Introducción a la Inteligencia Artificial

Facultad de Ingeniería
Universidad de Buenos Aires

Índice

Clase 6

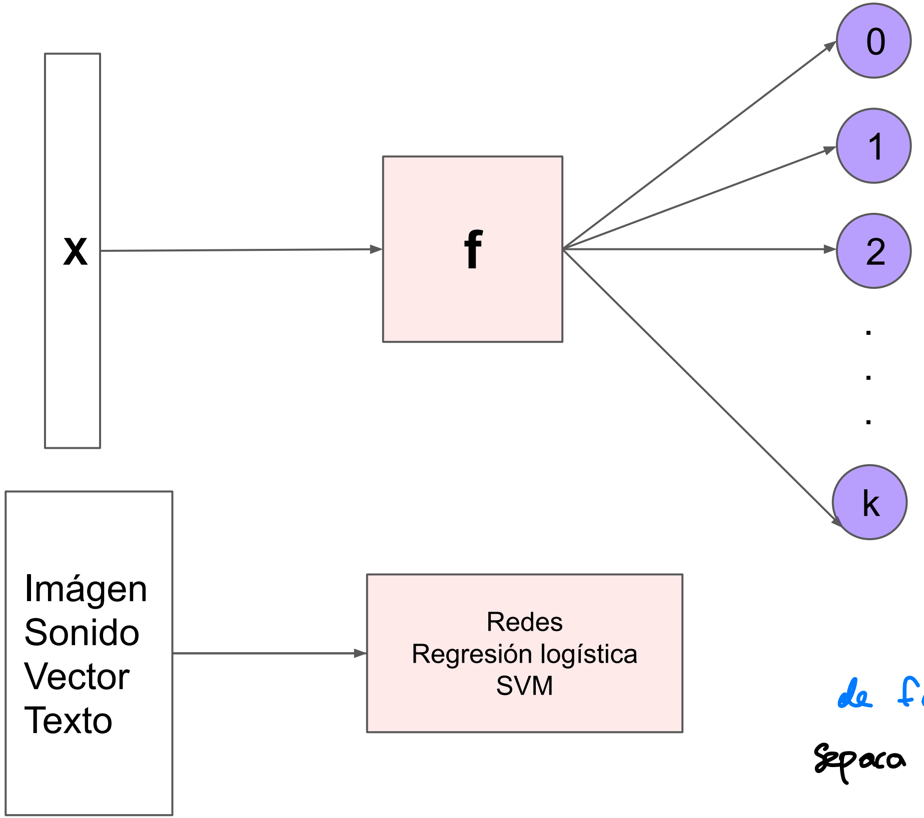
1. Clasificación Binaria
   a. Motivación
   b. Regresión Logística - Ejercicio de Aplicación
   c. Regresión Logística - Teoría
2. Clasificación Multiclase
   a. Motivación
   b. Softmax
   c. Ejercicio de Aplicación
3. Ejercicio integrador

IA — ML

- Tareas a resolver
  - regresion $x \in \mathbb{R}^{n \times p}$, $y \in \mathbb{R}^{r}$
  - clasificación $x \in \mathbb{R}^{n \times p}$, $y \in \mathbb{K}$
    - $\mathbb{K} = \{\pi_i \;\; i:1,N\}$
- Métricas → performance
  - → overfitting
  - → underfitting
- Preprocesamiento de datos → herramientas p/ adaptar la entrada a razón de mejorar la salida
- Modelos
  - → tipo de solución
    - → cerrada (OLS)
    - → numérica (perceptrón)
  - → tipo de problema
    - → supervisado
    - → no supervisado

Clasificación

Dado un $X \in D \subset \mathbb{R}^{n \times p}$ (un dataset de p features con n datos), X es la entrada e $Y \in K$ es la salida



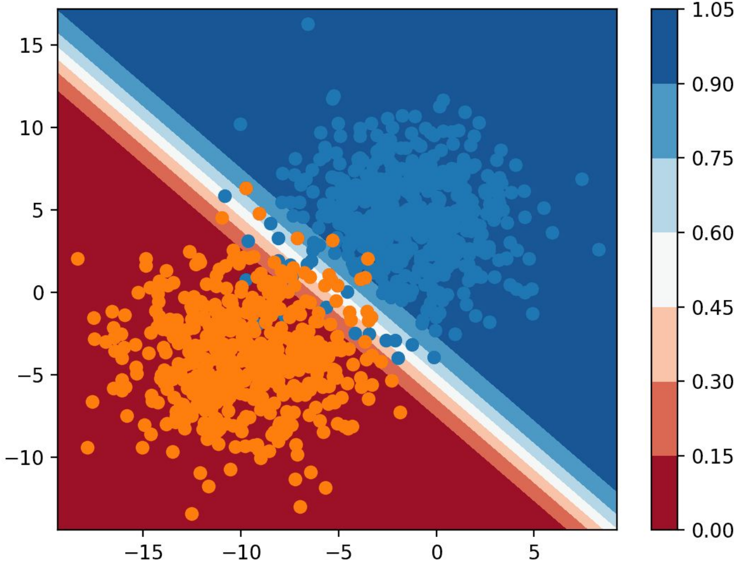
$$f : R^D \to \{0, 1, 2, \cdots, k\}$$

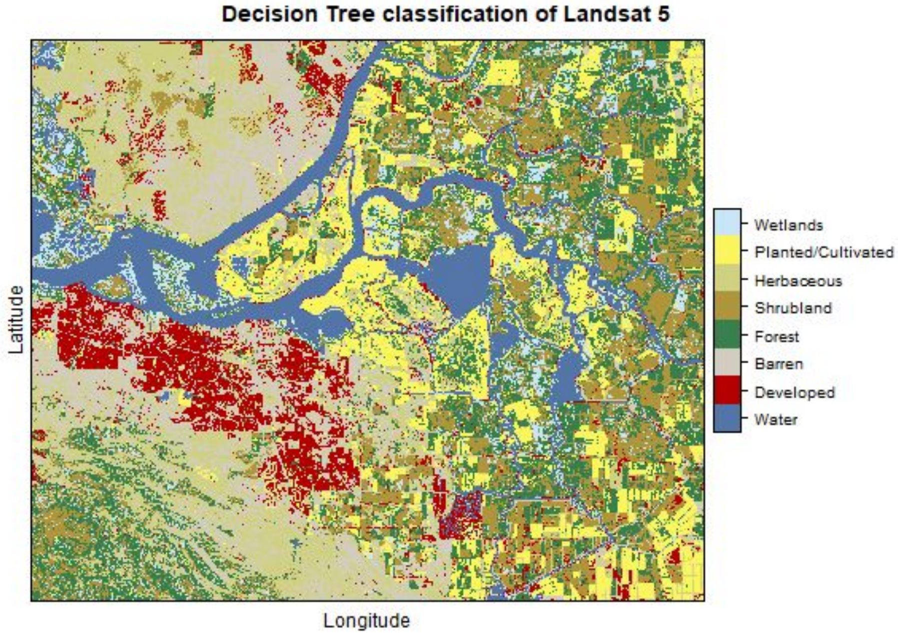
Supuesto importante: las clases de Y son separables.

$$\forall x_{ij}\ \exists\, y_i / y_i \in K \quad f(y)\ \exists!$$

Esto nos dice que en el espacio de features $\exists$ un hiperplano (región) que separa mis datos.

