Studying the Effects of Intrinsic Rewards for Policy Gradient Methods

Ethan Rathbun
ethan.rathbun@uconn.edu

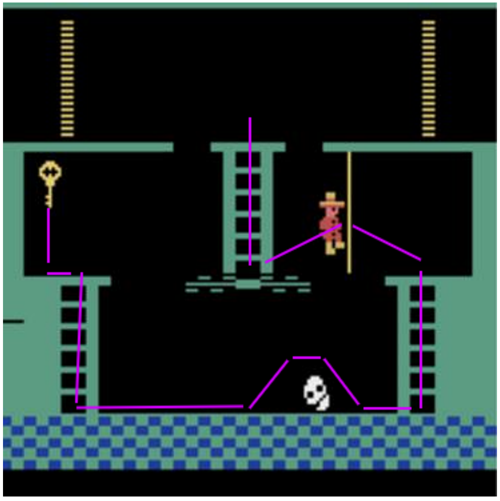
Abstract

This paper aims to further explore the effects of intrinsic rewards for policy gradient methods, building off the work of Zeyu Zheng, Junhyuk Oh, and Santinder Singh in their paper "On Learning Intrinsic Rewards for Policy Gradient Methods" [1] utilizing their algorithm, LIRPG. This paper replicates the results of the original work while also performing additional experiments: one involving agents operating without any extrinsic reward and another testing the algorithm using a different benchmark, "Montuzema's Revenge" for the Atari 2600.

Keywords: Reinforcement Learning; Intrinsic Reward

1 Introduction

Figure 1: First Screen of "Montezuma's Revenge" including the elaborate path (pink) towards the first reward (gold key), Atari 2600 (1983)



Reinforcement Learning is a rapidly growing subfield of Machine Learning and Artificial Intelligence. Its mechanisms are heavily based upon a theory of Psychology called Behaviorism which posits that all intelligent behavior is learned through a process of associating actions with rewards or punishments. While Behaviorism was once a heavily researched theory, it has since been mostly abandoned by the Psychology community as they have found that human behavior cannot be sufficiently explained by behaviorist principals.

In spite of this Reinforcement Learning has, for the most part, still stuck with the behaviorist model. Using Reinforcement Learning techniques autonomous agents are able to solve complicated problems, however there are some tasks these methods struggle with. In particular, agents often struggle in environments with sparse or unclear rewards. This follows simply from the main principals of Reinforcement Learning, if there is no reward or punishment there is nothing to optimize over. Such tasks often require the agent to spend a long time exploring the action and state spaces in the absence of reward.

One classic example of such a task comes in the form of an Atari game 2600 game, "Montezuma's Revenge" (figure 1). In this game the player has to explore multiple rooms to find items and use them to solve puzzles. The player has to determine which items to pick up and hold in their limited inventory along with figuring out how to use the items. Additionally, unlike most games of its era, in "Montezuma's Revenge" the path towards achieving a higher score can be rather elaborate or unclear at the beginning of the game. This requires the player to intelligently explore the environment and figure out which set of actions lead to the best reward, often requiring a great amount of trial and error.

There are a multitude of proposed methods for solving issues relating to sparse rewards. The subject of this paper is to explore the effectiveness of one of these methods, namely the implementation of intrinsic reward. The intuition behind intrinsic reward is that, in addition to the usual extrinsic reward, the agent is given some intrinsic reward for learning new things about its environment. The intrinsic reward is calculated using an internal, critic model which determines the intrinsic reward based upon the agent's current state and actions. Overall, this mechanism can be seen as a type of "curiosity" in which the agent is rewarded for discovering new behavior.

2 Resources Used

This paper is primarily based upon the works of Zeyu Zheng, Junhyuk Oh, and Santinder Singh in their paper "On Learning Intrinsic Rewards for Policy Gradient Methods" [1]. The authors provide the code they used to run their experiments, this code was used as the foundation for all experiments run in this paper.

The code uses Reinforcement Learning and Machine Learning libraries like gym, Tensorflow, and Numpy and is coded in python. Many of these libraries do not need to be interacted with directly as the original code provided by the authors serves as a great, high-level interface for running experiments. Additionally, matplotlib was used to plot the experimental results seen in the next section.

