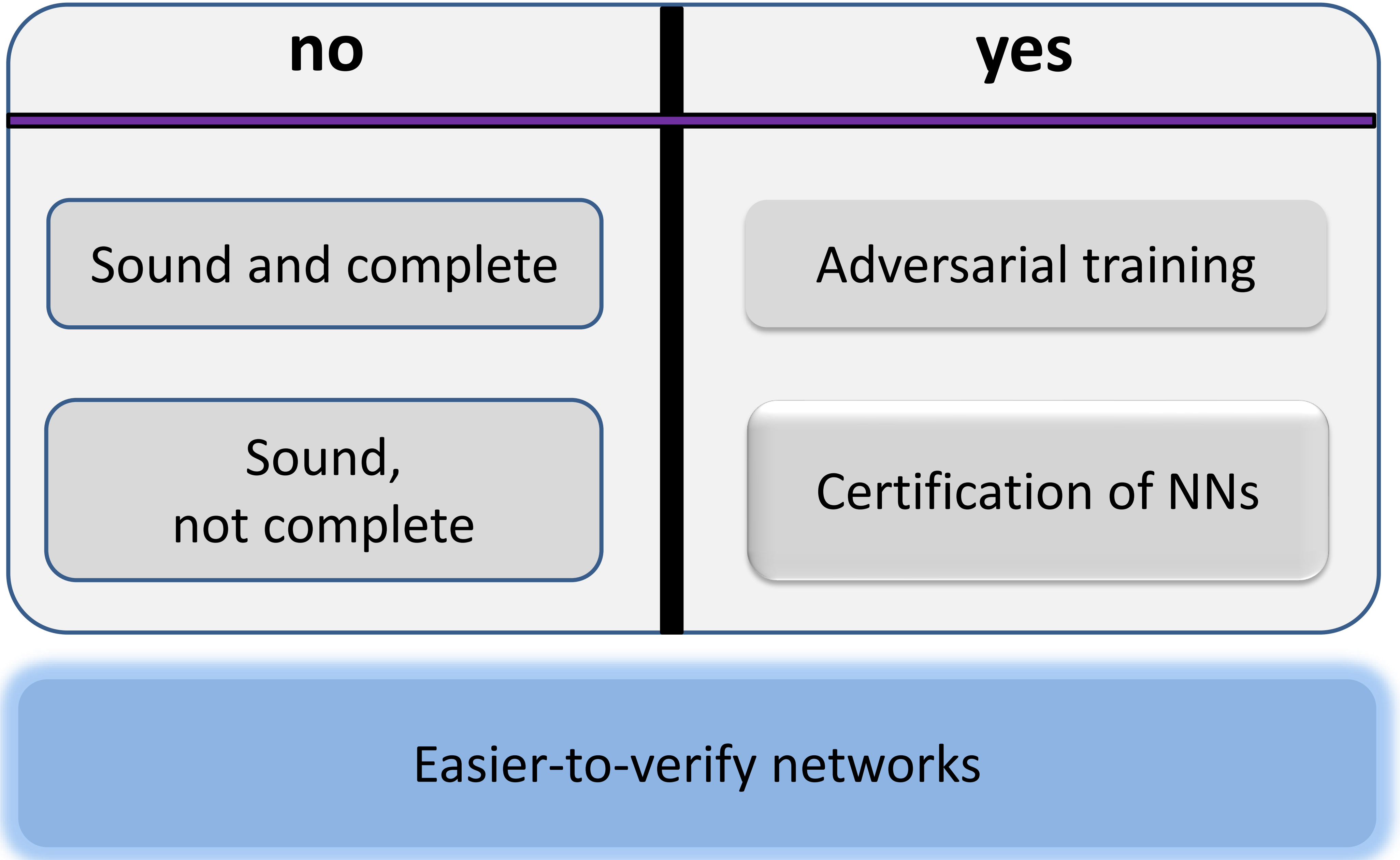
Certification of NNs

Limitation:

- work with relaxation, an upper bound on the can be quite loose
- the loss is much more complex than in a non-adv training

(accuracy drops, scalability issues)

Do we augment training?



Easier-to-verify networks

Strength: train a network that is easier to verify for existing decision procedures

- Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability
  Kai Y. Xiao, Vincent Tjeng, Nur Muhammad (Mahi) Shafiullah, Aleksander Madry, ICLR'19
- In Search for a SAT-friendly Binarized Neural Network Architecture
  Nina Narodytska, Hongce Zhang, Aarti Gupta, Toby Walsh, ICLR20

Easier-to-verify networks

Limitation: no guarantees on robustness

Do we augment training?

no	yes
Sound and complete	

Easier-to-verify networks

Do we augment training?

no	yes
Sound and complete	

Binarized Neural Networks

Easier-to-verify networks

Why BNNs?

Binarized neural networks: Training deep **neural networks** with weights and activations constrained to+ 1 or-1
M Courbariaux, I Hubara, D Soudry, R El-Yaniv… - arXiv preprint arXiv …, 2016 - arxiv.org
We introduce a method to train Binarized Neural Networks (BNNs)-neural networks with binary weights and activations at run-time. At training-time the binary weights and activations are used for computing the parameters gradients. During the forward pass, BNNs drastically …
☆ 99 Cited by 925 Related articles All 9 versions »

Binarized neural networks
I Hubara, M Courbariaux, D Soudry… - Advances in **neural** …, 2016 - papers.nips.cc
We introduce a method to train Binarized Neural Networks (BNNs)-neural networks with binary weights and activations at run-time. At train-time the binary weights and activations are used for computing the parameter gradients. During the forward pass, BNNs drastically …
☆ 99 Cited by 470 Related articles All 5 versions »

Xnor-net: Imagenet classification using binary convolutional **neural networks**
M Rastegari, V Ordonez, J Redmon… - European Conference on …, 2016 - Springer
… Because, at inference we only perform forward propagation with the **binarized** weights … Similar to **binarization** in the forward pass, we can **binarize** \(g^{in}\) in the backward pass … Our **binarization** technique is general, we can use any CNN architecture …
☆ 99 Cited by 1373 Related articles All 8 versions

Compactness

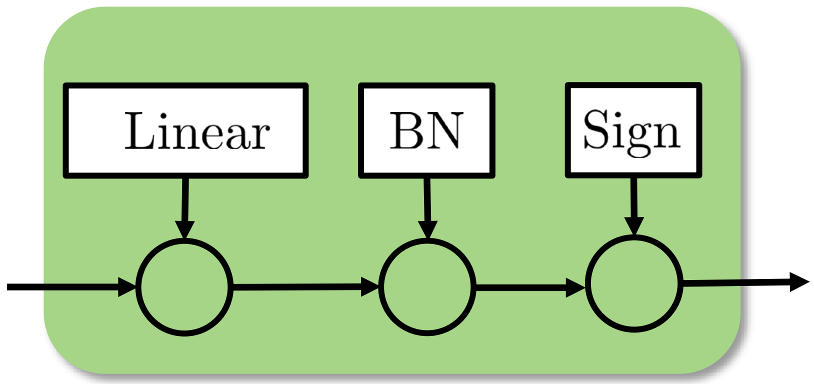
- Only 1 bit per weight, {-1,1}
- Can be deployed on embedded devices

Inference efficiency

- fast binary matrix multiplication
  (7X speed up on GPU)
- "Accelerating Binarized Neural Networks: Comparison of FPGA, CPU, GPU, and ASIC" IEEE'2016

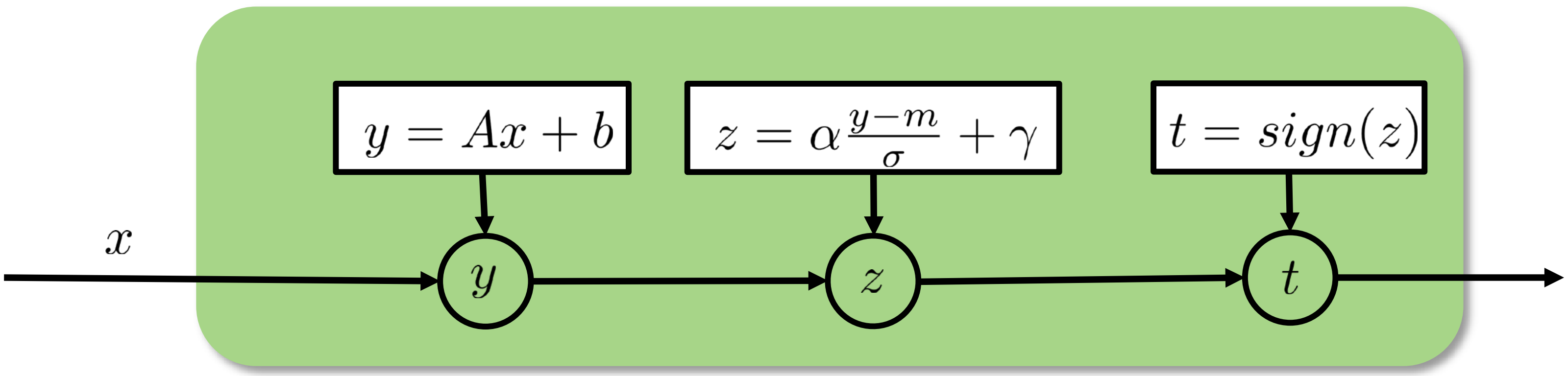
Structure of BNNs

Binarized Neural Networks

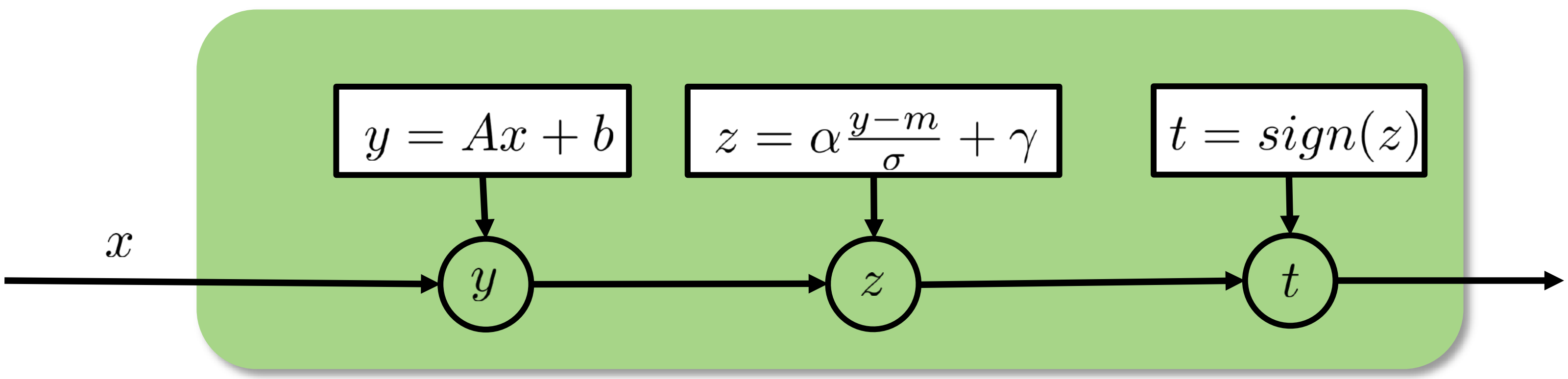


Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1
Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, Yoshua Bengio

