EXAM 24T2 FCruz

Note for current students: This exam is an old version taken in-person in a similar course. It is given to you just to study the concepts and not as an example of the coming exam in our course.

To be consider before starting:

- The time allotted for the exam is 2 hours, plus 15 minutes.
- The exam contains 12 multiple-choice questions.
- Mark varies for each question as shown in the summary table.
- Total marks are 50, worth 50% of the total marks for the course.
- There is a hurdle in the exam of 40% equivalent to 20 marks.
- This exam cannot be copied, forwarded, or shared in any way.
- Students have been reminded of the UNSW rules regarding Academic Integrity and Plagiarism.
- You must complete all of your work within the exam time. There is no extra time. No late submissions will be accepted.
- The only materials you can use during the exam are textbooks or notes (hard-copy only) and a UNSW-approved calculator.

The summary of questions is as follows:

#	Topic	Question	Marks
1	Knowledge representation	Predicate logic	4
2	Neural networks	Normalisation	4
3	Neural networks	Forward propagation	6
4	Search	DFS	3
5	Reinforcement learning	Temporal difference	4
6	Reinforcement learning	Action selection	4
7	Optimisation	Genetic algorithm	4
8	Computer vision	Stereo vision	4
9	Language processing	Grammar parsing	3
10	Language processing	Minimum edit distance	6
11	Uncertain reasoning	Bayes nets	6
12	Intelligent robotics	Explainability	2
	Total		50

TOPIC: Knowledge representation – Logic [4 marks]

Consider the following predicate logic expressions. Which one are true?

1. **The predicate logic of "There exists a student who has taken both calculus and linear algebra" is $\exists x(S(x) \wedge C(x) \wedge L(x))$ where**
   S(x) represents "x is a student."
   C(x) represents "x has taken calculus."
   L(x) represents "x has taken linear algebra."

2. **The predicate logic of "Every person who owns a car has a driver's license" is $\forall x(P(x) \wedge O(x) \wedge D(x))$ where** $\forall x(P(x) \wedge O(x) \rightarrow D(x))$
   P(x) represents "x is a person."
   O(x) represents "x owns a car."
   D(x) represents "x has a driver's license."

3. **The predicate logic of "Some employees who work more than 40 hours a week do not receive overtime pay" is $E(x) \wedge W(x) \wedge \neg O(x)$ where** $\exists x(E(x) \wedge W(x) \wedge \sim O(x))$
   E(x) represents "x is an employee."
   W(x) represents "x works more than 40 hours a week."
   O(x) represents "x receives overtime pay."

4. **The predicate logic of "If any customer buys a product and the product is defective, then the customer will return the product" is $\forall x(C(x) \wedge P(x) \wedge D(x) \rightarrow R(x))$ where**
   C(x) represents "x is a customer."
   P(x) represents "x buys a product."
   D(x) represents "The product bought by x is defective."
   R(x) represents "x will return the product."

a. Both 1 and 4 are true
b. 2, 3 and 4 are true
c. Both 1 and 2 are true
d. All 1,2,3,4 are true
e. None

TOPIC: Neural Networks – Normalisation [4 marks]

Consider the following dataset that needs to be normalised using min-max normalization to the range [-1, 1]. Using the min-max normalisation, what are the values of x_1 and x_2 for the training example c?

Training example	x_1	x_2
a	5	50
b	10	60
c	15	70
d	20	80

a. $x_1 = 0.3$, $x_2 = 0.3$
b. $x_1 = 0.5$, $x_2 = 0.5$
c. $x_1 = 1.0$, $x_2 = 1.0$
d. $x_1 = -0.5$, $x_2 = -0.5$
e. $x_1 = -0.3$, $x_2 = 0.25$

$$x_n = \frac{2(x - x_{min})}{x_{max} - x_{min}} - 1$$

$$x_1 = \frac{2(15-5)}{20-5} - 1 = \frac{4}{3} - 1 = \frac{1}{3}$$

$$x_2 = \frac{2(70-50)}{80-50} - 1 = \frac{4}{3} - 1 = \frac{1}{3}$$

Explanation:

Since the range is [-1, 1], the normalization formula to be used is,

$$X_n = 2*(X-X_{min})/(X_{max}-X_{min}) - 1;\ X_n \in [-1,1]$$

- For x_1:

$X_{min} = 5$ (example a) ; $X_{max} = 20$ (example d)

Value for c = 15

Normalized x_1 = 2 * (15 - 5) / (20 - 5) - 1

= 2 * (10 / 15) - 1

= 2 * (2/3) - 1 = 4/3 - 1

= 1/3 ≈ 0.33

- For x2:

$X_{min} = 50$ (example a) $X_{max} = 80$ (example d)

Value for c = 70

Normalized x_2 = 2 * (70 - 50) / (80 - 50) - 1

= 2 * (20 / 30) - 1

= 2 * (2/3) - 1

= 4/3 - 1

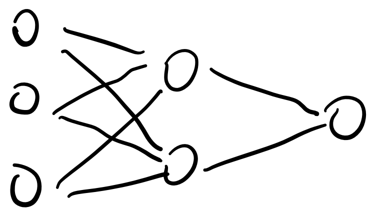
= 1/3 ≈ 0.33

TOPIC: Neural Networks – Forward propagation [6 marks]

Given the following neural network architecture and weights, what is the approximate output of the network for the input $x = [1, 0, 1]$?

Architecture:

- Input Layer: 3 neurons
- Hidden Layer: 2 neurons with ReLU activation function
- Output Layer: 1 neuron with $sigmoid$ activation function



Weights:

- Input to Hidden Layer weights:

$$W_i = \begin{pmatrix} 0.2 & 0.4 \\ 0.6 & 0.5 \\ 0.1 & 0.3 \end{pmatrix}$$

- Hidden Layer biases: $b_i = [0.1, 0.2]$
- Hidden to Output Layer weights: $W_o = [0.7, 0.9]$
- Output Layer bias: $b_o = 0.5$

Note:

- $ReLU$ activation function: $ReLU(x) = max(0, x)$
- $Sigmoid$ activation function: $\sigma(x) = \frac{1}{1+e^{-x}}$

a. 1.13
b. 0.34
c. -0.09
d. 0.83
e. -1.06

Explanation:

```
Input Vector X: [1 0 1]

Input Weights Wi:

[[0.2 0.4]

 [0.6 0.5]

 [0.1 0.3]]

Input Biases bi: [0.1 0.2]

Output Weights Wo: [0.7 0.9]

Output Bias bo: 0.5
```

$$z_i = W_i^T x + b_i$$

$$= \begin{pmatrix} 0.2 & 0.6 & 0.1 \\ 0.4 & 0.5 & 0.3 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$$

$$= \begin{pmatrix} 0.3 \\ 0.7 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$$

$$= \begin{pmatrix} 0.4 \\ 0.9 \end{pmatrix}$$

$$a_i = ReLU(z_i) = \begin{pmatrix} 0.4 \\ 0.9 \end{pmatrix}$$

$$z_o = w_o^T a_i + b_o$$

$$= (0.7 \;\; 0.9) \begin{pmatrix} 0.4 \\ 0.9 \end{pmatrix} + 0.5$$

$$= 1.59$$

$$\sigma(1.59) = 0.83$$

```
Detailed Multiplication between Input Vector and Wi:

Neuron 1:
  X[0] * Wi[0,0] = 1 * 0.2 = 0.20
  X[1] * Wi[1,0] = 0 * 0.6 = 0.00
  X[2] * Wi[2,0] = 1 * 0.1 = 0.10
  Sum = 0.30
  Sum + Bias = 0.40

Neuron 2:
  X[0] * Wi[0,1] = 1 * 0.4 = 0.40
  X[1] * Wi[1,1] = 0 * 0.5 = 0.00
  X[2] * Wi[2,1] = 1 * 0.3 = 0.30
  Sum = 0.70
  Sum + Bias = 0.90
...
zo = np.dot(a1, Wo) + bo: 1.5899999999999999
a2 = Sigmoid(zo) = 0.8306161030659813

The output of the network for the input [1, 0, 1] is approximately 0.831
```

TOPIC: Search – DFS [3 marks]

Given the following pseudocode for a DFS algorithm, what is the missing condition that prevents cycles?

```
def dfs(graph, start, goal, visited=set()):
    if start == goal:
        return [start]
    visited.add(start)
    for neighbor in graph[start]:
        # <<<add missing condition here>>>
            path = dfs(graph, neighbor, goal, visited)
            if path:
                return [start] + path
    return []
```

a. Check if the neighbour is equal to the goal.
b. Check if the start node is already in the visited set.
c. Check if the graph contains cycles.
d. Check if the neighbour is not in the visited set before making the recursive call.
e. Check if the goal node has been reached.

