UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

CS440/ECE448 Artificial Intelligence

Exam 3

Spring 2023

May 9, 2023

Your Name: _____

Your NetID: _____

Instructions

- Please write your name on the top of every page.
- Have your ID ready; you will need to show it when you turn in your exam.
- This will be a CLOSED BOOK, CLOSED NOTES exam. You are permitted to bring and use only one 8.5x11 page of notes, front and back, handwritten or typed in a font size comparable to handwriting.
- No electronic devices (phones, tablets, calculators, computers etc.) are allowed.
- SHOW YOUR WORK. Correct answers derivation may not receive full credit if you don't show your work.
- Make sure that your answer includes only the variables that it should include, but DO NOT simplify explicit numerical expressions. For example, the answer $x = \frac{1}{1+\exp(-0.1)}$ is MUCH preferred (much easier for us to grade) than the answer $x = 0.524979$.

Possibly Useful Formulas

$$P(X=x|Y=y)P(Y=y) = P(Y=y|X=x)P(X=x)$$
$$P(X=x) = \sum_y P(X=x,Y=y)$$
$$E\left[f(X,Y)\right] = \sum_{x,y} f(x,y)P(X=x,Y=y)$$

$$\textbf{Precision,Recall} = \frac{TP}{TP+FP}, \frac{TP}{TP+FN}$$
$$\textbf{MPE=MAP:}\quad f(x) = \arg\max\left(\log P(Y=y) + \log P(X=x|Y=y)\right)$$
$$\textbf{Naive Bayes:}\quad P(X=x|Y=y) \approx \prod_{i=1}^{n} P(W=w_i|Y=y)$$

$$\textbf{Laplace Smoothing:}\quad P(X=x|Y=y) = \frac{\text{Count}(X=x,Y=y)+k}{\text{Count}(Y=y)+k|X|},\ \ |X| = \#\text{ possible distinct values of } X$$
$$\textbf{Fairness:}\quad P(Y|A) = \frac{P(Y|\hat{Y},A)P(\hat{Y}|A)}{P(\hat{Y}|Y,A)}$$

$$\textbf{Linear Regression:}\quad \varepsilon_i = f(x_i) - y_i = b + w@x_i - y_i$$
$$\textbf{Mean Squared Error:}\quad \text{MSE} = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i^2$$
$$\textbf{Linear Classifier:}\quad f(x) = \arg\max_k w_k@x + b$$

$$\textbf{Cross-Entropy:}\quad \mathscr{L} = -\frac{1}{n}\sum_{i=1}^{n} \log f_{y_i}(x_i)$$
$$\textbf{Softmax:}\quad \operatorname*{soft\,max}_c(w@x+b) = \frac{\exp(w_c@x+b_c)}{\sum_{k=0}^{V-1}\exp(w_k@x+b_k)}$$
$$\textbf{Softmax Error:}\quad \varepsilon_{i,c} = \begin{cases} f_c(x_i)-1 & c=y_i \\ f_c(x_i)-0 & \text{otherwise} \end{cases}$$
$$\textbf{Gradient Descent:}\quad w \leftarrow w - \eta\nabla_w\mathscr{L}$$

$$\textbf{Neural Net:}\quad h = \text{ReLU}\left(b_0 + w_0@x\right),\quad f = \operatorname*{soft\,max}\left(b_1 + w_1@h\right)$$
$$\textbf{Back-Propagation:}\quad \frac{\partial\mathscr{L}}{\partial h_j} = \sum_k \frac{\partial\mathscr{L}}{\partial f_k}\times\frac{\partial f_k}{\partial h_j},\quad \frac{\partial\mathscr{L}}{\partial w_{0,k,j}} = \frac{\partial\mathscr{L}}{\partial h_k}\times\frac{\partial h_k}{\partial w_{0,k,j}}$$

$$\textbf{Consistent Heuristic:}\quad h(p) \le d(p,r) + h(r)$$
$$\textbf{Alpha-Beta Max Node:}\quad v = \max(v,\text{child});\quad \alpha = \max(\alpha,\text{child})$$
$$\textbf{Alpha-Beta Min Node:}\quad v = \min(v,\text{child});\quad \beta = \min(\beta,\text{child})$$
$$\textbf{Variance Network:}\quad \mathscr{L} = \frac{1}{n-1}\sum_{i=1}^{n}\left(f_2(x_i) - \left(f_1(x_i) - x_i\right)^2\right)^2$$

