Loss Functions

- MSE $= \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}$

- MAE $= \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$

- Squared error $= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

- cross entropy → logistic Regression

$$\text{Loss} = -y * \log(\hat{y}) - (1-y) * (1-\hat{y})$$

for binary cross entropy.

$$\begin{cases} -\log(1-\hat{y}) & \text{if } y=0 \\ -\log(\hat{y}) & \text{if } y=1 \end{cases}$$

- Multi class cross entropy loss

$$L(x_i, y_i) = -\sum_{j=1}^{c} y_{ij} * \log(\hat{y}_{ij})$$

categorical cross entropy.

c → no. of classes in o/p

o/p is multiclass e.g. [good, bad, neutral]

	f1	f2	f3	o/p	One hot encoding b1	b2	b3
e.g.	2	3	4	Good	1	0	0
	4	5	7	Bad	0	1	0
	3	8	9	Neutral	0	0	1

y_{ij}

o/p y_i is one hot encoded vector.

$$Y_i = \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $y_{ij} = \begin{cases} 1 & \text{if } i^{th} \text{ element is in class } j \\ 0 & \text{else} \end{cases}$

How to calculate $\hat{y}_{ij}$

use softmax Activation function $\sigma(z)$

$$\sigma(z) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

e.g. Against vector [10, 20, 30, 40]

for 10 $\sigma(z) = \frac{e^{10}}{e^{10}+e^{20}+e^{30}+e^{40}}$

for 20 $\sigma(z) = \frac{e^{20}}{e^{10}+e^{20}+e^{30}+e^{40}}$

ie. probability

Tanh

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1 = 2\,\text{sigmoid}(2x) - 1$$

$$f'(x) = 1 - f(x)^2$$

Sigmoid

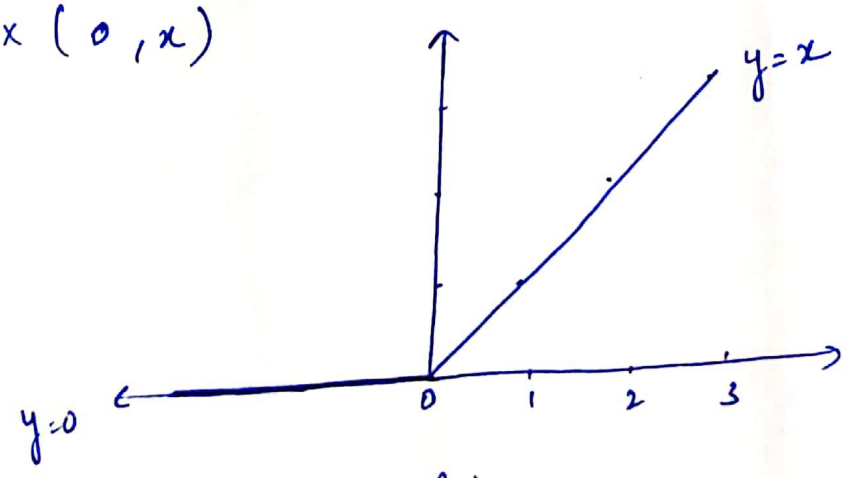
$$f(x) = \frac{1}{1+e^{-x}}$$

$$f'(x) = \frac{f(x)}{1 - f(x)} \qquad f'(x) = f(x)\,(1 - f(x))$$

Relu — Rectified Linear Unit

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \qquad f'(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

