Effects of Random Reward Noise on Common Reinforcement Learning Methods

Daniel Dai
McGill University
daniel.p.dai@mail.mcgill.ca

Youssef Samaan
McGill University
youssef.samaan@mail.mcgill.ca

Abstract

In this paper, we analyze the performance of common reinforcement learning algorithms in environments with inherent noisy rewards. Specifically, we evaluate Q-learning, Expected SARSA, Deep Q-learning, and Double Deep Q-learning in Gym library environments of Cart Pole and Acrobot using both normal distribution and uniform distribution of reward noise with various levels of variance. Each member contributed equally with Youssef focusing more on the code and Daniel focusing more on the report.

1 Introduction

We aim to assess how different reinforcement learning techniques perform where rewards exhibit some degree of noise in environments from the Gym library. Our approach involves adding noise to the rewards to evaluate the algorithms' robustness. We compare the performance of two categories of reinforcement learning algorithms of tabular and neural networks. Specifically, we test tabular methods for Q-learning and Expected SARSA along with neural network methods for Deep Q-learning, and Double Deep Q-learning, implementing each algorithm from scratch. The choices of the categories and specific algorithms give us a good baseline to start analyzing the effects of noisy rewards with respect to the general types of reinforcement learning algorithms. The two Gym library environments we use are Cart Pole and Acrobot to give us a familiar testing area to be able to better control and analyze these baseline effects.

The motivation for this study stems from the observation that in some games and real-life scenarios, good actions may lead to unexpected outcomes, while bad actions may not be as severely punished. We seek to determine how these reinforcement learning techniques learn the optimal policy in such environments (see Section 2 for more details). While previous studies have explored the impact of noise such as on agent parameters [3], deep learning network weights [1], randomized rewards for multi-agents [4], or randomized value functions for efficient exploration with generalization [2] (see Section 3 for more details). Our focus is on adding small noise based on a random distribution directly to the rewards obtained from the environment. Specifically, we experiment with both normal and uniform distributions with various variances as noise. This approach allows us to specifically assess the algorithms' ability to find the optimal policy despite noisy reward signals.

2 Background

Reinforcement learning algorithms operate in dynamic environments where information such as rewards are often subject to a variety of different types of noise coming from a variety of sources. These sources are usually incredibly diverse and accounting for all of them is often infeasible. However, understanding how individual sources of noise could affect the performance of reinforcement learning algorithms could aid in better understanding the robustness of these algorithms in real-world scenarios. This is why in this study we focus on the effect of noise on the rewards of the environment which can be caused by many sources. These sources include things like general environment noise which in this case could contribute to the reward signal. Or randomized rewards which make the environment even more challenging given that the agent will not be able to get reliable feedback to train on. Other sources also include adversarial rewards which are rewards that are specifically designed to mislead the agent, leading to a higher degree of noise in the overall reward signal. In this paper, we more specifically focus on understanding general environment random noise although the idea is that the other sources of reward noises would also fundamentally apply in terms of the behavior of these algorithms. Understanding how reinforcement learning techniques perform under such conditions is essential for real-world applications where the environment is often noisy.

