University of Tartu

Faculty of Mathematics and Computer Science

Institute of Computer Science

Kristjan Korjus, Ilya Kuzovkin, Ardi Tampuu, Taivo Pungas

# Replicating the Paper "Playing Atari with Deep Reinforcement Learning"[MKS+13]

## Technical Report

MTAT.03.291 Introduction to Computational Neuroscience

Tartu 2014

# Contents

# Introduction

In the recent years the popularity of the method called *deep learning* [REF] has increased noticeably in the machine learning community. Deep leaning was successfully applied to speech regonition [REF], something else [REF] and something else [REF]. In all these studies the performance of the resulting system was better than other machine learning methods were able to achieve so far.

The core of deep learning method is an artificial neural network. One of the properties, which gives this family of learning algorithms a special place, is the ability to extract "meaningful" (from the human perspective) *concepts* from the data by combining the features based on the structure of the data. The extracted concepts sometimes have clear interpretation and that makes us feel as if machine indeed has *learned* something. Here we step into the realm of artificial intelligence, the possibility of which never stops to fascinate our minds.

One of the recent works, which brings together deep learning and artificial intelligence is a paper "Playing Atari with Deep Reinforcement Learning"[MKS+13] published by Deep-Mind[1] company. The paper describes a system, which combines deep learning methods and *reinforcement learning* in order to create a system that is able to learn how to play simple computer games. It is worth mentioning that the system has access only to the visual information (screen of the game) and the scores. Based on these two inputs the system learns to understand which moves are good and which are bad depending on the situation on the screen. Notice that the human player uses exactly same information to evaluate his performance and adapt his playing strategy. The reported result shows that the system was able to master several different games and play some of them better than the human player.

This result can be seen as a step forward to the truly intelligent machines and thus it fascinates us. The goal of this project is to create an open-source analogue of such a system using the description provided in the paper.

---

[1]http://deepmind.com

# 1 A bird's-eye view of the system

Before we go into the details, let us describe the overall architecture of the system and show how the building blocks are put together.

## 1.1 The task

The system receives a picture of a game screen (an example is shown in Figure 1) and chooses an action to take. It then executes this action and is told whether the score increased, decreased or did not change. Based on this information and playing a large number of games, the system needs to learn to improve its performance in the game.
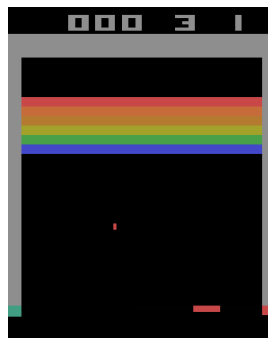


Figure 1: A game screen of Breakout

## 1.2 Reinforcement learning

This is a techinque...

### 1.2.1 Exploration-exploitation

multiarmed bandits

## 1.3 Neural network

general description, we have input – images, scores (where those go?), then hidden layers which correspond to blabla and output layer – actions. Learning happens by updating weights blabla

## 1.4 Learning process

gradient descent blabla

# 2 Components of the system

here goes the detailed description of everything (aka the hardest part)

## 2.1 Convolutional neural network

...

## 2.2 Q-learning

...

## 2.3 Something else

...

## 2.4 And once again how all those things are put together

# 3 Implementation details

Programming language: Python 2.7 (32-bit).

Python libraries: Pillow, NumPy, Theano.

ATARI emulation environment: ALE (Arcade Learning Environement).

## 3.1 Agent

...

## 3.2 Memory

...

## 3.3 Preprocessing

The game screens are preprocessed by first cropping the original 160x210-pixel image to a 160x160 region of interest, which is then downscaled to a 84x84 image.

The colors from ATARI's NTSC palette are converted to RGB using a conversion table [2]. The RGB representation is then converted to grayscale according to the weighted combination $0.21r + 0.71g + 0.07b$. This should produce a representation close to human perception (humans are more sensitive to green than other colours). An example of a preprocessed image is shown in Figure 2.



Figure 2: A preprocessed game screen of Breakout

## 3.4 Any other implementational stuff which is worth describing

...

---

[2]http://www.biglist.com/lists/stella/archives/200109/msg00285.html

# 4 Results

Our system rules!

## 4.1 Performance measures

## 4.2 Comparison to human player

...

## 4.3 Comparison to the paper

...

## 4.4 Something else

...

## 4.5 Applications and future blah

...

# Appendix A: Running instructions

to test the system you should do this and that

you will see that blabla

this will make you happy

also go to github read wiki/code there for more details

# References

[MKS+13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.