Explanation:

```
def dfs(graph, start, goal, visited=set()):
    if start == goal:
        return [start]
    visited.add(start)
    for neighbour in graph[start]:
        if neighbour not in visited:
            path = dfs(graph, neighbour, goal, visited)
            if path:
                return [start] + path
    return []
```

TOPIC: Reinforcement learning – TD prediction [4 marks]

Consider an agent using Temporal Difference (TD) learning to estimate the value function V for states S1 and S2. The agent follows a policy π that chooses actions based on the current state. The observed transitions and rewards are:

- Transition from S1 to S2 with a reward R = 4
- Transition from S2 back to S1 with a reward R = 1

Assume the current value estimates are V(S1) = 2 and V(S2) = 3. Using a learning rate α = 0.5 and a discount factor γ = 0.9, update the value of V(S1) using the TD(0) method, as follows:

$$V(s) \leftarrow V(s) + \alpha[R + \gamma V(s') - V(s)]$$

What is the updated value of V(S1)?

$V(S1) = 2 + 0.5(4 + 0.9(3) - 2)$

$= 4.35$

a. 2.95

b. 3.45

c. 4.35

d. 3.50

e. 4.50

Explanation:

Given:

- V(s1) = 2 (current estimate)
- V(s2) = 3
- R = 4 (reward for transition from s1 to s2)
- α = 0.5 (learning rate)
- γ = 0.9 (discount factor)

For s1:

V(s1) = 2 + 0.5[4 + 0.9 × 3 − 2]

= 2 + 0.5[4 + 2.7 − 2]

= 2 + 0.5[4.7]

= 2 + 2.35

= 4.35

TOPIC: Reinforcement learning – Action selection [4 marks]

Consider two reinforcement learning agents using the ε-greedy action selection method with the following features:

- The first agent has an initial ε = 0.5. Then, it follows a strategy where epsilon decreases by 0.1 after, every 100 epochs.
- The second agent has ε = 0.6 which remains fixed during the learning process.
- Both agents draw a random number using a uniform distribution.

Considering the times each agent will perform exploration during the first 500 episodes, what is the difference between the number of times agent 2 explores and the number of times agent 1 explores?

Agent 1: (0.5 + 0.4 + 0.3 + 0.2 + 0.1) × 100
= 150
Agent 2: 0.6 × 500 = 300
difference = 300 − 150 = 150

a. 100
b. 150
c. 590
d. 50
e. 450

Explanation:

agent 1 = (0.5 * 100) + (0.4 * 100) + (0.3 * 100) + (0.2 * 100) + (0.1 * 100)
= 50+40+30+20+10
= 150

Agent 2 = 0.6 * 500 = 300
Diff = 150

TOPIC: Optimisation – Genetic algorithm [4 marks]

Consider two chromosomes in a genetic algorithm where symbol ‘|’ marks the crossover point. The fitness function is defined as the number of ‘1’ in a chromosome.

Chromosome 1: 1100 | 1010

Chromosome 2: 1010 | 1011

- What are the fitness values of original chromosomes?
- What would be the two offspring or children after the crossover?
- What are the fitness scores of both offsprings?
- Perform bit flip mutations at positions 2 and 5 on both offspring with 0 based index from left to right. What are the new offsprings?
- An offspring is considered fit if its fitness score > 5. How many of the offspring after mutation are fit?

a.

Original fitness values: 4, 5

Offsprings after crossover: 11001011, 10101010

Fitness score after crossover: 5, 4

Mutated offspring: 11101111, 10001110

Number of fit offspring: 1

b.

Original fitness values: 4, 5

Offsprings after crossover: 11011011, 11110101

Fitness score after crossover: 6, 6

Mutated offspring: 11111111, 11010011

Number of fit offspring: 2

c.

Original fitness values: 4, 5

Offsprings after crossover: 11011011, 10101010

Fitness score after crossover: 6, 4

Mutated offspring: 11111111, 10001110

Number of fit offspring: 1

d.

