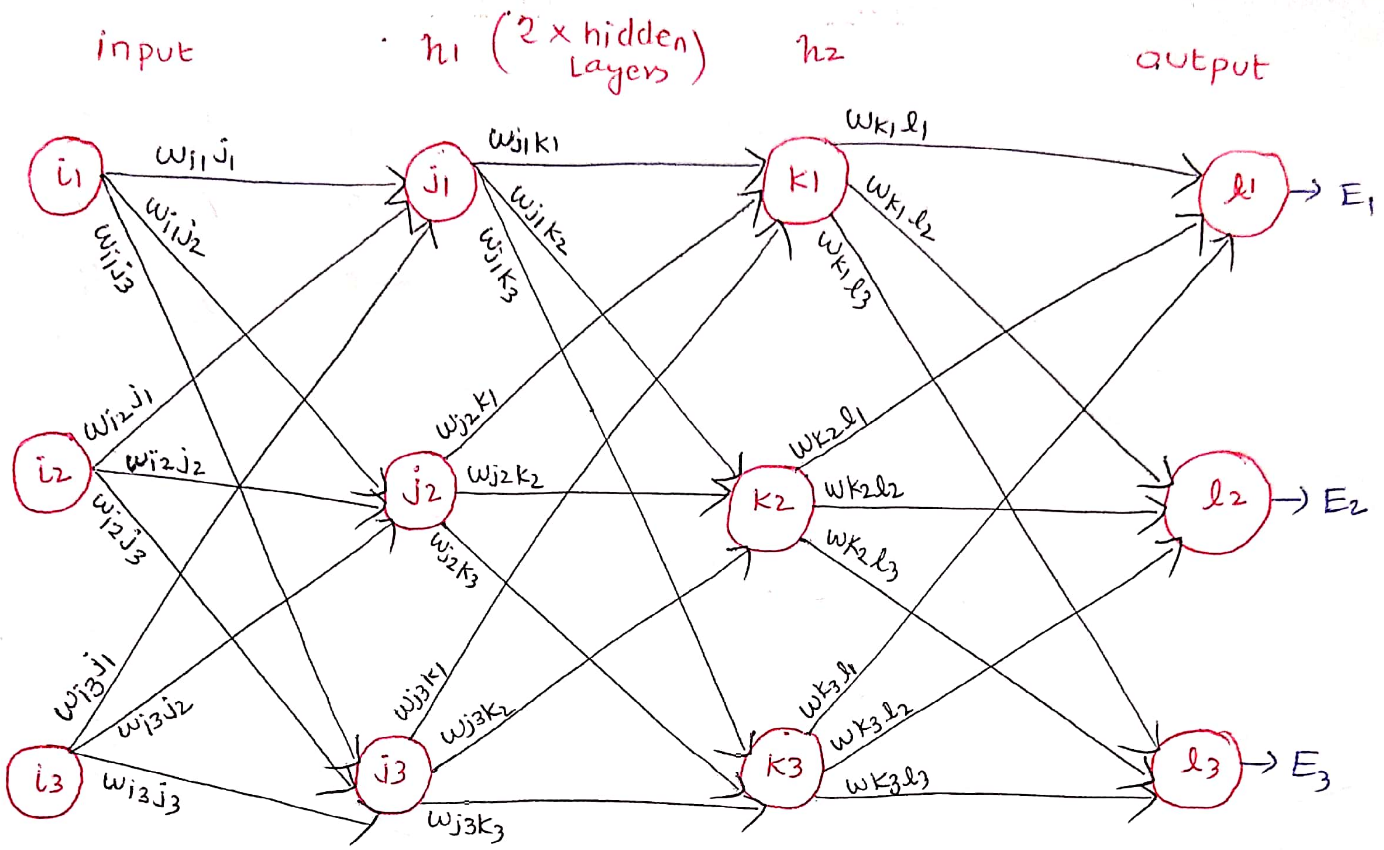
Backpropagation Algorithm



Architecture : Forward neural network with 2 hidden Layers and all layers have 3 neurons.

→ 1st & 2nd Hidden layers are gated by Relu and sigmoid activation functions.

→ output layer have softmax as activation function.

Network Initialization :

→ All the input values, output values and the weights considered aftermath are highly arbitrary in nature.

→ Error is Computed using Cross-Entropy

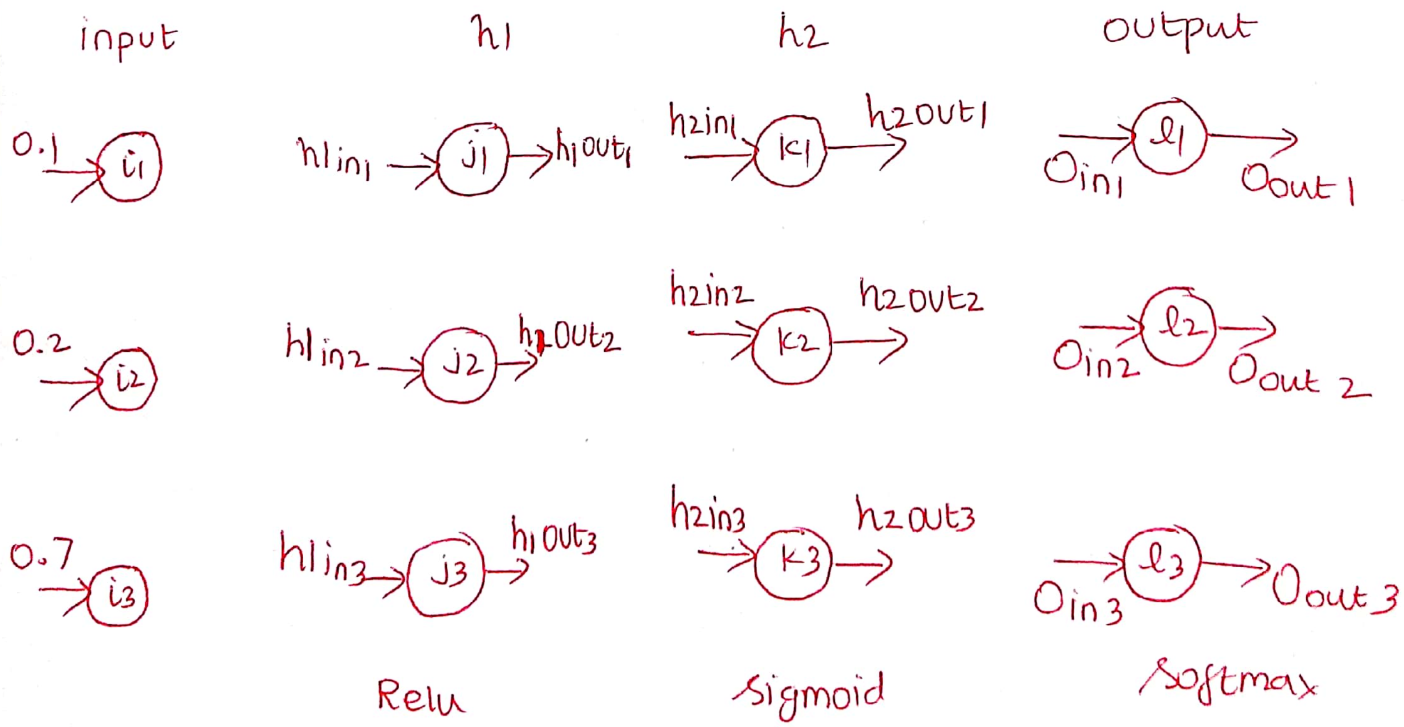
$$\text{inputs} = \begin{bmatrix} 0.1 & 0.2 & 0.7 \end{bmatrix}$$

$$W_{ij} = \begin{bmatrix} W_{i_1 j_1} & W_{i_1 j_2} & W_{i_1 j_3} \\ W_{i_2 j_1} & W_{i_2 j_2} & W_{i_2 j_3} \\ W_{i_3 j_1} & W_{i_3 j_2} & W_{i_3 j_3} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.3 & 0.2 & 0.7 \\ 0.4 & 0.2 & 0.9 \end{bmatrix}$$

$$W_{jk} = \begin{bmatrix} W_{j_1 k_1} & W_{j_1 k_2} & W_{j_1 k_3} \\ W_{j_2 k_1} & W_{j_2 k_2} & W_{j_2 k_3} \\ W_{j_3 k_1} & W_{j_3 k_2} & W_{j_3 k_3} \end{bmatrix} = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.3 & 0.5 & 0.7 \\ 0.6 & 0.4 & 0.8 \end{bmatrix}$$

$$W_{kl} = \begin{bmatrix} W_{k_1 l_1} & W_{k_1 l_2} & W_{k_1 l_3} \\ W_{k_2 l_1} & W_{k_2 l_2} & W_{k_2 l_3} \\ W_{k_3 l_1} & W_{k_3 l_2} & W_{k_3 l_3} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.4 & 0.8 \\ 0.3 & 0.7 & 0.2 \\ 0.5 & 0.2 & 0.9 \end{bmatrix}$$

$$\text{Output} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \end{bmatrix}$$



Matrix operations : Layer - 1

Bias terms

$$[i_1 \quad i_2 \quad i_3]\begin{bmatrix} W_{i_1 j_1} & W_{i_1 j_2} & W_{i_1 j_3} \\ W_{i_2 j_1} & W_{i_2 j_2} & W_{i_2 j_3} \\ W_{i_3 j_3} & W_{i_2 j_2} & W_{i_3 j_3} \end{bmatrix} + [b_{j_1} \; b_{j_2} \; b_{j_3}] = [h1_{in_1} \; h1_{in_2} \; h1_{in_3}]$$

