Final Examination

Term: __Spring__________ **Year:** __2023______

Student name:

______________________ ______________________ ______________________
(First name) (Middle name) (Last name)

Waterloo student identification number: ____

Course abbreviation and number: _____CS486/686_______________________________________

Course title: ______Introduction to Artificial Intelligence____________________________

Section(s): _______001-002-003____

Sections combined course(s): _____001-002-003____

Section numbers of combined course(s): ___001-002-003___________________________________

Name of instructor(s): _____Pascal Poupart and Sriram Ganapathi_________________________

Date of exam: ______August 4, 2023____

Exam period start time: ___16:00_________________ Exam period end time: ____18:30__________________

Duration of exam: ______2 hours 30 minutes____

Number of exam pages: (includes cover page) __34_______________________________________

Exam type: (select one)

[✓ Closed book] Special materials Open book

Materials allowed: (select one)

No additional materials are allowed

[✓ Materials allowed are listed below]

___Non programmable calculator____

Exams are printed double sided on white paper.

[✓ Select this box if second side of paper is to be used for rough work calculations.]

Marking scheme:

Question	Score	Question	Score
1	/16	6	/12

Question	Score	Question	Score
2	/12	7	/12
3	/12	8	/12
4	/12		
5	/12	Total	/100

Question 1 [16 pts] Are the following statements true or false? No justification required.

a) **[2 pts]** Temporal difference methods update the current value estimate at a state using the returns from a full trajectory until the end of the episode from that state.

False (these are monte-carlo methods. TD methods use one step returns to update the current estimates).

b) **[2 pts]** Model-based RL methods are sample inefficient compared to model-free methods.

False (model based methods are more sample efficient as arbitrary experience samples can be obtained from the transition and reward models to update the value estimates. The agent does not need to wait for true experiences from interactions with the environment to update the value estimate for a particular state action pair).

c) **[2 pts]** A cooperative stochastic game is guaranteed to have a unique pure strategy Nash Equilibrium with unique strategies for each player.

False (they can have multiple NE having the same utilities.)

d) **[2 pts]** A dominant strategy equilibrium is guaranteed to be a Nash equilibrium as well.

True

e) **[2 pts]** A Pareto optimal outcome is guaranteed to Pareto dominate every other outcome in a normal form game.

False (the Pareto optimal outcome is not Pareto dominated by any other outcome)

f) **[2 pts]** In Thompson Sampling, sampling a finite number of samples from the posterior distribution enables exploration.

True

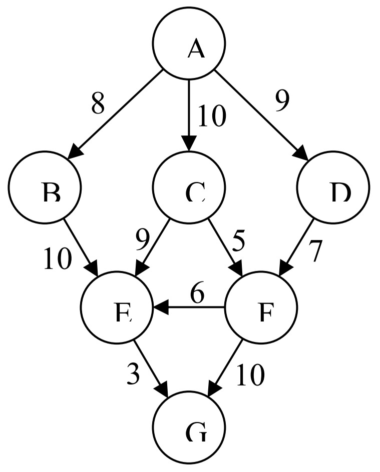
g) **[2 pts]** The backtracking search algorithm for constraint satisfaction problems is guaranteed to find a solution when there exists one.

Accept both True/False since it is true in general, but there are edge cases where it is false (i.e., when the branching factor is infinite or the length of a path is infinite)

h) **[2 pts]** In a decision network, dynamic programming optimizes the decisions in chronological order since later decisions depend on earlier decisions.
False (dynamic programming optimizes decisions in the reverse order)

Question 2 [12 pts] Search techniques

a) **[8 pts]** Consider the following graph and 2 heuristics for estimating the cost of reaching the goal node G from any other node:



