Nwabt

ANN के lab में Algorithm implement karne python से. Jaise Perceptron, McCulloch etc.

Research

See kahan use hora ANN. Application of ANN के liye karne है → ~~ANN~~ Disease classification, Stock market prediction,

See jahan

→ ANN में → Numerical में ~~लम्बा~~ लम्बा इतना करके (नहीं) | पूरे steps लिखना in numerical then full marks

→ Theory में लम्बा Ans लिखना for high marks.

→ Mid sem syllabus → till Perceptron from starting

Class Test " → " hebb

ANN

→ Practical sub

→ Start studying research paper.

→ 2-3 students group. Make a research paper.

a) Not necessary to stick to single layer feed fwd. You can use Deep Learning

b) Hands on implementation.

c) PPt

d) Pick up a recent research paper 2018-19.

e) Eg → Prediction, Weather Pred., Stock Pred., Medical field, Sentiment Analysis.

f) Read a Research paper. And make a ppt. on what u understood.
Agr interest aaye then its optional to implement.

→ ① Books by Sivanandam & Deepa Soft Computing (purani wali)

② " Principal of Neural N/w

① & ② का content same
we'll follow ①.

→ Syllabus = theory + numerical.
(आधा) (आधा)

We can directly talk to mem

→ Implementation of practical is in python

12 Jan 2022

Introduction

Biological Neuron

→ 2 neurons connect at synapse

→ chemicals

→ Eg: ① touch hot pan, hand retract

② touch sharp object

③ Eye drop coming near to eye, Eye starts blinking

→ first things that come to our mind,
'a' for apple.
'a' for aeroplane

→ But not these are so common.

→ So Biological Neuron basically memorization

→ So ANN takes capability of memorization & generation.

→ Some people have pictographic memory.

→ When F Exam is there, we read ques. first various words, we try to associate it

with the words studied before

→ E.g. People wake up, need a glass of water or tea. It becomes a pattern. If water is not got, then ~~day~~ ~~one~~ we cannot start our day.

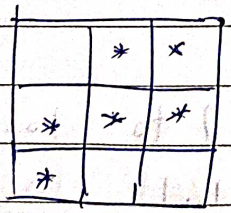
⇒ Ring bell ⇒ pet knows time to eat
⇓
comes running.

⇒ Adaptation in human brain = learning in ANN (by changing weights)

→ BNN has graceful degradation
E.g. old people forget.

ANN also have graceful". But ANN ~~to~~ it we use term "fault tolerance".

→ To make P



I trained my algorithm for all capital letters & small letters.

And I give the above, it's an ~~comp~~ incomplete

pattern.
Ya to wo ~~data~~ it gives "Not found" or a wrong letter.
or it gives P.

Desired o/p = P

other o/p = "Not found" or P some other letter.

when ~~a~~ desired o/p (not) come then ANN ~~it~~ graceful degration.

So Now ANN is trained more.
But upto a certain point, so that overtraining ~~nt~~ hojaye.

~~At~~ At old age, BNN cannot be retrained!
old age ~~ni~~ learning difficult.

learning

2 sections of IT (Roll no. 1 to 100 → Sec^n 1
~~100~~ ≥ 101 → Sec^n 2)

This is classification.

We have dataset. 3 ~~sets~~ parts:
① Training dataset.
② Testing
③ Validation.

In most of the situation, Testing ds = Validat^n ds.

But also they can be diff. 60% Training ds
33% Testing "
7% Validation "

Eg. Student taught "Apple is Red". (Training)
Then ~~she~~ he given "~~Apple~~ Green Apple. (Testing)
Then he ने Apple रहा या नहीं।

Ideally Testing & Validation में No difference.
Both में checking.
Agar practically dekhe, koi medical disease diagnose,
Then testing के baad, in validation I am
confirming the results ~~got in Vali~~. (got in Testing)
in the validation phase.

2) Supervised learning
learning under a teacher.
learning a test.

Eg. of Supervised learning → Classification.

We may divide int 2 classes, 3 classes, etc

{ IRIS flower, ⇒ UCI repo, kaggle }

↓ class Virginica, ...

$X_{i/p}$ = input

y = actual o/p.

Ex → 2 sections. Roll no 101 was classified into sec 1.
But its ~~was~~ ~~does~~ actually ~~is~~ belong to section 2.
y = section 1
D = " 2.

So error signal found $(D - y)$.
send ANN retrain kota.
(update your w/w)

If $y \neq D$, we keep on training our system until $y = D$.
When $y = D$, then stop.

13 Jan 2022

Supervised learning

We know both ques & desired o/p.

Training pair

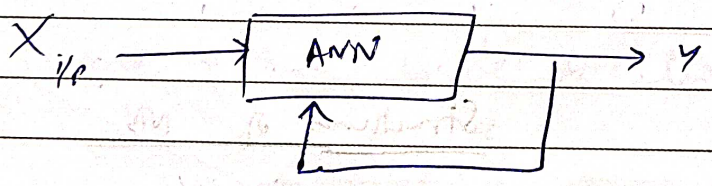
Practically Y cannot be exactly D.
" Y should be as close as possible to D.
for eg., error of 0.0001 is tolerable.

In Retraining we can:
① Update weights
② Change activation functions.

Unsupervised Learning

⇒ "learning from your own experience".

Eg→ In college, when offline, there was no teacher for telling where to sit. Sit with whom.
We choose our friends & sit with them.
clustering happening



Eg: Multiple balls Red, Blue, Green.
3 boxes for each. Segregate them
But there is also 1 pink ball. where to put.
Put in Red box, bcz pink is nearer to red.

This is unsupervised.

Critique kota. ~~Ye kya galat hai ye hi batata ki tera percent sahi ya galat ye batata~~

Page No. 34
Date: / /

Reinforcement

→ It lies b/w Supervised & Unsupervised.

→ Yahan desired o/p nahi hai.

→ Dataset nahi hai train karne ke liye.

Ex. Recommender systems in Youtube, Spotify.

Youtube pe dekha kuch. Then usse ke ke hisaab se dikhane lagega.

→ Reinforcement = best of both sup. & unsup.

unsup. me RE ne uthaya ki main khud seekhunga -

~~!! sup. me ye uthaya ki grading mil milega~~

=> Supervised learning is more efficient than Unsup.

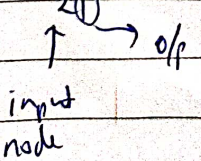
But we cannot say Sup. is faster than Unsup.

Structure of NN

Input layer is **not** counted in working layers

∴ 2 layers with computation

I/p layer pe computation (nil) hota.



⇒ For logical,

⇒ For stock market predict, weather prediction, find whether in some area soil is fertile ~~or~~ or not, predicting ~~desi~~ disease etc., single layer architecture (not) sufficient.
~~Sing~~

Multi-layer Feedforward Spp.

2 To make it : 2 ways :

① Constructive.
Start by a simple st^r. Add more neurons while training.
Agr overtraining hone lage, then hidden nodes reduce.

② Pruning → Big st^r start. delete nodes.

⇒ Hidden layer ki weights denote by "v"

feed forward → bus single direction.

~~Single layer R. Recurrent N/w~~

⇒ Recurrent N/w are used for memory.

Single layer Rec. N/w

Node ke ~~ki~~ o/p khud ki st. & other nodes ki.

Feedback v/s Recurrent

~~In feedback~~
When we are dealing with single o/p neuron, then we say "feedback".
Else we use term **Recurrent**.

⇒ Recurrent N/ws are highly unstable.
Bcoz " " we cannot update multiple weights at a time, bcz ~~ek~~ ek wt. update (~~se~~ se o/p change wahan, & then wahan se peeche aayege then aage aayega.

? And aap agr koi aur jgh bhi simultaneously ~~upd~~ wt. update then maybe kisi ~~k~~ node ~~पे~~ ~~2 updations~~ ke i/p पे 2 diff. o/ps आयेगा।

Learning Summary

1) Supervised
- Teacher hai guide karne ~~key~~ ke liye. ~~ye~~ Harr step पे ~~batt~~ batata kya galt kreg. } Adv
- Grading bhi ~~hoti~~ hoti } disadv.

2) Unsupervised
- koi ni hai kuch btane wala ki kya ~~kitti~~ galti kre } disadv
- Khud se learning } adv

3) Reinforcement

- Koi hai guide karne ko. Wo ye btayega ki kya galti kare ho. } Adv. of Sup. taken & disadv of unsup removed
- Grading nhi hoti } disadv of Sup. not taken
- Khud se learning } Adv. of unsup. taken.

As Reinforcement me dono ke Adv. aagye & disadv rah gaye, isliye its called best of both worlds.

→ In computer level, grading means Confusion Matrix me 100 me sahi bta diye toh 90% accuracy.

---

17 Jan 2022 ~~Activation functions~~

→ Supervised ka eg. – classification
Unsup. " – clustering ?

Reinforcement : Recommender system

Activation function

Every i/p is denoted by x

$x_1, x_2, \ldots, x_n$

y_{in} → net i/p

Net i/p $y_{in} = \text{I/p} \times \text{wt.}$

$y_{in} = x \cdot w$ (net i/p y_{in} is ~~actual o/p~~)

To make learning better & to introduce Non-linearity we use ~~activation~~ activation function.

Activation fun^n = Squashing fun^n = Transfer fun^n

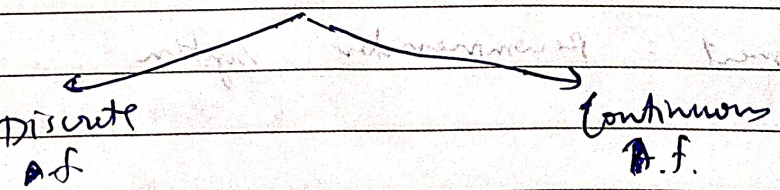
$x \cdot w$ = linear.

