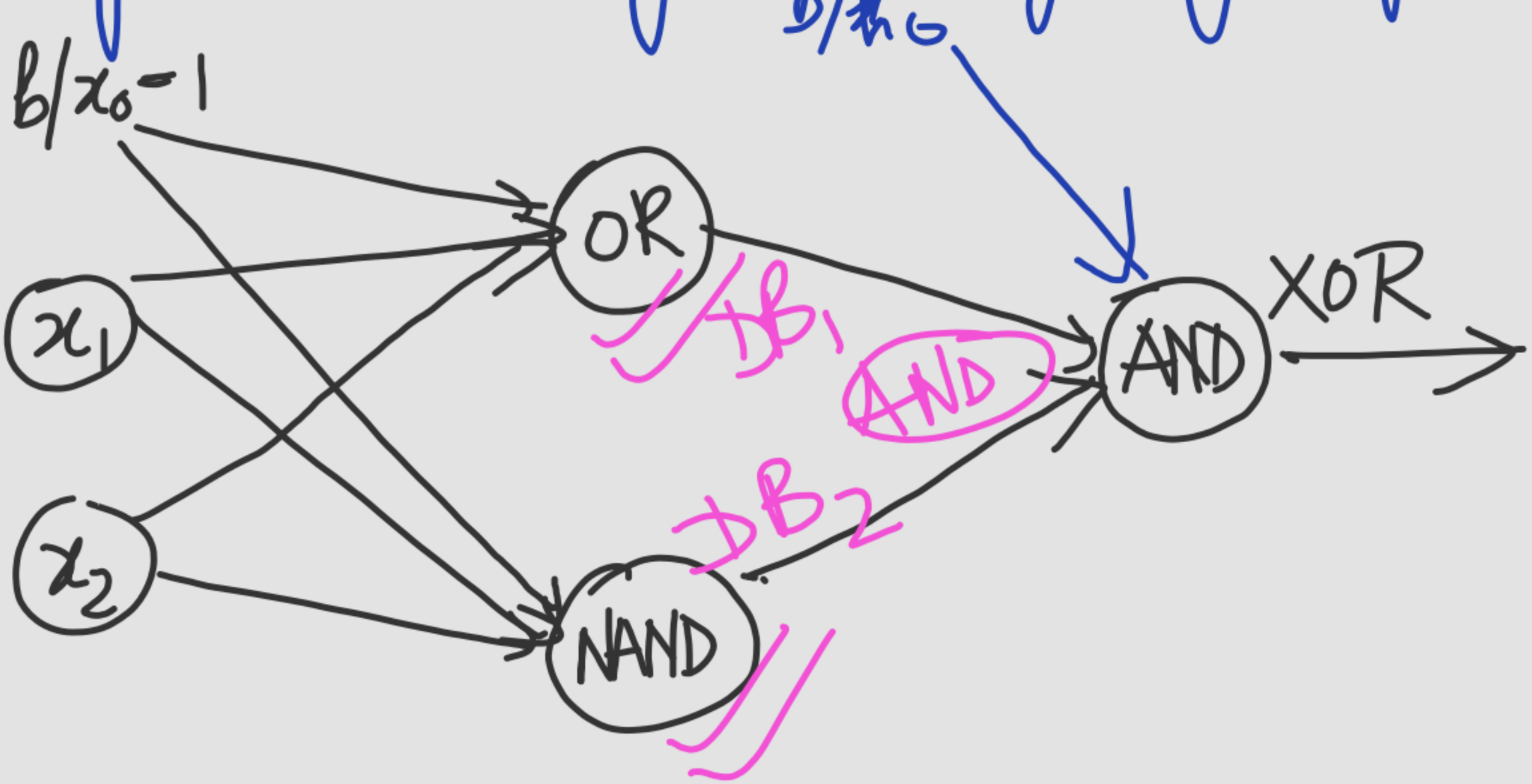
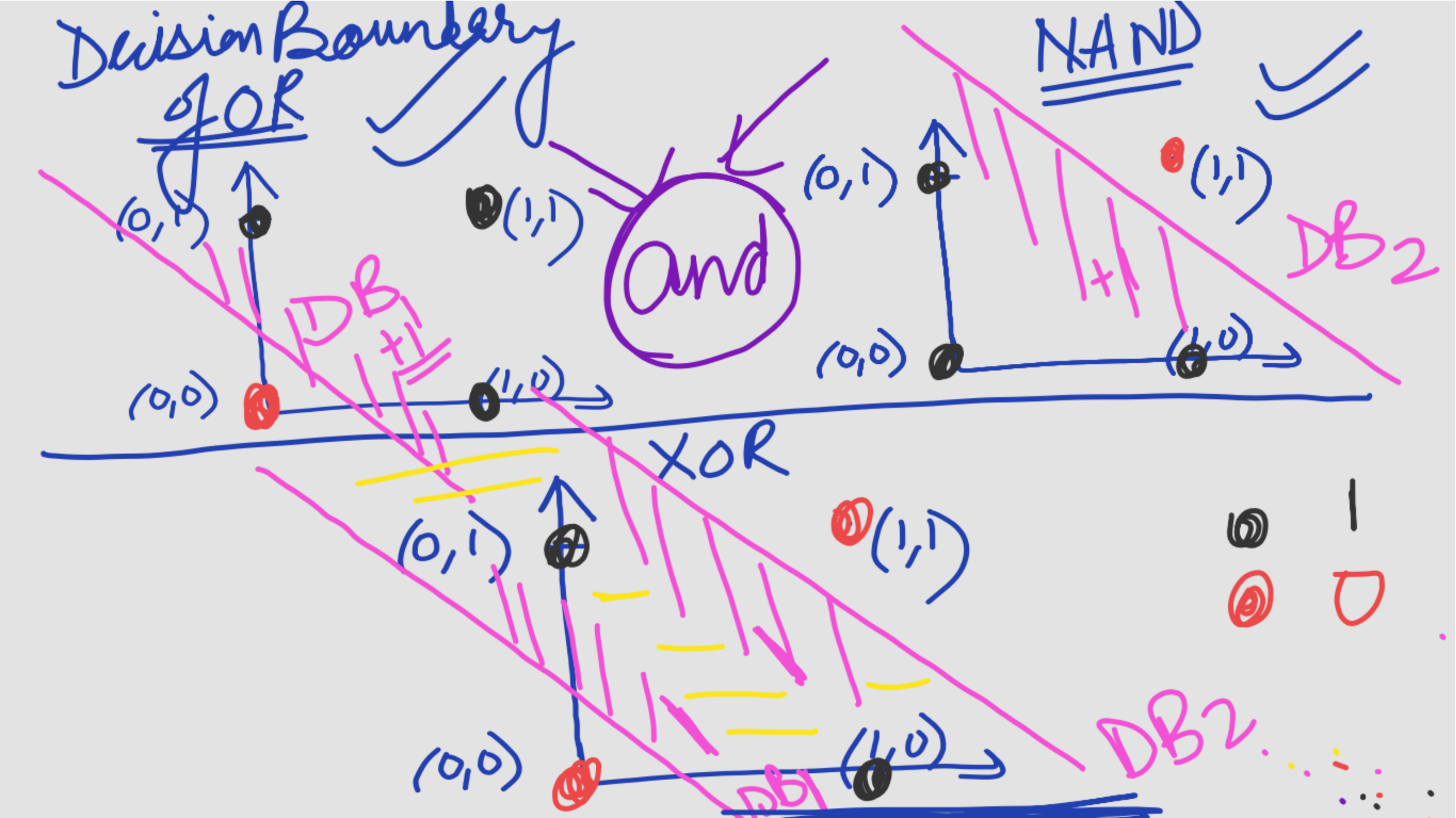
Decision Boundary of XOR when implemented using multilayer perceptron



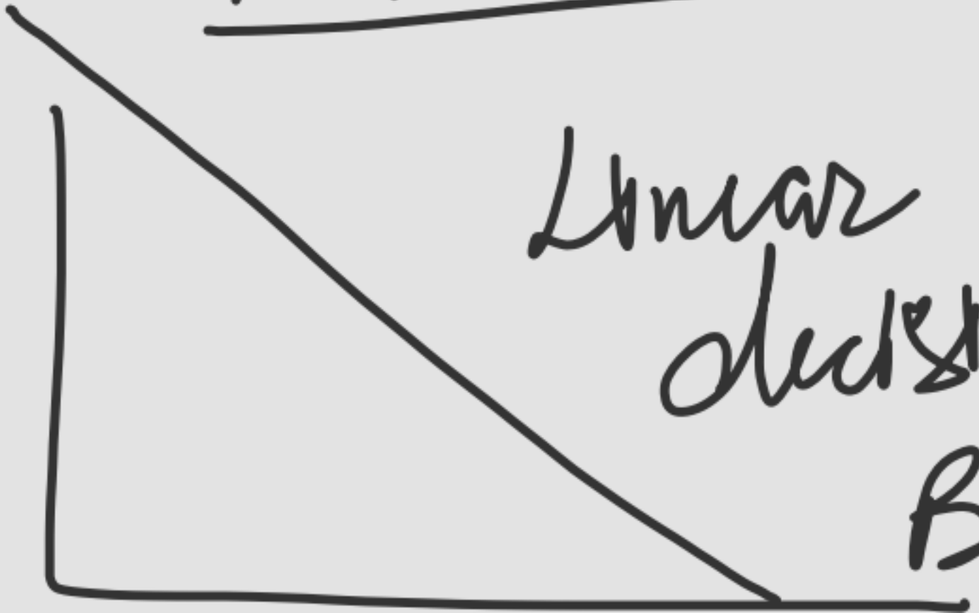

Decision Boundary of OR
(0,1)
(1,1)
(0,0)
(1,0)
DB1
+1
AND
NAND
(0,1)
(1,1)
(0,0)
(1,0)
DB2
+1
XOR
(0,1)
(1,1)
(0,0)
(1,0)
DB1
DB2
1
0

