Controller Synthesis and its Magical Futures

Anna Lukina

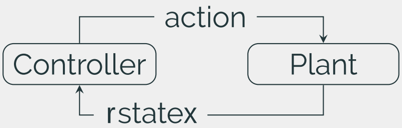
March 8, 2021

@ Synthesis of Models and Systems Seminar

IST AUSTRIA
Institute of Science and Technology

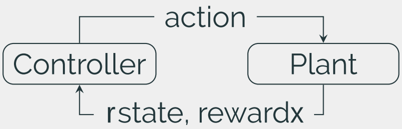
SIMONS INSTITUTE
for the Theory of Computing

Controller Design



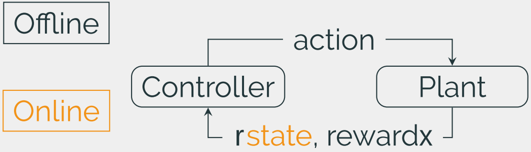
Controller Design via Deep RL

Offline



Goal: maximize expected rewards

Controller Design via Deep RL



Goal: maximize expected rewards

What if the agent has never seen this state offline?

Key Contributions

Current research focus: reliable design of learned systems via combination of formal methods and machine learning

(FMCAD-20) Formal Methods with a Touch of Magic. Alizadeh Alamdari, Avni, Henzinger, **Lukina**.

(ECAI-20) Outside the Box: Abstraction-Based Monitoring of Neural Networks. Henzinger, **Lukina**, Schilling.

(ATVA-17) Attacking the V: on the Resiliency of Adaptive-Horizon MPC. Smolka, Tiwari, Esterle, **Lukina**, Yang, Grosu.

(TACAS-17) ARES: Adaptive Receding-Horizon Synthesis of Optimal Plans. **Lukina**, Esterle, Hirsch, Bartocci, Yang, Tiwari, Smolka, Grosu.

A Touch of Magic 🪄

Formal Methods with a Touch of Magic [FMCAD 2020]

Reactive Synthesis

Given a specification ′, finds a controller that ensures the plant satisfies ′

Deep RL

Optimizes performance

Formal Methods with a Touch of Magic [FMCAD 2020]

Reactive Synthesis

Given a specification ′, finds a controller that ensures the plant satisfies ′

No performance guarantee

Deep RL

Optimizes performance

No correctness guarantee

Formal Methods with a Touch of Magic [FMCAD 2020]

Reactive Synthesis

Given a specification ′, finds a controller that ensures the plant satisfies ′

A model

Deep RL

Optimizes performance

Follow advice as closely as possible

Formal Methods with a Touch of Magic [FMCAD 2020]

Reactive Synthesis

Given a specification ′, finds a controller that ensures the plant satisfies ′

Follow advice as closely as possible

A model

Deep RL

Optimizes performance

Neural networks are large and uninterpretable

Formal Methods with a Touch of Magic [FMCAD 2020]

Reactive Synthesis

Given a specification ′, finds a controller that ensures the plant satisfies ′

Follow MB as closely as possible

A model

Deep RL

Optimizes performance

Magic Book

Approximates the agent and small enough for formal methods

Formal Methods with a Touch of Magic [FMCAD 2020]

Reactive Synthesis

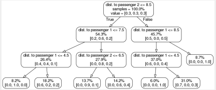
Given a specification ′, finds a controller that ensures the plant satisfies ′

Follow MB as closely as possible

A model

Deep RL

Optimizes performance

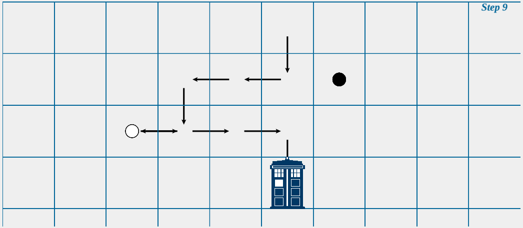


Magic Book

Approximates the agent and small enough for formal methods

Verification of the Magic Book [FMCAD 2020]

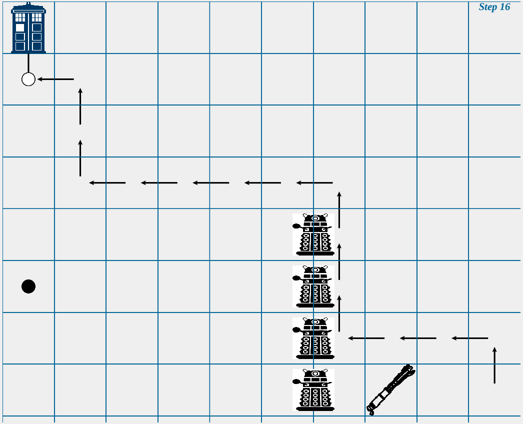
- “the taxi never enters a loop in which no passenger is collected”