Relu operation :

$$relu = \max(0, x)$$

$$[h1out_1 \quad h1out_2 \quad h1out_3] = [\max(0, h1_{in_1}) \quad \max(0, h1_{in_2}) \quad \max(0, h1_{in_3})]$$

Ex: (Kindly change the same in prev upper right corner)

$$[0.1 \quad 0.2 \quad 0.7] \times \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.3 & 0.2 & 0.7 \\ 0.4 & 0.3 & 0.9 \end{bmatrix} + [1 \quad 1 \quad 1] = [1.35 \quad 1.27 \quad 1.8]$$

$$[h1out_1 \quad h1out_2 \quad h1out_3] = [1.35 \quad 1.27 \quad 1.8]$$

— end of Layer 1 —

Matrix operations : Layer - 2

$$\begin{bmatrix} h1out_1 & h1out_2 & h1out_3 \end{bmatrix} \begin{bmatrix} W_{j1k1} & W_{j1k2} & W_{j1k3} \\ W_{j2k1} & W_{j2k2} & W_{j2k3} \\ W_{j3k1} & W_{j3k2} & W_{j3k3} \end{bmatrix} + \begin{bmatrix} b_{k1} & b_{k2} & b_{k3} \end{bmatrix} = \begin{bmatrix} h2in_1 & h2in_2 & h2i \end{bmatrix}$$

Sigmoid operation :

$$\text{Sigmoid} = \frac{1}{(1+e^{-x})}$$

$$\begin{bmatrix} h2out_1 & h2out_2 & h2out_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{(1+e^{-h2in_1})} & \frac{1}{(1+e^{-h2in_2})} & \frac{1}{(1+e^{-h2in_3})} \end{bmatrix}$$

$$\begin{bmatrix} 1.35 & 1.27 & 1.8 \end{bmatrix} \times \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.3 & 0.5 & 0.7 \\ 0.6 & 0.4 & 0.8 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \begin{array}{ccc} h2in_1 & h2in_2 & h2in_3 \\ \end{array} \begin{bmatrix} 2.73 & 2.76 & 4.001 \end{bmatrix}$$

$$\begin{bmatrix} h2out_1 & h2out_2 & h2out_3 \end{bmatrix} = \begin{bmatrix} 0.938 & 0.94 & 0.98 \end{bmatrix}$$

— end of layer 2 —

Example Calculations :

$$\frac{1}{1+e^{-h1in_1}} = \frac{1}{1+e^{-2.73}} = \frac{1}{1.06521} = 0.938\ldots$$

Matrix operations : Layer 3 (B3)

$$\begin{bmatrix} h_{2out1} & h_{2out2} & h_{2out3} \end{bmatrix} \times \begin{bmatrix} W_{k1l1} & W_{k1l2} & W_{k1l3} \\ W_{k2l1} & W_{k2l2} & W_{k2l3} \\ W_{k3l1} & W_{k3l3} & W_{k3l3} \end{bmatrix} + \begin{bmatrix} b_{l1} & b_{l2} & b_{l3} \end{bmatrix} = \begin{bmatrix} O_{in1} & O_{in2} & O_{in3} \end{bmatrix}$$

Softmax operation

$$\text{Softmax} = \frac{e^{O_{ina}}}{\sum_{a=1}^{3} e^{O_{ina}}}$$

$$\begin{bmatrix} O_{out1} & O_{out2} & O_{out3} \end{bmatrix} = \begin{bmatrix} \dfrac{e^{O_{in1}}}{\sum_{a=1}^{3} e^{O_{ina}}} & \dfrac{e^{O_{in2}}}{\sum_{a=1}^{3} e^{O_{ina}}} & \dfrac{e^{O_{in3}}}{\sum_{a=1}^{3} e^{O_{ina}}} \end{bmatrix}$$

$$\begin{bmatrix} 0.938 & 0.94 & 0.98 \end{bmatrix} \times \begin{bmatrix} 0.1 & 0.4 & 0.8 \\ 0.3 & 0.7 & 0.2 \\ 0.5 & 0.2 & 0.9 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} = \begin{array}{ccc} O_{in1} & O_{in2} & O_{in3} \\ \end{array} \begin{bmatrix} 1.865 & 2.229 & 2.820 \end{bmatrix}$$

$$\begin{bmatrix} O_{out1} & O_{out2} & O_{out3} \end{bmatrix} = \begin{bmatrix} 0.1985 & 0.2855 & 0.5158 \end{bmatrix}$$

—— end of Output Layer ——

Example Calculation

$$O_{out1} = \frac{e^{O_{in1}}}{e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}}} = \frac{e^{1.865}}{e^{1.865} + e^{2.229} + e^{2.820}} = \frac{6.4559}{(6.4559 + 9.290 + 16.77)}$$

$$e^{ab} \equiv e^{a} + e^{b}$$

$$= \frac{6.4559}{32.5227} = 0.1985$$

The Actual outputs should be $\begin{array}{ccc} y_1 & y_2 & y_3 \\ \left[1 \right. & 0 & \left. 0 \right] \end{array}$

But we've got outputs as $\begin{array}{ccc} \left[0.1985 \right. & 0.2855 & 0.5158 \\ O_{out1} & O_{out2} & O_{out3} \end{array}$ -

$\therefore$ Error :

$\sum P(x) \log(q(x))$

Cross Entropy formula $= \left(-\frac{1}{n}\right)\left[\sum_{i=1}^{3} \left(y_i \times \log(O_{outi})\right) + \left((1-y_i) \times \log((1-O_{outi}))\right)\right]$

(n = no. of examples in a batch)

Example :

$$\text{CE Error} = (-1)\Big[\left[(y_1)\log(0.1985) + (1-y_1)\log(1-0.1985)\right] + \left[(y_2)\log(0.2855) + (1-y_2)\log(1-0.2855)\right] + \left[(y_3)\log(0.5158) + (1-y_3)\log(1-0.5158)\right]\Big]$$

$$= (-1)\Big[\left[1.\log(0.1985) + \cancel{(1-1)\log}(1-0.1985)\right] + \left[\cancel{(0)\log(0.2855)} + (1-0)\log(1-0.2855)\right] + \left[\cancel{(0)\log(0.5158)} + (1-0)\log(1-0.5158)\right]\Big]$$

$$= (-1)\left[\log(0.1985) + \log(0.7145) + \log(0.4842)\right]$$

$$= (-1)\left[-0.70223 - 0.1459 - 0.3149\right]$$

$$= (-1)(-1.16321)$$

$$\text{CE Error} = 1.16321$$

— END OF FORWARD PASS —

Handy Derivatives :

$$\text{Sigmoid} = \frac{1}{(1+e^{-x})} \qquad \left(\frac{u}{v}\right) = \frac{vu' - uv'}{v^2}$$

$$\frac{d(\;)}{dx} = \frac{(1+e^{-x})(0) - (1)(1+e^{-x})(-1)}{(1+e^{-x})^2}$$

$$= \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1}{(1+e^{-x})} \cdot \frac{e^{-x}}{(1+e^{-x})}$$

$$= \frac{1}{(1+e^{-x})} \left(\frac{e^{-x}+1-1}{(1+e^{-x})} \right) = \frac{1}{(1+e^{-x})} \cdot \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}} \right)$$

$$= \frac{1}{(1+e^{-x})} \cdot \left(1 - \frac{1}{1+e^{-x}} \right)$$

$$= \text{sigmoid}\,(1 - \text{sigmoid})$$

$$\therefore \frac{d(\text{sigmoid})}{dx} = \text{sigmoid}\,(1 - \text{sigmoid})$$

Relu :

$$relu = \max(0, x)$$

$$\left.\begin{aligned} &\text{if } x > 0, \quad \frac{\partial(relu)}{\partial x} = 1 \\ &\text{if } x < 0, \quad \frac{\partial(relu)}{\partial x} = 0 \end{aligned}\right\}$$

Softmax :

$$\text{Softmax} = \frac{e^{x_a}}{\sum_{a=1}^{n} e^{x_a}} = \frac{e^{x_1}}{e^{x_1} + e^{x_2} + e^{x_3}} \quad \left(\frac{u}{v}\right)$$

$$\frac{\partial(\text{Softmax})}{\partial x_1} = \frac{vu' - uv'}{v^2} = \frac{(e^{x_1} + e^{x_2} + e^{x_3})(e^{x_1})(1) - (e^{x_1})(e^{x_1})(1+0+0)}{(e^{x_1} + e^{x_2} + e^{x_3})^2}$$

$$= \frac{(\cancel{e^{x_1} e^{x_1}} + e^{x_2} e^{x_1} + e^{x_3} e^{x_1}) - (\cancel{e^{x_1} e^{x_1}})}{(e^{x_1} + e^{x_2} + e^{x_3})^2}$$

$$= \frac{e^{x_1} e^{x_2} + e^{x_1} e^{x_3}}{(e^{x_1} + e^{x_2} + e^{x_3})^2}$$

$$\frac{\partial(\text{Softmax})}{\partial x_1} = \frac{e^{x_1}(e^{x_2} + e^{x_3})}{(e^{x_1} + e^{x_2} + e^{x_3})^2}$$

→ Let's Backpropagate, starting from the last layer upto the first layer.

$$\frac{\partial E_1}{\partial out_1} = \underbrace{\left(-\frac{1}{1}\right)}_{n=1} \overbrace{\frac{\partial\left[\left(y_1 * \log(O_{out1})\right) + (1-y_1) * \log(1-O_{out1})\right]}{\partial O_{out1}}}^{E1}$$

$$\frac{\partial E_1}{\partial out_1} = (-1)\left[(y_1) * \left(\frac{1}{O_{out1}}\right) + (1-y_1) * \left(\frac{1}{1-O_{out1}}\right)\right]$$

