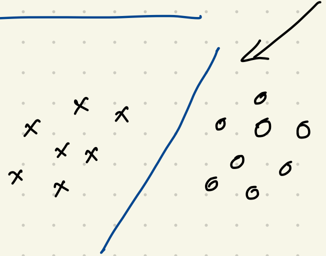
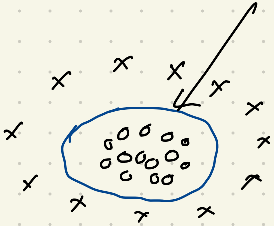
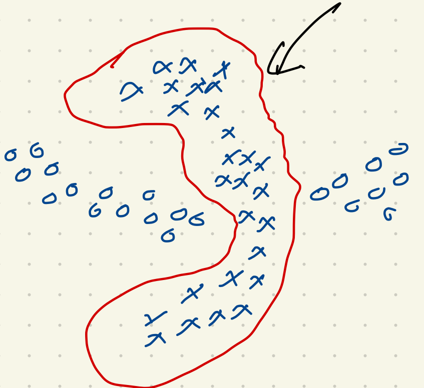
Neural Networks :



linear function



nonlinear function



more complex nonlinear function

$D = \{(x^i, y^i)\}_{i=1}^{N}$, $x^i \in \mathbb{R}^d$, $y^i \in \{+1, -1\}$

find $h_{w,b}(x^i) \approx y^i$, $\forall\, i = 1, \dots, N$

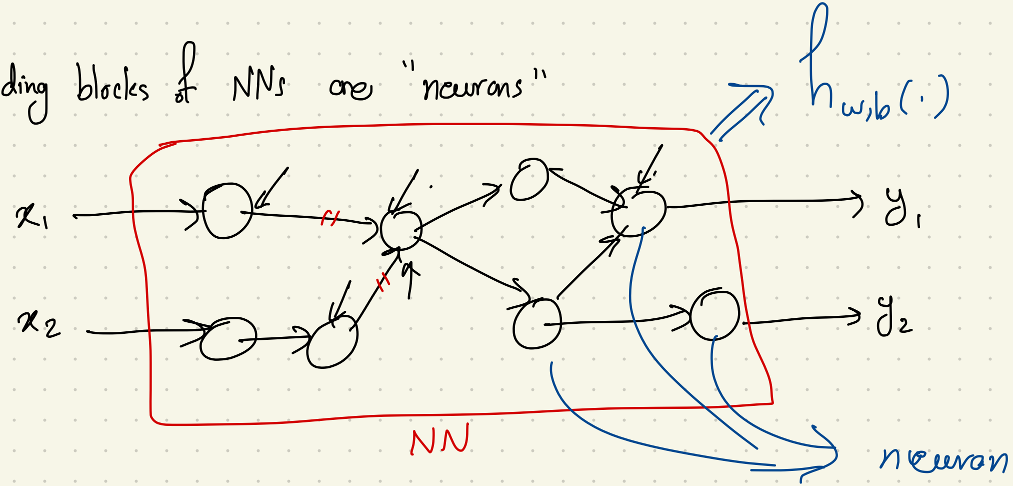
$$x^i \longrightarrow y^i$$

$$w^T x^i + b \simeq h_{w,b}(x^i)$$

$$w^T \varphi(x^i) + b \simeq h'_{w,b}(x^i)$$

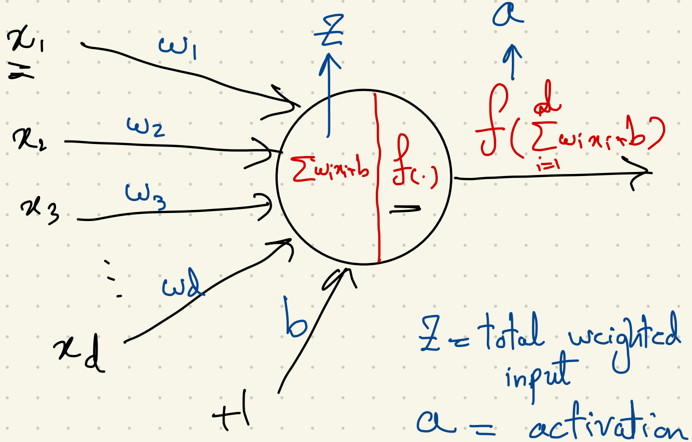
NNs : architectures that allow learning $\underbrace{\text{complex functions}}_{\text{linear / nonlinear}}$ that can map input data to outputs.

