Q-Learning for Blackjack: Comparing Simple and Full Action-Space Agents

Ziyan Lai

Abstract—This project explores training reinforcement learning agents to play Blackjack using Q-learning. Two types of agents are developed: a simple agent with two actions (hit or stand), and an advanced agent with five actions (hit, stand, double down, surrender, and split). Their performance is compared against a rule-based basic strategy using metrics such as expected reward, win rate, and reward variance. Results show that both Q-learning agents outperform the basic strategy in terms of expected reward, and that access to additional actions significantly improves the agent's mean-variance performance.

1 INTRODUCTION

In this study, I trained two agents using Q-learning to play the game of Blackjack. One agent is restricted to two actions—hit or stand—while the other can choose from all five typical actions: hit, stand, double down, surrender, and split. Their performance is evaluated and compared with a basic strategy agent, which follows a fixed rule set without learning. Key performance metrics such as expected reward, win rate, and standard deviation are used for comparison.
Ultimately, the Q-learning agents—despite having no prior knowledge of the game—were able to match or exceed the performance of the rule-based baseline. The agent with full action access demonstrated the best overall performance, especially in reward expectation, while maintaining stable variance.

2 AGENTS

We describe the three agent types and how they were developed to play Blackjack:

- Basic Strategy Agent: A static agent that follows fixed rules and does not learn. It serves as a performance benchmark

- Hit-or-Stand Agent: A learning agent trained using Q-learning, limited to only hit or stand decisions
- All-Actions Agent: A more complex Q-learning agent that can take all five possible actions, though certain actions like double down, surrender, and split are only available on the first move

2.1 Q-learning

Q-learning is a model-free reinforcement learning algorithm that teaches an agent how to act by learning the value of actions in different situations. It uses a Q-table to estimate the expected cumulative reward for each (state, action) pair.

At each step, the agent updates the Q-value based on the reward received and the estimated value of the next state:

$$Q_{new}(S_t, A_t) := (1 - \alpha) \cdot Q_{old}(S_t, A_t) + \alpha \cdot [r_{t+1} + \gamma \cdot max_a(S_{t+1}, a)]$$

where S is the state, A the action, α the learning rate, r the reward, and γ the discounting factor. Over time, by balancing exploration (random actions) and exploitation (choosing the best-known action), the agent converges to an optimal policy—without prior knowledge of the game rules.

2.2 Basic Strategy Agent

The basic strategy agent supports three actions: hit, stand, and double down. It follows the simple decision rules below:

- Hit when hand value is 12–16 and the dealer's face-up card is 7 or higher
- Stand when hand value is 17 or higher
- Double down when hand value is 10 or 11 and the dealer's face-up card is 9 or lower

2.3 Hit-or-Stand Agent

This agent is trained using Q-learning but can only perform two actions: hit or stand. It learns through millions of simulated hands and updates its policy based on cumulative rewards.

2.4 All-Actions Agent

This Q-learning agent is trained with the full set of Blackjack actions: hit, stand, double down, surrender, and split. Context-specific constraints are enforced—e.g., double down, surrender, and split are only allowed on the first round of a hand. This agent explores a larger decision space and is expected to capture more nuanced strategies.

2.5 Evaluation Metrics

The following metrics are used to cross compare the performances of the above three agents:

Expectation of reward, standard deviation of reward, win rate, tie rate, and lose rate

In game theory and finance, the mean-variance tradeoff is a core principle: we aim to maximize expected return while minimizing risk (variance). This tradeoff is especially relevant here—an agent might win more often but with greater volatility, or lose more often but in a smarter, more controlled way!

It’s also worth noting that Q-learning doesn’t optimize for win rate—it optimizes for expected rewards. For example, surrendering may decrease the chance of winning, but it’s often the optimal choice to preserve capital. Still, win rate is included as an intuitive benchmark against the basic strategy.

