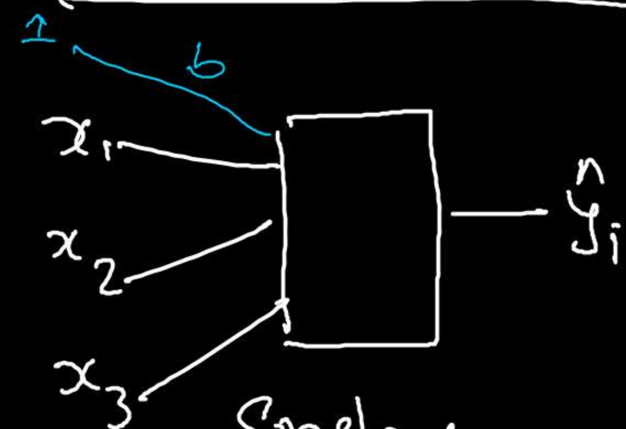
Deep Neural Netw

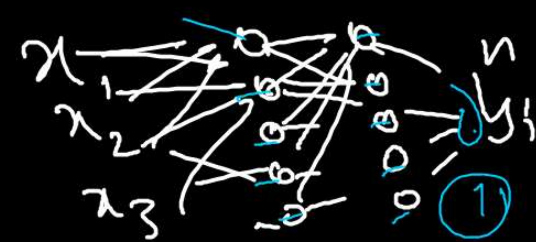
Agenda :-

1. Notation
2. Backpropogation method
3. Data Normalization
4. early stopping
5. case study
6. Dropout

Multi Layer perceptron

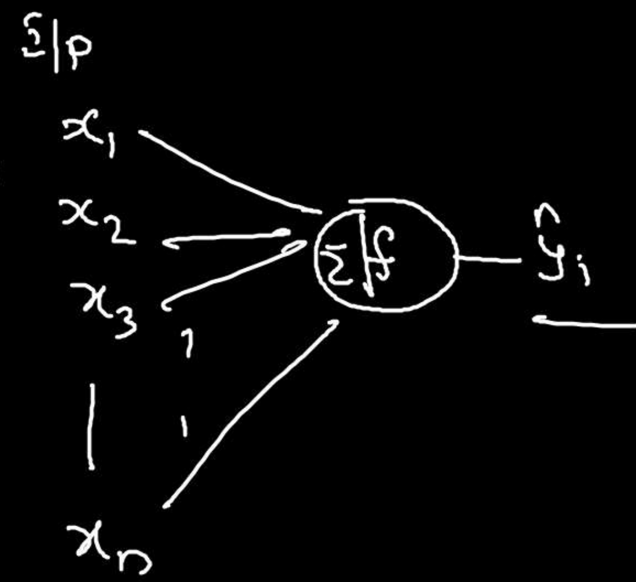


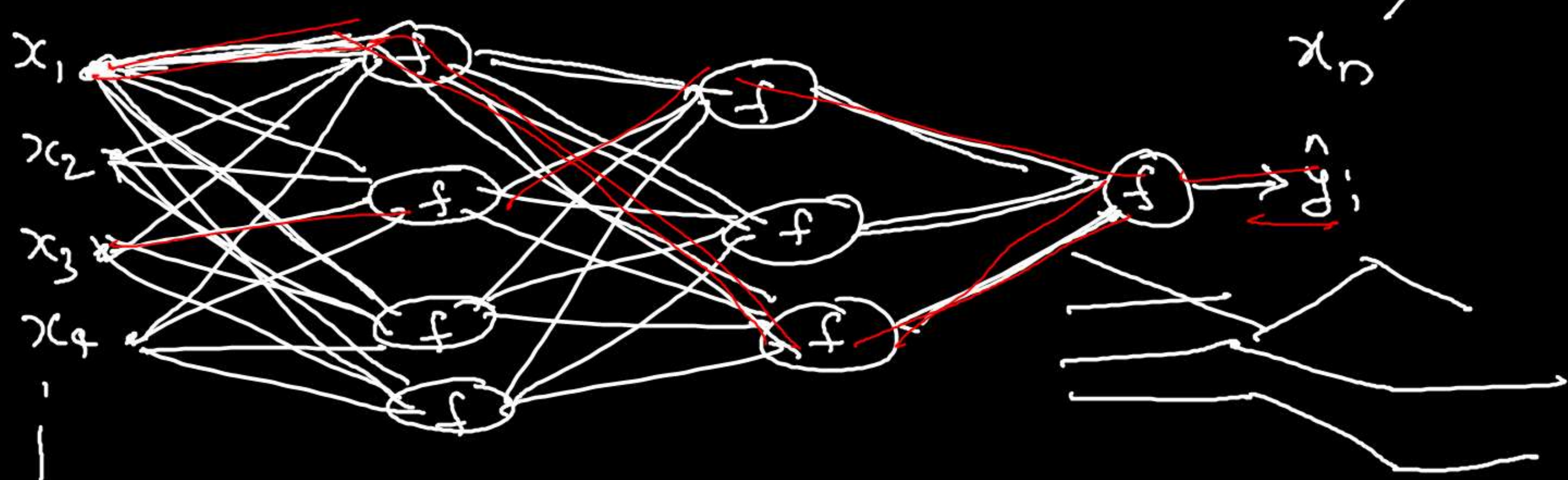
Single Layer



Multilayer Perceptron

Perceptron — Single Neuron →





Q:- Why should we care about MLP? — Intern —

① Biological Inspiration

Neuroscience — human, rats, ants, monkeys etc.