Building blocks of NNs are "neurons"



$$\begin{pmatrix}x_1\\x_2\end{pmatrix} \xrightarrow[h_{w,b}(\cdot)]{} \begin{pmatrix}y_1\\y_2\end{pmatrix}$$

A simple neuron can be modeled as follows



inputs: $x_1, x_2, \cdots, x_d, +1$

parameters: $w_1, w_2, \cdots, w_d, b$

$$\text{output} = f(w_1x_1 + w_2x_2 + \cdots + w_d x_d + b)$$

$$= f\left(\sum_{i=1}^{d} w_i x_i + b\right)$$

z = total weighted input

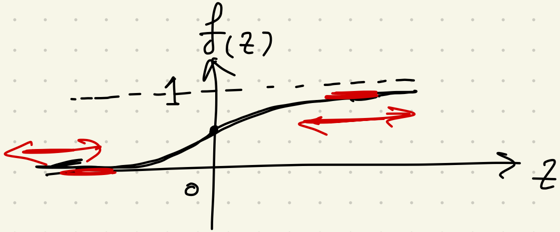
a = activation / output

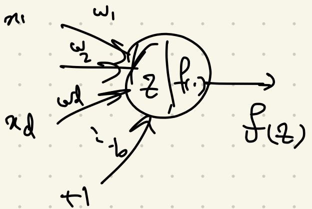
Common choices of activation functions :

- Sigmoid function

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

$\in [0,1]$





$z \to \infty \Rightarrow +1$

$z = 0 \Rightarrow 0.5$

$z \to -\infty \Rightarrow 0$

Good for modeling binary outputs (spam or not)

probabilities (p(email is spam) , p('car in image'))

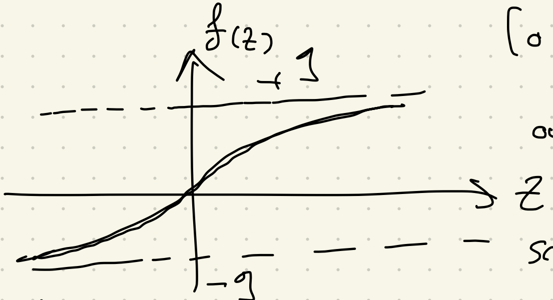
scores that are bounded (student grade [0,100]

$\downarrow$

[0, 1])

- Tangent Hyperbolic function

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



output {+1,-1}

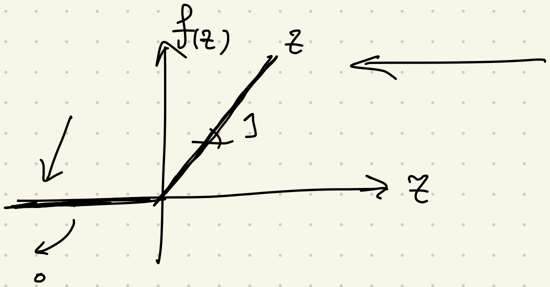
score [-1,+1]

