Inteligencia Artificial

Clase 22: aprendizaje supervisado - clasificación

iimas

Para el día de hoy…

- Regresión logística
- Redes neuronales
- Gradiente descendente

Motivación

Hasta ahora hemos visto el caso donde las variables de salida están en los reales

En muchos de los casos esto no es así y se encuentran dados por categorías

Este problema se conoce como clasificación

La tarea

Proveer de decisiones basadas en ejemplos que permitan que nuevas observaciones sean clasificadas.

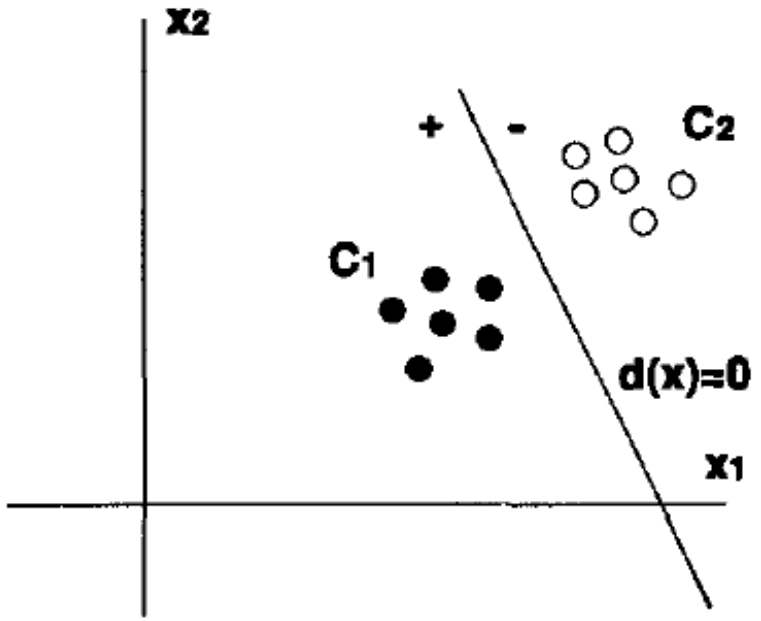
Esto puede ser visto como encontrar la “frontera”

Un ejemplo

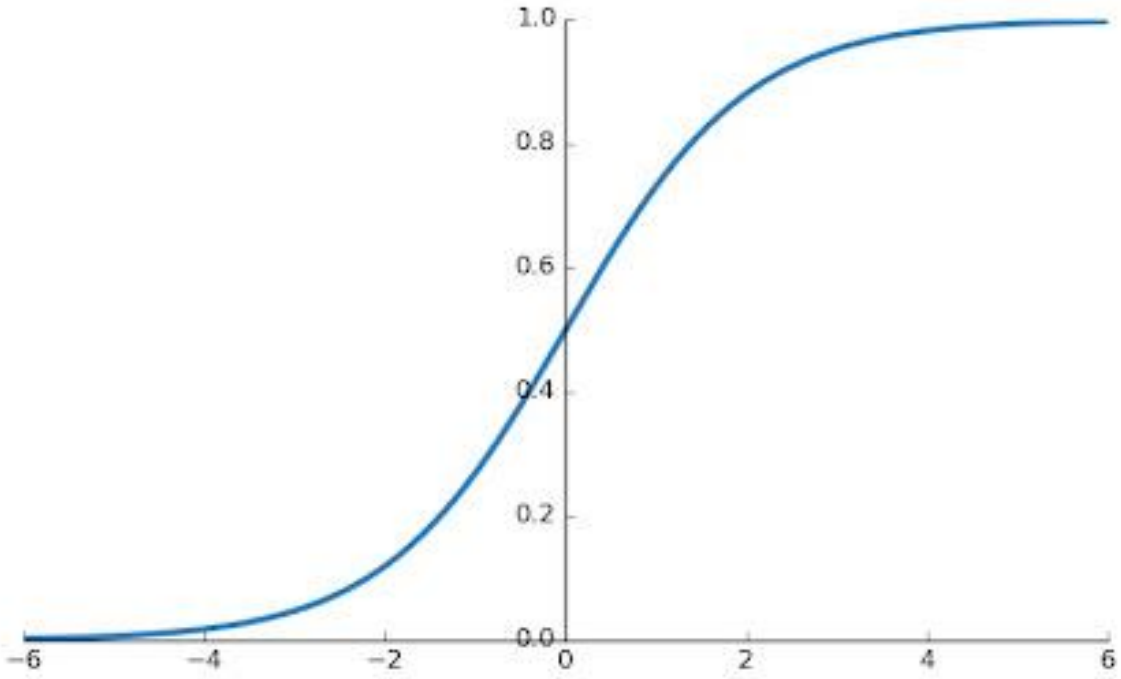
- Sean C_1 y C_2 dos clases de patrones. Cada observación es un vector $x = (x_1, x_2)^T$
- Las dos poblaciones pueden ser separadas por una línea recta
- Sea $d(x) = 0$ dicha línea, entonces

$$d(x) = w_1 x_1 + w_2 x_2 + w_3 = 0$$

- $d(x)$ puede ser "ajustado" tal que
  - $d(x) > 0$ para todo $x \in C_1$
  - $d(x) < 0$ para todo $x \in C_2$
- Entonces, $d(x)$ es una función de decisión lineal

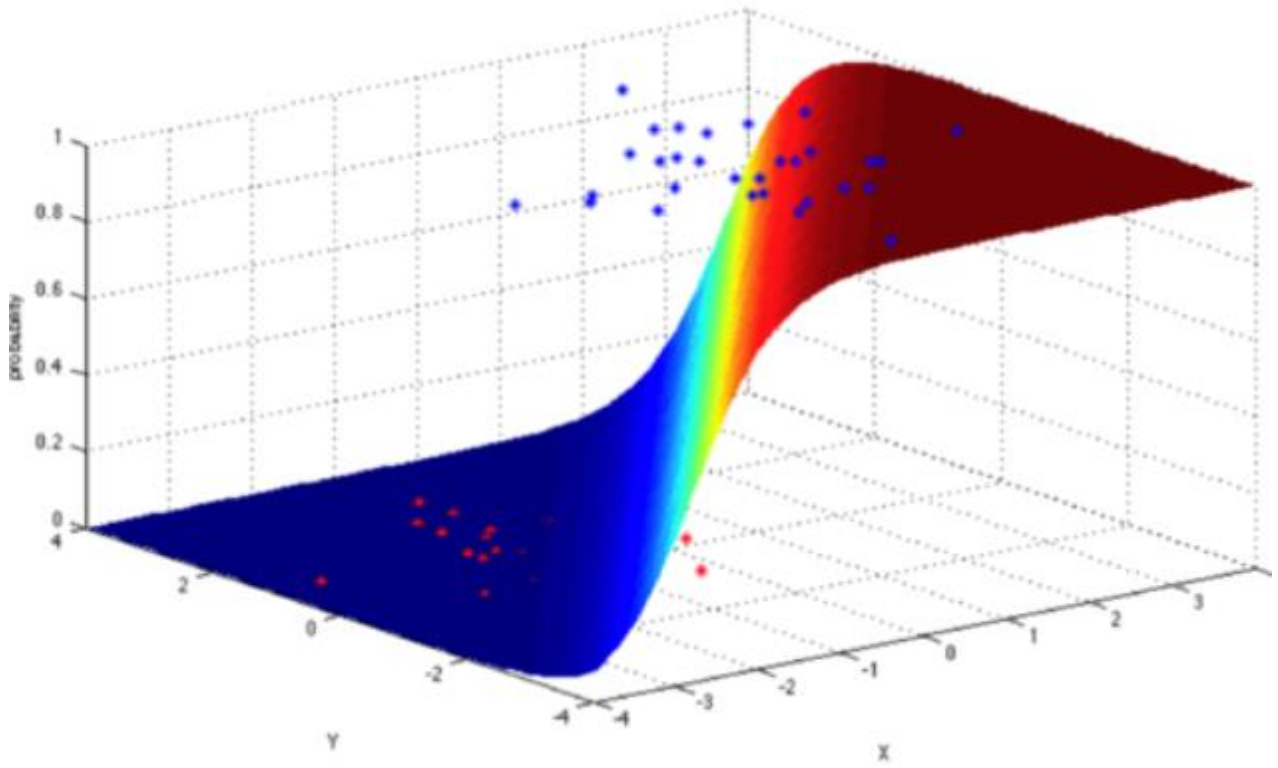


Función sigmoide



- $sigm(\eta)$ se le conoce como la función sigmoide, logísitca o logit:
- $sigm(\eta) = \frac{1}{1+e^{-\eta}} = \frac{e^{\eta}}{e^{\eta}+1}$, donde $\eta = X\theta$

Separación lineal del hiperplano



- La separación ocurre cuando:

$$P(y_i|X_i\theta) = sigm(X_i\theta) = \frac{1}{2}$$

- De forma equivalente

$$X_i\theta = 0$$

Regresión logística

- Especifica la probabilidad de una salida binaria $y_i \in \{0,1\}$ dada la entrada x_i

$$p(y|X,\theta) = \prod_{i=1}^{n} Ber(y_i|sigm(x_i\theta))$$

- Sustituyendo

$$\prod_{i=1}^{n}\left[\frac{1}{1+e^{-x\theta}}\right]^{y_i}\left[1-\frac{1}{1+e^{-x\theta}}\right]^{1-y_i}$$

Donde $x_i\theta = \theta_0 + \sum_{j=1}^{d}\theta_j x_{ij}$ y sea $\pi = 1 + e^{-x\theta}$ entonces

$$J(\theta) = -\log p(y|x,\theta) = -\sum_{i=1}^{n} y_i \log\pi_i + (1-y_i)\log(1-\pi_i)$$

El gradiente y la hessiana de la regresión logística binaria

- $\nabla J(\theta) = \sum_{i=1}^{n} x_i^T(\pi_i - y_i) = X^T(\pi - y)$
- $\nabla^2 J(\theta) = \sum_i \pi_i(1-\pi_i)x_i x_i^T = X^T diag\big(\pi_i(1-\pi_i)\big)X$

