Reinforcement Learning: Chapter 1 Exercises

Name: Eli Andrew

(a) **Exercise 1.1: Self-Play** Suppose, instead of playing against a random opponent, the reinforcement learning algorithm described above played against itself, with both sides learning. What do you think would happen in this case? Would it learn a different policy for selecting moves?

- The random opponent described in the example is assummed to play with a constant policy. So, the first change in this new scenario is that the algorithm is playing against an opponent that changes its policy over time. Now that the opponent will learn from its experience and update its policy, both players will be simulataneously adjusting their policy based on what the other is doing. This will lead to both players learning the optimal policy for an optimal opponent which in tic-tac-toe means a draw. However, since in this scenario the opponent is not just another algorithm but the *same* algorithm that is trying to win for one side, we will probably see the algorithm playing strong moves for its side and weak moves for its opponent in order to make it more likely that it will win.

(b) **Exercise 1.2: Symmetries** Many tic-tac-toe position appear different but are really the same because of symmetries. How might we amend the learning process described above to take advantage of this? In what ways would this change improve the learning process? Now think again. Suppose the opponent did not take advantage of symmetries. In that case, should we? Is it true, then, that symmetrically equivalent positions should necessarily have the same value?

- To amend the learning process to take advantage of symmetries we can modify our state representations to view symmetric states as the same state. This would allow us to have a smaller search space and to therefore learn the optimal policy faster. If we do not take advantage of symmetries then we are allowing our learning process to pick up on alternate outcomes of symmetric states. For example, it could be the case that our opponent plays a different way in symmetrically equivalent states and if we had a reduced state space then we wouldn't pick up on those differences. So, it likely does not make sense to assume that symmetrically equivalent positions should necessarily have the same value, but should be considered on a case by case basis.

(c) **Exercise 1.3: Greedy Play** Suppose the reinforcement learning player was *greedy*, that is, it always played the move that brought it to the position that it rated the best. Might it learn to play better, or worse, than a nongreedy player? What problems might occur?

- A greedy player would always select what it currently "thinks" is best. In many cases this might be the actual best move, however, in certain cases it might miss

potentially useful moves it has yet to try. These would likely be moves that require entering a lower valued state before eventually reaching a higher valued state later on. Therefore, it is likely it would learn to play worse than the nongreedy player who would be able to view the game as a whole rather than as just doing the best thing for the current move.

(d) **Exercise 1.4: Learning from Exploration** Suppose learning updates occurred after *all* moves, including exploratory moves. If the step-size parameter is appropriately reduced over time (but not the tendency to explore), then the state values would converge to a different set of probabilities. What (conceptually) are the two sets of probabilities computed when we do, and when we do not, learn from exploratory moves? Assuming that we do continue to make exploratory moves, which set of probabilities might be better to learn? Which would result in more wins?

- Conceptually, the difference between the probabilities computed when updating for all moves vs. only for non-exploratory moves, is that in the former we are learning probabilities that take into account the price that must be paid for exploration, while in the latter we are not paying any price for exploration. The former strategy would result in probabilities that accurately account for the many possibilities of movements from a given state, while the latter would only be accounting for the current most profitable move from a given state. Therefore, the former strategy would result in more discipled exploration while the latter would be more careless. In the short term I would expect the former strategy to outperform the former strategy, however, in the long term I would expect the opposite, since the latter would have explored more of the state space.

(e) **Exercise 1.5: Other Improvements** Can you think of other ways to improve the reinforcement learning player? Can you think of any better way to solve the tic-tac-toe problem as posed?

- Use initial state values that are more specific than 0.5 for all non-terminal states.
- Use a β update discount for symmetric states.