Redes neuronales — Clase #5 — 13/08/2025

Derivación automática

$$f(x_1, x_2) = \ln(x_1) + x_1 \cdot x_2 - \sin(x_2)$$

Evaluación hacia adelante

$v_{-1} = x_1$

$v_0 = x_2$

$v_1 = \ln v_{-1}$

$v_2 = v_{-1} \times v_0$

$v_3 = \sin v_0$

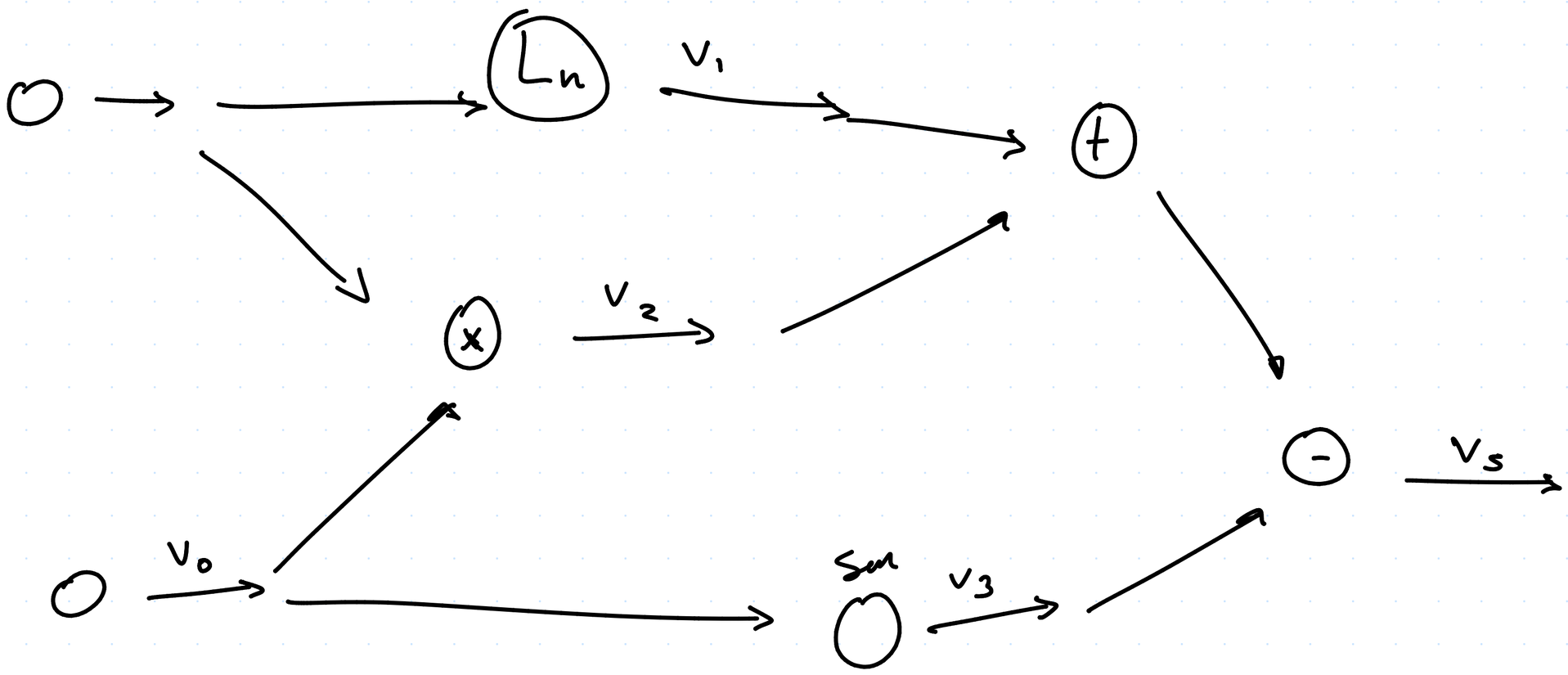
$v_4 = v_1 + v_2$

$v_5 = v_4 - v_3$

$y = v_5$

2	
5	
$v_1 = \ln 2 = 0.69$	
$v_2 = 2 \times 5 = 10$	
$v_3 = \sin 5$	
$v_4 = 0.69 + 10 = 10.69$	
$v_5 = 10.69 - 0.95$	
11.65	

Diagrama del proceso (perceptrons)



Derivación hacia adelante. ; Ejemplo $\frac{\partial y}{\partial x_1}$

$\dot{V}_{-1} = \dot{x}_1$	1
$\dot{V}_0 = \dot{x}_2$	0
$\dot{V}_1 = \frac{\dot{V}_{-1}}{V_{-1}}$	$\dot{V}_1 = \frac{1}{2}$
$\dot{V}_2 = \dot{V}_{-1} \times V_0 + V_{-1} \times \dot{V}_0$	$\dot{V}_2 = 1 \times 5 + 2 \times 0 = 5$
$\dot{V}_3 = \cos V_0 \cdot \dot{V}_0$	$\dot{V}_3 = \cos(5) \cdot 0 = 0$
$\dot{V}_4 = \dot{V}_1 + \dot{V}_2$	$\dot{V}_4 = \frac{1}{2} + 5 = 5.5$
$\dot{V}_5 = \dot{V}_4 - \dot{V}_3$	$\dot{V}_5 = 5.5 - 0$

