

## Aprendizaje No Supervisado

Aprendizaje sin tener un objetivo

Clasifican los datos por similitud por ej.

→ En grupos.

→ Técnicas: Clustering (clusters = grupos)

## Aprendizaje Supervisado

Aprendizaje teniendo un objetivo → Predecir la  
var. respuesta

Saber si se va a comprar un libro o  
no.

→ Clasificadores: Bayesianos  
Árboles  
Redes Neuronales  
KNN

# T 3 - Aprendizaje Supervisado

Crear algoritmos capaces de predecir dado un dato, el valor de la variable respuesta

la variable que se pretende aprender

También sería la R. neuronal o crees

Métodos  
Clasificación/  
Algoritmos  
de datos

¿ KNN que es ?  
algoritmo de  
decisión es ?

Árboles de decisión : TD3 ≈ J48

Trabaja con variables categóricas → Decisión a través de Reglas  
Respuesta estricta

Clasificación Bayesiana Naive Bayes  
Silenciamiento de Laplace → a través de probabilidad  
No estricta

KNN (veano más cercano) → N individuos de datos entreno más similares a el dato que quiero probar,  
y veo que salen más si SI o NO,  
y predigo lo más común.

- ① Euclídea
- ② Manhattan

...

\*

Zero R  $\rightsquigarrow$  favorece lo más común  $\Rightarrow$  Desbalanceo de clases

- Palabras claves**
- |                         |                       |
|-------------------------|-----------------------|
| 1) Variable categóricas | 4) Validación cruzada |
| 2) Entropía / Ganancia  | 5) Sobreajuste        |
| 3) Raíz                 | 6) Poda               |

5) Sobreajuste  $\Rightarrow$  Rendimiento con datos entreno ↑↑ que el Rend. con datos prueba.

↓  
Describir a cada individuo, y perder la generalización de los datos.

2) ID3, (selecciona los atributos) en función de la entropía  $\Rightarrow$  ↓↓ Entropía  $\Rightarrow$  ↑↑ Ganancia  
 ↗ Desorden de los datos  
 crea el árbol

1) Son aquellas características (atributos) de un individuo que tiene un dominio de valores limitado

un rango

Ej: Marca coches  $\Rightarrow$  Seat  $\Rightarrow$  Renault

Si son numéricos se crean intervalos (categorías) ó  
 \* con colores  
 se discretizan

6) La poda es una técnica que consiste en eliminar nodos de nuestro árbol, y ver si mejora su rendimiento. ~ Repite esto tantas veces como Nodos haya,

Si quito un Nodo  $\Rightarrow$  elimino sus hijos

aunque si quito la Raíz  $\Rightarrow$  Se convierte en Zero R.

3) La Raíz del árbol es la variable con < entropía con el conjunto de datos iniciales, del que colgarán el resto de nodos

4) La validación cruzada es una técnica para medir el rendimiento de mi tipo de clasificador ante mi conjunto de datos

→ Consiste en dividir todo el conjunto de datos en n partes, e ir usan una proporción de partes para entrenamiento y otra para prueba en cada iteración. (n iteraciones)

\* ¿Cómo se si mi forma de clasificación es buena?

Tengo que decidir un buen sistema de clasificación, en función de como se adapte este a mis datos.

CdG: el árbol decisión  
la red neuronal

→ Haría Validación cruzada por ejemplo con KNN, J48, Red Neuronal. ⇒ El clasificado "65%" que mejor se adapta y predice sería el J48 (pa lo que lo escogí). 70% 85%

## VALIDACION CRUZADA

sistema para decidirlo

Divido mi conjunto de datos en k partes.

deg. entre  
- 10  
árboles

Uso 100 datos

para crea → 90/10 → para probar

80/20 ↘  
 $K=10$

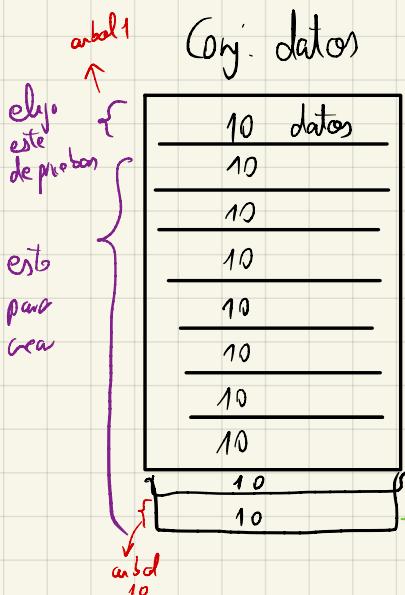
↓  
 $K=5$

conj. datos

proporción

→ 90/10, si fuera 80/20, dividiría mi conjunto de datos en 5 partes

... voy alternando después usaré otros 10 de prueba y así



este sería el rendimiento del árbol 10 usando de prueba los últimos 10 datos y para crear los 90 primeros

(Haces esto con los 10 árboles) NO  
y te quedas con el mejor. SÉ

Media del rendimiento de los 10 árboles M.

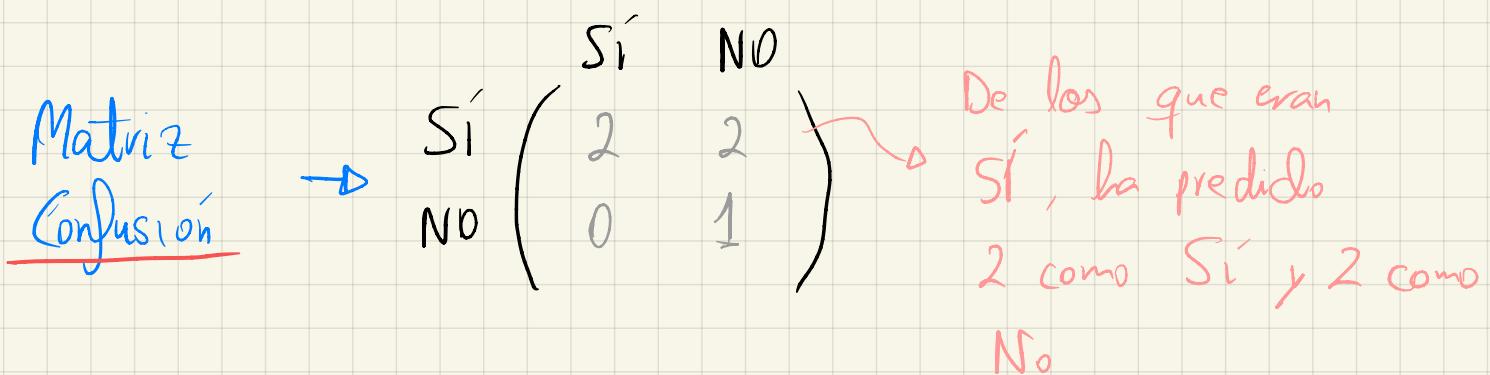
# Calcular el rendimiento de un clasificador :

→ Precision Normal

Simplemente hay que medir la proporción de aciertos en las predicciones que ha hecho mi clasificador de la van. respuesta de mi conjunto de prueba.

	Real	Predicción	
I1	Sí	No	→ X (fallo)
I2	Sí	Sí	✓ (acertó)
I3	Sí	Sí	✓
I4	No	No	✓
I5	Sí	No	X

Rendimiento de mi clasificador =  $\frac{3}{5} = 60\%$



→ Precision Balanceada

Es una medida de rendimiento mucha más justa,  
surge de luchar el desbalanceo de clases.

Interpretación de la matriz de confusión:

En la matriz de confusión se puede apreciar fácilmente la predicción del algoritmo:

De 4315 individuos que No fueron transportados, clasifica todos a que Si lo fueron → 4315 FALLOS

De 4378 ind que Si fueron transportados, clasifica a todos como que Si lo fueron → 4378 ACIERTOS

Las evalúa todas a Si.

Evidentemente en este clasificador existe un gran problema de desbalanceo de clases.  
(Aunque en el rendimiento 53% no se aprecia notablemente debido a que el conjunto de prueba está bastante equitativo). Por lo que no se ve con claridad como de torpe es este clasificador, para ver un rendimiento mucho más "justo" existe la precisión balanceada.

Que en este caso saldría un 50%, ya que tendría un 100% de efectividad para los Si y un 0% de efectividad para los No.

Más claro aquí ~

$$\begin{pmatrix} 9 & 0 \\ 1 & 0 \end{pmatrix}$$

Esta matriz tendría un 90% de acierto,

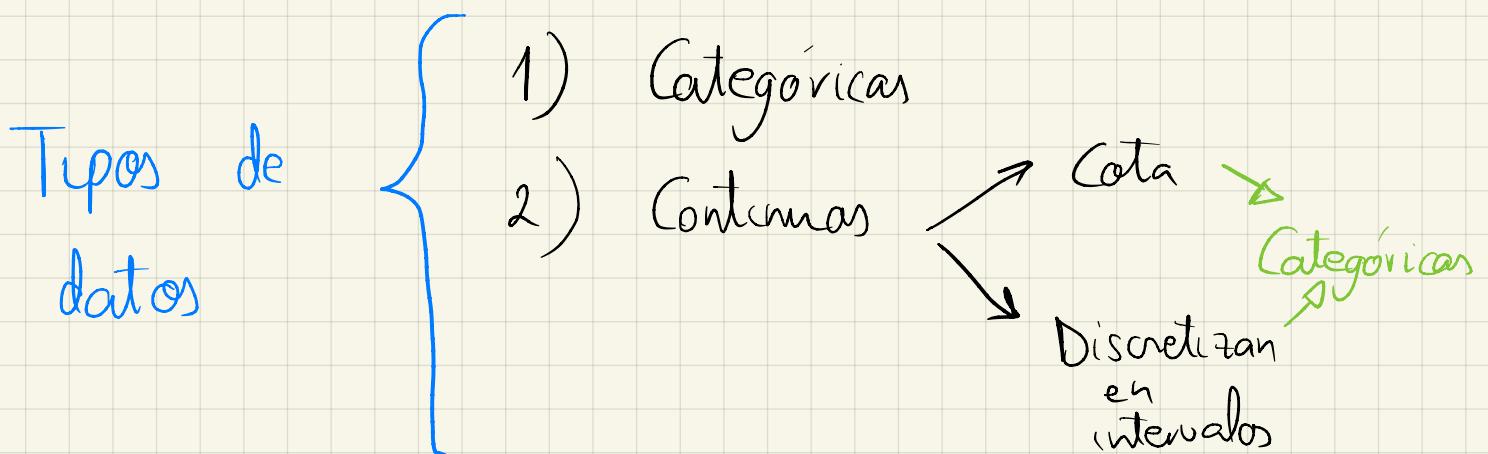
Sin embargo, hay un gran desbalanceo de clases ya que el algoritmo te predice a todo que Si ⇒ 100% en los Si y 0% de acierto en los No.

