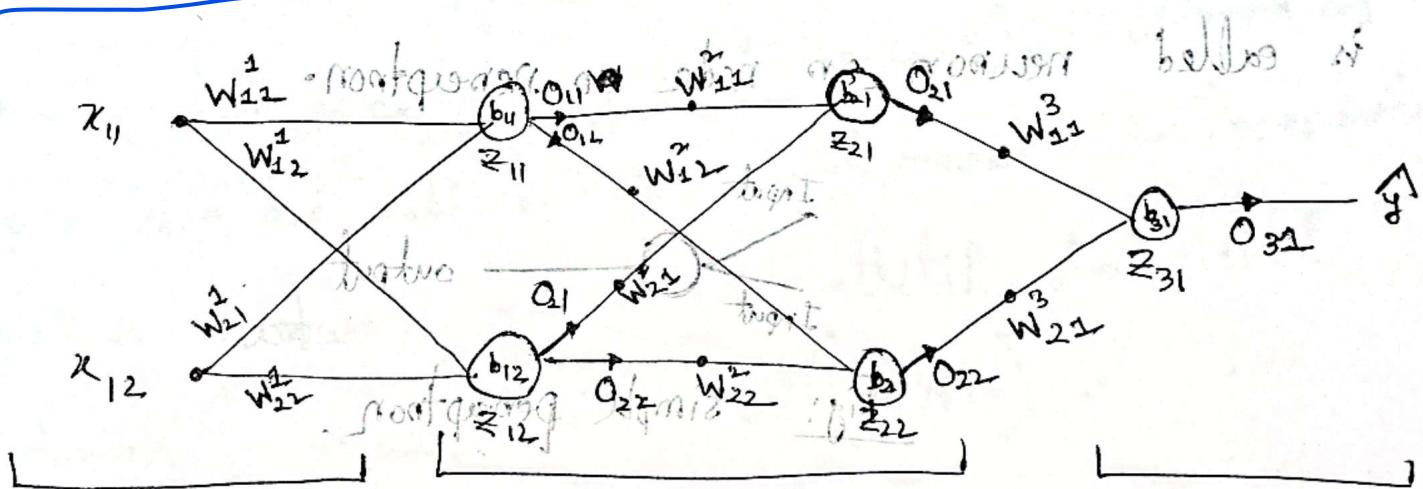


Neural Network

1. Component of a neural network and discuss the similarities of neural network.

A neural network is a computation model inspired by human brain. And its component are:

- (i) Input layer
- (ii) Hidden layers
- (iii) Output layer
- (iv) Neuron Node
- (v) Weights
- (vi) Bias
- (vii) Activation function
- (viii) Loss function
- (ix) Optimizer



input layer

hidden layer

output layer

Backpropagation:

Fig: simple artificial neural network

$$(\text{Loss} \rightarrow g \rightarrow z_{31} \rightarrow o_{21} \rightarrow z_{21} \rightarrow o_{11} \rightarrow z_{11} \rightarrow w_{11})$$

computational graph for updating w_{11}

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial L}{\partial g} \times \frac{\partial g}{\partial z_{31}} \times \frac{\partial z_{31}}{\partial o_{21}} \times \frac{\partial o_{21}}{\partial z_{21}} \times \frac{\partial z_{21}}{\partial o_{11}} \times \frac{\partial o_{11}}{\partial z_{11}} \times \frac{\partial z_{11}}{\partial w_{11}}$$

Hannan's Law

$$Z_{11} = X_{11} W_{11}^1 + X_{21} W_{21}^1 + b_{11}$$

activating self loops b_{11} from Hannan layer \rightarrow to frenoprod

Activation function (Z_{11})

