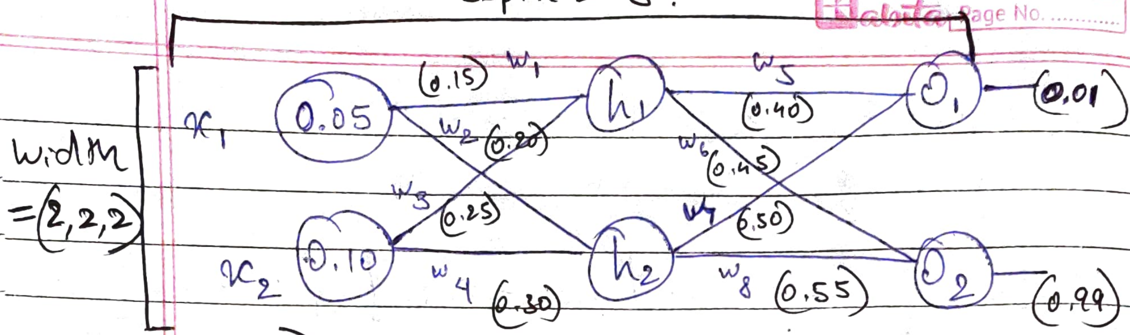
Lab-2 (a). Deep Neural Network Prediction Model

Depth = 3.



Ist layer (I/p to h)

$b_1 = 0.35$ $\qquad$ $b_2 = 0.60$

$$h_1(in) = w_1 \cdot x_1 + w_3 x_2 + b_1$$
$$= (0.15 \times 0.05) + (0.25 \times 0.10) + 0.35$$
$$= 0.377$$

$$h_2(in) = w_2 x_1 + w_4 x_2 + b_1$$
$$= (0.2 \times 0.05) + (0.3 \times 0.10) + 0.35$$
$$= 0.39.$$

Matrix represent.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cdot \begin{bmatrix} w_1 & w_3 \\ w_2 & w_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_1 \end{bmatrix}$$

$$= \begin{bmatrix} x_1 w_1 + w_3 x_2 + b_1 \\ x_1 w_2 + x_2 w_4 + b_1 \end{bmatrix} \xrightarrow[\text{(Sigmoid)}]{\text{Activation}} \begin{bmatrix} h_{1\,out} \\ h_{2\,out} \end{bmatrix}$$

$$h_1(out) = \frac{1}{1 + e^{-h_1(in)}} = 0.5932 \quad \text{(sigmoid)}$$

$$h_2(out) = \frac{1}{1 + e^{-h_2(in)}} = 0.5968. \quad \text{(Sigmoid)}$$

2nd layer (h to o/p).

$O \rightarrow$ actual output.
$\hat{O} \rightarrow$ Output calculated

$$O_1(in) = w_5 \, h_1(out) + w_7 \, h_2(out) + b_2$$

Kabita Date .... Page No. ....

$= (0.4 \times 0.593 + 0.45 \times 0.596 + 0.6)$

$= 1.105$

$\hat{O}_{1(out)} = \dfrac{1}{1 + e^{-O_{1(in)}}} = 0.7513$

$O_{2(in)} = w_6 \, h_{1(out)} + h_{2(out)} \, w_8 + b_2$

$= (0.45 \times 0.593) + (0.596 \times 0.55) + \underset{0.60}{b_2}$

$= 1.1946$

$\hat{O}_{2(out)} = \dfrac{1}{1 + e^{-O_{2(in)}}} = 0.7729.$

~~Matrix~~ Representation

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \begin{bmatrix} w_5 & w_7 \\ w_6 & w_8 \end{bmatrix} + \begin{bmatrix} b_2 \\ b_2 \end{bmatrix}$$

$$= \begin{bmatrix} w_5 h_{1(out)} + w_7 h_{2(out)} + b_2 \\ w_6 h_{1(out)} + w_8 h_{2(out)} + b_2 \end{bmatrix} \xrightarrow[\text{(Sigmoid)}]{\text{Activation}} \begin{bmatrix} O_{1(out)} \\ O_{2(out)} \end{bmatrix}$$

* loss functn = Error $= \sum \frac{1}{2} (\underset{\text{target}}{O} - \underset{\text{calculated}}{\hat{O}})^2$ ~~actual~~

$E\hat{O}_1 = 0.274$ $\qquad E O_2 = 0.0235.$

$E\,total = E O_1 + E O_2$

$= \underline{0.2983}.$

* learning of the neural network takes place on the basis of a sample of

the population under study. During the course of learning, we compare the value delivered by the output unit with actual value. After that adjust the weights of all units so to improve the prediction.

We can use gradient descent as an learning algorithm — which is used to find the local minimum of a function. Then we can adjust the weights ($w_1 - w_8$) so as to reduce the loss/error function to minimum ≈ 0.