Practice Final Exam

University of Illinois at Urbana-Champaign
CS440/ECE448 Artificial Intelligence

The actual final will be on Compass, Wednesday, May 13, 2020, 7-10pm

Your Name: ______________________________

Your NetID: ______________________________

Instructions

- The final exam will be OPEN BOOK, OPEN NOTES, OPEN INTERNET. You may search for solutions, or run them in python, but you MAY NOT CONSULT with any other human being.
- On the final, there will be about 20 questions. About 8 of those questions will cover material that was on the first two exams; about 12 will cover material from the last part of the course. This practice exam covers only the last part of the course. For practice on the first part of the course, please see posted practice and actual exam solutions from midterm 1 and midterm 2.

Question 1 ***(5 points)***

What is a linear classifier?

Solution: A classifier that computes its decision based on a linear function of the data. If the feature vector is $(x_1, \ldots, x_d)$, then the classification function is given by $\text{sgn}(w_1 x_1 + \ldots + w_d x_d + b)$.

Question 2 ***(5 points)***

The softmax function is defined as

$$a_k = \frac{\exp(z_k)}{\sum_j \exp(z_j)}$$

Find da_5/dz_3 in terms of z_3, z_5, a_3 and/or a_5.

Solution:

$$\frac{da_5}{dz_3} = -\frac{\exp(z_5)}{(\sum_j \exp(z_j))^2}\frac{d\sum_j \exp(z_j)}{dz_3} = -\frac{\exp(z_5)\exp(z_3)}{(\sum_j \exp(z_j))^2} = -a_5 a_3$$

Question 3 ***(5 points)***

An image classification algorithm is being trained using the multiclass perceptron learning rule. There are 10 classes, each parameterized by a weight vector $\vec{w}_k$, for $0 \leq k \leq 9$. During the last round of training, all of the training tokens were correctly classified. Which of the weight vectors were updated, and why?

Solution: None. The perceptron learning rule updates the weight vectors only if the classifier makes a mistake.

Question 4 ***(10 points)***

You are a Hollywood producer. You have a script in your hand and you want to make a movie. Before starting, however, you want to predict if the film you want to make will rake in huge profits, or utterly fail at the box office. You hire two critics A and B to read the script and rate it on a scale of 1 to 5 (assume only integer scores). Each critic reads it independently and announces their verdict. Of course, the critics might be biased and/or not perfect, therefore you may not be able to simply average their scores. Instead, you decide to use a perceptron to classify your data. There are three features: a constant bias, and the two reviewer scores. Thus $f_0 = 1$ (a constant bias), f_1 = score given by reviewer A, and f_2 = score given by reviewer B.

Movie Name	A	B	Profit
Pellet Power	1	1	No
Ghosts!	3	2	Yes
Pac is bac	4	5	No
Not a Pizza	3	4	Yes
Endless Maze	2	3	Yes

(a) (5 points) Train the perceptron to generate $\hat{Y} = 1$ if the movie returns a profit, $\hat{Y} = -1$ otherwise. The initial weights are $w_0 = -1$,$w_1 = 0$,$w_2 = 0$. Present each row of the table as a training token, and update the perceptron weights before moving on to the next row. Use a learning rate of $\alpha = 1$. After each of the training examples has been presented once (one epoch), what are the weights?

Solution: The first row of the table is correctly classified, therefore the weights are not changed. The second row is incorrectly classified, therefore the weights are updated as $W = W + YF = [0, 3, 2]$. Using these weights results in misclassification of the third row, therefore the weights are updated again to $[-1, -1, -3]$. Using these weights results in misclassification of the fourth row, therefore the weights are updated again to $[0, 2, 1]$. These weights correctly classify the fifth row.

(b) (3 points) Suppose that, instead of learning whether or not the movie is profitable, you want to learn a perceptron that will always output $\hat{Y} = +1$ when the total of the two reviewer scores is more than 8, and $\hat{Y} = -1$ otherwise. Is this possible? If so, what are the weights w_0, w_1, and w_2 that will make this possible?

Solution: Yes, a perceptron can learn this function. Any weights such that $w_1 = w_2$ and $w_0 = -8w_1$ are correct; for example, the weights $[-8, 1, 1]$.

