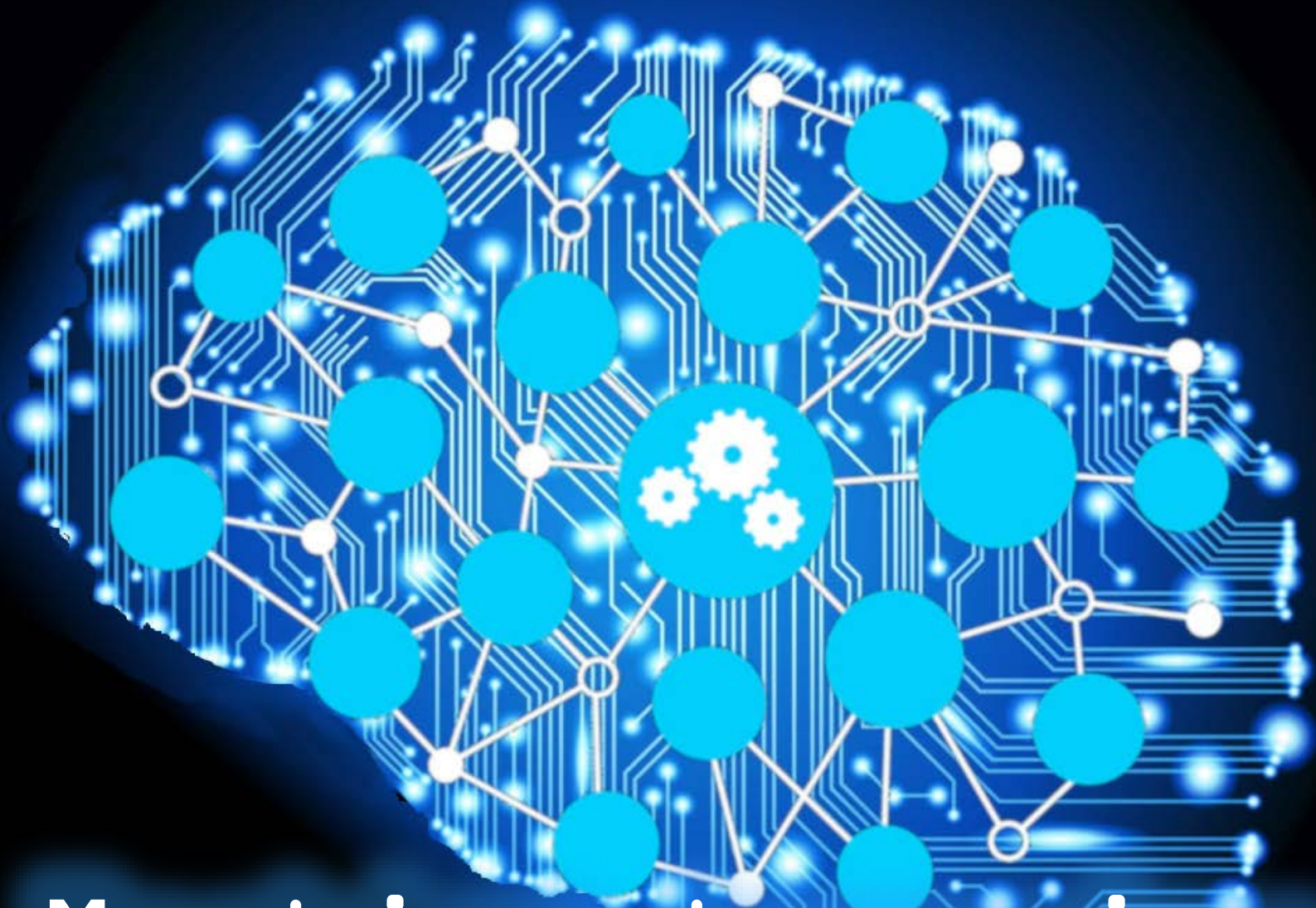




Machine Learning

I. Introducción al Machine Learning y a la Regresión

9 9 9 9

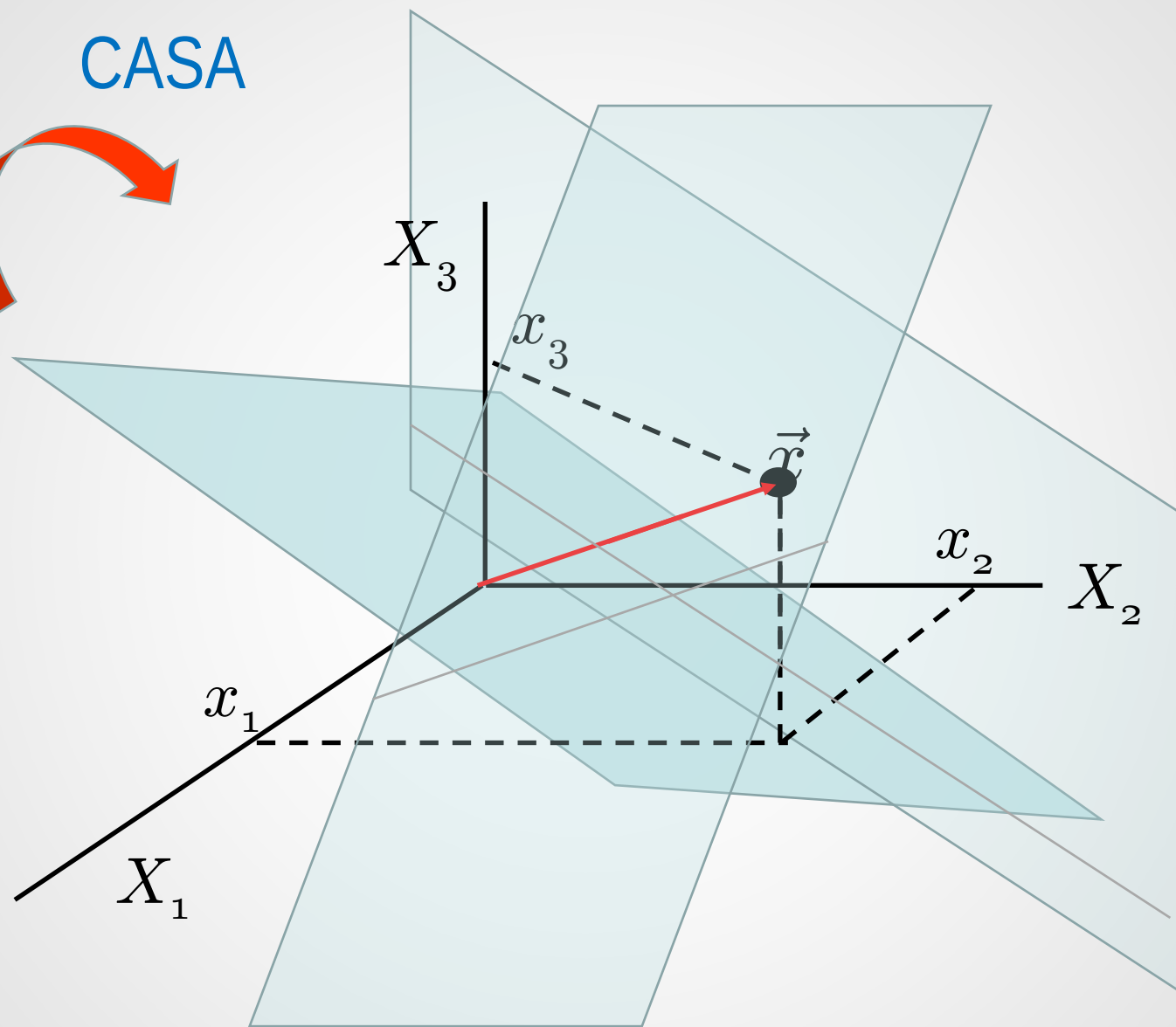


CASA



CASA

COCHE





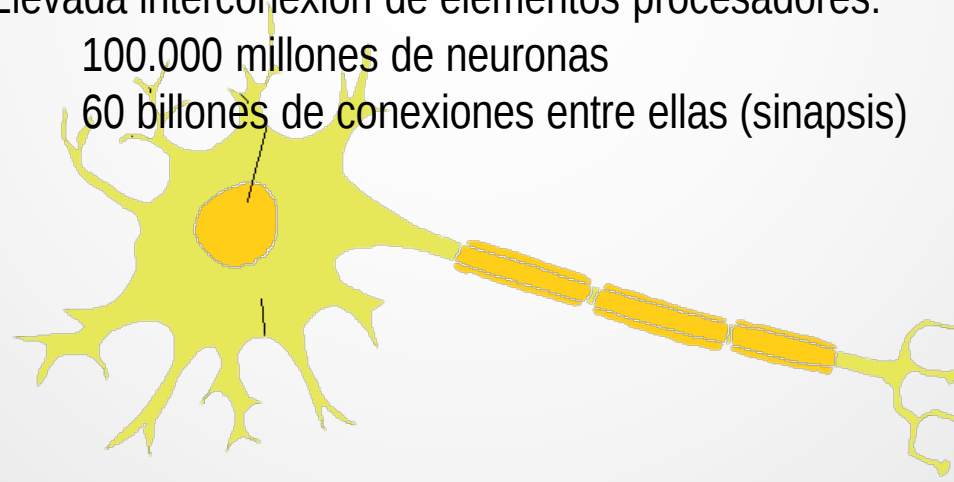
¿Cómo hace el cerebro para manejar tantas variables?

