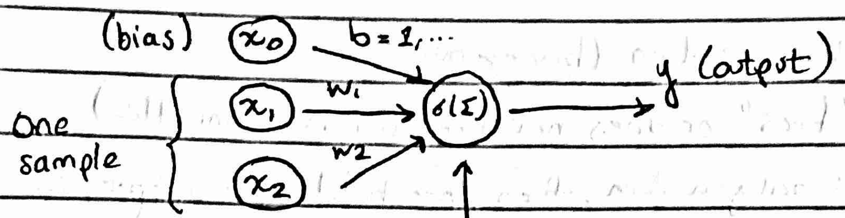
Discussion: Perceptrons

Neural networks are inspired by how neurons in the brain function & are connected. However, they work with math ☺

Single neurons (node, unit)

Neurons are used to model nonlinear data, usually. (can't be approximated by $y = mx + b$)



The node computes

some bias

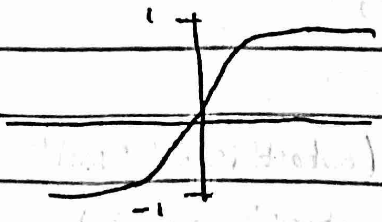
$$f(w_1 x_1 + w_2 x_2 + x_0)$$

- where f is the activation function (in a logistic unit, this would be sigmoid; in a perceptron, a threshold)

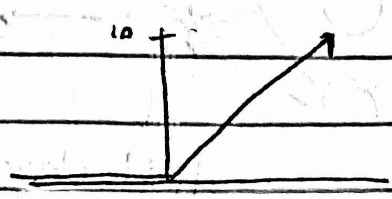
★ The weights (w_1, w_2) start randomly and are updated through backpropagation in a neural network

Other examples of activation functions:

tanh



ReLU



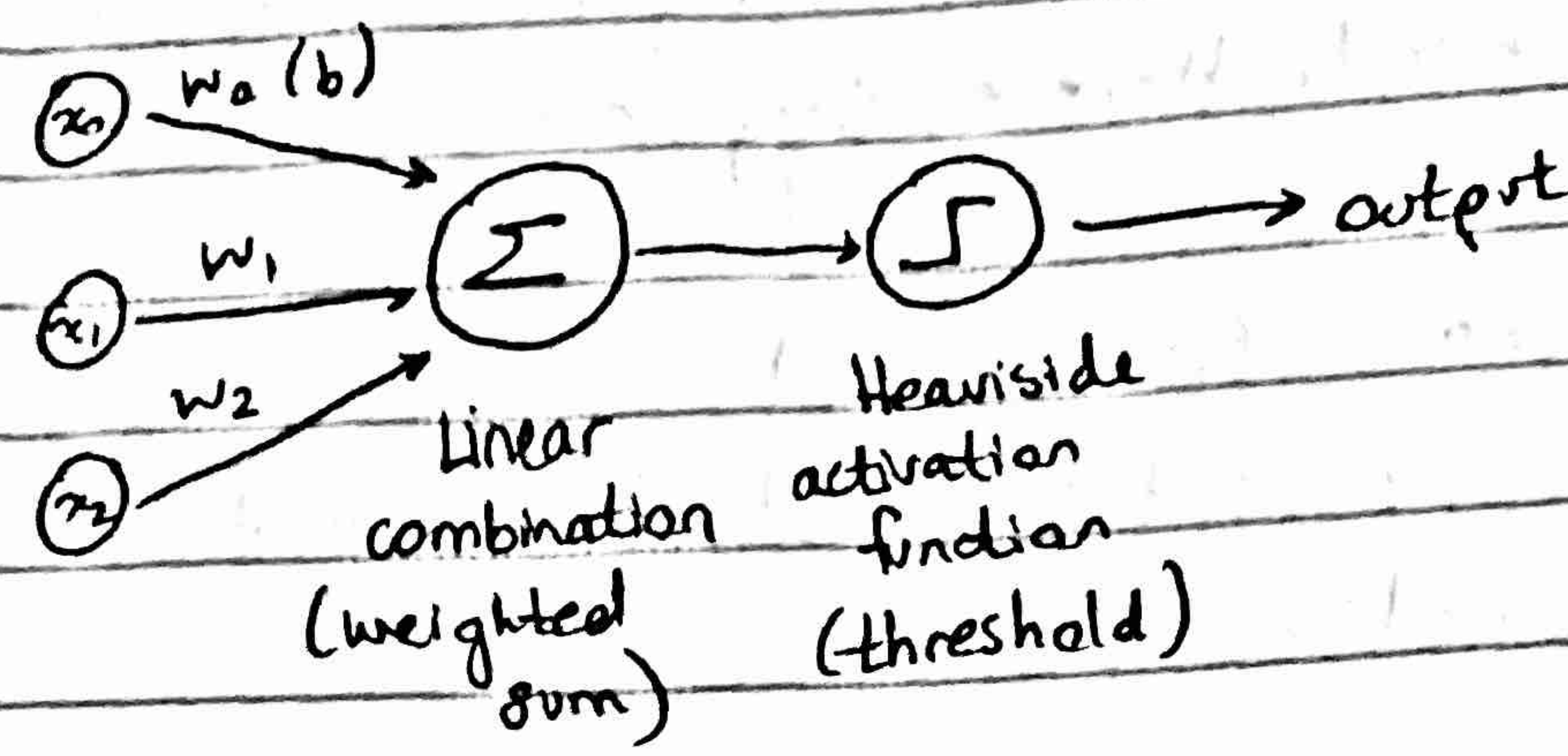
$\tanh(x) = 2\sigma(2x) - 1$

$[-1, 1]$

$\text{ReLU}(x) = \max(0, x)$

$[0, \infty)$

Perceptron (a specific case of an artificial neuron)



* No backpropagation (binary output)