(c) (2 points) Instead of either part (a) or part (b), suppose you want to learn a perceptron that will always output $\hat{Y} = +1$ when the two reviewers agree (when their scores are exactly the same), and will output $\hat{Y} = -1$ otherwise. Is this possible? If so, what are the weights w_0, w_1 and w_2 that will make this possible?

Solution: This problem is the arithmetic complement of the XOR problem, therefore it is not linearly separable, and cannot be learned by a perceptron.

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

Gradient descent is guaranteed to find a set of model parameters that has the smallest possible loss function on the training corpus.

○ True

√ **False**

Explain:

Solution: Gradient descent finds a local optimum of the loss function, but it is not guaranteed to find a global optimum.

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

A naive Bayes classifier can be implemented as a linear classifier.

√ **True**

○ False

Explain:

Solution: The log of the naive Bayes probability can be written as

$$\ln P(X=x, F_1=f_1, \ldots) = \beta + \sum_{i=1}^{N} [F_i = f_i] w_i$$

where $\beta = \ln P(X=x)$ is the prior, $w_i = \ln P(F_i = f_i | X = x)$, and [Proposition] is the unit indicator function (equal to one if Proposition is true, equal to zero otherwise).

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

We want to implement a classifier that takes two input values, where each value is either 0, 1 or 2, and outputs a 1 if at least one of the two inputs has value 2; otherwise it outputs a 0. Can this function be learned by a Perceptron? If so, construct a Perceptron that does it; if not, why not.

Solution: In this case the input space of all possible examples with their target outputs is:

	0	1	2
2	1	1	1
1	0	0	1
0	0	0	1

Since there is clearly no line that can separate the two classes, this function is not linearly separable and so it cannot be learned by a Perceptron.

Question 8 ***(5 points)***

After t iterations of the "Value Iteration" algorithm, the estimated utility $U(s)$ is a summation including terms $R(s')$ for the set of states s' that can be reached from state s in at most $t-1$ steps.

√ **True**

○ False

Explain:

Solution: Value iteration starts with $U(s) = 0$. Each iteration updates $U(s)$ by adding $R(s)$, plus the maximum over all actions of the expected utility $U(s')$ of the state s' that can be reached from state s in one step. In t iterations of this algorithm, one accumulates rewards from states that are up to $t-1$ steps away.

Question 9 ***(12 points)***

ATARA is an Automatic Telephone-based Airplane Reservation Agent.

In order to make an airplane reservation, ATARA needs to learn the user's starting city, ending city, and date of travel (she always asks in that order). When she starts each dialog, she knows none of these things.

During each of her dialog turns, ATARA has the option of asking for 1 or 2 pieces of information. Unfortunately, her speech recognizer makes mistakes. If she asks for 1 piece of information, she always gets it. If she asks for 2 pieces of information, then she gets both pieces of information with probability $\left(\frac{1}{2}\right)$, but with probability $\left(\frac{1}{2}\right)$, she gets nothing.

ATARA receives a reward of $R(s) = 10$, and ends the dialog, when she has correctly recognized all 3 pieces of information. Otherwise, she gets a reward of $R(s) = -1$ for each dialog turn during which she has not finished the dialog.

(a) (1 point) What is the set of states for this Markov decision process?

Solution: The states are $s \in \{0, 1, 2, 3\}$, specifying the number of pieces of information ATARA has collected.

(b) (1 point) What is the set of actions?

Solution: The actions are $a \in \{1, 2\}$, representing the number of pieces of information that ATARA requests.

(c) (3 points) Write the transition probability table $P(s'|s, a)$.

Solution:

(s, a)	0	1	2	3
$(0, 1)$	0	1	0	0
$(0, 2)$	1/2	0	1/2	0
$(1, 1)$	0	0	1	0
$(1, 2)$	0	1/2	0	1/2
$(2, 1)$	0	0	0	1
$(2, 2)$	0	0	1/2	1/2

(d) (2 points) Use value iteration to find $U(s)$, the utility of each state, assuming a discount factor of $\gamma = 1$.

