# Bloque 2
# Aprendizaje Automático

# Tema 2:
# Aprendizaje Supervisado:

# Regresión Logística

## DOCENCIA VIRTUAL

**Responsable del Tratamiento:** Universitat Politècnia de València (UPV)

**Finalidad:** Prestación del servicio público de educación superior en base al interés público de la UPV (Art. 6.1.e del RGPD).

**Ejercicio de derechos y segunda capa informativa:** Podrán ejercer los derechos reconocidos en el RGPD y la LOPDGDD de acceso, rectificación, oposición, supresión, etc., escribiendo al correo dpd@upv.es.

Para obtener más información sobre el tratamiento de sus datos puede visitar el siguiente enlace: https://www.upv.es/contenidos/DPD.

**Propiedad Intelectual:** Uso exclusivo en el entorno del aula virtual.

Queda prohibida la difusión, distribución o divulgación de la grabación de las clases y particularmente su compartición en redes sociales o servicios dedicados a compartir apuntes.

La infracción de esta prohibición puede generar responsabilidad disciplinaria, administrativa y/o civil.

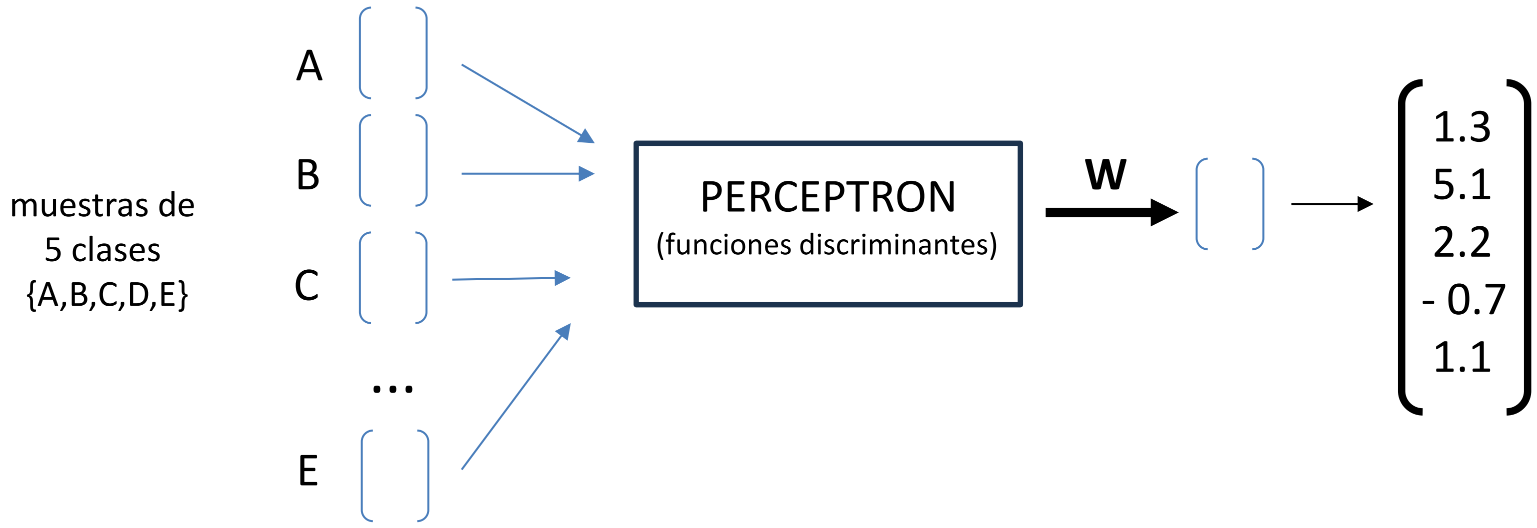
# Tema 2.2- Regresión Logística

1. Introducción. Motivación.
2. Codificación *one-hot* y distribución categórica
3. La función *softmax*
4. Modelo probabilístico de clasificación con *softmax*
5. Regresión logística
6. Aprendizaje por máxima verosimilitud
7. Aprendizaje por descenso de gradiente

# Bibliografía

- Kevin P. Murphy. Probabilistic Machine Learning: An Introduction. MIT Press, 2022

# 1. Introducción. Motivación.




Los clasificadores que utilizan funciones discriminantes lineales **no son probabilísticos.**

En un problema de clasificación, más que predecir el valor de la función lineal de cada clase, nos interesa predecir la probabilidad de que una muestra pertenezca a cada clase.

La regresión lineal no nos sirve porque puede devolver valores mayores de 1 y menores de 0. La probabilidad de pertenencia de una muestra a una clase tomará valores entre [0,1] y para ello usaremos la regresión logística en lugar de la regresión lineal.

# 2. Codificación one-hot y distribución categórica

## Variable categórica

Es una variable aleatoria que toma un valor de un conjunto finito de categorías (no ordenadas). Ejemplos:

$y \in$ {bicicleta, transporte-público, coche} (etiqueta de clase)
$y \in$ {clavel, tulipán, rosa} (etiqueta de clase)
$y \in$ color RGB (red, green, blue)
$y \in$ palabras de un vocabulario

# 2. Codificación *one-hot* y distribución categórica

Codificación *one-hot* de una variable $y$ que toma un valor entre C posibles {1, …, C}

$$\text{one-hot}(y) = \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_C \end{pmatrix}$$

vector *one-hot* de C componentes donde

$y_1$: indica pertenencia de la variable $y$ a la clase 1

$y_2$: indica pertenencia de la variable $y$ a la clase 2

….

Equivalentemente:

función identidad

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_C \end{pmatrix} = \begin{pmatrix} \mathbb{I}(y=1) \\ \vdots \\ \mathbb{I}(y=C) \end{pmatrix} \in \{0,1\}^C \quad \text{con} \quad \sum_c y_c = 1$$

$$\mathbb{I}(y=1) = \begin{cases} 1 & \text{si } y = 1 \text{ es verdadero, es decir } y \text{ pertenece a la clase 1} \\ 0 & \text{en caso contrario, es decir, si } y \text{ no pertenece a la clase 1} \end{cases}$$

# 2. Codificación *one-hot* y distribución categórica

Ejemplo: supongamos una variable categórica que toma un valor entre 3 clases, C = {1,2,3}

$y$: variable categórica (por ej., toma el valor 1, $y$=1)

$\boldsymbol{y}$: variable categórica en forma de vector *one-hot*

one-hot ($y$) = $\boldsymbol{y} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$

$y$: variable categórica (por ej., toma el valor 2, $y$=2)

$\boldsymbol{y}$: variable categórica en forma de vector *one-hot*

one-hot ($y$) = $\boldsymbol{y} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$

$y$: variable categórica (por ej., toma el valor 3, $y$=3)

$\boldsymbol{y}$: variable categórica en forma de vector *one-hot*

one-hot ($y$) = $\boldsymbol{y} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

# 2. Codificación *one-hot* y distribución categórica

## Distribución categórica

Es una distribución de probabilidad discreta entre las C posibles categorías (clases) de una variable categórica.

Una distribución categórica se modela con un vector de parámetros llamado $\boldsymbol{\theta}$ que está formado por valores entre 0 y 1 tal que la suma de todos los valores del vector $\boldsymbol{\theta}$ es 1

$$\boldsymbol{\theta} \in [0,1]^C \text{ tal que } \sum_c \theta_c = 1$$

Ejemplo:

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} = (0.4, 0.5, 0.1)^t = \begin{pmatrix} 0.4 \\ 0.5 \\ 0.1 \end{pmatrix}$$

probabilidad de que la variable categórica pertenezca a la clase 1

probabilidad de que la variable categórica pertenezca a la clase 2

$$p(y = c|\,\boldsymbol{\theta}) = \theta c \qquad\qquad p(y = 1|\,\boldsymbol{\theta}) = \theta_1 = 0.4$$

# 2. Codificación one-hot y distribución categórica

Se puede representar la distribución categórica de una variable $y$ utilizando su codificación *one-hot* con C elementos, los cuales serán todos 0 excepto el elemento correspondiente a la etiqueta de la clase de $y$.

$$\boxed{\mathrm{Cat}(\boldsymbol{y} \mid \boldsymbol{\theta})} = \prod_{c=1}^{C} \theta_c^{y_c}$$

elemento $c$ del vector *one-hot*

elemento $c$ del vector de parámetros $\boldsymbol{\theta}$

distribución categórica de una variable $y$
en forma de vector one-hot ($\boldsymbol{y}$)

Ejemplo:

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} = \begin{pmatrix} 0.4 \\ 0.5 \\ 0.1 \end{pmatrix}$$

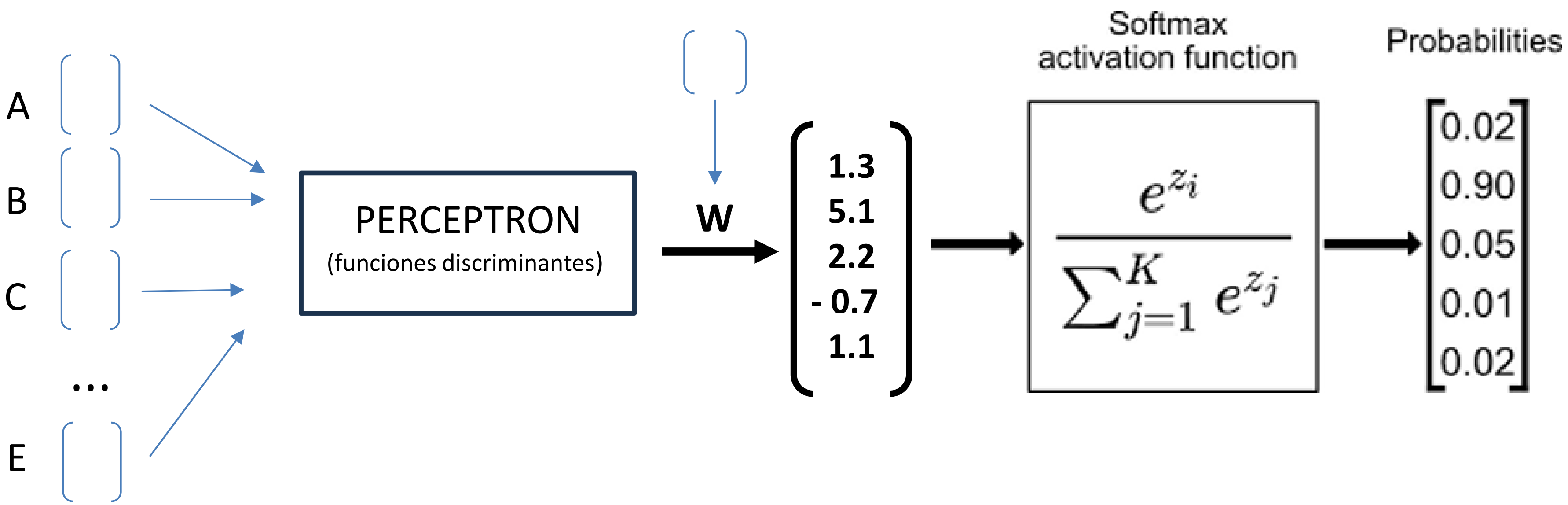
$p(y = 1 |\, \boldsymbol{\theta})$ =
$\mathrm{Cat}\,(\boldsymbol{y} = (1\ 0\ 0)^t | \boldsymbol{\theta}) =$
$0.4^1 \cdot 0.5^0 \cdot 0.1^0 = 0.4$

