Prepared and Written By

RAZEEN AHMED

CSE-422

[ARTIFICIAL INTELLIGENCE]

HANDWRITTEN NOTE

Computer Science and Engineering

BRAC University

Github: github.com/razeen

LinkedIn: linkedin.com/in/razeenahmed

11.00 TRZ sir class 28/02/25

INFORMED SEARCH (Slide) :→

TRZ — Online Class

Uninformed Search

(i) BFS

(ii) DFS

To address the weakness there are some other algorithm →

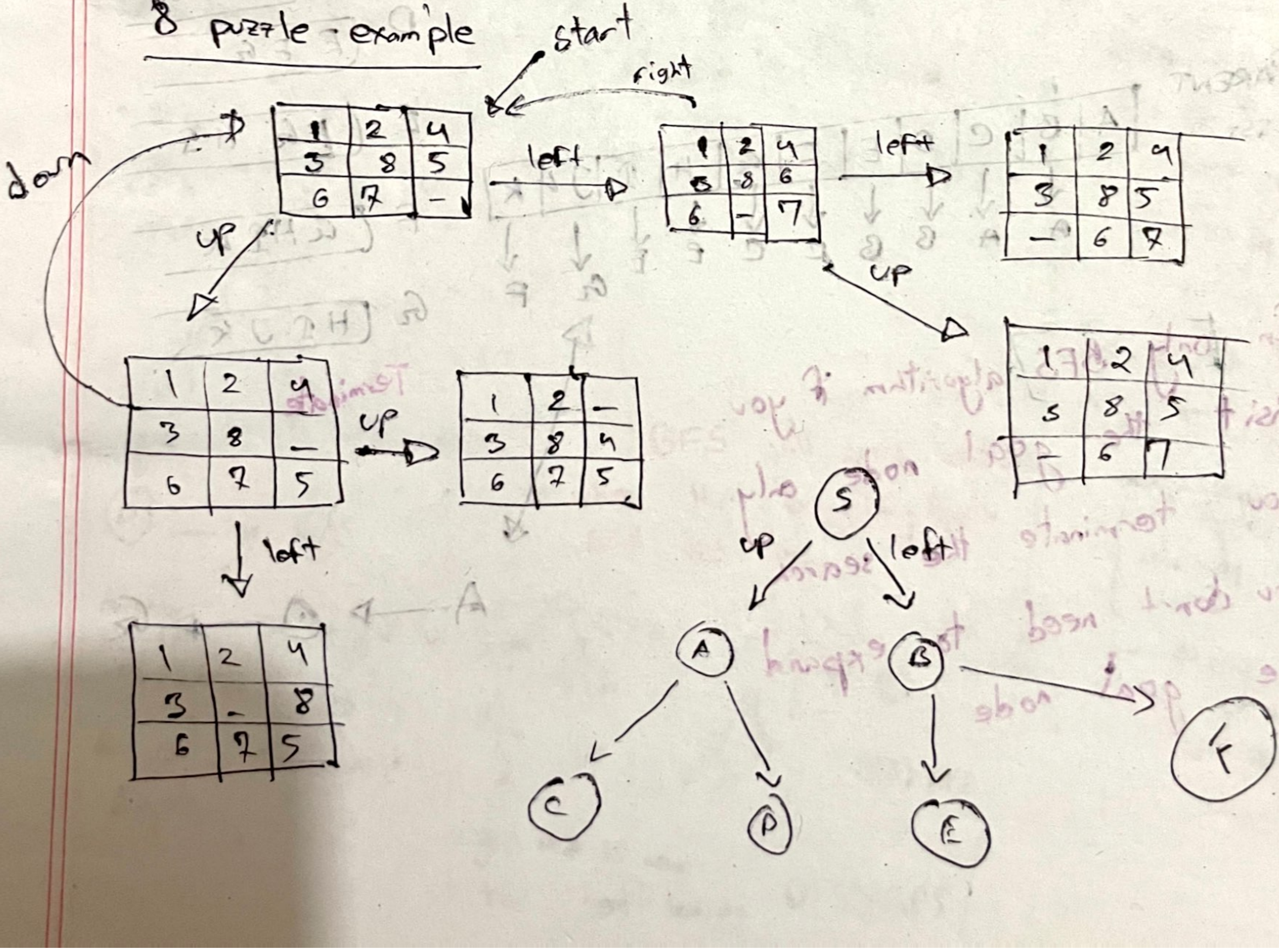
(i) Depth limited Search

(ii) Iterative deepening search

(iii) Uniform Cost Search

(iv) Bidirectional Searc

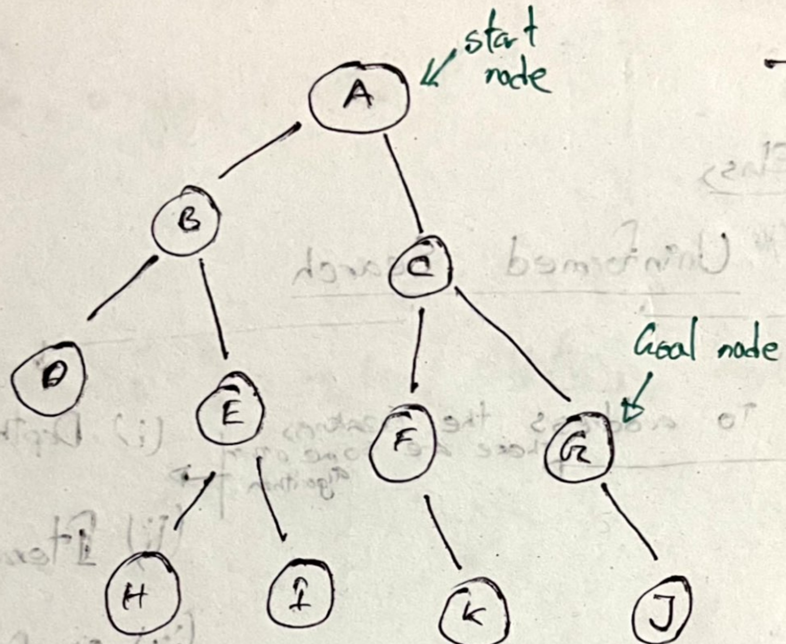
8 puzzle example



what is optimal!

→ If there is multiple path to reach a goal node and algorithm selects the most efficient way to reach it then its optimal. [In all possible scenario]

BFS → [BREDTH FIRST SEARCH]



→ It works well with unweighted graph

QUECE (FIFO)

	A
A	B C
B	C D E
C	D E F G
D	E F G
E	F G H I
F	G H I K
G	H I J K

Terminate

PARENT LIST

A	B	C	D	E	F	G	H	I	J	K
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	A	A	B	B	C	C	E	E	G	F

In [only BFS] algorithm if you visit the goal node only you terminate the search you don't need to expand the goal node

A → C → G

If a certain goal exists in a certain ~~tree~~ tree or state space and any searching algorithm guarantees to find that goal node in all possible scenario then its complete

Time Complexity:- $O(b^d)$ / $O(b^{d+1})$ expand goal node

Space Complexity: $O(b^d)$ / $O(b^{d+1})$ expand goal node

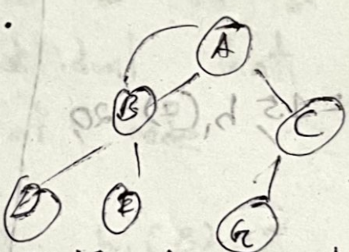
Completeness: Complete

Optimal:- ~~Not~~ optimal.

$b^0 + b^1 + b^3 + \dots - b^d$

$\downarrow$ 8 $\quad\downarrow$ 3 $\quad\downarrow$ 9

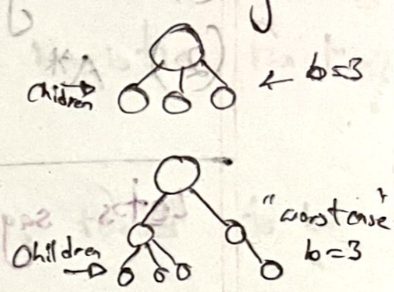
Why Complete?



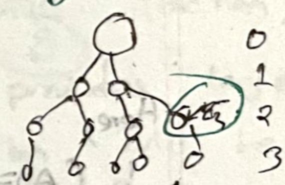
→ able to avoid loop

→ visits all nodes

b → branching factor



d → level of the goal node



d = 2

~~GRAPH~~ TREE SEARCH

↳ if there is no checking for a already visited node.

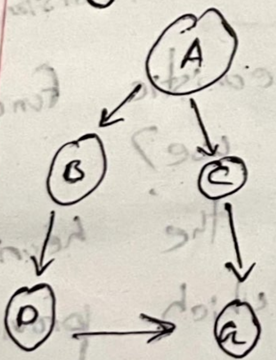
GRAPH SEARCH

↳ If there is a checking for a already visited node

→ why check?

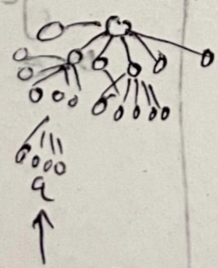
To avoid loop

Why Optimal?

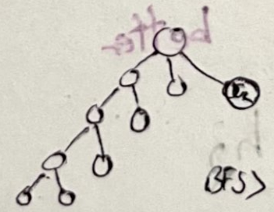


always selects the effecient path of shallowest depth.

BFS will work better when the branching factor will be low.



↑ goal node is near



BFS > DFS

DFS > BFS

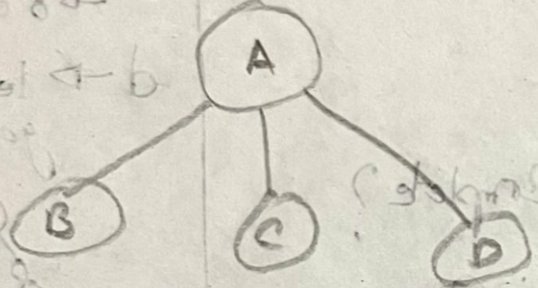
<u>7R7 Class</u>

INFORMED SEARCH

(1) Greedy Best first Search

(2) A* Search

Lets say →



Here,

$h_1(A) = 12$, $h_1(B) = 15$, $h_1(C) = 20$,

$h_1(D) = 9$

and,

$h_2(A) = 7$, $h_2(B) = 9$, $h_2(C) = 17$

$h_2(D) = 8$

We can observe "for all nodes"

$$h_1(n) \geq h_2(n)$$

So, h_1 will be better

h_1 dominates h_2

BETTER TO CHOOSE h1.

There may be multiple heuristic function for a single proble

For example:-

h_1 → No. of misplaced tiles

h_2 → Manhattan distance.

→ A single box is how much away from its desired position

1	3	2
4	7	8
6	5	

CURRENT

1	2	3
4	5	6
7	8	

GOAL

$h_2 = 1 + 1 + 2 + 2 + 3 +$

$\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$

$3 \quad 2 \quad 7 \quad 8 \quad 6 \quad 5$

→ using manhattan distance

which heuristic function should we use?

→ Usually the heuristic functions which provide us larger values that is the one we should choose

But! what if there are mixed number of heuristics which are greater between the two heuristic function?

Soln → Two way →

(i) We will select the heuristic function is found to be greater for most of the nodes.

(ii) We will select best of both the heuristic function. That is, the heuristic function which provides larger value for an individual that h function will be assigned for that node

example

$h_1(A) = 12$, $h_1(B) = 8$, $h_1(C) = 20$

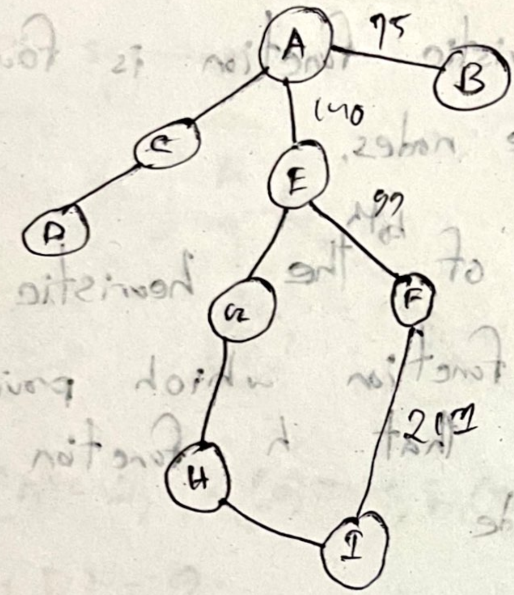
$h_2(A) = 7$, $h_2(B) = 9$, $h_2(C) = 12$

→ $h_f(A) = 12$, $h_f(B) = 9$, $h_f(C) = 20$

Combine

GREEDY BEST FIRST SEARCH.

Uses only heuristics [GRAPH



	A_{366}
A_{366}	E_{253} C_{329} B_{374}
E_{253}	C_{329} B_{374} F_{178} G_{193}
F_{178} ~~C_{329}~~	C_{329} B_{374} G_{193} I_0
I_0 ~~G_{193}~~	C_{329} B_{374} G_{193}

A → E → F → I

→ 140 + 99 + 211

→ 450

But!

→ 418 £ is the optimal path

So, GBFS is not providing a optimal path.

