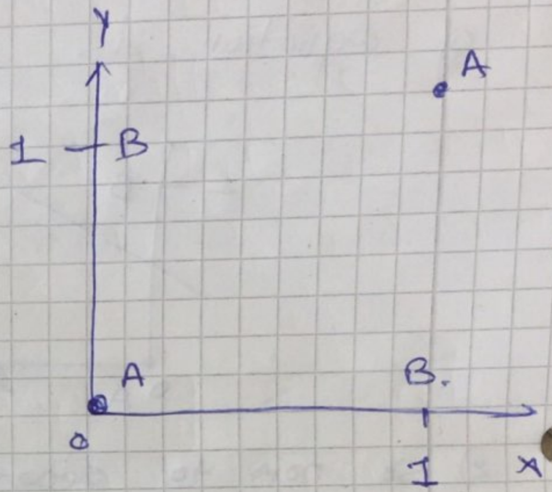
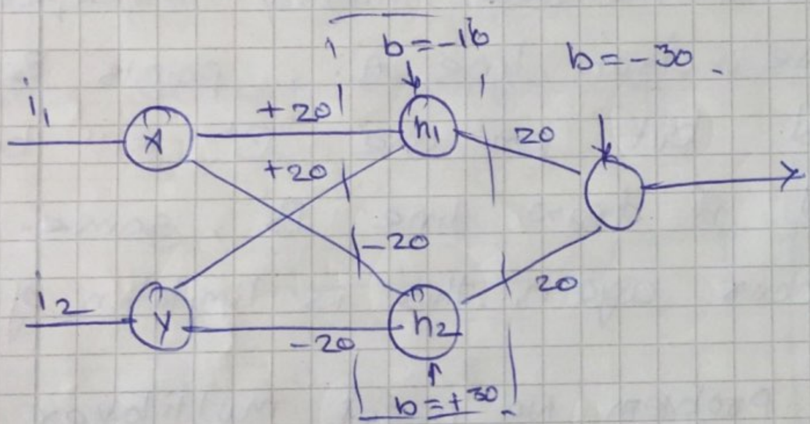
* Multilayer perceptron.

1) Problem regarding single layer perceptron can be overcomed by MLP

2) lets consider same example of XOR to find o/p of XOR.

x	y	XOR (o/p)
0	0	0
0	1	1
1	0	1
1	1	0.



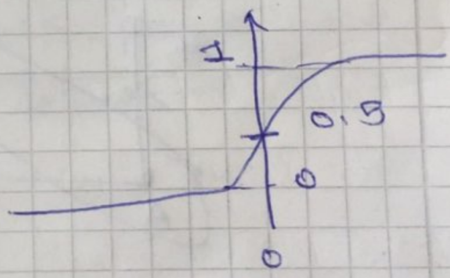


Hidden layer.

if Calculate for h_1 & h_2.

h_1 = Sigmoid ($20x + 20y - 10$)

h_2 = Sigmoid ($-20x - 20y + 30$)



3) Now if we take center of sigmoid ie as threshold of hi and h_2 and ~~put~~ compar abov equation with it it will be like thf

