

# Análisis Matemático para Inteligencia Artificial

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

Especialización en Inteligencia Artificial

23/9/2022

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

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

- escalar, si  $n = 1$ ,

- campo escalar,  $n > 1$ .

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

- vectorial, si  $n = 1$ ,

- campo vectorial,  $n > 1$ .

$f(x) = x^3$

$f(x) = |x|$



$f(x) = x^2$   $f: \mathbb{R} \rightarrow \mathbb{R}$

$f(x) = \frac{1}{\sqrt{x}}$  ; Dom  $f = \{x \in \mathbb{R} : x > 0\}$

$f(x, y) = x^2 + y^2$   $f: \mathbb{R}^2 \rightarrow \mathbb{R}$

$f(x) = (x^2, 2x)$  se suele decir  $x$ : parámetro

$f(x, y) = (x^2, 0, xy)$   $f: \mathbb{R}^2 \rightarrow \mathbb{R}^3$

→ **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:

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

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. Por ej.  $f(x, y) = x^2 + y^2$

$L_0 = \{\vec{x} \in \mathbb{R}^2 : \vec{x} \in D = \mathbb{R}^2 \wedge x^2 + y^2 = 0\}$   
 $= \{(0, 0)\}$  y  $L_1 = \{\vec{x} \in \mathbb{R}^2 : x^2 + y^2 = 1\}$



$L_4 = \{\vec{x} \in \mathbb{R}^2 : x^2 + y^2 = 4\}$   
 $L_{-1} = \emptyset$


# 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:

*derivada respecto a  $x_1$*

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

*$D_{\frac{\partial}{\partial x_1}} f$*


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

Se define el **gradiente** como:  $\nabla f = \left( \frac{\partial f}{\partial x_1} \dots \frac{\partial f}{\partial x_n} \right)$ . *(evaluado en un pto) y  $\|\nabla f\|$  mide la prod. máxima*

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

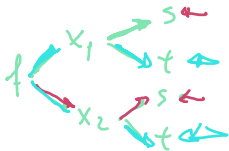
- vectoriales: Sea  $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $(x_1, \dots, x_n)^T \mapsto (f_1((x_1, \dots, x_n)^T), \dots, f_m((x_1, \dots, x_n)^T))$ , se define el **jacobiano** como:

$$J_f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \rightarrow \begin{matrix} \nabla f_1(x_1, \dots, x_n) \\ \vdots \\ \nabla f_m \end{matrix}$$

# Regla de la Cadena en forma matricial

$f(g(s,t))$  comp. de funciones

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



$$\begin{cases} \frac{\partial f}{\partial s} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial s} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial s} \\ \frac{\partial f}{\partial t} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial t} \end{cases}$$

$\cos(s^2 + 2t)$

$f'(g(s)) \cdot g'(s) = -\sin(s^2 + 2) \cdot (2s)$

Y luego

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

Recordemos reglas de derivación:

$\frac{\partial(f+g)(s)}{\partial s} = \frac{\partial f}{\partial s} + \frac{\partial g}{\partial s}$

$(f+g)'(s) = f'(s) + g'(s)$

$\frac{\partial(fg)(s)}{\partial s} = \frac{\partial f}{\partial s} g(s) + f(s) \frac{\partial g}{\partial s}$

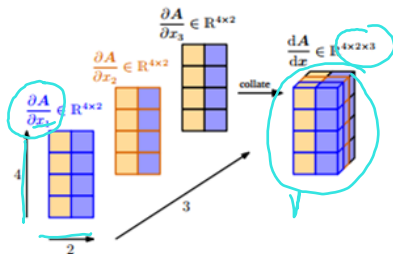
$f'(x) g(x) + f(x) g'(x)$

$\frac{\partial(f/g)(s)}{\partial s} = \left[ \frac{\partial f}{\partial s} \cdot g(s) - f(s) \cdot \frac{\partial g}{\partial s} \right] \cdot \frac{1}{(g(s))^2}$

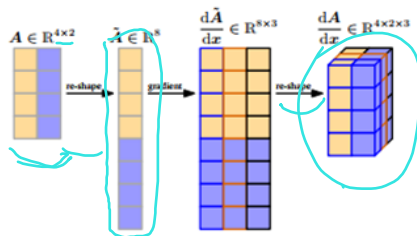
# Derivada de matrices



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  $4 \times 2$  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:

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

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

$$\frac{\partial^2 f}{(\partial x)^2} = \frac{\partial}{\partial x} \left( \frac{\partial f}{\partial x} \right)$$

