Deep Reinforcement Learning in Duckietown environment
Mély Megerősítéses Tanulás Duckietown környezetben

Katica, Bozsó Bence, Pap Bálint, Pelenczei

Abstract

This paper serves as the documentation of our group project for course VITMAV45 at Budapest University of Technology and Economics.

The Reinforcement Learning-based approaches are getting more and more general in literature, especially in the field of control tasks to reach a certain level of automation. This study deals with the problem of Lateral Control, performing the task of Lane Following, while providing the agent a fully empty track. That means only the standard curvature of the road had to be followed, no intersections or extra obstacles were present. The training was carried out in the Duckietown eco-system using only the official simulator part, the so-called Gym-Duckietown. In this paper, we propose a trained PPO agent for the previously described task with the use of a highly efficient reward concept that make the learning converge relatively fast.

A Megerősítéses Tanulás alapú megoldások egyre gyakoribbá váltak a szakirodalomban, legfőképpen a különböző vezérlési problémák területén annak érdekében, hogy az automatizáltság kívánt (lehetséges) szintje elérhetővé váljon. Ez a tanulmány a Laterális Vezérlés problémakörével foglalkozik egy szimpla Sávkövetés feladaton keresztül, amely közben az ágensnek teljesen üres utat biztosítunk. Ez azt jelenti, hogy az ágensnek csak az út görbületével kell foglalkoznia és azt követnie, miközben se kereszteződés, se egyéb akadály nem áll az útjába. A tanítási folyamat a Duckietown környezetében valósult meg, azon belül is csak annak egy részét, az úgynevezett Gym-Duckietown hivatalos szimulátort felhasználva. A tanulmány során egy PPO algoritmussal tanított ágens kerül bemutatásra az előzőleg ismertetett feladat megoldására egy hatékony rewarding koncepció felhasználásával, amely a tanulás konvergenciáját gyorsítja.

1 Introduction

Duckietown is a modular robotics and AI ecosystem. It provides hardware, software, simulation, datasets and learning materials to encourage both teaching and learning. Multiple different approaches exist for programming self-driving vehicles to solve different kinds of challenges, such as collision avoidance or simple lane following. In this paper we propose a both precise and safe approach to the Lane Following (LF) problem and finally evaluate the trained agent based on widely used general metrics.

Autonomous driving is a constantly evolving sector of today's AI applications. A varied set of possible solutions exist. During this project we focused on grasping one branch (RL) and deepening our knowledge in it. The fact that the results can be tested on hardware as well was appealing.

1.1 Related work

The application of Artificial Intelligence (AI) can be seen in many other autonomous driving related tasks[1][2][3][4], and outstanding results have already been achieved in solving this specific problem as well [5]. The latest trend to solve the LF problem is the use of Reinforcement Learning (RL) based algorithms, since they have shown impressive results in this realm [6][7] and also in several other applications [8]. Therefore for this specific problem, based on the results collected so far by the literature, we applied RL. In this chapter, we provide a brief overview on the different solutions that were proposed to deal with the LF problem.

These papers [9][10] demonstrate that RL can be successfully applied in this field. The reason and basis for the frequent use of RL in the LF problem is the extreme versatility provided by the framework in the area of sequential decision-making problems.

1.2 Contribution

This paper deals with the design of a lateral control agent that was trained for the Lane Following task successfully using the so-called Proximal Policy Optimization (PPO) algorithm. Finally the paper evaluates the trained agent based on widely used general metrics, which makes it relatively easy to compare with other solutions.

2 Environment

As mentioned before, and also as the official motto says "Duckietown is a modular robotics and AI ecosystem with tightly integrated components designed to provide joyful learning experiences." [11]. The Duckietown ecosystem has various advantages over other environments from which one is the fully configured simulator part: with the use of Open-AI Gym in the Gym-Duckietown one can train their agents on simulated roads, intersections etc. On the other hand, hardware is also provided in the form of so called Duckiebots.