¿Cómo podemos hacerlo nosotros automáticamente?

Elevada interconexión de elementos procesadores:

100.000 millones de neuronas

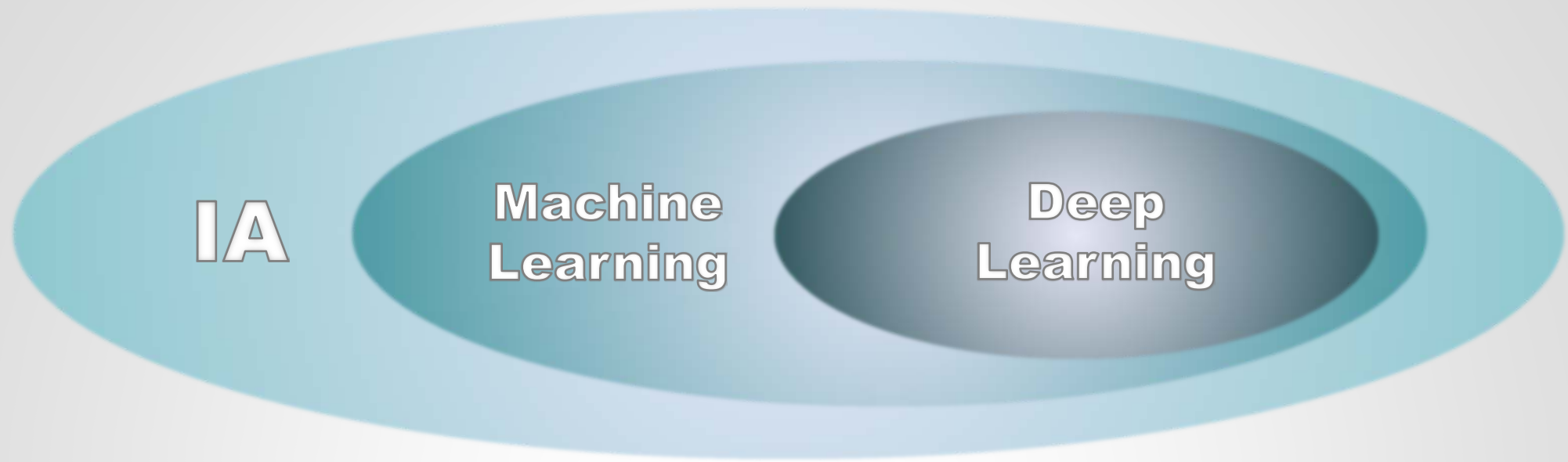
60 billones de conexiones entre ellas (sinapsis)

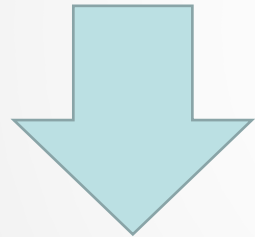
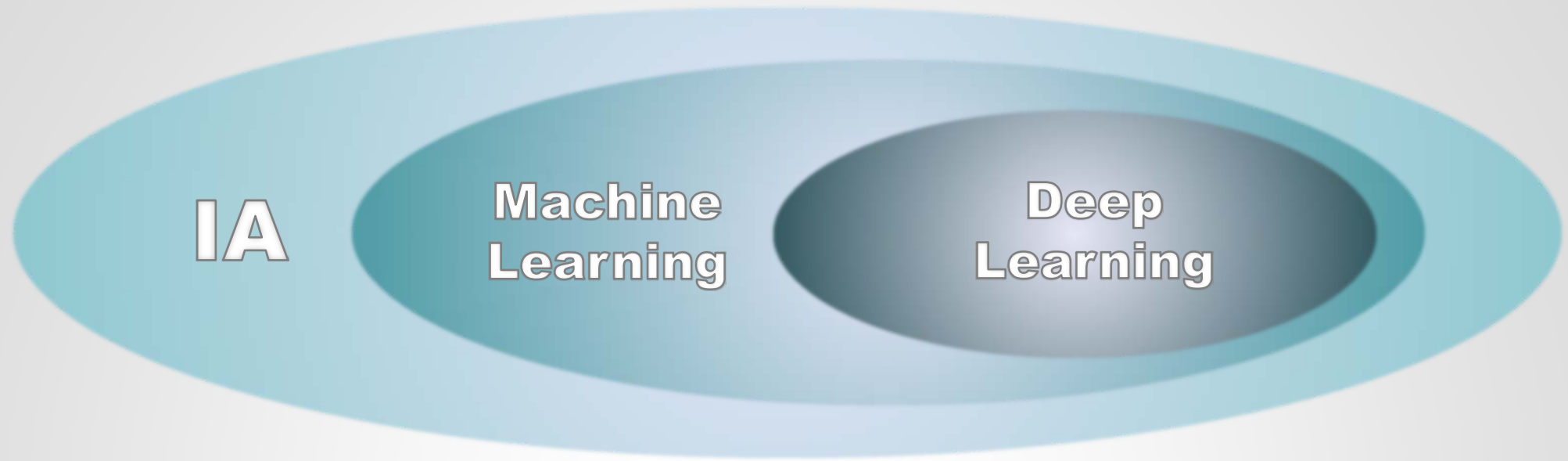




