# History of Neural Networks

Deep learning

→ Perceptron → 1957 [Rosenblatt]

→ logistic regression

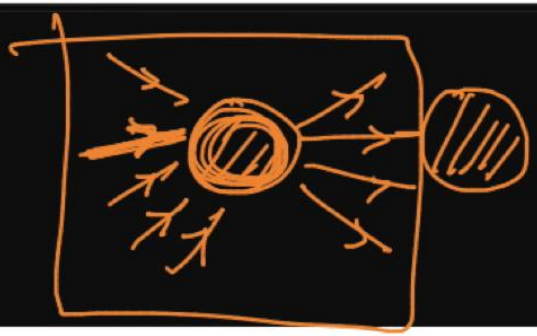
Russian – English

→ Alan Turing [Computer]

→ "loosely" inspired from biology

Contents - Google Docs | 0*v4f4-nMoRMNrtUZG.png | Chekuri Srikan...

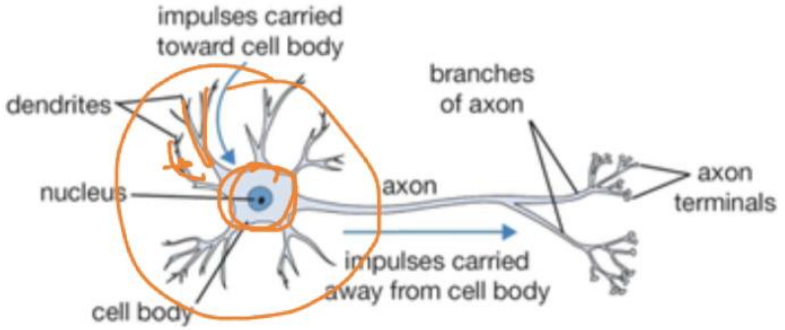
Secure | https://cdn-images-1.medium.com/max/1600/0*v4f4-nMoRMNrtUZG.png

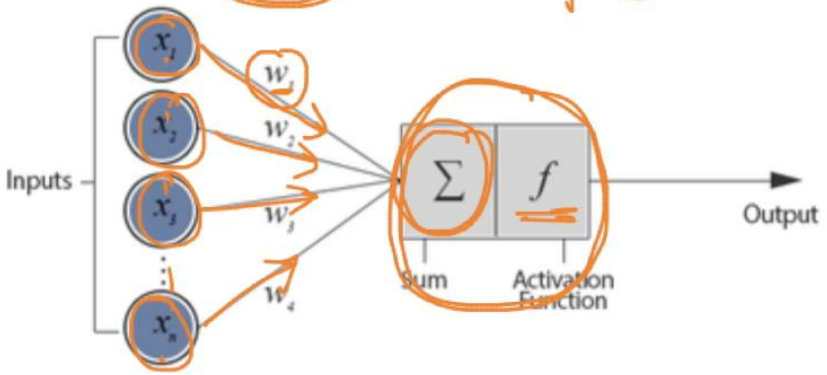


most-simple

## Biological Neuron versus Artificial Neural Network

$f(w_1x_1+w_2x_2+\cdots+w_nx_n)$

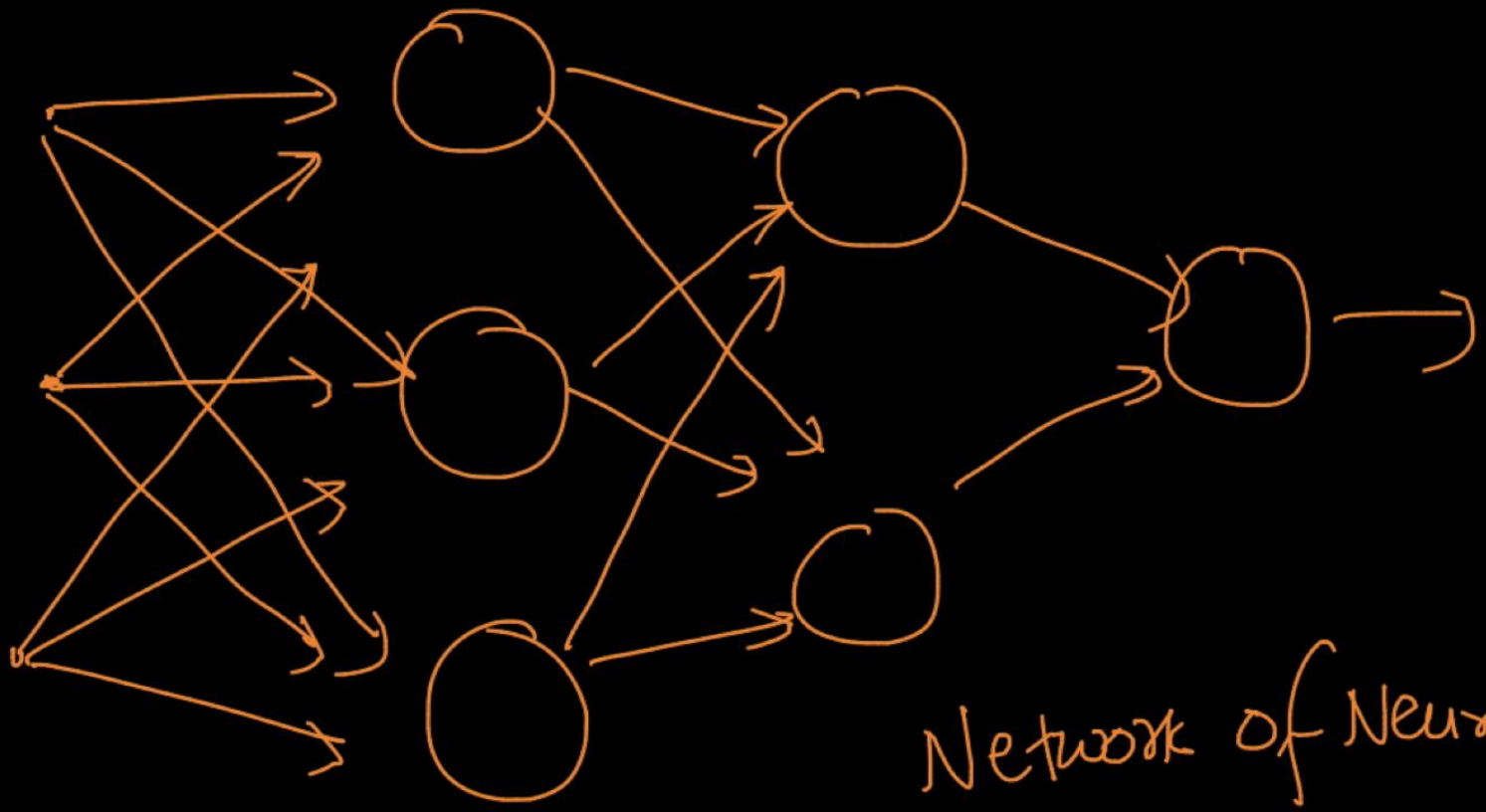


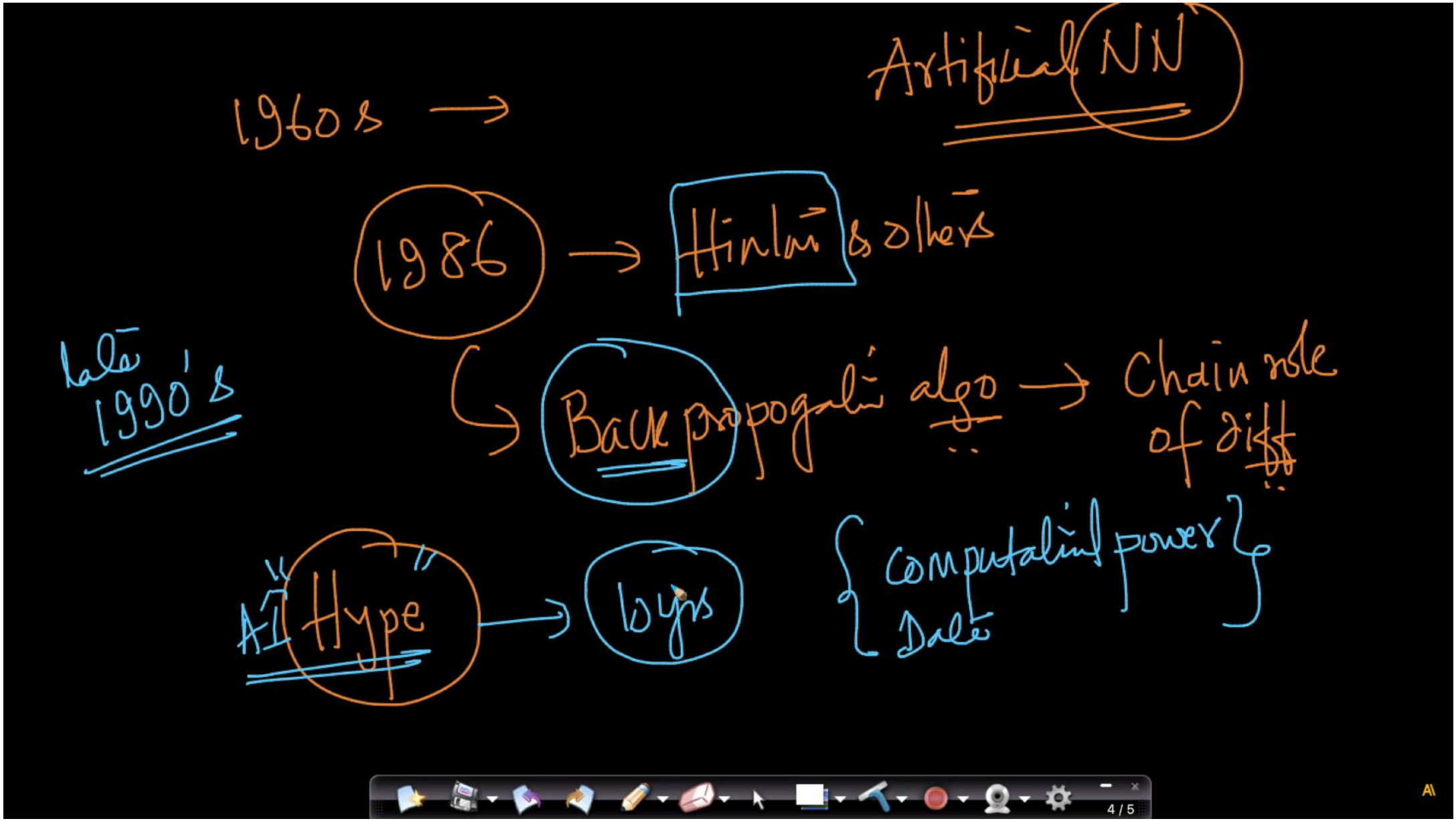

2 / 2

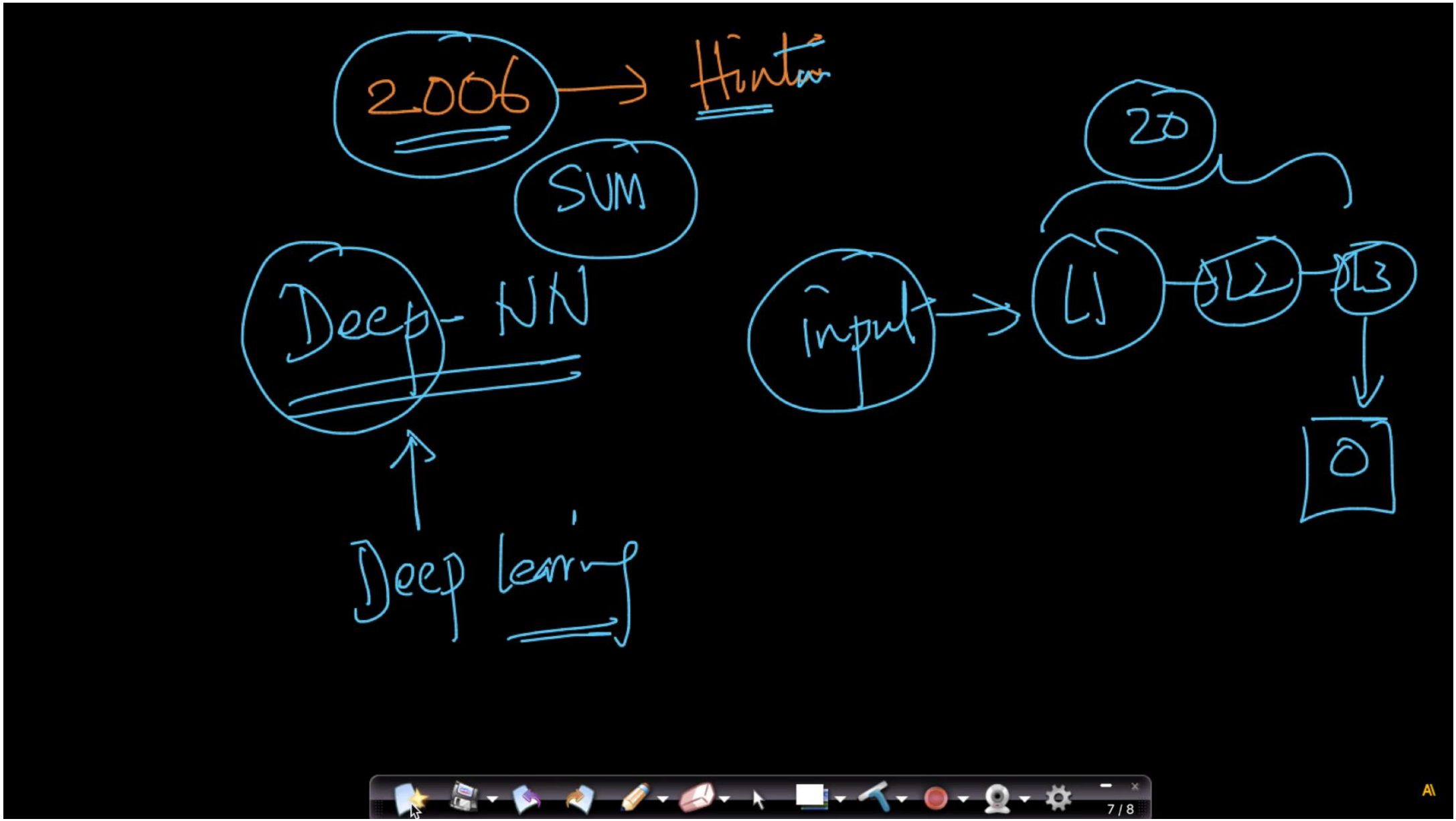
APPLIED COURSE AI

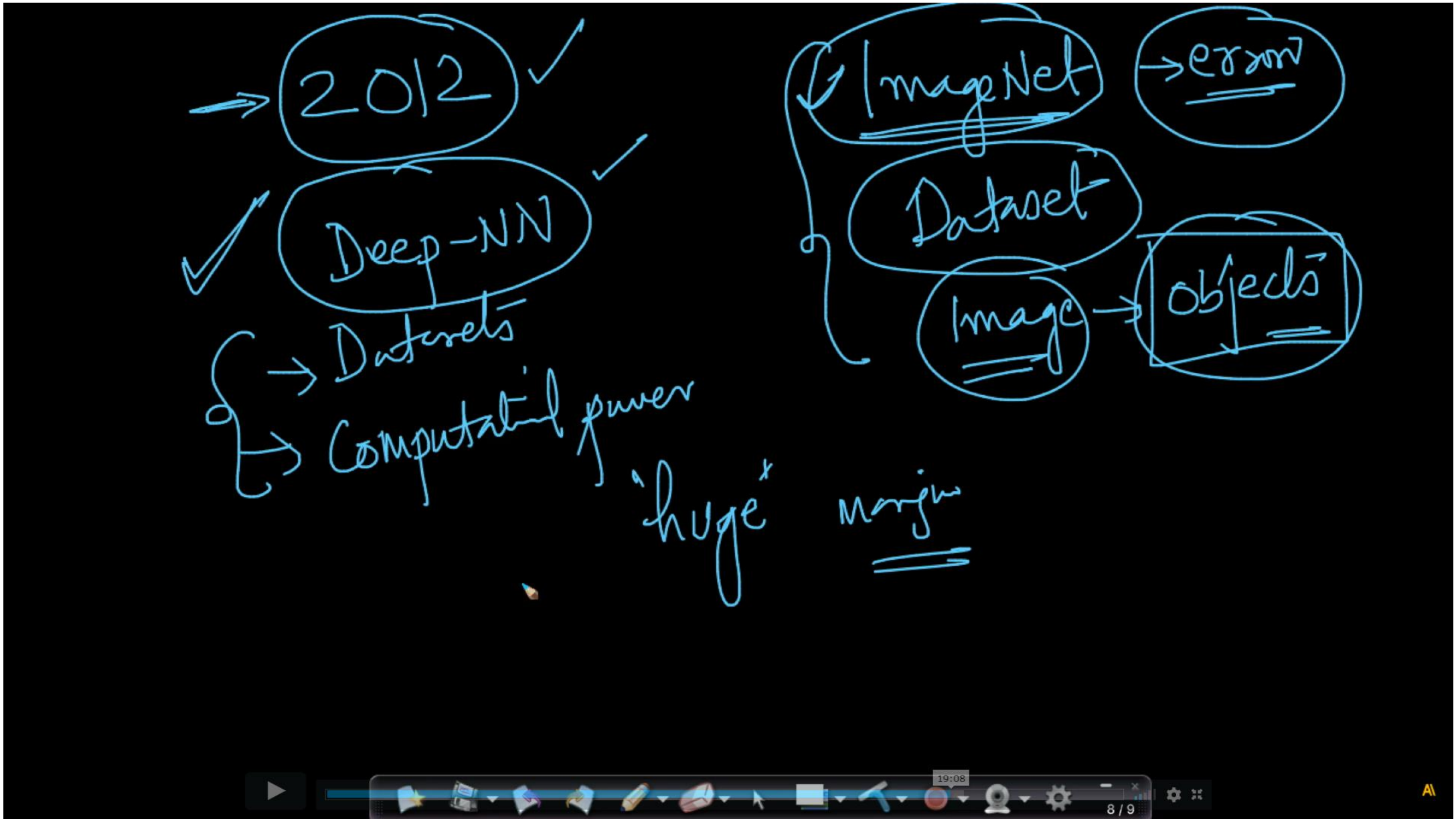
Brain

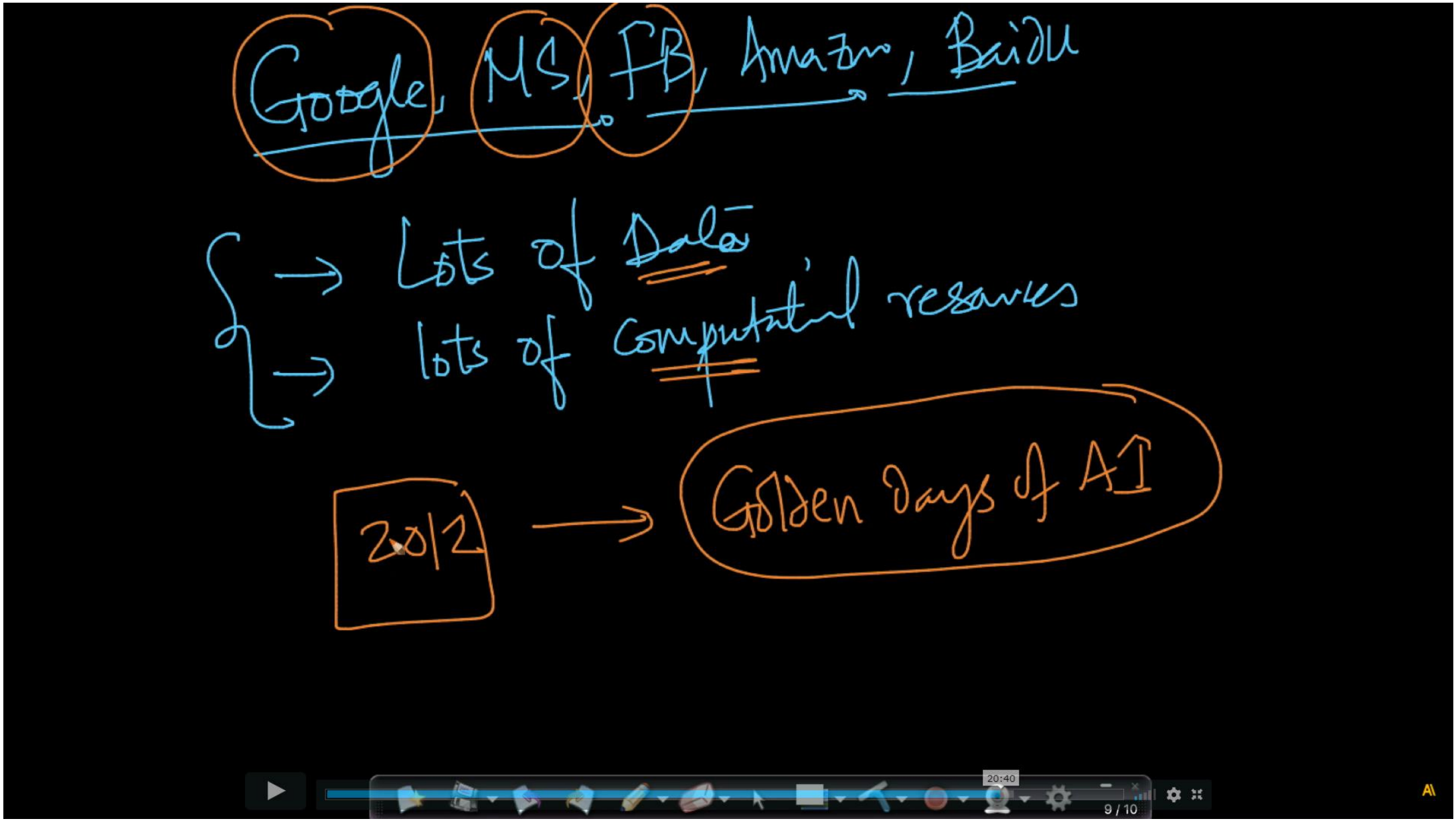


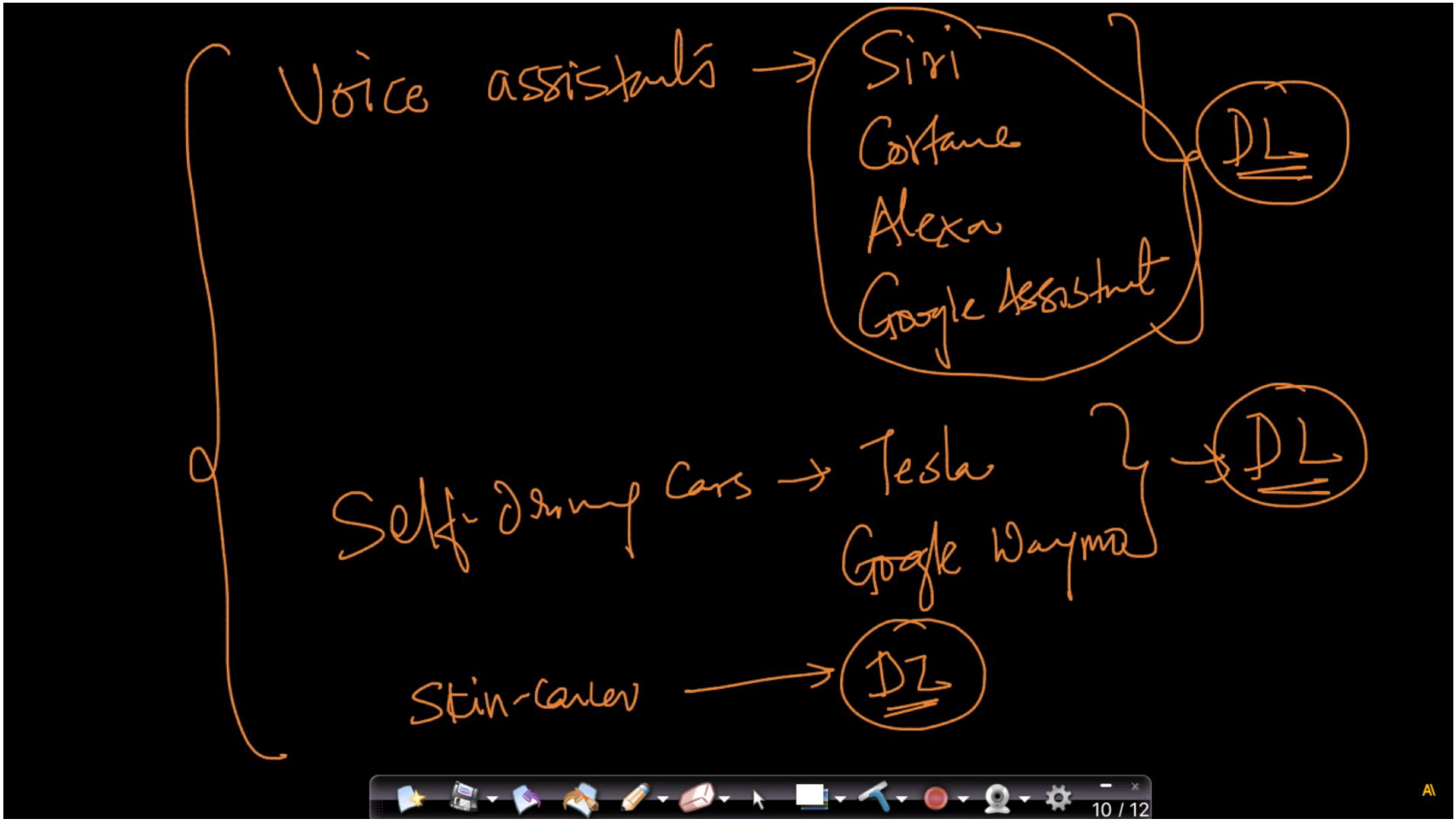
Network of Neurons

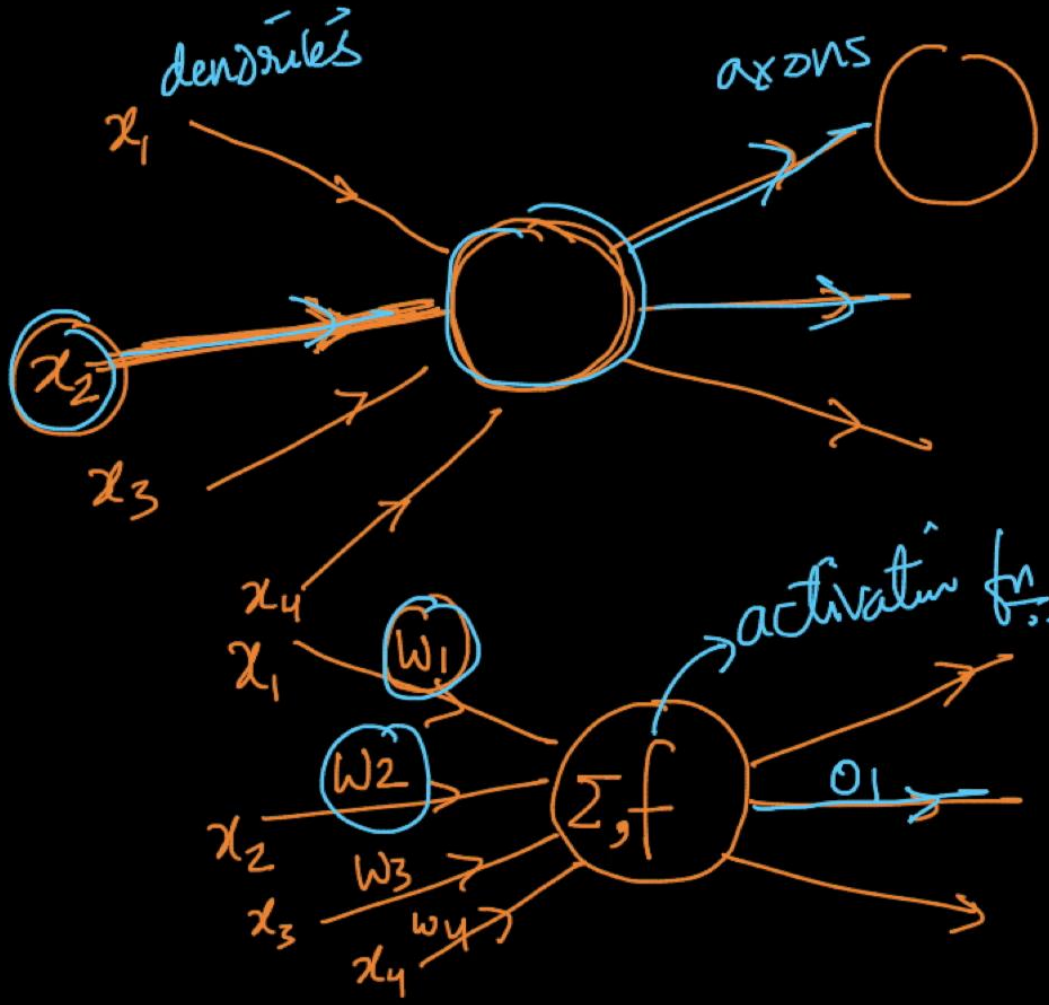

1960s →
Artificial NN
1986 → Hinton & others
late 1990's
↳ Back propogation algo → Chain rule of diff
AI Hype → boys
{ computational power
Data }


2006
Hinton
SUM
Deep - NN
Deep learning
input
L1
L2
L3
20
O
7 / 8


2012
Deep-NN
Datasets
Computational power
"huge" Margin
ImageNet
error
Dataset
Image
objects
19:08
8 / 9


Google, MS, FB, Amazon, Baidu
→ Lots of Data
→ lots of Computational resources
2012
Golden Days of AI
20:40
9 / 10


Voice assistants
Siri
Cortana
Alexa
Google Assistant
DL
Self-driving Cars
Tesla
Google Waymo
DL
Stin-Carlev
DL
10 / 12

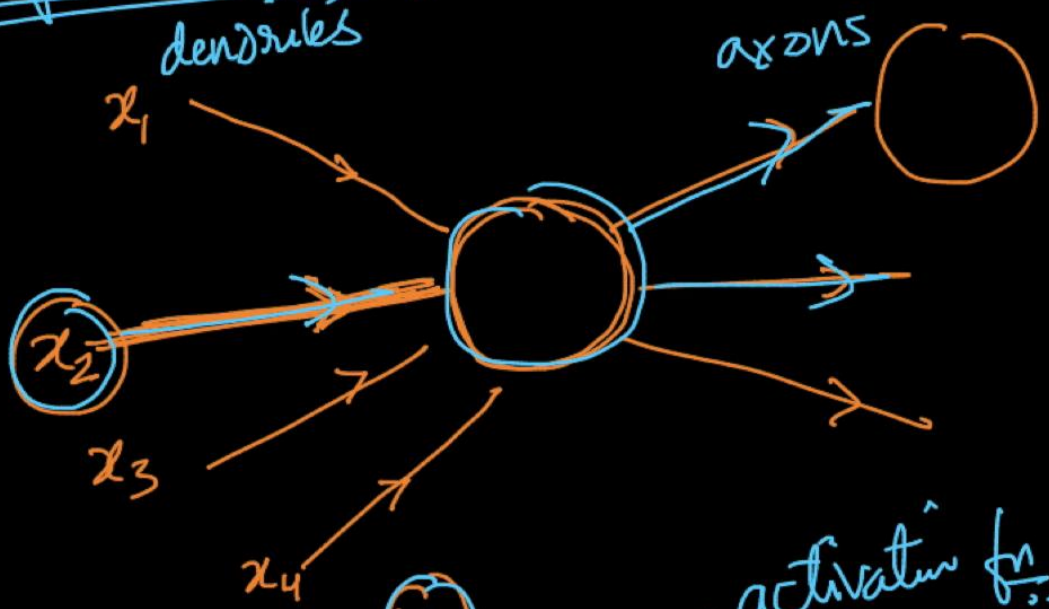



{ simplified view of a Neuron }

{ activated fired }

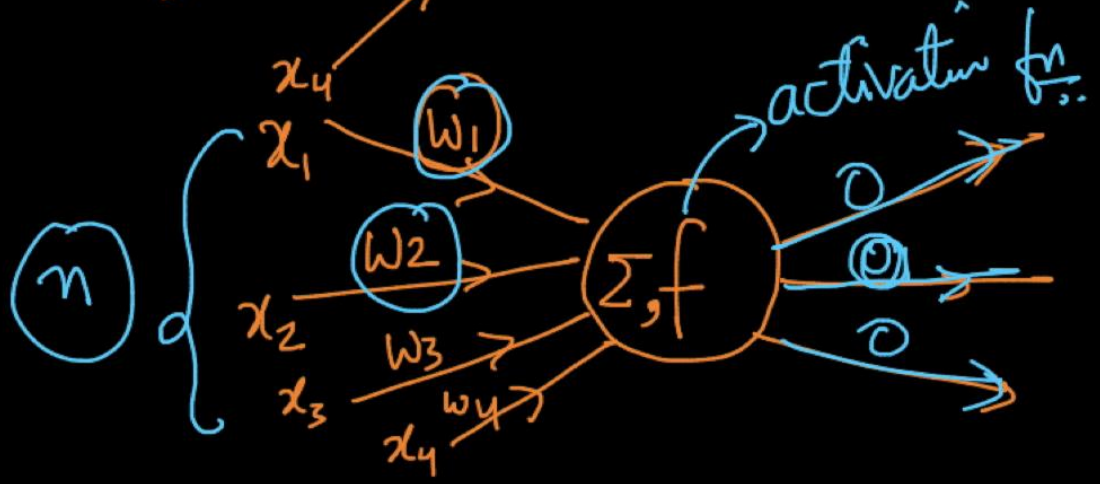
$$o_1 = f(w_1x_1 + w_2x_2 + \cdots + w_nx_n)$$

Outgoing vertices / edges / lines → axons

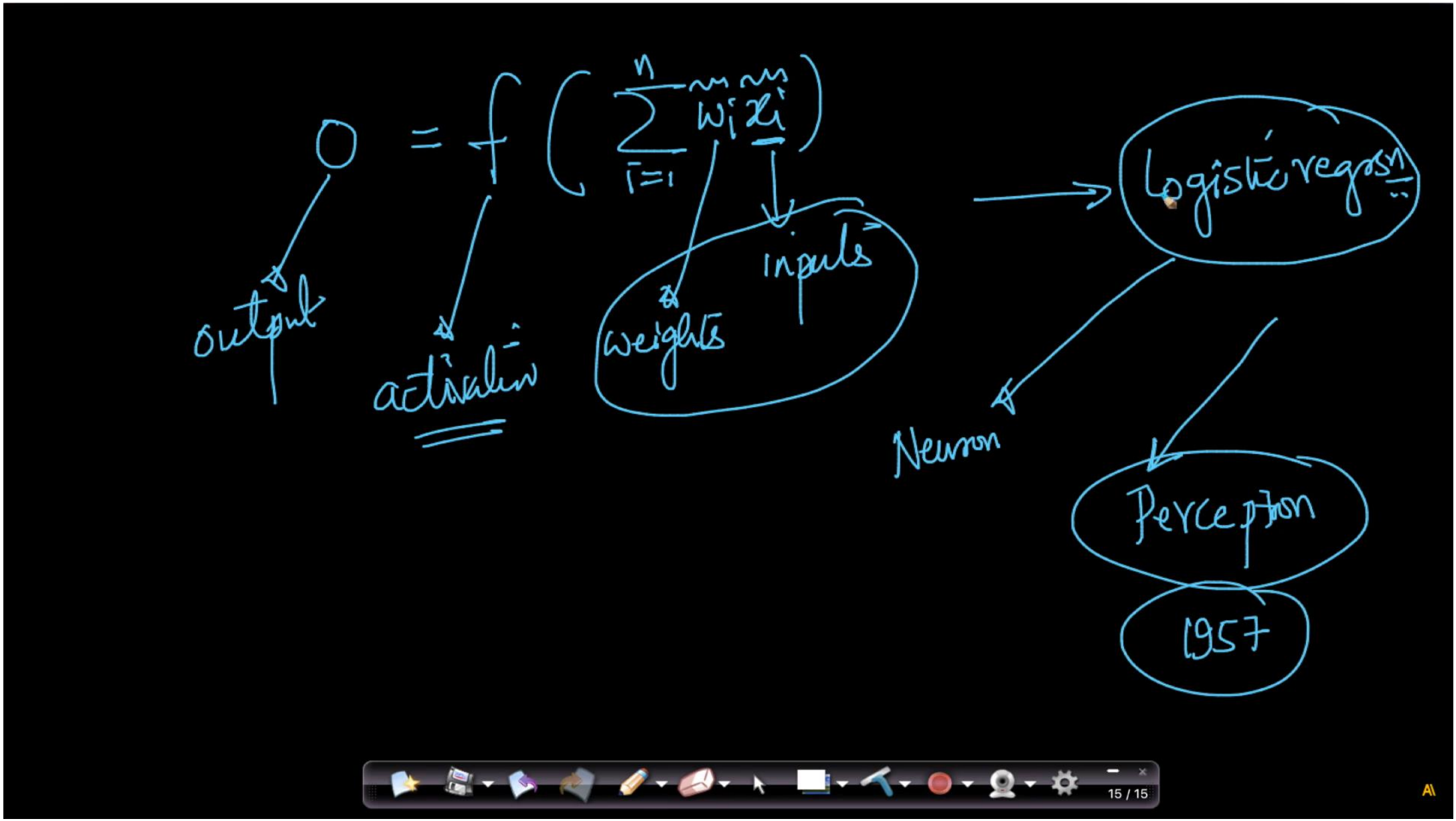



{ simplified view of a Neuron }

{ activated / fired }




$$a = f(w_1x_1 + w_2x_2 + \cdots + w_nx_n)$$


$o = f\left(\sum_{i=1}^{n} w_i x_i\right)$
output
activation
weights
inputs
Neuron
logistic regression
Perceptron
1957

biological nueral - Google Sear × | Five Numbers to Remember ab × | Chekuri Srikan...