Node	**h1**	**h2**
A	6	21
B	14	13
C	5	3
D	16	13
E	4	2
F	9	8
G	0	0

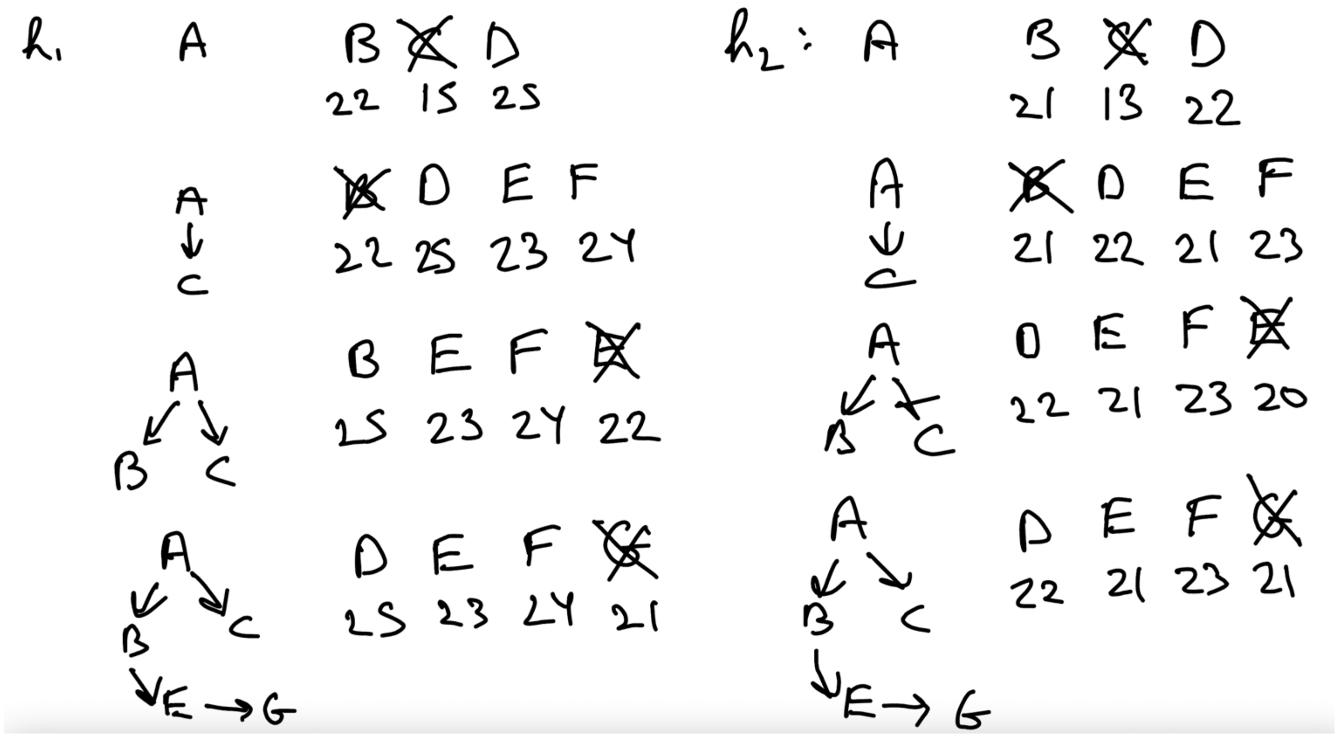
i) **[4 pts]** Indicate for each heuristic, h1 and h2, whether a least cost path is guaranteed to be found when searching for a path from any node to the goal node G using A*? Explain briefly why.

h_1 is not guaranteed to find a least cost path since it is not admissible (h_1(B)=14 which overestimates the lowest cost of 13)

h2 is not guaranteed to find a least cost path since it is not consistent. More precisely, the cost estimate is not increasing monotonically from A to C: $21 = h_2(A) > 10+h_2(C) = 13$

.

ii) [4 pts] Indicate the path found and the nodes expanded to go from A to G in A* search for each heuristic.



h₁ A — B 22, ~~C~~ 15, D 25
A→C — ~~B~~ 22, D 25, E 23, F 24
A→B, A→C — B 25, E 23, F 24, ~~E~~ 22
A→B, A→C, B→E→G — D 25, E 23, F 24, ~~G~~ 21
h₂: A — B 21, ~~C~~ 13, D 22
A→C — ~~B~~ 21, D 22, E 21, F 23
A→B, A→C — D 22, E 21, F 23, ~~E~~ 20
A→B, A→C, B→E→G — D 22, E 21, F 23, ~~G~~ 21

Question 2 (continued)