$$\textbf{Unification:} \quad U = S(P) = S(Q);\ U \Rightarrow \exists x : Q;\ U \Rightarrow \exists x : P$$

$$\textbf{Bayes Rule:} \quad P(Y=y|X=x) = \frac{P(X=x|Y=y)P(Y=y)}{\sum_{y'} P(X=x|Y=y')P(Y=y')}$$

$$\textbf{Unnormalized Relevance:} \quad \tilde{R}(f_c, x_d) = \frac{\partial f_c}{\partial x_d} x_d f_c$$

$$\textbf{Normalized Relevance:} \quad R(f_c, x_d) = \frac{\frac{\partial f_c}{\partial x_d} x_d}{\sum_{d'} \frac{\partial f_c}{\partial x_{d'}} x_{d'}} f_c$$

$$\textbf{Softmax:} \quad \operatorname*{softmax}_j(e) = \frac{\exp(e_j)}{\sum_k \exp(e_k)}$$

$$\textbf{Softmax Deriv:} \quad \frac{\partial \operatorname{softmax}_m(e)}{\partial e_n} = \operatorname*{softmax}_m(e)\delta[m-n] - \operatorname*{softmax}_m(e)\operatorname*{softmax}_n(e), \quad \delta[m-n] = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases}$$

$$\textbf{Viterbi:} \quad v_t(j) = \max_i v_{t-1}(i) a_{i,j} b_j(x_t)$$

$$\textbf{Transformer:} \quad c_i = \operatorname{softmax}(q_i @ k^T) @ v$$

$$\textbf{Pinhole Camera:} \quad \frac{x'}{f} = -\frac{x}{z}, \quad \frac{y'}{f} = -\frac{y}{z}$$

$$\textbf{Convolution:} \quad w_{k,l} * x_{k,l} = \sum_i \sum_j w_{k-i,l-j} x_{i,j}$$

$$\textbf{Kalman Prediction :} \quad \mu_{t|t-1} = \mu_{t-1|t-1} + \mu_\Delta, \quad \sigma^2_{t|t-1} = \sigma^2_{t-1|t-1} + \sigma^2_\Delta$$

$$\textbf{Kalman Gain :} \quad k_t = \frac{\sigma^2_{t|t-1}}{\sigma^2_{t|t-1} + \sigma^2_\varepsilon}, \quad \sigma^2_{t|t} = \sigma^2_{t|t-1}(1-k_t)$$

$$\textbf{Kalman Update:} \quad \mu_{t|t} = \mu_{t|t-1} + k_t\left(x_t - \left(\mu_{t|t-1} + \mu_\varepsilon\right)\right)$$

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

$$\textbf{Value Iteration:} \quad U_i(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) U_{i-1}(s')$$

$$\textbf{Policy Evaluation:} \quad U_i(s) = R(s) + \gamma \sum_{s'} P(s'|s,\pi_i(s)) U_i(s')$$

$$\textbf{Policy Improvement:} \quad \pi_{i+1}(s) = \arg\max_a R(s) + \gamma \sum_{s'} P(s'|s,a) U_i(s')$$

$$\textbf{Q-Learning:} \quad Q_{t+1}(s,a) = Q_t(s,a) + \alpha\left(Q_{\text{local}}(s,a) - Q_t(s,a)\right)$$

$$\textbf{TD Learning:} \quad Q_{\text{local}}(s_t, a_t) = R(s_t) + \gamma \max_{a'} Q_t(s_{t+1}, a')$$

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

$$\textbf{Imitation Learning:} \quad \mathscr{L} = -\log \pi_a(s)$$

$$\textbf{Deep Q Learning:} \quad \mathscr{L} = \frac{1}{2}\left(Q_t(s,a) - Q_{\text{local}}(s,a)\right)^2$$

$$\textbf{Actor-Critic:} \quad \mathscr{L} = -\sum_a \pi_a(s) Q(s,a)$$

$$\textbf{Inverse Kinematics:} \quad \mathscr{C}_{\text{obs}} = \{q : \exists b : \phi_b(q) \in \mathscr{W}_{\text{obs}}\}$$

