

Inteligencia

Artificial

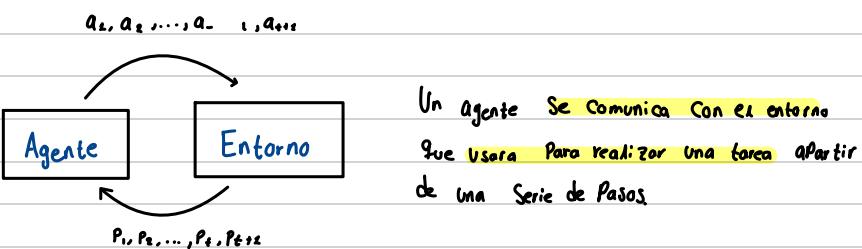
# Inteligencia Artificial

¿Qué es realmente la IA?

- Sistemas que piensan como humanos.
- Sistemas que actúan como humanos.
- Sistemas que piensan racionalmente.
- Sistemas que actúan racionalmente.

## Agente racional

Es cualquier cosa que perciba su ambiente a través de sensores y actúe sobre él mediante actuadores.



Un agente se comunica con el entorno que usará para realizar una tarea apartir de una serie de pasos.

## Agente:

Para diseñar un agente, se debe definir:

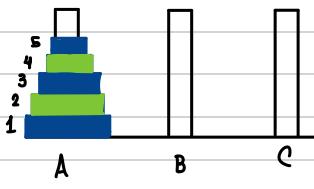
Performance

Entorno

Actuadores

Sensores

## Problema 1:



$$S = (S_1, \dots, S_5) \leftarrow \text{Número de discos}$$

$$S_i \in \{A, B, C\} \leftarrow \text{Torre asignada}$$

$$|S| = 3^5 = 243 \text{ Estados.}$$

## Problema 2:



$$S = (E, O, L)$$

$$E = (0, 1, 2, 3)$$

$$O = (0, 1, 2, 3)$$

$$L = (I, D)$$

$$|S| = 4 \times 4 \times 2 = 32$$

## ENTORNO

Entorno Determinista: Poder calcular el valor exacto.

Entorno Estático: Se va usar una única vez, no cambia en el tiempo.

Entorno Observable: Se puede observar todo el estado.

Entorno Discreto: Va a pertenecer a un grupo finito de estados.

Entorno Continuo: Va a pertenecer a un grupo real de estados.

$$S = (s_1, \dots, s_n) \in D_1 \times \dots \times D_n \quad (\text{ESTADO})$$

→ Determinista, Estático, Observable, discreto:

$$S = f(a) \quad \begin{array}{|c|c|} \hline \text{acciones} & a \in A \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline a \in A(s) & (\text{acciones legales en } s) \\ \hline \end{array}$$

→ Determinista, dinámico, Observable, discreto:

$$S_{k+1} = f(S_k, a_k)$$

→ Determinista, Estático, Parcialmente Observable, discreto:

$$S_{k+1} = f(S_k, a_k)$$

$$P_k = g(S_k)$$

→ Determinista, Estático, Parcialmente Observable, Continuo:

$$\dot{x} = f(x_t, a_t)$$

$$P = g(x_t)$$

→ Estocástico, Estático, Observable, Discreto

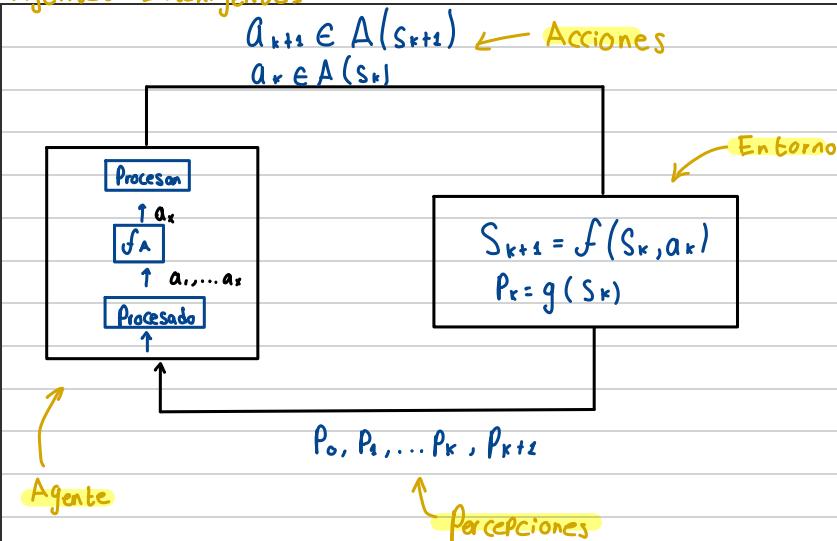
$$S = \{S^{(1)}, \dots, S^{(m)}\} \quad m = \text{Cardinalidad}(S)$$

