NEURAL NETWORKS

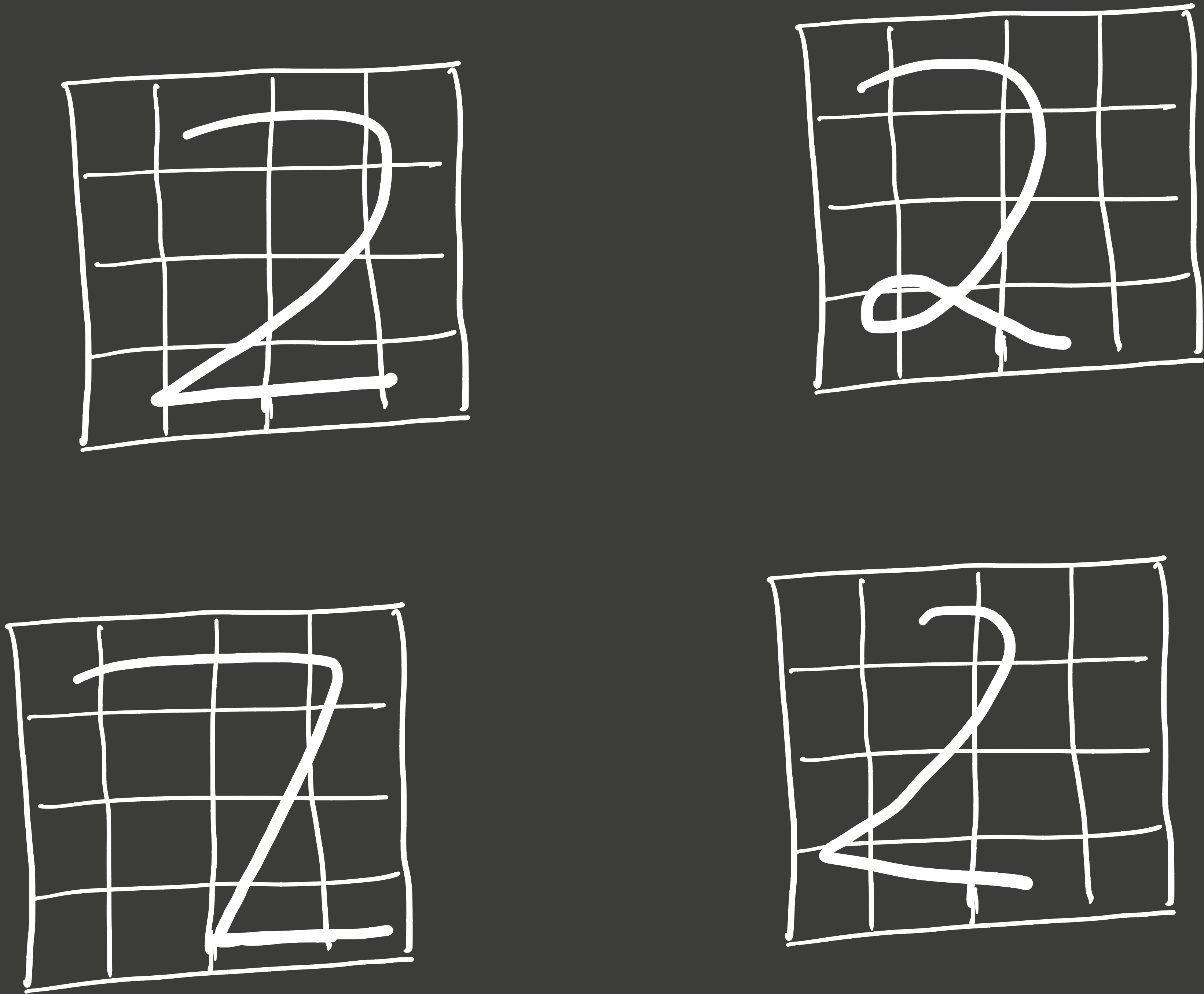
* ORIGINALLY BIOLOGICALLY INSPIRED
* STATE-OF-THE ART IN MOST FIELDS
* TURING AWARD WINNERS - BENGIO, LECUNN, HINTON

Q). What is the following?

2 3 4 4

* EASY FOR US TO RECOGNIZE

* WHAT ABOUT COMPUTERS?

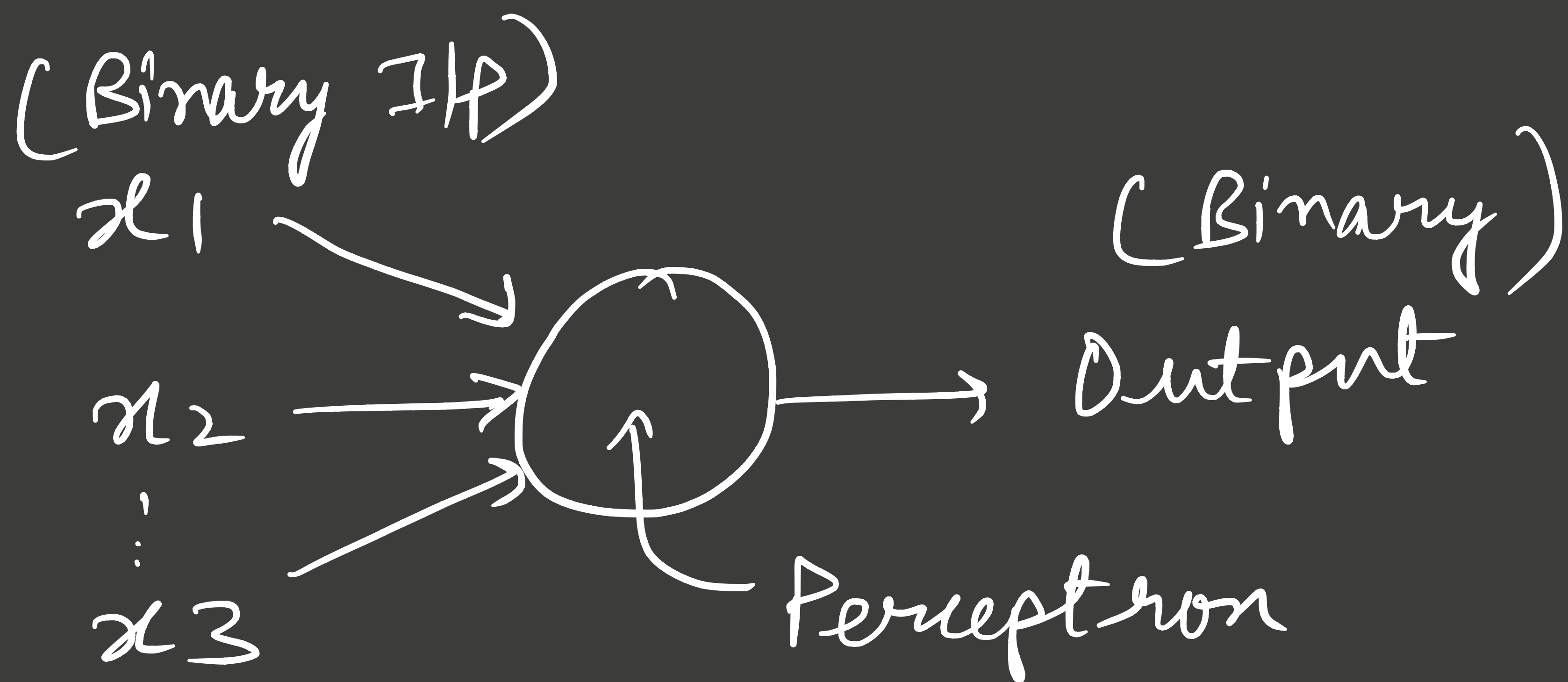


* COMPUTER SEES 16 pixels
* NON-TRIVIAL TO WRITE PROGRAM!

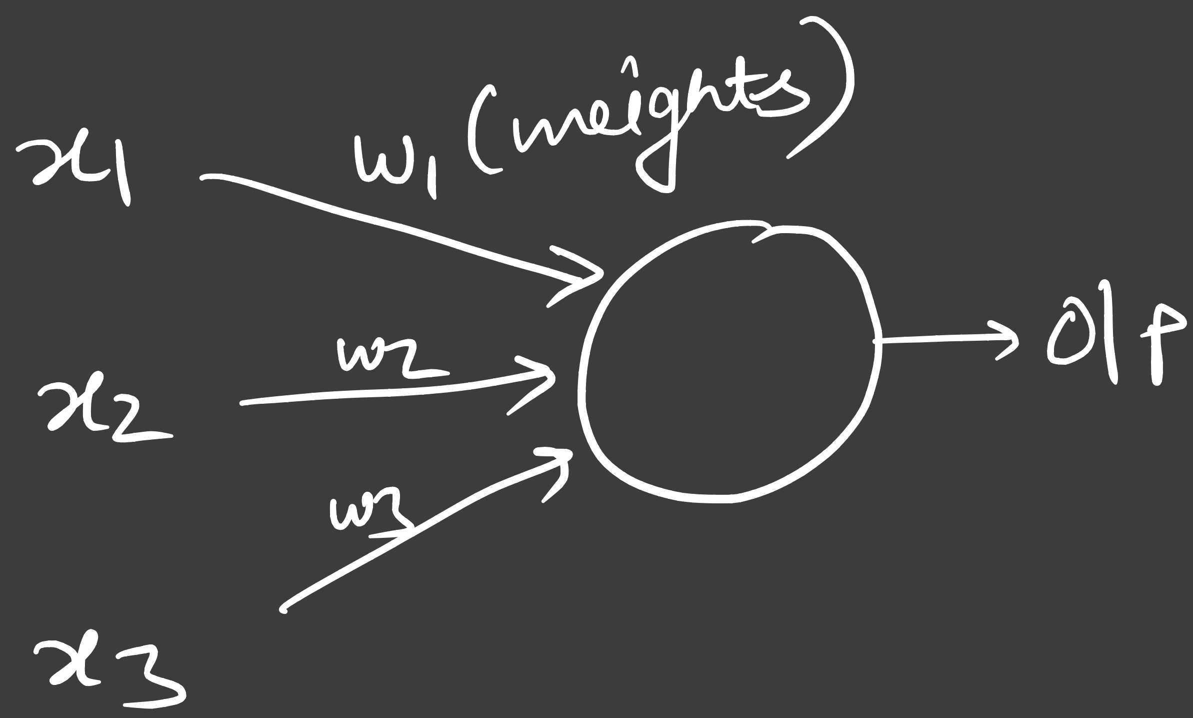
$\Longrightarrow$ LEARNING!

Perceptron

* Artificial neuron developed in 1950s/60s by Rosenblatt inspired by Mc Culloch & Pitts

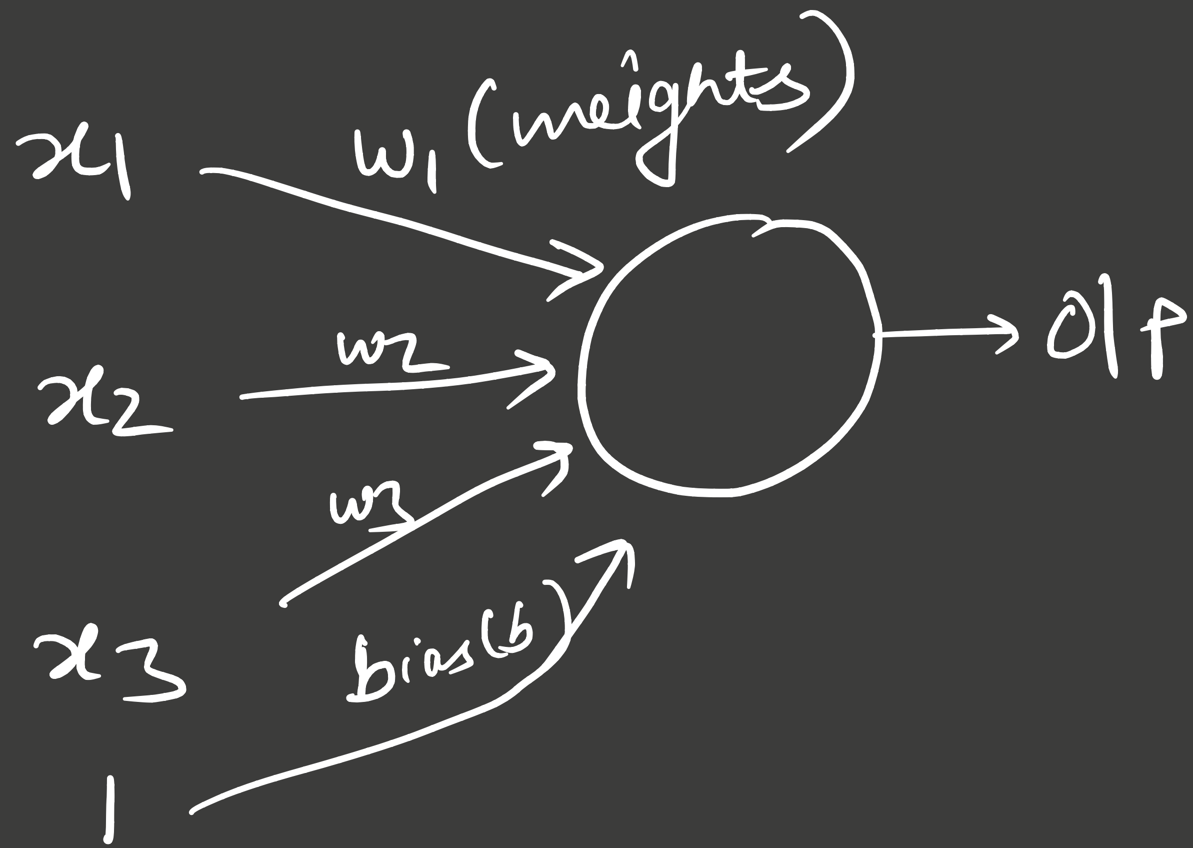


Perceptron



$$O/P = \begin{cases} 0 \; ; & \text{if } \sum w_i x_i \leq \text{threshold} \\ 1 \; ; & \sum w_i x_i > \text{threshold} \end{cases}$$

Perceptron



$$O/P = \begin{cases} 0 \; ; & \text{if } \sum w_i x_i + b \leq 0 \\ 1 \; ; & \sum w_i x_i + b > 0 \end{cases}$$

Perceptron

Q). For 2 i/p learn perceptron for binary AND.

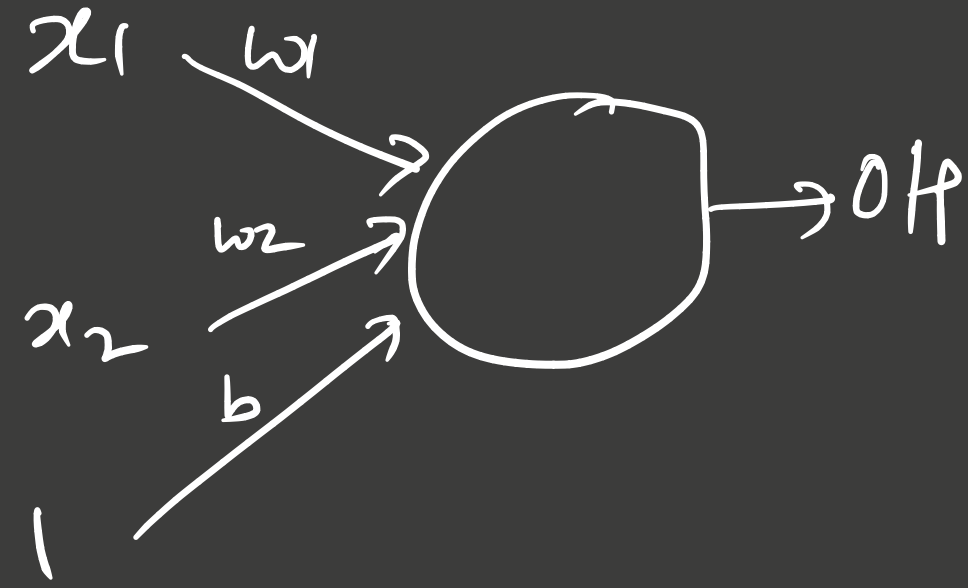
x_1	x_2	Out
0	0	0
0	1	0
1	0	0
1	1	1

Perceptron

Q). For 2 i/p learn perceptron for binary AND.

