

# Playing ATARI with Model Based Reinforcement Learning

Alexander Bertsch and Mel Roderick

February 28, 2016

## 1 Objectives

We seek to improve upon the performance of “Playing Atari with Deep Reinforcement Learning”, by Mnih et al. We seek to find out whether a model based approach could improve upon their results. We plan to use the same environment and similar games in order to more accurately compare our findings.

## 2 Current Practice

Within Mnih et al.’s paper, they present a model of deep reinforcement learning which they use to play Atari games using the Arcade Learning Environment. They use a modified Q-learning method in which each state in the MDP is a vector of pixels. In order to learn the value function  $Q(s, a)$ , they run stochastic gradient descent on a CNN. Their results revolve around using this CNN rather than a linear function approximator to find  $Q^*(s, a)$ . Their model outperformed an expert on several games, including Breakout, Enduro, and Pong. However, on other games, including Q\*bert, Seaquest, and Space Invaders, their model performed much worse than an expert. Their models were all built on top of the Arcade Learning Environment.

The Arcade Learning Environment (ALE) is a simple object-oriented framework that aids development of AI agents for Atari 2600 games. The environment is built on top of an emulator and abstracts away emulation from agent design. Game score and the end-of-game signal for a large amount of games. In addition, ALE provides a set of visualization tools that can simplify analysis of agent performance.

In our model, we plan to use object recognition to isolate objects in the games. Detection is typically done through using edge filters and then finding connected components within an image. Recognition is typically done through training a bag-of-words model on a series of images. From here, specific objects can be isolated and recognized.

Non-model-based methods are limited. As the state space grows and as transition functions becoming increasingly complex, these methods begin to become increasingly unwieldy. Conversely, model-based RL tends to grow in complexity slower [2].

## 3 Approach

Contrasting with Mnih et al.’s CNN based approach, we instead plan to use BURLAP’s OO-MDP structure along with object recognition. We first plan to create some method of recognizing unique objects within the ALE environment. We expect to have our model “play” a game one or more times, see objects as they occur, and categorize them. We would then run a classifier every frame

to determine the current state, all of the objects with their Cartesian coordinates. To facilitate BURLAP, we would need to create a set of our own propositional functions.

Each propositional function would be specific to an Atari game. Consider Space Invaders versus Pong. In Space Invaders, we would likely want to keep track of objects' relative horizontal locations to the agent. Whereas in Pong, we are much more interested in the relative vertical and horizontal locations of the ball to the paddle. Thus, our propositional functions will need to be specialized to each game, although the degree of specialization is unlikely to be that high.

As seen in "An Object-Oriented Representation of Efficient Reinforcement Learning" by Diuk et al. (Rutgers), model-based approaches for RL have been shown to greatly improve the learning rate of MDPs. We hope to have similar results in our work, that is we hope to improve upon what was found in Mnih et al. by using a model based approach.

## 4 Impact

First and foremost, our AI will be really cool to watch.

On a more serious note, this approach could be a proof of concept of vision-based reinforcement learning. Vision-based reinforcement learning is becoming a popular field, but almost all current implementations are model-free. We hope to show that a model-based version could be more effective. In theory, our results could be applied to robotics as well.

## 5 Timeframe

We expect this to be a rather involved project. We need to first design object recognition algorithms for Atari games. Likewise, we'll need to create several sets of propositional functions. We'll then need to collect both of those and use them within BURLAP. Setting up ALE and connecting it to BURLAP will in and of itself be complicated, although doable well within the first checkin. We plan to begin by working with only Space Invaders and broadening our search if time permits.

By the midterm checkin, we plan to have integrated ALE with BURLAP and created an initial implementation of an object recognition system for Atari games. Specifically with Space Invaders, we plan to have a system that can identify the various ships on the screen, as well as barriers and bullets.

By the final checkin, we will have created a set of propositional functions and used those to train OO-MDPs and will have hopefully achieved some positive results.

## References

- [1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller, *Playing Atari with Deep Reinforcement Learning*, DeepMind Technologies, London, 2013.
- [2] Carlos Diuk, Andre Cohen, Michael L. Littman, *An Object-Oriented Representation of Efficient Reinforcement Learning*, Rutgers University, Piscataway, New Jersey, 2008.
- [3] M. G. Bellemare, Y. Naddaf, J. Veness, M. Bowling, *The Arcade Learning Environment: An Evaluation Platform for General Agents*, Journal of Artificial Intelligence Research, 2013.