h_1 = Sigmoid ($20x + 20y - 10$) = 0.5

h_2 = Sigmoid ($-20x - 20y + 30$) = 0.5

Scanned with CamScanner

if we solve for. y value

$20x + 20y - 10 = 0.5$

$20(x+y) = 10.5$

$x + y = \frac{10.5}{20}$

$h_1 = \boxed{y \approx 0.5 - x}$

~~for~~

$-20x - 20y + 30 = 0.5$

$-20(x+y) = -29.5$

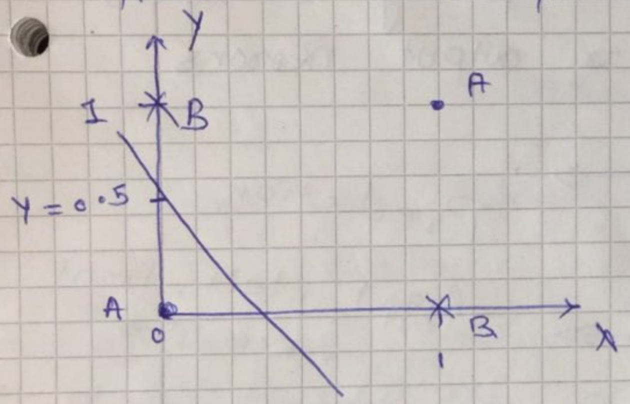
$x + y = +\frac{29.5}{20}$

$h_2 = \boxed{y \approx 1.5 - x}$

(4) if we plot h_1 ~~and~~ h_2 on graph.

$h_1 \approx y \approx 0.5 - x$

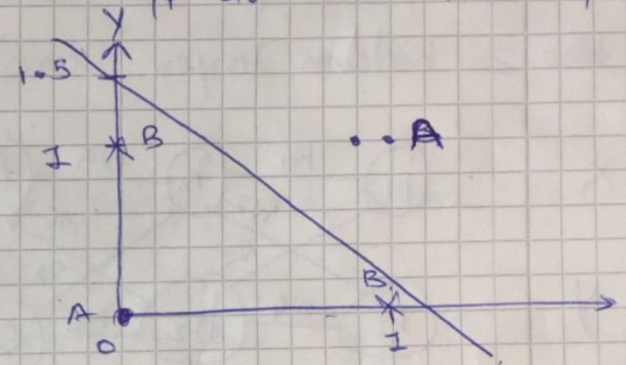
o/p due to hidden layer '1'



$y = 0.5 - x$

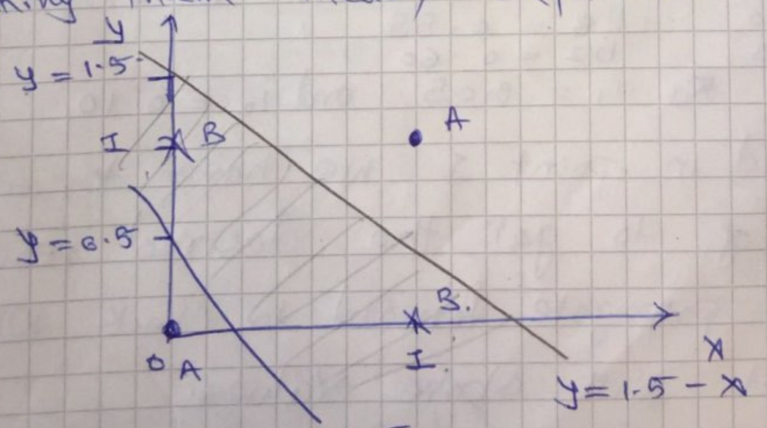
$h_2 = y \approx 1.5 - x$

o/p due to hidden layer '2'



unit 1

(5) so o/p of hidden layer 1 and hiddeden layer unit 2 are given to final o/p neuron that high lights or make the points 'B' towards 1, therefor Combined result can be plotted as points 'B' more towards '1' at o/p & 'A' as a Making them linearly Seperable.



$y = 1.5 - x$ by h_2

$y = 0.5 - x$ by h_1

final o/p

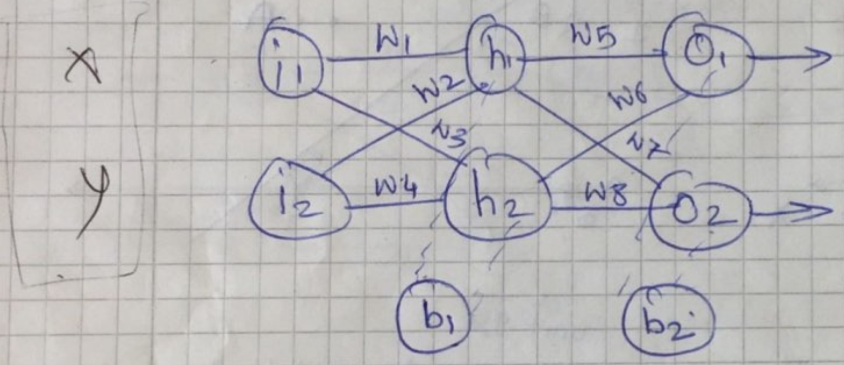
x	y	train expecte	actua
0	0	0	0.004
0	1	1	0.998
1	0	1	0.997
1	1	0	0.002

hebbian theory – Neurons that fire together wire together.