$w_1 = 1;\ w_2 = 1;\ b = -1.5$

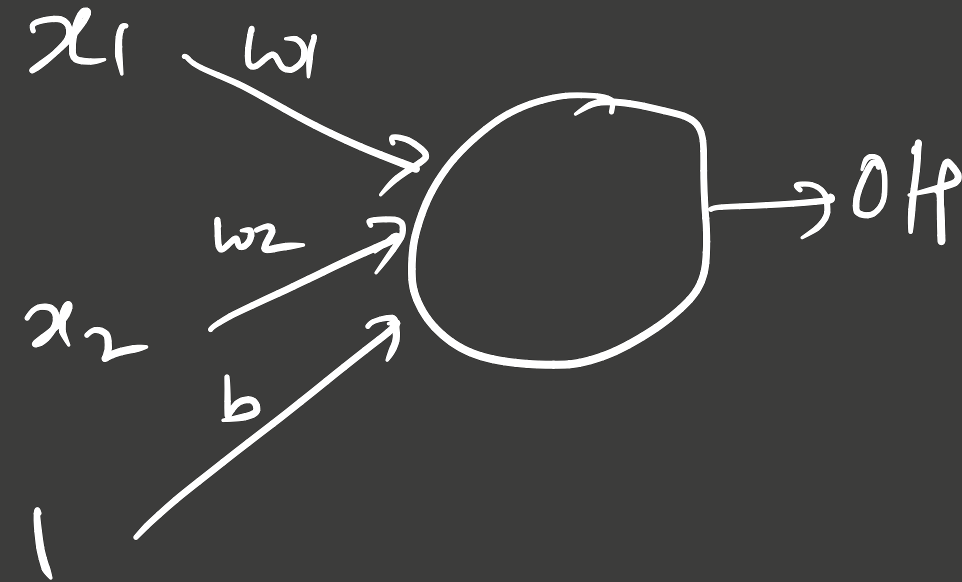
x_1	x_2	Out
0	0	0
0	1	0
1	0	0
1	1	1



Perceptron

Q). For 2 i/p learn perceptron for binary OR

x_1	x_2	Out
0	0	0
0	1	1
1	0	1
1	1	1

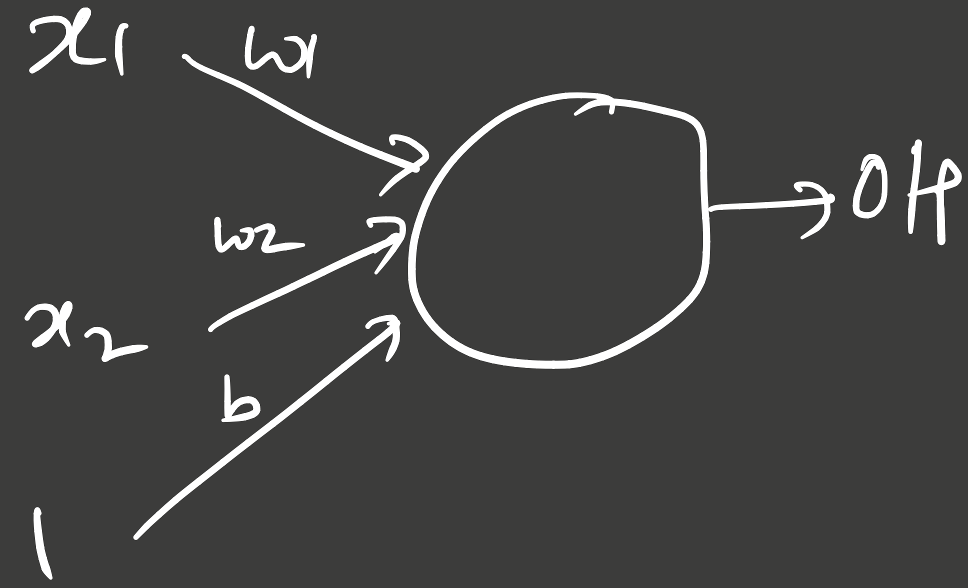


Perceptron

Q). For 2 i/p learn perceptron for binary OR

$w_1 = 1 \quad w_2 = 1 \; ; b = -0.5$

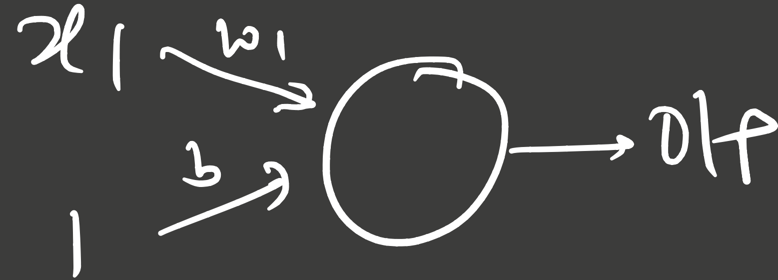
x_1	x_2	Out
0	0	0
0	1	1
1	0	1
1	1	1



Perceptron

Q). For 1 i/p learn perceptron for binary NOT

x_1	Out
0	1
1	0

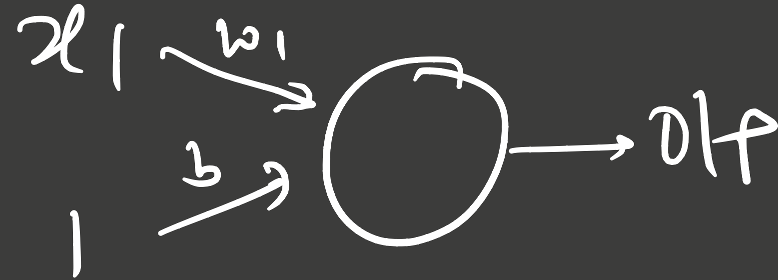


Perceptron

Q). For 1 i/p learn perceptron for binary NOT

$w_1 = -1 ; b = 0.5$

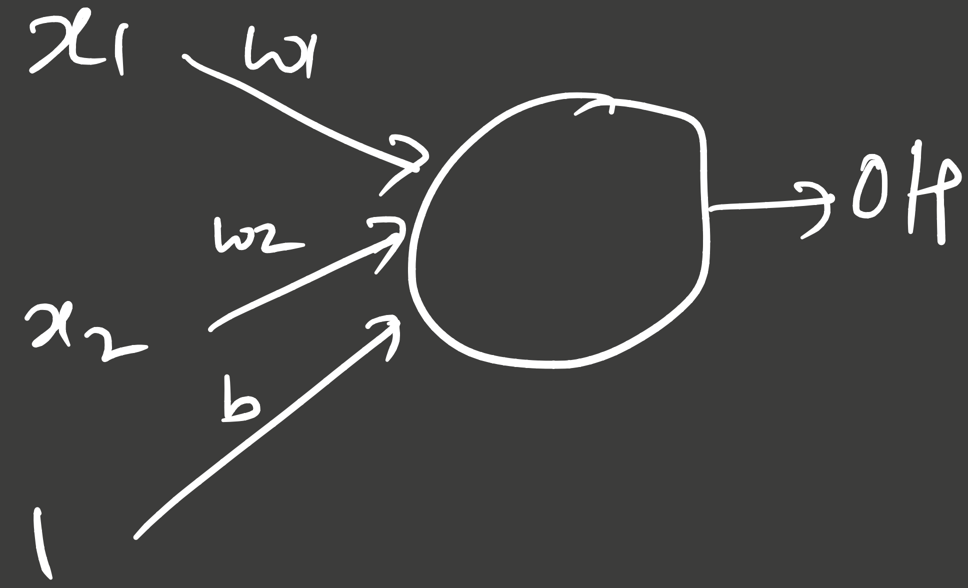
x_1	Out
0	1
1	0



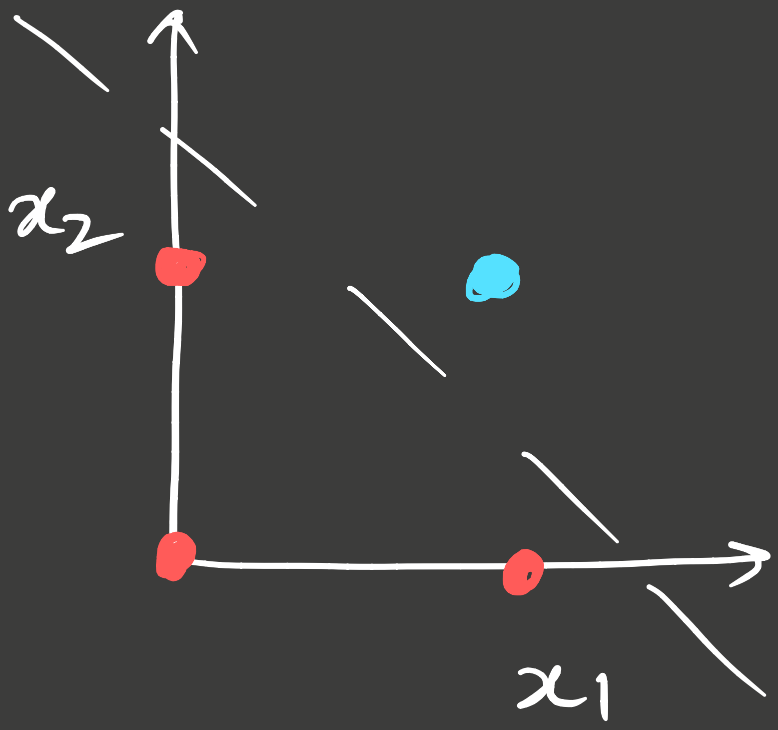
Perceptron

Q). For 2 i/p learn perceptron for binary XOR

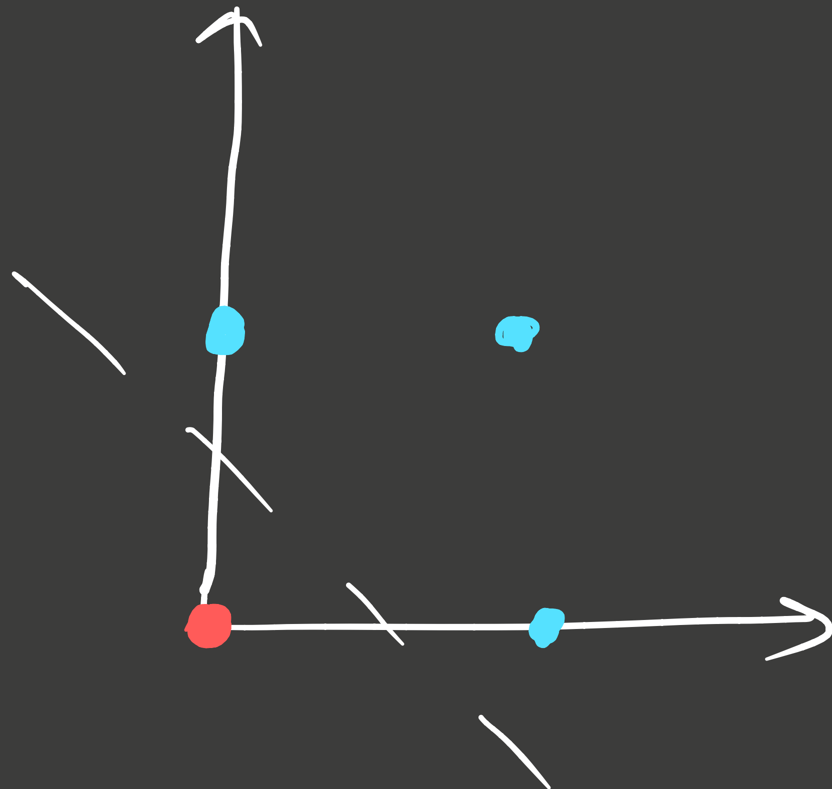
x_1	x_2	Out
0	0	0
0	1	1
1	0	1
1	1	0



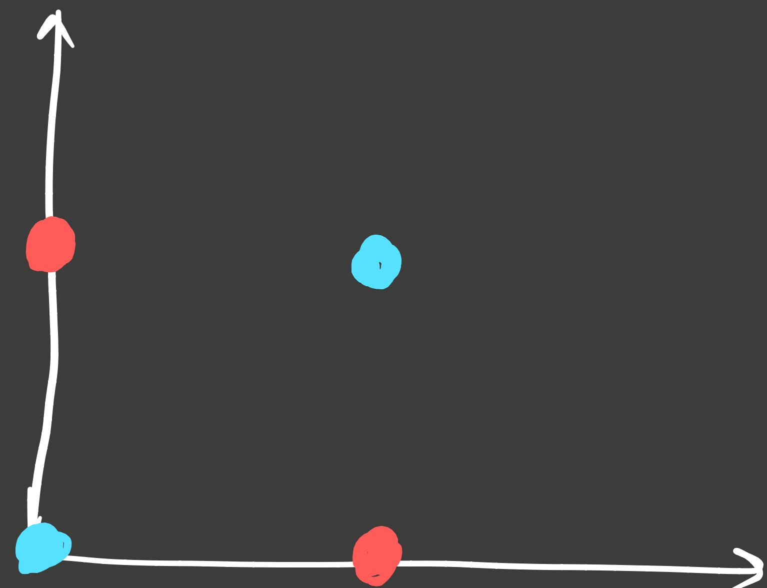
AND



OR



XOR



NON-LINEARLY SEPARABLE

Perceptron

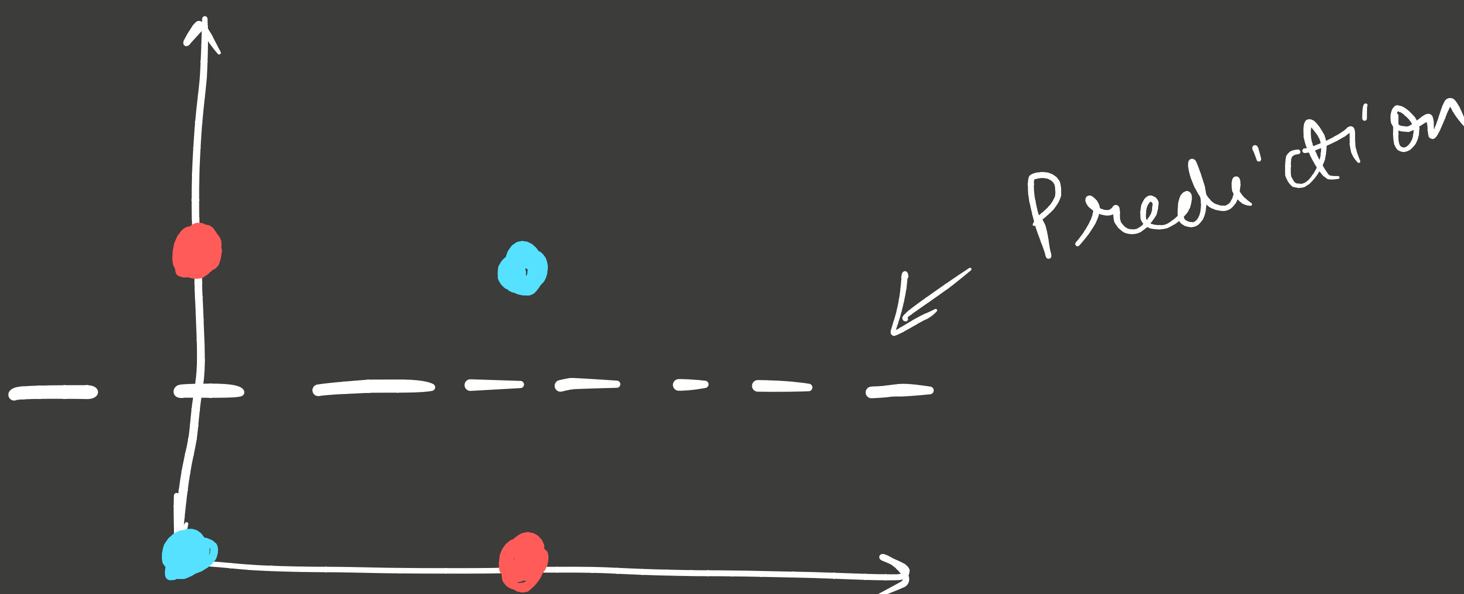
Cost funcⁿ

$$J(w_1, w_2, b) = \frac{1}{4}\sum_{i=1}^{4}(y_i - \hat{y}_i)^2$$

Optimum

$$w_1 = w_2 = 0$$

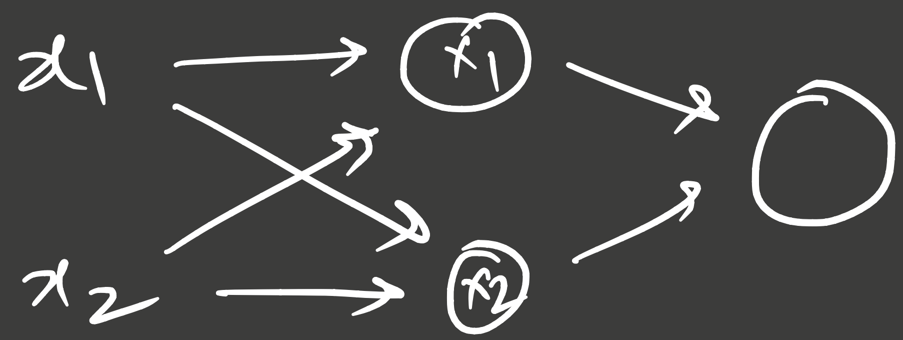
$$b = 1/2$$



Perceptron

Let's add more neurons to learn XOR

(Bias implicit)



Can this network of perceptrons learn XOR?

$$\hat{y} = w_1 X_1 + w_2 X_2 + b$$

$$X_1 = \mu_1 x_1 + \mu_2 x_2 + \mu_0$$

$$X_2 = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_0$$

$$\therefore \boxed{\hat{y} = \gamma_1 x_1 + \gamma_2 x_2 + \gamma_0}$$

Still linear!!

Need some non-linearity.

How?

Activation functions!

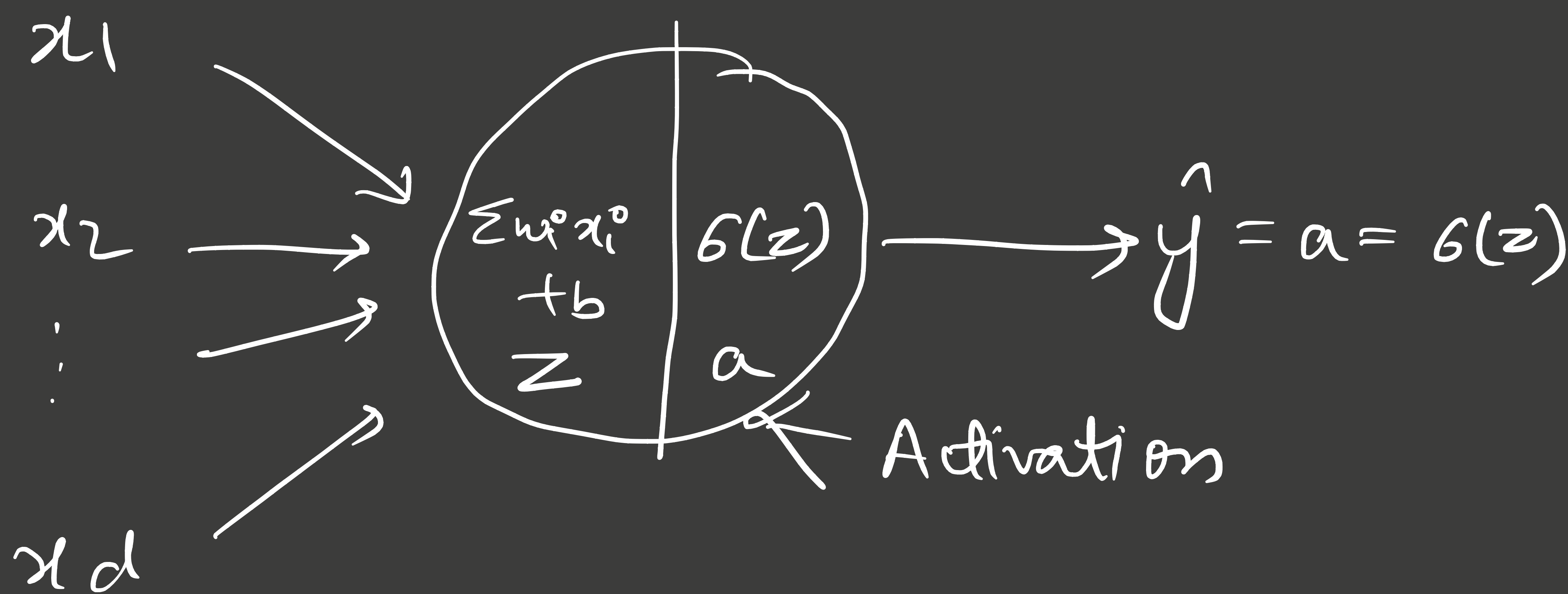
Activation Functions

1) Add non-linearity

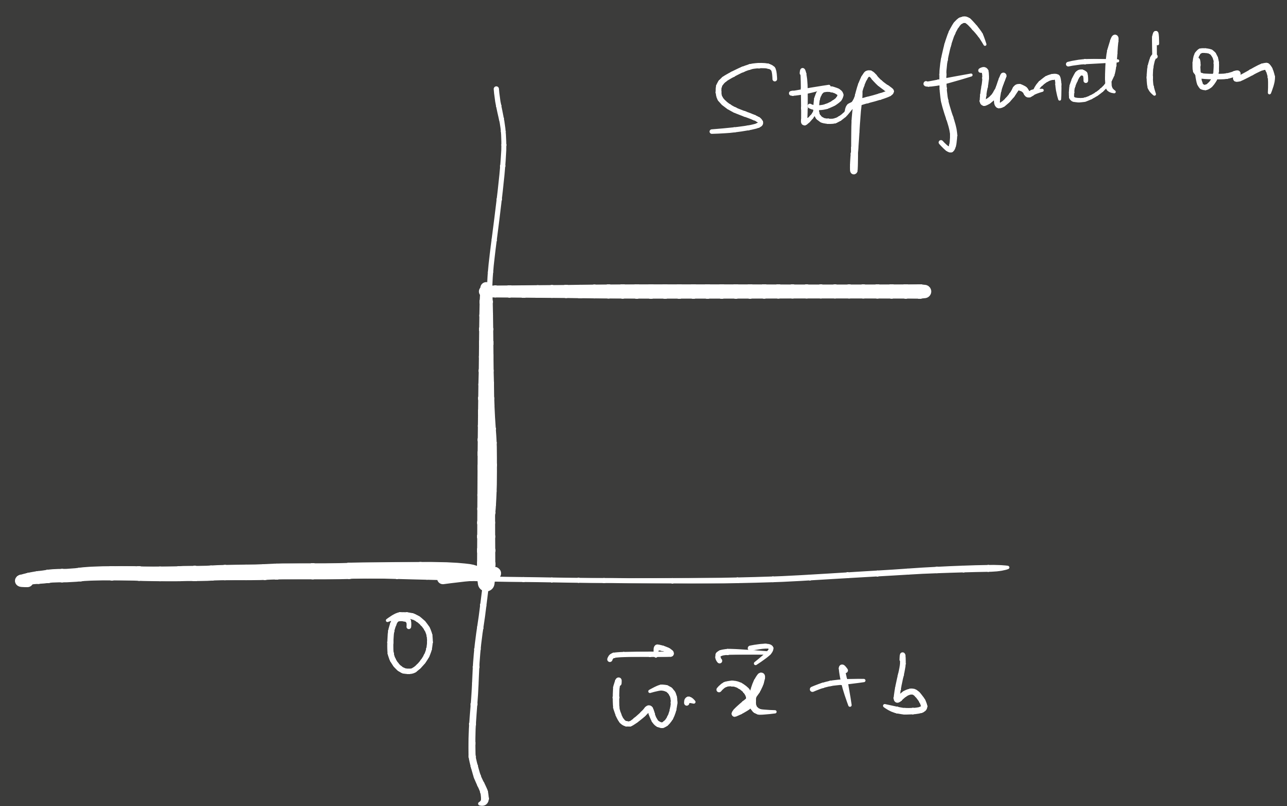
2) Ensures small change in weights / bias
   $\Rightarrow$ small change in o/p