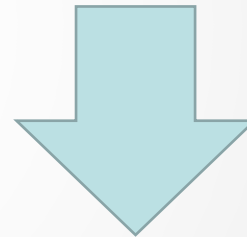
**¿Cómo podemos enseñar a los niños a aprender?
ENSEÑÁNDOLOS A APRENDER**

IMITÁNDOLO

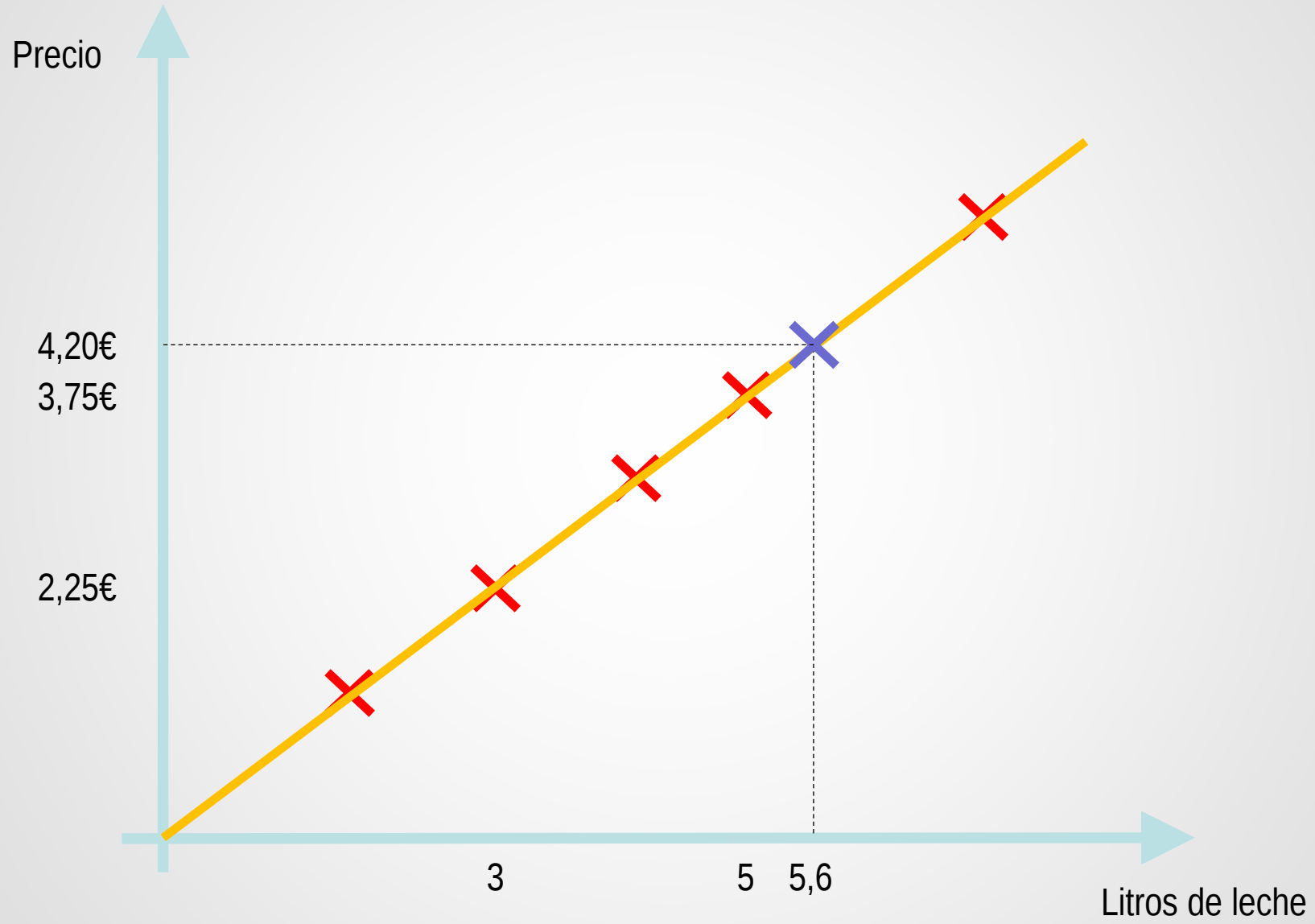


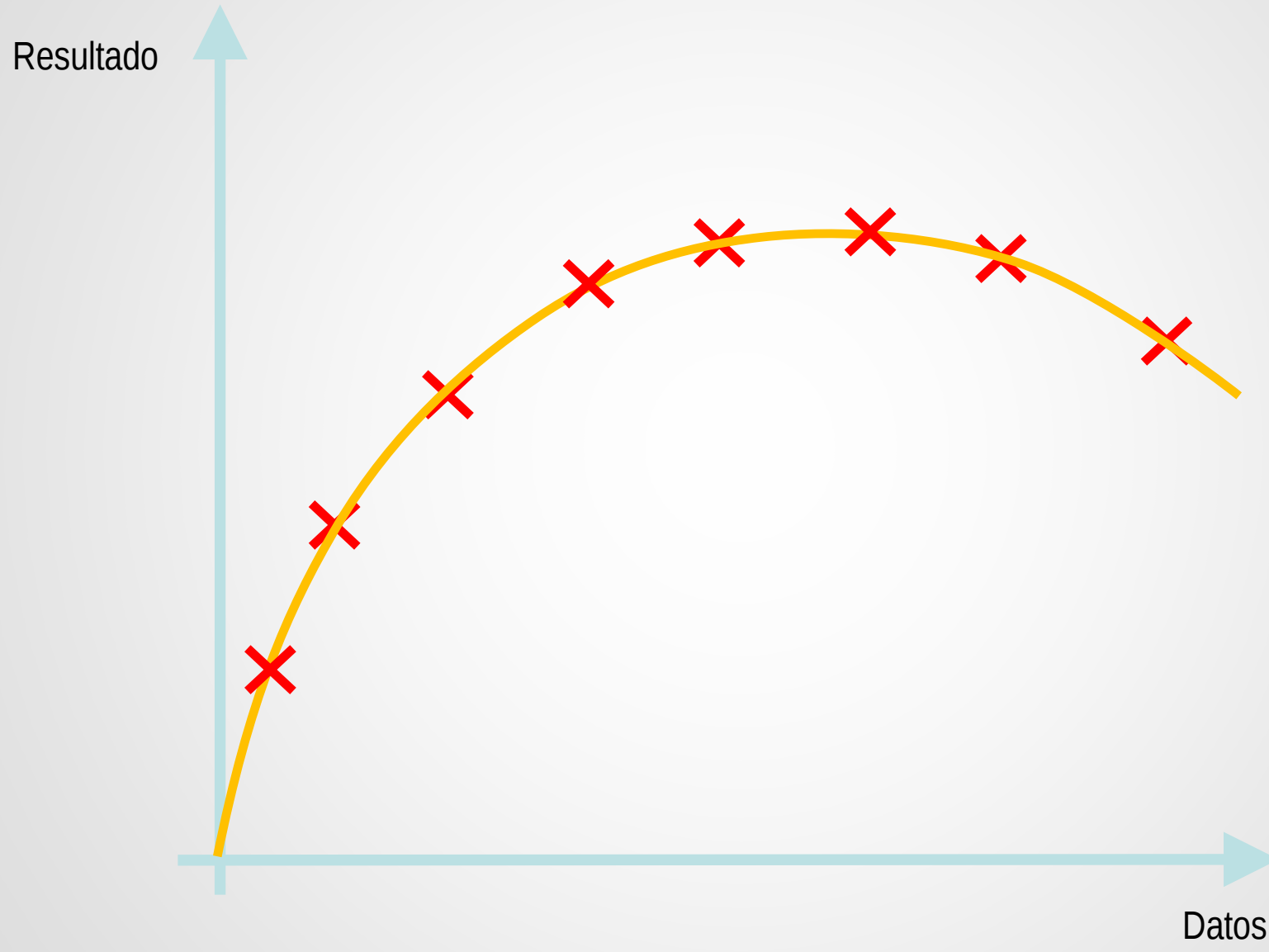


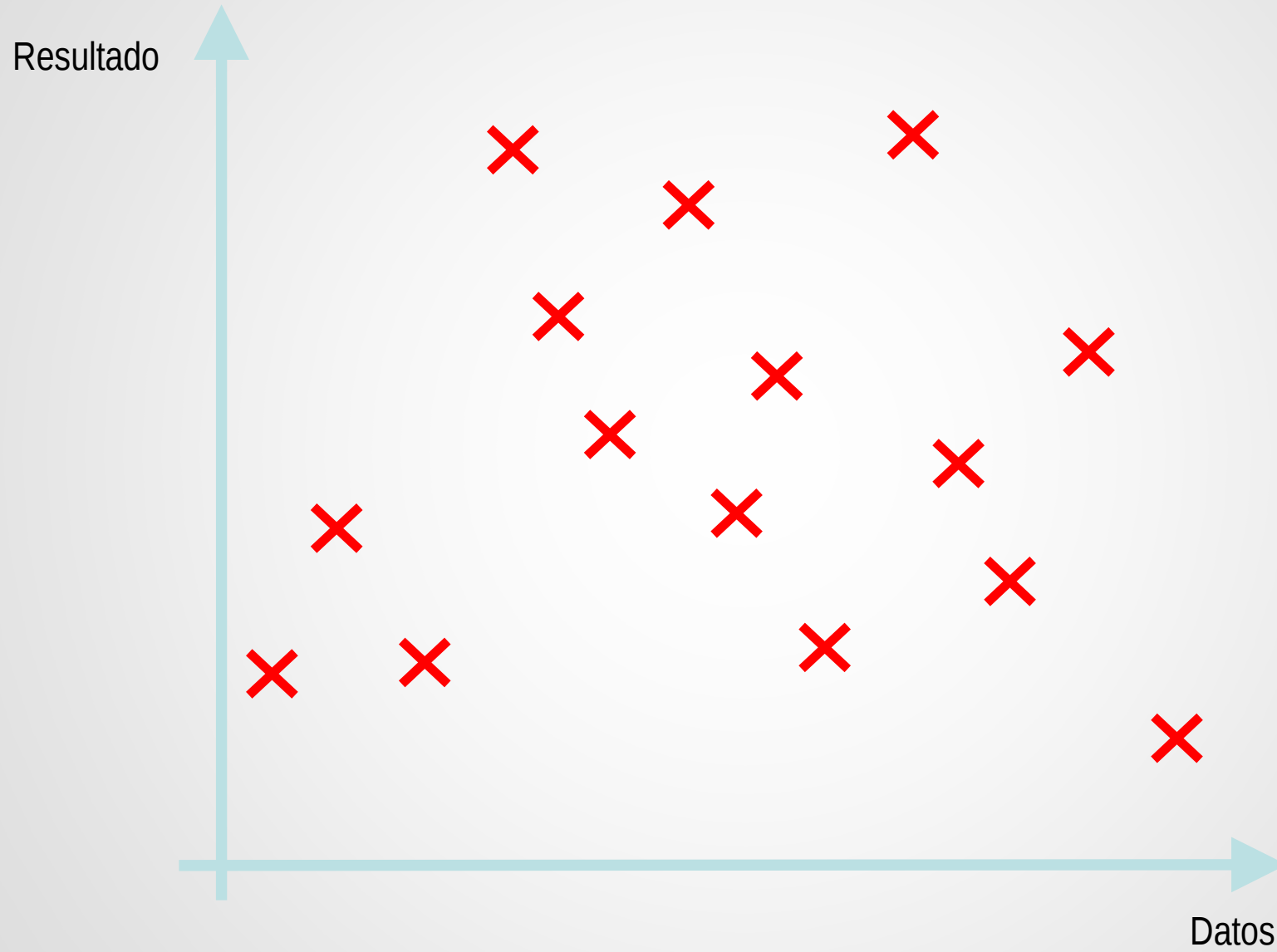
Regresión

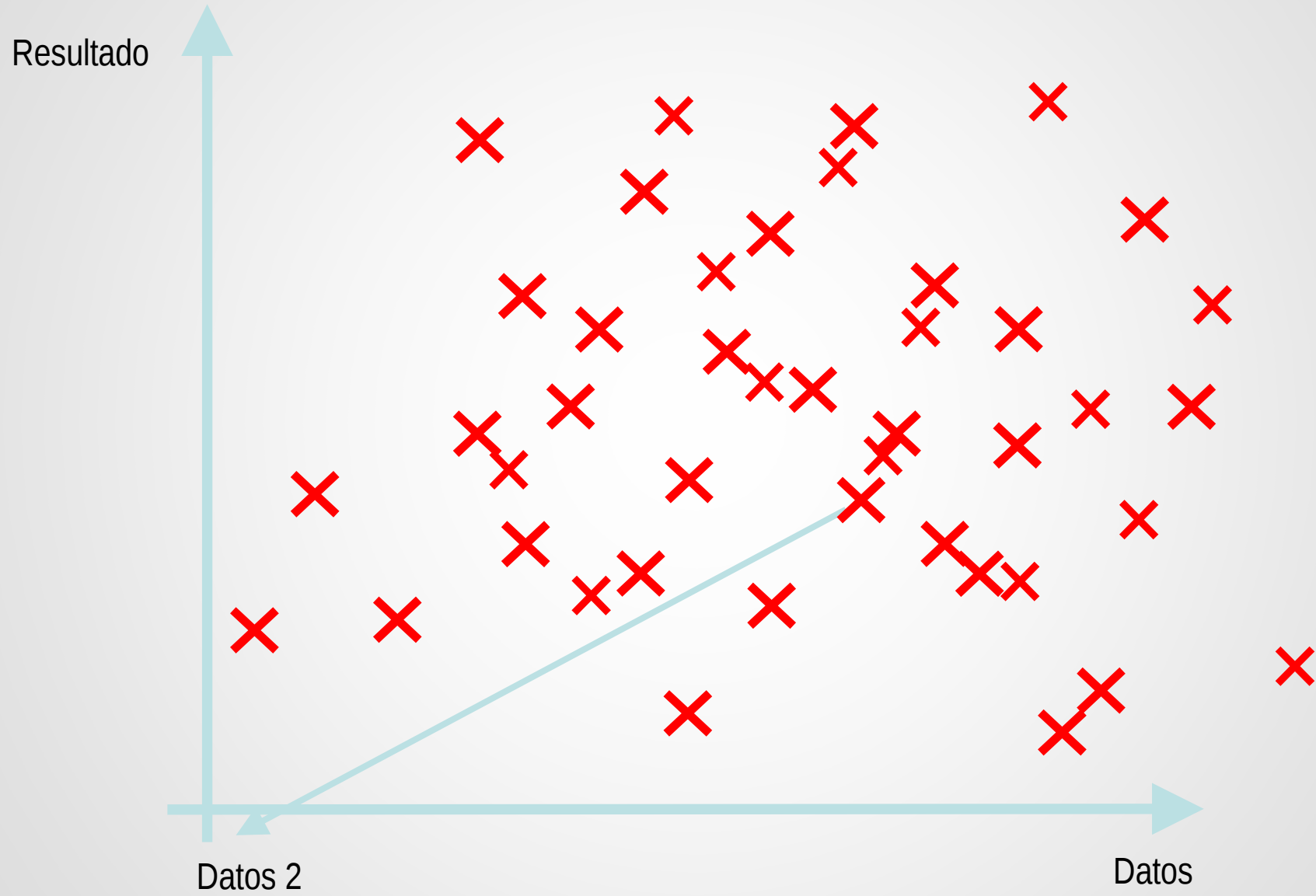


Clasificación