Binarized Neural Networks

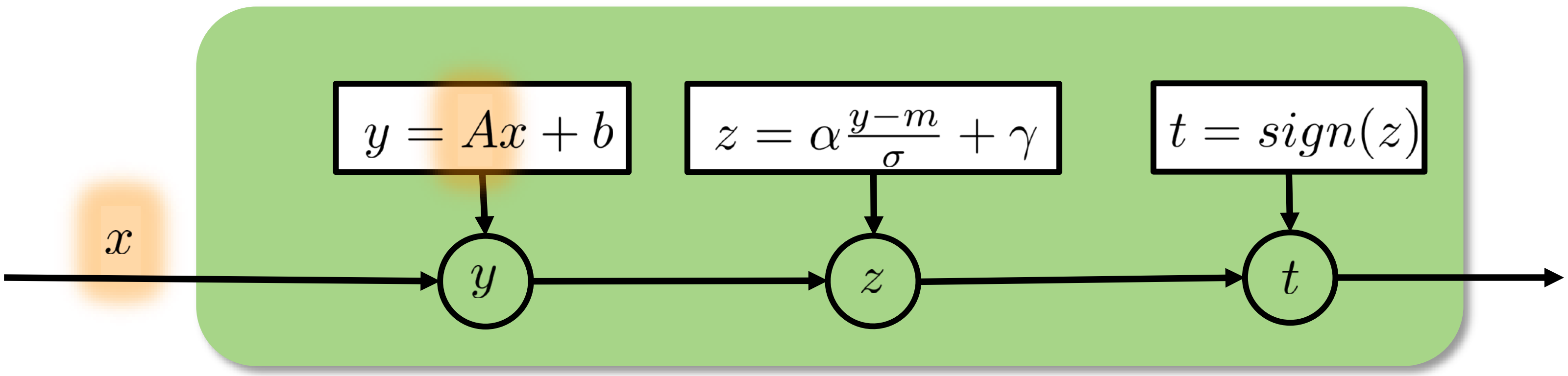


Binarized Neural Networks



$x, a_{i,j} \in \{-1, 1\}$ $\quad$ $b, \alpha, m, \sigma, \gamma \in \mathbf{R}$

Binarized Neural Networks



$x, a_{i,j} \in \{-1, 1\}$ $\quad b, \alpha, m, \sigma, \gamma \in \mathbf{R}$

BNNs and logic-based reasoning

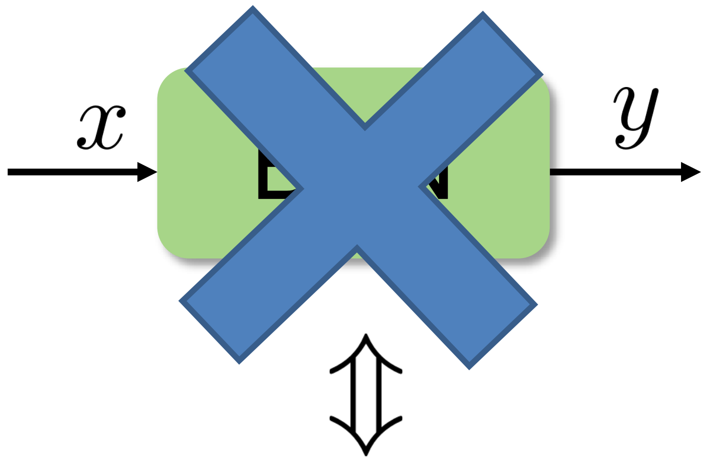
BNNs and Logic

BNNs and Logic

$$\Updownarrow$$

$$SAT(y = BNN(x))$$

BNNs and Logic



$$SAT(y = BNN(x))$$

BNNs and Logic

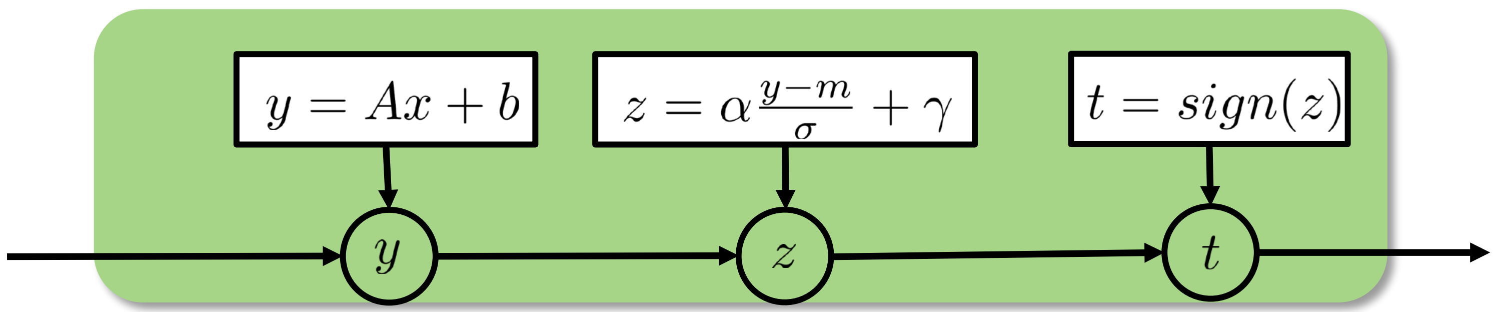
$$SAT(y = BNN(x))$$

BNNs and Logic

$$BinBNN(x, y) := SAT(y = BNN(x))$$

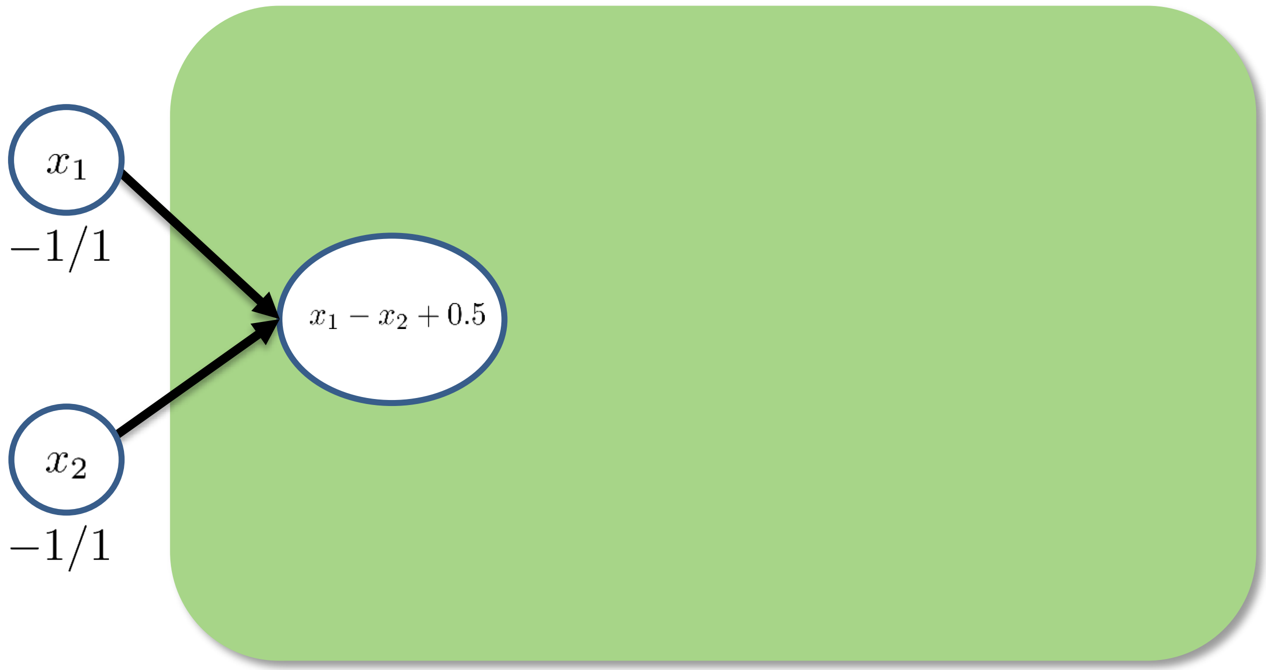
Translation: BNN to SAT

Translation: BNN to SAT

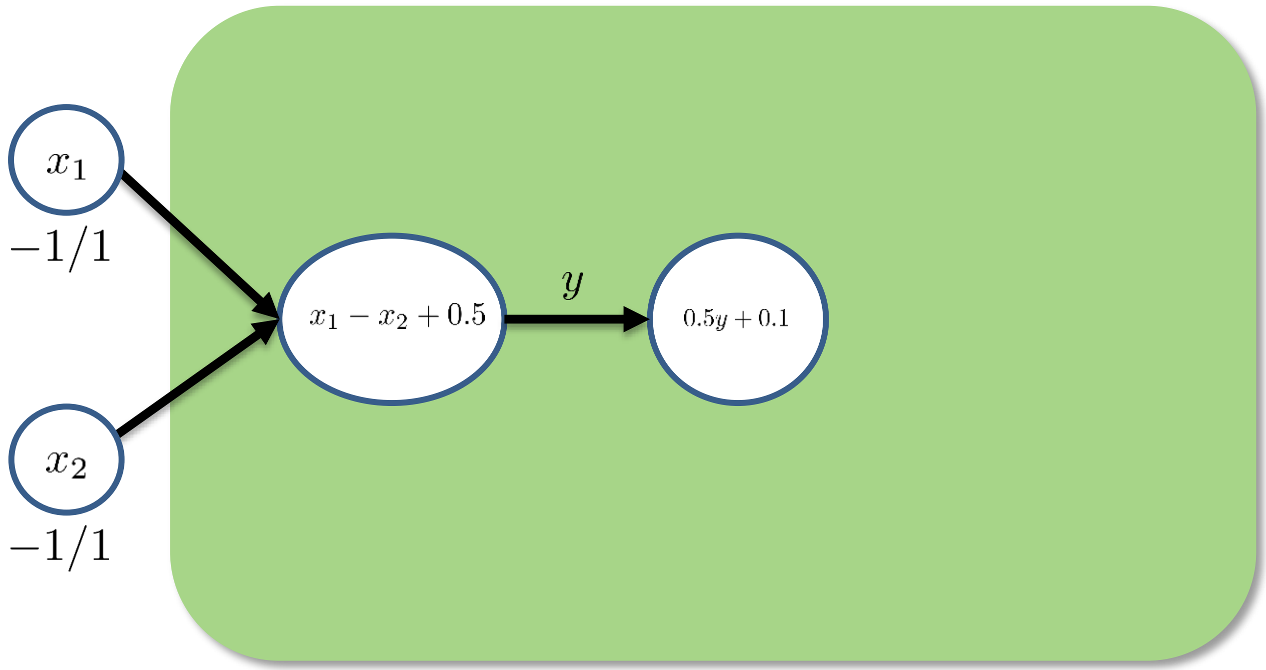


Translation: BNN to SAT

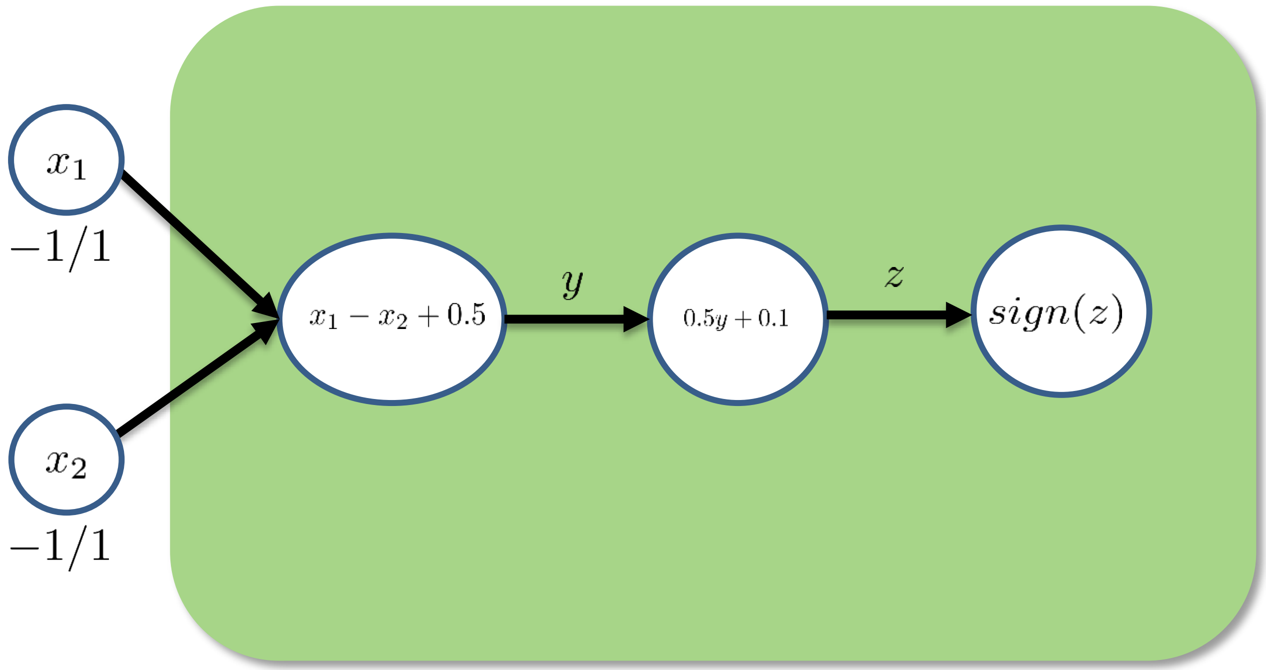
Translation: BNN to SAT



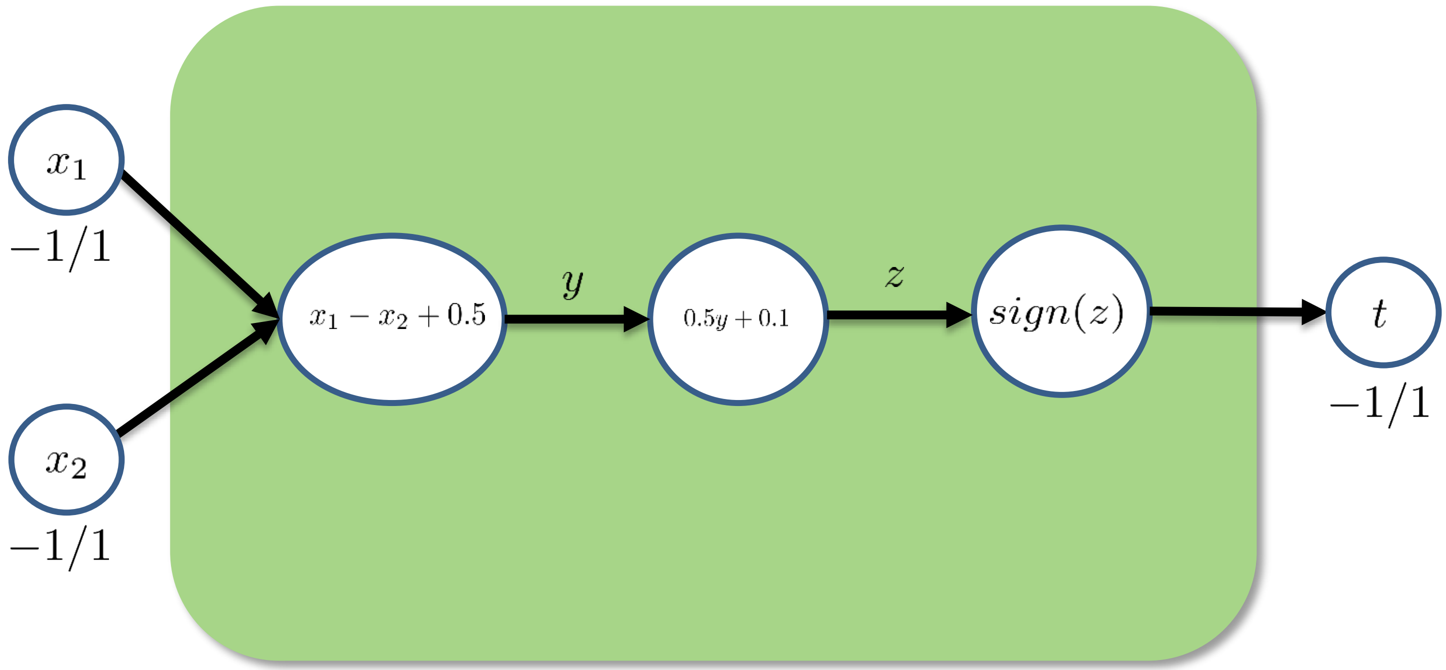
Translation: BNN to SAT



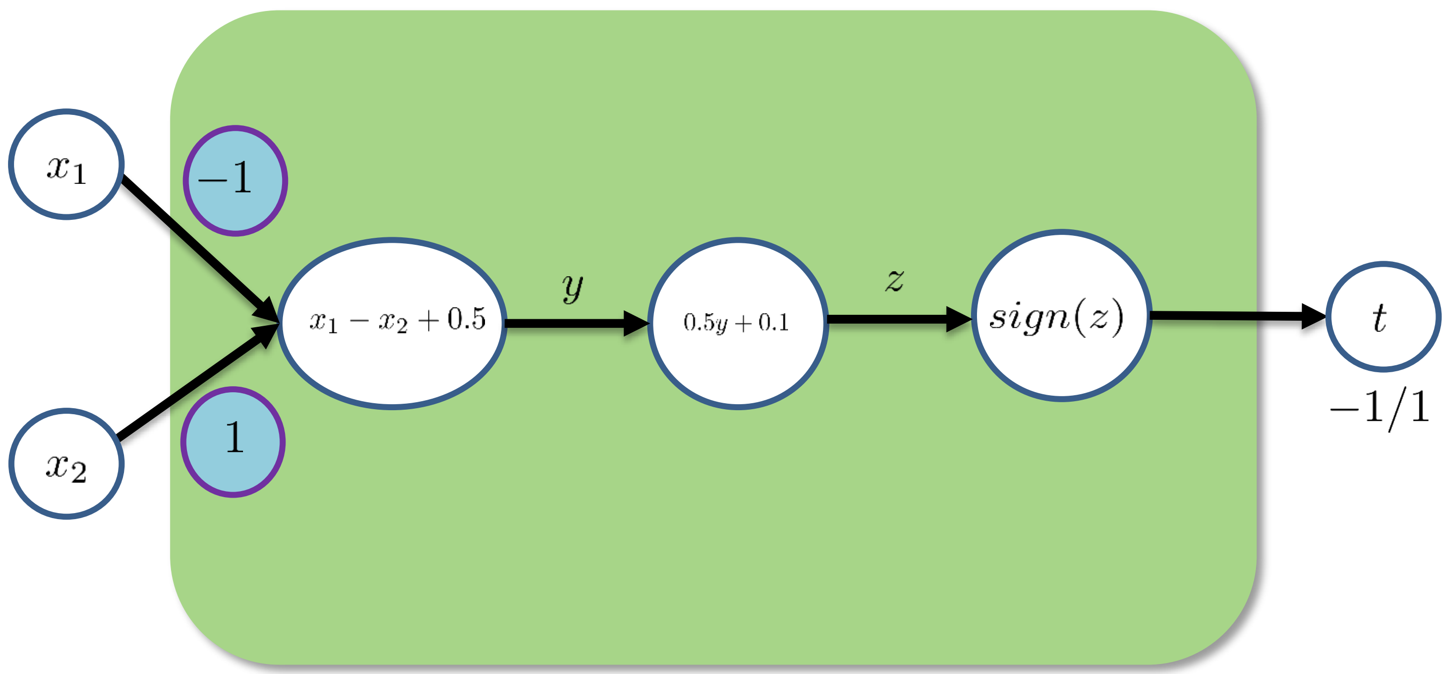
Translation: BNN to SAT



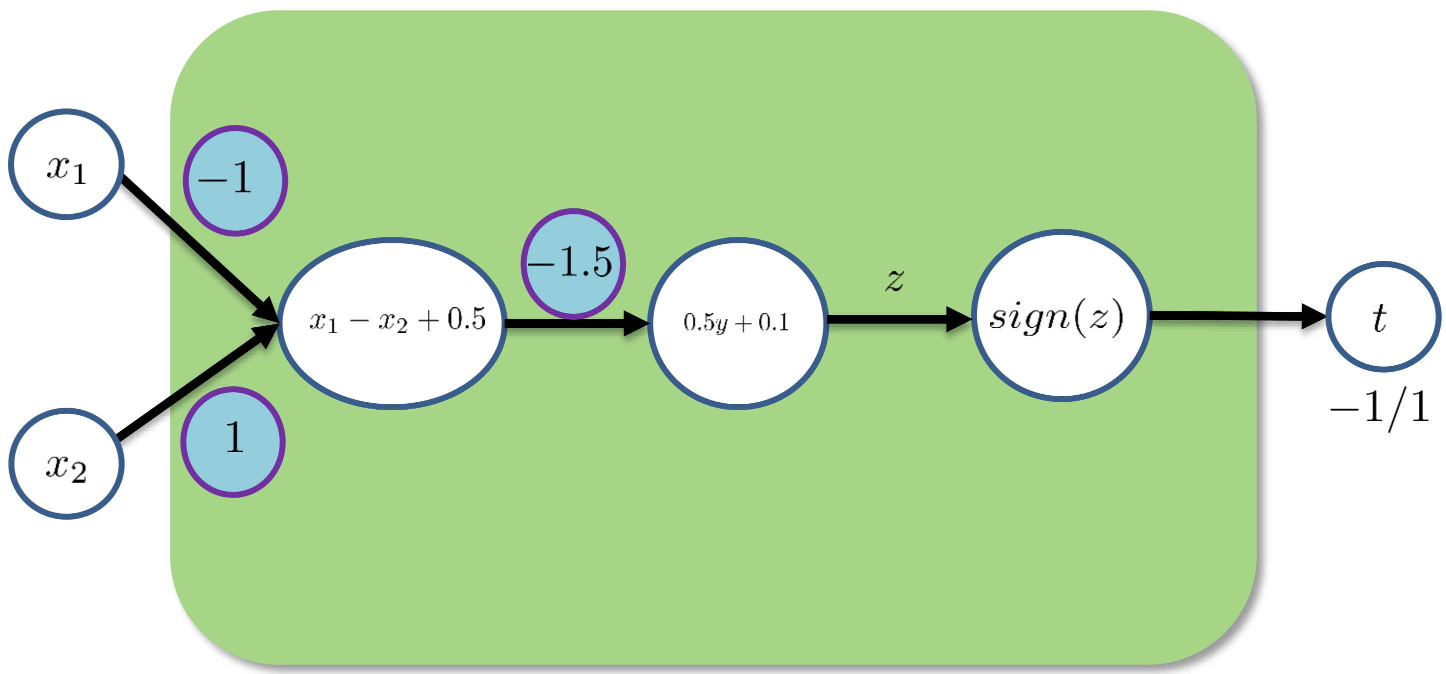
Translation: BNN to SAT



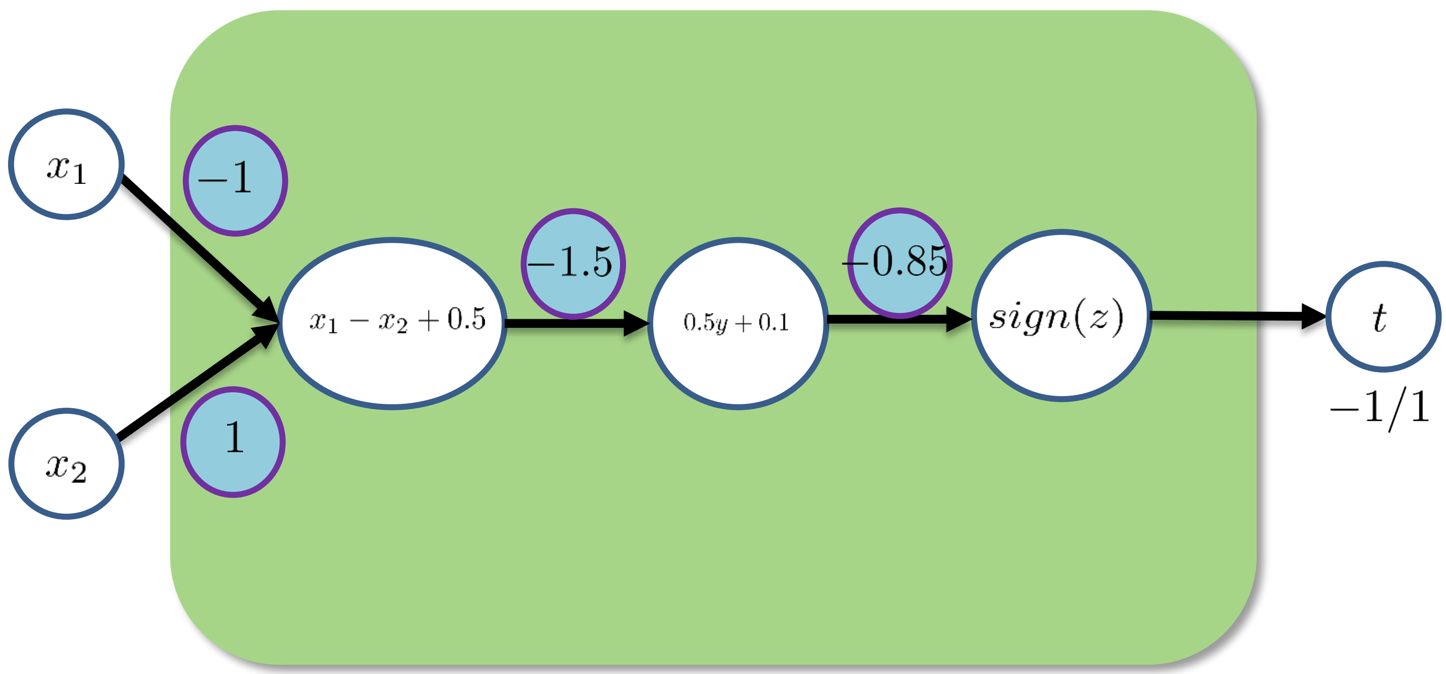
Translation: BNN to SAT



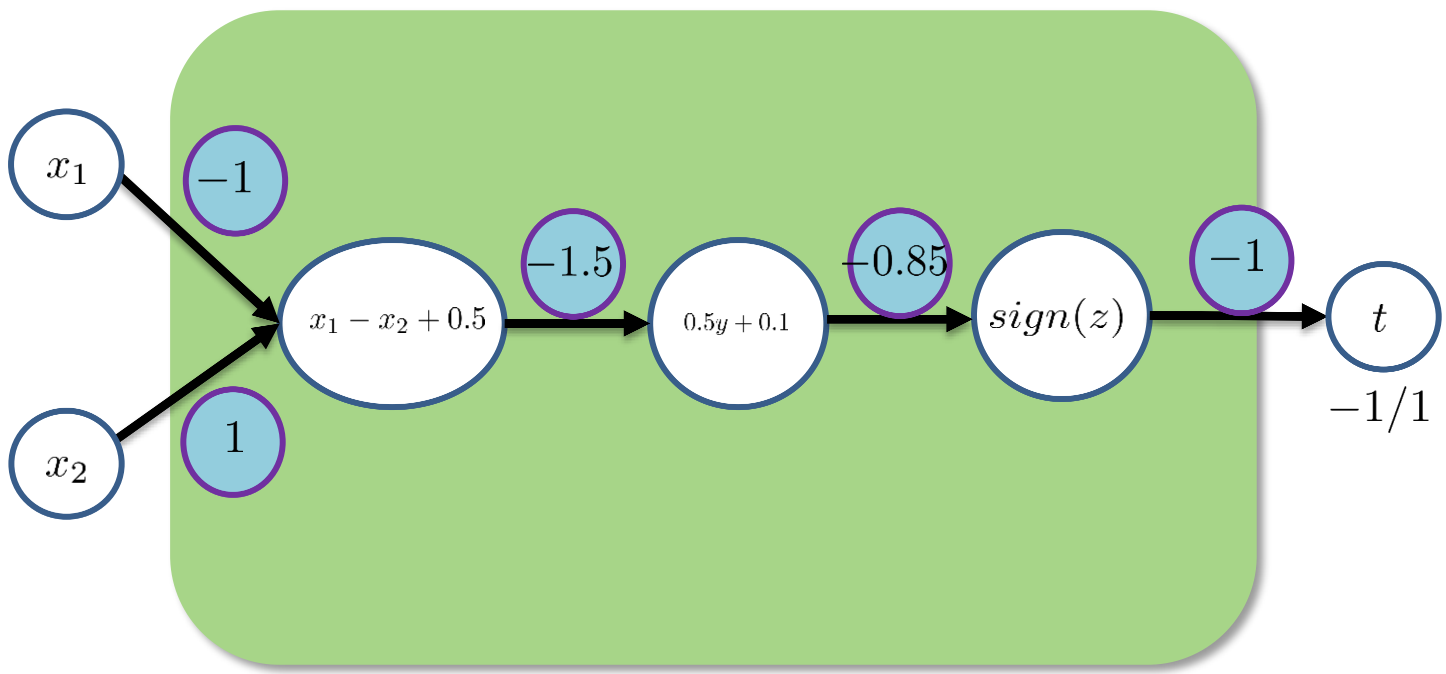
Translation: BNN to SAT



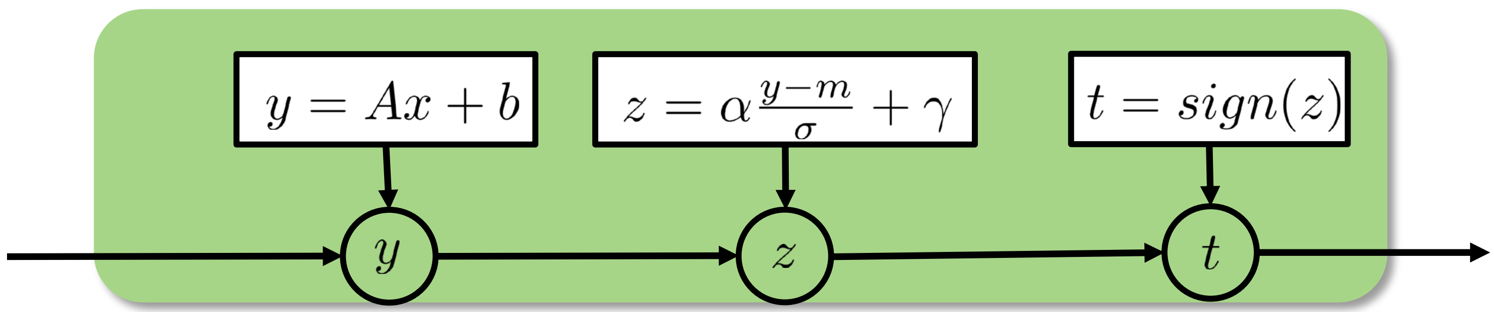
Translation: BNN to SAT



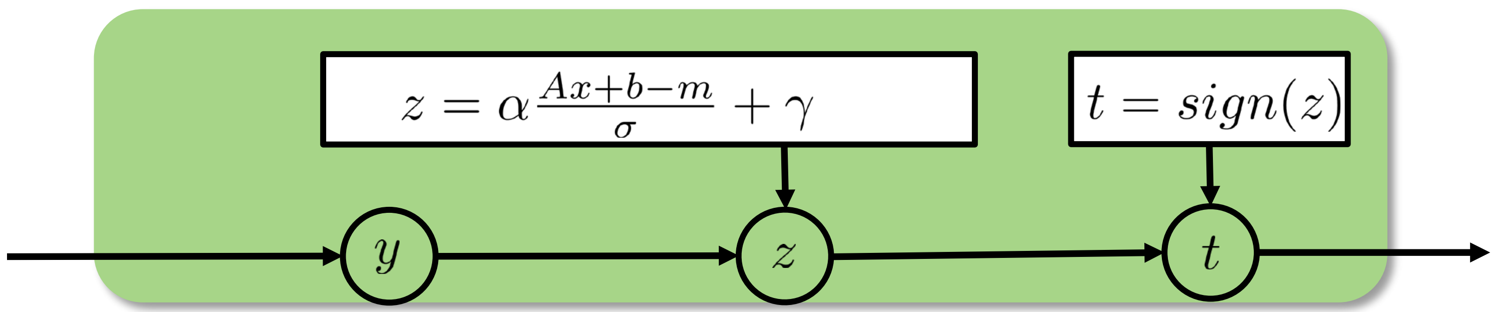
Translation: BNN to SAT



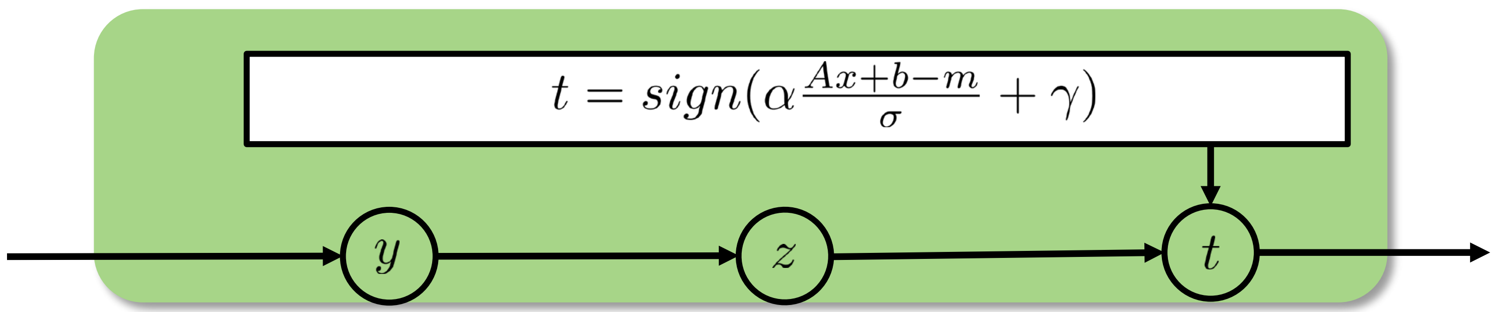
Translation: BNN to SAT



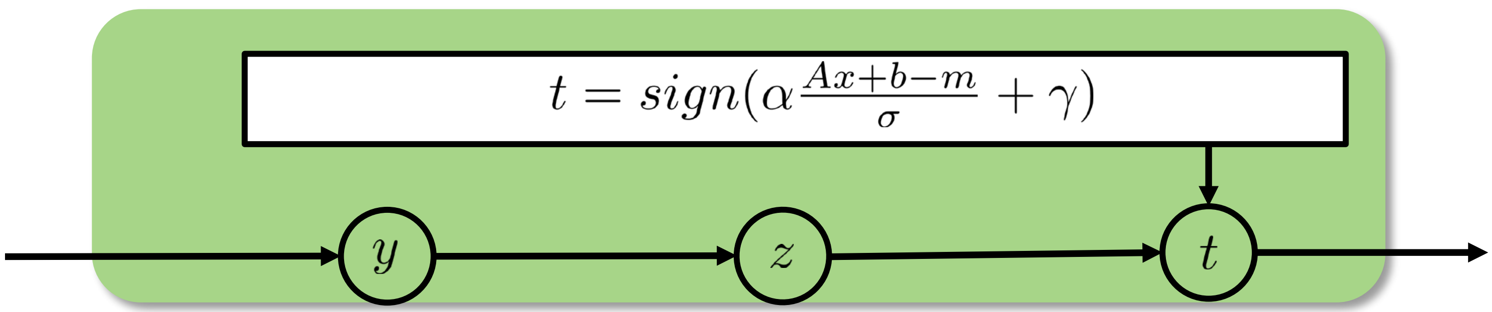
Translation: BNN to SAT



Translation: BNN to SAT



Translation: BNN to SAT



$$t_i = sign\left(\alpha \frac{(a_{i,1}x_1 + \ldots + a_{i,n}x_n + b) - m}{\sigma} - \gamma\right)$$

Translation: BNN to SAT

$$t_i = sign\left(\alpha \frac{(a_{i,1}x_1 + \ldots + a_{i,n}x_n + b) - m}{\sigma} - \gamma\right)$$

Translation: BNN to SAT

$$\left(\alpha\frac{(a_{i,1}x_1+\ldots+a_{i,n}x_n+b)-m}{\sigma}-\gamma\geq 0\right)\Leftrightarrow t_i=1$$

Translation: BNN to SAT

$$\left(\alpha\frac{(a_{i,1}x_1+\ldots+a_{i,n}x_n+b)-m}{\sigma}-\gamma\geq 0\right)\Leftrightarrow t_i=1$$

assuming $\alpha > 0$

Translation: BNN to SAT

$$\left(\alpha\frac{(a_{i,1}x_1+\ldots+a_{i,n}x_n+b)-m}{\sigma}-\gamma\geq 0\right)\Leftrightarrow t_i=1$$

$$\Updownarrow$$

assuming $\alpha > 0$

$$\left(l_1+\ldots+l_n\geq\left\lceil\frac{\sigma(-\gamma)}{\alpha}+m-b+c\right\rceil\right)\Leftrightarrow t_i=1$$

where

$a_{i,j}=1\Rightarrow l_j=x_j,$

$a_{i,j}=-1\Rightarrow l_j=\bar{x}_j$

Translation: BNN to SAT

$$\left(\alpha \frac{(a_{i,1}x_1 + \ldots + a_{i,n}x_n + b) - m}{\sigma} - \gamma \geq 0\right) \Leftrightarrow t_i = 1$$

assuming $\alpha > 0$

$$\left(l_1 + \ldots + l_n \geq \left\lceil \frac{\sigma(-\gamma)}{\alpha} + m - b + c \right\rceil\right) \Leftrightarrow t_i = 1$$

where

$a_{i,j} = 1 \Rightarrow l_j = x_j,$

$a_{i,j} = -1 \Rightarrow l_j = \bar{x}_j$

Translation: BNN to SAT

$$\left(\alpha \frac{(a_{i,1}x_1 + \ldots + a_{i,n}x_n + b) - m}{\sigma} - \gamma \geq 0\right) \Leftrightarrow t_i = 1$$

$$\Updownarrow$$

assuming $\alpha > 0$

$$\left(l_1 + \ldots + l_n \geq \left\lceil \frac{\sigma(-\gamma)}{\alpha} + m - b + c \right\rceil\right) \Leftrightarrow t_i = 1$$

where
$a_{i,j} = 1 \Rightarrow l_j = x_j,$
$a_{i,j} = -1 \Rightarrow l_j = \bar{x}_j$

Translation: BNN to SAT

$$\left(\alpha \frac{(a_{i,1}x_1+\ldots+a_{i,n}x_n+b)-m}{\sigma} - \gamma \geq 0\right) \Leftrightarrow t_i = 1$$

$$\Updownarrow$$

assuming $\alpha > 0$

$$\left(l_1 + \ldots + l_n \geq \lceil \frac{\sigma(-\gamma)}{\alpha} + m - b + c \rceil\right) \Leftrightarrow t_i = 1$$

where

$a_{i,j} = 1 \Rightarrow l_j = x_j,$

$a_{i,j} = -1 \Rightarrow l_j = \bar{x}_j$

Translation: BNN to SAT

$$\left(\alpha\frac{(a_{i,1}x_1+\ldots+a_{i,n}x_n+b)-m}{\sigma}-\gamma\geq 0\right)\Leftrightarrow t_i=1$$

assuming $\alpha > 0$

$$\Updownarrow$$

$$(l_1+\ldots+l_n\geq k)\Leftrightarrow t_i=1$$

where

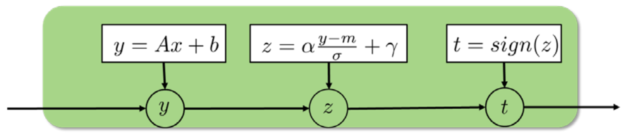
$a_{i,j}=1\Rightarrow l_j=x_j,$

$a_{i,j}=-1\Rightarrow l_j=\bar{x}_j$

Translation: BNN to SAT

$$(l_1 + \ldots + l_n \geq k) \Leftrightarrow t_i = 1$$

Translation: BNN to SAT

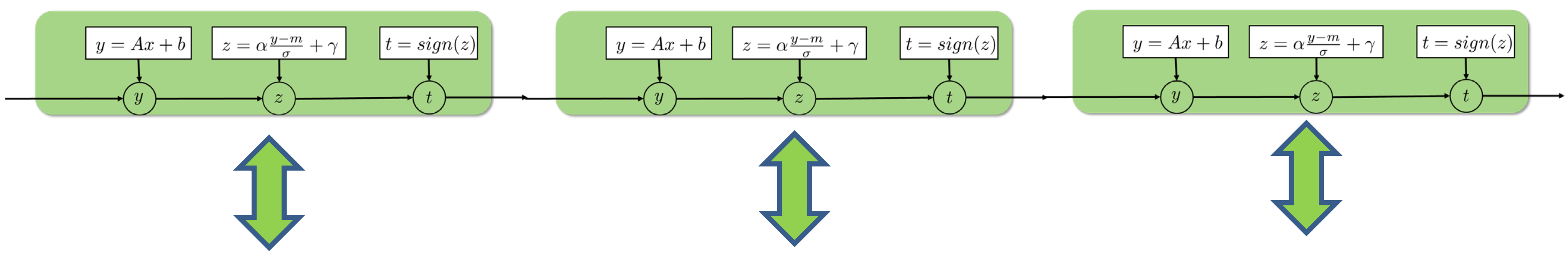


$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

$$\vdots$$

$$(l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m = 1$$

Translation: BNN to SAT



$$\begin{matrix}(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1^1 = 1 \\ \vdots \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m^1 = 1\end{matrix} \wedge \begin{matrix}(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1 \\ \vdots \quad . \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m = 1\end{matrix} \wedge \begin{matrix}(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1^p = 1 \\ \vdots \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m^p = 1\end{matrix}$$

