RECENT SUCCESSES OF NN

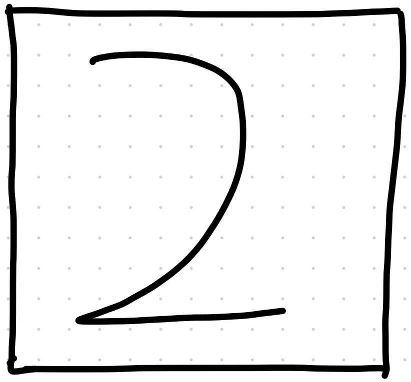
* state-of-the-art (SOTA) in most fields

PARADIGM CHANGE

PARADIGM CHANGE

2

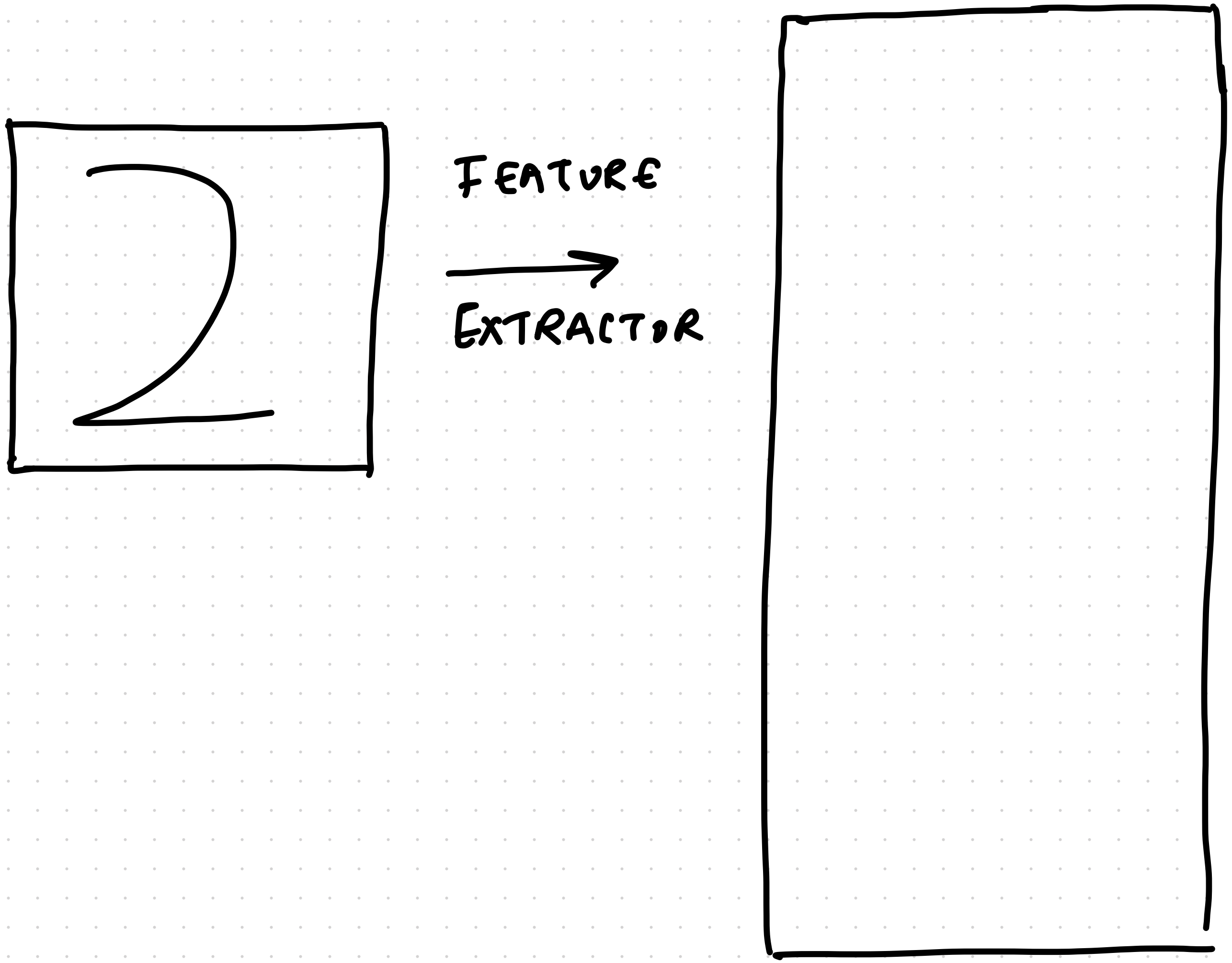
PARADIGM CHANGE



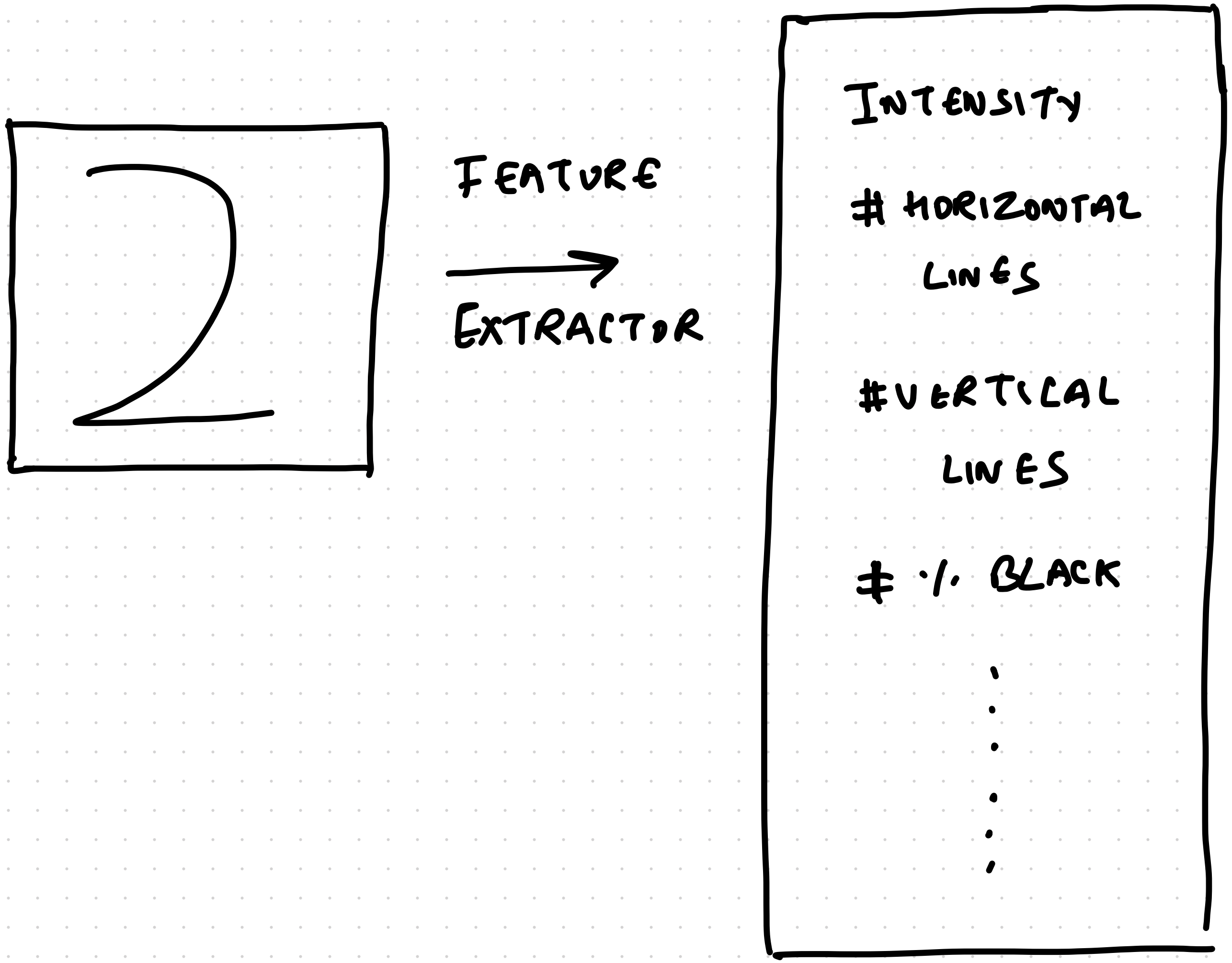
FEATURE

EXTRACTOR

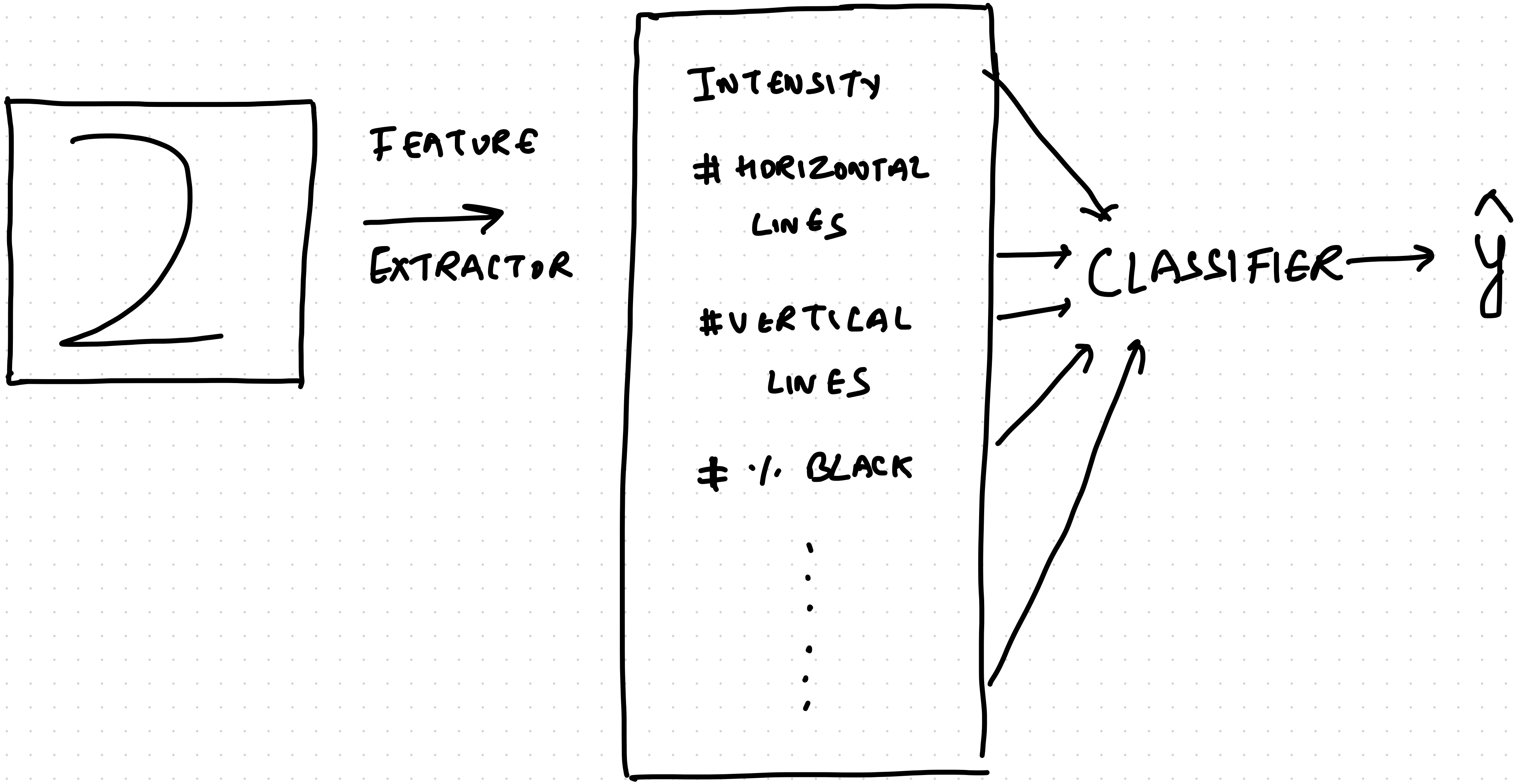
PARADIGM CHANGE



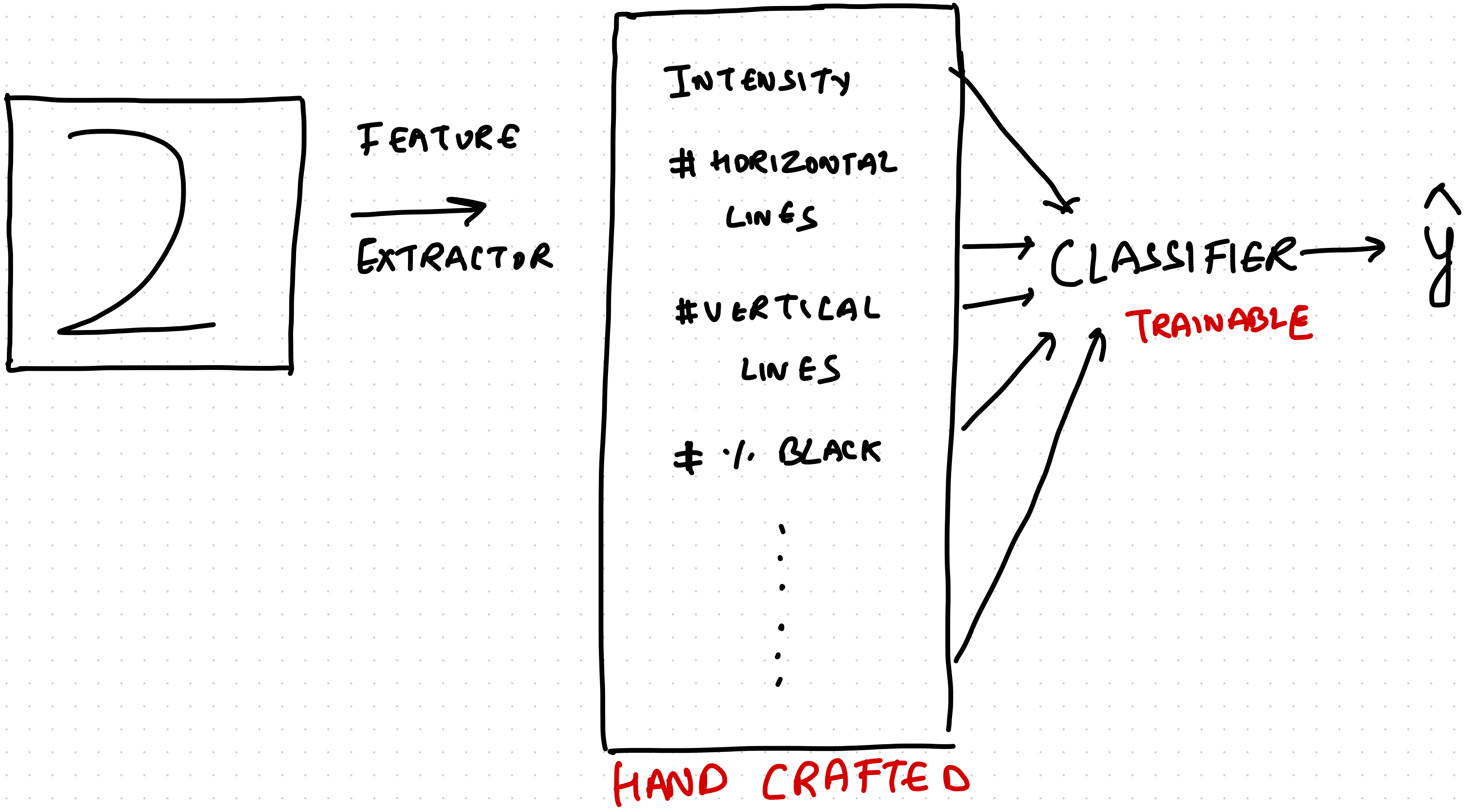
PARADIGM CHANGE



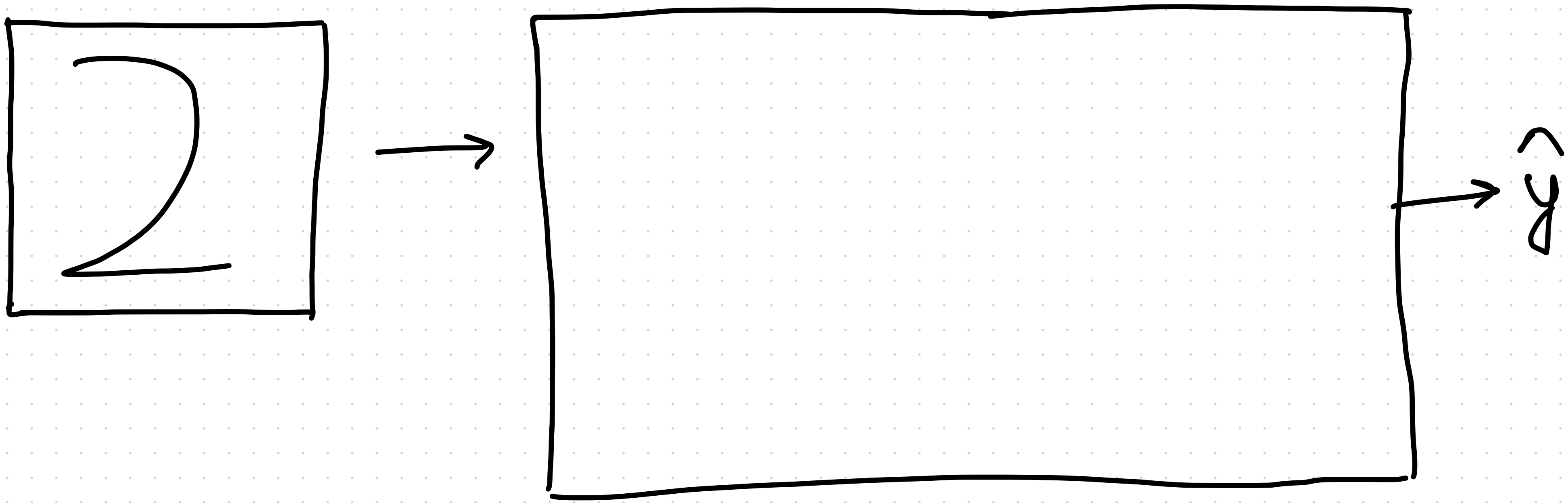
PARADIGM CHANGE



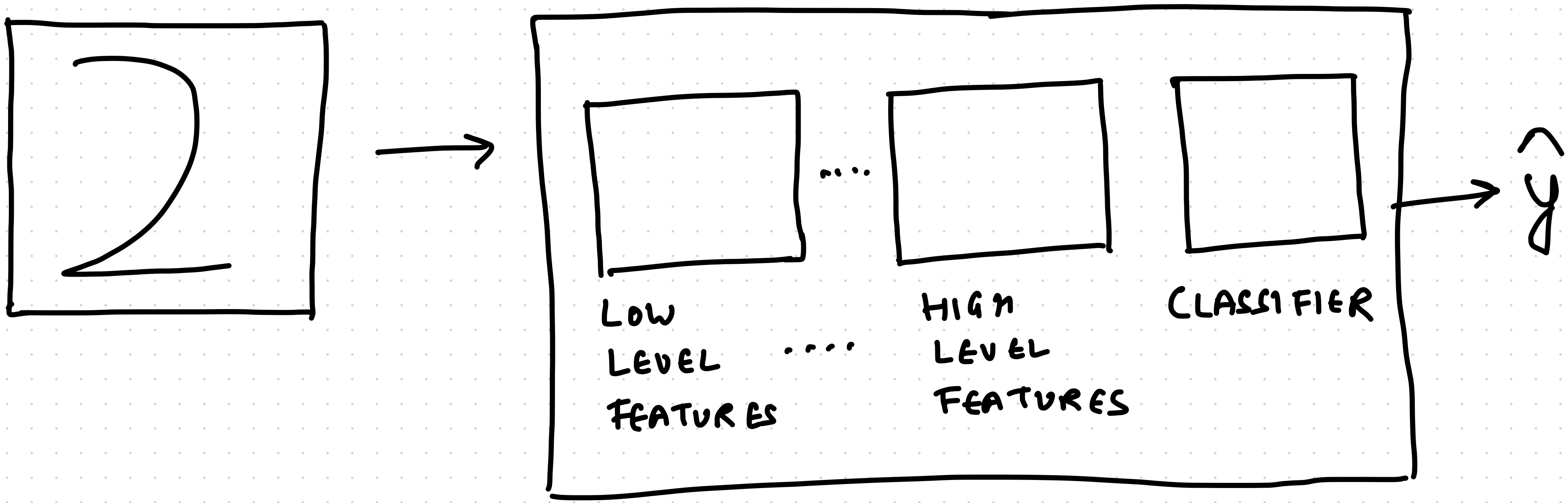
PARADIGM CHANGE



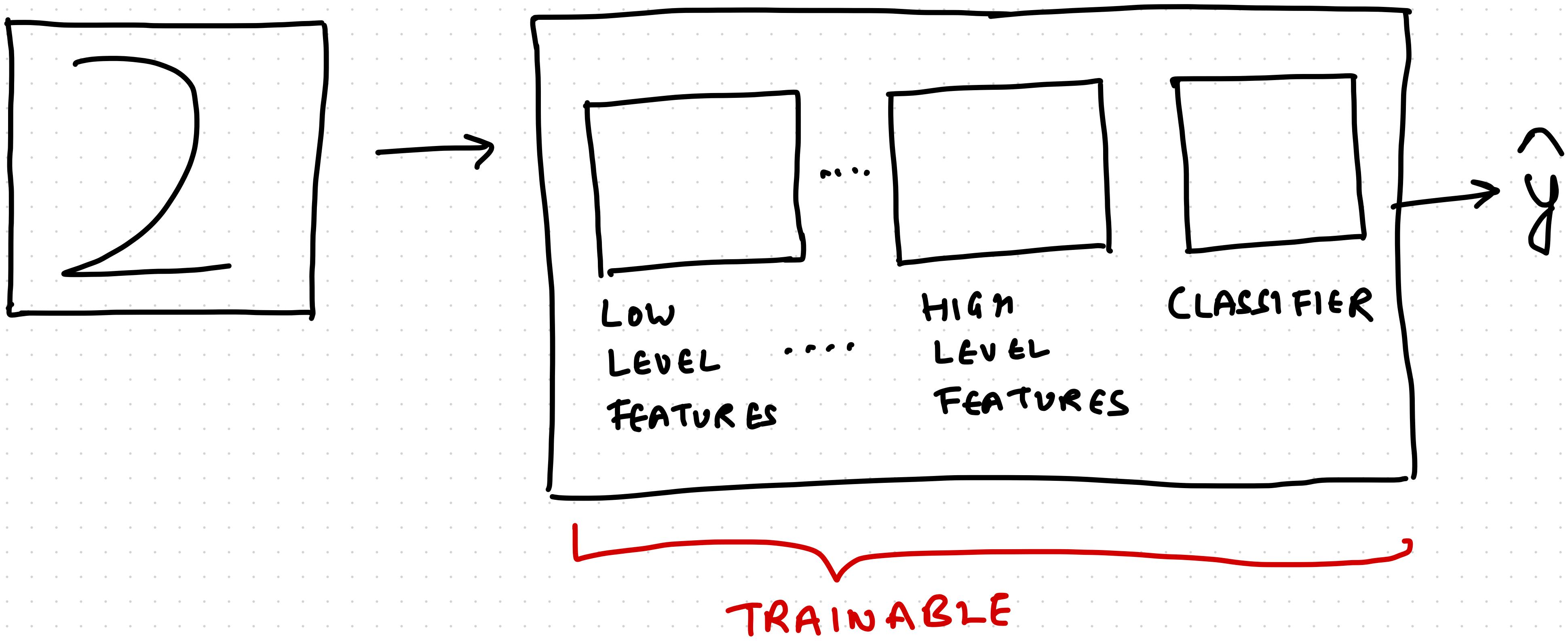
PARADIGM CHANGE (NNs)



PARADIGM CHANGE (NNS)



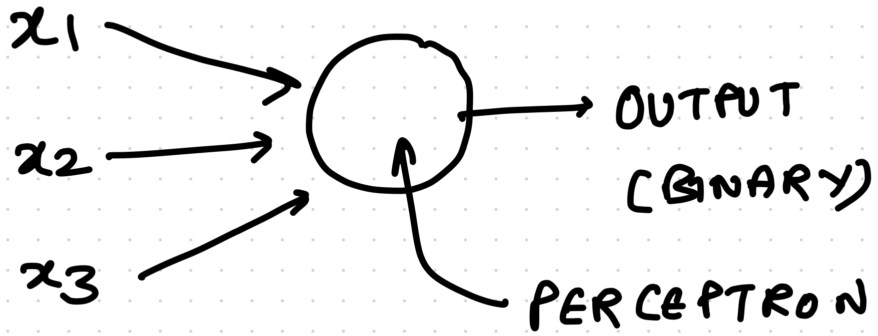
PARADIGM CHANGE (NNS)



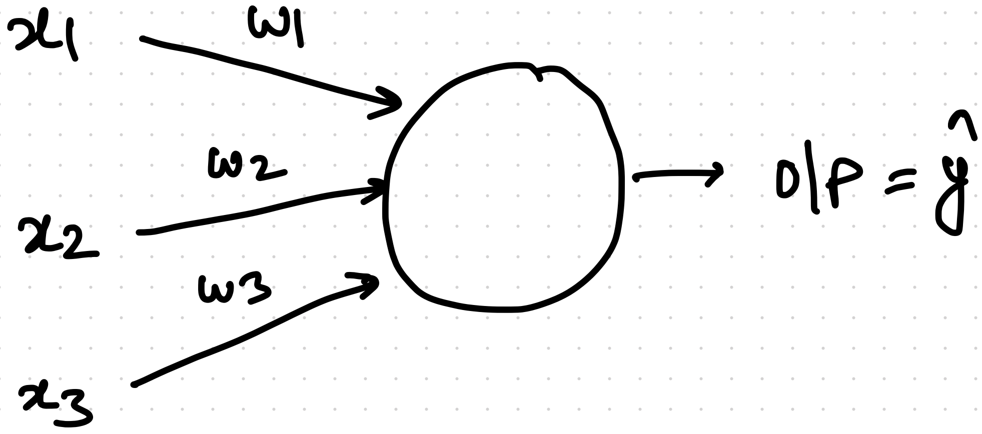
PERCEPTRON

— ARTIFICIAL NEURON DEVELOPED BY ROSENBLATT IN 1960S INSPIRED BY MC CULLOCH & PITTS

BINARY I/P

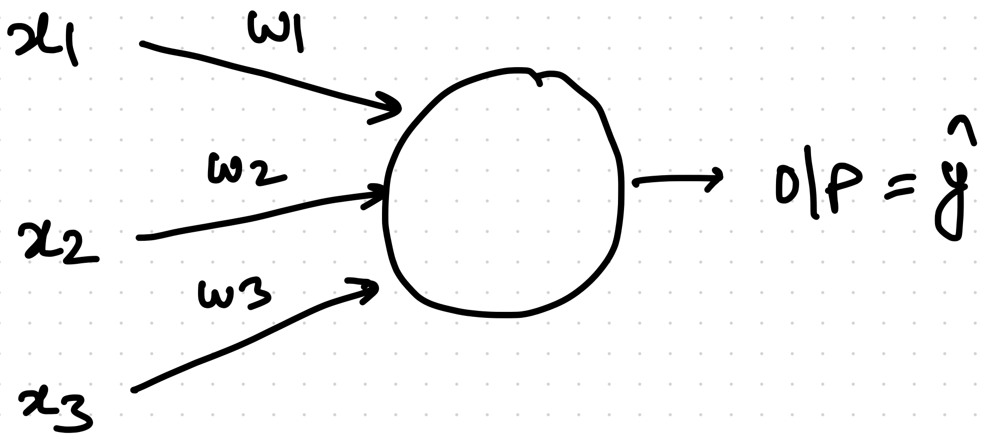


PERCEPTRON



$$O|P = \hat{y} = \begin{cases} 0 \; ; & \sum w_i x_i \leq \text{THRESHOLD} \\ 1 \; ; & \sum w_i x_i > \text{THRESHOLD} \end{cases}$$

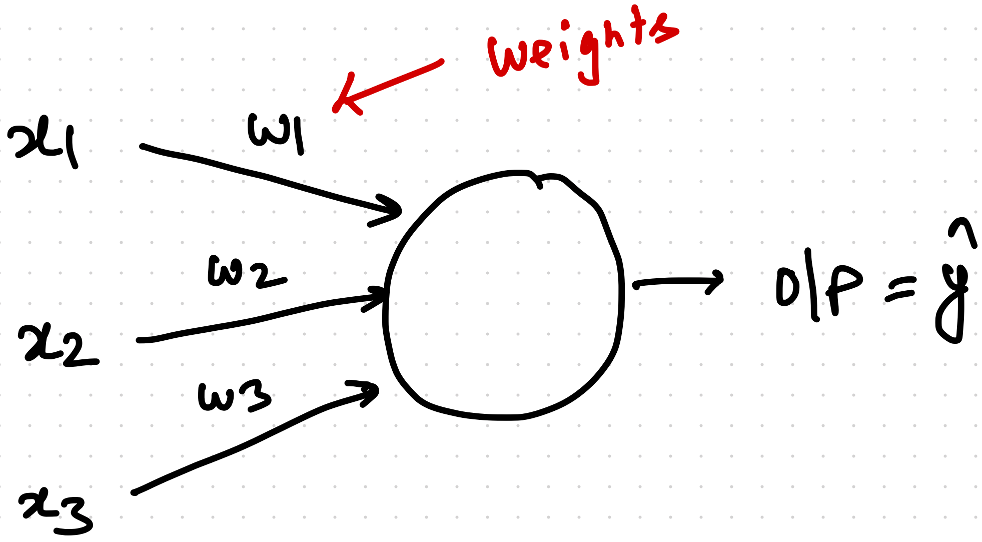
PERCEPTRON



$$O|P = \hat{y} = \begin{cases} 0 \ ; & \sum w_i x_i \leq \text{THRESHOLD} \\ 1 \ ; & \sum w_i x_i > \text{THRESHOLD} \end{cases}$$