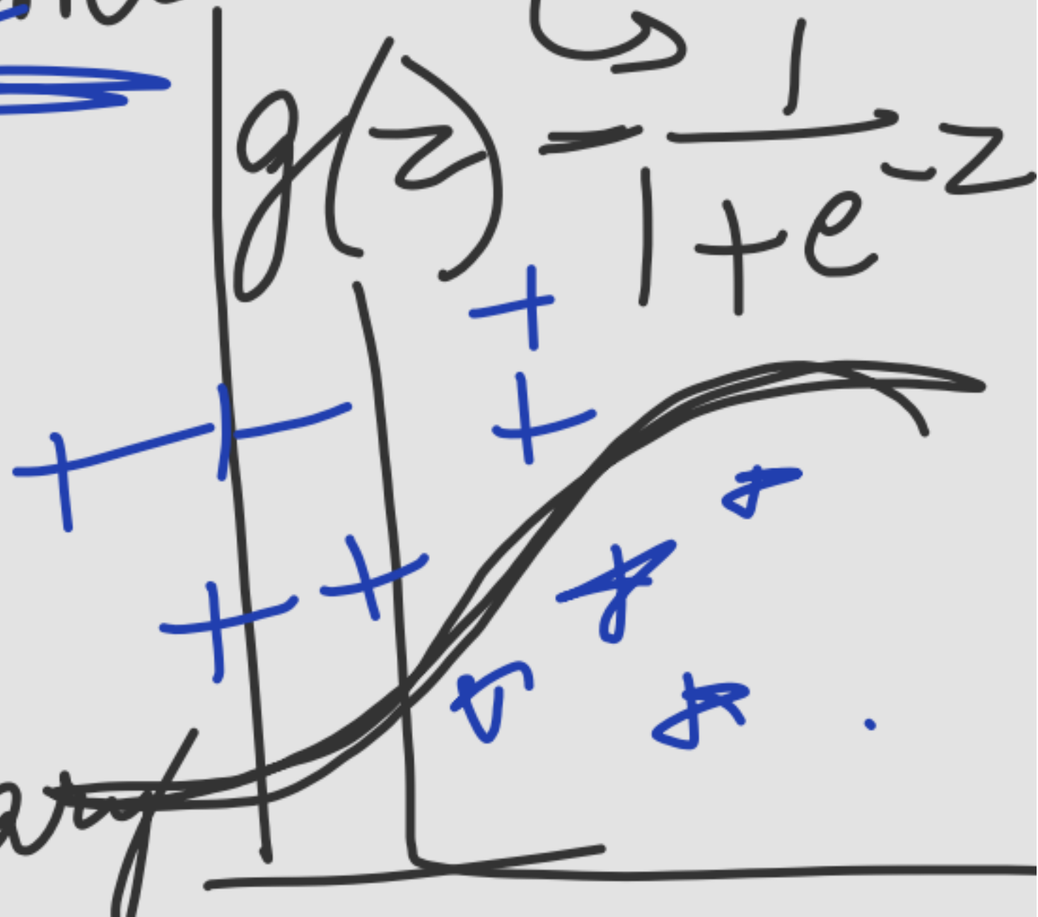
Activation Function → Non-linearity

Sigmoid Func

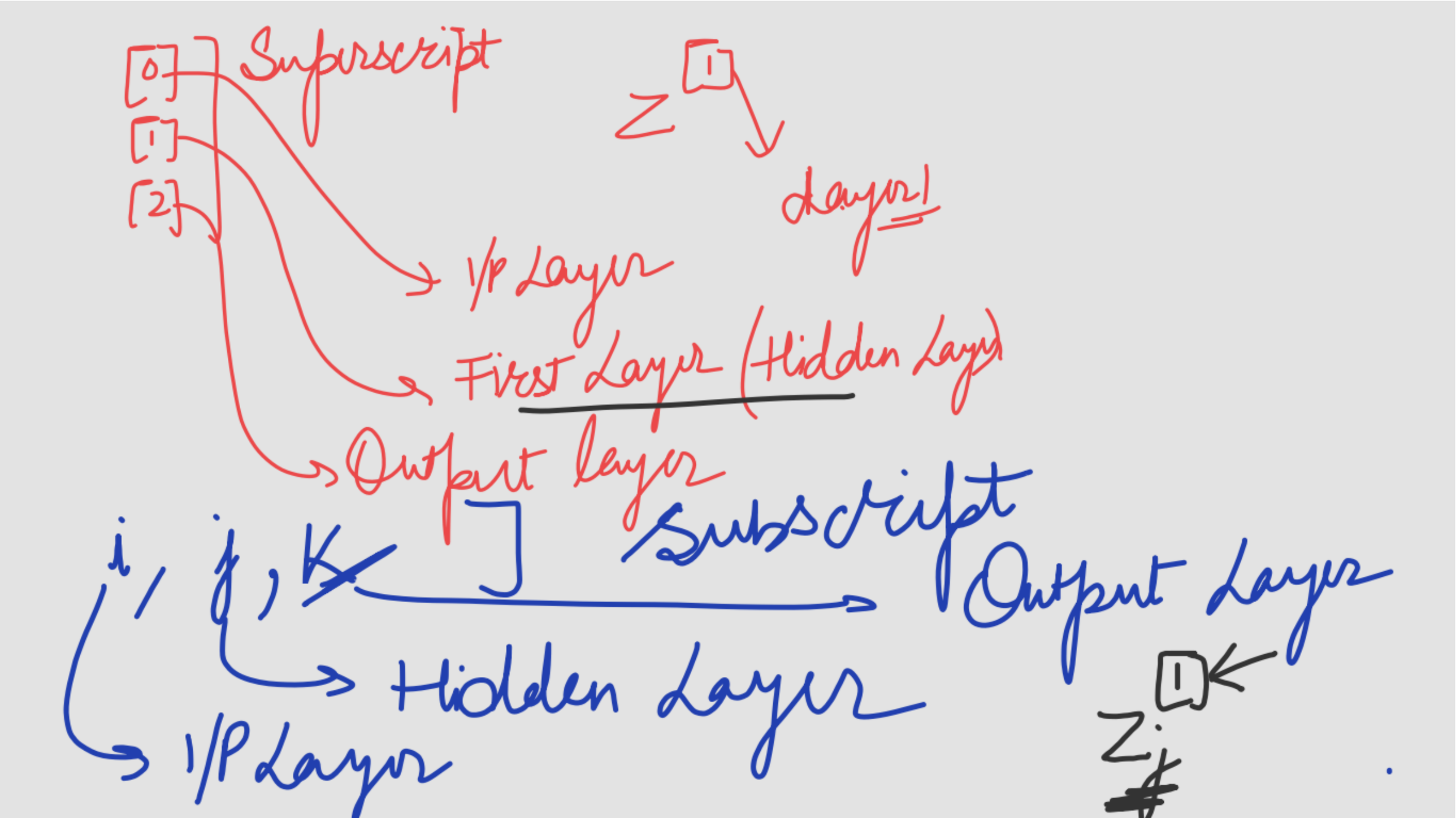
$$z = w_1 x_1 + w_2 x_2 + b$$

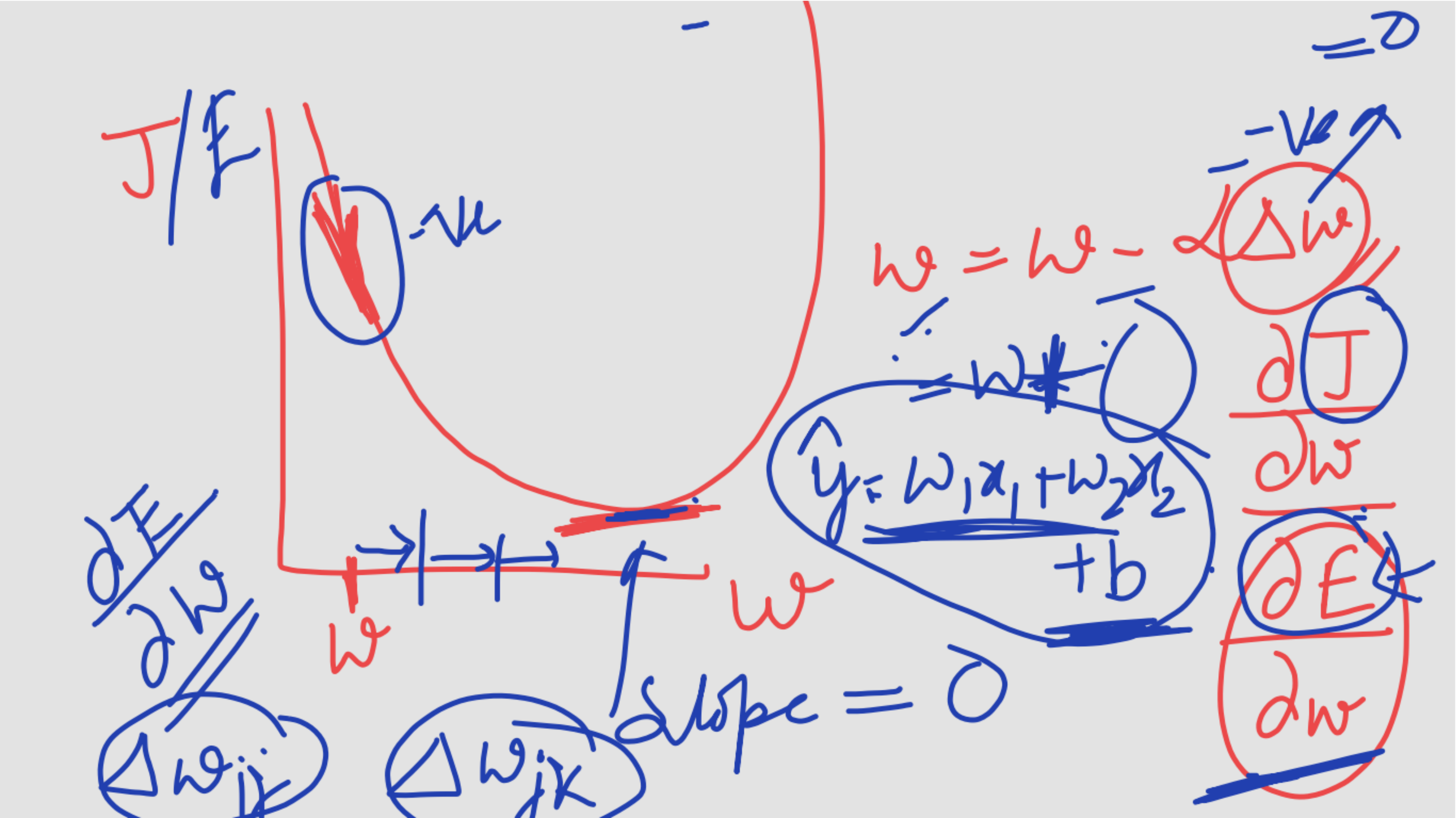
$$g(z) = \frac{1}{1+e^{-z}}$$

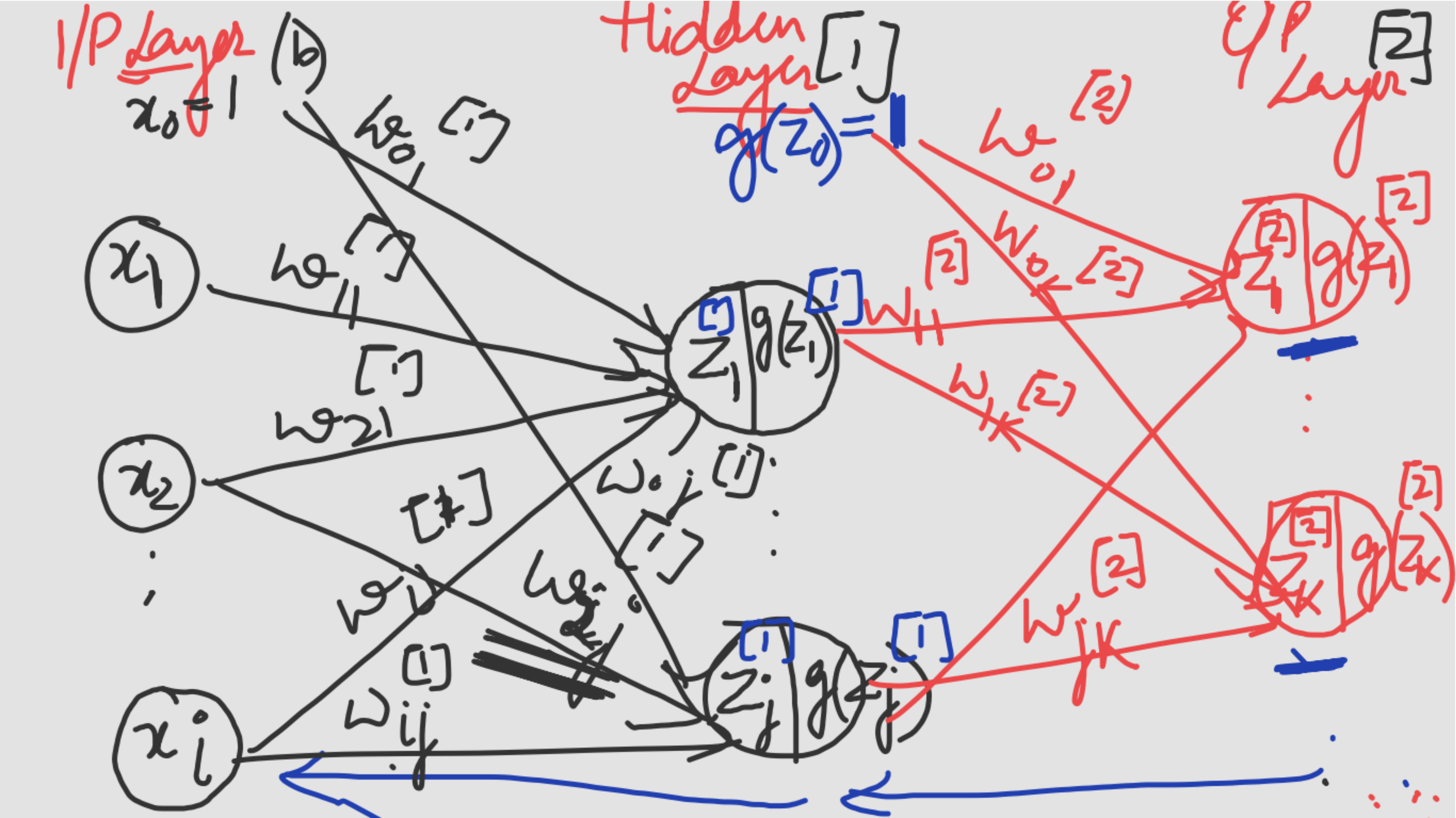




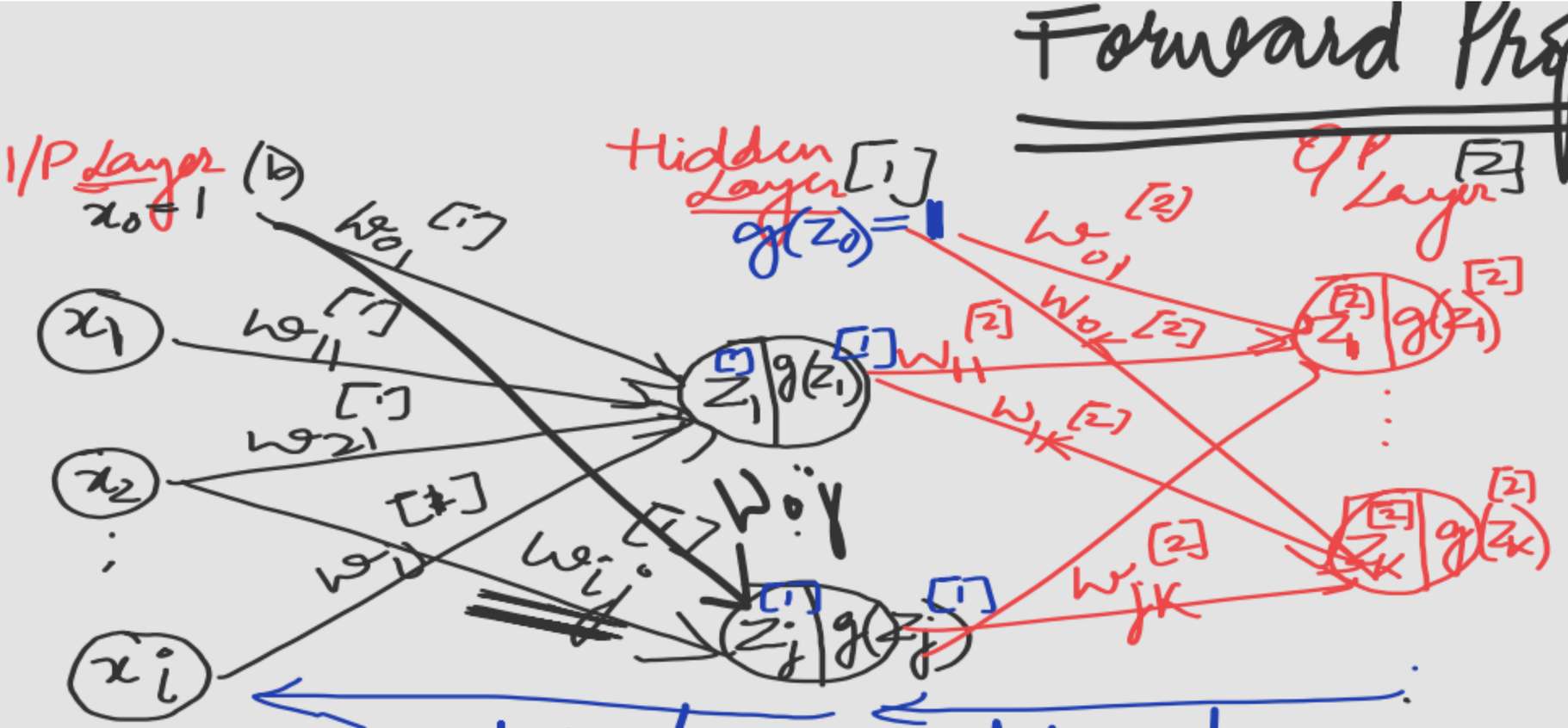
Forward & Backward Propagation


[0]
[1]
[2]
Superscript
Z [1]
Layer1
I/P Layer
First Layer (Hidden Layer)
Output layer
i, j, k
Subscript
Output Layer
Hidden Layer
I/P Layer
[1]
Zi


J/E
-ve
w
w
slope = 0
$\hat{y} = w_1 x_1 + w_2 x_2 + b$
$w = w - \alpha \Delta w$
-ve
$\frac{dJ}{dw}$
$\frac{dE}{dw}$
$\frac{dE}{dw}$
Δw_{jk}
Δw_{jk}