∴ By symmetry,

$$\begin{bmatrix} \frac{\partial E_1}{\partial O_{out1}} \\ \frac{\partial E_2}{\partial O_{out2}} \\ \frac{\partial E_3}{\partial O_{out3}} \end{bmatrix} = \begin{bmatrix} (-1)\left[(y_1)\left(\frac{1}{O_{out1}}\right) + (1-y_1)\left(\frac{1}{1-O_{out1}}\right)\right] \\ (-1)\left[(y_2)\left(\frac{1}{O_{out2}}\right) + (1-y_2)\left(\frac{1}{1-O_{out2}}\right)\right] \\ (-1)\left[(y_3)\left(\frac{1}{O_{out3}}\right) + (1-y_3)\left(\frac{1}{1-O_{out3}}\right)\right] \end{bmatrix}$$

∴ For our calculation,

$$\begin{bmatrix} \frac{\partial E_1}{\partial O_{out1}} \\ \frac{\partial E_2}{\partial O_{out2}} \\ \frac{\partial E_3}{\partial O_{out3}} \end{bmatrix} = \begin{bmatrix} (-1)\left[(1)\left(\frac{1}{0.1985}\right) + (0)(*)\right] \\ (-1)\left[(0)(*) + (1-0)\left(\frac{1}{1-0.2855}\right)\right] \\ (-1)\left[0(*) + (1-0)\left(\frac{1}{1-0.5158}\right)\right] \end{bmatrix}$$

$$\underbrace{\begin{matrix} y_1 = 1 \\ y_2 = 0 \\ y_3 = 0 \end{matrix}}_{\text{expected outputs}}$$

$O_{out1} = 0.1985$

$O_{out2} = 0.2855$

$O_{out3} = 0.5158$

Finally,

$$\begin{bmatrix} \frac{\partial E_1}{\partial O_{out1}} \\ \frac{\partial E_2}{\partial O_{out2}} \\ \frac{\partial E_3}{\partial O_{out3}} \end{bmatrix} = \begin{bmatrix} -5.0377 \\ -1.3995 \\ -2.0652 \end{bmatrix}$$

Next let's calculate the derivative of each output with respect to their inputs.

$$\frac{\partial O_{out_1}}{\partial O_{in_1}} = \frac{\partial\left(e^{O_{in1}} / (e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}})\right)}{\partial O_{in_1}}$$

(refer previous derivative

$$\frac{\partial O_{out1}}{\partial O_{in1}} = \frac{e^{O_{in1}} \times (e^{O_{in2}} + e^{O_{in3}})}{(e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}})^2}$$

(Derivative of softmax wrt output layer input

By symmetry we can calculate other derivatives also,

$$\begin{bmatrix} \frac{\partial O_{out1}}{\partial O_{in1}} \\ \frac{\partial O_{out2}}{\partial O_{in2}} \\ \frac{\partial O_{out3}}{\partial O_{in3}} \end{bmatrix} = \begin{bmatrix} \frac{e^{O_{in1}} \times (e^{O_{in2}} + e^{O_{in3}})}{(e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}})^2} \\ \frac{e^{O_{in2}} \times (e^{O_{in1}} + e^{O_{in3}})}{(e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}})^2} \\ \frac{e^{O_{in3}} \times (e^{O_{in1}} + e^{O_{in2}})}{(e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}})^2} \end{bmatrix}$$

(matrix of derivative of softmax wrt output layer input.

$$\begin{bmatrix} \frac{\partial O_{out1}}{\partial O_{in1}} \\ \frac{\partial O_{out2}}{\partial O_{in2}} \\ \frac{\partial O_{out3}}{\partial O_{in3}} \end{bmatrix} = \begin{bmatrix} e^{1.865} \times (e^{2.229} + e^{2.820}) / (e^{1.865} + e^{2.229} + e^{2.820})^2 \\ e^{2.229} \times (e^{1.865} + e^{2.820}) / (e^{1.865} + e^{2.229} + e^{2.820})^2 \\ e^{2.820} \times (e^{1.865} + e^{2.229}) / (e^{1.865} + e^{2.229} + e^{2.820})^2 \end{bmatrix}$$

$O_{in_1} = 1.865$

$O_{in2} = 2.229$

$O_{in3} = 2.820$

$$\begin{bmatrix} \partial O_{out1} / \partial O_{in1} \\ \partial O_{out2} / \partial O_{in2} \\ \partial O_{out3} / \partial O_{in3} \end{bmatrix} = \begin{bmatrix} 0.1591 \\ 0.2040 \\ 0.3685 \end{bmatrix}$$

Values of derivatives of softmax wrt output layer input.

For each input to neuron let us Calculate the derivative with respect to each weight. Now let us look at the final derivative.

$$\frac{\partial O_{in1}}{\partial W_{k1l1}} = \frac{\partial\left[(h2out_1 * W_{j1k1}) + (h2out_2 * W_{j2k1}) + (h2out_3 * W_{j3k1}) + b_{l1}\right]}{\partial W_{k1l1}}$$

$$\frac{\partial O_{in1}}{\partial W_{k1l1}} = h2out1$$

(Derivative of input to output layer wrt weight)

$\therefore$ By symmetry, the other derivatives are,

$$\begin{bmatrix} \frac{\partial O_{in1}}{\partial W_{k1l1}} \\ \frac{\partial O_{in1}}{\partial W_{k2l1}} \\ \frac{\partial O_{in1}}{\partial W_{k3l1}} \end{bmatrix} = \begin{bmatrix} h2out_1 \\ h2out_2 \\ h2out_3 \end{bmatrix} = \begin{bmatrix} 0.938 \\ 0.94 \\ 0.98 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial O_{in2}}{\partial W_{k1l2}} \\ \frac{\partial O_{in2}}{\partial W_{k2l2}} \\ \frac{\partial O_{in3}}{\partial W_{k3l2}} \end{bmatrix} = \begin{bmatrix} h2out_1 \\ h2out_2 \\ h2out_3 \end{bmatrix} = \begin{bmatrix} 0.938 \\ 0.94 \\ 0.98 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial O_{in3}}{\partial W_{k1l3}} \\ \frac{\partial O_{in3}}{\partial W_{k2l3}} \\ \frac{\partial O_{in3}}{\partial W_{k3l3}} \end{bmatrix} = \begin{bmatrix} h2out_1 \\ h2out_2 \\ h2out_3 \end{bmatrix} = \begin{bmatrix} 0.938 \\ 0.94 \\ 0.98 \end{bmatrix}$$

(Values of derivative of i/p to O/p layer wrt weights)

Finally let us calculate the change in

$W_{k1 l1}$ (weight from k1 to l1 neuron)

which is simply, $\frac{\partial E_1}{\partial W_{k1 l1}}$ (Derivative of error wrt weight)

Using chain rule : $\frac{\partial E_1}{\partial W_{k1 l1}} = \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial W_{k1 l1}}$

(chain rule breakdown of error derivative)

$$\therefore \delta W_{kl} = \begin{bmatrix} \frac{\partial E_1}{\partial W_{k1 l1}} & \frac{\partial E_2}{\partial W_{k1 l2}} & \frac{\partial E_3}{\partial W_{k1 l3}} \\ \frac{\partial E_1}{\partial W_{k2 l1}} & \frac{\partial E_2}{\partial W_{k2 l2}} & \frac{\partial E_3}{\partial W_{k2 l3}} \\ \frac{\partial E_1}{\partial W_{k3 l1}} & \frac{\partial E_2}{\partial W_{k3 l2}} & \frac{\partial E_3}{\partial W_{k3 l3}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial W_{k1 l1}} & \frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial W_{k1 l2}} & \frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial W_{k1 l3}} \\ \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial W_{k2 l1}} & \frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial W_{k2 l2}} & \frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial W_{k2 l3}} \\ \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial W_{k3 l1}} & \frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial W_{k3 l2}} & \frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial W_{k3 l3}} \end{bmatrix}$$

(matrix form of all derivatives only in layer 3)

B7

$$\delta W_{kl} = \begin{bmatrix} \delta W_{k1l1} & \delta W_{k1l2} & \delta W_{k1l3} \\ \delta W_{k2l1} & \delta W_{k2l2} & \delta W_{k2l3} \\ \delta W_{k3l1} & \delta W_{k3l2} & \delta W_{k3l3} \end{bmatrix}$$

$$\delta W_{kl} = \begin{bmatrix} -5.0377 * 0.1591 * 0.938 & -1.3995 * 0.204 * 0.938 & -2.0652 * 0.3685 * 0.938 \\ -5.0377 * 0.1591 * 0.94 & -1.3995 * 0.204 * 0.94 & -2.0652 * 0.3685 * 0.94 \\ -5.0377 * 0.1591 * 0.98 & -1.3995 * 0.204 * 0.98 & -2.0652 * 0.3635 * 0.98 \end{bmatrix}$$

$$\delta W_{kl} = \begin{bmatrix} -0.7518 & -0.2677 & -0.7138 \\ -0.7534 & -0.2683 & -0.7153 \\ -0.7854 & -0.2797 & -0.7458 \end{bmatrix}$$