Regresión lineal

Caracterización del problema:

Punto de partida $\mathbf{x}$ (o datos de partida, señal de partida...).

Se busca un conjunto de variables $\mathbf{w}_{opt}$ que, aplicados a $\mathbf{x}$ por medio de una función f conocida, conduzcan a un valor deseado y conocido d .

$$d = f(\mathbf{x}, \mathbf{w}_{opt})$$

Pretendemos relacionar $\mathbf{x}$ y d ajustando los parámetros de una función conocida.

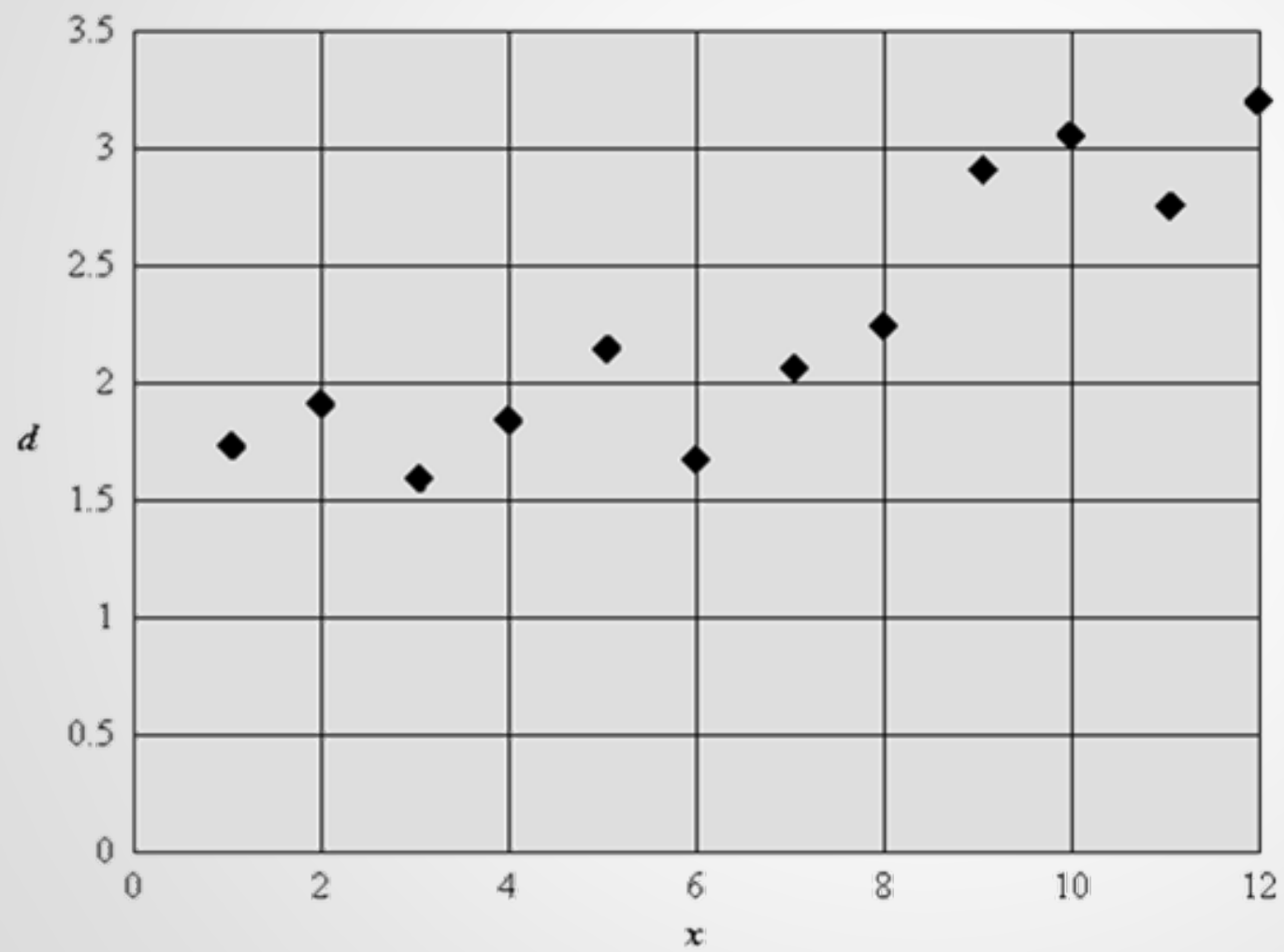
En **regresión lineal** se supone que la función es una recta (suponiendo un caso 1-D)

$$d \approx wx + b$$

Siempre se parte de un conjunto de datos conocidos que se emplean para el ajuste de la función