— Neural Structure Network —

② Mathematical Arguments :-

regression :- $\{x_i, y_i\}$, $\boxed{x_i \in \mathbb{R}^n, y \in \mathbb{R}}$ ← Real Num

eg :- $\boxed{2 * \sin(x^2) + \text{sqrt}(x * 5)}$

LR :-

$y = m_1x_1 + c + m_2x_2 + m_3x$

plot $(2 * \sin(x^2) + \sqrt{5x})$

$y = f(x)$

x_1	y
1	1
2	4
3	9

square(x)

$y = f(x)$

NN

Let,

f_1 = add()

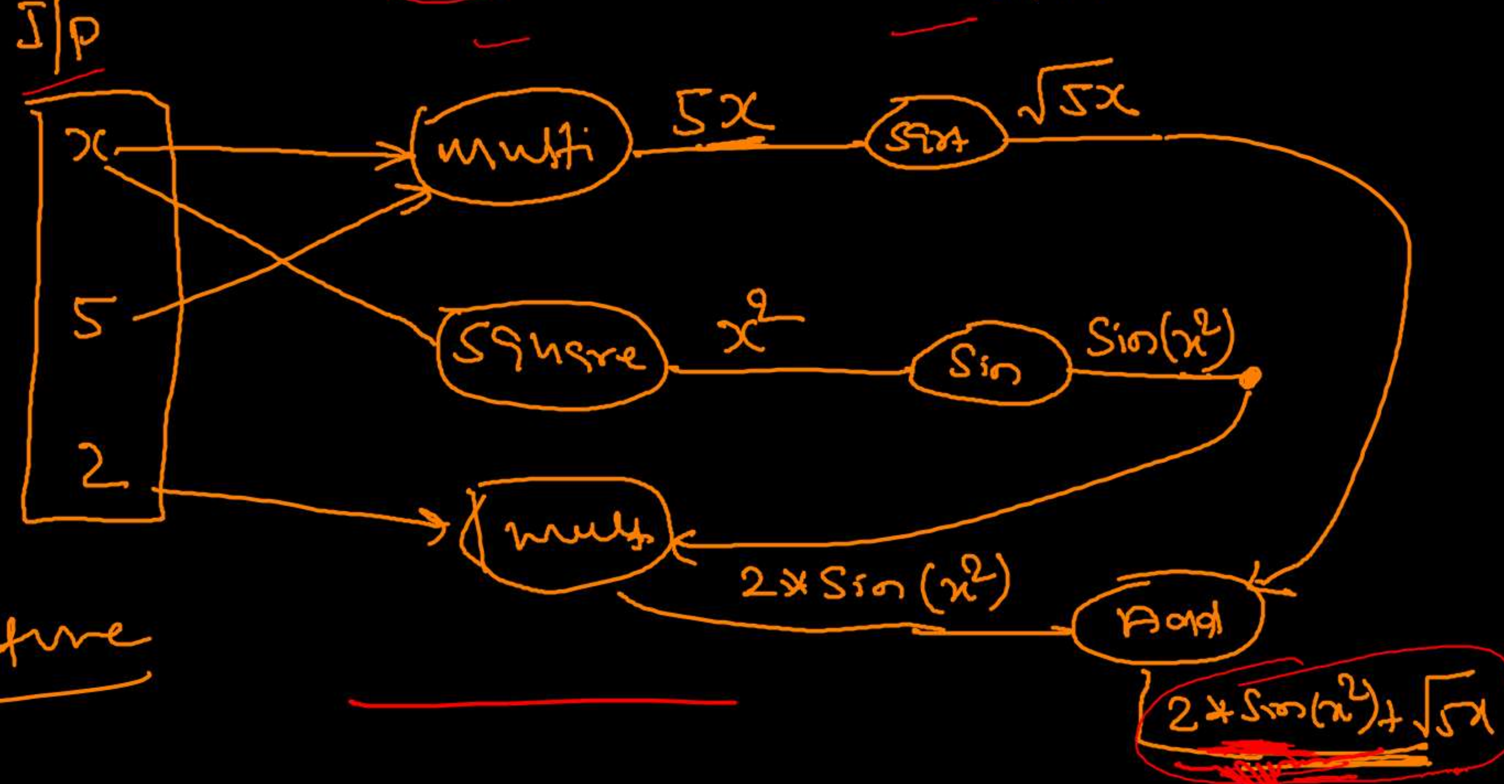
f_2 = square()

f_3 = sqrt()

f_4 = sin()

f_5 = multi()

