Baby’s First Reinforcement Learning

Brandon Foltz

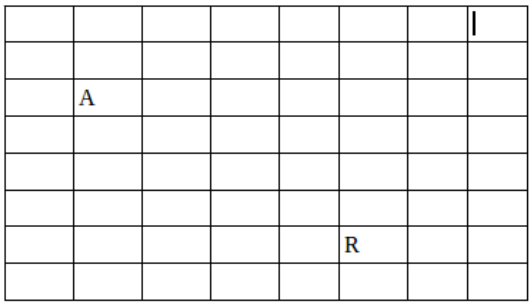
Introduction

This project is intended to provide a basic understanding of reinforcement learning algorithms, including their motivations and various techniques for implementing them. Reinforcement learning refers to a particular set of machine learning techniques which enable an “agent” to navigate some environment and learn to maximize some “reward”. This could be something like playing a video game, or driving a car without crashing, for example.

Reinforcement learning has seen considerable successes in recent years. Groups such as OpenAI and Google Deepmind have successfully trained agents to play Atari games[1] with super-human skill. More recently, groups have trained agents to play considerably more complex games like Dota II. In each of these cases, the AI agent is trained by allowing it to explore the environment, and discovering on it’s own which sequences of actions lead to maximization of the chosen reward.

Proposed Work

For a very first foray into the realm of reinforcement learning, I propose a very simple environment in which a single agent aims to move about on a grid and collect “food pellets” that appear on the grid. Collecting these food pellets would be the event which releases some “reward” to the agent (agent labeled ‘A’ and reward labeled ‘R’ in the example image to the right. You could think of this as a simplified version of the classic “snake” game, where the player moves about to collect food but in this case the player (or AI agent) does not need to worry about colliding with it’s own tail.



This is of course a trivial problem to solve for any programmer, as the intuitive goal is for the agent to move in the direction of the reward at all times. However I wish to implement a Markov Decision Process[2] training algorithm in which the probabilities of selecting an action are learned by the history of actions that led to the collection of a reward.

I chose a Markov Decision Process as the first implementation because it is perhaps the easiest tool to understand for solving this type of problem, especially an example as simple as this. Hence, “Baby’s First” reinforcement learning algorithm. If I am successful at this implementation and time allows, I would like to attempt to train a neural network to solve the same problem by using the PyTorch[3] deep learning library. I have some experience using PyTorch to build neural networks for classification tasks, but have not applied them to a task like the one described above. This would allow me to (potentially) explore two different methods of solving the same problem in a reinforcement learning context, and discover the pros and cons of each.

Timeline (Week 1)

I am confident that I will be able to implement the basic “game” environment code, whereby a human could navigate the agent around the grid and collect rewards themselves. Perhaps I will even be able to find an existing implementation of something very similar to this (such as the snake game) and adapt it to my needs. This code would include functions that an AI agent could use to navigate around the environment and collect information about it as well.

Timeline (Week 2)

The implementation and training algorithm for a Markov Decision Process should be possible to complete by the end of week 2. This is the very minimum I would expect to complete with the time allotted, and so I will include all the details of this experience in the final report.

Final Report

The final report will contain what I have learned in the implementation of the agents environment and the training of the Markov Decision Process. Training a neural network to complete the same task might be ambitious in the given time, but at minimum I will include how I expect this would work and a proposed approach in a “future work” section of the report. The final deliverables will include all the code I wrote such that my experiments could be repeated by someone else.

References

[1] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[2] https://en.wikipedia.org/wiki/Markov_decision_process

[3] https://pytorch.org/