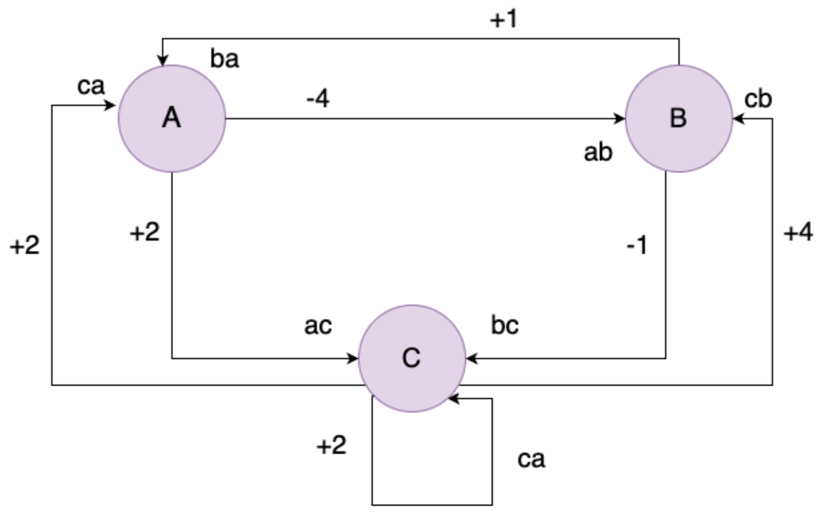
b) **[4 pts]** Suppose that for a given search problem you design a heuristic that overestimates the cost-to-goal by at most ε. In the worst case, how far from optimal will be the solution path found by A* when using this heuristic? Explain briefly your reasoning.

In the worst case, the solution path found may cost ε more than the optimal one.

Explanation: The optimal path may be overestimated by at most ε and therefore the algorithm would not find a solution that costs more than the optimal solution + ε

Question 3 [12 pts] Consider the following Markov Decision Process (MDP) with discount factor γ = 0.9. Upper case letters A, B, C represent states, edges represent the actions (lower case letters ab, ac, ba, ca, cb), and signed integers represent rewards. The reward function is conditioned on the current state and action only (for example, performing action ab at state A produces a reward of -4 irrespective of the next state). The transition probabilities are as follows: P(B|A, ab) = 1, P(C|A, ac) = 1, P(A|B, ba) = 1, P(C|B,bc) = 1, P(A|C,ca) = 3/4, P(C|C,ca) = 1/4, P(B|C,cb) = 1.

Hint: Bellman's equation is $V_t(s) = \max_a R(s,a) + \gamma \Sigma_{s'} \Pr(s'|s,a) V_{t-1}(s')$



a) **[6 pts]** Consider a policy $\pi_1(s)$ that provides $\pi_1(A)$ = ab, $\pi_1(B)$ = bc and $\pi_1(C)$ = ca actions from state s. Assume that we are running an iterative policy evaluation to determine the value of this policy. Also assume that we are starting with an initial value function of $V_0(A) = V_0(B) = V_0(C) = 1$, where the subscript denotes each iterative step of iterative policy evaluation. Now compute the values obtained for two iterations of iterative policy evaluation (i.e., one value for each state for every iteration) to compute a new value function $V_1(s)$ and $V_2(s)$.

Iteration 1:

$$V_1(A) = R(A, ab) + \gamma \sum_{s'} Pr(s'|s, \pi(s)) V^{\pi}(s') = -4 + (0.9 \times V_0(B)) = -3.1$$

$$V_1(B) = R(B, bc) + \gamma \sum_{s'} Pr(s'|s, \pi(s)) = -1 + 0.9 \times V_0(C) = -0.1$$

$$V_1(C) = R(C, ca) + \gamma \sum_{s'} Pr(s'|s, \pi(s)) V^{\pi}(s') = 2 + 0.9 \times (0.25 \times V_0(C) + 0.75 \times V_0(A))$$
$$= 2.9$$