3 RESULTS

Each agent was trained independently by playing 50 million hands, allowing it to optimize its Q-table through reinforcement learning. The results are presented in two scenarios: when the agent holds a soft hand and when it holds a hard hand. A soft hand refers to any hand that contains an Ace counted as 11 without causing a bust.

3.1 Hit-or-Stand Agent

When the agent is restricted to only hitting or standing, the resulting strategy is intuitive. For soft hands (Figure 1), the agent plays more aggressively—continuing to hit until the hand value exceeds 18—since the risk of busting is lower. In contrast, for hard hands (Figure 2), which are less flexible, the agent adopts a more conservative approach and tends to stand earlier. The decision to hit or stand also depends on the dealer's upcard, as the agent adjusts its risk based on the perceived threat from the dealer's potential hand.

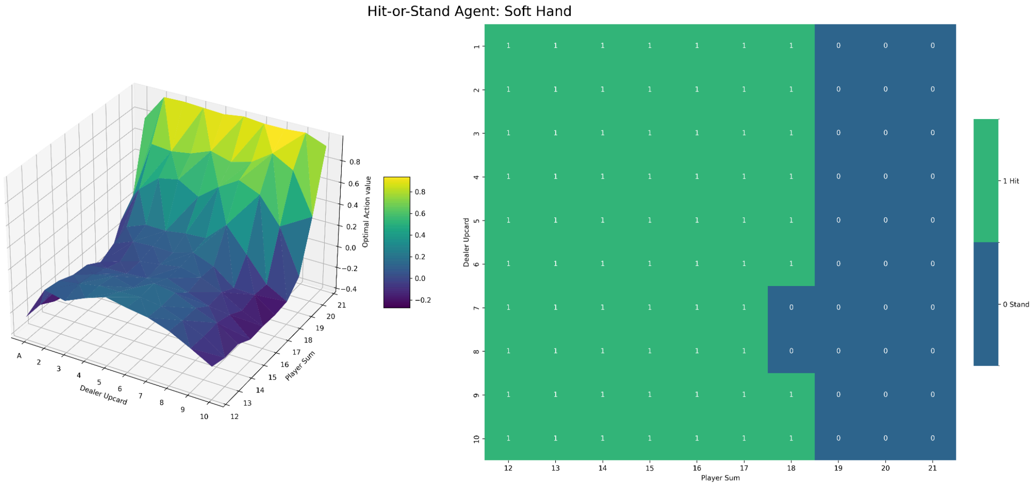


Figure 1—Hit-or-Stand Agent's optimal Q-value (left) and optimal action (right) for soft hands.

