Comparative Analysis of Q-Learning and DQN in Reinforcement Learning: Efficacy in Complex Environments and the Role of Replay Buffers

Fuzail Hamid Banday | 20408698 | Mechatronics Engineering | University of Nottingham

I. Introduction

Reinforcement learning (RL) has emerged as a pivotal technique in artificial intelligence, enabling algorithms to learn optimal behaviors through trial and error. [1]

This study focuses on two prominent RL methods: Q-Learning, a value-based approach, and Deep Q-Networks (DQN), which integrates a neural network into Q-Learning.

Q-learning is a model-free reinforcement learning algorithm where an agent learns to make decisions by interacting with an environment. It focuses on learning the value of taking certain actions in specific states, known as Q-values. Q-values are represented in a table along with the corresponding state and action pairs. After an agent has been trained the Q-table is used to find the highest Q-value for a particular state, and then the corresponding action is chosen.

DQN on the other hand uses deep neural networks to approximate the Q-function instead of a Q-table. While Q-learning is suitable for environments with discrete, small state spaces due to its reliance on a Q-table to store values, DQN excels in handling larger, more complex state spaces, including those with high-dimensional sensory inputs like images. The neural network takes the state as input and generates Q-values for all possible actions as output. However, the convergence of DQN is less assured compared to regular Q-learning due to non-stationarity issues caused by updates to the neural network. To mitigate these issues, techniques like experience replay and target networks are employed in DQN. [2]. In contrast to Q-learning, DQN isn't constrained by the need for discretizing the state space, as its neural network architecture can seamlessly handle continuous action spaces.

While both Q-Learning and DQN have demonstrated effectiveness in simpler environments such as CartPole, their adaptability and performance in more complex scenarios have not been extensively studied. This research aims to bridge this gap by examining these algorithms in the Acrobot environment, a notably more intricate scenario than CartPole. The following question was posed that needed to be addressed.

1. How does Q-Learning fare against DQN in environments with a higher degree of complexity than CartPole?
2. Does the integration of replay buffers in Q-Learning show similar benefits to DQN?

This study implements Q-Learning and DQN to solve Acrobot-v1 and CartPole and comapres key metrics in an attempt to address the above question. Furthermore, this study explores the incorporation of a replay buffer into the Q-Learning algorithm to see how it investigates how it effects the learning rate . Normally, Replay buffers are used with DQN primarily to improve learning stability and efficiency. They do this by storing past experiences and randomly sampling from them for training, which helps break correlations in sequential data.

II. Background

The foundational work of Patrick Sunden in "Q-Learning and Deep Q-Learning In OpenAI Gym CartPole Classic Control Environment" provides a comparative analysis of Q-Learning and DQN in a basic setting. While this research establishes a benchmark in simpler environments, it identifies a gap in understanding these algorithms' performance in more complex scenarios. Our study builds

upon this gap, extending the investigation to the Acrobot environment and furthering into by investigation implementation of replay buffers into the Q-Learning agent.

Furthermore, a novel aspect of our research involves the integration of replay buffers into the Q-Learning framework. This approach is inspired by the success of replay buffers in stabilizing and improving the learning process in DQN. By adopting this technique, typically reserved for more advanced deep learning-based algorithms, into Q-Learning, our study explores uncharted territory. We aim to investigate whether the inclusion of replay buffers can enhance the performance of Q-Learning in complex environments, addressing a notable limitation of traditional Q-Learning methods.

Further expanding the background of our study, the research presented in 'A Deeper Look at Experience Replay' offers pivotal insights into the nuances of experience replay in deep reinforcement learning.

III. Methodology

This study's methodology revolves around the development and evaluation of Q-Learning and DQN algorithms in two distinct environments: CartPole and Acrobot. The approach is designed to compare the performance of these algorithms in varying levels of environmental complexity

A. Implementation and Tuning in CartPole

Initially, a basic Q-Learning agent was set up for the CartPole environment to validate the findings of Patrick Sunden's research. The state space of CartPole, consisting of four continuous variables, was discretized into ten segments each, forming a discrete Q-table.

Following the setup, the Q-Learning agent underwent hyperparameter tuning to optimize learning efficiency. Parameters like learning rate, discount factor, and most important, epsilon decay rate, performance was then measured based on average rewards and episode lengths.

Simultaneously, the DQN algorithm was implemented for CartPole using ClearRL [3].