Clasificación



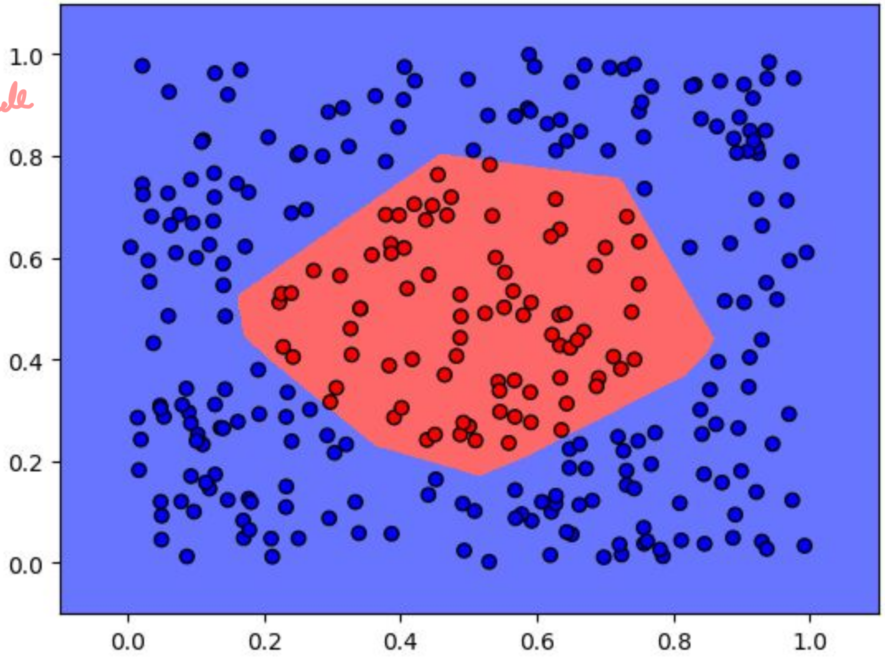


Clasificación binaria

$$f(x) = \begin{cases} 1 & \text{si } \mathbb{C}_1 \\ 0 & \text{o.w.} \end{cases}$$

Clasificación Binaria - Ejemplos

- Detección de fraudes — es fraude / no es fraude
- Diagnóstico médico — pos / neg
- Detección de spam — spam / ham
- Sentiment Analysis — + / -
- Detección de objetos — es obj1 / no es obj1
- Outliers



Clasificación Binaria - Diagnóstico médico

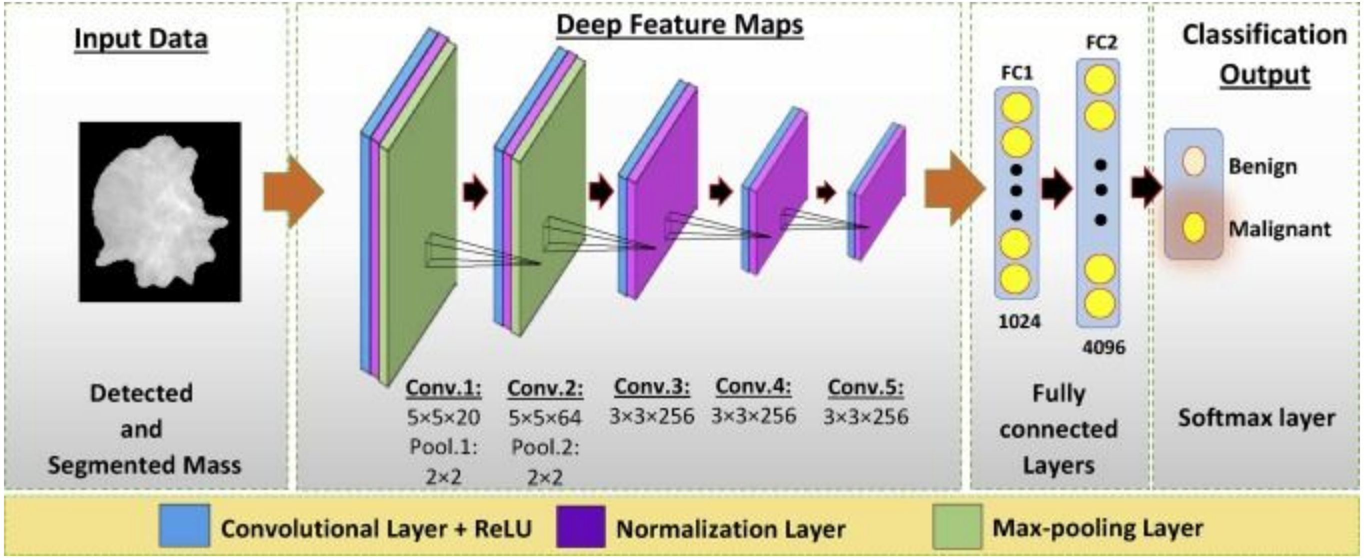


Imagen de: “A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification“

Clasificación Binaria - Spam detection

cabecera de clasif → Fully Conected layer

$\mathbb{R}^{796 \times 512 \times 796} \to \mathbb{R}^2$

12 bloques de transformers apilados + cabecera de clasif.