2.1 State

State representation was provided by the Gym-Duckietown Simulator in the form of images from a forward-looking monocular camera. These 640×480 RGB images are preprocessed before being fed to the model. The processings are carried out by wrappers, whose purpose is to to add functionality to environments, like modify rewards or observations. Such wrappers were implemented by us as well. The **CropWrapper**'s purpose is to mitigate the processing of unnecessary data part such as the sky, thus providing the agent only useful information of the track. In addition a **ResizeWrapper** is used to lower the computational costs of the observation. Finally the **DtRewardWrapper** was responsible for improving the agent's decision making ability, which will be presented later in this document.

2.2 Action

The simulator's agents are virtual replicas of the differential wheeled Duckiebots, thus in order to control them, the velocities of both left and right wheels have to be specified. Since there is no need for moving backwards, these values can be mapped between 0 and 1.

2.3 Reward

The rewarding concept is one of the most important parts of the Reinforcement Learning framework, as the agents' performance is highly relied on it. The reason for this, is that it is the only (scalar) feedback from the environment which the agent receives and later uses to understand the inner dynamics, which is basically the core of the learning process. As a result, the right choice of rewarding concept can improve the success of the training and thus the performance of the trained agent (model).
In our rewarding concept we compute the reward according to the following equation:

$$R(s, d, \varphi) = 300 * R_{dist}(s) + R_{lanedist}(d) + R_{angle}(\varphi)$$

In this equation s stands for the distance the agent travelled in the right lane, d means the distance between the agent and the center of the right lane and φ is the angle between the tangent of the car and the center line of the lane. This reward function only rewards the agent if it moves forward in the right lane, otherwise it punishes with $R = -1$

3 Algorithm

The ever-increasing development of computing and hardware resources has enabled the explosive development of AI and Machine Learning (ML). In addition, Deep Learning (DL) has provided an opportunity for ML renewal: recent results show that the combination of DL and RL (DRL) can outperform human performance, even in some more complex tasks [13].

Reinforcement Learning can be interpreted as a process of tuning the Artificial Neural Network(s) (ANN). Unlike Supervised Learning (SL), which is also a type of ML, RL does not need a large amount of pre-labeled training data that is difficult or even impossible to collect. Training samples of RL are generated through simple interactions between the agent and the environment. During these interactions, the agent receives a scalar feedback from the environment, the so-called reward, which then the agent tries to maximize in order to achieve the optimal behavior. The role of the environment is therefore twofold in the process: on the one hand, it must produce the information from which the agent will be able to make a decision after processing it, this is called the state of the environment. On the other hand, it must qualify the change in the state of the environment in some way. In this case, the consequence of the agent's decision or the indicator that qualifies the decision itself is created by the environment in the form of a single scalar, this is the reward.

3.1 Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a policy based Reinforcement Learning algorithm, more specifically based on the principles of a vanilla policy gradient method. Because of the high rate of similarities, this algorithm also has an essential advantage over the (Q-)value based methods, just like the policy gradient: the convergence using this algorithm is always guaranteed to a certain level, namely to a local minimum.

In all the policy-based methods our goal is to tune the weights of the neural network to approximate a probability distribution over the available actions. In the case of PPO, the main objective is as follows:

$$L(\theta) = \mathbb{E}_t[min(r_t(\theta)A_t, clip(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t)]$$

In our project we opted for utilizing the ready to use implementation provided by Ray[14] framework.

