

# Recap Aprendizaje de Máquinas

Visión por Computador II

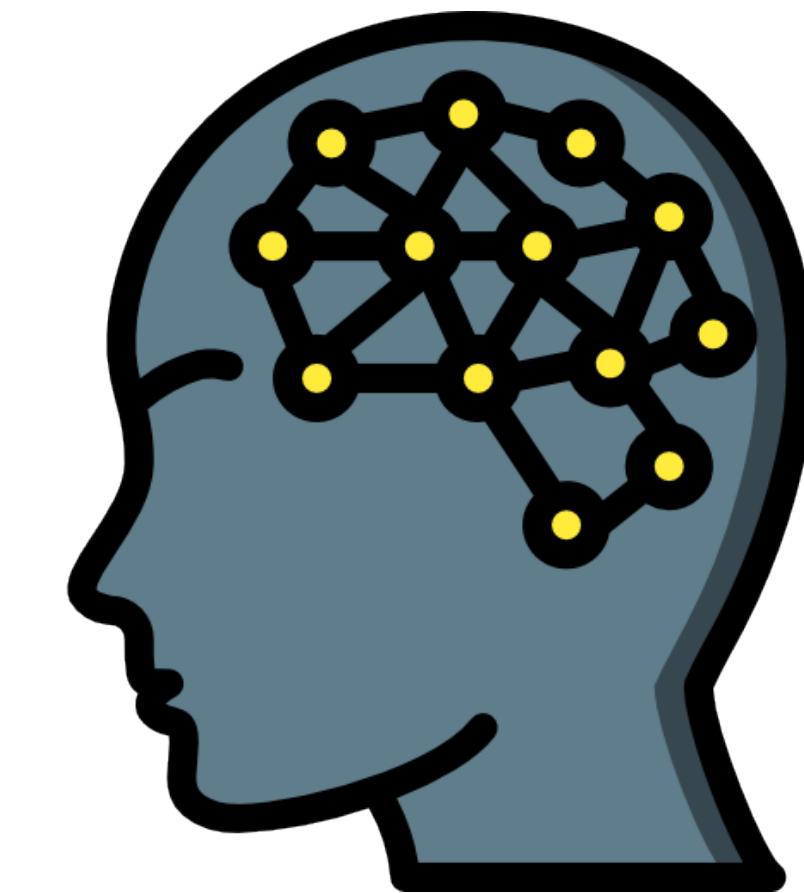
# Contenido

- Intro AI
- Regresión lineal
- Función de costo - MSE
- Gradient descent
- Feature scaling

# Inteligencia Artificial

## ¿Qué es IA?

“Busca que los computadores **hagan cosas que las mentes pueden hacer** algunas son descritas como *inteligencia (Razonar)* otras no (*Visión*) pero involucran habilidades psicológicas” - Margaret A. Boden



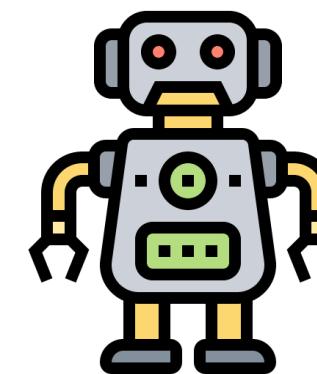
**IA usa diferentes técnicas para enfrentarse a tareas diferentes**

# Inteligencia Artificial

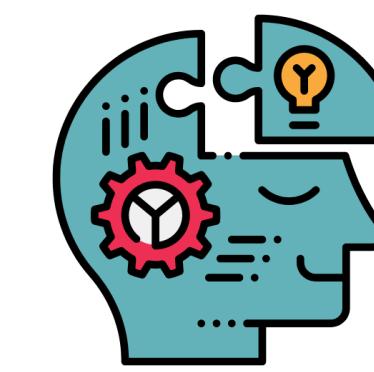
## ¿Qué es IA?

**Actuar de forma humana: La aproximación de la prueba de Turing**

**Automatización de actividades que normalmente atribuimos al pensamiento y racionalidad humana**



**Robotics**



**Machine Learning**



**Computer Vision**



**Natural Language Processing**



**Expert Systems**

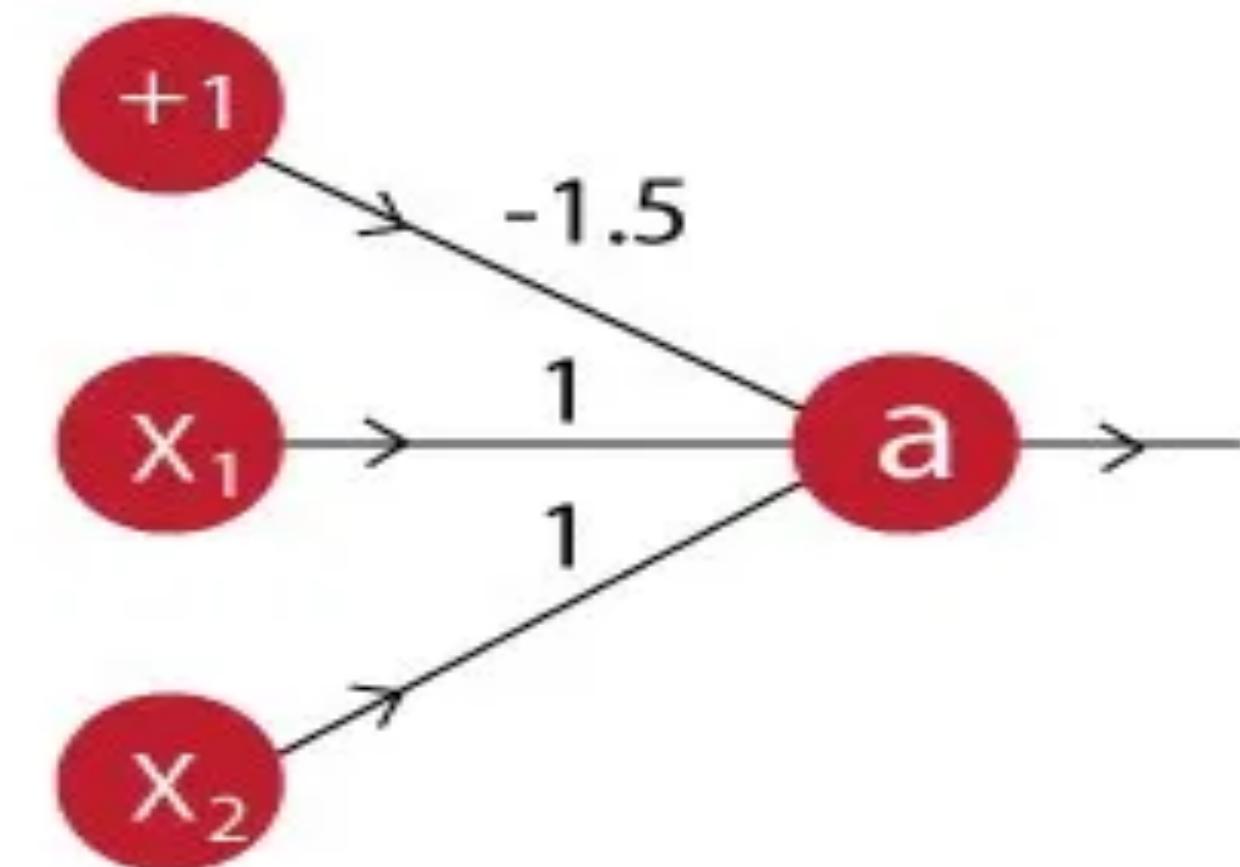
# Inteligencia Artificial

## Historia

### 1943 Gestación de la inteligencia artificial

- Warren McCulloch y Walter Pitts
  - Modelo de una neurona artificial
  - Calcular cualquier función y conexiones lógicas (and, or...),
- Marvin Minsky and Dean Edmonds
  - Construyen la primera Red Neuronal.
  - Usaron 3000 válvulas para simular una red de 40 Neuronas

Diagram 2: AND



# Inteligencia Artificial

## Historia

**1952** Entusiasmo y grandes expectativas: *Gran éxito en casos muy simplificados*

The **Turing test** is developed by Alan Turing to test whether a machine is capable of human intelligent behaviour or not.

**1969** Dosis de realidad

- Falta de conocimiento del dominio
- Knowledge-based systems

**1980** “AI” se convierte una industria

The **First Robot** is introduced on an assembly line at General Motors.



John McCarthy, an American computer scientist, coined the term '**Artificial Intelligence**'.

**IBM's Deep Blue** - a chess playing computer - beats then chess world champion, Garry Kasparov.

# Inteligencia Artificial

## Historia

- **1986** Regreso de Redes Neuronales
- **Mitad de los 80s:** Reinventado algoritmo de “**backpropagation**”
- Investigación de redes neuronales:
  - Arquitecturas, algoritmos y eficiencia
  - Modelar propiedades actuales de las neuronas



**Qué es Algoritmo de “backpropagation”**

# Inteligencia Artificial

## Historia

### 2001 - Disponibilidad de grandes datasets

- Tiene mas sentido preocuparse por los datos que por el algoritmo a aplicar
- Un algoritmo mediocre con un dataset de 100 millones supera a uno mejor con 1 millón



# Inteligencia Artificial

## Historia



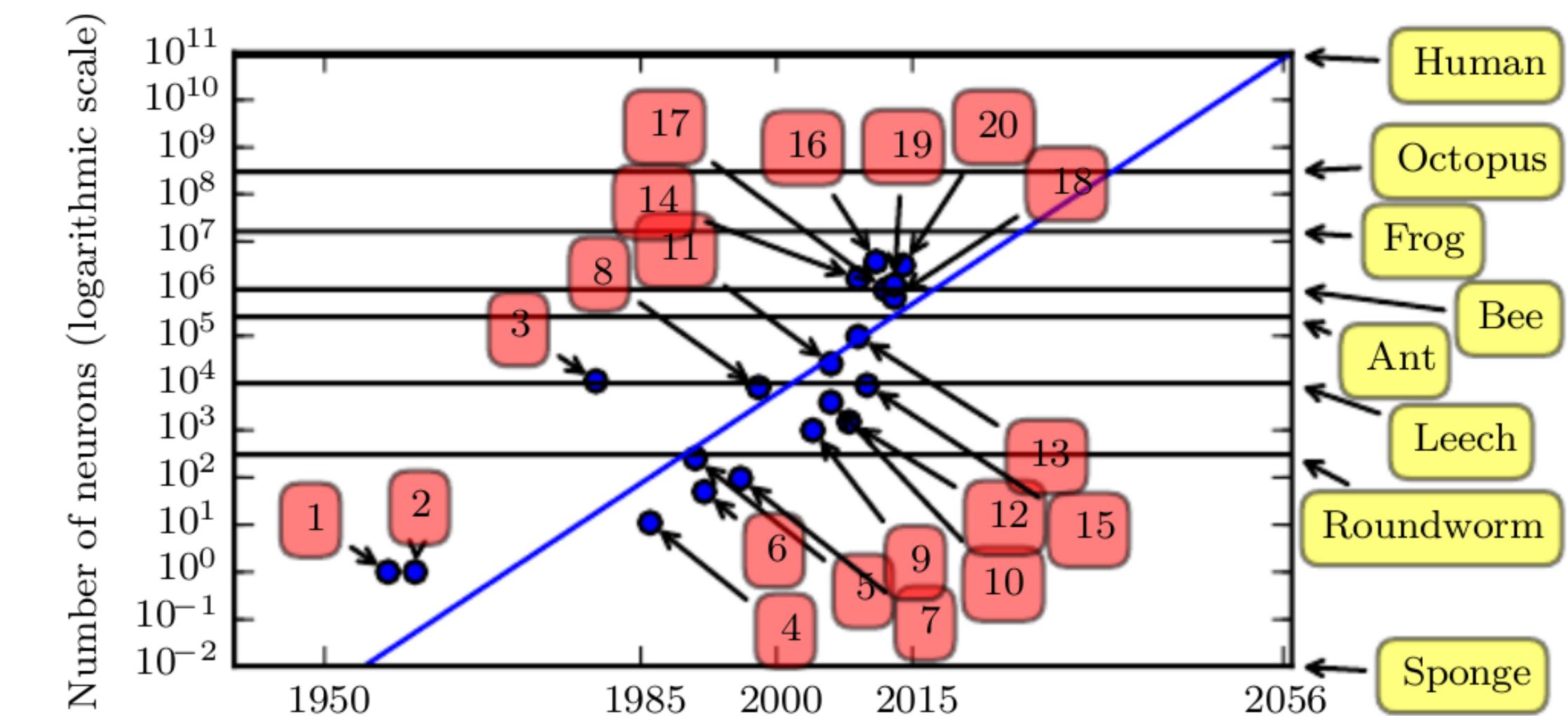
**“No es quién tenga el mejor algoritmo quien gana, es quien tenga más datos” - Banko y Brill, 2005**

