Q-learning

u1541147 Tseng,Sheng Hung, u1527521 Chen,Kuan Yu

2025,2,19

Part 1. Learning Curve and Performance Comparison

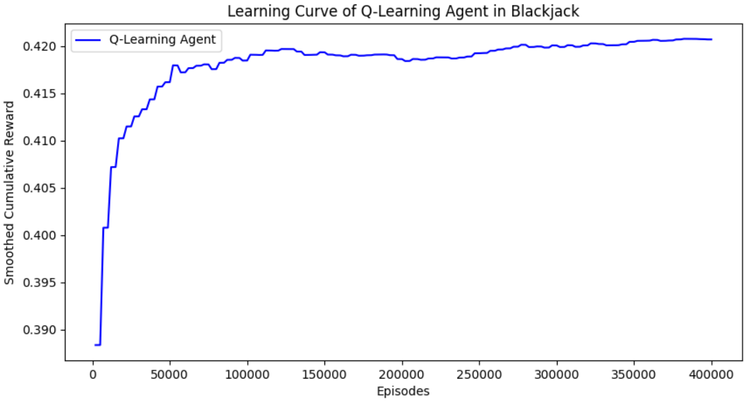

Figure 1: Learning Curve of the Q-Learning Agent in Blackjack

As shown in Figure 1, the x-axis is the number of episodes (up to approximately 400,000), while the y-axis shows the **smoothed cumulative reward** over time.

- Early in training, the cumulative reward starts relatively low (around 0.39) and rises to around 0.42 as training continues.
- Although the curve is somewhat noisy, there is a clear upward trend that eventually stabilizes.
- This gradual increase and eventual plateau indicate that the **agent is steadily improving its policy** (by refining its Q-values) and converging to a more optimal strategy.

Q-Learning vs. Random Policy

Figure 2 shows a comparison between the Q-Learning agent and a purely random policy:

- The x-axis represents the number of episodes (up to around 400,000), and the y-axis is the agent's **win rate** measured over a rolling window of 1000 episodes.
- **Q-Learning Policy (blue)**: The win rate generally fluctuates but remains around 0.42–0.44, indicating the Q-Learning agent consistently outperforms random play.
- **Random Policy (orange)**: The random policy's win rate hovers between 0.37–0.39, showing significantly weaker performance.

This demonstrates that **Q-Learning** yields a better long-term success rate than random actions, confirming that the agent is learning to make more effective decisions as training progresses.

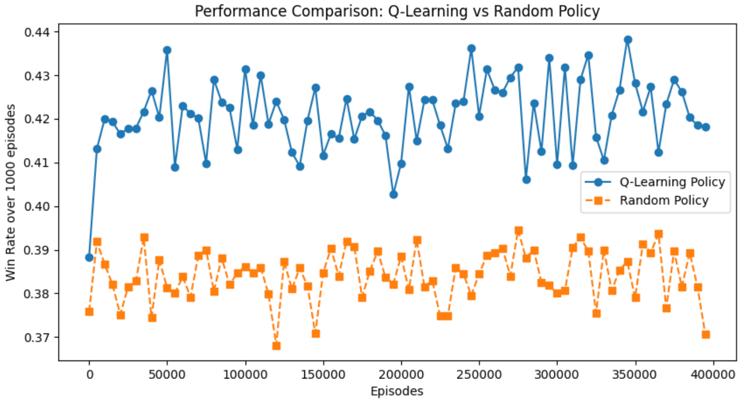

Figure 2: Performance Comparison: Q-Learning vs. Random Policy

Overall Findings

- **Monotonically Improving Performance:** The Q-Learning agent exhibits a learning curve that, despite some noise, shows an upward trend in cumulative reward. This implies that the agent's actions become progressively better over time.
- **Comparison with Random Policy:** The Q-Learning agent's performance (in terms of win rate and cumulative reward) is consistently higher than that of a purely random policy. This confirms that learning is occurring and that the agent's decision-making is guided by the Q-values rather than chance.
- **Convergence:** The smoothed reward eventually plateaus, suggesting the agent has reached a reasonably stable policy under the given state/action space and reward structure.

