Q.1. Deep Learning Assign 2     Yash Patel, Gaurav Maheshwari
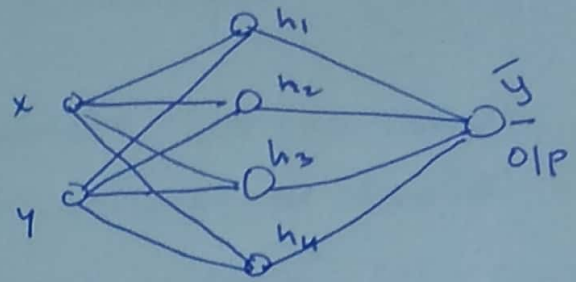
Now, we have to first get the eq^n's of lines (segments)

$$\overline{PQ} = x - 3y - 2 = 0. \quad (h_1)$$

$$\overline{PK} = 2x - y + 6 = 0 \quad (h_2)$$

$$\overline{RK} = x - 2y + 6 = 0. \quad (h_3)$$

$$\overline{QR} = x + 3y - 14 = 0. \quad (h_4).$$



Now, we are given as inputs $(x, y)$, in order to design a NN, each hidden layer has 4 perceptrons and o/p layer has only one perceptron.

Also, each perceptron in the hidden layer models one line segment of the trapezoid. And we can do AND/OR accordingly to classify the pts in class A & class B.

Hence, the o/p layer perceptron is AND/OR gate

Now, as we have designed the NN, we need the wts & bias. to calculate

$$h_1 = \left( [x \quad y] \begin{bmatrix} 1 \\ -3 \end{bmatrix} \right) - 2. \rightarrow bias$$
→ wts

$$h_2 = \left( [x \quad y] \begin{bmatrix} 2 \\ -1 \end{bmatrix} \right) + 6. \rightarrow bias$$
→ wts

$$h_3 = \left( [x \quad y] \begin{bmatrix} 1 \\ -2 \end{bmatrix} \right) + 6 \rightarrow bias$$
→ wts

$$h_4 = \left( [x \quad y] \begin{bmatrix} 1 \\ +3 \end{bmatrix} \right) - 14. \rightarrow bias$$
→ wts
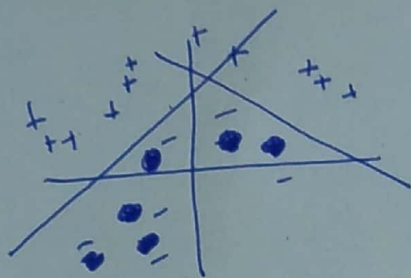
Activatn function : Sgn(x). i.e.

$$f(x) = \begin{cases} -1 & x < 0 \\ +1 & x > 0 \\ 0 & x \to 0. \end{cases}$$
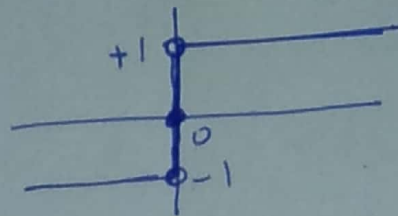
→ Using Sgn(x) as activatn functn, we can classify easily whether a pt. $(x, y)$ is in class (A) or class (B).

→ Hence, we can implement this logic with an AND gate



Output layer wts $[1, -1, -1, -1]$ and bias 2.

Output Classes; if $\bar{y} = $ output $ = 1$, then class B

$\bar{y} = -1$ , then class A.

## Q.2

(a) Following equations are used:

$S(x) = $ sigmoid $(x)$

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

$h_1 = S(w_1 x_1 + w_2 x_2 + b_1) = 0.596$

$h_2 = S(w_3 x_1 + w_4 x_2 + b_1) = 0.608$

$S'(x) = S(x)(1 - S(x))$

$o_1 = S(w_5 h_1 + w_6 h_2 + b_2) = 0.758$

$o_2 = S(w_7 h_1 + w_8 h_2 + b_2) = 0.779$

Now, we know that total error is used as mean squared error. Hence,

Error $E = \frac{1}{2}\left((o_1 - y_1)^2 + (o_2 - y_2)^2\right)$.

$$= \frac{1}{2} \times 0.446 = \boxed{0.2238}$$

(b) Now, for back propagation we need to update the wt's & bias by doing gradient descent and then update the wt's & bias by this formula's.

$$\text{new\_weight} = \text{old\_weight} - \underset{0.5}{\alpha} \cdot \underbrace{\text{grad\_descent}}_{\substack{\text{w.r.t.w}}} = w$$

$$\text{new\_bias} = \text{old\_bias} - \underset{0.5}{\alpha} \cdot \underbrace{\text{grad\_descent\_b}}_{\text{w.r.t. b}}$$

In order to update all the wt's, we need to differentiate Error "E" w.r.t. all the wt's. Also as most of the terms in the differentiation are common, I will write in the following manner.