# Inteligencia Artificial

## Hoy en día

IA Explotó recientemente por:

- Poder de cómputo
- Datasets grandes y sin costo
- Avances en Machine Learning



Tecnologías Claves que soportan IA

**Big Data & Machine Learning**

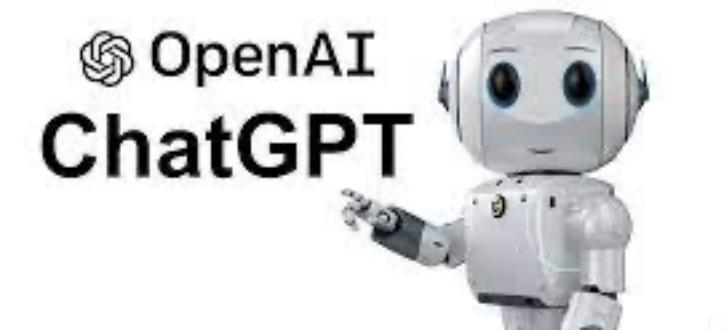
2016 - Deep Learning by  
Ian Goodfellow

# Inteligencia Artificial

Hoy en día

## Usos más comunes (Vida diaria)

- Interpretación visual
- Procesamiento de lenguaje
- Sistemas de recomendación
- Detección de anomalías

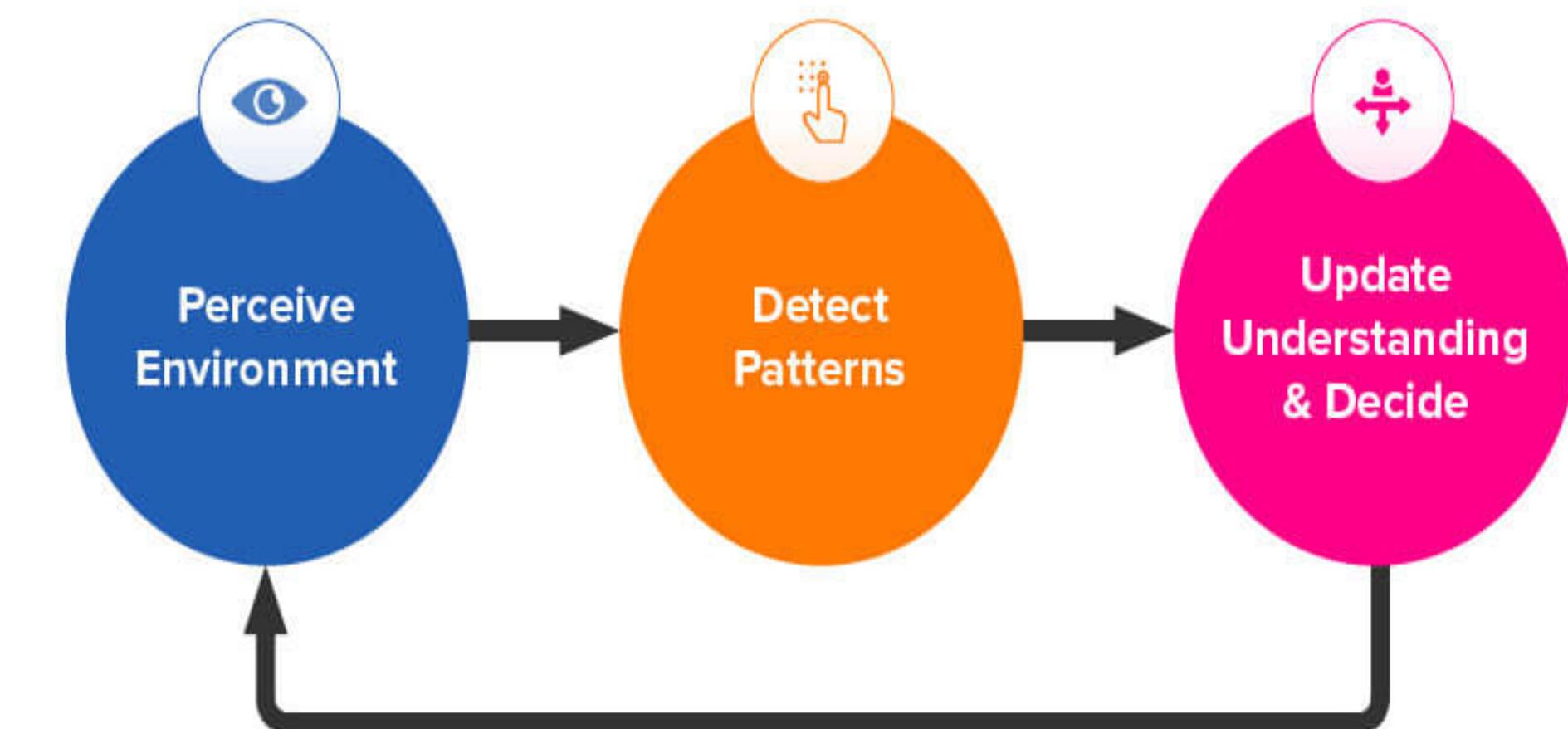


# Inteligencia Artificial

Hoy en día

## Como funcionan?

- Gran cantidad de datos
- modelo ajustado con experiencias pasadas
- Procesar nueva observación
- Usar los patrones encontrados en los datos para tomar una decisión (predicción)



[su.org/resources/exponential-guides/the-exponential-guide-to-artificial-intelligence](http://su.org/resources/exponential-guides/the-exponential-guide-to-artificial-intelligence)

# Inteligencia Artificial

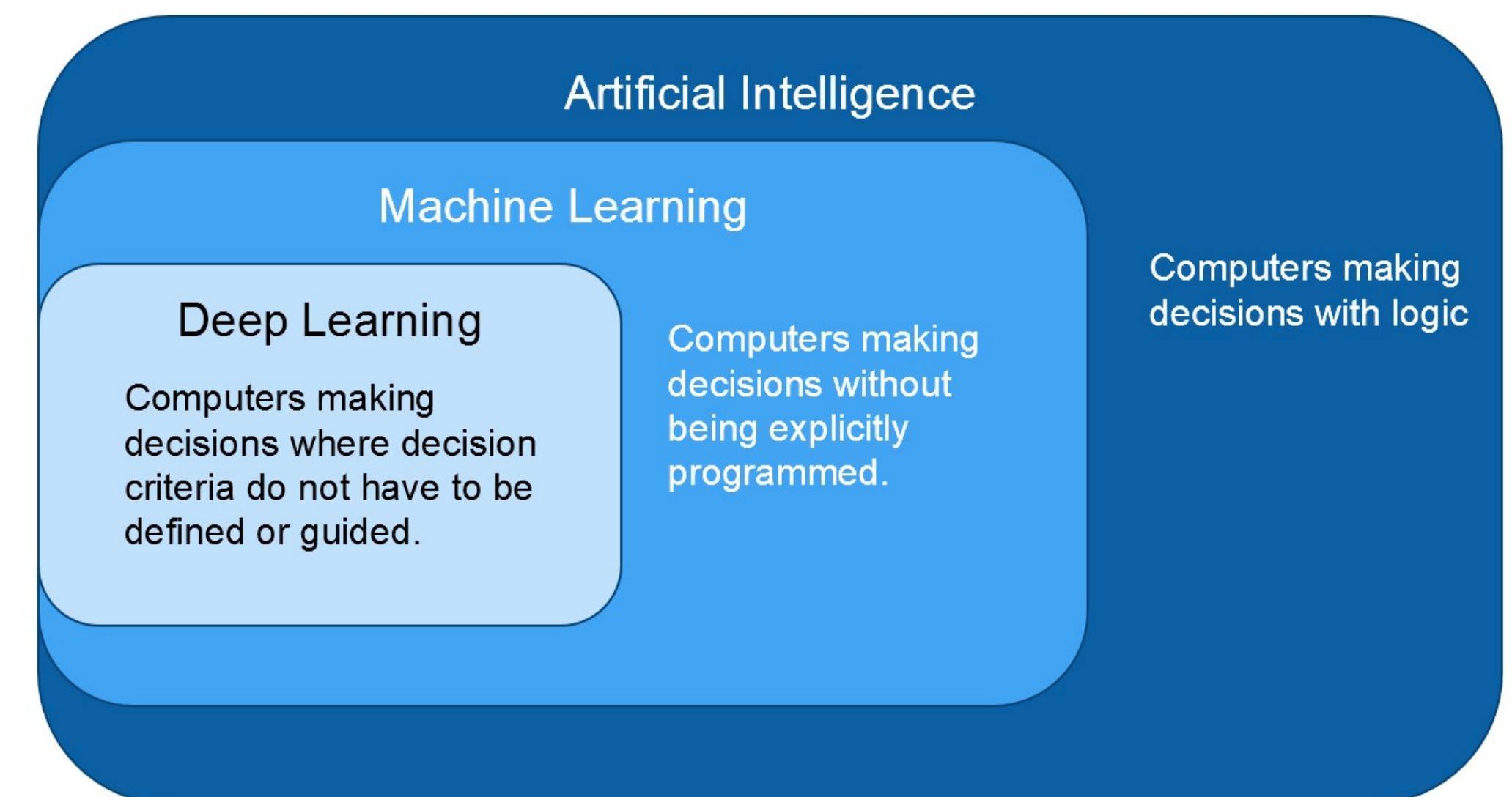
Hoy en día

## Machine Learning

- Método aprende de experiencia
- Realiza una tarea sin instrucciones explícitas

## Deep Learning

Representación automática de los datos

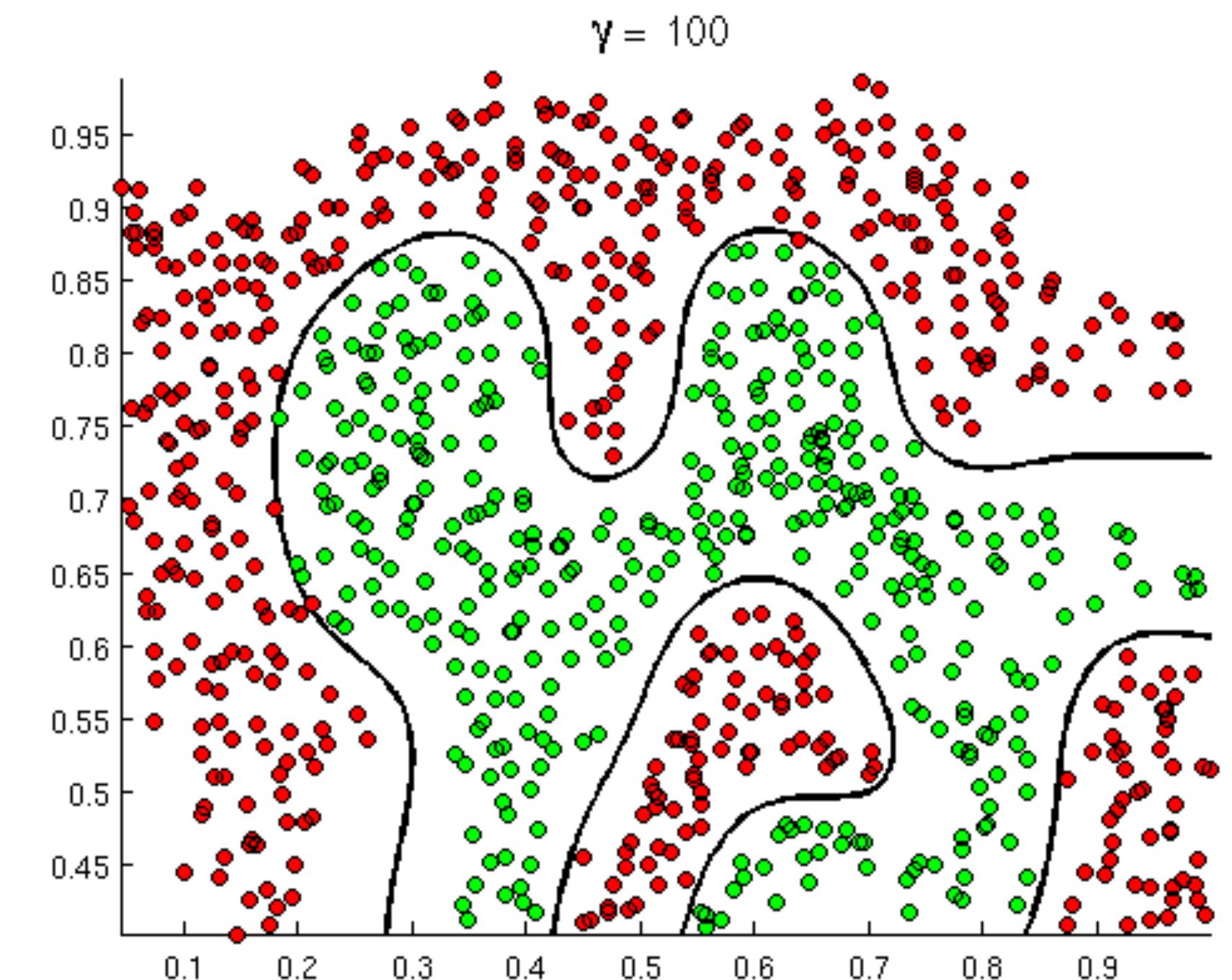


[ni.com/es-co/innovations/white-papers/18/  
deploying-deep-learning-models-to-ni-  
hardware](http://ni.com/es-co/innovations/white-papers/18-deploying-deep-learning-models-to-ni-hardware)

# Machine Learning

## ¿Qué es?

Es una aplicación de Inteligencia Artificial que da al sistema la **habilidad de aprender automáticamente** y mejorar de experiencias **sin ser explícitamente programada**



# Machine Learning

¿Cuándo usarlo?

