Reinforcement Learning with Hard Constraints for Autonomous Driving: CS234 project milestone

Philippe Weingertner, Vaishali G Kulkarni
`pweinger@stanford.edu, vaishali@stanford.edu`

Project Mentor: Ramtin Keramati

1 Introduction

Reinforcement Learning (RL) has demonstrated its capability to learn efficient strategies on many different and complex tasks. In particular, in games like chess and go, the best human players have lost against RL algorithms (Silver et al. [8]). There is a growing traction for applying such RL algorithms to complex robotics tasks like Autonomous Driving. Nevertheless with Autonomous Driving we are dealing with additional challenges. We are in a partially observable environment where enforcing safety is of paramount importance. As a consequence, considering safety via a reward and the optimization of a statistical criteria is not sufficient. Hard Constraints have to be enforced all the time. We propose to study how the RL optimization criteria can be modified to deal with hard constraints; how algorithms like DQN could be modified to cope with such hard constraints and more generally how an RL agent could be integrated in a Decision Making module for Autonomous Driving to provide efficient and scalable strategies while still providing safety guarantees. So we propose to address the following problem formulation:

$$\underset{\theta}{max}\, \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi_\theta(s_t))]$$

$$\text{s.t. } lower\_bound(C_i(s_t, a_t)) \geq \text{Margin}_i \; \forall i \in [\![1, K]\!]$$

where the expectation corresponds to the statistical RL objective subject to a set of safety constraints.

We tackle the problem of safe control in physical systems where certain quantities have to be kept constrained. For an autonomous vehicle we must always maintain its distance from obstacles above some margin. But in fact the real state of the world is only partially observable and the Driving Models of surrounding cars are not known exactly: so we are dealing with uncertainty and our constraints are actually a set of random variables C_i which we want to lower bound. Note that in most of the references the constraints are only considered in expectation via a constraint of type $J_{C_i}^{\pi} = E_{\pi}\left[C_i(s,a)\right]$ whereas here we are interested in enforcing stronger constraints.

Project Milestone Updates:

- We will use a single hard constraint which is related to the min Time To Collision that is computed between the ego vehicle and the predicted trajectories of the surrounding vehicles. This min TTC shall be greater or equal than a threshold margin.

2 Background/Related Work

In Mirchevska et al. [7] a DQN network is used for tactical decision making in an autonomous driving pipeline but the DQN algorithm itself is not modified to handle hard constraints and the safety is guaranteed by checking the output of the RL algorithm. Our objective here, in contrast, would be to have an RL algorithm that is directly dealing with hard constraints to avoid frequent and sub-optimal actions masking. A review of the different safe RL techniques has been done in García and Fernández [5]. Some techniques mainly deal with soft constraints by either reshaping the reward or trying to minimize the variance related to the risk of making unsafe decisions, while other try to handle hard constraints. Garcia et al. have analyzed and categorized safe RL techniques in two families of approaches: one consists in modifying the exploration process while the other consists in modifying the optimality criterion. In Leurent et al. [6] the RL objective is replaced by a surrogate objective which captures hard constraints and handles model uncertainty by defining a lower bound of the expectation objective. In Achiam et al. [1] constrained policy optimization is solved with a modified trust-region policy gradient. The algorithm's update rule projects the policy to a safe feasibility set in each iteration. But the policy is kept within constraints only in expectation. In Dalal et al. [4] they directly add to the policy a safety layer that analytically solves an action correction formulation per each state. This safety layer is learned beforehand but is approximated by a first order linear approximator. In Tessler, Mankowitz, and Mannor [9] and in Bohez et al. [2] the proposed approaches are completely in line with our objective here: modifying the RL objective such that it deals directly with hard constraints. But there is no closed form solution for such a problem and a Lagrangian relaxation technique is used for solving the constrained optimization problem. Given a Constrained Markov Decision Process (CMDP), the unconstrained problem is transformed to $\underset{\lambda \geq 0}{min} \, \underset{\theta}{max} \; L(\lambda, \theta) = \underset{\lambda \geq 0}{min} \, \underset{\theta}{max} \; [J_R^{\pi_\theta} - \lambda(J_C^{\pi_\theta} - \alpha)]$ where L is the Lagrangian and λ the Lagrange multiplier (a penalty coefficient). We propose to study how such techniques could be applied to the Decision Making process of an Autonomous Driving pipeline and we will benchmark different RL algorithms, modified to cope with hard constraints, in an Anti Collision Tests setting.

3 Approach

A DQN network will be used as a baseline (we may change later to Policy Gradients or Actor Critic. This is a topic for further refinement). We will consider 3 type of modifications. From the most simple, conceptually, to the most complex, we will:

1. Propose a post DQN safety check. While the DQN network will compute $Q(s, a_i)$ values for every possible actions, we want to exclude the actions that are unsafe, before deciding what action to take (before taking the $argmax_{a_i} \; Q(s, a_i)$). This type of approach is used in a paper from BMW Group by Mirchevska et al. [7].
2. Modify the DQN training algorithm and especially the exploration process so that only safe actions are explored. Similar to Bouton et al. [3], the idea is to derive an exploration strategy that constrains the agent to choose among actions that satisfy safety criteria. Hence the search space of policies is restricted to a "safe" or safer subspace of policies.
3. Replace the RL objective $\underset{\theta}{max} \, \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi_\theta(s_t))]$ by an objective taking into account hard constraints

   $$\underset{\theta}{max} \, \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi_\theta(s_t))]$$

   $$\text{s.t. } lower\_bound(C_i(s_t, a_t)) \geq \text{Margin}_i \; \forall i \in [\![1, K]\!]$$

   and study how RL algorithms like DQN should be modified to account for this new objective. In a recent paper from DeepMind from Bohez et al. [2], this type of approach is applied to a realistic, energy-optimized robotic locomotion task, using the Minitaur quadruped developed by Ghost Robotics. But the constraints were considered only in expectation.