I/P Layer
(b)
$x_0 = 1$
x_1
x_2
x_i
$w_{01}^{[1]}$
$w_{11}^{[1]}$
$w_{21}^{[1]}$
$w_{i1}^{[1]}$
$w_{0j}^{[1]}$
$w_{ij}^{[1]}$
Hidden Layer [1]
$g(z_0) =$
$z_1^{[1]}$ $g(z_1)$
$z_j^{[1]}$ $g(z_j)$
$w_{01}^{[2]}$
$w_{0k}^{[2]}$
$w_{11}^{[1]}$
$w_{1k}^{[2]}$
$w_{jk}^{[2]}$
O/P Layer [2]
$z_1^{[2]}$ $g(z_1)^{[2]}$
$z_k^{[2]}$ $g(z_k)^{[2]}$

Forward Propagation



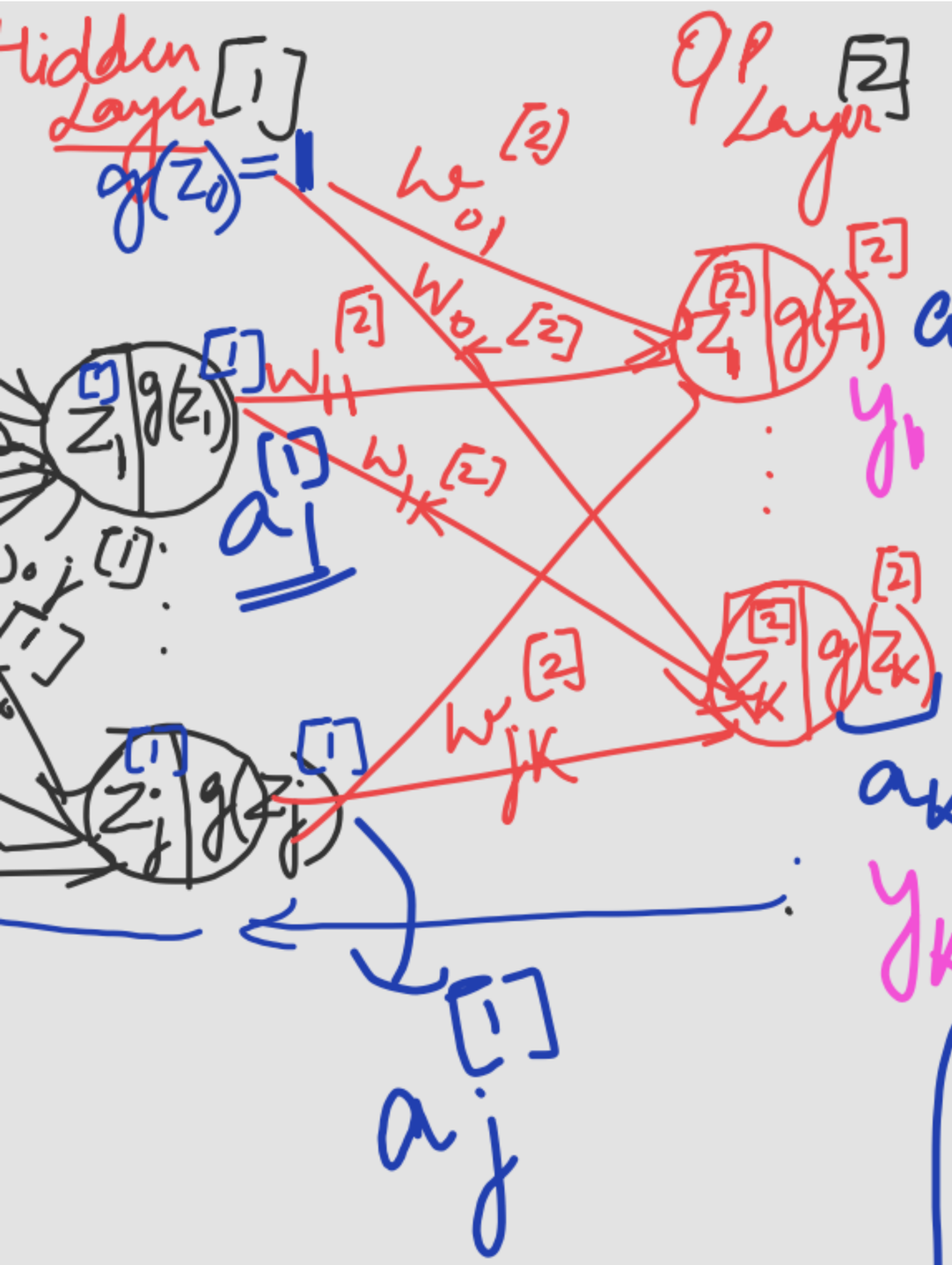
① Compute O/P of Hidden Layer

$$z_j^{[1]} = x_0 w_{0j}^{[1]} + x_1 w_{1j}^{[1]} + x_2 w_{2j}^{[1]} + \cdots + x_i w_{ij}^{[1]}$$

$$= \sum_i x_i w_{ij}^{[1]}$$

$$g(z_j)^{[1]} = a_j^{[1]} = a(z_j)^{[1]}$$

$$= \frac{1}{1 + e^{-z_j^{[1]}}}$$



(2) Compute the outputs of Output Layer

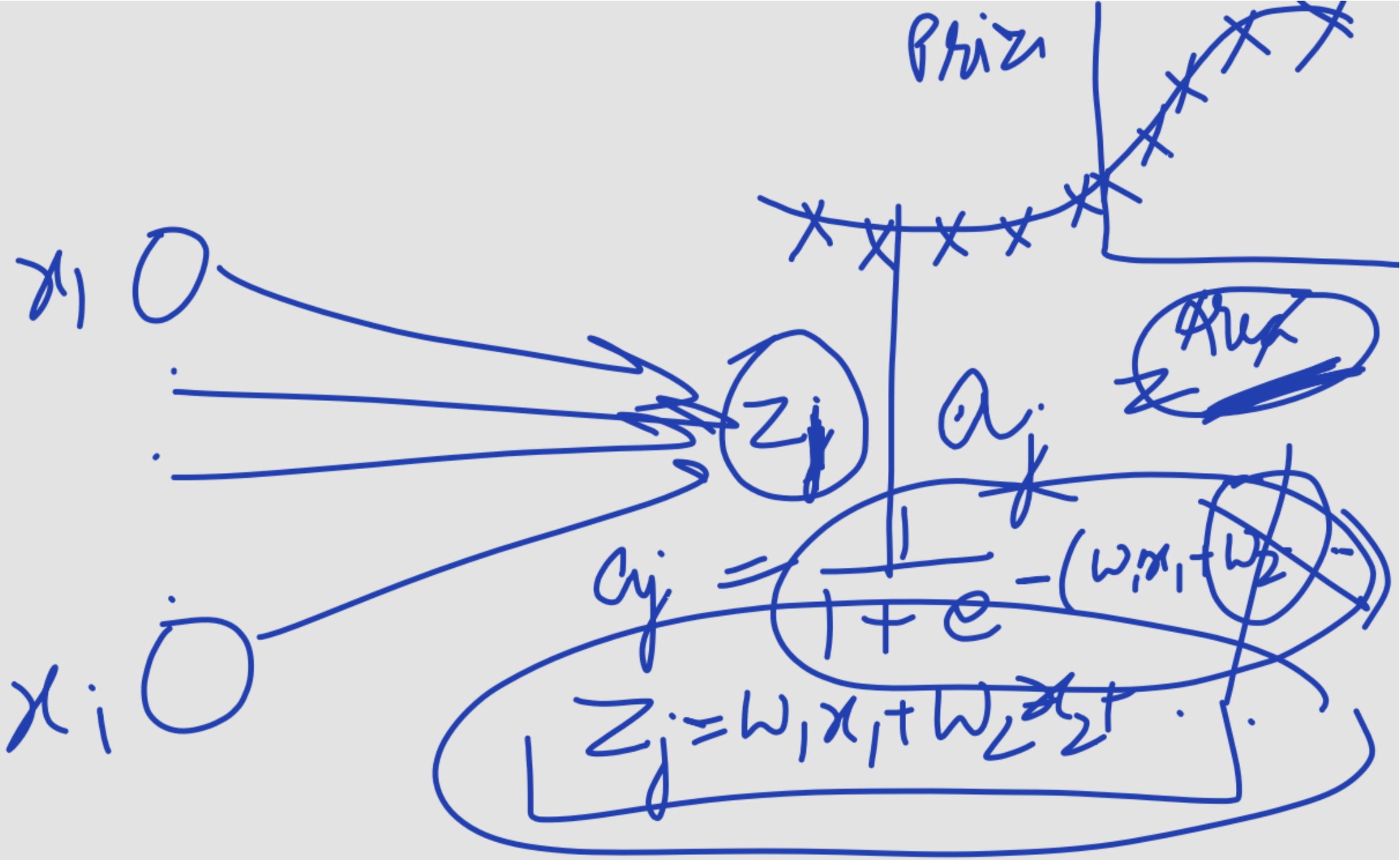
$$z_K^{[2]} = w_{0K}^{[2]} \cdot a_0^{[1]} + w_{1K}^{[2]} a_1^{[1]} + \cdots$$

$$w_{jK}^{[2]} a_j^{[1]}$$

$$= \sum_j w_{jK}^{[2]} a_j^{[1]}$$

$$a_K^{[2]} = \frac{1}{1 + e^{-z_K^{[2]}}}$$

$\therefore$


x_1
x_i
z_j
a_j
$a_j = \frac{1}{1+e^{-(w_1 x_1 + w_2 \ldots}}$
$z_j = w_1 x_1 + w_2 x_2 + \ldots$