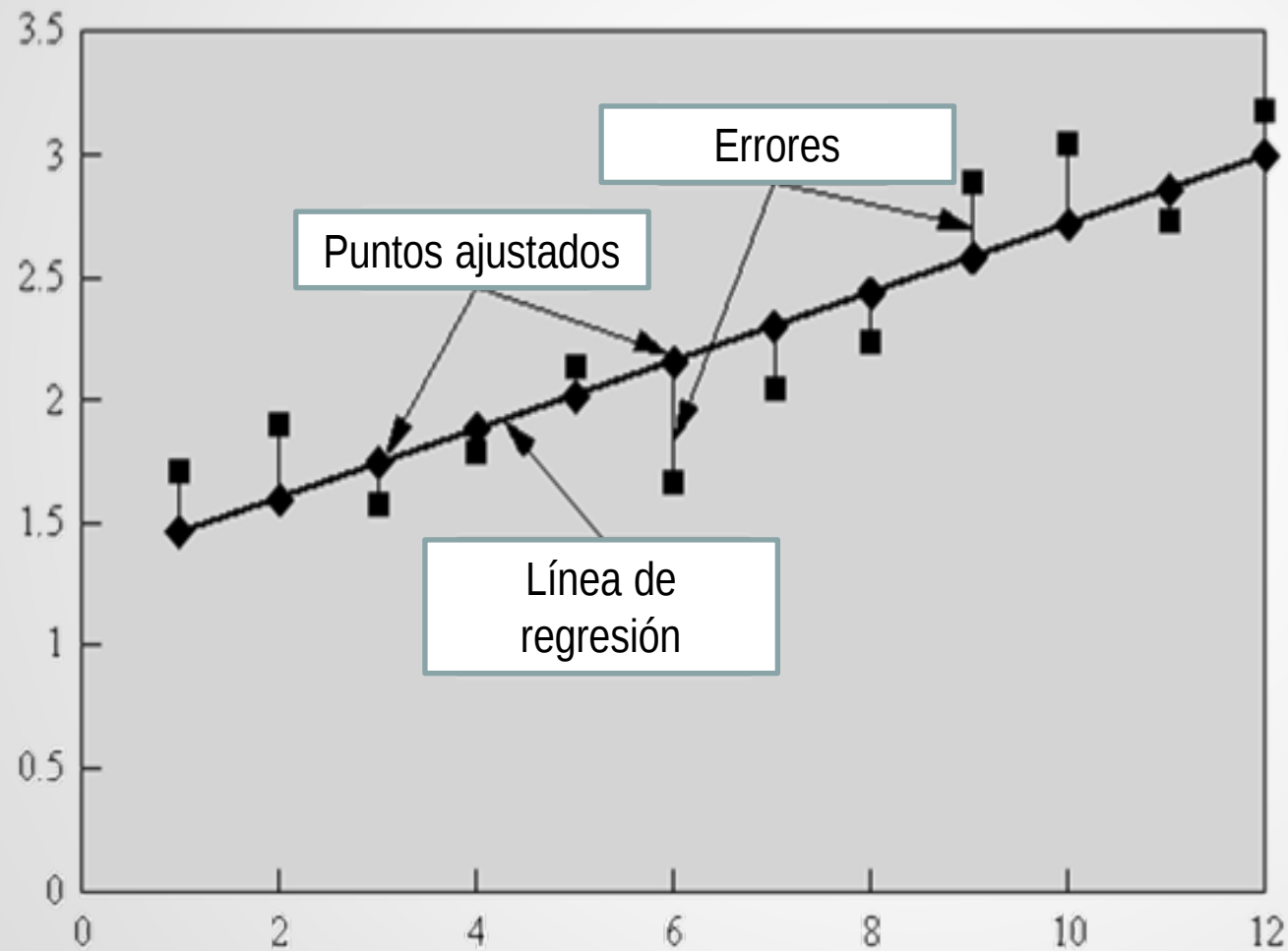
Datos de entrenamiento

x	d
1	1.72
2	1.90
3	1.57
4	1.83
5	2.13
6	1.66
7	2.05
8	2.23
9	2.89
10	3.04
11	2.72
12	3.18



x	d
1	1.72
2	1.90
3	1.57
4	1.83
5	2.13
6	1.66
7	2.05
8	2.23
9	2.89
10	3.04
11	2.72
12	3.18