Ejemplo  $f: D \subset \mathbb{R}^2 \rightarrow \mathbb{R}$   $f(x,y) = 2xy^2$

$$\frac{\partial f}{\partial x}(x,y) = 2y^2 \quad \begin{cases} \frac{\partial^2 f}{\partial x^2}(x,y) = 0 \\ \frac{\partial^2 f}{\partial y \partial x}(x,y) = 4y \end{cases}$$

$$\frac{\partial f}{\partial y}(x,y) = 4xy \quad \begin{cases} \frac{\partial^2 f}{\partial x \partial y}(x,y) = 4y \\ \frac{\partial^2 f}{\partial y^2}(x,y) = 4x \end{cases}$$

Si  $H$  es simétrica vimos que  
 $\exists S: S^T H S = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$  forma cuadrática

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 0 & 4y \\ 4y & 4x \end{bmatrix}$$

Teorema si las derivadas  
ordenadas existen y son  
continuas  $\Rightarrow$  son iguales  
**MATRIZ SIMÉTRICA**

# Diferenciación Automática

Sean, para una función  $f$ :

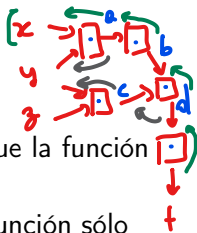
- $x_1, \dots, x_d$  las variables de entrada
- $x_{d+1}, \dots, x_{D-1}$  las variables intermedias
- $x_D$  la variable de salida
- $g_i$  funciones elementales
- $Hij(x_i)$  el conjunto de nodos hijos de cada  $x_i$

```
class Quad:  
    def f(self, x):  
        return x^2  
    def f'(self, x):  
        return 2 * x
```

```
class Prod:  
    def f(x, y):  
        return x * y  
    def f'(x, y):  
        return (y, x)
```

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:

$$\frac{df}{dx} = \frac{df}{dD} \cdot \frac{dD}{db} \cdot \frac{db}{da} \cdot \frac{da}{dx}$$
$$\frac{\partial f}{\partial x_i} = \sum_{x_j \in Hij(x_i)} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i} = \sum_{x_j \in Hij(x_i)} \frac{\partial f}{\partial g_j} \frac{\partial x_j}{\partial g_j}$$



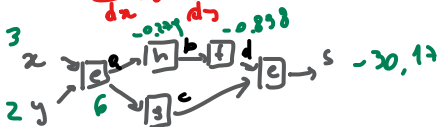
- La diferenciación automática se puede utilizar siempre que la función pueda representarse como un grafo de cómputo.
- La gran ganancia de este mecanismo está en que cada función sólo precisa saber cómo derivarse a sí misma, permitiendo OOP.



# Diferenciación Automática: ejemplo

Sean  $e(x, y) = xy$ ,  $f(x) = 3x$ ,  $g(x) = x^2$ ,  $h(x) = \sin(x)$

$$\frac{de}{dx} = y, \frac{da}{dy} = x, f' = 3, g' = 2x, h' = \cos(x)$$



$$\frac{ds}{da} = \frac{ds}{db} \cdot \frac{db}{da} + \frac{ds}{dc} \cdot \frac{dc}{da}$$

$$\frac{ds}{dx} = \frac{ds}{da} \cdot \frac{da}{dy} \cdot \frac{dy}{dx} + \frac{ds}{dc} \cdot \frac{dc}{da} \cdot \frac{da}{dx} = \frac{ds}{da} \cdot \frac{da}{dx}$$

$$\frac{dc}{dx}(d, c) = 36 \cdot 3 \cdot \cos(-0.838) \cdot 2 + (-0.838) \cdot 12 \cdot 2 = 0.961$$

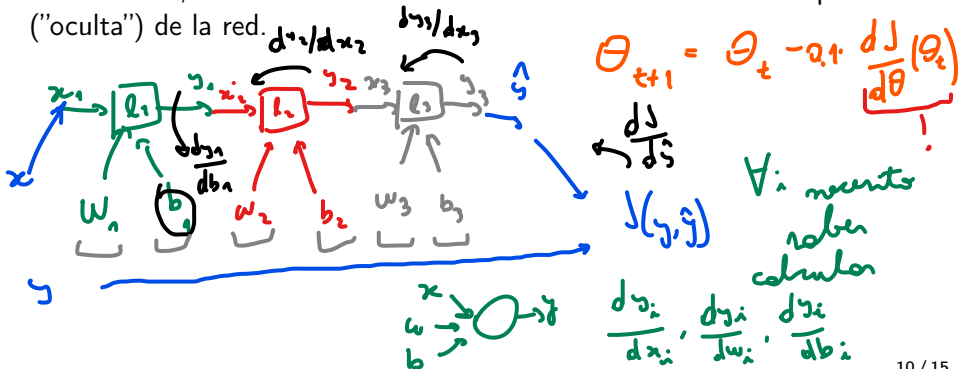
$$\frac{ds}{dy} = \frac{ds}{da} \cdot \frac{da}{db} \cdot \frac{db}{da} \cdot \frac{da}{dy} + \frac{ds}{dc} \cdot \frac{dc}{da} \cdot \frac{da}{dy} =$$

