University of Asia Pacific

Department of CSE

Name: Rashik Rahman

Reg ID: 17201012

Year: 4th

Semester: 1st

Course Code: CSE 403

Course Title: Artificial Intelligence

Date: 25.04.2021

"During Examination and upload time I will not take any help from anyone. I will give my exam all by myself."

University of Asia Pacific

Admit Card

Final-Term Examination of Fall, 2020

Financial Clearance	PAID

Registration No : 17201012

Student Name : Rashik Rahman

Program : Bachelor of Science in Computer Science and Engineering

SI.NO.	COURSE CODE	COURSE TITLE	CR.HR.	EXAM. SCHEDULE
1	CSE 400	Project / Thesis	3.00	
2	CSE 330	Industrial Training	1.50	
3	CSE 401	Mathematics for computer Science	3.00	
4	CSE 403	Artificial Intelligence and Expert Systems	3.00	
5	CSE 404	Artificial Intelligence and Expert Systems Lab	1.50	
6	CSE 405	Operating Systems	3.00	
7	CSE 406	Operating Systems Lab	1.50	
8	CSE 407	ICTLaw, Policy and Ethics	2.00	
9	CSE 410	Software Development	1.50	
10	CSE 427	Topics of Current Interest	3.00	
		Total Credit:	23.00	

1. Examinees are not allowed to enter the examination hall after 30 minutes of commencement of examination for mid semester examinations and 60 minutes for semester final examinations.

2. No examinees shall be allowed to submit their answer scripts before 50% of the allocated time of examination has elapsed.

3. No examinees would be allowed to go to washroom within the first 60 minutes of final examinations.

4. No student will be allowed to carry any books, bags, extra paper or cellular phone or objectionable items/incriminating paper in the examination hall.
Violators will be subjects to disciplinary action.

This is a system generated Admit Card. No signature is required.

Admit Card Generation Time: 25-Apr-2021 02:11 AM

17201012

Answer to the Q. No. 1(a)

A back-propagation neural network is a multi-layer network and the layers are fully connected, that is every neuron in each layer is connected to every other neuron in the adjacent forward layer. As it is a fully connected so we can refer it as a sequential network.

Back-propagation is basically an algorithm of propagating the total loss back into the neural network to know how much of the loss every node is responsible for, and subsequently updating the weights.

(2)
17201012

Answer to the Q. NO. 1-(b)

Given,

~~$x_1 = 0$~~

$$\left.\begin{aligned} x_1 &= 1 \\ x_2 &= 0 \\ x_3 &= 1 \end{aligned}\right\} \text{inputs}$$

$y = 1 \} \rightarrow$ output

$\theta = 0.2 \rightarrow$ threshold

$\alpha = 0.1 \rightarrow$ learning rate

Now,

$w_1 = 12\%2 - 0.2 = -0.2$

$w_2 = 12\%3 - 0.5 = -0.5$

$w_3 = 0.1$

(i) $\hat{y} = \text{step}\left[\sum_{i=1}^{3} x_i w_i - \theta\right]$

$\Rightarrow \hat{y} = \text{step}[x_1 w_1 + x_2 w_2 + x_3 w_3 - \theta]$

$= \text{step}[1 \times (-0.2) + 0 \times (-0.5) + 1 \times 0.1 - 0.2]$

$= \text{step}[-0.2 + 0.1 - 0.2]$

$= \text{step}[-0.3]$

$\therefore \hat{y} = 0$

$$step[x] = \begin{cases} 0 ; & \text{if } x < 0 \\ 1 ; & \text{if } x \geq 0 \end{cases}$$

ii

error, $e = y - \hat{y} = 1 - 0 = 1$

$w_1 = w_1 + \cancel{\alpha x_1 + e}\ \alpha x_1 e$

$= -0.2 + 0.1 \times 1 \times 1 = -0.1$

$\cancel{= 1 - 0.1 = 0.9}$

$w_2 = w_2 + \cancel{\alpha x_2 + e}\ \alpha x_2 e$

$= -0.5 + \cancel{0.1 \times (-0.5)}\ 0.1 \times 0 \times 1$

$= -0.5$

$w_3 = w_3 + \cancel{\alpha x_3 + e}\ \alpha x_3 e$

$= 0.1 + 0.1 \times 1 \times 1 = 0.2$

$\cancel{= 1 + 0.2 = 1.2}$

$\therefore$ New updated weights, $(w_1, w_2, w_3) = (-0.1, -0.5, 0.2)$ $\cancel{(0.9, 0.5, 1.2)}$

Ans

4

17201012

Answer to the Q. No. 2(a)

$$P(X \mid Y, Z) \; ; \; X \perp\!\!\!\perp Y \mid Z$$

Query variable   hidden variable   evidence variable

i) Query variable: The variable which's ~~probabili~~ probability is needed to know.

ii) Hidden variable: The variable that is irrelevant to query variable and independent from it, thus can be excluded from the equation.

iii) Evidence variable: The variable based on which we calculate the probability of query variable. It is a given event based on it query variable's probability is calculated.