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

You have been assigned to create an AIAA — an artificially intelligent archeological assistant. Your AIAA will automatically classify necklaces as being of either Atlantean or Numenorean manufacture ($Y \in \{\text{atlantis}, \text{numenor}\}$). Each necklace is characterized by three observable features: the type of metal ($M \in \{\text{gold}, \text{silver}, \text{platinum}\}$), the number of gemstones ($0 \leq N \leq 49$), and the weight of the necklace, in grams, which is an integer in the range $1 \leq W \leq 10,000$.

(a) How many real numbers are required to specify the joint probability distribution $P(M,N,W,Y)$? Why?

Solution: $2 \times 3 \times 50 \times 10,000$ parameters: one for each unique combination of the variables (M,N,W,Y). The probabilities must add up to one, so $2 \times 3 \times 50 \times 10,000 - 1 = 2,999,999$ is also an acceptable answer.

(b) How many real numbers are required to specify a naive Bayes approximation of $P(M,N,W,Y)$, and what are they?

Solution: The parameters are:

- $P(Y)$ - two
- $P(M|Y)$ - $2 \times 3 = 6$
- $P(N|Y)$ - $2 \times 50 = 100$
- $P(W|Y)$ - 20,000

The total parameter count is 20,108.

Question 2 ***(6 points)***

Imagine a two-layer neural net with input vector x, output vector f, and with the following architecture:

$$
\begin{aligned}
f_k &= \text{soft}\max(p) \\
p_k &= \sum_j w_{2,k,j} h_j \\
h_j &= \text{ReLU}(z_j) \\
z_j &= \sum_i w_{1,j,i} x_i
\end{aligned}
$$

What is $\frac{\partial p_k}{\partial z_j}$? Write your answer in terms of the variables x_m, z_m, h_m, p_m, f_m, $w_{1,m,n}$, and/or $w_{2,m,n}$ for any values of m and n that you find to be convenient. Show the steps in the chain rule that are necessary to derive your answer.

Solution:

$$
\begin{aligned}
\frac{\partial p_k}{\partial z_j} &= \frac{\partial p_k}{\partial h_j} \frac{\partial h_j}{\partial z_j} \\
&= w_{2,k,j} u(z_j)
\end{aligned}
$$

Question 3 ***(6 points)***

Prove that every consistent A* search heuristic is also admissible.

Solution: A consistent heuristic is defined as

$$h(p) \leq d(p,r) + h(r)$$

for any pair of nodes p and r. An admissible heuristic is defined as

$$h(p) \leq d(p, \text{Goal})$$

which is a special case of the definition of consistent heuristic, specifically, for the case when $r = \text{Goal}$, and $h(\text{Goal}) = 0$.

Question 4 ***(6 points)***

Consider the problem of trying to prove that birds aren't real. You have the following goalset:

$$\mathscr{G} = \{\text{NOT-REAL}(\text{birds})\}$$

In the attempt to prove your goalset, you will make use of the following rule database:

$$\mathscr{D} = \left\{ \begin{array}{rl} \text{HAVE}(u, \text{wings}) & \Rightarrow \text{FLY}(u) \\ \text{HAVE}(x, \text{feathers}) & \Rightarrow \text{HAVE}(x, \text{scales}) \\ \text{FLY}(y) \wedge \text{HAVE}(y, \text{scales}) & \Rightarrow \text{NOT-REAL}(y) \end{array} \right\}$$

(a) After one step of backward chaining, what is the goalset?

Solution:

$$\mathscr{G} = \{\text{FLY}(\text{birds}) \wedge \text{HAVE}(\text{birds}, \text{scales})\}$$

(b) There are two different goalsets that might result from the second step of backward chaining. What are they?