⇒ Hum a.f. isliye use krte jisse hmara o/p desired o/p ke close aane lage

⇒ ~~Yh not~~ You can use different a.f.s in different layers.

Types

(1)



* Discrete f. cannot be differentiate.

Continuous f. also called Sigmoidal f.

We can always use Sigmoidal f in o/p layer.

⇒ Taken them Yes/No choice → use discrete A-f.

⇒ Non-Sigmoidal A.f. will perform ~~if~~ as good as sigmoidal.

⇒ Hidden layer ⇒ Non-sigmoidal ReLU performs best.

⇒ Binary step, → threshold.
$\theta > 0$

θ = ~~Kitta step~~ that ~~per~~ particular point where a neuron ~~Neuron~~ ~~or~~ goes in firing mode.

Q) When a neuron ~~be~~ goes in firing mode?

Ans) ~~a Neuron~~ When neuron Fi $(x.w) \geq \theta$
net ~~in~~ input.

Otherwise the neuron is not in firing mode.

* Always, neuron ek baar fire hota, it will come into non-firing mode & gets its strength back & then goes back into firing mode

* $y_{in} = x \cdot w$ → net i/p

  * if $y_{in} \geq \theta$ → neuron will fire
    $y_{in} < \theta$ → won't fire.

$y = f(y_{in})$ → activation function

⇒ Binary Step : $f(x) = \begin{cases} 1, & \text{if } x \geq \theta \\ 0, & x < \theta \end{cases}$

1 → neuron firing mode
0 → neuron not firing mode

⇒ Bipolar Step : $f(x) = \begin{cases} 1, & \text{if } x \geq \theta \\ -1, & x < \theta \end{cases}$

1 → neuron firing mode
-1 → neuron not firing mode

(3) ~~Sf~~ Sigmoidal

a) Binary / Unipolar

λ = steepness

$\lambda = 0.5$ gives good result

In exam, agar λ not given, assume $\lambda = 1$.

λ change kare se we can go close to our desired o/p. We have to also make a graph desired o/p v/s actual o/p. And do hit and trial kake dekha jaaega.

★ ~~When~~ When we'll ~~do~~ use Backpropagation Algo, then we should use sigmoidal Activation func, bcoz its ~~used~~ easy to calculate error signal in them.

⇒ Look out for different datasets that can be used as:

① Super. learning
② Unsup
③ Reinforc

~~Aaa~~

19 Jun 2022.

Page No. 42
Date: / /

activation_function_points wala pdf samajh nhi aayi ??

Monotonicity

If size of 1 variable ↑es, then size of another var ↓

OR

★ Activation function should be non-increasing or non-decreasing.

⇒ If we keep on increasing of the weight, then at some point our o/p may not affected. Our system may be overtrained.

⇒ How much ~~activation func~~ weight we should ↑e depends on ~~we~~ activation funcⁿ chosen.

→ We have to increase wt. slowly & not change it abruptly.

⇒ Sigmoid activation funcⁿ are always ~~monotono~~ Monotonic, continuous & differentiable everywhere.

↓ Eg ⇒ 50 min class Agr continuous 50 min class, kitta retain? Poora to yaad nhi rehta

epoch = how many times we run our system.
We decide how many times we run the system
Say till 100 epochs, error = 0.5 (acceptable)
But maybe so that our team have error. (greed)
Then I will overtrain my system. Then error go to
0.8, 0.9 or 1.

Similarly in ANN, ek point pe aata ANN stops learning
Then it won't learn.

How to decide no of epochs?
If I have 100 instances, then its tested ki 1000 epochs
ke liye acha rahega.

Agar > 100 instances, 500 instances, I can go upto 10000
epochs.

⇒ Say I wanna do on Breast cancer project, I will read its research papers.

Un research papers me ray hoti me 10k, 15k,
20k epochs.
Kisi me > 20k bhi suna!
So mujhe no. of epochs ka idea lag jayega.

Ideally no. of epochs mathematically nikal sakte.

⇒ For logic gates, Perceptron 2-3 epochs fine

⇒ For images, agar batch bnaya, 100 epochs is fine.

VGP ??

Vanishing Gradient problem (V.G.P)

Sigmoidal A.f. से hoti hai (V.G.P.).

To avoid this V.G.P, we use non-sigmoidal ~~Problem~~ a.f. → ReLU, leaky ReLU, softmax.

~~These~~

ANN में all activation functions are monotonic.

Agar backprop करते हैं तो उसमें function को differentiable bhi hona padega.

⇒

20 Jan 2022

⇒ Bias = extra input.

⇒ Binary = 0, 1

Bipolar = -1, 1

⇒ Bias denoted by 'b'.

⇒ bias of always value 1.
वाले I/P

⇒ We have to always use bias

(α)

⇒ Learning rate → speed at which we want our system to learn/get trained.

↳ Convergence = getting closer to desired o/p

⇒ If we start with $\alpha = 0.1$. Its very high.

~~We want to~~ ↳

We try to keep α as small as possible.

Like $\alpha = 0.001$, $\alpha = 0.01$, $\alpha = 0.02$ ...

Low but ~~to~~ overtraining ka ho jisse

⇒ Bias v/s threshold

Threshhold = the ~~limit~~ value at which neuron is fired

Actual Bias & threshhold are diff..

Mc Culloh Pitts Model

↳ not a computational model.

↳ used to design logic gates.

↳ Whenever u. I have to any able that neuron will fire or not.

↳ ✱ MCP model always work on Single Layer feed forward n/w

↳ I/p layer & o/p layer & no hidden layer

→ Linear Separability → If i/ps can be seperately seperated by drawing a straight line (in 2D) or by hyperplane (in 3D), then we call it Linear seperably.

General

Class 1 = (+)

Class 2 = (−)

→ Single o/p layer for MCP

→ MCP works only on binary data (0, 1)

→ 1 → neuron firing mode

→ 0 → non-firing mode

→ ~~If any excitatory wt.~~

→ All inhibitory wt. should be same.

excitatory

inhibitory & excitatory ~~can be~~ are different.

⇒ Bias is an extra external i/p.

Page No. 47
Date: / /

24 Jan 2022

Mc Culloch Pitts Neuron

MCP Model :- (Book Page 27, Pdf Page 23)

⇒ directed weight paths

⇒ neuron may / may not fire

⇒ weights may be excitatory (+) or inhibitory (−)

⇒ All excitatory wt.s into a particular neuron have same weights.

⇒ fixed Threshold

$$f(y_{in}) = \begin{cases} 1, & \text{if } y_{in} \geq \theta \\ 0 & \text{if } y_{in} < \theta \end{cases}$$

⇒ $y_{in} = x_1 w_1 + x_2 w_2 + x_3 w_3 + \dots + x_n w_n$

$$= \sum_{i=1}^{n} x_i w_i$$

⇒ for inhibition to be absolute,

$$\theta > nw - p$$

We don't will do not use it.

Instead derivation mai learn

Here θ is bias

→ o/p will fire if it recieves 'k' or more excitatory i/ps but no inhibitory

$(kw) \geq \theta > (k-1)w$

actually $x_1 = 1$, $x_2 = 1$, ..., $x_k = 1$ } it fires when all i/ps 1 means high

Q) AND Table using MCP (Book Pg 34 (book), Q 4)

	x_1	x_2	y	$y_{in} = w_1x_1 + w_2x_2$ ~~Assume $w_1 = w_2 = 1$~~
1)	1	1	1	→ 2
2)	1	0	0	→ 1
3)	0	1	0	→ 1
4)	0	0	0	→ 0

Assume $w_1 = w_2 = 1$

There are 4 cases.

$y_{in_1} = w_1x_1 + w_2x_2 = 1 \times 1 + 1 \times 1 = 2$

$y_{in_2} = 1 \times 1 + 1 \times 0 = 1$

$y_{in_3} = 1 \times 0 + 1 \times 0 = 1$

$y_{in_4} = 1 \times 0 + 1 \times 0 = 0$

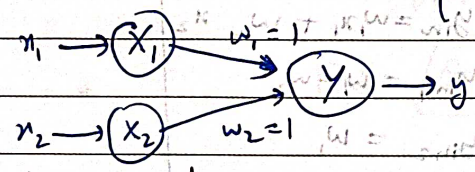
All Realizations ~~that~~ ques. ~~of~~ (AND, OR etc) ~~of~~ architecture ~~bhi~~ banana.

Page No. 49
Date: / /

We want y to be 1 only in 1st case - i.e. we want O/p neuron to fire only in 1st case.
And 1st case ~~of~~ net i/p (y_{in}) is 2.
∴ θ ~~of~~ 2 rakh dete.

$\boxed{\theta = 2}$

$$y = f(y_{in}) = \begin{cases} 1, & y_{in} \geq 2 \\ 0, & y_{in} < 2 \end{cases}$$



~~Q)~~ Realize logical OR using MCP model.

In all these type ques weights hit & trial ~~of~~ ~~of~~ lene. And kaise bhi weights le sakte (+) ya (−).

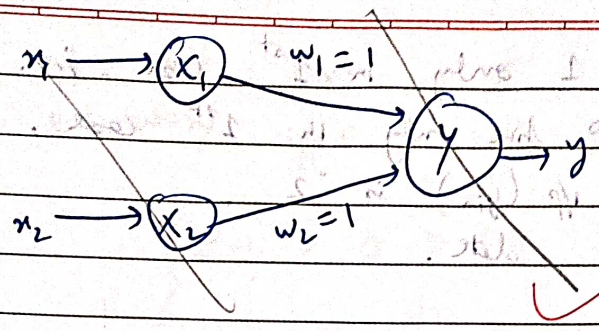
(a)) OR . ~~Assume w_1 =~~ Assume $w_1 = w_2 = 1$

net input

	x_1	x_2	y	$y_{in} = w_1x_1 + w_2x_2$	y_{in}
①	1	1	1	$y_{in_1} = w_1 + w_2$	2
②	1	0	1	$y_{in_2} = w_1$	1
③	0	1	1	$y_{in_3} = w_2$	1
④	0	0	0	$y_{in_4} = 0$	0

Take $\boxed{\theta = 1}$. Then cases ①, ②, ③ ~~of~~ fire.

$$y = f_1(y_{in}) = \begin{cases} 1, & y_{in} \geq 1 \\ 0, & y_{in} < 1 \end{cases}$$



Q) Realize ~~OR~~ logical OR using MCP model.

