UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

CS440/ECE448 Artificial Intelligence

Exam 3

Spring 2022

May 13, 2022

Your Name: _____

Your NetID: _____

Instructions

- Please write your NetID on the top of every page.
- This is a CLOSED BOOK exam. You will be permitted to bring two 8.5x11 pages of handwritten notes (front & back).
- Calculators are not permitted. You need not simplify explicit numerical expressions.
- There is scratch paper available on the last page of the exam.

Possibly Useful Formulas

Naïve Bayes: $P(X=x,Y=y) \approx P(Y=y)\prod_{i=1}^{n} P(W=w_i|Y=y)$

Laplace Smoothing: $P(w) = \dfrac{\text{Count}(w)+k}{\sum_w \text{Count}(w)+k(1+\sum_w 1)}$

Perceptron: $\vec{w}_y = \vec{w}_y + \eta\vec{x}, \;\; \vec{w}_{f(\vec{x})} = \vec{w}_{f(\vec{x})} - \eta\vec{x}$

Logistic Regression: $\nabla_{\vec{w}_c}\mathscr{L}_i = \nabla_{\vec{w}_c}\left(-\ln\dfrac{e^{\vec{w}_{c_i}^T\vec{x}_i}}{\sum_k e^{\vec{w}_k^T\vec{x}_i}}\right) = \left(\dfrac{e^{\vec{w}_{c_i}^T\vec{x}_i}}{\sum_k e^{\vec{w}_k^T\vec{x}_i}} - y_{i,c}\right)\vec{x}_i$

Neural Net: $\xi_j^{(l)} = b_j^{(l)} + \sum_k w_{j,k}^{(l)} h_k^{(l-1)}, \quad h_j^{(l)} = g^{(l)}\left(\xi_j^{(l)}\right)$

Back-Propagation: $\dfrac{\partial\mathscr{L}}{\partial h_k^{(l-1)}} = \sum_j \dfrac{\partial\mathscr{L}}{\partial h_j^{(l)}}\dfrac{\partial h_j^{(l)}}{\partial h_k^{(l-1)}}$

Admissible Heuristic: $h(n) \le d(n)$

Consistent Heuristic: $h(m)-h(n) \le d(m)-d(n)$ if $d(m)-d(n) \ge 0$

Conditional Probability: $P(A,B,C) = P(A)P(B|A)P(C|A,B)$

Marginal Probability: $P(A,C) = \sum_b P(A,B=b,C)$

Viterbi Algorithm: $v_{j,t} = \max_i v_{i,t-1}a_{j,i}b_{j,k}$

Mixed Nash Equilibrium: $(1-p)w + px = (1-p)y + pz$

Expectiminimax: $U(s) = \max_a \sum_{s'} P(s'|s,a)U(s')$ or $U(s) = \min_a \sum_{s'} P(s'|s,a)U(s')$

Bellman's Equation: $U(s) = R(s) + \gamma\max_a \sum_{s'} P(s'|s,a)U(s')$

Value Iteration: $U_{t+1}(s) = R(s) + \gamma\max_a \sum_{s'} P(s'|s,a)U_t(s')$

Policy Evaluation: $U^{\pi}(s) = R(s) + \gamma\sum_{s'} P(s'|s,a)U^{\pi}(s')$

Policy Improvement: $\pi_{t+1}(s) = \arg\max_a \sum_{s'} P(s'|s,a)U^{\pi_t}(s')$

Q-Learning: $Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha\left(Q_{local}(s_t,a_t) - Q_t(s_t,a_t)\right)$

TD: $Q_{local}(s_t,a_t) = R_t(s_t) + \gamma\max_{a\in\mathscr{A}(s_{t+1})} Q_t(s_{t+1},a')$

SARSA: $Q_{local}(s_t,a_t) = R_t(s_t) + \gamma Q_t(s_{t+1},a_{t+1})$

Deep Q: $\mathscr{L} = \dfrac{1}{2}E\left[Q_t(\vec{s}_t,a_t) - Q_{local}(\vec{s}_t,a_t)\right]$

