FRA-UNIted — Team Description 2020

Thomas Gabel, Philipp Klöppner, Yalcin Eren, Fabian Sommer, Steffen Breuer, Robert Litschel, Niklas Müller, Eicke Godehardt

Faculty of Computer Science and Engineering
Frankfurt University of Applied Sciences
60318 Frankfurt am Main, Germany
{tgabel|godehardt}@fb2.fra-uas.de, phil@philk.me,
{eren|fsommer|sbreuer|litschel|niklasmu}@stud.fra-uas.de

Abstract. The main focus of FRA-UNIted's effort in the RoboCup soccer simulation 2D domain is to develop and to apply machine learning techniques in complex domains. In particular, we are interested in applying reinforcement learning methods, where the training signal is only given in terms of success or failure. In this paper, we review some of our recent efforts taken during the past year, putting a special focus on a new Python-based framework for performing reinforcement learning experiments in the context of 2D soccer simulation.

1 Introduction

The soccer simulation 2D team FRA-UNIted is a continuation of the former Brainstormers project which has ceased to be active in 2010. The ancestor Brainstormers project was established in 1998 by Martin Riedmiller, starting off with a 2D team which had been led by the first author of this team description paper since 2005. Over the years, a number of sister teams emerged (e.g. the Tribots, Twobots, or Icebots) participating in real robot leagues. Our efforts in the RoboCup domain have been accompanied by the achievement of several successes such as multiple world champion and world vice champion titles as well as victories at numerous local tournaments.

While the real robot teams mentioned were closed down entirely, the 2D team has been in suspended mode since 2010 and was re-established in 2015 at the first author's new affiliation, Frankfurt University of Applied Sciences, reflecting this relocation with the team's new name FRA-UNIted.

As a continuation of our efforts in the ancestor project, the underlying and encouraging research goal of FRA-UNIted is to exploit artificial intelligence and machine learning techniques wherever possible. Particularly, the successful employment of reinforcement learning (RL, [15]) methods for various elements of FRA-UNIted's decision making modules — and their integration into the competition team — has been and is our main focus. Moreover, the extended use of the FRA-UNIted framework in the context of university teaching has moved into our special focus. So, we aim at employing the 2D soccer simulation domain as a fundament for teaching agent-based programming, foundations of multi-agents systems as well as applied machine learning algorithms.

In this team description paper, we refrain from presenting approaches and ideas we already explained in team description papers of the previous years [1]. Instead, we focus on recent changes and extensions to the team as well as on reporting partial results of work currently in progress. We start this team description paper, however, with a short general overview of the FRA-UNIted framework. Note that, to this end, there is some overlap with our older team description papers including those written in the context of our ancestor project (Brainstormers 2D, 2005–2010) which is why the interested reader is also referred to those publications, e.g. to [3, 11].

1.1 Design Principles

FRA-UNIted relies on the following basic principles:

- There are two main modules: the world module and decision making
- Input to the decision module is the approximate, complete world state as provided by the soccer simulation environment.
- The soccer environment is modeled as a Markovian Decision Process (MDP).
- Decision making is organized in complex and less complex behaviors where the more complex ones can easily utilize the less complex ones.
- A large part of the behaviors is learned by reinforcement learning methods.
- Modern AI methods are applied wherever possible and useful (e.g. particle filters are used for improved self localization).

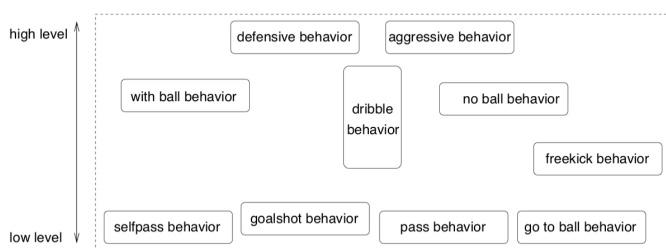

Fig. 1. The Behavior Architecture

1.2 The FRA-UNIted Agent

The decision-making process of the FRA-UNIted agent is inspired by behavior-based robot architectures. A set of more or less complex behaviors realize the agents' decision making as sketched in Figure 1. To a certain degree this architecture can be characterized as hierarchical, differing from more complex behaviors, such as “no ball behavior”, to very basic, skill-like ones, e.g. “pass behavior”. Nevertheless, there is no strict hierarchical sub-divisioning. Consequently, it is also possible for a low-level behavior to call a more abstract one. For instance, the behavior responsible for intercepting the ball may, under certain circumstances, decide that it is better to not intercept the ball, but to focus on more defensive tasks and, in doing so, call the “defensive behavior” and delegating responsibility for action choice to it.