Solution: The utility estimate at each iteration is given as follows; after $t = 5$, the utility no longer changes.

t	0	1	2	3
1	0	0	0	0
2	-1	-1	-1	10
3	-2	3.5	9	10
4	2.5	8	9	10
5	7	8	9	10

Question 10 ***(5 points)***

What is the optimal policy defined by the Bellman equation?

Solution:

$$\pi^*(s) = \arg\max_a \sum_{s'} P(s'|s,a)U(s')$$

Question 11 ***(5 points)***

In a Markov Decision Process with finite state and action sets, model-based reinforcement learning needs to learn a larger number of trainable parameters than model-free reinforcement learning.

√ **True**

○ False

Explain:

Solution: Model-based learning needs to learn $P(s'|s,a)$, a set of $N_S^2 N_a$ parameters, where N_s is the number of states, N_a the number of actions. Model-free learning needs to learn $Q(s,a)$, a set of only $N_s N_a$ trainable parameters.

Question 12 ***(5 points)***

When we apply the Q-learning algorithm to learn the state-action value function, one big problem in practice may be that the state space of the problem is continuous and high-dimensional. Discuss at least two possible methods to address this.

Solution:

1. Discretize the state space.
2. Design a lower-dimensional set of discrete features to represent the states.
3. Use a parametric approximator (e.g., a neural network) to estimate the Q function values and learn the parameters instead of directly learning the state-action value functions.

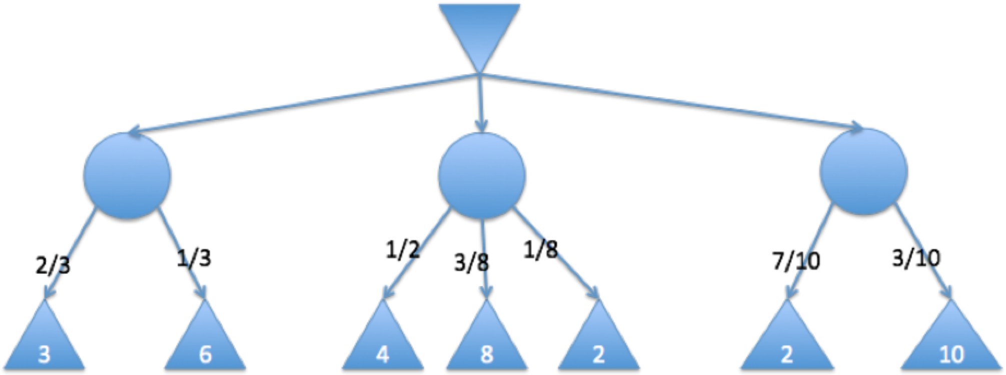
Question 13 *(5 points)*

How can randomness be incorporated into a game tree? How about partial observability (imperfect information)?

Solution: Randomness is incorporated using the expectiminimax algorithm, in which max tries to maximize the expected score, min tries to minimize the expected score. Partial observability is incorporated using a minimax state tree in which neither player knows for sure which state they're in; the max player chooses an action that maximizes the minimum payoff over all of the states he might be in.

Question 14 *(5 points)*

Consider the following expectiminimax tree:



Circle nodes are chance nodes, the top node is a min node, and the bottom nodes are max nodes.

(a) For each circle, calculate the node values, as per expectiminimax definition.

Solution: From left to right: 4, 5.25, 4.4.

(b) Which action should the min player take?

Solution: The first action.

Question 15 ***(5 points)***

What additional difficulties does dice throwing or other sources of uncertainty introduce into a game?

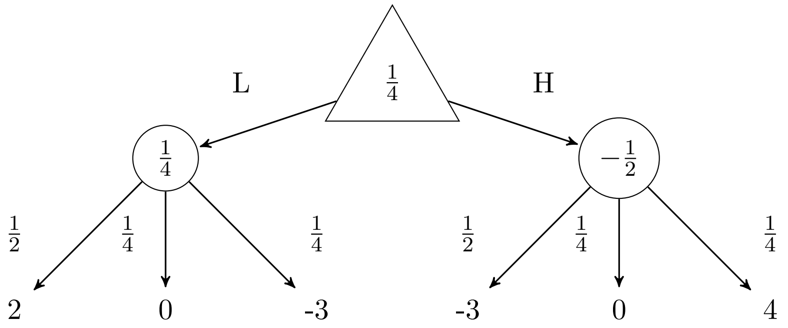
Solution: Uncertainties introduce probabilities into the game. Expectiminimax is used to find solutions for these type of games. Expectiminimax doubles the number of levels as compared to minimax, because there is randomness after every play. Expectiminimax has nasty branching factor and often times defining evaluation functions and pruning algorithms are difficult.

