

Análisis Matemático para Inteligencia Artificial

Verónica Pastor (vpastor@fi.uba.ar),
Martín Errázquin (merrazquin@fi.uba.ar)

Especialización en Inteligencia Artificial

29/7/2022

Repaso

- ① En los videos de repaso definimos funciones de cuyo dominio y codominio eran los reales, la gráfica de la función se representa en \mathbb{R}^2 .
- ② Toda función f describe el cambio de una magnitud (v. dependiente) en términos de otra (v. independiente), cuando esta variable se mueve en cierto intervalo $[x_0, x_0 + h]$ la variación total se mide como $f(x_0 + h) - f(x_0)$.
- ③ Mientras que la variación media es $\frac{f(x_0+h)-f(x_0)}{(x_0+h)-x_0}$. Geométricamente, podemos ver la variación media como la pendiente de la recta secante.
- ④ Cuando hacemos que $h \rightarrow 0$, ...



... esto nos conduce a la definición de derivada de f en

x_0 :

$$\lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h} := f'(x_0)$$

Clasificación de funciones

Dada $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$.

dominio

- Si $m = 1$ diremos que es una función

- escalar, si $n = 1$,

- campo escalar, $n > 1$. Graf representa una super $n = 2$



- Si $m > 1$ diremos que es una función

- vectorial, si $n = 1$,

- campo vectorial, $n > 1$.

Conjuntos de Nivel Dada $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ el conjunto de nivel k de f , $L_k \subset \mathbb{R}^n$, definido por:

$$L_k = \{x \in \mathbb{R}^n / x \in D \wedge f(x) = k\}$$

$$f(x,y) = \sqrt{x+y}$$

$$\text{Dom } f = \{(x,y) : x+y \geq 0\}$$

$$k \in \text{Im } f$$

La representación geométrica de L_k se obtiene identificando gráficamente los puntos del dominio de la función para los cuales el valor de f es igual a k , para graficar no es necesario agregar un eje.

Ej $f(x,y) = x^2 + y^2$

$$L_{-1} : x^2 + y^2 = -1$$

$$L_0 : x^2 + y^2 = 0$$

$$L_1 : x^2 + y^2 = (\sqrt{2})^2$$

$$\text{Dom } f : \mathbb{R}^2 \quad \text{Im } f : \mathbb{R}^+ \quad L_1 : x^2 + y^2 = 1 \quad L_4 : x^2 + y^2 = 4 = z^2$$



Derivando campos ...

- escalares: Sea $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$, $(x_1, \dots, x_n)^T \mapsto f((x_1, \dots, x_n)^T)$, se definen las **derivadas parciales** como:

$$\frac{\partial f}{\partial x_1} = \lim_{h \rightarrow 0} \frac{f(x_1 + h, x_2, \dots, x_n) - f(x_1, x_2, \dots, x_n)}{h}$$

; fijas ;

$$\frac{\partial f}{\partial x_n} = \lim_{h \rightarrow 0} \frac{f(x_1, x_2, \dots, x_n + h) - f(x_1, x_2, \dots, x_n)}{h}$$

Se define el **gradiente** como: $\nabla f = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)$.

equiv. $f'(x_0)$

Importante: El gradiente apunta en la dirección de máximo crecimiento.

- vectoriales: Sea $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$,

$(x_1, \dots, x_n)^T \mapsto (f_1((x_1, \dots, x_n)^T), \dots, f_m((x_1, \dots, x_n)^T))$, se define el **jacobiano** como:

$$\frac{\partial f_1}{\partial x}(x, y) = 1 + 0 = 1 \quad \frac{\partial f_2}{\partial x}(x, y) = 3y$$

$$\frac{\partial f_1}{\partial y}(x, y) = 1 \quad \frac{\partial f_2}{\partial y}(x, y) = \cos(x)$$

