# Blackjack Reinforcement Learning Project Report

**Introduction**

Report on the 21-point Reinforcement Learning project

My talk was about a reinforcement learning (RL) project and training an agent to play the classic card game Blackjack using Q-learning.

The main goal is to develop an agent that can make optimal decisions, maximizing the reward by improving its policy over multiple episodes.

The rules of the game, in playing cards, 2-9 represent the corresponding number size, 10, J, Q, K represent 10, and A represent 1 or 11. Each round is initially dealt 2 cards, if the player can choose to continue drawing more cards or stop depending on the size of the card. After stopping, compare the size with the dealer, 21 points are the largest, if greater than 21 points, it is called burst card, determined to lose.

手机屏幕的截图

AI 生成的内容可能不正确。 图片包含 应用程序

AI 生成的内容可能不正确。

**game interface**

**Identity interpretation**

**Agent：**player

**Environment**：blackjack- v1

**State**：player card sum number，The dealer's card，Usable Ace

**Action**： 0 = Hit ， 1 = Stand

**Reward** : Win = +1 Draw = 0 Lose = -1

**Policy**：num < 17: Hit. Num >= 17: Stand

**Environment Details**

Environment: Blackjack-v1 from the Gymnasium library

State Space: (Player's total sum, Dealer's visible card, Usable Ace flag)

Action Space: Two possible actions: Hit (take another card) or Stand (stop)

**Algorithm: Q-learning setting**

alpha = Learning rate (0.1)

gamma = Discount factor (0.9)

epsilon = greedy strategy (1,0)

r = Reward received after taking action a

s' = Next state

**Training Process**

Total Episodes: **100** episodes

Epsilon-Greedy Strategy: Used to balance exploration (random actions) and exploitation (optimal actions). The epsilon value started at **1.0** and gradually decayed to **0.40** by the 100th episode.

Reward Tracking: Cumulative rewards were tracked across episodes to measure performance improvement.

**Results Analysis**

The following insights were drawn from the output data and training plot:

**Early Stage (Episode 0-30):**

At Episode 0, the average reward was -1.00, indicating **poor initial performance** due to random action selection.

As epsilon decayed to **0.90** (Episode 10), the average reward improved to **-0.20**, suggesting early signs of learning.

**Mid Stage (Episode 30-60):**

By Episode 50, the model achieved a **positive average reward** of **0.10**, indicating improved strategy selection.

The plot shows increased fluctuations, demonstrating exploration during training.

**Late Stage (Episode 60-100):**

By Episode 90, the average reward was **-0.30**, with visible performance drops. This is likely due to remaining exploration and the unpredictable nature of Blackjack.

**Key Observation:** The model showed overall improvement in strategy, with average rewards rising from **-1.00** to **0.10** during peak episodes. However, results fluctuated due to the **randomness** inherent in Blackjack.

文本

AI 生成的内容可能不正确。

**Image analytics**

The plot shows the **Average Reward** trend over 100 episodes.

The fluctuating trend is typical in RL due to the trade-off between exploration and exploitation.

图表, 折线图

AI 生成的内容可能不正确。

**Improvements**

Increasing the number of training episodes to 500 or 1000 would improve convergence.

Fine-tuning the epsilon decay rate for better exploitation in later stages.

Experimenting with different reward structures to accelerate learning.