GRAPH GBFS:-

NOT OPTIMAL

~~GRAPH~~ complete: Yes [for finite tree]

NOTE:-

For "infinite tree" GRAPH GBFS is ~~ino~~ not complete.

[TREE] GBFS → Doesn't track the parent of the nodes.

~~Not~~ co.

Completeness ⇒ Not complete → why? There is an infinite loop.

Solve → Tree GBFS

Time & Space complexity for TREE & Graph GBFS

T.C. → $O(b^m)$

S.C. → $O(b^m)$

A* SEARCH

$$\underbrace{f(n)}_{\text{Path Cost}} = \underbrace{h(n)}_{\text{heuristic cost}} + \underbrace{g(n)}_{\text{path cost}}$$

→ **Admissibility :-**

$$\underbrace{h(n)}_{\text{heuristic cost}} \leq h_{\text{actual-cost / path cost}}(n)$$

[from goal node]

[It should be applicable for "all" nodes]

EXAMPLE (when heuristic is not admissible) :-

$\boxed{H = 138}$ Changed!

A_{366}

A_{366} | B_{449} C_{447} E_{393}

E_{393} | B_{449} C_{447} G_{413} F_{417}

G_{413} | B_{449} C_{447} F_{417} H_{456}

F_{417} | B_{449} C_{447} ~~B~~ H_{456} I_{450}

C_{447} | B_{449} H_{456} I_{450} $D_{47?}$

B_{449} | B_{455} I_{450} $D_{47?}$

I_{450} | H_{458} D_{473}

Now, its providing sub optimal rate which is

$A \rightarrow E \rightarrow F \rightarrow I \Rightarrow 450$

Is it optimal?

→ h admissible → optimal

h not " → not optimal.

Time Complexity :- $O(b^{\epsilon d})$ ← for heuristic quality

Space Complexity :- $O(b^{\epsilon d})$

CONSISTENCY :-

$h(A) \quad h(B) \quad h(G)$

$A \xrightarrow{c(A,B)} B \xrightarrow{c(B,G)} G$

Here,

$h(A) \leq c(A,B) + h(B)$

$h(B) \leq C(B,G) + h(G)$

Admissible but not consistent :

$h=4 \quad h=1 \quad h=0$

$A \xrightarrow{1} B \xrightarrow{3} G$

$h(A) \leq C(A,B) + h(B)$

$4 \leq 1 + 1$ ✗

(not consistent)

$h(A) \leq h_{ac}(A)$

$4 \leq 4$ ✓

(~~Consistent~~ Admissible)

Completeness :-

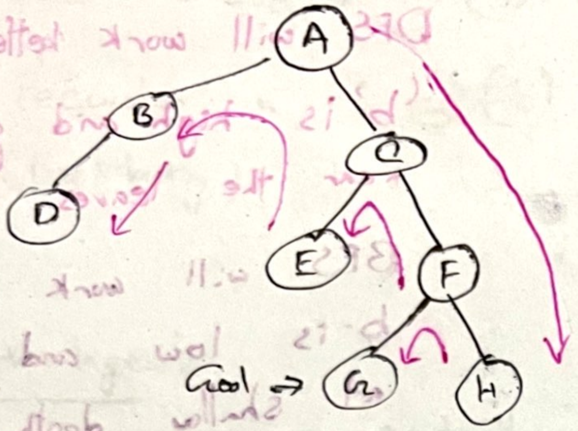
GRAPH SEARCH A* :- h ~~con~~ admissible → complete

TREE " A* :- h both admissible and consistent

↳ doesn't track back ~~the~~ track the visited nodes.

UNINFORMED SEARCH

Depth First Search (DFS):
- → stack based simulation (LIFO)
- → Backtracking (Recursion)



Re (Backtracking based)

A

AC

ACF

ACFH

No nonvisited children left

H ACF

ACFG

G ACF

F AC

~~E~~ ACE

E AC

C A

A ~~B~~

AB

ABD

D AB

B A

A

stack based

A

A BC

C BEF

F BEGH

H BEG

G BE (terminate)

E B~~E~~

B D

D

TIME COMPLEXITY :- $O(b^m)$ $m \rightarrow$ max depth

b: branching factor

DFS will work better when — 'b' is high and goal is near the leaves.

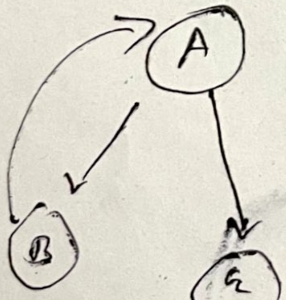
BFS will work better when b is low and goal is at shallow depth.

Completeness: No

(i) if $m \rightarrow \infty$:-

Cannot guarantee to find goal node as it might get stuck to infinite branch.

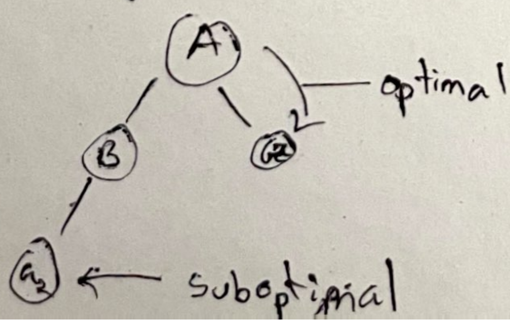
(ii) If there is a ~~tow~~ loop in ~~grah~~ graph and we are applying tree DFS



A
AB
ABA
ABAB. . .

Optimality :- Not optimal

why?



A
AB
A.BG_2
G_2 AB

~~A — A~~
A — B — G_2

SPACE COMPLEXITY :

stack based : $O(b*m)$ → $b+(m-1)(b-1)$

recursion " : $\cancel{O(b)}$ $O(m)$ → more efficient
(Backtracking)

Advantage :-

(i) less space complexity

Disadvantage :-

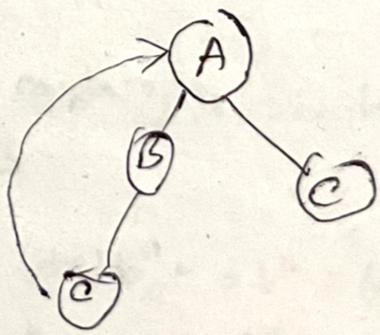
(i) Neither complete nor optimal.

Depth Limited Search (DLS)

→ To address: if the goal is situated in shallow dep

· it cannot guarantee to find goal node as it might get stuck to infinite branch (considering the branch is infinite)

☆ limit set ————→ max depth.



Tree DFS

A L=0

AB L=1

ABC L=2

ABCA L=3

ABCAB L=4 (max = 4)

B ABCA

A ABC

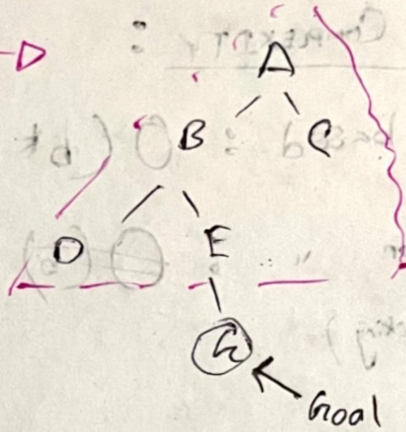
C AB

B A

AG

G A

COMPLETE :- No   why? —▷ :



set L=2

only traversed

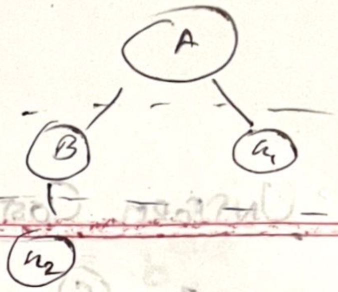
~~count~~ can't visit the goal node

So, it is dependent on threshold.

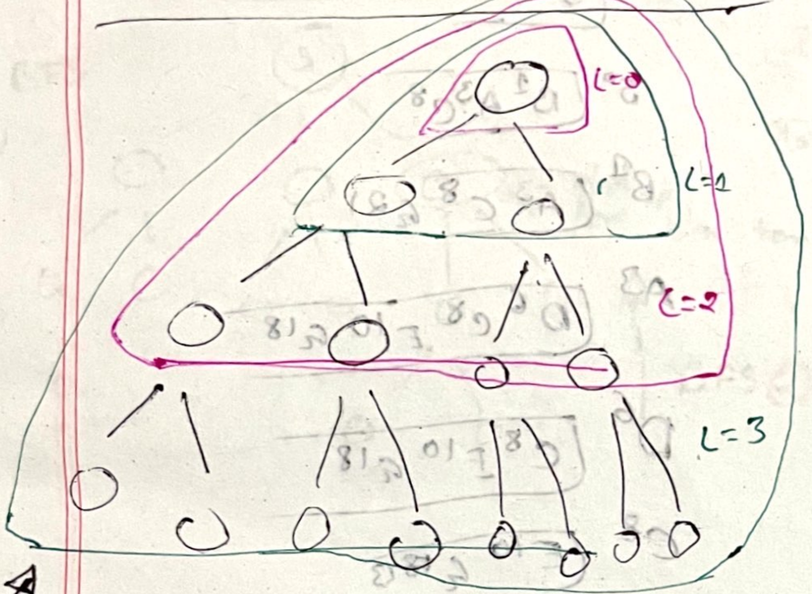
T.C. $\Rightarrow O(b^L)$

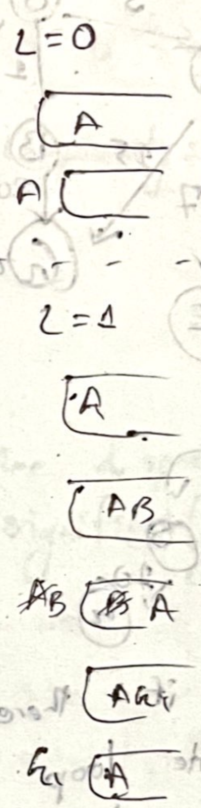
S.C. $\Rightarrow O(b*L)$ stack based

$O(L)$ back tracking.



Iterative Deepening Search :-





somehow we could replicate the BFS.

Space Complexity :- $O(bd)$ [stack]

terminate

$O(d)$ [backtracking]

A → G_1 is the optimal path.

d = depth of the goal node.

Time Complexity :- $O(b^d)$ → time complexity is similar to BFS

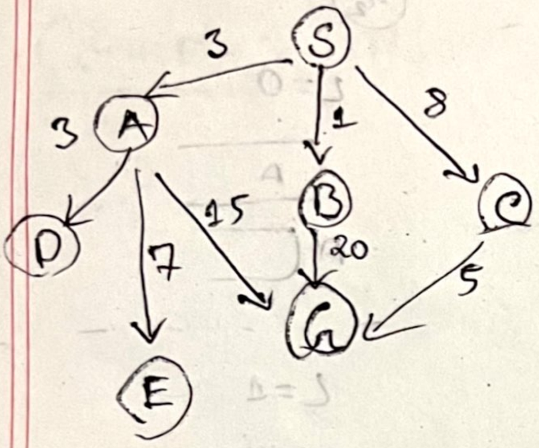
→ $(d+1)\,b^0 + d\,b^1 + (d-1)\,b^2 + \dots + b^d$

For the above tree:

L = 0 → 1
L = 1 → 3
L = 2 → 7
L = 3 → 15

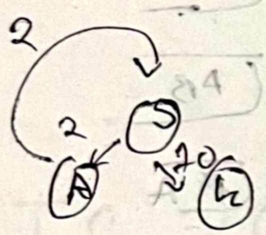
DFS → 15 nodes
IDS → 1+3+7+15
→ 11+15

UNIFORM COST SEARCH (BFS VARIANT FOR UNWEIGHTED GRAPH)



	S^0
S^0	$B^1\ A^3\ C^8$
B^1	$A^3\ C^8\ G^{21}$
A^3	$D^6\ C^8\ E^{10}\ G^{18}$
D^6	$C^8\ E^{10}\ G^{18}$
C^8	$E^{10}\ G^{\cancel{18}\,13}$
E^{10}	G^{13}
G^{13}	

have to terminate while expanding Goal node



even if there is infinite loop Goal node-10 will eventually ~~to~~ be ~~exp~~ expanded.

OPTIMAL: YES

COMPLETE: YES

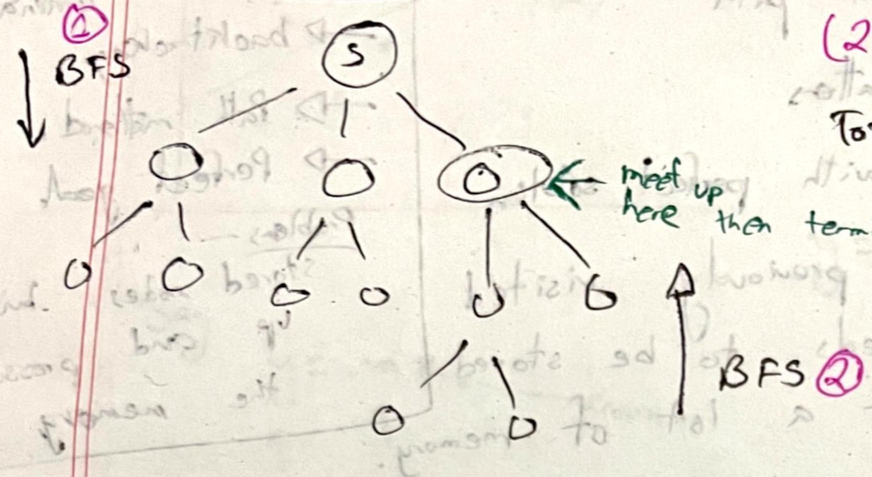
T.C. & S.C: $O(b^d) \rightarrow O\left(b^{\lfloor C^*/\epsilon \rfloor + 1}\right)$

because we need to expand the goal node.