Answer to the Q.No. 2 (b)

Now,

~~$q = 12 \div 1000 =$~~

$q = 12/1000 = 0.012$

$p = \sqrt{12}/100 = 0.035$

$u = 12/10000 = 0.0012$

$t = 1 - q = 1 - 0.012 = 0.988$

$R = 1 - p = 1 - 0.035 = 0.965$

$s = 1 - u = 1 - 0.0012 = 0.998$

Transition matrix, A =

$$X_{t-1} \quad \begin{array}{c} \text{Pizza} \\ \text{Burger} \\ \text{Hotdog} \end{array} \begin{bmatrix} 0.988 & 0.012 & 0 \\ 0.035 & 0 & 0.965 \\ 0.0012 & 0 & 0.998 \end{bmatrix}$$
$$\qquad\qquad\quad \text{pizza} \quad \text{burger} \quad \text{hotdog}$$
$$X_t$$

given, $P(X_1) = [\underset{\text{pizza}}{0}, \underset{\text{burger}}{1}, \underset{\text{hotdog}}{0}]$

⑥
17201012

$$P(X_2 = pizza) = P(X_2 = pizza \mid X_1 = pizza)\, P(X_1 = pizza)$$
$$+ P(X_2 = pizza \mid X_1 = burger)\, P(X_1 = burger)$$
$$+ P(X_2 = pizza \mid X_1 = hotdog)\, P(X_1 = hotdog)$$

$$= 0.988 \times 0 + 0.035 \times 1 + 0.0012 \times 0$$

$$= 0.035$$

$$P(X_2 = burger) = P(X_2 = burger \mid X_1 = pizza)\, P(X_1 = pizza)$$
$$+ P(X_2 = burger \mid X_1 = burger)\, P(X_1 = burger)$$
$$+ P(X_2 = burger \mid X_1 = hotdog)\, P(X_1 = hotdog)$$

$$= 0.012 \times 0 + 0 \times 1 + 0 \times 0$$

$$= 0$$

$$P(X_2 = hotdog) = P(X_2 = hotdog \mid X_1 = pizza)\, P(X_1 = pizza)$$
$$+ P(X_2 = hotdog \mid X_1 = burger)\, P(X_1 = burger)$$
$$+ P(X_2 = hotdog \mid X_1 = hotdog)\, P(X_1 = hotdog)$$

$$= 0 \times 0 + 0.965 \times 1 + 0.998 \times 0$$

$$= 0.965$$

$$\therefore P(X_2) = [\underset{pizza}{0.035}, \underset{burger}{0}, \underset{hotdog}{0.965}]$$

Scanned with CamScanner

$\therefore P(X_3 = pizza) = P(X_3 = pizza \mid X_2 = pizza)\, P(X_2 = pizza)$
$+ P(X_3 = pizza \mid X_2 = burger)\, P(X_2 = burger)$
$+ P(X_3 = pizza \mid X_2 = hotdog)\, P(X_2 = hotdog)$

$= 0.988 \times 0.035 + 0.035 \times 0 + 0.0012 \times 0.965$

$= \cancel{0.035 +}\ 0.0346 + 0 + \cancel{0.001158}\ 0.001158$

$= 0.0358$

$\therefore$ The predicted probability that they will serve pizza on 3^{rd} day is 0.0358 on 3.58%.

Ans.

Scanned with CamScanner

8

17201012

Answer to the Q. No. 3(a)

A ~~deg~~ linguistic hedge or modifier is an operation that modifies the meaning of a term or fuzzy set. Hedges are some word that modify the linguistic variable of a fuzzy set.

Hedges are basically modifiers that gives a degree of membership to a fuzzy set.

Example:

If we consider pressure as a fuzzy set then hedges will be very pressure, less pressure, ~~extem~~ extremly pressre.

