Reinforcement learning on Space Invaders

Strobel Maximilian
max.strobel@tum.de

Werhahn Maximilian
maxi.werhahn@gmx.de

Kiener Martin
martin.kiener@tum.de

Seferis Emmanouil
ga62cac@tum.de

Project Proposal

1. Introduction

We want to train an intelligent agent, who is capable of playing Space Invaders at a human level. During a first step we want to implement a deep Q-network (DQN) with convolutional layers to get above the score of random play. In the next step we want to gain average human performance by optimizing hyperparameters. Afterwards we attempt to optimize the networks architecture to perform on an above human level. Finally we compare our results with the results in the paper on that particular game and with other AI algorithms.

1.1. Related Works

We were inspired by the paper ”Human-level control through deep reinforcement learning” [2], which proposed an approach how to train convonlutional neural networks on classic Atari 2600 games. In the mentioned paper the authors were able to train an agent with skills classified higher than a professional human games tester. In the beginning we try to reproduce the results of this paper. After successfully implementing this architecture we try to exceed the results of this paper in this particular game.

2. Dataset

For our project we decided to use OpenAI Gym as framework for the game. OpenAI Gym is a toolkit for developing and comparing reinforcement learning algorithms implemented in Python. It allows us to gain information about the current state of the game in form of rewards and images. Instead of images it is also possible to use the memory of the current state of the game to directly access the positions of all entities in the game. It has also an API to control the agent in the selected environment.

Due to the fact that we use an reinforcement learning approach for our project, we do not need any labeled data and use the OpenAI Gym framework to gather inputs for our neural network. As score function we will use the rewards, that we can access from the Open AI Gym API. The input of the network is either memory or images. The output on the other hand is an amount of actions taken(left/right/shoot).

3. Methodology

3.1. Architecture

As already mentioned in the introduction we try to implement the already existing architecture of a DQN. After the performance evaluation of a DQN, we try to optimize this architecture to improve the results by adding BatchNorm or MaxPool layers, since the original approach did only use conv, relu and fc layers.

As framework for our implementation we use PyTorch.

3.2. Training

We do not use pretrained networks, but train the network from scratch. During the training we want to use a population based approach [1]. With population based training we try to accelerate the training and also gain performance benefits.

3.3. Ressource management

In our project we want to use the Google Cloud Platform and also the machines of the lab for a faster training on GPUs. For development and document sharing we considered to use GitHub.

3.4. Division of labour

4. Outcome

We try to build an agent with above human level skills on Space Invaders. We evaluate our own network performance and compare it with the results from the paper and other existing implementations, which are also provided by OpenAI Gym.

References

[1] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, C. Fernando, and K. Kavukcuoglu. Population based training of neural networks, 2017. 1

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou,

H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529 EP –, Feb 2015. 1