C^* = total path cost from root to goal

ϵ = Avg step size

BIDIRECTIONAL SEARCH



(1) T.C. $\rightarrow b^{d/2}$

(2) T.C. $\rightarrow b^{d/2}$

Total time Complexity $= b^{d/2} + b^{d/2}$

$= 2\,b^{d/2}$

$\simeq b^{d/2}$

Time & space complexity significantly reduce

LOCAL SEARCH

- Not concerned with path
- Solution itself matters
- Not concerned with perfect solution
- No need store previously visited nodes doesn't needs to be stored which saves ~~+~~ a lot of memory.

Informed search criteria –

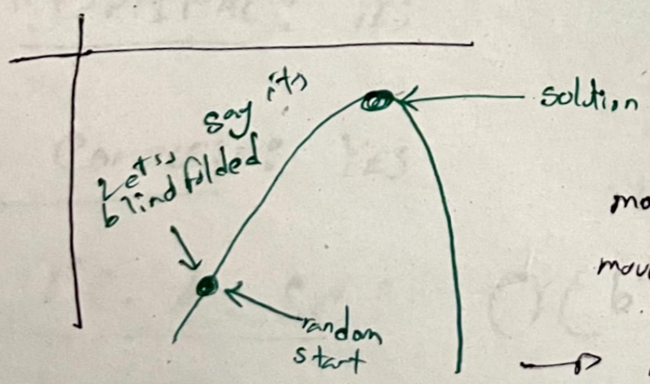
- backtracking
- Path mattered
- Perfect goal

Problems –
stored nodes build up and pressurises the memory

Advantage :-

(i) less memory
(ii) less time
(iii) Applicable for large size graph.

Hill Climb Search :-

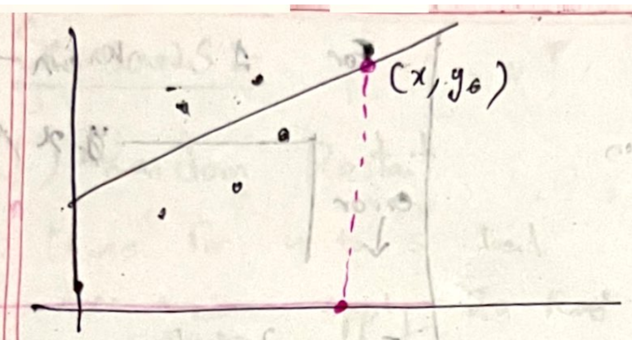


move right → going upward [So it should be preperform
move left → " downward

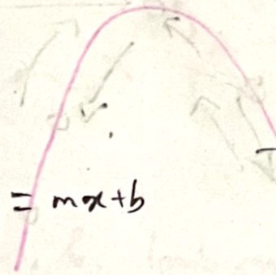
→ move left g. dw + move right g. dw

So, we are at the topmost point.

linear regression



x_1	y_1
x_2	y_2
⋮	⋮
x_6	?



$y = mx + b$

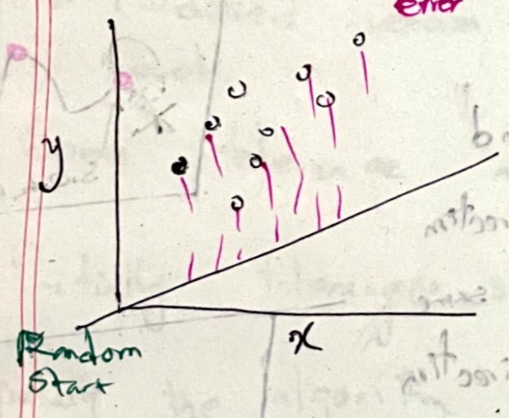
Say

$y = m x$

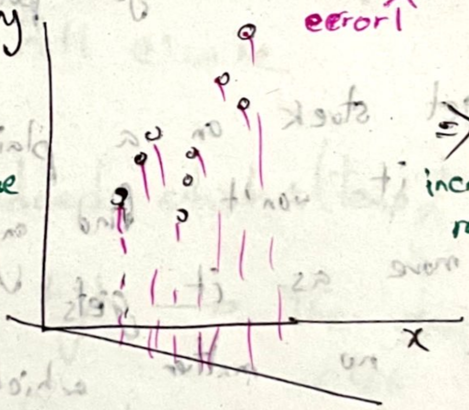
y ← known, m ← unknown, x ← known

unknown → we have to find out the 'm' using local search which will fit the straight line

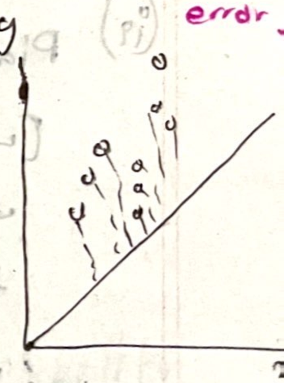
For example:-



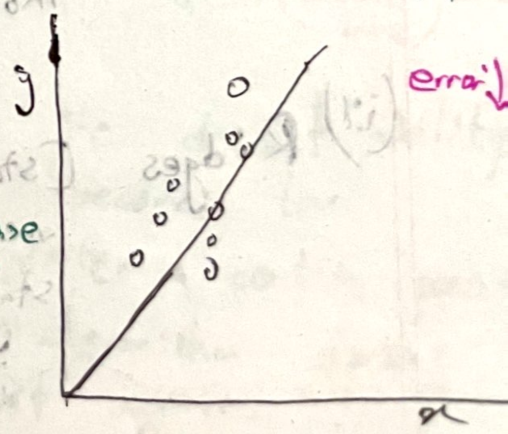
⇒ decrease m



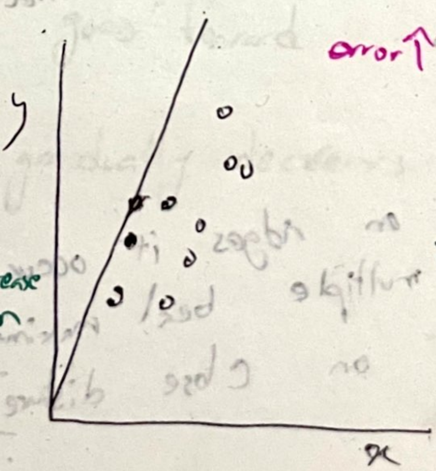
⇒ increase m



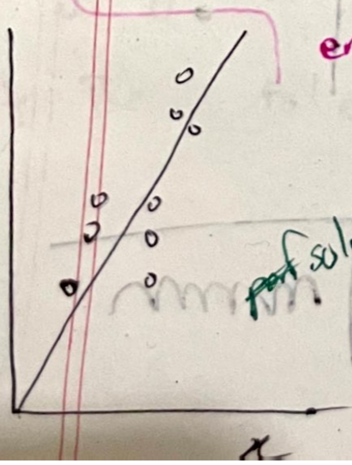
⇓ increase m



⇐ increase m

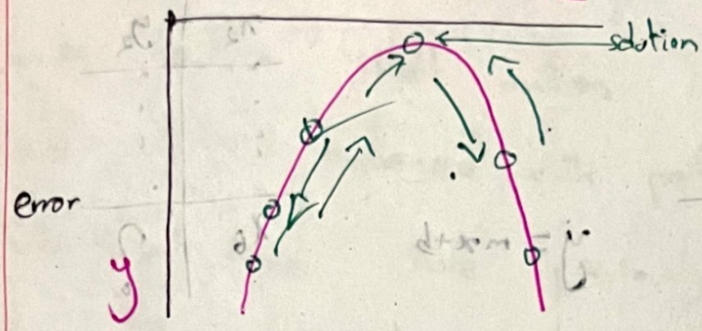


⇐ decrease m

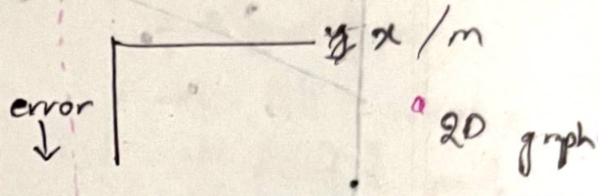


perfect solution!

SUMMARY OF THE ITERATION →

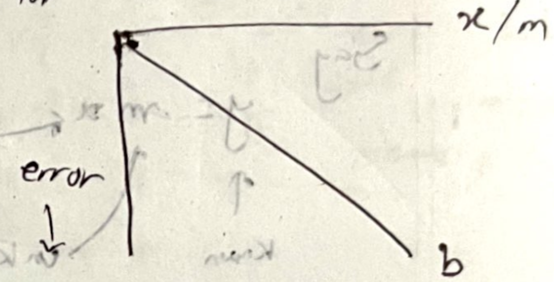


For 1 unknown →



"2D graph"

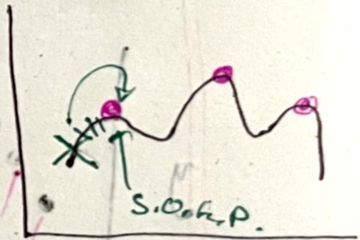
For 2 unknow



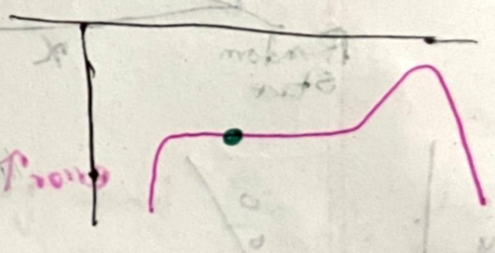
"3D graph"

disadvantage:-

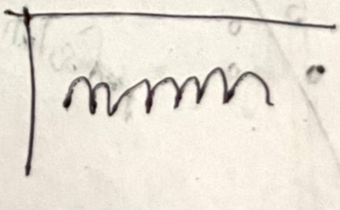
(i) Local maxima problem [Depending on the start point it will return the sub-optimal goal point]



(ii) Plateus
[will get stuck on a plain land where it won't find any direction to move as it gets the same value no matter which direction it moves]



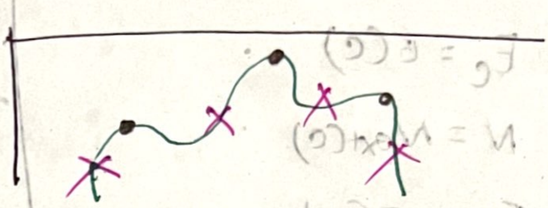
(iii) Ridges [stuck on ridges it occurs when multiple local maxima stays on close distance]



SOLUTIONS :-

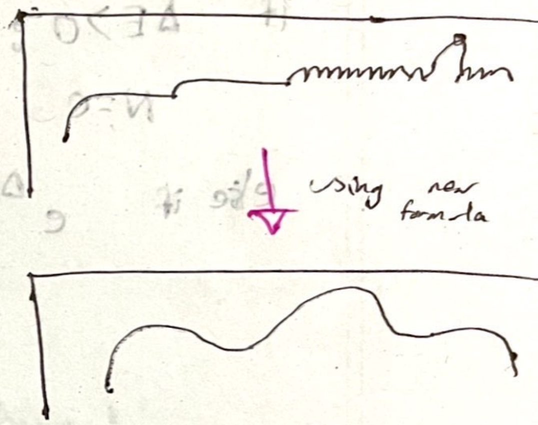
(i) Random Restart

[Like for 4 to 5 local ~~maxa~~ maxima apply 20 Rand restart] → For, plateaus, & local maxima.



(ii) Problem reformulation

[The complex hill problem will become a simple ~~hill~~ hill problem]



(iii) Simulated Annealing.

→ Modified version of Hill climbe search.

Some tolerance are added which is (temp.)

☆ Initially tolerance stays high → The descent ability also high. (accepts error more)

gradually the algorithm goes forward

☆ The tolerance gradually decreases → the descent ability becomes low

after a certain point The algorithm doesn't descent anymore.

Algorithm

$E_C = E(C)$

$N = Next(C)$

$E_N = E(N)$

$\Delta E = E_N - E_C$

if $\Delta E > 0$:

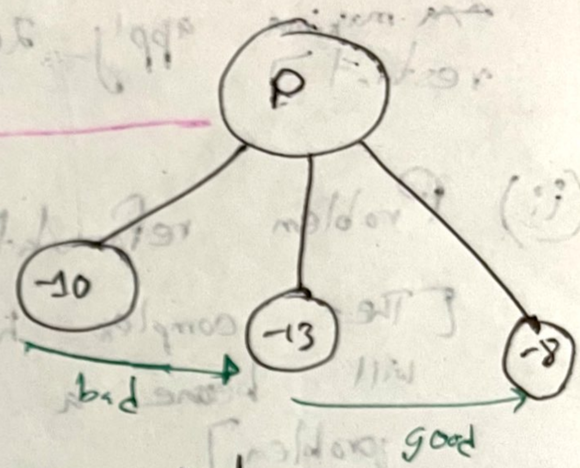
    N=C

else if $e^{\Delta E/T} > rand(0,1)$:

    N=C

TYPES OF HILL CLIMB :-

☆ First choice HC →



It. selects the first child which is better than the parent (Much faster)

Genetic Algorithm

(i) Selection

↓

(ii) Cross over [Offspring creation]

↓

(iii) Mutation.