2 Model-free Reinforcement Learning in RoboCup

From the very beginnings of the RoboCup initiative and, particularly, its 2D soccer simulation sub-league, many teams, including ours, used the domain of robotic soccer simulation as an environment for experiments in machine learning and reinforcement learning. During a soccer match, our agent uses a static algorithm to determine the appropriate behavior algorithm for the given situations (cf. Section 1.2). Many of these behavior algorithms, such as NeuroHassle [4], were trained using reinforcement learning. A more comprehensive review of our various efforts for applying reinforcement learning in robot soccer can be found in [13].

During the last year, we worked on the more general challenge of model-free end-to-end reinforcement learning in the context of RoboCup 2D soccer simulation. This means an agent learns making action decisions autonomously only by perceiving the soccer environment and getting an external reward signal. Similar research has already been done by Hausknecht and Stone in 2015 in the Half-Field Offense mode of RoboCup [6, 5] as well as by Gabel and Riedmiller in 2007 [12]. The main difference of our recent research compared to the referred works is the use of world-models for the agent’s decision making process.

In the following, we will present the results of our recent research on the use of deep reinforcement learning for model-free end-to-end learning in RoboCup soccer simulation.

2.1 Framework

Our agent utilizes the n++ framework [10] for artificial neural network-based machine learning which has proven itself over the past couple of decades. Unfortunately, it does not support modern deep learning algorithms, thus we decided to use the state-of-the-art machine learning framework TensorFlow [14, 2].

We assumed that an agent that does end-to-end learning would not need the very elaborate functionalities of our FRA-UNIted code base, which is why we decided to develop a new, light-weight agent for the purpose of this research project. Although the existing agent is written in C++, we thought implementing the new agent in Python would be a better fit. Not only could we benefit from the flexibility of the language, but many machine learning frameworks, including TensorFlow, work best with Python and are optimized for performance by using C++ and CUDA under the hood.

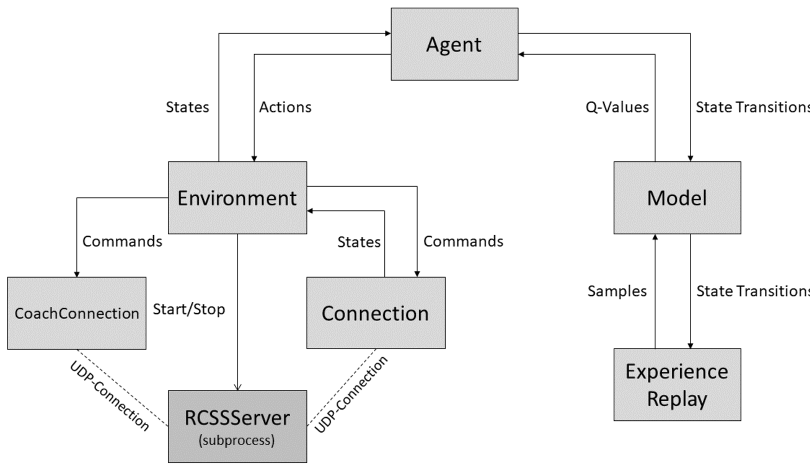

Fig. 2. Overview of the agent's classes and their relationship in the context of the learning process. Source: [7]

Figure 2 shows a high-level overview of the agent's software architecture.

The *Environment* encapsulates management of RoboCup simulation server instances and facilitates communication with the server. It exposes a simple action-perception-interface to the agent. The *Model* class is a placeholder for the agent's decision making model, in our case an advanced Q-model that reports action-values to the agent and optimizes itself with information it receives from the agent.

2.2 Learning

We chose Deep Q-learning (DQN) as our learning algorithm which was first introduced by Mnih et al. in 2013 [9] for solving Atari video games using deep neural networks for approximating the optimal action-value function.

Reinforcement learning algorithms such as DQN require the problem to be formulated as a Markov decision process (MDP), which is fairly intuitive for the domain of robotic soccer simulation. The agent's perception space becomes the MDP's state space and its available actions such as dashing, kicking and turning give the action space. State transitions are computed stochastically by the soccer simulation server. In summary, we treated the soccer simulation as a stochastical partially-observable Markov decision process, although the current experiments use the server's *fullstate*-mode, enabling full observability for agents.