In summary, these results illustrate that Q-Learning is effective in this Blackjack setting, showing higher win rates and cumulative rewards than random play and demonstrating a clear learning trajectory as the agent's policy improves with experience.

Part 2: Landing on the Moon using DQN

Figure 3 shows the training curve of our DQN agent on the Lunar Lander environment. The x-axis represents the training episode index (from 1 to 1000), while the y-axis shows the total reward per episode. Although the rewards start off quite low (often negative), they steadily improve over time, indicating that the agent is learning to stabilize its landings and avoid crashes.

Performance Comparison: DQN vs. Random Policy

To further demonstrate learning, Figure 4 compares the final DQN policy with a purely random policy. Each box plot shows the distribution of total rewards across multiple evaluation episodes. The random policy typically achieves negative rewards (often corresponding to crashes), whereas the DQN policy consistently achieves much higher rewards.

Findings and Discussion

- **Evidence of Learning:** The DQN learning curve clearly shows an upward trend in total rewards, indicating that the agent's policy is steadily improving over time.

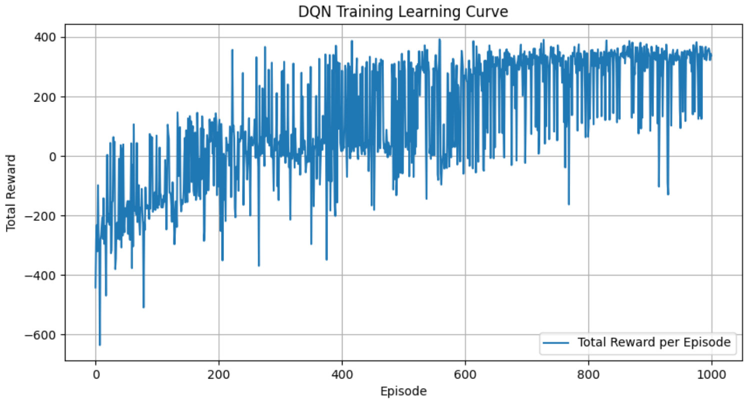

Figure 3: DQN Training Learning Curve: Total Reward per Episode over 1000 Episodes

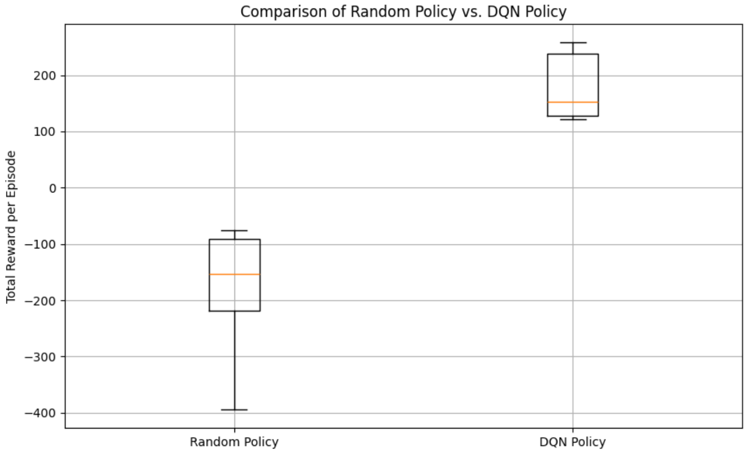

Figure 4: Comparison of Random Policy vs. DQN Policy: Total Reward per Episode

- **Comparison with Random Policy:** The DQN policy outperforms random actions by a significant margin. The box plot highlights that the DQN agent's rewards are much higher on average and have less variance.
- **Performance Relative to Humans:** While some human players may be able to land smoothly, the trained DQN agent can often achieve comparable or better performance in terms of consistent, successful landings.

In conclusion, the DQN agent learns to control the lunar lander effectively, surpassing the performance of a random policy. The training curve and box plot together provide strong evidence that the policy improves over time, ultimately achieving stable and higher-reward landings.

Extra Credit 1

In this experiment, we compare two popular exploration strategies in Q-Learning for Blackjack:

- **ϵ-greedy**, where the agent selects a random action with probability ϵ and otherwise selects the greedy action according to its current Q-values.
- **Boltzmann (Softmax) exploration**, where the probability of selecting an action is proportional to $\exp(Q(s,a)/T)$, with T denoting the "temperature" parameter.