Error

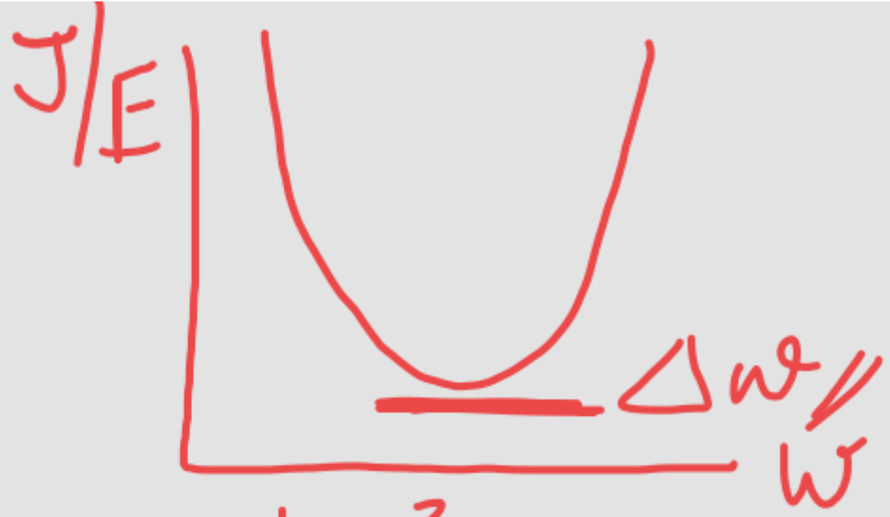
$$\mathcal{L} = \left(y_1^{[2]} - a_1^{[2]}\right)^2 + \ldots + \left(y_k^{[2]} - a_k^{[2]}\right)^2$$

Error

$$= \frac{1}{2} \sum_k \left(y_k^{[2]} - a_k^{[2]}\right)^2$$

J Cost

Backpropagation Algo



$$\frac{\partial E}{\partial w} = \Delta w$$

slope

$$z = w_1 x_1 + w_2 x_2 + b$$

$$a = \frac{1}{1+e^{-z}}$$

$$E = \frac{(y-a)^2}{2}$$

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial a} * \frac{\partial a}{\partial z} * \frac{\partial z}{\partial w_1}$$

$$a = 2u - v$$

$$u = 2w_1 - 4w_2$$

$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial a} \times \frac{\partial a}{\partial u} \times \frac{\partial u}{\partial w_2}$$

$$= \frac{2a}{2} * 2 * -4 =$$

$$-8a$$

$$\frac{\partial E}{\partial w} = \Delta w \quad \text{slope}$$

$$z = w_1 x_1 + w_2 x_2 + b$$

$$a = \frac{1}{1+e^{-z}}$$

$$E = \frac{1}{2}(y-a)^2$$

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial a} * \frac{\partial a}{\partial z} * \frac{\partial z}{\partial w_1}$$

$$\frac{\partial z}{\partial w_1} = x_1$$

$$\frac{\partial E}{\partial a} = \frac{2}{2}(y-a) * -1$$

$$\frac{\partial a}{\partial z} = \frac{\partial \left(1+e^{-z}\right)^{-1}}{\partial z}$$

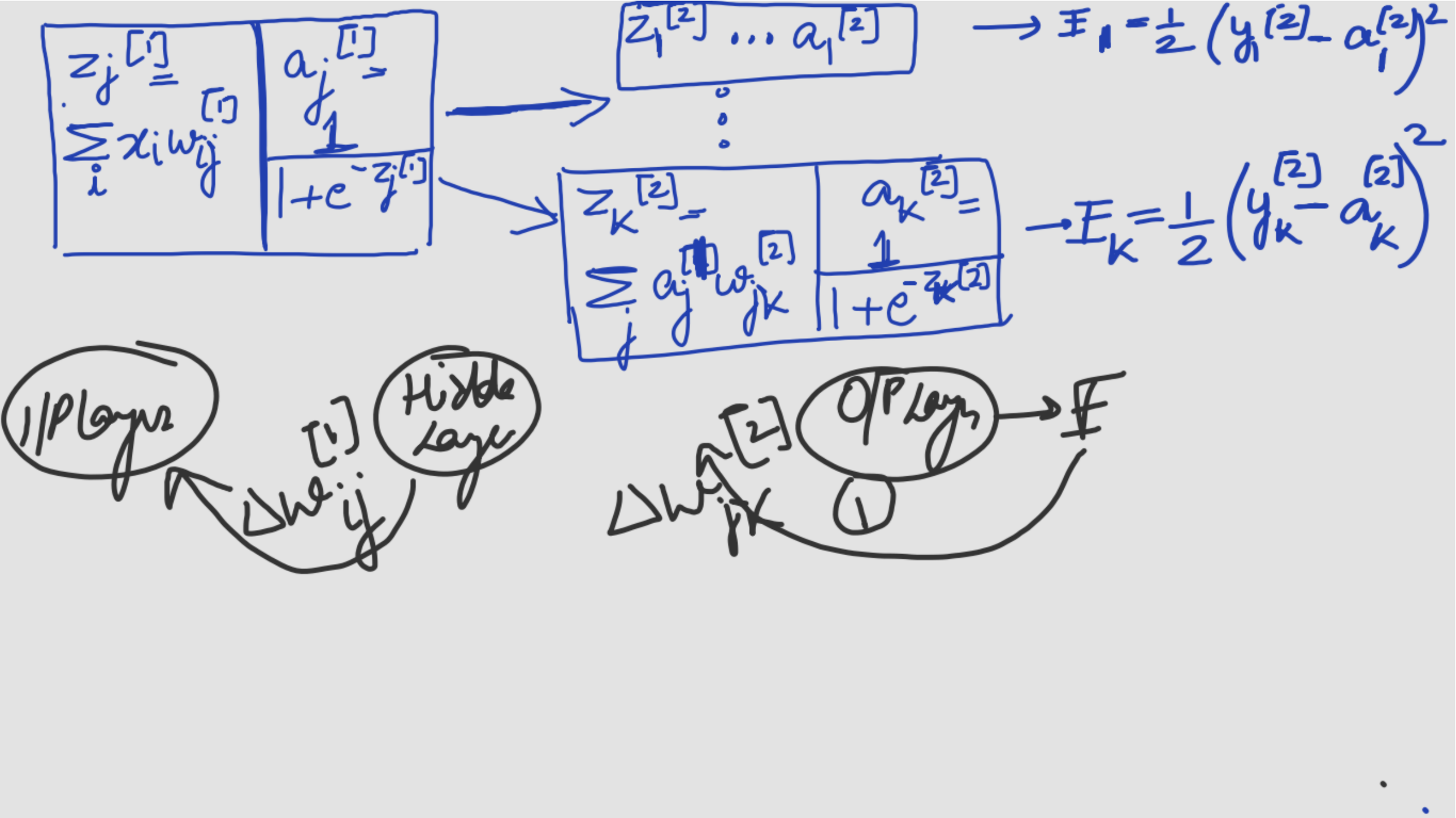
$$= -1\left(1+e^{-z}\right)^{-2} * -e^{-z}$$