Question 16 ***(5 points)***

Consider the following game, called "High/Low." There is an infinite deck of cards, half of which are 2s, one quarter are 3s, and one quarter are 4s. The game starts with a 3 showing. After each card, you say "High" or "Low," and a new card is flipped. If you are correct (e.g., you say "High" and then the next card is higher than the one showing), you win the points shown on the new card. If there is a tie (the next card equals the one showing), you get zero points. If you are wrong (e.g., you say "High" and then the next card is lower than the one showing), then you lose the amount of the card that was already showing.

Draw the expectimax tree for the first round of this game and write down the expected utility of every node. What is the optimal policy assuming the game only lasts one round?

Solution:



Optimal policy: L

Question 17 ***(5 points)***

Give an example of a coordination game and an anti-coordination game. For each game, write down its payoff matrix, list dominant strategies and pure strategy Nash equilibria (if any).

Solution: The stag hunt is a coordination game. The payoff matrix (player 1's payoff listed first inside each ssquare) is:

	Player 2: Cooperate	Player 2: Defect
Player 1: Cooperate	2, 2	0,1
Player 1: Defect	1,0	1,1

It has no dominant strategy. It has two pure-strategy Nash equilibria: (C,C) and (D,D).

The game of Chicken is an anti-coordination game. The payoff matrix (player 1's payoff listed first inside each ssquare) is:

	Player 2: Chicken	Player 2: Drive
Player 1: Chicken	0, 0	-1,1
Player 1: Drive	1,-1	-10,-10

It has no dominant strategy. It has two pure-strategy Nash equilibria: (C,D) and (D,C).

Question 18 ***(5 points)***

In the lectures, we covered dominant strategies of simultaneous move games. We can also consider minimax strategies for such games, defined in the same way as for multi-player alternating games, except that now, both players make their decision before they have seen what the other player will do. What would be the minimax strategies in the Prisoners Dilemma, Stag Hunt, and Game of Chicken? If both players follow the minimax strategy, does the game outcome differ from the Nash equilibria? When/why would one prefer to choose a minimax strategy rather than a Nash equilibrium?

Solution: Minimax solution maximizes, over all of your possible actions, the minimum, over all of your opponent's possible actions, of your reward.

- Prisoner's Dilemma: Defect. Result is also the Nash equilibrium.
- Stag Hunt: take the Hare. Result is one of the two Nash equilibria.
- Game of Chicken: chicken out. Result is not a Nash equilibrium.

Nash equilibrium is the outcome achieved if each player, knowing the other player's action, has no reason to change their own action: it assumes that you know the other player's action. Minimax makes more sense if you want to limit your losses, and have no way to predict the other player's behavior.

Question 19 ***(5 points)***

Consider the following game:

	Player A: Action 1	Player A: Action 2

Player B:
Action 1
	A=3
	B=2
		A=0
		B=0

Player B:
Action 2
	A=1
	B=1
		A=2
		B=3

(a) Find dominant strategies (if any).

Solution: A dominant strategy is defined as a strategy whose outcome is better for the player regardless of the strategy chosen by the other player. Let's first look for dominant strategies for A: Suppose B chooses Action1. A gets 3 if it chooses Action1 or 0 if it chooses Action2. So it shoould choose Action1. Now suppose B chooses Action2. A gets 1 if it chooses Action1 or 2 if it chooses Action2. So it should choose Action2. Thus there is no dominant strategy for A. Let's look at B: Suppose A chooses Action1. B gets 2 if it chooses Action1 or 1 if it chooses Action2. So it should choose Action1. Now suppose A chooses Action2. B gets 0 if it chooses Action1 or 3 if it chooses Action2. So it should choose Action2. Thus there is also no dominant strategy for B.

(b) Find pure strategy equilibria (if any).