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



Relu

Leaky Relu

$$f(x) = \max(0.01 * x, x)$$

$$f(x) = \begin{cases} x & x \geq 0 \\ 0.01 * x & x \leq 0 \end{cases}$$

Softmax

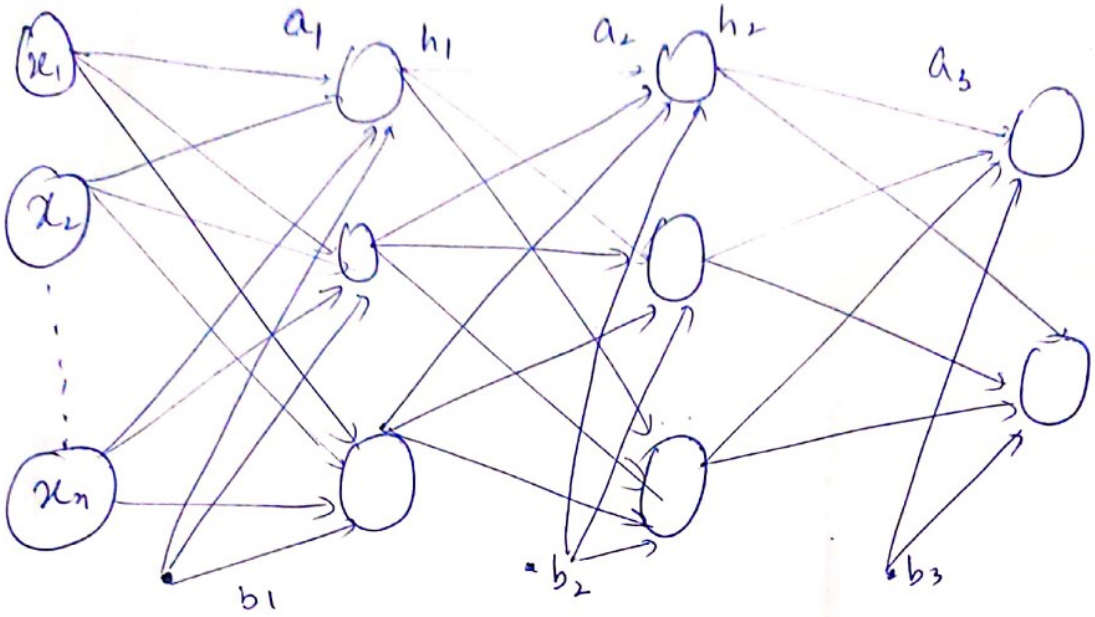
Probability distribution

$$f(x) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

e.g.

Car	bike	scooter
0.7	0.2	0.1

Feed Forward N/W



I/P layer (L-1) hidden layers

here it is 2

Properties

1. n-dimensional vector as input to the n/w
2. L-1 hidden layers having n neurons each.
3. One O/P Layer with K neurons
4. Each Neuron in hidden Layer & O/P Layer has 2 forms → pre-activation (a_i) & activation (h_i)
5. I/P Layer is 0^{th} Layer, O/P Layer is L^{th} Layer
6. $W_i \in \mathbb{R}^{n \times n}$ and $b_i \in \mathbb{R}^n$ are the weights and bias b/w Layers i-1 and i $(0 < i < L)$

→ $W_L \in \mathbb{R}^{n \times K}$ and $b_L \in \mathbb{R}^K$ are the weight and bias b/w the last hidden layer and the o/p layer ($L = 3$ in this case)

(1) Pre-activation fxn at any layer i

$$a_i(x) = b_i + W_i h_{i-1}(x)$$

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \end{bmatrix} \begin{bmatrix} W_{111} & W_{112} & W_{113} \\ W_{121} & W_{122} & W_{123} \\ W_{131} & W_{132} & W_{133} \end{bmatrix} \begin{bmatrix} h_{01} = x_1 \\ h_{02} = x_2 \\ h_{03} = x_3 \end{bmatrix}$$

$$\underset{(3\times1)}{a_1} = \underset{(3\times1)}{b_1} + \underset{(3\times3)\cdot(3\times1)}{W_1 h_0}$$

$$= \begin{bmatrix} \sum W_{11i} x_i + b_{11} \\ \sum W_{12i} x_i + b_{12} \\ \sum W_{13i} + b_{13} \end{bmatrix}$$

(2) Activation at layer i is

$$h_i(x) = g(a_i(x))$$

$$g(a_i) = \sigma(a_i) = \frac{1}{1+e^{-a_i}}$$

Sigmoid fxn

$$\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \end{bmatrix} = g\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} = \begin{bmatrix} g(a_{11}) \\ g(a_{12}) \\ g(a_{13}) \end{bmatrix}$$

$$g(a_{11}) = \frac{1}{e^{-a_{11}}+1}$$

$g \rightarrow$ activation fxn.

