

Redes Neuronales Multilayer Perceptron

Cristian López Del Alamo

clopezd@utec.edu.pe

IPRODAM3D - Research group

2022



Programa



1. Introducción
2. Multilayer Perceptron
3. Arquitectura
4. Algoritmo Back Propagation
5. Entrenamiento
6. Testing
7. Aplicaciones

1

Introducción

UTEC

¿Qué hemos visto?

Aprendizaje supervisado (Clasificación)

- Regresión Logística
- SVM
- KNN
- Árboles de Decisión
- Ensemble method
 - Random Forest
 - Boosting
 - Bagging

Métricas

- Matriz de confusión
- Precision, Recall
- F1 - Score

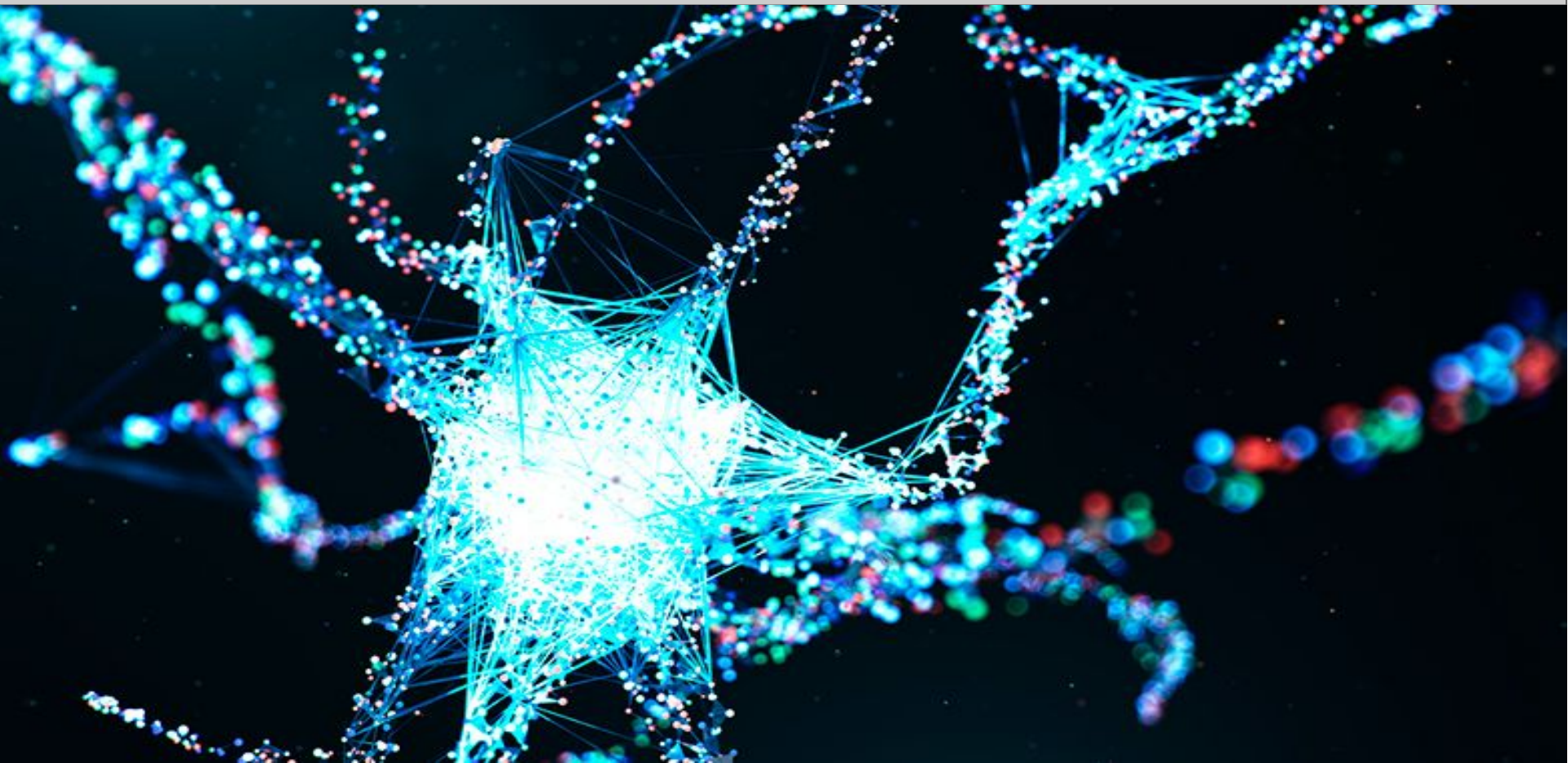
Aprendizaje no supervisado (clustering)

- Kmeans
- DBSCAN
- Mean Shift
- GMM
- Algoritmos Aglomerativos

Métricas

- Silhouette
- Purity
- Entropy
- MI
- Rand Index

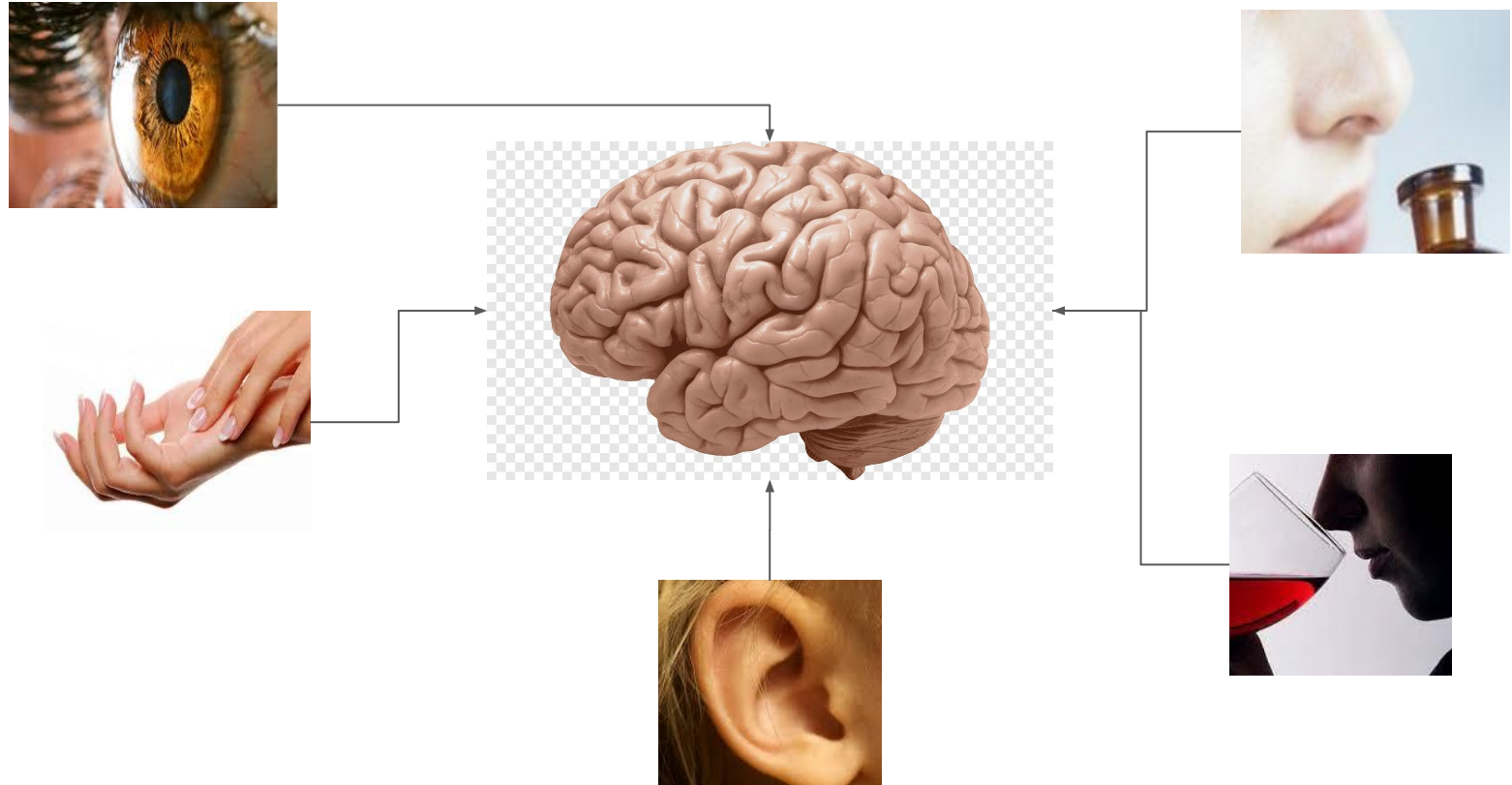
Redes Neuronales Artificiales



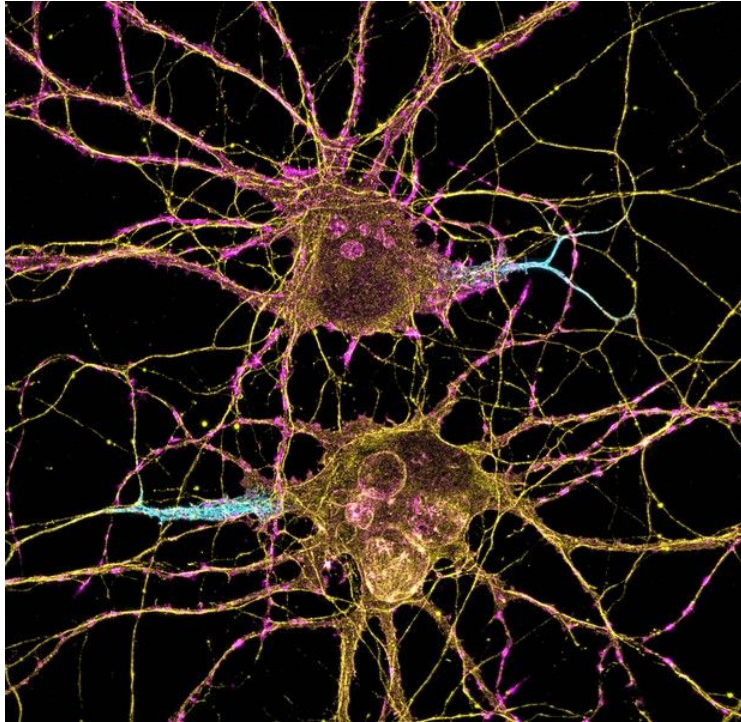
1. Objetivo

- Comprender qué es y cómo funciona una red Neuronal como clasificador.
- Comprender qué le ocurre a un vector cuando lo multiplicamos a una matriz
- Qué hace realmente una MLP a los datos y cómo es que los clasifica.