$\dot{y} = \dot{v}_5$

$\dot{y} = 5.5$

Derivación hacia atrás (Ejemplo)

$$\frac{\partial y}{\partial v_1} = \frac{\partial y}{\partial v_4}\frac{\partial v_4}{\partial v_1}$$

$$\frac{\partial y}{\partial v_2} = \frac{\partial y}{\partial v_4}\frac{\partial v_4}{\partial v_2}$$

$$\frac{\partial y}{\partial v_3} = \frac{\partial y}{\partial v_5}\left(\frac{\partial v_5}{\partial v_3} + \frac{\partial v_4}{\partial v_3}\right) \to 0$$

$$\frac{\partial y}{\partial v_4} = \frac{\partial y}{\partial v_5}\frac{\partial v_5}{\partial v_4}$$

$$\frac{\partial y}{\partial v_5} = 1$$

$$\delta_{-1} = \delta_1 \frac{\partial v_1}{\partial v_{-1}} + \delta_2 \frac{\partial v_2}{\partial v_{-1}}$$

$$\delta_0 = \delta_2 \frac{\partial v_2}{\partial v_0} + \delta_3 \frac{\partial v_3}{\partial v_0}$$

$$\delta_1 = \delta_4 \frac{\partial v_4}{\partial v_1}$$

$$\delta_2 = \delta_4 \frac{\partial v_4}{\partial v_2}$$

$$\delta_3 = \delta_5 \frac{\partial v_5}{\partial v_3}$$

$$\delta_4 = \delta_5 \frac{\partial v_5}{\partial v_4}$$

$$\delta_5 = 1$$

$$s_{-1} = \frac{1}{2} + 1.5 = 5.5 \quad !!$$

$$\delta_{-1} = 1 \frac{1}{v_{-1}} + 1$$

$$\delta_0 = 2 - \cos(s) = 1.7$$

$$\delta_0 = v_{-1} + (-1)\cos(v_0)$$

$$\delta_1 = 1 \cdot 1 = 1$$

$$\delta_2 = 1 \cdot 1 = 1$$

$$\delta_3 = 1 \cdot (-1) = -1$$

$$\delta_4 = 1 \cdot (1) = 1$$

$$\delta_5 = 1$$

- Pytorch
- Tensorflow

} Herramientas para la derivación.

numpy no tiene asociada ciertas derivadas que tensorflow si tiene:

+ code | markdown

i.e. `tensor.sum()`

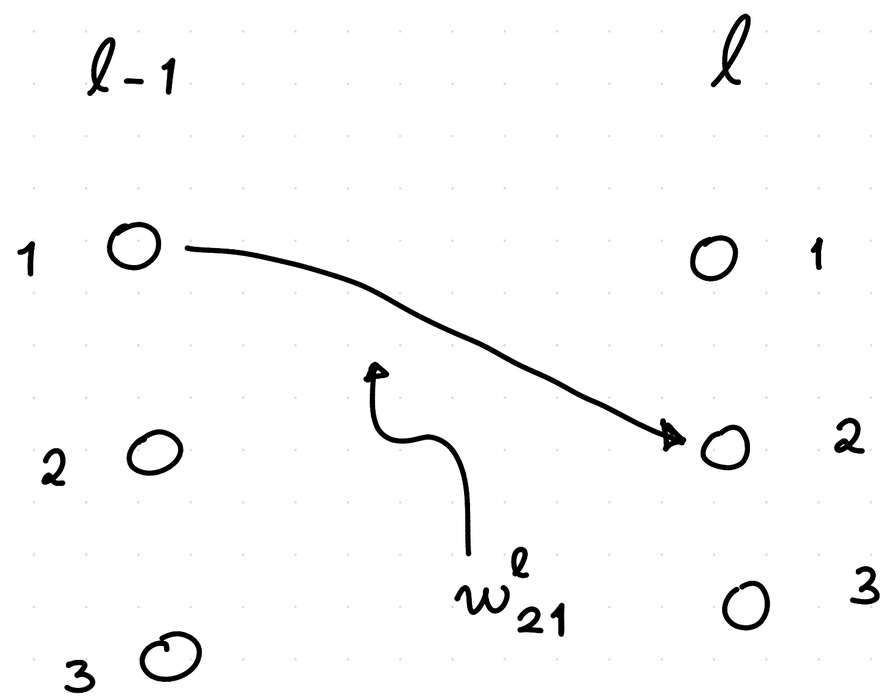
usamos la función tensor

Algoritmo backpropagation

Notación:

w^l_{jk}: Peso del enlace entre neurona $k \to j$ de la capa l.

Nuestra red:



b^l_j: bias de la neurona j en la capa "l"

a^l_j: activación (salida) de la neurona j de la capa "l".

$$a^l_j = \sigma\left(\sum_k w^l_{jk}\, a^{l-1}_k + b^l_j\right)$$

Sesgo

También podemos definir

Función de activación (sigmoide, ReLU, tanh, etc)

$$\sigma(\vec{z})_i \equiv \sigma(z_i)$$

De modo que es posible escribir:

$$a^l = \sigma\underbrace{(w^l a^{l-1} + b^l)}_{z^l}$$

Nota:

/

$$C = \frac{1}{2n} \sum_{x} \| y(x) - a \|^2$$

(Función de costo)

sobre datos de entrenamiento

Entonces...

$$C = \frac{1}{n} \sum_{x} C_x \quad ; \quad \text{con} \quad C_i = \frac{1}{2} \| y(x_i) - a \|^2$$