Solution: A Nash Equilibrium is a set of strategies such that no player can get a bigger payoff by switching strageties, provided the other player sticks with the same strategy. There are two: (A: Action1, B: Action1) or (A: Action2, B: Action2).

Question 20 ***(5 points)***

In each square, the first number refers to payoff for the player whose moves are shown on the row-label, the second number refers to payoff for the player shown on the column label.

	A	B	C
A	0, 0	25, 40	5, 10
B	40, 25	0, 0	5, 15
C	10, 5	15, 5	10, 10

(a) Are there any dominant strategies? If so, what are they? If not, why not?

Solution: No. The best move, for each player, depends on what the other player does.

(b) Are there any pure-strategy Nash equilibria? If so, what are they? If not, why not?

Solution: (A,B), (B,A), (C,C)

(c) Are there any Pareto-optimal solutions? If so, what are they? If not, why not?

Solution: (A,B) and (B,A).

Question 21 ***(5 points)***

Suppose that both Alice and Bob want to go from one place to another. There are two routes R1 and R2. The utility of a route is inversely proportional to the number of cars on the road. For instance, if both Alice and Bob choose route R1, the utility of R1 for each of them is 1/2.

(a) Write out the payoff matrix.

Solution:

	Alice R1	Alice R2
Bob R1	A:0.5, B:0.5	A:1, B:1
Bob R2	A:1, B:1	A:0.5, B:0.5

(b) Is this a zero-sum game? Why or why not?

Solution: No. The rewards for Bob and Alice do not sum to zero.

(c) Find dominant strategies, if any. If there are no dominant strategies, explain why not.

Solution: There is no dominant strategy for either player. The best strategy for each player depends on the strategy of the other player.

(d) Find pure strategy equilibria, if any. If there are no pure strategy equilibria, explain why not.

Solution: There are two: (Alice=R1,Bob=R2) and (Alice=R2,Bob=R1).

(e) Find the mixed strategy equilibrium.

Solution: Alice chooses R1 with probability p, and R2 with probability $1-p$. p must be chosen so that Bob's reward is independent of the action he takes.

- Bob's Reward(R1)$= 0.5p + (1-p) = 1 - 0.5p$
- Bob's Reward(R2)$= p + 0.5(1-p) = 0.5 + 0.5p$

Setting the two rewards equal, we find $p = 0.5$.

Question 22 ***(5 points)***

The “Battle of the Species” game is defined as follows. Imagine a cat and a dog have agreed to meet for the evening, but they forgot whether they were going to meet at a frisbee field or an aquarium. The dog prefers the frisbee field and the cat prefers the aquarium. The payoff for each one’s preferred activity is 4 and the payoff for the non-preferred activity is 3 assuming the cat and the dog end up at the same place. If they end up at different places, each gets a 1 if they are at their preferred place, and 0 if they are at their non-preferred place.

(a) Give the normal form (matrix) representation of the game.

Solution:

	Dog: Frisbee	Dog: Aquarium
Cat: Frisbee	C:3,D:4	D:0,C:0
Cat: Aquarium	C:1,D:1	C:4,D:3

(b) Find dominant strategies (if any). Briefly explain your answer.

Solution: None. A dominant strategy is a strategy that maximizes the player’s payoff regardless of what the other player does. In this case, if one player chooses frisbee, the other one should choose frisbee, and if one chooses aquarium, the other one should choose aquarium. Therefore, there is no dominant strategy.

(c) Find pure strategy equilibria (if any). Briefly explain your answer.

Solution: (Dog: frisbee; Cat: frisbee); (Dog: aquarium; Cat: aquarium). From either of these two states, no player can get a bigger payoff from changing actions unilaterally.

Question 23 ***(5 points)***

You have a two-layer neural network trained as an animal classifier. The input feature vector is $\vec{x} = [x_1, x_2, x_3, 1]$, where x_1, x_2, and x_3 are some features, and 1 is multiplied by the bias. There are two hidden nodes, and three output nodes, $\vec{y}^* = [y_1^*, y_2^*, y_3^*]$, corresponding to the three output classes $y_1^* = \Pr(\text{dog}|\vec{x})$, $y_2^* = \Pr(\text{cat}|\vec{x})$, $y_3^* = \Pr(\text{skunk}|\vec{x})$. The hidden layer uses a sigmoid nonlinearity, the output layer uses a softmax.

