Introducción a la Inteligencia Artificial
Facultad de Ingeniería
Universidad de Buenos Aires

Índice

Clase 5

IA — ML

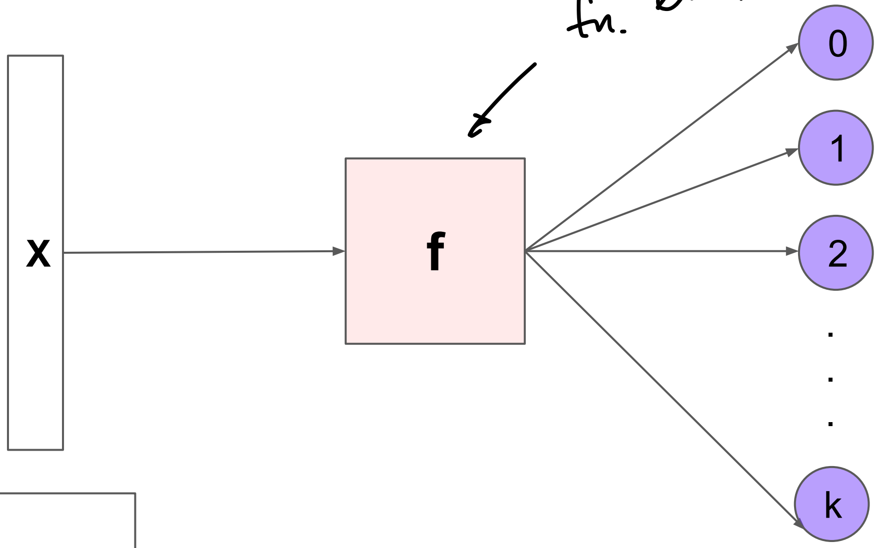
- ML — Tareas a resolver
  - regresión → $x \in \mathbb{R}^{n\times p}$, $y \in \mathbb{R}^{r}$
  - clasificación → $x \in \mathbb{R}^{n\times p}$, $y \in \mathbb{K}$

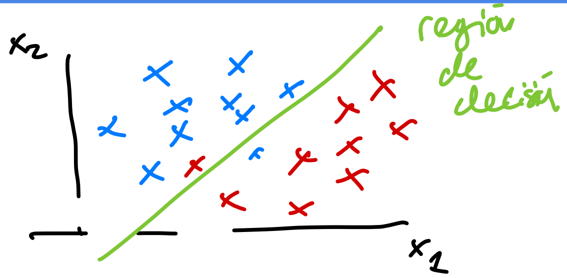
    $\mathbb{K} = [\pi_i \,/\, i \in 1,r]$
- ML → Modelos
  - tipo de solución
    - → cerrada
    - → numérica
  - tipo de problema
    - → supervisado
    - → no supervisado

Clasificación

considero $X \in D \subset \mathbb{R}^{n \times p}$

fn. etiquetadora



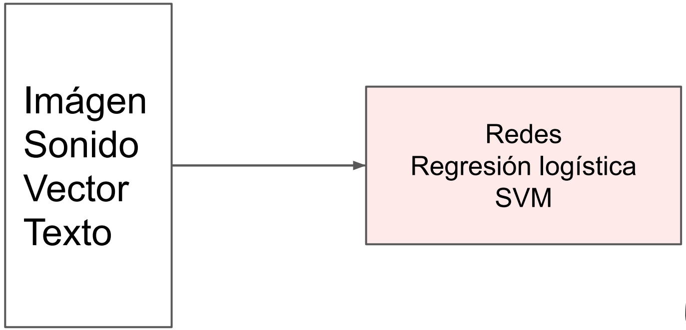


función etiquetadora

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

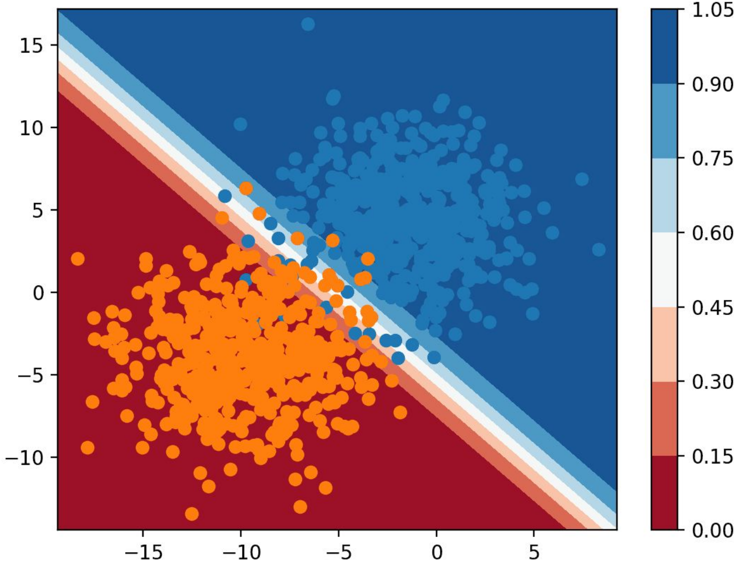
Acá estamos haciendo un supuesto SUPER importante, Estamos diciendo que Y es separable:

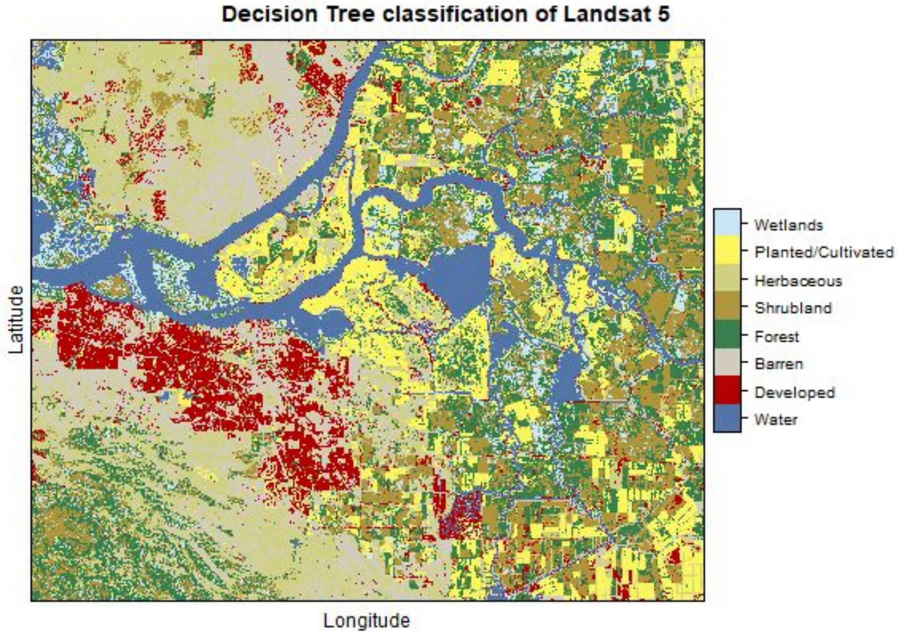
$$\forall\, x_{ij}\ \exists\, y_i \,/\, y_i \in K \wedge f(y_i)\ \exists!$$



Esto nos dice que en el espacio de features $\exists$ una región (hiperplano) que separa los datos