   Desirable for learning

3) In some cases,
   maps $[-\infty, \infty] \rightarrow [a, b]$


x1
x2
xd
Σwixi
+b
Z
σ(z)
a
ŷ = a = σ(z)
Activation

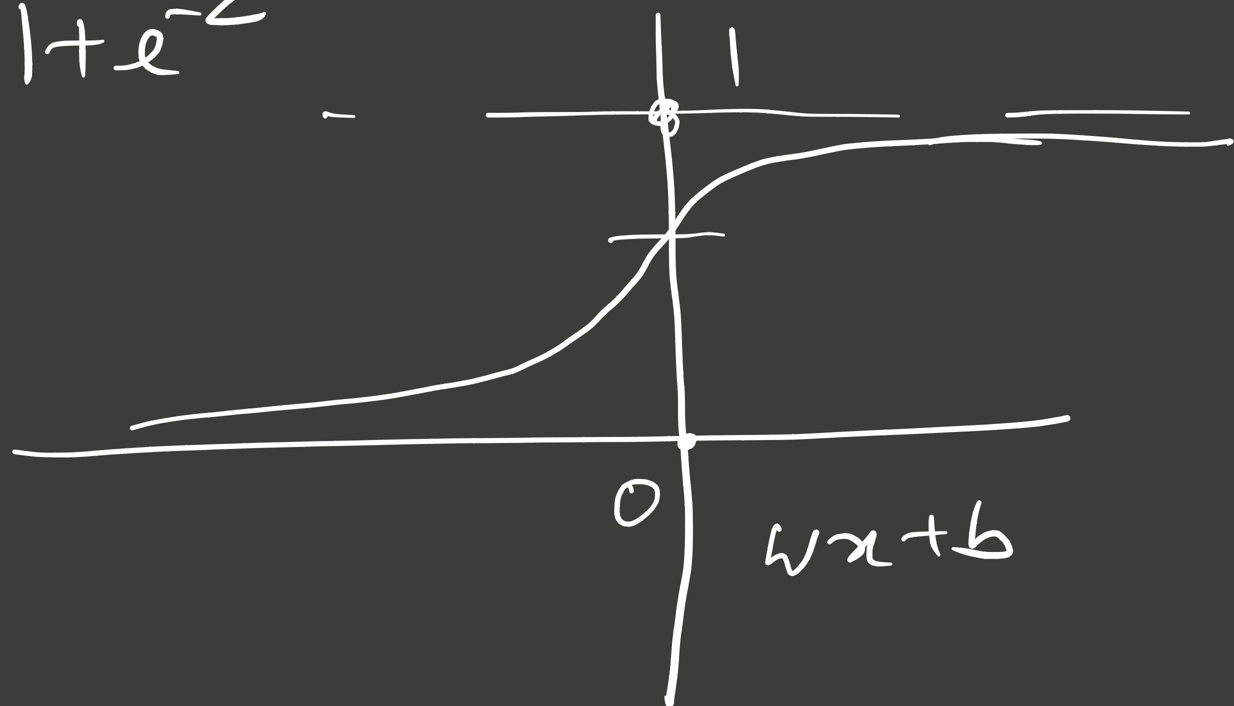
Perceptron Algebraic Form



Small change in w or b can lead to large change in o/p.

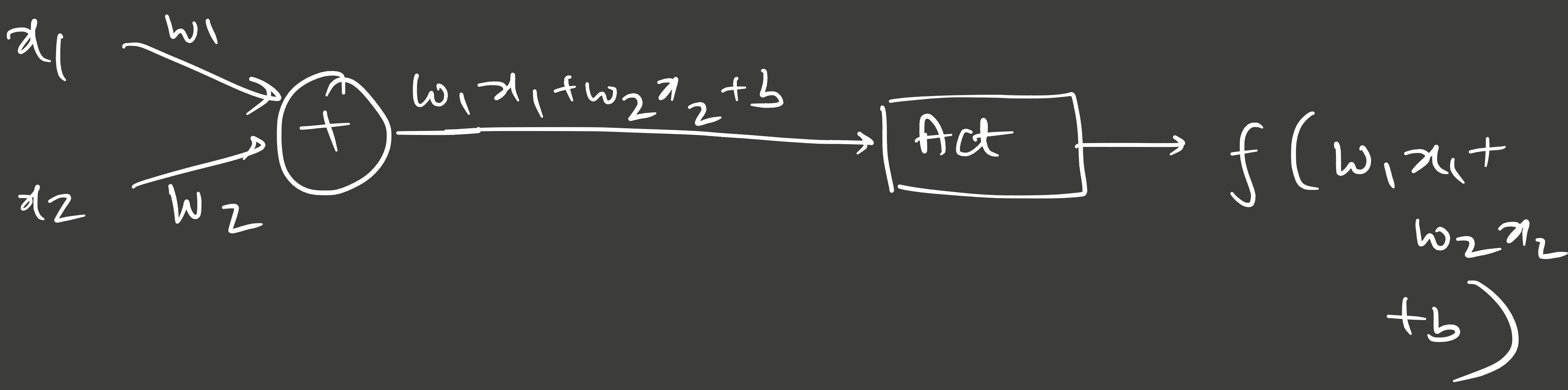
Sigmoid Neuron

* "Smoothed" out step function
* $\sigma(z) = 1/1+e^{-z}$



Small change in $w \Rightarrow$ Small change in Output

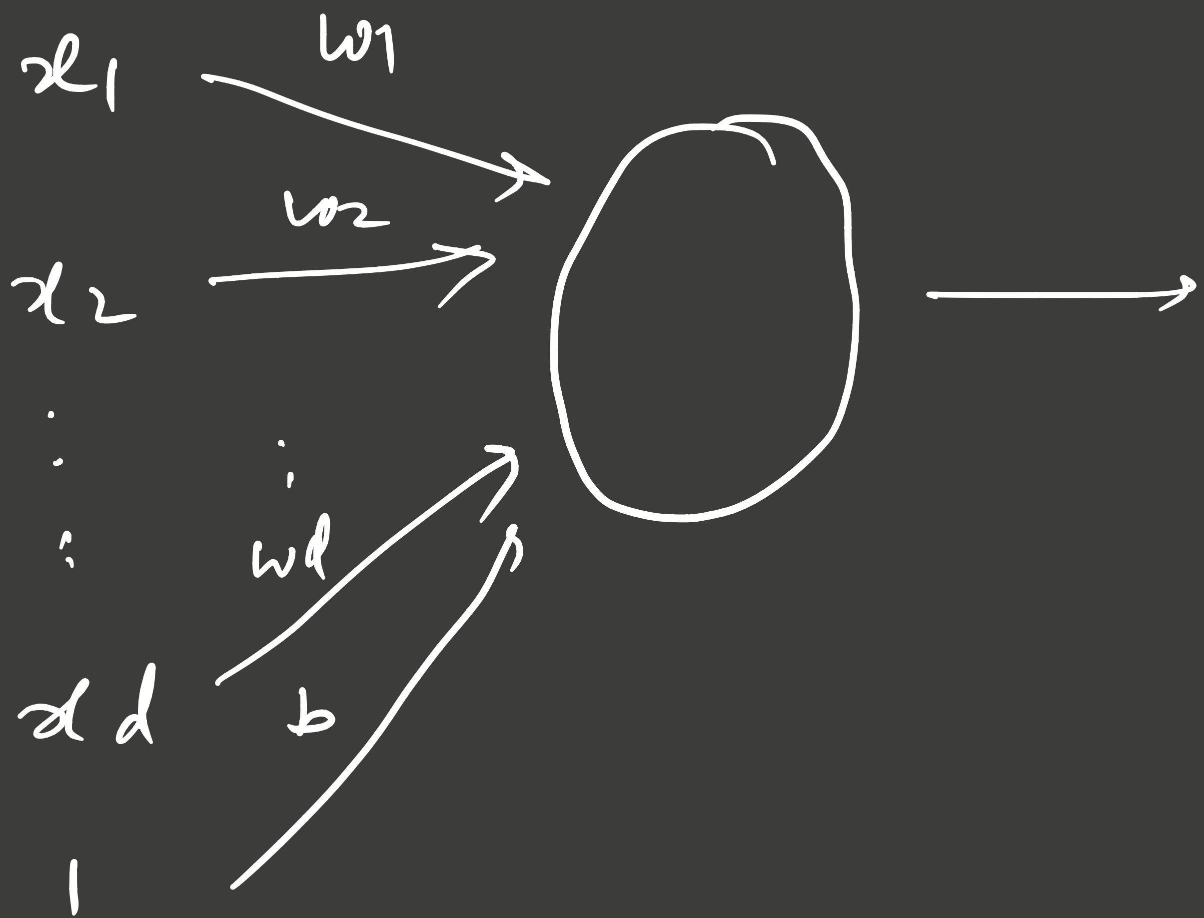
Activation Function



o/p is $f(wx+b)$

f is non-linear

SIGMOID UNIT



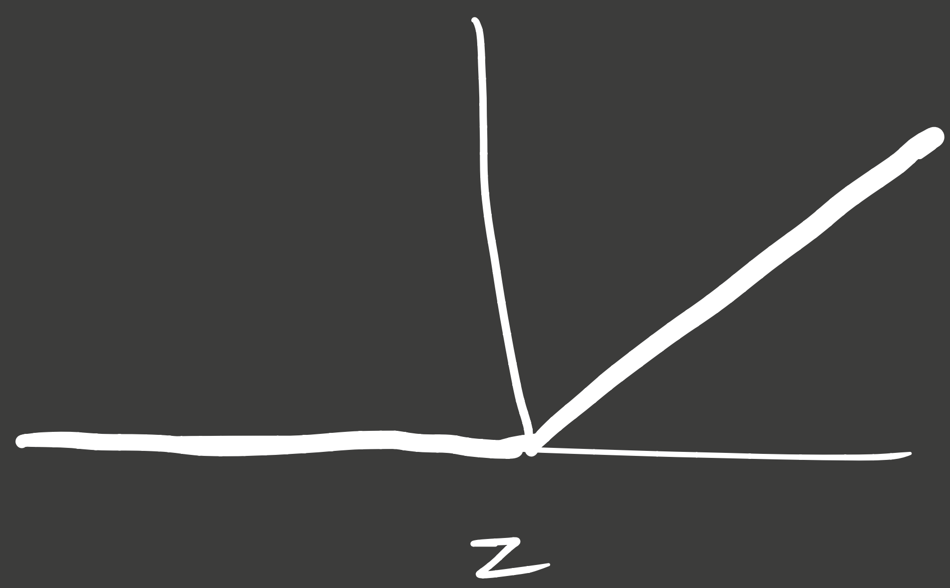
O/P = Activation $\left(\sum w_i x_i + b\right)$

$$= \sigma\left(\sum w_i x_i + b\right)$$

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

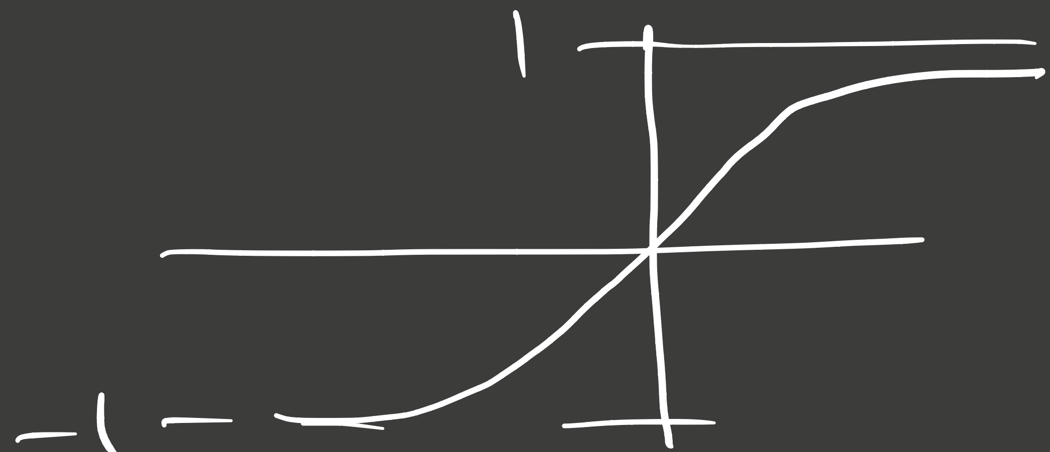
Other activation functions

Rectified Linear Unit (ReLU)