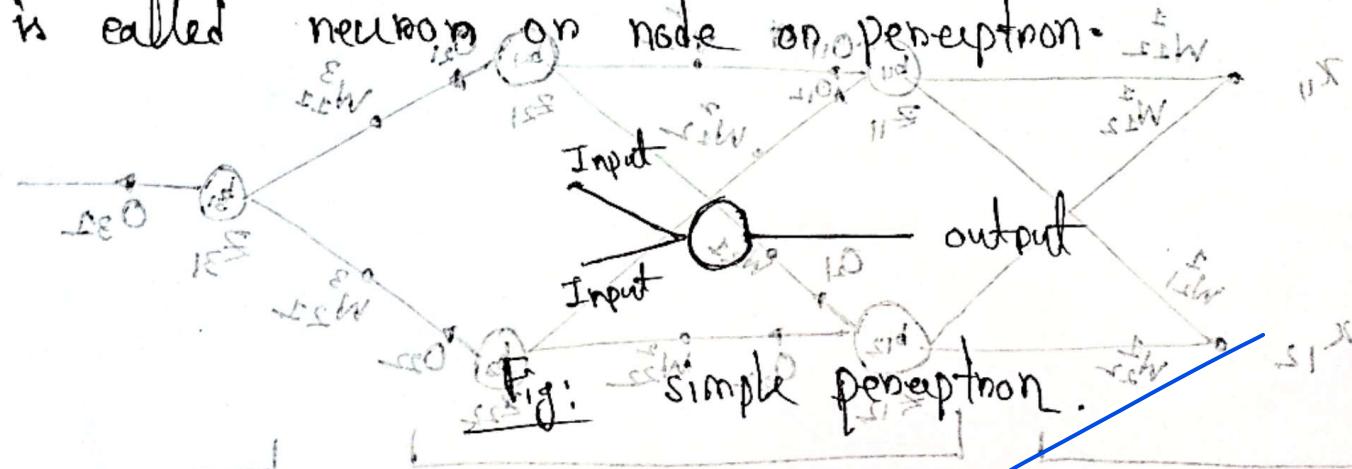
Hannan layer \rightarrow

$Z_{21} = \text{Opri} W_{12}^2 + \text{Opri} W_{22}^2 + b_{21}$ to brain Hannan layer A

Original Activation function (Z_{21}) to frenoprod with brain visual

(iii) with (iv) 2 stages (v) start multiple (vi) neurons on node 9 + 5
Here, the ANN made up multiple neurons on node 9 + 5
classifying (xi) with prob 2nd (iiii) a robust network (xii) of a neural network

is called neuron or node in perceptron.



