Verification for Machine Learning, Autonomy, and Neural Networks Survey
by Weiming Xiang, Patrick Musau, Ayana A. Wild, Diego Manzanas Lopez, Nathaniel Hamilton, Xiaodong Yang, Joel Rosenfeld, Taylor T. Johnson

Eric Langlois
University of Toronto
Feb. 11, 2019
1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software
Table of Contents

1 Safe Monitoring and Control

2 Intelligent Control

3 Specification Inference and Learning
   • Automata Learning

4 Safe Reinforcement Learning

5 Verification of Neural Networks

6 Available Software
Stability in Control Theory

If the system is near an equilibrium, will it stay there?

**Lyapunov stable** For any radius $\epsilon$, if we start close enough ($<\delta$) to the equilibrium, then the system will stay close ($<\epsilon$) forever.

**Asymptotically stable** If we start close enough ($<\delta$) to the equilibrium then the system will converge to the equilibrium.
Table of Contents

1 Safe Monitoring and Control

2 Intelligent Control

3 Specification Inference and Learning
   - Automata Learning

4 Safe Reinforcement Learning

5 Verification of Neural Networks

6 Available Software
Tiny Neural Network Proportional-Integral-Derivative (PID) Controller

Learn weights online

---

Cited Work on Intelligent Control

- Neural Network Controllers
  - Multilayer Perceptron
  - Diagonal Recurrent
  - Radial Basis Function

- Model Predictive Control

- Online learning methods

- Often guarantee Lyapunov stability (under assumptions)

- Semi-globally uniformly ultimately bounded

- Stability while learning the optimal policy
Controller performs optimization over predicted future state


Table of Contents

1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software
Referenced results:

- Learn Signal Temporal Logic (STL) formulas from positive examples traces only
- Scalable framework to mine STL specifications from a closed-loop model of a system’s behaviour
- Boolean formula learning
- Modelling system uncertainty with Gaussian Processes (GP)
  - Apply reachability analysis: verify unsafe states avoided
Gaussian Process

Represents an uncertainty distribution over functions.
Hamilton-Jacobi Reachability

Can worse-case dynamics force the system into undesirable states?

Value Alignment

Ensure objectives of AI systems match human values.

Subsection 1

Automata Learning
Automata

**discrete**  A discrete set of states.

**continuous**  Continuous states.

**hybrid**  Finite discrete states with continuous variables

**timed**  Finite automaton with a set of real-valued clocks.
           Sub-class of hybrid automata.
Automata Learning

Learn the structure of an automaton by observing state traces (passive) or posing queries (active).

Referenced results:

- Limited learning of hybrid automata
- Efficient learning of 1-clock timed automata
- Theory of learning timed automata
- Deterministic real-time automaton learning

Application:

learnlib.de

LearnLib
An open framework for automata learning
Table of Contents

1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software
Section 4

Safe Reinforcement Learning
A tuple \((S, A, T, R, p_0, \gamma)\) where

- \(S\) is a set of states
- \(A\) is a set of actions
- \(T : S \times A \rightarrow S\) is a probabilistic transition function
- \(S \times A \times S \rightarrow \mathbb{R}\) is a reward function
- \(p_0\) specifies the initial state distribution
- \(\gamma \in [0, 1]\) is a discount factor

Objective: Find a policy \(\pi : S \times A \rightarrow \mathbb{R}\) that maximizes the cumulative discounted reward

\[
\sum_{t=0}^{\infty} \gamma^t R(s_t, r_t, s_{t+1})
\]
“Safe Reinforcement Learning can be defined as the process of learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes.”

Fundamental approaches:

1. Alter the optimization criteria
2. Modify the exploration process

---

Safe RL Research

- Safe RL developed based on Lyapunov stability verification
- **Felix Berkenkamp et al.** “Safe Model-based Reinforcement Learning with Stability Guarantees”. In: *NIPS*. 2017, pp. 908–919

- Reachability-base approach
  - Vulnerable to model inaccuracies
- Model environment with Gaussian Processes
Safe RL Research (continued)

- Validate actions with temporal logic.

- Probabilistic model checking to verify and repair learned policies.

- Combination of formal verification and runtime monitoring

- Training an intervention model from human oversight
1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software
Neural Network Verification Approaches

Validating even simple properties about their behaviour is NP-complete

**Reachability** Estimate the possible set of network outputs over some input distribution.
- Software: AI² safeai.ethz.ch
- Practical for large networks

**Sensitivity** Measure maximal output deviation for bounded input deviation.

**Falsification** Find adversarial examples.

**Adversarial Robustness** Be resilient against adversarial examples
- An attack-independent robustness metric.
- Conditions under which no adversarial examples exist
Table of Contents

1. Safe Monitoring and Control
2. Intelligent Control
3. Specification Inference and Learning
   - Automata Learning
4. Safe Reinforcement Learning
5. Verification of Neural Networks
6. Available Software
### Verification Tools

Reluplex, Sherlock, AnalyzeNN, AI², PLNN, Planet, NeVer, VeriDeep, DeepGo, $L_0$-TRE, SafeCV, Certified ReLU Robustness, NNAF, Convex Adversarial

### Testing Tools

DeepCover, DeepXplore, DeepConcolic
Questions?