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

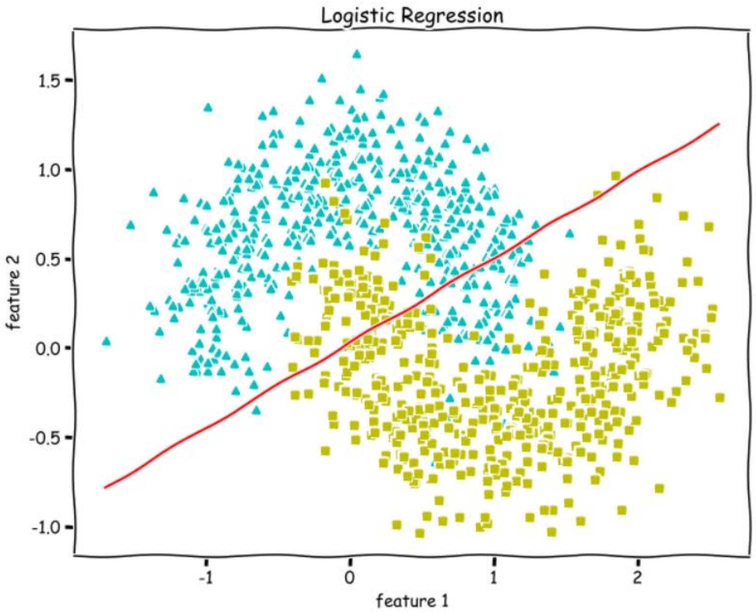
Resumen

IA → ML

- ML → Tareas
  - → Regresión
  - → clasificación
- ML → Metricas → performance
  - → underfitting
  - → overfitting
- ML → Preprocesamiento de datos → Adaptar la entrada a lo que pide el modelo
- ML → modelos
  - → solución
    - → cerrada → Regresión lineal
    - → sol. numérica → GD
  - → tipo de problema
    - → supervisado
    - → no supervisado

Clasificación Binaria - Motivación

- Credit Card Fraudulent Transaction Detection
- Medical Diagnosis
- Spam Detection
- Sentiment Analysis
- Binary Image Classification



Clasificación: queremos mapear $X \in \mathbb{R}^{m \times n}$ en K clases

① Clases separables: cada input tiene un único output.
En este régimen, el espacio se divide en regiones de decisión.

modelos → lineales → Reg. logística, SVM, LDA
↳ no lineales → KNN

$\mathbb{D} = \{(\bar{x}_1, y_1), \ldots, (\bar{x}_n, y_n)\}$, $y_i = (0, \ldots, 1, \ldots, 0)$ vector one-hot

$$y_{ij} = \begin{cases} 1 & \text{si } \bar{x}_i \text{ pertenece a } j \\ 0 & \text{o.w.} \end{cases}$$

Métricas

Matriz de confusión:

		Preal	
		T	F
GL	T	TP	FN
	F	FP	TN

$$\text{Accuracy} = \frac{TP + TN}{\#P + \#N} \in [0,1]$$

$$\text{specificity (TN rate)} = \frac{TN}{TN + FP}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{Recall (TP rate)} = \frac{TP}{P}$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

$$F_\beta = (1+\beta^2) \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

GL: gold label, Ground Truth, Valor verdadero

TP: # muestras positivas que predije positivas.
TN: # " negativas " " negativas.
FN: # " " " " positivos
FP: # " positivas " " negativos

Formas de clasificar:

definir un proc de scoring $\rightarrow$ resuelve la decisión

+ complejidad en el modelo $\Rightarrow$ + complejidad en el computo $\overset{?}{\Rightarrow}$ mejor modelo.

① **Modelos Generativos:** modelar la distrib. de inputs y de outputs $\Rightarrow$ Permite generar y samplear datos sintéticos (GAN's).

② **Modelos discriminantes:** planteo una distrib. $\mathbb{P}(C_k | X)$ $\rightsquigarrow$ usando inferencia Bayesiana obtengo mi modelo. Una vez obtenido usamos el modelo.