To evaluate our learning framework with the DQN algorithm, we conducted several different experiments. In the following, we will present two experiments where a single agent was supposed to run to a fixed target position on the playing field.

In the first experiment, the agent had four discrete actions, dashing forwards, backwards and sidewards. It perceived its own position on the field and was rewarded for approaching the target position. The second experiment had the same goal, but the agent was only allowed to dash forward and turn its body, thus making temporal-difference learning mandatory, as turning itself would not earn the agent a positive reward. Figure 3 shows the learning curves of these experiments.

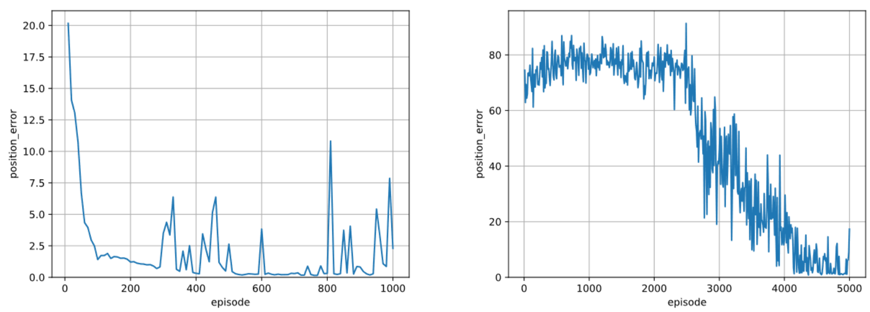

Fig. 3. Learning curves of the first and second experiment, displaying the average position error in each episode's terminal state, smoothed over ten states. Source: [7]

The graphs show that the agent reaches the goal faster with omnidirectional dashes. When using only forward dashes with turns, the agent takes a longer time learning but eventually also solves this task. This difference might be due to the delayed reward signal in the latter setting.

2.3 Next Steps

Our recent work serves as a proof-of-concept for DQN in the context of RoboCup without relying on a model of the simulation. Also, the performance of the Python agent proofed to be excellent for the given task [7] as it allowed us to increase the learning speed approximately by a factor of 30.

In the future, we would like to extend this approach to multi-agent problems in the domain of robotic soccer simulation. Furthermore, we would like to evaluate different learning algorithms for the same tasks, such as DDPG [8] for parameterized actions and dueling DQN [16] as a more advanced variant of deep Q-learning.

In conclusion, our work shows that the domain of RoboCup 2D soccer simulation still offers a significant challenge in reinforcement learning research and we are keen to advance our efforts in this area of research.

3 Continuous Integration Environment

In order to ensure a continuous improvement of our team, we had the main idea to implement a continuous integration (CI) environment. In recent years, improvements made by the team were tested by long game series, which had to be started manually, since judging changes or improvements needs averages over 1000 games or more. Thereby, only the final result of the game was utilized for giving a judgement about recent changes. Now, the idea is to automate these test series and the evaluation of the games and, in addition to the final results of the games, to generate and analyze further statistics about the progress of the game.

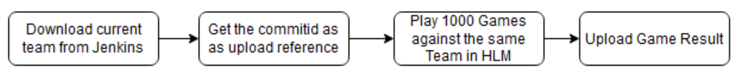

Fig. 4. Continuous Integration Workflow

Implementation The process of automated test games can be divided into three parts (cf. Figure 4). In the first step, after a new commit into the team Git repository, the current state of the repository is pulled and compiled using Jenkins. Jenkins was chosen because it is a widely used CI tool. The next step is to start test games using HLM[1], a Ruby-based tool for creating and running entire tournaments for the RoboCup 2D soccer simulation[2]. The test games run in parallel on 20 computers, each computer can play up to approximately 50 games a night, so a total of 1000 games in each night. In the last step, a statistic is created for each game, which, on the one hand, provides the final result of the game and, on the other hand, also determines statistics like the number of cards, free kicks, etc. of each team. In addition, statistics such as the percentage of ball possession of each team are also created. The statistics gained from each game is then uploaded back to a server with the current commit identifier. The commit identifier helps to keep different versions of the teams apart and to make them comparable.

First Results The main advantage of this implementation is that every night a large amount of test games are started in an automatized manner and the results are evaluated automatically, as well. In addition, you have a central location to display statistics that are more meaningful than just who won and who lost. We hope to utilize this tool especially during championship tournaments in order to evaluate late changes made on-site in an off-site setting.

