Digital Signal Processing 23/7/24

D.S.P

Soft Computing — A system based on the concepts of introducing the input, output & processing with finding the solution of a praticular problem such that the outcome may not have the exact solution, that computing for the solution is called soft computing.

The characteristics of soft computing are imprecise, uncertainity, partially truth, approximation.

The distribution of hard and soft computing

Hard Computing
↳ Precise model
↓
→ Symbolic, logic reasoning
→ Traditional numeric modeling & search

Soft Computing
↳ Approximate model
↓
→ Approximate reasoning
→ Functional, randomized search

Constitutes of soft computing

i) Fuzzy Systems
ii) Evalutionary Computing
iii) Neural network

Fuzzy
↓
soft

Applications of soft computing

i) Hand written recognization

ii) Image Processing & computer vision.
iii) Automotive systems & manufacturing
iv) Decision support system
v) Power systems
vi) Fuzzy logic controller
vii) Process controller
viii) Speech & recognization system etc.

Fuzzy system

A fuzzy system is based on fuzzy theory. A fuzzy set A is defined as

$$\boxed{A = \{(x, \mu_A(x))\}}$$

$\mu_A(x)$ = membership grade [membership function] value (to calculate m-value / grade)

- the chances lies b/w 0 to 1.

$$0 \le \mu_A(x) \le 1$$

eg I will go to school

i.e x will go to A

A = {("a", 0.3), ("b", 0.5), ("e", 0.0)}
name chances

Basic Operations of a fuzzy set

i) Union
ii) Intersection
iii) Complement

suppose there are 2 sets A, B

$A = \{(x, \mu_A(x))\}$
$B = \{(x, \mu_B(x))\}$

i) $\boxed{\mu_{(A \cup B)} = \max\{\mu_A(x), \mu_B(x)\}}$

ii) $\boxed{\mu_{(A \cap B)} = \min(\mu_A(x), \mu_B(x)\}}$

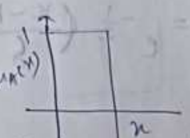
iii) $\boxed{\mu_{\bar{A}} = \{1 - \mu_A(x)\} \quad \mu_{\bar{B}} = \{1 - \mu_B(x)\}}$

Eg
$A = \{(x_1, 0.5), (x_2, 0.7), (x_3, 0.1)\}$
$B = \{(x_1, 0.8), (x_2, 0.9), (x_4, 0.3)\}$
$A \cup B = \{(x_1, 0.8), (x_2, 0.9), (x_3, 0.1), (x_4, 0.3)\}$
$A \cap B = \{(x_1, 0.5), (x_2, 0.3), (x_3, 0.9)\}$
$\bar{B} = \{(x_1, 0.2), (x_2, 0.1), (x_4, 0.7)\}$

The common membership functions

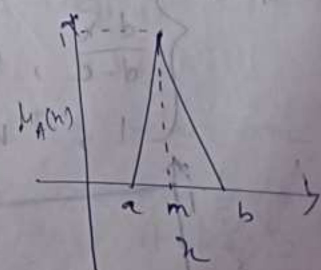
i) Singleton MF

$$\mu_A(x) = \begin{cases} 1, & \text{if } x = a \\ 0, & \text{otherwise} \end{cases}$$

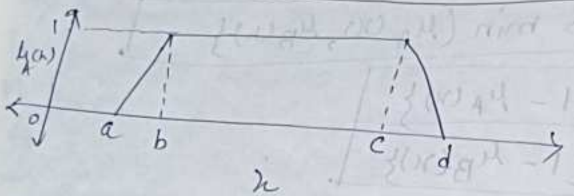


ii) Triangular membership function

$$\mu_A(x) = \begin{cases} 0, & \text{if } x \le a \\ \dfrac{x-a}{m-a}, & a < x \le m \\ \dfrac{b-x}{b-m}, & m < x < b \\ 0 & x \ge b \end{cases}$$



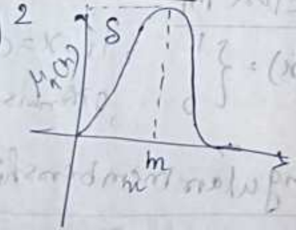
iii) Trapizoidal membership function



$$\mu_A(x) = \begin{cases} 0, & x \le a \text{ or } x \ge d \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \frac{d-x}{d-c}, & c \le x \le d \\ 0, & x \ge d \end{cases}$$

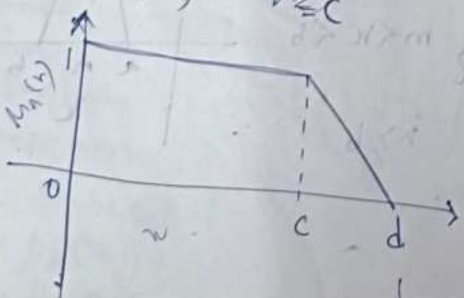
iv) Gaussian membership func

$$\mu_A(x) = e^{-\frac{1}{2}\left(\frac{x-m}{\delta}\right)^2}$$



v) Z-shaped MF

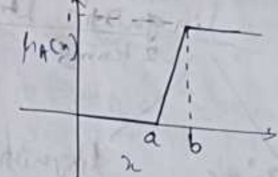
$$\mu_A(x) = \begin{cases} 0, & x \ge d \\ \frac{d-x}{d-c}, & c \le x \le d \\ 1, & x \le c \end{cases}$$



this all are called fuzzifiers

vi) S-shaped MF

$$\mu_A(x) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & x \ge b \end{cases}$$



25/7/24

Fuzzy set

A = {4, 5, 6} → this is a set

A = {(4, 0.5), (5, 0.2), (6, 0.3)}

the possibility of this fuzzy value or MF

{(A, $\mu_A(x)$)}

$\mu_A(x)$ = {0.5, 0.2, 0.3}

(prev class)

The characteristics of diff. MF

→ It defines the fuzzifier (to make the hand problems soft)

The use of fuzzy set & it's value are used to define the natural language of artificial lang. this type of lang is called linguistic language ← linguistic lang

Linguistic lang
(2 terms)

Linguistic variable (fuzzy ~~set~~ variable) — Linguistic term (fuzzy value)

Eg
i) Suppose the ling. varn. is height → ling. term: 'short', 'tall', 'very short', etc

ii) Speed (ling. varn.) $\xrightarrow{\text{ling. term}}$ 'low', 'fast', 'steady'

IF-THEN rule

The formulation of ling. lang. is based on ling. rules. The rule is called IF THEN rule. It will convert all the rules into uniform.

Eg

R1: If dist is small then speed is low.

R2: If dist is medium then speed is steady

R3: If dist is large then speed is high.

• The soln of a particular problem with a ling. lang. there is a requirement of fuzzy system, fuzzy inference system, fuzzy expert system. (automation)

The block diagram of fuzzy inference system
(How to soln fuzzy prob. with inf sys)

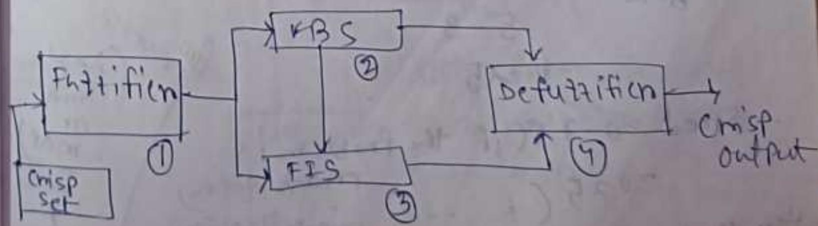
A FIS has 4 components—

i) **Fuzzifier** — This is the MF on m grade

ii) **Knowledge Base** — This may contains the set of rules. (I+P→O (processing))

iii) **Fuzzy inference engine** — It performs the reasoning of fuzzifiers with incorporating the knowledge based rules with the predefined fuzzy set operations. (OR, AND, NOT).

iv) **Defuzzifier** — Convert the outcome of FIS into real value outcome. (Soft → Hard)



Imp type of qn in exam

crisp set – It is a conventional set where an element is either a members of the set or not. They are also known as 'classical' / 'ordinary' sets.

crisp o/p – It means to a definite & precise result.

ii) ⊛ Those rules. (given) what will be my speed if dist. is 4.5 km. (with the help of ...)

Step – i) 1st we have to find ling. varr. & ling. term.

ii) Then the fuzzy set

no. of ling var = no. of fuzzy sets

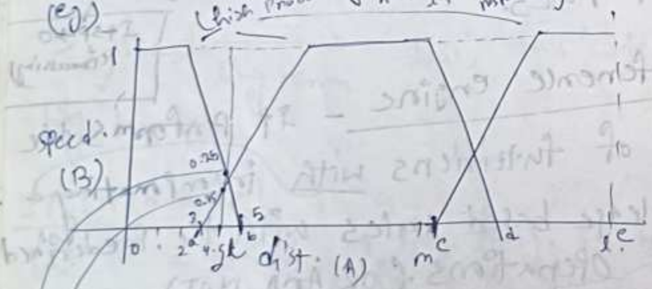
(eg) $A = dist = \{s, m, l\}$

$B = speed = \{l, st, hi\}$

iii) we have to define the membership func using this 2 sets

CMF = membership func

(eg) (this probability of speed l., st., hi. respectively)



$\frac{x-a}{b-a} = \frac{4.5-3}{5-3} = \frac{1.5}{2} = 0.75$

$\frac{b-x}{b-a} = \frac{5-4.5}{5-3}$

$= 0.25$

∴ Speed $= 0.75$ (if the problem is maximization)

$= 0.25$ (" " minimization)

fuzz process (i.e either min or max) ans

Here, speed $= \mu_1 = 0.75$, $\mu_2 = 0.25$ (this is called aggrigation)

Tricks / techniques

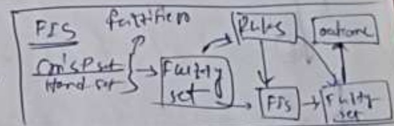
i) The aggrigation of fuzzifier & rules are done in FIS.

ii) There are 2 techniques in FIS –

a) mamdani FIS
b) sugeno FIS
(popular FIS rules)

6/8/24

Defi. mamdani FIS



Step 1: Determine the fuzzy rules.

Step 2: Apply fuzzifier to make the i/p into fuzzy set

Step 3: Finding the rules' strength (RS). → It will build a crisp outcome

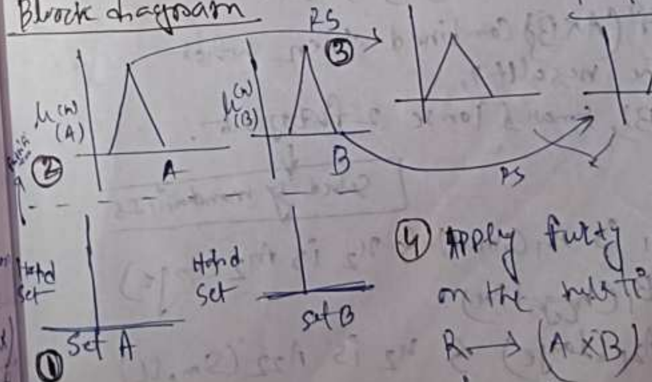
Step 4: Processing of FIS.

Step 5: Combine all the outcomes to form a O/P fuzzy set.

Step 6: Apply defuzzifier

Hard set – some points in the set, some will be not in it

Block diagram



④ Apply fuzzy operations on the relation set R

$R \rightarrow (A \times B)$ (max, min)

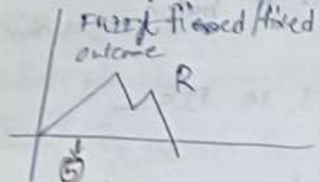
Binary operation

2 set operation – Fuzzy relation set

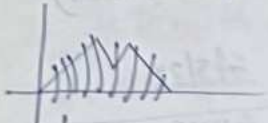
↓

Final result

Fuzzy fixed/fixed outcome



(5) Defuzzification



↓ (6)

$A = \{1,2,3\}$
$B = \{4,5,6\}$
$R = A \times B = \{(\ ),(\ ),(\ )\}$
Binary set
↓
for binary set → fuzzy operation on the relating set

$\boxed{\text{if single then fuzzy operation}}$

Rule: to build this kind of relationship in b/w 2 sets.

Eg (A,B)

1) If A is small then B is large.
If A is large then B is small.

A = {1,3,4}
B = {4,5,6}

Find out if (A×B) combined then what will be the result?
[2 sets (A,B), small & large 2 fuzzy set.
fuzzy)

$\boxed{\text{Solved by mamdani FIS}}$

2) R1: If x_1 is A_{11} (small) & x_2 is A_{12} (large) then, y is B_1 (negative).
R2: If x_1 is A_{21} (large) & x_2 is A_{22} (small) then, y is B_2 (positive).