NEURONS "FIRE" ABOVE THRESHOLD

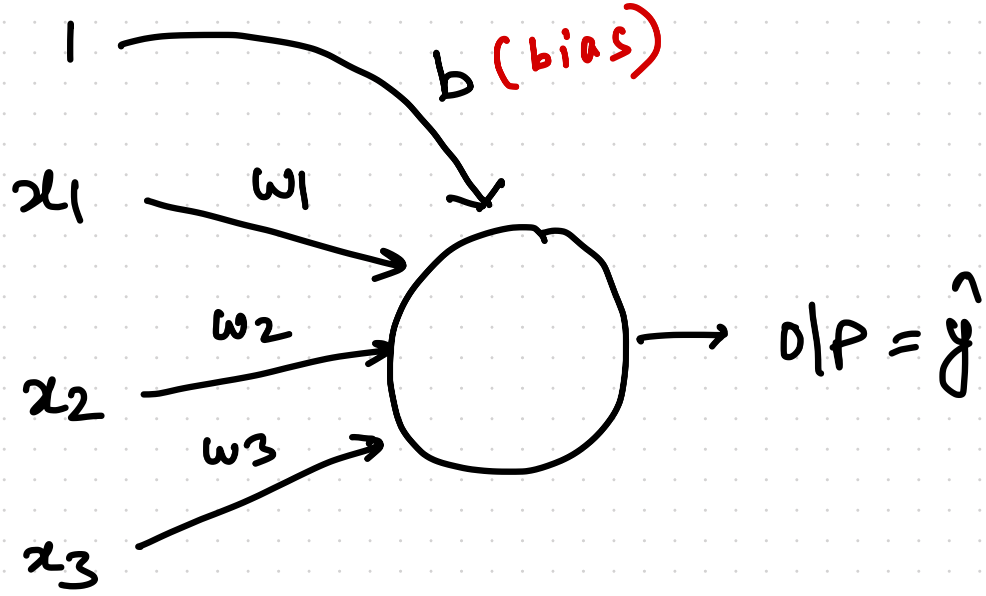
PERCEPTRON



$$O|P = \hat{y} = \begin{cases} 0 \; ; & \sum w_i x_i \le \text{THRESHOLD} \\ 1 \; ; & \sum w_i x_i > \text{THRESHOLD} \end{cases}$$

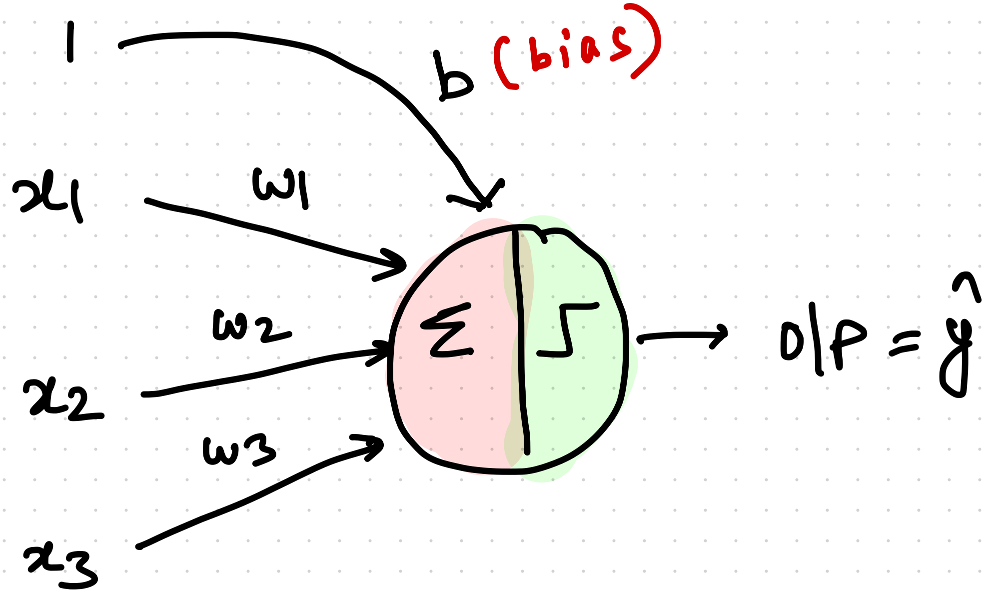
NEURONS "FIRE" ABOVE THRESHOLD

PERCEPTRON



$$O|P = \hat{y} = \begin{cases} 0 \; ; & \sum w_i x_i + b \le 0 \\ 1 \; ; & \sum w_i x_i + b > 0 \end{cases}$$

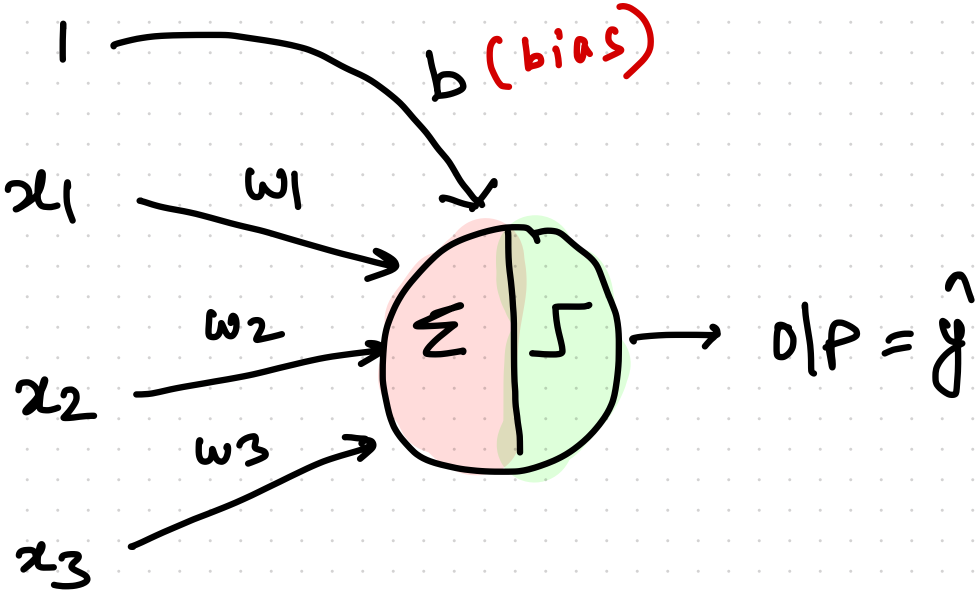
PERCEPTRON



NEURON HAS 2 COMPONENTS

1. SUMMATION
2. ACTIVATION : STEP FUNCTION

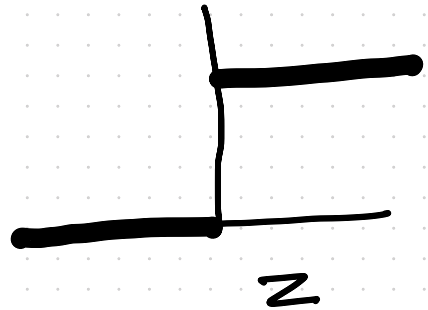
PERCEPTRON



NEURON HAS 2 COMPONENTS

1. SUMMATION
2. ACTIVATION : STEP FUNCTION

(SIGN/STEP)
Activation



LEARNING BINARY GATES

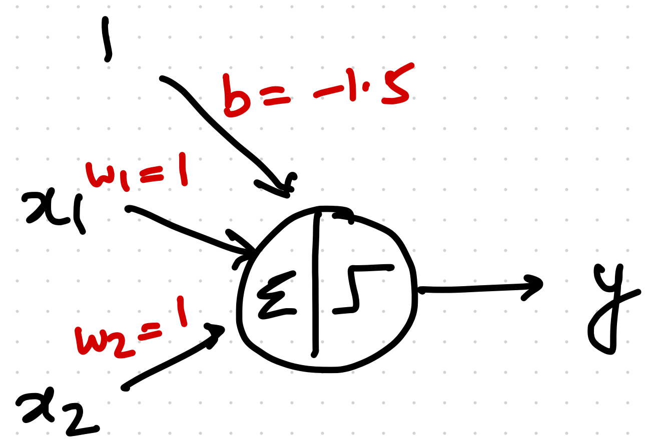
Q) FOR 2 I/PS x_1 & x_2 learn w_i's and b for BINARY AND

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

LEARNING BINARY GATES

Q) FOR 2 I/ps x_1 & x_2 learn w_i's and b for BINARY AND

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1



LEARNING BINARY GATES

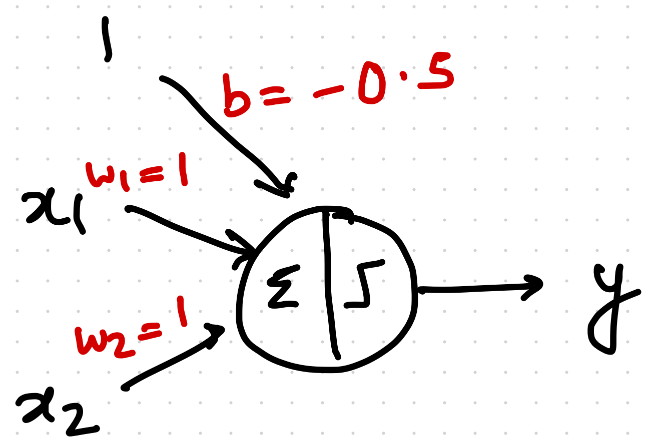
Q) FOR 2 I/ps x_1 & x_2 learn w_i's and b for BINARY OR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

LEARNING BINARY GATES

Q) FOR 2 I/Ps x_1 & x_2 learn w_i's and b for BINARY OR

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1



LEARNING BINARY GATES

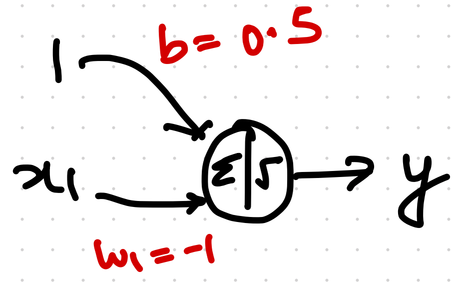
Q) FOR 1 I/ps x_1 learn w_i's and b for UNARY NOT

x_1	y
0	1
1	0

LEARNING BINARY GATES

Q) FOR 1 I/Ps x_1 learn w_i's and b for UNARY NOT

x_1	y
0	1
1	0



LEARNING BINARY GATES

Q) FOR 2 I/ps x_1 & x_2 learn w_i's and b for NAND

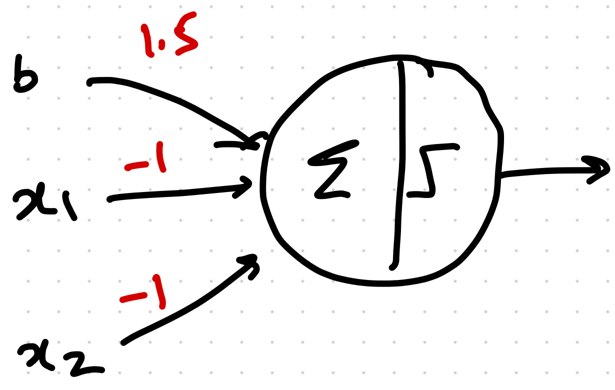
x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

LEARNING BINARY GATES

Q) FOR 2 I/ps x_1 & x_2 learn w_i's and b for NAND

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

APPROACH #1

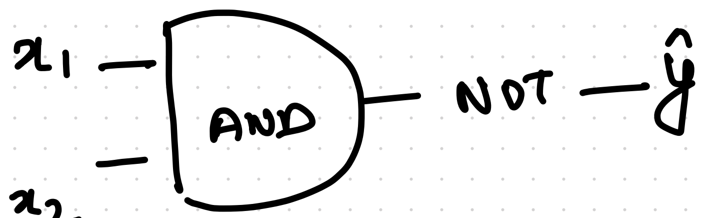


LEARNING BINARY GATES

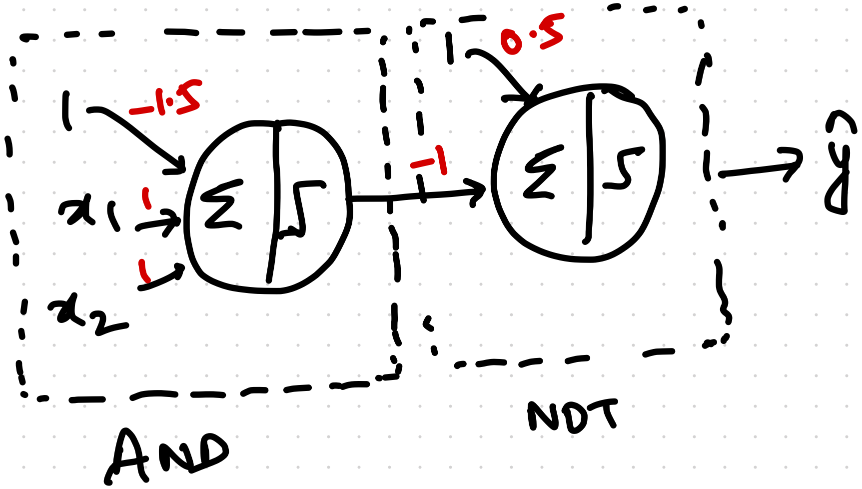
Q) FOR 2 I/Ps x_1 & x_2 learn w_i's and b for NAND

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

APPROACH #2



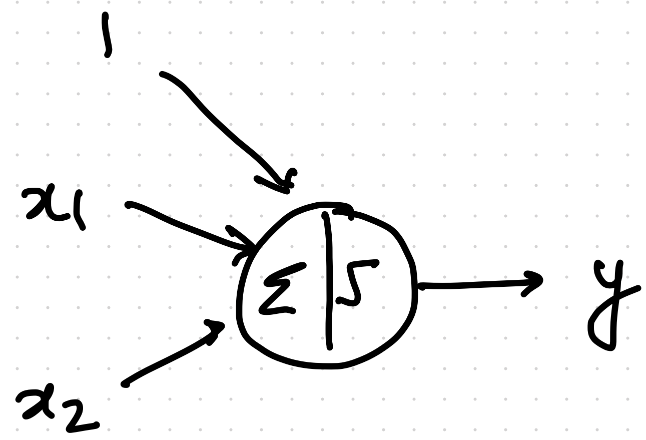
or



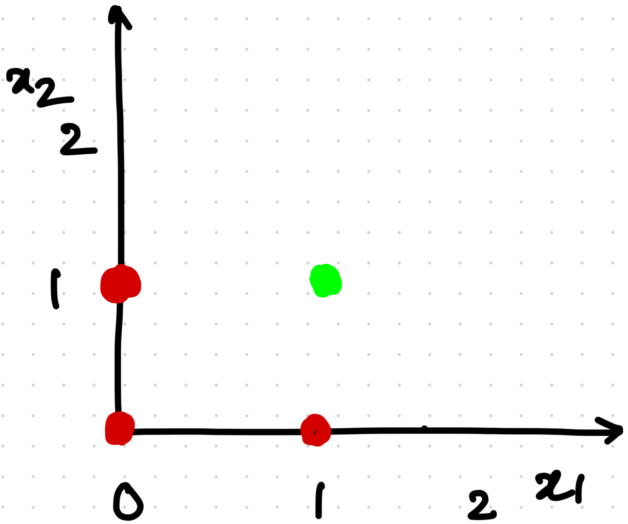
LEARNING BINARY GATES

Q) FOR 2 I/ps x_1 & x_2 learn w_i's and b for BINARY XOR

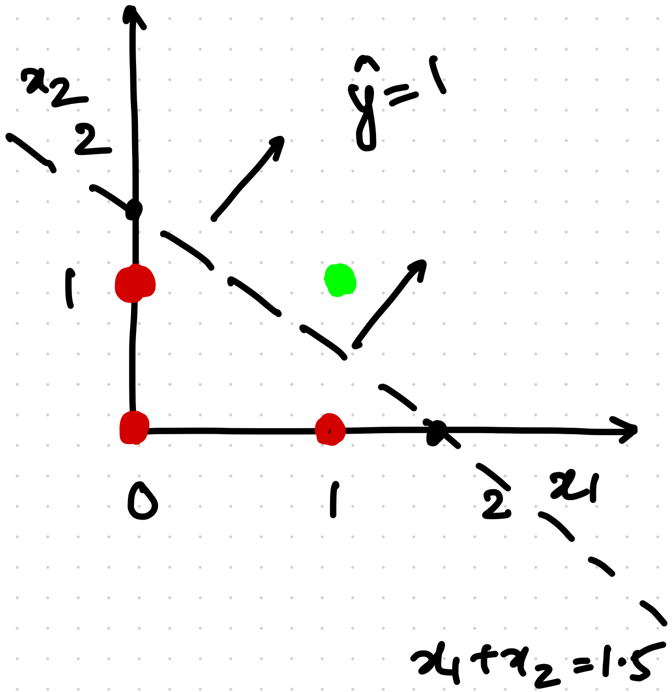
x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