MLP- structure



high-school :- Functional composition

$f(g(x)) = f \circ g\,(x)$ $\quad$ $g \circ f(x) \rightarrow g(f(x))$

$f1 = add$

$f2 = square$

$f3 = sqrt$

$f4 = sin$

$f5 = multi$

$$\boxed{f(x) = 2 * \sin(x^2) + sqrt(x * 5)}$$

$\sqrt{x * 5}$

$x * 5 = f5(x, 5)$

$sqrt(x * 5) = f3(f5(x, 5))$

$x^2 = f2(x)$

$\sin(x^2) = f4(f2(x))$

$2 * \sin(x^2) = f5(2, f4(f2(x)))$

$2 * \sin(x^2) + sqrt(x * 5)$

$= f_1(f5(2, f4(f2(x))), f_3(f5(x, 5)))$

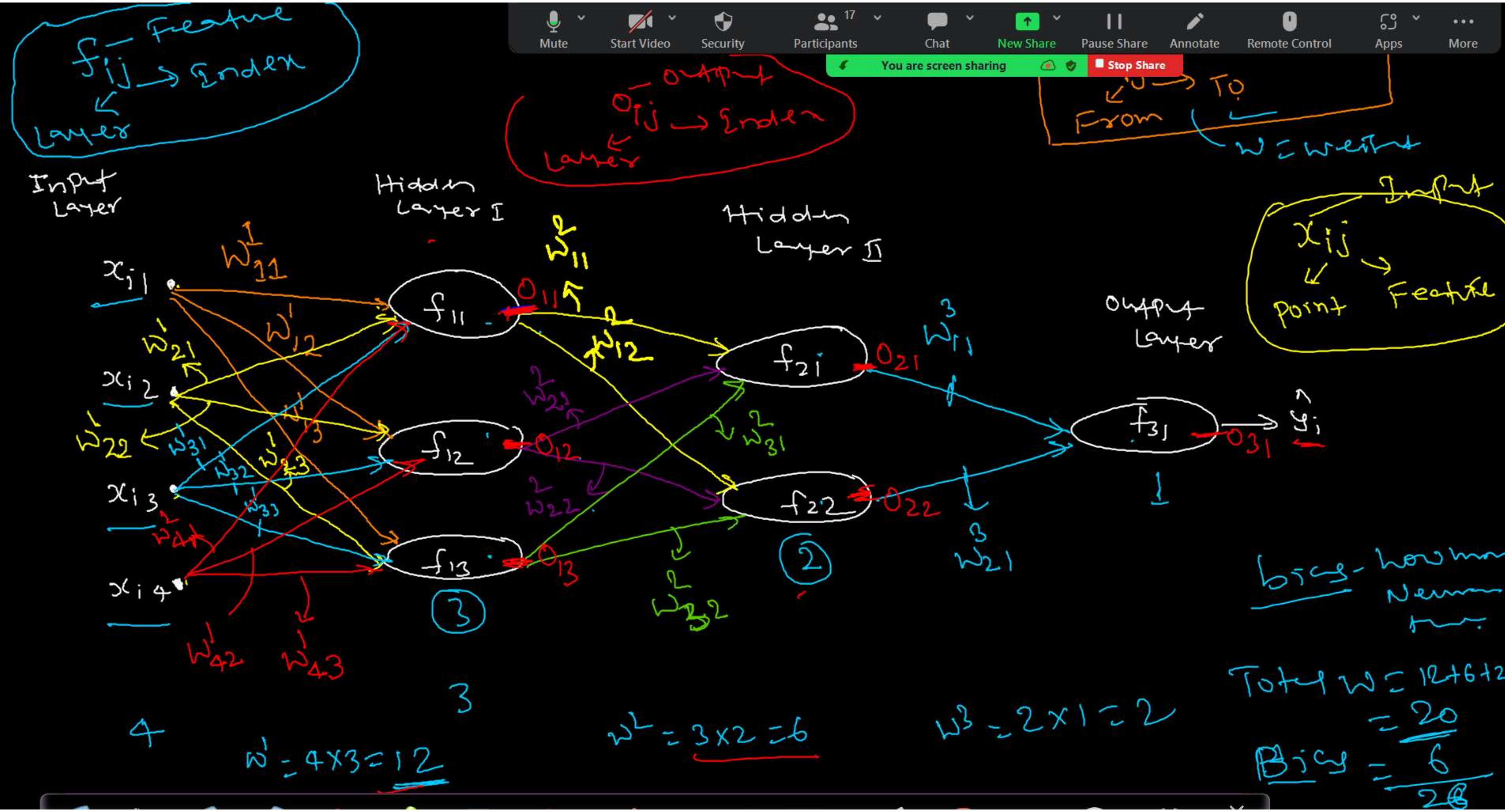
MLP - Graphical way of representation ($f \circ g$, $g \circ f$) - Function Compo

MLP is very powerful structure which can handle linear or non-linear or such by using DNN | MLP Connection