Find A_{33} of x_1 & A_{23} of x_2.

Small → A_{11}, A_{22}
Large → A_{12}, A_{21}

neg & post → outcome

$\mu_{A_{11}}(5)$ & $\mu_{A_{12}}(6)$
↓
B_1 (-ve)

$\boxed{\text{mamdani FIS}}$

Imp
fuzzification
Def
Relation

Sugeno FIS defi.

In this fuzzy model the format of the rule will be -

If x is A & y is B then $z = f(x,y)$ → It will bring a func

↓ Diff b/w mamdani & sugeno FIS

The FI process under Sugeno are -

i) Fuzzifying the i/ps
ii) Applying the fuzzy operator.

Eg

1) If $7 = x$ & $9 = y$ then, $z = ax + by + c$

one of the fi'sde ← 7 is a fuzzy set
7 (so 7 = x, not x = 7)
[i.e next time it can be 9, 10 etc]

Comparison b/w mamdani & Sugeno FIS

MFIS	SFIS
i) The o/p membership func - Here this is basically the fuzzy outcome MF (a single val. with membership grade)	Here the MF will be either linear or constant

ii) The aggrigation & defuzzification are diff. in both this cases	Here the aggrigation & defuzzifier are diff.
iii) Less mathematical rules exists than Sugeno	more mathematical rules exists than mamdani.

Fuzzy relations (R)

Lets, suppose there are 2 fuzzy sets A & B. Then the mapping of A to B (A → B) is defined as a fuzzy relationship

$\boxed{R: A \to B}$, i.e, R will contain $A \times B$, ∴ $\boxed{R: (A \times B)}$

so, the fuzzy relation set R is defined as, $\boxed{R = \{(x, y), \mu_R(x, y)\}}$

where $\boxed{A = (x, \mu_A(x))}$ $\boxed{B = \{y, \mu_B(y)\}}$

Eg ①

$A = \{(1, 0.5), (2, 0.3), (3, 0.7)\}$

$B = \{(7, 0.5), (9, 0.2), (11, 0.6)\}$

$R = \{((1,7), 0.5), ((1,9), 0.5), ((1,11), 0.5), ((2,7), 0.5), ((2,9), 0.3),$
$((2,11), 0.3), ((3,7), 0.7), ((3,9), 0.7), ((3,11), 0.7)\}$

If $A = \{ -- (13, 0.3)\}$

$R = \{ -- ((13, 9)), 0.3)\}$

$\mu_R(x,y)$ is based on the given rules of combining the 2 sets.

R1: $\boxed{\text{If } (x<y) \text{ then, } \max(\mu_A(x), \mu_B(y))}$

② Establish a binary FR. R with the given rules:

R1: $R(1,1) = R(2,2) = R(3,3) = 1$

R2: $R(1,2) = R(2,1) = R(2,3) = R(3,2) = 0.8$

R3: $R(1,3) = R(3,1) = 0.3$

From the membership function μ_R?

$A = \{1, 2, 3\}$, $B = \{1, 2, 3\}$

$\mu_R = \{\cancel{(1,1)}, \ldots\}$, $\begin{cases} 1 & \text{if } x = y \\ 0.8 & \text{if } |x-y| = 1 \\ 0.3 & \text{if } |x-y| = 2 \end{cases}$

matrix rep. of R on R ← [this R will combine A & B]

$$\mu_R = \begin{array}{c|c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 1 & 0.8 & 0.3 \\ 0.8 & 1 & 0.8 \\ 0.3 & 0.8 & 1 \end{bmatrix} \end{array}$$

$\boxed{R(x,y) = \{(x,y), \mu_R(x,y)\}}$
$\boxed{S(y,z) = \{(y,z), \mu_S(y,z)\}}$

• $R(A, B) \underset{\text{(join)}}{\times} S = (B, C)$ ...
[Let there is one more set $T = (A, C)$]

Lets suppose there are 2 binary fuzzy relations R & S. So, there are 3 major fuzzy operations will be applied on this R & S defined as

i) Union – The union of R & S relationship or max relation

$R \rightarrow x \times y, \quad S \rightarrow y \times z$

i) $\boxed{R \cup S = ((x,z), \mu_{R \cup S})}$

$\boxed{\mu_{R \cup S} = \max\{\mu_R(x,y), \mu_S(y,z)\}}$

(4) ii) Intersection relation or min relation

$\boxed{R \cap S = \{(x,z); \mu_{R \cap S}\}}$

$\boxed{\mu_{R \cap S} = \min(\mu_R(x,y), \mu_S(y,z))}$

(5) iii) Complement

$\boxed{\mu_{R'}(x,y) = 1 - \mu_R(x,y)}$

$\boxed{\mu_{S'}(y,z) = 1 - \mu_S(y,z)}$

$\boxed{\begin{array}{l} R' = \{(x,y), \mu_{R'}(x,y)\} \\ S' = \{(y,z), \mu_{S'}(y,z)\} \end{array}}$

Eg

(1) R = "x is considerably longer than y"

S = "x is very close to y"

$$R = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{bmatrix} 0.8 & 0.1 & 0.1 & 0.7 \\ 0.0 & 0.8 & 0.0 & 0.0 \\ 0.9 & 1.0 & 0.7 & 0.8 \end{bmatrix} \end{array} \qquad S = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{bmatrix} 0.4 & 0.0 & 0.9 & 0.6 \\ 0.9 & 0.4 & 0.5 & 0.7 \\ 0.4 & 0.0 & 0.8 & 0.5 \end{bmatrix} \end{array}$$

perform, $R \cup S = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{bmatrix} 0.8 & 0.1 & 0.9 & 0.7 \\ 0.9 & 0.8 & 0.5 & 0.7 \\ 0.9 & 1.0 & 0.8 & 0.8 \end{bmatrix} \end{array}$

[OR means performance will increase] (no 0.0)

intersection / min = at least 0, but not better than union / max.

$$R \cap S = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_1 & y_2 & y_3 \end{array} \\ \begin{bmatrix} 0.4 & 0.0 & 0.1 & 0.6 \\ 0.0 & 0.4 & 0.0 & 0.0 \\ 0.4 & 0.0 & 0.7 & 0.5 \end{bmatrix} \end{array}$$

Soft computing → Fuzzy set theory + genetic algos

$$\bar{R} = 1 - R = \begin{bmatrix} 0.2 & 0.9 & 0.9 & 0.3 \\ 1.0 & 0.2 & 1.0 & 1.0 \\ 0.1 & 0.0 & 0.3 & 0.2 \end{bmatrix}$$

$$\bar{S} = 1 - S = \begin{bmatrix} 0.6 & 1.0 & 0.1 & 0.4 \\ 0.1 & 0.6 & 0.5 & 0.3 \\ 0.6 & 1.0 & 0.2 & 0.5 \end{bmatrix}$$

Projection & composition (next day)

Linguistic variable – It is also called fuzzy variable.

It's a variable that uses words / sentences from a language as its values. It can be defined using linguistic terms like low, avg, high instead of numerical vals. (Used in robot interactions, computer simulations etc). Eg – Age = {child, young, old} etc.
↓ L.V. ↓ L.T.

Linguistic terms – It is also known as fuzzy value.

It is a word / group of words that designates something, especially in a particular field.

Eg – Age = {child, young, old} etc
↓ L.V. ↓ L. terms.

[Fuzzy relation set - A fuzzy relation is an extension of an ordinary relation. It allows expressions involving ambiguity such as 'x & y are almost the same' or 'z is much bigger than w'. Let X & Y be 2 sets of interest] 11

8/8/24

⑥ Self relation

R, R or S, S

⑦ Projection on R

$R = \{(x,y), \mu_R(x,y)\}$

$A = \{x, \mu_A(x)\}$

$B = \{y, \mu_B(y)\}$

R	y_1	y_2	y_3	y_4
x_1	0.5	0.3	0.1	0.6
x_2	0.9	0.2	0.7	1
x_3	0.11	0.4	0.18	0.72

Eg - Projection of x on R, that means $\boxed{\Pi_R(x)}$

$$\boxed{\Pi_R(x) = \max\{(x,y) \in R, \forall y\}}$$

eg

$\Rightarrow \Pi_R(x_1) = 0.6$ ← max value

o.p
(x_1,y_1) (x_1,y_2)
(x_1,y_3) (x_1,y_4)

[op = ordered pair]

→ [x fix, measure on y]

$\Rightarrow \Pi_R(x_2) = 1$

$\Rightarrow \Pi_R(x_3) = 0.72$

~~(Projection of y on R~~
~~$\Pi_R(y)$)~~

⑧ Projection of y on R

$$\boxed{\Pi_R(y) = \max\{(x,y) \in R, \forall x\}} \rightarrow [\text{y fixed, measure on } x]$$

$\Pi_R(y_1) = 0.9$

$\Pi_R(y_2) = 0.7$

$\Pi_R(y_3) = 0.7$

$\Pi_R(y_4) = 1$

[this is only for R, if 2 relations then have to find out Projection on that composite relation]

⑨ Composition of R & S

R → A×B   x×y
S → B×C   y×z
are binary

Let suppose R & S are 2 fuzzy relations, then the composition of R & S is denoted as - $\boxed{R \circ S} = m$

not [m = It is also a relation]

$m = R \circ S = \{(x,z) ; (x,y) \in R_{x \times y}$ & $(y,z) \in S_{y \times z}\}$, $\mu_{R \circ S}(x,z)$

Such that

$$\begin{bmatrix} \overset{R}{(x,y)} \quad \overset{S}{(y,z)} \\ (x,z) \end{bmatrix}$$

$$\boxed{\mu_{R \circ S}(x,z) = \max\{\min(\mu_R(x,y), \mu_S(y,z))\}}$$

fuzzy membership val. of this

max - min principle

[St (on above) max-min same thing (rev)]

Eg (1)

Let R is given by

R	verdant	Half mature	mature
Green	1	0.2	0
yellow	0.3	1	0.4
Red	0	0.2	1

x (rows), m. value y (columns)

(G, V) = 1
(y, H m) = 0.2
these are o.p, this o.p is stored in R, R is the matrix rep. of the relation R. same for S.

Let S is given by

S	Sour	testless	Sweet
verdant	1	0.2	0
Half mature	0.7	1	0.5
mature	0	0.7	1

y (rows), z (columns)

fuzzy relation T

R o S	Sour	testless	Sweet
Green	1	0.2	✓
yellow	0	✓	✓
Red	✓	✓	✓

f.r.

R.S	S.	T.	Sw
G.	1	-	-
Y.			
R.			

$\llcorner$ 0.14, 0 ↓ max

(1 0.2 0) ... 0, 0.2, 1

$(1, 0.2, 0) \xrightarrow{\min} \begin{pmatrix} 1 \\ 0.7 \\ 0 \end{pmatrix}$

↓ max

1

→ Composite relation b/w 2 fuzzy relation R & S.

ASS (H/W)

(3) Eg (2)

Consider 2 relations R & S, given that

R =

R	y_1	y_2	y_3	y_4
x_1	0.73	0.6	0.8	0.11
x_2	0.71	0.8	0.7	0.9
x_3	0.1	0.11	0.31	0.7

S =

S	z_1	z_2	z_3
y_1	.52	0.3	0.12
y_2	.81	.91	0.3
y_3	.71	.82	.72
y_4	.81	.71	.34

(1) R o S ? (max-min operation)

(2) min-max operation (rev)

(3) ~~max~~ . max - . (Product) [R o S ?]

