Work proposal

Artur Magalhães R. dos Santos

May 19, 2021

Proposal

This work consists in using Offline Reinforcement Learning techniques to improve the sample efficiency of reinforcement learning agents that operate in a simplified self-driving simulator known as *Duckietown Gym*6 environment. The simulator approximates the type of self-driving environments of small sized robots "driving" through streets of a mock city that are used in the annual AI Driving Challenge competition. It thus presents the main challenges of building self-driving agents in a controlled setting.

Reinforcement Learning, as defined by Sutton & Barto 1, is "*learning what to do—how to map situations to actions—so as to maximize a numerical reward signal*". The framework consists of an agent interacting with an environment and receiving some kind of reward. The agent must discover and learn about the environment on its own, choosing between the available actions. Basically, as described before, the RL framework contains: an environment, an reward, actions and observations.

Generally, the environment is where the agent act upon. The agent, with its set of available actions - which may vary depending on the problem been approached - decides which one to choose, given what is has learned from the those interactions. It collects observations, that describe what the environment looks like after taking an action, which them are used on the learning process. The reward is one of the most important parts of the learning loop, due to its effect on the agent behavior. It is the base metric to inform the agent if it has performed well or badly given it took an specific action on the current state. An environment modeled with rewards that don't match the expected agent behavior may lead to the agent having a poor performance.

This research area has grown over time, specially after the emergence of deep learning techniques, which use deep neural networks for function approximation. Many works have established remarkable results in tasks which humans were thought to be unbeatable, such as Go[2], Starcraft[3], and many more. Besides its applications related to gaming, reinforcement learning is used in robotics, recommendation systems, biology and more.

Offline Reinforcement Learning, also known as Batch Reinforcement Learning, is a data-driven approach to the reinforcement learning problem setting. Differently from the standard RL framework, Offline RL makes use of a fixed data set containing observations of states, actions, and rewards, derived from previous and random interactions, human demonstrations, or demonstrations from related tasks. An agent interacts with this data to gain knowledge on the task at hand.

One of the main motives behind Offline RL is that in many real world problems, collecting new data can be cumbersome, both in terms of cost and required labor. Some particular tasks require human labor to produce data, and have limitations on experimentation time - RL models may take a long time on training, and only have a acceptable behavior after hundreds of thousands of learning steps. Also, some settings may require restrictions on safety - which kind of that can be used to create a model - and Offline RL is notably interesting for approaching this scenarios. Other research areas, such as NLP and an computer vision already rely on models that were previously trained on huge data sets, and are refined for solving particular tasks. In that regard, Offline RL relates to these other research areas, and bring those ideas of using trained models and only refining them for different tasks into RL.

The main challenges posed by Offline RL are mainly on how to use the data properly, that is, learn from the data set but being able to generalize the behavior outside it[4]. In most cases, the data set contains data that may be not reliable (partially complete interactions, sub-optimal behavior), posing a more challenging situation to the learning problem. Furthermore, most Offline RL algorithms don't obtain much improvement when learning from an offline data set and fine-tuning with an online environment. Problems related to fitting a behavior model, which is modeled after the offline data, may occur due to the algorithms being over conservative - meaning they don't want risk over trying visiting new states. When the behavior model is inaccurate, it becomes conservative over new data and it isn't able to improve much over online interactions, leading to poor improvements and slow learning.

Some established works [5] bridge this gap between Offline RL - which tends to be over conservative but has advantages on data usage and collection - and RL - which is able to learn on a online setting, balancing exploration and exploitation on the interactions it takes. These 2 approaches combined may be used to accelerate the learning process, getting the advantages of both of them.

On our work, we will use Offline RL techniques to speed learning on a self-driving simulator, Duckietown Gym[6]. Duckietown Gym is an environment, compatible with Open AI Gym framework, which simulates small sized robots "driving" along a city. It contains streets, intersections, houses, and is customizable: you may add more cars into the city, traffic lights, changing the landscape of the streets, and more. This additions pose more challenges to an agent learning process, and its what we incrementally plan to do.

The main steps of this work are:

1. studying and researching about techniques and methods on online and offline reinforcement learning
2. implement the code related to these methods and perform experiments on them
3. write the monograph and present the results

From these 3 main steps, we may detailed them as:

- From 1, deepen the studies in reinforcement learning and its state of the art methods, such as Deep Deterministic Policy Gradient (DDPG), Soft Actor-Critic methods. Also, get in touch with Offline RL research area, through papers and tutorials available. UC Berkeley provides extensive materials on this domain, and we plan to use them. Furthermore, learn about Offline RL algorithms, such as Conservative Q-Learning (CQL). **Deadline: July 2021**
- From 2, to implement the algorithms, we need to study the environment. First step is defining the observation an agent gets: the observation may consist of different parameters, and we need to evaluate which are important for the agent to learn properly. Duckietown environment allows you to use positions (x, y), angular velocity, the car's front view, and many more. This is a important step that will impact on how well the agent performs. Also, we plan to implement an online reinforcement learning agent with DDPG at this step. **Deadline: August 2021**.

  Also, we need to implement some baseline RL algorithms on this environment for performance and speed comparisons: to evaluate our work performance and time needed for learning, we need to compare between online RL and offline RL algorithms. This will provide evidence that speed and quality were increased with our approach on Duckietown environment. **Deadline: September/October 2021**.

  Another important step is to collect the fixed data set from interactions: Offline RL requires that we use a fixed data set for training, so collecting interactions - random, human and with some kind of behavior - is what will enable us to implement these methods. We also plan to implement CQL at this step. **Deadline: October 2021**.

  This is step where we implement the proposed speed up technique: the actual established method of speeding up learning is probably the final step, being the result of our work.
- Final step is writing the monograph, along with presenting the results. It is possible that on this validation we decide to add more experiments or some minor additions. **Deadline: November 2021**.

References

1. Richard S. Sutton and Andrew G. Barto. 2018. "Reinforcement Learning: An Introduction". A Bradford Book, Cambridge, MA, USA.

2. Silver, David et al. 2016. "Mastering the game of Go with deep neural networks and tree search". Nature. 529. 484-489. 10.1038/nature16961.

3. Vinyals, O., Babuschkin, I., Czarnecki, W.M. et al. 2019. "AlphaStar: Grandmaster level in StarCraft II using multi-agent reinforcement learning". Nature. 575. 350–354. 10.1038/s41586-019-1724-z.

4. Siegel, Noah Y., et al. 2020. "Keep doing what worked: Behavioral modelling priors for offline reinforcement learning." arXiv preprint arXiv:2002.08396.

5. Nair, Ashvin, et al. 2020. "Accelerating online reinforcement learning with offline datasets." arXiv preprint arXiv:2006.09359 (2020).

6. Chevalier-Boisvert et al. 2018. Duckietown Environments for OpenAI Gym. GitHub repository.