Achieving Optimal Blackjack Play Through Double Q-Learning

COMP3106 Final Project

Qayam Damji, Shri Vaibhav Mahesh Kumar, Daniel Tam

Carleton University

December 1, 2024

Outline

Background and Motivation

Why Blackjack?

- Perfect blend of skill and probability
- Well-defined rules with complex decision spaces
- Real-world application potential
- Ideal for testing AI adaptation capabilities

Research Goals

- Develop optimal playing strategies using RL
- Test effectiveness of Double Q-Learning
- Compare performance against traditional strategies
- Integrate card counting for enhanced decision-making

Related Prior Work

Q-Learning Evolution

- **Original Q-Learning (Watkins)**
  - Value function approximation
  - State-action pair evaluation
  - Temporal difference learning
- **Double Q-Learning (van Hasselt)**
  - Addresses maximization bias
  - Dual estimator approach
  - Improved stability in stochastic environments

Industry Applications

- FAIR's poker AI breakthrough (2017)
- Professional player defeat milestone
- Reinforcement learning in imperfect information games

Basic Game Rules

Card Values

- 2-10: Face value
- Jack, Queen, King: 10
- Ace: 1 or 11 (flexible)

Objective

- Beat dealer's hand
- Get closest to 21
- Don't exceed 21 (bust)

Player Actions

- Hit: Request another card
- Stand: Keep current hand
- Split: Divide matching pairs
- Double Down: Double bet, one card

Dealer Rules and Game Flow

Dealer Constraints

- Must hit on 16 or below
- Must stand on hard 17 or above
- Some casinos require hit on soft 17
- No splitting or doubling down

Game Resolution

- Player bust: Immediate loss
- Dealer bust: All standing players win
- Higher hand wins (if no busts)
- Equal hands: Push (tie)
- Natural blackjack pays 3:2

Card Counting Fundamentals

Hi-Lo System

- **Low cards (2-6):** $+1$
- **Mid cards (7-9):** 0
- **High cards (10-A):** -1

Running Count vs True Count

- Running Count $=$ Sum of card values seen
- True Count $=$ Running Count $\div$ Decks Remaining
- Positive count: Advantage to player
- Negative count: Advantage to dealer

State Space Design

State Space Components:

$(player\_value, has\_usable\_ace, dealer\_upcard, count\_bucket, is\_pair, pair\_value$

- **player_value** $\in$ [4,21]
- **has_usable_ace** $\in$ 0,1
- **dealer_upcard** $\in$ [1,10]
- **count_bucket** $\in$ -1,0,1
- **is_pair** $\in$ 0,1
- **pair_value** $\in$ [0,10]

Action Space and Constraints

Action Space: $A = \{0 \text{ (Stand)}, 1 \text{ (Hit)}, 2 \text{ (Split)}\}$

Action Constraints

- **Stand (0):**
  - Always available
  - Ends player's turn
- **Hit (1):**
  - Available if not busted
  - Draws one card
- **Split (2):**
  - Requires matching pair
  - Maximum 3 splits
  - Each hand gets new card

Q-Learning Implementation

Core Update Equation

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r(s) + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

Where:

- α: Learning rate
- γ: Discount factor
- $r(s)$: Immediate reward
- s': Next state

Dynamic Learning Rate

$$\alpha(s,a) = \max(\alpha_0 \cdot \delta^{N(s,a)}, \alpha_{\min})$$

- $N(s,a)$: Visit count
- δ: Decay rate

Double Q-Learning Implementation

Dual Q-Tables

- Maintains two Q-value estimators (Q_1, Q_2)
- Reduces overestimation bias
- Randomly updates one table per step

Update Function

$$Q_1(s,a) \leftarrow Q_1(s,a) + \alpha[R + \gamma Q_2(s', \arg\max_{a'} Q_1(s',a')) - Q_1(s,a)]$$

Key Features:

- Action selection from Q_1
- Value estimation from Q_2
- Decorrelated maximum value estimation

Reward Structure

$$R(p, d) = \begin{cases} -1.2b & \text{if } p > 21 \text{ (bust)} \\ 1.1b & \text{if } d > 21 \text{ (dealer bust)} \\ 1.5b & \text{if } p = 21 \text{ (natural)} \\ 1.1b & \text{if } p > d \text{ and } p \geq 20 \\ b & \text{if } p > d \\ -b & \text{if } p < d \\ 0 & \text{if } p = d \end{cases}$$