B. Adaptation to Acrobot Environment

The next phase involved adapting the Q-Learning model from CartPole to the Acrobot environment. This transition required significant modifications, including an update to the observation space to accommodate Acrobot's unique state variables and a reconfiguration of the Q-table to reflect the new state-action space. The discretization had to be reconsidered as well. For the angular velocities of the first and second links (t1 and t2), 20 and 40 segments were used for discretization, respectively. The remaining four state variables were each discretized into 10 segments. DQN experiments were conducted for the Acrobot again using CleanRL.

C. Replay Buffers in Q-Learning

A novel aspect of this study involves the integration of replay buffers into the Q-Learning framework. This integration aims to investigate the potential improvements in learning efficiency and stability, drawing on the benefits seen in DQN implementations. The replay buffer in Q-Learning stores and randomly samples past experiences, which is hypothesized to enhance the learning process.

D. Comparative Analysis

Throughout the experimentation process, both Q-Learning and DQN algorithms were evaluated for their learning efficiency, adaptability to environmental complexity, and overall performance in both the CartPole and Acrobot environments. This comparative analysis aimed to highlight the strengths and limitations of each algorithm in different contexts.

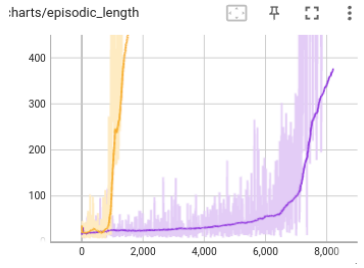
IV. Results

A. Cartpole

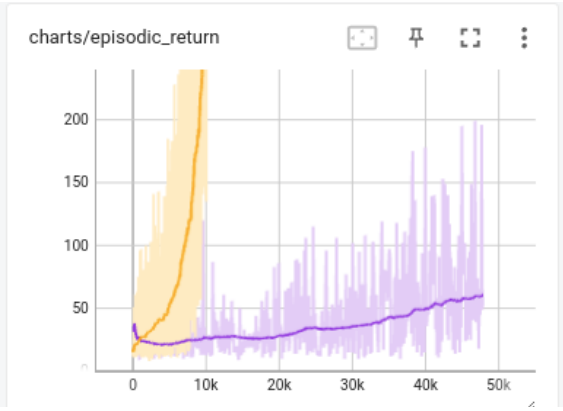
1) DQN

Yellow - After Tuning

Purple - Before Tuning

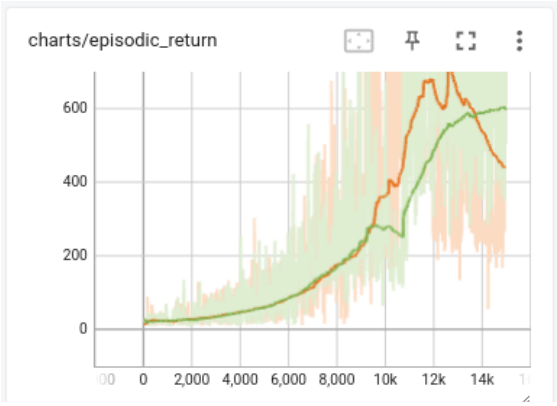


2) Q-Learning without replay buffer



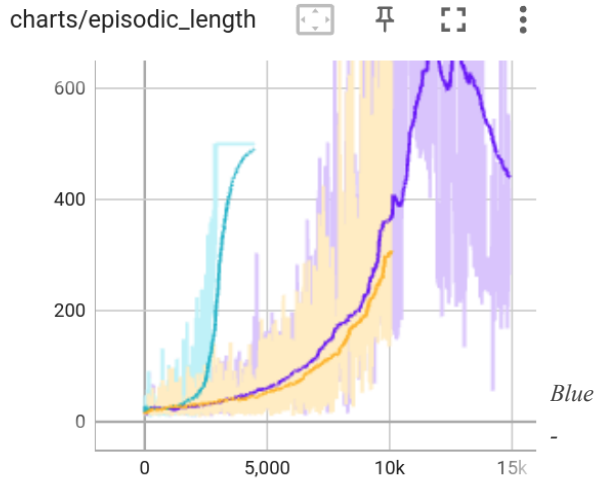
Yellow - After Tuning; Purple - Before Tuning