tucks



Neuron 1

Synapse

Neuron 2

All the neurons

in brain has

billion of neurons

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

10^16

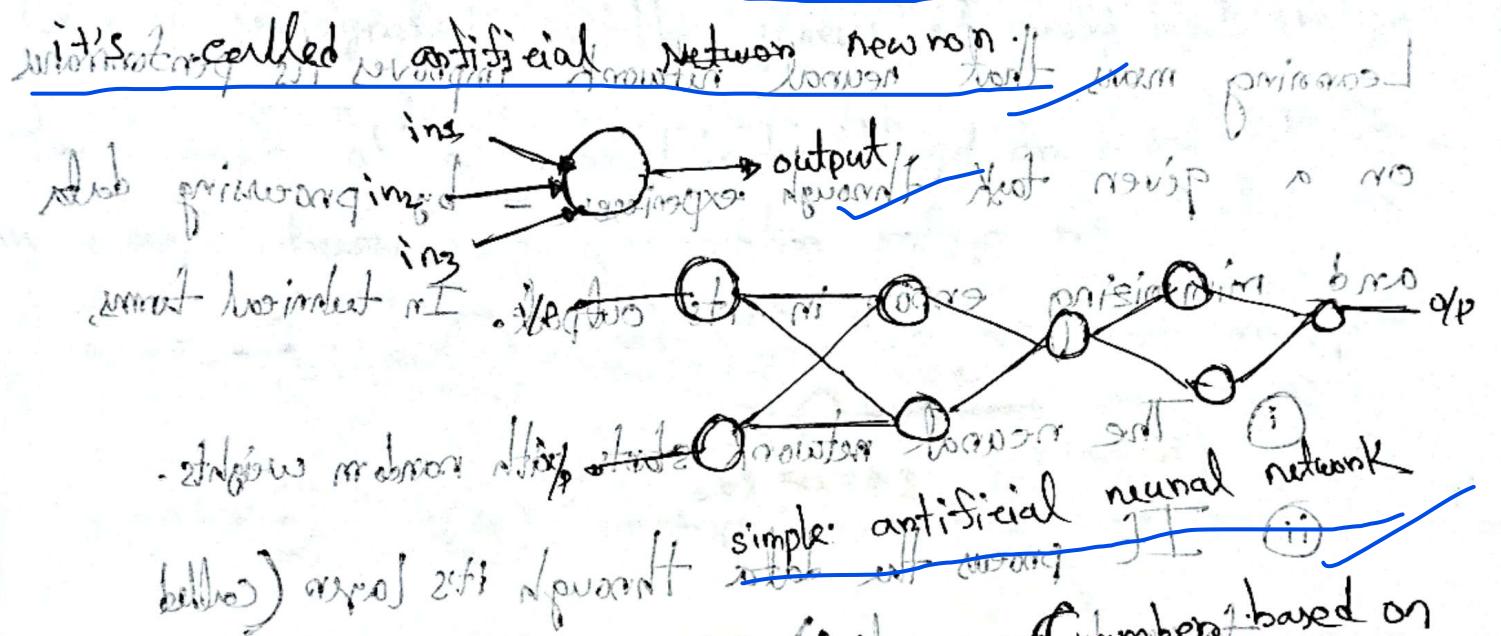
10^16

10^16

10^16

10

interconnected with each other making a big network called neural network. "Similarly, a perceptron or



Similarly, thousand of artificial neurons (number based on input data and model complexity) interconnected with each other make a network and we call it Artificial Network. Neural Network (ANN) is

function that takes input and provides output

Ques :-

(perceptron). below unit gives wrong output (iv)

~~What does a neural network do? How does it work?~~

2. What does "learning". It means adjusting to neural nets.

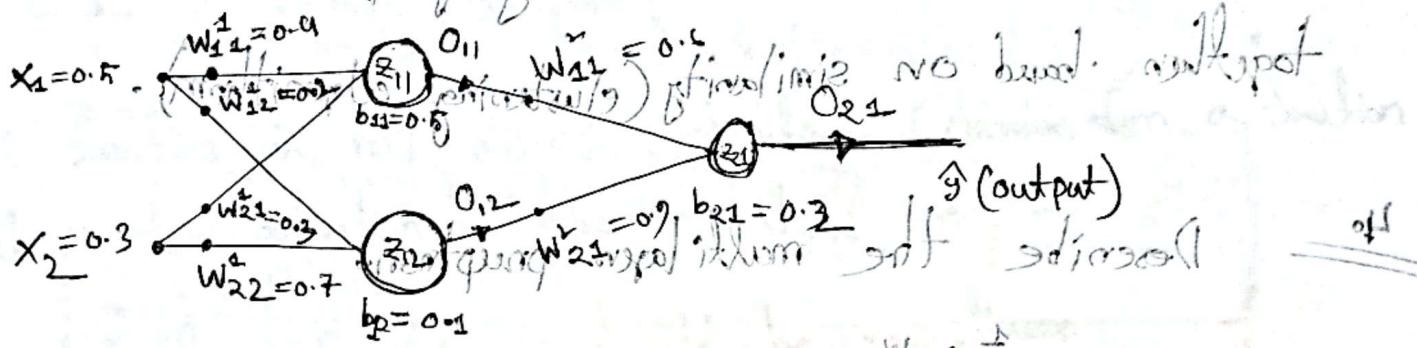
Learning means that ~~forward~~ ^{backward} ~~network~~ ^{loss} improves its performance on a given task through ~~experience~~ ^{experience} by processing data and minimizing error in its output. In technical terms,

- i The neural network starts with random weights.
- ii It processes the data through its layers (called forward propagation) with the intention to draw it parallel to the target output. The target output is compared with the expected output. However, the output is wrong.
- iii The difference between them is error or loss.
- iv Using the backpropagation and gradient descent, the neural network adjusts its weights.
- v Repeating this process many times called (training).