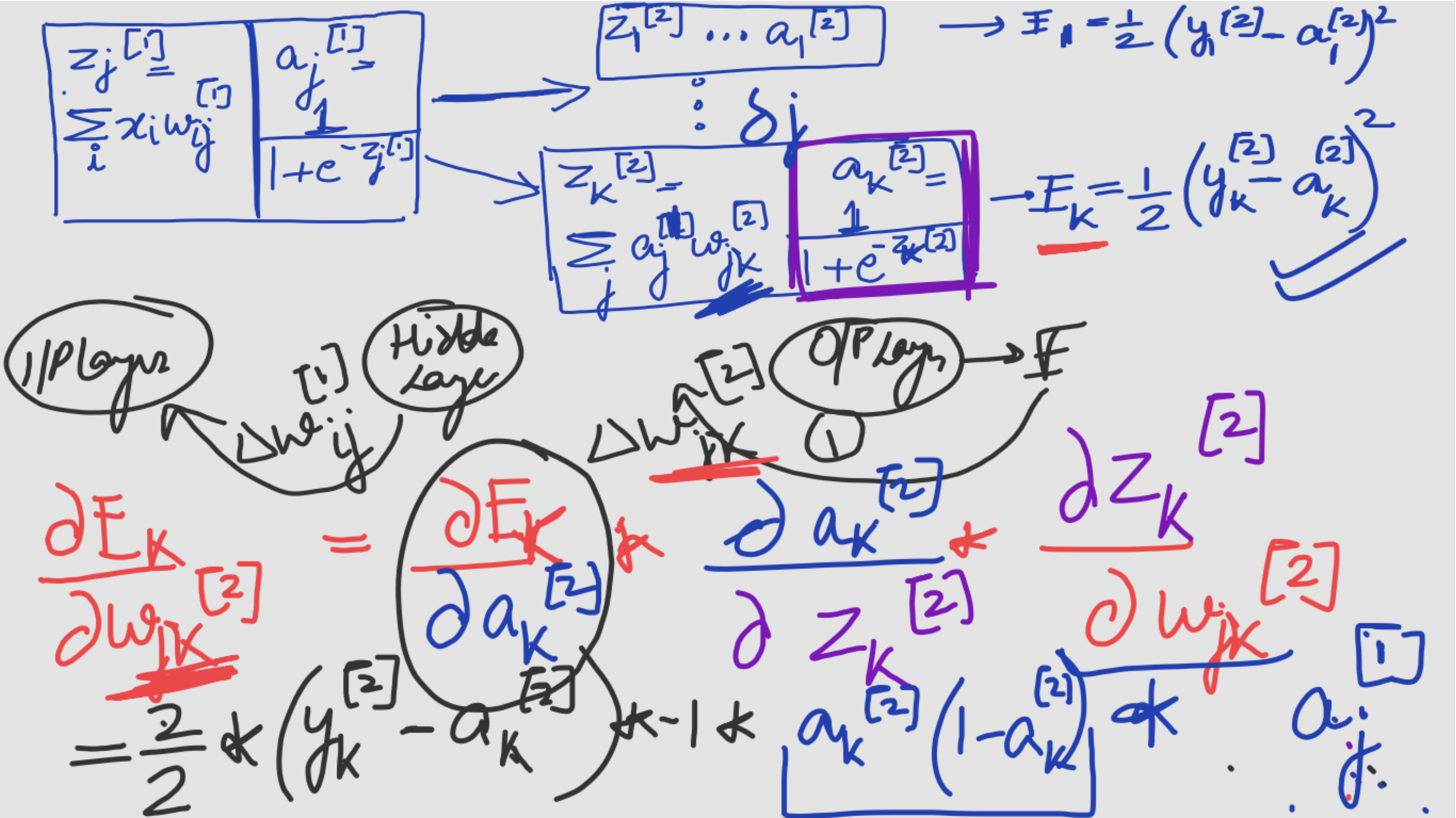
$$= \frac{+e^{-z}}{(1+e^{-z})^2}$$

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

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

$$= a(1-a)$$

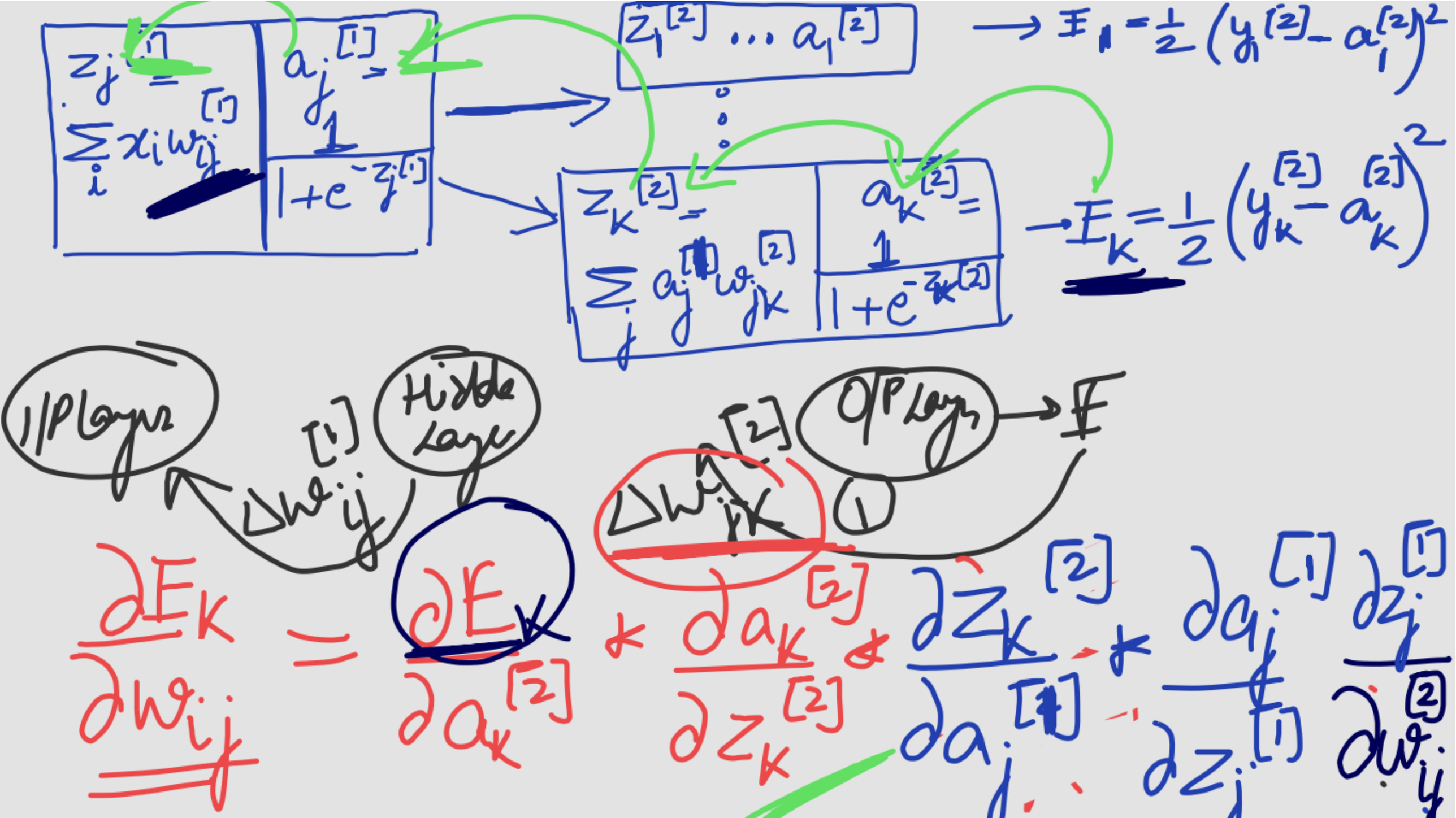

$z_j^{[1]} = \sum_i x_i w_{ij}^{[1]}$
$a_j^{[1]} = \frac{1}{1+e^{-z_j^{[1]}}}$
$z_1^{[2]} \dots a_1^{[2]}$
$\rightarrow E_1 = \frac{1}{2}(y_1^{[2]} - a_1^{[2]})^2$
$z_k^{[2]} = \sum_j a_j^{[1]} w_{jk}^{[2]}$
$a_k^{[2]} = \frac{1}{1+e^{-z_k^{[2]}}}$
$\rightarrow E_k = \frac{1}{2}(y_k^{[2]} - a_k^{[2]})^2$
I/P Layer
Hidde Layer
$\Delta w_{ij}^{[1]}$
O/P Layer
$\Delta w_{jk}^{[2]}$
E

