

440 Project Proposal

Shixi Chen*

Bokyu Seo†

March 2024

1 Introduction

This project will explore the application of reinforcement learning to optimize betting strategies against a mock gambling game supported by a Discord bot named Jeonjeok.

The rules of betting game are as follows:

1. Agent starts with \$100,000
2. Agent can choose to bet some amount of money (minimum \$0.50)
3. Initial winning probability is 43%
4. If the bet is lost, the bet amount is lost, and the probability of winning a bet increases by a random value between 0.35% and 1.6% (uniform random).
5. If the bet is won, there is an 87% chance of winning double the bet amount, a 10% chance of winning triple the bet amount, and a 3% chance of winning five times the bet amount. Additionally, the probability of winning a bet resets to 43%.

RL is beyond the topics we covered in class, but a subject we are very curious to learn about. We thought this problem statement will be a good middle ground between exploration of fundamental RL techniques as well as serve as a starting point to solve more difficult problems in the future including other forms of stochastic betting games and stock trading applications.

2 Project Goal

The primary goal of this project will be to implement a RL model that will be able to learn the optimal betting strategies given the rules above. In other words, the model should be able to determine the optimal amount of money to bet given dynamically changing state information. It is important to note that a closed form "solution" of an optimal betting amount can be calculated as illustrated in this article. We will use this result to gauge the performance of our model in the final analysis. If time allows, we will explore variations of the betting game by adding/changing rules to introduce new challenges.

*Student Number: 65466419

†Student Number: 13749536

3 Project Roadmap

Our Plan consists of 3 phases:

1. Research Phase
2. Implementation Phase
3. Evaluation Phase

Research Phase:

- At a high level, identify and describe the state space, action space, and reward system based on the rules the conditions for terminating an episode
- Research existing reinforcement learning applications in the "stock trading" domain to identify strategies and best practices. One potential paper that could be used for reference *Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy*

Implementation Phase:

- Implement game's states(e.g, current capital, current win rate, bet size, etc.)
- Implement action space(e.g, minimal/maximum amount to bet followed by the game rule)
- Implement reward signal that the agent receives after taking an action
- Define training episodes, the maximum number of steps per episode
- Implement the Q-learning algorithm using the Bellman equation
- Incorporate the **Gymnasium** library (formerly known as **OpenAI Gym**) into the implementation steps above

Evaluation Phase

- Keep track of the total amount of profit the agent accumulates over time(sum the rewards across all episodes), the higher, the better
- Evaluate the percentage of winning bets out of the total number of bets made
- Measure the convergence time of the RL agent. We can track the number of episodes required for the agent's performance metrics to stabilize

4 Contribution

The goal of this project is to create a well-documented, beginner-friendly notebook for newcomers interested in starting their journey in Reinforcement Learning(RL), assuming minimal background knowledge in machine learning, mathematics, and computer science. This project aims to offer step-by-step guidance into a practical application of RL. In the context of stock trading, we will introduce core concepts of RL, including the definition of states, actions, and rewards, and delve into more crucial topics like Markov Decision Process(MDP), Value Function, the Bellman Equation, and the Q-Learning algorithm. Our goal is to make learning RL more accessible and comprehensible.