* Either "fires" or does not fire (no in the middle)
* linear: only a sum, then threshold for output, i.e.

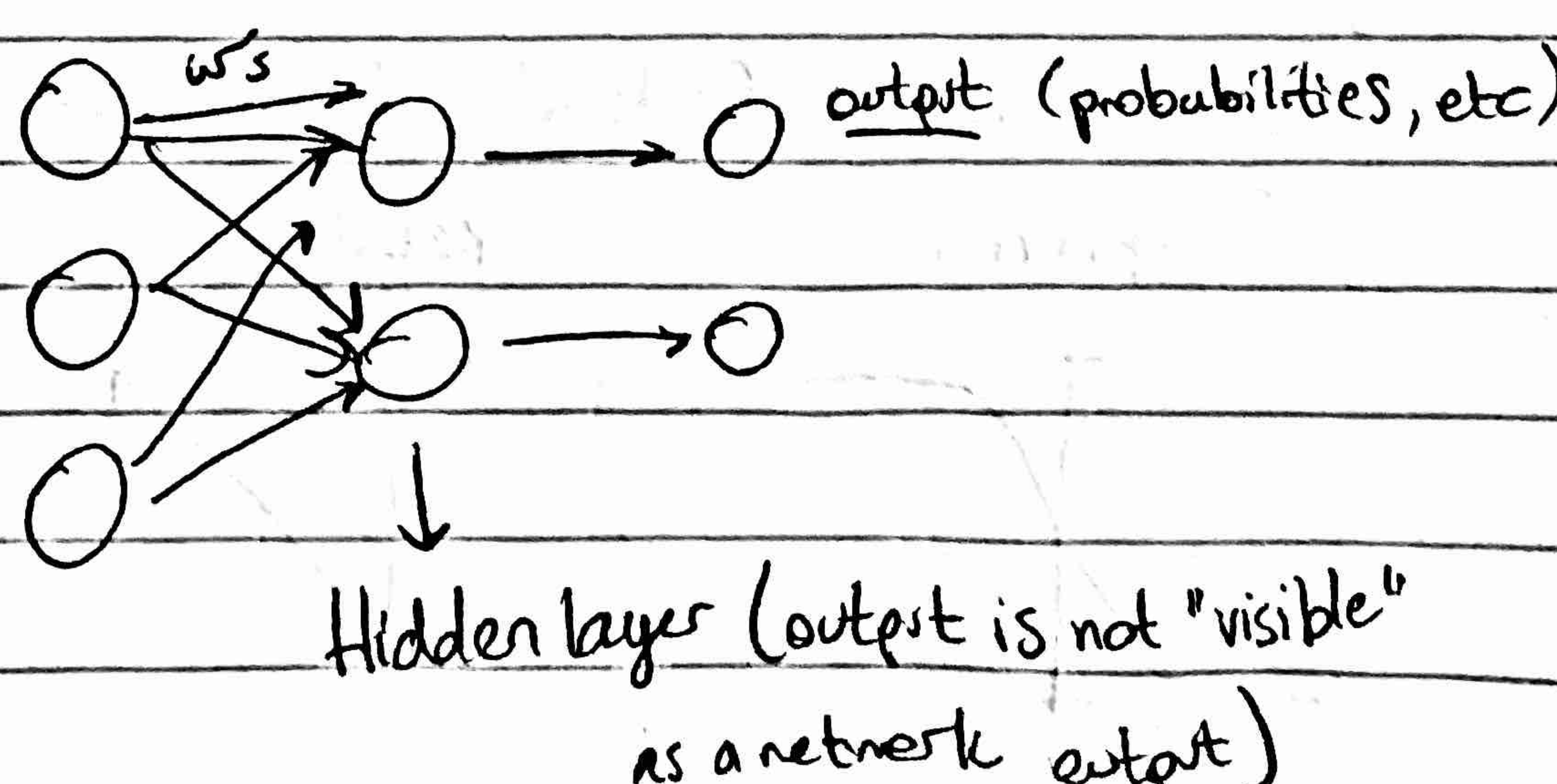
$$y = \text{thres}(\underbrace{x_0 + x_1 w_1 + x_2 w_2}_{\text{Linear}})$$