3) Q-Learning with replay buffer



Green - Batch Size = 10; Orange - Batch Size = 32

Final Graphs



Blue - DQN; Yellow - QL; Purple - QL with Buffer

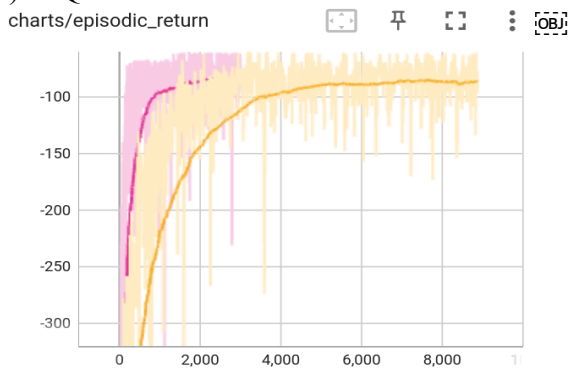
In the CartPole environment, DQN significantly outperformed Q-Learning, achieving the maximum reward rapidly within about 2500 episodes. In comparison, Q-Learning, without a forced termination mechanism, took three times longer to reach just half of DQN's maximum reward.

The implementation of an experience replay buffer in Q-Learning diQ-Learning did not show notable improvements Q-Learning did not show notable improvements d not show notable improvements in performance. Additionally, this integration considerably slowed down the learning process, it took much longer to get to the same 10,000 episodes.

These results highlight DQN's superior efficiency in the CartPole environment and raise questions about the practical effectiveness of replay buffers in Q-Learning under these specific conditions.

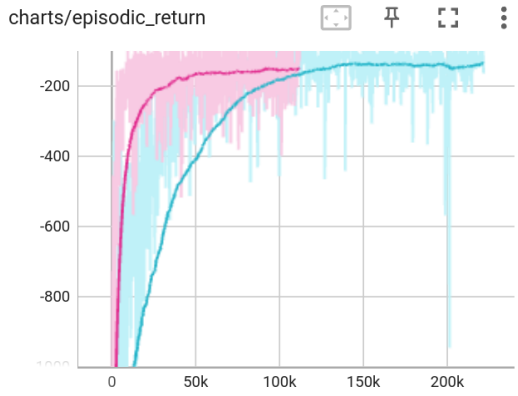
B. Acrobot

1) DQN



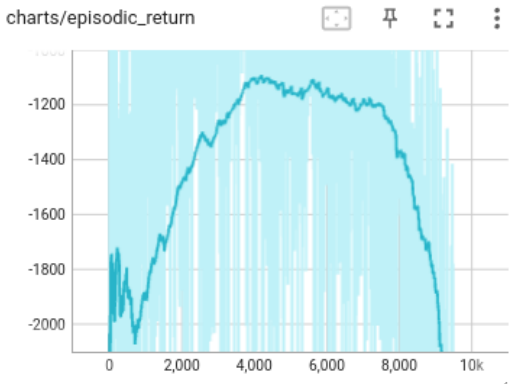
Yellow - After Tuning; Yellow - Before Tuning

2) Q-Learning without replay buffer



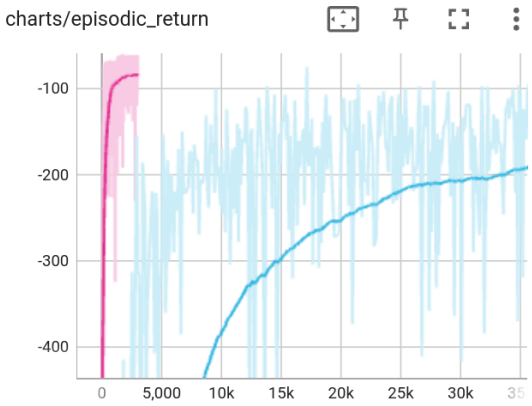
Pink - After Tuning; Blue - Before Tuning

3) Q-Learning with replay buffer



Blue - Batch Size = 10

Final Graphs



Pink - DQN; Blue - QL Without Buffer

In the Acrobot environment, DQN's efficiency was again prominent, learning to solve episodes in under 100 timesteps after only 2500 training episodes. In contrast, Q-Learning required about 150,000 episodes to reach a similar mean episodic reward, even after rigorous hyperparameter tuning.

The implementation of a replay buffer in Q-Learning yielded very weird results and needs further investigation. The learning was very slow with the buffer added, so much so that there wasnt enough time to study it prperly.