Q3

Explain forward propagation and self-organization.

Forward propagation (propagation in the) process of moving inputs through the layers of a neural network to get an output. Let's take an example below showing illustrations make it easier to understand.



Data set

Input 1	Input 2	Tan(θ)
0.5	0.3	1

and, let's take

activation function as sigmoid.

$$O_{11} = \frac{1}{1 + e^{-z_{11}}} = \frac{1}{1 + e^{-(0.5 \times 0.4 + 0.3 \times 0.2 + 0.5)}} = 0.68$$

$$O_{21} = \frac{1}{1 + e^{-z_{21}}} = \frac{1}{1 + e^{-(0.68 \times 0.9 + 0.3 + 0.3)}} = 0.76$$

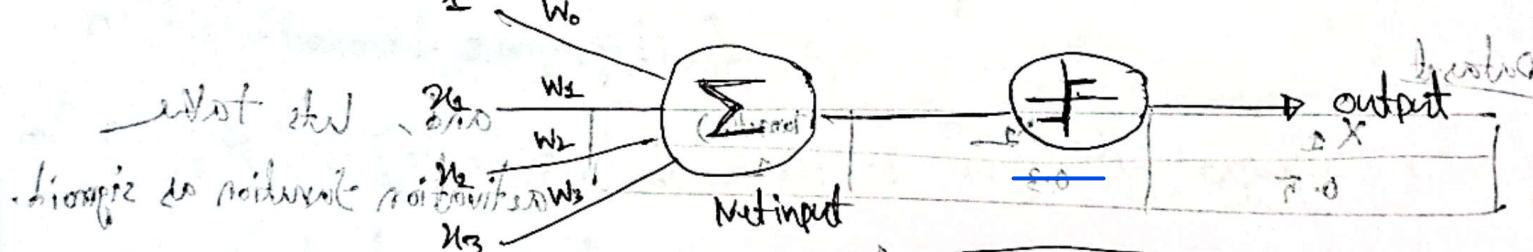
$$Y = O_{21} = 0.76$$

The process of getting output from input is called forward propagation.

~~Self-Organization~~ \rightarrow Self organization is a learning process without supervision (Unsupervised learning). The network ~~itself~~ organizes data itself and finds a pattern of data. The system automatically groups similar inputs together based on similarity (clustering algorithm).

~~40~~

Describe the multi-layered perceptron.



$$y_i = f_i = F(\sum w_j x_j + b) = F(\sum w_j x_j + \sum b_j) = F(\sum (w_j + b_j)x_j) = F(\sum \Delta_j x_j) = F(\Delta)$$