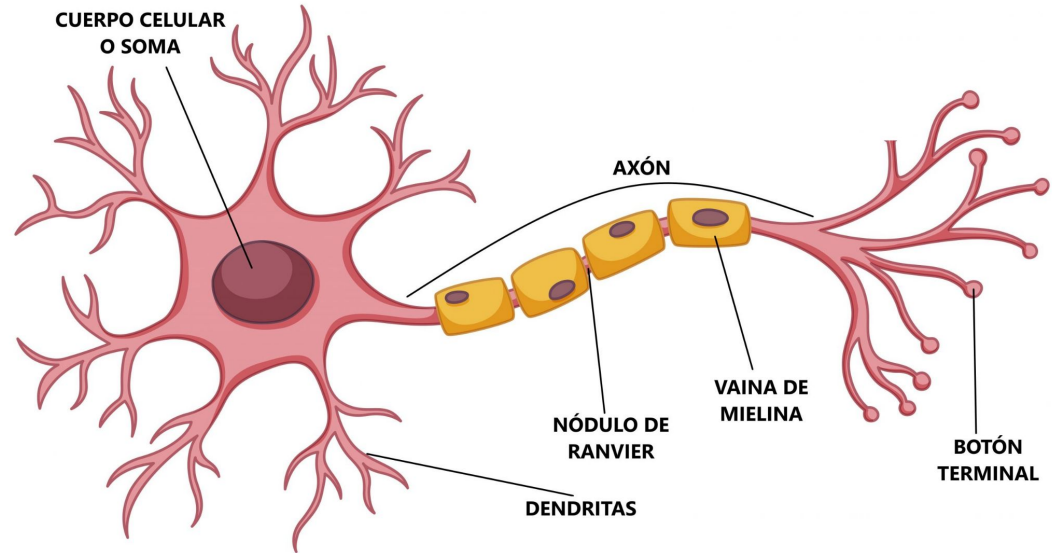
2. Redes Neuronales



2. Redes Neuronales



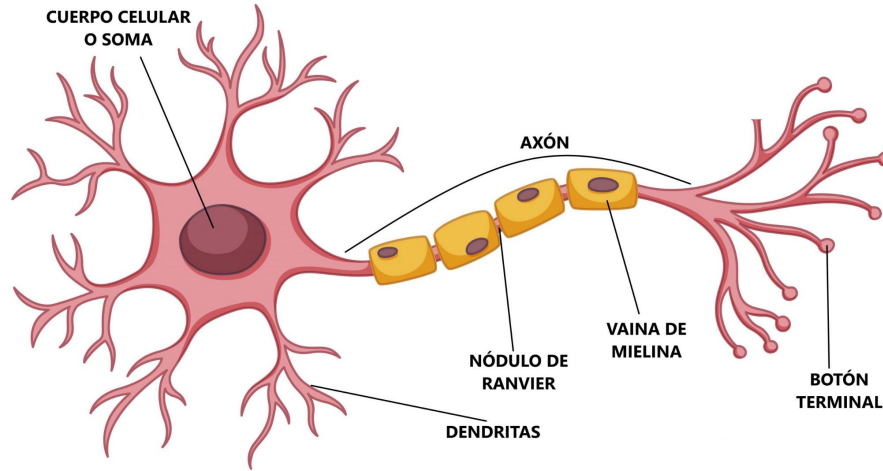
Fuente: [Click](#)



Fuente: [Click](#)

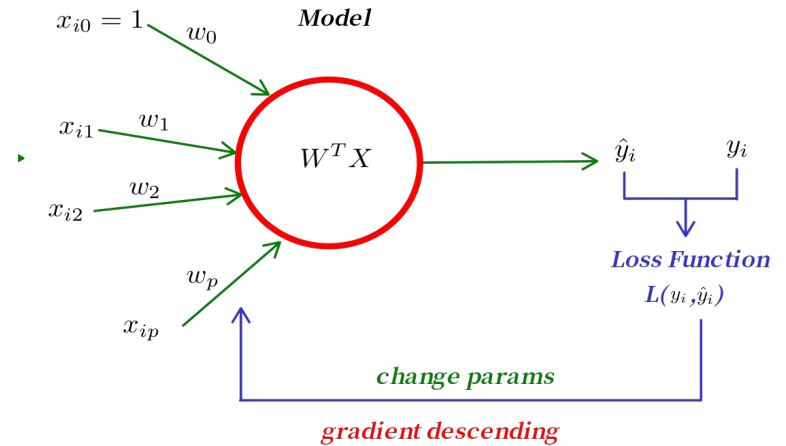
2. Redes Neuronales

Neurona Biológica



Fuente: [Click](#)

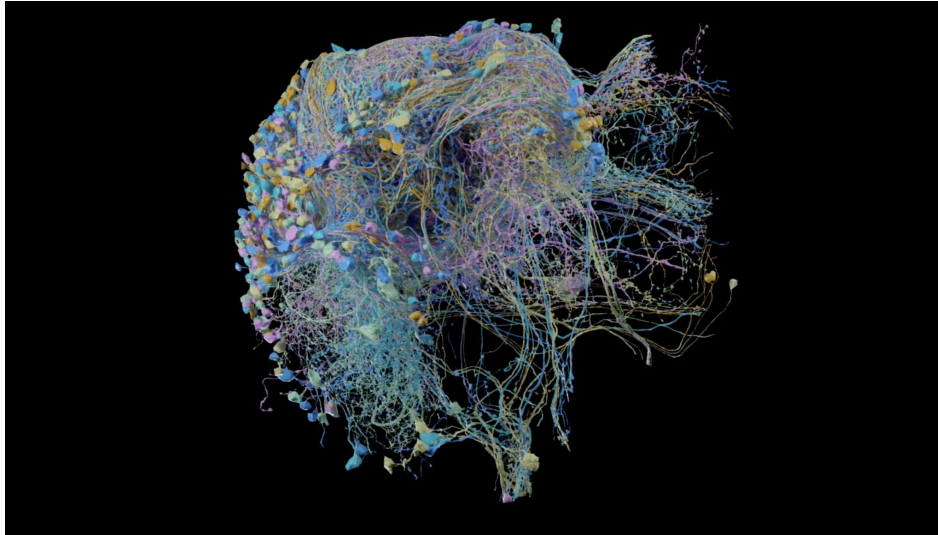
Neurona Artificial



Fuente: [Click](#)

2. Redes Neuronales

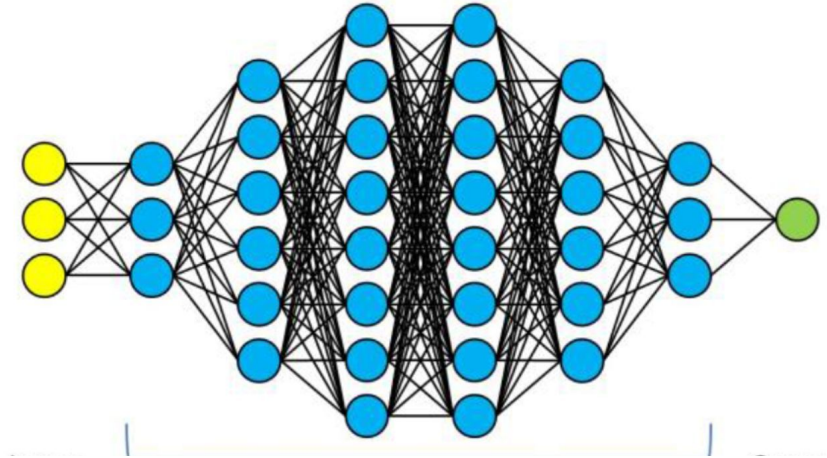
Red Neuronal Biológica



- 100000 neuronas
- Millones de Millones de conexiones

Fuente: [Click](#)

Red Neuronal Artificial



Fuente: [Click](#)