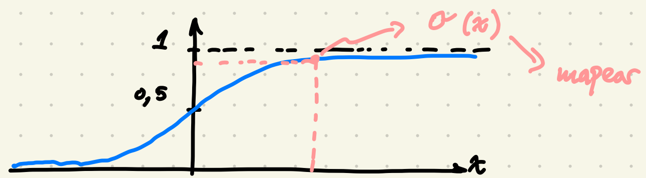
③ **Modelos func. ó discriminantes:** buscamos $f(x)$ $\mathbb{R}^{n \times m} \rightarrow [c_1, c_2, \ldots, c_k]$

Regresión Logística

Usamos la función logistica (sigmoidea / sigmoide) $\sigma(a) = (1 + e^{-a})^{-1}$ $\rightarrow$ pertenece a la familia de fn. squashing:

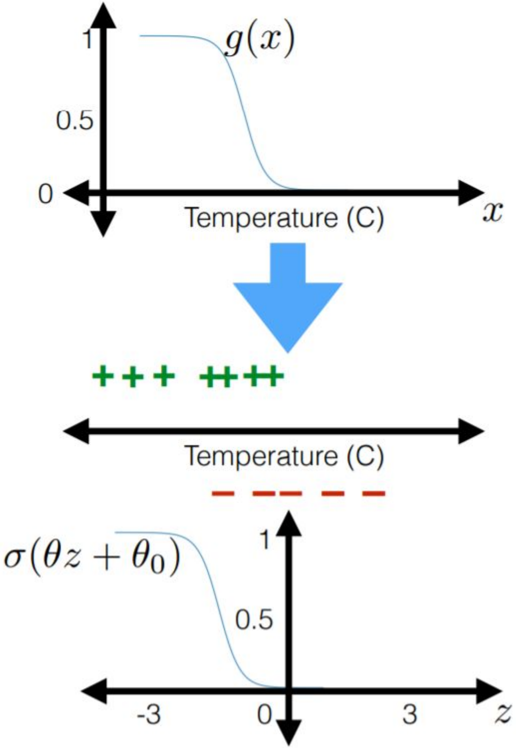
$f: x, a \rightarrow y$

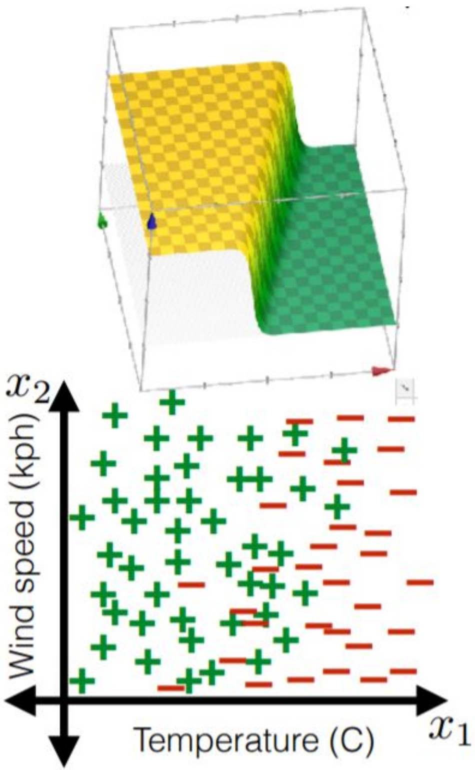
