USN | | | | | | | | | |

CMRIT

Internal Assessment Test 2 – Dec. 2021

<table>
<tr><td>Sub:</td><td colspan="5">ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING</td><td>Sub Code:</td><td>18CS71</td><td>Branch:</td><td colspan="3">CSE</td></tr>
<tr><td>Date:</td><td>16/12/2021</td><td>Duration:</td><td>90 mins</td><td>Max Marks:</td><td>50</td><td>Sem / Sec:</td><td colspan="3">7/A,B,C</td><td colspan="2">OBE</td></tr>
<tr><td colspan="9"><u>Answer any FIVE FULL Questions</u></td><td>MARKS</td><td>CO</td><td>RBT</td></tr>
<tr><td>1</td><td colspan="8">Explain the concept of decision tree learning. Construct the decision tree for the instances given in Table below.</td><td>[10]</td><td>CO2</td><td>L2</td></tr>
</table>

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

3 + 7

- Decision Trees are a type of Supervised Machine Learning
- The data is continuously split according to a certain parameter.
- The tree can be explained by two entities, namely decision nodes and leaves.
- Decision Tree Learning is one of the widely used practical method for inductive inference
- inductive inference is based on a generalization from a finite set of past observations, extending the observed pattern or relation to other future instances or instances occurring elsewhere

Answer:

Entropy of Dataset = - [(9/4) log (9/14) + (5/14). Log (5/14) = 0.941

Entropy(S,outlook) = - (5/14) * [(2/5) log (2/5) + (3/5) log(3/5)

- 4/14 * 0

- (5/14) * [(2/5) log (2/5) + (3/5) log(3/5)

= 0.693

Information Gain(S,outlook) = 0.940- 0.693 = 0.246

Entropy(S, Temp) = - (4/14) * -1

- (6/14) *[(2/6) log(2/6) + (4/6) log (4/6)]

- (4/14) * [(1/4)log(1/4) + (3/4) log(3/4)]

= 0.911

Information Gain(S, Temp = 0.940 = 0.911 = 0.029

Entropy(S, Humidity) = - (7/14) * [(4/7) log(4/7) + (3/7) log(3/7)]

- (7/14) * [(6/7) log(6/7) + (1/7) log(1/7)

= 0.789

Information Gain(S, Humidity) = 0.940-0.789 = 0.151

Entropy(S, Wind) = - (6/14) * (-1)

- (8/14) * [(6/8) log(6/8) + (2/8) log(2/8)]

= 0.892

Info Gain(S, Wind) = 0.940-0.892 = 0.048

Attribute "Outlook" has maximum info gain. Hence root of the tree will be "Outlook"

Nodes at next Level

1. "Sunny" branch

Humidity divides the subset of samples perfectly and has entropy of 0. Hence humidity will be the node at the "Sunny" branch

2. "Overcast" branch

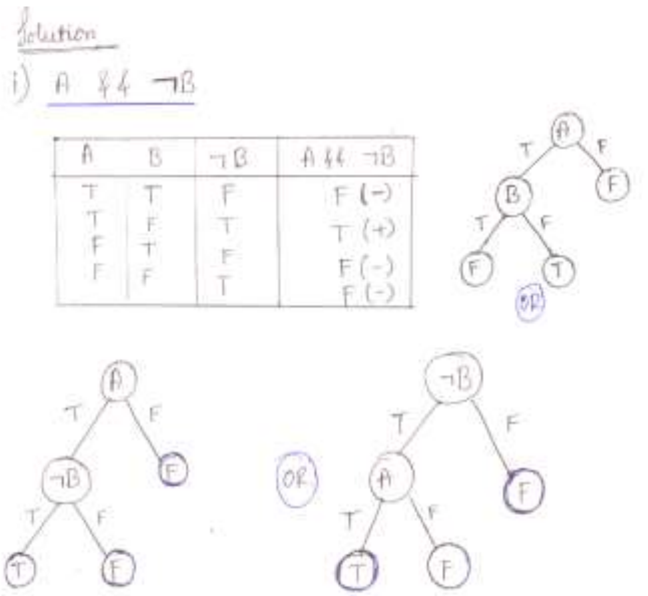
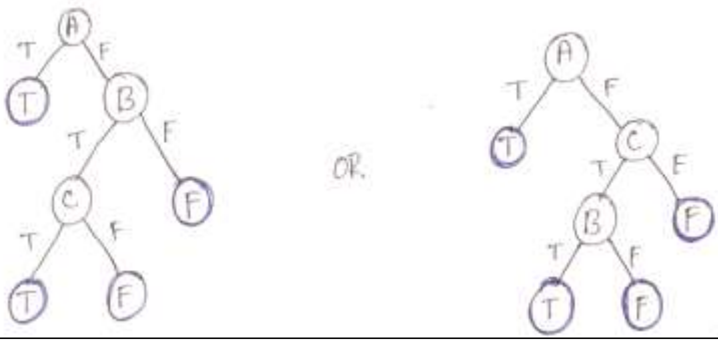
All samples are "Yes" class. Hence a leaf node with "Yes" is at this level

