239AS Project Milestone Report S2021

Samuel Gessow
sgessow@ucla.edu
604781350

Sunay Bhat
sunaybhat1@ucla.edu
905629072

Yi-Chun Hung
yichunhung@ucla.edu
705428593

Vahe Gyuloglyan
vgyulogl@ucla.edu
905528327

Abstract

This milestone progress report outlines a novel take on the standard cart-pole reinforcement learning problem we are investigating for our course project. We are exploring a multi-cart version in which two carts operate two poles that are jointed together and support a pendulum. This extension will allow us to explore the standard cart-pole problem with considerably more complex dynamics. We will then extend this environment to a multi-agent version in which each cart is an independent agent, thus allowing for more complex reward structures and dynamics beyond the base multi-cart problem. Our intention was to expand on the dominant algorithmic techniques in the literature for the more complex dynamics in our environment. In addition, we intend to play with the different types of reward combinations to better understand the multi-agent problem feasibility.

1 Introduction

The original cart-pole problem is well studied in reinforcement learning (RL) [1], and our extension will allow us to explore more advanced dynamics. Using Pymunk for the physics simulation and OpenAi Gym for the RL framework, we developed the "carts-poles" test environment. An illustration of the environment can be seen in Figure 1 below. There are twelve total state variables (2 per cart, 2 per pole). In Figure 1, a rendering of our completed environment can be seen as well.

Figure 1: The illustration and render of our environment, including two carts, three joints, and a pendulum.

As detailed in our project proposal, our initial literature search focused on extensions of the cart-pole problem, particularly double [2] and triple pendulums [3]. There is plenty of research in more complex cart-pole or pendulum problems, but none that we found which explore a more complex variation with two-carts, and thus the possibility for two agents cooperating to achieve the same result. The dynamics of the real world often include complex systems which have high degrees of non-linearity and multiple "agents" that operate on imperfect or minimal information. Our goal is to begin to explore such dynamics with the state-of-the-art RL techniques and discuss our results.

Based on the definition of success in previous cart-pole problems, we define a successful completion as a method that can stabilize the pendulum for 200 seconds starting at any angle between -.2 and .2 radians.

Preprint. Under review.

2 Preliminary Results

Our initial testing was focused on proving that the system can be controlled using a single agent such that the pendulum never deviates $pi/8$ in either direction from the center and the carts never go 4 meters from the starting position in either direction. The agent can execute 1 of 9 actions consisting of combinations of applying a left, a right, or no force on each cart independently. With the environment generated, the next step was implementing and testing a variety of reinforcement learning algorithms to explore which ones were the most successful.

One algorithm was the Deep Q Network, or DQN, that applies a neural network to the traditional Q-learning algorithm. Through a variety of runs that involved hyper-parameter tuning, we discovered the best methodology was to start the pendulum angle at a random position within the allowed range to gather more experience quickly. The episode length vs episode number is shown below in Figure 2 (a).

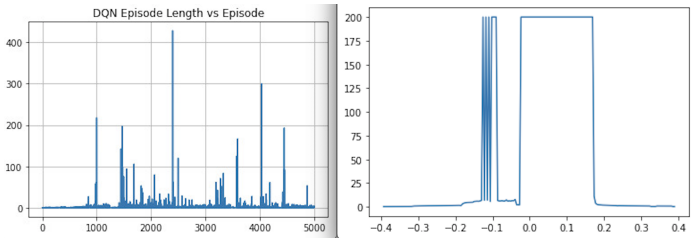

Figure 2: a. Train: Episode Length over iterations b. Results: Episode length vs starting angle

Clearly, the DQN methodology doesn't converge nicely, but this was expected, hence the search for a network with a running average [4]. The neural network that produced the most consistent results across the widest range of starting angles had the results shown in Figure 2 (b) above.

The next method we implemented to improve performance was an "actor-critic" method. In this method, two neural networks, each with two linear hidden layers of 128x256, one for the actor and one for the critic, are used [5]. The actor trains a policy, while the critic trains a value estimator, both using a stochastic gradient descent optimizer. Although this method is quite stable, the training time is considerable. Below in Figure 3 (a), the training curve is shown for 10,000 iterations, still only reaching a maximum episode length of 17 seconds. But the average length, Figure 3 (b), is clearly moving up steadily. The results in Figure 3 (c) are from an agent trained with about 20,000 iterations. This performance shows better results compared to the DQN, though at the cost of considerably more training time.