(logistic, tanh, linear, Relu ....)

(3) Output at L^{th} layer

$$y = f(x) = h_L(x) = O(a_L(x))$$

$O \rightarrow$ output activation fxn (softmax etc.)

(4) Data $= \{x_i, y_i\}_{i=1}^{N}$

(5) Model :

$$\hat{y} = f(x_i) = O\left(W^3 g\left(W^2 g\left(W^1 x + b_1\right) + b_2\right) + b_3\right)$$

(6) Parameters

$\theta = W_1, W_2 \dots W_L$ or size $\mathbb{R}^{n \times n}$

$b = b_1, b_2 \dots b_L$ size $\mathbb{R}^{n \times 1}$

(7) Algorithm

Gradient Descent with BP

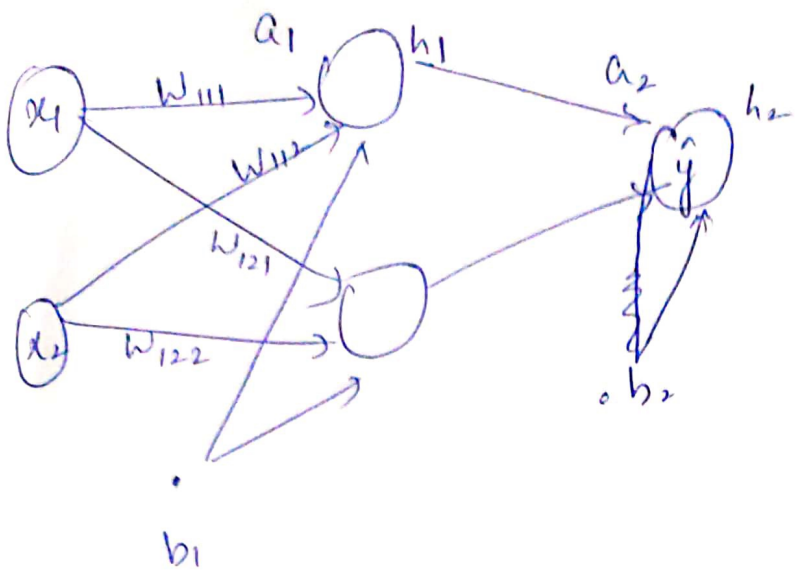
(8) Loss / Error Fxn / Objective Fxn

$$\min \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} \left(\hat{y}_{ij} - y_{ij}\right)^2$$

[sum of squared fxn]

$$= \min \mathcal{L}(\theta)$$

Single hidden layer feed Forward example.



x_1	x_2
0.1	0.3
0.7	0.2
0.2	0.4

$$a_1 = b_1 + W_1 h_0$$

(1)
$$\begin{bmatrix} a_{11} \\ a_{12} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{12} \end{bmatrix} + \begin{bmatrix} W_{111} & W_{112} \\ W_{121} & W_{122} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

a_1 : $\begin{bmatrix} a_{11} \\ a_{12} \end{bmatrix}$; b_1 : $\begin{bmatrix} b_{11} \\ b_{12} \end{bmatrix}$; W_1 : $\begin{bmatrix} W_{111} & W_{112} \\ W_{121} & W_{122} \end{bmatrix}$; h_0 : $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Randomly selected parameters

$$W_1 = \begin{bmatrix} 0.5 & 0.4 \\ 0.9 & 1.0 \end{bmatrix}$$

$$b_1 = \begin{bmatrix} -0.1 \\ -0.1 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} \\ a_{12} \end{bmatrix} = \begin{bmatrix} -0.1 \\ -0.1 \end{bmatrix} + \begin{bmatrix} 0.5 & 0.4 \\ 0.9 & 1.0 \end{bmatrix} \begin{bmatrix} 0.1 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 0.07 \\ 0.29 \end{bmatrix}$$