Tareas que son **muy complejas**

- Tareas realizadas por animales o humanos
- Tareas fuera de la capacidad humana



**Adaptabilidad**

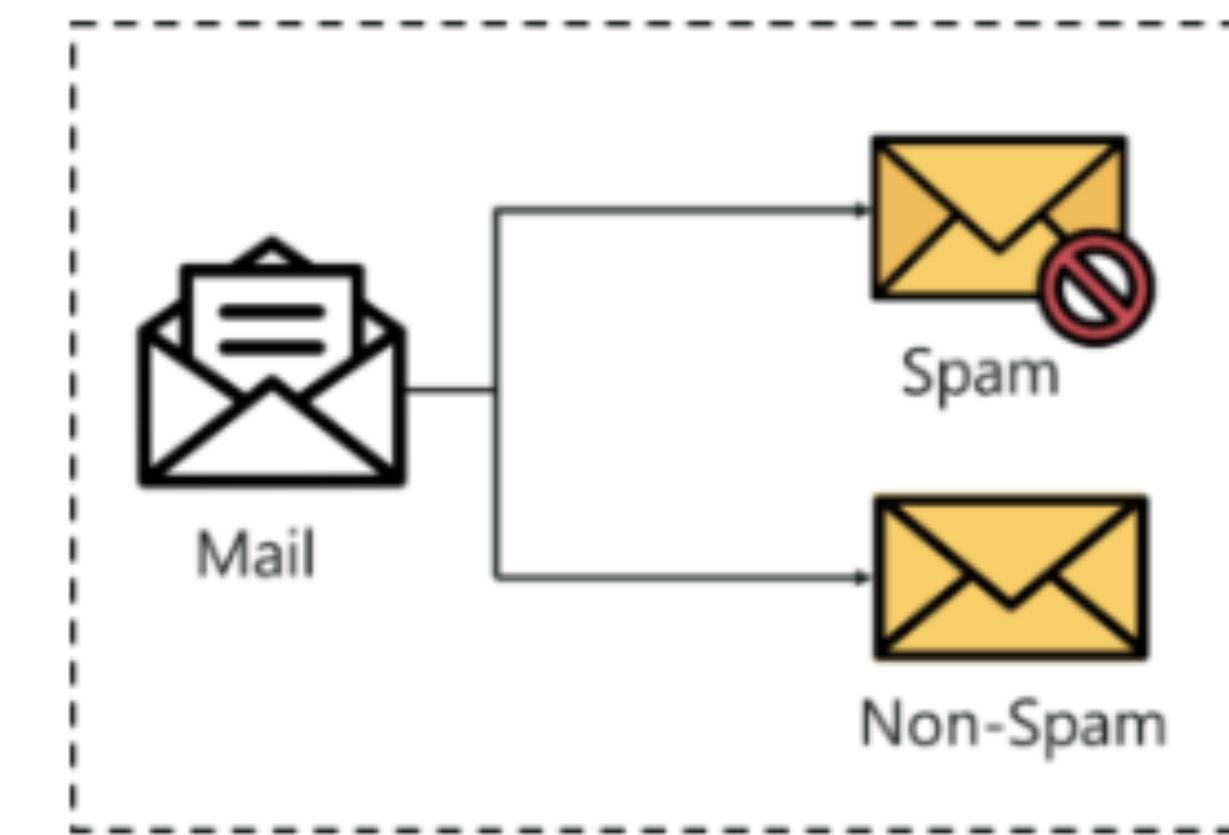
Tareas que **cambien con el tiempo** o con diferentes usuarios

# The Task, $T$

- El aprendizaje automático nos permite abordar tareas que son **demasiado difíciles de resolver con programas fijos escritos y diseñados por seres humanos.**
- El aprendizaje es nuestro medio para alcanzar la capacidad de realizar la tarea.

## La Tarea

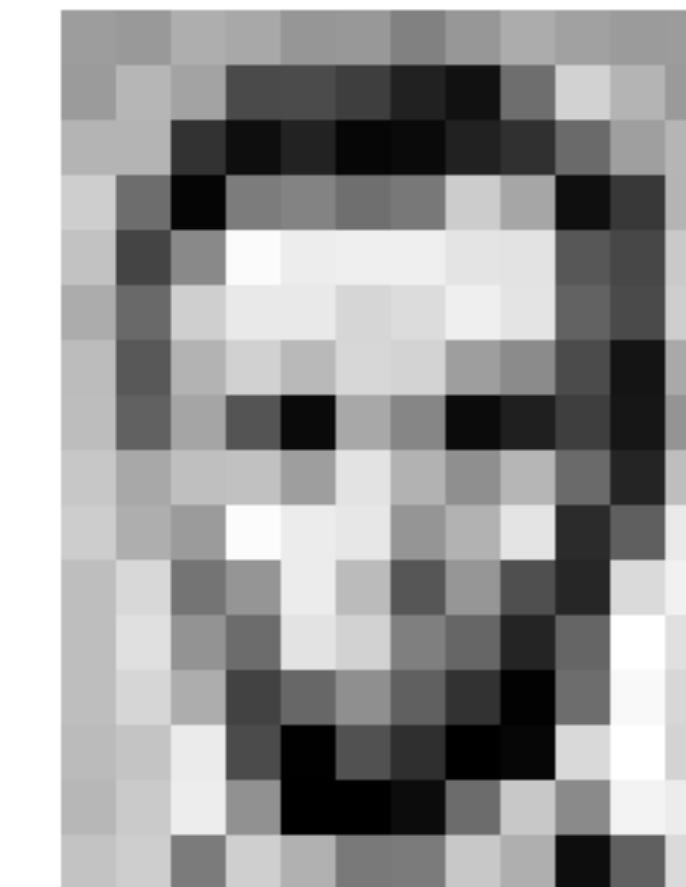
*Cómo el sistema de aprendizaje automático debe procesar un **ejemplo***



**Tarea de clasificación de SPAM**

# The Task, $T$

- **Un ejemplo u observación** es una colección de características que se han medido cuantitativamente de algún objeto o evento
- Queremos que procese el sistema de aprendizaje automático.
- Ejemplo como vector  $x \in R^n$  cada entrada  $x_i$  es una característica (o medición)



157	153	174	168	150	152	129	151	172	161	155	166
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	105	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	89	179	209	185	215	211	158	199	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	209	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	199	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	209	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

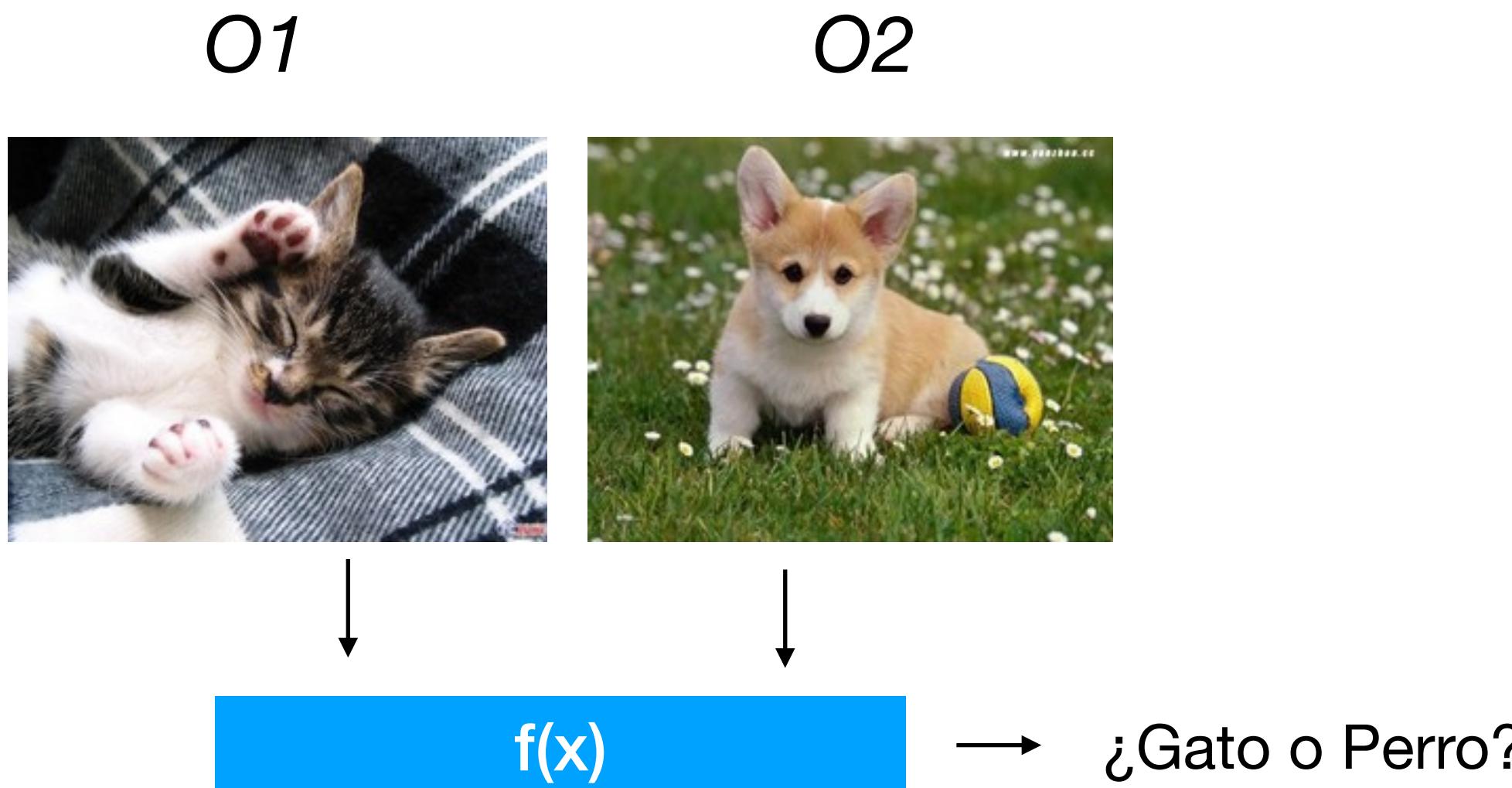
Las características de una imagen suelen ser los valores de los píxeles de la imagen.