Imitation Learning: $\mathscr{L} = -\log\pi_{a_t}(\vec{s}_t)$

Actor-Critic: $\mathscr{L} = -\sum_a \pi_a(\vec{s})Q_t(\vec{s},a)$

Question 1 ***(7 points)***

The planet Ilus is home to two species of giant insects, called buzzers and biters. Only biters are dangerous; fortunately, biters are ten times less common than buzzers. A survey team has determined that buzzers are beige-colored with probability p, otherwise they are gray; biters are beige-colored with probability q, otherwise they are gray. The settlers of First Landing have discovered a nest of insects; they have observed ten, of which eight were beige and two were gray. Under what condition should they conclude that the insects are most probably biters? Your answer should be an inequality in terms of p and q.

Question 2 ***(7 points)***

Consider the following neural net:

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \text{ReLU}\left(\begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right)$$

$$f = u_1 h_1 + u_2 h_2$$

$$\mathscr{L} = \frac{1}{2}(f - y)^2$$

Write $\frac{\partial \mathscr{L}}{\partial w_{2,1}}$ in terms of any of the variables listed above. Be sure to consider separately, if necessary, the cases $h_1 = 0$ vs. $h_1 > 0$ and $h_2 = 0$ vs. $h_2 > 0$.

Question 3 ***(7 points)***

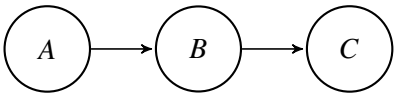
A rogue AI named Richard Feynman is loose on the internet, and is trying to find the safest path from Sandia to NCSA via intervening network routers. The total cost of the path from router i to NCSA is $d_i = -\ln q_i$, where q_i is the probability that Richard can successfully follow the entire path without getting caught at any of the routers between router i and NCSA. The probability of success is upper-bounded by $q_i \leq e^{-m_i}$, where m_i is the physical distance (in miles) between router i and NCSA. Consider an A* search using the heuristic $h(i) = m_i$.

(a) Is this information sufficient to prove that the heuristic is admissible? If so, prove it. If not, give a counter-example.

(b) Is this information sufficient to prove that the heuristic is consistent? If so, prove it. If not, give a counter-example.

Question 4 ***(7 points)***

Consider the Bayesian network shown below, where all three random variables (A, B and C) are binary.



Suppose the model has the following parameters:

$$\begin{aligned} P(A=T) &= p \\ P(B=T|A=F) &= q \\ P(B=T|A=T) &= r \\ P(C=T|B=F) &= s \\ P(C=T|B=T) &= t \end{aligned}$$

where p, q, r, s and t are some constants. Under what conditions is $P(A=T|B=T,C=T) > P(A=T)$? Specify your answer as an inequality in terms of the model parameters p, q, r, s, and t.

$$\frac{P(A=T,B=T|C=T)}{P(B=T|C=T)} = \frac{P(A=T,B=T,C=T)}{P(B=T,C=T)}$$

Question 5 ***(7 points)***

Alice and Bob play the game "Rock, Paper, Scissors" using the standard hand shapes (rock, paper, and scissors), but they use their own special rules to determine the winner of each game. Specifically, they use the following payoff matrix. In this matrix, the number above each diagonal gives the points won by Bob, the number below the diagonal gives the points won by Alice. For example, if Alice chooses "Rock" and Bob chooses "Paper" (denote this outcome as (A=Rock,B=Paper)) then Alice earns -6 points, and Bob earns 1.

Alice's Move	Bob's Move: Rock	Bob's Move: Paper	Bob's Move: Scissors
Rock	-2 / -8	1 / -6	-5 / 4
Paper	7 / 2	-10 / 9	3 / -9
Scissors	10 / 8	-4 / -7	-3 / 3

(a) Are there any fixed-strategy Nash equilibria for this game? If so, what are they?

(b) What is/are the Pareto optimal outcome(s) for this game?

(c) Suppose that Bob chooses "Rock" with probability p, "Paper" with probability q, and "Scissors" with probability $1-p-q$. Write two equations that could be solved to find values of p and q such that it is rational for Alice to choose her move at random. Do not simplify.