(4) max-avg (~~Avg~~ Avg → max

$\left(\frac{1+1}{2}, \frac{0.2+0.7}{2}, \frac{0+0}{2}\right)$

max avg	S.	T.	Sw
G	1	-	-
Y	-	-	-
R	-	-	-

(1)

R o S	z_1	z_2	z_3	z_4
x_1	0.71	0.8	0.72	
x_2	0.81	0.8	0.7	
x_3	0.7	0.7	0.34	

(max-min)

(0.52, 0.6, 0.71, 0.11)
(0.3, 0.6, 0.8, 0.11)
(0.12, 0.3, 0.72, 0.11)
(0.52, 0.8, 0.7, 0.81)
(0.31, 0.8, 0.7, 0.71)
(0.12, 0.3, 0.7, 0.34)
(0.1, 0.11, 0.31, 0.7)
(0.1, 0.11, 0.31, 0.7)
(0.1, 0.11, 0.31, 0.34)

(2)

R o S	z_1	z_2	z_3
x_1	0.73	0.71	0.34
x_2	0.71	0.71	0.71
x_3	0.52	0.31	0.12

(min-max)

(0.73, 0.81, 0.8, 0.81)
(0.73, 0.91, 0.82, 0.71)
(0.73, 0.6, 0.8, 0.34)
(0.71, 0.81, 0.71, 0.9)
(0.71, 0.91, 0.82, 0.9)
(0.71, 0.8, 0.72, 0.9)
(0.52, 0.81, 0.71, 0.81)
(0.31, 0.91, 0.82, 0.71)
(0.12, 0.3, 0.72, 0.7)

(3)

..	z_1	z_2	z_3
x_1	0.6	0.7	0.6
x_2	0.7	0.7	0.5
x_3	0.6	0.5	0.238

(0.38, 0.5, 0.6, 0.1)
(0.2, 0.5, 0.7, 0.1)
(0.1, 0.3, 0.6, 0.03)
(0.4, 0.6, 0.5, 0.7)
(0.2, 0.7, 0.6, 0.6)
(0.1, 0.2, 0.5, 0.3)
(0.012, 0.033, 0.2, 0.238)
(0.52, 0.1, 0.2, 0.6)
(0.03, 0.1, 0.3, 0.5)

(0.6, 0.7, 0.8, 0.5)
(0.5, 0.8, 0.81, 0.4)
(0.4, 0.5, 0.8, 0.2)
(0.6, 0.81, 0.7, 0.7)
(0.5, 0.81, 0.8, 0.81)

(5) -~	z_1	z_2	z_3
x_1	0.8	0.81	0.8
x_2	0.9	0.81	0.8
x_3	0.8	0.7	0.52

(0.4, 0.6, 0.8, 0.6)
(0.3, 0.5, 0.5, 0.8)
(0.2, 0.5, 0.6, 0.7)
(0.1, 0.27, 0.51, 0.52)

General formulation (Generalized eqn for this 4)

The M.F of max-min, min-max, max-product, max-av
(o) (o) (-.) (-~)
is based on max star composition.

$\boxed{R \star S = \{(x,z) : \mu_{R \star S}(x,z)\}}$ → $\boxed{R \star S = \{(x,z) : (x,y) \in R_{x,y}, (y,z) \in R_{y,z}, \mu_{R \star S}\}}$

$\boxed{\mu_{R \star S}(x,z) = \max [\mu_R(x,y) \star \mu_S(y,z)]}$

This all are binary fuzzy relationship (R, S, R★S)

• This relation will be used in FIS.

Fire – If all constraints of a rule are satisfied the rule is said to fire.

FIS – A FIS is a method that interprets the values in the I/P vector &, based on some sets of rules, assigns vals to the O/P vector.

Defuzzification Imp

It is the process to convert fuzzy into crisp value.

methods

1. max-membership principle
2. centroid method
3. weighted Avg method
4. mean-max principle
5. centre of sum
6. First or last maxima

Among this methods the centroid method & the weighted avg method are commonly used, due to it's computational nature & the derived outcomes are feasible.

Centroid method – Also known as centre of area or center of gravity (COA / COG)

$$\boxed{z \text{ or } COG = \frac{\sum_{z_i \in Z} \mu_c(z_i) \times z_i}{\sum_{z_i \in Z} \mu_c(z_i)}}$$

Eg – $C = \{z, \mu_c(z)\}$

$C = \{(4, 0.5), (6, 0.7), (3, 0.7)\}$

$$COG = \frac{(0.5 \times 4) + (0.7 \times 6) + (0.7 \times 3)}{0.5 + 0.7 + 0.7}$$

[Here here if there will be more than set, then that result will be COG 2]

• For 1 set → 1 COG, for 2 sets → 2 COGs

Pos	S	T	W
			0.1
G	1	0.2	
y	0.5	0.C	0.3
R	0.1	0.7	1

2 sets R & S, so there should be 2 COG

$Pos = [\{(G, S), 1\}, \{(G, T), 0.2\} \ldots]$

$= [\{(G*1), S*1\}, \{(G*0.2), (T*0.2)\}, \{(G*0.1) (SW*0.1)\} \ldots$

(2, 3), 0.5 m. value

(2*0.5, 3*0.5)

COG1 COG2

G*1 = COG1, S*1 = COG2

G*0.2 = COG1, T*0.2 = COG2

G*0.1 = COG1, SW*0.1 = COG2 & so on.

$COG_R = \dfrac{G*1 + G*0.2 + \ldots}{1 + 0.2 + 0.1 \ldots}$

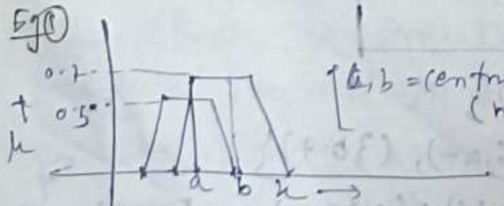
$COG_S = \dfrac{S*1 + S*0.5 + \ldots}{1 + 0.5 + \ldots}$

weighted Avg method

$$\boxed{WAV = \frac{\sum \mu_C(\bar{z}) \cdot \bar{z}}{\sum \mu_C(\bar{z})}}$$

$\bar{z}$ = centroid of each m.F.

Eg(1)



[a, b = centroid value (mean value)]

$$\boxed{WAV = \frac{0.5*a + 0.7*b}{0.5 + 0.7}}$$

(2) $A = \{(1, 0.3), (2, 0.5), (3, 0.7)\}$

$B = \{(2, 0.7), (5, 0.8), (9, 0.7)\}$

~~WAV~~ Centroid of $A = \dfrac{1+2+3}{3} = 2$

$B = \dfrac{2+5+9}{3} = 5.33$

$WAV = \dfrac{2*0.5 + 5.33*0.8}{0.5+0.8}$ nearest val [(clst m.v.) / m.v]

• If C. of. B = 6, then not in set, then Avg (5, 9) & take the value. [i.e $\dfrac{0.8+0.7}{2} = 0.75$ i.e 0.75*6]

Lab assignment

(6) Consider the 2 relations R & S where

a) $R_{5 \times 6}$, $S_{6 \times 3}$. Establish –

i) max-min

ii) min-max

iii) max-product

iv) max-Avg

on the composite relationship of R & S. Considering the vals of R & S randomly b/w 0 & 1.

b) Plot the graph of the derived 4 principles.

(7) Let A & B are 2 fuzzy sets which have 20 elements with 20 membership values/grades.

i) Find centers of gravity (COG)

ii) WAV

C. ass.

22/8/24

Clw

Genetic algorithm (GA)

i) This algo is a part of soft computing to solve problems which need optimization. The process of solⁿ of genetic algo is a subclass of ~~two~~ evolutionary computing.

The basic fundamental design for GA base problem solⁿ is based on the elements of human body cells. The important elem. of a (human) cell is chromosomes. The genetic info is stored in chromosomes.

ii) Each chromosome is build of ~~a~~ DNA.

iii) There are 23 pairs of chromosomes on each cells.

→ 1 pair of chromosome

iv) Each chromo. is divided into some genes'.

→ unique pos / Gene

v) The possibilities of genes' form 1 property is called allele.

vi) Every gene has unique position on the chromo, that is called locus.

viii) Hence the ~~summerization~~ summarization of this points from the computational point of views is as follows –

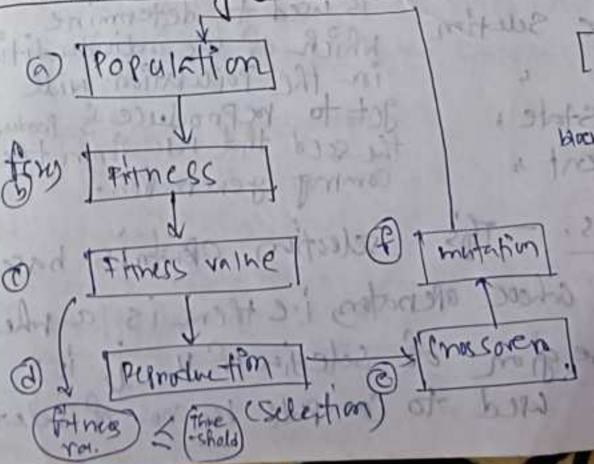
Natural A.	G. A (from computer pov.)
Chromosome →	String
Gene →	Feature or character
Allele →	Feature value
Locus →	String position
Genotype →	the general struct of chromosome
Phenotype →	the encoding of genotype info is called phenotype

01001011
↓ ↓ ↓ conversion
A B C
→ Phenotype

X → [1 . . . 1]
↑
String

- The problems that needs optimisation can be solved by G.A. GA can solve any type of optimization solⁿ.

The block dig. for the solⁿ of an optimization problem using G.A is ~~that~~



[Each of this ~~diff~~ block needs small small algos]

Population

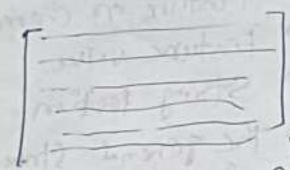
Selection from whole population - Random selection

i) Population means collection of chromosomes.

Each person is samples i.e population many

Sample $\xrightarrow{\text{means}}$ data $\xrightarrow{\text{means}}$

Strings $\xrightarrow{\text{means}}$ chromosomes



Population

A population is the entire group that we want to draw conclusions about. A sample is the specific group that we will collect data from.

Fitness

• = imp point

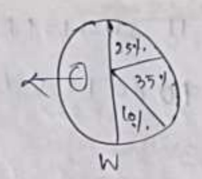
• Each steps of GA operations ~~in~~ need a computational step for the soln. For eg, there are some technique / solns for selection, crossover & mutation.

• The rep. of each chromosome is called encoding & the initial ~~ending~~ rep. is binary ~~and~~ encoding. (01.0111 → gene encoding)

Selection techniques → It is a process that is used to determine which of the individualities in the population will get to reproduce & produce the seed that will form the coming generation.

① Roulette selection
② Rank "
③ Steady state "
④ Turnament "

1) Roulette s. - This selection operator is based on the wheel operator, i.e there is a wheel W, the region of selection of W is it's center, is used to compute the no. of selected chromo



The probability of getting selected / to be presented is rep by this wheel.

2) Rank selection - Here the chromo. are selected based on their ranks which are computed by the fitness value.

3) Steady selection - this selection of chromo. is based on steady state comparisons of fitness value corresponds to each chromo.

4) Turnaments :- Here the chromo. are selected based on groups of ~~wheels~~ win / lose of fitness value of the chromo.

Crossover

This operation have ~~4~~ 3 process -

i) one-point crossover

Chromo A : 1010|1010 → A' : 1010 1011

Chromo B : 1110|1011 → B' : 1110 1010

Based on 1 point cross over happens.

ii) two point crossovers

A : 10|10|1010 → A'' : 10101010

B : 11|10|1011 → B'' : 11101011

Based on 2 points cross over happens.

iii) multipoint crossovers

change – | – fixed – | – change – | – fixed ... like that ← this is for 3 points (more)

A : 11|00|11|01   A''' : 11 00 11 11

B : 10|00|11|11 ⟶ B''' : 10 00 11 01

fixed change fixed change

ii) Uniform crossover

1|0|1|0|1 ⟶ 1 1 1 0 1

1|1|1|0|1   1 0 1 0 1

fixed change f. c. f.

f c f c f c ⟶ can do randomly P.T.O.

For each points cross over will happen.

Mutation operation

This operation is based on flipping of genetic infor for particular chromosomes.

For eg, Chromo → 1001010

0110101 } complement

Eg of GA sol. using genotype

Eg Sol (Exam)

1) Solve the given maximization func, $f(x) = x^2$ when x ranges from 0 to 31. [$x \to [0,31]$], by using the GA with applying-

i) Encoding technique (Binary) (5 length chromo- binary rep - 5 bits)

ii) Selection operation (Roulette wheel)

iii) Crossover (one-point). ~~Find the value of x~~ Find the value of x (best val) for which fun is max?

Suppose given take min initial values (as chromo. (population)

Pair A) $\xrightarrow[\text{take}]{\text{then}}$ 4

[can also take - 8, 10 ....., but large will take time, so preferred - 4, 5 .....]

Binary rep.

(min 8 bits → $\frac{\text{for } 25 (eg)}{00011001}$)
(needed)

(in exam 5/4 bits - 11001)

Let given initial values (x = 13, 24, 8, 19)

Each x is chromo.