$z \to \infty \Rightarrow 1$

$z = 0 \Rightarrow 0$

$z \to -\infty \Rightarrow -1$

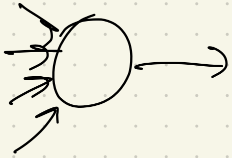
* Rectifier Linear activation function (ReLu)



$$\text{Relu}(Z) = \max\{0, Z\}$$

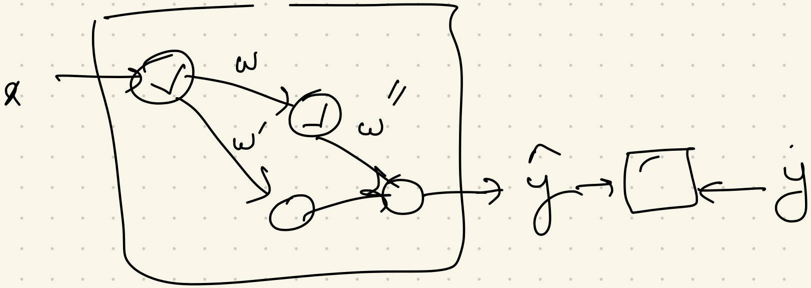
$$= \begin{cases} Z & Z \geq 0 \\ 0 & Z \leq 0 \end{cases}$$

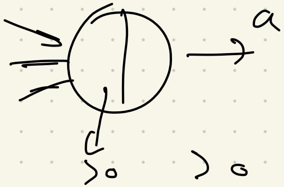
$$\Rightarrow \frac{\partial \text{Relu}(Z)}{\partial Z} = \begin{cases} 1 & Z > 0 \\ \text{undefined} & Z = 0 \\ 0 & Z < 0 \end{cases}$$



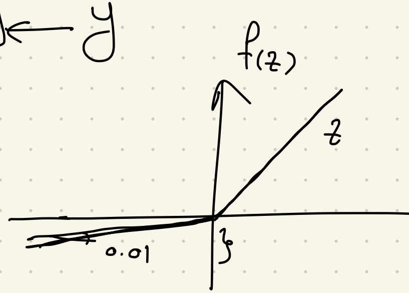
⇒ good for modeling outputs $[0, \infty)$ → stockvalue \$

temperature (kelvin)



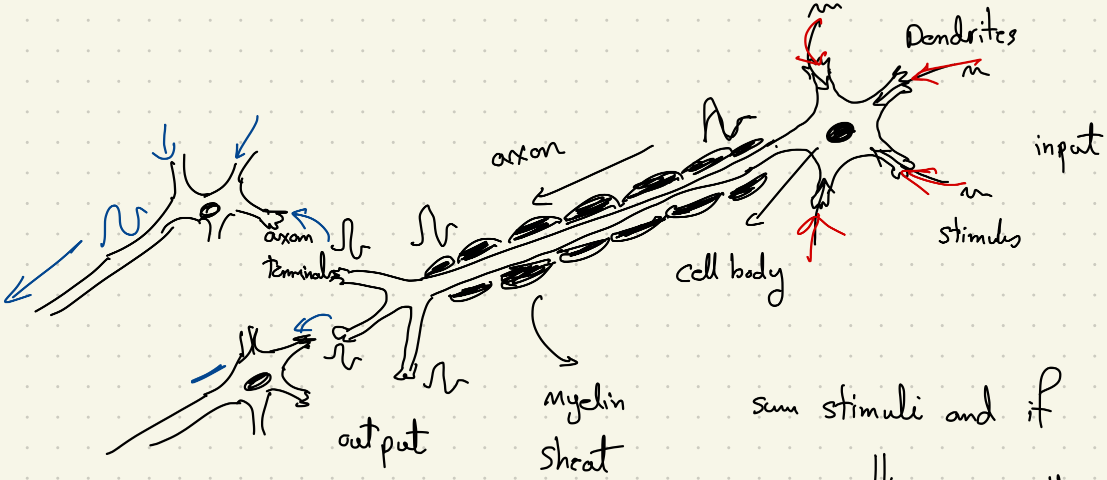


* Leaky Relu function



Relu is more biologically plausible!