Pero como mi conjunto de datos está sesgado no más ejemplos Si que no. Tiene un 90% de acierto

Para corregirlo → Precisión balanceada

→ 0% en los No → 100% en Si para cada valor de la var  
 $\frac{0/1 + 9/9}{2} = 50\%$  (Media de las precisiones respuesta)

\* Los árboles : Trabajan con variables categóricas



- No hace falta Normalizar (Pero se suele hacer de todas formas) ya que NO participan las distancias

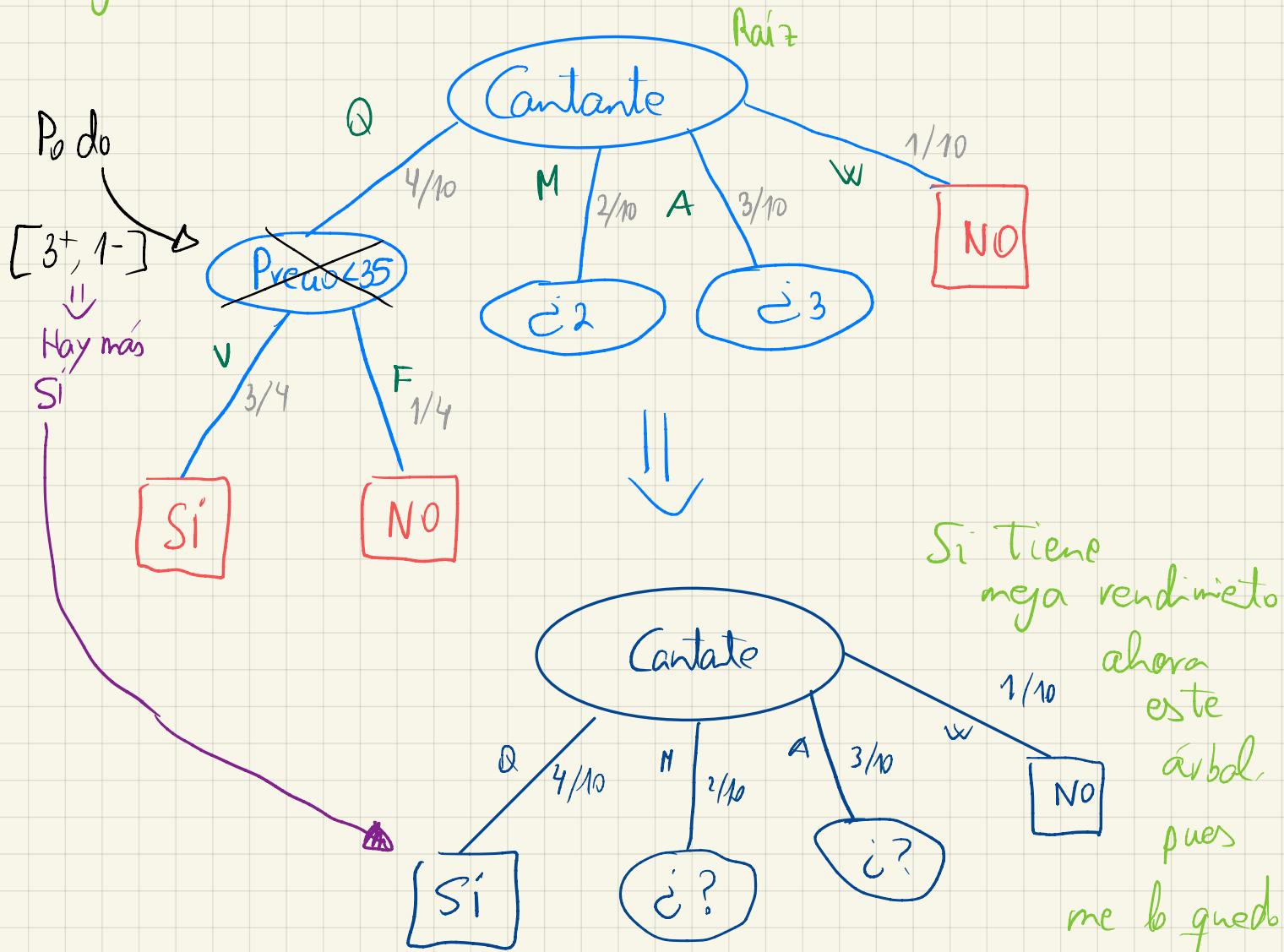
→ PODA (solo existe en los árboles)

La poda es un técnica que consiste en eliminar nodos de nuestro árbol, y ver si mejora su rendimiento. Repito esto tantas veces como Nodos haya,

Si quito un Nodo  $\Rightarrow$  elimino sus hijos aunque si quito la Raíz  $\Rightarrow$  Se convierte en Zero R.

La poda es un proceso que se le aplica a los clasificadores de tipo árbol que consiste en iterativamente ir cortando nodos y observando su rendimiento de acierto con esto. El objetivo de este proceso es simplificar el árbol, y al igual que pasa con la acotación del número mínimo de individuos en las hojas, evitar que el clasificador sobreaprenda y caiga en individualidades, perdiendo así patrones del gran volumen de los datos.

Ejemplo :



Habrá que hacerlo para todos sus nodos

Hacer de abajo a arriba.

Ejecución de árbol ↘

**Ejercicio 5.** Aplicar el algoritmo ID3 para construir un árbol de decisión consistente con los siguientes ejemplos, que nos ayude a decidir si comprar o no un CD nuevo.

Ejemplo	CANTANTE	DISCOGRÁFICA	GÉNERO	PRECIO	TIENDA	COMPRAR
$E_1$	Queen	Emi	rock	30	Mixup	sí
$E_2$	Mozart	Emi	clásico	40	Virgin	no
$E_3$	Anastacia	Corazón	soul	20	Virgin	sí
$E_4$	Queen	Sony	rock	20	Virgin	sí
$E_5$	Anastacia	Corazón	soul	30	Mixup	sí
$E_6$	Queen	Sony	rock	30	Virgin	sí
$E_7$	Wagner	Sony	clásico	30	Mixup	no
$E_8$	Anastacia	Corazón	soul	30	Virgin	no
$E_9$	Queen	Emi	rock	40	Virgin	no
$E_{10}$	Mozart	Sony	clásico	40	Mixup	sí

Considerar los siguientes ejemplos como conjunto de prueba y obtener la medida de rendimiento del árbol obtenido.

Ejemplo	CANTANTE	DISCOGRÁFICA	GÉNERO	PRECIO	TIENDA	COMPRAR
$E_{11}$	Queen	Emi	rock	30	Virgin	sí
$E_{12}$	Anastacia	Corazón	soul	20	Virgin	no
$E_{13}$	Queen	Sony	rock	20	Virgin	no
$E_{14}$	Anastacia	Corazón	soul	30	Virgin	no
$E_{15}$	Queen	Sony	rock	40	Virgin	no
$E_{16}$	Mozart	Sony	clásico	40	Mixup	sí

## Variables Categóricas

¿ Cuándo compran un disco?

$$\text{Dom}(\text{Cantante}) = \{ \text{Queen}, \text{Mozart}, \text{Ana}, \text{Wag} \}$$

$$\text{Dom}(\text{Discográfica}) = \{ \text{Emi}, \text{Corazón}, \text{Sony} \}$$

$$\text{Dom}(\text{Género}) = \{ \text{Rock}, \text{Clásico}, \text{Soul} \}$$

$$\text{Dom}(\text{Tienda}) = \{ V, M \}$$

Variable Contínua // Aunque también podríamos tratarla como categórica con los valores que aparecen

$$\text{Dom}(\text{Precio}) = [0, \infty) \rightarrow \text{Pasar a Categórica}$$

## Variable respuesta / objetivo

$$\text{Dom( Compran)} = \{ \text{Sí}, \text{No} \}$$

### Aplicar ID3

Raíz? Calculo entropía

Lo miro en la tabla?