(a))

	x_1	x_2	y	$y_{in} = w_1 x_1 + w_2 x_2$
(1)	1	1	1	$y_{in_1} = w_1 + w_2$
(2)	1	0	1	$y_{in_2} = w_1$
(3)	0	1	1	$y_{in_3} = w_2$
(4)	0	0	0	$y_{in_4} = 0$

Ab mein ne kaha ki wait and find ~~of~~ weights decide.

Pehle, take $w_1 = 1$, $w_2 = -1$

$w_1 = -1$, $w_2 = 1$

$w_1 = 1$, $w_2 = 1$.

Take $w_1 = 1$, $w_2 = -1$;

$y_{in_1} = 0$
$y_{in_2} = 1$
$y_{in_3} = -1$
$y_{in_4} = 0$

} inpe fire karena chahte

Agar $\theta = 0$ ✗, $\theta = -1$ ✗, $\theta = 1$ ✗, $\theta = 2$ ✗

$\theta = -2$ ✗

Isse nahi hoga.

Take $w_1 = -1$, $w_2 = 1$, again find all y_{in}.
Isse bhi nhi hoga.

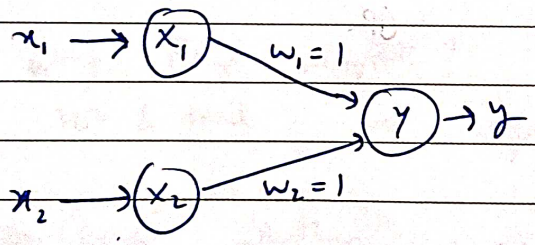
Now, $w_1 = 1$, $w_2 = 1$ ⇒

$y_{in_1} = 2$
$y_{in_2} = 1$
$y_{in_3} = 1$
$y_{in_4} = 0$

Take $\theta = 1$ ✓

∴ We finally Assume $w_1 = 1$, $w_2 = 1$ & $\theta = 1$

$$y = f(y_{in}) = \begin{cases} 1, & y_{in} \geq 1 \\ 0, & y_{in} < 1 \end{cases}$$



☆ In general in any model हमें बस check करना hota hi y_{in} का θ से साथ kya relation

If $y_{in} \geq \theta$ → neuron will fire
$y_{in} < \theta$ → " " not fire.

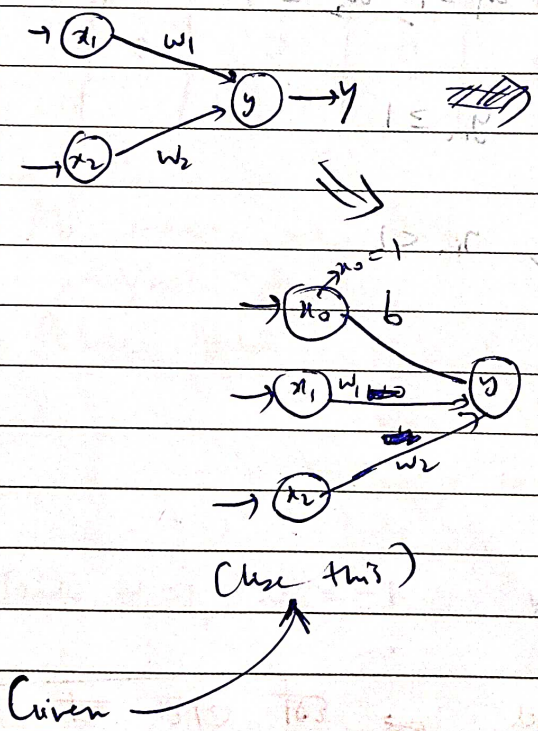
No. of excitatory या inhibitory wt.s से No मतलब
(Bas ye yaad rakhna hai ek inhibitory to not fire)

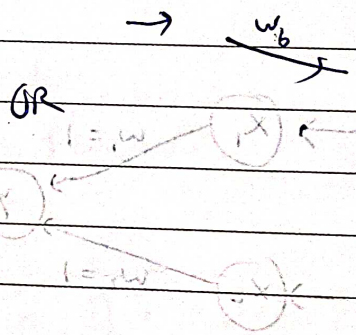
Linear seperability

Roll no 1 to 50 → batch 1
" 50 - 100 → " 2

We have seperated the Batches. A line is drawn. Linear seperable line
(decision boundary)

→ If some point lies on decision boundary then it cannot be classified





$$y_{in} = b + x_1 w_1 + x_2 w_2$$

In general, $$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

Now, for a point on decision boundary, $y_{in} = 0$

$b + x_1 w_1 + x_2 w_2 = 0$

⇒ Assume x_1 as x, x_2 as y (Maths)

⇒ $$x_2 = -\frac{b}{w_2} - x_1 \frac{w_1}{w_2}$$

$y = mx + c$

By comparing x_2 eq. with $y = mx + c$,

$$c = -\frac{b}{w_2}, \quad m = -\frac{w_1}{w_2}$$

Now we can find weights.

why find weights?
Hit & trial ni layenge pdta.

⇒ Check whether a given gate is linear seperable or not?

2 things to do

① Make

See ques on next page to see how to do working.

⇒ Whenever we use linear Seperability, then we have bipolar i/ps & targets.

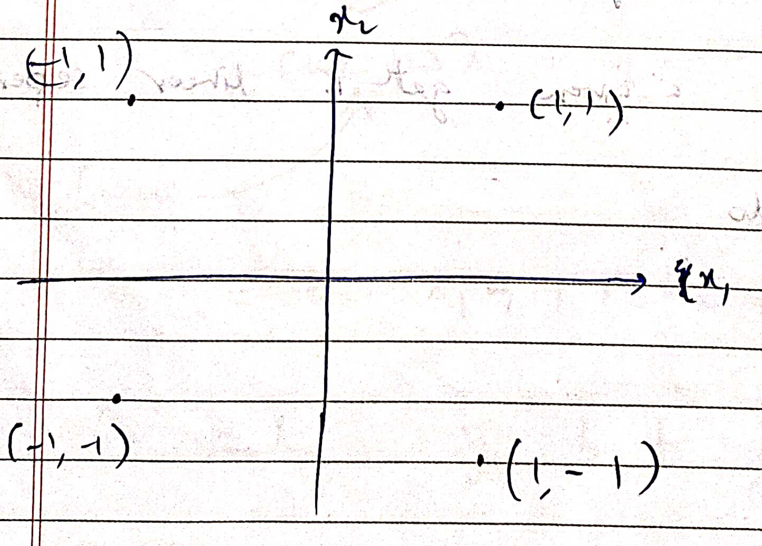
∴ Linear Seperability से हम MCP model nikalge skte, pas MCP use kr skte

Q) Check whether OR is linear separable or not (bipolar)

x_1	x_2	y
1	1	1
1	−1	1
−1	1	1
−1	−1	−1

0 की jgh −1

⇒ Now make graph, & plot all (x_1, x_2) combinations



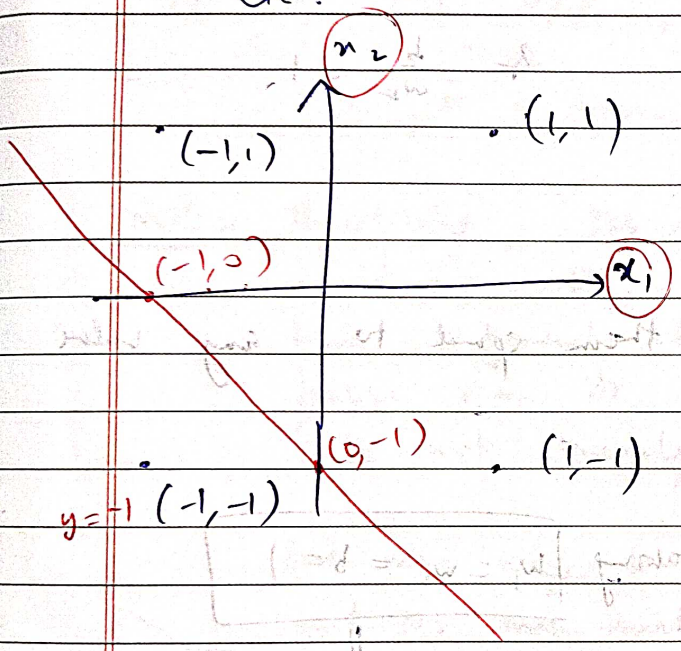
The combinations (1, 1), (1, −1), (−1, 1) belong to same class bcz input same o/p $y = 1$.

So make any line, such that these 3 points fit seperate from the 4th point.

Jaise (-1, 0) to (0, -1). [Ye take this]

Jaise (-1, 0) to (0, -2)

etc.



Hum aisi line banate hai ki koshish hote ki dt i/p's jinka y same aaye dt line ki ek taraf aajaye.

The linear seperable wali line ka 2 Points known. So we can find its eqn.

$(x_1, x_2) = (-1, 0)$ & $(0, -1)$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{-1 - 0}{0 - (-1)} = -1.$$

$c = -1$

$\therefore y = -x - 1$ $\quad$ $(x_1 = -x_1 - 1)$

And eq. is, $x_2 = -\frac{x_1 w_1}{w_2} - \frac{b}{w_2}$.

Comparing these 2 eqn.

$\frac{w_1}{w_2} = 1$ $\quad$ & $\quad$ $\frac{b}{w_2} = 1$

$\therefore \boxed{w_1 = w_2 = b}$

We can take them equal to any value say 1, 2, 3 etc.