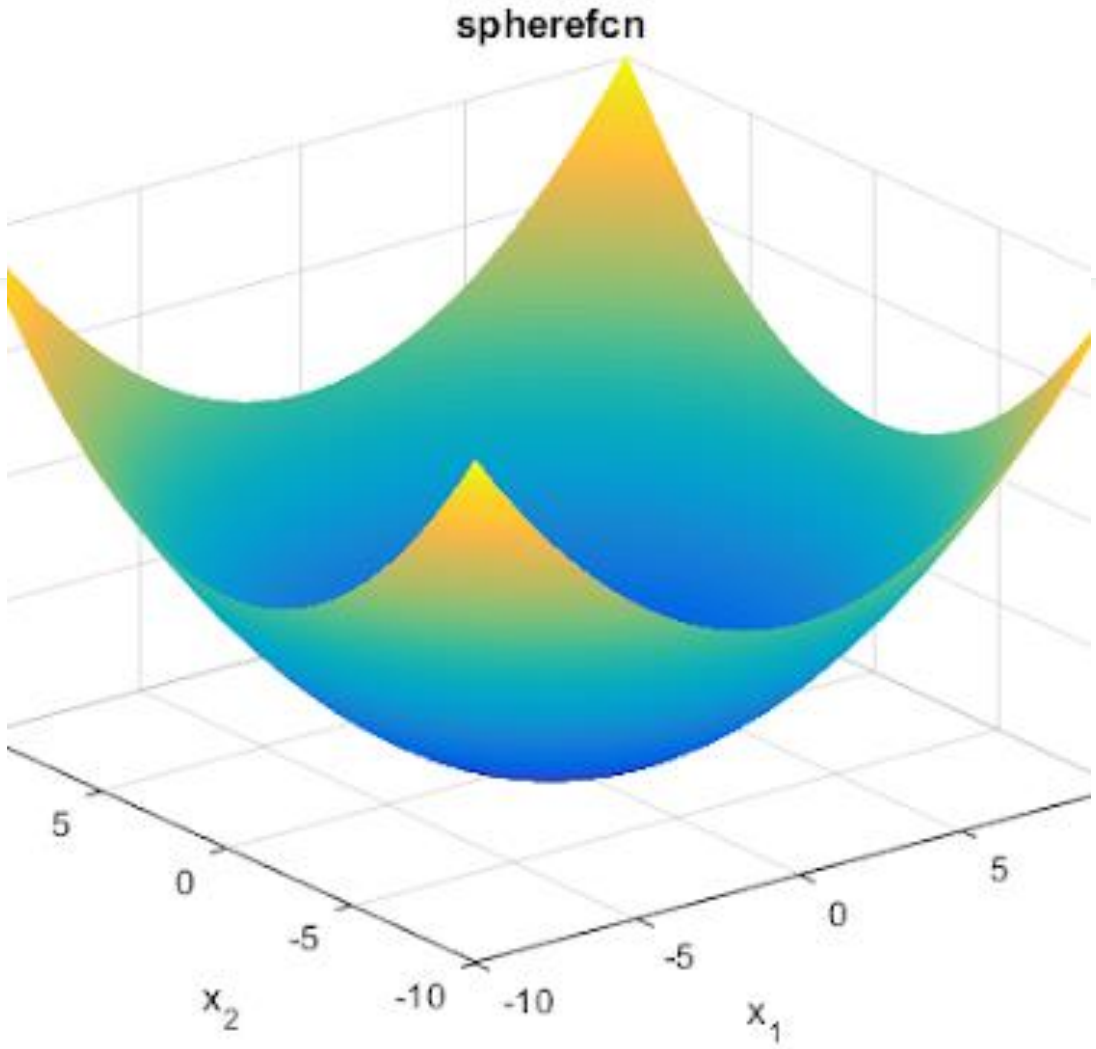
Donde $\pi_i = sigm(x_i\theta)$

Es posible probar que $\nabla^2 J(\theta)$ es positiva definida

Métodos de búsqueda directa

- Eligen una dirección p_k y buscan en esa dirección desde algún valor x_k algún punto x_{k+1} tal que $f(x_{k+1}) < f(x_k)$. La distancia a moverse puede encontrarse resolviendo

$$\min_{\alpha>0} f(x_k + \alpha p_k)$$



Gradiente descendente

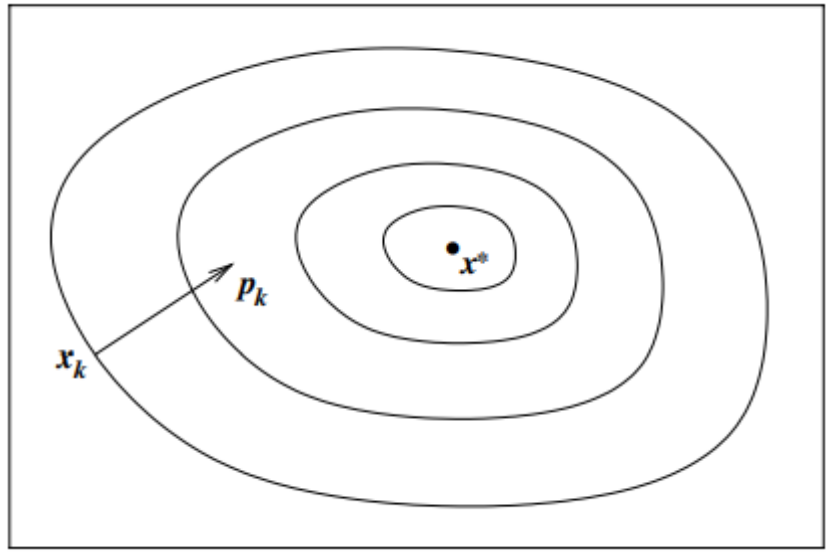
- La iteración está dada por:

$$x_{k+1} = x_k + \alpha p_k,$$

Donde $\alpha_k \in \mathbb{R}^+$ es el tamaño de paso.

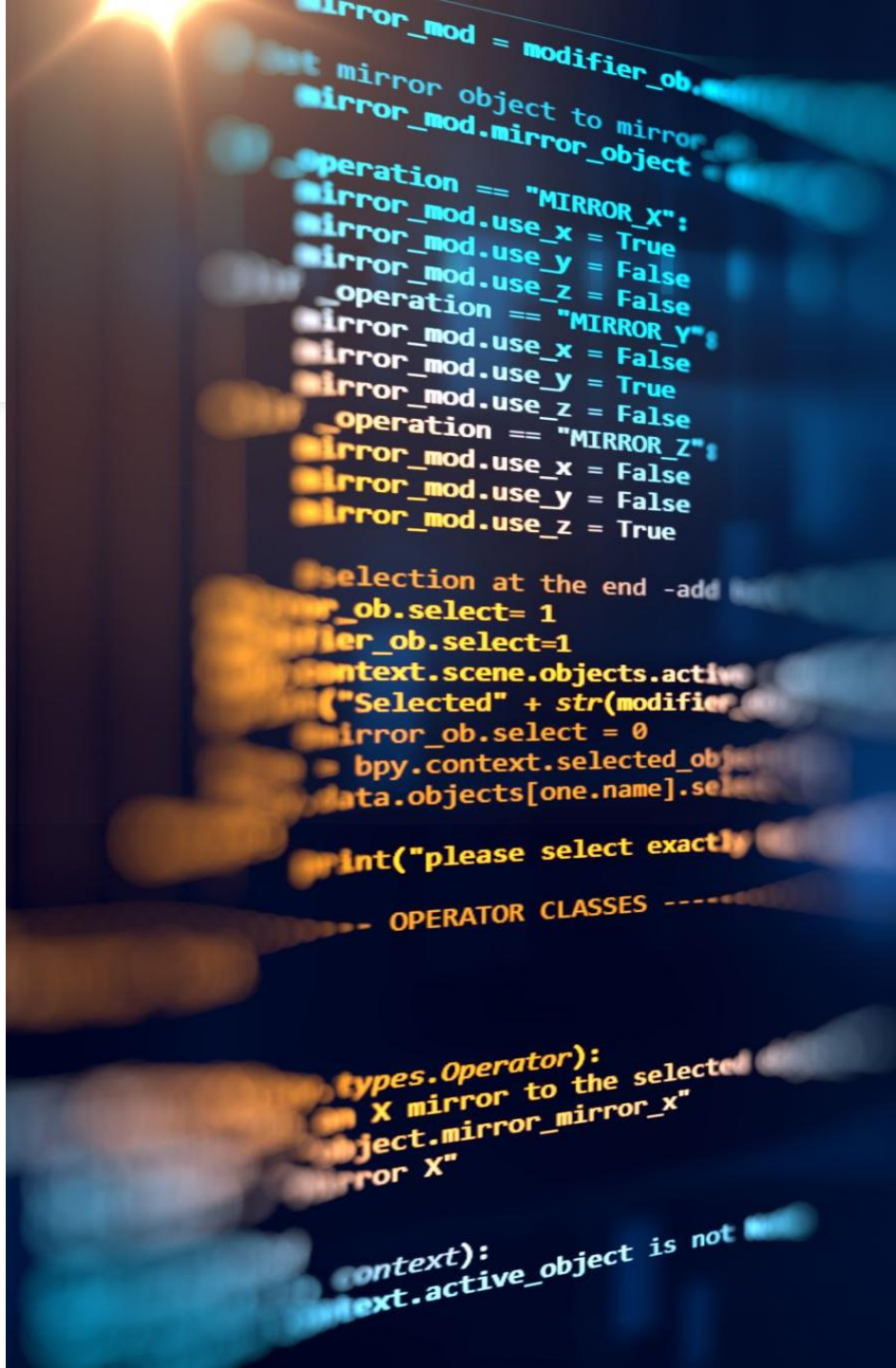
Normalmente se requiere que p_k sea una dirección descendente ($p_k^T \nabla \mathrm{f_k} < 0$).

La elección obvia es $-\nabla f_k$ que da a lugar al método de gradiente descendente.