$\mathbb{R} \rightarrow [0,1]$



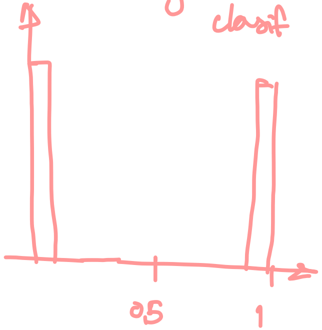
Clasificación Binaria - Motivación

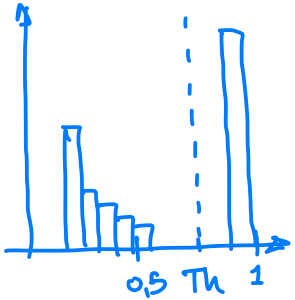
$(x, y) \to (C, P(C))$





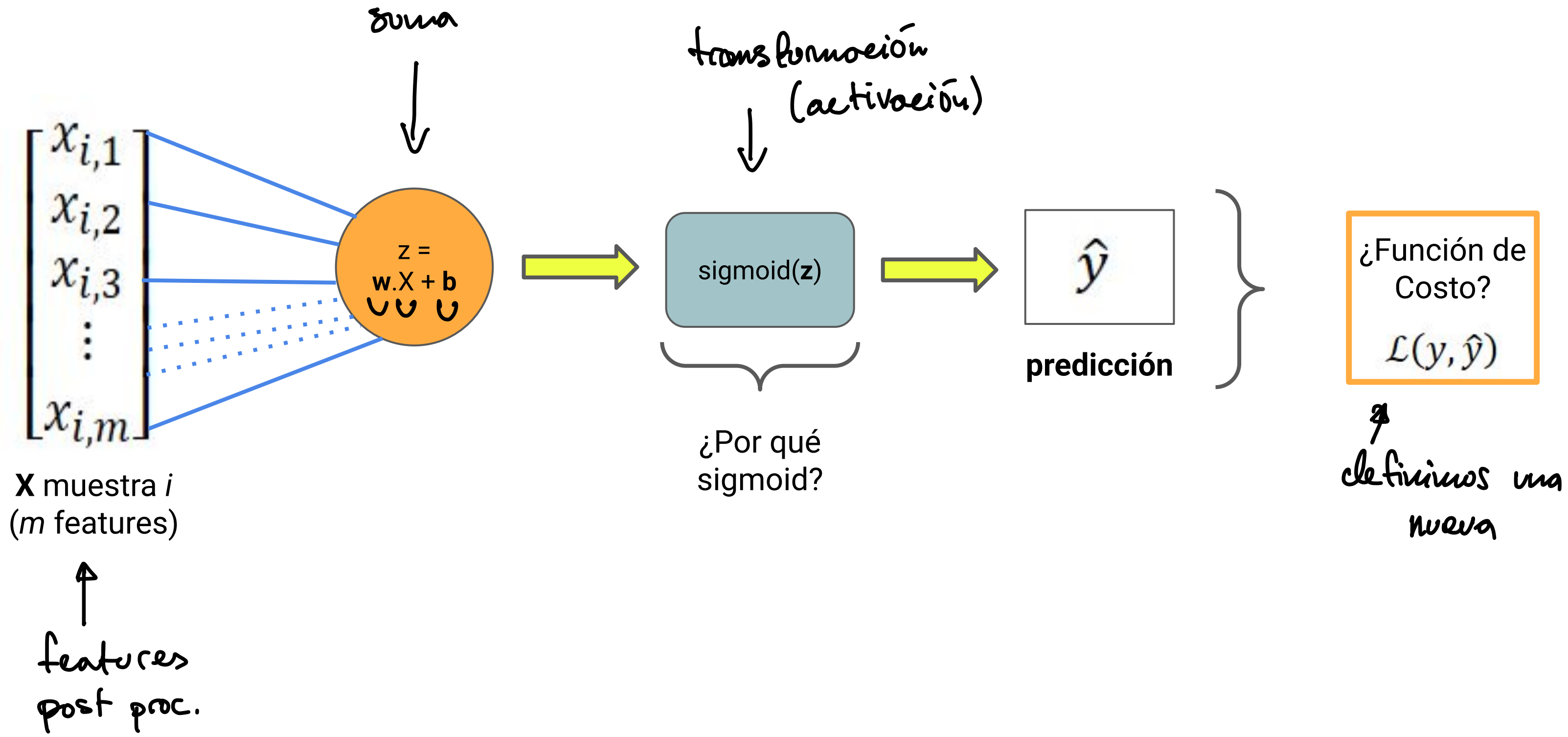
histograma de clasif





Th por default es 0,5

Regresión Logística



queremos mapear $\sigma(a)$, partimos de $z = w^t x$

$\sigma(z) = \sigma(w^t x)$ $\qquad$ $\partial_a \sigma(a) = \sigma(a)(1-\sigma(a))$ (propiedad)