Solution:

$$\mathscr{G} = \{\text{HAVE}(\text{birds}, \text{wings}) \wedge \text{HAVE}(\text{birds}, \text{scales})\}$$
$$\mathscr{G} = \{\text{FLY}(\text{birds}) \wedge \text{HAVE}(\text{birds}, \text{feathers})\}$$

Question 5 ***(6 points)***

Mary drinks a lot of coffee. At Espresso Royale (ER), half the time she buys espresso ($b_{0,0} = 0.5$), half the time she buys latte ($b_{0,1} = 0.5$). After spending an hour at ER, she either buys another drink from ER (with probability $a_{0,0} = \frac{1}{3}$), or travels to Cafe Bene (CB) with probability $a_{0,1} = \frac{2}{3}$. At CB she buys latte with probability $b_{1,1} = \frac{3}{4}$, and espresso with probability $b_{1,0} = \frac{1}{4}$. After spending an hour at CB she either buys another drink at CB (with probability $a_{1,1} = \frac{4}{5}$), or travels to ER (with probability $a_{1,0} = \frac{1}{5}$). On Saturday at noon she was seen at ER; at 1:00 she bought a latte. Conditioned on these observations, what is the probability that she was at ER at 1:00? Be sure to express your answer in terms of the variables $a_{m,n}$ and $b_{m,k}$, for appropriate values of m and k, before you substitute in the provided numbers.

Solution:

$$\begin{aligned} P(Y_1 = \text{ER} | Y_0 = \text{ER}, X_1 = \text{latte}) &= \frac{P(Y_1 = \text{ER}, X_1 = \text{latte} | Y_0 = \text{ER})}{P(X_1 = \text{latte} | Y_0 = \text{ER})} \\ &= \frac{a_{0,0} b_{0,1}}{a_{0,1} b_{1,1} + a_{0,0} b_{0,1}} \\ &= \frac{\left(\frac{1}{3}\right)\left(\frac{1}{2}\right)}{\left(\frac{1}{3}\right)\left(\frac{1}{2}\right) + \left(\frac{2}{3}\right)\left(\frac{3}{4}\right)} \end{aligned}$$

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

Let (x, y, z) be the 3-dimensional position of a point in the real world, where x is measured in meters west of your camera, y is measured in meters above your camera, and z is measured in meters north of your camera (your camera is facing north). Your camera has a focal length of f meters; points on the film are specified by the coordinates (x', y'), measured in meters. Suppose that there are two parallel lines, in the real world, whose images, in your camera, converge on the vanishing point $(x', y') = (a, b)$. Specify the location, in the real world, of one of those two lines. Your specification will probably be a pair of linear equations in terms of the variables (x, y, z, a, b, f), and in terms of two additional parameters that are not given in this problem statement. What are the new parameters that you have to invent, and why?

Solution: The pinhole camera equations are

$$\frac{x'}{f} = -\frac{x}{z}, \quad \frac{y'}{f} = -\frac{y}{z}$$

In the limit as $z \to \infty$, we have that

$$\frac{x'}{f} = -\lim_{z\to\infty} \frac{x}{z} = a, \quad \frac{y'}{f} = -\lim_{z\to\infty} \frac{y}{z} = b$$

if and only if

$$x = -\left(\frac{a}{f}\right) z + x_0$$
$$y = -\left(\frac{b}{f}\right) z + y_0$$

where (x_0, y_0) is a constant offset of the line, distinguishing it from all other parallel lines that converge to the point (x', y'). The constant offset must be invented because the information in the problem statement is insufficient to determine it, because $\lim_{z\to\infty} \frac{x_0}{z} = 0$ for any constant x_0.

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

Imagine a one-layer CNN that computes its filter output, $z_{k,l}$, from its input $x_{i,j}$ according to

$$z_{k,l} = \sum_{i,j} w_{k-i,l-j} x_{i,j}$$

The filter output is then passed through ReLU nonlinearities and then averaged to create the neural net output, as

$$h_{k,l} = \text{ReLU}(z_{k,l})$$
$$f = \frac{1}{mn} \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} h_{k,l}$$

The network is trained using using one step of stochastic gradient descent with input image x and output target value y, using the loss function

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

In terms of any desired elements of f, h, z, w, and/or x, what is $\frac{\partial \mathscr{L}}{\partial w_{i,j}}$? Be sure to show the derivation of your answer using the chain rule.