* Notation

$\mathbb{R}$ — Real Num

$D = \{x_i, y_i\}$, $x_i \in \mathbb{R}^4$, $y_i \in \mathbb{R}$ — Regression



Mute
Start Video
Security
Participants
Chat
New Share
Pause Share
Annotate
Remote Control
Apps
More
You are screen sharing
Stop Share
f — Feature
fij → Index
Layer
O — Output
Oij → Index
Layer
To
From
W = weight
Input
xij
point
Feature
Input Layer
Hidden Layer I
Hidden Layer II
Output Layer
xi1
xi2
xi3
xi4
f11
f12
f13
f21
f22
f31
O11
O12
O13
O21
O22
O31
yi
bias - how many Neurons
w' = 4x3 = 12
w² = 3x2 = 6
w³ = 2x1 = 2
Total w = 12+6+2
= 20
Bias = 6
26

$$W_1 = \begin{bmatrix} w^1_{11} & w^1_{12} & w^1_{13} \\ w^1_{21} & w^1_{22} & w^1_{23} \\ w^1_{31} & w^1_{32} & w^1_{33} \\ w^1_{41} & w^1_{42} & w^1_{43} \end{bmatrix}_{4 \times 3}$$

how weight calculate —

— weight

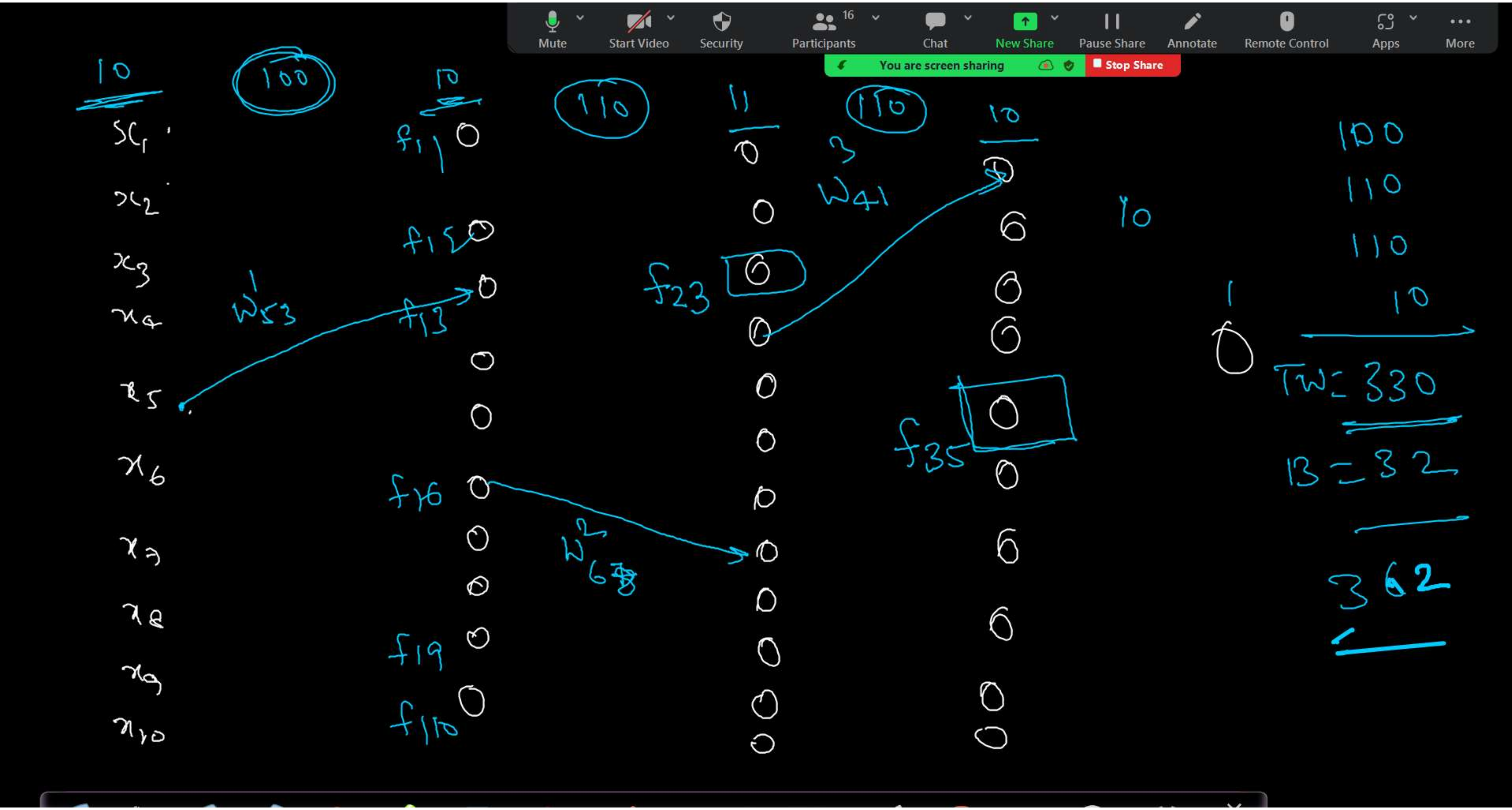
Copper

Xavier / Glorot

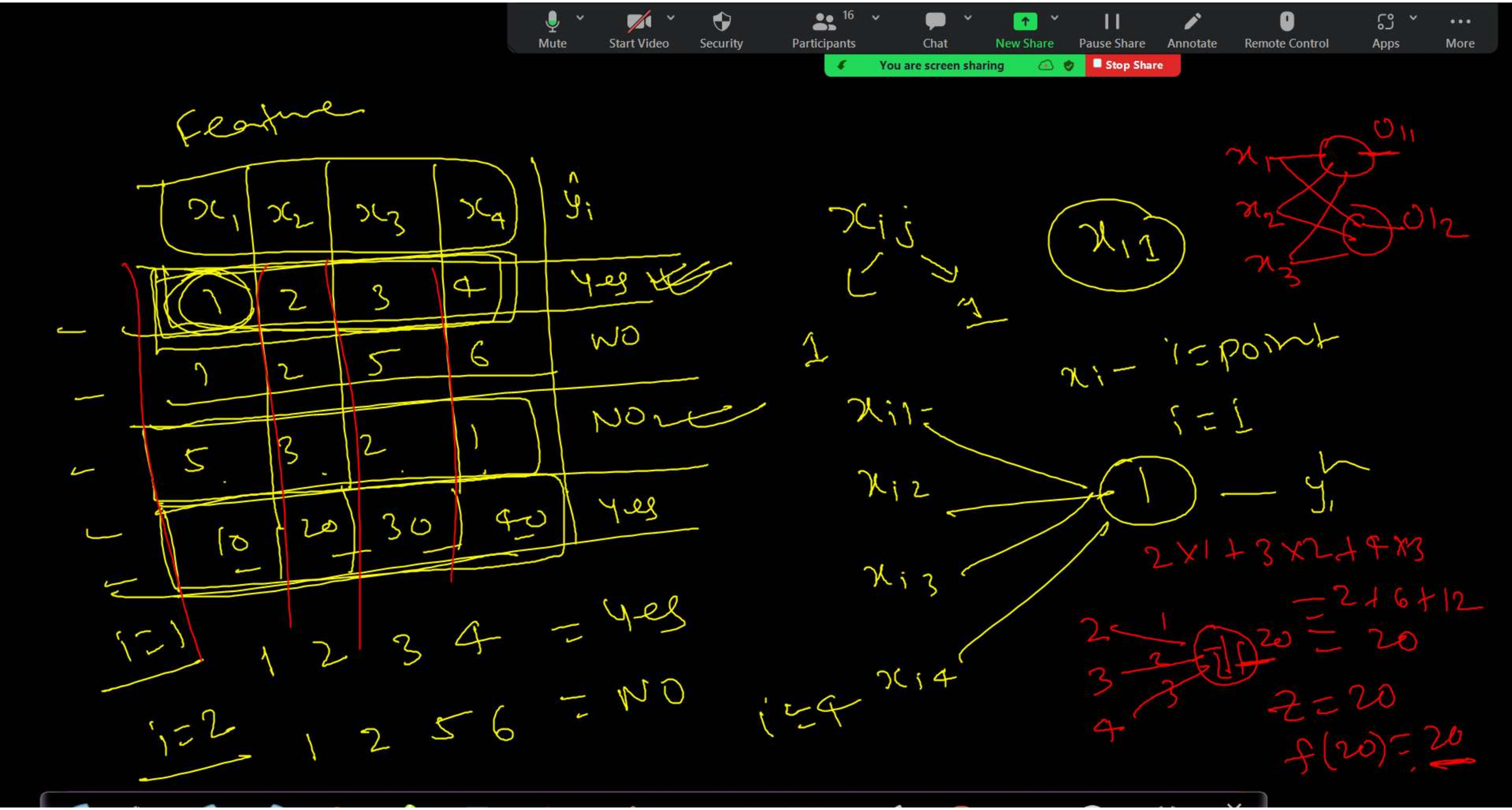
He ---

$$W_2 = \begin{bmatrix} w^2_{11} & w^2_{12} \\ w^2_{21} & w^2_{22} \\ w^2_{31} & w^2_{32} \end{bmatrix}_{3 \times 2}$$