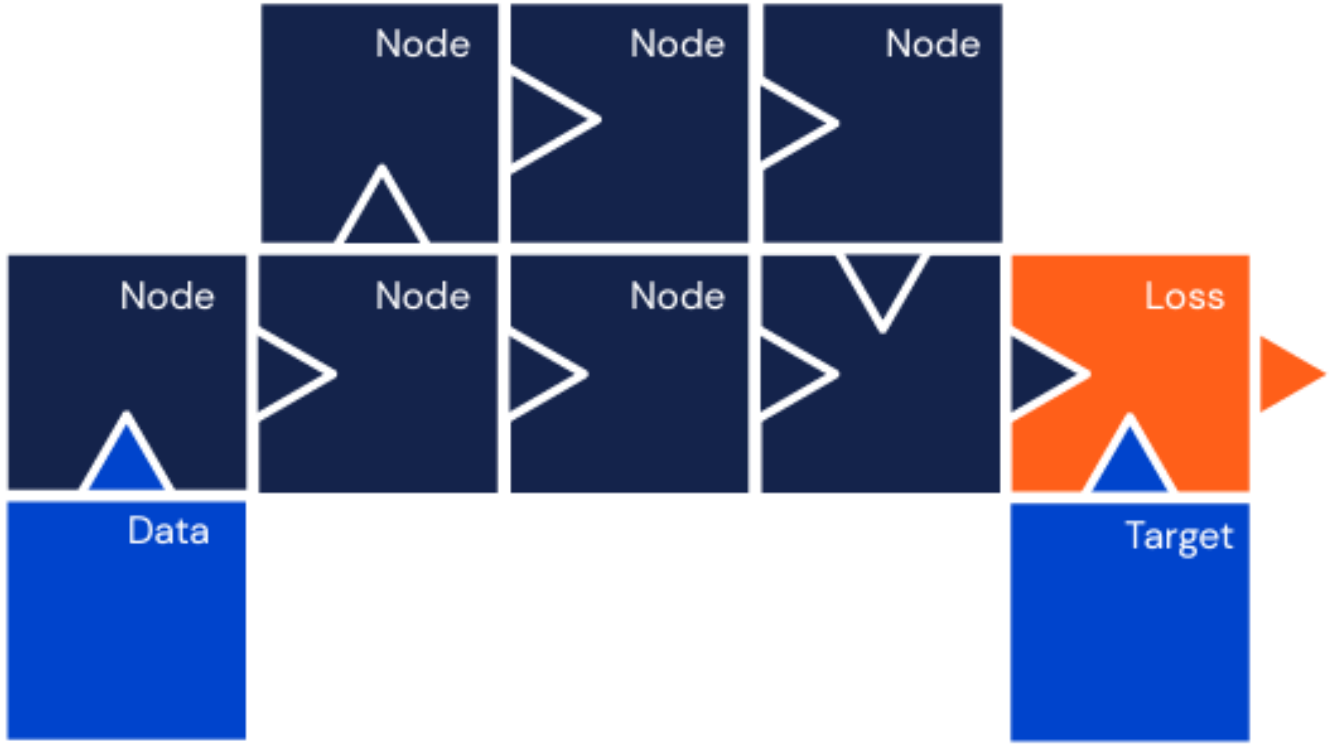


El algoritmo

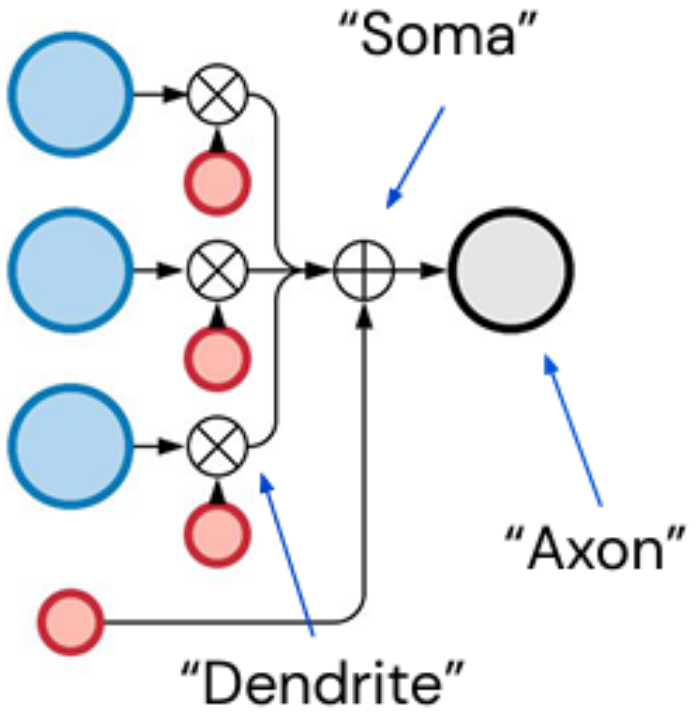
- Dado $f,\ \nabla f,\ x_0 \in \mathbb{R}^n$
- $x_k = x_0$
- Mientras condición_de_paro:
  - $p_k = -\nabla f(x_k)$
  - Calcular tamaño de paso α_k
  - $x_{k+1} = x_k + \alpha p_k$
- fin



Redes neuronales



Red neuronal artificial

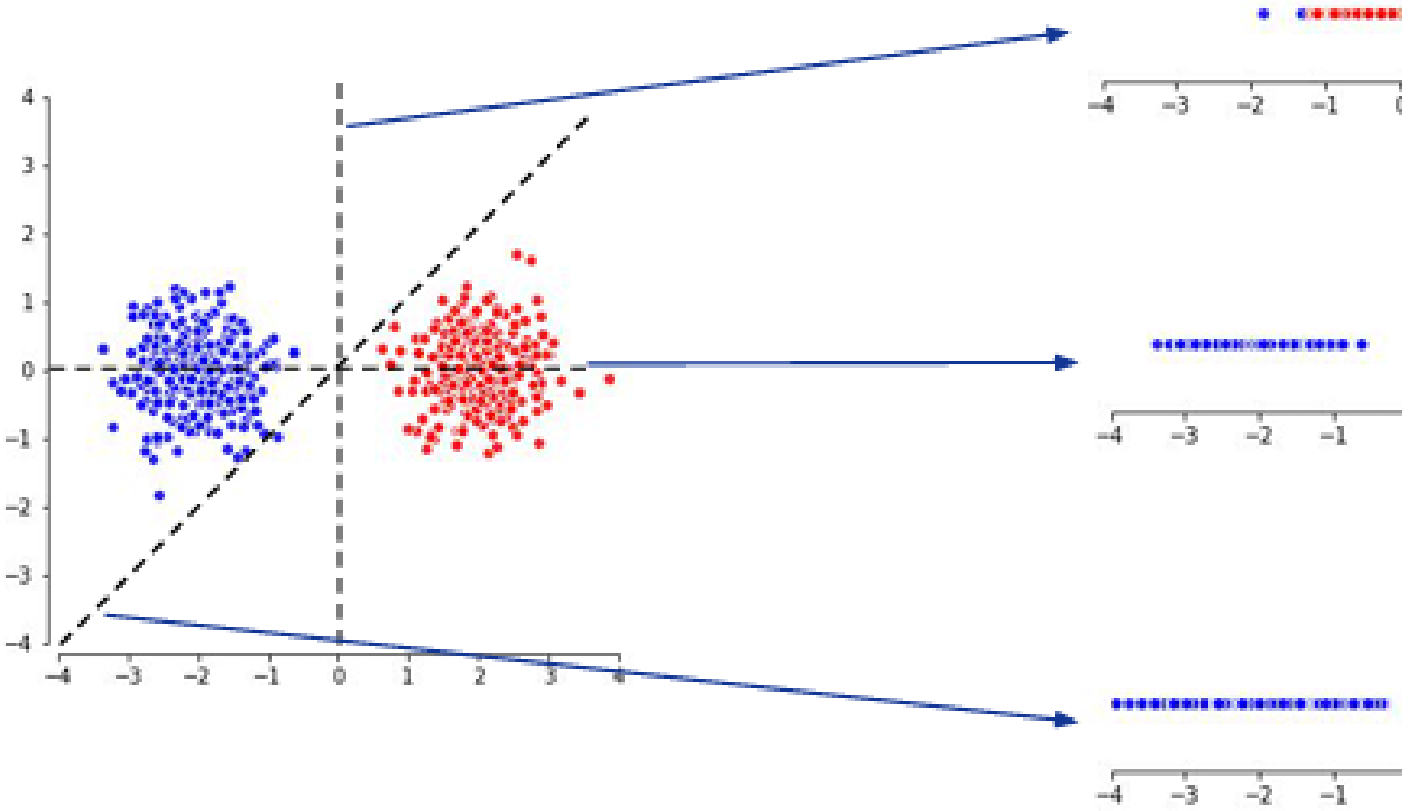


$$\sum_{i=1}^{d} \mathbf{w}_i \mathbf{x}_i$$

$$\sum_{i=0}^{d} \mathbf{w}_i \mathbf{x}_i$$

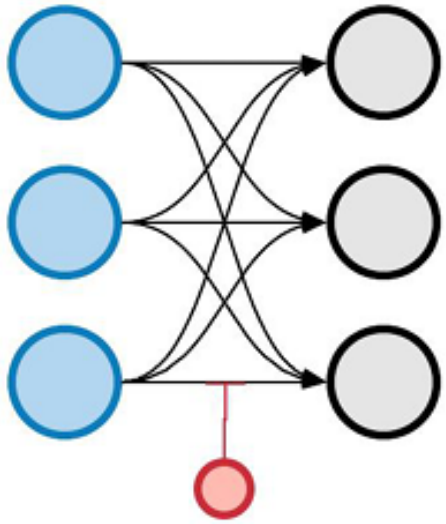
- El objetivo es reflejar observaciones neurofisiológicas y no reproducir su dinámica
- Fácil de componer
- Representa computación simple
- Tiene conexiones para inhibir y excitar

Neurona artificial



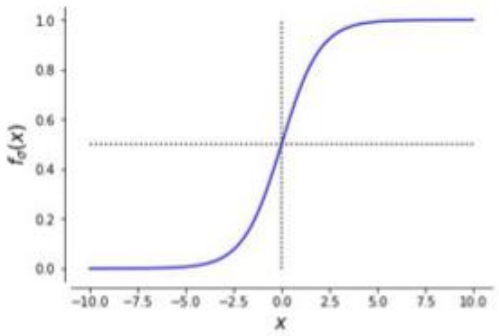
Capa lineal

- En aprendizaje máquina linear en realidad quiere decir afin
- Las neuronas en una capa son llamadas unidad
- Los parámetros son llamados pesos



$$h(\mathbf{x}, \mathbf{w}, b) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