BinBNN

Logic-based analysis of BNNs

Logic-based analysis of BNNs

Verification

Quantitative reasoning

Compilation

Logic-based analysis of BNNs

Properties verification using SAT solvers

Nina Narodytska, Shiva Prasad Kasiviswanathan, Leonid Ryzhyk, Mooly Sagiv, and Toby Walsh.
Verifying properties of binarized deep neural networks AAAI'18
Elias B. Khalil, Amrita Gupta, Bistra Dilkina:
Combinatorial Attacks on Binarized Neural Networks ICLR'19

Quantitative reasoning using approximate methods

Nina Narodytska, Aditya A. Shrotri, Kuldeep S. Meel, Alexey Ignatiev, João Marques-Silva:
Assessing Heuristic Machine Learning Explanations with Model Counting SAT'19.

Quantitative Verification of Neural Networks And its Security Applications
Teodora Baluta, Shiqi Shen, Shweta Shinde, Kuldeep S. Meel, Prateek Saxena

Reasoning via knowledge compilation

Verifying Binarized Neural Networks by Local Automaton Learning
Andy Shih and Adnan Darwiche and Arthur Choi

Work with small networks

Work with small networks

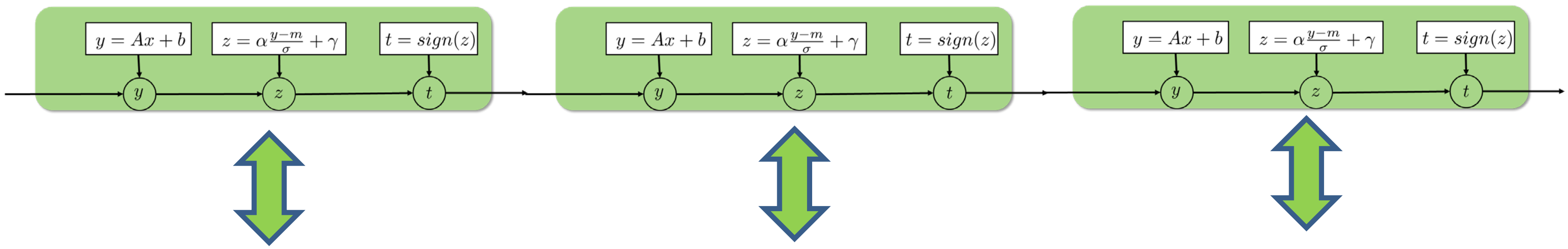
- Properties verification using SAT solvers
  - < 200K (robustness with a very small epsilon)
- Quantitative reasoning using approximate methods
  - < 51K
- Knowledge compilation, e.g. BDD, SDD
  - < 10K

Do we augment training?

no	yes
Sound and complete	

Easier-to-verify networks

Translation: BNN to SAT



$$\begin{array}{c}(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1^1 = 1 \\ \vdots \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m^1 = 1\end{array} \wedge \begin{array}{c}(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1 \\ \vdots \quad . \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m = 1\end{array} \wedge \begin{array}{c}(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1^p = 1 \\ \vdots \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m^p = 1\end{array}$$

BinBNN

“Neuron” constraint

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

$\ldots$

“Neuron” constraint

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

…

Number of variables

Reification means no propagation!

"Neuron" constraint

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

$\ldots$

Number of variables

Reification means no propagation!

+ reduce #vars

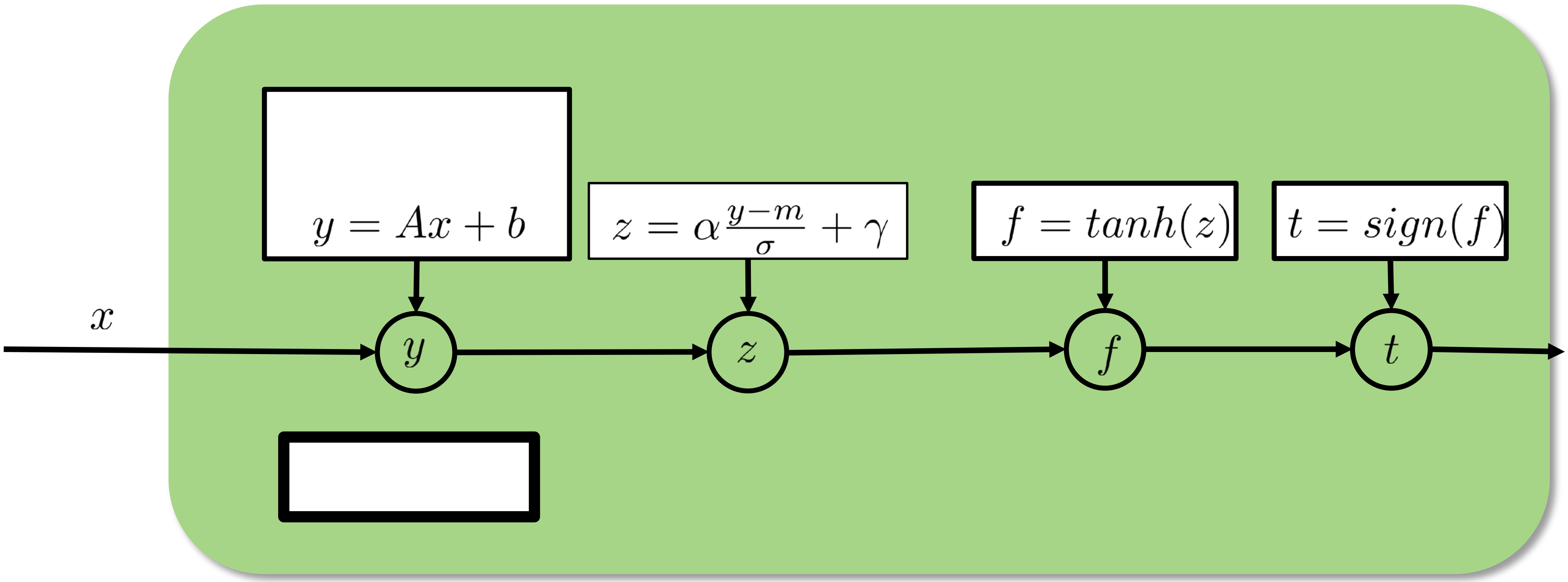
+ eliminate reifications

We can train a BNN so that

+ reduce #vars

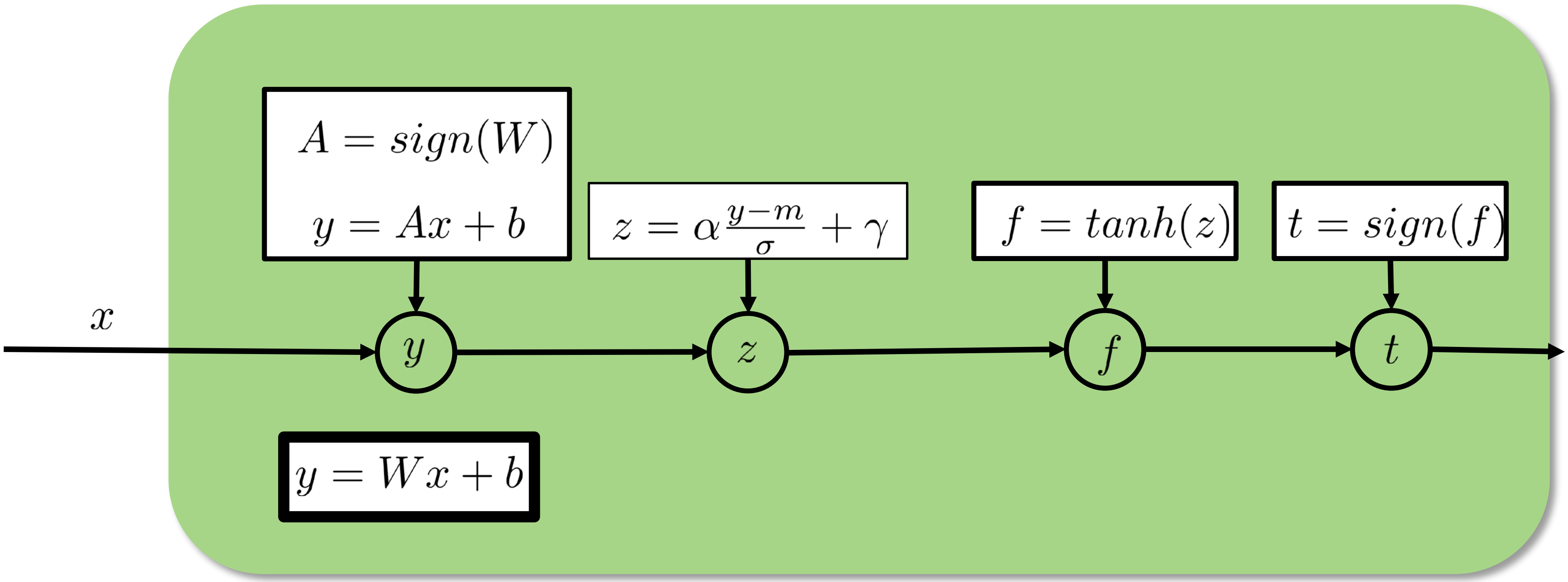
+ eliminate reifications

Binarized Neural Network



$x, a_{i,j} \in \{-1, 1\}$ $b, \alpha, m, \sigma, \gamma, W \in \mathbf{R}$

Binarized Neural Network



$x, a_{i,j} \in \{-1, 1\}$ $\qquad$ $b, \alpha, m, \sigma, \gamma, W \in \mathbf{R}$

Ternary quantization

BNN+: Improved Binary Network Training
Sajad Darabi, Mouloud Belbahri, Matthieu Courbariaux, Vahid Partovi Nia

Ternary quantization

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

where

$a_{i,j} = 1 \Rightarrow l_j = x_j,$

$a_{i,j} = -1 \Rightarrow l_j = \bar{x}_j$

Ternary quantization

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

where

$a_{i,j} = 1 \Rightarrow l_j = x_j,$

$a_{i,j} = 0 \Rightarrow l_j = 0,$

$a_{i,j} = -1 \Rightarrow l_j = \bar{x}_j$

L1+Ternary quantization

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

where

$$a_{i,j} = 1 \Rightarrow l_j = x_j,$$
$$a_{i,j} = 0 \Rightarrow l_j = 0,$$
$$a_{i,j} = -1 \Rightarrow l_j = \bar{x}_j$$

Add L1 regularization

L1+Ternary quantization

1. Train a BNN
2. Build a distribution of absolute values of weights
3. Select a percentile (40%, 60%), t= 0.03
4. Train a ternary BNN with the two-sided threshold t

$$a_{i,j} = \begin{cases} 0 & \text{if } |w_{i,j}| \leq t \\ sign(w_{i,j}) & \text{otherwise} \end{cases}$$

Stabilization of SIGN

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

$$LB_{(l_{1,1}+\ldots+l_{1,n}-k_1)} \geq 0$$

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

$$LB_{(l_{1,1}+\ldots+l_{1,n}-k_1)} \geq 0 \qquad t_1 = 1$$

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

$$UB_{(l_{1,1}+\ldots+l_{1,n}-k_1)} < 0$$

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

$$UB_{(l_{1,1}+\ldots+l_{1,n}-k_1)} < 0 \qquad\qquad t_1 = 0$$

Stabilization of SIGN

Encourage LB and UB of a neurons to take the same sign:

$$sign(UB_{i,j}) = sign(LB_{i,j})$$

Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability
Kai Y. Xiao, Vincent Tjeng, Nur Muhammad (Mahi) Shafiullah, Aleksander Madry

Stabilization of SIGN

Encourage LB and UB of a neurons to take the same sign:

$$-sign(UB_{i,j}) * sign(LB_{i,j})$$

Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability
Kai Y. Xiao, Vincent Tjeng, Nur Muhammad (Mahi) Shafiullah, Aleksander Madry