Solution:

$$\frac{\partial \mathscr{L}}{\partial w_{i,j}} = \sum_{k,l} \frac{\partial \mathscr{L}}{\partial f} \frac{\partial f}{\partial h_{k,l}} \frac{\partial h_{k,l}}{\partial z_{k,l}} \frac{\partial z_{k,l}}{\partial w_{i,j}}$$
$$= \frac{1}{mn} \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} (f-y) u(z_{k,l}) x_{k-i,l-j}$$

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

A robot is delivering drinks to people on the quad. The terrain is uncertain; it believes its current position to be $(x_{t|t-1}, y_{t|t-1})$, but it is uncertain; these estimates have variances of $(\sigma^2_{x,t|t-1}, \sigma^2_{y,t|t-1})$. In order to inmprove the accuracy of its estimates, the robot takes a measurement of distances to nearby buildings; the measurement specifies the robot's location to be $(x_{\text{obs}}, y_{\text{obs}})$ with no bias, but with measurement variances of $(\sigma^2_{x,\text{obs}}, \sigma^2_{y,\text{obs}})$. How can it combine its previous belief with these new measurements in order to find an improved estimate of its location, what is the value of the improved estimate, and what is the variance of the improved estimate? Assume that the robot's current estimate is independent of the measurement noise, and that the x and y components of each are independent, and that the measurement noise has zero expected value.

Solution: It can use the Kalman gain:

$$(k_x, k_y) = \left(\frac{\sigma^2_{x,t|t-1}}{\sigma^2_{x,t|t-1} + \sigma^2_{x,\text{obs}}}, \frac{\sigma^2_{y,t|t-1}}{\sigma^2_{y,t|t-1} + \sigma^2_{y,\text{obs}}} \right)$$

The improved estimates are then:

$$(x_{t|t}, y_{t|t}) = \left(k_x x_{t|t-1} + (1-k_x) x_{\text{obs}}, k_y y_{t|t-1} + (1-k_y) y_{\text{obs}} \right),$$

with variances of:

$$(\sigma^2_{x,t|t}, \sigma^2_{y,t|t}) = \left(\sigma^2_{x,t|t-1}(1-k_x), \sigma^2_{y,t|t-1}(1-k_y) \right)$$

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

Consider an MDP with two states ($s \in \{0,1\}$), two actions ($a \in \{0,1\}$), with rewards of $R(0) = -10$ and $R(1) = 10$, and with the following transition probabilities:

s'	$P(s'\|s=0,a=0)$	$P(s'\|s=0,a=1)$	$P(s'\|s=1,a=0)$	$P(s'\|s=1,a=1)$
0	0.4	0.7	0.9	0.2
1	0.6	0.3	0.1	0.8

Consider trying to solve for the utilities of the two states, $U(0)$ and $U(1)$, assuming $\gamma = \frac{1}{2}$, using policy iteration. Consider starting with the policy $\pi_1(0) = 0$, $\pi_1(1) = 1$. Write two equations that could be solved to find the policy-dependent utilities $U_1(0)$ and $U_1(1)$ given this policy. Write your equations in terms of the variables $R(0), R(1), P(0|0,0), P(1|0,0), P(0|0,1), P(1|0,1), P(0|1,0), P(1|1,0), P(0|1,1)$, $P(1|1,1)$, and/or γ first, then substitute in the provided numerical values.

Solution:

$$U_1(0) = R(0) + \gamma\left(\sum_{s'} P(s'|s=0,a=0)U_1(s')\right)$$

$$U_1(1) = R(1) + \gamma\left(\sum_{s'} P(s'|s=1,a=1)U_1(s')\right)$$

With the values filled in, this is:

$$U_1(0) = -10 + \frac{1}{2}\left(0.4U_1(0) + 0.6U_1(1)\right) \qquad U_1(1) = 10 + \frac{1}{2}\left(0.2U_1(0) + 0.8U_1(1)\right)$$

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

Consider two different exploration vs. exploitation tradeoff strategies: epsilon-first, and epsilon-greedy. Now consider a hybrid strategy in which you explore every (state,action) pair $N = \frac{1}{\varepsilon}$ times first, then exploit it $100(1-\varepsilon)\%$ of the trials thereafter.