\* Como es estocástico Puede Pasar Una Cosa O la Otra entonces podemos sacar la Probabilidad de estar en un estado \*

$$Pr[S/a] = \begin{bmatrix} Pr[S=S^{(1)}|a] \\ Pr[S=S^{(2)}|a] \\ \vdots \\ \vdots \\ Pr[S=S^{(m)}|a] \end{bmatrix} \quad * \text{ La suma de las probabilidades es } 1*$$

→ Podemos usar la función de desempeño para evaluar un agente racional.

## Agentes Inteligentes



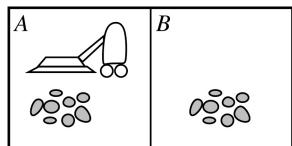
Los agentes incluyen humanos, robots, Softbot, termostatos, etc.

La función del agente se genera desde los historiales de percepción a las acciones.

$$f: P^* \rightarrow A$$

E1 Programa del agente se ejecuta en la arquitectura física para producir  $f$

E1 Problema de la aspiradora



Percepciones: Localización y contenido. EJ [A, Sucio]

Acciones: Izq, Der, Limpiar, Nada.

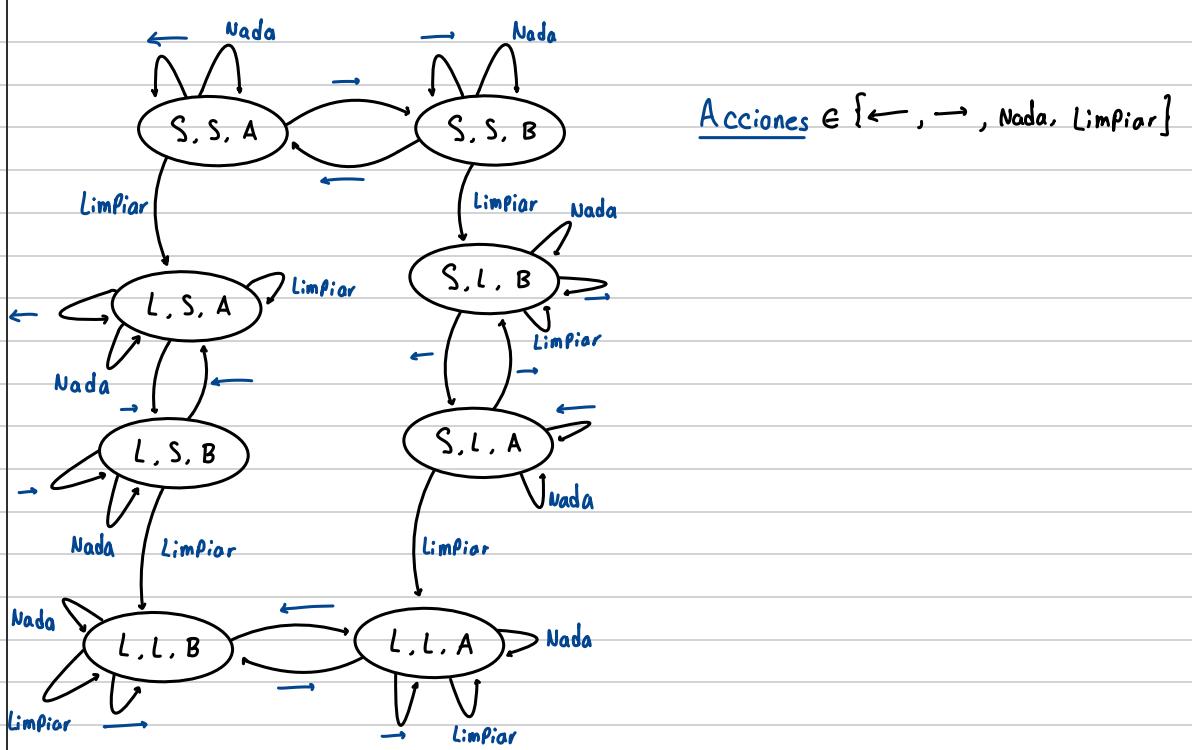
Agente

- Parcialmente Observable.
- Estocástico.
- Determinista.
- Discreto.