2. Redes Neuronales : Aproximador Universal

$$f(\text{img_dog}) = \text{"Perro"}$$

$$f(\text{img_giraffe_family}) = \text{img_giraffe_family}$$

$$f(\text{img_dog_cat}) = \text{img_mask_dog_cat}$$

$$f(\text{img_fingerprint}) = \text{img_fingerprint}$$

$$f(\text{img_jet}) = \text{img_jet_red}$$

$$f(\text{audio_seagull}) = \text{"Gaviota"}$$

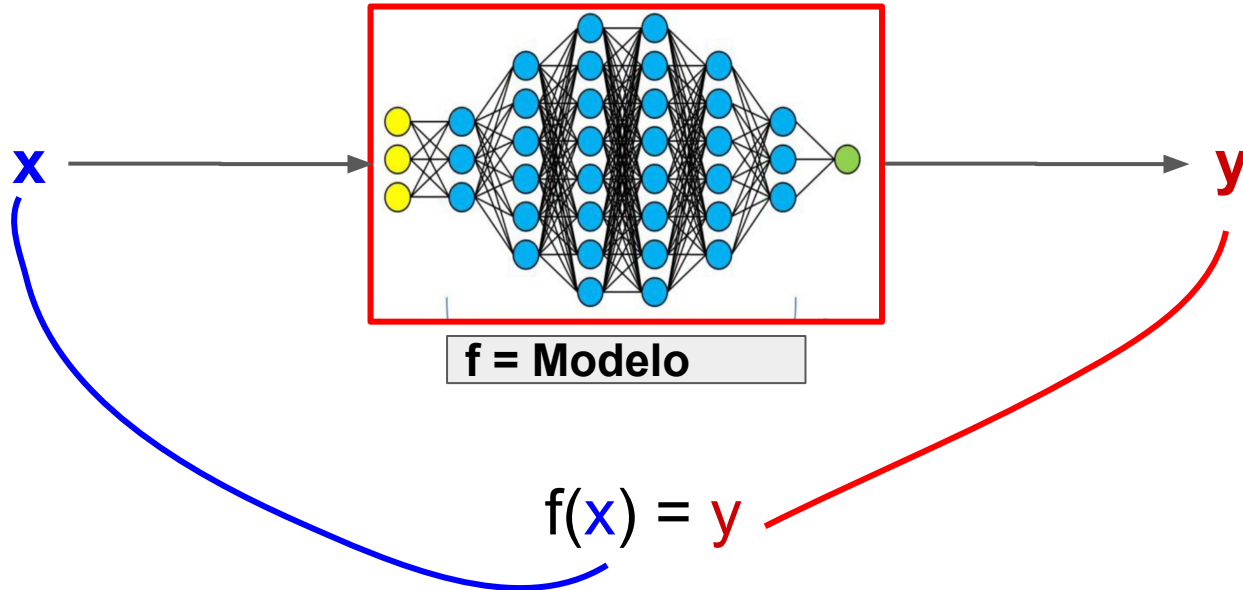
2. Redes Neuronales : Aproximador Universal

Cualquier **función continua** definida en un **hipercubo unitario n-dimensional** puede ser **aproximado** por una **suma finita** de :

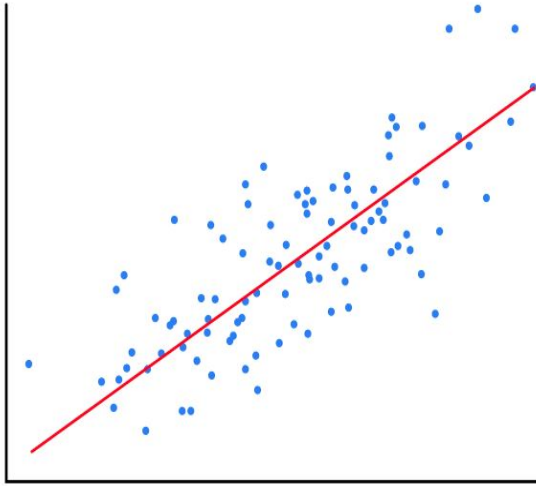
$$\sum_{i=0}^n v_i \phi(WX + B),$$

Donde $v_i, b_i \in R, W \in R^n$, y ϕ es una **función continua discriminatoria**

2. Redes Neuronales



2. Redes Neuronales : Regression Model



Model: $f(x) = w_0 + w_1x$

Parameters: w_0, w_1

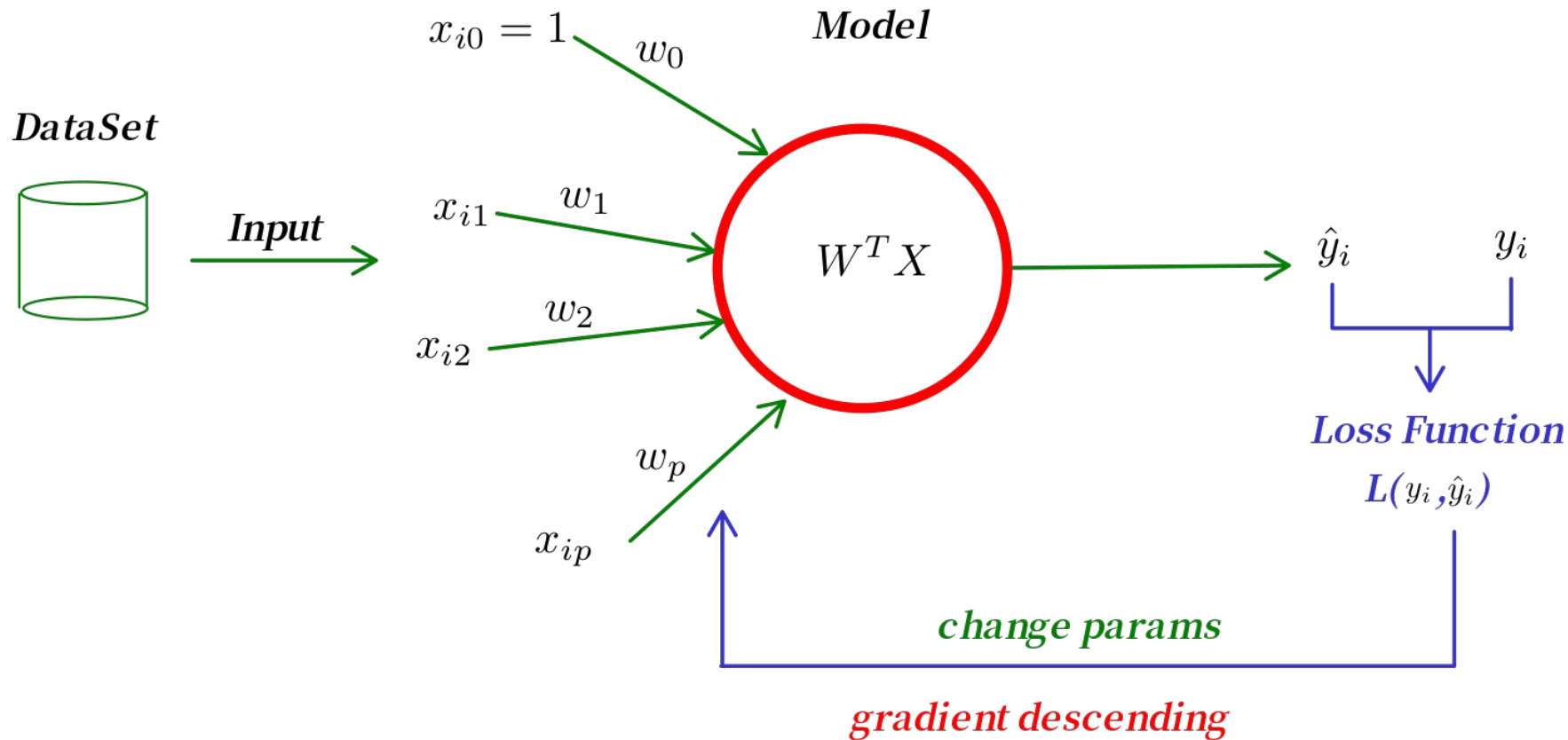
loss: $loss(y, f(x)) = \frac{1}{2n} \sum_{i=0}^n (y_i - f(x_i))^2$

gradiente: $\frac{\partial loss}{\partial w_0} = \frac{1}{n} \sum_{i=0}^n (y_i - f(x_i))(-1)$

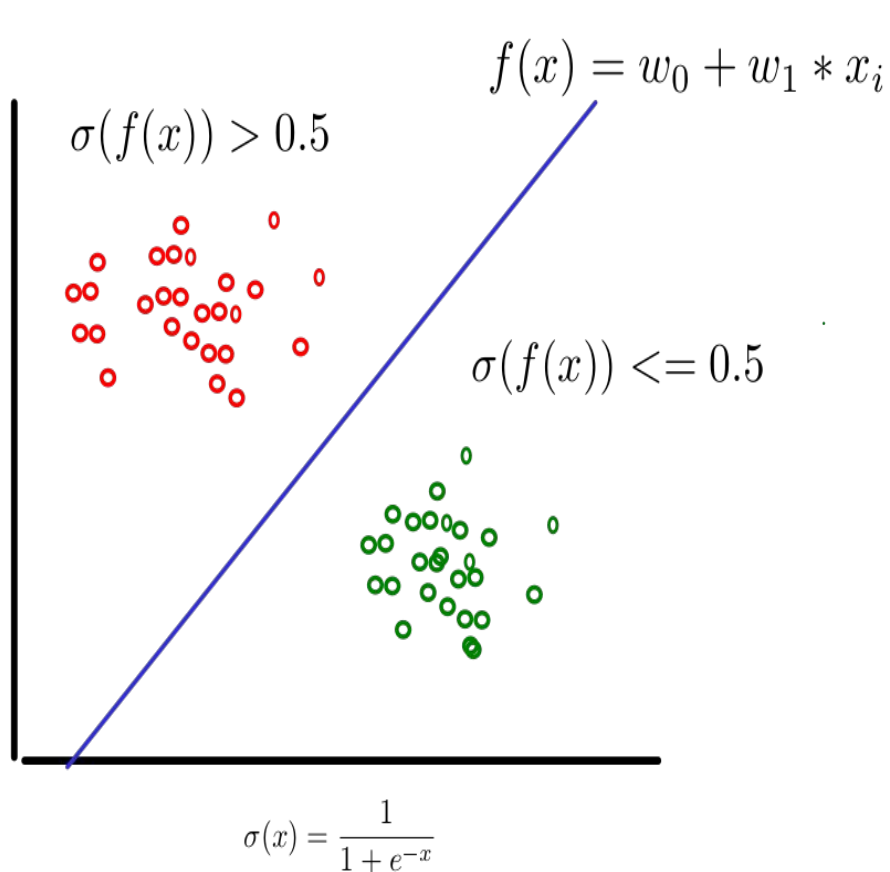
$$\frac{\partial loss}{\partial w_0} = \frac{1}{n} \sum_{i=0}^n (y_i - f(x_i))(-1)$$
$$\frac{\partial loss}{\partial w_1} = \frac{1}{n} \sum_{i=0}^n (y_i - f(x_i))(-x_i)$$