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

Verosimilitud: $\mathbb{P}(y/X) = \prod_{n=1}^{N} \hat{y}_n^{y_n}(1-\hat{y}_n)^{1-y_n}$

N = # muestras
$\hat{y}$ = pred
y = GL

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

$$\max_w \sum_{i=1}^{N} \ln(\mathbb{P}_w(y_i = y_i / \bar{x}_i = x_i))$$

$$\max_w \sum_{i=1}^{N} \ln(\sigma(w^t x)^{y_i} \cdot (1-\sigma(w^t x))^{1-y_i})$$

$$\max_w \sum_{i=1}^{N} y_i \ln \sigma(w^t x) + (1-y_i) \cdot \ln(1-\sigma(w^t x))$$

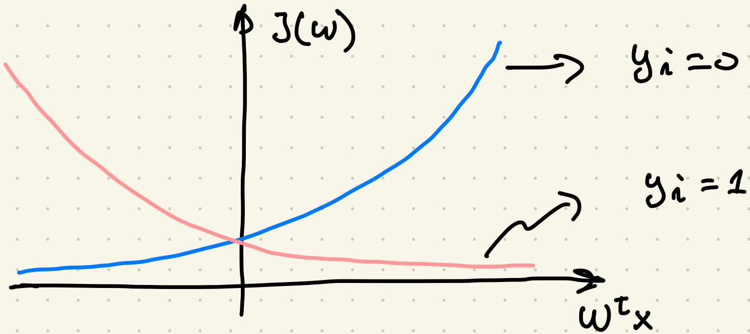
$\times(-1)$

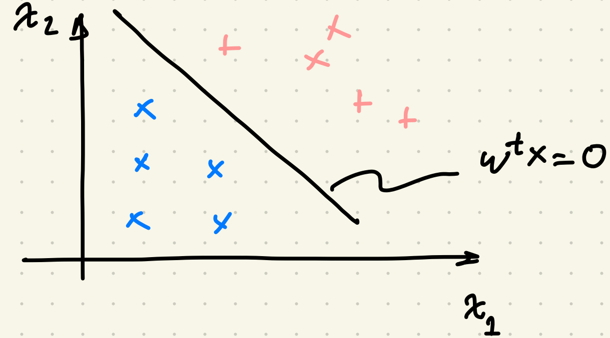
$$\min_w \underbrace{\sum_{i=1}^{N} -y_i\, a - (1-y_i)\, b}_{①}$$

a ① podemos aproximarlo usando:

$$J(w) = \frac{1}{n} \sum_{i=1}^{N} [-y_i\, ⓐ - (1-y_i)\, ⓑ]$$

binary cross entropy

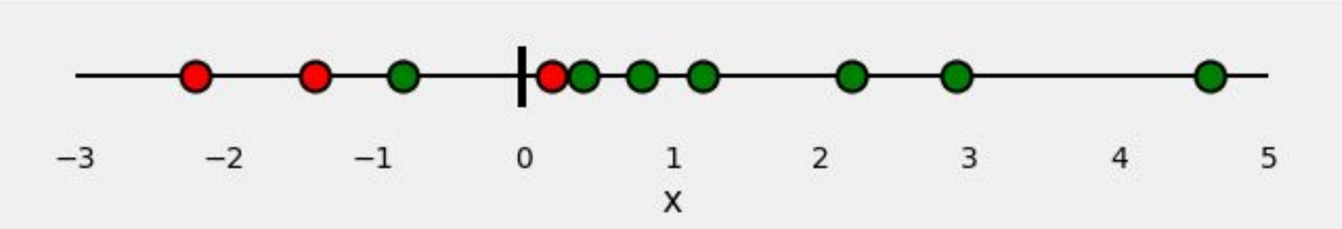


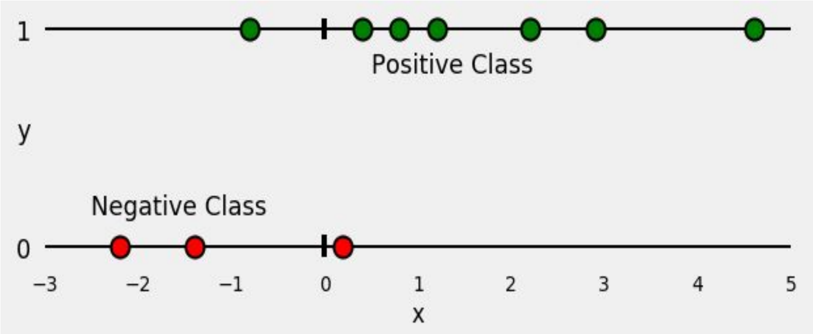