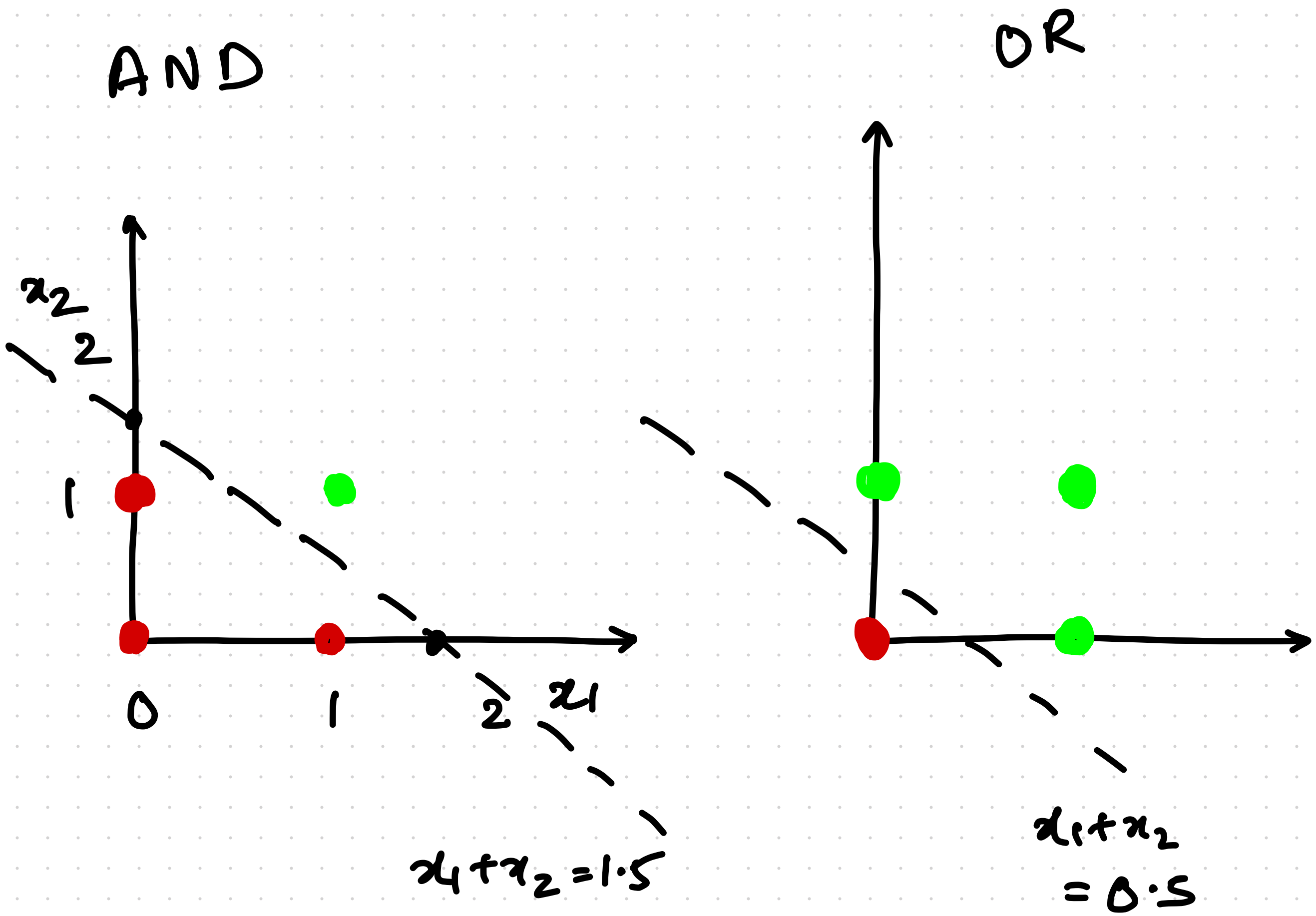


AND

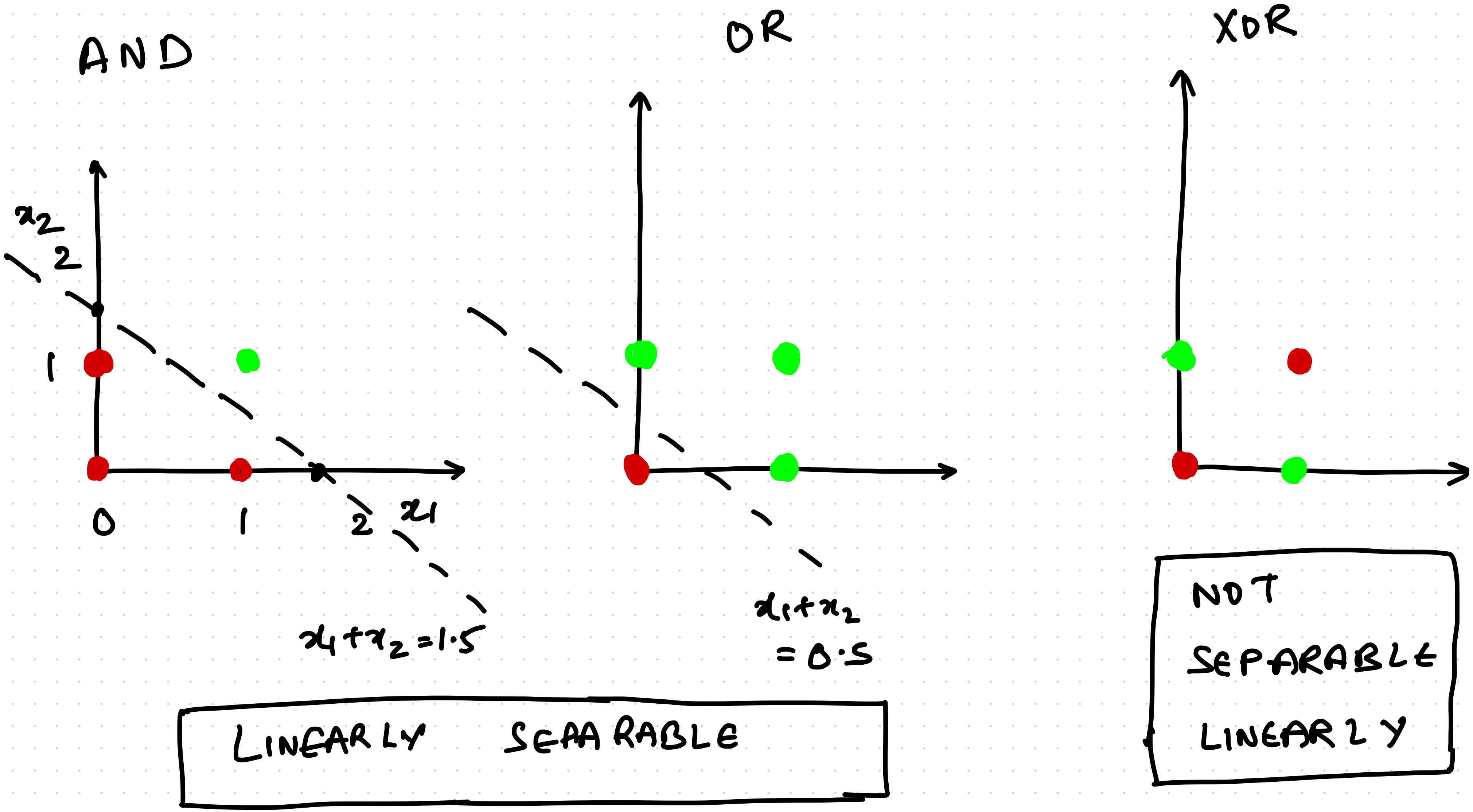


AND

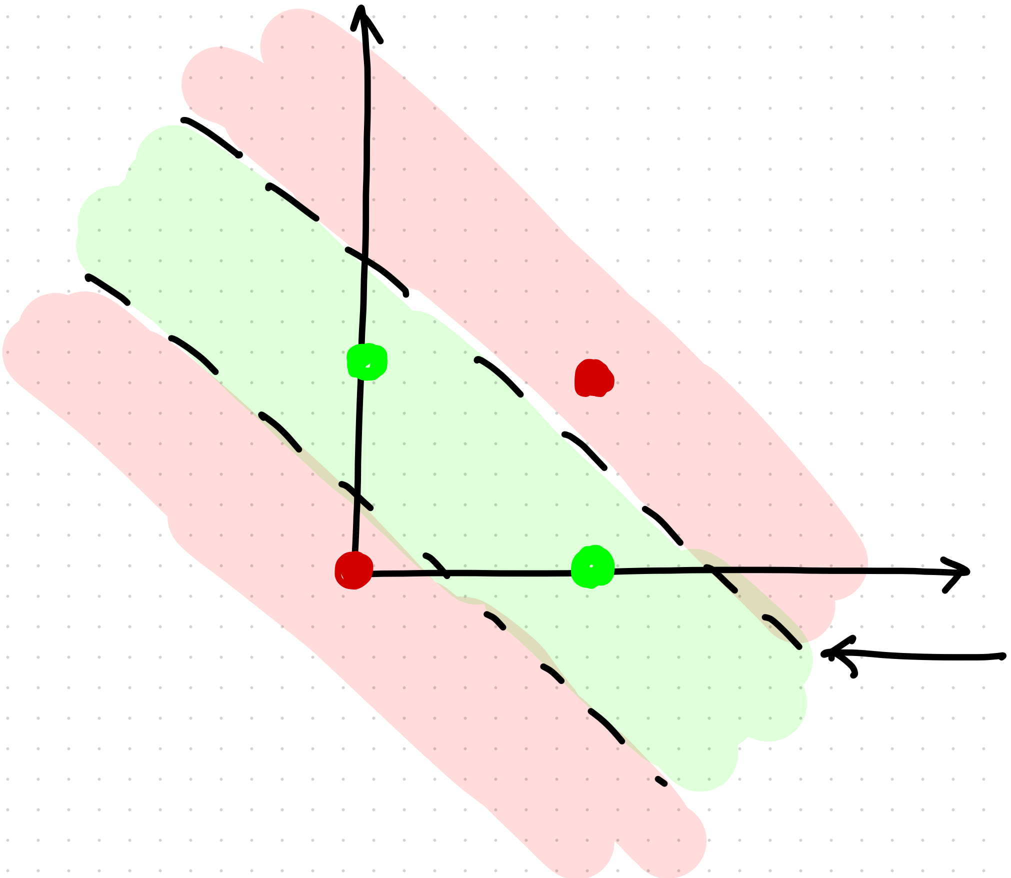




AND
OR
x_2
2
1
0
1
2
x_1
$x_1+x_2=1.5$
x_1+x_2
$=0.5$

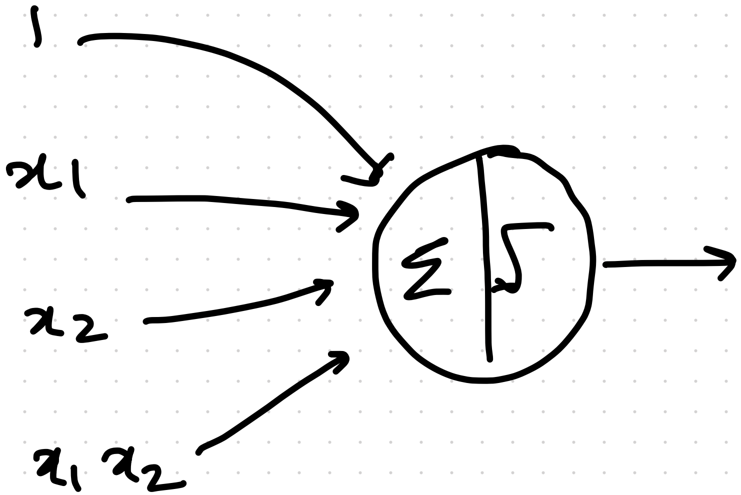

AND
OR
XOR
x2
2
1
0
1
2
x1
x1+x2=1.5
x1+x2
=0.5
LINEARLY SEPARABLE
NOT
SEPARABLE
LINEARLY

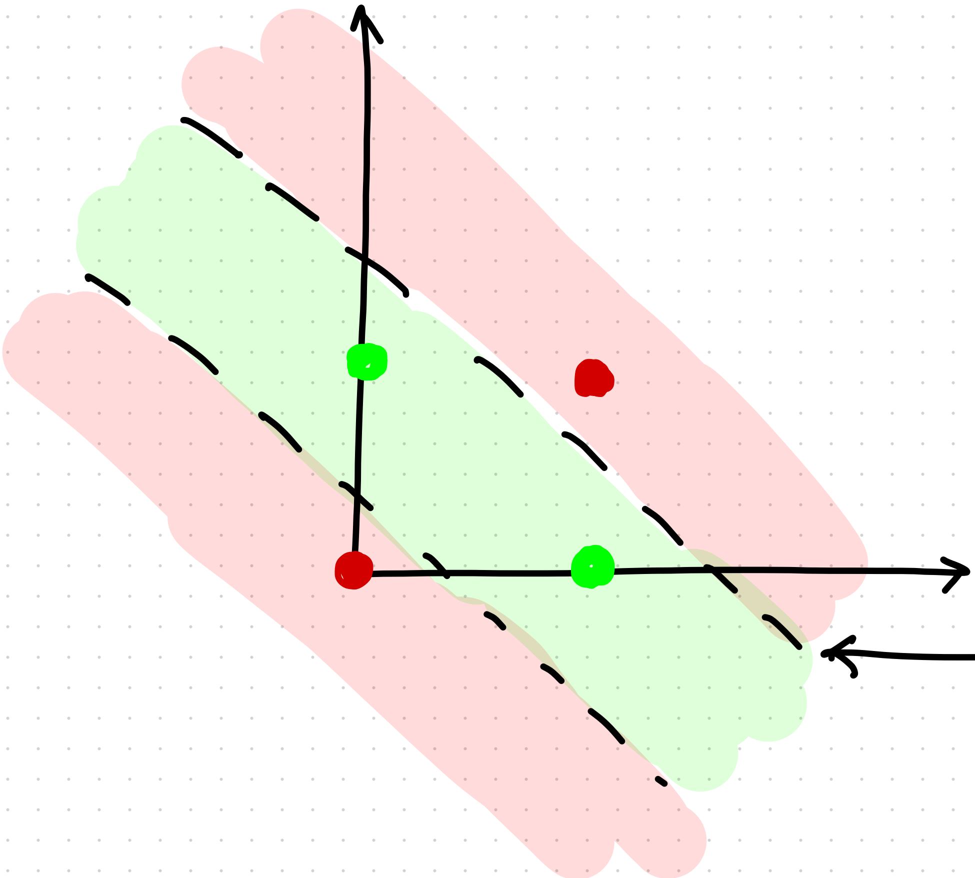
XOR CLASSIFICATION



Decision surface desired

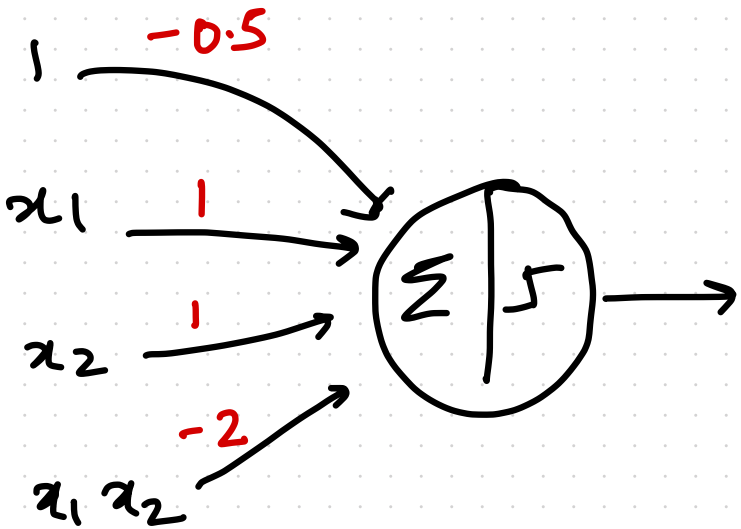
XOR CLASSIFICATION

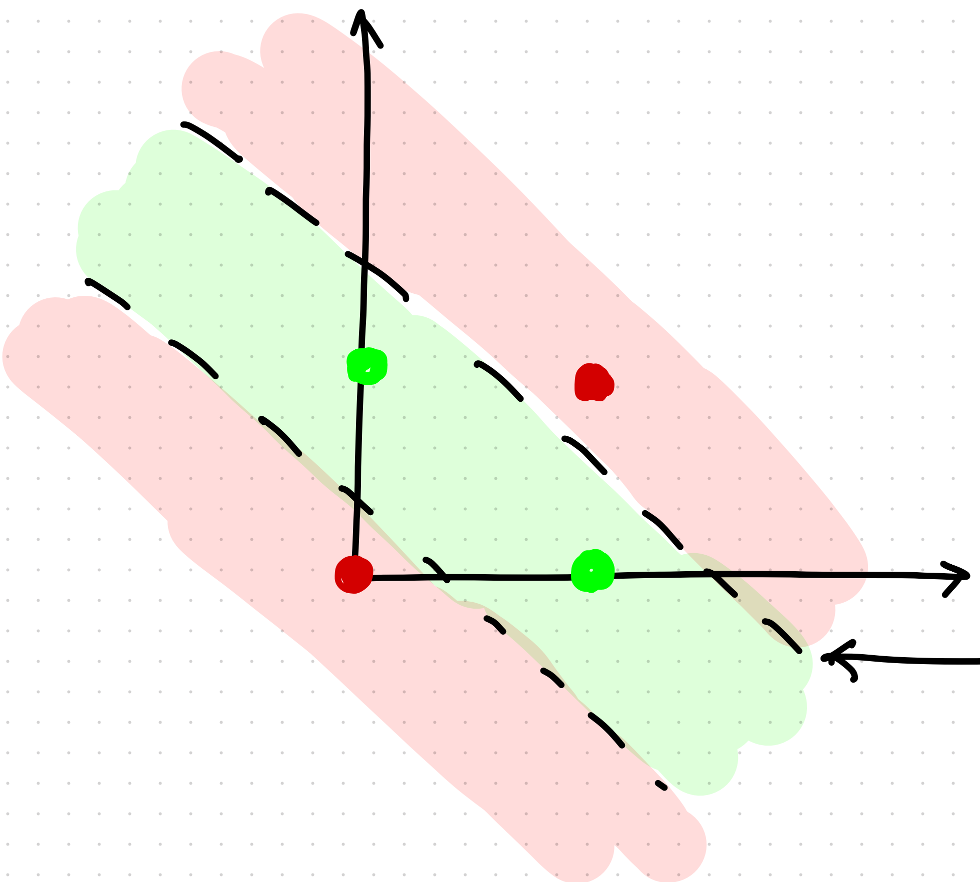




Decision surface desired

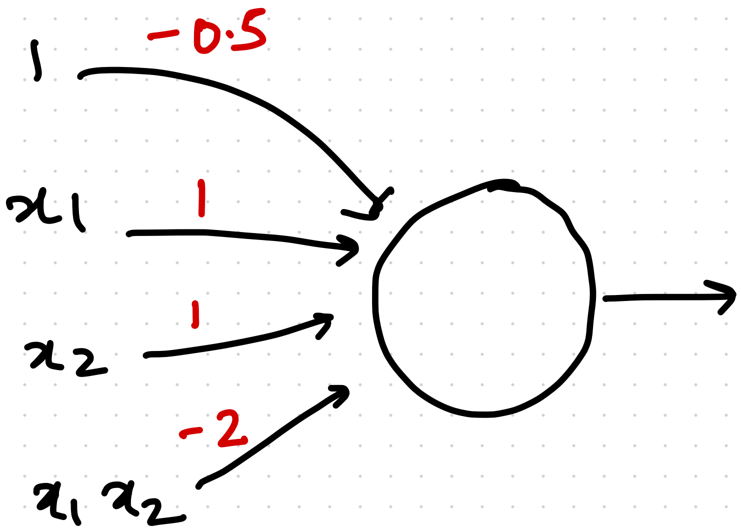
XOR CLASSIFICATION

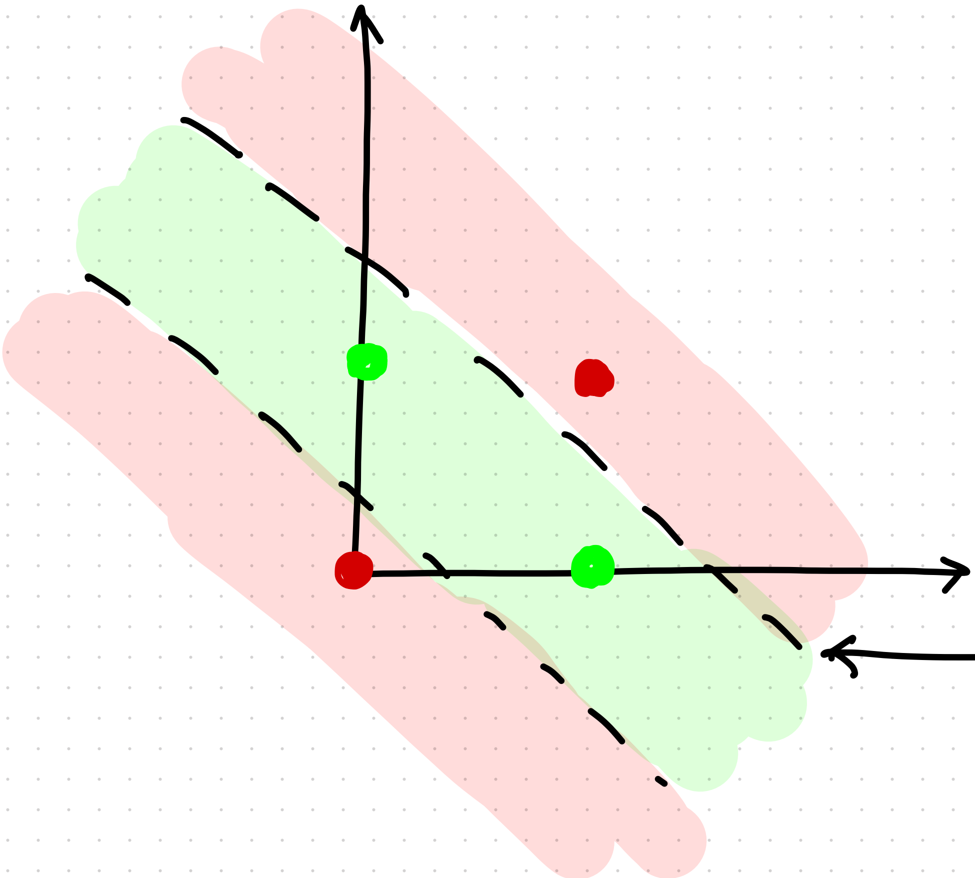




Decision surface desired

XOR CLASSIFICATION



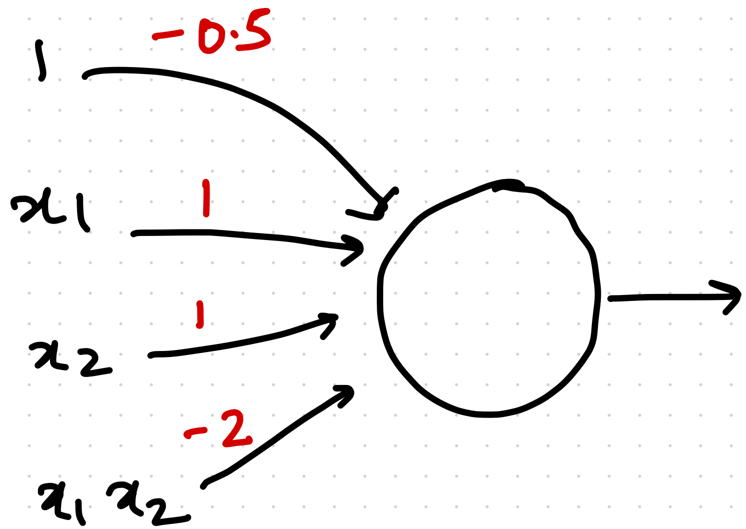


Decision surface desired

FOR $x_1 = 0$; $x_2 = 0$; we get

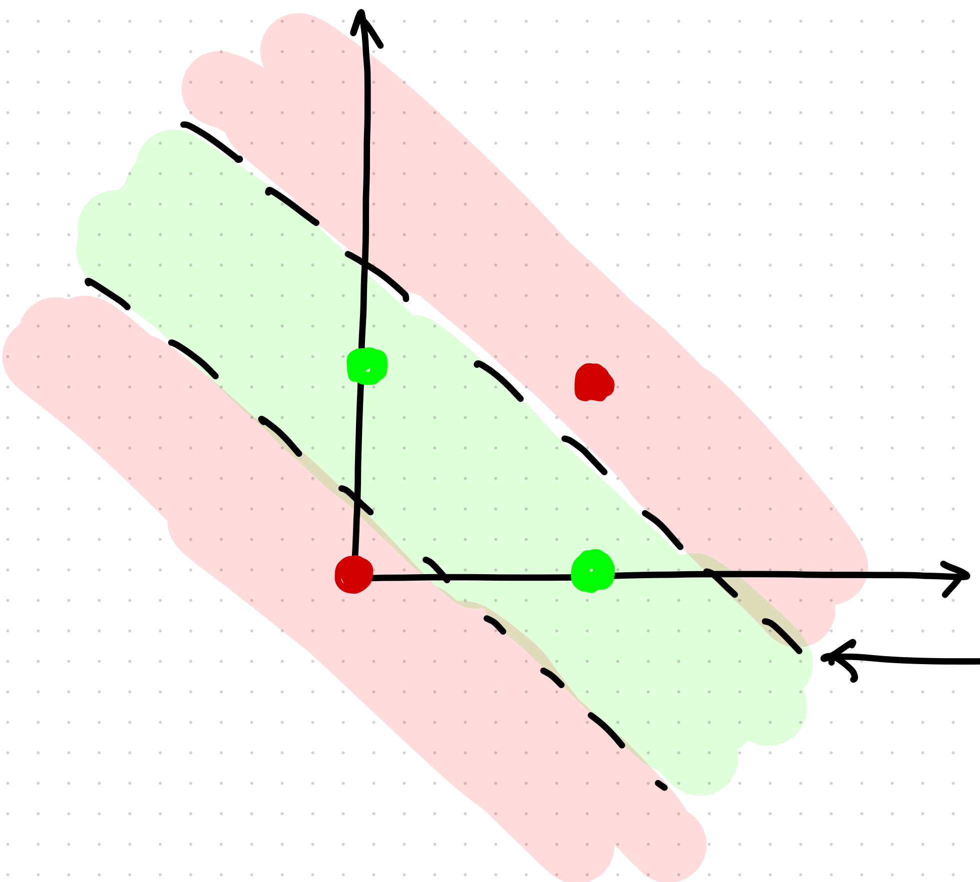
$\hat{y} = \{-0.5 \le 0\}$ = RED CLASS

XOR CLASSIFICATION



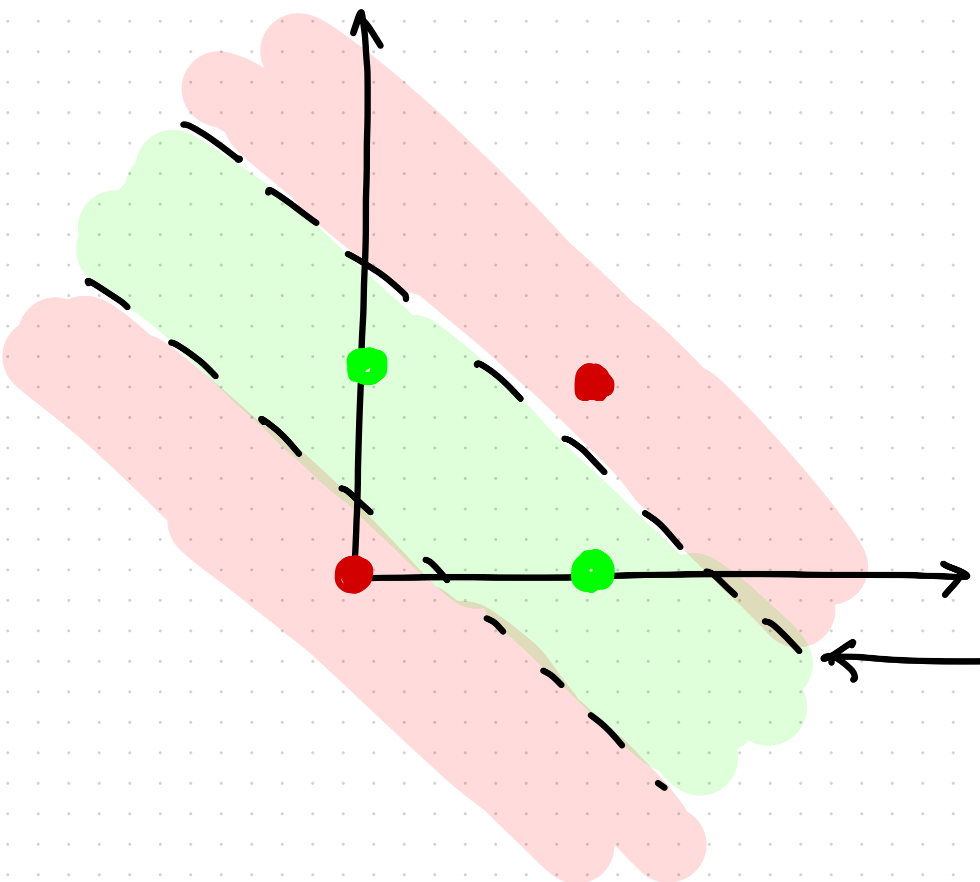
FOR $x_1 = 0$; $x_2 = 1$; we get

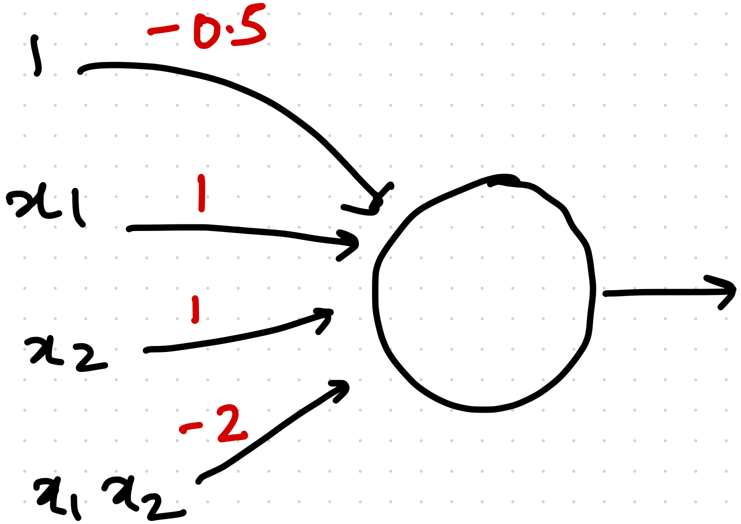
$\hat{y} = \{-0.5 + 1 \le 0\} =$ GREEN CLASS



Decision surface desired

XOR CLASSIFICATION



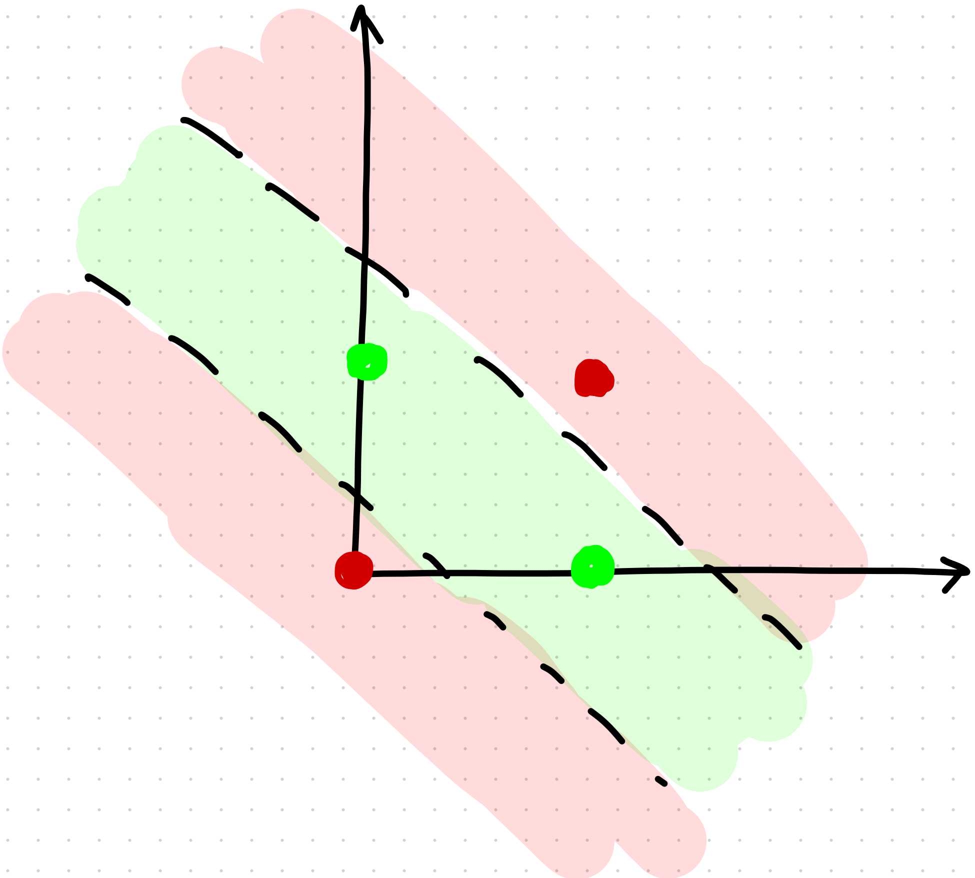


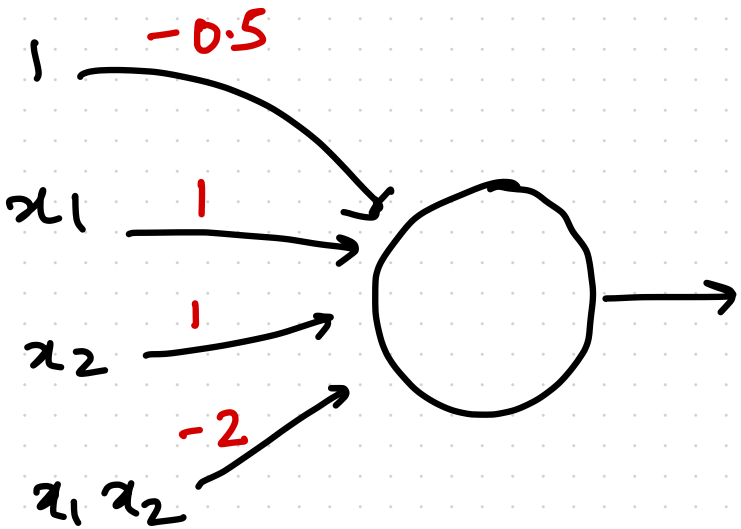
FOR $x_1 = 1$; $x_2 = 0$; we get

$\hat{y} = \{0.5 \le 0\} =$ GREEN CLASS

Decision surface desired

XOR CLASSIFICATION



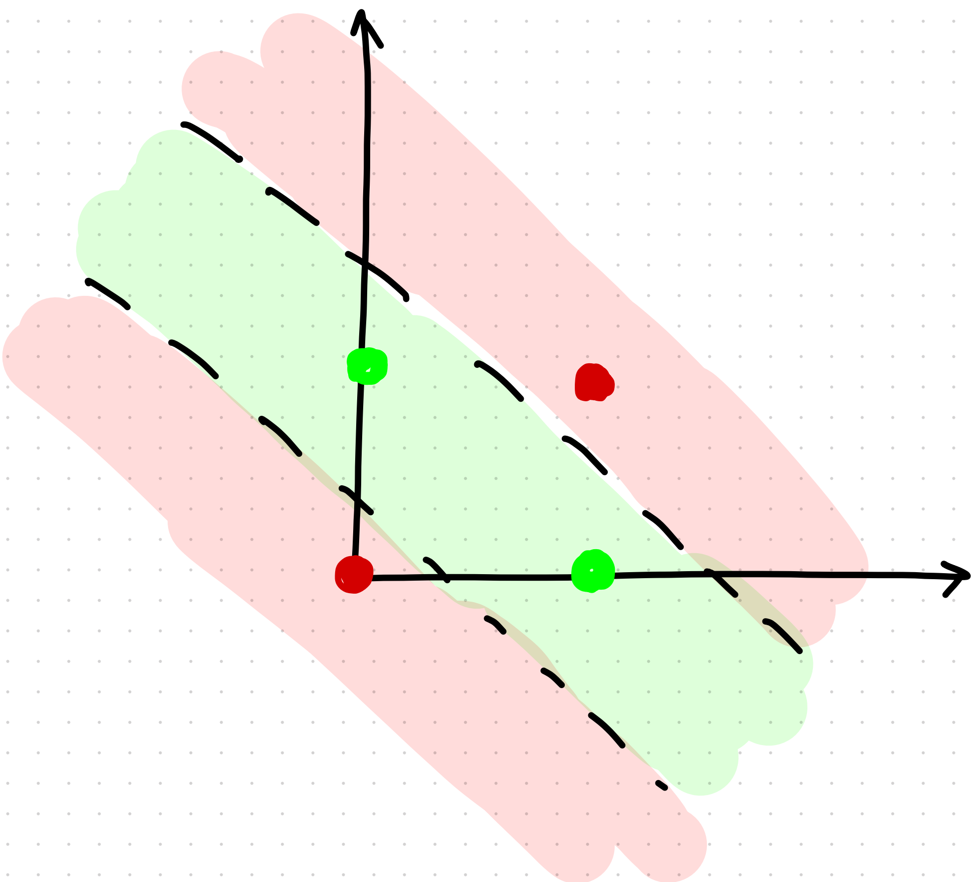


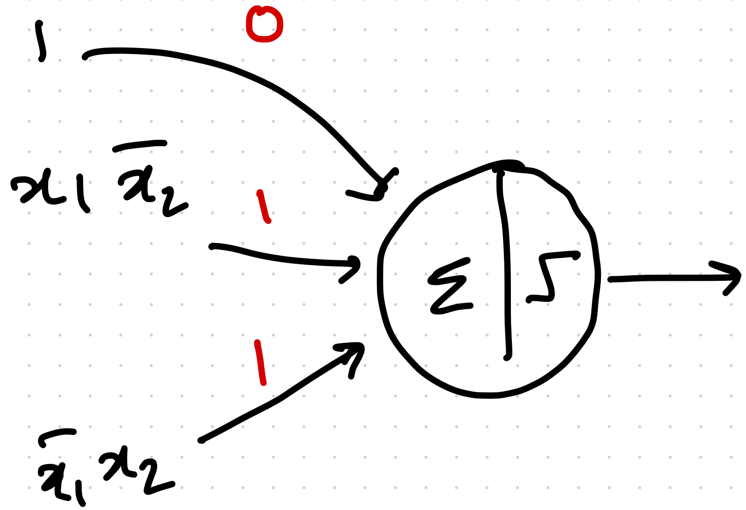
FOR $x_1 = 1$; $x_2 = 1$; we get

$\hat{y} = \{-0.5 \le 0\} =$ RED CLASS

CAN ADD NON-LINEARITY BY HAND-CRAFTING FEATURES !

