Optimization 2
Project 3 – Reinforcement Learning

Deliverables

One python (.ipynb or .py) file and one PDF file, submitted to Canvas. Your report should go into some detail about how you solved the problem, include some graphs that explain your results, and include relevant code chunks in the final output. 66% of your grade will be based on whether you create a good/thorough RL or not, the remaining 34% will be based on the quality the presentation of your analysis. We will re-run your code. If we don't score the same as you report or your python file doesn't run, we will go through your code and give partial credit accordingly. The easier it is to read your code the easier it is for us to understand what you're doing, so use a lot of comments in your code!

Problem Overview

You work for a board game company. Your company is considering using RL to find good playing strategies actions when playing the game. Your company is very new to RL, so your boss wants you to explore RL on a simple game first. You will do this for a variation of Connect-4, called PopOut. Rules for the basic version of Connect-4 and PopOut can be found at: https://en.wikipedia.org/wiki/Connect_Four. The basic idea of PopOut is that on your turn you can drop a checker into a column that has open spaces (like in standard connect4) OR you can choose to remove one of your checkers from the bottom row, if one of your checkers is on the bottom row. If you remove a checker from the bottom row then the checkers above the one you remove will then drop down one row. You cannot remove one of your opponent's checkers on the bottom row. You will solve this problem using a self-play reinforcement learning agent.

Specifics

1) Build a reinforcement learning agent that plays PopOut. We have built 2 simple engines in class that did not perform very well on Pong. Your job is to write one that does perform well on PopOut! You may use any of the strategies we talked about in class to improve the performance. You can try actor/critic, you can try a double DQN network, you can use a memory buffer, or you can use a linear annealed strategy. The key difference between the game we played in class and this problem is that the opponent is also controlled by you. You should train one neural network to play the black checkers, and one neural network to play the red checkers. You should simultaneously train both neural networks. You can train these for as long as you like, but each one should be able to easily beat (at least 99 out of 100 games) an opponent that plays completely at random.
2) You will not use gym for this assignment. Code that determines how the board changes upon each play will be provided, but it will be your responsibility to determine if a move is legal or not: you cannot put a checker in a full column and you cannot remove a checker from the bottom row if it isn't your color. Code that checks the board to see if someone has won the game will also be provided.
3) Now comes the fun part. We will have a class tournament to see who trains the best reinforcement learning engine for PopOut. Each member of the winning team will receive 5% extra credit points on the final exam, but not to exceed 100%. The tournament will be structured as follows:
    a. There are 24 groups. You will be split into 8 pools of 3 groups each. In each pool you will play the 2 other teams in your pool. Each time you play an opponent you will play 5 games. Before you play any games, you will randomly decide who goes first on the first game, and then you will take turns going first on each subsequent game. That means every team will play 10 games against opponents. The team with the most wins out of those 10 in your pool will advance to bracket A. Ties will be broken by the number of moves required to win, fewer moves played in those 10 games is better. If you're still tied, flip a coin. Eight teams will advance to bracket A. The remaining teams will move to bracket B.
    b. Once you are in bracket A or B, you will again play 5 games against your opponent, randomly deciding who goes first on the first game and then taking turns going first after that. As in a standard bracket, the winner moves on and the loser is done. Each bracket will have a winner. The winner of bracket A will play the winner of bracket B in class on April 18. The winner of this game will get the 5% extra credit.
    c. This tournament will be played over several days. You can keep training your engine as the tournament progresses if you want to. The schedule for the tournament will be published soon.

4) Pretend you are a developer at the gaming company. Your boss is interested in potentially using RL to create opponents so single players can play your games. Your team has been asked to write a report about the effectiveness of RL. Write this project as if this is what you're going to deliver to your boss. Your boss is pretty technical and understands RL, so don't be afraid to include quantitative material. Your boss is also busy, so be sure to include some visualizations to get the important points across. Would it be beneficial to hire an RL expert to work at your company? Include a summary of your performance in the tournament in your report.