Question 6 ***(7 points)***

I have just invented a zero-sum, two-person sequential game called Tic-Tac-Go. The game is played on a three-by-three grid; denote each square as (r,c), where the row is $1 \leq r \leq 3$ and the column is $1 \leq c \leq 3$. In this game, Max places an X on one of the squares $(1,1)$, $(1,2)$, or $(2,2)$. Min then places an O on one of the squares next to the X (right, left, up, or down). Max then earns one point for every empty square that is closer to the X than to the O (lower Manhattan distance). For example, the three tables below show cases in which Max wins 2, 5, and 5 points, respectively:

X	O	

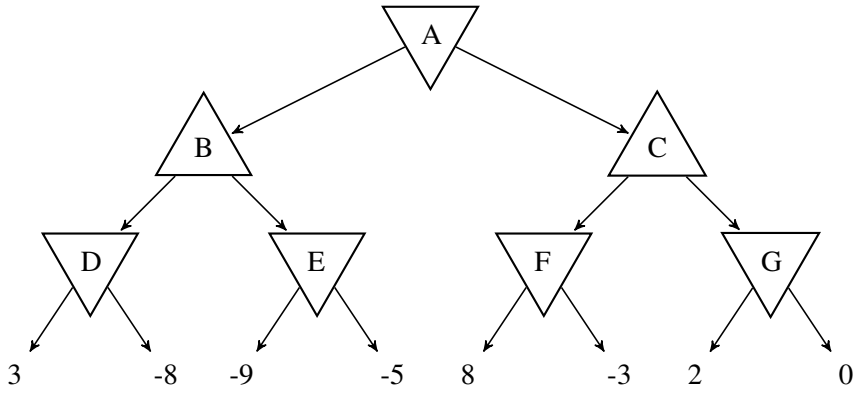
O	X	

	X	O

Draw a complete minimax tree for this game. Use upward-pointing triangles for Max nodes, and downward-pointing triangles for Min nodes. Write the numerical value, for Max, of each terminal node.

Question 7 ***(7 points)***

Consider using the alpha-beta algorithm to search the tree below. When the alpha-beta algorithm starts, the A node has $\alpha = -\infty$, $\beta = \infty$. When any node is expanded, it begins by inheriting α and β from its parent, and may then update α and β after evaluating its children.



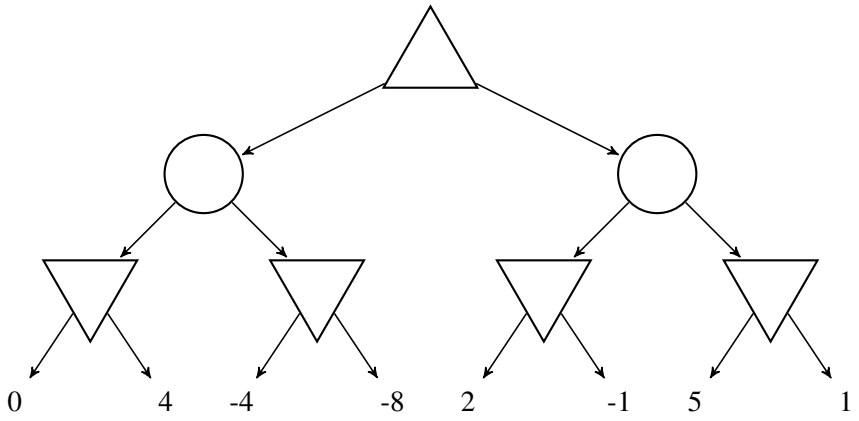
(a) When the E node is first expanded, what value of α does it inherit?

(b) When the C node is first expanded, what value of β does it inherit?

(c) In the tree above, draw an X through any arrow leading to a node that will not be expanded during the alpha-beta search.