Clasificación



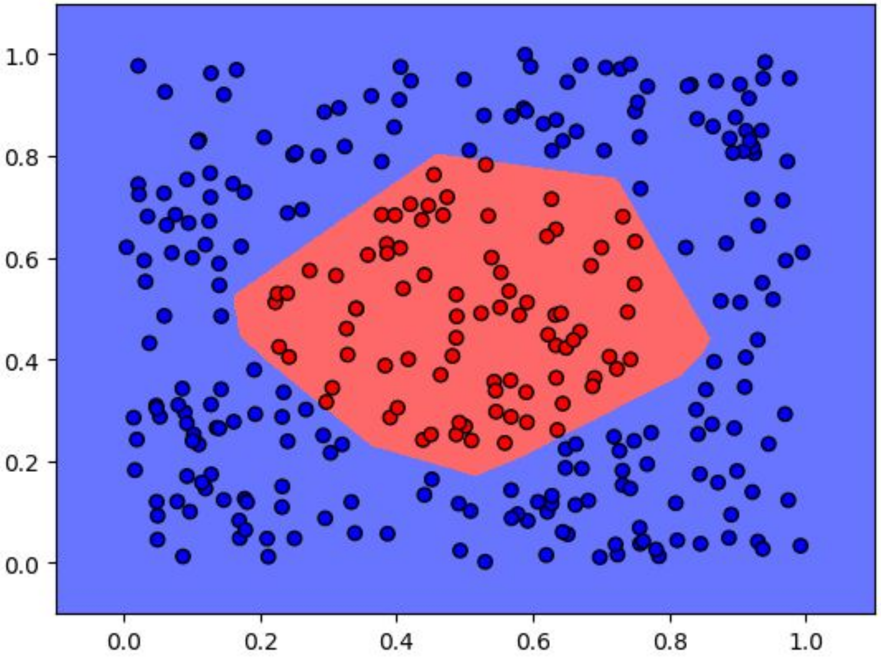


Clasificación binaria

$$f(x_i) = \begin{cases} 1 & \text{si } y_i \in C_1 \\ 0 & \text{o.w.} \end{cases}$$

Clasificación Binaria - Ejemplos

- Detección de fraudes → es fraude / es legal
- Diagnóstico médico → es maligno / es benigno
- Detección de spam
- Sentiment Analysis
- Detección de objetos
- Outliers



Clasificación Binaria - Diagnóstico médico

bloque conv. buffer cabecera de clasif

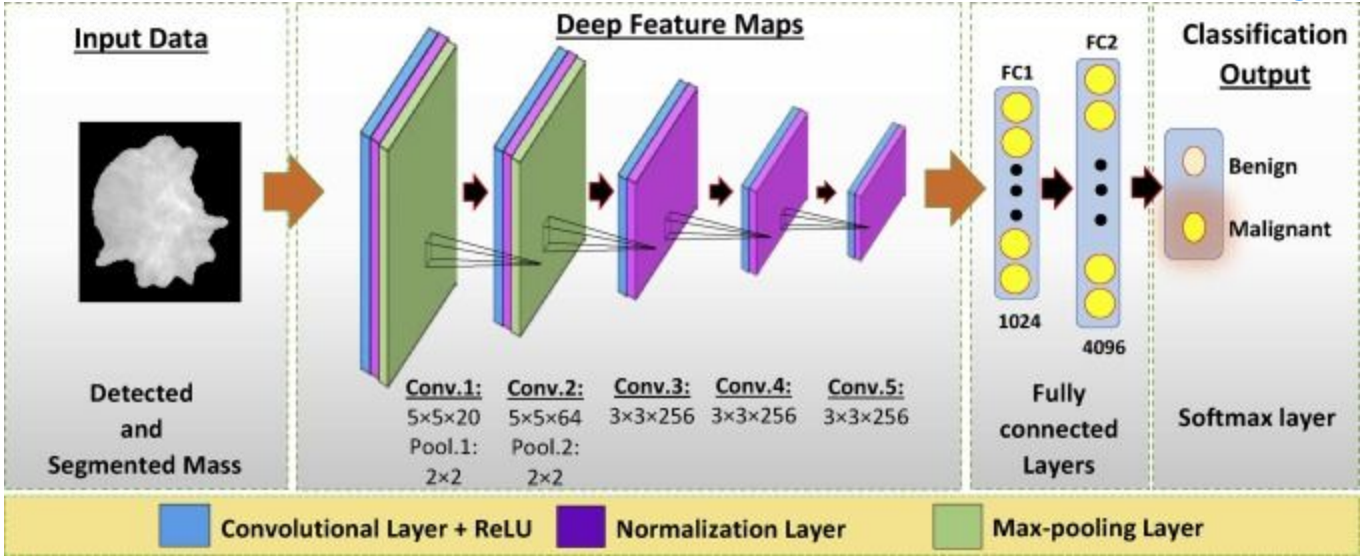
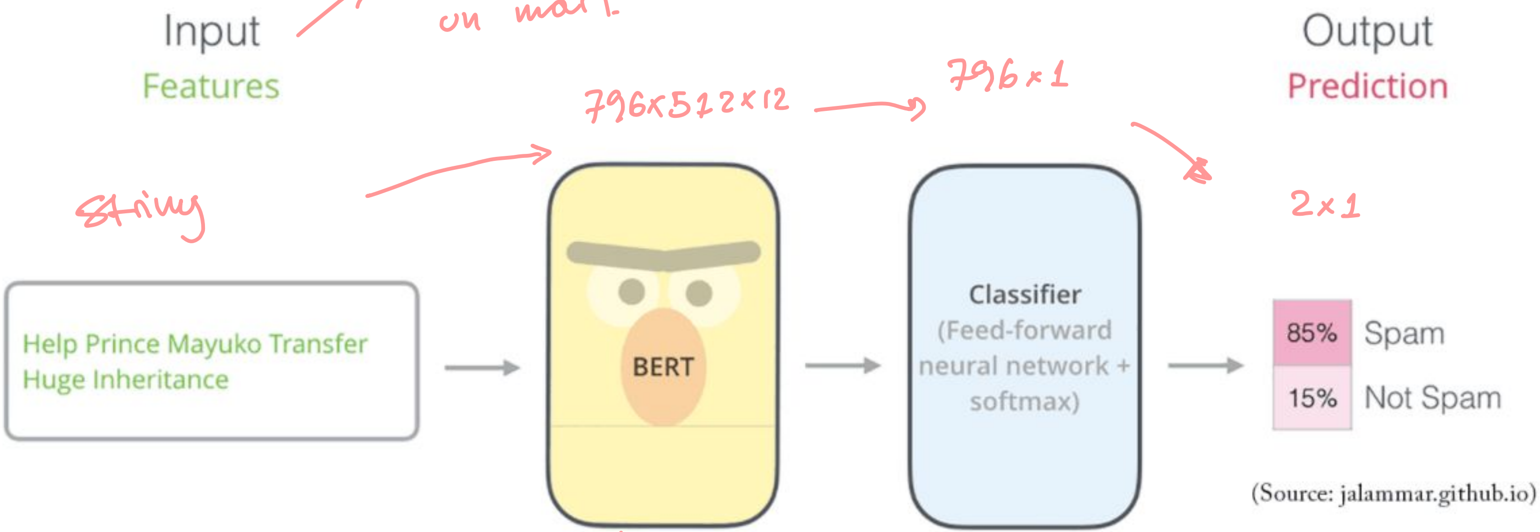