Secure | https://developingchild.harvard.edu/resources/five-numbers-to-remember-about-early-childhood-development/

## 1 More Than 1 Million New Neural Connections Per Second*

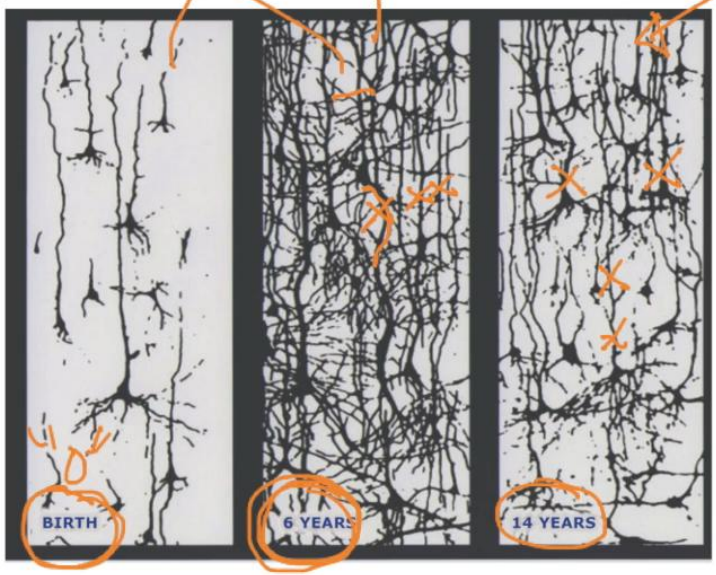



Image source: Conel, JL. The postnatal development of the human cerebral cortex. Cambridge, Mass: Harvard University Press, 1959.

The early years matter because, in the first few years of life, more than 1 million new neural connections are formed every second.* Neural connections are formed through the interaction of genes and a baby's environment and experiences, especially "serve and return" interaction with adults, or what developmental researchers call contingent reciprocity. These are the connections that build brain architecture – the

1 More Than 1 Million New Neural Connections Per Second*
2 18 Months: Age At Which Disparities in Vocabulary Begin to Appear
3 90 - 100% Chance of Developmental Delays When Children Experience 6 - 7 Risk Factors
4 3:1 Odds of Adult Heart Disease After 7 - 8 Adverse Childhood Experiences
5 $4 - $9 in Returns For Every Dollar Invested in Early Childhood Programs

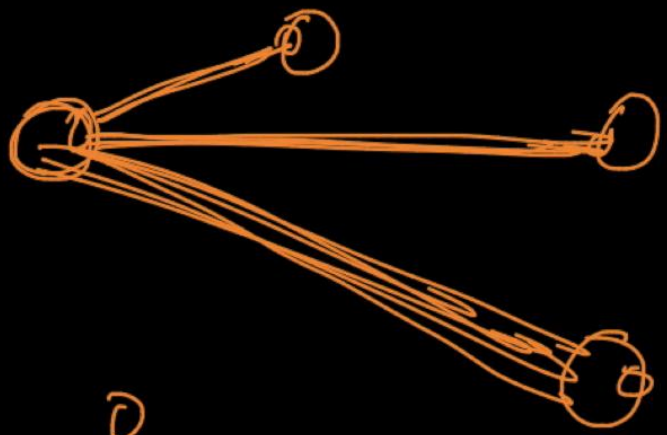
Join our mailing list to receive news and other updates from the Center.

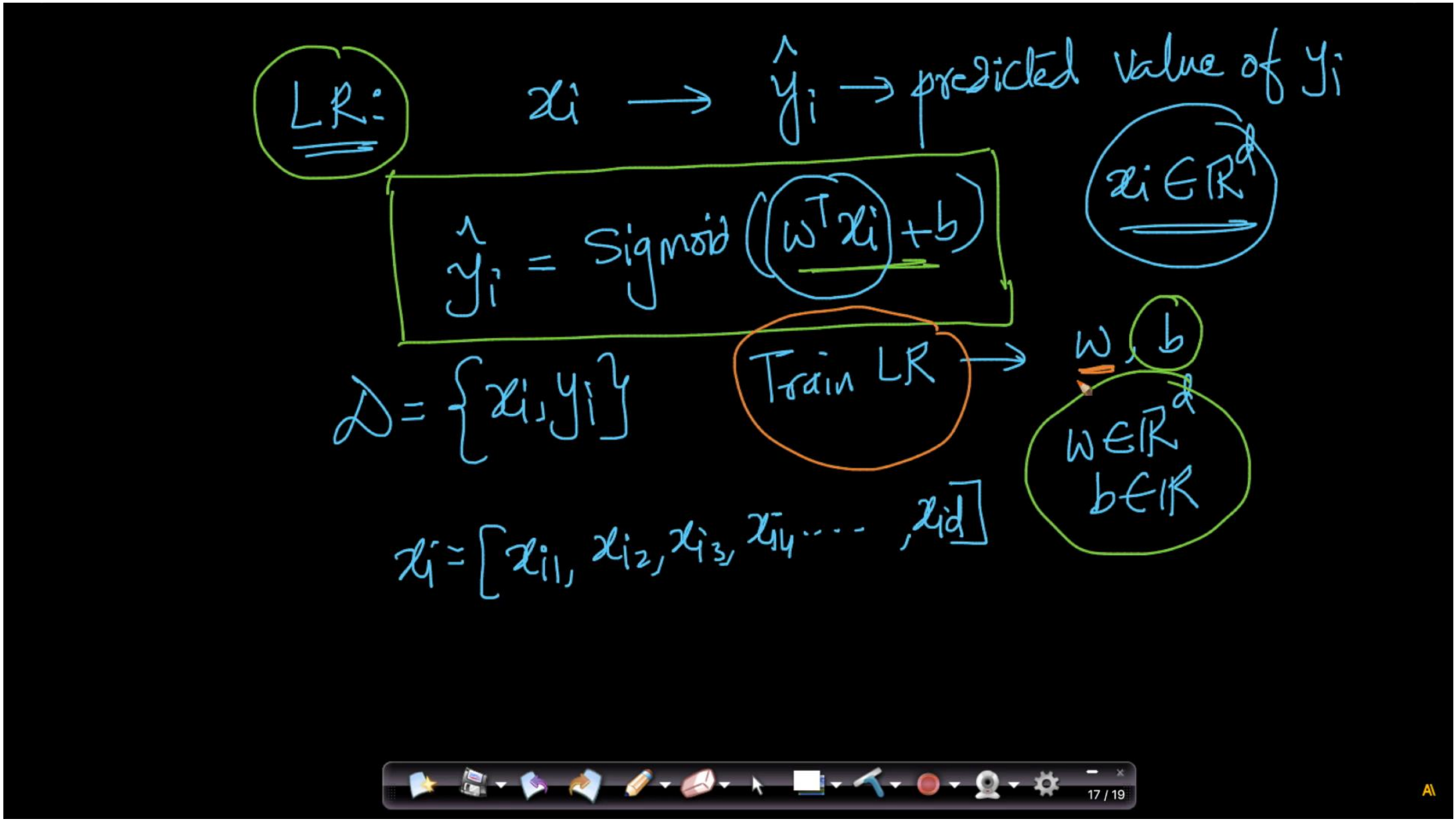
Your email address

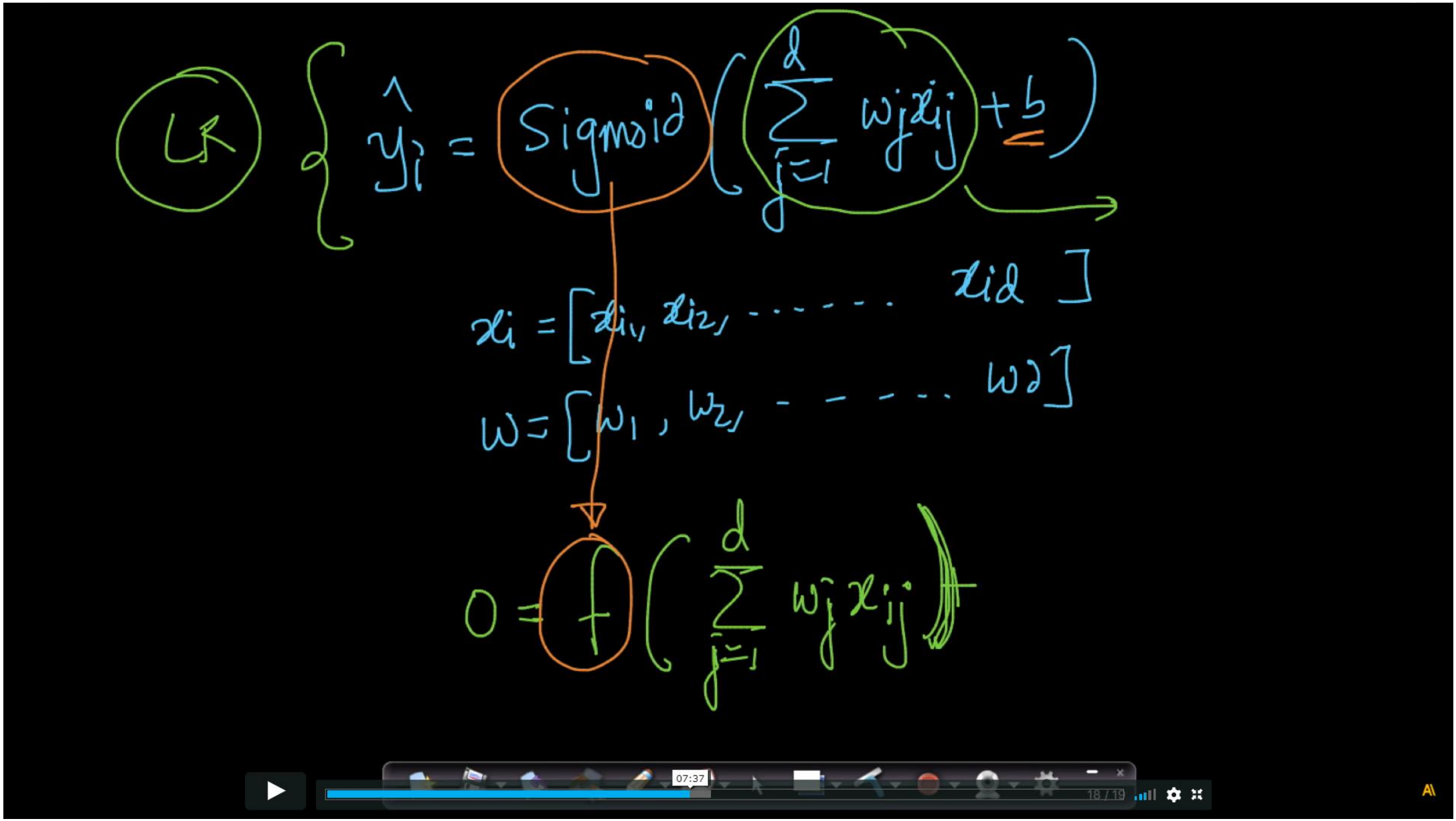
Subscribe

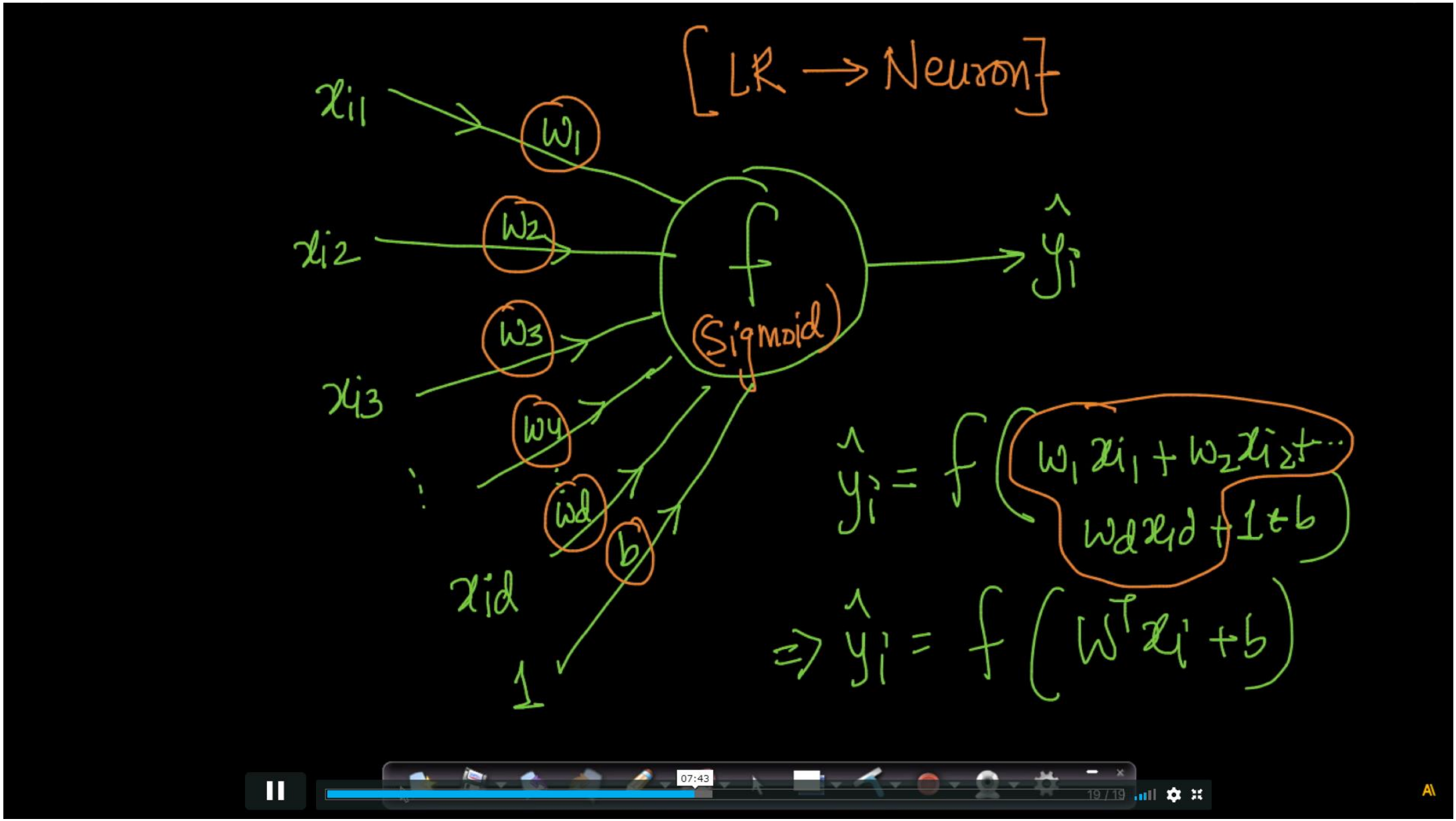
13 / 13

all biological being ⇒ (weights) as neural connection




LR:
$x_i \longrightarrow \hat{y}_i \rightarrow$ predicted value of $y_i$
$\hat{y}_i = \text{Sigmoid}(w^T x_i + b)$
$x_i \in \mathbb{R}^d$
$\mathcal{D} = \{x_i, y_i\}$
Train LR
$w, b$
$w \in \mathbb{R}^d$
$b \in \mathbb{R}$
$x_i = [x_{i1}, x_{i2}, x_{i3}, x_{i4} \cdots, x_{id}]$
17 / 19

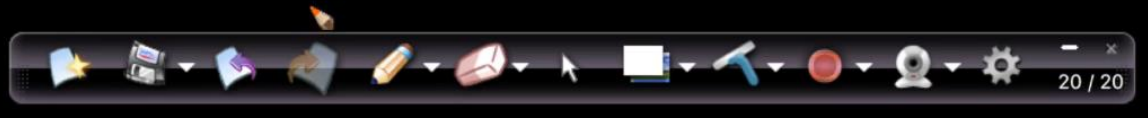

LR
$\hat{y}_i = \text{Sigmoid}\left(\sum_{j=1}^{d} w_j x_{ij} + b\right)$
$x_i = [x_{i1}, x_{i2}, \ldots\ldots x_{id}]$
$w = [w_1, w_2, \ldots\ldots w_d]$
$o = f\left(\sum_{j=1}^{d} w_j x_{ij}\right)$
07:37
18 / 19

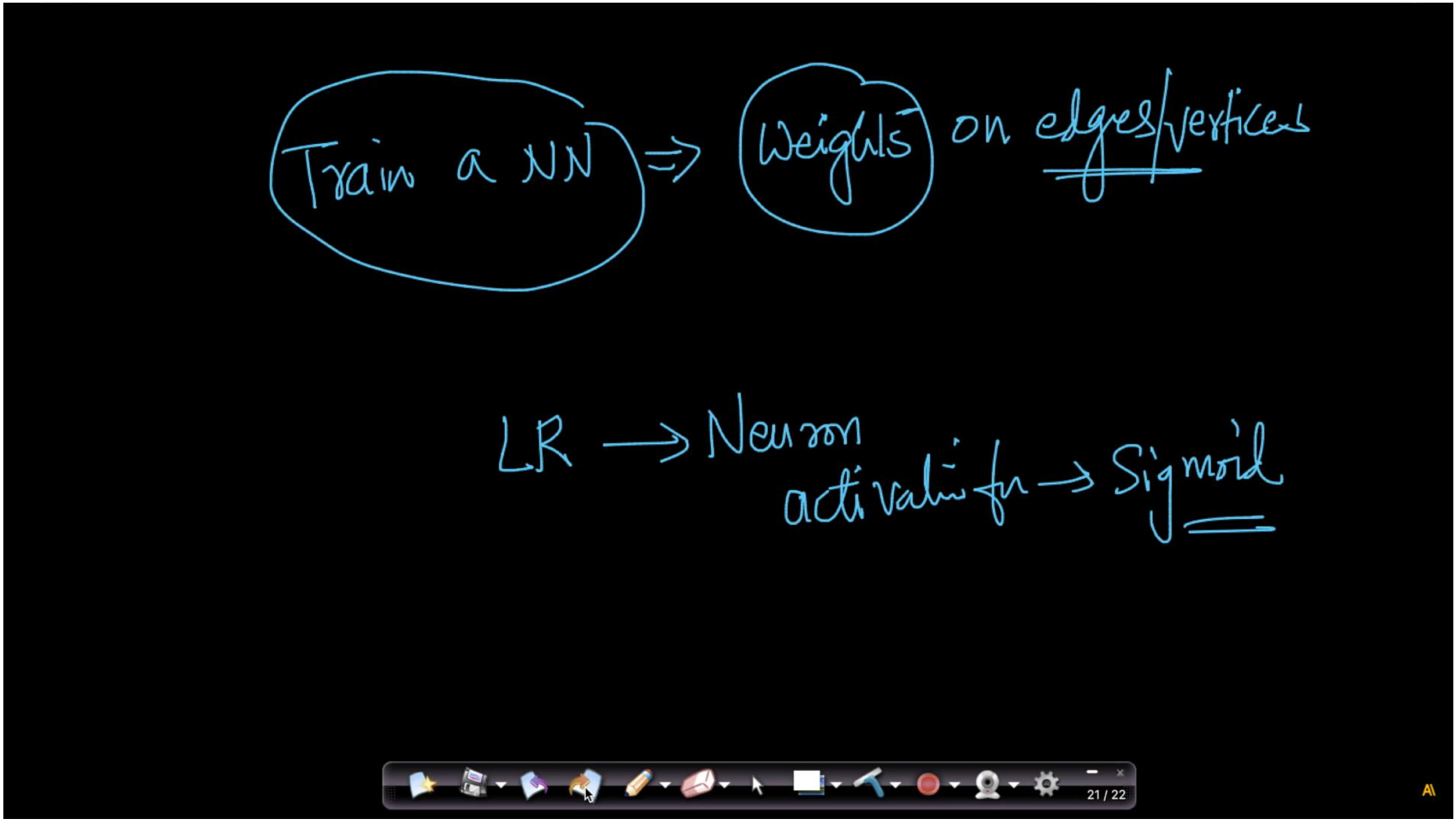

[LR → Neuron]
$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{id}$
1
$w_1$
$w_2$
$w_3$
$w_4$
$w_d$
$b$
$f$
(Sigmoid)
$\hat{y}_i$
$\hat{y}_i = f(w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id} + 1 * b)$
$\Rightarrow \hat{y}_i = f(w^T x_i + b)$
07:43
19 / 19

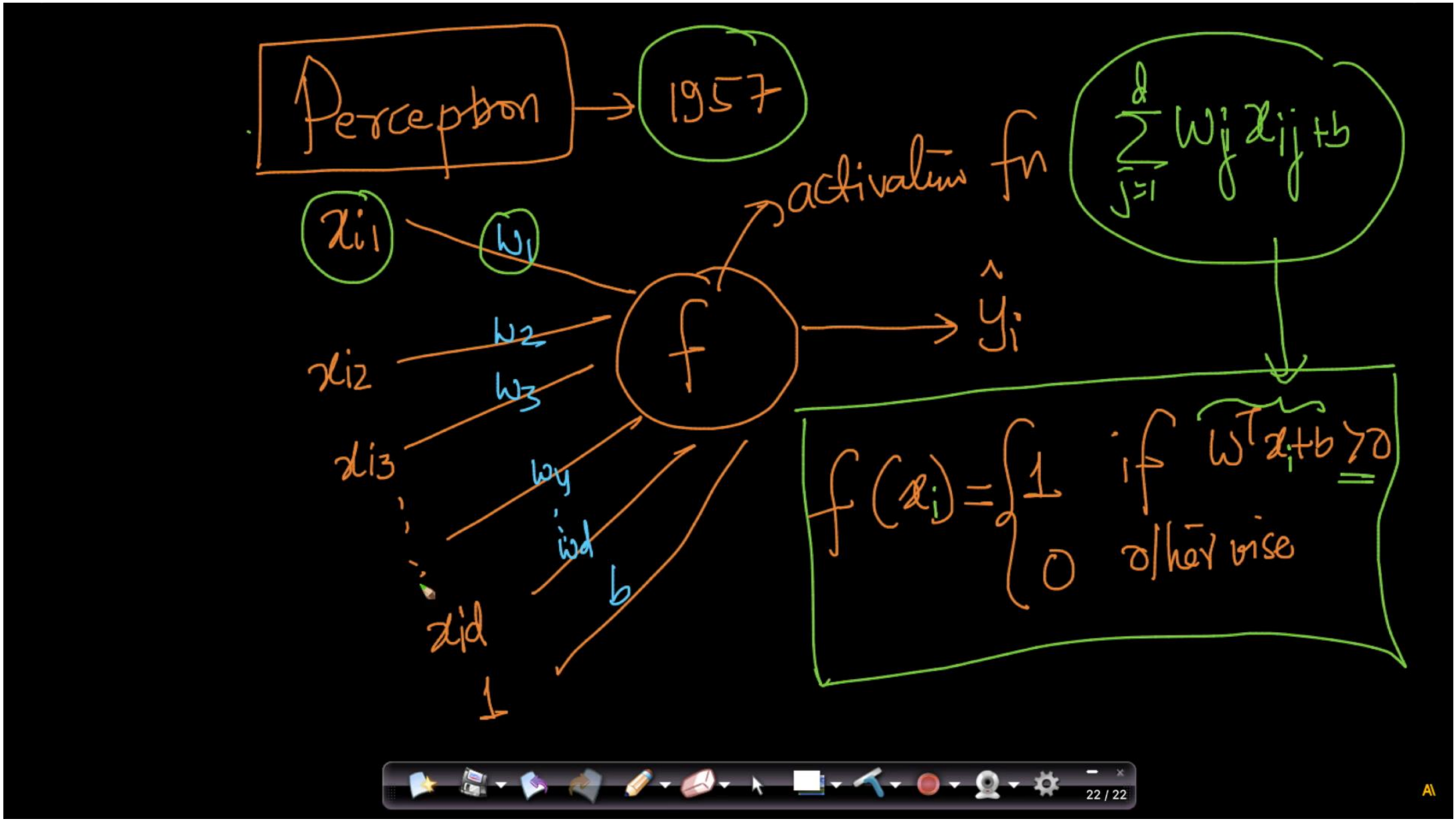
$D = \{x_i, y_i\}$

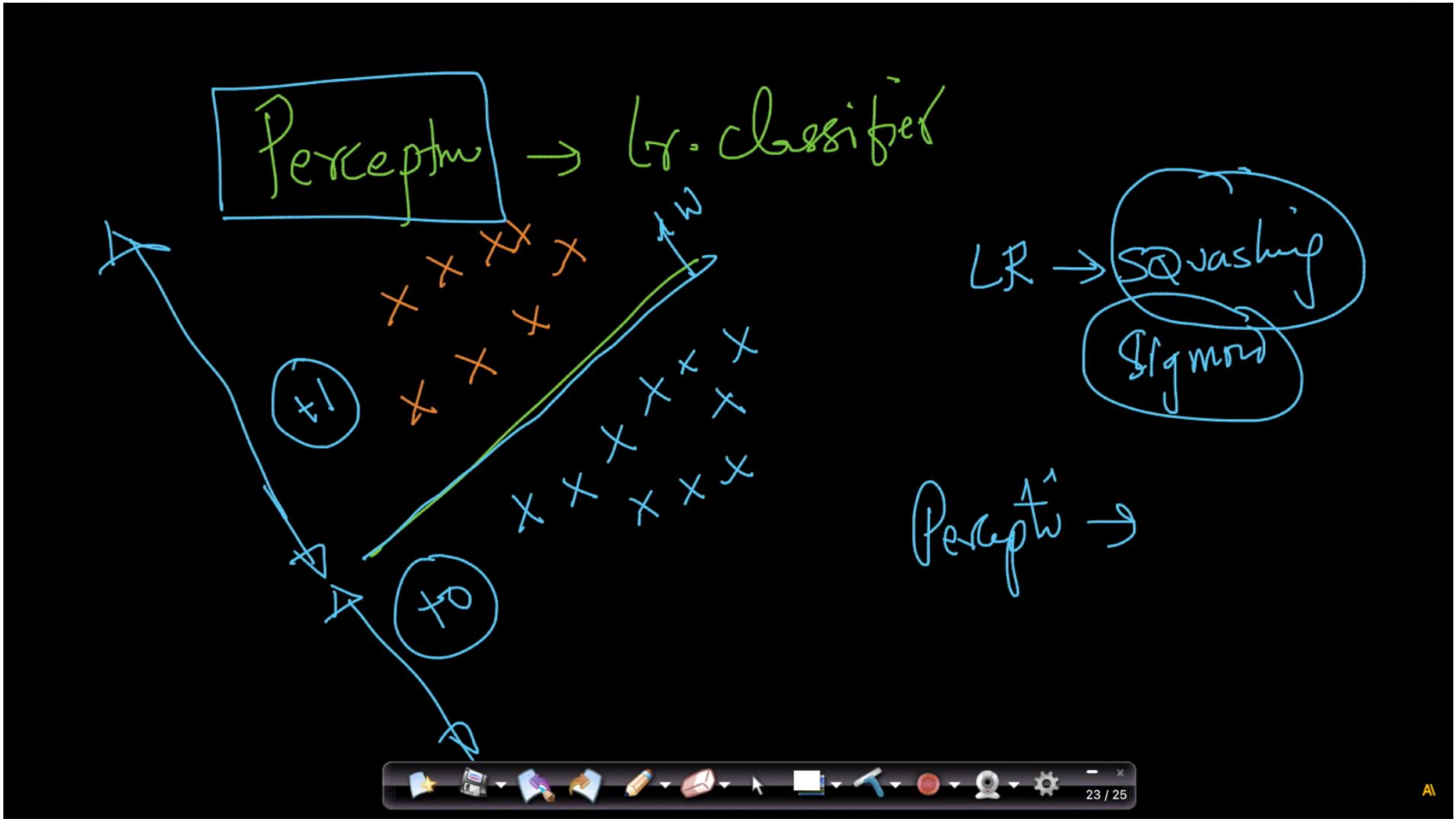
Task:- $w_i, b$ $\quad i = 1 \to d$

SGD → lr. regression & optimizn




Train a NN ⇒ Weights on edges/vertices
LR → Neuron
activation fn → Sigmoid
21 / 22


Perceptron
1957
activation fn
$\sum_{j=1}^{d} w_j x_{ij} + b$
$x_{i1}$
$w_1$
$x_{i2}$
$w_2$
$x_{i3}$
$w_3$
$w_4$
$w_d$
$x_{id}$
1
b
f
$\hat{y}_i$
$f(x_i) = \begin{cases} 1 & \text{if } w^T x_i + b \geq 0 \\ 0 & \text{otherwise} \end{cases}$
22 / 22


Perceptron → Lr. classifier
LR → Sqashing
Sigmoid
Percepti →
+1
+0
w
23 / 25


↓

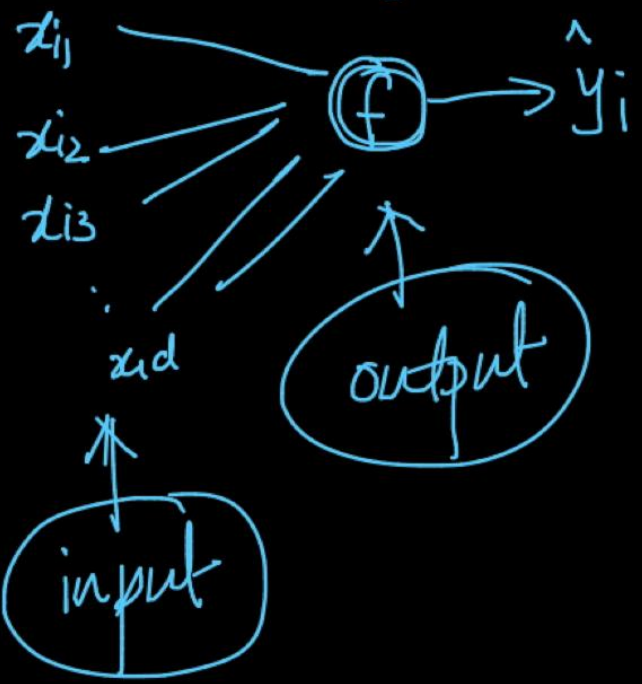
Simple single Neuron models

# Multi-layered Perceptron (MLP)

Perceptron → Single Neuron

↳ logistic-regression

Single-Neuron

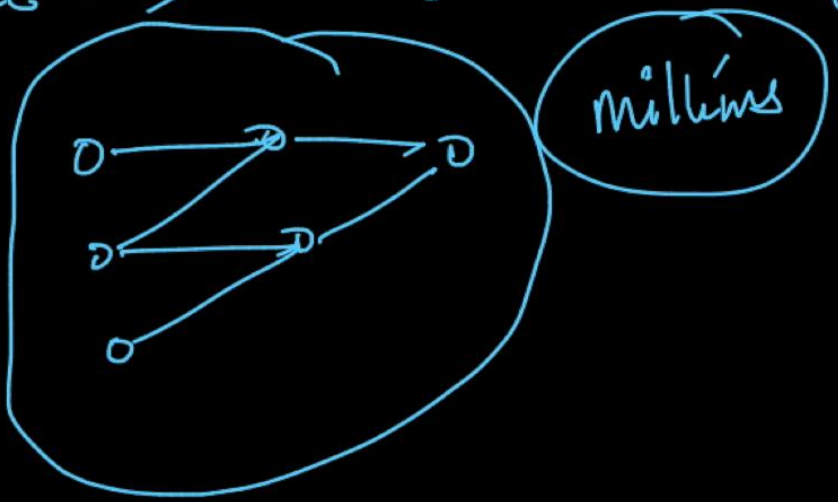



{ Connected neurons network

# (Q) Why should we Care about MLP?

## (a) biological inspiration

Neuroscience → humans, rats, monkeys ✓ ✓ ✓




Neuron → Perceptron

(c) Mathematical

regression

$\{x_i, y_i\} = D$

$y_i = F(x_i)$

$y_i \in \mathbb{R}$

Contents - Google Docs | plot(2*sin(x^2)+sqrt(x*5)) - G | Chekuri Srikan...

Secure https://www.google.co.in/search?ei=EzqvWpC6HcrcvATRkJG4Aw&q=plot%282*sin%28x%5E2%29%2Bsqrt%28x*5%29%29&oq=plot%282*sin%28x...