Step 1: Convert each initial value of x into binary rep. of length 5 bits. (Binary encoding)

Step 2 Compute table (Selection)

String No.	Initial population	x	$f(x)$	Probability count $f(x)/\sum f(x)$	Expected count	Actual count
1	01101	13	169	0.14	0.58	1
2	11000	24	576	0.49	1.97	2
3	01000	8	64	0.06	0.22	0
4	10011	19	361	0.31	1.23	1
			$\sum f(x)$ = sum 1170			

4 initial population

(how many times will it come)

Expected count $= \frac{f(x)}{\text{avg } f(x)} = \frac{f(x)}{\left(\frac{\sum f(x)}{N}\right)} = \frac{f(x)}{\bar{f}(x)}$

Avg $= \frac{\sum f(x)}{N} = \frac{1170}{4} \approx 293$

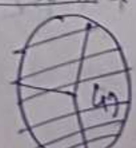
Probability count $= \frac{f(x)}{\sum f(x)}$ → (prob. of being present / getting selected)

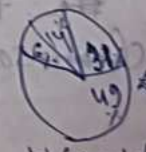
} for roulette wheel

Actual count = nearest value of the fraction ($0.58 \approx 1$, $1.97 \approx 2$)

(round off)

(round off of expected count rep. sum used)





From probability count rep the prob. of getting selected

Strm = Strength

W (roulette wheel)

(Crossover)

Step 3

Apply the cross over. 2 chromos when are crossed, the new chromo is called off springs. so compute off springs from the cross over.

$$\left[\begin{array}{l} 1 \rightarrow 1 \text{ times the chromo will come} \\ 2 \rightarrow 2 \text{ ---} \\ 0 \rightarrow 0 \text{ --- (not come)} \\ 1 \rightarrow 1 \text{ ---} \end{array}\right] \text{ selection process}$$

S1: 01101 → (1 time comes)

S2: 11000 → (1 time comes) ⎤
S3: 11000 → (1 time comes) ⎦ → (2 times come)

S4: 10011 (1 times comes)

For eg — given – one-point crossover → corners split

0 1 1 0 | 1	off S1:	01100	→ S_1'
1 1 0 0 | 0	off S2:	11001	→ S_2'
1 1 | 0 0 0	off S3:	11011	→ S_3'
1 0 | 0 1 1	off S3:	10000	→ S_4'

middle split

(For exam split pos will be given like- from 1st pos 3rd pos, last pos, middle pos like that) ← Crossover point

Step 4: Best selection (finding the best val of x)

Compute the table for the best selection of the chromosomes.

We'll again compute the same table with f(x) & this step is a final & mandatory for all qns.

off-springs	population	x	$f(x)$	P.C.	E.C	A.C
1. S_1'	01100	12	144	0.08	0.33	0
2. S_2'	11001	25	625	0.35	1.43	1
3. S_3'	11011	27	729	0.42	1.67	2
4. S_4'	1000	16	256	0.15	0.58	1

(no significancy)

$\Sigma f(x) = 1754$

$\frac{\Sigma f(x)}{N} = 438.5$

$x = 27$, for which the value of f(x) is max. As in prev table 24 was best, but in 4th step we can find best, ~~that~~ it that is 27. (for 27, $x^2 = 729$, for 24, x^2 (f(x)) = 576). So, the ans is 27. (If we stop at step 3, the ans will be 24, which is wrong, So we must keep it in mind that step 4 gives the actual ans / best selection val.)

Assignment (h/w)

(4) Solve the above given problem for (i) $f(x) = x^2+2x$
(ii) $f(x) = x^3$.
Initial values $[x = 13, 24, 8, 19]$ (same as prev)

(5) Solve ass 1 additionally mutation after crossover. (mutation = bit flipping) ~~same as~~

(4) i) $f(x) = x^2 + 2x$

$x = 13, 24, 8, 19$
(1) (2) (3) (4)

Step 1 (encoding technique (binary) – 5 length chromo = 5 bits)

Convert each initial val's of x into binary rep. of length 5 bits.

Step 2. (Selection operation – Roulette wheel)

Compute table

Str. no.	Initial population	x	$f(x)$	P.C	E.C	A.C
1	01101	13	195	0.21	0.85	1
2	11000	24	624	0.68	2.72	3
3	01000	8	80	0.09	0.35	0
4	10011	19	399	0.43	1.74	2

summation $\sum f(x) = 918$

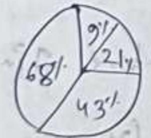
Avg $= \frac{\sum f(x)}{N} = \frac{918}{4} = 229.5$

$P.C = \frac{f(x)}{\sum f(x)}$, $E.C = \frac{f(x)}{\left(\frac{\sum f(x)}{N}\right)}$, A.C = roundoff(E.C)

If decimal val of a chromo is greater → chance of getting selected is greater

But it doesn't implies that, that chromo is the best selection / best val.

It may or may not be



← w (Roulette wheel)

Step 3 : (Crossover = one-point)

Apply crossover. When 2 chromos are crossed over the newly generated chromo is called off-spring. So compute off-springs from the cross over:

S1: 01101 → 1 time (comes)
S2: 11000 → 3 times
S3: 01000 → 0 times (doesn't come)
S4: 10011 → 2 times

} Selection Process (can skip this too)

~~Selection Process~~

given, one-point crossover

the split pos is not given here (but will be given in the exam). So let's do as prev.

0 1 1 0 | 1 → off S1 : 01100 → S1'
1 1 0 0 | 0 → off S2 : 11001 → S2'
1 1 0 | 0 0 → off S3 : 11000 → S3'
1 1 0 | 0 0 → off S4 : 11000 → S4'
1 0 | 0 1 1 → off S5 : 10011 → S5'
1 0 | 0 1 1 → off S6 : 10011 → S6'

S1 : 01101 (1 time)
S2 : 11000
S3 : 11000 (3 times)
S4 : 11000
S5 : 10011
S6 : 10011 (2 times)

Step 4 (Best Selection – finding the best val of x)

off spr	Population	x	$f(x)$	P.C	E.C	A.C
1. S1'	01100	12	168	~~3.44~~ 0.05	0.29	0
2. S2'	11001	25	675	0.19	1.15	1 (Actually comes more than 1)
3. S3'	11000	24	624	0.18	1.07	1
4. S4'	11000	24	624	0.18	1.07	1
5. S5'	10011	19	399	0.11	0.68	1
6. S6'	10011	19	399	0.11	0.68	1

is best val.

Prob. of getting selected == comes more == selecting / best val.

$\sum f(x) = 3514$

$\frac{\sum f(x)}{N} = 585.67$

∴ the best val. = 25 for which $f(x)$ is max.

(ii) $f(x) = x^3$

$x = 13, 24, 8, 19$
(1) (2) (3) (4)

Step 1 (encoding technique (binary) – 5 length chromo = 5 bits)

Convert each initial vals of x into binary rep of length 5 bits.

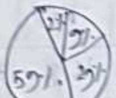
Step 2 (selection function – Roulette wheel)

Compute table.

Str no	Initial Population	x	$f(x)$	P.C	E.C	A.C
1	01101	13	2197	0.09	0.38	0
2	11000	24	13824	0.59	2.36	2
3	01000	8	512	0.02	0.09	0
4	10011	19	6859	0.29	1.17	1

$\Sigma f(x) = 23392$

$\frac{\Sigma f(x)}{N} = 5848$



← Roulette wheel

$P.C = \frac{f(x)}{\Sigma f(x)}$, $E.C = \frac{f(x)}{Avg(f(x))}$ or $\frac{f(x)}{\left(\frac{\Sigma f(x)}{N}\right)}$, A.C = round off (E.C)

Step 3 (Crossover = one-point)

Apply crossover when 2 chromos are crossed over the newly generated chromo is called off-spring. So compute off-springs from the cross over.

S1: 01101 → 0 time (doesn't come)
S2: 11000 → 2 times
S3: 01000 → 0 time (doesn't come)
S4: 10011 → 1 time
} selection process

S1: 11000 (2 times)
S2: 11000
S3: 10011 (1 time)

given one-point crossover

the split is not given here (but will be given in the exam). So let's do as prev (take any random 1-1(2) i.e. pair & continue the same)

11|000 → off S_1: 11011 → S_1'
10|011 → off S_2: 10000 → S_2'

Step 4 (Best selection – Finding the best val. of x)

off str	Population	x	$f(x)$	P.C	E.C	A.C
1. S_1'	11011	27	19683	0.83	1.7	2
2. S_2'	10000	16	4096	0.11	0.34	0

$\Sigma f(x) = 23779$

$\frac{\Sigma f(x)}{N} = 11889.5$

∴ the best val. = 27 for which $f(x)$ is max.

(5) (easy)

after step 3, step 4 (mutation = Bit flipping)

[Flip choose any random str, or more than 1 random str or all strs & do 0→1, 1→0 randomly, i.e. not for all bits in a str, do randomly, e.g. if you choose 01100 then 11101 like that]

	mutated str
S_1': 01100 →	S_1'': 11101
S_2': 11001 →	S_2'': 11001
S_3': 11000 →	S_3'': 11010
S_4': 11000 →	S_4'': 11100
S_5': 10011 →	S_5'': 10001
S_6': 10011 →	S_6'': 10000

[Preferred = If any str has the highest val, then don't change it (no bit flip), but it isn't restricted, we can flip] [Bit flipping → we can do 1 or more than 1 bit flipping]

Step 5 (Best selection – Finding best val of x)

new off str	Population	x	$f(x)$	P.C	E.C	A.C
S_1''	11101	29		877	0.24	1.5	2
S_2''	11001	25		675	0.18	1.1	1
S_3''	11010	26		728	0.19	1.2	1

$\Sigma f(x) = 3703$

$\frac{\Sigma f(x)}{N} = 617.2$

s_4'' = 11100 → 28			840	0.23	1.4	1
s_5'' = 10001 → 17			323	0.09	0.52	1
s_6'' = 10000 → 16			288	0.08	0.46	0

∴ the best val. = 29 (not 24 or 25), for which f(x) is max

ii) after step 3, step 4

mutated str

s_1' : 11011 → s_1'' : 11011

s_2' : 10000 → s_2'' : 11111

step 5

new off spr	Population	x	f(x)	P.C	F.C	A.C
1. s_1''	11011	27	19683	0.39	0.8	1 (tensed to 1)
2. s_2''	11111	31	29791	0.60	1.2	1 (actually more than 1)

$f(x) = x^3$

$\Sigma f(x) = 494$

$\frac{\Sigma f(x)}{N} = 247$

∴ the best val. = 31 (not 24 or 27), for which f(x) is max.

Fitness – It is a metric that evaluates how well a solution meets the goals of a problem. It's also known as an evaluation function.

Fitness value – Fitness shows to what extent a genotype is favored by natural selection. Fitness values are b/w 0 & 1. The fittest individual has a fitness of 1, & the fitness of the other members of the population can be expressed as 1 − s, where s is the selection coefficient. $\boxed{\text{Fitness val.}} \leq \boxed{\text{Threshold}}$

Reproduction – Reproducibility is the ability to replicate an experiment & obtain the same results by using the same methodology.

Crossover – It is a genetic operator used to vary the programming of a chromo. or chromos from 1 generation to the next.

mutation – In simple terms, mutation may be defined as a small random tweak in the chromosome, to get a new soln. [It has been observed that mutation is essential to the convergence of the GA while crossover is not.]

One Point Crossover

In this one-point crossover, a random crossover point is selected & the tails of ~~this~~ its two parents are swapped to get new off-springs.

0	1	2	3	4	5	6	7	8	9
5	8	9	4	2	3	5	7	5	8

(fixed) (change)

→

0	1	2	3	4	3	5	7	5	8
5	8	9	4	2	5	6	7	8	9

multi point crossover (two-point it under it)

[It is a generalization of the one-point crossover wherein alternating segments are swapped to get new off-springs.]

0	1	2	3	4	5	6	7	8	9
5	8	9	4	2	3	5	7	5	8

(fixed) (change) (fixed)

→

0	1	2	4	2	3	6	7	8	9
5	8	9	3	4	5	5	7	5	8

Uniform crossover – In a uniform crossover we don't divide the chromo. into segments, rather we treat each gene separately. In this

0	1	2	3	4
5	6	7	8	9

→

5	1	7	8	4
0	6	2	3	9

randomly select

(not total complement of I/P)

Selection — Crossover — mutation

$S_1 \xrightarrow{\text{wings}} S_1' \xrightarrow{\text{wings}} S_1''$

$S_2 \longrightarrow S_2' \longrightarrow S_2'' \xrightarrow{\text{wings}}$ final result (best val)

throw-swing / wings: $S_3 \longrightarrow S_3' \longrightarrow S_3''$

$S_n \longrightarrow S_n' \longrightarrow S_n''$

(2)

27/8/24

Q. (Imp. exam)

GA algo by using chromo with binary rep.