I am taking $\boxed{w_1 = w_2 = b = 1}$

$\Downarrow$

Now,

x_1	x_2	x_0	$y_{in} = c$
1	1	1	3
1	-1	1	1
-1	1	1	1
-1	-1	1	-1

agar XOR linear separable nhi hoga then desion boundary bana hi ni payenge

Table $\theta = 1$

so $y_{in} = 3$ so $y = 1$. ⇒ $y_1 = 1$
$y_2 = 1$
$y_3 = 1$
$y_4 = -1$

Now as this o/p, matches with the initial OR table

∴ OR gate is linear seperable.

NOTE → Theoritically, Jab humne graph banaya that x_1 & x_2 ka. Then agar ek line se hum different classes ko separate kr paare then its linearly seperable.
Else not linearly seperable.

But use ager ki working dikhani hai poori, just for ~~neat~~ marks.

(*) अगर डाटा gets linearly seperable only then uspar ~~[illegible]~~ MCP लगा सकते।

Q)

Upload a word doc that has code & SS of ~~o/p~~ output code.

1 & -1 के wt. के लिए 4 combinations

O/p) ~~[illegible]~~, final weights

Use python only

---

31 Jan 2022

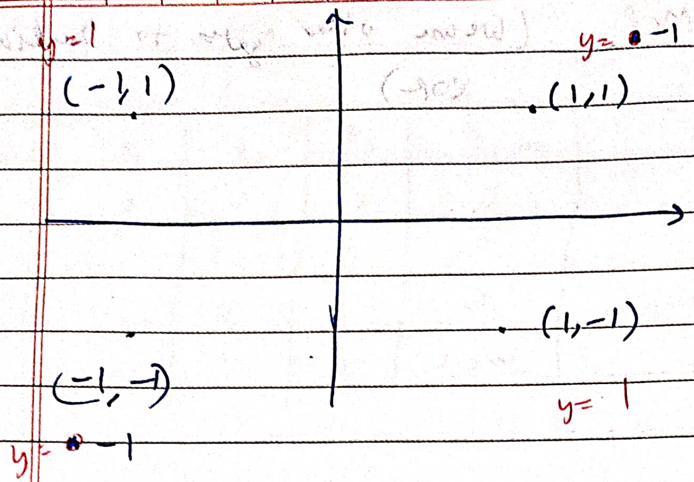
→ Inputs को classify into respective classes in Linear seperability.

→ अगर 2 classes में divide करते।

⇒ XOR (same i/p then O/p 0, else 1.) (Bipolar i/ps)

x1	x2	y
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

* MCP model works only on single layer feed feed architechture



Hum aisi line bana hi (नहीं) पाए ki y=1 line ki ek side आजाए & y=−1 doosri side.

We are not able to draw a line so as to clessify the points on the basis of value of y.

So XOR is (not) linearly seperable.
Hence MCP not valid.
Then ~~to~~ hum wo saari calculation bhi ni kar payenge & w_1, w_2 nhi nikal payenge.

So we need to change the architechture.

Basically $y = x_1\bar{x}_2 + \bar{x}_1 x_2$

PTO

⇒ XOR Gate using MCP (We use other gates to realize XOR)

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

$y = x_1\bar{x}_2 + \bar{x}_1 x_2$

$x_1\bar{x}_2 = z_1$, $\bar{x}_1 x_2 = z_2$

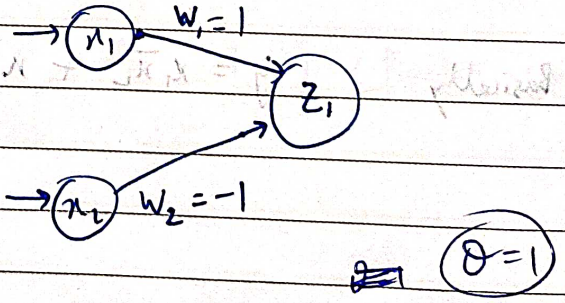
$\therefore z_1 = x_1\bar{x}_2$ → 1st single layer feed fwd arch.

$z_2 = \bar{x}_1 x_2$ → 2nd

$y = z_1$ OR z_2 → 3rd

For $z_1 = x_1\bar{x}_2$

x_1	x_2	z_1	$z_{1,in}$
0	0	0	0
0	1	0	w_2
1	0	1	w_1
1	1	0	$w_1 + w_2$

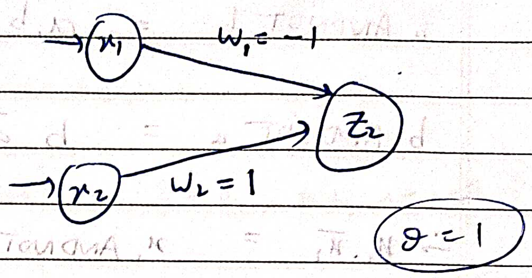


NOTE → Exam में hum ने $x_1\bar{x}_2$ & $\bar{x}_1 x_2$ दी w_1, w_2 directly likh sakte * No hit & trial required. ये bohot common है यार ये याद रखना।

Linear separability It Always Bipolar I/Ps.
But realization in MCP It take Binary i/p

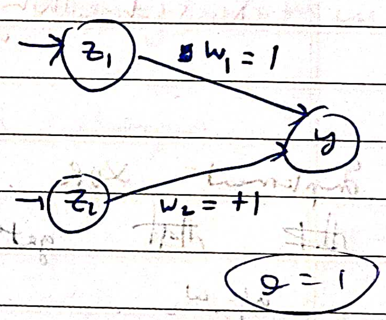
for $Z_2 = \bar{x}_1 x_2$

x_1	x_2	Z_2	$Z_{2,in}$
0	0	0	0
0	1	1	w_2
1	0	0	w_1
1	1	0	$w_1 + w_2$

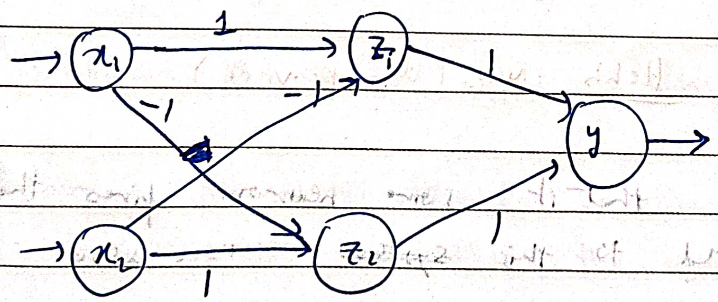


Now, $y = Z_1 + Z_2$ (OR)

Z_1	Z_2	y	y_{in}
0	0	0	0
0	1	1	w_2
1	0	1	w_1
1	1	1	$w_1 + w_2$



Combining all these 3,



Varsney

Z_1, Z_2, y ... की $f(\ )$ ...
≥ 1

So the XOR gate can be realized using MCP model.

Basically XOR gate is realized using other logic gates which are further real

NOTE → ANDNOT → 2nd wala negation

a ANDNOT b = $a.\bar{b}$

b ANDNOT a = $b.\bar{a}$

∴ $x_1 . \bar{x}_2$ = x_1 ANDNOT x_2

$\bar{x}_1 x_2$ = x_2 ANDNOT x_1.

NOTE → In case of XOR we clubbed 3 individual models. Sabki kisi ka apne θ Extra. No such thing as overall θ.

Q) Implement XOR.
Aage gets le liye we can hardcode w_1, w_2.

---

7 Feb 2022 Learning Rule

Hebb N/W (Unsupervised)

Its target is that if some neuron fires then neurons connected to this ~~system~~ neuron also progress in this direction.

We cannot control i/ps. But we can set weights.

→ Generally in Hebb, we initialize weights with value 0

⇒ Training pair, s:t

→ $\Delta W = r x \alpha$

α = learning rate (set by us) (Eg → 0.1, 0.2, 0.4 etc)

→ If in ques, α not given take $\alpha = 1$.

→ Practically we keep $\alpha = 0.1, 0.01$.

⇒ r = learning signal (depends on the learning rule we use)

→ When learning rule changes, only r changes.

→ x = i/p to that particular neuron.

→ In Hebb, whatever o/p is given to me I am assuming it to be ~~correct~~ desired o/p. I am not retraining my system. I'll just update the weights.

Ans: Hebbian ~~is~~ comes under Unsupervised.

Reason → No error signal is generated. No retraining. ~~Nobo~~ Nobody is telling the n/w.

∴ Unsupervised.

→ In Hebb, we assume कि जो desired o/p given है is its also my actual o/p.

We don't calculate y_{in} & y. So हमें अपने o/p से मतलब नहीं।

⇒ Let's say we have 10 i/p, उन सभी के लिए 1 बार sum. Then this is 1 epoch.

⇒ So 1 epoch ≡ running your system for all i/ps.

⇒ We can

⇒ In Hebb, यदि change in wt. is increasing then it means we are overtraining so stop.

⇒ In the exam, if nothing is mentioned, then we have to do for 1 epoch.

⇒ (1) Hebbian uses unsupervised learning

⇒ (2) wt. initialised to 0

(3) It only works on Bipolar. (i/p is 1 or -1)

(4) Whatever i/p & o/p are given they are as it is presented to the system

Q) Train your logical AND/NAND using Hebb learning rule

Not given → 1 epoch

x_1	x_2	x_0 (bias का i/p)	y
1	1	1	1
-1	-1	1	-1
-1	1	1	-1
1	-1	1	-1

→ 1 epoch = All i/p pairs
यदि train करना है
यहाँ 4 i/p pairs
∴ 1 epoch = 4 iterations

$\gamma = y$ in Hebb

Page No. 65
Date: / /

Assum $\alpha = 1$

Initially, $w_1 = w_2 = b = 0$

1. For i/p, $[x_1 \; x_2 \; x_b] = [1, 1, 1]$
$y = 1$ → training pair

w_1 new $= 0 + \gamma x_1 = 0 + (y)(x_1)(1)$
$= 0 + (1)(1)(1) = 1$

w_2 new $= 0 + (1)(1)(1) = 1$

$b_{new} = 0 + (1)(1)(1) = 1$

Now these values of w_1 new, w_2 new & b_{new} will act as old values for the next training pair.