We vary the initial ϵ (for ϵ-greedy) and the initial temperature T (for Boltzmann) and observe the learning performance.

Epsilon-Greedy Exploration

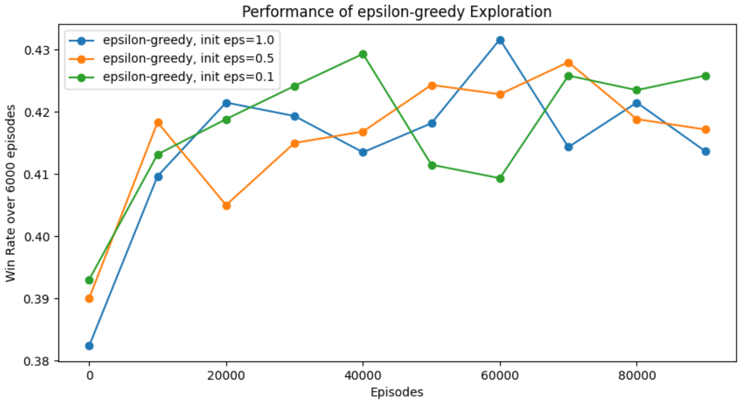

Figure 5: Performance of ϵ-greedy Exploration with Different Initial ϵ

Figure 5 shows the win rate (over a rolling window of 60,000 episodes) when using ϵ-greedy exploration with initial $\epsilon \in \{1.0, 0.5, 0.1\}$. We can draw a few observations:

- **High initial ϵ (e.g., 1.0):** The agent explores aggressively at the start. While this can slow initial learning, it often leads to better coverage of the state-action space and potentially higher final performance.
- **Medium initial ϵ (0.5):** This strategy provides a balance between exploration and exploitation, leading to a relatively smooth increase in win rate.
- **Low initial ϵ (0.1):** The agent relies more on exploitation from the beginning, potentially reaching a decent policy quickly but risking suboptimal exploration.

Boltzmann (Softmax) Exploration

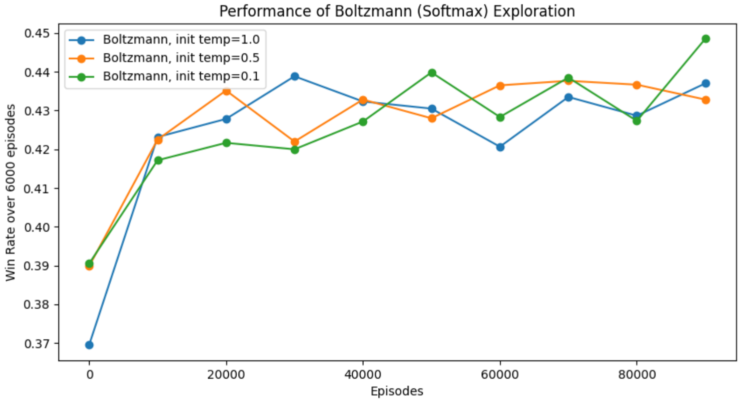

Figure 6: Performance of Boltzmann (Softmax) Exploration with Different Initial Temperature T

In Figure 6, we compare the win rate using Boltzmann (Softmax) exploration with initial temperature $T \in \{1.0, 0.5, 0.1\}$. Here:

- **High temperature ($T = 1.0$):** The agent more frequently explores less-favored actions early on. This can help avoid local optima, but it might also delay convergence.
- **Moderate temperature ($T = 0.5$):** Balances exploration and exploitation; the policy still occasionally tries suboptimal actions, but is more likely to pick actions that have higher estimated Q-values.
- **Low temperature ($T = 0.1$):** The agent is heavily biased toward actions with higher Q-values, often converging faster but risking less exploration.