$p(y = 3 |\, \boldsymbol{\theta})$ =
$\mathrm{Cat}\,(\boldsymbol{y} = (0\ 0\ 1)^t | \theta) =$
$0.4^0 \cdot 0.5^0 \cdot 0.1^1 = 0.1$

# 3. La función *softmax*

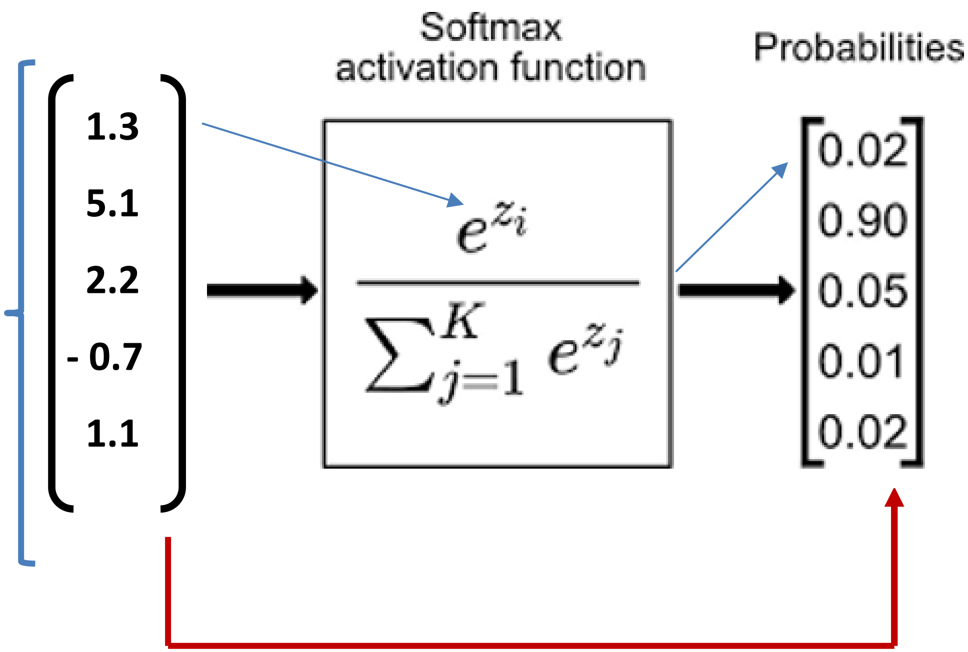
La función *softmax* es una función de activación muy utilizada en la capa de salida de una red neuronal para tareas de clasificación.

Se utiliza para convertir un conjunto de valores en probabilidades que sumen 1. En nuestro caso, lo vamos a utilizar para devolver la distribución de probabilidad de cada una de las clases soportadas en el modelo.



# 3. La función *softmax*: definición

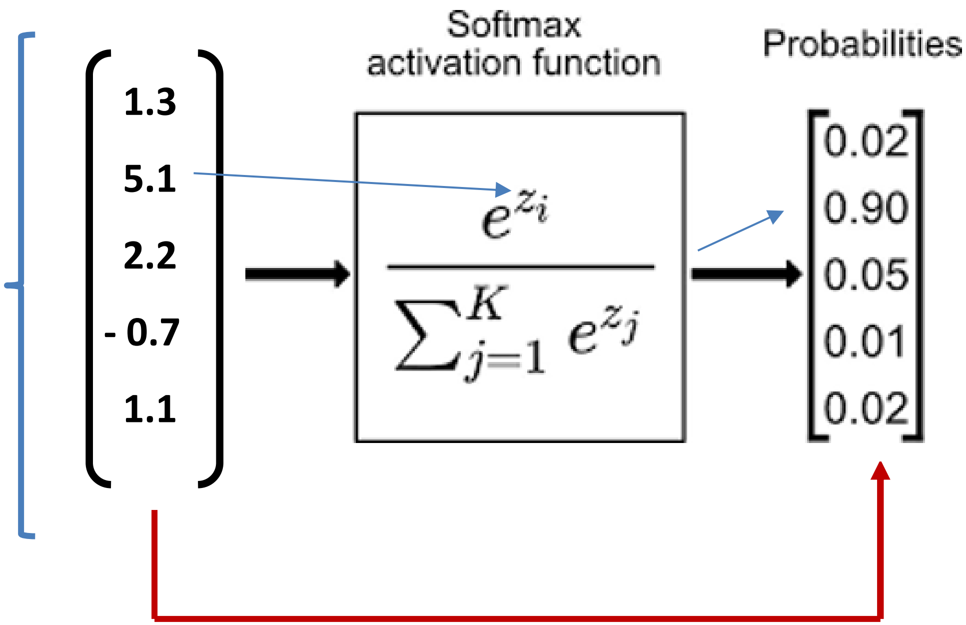
valores de las funciones lineales de 5 clases aplicadas a una muestra




pasar de una escala (-∞, + ∞) a una escala [0,1]
aplicar normalización

# 3. La función *softmax*: definición

valores de las funciones lineales de 5 clases aplicadas a una muestra




pasar de una escala (-∞, + ∞) a una escala [0,1]

aplicar normalización

Normalización en base al número $e$ tiene propiedades interesantes:

- $e^x$ es fácilmente diferenciable
- permite interpretar $z_i$ como log-probabilidades (logaritmos de probabilidades) no normalizadas a las que se denomina ***logits***

# 3. La función *softmax:* ejemplo

Dada la muestra $y = \begin{pmatrix} 0.3 \\ 3.5 \end{pmatrix} \begin{matrix} y1 \\ y2 \end{matrix}$ que sabemos que pertenece a la clase **A**

**A**

y dadas las funciones discrimiantes

$$g_A(y) = -1 - 3y_1 + y_2$$

$$g_B(y) = -3 + 3y_1 - y_2$$

Aplicamos las funciones discriminantes a la muestra

$$g_A(y) = -1 - 3y_1 + y_2$$

$$g_B(y) = -3 + 3y_1 - y_2$$

$$\begin{pmatrix} 0.3 \\ 3.5 \end{pmatrix} \downarrow \begin{pmatrix} -1 & -3 & 1 \\ -3 & 3 & -1 \end{pmatrix} \longrightarrow$$

$$\boldsymbol{a} = (a_1, a_2)^t$$

$$\begin{pmatrix} 1.6 \\ -5.6 \end{pmatrix}$$

**vector de logits**

# 3. La función *softmax:* ejemplo

$$g_A(y) = -1 - 3y_1 + y_2$$

$$g_B(y) = -3 + 3y_1 - y_2$$

$$\xrightarrow{\begin{pmatrix} 0.3 \\ 3.5 \end{pmatrix}} \begin{pmatrix} 1.6 \\ -5.6 \end{pmatrix} \xrightarrow{\textit{softmax}} \begin{pmatrix} 0.999 \\ 0.001 \end{pmatrix}$$

$$\boldsymbol{a} = (a_1, a_2)^t$$

$$softmax(a_1) = \frac{e^{1.6}}{e^{1.6} + e^{-5.6}} = \frac{4.953}{4.953+0.004} = 0.999$$

$$softmax(a_2) = \frac{e^{-5.6}}{e^{-5.6} + e^{1.6}} = \frac{0.004}{0.004+4.953} = 0.001$$

# 3. La función *softmax*: revisión logaritmos

número de Euler    $e = 2.718281828 \ldots$

$\log_e x = \ln x$ $\Longrightarrow$ Usaremos siempre log x para referirnos al logaritmo natural de x

En teoría de la probabilidad y ciencias de la computación, una log-probabilidad (***log-probability***) es el logaritmo de una probabilidad. Se utiliza para representar probabilidades en una escala logarítmica en lugar del intervalo estándar [0,1].

$\log(0.1) = -2.30$    $\log(0.5) = -0.69$    $\log(1)=0$ porque $e^0 = 1$

La utilización de logaritmos permite:

- visualizar mejor grandes variaciones en los datos (probabilidades muy pequeñas y muy grandes)
- analizar las variaciones en términos porcentuales y no absolutos

| **Pinicial** | **Pfinal** | **incremento** |
|---|---|---|
| 0.1 | 0.2 | 0.1 |
| log(0.1)= -2.3 | log(0.2)= -1.6 | +0.7 |
| 0.5 | 1 | 0.5 |
| log(0.5)= -0.69 | log(1)= 0 | +0.7 |

# 3. La función *softmax*: revisión logaritmos

El logaritmo de un número más pequeño de 1 es siempre negativo:

$\log(0.01) = -4.61$ porque $e^{-4.61} = 0.01$

Logaritmo negativo (neg-log): es el logaritmo del inverso de un número y se representa con el signo - . El logaritmo negativo de x se representa como $-\log x$.