Genotype (010011101---) → Phenotyping (015678 or abcde)

eg of GA Soln using phenotype

Suppose a GA uses chromo of the form $x = abcdefgh$ with a fixed length of 8 genes. Each gene can be any digit b/w 0 to 9 i.e. {0, 1, ..., 9}.

Let the fitness func of the given func (statement)

$f(x) = (a+b) - (c+d) + (e+f) - (g+h)$ ← will be given in exam

Solve a GA Problem to maximize this func with the given initial vals of chromos:

$x_1 = 65413532$ (a b c d e f g h)

$x_2 = 87126601$

$x_3 = 23921285$

$x_4 = 41852094$

Following operators — ← will be given in exam

i) evaluate the fitness of each individual.

ii) Crossover using one point at the middle b/w the 2 higher rank values. $[x_1, x_2]$

iii) Cross the 2nd & 3rd rank val at the pos of b & f. $[x_1, x_3]$

iv) Cross the 1st & 3rd rank uniform.

→ maximize the fun ($f(x)$) $[x_2, x_3]$

(In this case f = fixed, c = change)

Ans (i)

(fit-ness val) $f(x) = (a+b) - (c+d) + (e+f) - (g+h)$

$f(x_1) = (6+5) - (4+1) + (3+5) - (3+2) = 9$ (2nd rank)

$f(x_2) = (8+7) - (1+2) + (6+6) - (0+1) = 23$ (1st rank)

$f(x_3) = (2+3) - (9+2) + (1+2) - (8+5) = -16$ (3rd rank)

$f(x_4) = (4+1) - (8+5) + (2+0) - (9+4) = -19$ (4th rank)

(ii)

$x_1 = 65|4135|32$ (a b c d e f g h)

$x_3 = 23|9212|85$

↓ off-spr

x_3: 65921232

x_4: 23413585

(iv)

$x_2 = 8|7|1|2|6|6|0|1$ (f c f c f c f c)

$x_3 = 2|3|9|2|1|2|8|5$

↓ off-spr

x_5: 83126205

x_6: 27921681

(ii)

$x_2 = 8712|6601$

$x_1 = 6541|3532$

↓ offspring

x_1: 87123532

x_2: 65416601

$f(x_1) = (8+7) - (1+2) + (3+5) - (3+2) = 15$

$f(x_2) = (6+5) - (4+1) + (6+6) - (0+1) = 17$

(we have to evaluate the fitness of offsprings also)

$f(x_3) = (6+5) - (9+2) + (1+2) - (3+2) = -2$

$f(x_4) = (2+3) - (4+1) + (3+5) - (8+5) = -5$

$f(x_5) = (8+3) - (1+2) + (6+2) - (0+5) = 11$

$f(x_6) = (2+7) - (9+2) + (1+6) - (8+1) = -4$

• If the above problem is maximization then the initial problem val. = 23 (as in 2nd case the val. doesn't increase)

But if there is a optimization, soln val. will be either 15 or 17, because this 15/17 are more nearer than 9 & 23 (The diff. b/w 9 & 23 is greater than diff. b/w 15 & 17)

∴ optimum value (fitted val) = 15 & 17

Assign (h/w)

(6) find the optimal solⁿ of the given prob

$f(x) = max[(a+b) - (c+d) + (e+f) - (g+h)]$

where the rep. of x will be 8 dig. & the digits only be from either 0 or 9.

$x: 99999999$
$x: 99999900$
------- like that

there are 4 initial vals.
x_1: 99 00 99 00
x_2: 99 000990
x_3: 99 099990
x_4: 99999990

use (i),(ii),(iii),(iv) GA, of the previous problem. (operations)

~~Then~~

- Naturally travelling salesman problem (TSP) is NP-complete problem & it also lies in combinational problem. So the optimal solⁿ can be found out using GA. For the TSP problem when using GA, genes rep. links b/w the cities. So, how many genes will be used in a chromo. for each individual of the no. of cities is 10. the ans. is 10 genes, when each gene rep. the path b/w 2 cities
- If the TSP problem is solved using phenotype GA then how many genes will be in alphabet of the algo.

Ans. Assume ~~the~~ the alphabet will consist of 45 genes, as there are 10 cities & will be connected with 9 pairs. So the formula for 'n' cities,

$$\boxed{\frac{n \cdot (n-1)}{2}}$$

$$= \frac{10 \cdot (10-1)}{2}$$

$= 45$

Genotype
no. of cities = n
∴ no. of genes needed for each individual chromo. = $\boxed{n}$

phenotype
no. of cities = n
∴ no. of genes needed in alphabet (chromo.) = $\boxed{\frac{n(n-1)}{2}}$

Assign (h/w)

(7) Write the short note on rank based steady state & tournament selection operation ~~tion~~ two (selection operator of GA)

(ascending)

		Rank
x_2	→ 23	1
x_1	→ 9	2
x_3	→ -16	3
x_4	→ -19	4

Assign (6)

Ans.

$f(x) = max[(a+b) - (c+d) + (e+f) - (g+h)]$

~~x~~ Rep. of x will b 8 dig. & digs only be from either 0 or 9.

$\therefore \underset{\text{(chromo)}}{x} = \underbrace{abcdefgh}_{8 = \text{length}} \in \{0 \text{ or } 9\}$ (a,b,c,d,e,f,g, h → genes)

egs- (given)
$x: 99999999$
$x: 99999900$
---- like that

given 4 initial values.

x_1, x_2, x_3, x_4 → chromo.

	a	b	c	d	e	f	g	h
x_1:	9	9	0	0	9	9	0	0
x_2:	9	9	0	0	0	9	9	0
x_3:	9	9	0	9	9	9	9	0
x_4:	9	9	9	9	9	9	9	0

Following operators (questions) (given) (in exam no need to write this part)

i) Evaluate the fitness of each individual.

ii) Crossover using one point at the middle b/w 2 highest rank values. $[x_1, x_2]$

iii) Crossover 2nd & 3rd rank val. at the position of b & f $[x_2, x_3]$ (i.e. after pos. b, f)

iv) Crossover the 1st & 3rd rank uniform. $[x_1, x_3]$

We have to find out the optimal soln of the given problem/func. $f(x)$:

(i) $f(x) = \max[(a+b) - (c+d) + (e+f) - (g+h)]$

$f(x_1) = \max[(9+9) - (0+0) + (9+9) - (0+0)] = 36$

$f(x_2) = \max[(9+9) - (0+0) + (9+9) - (9+0)] = 27$

$f(x_3) = \max[(9+9) - (0+9) + (9+9) - (9+0)] = 18$

$f(x_4) = \max[(9+9) - (9+9) + (9+9) - (9+0)] = 9$

		Rank
x_1	$\rightarrow 36 \rightarrow$	1
x_2	$\rightarrow 27 \rightarrow$	2
x_3	$\rightarrow 18 \rightarrow$	3
x_4	$\rightarrow 9 \rightarrow$	4

(ii)
a b c d e f g h

x_1: 9900 | 9900
x_2: 9900 | 9990

$\xrightarrow{\text{off-spring}}$

a b c d e f g h

x_1': 9900 9990
x_2': 9900 9900

[I'm using x_1', x_2' ...]

(iii)
x_2: 99 | 0099 9 | 90
x_3: 99 | 099 9 | 90

$\xrightarrow{\text{off-spr}}$

x_3': 990999 90
x_4': 990099 90

[In exam don't demonstrate this much (1st part just do it)]

(iv)
a b c d e f g h

x_1: 9 | 9 | 0 | 0 | 9 | 9 | 0 | 0
x_3: 9 | 9 | 0 | 9 | 9 | 9 | 9 | 0

$\xrightarrow{\text{off-spr}}$

x_5': 9 9 0 9 9 9 0 0
x_6': 9 9 0 0 9 9 9 0

$f(x_1') = \max[(9+9) - (0+0) + (9+9) - (9+0)] = 27$

$f(x_2') = \max[(9+9) - (0+0) + (9+9) - (0+0)] = 36$

$f(x_3') = \max[(9+9) - (0+9) + (9+9) - (9+0)] = 18$

$f(x_4') = \max[(9+9) - (0+0) + (9+9) - (9+0)] = 27$

$f(x_5') = \max[(9+9) - (0+9) + (9+9) - (0+0)] = 27$

$f(x_6') = \max[(9+9) - (0+0) + (9+9) - (9+0)] = 27$

[Here sir has followed i.e. (given) so I'll do same]

[Otherwise can be done randomly. (simply e.g.)]

optimal val. (fitted val.) = 27 & 36

[While sort optimization see the highest 2 vals & take that pair where the diff. b/w those 2 vals is min.]

Assign (7)

Ans. Both rank-based steady state & tournament selection are selection operators used in genetic algos to favor better individuals for the mating pool. They are considered more effective than proportional selection in maintaining convergence pressure.

Rank-based Steady State

i) Ranks the population & assigns fitness to each chromo. based on its rank.

ii) Employs a combination of block death & ranking-based birth.

iii) Maintains diversity & avoids ~~prem~~ premature convergence.

Tournament Selection

i) Involves running 'tournaments' among randomly chosen individuals.

ii) The winners of each tournament is selected for crossover.

iii) Tournament size can be adjusted to control selection pressure.

iv) Deterministic tournament selection chases the best individual in each tournament.

Diff.

Feature	Rank-based Steady State	Tournament selection
i) ~~feat~~ selection mechanism –	Ranking based fitness	tournament competition.
ii) selection pressure –	Lower	Higher
iii) Implementation complexity –	more complex	Simpler
iv) Computational cost –	Higher	lower
v) Diversity –	maintained well	maintained well.

- Choice b/w the 2 depends on the factors like the Problem complexity & desired selection pressure.

NP-complete Problem meaning – In CS, an NP-complete problem is a computational problem that's considered to be among the most difficult problems in the NP (non-deterministic polynomial) class. / NP-complete problems, any of a class of computational problems for which no efficient soln algo has been found.

dN

Artificial neural network (ANN) 17/9/24

KNN

mat — label

X_train — y_train

X_test — y_test

i) $M = f(X\_train, y\_train)$ — learning parameter (depending upon regressor, classifier)

ii) $y\_pred = m(X\_test)$ fixed

iii) $Acc = P(y\_test, y\_pred)$

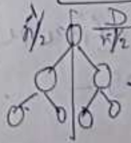
[f = classifier
m = model
y-pred = predicted C.L.
Acc → Accuracy]

- K-NN ⟶ $\boxed{(K=1)_{NN} = MDC}$
- the dim of 2 feature sets will be always same otherwise can't apply euclidean distance. (MDC = min distance classifier = euclidean distance)

$[d = 1 \times N \checkmark$, $d = 2 \times N$; $d = 1 \times N_1$, $d = 1 \times N_2$ ✗ $dist(d, d)]$

- $\boxed{max \propto \frac{1}{min}}$

Decision tree



$\frac{D}{n/2}$, $n/2$ ⟶ $T \to O(n \log n)$

KNN

$\to O(m^2)$, m

- $O(D) << O(MDC)$
Decision tree, KNN

• If the no. of features are more, then the KNN will fail, but decision-tree will pass.

• KNN is good for low amount of features.

• If the features are very large then decision-tree also fails, that's why ANN is used then.

When to use which classifiers

• The main factor is the classifier to be used to which data set

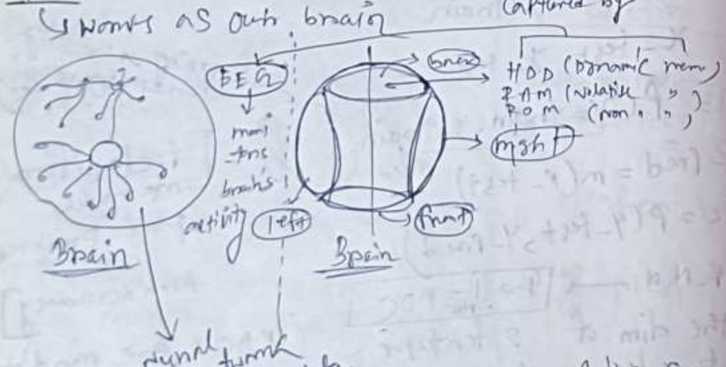
• If the features are too high both D (Decision tree), MDC (KNN alternative) like fail.