$f(z) = \max\{0, z\}$

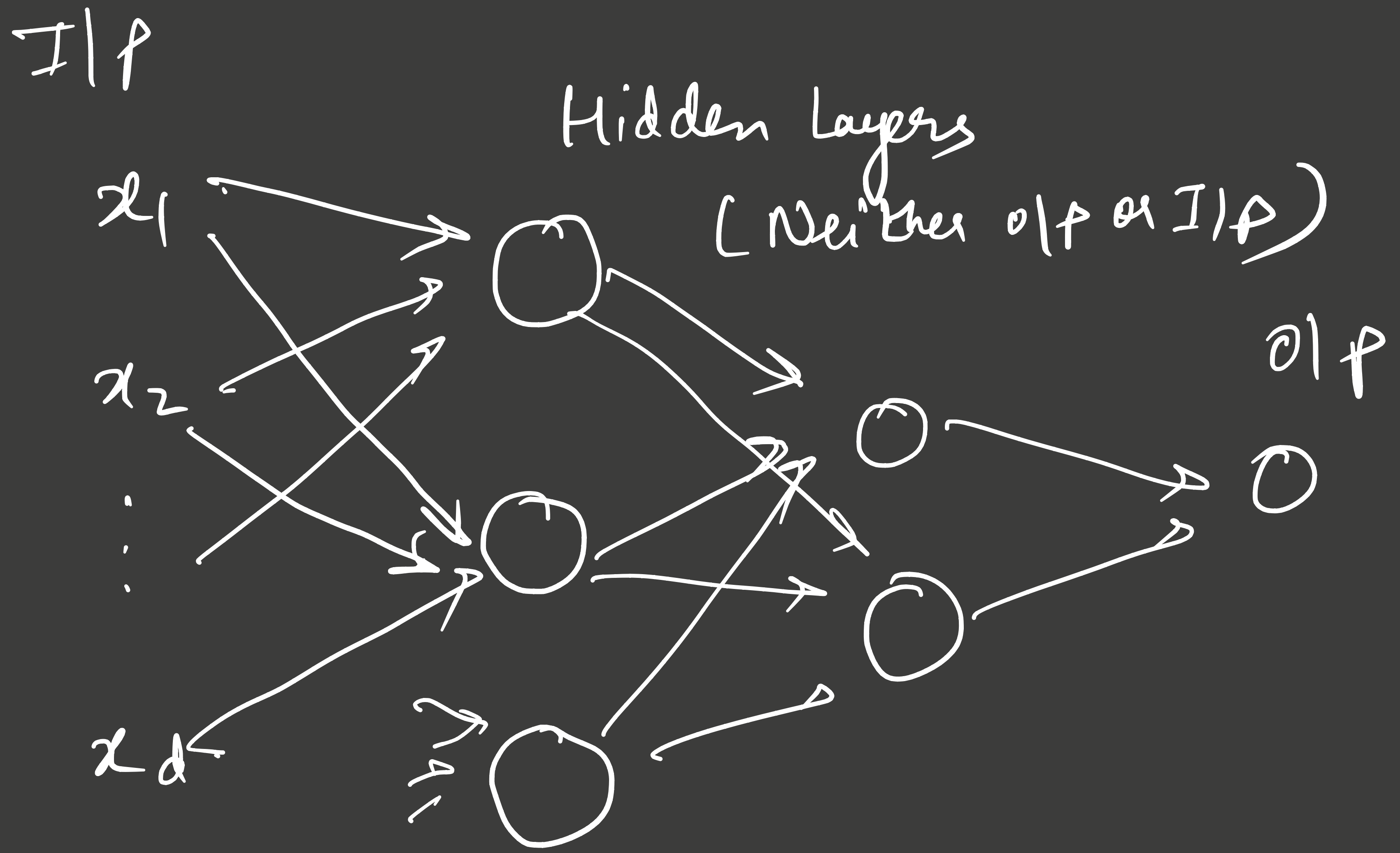
Tan H



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

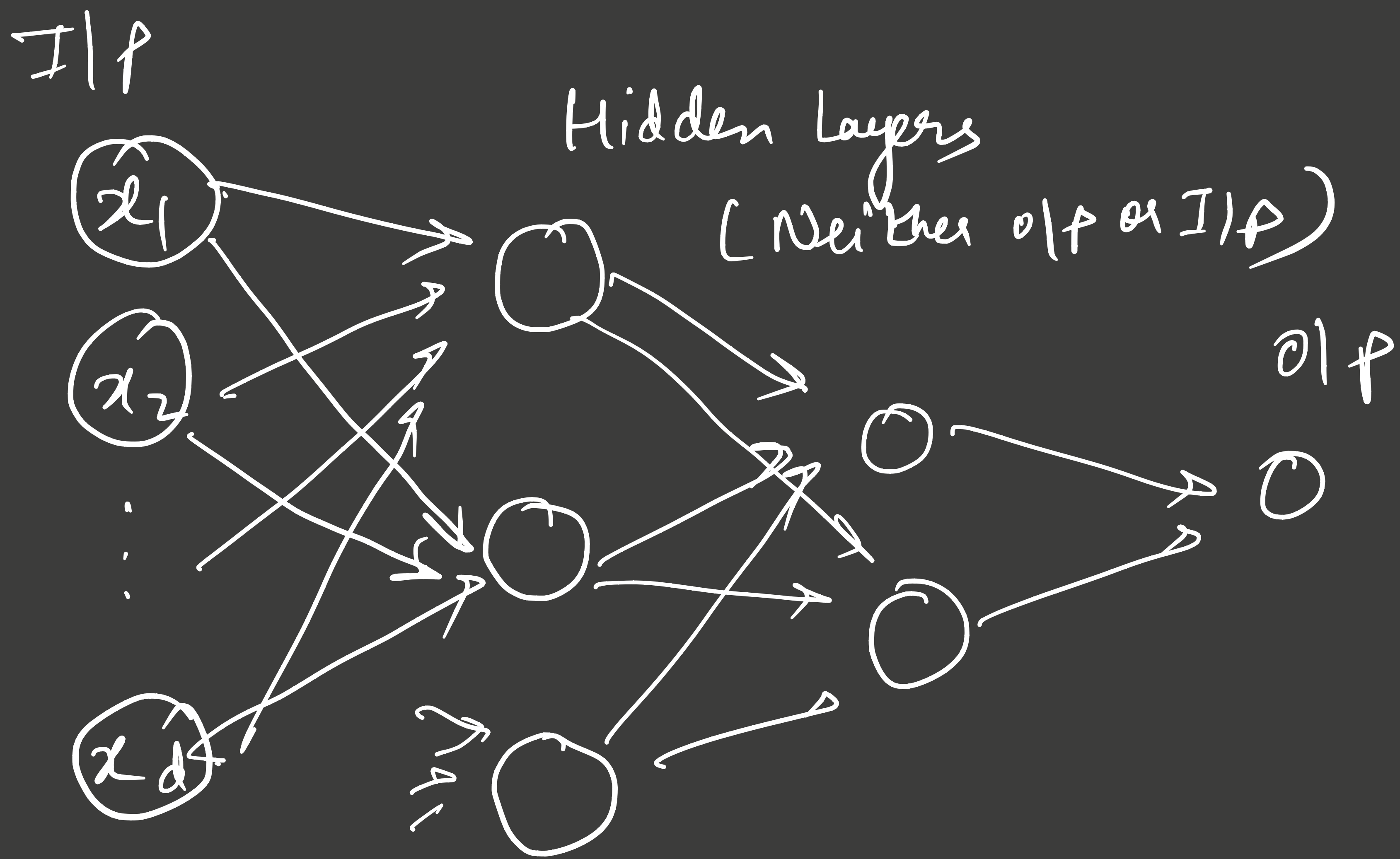
Simplest Neural Networks

MULTI LAYER PERCEPTRON (MLP)
(Layers of Sigmoid Units)



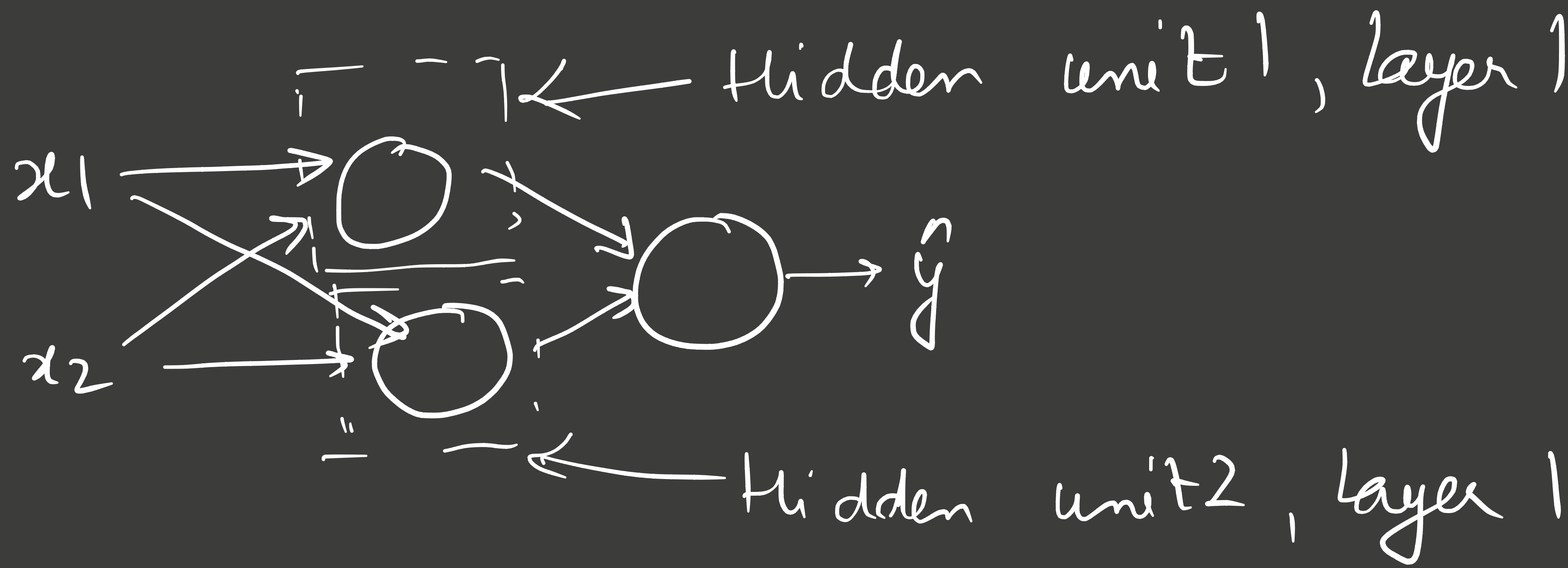
Simplest Neural Networks

MULTI LAYER PERCEPTRON (MLP)

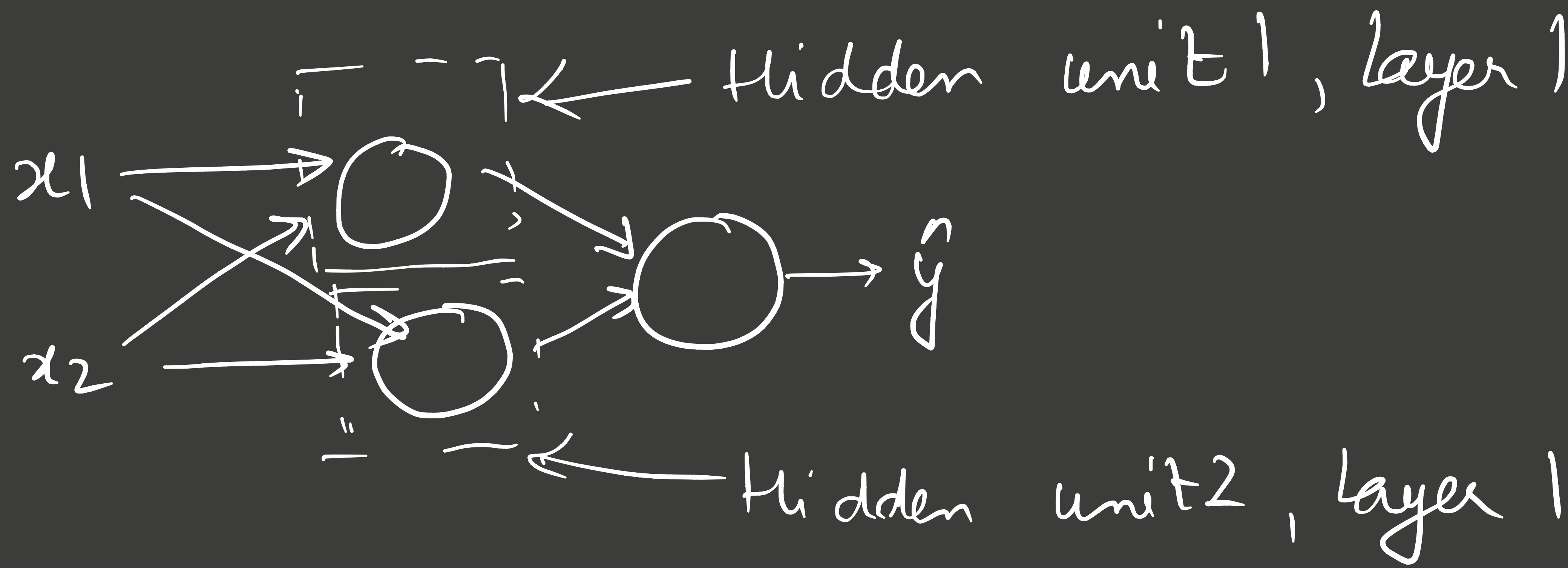


I/p also
a layer

XOR USING MLP



XOR USING MLP

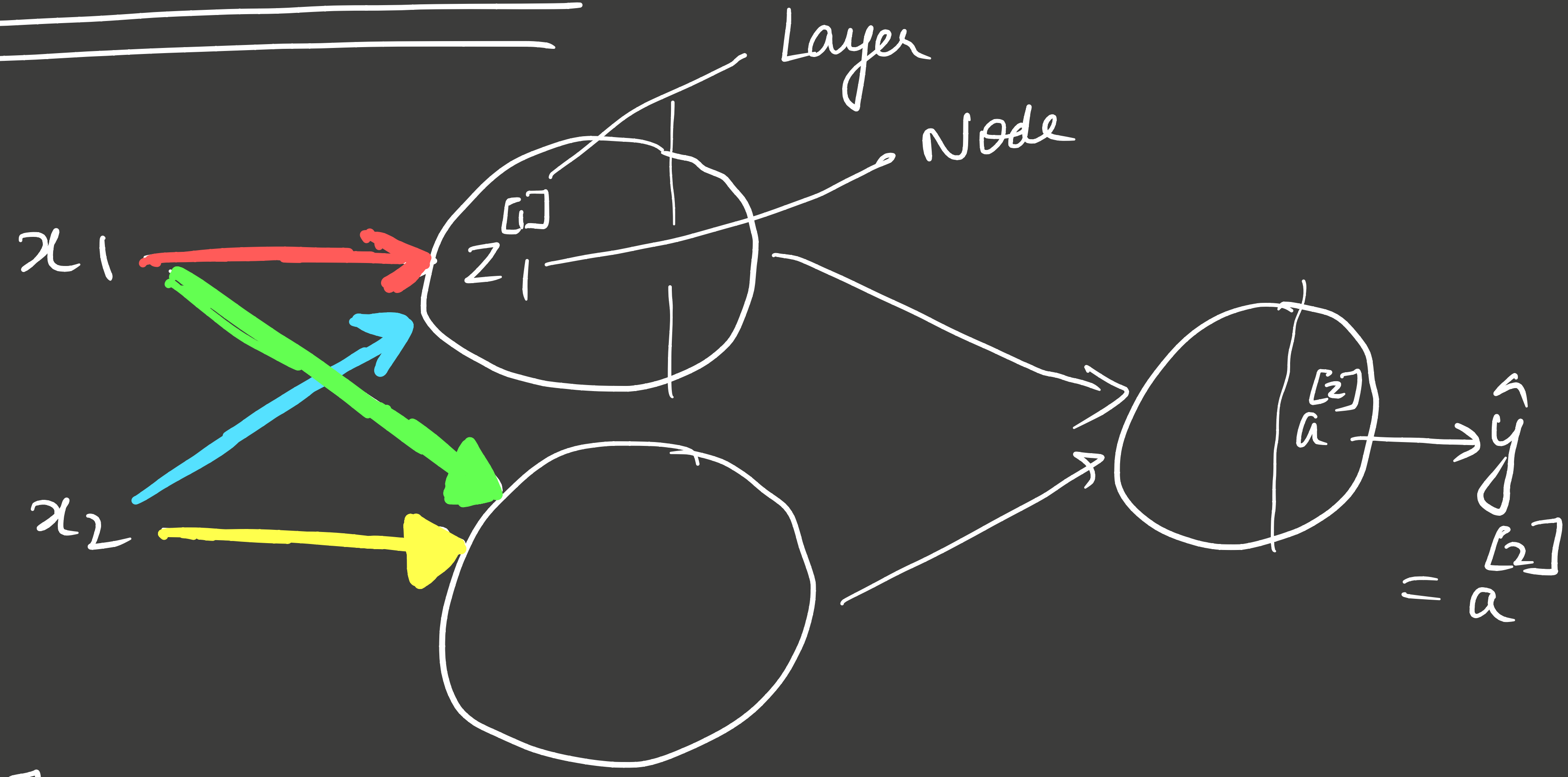


$\hat{y}$ = Activation of Layer 2 o/p

$= a^{[2]}$

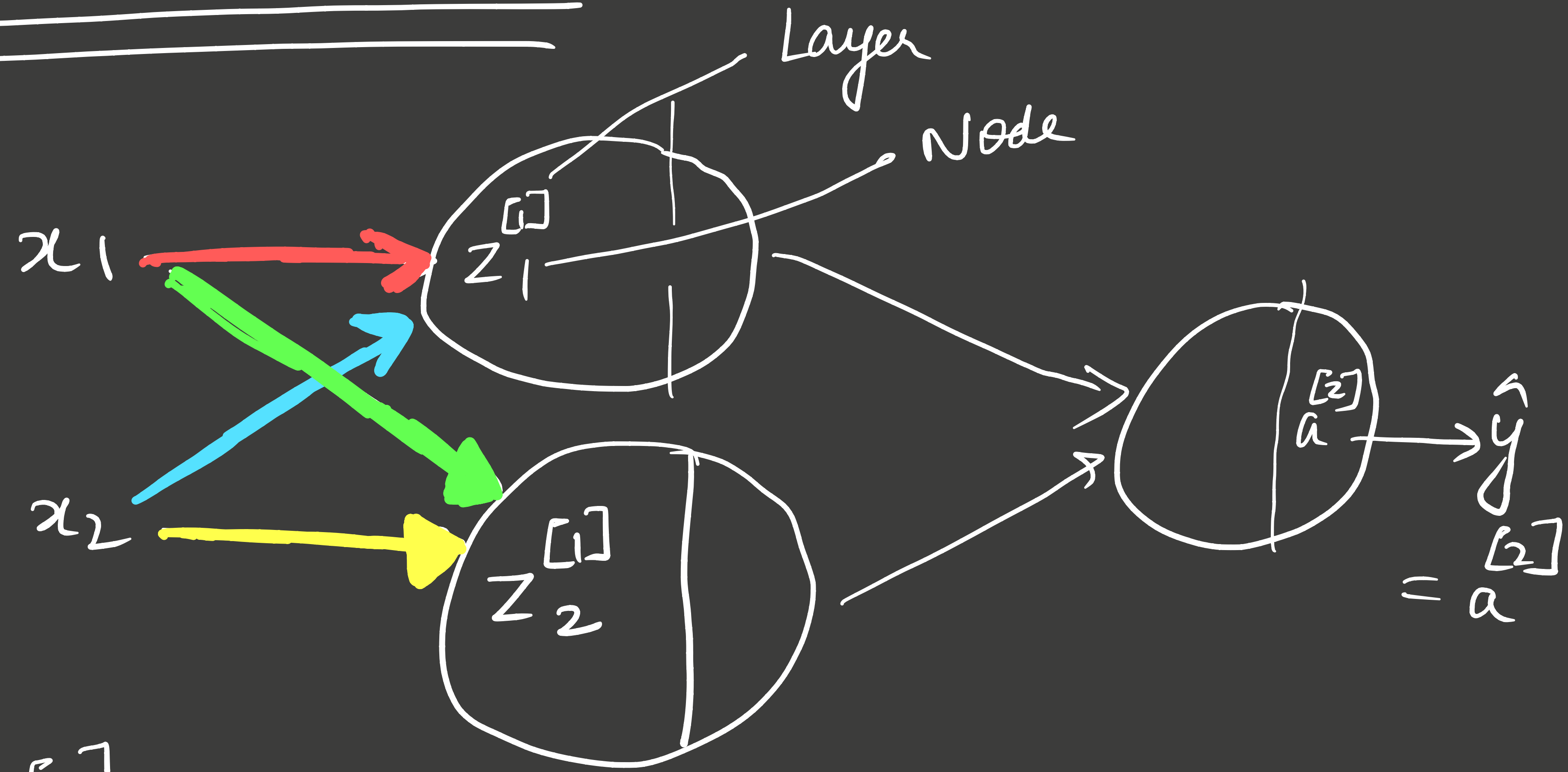
I/p = Activation of layer 0 $= a^{[0]} = [x_1 \; x_2]_{1\times 2}$

XOR USING MLP



$$Z_1^{[1]} = \underbrace{\begin{bmatrix} x_1 & x_2 \end{bmatrix}}_{a^{[0]}} \underbrace{\begin{bmatrix} \bullet \\ \bullet \end{bmatrix}}_{w_1^{[1]}} + b_1^{[1]}$$

XOR USING MLP



$$z_1^{[1]} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} \bullet \\ \bullet \end{bmatrix} + b_1^{[1]}$$

$a^{[0]}$ $w_1^{[1]}$

$$z_2^{[1]} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} \bullet \\ \bullet \end{bmatrix} + b_2^{[1]}$$

$a^{[0]}$ $w_2^{[1]}$

XOR USING MLP

$$x = [x_1 \quad x_2] = a^{[0]}$$

$$\underset{1\times1}{z_1^{[1]}} = \underset{1\times2}{a^{[0]}} \; \underset{2\times1}{w_1^{[1]}} + \underset{1\times1}{b_1^{[1]}}$$