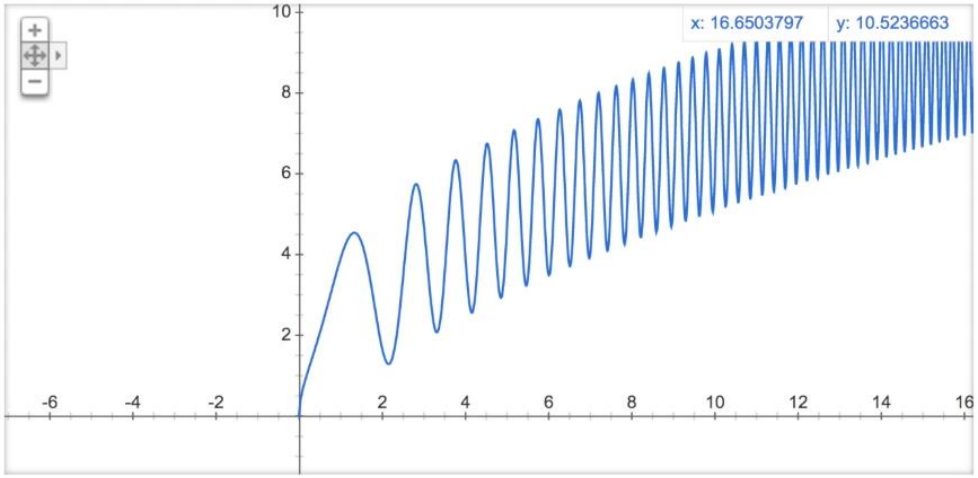
Google

plot(2*sin(x^2)+sqrt(x*5))

All Maps Images News Videos More Settings Tools

About 1,25,00,000 results (0.75 seconds)

## Graph for 2*sin(x^2)+sqrt(x*5)




More info

$F(x) = y$

$F(x) = 2\sin(x^2) + \sqrt{5x}$

### GraphSketch

https://graphsketch.com/

GraphSketch.com. **Graph Plot** Save Click here to download this **graph**. Permalink Permanent link to this **graph** page. ... **graph** of a function, you can just go to http://graphsketch.com/[function], like http://graphsketch.com/**sin(x)**. You can even separate multiple equations with commas, like http://graphsketch.com/**sin(**

Wolfram|Alpha Examples: Plotting & Graphics

8 / 8

APPLIED COURSE

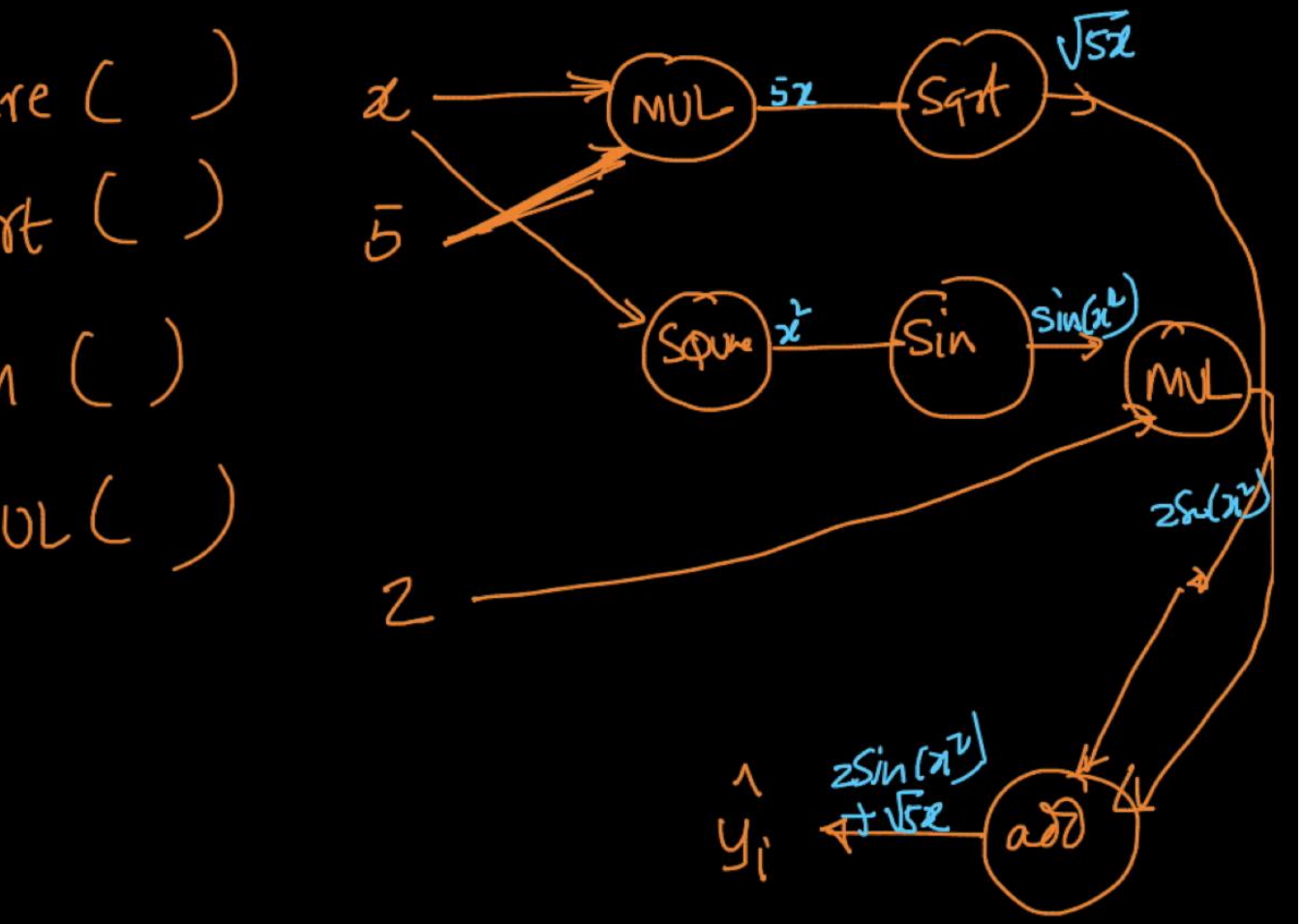
$f_1 \rightarrow$ add ( )

$f_2 \rightarrow$ square ( )

$f_3 \rightarrow$ sqrt ( )

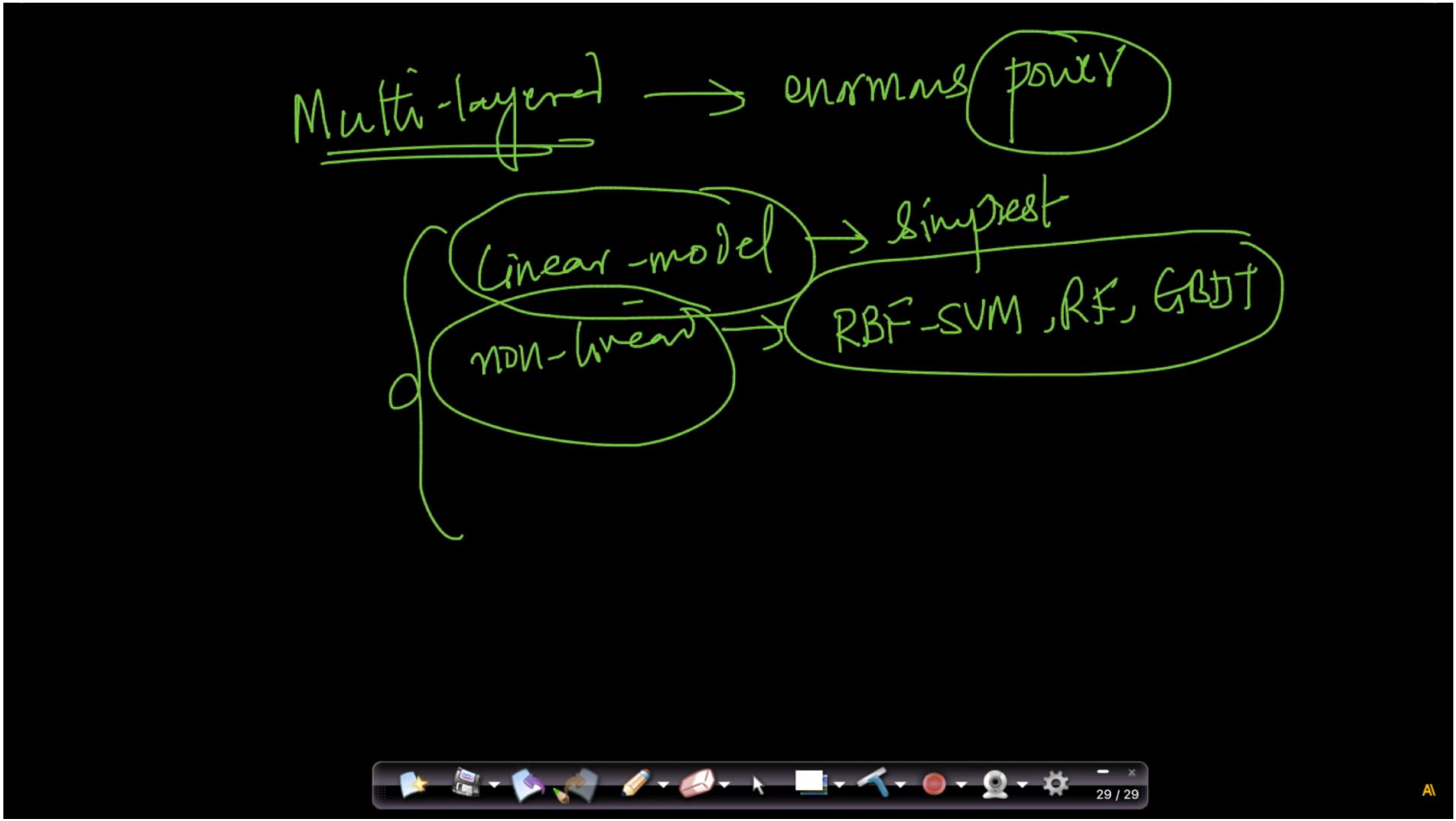
$f_4 \rightarrow$ Sin ( )

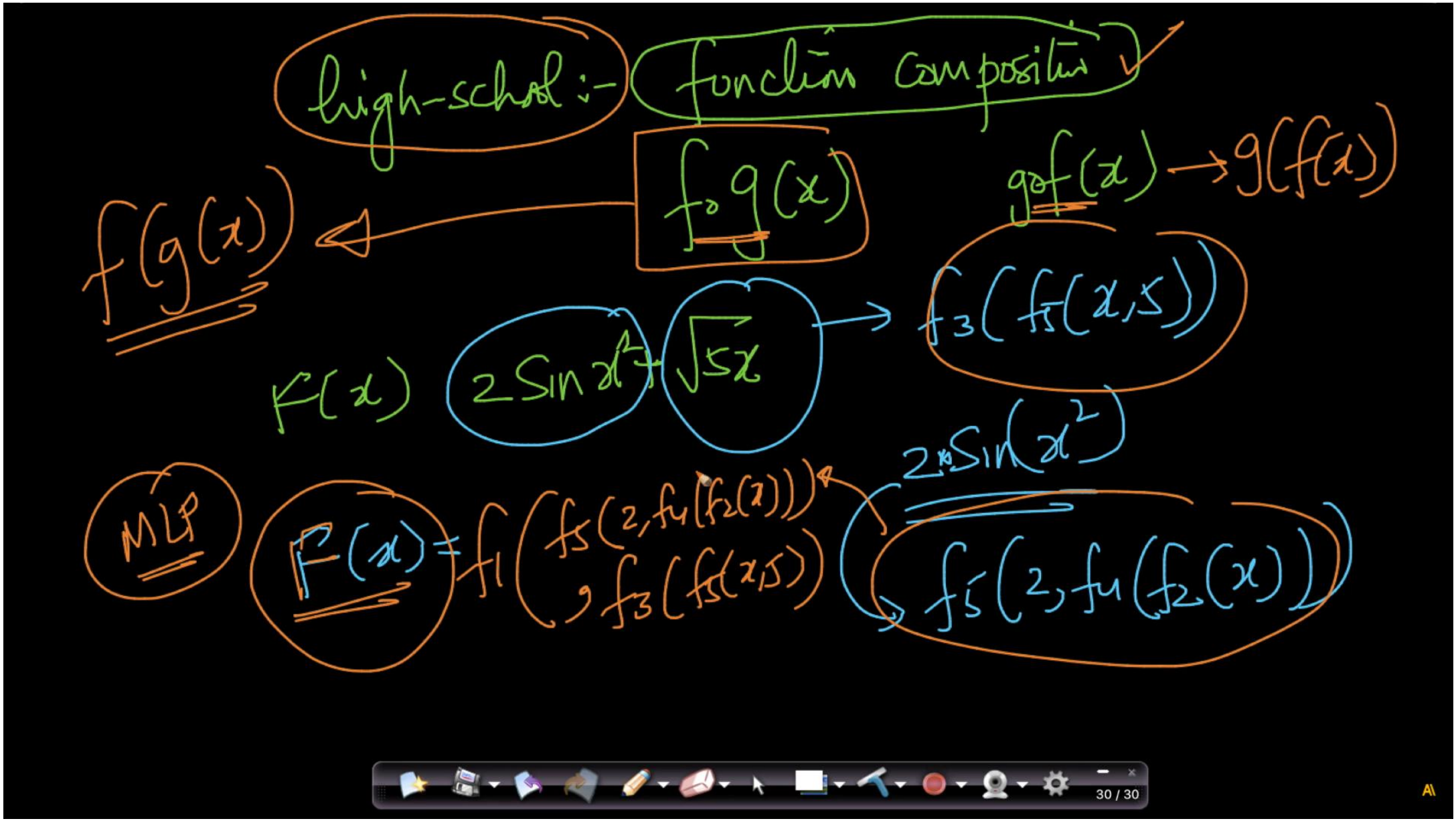
$f_5 \rightarrow$ MUL ( )



$$F(x) = 2\sin(x^2) + \sqrt{5x}$$

By using multi-layered structure,
we can come up with complex
Math-fns to solve your regression
problems

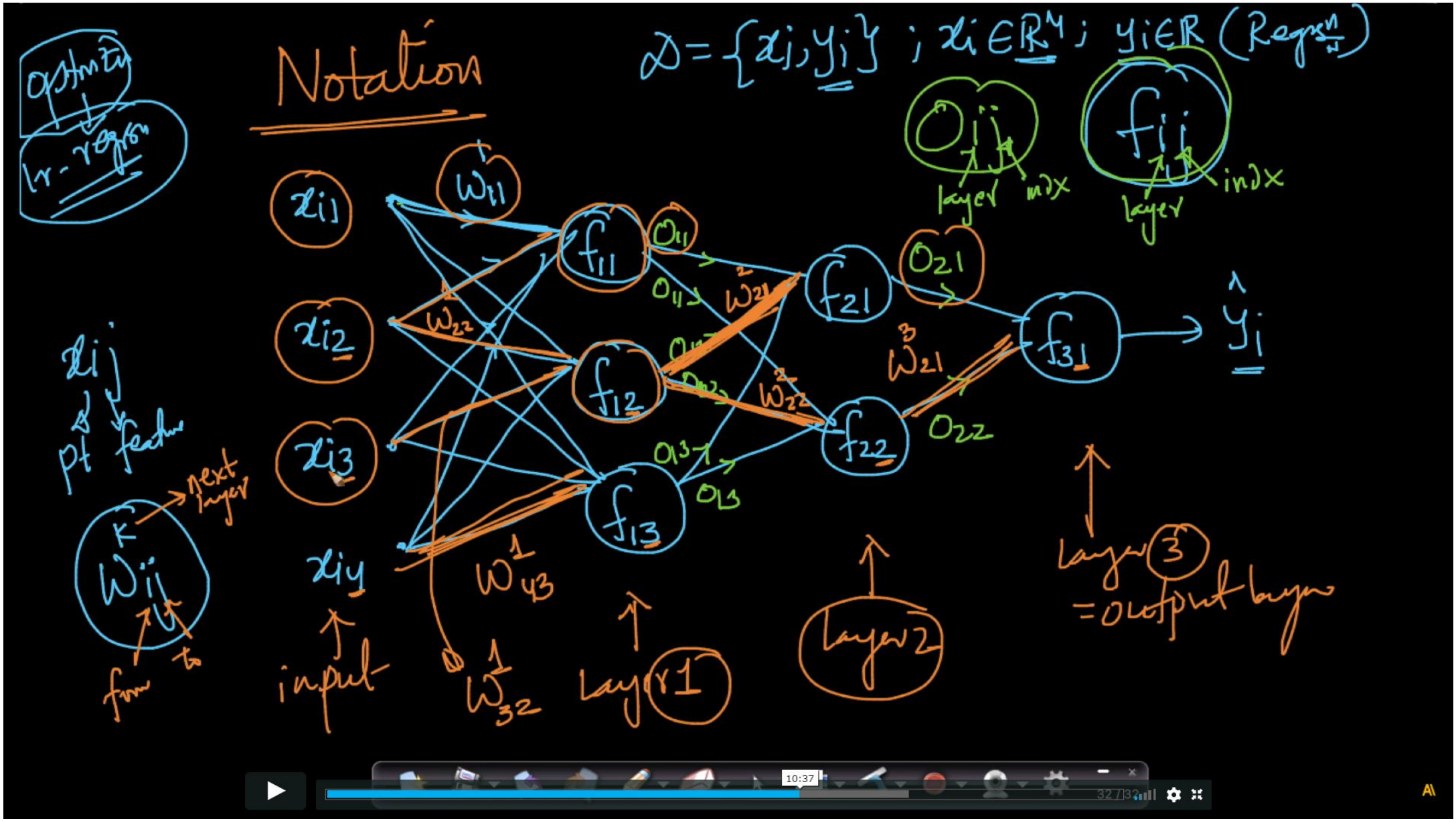

Multi-layered
enormous
power
Linear-model
simplest
non-linear
RBF-SVM, RF, GBDT

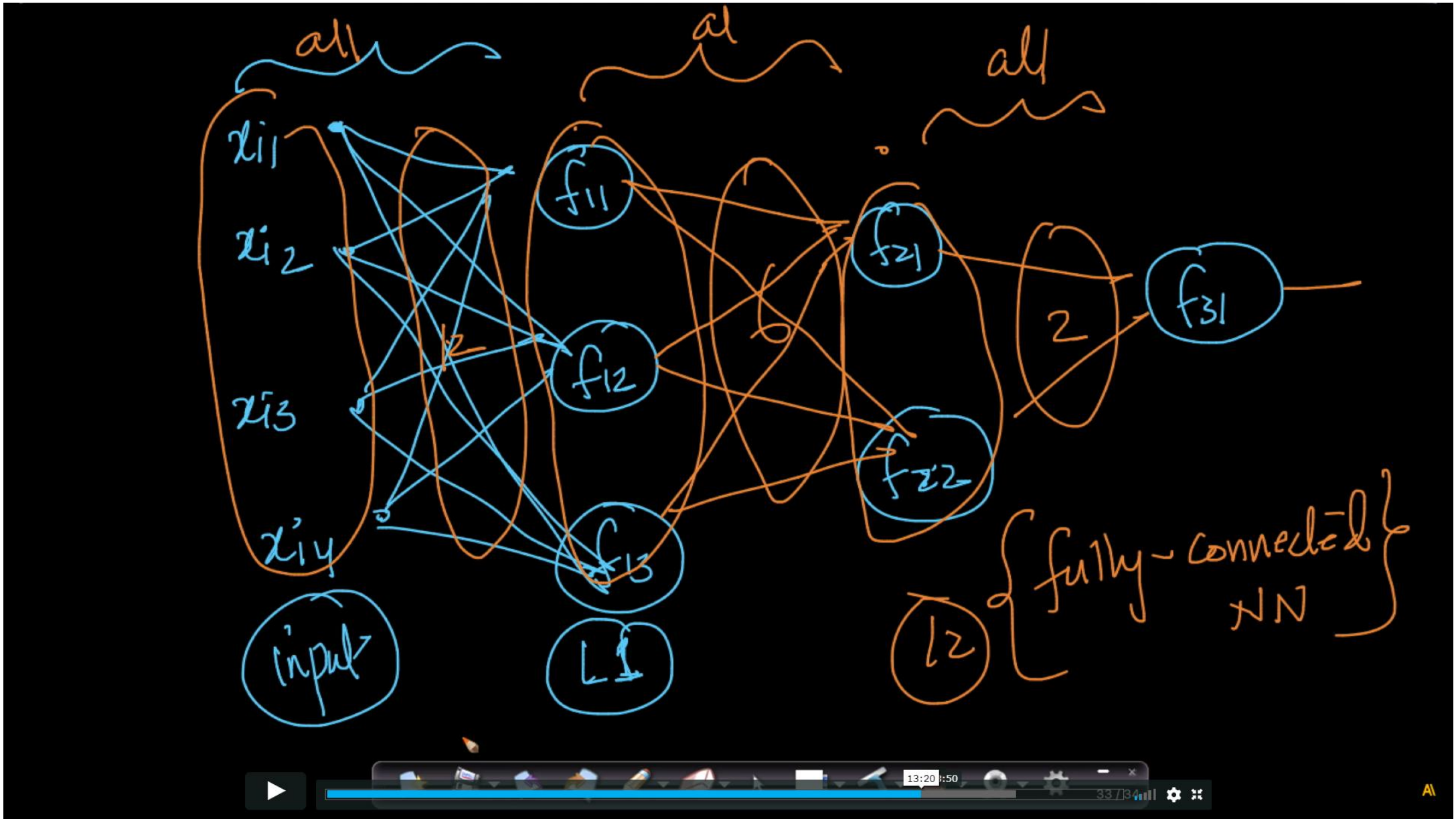

high-school :-
function composition
f∘g(x)
f(g(x))
gof(x) → g(f(x))
F(x)
2 Sin x²
√5x
f3(f1(x,5))
2*Sin(x²)
MLP
F(x) = f1(f5(2, f4(f2(x))), f3(f1(x,5)))
f5(2, f4(f2(x)))
30 / 30

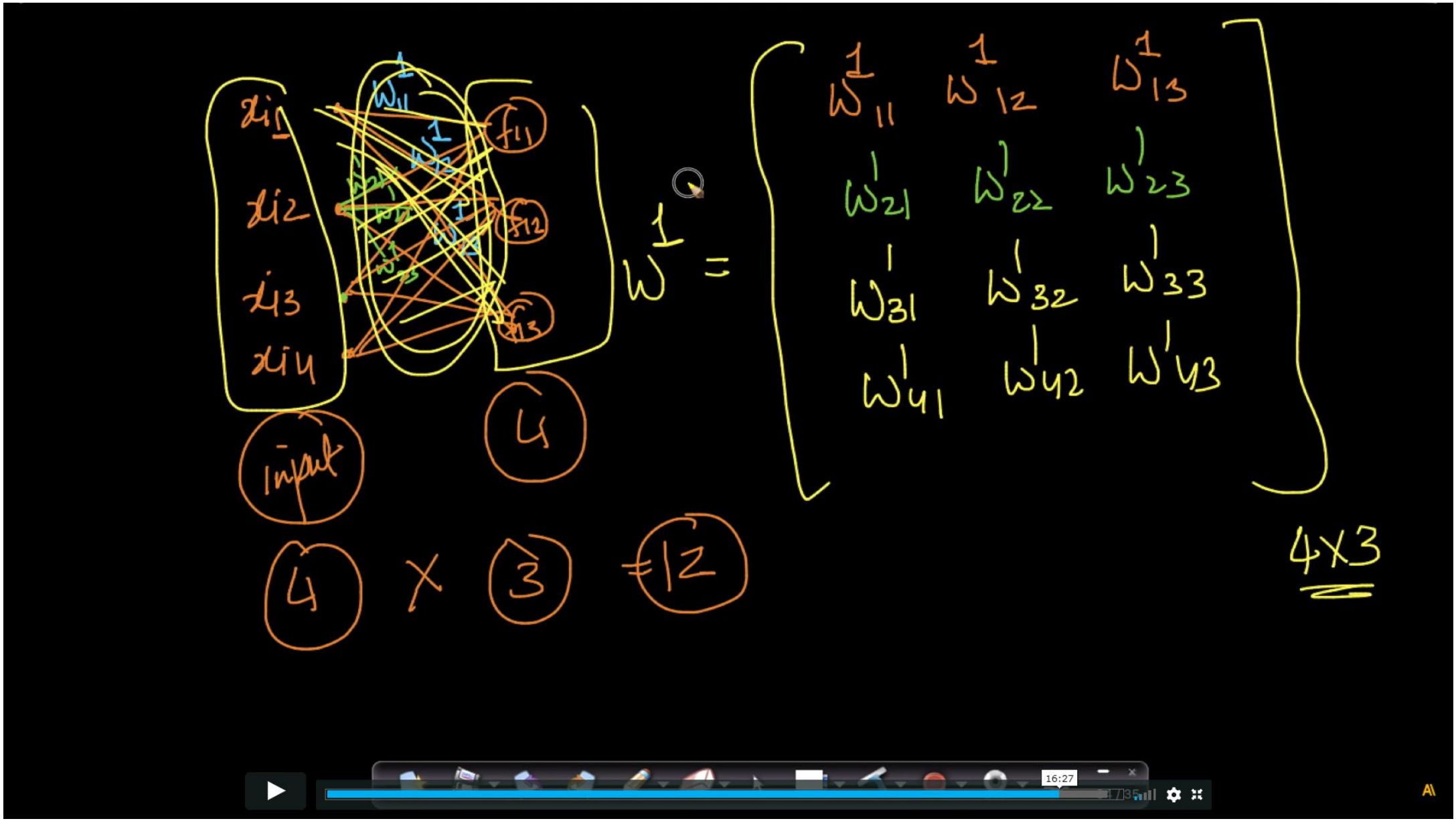
MLP :- graphical way of representing fog, gof

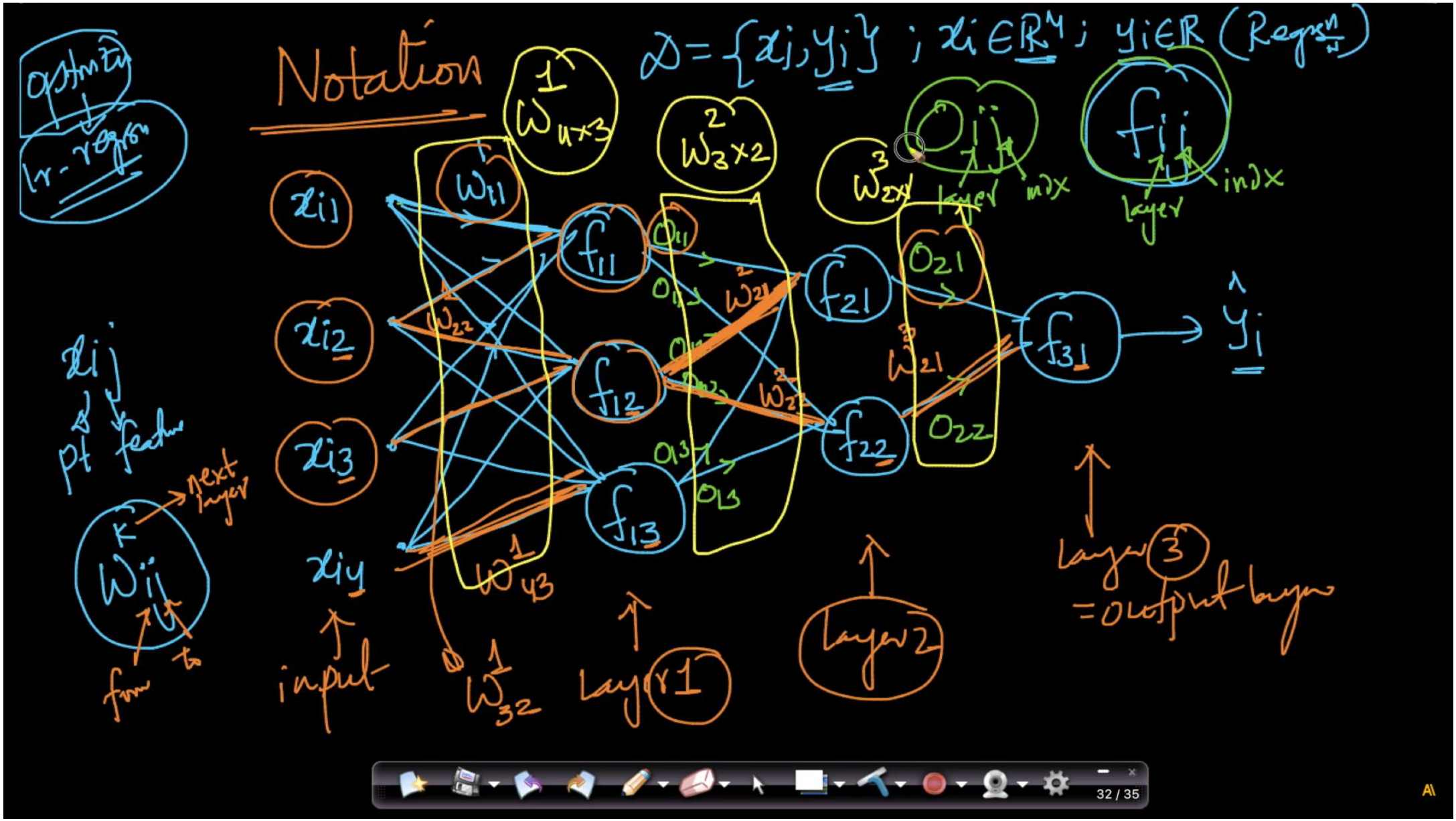
ML - Store → powerful models

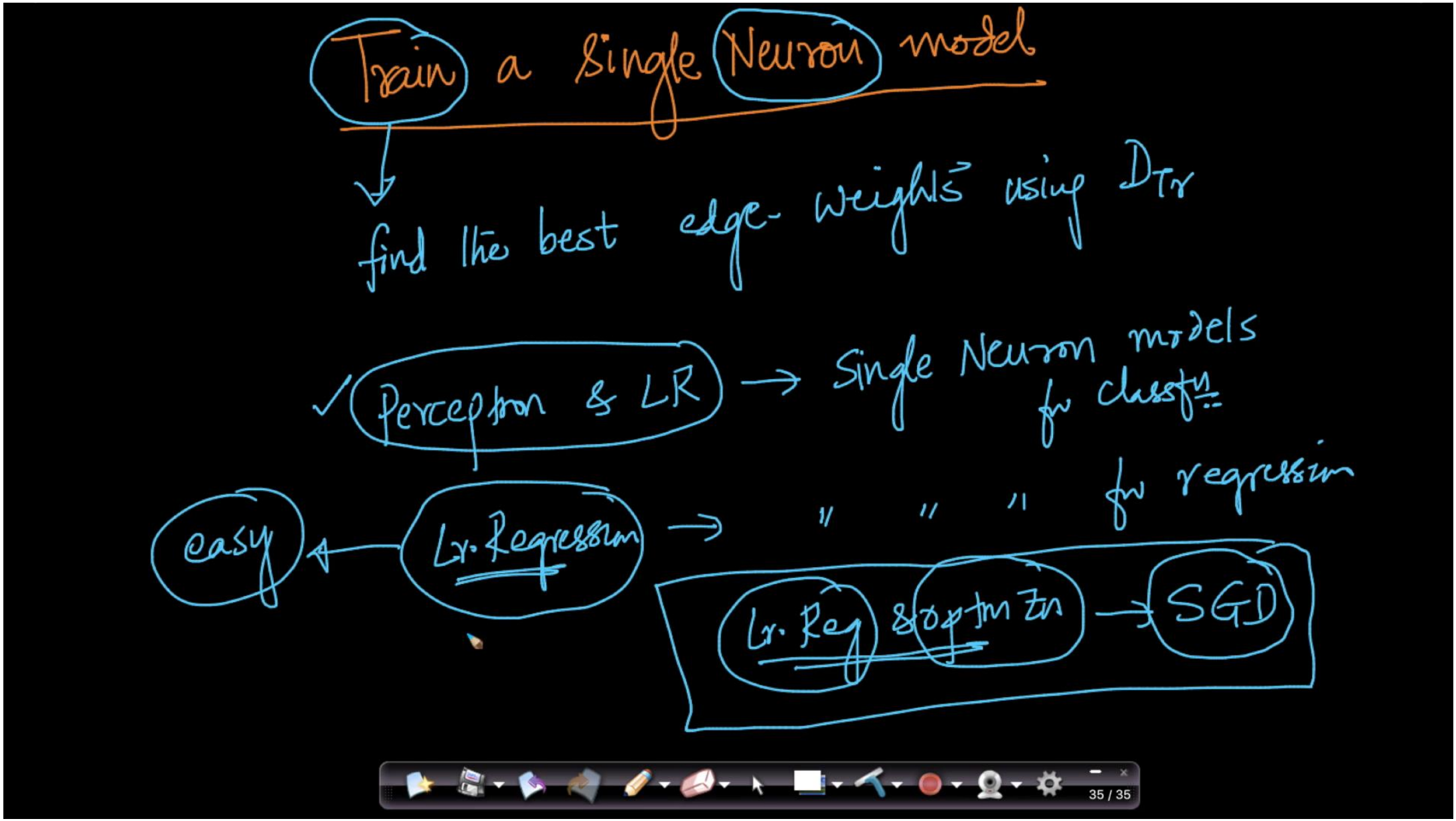
(overfit - very easily)

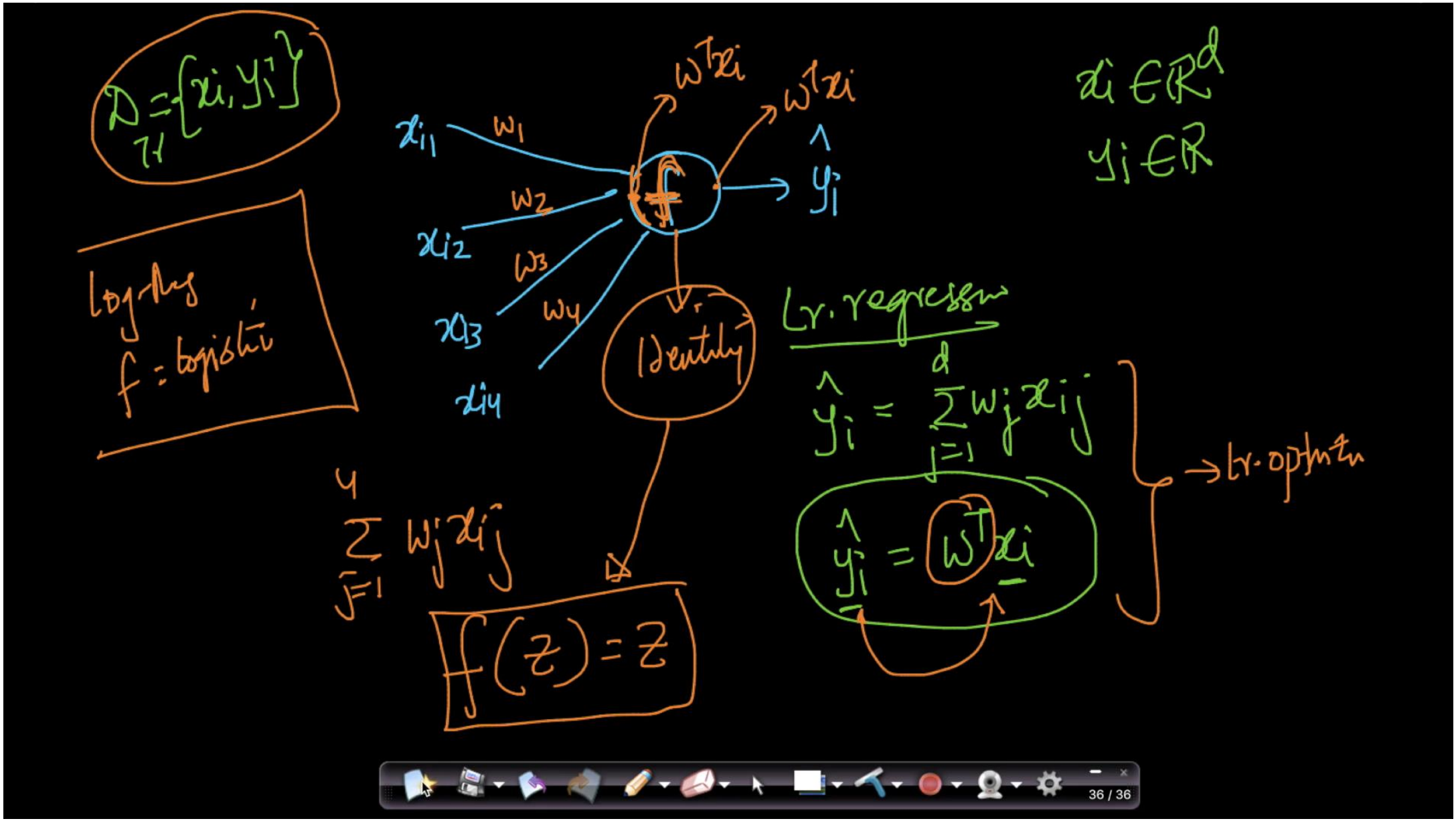

Notation
$\mathcal{D} = \{x_i, y_i\}$ ; $x_i \in \mathbb{R}^N$ ; $y_i \in \mathbb{R}$ (Regression)
optimizer
lr - regularization
$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{iy}$
input
$w_{11}$
$w_{22}$
$w^1_{43}$
$w^1_{32}$
$f_{11}$
$f_{12}$
$f_{13}$
$o_{11}$
$o_{13}$
$w^2_{21}$
$w^2_{22}$
$f_{21}$
$f_{22}$
$o_{21}$
$o_{22}$
$w^3_{21}$
$f_{31}$
$\hat{y}_i$
layer
index
$f_{ij}$
$o_{ij}$
$x_{ij}$
pt
feature
$W^k_{ij}$
next layer
from
to
layer 1
layer 2
layer 3
= output layer
10:37
32 / 32


all
al
all
$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{i4}$
$f_{11}$
$f_{12}$
$f_{13}$
$f_{21}$
$f_{22}$
2
$f_{31}$
input
L1
l2
{fully-connected NN}
13:20
33 / 34