# 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:

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

*Handwritten notes:*  $\mathbb{R} \cdot \mathbb{R} + \mathbb{R} \rightarrow \text{transf. de } Ax+B$   
 $\mathbb{R} \rightarrow \mathbb{R} \rightarrow z = w \cdot x + b$

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

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

*Handwritten note:*  $\frac{dJ}{d\hat{y}} = dY$  (underlined)

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

*Handwritten note:*  $\frac{\partial \hat{y}}{\partial z} = g'(z)$  (underlined)

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

*Handwritten note:*  $\frac{\partial J}{\partial W} = dY \cdot g'(z) \cdot x$  (underlined)

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

*Handwritten note:*  $\frac{\partial J}{\partial b} = dY \cdot g'(z) \cdot 1$  (underlined)

## Redes neuronales (II)

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

$$\hat{y} = g(\underbrace{W}_{1 \times n} \cdot \underbrace{x}_{n \times 1} + \underbrace{b}_1) \quad x = (x_1, \dots, x_n)^T$$

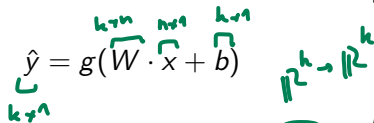
Entonces ahora para cada elemento de  $W = (w_1, \dots, w_n)$  vale lo anterior, y por tanto se puede comprobar que

$$\frac{\partial J}{\partial W} = \nabla_J(W) = \left( \underbrace{\frac{\partial J}{\partial w_1}}_1, \dots, \underbrace{\frac{\partial J}{\partial w_n}}_1 \right) = \underbrace{dY}_{1 \times 1} \cdot \underbrace{g'(z)}_1 \cdot \underbrace{x^T}_{1 \times n} \in \mathbb{R}^{1 \times n}$$
$$\frac{\partial J}{\partial b} = \underbrace{dY}_1 \cdot \underbrace{g'(z)}_1 \in \mathbb{R}^1$$

## 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:

$$\hat{y} = g(\overbrace{W}^{k \times n} \cdot \overbrace{x}^{n \times 1} + \overbrace{b}^{k \times 1})$$


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:

$$\frac{\partial J}{\partial b} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial b} = \begin{pmatrix} dY_1 \\ \vdots \\ dY_k \end{pmatrix} \odot \begin{pmatrix} g'(z_1) \\ \vdots \\ g'(z_k) \end{pmatrix} \odot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = dY \odot g'(z)$$

$(dY_1 \cdot g'(z_1) \cdot 1)$   
 $\vdots$

*element-wise*

Ahora para cada elemento de  $W$  tenemos:

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

$k \times 1$     $k \times 1$     $1 \times n$

$\downarrow$     $\downarrow$     $\downarrow$

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

$k \times n$     $k \times 1$     $1 \times n$     $k \times 1$     $1 \times n$     $k \times n$  ✓

# 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.

$$\frac{\partial J}{\partial x_i} = \sum_{j=1}^k dY_j \cdot g'(z_j) \cdot W_{j,i} = \left\langle \begin{pmatrix} dY_1 \cdot g'(z_1) \\ \vdots \\ dY_k \cdot g'(z_k) \end{pmatrix}, \begin{pmatrix} W_{1,i} \\ \vdots \\ W_{k,i} \end{pmatrix} \right\rangle = \langle dY \odot g'(z), W_{:,i} \rangle = \underline{W_{:,i}^T} \cdot (dY \odot g'(z))$$

Handwritten notes on the right side of the equation:

- $\angle x, y^T = x^1 \cdot y^1 = \dots = \sum x^i \cdot y^i$
- $x_i \rightarrow z_1, z_2, \dots, z_k$  (with arrows pointing from  $x_i$  to each  $z_j$ )
- $W_{1,i} \rightarrow z_1, W_{2,i} \rightarrow z_2, \dots, W_{k,i} \rightarrow z_k$  (with arrows pointing from each  $W_{j,i}$  to  $z_j$ )

En forma vectorizada:

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

Handwritten notes on the right side of the equation:

- $n \times k$  (under  $W^T$ )
- $k \times 1$  (under  $dY \odot g'(z)$ )
- $n \times 1$  (next to the final result)

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