$$\frac{\delta E}{\delta w_1} = \frac{\delta E}{\delta h_1} \times \frac{\delta h_1}{\delta w_1} \qquad \& \qquad \frac{\delta E}{\delta w_2} = \frac{\delta E}{\delta h_1} \times \frac{\delta h_1}{\delta w_2}.$$

$$\frac{\delta E}{\delta h_1} = \underbrace{(0_1 - y_1)\, 0_1\,(1-0_1)\,\omega_5}_{0.120} + \underbrace{(0_2 - y_2)\, 0_2\,(1-0_2)\,\omega_7}_{-0.020} \quad \boxed{@} \;\; \overset{\|}{\underset{0.038}{}}$$

$$\frac{\delta h_1}{\delta \omega_1} = \underbrace{h_1(1-h_1)\,x_1}_{0.240} = 0.024 \text{ in} \qquad \frac{\delta h_1}{\delta \omega_2} = \underbrace{h_1(1-h_1)\,x_2}_{0.240} = 0.096.$$

Here only $x_1$ & $x_2$ diffe in $\dfrac{\delta h_1}{\delta \omega_1}$ & $\dfrac{\delta h_2}{\delta \omega_2}$.

So I will write

$$\frac{\delta}{\delta \omega_{1,2}} \overset{h_1}{} = h_1(1-h_1)\,\omega_{1,2} \quad - \quad \boxed{\cancel{6}}$$

$\therefore$  $\dfrac{\delta E}{\delta \omega_{1}} = \underbrace{@}_{0.038} \times \cancel{6} \times 0.024.$  $\qquad \dfrac{\delta E}{\delta \omega_2} = @ \times 0.096.$

$$= 3.648 \times 10^{-3}.$$

$\therefore$  $\dfrac{\delta E}{\delta \omega_1} = 9.12 \times 10^{-4}.$

$$\frac{\delta E}{\delta h_2} = \underbrace{(0_1 - y_1)\, 0_1\,(1-0_1)\,\omega_6}_{0.120} + \underbrace{(0_2 - y_2)\, 0_2\,(1-0_2)\,\omega_8}_{-0.020} \overset{\textcircled{6}}{=} 0.048$$

$$\frac{\partial h_2}{\partial \omega_3} = \underbrace{h_2(1-h_2)\,x_1}_{0.238} = 0.023 \text{ in} \qquad \frac{\partial h_2}{\partial \omega_4} = \underbrace{h_2(1-h_2)\,x_2}_{0.238} = 0.095$$

$$\frac{\delta E}{\delta \omega_3} = \underbrace{\textcircled{6}}_{0.048} \times 0.023. \qquad \frac{\delta E}{\delta \omega_4} = \textcircled{6} \times 0.095$$

$$= 1.104 \times 10^{-3} \qquad\qquad\qquad = 4.56 \times 10^{-3}$$

$$\frac{\partial E}{\partial \omega_5} = \underbrace{(0_1 - y_1)\, 0_1\,(1-0_1)}_{0.120}\, \underbrace{h_1}_{0.896} = 0.071$$

$$\frac{\partial E}{\partial \omega_6} = \underbrace{(0_1 - y_1)\, 0_1\,(1-0_1)}_{0.120}\, \underbrace{h_2}_{0.608} = 0.072$$

$$\frac{\partial E}{\partial w_7} = \underbrace{(O_2 - Y_2) O_2 (1 - O_2)}_{-0.020} h_1 = -0.011$$

$$\frac{\partial E}{\partial w_8} = \underbrace{(O_2 - Y_2) O_2 (1 - O_2)}_{-0.020} h_2 = -0.012$$

Now, we will update the wts accordingly

$$\text{new\_}w_1 = w_1 - \alpha \cdot \frac{\partial E}{\partial w_1}$$

$$= 0.1 - 0.5 \times 9.12 \times 10^{-4}.$$

$$= 0.099544$$

$$\text{new\_}w_2 = w_2 - \alpha \cdot \frac{\partial E}{\partial w_2}$$

$$= 0.2 - 0.5 \times 3.648 \times 10^{-3} = 0.198176$$

$$\text{new\_}w_3 = w_3 - \alpha \cdot \frac{\partial E}{\partial w_3} = 0.19948$$

$$\text{new\_}w_4 = w_4 - \alpha \cdot \frac{\partial E}{\partial w_4} = 0.29772.$$

$$\text{new\_}w_5 = w_5 - \alpha \frac{\partial E}{\partial w_5} = 0.3645$$

$$\text{new\_}w_6 = w_6 - \alpha \frac{\partial E}{\partial w_6} = 0.464$$

$$\text{new\_}w_7 = w_7 - \alpha \cdot \frac{\partial E}{\partial w_7} = 0.5055$$

$$\text{new\_}w_8 = w_8 - \alpha \cdot \frac{\partial E}{\partial w_8} = 0.606.$$

Now feed forward analysis with new wt's.

$$new\_h_1 = \cancel{S(0.4+9994)} \cancel{- 0.666} 0.596105952$$

$$new\_h_2 = \cancel{S(0.4+8.9076)} \cancel{- 0.447} 0.608029304.$$

$$new\_0_1 = 0.750148831$$

$$new\_0_2 = 0.780708049.$$

Now, new $\_$ Error = 0.2184.

Yess, the error after 1st iteration of back propogation has ~~increased~~ decreased !!.