# Machine Learning

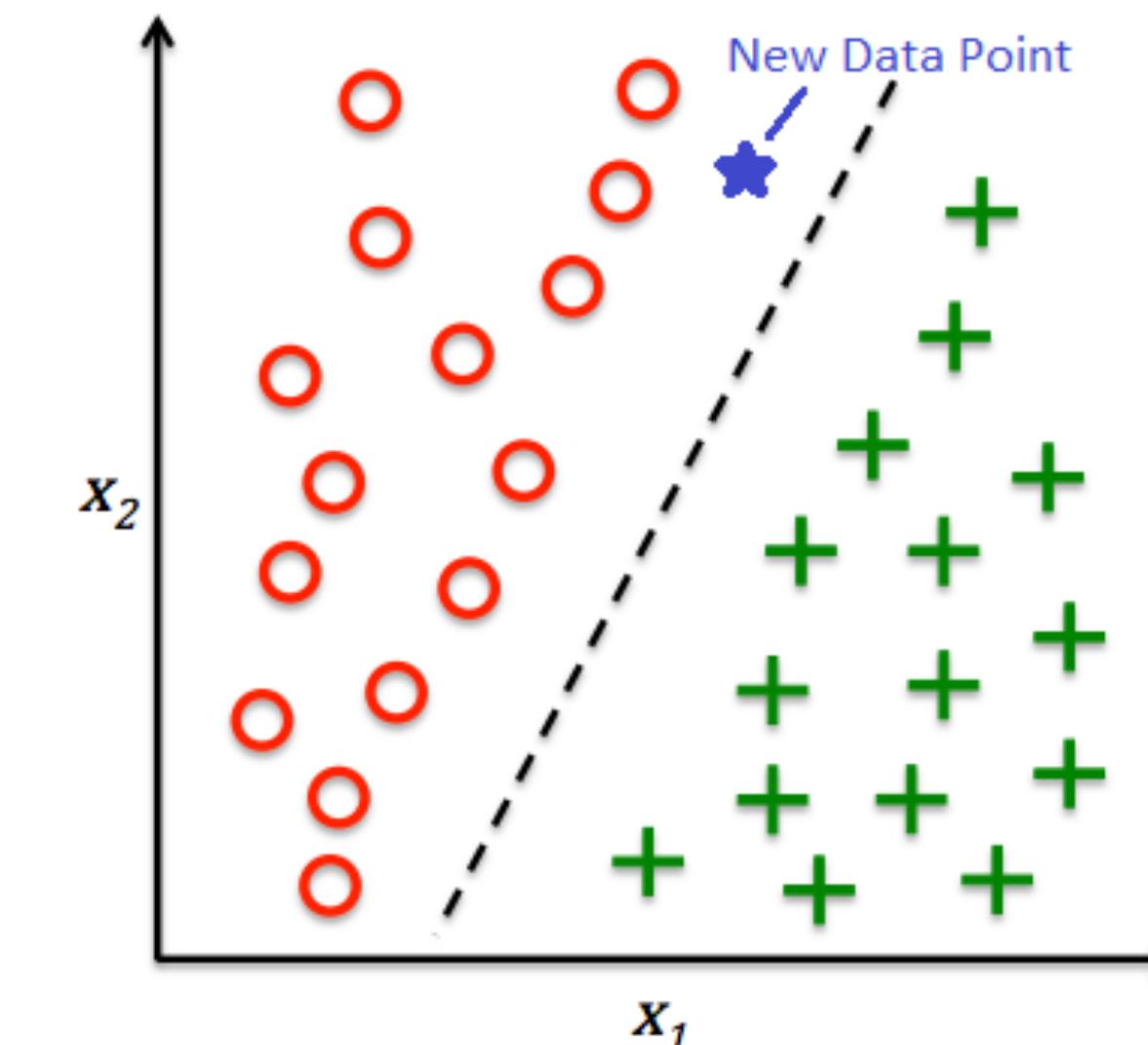
## Tasks

### Clasificación

Se pide al programa informático que especifique a cuál de  $k$  categorías pertenece una entrada



Producir una función  $f: \mathbb{R}^n \rightarrow \{1, \dots, n\}$   
Cuando  $y = f(x)$ , El modelo asigna una entrada descrita por el vector  $x$  a una categoría identificada por un código numérico  $y$



# Machine Learning

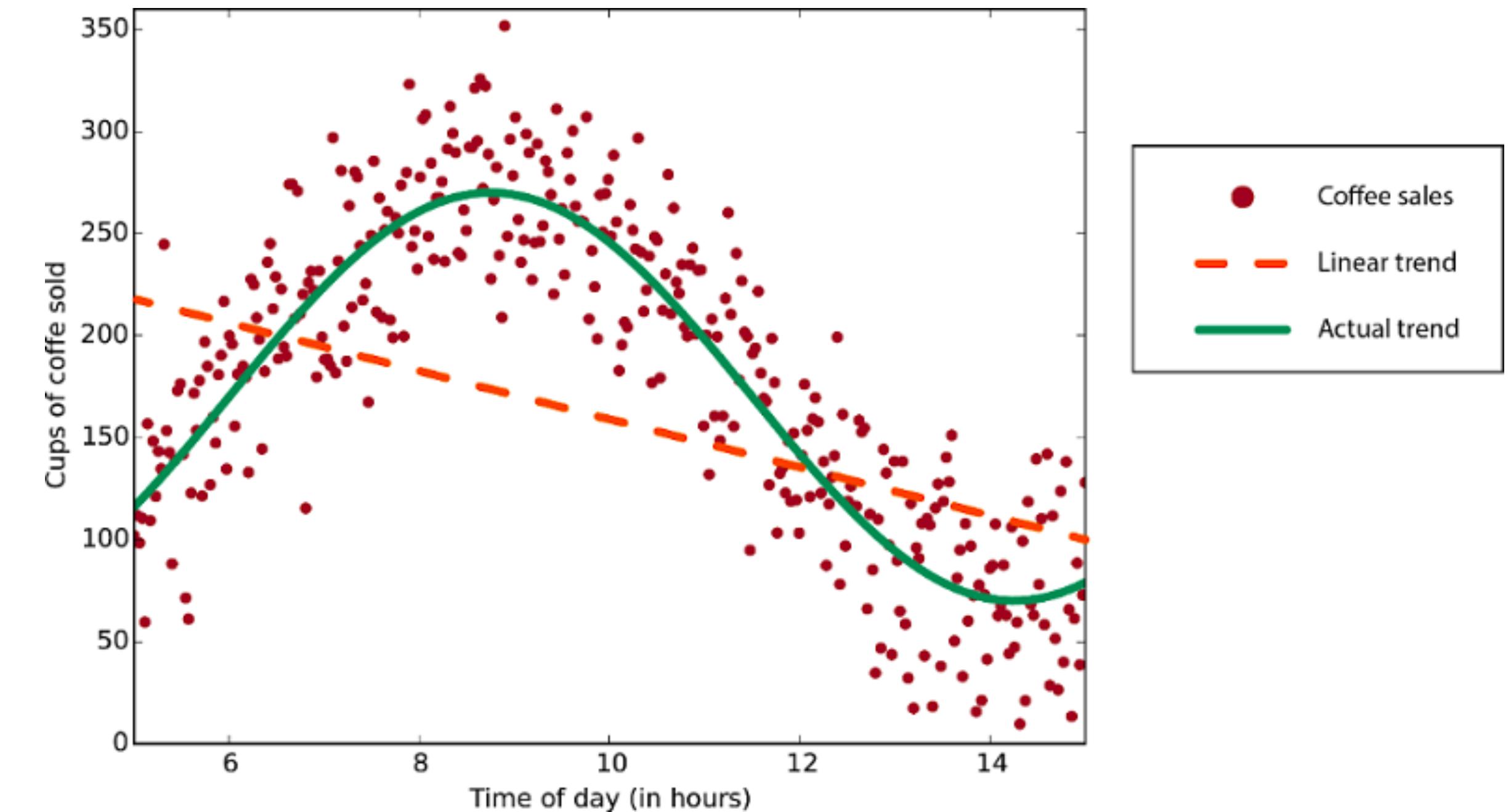
## Tasks

### Regresión

- Se pide al programa que prediga un valor numérico a partir de una entrada
- El algoritmo de aprendizaje produce una función

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

- La variable de salida es real y continua tal como el salario o el peso

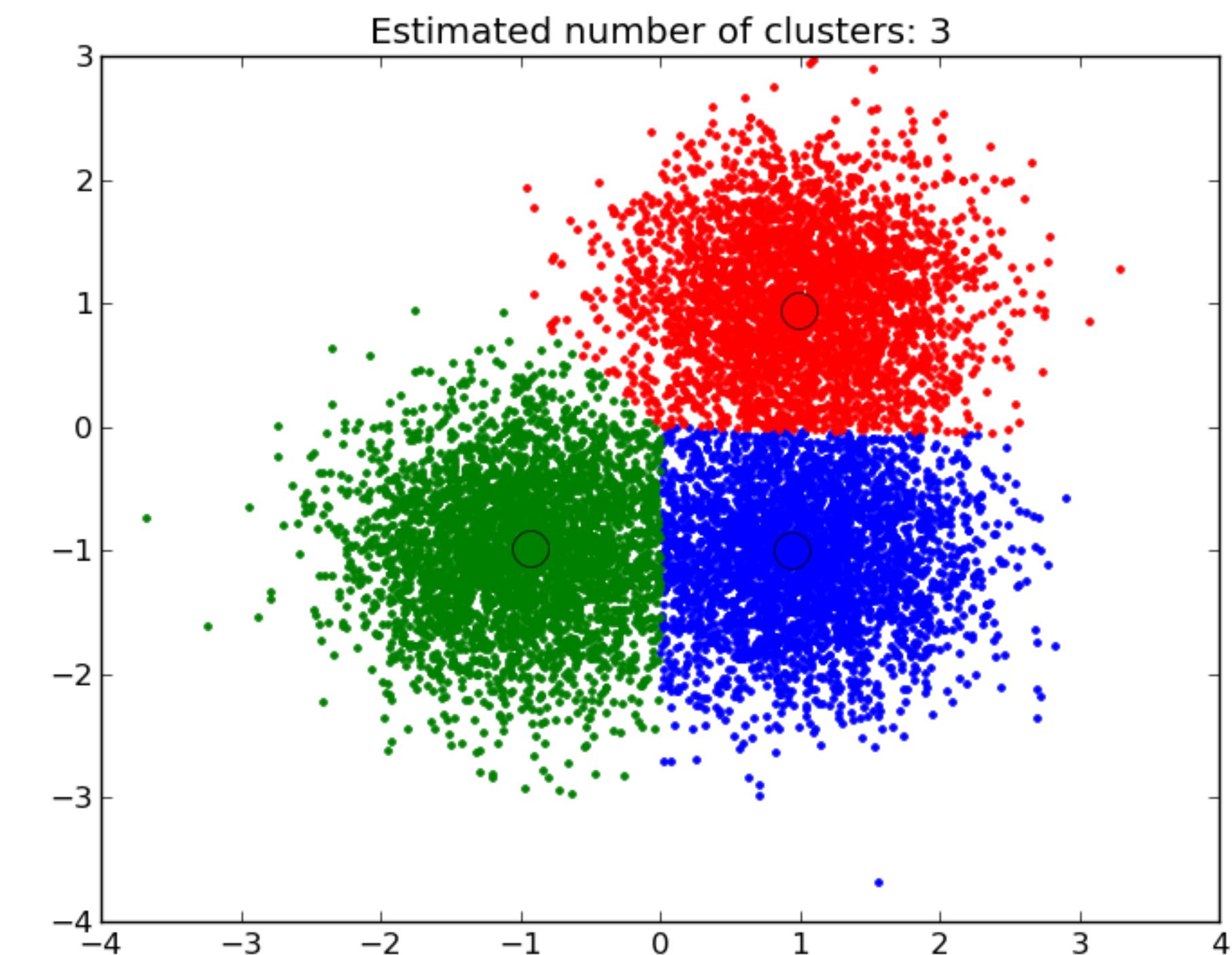


# Machine Learning

## Tasks

### Agrupamiento (Clustering)

- Es usado como el proceso para encontrar
  - Estructura significativa
  - Procesos explicativos
  - Características
  - Agrupamiento en un conjunto de datos
- Tarea de dividir una población o “data-points” en un número de grupos



# Machine Learning

## Tasks

### Otras tareas

- Captioning
  - Traducción
  - Detección de anomalías
  - Image generation
  - Denoising
- ...



DALL-E 2's interpretation of "A photo of an astronaut riding a horse."

# The experience, $E$

- Los algoritmos de aprendizaje automático se pueden clasificar en:
  - No supervisados
  - Supervisados
- Tipo de experiencia que se les permite tener durante el proceso de aprendizaje (Entrenamiento).