↓  
3 Sí, 1 No

\* Cantante

$$\left\{ \begin{array}{l} Q \rightarrow 4/10 \cdot [3^+, 1^-] = 0'4 \cdot 0'811 \\ + \\ M \rightarrow 2/10 \cdot [1^+, 1^-] = 0'2 \cdot 1 \\ + \\ A \rightarrow 3/10 \cdot [2^+, 1^-] = 0'3 \cdot 0'918 \\ + \\ W \rightarrow 1/10 \cdot [0^+, 1^-] = 0'1 \cdot 0 \end{array} \right. +$$

Entropía → 0'7998

\* Discográfica

$$\left\{ \begin{array}{l} Emi \rightarrow 3/10 \cdot [1^+, 2^-] = 0'918 \cdot 0'3 \\ Cova \rightarrow 3/10 \cdot [2^+, 1^-] = 0'918 \cdot 0'3 \\ Sony \rightarrow 4/10 \cdot [3^+, 1^-] = 0'811 \cdot 0'4 \end{array} \right. +$$

0'8752

\* Género

$$\left\{ \begin{array}{l} Rock \rightarrow 4/10 \cdot [3^+, 1^-] = 0'811 \cdot 0'4 \\ Clasic \rightarrow 3/10 \cdot [1^+, 2^-] = 0'918 \cdot 0'3 \\ Soul \rightarrow 3/10 \cdot [2^+, 1^-] = 0'918 \cdot 0'3 \end{array} \right. +$$

0'8752

\* Tienda

$$\left\{ \begin{array}{l} M \rightarrow 4/10 \cdot [3^+, 1^-] = 0'811 \cdot 0'4 \\ V \rightarrow 6/10 \cdot [3^+, 3^-] = 1 \cdot 0'6 \end{array} \right. +$$

0'9244

\* Precio → Pasar a Categóricas elegir Cota

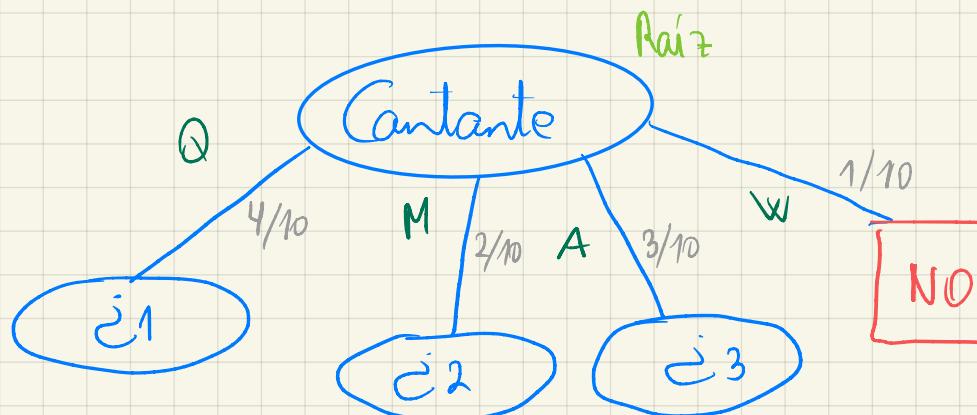
	20	30	40
Cotas	E 3/4 Sí/NO ↓ 25	1/5 / 6/7/8 Sí/Sí/Sí/No/No ↓	2/9/10 No/No/81 ↑ 35 Cambia de No a Sí

$$\left\{ \begin{array}{l} < 25 \\ < 35 \end{array} \right. \left\{ \begin{array}{l} V \rightarrow 2/10 \\ F \rightarrow 8/10 \end{array} \right. \cdot \begin{bmatrix} 1^+, 1^- \end{bmatrix} = \frac{0.2 \cdot 1}{1}$$

$$\left. \begin{array}{l} V \rightarrow 7/10 \\ F \rightarrow 3/10 \end{array} \right. \cdot \begin{bmatrix} 4^+, 3^- \end{bmatrix} = \frac{0.985 \cdot 0.7}{0.9649}$$

$$\left. \begin{array}{l} V \rightarrow 3/10 \\ F \rightarrow 7/10 \end{array} \right. \cdot \begin{bmatrix} 1^+, 2^- \end{bmatrix} = \frac{0.918 \cdot 0.3}{0.9649}$$

→ Raíz será la menor Entropía → Cantante.



↓ 1 → Volver a calcular entropías pero esta vez solo de los  $\frac{4}{10}$  que son "Queon"

\* Discográfica

$$\left\{ \begin{array}{l} E \rightarrow 2/4 \cdot [1^+, 1^-] = 0'5 \cdot 1 \\ C \rightarrow \\ S \rightarrow 2/4 \cdot [2^+, 0^-] = 0'5 \cdot 0 \end{array} \right. + \underline{\underline{0'5}}$$

\* Género

$$\left\{ R \rightarrow 4/4 \cdot [3^+, 1^-] = 0'811 \cdot 1 = 0'811 \right.$$

\* Tienda

$$\left\{ \begin{array}{l} M \rightarrow 1/4 \cdot [1^+, 0^-] = 0'25 \cdot 0 \\ V \rightarrow 3/4 \cdot [2^+, 1^-] = \underline{\underline{0'918 \cdot 0'75}} + 0'685 \end{array} \right.$$

\* Precio

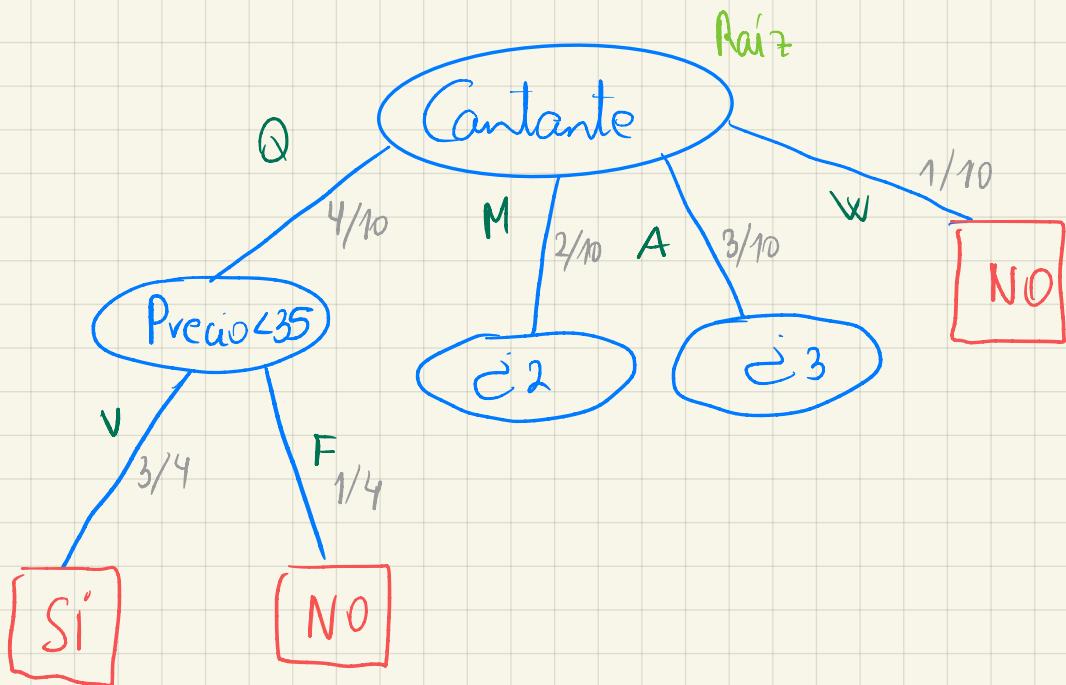
20      30      40

E4	E1,6	E9
SÍ	SÍ	NO

$$\left\{ < 35 \rightarrow \left\{ \begin{array}{l} V \rightarrow 3/4 \cdot [3^+, 0^-] = 0 \\ F \rightarrow 1/4 \cdot [0, 1^-] = 0 \end{array} \right. \right. \underline{\underline{0}}$$

~~12~~

$\hat{c}1 \rightsquigarrow$  Precio < 35  
(ota)



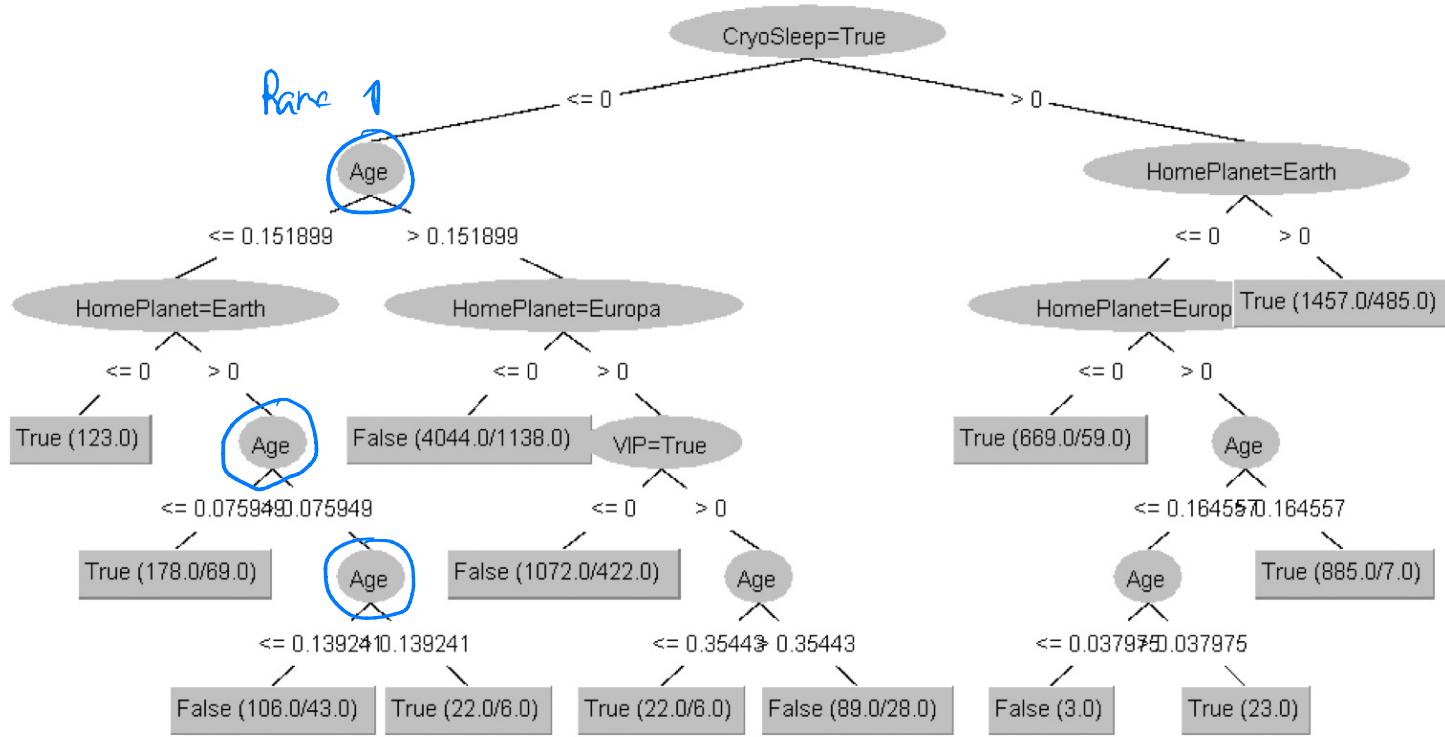
Ahora habría que calcular  $c_2$ , con las 2/10 de Montant, y probar todas las variaciones posibles cantante.