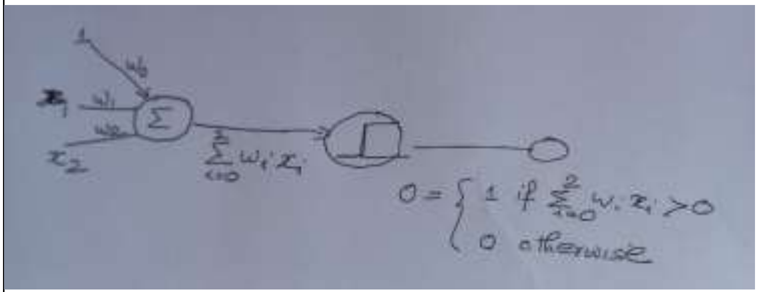
3. "Rain" branch

Wind divides the subset of samples perfectly and has entropy of 0. Hence Wind will be the node at the "Rain" branch

Nodes at level 3

1. (Outlook=Sunny, Humidity=High) branch
   Leaf node "No"
2. (Outlook=Sunny, Humidity=Normal) branch
   Leaf node "Yes"
3. (Outlook=Rain, Wind=Weak) branch
   Leaf node "Yes"
4. (Outlook=Rain, Wind=Strong) branch
   Leaf node "No"

2 (2)	Construct the decision tree to represent the following Boolean functions:i) A ∧ ¬ B ii) A∨ [B∧ C]	[05]	CO2	L2
	i) A ∧ ¬ B  ii) A∨ [B∧ C] 	2.5+2.5		
2 (b)	How a single perceptron can be used to represent the Boolean functions such as AND, OR.	[05]	CO2	L2

	 Many values of w0, w1 and w2 possible for AND and OR nodes to represent AND and OR nodes. AND Node: w0= -0.6, w1=0.5, w2=0.5 OR Node: w0 = -0.4, w1=0.5, w2=0.5	2.5+2.5		
3	Explain the back propagation algorithm. Why is it not likely to be trapped in local minima?	[7+3]	CO2	L2
	BACKPROPAGATION (*training_example*, η, n_{in}, n_{out}, n_{hidden}) *Each training example is a pair of the form ($\vec{x}$, $\vec{t}$), where ($\vec{x}$) is the vector of network input values, ($\vec{t}$) and is the vector of target network output values.* *η is the learning rate (e.g., .05). n_i is the number of network inputs, n_{hidden} the number of units in the hidden layer, and n_{out} the number of output units.* *The input from unit i into unit j is denoted x_{ji}, and the weight from unit i to unit j is denoted w_{ji}* • Create a feed-forward network with n_i inputs, n_{hidden} hidden units, and n_{out} output units. • Initialize all network weights to small random numbers • Until the termination condition is met, Do • For each ($\vec{x}$, $\vec{t}$), in training examples, Do *Propagate the input forward through the network:* 1. Input the instance $\vec{x}$, to the network and compute the output o_u of every unit u in the network.	7+3		

Propagate the errors backward through the network:

2. For each network output unit k, calculate its error term δ_k

$$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$$

3. For each hidden unit h, calculate its error term δ_h

$$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{h,k}\delta_k$$

4. Update each network weight w_{ji}

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

Where

$$\Delta w_{ji} = \eta \delta_j x_{i,j}$$

Large number of dimensions can provide an escape route from local minima. The sigmoid function which has small slope for large positive and negative values of linear sum also helps in avoiding local minima in most cases.

4 (a)	Write Gradient descent algorithm to train a linear unit along with the derivation	[05]	CO2	L2
		2.5+2.5		

GRADIENT-DESCENT($training\_examples, \eta$)

Each training example is a pair of the form $\langle \vec{x}, t \rangle$, where $\vec{x}$ is the vector of input values, and t is the target output value. η is the learning rate (e.g., .05).

- Initialize each w_i to some small random value
- Until the termination condition is met, Do
  - Initialize each Δw_i to zero.
  - For each $\langle \vec{x}, t \rangle$ in $training\_examples$, Do
    - Input the instance $\vec{x}$ to the unit and compute the output o
    - For each linear unit weight w_i, Do

      $$\Delta w_i \leftarrow \Delta w_i + \eta(t - o)x_i$$
  - For each linear unit weight w_i, Do

    $$w_i \leftarrow w_i + \Delta w_i$$

Derivation

$$\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i}\frac{1}{2}\sum_d (t_d - o_d)^2 \\
&= \frac{1}{2}\sum_d \frac{\partial}{\partial w_i}(t_d - o_d)^2 \\
&= \frac{1}{2}\sum_d 2(t_d - o_d)\frac{\partial}{\partial w_i}(t_d - o_d) \\
&= \sum_d (t_d - o_d)\frac{\partial}{\partial w_i}(t_d - \vec{w} \cdot \vec{x}_d) \\
\frac{\partial E}{\partial w_i} &= \sum_d (t_d - o_d)(-x_{i,d}) \qquad \text{equ. (6)}
\end{aligned}$$