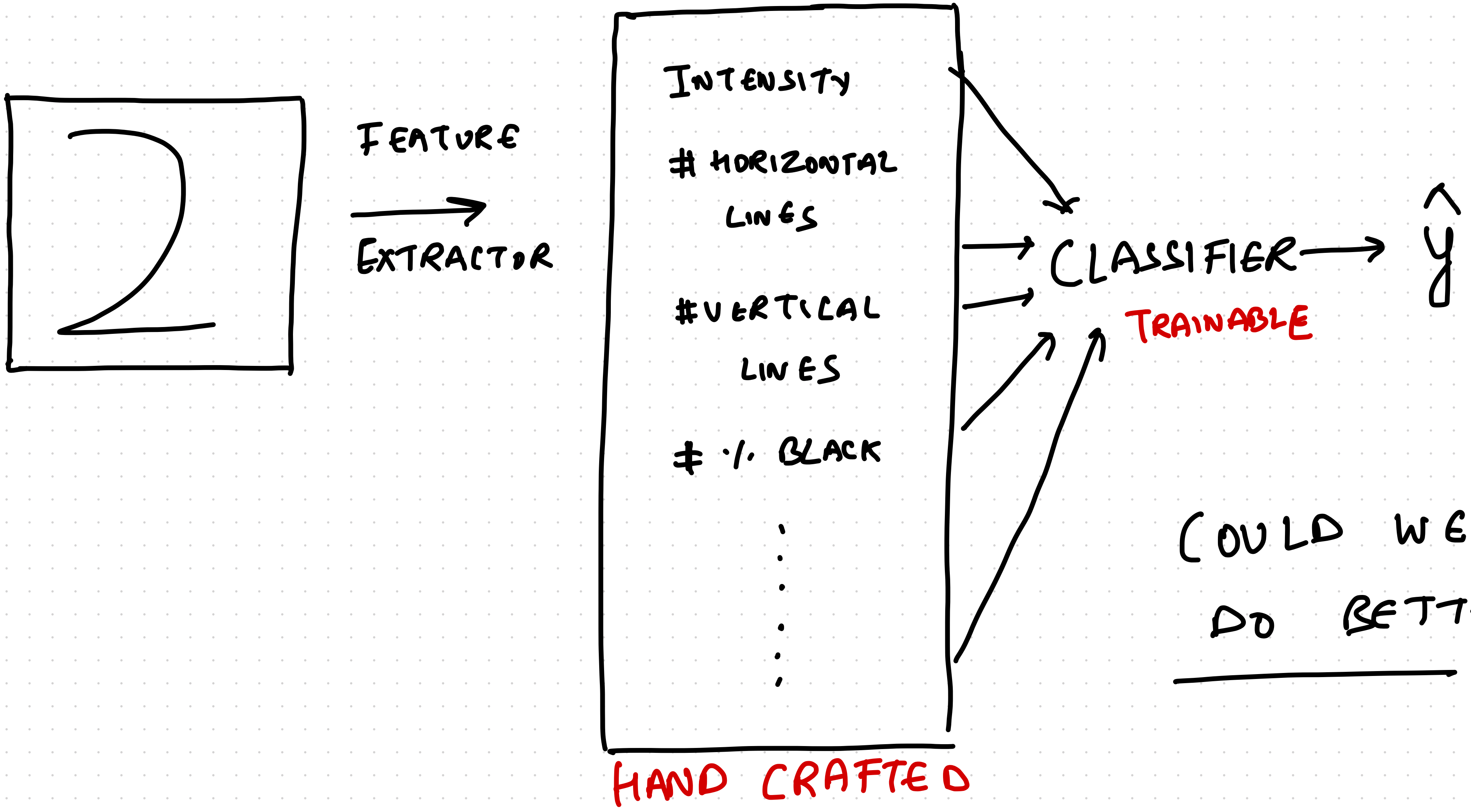
XOR CLASSIFICATION





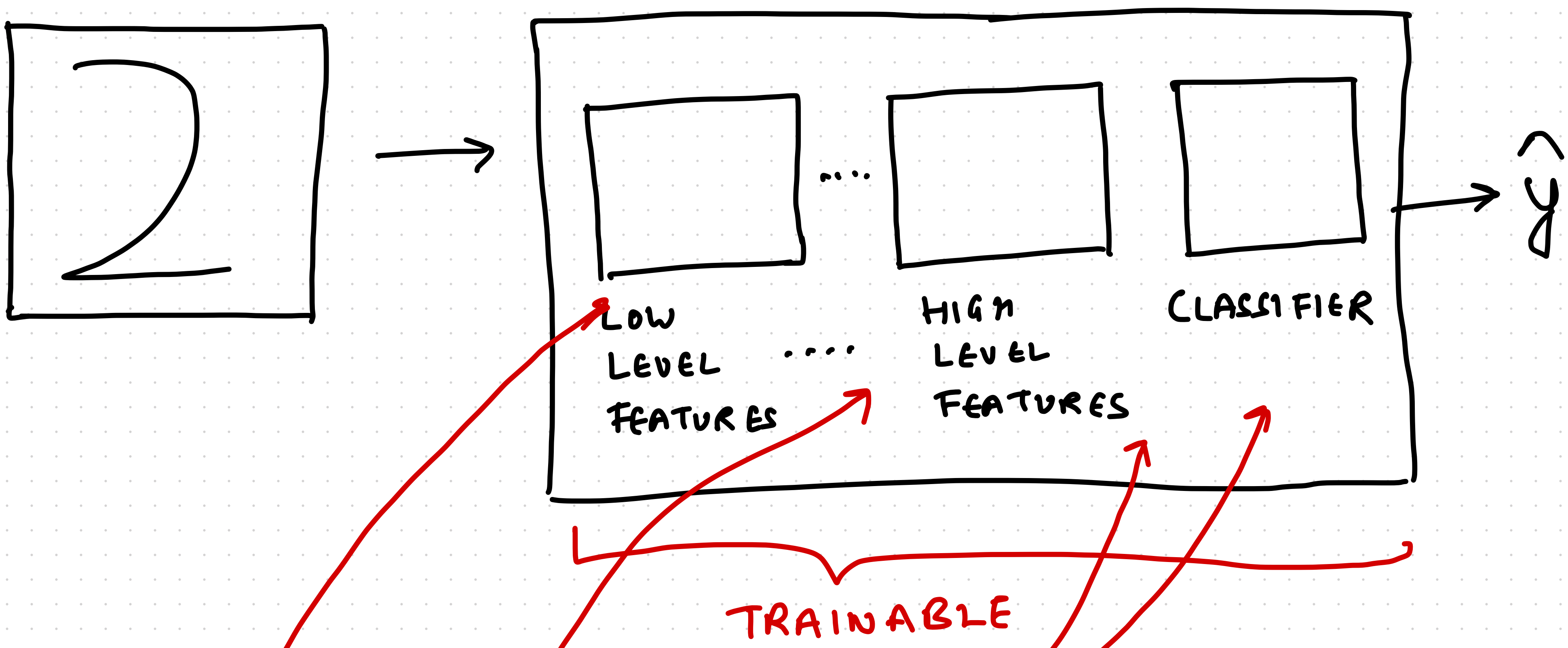
CAN ADD NON-LINEARITY BY HAND-CRAFTING FEATURES!

PARADIGM CHANGE



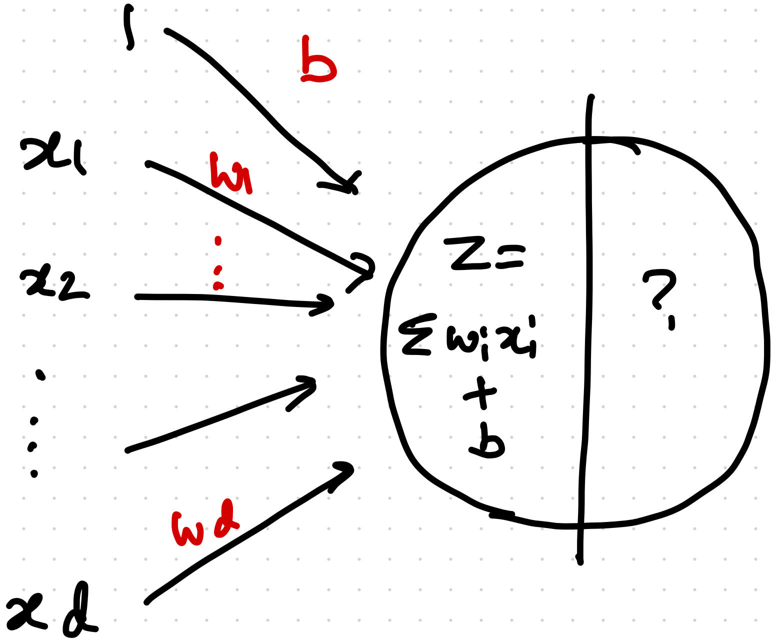
COULD WE DO BETTER?

PARADIGM CHANGE (NNS)



WE NEED NON-LINEARITY

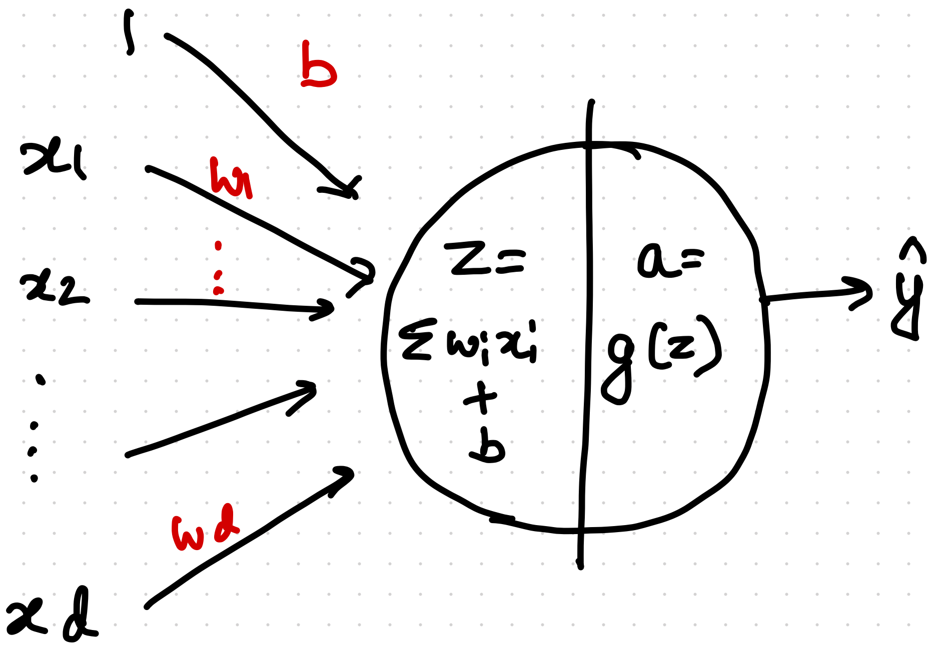
BACKPROPAGATION SUPPORTED ACTIVATIONS



key idea: use activation similar to

but differentiable

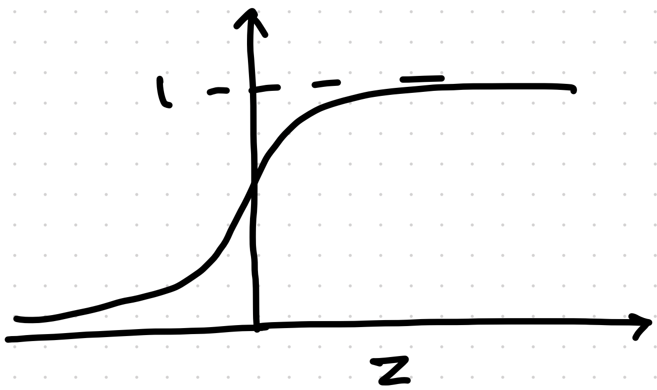
ADDING A NON-LINEARITY



$g(z)$: NON LINEAR TRANSFORMATION

ACTIVATION FUNCTIONS

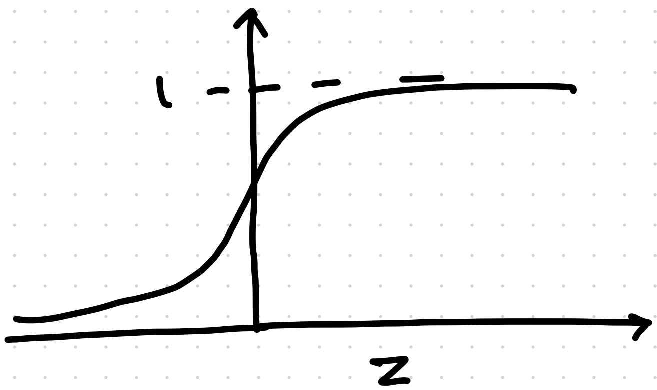
SIGMOID



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

ACTIVATION FUNCTIONS

SIGMOID



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

Q): If we have 1 Neuron

&

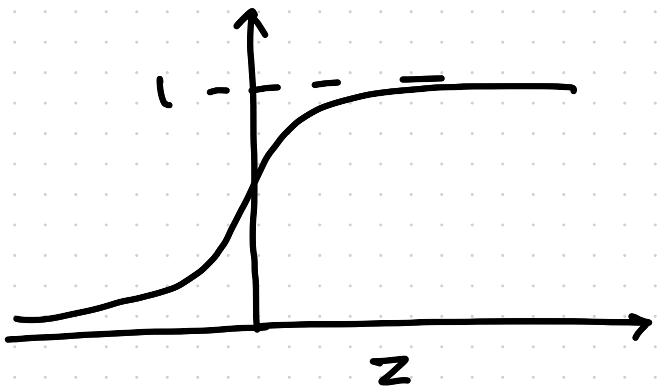
$$g(z) = \frac{1}{1+e^{-z}}$$

what do we

get?

ACTIVATION FUNCTIONS

SIGMOID



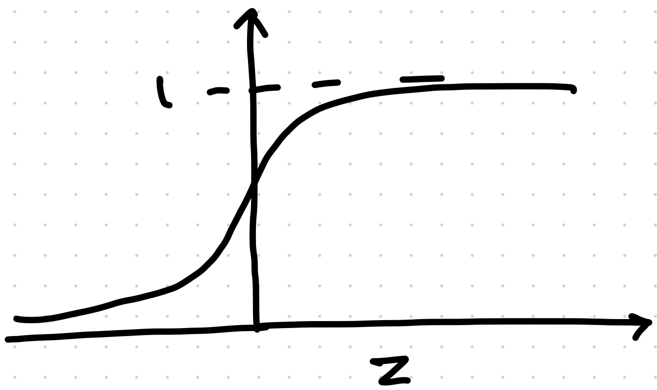
$$g(z) = \frac{1}{1+e^{-z}}$$

Q): If we have 1 Neuron & $g(z) = \frac{1}{1+e^{-z}}$ what do we get?

Logistic Regression

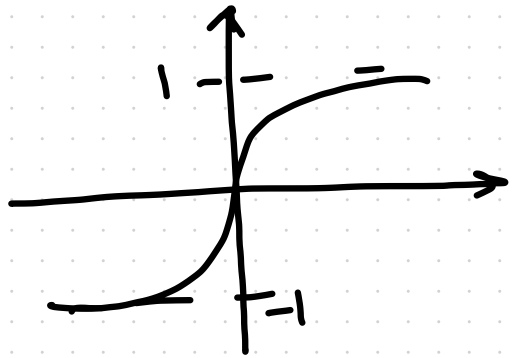
ACTIVATION FUNCTIONS

SIGMOID



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

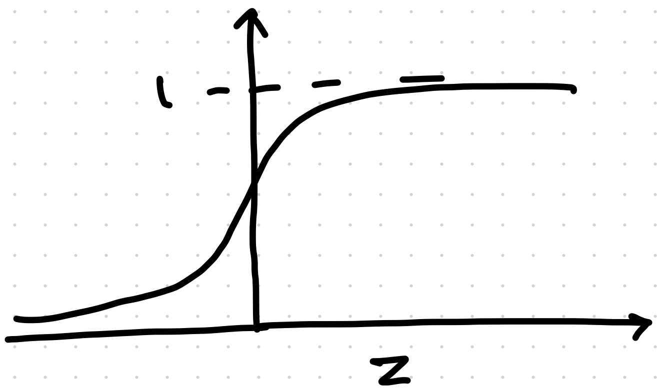
TANH



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

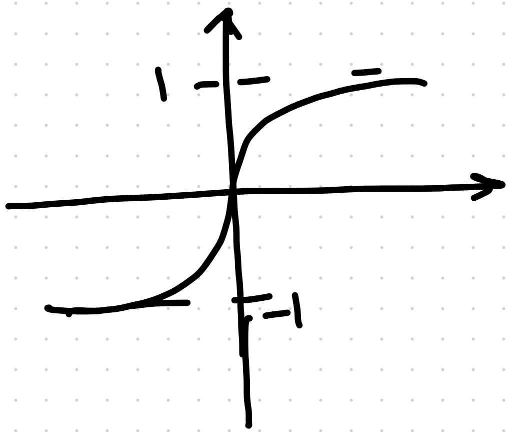
ACTIVATION FUNCTIONS

SIGMOID



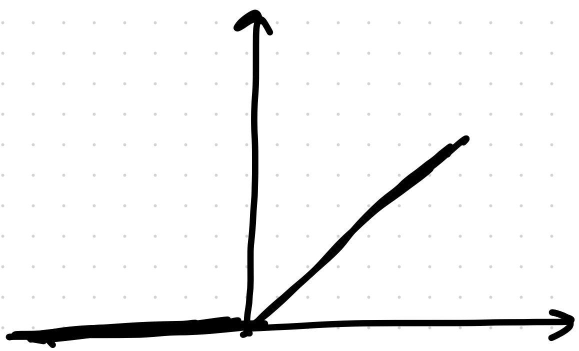
$$g(z) = \frac{1}{1+e^{-z}}$$

TANH



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

RELU



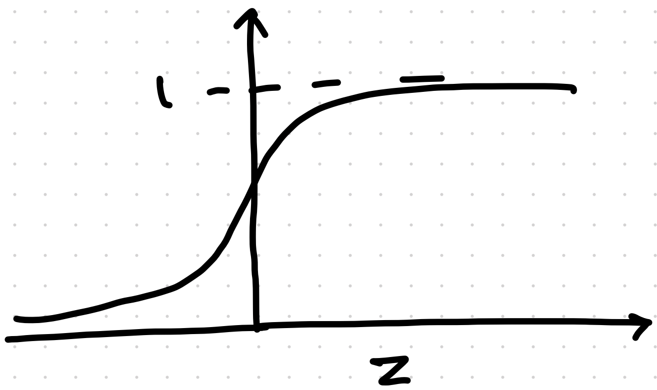
$$g(z) = \begin{cases} z; & z \geq 0 \\ 0; & o/w \end{cases}$$

or

$$g(z) = \max(0, z)$$

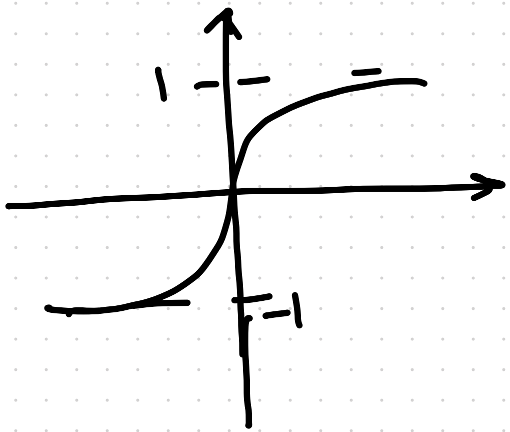
ACTIVATION FUNCTIONS

SIGMOID



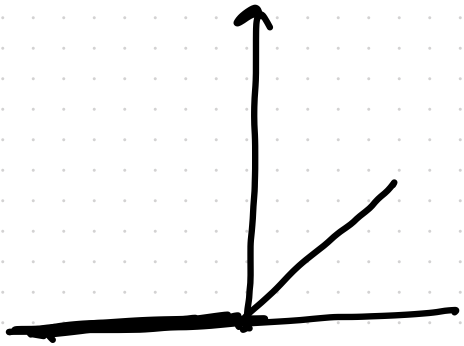
$$g(z) = \frac{1}{1+e^{-z}}$$

TANH



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

ReLU

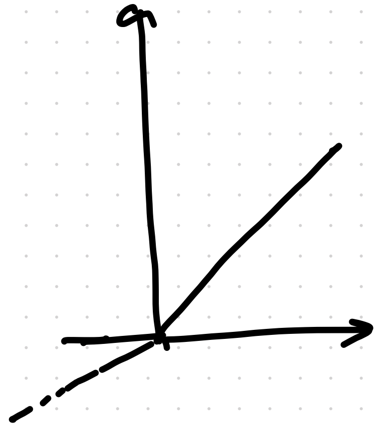


$$g(z) = \begin{cases} z; & z \geq 0 \\ 0; & o/w \end{cases}$$

or

$$g(z) = \mathrm{MAX}(0, z)$$

Leaky ReLU

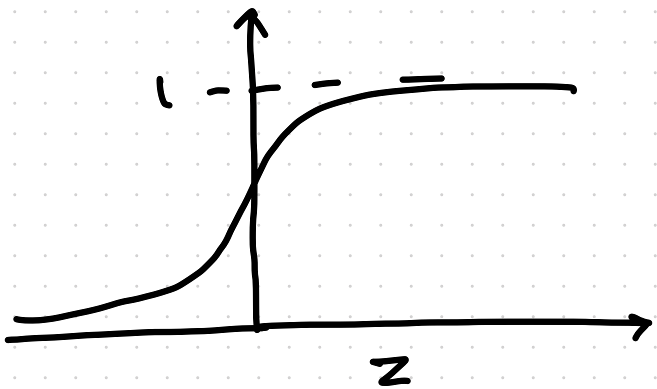


$$g(z) = \mathrm{MAX}(\alpha z, z)$$

$$\alpha \to 0$$

ACTIVATION FUNCTIONS

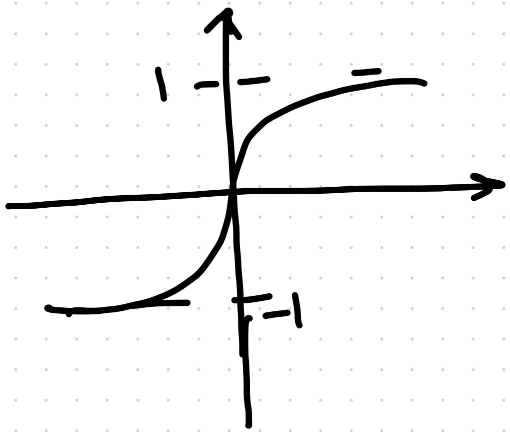
SIGMOID



USEFUL FOR PROBABILISTIC ESTIMAES

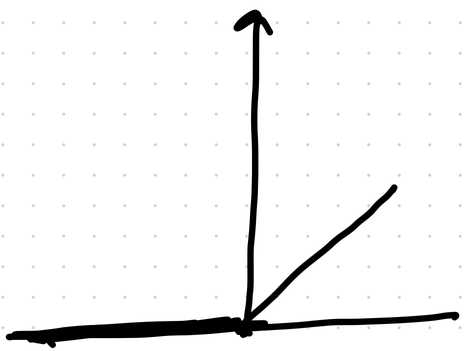
∴ B/W 0 & 1

TANH



USEFUL IF DATA TRANSFORMED WITH MEAN 0

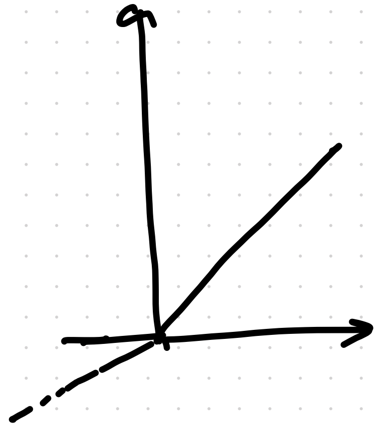
ReLU



GAME CHANGER (Default)

Good learning for |z| = high

Leaky ReLU



Similar to ReLU

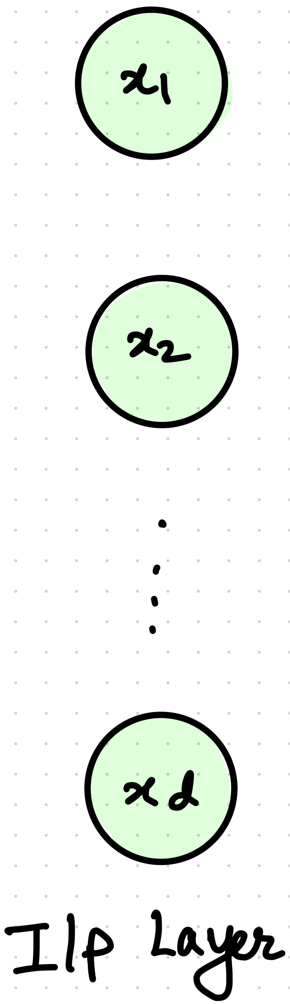
Learns for z < 0 also

DESIRABLE ATTRIBUTES OF ACTIVATION FUNCTIONS

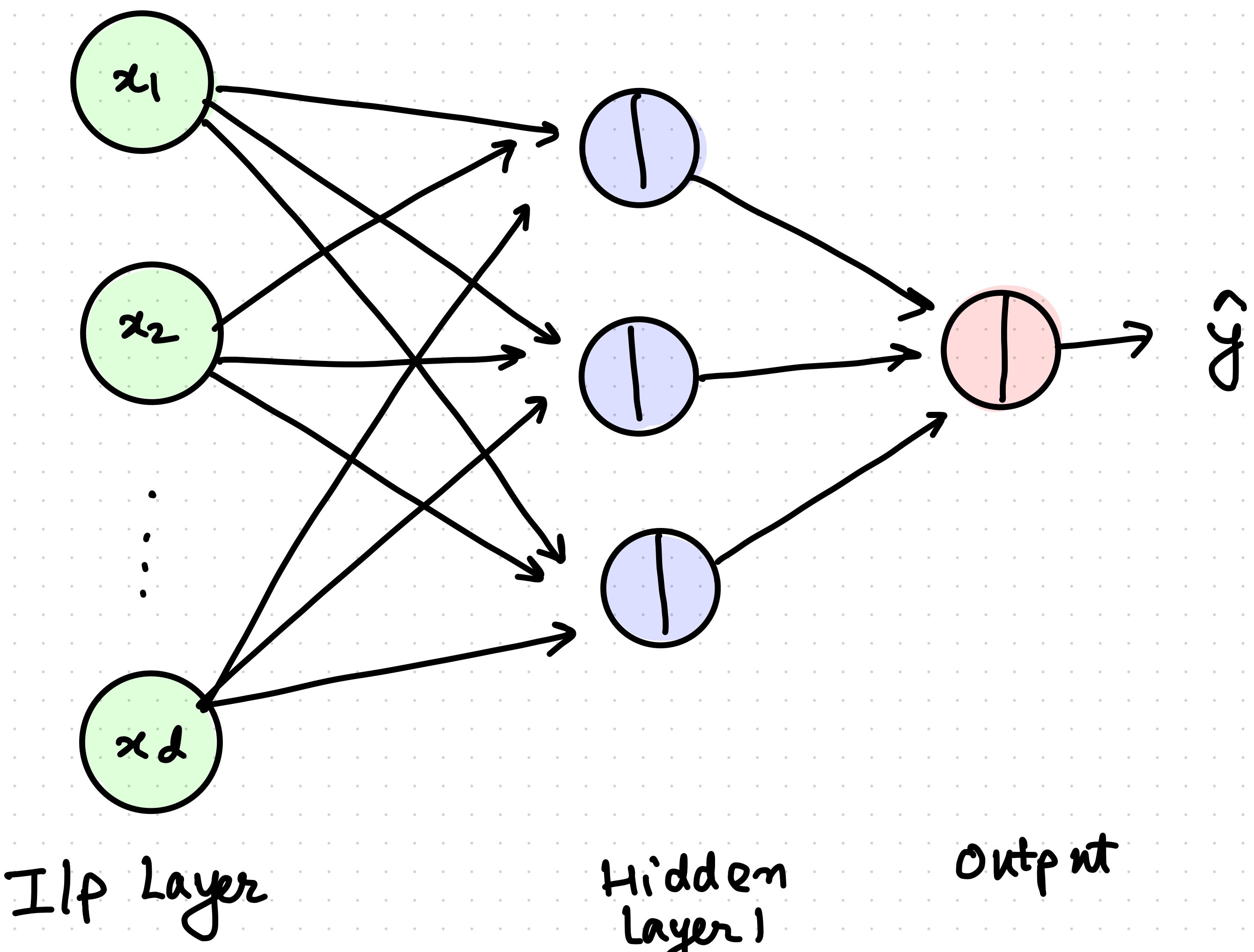
1) NON- LINEAR

2) (MOSTLY) SMALL CHANGE IN I/P $\Rightarrow$ SMALL CHANGE IN O/P

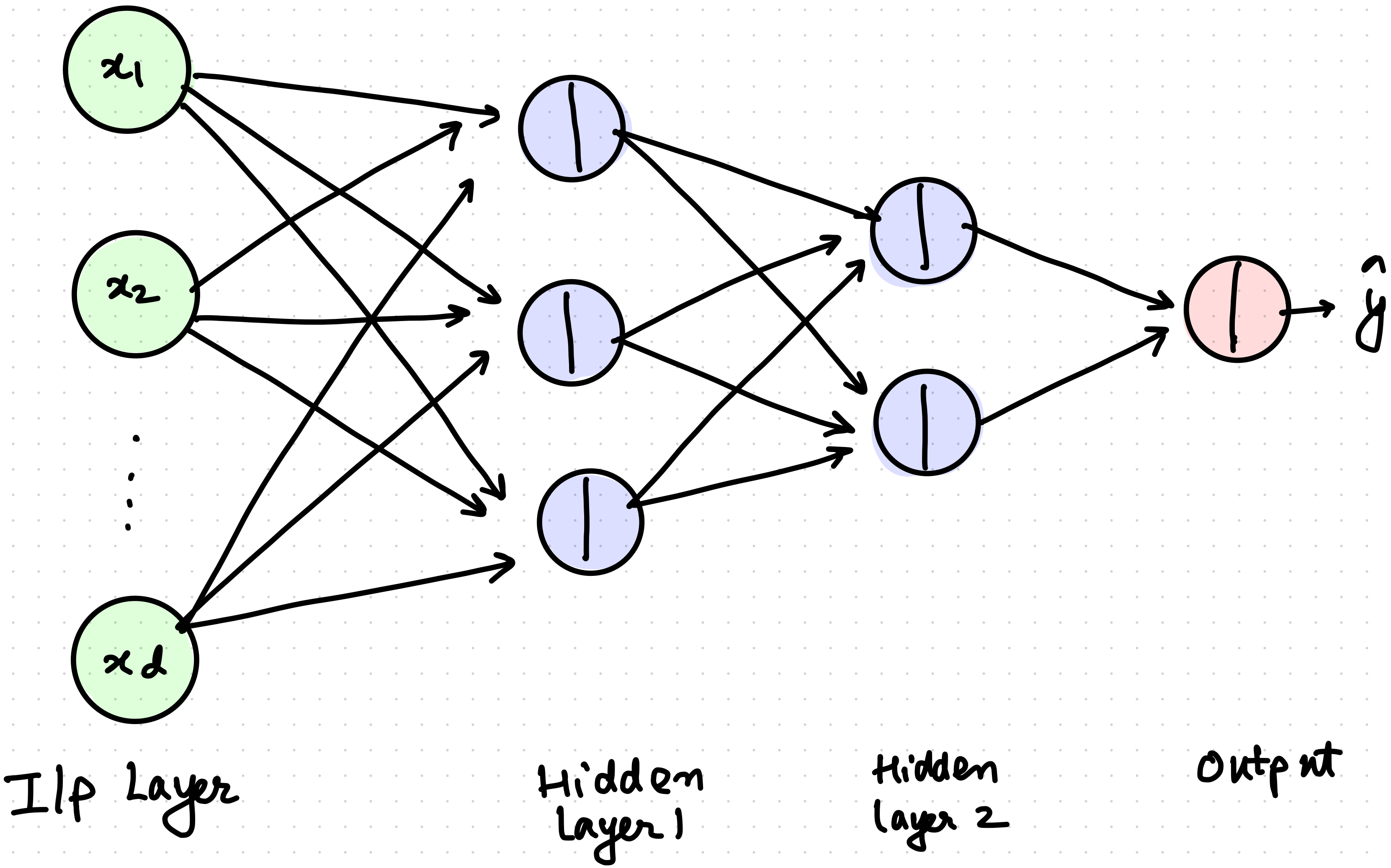
1 LAYER PERCEPTRON (NN)