V. Discussion

The results from both the CartPole and Acrobot environments consistently demonstrate the superior efficiency of the DQN algorithm over Q-Learning. In the simpler CartPole environment, DQN rapidly achieved maximum rewards, whereas Q-Learning lagged significantly in terms of the number of episodes required to reach comparable performance levels. This disparity became even more pronounced in the Acrobot environment, where the complexity and challenge were higher. DQN adapted quickly to this environment, solving episodes in a fraction of the time taken by Q-Learning. These findings underscore the strength of DQN in quickly adapting to and mastering both simple and complex environments.

The integration of replay buffers into Q-Learning, an innovative approach in our study, yielded mixed results. In the CartPole environment, the addition of replay buffers did not translate into a significant performance improvement for Q-Learning. Moreover, it introduced a notable decrease in computational efficiency, slowing down the learning process significantly. In Acrobot, replay buffers made the simulation so slow that it was not possible to run enough studies and draw conclusion. This outcome suggests that the benefits of replay buffers, well-established in the context of DQN, may not directly apply to Q-Learning.

Given the mixed results with the replay buffers in Q-Learning, future research could explore optimizing the implementation of replay buffers. Additionally, further studies could delve into understanding the underlying reasons for the varying impacts of replay buffers in different environments and algorithmic setups.

This study contributes to the understanding of how Q-Learning and DQN perform in reinforcement learning environments of varying complexities. While DQN shows clear advantages in terms of learning efficiency, the exploration into enhancing Q-Learning with replay buffers opens up avenues for further research and optimization in the field of reinforcement learning.