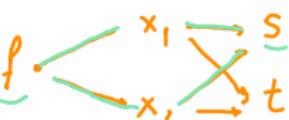
$$J_f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

$$f(x, y) = \left(\frac{x-y}{f_1}, \frac{3y}{f_2}, \sin(y) \right)$$

$$J = \begin{bmatrix} 1 & 1 \\ 3y & 3x \\ \cos y & 0 \end{bmatrix}$$

Regla de la Cadena en forma matricial

Sea $f(x_1(s, t), x_2(s, t))$



$$\frac{\partial f}{\partial s} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial s} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial s}$$

$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial t}$$

$$f(g(x)) = \operatorname{sen}(x^2)$$

$$g(x) = x^L$$

$$f(j) = \operatorname{sen} y$$

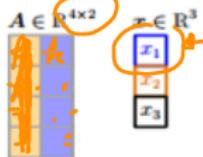
Y luego

$$\frac{df}{d(s, t)} = \frac{df}{dx} \frac{dx}{d(s, t)} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1}{\partial s} & \frac{\partial x_1}{\partial t} \\ \frac{\partial x_2}{\partial s} & \frac{\partial x_2}{\partial t} \end{bmatrix} \rightarrow \nabla x_1 \quad \nabla x_2$$

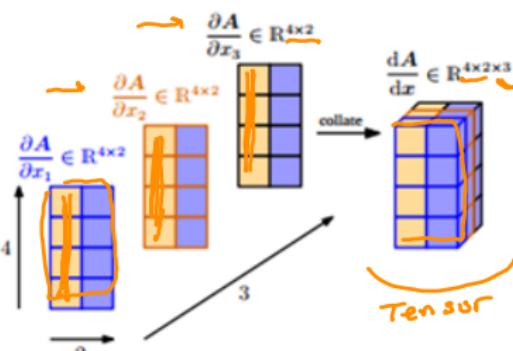
Recordemos reglas de derivación:

- $\frac{\partial(f+g)(s)}{\partial s} = \frac{\partial f}{\partial s} + \frac{\partial g}{\partial s}$
- $\frac{\partial(fg)(s)}{\partial s} = \frac{\partial f}{\partial s}g(s) + f(s)\frac{\partial g}{\partial s}$
- $\frac{\partial(f/g)(s)}{\partial s} = \frac{g(s) - f(s)}{(g(s))^2}$

Derivada de matrices



Partial derivatives:



(a) Approach 1: We compute the partial derivative $\frac{\partial A}{\partial x_1}, \frac{\partial A}{\partial x_2}, \frac{\partial A}{\partial x_3}$, each of which is a 4×2 matrix, and collate them in a $4 \times 2 \times 3$ tensor.



(b) Approach 2: We re-shape (flatten) $A \in \mathbb{R}^{4 \times 2}$ into a vector $\tilde{A} \in \mathbb{R}^8$. Then, we compute the gradient $\frac{d\tilde{A}}{dx} \in \mathbb{R}^{8 \times 3}$. We obtain the gradient tensor by re-shaping this gradient as illustrated above.

Matriz Hessiana

La **matriz Hessiana** es aquella cuyas derivadas de orden 2 de f respecto a $x \in \mathbb{R}^n$ se ubican:

derivada con respecto a x_1 dos veces

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_i} & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

derivada segunda

s: $\nabla f = -P.C.R.$

$|H| > 0$ $\frac{\partial^2 f}{\partial x_1^2} > 0$ min loc

$|H| < 0$ $\frac{\partial^2 f}{\partial x_1^2} < 0$ max loc

$|H| = 0$ crit. no clásica



Teorema: $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in C^2$ (tres derivadas segundas y cont) ent. la matriz hessiana es simétrica. Es decir las derivadas cruzadas son iguales

Ej $f: \mathbb{R}^2 \rightarrow \mathbb{R}$, $f(x, y) = 3xy^2$. Luego $H =$

$$\frac{\partial f}{\partial x} = 3y^2 \quad \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right) = \frac{\partial^2 f}{\partial x^2} = 0$$

$$\frac{\partial f}{\partial y} = 6xy \quad \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial x} \right) = 6y$$

$$\frac{\partial^2 f}{\partial x \partial y} = 6x \quad \frac{\partial^2 f}{\partial y^2} = 6y$$