A state at time i: $\langle x_0^i, y_0^i, \ldots, x_k^i, y_k^i \rangle$

Control Synthesis by the Magic Book [FMCAD 2020]

- "reach a gas station every *t* time steps"



Performance and Explainability [FMCAD 2020]

Num. of collected passengers	RF(5,6)	Wizard
Avg. performance	154	159
Max. performance	194	200
Synthesis avg. performance	96	–

Performance and Explainability [FMCAD 2020]

Num. of collected passengers	RF(5,6)	Wizard
Avg. performance	154	159
Max. performance	194	200
Synthesis avg. performance	96	–

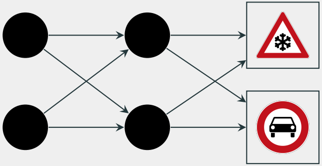
“Passenger 2 is collected first”: $_{1\&i\&`}\ x_j^i \quad x_j^0\ 0\ y_j^i \quad y_j^0\ \text{¾}\ j \quad 1;3$

$$x_2^{`} \quad x_2^0\ 0\ y_2^{`} \quad y_2^0$$

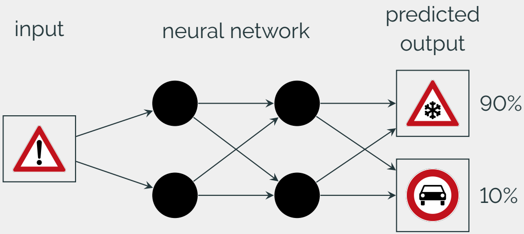
Bound	Passenger 2	
	runtime	succ. ratio
6	0.25 s	85 %
7	0.30 s	87.2 %
8	0.36 s	89.9 %
9	0.47 s	82.2 %

Reaction to Novel Input Classes

neural network



Reaction to Novel Input Classes

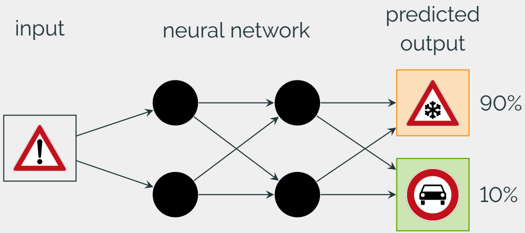


Must output "do not know"

Outside the Box

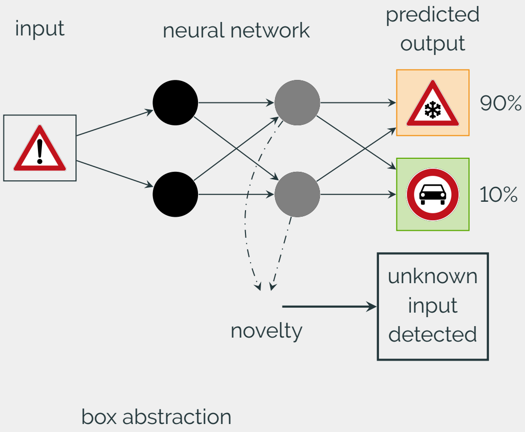
Abstraction-Based Monitoring of Neural Networks

[ECAI 2020]



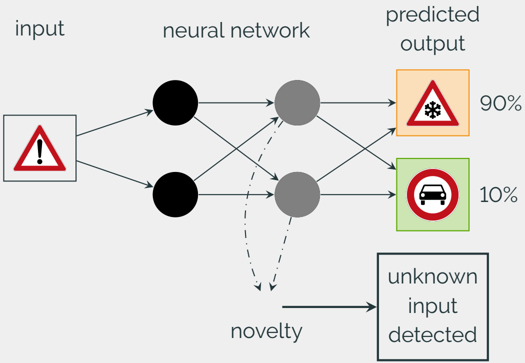
Abstraction-Based Monitoring of Neural Networks

[ECAI 2020]



Abstraction-Based Monitoring of Neural Networks

[ECAI 2020]

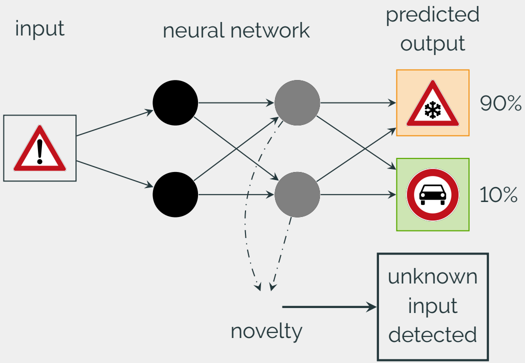


box abstraction

1. Computationally cheap

Abstraction-Based Monitoring of Neural Networks

[ECAI 2020]