Stabilization of SIGN

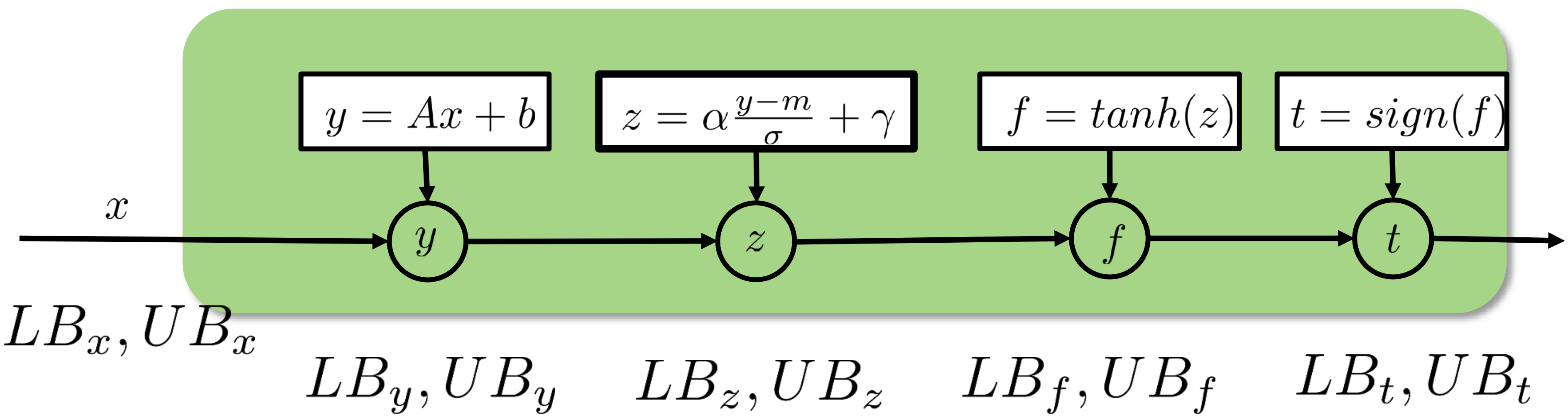
Encourage LB and UB of a neurons to take the same sign:

$$\cancel{-sign(UB_{i,j}) * sign(LB_{i,j}}\,)$$

$$-tanh(1 + UB_{ij}LB_{ij})$$

Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability
Kai Y. Xiao, Vincent Tjeng, Nur Muhammad (Mahi) Shafiullah, Aleksander Madry

Stabilization of SIGN



Sparse+L1+StableSign

BNNs	MNIST		FASHION		MNISTBG	
	%	#prms	%	#prms	%	#prms
Vanilla	96.5	$623K$	82.1	$623K$	74.3	$623K$
Sparse	96.4	$32K$	84.1	$37K$	78.2	$41K$
Sparse+Stable	95.9	$32K$	83.2	$37K$	78.3	$38K$
Sparse+L1	96.0	$20K$	83.7	$35K$	78.4	$36K$
Sparse+L1+Stable	95.2	$20K$	82.9	$37K$	80.0	$34K$

Sparse+L1+StableSign

BNNs	MNIST		FASHION		MNISTBG	
	%	#prms	%	#prms	%	#prms
Vanilla	96.5	$623K$	82.1	$623K$	74.3	$623K$
Sparse	96.4	$32K$	84.1	$37K$	78.2	$41K$
Sparse+Stable	95.9	$32K$	83.2	$37K$	78.3	$38K$
Sparse+L1	96.0	$20K$	83.7	$35K$	78.4	$36K$
Sparse+L1+Stable	95.2	$20K$	82.9	$37K$	80.0	$34K$

Sparse+L1+StableSign

BNNs	MNIST		FASHION		MNISTBG	
	%	#prms	%	#prms	%	#prms
Vanilla	96.5	$623K$	82.1	$623K$	74.3	$623K$
Sparse	96.4	$32K$	84.1	$37K$	78.2	$41K$
Sparse+Stable	95.9	$32K$	83.2	$37K$	78.3	$38K$
Sparse+L1	96.0	$20K$	83.7	$35K$	78.4	$36K$
Sparse+L1+Stable	95.2	$20K$	82.9	$37K$	80.0	$34K$

Sparse+L1+StableSign

BNNs	MNIST		FASHION		MNISTBG	
	%	#prms	%	#prms	%	#prms
Vanilla	96.5	$623K$	82.1	$623K$	74.3	$623K$
Sparse	96.4	$32K$	84.1	$37K$	78.2	$41K$
Sparse+Stable	95.9	$32K$	83.2	$37K$	78.3	$38K$
Sparse+L1	96.0	$20K$	83.7	$35K$	78.4	$36K$
Sparse+L1+Stable	95.2	$20K$	82.9	$37K$	80.0	$34K$

Sparse+L1+StableSign

BNNs	MNIST		FASHION		MNISTBG	
	%	#prms	%	#prms	%	#prms
Vanilla	96.5	623K	82.1	623K	74.3	623K
Sparse	96.4	32K	84.1	37K	78.2	41K
Sparse+Stable	95.9	32K	83.2	37K	78.3	38K
Sparse+L1	96.0	20K	83.7	35K	78.4	36K
Sparse+L1+Stable	95.2	20K	82.9	37K	80.0	34K

Sparse+L1+StableSign

BNNs	MNIST #vars/#cls	FASHION #vars/#cls	MNISTBG #vars/#cls
Sparse	63K/224K	34K/116K	24K/80K
Sparse+Stable	42K/146K	19K/58K	12K/36K
Sparse+L1	8K/20K	34K/115K	17K/53K
Sparse+Stable+L1	11K/33K	12K/33K	10K/28K

Outline

Motivation

Adversarial attacks

Verification methods

SAT-based verification of Binarized NNs

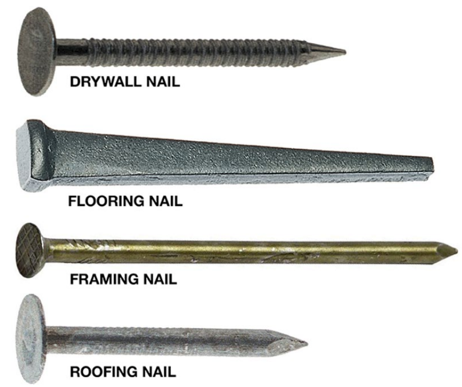
Where we are

Verification methods

Nails

Where we are

Verification methods



Nails

Where we are

VNN-LIB
Verification of Neural Networks

HOME ABOUT NEWS STANDARD BENCHMARKS SOFTWARE CREDITS

Home

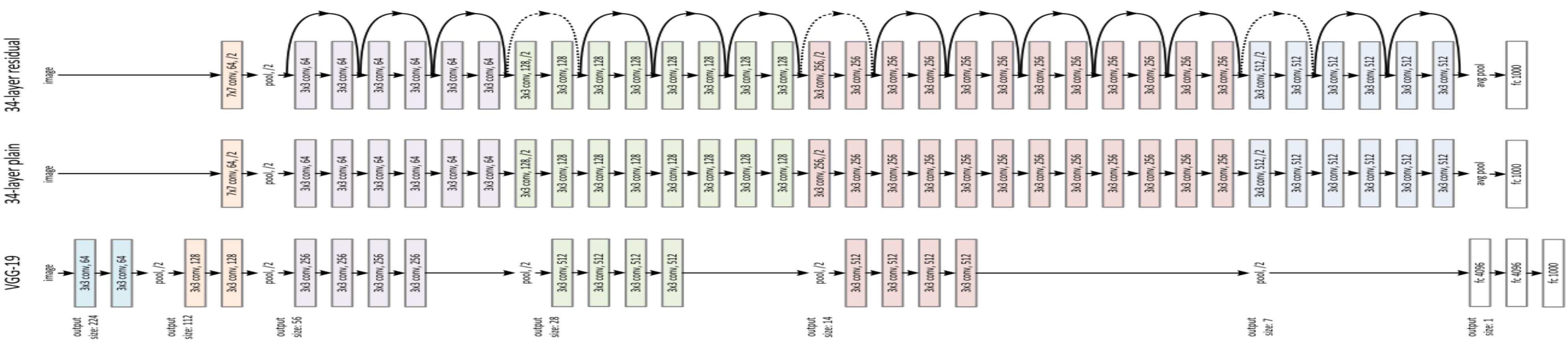
VNN-LIB is an international initiative whose aim is to encourage collaboration and facilitate research and development in Verification of Neural Networks (VNN).

The goals of VNN-LIB are:

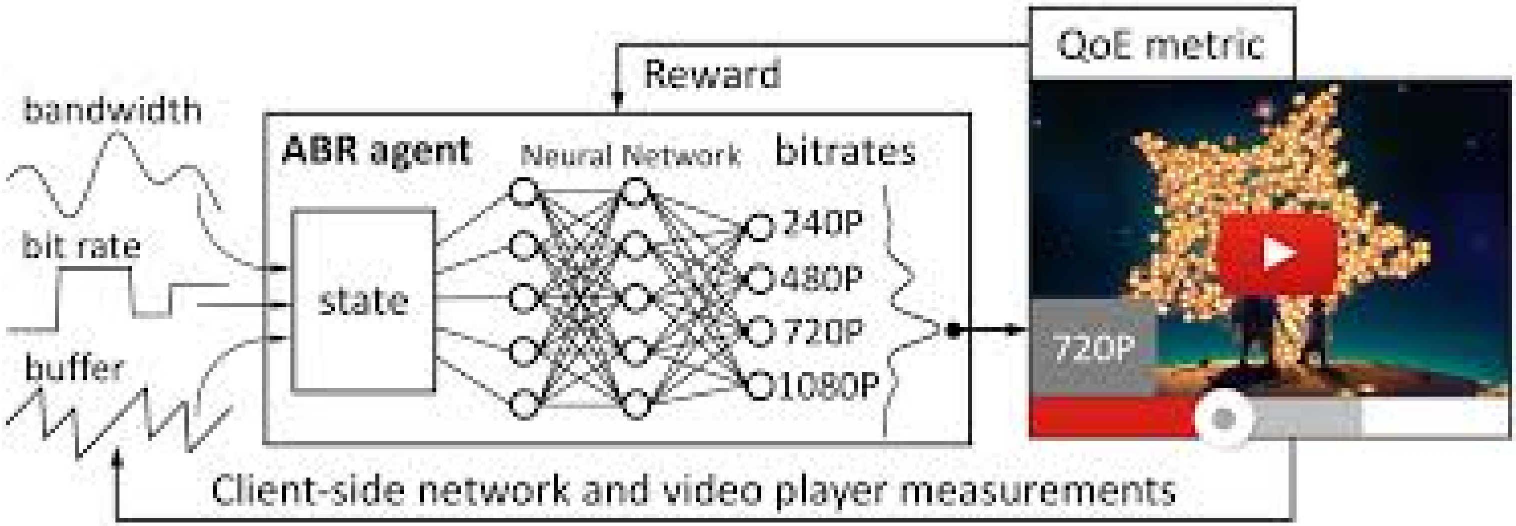
- Develop a cohesive community around VNN by connecting developers and researchers working in this domain.
- Establish a common format for the exchange of Neural Networks and their properties.
- Provide the community with a library of established common benchmarks for VNN tools.
- Provide and maintain a common repository for tools and resources useful to the VNN community.

The initiative and this site are still in their embryonal stages: your collaboration is essential to grow and improve VNN-LIB, so do not hesitate to send us feedback, comments and suggestions.

What is next?


34-layer residual
34-layer plain
VGG-19
output size: 224
output size: 112
output size: 56
output size: 28
output size: 14
output size: 7
output size: 1

What is next?


QoE metric
Reward
bandwidth
ABR agent
Neural Network
bitrates
bit rate
state
240P
480P
720P
1080P
buffer
720P
Client-side network and video player measurements

What is next?

1. Verification is a very important tool to analyze NNs

2. Smaller networks are useful in many practical applications

Thanks!

Rigorous Verification and Explanation of ML Models

Part 4

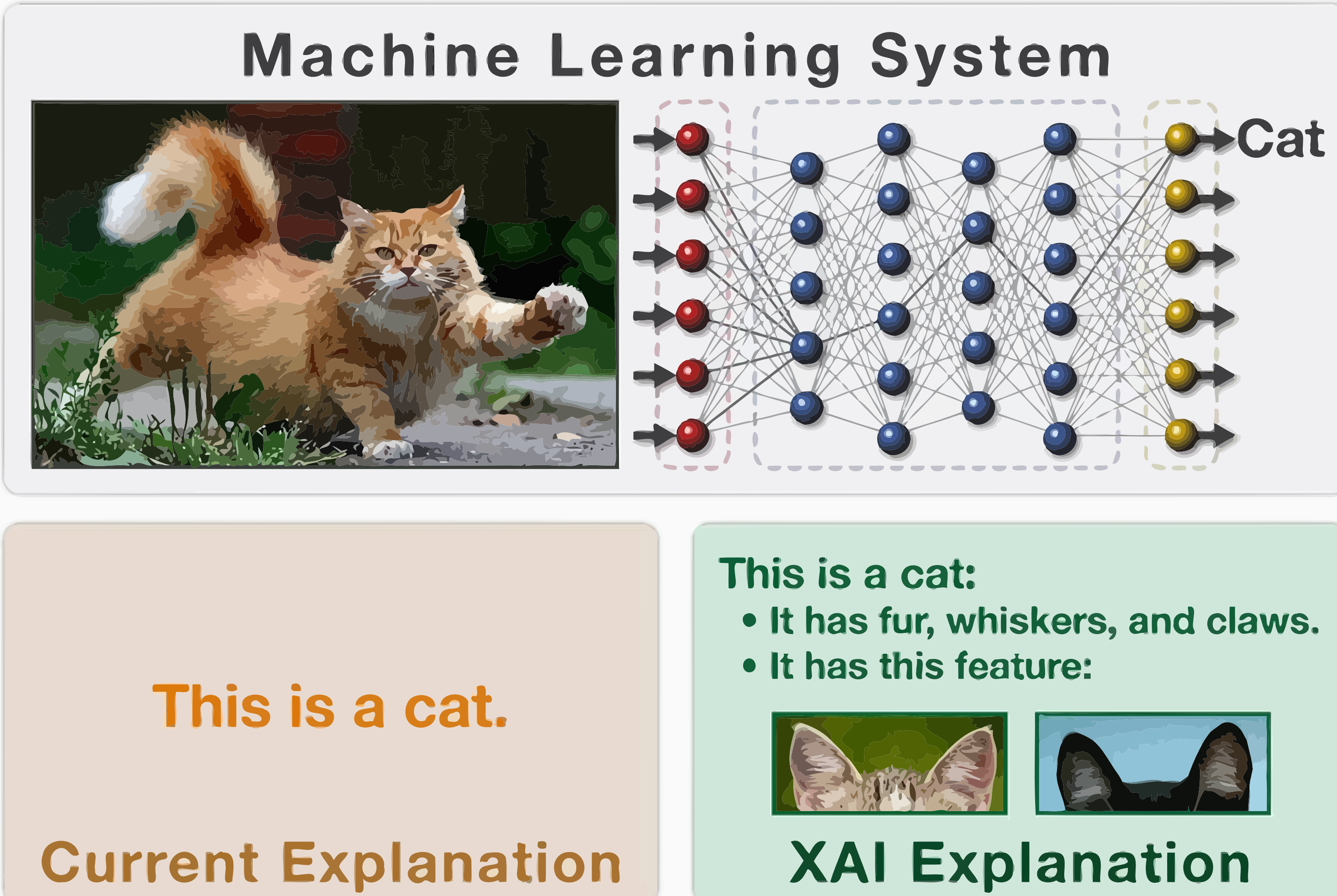
A. Ignatiev, J. Marques-Silva, K. Meel & N. Narodytska

Monash Univ, ANITI@Univ. Toulouse, NU Singapore & VMWare Research

February 08, 2020 | AAAI Tutorial SP1

Computing Explanations

What do we want to achieve?



A recap: approaches to XAI

interpretable ML models

(decision trees, lists, sets)

A recap: approaches to XAI

interpretable ML models

(decision trees, lists, sets)

explanation of ML models “on the fly”

(post-hoc explanation)

Why? or Why not? explanations

why? **why not?**

*(**why** did (**not**) I get a loan?)*

Why? or Why not? explanations

why? **why not?**

*(**why** did (**not**) I get a loan?)*

abductive ***contrastive***

Heuristic approaches exist

State of the art (heuristics)

heuristic approaches exist

(e.g. **LIME**, **Anchor**, or **SHAP**) [RSG16, RSG18, LL17]

State of the art (heuristics)

heuristic approaches exist

(e.g. **LIME**, **Anchor**, or **SHAP**) [RSG16, RSG18, LL17]

- **local** explanations

State of the art (heuristics)

heuristic approaches exist

(e.g. **LIME**, **Anchor**, or **SHAP**) [RSG16, RSG18, LL17]

- **local** explanations
- **no** guarantees

State of the art (heuristics)

heuristic approaches exist

(e.g. **LIME**, **Anchor**, or **SHAP**) [RSG16, RSG18, LL17]

- **local** explanations
- **no** guarantees

⇒ **(un-)reliable?**

Rigorous approaches

State of the art (rigorous approaches)

alternative is to use logic

State of the art (rigorous approaches)

alternative is to use logic

(reasoning over formal models)

State of the art (rigorous approaches)

alternative is to use logic

(reasoning over formal models)

⇓

- **search**

State of the art (rigorous approaches)

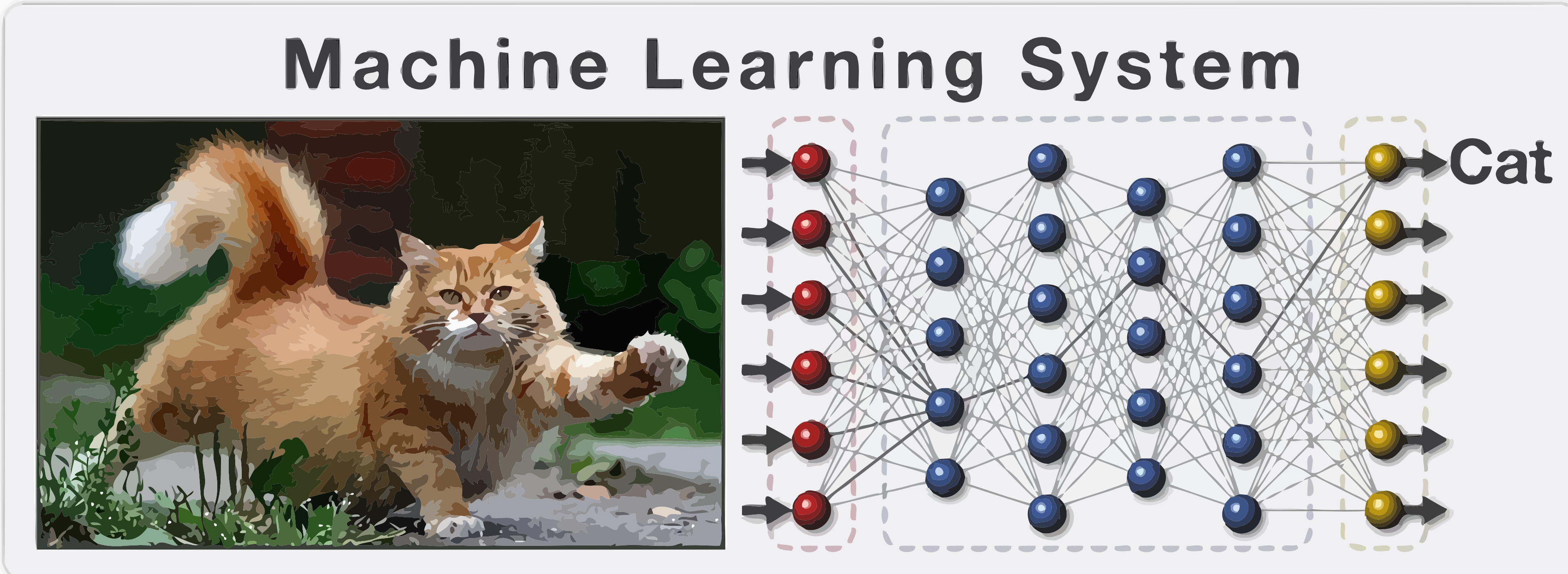
alternative is to use logic

(reasoning over formal models)