2. For i/p, $[1 \quad -1 \quad 1]$
$y = -1$

$w_{1\,new} = w_{1\,old} + \gamma x_1$
$= 1 + (y)(x_1)(1)$
$= 1 + (-1)(1)(1) = 1 - 1 = 0.$

$w_{2\,new} = 1 + (-1)(-1)(1) = 2$

$b_{new} = 1 + (-1)(1)(1) = 0.$

3. for i/p, $[-1 \quad 1 \quad 1]$,
$y = -1$

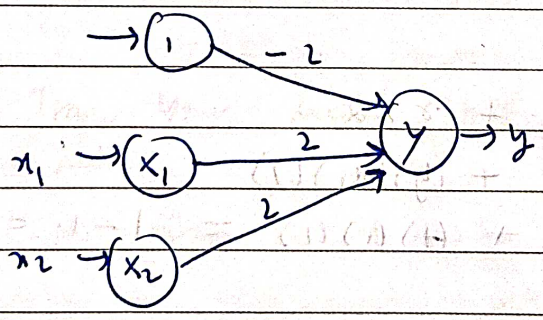
$w_{new} = w_{old} + y x \alpha = w_{old} - x$

$w_{1,new} = 0 - (-1) = 1$
$w_{2,new} = 2 - (1) = 1$
$b_{new} = 0 - 1 = -1$

4) for i/p, $[-1 \quad -1 \quad 1]$
$y = -1$

$w_{new} = w_{old} + y x \alpha = w_{old} - x$

$w_{1,new} = 1 - (-1) = 2$
$w_{2,new} = 1 - (-1) = 2$
$b_{new} = -1 - 1 = -2$



(Ans)

If α not given, assume it to be 1 & write it that you are assuming it

NOTE: In exam hume aise har i/p pair ke liye Poora calculation karna hai. Similarly skip ni chodna. And always make the architecture

12 Feb 2022

~~1) Apply Hebb learning rule in logical ANDNOT gates.~~

Q) Find the new wts after applying Hebb learning rule to logical ANDNOT gate after one epoch of training. Assume learning rate as 1

x_1	x_2	x_0	$y = x_1 \bar{x}_2$
1	1	1	-1
1	-1	1	1
-1	1	1	-1
-1	-1	1	-1

Initially $w_1 = w_2 = b = 0$

1) for i/p $[x_1\ x_2\ x_0] = [1, 1, 1]$

$y = -1$

$w_{1\,new} = 0 + (-1)(1)(1) = -1$

$w_{2\,new} = 0 + (-1)(1)(1) = -1$

$b_{new} = 0 + (-1)(1)(1) = -1$

2) for i/p $[x_1\ x_2\ x_0] = [1\ -1\ 1]$

$y = 1$

$w_{1\,new} = -1 + (1)(1)(1) = 0$

$w_{2\,new} = -1 + (1)(-1)(1) = -2$

$b_{new} = -1 + (1)(1)(1) = 0$

3) for i/p, $[x_1 \; x_2 \; x_0] = [-1 \quad 1 \quad 1]$

$y = -1$

$w_{1\,new} = 0 + (-1)(-1)(1) = 1$

$w_{2\,new} = -2 + (-1)(1)(1) = -3$

$b_{new} = 0 + (-1)(1)(1) = -1$

4) for $[-1 \quad -1 \quad 1]$

$y = -1$

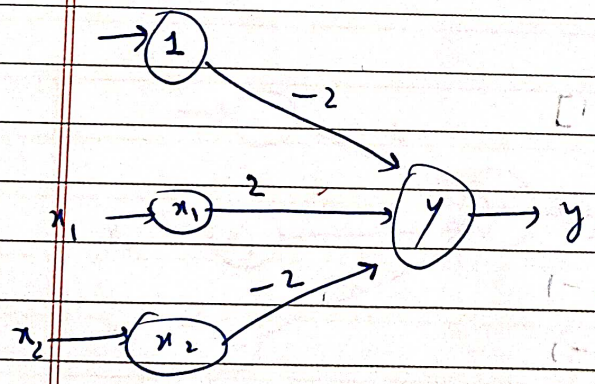
$w_{1\,new} = 1 + (-1)(-1)(1) = 2$

$w_{2\,new} = -3 + (-1)(-1)(1) = -2$

$b_{new} = -1 + (-1)(1)(1) = -2$

∴ these are final wt.s

1 epoch complete.



Always make architecture after an epoch.

NOTE:- Hebb learning rule points

1. Always ~~रख~~ take bias in ~~training~~ Hebb learning rule. Bias i/p always 1

2. Training pairs को रख है particular order follow karo men wala hamesha

x_1	x_2
1	1
1	-1
-1	1
-1	-1

⇒ 3. NOTE:-

$\Delta W = r x \alpha$

r = learning signal.

~~r changes per fn~~

अलग learning rules के लिए r different

In Hebb learning rule, $\boxed{r = y}$

$W_{new} = W_{old} + \Delta W$

⇒ Order of training pairs different करने से final Ans of an epoch different aa sakta. But in diff wts से efficiency par (no) fark.

Class 1 = Positive output ($y=1$)
Class 2 = Negative " ($y=-1$) → target

Page No. 70
Date: / /

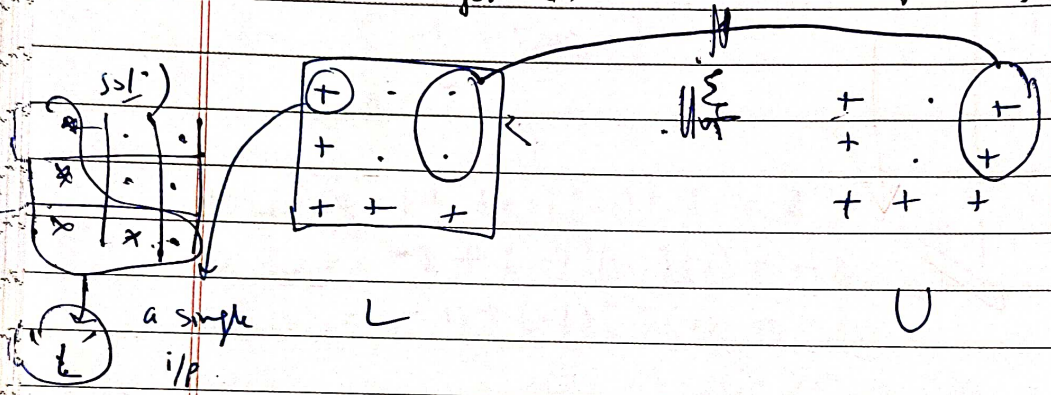
Always bias i/p 1 ही लेना होता

3 Feb 2022

Q) Classify L & U. using Hebb n/w. | 3×3 Matrix

L belongs to class 1. U belongs to class 2

i/p = +1 for (+) & -1 for (.)



∴ Total 9 (nine) i/ps.

$x_1, x_2, \dots x_9$

x_0

And an extra i/p 1 for bias x_0.

∴ Total 10 inputs.

L belongs to class 1 $\Rightarrow y = 1$

U class 2 $\Rightarrow y = -1$

Table ले train करना

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

Table

Target value means output (y). in Hebb, here in Hebb we assume Target as y..

Page No. 71
Date: / /

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y
L	1	1	-1	-1	1	-1	-1	1	1	1	1
U	1	1	-1	1	1	-1	1	1	1	1	-1

Pehle L ke liye train karo. Then use jo wts aayenge unhe use as old for next i.e. U. Then train for U.

2 ही i/p के लिए 2 बार train करना.

Initially $w_1 = w_2 = \ldots = w_9 = b = 0$

And assume $\alpha = 1$

1) for i/p L,

$y = 1$

$w_{1new} = 0 + (1)(1)(1) = 1$
$w_{2new} = 0 + (1)(-1)(1) = -1$
$w_{3new} = 0 + (1)(-1)(1) = -1$
$w_{4new} = 0 + (1) = 1$
$w_{5new} = 0 + (-1) = -1$
$w_{6new} = 0 + (-1) = -1$
$w_{7new} = = 1$
$w_{8new} = = 1$
$w_{9new} = = 1$
$b_{new} = 0 + (1) = 1$

→ L के लिए जहाँ + था उसके corresponding wt. +1 जहाँ . वहाँ -1.