Iteration 2:

$$V_2(A) = R(A, ab) + \gamma \sum_{s'} Pr(s'|s, \pi(s)) V^{\pi}(s') = -4 + (0.9 \times V_1(B)) = -4.09$$

$$V_2(B) = R(B, bc) + \gamma \sum_{s'} Pr(s'|s, \pi(s)) = -1 + 0.9 \times V_1(C) = 1.61$$

$$V_2(C) = R(C, ca) + \gamma \sum_{s'} Pr(s'|s, \pi(s)) V^{\pi}(s') = 2 + 0.9 \times (0.25 \times V_1(C) + 0.75 \times V_1(A)) = 0.56$$

Question 3 (continued)

b) **[3 pts]** Apply one iteration of greedy policy improvement to compute a new, deterministic policy $\pi_2(s)$ assuming that $V_1(A) = V_1(B) = 2$ and $V_1(C) = 5$ (subscript denotes policy iteration steps) are the converged values obtained at the end of iterative policy evaluation.

Solution:

$$\pi(A) = \mathrm{arg}\underset{a}{max}(R(A, ab) + \gamma \sum_{s'} Pr(s'|s, ab)V(s'), R(A, ac) + \gamma \sum_{s'} Pr(s'|s, ac)V(s'))$$

$$= \mathrm{arg}\underset{a}{max}(-4 + 0.9 \times 2{,}2 + 0.9 \times 5)$$

$$= \mathrm{arg}\underset{a}{max}(-2.2{,}6.5) = ac$$

$$\pi(B) = \mathrm{arg}\underset{a}{max}(R(B, ba) + \gamma \sum_{s'} Pr(s'|s, ba)V(s'), R(B, bc) + \gamma \sum_{s'} Pr(s'|s, bc)V(s'))$$

$$= \mathrm{arg}\underset{a}{max}(1 + 0.9 \times 2, -1 + 0.9 \times 5)$$

$$= \mathrm{arg}\underset{a}{max}(2.8{,}3.5) = bc$$

$$\pi(C) = \mathrm{arg}\underset{a}{max}(R(B, ca) + \gamma \sum_{s'} Pr(s'|s, ba)V(s'), R(B, cb) + \gamma \sum_{s'} Pr(s'|s, bc)V(s'))$$

$$= \mathrm{arg}\underset{a}{max}(2 + 0.9 \times (0.25 \times 5 + 0.75 \times 2){,}4 + 0.9 \times 2)$$

$$= \mathrm{arg}\underset{a}{max}(4.475{,}5.8) = cb$$

c) **[3 pts]** Apply one iteration of value iteration (i.e., compute one value for each state) to compute a new value function $V_2(s)$ assuming $V_1(A) = V_1(B) = V_1(C) = 2$.

Solution

$V_2(A) = max(R(A, ab) + \gamma \times \sum_{s'} Pr(s'|s, ab)V_1(s'), R(A, ac) + \gamma \times \sum_{s'} Pr(s'|s, ac)V_1(s'))$
$= max(-4 + 0.9 \times 2,2 + 0.9 \times 2) = 3.8$

$V_2(B) = max(R(B, ba) + \gamma \times \sum_{s'} Pr(s'|s, ba)V_1(s'), R(B, bc) + \gamma \times \sum_{s'} Pr(s'|s, bc)V_1(s'))$
$= max(1 + 0.9 \times 2, -1 + 0.9 \times 2) = 2.8$

$V_2(C) = max(R(C, ca) + \gamma \times \sum_{s'} Pr(s'|s, ba)V_1(s'), R(C, cb) + \gamma \times \sum_{s'} Pr(s'|s, bc)V_1(s'))$
$= max(2 + 0.9 \times 2,4 + 0.9 \times 2) = 5.8$

Question 4 [12 pts] Reinforcement Learning

a) **[4 pts]** Consider a system with two states (s_1 and s_2) and two actions (a_1 and a_2). Perform tabular Q-learning using a fixed learning rate of $\alpha = 0.8$ and a discount factor of $\gamma = 0.9$ for each step. The state-action value table entries Q(s, a) are initialized to zero. For each step the current state, action, reward, and next state are given as (s_k, a_k, r_k, s_{k+1}). Note that rewards are obtained immediately upon execution of an action at a state (independent of the next state). Here is the experience the agent has in the first four iterations respectively:

(s_1, a_1, 5, s_1)
(s_1, a_2, 10, s_2)
(s_2, a_1, 5, s_1)
(s_1, a_1, −5, s_2)

Find the value for Q(s_1, a_1), Q(s_1, a_2), Q(s_2, a_1) and Q(s_2, a_2) using the experiences above by iteratively updating Q-values for each experience using the tabular Q-learning algorithm. Each update uses the most recent Q-estimates in the Q-table for the updates.