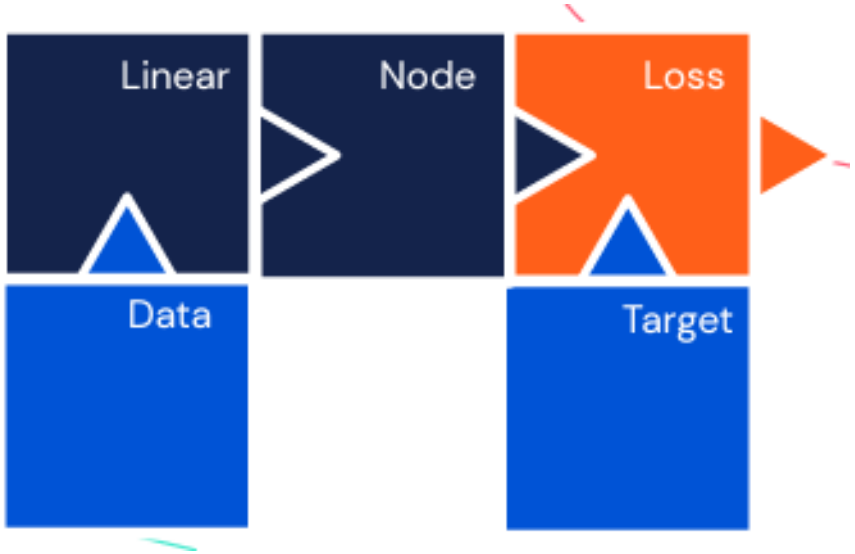
$$f_{\text{linear}}(\mathbf{x}, \mathbf{W}, \mathbf{b}) = \mathbf{W}\mathbf{x} + \mathbf{b}$$



$$f_\sigma(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{x}}}$$

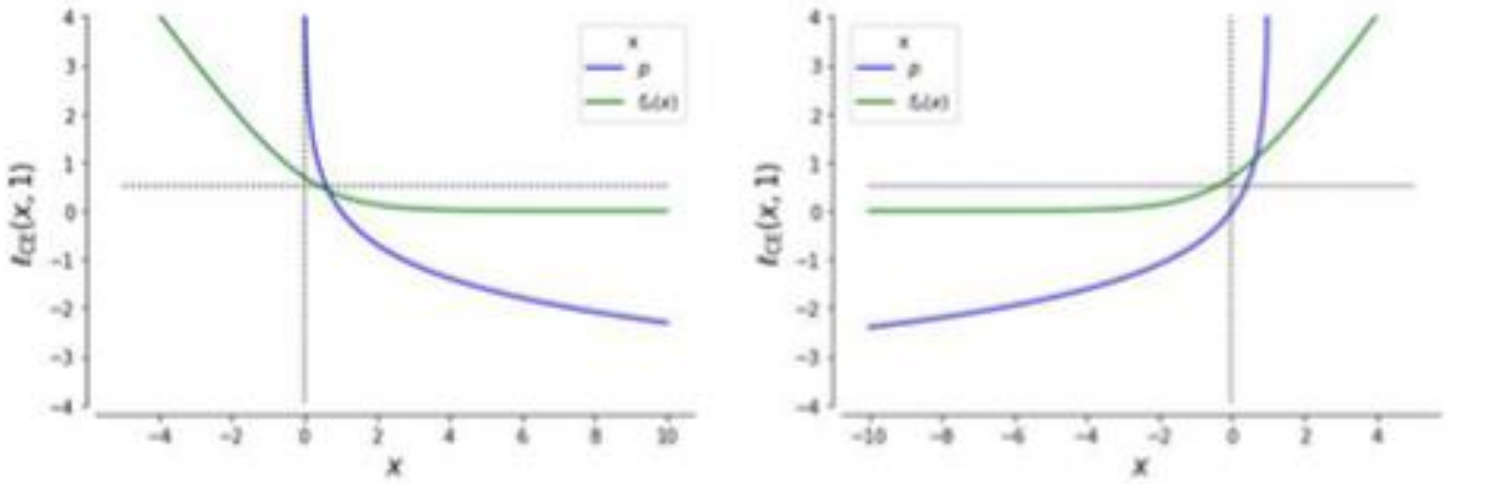
$$f_\sigma(\mathbf{x}) = \frac{e^{\mathbf{x}}}{e^{\mathbf{x}}+1}$$

Red neuronal de una capa



- Las funciones de activación son llamadas no linealidades
- Las funciones de activación se aplican a cada punto
- Producen estimaciones de probabilidad
- Se saturan
- Las derivadas desaparecen

Entropia cruzada



$$\ell_{\mathrm{CE}}(\mathbf{p}, \mathbf{t}) = -[\mathbf{t} \log \mathbf{p} + (1 - \mathbf{t}) \log(1 - \mathbf{p})]$$

- La entropía cruzada también es llamada log verosimilitud negativa o pérdida logística
- Es numéricamente inestable

Softmax

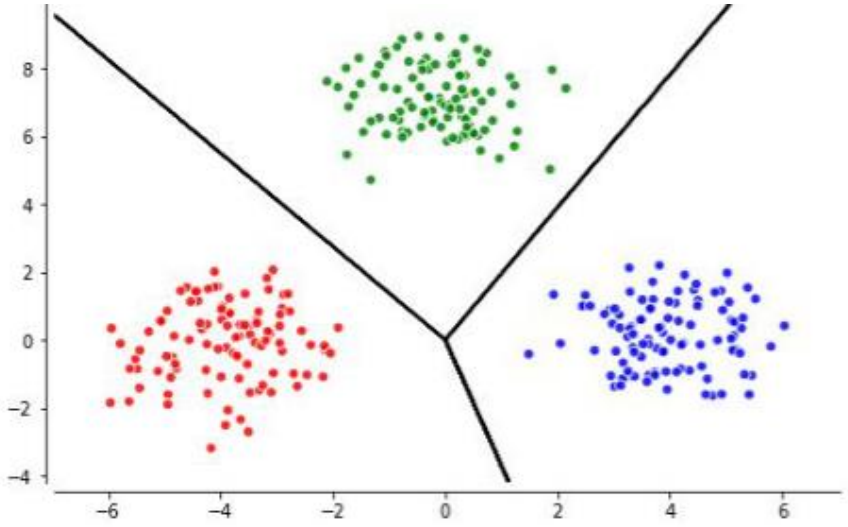
$$f_{\text{sm}}(\mathbf{x}) = \frac{e^{\mathbf{x}}}{\sum_{j=1}^{k} e^{\mathbf{x}_j}}$$

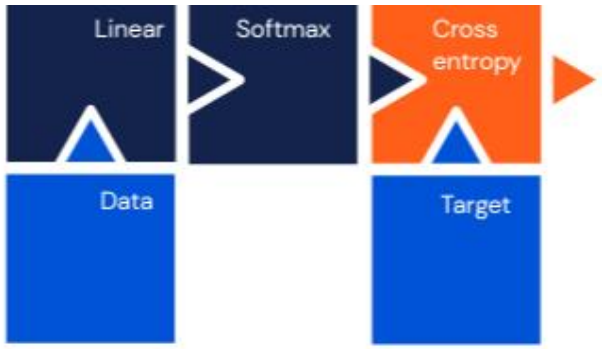
$$f_{\text{sm}}([x, 0]) = \left[\frac{e^{x}}{e^{x} + e^{0}}, \frac{e^{0}}{e^{x} + e^{0}}\right]$$

$$= [f_{\sigma}(x), 1 - f_{\sigma}(x)]$$

- Generalización de la sigmoide
- Produce estimación de probabilidad
- Se satura
- Las derivadas desaparecen

Softmax + cross entropy

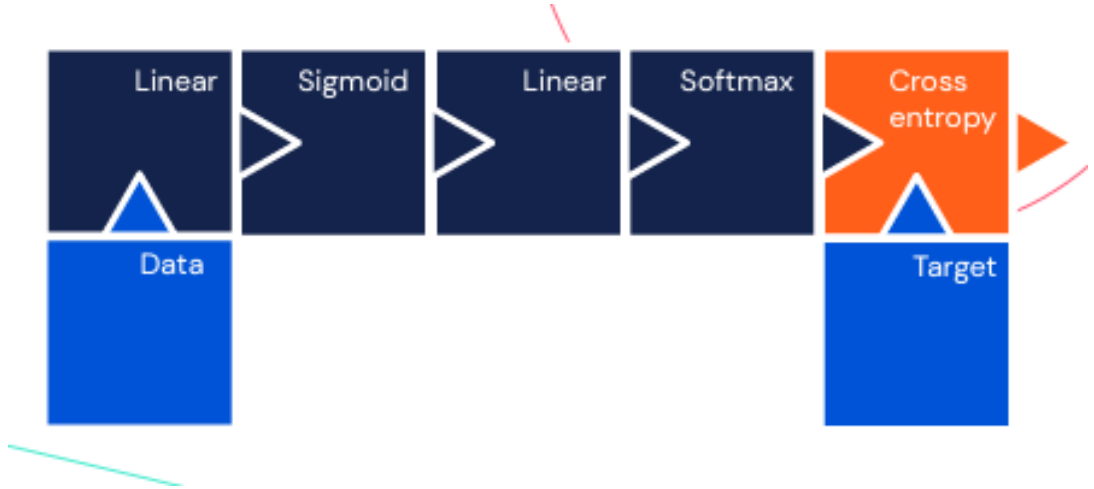


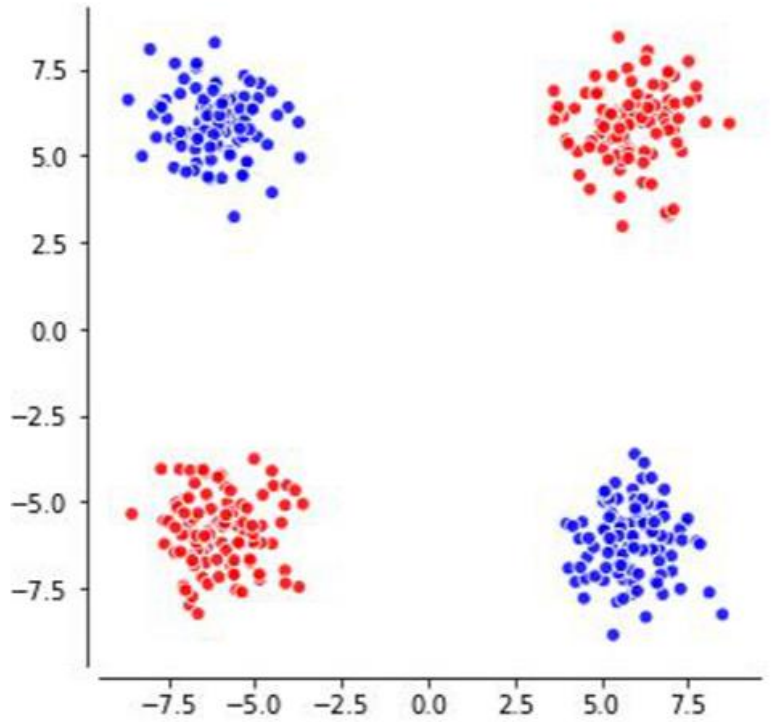


- Numéricamente estables
- Equivalente al modelo de regresión logística multinomial
- Muy utilizada en clasificación y RL