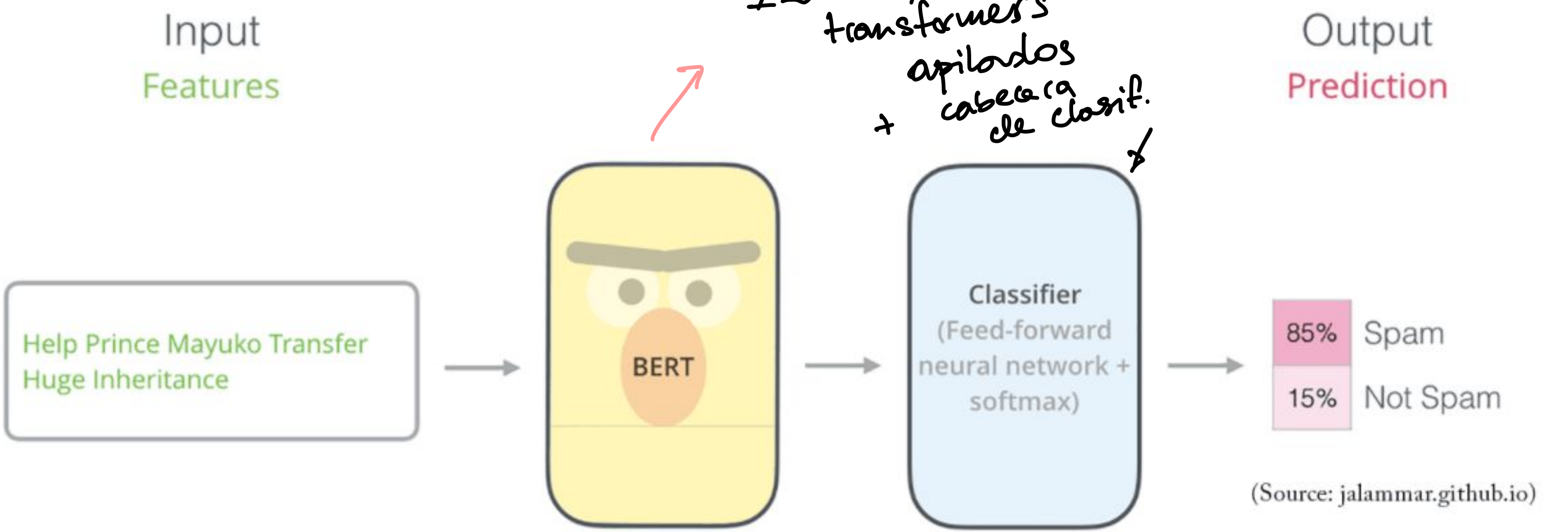
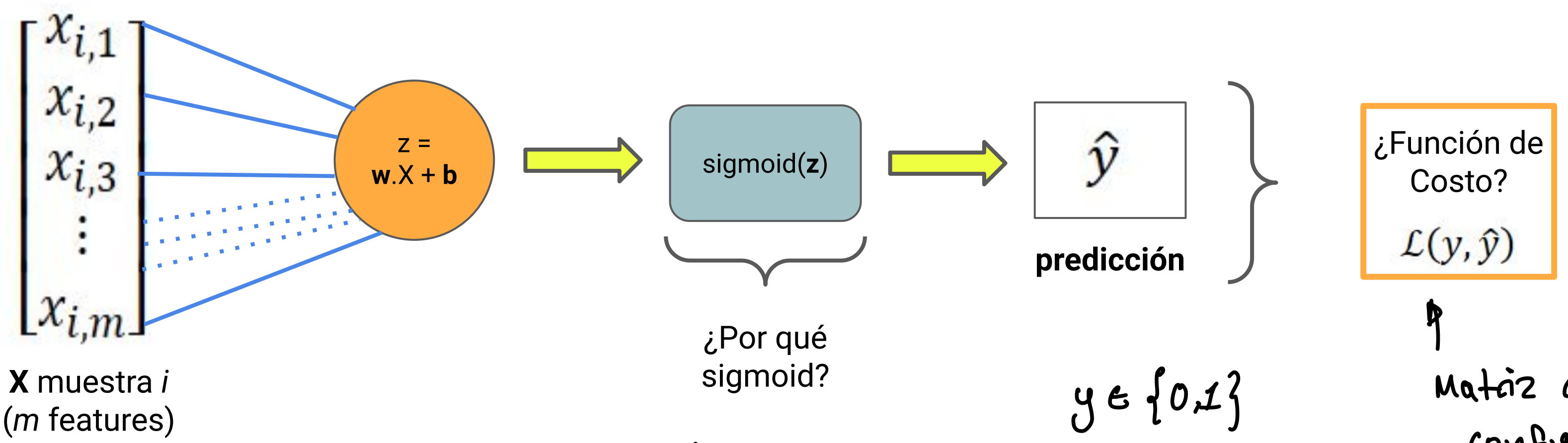


Imagen de: “Tutorial: Fine tuning BERT for Sentiment Analysis“

Regresión Logística → Extensión OLS p/clasificación



$y \in \{0,1\}$

Matriz de confusión

$X_i \in \mathbb{R}^m$

$z \in \mathbb{R}$

$W \in \mathbb{R}^{mxp}$

$b \in \mathbb{R}$

$\phi(z)$

$\phi : \mathbb{R} \mapsto [0,1]$

Métricas de performance en clasificación binaria:

Cómo no podemos calcular MSE (se mueve, reposo) $\Rightarrow$ construimos la matriz de confusión.

tengo
- y_{real}: Golden Label, Ground truth, True values
- y_{pred}

y_{real} ↓ \ y_{pred} →	T	F
T	TP	FN
F	FP	TN

- Accuracy: $\dfrac{TP + TN}{\# \text{muestras}} \in [0, 1]$
- specificity (TN rate): $\dfrac{TN}{TN + FP}$
- precision: $\dfrac{TP}{TP + FP}$
- recall $\dfrac{TP}{TP + FN}$

$$\left.\begin{array}{l} \text{precision} \\ \text{recall} \end{array}\right\} F_1 : 2 \, \frac{\text{prec.} \cdot \text{recall}}{\text{prec} + \text{recall}}$$

Formas de Clasificar

+ Definimos un proc de Scoring $\Rightarrow$ Cómo resolver la clasif.

$\uparrow$ complejidad $\Rightarrow$ $\uparrow$ computo $\overset{?}{\Rightarrow}$ mejor modelo

1. Modelo generativos: modelar las distrib de I/O $\Rightarrow$ generar la salida (permite generar información sintética) (GAN's)

2. Modelos discriminantes: planteo $\mathbb{P}(C_k | \bar{\bar{x}})$ $\Rightarrow$ mediante métodos de inferencia voy a estimar $\mathbb{P}$.