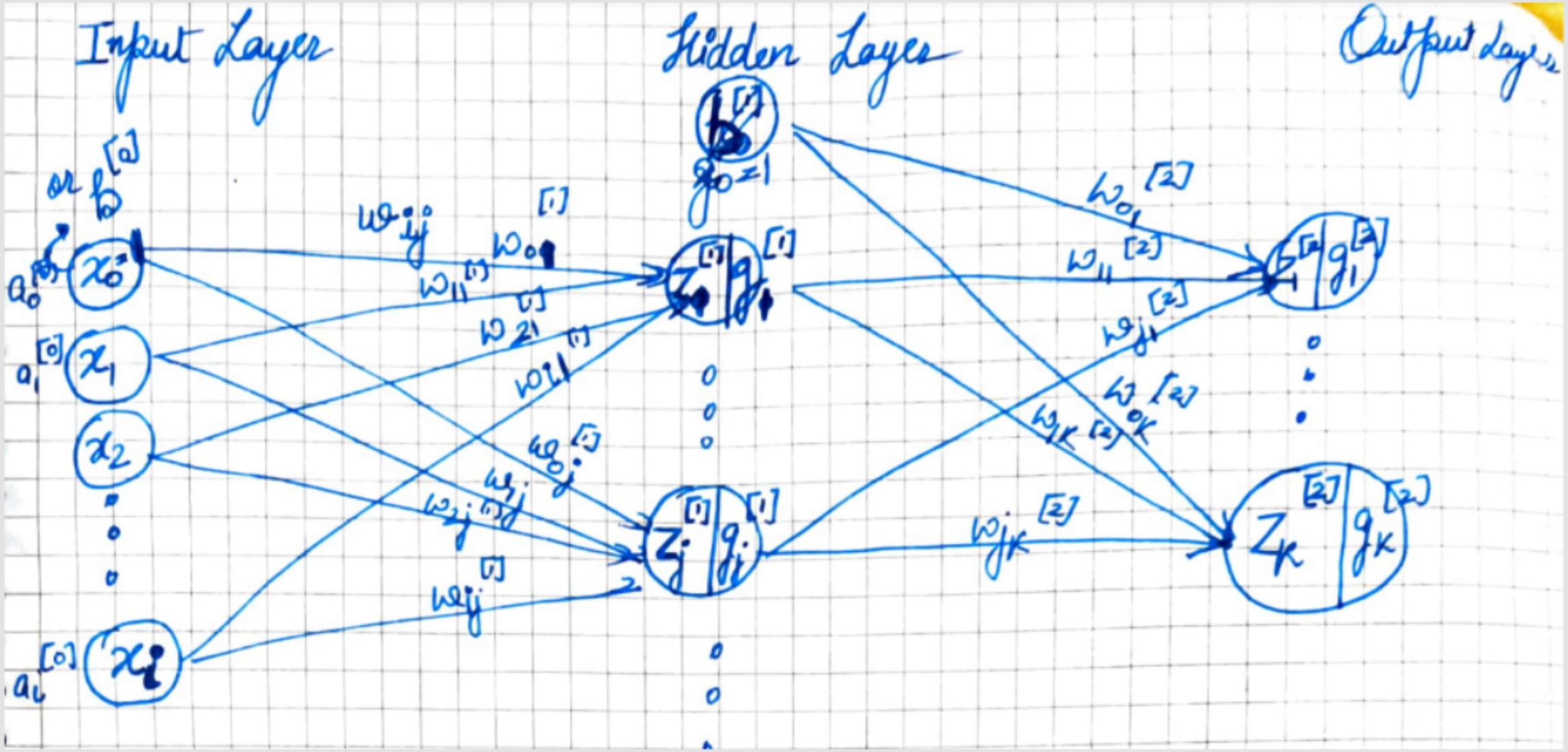

$z_j^{[1]} = \sum_i x_i w_{ij}^{[1]}$
$a_j^{[1]} = \frac{1}{1+e^{-z_j^{[1]}}}$
$z_1^{[2]} \dots a_1^{[2]}$
δ_j
$z_k^{[2]} = \sum_j a_j^{[1]} w_{jk}^{[2]}$
$a_k^{[2]} = \frac{1}{1+e^{-z_k^{[2]}}}$
$E_1 = \frac{1}{2}(y_1^{[2]} - a_1^{[2]})^2$
$E_k = \frac{1}{2}(y_k^{[2]} - a_k^{[2]})^2$
I/P Layer
Hidden Layer
O/P Layer
F
$\Delta w_{ij}^{[1]}$
$\Delta w_{jk}^{[2]}$
$\frac{\partial E_k}{\partial w_{jk}^{[2]}} = \frac{\partial E_k}{\partial a_k^{[2]}} * \frac{\partial a_k^{[2]}}{\partial z_k^{[2]}} * \frac{\partial z_k^{[2]}}{\partial w_{jk}^{[2]}}$
$= \frac{2}{2} * (y_k^{[2]} - a_k^{[2]}) * -1 * a_k^{[2]}(1-a_k^{[2]}) * a_j^{[1]}$

$$\frac{\partial E_k}{\partial w_{jk}^{[2]}} = -1 * (y_k - a_k^{[2]}) * a_k^{[2]} * (1 - a_k^{[2]}) * a_j^{[1]}$$

$$\delta_k = (y_k - a_k^{[2]}) * a_k^{[2]} * (1 - a_k^{[2]})$$

$$W_{jk} = w_{jk} - \alpha * \delta_k * a_j^{[1]}$$


$z_j^{[1]} = \sum_i x_i w_{ij}^{[1]}$
$a_j^{[1]} = \frac{1}{1+e^{-z_j^{[1]}}}$
$z_1^{[2]} \dots a_1^{[2]}$
$z_k^{[2]} = \sum_j a_j^{[1]} w_{jk}^{[2]}$
$a_k^{[2]} = \frac{1}{1+e^{-z_k^{[2]}}}$
$\rightarrow E_1 = \frac{1}{2}(y_1^{[2]} - a_1^{[2]})^2$
$\rightarrow E_k = \frac{1}{2}(y_k^{[2]} - a_k^{[2]})^2$
I/P Layer
Hidden Layer
O/P Layer
$\Delta w_{ij}^{[1]}$
$\Delta w_{jk}^{[2]}$
$\rightarrow E$
$\frac{\partial E_k}{\partial w_{ij}} = \frac{\partial E_k}{\partial a_k^{[2]}} * \frac{\partial a_k^{[2]}}{\partial z_k^{[2]}} * \frac{\partial z_k^{[2]}}{\partial a_j^{[1]}} * \frac{\partial a_j^{[1]}}{\partial z_j^{[1]}} * \frac{\partial z_j^{[1]}}{\partial w_{ij}^{[1]}}$


Input Layer
Hidden Layer
Output Layer
or $b^{[0]}$
x_0
x_1
x_2
x_i
$a_0^{[0]}$
$a_1^{[0]}$
$a_i^{[0]}$
$b^{[1]}$
$g_0 = 1$
$z_j^{[1]}$ $g_j^{[1]}$
$w_{ij}^{[1]}$
$w_{0j}^{[1]}$
$w_{1j}^{[1]}$
$w_{2j}^{[1]}$
$w_{ij}^{[1]}$
$b^{[2]}$
$w_{01}^{[2]}$
$w_{11}^{[2]}$
$w_{j1}^{[2]}$
$w_{0K}^{[2]}$
$w_{jK}^{[2]}$
$z_1^{[2]}$ $g_1^{[2]}$
$z_K^{[2]}$ $g_K^{[2]}$

Forward Propogation

1. Computation of Outputs of Hidden Layer

$$z_j^{[1]} = x_0 w_{0j}^{[1]} + x_1 w_{1j}^{[1]} + x_2 w_{2j}^{[1]} + \dots + x_i w_{ij}^{[1]}$$

$$= \sum_i x_i w_{ij}^{[1]}$$

$$a_j^{[1]} = g(z_j)^{[1]} = g_j^{[1]} = \frac{1}{1+e^{-z_j^{[1]}}}$$

2. Computation of Outputs of Output Layer

$$z_k^{[2]} = a_0^{[1]} w_{0k}^{[2]} + a_1^{[1]} w_{1k}^{[2]} + \dots + a_j^{[1]} w_{jk}^{[2]}$$

$$z_k^{[2]} = \sum_j a_j^{[1]} w_{jk}^{[2]}$$

$$a_k^{[2]} = g(z_k)^{[2]} = g_k^{[2]} = \frac{1}{1+e^{-z_k^{[2]}}}$$

In general

$$z_i^{[l]} = \sum_k w_{ki}^{[l]} a_k^{[l-1]}$$

and

$$a_i^{[l]} = \frac{1}{1+e^{-z_i^{[l]}}}$$

$$\mathcal{E} = \frac{1}{2} \sum_k \left(y_k^{[2]} - a_k^{[2]} \right)^2$$

BACKPROPOGATION

① Create a feed-forward neural networks with $n^{[0]}$ input units, $n^{[1]}$ hidden units & $n^{[2]}$ output units.

② Initialize all network weights to small random values (between -0.5 & 0.5)

③ Until Termination Condition is met (# of iterations, Total Error almost same / change in weights almost zero w.r.t. last few iterations)

Do

For Each Training Example, Do #Stochastic Gradient Descent

3.1 Propogate the I/P forward through the network i.e.
→ Compute outputs of neurons/units at layer [1] using inputs of layer [0]
→ Forward propogate the outputs of layer [1] to compute the outputs of layer [2]

3.2 For each network output unit K, calculate error E_K as

$$\delta_K \left(\frac{\partial E}{\partial z_K^{[2]}}\right) = \left(y_K - a_K^{[2]}\right) * a_K^{[2]} * \left(1 - a_K^{[2]}\right)$$

3.3 For each hidden unit j, calculate error E_j as

$$\delta_j \left(\frac{\partial E}{\partial z_j^{[1]}}\right) = a_j^{[1]} * \left(1 - a_j^{[1]}\right) \sum_K w_{jK}^{[2]} \delta_K$$

Update network weights

3.4

For weights associated with Output layer

$$w_{jk}^{[2]} = w_{jk}^{[2]} + \Delta w_{jk}^{[2]}$$

$$\Delta w_{jk}^{[2]} = \left(\alpha \frac{\partial E}{\partial w_{jk}^{[2]}}\right) = \frac{\partial E}{\partial z_k^{[2]}} * \frac{\partial z_k^{[2]}}{\partial w_{jk}^{[2]}}$$