↓

(iv) Goal Test [fitness calculation] $\xrightarrow{\text{pass}}$ END

For Example:

n-queen problem →

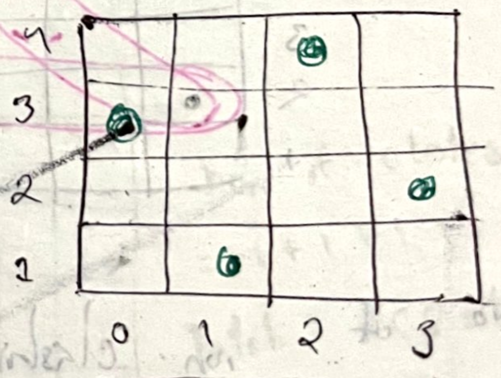
Given,

(i) $n \times n$ board

(ii) n no. of queens → No vertical, horizontal or diagonal clash.

now,

$n = 4$ → 4 queen problem →



3	1	4	2
0	1	2	3

EXAMPLE

8 queen problem ⟶ board size 8×8

no. of queens ⟶ 8

Phase-1 (Selection) :-

randomly generated

Init population

⟶ 2 4 4 1 5 1 2 4 board 1 ⟶ C1

⟶ 3 2 5 4 3 2 1 3 board 2 ⟶ C2

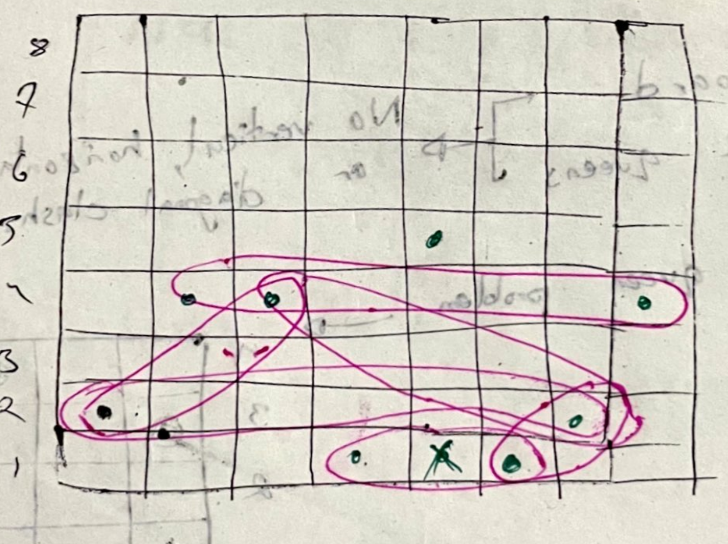
⟶ 3 2 7 5 2 4 1 1 board 3 ⟶ C3

⟶ 2 4 7 4 8 5 5 2 board 4 ⟶ C4

Chromosomes

⟶

board 1



no. of clashes ⇒

= 1+1+1+1+1 + 3C2

= 5+3

= 8 pairs

Total combination ⇒

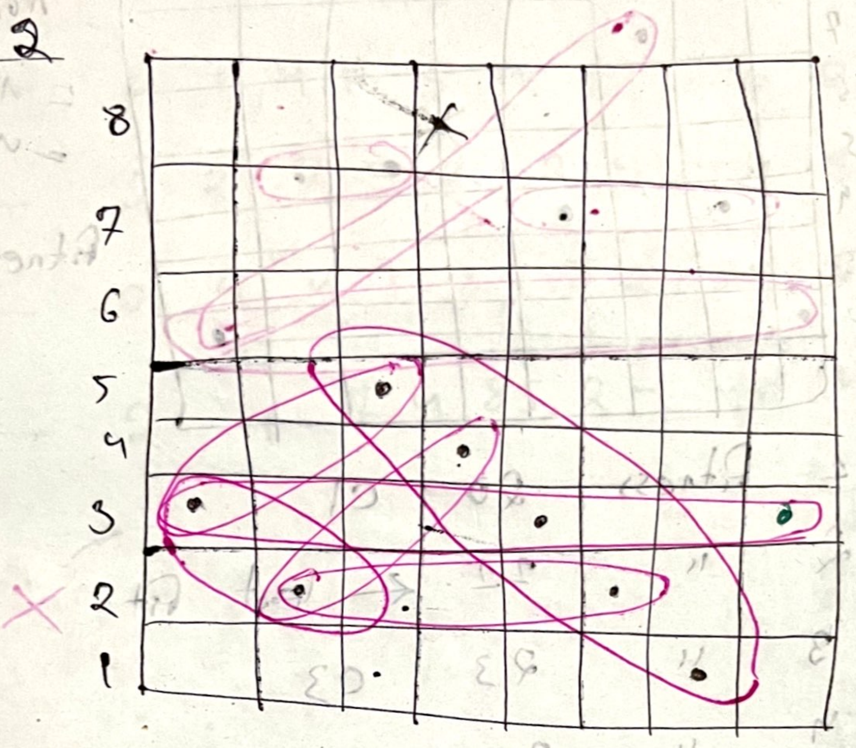
= 8C2

= 28

No. of non clashing pairs :-

= 28 − 8 = 20 pairs ~~28 pairs~~

→ board 2



no. of clashes →

$= 1+1+1+1 + {}^{3}C_2 + {}^{5}C_2$

$= 17$

Total $= {}^{8}C_2$

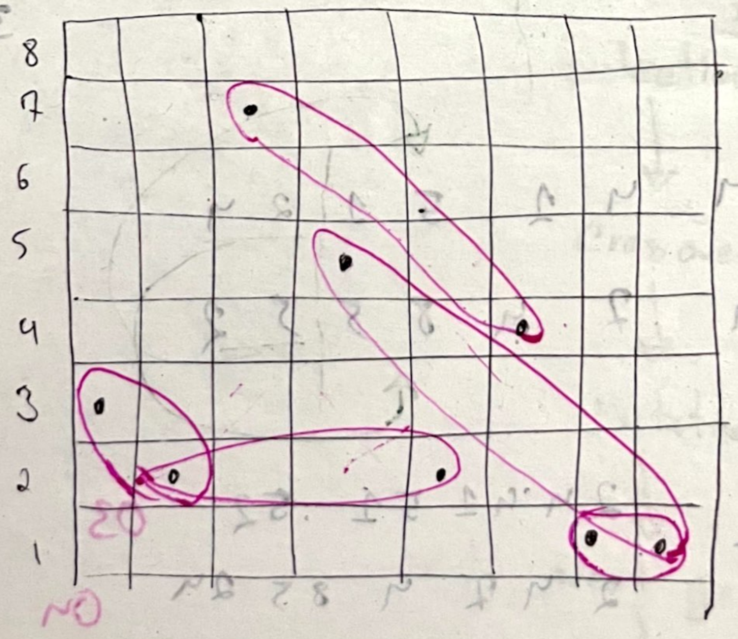
$= 28$

No. of non clashing pairs / fitness :-

$= 28 - 17$

$= 11$

→ board 3



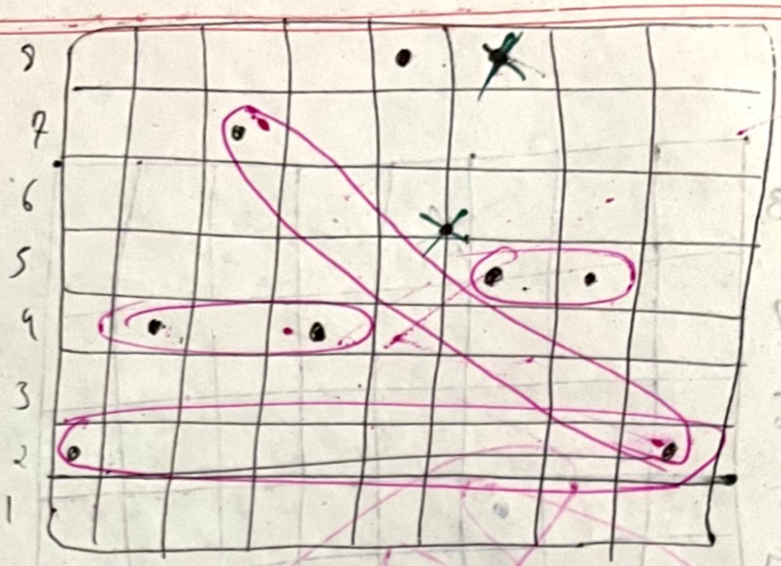
no. of clashes →

$= 1+1+1+1+1 = 5$

Total $= {}^{8}C_2 = 28$

Fitness $= 28 - 5 - 23$

board 4 →



no. of clashes →
= 1+1+1+1
= 4

fitness = 28 − 4 = 24

SELECTION

Now

board 1	fitness	20	C1	
" 2	"	22	← least fit ✗	C2
" 3	"	23	C3	
" 4	"	24	C4	

CROSSOVER

point of cross-over

C1: 2 4 4 1 5 | 1 2 4 → 2 4 4 1 5 4 1 1 C1

C3: 3 2 7 5 2 | 4 1 1 crossover 3 2 7 5 2 2 1 4 O2

C1: 2 4 4 1 5 1 | 2 4

C4: 2 4 7 4 8 5 | 5 2

Cross over

2 4 4 1 5 1 5 2 O3

2 4 7 4 8 5 2 4 O4

MUTATION :-

01 — 2 4 4 1 5 4 1 1 → 2 8 4 ~~4~~ 1 5 4 1 1

02 — 3 2 7 5 2 1 2 4 → 3 2 7 5 2 1 2 4

03 — 2 4 4 1 5 1 5 2 → 2 4 4 1 5 3 5 2

04 — 2 4 7 4 8 5 2 4 → 4 4 7 3 8 5 2 4

↓

NEXT GEN POPULATION

↓

fitness Calculation

↓

GOAL TEST

if → no. of clash = 0 → end

else → selection

↓

Cross over

↓

mutation

↓

Next gen population

→ Repeat the process

A* SEARCH Imp. POINTS →

$h(a) \leq C(a,b) + h(b)$

* Tree version of A* optimal if $h(n)$ is admissible
* Graph " " " " " " consistent.

Reduce memory → Iterative deepening

LOCAL

FINAL

PROBABILITY THEORY

Events / Variables :-

Event A → A, $\neg A$

Event B → B, $\neg B$

$0 \le P(A) \le 1$

var weather = ⟨ Sunny, cloudy, snowy, rainy ⟩

= ⟨ 0.72, 0.1, 0.08, 0.1 ⟩

$P(\text{Sunny}) = 0.72$

$P(\text{cloudy}) = 0.1$

☆ $P(A \cap B) = P(A, B) = P(A \text{ and } B)$ ↔ Joint Probability

☆ $P(A \vee B) = P(A \text{ or } B)$

→ $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

JOINT PROBABILITY

A → A, $\neg A$

B → B, $\neg B$

1 ←
- $P(A \cap B)$
- $P(\neg A \cap B)$
- $P(A \cap \neg B)$
- $P(\neg A \cap \neg B)$

→ JOINT PROBABILITY DISTRIBUTION

	A	¬A
B	$P(A \cap B)$	$P(\neg A \cap B)$
¬B	$P(A \cap \neg B)$	$P(\neg A \cap \neg B)$

JPB Table

Marginal Probability

In a joint probability distribution there are 5 variables.
If we form a joint probability ~~joint~~ distribution with that 5 variable ~~how~~ then how many value will there be in the table →

Marginal Probability → $P(A) = P(A \wedge B) + P(A \wedge \neg B)$.

$P(\text{burglary}) = P(\text{burglary} \wedge \text{Alarm}) + P(\text{burglary} \wedge \neg\text{Alarm})$

	Alarm	¬Alarm
Burglary	.09	.01
¬Burglary	.1	.8

CONDITIONAL PROBABILITY:

~~P(alarm | Burgla~~

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} \quad / P(A \cap B)$$

$$P(\text{alarm} | \text{Burglary}) = \frac{P(\text{alarm} \wedge \text{burg})}{P(\text{burglary})}$$

$$= \frac{0.09}{0.09 + 0.01}$$

$$= 0.9$$

3 variables :- (OPBT)

	t		¬t	
	x	¬x	x	¬x
C	~~0.01~~ 0.108	0.012	0.072	0.008
¬C	0.016	0.064	0.144	0.576

$P(t \wedge c) = 0.108 + 0.012$

$= 0.12$

$P(\neg c \wedge x) = 0.016 + 0.144$

$= 0.16$

$P(\neg x) = 0.012 + 0.064 + 0.008 + 0.576$

↑ Marginal Probability

$$P(t | \neg x) = \frac{P(t \wedge \neg x)}{P(\neg x)}$$

$$= \frac{0.012 + 0.064}{0.012 + 0.064 + .008 + 0.576}$$

$2^5 = 32$

$$P(t \mid c \wedge catch) = \frac{P(toothache \wedge cavity \wedge catch)}{P(cavity \wedge catch)}$$

INDEPENDENCE :-

(i) Full Independence : 'A' does not effect event "B", A & B are independent. $P(A \wedge B) = P(A) * P(B)$

(ii) Conditional Independence :-

→ A, B are independent | Given, → C

☆ $P(A \wedge B \mid C) = P(A \mid C)\, P(B \mid C)$

↓

$$\frac{P(A \wedge B \wedge C)}{P(C)} = \cancel{A(}\, P(A \mid C)\, P(B \mid C)$$

☆ $P(A \wedge B \wedge C) = P(A \mid C)\, P(B \mid C)\, P(C)$

EXAMPLE :-

Is smart conditionally independent of prepared, given study

$P(smart \wedge prepared \mid study) = P(smart \mid study)\, P(prepared \mid study)$

If left h.s. = Right h.s. then, Sm is cond. independent of prep given study.