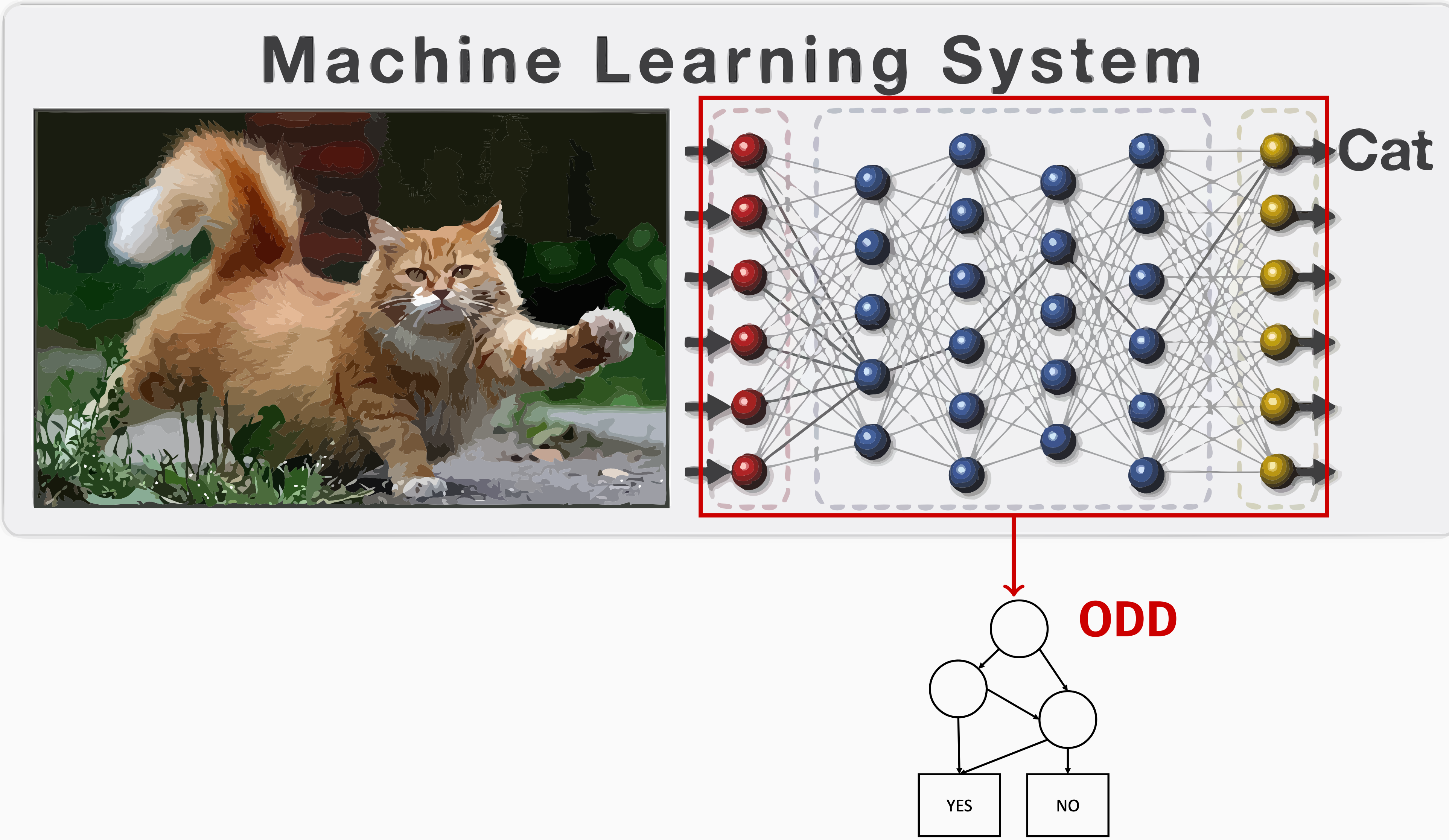
- **search**
- **compilation**

Compilation-based approach

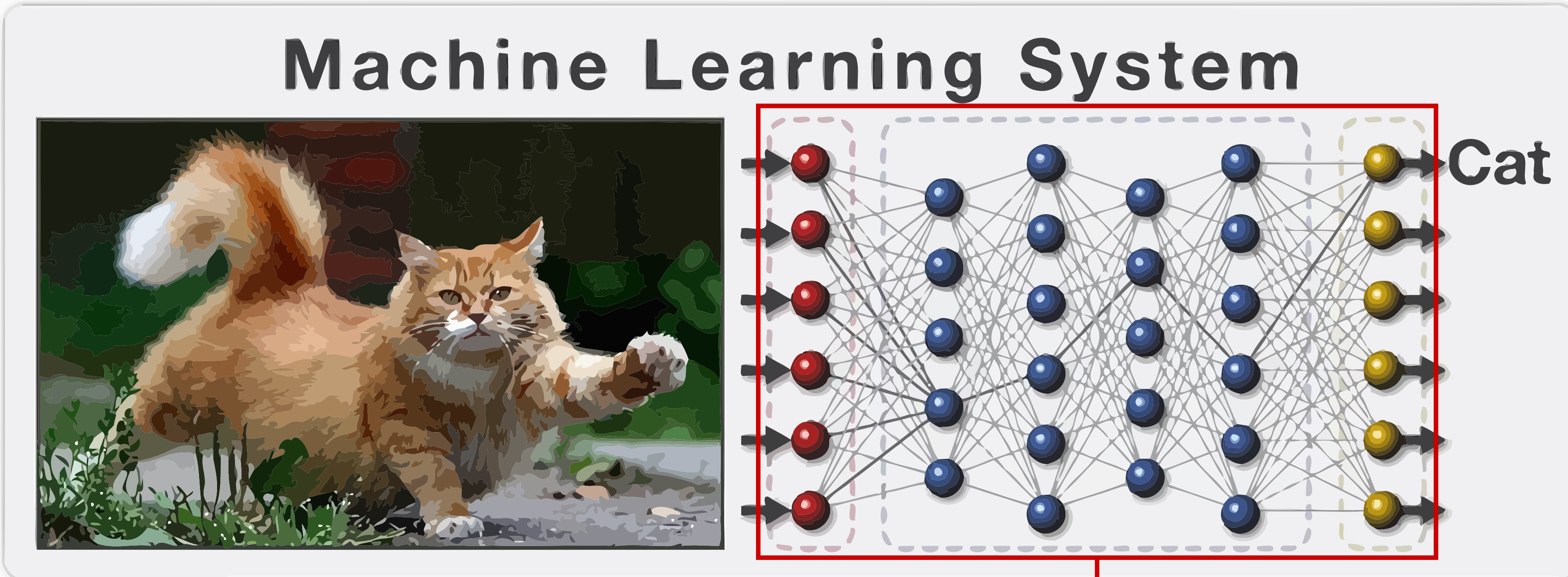
Compiling a classifier

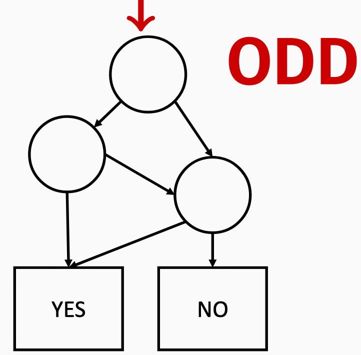


Compiling a classifier



Compiling a classifier





perform operations on
tractable representation

The idea is that

once you have an ODD:

The idea is that

once you have an ODD:

- compute **MC explanations** [SCD18]

  "Which positive features are responsible for a yes decision?"

  "Which negative features are responsible for a no decision?"

once you have an ODD:

- compute **MC explanations** [SCD18]

  "Which positive features are responsible for a yes decision?"

  "Which negative features are responsible for a no decision?"

- compute **PI explanations** [SCD18]

  "Which features (+ or -) make the other features irrelevant?"

once you have an ODD:

- compute **MC explanations** [SCD18]
  "Which positive features are responsible for a yes decision?"
  "Which negative features are responsible for a no decision?"

- compute **PI explanations** [SCD18]
  "Which features (+ or -) make the other features irrelevant?"

- perform **verification queries** [SDC19]
  counting of counterexamples, computing their probabilites and common characteristics

What ML models can we compile?

- **Naïve Bayes** [CD03]

What ML models can we compile?

- **Naïve Bayes** [CD03]
- **Latent Tree** [SCD18]

What ML models can we compile?

- **Naïve Bayes** [CD03]
- **Latent Tree** [SCD18]
- **General BN** [SCD19]

What ML models can we compile?

- **Naïve Bayes** [CD03]
- **Latent Tree** [SCD18]
- **General BN** [SCD19]
- **BNN and CNN** [SDC19]

Pros and cons of ML model compilation

reasoning about explanations in polynomial time

Pros and cons of ML model compilation

reasoning about explanations in polynomial time

but

Pros and cons of ML model compilation

reasoning about explanations in polynomial time

but

difficult to compute an ODD

Pros and cons of ML model compilation

reasoning about explanations in polynomial time

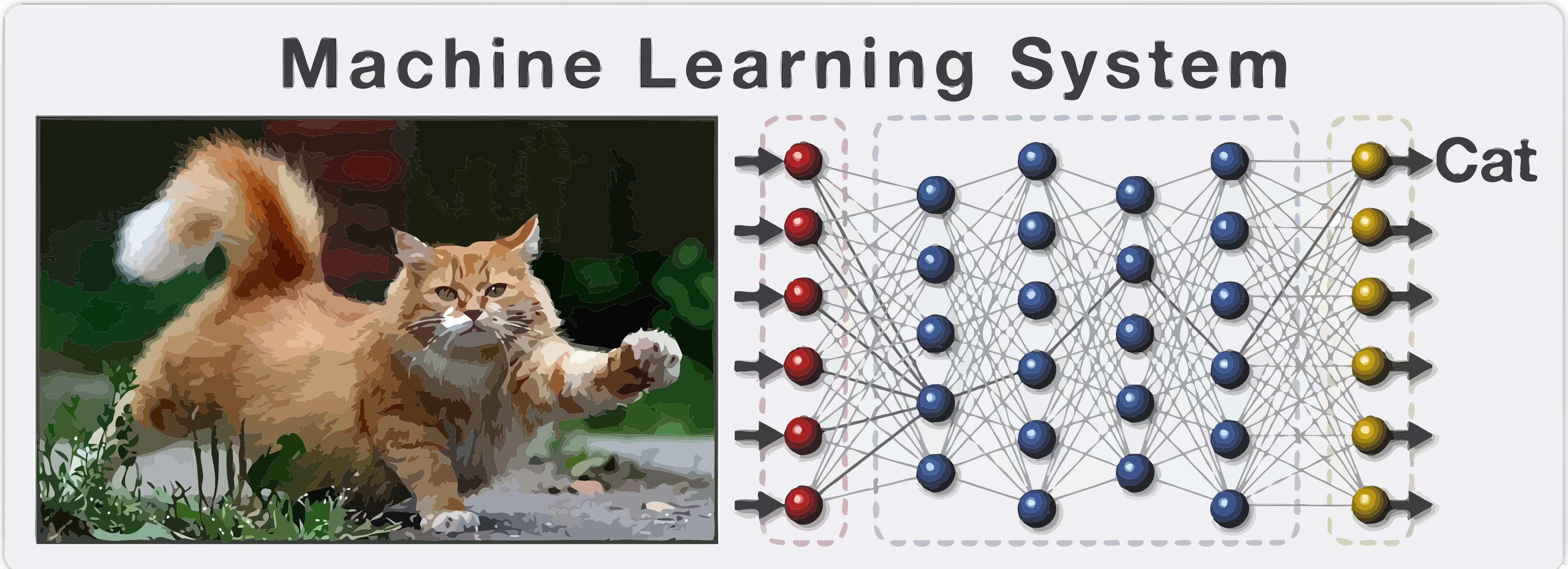
but

difficult to compute an ODD

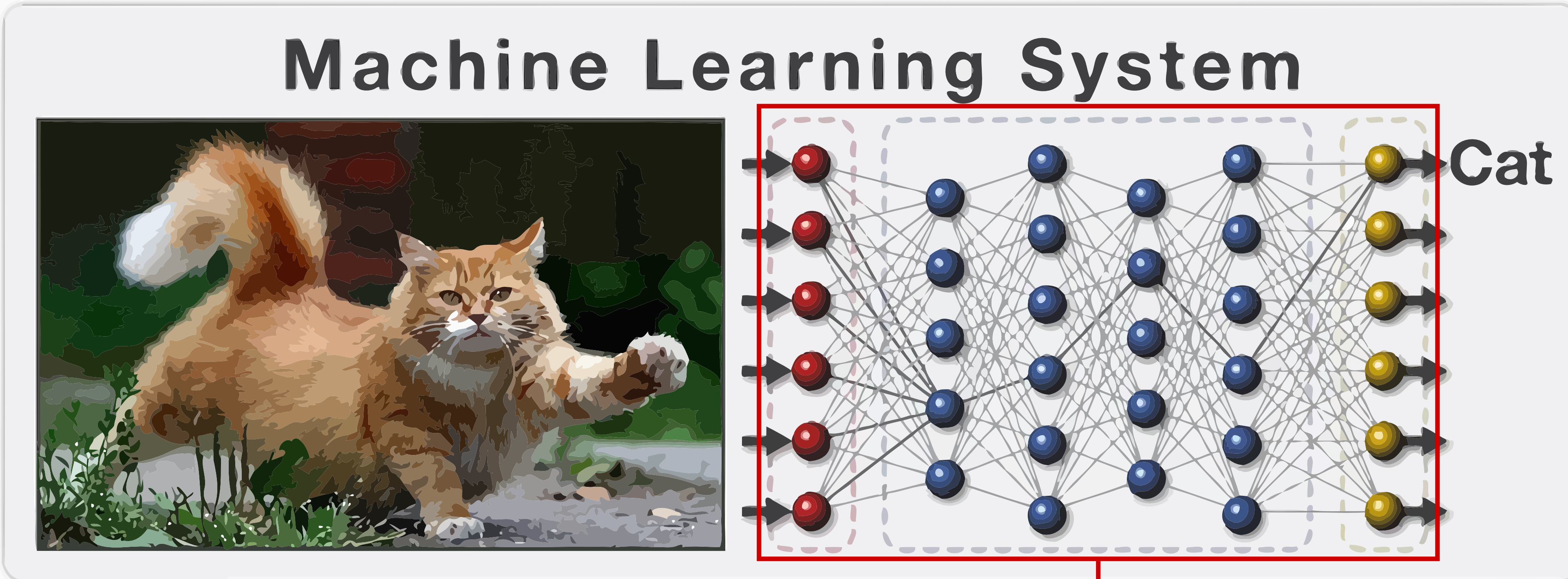
ODD can be *large*

Search-based explanations

From ML model to logic

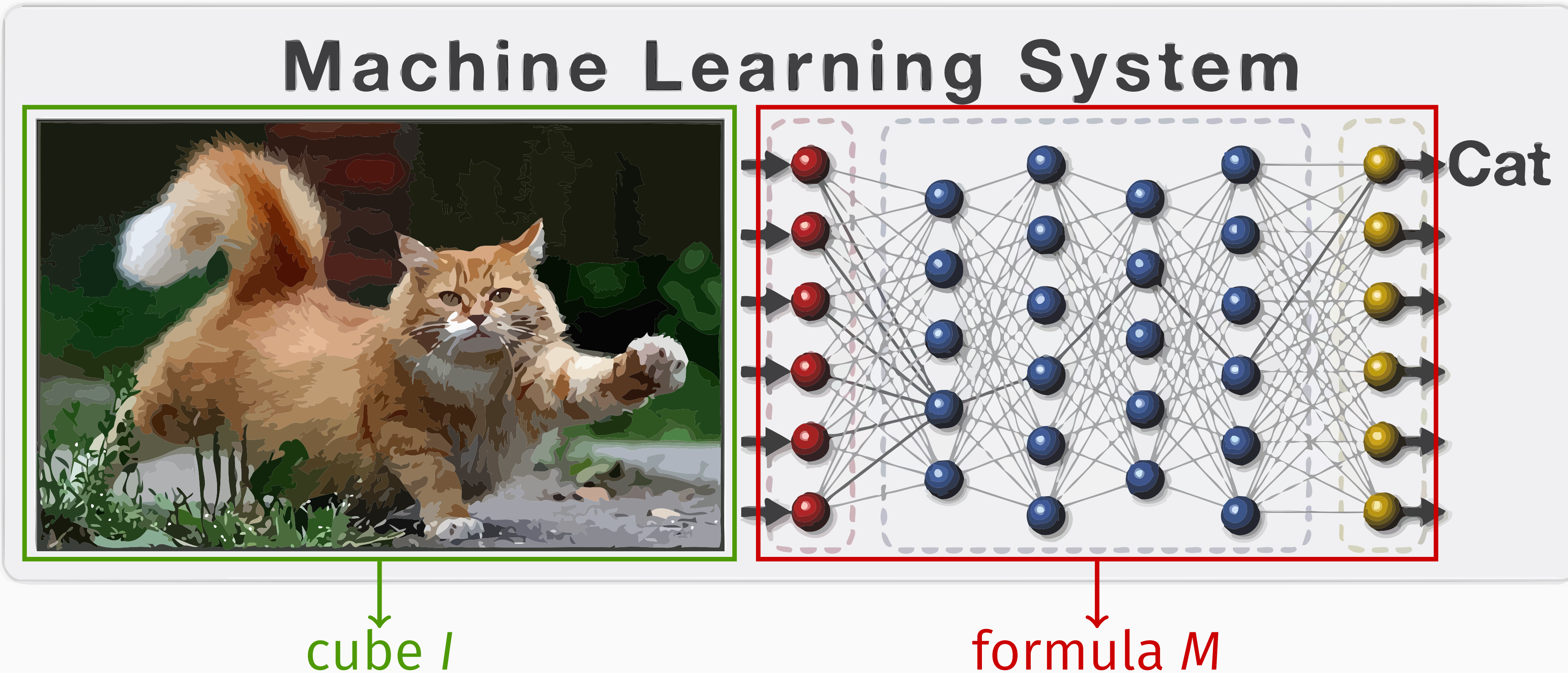


From ML model to logic

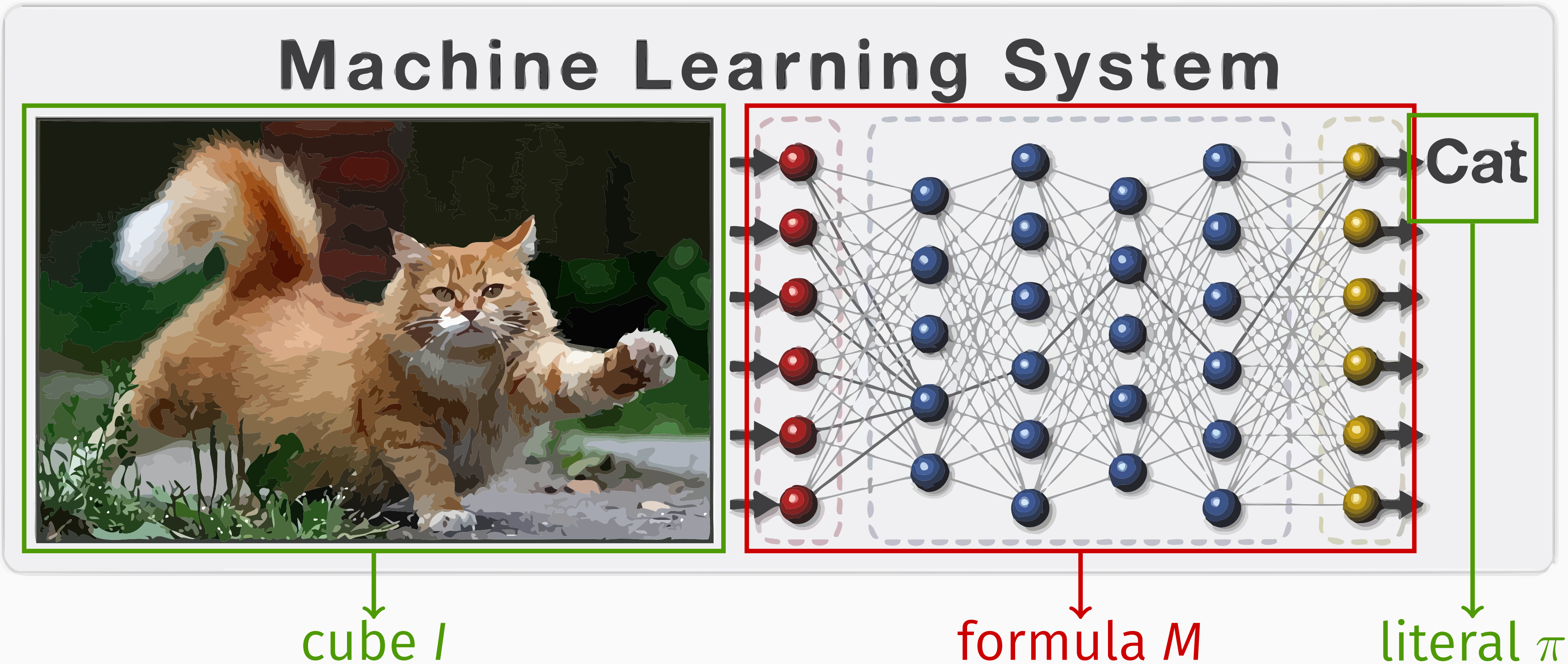


formula M

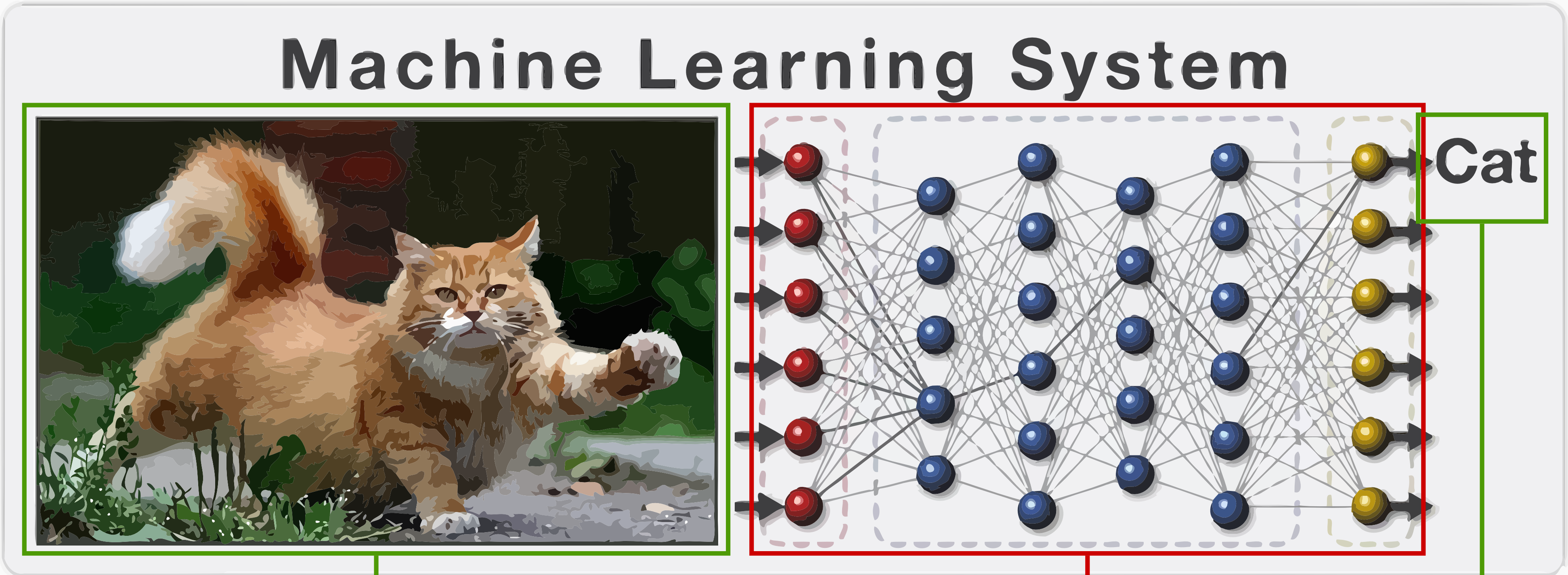
From ML model to logic



From ML model to logic



From ML model to logic



cube I formula M literal π

$$I \wedge M \vDash \pi$$

Abductive explanations of ML models

[INMS19]

given a ***classifier*** M, a ***cube*** I and a ***prediction*** π,

Abductive explanations of ML models

[INMS19]

given a ***classifier*** M, a ***cube*** I and a ***prediction*** π,

compute a (**cardinality- or subset-**) minimal $E_m \subseteq I$ s.t.