3 Related Work

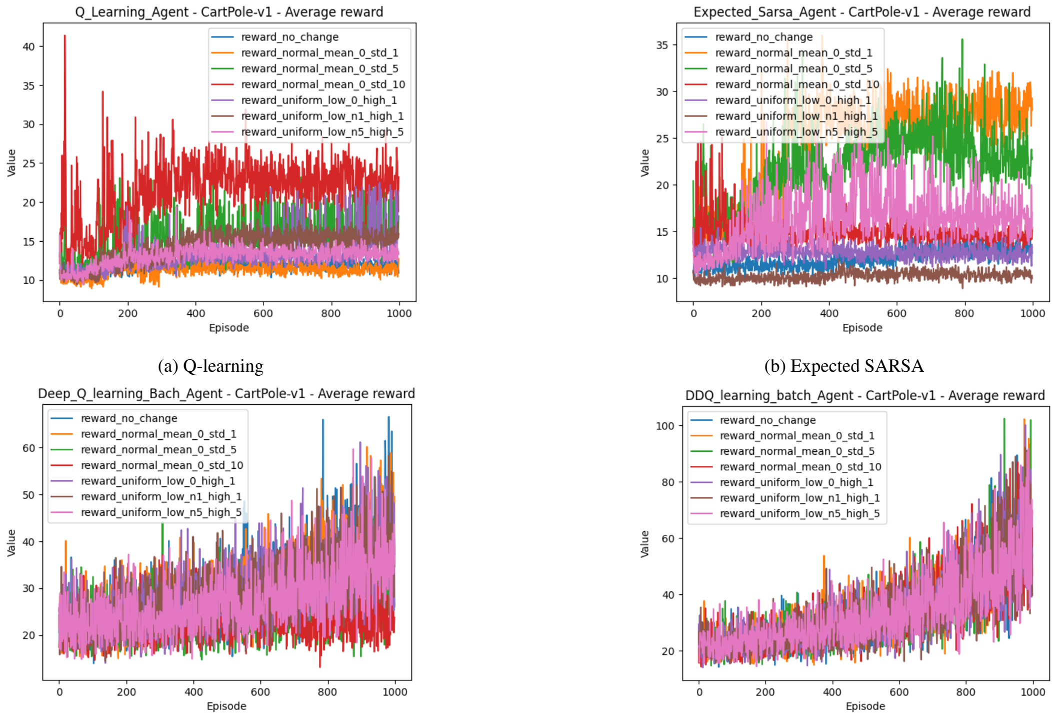
The idea of analyzing noise in either the environment or artificial noise added to the rewards is not new. Even within the idea of noise, there are different objectives such as adding noise to improve performance, analyzing ways of dealing with general environmental noise, or simply analyzing the effects of noise. In fact, the many studies that have been done on this subject are what inspired us to specifically focus on analyzing the effects of reward noise. An example of such a study is the paper Parameter Space Noise For Exploration [3] which focuses on directly adding noise to the actual parameters of the agent allowing alteration on a more fundamental level. By doing so, it hopes that the noise leads to more consistent exploration than traditional action space noise. A similar paper is NoisyNet [1] which instead focuses on adding noise to the weights of the neural network. The idea is that by adding noise to the weights, the network can learn to be more robust to perturbations in the weights. Another example is the paper Deep Exploration via Bootstrapped DQN [2] which explores randomized value functions for efficient exploration with generalization. In general, we see that these studies related to noise in reinforcement learning are often directly correlated with the consistency and robustness of the agent's decision-making process in the face of noise. In these studies, noise is usually added to some fundamental part of the agent to aid in the process of exploration which helps the agent counteract the inherent randomness in the environment.

There is a study that also deals with reward noise in the paper Reward-Randomized Policy Gradient [4] which focuses on adding noise to the rewards which allows multi-agent systems to explore more risky cooperation strategies. This is similar to our study but we focus more on understanding how the different types of noises affect the different reinforcement learning algorithms for single agents rather than focusing on the exploration or performance aspect.

4 Methodology

The fundamental process of this study relies on the actual distribution of noise we add to the rewards to then give to the agent to learn from, thereby simulating the random inherent reward noise that could be present in the environment. We decided to focus on two types of noise, specifically based on the normal and uniform distribution with various variances/ranges. The reason is that not only are these extremely common distributions in the real world but also we wanted to also see how the actual distribution of the noise affects the learning process of the agent. The two environments we chose were Cart Pole and Acrobot since these environments are relatively simple and inherently not supposed to be noisy stochastically speaking. This allows us to better control the noise we add to the rewards and better understand the effects of the noise on the learning process of the agent. Moreover, the difficulties of these environments are also different which would inform us more about these effects. For our choice of reinforcement learning algorithms, we decided to focus on two categories of algorithms, tabular and neural networks. For tabular, we chose Q-learning and Expected SARSA, and for neural networks, we chose Deep Q-learning and Double Deep Q-learning. These algorithms are widely studied and the difference in complexity of these algorithms also allows us to better understand the difference in how robust they are to random feedback. In general, we try to choose two for each point of comparison like environments and algorithms ensuring not to saturate all our time but to also have another reference for comparison. In the end, our methodology is based on the fact that in this study, we care more about trend consistency to be able to compare the algorithms and the different levels of noise rather than actual concrete performance statistics. This means our results will be more focused on the trends we see in the reward graphs as it will show us how the agent reacts to such random noises.