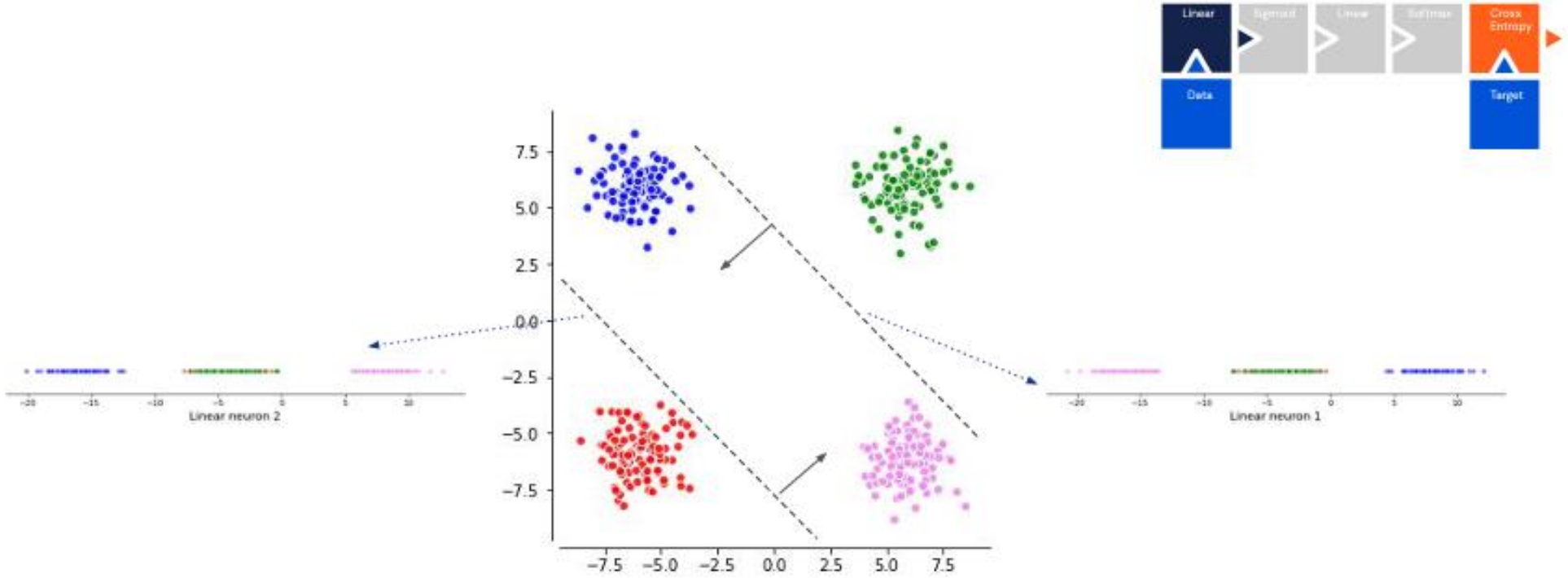
$$\ell_{\mathrm{CE}}(f_{\mathrm{sm}}(\mathbf{x}), \mathbf{t}) = -\sum_{j=1}^{k} \mathbf{t}_j \log[f_{\mathrm{sm}}(\mathbf{x}_j)] = -\sum_{j=1}^{k} \mathbf{t}_j [\mathbf{x}_j - \log \sum_{l=1}^{k} e^{\mathbf{x}_l}]$$

Redes de dos capas



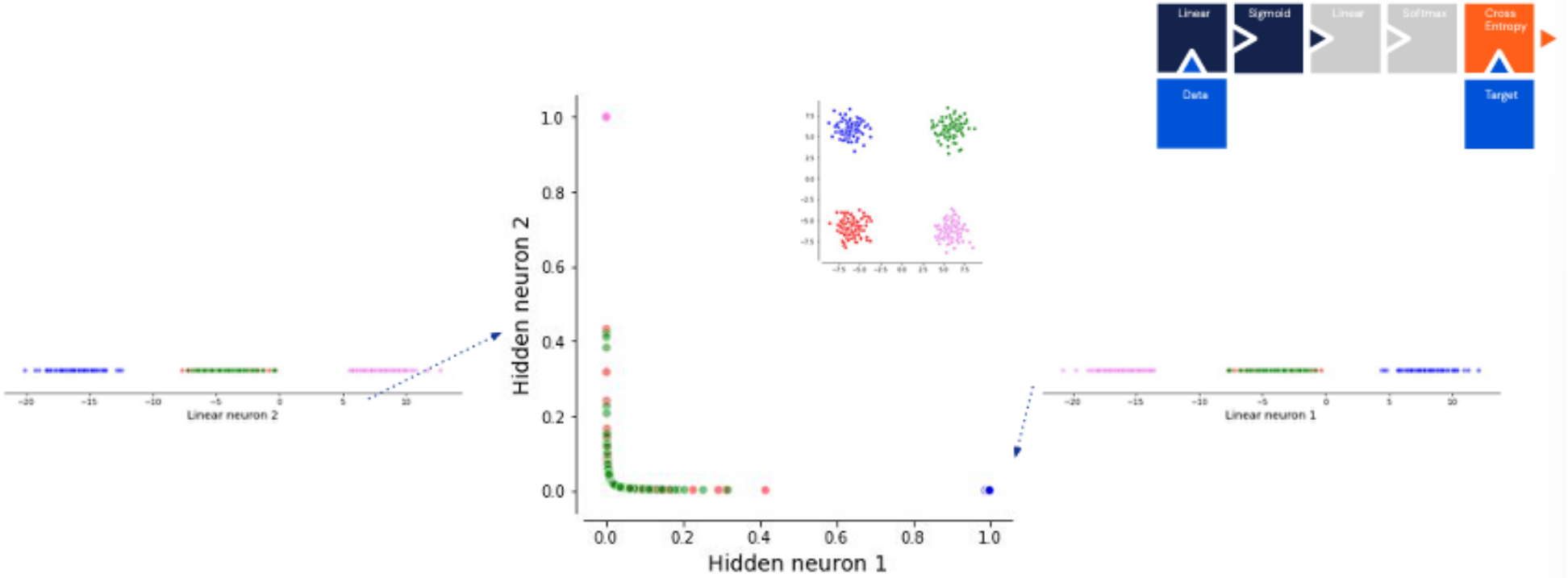


El ejemplo I



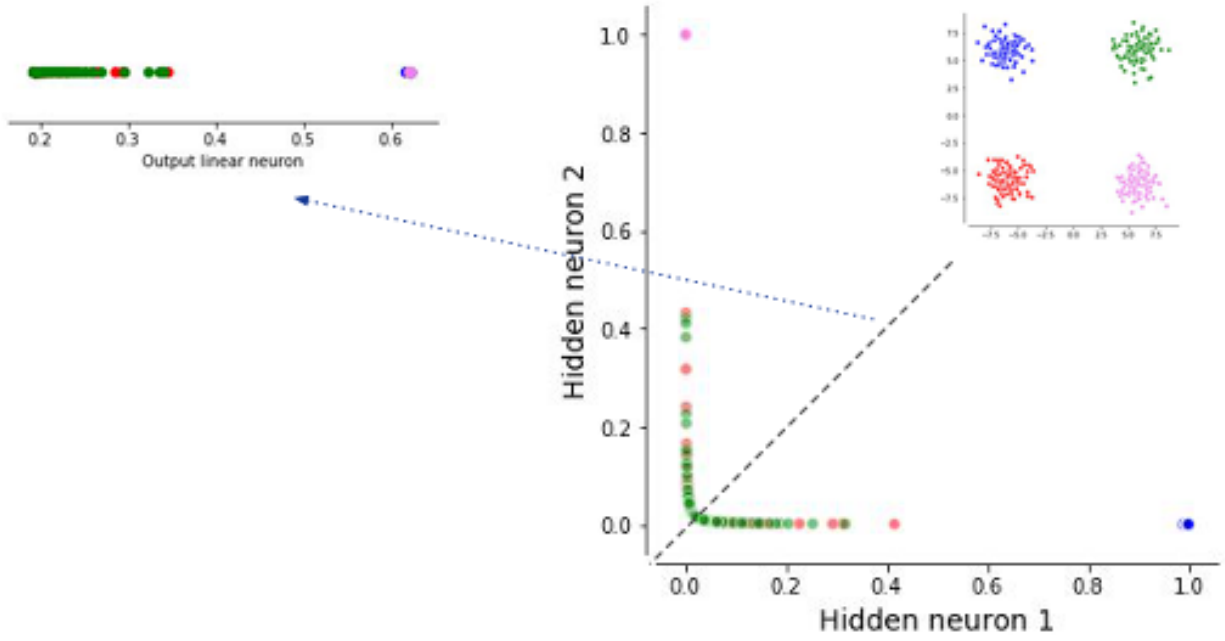
$$\mathbf{W} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -4 \\ -4 \end{bmatrix}$$

El ejemplo II



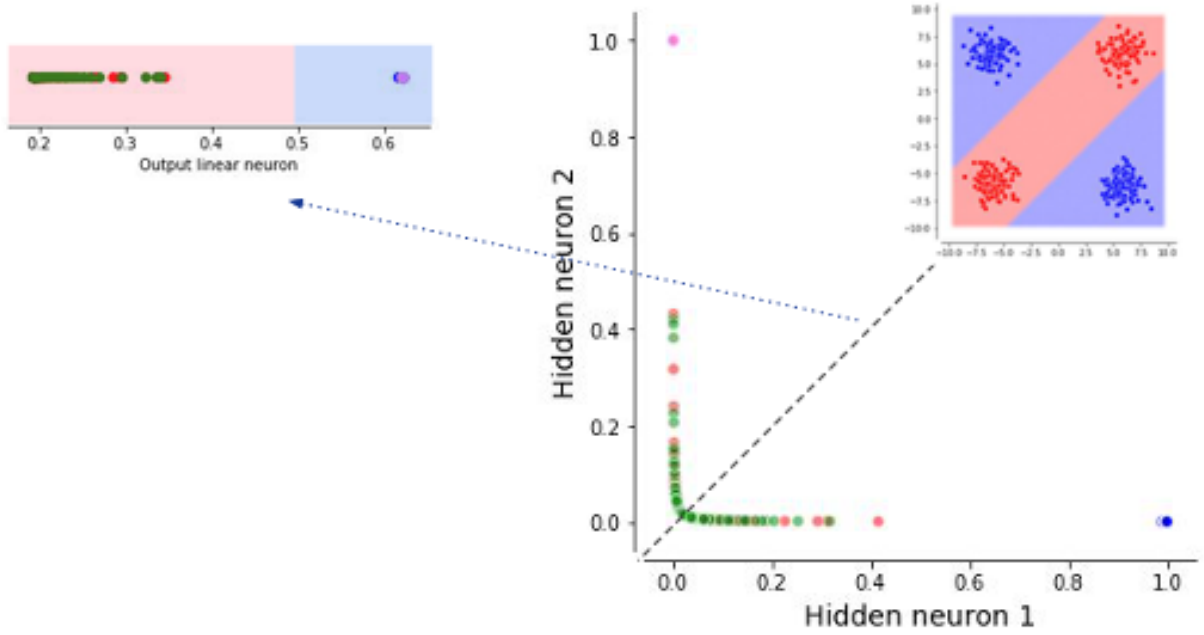
$$\mathbf{W} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -4 \\ -4 \end{bmatrix}$$