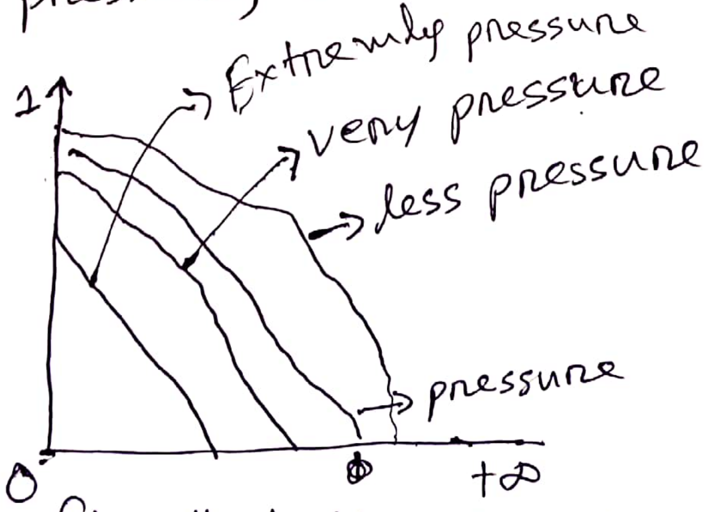


Fig: illustration of hedge with respect to fuzzy ~~so~~ values.

Answer to the Q. No. 3(b)

Now,

$A = 012/1000 = 0.012$

$B = 012/2000 = 0.006$

$C = 012/4000 = 0.003$

$D = 1 - A = 1 - 0.012 = 0.988$

$E = 1 - B = 1 - 0.006 = 0.994$

very good programmer membership = [0.012, 0.006, 0.003, 0.988, 0.994]

we know,

$good = \sqrt{very\ good}$

$\therefore$ extremely good = $(good)^3$

$\therefore$ more or less good = $(good)^{\frac{1}{2}}$

so,

good programmer membership = $[\sqrt{0.012}, \sqrt{0.006}, \sqrt{0.003}, \sqrt{0.988}, \sqrt{0.994}]$

$= [\cancel{0.035}\ 0.11, 0.077, 0.055, 0.994, 0.997]$

(10)

17201012

(i)

extremly good membership $= [(0.11)^3, (0.077)^3, (0.055)^3, (0.994)^3, (0.997)^3]$

$= [0.001, 0.0005, 0.0001, 0.982, 0.991]$

(ii)

more or less good membership $= [(0.11)^{\frac{1}{2}}, (0.077)^{\frac{1}{2}}, (0.055)^{\frac{1}{2}}, (0.994)^{\frac{1}{2}}, (0.997)^{\frac{1}{2}}]$

$= [~~0.0~~ 0.33, 0.28, 0.23, 0.997, 0.998]$

Graphical representation:

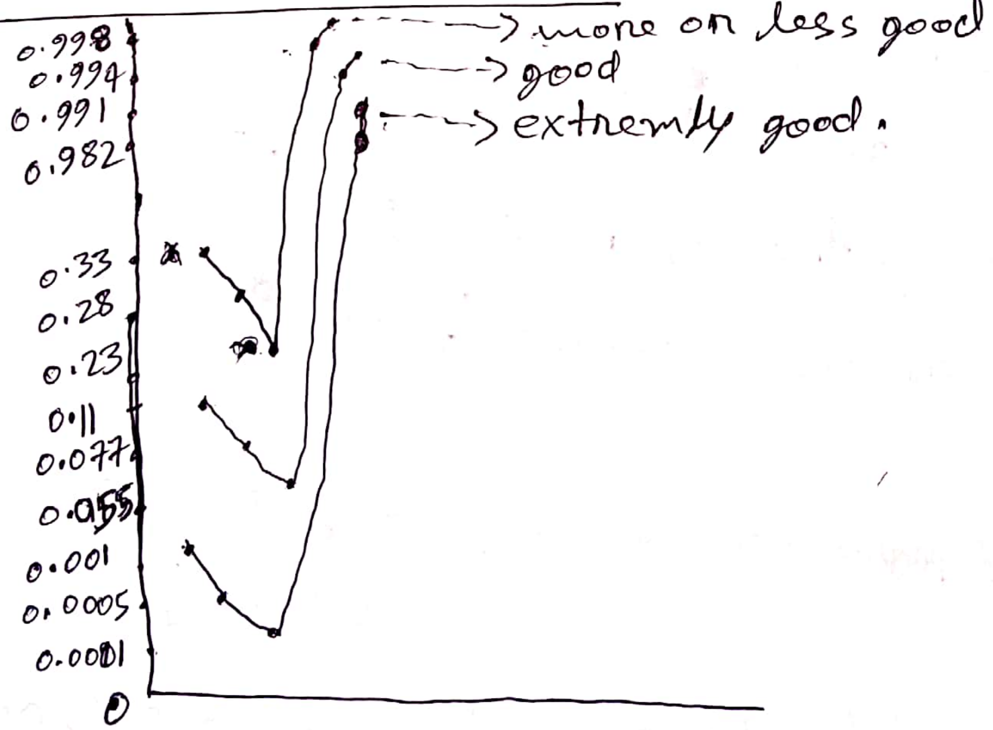


Fig: graphical representation.