5 Experiment



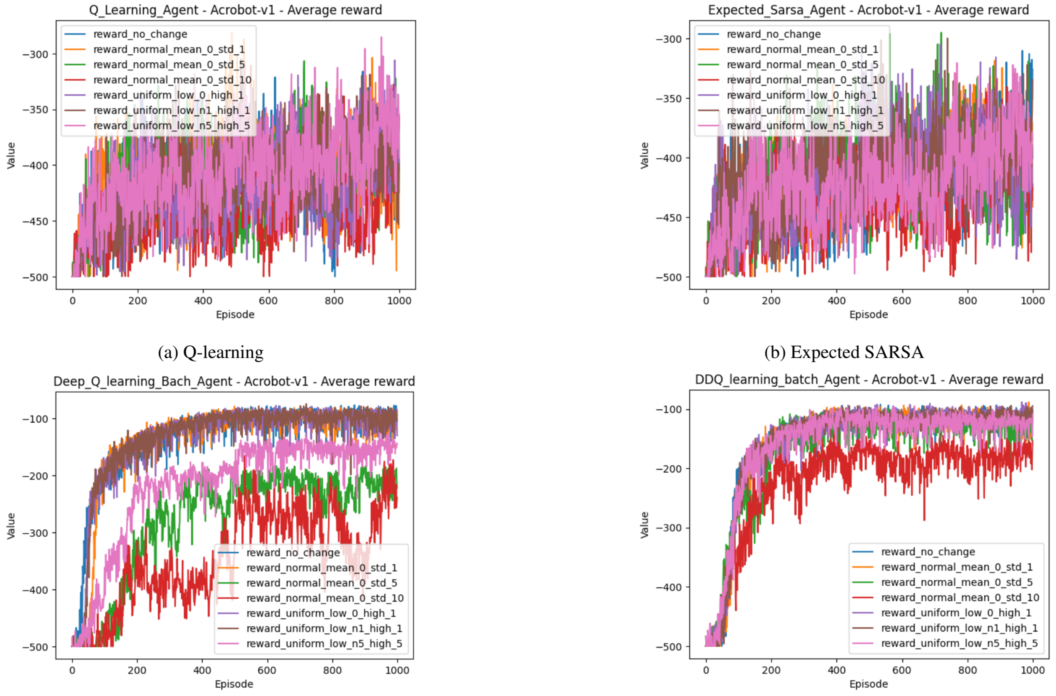
(a) Q-learning

(b) Expected SARSA

(c) Deep Q-learning

(d) Double Deep Q-learning

Figure 1: Average Return for Cart Pole Gym Environment



(a) Q-learning

(b) Expected SARSA

(c) Deep Q-learning

(d) Double Deep Q-learning

Figure 2: Average Return for Acrobot Gym Environment

For the tabular algorithms, we used; an epsilon-greedy strategy with an initial epsilon of 0.1, a decay decrement of 1e-5, and a minimum of 0.001; tiling with 2 tiles and 5 bins; gamma of 1; learning rate of 0.1; and weight initialization of uniform distribution from -0.001 to 0.001. For the neural network algorithms, we used; an epsilon-greedy strategy with an initial epsilon of 1, a decay decrement of 1e-5, and a minimum of 0.1; gamma of 0.99; learning rate of 0.0001; and batch learning with a batch size of 32 and memory size of 5000. The network consisted of 5 layers of [(Observation Space, 128),(128, 256), (256, 128), (128, 64), (64, Number of actions)] with ReLU activation function, an Adam optimizer and Mean Squared Error loss function. We use batch learning since it performs better than stochastic learning which will help identify trends more easily. For the reward noise distributions, we used a normal distribution with a mean of 0 and variances of 1, 5, and 10; and uniform distributions with ranges [0, 1], [-1, 1], and [-5, 5]. We trained each algorithm with each distribution setting with 1000 episodes averaging 10 independent runs on the two Gym environments of Cart Pole and Acrobot.

For Q-learning in Cart Pole (Figure 1a), we see that higher variance reward noises perform better with normal distribution with the highest variance performing around double with no reward noise. Given the nature of Q-learning using the max function, a higher variance led to higher Q values causing overestimation but in this specific environment, that overestimation seemed to pay off as the high rewards were promoting the optimal policy more. Another note is the normal distribution of 0 to 1 seemed to be a lot more unstable or in other words, randomly applying exclusively positive rewards seemed to subsequently show more random results. But in general, there was no significant difference between normal and uniform distribution. Expected SARSA on Cart Pole (Figure 1b) is similar to Q-learning except it seemed to do the best when the reward noise was small doing almost three times better than with no noise. Compared to Q-learning, given the nature of Expected SARSA, this would align with the fact that Expected SARSA is less sensitive given its greater averaging effects than Q-learning. As for the Acrobot environment (Figure 2a, 2b), this more complex environment resulted in the extra reward noises not having much of an effect, regardless of variance. In the Acrobat environment, any variance of reward noise didn't seem to have an effect.