$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{i4}$
$f_{11}$
$f_{12}$
$f_{13}$
$W^1 =$
$W^1_{11}$ $W^1_{12}$ $W^1_{13}$
$W^1_{21}$ $W^1_{22}$ $W^1_{23}$
$W^1_{31}$ $W^1_{32}$ $W^1_{33}$
$W^1_{41}$ $W^1_{42}$ $W^1_{43}$
4X3
Input
4
4 X 3 = 12
16:27


Notation
$D = \{x_i, y_i\}$ ; $x_i \in R^4$ ; $y_i \in R$ (Regression)
optimizer
Lr - regularization
$x_{ij}$
pt feature
$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{i4}$
input
$W^1_{4\times3}$
$W^2_{3\times2}$
$W^3_{2\times1}$
$W^1_{43}$
$W^1_{32}$
$f_{11}$
$f_{12}$
$f_{13}$
$f_{21}$
$f_{22}$
$f_{31}$
$O_{11}$
$O_{12}$
$O_{13}$
$O_{21}$
$O_{22}$
$W^2_{21}$
$W^2_{22}$
$W^3_{21}$
$\hat{y}_i$
layer 1
layer 2
layer 3 = output layer
$W^k_{ij}$
next layer
from
to
layer
index
$f_{ij}$
$O_{ij}$


Train a Single Neuron model
find the best edge-weights using $D_{Tr}$
✓ Perceptron & LR → Single Neuron models
fr classfn
" " " fr regression
easy
Lr. Regression →
Lr. Reg & optmzn → SGD
35 / 35


$D = \{x_i, y_i\}_{i=1}^{n}$
$x_i \in \mathbb{R}^d$
$y_i \in \mathbb{R}$
$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{i4}$
$w_1$
$w_2$
$w_3$
$w_4$
$f$
$w^T x_i$
$w^T x_i$
$\hat{y}_i$
Identity
Log-Reg
f = logistic
Lr. regression
$\hat{y}_i = \sum_{j=1}^{d} w_j x_{ij}$
$\hat{y}_i = w^T x_i$
→ Lr. optimization
$\sum_{j=1}^{4} w_j x_{ij}$
$f(z) = z$
36 / 36

## Lr. regrsn:

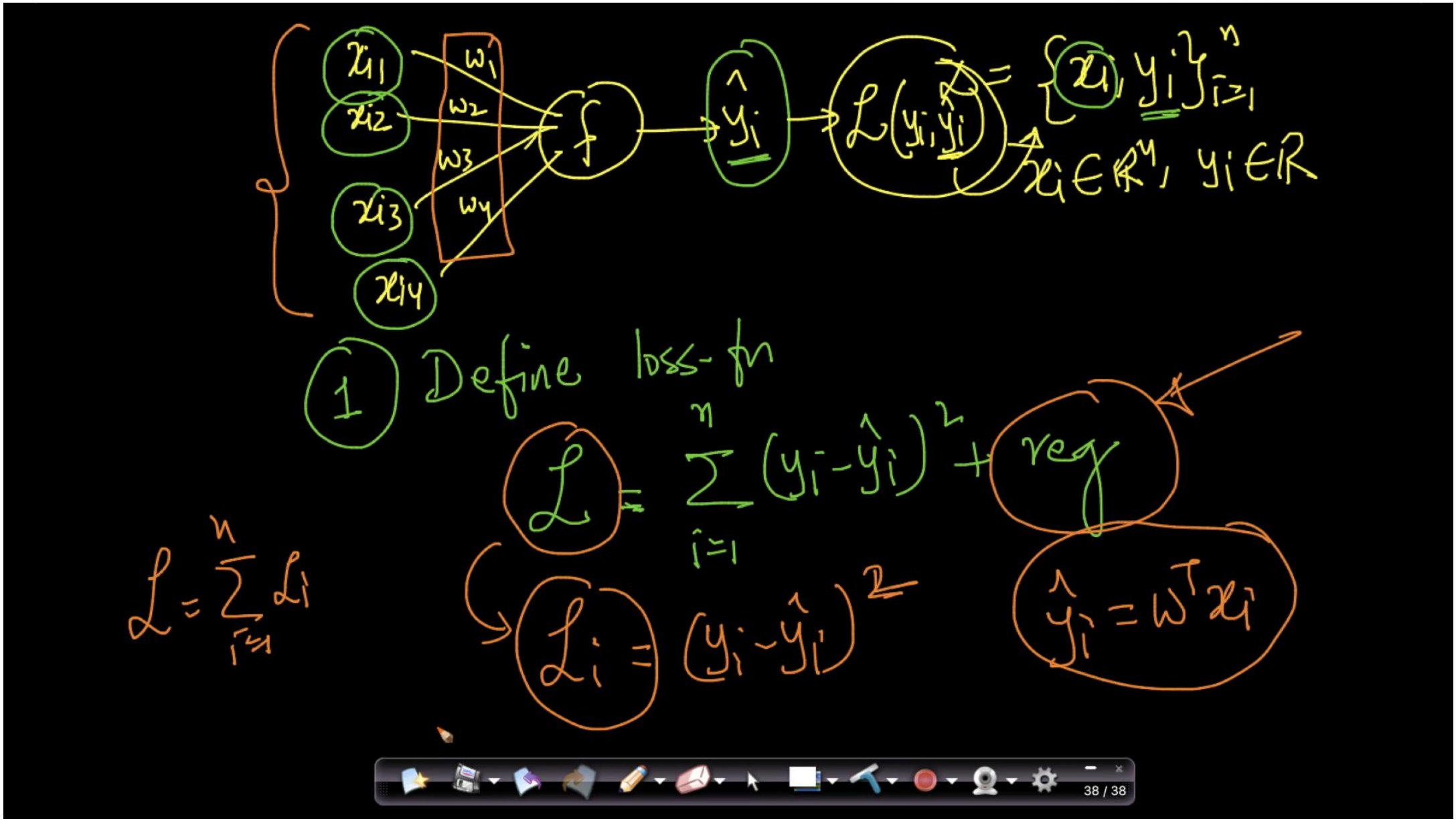
$\rightarrow$ #pts Tr.data

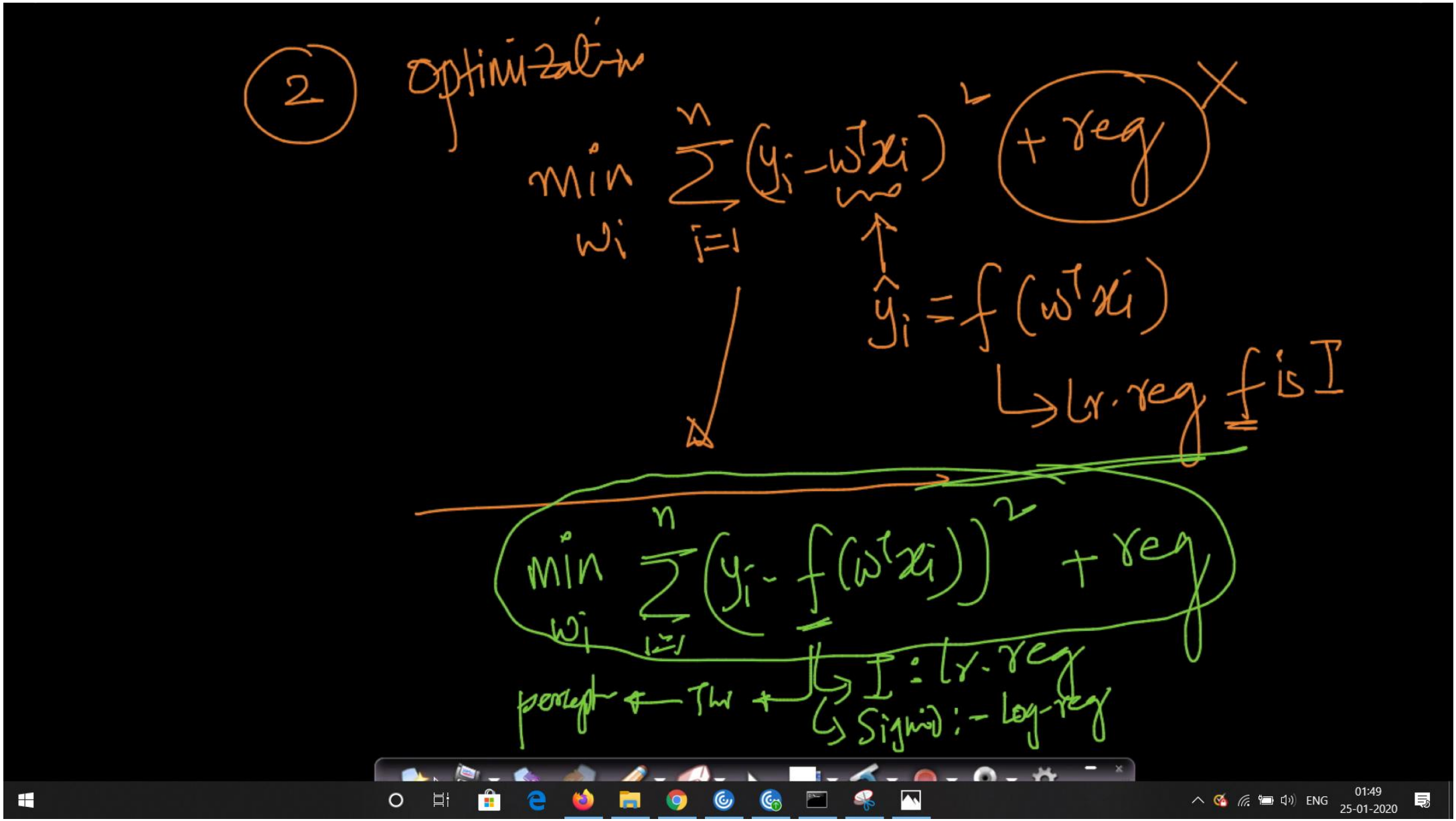
$$\min_{w_i} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \text{reg}$$

$$\hat{y}_i = w^T x_i$$

optmzn

$$\min_{w_i} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \|w\|_2^2$$


$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{i4}$
$w_1$
$w_2$
$w_3$
$w_4$
$f$
$\hat{y}_i$
$\mathcal{L}(y_i, \hat{y}_i)$
$\{x_i, y_i\}_{i=1}^{n}$
$x_i \in \mathbb{R}^4$, $y_i \in \mathbb{R}$
1 Define loss-fn
$\mathcal{L} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + reg$
$\mathcal{L} = \sum_{i=1}^{n} \mathcal{L}_i$
$\mathcal{L}_i = (y_i - \hat{y}_i)^2$
$\hat{y}_i = w^T x_i$
38 / 38


2
optimization
$\min_{w_i} \sum_{i=1}^{n} (y_i - w^T x_i)^2$
+ reg
X
$\hat{y}_i = f(w^T x_i)$
↳ lr. reg
f is I
$\min_{w_i} \sum_{i=1}^{n} (y_i - f(w^T x_i))^2 + reg$
perceptron ← Thr ← I : lr. reg
Sigmoid :- Log-reg
01:49
25-01-2020
ENG

$$w^* = \arg\min_{w} \sum_{i=1}^{n} \left(y_i - f(w^T x_i)\right)^2 + reg$$

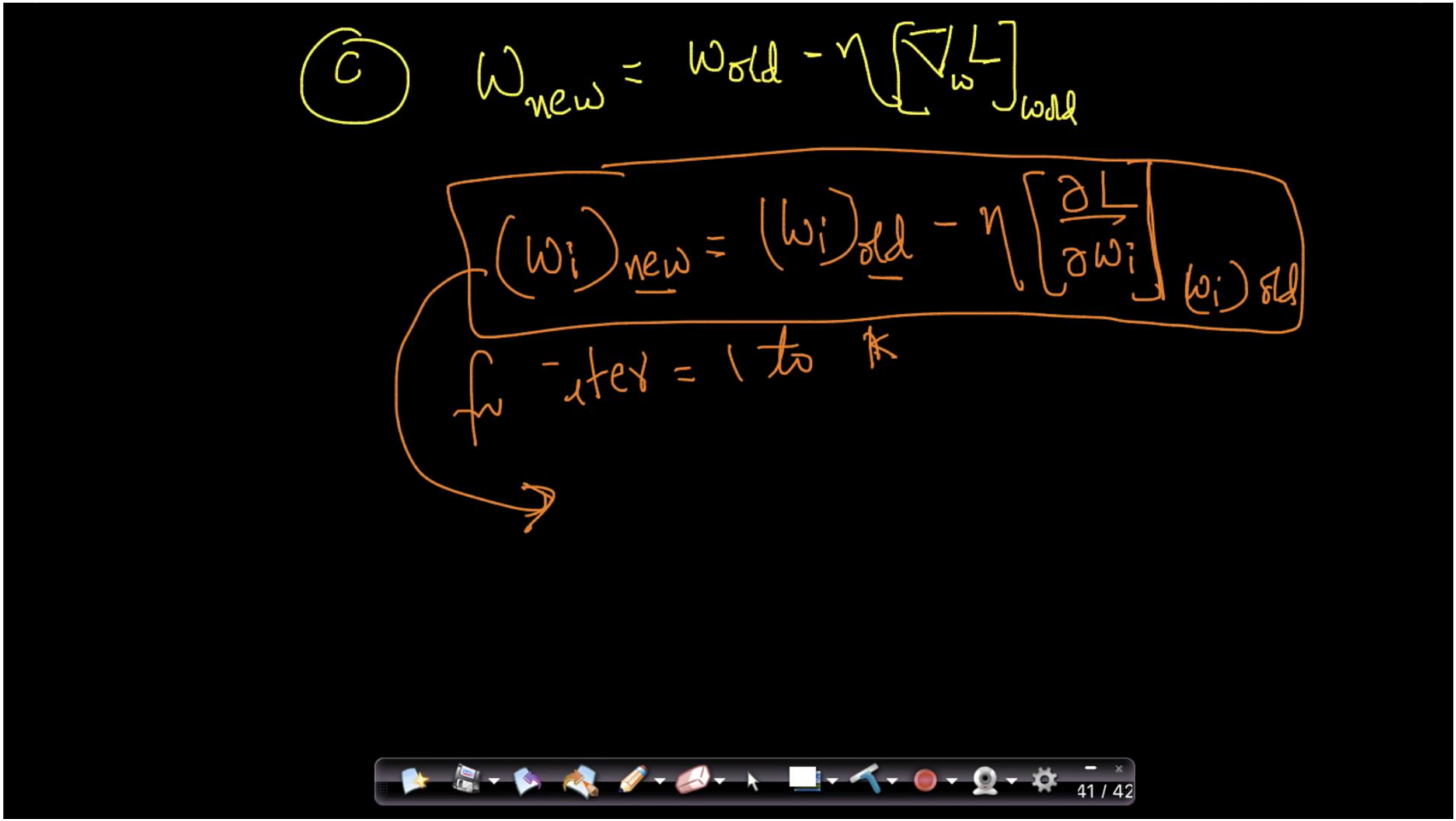
3. Solve the optimization problem

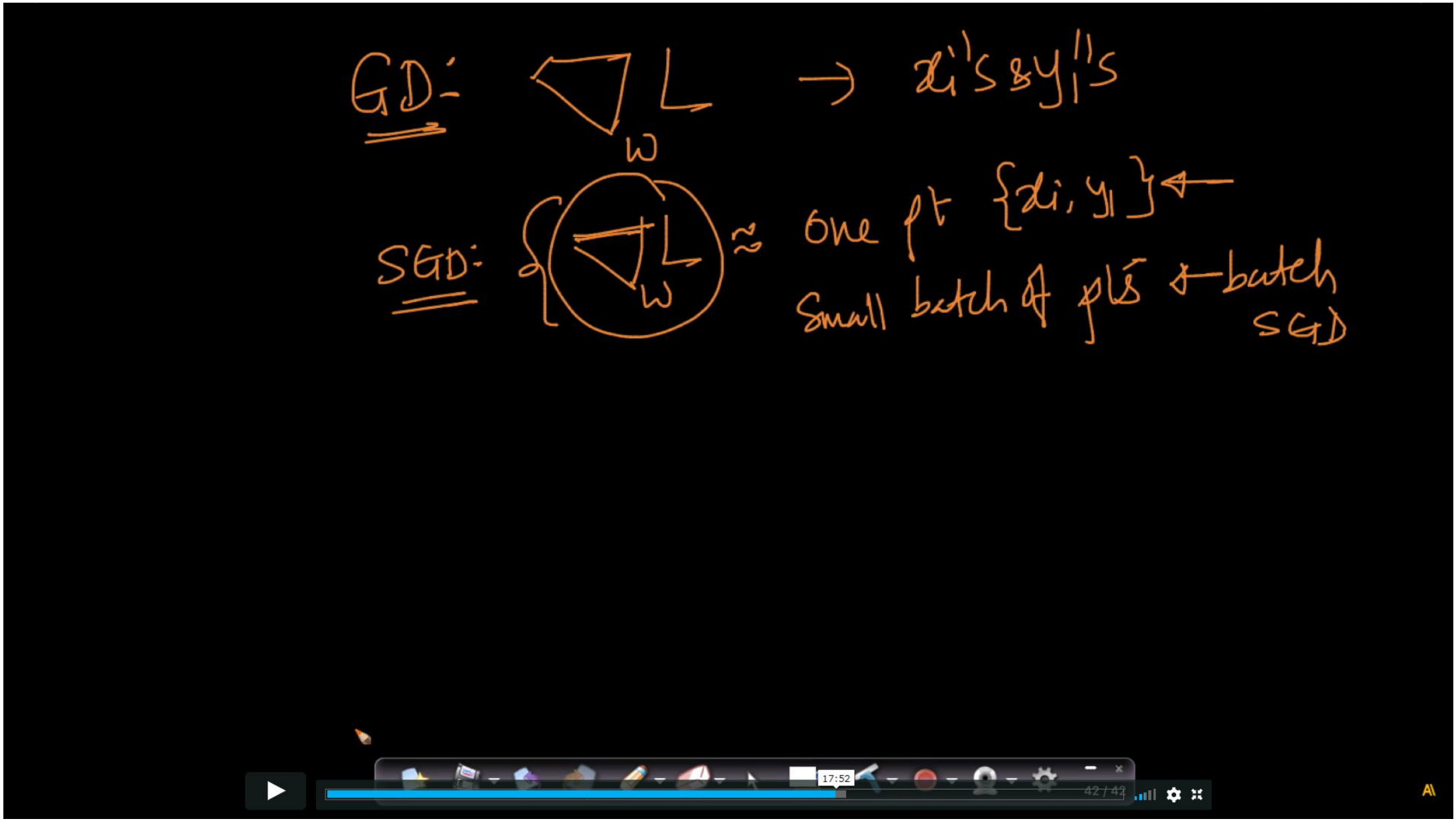
   a. initialization of $w_i$'s → random ✓ — SGD

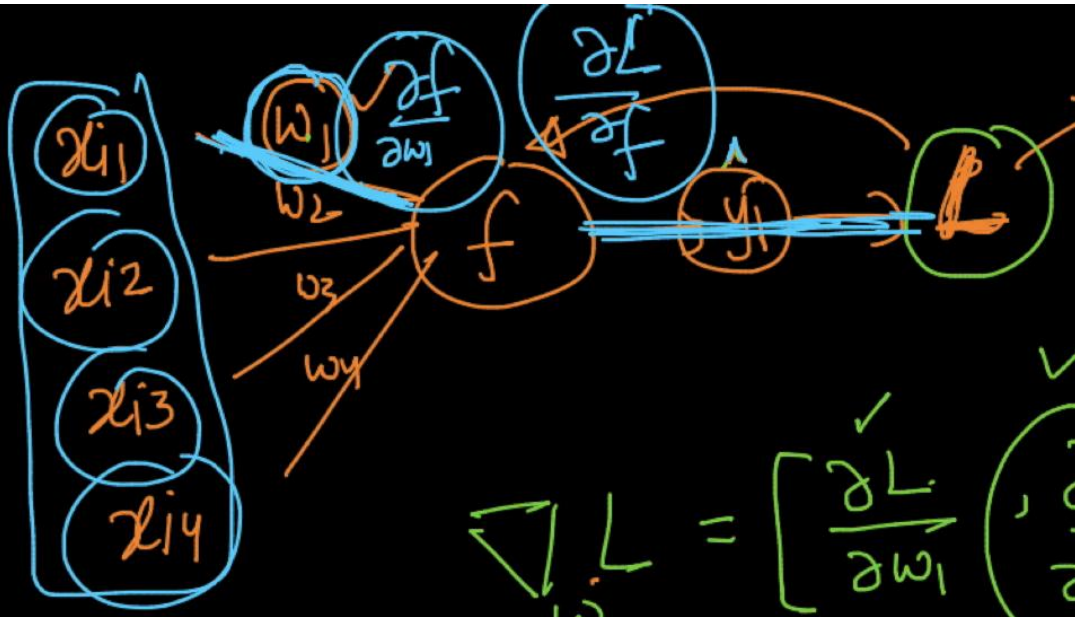
   b. $\nabla_w L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \vdots \\ \frac{\partial L}{\partial w_4} \end{bmatrix}$

$w \in \mathbb{R}^4$

$x_i \in \mathbb{R}^4$


C
$w_{new} = w_{old} - \eta [\nabla_w L]_{wold}$
$(w_i)_{new} = (w_i)_{old} - \eta \left[\frac{\partial L}{\partial w_i}\right]_{(w_i)old}$
for iteration = 1 to *
41 / 42


GD:
$\nabla_w L$
$\rightarrow$ $x_i$'s & $y_i$'s
SGD:
$\approx$
one pt $\{x_i, y_i\}$
small batch of pts
batch SGD
17:52
42 / 43

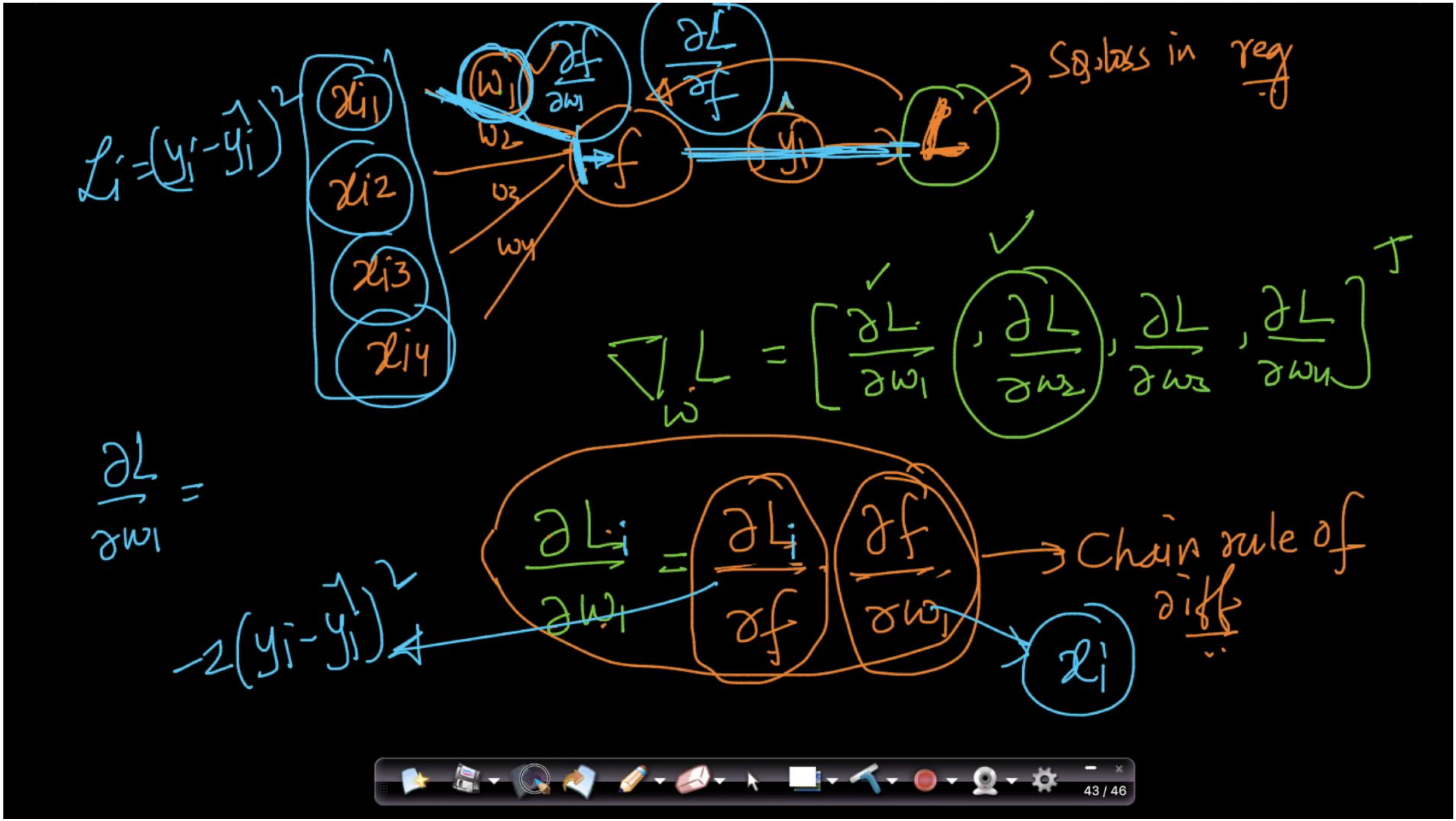


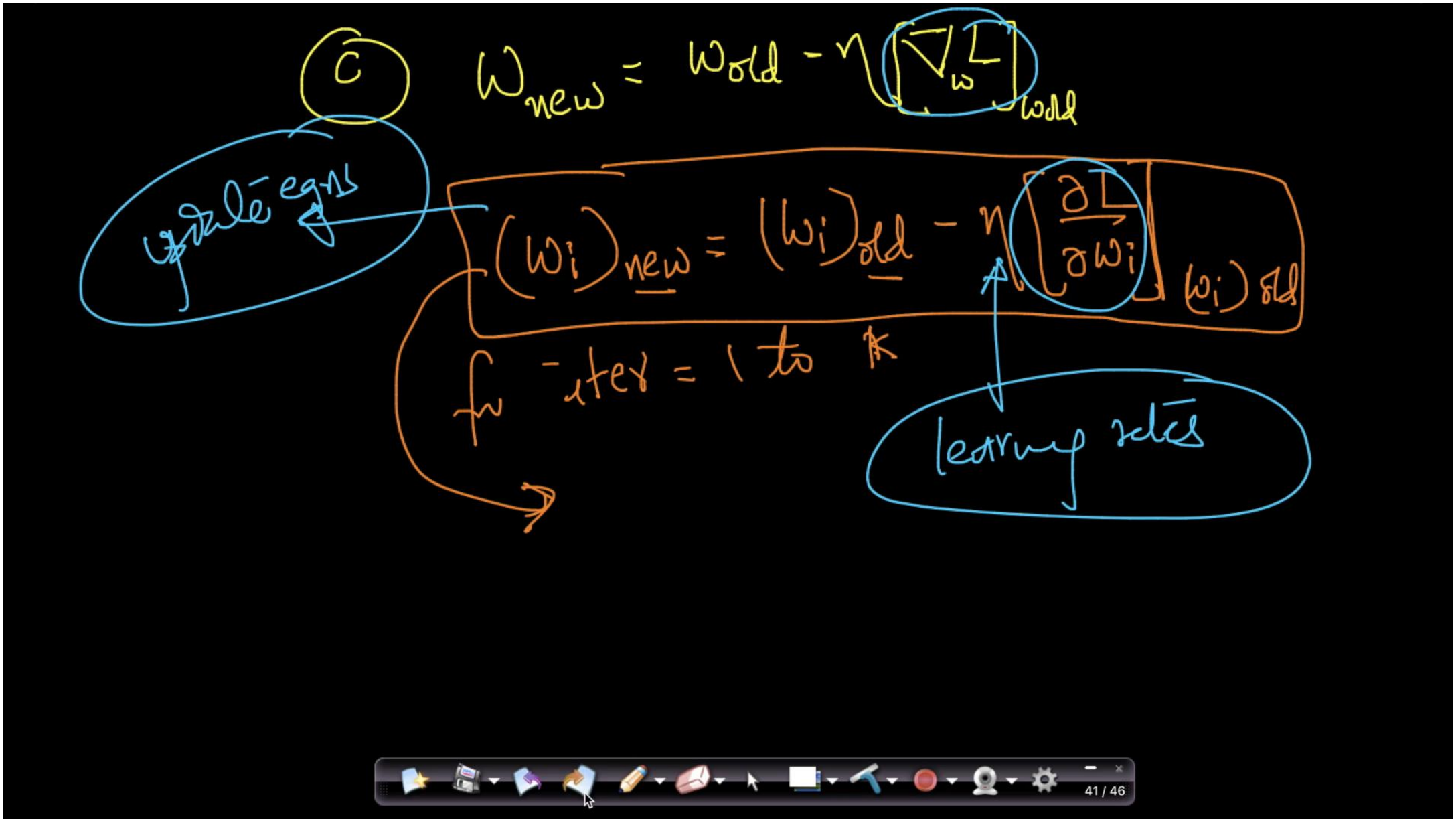

$$\nabla_w L = \left[\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \frac{\partial L}{\partial w_3}, \frac{\partial L}{\partial w_4}\right]^T$$

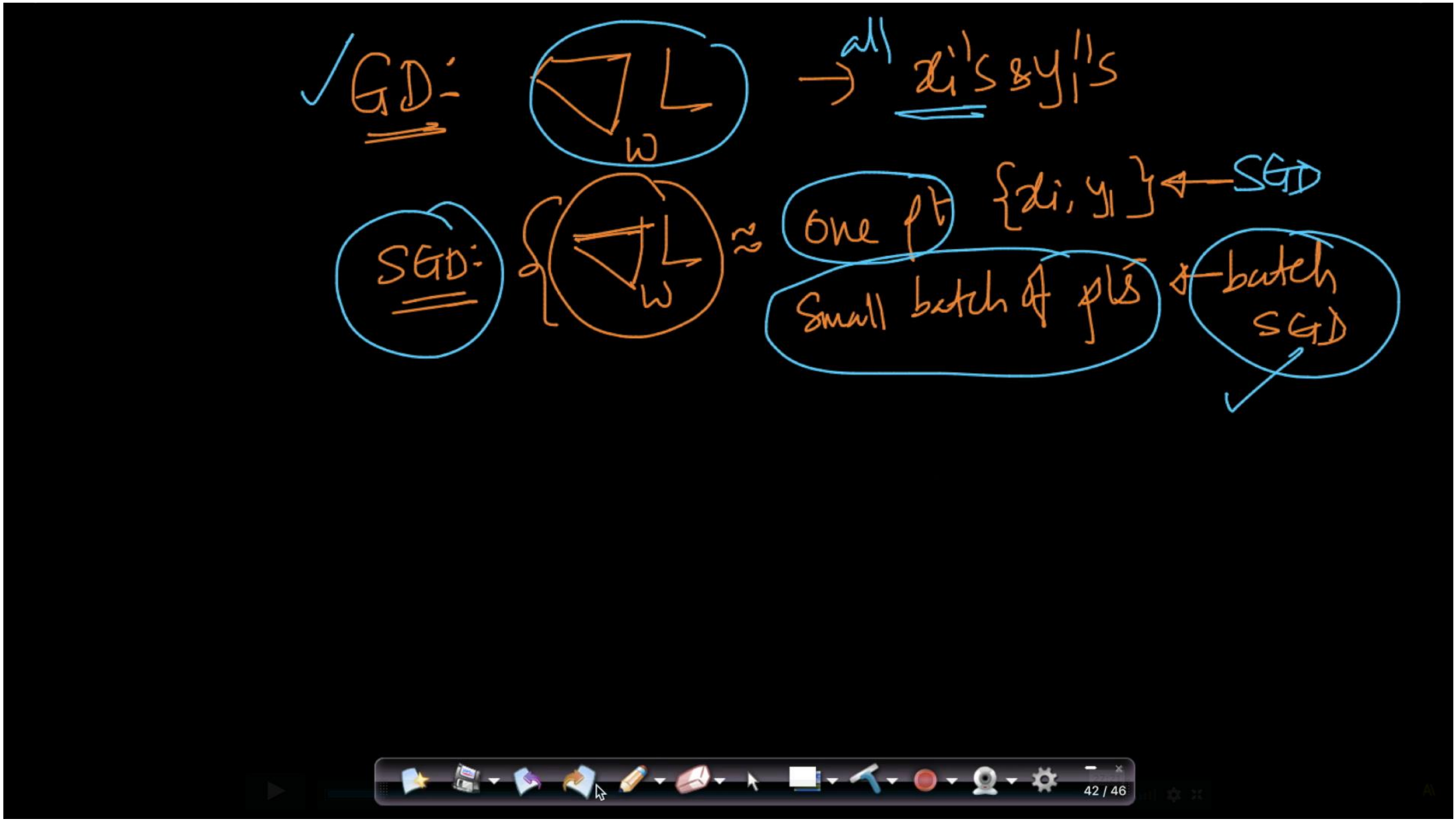
$$\frac{\partial L}{\partial w_1} =$$

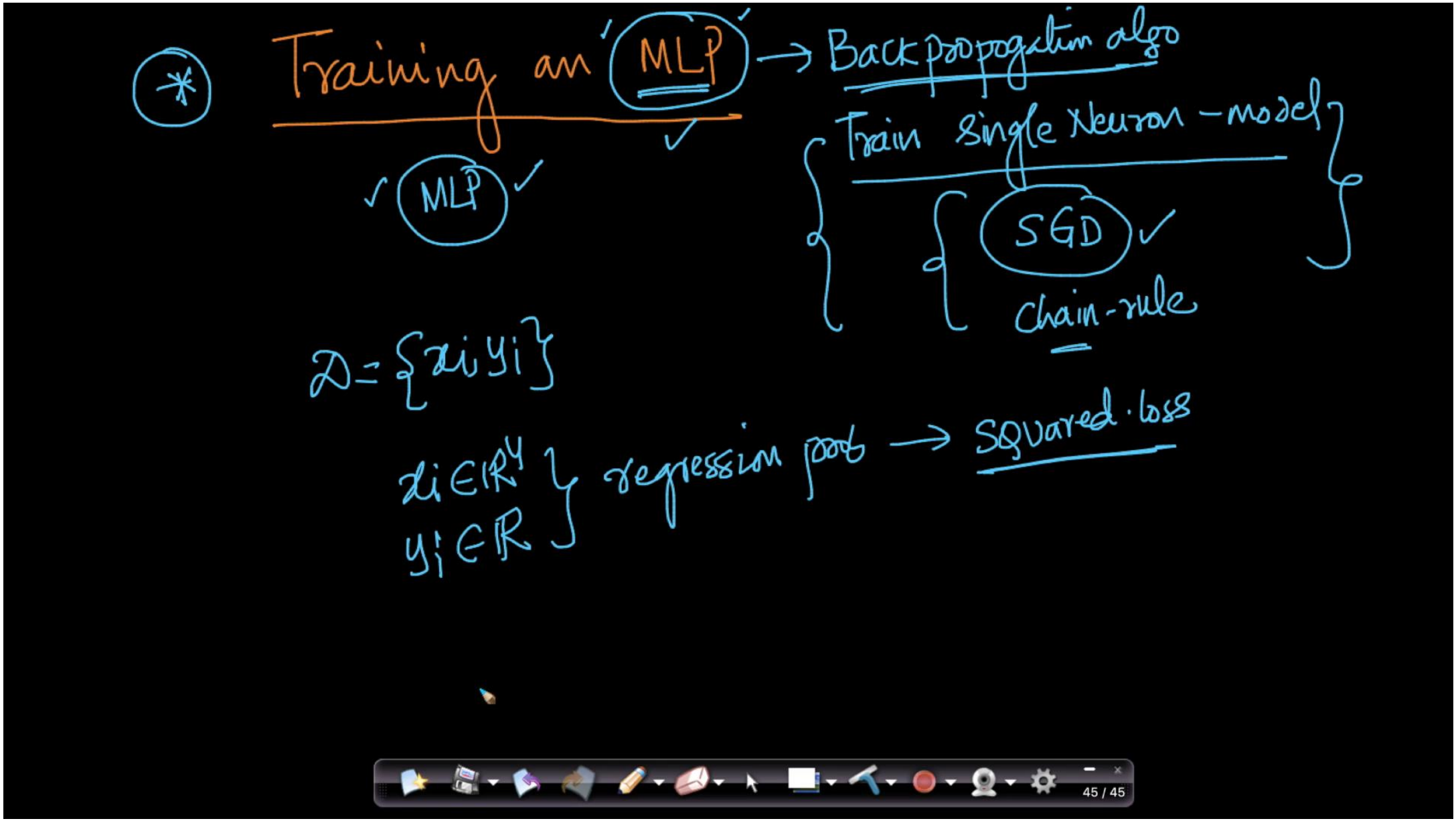
$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial f} \frac{\partial f}{\partial w_1}$$