* Only for 2-class

Multi-Layer Perceptron (MLP) / Feed-Forward NN

Chain several neurons together
Allows for multiclass classification
Adds hidden layer

* Deep NN: many hidden layers



Why do we use hidden layers?

- To capture more complexity (i.e. learn colors, edges, then ears/eyes/nose, then full face for facial recognition)
- Universal approximators: with enough hidden layers, an NN can model <u>any function</u> (Michael Nielsen explains this well — The Universal Approximation Theorem for neural networks on YouTube)

Example

Approximating OR with a perceptron

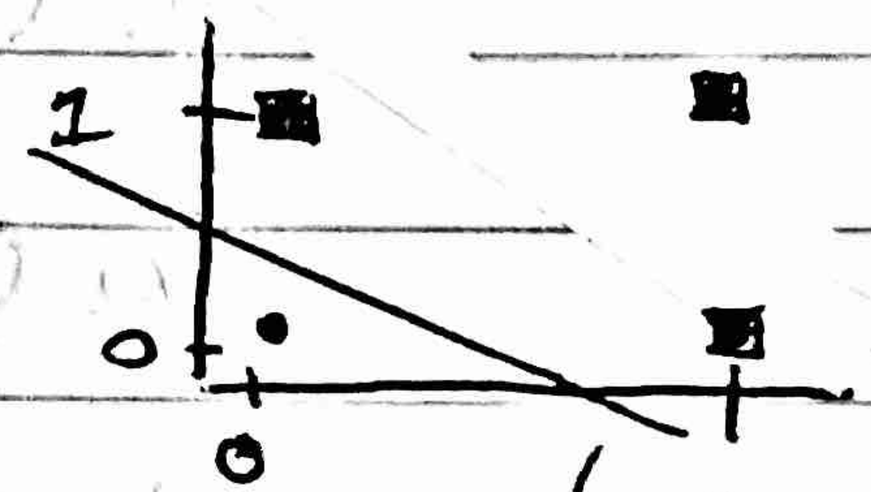
OR: $(0,0) \rightarrow 0$

$(0,1) \rightarrow 1$

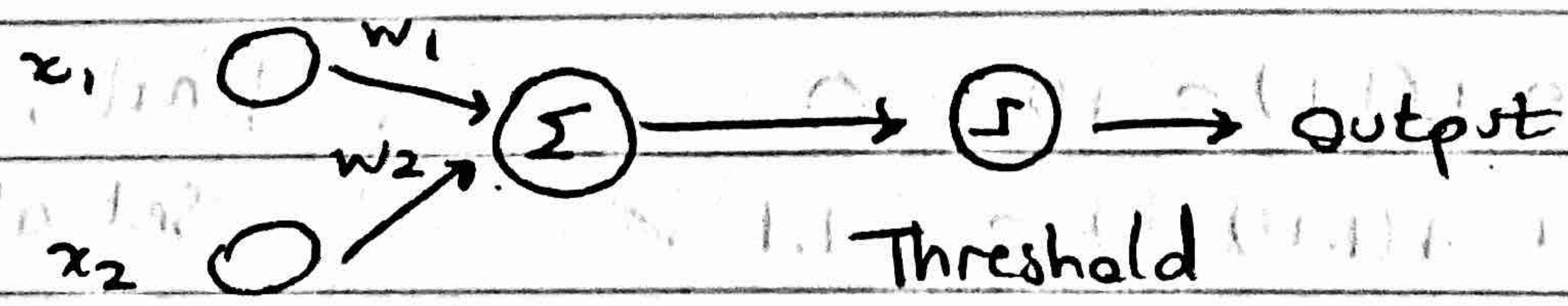
$(1,0) \rightarrow 1$

$(1,1) \rightarrow 1$

Linearly separable



Something like this separates data

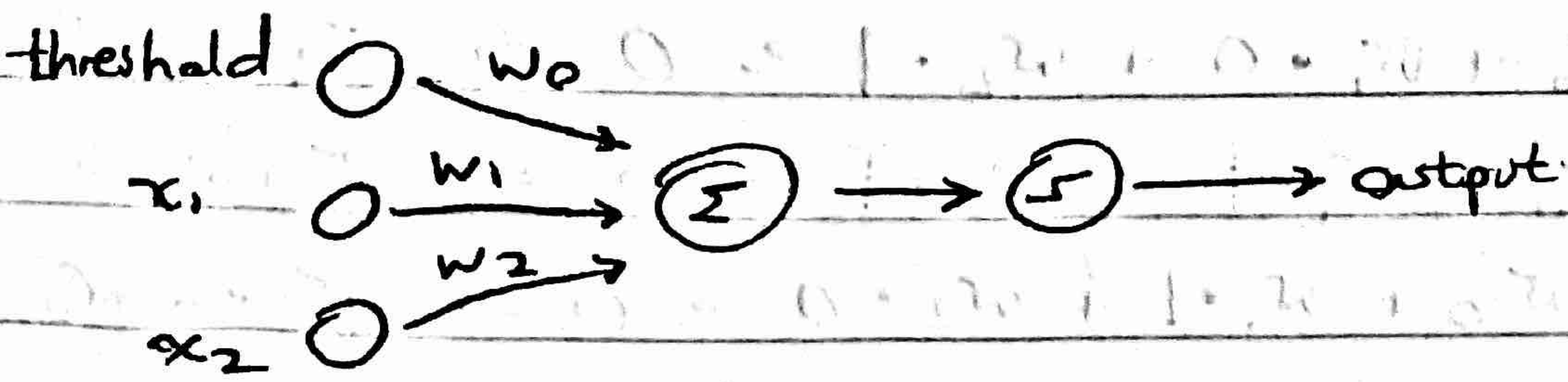


$x_1 w_1 + x_2 w_2 \gtrless$ threshold

$x_1 w_1 + x_2 w_2 -$ threshold $\gtrless 0$

This is the hyperplane to divide the space. (Remember $b = -T$?)