(a) (2 points) A Maltese puppy has the feature vector $\vec{x} = [2, 20, -1, 1]^T$. Suppose all weights and biases are initialized to zero. What is $\vec{y}^*$?

Solution: If all weights and biases are zero, then the excitation of each hidden node is $0 \times 2 + 0 \times 20 + 0 \times (-1) + 0 \times 1 = 0$. With zero input, the sigmoid $1/(1 + \exp(-f)) = 0.5$, but weights in the last layer are also all zero, so the excitations at the last layer are all zero. With a softmax nonlinearity, every output node is computing $\exp(0)/\sum_{i=1}^{3}\exp(0) = 1/3$. So

$$\vec{y}^* = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$$

(b) (3 points) Let w_{ij} be the weight connecting the i^{th} output node to the j^{th} hidden node. What is dy_2^*/dw_{21}? Write your answer in terms of y_i^*, w_{ij}, and/or the hidden node activations h_j, for any appropriate values of i and/or j.

Solution: Let's use the notation f_i as the excitation of the i^{th} output node. That allows us to write the softmax as:

$$y_2^* = \frac{\exp(f_2)}{\sum_{j=1}^{3}\exp(f_j)}, \quad f_j = \sum_i w_{ji} h_i$$

Then:

$$\begin{aligned}
\frac{dy_2^*}{dw_{21}} &= \frac{1}{\sum_{i=1}^{3}\exp(f_i)} \frac{d\exp(f_2)}{dw_{21}} + \exp(f_2)\frac{d(1/\sum_i \exp(f_i)))}{dw_{21}} \\
&= \frac{1}{\sum_{i=1}^{3}\exp(f_i)} \exp(f_2)\frac{df_2}{dw_{21}} + \exp(f_2)\left(-\frac{1}{(\sum_{i=1}^{3}\exp(f_i))^2}\right)\frac{d(\sum\exp(f_i)))}{dw_{21}} \\
&= \frac{\exp(f_2)}{\sum_{i=1}^{3}\exp(f_i)} h_1 - \frac{\exp(f_2)}{(\sum_{i=1}^{3}\exp(f_i))^2}\frac{d\exp(f_2)}{dw_{21}} \\
&= \frac{\exp(f_2)}{\sum_{i=1}^{3}\exp(f_i)} h_1 - \frac{\exp(f_2)}{(\sum_{i=1}^{3}\exp(f_i))^2}\exp(f_2)\frac{df_2}{dw_{21}} \\
&= \frac{\exp(f_2)}{\sum_{i=1}^{3}\exp(f_i)} h_1 - \frac{\exp(f_2)^2}{(\sum_{i=1}^{3}\exp(f_i))^2} h_1 \\
&= y_2^* h_1 - (y_2^*)^2 h_1
\end{aligned}$$

Question 24 ***(10 points)***

A cat lives in a two-room apartment. It has two possible actions: purr, or walk. It starts in room $s_0 = 1$, where it receives the reward $r_0 = 2$ (petting). It then implements the following sequence of actions: a_0 =walk, a_1 =purr. In response, it observes the following sequence of states and rewards: $s_1 = 2$, $r_1 = 5$ (food), $s_2 = 2$.

(a) (3 points) The cat starts out with a Q-table whose entries are all $Q(s, a) = 0$, then performs one iteration of TD-learning using each of the two SARS sequences described above (one iteration/time step, for two time steps). Because the cat doesn't like to worry about the distant future, it uses a relatively high learning rate ($\alpha = 0.05$) and a relatively low discount factor ($\gamma = \frac{3}{4}$). Which entries in the Q-table have changed, after this learning, and what are their new values?