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 0 & 6y \\ 6y & 6x \end{bmatrix}$$

Matriz sim $H = H^T$, $6y = 6y$

ES: $S^T H S = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$

Formas Cuadráticas

Aplicación: Polinomio de Taylor

Sea f un campo escalar $f : \mathbb{R}^n \rightarrow \mathbb{R}$, asumiendo que posee derivadas parciales de todo orden en un entorno de un punto $a \in \mathbb{R}^n$, se define el **polinomio de Taylor** de grado k :

$$P_k(x) = f(a) + \underbrace{\sum_{i=1}^n \frac{\partial f}{\partial x_i}(a)(x_i - a_i)}_{\text{Termo lineal}} + \frac{1}{2!} \underbrace{\sum_{i,j=1}^n \frac{\partial^2 f}{\partial x_i \partial x_j}(a)(x_i - a_i)(x_j - a_j)}_{\text{Termo de segundo orden}} + \dots + \dots + \underbrace{\frac{1}{k!} \sum_{\text{indices}} \frac{\partial^k f}{\partial x_{i_1} \dots \partial x_{i_k}}(a)(x_{i_1} - a_{i_1}) \dots (x_{i_k} - a_{i_k})}_{\text{Termo de grado } k}$$

Función escalar

$$P_k(x) = f(a) + f'(a)(x-a) + \frac{f''(a)(x-a)^2}{2!} + \dots$$



Diferenciación Automática

$$\frac{df}{dx} = \frac{dg}{df} \cdot \frac{df}{dx}$$
$$\frac{dg}{df} = \frac{dh}{g} \cdot \frac{di}{f} \frac{df}{di} \frac{dg}{dh} \cdot \frac{dh}{f}$$

Sean, para una función f :

- x_1, \dots, x_d las variables de entrada
- x_{d+1}, \dots, x_{D-1} las variables intermedias
- x_D la variable de salida
- g_i funciones elementales
- $Hij(x_i)$ el conjunto de nodos hijos de cada x_i



$$z = x^2 + \sqrt{h} \cdot x^2$$

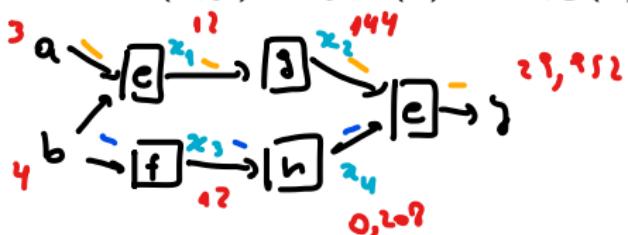
Así queda definido un **grafo de cómputo**. Recordando que $f = D$, tenemos que $\frac{\partial f}{\partial x_D} = 1$. Para las otras variables x_i aplicamos la regla de la cadena:

$$\frac{\partial f}{\partial x_i} = \sum_{x_j \in Hij(x_i)} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i} = \sum_{x_j \in Hij(x_i)} \frac{\partial f}{\partial g_j} \frac{\partial g_j}{\partial x_i}$$

- La diferenciación automática se puede utilizar siempre que la función pueda representarse como un grafo de cómputo.
- La gran ganancia de este mecanismo está en que cada función sólo precisa saber cómo derivarse a sí misma, permitiendo OOP.