El ejemplo III



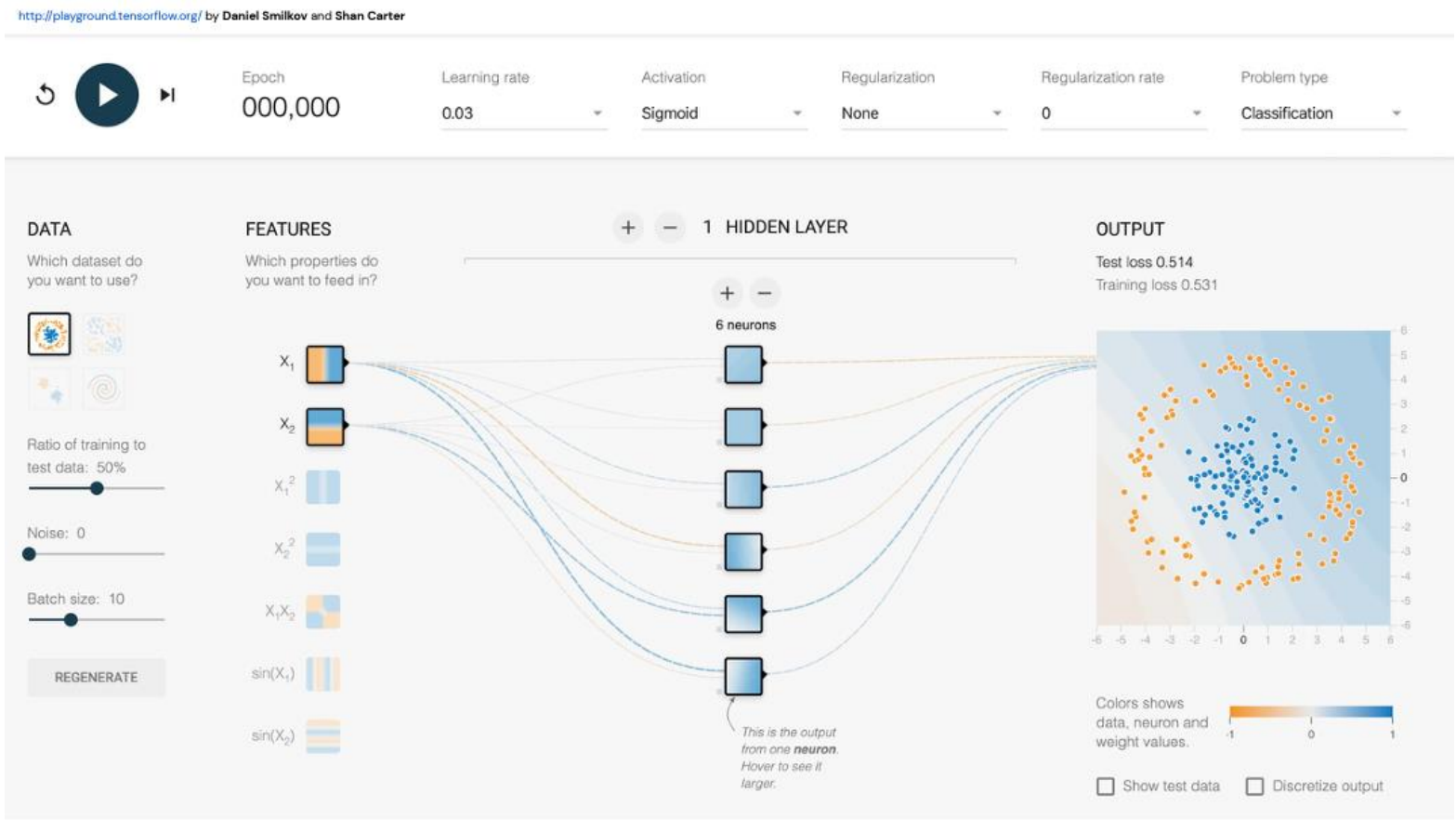
$$\mathbf{W} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -4 \\ -4 \end{bmatrix}$$

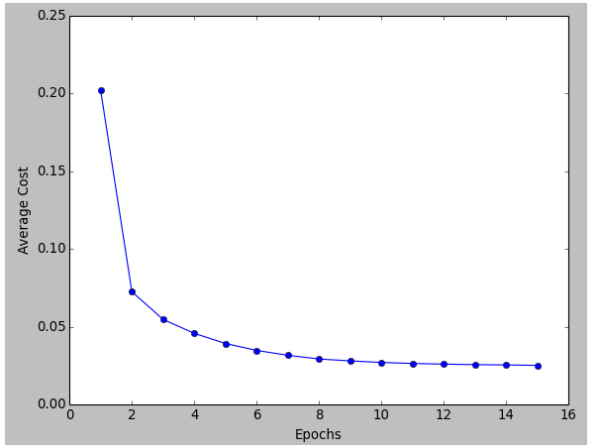
El ejemplo IV

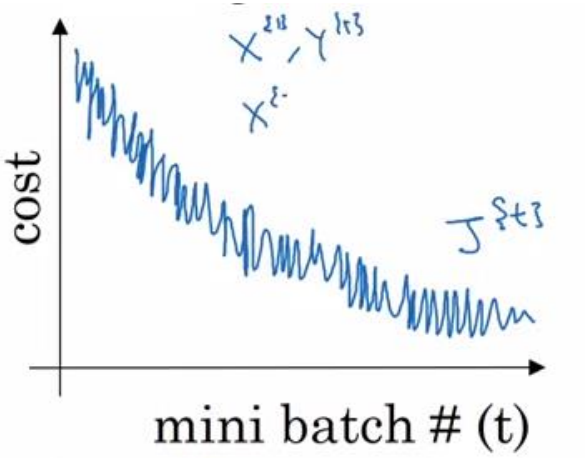


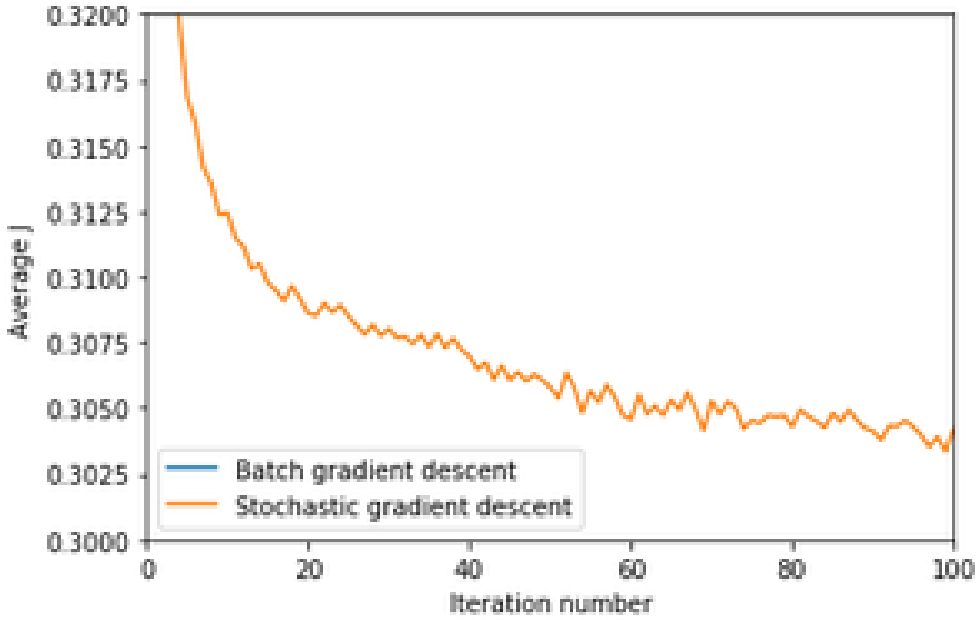
$$\mathbf{W} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -4 \\ -4 \end{bmatrix}$$

Para ver más


http://playground.tensorflow.org/ by Daniel Smilkov and Shan Carter
Epoch
000,000
Learning rate
0.03
Activation
Sigmoid
Regularization
None
Regularization rate
0
Problem type
Classification
DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE
FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
6 neurons
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.514
Training loss 0.531
Colors shows data, neuron and weight values.
Show test data
Discretize output







Aprendizaje en línea

- Batch

$$\theta_{k+1} = \theta_k + \eta \sum_{i=1}^{n} x_i^T (y_i - x_i \theta_k)$$

- Online

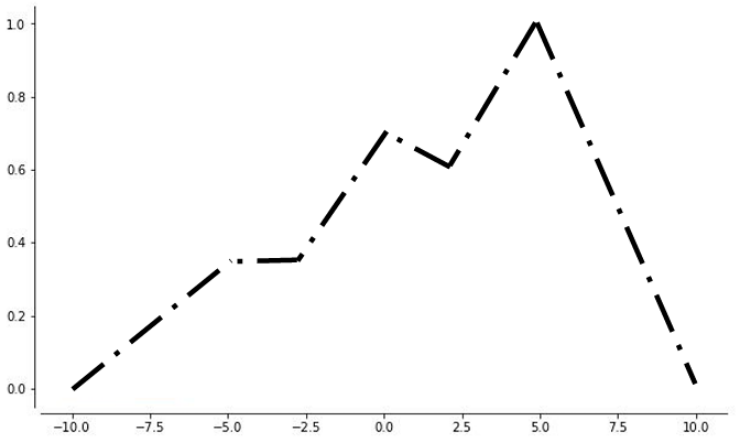
$$\theta_{k+1} = \theta_k + \eta x_k^T (y_k - x_k \theta_k)$$