$\therefore$ Considering an learning rate of 0.01, we get our final weight matrix as,

$$W_{kl} = \begin{bmatrix} W_{k1l1} - (lr * \delta W_{k1l1}) & W_{k1l2} - (lr * \delta W_{k1l2}) & W_{k1l3} - (lr * \delta W_{k1l3}) \\ W_{k2l1} - (lr * \delta W_{k2l1}) & W_{k2l2} - (lr * \delta W_{k2l2}) & W_{k2l3} - (lr * \delta W_{k2l3}) \\ W_{k3l1} - (lr * \delta W_{k3l1}) & W_{k3l2} - (lr * \delta W_{k3l2}) & W_{k3l3} - (lr * \delta W_{k3l3}) \end{bmatrix}$$

$$W_{kl} = \begin{bmatrix} 0.1 - (0.01 * -0.7518) & 0.4 - (0.01 * -0.2677) & 0.8 - (0.01 * -0.7138) \\ 0.3 - (0.01 * -0.7534) & 0.7 - (0.01 * -0.2683) & 0.2 - (0.01 * -0.7153) \\ 0.5 - (0.01 * -0.7854) & 0.2 - (0.01 * -0.2797) & 0.9 - (0.01 * -0.7458) \end{bmatrix}$$

$$W_{kl} = \begin{bmatrix} 0.107518 & 0.402677 & 0.807138 \\ 0.307534 & 0.702683 & 0.207153 \\ 0.507854 & 0.202797 & 0.907458 \end{bmatrix}$$

modified weights of Kl neurons after backprop

Now let's move to the next Layer in Backprop,

Backpropagating the error, (HL1 - HL2) weights

$$\frac{\partial h2out1}{\partial h2in1} = \frac{\partial\, \text{sigmoid}(h2in1)}{\partial h2in1}$$

$$\frac{\partial h2out1}{\partial h2in1} = \text{sigmoid}(h2in1) * (1 - \text{sigmoid}(h2in1))$$

$$\therefore \begin{bmatrix} \frac{\partial h2out1}{\partial h2in1} \\ \frac{\partial h2out2}{\partial h_2in2} \\ \frac{\partial h2out3}{\partial h2in3} \end{bmatrix} = \begin{bmatrix} \text{Sigmoid}(h2in1) * (1 - \text{sigmoid}(h2in1)) \\ \text{Sigmoid}(h2in2) * (1 - \text{sigmoid}(h2in2)) \\ \text{sigmoid}(h2in3) * (1 - \text{sigmoid}(h2in3)) \end{bmatrix}$$

(Derivative of Sigmoid o/p wrt layer 2 input

$$\begin{bmatrix} \frac{\partial h2out1}{\partial h2in1} \\ \frac{\partial h2out2}{\partial h2in2} \\ \frac{\partial h2out3}{\partial h2in3} \end{bmatrix} = \begin{bmatrix} \text{Sig}(2.73) * (1 - \text{sig}(2.73)) \\ \text{Sig}(2.76) * (1 - \text{sig}(2.76)) \\ \text{sig}(4.001) * (1 - \text{sig}(4.001)) \end{bmatrix} \quad \left(\text{Sig} = \frac{1}{1+e^{-x}}\right)$$

$$\begin{bmatrix} \frac{\partial h2out1}{\partial h2in1} \\ \frac{\partial h2out2}{\partial h2in2} \\ \frac{\partial h2out3}{\partial h2in3} \end{bmatrix} = \begin{bmatrix} 0.05747 \\ 0.05598 \\ 0.01764 \end{bmatrix}$$

$$\left(\frac{1}{1+e^{-2.73}}\right) * \left(1 - \left(\frac{1}{1+e^{-2.73}}\right)\right) = 0.05747$$

(values of derivative of O/p of layer-2 wrt Layer-1)

For each input to neuron let us Calculate the derivative wrt each weight.

$$\frac{\partial h_{2in1}}{\partial W_{j1k1}} = \frac{\partial\left[(h_{1out1} * W_{j1k1}) + (h_{1out2} * W_{j2k1}) + (h_{1out3} * W_{j3k1}) + b_{k1}\right]}{\partial W_{j1k1}}$$

$$\frac{\partial h_{2in1}}{\partial W_{j1k1}} = h_{1out1}$$ (Derivative of Layer 2 i/p wrt weight)

∴ By symmetry,

$$\begin{bmatrix} \frac{\partial h_{2in1}}{\partial W_{j1k1}} \\ \frac{\partial h_{2in1}}{\partial W_{j2k1}} \\ \frac{\partial h_{2in1}}{\partial W_{j3k1}} \end{bmatrix} = \begin{bmatrix} h_{1out1} \\ h_{1out2} \\ h_{1out3} \end{bmatrix} = \begin{bmatrix} 1.35 \\ 1.27 \\ 1.8 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial h_{2in2}}{\partial W_{j1k2}} \\ \frac{\partial h_{2in2}}{\partial W_{j2k2}} \\ \frac{\partial h_{2in2}}{\partial W_{j3k2}} \end{bmatrix} = \begin{bmatrix} h_{1out1} \\ h_{1out2} \\ h_{1out3} \end{bmatrix} = \begin{bmatrix} 1.35 \\ 1.27 \\ 1.8 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial h_{2in3}}{\partial W_{j1k3}} \\ \frac{\partial h_{2in3}}{\partial W_{j2k3}} \\ \frac{\partial h_{2in3}}{\partial W_{j3k3}} \end{bmatrix} = \begin{bmatrix} h_{1out1} \\ h_{1out2} \\ h_{1out3} \end{bmatrix} = \begin{bmatrix} 1.35 \\ 1.27 \\ 1.8 \end{bmatrix}$$ (Values of derivative layer-2 i/p wrt to weight)

Now let's Calculate derivative of W_{j3k1}

(wt from j3 to k1)

i.e $\frac{\partial E_{total}}{\partial W_{j3k1}}$ (Derivative of error wrt weight j3-k1) (last node in 2nd HL)

Using chain rule,

$$\frac{\partial E_{total}}{\partial W_{j3k1}} = \frac{\partial E_{total}}{\partial h2out_1} * \frac{\partial h2out_1}{\partial h2in_1} * \frac{\partial h2in_1}{\partial W_{j3k1}}$$ (chain rule of derivative of error wrt wt)

∴ By symmetry, the final matrix is,

$$\delta_{W_{jk}} = \begin{bmatrix} \frac{\partial E_{total}}{\partial W_{j1k1}} & \frac{\partial E_{total}}{\partial W_{j1k2}} & \frac{\partial E_{total}}{\partial W_{j1k3}} \\ \frac{\partial E_{total}}{\partial W_{j2k1}} & \frac{\partial E_{total}}{\partial W_{j2k2}} & \frac{\partial E_{total}}{\partial W_{j2k3}} \\ \frac{\partial E_{total}}{\partial W_{j3k1}} & \frac{\partial E_{total}}{\partial W_{j3k2}} & \frac{\partial E_{total}}{\partial W_{j3k3}} \end{bmatrix}$$

fill their derivatives

$$\delta_{W_{jk}} = \begin{bmatrix} \frac{\partial E_{total}}{\partial h2out_1} * \frac{\partial h2out_1}{\partial h2in_1} * \frac{\partial h2in_1}{\partial W_{j1k1}} & \frac{\partial E_{total}}{\partial h2out_2} * \frac{\partial h2out_2}{\partial h2in_2} * \frac{\partial h2in_2}{\partial W_{j1k2}} & \frac{\partial E_{total}}{\partial h2out_3} * \frac{\partial h2out_3}{\partial h2in_3} * \frac{\partial h2in_3}{\partial W_{j1k3}} \\ \frac{\partial E_{total}}{\partial h2out_1} * \frac{\partial h2out_1}{\partial h2in_1} * \frac{\partial h2in_1}{\partial W_{j2k1}} & \frac{\partial E_{total}}{\partial h2out_2} * \frac{\partial h2out_2}{\partial h2in_2} * \frac{\partial h2in_2}{\partial W_{j2k2}} & \frac{\partial E_{total}}{\partial h2out_3} * \frac{\partial h2out_3}{\partial h2in_3} * \frac{\partial h2in_3}{\partial W_{j2k3}} \\ \frac{\partial E_{total}}{\partial h2out_1} * \frac{\partial h2out_1}{\partial h2in_1} * \frac{\partial h2in_1}{\partial W_{j3k1}} & \frac{\partial E_{total}}{\partial h2out_2} * \frac{\partial h2out_2}{\partial h2in_2} * \frac{\partial h2in_2}{\partial W_{j3k2}} & \frac{\partial E_{tot}}{\partial h2out_3} * \frac{\partial h2out_3}{\partial h2in_3} * \frac{\partial h2in_3}{\partial W_{j3k3}} \end{bmatrix}$$

(Final matrix of derivatives of weights $\{jk\}$)

But we have already Calculated 2nd and 3rd term for all terms in each matrix, See before (0, -1) pages and validate. ∴ only the first term needs to be Computed and they all are same in all columns,

$\frac{\partial E_{total}}{\partial h2out_{1,2,3}} \dots$

Error Breakdown ex:

$$\frac{\partial E_{total}}{\partial h2_{out1}} = \frac{\partial E_1}{\partial h2_{out1}} + \frac{\partial E_2}{\partial h2_{out1}} + \frac{\partial E_3}{\partial h2_{out1}}$$

$\therefore$ Each term individually corresponds to,

$$\frac{\partial E_1}{\partial h2_{out1}} = \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial h2_{out1}}$$

$$\frac{\partial E_2}{\partial h2_{out1}} = \frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial h2_{out1}}$$

$$\frac{\partial E_3}{\partial h2_{out1}} = \frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial h2_{out1}}$$

$\therefore$ By symmetry, we get the final matrix as,

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h2_{out1}} \\ \frac{\partial E_{total}}{\partial h2_{out2}} \\ \frac{\partial E_{total}}{\partial h2_{out3}} \end{bmatrix} = \begin{bmatrix} \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial h2_{out1}} & \frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial h2_{out1}} & \frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial h2_{out1}} \\ \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial h2_{out2}} & \frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial h2_{out2}} & \frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial h2_{out2}} \\ \frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * \frac{\partial O_{in1}}{\partial h2_{out3}} & \frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * \frac{\partial O_{in2}}{\partial h2_{out3}} & \frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * \frac{\partial O_{in3}}{\partial h2_{out3}} \end{bmatrix}$$

derivative of i/p to each o/p Layer wrt o/p of HL(2), i.e corresponding weight which connects both the Layers.