NAIVE BAYES

Bayes Theorem $\rightarrow$ $P(A|B) = \dfrac{P(B|A)\,P(A)}{P(B)}$

Example:-

$P(HIV) = 0.008$ $\quad$ $P(\neg HIV) = 0.992$

$P(T|HIV) = 0.95$ $\quad$ $P(\bar{T}|HIV) = 1 - 0.95 = \cancel{0.5}\ 0.05$

$(P(T|\neg HIV) = 0.95)$ $\quad$ $P(T|\neg HIV) = 1 - 0.95 = 0.05$

$$P(HIV|T) = \frac{P(T|H)\,\cancel{P(PHI)}\,P(HIV)}{\cancel{P(HIV)}\ P(T)}$$

$$= \frac{P(T|H)\ P(HIV)}{\underline{P(T|HIV) * P(HIV) + P(T|\neg HIV) * P(\neg HIV)}}$$

No need while comparison

$= 0.95 \times 0.008$

$= 0.0076$

$P(\neg HIV|T) = P(T|\neg HIV) \times P(\neg HIV)$

$= 0.05 \times 0.992$

$= 0.0496$

Find the probability ~~P~~ if a person is tested and found HIV.

$$P(HIV|T) = \frac{P(T|HIV)\,P(HIV)}{P(T)}$$

$$= \frac{P(T|HIV)\,P(HIV)}{P(T|HIV)\,P(HIV) + P(T|\neg HIV)\,P(\neg HIV)}$$

$$= \frac{0.95 \times 0.008}{0.95 \times 0.008 + 0.05 \times 0.992}$$

$$= 0.1329$$

N.B. – 2

Temp	Play Tennis
w / 70	yes
c / 32	no
w / 65	no
w / 75	yes
c / 30	no
w / 75	yes
w / 72	no

Threshold →

50 < warm °F   50 > cold

If there is a numeric value on the table then first change it to categorical.

Here,

$$P(\text{Play Tennis} \mid \text{warm}) = \frac{P(\text{warm} \mid \text{Play Tennis})\; P(\text{Play tennis})}{P(\text{warm})}$$

$$= \frac{\frac{3}{3} * \frac{3}{7}}{\frac{5}{7}}$$

$$= \frac{3}{5}$$

$$= 0.6$$

⇒ Calculate the player will play tennis or not given the weather is warm.

Here,

$$P(\text{Play Tennis} \mid \text{warm}) = \frac{P(\text{warm} \mid PT)\; P(PT)}{P(\text{warm})}$$

$$= P(\text{warm} \mid PT)\; P(PT)$$

$$= 3/3 * 3/7$$

$$P(\neg \text{Play tennis} \mid \text{warm}) = \frac{P(\text{warm} \mid \neg PT)\; P(\neg PT)}{P(\text{warm})}$$

$$= P(\text{warm} \mid \neg PT)\; P(\neg PT)$$

$$= 2/4 * 4/7$$

$$= 2/7$$

→ NAIVE BAYES CLASSIFFIER is based on the Bayes theorem with independence assumptions between ~~part~~ predictors.

EXAMPLE

→ Calculate if the player will play tennis given outlook is sunny, temp is cool, humidity is high, wind is strong.

[$P(B \wedge C \wedge D \mid A)$ which difficult to compute, can instead be substituted by a naive approximation that assumes for a given class, the values of the attributes to be independent

So, $P(B \wedge C \wedge D \mid A)$ replaced by → $P(B \mid A) \times P(C \mid A) \times P(D \mid A)$

→

$$P(x_1, x_2, x_3, \ldots, x_n \mid c) = P(x_1 \mid c)\, P(x_2 \mid c)\, P(x_3 \mid c) \ldots P(x_n \mid c)$$

why → If in the multiplication if in row is '0', then the entire calculation will become '0'. So, there will be zero probability problem.

→ To ~~slo~~ solve this problem we will introduce the concept of "

Q. If they says construct the $\boxed{\text{learning phase.}}$

Soln →

Outlook	Yes	No
Sunny	2/9	3/5
Rain	3/9	2/5
Overcast	4/9	0/5

temp	yes	no
Hot	2/9	2/5
Mild	4/9	1/5
Cool	3/9	2/5

~~Hum~~

Hum	yes	no
High	3/9	4/5
Normal	6/9	1/5

Wind	yes	no
weak	6/9	5/5
strong	3/9	0/5

$P(\text{Play}_{T.T.}) = \frac{9}{14}$ $P(\neg \text{Play } T.) = \frac{5}{14}$

Here,

while comparison we do not need the denominator

$P(\text{Sunny} \wedge \text{cool} \wedge \text{high} \wedge \text{strong} \mid \text{play}) = P(\text{sunny} \mid \text{play})\, P(\text{cool} \mid \text{play})\, P(\text{high} \mid \text{play})$
$P(\text{strong} \mid \text{play})\ P(\text{play})$

$= 2/9 * 3/9 * 3/9 * 3/9 * 9/14$

$= 0.0053$

$P(\text{Sunny} \wedge \text{cool} \wedge \text{high} \wedge \text{strong} \mid \neg \text{Play}) = P(\text{sunny} \mid \neg \text{play})\, P(\text{cool} \mid \neg \text{play})$
$P(\text{high} \mid \neg \text{play})\ P(\text{strong} \mid \neg \text{play})$
$P(\neg \text{play})$

$$= \frac{3}{5} * \frac{1}{5} * \frac{4}{5} * \frac{3}{5} * \frac{5}{14}$$

$$= 0.0206.$$

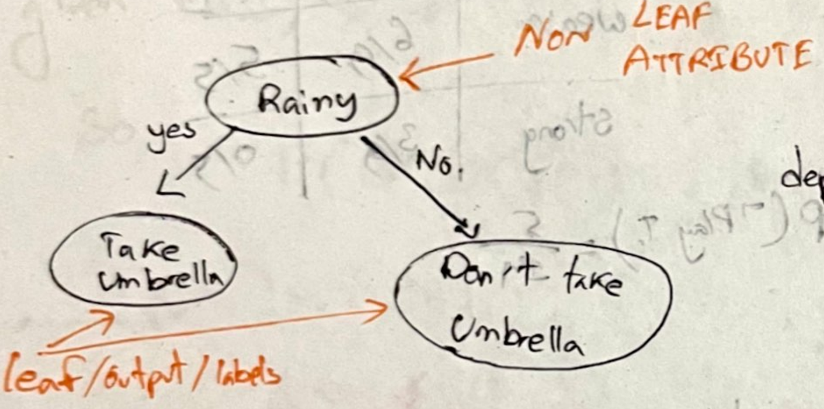
prev page

$$p(play \mid sunny \wedge \cancel{play}\ cool \wedge high \wedge strong) = \frac{p(sunny \wedge cool \wedge high \wedge strong \mid play)\, p(play)}{\cancel{p(sunny \wedge cool \wedge high \wedge strong)}}$$

$$p(\neg play \mid sunny \wedge cool \wedge high \wedge strong) = \frac{p(sunny \wedge cool \wedge high \wedge strong \mid \neg play)\, p(\neg play)}{\cancel{p(sunny \wedge cool \wedge high \wedge strong)}}$$

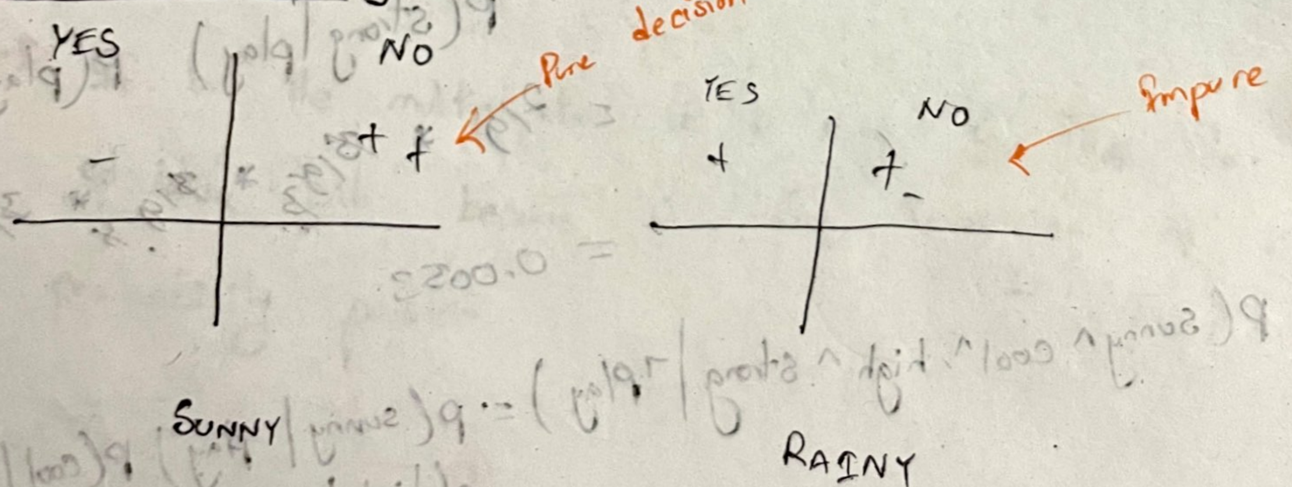
prev page

Decision Tree



depth = 1

Information Purity



→ Sunny is more ~~the~~ pure than rainy → sunny should have more priority than rainy.

→ sunny has more information

To calculate ~~impor~~ information purity we use entropy:

ENTROPY value high $\longrightarrow$ less pure

" " low $\longrightarrow$ more pure

No need while Comparing $\longrightarrow$

$\text{Entropy} = \sum_{i=1}^{n} (-P_i \log P_i)$ i ⇒ number of labels

Example -

Sunny	Rainy	Decision
Y	N	Don't take
No	Y	Take
N	N	Take

$\text{Entropy}(\text{Sunny} = \text{Yes}) = -P(DT)\log_2 P(DT) - P(TU)\log_2 P(TU)$

$= -\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1}$

$= -\log_2 1$

$= 0$

$\text{Entropy}(\text{Sunny} = \text{No}) = -P(DT)\log_2 P(DT) - P(TU)\log_2 P(TU)$

$= -\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}$

$= 0$

$\text{Entropy}(\text{Rainy} = \text{No}) = -P(DT)\log_2 P(DT) - P(TU)\log_2 P(TU)$

$= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}$

$= 1$

$\boxed{\log_2 \text{ base}}$

DECISION TREE [P-2]

~~E(Sunny == Yes)~~ Entropy → For every category there will be ~~certain~~ certain entropy.