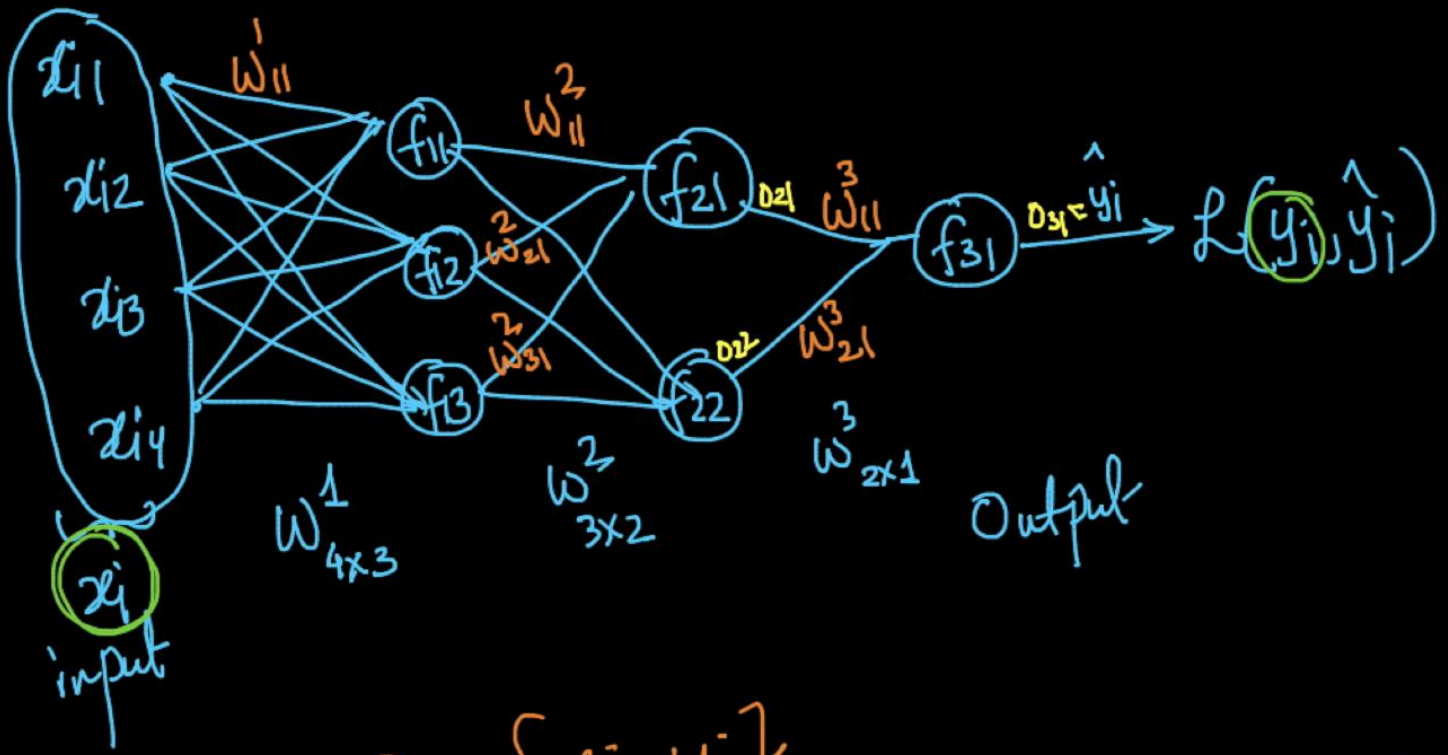
→ Chain rule of diff


$L_i = (y_i - \hat{y}_i)^2$
$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{i4}$
$w_1$
$w_2$
$w_3$
$w_4$
$\frac{\partial f}{\partial w_1}$
$f$
$\frac{\partial L}{\partial f}$
$\hat{y}_i$
$L$
Sq. loss in reg.
$\nabla_w L = \left[\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \frac{\partial L}{\partial w_3}, \frac{\partial L}{\partial w_4}\right]^T$
$\frac{\partial L}{\partial w_1} =$
$\frac{\partial L_i}{\partial w_1} = \frac{\partial L_i}{\partial f} \frac{\partial f}{\partial w_1}$
Chain rule of diff
$-2(y_i - \hat{y}_i)$
$x_i$


C
$w_{new} = w_{old} - \eta \nabla_w L \Big|_{w_{old}}$
update eqns
$(w_i)_{new} = (w_i)_{old} - \eta \frac{\partial L}{\partial w_i} \Big|_{(w_i)_{old}}$
for iter = 1 to *
learning rates


GD:
$\nabla_w L$
$\rightarrow^{all}$ $x_i$'s & $y_i$'s
SGD:
$\nabla_w L \approx$
one pt
$\{x_i, y_i\}$ ← SGD
Small batch of pts
← batch SGD

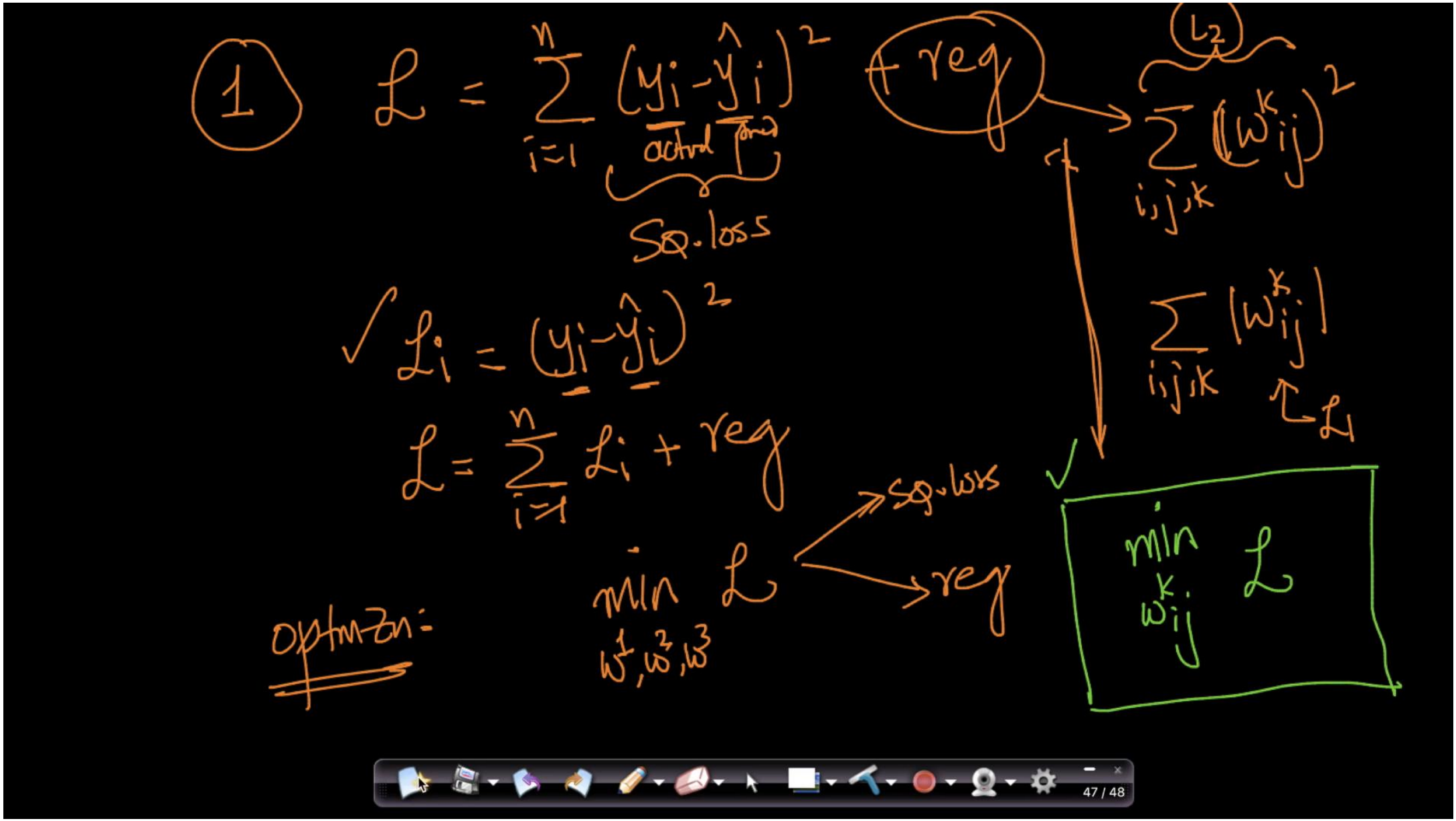

Training an MLP
MLP
Backpropogation algo
Train Single Neuron - model
SGD
Chain-rule
$\mathcal{D} = \{x_i, y_i\}$
$x_i \in \mathbb{R}^4$
$y_i \in \mathbb{R}$
regression prob
Squared loss
45 / 45

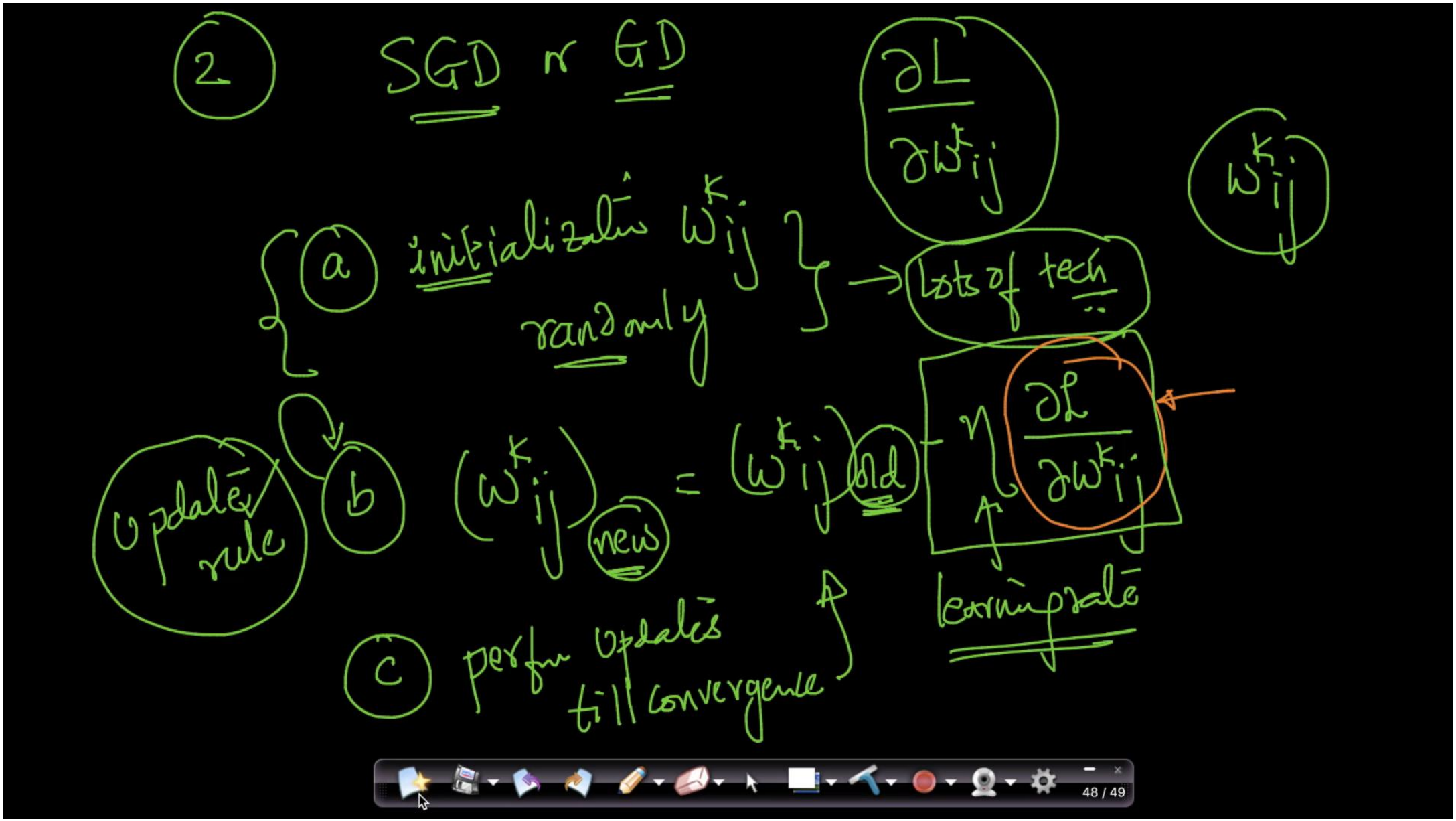


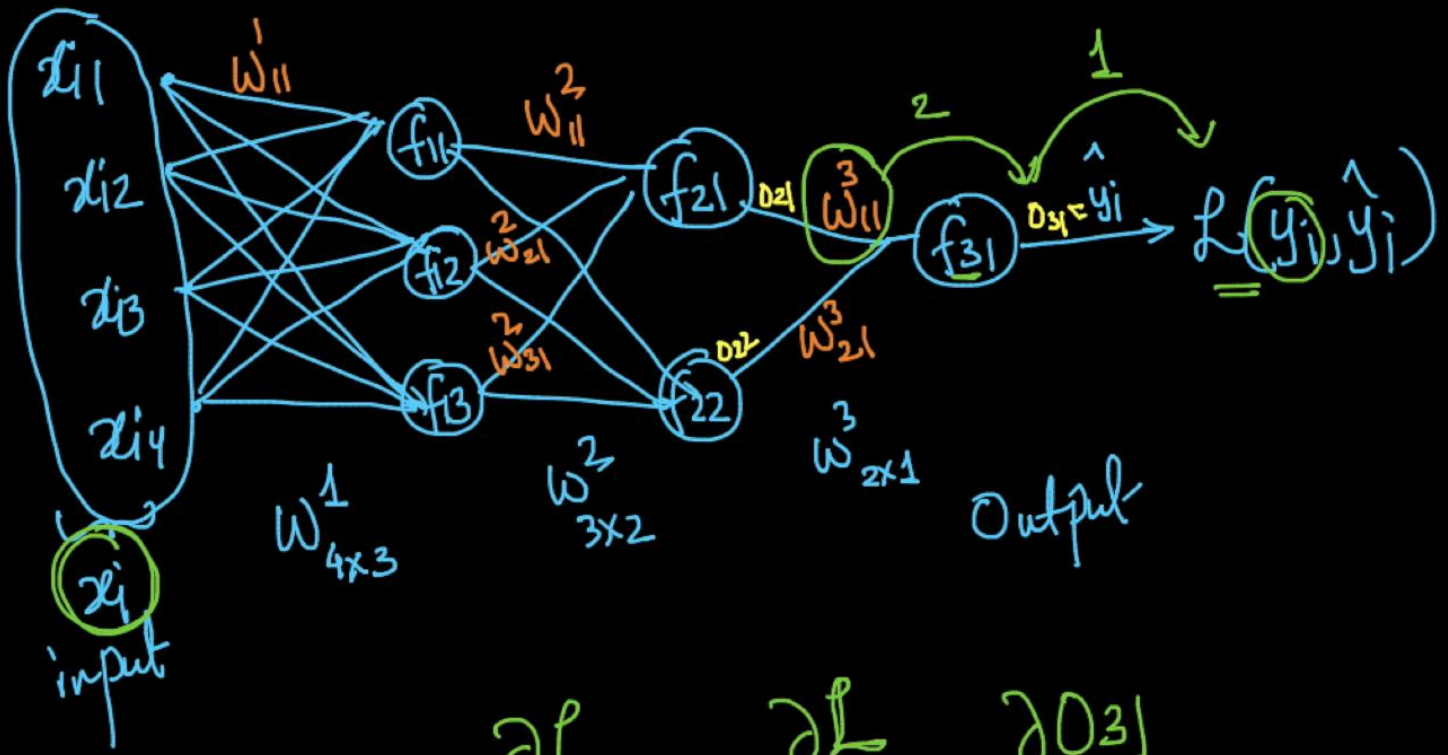

$$D = \{x_i, y_i\}$$

determine $w^1_{4\times3}, w^2_{3\times2}, w^3_{2\times1}$ = 20 weights

12 6 2


1
$\mathcal{L} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$
actual pred
Sq. loss
+ reg
$L_2$
$\sum_{i,j,k} (w_{ij}^k)^2$
$\mathcal{L}_i = (y_i - \hat{y}_i)^2$
$\mathcal{L} = \sum_{i=1}^{n} \mathcal{L}_i + reg$
$\sum_{i,j,k} |w_{ij}^k|$
$L_1$
optimize:
$\min_{w^1, w^2, w^3} \mathcal{L}$
Sq. loss
reg
$\min_{w_{ij}^k} \mathcal{L}$
47 / 48


2
SGD or GD
a
initialize $w_{ij}^k$ randomly
lots of tech
$\frac{\partial L}{\partial w_{ij}^k}$
$w_{ij}^k$
b
update rule
$(w_{ij}^k)_{new} = (w_{ij}^k)_{old} - \eta \frac{\partial L}{\partial w_{ij}^k}$
learning rate
c
perform updates till convergence
48 / 49



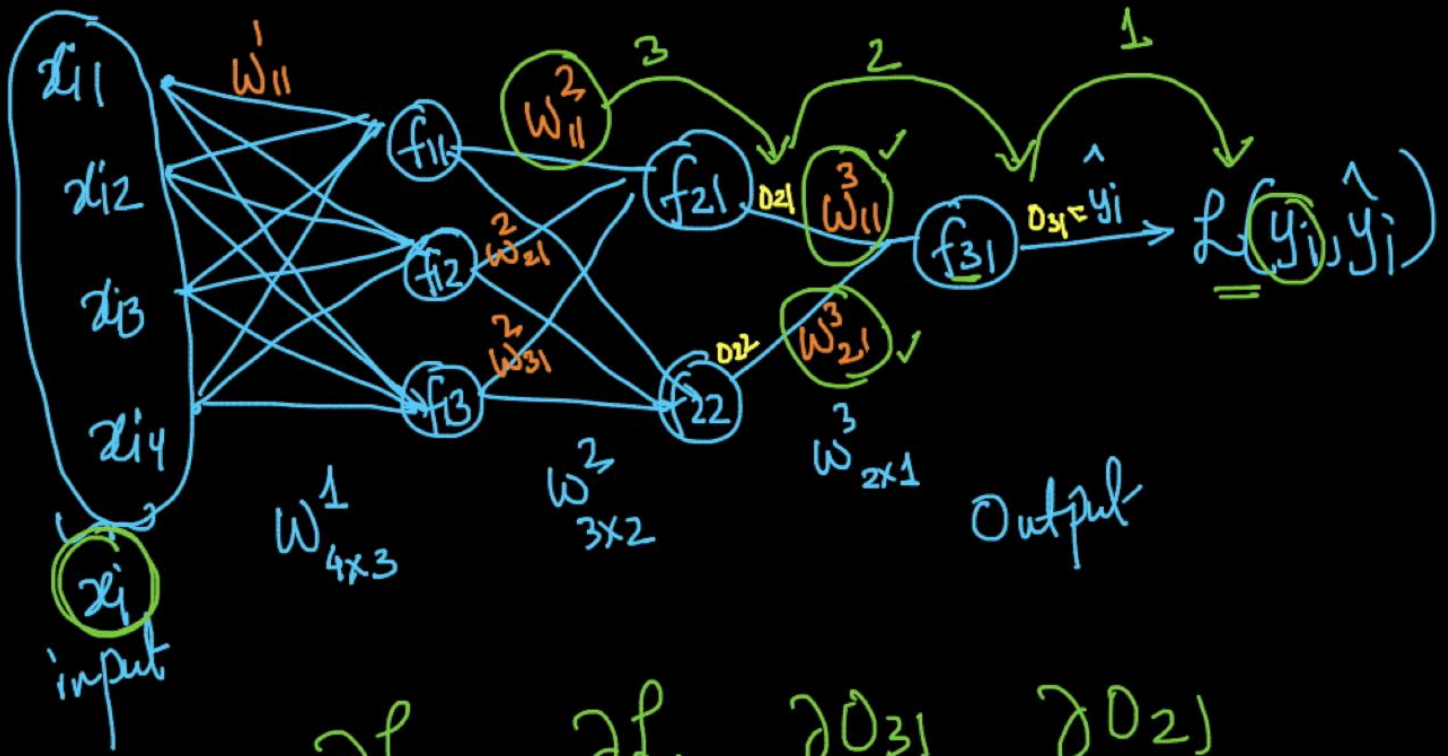

$$\frac{\partial \mathcal{L}}{\partial w^3_{11}} = \frac{\partial \mathcal{L}}{\partial O_{31}} \cdot \frac{\partial O_{31}}{\partial w^3_{11}}$$

$\frac{\partial \mathcal{L}}{\partial O_{31}}$: 1 $\qquad$ $\frac{\partial O_{31}}{\partial w^3_{11}}$: 2

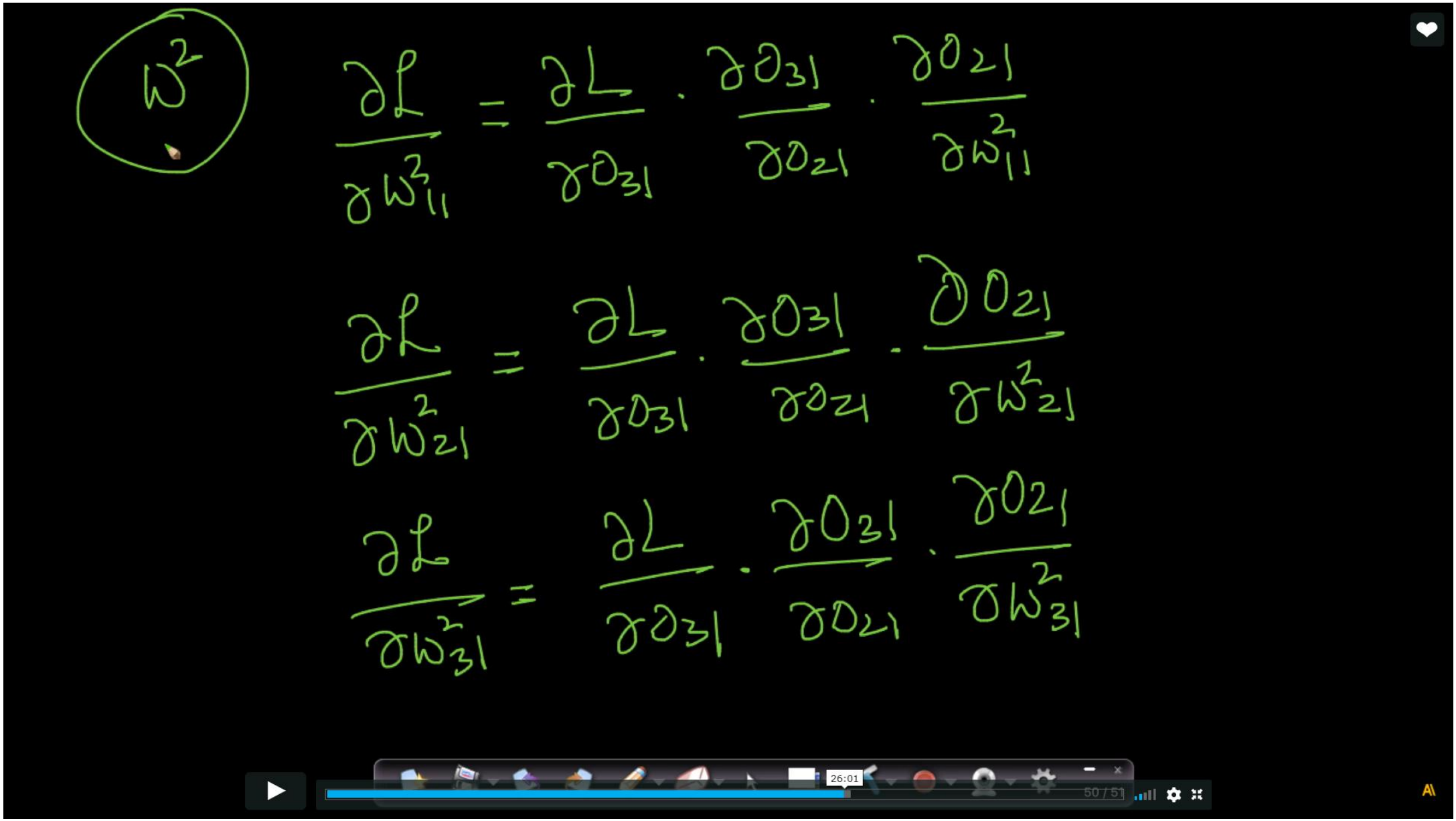
$w^3$

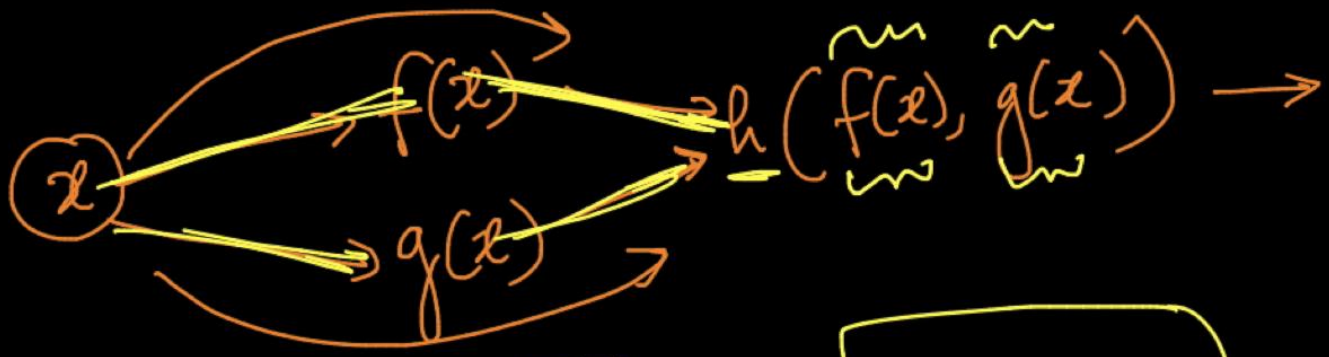
$$\frac{\partial L}{\partial w^3_{11}} = \frac{\partial L}{\partial o_{31}} \cdot \frac{\partial o_{31}}{\partial w^3_{11}}$$ ← chain rule

$$\frac{\partial L}{\partial w^3_{21}} = \frac{\partial L}{\partial o_{31}} \cdot \frac{\partial o_{31}}{\partial w^3_{21}}$$ ← chain rule




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

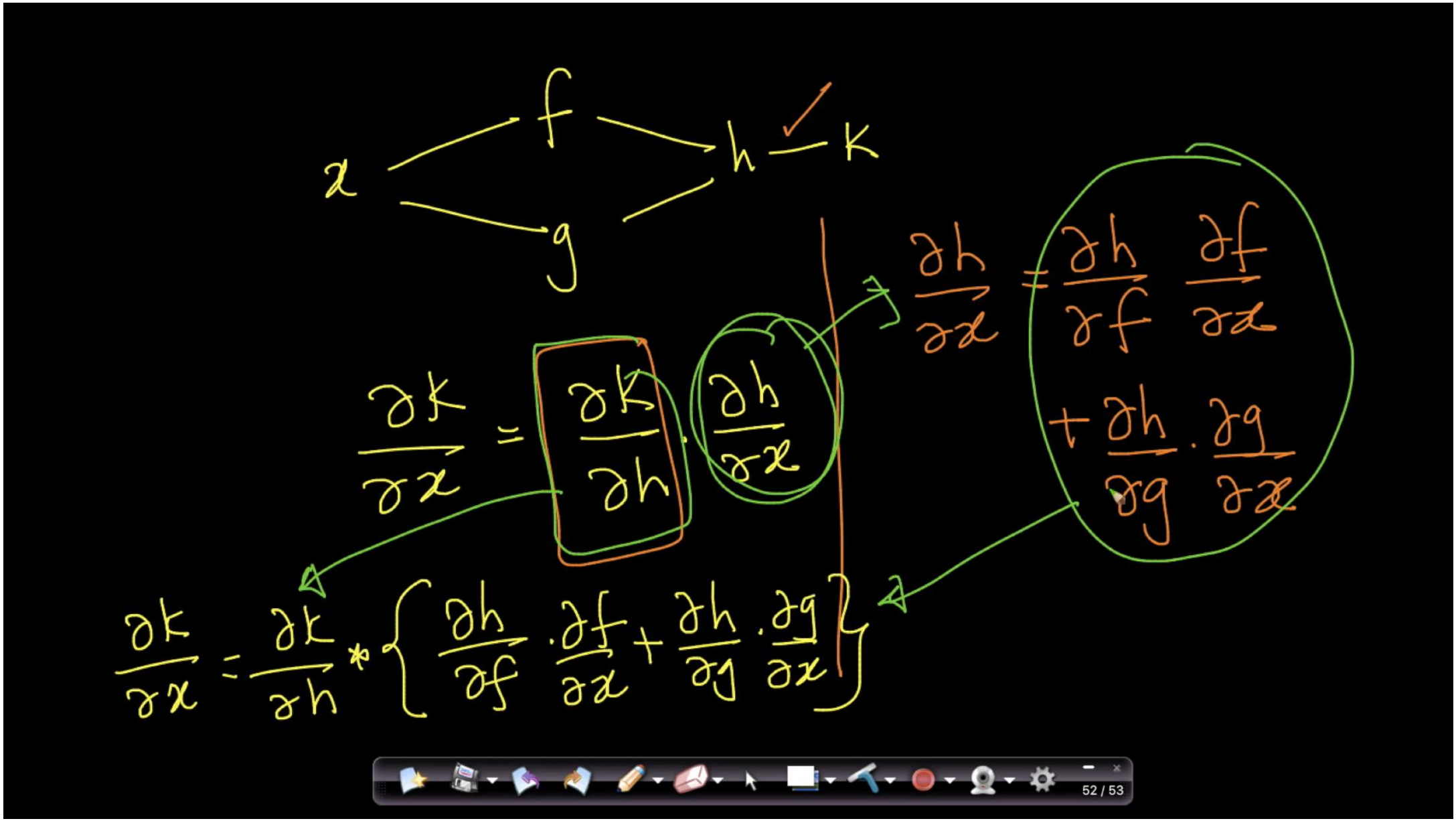

$W^2$
$\frac{\partial L}{\partial w_{11}^2} = \frac{\partial L}{\partial o_{31}} \cdot \frac{\partial o_{31}}{\partial o_{21}} \cdot \frac{\partial o_{21}}{\partial w_{11}^2}$
$\frac{\partial L}{\partial w_{21}^2} = \frac{\partial L}{\partial o_{31}} \cdot \frac{\partial o_{31}}{\partial o_{21}} \cdot \frac{\partial o_{21}}{\partial w_{21}^2}$
$\frac{\partial L}{\partial w_{31}^2} = \frac{\partial L}{\partial o_{31}} \cdot \frac{\partial o_{31}}{\partial o_{21}} \cdot \frac{\partial o_{21}}{\partial w_{31}^2}$
26:01
50 / 51