Diferenciación Automática: ejemplo

Sean $e(x, y) = xy$, $f(x) = 3x$, $g(x) = x^2$, $h(x) = \operatorname{sen}(x)$



$$\begin{aligned} &\cdot \frac{de}{dx} = y, \quad \frac{de}{dy} = x \\ &\cdot f' = 3 \\ &\cdot g' = 2x \\ &\cdot h' = \cos(x) \end{aligned}$$

$$\begin{aligned} \frac{de}{da} &= \frac{dy}{dx_2} \cdot \frac{dx_2}{dx_1} \cdot \frac{dx_1}{da} = \frac{de}{dx}(x_1, x_2) \cdot g'(x_2) \cdot \frac{de}{dx}(a, b) = x_2 \cdot 2 \cdot x_1 \cdot b \\ &= 0.208 \cdot 2 \cdot 12 \cdot 4 \end{aligned}$$

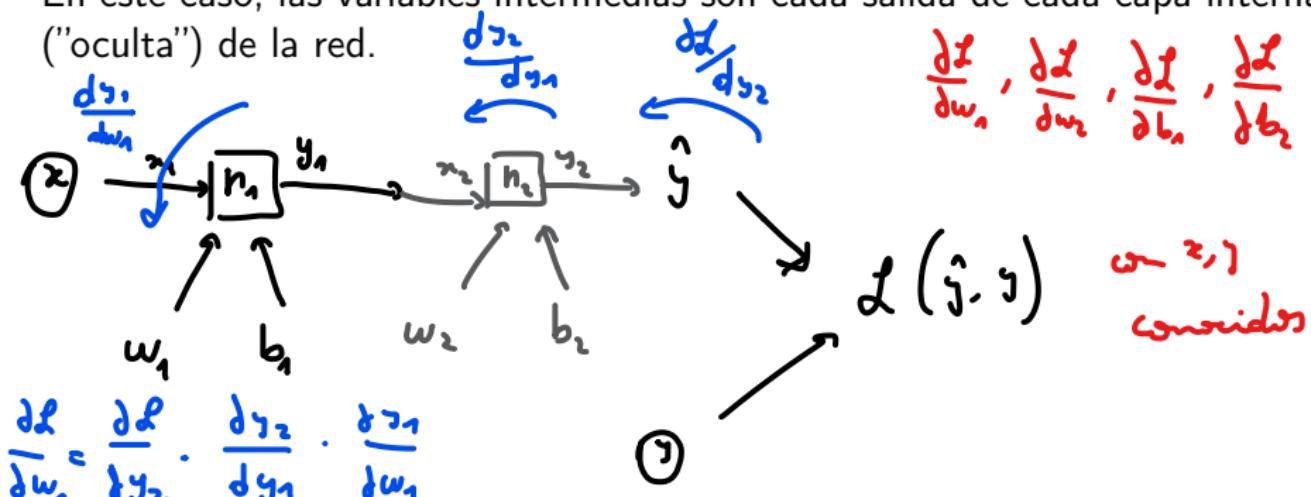
$$\begin{aligned} \frac{dy}{db} &= \frac{dy}{dx_1} \cdot \frac{dx_1}{db} + \frac{dy}{dx_2} \cdot \frac{dx_2}{db} = \frac{dy}{dx_1} \cdot \frac{dx_1}{db} + \frac{dy}{dx_2} \cdot \frac{dx_2}{db} + \frac{dy}{dx_3} \cdot \frac{dx_3}{db} + \frac{dy}{dx_4} \cdot \frac{dx_4}{db} \\ &= \frac{de}{dx}(x_1, x_2) \cdot g'(x_2) \cdot \frac{de}{dx}(a, b) + \frac{de}{dx}(x_2, x_3) \cdot h'(x_3) \cdot f'(b) \end{aligned}$$

Backpropagation

¿Dónde se aplica la diferenciación automática? En **Backpropagation** (o simplemente Backprop), el algoritmo utilizado para entrenar redes neuronales.

¿Qué función cumple? La de computar las derivadas de la función de error/costo respecto de *cada* parámetro de la red neuronal.

En este caso, las variables intermedias son cada salida de cada capa interna ("oculta") de la red.

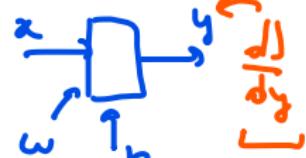


Redes neuronales (I)

Un perceptrón/neurona es un estimador de la forma:

$$\hat{y} = g(\underbrace{w \cdot x + b}_z)$$

$$\frac{\partial J}{\partial x}, \frac{\partial J}{\partial w}, \frac{\partial J}{\partial b}$$



donde en su forma más simple $x, y, w, b \in \mathbb{R}$ y $g : \mathbb{R} \rightarrow \mathbb{R}$ es una función no lineal como puede ser la sigmoidea $\sigma(z) = \frac{1}{1+e^{-z}}$.