Solution:

- $t = 0$:

$$\begin{aligned} Q_{local} &= r_0 + \gamma \max_a Q(s_1, a) = 2 + 0 = 2 \\ Q(1, \text{walk}) &\leftarrow Q(1, \text{walk}) + \alpha(Q_{local} - Q(1, \text{walk})) \\ &= 0 + 0.05(2 - 0) = 0.1 \end{aligned}$$

- $t = 1$:

$$\begin{aligned} Q_{local} &= r_1 + \gamma \max_a Q(s_2, a) = 5 + 0 = 5 \\ Q(2, \text{purr}) &\leftarrow Q(2, \text{purr}) + \alpha(Q_{local} - Q(2, \text{purr})) \\ &= 0 + 0.05(5 - 0) = 0.25 \end{aligned}$$

So the changed values are $Q(1, \text{walk}) \leftarrow 0.1$ and $Q(2, \text{purr}) \leftarrow 0.25$.

(b) (2 points) Instead of model-free learning, the cat decides to implement model-based learning. It estimates $P(s'|s, a)$ using Laplace smoothing, with a smoothing parameter of $k = 1$, using the two SARS observations listed at the start of this problem. What are the new values of $P(s'|s = 2, a = \text{purr})$ for $s' \in \{1, 2\}$?

Solution:

$$P(s' = 1|s = 2, a = \text{purr}) = \frac{1 + \text{Count}(s_t = 2, a_t = \text{purr}, s_{t+1} = 1)}{2 + \sum_{s'} \text{Count}(s_t = 2, a_t = \text{purr}, s_{t+1} = s')} = \frac{1}{3}$$

$$P(s' = 2|s = 2, a = \text{purr}) = \frac{1 + \text{Count}(s_t = 2, a_t = \text{purr}, s_{t+1} = 2)}{2 + \sum_{s'} \text{Count}(s_t = 2, a_t = \text{purr}, s_{t+1} = s')} = \frac{2}{3}$$

(c) (3 points) After many rounds of model-based learning, the cat has deduced that $R(1) = 2$, $R(2) = 5$, and $P(s'|s, a)$ has the following table:

a:	purr		walk	
s:	1	2	1	2
$P(s'=1\|s,a)$	2/3	1/3	1/3	2/3
$P(s'=2\|s,a)$	1/3	2/3	2/3	1/3

The cat decides to use policy iteration to find a new optimal policy under this model. It starts with the following policy: $\pi(1)=$ purr, $\pi(2)=$ walk. Now it needs to find the policy-dependent utility, $U^\pi(s)$. Again, because the cat doesn't care about the distant future, it uses a relatively low discount factor ($\gamma=3/4$). Write two linear equations that can be solved to find the two unknowns $U^\pi(1)$ and $U^\pi(2)$; your equations should have no variables in them other than $U^\pi(1)$ and $U^\pi(2)$.

Solution: The two equations are

$$U^\pi(1) = R(1) + \frac{3}{4}\sum_{s'} P(s'|1,\pi(1))U^\pi(s')$$
$$U^\pi(2) = R(2) + \frac{3}{4}\sum_{s'} P(s'|2,\pi(2))U^\pi(s')$$

Plugging in the given values of all variables, we have

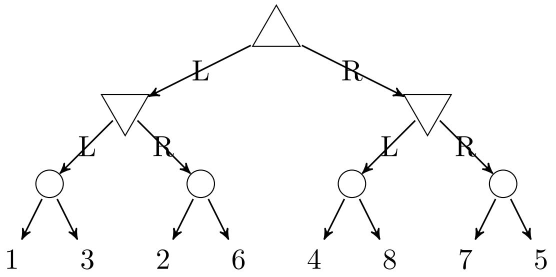
$$U^\pi(1) = 2 + \frac{3}{4}\left(\frac{2}{3}U^\pi(1) + \frac{1}{3}U^\pi(2)\right)$$
$$U^\pi(2) = 5 + \frac{3}{4}\left(\frac{2}{3}U^\pi(1) + \frac{1}{3}U^\pi(2)\right)$$

(d) (2 points) Since it has some extra time, and excellent python programming skills, the cat decides to implement deep reinforcement learning, using an actor-critic algorithm. Inputs are one-hot encodings of state and action. What are the input and output dimensions of the actor network, and of the critic network?