Answer to the Q No. 4

$T_1 = 12\% 3 + 2 = 2$

$T_2 = -7$

$T_3 = -2$

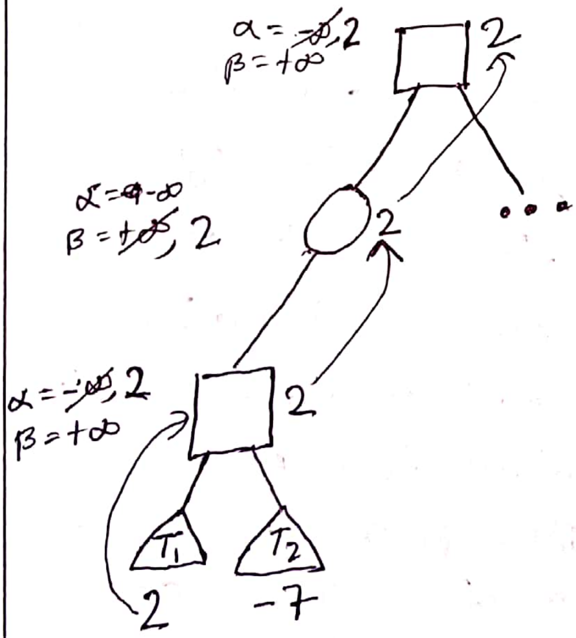
$T_4 = T_1 + 3 = 2 + 3 = 5$

Let,

$\alpha = -\infty$

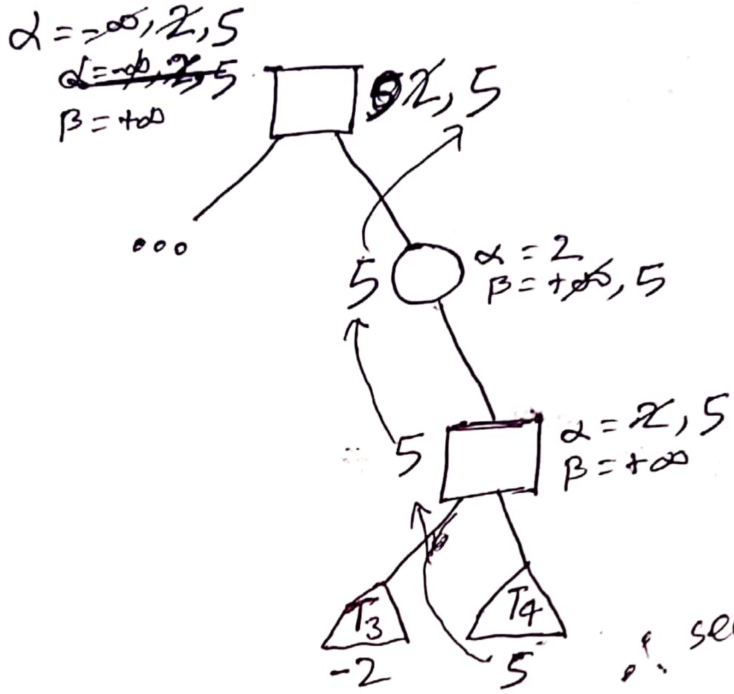
$\beta = +\infty$

Left side:



(12)
17201012

Right side:



$\therefore$ search result = 5.

Explaination:

The base MiniMax ~~pruning~~ search algorithm is very slow considering a complex problem. Because in complex problem branching factors are very high. This can be solved by alpha-beta pruning. It is an optimizer technique that reduces the time complexity of MiniMax pruning. It is basically a modified version of MiniMax search in which we can eliminate or cut of a part of tree in such a way that the end result doesn't change. Here we take two parameters α and β and the prune condition is $\alpha \geq \beta$.

The alpha-beta pruning to a standard minimax algorithm returns the ~~max~~ same move as the standard algorithm but it removes all the nodes which are not really ~~xx~~ affecting the final decision but making algorithm slow.

So we can conclude alpha-beta pruning reduces the search space, time complexity ~~the~~ of the search in game tree. Thus alpha-beta prunings makes the search very fast when compared to minimax algorithm.

~~The final tree stands~~