Si se define la función $J(W, b)$ de error respecto de los parámetros W y b se puede comprobar que, definiendo $z = w \cdot x + b$ y suponiendo conocido

$$\frac{dJ}{d\hat{y}} = dY \in \mathbb{R}$$



$$\frac{\partial \hat{y}}{\partial z} = g'(z) \in \mathbb{R}$$

$$\frac{\partial J}{\partial W} = \frac{dJ}{d\hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial W} = dY \cdot g'(z) \cdot x \in \mathbb{R}$$

$$\frac{\partial J}{\partial b} = \frac{dJ}{d\hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial b} = dY \cdot g'(z) \cdot 1 \in \mathbb{R}$$

Redes neuronales (II)

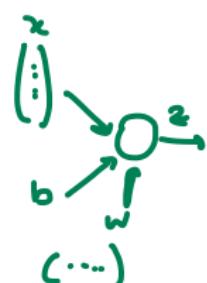
Si ahora consideramos múltiples entradas, es decir $x \in \mathbb{R}^n$, $W \in \mathbb{R}^{1 \times n}$:

$$x = (x_1, x_2, \dots, x_n)$$

$$\hat{y} = g(\underbrace{W \cdot x + b}_{\text{n+1}}) = g(w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n + b)$$

Entonces ahora para cada elemento de $W = (w_1, \dots, w_n)$ vale lo anterior, y por tanto se puede comprobar que

$$\frac{\partial J}{\partial W} = \nabla_J(W) = (\underbrace{dY \cdot g'(z)}_{\text{R}} \cdot x_1, \dots, \underbrace{dY \cdot g'(z)}_{\text{R}} \cdot x_n) = \underbrace{dY \cdot g'(z)}_{\text{R}} \cdot \underbrace{x^T}_{n \times 1}$$
$$dY \cdot g'(z) \cdot (x_1, \dots, x_n)$$



$$\frac{\partial J}{\partial b} = \underbrace{dY \cdot g'(z)}_{\text{R}} \in \mathbb{R} \checkmark$$

$$\in \mathbb{R}^{n \times n} \checkmark$$

Redes neuronales (III)

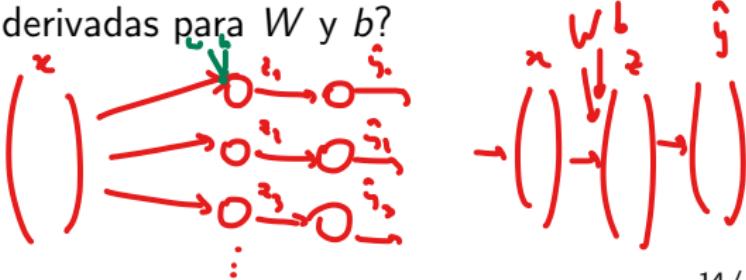
Una capa en una red neuronal se define como un vector de k neuronas en paralelo. Una propiedad atractiva de este formato es que se puede considerar a la salida de una capa $y \in \mathbb{R}^k$ como simplemente el x de la capa siguiente. Por convención (y eficiencia computacional) se suele utilizar la misma no-linealidad g para todas las neuronas de la capa.

Nuevamente tenemos:

$$\hat{y} = g(W \cdot x + b)$$

donde $x \in \mathbb{R}^n$, $W \in \mathbb{R}^{k \times n}$, $b \in \mathbb{R}^k$ y se conviene $g(z) = \begin{pmatrix} g(z_1) \\ \vdots \\ g(z_k) \end{pmatrix}$

¿Y ahora cómo se calculan las derivadas para W y b ?