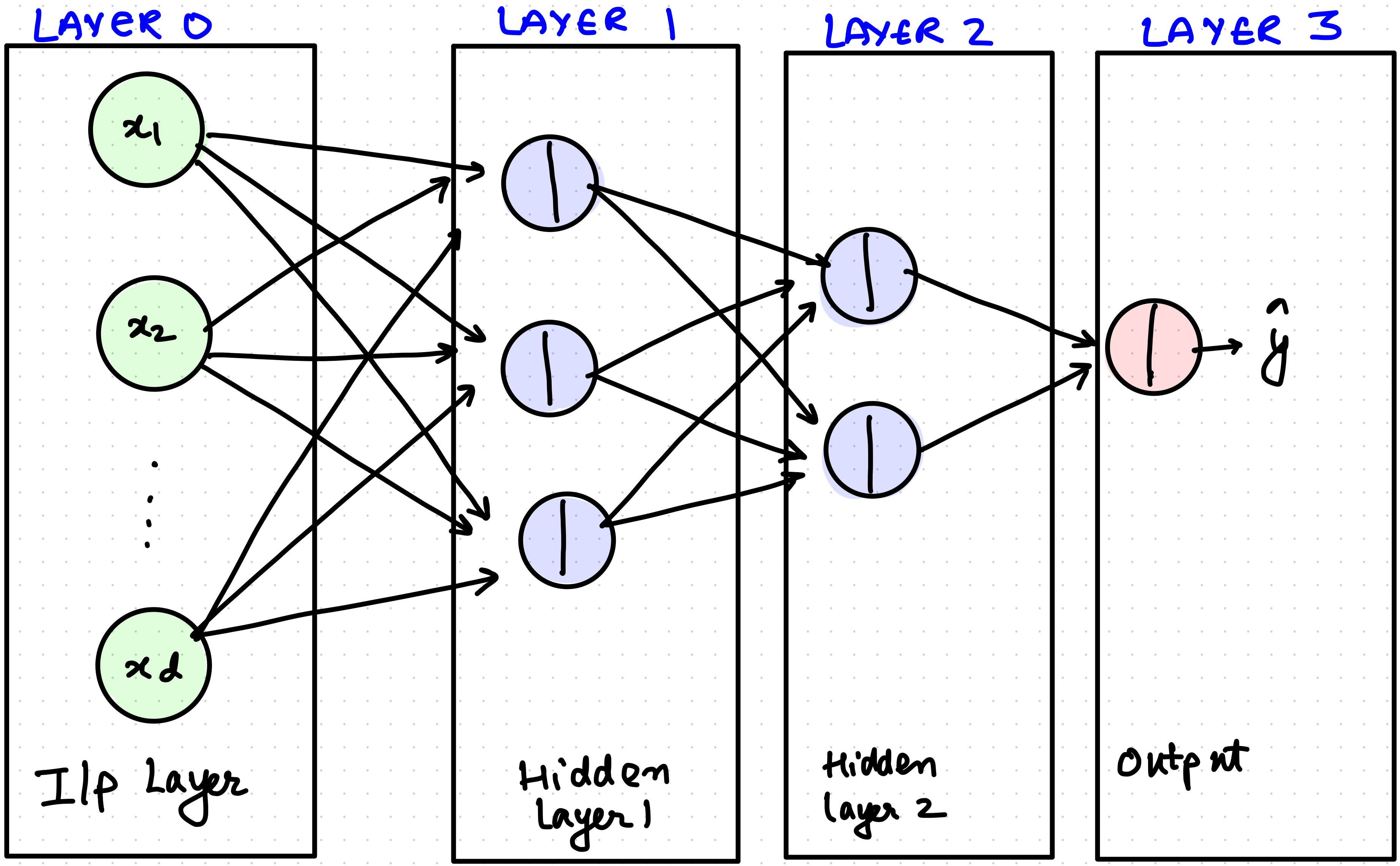
1- LAYER PERCEPTRON (NN)

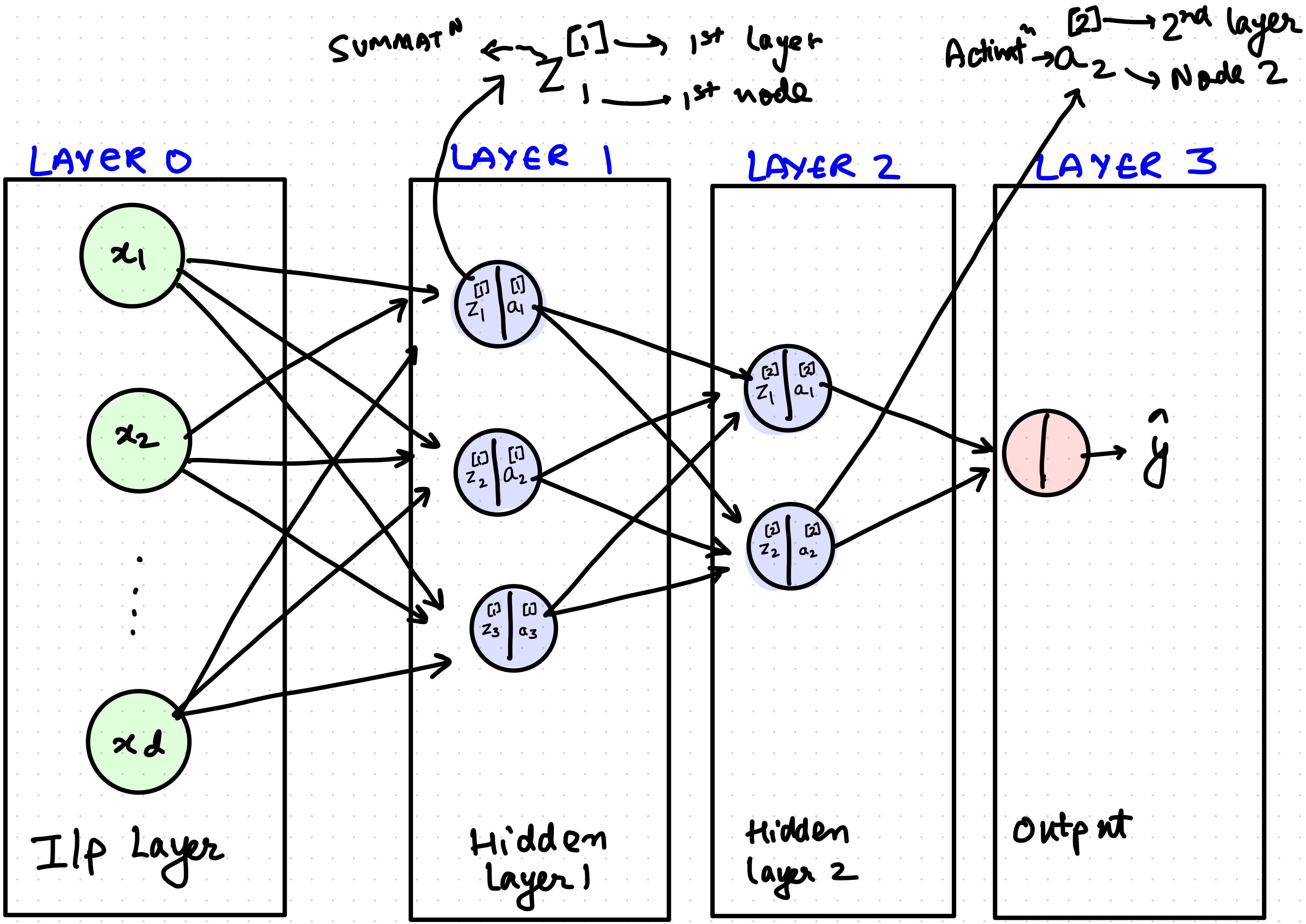


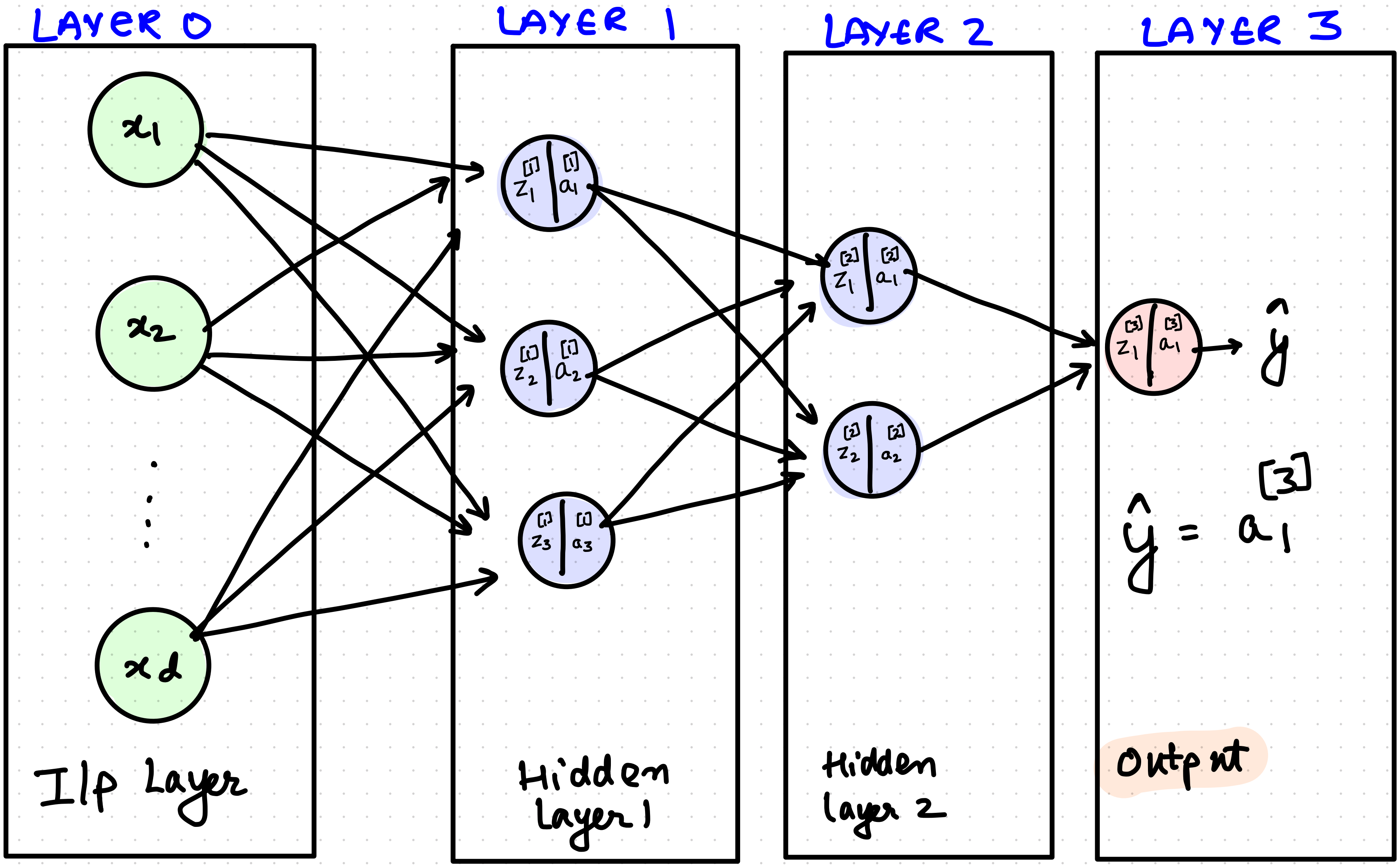
MULTI - LAYER PERCEPTRON



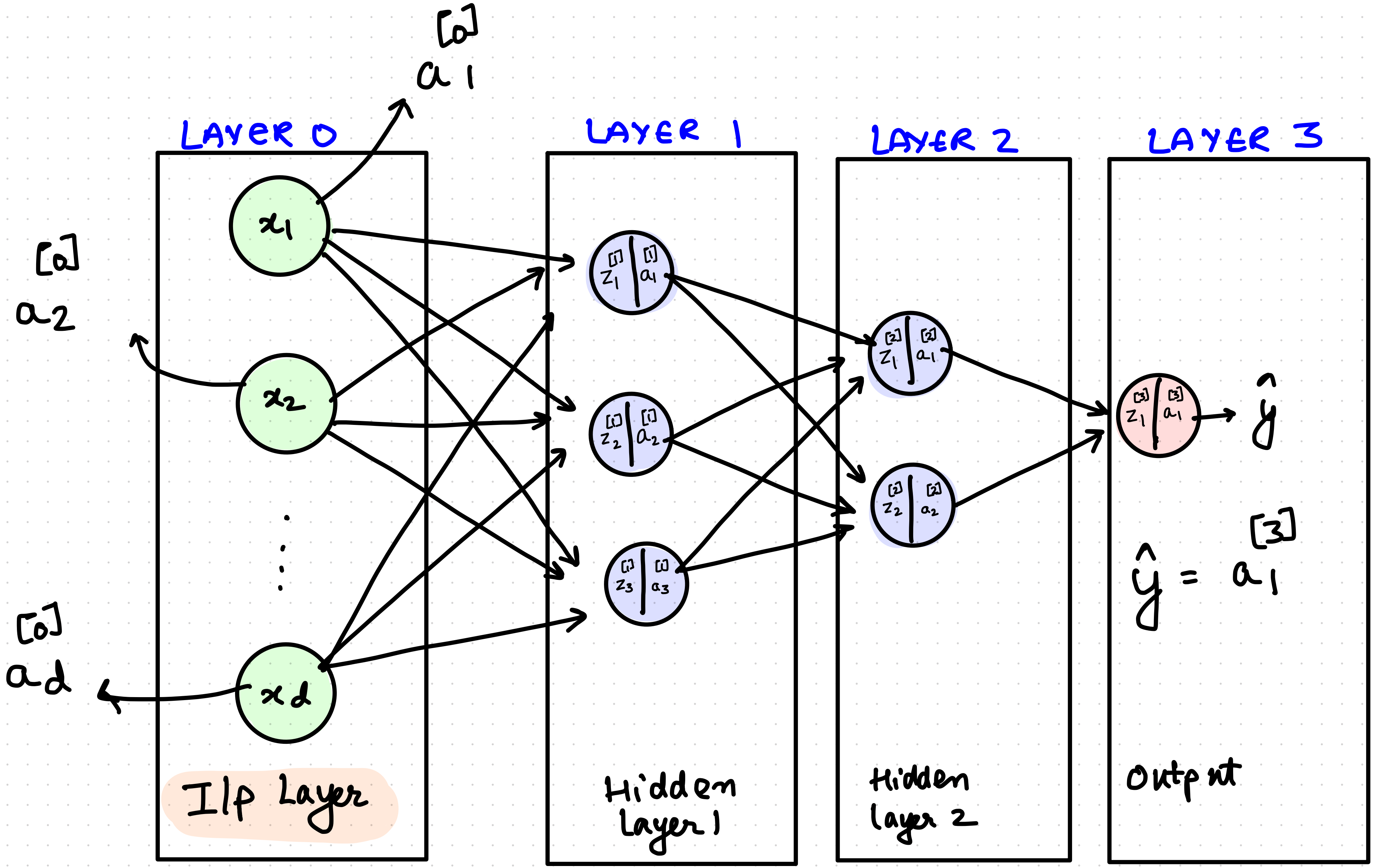
MULTI-LAYER PERCEPTRON




SUMMATN
1st layer
1st node
Actinⁿ
2nd layer
Node 2
LAYER 0
LAYER 1
LAYER 2
LAYER 3
I/p Layer
Hidden layer 1
Hidden layer 2
Output

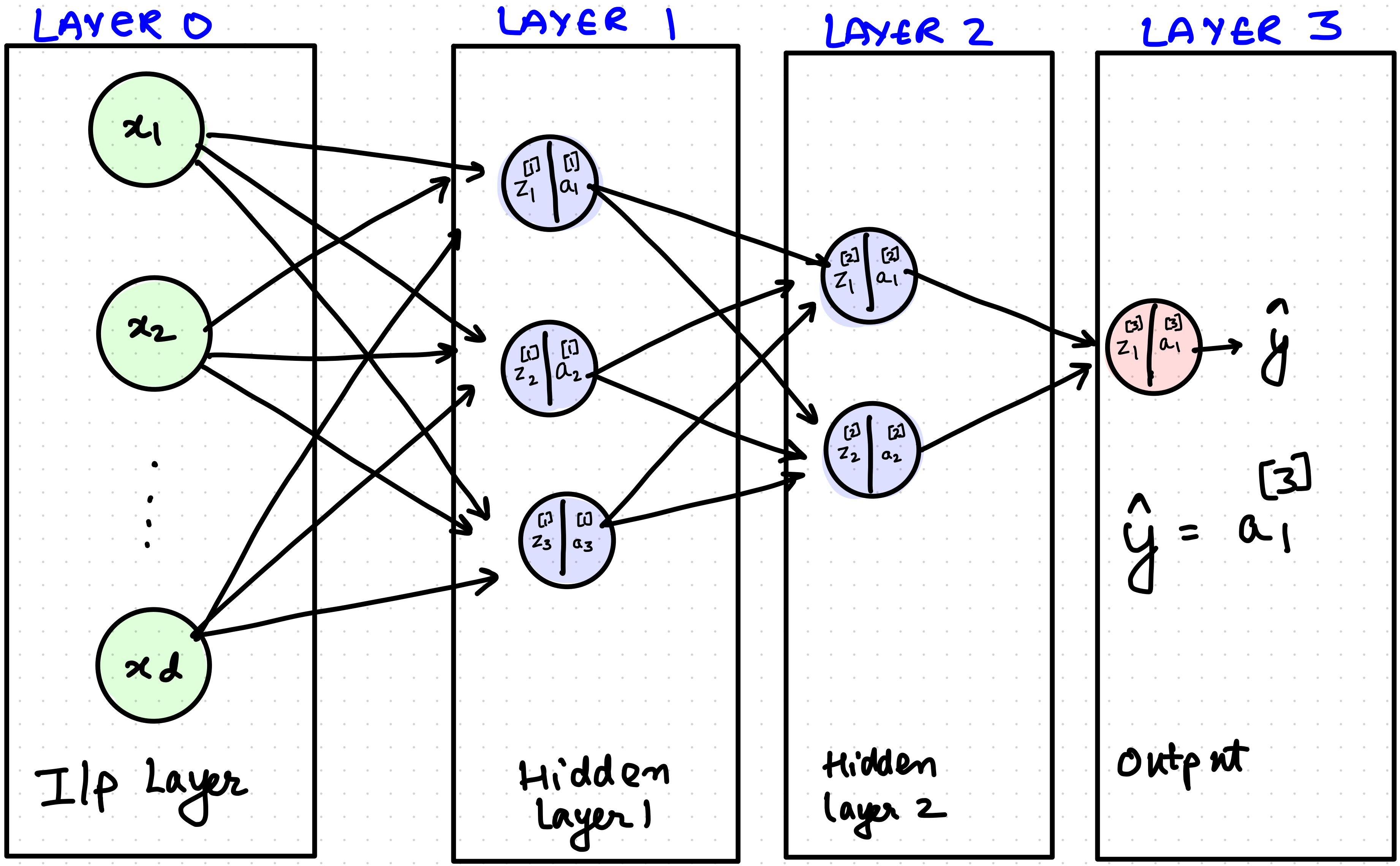


LAYER 0
LAYER 1
LAYER 2
LAYER 3
x_1
x_2
x_d
I/p Layer
$z_1^{[1]}$ $a_1^{[1]}$
$z_2^{[1]}$ $a_2^{[1]}$
$z_3^{[1]}$ $a_3^{[1]}$
Hidden layer 1
$z_1^{[2]}$ $a_1^{[2]}$
$z_2^{[2]}$ $a_2^{[2]}$
Hidden layer 2
$z_1^{[3]}$ $a_1^{[3]}$
$\hat{y}$
$\hat{y} = a_1^{[3]}$
Output

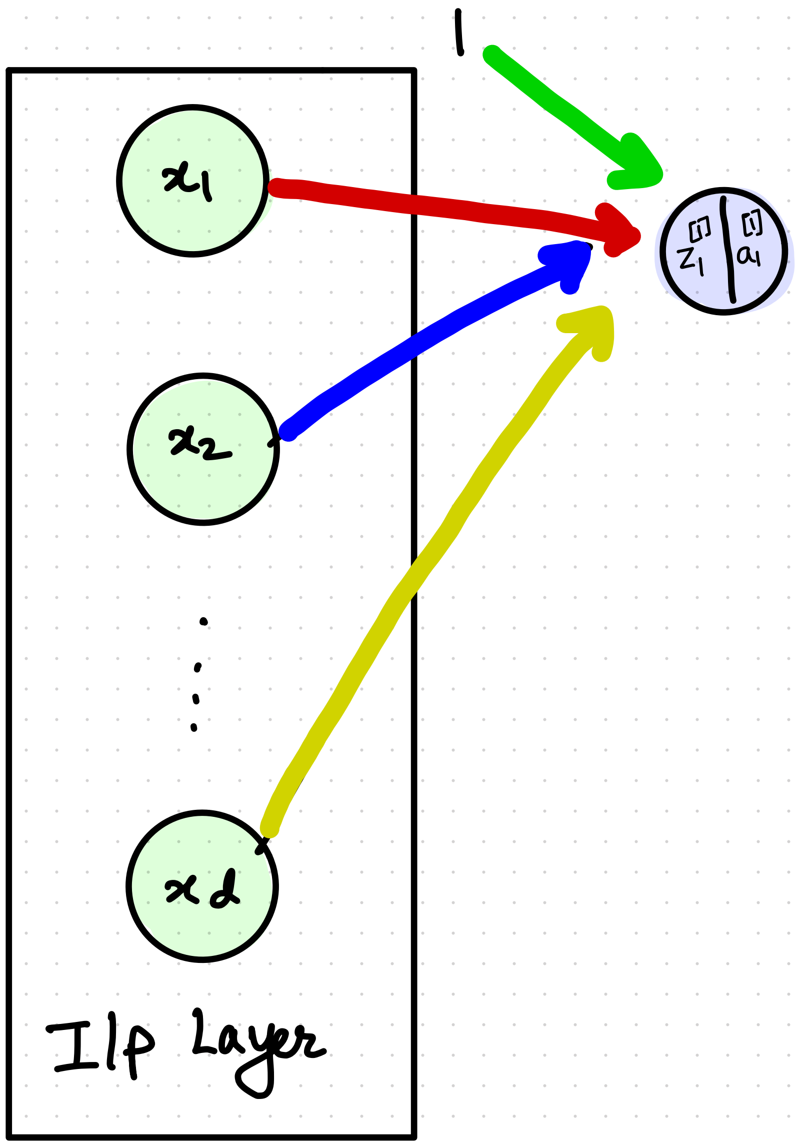


LAYER 0
LAYER 1
LAYER 2
LAYER 3
$a_1^{[0]}$
$a_2^{[0]}$
$a_d^{[0]}$
x_1
x_2
x_d
$z_1^{[1]}$ $a_1^{[1]}$
$z_2^{[1]}$ $a_2^{[1]}$
$z_3^{[1]}$ $a_3^{[1]}$
$z_1^{[2]}$ $a_1^{[2]}$
$z_2^{[2]}$ $a_2^{[2]}$
$z_1^{[3]}$ $a_1^{[3]}$
$\hat{y}$
$\hat{y} = a_1^{[3]}$
I/p Layer
Hidden layer 1
Hidden layer 2
Output

$$I/P = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} a_1^{[0]} \\ \vdots \\ a_d^{[0]} \end{bmatrix} = a^{[0]}_{d \times 1}$$



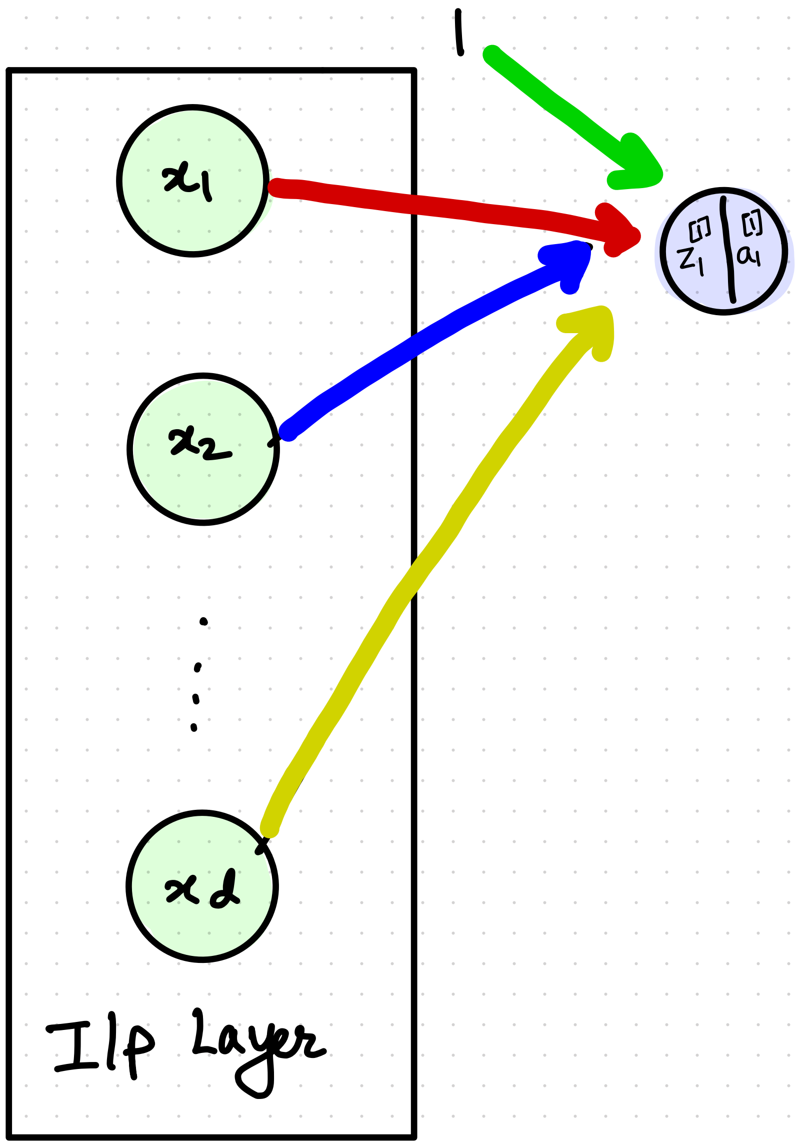
CONSIDER SINGLE NEURON (LAYER 1, NODE 1)



$$z_1^{[1]} = 1 * b_1^{[1]} + x_1 * w_{1,1}^{[1]} + x_2 * w_{1,2}^{[1]} + \cdots + xd * w_{1,d}^{[1]}$$

bias (layer, Node)