These 1st & 2nd terms are already computed, we need to find only the 3rd one in each matrix term $\frac{\partial O_{in1}}{\partial h2_{out1}}$..

But
$$\begin{bmatrix} \frac{\partial O_{in1}}{\partial h2_{out1}} & \frac{\partial O_{in2}}{\partial h2_{out1}} & \frac{\partial O_{in3}}{\partial h2_{out1}} \\ \frac{\partial O_{in1}}{\partial h2_{out2}} & \frac{\partial O_{in2}}{\partial h2_{out2}} & \frac{\partial O_{in3}}{\partial h2_{out2}} \\ \frac{\partial O_{in1}}{\partial h2_{out3}} & \frac{\partial O_{in2}}{\partial h2_{out3}} & \frac{\partial O_{in3}}{\partial h2_{out3}} \end{bmatrix} \equiv \begin{bmatrix} W_{k1l1} & W_{k1l2} & W_{k1l3} \\ W_{k2l1} & W_{k2l2} & W_{k2l3} \\ W_{k3l1} & W_{k3l2} & W_{k3l3} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h2_{out1}} \\ \frac{\partial E_{total}}{\partial h2_{out2}} \\ \frac{\partial E_{total}}{\partial h2_{out3}} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * W_{k1l1}\right) + \left(\frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * W_{k1l2}\right) + \left(\frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * W_{k1l3}\right) \\ \left(\frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * W_{k2l1}\right) + \left(\frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * W_{k2l2}\right) + \left(\frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * W_{k2l3}\right) \\ \left(\frac{\partial E_1}{\partial O_{out1}} * \frac{\partial O_{out1}}{\partial O_{in1}} * W_{k3l1}\right) + \left(\frac{\partial E_2}{\partial O_{out2}} * \frac{\partial O_{out2}}{\partial O_{in2}} * W_{k3l2}\right) + \left(\frac{\partial E_3}{\partial O_{out3}} * \frac{\partial O_{out3}}{\partial O_{in3}} * W_{k3l3}\right) \end{bmatrix}$$

[final matrix of derivative of total error wrt o/p of HL-2]

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h2_{out1}} \\ \frac{\partial E_{total}}{\partial h2_{out2}} \\ \frac{\partial E_{total}}{\partial h2_{out3}} \end{bmatrix} = \begin{bmatrix} (-5.0377 * 0.1591 * 0.1) + (-1.3995 * 0.204 * 0.4) + (-2.0652 * 0.3685 * 0.8) \\ (-5.0377 * 0.1591 * 0.3) + (-1.3995 * 0.204 * 0.7) + (-2.0652 * 0.3685 * 0.2) \\ (-5.0377 * 0.1591 * 0.5) + (-1.3995 * 0.204 * 0.2) + (-2.0652 * 0.3685 * 0.9) \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h2_{out1}} \\ \frac{\partial E_{total}}{\partial h2_{out2}} \\ \frac{\partial E_{total}}{\partial h2_{out3}} \end{bmatrix} = \begin{bmatrix} (-0.0801) + (-0.1141) + (-0.6088) \\ (-0.2404) + (-0.1998) + (-0.1522) \\ (-0.4007) + (-0.0570) + (-0.6849) \end{bmatrix} = \begin{bmatrix} -0.8030 \\ -0.5954 \\ -1.1426 \end{bmatrix}$$

∴ our final matrix

$$\delta W_{jk} = \begin{bmatrix} \frac{\partial E_{total}}{\partial W_{j1k1}} & \frac{\partial E_{total}}{\partial W_{j1k2}} & \frac{\partial E_{total}}{\partial W_{j1k3}} \\ \frac{\partial E_{total}}{\partial W_{j2k1}} & \frac{\partial E_{total}}{\partial W_{j2k2}} & \frac{\partial E_{total}}{\partial W_{j2k3}} \\ \frac{\partial E_{total}}{\partial W_{j3k1}} & \frac{\partial E_{total}}{\partial W_{j3k2}} & \frac{\partial E_{total}}{\partial W_{j3k3}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial E_{total}}{\partial h2_{out1}} * \frac{\partial h2_{out1}}{\partial h2_{in1}} * \frac{\partial h2_{in1}}{\partial W_{j1k1}} & \frac{\partial E_{total}}{\partial h2_{out2}} * \frac{\partial h2_{out2}}{\partial h2_{in2}} * \frac{\partial h2_{in2}}{\partial W_{j1k2}} & \frac{\partial E_{tot}}{\partial h2_{out3}} * \frac{\partial h2_{out3}}{\partial h2_{in3}} * \frac{\partial h2_{in3}}{\partial W_{j1k3}} \\ \frac{\partial E_{total}}{\partial h2_{out1}} * \frac{\partial h2_{out1}}{\partial h2_{in1}} * \frac{\partial h2_{in1}}{\partial W_{j2k1}} & \frac{\partial E_{total}}{\partial h2_{out2}} * \frac{\partial h2_{out2}}{\partial h2_{in2}} * \frac{\partial h2_{in2}}{\partial W_{j2k2}} & \frac{\partial E_{tot}}{\partial h2_{out3}} * \frac{\partial h2_{out3}}{\partial h2_{in3}} * \frac{\partial h2_{in3}}{\partial W_{j2k3}} \\ \frac{\partial E_{total}}{\partial h2_{out1}} * \frac{\partial h2_{out1}}{\partial h2_{in1}} * \frac{\partial h2_{in1}}{\partial W_{j3k1}} & \frac{\partial E_{total}}{\partial h2_{out2}} * \frac{\partial h2_{out2}}{\partial h2_{in2}} * \frac{\partial h2_{in2}}{\partial W_{j3k2}} & \frac{\partial E_{tot}}{\partial h2_{out3}} * \frac{\partial h2_{out3}}{\partial h2_{in3}} * \frac{\partial h2_{in3}}{\partial W_{j3k3}} \end{bmatrix}$$

[final matrix of derivative of wt connecting HL - 1 & 2]

$$\delta W_{jk} = \begin{bmatrix} (-0.8030 * 0.05747 * 1.35) & (-0.5954 * 0.05598 * 1.35) & (-1.1426 * 0.01764 * 1.35) \\ (-0.8030 * 0.05747 * 1.27) & (-0.5954 * 0.05598 * 1.27) & (-1.1426 * 0.01764 * 1.27) \\ (-0.8030 * 0.05747 * 1.8) & (-0.5954 * 0.05598 * 1.8) & (-1.1426 * 0.01764 * 1.8) \end{bmatrix}$$

$$\delta W_{jk} = \begin{bmatrix} -0.06230 & -0.04499 & -0.02720 \\ -0.05860 & -0.04232 & -0.02559 \\ -0.08306 & -0.05999 & -0.03627 \end{bmatrix}$$

(Let say $Lr = 0.01$), ∴ final weight matrix is

$$W_{jk}^{(new)} = \begin{bmatrix} W_{j1k1} - (Lr * \delta W_{j1k1}) & W_{j1k2} - (lr * \delta W_{j1k2}) & W_{j1k3} - (lr * \delta W_{j1k3}) \\ W_{j2k1} - (lr * \delta W_{j2k1}) & W_{j2k2} - (lr * \delta W_{j2k2}) & W_{j2k3} - (lr * \delta W_{j2k3}) \\ W_{j3k3} - (lr * \delta W_{j3k1}) & W_{j3k2} - (lr * \delta W_{j3k2}) & W_{j3k3} - (lr * \delta W_{j3k3}) \end{bmatrix}$$

$$\bar{W}_{jk} = \begin{bmatrix} 0.2 - (0.01 * -0.06230) & 0.3 - (0.01 * -0.04499) & 0.5 - (0.01 * -0.0272) \\ 0.3 - (0.01 * -0.05860) & 0.5 - (0.01 * -0.04232) & 0.7 - (0.01 * -0.02559 \\ 0.6 - (0.01 * -0.08306) & 0.4 - (0.01 * -0.05999) & 0.8 - (0.01 * -0.03627 \end{bmatrix}$$

$$\bar{W}_{jk} = \begin{bmatrix} 0.200623 & 0.30044 & 0.50027 \\ 0.300586 & 0.500423 & 0.70025 \\ 0.600830 & 0.400599 & 0.800362 \end{bmatrix}$$

→ [final modified matrix of W_{jk}]

Backpropagating the error - (Input layer - HL 1) weights (B11)

$$\frac{\partial hl_{out1}}{\partial hl_{in1}} = \frac{\partial(Relu(hl_{in1}))}{\partial hl_{in1}}$$ (Derivative of HL 1 O/p wrt its input)

we know the derivative of relu, i.e since all i/p are positive, we get O/p as 1.

$$\begin{bmatrix} \frac{\partial hl_{out1}}{\partial hl_{in1}} \\ \frac{\partial hl_{out2}}{\partial hl_{in2}} \\ \frac{\partial hl_{out3}}{\partial hl_{in3}} \end{bmatrix} = \begin{bmatrix} 1.0 \\ 1.0 \\ 1.0 \end{bmatrix}$$