1. Find an advantage that epsilon-first and the hybrid strategy have, relative to epsilon-greedy.
2. Find an advantage that epsilon-greedy and the hybrid strategy have, relative to epsilon-first.
3. Find an advantage that either epsilon-first or epsilon-greedy have, relative to both of the other two strategies.

Solution:

1. Both epsilon-first and the hybrid strategy learn a lot about the environment relatively quickly.
2. Both epsilon-greedy and the hybrid strategy will eventually have an infinite number of trials for every (state,action) pair, so both strategies will eventually converge to perfect knowledge of the environment.
3. Epsilon-first has the advantage that, after the first $N = 1/\varepsilon$ trials of every (state,action) pair, the rest of the trials are spent exploiting the available knowledge, i.e., maximizing reward. Both epsilon-greedy and the hybrid strategy spend $100\varepsilon\%$ of their trials exploring, during which time they are not seeking to maximize reward.

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

Consider an MDP with two actions ($a \in \{R, L\}$), and three states ($s \in \{0, 1, 2\}$). The agent has been accumulating its experiences in an experience replay buffer. The buffer now contains (state,action,reward,new state) (s, a, r, s') tuples for six randomly selected trials, as shown in this table:

Trial ID	s	a	r	s'
1	1	R	10	2
2	1	R	10	0
3	0	L	20	1
4	0	L	20	1
5	0	L	20	0
6	2	R	5	1

Using Laplace smoothing with a smoothing coefficient of k, what are $P(s'|s=0, a=L)$ for $s' \in \{0, 1, 2\}$? Write down the general formula for Laplace smoothing, then fill in the numerical values for the problem.

Solution: The general formula for Laplace smoothing is

$$P(X = x|Y = y) = \frac{\text{Count}(X = x, Y = y) + k}{\text{Count}(Y = y) + k|X|},$$

where $|Y|$ is the number of logically distinct possible values of X. For this problem, the results are:

$$P(s' = 0|s = 0, a = L) = \frac{1 + k}{3 + 3k}$$
$$P(s' = 1|s = 0, a = L) = \frac{2 + k}{3 + 3k}$$
$$P(s' = 2|s = 0, a = L) = \frac{k}{3 + 3k}$$

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

Gepetto the toymaker is able to make 0, 1, or 2 wooden toys in any given day. The state of his shop is well summarized by s =the number of toys that he has for sale. On May 8, 2023, he has $s_t = 7$ toys for sale, and earns $R(7) = \$50$. He decides to produce $a_t = 2$ new toys. On May 9, he learns that he now has $s_{t+1} = 8$ toys left in the store; on this day, he decides to produce $a_{t+1} = 0$ new toys. Prior to the May 8 action, he estimated that the quality of each (state,action) pair is as given in the following table:

	$Q_t(s,a=0)$	$Q_t(s,a=1)$	$Q_t(s,a=2)$
$s=7$	25	60	90
$s=8$	10	100	30

Find two different ways in which Gepetto might update $Q_t(s,a)$ to compute $Q_{t+1}(s=7,a=2)$. Specifically, find $Q_{t+1}(s=7,a=2)$ using both TD-learning and SARSA. Both updates should be written as functions of the learning rate, α, and the discount factor, γ; there should be no other variables on the right-hand-side of your answer.

Solution: Using TD-learning and SARSA, respectively, Qlocal is:

$$\textbf{TD Learning:} \quad Q_{\text{local}}(s_t,a_t) = R(s_t) + \gamma \max_{a'} Q_t(s_{t+1},a')$$

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

Plugging in the numbers from this problem, we have

$$\textbf{TD Learning:} \quad Q_{\text{local}}(7,2) = 50 + 100\gamma$$

$$\textbf{SARSA:} \quad Q_{\text{local}}(7,2) = 50 + 10\gamma$$

The resulting Q-learning updates are

$$\textbf{TD Learning:} \quad Q_{t+1}(7,2) = 90 + \alpha(50 + 100\gamma - 90)$$

$$\textbf{SARSA:} \quad Q_{\text{local}}(7,2) = 90 + \alpha(50 + 10\gamma - 90)$$

which can also be written as:

$$\textbf{TD Learning:} \quad Q_{t+1}(7,2) = (1-\alpha)90 + \alpha(50 + 100\gamma)$$

$$\textbf{SARSA:} \quad Q_{\text{local}}(7,2) = (1-\alpha)90 + \alpha(50 + 10\gamma)$$