Back propogation.

① Back propogation basically is nothing but adjusting the weights in NN according to difference between actual output and desired output, we will try to look by taking an example to understand back propogation.

② In this example we take example of 2 inputs, 2 hidden layer, and 2 output neurons



Prediction.

Right, left, front back, sto.

③ Here lets assume the input, weights and the expected outputs

$w_1 = 0.15$	$w_5 = 0.40$	expected output
$w_2 = 0.20$	$w_6 = 0.45$	$O_1 = 0.01 \approx 0$
$w_3 = 0.25$	$w_7 = 0.50$	$O_2 = 0.99 \approx 1$
$w_4 = 0.30$	$w_8 = 0.55$	$b_1 =$
$b_1 = 0.35$	$b_2 = 0.60$	

inputs :– $i_1 = 0.05$ and $i_2 = 0.10$.

④ As we said in point 1 we have to adjust the weight of to get the desired result at output, lets propogate forward to check what o/p we get with above values.

(a) The forward pass:

i) first we calculate output of hidden layer neurons h_1 and h_2.

$$net\ h_1 = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$

$$= 0.15 \times 0.05 + 0.20 \times 0.10 + 0.35 \times 1$$

$$= \cancel{0.0075}\ 0.0075 + 0.02 + 0.35$$

$$= 0.3775$$

~~out h_1~~ = lets consider output as sigmoid function.

$$out\ h_1 = \frac{1}{1+e^{-net h_1}} = \frac{1}{1+e^{-(0.3775)}}$$

$$= 0.5932.$$

Same we calculate for net_2 and $out h_2$.

$$\therefore out\ h_2 = 0.596884.$$

ii) Now we continue this process for calculating the output of o_1 and o_2.

$$\therefore net_{o_1} = w_5 \times out\ h_1 + w_6 \times out\ h_2 + b_2 * 1$$

$$= 0.40 \times 0.5932 + 0.45 \times 0.59688 + 0.60$$

$$= 1.105906$$

$$\therefore out_{o_1} = \frac{1}{1+e^{-net_{o_1}}} = \frac{1}{1+e^{-(1.105906)}} = 0.75136$$

Similarly

$$out_{o_2} = \frac{1}{1+e^{-net_{o_2}}} = 0.772928.$$

A	B	C	Y	
0	0	1	0	1
1	1	0	1	0
1	0	1	1	1
0	1	1	0	0
1	0	0	1	
0	1	0	0	

③ Calculating the total error:

total error is nothing but target - output square.

$$E_{total} = \frac{1}{2} \sum (target - output)^2$$

the target output for O_1 is 0.01 but as seen ~~E_1~~ $out_{O_1} = 0.75136$.

$$\therefore E_{O_1} = \frac{1}{2}(target_{O_1} - out_{O_1})^2$$

$$= \frac{1}{2}(0.01 - 0.75136)^2$$

$$= 0.2748.$$

Repeating the process for E_{O_2}

$$\therefore E_{O_2} = \underline{0.02356026}$$

$\therefore$ total error $= E_{O_1} + E_{O_2} = 0.2748 + 0.023 = 0.2983$

④ Now that we know the total error, based on that we can adjust the weights so that we get closer to actual output.

<u>output layer.</u>

~~So basica~~ lets consider example of weight w_5 we need to calculate how much change w_5 affects total error E_{total}

i.e $\frac{\partial E_{total}}{\partial w_5}$

to calculate above equation we need chain rule.

$$\therefore \frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w_5}$$

Chain rule.

