INSTITUT FÜR INFORMATIK
Machine Learning
Universitätsstr. 1 D–40225 Düsseldorf

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Actor-Critic Reinforcement Learning With Experience Replay

Julian Robert Ullrich

Bachelorarbeit

Beginn der Arbeit:	25. Juli 2018
Abgabe der Arbeit:	25. Oktober 2018
Gutachter:	Univ.-Prof. Dr. S. Harmeling
	Univ.-Prof. Dr. M. Leuschel
Betreuer:	Julius Ramakers

Erklärung

Hiermit versichere ich, dass ich diese Bachelorarbeit selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Düsseldorf, den 25. Oktober 2018

Julian Robert Ullrich

Abstract

Deep Reinforcement Learning and policy gradient methods majorly contributed to the most recent advances in the field of Artificial Intelligence. These Methods enabled machines to surpass human performance for Atari console games (Mnih et al., 2013), boardgames like Chess, Shogi (Silver et al., 2017a) or Go (Silver et al., 2017b) and most recently even complex team-based computer games (OpenAI, 2018).

As environments get more complex the cost of simulating the environment increases and often outweights the isolated computational cost of training the agent, making sample efficient methods nessecary.

This thesis will take a look at off-policy methods and learning from previously sampled data, with the main focus being the implementation and evaluation of the "Actor-Critic with Experience Replay"(ACER) algorithm proposed by Wang et al., 2016 on the Atari 2600 console games.

Contents

1 Introduction

Sutton and Barto (1998) describes the reinforcement learning task as "learning what to do". Acting optimal within an unknown environment can be very difficult. The field within machine learning adressing this problem is called reinforcement learning.

The reinforcement problem consist of an *agent* taking *actions* within some sort of *environment*. By interacting with the *environment* the *agent* receives *rewards*.

The goal of reinforcement learning is to create fast and reliable learning algorithms for the *agent* to gain the maximum *reward*

Environments can range from simple tasks, like balancing a pole to very complex and demanding tasks where *environment states* are given as pixels, continuous control problems or real life robotic tasks. This thesis will work with the Atari 2600 environments offered by OpenAI (Brockman et al., 2016)

By combining deep learning techniques (Hinton and Salakhutdinov, 2006) with reinforcemen learning, the problems posed by most of the Atari games can easily be solved.

However complex environment like the Atari 2600 games can often be costly to simulate.

ACER (Wang et al., 2016) provides a sample efficient learning agent. This work aims at implementing and evaluating the Algorithm

This thesis will provide a short overview for important concepts of reinforcement learning

To lay out the foundation for ACER, policy gradient, specifically Actor-Critic methods and the Advantage Actor Critic(A3C) - Algorithm (Mnih et al., 2016) are looked upon, followed by an introduction to some approaches of Off-Policy learning.

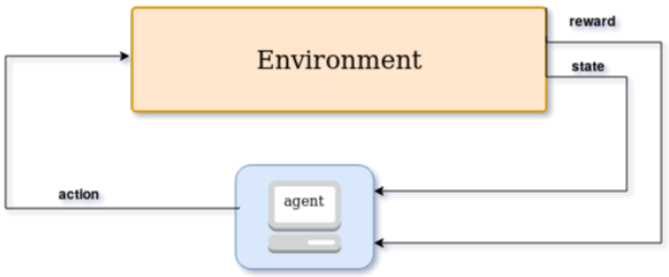
Finally the ACER- Algorithm is presented, implemented and evaluated.

2 Reinforcement Learning Framework

The core opponent of the reinforcement learning framework are the *agent* and the *environment*.

An agent interacts with the *environment* over time, by taking in the environment state, evaluating the state and deciding on an *action*. A core question is the one of exploration and exploitation. In order to learn the best behaviour the *agent* needs to make sure to explore the *environment* to avoid getting stuck on local maxima, however at some point the gained knowledge should be used to achieve the best possible reward.

Whenever the agent interact with the environment, a reward and a new state are given to him in return.



2.1 Elements of Reinforcement Learning

Sutton and Barto (1998) names 4 core elements of the reinforcement learning framework.

Policy

The behaviour of the agent within at any given time is determined by the *policy*. A policy can roughly be described as a mapping of states to an action or a distribution over actions.

Reward Signal

The problem posed by an environment is defined through the reward function. The goal of the learning agent is to receive the maximum accumulated future reward at any given time. One of the most important features of reinfocement learning is the fact, that rewards are often very delayed. Good opening moves in for example Chess will play a major role in winning the game, which however usually occurs at a much later stage.

Value Function

The value of a state (or a state - action pair) describes how much more reward can be earned from this state onwards. Values represent the sum of the future rewards, and indicate the long term desirability of states.

Environment Model

In order to solve a problem, a model of the environment can be learned and used for planning. A model can be used to predict future states and rewards before they happen. Model-based and model-free reinforcement learning methods, which explicitly learn by trial and error both play an important role in reinforcement learning.

2.2 Markov Decision Process

The sequential decision making process of the *agent* can be more formally described as a Markov decision process (MDP).

The sequential decision making process is given by a sequence of states, actions and rewards:

$S_0, A_0, R_0, S_1, A_1, R_1, S_2, A_2, R_2, \ldots, S_t, A_t, R_t, S_{t+1}$

Within this thesis a finite environment is assumed. We call a state *Markov* or say it has *Markov property* if it only depends on it's predecessor rather than the whole history.

$Pr(s_{t+1} = s', r_1 = r \mid s_t, a_t, r_t, s_{t-1}, a_{t-1}, \ldots, r1, a0, s0) = Pr(s_{t+1} = s', r_1 = r \mid s_t, a_t)$

A finite discounted Markov decision process $MDP(S, A, P_a, R_a, \gamma)$ contains a finite set of states S, a finite set of Actions A, the transition probablity to end up in state s' if action a is taken in state s $P^a_{ss'} = Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$, the reward function $R^a_{ss'}$, and the discount factor $\gamma \in [0, 1)$, used to define the importance of immediate reward in contrast to future reward.

References

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba (2016). "OpenAI Gym". In: *CoRR* abs/1606.01540.

G. E. Hinton and R. R. Salakhutdinov (2006). "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786, pp. 504–507. eprint: http://science.sciencemag.org/content/313/5786/504.full.pdf.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu (2016). "Asynchronous Methods for Deep Reinforcement Learning". In: *CoRR* abs/1602.01783.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller (2013). "Playing Atari with Deep Reinforcement Learning". In: *CoRR* abs/1312.5602. arXiv: 1312.5602.

OpenAI (2018). *OpenAI Five*.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis (2017a). "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". In: *CoRR* abs/1712.01815. arXiv: 1712.01815.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis (2017b). "Mastering the game of Go without human knowledge". In: *Nature* 550. Article.

Richard S. Sutton and Andrew G. Barto (1998). *Introduction to Reinforcement Learning*. 1st. MIT Press.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Rémi Munos, Koray Kavukcuoglu, and Nando de Freitas (2016). "Sample Efficient Actor-Critic with Experience Replay". In: *CoRR* abs/1611.01224. arXiv: 1611.01224.

List of Figures

List of Tables