Extend vector so we don't need to deal w/ threshold:



$h(x) = 1(w^T x > t)$

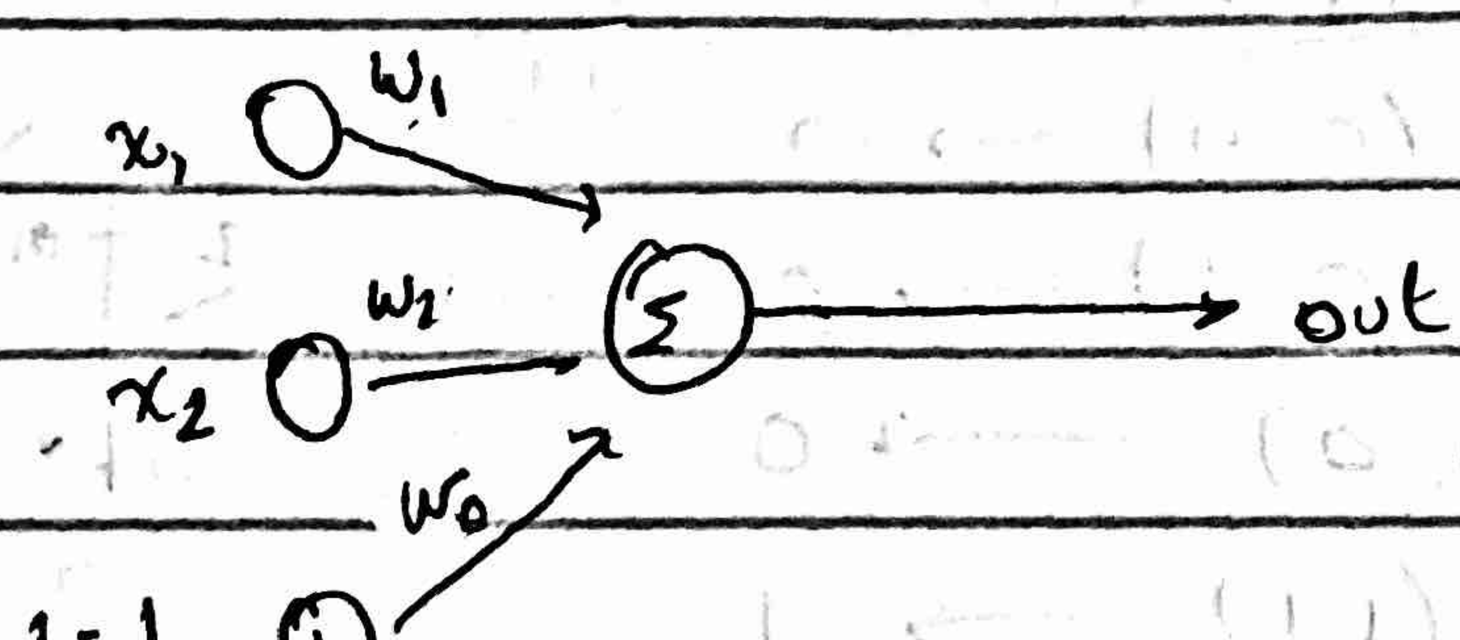
Assume threshold = 0

x_1	x_2	OR (expected)	Output
0	0	0	$w_0 + w_1x_1 + w_2x_2 < 0$
0	1	1	$w_0 + w_1x_1 + w_2x_2 \geq 0$
1	0	1	$w_0 + w_1x_1 + w_2x_2 \geq 0$
1	1	1	$w_0 + w_1x_1 + w_2x_2 \geq 0$

Neuron fires

When we solve these out, we can figure out w_1, w_2

Let's think about it logically:



(0,0) $(1.1)0 + (1.1)0 + 0 = 0$

(1,0) $(1.1)1 + (1.1)0 + 0 = 1.1$ ✓

(0,1) same

(1,1) $(1.1)1 + (1.1)1 + 0 = 2.2$ ✓

What is general set of soln's?

$$w_0 + w_1 \cdot 0 + w_2 \cdot 0 < 0 \Rightarrow w_0 < 0$$

$$w_0 + w_1 \cdot 0 + w_2 \cdot 1 \geq 0 \Rightarrow w_2 > -w_0$$

$$w_0 + w_1 \cdot 1 + w_2 \cdot 1 \geq 0 \Rightarrow w_1 + w_2 > -w_0$$

$$w_0 + w_1 \cdot 1 + w_2 \cdot 0 \geq 0 \Rightarrow w_1 > -w_0$$

$w_0 = -0.5$

$w_0 = -1, \; w_1 = 1.1 \quad w_2 = 1.1$