Hint: Temporal difference update equation: $Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$

Use the first experience (s_1, a_1, 5, s_1) for this update
Q(s_1, a_1) =

$$Q(s_1,a_1) = Q(s_1,a_1) + \alpha(r + \gamma \max_{a'} Q(s_1,a') - Q(s_1,a_1))$$
$$= 0 + 0.8(5 + 0.9 \times 0 - 0) = 4$$

Use the second experience (s_1, a_2, 10, s_2) for this update
Q(s_1, a_2) =

$$Q(s_1,a_2) = Q(s_1,a_2) + \alpha(r + \gamma \max_{a'} Q(s_2,a') - Q(s_1,a_2))$$
$$= 0 + 0.8(10 + 0.9 \times 0 - 0) = 8$$

Use the third experience (s_2, a_1, 5, s_1) for this update
Q(s_2, a_1) =

$$Q(s_2,a_1) = Q(s_2,a_1) + \alpha(r + \gamma \max_{a'} Q(s_1,a') - Q(s_2,a_1)$$
$$= 0 + 0.8(5 + 0.9 \times 8 - 0) = 9.76$$

Use the fourth experience (s_1, a_1, −5, s_2) for this update
$Q(s_1, a_1) =$

$$Q(s_1, a_1) = Q(s_1, a_1) + \alpha(R + \gamma \times \max_{a'} Q(s_2, a') - Q(s_1, a_1))$$

$$= 4 + 0.8(-5 + 0.9 \times 9.76 - 4) = 3.8272$$

Question 4 (continued)

b) **[8 pts]** Consider the following pseudo-code for an implementation of the Q-learning algorithm for a discrete state and action space environment:

```
Algorithm: Q-learning
-------------------------------------------
1: Set t = 0, get the initial state $s_0$.
2: For all states s, and actions $a \in A$, let $Q_0(s, a) = 0$. Let $\varepsilon = 0.05$ and $\alpha = 0.1$
3: Define policy derived from Q to return a random action with probability $\varepsilon$, and a
   greedy action (based on current Q-estimates) with probability $1 - \varepsilon$
4: while Q has not converged do
5:     Select and execute action a at current state s using the policy derived from Q
6:     Observe the next state s′ and reward r
7:     Update Q-values: $Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$
8:     Let $s \leftarrow s'$
9: end while
```

i) [2 pts] The pseudo-code has a while loop that tests for convergence. Would the Q-values converge as it is? If not, what changes would you do to make sure that the pseudo-code would converge?

This algorithm will not converge as the learning rate is fixed. To fix the problems, update the learning rate using the formula $\alpha \leftarrow 1/n(s,a)$ where $n(s,a)$ denotes the number of times the current state s and action a have been visited.

ii) [2 pts] Let's consider other exploration techniques instead of ε-greedy. Mention two other exploration strategies you could consider.

The two other techniques are Boltzmann exploration and Upper confidence bound (UCB).

Question 4 (continued)

Algorithm: Q-learning (repeated for convenience)

```
------------------------------------------
1: Set $t = 0$, get the initial state $s_0$.
2: For all states $s$, and actions $a \in A$, let $Q_0(s, a) = 0$. Let $\varepsilon = 0.05$ and $\alpha = 0.1$
3: Define policy derived from Q to return a random action with probability $\varepsilon$, and a
   greedy action (based on current Q-estimates) with probability $1 - \varepsilon$
4: while Q has not converged do
5:        Select and execute action a at current state s using the policy derived from Q
6:        Observe the next state $s'$ and reward r
7:        Update Q-values:  $Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$
8:        Let $s \leftarrow s'$
9: end while
```

iii) [2 pts] Let's modify the implementation of Q-learning to return the transition and reward model of the environment as well. What additional lines of code would you insert between lines 6 and 7 in the provided pseudo-code to update the transition and the reward model?