## Dataset

- La Experiencia es dada a través de un dataset
- Colección de muchos ejemplos (Data points)
- Subconjunto de ejemplos del dominio

8	9	0	1	2	3	4	7	8	9	0	1	2	3	4	5	6	7	8	6
4	2	6	4	7	5	5	4	7	8	9	2	9	3	9	3	8	2	0	5
0	1	0	4	2	6	5	3	5	3	8	0	0	3	4	1	5	3	0	8
3	0	6	2	7	1	1	8	1	7	1	3	8	9	7	7	6	7	4	1
7	5	1	7	1	9	8	0	6	9	4	9	9	3	7	1	9	2	2	5
3	7	8	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	0
1	2	3	4	5	6	7	8	9	8	1	0	5	5	1	9	0	4	1	9
3	8	4	7	7	8	5	0	6	5	5	3	3	3	9	8	1	4	0	6
1	0	0	6	2	1	1	3	2	8	8	7	8	4	6	0	2	0	3	6
8	7	1	5	9	9	3	2	4	9	4	6	5	3	2	8	5	9	4	1
6	5	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
8	9	0	1	2	3	4	5	6	7	8	9	6	4	2	6	4	7	5	5
4	7	8	9	2	9	3	9	3	8	2	0	9	8	0	5	6	0	1	0
4	2	6	5	5	5	4	3	4	1	5	3	0	8	3	0	6	2	7	1
1	8	1	7	1	3	8	5	4	2	0	9	7	6	7	4	1	6	8	4
7	5	1	2	6	7	1	9	8	0	6	9	4	9	9	6	2	3	7	1
9	2	2	5	3	7	8	0	1	2	3	4	5	6	7	8	0	1	2	3
4	5	6	7	8	0	1	2	3	4	5	6	7	8	9	2	1	2	1	3
9	9	8	5	3	7	0	7	7	5	7	9	9	4	7	0	3	4	1	4
4	7	5	8	1	4	8	4	1	8	6	4	6	3	5	7	2	5	9	

Datasets de dígitos escritos a mano (MNIST dataset)

# Machine Learning

## Tipos de aprendizaje

### Supervisado

- Experimentan un conjunto de datos con:
  - Las características
  - Asociado a una etiqueta u objetivo.
- Algoritmos aprenden a asociar una entrada con una salida
- Dataset contiene entrada x - salidas y
- Algunos casos las salidas son difíciles de obtener



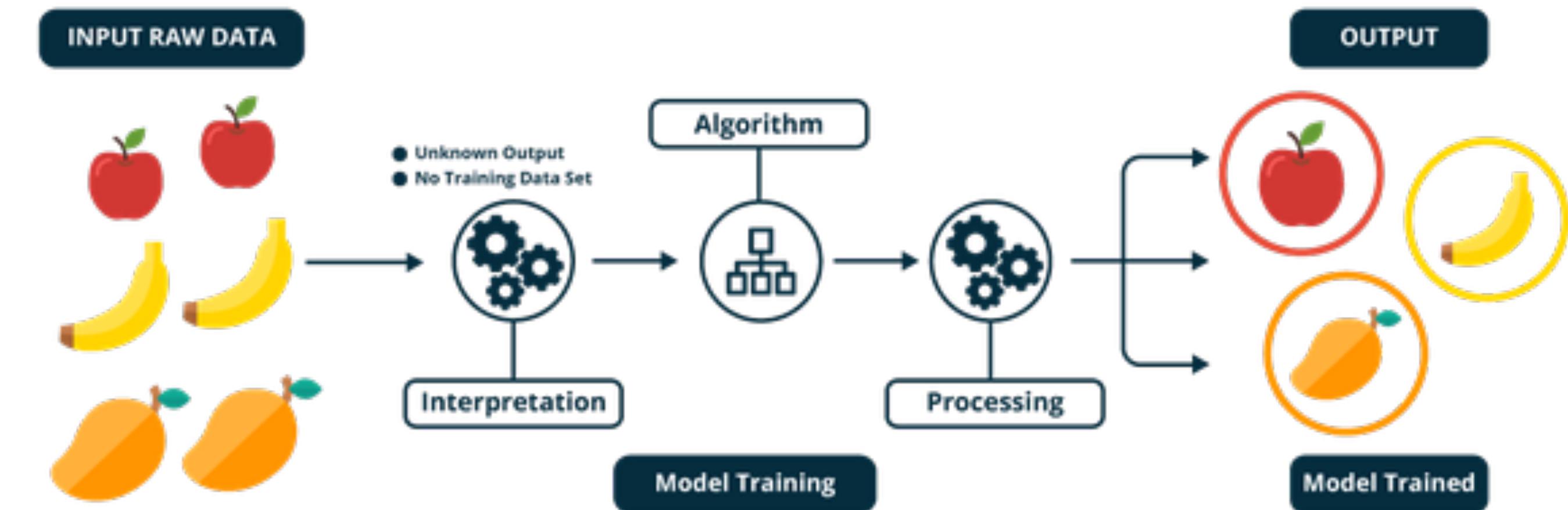
# Machine Learning

## Tipos de aprendizaje

### Sin Supervisión

*Experimentan un conjunto de datos que contiene muchas características y aprenden propiedades útiles de la estructura de este conjunto de datos*

- Solo experimentan las “features”
- Extraer información de una distribución de datos
- No requiere labor humana para anotar los ejemplos
- No una señal de supervisión



# Datasets

- Una matriz que contiene un ejemplo diferente en cada fila.
- Cada columna de la matriz corresponde a una característica diferente.

## Iris dataset

- Uno de los conjuntos de datos más antiguos
- Mediciones de diferentes partes de 150 plantas de iris.
- Tres tipos de Iris: Setosa, Versicolour, Virginica
- Cada planta individual corresponde a un ejemplo.
- Características: longitud del sépalo: anchura del sépalo, longitud del pétalo y anchura del pétalo

SL	SW	PL	PW	Specie
5.1	3.5	1.4	0.2	0
4.9	3	1.4	0.2	0
...	...	...	...	...
5.7	2.8	4.1	1.3	2

Iris dataset (Fisher, 1936).

<https://archive.ics.uci.edu/ml/datasets/iris>

# Datasets

## Notación

$m$	Número de muestras
$x$	Variables de entrada o características
$y$	Variable de salida u objetivo
$(x^{(i)}, y^{(i)})$	Muestra específica $i$

$X$

$y$

<b>SL</b>	<b>SW</b>	<b>PL</b>	<b>PW</b>	<b>Specie</b>
5.1	3.5	1.4	0.2	0
4.9	3	1.4	0.2	0
...	...	...	...	...
5.7	2.8	4.1	1.3	2

m

$X \in \mathbb{R}^{150 \times 4}$

$y \in \mathbb{R}^{150}$

# Regresión lineal

- Algoritmo sencillo de aprendizaje automático
- Tomar un vector  $x \in \mathbb{R}^n$  y predecir un valor escalar  $y \in \mathbb{R}$
- La salida de la regresión lineal es una función lineal de la entrada
- Donde  $w \in \mathbb{R}^n$  es un vector de parámetros
- **La predicción es la suma ponderada de las características por sus pesos ( $\theta$ )**

$$\hat{y} = \theta_0 + \theta_1 x$$

$$\hat{y} = \theta^T x$$

$$\begin{matrix} \theta_0 & \theta_1 & \theta_2 & \theta_3 \\ \bullet & & & \\ \begin{matrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{matrix} & = & \hat{y} \end{matrix}$$

- El primer término se llama Bias ( $\theta_0$ )

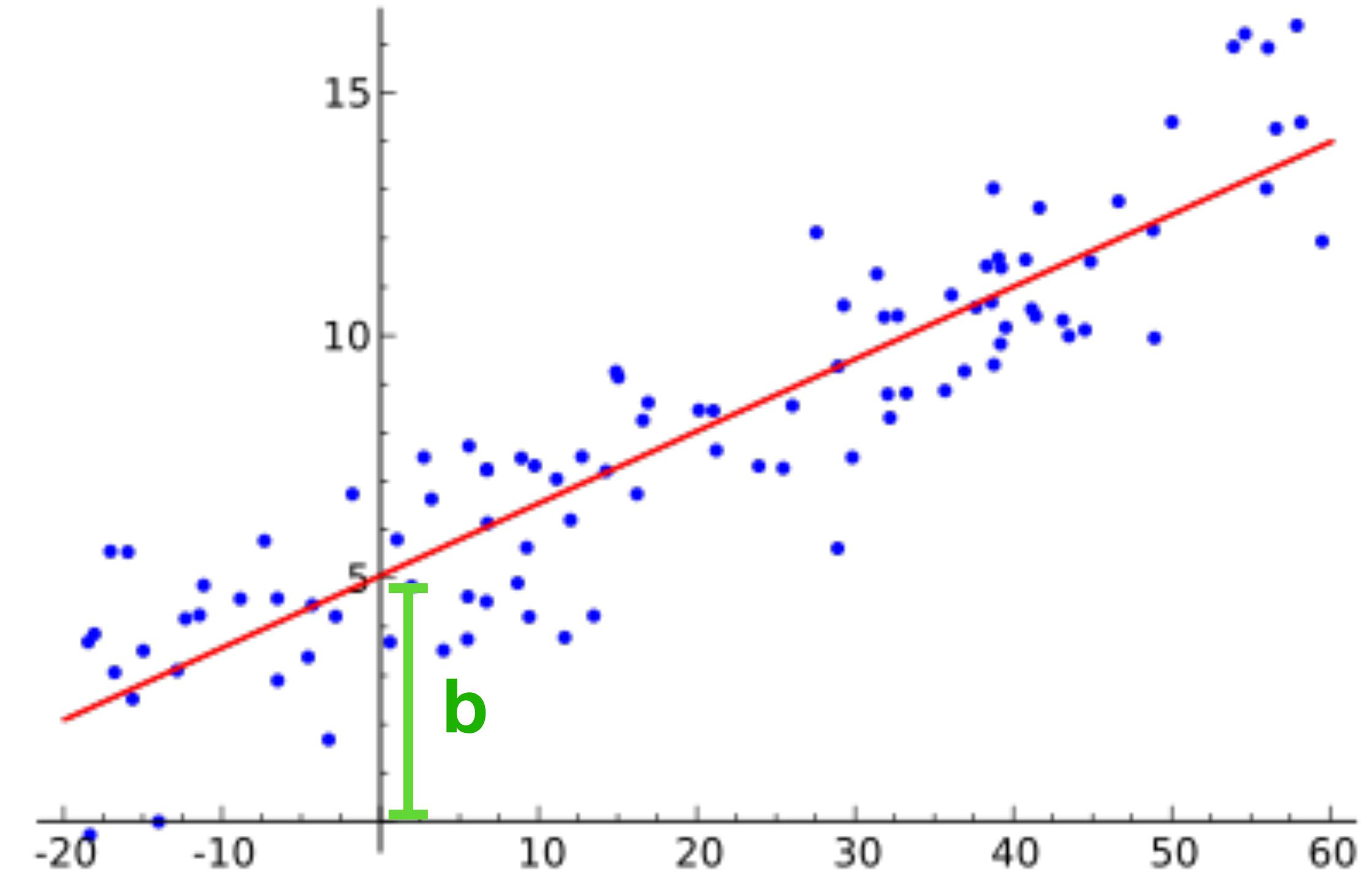
$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

# Bias term

- El intercepto  $\theta_0$  o  $b$  es llamado el término *Bias*
- La salida está sesgada hacia  $b$  en ausencia de cualquier entrada