• • •

Terminar

! Si al final no es exacto  $\{[4^+, 3^-]\}$  → habrá  $\boxed{Sí} \rightarrow$  para la mayoría  
En caso empate, según el según alguno primo  $\rightarrow$   $[4^+, 4^-]$

# \* Atención



Normalmente cuando en una Rama ya existe un NODO de una variable, ya se quita para elegir sus hijos. Sin embargo, cuando resolvemos variables continuas (numéricas) por cotas, NO pasa eso. Si no que sigue siendo una variable candidata para ser Nodo hijo.

Si la discretizasenos No pasaría esto

# \* Clasificación Bayesiana

$\frac{2}{5}$

\* Lo que hemos visto en prácticas :

EJ.	CIELO	TEMPERATURA	HUMEDAD	VIENTO	JUGAR TENIS
$D_1$	SOLEADO	ALTA	ALTA	DÉBIL	-
$D_2$	SOLEADO	ALTA	ALTA	FUERTE	-
$D_3$	NUBLADO	ALTA	ALTA	DÉBIL	+
$D_4$	LLUVIA	SUAVE	ALTA	DÉBIL	+
$D_5$	LLUVIA	BAJA	NORMAL	DÉBIL	+
$D_6$	LLUVIA	BAJA	NORMAL	FUERTE	-
$D_7$	NUBLADO	BAJA	NORMAL	FUERTE	+
$D_8$	SOLEADO	SUAVE	ALTA	DÉBIL	-
$D_9$	SOLEADO	BAJA	NORMAL	DÉBIL	+
$D_{10}$	LLUVIA	SUAVE	NORMAL	DÉBIL	+
$D_{11}$	SOLEADO	SUAVE	NORMAL	FUERTE	+
$D_{12}$	NUBLADO	SUAVE	ALTA	FUERTE	+
$D_{13}$	NUBLADO	ALTA	NORMAL	DÉBIL	+
$D_{14}$	LLUVIA	SUAVE	ALTA	FUERTE	-

→ Ejemplo de Clasificador: Naive Bayes

- ▶ Supongamos que queremos predecir si un día soleado, de temperatura suave, humedad alta y viento fuerte es bueno para jugar al tenis
- ▶ Así que necesitamos estimar todas estas probabilidades, lo que hacemos simplemente calculando frecuencias en la tabla anterior:

→  $p(+)$  =  $9/14$ ,  $p(-)$  =  $5/14$ ,  $p(\text{soleado}|+)$  =  $2/9$ ,  
 $p(\text{soleado}|-) = 3/5$ ,  $p(\text{suave}|+) = 4/9$ ,  $p(\text{suave}|-) = 2/5$ ,  
 $p(\text{alta}|+) = 3/9$ ,  $p(\text{alta}|-) = 4/5$ ,  $p(\text{fuerte}|+) = 3/9$  y  
 $p(\text{fuerte}|-) = 3/5$

3 de los 5 días malo hacia viento fuerte

$p(+)$  → prob. de Jugar

$p(\text{alta}|+)$  ~ prob. de que en un día que juegue, la humedad sea alta

Predigo que NO es un día bueno

- Por tanto, las dos probabilidades a posteriori son:

$$\textcircled{+} \rightarrow P(+)P(\text{soleado}|+)P(\text{suave}|+)P(\text{alta}|+)P(\text{fuerte}|+) = \alpha 0.007$$
$$\textcircled{-} \rightarrow P(-)P(\text{soleado}|-)P(\text{suave}|-)P(\text{alta}|-)P(\text{fuerte}|-) = \alpha 0.041$$

→ Normalizado

- Si queremos el valor de probabilidad asociado, normalizamos (sumamos los valores anteriores, y dividimos cada uno por la suma). Así, quedaría:

$$P(+|\text{soleado, suave, alta, fuerte}) = 0.14637 \rightarrow 14\% \quad \frac{\frac{0.007}{0.048}}{0.007}$$
$$P(-|\text{soleado, suave, alta, fuerte}) = 0.85363 \rightarrow 85\% \quad \frac{\frac{0.041}{0.048}}{0.041}$$

## → Suavizado de Laplace

Por problemas en el conj. entrenamiento (escasez de datos) pasa que por ejemplo no hay ningún dato en el que en un día bueno haya habido viento fuerte  $\Rightarrow$

$$P(\text{fuerte}/+) = 0/9 = 0 \Rightarrow p(+) \cdot p(\text{soleado}|+) \dots = 0$$

Pero eso es Inreal, que no haya o que haya pocos ejemplos haría que todos los demás variables no contaran.

Para evitarlo se suele hacer el suavizado de Laplace:

En el caso malo

$$P(\text{SOLEADO} | +) = \frac{0}{9} = \frac{0+1}{9+3} = \frac{1}{12}$$

Caso Real con mis datos (con

$$P(\text{SOLEADO} | +) = \frac{2+k}{9+k \cdot 3} = \frac{3}{12}$$

3 pK  
hay 3 valores  
posibles de  
ese atributo  
{soleado,  
nublado,  
lluvioso}

Caso Real Sin Suavizado

$$P(\text{SUAVE} | -) = \frac{2}{5}$$

$$k=1$$

$$P(\text{SUAVE} | -) = \frac{2+k}{5+3k}$$

Caso Real  
con

k bajas

k altas  $\rightarrow$  suavizar extremos

•

# Clasificador: El vecino

- Arb
- - Bayesianos
- El vecino más cercano

mas cercano → visto en prácticas

---

KNN ~ N vecinos mas cercanos

(Elevan alguna variable a un rango de valores (0,1))  
Edades [0 - 80] Edad = 40 años  
[0 - 1] ~ 0.5

↳ 3 tipos de distancias

mas típicas. valor  
[0, 1] pasar a

Normaliza

KNN es un tipo de clasificador que para predecir la variable objetivo de un individuo prueba se basa en las semejanzas (a través de distancias) con K individuos de entrenamientos y sus variables respuesta.

Para aplicar este algoritmo es muy importante que los datos estén normalizados → esto consigue que en las distancias entre los individuos no se ponderen algunas variables por encima de otras, sino que todas tengan un peso equitativo en la decisión.

Hay que tener en cuenta que el valor de k es crucial para el buen funcionamiento del algoritmo. Un valor de K alto te proporcionará una mayor seguridad en tu predicción ya que será más difícil que caigas en individualidades, mientras que si este K es demasiado alto, pecarás de que las distancias de los individuos seleccionados para la decisión serán mayores => menos certeza (datos más diferentes)

También podemos destacar que lo ideal es que k sea un valor impar, y así evitar un posible empate.

Por ejemplo si fuera k=5, y las variables respuesta de los 5 más semejantes fueran SI, NO, NO, SI y SI. KNN predeciría el individuo prueba como SI.

\* Hace falta normalizar para que todas las distancias entre  $\neq$  tipos de variables ponderen lo mismo.  
→ Sistema de similitud JUSTO.

## ○ Distancia entre individuos

► Distancias más usadas en la práctica:

► Euclídea:  $d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$

► Manhattan:  $d_m(x, y) = \sum_{i=1}^n |x_i - y_i|$

► Hamming: número de componentes en las que se difiere. variable

→ ► La euclídea se usa cuando cada dimensión mide propiedades similares y la Manhattan en caso contrario; la distancia Hamming se puede usar aún cuando los vectores no sean numéricos.

(► Normalización: cuando *no* todas las dimensiones son del mismo orden de magnitud, se normalizan las componentes (restando la media y dividiendo por la desviación típica))

→ (► Estas distancias también son útiles para el algoritmo k-medias que se aplica como técnica de *clustering*. )

## Como tratan los datos para KNN

1) Variables categoricas

Nominales →  
Ej: {Rojo, Verde}  
con + valores se crean columnas

Binizar  
One Hot Encoding  
Rojo = 1  
Verde = 0

Nominales son las que la diferencia entre los valores sea únicamente que sean distintos.

Ordinal, la diferencia depende del valor → Mayor diferencia entre 1º p y 8º p que entre 3º p y 8º p

Ordinales → Asocia a valor numérico según el orden  
Ej: {1-planta, 3-plata, 8-p}

1-planta = 1      8-p = 8  
3-plata = 3

2) Normalizar

**Ejercicio 10.** El siguiente conjunto de entrenamiento muestra los valores de los atributos AT1 y AT2 y la clasificación de 8 pacientes. Los atributos AT1 y AT2 pueden tomar cualquier valor real. Los datos de clasificación pueden ser 0 o 1.