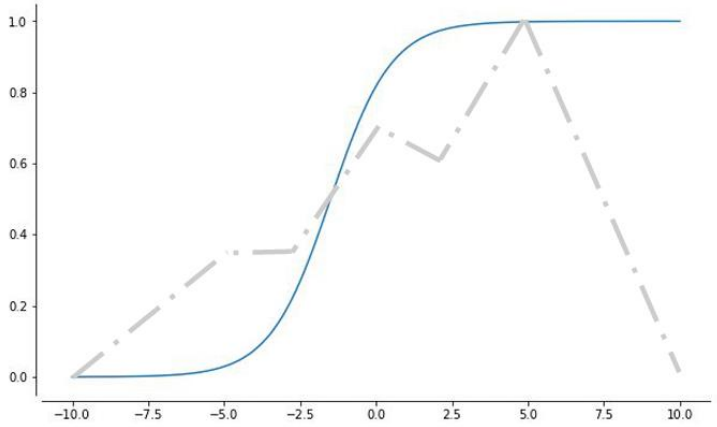
- Mini-batch

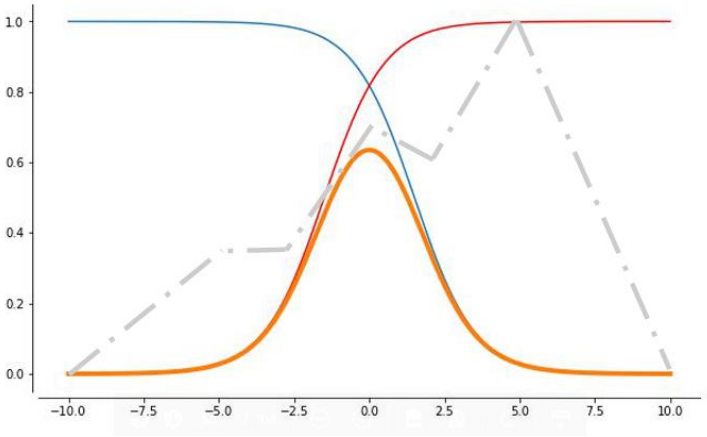
$$\theta_{k+1} = \theta_k + \eta \sum_{i=1}^{20} x_i^T (y_i - x_i \theta_k)$$

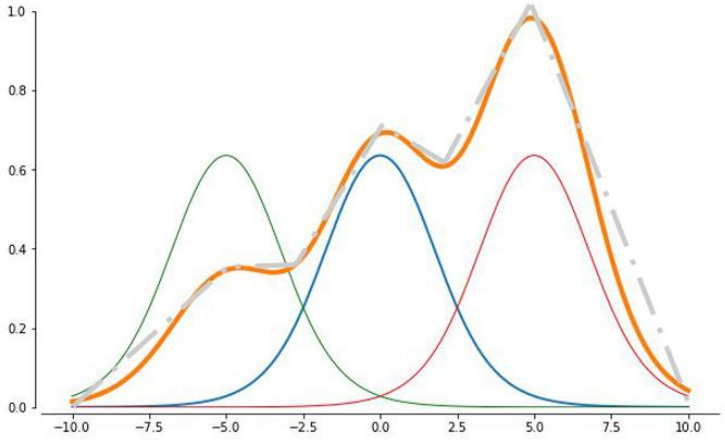
Teorema de aproximación universal

- Para cualquier función continua desde un hipercubo $[0,1]^d$ a los números reales, y para cualquier épsilon positivo, existe una red neuronal basada en sigmoide con una capa oculta que obtiene a lo más un error épsilon en el espacio de funciones
- Redes grandes puede aproximar pero no representar cualquier función suave







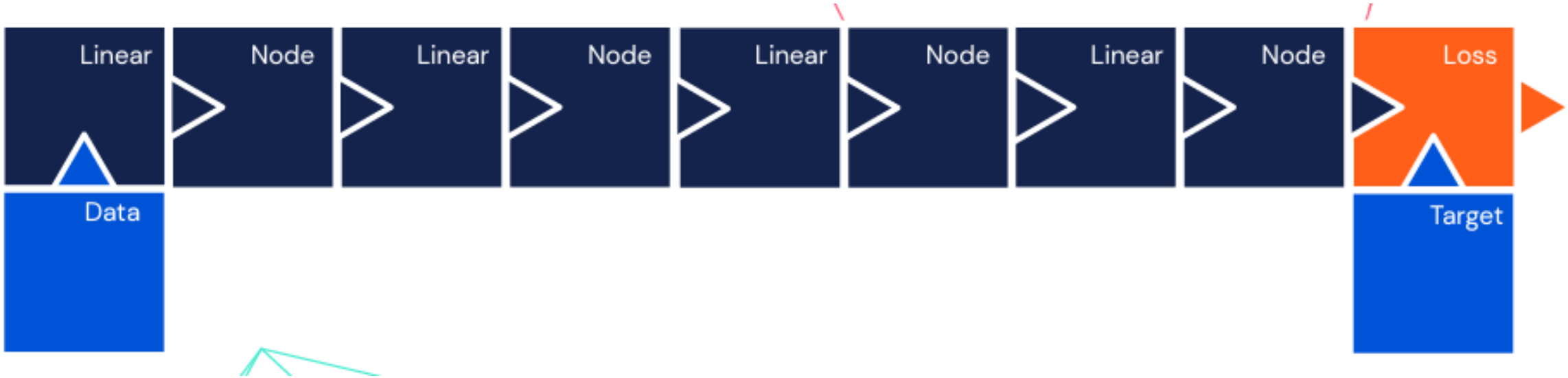


Intuición

Hace 20 años…

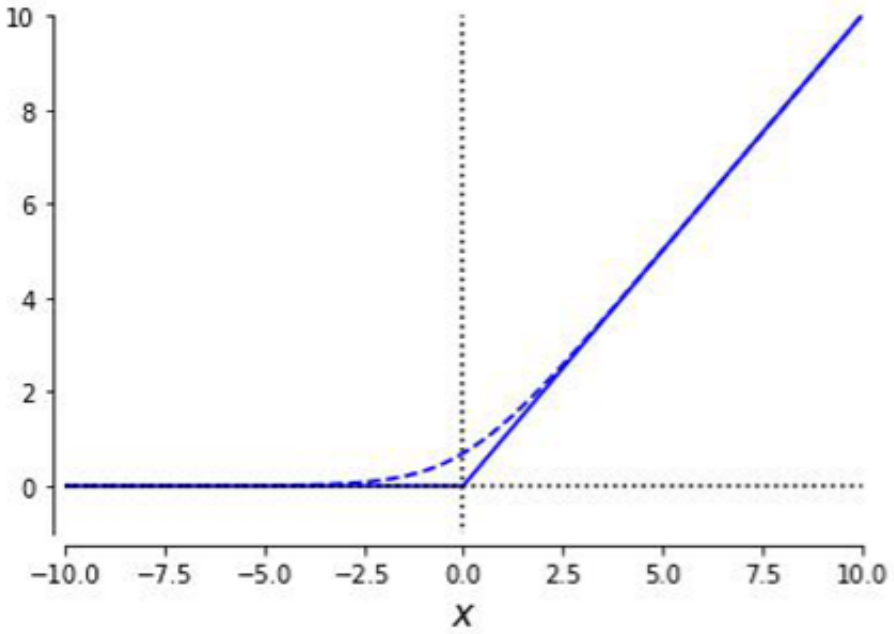
- Solo se utilizaba una capa adicional al modelo de regresión logística
- El enfoque estaba en optimización convexa
- Uso de pocos datos

Redes neuronales profundas



Rectified Linear Unit (ReLU)

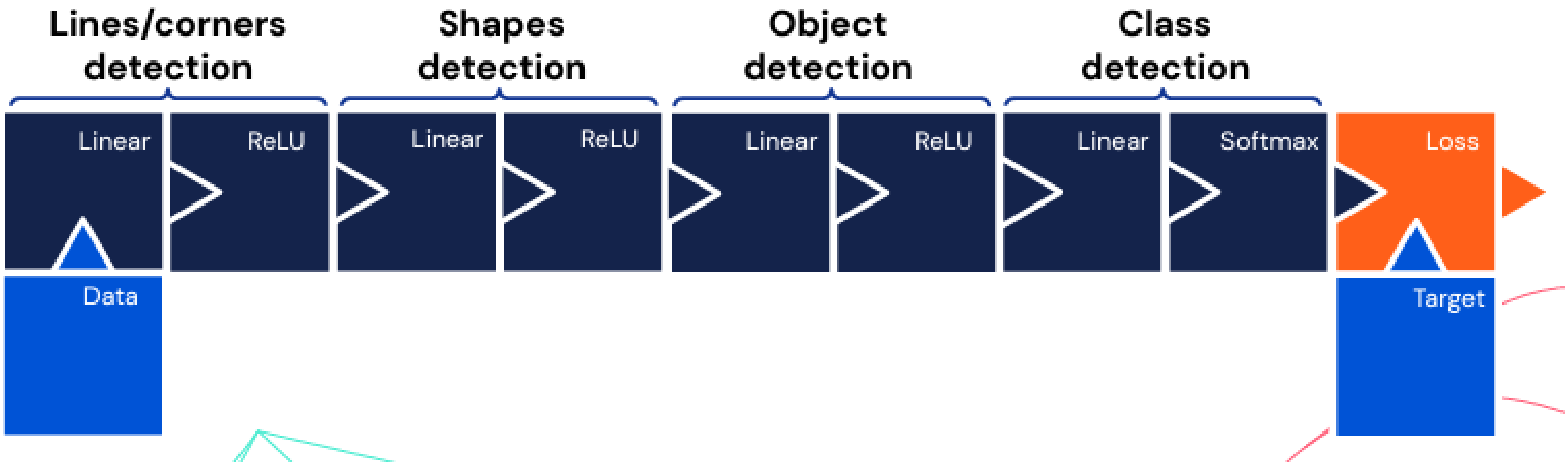
- Una de las funciones más utilizadas
- Las derivadas no desaparecen
- Puede ocurrir que neuronas mueran
- Técnicamente no es diferenciable en 0



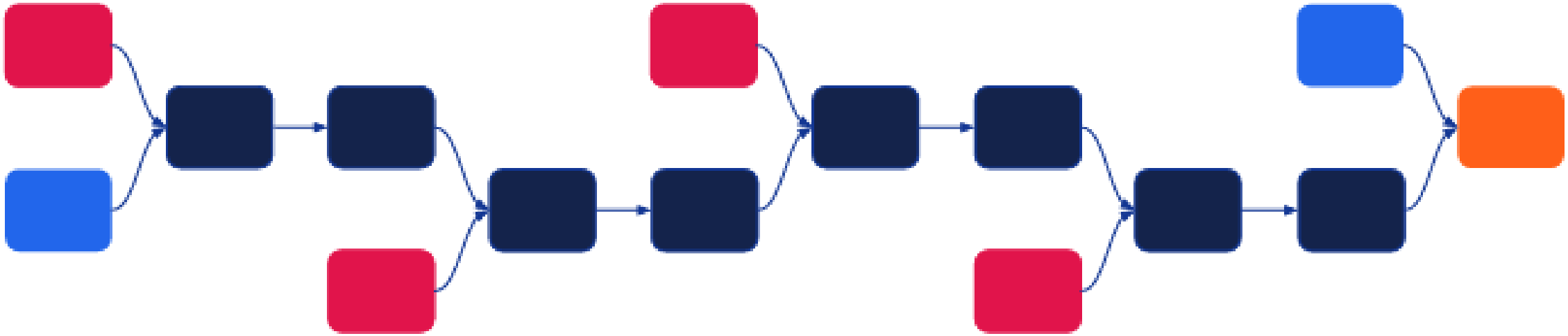
$$f_{\text{relu}}(\mathbf{x}) = \max(0, \mathbf{x})$$

$$f_{\text{sp}}(\mathbf{x}) = \log(1 + e^{\mathbf{x}})$$

Un ejemplo


Lines/corners detection
Shapes detection
Object detection
Class detection
Linear
ReLU
Linear
ReLU
Linear
ReLU
Linear
Softmax
Loss
Data
Target

