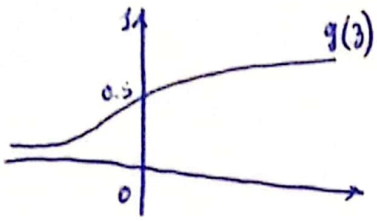
activation functions : Fonctions de seuillages qui peuvent se décomposer en 3 parties [partie non activée ; au dessous du seuil / phase de transition : aux alentours du seuil / partie activée : au dessus du seuil]

① sigmoid

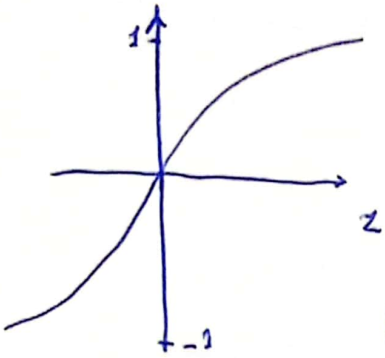


$$a = \frac{1}{1+e^{-3}}$$

$$g'(3) = g(3)\,(1-g(3))$$

less functions
vanishing gradients
exploding gradients

②



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

$$g'(3) = 1 - \left(\tanh(3)\right)^2$$

almost works better than sigmoid because it centers the data ⇒ makes learning easier

disadvantage of these 2 :

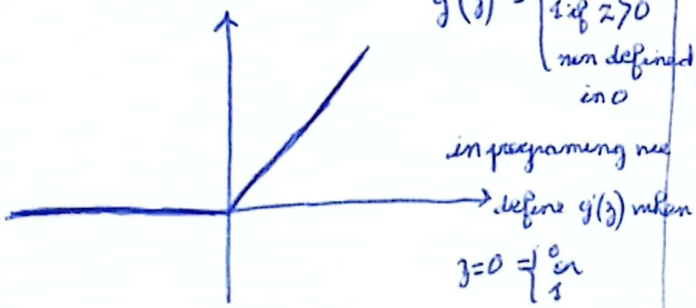
if $z \to +\infty$ ou $z \to -\infty$:

the slope $\to 0$ ⇒ gradient descent is very slow

③ Rectified linear unit : ReLU

$$g(3) = \max(0, 3)$$

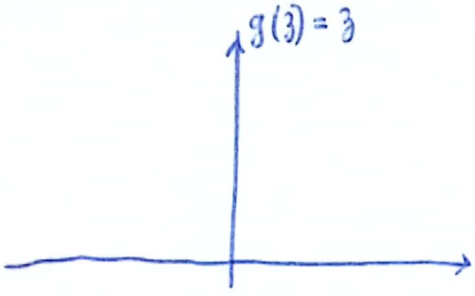
$$g'(3) = \begin{cases} 0 \text{ if } z<0 \\ 1 \text{ if } z>0 \\ \text{non defined in } 0 \end{cases}$$



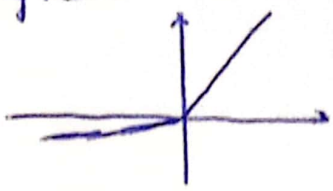
in programming we define $g'(3)$ when $3=0$ = 0 or 1

④ Linear activation function

$$g(3) = 3$$



⑤ Leaky ReLU partie activée : au dessus du seuil



not very used in practice

$$g(3) = \max(0{,}01\,3,\ 3) \qquad g'(3) = \begin{cases} 0{,}01 \text{ if } 3<0 \\ 1 \text{ if } 3>0 \end{cases}$$

⑥ ELU $= \begin{cases} \alpha\,(e^{x}-1) \text{ si } x<0 \\ x \text{ sinon} \end{cases}$

Fonctions d'activations usuelles

~~sigmoïde~~	Image	centré en 0	saturation	evanescence	
sigmoid + cross entropy for binary classification (Log Loss)	[0,1]	Non	valeurs + et -	oui	Bornée
Tanh	[-1,1]	Oui	valeurs + et -	oui	Bornée
ReLU + HSE for regression (+ values)	$[0,+\infty[$	Non	valeurs -	Oui (moins que Sigmoïde et tanh)	
Leaky ReLU	$]-\infty,+\infty[$	non	non	non	facile
Linear + HSE for regression					

softmax + categorical entropy Loss for multiclass

choosing activation function for output layer

binary classification : $y \in \{0, 1\}$

⇒ sigmoid : used only in this case (output of binary classification)

regression : $y = + / -$

⇒ linear activation function

regression : y ne prend que des valeurs positives

⇒ ReLU

for hidden layer :

most common choice : ReLU

faster computation ⇒ faster learning

(gradient descent will be slow if we have a lot of flat portions because $\frac{\partial}{\partial W} J(W,b) = 0$ when $g(z)$ is flat)

⚠ don't use linear activation in hidden layers

---

all linear + output linear

⇒ linear regression

hidden layers linear + output sigmoid

⇒ logistic regression without hidden layer

sigmoid

La dérivée de la fonction sigmoïde est à valeurs dans $[0, \frac{1}{4}]$

⇝ diminue l'amplitude des gradients propagés à travers les couches du réseau de neurones

⇒ pb d'évanescance des gradients

ReLU

gradient égal à $\begin{cases} 0 \text{ dans la zone désactivée} \\ 1 \text{ dans la zone activée} \end{cases}$

⇒ améliore la convergence

Mais

pas dérivable en 0

dérivée nulle sur $\mathbb{R}^-$

à valeurs dans $\mathbb{R}^+$

⇝ optimisation plus compliquée si beaucoup de couches cachées

leaky ReLU

moins d'occurences où le gradient est nul

⇝ meilleure convergence de la descente de gradient