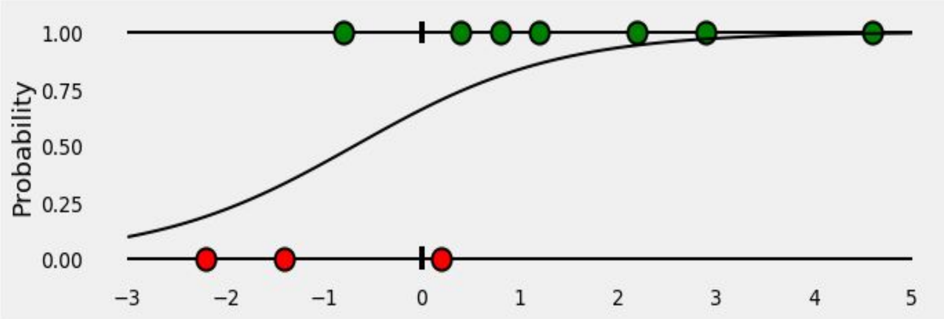
$$\nabla_w J(w) = \emptyset$$

→ GD
→ SGD
→ GDB

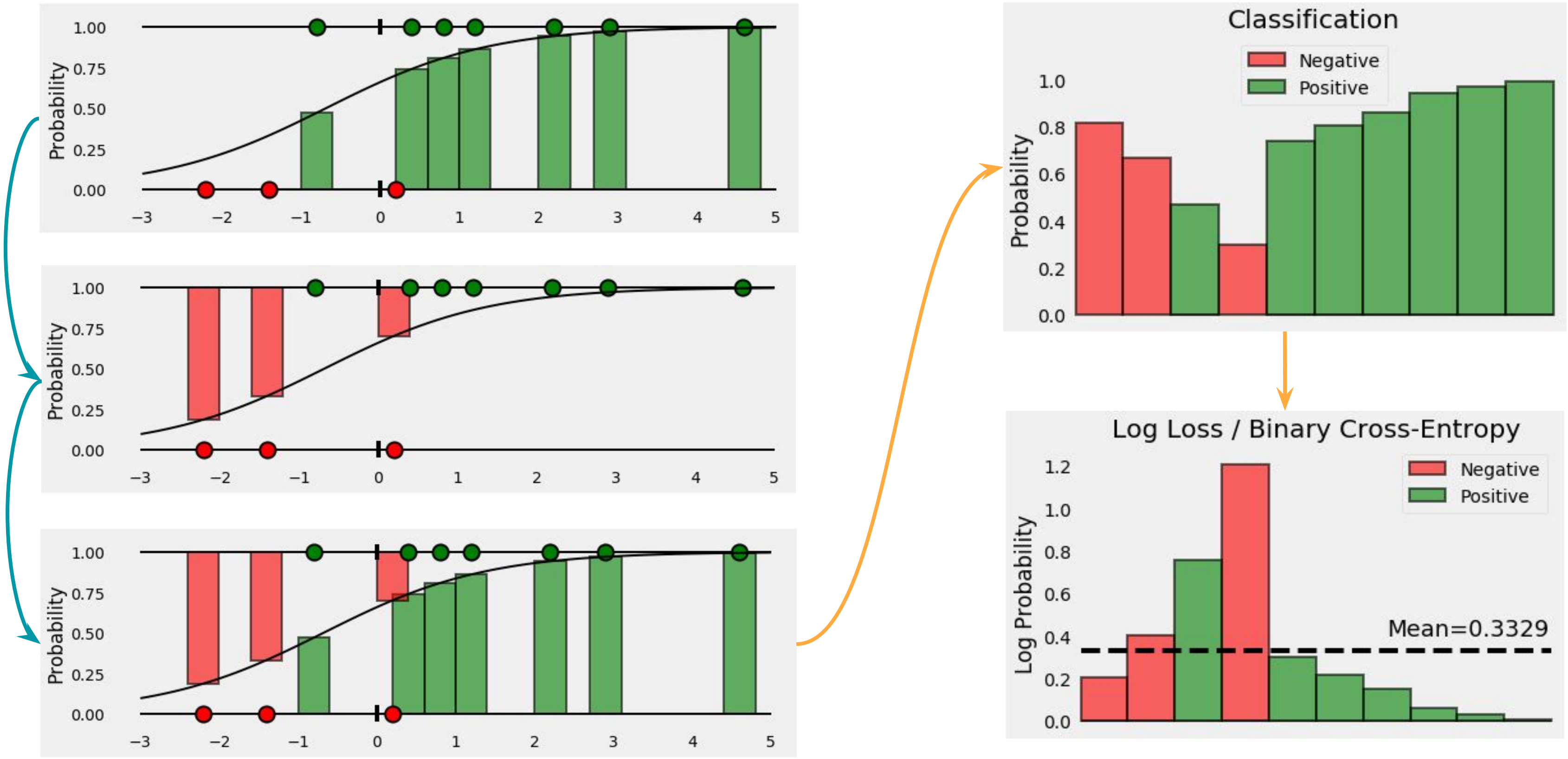
Regresión Logística





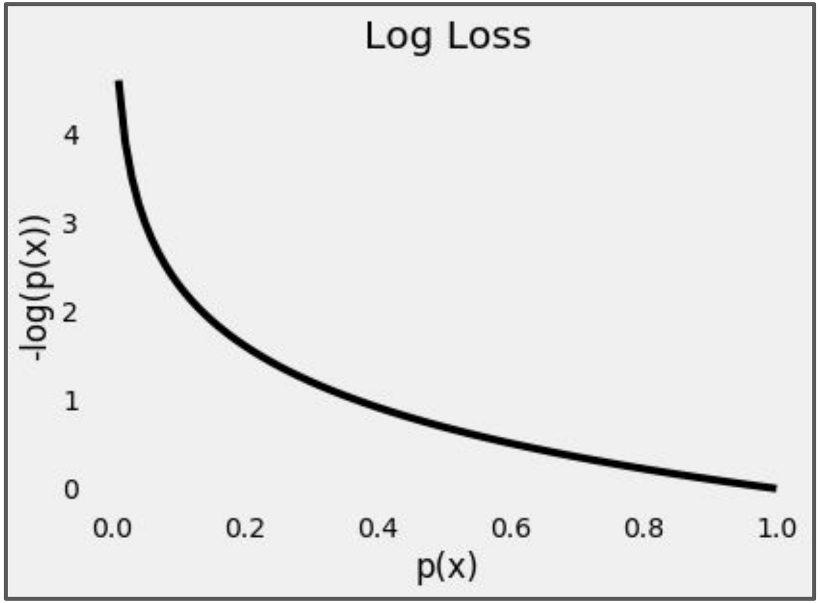