(2) $h_1 = g(a_1) = g\begin{bmatrix} 0.07 \\ 0.29 \end{bmatrix} = \begin{bmatrix} \sigma(0.07) \\ \sigma(0.29 \end{bmatrix}$

$$\sigma(0.07) = \frac{1}{1+e^{-0.07}}$$

$$\sigma(0.29) = \frac{1}{1+e^{-0.29}}$$

(3) $$\begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} = \begin{bmatrix} b_{21} \\ b_{22} \end{bmatrix} + \begin{bmatrix} W_{211} & W_{212} \\ W_{221} & W_{222} \end{bmatrix} \begin{bmatrix} \sigma(0.07) \\ \sigma(0.29) \end{bmatrix}$$

(4) $y_{11} = h_2 = O(a_2) = O\begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix}$

$O \rightarrow$ softmax.

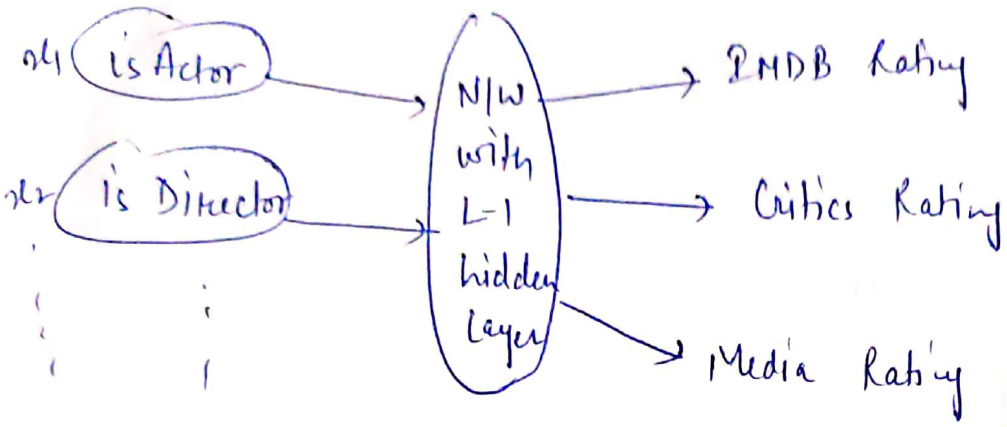
h_2 is final predicted o/p denoted as $\hat{y}$

(5) $\text{error} = \sum_{i=1}^{2} (y - \hat{y})^2$

choice of loss function

→ depends on problem in hand.

eg. ①



y_i
e.g. $\{ \underset{7.5}{y_{i1}} \quad \underset{8.2}{y_{i2}} \quad \underset{7.7}{y_{i3}} \}$

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{3} \left(\hat{y}_{ij} - y_{ij} \right)^2$$

O/P fuction shoud be such that it should not be bounded b/w 0 to 1.

∴ cant be Logistic

Next option is O/P fxn to be linear

$$f(x) = h_L = O(a_L)$$
$$= W_{O a_L} + b_O$$

∴ Which O/P function with suit.

Linear. As the Problem is <u>Regression</u>

kind of Problem

Example 2 $y_1 = \begin{matrix} R & B & W \\ [0.35 & 0.25 & 0.4] \end{matrix}$

True Probability :| drawing a ball from a knapsack with color Red/Blw/White.

suppose $\hat{y}_1 = [0.25 \quad 0.45 \quad 0.3]$

$$L(\theta) = \sum_{i=1}^{3} (y_i - \hat{y}_i)^2$$

@ I/P function should be such which should return bounded value b/w $0 \to 1$.