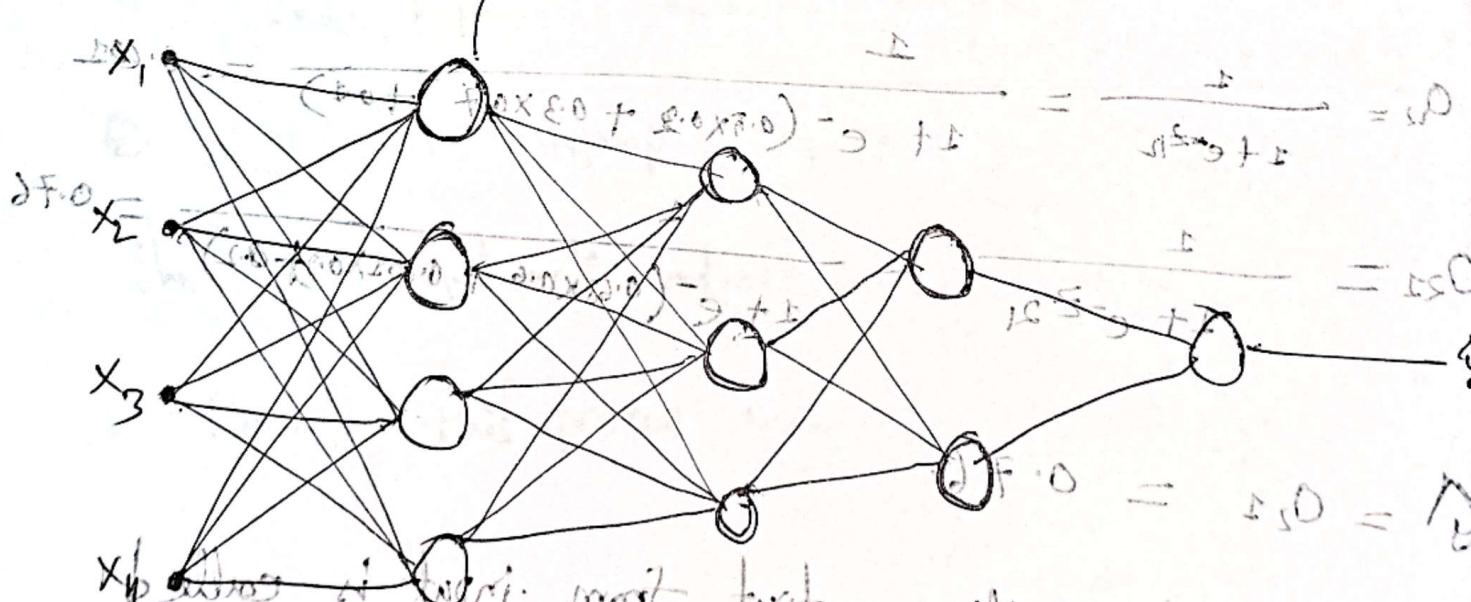


Fig: Multi-layer Perceptron (MLP)

Q5: Illustrate and derive the standard activation function.

~~element of first without probabilities~~

In artificial neural networks each weighted sum of input passes through a ~~scalar~~ value function called activation function.

All function is not activation function. Criterion for a function to be activation function.

on step-function = ReLU

should be non-linear.

i Activation function

(sigmoid)

$\frac{x_1 - x_2}{x_3 + x_4} = \text{ReLU}$

ii Activation function

(tanh)

should be differentiable.

iii Activation function

(softmax)

$\frac{x_1}{x_2 + x_3} = \text{ReLU}$

iv Activation function

(ReLU)

numerically inexpensive.

v Activation function

(elu)

$\frac{x_1}{x_2 + x_3} = \text{ReLU}$

vi Activation function

(softmax)

zero centered

vii Activation function

(softmax)

non-saturating

means bound within ~~a~~ range. Like, saturating function sigmoid output range ($0 \sim 1$) and tanh output range ($-1, 1$). For very very higher input sigmoid give always 1 same for tanh that why it's faces vanishing gradient problem.

~~• Various activation functions are discussed in short~~

Activation function with formula

Function	Formula	Range	Remarks
sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	(0, 1)	Prone to vanishing gradient
ReLU	$f(x) = \max(0, x)$	(0, ∞)	may cause "dead neuron" when input $x \leq 0$
Leaky ReLU	$f(x) = \begin{cases} x, & x > 0 \\ 0.01x, & x \leq 0 \end{cases}$	(- ∞ , ∞)	Solve the dead neuron problem
tanh activation	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	(-1, 1)	zero-centered (for normalization) fast converges than sigmoid
softmax	$f(x) = \frac{e^{x_i}}{\sum_j e^{x_j}}$	(0, 1)	Converts output into probabilities