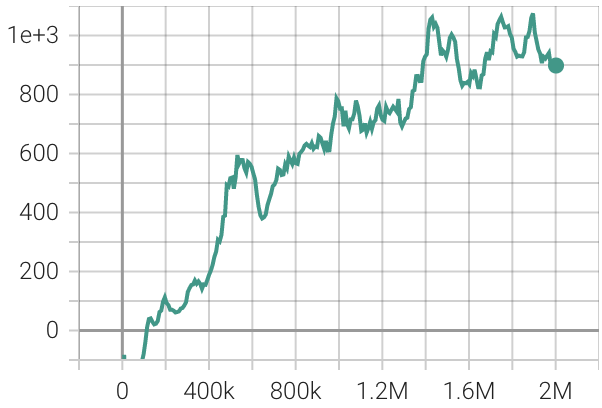

Figure 1: Average reward during 2M training timesteps

4 Conclusion

In this study, a Reinforcement Learning agent has been introduced that was trained using the Gym-Duckietown simulator with the so-called PPO algorithm. The task of the agent was to follow the lane while driving through the whole map. In order to achieve such results, an advanced reward formulation has been showed, which made the training and the overall performance of the agent stable.

4.1 Future plans

Based on the results acquired in our work so far, it seems reasonable to implement our solution in the hardware that is provided by the Duckietown framework. As doing so we would be able to process additional results and data, and thus also improve the agent's performance to an even higher level. In addition, we would like to continue this project by adding extra features to our agent, for instance to have the capability to circumnavigate other objects. Another option might be making it able to drive together with other Duckiebots, but in this case we would combine our solution with the use of Multi-Agent Reinforcement Learning (MARL). The MARL framework would make it possible for the Duckiebots to communicate with each other [15], which would unquestionably help to solve more complex problems like this.

References

[1]Sattel, T., Brandt, T. (2008). From robotics to automotive: Lane-keeping and collision avoidance based on elastic bands. Vehicle System Dynamics, 46(7), 597-619.

[2] Weiss, T., Schiele, B., Dietmayer, K. (2007, June). Robust driving path detection in urban and highway scenarios using a laser scanner and online occupancy grids. In 2007 IEEE Intelligent Vehicles Symposium (pp. 184-189). IEEE.

[3]Lai, Y. K., Ho, C. Y., Huang, Y. H., Huang, C. W., Kuo, Y. X., Chung, Y. C. (2018, October). Intelligent vehicle collision-avoidance system with deep learning. In 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS) (pp. 123-126). IEEE.

[4] Wei, J., Dolan, J. M., Litkouhi, B. (2010, June). A prediction-and cost function-based algorithm for robust autonomous freeway driving. In 2010 IEEE Intelligent Vehicles Symposium (pp. 512-517). IEEE.

[5] Taylor, C. J., Košecká, J., Blasi, R., Malik, J. (1999). A comparative study of vision-based lateral control strategies for autonomous highway driving. The International Journal of Robotics Research, 18(5), 442-453.

[6] Kővári, B., Hegedüs, F., Bécsi, T. (2020). Design of a Reinforcement Learning-Based Lane Keeping Planning Agent for Automated Vehicles. Applied Sciences, 10(20), 7171.

[7] Nageshrao, S., Tseng, H. E., Filev, D. (2019, October). Autonomous highway driving using deep reinforcement learning. In 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC) (pp. 2326-2331). IEEE

[8] Sallab, A. E., Abdou, M., Perot, E., Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. Electronic Imaging, 2017(19), 70-76.

[9]Smart, W. D., Kaelbling, L. P. (2002, May). Effective reinforcement learning for mobile robots. In Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292) (Vol. 4, pp. 3404-3410). IEEE.

[10] Kalapos, A., Gór, C., Moni, R., Harmati, I. (2020, October). Sim-to-real reinforcement learning applied to end-to-end vehicle control. In 2020 23rd International Symposium on Measurement and Control in Robotics (ISMCR) (pp. 1-6). IEEE.

[11] Almási, P., Moni, R., Gyires-Tóth, B. (2020). Robust reinforcement learning-based autonomous driving agent for simulation and real world. arXiv preprint arXiv:2009.11212.

[12] Duckietown official website: https://www.duckietown.org/

[13]Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... Hassabis, D. (2015). Human-level control through deep reinforcement learning. nature, 518(7540), 529-533. [14]https://www.ray.io

[15] Shalev-Shwartz, S., Shammah, S., Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295.