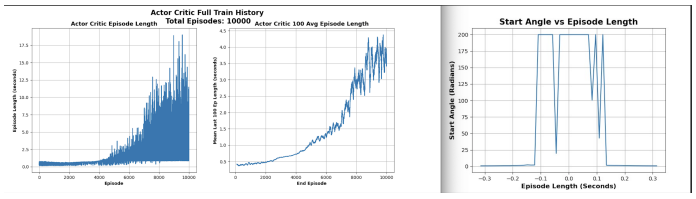

Figure 3: a. Training Episode Lengths b. Train 100 avg c. Results: Episode length vs starting angle

It is clear the DQN and Actor-Critic methods do not solve the problem to our top specification, and we need to explore other algorithms to get the best performance. Two of the additional algorithms are the DDQN as proposed by [6], and the Rainbow method as proposed in [7], which we will explore in detail for the final report.

Finally, we began to explore the two-agent problem, initially using the actor-critic framework as it lent itself well to the extension. We simply instantiated another actor-critic pair, with each actor independent choosing one of three actions (move left, move right, don't move). We also did this using two "information states". In the "full" state, all twelve state variables are provided to both agents, and it is the stochastic differences and independent decision-making generating the increased dynamics. In the "partial" state, each agent only knows its own cart's velocity and position and its own pole's

angle and angular velocity, along with the pendulum pole's angle and angular velocity. The training and initial results for both can be seen in Figure 4 (full info state) and in Figure 5 (partial info state) below.

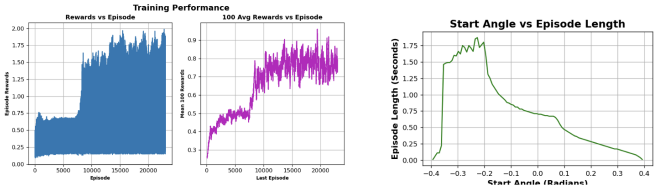

Figure 4: Two Agent (Full Info) Train and Results

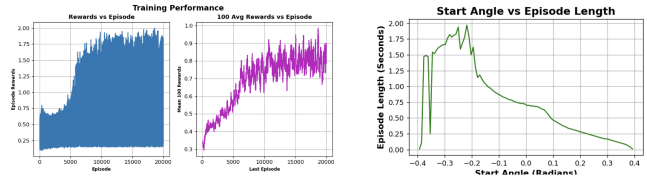

Figure 5: Two Agent (Partial Info) Train and Results

It is clear both agents have a performance preference for one side. In addition, progress is very slow, and the final performance is quite poor, with the two-agents not exceeding 2 seconds even after 20,000 iterations. While the "full" info state performs slightly better, much work is needed to understand why this difference is so minor and both seem to perform so poorly compared to the single-agent baseline. We plan to spend considerable time trying new methods and playing with the reward structure to get much better multi-agent performance. In addition, we will seek to understand why there is a strong "left side performance bias" in the multi-agent problem if it persists.

References

[1] Swagat Kumar. Balancing a cartpole system with reinforcement learning - A tutorial. *CoRR*, abs/2006.04938, 2020.

[2] Fredrik Gustafsson. Control of inverted double pendulum using reinforcement learning. 2016.

[3] Tobias GlüCk, Andreas Eder, and Andreas Kugi. Swing-up control of a triple pendulum on a cart with experimental validation. *Automatica*, 49(3):801–808, March 2013.

[4] Zachary C. Lipton, Jianfeng Gao, Lihong Li, Jianshu Chen, and Li Deng. Combating reinforcement learning's sisyphean curse with intrinsic fear. *CoRR*, abs/1611.01211, 2016.

[5] Sean L. Barton, Nicholas R. Waytowich, Erin G. Zaroukian, and Derrik E. Asher. Measuring collaborative emergent behavior in multi-agent reinforcement learning. *CoRR*, abs/1807.08663, 2018.

[6] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning, 2015.

[7] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Daniel Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *CoRR*, abs/1710.02298, 2017.