* can we use Sigmoid as it returns b/w $0 \to 1$

No. Because the probability distribution should sum to 1.

Softmax is the solution.

$$a_L = b_L + W_L \cdot h_{L-1}$$

$$\hat{y}_j = O(a_L)_j = \frac{e^{a_{L,j}}}{\sum_{i=1}^{K} e^{a_{L,i}}}$$

e.g. $a_L = \begin{matrix} a_{L1} & a_{L2} & a_{L3} \\ [10 & -20 & 30] \end{matrix}$

$\hat{y} = [\hat{y}_1 \quad \hat{y}_2 \quad \hat{y}_3]$

$$\hat{y}_i = \frac{e^{10}}{e^{10} + e^{-20} + e^{30}}$$

[Softmax fun]

y, $\hat{y}$ both are probability distributions

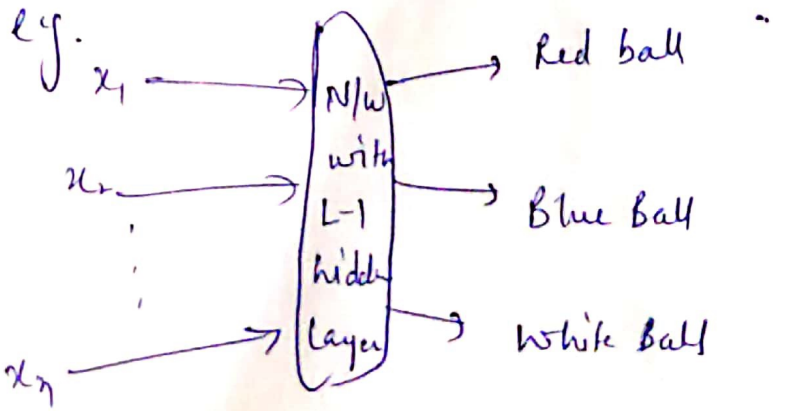
∴ $\mathcal{L}(\theta)$ preferred will be cross entropy rather than squared error.

$$\mathcal{L}(\theta) = -\sum_{i=1}^{K} y_i \log \hat{y}_i$$

$y_i = 1$ if i = true class (t)

$= 0$ otherwise

∴ $\mathcal{L}(\theta) = -\log \hat{y}_t$



$y_i = [1 \quad 0 \quad 0]$
R B W

means ball drawn is Red, the true class here.

∴ for classification Problems where O/P is one of the classes between 1 and K

objective fxn is

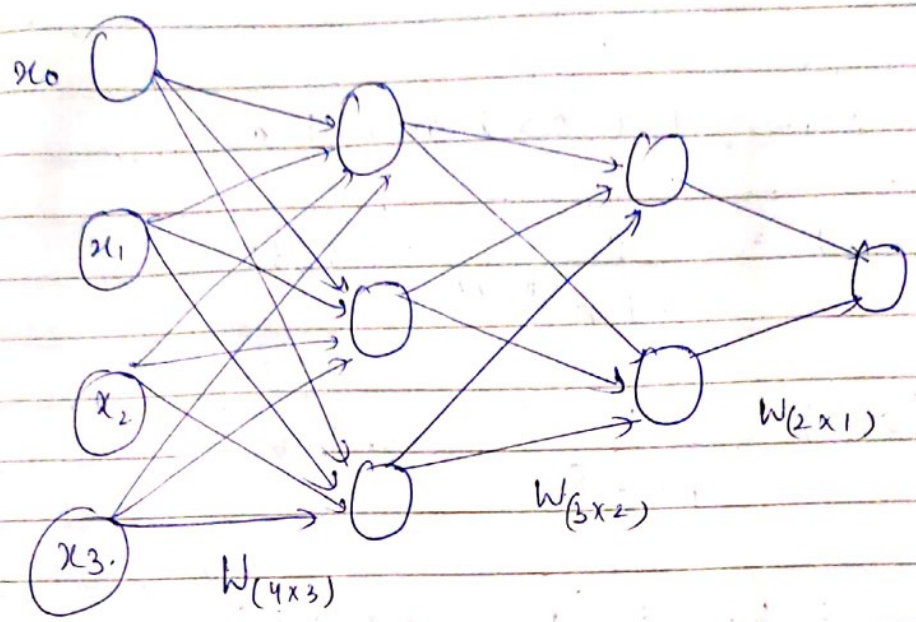
$$\text{minimize}_{\theta}\ \mathcal{L}(\theta) = -\log \hat{y}_t$$