3. modelos de fn. discriminante: buscamos $f: \mathbb{R}^{n \times m} \mapsto \{C_1, \dots, C_k\}$

   $\hookrightarrow$ Regresión logística: (la idea es usar lo que sabemos de OLS).

   parto de suponer $y \sim \text{Bernouille}(1, \pi_i) \Rightarrow E(y) = \pi_i$, $Var(y) = \pi_i(1-\pi_i)$

me gustaría obtener algo como esto:

$$\pi_i = \bar{\bar{X}}^t \cdot \bar{\beta} \quad ; \quad \bar{\beta} \in \mathbb{R}^{p \times 1}$$

$\pi_i \in [0,1]$, pero $\bar{\bar{X}}^t \bar{\beta}$ no lo hace $\Rightarrow$ tengo que mapear el prob.

busco $g : [0,1] \longmapsto \mathbb{R}_{>0} \rightsquigarrow \text{odd}_i = \dfrac{\pi_i}{1-\pi_i}$

odd$_i$ - razón de probabilidad

es el cociente entre $P(A)$ ∧ $P(\bar{A})$

Vamos a tomar el log de odds, con esto obtengo una función biyectiva $\mathbb{R}_{>0} \longmapsto \mathbb{R}$. Esta transf. se llama logit (log odds ratio):

$$\eta_i = \text{logit}(\pi_i) = \bar{\bar{X}}^t \cdot \bar{\beta}$$

existe antilogit : $\pi_i = \text{antilogit}(\eta_i) = \dfrac{e^{\eta_i}}{1 + e^{\eta_i}}$

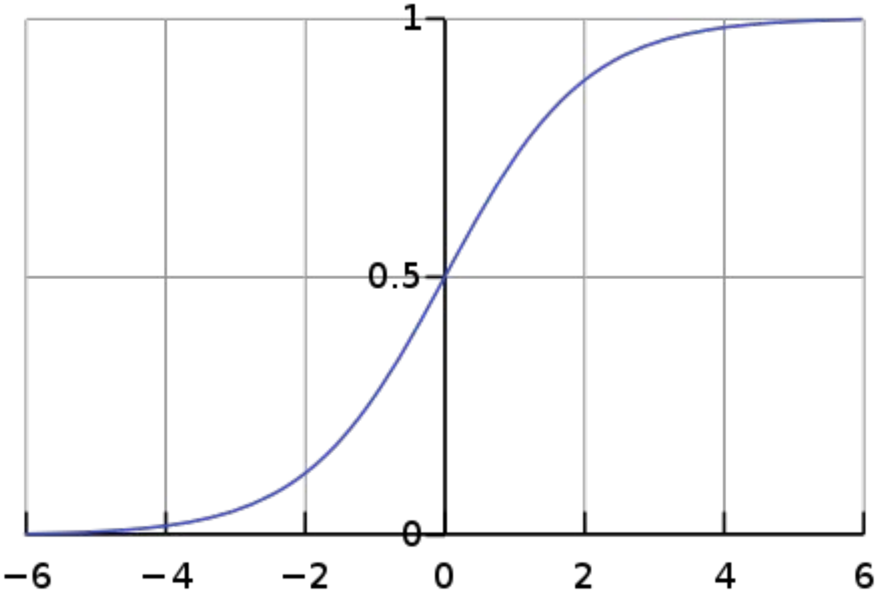
sigmoid

Logistic function

$S(x)$
$\sigma(x)$ $\in [0,1]$

$\mathbb{R} \mapsto [0,1]$

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} = 1 - S(-x).$$



Modelo de regresión Logística:

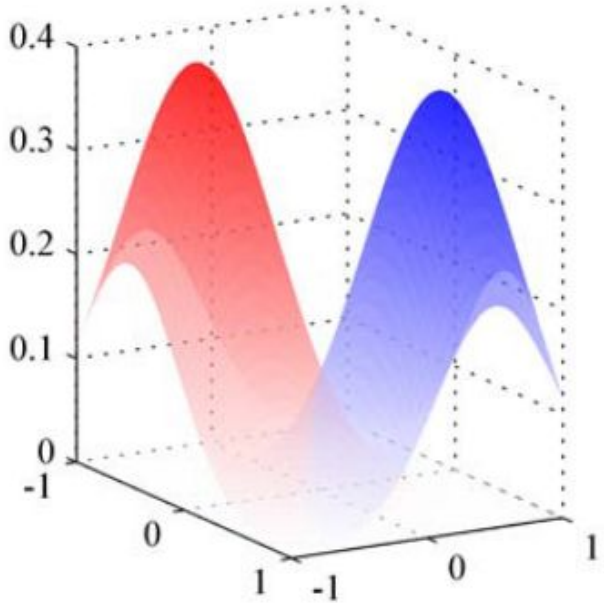
sean $y_1, \dots, y_n$ realizaciones de un proc Bernouille$(1, \pi_i)$.

Asumimos que existe una relación lineal entre x_i (datos) y el logit de la razón de prob:

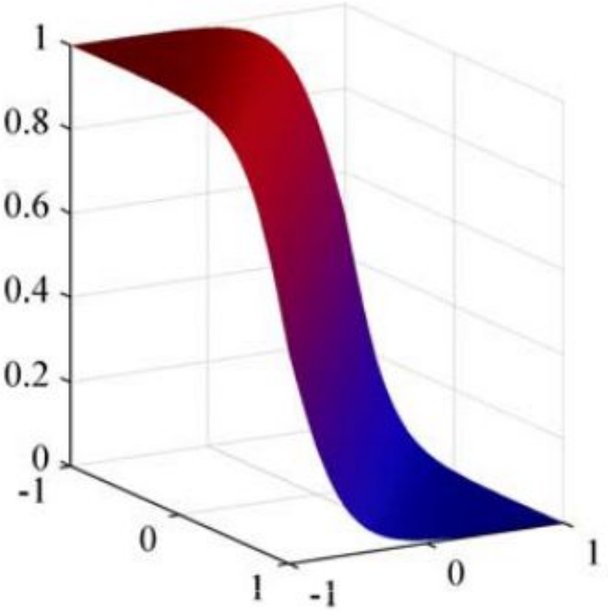
$$\text{logit}(\pi_i) = \beta_0 + \sum_{j=0}^{p} \beta_j x_{ij}$$

Este modelo queda definido como un modelo lineal generalizado con resp. binomial y fn. de enlace logit.