Abductive explanations of ML models

[INMS19]

given a ***classifier*** M, a ***cube*** I and a ***prediction*** π,

compute a (**cardinality- or subset-**) minimal $E_m \subseteq I$ s.t.

$$E_m \wedge M \not\models \perp$$

and

$$E_m \wedge M \models \pi$$

Abductive explanations of ML models

[INMS19]

given a ***classifier*** M, a ***cube*** I and a ***prediction*** π,

compute a (**cardinality- or subset-**) minimal $E_m \subseteq I$ s.t.

$$E_m \land M \not\models \bot$$

and

$$E_m \land M \models \pi$$

⇩

***iterative* explanation procedure**

Computing primes

1. $E_m \wedge M \not\models \bot$

Computing primes

1. $E_m \wedge M \not\models \bot$ — ***tautology***

Computing primes

1. $E_m \wedge M \not\models \bot$ — ***tautology***
2. $E_m \wedge M \models \pi$

Computing primes

1. $E_m \wedge M \not\models \bot$ — ***tautology***
2. $E_m \wedge M \models \pi \;\Leftrightarrow\; E_m \models (M \rightarrow \pi)$

Computing primes

1. $E_m \wedge M \not\models \bot$ — ***tautology***
2. $E_m \wedge M \models \pi \iff E_m \models (M \rightarrow \pi)$

$$\Downarrow$$

E_m is a ***prime implicant*** of $M \rightarrow \pi$

Computing one subset-minimal explanation

Input: model M, initial cube I, prediction π
Output: ***Subset-minimal*** explanation E_m

```
begin
   for l ∈ I :
      if Entails(I \ {l}, M → π) :        # make an (entailment) oracle call
         I ← I \ {l}
   return I
end
```

Computing one cardinality-minimal explanation

cardinality-minimal explanations can be computed

Computing one cardinality-minimal explanation

cardinality-minimal explanations can be computed

(following **implicit-hitting set** based approach) [IMM16]

Computing one cardinality-minimal explanation

cardinality-minimal explanations can be computed

(following **implicit-hitting set** based approach) [IMM16]

$\Downarrow$

but it is **hard for** Σ_2^P [INMS19]

Computing one cardinality-minimal explanation

cardinality-minimal explanations can be computed

(following **implicit-hitting set** based approach) [IMM16]

but it is **hard for Σ_2^P** [INMS19]

(**worst-case exponential** number of oracle queries)

Experimental setup

- implementation in Python
    - supports **SMT** solvers through PySMT
        - **Yices2** used
    - supports **CPLEX 12.8.0**

Experimental setup

- implementation in Python
  - supports **SMT** solvers through PySMT
    - **Yices2** used
  - supports **CPLEX 12.8.0**
- **ReLU-based** neural networks [FJ18]
  - one *hidden* layer with $i \in \{10, 15, 20\}$ neurons

Experimental setup

- implementation in Python
  - supports **SMT** solvers through PySMT
    - **Yices2** used
  - supports **CPLEX 12.8.0**
- **ReLU-based** neural networks [FJ18]
  - one *hidden* layer with $i \in \{10, 15, 20\}$ neurons
- benchmarks selected from:
  - **UCI** Machine Learning Repository
  - **Penn** Machine Learning Benchmarks
  - **MNIST** Digits Database

Experimental setup

- implementation in Python
  - supports **SMT** solvers through PySMT
    - **Yices2** used
  - supports **CPLEX 12.8.0**
- **ReLU-based** neural networks [FJ18]
  - one *hidden* layer with $i \in \{10, 15, 20\}$ neurons
- benchmarks selected from:
  - **UCI** Machine Learning Repository
  - **Penn** Machine Learning Benchmarks
  - **MNIST** Digits Database
- Machine configuration:
  - Intel Core i7 2.8GHz, 8GByte
  - time limit — 1800s
  - memory limit — 4GByte

Some of the experimental results

Dataset			Minimal explanation			Minimum explanation		
			size	SMT (s)	MILP (s)	size	SMT (s)	MILP (s)
australian	(14)	**m**	1	0.03	0.05	—	—	—
		a	8.79	1.38	0.33	—	—	—
		M	14	17.00	1.43	—	—	—
backache	(32)	**m**	13	0.13	0.14	—	—	—
		a	19.28	5.08	0.85	—	—	—
		M	26	22.21	2.75	—	—	—
breast-cancer	(9)	**m**	3	0.02	0.04	3	0.02	0.03
		a	5.15	0.65	0.20	4.86	2.18	0.41
		M	9	6.11	0.41	9	24.80	1.81
cleve	(13)	**m**	4	0.05	0.07	4	—	0.07
		a	8.62	3.32	0.32	7.89	—	5.14
		M	13	60.74	0.60	13	—	39.06
hepatitis	(19)	**m**	6	0.02	0.04	4	0.01	0.04
		a	11.42	0.07	0.06	9.39	4.07	2.89
		M	19	0.26	0.20	19	27.05	22.23
voting	(16)	**m**	3	0.01	0.02	3	0.01	0.02
		a	4.56	0.04	0.13	3.46	0.3	0.25
		M	11	0.10	0.37	11	1.25	1.77
spect	(22)	**m**	3	0.02	0.02	3	0.02	0.04
		a	7.31	0.13	0.07	6.44	1.61	0.67
		M	20	0.88	0.29	20	8.97	10.73

Some of the experimental results

Dataset			Minimal explanation			Minimum explanation		
			size	SMT (s)	MILP (s)	size	SMT (s)	MILP (s)
australian	(14)	**m**	1	0.03	0.05	—	—	—
		a	8.79	1.38	0.33	—	—	—
		M	14	17.00	1.43	—	—	—
backache	(32)	**m**	13	0.13	0.14	—	—	—
		a	19.28	5.08	0.85	—	—	—
		M	26	22.21	2.75	—	—	—
breast-cancer	(9)	**m**	3	0.02	0.04	3	0.02	0.03
		a	5.15	0.65	0.20	4.86	2.18	0.41
		M	9	6.11	0.41	9	24.80	1.81
cleve	(13)	**m**	4	0.05	0.07	4	—	0.07
		a	8.62	3.32	0.32	7.89	—	5.14
		M	13	60.74	0.60	13	—	39.06
hepatitis	(19)	**m**	6	0.02	0.04	4	0.01	0.04
		a	11.42	0.07	0.06	9.39	4.07	2.89
		M	19	0.26	0.20	19	27.05	22.23
voting	(16)	**m**	3	0.01	0.02	3	0.01	0.02
		a	4.56	0.04	0.13	3.46	0.3	0.25
		M	11	0.10	0.37	11	1.25	1.77
spect	(22)	**m**	3	0.02	0.02	3	0.02	0.04
		a	7.31	0.13	0.07	6.44	1.61	0.67
		M	20	0.88	0.29	20	8.97	10.73

Some of the experimental results

Dataset			Minimal explanation			Minimum explanation		
			size	SMT (s)	MILP (s)	size	SMT (s)	MILP (s)
australian	(14)	**m**	1	0.03	0.05	—	—	—
		a	8.79	1.38	0.33	—	—	—
		M	14	17.00	1.43	—	—	—
backache	(32)	**m**	13	0.13	0.14	—	—	—
		a	19.28	5.08	0.85	—	—	—
		M	26	22.21	2.75	—	—	—
breast-cancer	(9)	**m**	3	0.02	0.04	3	0.02	0.03
		a	5.15	0.65	0.20	4.86	2.18	0.41
		M	9	6.11	0.41	9	24.80	1.81
cleve	(13)	**m**	4	0.05	0.07	4	—	0.07
		a	8.62	3.32	0.32	7.89	—	5.14
		M	13	60.74	0.60	13	—	39.06
hepatitis	(19)	**m**	6	0.02	0.04	4	0.01	0.04
		a	11.42	0.07	0.06	9.39	4.07	2.89
		M	19	0.26	0.20	19	27.05	22.23
voting	(16)	**m**	3	0.01	0.02	3	0.01	0.02
		a	4.56	0.04	0.13	3.46	0.3	0.25
		M	11	0.10	0.37	11	1.25	1.77
spect	(22)	**m**	3	0.02	0.02	3	0.02	0.04
		a	7.31	0.13	0.07	6.44	1.61	0.67
		M	20	0.88	0.29	20	8.97	10.73

Some of the experimental results

Dataset			Minimal explanation			Minimum explanation		
			size	SMT (s)	MILP (s)	size	SMT (s)	MILP (s)
australian	(14)	**m**	1	0.03	0.05	—	—	—
		a	8.79	1.38	0.33	—	—	—
		M	14	17.00	1.43	—	—	—
backache	(32)	**m**	13	0.13	0.14	—	—	—
		a	19.28	5.08	0.85	—	—	—
		M	26	22.21	2.75	—	—	—
breast-cancer	(9)	**m**	3	0.02	0.04	3	0.02	0.03
		a	5.15	0.65	0.20	4.86	2.18	0.41
		M	9	6.11	0.41	9	24.80	1.81
cleve	(13)	**m**	4	0.05	0.07	4	—	0.07
		a	8.62	3.32	0.32	7.89	—	5.14
		M	13	60.74	0.60	13	—	39.06
hepatitis	(19)	**m**	6	0.02	0.04	4	0.01	0.04
		a	11.42	0.07	0.06	9.39	4.07	2.89
		M	19	0.26	0.20	19	27.05	22.23
voting	(16)	**m**	3	0.01	0.02	3	0.01	0.02
		a	4.56	0.04	0.13	3.46	0.3	0.25
		M	11	0.10	0.37	11	1.25	1.77
spect	(22)	**m**	3	0.02	0.02	3	0.02	0.04
		a	7.31	0.13	0.07	6.44	1.61	0.67
		M	20	0.88	0.29	20	8.97	10.73

Some of the experimental results

Dataset			Minimal explanation			Minimum explanation		
			size	SMT (s)	MILP (s)	size	SMT (s)	MILP (s)
australian	(14)	**m**	1	0.03	0.05	—	—	—
		a	8.79	1.38	0.33	—	—	—
		M	14	17.00	1.43	—	—	—
backache	(32)	**m**	13	0.13	0.14	—	—	—
		a	19.28	5.08	0.85	—	—	—
		M	26	22.21	2.75	—	—	—
breast-cancer	(9)	**m**	3	0.02	0.04	3	0.02	0.03
		a	5.15	0.65	0.20	4.86	2.18	0.41
		M	9	6.11	0.41	9	24.80	1.81
cleve	(13)	**m**	4	0.05	0.07	4	—	0.07
		a	8.62	3.32	0.32	7.89	—	5.14
		M	13	60.74	0.60	13	—	39.06
hepatitis	(19)	**m**	6	0.02	0.04	4	0.01	0.04
		a	11.42	0.07	0.06	9.39	4.07	2.89
		M	19	0.26	0.20	19	27.05	22.23
voting	(16)	**m**	3	0.01	0.02	3	0.01	0.02
		a	4.56	0.04	0.13	3.46	0.3	0.25
		M	11	0.10	0.37	11	1.25	1.77
spect	(22)	**m**	3	0.02	0.02	3	0.02	0.04
		a	7.31	0.13	0.07	6.44	1.61	0.67
		M	20	0.88	0.29	20	8.97	10.73

Comparing quality to compilation-based approach

- **“Congressional Voting Records”** dataset

Comparing quality to compilation-based approach

- **“Congressional Voting Records”** dataset
- `(0 1 0 1 1 1 0 0 0 0 0 0 1 1 0 1)` — data sample **(16 features)**

Comparing quality to compilation-based approach

- **"Congressional Voting Records"** dataset
- `(0 1 0 1 1 1 0 0 0 0 0 0 1 1 0 1)` — data sample **(16 features)**

smallest size explanations computed by **compilation for BN**: [SCD18]

- `(    0 1 1   0 0 0         1 1 0   )` — **9 literals**
- `(    0 1 1 1   0 0         1 1 0   )` — **9 literals**

Comparing quality to compilation-based approach

- **"Congressional Voting Records"** dataset
- (0 1 0 1 1 1 0 0 0 0 0 0 1 1 0 1) — data sample **(16 features)**

smallest size explanations computed by **compilation for BN**: [SCD18]

		0	1	1		0	0	0				1	1	0		— **9 literals**
		0	1	1	1		0	0				1	1	0		— **9 literals**

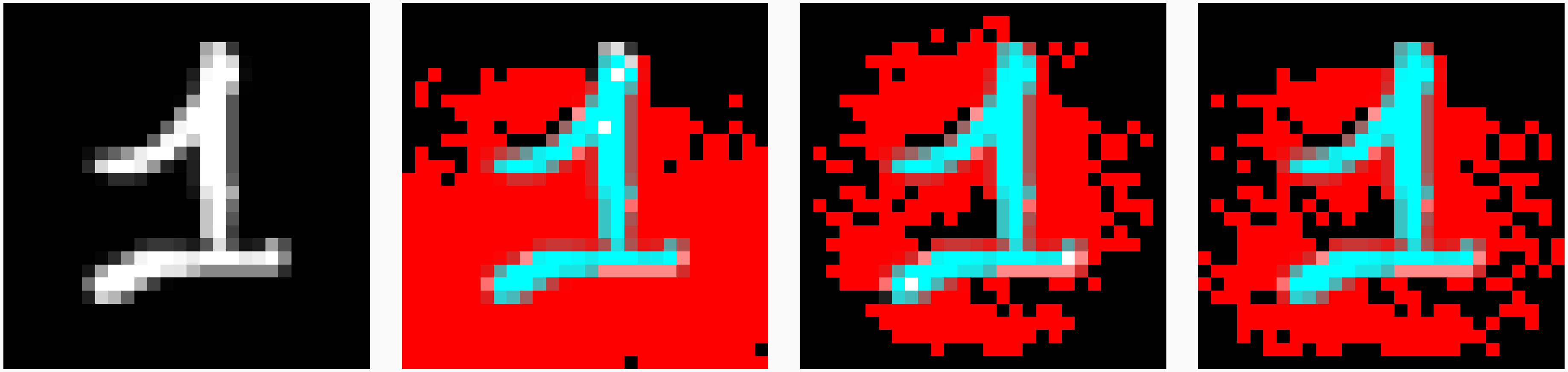
subset-minimal explanations computed by **search for ReLU-NNs**: [INMS19]

			1					0		0				0		— **4 literals**
			1					0		0						— **3 literals**
		0	1					0		0				0		— **5 literals**
		0	1					0		0					1	— **5 literals**

What does it mean?

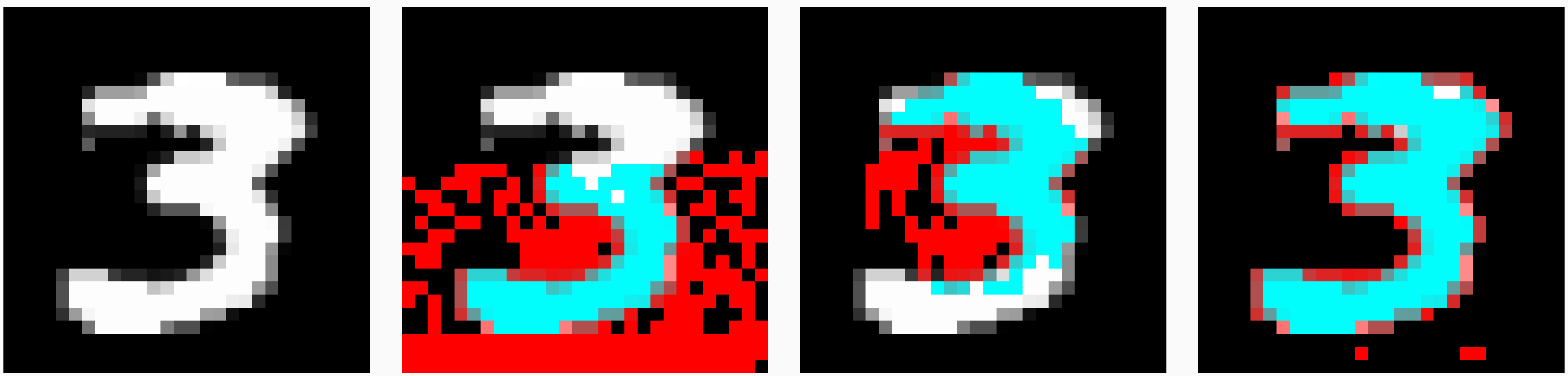
explanations can *hint* on the classifier quality!

MNIST examples



(a) (b) (c) (d)

Figure 1: Possible minimal explanations for digit one.



(a) (b) (c) (d)

And so what?

explanations are not equally good!

Summary on search-based explanations

principled approach to XAI

Summary on search-based explanations

principled approach to XAI

based on **abductive reasoning**

Summary on search-based explanations

principled approach to XAI

based on **abductive reasoning**

applies a **reasoning oracle**, e.g. SMT or MILP

Summary on search-based explanations

principled approach to XAI

based on **abductive reasoning**

applies a **reasoning oracle**, e.g. SMT or MILP

provides **minimality guarantees**

Summary on search-based explanations

principled approach to XAI

based on **abductive reasoning**

applies a **reasoning oracle**, e.g. SMT or MILP

provides **minimality guarantees**

global explanations!

What next?

What next?

enumeration of **explanations**?

What next?

enumeration of **explanations**?

preferences over explanations?

What next?

enumeration of **explanations**?

preferences over explanations?

assessment of heuristic approaches!

Assessing heuristic approaches

Heuristic approaches – a recap

heuristic approaches

(e.g. **LIME**, **Anchor**, **SHAP**) [RSG16, RSG18, LL17]

Heuristic approaches – a recap

heuristic approaches
(e.g. **LIME**, **Anchor**, **SHAP**) [RSG16, RSG18, LL17]

local explanations

Heuristic approaches – a recap

heuristic approaches
(e.g. **LIME**, **Anchor**, **SHAP**) [RSG16, RSG18, LL17]

local explanations
no minimality **guarantees**

Assessment setup

[INM19]

how good are heuristic explanations?