Regresión Logística



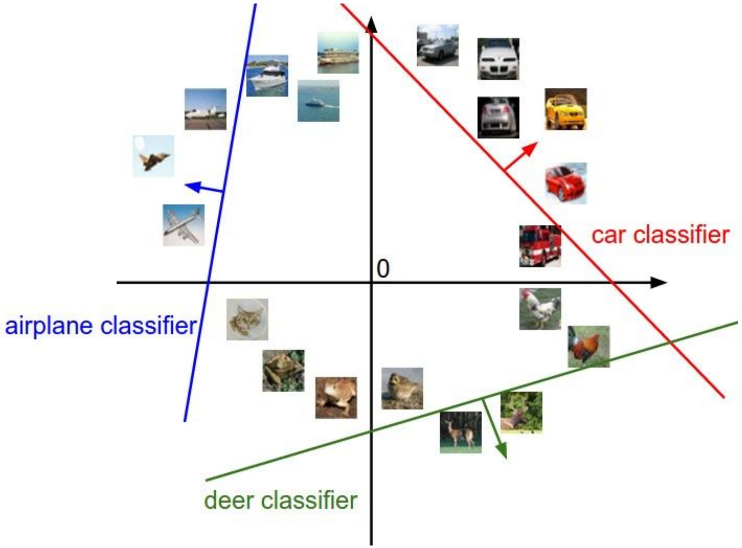
Regresión Logística

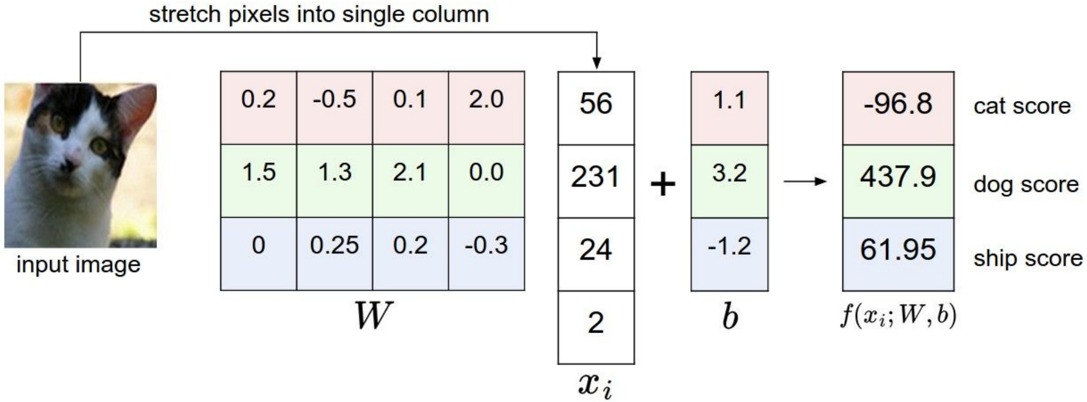
$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$