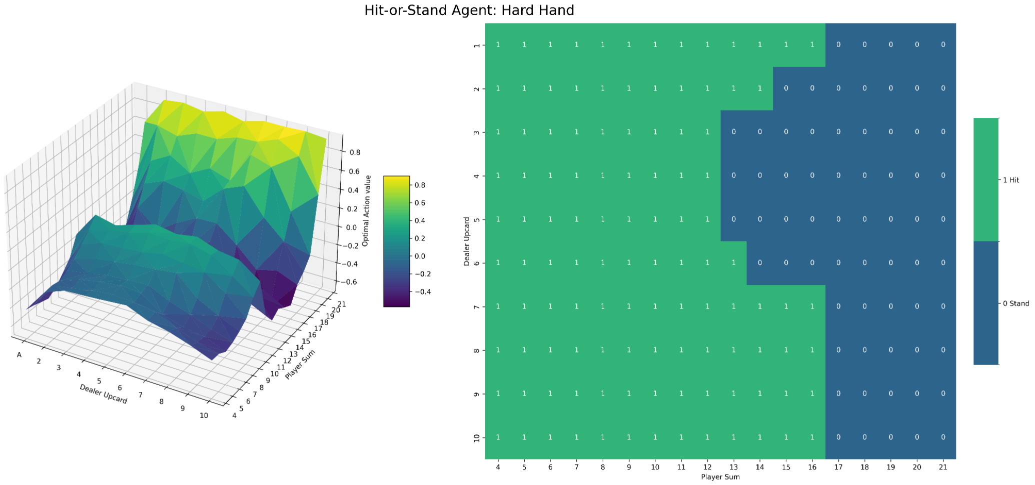


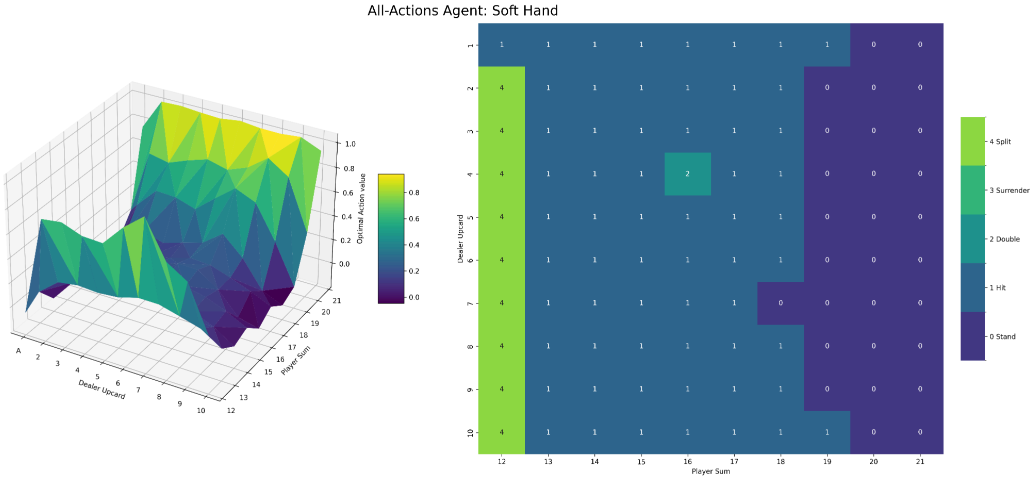
Figure 2—Hit-or-Stand Agent's optimal Q-value (left) and optimal action (right) for hard hands.

3.2 All-Actions Agent

When the agent is allowed to take additional actions—such as splitting, surrendering, or doubling—the strategy becomes more nuanced.

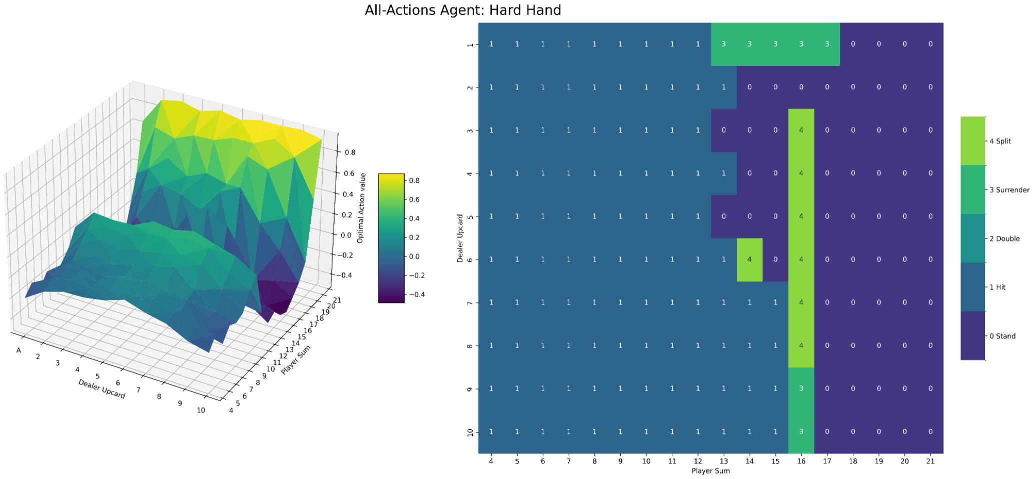
For soft hands (Figure 3), the agent consistently chooses to split a pair of Aces (which appears as a soft hand of 12) unless the dealer also shows an Ace. This makes sense, as splitting increases the chance of drawing strong hands, but doing so against an Ace is riskier. Interestingly, surrender is never chosen with soft hands, as the flexibility of the Ace reduces the downside of continuing to play the hand.