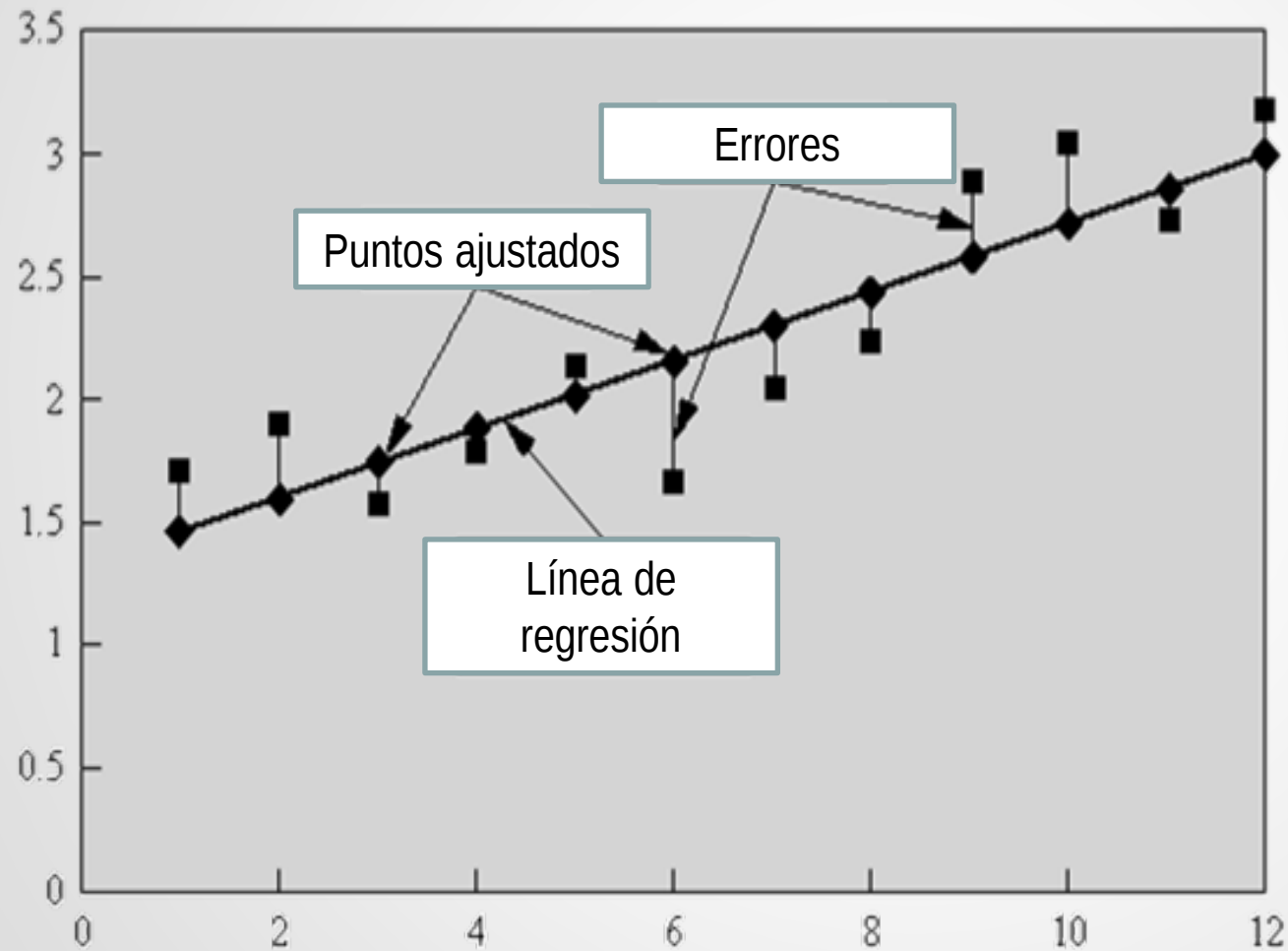
$$d_i = wx_i + b \cdot 1 + \epsilon_i$$



x	d
1	1.72
2	1.90
3	1.57
4	1.83
5	2.13
6	1.66
7	2.05
8	2.23
9	2.89
10	3.04
11	2.72
12	3.18

$$d_i = wx_i + b \cdot 1 + \epsilon_i$$

$$d_i = wx_i + bx_{2i} + \epsilon_i$$



x	d	x_2
1	1.72	1
2	1.90	1
3	1.57	1
4	1.83	1
5	2.13	1
6	1.66	1
7	2.05	1
8	2.23	1
9	2.89	1
10	3.04	1
11	2.72	1
12	3.18	1

$$d_i = w_1x_{i1} + w_2x_{i2} + w_3x_{i3} + \dots + w_Dx_{iD}$$

1D Recta

2D Plano

>2D Hiperplano

Objetivo: encontrar los coeficientes (uno es b)

Para poder realizar el ajuste es necesario definir claramente cuál es el criterio de error empleado. El más extendido es el error cuadrático medio (MSE):

$$\text{minimize } \frac{1}{N} \sum_{i=1}^N \epsilon_i^2 \qquad \epsilon_i = d_i - \sum_{j=1}^D w_j x_{ij}$$

Si pretendemos el mínimo error, debemos encontrar el mínimo de la función de coste respecto de cada variable, derivando e igualando a cero (D derivadas).

Se llega a:

$$\sum_{i=1}^N x_{ik} d_i = \sum_{j=1}^D w_j \sum_{i=1}^N x_{ij} x_{ik}, \qquad j = 1 \dots D$$

Son la versión determinista de las **ECUACIONES NORMALES**

N es el número de datos de entrenamiento

D es la dimensión del problema (-1, porque se incluye el término de continua b como coeficiente)

Si definimos:

- Autocorrelación de las muestras de entrada de cada par de “dimensiones”, k y j :

$$R_{kj} = \frac{1}{N} \sum_{i=1}^N x_{ik} x_{ij}$$

- Correlación cruzada de la entrada de índice j y la respuesta deseada:

$$p_j = \frac{1}{N} \sum_{i=1}^N x_{ij} d_i$$

Podemos entonces formar una matriz y un vector con esos valores y expresar las ecuaciones normales matricialmente:

$$\mathbf{p} = \mathbf{R} \mathbf{w}_{opt}$$

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{p}$$

Esta es la forma básica de REGRESIÓN.

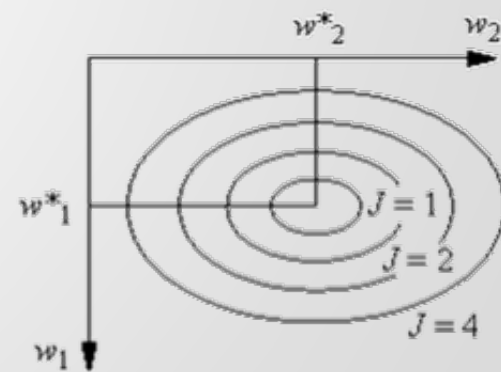
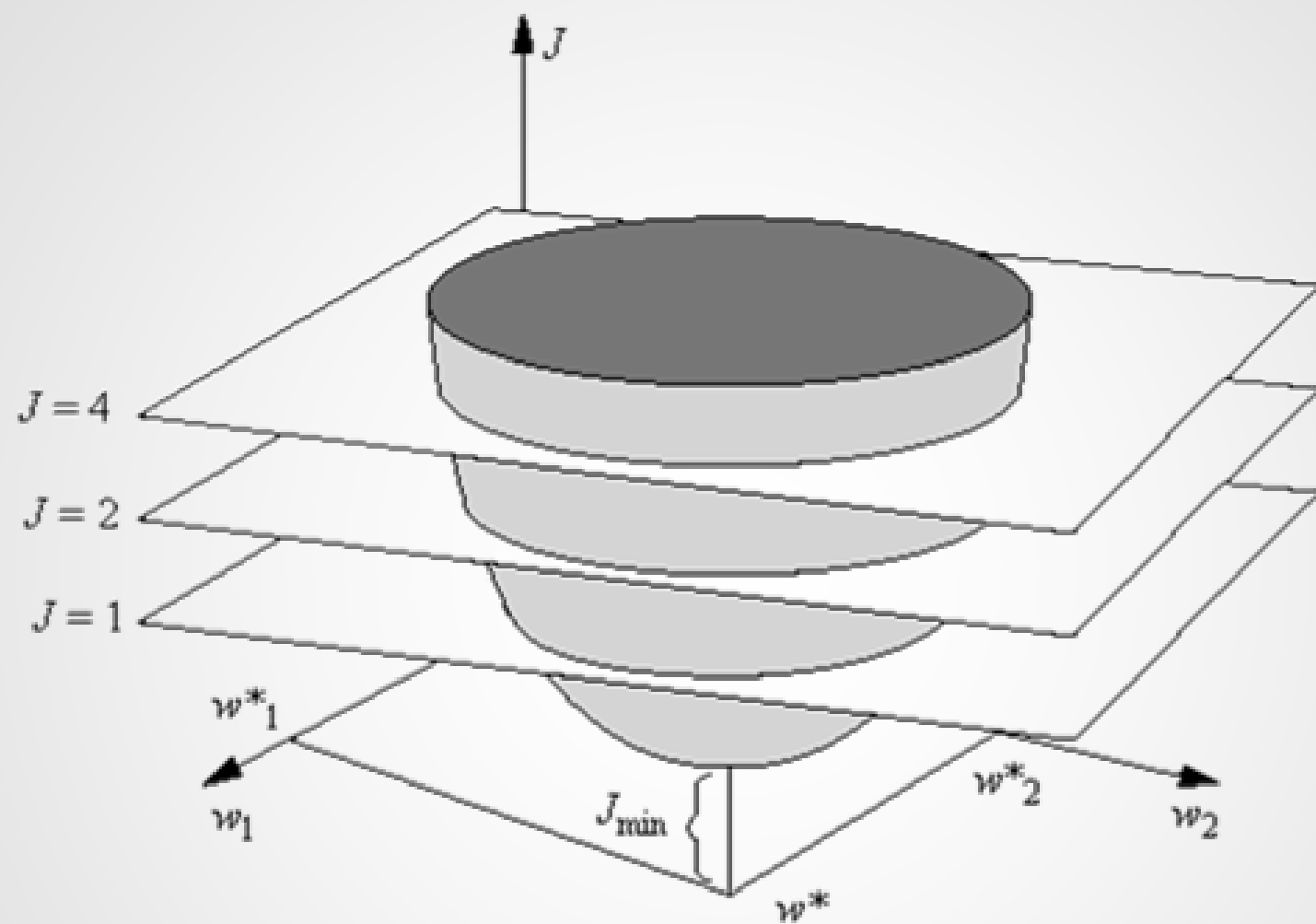
En regresión no lineal cambia la función que relaciona entrada y salida: hay que elegir una y ajustar sus parámetros siguiendo exactamente el mismo procedimiento.