Estados:

- $S = \{ \text{Cuarto 1}, \text{Cuarto 2}, \text{robot} \}$
- Cuarto 1  $\in \{\text{Limpio}, \text{Sucio}\}$
- Cuarto 2  $\in \{\text{Limpio}, \text{Sucio}\}$
- robot  $\in \{\text{Cuarto 1}, \text{Cuarto 2}\}$

$$|S| = 2 \cdot 2 \cdot 2 = 8 //$$

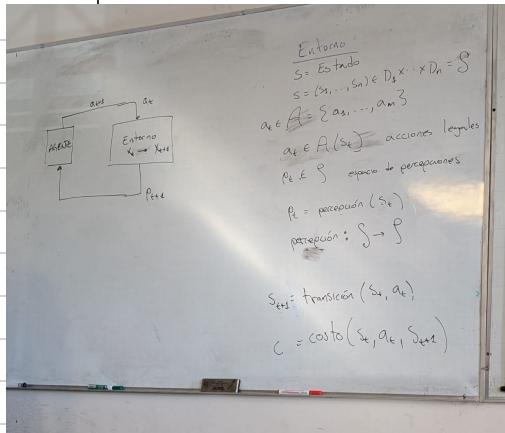


## Tipos de Agentes

## Entorno

$S = \text{Estado}$

$s = (s_1, s_2, \dots, s_n) \in (D_1, D_2, \dots, D_n)$



$f_A$  es desconocida

$$P \rightarrow [f_A(p)] \rightarrow a$$

$$f_A: X \rightarrow Y$$

Típicamente  $X = \mathbb{R}^n$

$$x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$$

Si  $y = \mathbb{R}$  Regresión.

Si  $y = \{-1, 1\}$  Clasificación Binaria.

Si  $y = \{C_1, C_2, \dots, C_k\}$  Clasificación en Varias Clases.

→ Podemos encontrar nuestra función  $f$  Se puede mediante datos del Pasado.

$\{x^{(1)}, \dots, x^{(n)}\}$  Una muestra de  $X$  de dist. desconocida

$$\mathcal{D} = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(n)}, y^{(n)})\}$$

$$\text{donde } y^{(i)} = f_A(x^{(i)}) + \hat{e} \quad \text{V.A. dist. Desc.}$$

El problema es encontrar una función  $h^*: X \rightarrow Y$   
tal que  $h^* \approx f_A \quad h^* \in H$  hipótesis Posibles

$$H = \{h_\theta \mid h_\theta: \mathbb{R} \rightarrow \mathbb{R} \text{ donde } h_\theta(x) = \alpha x, \alpha \in \mathbb{R}\}$$

$$h: X \times \Theta \rightarrow Y, \Theta = \mathbb{R}^p$$

$$h(x^{(i)}, \theta) = \hat{y}^{(i)}$$

$$h_\theta(x^{(i)}) = \hat{y}^{(i)} \quad \text{Si } \theta \text{ es un vector de parámetros fijos.}$$

$$h_\theta: X \rightarrow Y$$

## Hipótesis

- 1:  $f: X \rightarrow Y$  existe y es desconocida
- 2: Tengo  $\{x^{(1)}, \dots, x^{(m)}\} \subseteq X$  Una muestra m de datos con distribución desconocida
- 3:  $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$  donde  $y^{(i)} = f(x^{(i)}) + e^{(i)}$  donde  $e^{(i)}$  son V.A.
- 4: Tenemos una función "Parametrizada"  $h: X \times \Theta \rightarrow Y$ , Tipicamente  $\Theta = \mathbb{R}^p$
- 5: Para un valor específico de  $\theta$ ,  $h_\theta: X \rightarrow Y$
- 6:  $h_\theta \in H$  Conjunto de hipótesis.
- 7: El aprendizaje Supervisado Consiste en encontrar  $h^* \in H$  tal que  $h^* \approx f$

