

STRESZCZENIE

Słowa kluczowe:

Dziedziny nauki i techniki zgodne z wymogami OECD:

ABSTRACT

Keywords:

OECD field of science and technology:

CONTENTS

| | |
|---|----|
| List of abbreviations and nomenclature..... | 6 |
| 1. Introduction | 7 |
| 2. Theoretical background | 9 |
| References | 10 |
| List of figures | 11 |
| List of tables | 12 |

LIST OF ABBREVIATIONS AND NOMENCLATURE

1. INTRODUCTION

Computer science defines Artificial Intelligence (AI) as the intelligent agents in form of any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. Progress has been made in developing capable agents for numerous domains using deep neural networks in conjunction with model-free reinforcement learning [7][9][10], where raw observations directly map to agent's actions. However, current state-of-the-art approaches usually are sample inefficient, they require thousands of millions of interactions with the environment, and lack the behavioural flexibility of human intelligence, hence the resulting policies poorly generalize to novel task in the same environment.

The other branching of reinforcement learning algorithms is called model-based reinforcement learning, which gives the agent access to (or learns) a model of the environment. There are many orthogonal ways of using the model: one can use the model for data augmentation for model-free methods[4], some methods use the model as the imagined environment to learn model-free policy in it[5], other methods focus on simulation-based search using the model[12] and there are even methods that integrate model-free and model-based approaches[13]. The model allows the agent to simulate an outcome of an action taken in a given state. The main upside to having the model is that it allows the agent to plan by thinking ahead, seeing what would happen for a range of possible choices, and explicitly deciding between possible options without the risk of the adverse consequences of trial-and-error in the environment - including making poor, irreversible decision. Agents can then distill the results from planning ahead into a policy. Even if the model needs to be learned first it can exploit additional unsupervised learning signals, thus it results in a substantial improvement in sample efficiency over methods that don't have a model. Furthermore, the same model can be used by the agent to complete other tasks in the same environment. Planning is different from learning. In the former the agent samples episodes of simulated experience and updates its policy based on them. In the latter the agent also updates its policy, but this time based on real experience gained through interaction with the environment. It's worth noting the symmetry which yields one important implication: algorithms for reinforcement learning can also become algorithms for planning, simply by substituting simulated experience in place of real experience.

Model-free methods are more popular and have been more extensively developed and tested than model-based methods. While model-free methods forego the potential gains in sample efficiency from using a model, they tend to be easier to implement and tune. The main downside of model-based reinforcement learning is that a ground-truth model of the environment is usually not available to the agent. If an agent wants to use a model in this case, it has to learn the model purely from experience, which creates several challenges. The biggest challenge is that bias in the model can be exploited by the agent, resulting in an agent which performs well with respect to the learned model, but behaves sub-optimally in the real environment. Model-learning in complex domains is fundamentally hard and requires function approximation, so resulting models are inherently imperfect. The performance of agents employing standard planning methods usually suffer

from model errors. Those errors compound during planning, causing more and more inaccurate predictions the further plans' horizon.

There are many real-world problems that could benefit from application of general AI system. Company called DeepMind, driven by their experience from creating winning Go search algorithm AlphaZero[12], published AlphaFold[3], a system that predicts protein structure. The 3D models of proteins that AlphaFold generates are far more accurate than any that have come before making significant progress on one of the core challenges in biology. Real-world applications of AI algorithms like this are often limited by the problem of sample inefficiency. In setting with e.g. a physical robot the AI agent can't afford much trial-and-error behaviour, that could cause damage to the robot, and do this over thousands of millions of time steps, for each task separately, in order to build a sufficiently large training dataset. Those machines work in the real world, not accelerated computer simulation, and often need a human assistance. To apply sample efficient model-based systems, that can generalize their knowledge, accurate models are needed.

This work explores progress in the model-learning domain and examine a possibility of using learned models in simulation-based search algorithms. Recently, there have been major steps taken in the domain of model-learning algorithms[2][8][1][6]. More accurate models, at least in short horizon, open the path for application of proofed simulation-based search planning algorithms like TD-Search[11] or AlphaZero[12] to complex planning problems in environments with discrete state-action spaces and without access to a ground-truth model. This should allow for improvements in data efficiency, generalization and agents performance for these problems. This work focuses on two benchmarks: a complex puzzle environment, Sokoban and a multi-task environment, MiniPacman.

2. THEORETICAL BACKGROUND

REFERENCES

- [1] Lars Buesing et al. "Learning and Querying Fast Generative Models for Reinforcement Learning". In: *arXiv e-prints* (2018). arXiv: 1802.03006.
- [2] Silvia Chiappa et al. "Recurrent Environment Simulators". In: *arXiv e-prints* (2017). arXiv: 1704.02254.
- [3] Richard Evans et al. "De novo structure prediction with deep-learning based scoring". Dec. 2018. URL: <https://deepmind.com/blog/alphafold/>.
- [4] Vladimir Feinberg et al. "Model-Based Value Estimation for Efficient Model-Free Reinforcement Learning". In: *arXiv e-prints* (2018). arXiv: 1803.00101.
- [5] David Ha and Jürgen Schmidhuber. "World Models". In: *arXiv e-prints* (2018). arXiv: 1803.10122.
- [6] Danijar Hafner et al. "Learning Latent Dynamics for Planning from Pixels". In: *arXiv e-prints* (2018). arXiv: 1811.04551.
- [7] Matteo Hessel et al. "Rainbow: Combining Improvements in Deep Reinforcement Learning". In: *arXiv e-prints* (2017). arXiv: 1710.02298.
- [8] Felix Leibfried, Nate Kushman, and Katja Hofmann. "A Deep Learning Approach for Joint Video Frame and Reward Prediction in Atari Games". In: *arXiv e-prints* (2016). arXiv: 1611.07078.
- [9] Volodymyr Mnih et al. "Asynchronous Methods for Deep Reinforcement Learning". In: *arXiv e-prints* (2016). arXiv: 1602.01783.
- [10] John Schulman et al. "Proximal Policy Optimization Algorithms". In: *arXiv e-prints* (2017). arXiv: 1707.06347.
- [11] David Silver, Richard S. Sutton, and Martin Müller. "Temporal-difference search in computer Go". In: *Machine Learning* 87.2 (2012), pp. 183–219. ISSN: 1573-0565. DOI: 10.1007/s10994-012-5280-0. URL: <https://doi.org/10.1007/s10994-012-5280-0>.
- [12] David Silver et al. "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". In: *arXiv e-prints* (2017). arXiv: 1712.01815.
- [13] Théophane Weber et al. "Imagination-Augmented Agents for Deep Reinforcement Learning". In: *arXiv e-prints* (2017). arXiv: 1707.06203.

LIST OF FIGURES

LIST OF TABLES