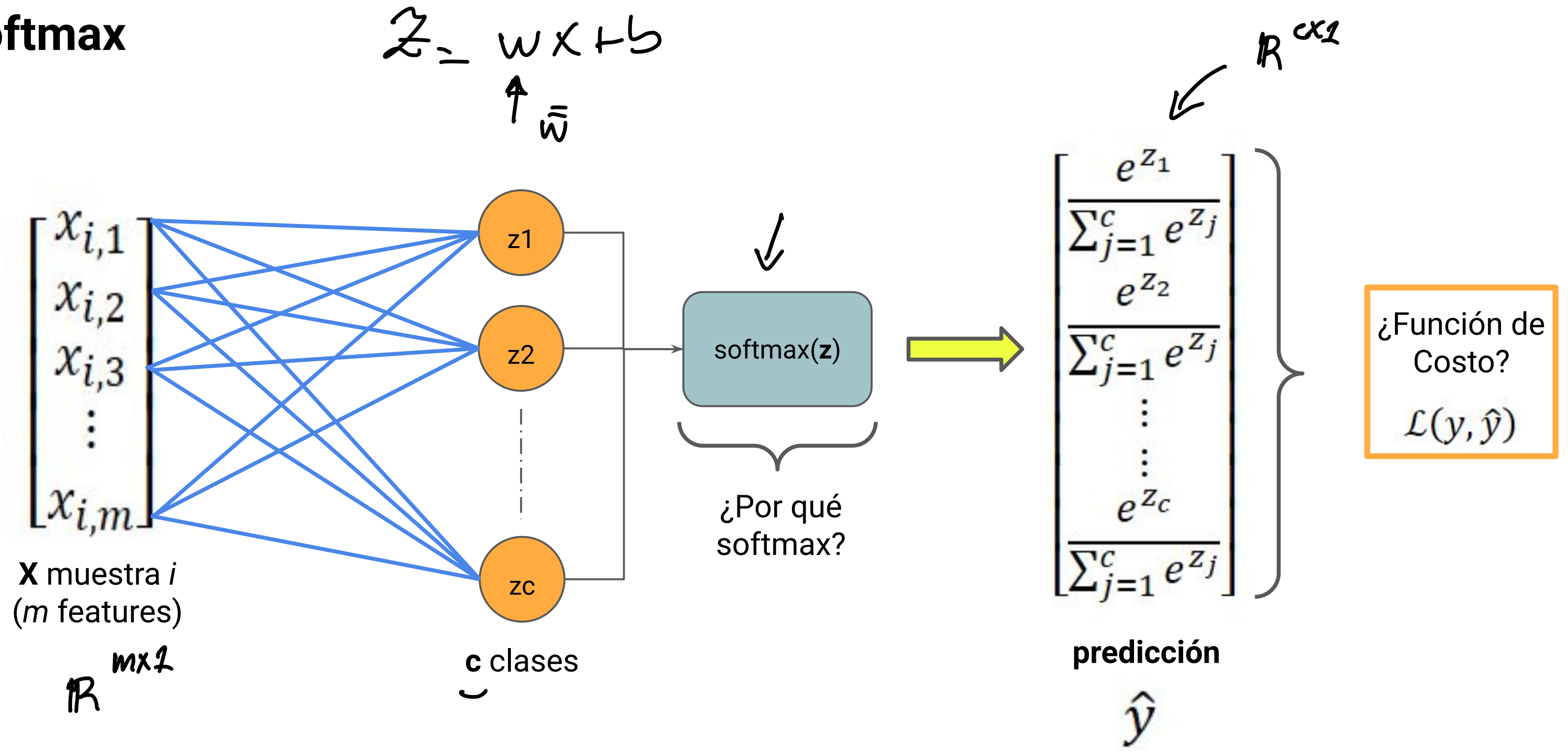
(1) REGRESIÓN LOGÍSTICA - EJERCICIO DE APLICACIÓN

Clasificación Multiclase - Motivación





Softmax



$z = wx + b$
$\bar{w}$
$\mathbb{R}^{c\times 1}$
$x_{i,1}$
$x_{i,2}$
$x_{i,3}$
$x_{i,m}$
X muestra *i*
(*m* features)
$\mathbb{R}^{m\times 1}$
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

probabilidad de ser clase y_i
dado los x_i y los coef W.

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

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

1 [Softmax Forma Gráfica]

2 [Softmax Visualización 3D]

Softmax

Derivación Cross-Entropy

Derivación Softmax

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

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

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

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

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

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

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

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

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

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

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

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

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

Usar gradiente descendente para actualizar W !!!

(1) SOFTMAX - EJERCICIO DE APLICACIÓN

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