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

Gepetto decides to implement deep-Q learning so that he can take advantage of much more information: his new state variable is a vector of six measurements including the prices of the last three toys sold, two measures of overall economic health, and the number of toys available for sale in his shop. Given this state vector, his neural network computes a vector of hidden nodes $h(s) = [h_0(s), \ldots, h_{n-1}(s)]$, then computes three outputs corresponding to the three possible actions, $a \in \{0,1,2\}$. The three outputs of his neural network are

$$Q_t(s,a) = \sum_{i=0}^{n-1} w_{t,a,i} h_i(s), \quad a \in \{0,1,2\},$$

where $w_{t,a,i}$ are weights that are trained using stochastic gradient descent:

$$w_{t+1,a,i} = w_{t,a,i} - \alpha \frac{\partial \mathscr{L}_t}{\partial w_{t,a,i}} \tag{1}$$

Suppose Gepetto measures the state vector s_t, receives a reward of $r_t = 200$, decides on the action $a_t = 2$, and then measures the resulting state vector s_{t+1} on the following day. On the basis of these measurements, propose a loss function $\mathscr{L}_t$ whose derivative could be used to update the neural network as shown in Eq. 1. Specify exactly the way in which your loss function depends on the discount factor γ and on the neural network outputs that might be computed from input vectors s_t ($Q_t(s_t,0), Q_t(s_t,1)$, and/or $Q_t(s_t,2)$) and/or from input vector s_{t+1} ($Q_t(s_{t+1},0), Q_t(s_{t+1},1)$, and/or $Q_t(s_{t+1},2)$).

Solution: The action a_{t+1} is not specified, so we can't use SARSA, so let's use TD-learning:

$$Q_{\text{local}}(s_t, a_t = 2) = R(s_t) + \gamma \max_{a'} Q_t(s_{t+1}, a') \qquad = 200 + \gamma \max_{a'} Q_t(s_{t+1}, a')$$

A useful loss function might be the squared distance between $Q_t(s_t,2)$ and $Q_{\text{local}}(s_t,2)$, which is

$$\mathscr{L} = \frac{1}{2}\left(Q_t(s_t,2) - 200 - \gamma \max_{a'} Q_t(s_{t+1}, a')\right)^2$$

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

Denote the derivative of the y^{th} output of a softmax w.r.t. the $(i,j)^{\text{th}}$ element of its weight matrix as

$$\frac{\partial \operatorname{softmax}_y(w@x)}{\partial w_{i,j}} = \sigma'_{y,i,j}(w@x)$$

In both imitation learning and actor-critic learning, the agent chooses action a with probability $\pi_a(s) = \operatorname{softmax}_a(w@h(s))$, where s is the state vector, $h(s)$ is a vector of hidden nodes, and w is a weight matrix. In both methods, $w_{i,j}$ is updated using the equation:

$$w_{i,j} \leftarrow w_{i,j} - \alpha \frac{\partial \mathscr{L}}{\partial w_{i,j}},$$

where α is a learning rate, and $\mathscr{L}$ is a loss function. Imitation learning and actor-critic learning differ in the definition of $\mathscr{L}$, the loss function. Suppose that, in iteration t, the state vector is s_t, the neural network generates output vector $\pi(s_t)$, the agent chooses action a_t, a human teacher is observed to perform action a_t^*, and an affiliated Q-learning network produces estimates $Q(s_t,a)$ of the quality of all possible actions.

1. In terms of s_t, $\pi(s_t)$, a_t, a_t^*, and/or $Q(s_t,a)$, what is $\mathscr{L}$ for imitation learning?
2. In terms of s_t, $\pi(s_t)$, a_t, a_t^*, and/or $Q(s_t,a)$, what is $\mathscr{L}$ for actor-critic learning?
3. In terms of s_t, $\pi(s_t)$, a_t, a_t^*, $Q(s_t,a)$, and/or $\sigma'_{y,i,j}(w@h(s))$, what is $\frac{\partial \mathscr{L}}{\partial w_{i,j}}$ for imitation learning?
4. In terms of s_t, $\pi(s_t)$, a_t, a_t^*, $Q(s_t,a)$, and/or $\sigma'_{y,i,j}(w@h(s))$, what is $\frac{\partial \mathscr{L}}{\partial w_{i,j}}$ for actor-critic learning?