For the neural network methods in the Acrobot environment (Figure 2c, 2d), we see that having no reward noise performs the best while reward noises with higher variance will perform worse. However, Double Deep Q-Learning (DDQ) (Figure 2d) handles higher deviated reward noises much better. This follows that DDQ inherently reduces overestimation bias where reward noise can greatly exacerbate this bias leading to sub-optimal policy decisions. In the Cart Pole environment (Figure 1c, 1d), any variance of reward noise didn't seem to have an effect.

6 Conclusion

The different variances of reward noises impacted these reinforcement learning methods in interesting ways. Q-learning seems to be sensitive to variance and performs better with higher variances. Though it is important to note that this could be a double-edged sword depending on the environment as it could greatly promote optimal actions it is possible it can also heavily promote sub-optimal actions. Expected SARSA is similar to Q-learning only for smaller variances. Both tabular methods did better with reward noise added possibly suggesting some exploration benefits with adding reward noise as studied in [4]. It also seems that reward noises for tabular methods don't matter as much for more complex environments as it struggled to even get decent results. The neural network reinforcement learning methods were a lot more robust to reward noise in both performance and stability as a consequence of neural networks being much more powerful. Along with that, Double Deep Q-learning was a lot more resilient to reward noises than Deep Q-learning. However, reward noise impacted these algorithms less when the environment was simpler.

7 Future Works

The main limiting factor of this paper was the limited resources and time. In the future, more algorithms, configurations, and environments should be tested to better understand the effects of reward noises. Especially analyzing the effects of environments with inherently noisy environments, though such environments are hard to test in the first place. This would also give a good opportunity to better understand the extent of reward sensitivity exhibited by the tabular algorithms. Finally as mentioned in Section 2, the effects of adversarial rewards could be interesting to research more.

References

[1] Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration, 2019.

[2] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn, 2016.

[3] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration, 2018.

[4] Zhenggang Tang, Chao Yu, Boyuan Chen, Huazhe Xu, Xiaolong Wang, Fei Fang, Simon Du, Yu Wang, and Yi Wu. Discovering diverse multi-agent strategic behavior via reward randomization, 2021.