↓ Remedy

ANN

ANN

↳ works as our brain



• We are manipulating data by our own choice so it is called ANN & considered as Soft Computing (History - Read once)

ANN

(1) It is defined based on an architecture, that is composed of some layers—

i) Input layer

ii) Hidden layer or layers

iii) output layers

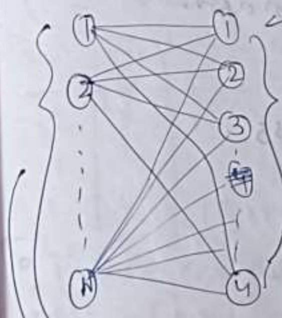
[(more hidden layers → more tc → less performance)]

• I/P, O/P → fixed

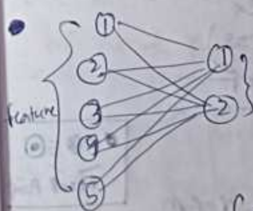
• Hidden layer/layers — 1 or more

• For any NN there is req. of atleast 2 hidden layers; if more complex → more H.L (Hidden lay.)]

A general architecture of ANN with 1 i/P (n features), 1 O/P (m-class) is given by —



<2-layer arch> [I/P ⇒ no. of features, O/P ⇒ no. of classes]

• 

classes → $\boxed{\text{No. of connections} \Rightarrow n \times m}$ $\Rightarrow 5 \times 2 = 10$

(Each feature is a node) i.e. each node is a feature

(Feature ∝ nodes)

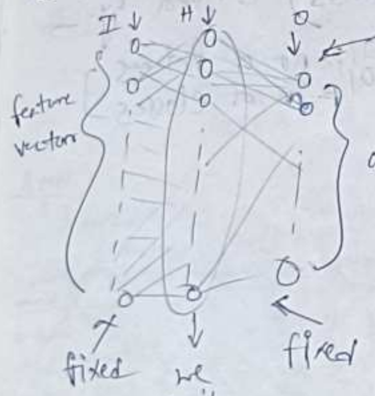
↓

5-2 NN

eg = 15-2, 1000-2 (1000 × 2) ---
(15 × 2)

- High no. of features → ANN will not work
  [eg = N × m = 10,000 × 2 = 20,000 connections]
  ↓ Remedy
  PCA

② A N-P-M NN is given by –



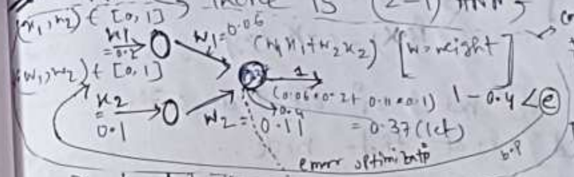
(Computational task → more complex math. forms)

[Test of Convergence – How many no. of layers will be req. The performance should be optimum.] ||

③ experimentally, it is not proven for a N-P-M NN arch, the no. of nodes in the hidden layer (P) / the no. of hidden layers are required to train an ANN. Sometimes we need a test of convergence to fix the arch. of ANN for the particular problem. This test of convergence needs optimizers, that is optimization algos.

③ Working flow of a 2-layer architecture –

Suppose, there is (2-1) ANN –



continuous process (error, process) till the val. converged – epoch

x_1	x_2	y
2	3	1
1	2	1
1	4	1

$f(x) = \tan(x)$
$\Rightarrow 0.37 = \tan(0.37)$
$\simeq 0.4$

Class == class problem

Dominance Priority

if $x \in [0, 2]$, $w \in [-1, 1]$

Brain: I/p → activation → Productivity

$e = |e_2 - e_1|$ = error optimization

- The range will not affect O/P, but weight vector must $\in [0, 1]$. But who will set w? RNG → Random no. generator

- If at o/p, can 0.37 reach to 1? If 0.37 reaches 1, it will not give any processing, but we need processing to enhance our model.

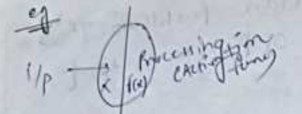
The working flow of a (N-M) ANN are based on –

i) N i/p features within the node.

ii) Assign the random values to the weight matrix $W \in [0, 1]$

iii) Perform process within the nodes of next layer.

eg



Activation func = tan, sigmoid etc. The rest will depnd upon the activ func

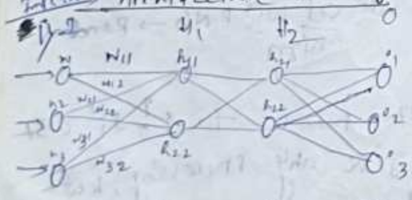
Here for i/p $x = w_1x_1 + w_2x_2 + \dots + w_Nx_N$
& processing means, activation func
$f(x) = e^{-x}$ (say)

iv) Compare the result of o/p node with the predefined val. at that node.

v) adjust the ~~weight~~ outcomes at the processing node with the optimizer.

Imp (SA)
① Diff optimizers – Gradient descent, NAG, RmsProp etc.
② Diff activation func – sigmoid, ReLU, Softmax, tanh etc. (their effects)

Imp (exam) Architecture drawing

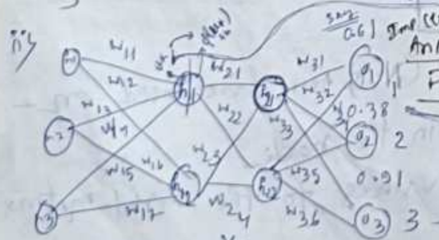


→ (h_{11}) & (h_{21}) what will be the w? → This is confusing so this

draw (exam)

3 – 2 – 2 – 3 (given)

ii)



0.61 Imp (exam)
Archi will be given, find the expression & value

Individual processing

$h_{11} = (\overbrace{w_{11}x_1 + w_{13}x_2 + w_{15}x_3}^{x_1}) \to$ exp. for h_1 (exam) $+ f(x_1)$ [∵ h_1 here there is 2 parts]

$h_{21} = (h_{11}w_{21} + h_{12}w_{23}) + \dots$ so on

Not to confuse

→ error (Root mean square error)

$f_2 = \sqrt{(1-0.61)^2 + (2-0.38)^2 + (1-0.91)}$

(rmse) $= 1.37$ (say) ↳ error value

- $1.37, 1.30, 1.02 \dots$ (error minimization → EM)
- error decreasing ⟶ performance increasing
  ✓ if not overfitting
  else underfitting

BP = Back Propagation

Imp (exam)
Assignment ⑧
i) Draw a 3-5-3 ANN architecture.
ii) Assign the weight values in the archi., with giving the i/p vector $\begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$, o/p vector $\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$
[Store the weight val. in weight matrix]

Forward & backward processing (epoch), batch

DP

$m_1, m_2 \leftarrow m_n$ (mcm)

- Brute force ($^{m}C_2$)
- DPP → $O(m^3)$
- computational (numerical methods) ⟶ $O(m^2)$ $O(m^T m)$ → transpose

& processing means, activation func

$f(x) = e^{-x}$ (say)

... of o/p node with

b.p = Back Propagation

now, for this (ii) the correct exp. & value will be

<u>exp</u>

$h_{11} = \varphi(u_k + b)$ let, $b=0$

(A.F = will be given) (if not given then jus write the exp.)

$= \varphi(w_{11}x_1 + w_{13}x_2 + w_{15}x_3 + b)$

(theme) f

$\therefore h_{11} = \varphi(w_{11}x_1 + w_{13}x_2 + w_{15}x_3 + b)$ → exp. of h_{11}

now, <u>value</u>

if, say given sigmoid A.F.

then, $h_{11} = \varphi(\underbrace{w_{11}x_1 + \dots + b}_{x})$, $b=0$

Say, $x = 2.1$

sigmoid AF $= \dfrac{1}{1+e^{-ax}}$

let, $a = 0.1$ $\therefore = \dfrac{1}{1+e^{-0.1\cdot 2.1}}$

$= 0.6$ (say)

$\therefore \varphi(2.1) = 0.6$ → value of h_{11}

$h_{21} = (h_{11}w_{21} + h_{12}w_{23}) + \dots$ so on

In h_1 (exam) ... $f(x_1)$ [∵ h_{11} here theme 2 parts] processing means, activation

$f(x) = e^{-x}$ (say)

Compare the result of o/p node with

b = bias

$h_{11} \xrightarrow{y_k} \left(\begin{matrix} w_1 \\ w_3 \\ w_5 \end{matrix} \Sigma \right) \xrightarrow{u_k} \boxed{\begin{matrix} A.F \\ \varphi(\cdot) \end{matrix}} \xrightarrow{y_k}$

$y_k = \varphi(u_k)$
$= \varphi(u_k + b)$

$u_k = \sum_{j=1}^{m} w_{kj} * x_j$

$y_k = \varphi\left(\underbrace{\sum_{j=1}^{m} w_{kj} * x_j + b}_{x}\right)$

$= \varphi(x)$

depending upon x

Threshold A.F | Sigmoid A.F | tangent hyperbolic A.F

let, Sigmoid A.F (given)

let $x = 5$, $\therefore y_k = \varphi(x)$
$a = 0.1$ $\quad = \varphi(5)$

$\frac{1}{1+e^{-ax}} = \frac{1}{1+e^{0.1 * 5}}$ $\quad = 0.56$

$= 0.56$ (say)

$\therefore y_k = 0.56$

this is the o/p of h_{11}

$\therefore h_{11} \xrightarrow{y_k = 0.56}$

this is the Correct explanation

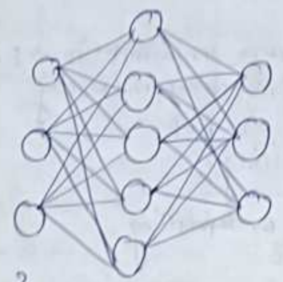
P.T.O

$h_{11} = (w_1, x_1$

$(h_{11} w_{21} + h_{12} w_{23}) + \ldots$

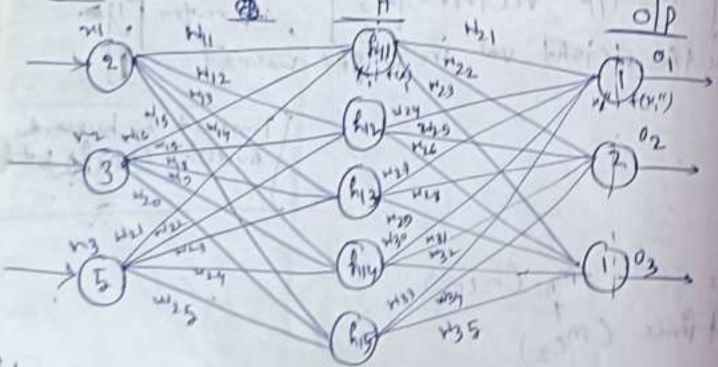
Ans to (8)

i) A 3-5-3 ANN archi.



3 - 5 - 3 ANN

ii) I/P



i/p vector = $\begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$ o/p vector = $\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$

let $w_{ij} \in [0, 1]$

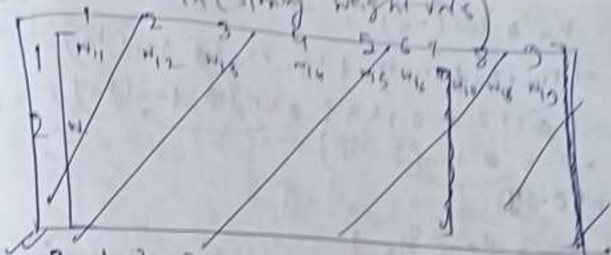
[let, h_{ij} / H $= f(W \cdot x) + f(x')$

$= f(w_1 x_1 + \cdots + w_{np} x_{n})$ [n-1 ...]

let, $f(x')$ [Productivity part / Processing / activation func] $= \tan(x')$

Si tangent ...
$\phi(x) \to f(x)$ ...
$\phi : f(x)$ ...

let, weight matrix (storing weight vals)



$$\begin{array}{c|c} & \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} - & w_{11}=0.1 & w_{12}=0.3 & w_{13}=0.1 & w_{14}=0.1 & w_{15}=0.4 & w_{16}=0.5 & w_{17}=0.1 & w_{18}=0.6 & w_{19}=0.1 \\ w_{20}=0.1 & w_{21}=0.1 & w_{22}=0.1 & w_{23}=0.2 & w_{24}=0 & w_{25}=1 & w_{26}=0 & w_{27}=0 & w_{28}=0 & w_{29}=0 \\ w_{30}=0.1 & w_{31}=0.1 & w_{32}=0 & w_{33}=0 & w_{34}=0 & w_{35}=0 & & & & \end{bmatrix} \end{array}$$

(all of the weight values are assumed)

$h_{11} = x_1' + f(x_1') = f(x')$

$= x_1' + \tan(x_1') = \tan(x')$ (like tangent hyperbolic)