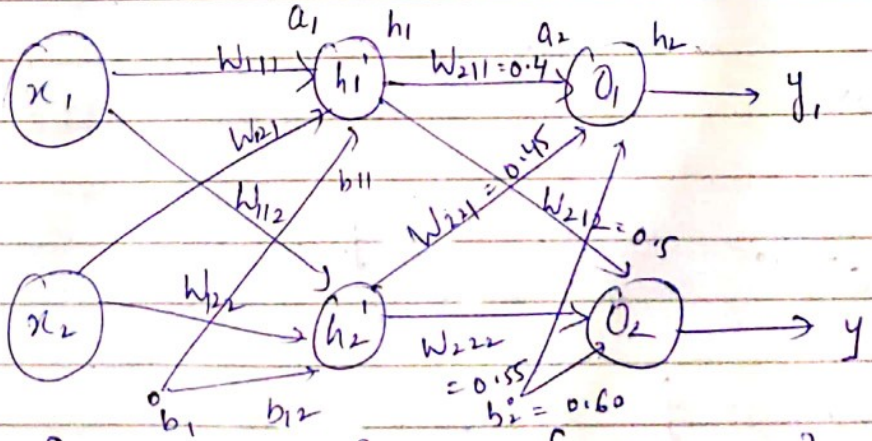
or

$$\text{maximize}_{\theta}\ -\mathcal{L}(\theta) = +\log \hat{y}_t$$

$$\theta = [W_1, W_2, \ldots, W_L, b_1, b_2, \ldots, b_L]$$

Back Propagation Example





$$\begin{bmatrix} W_{111} & W_{112} \\ W_{121} & W_{122} \end{bmatrix} \cdot \begin{bmatrix} 0.15 & 0.25 \\ 0.20 & 0.30 \end{bmatrix}$$

For layer $i=1$:

$$\begin{bmatrix} a_{11} \\ a_{12} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{12} \end{bmatrix} + \begin{bmatrix} W_{111} & W_{112} \\ W_{121} & W_{122} \end{bmatrix}^T \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$a_{11} = b_{11} + (W_{111} * x_1 + W_{121} * x_2)$$

