How to use Imagination-Augmented Agents for Deep Reinforcement Learning

Angela Denninger
angela.denninger@tum.de

Florian Klemt
florian.klemt@tum.de

Felix Schober
felix.schober@tum.de

Max-Philipp Schrader
max.schrader@manageandmore.de

Project Proposal

1. Introduction

During this years NIPS 2017 conference the paper *Imagination-Augmented Agents for Deep Reinforcement Learning* [7] was presented by a team of researchers at DeepMind. In the paper a novel architecture for deep reinforcement learning, called Imagination-Augmented Agents (I2As), was proposed. The I2A architecture has shown improvements on performance and robustness as well as data efficiency compared to other approaches

Within the scope of our project we want to develop an implementation of I2As and compare it with a simple Reinforcement Learning algorithm also implemented by out team. We were not able to find any implementation of the I2A architecture. Therefore, our goal is to examine the I2A architecture and provide reusable code to the public.

1.1. Related Works

There have been a many publications and implementations of Reinforcement Learning for Atari games based only on pixel input. The first and most known work has been published by DeepMind in 2013 [5]. Following this approach several researches published work in this area of Reinforcement Learning [2, 6, 8, 3, 9]. One of the most recent publications is our input paper published during this years NIPS conference [7], which introduced a novel deep reinforcement learning architecture. Within the paper the researchers use the standard free-model presented by [4] as their baseline.

2. Dataset

For developing and benchmarking our Deep Reinforcement implementation we will be using traditional Atari Games. Therefore, we will work with the Atari games environment provided by OpenAI Gym [1]. The OpenAI Gym allows us to get the current state of the game, including the display and a reward system. In addition, we can perform actions within the environment to change the state. The input for our network will be the current state of the environment whereas the output will be the action to be performed next.

- OpenAI Gym [1]
- **Input:** Environment State of the Atari game, where the main input is the pixel map.
- **Output:** Next action to be performed in the environment

3. Methodology

First, we will develop a simple reinforcement learning model (RLM), which we will later use as a baseline to compare it against our implementation of the I2A architecture. In the next step we will implement a model based on the I2A architecture.

One of the more difficult parts for us during the planing process was the task distribution. We thought about the following work packages:

1. OpenAI Gym Setup/Training
2. baseline implementation
3. I2A architecture implementation
4. Analysis of results and presentation

Where 3. would be done by two people. Overall we are not sure if our planing is perfect and we would love to have your feedback on this one.

During the project we will follow these steps

- Implementation of a classical RLM playing Atari games
- Implementation of the I2A architecture
- Comparing our I2A implementation to the RLM baseline and the results published at [7] on the game Sokoban
- Learning to play a subset of all 2600 Atari games

4. Outcome

At the end of our project our goal is to have working implementation of the I2A architecture, which is able to outperform our baseline implementation. As an additional goal we will try to get as close as possible to the published results in the paper [7]. Due to the novelty of the paper we can not guaranty to have the I2A architecture implemented. However, we are positive to reach our goals and will do our best during the project.

References

[1] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016. 1

[2] X. Guo, S. Singh, H. Lee, R. L. Lewis, and X. Wang. Deep learning for real-time atari game play using offline monte-carlo tree search planning. In *Advances in neural information processing systems*, pages 3338–3346, 2014. 1

[3] M. Hausknecht and P. Stone. Deep recurrent q-learning for partially observable mdps. *CoRR, abs/1507.06527*, 2015. 1

[4] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016. 1

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. 1

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. 1

[7] S. Racanière, T. Weber, D. Reichert, L. Buesing, A. Guez, D. Jimenez Rezende, A. Puigdomènech Badia, O. Vinyals, N. Heess, Y. Li, R. Pascanu, P. Battaglia, D. Hassabis, D. Silver, and D. Wierstra. Imagination-augmented agents for deep reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5694–5705. Curran Associates, Inc., 2017. 1, 2

[8] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1889–1897, 2015. 1

[9] Z. Wang, N. de Freitas, and M. Lanctot. Dueling network architectures for deep reinforcement learning. *CoRR*, abs/1511.06581, 2015. 1