3 Experiments

3.1 Considerations

Due to time limitations, only a few, shortened experiments could be run. In the original paper the authors ran all of their Atari experiments for 50 million time steps, in this paper the timesteps will be limited to 1 million or 5 million. This restriction means that the experiments were carefully chosen in order to, hopefully, be the most insightful ones to run.

3.2 General Experimental Setup

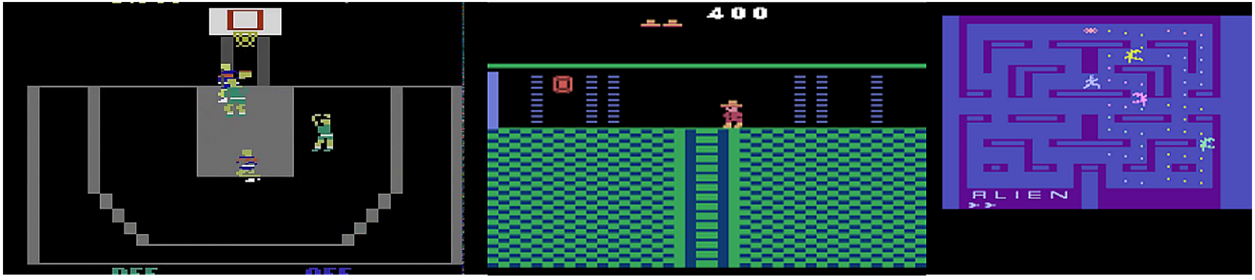
The framework provided by the authors has a great amount of hyper parameters to be chosen. For this paper one primary hyper parameter was chosen to be experimented with, λ, the intrinsic reward coefficient. λ determines how heavily the intrinsic reward is weighed in the total return calculations for the agent, this is done by multiplying the intrinsic reward by λ and adding it to the extrinsic reward. In the original paper they experimented with $\lambda \in \{0.003, 0.005, 0.01, 0.02, 0.03, 0.05\}$, for this paper we will explore values $\lambda \in \{0, .01, .05\}$.

All other hyper parameters, aside from the game being played or the number of steps, are kept constant. The model being used is the "CNN-Int" which is a modified convolutional neural network designed by the authors of the original paper to learn using extrinsic and intrinsic reward. Inside this model there are two sub-models, a policy network that makes decisions and a critic that determines the amount of intrinsic reward given to the agent. The learning rate for both of these models is a constant value of .0007, the original paper uses a linearly decreasing learning rate, but for these shorter experiments a constant learning rate is more desirable.

The plots shown in the next subsection will plot average reward or average game length as a function of time steps, both are calculated using the past 100 games.

