Safe Reinforcement Learning

Anand Balakrishnan

November 22, 2021

University of Southern California

Overview

Introduction

Introduction

- (Deep) Reinforcement Learning has become incredibly popular in recent years due to the ability of RL algorithms to generalize well in highly complex environments.
- Reinforcement learning involves training an agent by making it repeatedly experience its environment and learn to "solve" the environment.
- The agent does this by learning a policy that maximizes a reward function specified by the user.

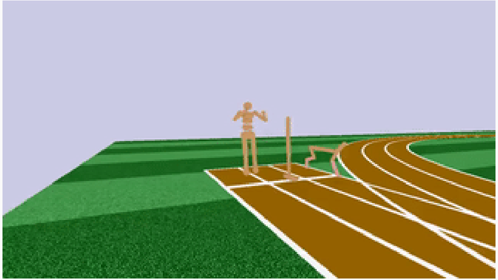

Figure 1: OpenAI Roboschool (`https://openai.com/blog/roboschool/`)

Problems with safety

- While these algorithms seem to perform really well with respect to the reward function, they may not do the exact behavior required of them.
- These “unexpected behavior” can translate to either unwanted maneuvers by the agent, or lead to the agent violating some safety condition.

Figure 2: “Concrete Problems in AI Safety”, Amodei et al.

- These problems are usually due to poorly-defined reward functions, or poorly explored state spaces.
    - “What don’t we know about the agent behavior?”
    - “What don’t we know about the environment?”
- There have been various approaches proposed to mitigate these issues, and these approaches fall under the broad category of “Safe Reinforcement Learning”.

Concrete problems

Concrete Problems

Broadly speaking, safety issues in reinforcement learning fall under the following problem categories:

- **Reward Hacking**: The agent finds a "cheat" for maximizing the rewards without actually doing the task.
- **Safe Exploration and Operation**: How can we explore/operate on our environment, without entering some bad/unsafe regions?

Note: There are technically more broad categories, but they may be less applicable to our class. To learn more, see Amodei et al. 2016; Leike et al. 2017.

Reward Hacking

The term *reward hacking* refers to the agent finding some configuration or set of configurations that maximize the reward without actually finishing the task.

An example of this is a “suicidal agent”: say we have a robot operating on a table. The goal of the robot is to complete some sequence of tasks, and until the robot does so, it gets a reward of -1 (so it’s a penalty) for every action it takes. More often than not, such a reward will force the robot to jump of the table and terminate the episodes with fewer penalized actions than actually have it attempt to complete the task.

Safe Exploration and Operation

This is a common problem posed in the context of robots that operate in mostly unknown environments, for example, the Mars rovers.

Say you want to design a controller that takes the Rover from its initial location to some final location without falling off a cliff, etc. Moreover, since it takes several minutes for any control signal to reach Mars from Earth, it may be beneficial for the robot to have some learning-based autonomy. How can the Rover safely take this journey without falling off a cliff or trying to cross some impossible-to-climb boulder?

Proposed Solutions

Reward Engineering and Shaping

- One option to fixing “reward hacking” is to prevent reward functions from being hackable!
- We can either *engineer* rewards to be this way, or we can massage/*re-shape* existing reward functions to be this way (Grześ 2017).

Inverse Reinforcement Learning

Goal: Learn a reward function from expert demonstrations.

Inverse Reinforcement Learning

Goal: Learn a reward function from expert demonstrations.

- Given a set of "expert" demonstrations, can we learn a reward function that allows a RL agent to closely mimic the experts?
- Several works have proposed to do this, including Abbeel and Ng 2004; Ramachandran and Amir 2007; Bıyık et al. 2021.

Temporal Logic Tasks

Goal: Incorporate formal verification techniques directly into the RL process, thereby mitigating the reward hacking problem directly.

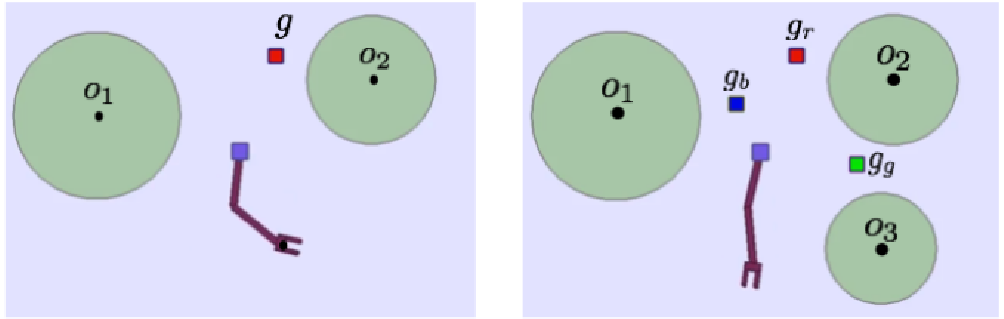

Figure 3: *Left:* Given the LTL specification $(\mathsf{F}\, g) \wedge \mathsf{G}(\neg o_1 \wedge \neg o_2)$, design a controller. *Right:* Given the LTL specification $(\mathsf{F}(g_r \wedge \mathsf{F}(g_g \wedge \mathsf{F}(g_b)))) \wedge \mathsf{G}(\neg o_1 \wedge \neg o_2 \wedge \neg o_3)$, design a controller. Image Credit: “Reinforcement Learning with Temporal Logic Rewards”, X. Li, Vasile, and C. Belta.