$-\log(0.01) = 4.61$ $\quad\quad$ $\log(1/0.01) = 4.61$

Neg-log se conoce frecuentemente como ***escala logarítmica*** y se utiliza para convertir valores muy pequeños de probabilidad en valores más legibles o manejables.

# 3. La función *softmax:* ejercicios propuestos

Dado el clasificador que se muestra abajo y la muestra $x = (1{,}1)^t$, aplica la función softmax a $x$ y calcula el vector de probabilidades que indica la probabilidad de que la muestra $x$ pertenezca a cada clase.

$$g_A(y) = -1 - 2y_1 + y_2$$

$$g_B(y) = -y_1 + 2y_2$$

Dado el clasificador que se muestra abajo y la muestra $x = (1{,}1)^t$, aplica la función softmax a $x$ y calcula el vector de probabilidades que indica la probabilidad de que la muestra $x$ pertenezca a cada clase.

$$g_A(y) = -3 - 2y_1 - y_2$$

$$g_B(y) = -2 - y_1$$

# 4. Modelo probabilístico de clasificación con softmax

Sea **G** un clasificador definido con C funciones discriminantes: $[g_1, \ldots, g_C]$

**G** puede representarse mediante un clasificador equivalente, **G'**, con funciones discriminantes normalizadas probabilísticamente: $[g'_1, \ldots, g'_C]$

clase que devuelve el clasificador

$$c(\boldsymbol{x}) = \underset{c}{\operatorname{argmax}}\ g_c(\boldsymbol{x})$$

$$= \underset{c}{\operatorname{argmax}}\ e^{g_c(\boldsymbol{x})}$$

$G = g_c(x)$

$G' = e^{g_c(x)}$

$G$ y $G'$ son equivalentes con $e^{g_c(x)} \in \mathbb{R}^{\geq 0}$ estrictamente creciente

$$= \underset{c}{\operatorname{argmax}}\ \frac{e^{g_c(\boldsymbol{x})}}{\sum_{c'} e^{g_{c'}(\boldsymbol{x})}}$$

$G = e^{g_c(x)}$

$G' = k \cdot e^{g_c(x)}$

$G$ y $G'$ son equivalentes con $k$ constante positiva (invariable con $c$)

Se puede, por tanto, definir un clasificador equivalente mediante la transformación de $g_C$ a $g'_C$ a través de la función softmax:

$$g'_c(\boldsymbol{x}) = \frac{e^{g_c(\boldsymbol{x})}}{\sum_{\tilde{c}} e^{g_{\tilde{c}}(\boldsymbol{x})}}$$

donde $g_c(x)$ son *logits* (log-probabilidades no normalizadas)

# 4. Modelo probabilístico de clasificación con softmax

***La función softmax:*** transforma un vector de ***logits*** (log-probabilidades no normalizadas) $G \in \mathbb{R}^C$ en uno de probabilidades $G' \in [0,1]^C$

$$G' = \mathcal{S}(G) = \left[ \frac{e^{g_1}}{\sum_{\tilde{c}} e^{g_{\tilde{c}}}}, \dots, \frac{e^{g_C}}{\sum_{\tilde{c}} e^{g_{\tilde{c}}}} \right]$$

función softmax aplicada al clasificador G

donde se cumple

$$0 \leq \mathcal{S}(G)_c \leq 1 \quad \text{y} \quad \sum_c \mathcal{S}(G)_c = 1$$

# 4. Modelo probabilístico de clasificación con softmax

***Modelo probabilístico de clasificación con softmax:*** se predicen las probabilidades de todas las clases a partir de un clasificador $G : \mathcal{X} \rightarrow \mathbb{R}^C$

G

$$\begin{pmatrix} \\ \\ \\ \end{pmatrix} \xrightarrow{G' = S(G)} \begin{pmatrix} p(y = 1 | \boldsymbol{x}) \\ p(y = 2 | \boldsymbol{x}) \\ \dots \\ \\ \end{pmatrix}$$

G’

vector de logits $\boldsymbol{a}$
para la muestra $\boldsymbol{x}$
resultado de aplicar **G**

vector de probabilidades $\boldsymbol{\mu}$
para la muestra $\boldsymbol{x}$
resultado de aplicar **G’**

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = \mathrm{Cat}(\boldsymbol{y} \mid \mathcal{S}(G)) = \prod_c (\mathcal{S}(G))_c)^{y_c}$$

# 4. Modelo probabilístico de clasificación con softmax

***Conveniencia del modelo en inferencia:*** la predicción de las probabilidades de todas las clases permite aplicar reglas de clasificación más generales que la clasificación por máxima probabilidad a posteriori. Por ejemplo, para la aplicación de funciones de error que son diferentes para cada clase.

***Conveniencia del modelo en aprendizaje:*** permite plantear el aprendizaje probabilísticamente, con criterios estándar como máxima verosimilitud; además, gracias a la softmax, $G$ se puede elegir libremente sin estar sujetos a las restricciones de la probabilidad.

# 5. Regresión logística

La regresión logística es un modelo de clasificación lineal que se basa en los mismos principios de la regresión lineal, se modela igualmente con una función lineal solo que ahora se aplica la función ***softmax***.

En resumen, la regresión logística es un modelo con ***softmax*** y funciones discriminantes lineales.

No hay diferencia con los clasificadores basados en funciones discriminantes lineales, a excepción de que ahora predecimos las probabilidades de todas las clases.

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \mathbf{W}) = \mathrm{Cat}(\boldsymbol{y} \mid \boldsymbol{\mu})$$

vectores de características de dimensión D

$$\boldsymbol{\mu} = \mathcal{S}(\boldsymbol{a}), \quad \boldsymbol{a} = f(\boldsymbol{x}; \mathbf{W}) = \mathbf{W}^t \boldsymbol{x}, \quad \mathbf{W} \in \mathbb{R}^{D \times C} \quad \text{y} \quad \boldsymbol{x} \in \mathbb{R}^D$$

softmax que nos devuelve el vector de probabilidades

vector que resulta de aplicar el clasificador W a una muestra x

W es una matriz de Dimensiones x Clases

# 5. Regresión logística: ejemplo

$$C = D = 2, \quad a_1 = g_1(x_1, x_2) = -x_1 - x_2 + 1, \quad a_2 = g_2(x_1, x_2) = x_1 + x_2 - 1$$

$$\boldsymbol{a} = f(\boldsymbol{x}; \mathbf{W}) = \mathbf{W}^t \boldsymbol{x} \quad \text{con} \quad \mathbf{W}^t = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \end{pmatrix} \quad \text{y} \quad \boldsymbol{x} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$$

| | $\boldsymbol{x}^t$ | $\boldsymbol{a}^t$ | $\mu_1 = \mathcal{S}(\boldsymbol{a})_1$ | $\mu_2 = \mathcal{S}(\boldsymbol{a})_2$ |
|---|---|---|---|---|
| $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ | $(1, 0, 0)$ | $(1, -1)$ | $\frac{e^1}{e^1+e^{-1}} = 0.8808$ | $\frac{e^{-1}}{e^1+e^{-1}} = 0.1192$ |
| $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ | $(1, 1, 1)$ | $(-1, 1)$ | $\frac{e^{-1}}{e^{-1}+e^1} = 0.1192$ | $\frac{e^1}{e^{-1}+e^1} = 0.8808$ |
| $\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$ | $(1, 0.5, 0.5)$ | $(0, 0)$ | $\frac{e^0}{e^0+e^0} = 0.5000$ | $\frac{e^0}{e^0+e^0} = 0.5000$ |

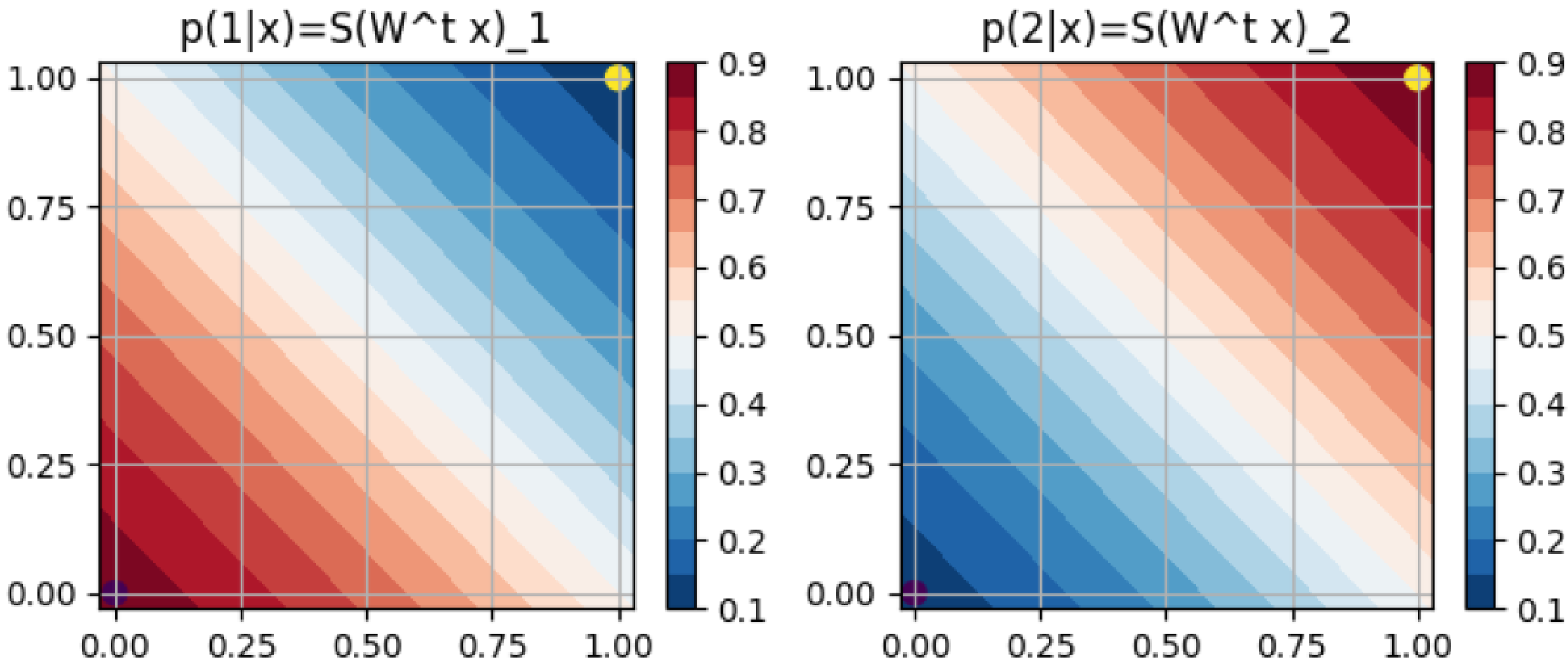
w1

$$\begin{pmatrix} 1 & -1 & -1 \end{pmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 1$$

