CSE257 Final Sample

Write your answers in the boxes. It is unlikely that you need to write much more beyond what can be fitted in the boxes, but if so you can ask for blank paper to write, and make sure you put name and PID on those when submitting. In fact, we recommend that you put name and PID on all of the following pages as well.

Use name and email that are the same as what you put in Gradescope.

Name:

PID:

Email:

Questions start from the next page. Don't flip to it until we say so.

Question 1. Consider the function $f(x_1, x_2) = -x_1^2 - x_2^2$ over $\mathbb{R}^2$. Answer the following.

1. Choose an arbitrary point x in the input space where the gradient is not zero. Write down an arbitrary descent direction (as a vector) and an arbitrary ascent direction at this x. Explain them using the definitions of descent/ascent directions. Recall that step size is not relevant in this definition.

2. Write down the Hessian matrix of the function f.

3. Show mathematically that the Newton direction at any non-zero point in the input space is an *ascent* direction for this function f.

Question 2. Give an example of a function (write down its analytic form, and it can be a piecewise function) for which it is likely that cross-entropy methods can find the global minimum, while the gradient descent will get stuck at some local minimum if it is not initialized well. Explain your answer, which should involve roughly describing the process of cross-entropy methods. No need for any numerical calculation.

Question 3. We will consider RL on a tiny control problem. Shown in Figure 1 is a simplified inverted pendulum (mass attached to an inelastic weightless rod; imagine balancing a broom upside-down on your hand, or a robot trying to stand up straight, or SpaceX trying to land a rocket vertically). The hope is to balance the pendulum at the upright position by giving a torque input at the end of the rod (the box with $\pm T$ indicating two choices of torque in opposite directions). The angular position is measured by the difference with the upright position, and tilting right is considered the positive direction for θ.

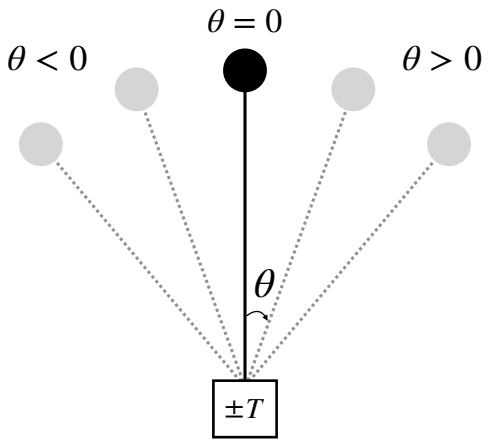

Figure 1: Inverted pendulum for the RL questions.

We use variable θ to represent the pendulum's angular position, and ω to represent its angular speed. Let $s = (\theta, \omega)$ be the current state, and $s' = (\theta', \omega')$ be the next state, after the system runs for a small time

step Δt. The discretized and very aggressively simplified dynamics of the pendulum is as follows:

$$\theta' = \theta + \omega\Delta t \quad (1)$$
$$\omega' = \omega + (k_1\theta + k_2 a)\Delta t \quad (2)$$
$$a \in \{0, T, -T\} \quad (3)$$

where k_1, k_2 are constants dependent on the physical parameters that we do not need to care about, so just assume they are just $k_1 = k_2 = 1$. These equations tell you how the current state (θ, ω) transitions to the next state (θ', ω') under an action a. This control action a is simply either 0 or T or $-T$, and we can also set $T = 1$. The time duration of each step Δt is also constant $\Delta t = 1$ (which is not small enough to physically justify the simplifications we made in the equations, but let's not worry about that). Both equations can involve some stochastic noise, which we choose to not worry about here as well.

The system dynamics is not be known to the learner, so the agent needs to learn the control policy by simulating the system and observing the effects of the actions. Define the reward function for each state $s = (\theta, \omega)$ to be $R(s) = -(\theta^2 + \omega^2)$. The hope is that by maximizing accumulated rewards, the agent can learn to maintain the pendulum at the upright position, in which case the accumulated rewards will be close to zero (and otherwise much more negative). Use the discount factor $\gamma = 0.5$.

1. For the task, is it enough to consider only the angular position θ as the system's state, without the angular velocity ω component? Why? (Recall what "Markovian" means)

2. Let the initial state be $s_0 = (0, 1)$. Take two steps of actions, $a_0 = T$ and $a_1 = -T$, which will unroll the system from s_0 to the next state s_1 and then to another state s_2, according to the equations in (1)-(2). We stop the simulation there and consider s_2 as a terminal state. Write down what s_1 and s_2 are (their numerical values as a vector), and the reward-to-go for each of these three states in this trajectory. Explain the answer according to definition of reward-to-go etc.

3. Consider the state $s = (0.5, 0)$. What is the value of this state under the optimal policy? (Hint: think about what the optimal action is, and even if the other actions have Q-values that are hard to compute, you can probably find out the max Q-value over actions which gives you the optimal state value.)

Question 4. Consider the minimax tree in Figure 2. This is the entire tree of the game, i.e., the leaf nodes are the actual terminal states of the game. The payoff 1 means Max wins, and 0 means Min wins.

Draw the snapshots of the trees that the MCTS algorithm constructs in each of its first three iterations. On each snapshot tree, annotate each non-root node in the tree with the number of visits and average win rate for the player in control of the node. (Wherever there is a random choice to make, the order is from left to right. For instance, when you need to break tie between two children nodes, prioritize the left one.)

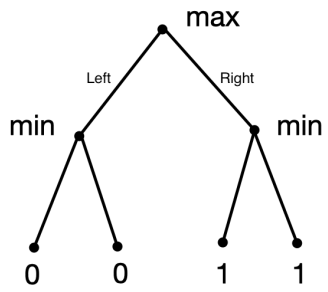

Figure 2: Tree for the MCTS Question