Where:

- p: Player's hand value
- d: Dealer's hand value
- b: Base reward unit

Epsilon-Greedy Exploration

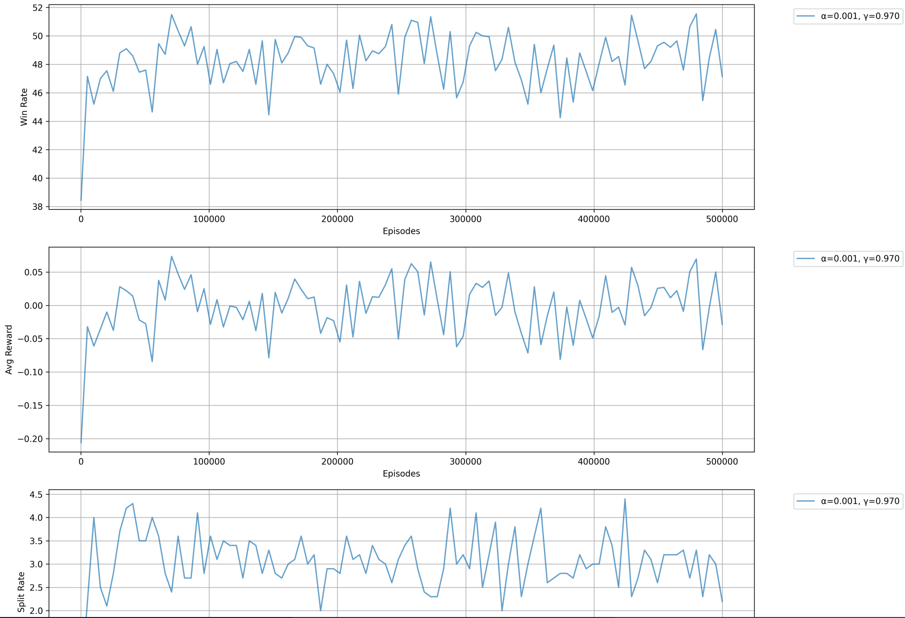
Action Selection Probability

$$P(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A|} & \text{if } a = \arg\max_{a'} Q(s, a') \\ \frac{\epsilon}{|A|} & \text{otherwise} \end{cases}$$

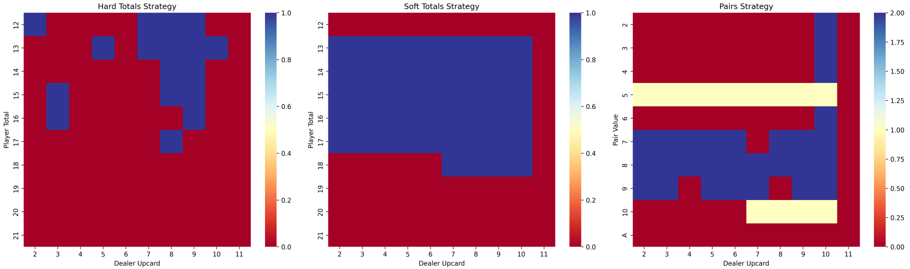
Adaptive Exploration Rate

$$\epsilon = \max(\epsilon_{\text{min}}, \epsilon \cdot \begin{cases} \delta_\epsilon \cdot 1.1 & \text{if improving} \\ \delta_\epsilon & \text{otherwise} \end{cases})$$

Training Results



Strategy Analysis



Hard Totals

- Stand on 17+
- Hit on 16- vs high cards
- Conservative vs dealer 2-6

Soft Totals

- Hit below soft 18
- Stand on soft 19+
- Strategic soft 18 play

Limitations

Performance Ceiling

- Win rate plateau at 48-50%
- Inherent house edge challenge
- Approaches theoretical maximum

Strategic Gaps

- Suboptimal split rate (3.05%)
- Room for reward function refinement
- Casino simulation fidelity limitations

Future Directions

Technical Improvements

- Enhanced split strategy training
- More sophisticated reward shaping
- Deeper card counting integration

Real-World Applications

- Dealer rule variations
- Multi-deck adaptability
- Real-time decision support
- Training tool development