Q) Calculating individual equations and then multiplying the result to get $\frac{\partial E_{total}}{\partial w_5}$

1) $\frac{\partial E_{total}}{\partial out_{o1}}$ - this is nothing but how much total error varies w.r.t out_{o1}.

We know ~~$E_{total} = \frac{1}{2}(target_{o1} - o$~~

$$E_{total} = \frac{1}{2}(target_{o1} - out_{o1})^2 + \underbrace{\frac{1}{2}(target_{o2} - out_{o2})^2}_{0}$$

$$\therefore \frac{\partial E_{total}}{\partial out_{o1}} = 2 \cdot \frac{1}{2}(target_{o1} - out_{o1})^{(2-1)} \times (-1) + 0$$

$$= -(target_{o1} - out_{o1})$$

$$= -(0.01 - 0.75136)$$

$$= 0.74136.$$

2) $\frac{\partial out_{o1}}{\partial net_{o1}}$ We know $out_{o1} = \frac{1}{1+e^{-net_{o1}}}$

derivative of sigmoid

$$\frac{\partial out_{o1}}{\partial net_{o1}} = out_{o1}(1 - out_{o1})$$

$$= 0.75135(1 - 0.75135)$$

$$= 0.1868156.$$

3) $\frac{\partial net_{o1}}{\partial w_5}$ $net_{o1} = w_5 \times out_{h1} + w_6 \times out_{h2} + b_2 \times 1$

$$\therefore \frac{d}{dx}(a \cdot b) = a \cdot \frac{d}{dx} b + b \cdot \frac{d}{dx} a. \text{ — Product rule.}$$

$$\frac{d(w_5 \cdot out_{h1})}{\partial w_5} = w_5 \frac{d}{\partial w_5} out_{h1} + out_{h1} \frac{d}{\partial w_5} w_5$$

$$= 0 + out_{h1} \cdot 1$$

$$\therefore \frac{\partial net}{\partial w_5} = out_{h1} = 0.5932.$$

Scanned with CamScanner

Putting it all together.

$$\frac{\partial E_{Total}}{\partial w_5} = \frac{\partial E_{total}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w_5}$$

$$= 0.74136 \times 0.18681 \times 0.5932$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.082167$$

(5) To decrease the weight we subtract this value from the current weight.

$$\boxed{\therefore w_5 = w_5 - \eta \cdot \frac{\partial E_{total}}{\partial w_5}}$$

η – learning rate, lets assume it to be 0.5

$0.01 \leq \eta \leq 0.9$

$$\therefore w_5^+ = w_5 - \eta \cdot \frac{\partial E_{total}}{\partial w_5}$$

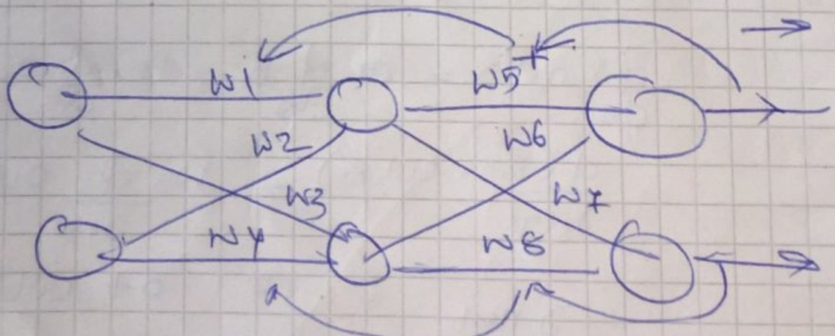
$$= 0.4 - 0.5 \times 0.0821$$

$$\boxed{w_5^+ = 0.3589}$$

(6) We can repeat it for w_6^+, w_7^+, w_8^+

next we'll continue the backward pass by calculating new values for w_1, w_2, w_3, w_4

i.e $\frac{E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} \times \frac{\partial out_{h1}}{\partial net_{h1}} \times \frac{\partial net_{h1}}{\partial w_1}$



→ Back Progate to update weight to bring output closer to desire output.