otherwise = 0

1

1

N

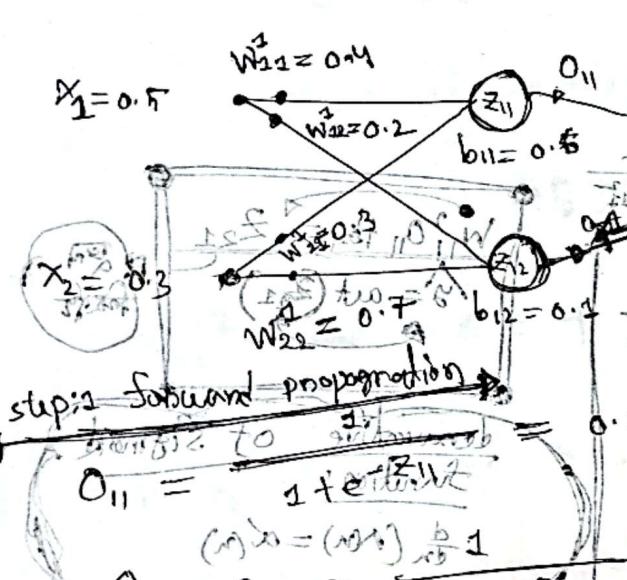
1

gradient will zero for right hand side
but the left side ($+ve$) spread both because natural
log is the log of natural log not $(+/-)$ other
function with which it is not. note

Q. How to adjust the weights of perceptron in it?

on state the back-propagation algorithm for learning in a multilayer network.

Let's take the below mlps.



step 1: forward propagation

$$O_{11} = \frac{1}{1 + e^{-w_{11}x_1 + b_{11}}} = \frac{1}{1 + e^{-0.4 \cdot 0.5 + 0.5}} = 0.68$$

step 2: calculate loss

$$\text{loss} = (y - O_{21})^2 = 0.076$$

$$(y - O_{21}) = (y - 0.76) = \frac{16}{66}$$

$$(y - O_{21}) \cdot \text{bias} = \frac{16}{66}$$

$$(y - O_{21}) \cdot w = \frac{16}{66}$$

$$[(y - O_{21}) \cdot w] \cdot [(y - O_{21}) \cdot w] =$$

$$(y - O_{21})^2 =$$

$$\begin{aligned} & \text{Dataset} \quad \text{target} \\ & \begin{array}{|c|c|c|c|} \hline & x_1 & x_2 & y \\ \hline 1 & 0.5 & 0.3 & 1 \\ \hline \end{array} \\ & \text{step 1: } \text{forward propagation} \quad \text{step 2: } \text{backward propagation} \\ & \text{Input} \quad \text{Output} \\ & \begin{array}{|c|c|c|c|} \hline & w_{11} & w_{12} & b_{11} & O_{11} \\ \hline 1 & 0.4 & 0.2 & 0.5 & 0.68 \\ \hline & w_{21} & w_{22} & b_{21} & O_{21} \\ \hline 2 & 0.6 & 0.3 & 0.2 & 0.76 \\ \hline \end{array} \\ & \text{loss} = (y - O_{21})^2 = (1 - 0.76)^2 = 0.076 \end{aligned}$$

$$0.076 \times (dF/dy) \times dF/dw =$$

$$0.076 \times 0.76 =$$

$$(0.076 - 0.76) \times 0.76 =$$

$$-0.508 \times 0.76 =$$

We have to define soft edges of wall

$$x_{11} = 0.5 \quad x_{12} = 0.3 \quad y = 1 \text{ (target)}$$

$$O_{11} = 0.688 \quad O_{12} = 0.601, \quad g = O_{21} = 0.76$$

$$W_{11}^2 = 0.6, \quad W_{21}^2 = 0.9$$

Now we update

~~$b_{21} = 0.2$~~ , Let's $\delta = 0.57$ from wall soft edge ab.

i) W_{11}^2 update:

$$W_{11}^2_{\text{new}} = W_{11}^2 - \delta \Delta L$$

$$\frac{\partial L}{\partial W_{11}^2} = \frac{\partial L}{\partial g} \times \frac{\partial g}{\partial z_{21}} \times \frac{\partial z_{21}}{\partial W_{11}^2}$$