Boletín  
5

Paciente	AT1	AT2	Cl
P1	0.5	0.1	1
P2	0.8	0.5	1
P3	0.6	0.0	1
P4	0.8	0.2	1
P5	0.3	0.3	1
P6	0.1	0.9	0
P7	0.2	0.7	0
P8	0.9	0.9	0

Clasifica la instancia (0.5,0.5) utilizando el conjunto de entrenamiento anterior con el método k-NN con  $k = 3$  y la distancia Manhattan.

$$d = \sqrt{(0.5 - 0.5)^2 + (0.5 - 0.5)^2} \rightarrow \text{Individuo.}$$

Busco los 3 ( $k$ ) más pareados  $\Rightarrow$  Ver las distancias  $\rightarrow$  Manhattan

$$D(d, p_i) = \left\{ \begin{array}{l} 1) |0.5 - 0.5| + |0.5 - 0.1| = 0.4 \leftarrow 1 \\ 2) 0.3 \leftarrow 1 \\ 3) 0.1 + 0.5 = 0.6 \\ 4) 0.3 + 0.3 = 0.6 \leftarrow 1 \\ 5) 0.4 \leftarrow 1 \\ 6) 2 \cdot 0.4 = 0.8 \\ 7) 0.3 + 0.2 = 0.5 \\ 8) 2 \cdot 0.4 = 0.8 \end{array} \right.$$

Explico  
al negro      Clasif  
variable  
respuesta  
indivi. 2      d  $\Rightarrow$  1

Predicció  
n de KNN

T4 \_ Aprendizaje

NO Supervisado

---

# Clustering

~> Agrupar datos en grupos :)

## → Clustering jerárquico

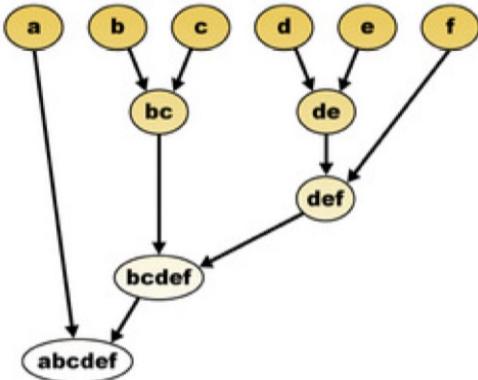
- ▶ Como hemos visto, el clustering de **partición estricta** divide al conjunto de datos en **clusters** que no tienen **ninguna estructura interna**.
- ▶ En cambio, el **clustering jerárquico** da **estructura interna a los clusters**. De hecho, de manera recursiva, *cada cluster* está *dividido en clusters internos*, en una estructura anidada que va *desde un cluster general conteniendo a todos los individuos, hasta clusters que contienen un único elemento*.

## Algunos de construcción de clusters

Dependiendo del orden en que se cree la jerarquía de clusters, los algoritmos de clustering jerárquico se dividen en dos tipos:

- ▶ **Clustering jerárquico aglomerativo**: Partimos **de clusters** conteniendo un **único ejemplo** y vamos agrupando los clusters, obteniendo **agrupamientos** cada vez de **mayor tamaño hasta** obtener un **cluster** final con **todos** los individuos.
- ▶ **Clustering jerárquico divisor**: **Empezamos con un cluster** con **todos** los individuos y vamos realizando **particiones** de los clusters obtenidos **hasta** obtener **clusters** conteniendo un **único ejemplo**.

Con independencia de que se use clustering aglomerativo o divisor, los resultados suelen representarse con una estructura de árbol llamada *dendrograma*:

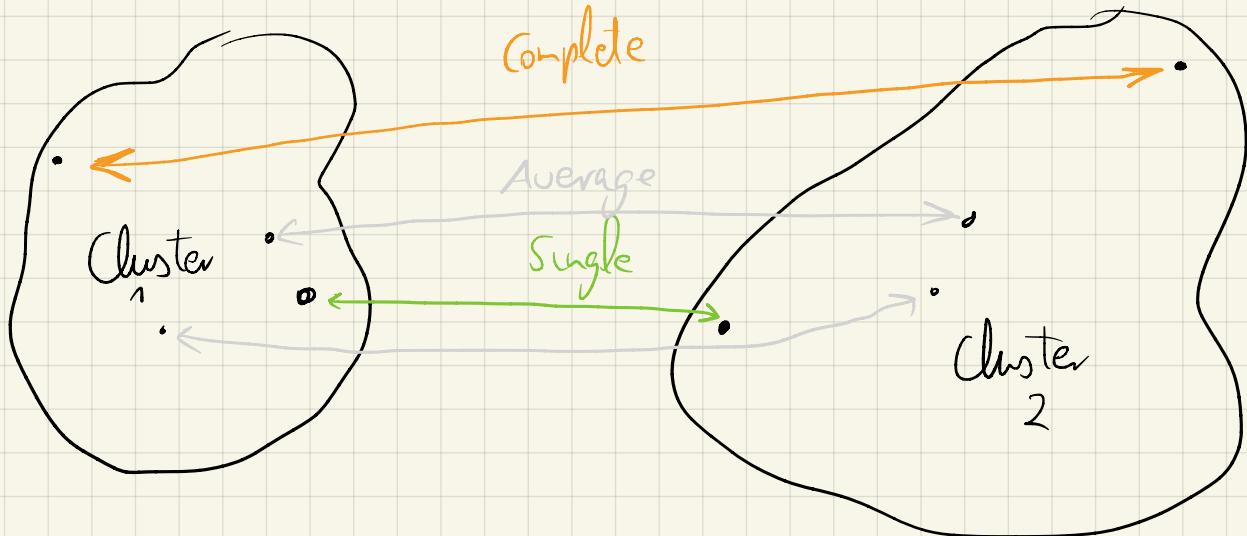


Ej.: de  
clustering  
aglomerativo

La raíz representa el conjunto completo y cada una de las hojas representa una instancia.

## o Distancia entre grupos (Clusters) ↳ de individuos

- **Single linkage:** La distancia entre dos clusters es la **distancia entre los objetos más cercanos** de los diferentes clusters. Dos clusters pueden estar conectados debido al *ruido*. No obstante, funciona bien si los clusters están suficientemente separados.
- **Complete linkage:** La distancia entre dos clusters es la **distancia entre los objetos más lejanos** de los diferentes clusters. Es efectiva cuando los clusters son pequeños y compactos.
- **Group average linkage algorithm:** La distancia entre dos clusters es la **distancia media de todos los pares de puntos procedentes de diferentes clusters**. También se conoce como *Unweighted pair group method average (UPGMA)*



**Ejercicio 11.** Consideremos el siguiente conjunto,  $D$ , de ejemplos para un problema de clasificación:

	$A_1$	$A_2$	$A_3$	$A_4$
$e_1$	5'1	3'5	1'4	0'2
$e_2$	4'9	3'0	1'4	0'2
$e_3$	7'0	3'2	4'7	1'4
$e_4$	6'4	3'2	4'5	1'5
$e_5$	6'3	3'3	6'0	2'5
$e_6$	5'8	2'7	5'1	1'9

Se pretende calcular, mediante el uso del algoritmo de clustering jerárquico aglomerativo, utilizando single linkage para calcular la distancia entre clusters, un dendrograma que responda al conjunto. Para ello se ha calculado la distancia (usando la distancia Manhattan) entre cada par de ejemplos de  $D$ .

	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$
$e_1$	0'7	6'7	6'0	8'3	6'9
$e_2$		6'8	6'1	8'6	6'6
$e_3$			0'9	3'2	2'6
$e_4$				2'7	2'1
$e_5$					2'6

$\nearrow$  ind. más cercanos

Diferencias manhattan entre los ind. (distancias indire)

## Clustering jerárquico aglomerativo

→ Estrategia de la distancia mínima o similitud máxima  
(amalgamiento simple: single linkage)

Un **algoritmo de construcción**:

Entrada: un conjunto de clusters de ejemplos (cada cluster contiene un único ejemplo) y una matriz inicial de distancias entre pares de ejemplos.

1. Se eligen los dos clusters **más cercanos** y se considera una nueva familia en la que ese par es sustituido por un cluster que contiene a ambos.
2. Se actualiza la matriz de distancias calculando los valores para cada par de clusters en la nueva familia (en cada paso, el número total de clusters disminuye en una unidad).
3. Se vuelve al paso 1 hasta que exista un único cluster

Es la que  
hemos  
usado  
en el  
ejercicio  
(hay más)

Inicial  $\rightarrow$  6 pambles subgrupos (un ind por grupo) Dendrograma

1) Más cercanos  $\rightarrow$  e1, e2 ( $d=0^{\circ}7$ )

	e3	e4	e5	e6	
{e1, e2}	6'7	6'0	8'3	6'6	cojo el más junto
e3		0'9	3'2	2'6	
e4			2'7	2'1	
e5				2'6	