CONSIDER SINGLE NEURON (LAYER 1, NODE 1)



bias layer 1, Node 1

$$z^{[1]}_1 = 1 * b^{[1]}_1 + x_1 * w^{[1]}_{1,1} + x_2 * w^{[1]}_{1,2} + \dots + x_d * w^{[1]}_{1,d}$$

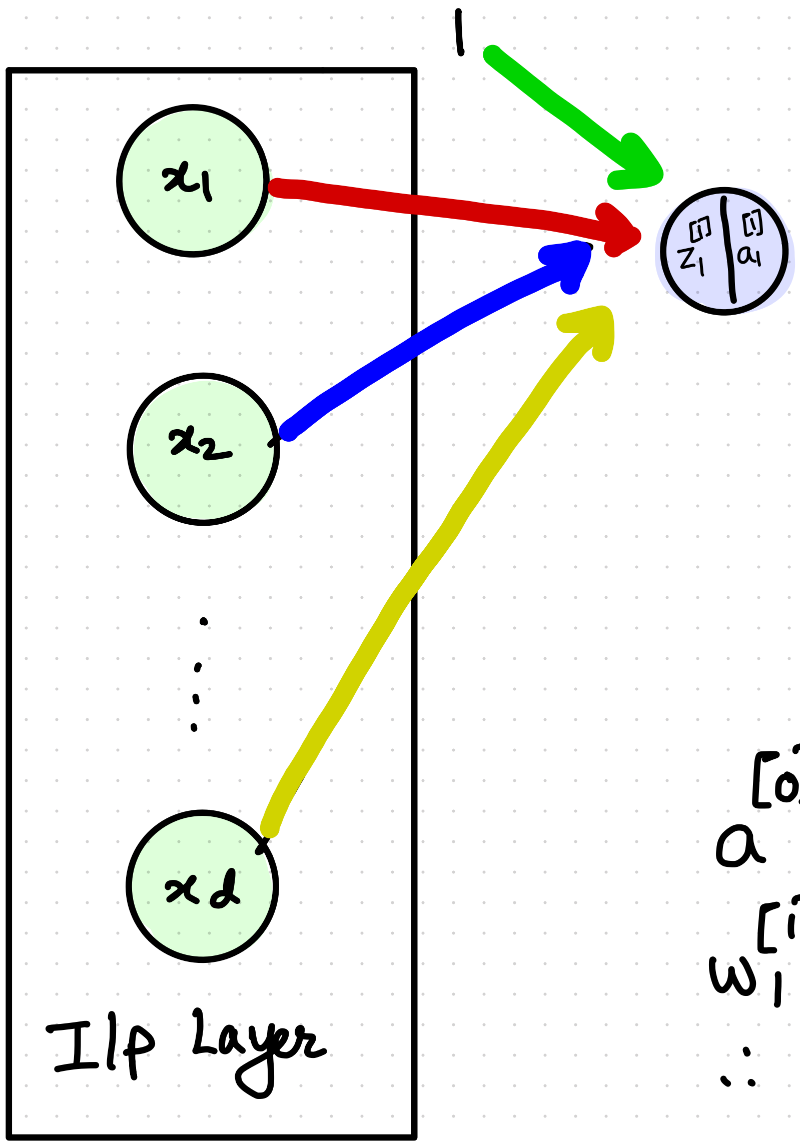
$w^{[l]}_{a,b}$

$[l]$ ← l^{th} layer

a → a^{th} node in l^{th} layer

b → b^{th} component of prev. layer activatn

CONSIDER SINGLE NEURON (LAYER 1, NODE 1)



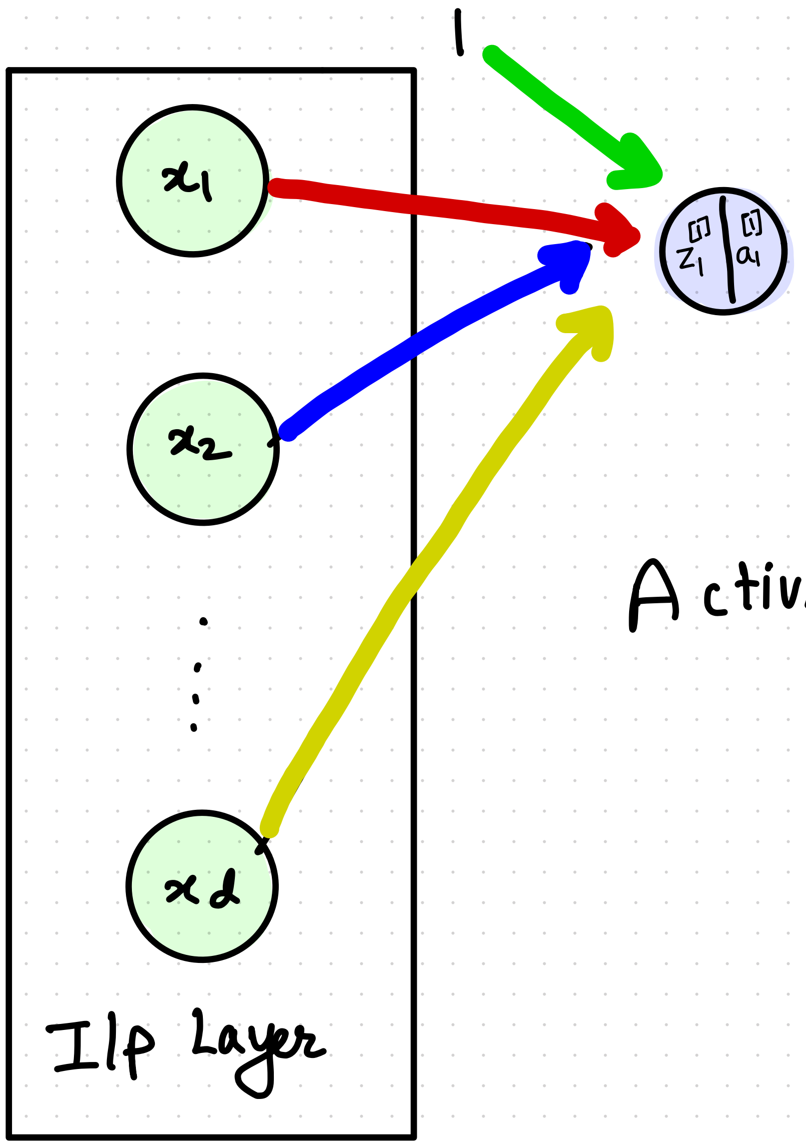
$$z_1^{[1]} = 1 * b_1^{[1]} + a_1^{[0]} * w_{1,1}^{[1]} + a_2^{[0]} * w_{1,2}^{[1]} + \dots + a_d^{[0]} * w_{1,d}^{[1]}$$

$$a^{[0]} \in R^D$$

$$w_1^{[1]} \in R^D$$

$$\therefore \; z_1^{[1]} = {w_1^{[1]}}^T a^{[0]} + b_1^{[1]}$$

CONSIDER SINGLE NEURON (LAYER 1, NODE 1)

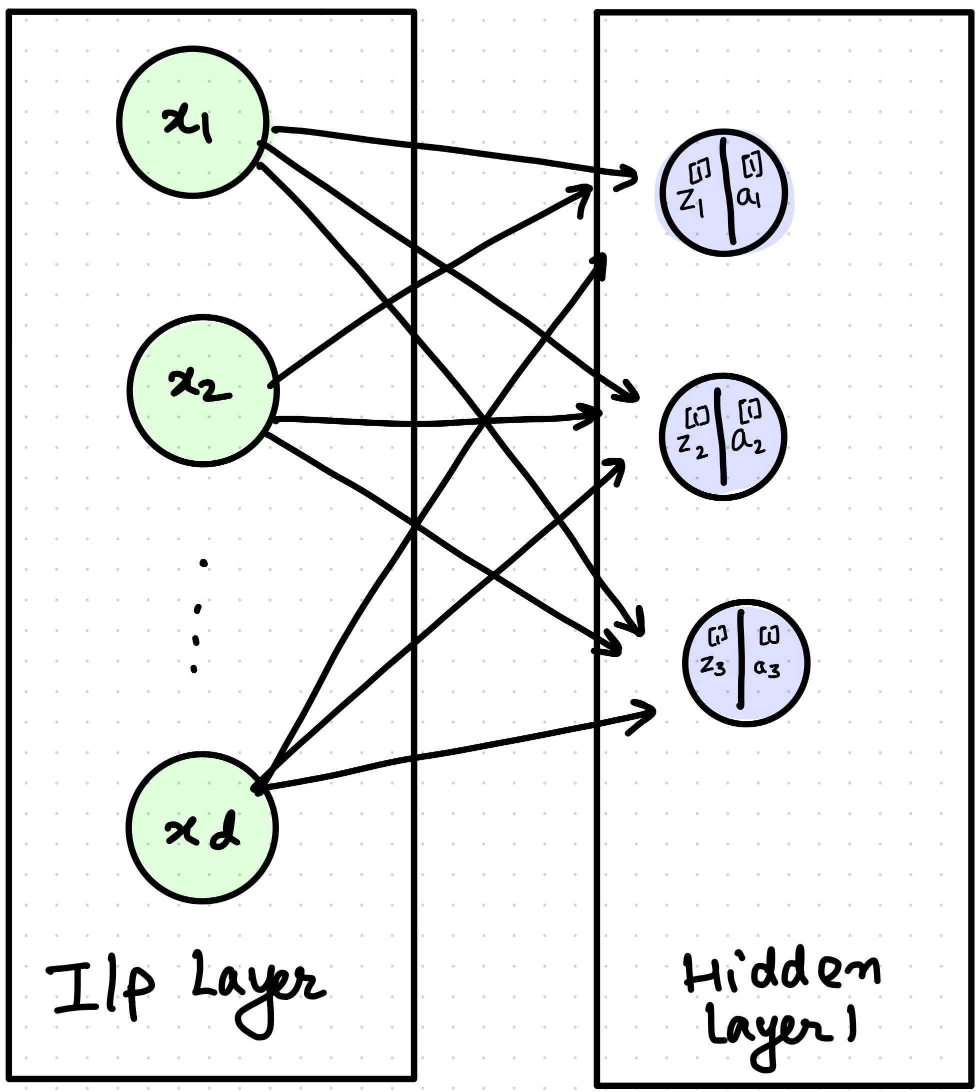


$$z_1^{[1]} = w_1^{[1]^T} a^{[0]} + b_1^{[1]}$$

Activatⁿ = $a_1^{[1]} = g(z_1^{[1]})$

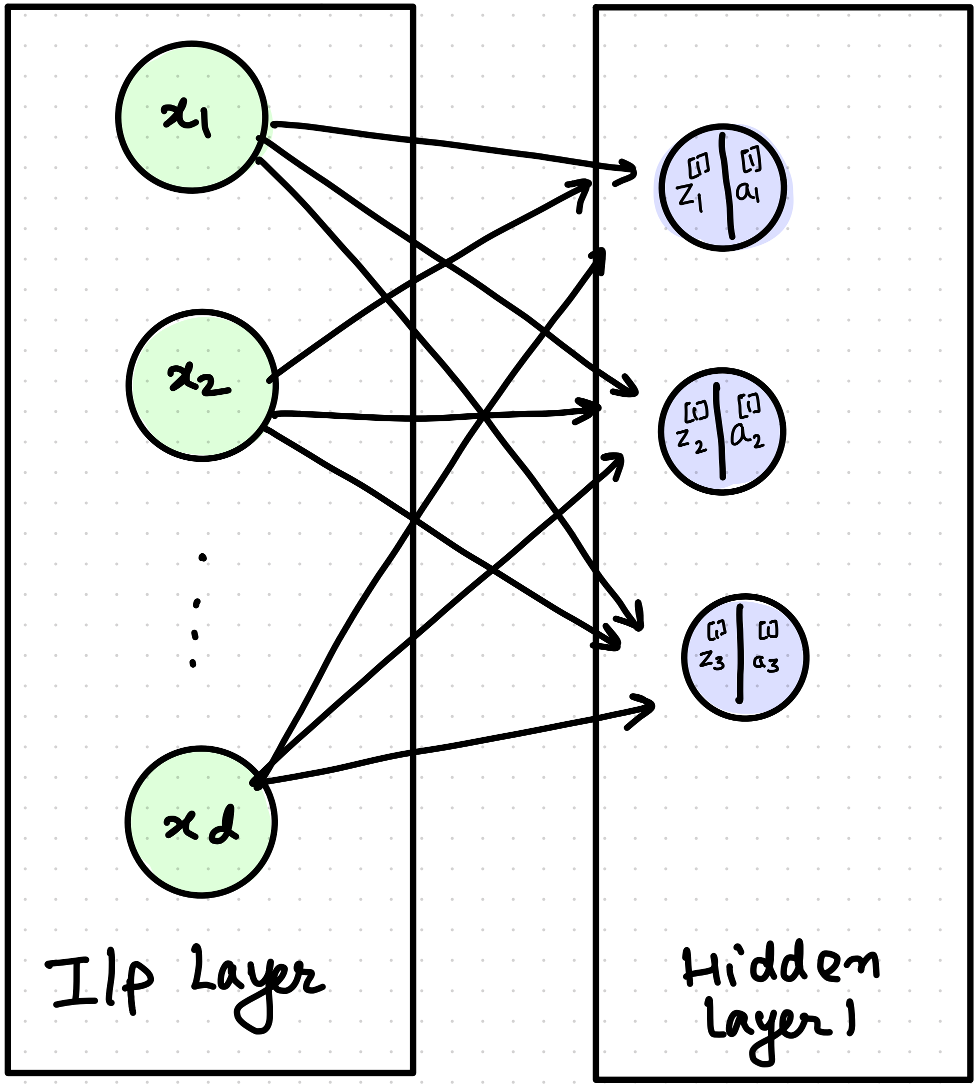
$$a_1^{[1]} \in R$$

FORWARD PROPAGATION



$$a_1^{[1]} = g\left({w_1^{[1]}}^T a^{[0]} + b_1^{[1]} \right)$$

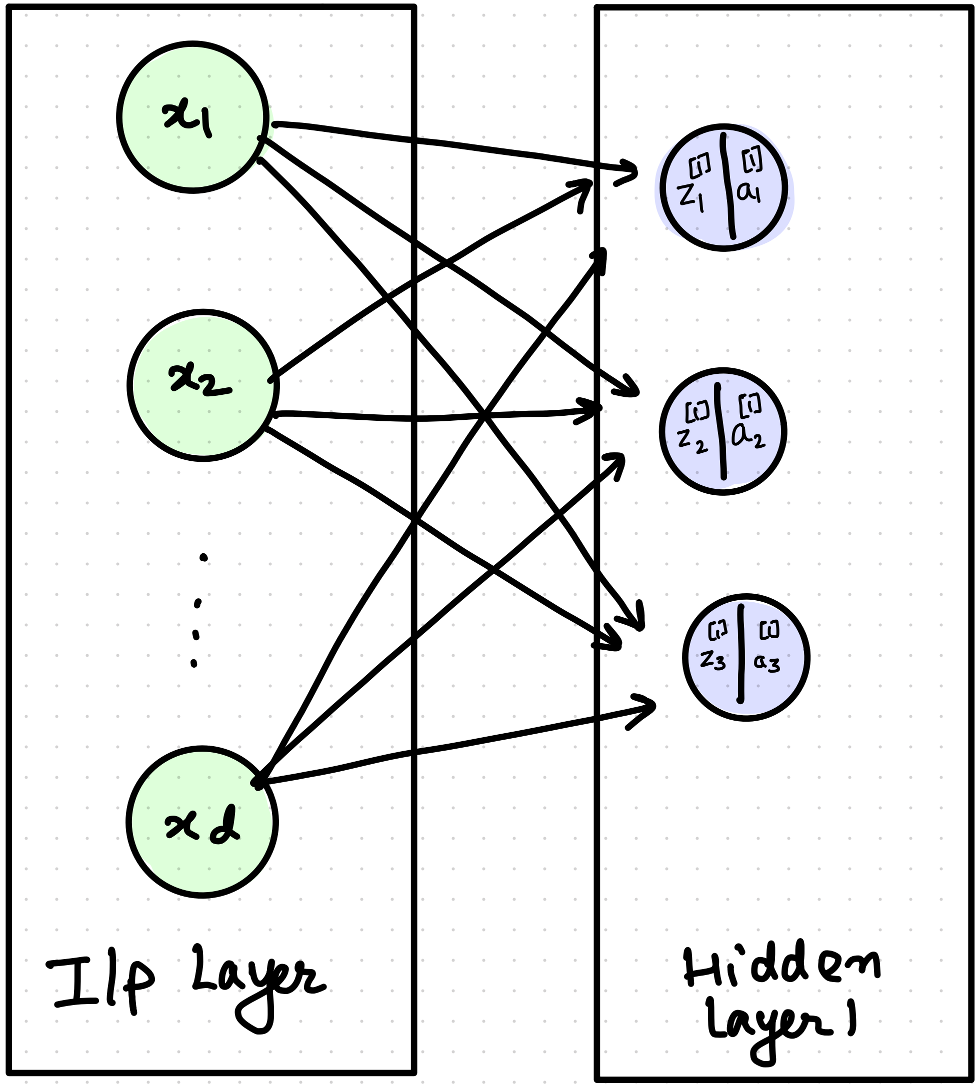
FORWARD PROPAGATION



$$a_1^{[1]} = g\left({w_1^{[1]}}^T a^{[0]} + b_1^{[1]}\right)$$

$$a_2^{[1]} = g\left({w_2^{[1]}}^T a^{[0]} + b_2^{[1]}\right)$$

FORWARD PROPAGATION

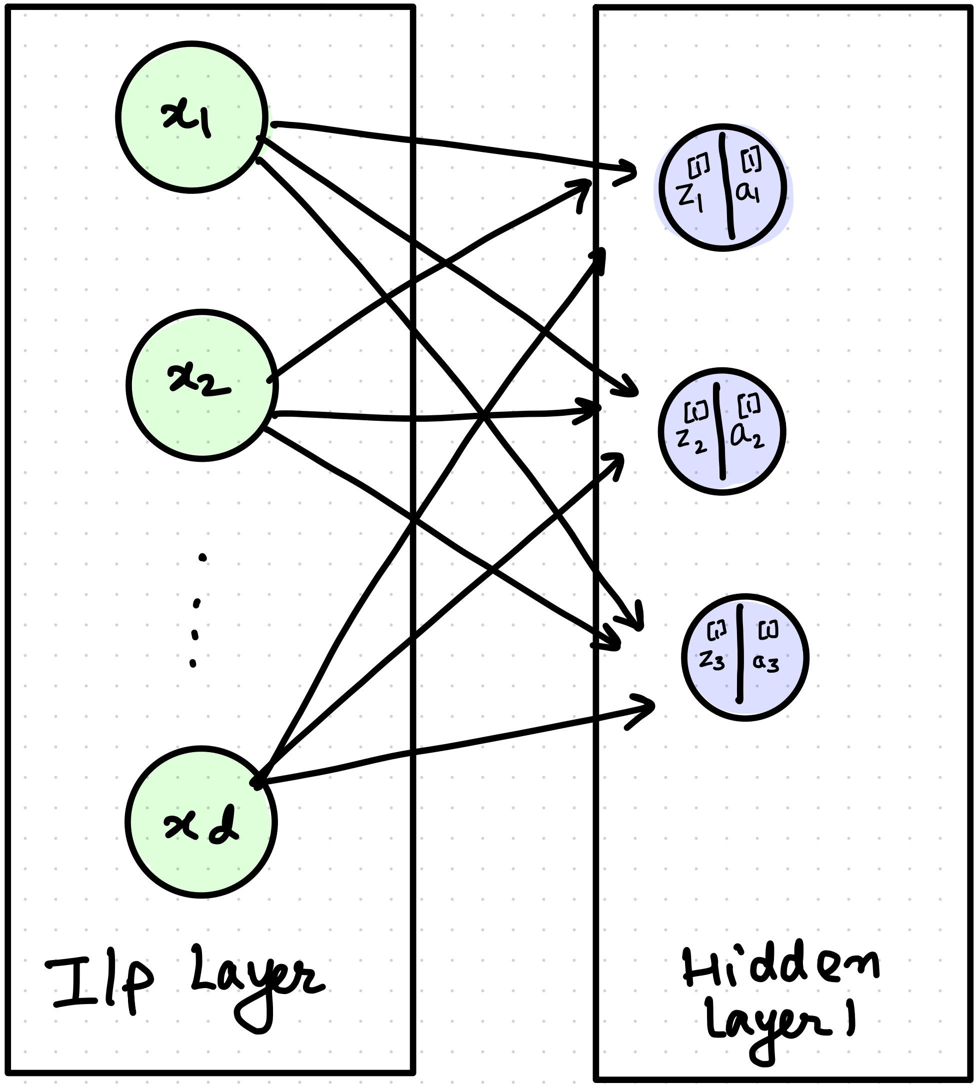


$$a_1^{[1]} = g\left(w_1^{[1]^T} a^{[0]} + b_1^{[1]}\right)$$

$$a_2^{[1]} = g\left(w_2^{[1]^T} a^{[0]} + b_2^{[1]}\right)$$

$$a_3^{[1]} = g\left(w_3^{[1]^T} a^{[0]} + b_3^{[1]}\right)$$

FORWARD PROPAGATION (VECTORISATION)

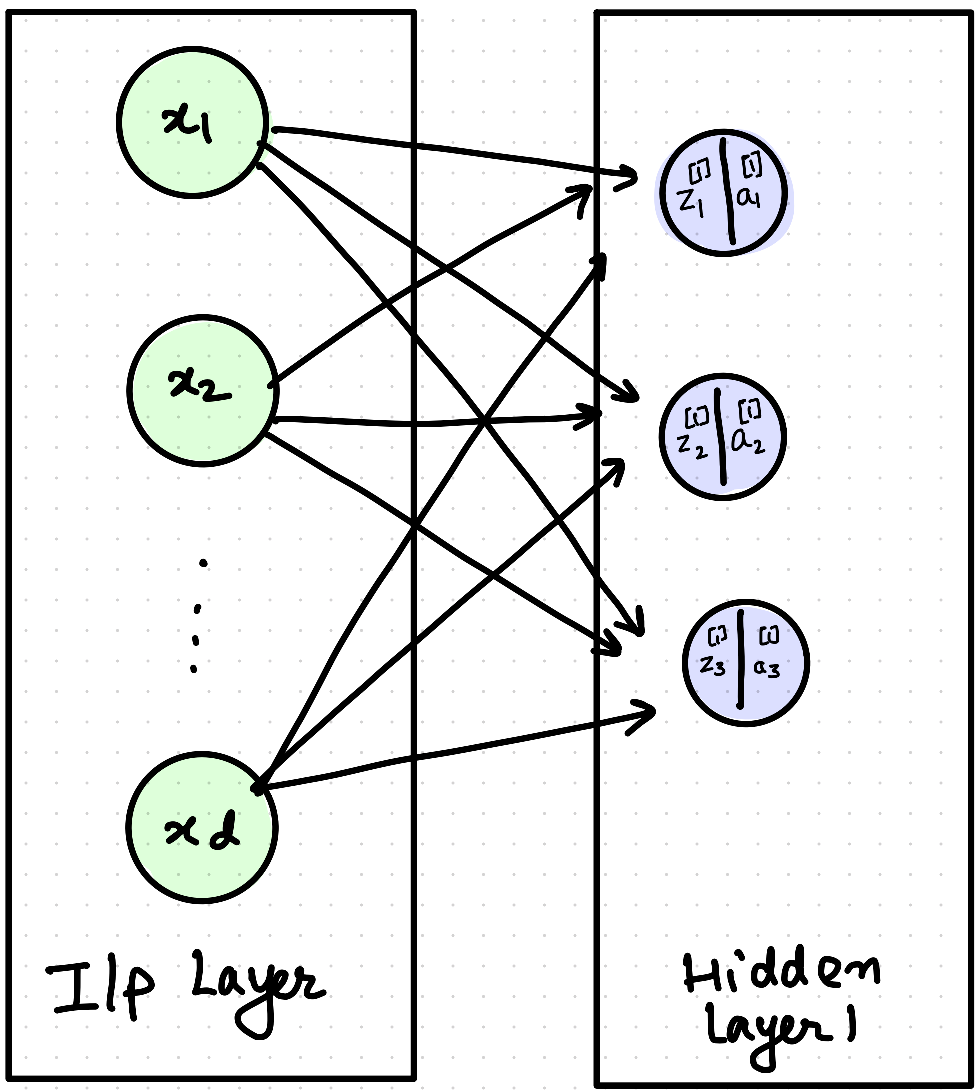


$$z_1^{[1]} = w_1^{[1]T} a^{[0]} + b_1^{[1]}$$

$$z_2^{[1]} = w_2^{[1]T} a^{[0]} + b_2^{[1]}$$

$$z_3^{[1]} = w_3^{[1]T} a^{[0]} + b_3^{[1]}$$

FORWARD PROPAGATION (VECTORISATION)

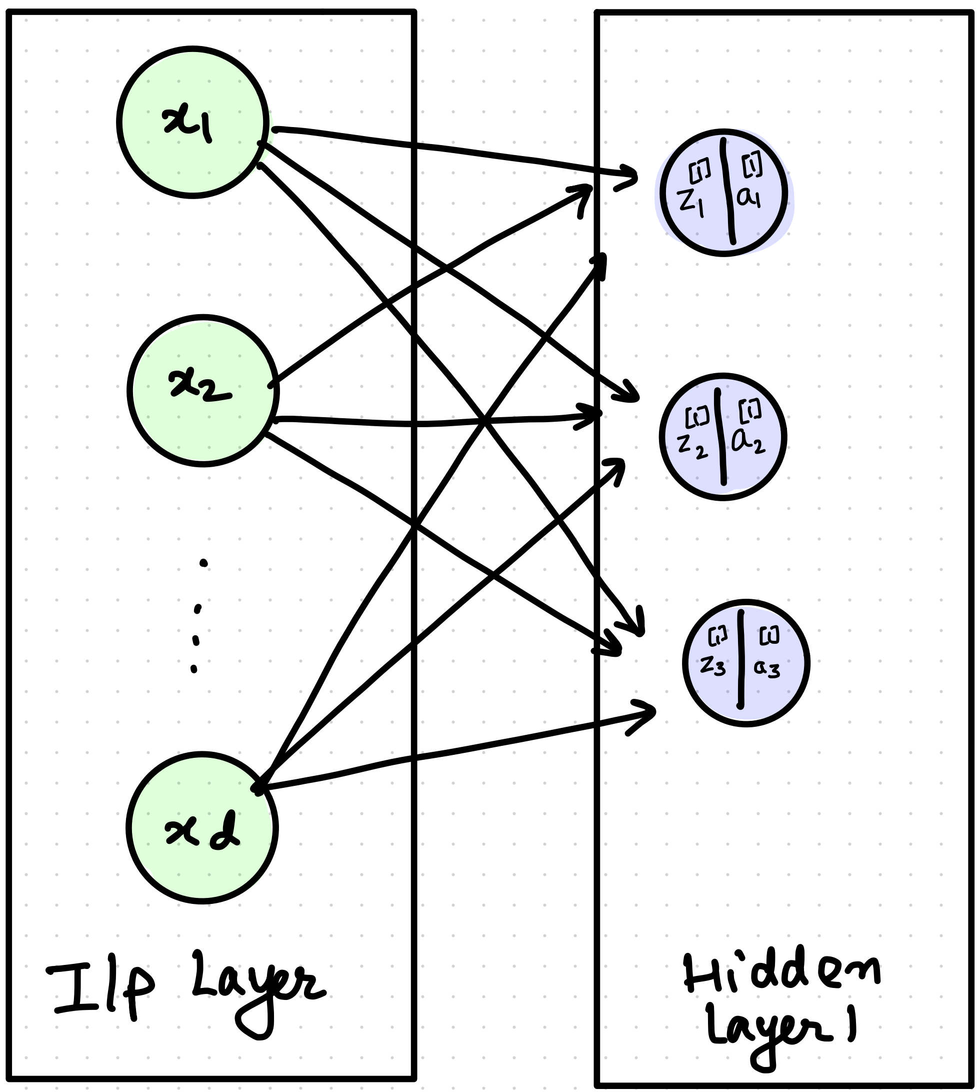


$$\underbrace{z_1^{[1]}}_{1\times1} = \underbrace{w_1^{[1]^T}}_{1\times3} \underbrace{a^{[0]}}_{3\times1} + \underbrace{b_1^{[1]}}_{1\times1}$$

$$\underbrace{z_2^{[1]}}_{1\times1} = \underbrace{w_2^{[1]^T}}_{1\times3} \underbrace{a^{[0]}}_{3\times1} + \underbrace{b_2^{[1]}}_{1\times1}$$

$$\underbrace{z_3^{[1]}}_{1\times1} = \underbrace{w_3^{[1]^T}}_{1\times3} \underbrace{a^{[0]}}_{3\times1} + \underbrace{b_3^{[1]}}_{1\times1}$$

FORWARD PROPAGATION (VECTORISATION)