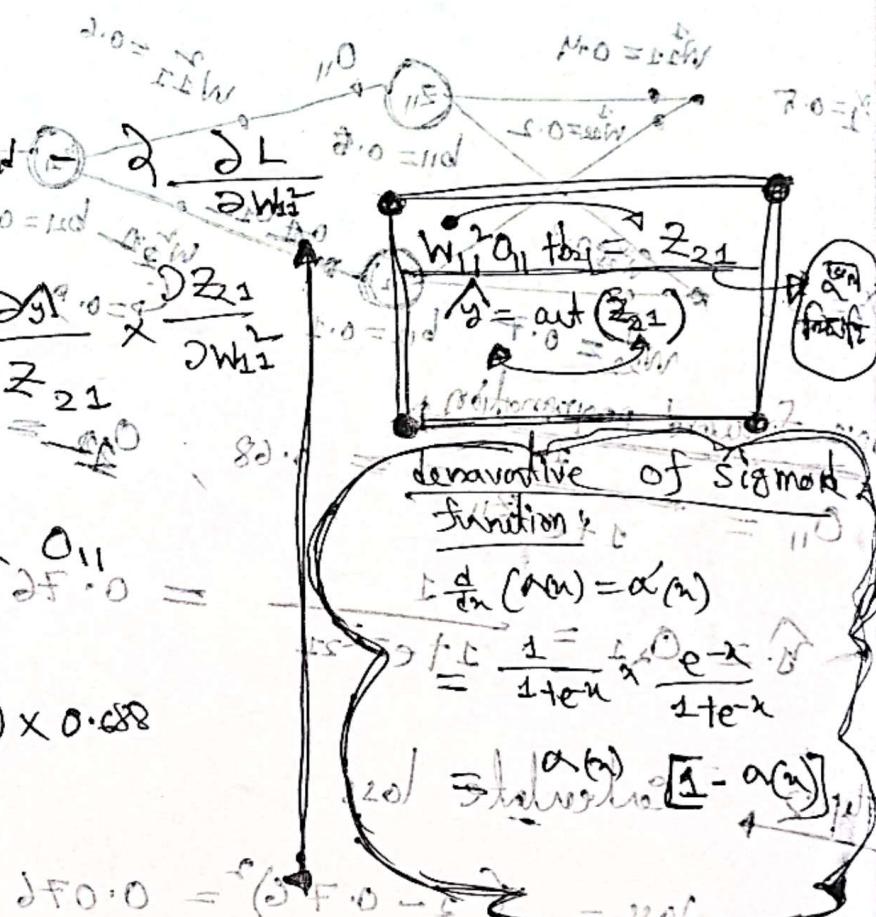
$$= -2(y - \hat{y}) \times g'(z - \hat{y}) \times O_{11}$$

$$= -2(1 - 0.76) \times 0.76 \times (1 - 0.76) \times 0.688$$

$$= -0.0602$$

$$W_{11}^2 = 0.6 - (0.5 \times -0.0602)$$

$$= 0.6301$$



$$\Delta L = (\hat{y} - y)^2 = -2(y - \hat{y})$$

$$\hat{y} = \text{sigmoid}(z_{21})$$

$$\frac{d\hat{y}}{dz_{21}} = \alpha'(z_{21})$$

$$= \alpha(z_{21}) [1 - \alpha(z_{21})]$$

$$= \hat{y} (1 - \hat{y})$$

(ii) For, w_{21}^1 ,

$$z_{21} = o_{11} w_{12}^1 + o_{12} w_{22}^1 + b_{21} \quad \text{--- (i)}$$

$$\hat{y} = \text{sigmoid}(z_{21}) \quad \text{--- (ii)}$$

$$\frac{\partial L}{\partial w_{21}^1} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial z_{21}} \times \frac{\partial z_{21}}{\partial w_{21}^1}$$

$$= -2(y - \hat{y}) \times \hat{y} \times (1 - \hat{y}) \times o_{12}$$

$$= -2(1 - 0.76) \times 0.76 (1 - 0.76) \times 0.60^1$$

$$= -0.053$$

$$w_{21}^1 = 0.9 + (0.5)(0.053) = 1.165$$

Similarly, we will adjust the value of b_{21} , b_{11} , b_{12} , w_{11}^1 , w_{12}^1 through back propagation.