Temporal Logic Tasks

Goal: Incorporate formal verification techniques directly into the RL process, thereby mitigating the reward hacking problem directly.

- Aksaray et al. 2016; X. Li, Vasile, and C. Belta 2017; Xiao Li and Calin Belta 2016; Balakrishnan and Deshmukh 2019; Lavaei et al. 2020 propose methods to directly translate Linear Temporal Logic and Signal Temporal Logic specifications into reward functions for continuous state-space systems.
- Sadigh et al. 2014; Hasanbeig, Abate, and Kroening 2018; Hahn et al. 2019 propose methods to synthesize reward functions using automata that accept LTL specifications.
- Fu and Topcu 2014 presents a methods for model-based learning that incorporates LTL-accepting automata.

Safe Exploration and Operation

Goal: Learn a policy to do some task that is safe during exploration and/or during operation.

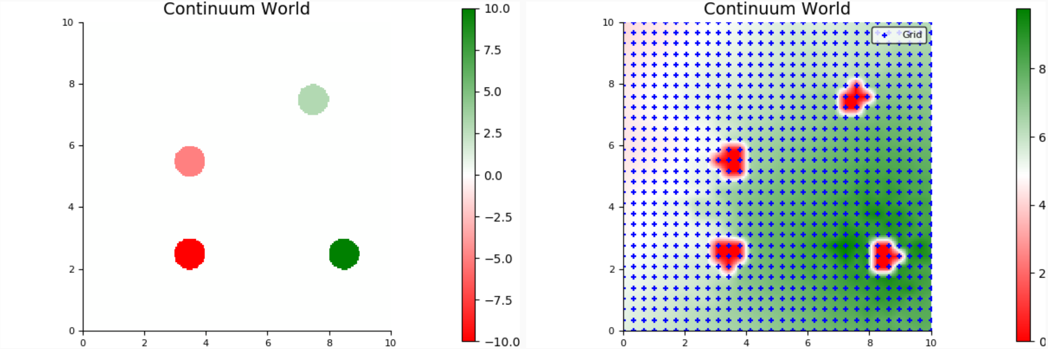

Figure 3: Before the environment is explored properly, we don't really know where the cliffs are until we fall there while trying to get to one of the goals. After the environment is explored, we can quantify "how safe" each position in the map is. Credit: Continuum World environment (`https://github.com/JuliaPOMDP/ContinuumWorld.jl`).

Safe Exploration and Operation

Goal: Learn a policy to do some task that is safe during exploration and/or during operation.

- Turchetta, Berkenkamp, and Krause 2016; Biyik et al. 2019; Roderick, Nagarajan, and Kolter 2020 propose methods to learn models of the environment efficiently without entering unsafe states.
- Wen and Topcu 2018; Ohnishi et al. 2018; Wang, Theodorou, and Egerstedt 2017 propose methods that use constraints (either as concrete constraints or using barrier certificates) to learn safe controllers for a system.
- In Alshiekh et al. 2018, the authors propose a *shielding*-based methods to prevent agents from taking unsafe/unwanted actions.

Conclusion

Conclusion

- Safe RL is an incredibly broad topic, where ideas from various fields meet to solve similar problems.
- In general, safety related issues are causes by some “unknowns” in either the learned behavior of the agent, or the structure of the environment the agent operates in.
- The way to mitigate these issues is by solving the “unknowns” in a principled manner.

References

References i

Amodei, Dario et al. (June 21, 2016). "Concrete Problems in AI Safety". In: arXiv: `1606.06565 [cs]`. URL: `http://arxiv.org/abs/1606.06565`.

Leike, Jan et al. (Nov. 27, 2017). "AI Safety Gridworlds". In: arXiv: `1711.09883 [cs]`. URL: `http://arxiv.org/abs/1711.09883`.

Grześ, Marek (2017). "Reward Shaping in Episodic Reinforcement Learning". In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '17. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, pp. 565–573.

Abbeel, Pieter and Andrew Y. Ng (July 4, 2004). "Apprenticeship Learning via Inverse Reinforcement Learning". In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML '04. New York, NY, USA: Association for Computing Machinery, p. 1. ISBN: 978-1-58113-838-2. DOI: `10.1145/1015330.1015430`. URL: `http://doi.org/10.1145/1015330.1015430`.