change
parameters: $w_i = w_i - \alpha \frac{\partial loss}{\partial w_i}$

Models: Regression Model



2. Redes Neuronales : Regresión Logística



Model: $f(x) = \frac{1}{1 + e^{-W^T X}}$

Parameters: $w_0, w_1, w_2, \dots, w_p$

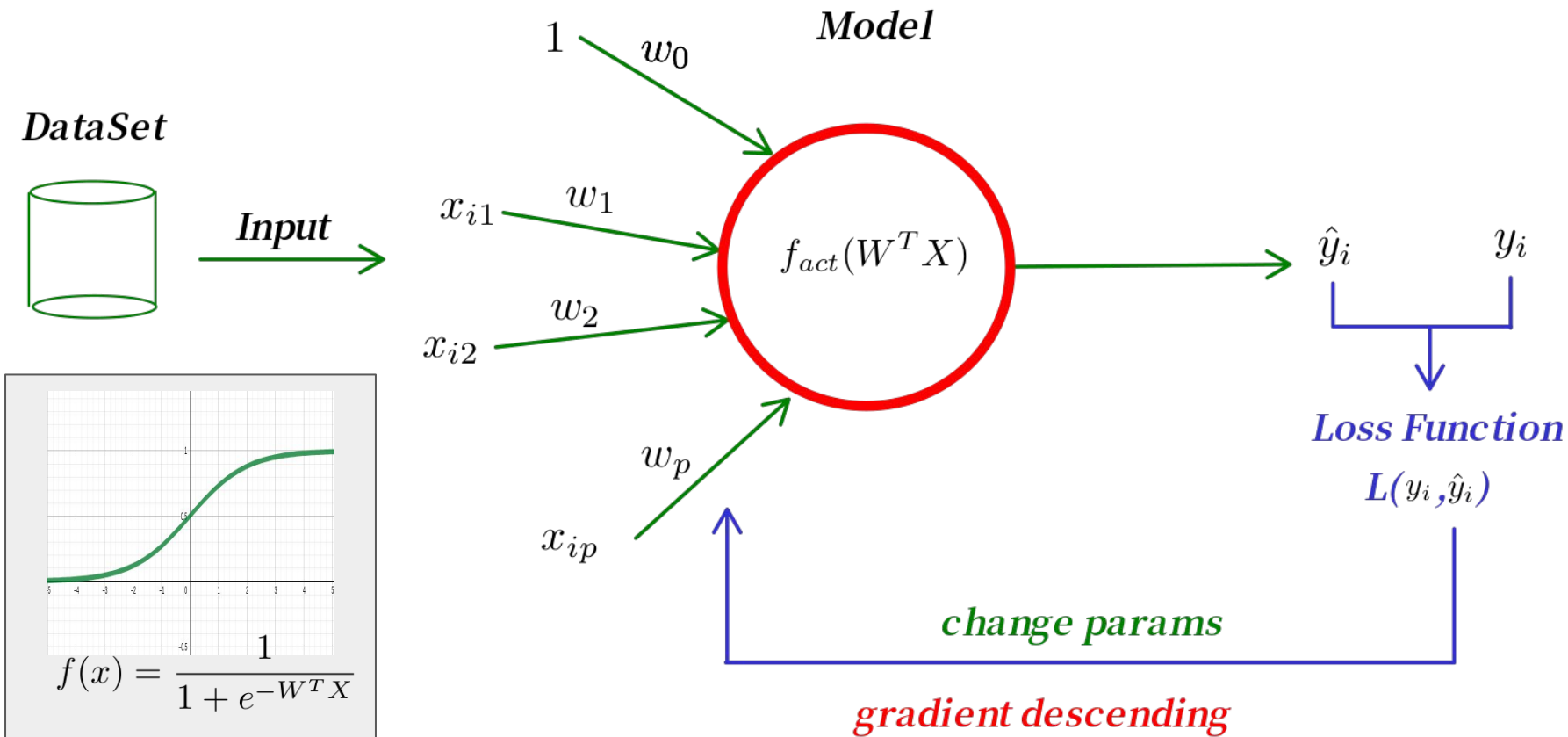
loss:

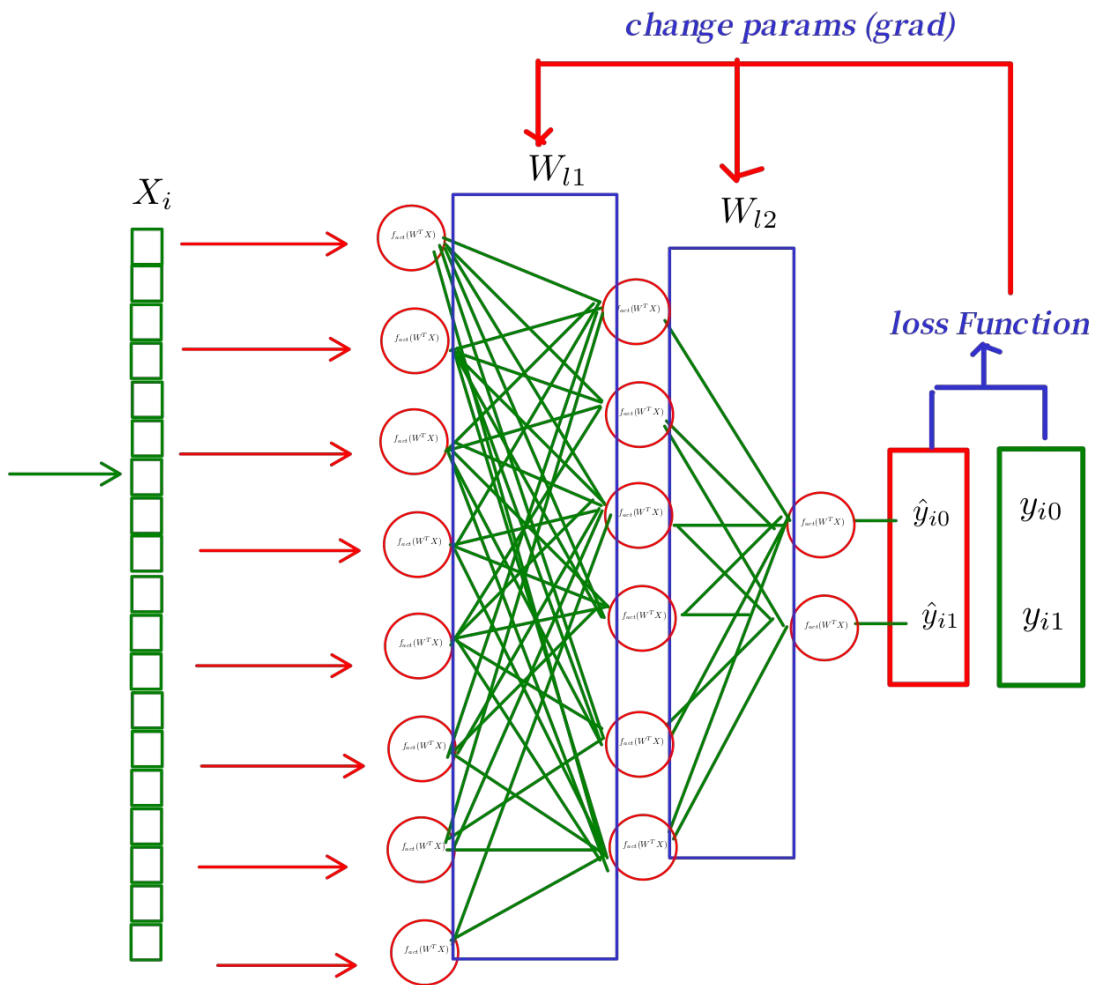
$$\text{loss}(y, f(x)) = -\frac{1}{n} \sum_{i=0}^n [y_i \log(f(x_i)) + (1 - y_i) \log(1 - f(x_i))]$$

gradiente: $\frac{\partial \text{loss}}{\partial w_j} = \frac{1}{n} \sum_{i=0}^n (y_i - f(x_i))(-x_{ij})$

change parameters: $w_i = w_i - \alpha \frac{\partial \text{loss}}{\partial w_i}$

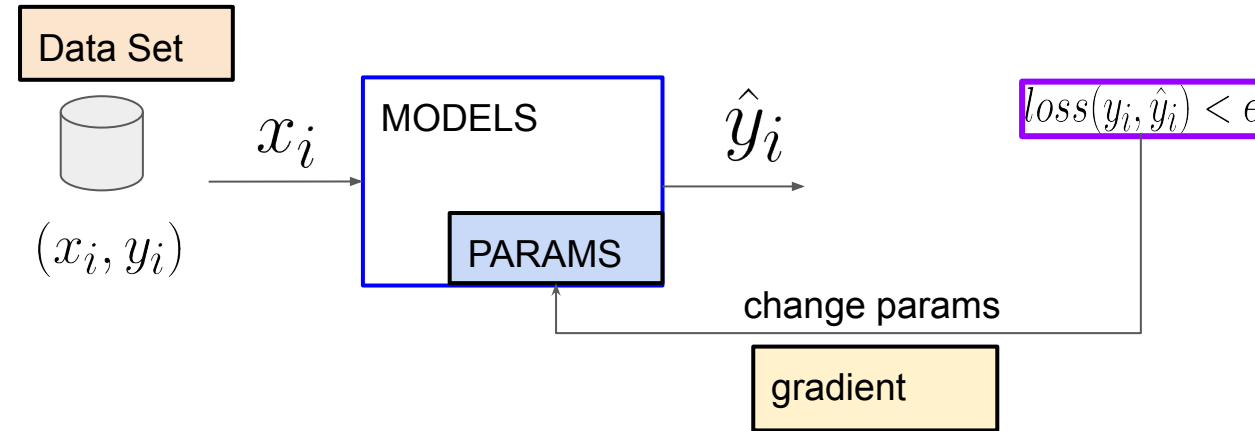
2. Redes Neuronales : Regresión Logística