Imagen de: "A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification"

Clasificación Binaria - Spam detection

en este caso es un mail



string

796x512x12

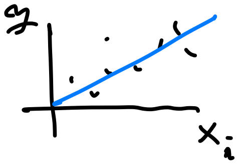
796x1

2x1

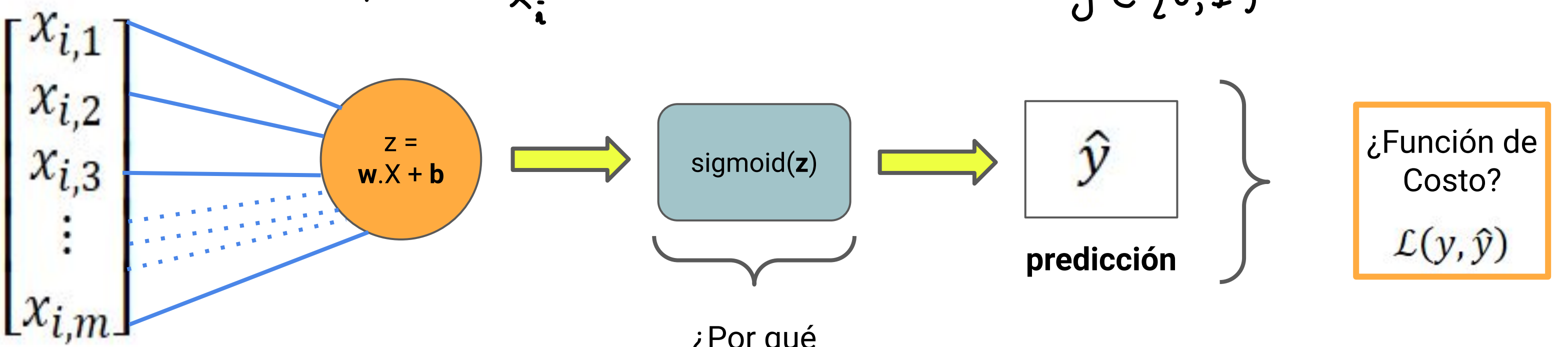
12 bloques transformes apilados con una cabecera de pooling para sacar el embedding

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

Regresión Logística



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



X muestra *i*
(*m* features)

¿Por qué sigmoid?

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

$z_i \in \mathbb{R}$

$\bar{W}_i \in \mathbb{R}^{1 \times m}$

$b \in \mathbb{R}$

$\phi(z)$

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

función de squashing

Métricas: en clasif. no podemos usar lo que veníamos usando MSE(gato, perro)? ⇝ en clasif. usamos la matriz de confusión.

real \ Pred	T	F
T	TP	FN
F	FP	TN

← prediction

GL*

$$y = \mathbb{I}_{c=K} = \begin{cases} 1 & \text{si } y_i = K \\ 0 & \text{si } y_i \neq K \end{cases}$$

* GL: Golden Label
Ground truth
Real values

- Accuracy $= \dfrac{TP + TN}{\#\text{muestras}} \in [0,1]$
- Specificity (TN rate) $= \dfrac{TN}{TN + FP}$
- precision $= \dfrac{TP}{TP+FP}$
- recall $= \dfrac{TP}{TP+FN}$
- $F_1 = 2 \cdot \dfrac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \dfrac{TP}{TP + \frac{1}{2}(FP+FN)}$
- $F_\beta = (1+\beta)^2 \dfrac{\text{prec.}\ \text{recall}}{\beta^2\,\text{prec} + \text{recall}}$ $(F_{0,25})$

Formas de clasificar:

+ Definimos un proc. de scoring → como resolvemos una clasif.

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

1. Modelos generativos: modelar las distrib. de I/O ⇝ generar la salida (permite generar información sintética) → GAN's

2. Modelos discriminantes: plantear $\mathbb{P}(C_k \mid \bar{\bar{x}})$ ⇝ utilizar métodos de inferencia (bayesiana) para estimar $\mathbb{P}$.

3. modelos de función discriminadora: buscamos $f: \mathbb{R}^{n \times m} \longmapsto \{c_1, \dots, c_k\}$

↳ Regresión logística (la usamos porque queremos explotar lo conocido).

partimos de suponer $y \sim \text{Bernouille}(1, \pi_i) \rightarrow \begin{cases} E(y) = \pi_i \\ Var(y) = \pi_i \cdot (1-\pi_i) \end{cases}$

me gustaría poder obtener algo así:

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

$\pi_i \in [0,1]$, pero $\bar{\bar{X}}^t \cdot \bar{\beta} \in \mathbb{R}$ $\Rightarrow$ tengo que plantear una fn. de mapeo.

buscamos $f: [0,1] \mapsto \mathbb{R}_{\geq 0} \rightsquigarrow odd_i = \dfrac{\pi_i}{1-\pi_i}$

razón de probabilidad

$$odd_i = \frac{P(A)}{P(\bar{A})}$$

Vamos a tomar el log de odds, esto me permite definir una función biyectiva $\mathbb{R}_{\geq 0} \mapsto \mathbb{R}$. Esta transf. se llama logit (log odds ratio):

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

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

fn. sigmoide
fn. logística

Logistic function

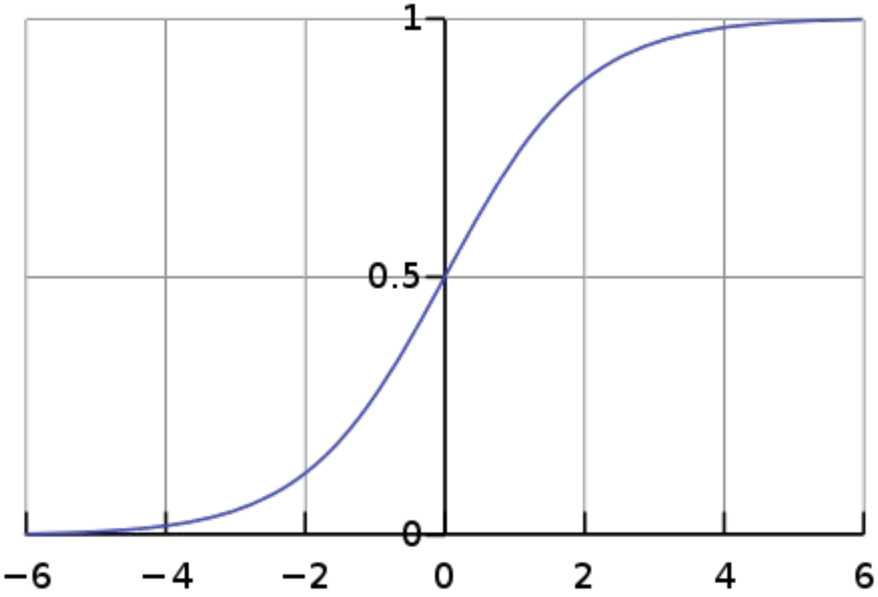
$S(x) = \sigma(x) \in [0,1]$

$\sigma(x)$

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

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

$\sigma(x, a)$



Modelo de regresión logística:

Sean $y_1, \dots, y_n$ realizaciones de un proc. Bernoulli $(1, \pi_i)$, asumimos que existe una relación lineal entre x_i (datos) y el logit de la razón de prob. de π_i:

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

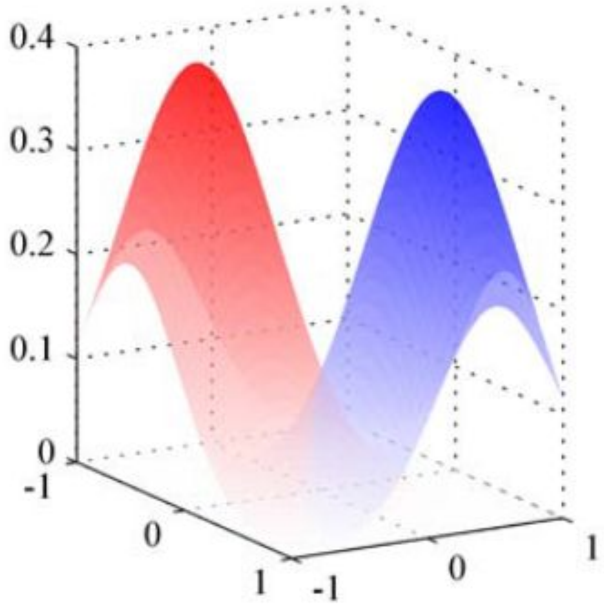
Este modelo es parte de la familia de modelos lineales generalizados

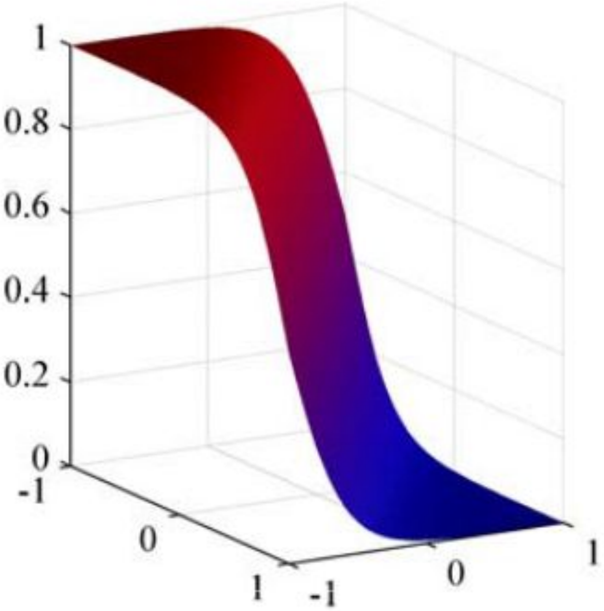
→ Es equivalente a decir modelo lineal generalizado con respuesta binomial y fn. de enlace logit.

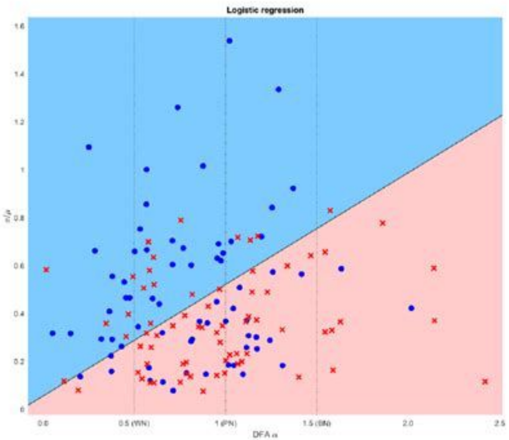
$$g(E(Y/X)) = \beta_0 + \sum_i \beta_i x_i$$

$y \sim f$; f distrib. de la respuesta $\wedge$ g es la función de linceo

Regresión Logística







Class-conditional - P(x|Cn)

Posterior - P(Cn|x)

¿Como obtenemos β?

queremos mapear $\sigma(z) = \sigma(w^t x)$ prop útil $\partial_a \sigma(a) = \sigma(a)(1-\sigma(a))$

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

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

$\hat{y}_i = \mathbb{P}(c/x)$

- N: cant. de muestras
- y : golden label
- $\hat{y}$: pred (prob)

$$l = \sum_{i=1}^{N} \ln\left(\mathbb{P}_w(y = y_i / X = x_i)\right) \rightsquigarrow \text{buscamos maximizar } l$$

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

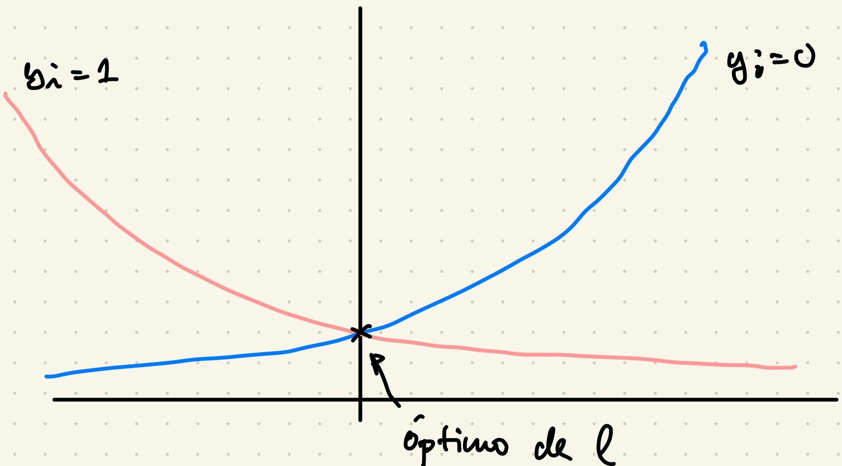
$$\max_w \sum_i y_i \underbrace{\ln \sigma(w^t x)}_{A} + (1-y_i) \underbrace{\ln\left(1 - \sigma(w^t x)\right)}_{B}$$

multiplicar $\times (-1)$:

$$\min_{w} \sum_{i} -y_i \, A - (1-y_i) \, B$$ $\rightsquigarrow$ si minimizo ℓ encuentro el óptimo

podemos buscar el mínimo usando la entropia binaria cruzada (binary crossentropy):

