Team GPT4 - Lunar Lander: DQN Report

Charankamal Brar, Johan Hernandez, Brittney Jones, Lucas Perry, Corey Young

University of Washington, *Tacoma*

TCSS 435: Artificial Intelligence

Professor Teredesai

May 7, 2025

Overview of DQN

Deep Q-Network (DQN) is a value-based reinforcement learning method that combines Q-learning with deep neural networks to approximate the optimal action-value function. It allows agents to make decisions in high-dimensional environments by learning from their interactions and optimizing the Q-values.

Description of chosen extension

We implemented Double DQN to reduce the overestimation bias in standard DQN. In standard DQN, the same network selects and evaluates the action that maximizes the Q-value. Double DQN decouples this process using the main network to select the action and the target network to evaluate its value. We conducted experiments with both variants under identical conditions to analyze the effectiveness of Double DQN in reducing overestimation bias, overestimation bias is a known limitation of standard DQN. The results are visualized using Matplotlib and Seaborn. Plots compare reward curves and success rates over episodes for both DQN implementations.

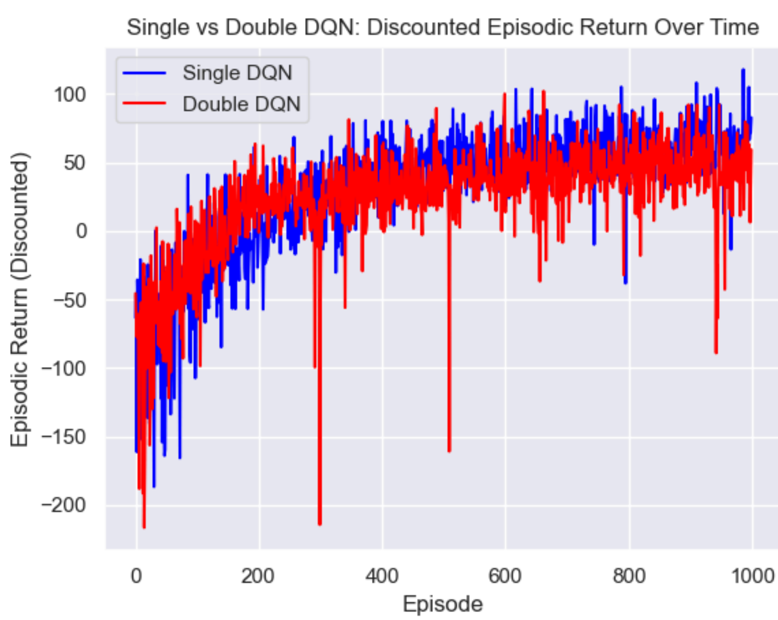
This leads to more stable learning and improved policy performance, particularly in environments with high reward variance.

Experimental results and insights

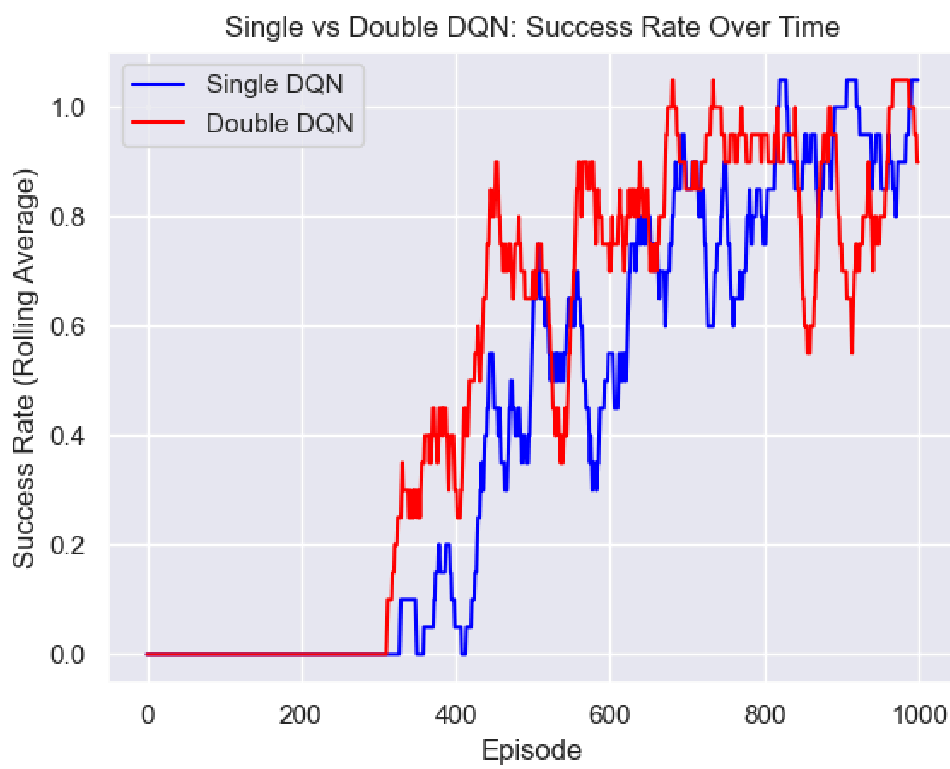
Training Plots:

Episodic Reward vs. Episode Number

Episodic Return vs. Episode Number



Landing Success Rate over Time (Episode)



Metrics Table (last 100 episodes average):

Double DQN Summary (Last 100 Episodes)

Metric	Value
Success Rate	0.83
Mean Reward	214.87
Median Reward	240.86
Mode Reward	238.08
Std. Dev. of Reward	90.44
Highest Reward	302.71
Lowest Reward	-152.61

Single DQN Summary (Last 100 Episodes)

Metric	Value
Success Rate	0.91
Mean Reward	235.31
Median Reward	249.06
Mode Reward	286.15
Std. Dev. of Reward	62.25
Highest Reward	295.22
Lowest Reward	1.08

Contributions

Charankamal Brar

Implemented the combined plots for Episodic Reward vs. Episode Number, Episodic Return vs. Episode Number, and Landing Success Rate over Time (Episode) to see the performance difference between standard DQN and Double DQN.

Johan Hernandez

Implemented circular replay buffer to assist the agent in learning from previous experience. Implemented the boilerplate for the QNetwork class. Added typing hints throughout various parts of the project to enforce type safety and code readability.

Brittney Jones

Core DQN Architecture Implemented a QNetwork class with two hidden fully connected layers (fc1_dims, fc2_dims) to approximate Q-values. Agent Implementation Wrote a DQNAgent class to Initialize both local and target networks. Use an epsilon-greedy policy for exploration vs. exploitation. Handle experience replay and learning updates using mini-batches sampled from memory.

Lucas Perry

Designed and formatted the .pdf. Analyzed dqn_agent, q_network, replay_buffer, and train_logger and fine tuned the documentation for readability. Made sure the report accurately reflected the contributions of the group.

Corey Young

Implemented Double DQN. Added functionality to choose which DQN model to run (single, double, or both). Developed a way to retrieve summarizing statistics. Designed summary table. Added dynamic labeling to plots.

Link to our GitHub repository

https://github.com/johan253/lunar-landing

References

Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533. https://doi.org/10.1038/nature14236

Silver, D. (n.d.). Reinforcement learning course (UCL). University College London. https://www.davidsilver.uk/teaching/

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press. https://incompleteideas.net/book/the-book-2nd.html

Van Hasselt, H., Guez, A., & Silver, D. (2016, March). Deep reinforcement learning with double q-learning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 30, No. 1).