## Error

$L: Y \times Y \rightarrow \mathbb{R}$  Función de Perdida

$$L(y, \hat{y}) = L(f(x), h^*(x))$$

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$$

$$= |y - \hat{y}| \quad y \in \mathbb{R}$$

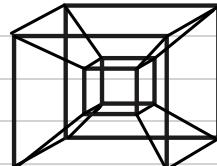
$$L(y, \hat{y}) = \begin{cases} 0 & \text{Si } y = \hat{y} \\ 1 & \text{Si } y \neq \hat{y} \end{cases} \quad y \in \{-1, 1\}$$

$$E_{out}(h^*) = \mathbb{E}_{x \in X} [L(f(x), h^*(x))]$$

$$f \approx h^* \text{ Si } y \text{ solo si } E_{out}(h^*) \approx 0$$

$$E_{in}(h^*) = \frac{1}{m} \sum_{i=1}^m L(y^{(i)}, h^*(x^{(i)}))$$

$$f \approx h^* \text{ Si } \begin{cases} E_{in}(h^*) \approx 0 \\ E_{in}(h^*) \approx E_{out}(h^*) \end{cases}$$



## Desigualdad de Hoeffding

$$\Pr(|E_{out}(h^*) - E_{in}(h^*)| > \epsilon) \leq 2e^{-2\epsilon^2 n}$$

donde  $n$  es el tamaño de la muestra

$$d_{vc}(H) \approx k \text{ Parámetros Independientes}$$

El aprendizaje es posible si:

$$10 * d_{vc}(H) \ll n \iff E_{in}(h^*) \approx E_{out}(h^*)$$

$$\begin{bmatrix} X^{(1)\top} \\ X^{(2)\top} \\ \vdots \\ X^{(m)\top} \end{bmatrix} \rightarrow \begin{bmatrix} (x_1^{(1)}, \dots, x_n^{(1)}) \\ (x_1^{(2)}, \dots, x_n^{(2)}) \\ \vdots \\ (x_1^{(m)}, \dots, x_n^{(m)}) \end{bmatrix} \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

$x^{(i)} \in \mathbb{R}^n$   
 $y^{(i)} \in \mathbb{R}$

$$X_{(m,n)} \quad y_{(m,1)}$$

$$D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$$

$$h_\theta(x) = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b = \sum_{j=1}^n w_j x_j + b = x^T w + b$$

S:  $X = (x_1, \dots, x_n) \in \mathbb{R}^n$   
 $w = (w_1, \dots, w_n) \in \mathbb{R}^n$

$$\Theta = (w_1, \dots, w_n, b) \in \mathbb{R}^{n+1}$$

$$h^* = \arg \min_{h \in H} E_{in}(h) \longleftrightarrow \Theta^* = (w^*, b^*) = \arg \min_{\substack{w \in \mathbb{R}^n \\ b \in \mathbb{R}}} E_{in}(h_{w,b})$$

$$w^*, b^* = \arg \min_{\substack{w \in \mathbb{R}^n \\ b \in \mathbb{R}}} \frac{1}{M} \sum_{i=1}^M \frac{1}{2} (y^{(i)} - h_{w,b}(x^{(i)}))^2$$

MSE  
Mean, Square, Error.

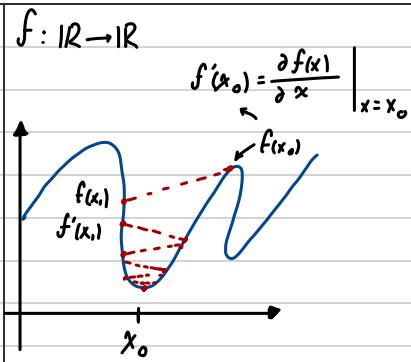
$$= \arg \min_{\substack{w \in \mathbb{R}^n \\ b \in \mathbb{R}}} \frac{1}{M} \sum_{i=1}^M \frac{1}{2} (y^{(i)} - x^{(i)\top} w - b)^2$$

