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?

Solution

Initially the algorithm will lose and win at random while learning. Eventually the algorithm will always tie after learning has converged. The algorithm will learn a different policy for selecting moves than if it was playing against a fixed opponent.

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?

Solution

We could reduce our state space by collapsing symmetrically equivalent states into just one state. Learning then happens in this reduced state space. Once learning is done, the values learnt for each state can be replicated to its symmetrically equivalent states.

Learning would happen much faster and stand a better chance of avoiding the curse of dimensionality.

We should take advantage of symmetries even if the opponent does not.

Yes - it is true that symmetrically equivalent positions should necessarily have the same value because the probability of winning is the same from each symmetrically equivalent position.

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?

Solution

A greedy player would play worse than a non-greedy player. I would expect the agent will learn to play worse than a non-greedy player. Suppose the agent is always playing greedily, the agent chooses “good” moves according to its own experience. Since its experience might be partial or limited, the greedy choice might exclude some better moves, which could be selected if there are exploratory 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?

Solution

The set of probabilities when we do not learn from exploratory moves is the value we can achieve by performing (our assumed) optimal move. The set of probabilities when we do learn from exploratory moves is the value we can achieve by performing (our assumed) optimal move corrupted by some random moves. I would expect the former is better and I will imagine it results more wins. We are interested to learn a policy which tells us what action to execute if a state is given. The exploratory move, however, is out of our interests: exploratory moves are just some random moves, which are not related to the state that the agent stands in.

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?

Solution

Not at the moment I cannot.