For example there are 3 category

PRACTICE PROBLEM

$$E(\text{Decision}) = -P(\text{yes}) \log_2 P(\text{yes}) - P(\text{no}) \log_2 P(\text{no})$$

$$= -\frac{9}{14} \times \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \log_2\left(\frac{5}{14}\right)$$

$$= 0.940$$

Now,

$IG(\text{Outlook}) \rightarrow$

$-P(\text{no}|\text{sunny}) \log_2 (\text{no}|\text{sunny}) - P(\text{yes}|\text{sunny}) \log_2 (\text{yes}|\text{sunny})$

$$E(\text{Outlook} == \text{sunny}) = -\frac{3}{5} \log_2 \frac{3}{5} - \cancel{\log} \frac{2}{5} \log_2 \frac{2}{5}$$

$$= 0.971$$

$$E(\text{Outlook} == \text{rainy}) = -\frac{3}{5} \log_2 \frac{3}{5} - \log_2 \frac{2}{5} \times \frac{2}{5}$$

Then a no. of rows of rainy.

$$= 0.971.$$

Highest IG value is root node.

$$P(\text{outlook}==\text{overcast}) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}$$

$$= 0$$

$$IG(\text{Outlook}) = E(\text{Deciesion}) - E(O==\text{Sunny})\,P(\text{sunny}) - E(O==\text{rainy})\,P(\text{rainy})$$

$$- E(O==\text{Overcast})\,P(\text{overcast})$$

$$= 0.94 - \frac{5}{14}\times 0.971 - \frac{5}{14}\times 0.971$$

$$- \frac{4}{14}\times 0$$

$$= 0.246$$

Again,

$IG(\text{Humidity}) \rightarrow$

$$E(\text{Humidity}==\text{high}) = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7}$$

$$= 0.985$$

$$E(\text{hum}==\text{Normal}) = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7}$$

$$= 0.591$$

Scanned with CamScanner

$IG(\text{Humidity}) = E(\text{decision}) - P(\text{high}) \times E(\text{hum} = \text{high})$

$- P(\text{hum} == \text{Normal})\ E(\text{hum} == \text{normal})$

$= 0.94 - \frac{7}{14} \times 0.985 - \frac{7}{14} \times 0.591$

$= 0.152$

$IG(\text{wind}) \longrightarrow$

$E(\text{wind} == \text{weak}) = -\frac{6}{8} \log_2 \frac{6}{8} - \log_2 \frac{2}{8} \times \frac{2}{8}$

$= 0.811$

$E(\text{wind} == \text{strong}) = -\frac{3}{6} \log \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6}$

$= 1$

$IG(\text{wind}) = 0.94 - \frac{8}{14} \times 0.811 - \frac{6}{14} \times 1$

$= 0.048$

Ig(Temp) $\rightarrow$

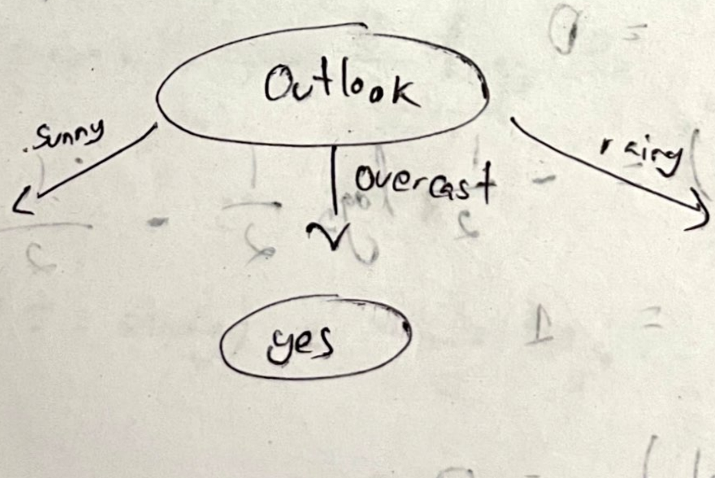
$$E(\text{Temp}==\text{Hot}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}$$
$$= 1$$

$$E(\text{temp}==\text{Mild}) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6}$$
$$= 0.918$$

$$E(\text{temp}==\text{cool}) = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}$$
$$= 0.81$$

$$Ig(\text{Temp}) = \cancel{0.94 - 0.\ \frac{4}{14}\times 1}$$
$$= 0.94 - \frac{4}{14}\times 1 - \frac{6}{14}\times 0.918 - \frac{4}{14}\times 0.81$$
$$= 0.029$$

Ig(Outlook) is the highest.



Outlook	Temp	Hum	Wind	D
~~Rain~~ Sunny	Hot	high	weak	N
~~Rain~~ sunny	~~Mild~~ Hot	high	Strong	N
Sunny	~~Cool~~ mild	high	w	N
Sunny	Cool	Normal	W	Y
Sunny	Mild	Normal	Str	Y

$$E(\text{decision}) = -\frac{2}{5} \times \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5}$$

$$= 0.971$$

$$E(\text{Temp} == \text{Hot}) = -\frac{2}{2} \log_2 \frac{2}{2} - \frac{0}{2} \log_2 \frac{0}{2}$$

$$= 0$$

$$E(T == \text{Mild}) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2}$$

$$= 1$$

$$E(T == \text{Cool}) = 0$$

$IG(\text{Temp}) = 0.971 - 0 - \frac{2}{5} \times 1 - 0$

$= 0.571$

$IG(\text{Humid}) \longrightarrow$

$E(\text{Hum} == \text{high}) = -\frac{3}{3} \log_2 \frac{3}{3} - \frac{0}{3} \log_2 \frac{0}{3}$

$= 0$

$E(\text{hum} == \text{Normal}) = 0$

$IG(\underset{\text{Sunny}}{\text{Hum}}) = 0.971 - 0 - 0$

$= 0.971$

$\cancel{IG} \; \cancel{IG(\text{wind})} = \cancel{\quad}$

$IG(\text{wind}) \longrightarrow$

$E(\text{wind} == \text{weak}) = -\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3}$

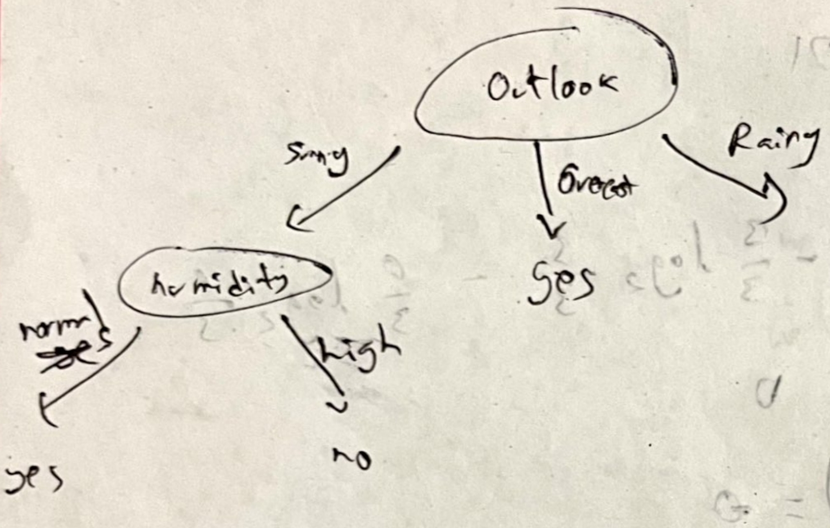
$= 0.918$

$E(\text{wind} == \text{strong}) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2}$

$= 1$

$IG(\text{windSunny}) = 0.971 - \frac{3}{5} \times 0.918 - \frac{2}{5} \times 1 = 0.02$

~~Highe~~ SI(Humidity|Sunny) is the highest,



Outlook	Temp	Hum	wind	Decision
Rain	Mild	High	W	yes
Rain	Cool	Normal	W	yes
Rain	Cool	Normal	Strong	No
Rain	Mild	Normal	W	yes
Rain	Mild	High	Strong	No

$$E(\text{decision}|\text{Rain}) = -\frac{3}{5}\log_2\frac{3}{5} - \left(\frac{2}{5}\right)\log_2\frac{2}{5}$$

$$= 0.971$$

$IG(Temp \mid Rain)$:-

$$E(Temp == Mild) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}$$

$$= 0.918$$

$$E(Temp == Cool) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}$$

$$= \cancel{0.918}\ 1$$

$$IG(T \mid R) = 0.971 - \frac{3}{5}\times 0.918 - \frac{2}{5}\times 1$$

$$= \cancel{0.033}\ 0.0202$$

$I(H \mid O == R)$:⟶

$$E(hum == High) = -\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}$$

$$= 0$$

$$E(hum == Normal) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}$$

$$= 0.918$$

$$IG(H \mid O = R) = 0.971 - 1 \times \frac{2}{5} - 0.918 \times \frac{3}{5}$$

$$= 0.0202$$

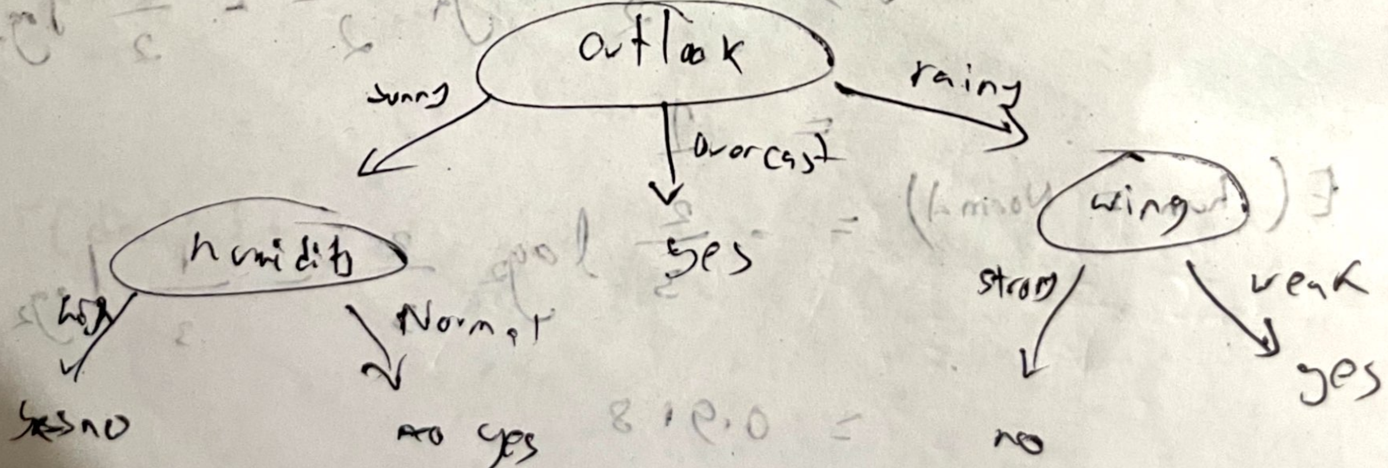
$IG(\text{wind} \mid \cancel{O = we}\ \text{wind})$

$$E(\text{wind} = \text{weak}) = -\frac{3}{3}\log_2 \frac{3}{3} - 0$$

$$= 0$$

$$E(\text{wind} = \text{strong}) = 0$$

Information gain of wind is the highest



Exam score

>80 → 8 P, 2 F

$50-80$ → 5 F 5 F

$50>$ → 1 P 9 F

$$E(\text{decision}) = 0.918$$

(a) $IG(\text{Exam scores}) = 0.918 - \frac{10}{30}\times 0.72 - \frac{10}{30}\times 1 - \frac{10}{30}\times 0.46$

$= 0.391$

$E(\text{ES= Above } 80\%) = -\frac{8}{10}\log_2\frac{8}{10} - \frac{2}{10}\log_2\frac{2}{10}$

$= 0.72$

$E(\text{ES= Between } 50\% \text{ and } 80\%) = -\frac{5}{10}\log_2\frac{5}{10} - \frac{5}{10}\log_2\frac{5}{10}$

$= 1$

$E(\text{Below } 50\%) = -\frac{1}{10}\log_2\frac{1}{10} - \frac{9}{10}\log_2\frac{9}{10}$

$= 0.46$

(b) E (LA = HA | Below 50%

$\cancel{E(D \mid LA = HA)} = -\frac{1}{3}\times\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}$

(c) $E(\cancel{D} \mid \cancel{LA = MA}) = -\frac{0}{7}\times\log_2\frac{0}{7} - \frac{7}{7}\log_2\frac{7}{7}$ LA=HA below 50%

$= 0$

LINEAR REGRESSION :-

It makes the prediction by plotting a straight line that best fits to the data set and "the way to plot this straight line" is by using something called "cost function" which is the measurement of how well the ~~meas~~ predictions are doing.

▸ The cost function or the cost value represents the error between the actual predictions and the actual value

$$y = \theta_0 + \theta_1 \ldots\ldots + \theta_n x_n$$

So, in order for the straight line to fit the ~~the~~ plot the cost function or THE ERROR ~~RER~~ REPRESENTATION should be minimum.

AND to minimize this we need to use "GRADIENT DESCENT ALGORITHM."

\# Cost function / Loss function / MSE : $\frac{1}{N} \sum_{i=1}^{N} (y_i - y_i')^2$

Mean squared error.

⇒ The gradient descent algorithm is the algorithm we use to minimize the cost function value so that our straight line fits best to the dataset.

~~STEPS~~

STEPS TO LINEAR REGRESSION :-

(i) Define the linear function → $y = mx + c$ / will be used as y_{pred} !

(ii) Define the loss function →

$$SSR = \sum_{i=1}^{n} (y - y_{pred})^2$$

(iii)

$$SSR = (1.4 - (m \times 0.5 + c))^2 + (1.9 - (m \times 2.3 + c))^2 + (3.2 - (m \times 2.9 + c))^2$$

used points

x	y
0.5	1.4
2.3	1.9
2.9	3.2

(iii) Find the derivatives of the loss function with respect to the unknown parameter :-

$$\frac{d(SSR)}{dm} = 2(1.4 - 0.5m - c)(-0.5) + 2(1.9 - 2.3m - c)(-2.3) + 2(3.2 - 2.9m - c)(-2.9)$$

(4) Assume values for the unknown parameters and start the algorithm.