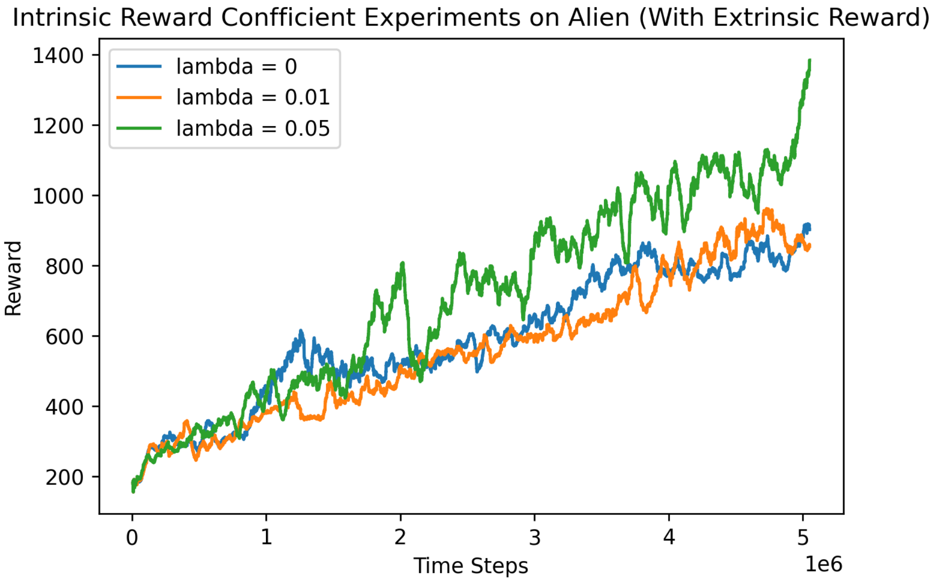
3.3 Results

Figure 2: Left to Right: “Double Dunk”, “Montezuma’s Revenge”, “Alien”; Atari 2600



The first Atari game chosen was “Alien” due to the rapid reward growth shown in the results of the original paper. As previously mentioned, the experiments had to be run for a shortened time, thus games that could be learned quickly was desirable. Alien was one of the easiest games for the various models to learn, hence its inclusion.

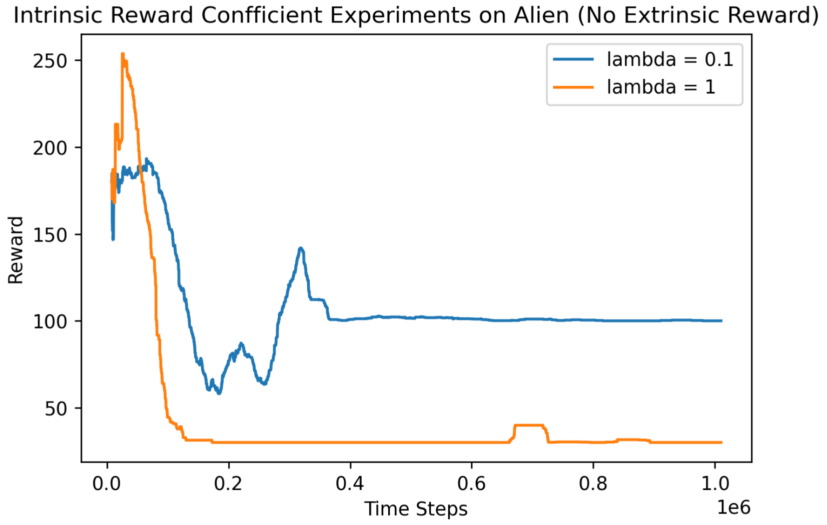
Figure 3: Experimental Results on “Alien” for Models with Extrinsic Reward Included



The first set of experiments on Alien were run for 5 million time steps. These experiments involve the same setup as described in section 3.2. The results show that, at this time scale, the model performed better with a larger value of λ. These results are consistent with those seen in the original paper, if this trend were to continue to a longer time scale we would expect that $\lambda = .05$ would have a significant lead over the other two models. The very end of the graph shows a sudden spike in performance of the $\lambda = .05$ model suggesting it might have been breaking out into a period of rapid improvement around when the experiment ended.

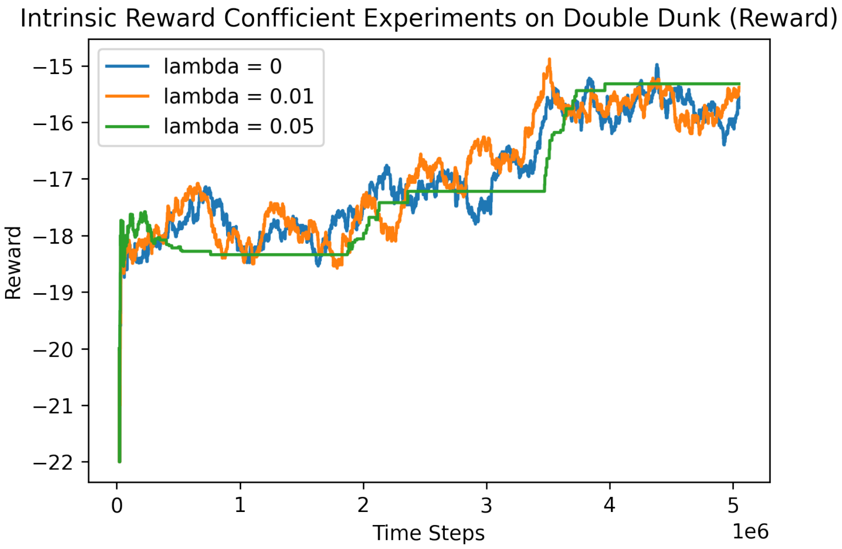
In addition to this experiment another experiment was performed using “Alien” as an environment. This experiment involved completely removing all extrinsic reward and requiring the agent to learn using intrinsic reward alone. In the original paper the authors were able to achieve great results on the Mujoco continuous control benchmark using intrinsic reward alone, however they did not explicitly state how they set up these experiments. Thus, for this paper, the experiments were run using larger values of λ in order to compensate for the lack of extrinsic reward. The results show that for both $\lambda = 1, .1$ the models stagnated and failed to improve. Perhaps there is a better hyper parameter setup to support training on only intrinsic rewards, however that it outside the scope of this paper.