Solution:
This would require a model-based algorithm. At each time step in the learning loop, an estimate of the transition model and reward model will be maintained and updated using empirical averages.

Transition Model: $Pr(s'|s, a) \overset{\square}{\leftarrow} \frac{n(s,a,s')}{n(s,a)}$

Reward Model: $R(s, a) \overset{\square}{\leftarrow} \frac{r+(n(s,a)-1)R(s,a)}{n(s,a)}$

iv) [2 pts] Let's extend this Q-learning implementation to a continuous state space environment. What Q-function representation would you use to allow Q-learning to work for continuous states?

Neural networks or some function approximator.

Question 5 [12 pts] Game Theory

There are two players, Alice and Bob, who are trying to decide between choosing to play football (F), baseball (B) or hockey (H). Here, Alice is the row player and Bob is the column player. They are in a two-player normal form game with utilities as shown in Table 1. Using this table answer the questions that follow.

	F	B	H
F	(2, 7)	(1, -4)	(-1, -1)
B	(2, 4)	(2, 4)	(-2, 5)
H	(-1, -1)	(4, 0)	(7, 2)

Table 1: Two-player normal form game

a) **[3 pts]** Are there any dominated strategies for the two players? If so, perform iterative elimination of dominated strategies until there are no more dominated strategies. Draw the resulting normal form game after each iterative elimination.

Solution:

For Bob, playing strategy B is strictly dominated by strategy H. The resulting normal form game after elimination is as follows:

	F	H
F	(2, 7)	(-1, -1)
B	(2, 4)	(-2, 5)
H	(-1, -1)	(7, 2)

In this resulting game, Alice has strategy B to be weakly dominated by F. The resulting normal form game after elimination is as follows:

	F	H
F	(2, 7)	(-1, -1)
H	(-1, -1)	(7, 2)

Question 5 (continued)

b) **[3 pts]** Consider the following normal form game with two players Alice and Bob, where Alice is the row player and Bob is the column player. Each player can choose between two actions C and D. Their utilities are provided in the table below.

	C	D
C	(6, 6)	(2, 8)
D	(8, 2)	(0, 0)

Table 2: Two-player normal form game

In Table 2, are there any pure strategy Nash equilibria? If so, write down all the pure strategy Nash equilibria, with your explanation for why these are pure strategy Nash equilibria.

Solution

There are two pure strategy Nash equilibria in this game. They are {C,D} and {D, C}. In other words, D is the best response of Bob when Alice plays C and C is the best response of Bob when Alice plays D and vice-versa.

c) **[3 pts]** In Table 2, are there any Pareto optimal outcomes? If so, write down all the Pareto optimal outcomes with your explanation for why these are Pareto optimal outcomes.

Solution:

(C,C), (C,D) and (D,C) are pareto optimal outcomes since no other outcome pareto dominates these outcomes.

Question 5 (continued)

d) **[3 pts]** Find the mixed strategy Nash equilibrium in the normal form game given in Table 2. Also report the utilities that Alice and Bob will get at the mixed strategy Nash equilibrium.

	C	D
C	(6, 6)	(2, 8)
D	(8, 2)	(0, 0)

Table 2: Two-player normal form game (repeated for convenience)

Let us assume that Alice plays C with probability p and Bob plays C with probability q.

If Bob has to be indifferent between its two actions:

$u_{Bob}(C) = 6p + (1-p)2 = 4p + 2$
$u_{Bob}(D) = 8p + (1-p)0 = 8p$
$\Longrightarrow p = 1/2$

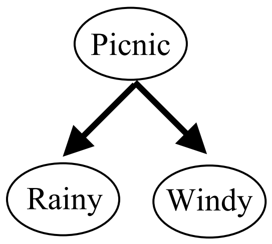
If Alice has to be indifferent between its two actions:

$u_{Alice}(C) = 6q + (1-q)2 = 4q + 2$
$u_{Alice}(D) = 8q = 8q$
$u_{Alice}(D) = 8q = 8q$
$\Longrightarrow q = 1/2$

Here the mixed strategy NE is obtained when Alice plays C with probability 1/2 and D with probability 1/2. In this case, Bob plays C with probability 1/2 and D with probability 1/2. The utility of Bob is 4 and the utility of Alice is also 4 at the mixed strategy NE.