Distancia Simple

	{e3, e4}	e5	e6
{e1, e2}	6'0	8'3	6'6
{e3, e4}	2'7	2'1	
e5		2'6	

	{e6, {e3, e4}}	e5
{e1, e2}	6'0	8'3
{e6, {e3, e4}}	2'6	

$\{e6, \{e3, e4\}\}, e5\}$

$\{e1, e2\}$

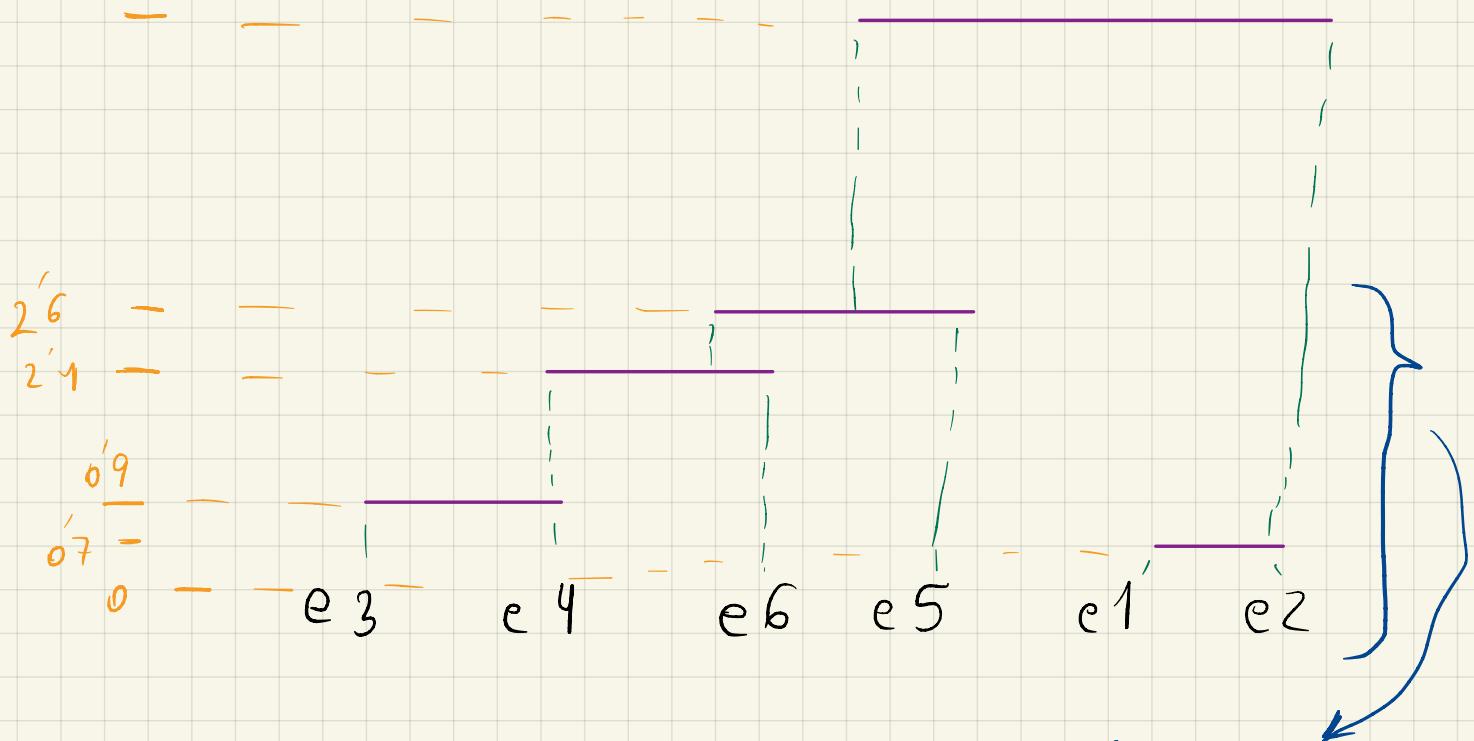
60

$\{e6, \dots, e5\}$

No hay una decisión buena

Dendrogramma ~

Sirve para saber bien con que grupos te quedas



aloneja  
te que das aquí  
y no más  
 $\{e1, e2\}$  con los  
otros pk están  
muy dispersos

# Ejemplo de diapos

## Estrategia de la distancia mínima o similitud máxima

Se parte de una población de 7 individuos.

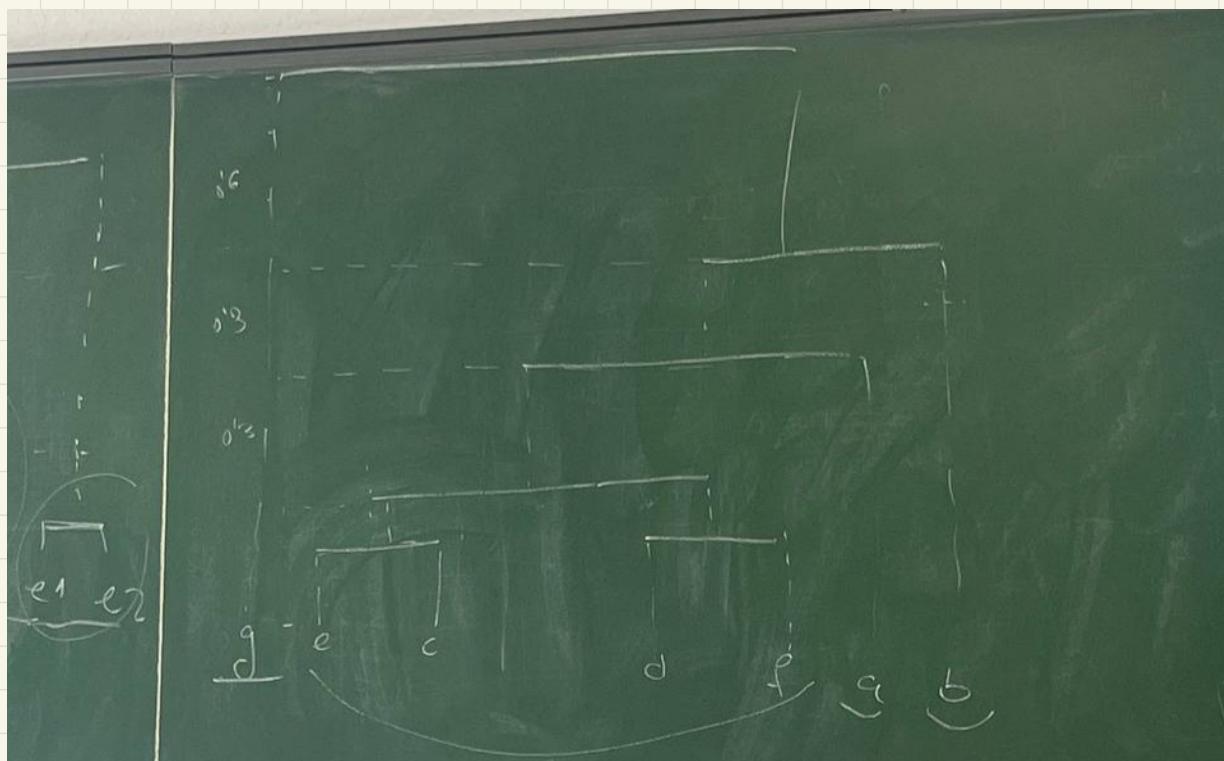
Clusters iniciales:  $\Delta_0 = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}\}$

Matriz de distancias iniciales de la población es:

	a	b	c	d	e	f	g
a	0						
b	2.15	0					
c	0.7	1.53	0				
d	1.07	1.14	0.43	0			
e	0.85	1.38	0.21	0.29	0		
f	1.16	1.01	0.55	0.22	0.41	0	
g	1.56	2.83	1.86	2.04	2.02	2.05	0

0 0 0

## Dendrograma



K-medidas

Algoritmo

as N-grupos q quieras obtener

Algoritmo



Ejercicio 3. Considera los puntos  $P_1 = (-3, 24)$ ,  $P_2 = (12, 40)$ ,  $P_3 = (4, 4)$  y  $P_4 = (12, -4)$ ; y los centros  $m_1 = (-3, 4)$  y  $m_2 = (12, 4)$ . Se pide aplicar el algoritmo de k-medias sobre los puntos  $P_1, \dots, P_4$  tomando  $m_1$  y  $m_2$  como centros iniciales hasta la primera modificación de los centros. Usar la distanza euclídea.

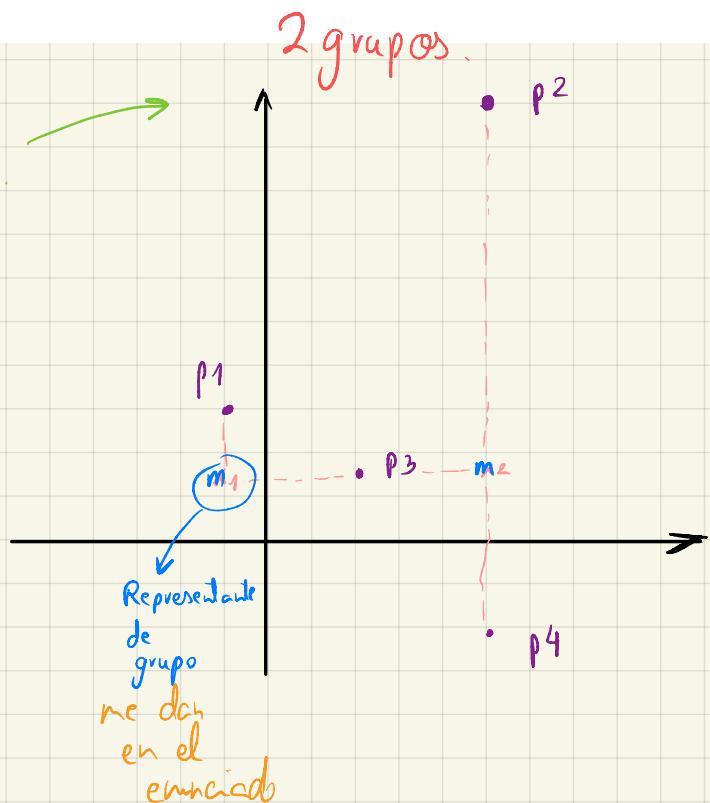
$$P_1 (-3, 24) \rightarrow \begin{cases} d(p_1, m_1) = 20 \\ d(p_1, m_2) = 35 \end{cases}$$