$= (w_{11} x_1 + w_{16} x_2 + w_{21} x_3) + \tan(x_1')$

$\boxed{\text{ultimately } x_1' \text{ is going in A.F. so has no effect}}$

$= (0.1 \times 2 + 0.5 \times 3 + 0.1 \times 5) + \tan(x_1')$

$= 2.2 + \tan(2.2)$

$= 0.04$

$\simeq \boxed{0.04}$ (o/p of h_{11})

(but this isn't correct approach) $\boxed{\text{P.T.O}}$

$h_{12} = x_2' + \tan(x_2')$

$= (w_{12} \times x_1 + w_{17} \times x_2 + w_{22} \times x_3) + \tan(x_2')$

$= (0.2 \times 2 + 0.1 \times 3 + 0.1 \times 5) + \tan(x_2')$

$= 1.2 + \tan(1.2)$

$\simeq \boxed{0.02}$

$h_{13} = x_3' + \tan(x_3')$

$= (w_{13} \times x_1 + w_{18} \times x_2 + w_{23} \times x_3) + \tan(x_3')$

$= (0.1 \times 2 + 0.6 \times 3 + 0.2 \times 5) + \tan(x_3')$

$= 3 + \tan(3) = \boxed{0.05}$

$h_{14} = h_4' * \tan(h_4')$
$= (w_{14} * h_1 + w_{19} * h_2 + w_{24} * h_3) * \tan(h_4')$
$= (0.2 * 2 + 0.7 * 3 + 0 * 5) * \tan(h_4')$
$= 2.5 * \tan(2.5)$
$= \boxed{0.09}$

$h_5 = h_5' * \tan(h_5')$
$= (w_{13} * h_1 + w_{20} * h_2 + w_{25} * h_3) * \tan(h_5')$
$= (0.4 * 2 + 0.1 * 3 + 1 * 5) * \tan(h_5')$
$= 6.1 * \tan(6.1)$
$= \boxed{0.21}$

let, O_{3} / $O_m = f(W.H) * f(x'')$
$= f(w_{21} * h_{11} \text{ ---- } w_{np} * h_p)$ [p = 11, 12, 13, 14, 15]

let, $f(x'') = \sin(x'')$

$O_1 = x_1'' * f(x_1'') = f(x_1'')$
$= x_1'' * \sin(x_1'') = \sin(x_1'')$
$= (h_{11} * w_{21} + h_{12} * w_{24} + h_{13} * w_{27} + h_{14} * w_{30} + h_{15} * w_{33}) * \sin(x_1'')$

$= (0.04 * 0.1 + 0.02 * 0 + 0.05 * 0 + 0.05 * 0.1 + 0.1 * 1) * \sin(x_1'')$

$= 0.109 * \sin(0.109)$
$= \boxed{2 * 10^{-3}}$

$O_2 = x_2'' * \sin(x_2'')$
$= (h_{11} * w_{22} + h_{12} * w_{25} + h_{13} * w_{28} + h_{14} * w_{31} + h_{15} * w_{34}) * \sin(x_2'')$
$= (0.04 * 0.1 + 0.02 * 1 + 0.05 * 0 + 0.05 * 0.1 + 0.1 * 0) * \sin(x_2'')$

$= 0.029 * \sin(0.029)$
$= \boxed{5.1 * 10^{-4}}$

$O_3 = x_3'' * \sin(x_3'')$
$= (h_{11} * w_{23} + h_{12} * w_{26} + h_{13} * w_{29} + h_{14} * w_{32} + h_{15} * w_{35}) * \sin(x_3'')$

$= (0.04 * 0.2 + 0.02 * 0 + 0.05 * 0 + 0.05 * 0.2 + 0.1 * 0) * \sin(x_3'')$
$= 0.018 * \sin(0.018)$
$= \boxed{3.1 * 10^{-4}}$

$\therefore O_1 = 2 * 10^{-3}$, $O_2 = 5.1 * 10^{-4}$, $O_3 = 3.1 * 10^{-4}$

depending upon this o/p vals. other calc. will be done & Prediction will happen.

we can now calculate RMSE, MSE etc & then by doing EM we can perform convergence & then can Predict like that

$e(RMSE) = \sqrt{(1 - 2 * 10^{-3})^2 + (2 - 5.1 * 10^{-4})^2 + (1 - 3.1 * 10^{-4})^2}$

$= 2.448$
$\simeq 2.5$

Prediction may will be given like, if $O_1 > 0.5 \to$ dog, $O_3 < 0.1 \to$ cat — like this

fresh (P.T.O)

$h_{14} = x_4' * \tan(x_4')$

(1)

i/p vector = $\begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$ o/p vector = $\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$

$w \, [w_{kj} \in [0,1]]$

[m = no. of i/ps; i = 1, 2, 3 …; j = 1, 2, 3 … m]

$h_{kj} / H = \varphi(x)$

$\varphi(x) = \varphi(ux + b)$ let b (bias) = 0

$= \varphi\left(\sum_{j=1}^{m} (w_{kj}' * x_j) + b\right)$

$= \varphi\left(\sum_{j=1}^{m} w_{kj} * x_j + b\right)$

let, A.F,

$\varphi(x) = \tan(x)$

(no need to do in exam)

$\therefore h_{11} = \varphi(w_{11} * x_1 + w_{16} * x_2 + w_{21} * x_3 + b)$

$= \varphi(2.2 + 0)$

$= \varphi(2.2)$

$\varphi(x) = \varphi(2.2)$, $\therefore x = 2.2$

$\therefore \varphi(2.2) = \tan(2.2) = \boxed{0.04}$ (y_k o/p of h_{11})

$h_{12} = \varphi(w_{12} * x_1 + w_{17} * x_2 + w_{22} * x_3 + b)$

$= \varphi(1.2 + 0)$

$= \varphi(1.2)$

$= \boxed{0.02}$ [$\because \varphi(x) = \varphi(1.2) = \tan(1.2) = 0.02$, $x = 1.2$]

$h_{13} = \varphi(w_{13} * x_1 + w_{18} * x_2 + w_{23} * x_3 + b)$

$= \varphi(3 + 0)$

$= \varphi(3) = \boxed{0.05}$

2

$h_{14} = \phi(w_{14} * h_1 + w_{19} * h_2 + w_{24} * h_3 + b)$

$= \phi(2.5 + 0)$

$= \phi(2.5)$

$= \boxed{0.04}$

$h_{15} = \phi(w_{15} * h_1 + w_{20} * h_2 + w_{25} * h_3 + b)$

$= \phi(6.1 + 0)$

$= \boxed{0.1}$

Let, O/$O_m = \phi(x')$

$\Rightarrow \phi(x') = \phi(u_K + b)$

$= \phi\left(\sum_{j=1}^{m} w_{kj} * h_{kj} + b\right)$ [i.e. here $h_j = \ldots$]

A.F.

Let's here $\phi(x') = \sin(x')$

(no need to do in exam)

$O_1 = \phi(h_{11} * w_{21} + h_{12} * w_{24} + h_{13} * w_{27} + h_{14} * w_{30} + h_{15} * w_{33} + b)$

$= \phi(0.109 + 0)$

$\phi(x') = \phi(0.109)$, $\therefore x' = 0.109$

$\therefore \phi(0.109) = \sin(0.109) = \boxed{2 * 10^{-3}}$

(2 of O_1 final o/p)

$O_2 = \phi(h_{11} * w_{22} + h_{12} * w_{25} + h_{13} * w_{28} + h_{14} * w_{31} + h_{15} * w_{34} + b)$

$= \phi(0.029)$

$= \boxed{5.1 * 10^{-4}}$

0.017 ϕ sin(1.079)

3

$O_3 = \phi(h_{11} * w_{23} + h_{12} * w_{26} + h_{13} * w_{29} + h_{14} * w_{32} + h_{15} * w_{35} + b)$

$= \phi(0.018 + 0)$

$= \phi(0.018)$

$= \boxed{3.1 * 10^{-4}}$

$\therefore O_1 = 2 * 10^{-3}$, $O_2 = 5.1 * 10^{-4}$, $O_3 = 3.1 * 10^{-4}$

Let, error optimization process is RMSE (may or may not be given in exam)

$\therefore e(RMSE) = \sqrt{(1 - 2 * 10^{-3})^2 + (2 - 5.1 * 10^{-4})^2 + (1 - 3.1 * 10^{-4})^2}$

$= 2.448$

$\simeq 2.5$

(If not asked then ok, if asked then the process (i.e. which process to use to get the 'e') will be given.)

MDC - [This is used to classify unknown image data to classes which minimize the distance b/w the image data & the class in multi-feature space.] //

Test of Convergence - A test of Convergence in a neural network checks if the model has finished learning. It happens when the error (or loss) stops decreasing after training for a while, meaning the model is as good as it can get with the current data. [This helps ensure the model has learned as much as it can from the data & prevents overfitting or underfitting.] //

Optimization algo - It is a/an algo mathematical procedure that finds the best soln to a problem by minimizing or maximizing a func. Eg - gradient descent, GA, Bayesian optimization etc.

Optimizer - An optimizer in a nn is an mathematical algo that helps improve the n/ws performance by adjusting its attributes, such as weights & learning rate.

Dominance Property - [A row is dominated if all its elements are less than or equal to the corresponding elements in another row, with at least one element being strictly less.] It is a way to compare strategies in a game to determine if one strategy is better than another.

Overfitting & underfitting - [They are 2 types of errors that can occur when training a machine learning model.] //

Overfitting - A model that performs well on training data but poorly on test data. [This can happen when a model is too complex, overfits a single data set, or learns too many details from the training data.] //

Underfitting - A model that performs poorly on training data & is unable to generalize to new data. [This can happen when a model is too simple or doesn't train long enough on enough data points.] //

Activation function - It determines which neurons should be activated as info. passes through the n/w. [Activation funcs are imp. as they allow neural n/ws to learn complex patterns & relationships in data.] //

Eg - Sigmoid, tanh, ReLU, Softmax, ELU etc.

Effects - i) These introduce non-linearity to a neural n/w, enabling it to learn complex patterns.

ii) Funcs like ReLU or Sigmoid determine how neurons fire & affects how well the n/w can capture relationships in the data.

Effects of optimizers - i) ~~It~~ these control how a neural n/w learns by adjusting the model's weights.
ii) Different optimizers like Adam or SGD affect the speed & stability of learning, influencing how fast the model converges & how well it avoids getting stuck in bad sol^ns.

Fn or error minimization - It is the process of reducing or correcting errors.

Forward Processing | Propagation - Here, the i/p data passes through the layers of the neural n/w, with each layer applying weights & activation funcs to compute the o/p (or prediction).

(2) Backward Processing | Propagation - Here, the n/w calculates the error b/w the predicted & actual values, then adjusts the weights by moving backward through the layers, using techniques like gradient descent to minimize the error.

Epoch - An epoch refers to one complete pass of the entire training dataset through the neural n/w during training. It helps the model learn from the data in multiple rounds.

Batch - A batch is a subset of the training data processed together before updating the model's weights. Instead of updating after each data point, the model updates after processing a batch, improving computational efficiency.

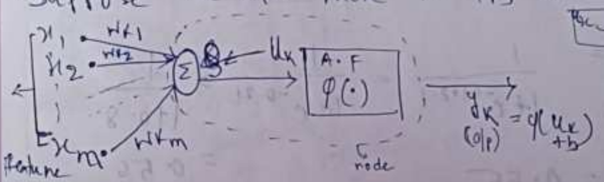
Loss function - It measures how well a neural n/w performs a task by calculating the diff. b/w the n/w's predictions & the actual values. (Aka cost func) (3)

19|9|24

ANN

Activation func (mathematical formulation)

Suppose a n/w where the i/ps are



& u_k be the processing node of a vector.

$$\boxed{u_k = \sum_{j=1}^{m} w_{kj} * x_j}$$

$$\boxed{y_k = \varphi(u_k)} = \varphi(u_k + b)$$

$$\varphi(u_k)$$

$$y_k = \varphi\left(\sum_{j=1}^{m} (w_{kj} * x_j) + b\right)$$

~~bias will be added to every $w_{kj} * x_j$~~

it will not produce much diff. from u_k, but will stop error

$e - y \cdot L$ → some error → bias → b

[• Bias (b) is always constant (0 to 1 → range)
↓
Activation func contains some error, to stop it we use bias.] → bias

of reducing ...

Forward Processing | propagation - Hence ...

the i/p data passes through the layer ... each layer ... neural n/w, with ... activation func ...