Hebb → Unsup

Page No. 72
Date: / /

2) for U,

$y = -1$

$w_{new} = w_{old} + y \times \alpha = w_{old} + (-1)x(1) = w_{old} - x$

$w_{1,new} = 1 - 1 = 0$

$w_{2\,new} = -1 - (-1) = 0$

$w_{3\,new} = -1 - 1 = -2$

$w_{4\,new} = 1 - 1 = 0$

$w_{5\,new} = -1 - (-1) = 0$

$w_{6\,new} = -1 - 1 = -2$

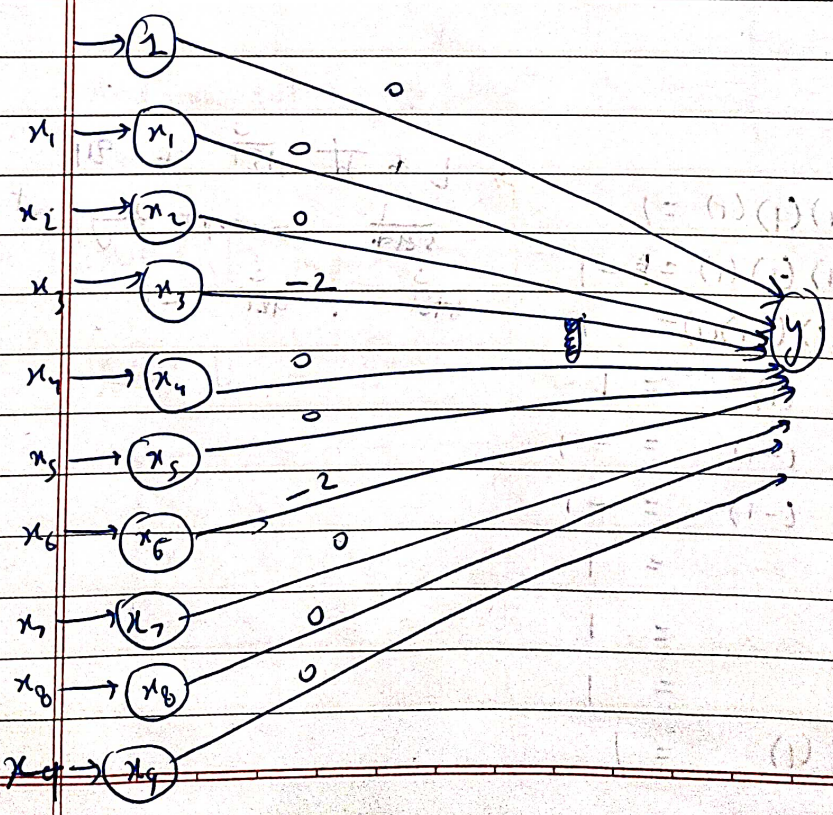
$w_{7\,new} = 1 - 1 = 0$

$w_{8\,new} = 1 - 1 = 0$

$w_{9\,new} = 1 - 1 = 0$

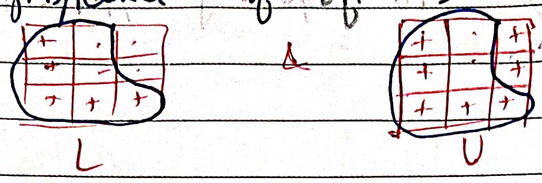
$b_{new} = 1 - 1 = 0$

1 epoch complete ✓



~~its not~~

* ~~Clustering~~
* significance of the initial 0 wts.



L & U ko itna part to same hi hoti isliye inn sbke wt.s 0 aaye after training for U.

* Clustering can ~~never~~ be supervised. Its ~~always~~ generally for unsup.

And its (not) like ki classificatⁿ is always sup. It can also be unsup.

* unsupervised learning → y given.
  - jo o/p y given hai usse learn ~~ko~~ krta hai

* Hebb is unsup. bcoz:
  1. jo i/p given hai uske as it is hum apne system ko train krte hain
  2. kahin pe net i/p calculate nhi krte hein
  3. we know whats the i/p and its class, the network has to just memorize it.
  4. Nobody is ~~watching~~ watching the n/w for errors

* In unsup, N/W is learning wt. by what is given to it and accordingly its finding o/p for an unknown pattern.

7 Feb 2022

PERCEPTRON

→ Assume initial set of Wt.s as 0

→ In Perceptron, we calculate ~~Add~~ Net i/p.
And we'll apply activation func^n on the Net i/p.
This is my actual o/p.
I'll compare this with ~~my~~ desired o/p.

① यदि No difference in them then no training req.

② If diff. is there, we'll do training, we'll update the wt.s

→ In diff from Hebb, here we calculate Net i/p.

→ ~~Except~~ Perceptron is supervised.
Bcoz wherever we see there is error (diff. b/w actual & desired o/p) ~~the~~ we tell the system that you are wrong (feeding error) and update your wt.s

→ All classifications that we do depend on Backpropagation.

→ In perceptron we use the activation function below

$$f(y_{in}) = \begin{cases} 1, & y_{in} > \theta \\ 0, & -\theta \le y_{in} \le \theta \\ -1, & y_{in} < -\theta \end{cases}$$

($f(y_{in})$: actual o/p)

→ In Perceptron, when we talk of logic gates, we take $\theta = 0$.

The above a.f. aisa hojayega

$$f(y_{in}) = \begin{cases} 1, & y_{in} > 0 \\ 0, & y_{in} = 0 \\ -1, & y_{in} < 0 \end{cases}$$

$\Rightarrow y = f(y_{in}) = y_{actual}$

(y : actual output)

→ t = desired o/p.

? y

→ when $y \neq t$,

$$w_{new} = w_{old} + \Delta w$$

$$\Delta w = t \times \alpha$$

$$b_{new} = b_{old} + \Delta b$$

$$t \times \alpha$$

bias का x is 1

$$t\alpha$$

Actually, ~~$\Delta w = \gamma \times \alpha$~~ $\Delta w = \gamma \times \alpha$

In perceptron, $\gamma = t$ (desired o/p)

b → weight ही है। = extra bias i/p का

So b के लिए हमेशा formula करने की जरूरत नहीं है।

→ Initially $w_1 = w_2 = w_3 = \dots = w_n = b = 0$.

→ If α not given, $\alpha = 1$

→ ~~Taken bki 2 classes → class 1 & class 2.~~

→ When only 2 classes are there, only 1 output neuron is required

* We will get ± range we can take $\theta = 0$ in Perceptron

→ If no. of classes > 2, then no. of o/p neurons required

→ If no. of epoch not given, 1 epoch ± range will

→ If α (not) given let it 1.

→ Perceptron is of Bipolar.

Q) AND gate using Perceptron

bias x_0	x_1	x_2	target t
1	1	1	+1
1	1	-1	-1
1	-1	1	-1
1	-1	-1	-1

(-1 & +1 or anything)

Initially $w_1 = w_2 = b = 0$

$\theta = 0$

$\alpha = 1$

Not given → 1 epoch

We will use a.f. $y = f(y_{in}) = \begin{cases} 1, & y_{in} > \theta \\ 0, & y_{in} = \theta \\ -1, & y_{in} < \theta \end{cases}$

~~Initially, $y_{in} = w_1 x_1 + w_2 x_2 + b x_0 = 0 + 0 + 0 = 0$~~

Start the training

★ Book se see Perceptron का flowchart (pdf Pg - 36)

Actually kisi bhi training pair को train karne se pehle, uss training pair ke liye hum y_{in} nikalenge → $f(y_{in})$ find, this is y.

Iss y को compare karenge with iss T.P. का t

★ If y != t, then train using this T.P.

And isse jo wts nikle that are presented to the next T.P.

If y == t, then kuch updation nhi hoga.

(Dono cases के लिए formulae given hai in flowchart)

~~Now kitte epochs~~

Now book के flow chart में end में इतनी दीली condit use ni karni. Mam ne ye btaya hai ye use karna hai.

1. For written exam, mam no. of epochs dengi hi, exactly utte epochs hi run karne hain.
2. For implementing in code/software, agar current epoch में (सारे) inputs पर wt. change huye hi nhi, then ye epoch complete karne ke baad stop.

Else current epoch karne ke baad next epoch start karna.

i.e. → अगर current epoch में सारे T.P.s ke liye y==t hojaye then kisi bhi wt. change nhi hoga compared to its prev., so ye last epoch hoga.

Bias hume khud se lene hota

Page No. 95
Date: / /

Q) AND gate using perceptron.

x_0	x_1	x_2	t
1	1	1	+1
1	1	−1	−1
1	−1	1	−1
1	−1	−1	−1

(−1 ~~ki~~ 0 smjo)

✓ Only for working table
Actually 0 ~~aur~~ ke hisaab se table likhte! then 0 replace → −1
~~Agar~~ Agar ki maths wali calculation me −1, −1 se shega

$b = w_1 = w_2 = 0$ (Initially)

$\alpha = 1$ (as not given)

Logic gate me ~~me~~ mein $\theta = 0$

① for i/p, $[x_0\ x_1\ x_2] = [1\ 1\ 1]$,

$$y_{in} = b + \sum_{i=1}^{2} w_i x_i = 0 + 0 = 0$$

$y = f(y_{in})$.

And we'll use activation func.,

$$y = f(y_{in}) = \begin{cases} 1, & y_{in} > 0 \\ 0, & y_{in} = 0 \\ -1, & y_{in} < 0 \end{cases}$$

Atleast ek bhi wt. change hua, to we'll say wts changed.

① for i/p $[x_0 \quad x_1 \quad x_2] = [1 \quad 1 \quad 1]$, $t = +1$

$$y_{in} = b + \sum_{i=1}^{2} w_i x_i = 0$$

$$y = f(y_{in}) = 0$$

$y \neq t \rightarrow$ train

$$w_{new} = w_{old} + t x \alpha = w_{old} + x$$

($t = 1$, $\alpha = 1$)

$b_{new} = 0 + 1 = 1$

$w_{1,new} = 0 + 1 = 1$

$w_{2,new} = 0 + 1 = 1$

~~wts changed → continue~~

compare from just previous wali w_1, w_2, b values.