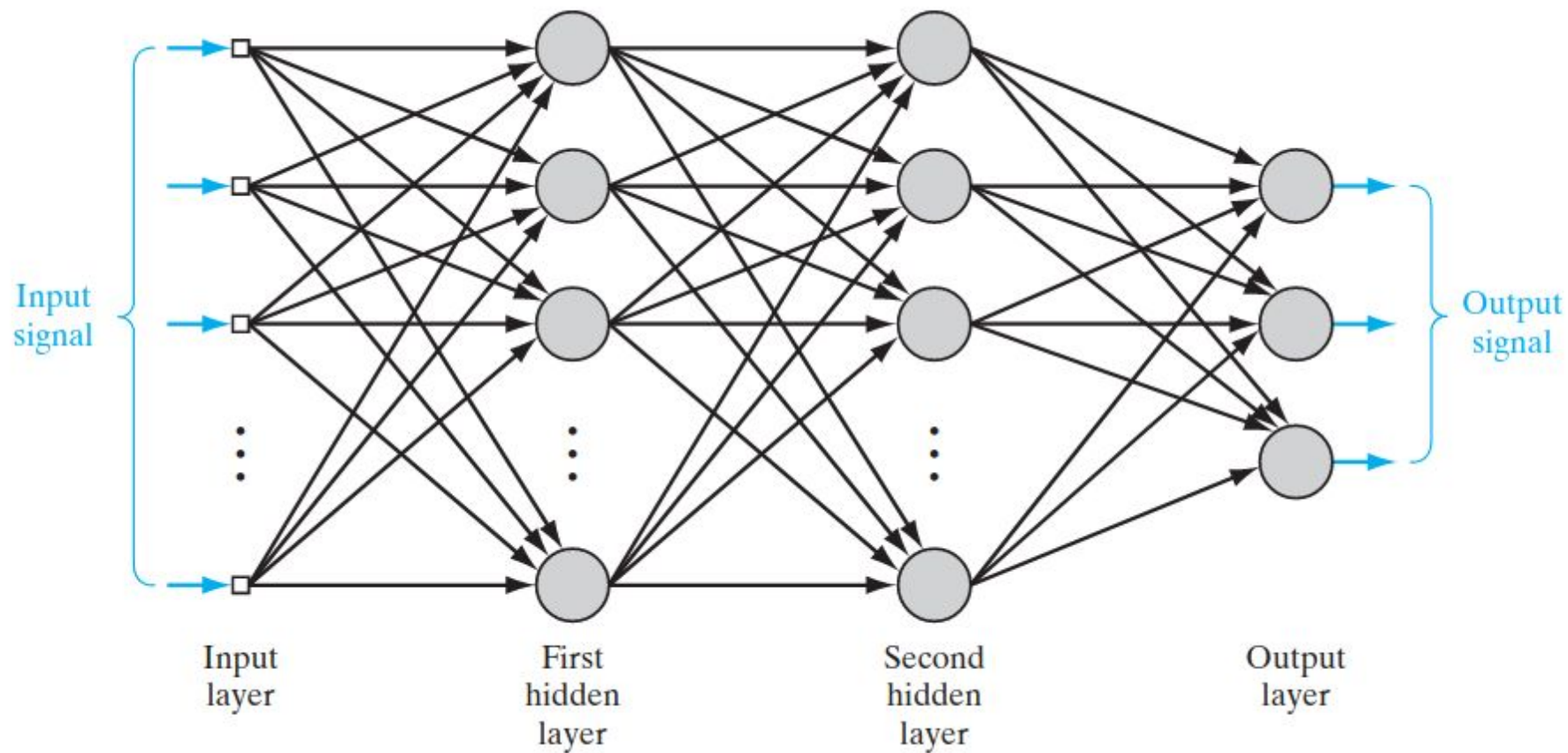
Explicación

2. Redes Neuronales : Modelos Supervisados



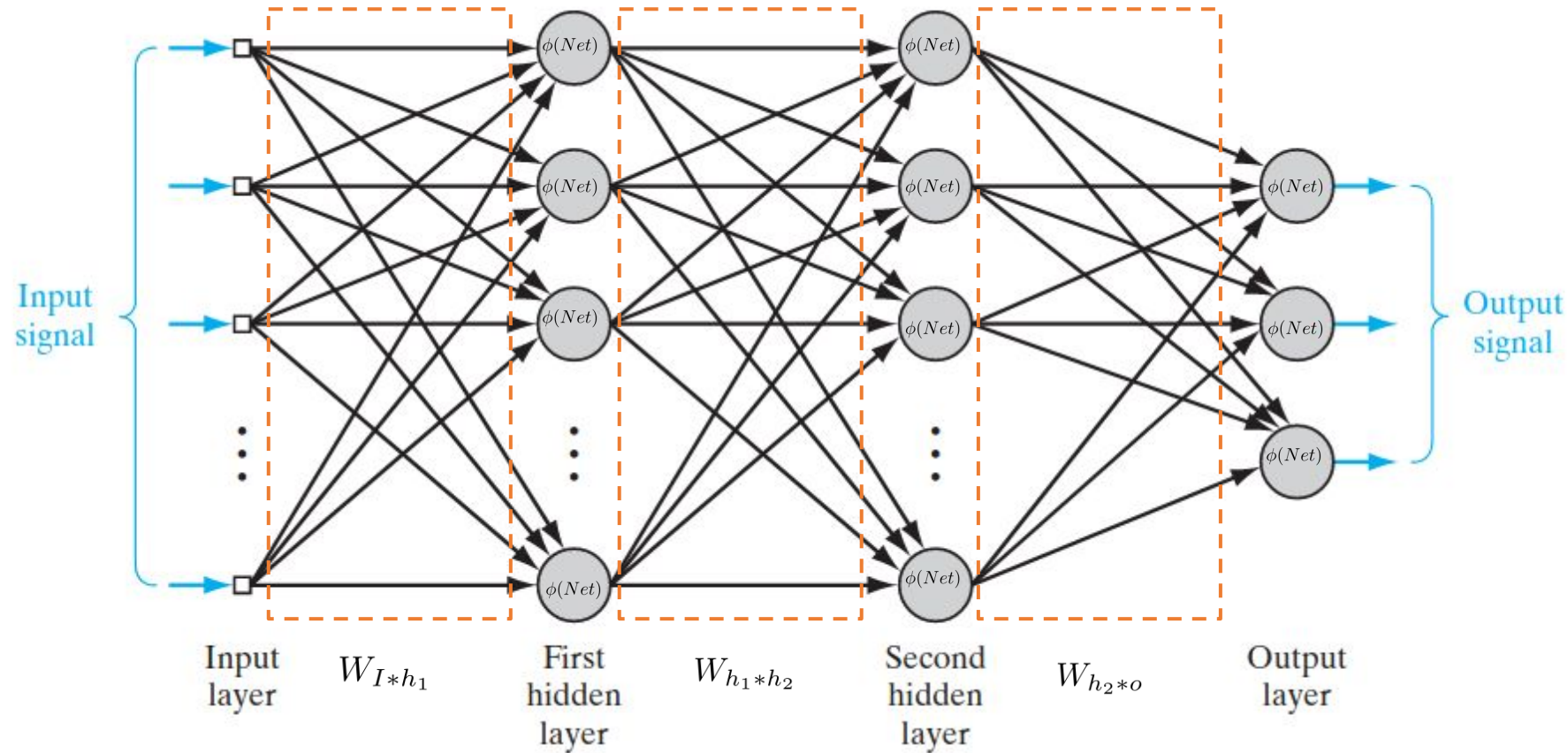
2 Multilayer Perceptron

The background of the slide is a photograph of a modern, multi-story building with a complex, geometric facade. The building features numerous balconies and large windows, and is set against a clear blue sky. The entire image is overlaid with a semi-transparent blue filter. The title '2 Multilayer Perceptron' is prominently displayed in the center-left area in a large, white, sans-serif font. The number '2' is significantly larger than the text 'Multilayer Perceptron'. On the right side of the building, the letters 'UTEC' are visible in a small, dark font.