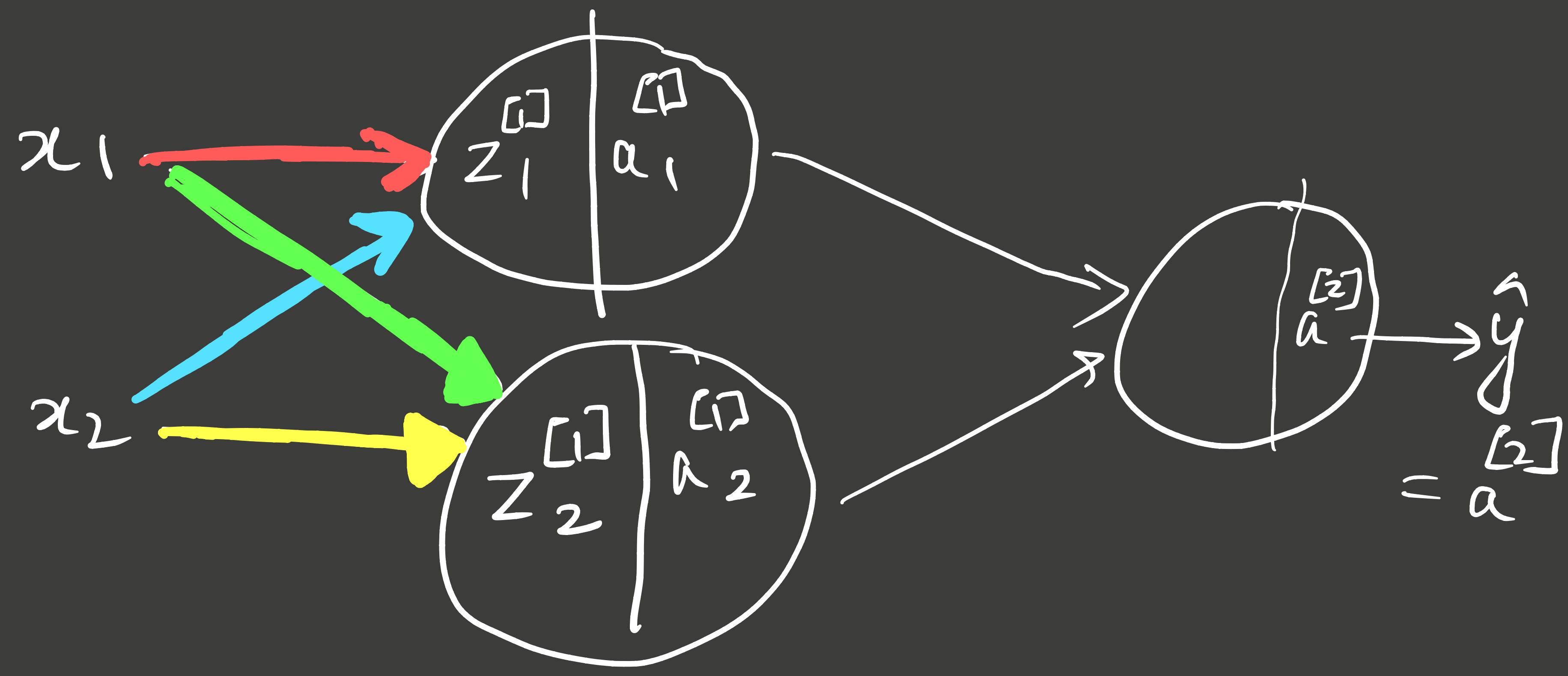
$$\underset{1\times1}{z_2^{[1]}} = \underset{1\times2}{a^{[0]}} \; \underset{2\times1}{w_2^{[1]}} + \underset{1\times1}{b_2^{[1]}}$$

$$\underset{1\times2}{\begin{bmatrix} z_1^{[1]} & z_2^{[1]} \end{bmatrix}} = \underset{1\times2}{a^{[0]}} \; \underset{2\times2}{\begin{bmatrix} w_1^{[1]} & w_2^{[1]} \end{bmatrix}} + \underset{1\times2}{\begin{bmatrix} b_1^{[1]} & b_2^{[1]} \end{bmatrix}}$$

XOR USING MLP

$$\underset{1\times 2}{Z^{[1]}} = \underset{1\times 2}{a^{[0]}}\ \underset{2\times 2}{w^{[1]}} + \underset{1\times 2}{b^{[1]}}$$

XOR USING MLP



$$a_1^{[1]} = \sigma\left(z_1^{[1]}\right)$$

$$a_2^{[1]} = \sigma\left(z_2^{[1]}\right)$$

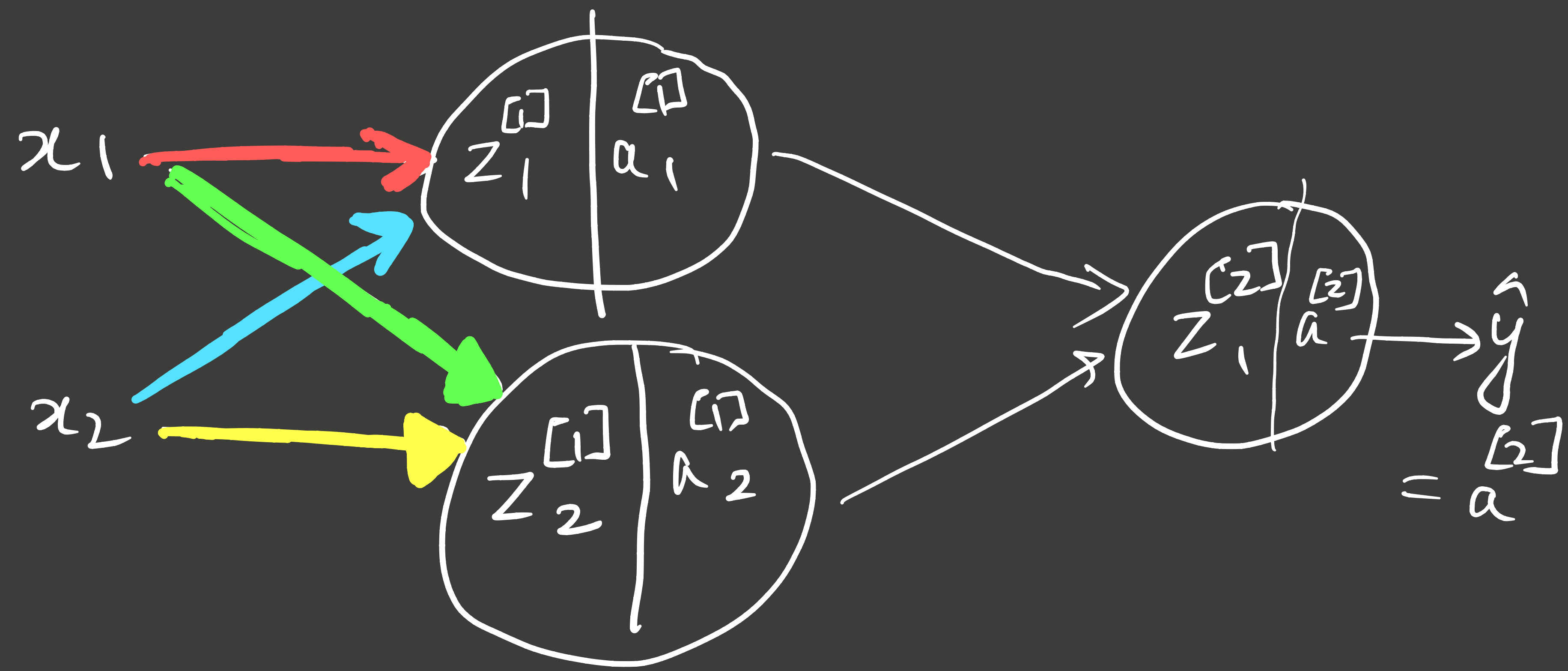
XOR USING MLP

$$a^{[1]} = \begin{bmatrix} a_1^{[1]} & a_2^{[1]} \end{bmatrix} = \sigma\left(z^{[1]}\right)$$

$$a_1^{[1]} = \sigma\left(z_1^{[1]}\right)$$

$$a_2^{[1]} = \sigma\left(z_2^{[1]}\right)$$

XOR USING MLP



$$z^{[2]} = a^{[1]} w^{[2]} + b^{[2]}$$

$$a^{[2]} = \sigma\left(z^{[2]}\right)$$

XOR USING MLP

ONLY 1 POINT

$x_1 = 1;\ x_2 = 1;\ y = 0$

$\sigma = \text{RELU}$

$W^{[1]} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix};\ b^{[1]} = \begin{bmatrix} 0 & -1 \end{bmatrix};\ W^{[2]} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

$b^{[2]} = 0$

XOR USING MLP

$$x_1 = 1;\ x_2 = 1;\ y = 0 \implies a^{[0]} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$w^{[1]} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix};\quad b^{[1]} = \begin{bmatrix} 0 & -1 \end{bmatrix}$$

$$z^{[1]} = a^{[0]} w^{[1]} + b^{[1]} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 1 \end{bmatrix}$$

$$a^{[1]} = \sigma\left(z^{[1]}\right) = \max\left\{[0, 0], [2, 1]\right\} = [2, 1]$$

XOR USING MLP

$x_1 = 1;\ x_2 = 1;\ y = 0 \Rightarrow a^{[0]} = [1 \quad 1]$

$a^{[1]} = [2 \quad 1]$ $\qquad$ $w^{[2]} = [1 \quad -2]\ ;\ b^{[2]} = 0$

$$z^{[2]} = [2 \quad 1]\begin{bmatrix} 1 \\ -2 \end{bmatrix} + 0 = 0$$

$$a^{[2]} = \sigma\left(z^{[2]}\right) = 0$$

$$\therefore\ \hat{y}(1, 1) = a^{[2]} = 0$$

XOR USING MLP

Let's redo for $x_1 = 0$; $x_2 = 1$; $y_{TRUE} = 1$

$$W^{[1]} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}; \quad b^{[1]} = \begin{bmatrix} 0 & -1 \end{bmatrix}; \quad W^{[2]} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}; \quad b^{[2]} = 0$$

XOR USING MLP

let's redo for $x_1 = 0$; $x_2 = 1$; $y_{TRUE} = 1$

$$w^{[1]} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}; \quad b^{[1]} = [0 \quad -1]; \quad w^{[2]} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}; \quad b^{[2]} = 0$$

$$z^{[1]} = a^{[0]} w^{[1]} + b^{[1]} = [0 \quad 1] \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + [0 \quad -1]$$

$$= [1 \quad 1] + [0 \quad -1] = [1 \quad 0]$$

$$a^{[1]} = \sigma\left(z^{[1]}\right) = \text{MAX}\{[0,0], [1,0]\} = [1 \quad 0]$$

XOR USING MLP

Let's redo for $x_1 = 0$; $x_2 = 1$; $y_{TRUE} = 1$

$w^{[1]} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$; $b^{[1]} = [0 \quad -1]$; $w^{[2]} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$; $b^{[2]} = 0$

$$a^{[1]} = [1 \quad 0]$$

$$z^{[2]} = a^{[1]} w^{[2]} + b^{[2]} = [1 \quad 0] \begin{bmatrix} 1 \\ -2 \end{bmatrix} + 0$$

$$= 1$$

$$a^{[2]} = \sigma\left[z^{[2]}\right] = 1$$

$$\therefore \hat{y}(0,1) = a^{[2]} = 1 = y_{TRUE}$$

COMPUTATION FOR 'M' instances

$$X = \begin{bmatrix} - & x_1 & - \\ - & x_2 & - \\ & \cdots & \\ - & x_M & - \end{bmatrix}_{M \times d}$$ where $x_i \in R^d$

$$X = \begin{bmatrix} a^{[0](1)} \\ a^{[0](2)} \\ \vdots \\ a^{[0](M)} \end{bmatrix}$$

(i) denotes instance #

COMPUTATION FOR 'M' instances

$$X = \begin{bmatrix} - x_1 - \\ - x_2 - \\ \cdots \\ - x_M - \end{bmatrix}_{M \times d} \quad \text{where } x_i \in \mathbb{R}^d$$

$$X = \begin{bmatrix} a^{[0](1)} \\ a^{[0](2)} \\ \vdots \\ a^{[0](M)} \end{bmatrix}$$

(i) denotes instance #

$$z^{[1](1)} = a^{[0](1)} w^{[1]} + b^{[1]}$$

$$z^{[1](2)} = a^{[0](2)} w^{[1]} + b^{[1]} \Rightarrow Z^{[1]} = A^{[0]} w^{[1]} + b^{[1]}$$

$$\cdots\cdots$$

$$A^{[0]} = \begin{bmatrix} a^{[0](1)} \\ \vdots \\ a^{[0](m)} \end{bmatrix}$$

MLP FOR XOR (OVER 'm' SAMPLES)

$$X = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} ; \quad y_{TRUE} = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^T$$

$$W^{[1]} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} ; \quad b^{[1]} = \begin{bmatrix} 0 & -1 \end{bmatrix}$$

$$\therefore Z^{[1]} = \underset{4\times2}{A^{[0]}} \; \underset{2\times2}{W^{[1]}} + \underset{1\times2}{b^{[1]}} \leftarrow \text{Broadcasted}$$

$$Z^{[1]} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & -1 \end{bmatrix}$$

MLP FOR XOR (OVER 'm' SAMPLES)

$$X = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \; ; \; y_{TRUE} = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^T$$

$$Z^{[1]} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

$$A^{[1]} = \sigma\left(Z^{[1]}\right) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

MLP FOR XOR (OVER 'm' SAMPLES)

$$X = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \; ; \; y_{TRUE} = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^T$$

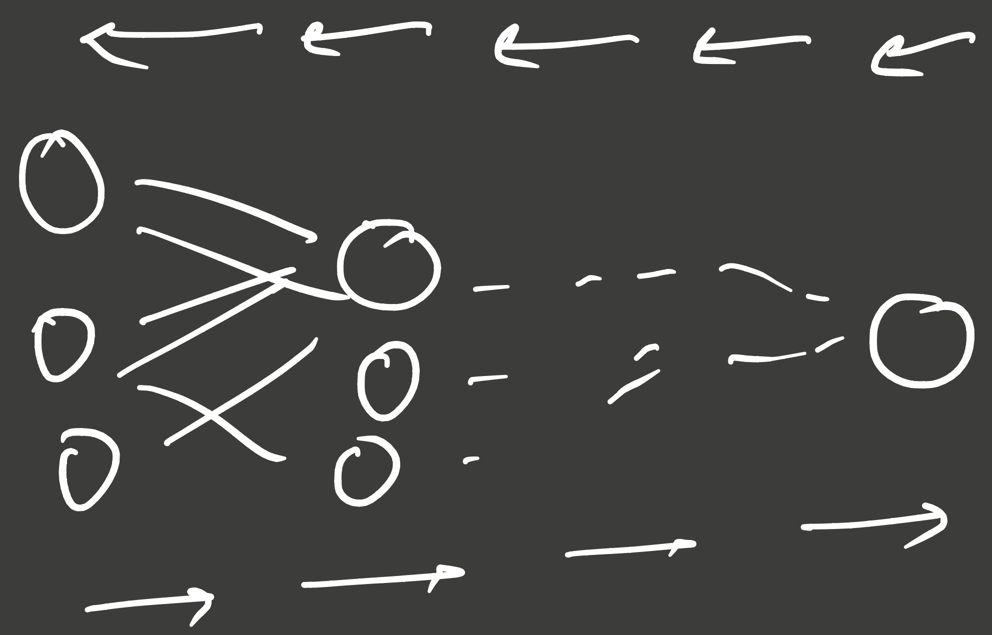
$$A^{[1]} = \sigma\left(Z^{[1]}\right) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

$$W^{[2]} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \; ; \; b^{[2]} = 0$$

$$\therefore Z^{[2]} = A^{[1]} W^{[2]} + b^{[2]} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \end{bmatrix} + 0 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$= y_{TRUE}$$