For each i/p to neuron, let's calculate derivative wrt each weight.

$$\frac{\partial hl_{in1}}{\partial W_{i1j1}} = \frac{\partial((I_{out1} * W_{j1k1}) + (I_{out2} * W_{i2j1}) + (I_{out3} * W_{j1i3}) + b_{j1})}{\partial W_{i1j1}}$$

$$\frac{\partial hl_{in1}}{\partial W_{i1j1}} = I_{out1}$$, ∴ By symmetry,

$$\begin{bmatrix} \frac{\partial hl_{in1}}{\partial W_{i1j1}} \\ \frac{\partial hl_{in1}}{\partial W_{i2j1}} \\ \frac{\partial hl_{in1}}{\partial W_{i3j1}} \end{bmatrix} = \begin{bmatrix} I_{1out1} \\ I_{out2} \\ I_{out3} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.7 \end{bmatrix} \quad \begin{bmatrix} \frac{\partial hl_{in2}}{\partial W_{i1j2}} \\ \frac{\partial hl_{in2}}{\partial W_{i2j2}} \\ \frac{\partial hl_{in2}}{\partial W_{j2i3}} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.7 \end{bmatrix} \quad \begin{bmatrix} \frac{\partial hl_{in3}}{\partial W_{i1j3}} \\ \frac{\partial hl_{in3}}{\partial W_{i2j3}} \\ \frac{\partial hl_{in3}}{\partial W_{i3j3}} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.7 \end{bmatrix}$$

Now let's calculate derivative of W_{i3j1} (wt from $i3 - j1$)

i.e

$$\frac{\partial E_{total}}{\partial W_{i2j1}} = \left(\begin{matrix}\text{Derivative of error} \\ \text{w.r.t wt } i2-j1\end{matrix}\right)\left(\begin{matrix}\text{Last node in} \\ \cancel{\text{1st}} \text{ Node} \\ \text{2nd}\end{matrix}\right)$$

using our chain rule,

$$\frac{\partial E_{total}}{\partial W_{i2j1}} = \frac{\partial E_{total}}{\partial h_1 out_1} * \frac{\partial hl\,out1}{\partial hl\,in_1} * \frac{\partial hl\,in1}{\partial W_{i2j1}}$$

Really Big term (So calculate separately & fill it back)

∴ By symmetry,

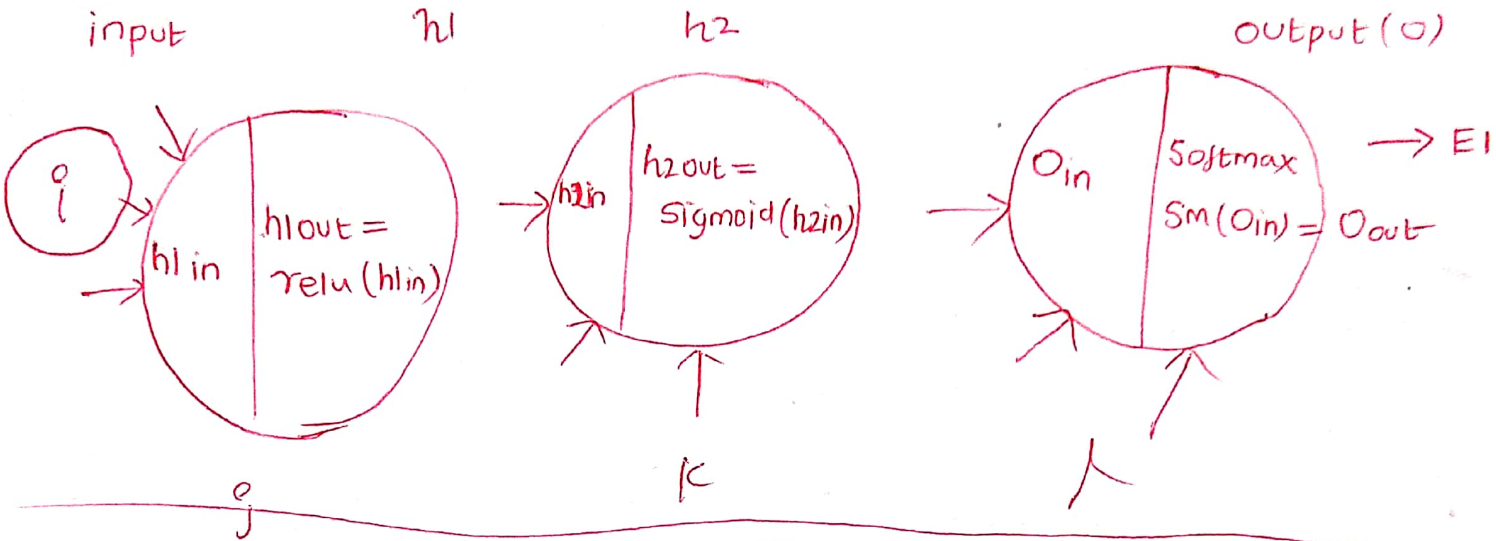
$$\delta W_{ij} = \begin{bmatrix} \frac{\partial E_{total}}{\partial W_{i1j1}} & \frac{\partial E_{total}}{\partial W_{i1j2}} & \frac{\partial E_{total}}{\partial W_{i1j3}} \\ \frac{\partial E_{total}}{\partial W_{i2j1}} & \frac{\partial E_{total}}{\partial W_{i2j2}} & \frac{\partial E_{total}}{\partial W_{i2j3}} \\ \frac{\partial E_{total}}{\partial W_{i3j1}} & \frac{\partial E_{total}}{\partial W_{i3j2}} & \frac{\partial E_{total}}{\partial W_{i3j3}} \end{bmatrix} \Rightarrow$$

$$= \begin{bmatrix} \frac{\partial E_{total}}{\partial hlout_1} * \frac{\partial hlout1}{\partial hl\,in_1} * \frac{\partial hl\,in1}{\partial W_{i1j1}} & \frac{\partial E_{total}}{\partial hlout_2} * \frac{\partial hlout_2}{\partial hlin_2} * \frac{\partial hlin_2}{\partial W_{i1j2}} & \frac{\partial E_{total}}{\partial hlout_3} * \frac{\partial hlout_3}{\partial hlin_3} * \frac{\partial hlin3}{\partial W_{i1j3}} \\ \frac{\partial E_{total}}{\partial hlout_1} * \frac{\partial hlout1}{\partial hlin_1} * \frac{\partial hlin1}{\partial W_{i2j1}} & \frac{\partial E_{total}}{\partial hlout_2} * \frac{\partial hlout_2}{\partial hlin_2} * \frac{\partial hlin_2}{\partial W_{i2j2}} & \frac{\partial E_{total}}{\partial hlout_3} * \frac{\partial hlout3}{\partial hlin_3} * \frac{\partial hlin3}{\partial W_{i2j3}} \\ \frac{\partial E_{total}}{\partial hlout_1} * \frac{\partial hlout1}{\partial hl\,in_1} * \frac{\partial hlin1}{\partial W_{i3j1}} & \frac{\partial E_{total}}{\partial hlout_2} * \frac{\partial hlout_2}{\partial hlin_2} * \frac{\partial hlin_2}{\partial W_{i3j2}} & \frac{\partial E_{total}}{\partial hlout_3} * \frac{\partial hlout3}{\partial hlin_3} * \frac{\partial hlin3}{\partial W_{i3j3}} \end{bmatrix}$$

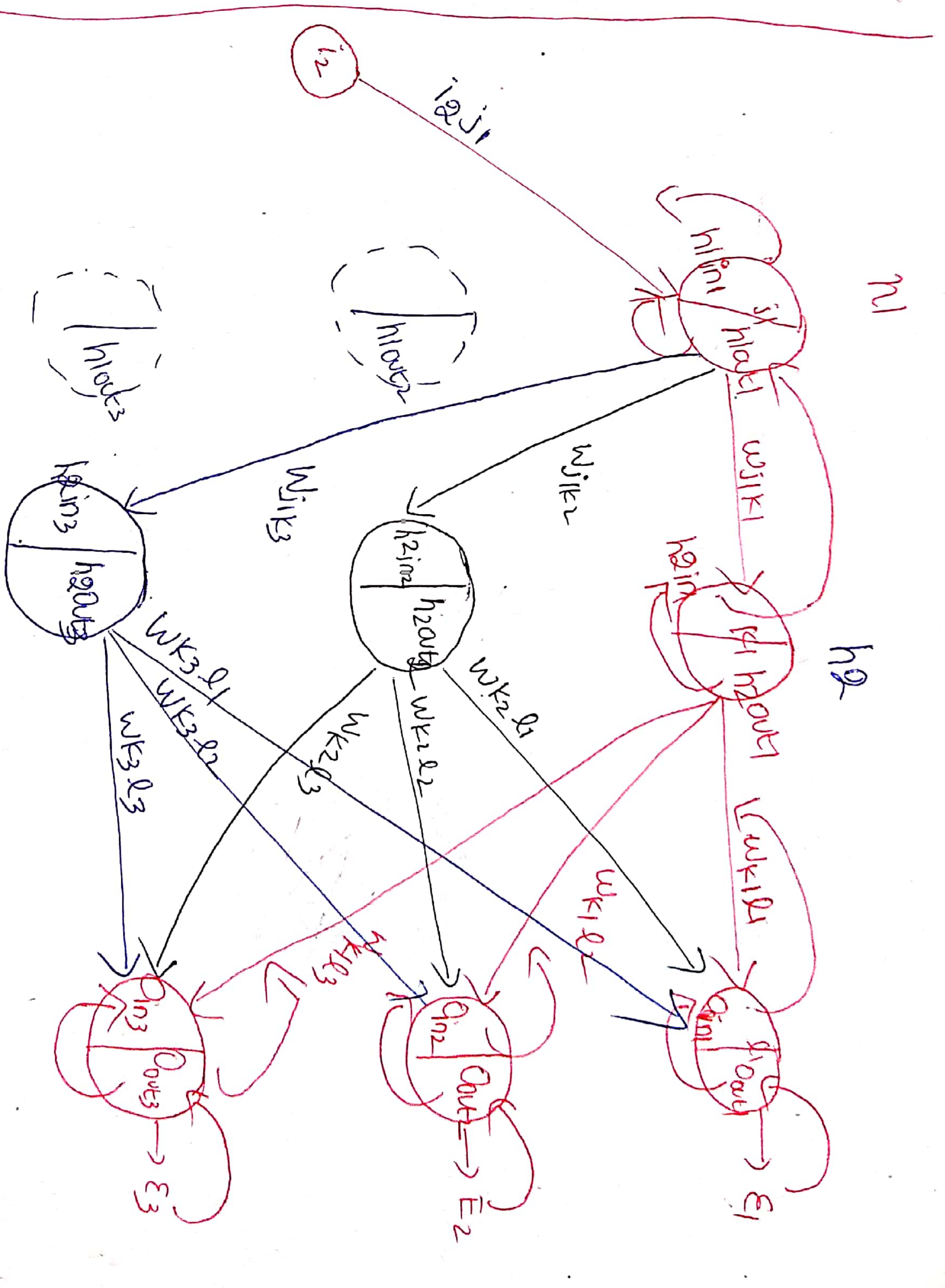
2nd & 3rd derivative already Calculated before