Figure 4: Experimental Results on “Alien” for Models without Extrinsic Reward



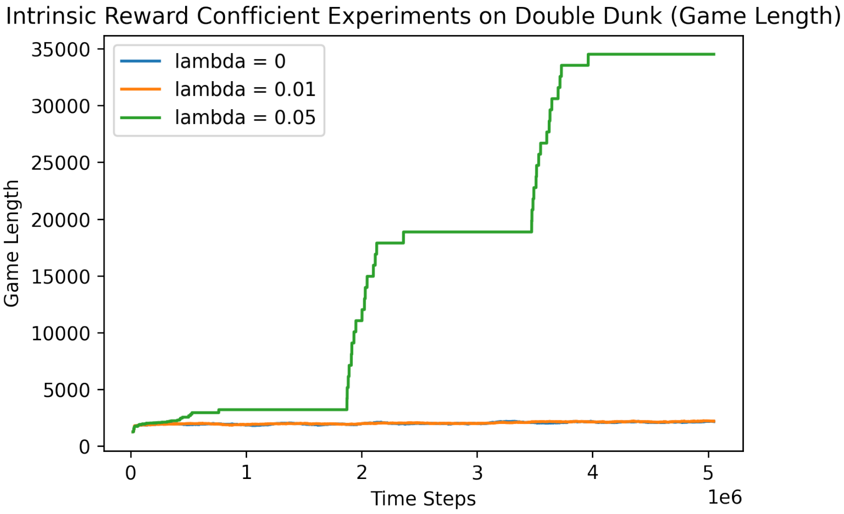
The next game chosen was “Double Dunk” since it showed the most extreme difference between the intrinsic reward and non-intrinsic reward models at the early stages of training in the original paper. One notable thing about this game is that, even for the original paper, the model’s average reward never goes to a positive number. Since this game is a 2v2 basketball game this suggests that the agent is never able to out-score the simple AI opponent included in the game. While this sounds like an indictment to the paper’s results it does make sense, the simple AI follows a pre-defined procedure that was designed by a human while the RL agent has to learn the game from scratch. Additionally, this game is notoriously hard for even human players to win.

Figure 5: Experimental Results on “Double Dunk”



A few observations can be made from these results, the first of which is that there was no clear improvement between the various models when it came to the metric of reward. These results go against the results of the original paper, however they are not incompatible for a few reasons. First, the paper presented results that optimized over more values of λ than were explored here, it is possible that a smaller value of λ was needed to get better results. Furthermore, the small time scale of this experiment may have restricted the intrinsic reward models from reaching their full potential, however they should have still out-performed the $\lambda = 0$ model if they were consistent with the results of the paper.