Project Milestone Updates:

- We will focus on step 3
- The surrounding vehicles will move according to an IDM driving model. This driving model depends on 5 parameters. The ego vehicle does not know (precisely) the parameters used by the surrounding vehicles. This is the main source of uncertainty.
- In order to define the hard constraint, in a real use case (not simulated) the ego vehicle shall estimate the Driving Model parameters of every surrounding vehicle with some probability

distribution to account for uncertainty. Here in simulation and in the context of this project, we will use some provided range of IDM parameters values: so the IDM model estimation is emulated. Then we will consider worst case scenarios leading to the smallest possible Time To Collision and assess how robust we are againt different level of uncertainties.

- So the hard constraint will be min TTC ≥ 10 where the Time To Collision is computed based on:
  - $acceleration_{ego-vehicle} = \pi_\theta(s_t)$ which depends on the parameters of the Neural Network
  - Predicted trajectories of surrounding vehicles which depend on a set of 5 IDM parameters per surrounding vehicle. We will simulate the fact that these parameters are estimated by the ego vehicle by providing range of values to the ego vehicle in the simulation framework.
  - So the hard constraint will involve the parameters θ of $\pi_\theta(s_t)$ such that min Time To Collision is above some threshold.
- In terms of implementing the hard constraint with Tensorflow we will most probably leverage on the following code tensorflow_constrained_optimization from Google research.

4 Experimental Results

Project Milestone Updates:

- A github directory has been created for the project: CS234_Project
- The legacy simulator code has been ported from Julia code (as used in CS238 project) to Python (for use in CS234 Project)
- The simulator code has been upgraded so that the surrounding vehicles move according to an IDM driving model. In previous CS238 project the surrounding vehicles were moving according to a very simple Constant Velocity model.

4.1 Simulator

We are upgrading the Anti Collision Tests environment, ACT, developed for a previous CS238 Stanford project. A vehicle has to drive from a point A to a point B as fast as possible, while avoiding other vehicles that are crossing its path and trying to minimize hard braking decisions. So it is a multi-objectives task where efficiency, comfort and safety objectives have to be optimized. While in the previous project we studied the applicability of POMDPs solvers for decision making in a context of sensors uncertainty, we will deal here with an even more challenging task: the uncertainty will be related to the other vehicles driving models. Initially other vehicles driving models were simple Constant Velocity models. Here we will use Intelligent Driver Models, IDM, depending on 5 parameters that will be unknown to the ego vehicle, and that will differ per vehicle. So it is a model-free setting: we do not know the model of the environment, the driving models of others, and we would like to learn to drive efficiently and safely in this context.

4.2 Evaluation Metrics

In order to measure success and benchmark different versions of the algorithms, we will use 3 metrics: a safety metric (percentage of collisions), an efficiency metric (time to goal), and a comfort metric (number of hard braking decisions or jerk). We want to enforce safety while not compromising too much efficiency or comfort: a safe AD vehicle that would use many regular hard braking decisions would not be acceptable and could even be dangerous for other vehicles.

5 Project Milestones Updates: Remaining Work / Next Steps

- Establish a baseline with RL Policy Gradient algorithm trained on ACT simulator framework. Provide evaluation metrics results.
- Meeting to be planned 5^{th} or 6^{th} of March with Vaishali, Ramtin, Philippe. Philippe who is located in France will be in California these days.

- Experiment with tensorflow_constrained_optimization from Google research on a simple hard constraint use case (e.g. such that the weights of the Neural Network are in a specific range)
- Implement code to compute the hard constraint min TTC $\geq$ some threshold. In a first step just log how we would deviate from our goal by not enforcing the hard constraint
- Combine Policy Gradient algorithm with the Hard Constraint code during Policy Gradient training
- Compare results with and without enforcing the hard constraint

References

[1] Joshua Achiam et al. “Constrained Policy Optimization”. In: (2017). URL: `http://arxiv.org/abs/1705.10528`.

[2] Steven Bohez et al. *Success at any cost: value constrained model-free continuous control*. 2019. URL: `https://openreview.net/forum?id=rJlJ-2CqtX`.

[3] Maxime Bouton et al. “Reinforcement learning with probabilistic guarantees for autonomous driving”. In: *Workshop on Safety Risk and Uncertainty in Reinforcement Learning, Conference on Uncertainty in Aritifical Intelligence (UAI)*. 2018. URL: `https://drive.google.com/open?id=1d2tl4f6GQgH1SERveTmPAR42bMVwHZAZ`.

[4] Gal Dalal et al. “Safe Exploration in Continuous Action Spaces”. In: (2018). URL: `http://arxiv.org/abs/1801.08757`.

[5] Javier García and Fernando Fernández. “A Comprehensive Survey on Safe Reinforcement Learning”. In: *Journal of Machine Learning Research* (2015). URL: `http://jmlr.org/papers/v16/garcia15a.html`.

[6] Edouard Leurent et al. “Approximate Robust Control of Uncertain Dynamical Systems”. In: 2018.

[7] Branka Mirchevska et al. “High-level Decision Making for Safe and Reasonable Autonomous Lane Changing using Reinforcement Learning”. In: 2018. DOI: `10.1109/ITSC.2018.8569448`.

[8] David Silver et al. “Mastering the game of Go without human knowledge”. In: *Nature* (2017). URL: `http://dx.doi.org/10.1038/nature24270`.

[9] Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. “Reward Constrained Policy Optimization”. In: abs/1805.11074 (2018). URL: `http://arxiv.org/abs/1805.11074`.