Solution:

1. $\mathscr{L} = -\log \pi_{a_t^*}(s_t)$
2. $\mathscr{L} = -\sum_a \pi_a(s_t) Q(s_t,a)$
3. $\frac{\partial \mathscr{L}}{\partial w_{i,j}} = -\frac{1}{\pi_{a_t^*}(s_t)} \sigma'_{a_t^*,i,j}(w@h(s))$
4. $\mathscr{L} = -\sum_a Q(s_t,a) \sigma'_{a,i,j}(w@h(s))$

Question 15 ***(7 points)***

Suppose a robot arm has a shoulder angle of θ radians, an upper arm of length L_1, an elbow angle of ϕ radians, and a lower arm of length L_2. Define a position b on the robot arm to be the point b meters from the shoulder, thus the forward kinematics are given by:

$$\phi_b\left(\begin{bmatrix}\theta\\ \phi\end{bmatrix}\right)=\begin{bmatrix}x\\ y\end{bmatrix}=\begin{cases}\begin{bmatrix}b\cos\theta\\ b\sin\theta\end{bmatrix} & 0\le b\le L_1\\ \begin{bmatrix}L_1\cos\theta+b\cos(\theta+\phi)\\ L_1\sin\theta+b\sin(\theta+\phi)\end{bmatrix} & L_1\le b\le L_1+L_2\end{cases}$$

In the workspace $\mathscr{W}$, there is an obstacle: a wall at $y=c$, which makes it impossible for any part of the robot to exist at any point $y\ge c$, where you may assume that $c>0$. This can be written as:

$$\mathscr{W}_{\text{obs}}=\{(x,y):y\ge c\}$$

Define $\mathscr{C}_{\text{obs}}$ to be the set of robot configurations $[\theta,\phi]$ that place any part of the robot arm within $\mathscr{W}_{\text{obs}}$. This can be written as

$$\mathscr{C}_{\text{obs}}=\{(\theta,\phi):P\},$$

where P is some inequality or disjunction of inequalities in terms of θ and ϕ. Find P.

Solution: There was an error in the problem statement. It should have been

$$\phi_b\left(\begin{bmatrix}\theta\\ \phi\end{bmatrix}\right)=\begin{bmatrix}x\\ y\end{bmatrix}=\begin{cases}\begin{bmatrix}b\cos\theta\\ b\sin\theta\end{bmatrix} & 0\le b\le L_1\\ \begin{bmatrix}L_1\cos\theta+(b-L_1)\cos(\theta+\phi)\\ L_1\sin\theta+(b-L_1)\sin(\theta+\phi)\end{bmatrix} & L_1\le b\le L_1+L_2\end{cases}$$

An answer is considered correct if it uses either the correct forward kinematics, or the forward kinematics specified in the problem statement.

The robot hits the obstacle if any part of its arm is in $\mathscr{W}_{\text{obs}}$. Literally, we can write this as:

$$P=(\exists b\in[0,L_1]:b\sin\theta\ge c)\vee(\exists b\in[L_1,L_1+L_2]:L_1\sin\theta+b\sin(\theta+\phi)\ge c)$$

or

$$P=(\exists b\in[0,L_1]:b\sin\theta\ge c)\vee(\exists b\in[L_1,L_1+L_2]:L_1\sin\theta+(b-L_1)\sin(\theta+\phi)\ge c)$$

It is also acceptable to note that we only hit the obstacle if either the elbow or the wrist are inside the obstacle. These two inequalities are a little bit simpler, they are just:

$$P=(L_1\sin\theta\ge c)\vee(L_1\sin\theta+L_2\sin(\theta+\phi)\ge c)$$

or

$$P=(L_1\sin\theta\ge c)\vee(L_1\sin\theta+(L_2-L_1)\sin(\theta+\phi)\ge c)$$

THIS PAGE IS SCRATCH PAPER