Puesto  
a ojo.

$$P_2 (12, 40) \rightarrow \begin{cases} d(p_2, m_1) \\ d(p_2, m_2) \end{cases} \leftarrow$$

$$P_3 (4, 4)$$

$$P_4 (12, -4)$$



representantes



$P_1$ ,  
 $P_2$ ,  
 $P_3$ ,  
 $P_4$

$M_1$   
 $M_2$

Gpo 1 recala  $M'_1$   
media  $M'_2$

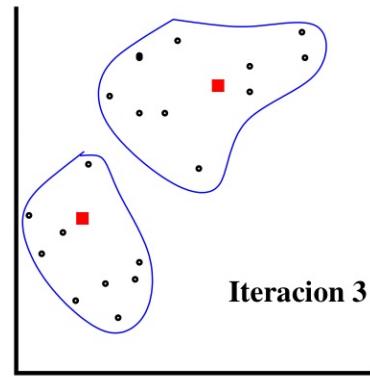
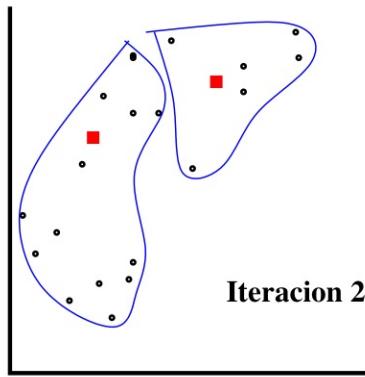
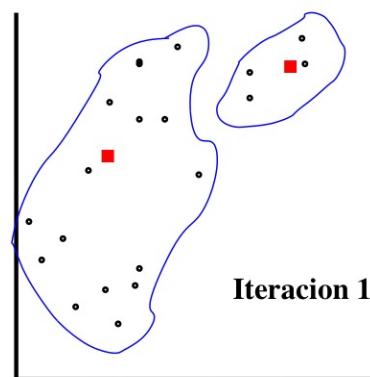
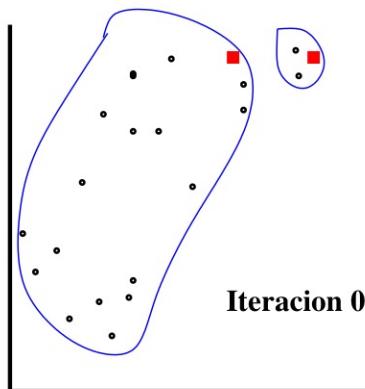
$P_1$   
 $P_2$   
 $P_3$   
 $P_4$

$M'_1$   
 $M'_2$

hasta que no cambie los grupos

Ahora habrá que recalcular los representantes con los nuevos grupos (haciendo las medias de los ptos), y con ellos volver a ver en qué grupo meter los ptos ... hasta que no cambien los grupos

### Idea gráfica intuitiva en el algoritmo de k-medias



# T5\_ Redes neuronales

---

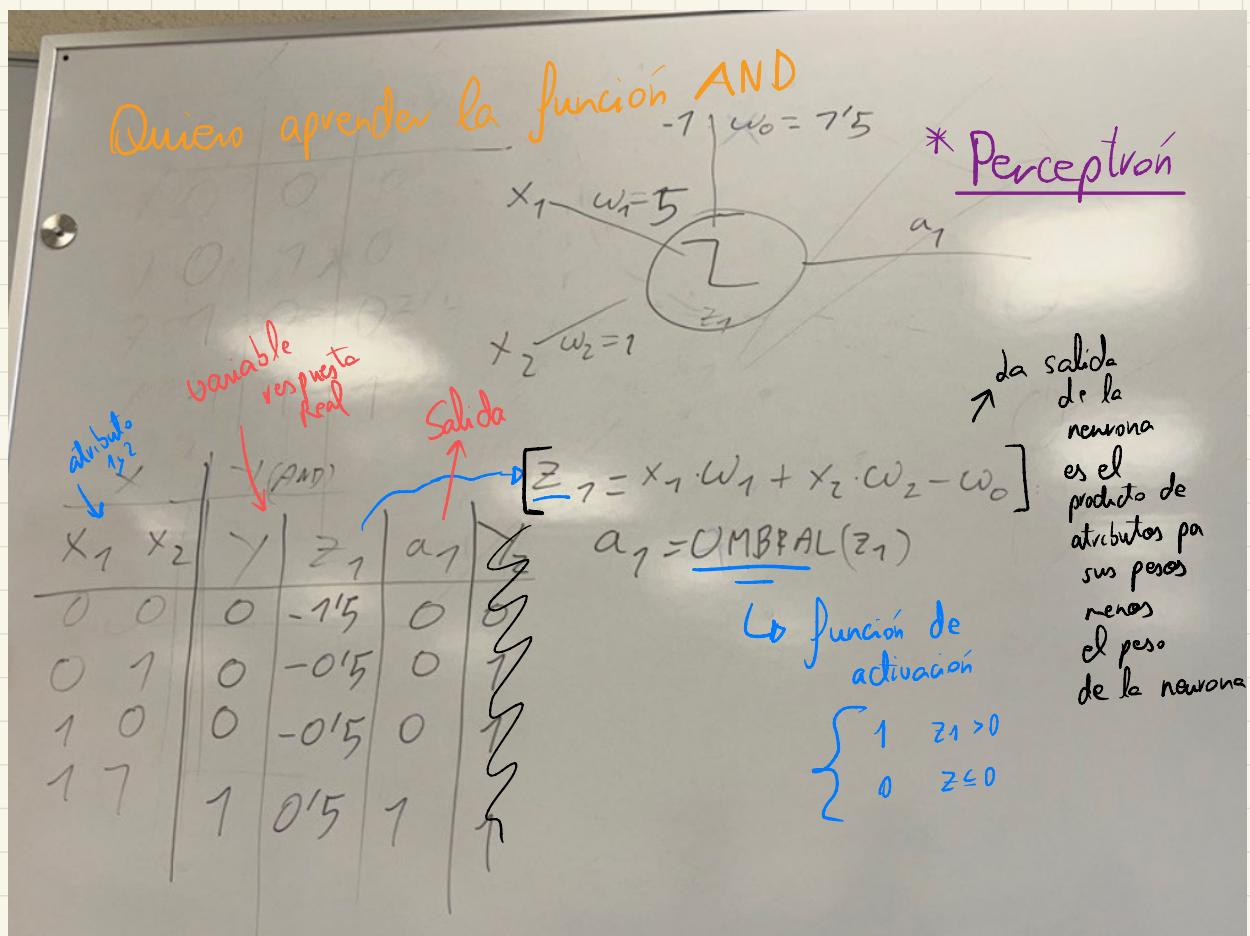
Clasificador  
Supervisado

# Redes Neuronales

→ Visto en Prácticas

Deep Learning → Parte del aprendizaje automático centrada en redes neuronales complejas

Perceptrón → Red de una sola neurona



# Quiero aprender la función XOR

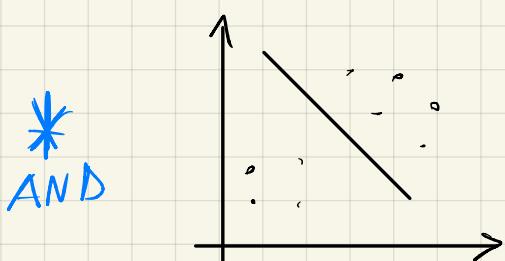
		XOR	$z_1$	$z_2$	$a_3$	
$x_1$	$x_2$	$y$	0	-1	-1	0
0	0	0	0	0	-1	0
1	0	1	0	0	-1	0
0	1	1	0	0	-1	0
1	1	0	1	1	1	

//

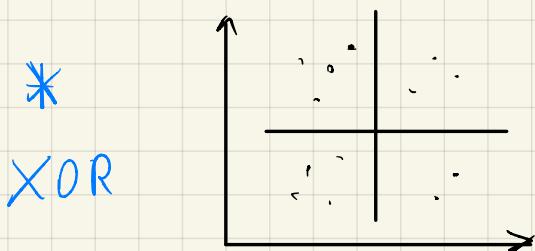
$\Rightarrow$  la predicción es mala, habría que corregir

Multicapa

		$-1 \text{ (AND)}$	$z_1$	$a_1$	$Y_2$	$a_2$	$a_2 = \text{OMBRAL}(z_1)$	
$x_1$	$x_2$	$y$	0	-1'5	0	0	0	
0	0	0	0	-0'5	0	1	0	
0	1	0	0	-0'5	0	1	0	
1	0	0	0	-0'5	0	1	0	
1	1	1	1	0'5	1	1	1	



$\Rightarrow$  Si mis datos son linealmente separables  $\Rightarrow$  1 neurona



$\Rightarrow$  2 neuronas

El proceso de aprendizaje será ir cambiando todos los pesos anteriores: Hasta reducir el error al minimo

→ Empiezo desde la ultima capa (salida)

Algoritmo de retro propagación

La ultima Capa

→ Corregir peso neuronas

Error neurona 7

$$\Delta_7 = (y_1 - \text{out}_1) \cdot g'(in_7)$$

Puedo cambiar pesos

$w_7$  y  $w_8$



$$\Delta_8 = (y_2 - \text{out}_2) \cdot g'(in_8)$$

$$w_7 \leftarrow w_7^{\text{actual}} - \eta \Delta_7$$

$$w_8 \leftarrow w_8^{\text{actual}} - \eta \Delta_8$$

Las capas intermedias → Corregir peso neuronas

$$\Delta_5 = g'(in_5) \cdot (w_5^7 \cdot \Delta_7 + w_5^8 \cdot \Delta_8)$$

el peso de la 5a la 8a lo que se le equivoque

$$\Delta_6 = g'(in_6) \cdot (w_6^7 \cdot \Delta_7 + w_6^8 \cdot \Delta_8)$$

$$w_5 \leftarrow w_5^{\text{actual}} - \eta \Delta_5 // w_6 \leftarrow w_6^{\text{actual}} - \eta \Delta_6$$