Findings and Conclusions

- **Overall Performance:** Both ϵ-greedy and Boltzmann exploration can achieve competitive results in Blackjack.
- **Impact of Parameters:**
  - With ϵ-greedy, a higher initial ϵ helps ensure thorough exploration, often leading to better final policies, though initial training can be slower.
  - With Boltzmann exploration, a moderate temperature usually provides a good balance. Very high or very low temperatures can harm performance by either exploring too broadly or not exploring enough.
- **Practical Implications:** The best choice of exploration method and parameters can depend on the size and complexity of the state-action space, as well as the desired trade-off between convergence speed and final performance.

In summary, both methods outperform purely random play in Blackjack, but the choice of hyperparameters (i.e., ϵ or temperature) significantly affects the learning trajectory and final policy quality.

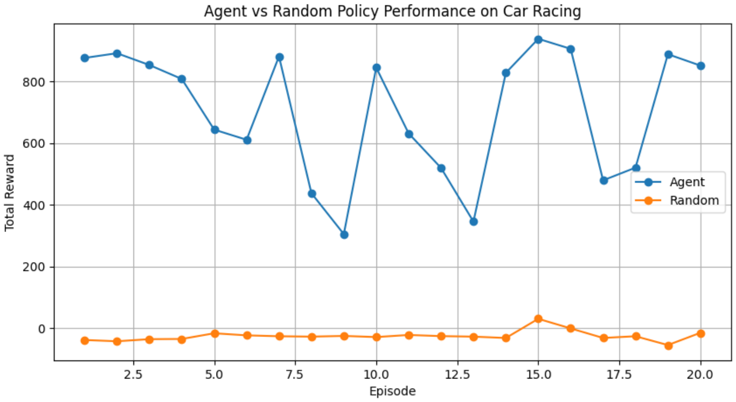

Figure 7: Agent vs. Random Policy Performance on Car Racing

Extra Credit 2

Agent vs. Random Policy Performance on Car Racing

Figure 7 compares the performance of the trained DQN agent against a purely random policy. The x-axis denotes the episode number, while the y-axis shows the total reward accumulated within each episode. Several observations can be made:

- The **DQN Agent** consistently achieves higher rewards than the random policy.
- There is some variability in the agent's rewards across episodes (due to stochasticity in the environment and exploration), but the overall performance remains well above that of the random policy.
- The random policy's rewards cluster around low or even negative values, reflecting the lack of control and frequent off-track driving or collisions.

Learning Curve for DQN on Car Racing

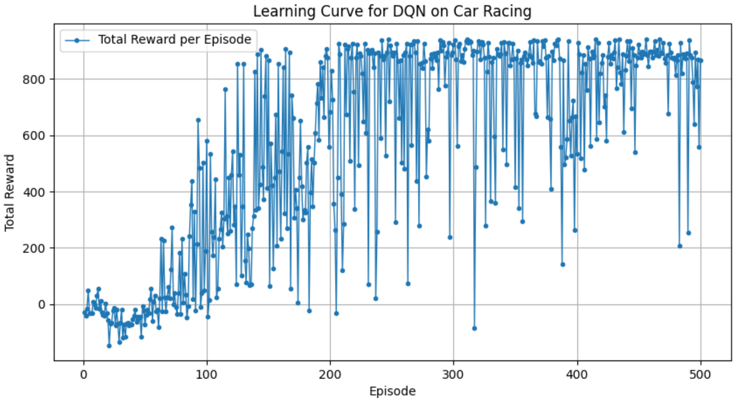

Figure 8: Learning Curve for DQN on Car Racing

Figure 8 shows how the total reward per episode evolves over the training process. As training progresses:

- Early episodes often yield low or negative rewards, indicating that the agent has not yet learned effective control strategies.

- Over time, the agent's performance improves, as evidenced by the general upward trend in rewards.
- Despite occasional dips (likely caused by continued exploration or challenging track conditions), the overall performance steadily increases, showing the DQN agent is learning a more effective policy.

Summary of Findings

- The **DQN Agent** significantly outperforms a random policy, accumulating higher rewards and demonstrating better control in Car Racing.
- The learning curve indicates that the agent's performance improves over time, eventually stabilizing at a consistently higher reward level.
- This suggests that Deep Q-Learning is an effective approach for this environment, enabling the agent to navigate the track more skillfully and avoid penalties.