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



$\nabla J_w = 0$

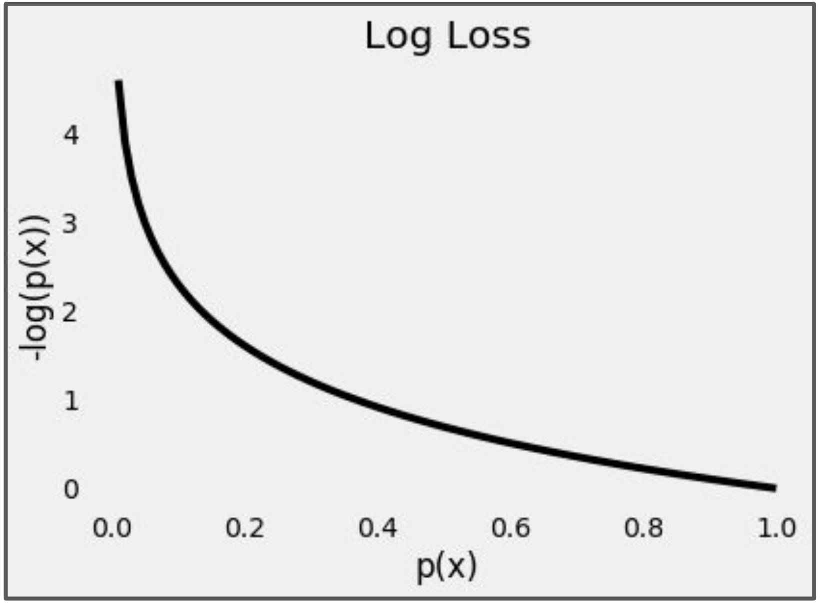
↳ GD

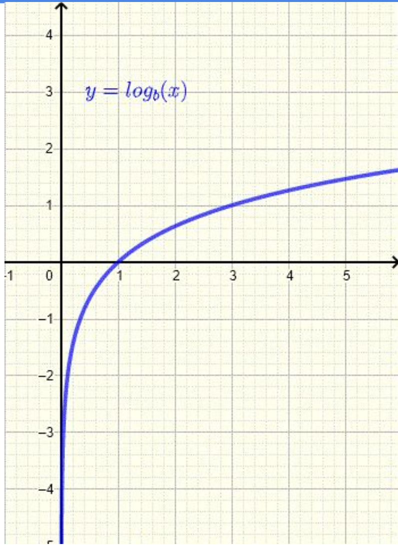
SGD

GDMB

Función de costo - Binary cross entropy

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





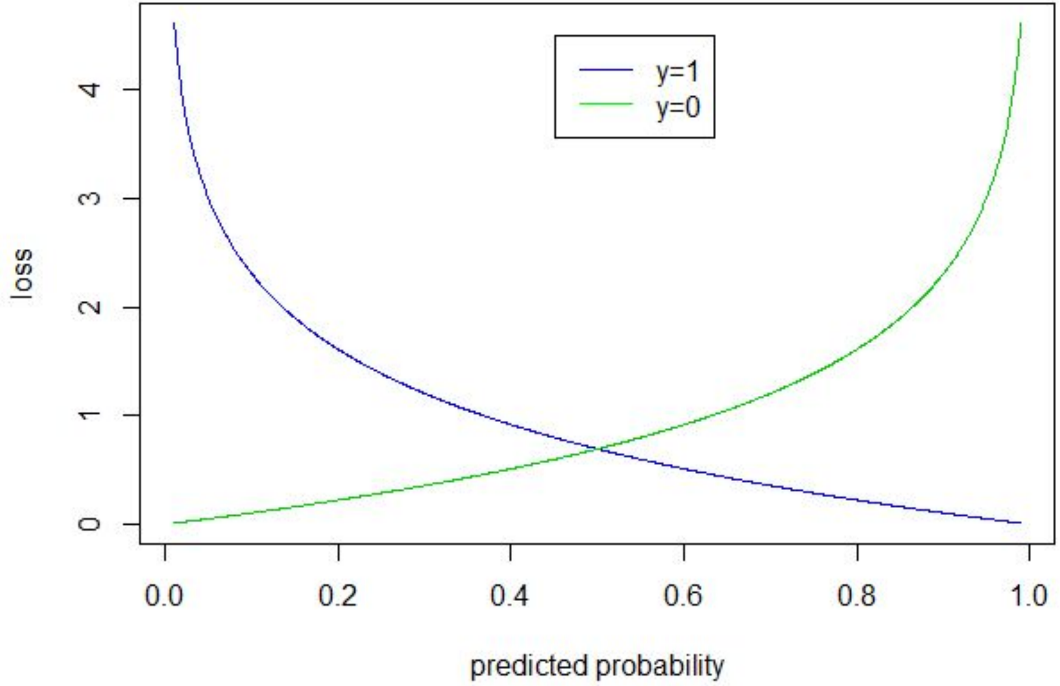
Función de costo - Binary cross entropy

$\log p(y_i) = \log(\sigma(w^t x))$

lo puedo calcular

dato

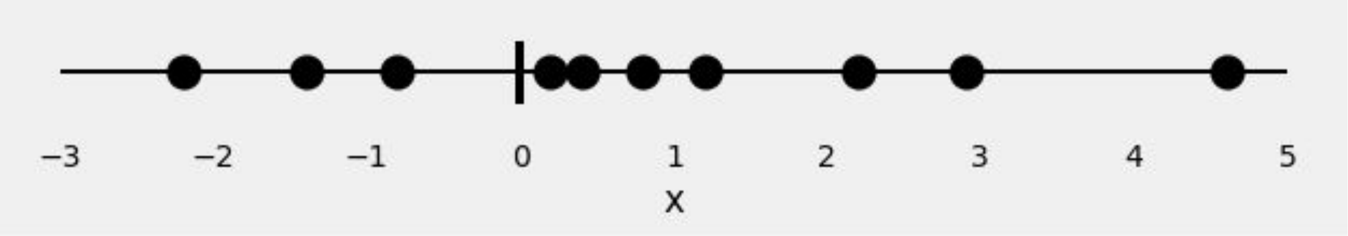
$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} \boxed{y_i \cdot log(p(y_i))} + \boxed{(1 - y_i) \cdot log(1 - p(y_i))}$$

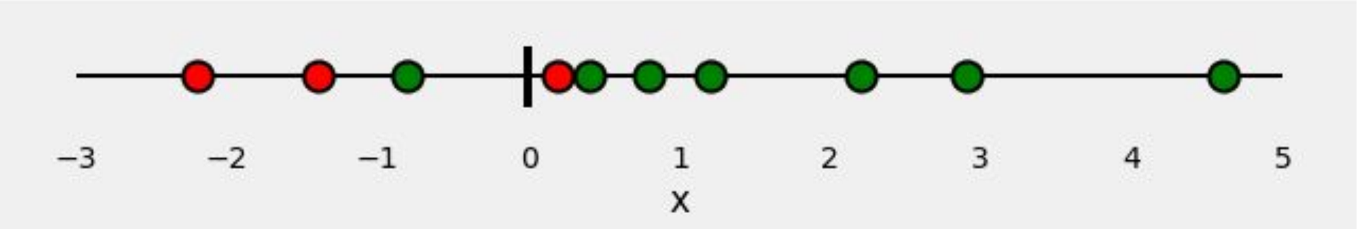


$w_{n+1} = w_n + \alpha \nabla w$

∇w ?