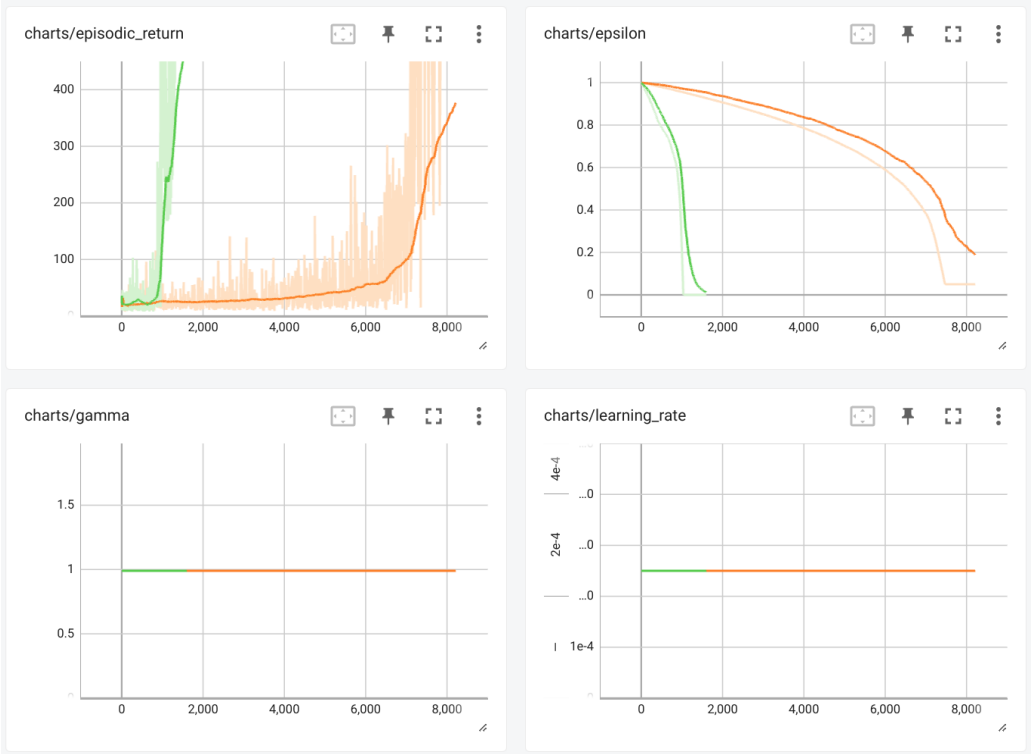
References

1. Jang, Beakcheol, Myeonghwi Kim, Gaspard Harerimana, and Jong Wook Kim. "Q-learning algorithms: A comprehensive classification and applications." IEEE access 7 (2019): 133653-133667.
2. Guide, Step, and Ankit Choudhary. 2023. “Deep Q-Learning | An Introduction To Deep Reinforcement Learning.” Analytics Vidhya. https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/.
3. Huang, Shengyi, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G. M. Araújo. “CleanRL: High-Quality Single-File Implementations of Deep Reinforcement Learning Algorithms.” Journal of Machine Learning Research 23, no. 274 (2022): 1–18. http://jmlr.org/papers/v23/21-1342.html.
4. Zhang, Shangtong, and Richard S. Sutton. “A Deeper Look at Experience Replay.” CoRR abs/1712.01275 (2017). http://arxiv.org/abs/1712.01275.
5. Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." In Proceedings of the AAAI conference on artificial intelligence, vol. 30, no. 1. 2016.
6. Liu, Ruishan, and James Zou. "The effects of memory replay in reinforcement learning." In 2018 56th annual allerton conference on communication, control, and computing (Allerton), pp. 478-485. IEEE, 2018.
7. Kiran, Mariam, and Melis Ozyildirim. "Hyperparameter tuning for deep reinforcement learning applications." arXiv preprint arXiv:2201.11182 (2022).
8. Eimer, Theresa, Marius Lindauer, and Roberta Raileanu. "Hyperparameters in Reinforcement Learning and How To Tune Them." arXiv preprint arXiv:2306.01324 (2023).

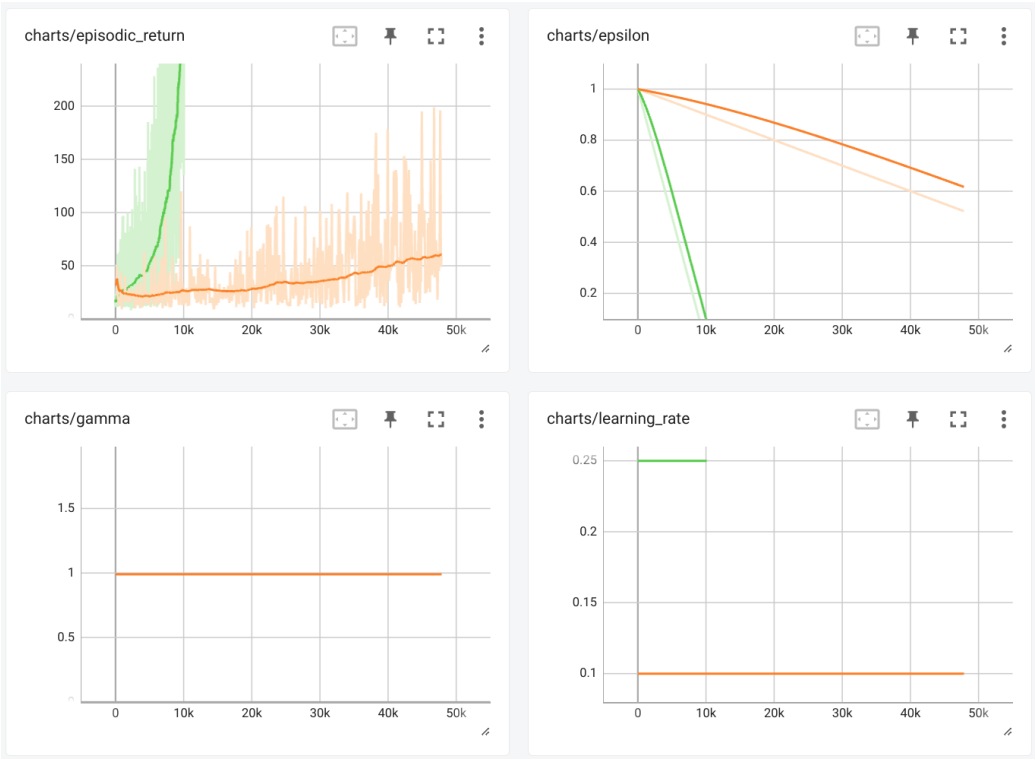
Appendix

Full results of all experiments:

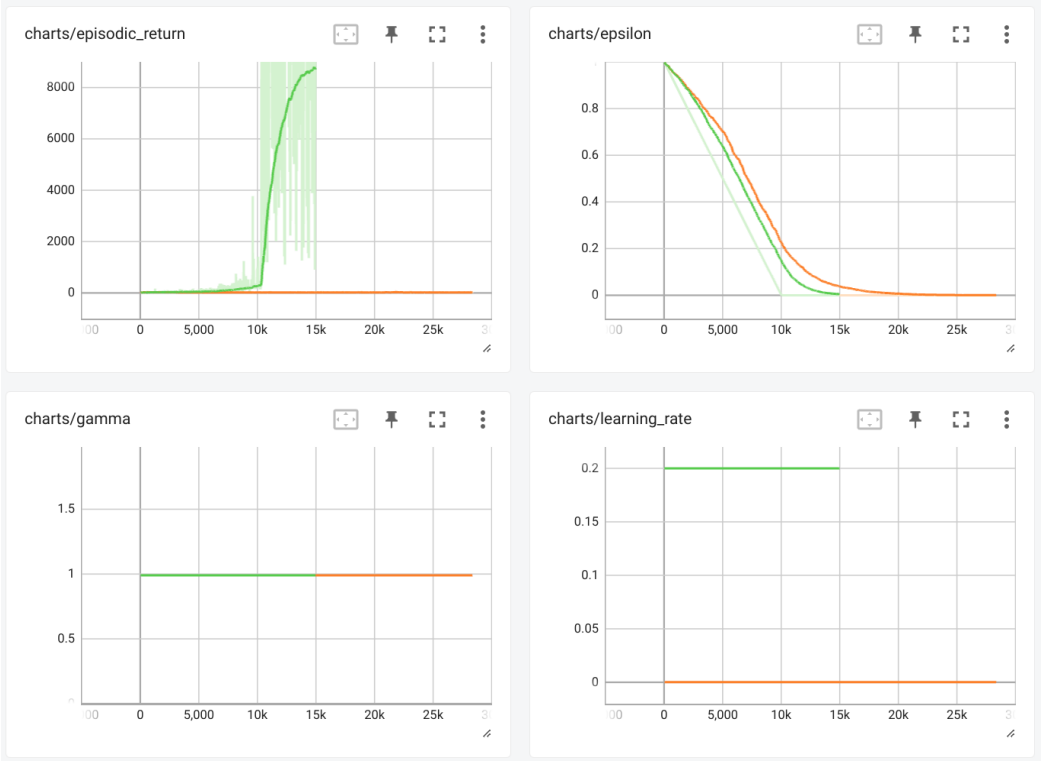
1) Cartpole DQN (Green: After tuning | Orange: Before Tuning):



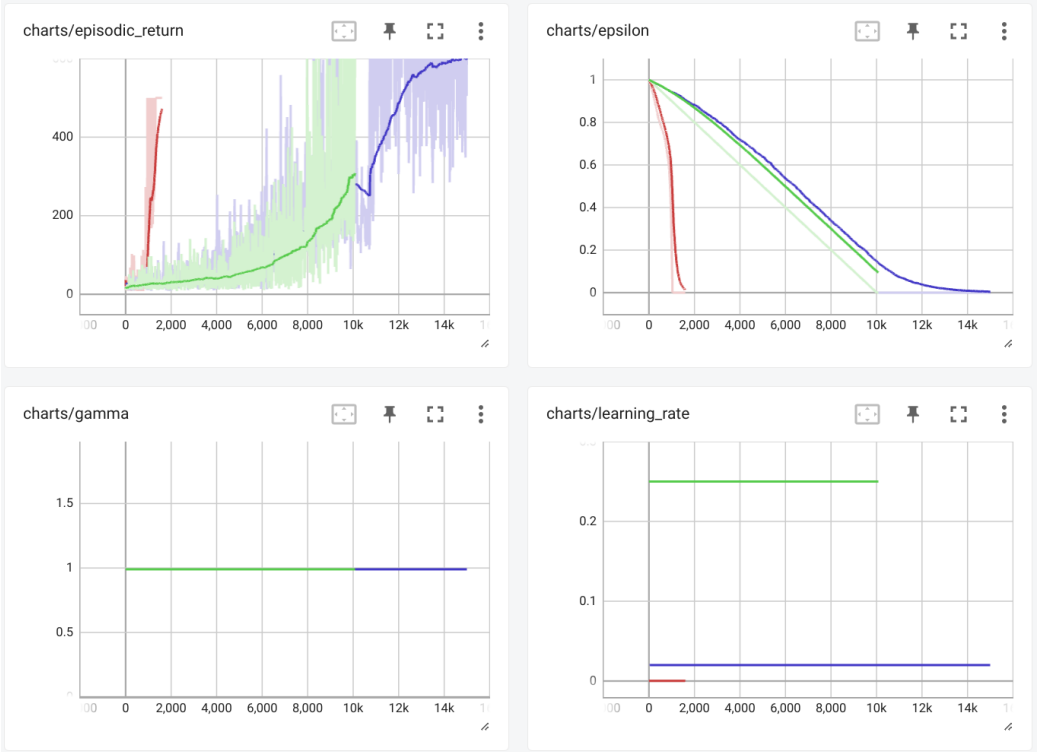
2) Cartpole Q-Learning without replay buffer (Green: After tuning | Orange: Before Tuning):