Grafos computacionales



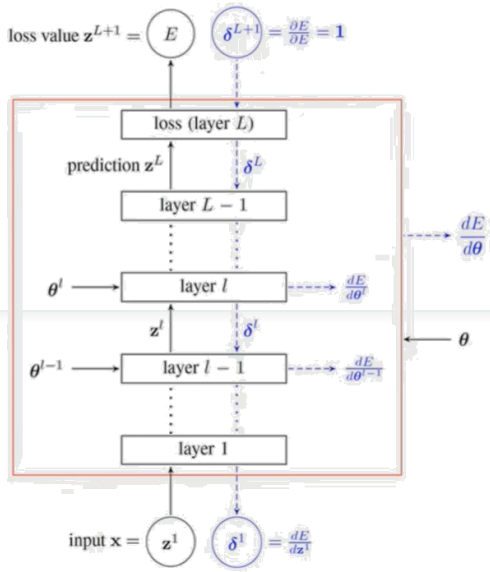
Aprendizaje

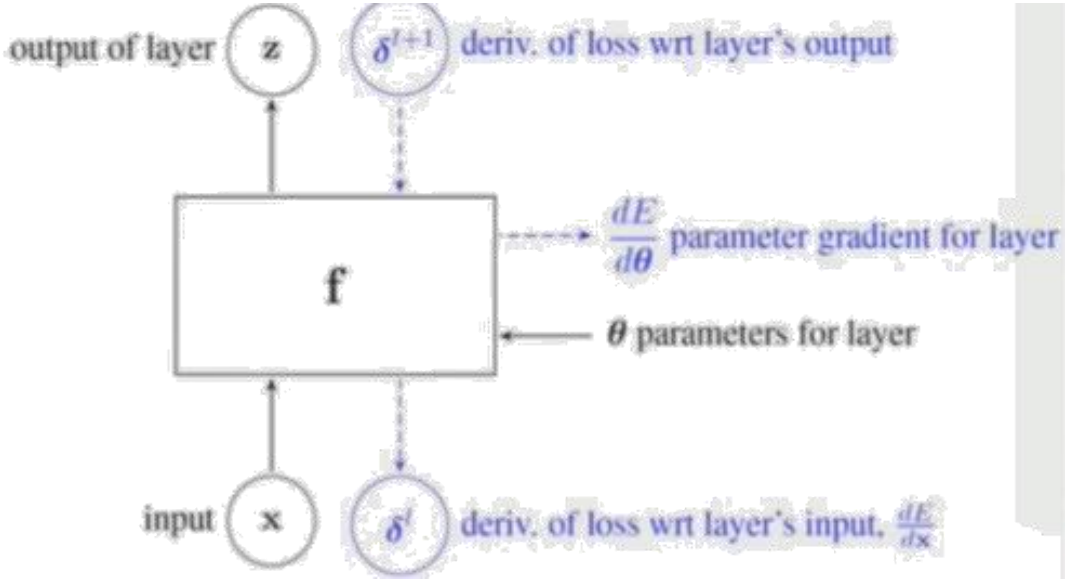
$f(\mathbf{x})$ Forward pass

$\mathbf{J}_{\mathbf{x}} f(\mathbf{x})$ Backward pass

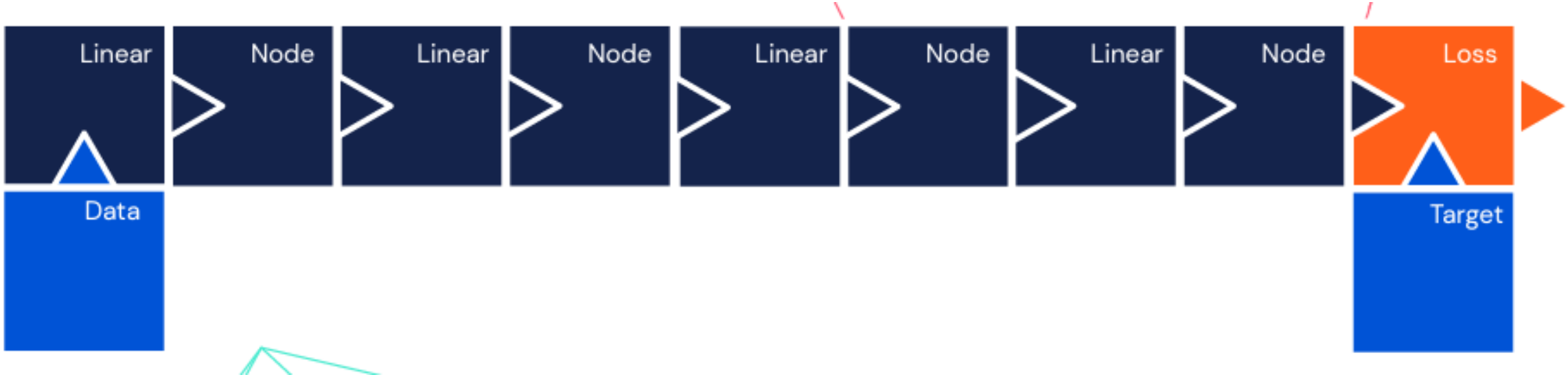
Modelo de composición

- Se necesitan 3 funciones
  - Forward
  - Backward
  - Derivada
- ¿Como componer las capas?
  - Secuencial
  - Recursivas
  - Redes

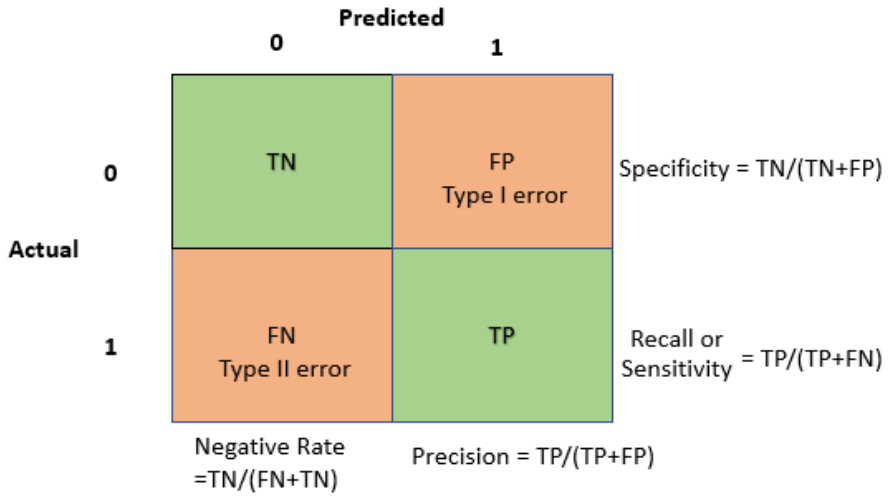




Redes neuronales profundas



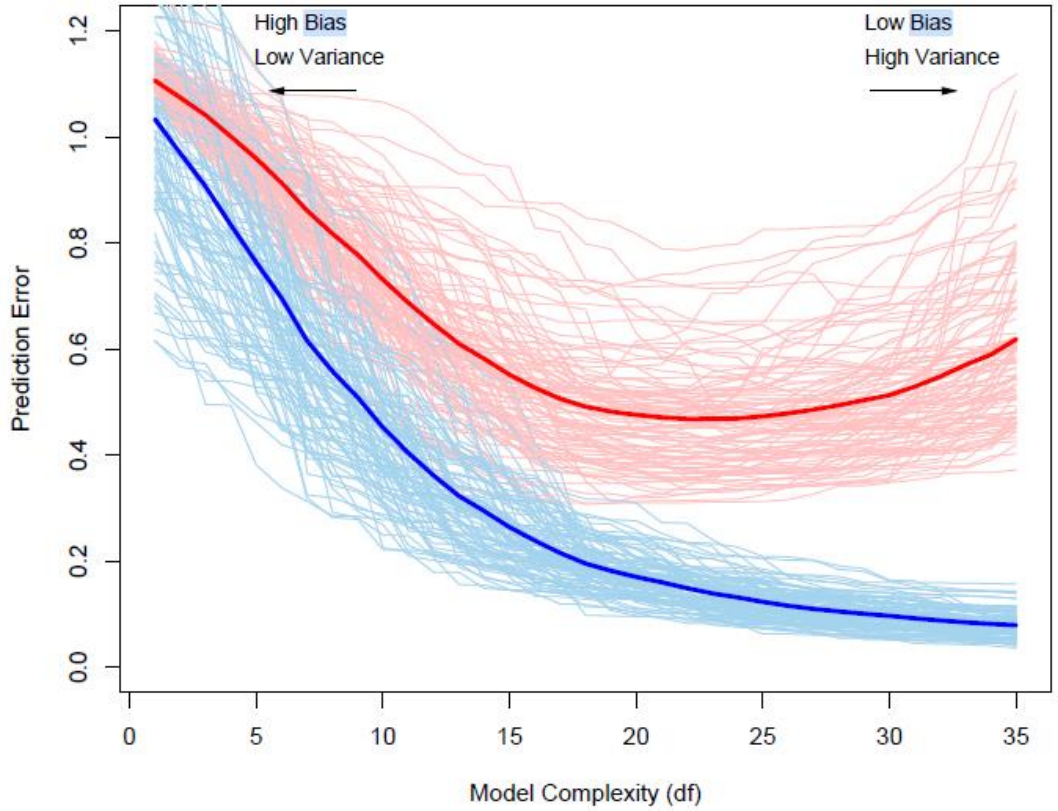
Indicadores de que tan bueno es nuestro clasificador



$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

$$\text{F1 - Score} = \frac{2*\text{Recall}*\text{Precision}}{\text{Recall} + \text{Precision}}$$

El sesgo y la varianza



Para la otra vez…

- Aprendizaje no supervisado