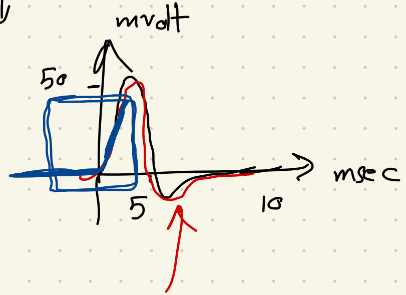
Biological Neuron

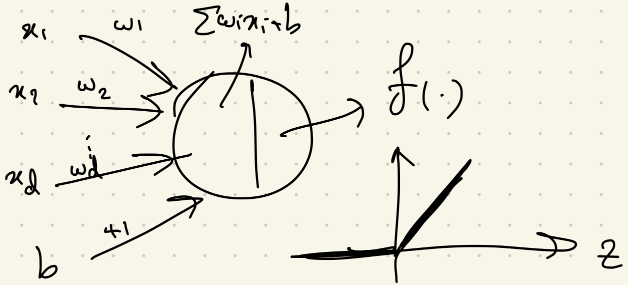


sum stimuli and if > thr

⇓

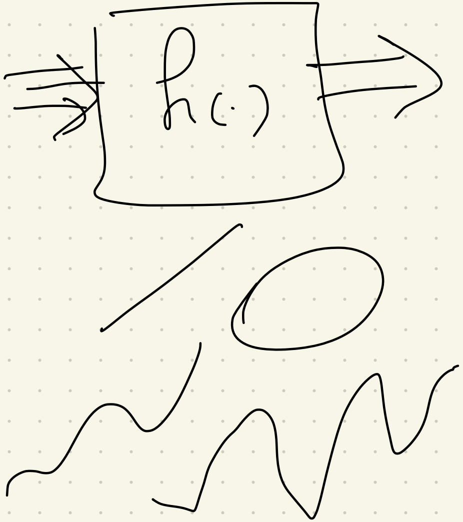
fire

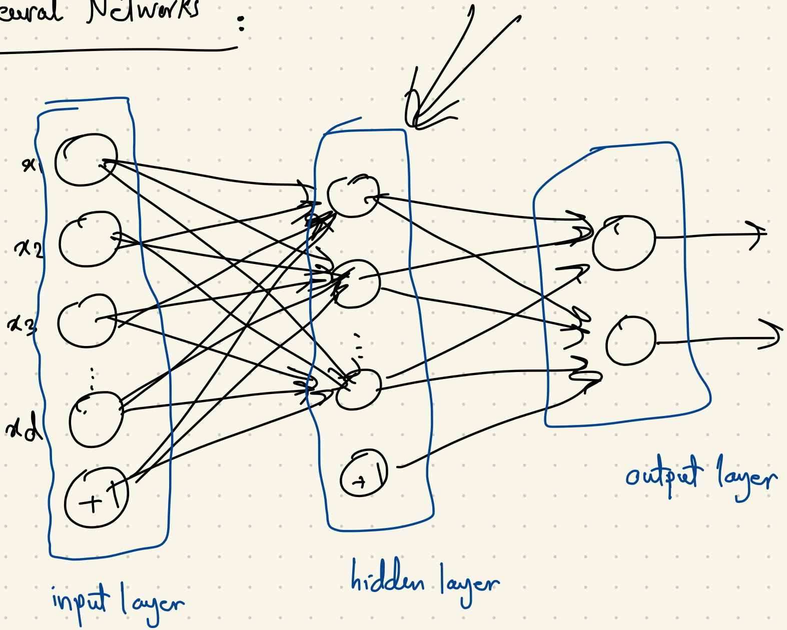




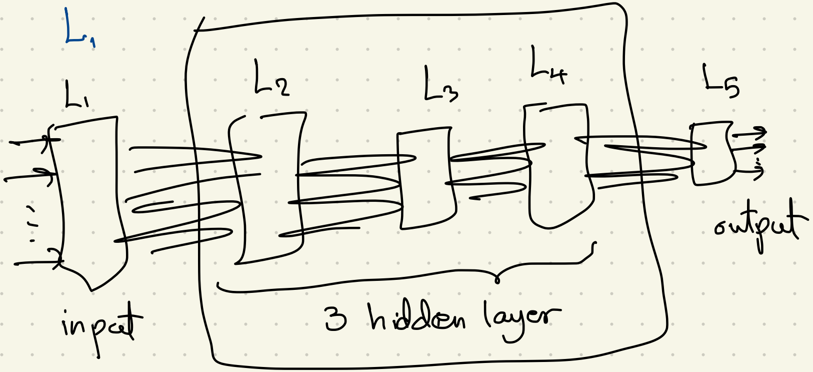
Back to Artificial Neural Networks :

Feedforward NNs :