⇒ These are presented to next training pair

② for i/p $[x_0 \quad x_1 \quad x_2] = [1 \quad 1 \quad -1]$, $t = -1$

$$y_{in} = b + \sum_{i=1}^{2} w_i x_i = 1 + 1(1) + 1(-1) = 1$$

$$y = f(y_{in}) = 1$$

$y \neq t \rightarrow$ train

③ $w_{new} = w_{old} + t x \alpha = w_{old} - x$

($t = -1$, $\alpha = 1$)

$b_{new} = 1 - 1 = 0$

$w_{1,new} = 1 - 1 = 0$

$w_{2,new} = 1 - (-1) = 2$

~~wts changed. Pehle b -1 था अब 0 हो गया~~ |
~~∴ continue~~

(3) For i/p $[x_0 \quad x_1 \quad x_2] = [1 \quad -1 \quad 1]$, $t = -1$

$$y_{in} = b + \sum_{i=1}^{n} w_i x_i = 0 + 0 + 2 = 2$$

$$y = f(y_{in}) = 1$$

$y != t \rightarrow$ train.

$$w_{new} = w_{old} + t x \alpha = w_{old} - x$$

($t = -1$, $\alpha = 1$)

$$b_{new} = 0 - 1 = -1$$
$$w_{1,new} = 0 - (-1) = 1$$
$$w_{2,new} = 2 - 1 = 1$$

~~wts changed → continue~~

(4) For i/p, $[x_0 \quad x_1 \quad x_2] = [1 \quad -1 \quad -1]$, $t = -1$

$$y_{in} = -1 + (-1) + (-1) = -3$$
$$y = f(-3) = -1$$

$y == t$ ✓ ~~but current epoch तो complete करनी है (hi train)~~

$$\boxed{w_{new} = w_{old}}$$

∴ no update करें। तो No need
(Perceptron का flowchart see)

$b_{new} = +1 - 1 = -2$

$w_{1,new} = 1 - (-1) = 2$

$w_{2,new} = 1 - (-1) = 2.$

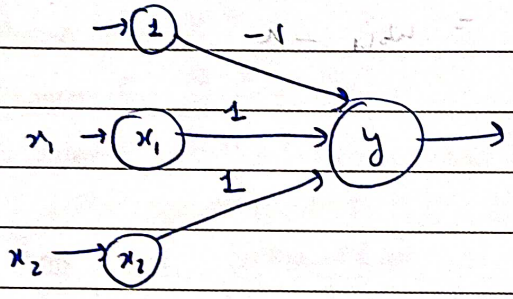
$b_{new} = b_{old} = -1$

$w_{1,new} = 1$

$w_{2,new} = 1$ ✓

1 epoch complete.

∴



Q) Type 2 ques → Pattern classification

Q) 4 inputs

Classify the following patterns

order of training (from top)

① (1, 1, 1, 1) ; ② (-1, 1, -1, -1) → target +1

③ (1, 1, 1, -1) ; ④ (-1, -1, -1, 1) → -1

$\alpha = 1$, $\theta = 0.2$

Run for 1 epoch.

Sol:) 4 inputs are ~~~~ given. Bias hume khud se lena hota.

∴ Total 5 inputs, x_0, x_1, x_2, x_3, x_4

Initially $b = w_1 = w_2 = w_3 = w_4 = 0$

	x_0	x_1	x_2	x_3	x_4	t
①	1	1	1	1	1	+1
②	1	−1	+1	−1	−1	+1
③	1	1	1	1	1	−1
④	1	1	−1	−1	1	−1

$\alpha = 1$, $\theta = 0.2$.

We'll use activation f^n, $y = f(y_{in}) = \begin{cases} +1, & y_{in} > 0.2 \\ 0, & -0.2 \le y_{in} \le 0.2 \\ -1, & y_{in} < -0.2 \end{cases}$

① For ① T.P, $t = +1$

$y_{in} = b + \sum_{i=1}^{4} w_i x_i = 0 + 0 = 0$

$y = 0$.

$y \neq t$ ⇒

$w_{new} = w_{old} + t x \alpha = w_{old} + x$

$b_{new} = 0 + 1 = 1$

$w_{1,new} = 0 + 1 = 1$

$w_{2,new} = \quad = 1$

$w_{3,new} = \quad = 1$

$w_{4,new} = \quad = 1$

② For T.P. ① , $t = +1$

$y_{in} = b + \sum_{i=1}^{4} w_i x_i = 1 + (-1) + (1) + (-1) + (-1)$

$= -1$

$y = f(y_{in}) = -1$

$y != t$

$w_{new} = w_{old} + t x \alpha \quad = w_{old} + x$

$\alpha = 1$

$b_{new} = 1 + 1 = 2$

$w_{1,new} = 1 + (-1) = 0$

$w_{2,new} = 1 + (1) = 2$

$w_{3,new} = 1 + (-1) = 0$

$w_{4,new} = 1 + (-1) = 0$

③ For 3rd T.P, $t = -1$

$y_{in} = b + \sum_{i=1}^{4} w_i x_i = 2 + 0 + 2(1) + 0 + 0 = 4$

$y = 1$

$y != t$

$w_{new} = w_{old} + t x \alpha \quad = w_{old} - x$

$\alpha = 1$

$b_{new} = 2 - 1 = 1$

$w_{1,new} = 0 - 1 = -1$

$w_{2,new} = 2 - 1 = 1$

$w_{3,new} = 0 - 1 = -1$

$w_{4,new} = 0 - (-1) = 1$

④ for 4th i.p, t = -1

$y_{in} = b + \sum_{i=1}^{4} w_i x_i = 1 + (-1)(1) + (1)(-1) + (-1)(-1) + (1)(1)$

$= 1$

$y \neq t$

$w_{new} = w_{old} + t x \alpha \quad (t = -1,\ x = y_1)$

$= w_{old} - x$

$\therefore b_{new} = 1 - 1 = 0$

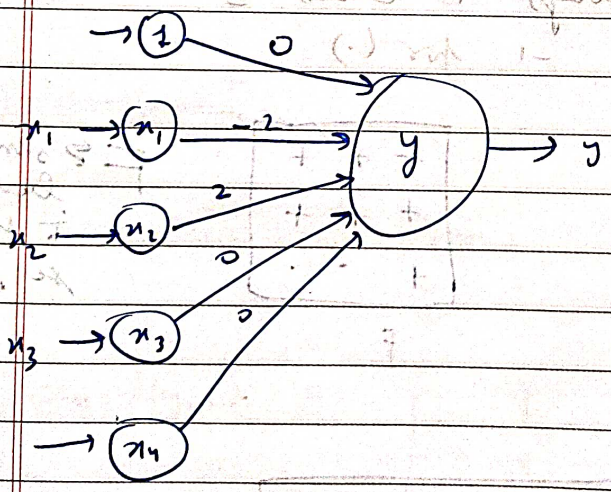
$w_{1,new} = -1 - 1 = -2$

$w_{2,new} = 1 - (-1) = 2$ ✓

$w_{3,new} = -1 - (-1) = 0$ ✓

$w_{4,new} = 1 - 1 = 0$ ✓

1 epoch complete



data linearly separable ??

As 2 classes ⇒ only 1 output neuron required.

Alt) This is alternate way (not for exam) (Just to ~~check~~ verify ans quickly)

Initially $b = w_1 = w_2 = w_3 = w_4 = 0$

x_0	x_1	x_2	x_3	x_4	t
1	1	1	1	1	+1
1	-1	+1	-1	-1	+1
1	1	1	1	-1	-1
1	1	-1	-1	1	-1

9 Feb 2022

8) Apply perceptron learning rule to classify pattern 'T' & 'f' in a 3×3 matrix where 'T' belongs to class 1 & 'f' belongs to class 2.

i/p = +1 for (+) & -1 for (·)

+	+	+
·	+	·
·	+	·

T

+	+	+
+	+	+
+	·	·

F

To symbol (+1) input ke liye hai usse symbol banate

F ko same length ka bnaya se karne

∴ ~~Total~~ 9 i/ps above

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

And an extra i/p x_0 for bias

~~Exception~~ If θ (not) given, take it 0.

∴ Total 10 i/p's.

T belongs to class 1 ⇒ o/p +1 ⇒ y = +1

F " " " 2 ⇒ " -1 ⇒ y = -1

Pehle 'T' ki train then 'F' ki train

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	t
T	1	1	1	1	-1	1	-1	-1	1	-1	+1
F	1	1	1	1	1	1	+1	1	-1	-1	-1

2 hi i/p's only so 2 ki train.

Initially $b = w_1 = w_2 = \dots = w_9 = 0$

And $\alpha = 1$ (assume)

~~1) for i/p I~~,

Also, ~~θ (not) given~~, take $\theta = 0$.

$$\therefore y = f(y_{in}) = \begin{cases} 1, & y_{in} > 0 \\ 0, & y_{in} = 0 \\ -1, & y_{in} < 0 \end{cases}$$

1) For i/p T, $t = +1$

$$y_{in} = b + \sum_{i=1}^{9} w_i x_i = 0$$

$$y = f(y_{in}) = 0$$

$y \neq t$

$$\therefore w_{new} = w_{old} + t \times \alpha \times x = w_{old} + x$$

~~$w_{1,new} = w_{1,old} + x_1 = 0+1$~~

$b_{new} = b_{old} + x_0 = 0 + 1 = 1$

$w_{1,new} = w_{1,old} + x_1 = 0 + 1 = 1$

$w_{2,new} = 1$

$w_{3,new} = 1$

$w_{4,new} = -1$

$w_{5,new} = 1$

$w_{6,new} =$

$w_{7,new} = -1$

$w_{8,new} = 1$

$w_{9,new} = -1$

Now these are presented to next FP.

2) for i/p f, $t = -1$

$$y_{in} = b + \sum_{i=1}^{9} w_i x_i = 1 + (1\times1) + (1\times1) + (1\times1) + (-1\times1) + (1\times1) + (-1\times-1) + (-1\times1) + (1\times-1) + (-1\times-1)$$

$$= 1+1+1+1-1+1+1-1-1+1$$

$$= 4$$

$y = f(y_{in}) = f(4) = 1$

$y != t$

$w_{new} = w_{old} + t \times \alpha = w_{old} - x$.