Here, $x = 3.1$, $y = ?$ $\Rightarrow$ $y = m\underset{\downarrow \text{find}}{}x^{3.1} + \underset{\downarrow \text{find}}{c}$ ⊳ No need

we get,

$$\frac{d(SSR)}{dm} = 2(1.4 - 0.5m - c)(-0.5) + 2(1.9 - 2.3m - c)(-2.3) + 2(3.2 - 2.9m - c)(-2.9) \quad \text{——— (i)}$$

⊳ Assume :-

ITERATION ⊳ 1

$m = 1$, $c = 0$, $\alpha = 0.01$ (learning rate) in (i)

$$\frac{d(SSR)}{dm} = 2(1.4 - 0.5 - 0)\times(-0.5) + 2(1.9 - 2.3 - 0)(-2.3) + 2(3.2 - 2.9 - 0)(-2.9)$$

$$\frac{d(SSR)}{dm} = -0.8$$

$$\text{step\_size}(m) = (-0.8)\times\alpha \qquad \Bigg| \quad \underset{\substack{\uparrow \\ \text{slope}}}{\frac{d(SSR)}{dm}} \times \underset{\substack{\uparrow \\ \text{learning rate}}}{\alpha}$$

$$= -0.8 \times 0.01$$

$$= -0.008$$

$$m_{new} = m_{old} - \alpha \times \frac{d(SSR)}{dm}$$

$$\Rightarrow m_{new} = m_{old} - \text{step size}(m)$$

$$= 1 - (-0.008)$$

$$\therefore m_{new} = 1.008$$

Again,

$$\frac{d(SSR)}{dc} = 2(1.4 - 0.5m - c)(-1) + 2(1.9 - 2.3m - c)(-1) + 2(3.2 - 2.9m - c)(-1)$$

$$\frac{d(SSR)}{dc} = -1.6 \quad \text{(substituting the assumed value)}$$

$$\text{step\_size}(c) = -1.6 \times 0.01$$

$$= -0.016$$

$$C_{new} = C_{old} - \text{step\_size}(c)$$

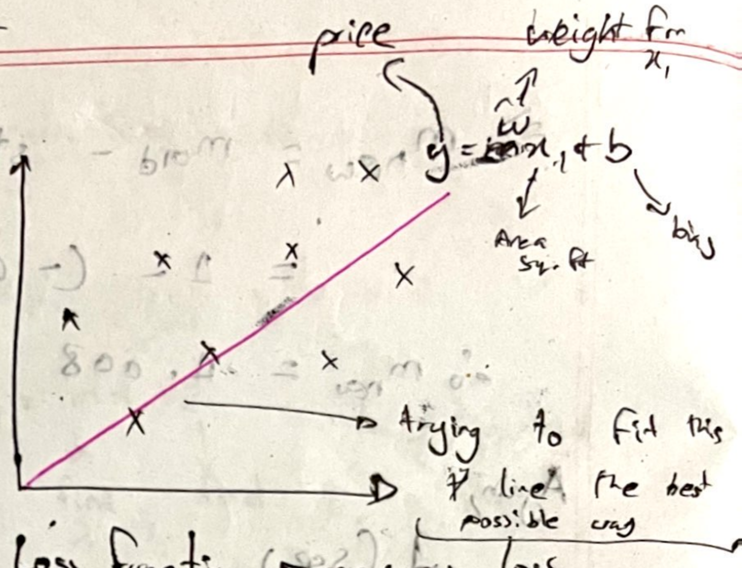
$$= 0 - (-0.016)$$

$$= 0.016$$

~~LO~~ LOGISTIC REGRESSION

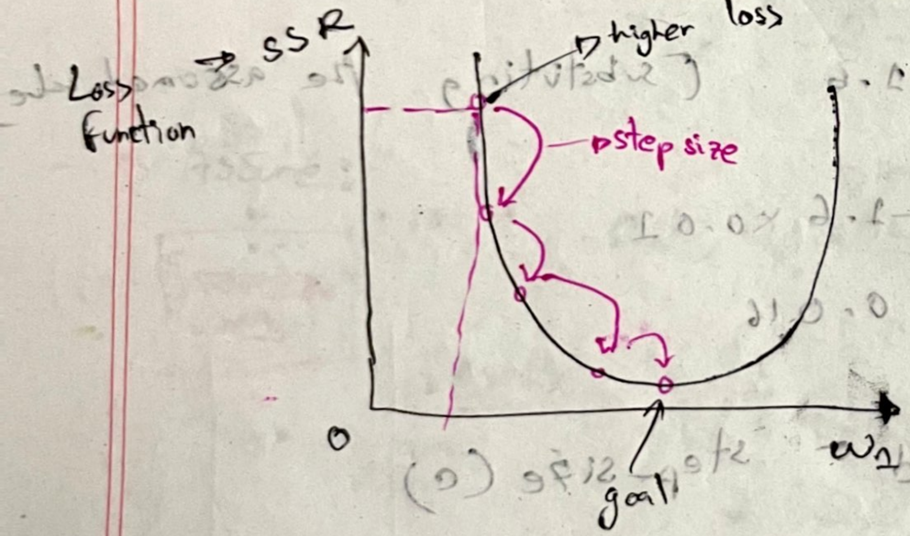
Linear regression flashback →

Area sq. ft	Price
–	–
–	–
–	–
–	



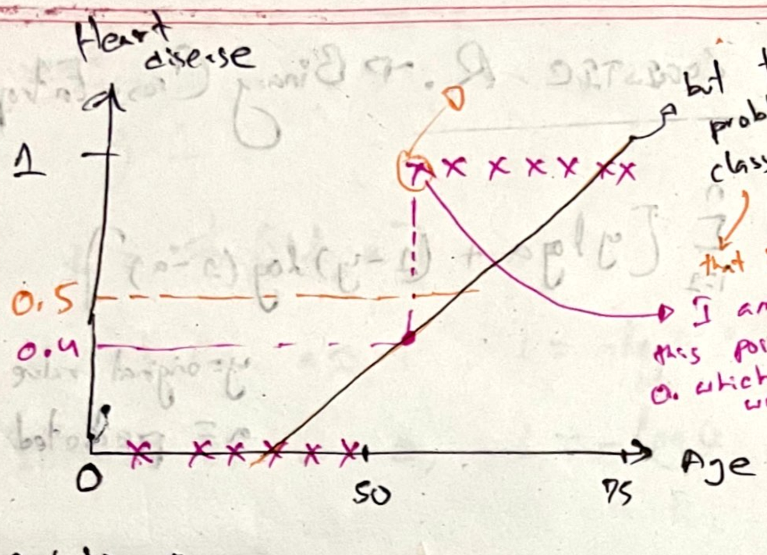
Loss function → measures loss

GRADIENT DESCENT →



$$w_{1(\text{new})} = w_{1(\text{old})} - \underbrace{\frac{d(SSR)}{dw_1}}_{\text{tangent}}$$

→ When the predicted ~~value~~ column is categorical we use Logistic ~~regree~~ regression to



but there is a problem, if I use this str. line to classify!

that is I am labelling this point as 0, which is wrong

No

0 → No heart disease

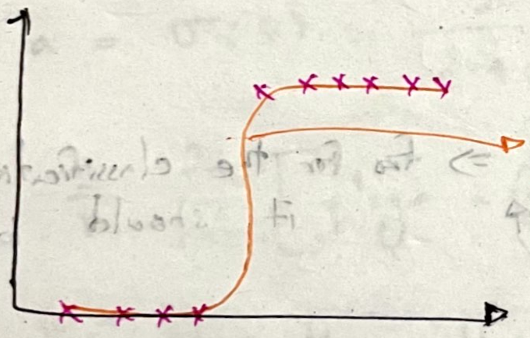
1 → heart disease

if $y \geq 0.5 \Rightarrow 1$ (H. D.)

elif $y < 0.5 \Rightarrow 0$ (N. H. D.)

Solution :-

Since, it's a non linear dataset we will try to fit a non-linear equation curve. So, fitting a sigmoid curve.



$y = \frac{1}{1+e^{-x}}$

$$Z = \omega_1 \cdot x_1 + b$$

Age

$$a \text{ or } \hat{y} = \sigma(z) = \frac{1}{1+e^{-z}}$$

predicted y

$$\therefore \; y_{pred} / \hat{y} / a = \frac{1}{1+e^{-Z}} = \frac{1}{1+e^{-\omega_1 x_1 + b}}$$

Why we are not taking MSE/SSR?

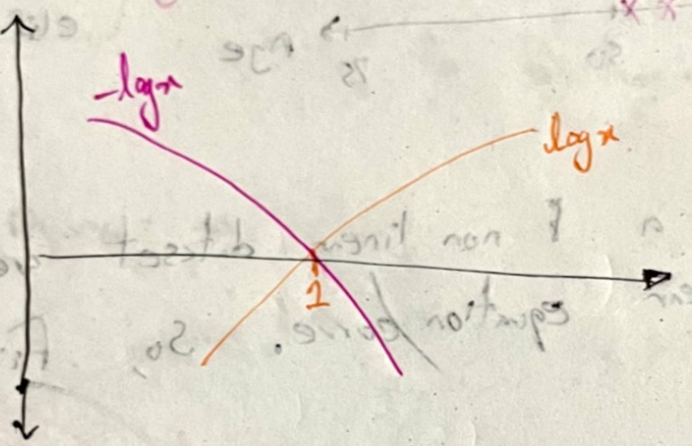
In that case we use to measure distance but ~~here~~ here we are not ~~since~~ measuring the distance.

LOSS FUNCTION OF LOGISTIC R. → Binary Cross Entropy

$$L = -\underbrace{\frac{1}{n}}_{\text{Avg mean}} \sum_{i=1}^{n} \left[y \lg a + (1-y)\log(1-a) \right]$$

y = original value

a = predicted value



Case-1

$y = 0,$

$L = -\log(1-a)$

if $y=0$ ⇒ ~~fro~~ for the classification to be correct it should be $a=0$.

why?

For correct classification, $a=0$ $\quad L = \cancel{-\log 1 = 0} -\log(1-0) = 0$

For Incorrect classification $a \simeq 1$, $\quad L = -\log(1-1) = -\log 0$

Case-2

$y=1$

$L = -[1 \log a]$

correct c. $a \simeq 1$, $L = -\log 1$

Incorrec c. $a = 0$, $L = -\log 0$

Now,

The goal is to optimize the parameter w_1, b.

$z = w_1 x_1 + b$

$\hat{y} = a = \sigma(z) = \frac{1}{1+e^{-z}}$ → It is the curve if trying to fit

$L = -\frac{1}{n} \sum_{i=1}^{n} \left[y \log a + (1-y) \log (1-a) \right]$ → loss function

So the derivative I have to find, i-

$\frac{dL}{dw_1}$, $\frac{dL}{db}$

Then I can optimize

$w_{1(new)} = w_{1(old)} - \alpha \frac{dL}{dw_1}$ | $b_{(new)} = b_{(old)} - \alpha \frac{dL}{db}$

So, our goal is to ~~w_1~~ ~~opt~~ get $\frac{dL}{dw_1}$ or optimize w_1 right! but for that →

To finding → $\frac{dL}{dw_1}$

$w_1 \to V_1$

$$z \to a \to L = -a..$$

$$\frac{dL}{dw_1} = \frac{dL}{da} \times \frac{da}{dz} \times \frac{dz}{dw_1}$$

Here,

$$\frac{dL}{da} = -\frac{y}{a} - \frac{1-y}{1-a} \times \frac{d}{da}(1-a)$$

$$= -\frac{y}{a} - \frac{1-y}{(1-a)} (-1)$$

$$= -\frac{y}{a} + \frac{1-y}{1-a}$$

$$= \frac{-y + ya + a - ay}{a(1-a)}$$

$$\frac{dL}{da} = \frac{a-y}{a(1-a)}$$

$$\frac{da}{dz} = \frac{d}{dz}\left(\frac{1}{1+e^{-z}}\right) = \frac{d}{dz}(1+e^{-z})^{-1}$$

$$= (-1)(1+e^{-z})^{-2}(-e^{-z})$$

$$= e^{-z} \times \frac{1}{(1+e^{-z})^2} = \frac{e^{-z}}{(1+e^{-z})^2}$$

$$\frac{da}{dz} = e^{-z} \times a^2$$

$$= \frac{1-a}{a} \times a^2$$

$$\therefore \frac{da}{dz} = a(1-a)$$

$$\cancel{a} \quad a = \frac{1}{1+e^{-z}}$$

$$1+e^{-z} = \frac{1}{a}$$

$$\Rightarrow e^{-z} = \frac{1}{a} - 1 = \frac{1-a}{a}$$

Now,

$$\frac{dL}{dw_1} = \frac{dL}{da} \times \frac{da}{dz} \times \frac{dz}{dw_1}$$

$$= \frac{a-y}{a(1-a)} \times a(1-a) \times \frac{d}{dw_1}(w_1 x_1 + b)$$

$$= \frac{a-y}{\cancel{a(1-a)}} \times \cancel{a(1-a)} \times \cancel{1} \; x_1$$

$$\frac{dL}{dw_1} = x_1 (a-y)$$

Finding $\frac{dL}{db}$ →

$$\frac{dL}{db} = \underbrace{\frac{dL}{da} \times \frac{da}{dz}}_{a-y} \times \underbrace{\frac{dz}{db}}_{1}$$

$$= (a-y)1$$

$$\frac{dL}{db} = a-y$$

if,

$$z = w_1 x_1 + w_2 x_2 + b$$

then we have to find "$\frac{dL}{dw_2}$" too

$$\frac{dL}{dw_2} = (a-y)\, x_2 \leftarrow \frac{d}{dw_2}(w_1x_1 + w_2x_2 + b)$$

Example :-

x_1	x_2	y
0.5	1.2	0
1.0	0.8	0
1.5	1.3	1
2.0	1.7	1

$$z = w_1 x_1 + w_2 x_2 + b$$

$$a = \frac{1}{1+e^{-z}}$$

ar objective is to find

$$\frac{dL}{dw_1},\ \frac{dL}{dw_2},\ \frac{dL}{db}$$

Initializing $w_1 = 0$, $w_2 = 0$, $b = 0$

x_1	x_2	y	$\frac{dL}{dw_1} = (a-y)x_1$	$\frac{dL}{dw_2} = (a-y)x_2$	$(a-y)$
0.5	1.2	0	0.25	0.6	0.5
1.0	0.8	0	0.5	0.4	0.5
1.5	1.3	1	-0.75	-0.65	-0.5
2.0	1.7	1	-1	-0.85	-0.5

$$\frac{dL}{dw_1} = (a-y)x_1 = \left(\frac{1}{1+e^{-w_1x_1-w_2x_2-b}} - y\right)x_1$$

$$\frac{dL}{dw_2} = (a-y)x_2 = \left(\frac{1}{1+e^{-w_1x_1-w_2x_2-b}} - y\right)x_2$$