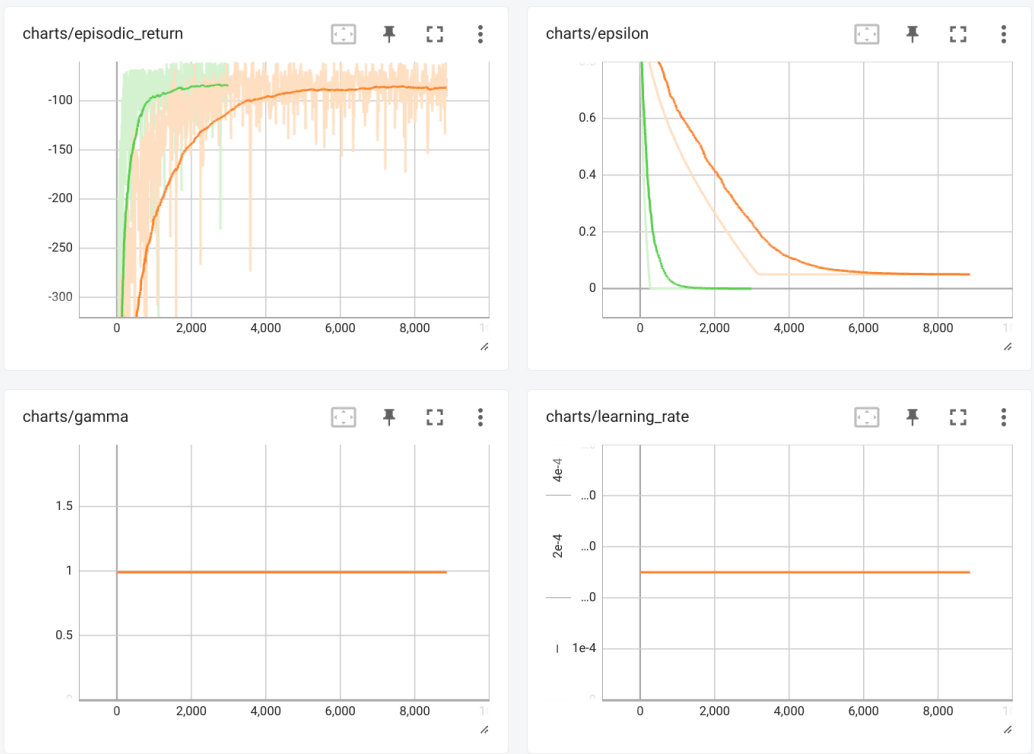
3) Cartpole Q-Learning with replay buffer (Green: After tuning | Orange: Before Tuning):