Forward

Forward



Forward

Forward

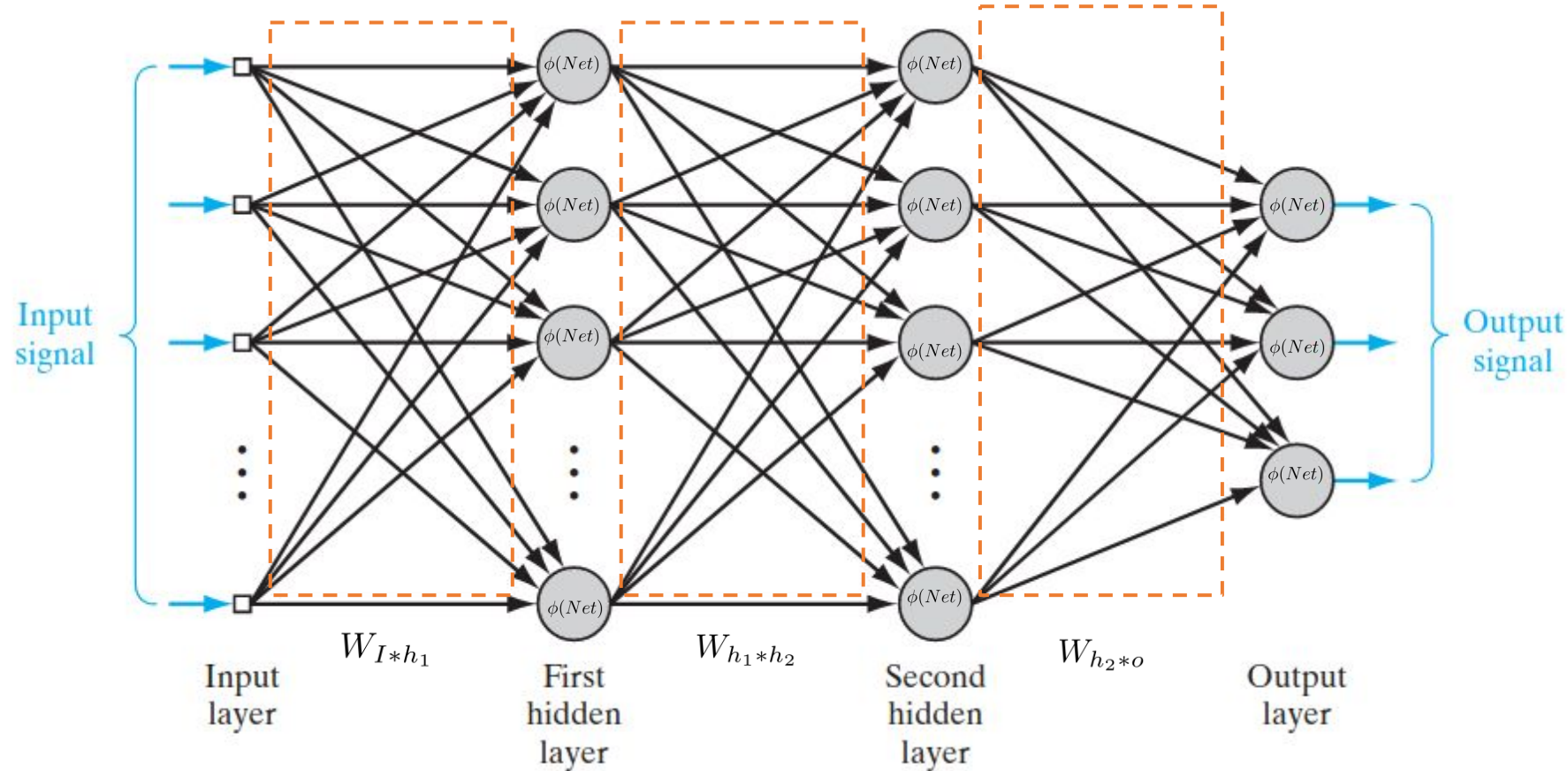
$$x_i \begin{matrix} c_1 & c_2 & c_3 & \dots & c_n \\ \boxed{} & \boxed{} & \boxed{} & \dots & \boxed{} \end{matrix} \times \begin{matrix} h_1^1 & h_2^1 & h_3^1 & \dots & h_k^1 \\ c_1 & \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ c_2 & \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ \dots & \dots & \dots & \dots & \dots \\ c_n & \boxed{} & \boxed{} & \boxed{} & \boxed{} \end{matrix} = \boxed{} \text{ f(} \boxed{} \text{ s}_1^1 \text{ s}_2^1 \text{ s}_3^1 \text{ } \dots \text{ s}_k^1 \text{) } \boxed{}$$

$$\begin{matrix} s_1^1 & s_2^1 & s_3^1 & \dots & s_k^1 \\ \boxed{} & \boxed{} & \boxed{} & \dots & \boxed{} \end{matrix} \times \begin{matrix} h_1^2 & h_2^2 & h_3^2 & \dots & h_m^2 \\ s_1^1 & \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ s_2^1 & \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ s_3^1 & \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ \dots & \dots & \dots & \dots & \dots \\ s_k^1 & \boxed{} & \boxed{} & \boxed{} & \boxed{} \end{matrix} = \boxed{} \text{ f(} \boxed{} \text{ s}_1^2 \text{ s}_2^2 \text{ s}_3^2 \text{ } \dots \text{ s}_m^2 \text{) } \boxed{}$$

$$\begin{matrix} s_1^2 & s_2^2 & s_3^2 & \dots & s_m^2 \\ \boxed{} & \boxed{} & \boxed{} & \dots & \boxed{} \end{matrix} \times \begin{matrix} O_1 & O_2 & \dots & O_z \\ s_1^2 & \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ s_2^2 & \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ s_3^2 & \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ \dots & \dots & \dots & \dots & \dots \\ s_k^2 & \boxed{} & \boxed{} & \boxed{} & \boxed{} \end{matrix} = \boxed{} \text{ f(} \boxed{} \text{ } \hat{y}_1 \text{ } \hat{y}_2 \text{ } \dots \text{ } \hat{y}_z \text{) } \boxed{} - \boxed{} \text{ y}_1 \text{ y}_2 \text{ } \dots \text{ y}_z \boxed{}$$

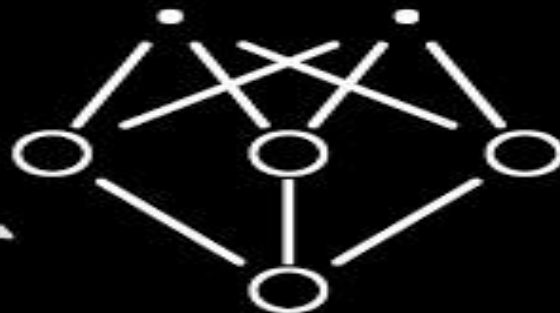
Loss Function (L₂)

¿Qué realmente está ocurriendo?

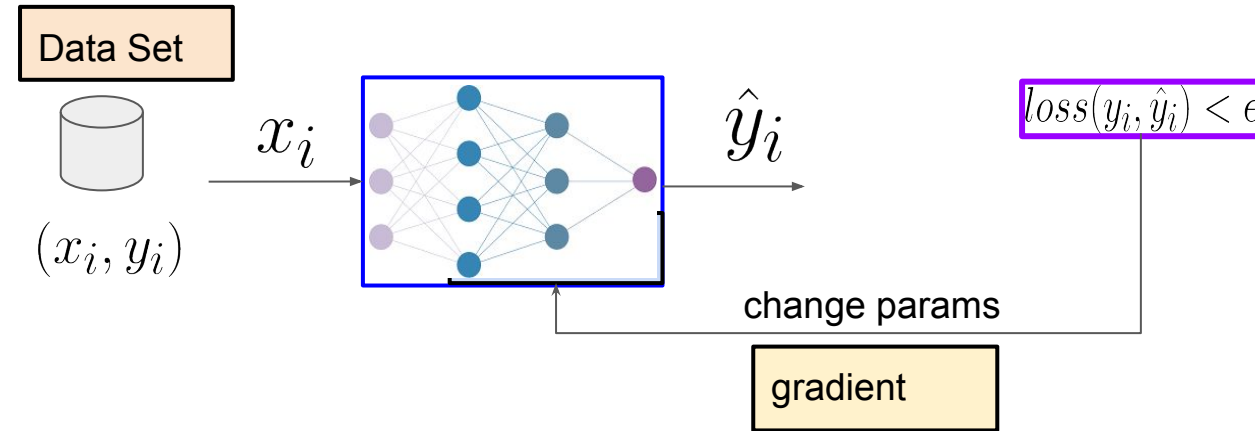


¿Qué le hace una matriz a un vector cuando se multiplican?

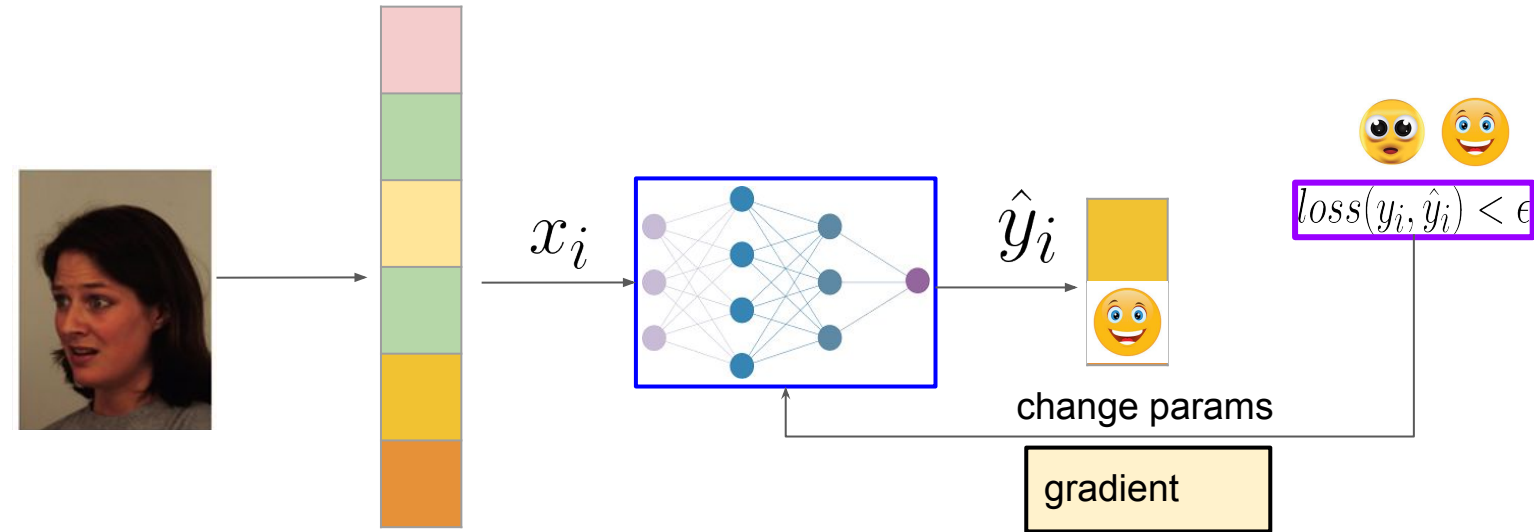
Neural Networks?