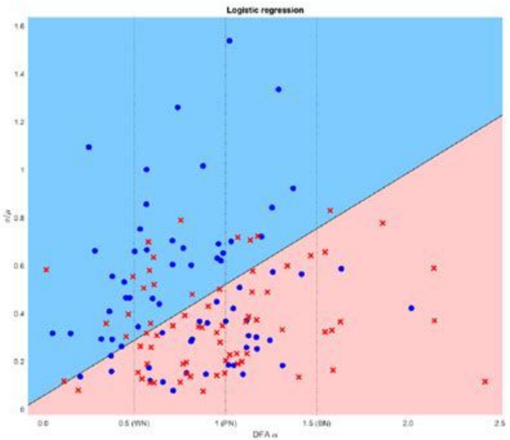
Regresión Logística



Class-conditional - P(x|Cn)



Posterior - P(Cn|x)



¿Cómo obtengo β?

quiero mapear $\sigma(z) = \sigma(w^t x)$ (propiedad de σ $\partial_a \sigma(a) = \sigma(a)(1-\sigma(a))$)

$$\partial_w \sigma(z) = \sigma(w^t x)(1 - \sigma(w^t x)) \quad ①$$

planteamos la fun. de verosim. : $\mathbb{P}(y/X) = \prod_{i=1}^{N} \hat{y}_i^{\,y_i} (1-\hat{y}_i)^{1-y_i} = \mathcal{L}$

N: cant muestras; $\hat{y}$: pred (prob)

y : GL

↳ $\hat{y}_i = \mathbb{P}(C/X)$

↳ $\ell = \log \mathcal{L}$

$$\ell = \sum_{i=1}^{N} \ln\left(\mathbb{P}_w(y_i = y_i \mid x_i = x_i)\right)$$ ⇝ buscamos maximizar ℓ

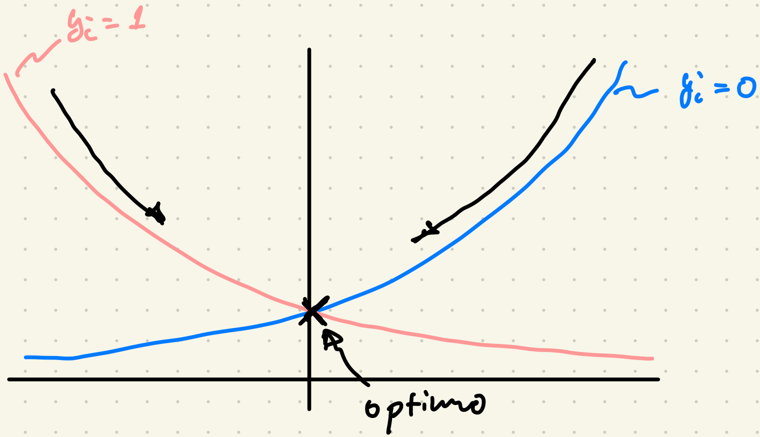
$$\max_w \sum_i \ln\left(\sigma(w^t x)^{y_i} \cdot (1-\sigma(w^t x))^{1-y_i}\right)$$

$$\max_w \sum_i y_i \underbrace{\ln \sigma(w^t x)}_{A} + (1-y_i) \underbrace{\ln(1-\sigma(w^t x))}_{B} \xrightarrow[\times(-1)]{} \min_w \sum_i -y_i A - (1-y_i) B$$

mi ℓ óptimo esta en $\min_{w} \sum_{i} -y_i (A) - (1-y_i) (B)$

puedo encontrar el minimo de ℓ usando la entropia binaria cruzada (binary crossentropy):

