Neural Network Theory and Applications
Homework Assignment #4
May, 28, 2020
Due at June, 8

1 Problem Introduction

The goal of this assignment is to apply reinforcement learning to solve a simple game, called *Easy 21*. This exercise is similar to the Blackjack example discussed in the class. However, please note that the rules of the card game are different and non-standard.

2 Environment Introduction

The rule of game of Easy 21 is defined as follows:

- Each draw from the deck results in a value between 1 and 10 (uniformly distributed) with a color of red (probability 1/3) or black (probability 2/3).
- At the start of the game both the player and the dealer draw one black card
- Each turn the player may either stick or hit. If the player hits then he/she draws another card from the deck. If the player sticks he/she receives no further cards
- The values of the player's cards are added (black cards) or subtracted (red cards). If the player's sum exceeds 21, or becomes less than 1, then he/she goes "bust" and loses the game (reward -1).
- If the player sticks then the dealer starts taking turns. The dealer always sticks on any sum of 16 or greater, and hits otherwise. If the dealer goes "bust", then the player wins (reward+1); Otherwise the outcome and reward is computed as follows: the player wins (reward+1) if player's sum is larger than the dealer's sum; the player loses (reward -1) if the player's sum is smaller than the dealer's sum; the reward is 0 if the two sums are the same.
- Assumption: The game is played with an infinite deck of cards (i.e. cards are sampled with replacement)

3 Problems

(1) Implementation of Easy21 environment

You should write an environment that implement the Easy21 game. Specifically, the environment should include a function *step*, which takes a state (dealer's first card, player's current sum) and an action (stick or hit) as inputs, and returns a next state (which may be terminal) and a reward.

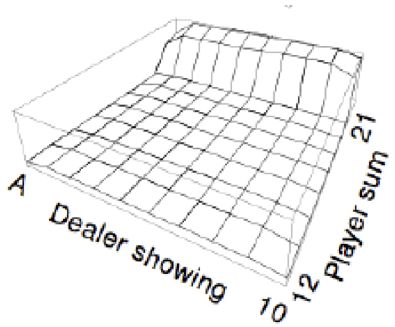
next state, reward = step (state, action)

You should treat the dealer's moves as part of the environment. In other words, i.e. calling step with a stick action will play out the dealer's cards and return the final reward and terminal state. There is no discounting ($\gamma = 1$). We will be using this environment for reinforcement learning.

(2) Q-learning in Easy21

Apply Q-learning to solve Easy21. Try different learning rates α, exploration parameter ε and episode numbers. Plot the learning curve of the return against episode number. Can you find the optimal policy in this game? Plot the optimal the state-value function using similar axes to the following figure taken from Sutton and Barto's Blackjack example.

$$V^*(s) = \max_a Q^*(s,a)$$


A
Dealer showing
Player sum