$$a_{12} = b_{12} + (W_{112} * x_1 + W_{122} * x_2)$$

$$a_{11} = ((0.15 * 0.05) + (0.2 * 0.1)) + 0.35$$

$$= 0.377$$

$$a_{12} = ((0.25 * 0.05) + (0.3 * 0.1)) + 0.35$$

$$h_{11} = \sigma(a_{11}) = \frac{1}{1+e^{-0.377}} = 0.3932$$

$$h_{12} = \sigma(a_{12}) = \frac{1}{1+e^{-a_{12}}} = 0.5968$$

$$a_{21} = ((w_{211} * h_{11}) + (w_{221} * h_{22})) + b_{21}$$

$$= ((0.4 * 0.5932) + (0.45 * 0.5968) + 0.6$$

$$= 1.105$$

$$h_{21} = \frac{1}{1+e^{-1.105}} = 0.7513 = \hat{y}_1$$

$$h_{22} = 0.7729 = \hat{y}_2$$

$$\text{error} = \frac{\sum_{i=1}^{2} (y_i - \hat{y}_i)^2}{n}$$

$$= \frac{(0.01 - 0.7513)^2 + (0.99 - 0.7729)^2}{2}$$

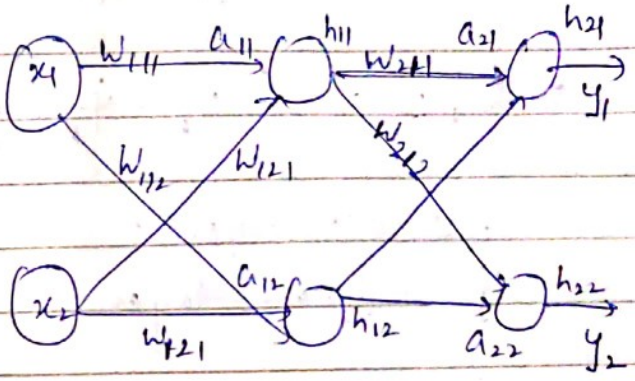
$$\text{error} = 0.2983$$

Minimize error

$$\frac{\partial \text{Error}}{\partial W_{211}} = \frac{\partial \text{Error}}{\partial y_1} * \frac{\partial y_1}{\partial a_{21}} * \frac{\partial a_{21}}{\partial W_{211}}$$

$$\frac{\partial \text{Error}}{\partial y_1} = \frac{\partial \text{Error}}{\partial h_{21}}$$

$$= \frac{\partial \left(\frac{1}{2} (y_1 - h_{21})^2 \right)}{\partial h_{21}}$$



[Apply chain rule]

(1) $$\frac{\partial \text{error}}{\partial h_{21}} = -(y_1 - h_{21})$$

OR

$$\boxed{\frac{\partial \text{error}}{\partial \hat{y}_K} = -(y_K - \hat{y}_K)}$$

(2) $$\frac{\partial y_1}{\partial a_{21}} = \frac{\partial (1 + e^{-y_1})^{-1}}{\partial a_{21}} = \frac{e^{-y_1}}{(1 + e^{-y_1})^2}$$

$$= \left(1 - \frac{1}{1 + e^{-y_1}}\right) \frac{1}{1 + e^{-y_1}}$$

$$\boxed{= y_1 (1 - y_1) \quad \text{or} \quad \frac{\partial y_1}{\partial a_{21}} = h_{21}(1 - h_{21})}$$

$$\frac{\partial a_{21}}{\partial w_{211}} = \frac{\partial\left((w_{211} * h_{11} + w_{221} * h_{12}) + b_{11}\right)}{\partial w_{211}}$$

$$= h_{11}$$

$$\nabla w_{211} = -(y_1 - h_{21}) \cdot h_{21}(1 - h_{21}) \cdot h_{11}$$

$$\nabla w^{l}_{kj} = -(y_k - \hat{y}_k)\,\hat{y}_k\,(1 - \hat{y}_k)\,h^{l}$$

~~k → o/p layer~~

~~j → hidden layer~~

$$\nabla w_{lij} = -(y_{actual_i} - h_{li}) \cdot h_{li}(1 - h_{li}) \cdot h_{i,l-1}$$

$l \rightarrow$ layer No.

$i \rightarrow$ Neuron no. I/P

$j \rightarrow$ output neuron no.

error = 0.29837

$$\frac{\partial \text{error}}{\partial W_{211}} = \frac{\partial \text{error}}{\partial h_{21}} * \frac{\partial h_{21}}{\partial a_{21}} * \frac{\partial a_{21}}{\partial W_{211}}$$