(B II.ii)

Network Structure in Detail



① $\frac{\partial E_1}{\partial W_{k1l1}}$



(I/P layer - HL1 weight adjustion)

Trying to compute the previous ticked one's

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h1_{out1}} \\ \frac{\partial E_{total}}{\partial h1_{out2}} \\ \frac{\partial E_{total}}{\partial h1_{out3}} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial E_{total}}{\partial h2_{out1}} * \frac{\partial h2_{out1}}{\partial h2_{in1}} * \frac{\partial h2_{in1}}{\partial h1_{out1}}\right) \\ \left(\frac{\partial E_{total}}{\partial h2_{out2}} * \frac{\partial h2_{out2}}{\partial h2_{in2}} * \frac{\partial h2_{in2}}{\partial h1_{out2}}\right) \\ \left(\frac{\partial E_{total}}{\partial h2_{out3}} * \frac{\partial h2_{out3}}{\partial h2_{in3}} * \frac{\partial h2_{in3}}{\partial h1_{out3}}\right) \end{bmatrix}$$

✗ → HAS 3 BRANCHES TAKE CARE

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h1_{out1}} \\ \frac{\partial E_{total}}{\partial h1_{out2}} \\ \frac{\partial E_{total}}{\partial h1_{out3}} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial E_{total}}{\partial h2_{out1}} * \frac{\partial h2_{out1}}{\partial h2_{in1}} * W_{j1k1}\right) + \left(\frac{\partial E_{total}}{\partial h2_{out2}} * \frac{\partial h2_{out2}}{\partial h2_{in2}} * W_{j1k2}\right) + \left(\frac{\partial E_{total}}{\partial h2_{out3}} * \frac{\partial h2_{out3}}{\partial h2_{in3}} * W_{j1k3}\right) \\ \left(\frac{\partial E_{total}}{\partial h2_{out1}} * \frac{\partial h2_{out1}}{\partial h2_{in1}} * W_{j2k1}\right) + \left(\frac{\partial E_{total}}{\partial h2_{out2}} * \frac{\partial h2_{out2}}{\partial h2_{in2}} * W_{j2k2}\right) + \left(\frac{\partial E_{total}}{\partial h2_{out3}} * \frac{\partial h2_{out3}}{\partial h2_{in3}} * W_{j2k3}\right) \\ \left(\frac{\partial E_{total}}{\partial h2_{out1}} * \frac{\partial h2_{out1}}{\partial h2_{in1}} * W_{j3k1}\right) + \left(\frac{\partial E_{total}}{\partial h2_{out2}} * \frac{\partial h2_{out2}}{\partial h2_{in2}} * W_{j3k2}\right) + \left(\frac{\partial E_{tot}}{\partial h2_{out3}} * \frac{\partial h2_{out3}}{\partial h2_{in3}} * W_{j3k3}\right) \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h1_{out1}} \\ \frac{\partial E_{total}}{\partial h1_{out2}} \\ \frac{\partial E_{total}}{\partial h1_{out3}} \end{bmatrix} = \begin{bmatrix} (-0.8030 * 0.05747 * 0.2) + (-0.5954 * 0.05598 * 0.3) + (-1.1426 * 0.01764 * 0.5) \\ (-0.8030 * 0.05747 * 0.3) + (-0.5954 * 0.05598 * 0.5) + (-1.1426 * 0.01764 * 0.7) \\ (-0.8030 * 0.05747 * 0.6) + (-0.5954 * 0.05598 * 0.4) + (-1.1426 * 0.01764 * 0.8) \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial E_{total}}{\partial h1_{out1}} \\ \frac{\partial E_{total}}{\partial h1_{out2}} \\ \frac{\partial E_{total}}{\partial h1_{out3}} \end{bmatrix} = \begin{bmatrix} (-0.00922) + (-0.00999) + (-0.01007) \\ (-0.01384) + (-0.01666) + (-0.01410) \\ (-0.02768) + (-0.01333) + (-0.01612) \end{bmatrix} = \begin{bmatrix} -0.02928 \\ -0.04127 \\ -0.05713 \end{bmatrix}$$

← 1st layer weights

$$\delta W_{ij} = \begin{bmatrix} \frac{\partial E_{total}}{\partial hlout_1} * \frac{\partial hlout_1}{\partial hlin_1} * \frac{\partial hlin_1}{\partial W_{i_1j_1}} & \frac{\partial E_{tot}}{\partial hlout_2} * \frac{\partial hlout_2}{\partial hlin_2} * \frac{\partial hlin_2}{\partial W_{i_1j_2}} & \frac{\partial E_{tot}}{\partial hlout_3} * \frac{\partial hlout_3}{\partial hlin_3} * \frac{\partial hlin_3}{\partial W_{i_1j_3}} \\ \frac{\partial E_{tot}}{\partial hlout_1} * \frac{\partial hlout_1}{\partial hlin_1} * \frac{\partial hlin_1}{\partial W_{i_2j_1}} & \frac{\partial E_{tot}}{\partial hlout_2} * \frac{\partial hlout_2}{\partial hlin_2} * \frac{\partial hlin_2}{\partial W_{i_2j_2}} & \frac{\partial E_{tot}}{\partial hlout_3} * \frac{\partial hlout_3}{\partial hlin_3} * \frac{\partial hlin_3}{\partial W_{i_2j_3}} \\ \frac{\partial E_{tot}}{\partial hlout_1} * \frac{\partial hlout_1}{\partial hlin_1} * \frac{\partial hlin_1}{\partial W_{i_3j_1}} & \frac{\partial E_{tot}}{\partial hlout_2} * \frac{\partial hlout_2}{\partial hlin_2} * \frac{\partial hlin_2}{\partial W_{i_3j_2}} & \frac{\partial E_{tot}}{\partial hlout_3} * \frac{\partial hlout_3}{\partial hlin_3} * \frac{\partial hlin_3}{\partial W_{i_3j_3}} \end{bmatrix}$$

$$\delta W_{ij} = \begin{bmatrix} -0.029289 * 1 * 0.1 & -0.04127 * 1 * 0.1 & -0.05713 * 1 * 0.1 \\ -0.029289 * 1 * 0.2 & -0.04127 * 1 * 0.2 & -0.05173 * 1 * 0.2 \\ -0.029289 * 1 * 0.7 & -0.04127 * 1 * 0.7 & -0.05173 * 1 * 0.7 \end{bmatrix}$$

$\delta W_{i_1j_1}$

$$\delta W_{ij} = \begin{bmatrix} -0.002928 & -0.004127 & -0.005713 \\ -0.005857 & -0.008254 & -0.011426 \\ -0.02050 & -0.028889 & -0.039991 \end{bmatrix}$$

$\delta W_{i_3j_3}$

∴ Let's update our weights with ($lr = 0.01$), we get final weight matrix as

$$\bar{W}_{ij} = \begin{bmatrix} W_{i_1j_1} - (lr * \delta W_{i_1j_1}) & W_{i_1j_2} - (lr * \delta W_{i_1j_2}) & W_{i_1j_3} - (lr * \delta W_{i_1j_3}) \\ W_{i_2j_1} - (lr * \delta W_{i_2j_1}) & W_{i_2j_2} - (lr * \delta W_{i_2j_2}) & W_{i_2j_3} - (lr * \delta W_{i_2j_3}) \\ W_{i_3j_1} - (lr * \delta W_{i_3j_1}) & W_{i_3j_2} - (lr * \delta W_{i_3j_2}) & W_{i_3j_3} - (lr * \delta W_{i_3j_3}) \end{bmatrix}$$

$$W_{ij} = \begin{bmatrix} 0.1 - (0.01 * -0.002928) & 0.2 - (0.01 * -0.004127) & 0.3 - (0.01 * -0.005713) \\ 0.3 - (0.01 * -0.005857) & 0.2 - (0.01 * -0.008254) & 0.7 - (0.01 * -0.011426) \\ 0.4 - (0.01 * -0.02050) & 0.3 - (0.01 * -0.028889) & 0.9 - (0.01 * -0.03199) \end{bmatrix}$$

$$\underset{(new)}{W_{ij}} = \begin{bmatrix} 0.10002928 & \begin{matrix} 0.40004127 \\ \cancel{0.399958} \end{matrix} & 0.30005713 \\ 0.30005857 & 0.20008254 & 0.7001142 \\ 0.400205 & 0.3002889 & 0.9003999 \end{bmatrix}$$