Assessment setup

[INM19]

how good are heuristic explanations?

let's check for boosted trees [CG16]

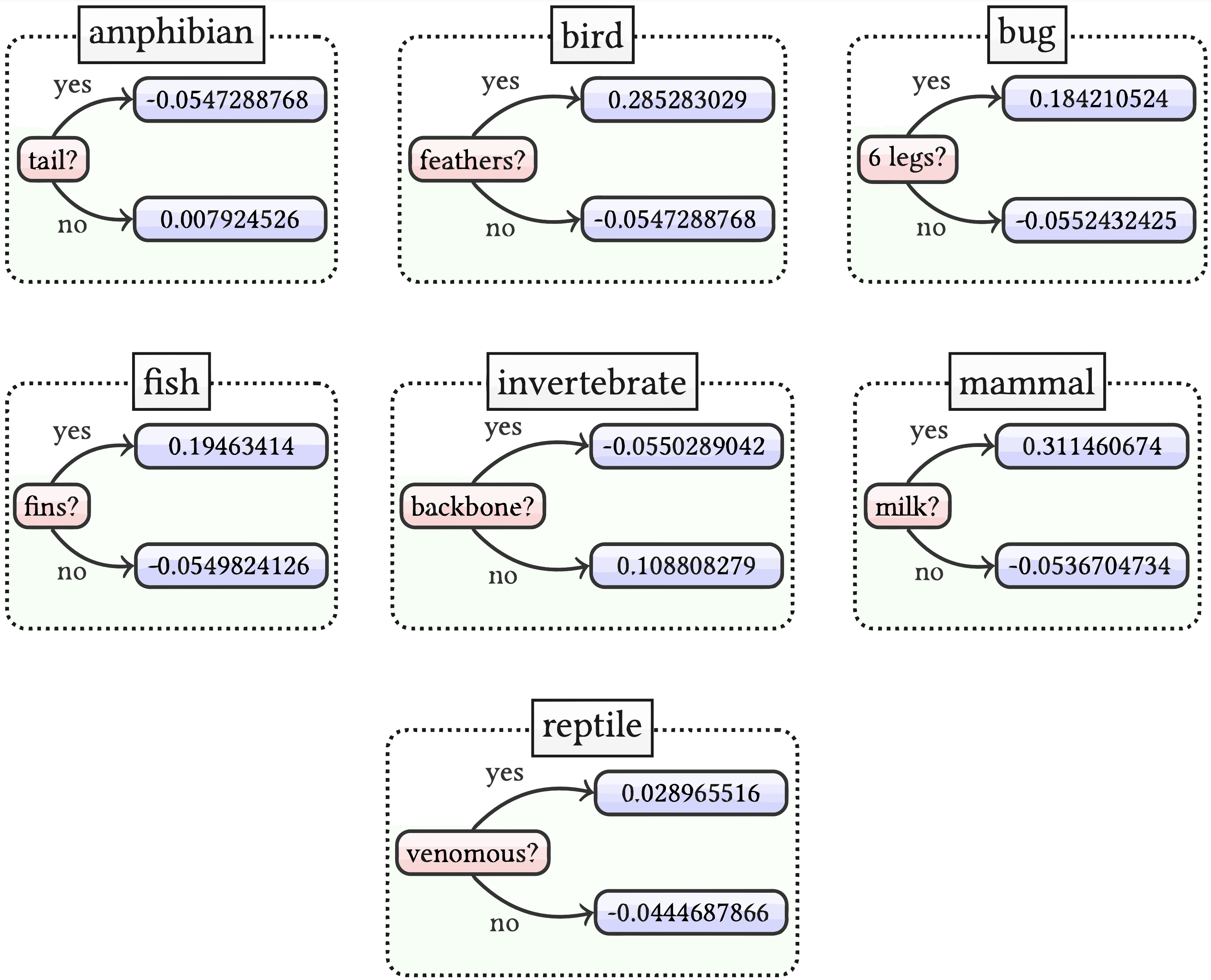
Assessment setup

[INM19]

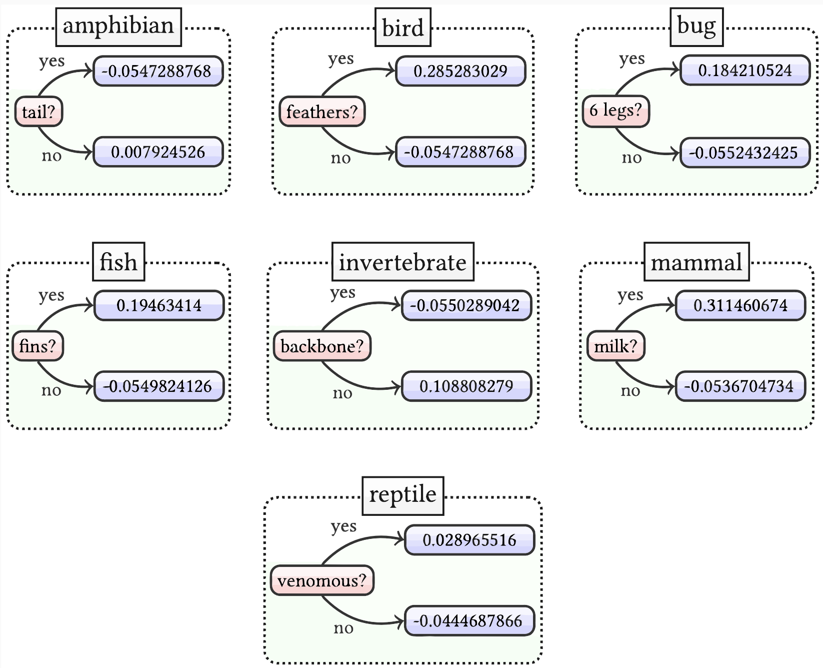
how good are heuristic explanations?

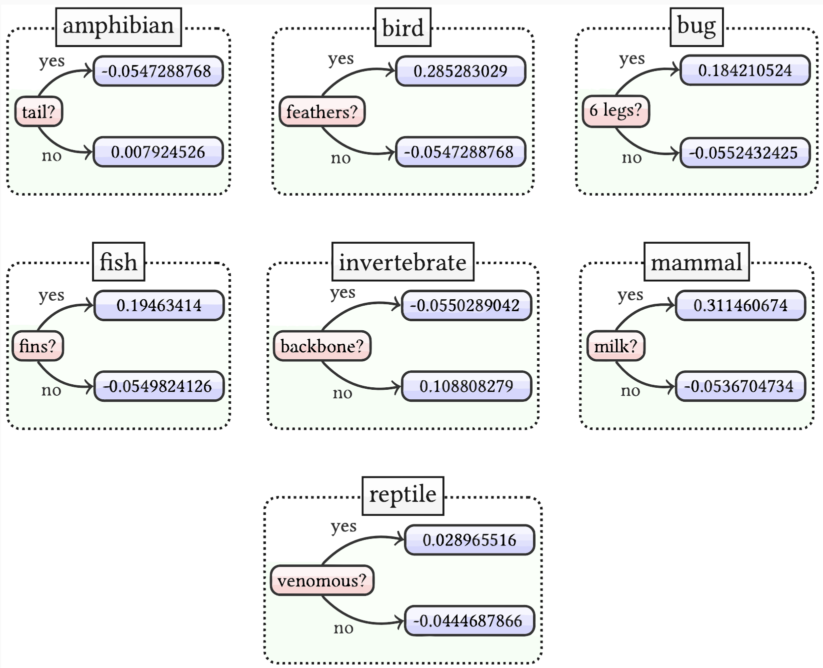
let's check for boosted trees [CG16]

(easy to encode) [BLM15, LMB17, VZY17, INM19]



amphibian
tail?
yes
-0.0547288768
no
0.007924526
bird
feathers?
yes
0.285283029
no
-0.0547288768
bug
6 legs?
yes
0.184210524
no
-0.0552432425
fish
fins?
yes
0.19463414
no
-0.0549824126
invertebrate
backbone?
yes
-0.0550289042
no
0.108808279
mammal
milk?
yes
0.311460674
no
-0.0536704734
reptile
venomous?
yes
0.028965516
no
-0.0444687866

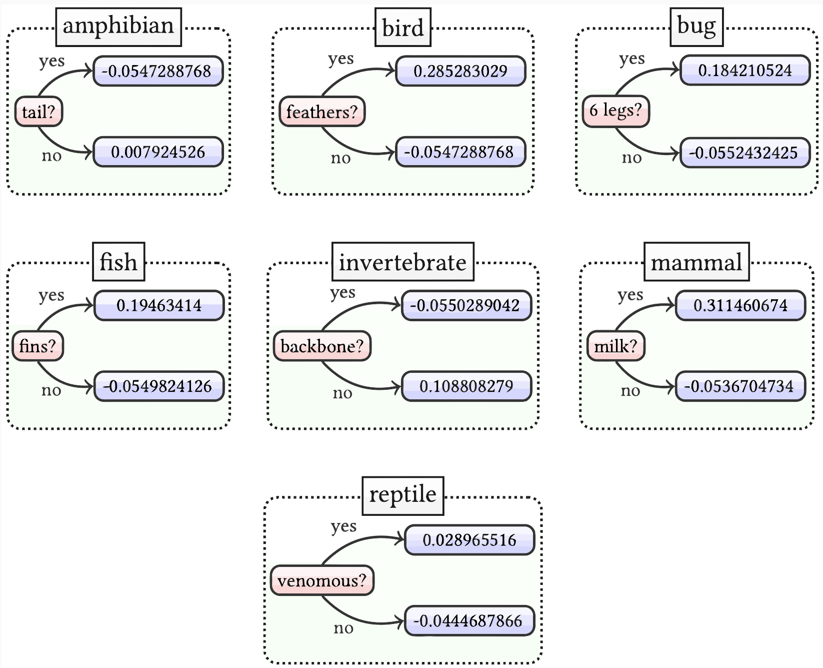

amphibian
yes
-0.0547288768
tail?
no
0.007924526
bird
yes
0.285283029
feathers?
no
-0.0547288768
bug
yes
0.184210524
6 legs?
no
-0.0552432425
fish
yes
0.19463414
fins?
no
-0.0549824126
invertebrate
yes
-0.0550289042
backbone?
no
0.108808279
mammal
yes
0.311460674
milk?
no
-0.0536704734
reptile
yes
0.028965516
venomous?
no
-0.0444687866



input instance:

IF (animal_name = pitviper) $\wedge$ $\neg$hair $\neg$feathers $\wedge$ eggs $\wedge$ $\neg$milk $\wedge$ $\neg$airborne$\wedge$ $\neg$aquatic $\wedge$ predator $\wedge$ $\neg$toothed $\wedge$ $\neg$fins$\wedge$ (legs $= 0$) $\wedge$ tail $\wedge$ $\neg$domestic $\wedge$ $\neg$catsize

THEN (class = reptile)



input instance:

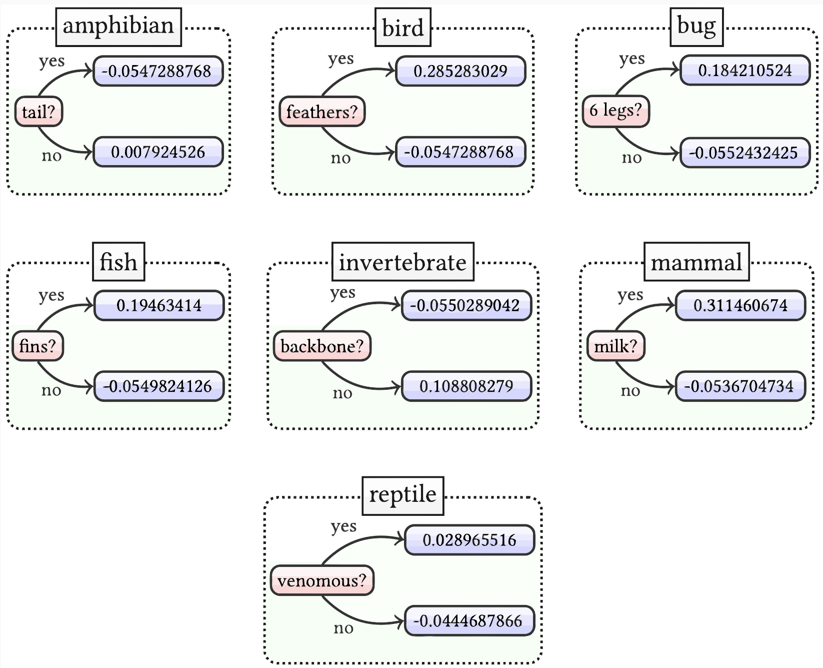
IF (animal_name = pitviper) $\wedge$ ¬hair ¬feathers $\wedge$ eggs $\wedge$ ¬milk $\wedge$ ¬airborne$\wedge$ ¬aquatic $\wedge$ predator $\wedge$ ¬toothed $\wedge$ ¬fins$\wedge$ (legs $= 0$) $\wedge$ tail $\wedge$ ¬domestic $\wedge$ ¬catsize

THEN (class = reptile)

Anchor's explanation:

IF ¬hair $\wedge$ ¬milk $\wedge$ ¬toothed $\wedge$ ¬fins

THEN (class = reptile)



input instance:

IF (animal_name $=$ pitviper) $\land$ ¬hair
¬feathers $\land$ eggs $\land$ ¬milk $\land$ ¬airborne$\land$
¬aquatic $\land$ predator $\land$ ¬toothed $\land$ ¬fins$\land$
(legs $= 0$) $\land$ tail $\land$ ¬domestic $\land$ ¬catsize
THEN (class $=$ reptile)

Anchor's explanation:

IF ¬hair $\land$ ¬milk $\land$ ¬toothed $\land$ ¬fins
THEN (class $=$ reptile)

counterexample!

IF (animal_name $=$ toad) $\land$ ¬hair
¬feathers $\land$ eggs $\land$ ¬milk $\land$ ¬airborne$\land$
¬aquatic $\land$ ¬predator $\land$ ¬toothed $\land$ ¬fins$\land$
(legs $= 4$) $\land$ ¬tail $\land$ ¬domestic $\land$ ¬catsize
THEN (class $=$ amphibian)

how?

how?

given $\mathcal{E}_h$, $\mathcal{E}_h \vDash (\mathcal{M} \to \pi)$

how?

given $\mathcal{E}_h$, $\mathcal{E}_h \vDash (\mathcal{M} \to \pi)$

how?

given $\mathcal{E}_h$, $\mathcal{E}_h \models (\mathcal{M} \to \pi)$

$\mathcal{E}_h \wedge \mathcal{M} \wedge \neg\pi$ — **satisfiable**

how?

given $\mathcal{E}_h$, $\mathcal{E}_h \models (\mathcal{M} \to \pi)$

$$\mathcal{E}_h \wedge \mathcal{M} \wedge \neg\pi$$ — **satisfiable**

(in fact, this formula can have **many models**)

Repairing heuristic explanations

Input: model $\mathcal{M}$, initial cube $\mathcal{I}$, heuristic explanation $\mathcal{E}_h$, prediction π
Output: ***Subset-minimal*** explanation $\mathcal{E}_m$

```
begin
    (I₁, I₂) ← (I \ E_h, E_h)
    for l ∈ I₁ :
        if Entails(I₁ ∪ I₂ \ {l}, M → π) :
            I₁ ← I₁ \ {l}
    for l ∈ I₂ :
        if Entails(I₁ ∪ I₂ \ {l}, M → π) :
            I₂ ← I₂ \ {l}
    return I₁ ∪ I₂
end
```

begin

$(\mathcal{I}_1, \mathcal{I}_2) \leftarrow (\mathcal{I} \setminus \mathcal{E}_h, \mathcal{E}_h)$

for $l \in \mathcal{I}_1$ **:**

  if Entails$(\mathcal{I}_1 \cup \mathcal{I}_2 \setminus \{l\}, \mathcal{M} \to \pi)$ **:**

    $\mathcal{I}_1 \leftarrow \mathcal{I}_1 \setminus \{l\}$

for $l \in \mathcal{I}_2$ **:**

  if Entails$(\mathcal{I}_1 \cup \mathcal{I}_2 \setminus \{l\}, \mathcal{M} \to \pi)$ **:**

    $\mathcal{I}_2 \leftarrow \mathcal{I}_2 \setminus \{l\}$

return $\mathcal{I}_1 \cup \mathcal{I}_2$

end

Repairing heuristic explanations

Input: model $\mathcal{M}$, initial cube $\mathcal{I}$, heuristic explanation $\mathcal{E}_h$, prediction π
Output: ***Subset-minimal*** explanation $\mathcal{E}_m$

```
begin
    (I1, I2) ← (I \ Eh, Eh)
    for l ∈ I1 :
        if Entails(I1 ∪ I2 \ {l}, M → π) :
            I1 ← I1 \ {l}
    for l ∈ I2 :
        if Entails(I1 ∪ I2 \ {l}, M → π) :
            I2 ← I2 \ {l}
    return I1 ∪ I2
end
```

$$(\mathcal{I}_1, \mathcal{I}_2) \leftarrow (\mathcal{I} \setminus \mathcal{E}_h, \mathcal{E}_h)$$

Repairing heuristic explanations

Input: model $\mathcal{M}$, initial cube $\mathcal{I}$, heuristic explanation $\mathcal{E}_h$, prediction π
Output: ***Subset-minimal*** explanation $\mathcal{E}_m$

```
begin
    (𝓘₁, 𝓘₂) ← (𝓘 \ 𝓔ₕ, 𝓔ₕ)

    for l ∈ 𝓘₁ :
        if Entails(𝓘₁ ∪ 𝓘₂ \ {l}, 𝓜 → π) :
            𝓘₁ ← 𝓘₁ \ {l}

    for l ∈ 𝓘₂ :
        if Entails(𝓘₁ ∪ 𝓘₂ \ {l}, 𝓜 → π) :
            𝓘₂ ← 𝓘₂ \ {l}

    return 𝓘₁ ∪ 𝓘₂
end
```

Repairing heuristic explanations

Input: model $\mathcal{M}$, initial cube $\mathcal{I}$, heuristic explanation $\mathcal{E}_h$, prediction π
Output: ***Subset-minimal*** explanation $\mathcal{E}_m$

```
begin
    (I_1, I_2) ← (I \ E_h, E_h)
    for l ∈ I_1 :
        if Entails(I_1 ∪ I_2 \ {l}, M → π) :
            I_1 ← I_1 \ {l}
    for l ∈ I_2 :
        if Entails(I_1 ∪ I_2 \ {l}, M → π) :
            I_2 ← I_2 \ {l}
    return I_1 ∪ I_2
end
```

incorrect explanation

$$\begin{array}{ll} \textbf{IF} & \neg\text{hair} \wedge \neg\text{milk} \wedge \neg\text{toothed} \wedge \neg\text{fins} \\ \textbf{THEN} & (\text{class} = \text{reptile}) \end{array}$$

incorrect explanation

IF $\neg\text{hair} \wedge \neg\text{milk} \wedge \neg\text{toothed} \wedge \neg\text{fins}$

THEN $(\text{class} = \text{reptile})$

repaired explanation

IF $\neg\text{feathers} \wedge \neg\text{milk} \wedge \text{backbone} \wedge$
$\neg\text{fins} \wedge (\text{legs} = 0) \wedge \text{tail}$

THEN $(\text{class} = \text{reptile})$

Refining heuristic explanations

Input: model $\mathcal{M}$, heuristic explanation $\mathcal{E}_h$, prediction π
Output: ***Subset-minimal*** explanation $\mathcal{E}_m$

```
begin
   for l ∈ E_h :
      if Entails(E_h \ {l}, M → π) :
         E_h ← E_h \ {l}
   return E_h
end
```

Assessment experiment

3 datasets from Anchor [RSG18]

Assessment experiment

3 datasets from Anchor [RSG18]

2 additional datasets from **FairML** and **ProPublica** [Fai16, ALMK16]
[FSV15, FFM+15, FSV+19]

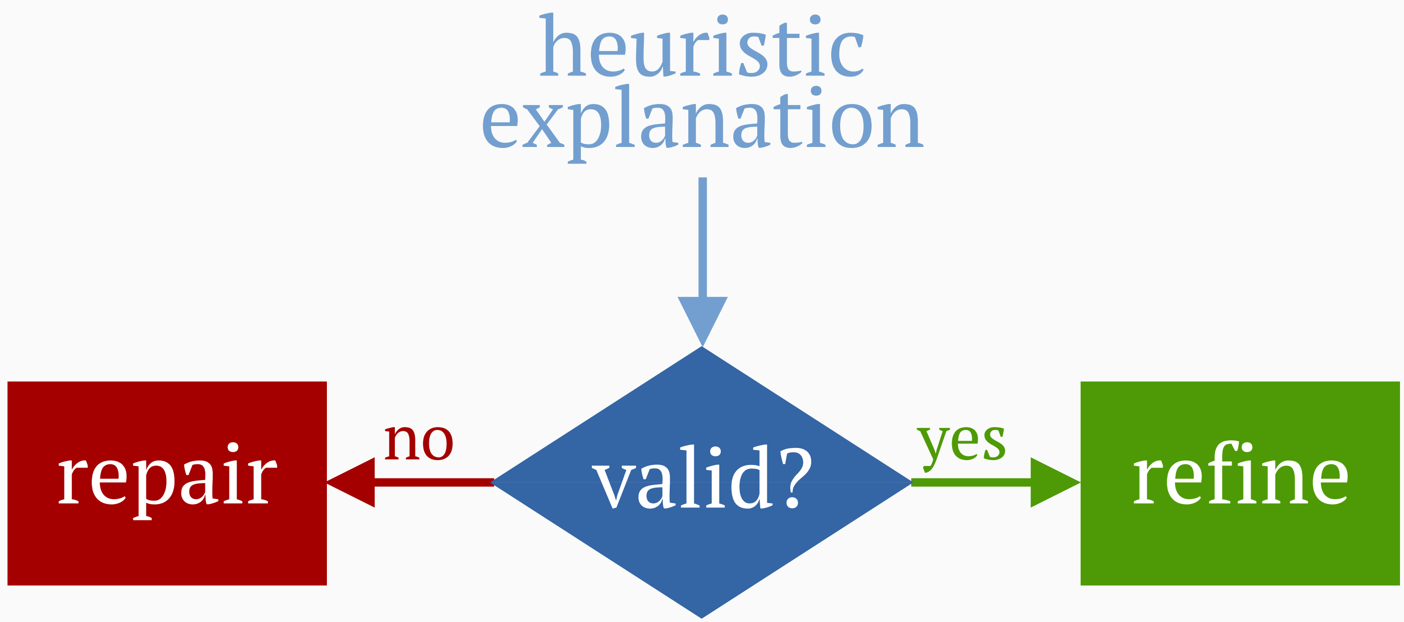
Assessment experiment

3 datasets from Anchor [RSG18]

2 additional datasets from **FairML** and **ProPublica** [Fai16, ALMK16] [FSV15, FFM+15, FSV+19]

target **all data samples**

Assessment experiment



Assessment experiment

Dataset	(# unique)	Explanations								
		optimistic			pessimistic			realistic		
		LIME	Anchor	SHAP	LIME	Anchor	SHAP	LIME	Anchor	SHAP
adult	(5579)	61.3%	80.5%	70.7%	7.9%	1.6%	10.2%	30.8%	17.9%	19.1%
lending	(4414)	24.0%	3.0%	17.0%	0.4%	0.0%	2.5%	75.6%	97.0%	80.5%
rcdv	(3696)	94.1%	99.4%	85.9%	4.6%	0.4%	7.9%	1.3%	0.2%	6.2%
compas	(778)	71.9%	84.4%	60.4%	20.6%	1.7%	27.8%	7.5%	13.9%	11.8%
german	(1000)	85.3%	99.7%	63.0%	14.6%	0.2%	37.0%	0.1%	0.1%	0.0%

Assessment experiment

Dataset	(# unique)	Explanations								
		optimistic			pessimistic			realistic		
		LIME	Anchor	SHAP	LIME	Anchor	SHAP	LIME	Anchor	SHAP
adult	(5579)	61.3%	80.5%	70.7%	7.9%	1.6%	10.2%	30.8%	17.9%	19.1%
lending	(4414)	24.0%	3.0%	17.0%	0.4%	0.0%	2.5%	75.6%	97.0%	80.5%
rcdv	(3696)	94.1%	99.4%	85.9%	4.6%	0.4%	7.9%	1.3%	0.2%	6.2%
compas	(778)	71.9%	84.4%	60.4%	20.6%	1.7%	27.8%	7.5%	13.9%	11.8%
german	(1000)	85.3%	99.7%	63.0%	14.6%	0.2%	37.0%	0.1%	0.1%	0.0%

so should we trust heuristic approaches?