Here,

$$\frac{dL}{dw_1} = \frac{1}{4}(0.25+0.5-0.75-1) = -0.25$$

$$\frac{dL}{dw_2} = \frac{1}{4}(0.6+0.4-0.65-0.85) = -0.125$$

$$\frac{dL}{db} = \frac{1}{4}(0.5+0.5-0.5-0.5) = \cancel{-0.2}\ 0$$

learning rate, $\alpha = 0.01$

For 1st iteration $w_{1\,(new)} = w_1 - \alpha \times \frac{dL}{w_1}$

$= 0 - 0.01 \times (-0.25)$

$\therefore w_{1\,(new)} = \cancel{1.25 \times 10^{-3}}\ 2.5 \times 10^{-3}$

$w_2 = w_2 - \alpha \times \frac{dL}{dw_2}$

$= 0 - (0.01) \times (-0.125)$

$\therefore w_{2\,(new)} = \cancel{0}.\ 1.25 \times 10^{-3}$

$b_{(new)} = b - \alpha . \frac{dL}{db}$

$= 0$

Now, if we asked for 2nd iteration →

use, $w_1 = 2.5 \times 10^{-3}$, $w_2 = 1.25 \times 10^{-3}$, $b = 0$

find all the values again in that table →

then find ~~v~~alue of ~~dy~~ :-

$\frac{dL}{dw_1}$, $\frac{dL}{dw_2}$, $\frac{dL}{db}$ ~~Ag~~ Again → w_1 new

w_2 new

b new

Part - 2 → what is the prediction for

x_1	x_2	y
1.17	1.5	??

$$a = \frac{1}{1+e^{-w_1 x_1 - w_2 x_2 - b}} = \frac{1}{1+e^{-2.5\times10^{-3}\times 1.17 - 1.25\times10^{-3}\times 1.5 - 0}}$$

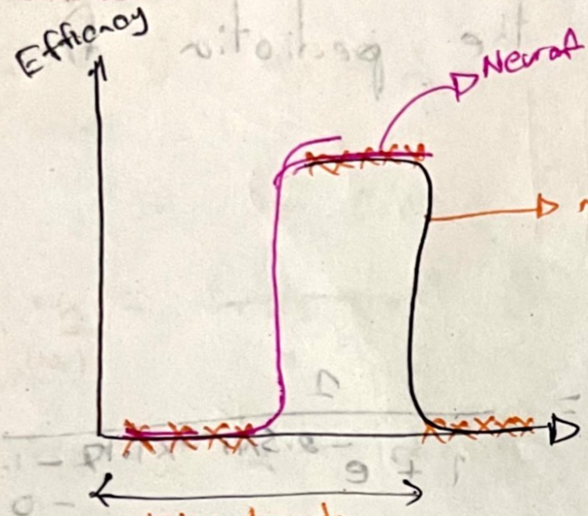
$a = 0.5$ → if $y \geq 0.5 \Rightarrow 1$

$y < 0.5 \Rightarrow 0$

$a \simeq 1$

NEURAL NETWORKS

single layer NN is perceptron.

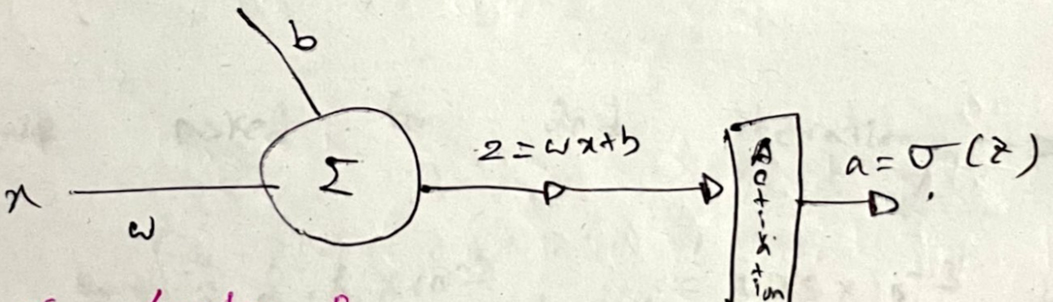


I can activate different neuron according to my usage.

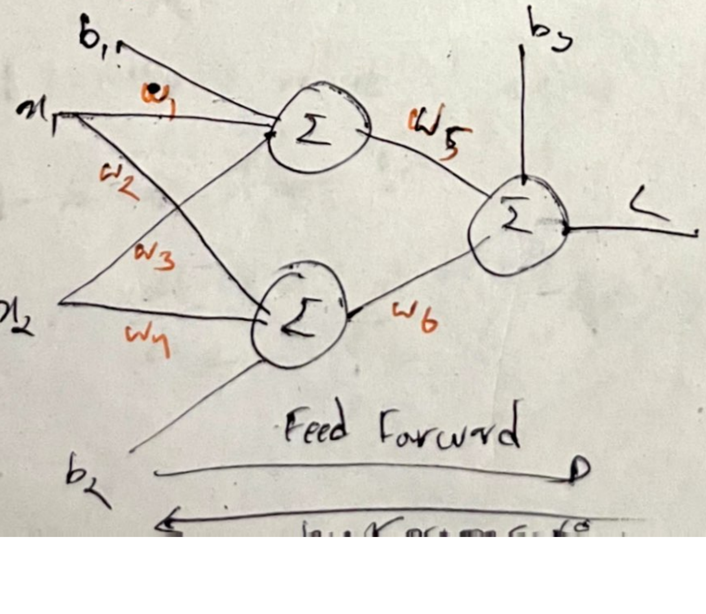
I can train different classifier according to different trend.

→ If I can combine multiple neuron together then we can ~~we can~~ each of ~~predi~~ predict each of the trend seperately

→ NN compiles lot of classifier together.



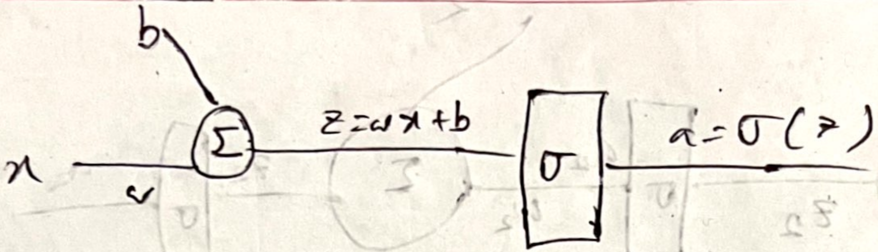
Same as Logistic R.



$\left.\begin{matrix} w = 6 \\ b = 3 \end{matrix}\right] 9$

Now, we ~~are~~ have to optimize 9 parameters for this simple network.

EXAMPLE:-



$w = 0.7$

$b = 0.1$

$\alpha = 1$

x	y	$(a-y)$	$(a-y)x$
0.1	0	0.54	0.054
0.2	0	0.56	0.112
0.3	1	−0.42	−0.127
0.4	??		

$$\frac{dL}{dw} = \frac{1}{3}(0.054 + 0.112 - 0.127)$$

$$= 0.013$$

$$\frac{dL}{db} = \frac{1}{3}(0.54 + 0.56 - 0.42)$$

$$= 0.227$$

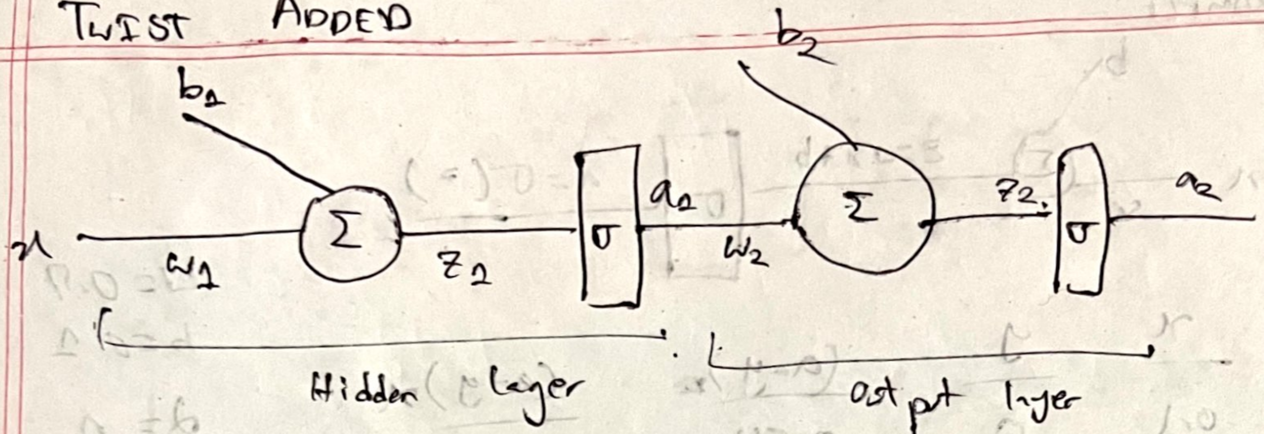
$$w = w - \frac{dL}{dw} \times \alpha = 0.7 - 0.013 \times 1 = 0.687$$

$$b = b - \frac{dL}{db} \times \alpha = 0.1 - 0.227 \times 1 = -0.127$$

Predict

$$a = \frac{1}{1 + e^{-(0.687) \times 0.4 + (-0.127)}} = 0.86 \approx 1 \quad (y \geq 0.5 \Rightarrow 1)$$

TWIST ADDED



Neuron 1

$z_1 = w_1 x + b_1$

$a_1 = \sigma(z_1) = \frac{1}{1+e^{-z_1}}$

Neuron - 2

~~$z_2 = w_2 = $~~

$z_2 = w_2 a_1 + b_2$

$a_2 = \sigma(z_2) = \frac{1}{1+e^{-z_2}}$

Find,

$\frac{dL}{dw_1}, \frac{dL}{db_1}, \frac{dL}{dw_2}, \frac{dL}{db_2}$

Forward Propagation & Backward propagation (Gradient descent)

Assume, $w_1 = 0.1, \quad b_1 = 0.1$

$w_2 = 0.2, \quad b_2 = 0.2$

x	y
10	0

Here,

$z_1 = w_1 x_1 + b_1$

$= 0.1 \times 10 + 0.1$

$z_1 = \frac{11}{10}$

$$a_1 = \frac{1}{1+e^{-z_1}} = \frac{1}{1+e^{-\frac{11}{10}}} = 0.75$$

$$z_2 = \cancel{w_2 a_2}\ w_2 a_1 + b_2$$

$$= (0.2 \times 0.75 + 0.2)$$

$$= 0.35$$

$$a_2 = \frac{1}{1+e^{-z_2}} = \frac{1}{1+e^{-0.35}} = 0.587$$

Forward / Feed forward Propagation

Backward Propagation ($\rightarrow$)

$$L = -\frac{1}{n} \sum_{i=1}^{n} \left[y \log a_2 + (1-y) \log(1-a_2) \right]$$

$$\frac{dL}{dw_2} = \frac{dL}{da_2} \times \frac{da_2}{dz_2} \times \frac{dz_2}{dw_2}$$

$$= (a_2 - y) \times \frac{d}{dw_2}(w_2 a_1 + b_2)$$

$$= (a_2 - y)\ a_1$$

$$\frac{dL}{db_2} = \frac{dL}{da_2} \times \frac{da_2}{dz_2} \times \frac{dz_2}{db_2} = (a_2 - y)$$

$$\frac{dL}{dw_1} = \frac{dL}{da_2} \times \frac{da_2}{dz_2} \times \frac{dz_2}{da_1} \times \frac{da_1}{dz_1} \times \frac{dz_1}{dw_1}$$

$$= (a_2 - y) \frac{d}{da_1}(w_2 a_1 + b_2) \frac{d}{dz_1}(1 + e^{-z})^{-1} \times \frac{d}{dw_1}(w_1 x + b_1)$$

$$= (a_2 - y) \times w_2 \times \cancel{a_1^2 (1 - a_1)} \; a_2 (1 - a_1) \times x$$

$$\frac{dL}{db_1} = (a_2 - y) \times w_2 \times a_2 (1 - a_1) \times 1$$