w2

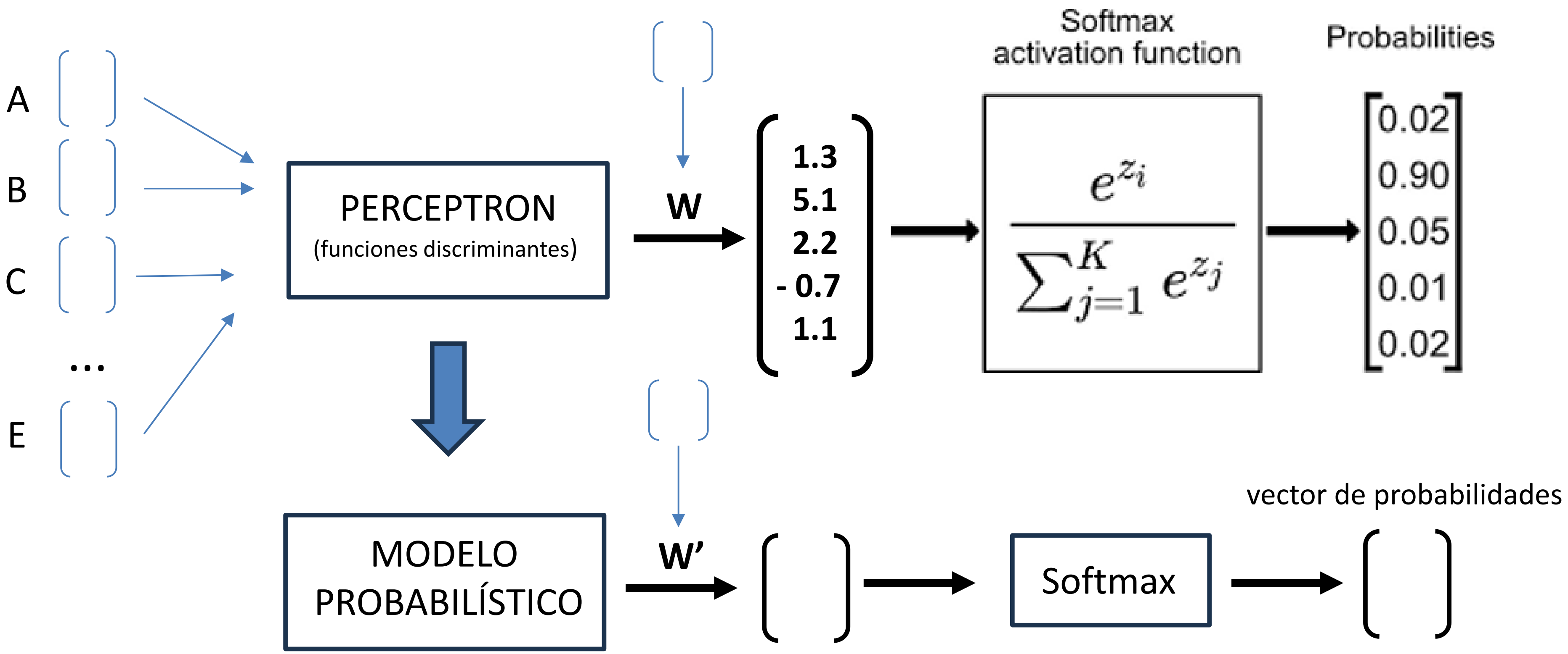
$$\begin{pmatrix} -1 & 1 & 1 \end{pmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = -1$$

# 5. Regresión logística: ejemplo



# 6. Aprendizaje por máxima verosimilitud

Hasta ahora hemos visto




Queremos aprender la matriz de pesos **W** probabilísticamente, es decir, con un modelo probabilístico en lugar de con un modelo que usa funciones discriminantes

# 6. Aprendizaje por máxima verosimilitud

**¿Qué es verosimilitud en un modelo probabilístico?**

*Regla de Bayes:*

$$P(y \mid x) = \frac{P(x,y)}{P(x)} = \frac{P(y)\,P(x \mid y)}{P(x)} = \frac{P(y)\,P(x \mid y)}{\sum_{y'} P(y')\,P(x \mid y')}$$

**probabilidad a posteriori**

verosimilitud

clase

probabilidad a posteriori → P(parámetro-a-estimar | dato)

verosimilitud (de los datos) → P(dato | parámetro-a-estimar)

**asumiendo un valor concreto para el parámetro que quiero estimar, ¿cómo de verosímil es que ese dé el dato?**

**probabilidad de que ocurra o se dé un determinado dato/muestra si es cierta la estimación que hemos efectuado o el estimador que hemos planteado**

# 6. Aprendizaje por máxima verosimilitud

## ¿Qué es verosimilitud de un modelo probabilístico?

Vamos a asumir que tenemos un problema de N=6 muestras que pertenecen a 3 clases C={1,2,3}. Mi dataset o conjunto de **datos** es:

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|---|
| | $\begin{bmatrix} \\ \end{bmatrix}$ | $\begin{bmatrix} \\ \end{bmatrix}$ | $\begin{bmatrix} \\ \end{bmatrix}$ | $\begin{bmatrix} \\ \end{bmatrix}$ | $\begin{bmatrix} \\ \end{bmatrix}$ | $\begin{bmatrix} \\ \end{bmatrix}$ |
| | C=3 | C=1 | C=3 | C=2 | C=2 | C=3 |
| variable que denota la clase de cada muestra | $y_1$=3 | $y_2$=1 | $y_3$=3 | $y_4$=2 | $y_5$=2 | $y_6$=3 |
| variable clase en formato one-hot | $\boldsymbol{y_1}=\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ | $\boldsymbol{y_2}=\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ | $\boldsymbol{y_3}=\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ | $\boldsymbol{y_4}=\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ | $\boldsymbol{y_5}=\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ | $\boldsymbol{y_6}=\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ |

# 6. Aprendizaje por máxima verosimilitud

**¿Qué es verosimilitud de un modelo probabilístico?**

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ |
|---|---|---|---|---|---|
| $\begin{bmatrix} \\ \end{bmatrix}$ | $\begin{bmatrix} \\ \end{bmatrix}$ | $\begin{bmatrix} \\ \end{bmatrix}$ | $\begin{bmatrix} \\ \end{bmatrix}$ | $\begin{bmatrix} \\ \end{bmatrix}$ | $\begin{bmatrix} \\ \end{bmatrix}$ |
| C=3 | C=1 | C=3 | C=2 | C=2 | C=3 |
| $y_1$=3 | $y_2$=1 | $y_3$=3 | $y_4$=2 | $y_5$=2 | $y_6$=3 |

- El conjunto de muestras se ha generado o escogido de forma independiente
- Queremos establecer el mecanismo que generó el conjunto de muestras, es decir, su función de distribución
- Particularmente, queremos estimar un vector de probabilidades $\boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix}$ tal que

$$\mathrm{p}(y_1{=}3 \mid x_1, \boldsymbol{\theta}) \cdot \mathrm{p}(y_2{=}1 \mid x_2, \boldsymbol{\theta}) \cdot \mathrm{p}(y_3{=}3 \mid x_3, \boldsymbol{\theta}) \cdot \mathrm{p}(y_4{=}2 \mid x_4, \boldsymbol{\theta}) \cdot \mathrm{p}(y_5{=}2 \mid x_5, \boldsymbol{\theta}) \cdot \mathrm{p}(y_6{=}3 \mid x_6, \boldsymbol{\theta})$$

sea el mayor valor posible

# 6. Aprendizaje por máxima verosimilitud

p($y_1$=3|$x_1, \boldsymbol{\theta}$) · p($y_2$=1|$x_2, \boldsymbol{\theta}$) · p($y_3$=3|$x_3, \boldsymbol{\theta}$) · p($y_4$=2|$x_4, \boldsymbol{\theta}$) · p($y_5$=2|$x_5, \boldsymbol{\theta}$) · p($y_6$=3|$x_6, \boldsymbol{\theta}$)

**Función de verosimilitud de un modelo probabilístico**

$$\mathrm{L}(\theta) = \prod_{n=1}^{N} p(y_n \mid x_n, \theta)$$

$\mathrm{L}(\theta)$ nos indica como de verosímil es que la muestra 1 pertenezca a su clase $y_1$ , y la muestra 2 pertenezca a su clase $y_2$ , y la muestra 3 pertenezca a su clase $y_3$ , …. dado que las muestras $x_n$ siguen una distribución $\theta$.

En otras palabras, $\mathrm{L}(\theta)$ es la probabilidad de observar las muestras (que la muestra 1 pertenezca a su clase $y_1$ , y la muestra 2 pertenezca a su clase $y_2$ …) cuando los datos se extraen de la distribución de probabilidad con parámetro $\theta$.