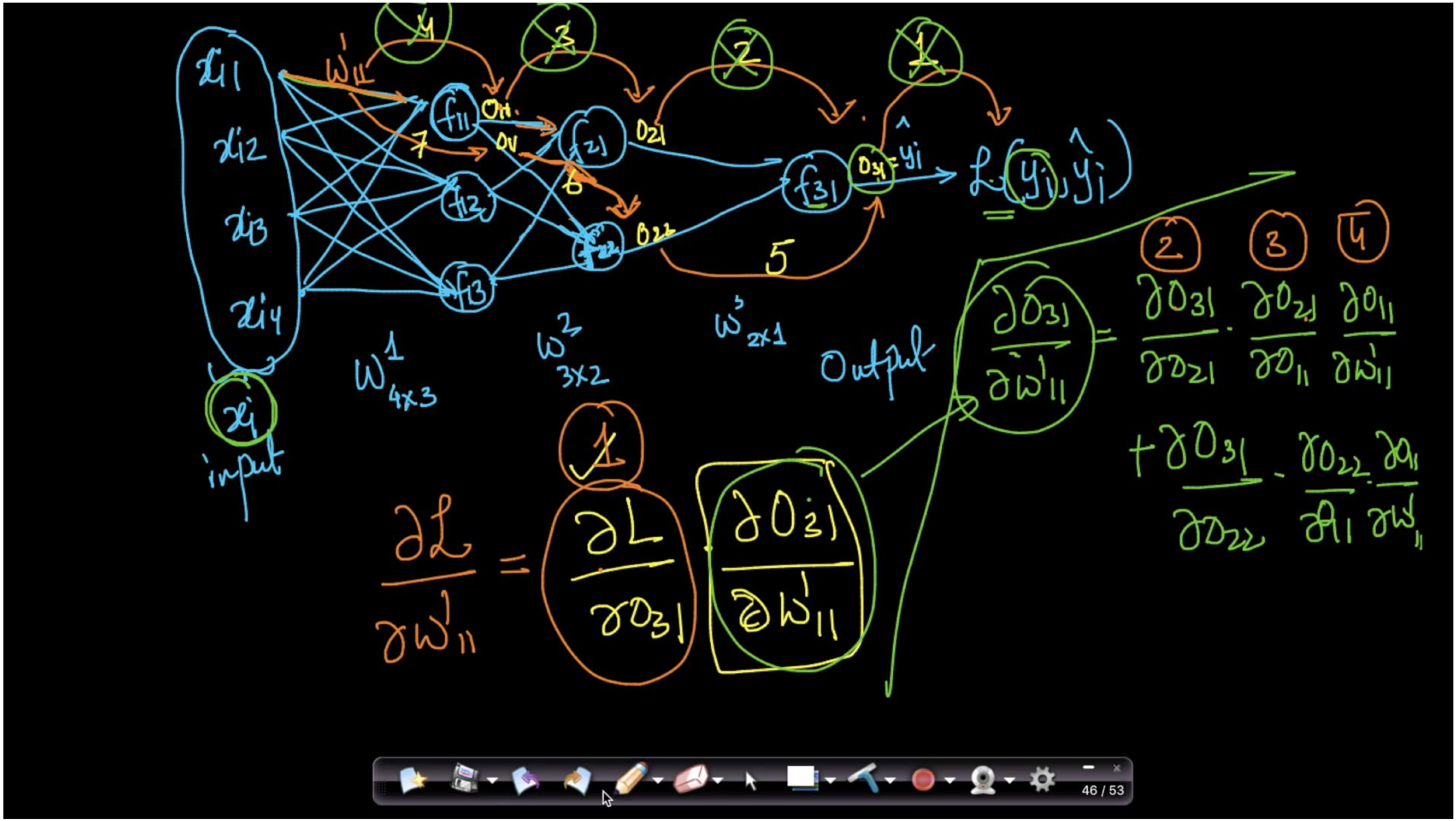



$$\frac{\partial h}{\partial x} = \frac{\partial h}{\partial f}\cdot\frac{\partial f}{\partial x} + \frac{\partial h}{\partial g}\cdot\frac{\partial g}{\partial x}$$

SUM

← chain rule

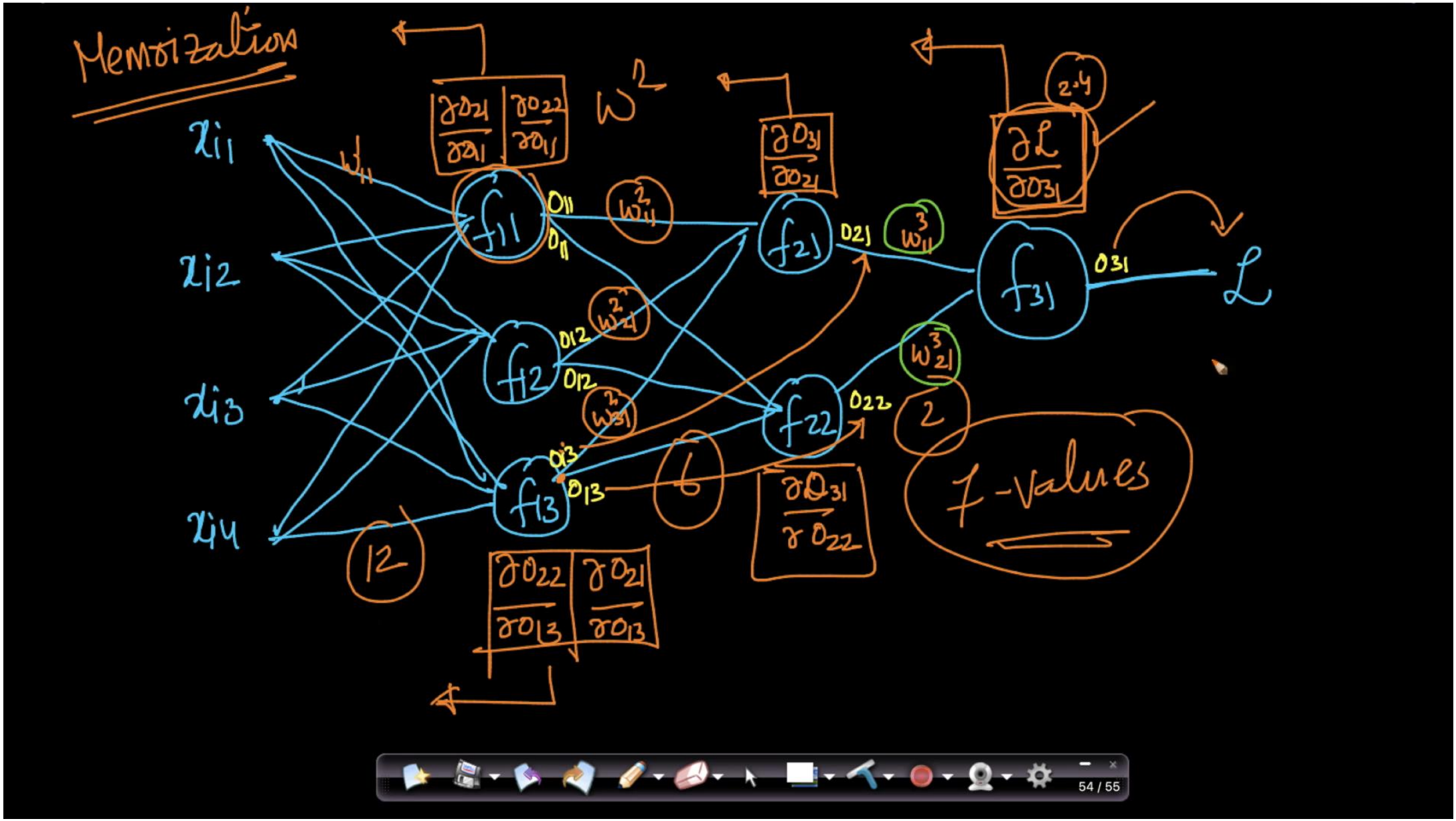

x
f
g
h
k
$\frac{\partial k}{\partial x} = \frac{\partial k}{\partial h} \cdot \frac{\partial h}{\partial x}$
$\frac{\partial h}{\partial x} = \frac{\partial h}{\partial f} \cdot \frac{\partial f}{\partial x} + \frac{\partial h}{\partial g} \cdot \frac{\partial g}{\partial x}$
$\frac{\partial k}{\partial x} = \frac{\partial k}{\partial h} * \left\{ \frac{\partial h}{\partial f} \cdot \frac{\partial f}{\partial x} + \frac{\partial h}{\partial g} \cdot \frac{\partial g}{\partial x} \right\}$
52 / 53


$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{i4}$
$x_i$
input
$W^1_{4\times3}$
$W^2_{3\times2}$
$W^3_{2\times1}$
Output
$f_{11}$
$f_{12}$
$f_{13}$
$f_{21}$
$f_{22}$
$f_{31}$
$O_{11}$
$O_{12}$
$O_{21}$
$O_{22}$
$O_{31}$
$\hat{y}_i$
$L(y_i, \hat{y}_i)$
$\frac{\partial L}{\partial w^1_{11}} = \frac{\partial L}{\partial O_{31}} \frac{\partial O_{31}}{\partial w^1_{11}}$
$\frac{\partial O_{31}}{\partial w^1_{11}} = \frac{\partial O_{31}}{\partial O_{21}} \cdot \frac{\partial O_{21}}{\partial O_{11}} \cdot \frac{\partial O_{11}}{\partial w^1_{11}} + \frac{\partial O_{31}}{\partial O_{22}} \cdot \frac{\partial O_{22}}{\partial O_{11}} \cdot \frac{\partial O_{11}}{\partial w^1_{11}}$
1
2
3
4
5
6
7
46 / 53

$w^1$

$$\partial w^1_{11} = \frac{\partial L}{\partial o_{31}} \cdot \left\{ \frac{\partial o_{31}}{\partial o_{21}} \cdot \frac{\partial o_{21}}{\partial o_{11}} \frac{\partial o_{11}}{\partial w^1_{11}} + \right.$$

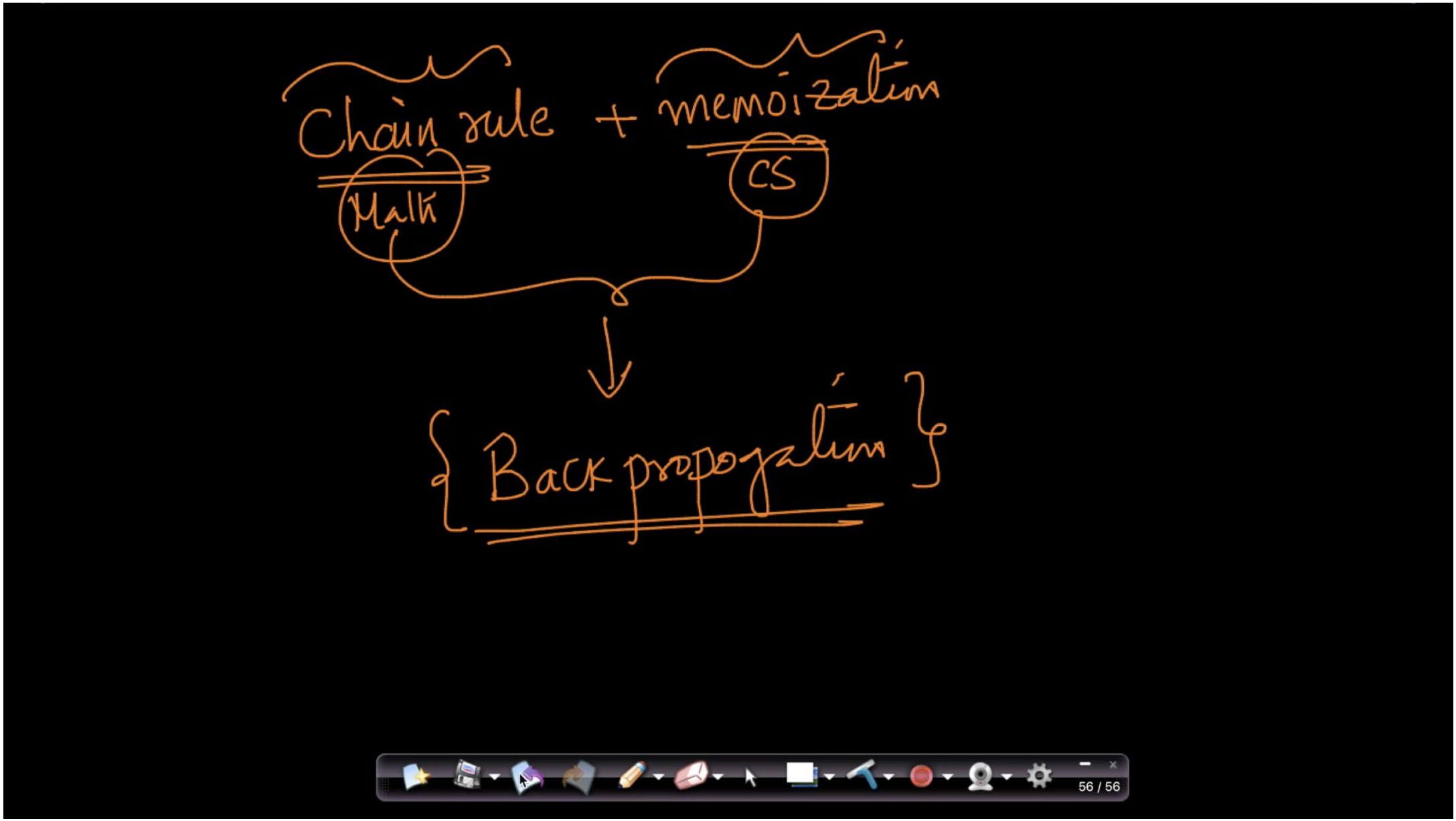
$$\left. \frac{\partial o_{31}}{\partial o_{22}} \cdot \frac{\partial o_{22}}{\partial o_{11}} \cdot \frac{\partial o_{11}}{\partial w^1_{11}} \right\}$$


Memoization
$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{i4}$
$w_{11}$
$f_{11}$
$f_{12}$
$f_{13}$
$f_{21}$
$f_{22}$
$f_{31}$
$O_{11}$
$O_{12}$
$O_{13}$
$O_{21}$
$O_{22}$
$O_{31}$
$w^2$
$w^2_{11}$
$w^2_{21}$
$w^2_{31}$
$w^3_{11}$
$w^3_{21}$
2.4
2
$\mathcal{L}$
f-values
$\frac{\partial O_{31}}{\partial O_{22}}$
$\frac{\partial O_{22}}{\partial O_{13}}$
$\frac{\partial O_{21}}{\partial O_{13}}$
54 / 55

# Memoization

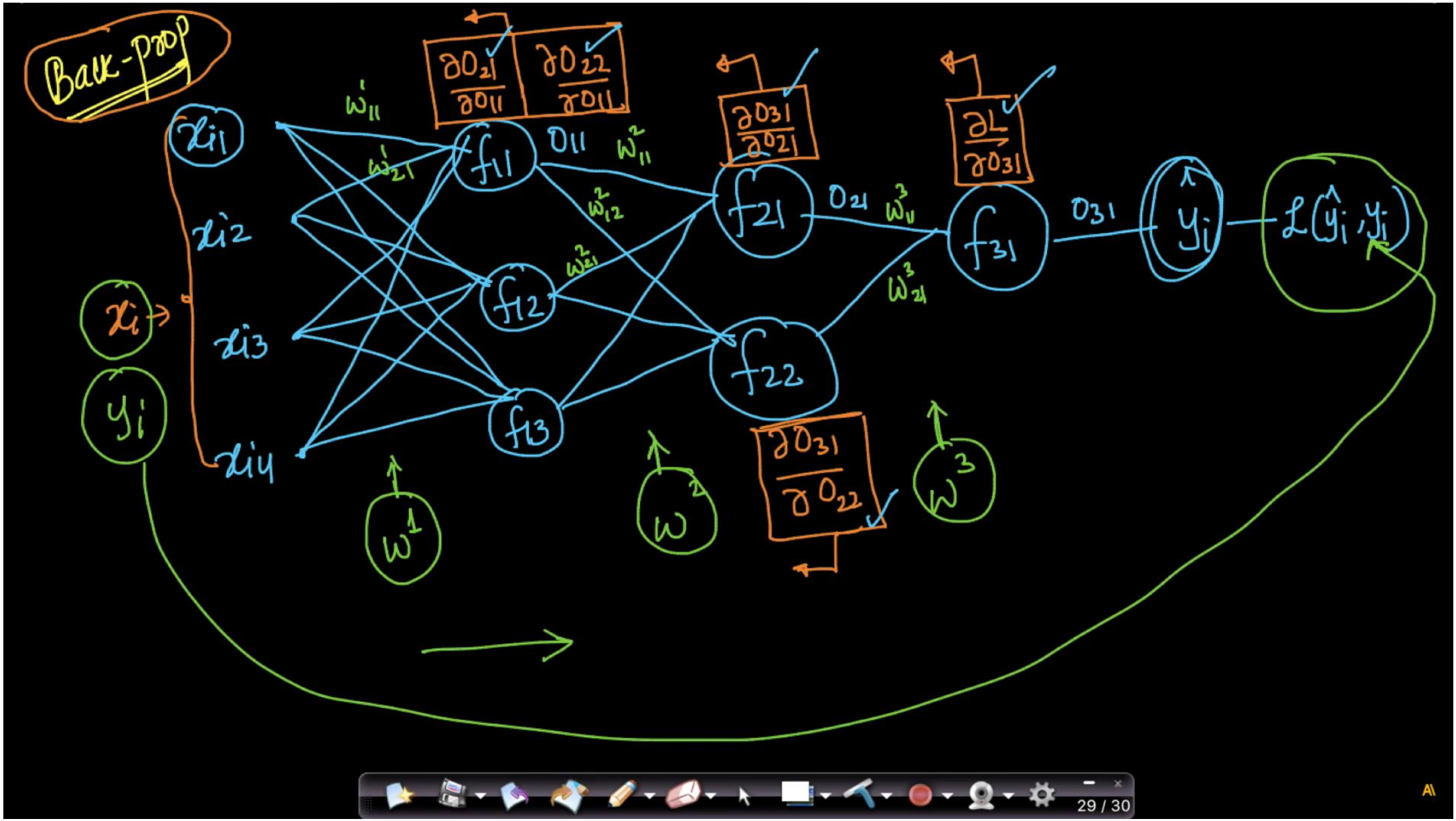
- if there is any operation that is used many times repeatedly,
  - it's a good idea to compute it once ⟶ store it → reuse it

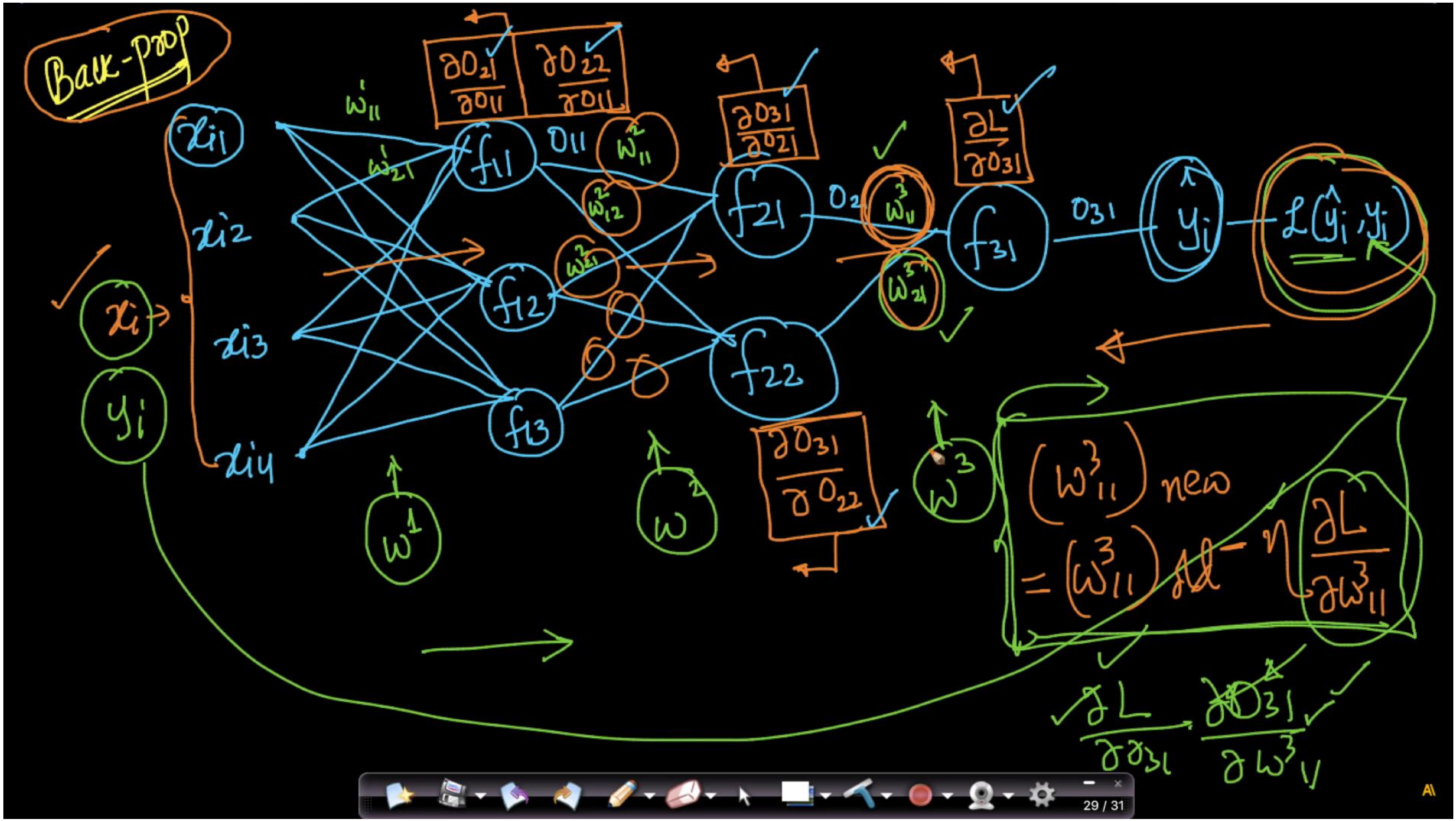
huge Speed-up ; slightly more memory

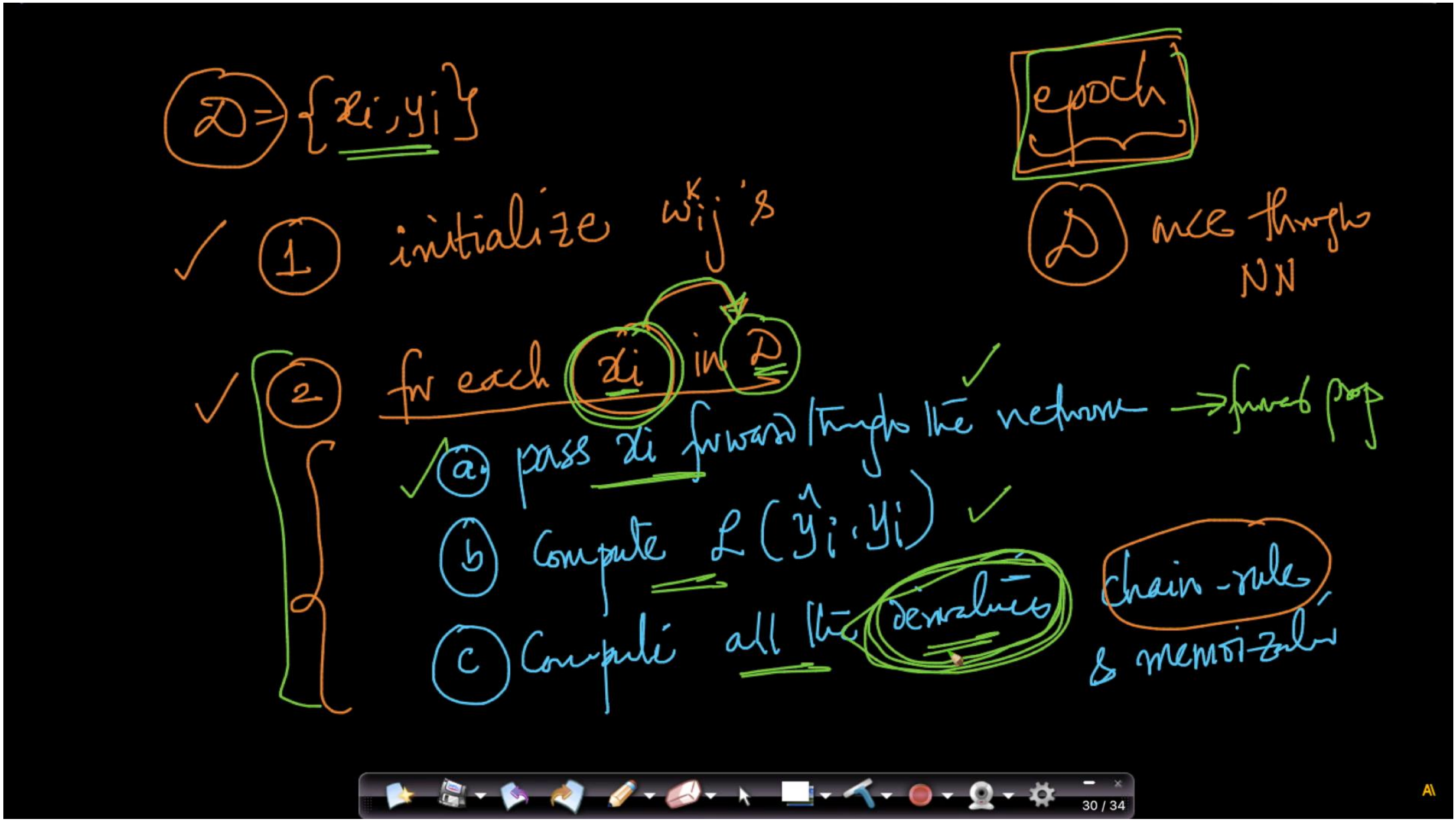

Chain rule + memoization
Math
CS
{Backpropagation}

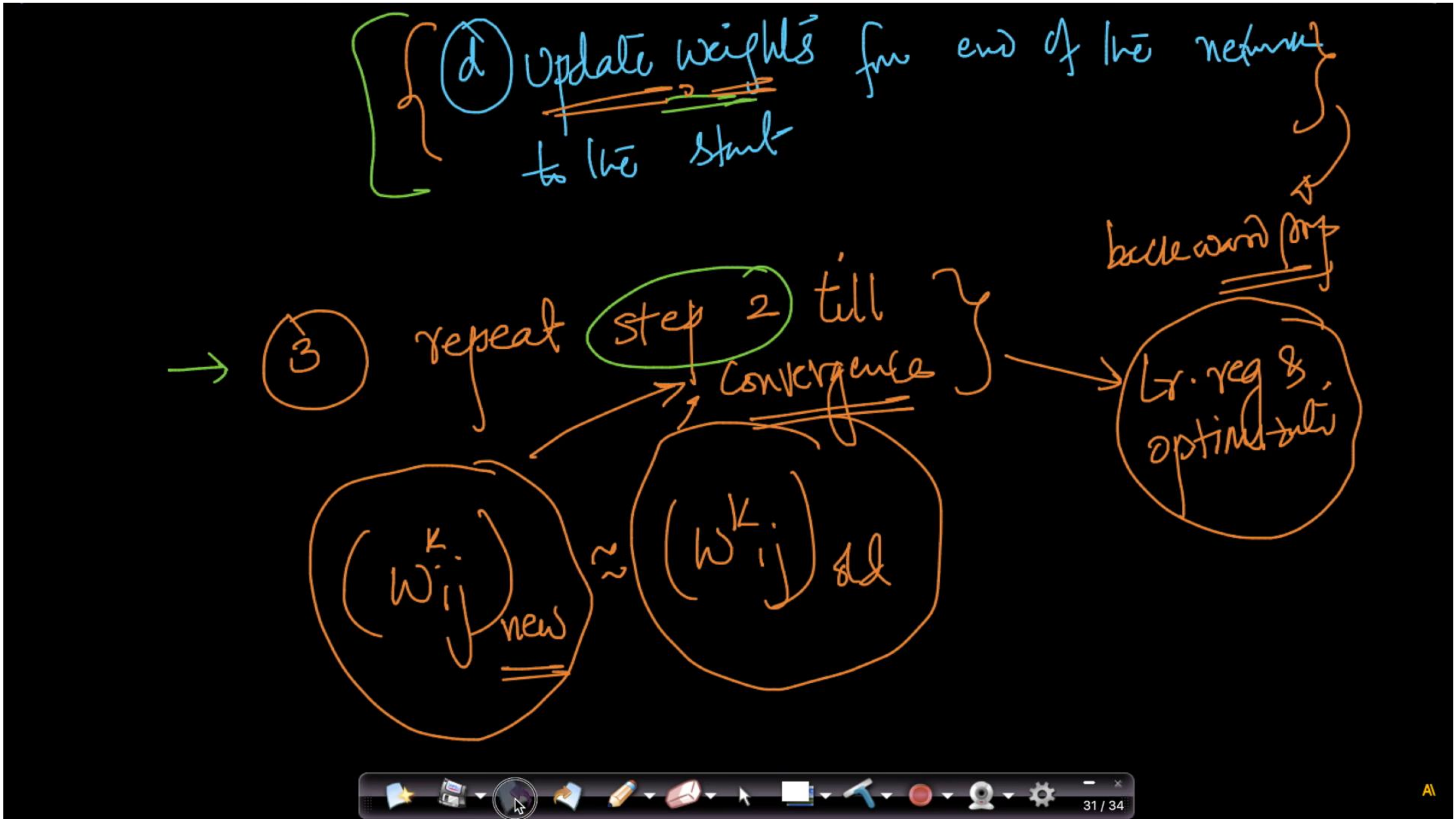
$\mathcal{D} = \{x_i, y_i\}$

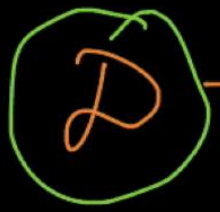
1. initialize $w_{ij}^k$'s
2. for each $x_i$ in $\mathcal{D}$
   - (a) pass $x_i$ forward through the network → forward pop
   - (b) Compute $\mathcal{L}(\hat{y}_i, y_i)$
   - (c) Compute all the derivatives chain-rule & memoization


Back-Prop
$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{i4}$
$x_i$
$y_i$
$w_{11}^1$
$w_{21}^1$
$\frac{\partial o_{21}}{\partial o_{11}}$
$\frac{\partial o_{22}}{\partial o_{11}}$
$f_{11}$
$f_{12}$
$f_{13}$
$o_{11}$
$w_{11}^2$
$w_{12}^2$
$w_{21}^2$
$w^1$
$w^2$
$\frac{\partial o_{31}}{\partial o_{21}}$
$f_{21}$
$f_{22}$
$o_{21}$
$w_{11}^3$
$w_{21}^3$
$\frac{\partial o_{31}}{\partial o_{22}}$
$w^3$
$\frac{\partial L}{\partial o_{31}}$
$f_{31}$
$o_{31}$
$\hat{y}_i$
$\mathcal{L}(\hat{y}_i, y_i)$


Back-Prop
$x_{i1}$
$x_{i2}$
$x_{i3}$
$x_{i4}$
$x_i$
$y_i$
$w^1_{11}$
$w^1_{21}$
$f_{11}$
$f_{12}$
$f_{13}$
$o_{11}$
$w^2_{11}$
$w^2_{12}$
$w^2_{21}$
$f_{21}$
$f_{22}$
$w^2$
$o_2$
$w^3_{11}$
$w^3_{21}$
$f_{31}$
$o_{31}$
$\hat{y}_i$
$\mathcal{L}(\hat{y}_i, y_i)$
$w^3$
$\frac{\partial o_{21}}{\partial o_{11}}$
$\frac{\partial o_{22}}{\partial o_{11}}$
$\frac{\partial o_{31}}{\partial o_{21}}$
$\frac{\partial o_{31}}{\partial o_{22}}$
$\frac{\partial \mathcal{L}}{\partial o_{31}}$
$(w^3_{11})_{new} = (w^3_{11})_{old} - \eta \frac{\partial \mathcal{L}}{\partial w^3_{11}}$
$\frac{\partial \mathcal{L}}{\partial o_{31}} \cdot \frac{\partial o_{31}}{\partial w^3_{11}}$
29 / 31


D = {xi, yi}
epoch
1 initialize $w^k_{ij}$'s
D once through NN
2 for each xi in D
a pass xi forward through the network → forward prop
b Compute $\mathcal{L}(\hat{y}_i, y_i)$
c Compute all the derivatives
chain-rule & memoization

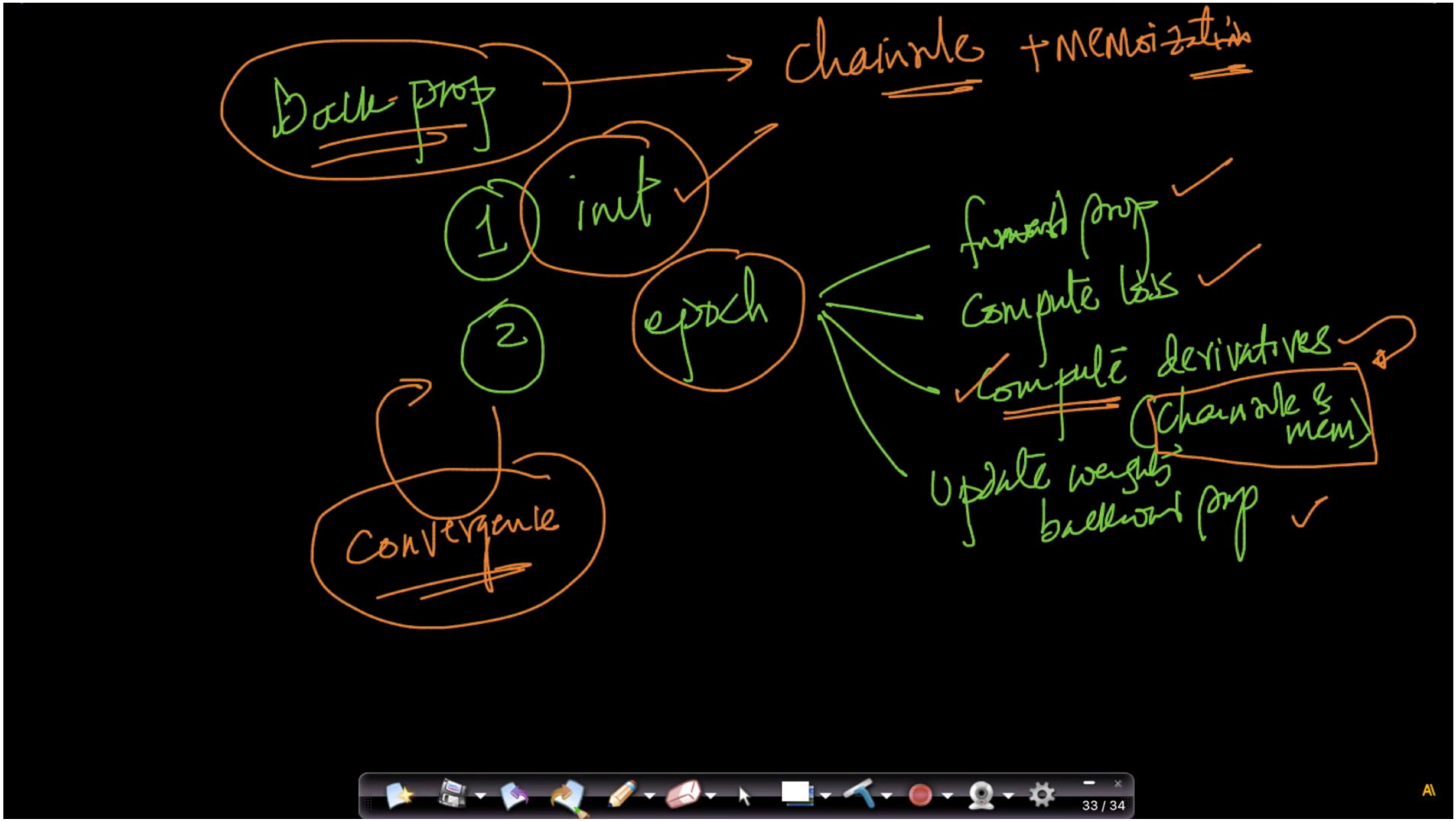

(4) Update weights from end of the network to the start
backward pass
3
repeat step 2 till convergence
Lr. reg & optimization
$(w_{ij}^{k})_{new} \simeq (w_{ij}^{k})_{old}$