$$J(w) = \frac{1}{N} \sum_{i} -y_i (A) - (1-y_i) (B)$$



$\nabla_w J = 0$

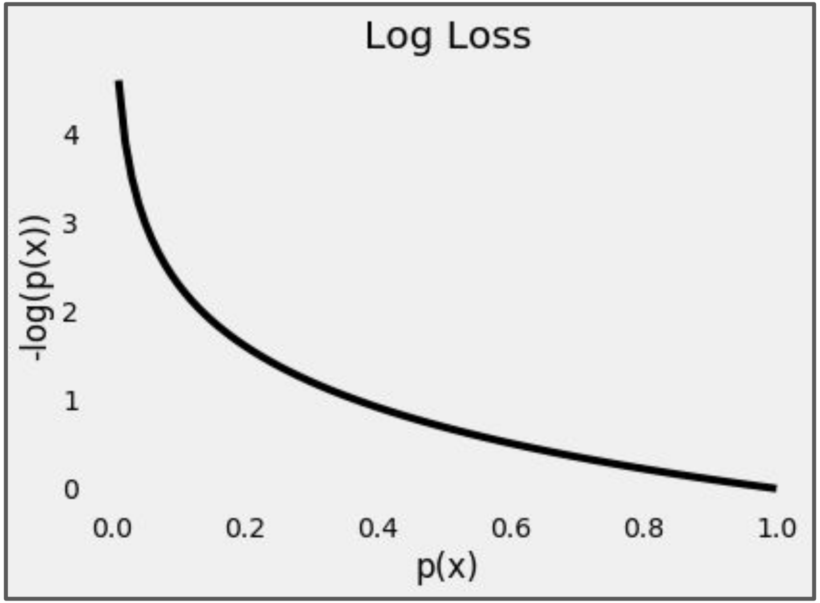
↳ GD

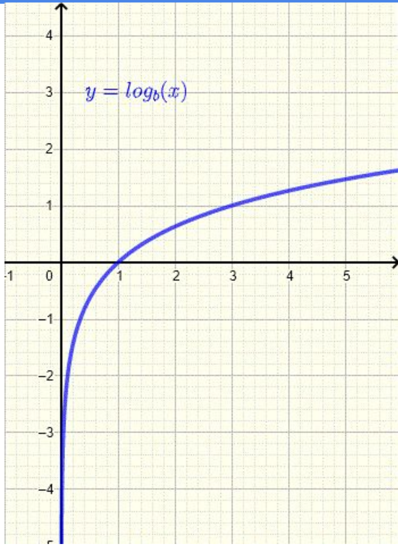
SGD

GD MB

Función de costo - Binary cross entropy

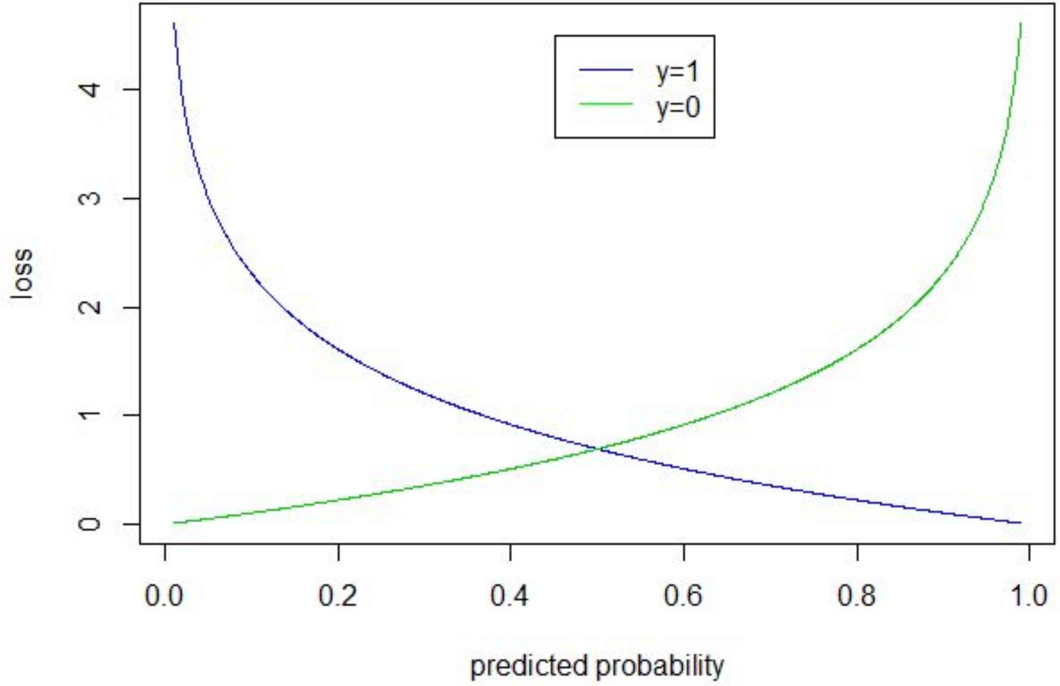
$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$





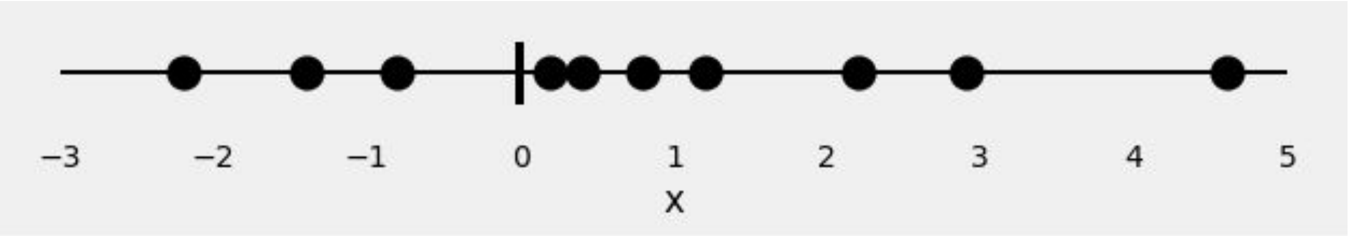
Función de costo - Binary cross entropy

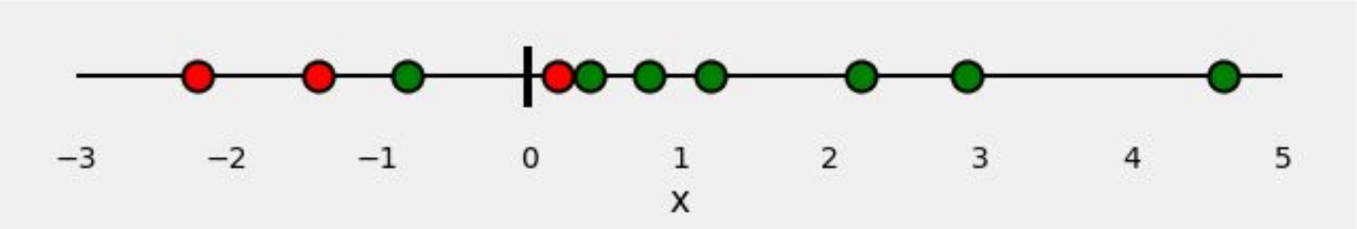
$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} \boxed{y_i \cdot log(p(y_i))} + \boxed{(1 - y_i) \cdot log(1 - p(y_i))}$$