# 6. Aprendizaje por máxima verosimilitud

Uno de los problemas que nos podemos encontrar es que al multiplicar muchas probabilidades nos va a quedar números demasiados pequeños que son difíciles de visualizar. Así que aplicamos el logaritmo.:

**Función de verosimilitud logarítmica (log-verosimilitud)**

$$\mathrm{LL}(\theta) = \ log \prod_{n=1}^{N} p(y_n \mid x_n, \theta)$$

y aplicando $\log(x \cdot y) = \log(x) + \log(y)$

$$\mathrm{LL}(\theta) = \ log \prod_{n=1}^{N} p(y_n \mid x_n, \theta) = \sum_{n=1}^{N} log\, p(y_n \mid x_n, \theta)$$

## 6. Aprendizaje por máxima verosimilitud

Vamos a aplicar la log-verosimilitud con el parámetro **W**

Dado un conjunto de vectores de pesos **W ,** podemos estimar la log-verosimilitud de **W**, y el valor **LL(W)** nos dará una medida de la verosimilitud de que las muestras pertenezcan a sus respectivas clases.

$$a_n = \mathbf{W}^t x_n$$

$$a = f(x; \mathbf{W})$$

$$\xrightarrow{S(a)}$$

$$\mu = \mathcal{S}(a) = \theta$$

$$p(y \mid x, \mathbf{W}) = \mathrm{Cat}(y \mid \mu)$$

# 6. Aprendizaje por máxima verosimilitud

Objetivo: establecer un criterio para aprender **W** a partir de un conjunto de datos de entrenamiento $\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$

Log-verosimilitud: log-probabilidad de $\mathcal{D}$ interpretada como una función de **W**

$$\mathrm{LL}(\mathbf{W}) = \log p(\mathcal{D} \mid \mathbf{W}) =$$

$$\log \prod_{n=1}^{N} p(\boldsymbol{y}_n \mid \boldsymbol{x}_n, \mathbf{W}) =$$

$$\sum_{n=1}^{N} \log \mathrm{Cat}(\boldsymbol{y}_n \mid \boldsymbol{\mu}_n) =$$

$$\sum_{n=1}^{N} \log \prod_{c=1}^{C} \mu_{nc}^{y_{nc}} =$$

$$\sum_{n=1}^{N} \sum_{c=1}^{C} y_{nc} \log \mu_{nc}$$

log-verosimilitud

$$\mathrm{LL}(\theta) = \log \prod_{n=1}^{N} p(y_n \mid x_n, \theta)$$

regresión lineal

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \mathbf{W}) = \mathrm{Cat}(\boldsymbol{y} \mid \boldsymbol{\mu})$$

$$\log(x \cdot y) = \log(x) + \log(y)$$

$$\mathrm{Cat}(\boldsymbol{y} \mid \boldsymbol{\theta}) = \prod_{c=1}^{C} \theta_c^{y_c}$$

# 6. Aprendizaje por máxima verosimilitud: ejemplo

- ***Ejemplo:*** log-verosimilitud de $\mathbf{W}^t = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \end{pmatrix}$ con dos datos $\mathcal{D} = \{((1,0,0)^t, (1,0)^t), ((1,1,1)^t, (0,1)^t)\}$

w1 $(1 \quad -1 \quad -1)$

w2 $(-1 \quad 1 \quad 1)$

x1 $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

c=1 $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$\boldsymbol{y_1}$

$y_{11} = 1$
$y_{12} = 0$

x2 $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

c=2 $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$\boldsymbol{y_2}$

$y_{21} = 0$
$y_{22} = 1$

# 6. Aprendizaje por máxima verosimilitud: ejemplo

***Ejemplo:*** log-verosimilitud de $\mathbf{W}^t = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \end{pmatrix}$ con dos datos $\mathcal{D} = \{((1,0,0)^t, (1,0)^t), ((1,1,1)^t, (0,1)^t)\}$

$$\begin{aligned} \mathrm{LL}(\mathbf{W}) &= \sum_{n=1}^{N} \log \prod_{c=1}^{C} \mu_{nc}^{y_{nc}} = \sum_{n=1}^{N} \sum_{c=1}^{C} y_{nc} \log \mu_{nc} \\ &= y_{11} \log \mu_{11} + y_{12} \log \mu_{12} + y_{21} \log \mu_{21} + y_{22} \log \mu_{22} \\ &= \log \mu_{11} + \log \mu_{22} \\ &= \log 0.8808 + \log 0.8808 = -0.1269 - 0.1269 = -0.2538 \end{aligned}$$

$\mathrm{LL}(\mathbf{W}) = -0.2538$ es la probabilidad de observar que la muestra 1 pertenece a la clase 1 y la muestra 2 pertenece a la clase 2 cuando se usa la matriz de pesos **W**.

***Aprendizaje por máxima verosimilitud:*** elegimos una $\mathbf{W}$ que otorgue máxima probabilidad a $\mathcal{D}$

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmax}} \ \mathrm{LL}(\mathbf{W})$$

# 6. Aprendizaje por máxima verosimilitud

**Verosimilitud logarítmica negativa (neg-log-verosimilitud)**

Es la log-verosimilitud con el signo cambiado y normalizada por el número de datos

$$\mathrm{NLL}(\mathbf{W}) = -\frac{1}{N}\mathrm{LL}(\mathbf{W}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_{nc}\log\mu_{nc}$$

***Ejemplo:*** neg-log-verosimilitud de $\mathbf{W}^t = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \end{pmatrix}$ con dos datos
$\mathcal{D} = \{((1,0,0)^t,(1,0)^t),((1,1,1)^t,(0,1)^t)\}$

$$\mathrm{NLL}(\mathbf{W}) = -\frac{1}{2}\mathrm{LL}(\mathbf{W}) = 0.1269$$

# 6. Aprendizaje por máxima verosimilitud

- ***Riesgo empírico con log-pérdida:*** es lo mismo que NLL

$$\mathcal{L}(\mathbf{W}) = \frac{1}{N}\sum_{n=1}^{N} \ell(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n) = \mathrm{NLL}(\mathbf{W})$$

con

$$\ell(\boldsymbol{y}_n, \hat{\boldsymbol{y}}_n) = -\log p(\boldsymbol{y}_n \mid \boldsymbol{\mu}_n) = -\sum_{c=1}^{C} y_{nc} \log \mu_{nc}$$

- Si el modelo asigna probabilidad uno a la clase correcta, la pérdida es nula
- Si no, la pérdida será positiva y será más grande cuanto menor sea la probabilidad asignada a la clase correcta

El riesgo con log-pérdida, pérdida logística o pérdida logarítmica, de una matriz de pesos W o NLL(W) es mucho más flexible que el riesgo con pérdida 01 como en el Perceptron:

- riesgo con pérdida 01: muestra bien clasificada (0), muestra mal clasificada (1)
- riesgo con log-pérdida: muestra bien clasificada (0), muestra mal clasificada (mayor pérdida cuanto menor sea la probabilidad asignada a la clase real)

# 6. Aprendizaje por máxima verosimilitud

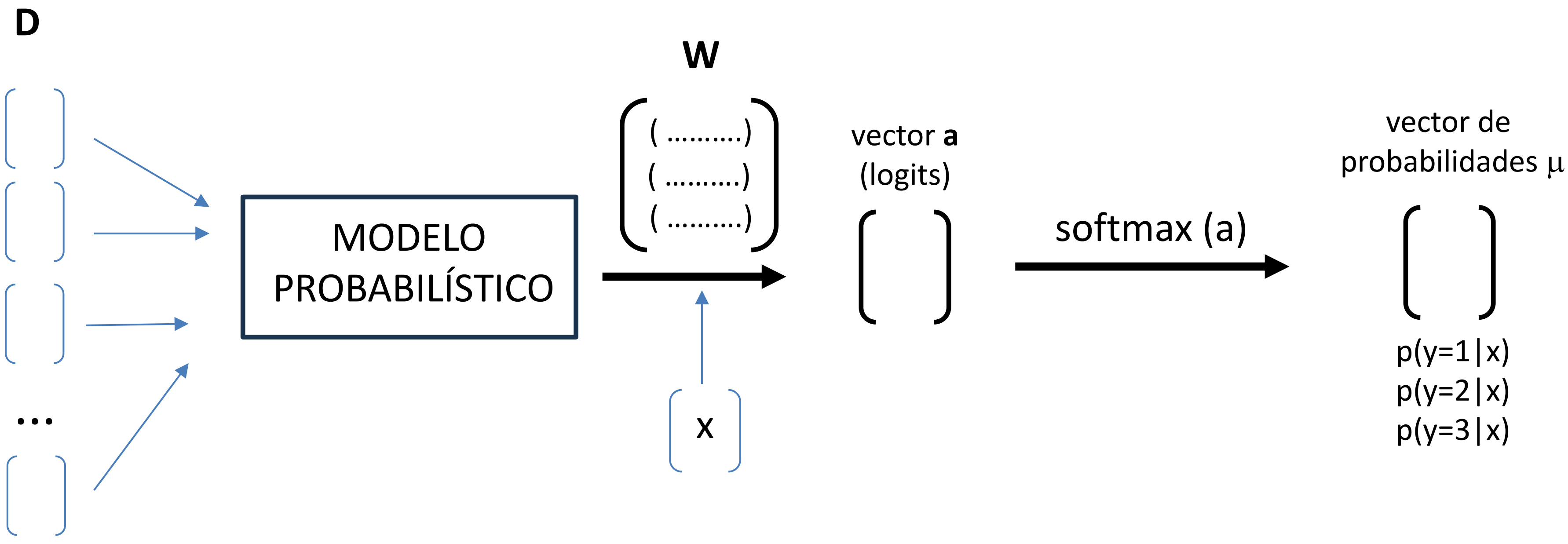
- ***Aprendizaje por mínima NLL:*** aprendizaje por máxima verosimilitud planteado como un problema de minimización

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}} \ \mathrm{NLL}(\mathbf{W})$$

La NLL o pérdida logarítmica:

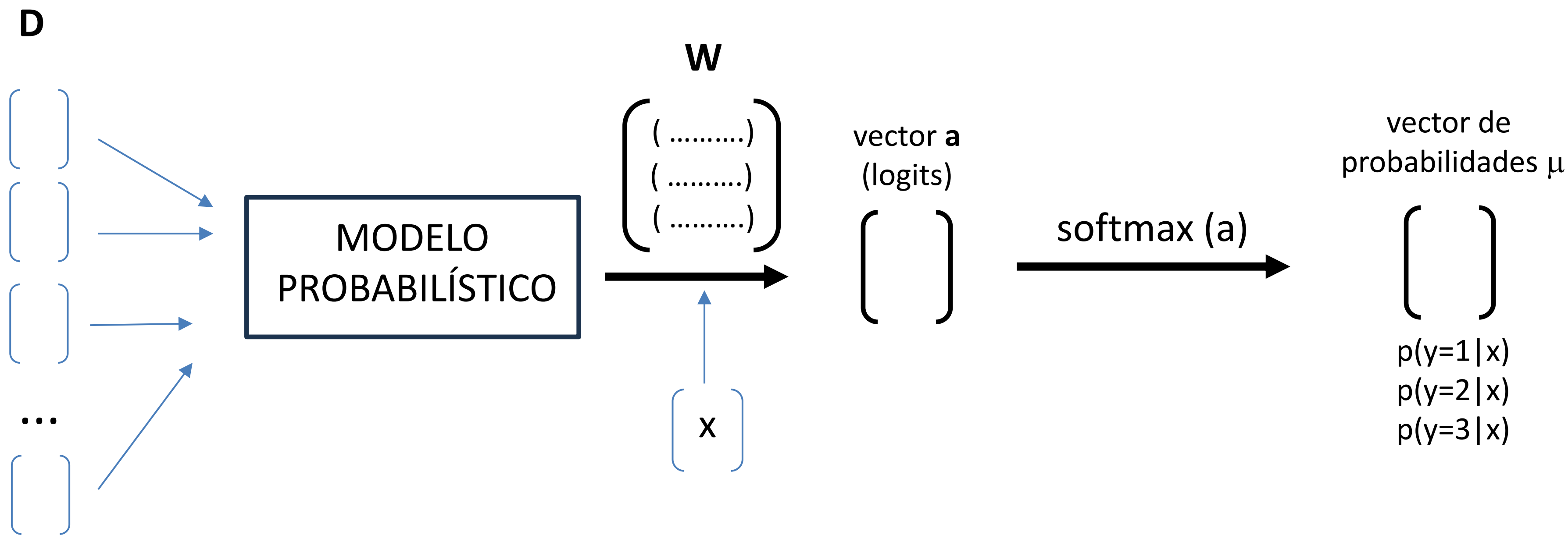
- es la métrica de evaluación de modelos de clasificación basada en probabilidades más importante
- es una métrica muy eficiente para comparar un modelo de aprendizaje con otro
- es derivable por lo que podemos minimizar NLL con técnicas estándar como descenso de gradiente
- el objetivo de cualquier modelo de aprendizaje automático probabilístico es minimizar este valor

# 6. RESUMEN




calcular la mejor matriz de pesos (W): la matriz que devolverá la probabilidad más alta de que las muestras pertenezcan a sus respectivas clases

# 6. RESUMEN



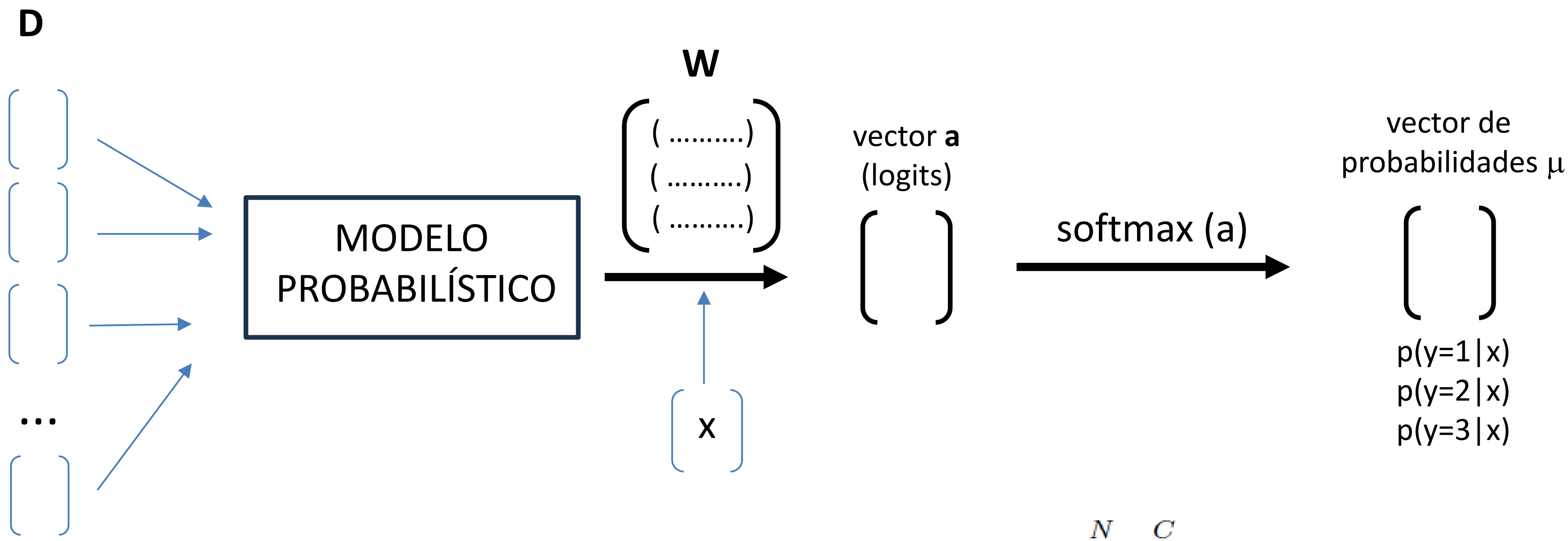

W1   P(D|W1)   p($y_1$=3|$x_1$, W1) · p($y_2$=1|$x_2$, W1) · p($y_3$=2|$x_3$, W1) .....

W2   P(D|W2)   p($y_1$=3|$x_1$, W2) · p($y_2$=1|$x_2$, W2) · p($y_3$=2|$x_3$, W2) .....

W3   P(D|W3)   p($y_1$=3|$x_1$, W3) · p($y_2$=1|$x_2$, W3) · p($y_3$=2|$x_3$, W3) .....

verosimilitud

# 6. RESUMEN



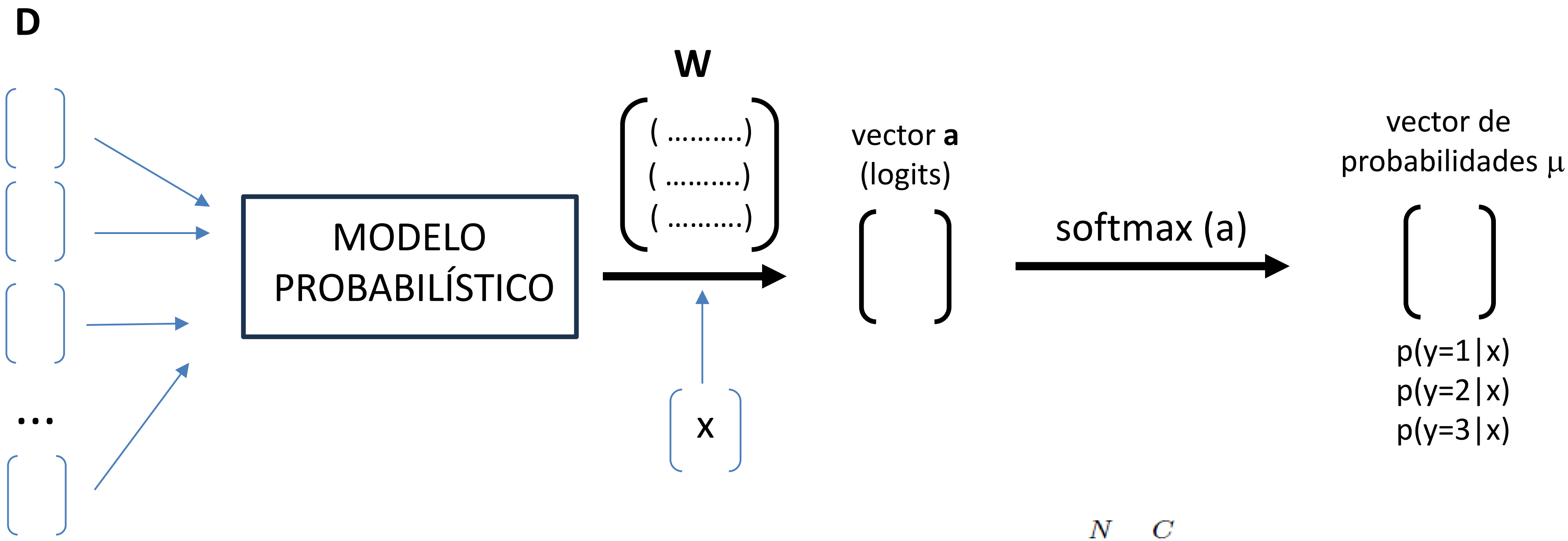

**LL(W1)** $\sum_{n=1}^{N} \sum_{c=1}^{C} y_{nc} \log \mu_{nc}$

| | | |
|---|---|---|
| W1 | log P(D\|W1) | $\boxed{\log p(y_1{=}3 \mid x_1, \mathrm{W1}) \cdot p(y_2{=}1 \mid x_2, \mathrm{W1}) \cdot p(y_3{=}2 \mid x_3, \mathrm{W1}) \;.....}$ |
| W2 | log P(D\|W2) | $\log p(y_1{=}3 \mid x_1, \mathrm{W2}) \cdot p(y_2{=}1 \mid x_2, \mathrm{W2}) \cdot p(y_3{=}2 \mid x_3, \mathrm{W2}) \;.....$ |
| W3 | log P(D\|W3) | $\log p(y_1{=}3 \mid x_1, \mathrm{W3}) \cdot p(y_2{=}1 \mid x_2, \mathrm{W3}) \cdot p(y_3{=}2 \mid x_3, \mathrm{W3}) \;.....$ |

log-verosimilitud

$$\max_i \; \mathbf{LL}(W_i)$$

# 7. RESUMEN




**LL(W1)** $\sum_{n=1}^{N} \sum_{c=1}^{C} y_{nc} \log \mu_{nc}$

| | | |
|---|---|---|
| W1 | log P(D\|W1) | log p($y_1$=3\|$x_1$, W1) · p($y_2$=1\|$x_2$, W1) · p($y_3$=2\|$x_3$, W1) ..... |
| W2 | log P(D\|W2) | |
| W3 | log P(D\|W3) | |

neg-log-verosimilitud

$$\mathbf{NLL(W1)} = -\frac{1}{N}\,\mathbf{LL(W1)}$$

$\min_i \ \mathbf{NLL}\ (\boldsymbol{W_i})$ → esto se calcula con el descenso de gradiente