La regresión resulta ser la versión determinista del FILTRADO ÓPTIMO.

La función de regresión se ha ajustado a partir de datos observados: ENTRENAMIENTO o APRENDIZAJE del modelo.

Al igual que el filtrado, puede hacerse de manera adaptativa de forma que siga aprendiendo dato a dato. Se basa en el estudio de la SUPERFICIE DE ERROR, que en este caso es

$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N \epsilon_i^2$$



Se pueden plantear, por tanto, un algoritmo de máxima pendiente...

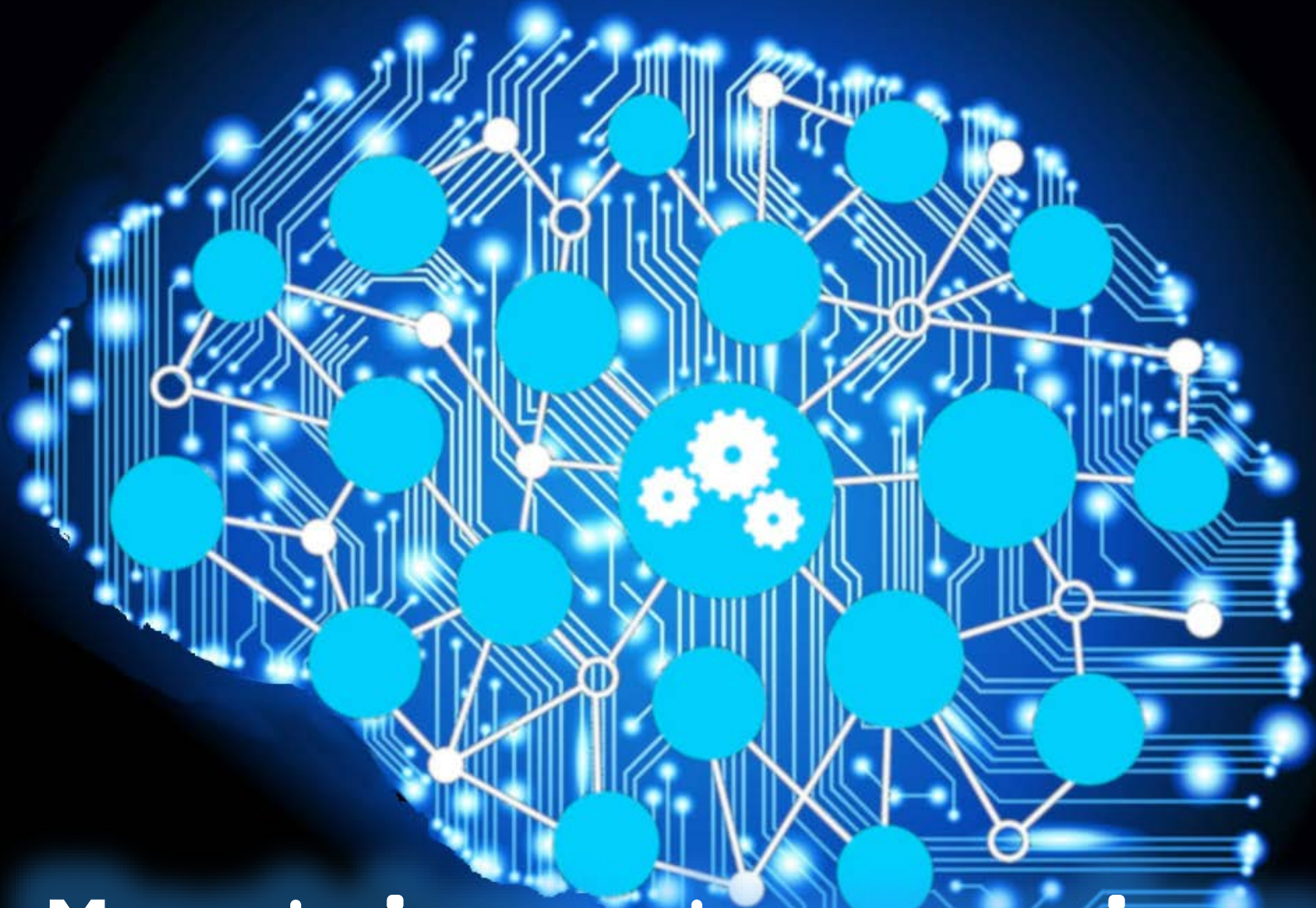
$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \nabla \mathbf{J}(k)$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu [\mathbf{R}\mathbf{w}(k) - \mathbf{p}]$$

... y un LMS

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \mu \cdot \epsilon_i \cdot \mathbf{x}_i$$

Una implementación mixta podría entrenar el modelo con los datos disponibles, y posteriormente hacerlo adaptativo para que siga aprendiendo con nuevos datos.



Machine Learning