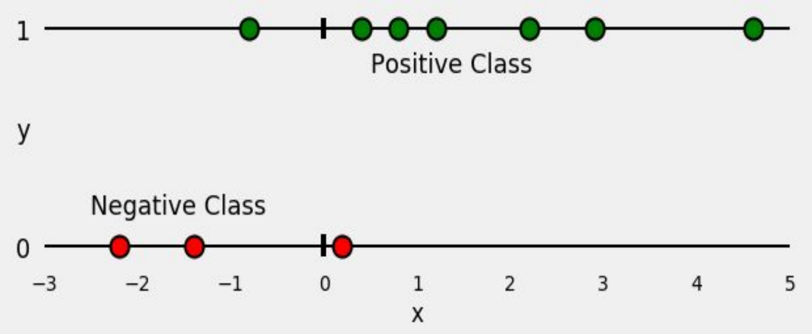
Regresión Logística

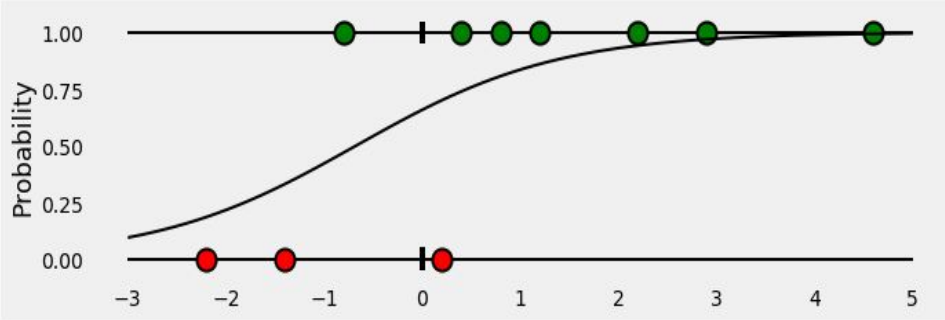




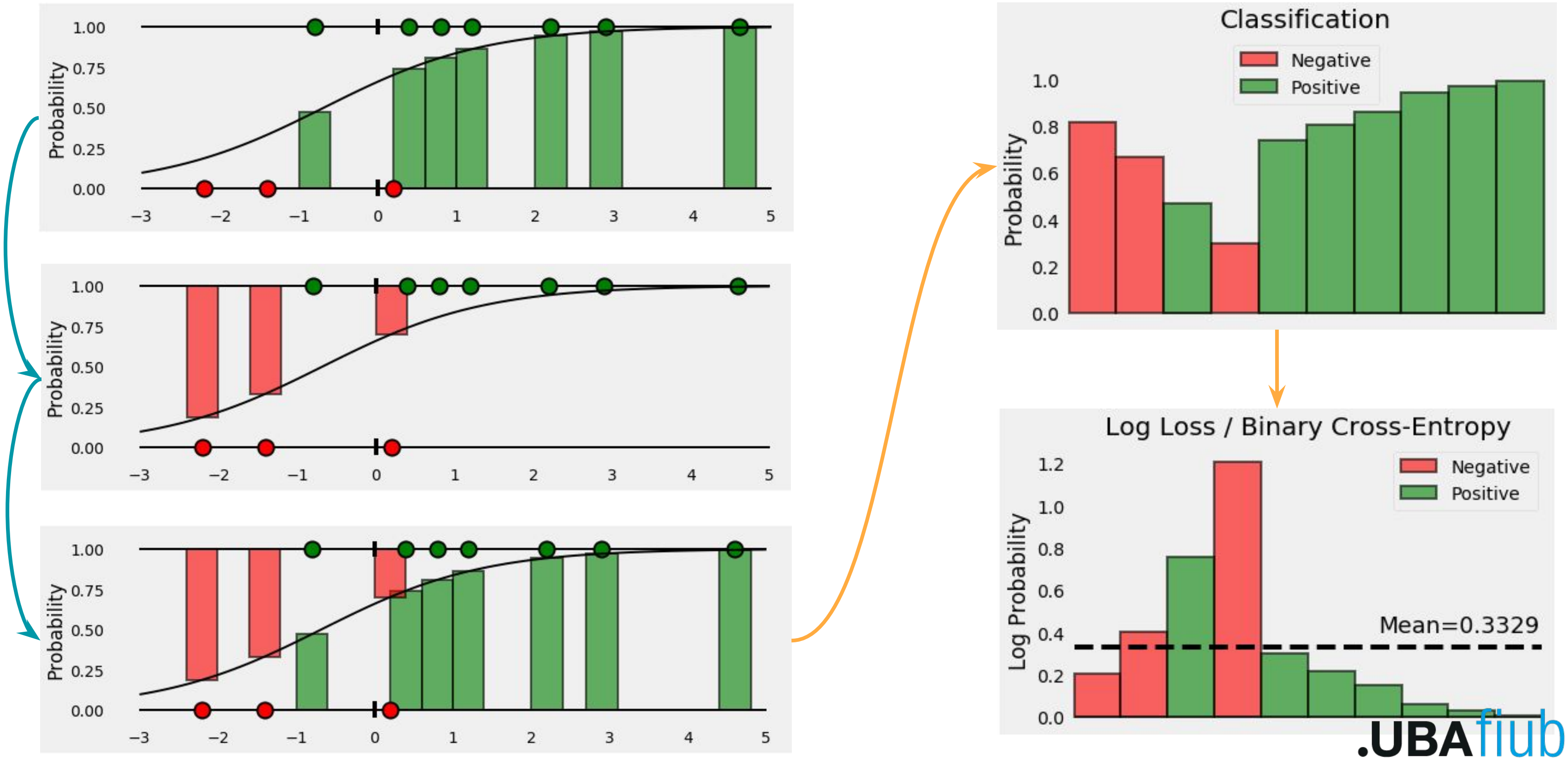
1: Verde, 0: Rojo

Regresión Logística

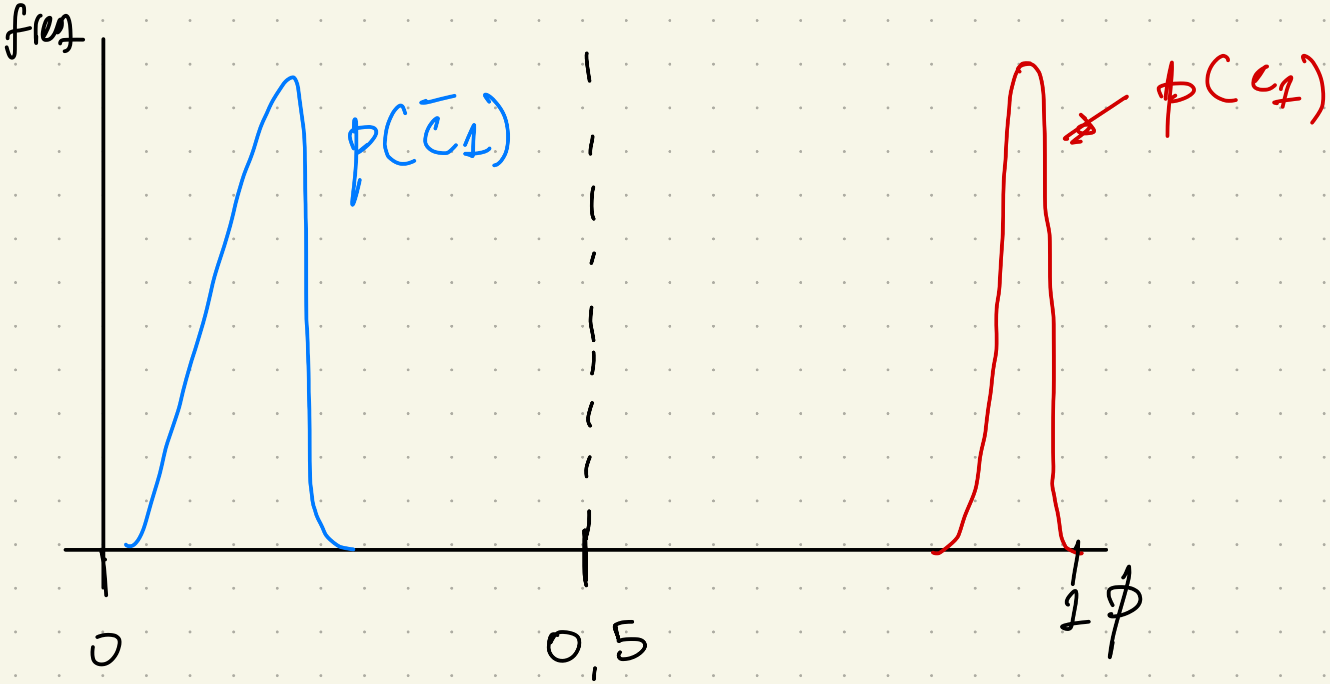




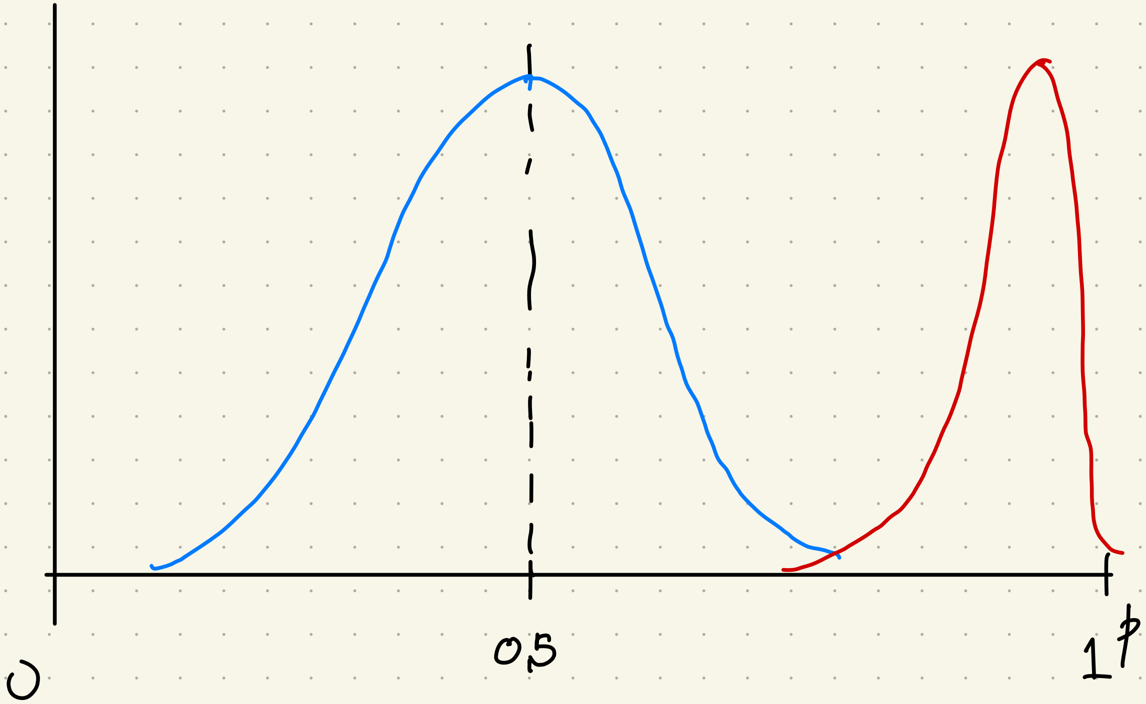
Regresión Logística



.UBA fiuba
FACULTAD DE INGENIERÍA