Original fitness values: 5, 4

Offsprings after crossover: 11001011, 11110101

Fitness score after crossover: 5, 6

Mutated offspring: 10110011, 10111101

Number of fit offspring: 2

e.

Original fitness values: 4, 5

Offsprings after crossover: 11001011, 10101010

Fitness score after crossover: 5, 4

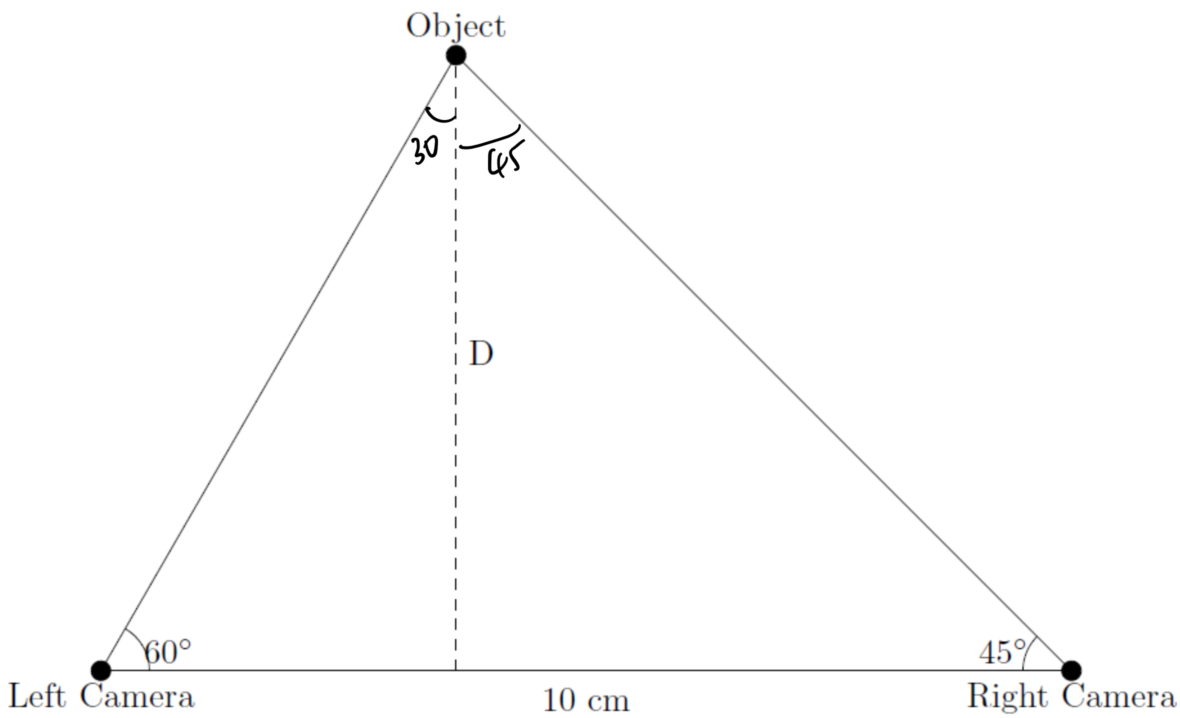
Mutated offspring: 11101111, 10101110

Number of fit offspring: 2

TOPIC: Computer vision – Stereo vision [4 marks]

Suppose there is a stereo camera system consisting of 2 cameras (like our eyes), with the distance between the centres of the 2 cameras being 10 cm. An object is located at the same height as these cameras. We measure the angles between the line of sight to the object and the line connecting to another camera as follows:

- Angle to the left camera (θ_L) = 60°
- Angle to the right camera (θ_R) = 45°



What is the distance D from the cameras to the object using the stereo camera setup?

$$\tan 30 + \tan 45 = \frac{10}{D}$$

$$D = \frac{10}{\tan 30 + \tan 45} = 6.34$$

a. 3.66
b. 6.34
c. 3.17
d. 1.83
e. 5.00

Explanation:

To find the distance to the object, we can use the formula for triangulation:

$D = B / \tan(\theta_L) + \tan(\theta_R)$

where:

• D is the distance from the cameras to the object.

• B is the baseline distance between the cameras (10 cm).