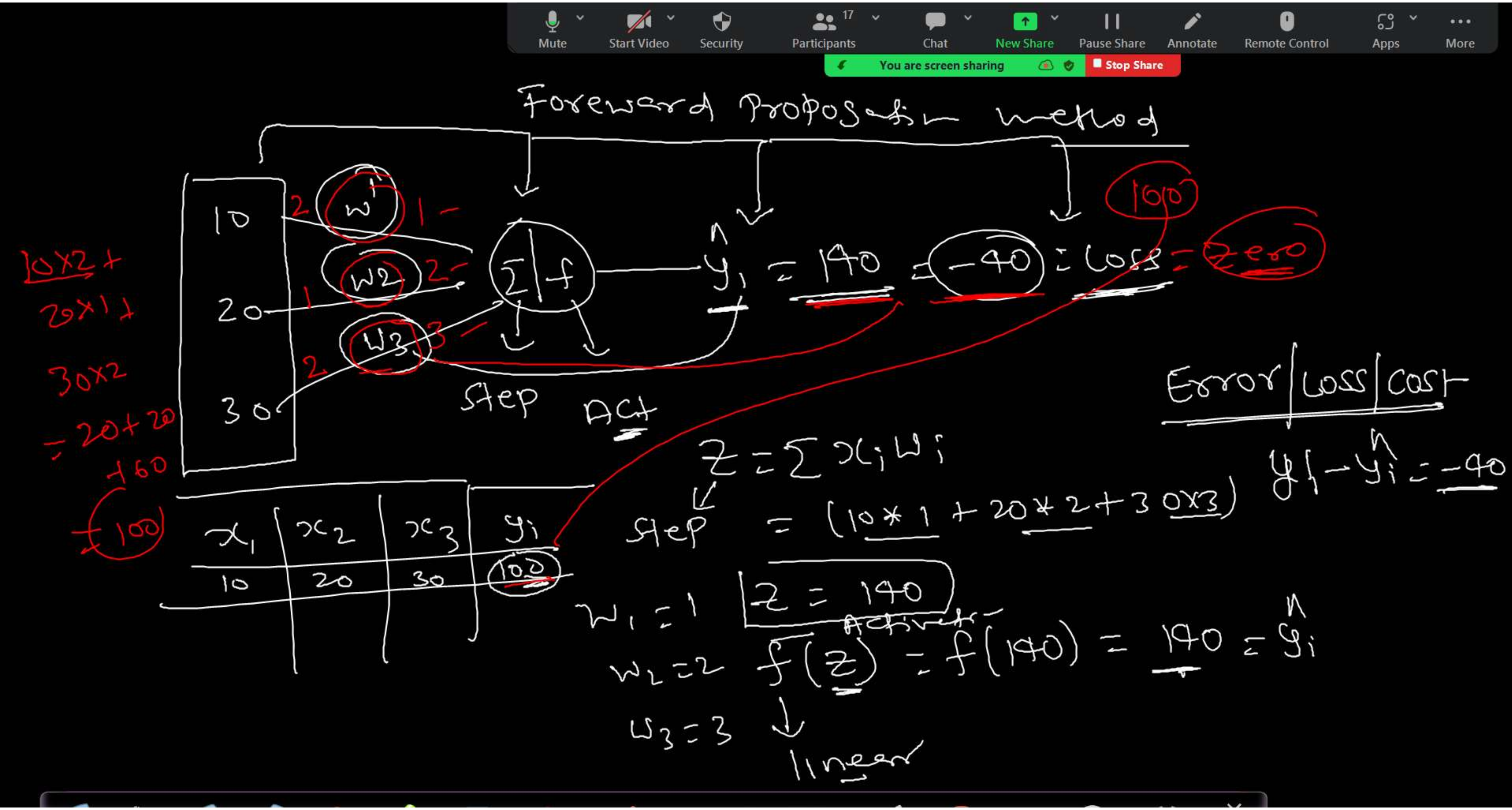
$$W_3 = \begin{bmatrix} w^3_{11} \\ w^3_{21} \end{bmatrix}_{2 \times 1}$$