FORWARD & BACKWARD PROPAGATION

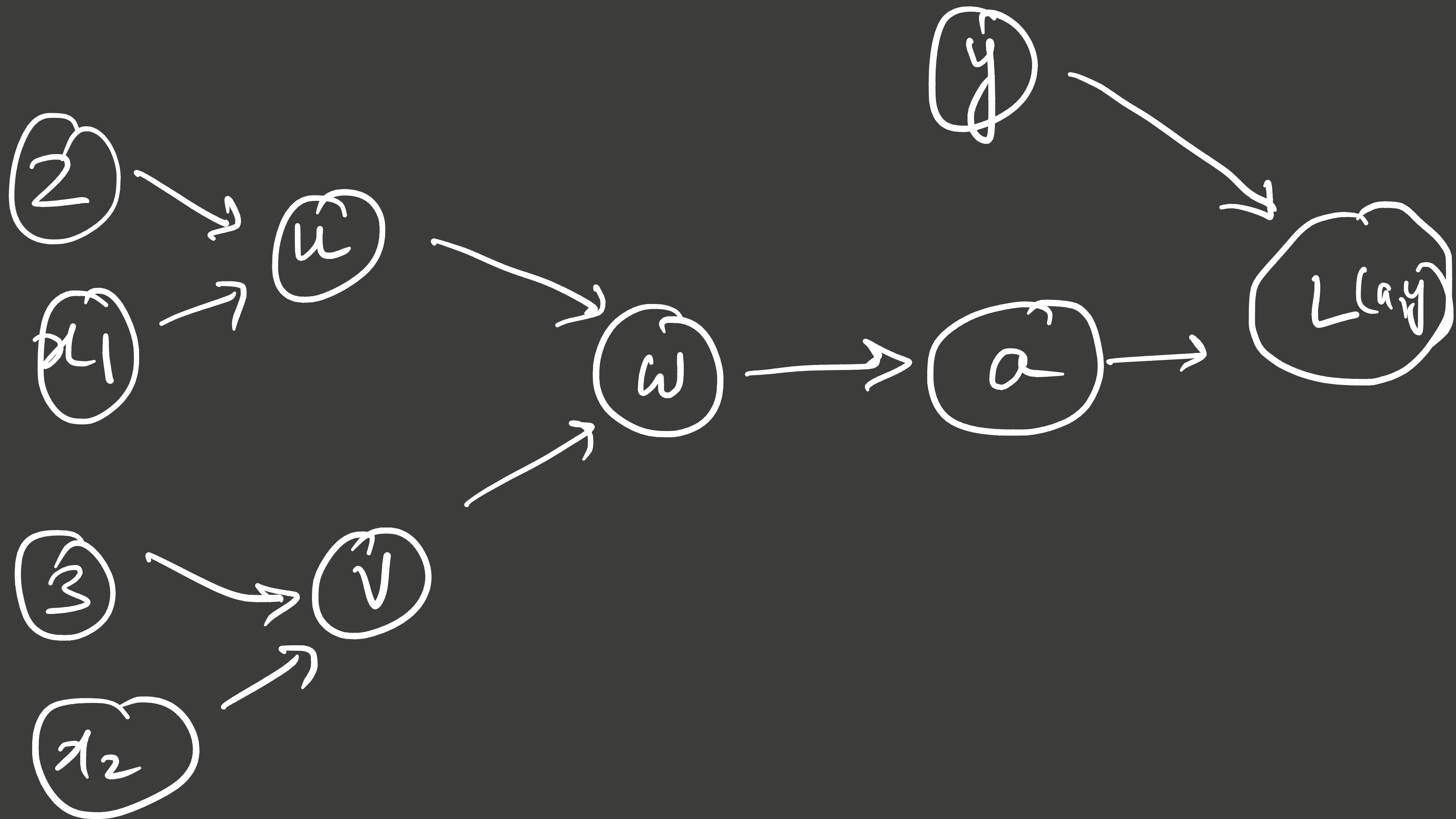


Backward propagation
compute derivatives
&
update
weights

Forward propagation
compute $\hat{y}$

COMPUTATION GRAPH

$\hat{y} = \sigma(2x_1 + 3x_2)$; LOSS $= L(\hat{y}, y)$

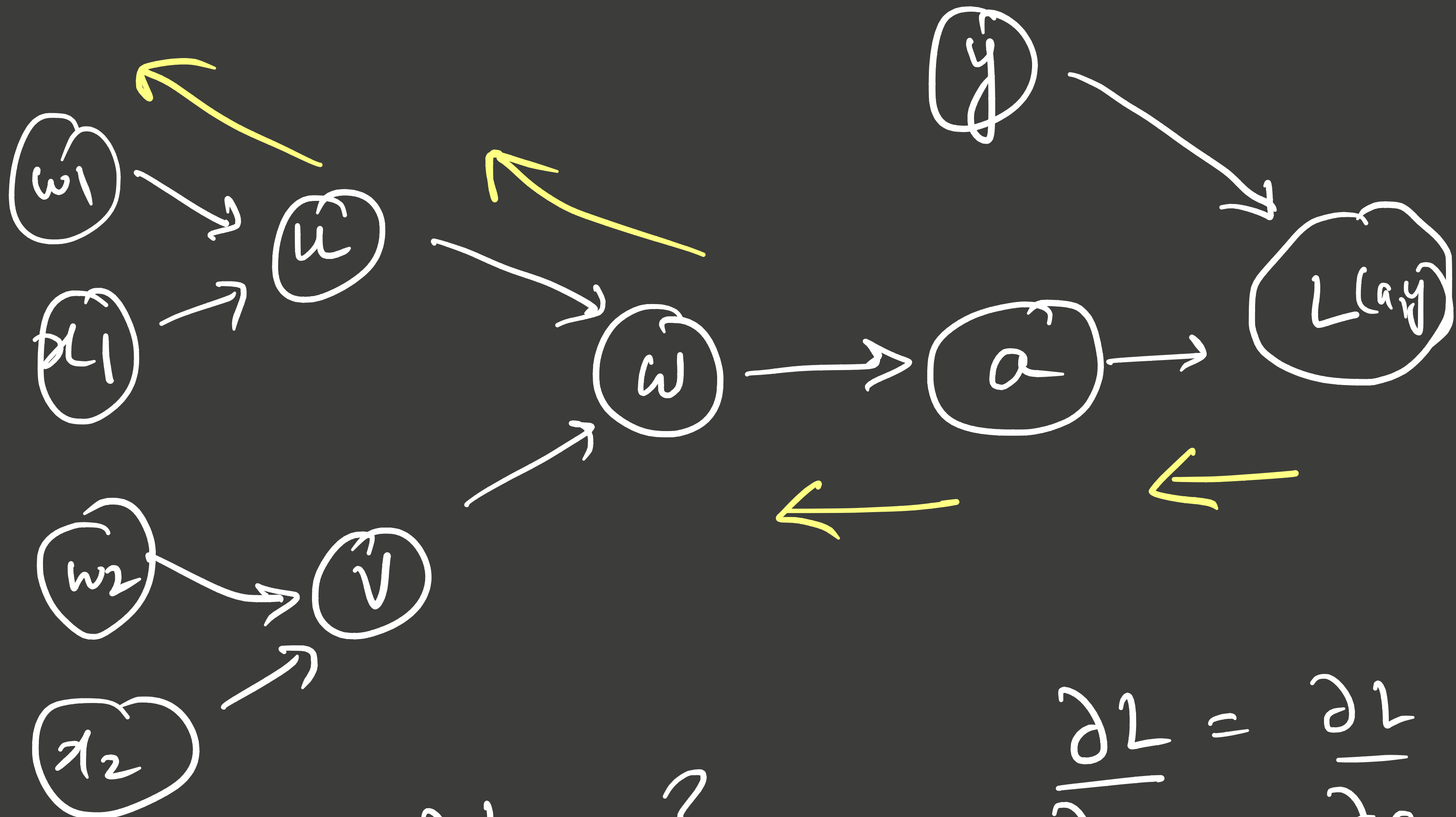


$u = 2 * x_1$; $v = 3 * x_2$

$w = u + v$; $a = \sigma(w)$

COMPUTATION GRAPH

$$\hat{y} = \sigma(w_1 x_1 + w_2 x_2) \; ; \; \text{LOSS} = L(\hat{y}, y)$$



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

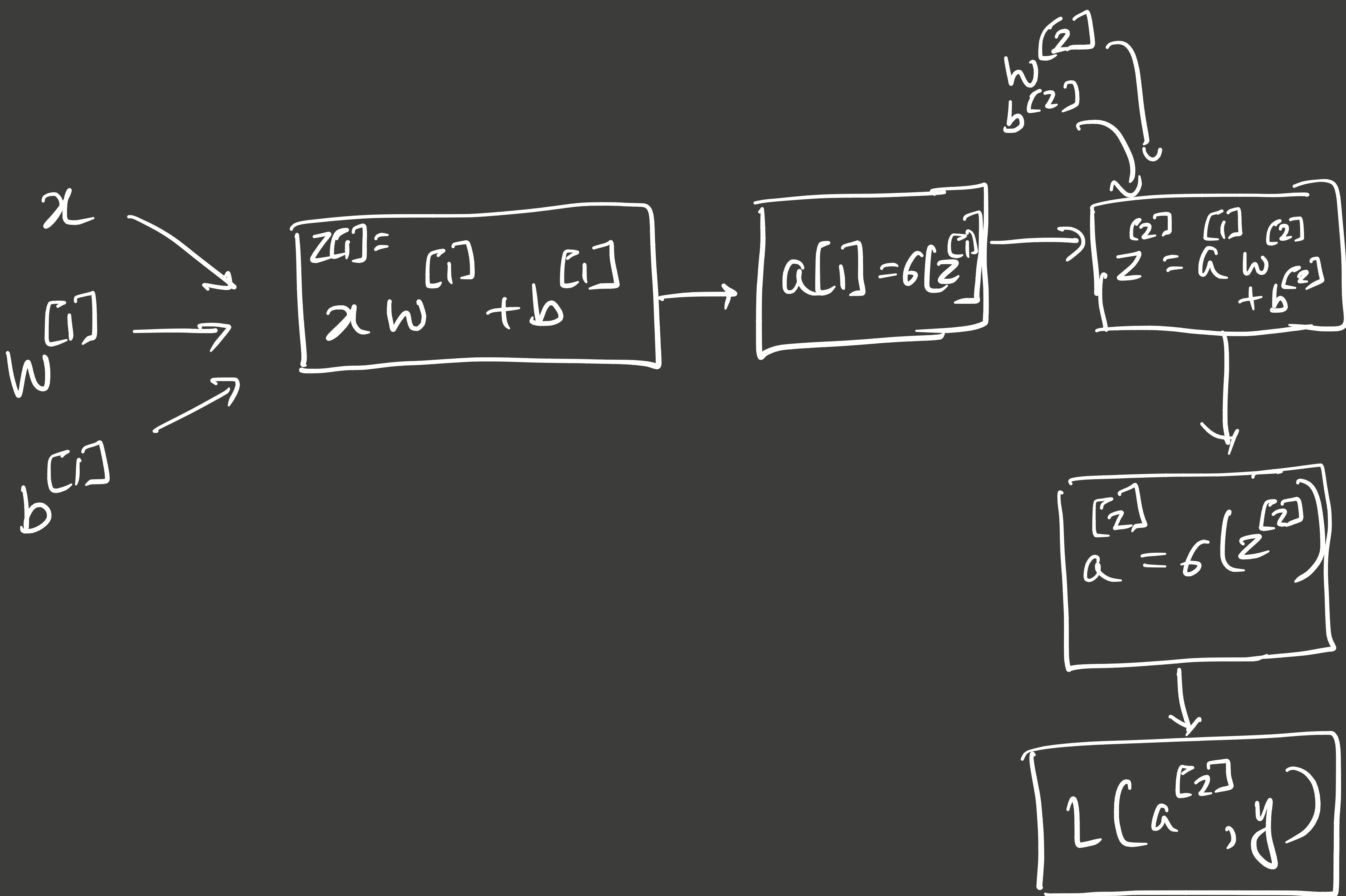
$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial a} \frac{\partial a}{\partial w} \frac{\partial w}{\partial v} \frac{\partial u}{\partial w_1}$$

Derivative of Activation Functions

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

$$\sigma(z) = \text{MAX}\{0, z\} \qquad \frac{\partial \sigma(z)}{\partial z} = \begin{cases} 0 \ ; & z < 0 \\ 1 \ ; & z > 0 \\ \text{undefined} \ ; & z = 0 \end{cases}$$

Backprogation for XOR Network



Backprogation for XOR Network

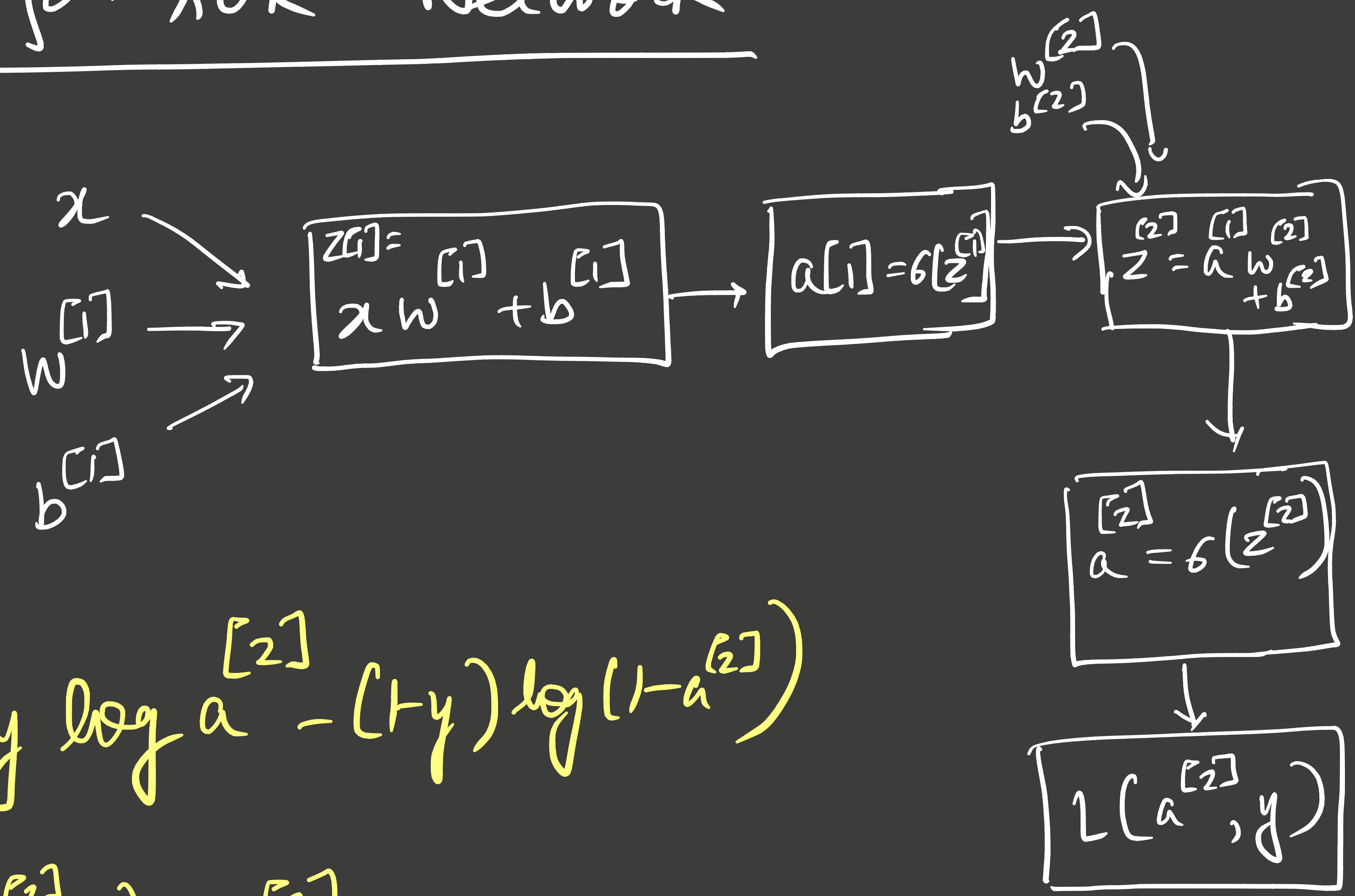
$$x,\ W^{[1]},\ b^{[1]} \rightarrow \boxed{Z^{[1]} = xW^{[1]} + b^{[1]}} \rightarrow \boxed{a^{[1]} = \sigma(z^{[1]})} \rightarrow \boxed{z^{[2]} = a^{[1]}w^{[2]} + b^{[2]}}$$

$$w^{[2]},\ b^{[2]} \rightarrow \boxed{z^{[2]} = a^{[1]}w^{[2]} + b^{[2]}} \rightarrow \boxed{a^{[2]} = \sigma(z^{[2]})} \rightarrow \boxed{L(a^{[2]}, y)}$$

$$L(a^{[2]}, y) = -y\log a^{[2]} - (1-y)\log(1-a^{[2]})$$

Q: $\dfrac{\partial L(a^{[2]}, y)}{\partial w^{[2]}} = ?$

Backprogation for XOR Network



$$L(a^{[2]}, y) = -y \log a^{[2]} - (1-y)\log(1-a^{[2]})$$

$$\frac{\partial L(a^{[2]}, y)}{\partial z^{[2]}} = \frac{\partial L(a^{[2]}, y)}{\partial a^{[2]}} \frac{\partial a^{[2]}}{\partial z^{[2]}}$$

$$\frac{\partial L(a^{[2]}, y)}{\partial a^{[2]}} = \frac{-y}{a^{[2]}} + \frac{(1-y)}{1-a^{[2]}} \quad \ldots\ldots (1)$$

Backprogation for XOR Network

$$
\begin{tikzcd}
x \arrow[dr] & & & & W^{[2]},\ b^{[2]} \arrow[d] \\
W^{[1]} \arrow[r] & \boxed{z^{[1]} = xW^{[1]} + b^{[1]}} \arrow[r] & \boxed{a^{[1]} = \sigma(z^{[1]})} \arrow[rr] & & \boxed{z^{[2]} = a^{[1]}W^{[2]} + b^{[2]}} \arrow[d] \\
b^{[1]} \arrow[ur] & & & & \boxed{a^{[2]} = \sigma(z^{[2]})} \arrow[d] \\
& & & & \boxed{L(a^{[2]}, y)}
\end{tikzcd}
$$

$$\frac{\partial a^{[2]}}{\partial z^{[2]}} = \sigma(z^{[2]})(1-\sigma(z^{[2]})) \quad \ldots \text{(2)}$$

(1) & (2)

$$\frac{\partial L(a^{[2]}, y)}{\partial z^{[2]}} = \left\{ -\frac{y}{a^{[2]}} + \frac{(1-y)}{1-a^{[2]}} \right\} \left\{ \sigma(z^{[2]})(1-\sigma(z^{[2]})) \right\} \quad \ldots \text{(3)}$$

Backprogation for XOR Network

$$
\begin{tikzcd}
x \arrow[dr] & & & & W^{[2]} \arrow[d] \\
W^{[1]} \arrow[r] & \boxed{z^{[1]} = xW^{[1]} + b^{[1]}} \arrow[r] & \boxed{a^{[1]} = \sigma(z^{[1]})} \arrow[r] & \boxed{z^{[2]} = a^{[1]}W^{[2]} + b^{[2]}} \arrow[d] & b^{[2]} \arrow[l] \\
b^{[1]} \arrow[ur] & & & \boxed{a^{[2]} = \sigma(z^{[2]})} \arrow[d] & \\
& & & \boxed{L(a^{[2]}, y)} &
\end{tikzcd}
$$

$$\frac{\partial L(a^{[2]}, y)}{\partial w^{[2]}} = \frac{\partial L(a^{[2]}, y)}{\partial z^{[2]}} \frac{\partial z^{[2]}}{\partial w^{[2]}}$$

$$= a^{[1]} \left\{ \frac{-y}{a^{[2]}} + \frac{(1-y)}{1-a^{[2]}} \right\} \left\{ \sigma(z^{[2]})(1-\sigma(z^{[2]})) \right\} \quad \ldots (4)$$

Similarly

$$\frac{\partial L(a^{[2]}, y)}{\partial b^{[2]}} = \left\{ \frac{-y}{a^{[2]}} + \frac{(1-y)}{1-a^{[2]}} \right\} \left\{ \sigma(z^{[2]})(1-\sigma(z^{[2]})) \right\} \quad \ldots (5)$$

Backprogation for XOR Network

$$
\begin{tikzcd}
x \arrow[dr] & & & W^{[2]},\ b^{[2]} \arrow[d] & \\
W^{[1]} \arrow[r] & \boxed{z^{[1]} = xW^{[1]} + b^{[1]}} \arrow[r] & \boxed{a^{[1]} = \sigma(z^{[1]})} \arrow[r] & \boxed{z^{[2]} = a^{[1]}w^{[2]} + b^{[2]}} \arrow[d] & \\
b^{[1]} \arrow[ur] & & & \boxed{a^{[2]} = \sigma(z^{[2]})} \arrow[d] & \\
 & & & \boxed{L(a^{[2]}, y)} &
\end{tikzcd}
$$

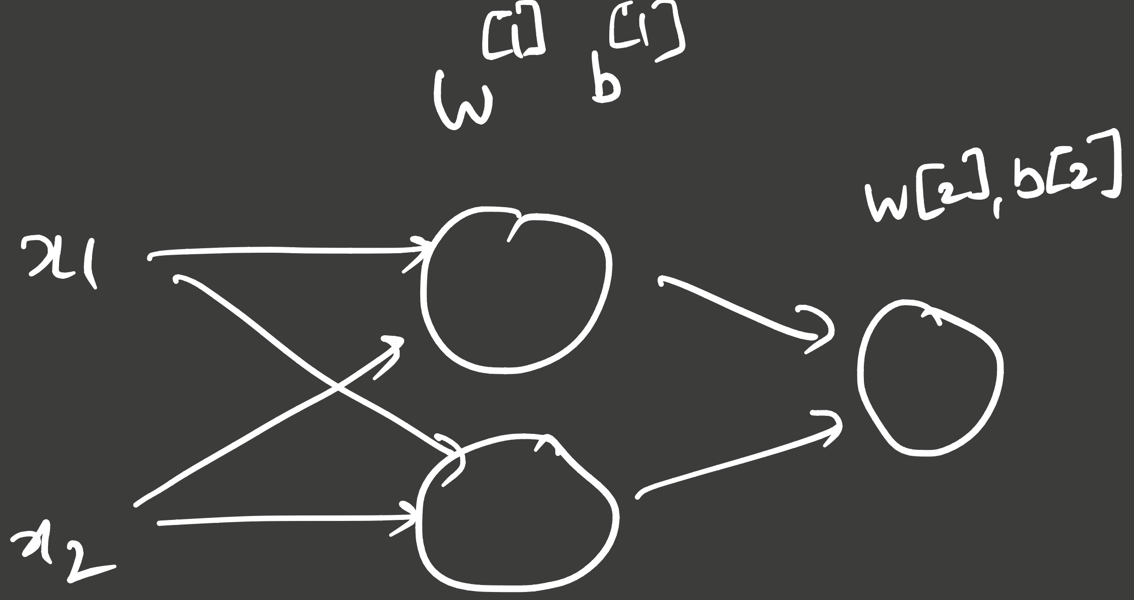
$$\frac{\partial L(a^{[2]}, y)}{\partial z^{[1]}} = \frac{\partial L(a^{[2]}, y)}{\partial z^{[2]}} \underbrace{\frac{\partial z^{[2]}}{\partial a^{[1]}}}_{w^{[2]}} \underbrace{\frac{\partial a^{[1]}}{\partial z^{[1]}}}_{\sigma(z^{[1]})(1-\sigma(z^{[1]}))}$$

$$\frac{\partial L(a^{[2]}, y)}{\partial w^{[1]}} = \frac{\partial L(a^{[2]}, y)}{\partial z^{[1]}} \frac{\partial z^{[1]}}{\partial w^{[1]}} = \frac{\partial L(a^{[2]}, y)}{\partial z^{[1]}} * x$$