.UBAfiuba FACULTAD DE INGENIERÍA

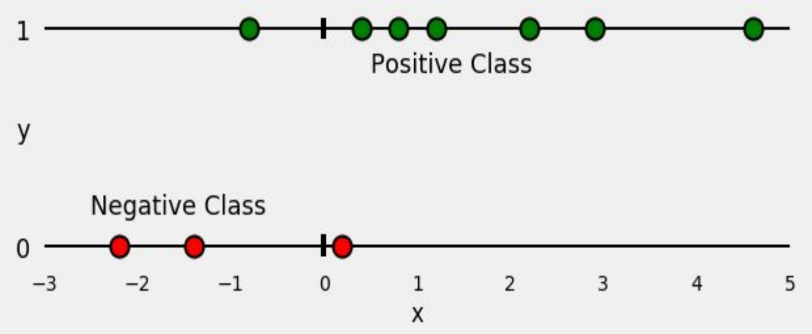
Regresión Logística

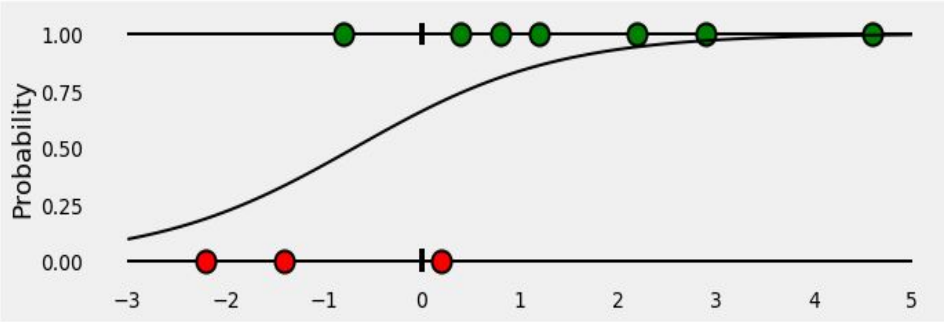




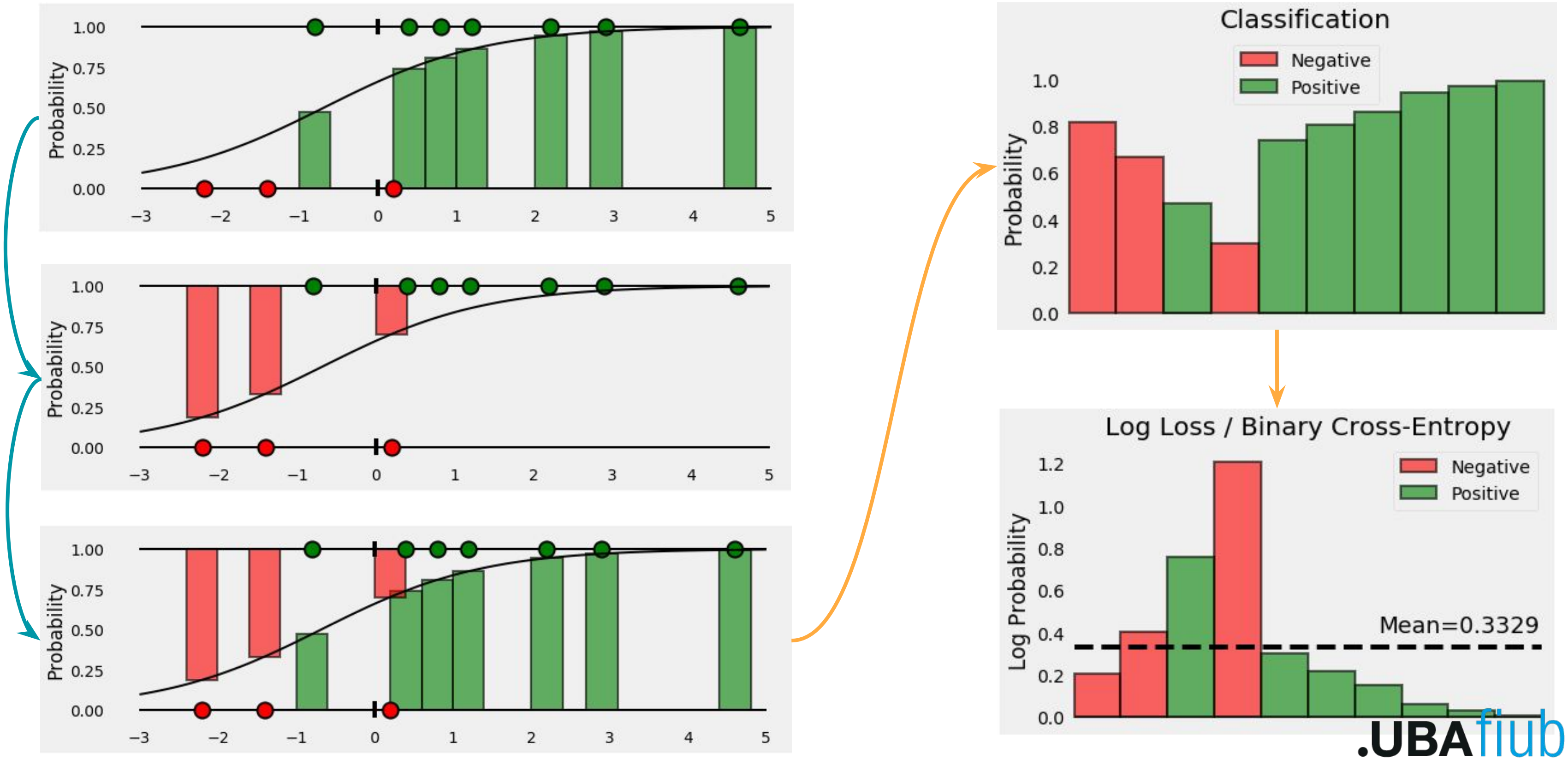
1: Verde, 0: Rojo

Regresión Logística



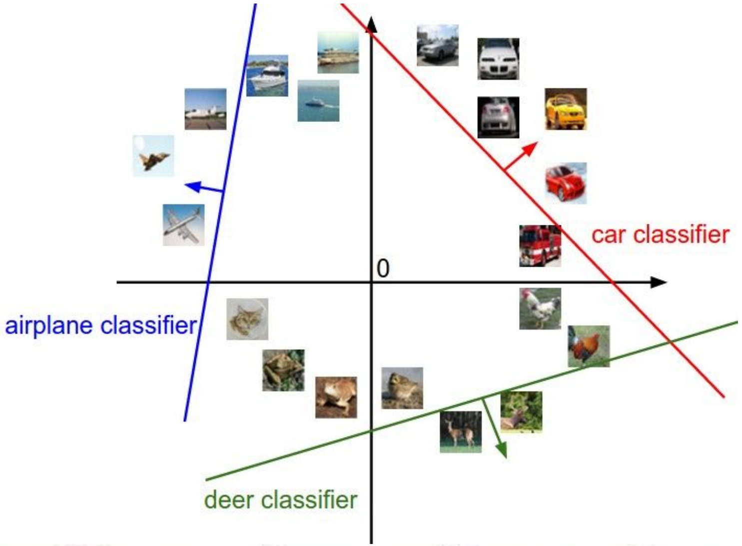


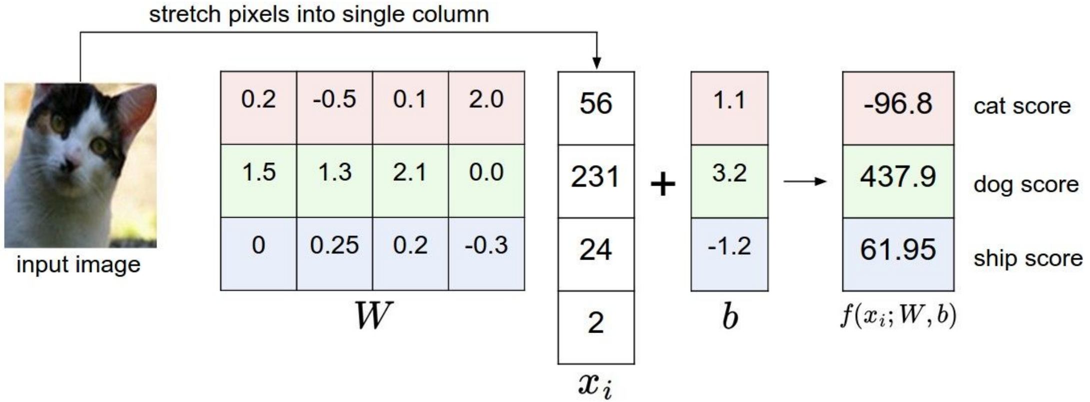
Regresión Logística



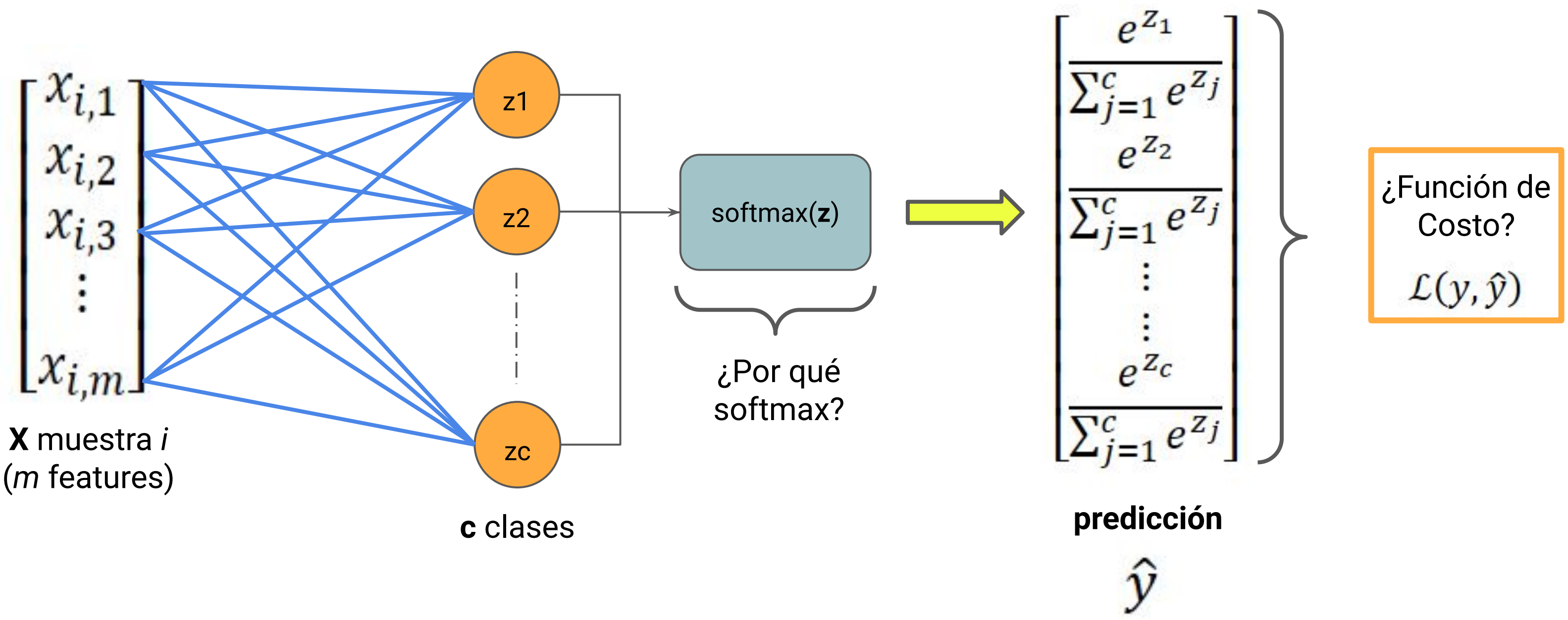
Clasificación multiclase

Clasificación Multiclase - Motivación





Softmax



$x_{i,1}$
$x_{i,2}$
$x_{i,3}$
$x_{i,m}$
X muestra *i*
(*m* features)
z1
z2
zc
c clases
softmax(**z**)
¿Por qué
softmax?
$\frac{e^{z_1}}{\sum_{j=1}^{c} e^{z_j}}$
$\frac{e^{z_2}}{\sum_{j=1}^{c} e^{z_j}}$
$\frac{e^{z_c}}{\sum_{j=1}^{c} e^{z_j}}$
predicción
$\hat{y}$
¿Función de
Costo?
$\mathcal{L}(y, \hat{y})$

Softmax

$$P(y_i \mid x_i; W) = \frac{e^{f_{y_i}}}{\sum_j e^{f_j}}$$

$$\underbrace{\frac{e^{f_{y_i}}}{\sum_j e^{f_j}} = \frac{Ce^{f_{y_i}}}{C\sum_j e^{f_j}} = \frac{e^{f_{y_i}+\log C}}{\sum_j e^{f_j+\log C}}}_{q(x)}$$

$$H(p,q) = -\sum_x p(x) \log q(x)$$

1 [Softmax Forma Gráfica]

2 [Softmax Visualización 3D]

Softmax

Derivación Softmax

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}$$

$$\frac{\partial p_i}{\partial z_k} = \frac{\partial \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}}{\partial z_k}$$

$$\frac{\partial p_i}{\partial z_k} = p_i(\delta_{ik} - p_k) \qquad \delta ik = \begin{cases} 1, i = k \\ 0, i \neq k \end{cases}$$