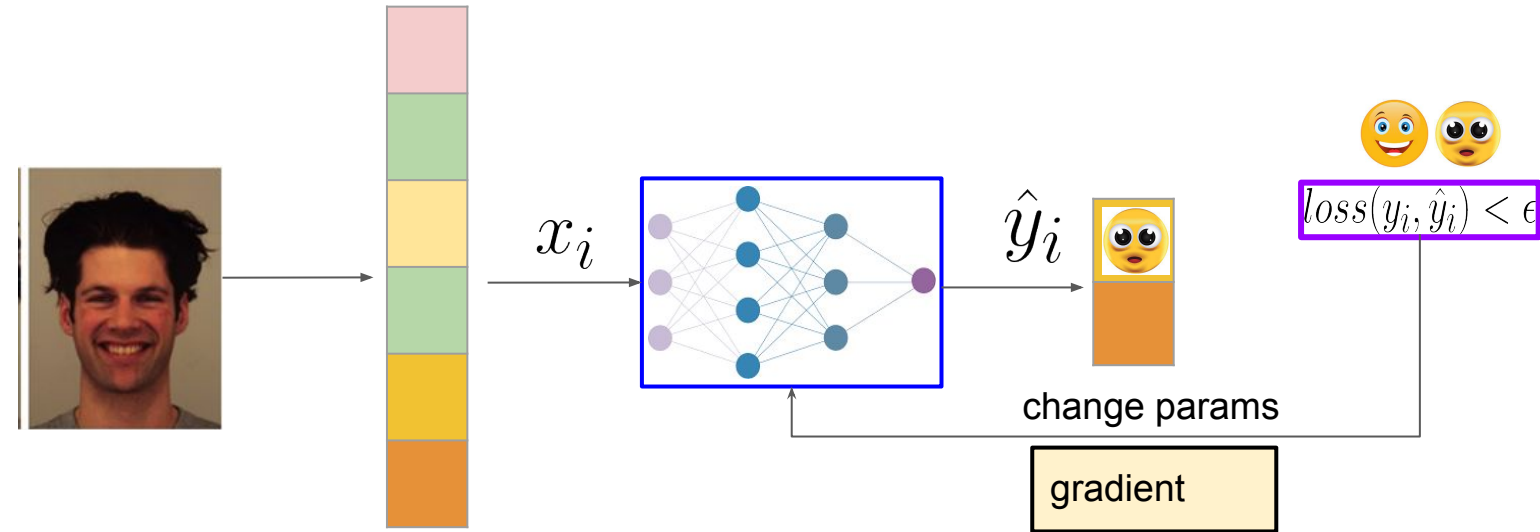
2. Redes Neuronales : Modelos Supervisados



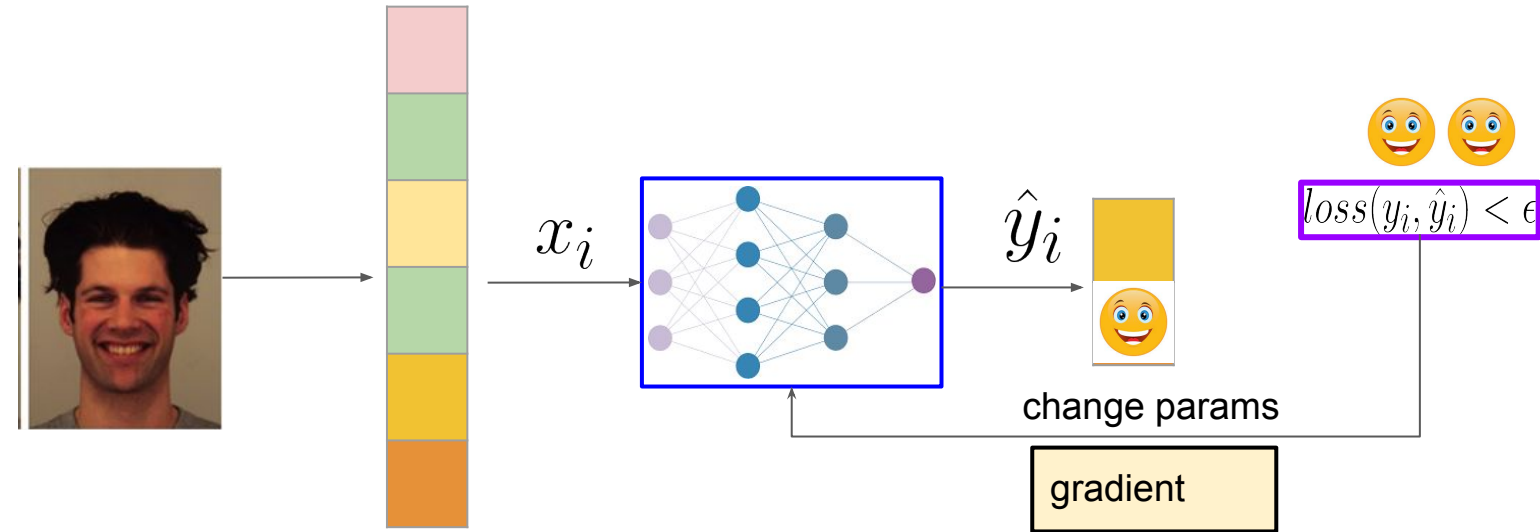
2. Redes Neuronales : Modelos Supervisados



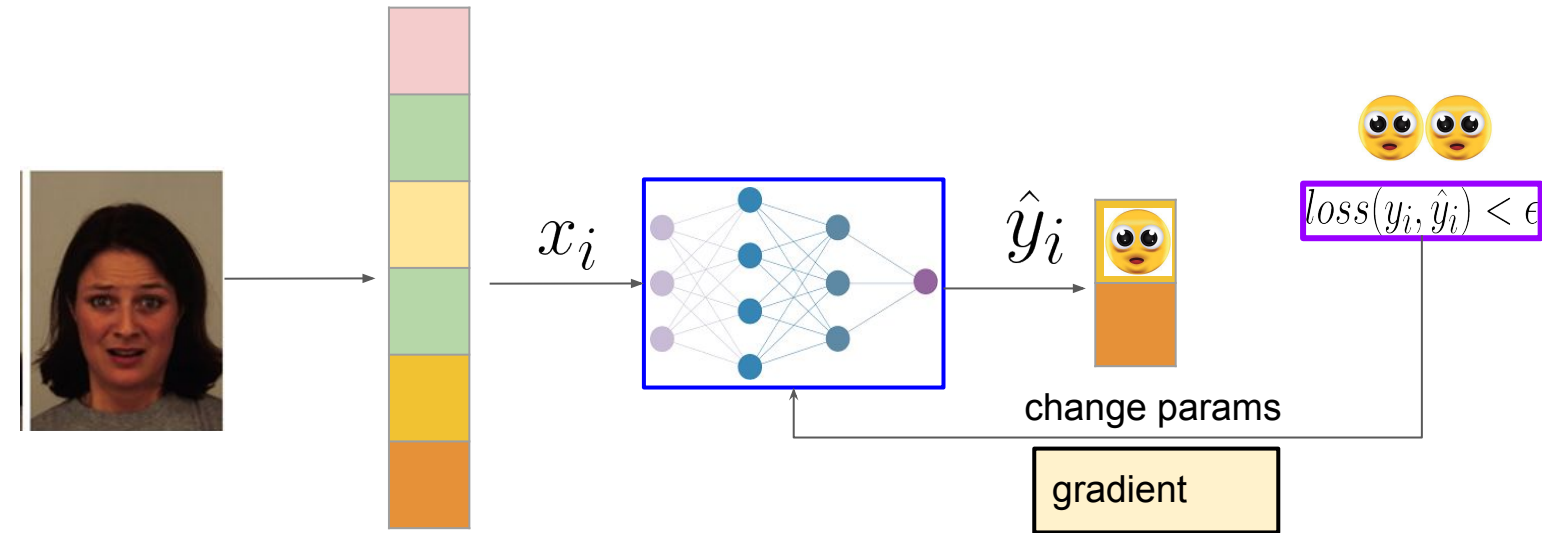
2. Redes Neuronales : Modelos Supervisados



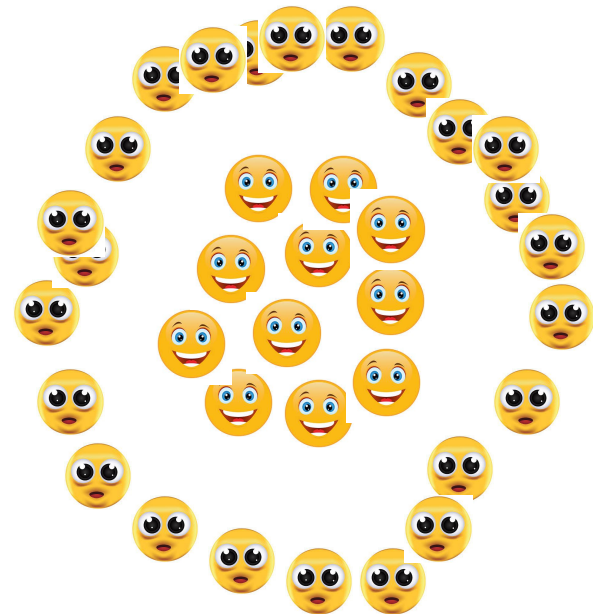
2. Redes Neuronales : Modelos Supervisados