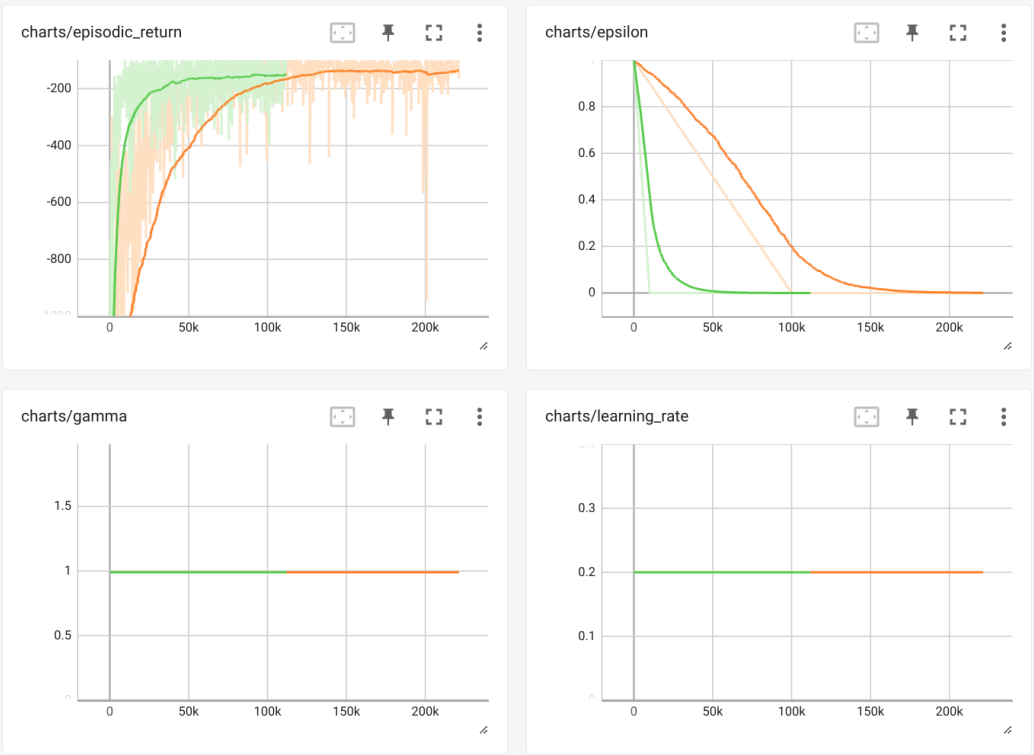
4) Combined Graphs (Red: DQN | Green: QL | Blue: QL with replay buffer):



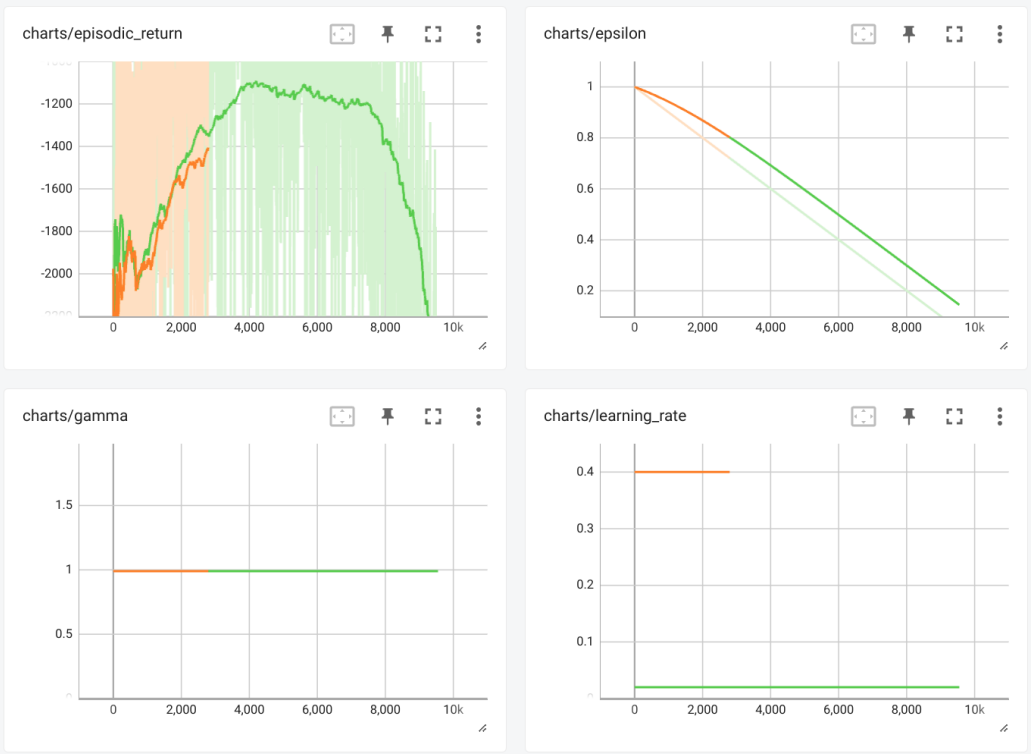
5) Acrobot DQN (Green: After tuning | Orange: Before Tuning):



6) Acrobot Q-Learning without replay buffer (Green: After tuning | Orange: Before Tuning):



7) Acrobot Q-Learning with replay buffer (Green: After tuning | Orange: Before Tuning):



8) Combined Graphs (Blue: DQN | Red: QL | Green: QL with replay buffer):