Figure 6: Experimental Results on "Double Dunk"

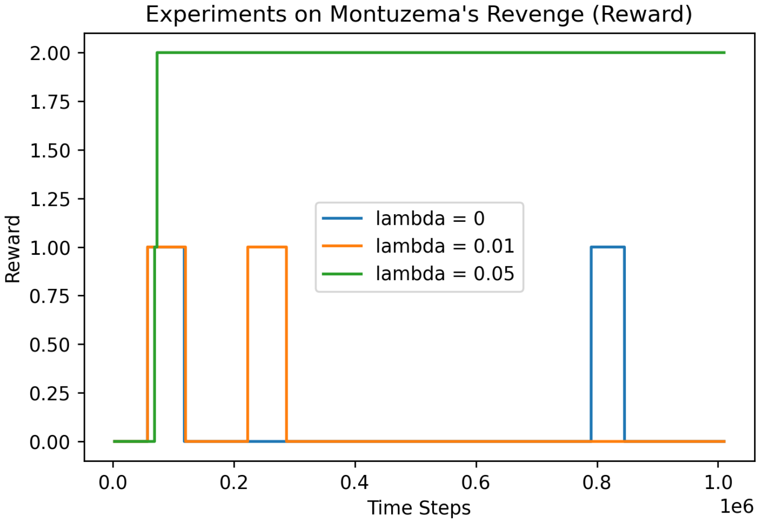


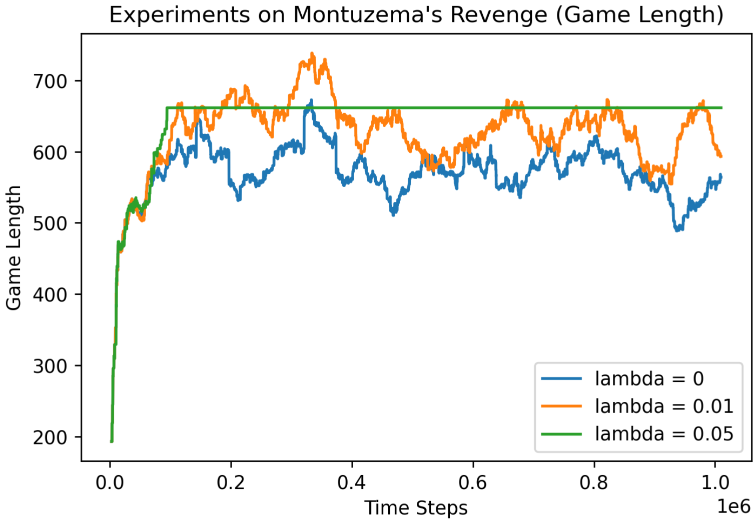
Another observation is in the shape of the model's progress for $\lambda = .05$ as both the reward and game length graphs regularly flat-line before increasing over the course of a few time steps. This behavior is rather interesting as it implies the model doesn't improve or worsen over long stretches of time, but still manages to learn something eventually. This may be indicative of the intrinsic reward critic needing to improve significantly before the main agent is willing to perform new actions again.

Lastly, the game time for the $\lambda = .05$ agent increased significantly while the game time for the other agents remained relatively constant. Without any kind of replay feature, as will be discussed in the section about "Montezuma's Revenge", it is impossible to say for sure what the cause of this is. In double dunk the game ends when one team scores 24 points, so this implies the agent is able to prevent the opponent from winning for a longer time than the other agents, but the way it achieves this is unclear. It is possible that the agent simply avoids having the ball stolen from them while not being able to score consistently.

The last game chosen, as alluded to in the introduction section, is "Montezuma's Revenge". This game was not tested in the original paper, hence why it was desirable to test it in this paper. The hope of this experiment is that the agent using intrinsic reward would be able to learn the game's mechanics much more quickly than the agent using extrinsic reward alone. The intuition is that the intrinsic reward agent would be rewarded for discovering new behavior, even if this behavior didn't directly result in extrinsic rewards, while the other agent would be restricted to seeking out sparse, extrinsic rewards alone.