# 6. Aprendizaje por máxima verosimilitud

## Planteamiento como un problema de minimización

Encontrar el vector de pesos W que minimice la log-pérdida de un conjunto de datos

- ***Ejemplo:*** $\mathcal{D} = \{((1,0,0)^t, (1,0)^t), ((1,1,1)^t, (0,1)^t)\}$; por simplicidad, suponemos que tenemos que elegir por mínima NLL entre

$$\mathbf{W}^t = \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \end{pmatrix} \quad \text{y} \quad \tilde{\mathbf{W}}^t = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & -1 \end{pmatrix}$$

$$\text{NLL}(\mathbf{W}) = -\frac{1}{2}\text{LL}(\mathbf{W}) = 0.1269$$

$$\text{NLL}(\tilde{\mathbf{W}}) = -\frac{1}{2}(\log \tilde{\mu}_{11} + \log \tilde{\mu}_{22}) = -\log \frac{e^{-1}}{e^{-1} + e^{1}} = \log(1 + e^2) = 2.1269$$

Elegimos **W** porque su NLL es menor.

# 7. Aprendizaje por descenso de gradiente

Supongamos una función $\mathcal{L}(\theta)$

**Gradiente de** $\boldsymbol{\mathcal{L}}(\boldsymbol{\theta})$ es la derivada parcial de la función respecto a sus parámetros:

$$\nabla\mathcal{L}(\theta) = \frac{d\mathcal{L}}{d\theta} = \frac{\partial\mathcal{L}}{\partial\theta}$$

**Dirección de descenso más pronunciada**: $-\nabla\mathcal{L}(\theta)$

# 7. Aprendizaje por descenso de gradiente

***Factor de aprendizaje:*** $\eta_i > 0$ juega el mismo papel que en Perceptrón; podemos elegir un valor pequeño constante, $\eta_i = \eta$

***Dirección de descenso más pronunciada:*** $-\boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})|_{\boldsymbol{\theta}_i}$ es el neg-gradiente de la función evaluada en $\boldsymbol{\theta}_i$

***Descenso por gradiente:*** algoritmo iterativo para minimizar una función $\mathcal{L}(\boldsymbol{\theta})$ a partir de un valor inicial de los parámetros $\boldsymbol{\theta}_0$ dado

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \eta_i \boldsymbol{\nabla}\mathcal{L}(\boldsymbol{\theta})|_{\boldsymbol{\theta}_i}$$

***Convergencia:*** si $\eta$ no es muy grande y la función es convexa (con forma de bol), converge a un mínimo (global)

# 7. Aprendizaje por descenso por gradiente: ejemplo

***Ejemplo:*** $\mathcal{L}(\theta) = \theta^2,\ \theta_0 = 10,\ \eta_t = 0.2,\ \frac{d\mathcal{L}}{d\theta} = 2\theta$ y tolerancia $0.01$

$$\nabla \mathcal{L}(\boldsymbol{\theta}) = \frac{d\mathcal{L}}{d\theta} = 2\theta$$

| $\boldsymbol{\theta}_i$ | $\boxed{-\eta_i \nabla \mathcal{L}(\boldsymbol{\theta})}$ | $\boldsymbol{\theta}_{i+1}$ |
|---|---|---|
| 10 | -4.0 | 6.0 |
| 6.0 | -2.4 | 3.6 |
| 3.6 | -1.44 | 2.16 |
| 2.16 | -0.864 | 1.296 |
| 1.296 | -0.5184 | 0.7776 |
| 0.7776 | -0.311 | 0.4666 |
| 0.4666 | -0.1866 | 0.2799 |
| 0.2799 | -0.112 | 0.168 |
| 0.168 | -0.0672 | 0.1008 |
| 0.1008 | -0.0403 | 0.0605 |
| 0.0605 | -0.0242 | 0.0363 |
| 0.0363 | -0.0145 | 0.0218 |
| 0.0218 | -0.0087 | 0.0131 |

$- 0.2 \cdot 2\theta = - 0.2 \cdot 2 \cdot 10 = - 4.0$

$- 0.2 \cdot 2\theta = - 0.2 \cdot 2 \cdot 6.0 = - 2.4$

# 7. Descenso por gradiente aplicado a regresion logística

- ***NLL:*** la NLL es una función convexa

$$\mathrm{NLL}(\mathbf{W}) = \frac{1}{N}\sum_{n=1}^{N} - \log p(\boldsymbol{y}_n \mid \boldsymbol{\mu}_n) \qquad \text{con} \quad \boldsymbol{\mu}_n = \mathcal{S}(\boldsymbol{a}_n) \quad \text{y} \quad \boldsymbol{a}_n = \mathbf{W}^t \boldsymbol{x}_n$$

- ***Gradiente de la NLL:*** haremos uso del siguiente resultado, sin demostrar

$$\begin{pmatrix} \frac{\partial\,\mathrm{NLL}}{\partial W_{11}} & \cdots & \frac{\partial\,\mathrm{NLL}}{\partial W_{1C}} \\ \vdots & \ddots & \vdots \\ \frac{\partial\,\mathrm{NLL}}{\partial W_{D1}} & \cdots & \frac{\partial\,\mathrm{NLL}}{\partial W_{DC}} \end{pmatrix} = \frac{\partial\,\mathrm{NLL}}{\partial \mathbf{W}^t} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t$$

- ***Descenso por gradiente aplicado a regresión logística:***

$$\mathbf{W}_0 = \mathbf{0}; \quad \mathbf{W}_{i+1} = \mathbf{W}_i - \eta_i \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t$$

# 7. Aprendizaje por descenso de gradiente

## Ejercicio de regresión logística

Dado un problema de clasificación en dos clases con dos datos bidimensionales $\mathcal{D} = \{((1,0,0)^t, (1,0)^t), ((1,1,1)^t, (0,1)^t)\}$:

- Realiza tres iteraciones del algoritmo de aprendizaje del modelo de regresión logística que minimiza la neg-log-verosimilitud con descenso por gradiente ($\eta = 1.0$) a partir de la matriz de pesos iniciales nulos.
- Calcula la probabilidad a posteriori de los datos a partir de la matriz de pesos final obtenida en el anterior apartado.
- Clasifica los datos por máxima probabilidad a posteriori.

# 7. Aprendizaje por descenso de gradiente: solución del ejercicio

| $n$ | $\boldsymbol{x_n}$ | $\boldsymbol{y_n}$ | $\boldsymbol{a_n} = (a_{n1}, a_{n2})$ | $\boldsymbol{\mu_n} = (S(a_{n1}), S(a_{n2}))$ | $\boldsymbol{\mu_n} - \boldsymbol{y_n}$ | $\boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t$ |
|---|---|---|---|---|---|---|
| 1 | $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ | $\begin{pmatrix} 0 & 0 \end{pmatrix}^t$ | $\frac{e^0}{e^0 + e^0} \quad \frac{e^0}{e^0 + e^0}$ <br> $\begin{pmatrix} 0.5 & 0.5 \end{pmatrix}^t$ | $\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ <br> $\begin{pmatrix} -0.5 & 0.5 \end{pmatrix}^t$ | $\begin{pmatrix} -0.5 & 0.5 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$ |
| 2 | $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ | $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ | $\begin{pmatrix} 0 & 0 \end{pmatrix}^t$ | $\frac{e^0}{e^0 + e^0} \quad \frac{e^0}{e^0 + e^0}$ <br> $\begin{pmatrix} 0.5 & 0.5 \end{pmatrix}^t$ | $\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ <br> $\begin{pmatrix} 0.5 & -0.5 \end{pmatrix}^t$ | $\begin{pmatrix} 0.5 & -0.5 \\ 0.5 & -0.5 \\ 0.5 & -0.5 \end{pmatrix}$ |

$$\begin{array}{c|c|c} & \mathbf{W_0}^t & \mathbf{W_1}^t \\ \hline \begin{array}{c} 1 \\ 2 \end{array} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & -0.25 & -0.25 \\ 0 & 0.25 & 0.25 \end{pmatrix} \end{array}$$

$$\boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t \qquad \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} -0.5 & 0.5 \end{pmatrix} = \begin{pmatrix} -0.5 & 0.5 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t \qquad \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0.5 & -0.5 \end{pmatrix} = \begin{pmatrix} 0.5 & -0.5 \\ 0.5 & -0.5 \\ 0.5 & -0.5 \end{pmatrix}$$

$$\sum_{n=1}^{N} \boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t \qquad \begin{pmatrix} -0.5 & 0.5 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0.5 & -0.5 \\ 0.5 & -0.5 \\ 0.5 & -0.5 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0.5 & -0.5 \\ 0.5 & -0.5 \end{pmatrix}$$

$$\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t \qquad \begin{pmatrix} 0 & 0 \\ 0.5 & -0.5 \\ 0.5 & -0.5 \end{pmatrix} \; / \; 2 = \begin{pmatrix} 0 & 0 \\ 0.25 & -0.25 \\ 0.25 & -0.25 \end{pmatrix}$$

matriz del gradiente
ó
gradiente de la NLL
$$\frac{\partial \, \mathrm{NLL}}{\partial \mathbf{W}^t}$$

$$\underset{\mathbf{W}_i}{\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}} - \begin{pmatrix} 0 & 0 \\ 0.25 & -0.25 \\ 0.25 & -0.25 \end{pmatrix} = \underset{\mathbf{W}_{i+1}}{\begin{pmatrix} 0 & 0 \\ -0.25 & 0.25 \\ -0.25 & 0.25 \end{pmatrix}}$$