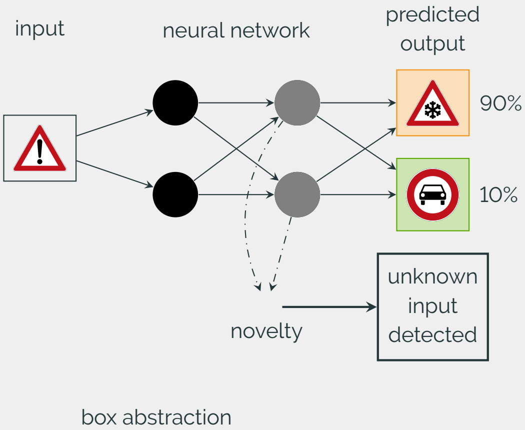


box abstraction

1. Computationally cheap
2. Effective in detecting novelties

Abstraction-Based Monitoring of Neural Networks

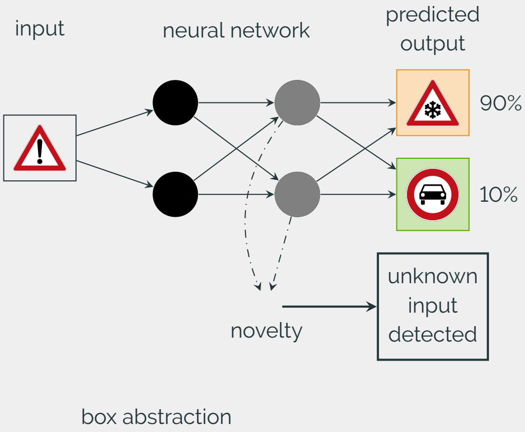
[ECAI 2020]



1. Computationally cheap
2. Effective in detecting novelties
3. Data and model independent

Abstraction-Based Monitoring of Neural Networks

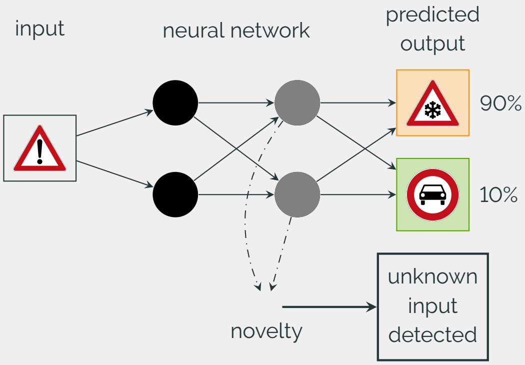
[ECAI 2020]



1. Computationally cheap
2. Effective in detecting novelties
3. Data and model independent
4. Easy to integrate

Abstraction-Based Monitoring of Neural Networks

[ECAI 2020]



box abstraction

1. Computationally cheap
2. Effective in detecting novelties
3. Data and model independent
4. Easy to integrate
5. Flexible to user configuration

Neural Networks in Dynamic Environments

Real-time object detection with neural networks[1]:

[1] https://pjreddie.com/darknet/yolo/

Neural Networks in Dynamic Environments

Real-time object detection with neural networks[1]:

- Trained on cats, dogs, and cars.

[1] https://pjreddie.com/darknet/yolo/

Neural Networks in Dynamic Environments

Real-time object detection with neural networks[1]:

- Trained on cats, dogs, and cars.
- Their food is frequently disappearing.

[1] https://pjreddie.com/darknet/yolo/

Neural Networks in Dynamic Environments

Real-time object detection with neural networks[1]:

- Trained on cats, dogs, and cars.
- Their food is frequently disappearing.

[1] https://pjreddie.com/darknet/yolo/

Neural Networks in Dynamic Environments

Real-time object detection with neural networks[1]:

- Trained on cats, dogs, and cars.
- Their food is frequently disappearing.

2

[1] https://pjreddie.com/darknet/yolo/

[2] https://rpubs.com/dgolicher/yolo

Open Problems □

Verification □ Learning

□ Scalability:

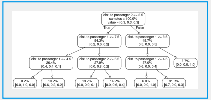
- Dimensionality of the input for controllers.
- Control synthesis for POMDPs.

Verification Learning

- Scalability:
  - Dimensionality of the input for controllers.
  - Control synthesis for POMDPs.
- Run-time performance vs. guarantees:
  - Real-time performance in self driving.
  - Assumptions about open world in control synthesis.

Verification □ Learning

- Scalability:
  - Dimensionality of the input for controllers.
  - Control synthesis for POMDPs.
- Run-time performance vs. guarantees:
  - Real-time performance in self driving.
  - Assumptions about open world in control synthesis.



- How to make model checkers generate good-for-learning counterexamples?
- How to use statistical model checking for learned controllers?