$t = -1, 1$

$\alpha = 1$

$w_{1,new} =$

$b_{new} = b_{old} - x_0 = 1 - 1 = 0$

$w_{1,new} = = 0$

$w_{2,new} = = 0$

$w_{3,new} = = 0$

$w_{4,new} = = -2$

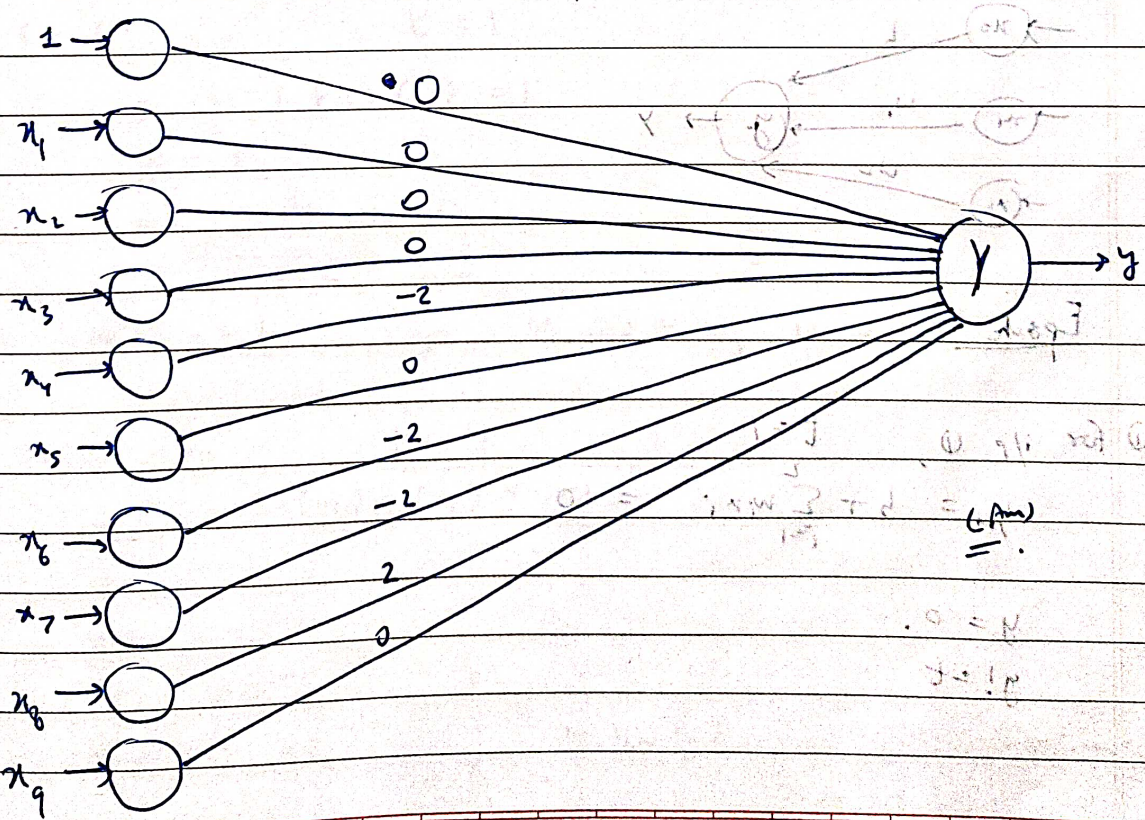
$w_{5,new} = = 0$

$w_{6,new} = = 0 - 2$

$w_{7,new} = = -2$

$w_{8,new} = = 2$

$w_{9,new} = = 0$



10 Feb 2022

Can we write in table format (Pg 83 book table 4)

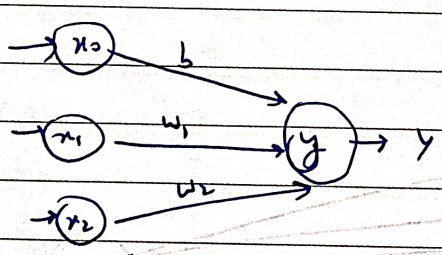
Q) Find weights after training logical OR gate for 2 epochs & using perceptron learning rule

A)

	x_0	x_1	x_2	t
①	1	1	1	1
②	1	-1	1	1
③	1	1	-1	1
④	1	-1	-1	-1

Initially, $b = w_1 = w_2 = 0$
$\alpha = 1$
$\theta = 0$
} Assume when not given in ques.

We'll use, $y = f(y_{in}) = \begin{cases} 1, & y_{in} > \theta \\ 0, & y_{in} = \theta \\ -1, & y_{in} < \theta \end{cases}$



Epoch 1

① for i/p ①, t = 1

$$y_{in} = b + \sum_{i=1}^{2} w_i x_i = 0$$

$y = 0$.

$y \neq t$

$w_{new} = w_{old} + t \times x \quad = w_{old} + x$

$b_{new} = 0 + 1 = 1$ ✓

$w_{1,new} = 0 + 1 = 1$ ✓

$w_{2,new} = 0 + 1 = 1$ ✓

② For ② i/p, $t = 1$

$y_{in} = b + \sum_{i=1}^{2} w_i x_i = 1 + (-1) + 1 = 1$

$y = f(y_{in}) = 1$

$y = t \rightarrow$ No updation. ✓

$b_{new} = b_{old} = 1$

$w_{1,new} = w_{1,old} = 1$

$w_{2,new} = 1$

③ For ③ i/p, $t = 1$

$y_{in} = 1 + 1 + (-1) = 1$

$y = 1$

$y = t \rightarrow$ No update

$\therefore b_{new} = 1, \quad w_{1,new} = 1, \quad w_{2,new} = 1$

④ For ④ i/p, $t = -1$

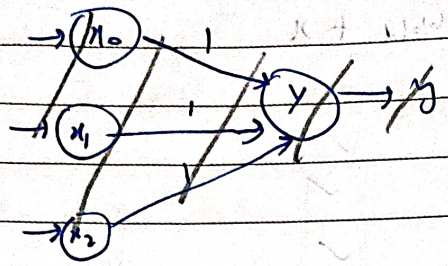
$y_{in} = 1 + (-1) + (-1) = -1$

$y = -1$

$y = t$

No update

$b_{new} = 1, \quad w_{1,new} = 1, \quad w_{2,new} = 1$



EPOCH 2

① For ① i/p, $t = 1$

$y_{in} = 1 + 1 + 1 = 3$

$y = 1$

$y = t \rightarrow$ No updat

$b_{new} = 1$, $w_{1,new} = 1$, $w_{2,new} = 1$,

② For ② i/p, $t = 1$

$y_{in} = 1 + (-1) + 1 = 1$

$y = 1$

$y = t \rightarrow$ No updat

$b_{new} = 1$, $w_{1,new} = 1$, $w_{2,new} = 1$

③ For ③ i/p, $t = 1$

$y_{in} = 1 + 1 + (-1) = 1$

$y = 1$

$y = t \rightarrow$ No updat

$b_{new} = 1$, $w_{1,new} = 1$, $w_{2,new} = 1$

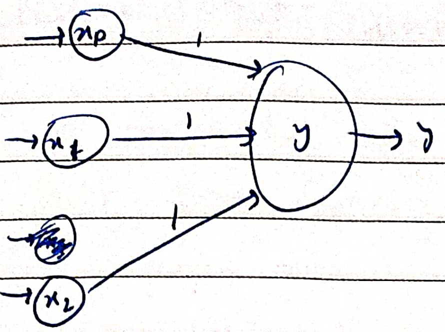
④ For ④ i/p $t = -1$

$y_{in} = 1 + (-1) + (-1) = -1$

$y = -1$

$t = y \rightarrow$ No updat

$b_{new} = 1$, $w_{1,new} = 1$, $w_{2,new} = 1$



(Ans)

* Book में ek table bnakr solve kiye hain aise wts nikalne ke ques. Jisme x_0, x_1, x_2 - - - t, yin, y, Δb, Δw_1, Δw_2 - - -, b, w_1, w_2 - - - ye saare columns hote.
But mam ne kaha table se mat krna in exam. Aise hi krna jaise mai copy me kra chahe 1 epoch ya zyada epoch.

* Say given, 10 epochs करो।
अगर 3^{rd} epoch में सारे i/ps (T.P.s) के में | $\Delta w = 0$ (सारे) आए (i.e. कोई भी wt. change ni hua), then aage wale epochs me bhi koi change ni hoga
So 3^{rd} epoch के end का Answer = 10^{th} epoch के end का Answer

Isliye 3^{rd} epoch पर stop. And iska answer is final answer.

Ye concept offline written exam me bhi use karna hai

Q) ~~Feel~~

* solve → numerical.
  Implement → coding.

Q) Implement any logic gate using Hebb learning rule.
↳ coding

kaise bhi kr skte. table bnakr ya lengthy solution

Hebb में No threshold

Hebb में y & t को compare करना नहीं होता को each Epoch में

$w_{new} = w_{old} + y x \alpha$ वाला formula

use hoga
तो each epoch में weights & bias update honge
hi honge. में
तो Hebb / no. of epochs given honge.
तो जो था कि hare i/p पर change जो है then
वो epoch last होगा that's of Perceptron

* Hebb is unsupervised. Hebb में basically जो data wo yaad krta jaata, Iska test ni le skte.

Perceptron but supervised hai, iska test de skte

⇒ α can never be any 0. α is learning rate. अगर α 0 होगया then learning stop. $\therefore \alpha \neq 0$.

15 Feb 2022

ADALINE N/W

→ Delta learning rule ← another name

→ Adaline n/w = Delta learning rule
= Gradient Descent learning rule
= Widrow Hoff " "

→ Why called Delta L.R.?
→ " " Gradient Descent?

→ In perceptron we were checking only that y & t are same or not.
t = y → no updt
t ! y → updt

We were not checking how close t & y are there.
Also we were always initialize with weights & bias 0.
What if kahin aur se

→ In Delta learning Rule, we can initialize weights & bias by any value other than 0.

MidSEM

Biological , Sigmoidal → Sivanandam

Non-sigmoidal → see on net. (GfG)
↳ ReLU , SiLU . Also see

Learning rate →

Wt. update

Bias

Epoch , MCP , Linear separability , Hebb , Perceptron

Read theory also from book. Flowchart & steps to form से भी पूछे

Also practise numerical

Class से ही Numerical हैं वैसे ही Paper से Ques आएंगे

Both theory & paper

→ Agar numerical diya then no need to write Algo or flowchart.

→ Different Ques से match columns.

→