Assessment experiment

Dataset	(# unique)	Explanations								
		optimistic			pessimistic			realistic		
		LIME	Anchor	SHAP	LIME	Anchor	SHAP	LIME	Anchor	SHAP
adult	(5579)	61.3%	80.5%	70.7%	7.9%	1.6%	10.2%	30.8%	17.9%	19.1%
lending	(4414)	24.0%	3.0%	17.0%	0.4%	0.0%	2.5%	75.6%	97.0%	80.5%
rcdv	(3696)	94.1%	99.4%	85.9%	4.6%	0.4%	7.9%	1.3%	0.2%	6.2%
compas	(778)	71.9%	84.4%	60.4%	20.6%	1.7%	27.8%	7.5%	13.9%	11.8%
german	(1000)	85.3%	99.7%	63.0%	14.6%	0.2%	37.0%	0.1%	0.1%	0.0%

so should we trust heuristic approaches?

or better not?

let’s go further!

let's go further!

what about measuring precision of Anchor's explanations? [NSM+19]

What about measuring precision of Anchor's explanations? [NSM+19]

given model $\mathcal{M}$, input $\mathcal{I}$, prediction π, and explanation $\mathcal{E}$:

$$\mathit{prec}(\mathcal{E}) = \mathbb{E}_{\mathcal{D}(\mathcal{I}' \supset \mathcal{E})}[\mathcal{M}(\mathcal{I}') = \pi]$$

What about measuring precision of Anchor's explanations? [NSM+19]

given model $\mathcal{M}$, input $\mathcal{I}$, prediction π, and explanation $\mathcal{E}$:

$$\mathit{prec}(\mathcal{E}) = \mathbb{E}_{\mathcal{D}(\mathcal{I}' \supset \mathcal{E})}[\mathcal{M}(\mathcal{I}') = \pi]$$

alternatively, do approximate model counting for:

$$\mathcal{E} \land \mathcal{M} \land \neg\pi$$

What about measuring precision of Anchor's explanations? [NSM+19]

given model $\mathcal{M}$, input $\mathcal{I}$, prediction π, and explanation $\mathcal{E}$:

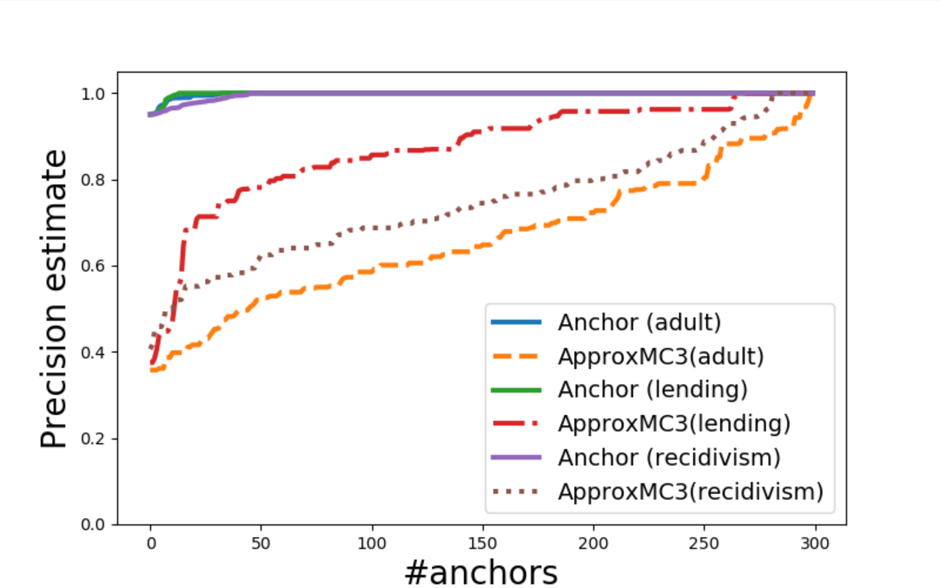
$$\mathit{prec}(\mathcal{E}) = \mathbb{E}_{\mathcal{D}(\mathcal{I}' \supset \mathcal{E})}[\mathcal{M}(\mathcal{I}') = \pi]$$

alternatively, do approximate model counting for:

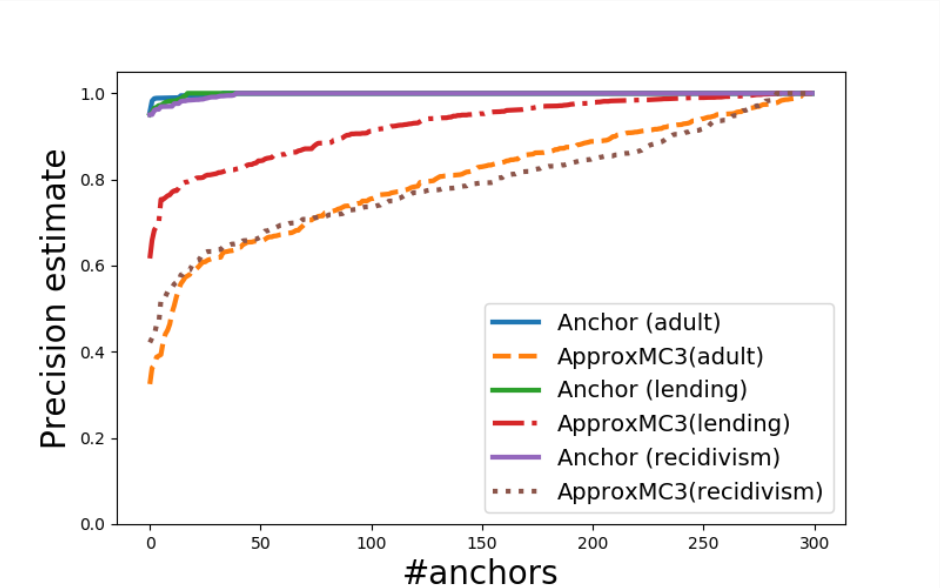
$$\mathcal{E} \wedge \mathcal{M} \wedge \neg\pi$$

(in fact, a bit more complicated but the idea is here)

Assessing heuristic explanations[1]



unconstrained feature space



samples with $\leq$ 50% difference

Summary

Summary

logic is helpful in XAI!

Summary

logic is helpful in XAI!

(for computing explanations but also assessing heuristic appoaches)

Summary

logic is helpful in XAI!

(for computing explanations but also assessing heuristic appoaches)

rigorous approach

Summary

logic is helpful in XAI!

(for computing explanations but also assessing heuristic appoaches)

rigorous approach

global explanations

Summary

logic is helpful in XAI!

(for computing explanations but also assessing heuristic appoaches)

rigorous approach

global explanations

minimality guarantees

Summary

logic is helpful in XAI!

(for computing explanations but also assessing heuristic appoaches)

rigorous approach

global explanations

minimality guarantees

(if one can encode and check entailment!)

challenges

challenges

scalability

(search or compilation?)

challenges

scalability

(**search** or **compilation**?)

other ML **models**, **reasoners**, **methods**?

challenges

scalability

(**search** or **compilation**?)

other ML **models, reasoners, methods**?

other types of explanations?

Future

challenges

scalability

(**search** or **compilation**?)

other ML **models, reasoners, methods**?

other types of explanations?

what about **other heuristic approaches?**

challenges

scalability

(**search** or **compilation**?)

other ML **models, reasoners, methods**?

other types of explanations?

what about **other heuristic approaches?**

hybrid approaches?

Future

relate XAI and verification?

References i

[ALMK16] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner.
Machine bias.
`http://tiny.cc/dd7mjz`, 2016.

[BLM15] Alessio Bonfietti, Michele Lombardi, and Michela Milano.
Embedding decision trees and random forests in constraint programming.
In *CPAIOR*, pages 74–90, 2015.

[CD03] Hei Chan and Adnan Darwiche.
Reasoning about Bayesian network classifiers.
In *UAI*, pages 107–115, 2003.

[CG16] Tianqi Chen and Carlos Guestrin.
XGBoost: A scalable tree boosting system.
In *KDD*, pages 785–794, 2016.

[Fai16] Auditing black-box predictive models.
`http://tiny.cc/6e7mjz`, 2016.

[FFM+15] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian.
Certifying and removing disparate impact.
In *KDD*, pages 259–268, 2015.

References ii

[FJ18] Matteo Fischetti and Jason Jo.
Deep neural networks and mixed integer linear optimization.
Constraints, 23(3):296–309, 2018.

[FSV15] Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian.
On algorithmic fairness, discrimination and disparate impact.
2015.

[FSV+19] Sorelle A. Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P. Hamilton, and Derek Roth.
A comparative study of fairness-enhancing interventions in machine learning.
In *FAT*, pages 329–338, 2019.

[IMM16] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva.
Propositional abduction with implicit hitting sets.
In *ECAI*, pages 1327–1335, 2016.

[INM19] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.
On validating, repairing and refining heuristic ML explanations.
CoRR, abs/1907.02509, 2019.

[INMS19] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.
Abduction-based explanations for machine learning models.
In *AAAI*, pages 1511–1519, 2019.

References iii

[LL17] Scott M. Lundberg and Su-In Lee.
A unified approach to interpreting model predictions.
In *NIPS*, pages 4765–4774, 2017.

[LMB17] Michele Lombardi, Michela Milano, and Andrea Bartolini.
Empirical decision model learning.
Artif. Intell., 244:343–367, 2017.

[NSM+19] Nina Narodytska, Aditya A. Shrotri, Kuldeep S. Meel, Alexey Ignatiev, and Joao Marques-Silva.
Assessing heuristic machine learning explanations with model counting.
In *SAT*, pages 267–278, 2019.

[RSG16] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin.
”why should I trust you?”: Explaining the predictions of any classifier.
In *KDD*, pages 1135–1144, 2016.

[RSG18] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin.
Anchors: High-precision model-agnostic explanations.
In *AAAI*, pages 1527–1535, 2018.

[SCD18] Andy Shih, Arthur Choi, and Adnan Darwiche.
A symbolic approach to explaining Bayesian network classifiers.
In *IJCAI*, pages 5103–5111, 2018.

References iv

[SCD19] Andy Shih, Arthur Choi, and Adnan Darwiche.
Compiling Bayesian network classifiers into decision graphs.
In *AAAI*, pages 7966–7974, 2019.

[SDC19] Andy Shih, Adnan Darwiche, and Arthur Choi.
Verifying binarized neural networks by Angluin-style learning.
In *SAT*, pages 354–370, 2019.

[VZY17] Sicco Verwer, Yingqian Zhang, and Qing Chuan Ye.
Auction optimization using regression trees and linear models as integer programs.
Artif. Intell., 244:368–395, 2017.

Rigorous Verification and Explanation of ML Models
Part 05

A. Ignatiev, J. Marques-Silva, K. Meel& N. Narodytska

Monash Univ, NU Singapore, VMWare Research & ANITI@Univ. Toulouse

February 08, 2020 | AAAI Tutorial SP1

1 Duality in Explanations

Overview

- Vast body of work on computing explanations (XPs)
  - Mostly heuristic approaches, with recent rigorous solutions
- Vast body of work on coping with adversarial examples (AEs)
  - Both heuristic and rigorous approaches

Overview

- Vast body of work on computing explanations (XPs)
  - Mostly heuristic approaches, with recent rigorous solutions
- Vast body of work on coping with adversarial examples (AEs)
  - Both heuristic and rigorous approaches
- Can XPs and AEs be somehow related?
  - Recent work observed that some connection existed, but formal connection has been elusive

Overview

- Vast body of work on computing explanations (XPs)
  - Mostly heuristic approaches, with recent rigorous solutions
- Vast body of work on coping with adversarial examples (AEs)
  - Both heuristic and rigorous approaches
- Can XPs and AEs be somehow related?
  - Recent work observed that some connection existed, but formal connection has been elusive
- Recent proposal of a (first) link between XPs and AEs [INM19]
  - Work exploits hitting set duality, first studied in model-based diagnosis [Rei87]

A well-known example

[RN10]

Example	Input Attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0–10	$y_1 =$ Yes
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30–60	$y_2 =$ No
x_3	No	Yes	No	No	Some	$	No	No	Burger	0–10	$y_3 =$ Yes
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10–30	$y_4 =$ Yes
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 =$ No
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0–10	$y_6 =$ Yes
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0–10	$y_7 =$ No
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0–10	$y_8 =$ Yes
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 =$ No
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10–30	$y_{10} =$ No
x_{11}	No	No	No	No	None	$	No	No	Thai	0–10	$y_{11} =$ No
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30–60	$y_{12} =$ Yes

A well-known example (Cont.)

- 10 features:

$$\{\text{A(lternate)}, \text{B(ar)}, \text{W(eekend)}, \text{H(ungry)}, \text{Pa(trons)}, \text{Pr(ice)}, \text{Ra(in)}, \text{Re(serv.)}, \text{T(ype)}, \text{E(stim.)}\}$$

- Example instance (x_1, with outcome $y_1 = \text{Yes}$):

$$\{\text{A}, \neg\text{B}, \neg\text{W}, \text{H}, (\text{Pa} = \text{Some}), (\text{Pr} = \$\$\$), \neg\text{Ra}, \text{Re}, (\text{T} = \text{French}), (\text{E} = 0\text{–}10)\}$$

- A possible decision set (obtained with some off-the-shelf tool, & function*):

$$
\begin{array}{llll}
\textbf{IF} & (\text{Pa} = \text{Some}) \land \neg(\text{E} = {>}60) & \textbf{THEN} & (\text{Wait} = \text{Yes}) \quad \text{(R1)} \\
\textbf{IF} & \text{W} \land \neg(\text{Pr} = \$\$\$) \land \neg(\text{E} = {>}60) & \textbf{THEN} & (\text{Wait} = \text{Yes}) \quad \text{(R2)} \\
\textbf{IF} & \neg\text{W} \land \neg(\text{Pa} = \text{Some}) & \textbf{THEN} & (\text{Wait} = \text{No}) \quad \text{(R3)} \\
\textbf{IF} & (\text{E} = {>}60) & \textbf{THEN} & (\text{Wait} = \text{No}) \quad \text{(R4)} \\
\textbf{IF} & \neg(\text{Pa} = \text{Some}) \land (\text{Pr} = \$\$\$) & \textbf{THEN} & (\text{Wait} = \text{No}) \quad \text{(R5)}
\end{array}
$$

Counterexamples & breaks

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \vDash (\mathcal{M} \rightarrow \rho)$, with $\rho \in \mathbb{K} \land \rho \neq \pi$

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \vDash (\mathcal{M} \rightarrow \rho)$, with $\rho \in \mathbb{K} \wedge \rho \neq \pi$

- Breaks:

  A literal τ_i breaks a set of literals $\mathcal{S}$ (each denoting a different feature) if $\mathcal{S}$ contains a literal inconsistent with τ_i

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \vDash (\mathcal{M} \rightarrow \rho)$, with $\rho \in \mathbb{K} \wedge \rho \neq \pi$

- Breaks:

  A literal τ_i breaks a set of literals $\mathcal{S}$ (each denoting a different feature) if $\mathcal{S}$ contains a literal inconsistent with τ_i

- Back to the example, consider prediction $(\text{Wait} = \text{Yes})$:

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \vDash (\mathcal{M} \rightarrow \rho)$, with $\rho \in \mathbb{K} \wedge \rho \neq \pi$

- Breaks:

  A literal τ_i breaks a set of literals $\mathcal{S}$ (each denoting a different feature) if $\mathcal{S}$ contains a literal inconsistent with τ_i

- Back to the example, consider prediction $(\mathsf{Wait} = \mathsf{Yes})$:
  - Using (R1) (and assuming a consistent instance), an explanation is:

$$(\mathsf{Pa} = \mathsf{Some}) \wedge \neg(\mathsf{E} = {>}60)$$

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \vDash (\mathcal{M} \rightarrow \rho)$, with $\rho \in \mathbb{K} \land \rho \neq \pi$

- Breaks:

  A literal τ_i breaks a set of literals $\mathcal{S}$ (each denoting a different feature) if $\mathcal{S}$ contains a literal inconsistent with τ_i

- Back to the example, consider prediction $(\text{Wait} = \text{Yes})$:
  - Using (R1) (and assuming a consistent instance), an explanation is:

$$(\text{Pa} = \text{Some}) \land \neg(\text{E} = {>}60)$$

  - Due to (R5), a counterexample is:

$$\neg(\text{Pa} = \text{Some}) \land (\text{Pr} = \$\$\$)$$

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \vDash (\mathcal{M} \to \rho)$, with $\rho \in \mathbb{K} \wedge \rho \neq \pi$

- Breaks:

  A literal τ_i breaks a set of literals $\mathcal{S}$ (each denoting a different feature) if $\mathcal{S}$ contains a literal inconsistent with τ_i

- Back to the example, consider prediction $(\mathsf{Wait} = \mathsf{Yes})$:
  - Using (R1) (and assuming a consistent instance), an explanation is:

$$(\mathsf{Pa} = \mathsf{Some}) \wedge \neg(\mathsf{E} = {>}60)$$

  - Due to (R5), a counterexample is:

$$\neg(\mathsf{Pa} = \mathsf{Some}) \wedge (\mathsf{Pr} = \$\$\$)$$

  - XP $\mathcal{S}_1 = \{(\mathsf{Pa} = \mathsf{Some}), \neg(\mathsf{E} = {>}60)\}$ breaks CEx $\mathcal{S}_2 = \{\neg(\mathsf{Pa} = \mathsf{Some}), (\mathsf{Pr} = \$\$\$)\}$ and vice-versa

Some preliminary results

1. Relationship between XPs with CEx's:

Some preliminary results

1. Relationship between XPs with CEx's:
    - Each XP breaks every CEx

Some preliminary results

1. Relationship between XPs with CEx's:
    - Each XP breaks every CEx
    - Each CEx breaks every XP

Some preliminary results

1. Relationship between XPs with CEx's:
    - Each XP breaks every CEx
    - Each CEx breaks every XP

$\therefore$ XPs can be computed from all CEx's (by HSD) and vice-versa

Some preliminary results

1. Relationship between XPs with CEx's:
   - Each XP breaks every CEx
   - Each CEx breaks every XP

   $\therefore$ XPs can be computed from all CEx's (by HSD) and vice-versa

2. Given instance $\mathcal{I}$, an AE can be computed from closest CEx

Revisiting the example

- Restaurant dataset
- ML model is decision set (shown earlier)
- Prediction is $(\text{Wait} = \text{Yes})$

- Global explanations:
  1. $(\text{Pa} = \text{Some}) \land \neg(\text{E} = {>}60)$
  2. $\text{W} \land \neg(\text{Pr} = \$\$\$) \land \neg(\text{E} = {>}60)$

- Counterexamples:
  1. $\neg\text{W} \land \neg(\text{Pa} = \text{Some})$
  2. $(\text{E} = {>}60)$
  3. $\neg(\text{Pa} = \text{Some}) \land (\text{Pr} = \$\$\$)$

- The XP's break the CEx's and vice-versa

2 Wrap-up

Many challenges

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?
- More efficient (and still rigorous) alternatives to prime-based explanations?

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?
- More efficient (and still rigorous) alternatives to prime-based explanations?
  - **Q:** Basis for developing safe heuristics?

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?
- More efficient (and still rigorous) alternatives to prime-based explanations?
  - **Q:** Basis for developing safe heuristics?
- Scaling the learning of interpretable models?

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?
- More efficient (and still rigorous) alternatives to prime-based explanations?
  - **Q:** Basis for developing safe heuristics?
- Scaling the learning of interpretable models?
  - **Q:** How to target large datasets?
  - **Q:** Mechanisms for avoiding overfitting?

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?
- More efficient (and still rigorous) alternatives to prime-based explanations?
  - **Q:** Basis for developing safe heuristics?
- Scaling the learning of interpretable models?
  - **Q:** How to target large datasets?
  - **Q:** Mechanisms for avoiding overfitting?
- Exploiting logic in learning black-box models [FBD+19]

Questions?

References i

[FBD+19] Marc Fischer, Mislav Balunovic, Dana Drachsler-Cohen, Timon Gehr, Ce Zhang, and Martin T. Vechev.
DL2: training and querying neural networks with logic.
In *ICML*, pages 1931–1941, 2019.

[INM19] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.
On relating explanations and adversarial examples.
In *NeurIPS*, pages 15857–15867, 2019.

[Rei87] Raymond Reiter.
A theory of diagnosis from first principles.
Artif. Intell., 32(1):57–95, 1987.

[RN10] Stuart J. Russell and Peter Norvig.
***Artificial Intelligence - A Modern Approach*.**
Pearson Education, 2010.