# 7. Aprendizaje por descenso de gradiente: solución del ejercicio

| $n$ | $\boldsymbol{x_n}$ | $\boldsymbol{y_n}$ | $\boldsymbol{a_n} = (a_{n1}, a_{n2})$ | $\boldsymbol{\mu_n} = (S(a_{n1}), S(a_{n2}))$ | $\boldsymbol{\mu_n} - \boldsymbol{y_n}$ | $x_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t$ |
|---|---|---|---|---|---|---|
| 1 | $\begin{pmatrix}1\\0\\0\end{pmatrix}$ | $\begin{pmatrix}1\\0\end{pmatrix}$ | $\begin{pmatrix}0 & 0\end{pmatrix}^t$ | $\dfrac{e^{0}}{e^{0}+e^{0}} \quad \dfrac{e^{0}}{e^{0}+e^{0}}$ <br> $\begin{pmatrix}0.5 & 0.5\end{pmatrix}^t$ | $\begin{pmatrix}0.5\\0.5\end{pmatrix} - \begin{pmatrix}1\\0\end{pmatrix}$ <br> $\begin{pmatrix}-0.5 & 0.5\end{pmatrix}^t$ | $\begin{pmatrix}-0.5 & 0.5\\0 & 0\\0 & 0\end{pmatrix}$ |
| 2 | $\begin{pmatrix}1\\1\\1\end{pmatrix}$ | $\begin{pmatrix}0\\1\end{pmatrix}$ | $\begin{pmatrix}-0.5 & 0.5\end{pmatrix}^t$ | $\dfrac{e^{-0.5}}{e^{-0.5}+e^{0.5}} \quad \dfrac{e^{0.5}}{e^{0.5}+e^{-0.5}}$ <br> $\begin{pmatrix}0.269 & 0.731\end{pmatrix}^t$ | $\begin{pmatrix}0.269\\0.731\end{pmatrix} - \begin{pmatrix}0\\1\end{pmatrix}$ <br> $\begin{pmatrix}0.269 & -0.269\end{pmatrix}^t$ | $\begin{pmatrix}0.269 & -0.269\\0.269 & -0.269\\0.269 & -0.269\end{pmatrix}$ |

| | $\mathbf{W_0}^t$ | $\mathbf{W_1}^t$ | $\mathbf{W_2}^t$ |
|---|---|---|---|
| 1 <br> 2 | $\begin{pmatrix}0 & 0 & 0\\0 & 0 & 0\end{pmatrix}$ | $\begin{pmatrix}0 & -0.25 & -0.25\\0 & 0.25 & 0.25\end{pmatrix}$ | $\begin{pmatrix}0.116 & -0.37 & -0.37\\-0.116 & 0.37 & 0.37\end{pmatrix}$ |

$$\boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t \quad \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \text{ X } \begin{bmatrix} -0.5 & 0.5 \end{bmatrix} = \begin{pmatrix} -0.5 & 0.5 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t \quad \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \text{ X } \begin{bmatrix} 0.269 & -0.269 \end{bmatrix} = \begin{pmatrix} 0.269 & -0.269 \\ 0.269 & -0.269 \\ 0.269 & -0.269 \end{pmatrix}$$

$$\sum_{n=1}^{N} \boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t \quad \begin{pmatrix} -0.5 & 0.5 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0.269 & -0.269 \\ 0.269 & -0.269 \\ 0.269 & -0.269 \end{pmatrix} = \begin{pmatrix} -0.231 & 0.231 \\ 0.269 & -0.269 \\ 0.269 & -0.269 \end{pmatrix}$$

$$\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n(\boldsymbol{\mu}_n - \boldsymbol{y}_n)^t \begin{pmatrix} -0.231 & 0.231 \\ 0.269 & -0.269 \\ 0.269 & -0.269 \end{pmatrix} / \quad 2 \quad = \begin{pmatrix} -0.116 & 0.116 \\ 0.135 & -0.135 \\ 0.135 & -0.135 \end{pmatrix}$$

matriz del gradiente $\frac{\partial\,\mathrm{NLL}}{\partial \mathbf{W}}$

$$\underbrace{\begin{pmatrix} 0 & 0 \\ -0.25 & 0.25 \\ -0.25 & 0.25 \end{pmatrix}}_{\mathbf{W}_i} - \begin{pmatrix} -0.116 & 0.116 \\ 0.135 & -0.135 \\ 0.135 & -0.135 \end{pmatrix} = \underbrace{\begin{pmatrix} 0.116 & -0.116 \\ -0.37 & 0.37 \\ -0.37 & 0.37 \end{pmatrix}}_{\mathbf{W}_{i+1}}$$

# 7. Aprendizaje por descenso de gradiente: solución final

Realizar la tercera iteración

**Resultado final**

| i | $\frac{\partial \, \mathrm{NLL}}{\partial \mathbf{W}}$ | $\mathbf{W}^t_{i+1}$ |
|---|---|---|
| 0 | $\begin{pmatrix} 0.0 & 0.25 & 0.25 \\ 0.0 & -0.25 & -0.25 \end{pmatrix}$ | $\begin{pmatrix} 0.0 & -0.25 & -0.25 \\ 0.0 & 0.25 & 0.25 \end{pmatrix}$ |
| 1 | $\begin{pmatrix} -0.12 & 0.13 & 0.13 \\ 0.12 & -0.13 & -0.13 \end{pmatrix}$ | $\begin{pmatrix} 0.12 & -0.38 & -0.38 \\ -0.12 & 0.38 & 0.38 \end{pmatrix}$ |
| 2 | $\begin{pmatrix} -0.11 & 0.11 & 0.11 \\ 0.11 & -0.11 & -0.11 \end{pmatrix}$ | $\begin{pmatrix} 0.23 & -0.49 & -0.49 \\ -0.23 & 0.49 & 0.49 \end{pmatrix}$ |

$$p(\boldsymbol{Y} \mid \boldsymbol{X}, \mathbf{W}) = \mathcal{S}(\boldsymbol{X}\mathbf{W}) = \begin{pmatrix} 0.61 & 0.39 \\ 0.18 & 0.82 \end{pmatrix} \text{ con } \boldsymbol{X} \in \mathbb{R}^{N \times D} \text{ y } \mathbf{W} \in \mathbb{R}^{D \times C}$$

Por máxima probabilidad a posteriori, el primer dato se clasifica en la primera clase y el segundo dato en la segunda clase.

# 7. Aprendizaje por descenso de gradiente: ejercicios a resolver

1 ☐ Sea un modelo de regresión logística en notación compacta (homogénea) para un problema de clasificaciónn en $C = 3$ clases y datos representados por vectores de dimensión $D = 2$.

$$p(y \mid \boldsymbol{x}, \mathbf{W}) = \text{Cat}(y \mid \mathcal{S}(\mathbf{W}^t \boldsymbol{x}) \text{ con } \mathbf{W}^t = \begin{pmatrix} 0 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{C \times D}.$$

Dado $\boldsymbol{x} = (0.5, 0.5)^t$, la probabilidad $P$ de que $\boldsymbol{x}$ pertenezca a la clase 1 es:

A) $P < 0.25$
B) $0.25 \leq P < 0.5$
C) $0.5 \leq P < 0.75$
D) $0.75 \leq P$

2 ☐ Indica cuál de las siguientes afirmaciones sobre regresión logística es *incorrecta* (o escoge la última opción si las tres primeras son correctas):

A) Regresión logística es un modelo probabilístico de clasificación basado en la función softmax

B) Al tratarse de un modelo probabilístico de clasificación, regresión logística permite aplicar reglas de decisión más generales que la MAP (decidirse por la clase de máxima probabilidad a posteriori)

C) Al tratarse de un modelo probabilístico de clasificación, regresión logística permite plantear su aprendizaje probabilísticamente, con criterios estándar como máxima verosimilitud

D) Las tres afirmaciones anteriores son correctas

# 7. Aprendizaje por descenso de gradiente: ejercicios a resolver

## Problemas

1. Sea un modelo de regresión logística en notación compacta (homogénea) para un problema de clasificación en $C = 3$ clases y datos representados por vectores de dimensión $D = 2$.

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \mathbf{W}) = \mathrm{Cat}(\boldsymbol{y} \mid \mathcal{S}(\mathbf{W}^t \boldsymbol{x})) \text{ con } \mathbf{W}^t = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{C \times D}$$

Actualiza el valor de $\mathbf{W}$ con una iteración de descenso por gradiente con el conjunto de entrenamiento $\mathcal{D} = \{(\boldsymbol{x} = (1,1,1)^t, y = 1)\}$ y factor de aprendizaje $\eta = 0.1$.

# 7. Aprendizaje por descenso de gradiente: ejercicios a resolver

2. La siguiente tabla presenta un conjunto de 2 muestras de entrenamiento de 2 dimensiones procedentes de 2 clases:

| $n$ | $x_{n1}$ | $x_{n2}$ | $c_n$ |
|---|---|---|---|
| 1 | 1 | 1 | 2 |
| 2 | 0 | 1 | 1 |

Adicionalmente, la siguiente tabla representa una matriz de pesos iniciales con los pesos de cada clase dispuestos por columnas:

| $\mathbf{w}_1$ | $\mathbf{w}_2$ |
|---|---|
| 0. | 0. |
| -0.25 | 0.25 |
| 0. | 0. |

Se pide:

a) (0.5 puntos) Calcula el vector de logits asociado a cada muestra de entrenamiento.

b) (0.25 puntos) Aplica la función softmax al vector de logits de cada muestra de entrenamiento.

c) (0.25 puntos) Clasifica todas las muestras de entrenamiento. En caso de empate, elige cualquier clase.

d) (0.5 puntos) Calcula el gradiente de la función NLL en el punto de la matriz de pesos iniciales.

e) (0.5 puntos) Actualiza la matriz de pesos iniciales aplicando descenso por gradiente con factor de aprendizaje $\eta = 1.0$.