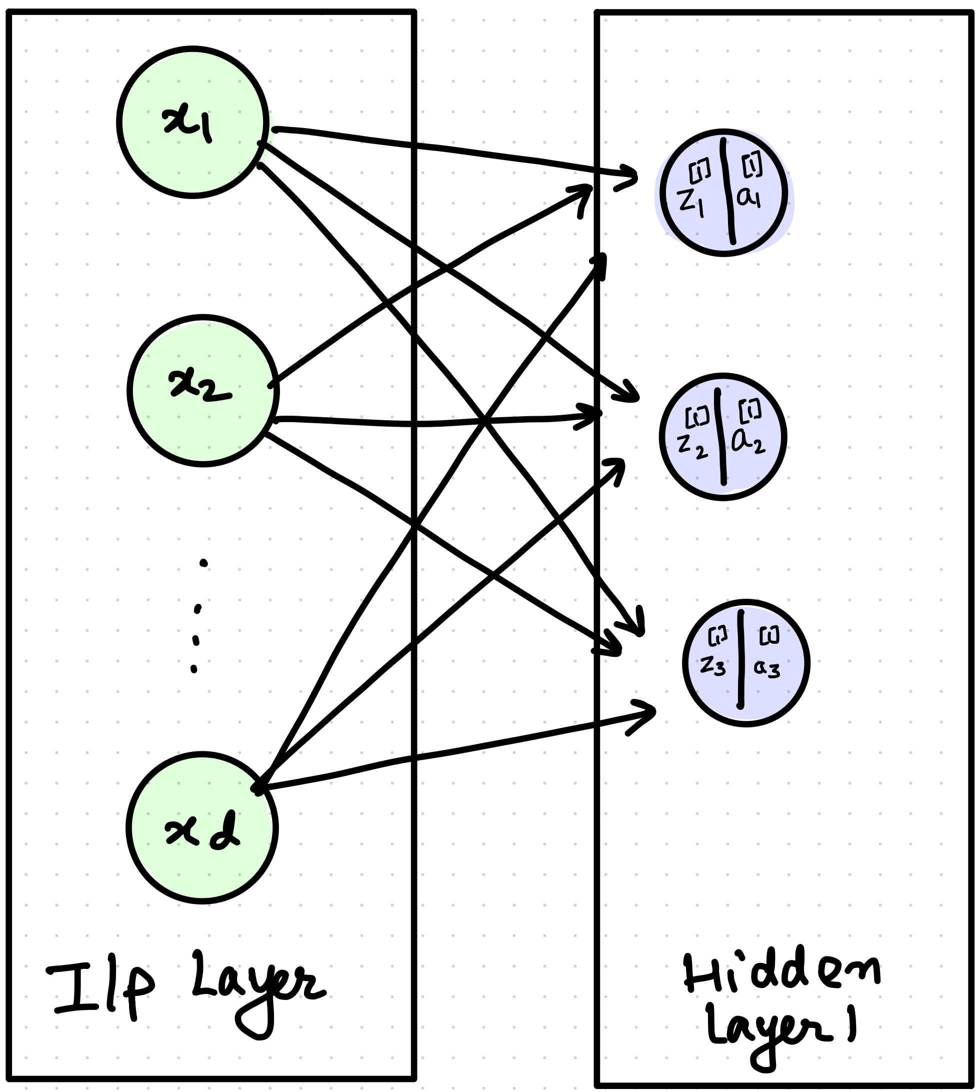
$$\underset{1\times1}{z_1^{[1]}} = \underset{1\times3}{w_1^{[1]T}} \underset{3\times1}{a^{[0]}} + \underset{1\times1}{b_1^{[1]}}$$

$$\underset{1\times1}{z_2^{[1]}} = \underset{1\times3}{w_2^{[1]T}} \underset{3\times1}{a^{[0]}} + \underset{1\times1}{b_2^{[1]}}$$

$$\underset{1\times1}{z_3^{[1]}} = \underset{1\times3}{w_3^{[1]T}} \underset{3\times1}{a^{[0]}} + \underset{1\times1}{b_3^{[1]}}$$

$$\underset{3\times1}{z^{[1]}} = \underset{3\times3}{\begin{bmatrix} — & w_1^{[1]T} & — \\ — & w_2^{[1]T} & \\ — & w_3^{[1]T} & \end{bmatrix}} \underset{3\times1}{a^{[0]}} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \end{bmatrix}$$

FORWARD PROPAGATION (VECTORISATION)

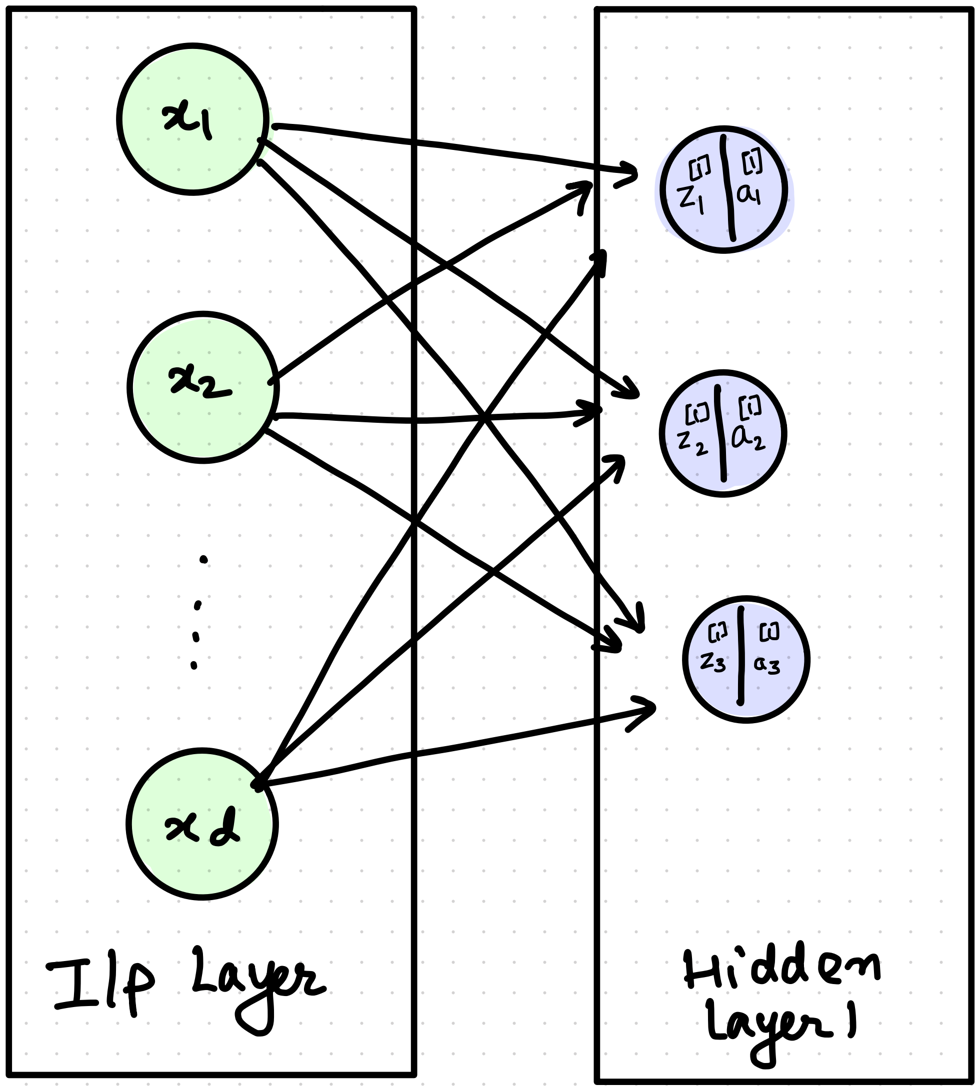


$$\underset{3\times1}{z^{[1]}} = \underset{3\times3}{\begin{bmatrix} - & w_1^{[1]T} & - \\ - & w_2^{[1]T} & \\ - & w_3^{[1]T} & \end{bmatrix}} \underset{3\times1}{a^{[0]}} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \end{bmatrix}$$

$$z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$$

capitals for matrices

FORWARD PROPAGATION (VECTORISATION)



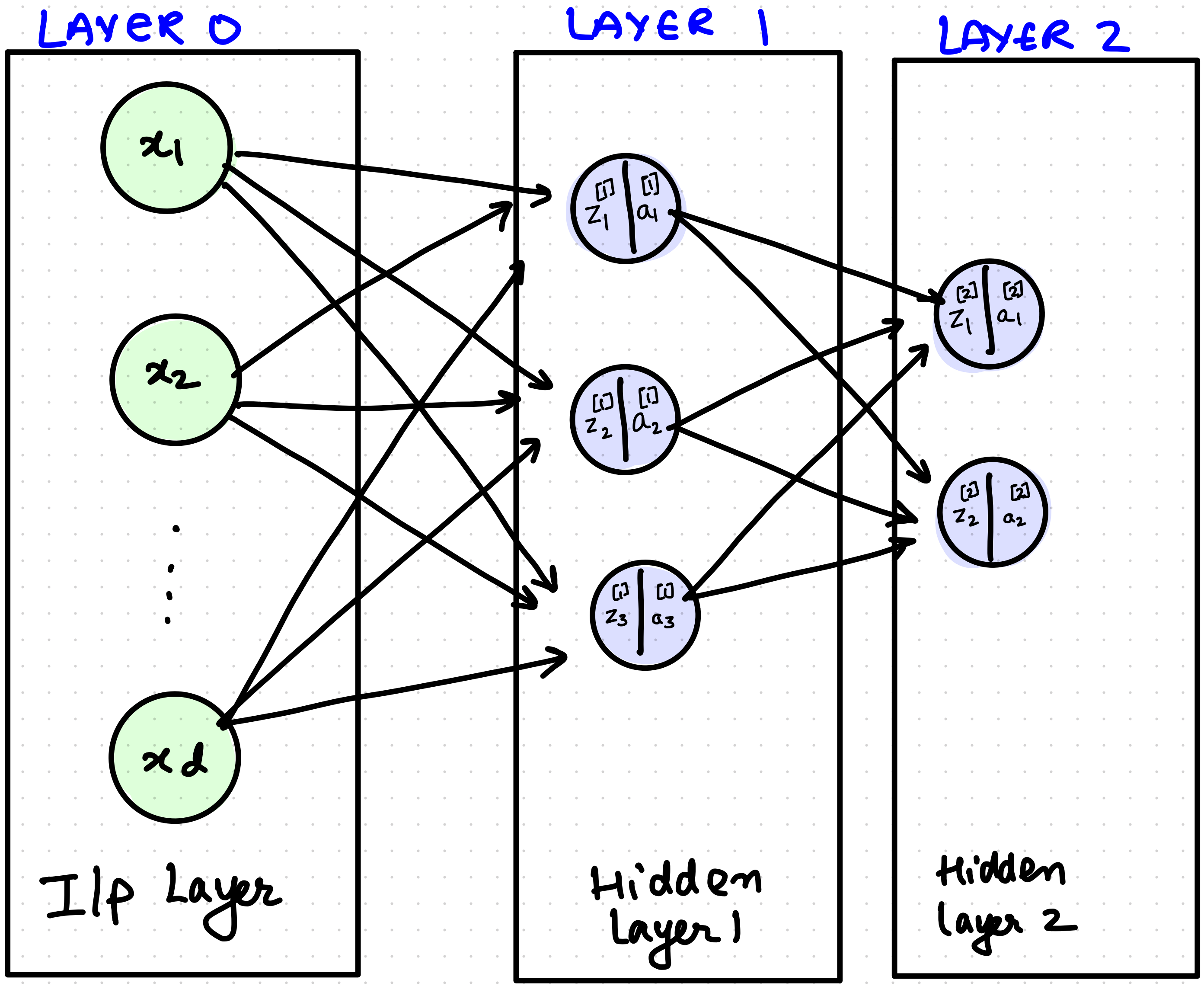
$$\underset{3\times1}{z^{[1]}} = \underset{3\times3}{\begin{bmatrix} — & w_1^{[1]T} & — \\ — & w_2^{[1]T} & — \\ — & w_3^{[1]T} & — \end{bmatrix}} \underset{3\times1}{a^{[0]}} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \end{bmatrix}$$

$$z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$$

capitals for metrices

$$a^{[1]} = g\left(z^{[1]}\right)$$

FORWARD PROPAGATION

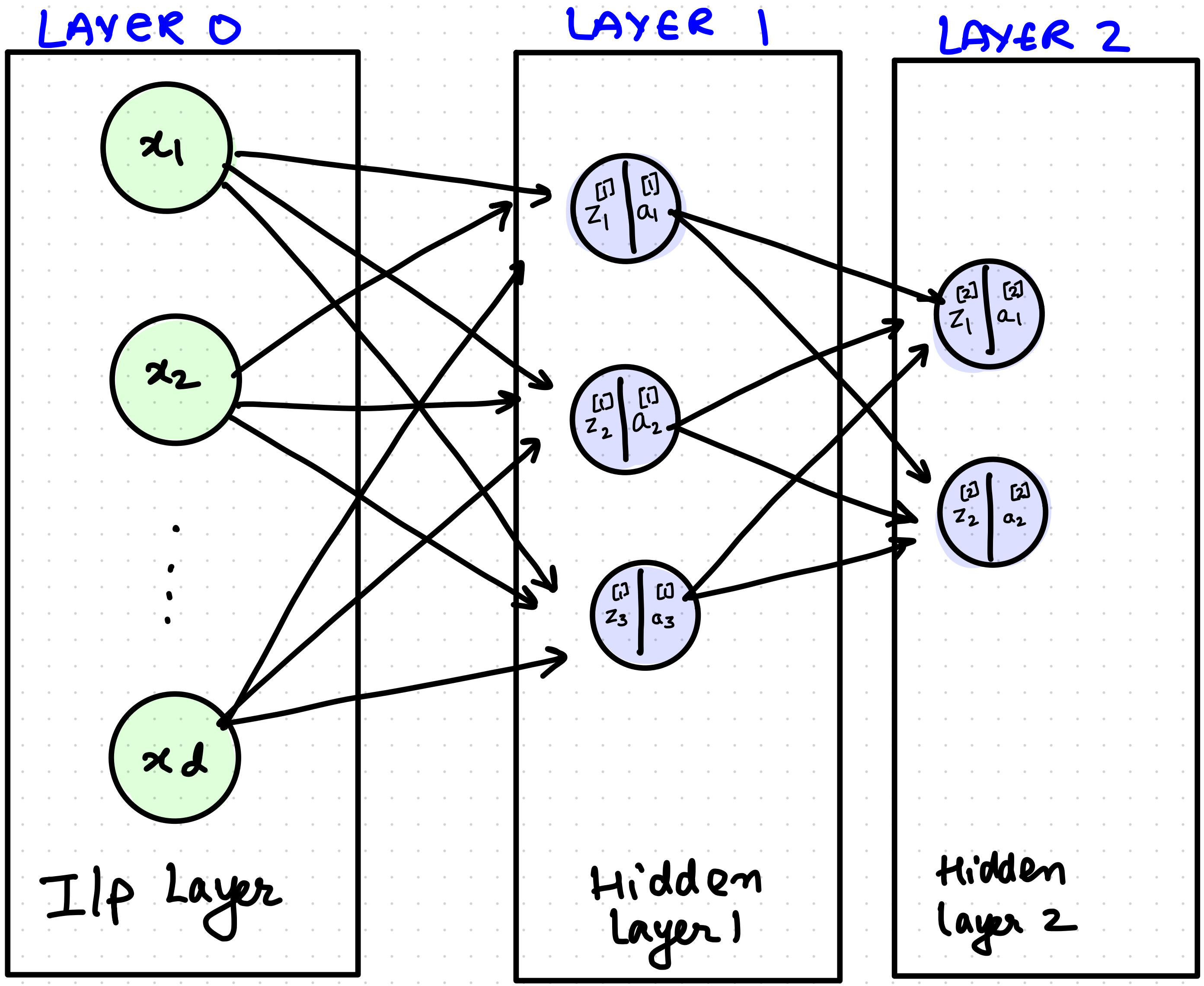


$$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$$

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

Q. Dim. of $W^{[2]}$?

FORWARD PROPAGATION



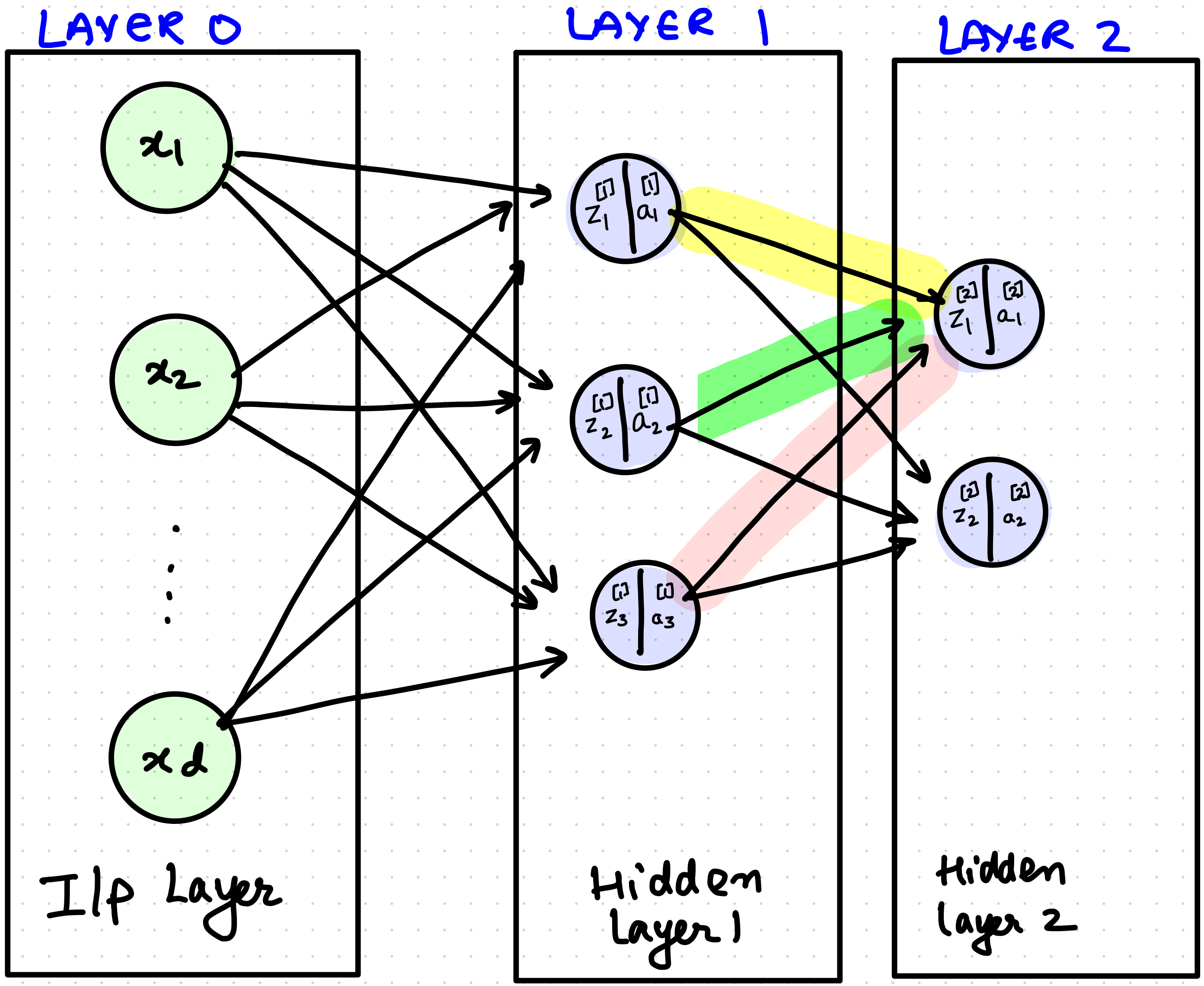
$$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$$

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

Q. Dim. of $W^{[2]}$?

$$W^{[2]} = \begin{bmatrix} - & w_1^{[2]} & - \\ - & w_2^{[2]} & - \end{bmatrix}$$

FORWARD PROPAGATION



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

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

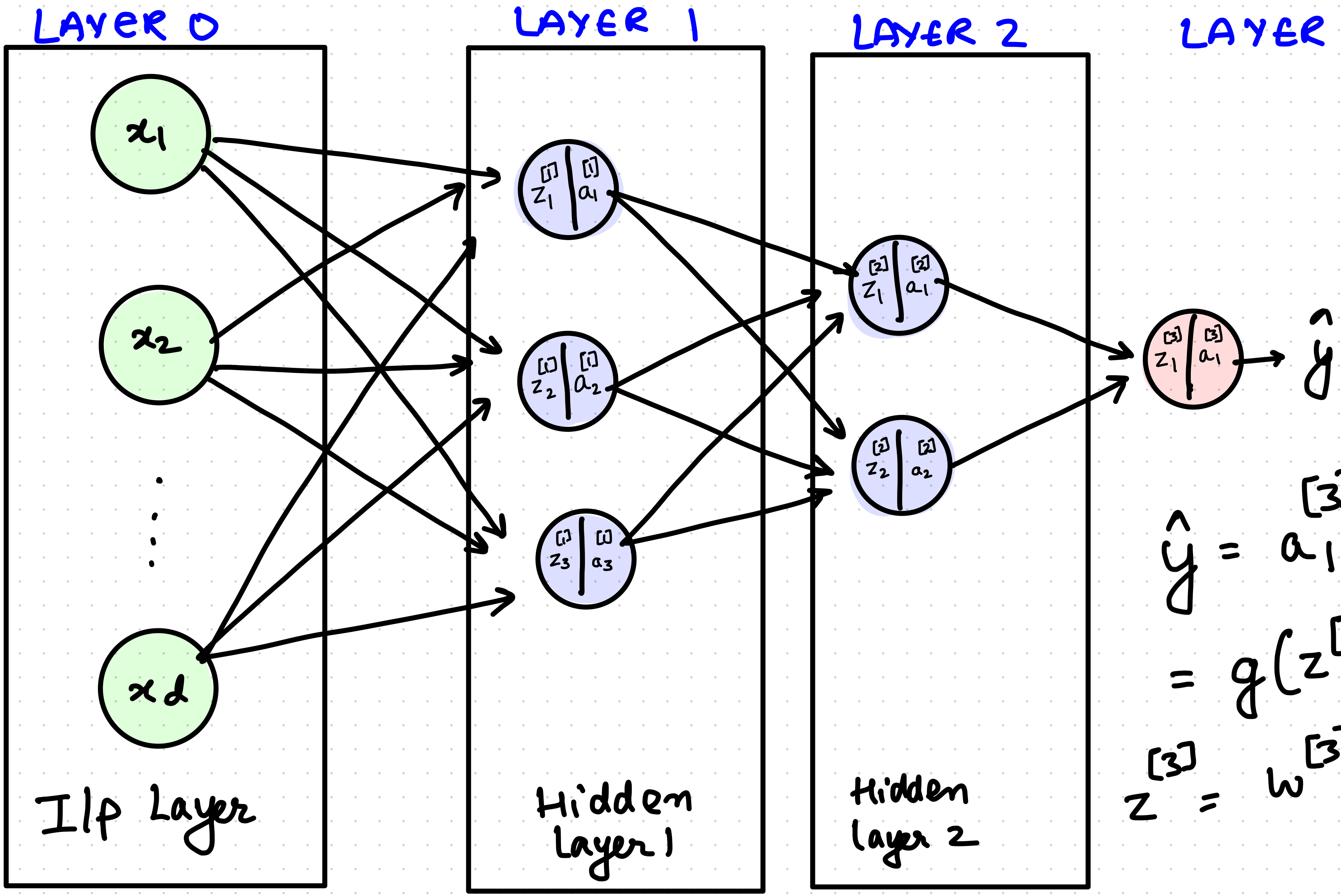
Q. Dim. of $w^{[2]}$?

$$w^{[2]} = \begin{bmatrix} - & {w_1^{[2]}}^T & - \\ - & {w_2^{[2]}}^T & - \end{bmatrix}$$