Redes neuronales (IV)

En el caso de b es simple:

$$\frac{\partial J}{\partial b} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial b} = \begin{pmatrix} dY_1 \\ \vdots \\ dY_k \end{pmatrix} \odot \underbrace{\begin{pmatrix} g'(z_1) \\ \vdots \\ g'(z_k) \end{pmatrix}}_{\text{elem.-wise}} \odot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = dY \odot g'(z) \in \mathbb{R}^k$$

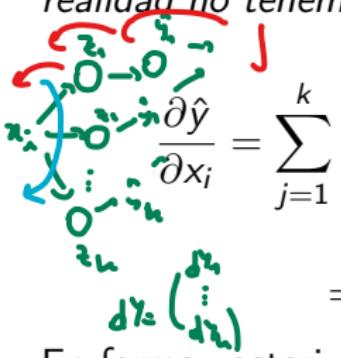
Ahora para cada elemento de W tenemos:

$$\frac{\partial J}{\partial W} = \begin{pmatrix} \frac{\partial J}{\partial W_{1,1}} & \cdots & \frac{\partial J}{\partial W_{1,n}} \\ \frac{\partial J}{\partial W_{2,1}} & \cdots & \frac{\partial J}{\partial W_{2,n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial J}{\partial W_{k,1}} & \cdots & \frac{\partial J}{\partial W_{k,n}} \end{pmatrix} = \begin{pmatrix} \nabla_J(W_{1,:}) \\ \vdots \\ \nabla_J(W_{k,:}) \end{pmatrix} \stackrel{(II)}{=} \begin{pmatrix} \overbrace{dY_1}^{\mathbb{R}^k} \cdot \overbrace{g'(z_1)}^{\mathbb{R}^n} \cdot \overbrace{x^T}^{\mathbb{R}^m} \\ \vdots \\ \overbrace{dY_k}^{\mathbb{R}^k} \cdot \overbrace{g'(z_k)}^{\mathbb{R}^n} \cdot \overbrace{x^T}^{\mathbb{R}^m} \end{pmatrix} =$$

$$= \begin{pmatrix} dY_1 \\ \vdots \\ dY_k \end{pmatrix} \odot \begin{pmatrix} g'(z_1) \\ \vdots \\ g'(z_k) \end{pmatrix} \cdot x^T = dY \odot g'(z) \cdot x^T \in \mathbb{R}^{k \times n}$$

Redes neuronales (V): Backpropagation

¿Cómo se encadena esto? Nosotros estamos dando por conocida la derivada del error respecto de la salida de la capa, $dY = \frac{dJ}{d\hat{y}}$, pero en realidad no tenemos idea si estamos en una capa intermedia o no.


$$\frac{\partial \hat{y}}{\partial x_i} = \sum_{j=1}^k dY_j \cdot g'(z_j) \cdot W_{j,i} = \langle \begin{pmatrix} dY_1 \cdot g'(z_1) \\ \vdots \\ dY_k \cdot g'(z_k) \end{pmatrix}, \begin{pmatrix} W_{1,i} \\ \vdots \\ W_{k,i} \end{pmatrix} \rangle =$$
$$= \langle dY \odot g'(z), W_{:,i} \rangle = \underbrace{W_{i,:}^T}_{1 \times k} \cdot \underbrace{dY \odot g'(z)}_{k \times 1} \in \mathbb{R}$$

En forma vectorizada:

$$dX = \frac{\partial J}{\partial x} = \begin{pmatrix} \frac{\partial J}{\partial x_1} \\ \vdots \\ \frac{\partial J}{\partial x_n} \end{pmatrix} = \begin{pmatrix} W_{1,:}^T \cdot dY \odot g'(z) \\ \vdots \\ W_{n,:}^T \cdot dY \odot g'(z) \end{pmatrix} = \underbrace{W^T}_{h \times k} \cdot \underbrace{dY \odot g'(z)}_{k \times 1} \in \mathbb{R}^{n \times 1} \quad \checkmark$$

Y ese dX no es otra cosa que el dY de la capa anterior!