Substituting Equation (6) into Equation (5) yields the weight update rule for gradient descent

	$$\Delta w_i = \eta \sum_{d \in D} (t_d - o_d)\, x_{id} \qquad \text{equ. (7)}$$			
4 (b)	Write the characteristics of the problems appropriate for solving using neural network.	[05]	CO1	L1
	ANN is appropriate for problems with the following characteristics •Instances are represented by many attribute value pairs. • The ta r g e t f unction o u t p ut m ay be discr e t e v a lued, rea l v alued, or a ve c tor of several real or discrete valued attributes. •The training examples may contain errors. •Long training times are acceptable. •Fast evaluation of the learned target function may be required •The ability of humans to understand the learned target function is not important	05 marks – one per characteristics		
5	Show the derivation of back propagation training rule for output unit weights.	[10]	CO2	L2
	For each training example d every weight $\boldsymbol{w_{ji}}$ is updated by adding to it $\boldsymbol{\Delta w_{ji}}$ $$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} \quad \text{.........equ. (1)}$$ where, E_d is the error on training example d, summed over all output units in the network $$E_d(\vec{w}) \equiv \frac{1}{2} \sum_{k \in output} (t_k - o_k)^2$$ **derive an expression for $\frac{\partial E_d}{\partial w_{ji}}$ in order to implement the stochastic gradient descent rule seen in Equation $\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}}$** **notice that weight w_{ji} can influence the rest of the network only through net_j.** Use chain rule to write $$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} x_{ji} \quad \text{......equ(2)}$$ Derive a convenient expression for $\frac{\partial E_d}{\partial net_j}$	10		

	$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j}\frac{\partial o_j}{\partial net_j}$ .....equ(3) To begin, consider just the first term in Equation (3) $\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j}\frac{1}{2}\sum_{k\in outputs}(t_k - o_k)^2$ The derivatives $\frac{\partial}{\partial o_j}(t_k - o_k)^2$ will be zero for all output units k except when $k = j$. We therefore drop the summation over output units and simply set $k = j$. $\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j}\frac{1}{2}(t_j - o_j)^2$ $= \frac{1}{2}2(t_j - o_j)\frac{\partial(t_j - o_j)}{\partial o_j}$ $= -(t_j - o_j)$ ......equ (4) 222 Next consider the second term in Equation (3). Since $o_j = \sigma(net_j)$, the derivative $\frac{\partial o_j}{\partial net_j}$ is just the derivative of the sigmoid function, which we have already noted is equal to $\sigma(net_j)(1 - \sigma(net_j))$. Therefore, $\frac{\partial o_j}{\partial net_j} = \frac{\partial \sigma(net_j)}{\partial net_j}$ $= o_j(1 - o_j)$ ....equ(5) Substituting expressions (4) and (5) into (3), we obtain $\frac{\partial E_d}{\partial net_j} = -(t_j - o_j)\ o_j(1 - o_j)$ ....equ(6) and combining this with Equations (1) and (2), we have the stochastic gradient descent rule for output units $\Delta w_{ji} = -\eta\frac{\partial E_d}{\partial w_{ji}} = \eta\ (t_j - o_j)\ o_j(1 - o_j)x_{ji}$ ....equ (7)			
6(a)	Explain the various approaches for knowledge representation	[05]	CO1	L1
	Different techniques for knowledge representation Simple relational knowledge Inheritable knowledge Inferential knowledge Procedural knowledge	5marks for 4 rep techniques		
6(b)	Discuss the issues in knowledge representation	[05]	CO1	L1
	• Are there any basic attributes of objects? • Are there any basic relationships among objects? • At what level should knowledge be represented? • How should sets be represented? • How should knowledge be accessed?	5 marks for min 5 issues		

CO PO Mapping

Course Outcomes		Blooms Level	Modules covered	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	Appraise the theory of Artificial intelligence and Machine Learning	**L1,L2,L3**	**1,2,3, 4,5**	2	3	1	1	-	1	-	-	-	-	-	2	1	0	1	3
CO2	Illustrate the working of AI and ML Algorithms	**L1,L2,L3**	**1,2,3, 4,5**	2	3	1	1	-	1	-	-	-	-	-	2	1	0	1	3
CO3	Demonstrate the applications of AI and ML	**L1,L2,L3**	**1,23,4 ,5**	2	3	1	1	1	2	-	-	-	-	-	2	1	0	1	3

COGNITIVE LEVEL	REVISED BLOOMS TAXONOMY KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PROGRAM OUTCOMES (PO), PROGRAM SPECIFIC OUTCOMES (PSO)				CORRELATION LEVELS	
PO1	Engineering knowledge	PO7	Environment and sustainability	0	No Correlation
PO2	Problem analysis	PO8	Ethics	1	Slight/Low
PO3	Design/development of solutions	PO9	Individual and team work	2	Moderate/ Medium
PO4	Conduct investigations of complex problems	PO10	Communication	3	Substantial/ High
PO5	Modern tool usage	PO11	Project management and finance		
PO6	The Engineer and society	PO12	Life-long learning		
PSO1	Develop applications using different stacks of web and programming technologies				
PSO2	Design and develop secure, parallel, distributed, networked, and digital systems				
PSO3	Apply software engineering methods to design, develop, test and manage software systems.				
PSO4	Develop intelligent applications for business and industry				