$$= \arg \min_{\substack{w \in \mathbb{R}^n \\ b \in \mathbb{R}}} \frac{1}{2M} \left[ (y^{(1)} - x^{(1)\top} w - b) \dots (y^{(m)} - x^{(m)\top} w - b) \right]$$

$$= \arg \min_{\substack{w \in \mathbb{R}^n \\ b \in \mathbb{R}}} \frac{1}{2M} \left[ y - \vec{x}_w - \vec{b} \right]^T \left[ y - \vec{x}_w - \vec{b} \right]$$

$$= \arg \min_{\theta \in \mathbb{R}} \frac{1}{2M} \left[ y - \left[ x, \vec{1} \right] \theta \right]^T \left[ y - \left[ x, \vec{1} \right] \theta \right]$$

$$\theta := [\vec{x}_e^\top \vec{x}_e]^{-1} \vec{x}_e^\top y$$



1:  $x_0$

2:  $x_1 \leftarrow x_0 - \eta f'(x_0)$

3:  $x_2 \leftarrow x_1 - \eta f'(x_1)$

:

K:  $x_{k+1} \leftarrow x_k - \eta f'(x_k)$  Descenso del gradiente

$$h_\theta(x) = w_0 x_0 + \dots + w_n x_n + b = w^T x + b = [x^T, 1] \theta \leftarrow \hat{y}$$

$$\theta^* = w^*, b^* = \arg \min_{\theta \in \mathbb{R}^{n+2}} \frac{1}{2M} \sum_{i=1}^M (y^{(i)} - [x^T, 1] \theta)^2$$

Para  $w$ :

$$\frac{\partial E_{in}(\theta)}{\partial w_j} = \frac{1}{M} \sum_{i=1}^M -2(y^{(i)} - \hat{y}^{(i)}) x_j^{(i)}$$

$$\frac{\partial E_{in}(\theta)}{\partial b} = \frac{1}{M} \sum_{i=1}^M -2(y^{(i)} - \hat{y}^{(i)})$$

$$E_{in}(\theta_k) = -\frac{1}{M} \begin{bmatrix} \sum_{i=1}^M (y^{(i)} - \hat{y}^{(i)}) x_1^{(i)} \\ \sum_{i=1}^M (y^{(i)} - \hat{y}^{(i)}) x_2^{(i)} \\ \vdots \\ \vdots \\ \sum_{i=1}^M (y^{(i)} - \hat{y}^{(i)}) x_n^{(i)} \end{bmatrix} = -\frac{1}{M} \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(m)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(m)} \\ \vdots & \vdots & & \vdots \\ x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(m)} \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} y^{(1)} - \hat{y}^{(1)} \\ y^{(2)} - \hat{y}^{(2)} \\ \vdots \\ y^{(m)} - \hat{y}^{(m)} \end{bmatrix}$$

## Programa que aprende

```
def Des_gr_f1(x, y, w_0, b_0, lr, max-epochs, e-tol):
    M = X.shape[0]
    w = w_0.copy()
    b = b_0.copy()
    hist = []

    for _ in range(max-epochs):
        y-est = X @ w + b
        Err = y - y-est
        hist.append(np.square(Err).mean())
        grad_w = -1/M * X.T @ Err
        d_b = Err.mean()
        w -= lr * grad_w
        b -= lr * d_b
        if np.abs(grad_w).max() < e-tol:
            break
    return w, b, hist
```

$$h_{\theta} = \text{Sign}(w_1 x_1 + w_2 x_2 + b)$$

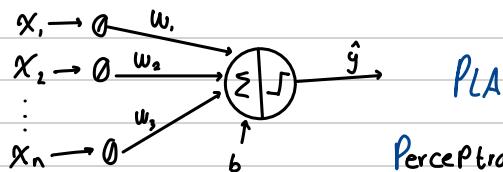
$$X \in \mathbb{R}^n$$

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

$$\mathcal{H} = \{h_{\theta} \mid h_{\theta} = \text{Sign}(w^T x + b), w \in \mathbb{R}^n, b \in \mathbb{R}, \theta = (w, b)\}$$

$$\text{loss}(y, \hat{y}) = \begin{cases} 1 & \text{si } \hat{y} \neq y \\ 0 & \text{en otro caso} \end{cases} = \max(-y \hat{y}, 0)$$

$$E_{in}(w, b) = \frac{1}{M} \sum_{i=1}^M \text{loss}(y^{(i)}, \text{Sign}(w^{(T)} X^{(i)} + b))$$