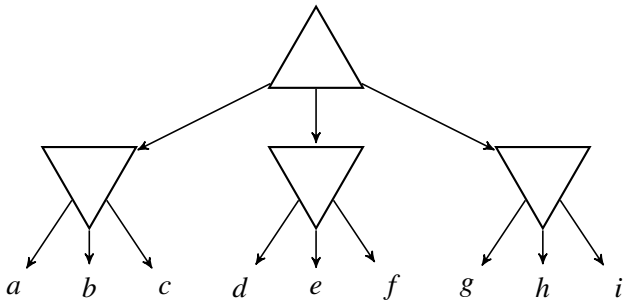
~~Question 8~~ ***(7 points)***

In the tree shown below, circles are chance nodes in which every child is equally likely. Upward-pointing triangles are max nodes, downward-pointing triangles are min nodes. Enter the numerical value of each node inside the corresponding node shape.



Question 9 ***(7 points)***

Stochastic search (a.k.a. Monte Carlo tree search) is a useful heuristic, but sometimes gives a value that is quite different from the real value of a node. Consider the tree below, where a, b, c, d, e, f, g, h, and i are some real-valued constants.



Consider two different methods for estimating the value of the top node:

1. Minimax search
2. Monte Carlo tree search: choose one of the three Max moves uniformly at random, then, from the resulting child, choose a Min move uniformly at random

The value returned by Monte Carlo tree search is random, but there is one rather unique arrangement of the variables a, b, c, d, e, f, g, h, and i that will cause Monte Carlo tree search to *always* return a value that is greater than or equal to the value returned by a Minimax search. What is that arrangement? Your answer should be an equality or inequality, or a set of equalities or inequalities, specifying a relationship among the variables a, b, c, d, e, f, g, h, and i that will cause Monte Carlo tree search to *always* return a value greater than or equal to the value returned by Minimax.

Question 10 ***(7 points)***

Once every ten years, Briar Rose awakens, and checks the value of her stock portfolio. With probability 60%, she finds that the Dow Jones Industrial Average (DJIA) has doubled while she slept; with probability 40%, she finds that it has halved. She then sells whatever assets she had during the past ten years, and uses all of the money for one of two things: either she purchases stocks (a portfolio whose value equals the DJIA), or she purchases gold (suppose that, in her world, the value of gold never changes). After taking one of these two actions, she goes back to sleep for ten more years.

Define the state of her finances in decade d to be a tuple specifying the value of her wealth ten years ago, w_{d-1}, and the value of her wealth now, w_d, thus $s_d = (w_{d-1}, w_d)$. Because the stock market either doubles or halves, while gold never changes, this tuple is always $s \in \{(x, 0.5x), (x,x), (x, 2x)\}$ for some real number x. Define the reward to be the amount she has earned this decade: $R(s_d) = w_d - w_{d-1}$. Consider using value iteration to find $U_t(s)$, the t-step approximation of the utility of state s. Starting with $U_0(s) = 0$ for all states, find $U_1(s)$ and $U_2(s)$ for each of the states $s \in \{(x, 0.5x), (x,x), (x, 2x)\}$. Your answer should be six equations, showing $U_1((x, 0.5x)), \ldots, U_2((x, 2x))$, each as a function of x and γ, where γ is the discount factor.

Question 11 ***(7 points)***

Consider using model-based reinforcement learning to learn the maze in an online game. Suppose that this maze is constructed so that every room in the maze has three exits: left (L), right (R), and forward (F). The maze is magic: the ending point of a tunnel sometimes changes at random, so, for example, taking tunnel L from room $s = 431$ will not always lead to the same place. Fortunately, each room in the maze has a sign specifying its room number, so that, even though you can't be sure where you're going, you can always be sure where you are. Every time you play the game, the software always starts you in room 431.

(a) Define $P(s'|s = 431, a = L)$ to be the probability of arriving in room s' if you start in room 431, and take tunnel L. Suppose you want to estimate $P(s'|s = 431, a = L)$ with a precision of roughly $\varepsilon = 0.1$. How many times do you need to perform action $a = L$ in room $s = 431$ in order to estimate $P(s'|s = 431, a = L)$ sith a precision of $\varepsilon = 0.1$?