Derivación Cross-Entropy

$$L = -\sum_i y_i log(p_i)$$

$$\frac{\partial L}{\partial z_i} = -\sum_j y_j \frac{\partial log(p_j)}{\partial z_i}$$

$$= -\sum_j y_j \frac{\partial log(p_j)}{\partial p_j} \times \frac{\partial p_j}{\partial z_i}$$

$$= -\sum_j y_j \frac{1}{p_j} \times \frac{\partial p_j}{\partial z_i}$$

$$\frac{\partial L}{\partial z_i} = -y_i(1 - p_i) - \sum_{j \neq i} y_j \frac{1}{p_j}(-p_j.p_i)$$

$$= -y_i(1 - p_i) + \sum_{j \neq i} y_j.p_i$$

$$= p_i\left(y_i + \sum_{j \neq i} y_j\right) - y_i$$

$$\frac{\partial L}{\partial z_i} = p_i - y_i$$

$$\frac{\partial z_i}{\partial W} = x_i$$

$$\frac{\partial L}{\partial W} = \sum_{i=1}^{N} (p_i - y_i)\, x_i$$

Usar gradiente descendente para actualizar W !!!

Bibliografía

- The Elements of Statistical Learning | Trevor Hastie | Springer
- An Introduction to Statistical Learning | Gareth James | Springer
- Deep Learning | Ian Goodfellow | https://www.deeplearningbook.org/
- Mathematics for Machine Learning | Deisenroth, Faisal, Ong
- Artificial Intelligence, A Modern Approach | Stuart J. Russell, Peter Norvig
- Understanding binary cross-entropy: a visual explanation | Daniel Godoy
- Visual Information Theory | Link
- https://cs231n.github.io/
- Classification and Loss Evaluation-Softmax and Cross Entropy Loss | Paras Dahal