WORKED OUT EXAMPLE

$x_1 = 1; x_2 = 1; y_{TRUE} = 0$; $\sigma^{[1]} = RELU$

$\sigma^{[2]} = RELU$



$$W^{[1]} = \underbrace{\begin{bmatrix} | & | & \\ w_1^{[1]} & w_2^{[1]} & \cdots \\ | & | & \end{bmatrix}}_{\text{\# Hidden units in layer 1}}$$

No. of I/ps = Dimensionality of $a^{[0]}$

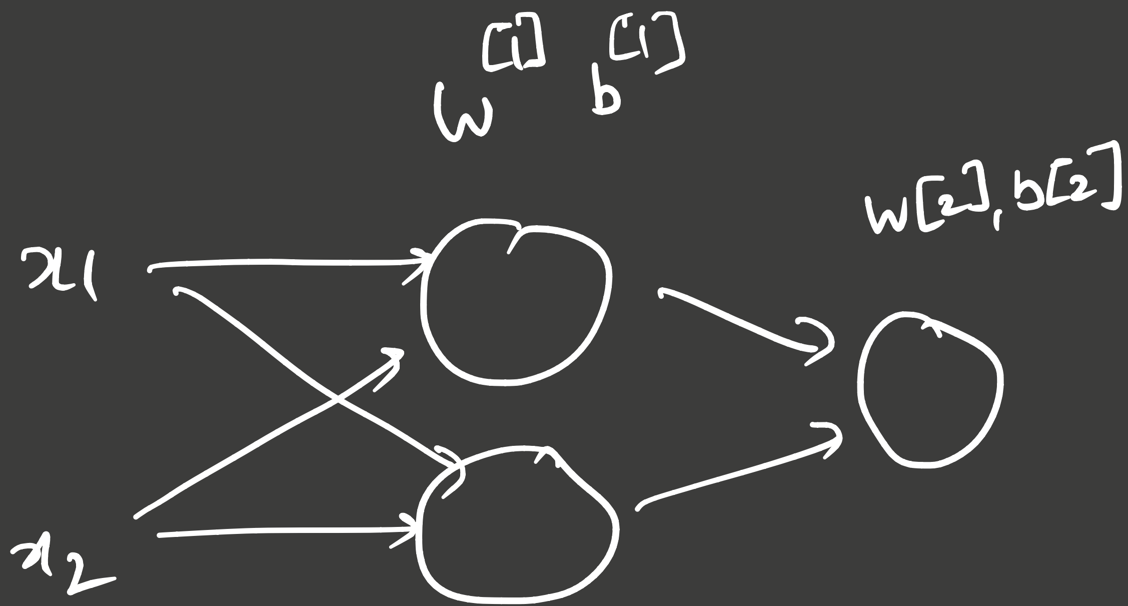
$b^{[1]} = [\quad]$

hidden units in Layer 1

WORKED OUT EXAMPLE

$x_1 = 1; x_2 = 1; y_{TRUE} = 0$; $\sigma^{[1]} = RELU$

$\sigma^{[2]} = RELU$



$$W^{[1]} = \begin{bmatrix} .1 & .2 \\ .0 & .1 \end{bmatrix} \qquad b^{[1]} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

RANDOM INIT

$$W^{[2]} = \begin{bmatrix} .1 \\ .2 \end{bmatrix} \qquad b^{[2]} = \begin{bmatrix} 0 \end{bmatrix}$$

WORKED OUT EXAMPLE

$x_1 = 1; x_2 = 1; y_{TRUE} = 0$; $\sigma^{[1]} = RELU$

$\sigma^{[2]} = RELU$

$$a^{[1]} = \sigma\left(a^{[0]} w^{[1]} + b^{[1]}\right) = \sigma\left\{[1 \quad 1] \begin{bmatrix} .1 & .2 \\ .0 & .1 \end{bmatrix} + [0 \quad 0]\right\} = [.1 \quad .2]$$

$$a^{[2]} = \sigma\left[a^{[1]} w^{[2]} + b^{[2]}\right] = \sigma\left[[.1 \quad .2]\begin{bmatrix} .1 \\ .2 \end{bmatrix} + [0]\right] = .05$$

$$w^{[1]} = \begin{bmatrix} .1 & .2 \\ .0 & .1 \end{bmatrix} \qquad b^{[1]} = [0 \quad 0]$$

$$w^{[2]} = \begin{bmatrix} .1 \\ .2 \end{bmatrix} \qquad b^{[2]} = [0]$$

WORKED OUT EXAMPLE

$x_1 = 1; x_2 = 1; y_{TRUE} = 0$; $\sigma^{[1]} = RELU$
$\sigma^{[2]} = RELU$

$$a^{[1]} = \sigma\left(a^{[0]} w^{[1]} + b^{[1]}\right) = \sigma\left\{\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} .1 & .2 \\ .0 & .1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix}\right\} = \begin{bmatrix} .1 & .2 \end{bmatrix}$$

$$a^{[2]} = \sigma\left[a^{[1]} w^{[2]} + b^{[2]}\right] = \sigma\left[\begin{bmatrix} .1 & .2 \end{bmatrix} \begin{bmatrix} .1 \\ .2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix}\right] = .05$$

let $L(a^{[2]}, y) = \frac{1}{2}\left\{a^{[2]} - y\right\}^2$

$$\therefore \frac{\partial L(a^{[2]}, y)}{\partial a^{[2]}} = a^{[2]} - y = .05 - 0 = .05$$

WORKED OUT EXAMPLE

$x_1 = 1; x_2 = 1; y_{TRUE} = 0$; $\sigma^{[1]} = RELU$

$\sigma^{[2]} = RELU$

$$a^{[1]} = \sigma\left(a^{[0]} w^{[1]} + b^{[1]}\right) = \sigma\left\{[1 \quad 1] \begin{bmatrix} .1 & .2 \\ .0 & .1 \end{bmatrix} + [0 \quad 0]\right\} = [.1 \quad .2]$$

$$a^{[2]} = \sigma\left[a^{[1]} w^{[2]} + b^{[2]}\right] = \sigma\left[[.1 \quad .2] \begin{bmatrix} .1 \\ .2 \end{bmatrix} + [0]\right] = .05$$

$$\therefore \boxed{\frac{\partial L(a^{[2]}, y)}{\partial a^{[2]}} = a^{[2]} - y = .05 - 0 = .05}$$

$$\frac{\partial L(a^{[2]}, y)}{\partial w^{[2]}} = \frac{\partial L(a^{[2]}, y)}{\partial a^{[2]}} \frac{\partial a^{[2]}}{\partial z^{[2]}} \frac{\partial z^{[2]}}{\partial w^{[2]}} = .05 * 1 * a^{[1]}$$

$$= [.005 \quad .01]$$

WORKED OUT EXAMPLE

$x_1 = 1; x_2 = 1; y_{TRUE} = 0$; $\sigma^{(1)} = RELU$

$\sigma^{(2)} = RELU$

$$a^{[1]} = \sigma\left(a^{[0]} w^{[1]} + b^{[1]}\right) = \sigma\left\{[1 \quad 1] \begin{bmatrix} .1 & .2 \\ .0 & .1 \end{bmatrix} + [0 \quad 0]\right\} = [.1 \quad .2]$$

$$a^{[2]} = \sigma\left[a^{[1]} w^{[2]} + b^{[2]}\right] = \sigma\left[[.1 \quad .2] \begin{bmatrix} .1 \\ .2 \end{bmatrix} + [0]\right] = .05$$

$$\frac{\partial L(a^{[2]}, y)}{\partial w^{[2]}} = [.005 \quad .01]$$

$$\frac{\partial L(a^{(2)}, y)}{\partial b^{(2)}} = \frac{\partial L(a^{[2]}, y)}{\partial a^{(2)}} * \frac{\partial a^{[2]}}{\partial z^{(2)}} * \frac{\partial z^{(2)}}{\partial b^{[2]}} = .05 * 1 = .05$$

$\therefore$ Update Rule: $w^{[2]} = w^{[2]} - \text{Learning Rate} * [.005 \quad .01]$

--

Digit Classifier using MLP

* 64×64 grayscale image
    * 0 → Black
      1 → white
      [0-1] → B/w Black & white

* I/p layer : $64 \times 64 = 4096$

Question1 : Is new digit 9 or not?

⇒ O/p size = 1 neurons

If hidden layer sizes are

100, 20, 1 (o/p layer)

what is # params?

If hidden layer sizes are

100, 20, 1 (o/p layer)

what is # params?

$$a^{[0]} = [\ \cdots\cdots\ ]_{1\times 4096}$$

$$w^{[1]} = \begin{bmatrix} \uparrow & & \\ 4096 & & \\ \downarrow & & \\ \cdots & & \end{bmatrix}_{4096 \times 100}$$

$\leftarrow 100 \rightarrow$

$$b^{[1]} = [\ \ ]_{1\times 100}$$

If hidden layer sizes are

100, 20, 1 (o/p layer)

what is # params?

$a^{[0]} = 1 \times 4096$; $w^{[1]} = 4096 * 100$; $b^{[1]} = 1 * 100$

$z^{[1]} = a^{[0]} w^{[1]} + b^{[1]} = 1 * 100$

$a^{[1]} = 1 * 100$

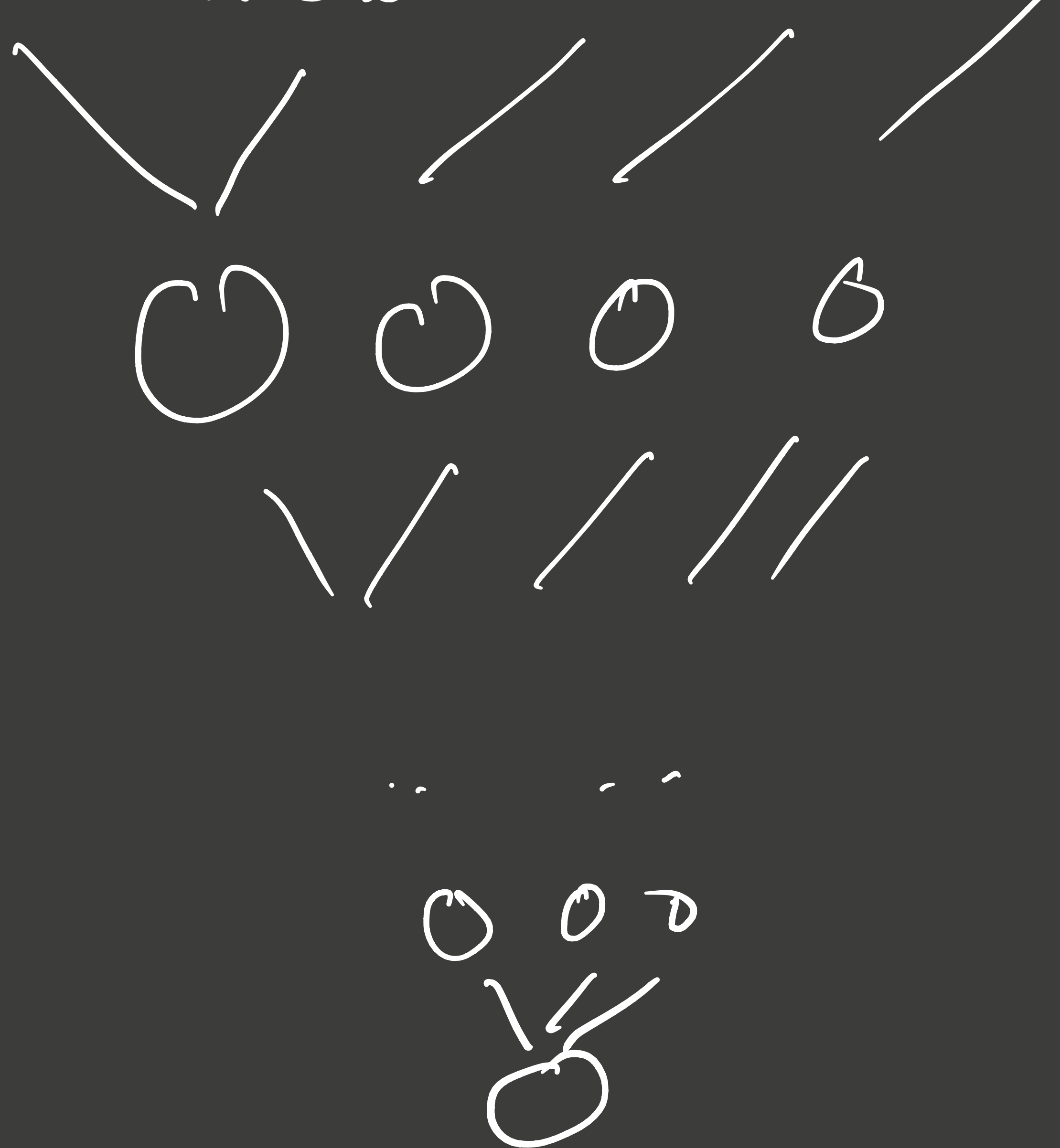
$w^{[2]} = 100 * 20$; $b^{[2]} = 1 * 20$

$w^{[3]} = 1 * 20$ $\quad$ $b^{[3]} = 1 * 1$

Total params = $\sum_{i=1}^{3} \text{size}(w^{[i]}) + \text{size}(b^{[i]})$

Case Study I : Housing price prediction

$x_i = \{$ Area, Distance to metro, # schools, ... $\}$



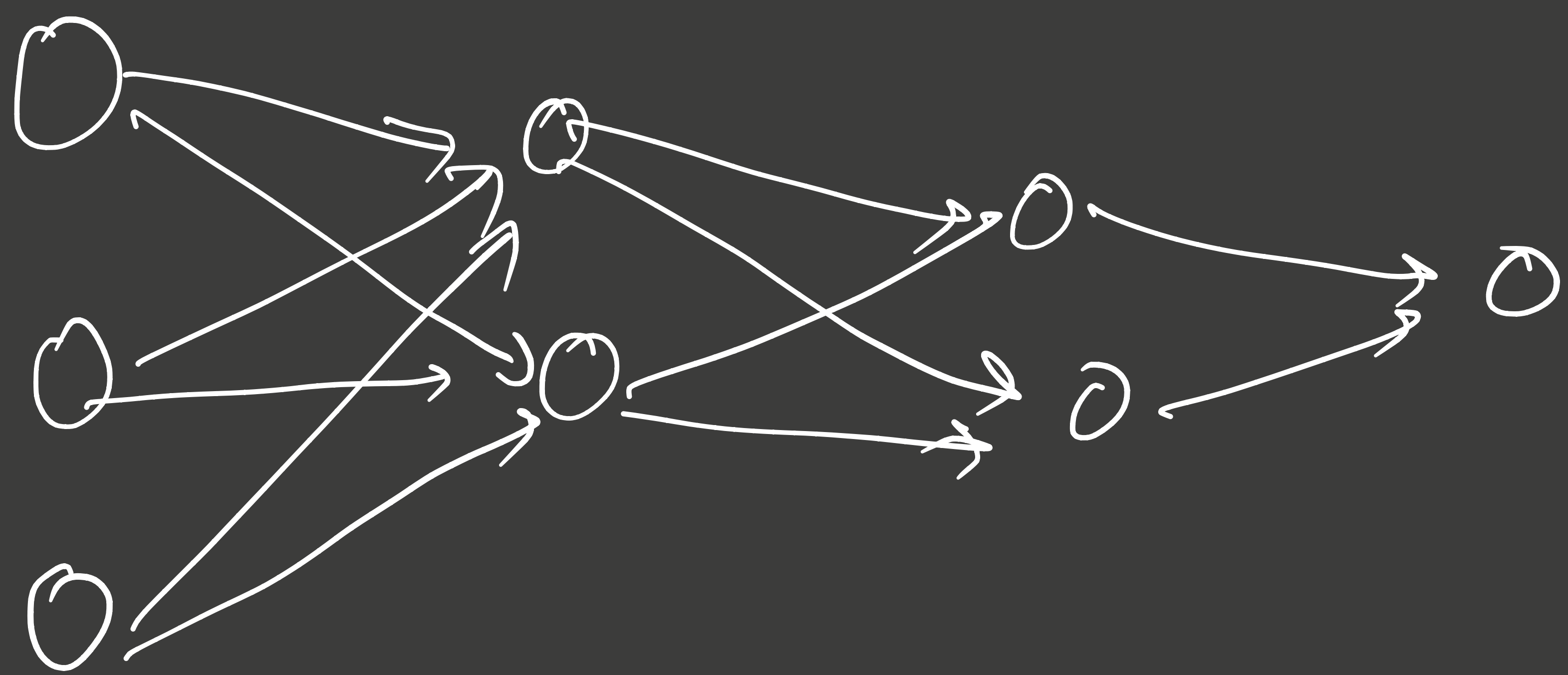
ReLU

Linear or ReLU activation

REGULARIZATION / PREVENT OVERFITTING

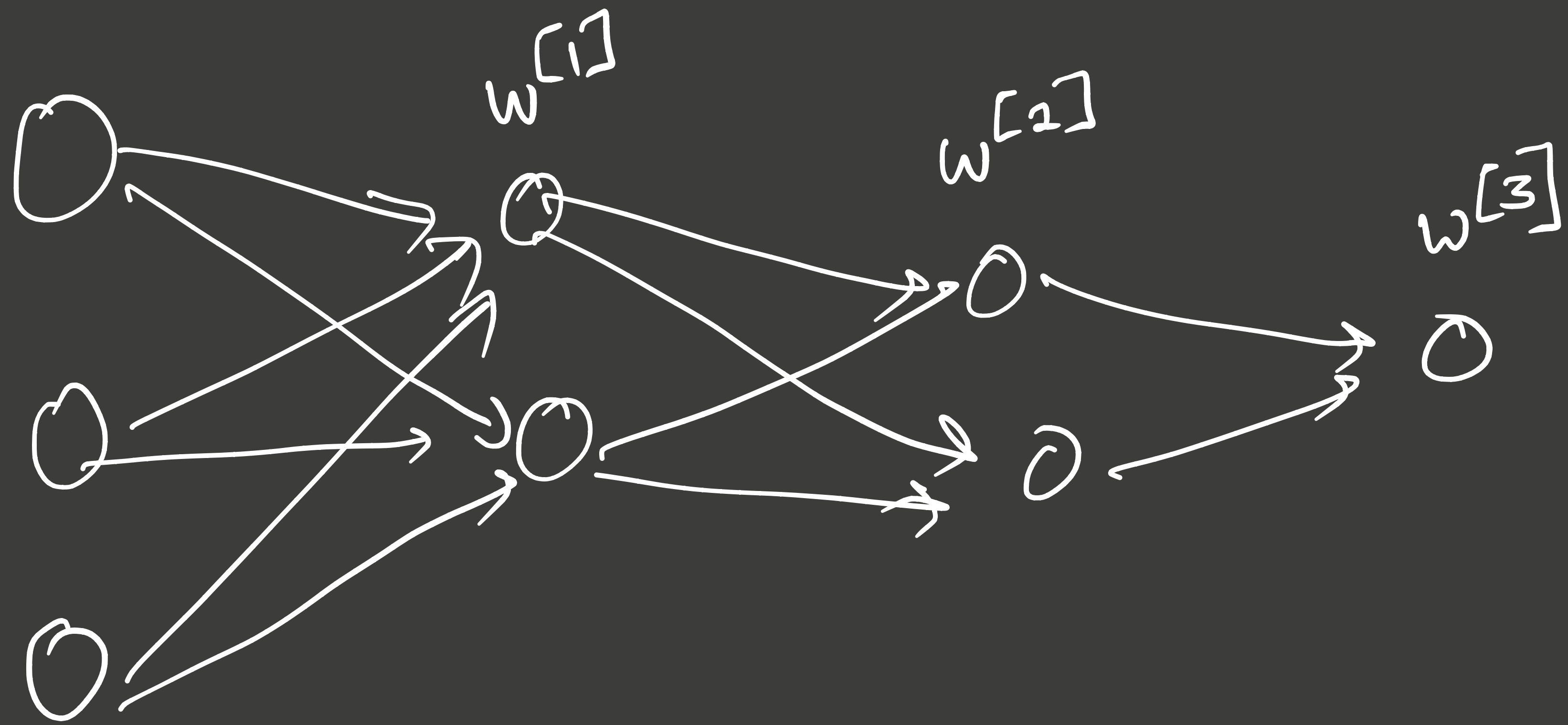
1. Dropout
2. Penalty Terms.
3. Data augmentation
4. Early stopping

Dropout



* with a probability 'p' keep a node...

