Project 2 Report

Blackjack Game Eeviroment

This is simple game. There are 52 cards in the deck, 1 to 10 and J, Q, K repeat 4 times and J, Q, K all means 10. The player is playing against the dealer. The player can choose to twist or stick. The player wins if the sum of the cards is 21 or less and the sum of the cards is greater than the sum of the dealer's cards. The player can also choose to stick if the sum of the cards is less than 21. The dealer will continue to draw cards until the sum of the cards is greater than 17. There are Ace cards in the deck. The value of the Ace card is 1 or 11. The value of the Ace card is 11 if the sum of the cards is less than 21. The value of the Ace card is 1 if the sum of the cards is greater than 21.

For this project, I implement a simple blackjack game environment. Each state is represented by a tuple of 3 elements: the player's sum, the dealer's showing card, and whether the player has a usable Ace. The player's sum is the sum of the player's cards. The dealer's showing card is the first card (J, Q, K are 10) of the dealer's cards. *The player has a usable Ace if the player has an Ace card.* This is diffrent with the rule of gymnasium's Blackjack. The player's sum is the sum of the player's cards and the value of the Ace card is 1 if the sum of the cards is greater than 21. For simple, if the sum of player's cards is greater than 21, the number in state is 22.

Methods

To solve the blackjack game, I implement four methods: Monte Carlo, Sarsa, Q-Learning, and Deep Q-Learning. The player will play the game multiple times and update the value function of (state action) pair based on the return of the game. And all of them use epsilon-greedy policy to choose the action.

Monte Carlo

Using first visit Monte Carlo method to solve the blackjack game.

Sarsa

Q-Learning

Deep Q-Learning