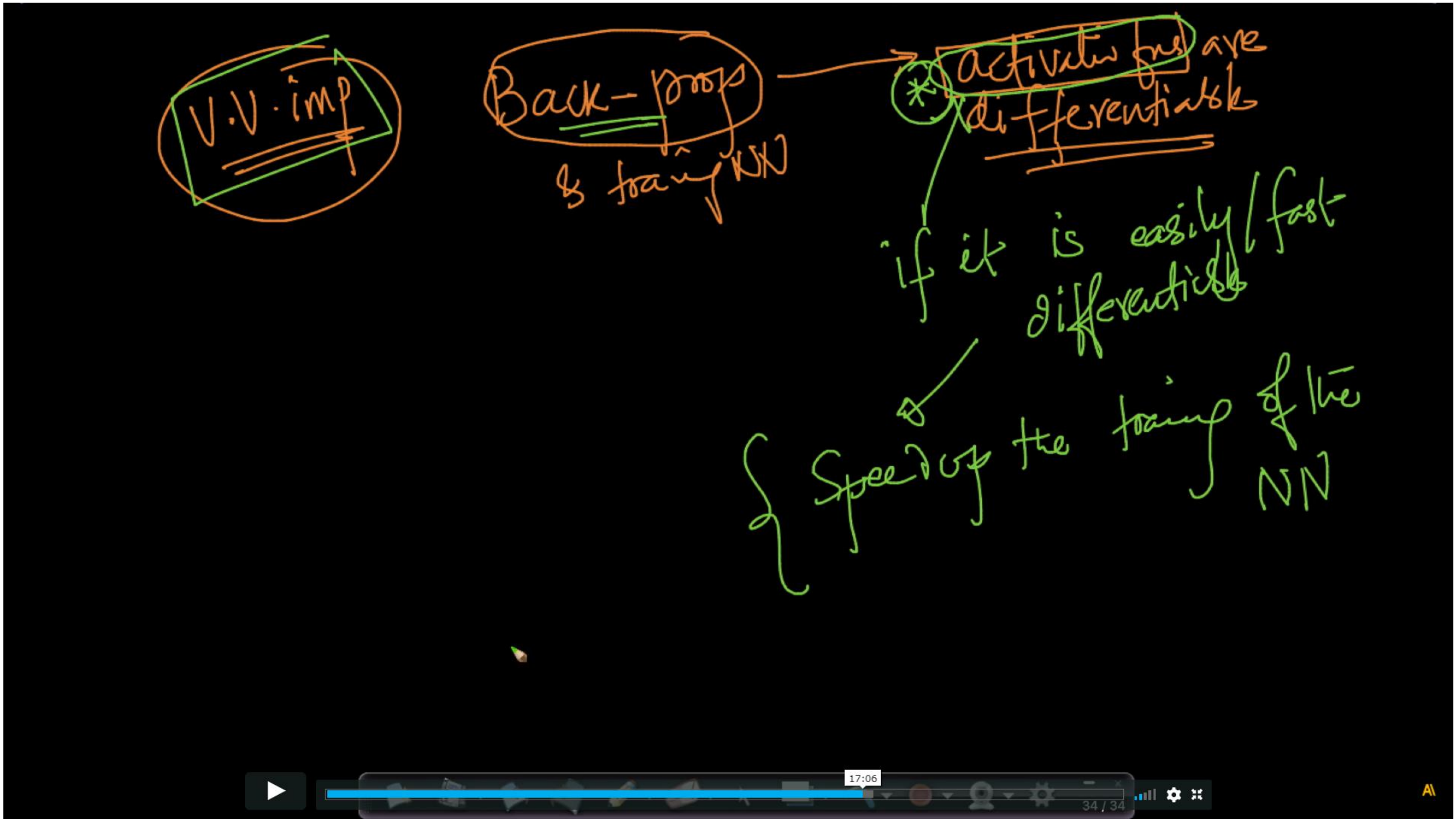


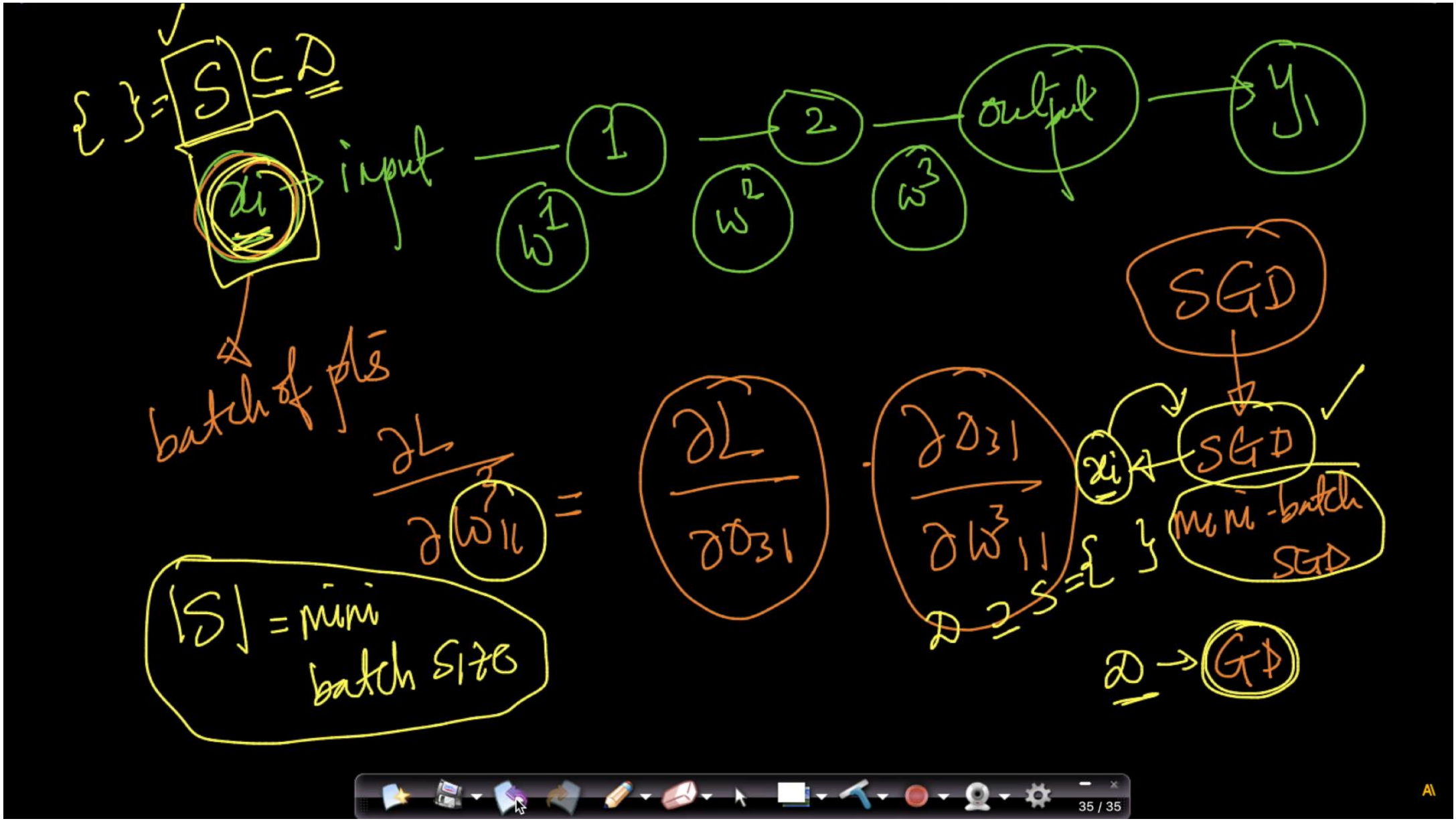

$\rightarrow \{x_i, y_i\} \rightarrow$ epochs

5 times $\rightarrow$ 5 epochs

multiple epochs


Batch prop
chainrule + MEMoization
1
init
2
epoch
forward prop
Compute loss
Compute derivatives
(chainrule & mem)
Update weights
backward pop
convergence

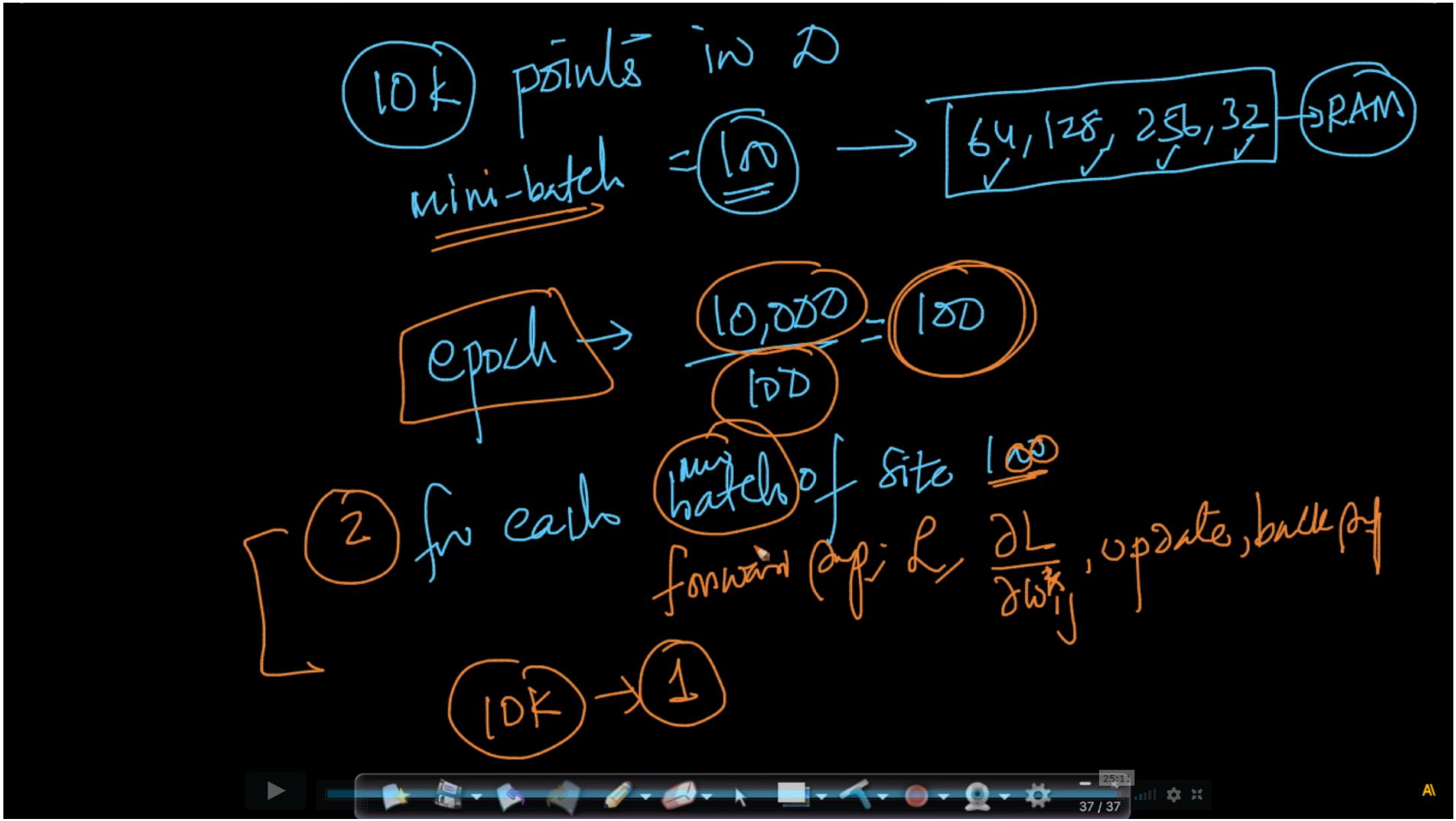

V.V.imp
Back-prop
& training NN
activation fns are
differentiable
if it is easily / fast
differentiable
Speed up the training of the
NN
17:06
34 / 34


S ⊆ D
input
1
2
output
$y_1$
$x_i$
batch of pts
$w^1$
$w^2$
$w^3$
SGD
$\frac{\partial L}{\partial w_{11}^3} = \frac{\partial L}{\partial o_{31}} \cdot \frac{\partial o_{31}}{\partial w_{11}^3}$
$x_i$
SGD
mini-batch SGD
$\{\ \}$
$S \subseteq D$
$|S|$ = mini batch size
$D \rightarrow$ GD
35 / 35

{ all of the data points in RAM is computing 'd' using D

mini-batch based batch-gd → most popular approach

$S = \{\ \} \subseteq D$


10k points in 2D
mini-batch = 100
64, 128, 256, 32
RAM
epoch
10,000
100
100
2 for each mini batch of size 100
forward pass & $\frac{\partial L}{\partial w_{ij}^k}$, update, back pass
10K
1

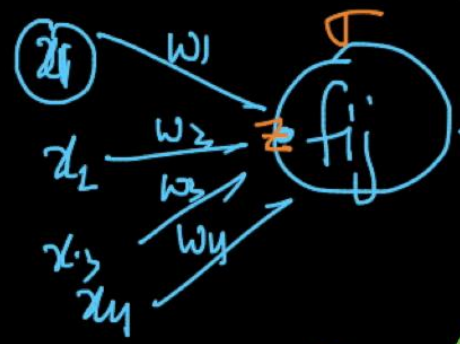
# Activation functions

$f_{ij}$

classical NN → 2006, 2012 → DL

1980s & 90s → Sigmoid & tanh

✓ differentiable
✓ easy to diff




$$z = \sum_i w_i x_i = w^T x$$

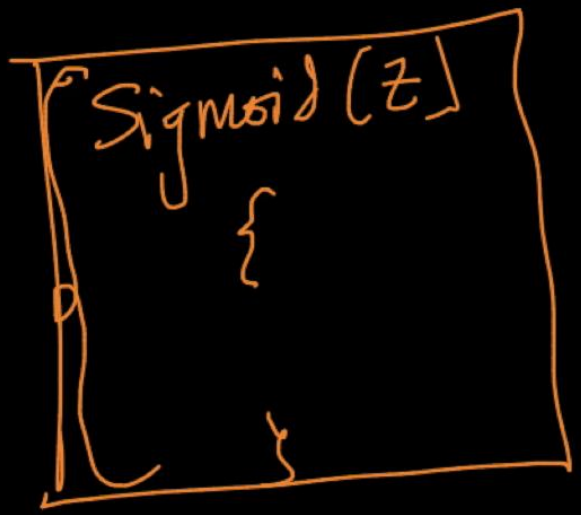
$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\sigma(z) = \frac{e^z}{1+e^z} \text{ or } \frac{1}{1+e^{-z}}$$

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

$$\checkmark \quad \frac{\partial \sigma}{\partial z} = \sigma(z)\left(1-\sigma(z)\right)$$




→ forward prop

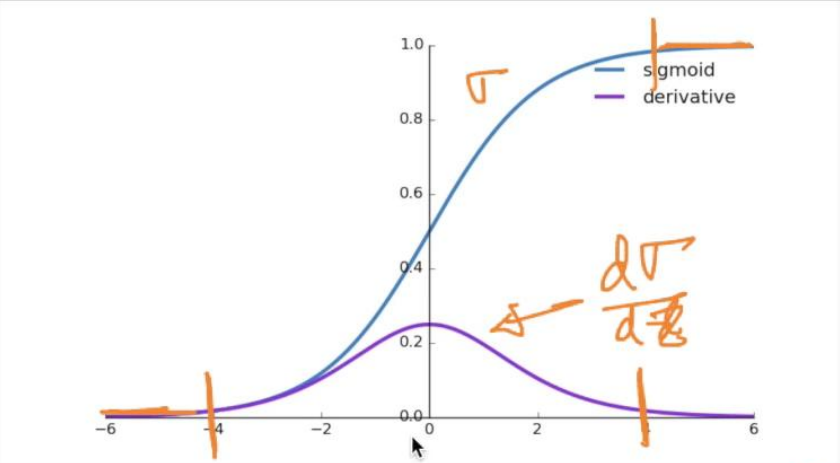
→ computing derivatives (backprop)

Contents - Google Docs | calculus - Derivative of sigmoi | Ronny Restrepo | Ronny Restrepo | Chekuri Srikan...

ronny.rest/blog/post_2017_08_10_sigmoid/

Ronny Restrepo

Portfolio Blog Tutorials Contact




## Comments

Note you can comment without any login by:

1. Typing your comment
2. Selecting "sign up with Disqus"
3. Then checking "I'd rather post as a guest"

3 Comments ronny.rest

Login

Recommend 11 Share

Sort by Best

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

556 / 55

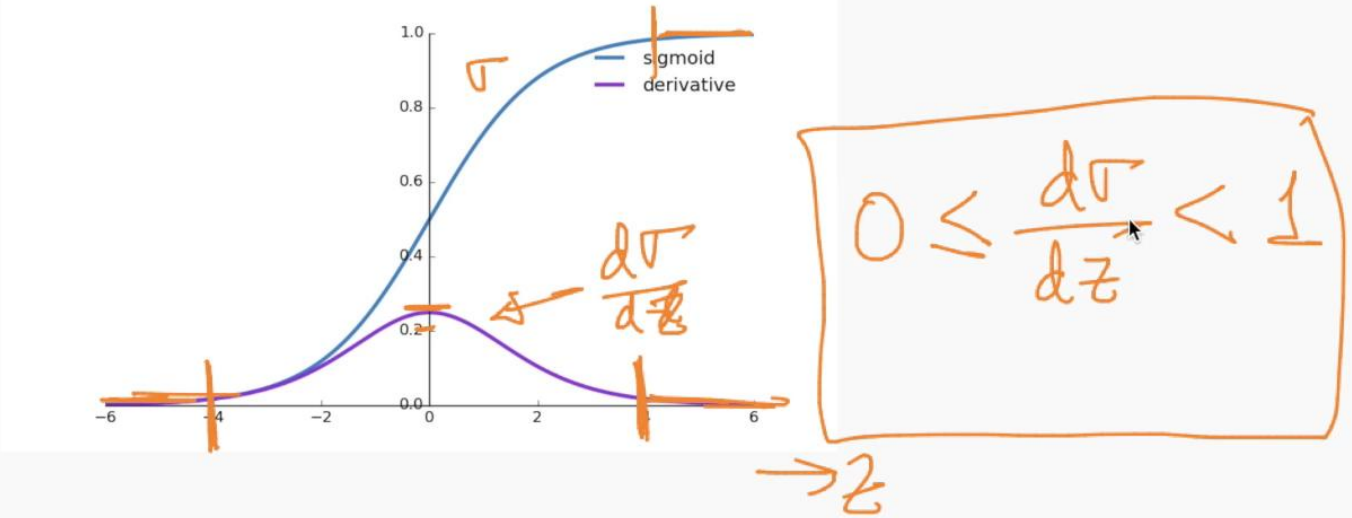
APPLIED COURSE AI

Contents - Google Docs | calculus - Derivative of sigmoi | Ronny Restrepo | Ronny Restrepo | Chekuri Srikan...

ronny.rest/blog/post_2017_08_10_sigmoid/

Ronny Restrepo

Portfolio Blog Tutorials Contact




## Comments

Note you can comment without any login by:

1. Typing your comment
2. Selecting "sign up with Disqus"
3. Then checking "I'd rather post as a guest"

3 Comments ronny.rest

Login

Recommend 11 Share

Sort by Best

Join the discussion...

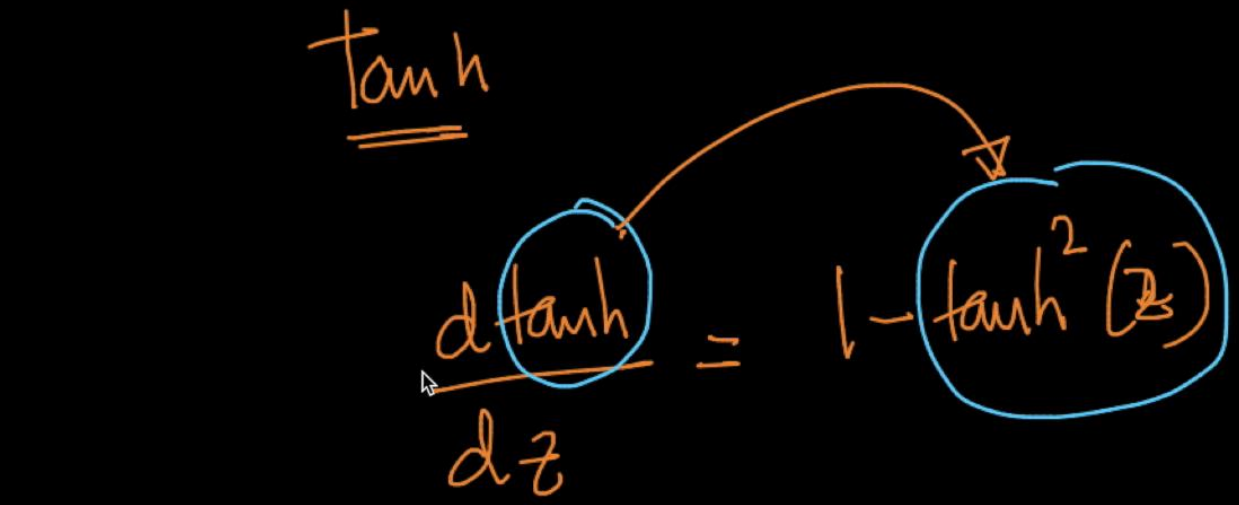
LOG IN WITH

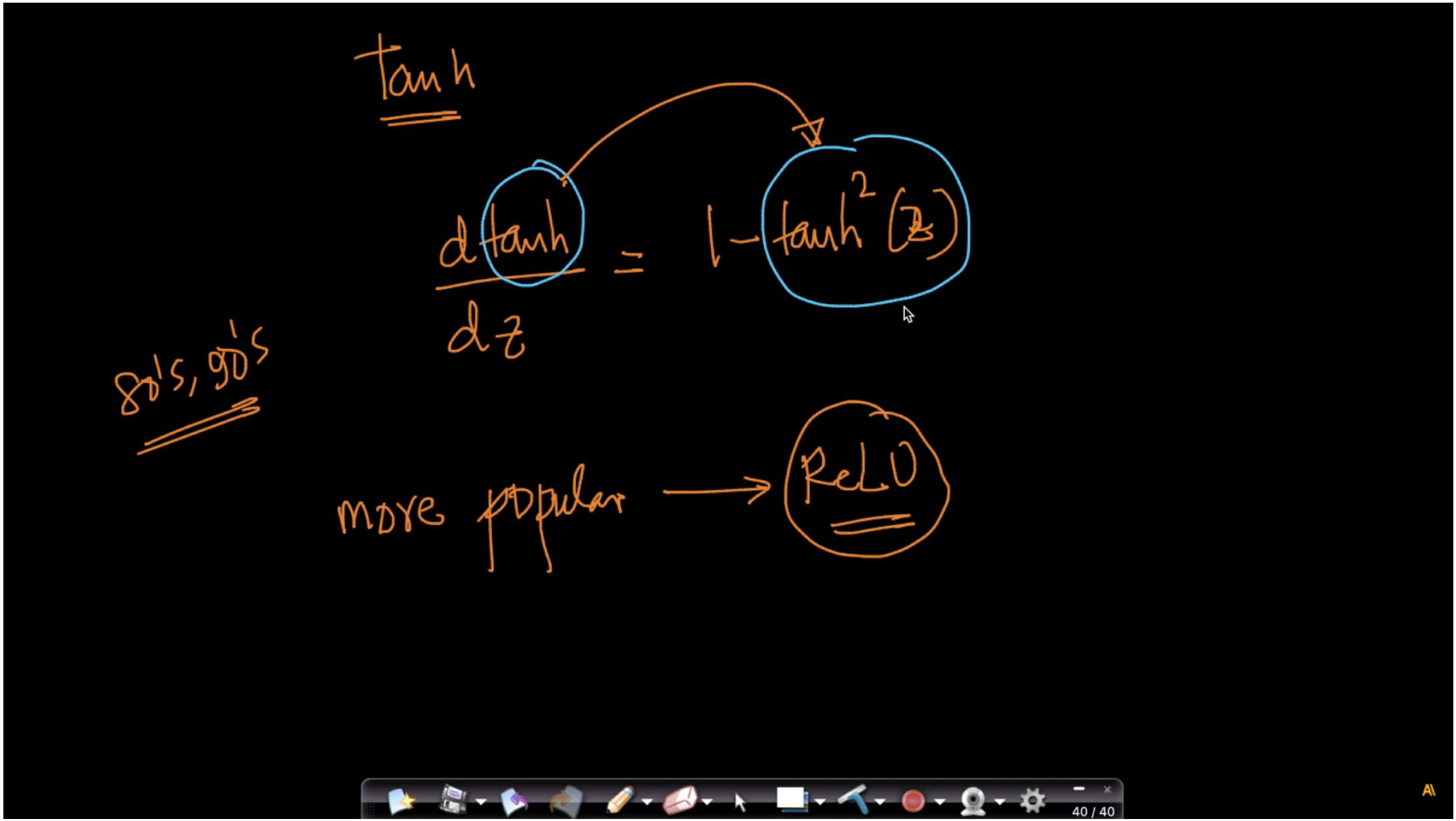
OR SIGN UP WITH DISQUS

Name

556 / 55

APPLIED COURSE AI


tanh
d tanh / dz = 1 − tanh²(z)


tanh
$\frac{d\tanh}{dz} = 1 - \tanh^2(z)$
80's, 90's
more popular
ReLU

Contents - Google Docs | calculus - Derivative of sigmoi | Ronny Restrepo | Ronny Restrepo | Chekuri Srikan...

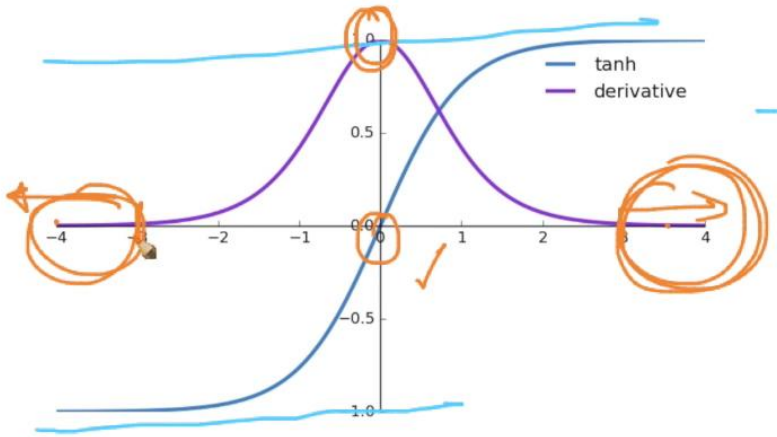
ronny.rest/blog/post_2017_08_16_tanh/

Ronny Restrepo

Portfolio Blog Tutorials Contact

```
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')

# Create and show plot
ax.plot(z,a, color="#307EC7", linewidth=3, label="tanh")
ax.plot(z,dz, color="#9621E2", linewidth=3, label="derivative")
ax.legend(loc="upper right", frameon=False)
fig.show()
```




optimization

1. -1 to 1

2. $0 \leq \frac{d\tanh}{dz} \leq 1$

## Comments

Note you can comment without any login by:

1. Typing your comment
2. Selecting "sign up with Disqus"
3. Then checking "I'd rather post as a guest"

0 Comments ronny.rest

13:43

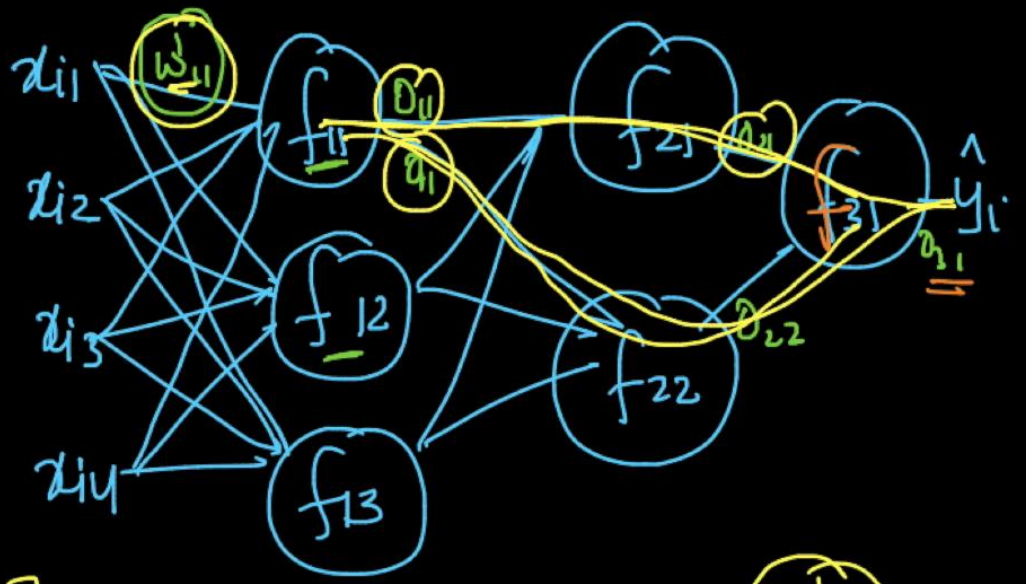
APPLIED COURSE

# {Vanishing gradients}

$o_{31} = f_{31}(\,)$

→ { $w$'s, $w^1$, $o^1$'s → NN

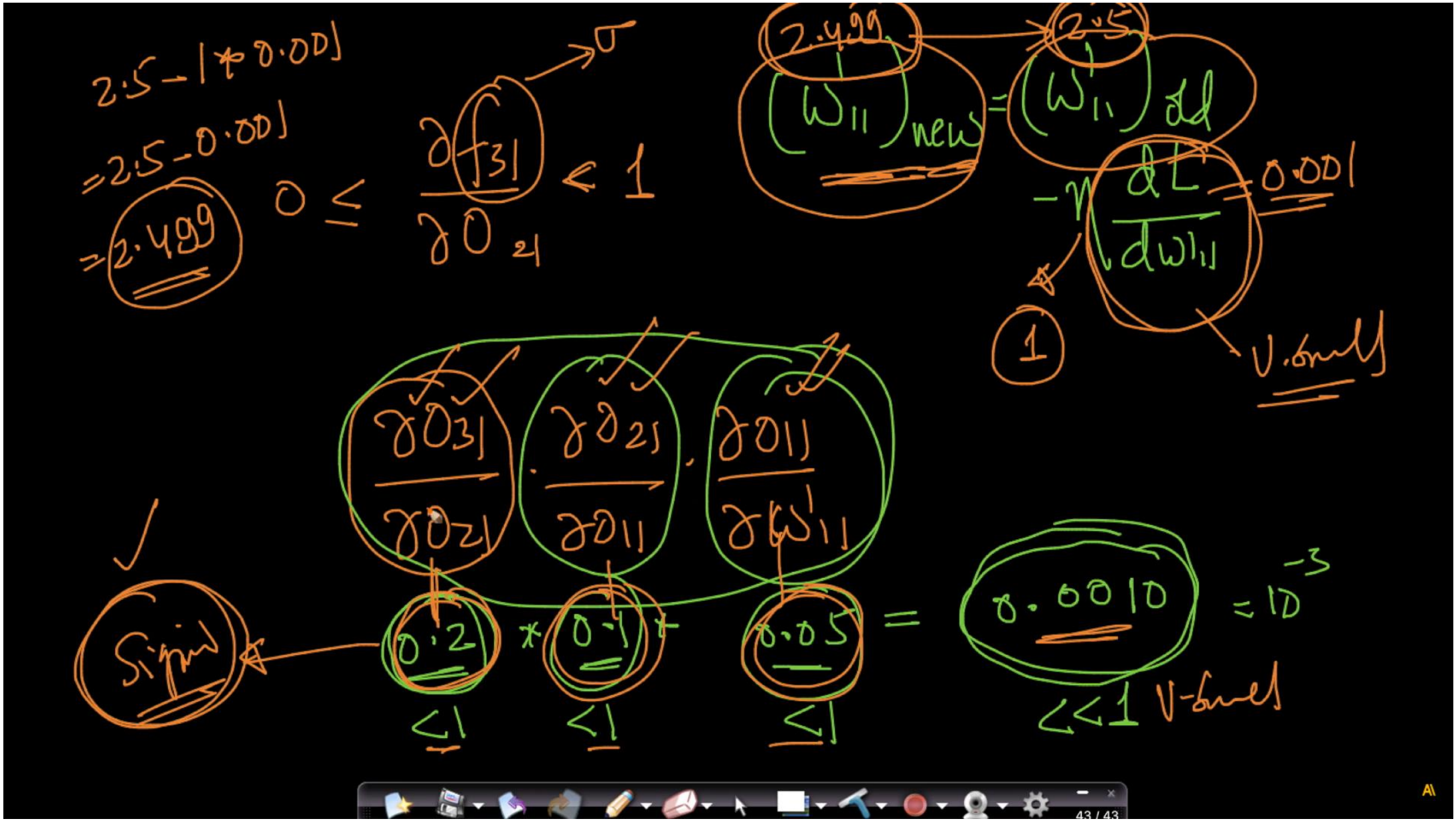
→ $f_{ij}$ → Sigmoid

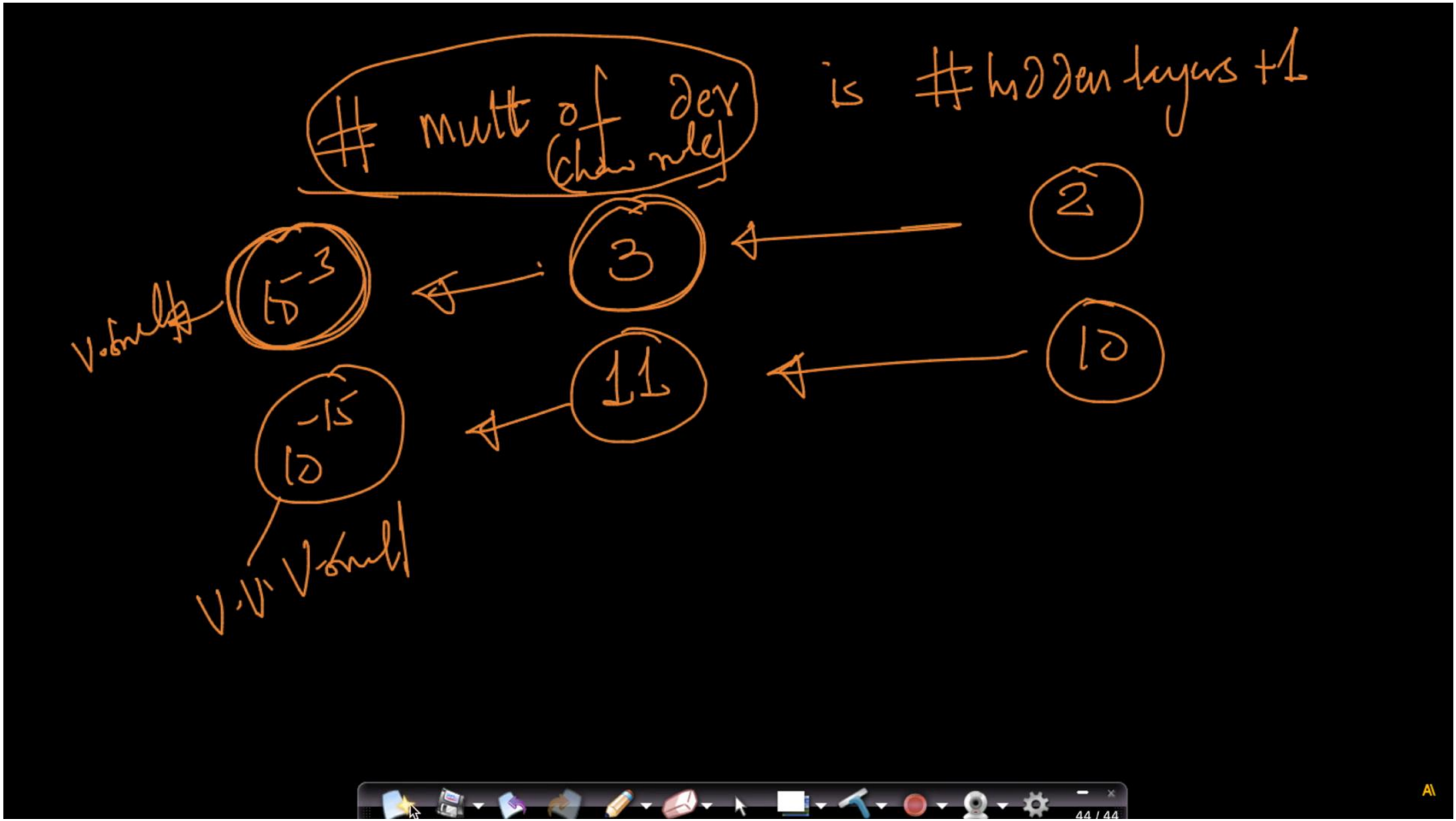


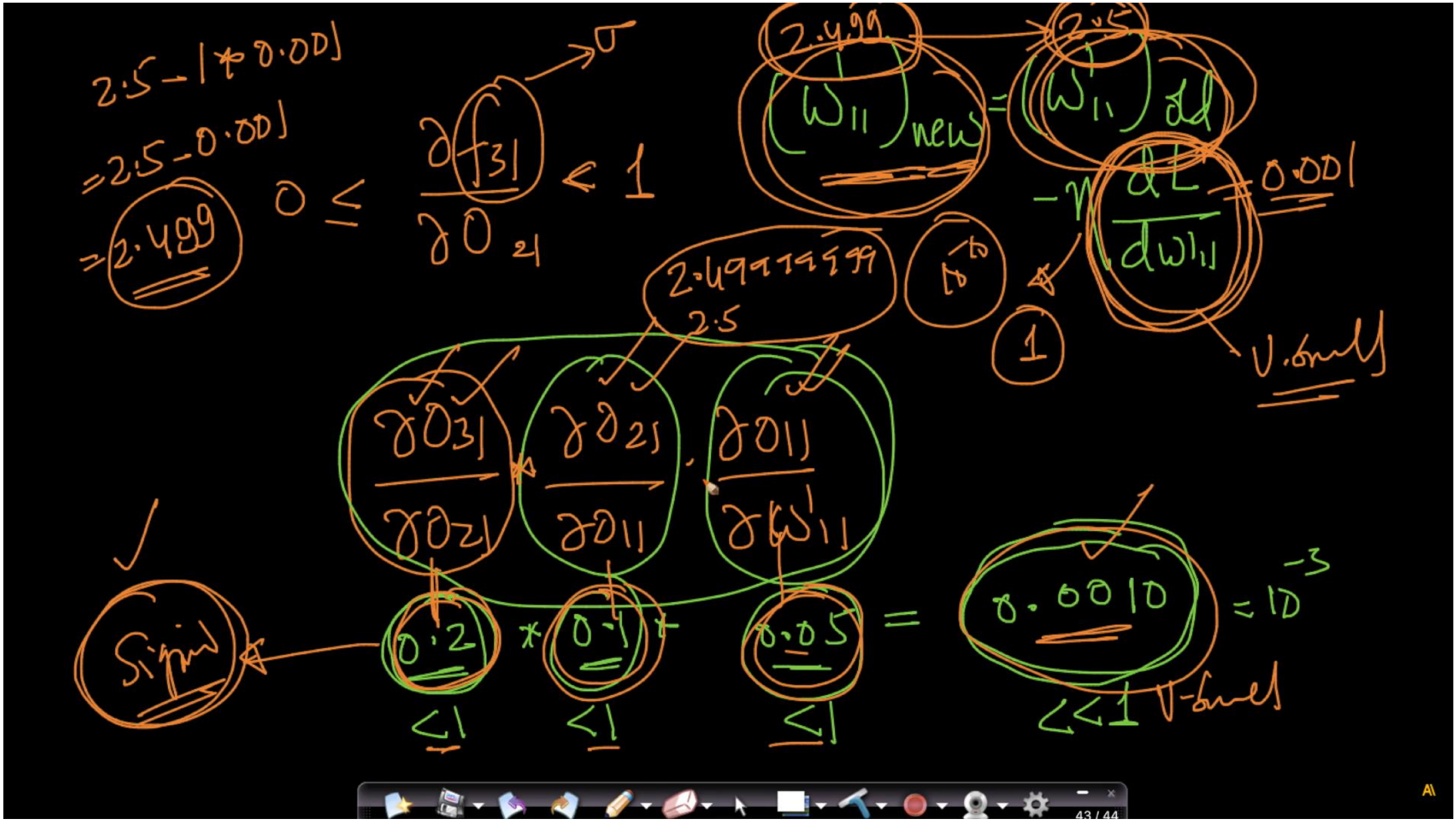

$$\frac{\partial L}{\partial w^1_{11}} = \frac{\partial L}{\partial o_{31}} \left[ \frac{\partial o_{31}}{\partial o_{21}} \cdot \frac{\partial o_{21}}{\partial o_{11}} \cdot \frac{\partial o_{11}}{\partial w^1_{11}} + \frac{\partial o_{31}}{\partial o_{22}} \cdot \frac{\partial o_{22}}{\partial o_{11}} \cdot \frac{\partial o_{11}}{\partial w^1_{11}} \right]$$

