Institute of Computer Science, University of Tartu

Summary of Playing Atari with Deep Reinforcement Learning

Kaspar Kadalipp

Abstract

This paper introduces are a new kind of deep reinforcement learning model that learns to play seven Atari 2600 games using only in-game screenshots as input with no prior knowledge about the games. The model is a convolutional neural network, trained with a variant of Q-learning, whose input is raw pixels and whose output is a value function estimating future rewards. This model manages to outperform multiple reinforcement learning models that take advantage of prior knowledge of the games and even manages to surpass a human in three games.

1 Introduction

Best-performing reinforcement learning models that learn to play games heavily rely on hand-crafted features combined with certain states mapped to certain actions. So, the performance heavily relies on the human factor. This paper demonstrates that a convolutional neural network can overcome this with minimal human input. The network is trained with a variant of the Q-learning [2] algorithm, with stochastic gradient descent. To alleviate the problems of correlated data and non-stationary distributions, an experience replay mechanism [3] which randomly samples previous transitions is used and therefore smooths the training distribution over many past behaviors.

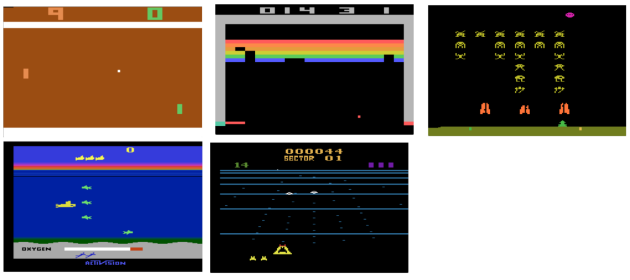

Figure 1: Screenshots from five Atari 2600 Games: (Left-to-right) Pong, Breakout, Space Invaders, Seaquest, Beam Rider.

The goal is the create a single neural network agent that's able to learn to play various Atari 2600 games in The Arcade Learning Environment (ALE) without any game-specific knowledge, just like a human would when trying out a game for the first time. This is challenging as the agent is faced with a diverse set of tasks and high dimensional game input (210 X 160 RGB video at 60 Hz). Figure 1 provides sample screenshots from five of the games used for training.

2 Background

The agent interacts with the Atari emulator in a sequence of actions, observations and rewards. At each step an action is chosen from a set of legal game actions, which is then passed to the emulator and as a result the state of the game changes. The agent receives a reward based on the game score.

Feedback about an action may only be received after many thousands of time steps. Since the agent only observes images of the current screen it must take into consideration many sequences of actions to fully understand the situation. All sequences in the emulator are assumed to be finite. This assumption allows to apply Markov's decision process (MDP) in which each sequence is a distinct state. The agent interacts with the emulator in a way that maximizes future rewards. Using Q-learning [2] is an efficient way to achieve just that.

3 Deep Q-Networks

The key difference in this algorithm is that it uses experience replay [3]. Consecutive frames are very correlated and have no variability, using this for training can cause unwanted feedback loops [4]. Rather than using samples as they are collected, they are put into an replay buffer and from that random samples are taken from there to train the model. This kind of sampling proved to be very important for this algorithm, as it gives equal importance to all transitions.

The neural network itself is small with only 3 hidden layers, uses large filters, ReLu for nonlinearity and outputs a value for every possible action in a game.

3.1 Training

To train the model it is important to interpret the change in score similarity in different games. To achieve that, all positive rewards were transformed to 1, negative rewards to -1 and no change in score to 0. This change allows using the same learning rate in all games. To speed up the model and enable more iterations, every 4th frame was skipped, this significantly reduces computation time and thus more games could be played. Space Invaders game required every 3rd frame to be skipped, otherwise frames with the laser would be skipped. They show that the reward in these kind of games is noisy but certainly improves over time, as can be seen from figure 2.

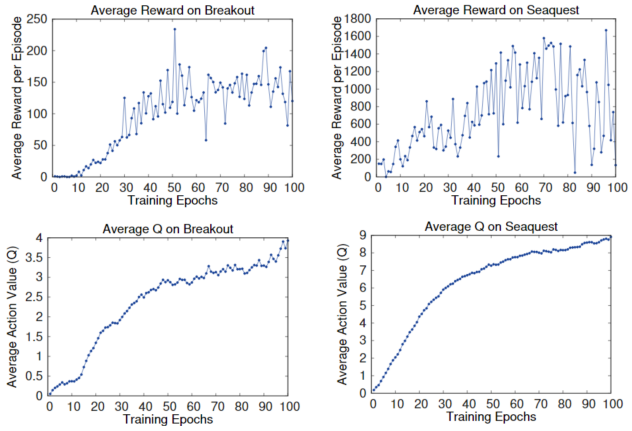

Figure 2: The upper two plots show the average reward per episode during training. The bottom two plots show the average maximum predicted action value. One epoch corresponds to 50000 minibatch weight updates.

They also had a figure inspecting the Q function in Seaquest. Q functions predicts what the future reward is going to be. Its value started to increase once a new enemy appears, reached its peak when the enemy was about the be destroyed and drops drastically once the enemy is destroyed, meaning there's no imminent future reward.

3.2 Evaluation

Most compared methods have some kind of very special feature engineered [5, 6]. If their method takes only RGB the other methods recognize that these are Atari games, they know that certain items have unique colors and they make unique channels to recognize them or even have hand crafted object detectors. So the comparison really isn't fair. But their algorithm (DQN) outperformed such algorithms almost everywhere, as can be seen in Table 1.

	B. Rider	Breakout	Enduro	Pong	Q*bert	Seaquest	S. Invaders
Random	354	1.2	0	−20.4	157	110	179
Sarsa [5]	996	5.2	129	−19	614	665	271
Contingency [6]	1743	6	159	−17	960	723	268
DQN	**4092**	**168**	**470**	**20**	**1952**	**1705**	**581**
Human	7456	31	368	−3	18900	28010	3690

Table 1: The table compares average total reward for various learning methods.

They also evaluated the performance of a human. In Table 1 the reward is the median score achieved in 2 hours of gameplay. The score achieved in simpler games beat the score of a human. However, there were still problems where the human achieved a vastly higher score than the model. They mainly attribute this to the complexity of the games, as they require to plan many steps ahead.

4 Conclusion

This paper introduced a new deep learning model for deep reinforcement learning and demonstrated its ability to master playing Atari 2600 games, using only raw pixels as input. This approach managed great results in six of the seven tested games, with no adjustment of the architecture or hyperparameters.

5 References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller. Playing Atari with Deep Reinforcement Learning, 2013.

[2] Christopher JCH Watkins and Peter Dayan. Q-learning.Machine learning, 8(3-4):279–292,1992.9

[3] Long-Ji Lin. Reinforcement learning for robots using neural networks. Technical report, DTICDocument, 1993.

[4] Richard Sutton and Andrew Barto.Reinforcement Learning: An Introduction. MIT Press,1998.

[5] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learningenvironment: An evaluation platform for general agents.Journal of Artificial IntelligenceResearch, 47:253–279, 2013.

[6] arc G Bellemare, Joel Veness, and Michael Bowling. Investigating contingency awarenessusing atari 2600 games. InAAAI, 2012.