- θ_L is the angle to the left camera (30°) (90° - 60° in this case).
- θ_R is the angle to the right camera (45°) (90°- 45° in this case).

Then, we calculate the tangent of each angle:
$\tan(\theta_L) = \tan(\pi/6) = \approx 0.57$
$\tan(\theta_R) = \tan(\pi/4) = 1$

Now, we can calculate the distance:
$D = 10 \text{ cm} / (0.57 + 1) \approx 6.34 \text{ cm}$

So, the distance from the eyes to the object is approximately 6.34 cm.

TOPIC: Language processing – Grammar parsing [3 marks]

Consider the following sentence:

> "I like apples and oranges when they are ripe"

Considering that:

- "I" and "they" are Pronouns (PR)
- "like" and "are" are Verbs (VB)
- "apples" and "oranges" are Nouns (N)
- "and" and "when" are Conjunctions (CJ)
- "ripe" is an Adjective (ADJ)

Which of the following four grammars (considering only derivations to non-terminal symbols) can parse the given sentence?

PR VB N CJ N CJ PR VB ADJ

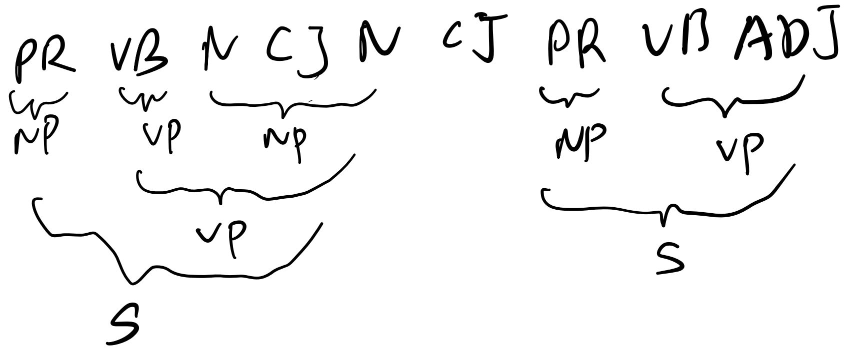
a)
S -> NP VP
NP -> PR | N | N CJ N
VP -> VB ADJ | VB NP

PR VB ADJ ✗

b)
S -> NP VP | S CJ S
NP -> PR | N
VP -> VB ADJ | VB N

NP VP CJ NP VP
= PR VB N CJ N ✗

c)
S -> NP VP | S CJ S
NP -> PR | N | N CJ N
VP -> VB ADJ | VB NP



d)
S -> NP VP | S CJ S
NP -> PR | N | N N
VP -> VB ADJ | VB CJ NP

Explanation:

S -> **S CJ S**
S CJ S -> **NP VP** CJ S
NP VP CJ S -> **PR** VP CJ S
PR VP CJ S -> ***I*** VP CJ S
I **VP** CJ S -> *I* **VB NP** CJ S
I **VB** NP CJ S -> *I* ***like*** NP CJ S
I like **NP** CJ S -> *I like* **N CJ N** CJ S
I like **N CJ N** CJ S -> *I like* ***apples and oranges*** CJ S
I like apples and oranges **CJ** S -> *I like apples and oranges* ***when*** S
I like apples and oranges when **S** -> *I like apples and oranges when* **NP VP**
I like apples and oranges when **NP** VP -> *I like apples and oranges when* **PR** VP

I like apples and oranges when **PR** VP -> *I like apples and oranges when* ***they*** VP
I like apples and oranges when they **VP** -> *I like apples and oranges when they* **VB NP**
I like apples and oranges when they **VB** NP -> *I like apples and oranges when they* ***are*** NP
I like apples and oranges when they are **NP** -> *I like apples and oranges when they are* **PR**
I like apples and oranges when they are **PR** -> *I like apples and oranges when they are* ***ripe***

TOPIC: Language processing – Minimum edit distance [6 marks]

Using the Levenshtein distance equation, what is the minimum edit distance between the words “robot” and “orbit”? Use the following operations and their associated costs:

- Insertion: cost 1
- Deletion: cost 1
- Substitution: cost 2

T	5	4	5	4	5	4
I	4	3	4	3	4	5
B	3	2	3	2	3	4
R	2	1	2	3	4	5
O	1	2	1	2	3	4
""	0	1	2	3	4	5
	""	R	O	B	O	T

a. 2
b. 4
c. 5
d. 3
e. 1

Explanation:

T	5	4	5	4	5	4
I	4	3	4	3	4	5
B	3	2	3	2	3	4
R	2	1	2	3	4	5
O	1	2	1	2	3	4
#	0	1	2	3	4	5
	#	R	O	B	O	T

TOPIC: Uncertain reasoning – Bayes nets [6 marks]

Suppose a certain medical test for a disease is known to have a false positive rate of 5% and a false negative rate of 10%. The prevalence of the disease in the population is 1%. If a person tests positive, what is the probability P1 that they actually have the disease? If the false negative rate increases to 15%, recalculate the probability P2 that a person actually has the disease given a positive test result. If the false positive rate decreases to 1%, while keeping the true positive rate at 90%, what is the new probability P3 that a person has the disease given a positive test result?

let p = positive
d = disease

$P_1 = P(p|d)$

$P(d) = 0.01$

a. P1 = 0.1421, P2 = 0.1311, P3 = 0.452
b. P1 = 0.1235, P2 = 0.1121, P3 = 0.386
c. P1 = 0.1538, P2 = 0.1466, P3 = 0.476
d. P1 = 0.1692, P2 = 0.1532, P3 = 0.491
e. P1 = 0.1312, P2 = 0.1022, P3 = 0.397

$P(p \mid \sim d) = 0.05$

$P(\sim p \mid d) = 0.1$

$d \to \sim p|d,\ p|d$; $\sim d$

$P_1 = P(d|p) = \dfrac{P(d \cap p)}{P(p)} = \dfrac{P(p|d)P(d)}{P(p)}$

$\dfrac{0.9 \times 0.01}{0.0585}$

$P(p|d) = 1 - P(\sim p|d) = 0.9$

$P(p) = P(d \cap p) + P(\sim d \cap p)$
$= 0.9 \times 0.01 + 0.05 \times 0.99$
$= 0.0585$

Explanation:

P(disease) = 0.01 (given)
P(positive | No disease) = 0.05 (given)
P(No positive | disease) = 0.10 (given)
P(disease | positive) = ??

P(disease | positive) = P(positive | disease)*P(disease) / P(positive)

=> P(positive | disease) = 1 - P(No positive | disease) = 1 - 0.1 = 0.90

=> P(positive) = P(Positive| Disease)*P(Disease) + P(Positive | No Disease)*P(No Disease)
= (0.90*0.01)+(0.05*0.99)
= 0.0585

Putting values in P(disease | positive) formula, P1(disease | positive) = 0.90 * 0.01 / 0.0585
= 0.1538

=> P(positive | disease) = 1 - P(No positive | disease) = 1 - 0.15 = 0.85
=> P(positive) = P(Positive| Disease)*P(Disease) + P(Positive | No Disease)*P(No Disease)
= (0.85*0.01)+(0.05*0.99)
= 0.058

Putting values in P(disease | positive) formula, P2(disease | positive) = 0.85 * 0.01 / 0.058
= 0.1466

=> P(positive | disease) = 1 - P(No positive | disease) = 1 - 0.1 = 0.90
=> P(positive) = P(Positive| Disease)*P(Disease) + P(Positive | No Disease)*P(No Disease)

= (0.90*0.01)+(0.01*0.99)
= 0.0189

Putting values in P(disease | positive) formula,

P3(disease | positive) = 0.90 * 0.01 / 0.0189
= 0.476

TOPIC: Intelligent robotics – Explainability [2 marks]

Consider an intelligent robot using reinforcement learning to find the exit of a maze autonomously. While navigating the maze, in a specific state, the robot faced three possible actions: turning to the left, turning to the right, or keeping going straight. The robot decided to turn to the left. At that moment, a human user required an explanation of why this action had been taken. Which of the following is a better explanation to be given by the robot?

a. I chose turning to the left because maximizes future collected reward.
b. I chose turning to the left because this action has the highest Q-value.
c. I chose turning to the left because this action gave the highest probability to find the exit.
d. I chose turning to the left because I performed an exploratory action.
e. I chose turning to the left because was the next action following the optimal policy.