For hard hands (Figure 4), the agent splits a pair of 8s when the dealer's upcard is weak (typically 3 through 8). However, when the dealer shows a strong upcard (9 or higher), the agent may opt to surrender—especially when holding mid-range sums like 15 or 16. In these cases, hitting risks a bust, and standing is unlikely to beat the dealer, making surrender the least costly option.



Dealer Upcard	12	13	14	15	16	17	18	19	20	21
1	1	1	1	1	1	1	1	1	0	0
2	4	1	1	1	1	1	1	0	0	0
3	4	1	1	1	1	1	1	0	0	0
4	4	1	1	1	2	1	1	0	0	0
5	4	1	1	1	1	1	1	0	0	0
6	4	1	1	1	1	1	1	0	0	0
7	4	1	1	1	1	1	0	0	0	0
8	4	1	1	1	1	1	1	0	0	0
9	4	1	1	1	1	1	1	0	0	0
10	4	1	1	1	1	1	1	1	0	0

Figure 3—All-Actions Agent's optimal Q-value (left) and optimal action (right) for soft hands.



Dealer Upcard	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	1	1	0	0	0	4	0	0	0	0	0
4	1	1	1	1	1	1	1	1	1	1	0	0	4	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1	0	0	0	4	0	0	0	0	0
6	1	1	1	1	1	1	1	1	1	1	4	0	4	0	0	0	0	0
7	1	1	1	1	1	1	1	1	1	1	1	1	4	0	0	0	0	0
8	1	1	1	1	1	1	1	1	1	1	1	1	4	0	0	0	0	0
9	1	1	1	1	1	1	1	1	1	1	1	1	3	0	0	0	0	0
10	1	1	1	1	1	1	1	1	1	1	1	1	3	0	0	0	0	0

Figure 4—All-Actions Agent's optimal Q-value (left) and optimal action (right) for hard hands.

3.3 Performance Comparison

Each agent played 500,000 hands under the same game conditions. The reward structure for each outcome is summarized below:

Table 1—Value of outcome.

Action	Outcome	Reward Value
Hit / Stand	Win	+1
	Tie	0
	Loss	-1
Double	Win	+2
	Tie	0
	Loss	-1
Surrender	Always	-0.5
Split	-	Sum of both hand's rewards

Observations:

1. Using **Q-learning**, the agent trained with only hit and stand actions was already able to slightly **outperform the basic strategy**, which allows hit, stand, and double.
2. When the agent has access to **all actions** (including double, surrender, and split), it achieves a higher expected reward without increasing variance—resulting in a **better mean-variance profile**.
3. As expected, agent with all actions has a **slightly lower win rate**. This is because **surrender counts as a loss** in win-rate terms but strategically reduces expected losses when used under the right conditions.

Table 2—Performances comparison across the three agents.

Agent	Hands Played	Expectation of Reward	Std. Dev. of Reward	Win Rate	Tie Rate	Loss Rate
Basic Strategy	500,000	-0.044	1.04	42.8%	8.9%	48.3%
Hit / Stand	500,000	-0.045	0.95	43.0%	9.4%	47.6%
All Actions	500,000	-0.028	0.95	41.2%	9.8%	49.0%

4 CONCLUSION

This study demonstrates that reinforcement learning can effectively train agents to play Blackjack with minimal assumptions. The agent trained with all available actions achieved the highest expected reward and best risk-adjusted performance, even outperforming the classic basic strategy. This shows the power of Q-learning in capturing nuanced decision-making under uncertainty.

5 REFERENCES

1. https://gymnasium.farama.org/tutorials/training_agents/blackjack_tutorial/
2. https://www.winstar.com/blog/how-to-play-blackjack-a-beginners-guide-to-rules-and-strategy/