$$w_1^{[2]} \in R^3$$

$$\therefore\ w^{[2]} \in R^{2\times 3}$$

FORWARD PROPAGATION

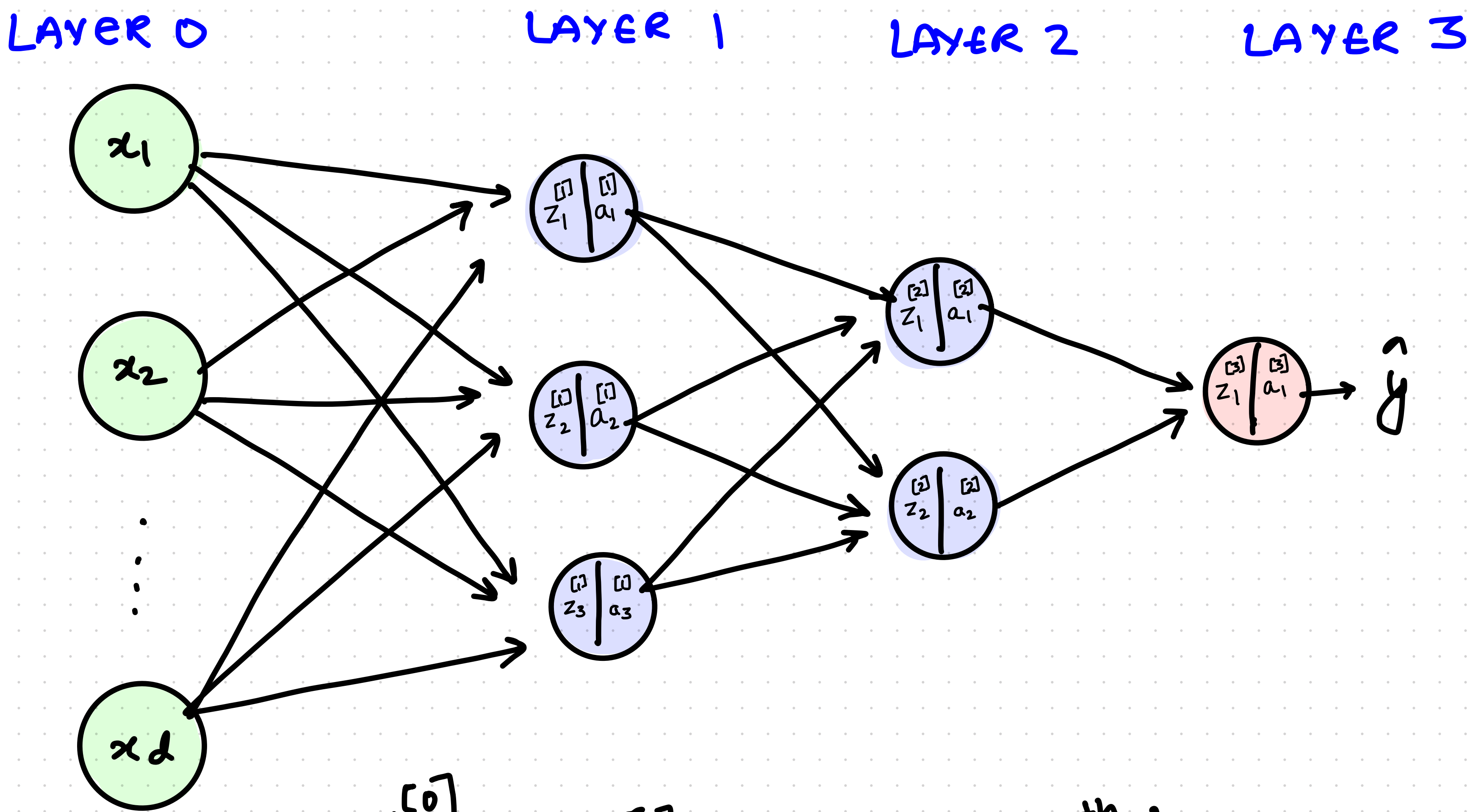


$$\hat{y} = a_1^{[3]}$$

$$= g\left(z^{[3]}\right)$$

$$z^{[3]} = w^{[3]} a^{[2]} + b^{[3]}$$

WHAT CAN WE SAY ABOUT SHAPES OF a, b, W

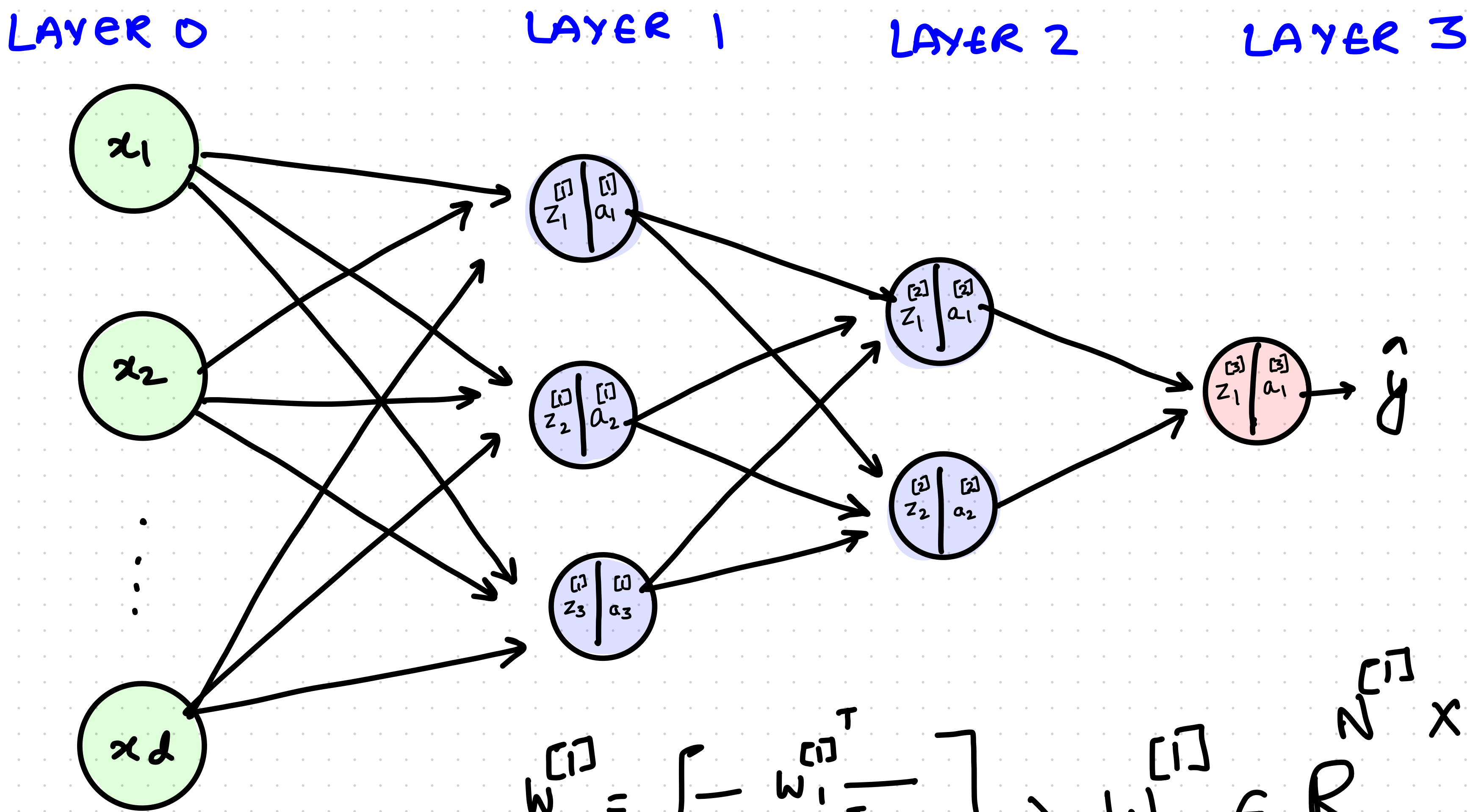


$a^{[0]} \in R^{d}$ or $R^{N^{[0]}}$

$d \rightarrow$ # i/p features

$N^{[0]}$ = # units in 0^{th} layer

WHAT CAN WE SAY ABOUT SHAPES OF a, b, W

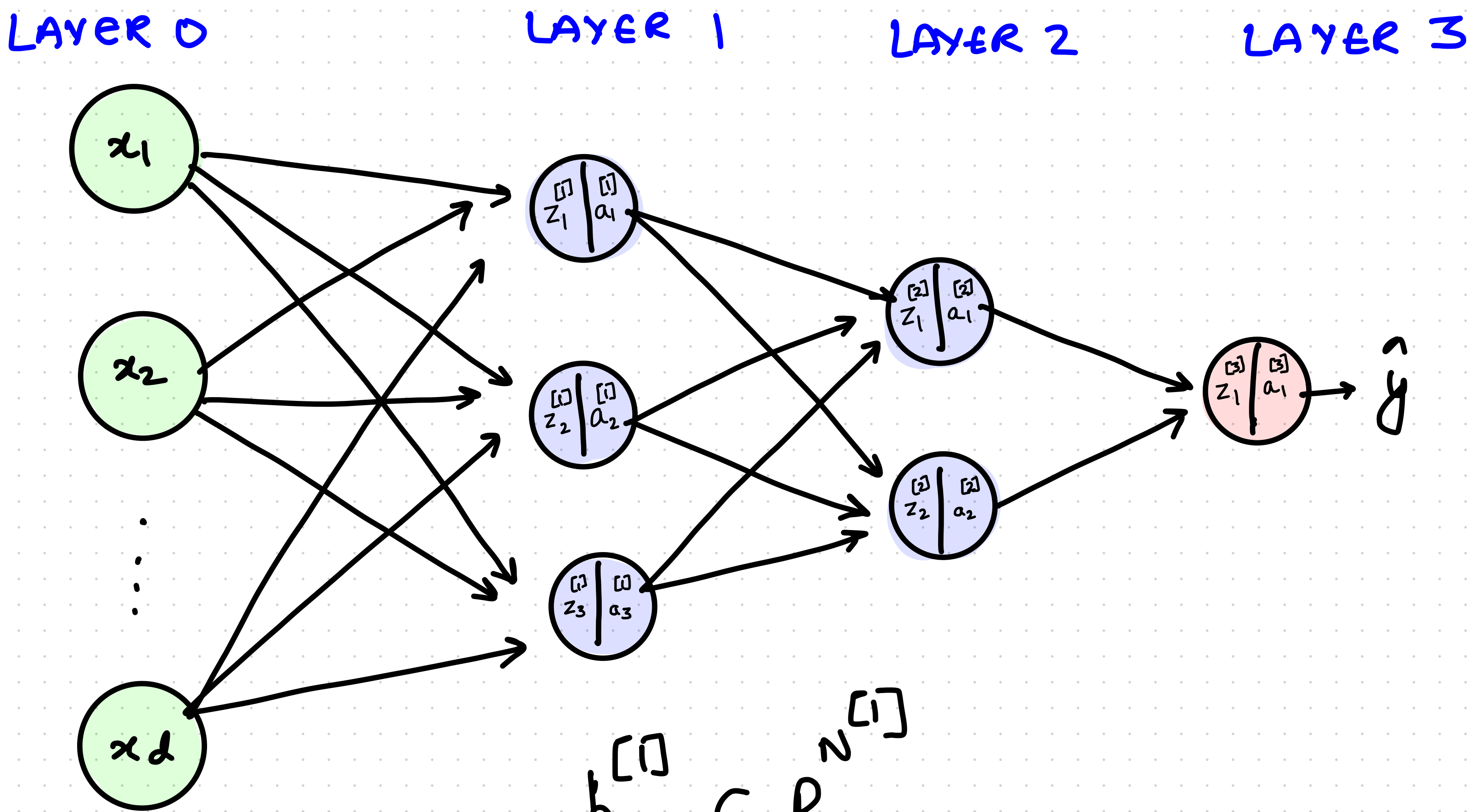


$a^{[0]} \in \mathbb{R}^d$

$d \rightarrow$ # i/p features

$$W^{[1]} = \begin{bmatrix} - & w_1^{[1]T} & - \\ - & w_2^{[1]T} & - \\ & \vdots & \\ - & w_{n^{[1]}}^{[1]T} & - \end{bmatrix} \Rightarrow W^{[1]} \in \mathbb{R}^{n^{[1]} \times n^{[0]}}$$

WHAT CAN WE SAY ABOUT SHAPES OF a, b, W



$b^{[1]} \in \mathbb{R}^{N^{[1]}}$

$a^{[0]} \in \mathbb{R}^{d}$

$d \rightarrow$ # i/p features

SUMMARY OF SHAPES

$N^{[l]}$: # NODES IN l^{th} Layer

$a^{[0]} \in R^{N^{[0]}}$

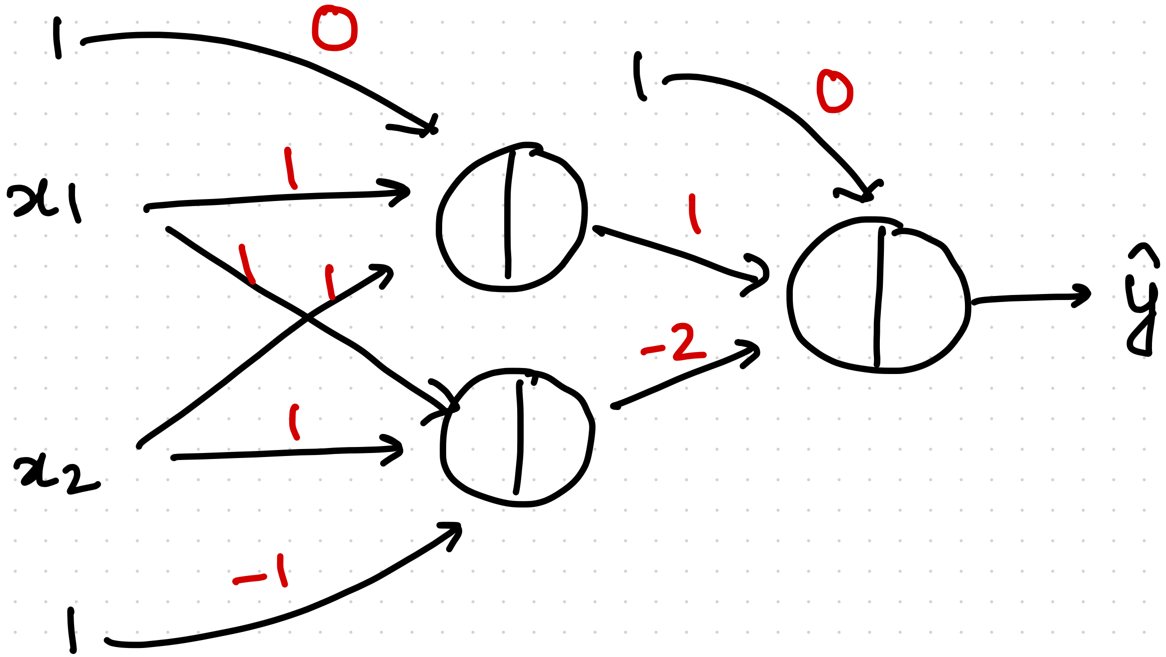
$W^{[l]} \in R^{N^{[l]} \times N^{[l-1]}}$

$b^{[l]} \in R^{N^{[l]}}$

$Z^{[l]} \in R^{N^{[l]}}$

$a^{[l]} \in R^{N^{[l]}}$

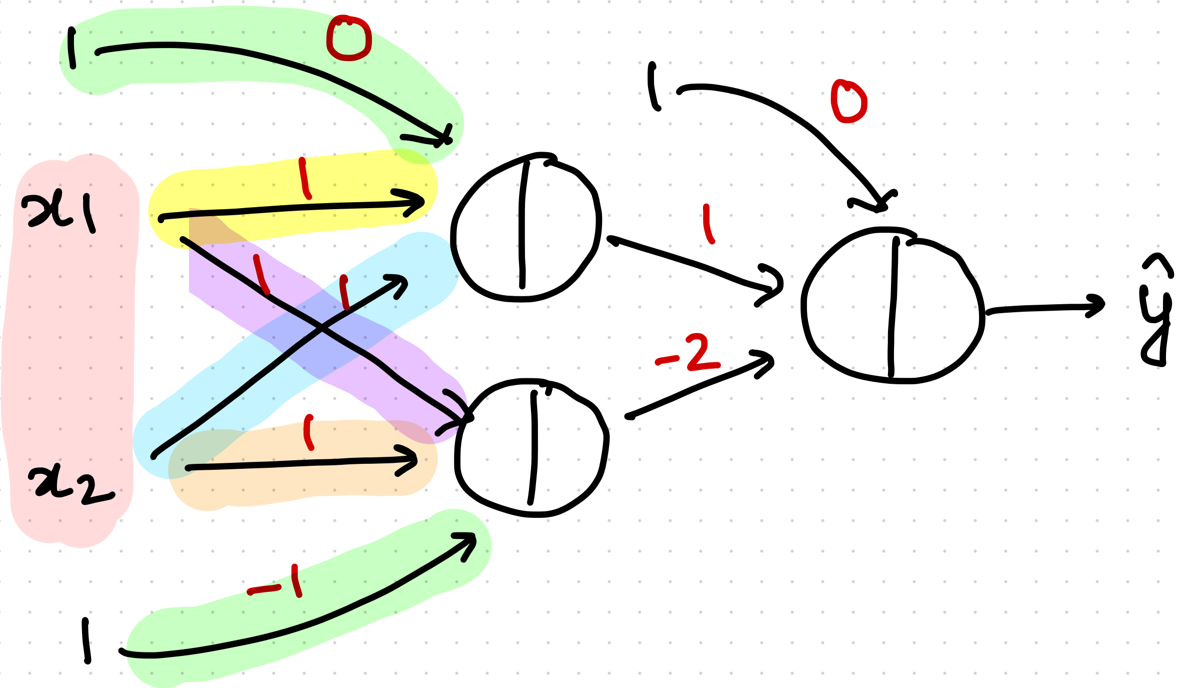
XOR USING "MLP" RELU



CONFIRM IF ABOVE N/W IS CORRECT FOR XOR.

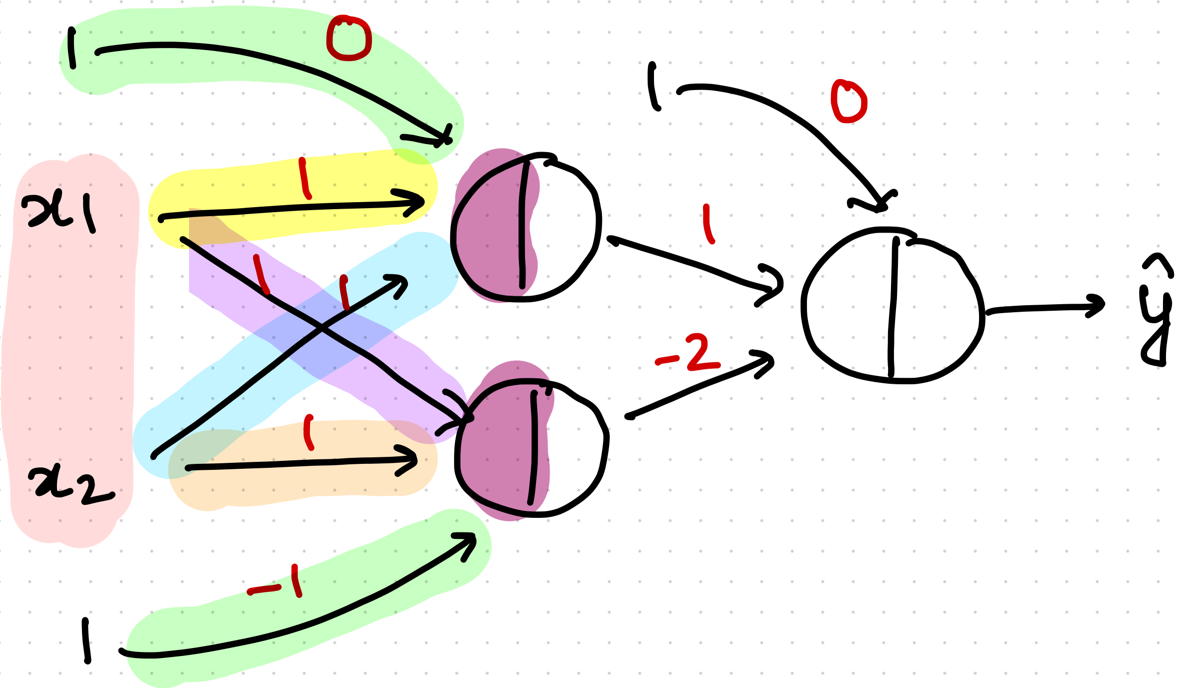
Start with $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $y_{TRUE} = 0$

XOR USING "MLP" RELU



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

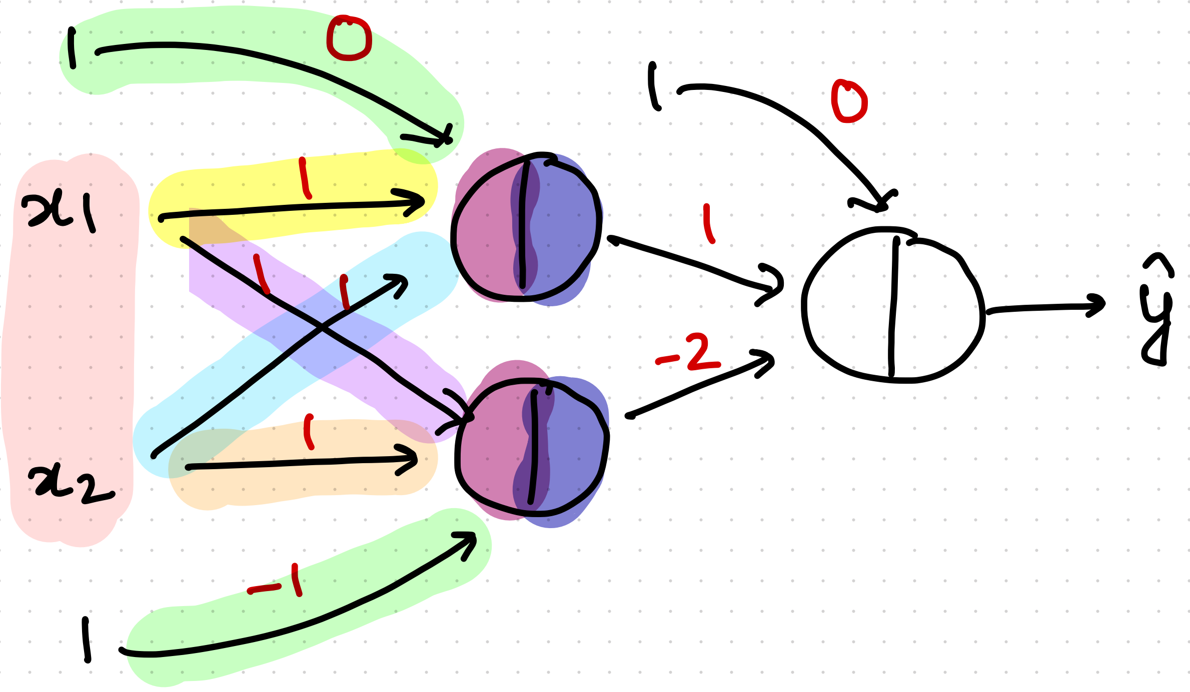
XOR USING "MLP" RELU



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

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

XOR USING "MLP" RELU

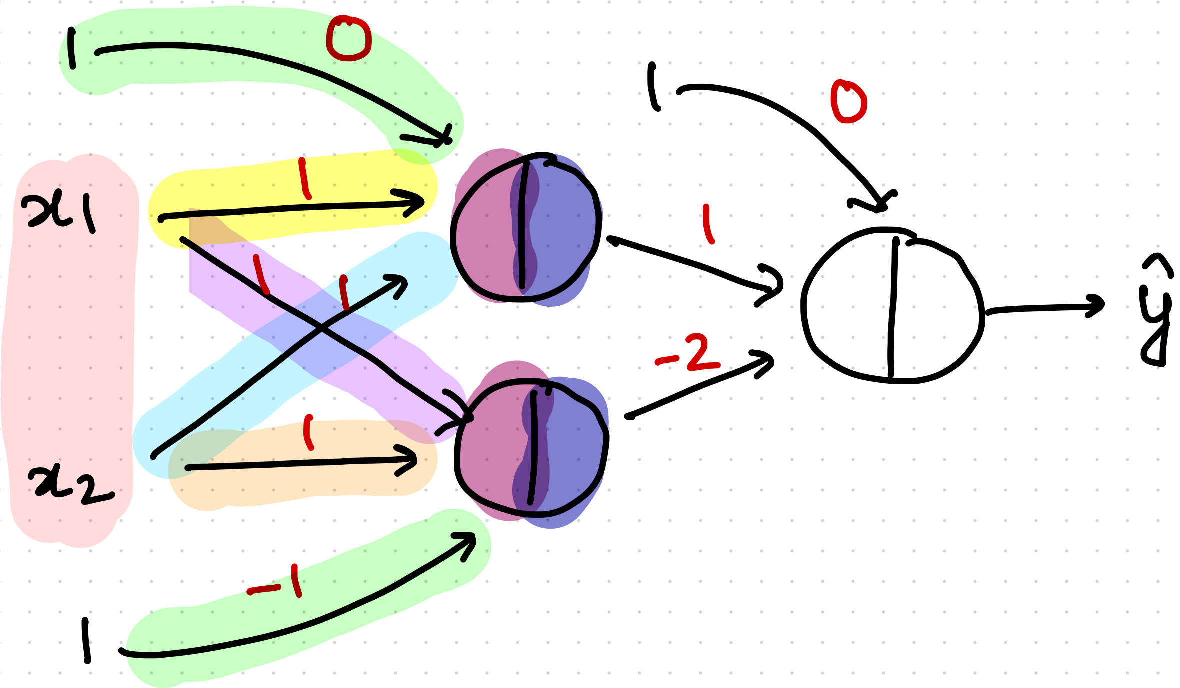


$$a^{[0]} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}; \quad b^{[1]} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}; \quad W^{[1]} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

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

$$a^{[1]} = \mathrm{RELU}\begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

XOR USING "MLP" RELU

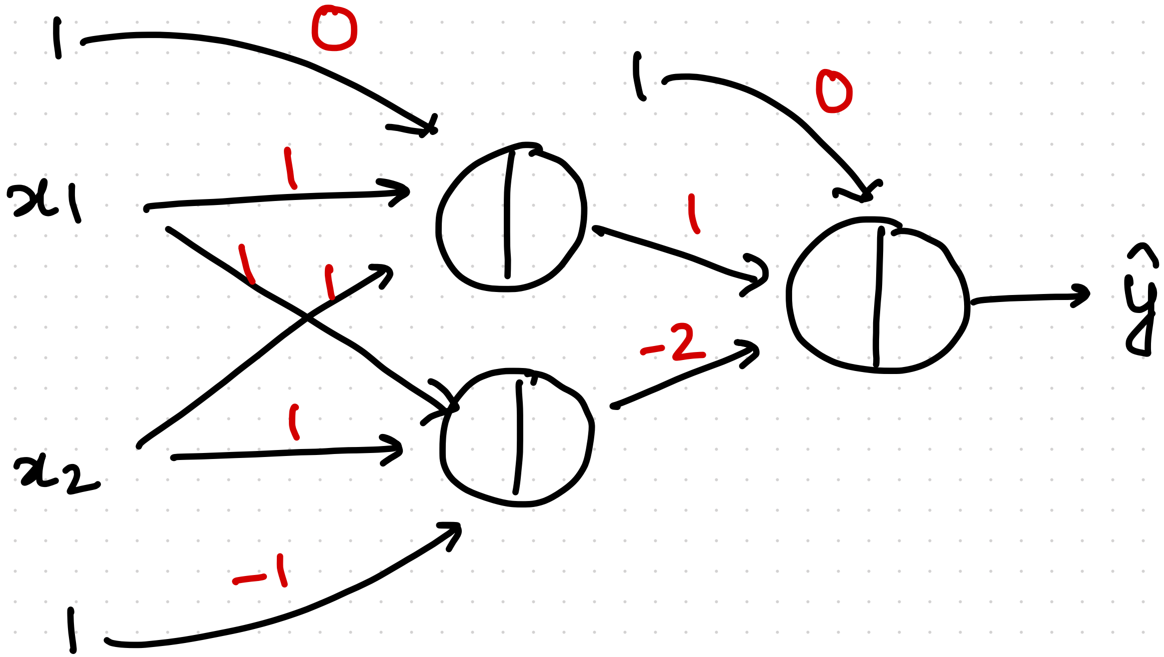


$$a^{[1]} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad ; \quad W^{[2]} = \begin{bmatrix} 1 & -2 \end{bmatrix} \quad ; \quad b^{[2]} = \begin{bmatrix} 0 \end{bmatrix}$$

$$z^{[2]} = \begin{bmatrix} 1 & -2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} = 0 \quad ; \quad a^{[2]} = \hat{y} = \mathrm{RELU}(0) = 0$$

✓

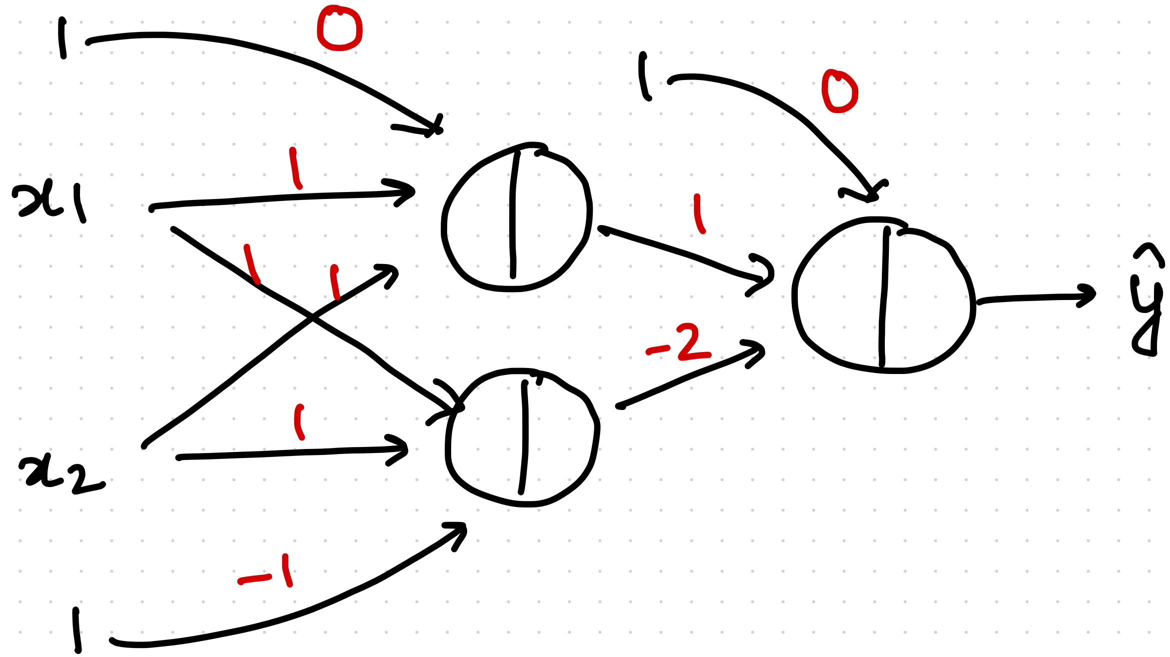
XOR USING "MLP" RELU



CONFIRM IF ABOVE N/W IS CORRECT FOR XOR.

Start with $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $y_{TRUE} = 1$

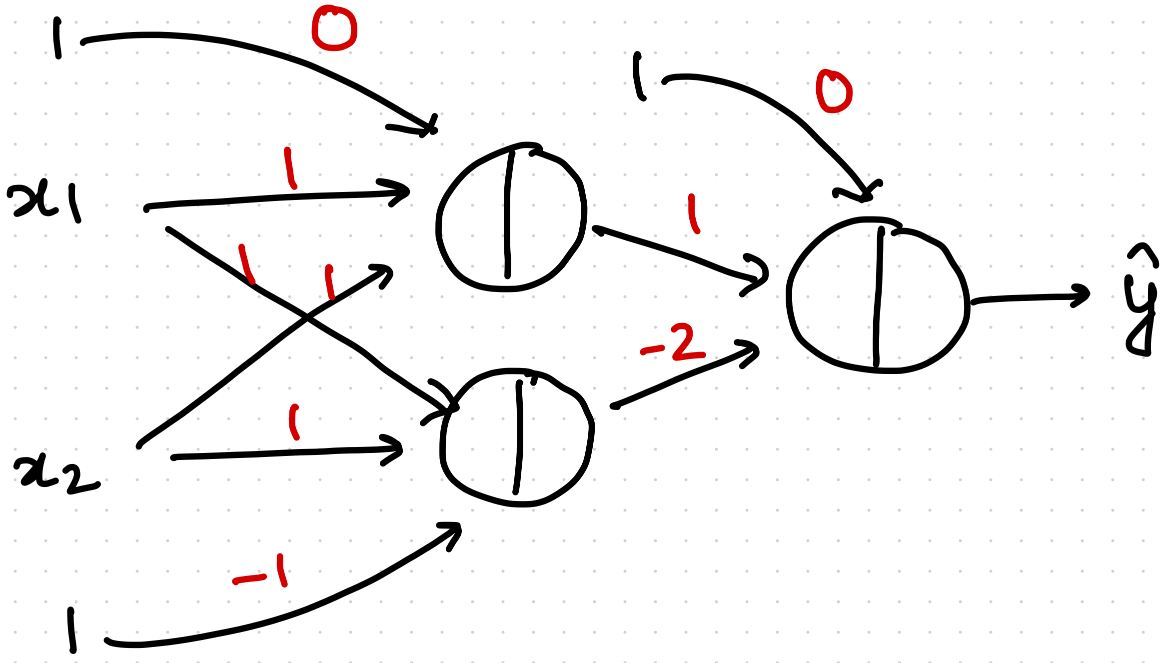
XOR USING "MLP" RELU



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

$$a^{[1]} = \text{RELU}\left(z^{[1]}\right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

XOR USING "MLP" RELU



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

$$a^{[1]} = RELU\left(z^{[1]}\right) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

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

$$a^{[2]} = RELU(1) = 1$$

$$= \hat{y}$$