## Corregir los pesos de las conexiones

$$W_5^7 \leftarrow W_5^7 + \eta \Delta_5 \cdot a_5$$

$$W_6^7 \leftarrow W_6^7 + \eta \Delta_7 \cdot \underline{a_6}$$

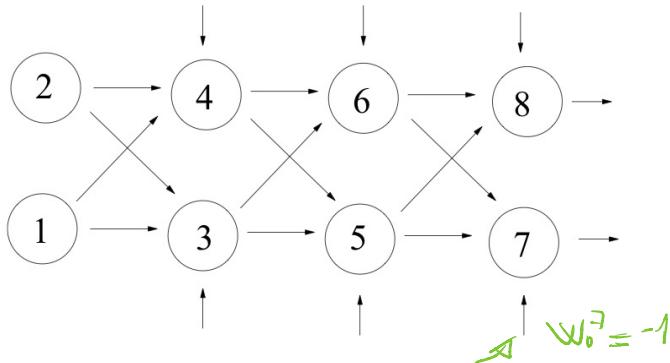
lo que  
le llega  
a la neurona 7

\* Función de activación → más común

puede ser una función lineal → debe ser DERIVABLE

$$g(z) = \frac{1}{1 - e^{-z}} \quad (Sigmoid)$$
$$z = \text{in}^d$$
$$g'(z) = g(z) \cdot (1 - g(z)) \quad \text{out}^d$$
$$g(z) = A + Bz \quad g'(z) = B$$

Ejercicio 3. Consideremos la siguiente red neuronal



Representaremos por  $w_i^j$  el peso asociado a la sinapsis desde la neurona  $i$  a la neurona  $j$  y por  $w_0^j$  el peso asociado al sesgo ( $a_0 = -1$ ) de la neurona  $j$ . Consideraremos como factor de aprendizaje  $\eta = 0.1$  y la función sigmoide como función de activación para todas las capas. Consideraremos que todos los pesos iniciales de la red neuronal valen 1. ¿Cuánto vale el peso  $w_0^6$  tras su actualización actualización, al utilizar el algoritmo de retropropagación para el ejemplo  $((1, 1), (1, 1))$ ? Detallar todos los pasos necesarios del algoritmo para obtener el valor pedido. [Nota: Para hacer los cálculos tomaremos los cuatro primeros decimales.]

$$\eta = 0.1$$

$$\begin{pmatrix} x_1 & x_2 \\ 1 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} y_1 & y_2 \\ 1 & 1 \end{pmatrix}$$

Entradas

Salidas Reales

$$w_{i,j} = 1$$

Capa 1

↓ salida primera neurona

$$a_1 = 1$$

$$a_2 = 1$$

lo que le llega a la 3

Capa 2 ?

$$in_3 = -w_0^3 + a_1 \cdot w_1^3 + a_2 \cdot w_2^3$$

↑ peso de la neurona 3

↑ la salida de la 1 · el peso de la entrada

↑ lo que llega de la 2

(lo que llega de la 1)

$$in_3 = 1$$

$$in_4 = 1$$

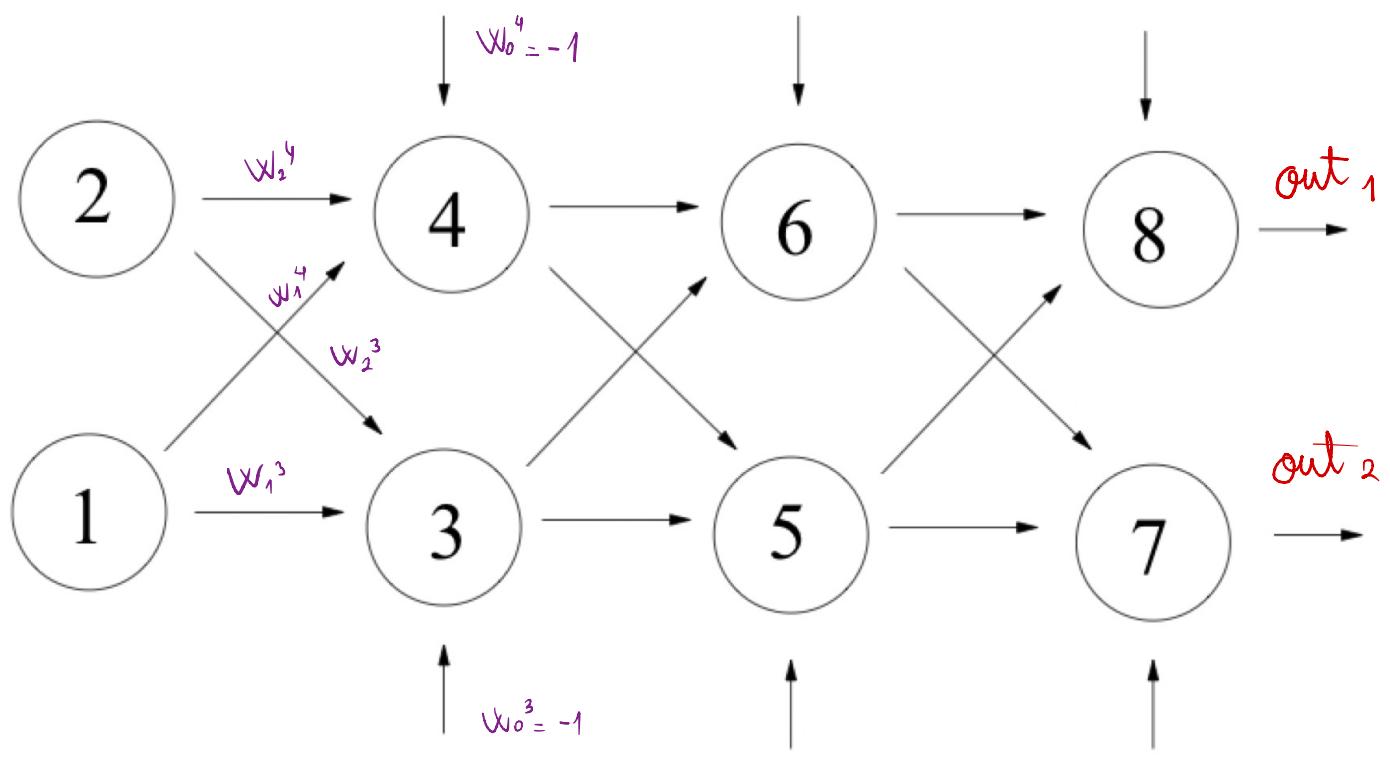
son iguales

por simetría → menos pesos

les llega lo mismo

Algoritmo Regla Delta

uno de Redes Neuronales



$$\text{out}_1 = \alpha_8 = g(\ln_8)$$

$$\text{out}_2 = \alpha_7 = g(\ln_7)$$

$$\text{Salida de la } 3 = a_3 = \text{Sig}(\ln_3) = 07310$$

$$a_4 = \text{Sig}(\ln_4) = 07310$$

función de activación = signo de

$$\ln_5 = -W_0^5 + a_4 \cdot W_4^5 + a_3 \cdot W_3^5 = 0462$$

$$\ln_6 = 0462$$

$$a_5 = 06135$$

Lo que sale de la 5 y 6

$$a_6 = 06135$$

$$\ln_7 = -1 + 06135 + 06135 = 0227$$

$$\ln_8 = 0227$$

$$a_7 = 05565 = g(\underset{\text{``}}{\ln_7})$$

$$a_8 = 05565$$

→ Corregir los pesos → del final al principio

$$\text{sig}'(z) = \text{sig}(z) \cdot (1 - \text{sig}(z))$$

Algoritmo de retropropagación

$$\begin{aligned} \Delta_8 &\leftarrow g'(\ln_8) \cdot (y_2 - a_8) = \\ &= 0.5565 \cdot (1 - 0.5565) \cdot (1 - 0.5565) = 0.1095 \end{aligned}$$

$$g'(\ln_8) = a_8 \cdot (1 - a_8)$$

sen (≡)

$$\Delta_7 \leftarrow 0.1095$$

$$\begin{aligned} W_6^7 &\leftarrow -\eta \Delta_7 \\ W_6^8 &\leftarrow \eta \Delta_8 \end{aligned}$$

Aunque aquí no me hacen falta pk son las últimas neuronas → Se lo que se han equivocado

Neurona intermedia

$$\Delta_6 \leftarrow g'(\ln_6) \cdot (W_6^7 \Delta_7 + W_6^8 \Delta_8) =$$

$$= a_6 \cdot (1 - a_6) \cdot (W_6^7 \Delta_7 + W_6^8 \Delta_8) =$$

$$= 0.6135 \cdot (1 - 0.6135) \cdot (1 \cdot 0.1095 + 1 \cdot 0.1095) =$$

$$= 0.53$$

En el examen da igual el resultado → puedo poner q NO tengo calculadora

$$W_6^6 \leftarrow W_6^6 - \eta \Delta_6$$

factor de aprendizaje

$$W_6^6 \leftarrow 1 - 0.1 \cdot 0.0519 = 0.9948$$

} Sinérvia

$$W_6^5 \leftarrow 0.9948$$

$$W_6^7 \leftarrow W_6^6 + \eta \Delta^7 \cdot a_6$$

el valor anterior

¿para que sirve?