2. Redes Neuronales : Modelos Supervisados



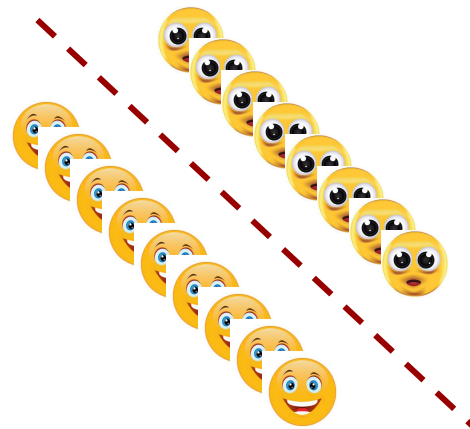
2. Redes Neuronales : Modelos Supervisados



Forward

- Incrementa dimensión
- Rota
- Escala
- Aplica no linealidad

Clasificación



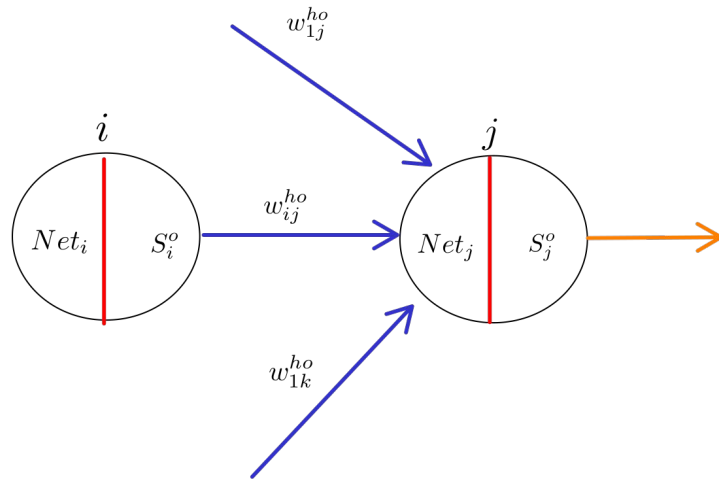
Forward

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & \dots & a_{3n} \\ a_{41} & a_{42} & a_{43} & a_{44} & \dots & a_{4n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & \dots & a_{nn} \end{pmatrix}$$



BACK PROPAGATION

Retropropagación capa hidden - output



Back Propagation : hidden - output

$$Net_j = s_1^h w_{1j}^{ho} + s_2^h w_{2j}^{ho} + \dots + s_i^h w_{ij}^{ho} + \dots + s_k^h w_{kj}^{ho}$$

$$s_j = \frac{1}{1 + e^{-Net}}$$

$$L = \sum_{i=1}^n (s_j^o - s_j^d) / 2$$

$$L = (s_1^o - s_1^d) / 2 + (s_2^o - s_2^d) / 2 + \dots + (s_j^o - s_j^d) / 2 + \dots + (s_{N_o}^o - s_{N_o}^d) / 2$$

Retropropagación capa hidden - output

Back Propagation : hidden - output

$$\frac{\partial L}{\partial w_{ij}^{ho}} = \boxed{\frac{\partial L}{\partial s_j^o}} * \boxed{\frac{\partial s_j^o}{\partial Net_j}} * \boxed{\frac{\partial Net_j}{\partial w_{ij}^{ho}}}$$

Regla de la cadena

Retropropagación capa hidden - output

Back Propagation : hidden - output

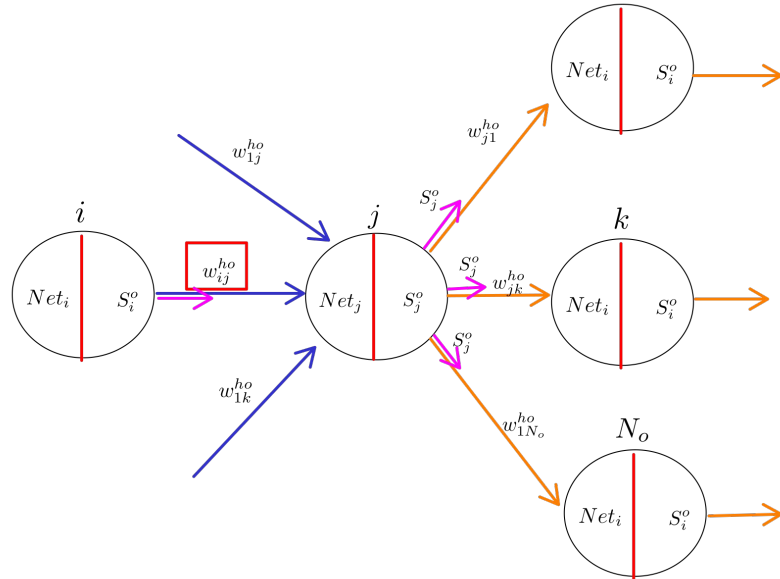
$$\frac{\partial L}{\partial s_j^o} = (s_j^o - s_j^d)$$

$$\frac{\partial s_j^o}{\partial Net_j} = s_j^o(1 - s_j^o)$$

$$\frac{\partial Net_j}{\partial w_{ij}^{ho}} = s_i^h$$

$$\frac{\partial L}{\partial w_{ij}^{ho}} = (s_j^o - s_j^d) s_j^o(1 - s_j^o) s_i^h$$

Retropropagación capa hidden - hidden



Back Propagation : hidden - hidden

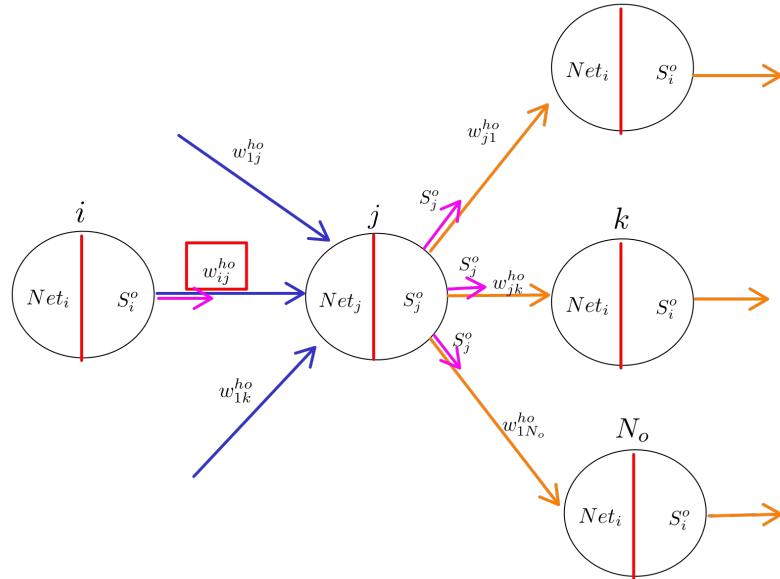
$$\frac{\partial L}{\partial w_{ij}^{hh+1}} = \sum_{k=1}^{N_o} \left(\frac{\partial L}{\partial s_k^o} * \frac{\partial s_k^o}{\partial Net_k} \right) * \frac{\partial s_j^{h+1}}{\partial Net_j} * \frac{\partial Net_j}{\partial w_{ij}^{hh+1}}$$

$$\frac{\partial L}{\partial w_{ij}^{ho}} = \left(\frac{\partial L}{\partial s_j^o} * \frac{\partial s_j^o}{\partial Net_j} \right) * \frac{\partial Net_j}{\partial w_{ij}^{ho}}$$

$$\delta_j = (s_j^o - s_j^d) s_j^o (1 - s_j^o)$$

$$\frac{\partial L}{\partial w_{ij}^{hh+1}} = \left(\sum_{k=1}^{N_o} (\delta_k) \right) * \frac{\partial s_j^{h+1}}{\partial Net_j} * \frac{\partial Net_j}{\partial w_{ij}^{hh+1}}$$

Retropropagación capa hidden - output



Back Propagation : hidden - output

$$\frac{\partial L}{\partial w_{ij}^{hh+1}} = \sum_{k=1}^{N_o} \left(\frac{\partial L}{\partial s_k^o} * \frac{\partial s_k^o}{\partial Net_k} \right) * \frac{\partial s_j^{h+1}}{\partial Net_j} * \frac{\partial Net_j}{\partial w_{ij}^{hh+1}}$$

$$\frac{\partial L}{\partial w_{ij}^{hh+1}} = \sum_{k=1}^{N_o} (\delta_k) * \frac{\partial s_j^{h+1}}{\partial Net_j} * \frac{\partial Net_j}{\partial w_{ij}^{hh+1}}$$

$$\frac{\partial L}{\partial w_{ij}^{hh+1}} = \sum_{k=1}^{N_o} (\delta_k) * s_j^{h+1} (1 - s_j^{h+1}) * s_i^h$$

7 Aplicaciones

The background of the slide is a photograph of a modern, multi-story building with a complex, geometric facade. The building features numerous balconies and large windows. The entire image is overlaid with a solid blue color. In the center, the number '7' is written in a large, white, sans-serif font, followed by the word 'Aplicaciones' in a smaller, white, sans-serif font. The building's name 'UTEC' is visible on the right side of the facade.

