Solutions for Chapter 1 Exercises

Chen Peng

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?

Answer: Since each agent can detect imperfections in the other's play, both policies converge to the optimal policy at equilibrium. Furthermore, because the opponent always acts optimally, the resulting policies are identical and correspond to the min-max policy.

Exercise 1.2: Symmetries Many tic-tac-toe positions 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?

Answer: To exploit symmetries, we can construct a reduced state space by merging symmetric states. This reduction decreases the overall size of the state space, leading to faster convergence. However, if the opponent does not exploit these symmetries, the agent should refrain from doing so as well. For instance, if the opponent behaves sub-optimally in one state but not in its symmetric counterparts, the agent should respond differently in each case to exploit the imperfection. As a result, symmetrically equivalent positions may carry different values.

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 non-greedy player? What problems might occur?

Answer: The greedy player may play worse than a non-greedy player, due to the lack of exploration. For example, it is less likely for the greedy agent to encounter states where the opponent makes poor moves.

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?

Answer: When learning excludes exploratory moves, the probabilities reflect the optimal policy. When exploratory moves are used for learning, the probabilities correspond to the ϵ-greedy policy. If we do continue to make exploratory moves, learning the probabilities of the ϵ-greedy policy is better, which accounts for the impact of exploratory moves and would result in more wins.

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?

Answer: One way to improve the RL player is to initialize state values using guidance from human experts. Starting with more informed estimates can significantly accelerate the learning process. Another approach is to have a human expert play against the opponent and then train a neural network to imitate the expert's behavior.