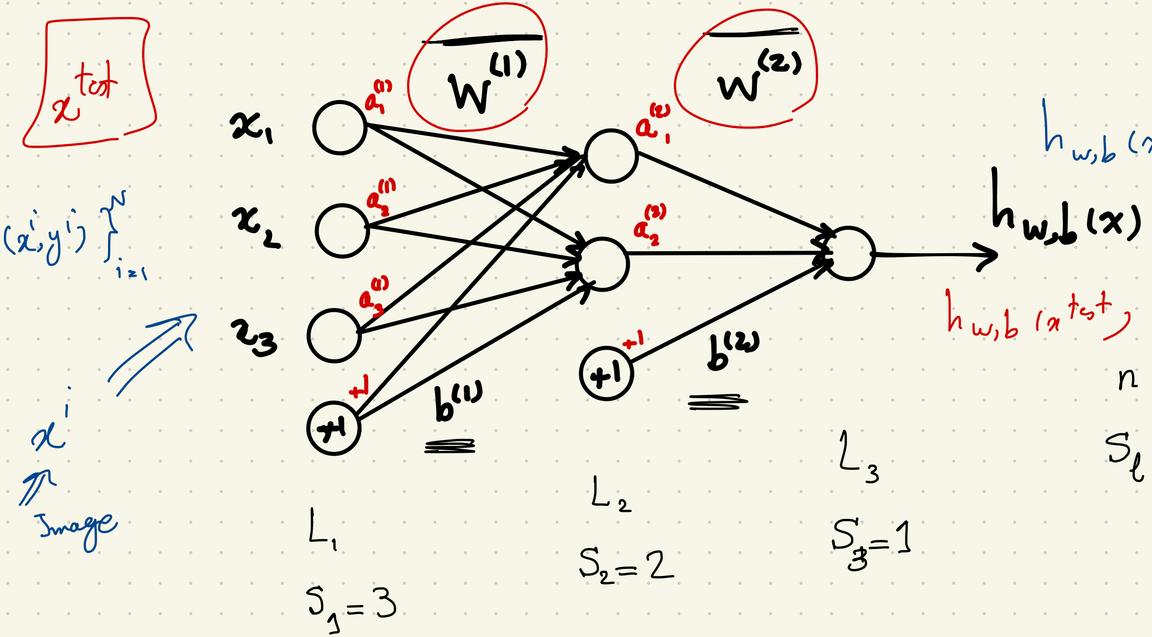
Shallow



Deep

$D = \{(x^i, y^i)\}_{i=1}^{N}$

$h_{w,b}(x^i) \approx y^i$ car/not car

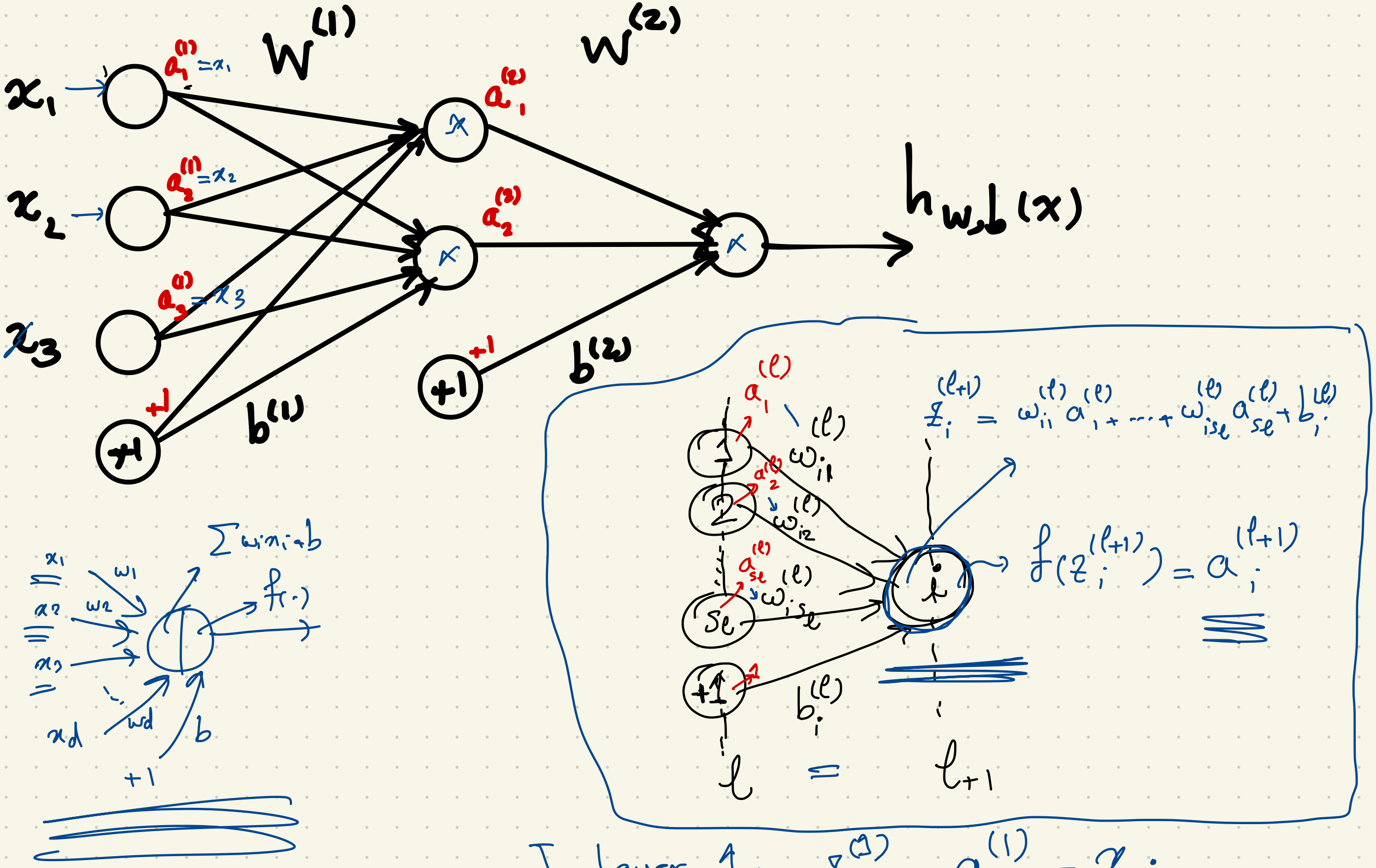


n = #layers

S_ℓ = # nodes in layer ℓ

(excluding intercept/+1)

Want to learn all weights and biases between different layers using training data!



$\sum w_i x_i + b$

$f(\cdot)$

$z_i^{(l+1)} = w_{i1}^{(l)} a_1^{(l)} + \cdots + w_{is_l}^{(l)} a_{s_l}^{(l)} + b_i^{(l)}$

$f(z_i^{(l+1)}) = a_i^{(l+1)}$

In layer 1: $z_i^{(1)} = a_i^{(1)} = x_i$