Ramachandran, Deepak and Eyal Amir (2007). "Bayesian Inverse Reinforcement Learning.". In: *IJCAI*. Vol. 7, pp. 2586–2591.

Bıyık, Erdem et al. (Aug. 4, 2021). *Learning Reward Functions from Diverse Sources of Human Feedback: Optimally Integrating Demonstrations and Preferences*. arXiv: `2006.14091 [cs]`. URL: `http://arxiv.org/abs/2006.14091`.

References ii

Li, X., C. Vasile, and C. Belta (Sept. 2017). “Reinforcement Learning with Temporal Logic Rewards”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3834–3839. DOI: `10.1109/IROS.2017.8206234`.

Aksaray, D. et al. (Dec. 2016). “Q-Learning for Robust Satisfaction of Signal Temporal Logic Specifications”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 6565–6570. DOI: `10.1109/CDC.2016.7799279`.

Li, Xiao and Calin Belta (June 2016). “A Hierarchical Reinforcement Learning Method for Persistent Time-Sensitive Tasks”. In: *arXiv:1606.06355 [cs]*. arXiv: `1606.06355 [cs]`.

Balakrishnan, Anand and Jyotirmoy V. Deshmukh (Nov. 2019). “Structured Reward Shaping Using Signal Temporal Logic Specifications”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3481–3486. DOI: `10.1109/IROS40897.2019.8968254`.

Lavaei, Abolfazl et al. (Mar. 2, 2020). *Formal Controller Synthesis for Continuous-Space MDPs via Model-Free Reinforcement Learning*. arXiv: `2003.00712 [cs, eess]`. URL: `http://arxiv.org/abs/2003.00712`.

Sadigh, Dorsa et al. (Dec. 2014). “A Learning Based Approach to Control Synthesis of Markov Decision Processes for Linear Temporal Logic Specifications”. In: *53rd IEEE Conference on Decision and Control*. 53rd IEEE Conference on Decision and Control, pp. 1091–1096. DOI: `10.1109/CDC.2014.7039527`.

Hasanbeig, Mohammadhosein, Alessandro Abate, and Daniel Kroening (Jan. 2018). “Logically-Constrained Reinforcement Learning”. In: *arXiv:1801.08099 [cs]*. arXiv: `1801.08099 [cs]`.

References iii

Hahn, Ernst Moritz et al. (2019). "Omega-Regular Objectives in Model-Free Reinforcement Learning". In: *Tools and Algorithms for the Construction and Analysis of Systems*. Ed. by Tomáš Vojnar and Lijun Zhang. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 395–412. ISBN: 978-3-030-17462-0. DOI: `10.1007/978-3-030-17462-0\\_27`.

Fu, Jie and Ufuk Topcu (July 12, 2014). "Probably Approximately Correct MDP Learning and Control With Temporal Logic Constraints". In: Robotics: Science and Systems X. Vol. 10. ISBN: 978-0-9923747-0-9. URL: `http://www.roboticsproceedings.org/rss10/p39.html`.

Turchetta, Matteo, Felix Berkenkamp, and Andreas Krause (2016). "Safe Exploration in Finite Markov Decision Processes with Gaussian Processes". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., pp. 4312–4320. URL: `http://papers.nips.cc/paper/6358-safe-exploration-in-finite-markov-decision-processes-with-gaussian-processes.pdf`.

Biyik, Erdem et al. (July 2019). "Efficient and Safe Exploration in Deterministic Markov Decision Processes with Unknown Transition Models". In: *2019 American Control Conference (ACC)*. 2019 American Control Conference (ACC), pp. 1792–1799. DOI: `10.23919/ACC.2019.8815276`.

Roderick, Melrose, Vaishnavh Nagarajan, and J. Zico Kolter (July 7, 2020). *Provably Safe PAC-MDP Exploration Using Analogies*. arXiv: `2007.03574 [cs, stat]`. URL: `http://arxiv.org/abs/2007.03574`.

Wen, Min and Ufuk Topcu (Dec. 3, 2018). “Constrained Cross-Entropy Method for Safe Reinforcement Learning”. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS’18. Red Hook, NY, USA: Curran Associates Inc., pp. 7461–7471.

Ohnishi, Motoya et al. (Jan. 2018). “Barrier-Certified Adaptive Reinforcement Learning with Applications to Brushbot Navigation”. In: *arXiv:1801.09627 [cs]*. arXiv: 1801.09627 [cs].

Wang, Li, Evangelos A. Theodorou, and Magnus Egerstedt (Oct. 15, 2017). *Safe Learning of Quadrotor Dynamics Using Barrier Certificates*. arXiv: 1710.05472 [cs]. URL: http://arxiv.org/abs/1710.05472.

Alshiekh, Mohammed et al. (Apr. 29, 2018). “Safe Reinforcement Learning via Shielding”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (1). ISSN: 2374-3468. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11797.