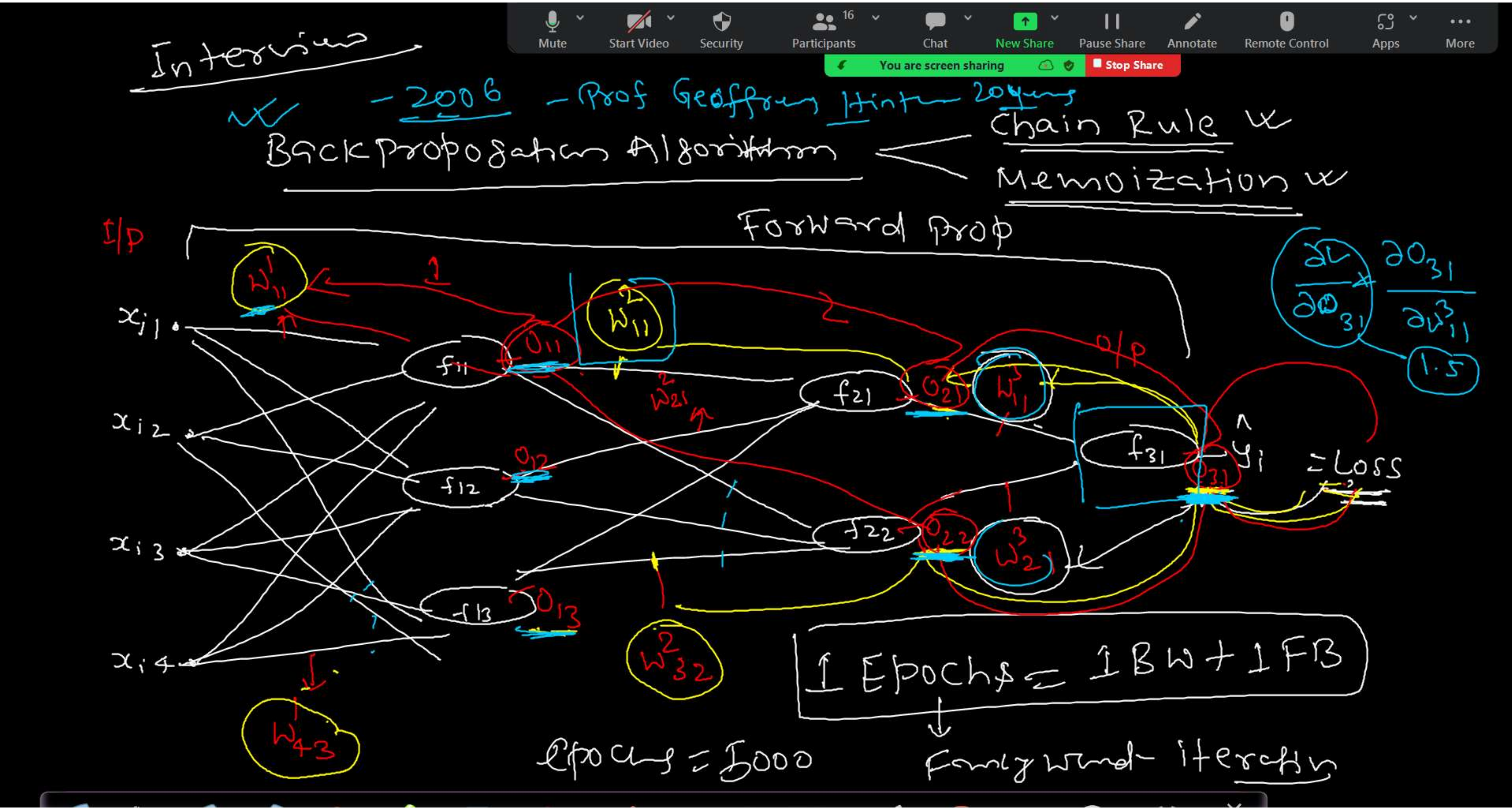


Mute
Start Video
Security
Participants
16
Chat
New Share
Pause Share
Annotate
Remote Control
Apps
More
You are screen sharing
Stop Share
10
100
10
110
11
110
10
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
f_{11}
f_{12}
f_{13}
f_{16}
f_{19}
f_{110}
f_{23}
f_{35}
W^1_{53}
W^2_{68}
W^3_{41}
10
100
110
110
10
TW = 330
B = 32
362



Mute
Start Video
Security
Participants
Chat
New Share
Pause Share
Annotate
Remote Control
Apps
More
You are screen sharing
Stop Share
Feature
x1 x2 x3 x4
ŷi
1 2 3 4 yes
1 2 5 6 NO
5 3 2 1 NO
10 20 30 40 yes
i=1 1 2 3 4 = yes
i=2 1 2 5 6 = NO
xij
x11
xi — i=point
i=1
xi1
xi2
xi3
xi4
i=4
yi
2×1+3×2+4×3
=2+6+12
=20
z=20
f(20)=20
x1 x2 x3
O11 O12


Mute
Start Video
Security
Participants 17
Chat
New Share
Pause Share
Annotate
Remote Control
Apps
More
You are screen sharing
Stop Share
Foreward Proposation method
10
20
30
w1
w2
w3
Σ | f
ŷ1 = 140 = -40 = Loss = zero
100
Step
Act
Error | Loss | cost
10×2 + 20×1 + 30×2 = 20 + 20 + 60 = 100
x1 | x2 | x3 | yi
10 | 20 | 30 | 100
z = Σ xi wi
step = (10 * 1 + 20 * 2 + 30 × 3)
yi - ŷi = -40
w1 = 1
z = 140
Activate
w2 = 2
f(z) = f(140) = 140 = ŷi
w3 = 3
linear



Mute
Start Video
Security
Participants 16
Chat
New Share
Pause Share
Annotate
Remote Control
Apps
More
You are screen sharing
Stop Share
Interview
2006 - Prof Geoffrey Hinton
Backpropogation Algorithm
Chain Rule
Memoization
Forward Prop
I/P
O/P
x_{i1}
x_{i2}
x_{i3}
x_{i4}
f_{11}
f_{12}
f_{13}
f_{21}
f_{22}
f_{31}
O_{11}
O_{12}
O_{13}
O_{21}
O_{22}
O_{31}
W^1_{11}
W^2_{11}
W^2_{21}
W^3_{11}
W^3_{21}
W^2_{32}
W^1_{43}
$\hat{y}_i$
= Loss
$\frac{\partial L}{\partial O_{31}} \times \frac{\partial O_{31}}{\partial w^3_{11}}$
1.5
1 Epochs = 1 BW + 1 FB
Epochs = 5000
iteration

Chain Rule

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial v} \cdot \frac{\partial v}{\partial x}$$

$$\frac{\partial L}{\partial w^3_{11}} = \boxed{\frac{\partial L}{\partial O_{31}}} * \frac{\partial O_{31}}{\partial w^3_{11}}$$

$$\frac{\partial L}{\partial w^3_{21}} = \boxed{\frac{\partial L}{\partial O_{31}}} * \frac{\partial O_{31}}{\partial w^3_{21}}$$

$$\frac{\partial L}{\partial w^2_{11}} = \boxed{\frac{\partial L}{\partial O_{31}}} * \frac{\partial O_{31}}{\partial O_{21}} * \frac{\partial O_{21}}{\partial w^2_{11}}$$