Question 6 [12 pts] Consider the problem of deciding whether or not to go on a picnic based on different attributes of the day. Here is a set of examples classified based on whether or not it was a good idea to go on a picnic. In this question you will use maximum likelihood to learn the parameters of the naïve Bayes model besides the table.

Example	Rainy	Windy	Warm	Summer	Sunday	Picnic
X_1	T	F	F	F	F	False
X_2	F	T	F	F	T	True
X_3	F	T	T	T	T	False
X_4	F	T	T	F	T	False
X_5	T	F	F	F	T	False
X_6	F	T	F	F	T	False



a) **[4 pts]** Compute the maximum likelihood (ML) hypothesis for the naïve Bayes model besides the table above. More precisely, below, fill in the conditional probability tables of the maximum likelihood hypothesis. Do not use add-one Laplace smoothing.

P(Picnic)	Picnic	
	T	F
	1/6=0.17	**5/6=0.83**

P(Rainy|Picnic)		Rainy	
		T	F
Picnic	True	**0**	**1**
	False	**2/5=0.4**	**3/5=0.6**

P(Windy|Picnic)		Windy	
		T	F
Picnic	True	**1**	**0**
	False	**3/5=0.6**	**2/5=0.4**

Question 6 (continued)

b) **[4 pts]** Using the maximum likelihood hypothesis computed in a), classify examples X_1 and X_2. More precisely, compute P(Picnic|Rainy,Windy) for X_1 and X_2 and select the class with the highest probability.

Classification of X_1:

$$P(P=T|R=T,W=F) \propto P(P=T)P(R=T|P=T)P(W=F|P=T)$$
$$=\left(\frac{1}{6}\right)(0)(0)=0$$

$$P(P=F|R=T,W=F) \propto P(P=F)P(R=T|P=F)P(W=F|P=F)$$
$$=\left(\frac{5}{6}\right)\left(\frac{2}{5}\right)\left(\frac{2}{5}\right)=\frac{2}{15}$$

Hence: $P(P=T|R=T,W=F)=\frac{0}{0+\frac{2}{15}}=0$ **and the prediction is Picnic=False**

Classification of X_2:

$$P(P=T|R=F,W=T) \propto P(P=T)P(R=F|P=T)P(W=T|P=T)$$
$$=\left(\frac{1}{6}\right)(1)(1)=\frac{1}{6}$$

$$P(P=F|R=F,W=T) \propto P(P=F)P(R=F|P=F)P(W=T|P=F)$$
$$=\left(\frac{5}{6}\right)\left(\frac{3}{5}\right)\left(\frac{3}{5}\right)=\frac{3}{10}$$

Hence: $P(P=T|R=T,W=F)=\frac{\frac{1}{6}}{\frac{1}{6}+\frac{3}{10}}=\frac{5}{14}=0.36$ **and the prediction is Picnic=False**

c) **[4 pts]** In the naïve Bayes model besides the table above, are Rainy and Windy independent given Picnic? Justify. Is this independence/dependence assumption realistic?

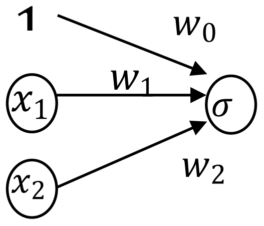
Rainy and Windy are independent of each other in the Naïve Bayes model.

According to the rules of d-separation (rule 2), observing picnic blocks the path between rainy and windy.

This independence assumption is not realistic since clearly wind and rain are correlated irrespective of whether we go on a picnic or not.

Question 7 [12 pts]

a) **[6 pts]** Consider the following sigmoid perceptron.



