Safe Reinforcement Learning via Shielding

Mohammed Alshiekh, Roderick Bloem, Ruediger Ehlers, Bettina y: Könighofer, Scott Niekum, Ufuk Topcu

Reproduced and Slides By: Carl Hildebrandt
October 20, 2019
Advances in machine learning have given rise to **autonomous systems**

One particular type of machine learning is **reinforcement learning** which is good at:

1. **Complicated Tasks**
   - https://towardsdatascience.com/planning-the-path-for-a-self-driving-car-on-a-highway-7134fddd8707
2. **Dynamic Environments**
3. **Safety**
Background

Reinforcement learning - discover policies that **maximize a reward**
Reinforcement learning - **does not guarantee safety**

Example:
Parallel Parking

Definition: An exploration process is called **safe** if no undesirable states are ever visited[14]

Two mostly isolated threads of work:

<table>
<thead>
<tr>
<th>Safety in reinforcement learning</th>
<th>Safety in formal methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Safety achieved by incorporating external knowledge[8][14]</td>
<td>• Receding horizon control which combines continuous control and discrete correctness guarantees[24]</td>
</tr>
<tr>
<td>• Using temporal logic to generate invariance properties[8][1]</td>
<td>• Simple safe controllers can be computed directly [9] and more complex ones can be computed using a combination of low level controllers[5]</td>
</tr>
<tr>
<td>• Using a teacher network [22][4]</td>
<td>• Shield synthesize to enforce safety properties on controller[3]</td>
</tr>
<tr>
<td>• Using a human teacher [15][19]</td>
<td>• …</td>
</tr>
</tbody>
</table>
**Target Task**

This paper sets out to generate a technique which *guarantees safety and correctness* of a learning agent.

They aim to do this using the *correctness guarantees provided by formal methods* and combine it with the *optimality provided by reinforcement learning*.

**Goals:**
1. Enforce safety properties in a traditional reinforcement learning setting.
2. Restrict the agent as little as possible by having minimum interference.
3. Create a system which clearly separates safety and optimality.

\[
\text{Correctness and Safety} \quad + \quad \text{Optimality} \quad = \quad \text{Safe Optimal Solution}
\]

\[
\text{Formal Methods} \quad \text{Reinforcement Learning} \quad \text{Safe Learning}
\]
Reinforcement learning leans by taking an action in an environment and getting a reward which it uses to update its policy:

The idea of this paper is to block actions which are unsafe, before they are enacted in the environment:
Proposed Solution

1. Generate a set of system specifications and an abstraction of the agents environment expressed as temporal logic.

2. Synthesize a reactive system (shield) which enforces the safety properties of the systems specifications.

3. Modify the learning loop (as shown on the right) by placing the shield in 1 of 2 places:
   1. Before the learning agent, thus removing any unsafe actions.
   2. After the learning agent, thus monitoring the selected actions and correcting them only if an unsafe action is chosen.

Fig. 1: Preemptive Shielding.

Fig. 2: Post-Posed Shielding.
Definition 1. Safe reinforcement learning is the process of learning an optimal policy while satisfying a temporal logic safety specification $\varphi^s$ during the learning and execution phases.

1. System specifications are given as temporal logic. For instance to state that an autonomous car must never run out of fuel: $G(\text{fuel}_\text{level} > 0)$

2. Convert the safety specifications into an automaton in which only safe states $F$ may be visited: $\varphi^s = \langle Q, q_0, \Sigma, \delta, F \rangle$

3. Convert the environment abstraction (often modeled as a Markov Decision Process) into an automaton: $\varphi^M = \langle Q, q_0, \Sigma, \delta, F \rangle$

4. Use reactive synthesis to enforce $\varphi^s$ by solving a safety game built from $\varphi^s$ and $\varphi^M$ which is won if the system only ever visits safe states $F$. 
They use four test environments to test their proposed solution to safety.

<table>
<thead>
<tr>
<th>Water Tank</th>
<th>Grid World</th>
<th>Self-Driving Car</th>
<th>Atari Seaquest™</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Water Tank" /></td>
<td><img src="image" alt="Grid World" /></td>
<td><img src="image" alt="Self-Driving Car" /></td>
<td><img src="image" alt="Atari Seaquest" /></td>
</tr>
<tr>
<td>Keep water warm while minimizing energy consumption. Water runs out of tank constantly. Cold water runs into take through controllable switch. Can not overflow or run empty.</td>
<td>Visit all colored regions in a given order while not crashing into walls or sitting on bombs for more than two consecutive steps.</td>
<td>Drive clockwise around the track. The car can only turn 7.5 degrees and moves 3 pixels each time step. Avoid crashing into walls.</td>
<td>Control a submarine to collect divers while avoiding obstacles. The submarine needs to surface before running out of oxygen.</td>
</tr>
</tbody>
</table>
The main experimental results are shown below.

**Water Tank**
- Shielded learning is shown in blue and green dashed lines.
- Unshielded learning is shown in red and gray solid lines.

**Grid World**
- Shielded learning is shown in the solid green line.
- Unshielded learning is shown in red dashed line and gray line. Gray line represents large negative penalty.

**Self-Driving Car**
- Red dashed line shows unshielded learning.
- Blue and black lines show shielded learning.

**Atari Seaquest™**
- Red dashed line shows unshielded learning.
- Blue solid line shows shielded learning.
Experimental Analysis

The main observation are:

<table>
<thead>
<tr>
<th>Water Tank</th>
<th>Grid World</th>
<th>Self-Driving Car</th>
<th>Atari Seaquest™</th>
</tr>
</thead>
</table>
| 1) Both unshielded and shielded learning reaches optimal policy.  
2) Shielded learning converges to optimal policy much faster. | 1) Both unshielded techniques get negative rewards - implying they violate a safety constraint during training.  
2) High negative reward does not convert to optimal policy | 1) Unshielded technique still crashes at the end of training.  
2) Shielded techniques learn more rapidly. | 1) Shielding did not change the performance of the learning agent.  
2) Safety properties never violated when shield was implemented. |
Why I selected this paper:

<table>
<thead>
<tr>
<th>Water Tank</th>
<th>Grid World</th>
<th>Self-Driving Car</th>
<th>Atari Seaquest™</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Person" /></td>
<td><img src="image2" alt="Person" /></td>
<td><img src="image3" alt="Person" /></td>
<td><img src="image4" alt="Person" /></td>
</tr>
</tbody>
</table>

- There are 4 scenarios (1 per group member):
  - The **entire approach** is applied to **each scenario**.
  - Reproducing 1 set of results requires understanding and implementing the entire paper.
- I selected to implement the self-driving car as this is the closest to my research.
Result Reproduction

Why replicating papers is beneficial:

• You get a **deeper understanding** of the approach.
• You confirm that the **results are correct**, and not tailored to look good.
• You benefit the developer by **checking their work**.

• I found a security issues on their Github repo and was able to submit a issue which was fixed and closed.

Here is a link to my review of the code:
https://github.com/hildebrandt-carl/SafeReinforcementLearning
Result Reproduction

My Results

Original Results
Conclusion and Future Work

- They provide a method for reinforcement learning under safety constraints expressed as temporal logic specifications.
- Their technique enforces safety constraints without changing the (often complex) inner workings of reinforcement learning algorithm.
- Demonstrate that the shielded agents perform at least as well as unshielded agents. However in most cases improve performance.
- They show that their shielded agents do not violate their safety constraints.
- Downside is that you need an approximate model which describes which actions are unsafe.
- Future work is not mentioned in this paper.
References are given the same number as in the paper for convenience.