END OF CALCULATION

OUR INITIAL (OLD) WEIGHTS ARE :

$$W_{ij} = \begin{bmatrix} W_{i_1j_1} & W_{i_1j_2} & W_{i_1j_3} \\ W_{i_2j_1} & W_{i_2j_2} & W_{i_2j_3} \\ W_{i_3j_1} & W_{i_3j_2} & W_{i_3j_3} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.4 & 0.3 \\ 0.3 & 0.7 & 0.7 \\ 0.5 & 0.2 & 0.9 \end{bmatrix}$$

$$W_{jk} = \begin{bmatrix} W_{j_1k_1} & W_{j_1k_2} & W_{j_1k_3} \\ W_{j_2k_1} & W_{j_2k_2} & W_{j_2k_3} \\ W_{j_3k_1} & W_{j_3k_2} & W_{j_3k_3} \end{bmatrix} = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.3 & 0.5 & 0.7 \\ 0.6 & 0.4 & 0.8 \end{bmatrix}$$

$$W_{kl} = \begin{bmatrix} W_{k_1l_1} & W_{k_1l_2} & W_{k_1l_3} \\ W_{k_2l_1} & W_{k_2l_2} & W_{k_2l_3} \\ W_{k_3l_1} & W_{k_3l_2} & W_{k_3l_3} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.4 & 0.8 \\ 0.3 & 0.7 & 0.2 \\ 0.5 & 0.2 & 0.9 \end{bmatrix}$$

$$input = [\,0.1 \quad 0.2 \quad 0.7\,]$$

OUR FINAL UPDATED WEIGHTS ARE

$$\Rightarrow \hat{W}_{ij} = \begin{bmatrix} 0.10002928 & 0.40004127 & 0.30005713 \\ 0.30005857 & 0.20008254 & 0.7001142 \\ 0.400205 & 0.3002889 & 0.9003999 \end{bmatrix}$$

$$\Rightarrow \hat{W}_{jk} = \begin{bmatrix} 0.200623 & 0.30044 & 0.50027 \\ 0.300586 & 0.500423 & 0.70025 \\ 0.600830 & 0.400599 & 0.800362 \end{bmatrix}$$

$$\Rightarrow \hat{W}_{kl} = \begin{bmatrix} 0.107518 & 0.402677 & 0.807138 \\ 0.307534 & 0.702683 & 0.207153 \\ 0.507854 & 0.202797 & 0.907458 \end{bmatrix}$$

$$\left.\begin{matrix} Exp \\ o/p \end{matrix}\right\} = [\,1 \quad 0 \quad 0\,]$$

NOW WITH THE NEWLY UPDATED WEIGHTS, LETS DO THE FORWARD PROPAGATION.

Matrix operations : Layer - 1

Bias

$$[i_1\ i_2\ i_3] \begin{bmatrix} W_{i1j1} & W_{i1j2} & W_{i1j3} \\ W_{i2j1} & W_{i2j2} & W_{i2j3} \\ W_{i3j3} & W_{i2j2} & W_{i3j3} \end{bmatrix} + [b_{j1}\ b_{j2}\ b_{j3}] = [h1_{in1}\ h1_{in2}\ h1_{in3}]$$

Relu operation :

$$[h1_{out1}\ h1_{out2}\ h1_{out3}] = [\mathrm{Max}(0, h1_{in1})\ \ \mathrm{Max}(0, h1_{in2})\ \ \mathrm{Max}(0, h1_{in3})]$$

O/p shape = 1x3

Ex: (with updated weights)

$$\underset{1\times3}{[0.1\ \ 0.2\ \ 0.7]} \times \underset{3\times3}{\begin{bmatrix} 0.10002928 & 0.40004127 & 0.30005713 \\ 0.30005857 & 0.20008254 & 0.7001142 \\ 0.400205 & 0.3002889 & 0.9003999 \end{bmatrix}} + [1\ 1\ 1]$$

$$[h1_{out1}\ h1_{out2}\ h1_{out3}] = [1.350158\ \ 1.270223\ \ 1.800308]$$

— end of layer - 1 —

Matrix operations : Layer - 2

$$[h1_{out1}\ h1_{out2}\ h1_{out3}]\begin{bmatrix} W_{j1k1} & W_{j1k2} & W_{j1k3} \\ W_{j2k1} & W_{j2k2} & W_{j2k3} \\ W_{j3k1} & W_{j3k2} & W_{j3k3} \end{bmatrix} + B = (h2_{in1}\ h2_{in2}\ h2_{in3})$$

Sigmoid operation : $\dfrac{1}{1+e^{-x}}$

$$[h2_{out1}\ h2_{out2}\ h2_{out3}] = \left[\frac{1}{1+e^{-h2_{in1}}}\ \ \frac{1}{1+e^{-h2_{in2}}}\ \ \frac{1}{1+e^{-h2_{in3}}}\right]$$

$$[1.350158\ \ 1.270223\ \ 1.800308] \times \begin{bmatrix} 0.200623 & 0.30044 & 0.50027 \\ 0.300586 & 0.500423 & 0.70025 \\ 0.600830 & 0.400599 & 0.800362 \end{bmatrix} + [1\ 1\ 1]$$

$$[h2_{in1}\ h2_{in2}\ h2_{in3}] = [2.73436\ \ 2.76249\ \ 4.00582]$$

$\downarrow$ Put in Sigmoid

$$[h2_{out1}\ h2_{out2}\ h2_{out3}] = [0.9390\ \ 0.94061\ \ 0.98211]$$

Matrix operations : Layer - 3

$$\begin{bmatrix} h2out_1 & h2out_2 & h2out_3 \end{bmatrix} \times \begin{bmatrix} Wk_1l_1 & Wk_1l_2 & Wk_1l_3 \\ Wk_2l_1 & Wk_2l_2 & Wk_2l_3 \\ Wk_3l_1 & Wk_2l_2 & Wk_3l_3 \end{bmatrix} + B$$

$$= \begin{bmatrix} O_{in1} & O_{in2} & O_{in3} \end{bmatrix}$$

Softmax Operation :

$$\text{Softmax} = \frac{e^{O_{ina}}}{\sum_{a=1}^{3} e^{O_{ina}}}$$

$$\begin{bmatrix} O_{out1} & O_{out2} & O_{out3} \end{bmatrix} = \begin{bmatrix} \frac{e^{O_{in1}}}{\sum e^{O_{ina}}} & \frac{e^{O_{in2}}}{\sum e^{O_{ina}}} & \frac{e^{O_{in3}}}{\sum e^{O_{ina}}} \end{bmatrix}$$

$$\begin{bmatrix} 0.9390 & 0.94061 & 0.98211 \end{bmatrix} \times \begin{bmatrix} 0.107518 & 0.402677 & 0.807138 \\ 0.307534 & 0.702683 & 0.207153 \\ 0.507854 & 0.202797 & 0.907468 \end{bmatrix} + [1\ 1\ 1]$$

$$= \begin{array}{ccc} O_{in1} & O_{in2} & O_{in3} \\ \end{array}$$
$$\begin{bmatrix} 1.8889 & 2.2382 & 2.84399 \end{bmatrix}$$

$$\begin{bmatrix} O_{out1} & O_{out2} & O_{out3} \end{bmatrix} = \begin{bmatrix} 0.19932 & 0.2826 & 0.5180 \end{bmatrix}$$

[—— end of o/p layer]

$$O_{out1} = \frac{e^{O_{in1}}}{e^{O_{in1}} + e^{O_{in2}} + e^{O_{in3}}} = \frac{e^{1.8889}}{e^{1.8889} + e^{2.2382} + e^{2.84399}} = \frac{6.612091}{33.1727}$$

$$O_{out1} = 0.19932$$

B15

The Actual outputs should be $\begin{array}{ccc} y_1 & y_2 & y_3 \\ \end{array}$ $[\,1 \quad 0 \quad 0\,]$

→ Old weights outputs $= [0.1985 \quad 0.2855 \quad 0.5158]$

→ Newly updated weights outputs $= [0.19932 \quad 0.2826 \quad 0.5180]$

inc ↑ ↓ (dec) ↓

∴ Error :

$$\text{Cross entropy} = \left(\frac{-1}{n}\right)\left[\sum_i y_i \times \log(O_{out\,i}) + (1-y_i) \times \log(1-O_{out\,i})\right]$$

Ex:

$$CE(\text{error}) = (-1)\left[(1)\log(0.19932) + \cancel{(1-1)\log}(1-0.19932) + \cancel{(0)\log}(0.2826) + (1-0)\log(1-0.2826) + \cancel{(0)\log}(0.5180) + (1-0)\log(1-0.5180)\right]$$

$$= (-1)\left[\log(0.19932) + \log(0.7174) + \log(0.482)\right]$$

$$= (-1)\left[-0.700449 - 0.144238 - 0.3169\right]$$

$$= (-1)(-1.16163)$$

$CE(\text{Error})\big|_{\text{old weights}} = 1.16321$

$CE(\text{Error})\big|_{\text{New weights}} = 1.16163$

Error getting reduced a little.

— END OF BACKPROPAGATION —