$$\hat{y} = \begin{cases} 1 & si\ p \geq 0,5 \\ 0 & o.w \end{cases}$$

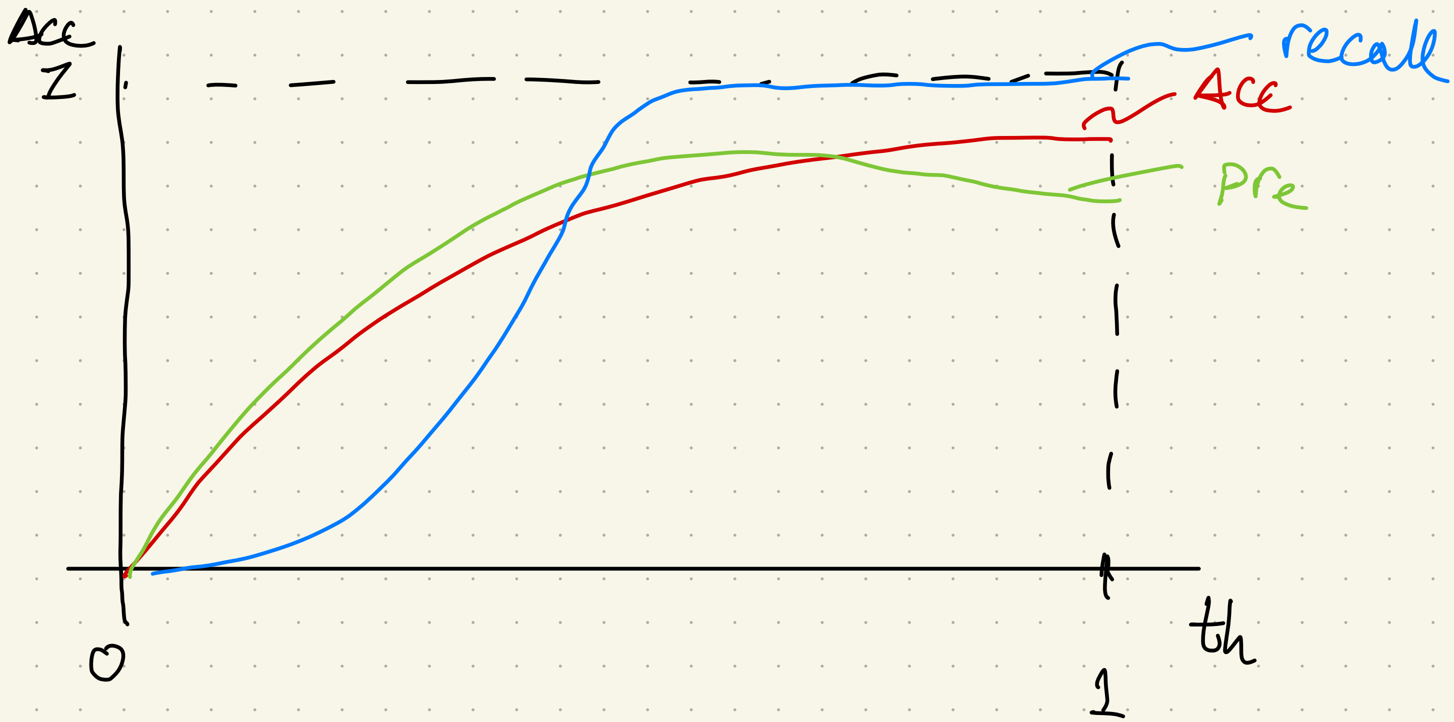


$$y = \begin{cases} 1 & si\ p > th \\ 0 & o.w \end{cases}$$

donde th lo voy a elegir de acuerdo a mis métricas

precisión (th) acc(th)

recall (th)



buscamos un threshold que maximice por ej. F_1.

y teniendo fijado el th. calculo la matriz de confusión final.