i) **[2 pts]** What is the output of this sigmoid perceptron for the inputs $x_1 = 3, x_2 = 0.5$? Tip: $\sigma(a) = \frac{1}{1+e^{-a}}$

$$\boldsymbol{\sigma}\big(1(-0.2) + 3(0.1) + 0.5(-0.2)\big) = \boldsymbol{\sigma}(0) = 0.5$$

ii) **[3 pts]** Suppose the correct output is $y = 0.1$. You'd like to train this perceptron by minimizing the squared error (i.e., $E = 0.5(y - output)^2$) with gradient descent. What is the partial derivative of the squared error with respect to weight w_1?
Tip: $\sigma'(a) = \sigma(a)(1 - \sigma(a))$

$$\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{w_1}} = 0.5(2)(\boldsymbol{y} - \boldsymbol{output})\frac{\partial \boldsymbol{Output}}{\partial \boldsymbol{w_1}}$$
$$= (\boldsymbol{y} - \boldsymbol{output})(\boldsymbol{\sigma}(\boldsymbol{a})\big(1 - \boldsymbol{\sigma}(\boldsymbol{a})\big)\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{w_1}}$$
$$= (\boldsymbol{y} - \boldsymbol{output})(\boldsymbol{\sigma}(\boldsymbol{a})\big(1 - \boldsymbol{\sigma}(\boldsymbol{a})\big)\boldsymbol{x_1}$$
$$= (0.1 - 0.5)(\boldsymbol{\sigma}(0.5)\big(1 - \boldsymbol{\sigma}(0.5)\big)3 = -0.1165$$

Question 7 (continued)

b) **[6 pts]** The “gradient vanishing” problem is a serious issue when training deep neural networks by backpropagation.

i) [3 pts] Describe the gradient vanishing problem

Gradient vanishing corresponds to the situation where the gradient gets smaller and smaller as we go down the neural network, to the point where the bottom layers are barely updated because the gradient of their weights is near 0.

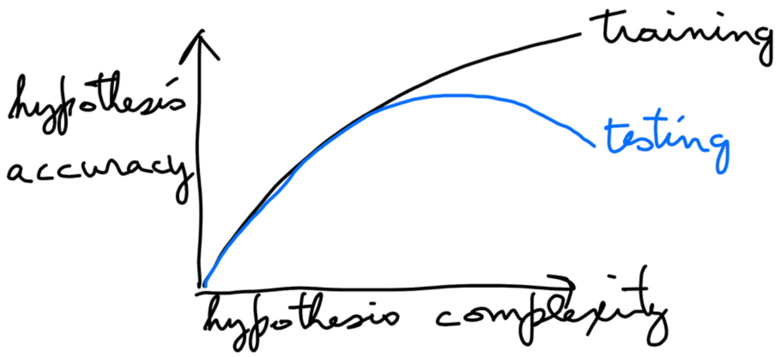
ii) [3 pts] Describe an approach to mitigate the gradient vanishing problem.

Several approaches are possible. One approach is sufficient for full marks. The following approaches are acceptable (other approaches may also be acceptable):

- **Use rectified linear units (ReLU)**
- **Use batchnorm or layernorm**
- **Pre-train the lower layers**
- **Use skip connections (also known as residual connections)**

Question 8 [12 pts] Overfitting is a common problem in machine learning.

a) **[4 pts]** Draw a graph illustrating the overfitting phenomenon. More precisely, the x-axis of your graph should correspond to *hypothesis complexity* and the y-axis should correspond to *hypothesis accuracy*. Draw two curves in your graph showing how the accuracy of hypotheses vary with their complexity for the training set and the test set when overfitting occurs.



b) [2 pts] Explain one possible cause of overfitting.

There are several possible causes. Only one cause is sufficient for full marks, but the following two causes are acceptable (other causes may also be acceptable):

- **Noise in the data (as a result the training set is not representative of the test set)**
- **Insufficient data (this could be insufficient training data, test data or both).**

c) [3 pts] Name a technique to prevent overfitting in decision tree learning.

Several techniques are possible. Only one technique is sufficient for full marks. The following two techniques are acceptable (other techniques may also be acceptable):

- **Prune statistically insignificant nodes using a statistical test such as the chi-squared test**
- **Do not split nodes with fewer data points than some threshold that is optimized by cross-validation**
- **Stop growing the tree early when cross-validation accuracy starts decreasing**

d) **[3 pts]** Name another technique to prevent overfitting in deep learning.

Several techniques are possible. Only one technique is sufficient for full marks. The following two techniques are acceptable (other techniques may also be acceptable):

- **Use dropout**
- **Use regularization (i.e., add a term to the loss function that penalizes large weights)**
- **Stop training early, i.e., when validation accuracy starts decreasing**