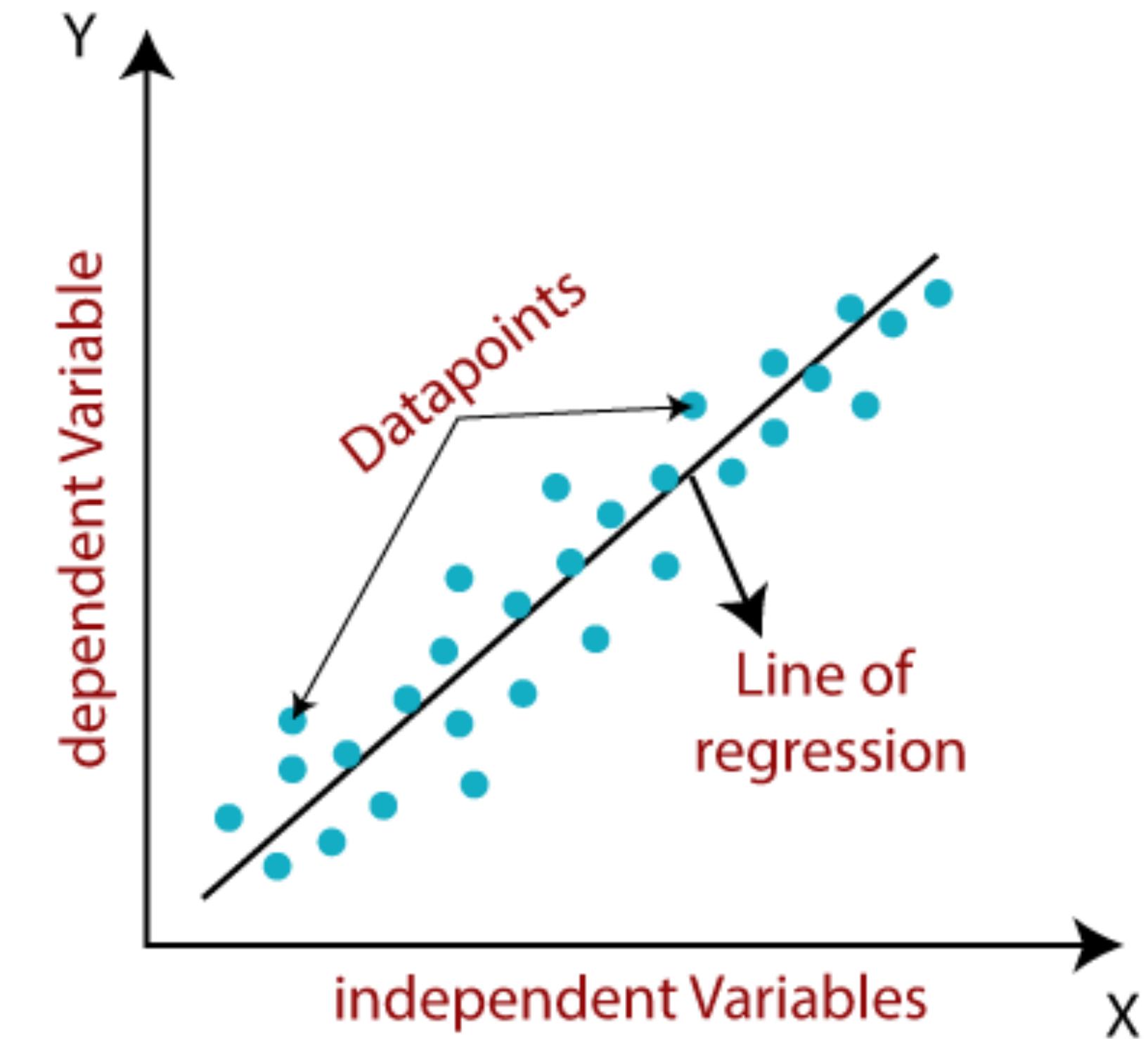
## De forma práctica

- En lugar de añadir el parámetro de sesgo  $\theta_0$
- Seguir utilizando el modelo sólo con ponderaciones
- Aumentar  $X$  con una entrada extra que siempre se establece en 1



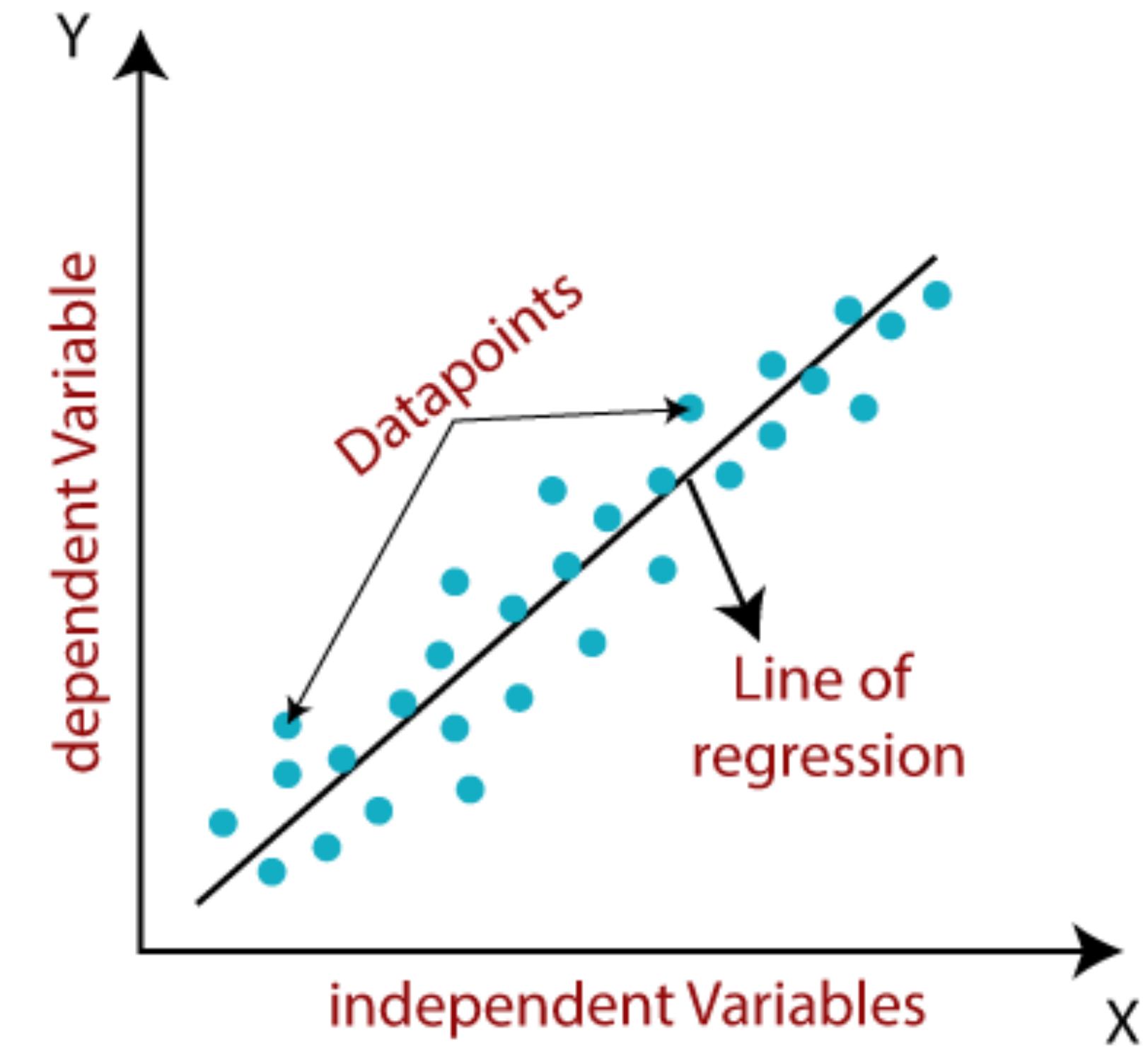
# Regresión lineal

- $w_i$  coeficiente de la característica  $x_i$  antes de sumar las contribuciones de todas las características.
- Pensemos  $w$  como un conjunto de pesos que determinan cómo afecta cada característica a la predicción.



# Regresión lineal

- Si una característica recibe un peso positivo, entonces aumentar el valor de esa característica aumenta el valor de nuestra predicción
- Si la ponderación de una característica es negativa, su valor disminuye.
- Si la ponderación de una característica es grande, tiene un gran efecto en la predicción.
- Si el peso de una característica es cero, no tiene ningún efecto en la predicción.

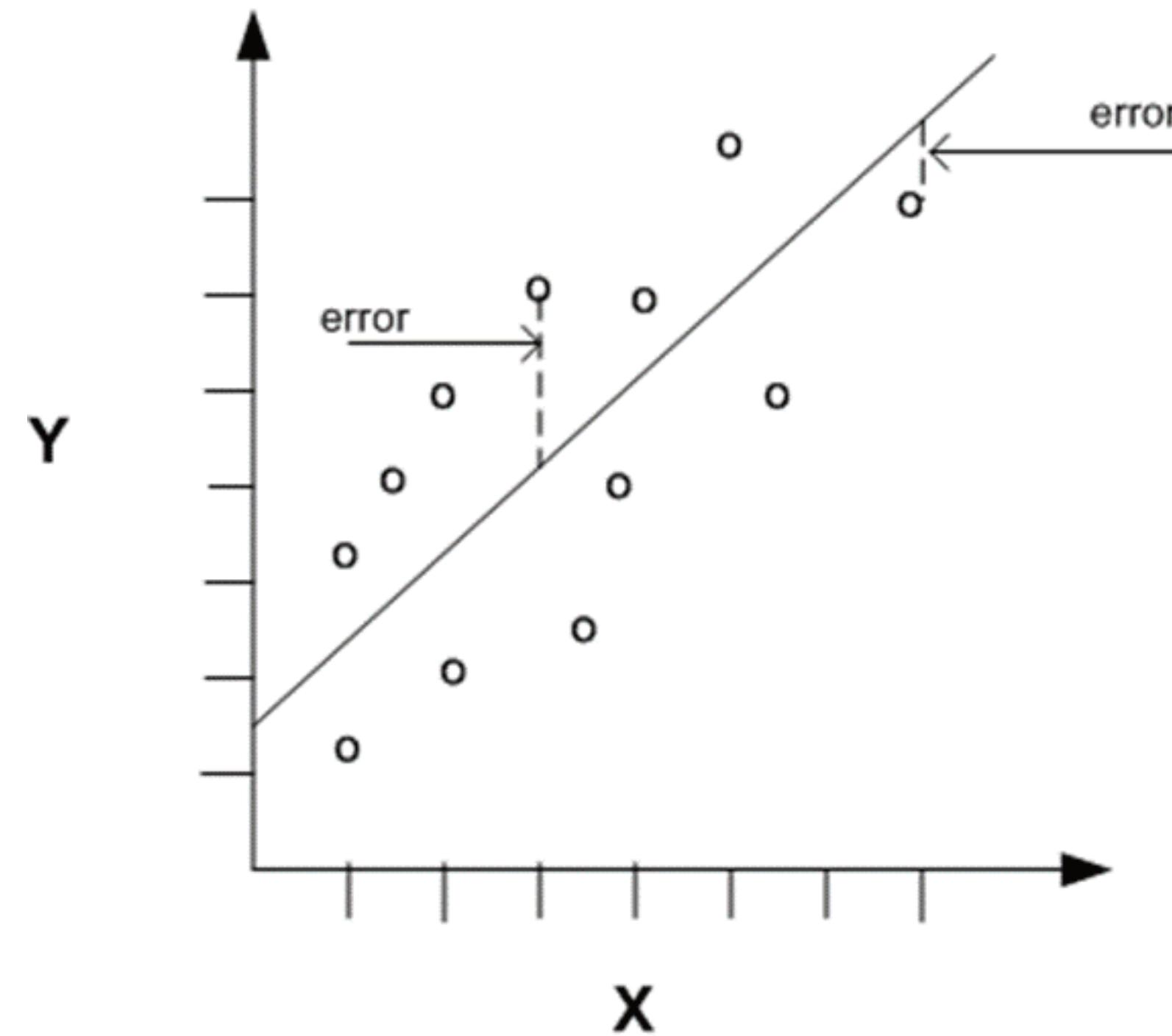


# Regresión lineal

- La tarea  $T$ : predecir  $y$  basado  $x$  calculando  $\hat{y} = \theta^T x$ .
- Necesitamos medir el rendimiento  $P$  de nuestro modelo
- Una forma de medir el rendimiento del modelo es calcular el **error cuadrático medio (MSE)** del modelo