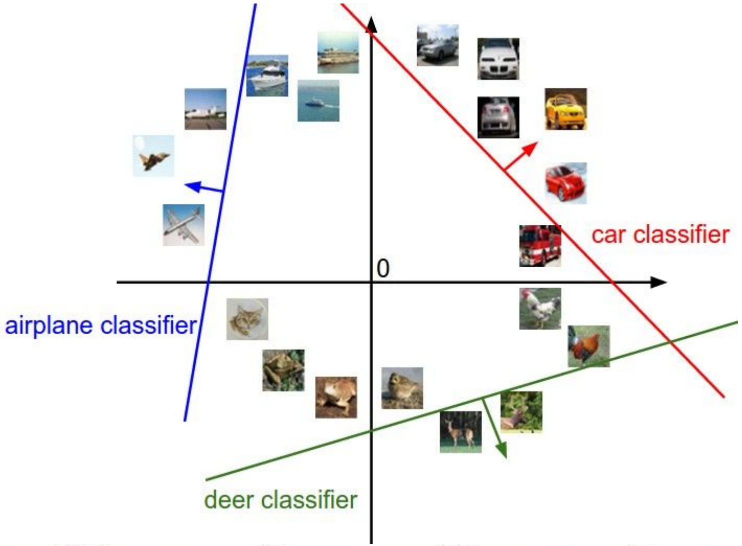
nota: cuando las prob $p(y=0)$ y $p(y=1)$ tienen mucho solapamiento. A veces conviene tener un tercer estado.

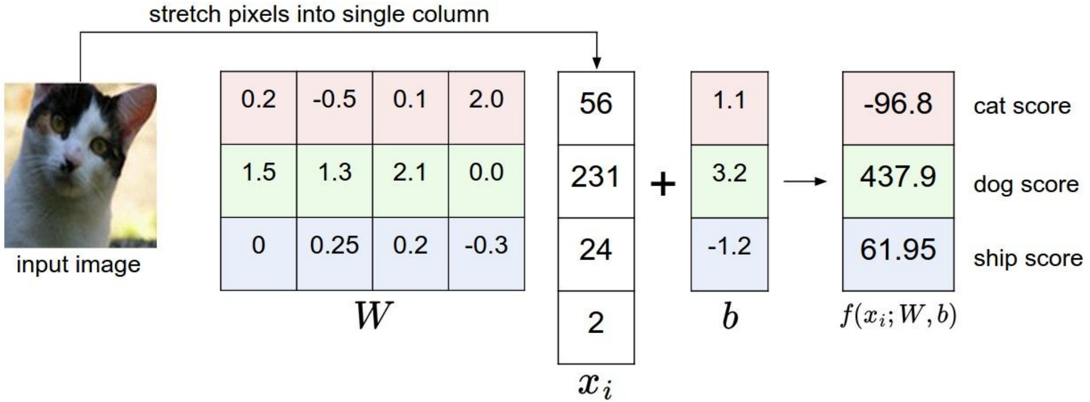
$$y \in [\text{True}, \text{False}, \text{NS/NC}]$$

Clasificación multiclase

$$f = \mathbb{I}_{(y=\pi_i)}$$

Clasificación Multiclase - Motivación

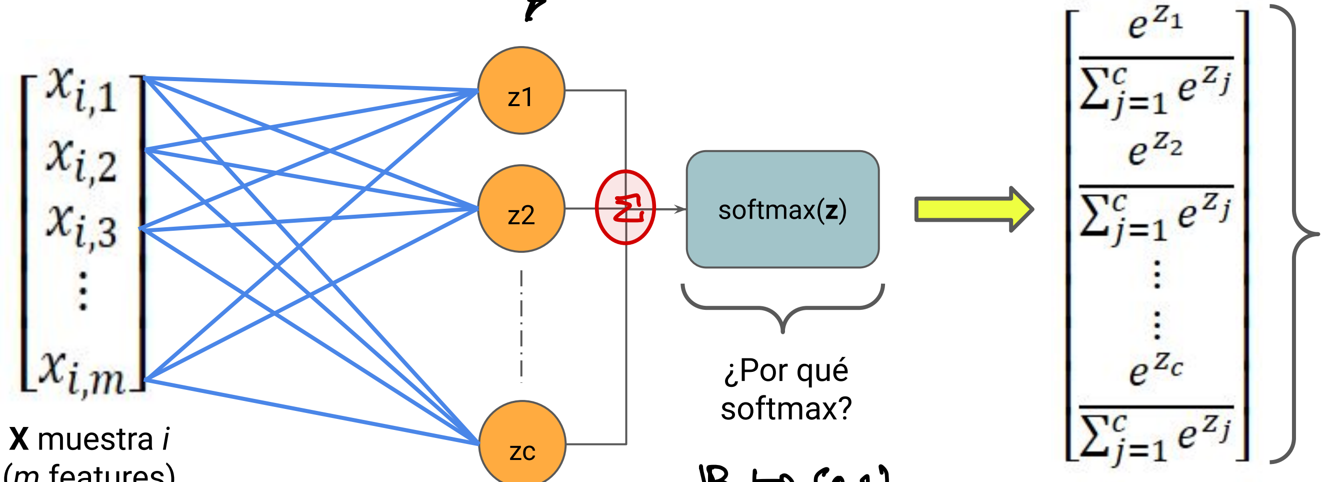




Softmax

Softmax

$z_j \in \mathbb{R}$ $\quad$ $z_j = w_j x_i + b$



X muestra *i*
(*m* features)

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

c clases

$j \in [1, c]$

¿Por qué softmax?

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

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

predicción

$\hat{y}$

¿Función de Costo?

$\mathcal{L}(y, \hat{y})$

matriz de conf. multilabel:

real \ pred	C_1	C_2	...	C_c
C_1	TP_{c_1}	$M_{c_2c_1}$	...	
C_2	$M_{c_1c_2}$	TP_{c_2}	...	
$\vdots$	$\vdots$	$\vdots$	$\ddots$	
C_c				TP_{c_c}

las métricas ahora son por clase:

Acc_{c_1}

Acc_{c_2}

$\vdots$

Acc_{c_c}

Además tenemos el overall del modelo:

$Acc_{mean} \rightarrow$ promedio

$Acc_{macro} \rightarrow$ " pesado por clase

$Acc_{micro} \rightarrow$ " " " soporte

Softmax

j: es la clase

c clases => $j \in \{1, \dots, c\}$

$$odds = \frac{P(A)}{P(\bar{A})}$$

$$\underbrace{P(y_i \mid x_i; W) = \frac{e^{f_{y_i}}}{\sum_j e^{f_j}} \qquad \frac{e^{f_{y_i}}}{\sum_j e^{f_j}} = \frac{Ce^{f_{y_i}}}{C\sum_j e^{f_j}} = \frac{e^{f_{y_i}+\log C}}{\sum_j e^{f_j+\log C}}}_{q(x)}$$

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

1 Softmax Forma Gráfica

2 Softmax Visualización 3D

los pesos w_{ij} se obtienen numéricamente con GD y amigos.

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 !!!

.UBA fiuba FACULTAD DE INGENIERÍA

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