(b) Consider the set of all rooms that can be reached by starting in room 431, and taking t tunnels in sequence (i.e., choose a tunnel, take it to a new room, then repeat this $t - 1$ more times). Call the set of rooms that can be reached in this way $\mathbb{S}_t$. As a function of t and ε, how many times must you play the game in order to estimate $P(s'|s, a)$ for every room $s \in \mathbb{S}_t$, and for every action $a \in \{L, R, F\}$, with a precision of ε?

Question 12 ***(7 points)***

One reason to use a replay buffer, in Q-learning, is that the Q-learning algorithm does not immediately learn about the long-term consequences of an action. For example, suppose that your replay buffer contains the following $(s_t, a_t, R(s_t), s_{t+1}, a_{t+1})$ tuples:

s_t	a_t	$R(s_t)$	s_{t+1}	a_{t+1}
431	L	-0.04	1024	R
1024	R	0.5	516	R

Consider using SARSA Q-learning to estimate $Q_t(s,a)$, starting with $Q_0(s,a) = 0$ for all s and a. When you estimate $Q_t(s,a)$, use every row of the replay buffer, not just the t^{th} row. Let α be the learning rate, and let γ be the discount factor; as a function of α and γ, find $Q_1(431,L)$, $Q_1(1024,R)$, and $Q_2(431,L)$.

Question 13 ***(7 points)***

Consider a deep Q-learning algorithm, in which $Q(\vec{s},a)$ is approximated, after t iterations of training, by

$$Q_t(\vec{s},a) = \vec{w}_t^T \vec{h}_t(\vec{s},a),$$

where $\vec{w}_t^T = [w_{t,1},\ldots,w_{t,5}]$ is a 5-dimensional weight vector, and $\vec{h}_t^T(\vec{s},a) = [h_{t,1}(\vec{s},a),\ldots,h_{t,5}(\vec{s},a)]$ is a 5-dimensional hidden node activation vector. Consider two particular state vectors, $\vec{s} = [0.3,0.6]^T$ and $\vec{s} = [-0.2,0.5]^T$, and two particular actions, $a \in \{L,R\}$. Suppose that, in response to these state vectors and these actions, the network currently computes the following hidden node activation vectors:

$\vec{s}^T$	a	$\vec{h}_t^T(\vec{s},a)$
$[0.3,0.6]$	L	$[a,b,c,d,e]$
$[0.3,0.6]$	R	$[f,g,h,i,j]$
$[-0.2,0.5]$	L	$[k,l,m,n,o]$
$[-0.2,0.5]$	R	$[p,q,r,s,t]$

Suppose that, in the t^{th} iteration of deep Q-learning, using only the t^{th} entry from the replay buffer, the output weight vector is $\vec{w}_t = [1,0,0,0,0]^T$, the input state is $\vec{s}_t = [0.3,0.6]^T$, the action is $a_t = R$, the reward is $R(\vec{s}_t) = -0.04$, the resulting state vector is $\vec{s}_{t+1} = [-0.2,0.5]^T$, and the loss function is

$$\mathscr{L} = \frac{1}{2}\left(Q_t(\vec{s}_t,a_t) - Q_{local}(\vec{s}_t,a_t)\right)^2$$

Assume a discount factor of γ, and assume TD-learning, not SARSA. In terms of any or all of the variables $a,b,c,\ldots,r,s,t$, and/or γ, what is the value of $\nabla_{\vec{w}_t}\mathscr{L}$?

Question 14 ***(7 points)***

A robot vacuum cleaner is vacuuming a square room that is D meters on a side. The robot is a disc of radius R meters. The room contains just one obstacle: a rectangular box, of length L and width W, located more than $2R$ meters away from any wall, with its sides parallel to the walls of the room. Because of the shape of the robot, it can never be closer than R to any wall, or to any edge of the box. Define the configuration space of the robot to be (x, y), the two-dimensional position of the center of the robot. In order to plan its trajectory, the robot discretizes the room into a set of $R \times R$ squares. How many such squares are reachable in the robot's configuration space?

SCRATCH PAPER

SCRATCH PAPER