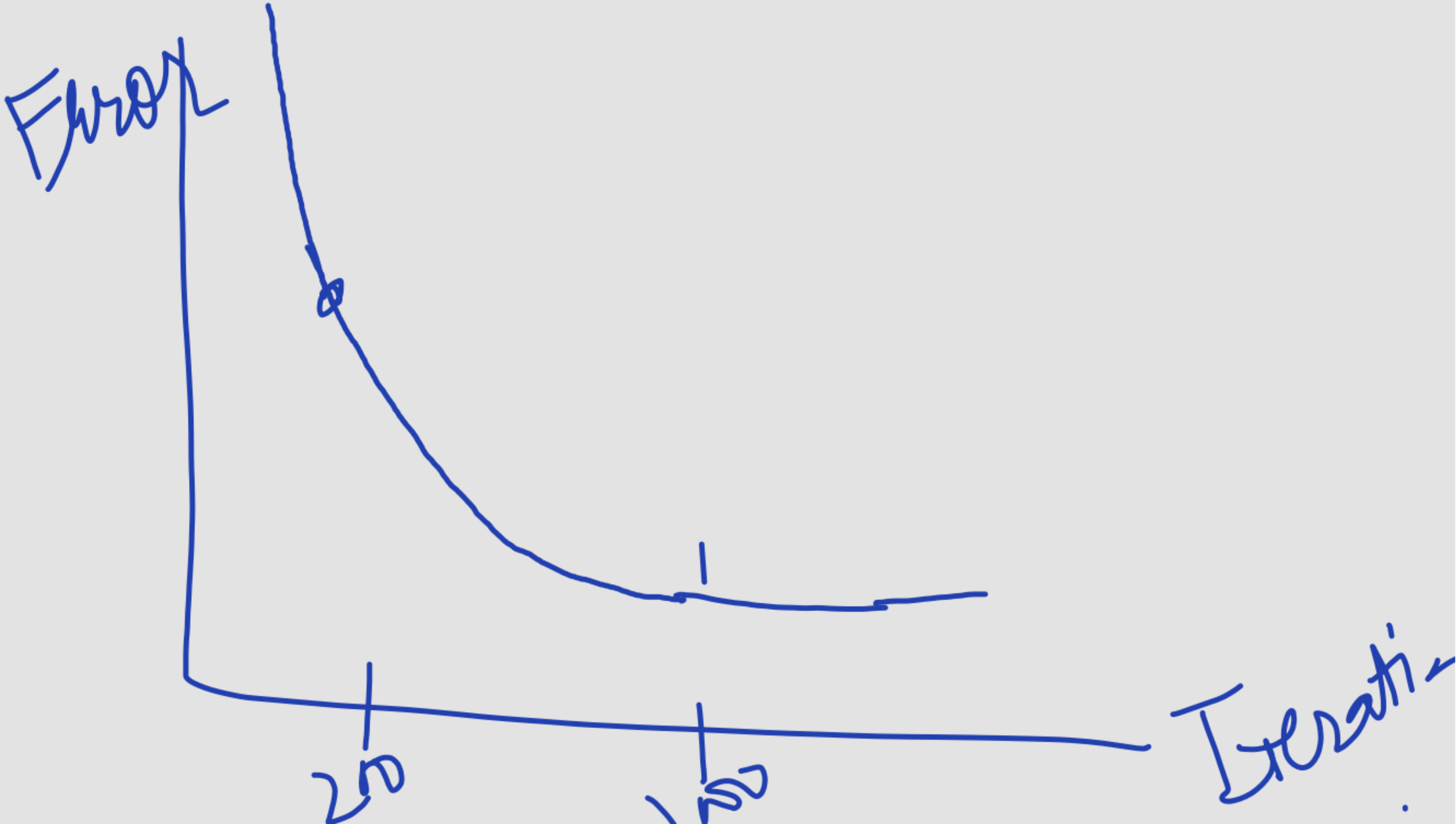
$\delta_k^{[2]} * a_j^{[1]}$

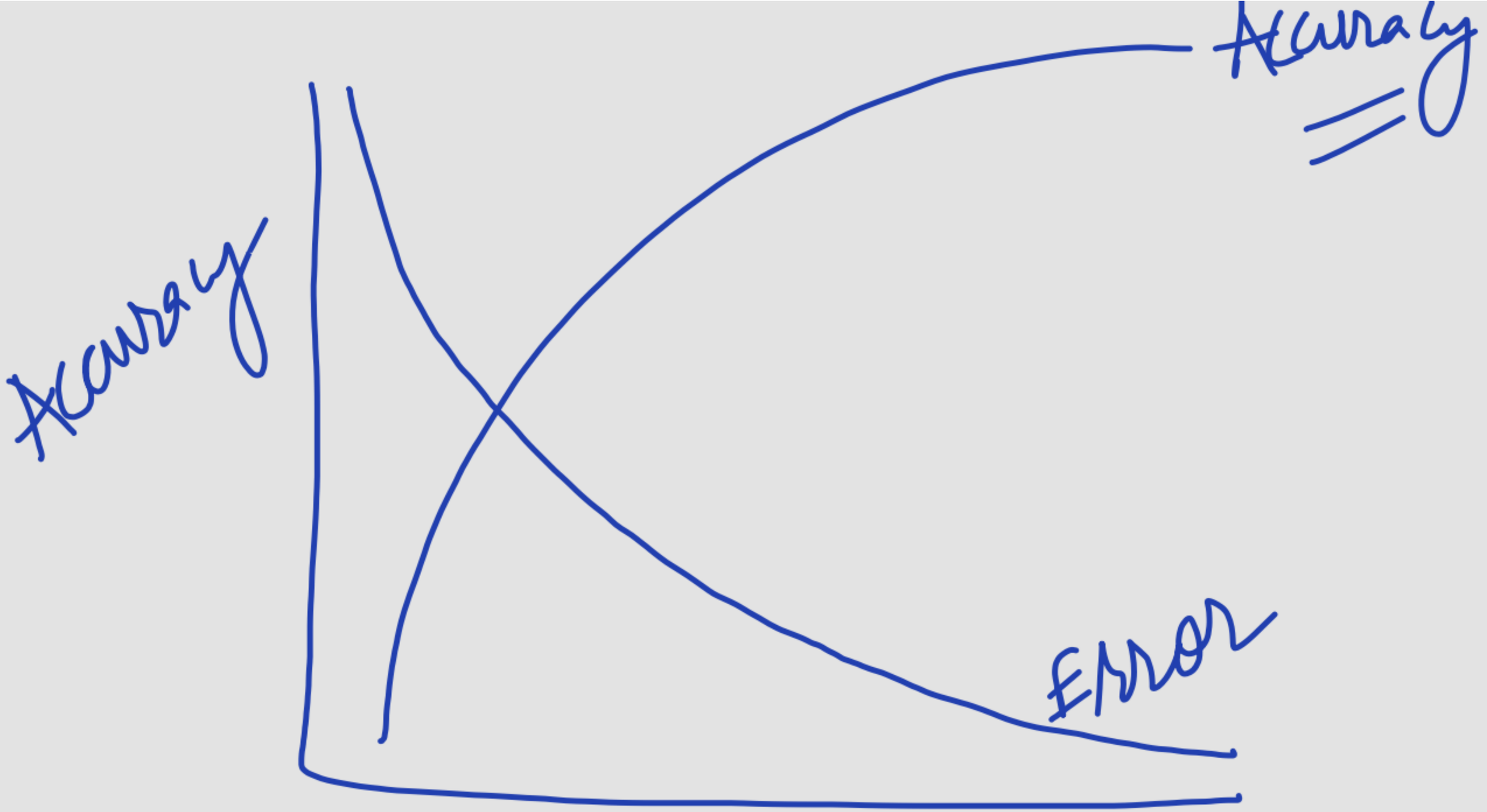
For weights associated with hidden layer

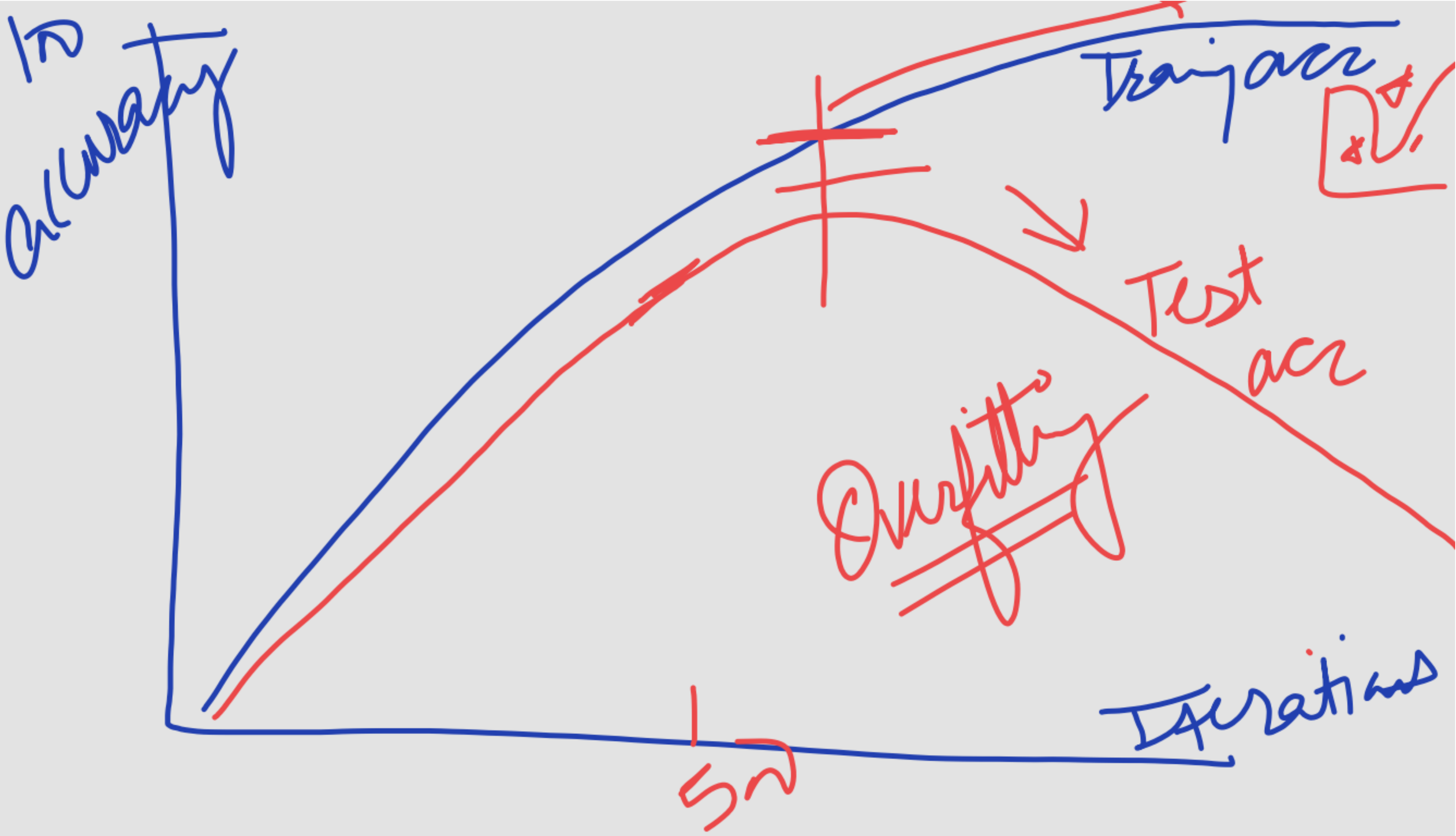
$$w_{ij}^{[1]} = w_{ij}^{[1]} + \Delta w_{ij}^{[1]}$$

$$\Delta w_{ij}^{[1]} = \left(\alpha \frac{\partial E}{\partial w_{ij}^{[1]}}\right) = \frac{\partial E}{\partial z_j^{[1]}} * \frac{\partial z_j^{[1]}}{\partial w_{ij}^{[1]}}$$

$\delta_j^{[1]} * x_i^{[0]}$


Error
1
250
Iterati


Accuracy
Accuracy
Error


Accuracy
Train acc
Test acc
Overfitting
50
Iterations