ANN

Activation

suppose

x_1, x_2, ..., x_m

feature vector &

$$y_k = \varphi\Big(\overbrace{\underbrace{\sum_{j=1}^{m}(w_{kj} * x_j)}_{u_k} + b}^{X}\Big)$$

$$y_k = \varphi\Big(\sum_{j=1}^{m}(w_{kj} * x_j + b)\Big) \rightarrow X$$

n/w

& actual values,

by moving backward

using techniques like gradient descent to minimize the error

An epoch refers to one complete ... through

u_k

$\varphi($

[eg. tan
$y(x) = \tan x$
$y(x+2) = \tan(x+2)$] → the diff. will not be much it will be very very small

- With the diff. formulation of $\phi(x)$ the diff. activation funcs are defined —

i) Threshold func

$$\boxed{\phi(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & x < 0 \end{cases}}$$

ii) Sigmoid activation func (A.F.)

$$\boxed{\phi(x) = \frac{1}{1+e^{-ax}}, \quad a = \text{learning Parameters}, \quad a \in [0,1]}$$

eg.

$\phi(x) = $ let $x = 2.1$ $\frac{1}{1+e^{-0.1 \times 2.1}} = \frac{1}{1+e^{-0.21}} = \frac{1}{1+0.8}$
$= 0.56$

$\therefore \phi(x) = 0.55$

$\therefore y_r = \phi(x) = 0.55$

let 1 set at O/P, $\therefore 1 - 0.55$ (error)

[• Forward & backward Pro. → W change → tuning]

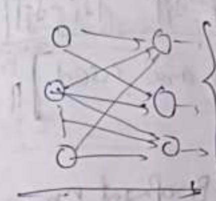
iii) Tangent hyperbolic func

$\phi(x) = \tanh(x)$

[• $\tan(x)$ v/s $\tanh(x)$]
Range ($-\infty$ to ∞) | Range will always give +ve (-1 to $+1$)

misterious topic

① F.F n/w



FF → Feed forward

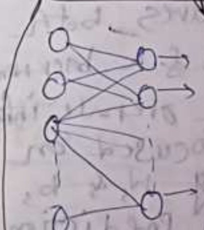
predicted level | actual level
y_j | y_i
y_8 | y_1
y_9 | y_2
y_{10} | y_3
y_{11} | y_4

$|y_1 - y_8|$
$|y_2 - y_9|$
$|y_3 - y_{10}|$
$|y_4 - y_{11}|$
→ (wrong eg)

Incom, $x_1 \ldots x_m$ given, $w_{k1} \ldots w_{km}$ given and also be v_k, $\phi(x)$, y_k being activation func

main diff. Doesn't need optimization.

② Back Propagation n/w



y_p	y_a
1	0
—	1
—	0

predicted val. | actual val.

Optimize error

eg of this (the upper side) not the eg of upper.

main diff. (needs error optimization)

The nature of error optimizer — suppose y_i be the ith actual label & y_j be the predicted jth level, so the error optimizer uses 2 diff. distance metric —

i) $\|.\|_1$ (Norm1)

ii) $\|.\|_2$ (Norm2)

whereas, $\boxed{\|.\|_1 \text{ means } |y_i - y_j| \rightarrow \text{abs (absolute difference)}}$

i.e, $\boxed{\Sigma \, abs(y_i - y_j)}$ (norm$_1$)

(norm$_2$) $\|.\|_2$ means $|y_i - y_j|^2$, $|y_i - y_j| \, |y_i ...$

but actually

$\boxed{|y_i - y_j|^2 = |y_i - y_j|^T \cdot |y_i - y_j|}$

(model's learning parameters i.e, Ws & b)

Based on this error optimizers are used.

• diff b/w them(↑)

Feed forward nn	Back propagation nn
① defi	① defi.
② Info flows unidirectionally from i/p to o/p layer without a feed back loop.	② It involves both forward & a backward pass, the overall info flow is focused on adjusting Ws & bs based on prediction errors.
③ No need of error optimization here, as ~~we~~ we don't calculate error here (no feed back loop).	③ It needs error optimization (feed back loop.)

Evolution ~~form~~ of pattern recognition from nn problem

{unsupervised learning}

I/P vector → unsupervised learning → Predict score + learning algo

Supervised learning

I/P vector → supervised learning → Supervised classification

• ern, etc everything will be based on supervised l., in the o/p always there will be unsupervised l.

performance learning → where we will put the diff etc (making decisions)

Perceptron theory

$$\boxed{y_k = \varphi\left(\underbrace{\sum_{j=1}^{n}(w_{kj} * x_j) + b}_{X}\right)}.$$

Here X is ~~...~~ hyperplain func, in the perceptron learning the hyperplain func is reduced to hyperplain eqn, that is $\boxed{X = 0}$, i.e.,

$$\boxed{\left(\sum_{j=1}^{K}(w_{kj} x_j) + b\right) = 0}$$

no need of ( )'s, i.e. $\sum w_{kj} x_j + b$

During learning the n/w for 2 class problem this eqn is reduced to 2 eqns;

$\boxed{\sum_{j=1}^{m} (w_{kj} x_j + b) \geq 0}$ --- ① } hyper plains ↓ means (4D)

$\boxed{\sum_{j=1}^{m} (w_{kj} x_j + b) < 0}$ --- ②

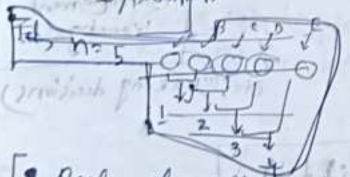
[j is depended on no. of i/ps]

The evolution of support vector machine using nn (svm in nn)

[• 2 class problem → Basic
• multiclass problem → ○ (○○ --- ○○) — treating as one, treating as 1-ve — 1 vs all, 2 class]

[• if n no. of classes then n binary forms/classification] — binary forms = classifier

Ex → n = 5



as

[• each classifier is trained to distinguish 1 class from the rest.]

eg, 2 ~~clusters~~ clusters (spam, not spam)

[2 binary forms (1 will distinguish spam, 1 — not spam) classifier]

The evolution of svm in neural n/ws involves combining the 2 methods, where neural networks learns to extract features from data, & svm is used as the final layer to make decisions. This helps improve classification by combining deep learning's feature extraction with svm's strong separation of classes.

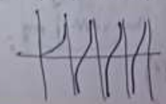
[Threshold func - It is an activation func that determines ~~th~~ whether a neuron should be activated or not based on whether the i/p val. is above or below a certain threshold.

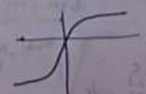
Sigmoid A.F. - It is a nonlinear func that's used in nns to help them learn from & understand complex patterns in data. (a.k.a logistic activation func)

Tangent hyperbolic func - the hyperbolic tangent (tanh) function is a nonlinear activation func ~~used~~ used in nns, particularly in hidden layers, autoencoders, & recurrent nns (RNNs). (It is also used in certain layers of feedforward nns).] ||

(no need)

tan	vs tanh
1) unbounded ($-\infty$ to ∞)	1) Bounded (-1 to 1)
2) Periodic	2) Non-periodic





Feed forward nn - It is a type of ~~anti~~ ANN that learns from i/p data to perform tasks like image classification. (aka multi layer perceptron (MLP))

Back Propagation nn - It's a ~~pe~~ nn that uses the back propagation algo to train itself through error correction.

2 class Problem - It is a binary classification problem, where the data is classified into 2 mutually exclusive catagories. Here, 1 class is usually considered as the normal state, while the other is considered as abnormal state. Eg -

i) Spam detection = 'not spam' (normal state)
'spam' (abnormal ")

multiclass ~~cla~~ Problem - It is a classification problem, that is a task that involves classifying data into more than 2 classes. [(Here a ml model learns patterns specific to each class, & then predicts, which class the new data belongs to.)] Eg -

i) [Deciding if an image contains a banana orange or apple (3 class, so 3 binary forms ban. | org & app; ban, app. | org; ban, org | app)

ii) Labelling a set of fruit images that includes oranges, apples, pears, bananas etc.]

24/9/24

Kernel function

In ML a data is rep. in the form of vector x. This x has a fixed dimention ~~to, depends~~ 'd', since the values in x are real, rep. will be $\boxed{x \in R^{1 \cdot d}}$ [real values (R)]

Suppose,

x be the subset of X, $[x \subseteq X, X \in R]$
where $X \in R$, (may not have fixed dim).
Then, to find the similarity b/w the various pattern of $x_1, x_2 \ldots x_n$ $x_i \in X$, requires a kernal function, this function is rep. by $\boxed{K(x, x')}$. This kernal function finds the similarity b/w the 2 ~~or more~~ patterns. The usability of this func are used in all classifiers & clustering algos / methods.

Types of kernal func

i) Linear K.F - It is rep. as,

$\boxed{K(x, x') = \langle \varphi(x), \varphi(x') \rangle}$ [$\varphi(x)$ means linear pattern of x
$x' \to$ transpose of x

$\langle\rangle \rightarrow$ inner product b/w 2 vectors (v_1, v_2)

eg $\langle (2\ 3\ 4) \begin{pmatrix}2\\3\\4\end{pmatrix} \rangle = 16$

ii) Polynomial K.F —

$$\boxed{\begin{aligned} K(x,x') &= \langle \phi(x)\ \phi'(x') \rangle \\ &= \langle x, x' \rangle^d \end{aligned}}$$

[$\phi(x)$ = Polynomial pattern of x]
like eg $\phi(x) = \cos(x^2 + c)$

$d = 2 \rightarrow$ binomial
$= 3 \rightarrow$ trinomial

iii) Radial basis K.F. —

$$\boxed{\begin{aligned} K(x,x') &= \langle \phi(x)\ \phi(x') \rangle \\ &= \exp\left(\tfrac{1}{2} \|x - x'\|^2\right) \\ &= \exp\left(\tfrac{2}{2} x^T x - \tfrac{1}{2}\|x\|^2 - \tfrac{1}{2}\|x'\|^2\right) \\ &= \exp(x^T x) * \exp(-\tfrac{1}{2}\|x\|^2) * \exp(-\tfrac{1}{2}\|x'\|^2) \end{aligned}}$$

where $\rightarrow$ [$\|x\|^2 = \|(2,3,4)\|^2 = 2^2 + 3^2 + 4^2$] $\Rightarrow$ np-complete timing

• The time complexity of this KF is np-complete.

$\boxed{\begin{aligned} &\|(7,9,0)\|^2 \\ &= 7^2 + 9^2 + 0^2 \\ &x = (7,9,0) \end{aligned}}$

iv) Graph K.F.

This is very complex K.F because the data structure of a graph, $G(V,E)$ is rep. by set of vertices $V = \{v_1, v_2, \ldots v_m\}$ & set of edges, $E = \{e_1, e_2, e_3 \ldots e_m\}$. so finding the similarity b/w 2 graph patterns requires both set of vertices & set of edges, so the rep. is complex & to finds the soln in the np-hard timing.

v) Convolution K.F. — This func is rep. as

$\boxed{K(x,x') = (\phi(x)\ \phi(x'))}$ [where x be a matrix]

Convolution $\rightarrow$ Finding this into this

[• How to find if 2 graphs are same
$\downarrow$
Find incidence matrix & adjacency matrix]

eg

$A = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}, B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ finding B in A

$$\boxed{\frac{A_1 B_1 + A_2 B_2 + A_3 B_3 + A_4 B_4}{\Sigma B_i}}$$

Linear K.F – It is a basic K.F used in ML when the data is linearly separable, that is, it can be separated using a single line. It's mostly used when there are a large no. of features in a particular data set.

Polynomial K.F – It is a K.F that's often used in SVMs to learn non-linear models. It works by mapping i/p data points into a higher-dimensional feature space using polynomial functions of the original features.

RBF K.F. – It is also known as kernel func, is applied to the distance to calculate every neuron's weight (influence). It's used to measure the similarity b/w 2 data points & is based on the euclidean distance b/w them.

Convolution K.F – Convolution kernels, aka filters, are small matrices (matrix in neural) used for the convolution operation. These kernels slide across the i/p data, performing element wise ~~mo~~ mul. with the corresponding pixels & producing a feature map that highlights specific patterns [features in the i/p.] ||

Rep. of Graph K.F. –

$$K(G_1, G_2) = \phi(G_1) \cdot \phi(G_2)$$ dot product

[where, G_1, G_2 = 2 graphs
$\phi(G)$ = feature rep. of graph G]

NP-hard problem (timing) – [A group of problems that if we can figure out how to solve one quickly, then we can solve any NP problem quickly]! NP hard problems are a class of computational problems that are considered to be at least as difficult as the most difficult problems in NP (non deterministic polynomial time).

(4) 1/10/24

Problem (lab)

① Consider a random ~~factor~~ vector that is a collection of random values of n dim (min n = 100). Fit Poisson distribution & gaussian distribution for this collected values.

② Use iris & heart disease database to perform i) classification task with 50-50% training & testing (protocol), using svm classifier.