Figure 7: Experimental Results on "Montuzema's Revenge"



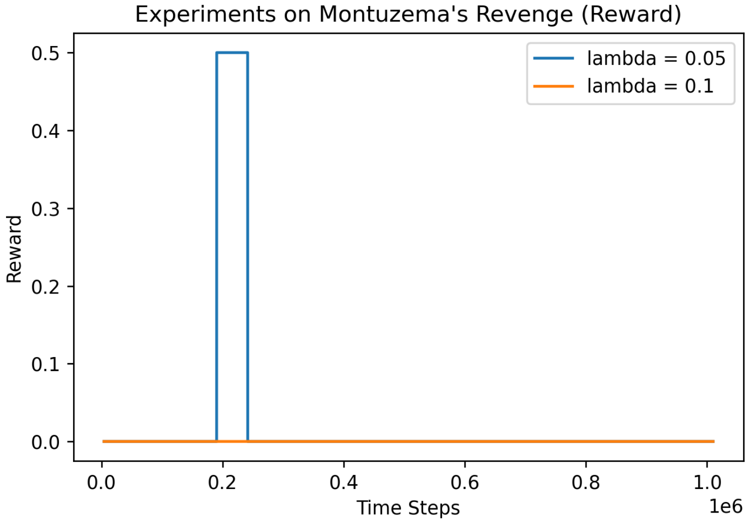


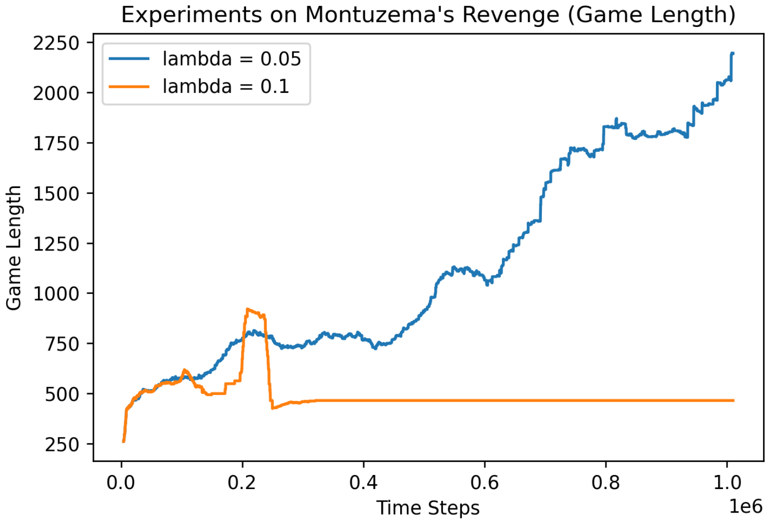
None of the models were able to find any long term success during the course of this experiment, with no model able to consistently score points. It should be noted that, in "Montezuma's Revenge" the first screen has a key worth 100 points. Since we are averaging over the last 100 games for each of these plots a value of 2 on the reward plot means that on 2 of the last 100 games the agent was able to gather the key. This behavior is rather strange, but perhaps indicative of the problems that arise with sparse reward.

Additionally, if the player opens the first locked door with the key they will be rewarded with 300 points, using the same logic as before we can conclude that none of these agents were able to escape the room otherwise they would have an average score of at least 4.

Strangely, for $\lambda = .05$ the results completely stagnated after around 150 thousand iterations. This implies that, at some point, the model stopped learning as it got stuck on some point of its optimization space with a gradient of 0. This performance was rather consistent across several random seeds, however it did occasionally learn to live for much longer while gaining no reward with some random seeds as seen in figures 8. With all this in mind, it is fairly clear that the algorithm did not fare well at this challenge at this time scale, however there are yet to be any Reinforcement Learning algorithm which can learn the game from scratch and play competently.

Figure 8: Additional Experimental Results on "Montuzema's Revenge"





Overall a big flaw of the "Montuzema's Revenge" experiments is a lack of visual game replay that could be analysed. The code written by the original authors does not allow for rendering the agent's game play and there was not enough time to find a solution to this problem for this paper. With a form of game replay one could see why the agent was only scoring occasionally or why it stagnated. Of particular interest would be the agent shown in figure 8 which is able to survive, but not score any points. It is possible the agent is simply standing still or taking random actions until they eventually lose rather than actually learning how to survive.

4 References

1. Zheng, Zeyu, Junhyuk Oh, and Satinder Singh. "On learning intrinsic rewards for policy gradient methods." Advances in Neural Information Processing Systems 31 (2018).