CSC 449 Advanced Topics in Artificial Intelligence

Deep Reinforcement Learning

Exam 2

Fall, 2022

Name: Bradley Dahlke ID#: 7407869 Score: 85

Your solutions to these problems should be uploaded to D2L as a single pdf file by the deadline. You may turn in the solution up to two days late, with a penalty of 10% per day, and you should only upload one version of your solutions.

This exam is individual and open book. You may consult any reference work. If you make specific use of a reference outside those on the course web page in solving a problem, include a citation to that reference.

You may discuss the course material in general with other students, but you must work on the solutions to the problems on your own.

It is difficult to write questions in which every possibility is taken into account. As a result, there may sometimes be "trick" answers that are simple and avoid addressing the intended problem. Such trick answers will not receive credit. As an example, suppose we said, use the chain rule to compute $\frac{\partial z}{\partial x}$ with $z = \frac{7}{y}$ and $y = x^2$. A trick answer would be to say that the partial deriviative is not well defined because y might equal 0. A correct answer might note this, but would then give the correct partial derivative when $y \neq 0$.

Reference Source: Sutton & Barto
Page 130

1. (40 pts) Consider the following pseudo-code for a faulty SARSA algorithm:

procedure SARSA(number of episodes $N \in \mathbb{N}$
discount factor $\lambda \in (0,1]$
learning rate $\alpha_n = \frac{1}{\log(n+1)}$)

Initialize matrices $Q(s,a)$ and $n(s,a)$ to $0, \forall s,a$ $(Q(\text{terminal}, \cdot) = 0$

Loop for Each Episode: **for** episode $k \in 1,2,3,\ldots,n$ **do**

Set our time steps $t \leftarrow 1$

Initialize s_1 ✓

~~Choose a_1 from a uniform distribution over the actions~~

while Episode k is not finished **do**

Take action a_t: observe reward r_t and next state s_{t+1}

Choose $\overset{A'}{a_{t+1}}$ from $\overset{S'}{s_{t+1}}$ using μ_t: an ε-greedy policy with respect to Q ✓

Until S is Terminal **if** The current state is terminal **then** ▷ *Compute target value*

$$y_t = 0$$

Should be r instead of 0

else

$$y_t = r_t + \overset{\gamma}{\max_a} Q(s_{t+1}, a)$$

end if

$n(s_t, a_t) \leftarrow n(s_t, a_t) + 1$

Update Q function:

$$Q(s_{t+1}, a_{t+1}) \leftarrow Q(s_t, a_t) - \alpha_{n(s_t,a_t)} (y_t - Q(s_t, a_t))$$

$t \leftarrow t+1$

end while

end for

end procedure

Find all of the mistakes in the algorithm. Explain why they are mistakes, and correct them.

(note: y is an intermediate calculation value.)

- This should be for the S_t & a_t not the future states when we update our Q.
- This should be having the first action taken based on ε-greedy Q policy.
- Discount factor should be denoted as γ (gamma), not lambda

50points

2. (60 pts) ich is defined through a sequence of policies ging (in the previous algorithm **after correctio** et becomes

$$y_t = r_t + \lambda \sum_a \pi_t(a|s_{t+1}) Q(S_{t+1}, a),$$

Sum of all probabilities actions • Q-val for that next state given action

where $\pi_t(a|s)$ is the probability that a is selected in state s under policy π_t.

a) What sequence of policies (π_t) should you choose so that the corresponding variant of SARSA is on-policy? This variant is called Expected SARSA.

We should be choosing policies such that our returned Q-values are in line with the policy from the get-go. While Expected SARSA doesn't necessarily have to be on-policy, if in this case we stick to the policies that converge, then we are in good shape.

b) Consider an off-policy variant of SARSA corresponding to a stationary policy $\pi = \pi_t \forall t$. Under this algorithm, do the Q values converge? If so, what are the limiting Q values? Justify your answer.

The policy does not change at all for each iteration, so theoretically, it could converge if either we are lucky with stoichastic moves, or have a deterministic policy that does converge, but sometimes this is not the case, as we could have a, for example, stationary policy in the gridworld project that goes up 100 percent of the time, that of which is guaranteed to never converge. So, ultimately, it depends on the policy altogether.