Solution: The actor network takes a state as input, thus its input dimension is 2 (if the input is a one-hot encoding of two states). It computes the probability that any given action is the best action, so its output dimension is 2 (if there are two possible actions). The critic takes, as input, an encoding of the state (two dimensions), and an encoding of the action (two dimensions, if the action is a one-hot encoding of two possible actions), for a total of 4 input dimensions. It computes, as output, a real-valued score $Q(s,a)$, which is a 1-dimensional (scalar) output.

Question 25 ***(5 points)***

Consider a game with eight cards ($c \in \{1, 2, 3, 4, 5, 6, 7, 8\}$), sorted onto the table in four stacks of two cards each. MAX and MIN each know the contents of each stack, but they don't know which card is on top. The game proceeds as follows. First, MAX chooses either the left or the right pair of stacks. Second, MIN chooses either the left or the right stack, within the pair that MAX chose. Finally, the top card is revealed. MAX receives the face value of the card (c), and MIN receives $9 - c$. The resulting expectiminimax tree is as follows:



(a) (2 points) Assume that the two cards in each stack are equally likely. What is the value of the top MAX node?

> **Solution:** Propagating backward using expectiminimax, we find that the value of the top node is 6.

(b) (3 points) Consider the following rule change: after MAX chooses a pair of stacks, he is permitted to look at the top card in any one stack. He must show the card to MIN, then replace it, so that it remains the top card in that stack. Define the belief state, b, to be the set of all possible outcomes of the game, i.e., the starting belief state is the set $b = \{1, 2, 3, 4, 5, 6, 7, 8\}$; the PREDICT operation modifies the belief state based on the action of a player, and the OBSERVE operation modifies the belief state based on MAX's observation. Suppose MAX chooses the action R. He then turns up the top card in the rightmost deck, revealing it to be a 7. What is the resulting belief state?

> **Solution:** After MAX chooses the right set of stacks, the PREDICT update step results in a belief state of $b = \{4, 8, 7, 5\}$. After he looks at the rightmost deck and finds that it contains a 7, the OBSERVE update step restricts the belief state to $b = \{4, 8, 7\}$.

Question 26 ***(5 points)***

In my house, there are M roommates, and N remaining cookies. We must decide how to allocate N cookies among M roommates.

(a) (2 points) First, assume that there are $N = 3$ roommates, and $M = 2$ cookies. We decide to use a VCG mechanism, and to put the proceeds of the auction into a cookie fund that will be used to buy more cookies. The three roommates offer bids of \$5, \$3, and \$6. Using the normal rules of a VCG mechanism, specify how much money goes into the cookie fund, and what is the net value received (value received minus price paid) by each of the three roommates.

> **Solution:**
> In the VCG mechanism, cookies go to the top $N = 2$ bidders, who each pay $b_{N+1} = \$3$, so the cookie fund gets \$6. The lowest bidder gets a net value of \$0, the other two each get a net value equal to $v_i - b_{N+1}$, which is \$(5-3)=\$2 in the case of one roommate, \$(6-3)=\$3 for the other.

(b) (3 points) Now suppose there are $M = 2$ roommates, and $N = 3$ cookies. One of the cookies is deluxe, and costs \$10, the other two cookies are regular, and each cost \$1. If one roommate takes the deluxe cookie and the other takes the regular, then the roommate with the deluxe cookie eats a cookie worth \$10, while the other roommate eats two cookies worth a total of \$2. If both roommates take regular cookies, then they each also eat half of the deluxe cookie, so they each eat 1.5 cookies worth a total of \$6. If both roommates try to take the deluxe cookie, then they start arguing, the dog eats all three cookies while they're busy arguing, and both roommates receive a value of \$0. Find the mixed-strategy Nash equilibrium for this game.

> **Solution:**
> The game is symmetric. Suppose my roommate chooses the deluxe cookie with a probability p, and regular with a probability of $1 - p$. It is rational for me to choose at random only if my expected rewards are equal, regardless of my action. If I choose the deluxe cookie, my expected reward is $0 \times p + 10 \times (1 - p)$. If I choose regular, my expected reward is $2 \times p + 6 \times (1 - p)$. Setting these equal, we learn that my rewards for either action are equal only if my roommate chooses the deluxe cookie with probability $p = 2/3$. Since the situation is symmetric, equilibrium exists only if each roommate chooses the deluxe cookie with a probability of $p = 2/3$.