## Dataset test

Para medir el desempeño se reserva una parte del dataset exclusivamente para la evaluación del modelo



$$MSE = \frac{1}{m} \sum_i (\hat{y}_i - y_i)^2$$

# Regresión lineal

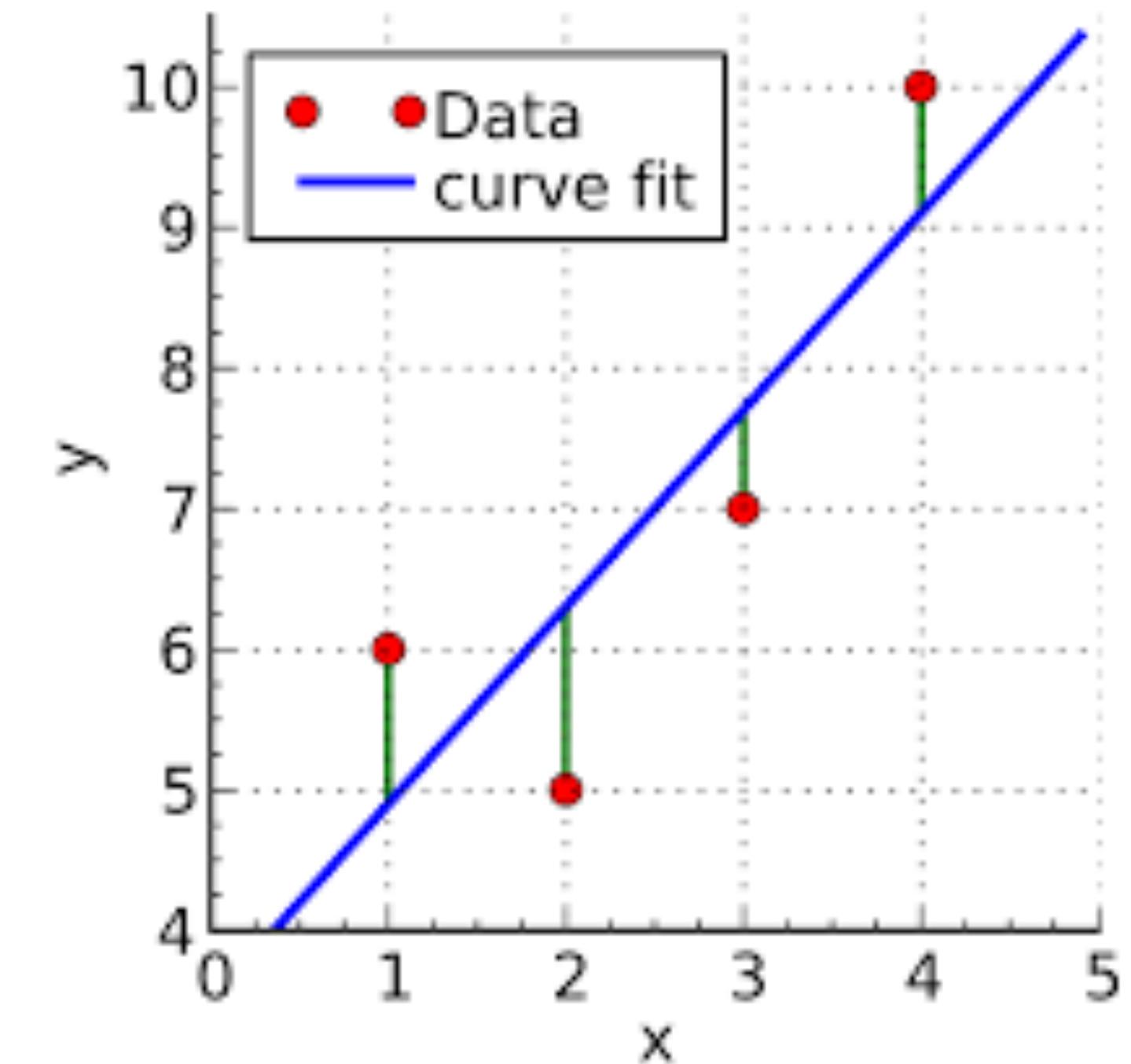
El error del modelo es la suma cuadrática de las diferencias de la predicción  $\hat{y}$  menos la respuesta correcta  $y$  de cada muestra  $i$

$$MSE = \frac{1}{m} \sum_i (\hat{y}_i - y_i)^2$$

Es equivalente a la distancia Euclíadiana entre las predicciones y los objetivos

$$MSE = \frac{1}{m} \|\hat{y} - y\|_2^2$$

Necesitamos diseñar un algoritmo que mejore los pesos  $w$  de forma que disminuya  $MSE$

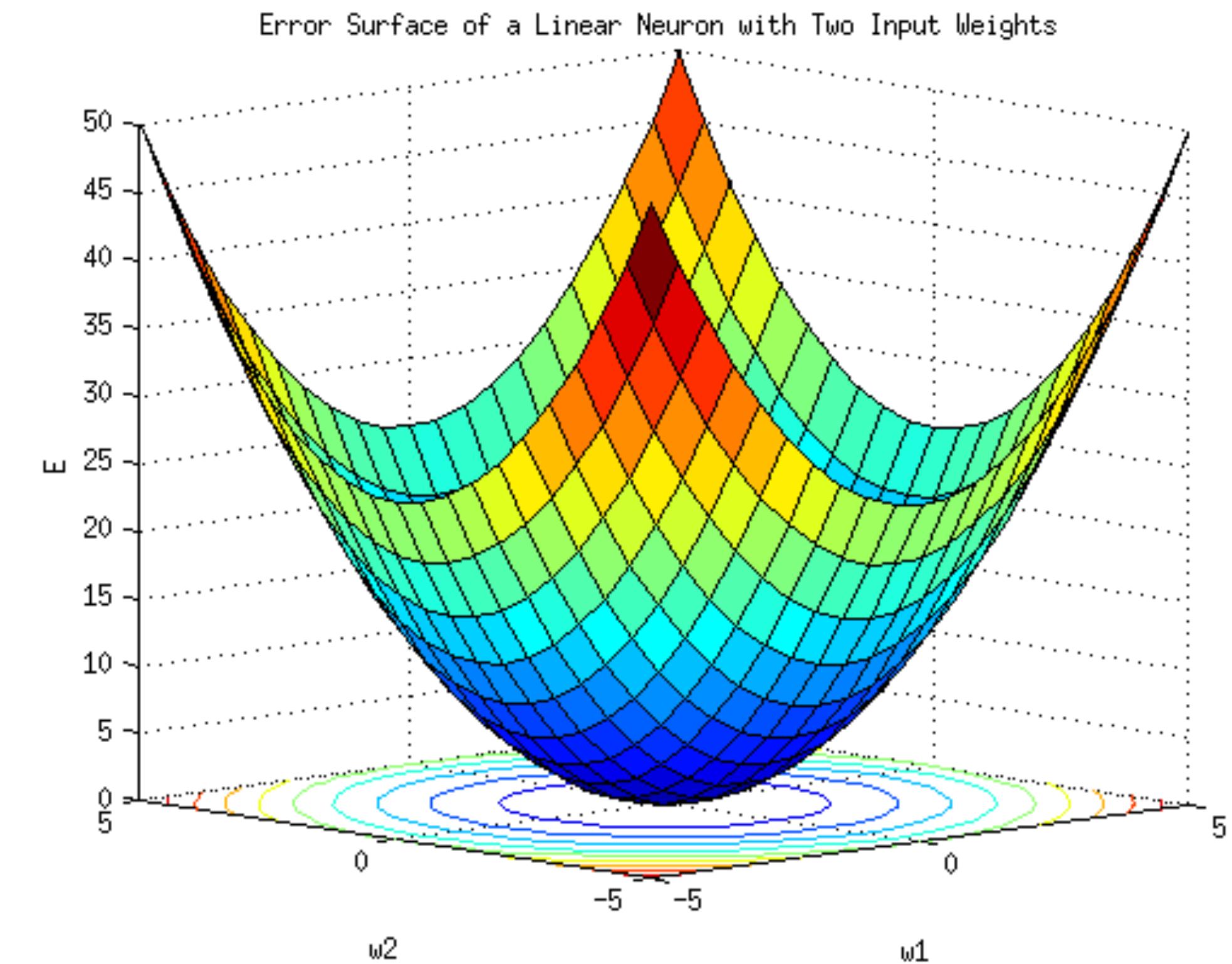


# Regresión lineal

- Para minimizar MSE se resuelve donde su gradiente es 0

$$\nabla_w \frac{1}{m} \|\hat{y} - y\|_2^2 = 0$$

- Soluciones:
  - Analítica: Normal equations
  - Iterativa: Gradient Descent



# Normal equations

Mínimo local es cuando la gradiente de  
MSE = 0

$$\nabla_w \|Xw - y\|_2^2 = 0$$

$$\nabla_w [(Xw - y)^T (Xw - y)] = 0$$

...

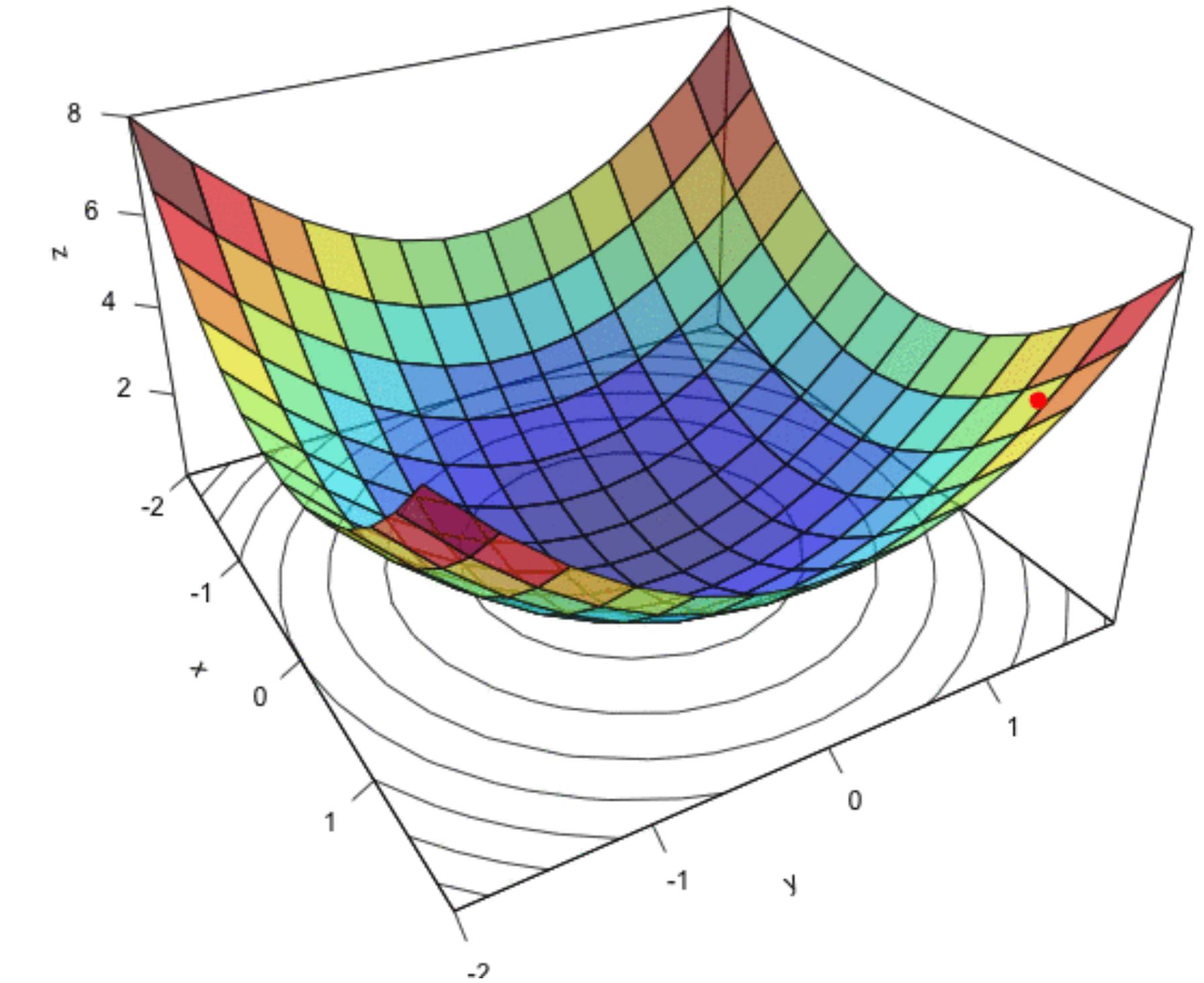
$$\underline{w = (X^T X)^{-1} X^T y}$$

## Anotaciones

- ✓ Se resuelve de forma directa (no hay que hacer iteraciones)
- ✓ No hay necesidad de manejar parámetros adicionales
- ✗ Solo funciona en el caso lineal
- ✗ Lento cuando se tienen muchas características ( $n > 10^4$ )
- ✗ Si  $(X^T X)$  no es invertible hay problemas

