Lab 2 - Natural Computational Methods

Alejandro Villaron, David Håkansson

March 2021

1 Task 1: Q-Learning

Question 1: Because of the number of steps involved on the path used to achieve the trophy. More specifically The more steps required to reach the goal the lower the optimal Q-value(assuming γ is not one, this is explained in question 4 in more detail).

Question 2: When we update the value of a state/action combination we move it towards the r and $\gamma max_{a'}(Q(s', a'))$ terms. But since in the beginning the max term will be zero for all state/action combinations and the reward is only available for one state/action combination in the first episode. This means it is only possible for one state action combination to be non-zero after the first episode. This continues for the later early episodes as well, with in the second episode an arrow will only be updated if it is the state/action pair that leads to the reward or a state/action pair that leads to the state that leads to the reward.

Question 3: Because there is the same distance to the goal from any of those two directions. One example could be state number 3, which has the vale 0.98 for both South and West arrows.

Question 4: Its because of the discount factor. This is because the optimal Q-value that a state action combination can obtain is γ^{n_i}, where n_i is the number of actions the state i is from the state that leads to the goal. This means that longer paths will have a lower maximum Q-value.

Question 5: Because after decreasing the exploration rate, the 4 actions doesn't have the same random probability anymore, and once it picks one successful action, it is reinforced and picked more times, forgetting about the other 3 actions.

Question 6: Yes. For the reason mentioned in question 4, with γ being 1 all the arrows will converge towards 1 rather than decrease depending on the distance to the goal.

Question 7: Every step is rewarded, so whenever a decision is taken it is reinforced and it will be more likely to happen, leading to loops. It also decreases the relative reward obtained when reaching the goal state since the other states are more valuable now. When decreasing the reward to -0.1, it reaches the goal state faster, because the longest paths are increasingly penalized.

Question 8: $Q(s,a) = Q(s,a) + \eta[r + \gamma Q(s', a') - Q(s,a)]$, the η determines how fast we learn, in other words how fast $Q(s,a)$ changes. The r is just the direct reward we gain by doing a certain action in a certain state. The γ is the discount. If we reach the goal we want to reinforce the path used to reach that goal but we also want the path to be short and in order to accomplish that we use the discount which makes the state, action pairs further from the goal, gain less from reaching the goal.

Question 9: $Q(2, down) = r+\gamma \max_{a'} Q(7, a') = 0.95{*}Q(7, down) = 0.95$

2 Task 2: SARSA

Question 10: Because SARSA doesn't assume all future actions are greedy. This means that failed exploratory moves will decrease the relevance of the ideal path, in order to avoid such failed moves again.

Question 11: With SARSA the arrows eventually converge to 0 in the non-adequate states. It could happen in Q-learning too, but it would require a way lower exploration rate, since negative movements are not punished as strongly as with SARSA. The red arrows do not converge and generally have a higher value in Q-Learning. In the ideal scenario the values for largest arrows in the shortest path will be the same for SARSA and Q-learning. This however assumes both have found the same shortest paths and the exploration rate is so small its negligible.

Question 12: The difference is the values of each action are initially very small, and with SARSA such values can decrease. Having such a big exploratory rate means most of the movement will be random, which will lead to a high amount of incorrect paths. Because of this, most of the states will barely increase its value. This combined with the decrease situations will lead to very low values in all the non-optimal paths.

Question 13: We expected the values to be in the range of what we obtained. However, we did not expect the error tiles to actually have a negative weight and decrease in a negative value the q values leading to them.

Question 14: We expected the non-optimal values were smaller than with Q-learning. We kind of expected the path in the lower rows had more priority due to being further of the danger zone, but we did not expect it had such a high priority as it actually has.

Question 15: Because Q-Learning picks the maximum value of the outcome step/action, and considers all future steps greedy. This means that once it founds the right path to the reward it doesn't take in consideration the possible failed routes in such path. However, with SARSA algorithm one failure modifies the previous steps in the path, making the values of the path (even if its the optimal path) worse. This would cause that the non-risky path is preferred over the optimal but risky path.

3 Task 3: Two Rooms

Question 16: Without guidance, the training took an average of 849 steps. However, after the guidance this value was reduced to an average of only 286. This means that having good initial weights significantly improves the performance.

Question 17: This might cause a bias in the training and it would prevent the robot to reach an optimal path that we didn't see during the guidance.

Question 18: Chess, checkers or any kind of game involving action-step leading to a win or lose scenario could be a good example. Since the robot's way of finding the starting point is different than the human approach, we could train it in order to reduce the numbers of steps (for example, not doing a loop of moving the horse back and forth many times in chess), and even if the training is biased, there would be still plenty of possible episodes for the robot to correct such bias and find an optimal path.

4 Feedback

Question 19: Lectures and slides, specially in order to check the equations involving the algorithms.

Question 20: Improve the breakout room selection, since most of the times not everyone can join automatically.