$$\frac{\partial L}{\partial w^2_{32}} = \boxed{\frac{\partial L}{\partial O_{31}}} * \frac{\partial O_{31}}{\partial O_{22}} * \frac{\partial O_{22}}{\partial w^2_{32}}$$

$$\frac{\partial L}{\partial w^1_{11}} = \frac{\partial L}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{21}} * \frac{\partial O_{21}}{\partial O_{11}} * \frac{\partial O_{11}}{\partial w^1_{11}} + \frac{\partial L}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{22}} * \frac{\partial O_{22}}{\partial O_{11}} * \frac{\partial O_{11}}{\partial w^1_{11}}$$

$$\frac{\partial L}{\partial w^1_{11}} = \boxed{\frac{\partial L}{\partial O_{31}}} * \frac{\partial O_{11}}{\partial w^1_{11}} \left\{ \frac{\partial O_{31}}{\partial O_{21}} * \frac{\partial O_{21}}{\partial O_{11}} + \frac{\partial O_{31}}{\partial O_{22}} * \frac{\partial O_{22}}{\partial O_{11}} \right\}$$

$f(x)$

$x \qquad\qquad h($

$g(x)$

$$\frac{\partial L}{\partial x} = \frac{\partial h}{\partial f} * \frac{\partial f}{\partial x} + \frac{\partial h}{\partial g} * \frac{\partial g}{\partial x}$$

Mute Start Video Security Participants 16 Chat New Share Pause Share Annotate Remote Control Apps More

You are screen sharing Stop Share

Memoization : it is not memorization, it is called

Memoization → Dynamic Programming

computer science

$$\frac{\partial L}{\partial w^3_{11}} = \boxed{\frac{\partial L}{\partial O_{31}}} \times \frac{\partial O_{31}}{\partial w^3_{11}}$$

$$\frac{\partial L}{\partial w^3_{21}} = \boxed{\frac{\partial L}{\partial O_{31}}} \times \frac{\partial O_{31}}{\partial w^3_{21}}$$

Compute once & reuse it.

* → The core idea of memoization is this

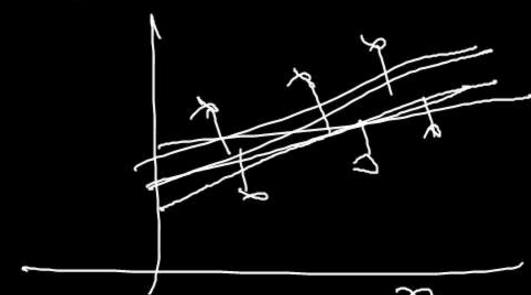
→ if there is any operation that is used many times, it's good idea to compute once & then store it for reuse purpose

→ This concept is known as Memoization

① Define Loss Function (example - Regression)

$Y = m_1 x_1 + m_2 x_2 + c$

$$\boxed{L = \underbrace{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}_{\text{Sq-Loss fn:}} + \text{regularization}}$$



$$\text{Error} = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

② Optimization

$$\boxed{\text{Loss (minimize)}}$$

$\downarrow$

Weight & bias

Mute Start Video Security Participants 17 Chat New Share Pause Share Annotate Remote Control Apps More

You are screen sharing Stop Share

③ Gradient Descent (Batch GD, Mini-batch GD, SGD)

(i) initialize w_{ij}^{k} randomly — lots of technique → we will cover later

(ii) Update weight — Rule :-

Formula:- $$\left(w_{ij}^{k}\right)_{new} = \left(w_{ij}^{k}\right)_{old} - \eta * \frac{\partial L}{\partial w_{old}}$$ ← V.V.Imp. BackProp.

η → eta / learning Rate (0 to 1.0)

$w_{old} = 50$

$w_{new} = 50 - 1 * 5$

$w_{new} = 45$

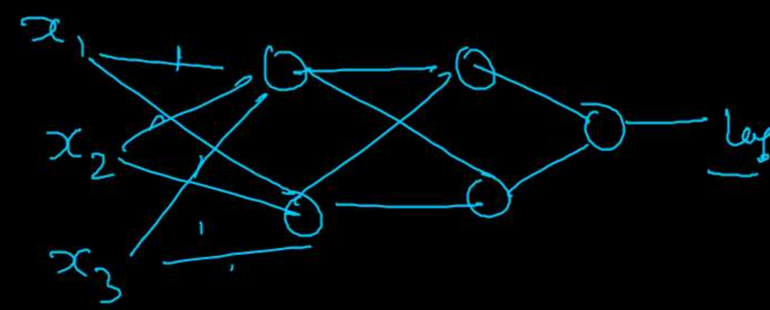
(ii) Perform updates till the convergence.

NOTE :- convergence means the new weight value and the old weight value are very close to each other.

$$\left\{\begin{array}{l} W_{old} = 50 \\ W_{new} = 49.9995 \\ W_{new_2} = 49.9998 \\ W_{new_3} = 49.9997 \end{array}\right.$$

2.45

Back-Prop



1. Initialize weight - randomly
2. epochs (1BP + 1FB) → Forward Prop
   → compute loss
   → compute derivatives (Chain Rule + memoization)
   → update weight in a backward prop
3. repeat step 2 till the convergence.