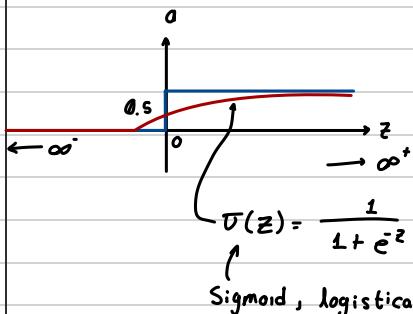
Perceptron Learning Algorithm.

$$\alpha = \Pr(y = 1 \mid X = x; \theta)$$

$$\hat{y} = \begin{cases} 1 & \text{si } a > 0 \\ -1 & \text{en otro caso} \end{cases}$$

$$a = f(w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b)$$

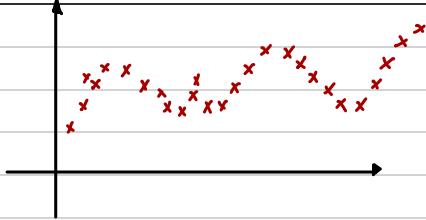
$$z = w \cdot x + w_0 x_0 + b = x^T w + b = x_e^T \theta$$



$$E_{in}(w, b) = \frac{1}{M} \sum_{i=1}^M \text{Loss}(a^{(i)}, \hat{a}^{(i)})$$

$$\text{loss}(a^{(i)}, \hat{a}^{(i)}) = \begin{cases} -\log(\hat{a}^{(i)}) & \text{si } a^{(i)} = 1 \\ -\log(1 - \hat{a}^{(i)}) & \text{si } a^{(i)} = 0 \end{cases}$$

$$= -a^{(i)} \log(\hat{a}^{(i)}) - (1 - a^{(i)}) \log(1 - \hat{a}^{(i)})$$



$$\phi: \mathbb{R}^n \rightarrow \mathbb{R}^n$$

$$x \in \mathbb{R}$$

$$x' = \phi(x) = (x, x^2, x^3, x^4)$$

$$x' = \phi(x) = (\phi_1(x), \phi_2(x), \phi_3(x), \dots, \phi_n(x))$$

### Expansion Polinomial

$$x = (x_1, x_2)$$

$$\phi(x) = (x_1, x_2, x_1^2, x_1 x_2, x_2^2)$$

$$\phi(x) = (x_1, x_2, x_1^2, x_1 x_2, x_2)$$

$$\frac{\partial E_{in}(w, b)}{\partial w_j} = \frac{1}{M} \sum_{i=1}^M \left( \frac{a^{(i)}}{\hat{a}^{(i)}} - \frac{1-a^{(i)}}{1-\hat{a}^{(i)}} \right) \hat{a}^{(i)} (1-\hat{a}^{(i)}) x_j^{(i)}$$

$$\frac{\partial E_{in}(w, b)}{\partial w_j} = \frac{\partial}{\partial w_j} \frac{1}{M} \sum_{i=1}^M \left[ -a^{(i)} \ln |\hat{a}^{(i)}| - (1-a^{(i)}) \ln (1-\hat{a}^{(i)}) \right]$$

$$= \frac{1}{M} \sum_{i=1}^M -\frac{\hat{a}^{(i)}}{a^{(i)}} \frac{\partial \hat{a}^{(i)}}{\partial w_j} + \frac{1-a^{(i)}}{1-\hat{a}^{(i)}} \frac{\partial \hat{a}^{(i)}}{\partial w_j}$$

$$= \frac{1}{M} \sum_{i=1}^M \left[ -\frac{a^{(i)}}{\hat{a}^{(i)}} + \frac{a^{(i)} + \hat{a}^{(i)}}{a^{(i)}} + \hat{a}^{(i)} - \frac{a^{(i)} + \hat{a}^{(i)}}{\hat{a}^{(i)}} \right] x_j^{(i)}$$

$$w^*, b^* = \arg \min_{w, b} E_{in}(w, b)$$

bajo

$$\sum_{j=1}^n w_j^2 \leq C$$