$$(-(0.01 - 0.751365)) * (0.751365 * (1 - 0.751365)) * 0.5932$$

$$\nabla W_{211} = 0.08216704$$

New updated $W_{211} = W_{211} - \alpha \nabla W_{211}$

$$= 0.4 - (0.60) * (0.08216704)$$

$$= 0.35069976$$

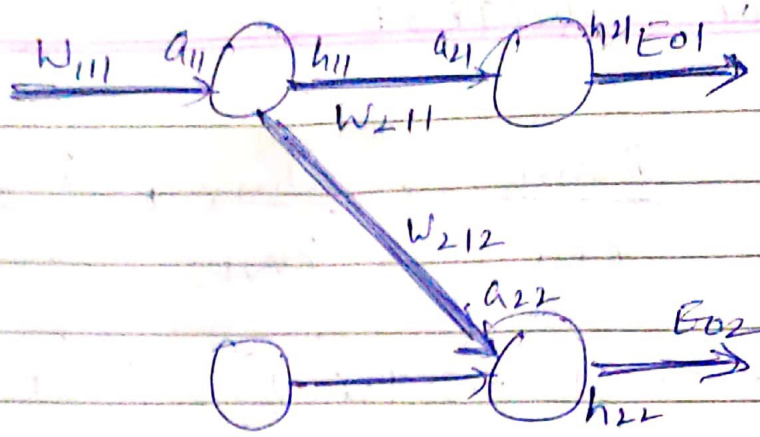
Likewise update

W_{212}, W_{221}, W_{222}.

* B/w I/P & hidden layer

Now solving W_{111}, W_{112}, W_{121}, W_{122}

$$\frac{\partial \text{error}}{\partial W_{111}} = \frac{\partial \text{error}}{\partial h_{11}} * \frac{\partial h_{11}}{\partial a_{11}} * \frac{\partial a_{11}}{\partial W_{111}}$$



$$\frac{\partial error}{\partial w_{111}} = \frac{\partial error}{\partial h_{11}} * \frac{\partial h_{11}}{\partial a_{11}} * \frac{\partial a_{11}}{\partial w_{111}}$$

$$\frac{\partial error}{\partial h_{11}} = \frac{\partial E_{01}}{\partial h_{11}} + \frac{\partial E_{02}}{\partial h_{11}}$$

$$\frac{\partial E_{01}}{\partial h_{21}} * \frac{\partial h_{21}}{\partial a_{21}} * \frac{\partial a_{21}}{\partial h_{11}} \quad : \quad \frac{\partial E_{02}}{\partial h_{22}} * \frac{\partial h_{22}}{\partial a_{22}} * \frac{\partial a_{22}}{\partial h_{11}}$$

↑ calculated above

$$= -(y_{actual_1} - h_{21}) \cdot h_{21}(1-h_{21}) * \frac{\partial((W_{211} * h_{11} + W_{221} * h_{12}) + b_{11})}{\partial h_{11}}$$

$$= -(y_{actual_1} - h_{21}) \cdot h_{21}(1-h_{21}) * W_{211}$$

$$\frac{\partial E_{02}}{\partial h_{22}} * \frac{\partial h_{22}}{\partial a_{22}} * \frac{\partial a_{22}}{\partial h_{11}}$$

$$-(y.actual_2 - h_{22}) \cdot h_{22}(1 - h_{22}) * w_{212} \quad \text{---(11)}$$

put values in (1)

$$(0.13849857 * 0.4) + (-0.03850982 * 0.5)$$

$$= 0.055399425 + (-0.01904919)$$

$$\frac{\partial error}{\partial h_{11}} = 0.036350306.$$

$$\frac{\partial h_{11}}{\partial a_{11}} = h_{11}(1 - h_{11})$$

$$= 0.241300709$$

$$\frac{\partial a_{11}}{\partial w_{111}} = \frac{\partial (w_{111} * x_1 + w_{121} * x_2 + b_{11})}{\partial w_{111}}$$

$$= x_1 = 0.05$$

$$\nabla w_{111} = w_{111} - \alpha \frac{\partial error}{\partial w_{111}}$$

$$= 0.15 - (0.6) * 0.000438568$$

$$= 0.149736859 2$$