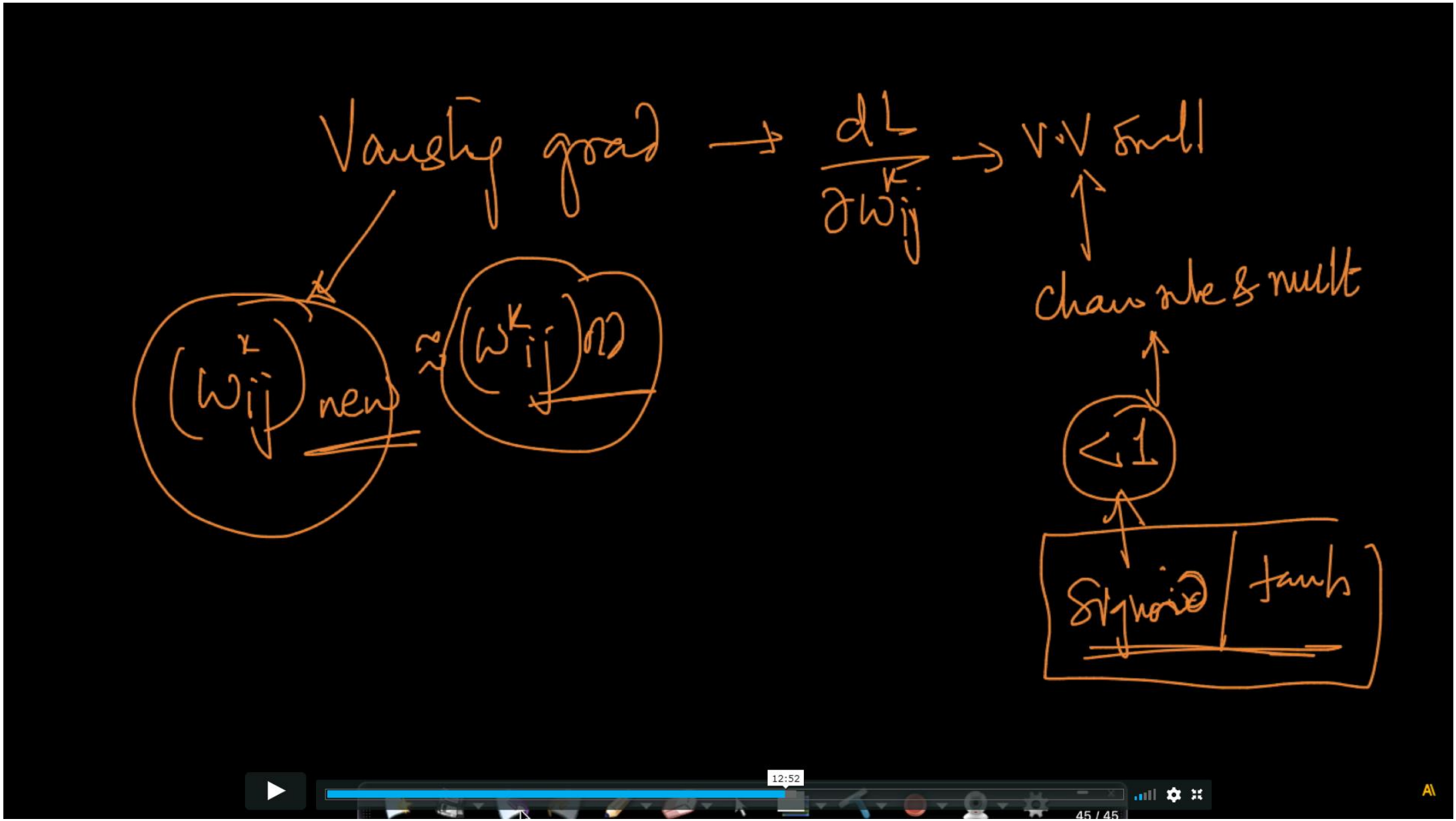
$$\left(w^1_{11}\right)_{new} = \left(w^1_{11}\right)_{old} - \eta \frac{\partial L}{\partial w^1_{11}}$$

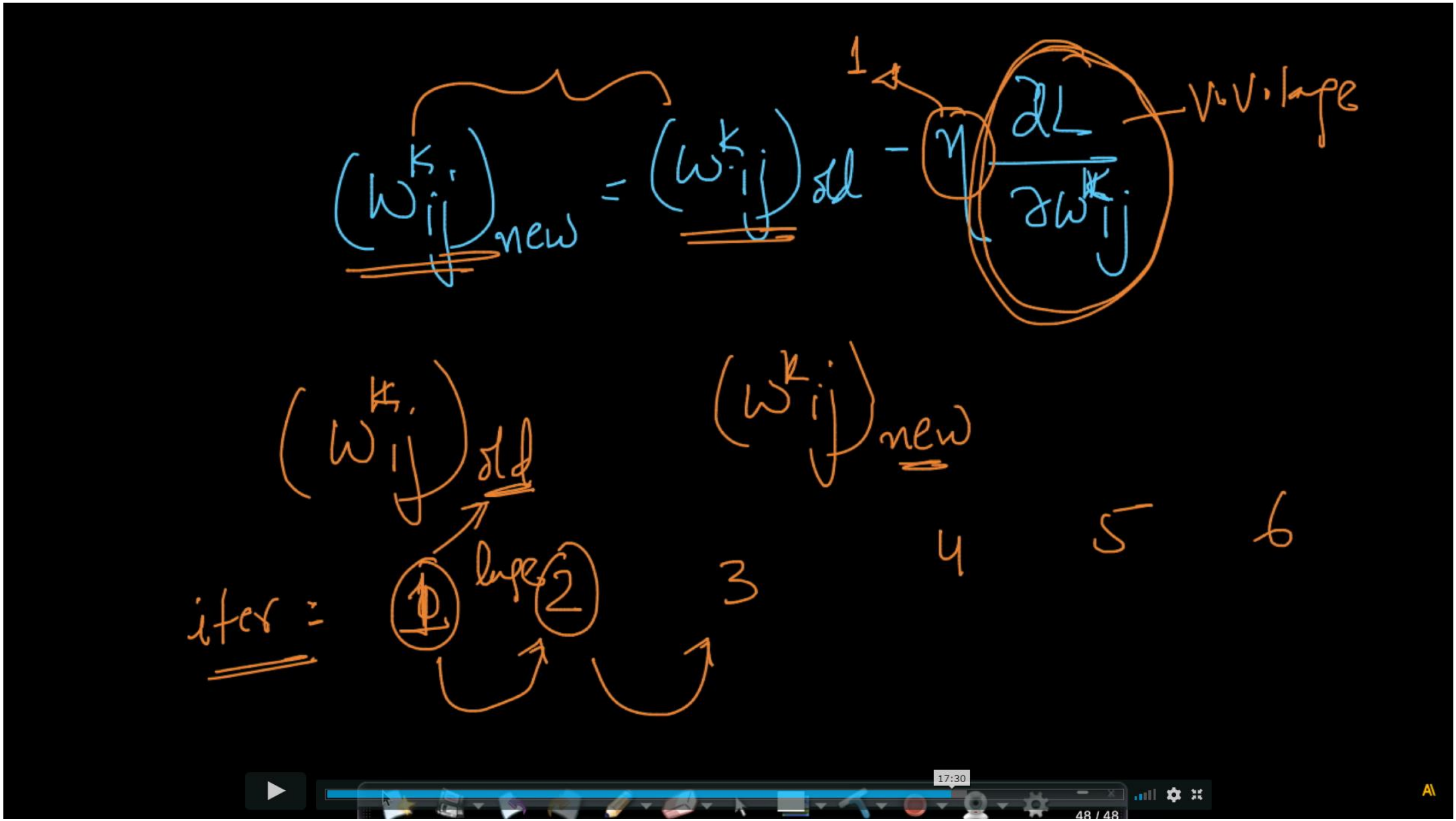
$$\frac{\partial o_{31}}{\partial o_{21}} = \frac{\partial f_{31}}{\partial o_{21}}$$

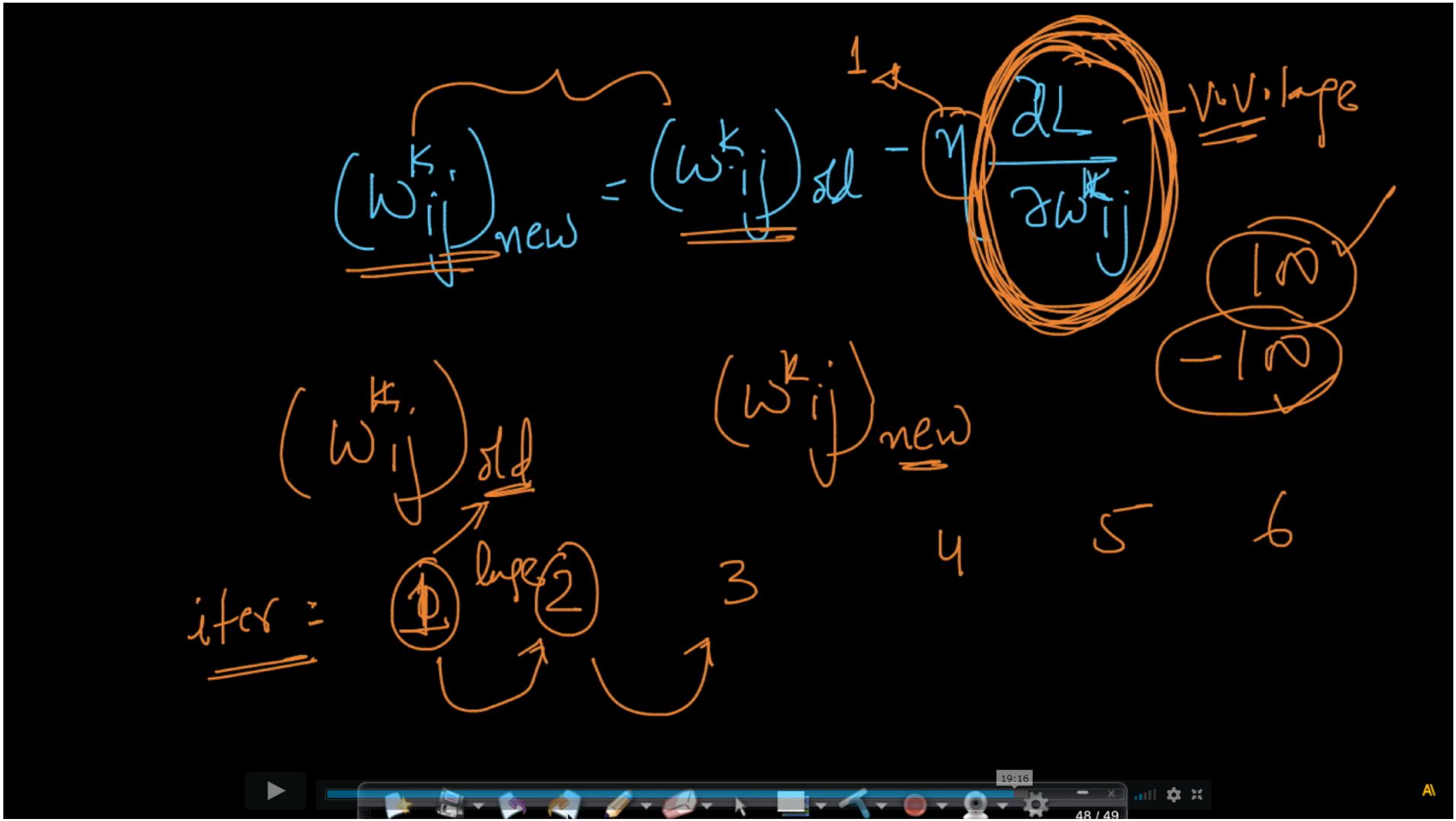

2.5 - 1 * 0.001
= 2.5 - 0.001
= 2.499
0 ≤ ∂f31 / ∂O21 ≤ 1
(w11)new = (w11)old - η ∂L/∂w11
0.001
1
V.small
∂O31/∂O21 . ∂O21/∂O11 . ∂O11/∂(w1)11
Signal
0.2 * 0.4 + 0.05 = 0.00010 = 10^-3
<1
<1
<1
<<1 V-small


# multi of der channels
is # hidden layers +1
3
2
11
10
-15
10
V·V Vdouble


2.5 - 1 * 0.001
= 2.5 - 0.001
= 2.499
0 ≤ ∂f31 / ∂O21 ≤ 1
2.499
(w11)new
2.5
(w11) old
- η ∂L / ∂w11
0.001
1
V. small
2.499999999
2.5
∂O31 / ∂O21
∂O21 / ∂O11
∂O11 / ∂(w1)11
Signal
0.2
* 0.1
0.05
= 0.00010
= 10^-3
<1
<1
<1
<<1 V-small


Vanishing grad
dL / dw_ij^k
v.v small
chain rule & mult
<1
sigmoid
tanh
(w_ij^k)new
(w_ij^k)n
12:52
45 / 45


$(w_{ij}^k)_{new} = (w_{ij}^k)_{old} - \eta \frac{\partial L}{\partial w_{ij}^k}$
1
Vi. Vi. Image
$(w_{ij}^k)_{old}$
$(w_{ij}^k)_{new}$
iter :
1
Image
2
3
4
5
6


$(w_{ij}^k)_{new} = (w_{ij}^k)_{old} - \eta \frac{\partial L}{\partial w_{ij}^k}$
1
Vo Voltage
$(w_{ij}^k)_{old}$
$(w_{ij}^k)_{new}$
iter :
1
layer
2
3
4
5
6
19:16
48 / 49

$(w_{ij}^{k})_1 = 2.2$

$(w_{ij}^{k})_2 = 10.8$

$(w_{ij}^{k})_3 = 100.6$

$\vdots$

exploding grad!

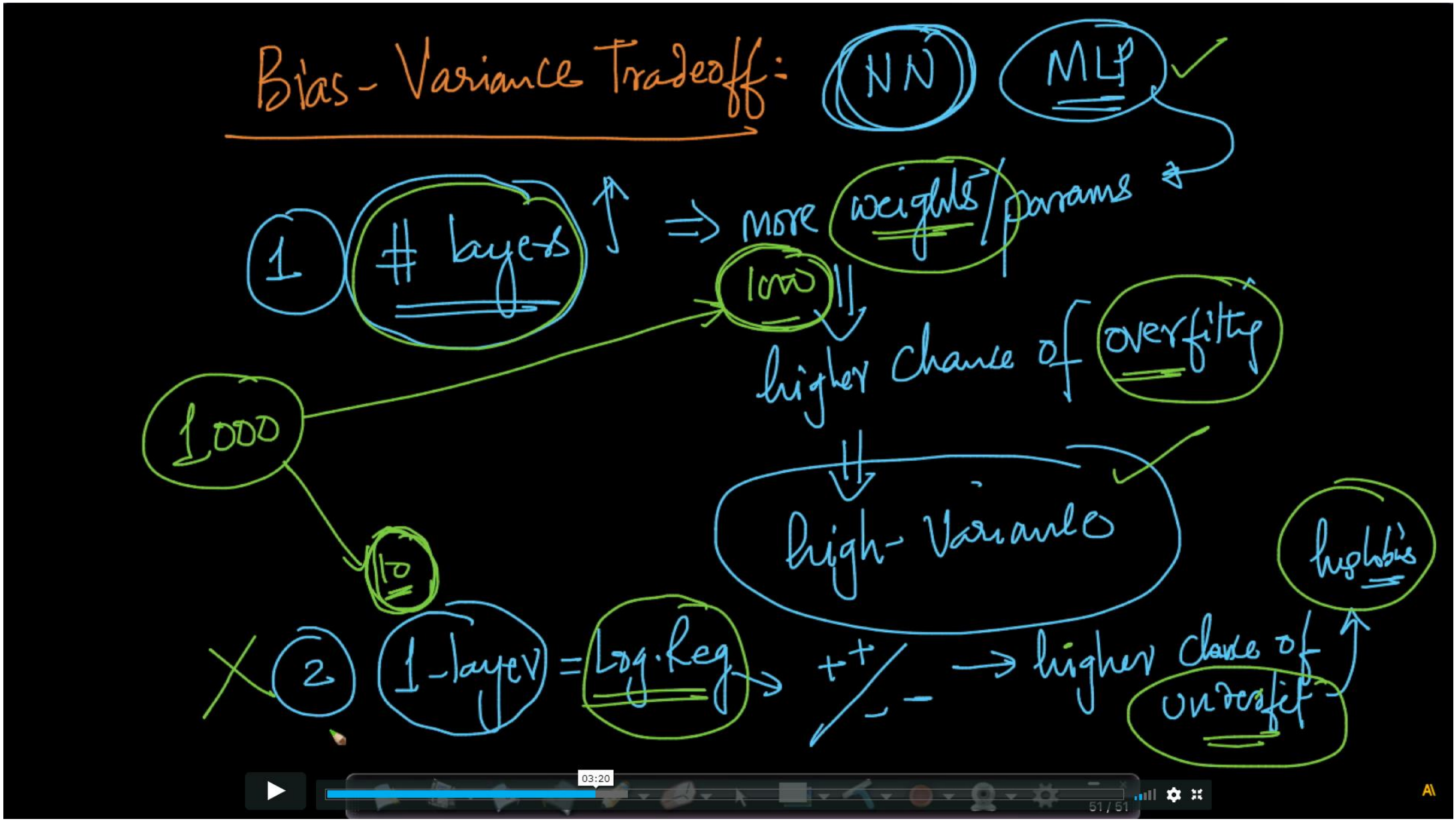
chain rule :- multiplication of derivatives

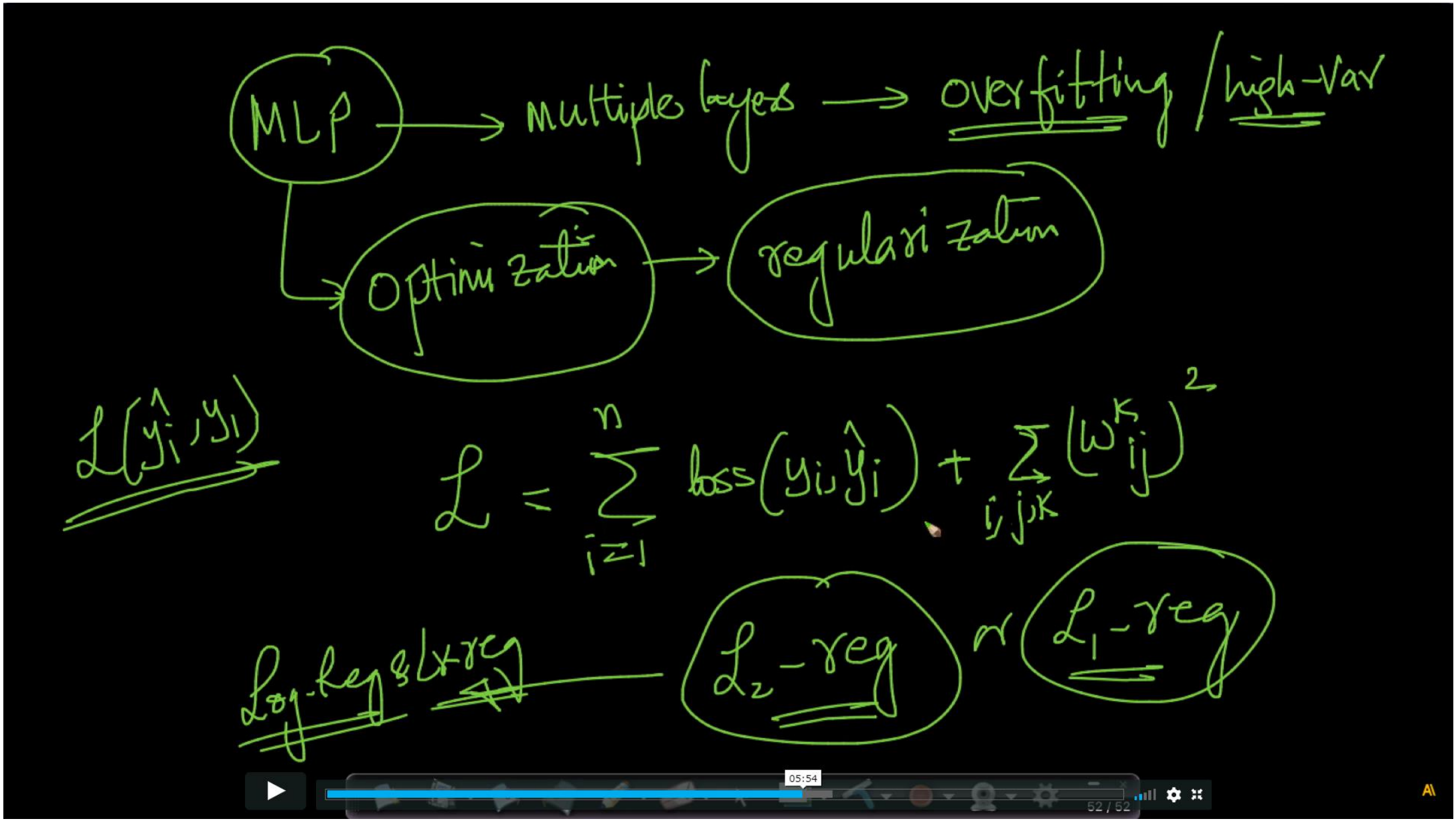
$f_{ij}$ : activation

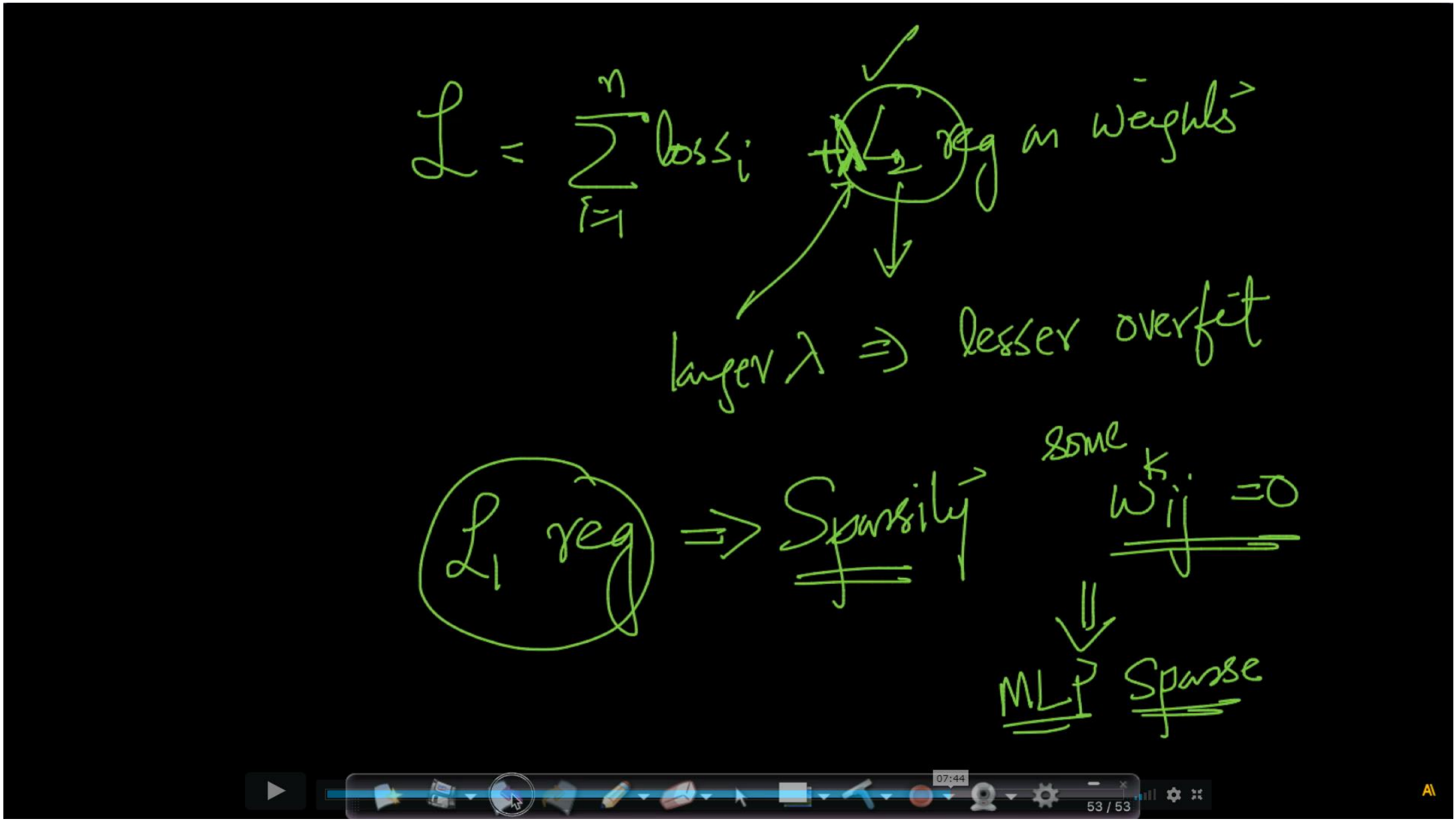
$$\frac{dL}{\partial w_{11}} = \frac{\partial o_{31}}{\partial o_{21}} \frac{\partial o_{21}}{\partial o_{11}} \frac{\partial o_{11}}{\partial w_{11}}$$

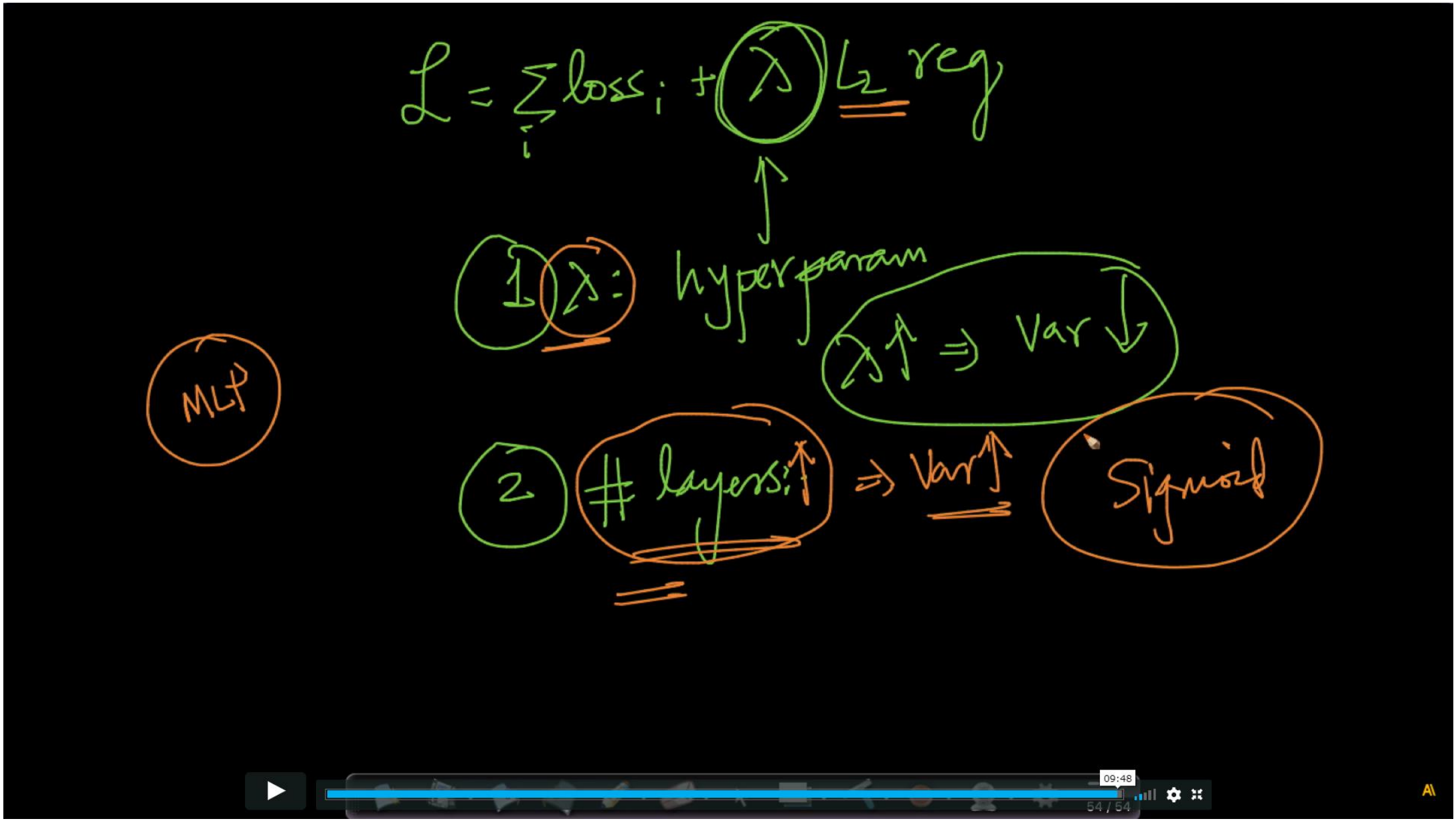
$?_1$ $?_1$ $?_1$

8 = 2 2 2


Bias - Variance Tradeoff:
NN
MLP
# layers
1
1000
more weights/params
low
higher chance of overfitting
high-Variance
high bias
10
2
1-layer = Log. Reg
++
--
higher chance of underfit
03:20
51 / 51


MLP
multiple layers
overfitting / high-var
optimization
regularization
$\mathcal{L}(\hat{y}_i, y_i)$
$\mathcal{L} = \sum_{i=1}^{n} loss(y_i, \hat{y}_i) + \sum_{i,j,k} (w_{ij}^{k})^2$
Log-reg & Lx-reg
$\mathcal{L}_2$-reg
or
$\mathcal{L}_1$-reg
05:54
52 / 52


$\mathcal{L} = \sum_{i=1}^{n} \text{loss}_i$
$+ L_2$ reg on weights
larger $\lambda \Rightarrow$ lesser overfit
$L_1$ reg $\Rightarrow$ Sparsity
some $w_{ij}^k = 0$
MLP Sparse
07:44
53 / 53


$\mathcal{L} = \sum_i \text{loss}_i + \lambda L_2 \text{ reg}$
1 $\lambda$: hyperparam
$\lambda\uparrow \Rightarrow$ Var $\downarrow$
MLP
2 # layers: $\uparrow$ $\Rightarrow$ Var $\uparrow$
Sigmoid
09:48
54 / 54