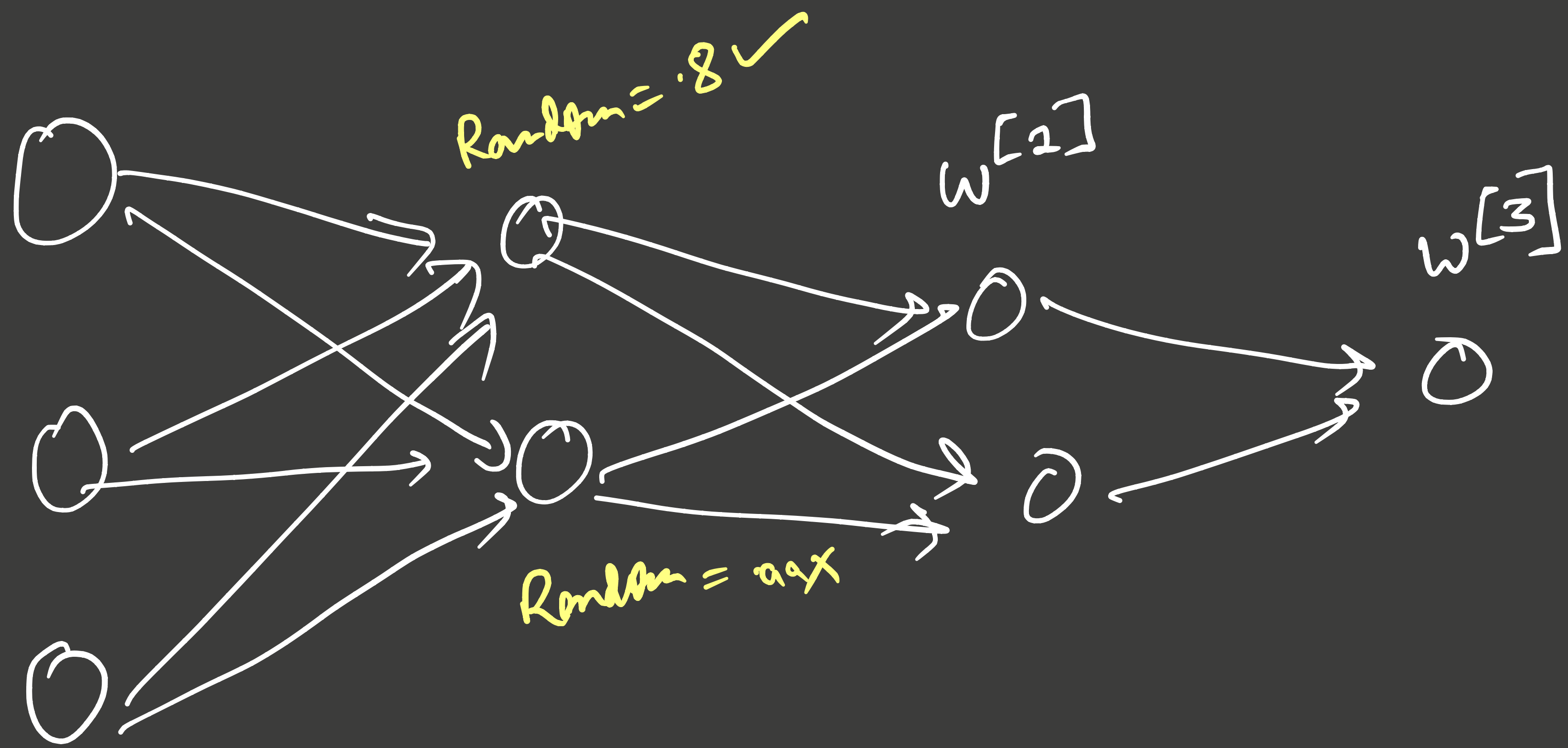
Dropout



* with a probability $\underline{p}$ (eg. 0.9) keep a node

* Random() < p : Keep ; Else : Drop

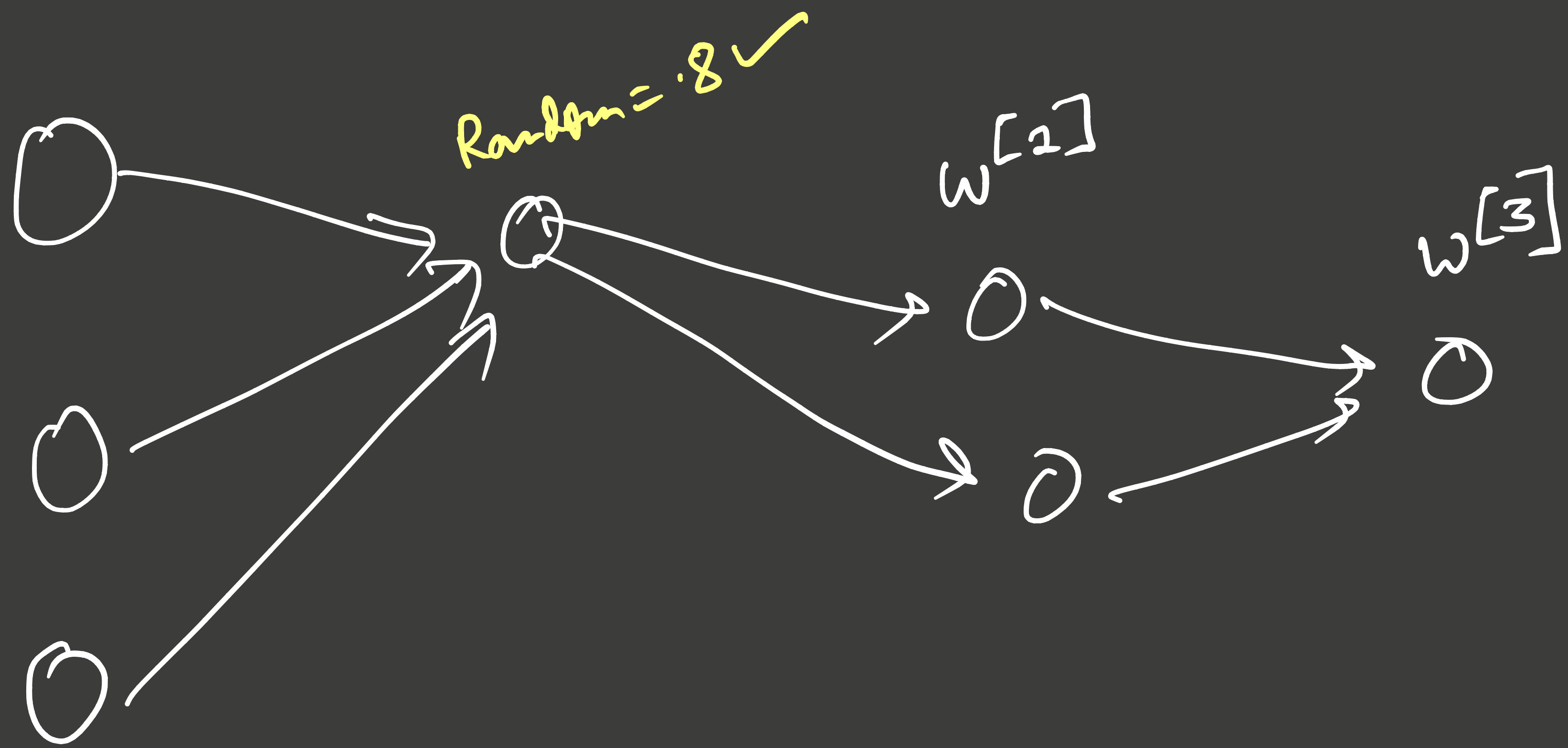
Dropout



* with a probability p (eg. 0·9) keep a node

* Random() < p : Keep ; Else : Drop

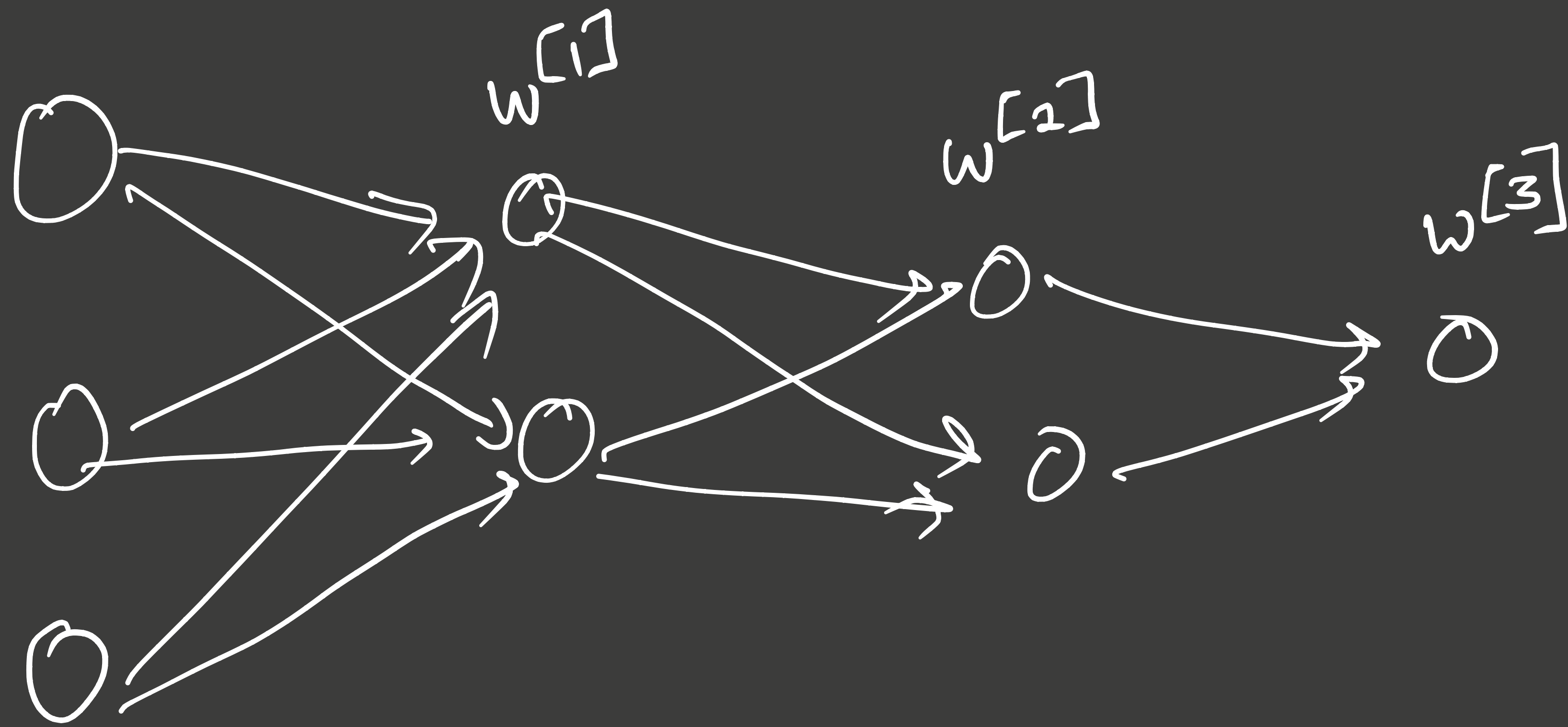
Dropout



* with a probability p (eg. 0.9) keep a node

* Random() < p : Keep ; Else : Drop

Dropout



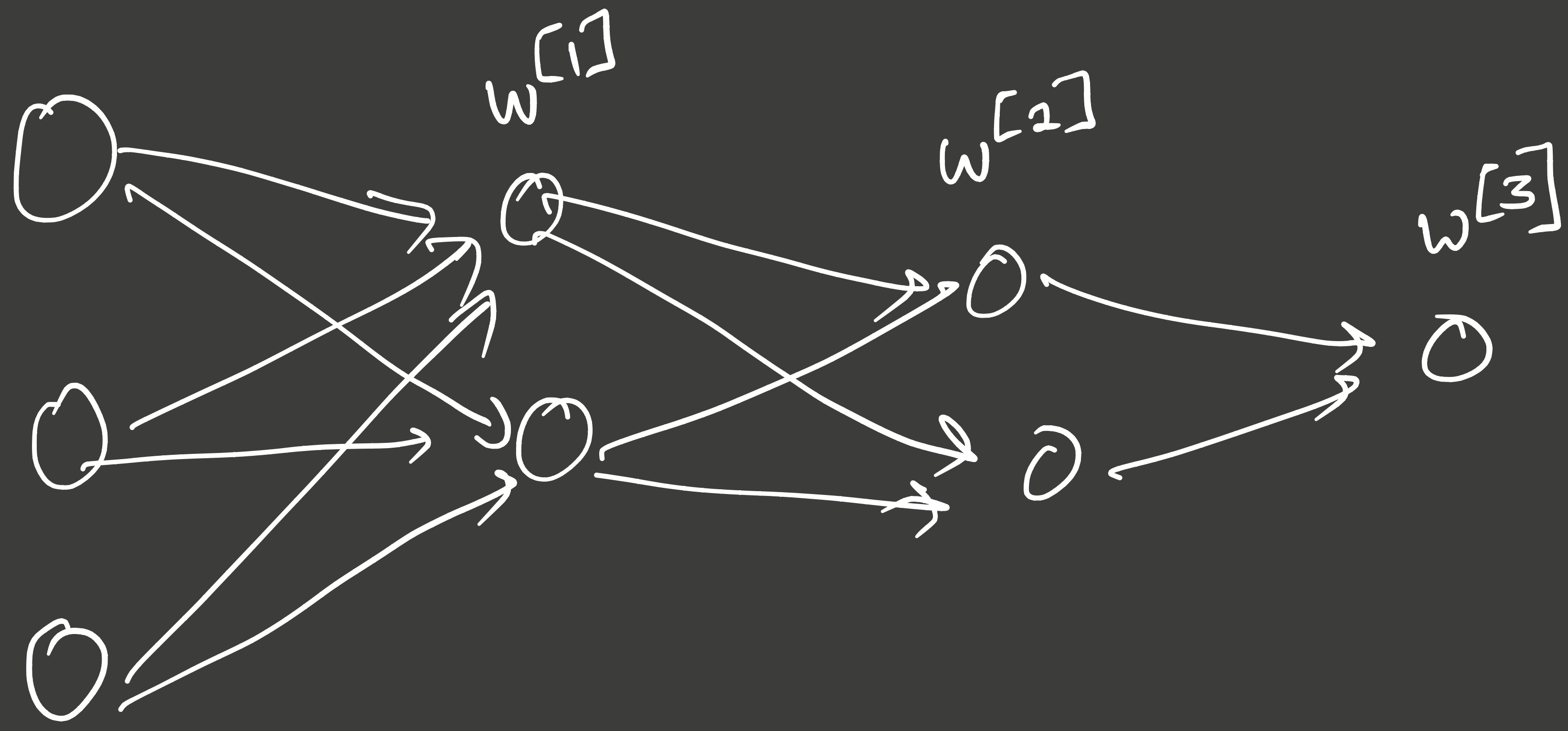
$A^{[1]} = A^{[1]} * \text{MASK}$

{Element-wise multiplication}

Mask = RANDOM($A^{[l]}$.shape[0], $A^{[l]}$.shape[1]) < p

$= \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Dropout



$$A^{[1]} = A^{[1]} * \text{MASK}$$

$$A^{[1]} = A^{[1]} / p$$

(why?

$\because$ we removed some nodes,

$E(A^{[1]})$ would reduce ...

$$\text{Mask} = \text{RANDOM}(A^{[1]}.\text{shape}[0], A^{[1]}.\text{shape}[1]) < p$$

$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Why Dropout works

① Smaller nets $\Rightarrow$ less overfitting

② Since nodes can be "shut" at random,

weight spread across nodes

$\Downarrow$

Shrinkage (akin L_2)

REGULARISATION USING L1|L2 PENALTY

$$J(w^{[1]}, b^{[1]}, \ldots w^{[l]}, b^{[l]} \ldots)$$

$$= \sum_i L(\hat{y}_i, y_i) + \lambda \sum_{l=1}^{L} \|w^{[l]}\|^2$$

LOSS

L2 REGULARISATION

Data Augmentation

Example: Add transformations of images to make train set "bigger"