[1] **H**ech **L**eague **M**anager by Andreas Hechenblaickner from the Austrian 2D Soccer Simulation team KickOffTUG (2004–2010)

[2] As a side note, the HLM is the tool which has been used for managing and running all RoboCup World Championships tournaments since 2009.

4 Conclusion

In this team description paper we have outlined the characteristics of the FRA-UNIted team participating in RoboCup's 2D Soccer Simulation League. We have stressed that this team is a continuation of the former Brainstormers project, pursuing similar and extended goals in research and development as well as for teaching purposes. Specifically, we have put emphasis on our most recent research activities and practical implementation of our results.

References

1. Gabel, T., Breuer, S., Sommer, F., Godehardt, E.: FRA-UNIted - Team Description 2019. In: RoboCup 2019: Robot Soccer World Cup XXIII, LNCS (CD Supplement). Springer, Sydney, Australia (2019)
2. Gabel, T., Klöppner, P., Godehardt, E., Tharwat, A.: Communication in Soccer Simulation: On the Use of Wiretapping Opponent Teams. In: RoboCup 2018: Robot Soccer World Cup XXII, LNCS. pp. 3–15. Springer, Montreal, Canada (2018)
3. Gabel, T., Riedmiller, M.: Brainstormers 2D - Team Description 2009. In: L. Iocchi, H. Matsubara, A. Weitzenfeld, C. Zhou, editors, RoboCup 2009: Robot Soccer World Cup XII, LNCS (CD Supplement). Springer, Graz, Austria (2009)
4. Gabel, T., Riedmiller, M., Trost, F.: A Case Study on Improving Defense Behavior in Soccer Simulation 2D: The NeuroHassle Approach. In: L. Iocchi, H. Matsubara, A. Weitzenfeld, C. Zhou, editors, RoboCup 2008: Robot Soccer World Cup XII, LNCS. pp. 61–72. Springer, Suzhou, China (2008)
5. Hausknecht, M., Mupparaju, P., Subramanian, S., Kalyanakrishnan, S., Stone, P.: Half Field Offense: An Environment for Multiagent Learning and Ad Hoc Teamwork. In: AAMAS Adaptive Learning Agents (ALA) Workshop (2016)
6. Hausknecht, M., Stone, P.: Deep Reinforcement Learning in Parameterized Action Space. arXiv (2015), https://arxiv.org/abs/1511.04143
7. Klöppner, P.: Untersuchungen zum Ende-zu-Ende-Lernen im simulierten Roboterfußball mit tiefem optimierendem Lernen (2019), Frankfurt University of Applied Sciences
8. Lillicrap, T., Hunt, J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous Control with Deep Reinforcement Learning (2015), arXiv, https://arxiv.org/abs/1509.02971
9. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing Atari with Deep Reinforcement Learning. arXiv (2013), https://arxiv.org/abs/1312.5602
10. Riedmiller, M., Braun, H.: A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: Proceedings of the IEEE International Conference on Neural Networks (ICNN). pp. 586–591. San Francisco, USA (1993)
11. Riedmiller, M., Gabel, T.: Brainstormers 2D - Team Description 2007. In: U. Visser, F. Ribeiro, T. Ohashi and F. Dellaert, editors, RoboCup 2007: Robot Soccer World Cup XI, LNCS (CD Supplement). Springer, Atlanta, USA (2007)
12. Riedmiller, M., Gabel, T.: On Experiences in a Complex and Competitive Gaming Domain: Reinforcement Learning Meets RoboCup. In: Proceedings of the 3rd IEEE Symposium on Computational Intelligence and Games (CIG 2007). pp. 68–75. IEEE Press, Honolulu, USA (2007)

13. Riedmiller, M., Gabel, T., Hafner, R., Lange, S.: Reinforcement Learning for Robot Soccer. Autonomous Robots 27(1), 55–73 (2009)
14. Shi, S., Wang, Q., Xu, P., Chu, X.: Benchmarking State-of-the-Art Deep Learning Software Tools. In: Proceedings of the 7th International Conference on Cloud Computing and Big Data (CCBD). pp. 99–104 (2016)
15. Sutton, R.S., Barto, A.G.: Reinforcement Learning. An Introduction. MIT Press/A Bradford Book, Cambridge, USA (1998)
16. Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., Freitas, N.: Dueling Network Architectures for Deep Reinforcement Learning (2015), https://arxiv.org/abs/1511.06581