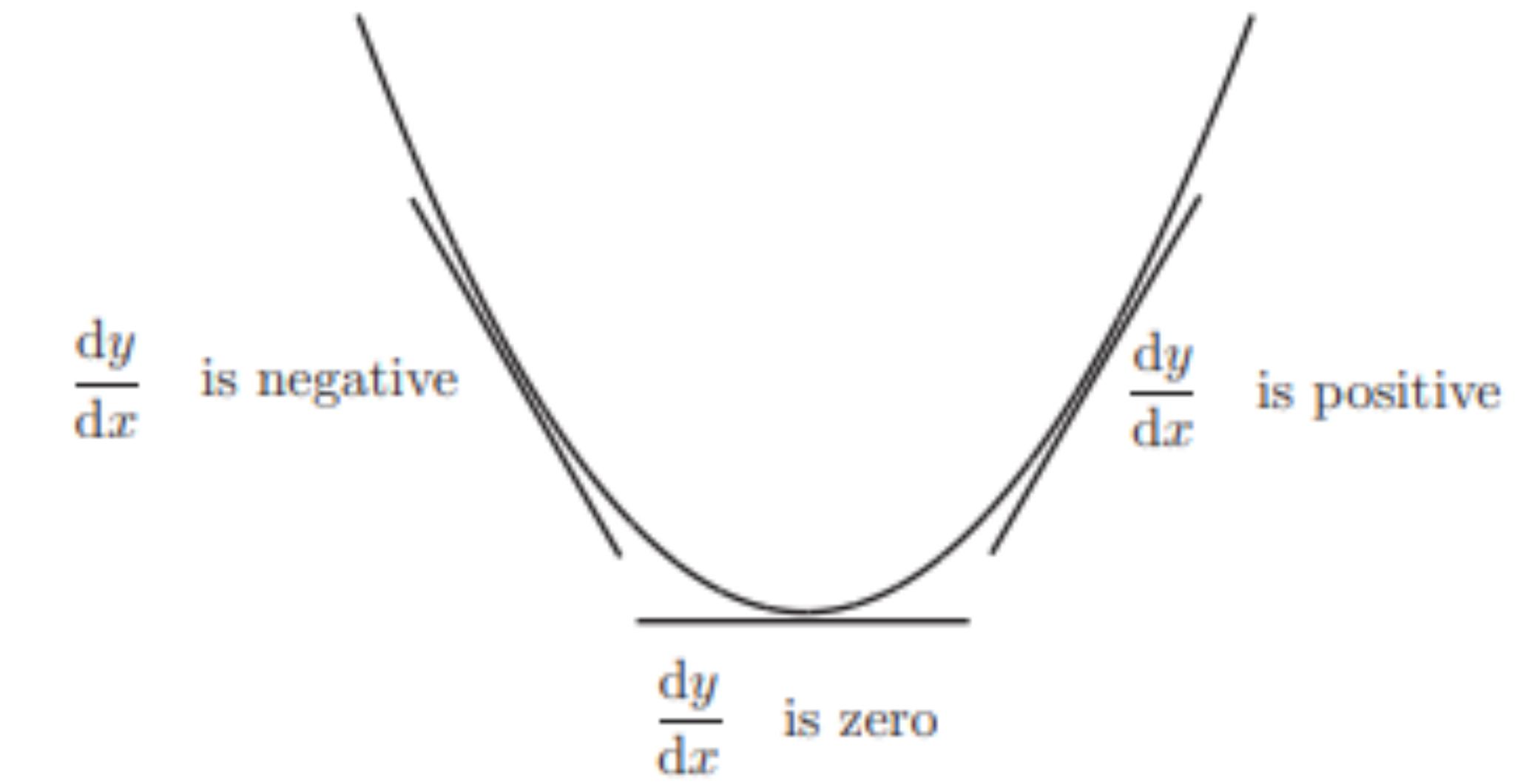
# Gradient Descent

- La mayoría de los algoritmos de aprendizaje profundo implican algún tipo de optimización.
- La optimización se refiere a la tarea de minimizar o maximizar alguna función  $f(\theta)$  alterando  $\theta$
- Nombres que recibe la función que deseamos minimizar  $J(\theta)$ : objective, cost, loss, error function



# Gradient Descent

- La derivada es útil para minimizar una función porque nos dice cómo cambiar  $\theta$  para conseguir una pequeña mejora en  $J(\theta)$
- Así pues, podemos reducir  $J(\theta)$  moviendo  $\theta$  en pequeños pasos con el signo contrario de la derivada

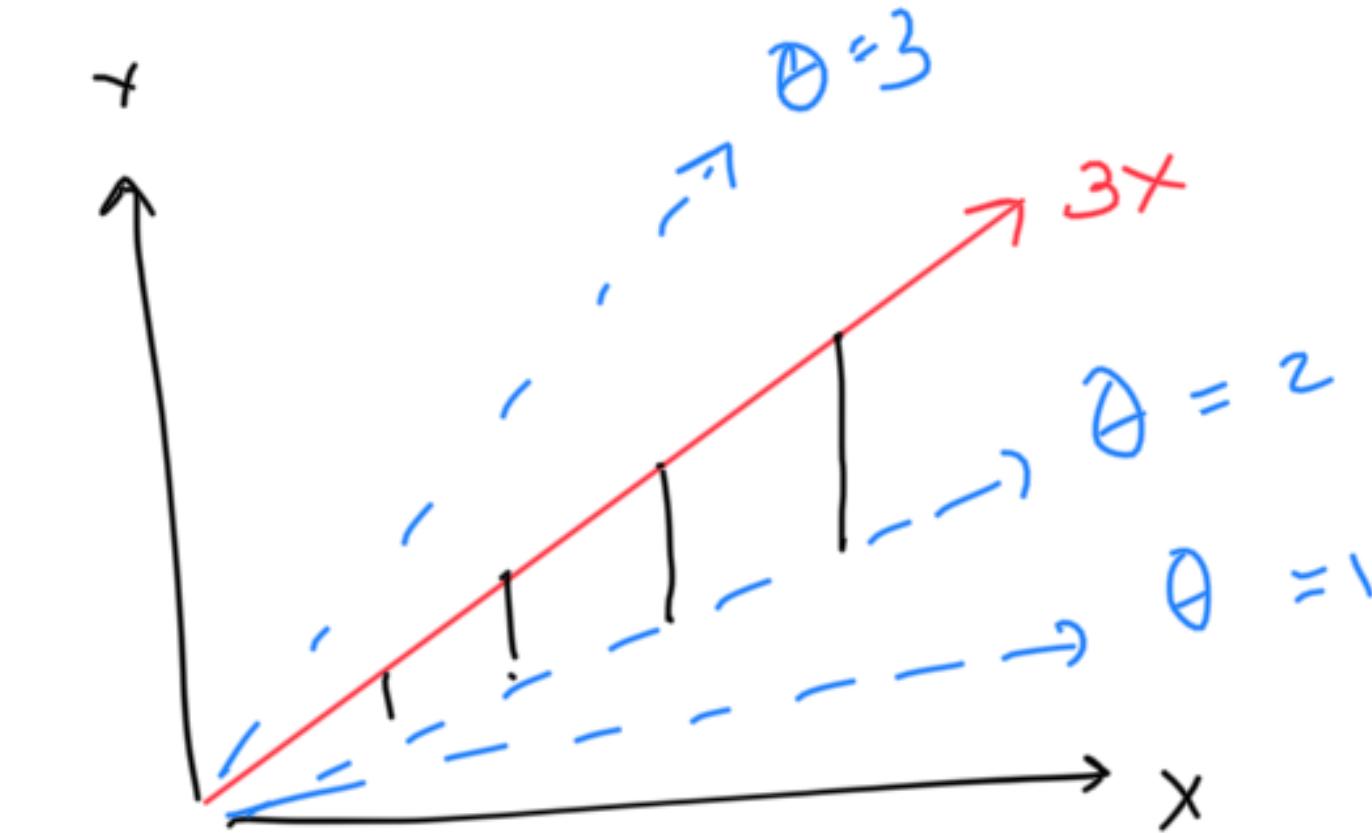


**Derivada: pendiente de la linea tangente en el punto evaluado**

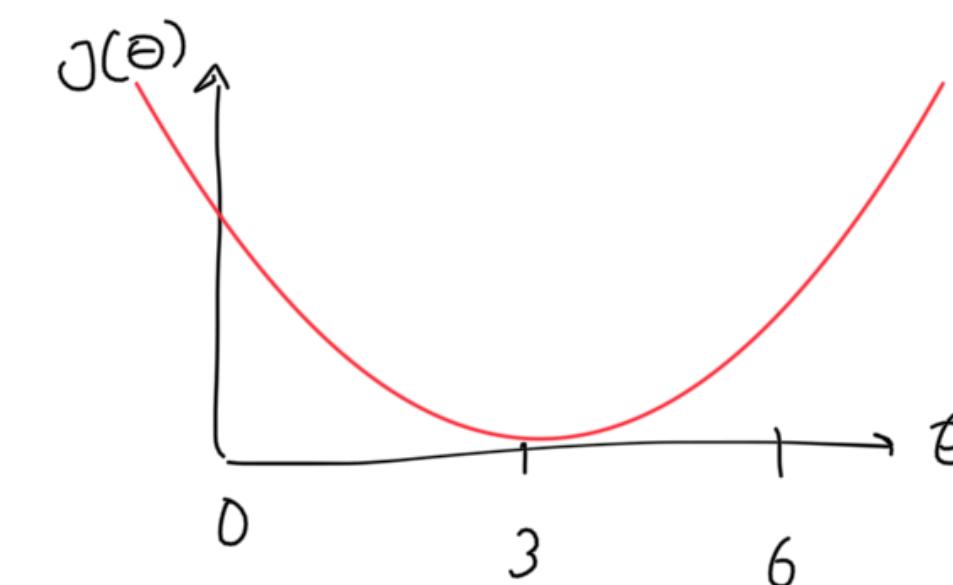
# Gradient Descent

## Idea principal (Caso teórico)

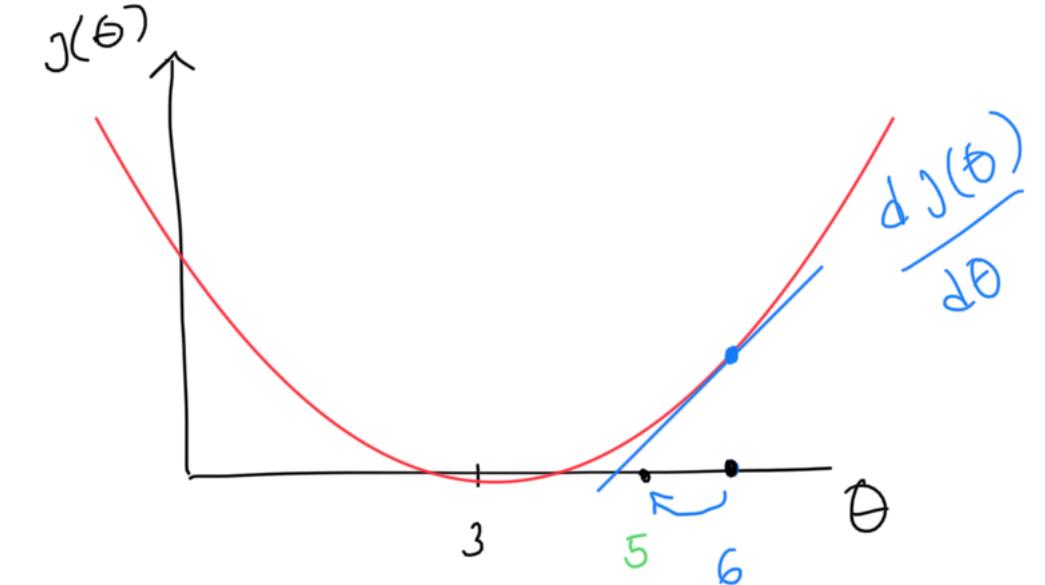
- Tenemos datos que pertenecen a la función  $y = 3x$  (Esta es la función que queremos recrear en rojo)
- Nuestras posibles soluciones (en azul) son el conjunto de líneas con pendientes  $\theta$
- No es viable evaluar todas las posibles soluciones.
- Con una solución evaluada decidir cuál es el próximo valor a evaluar que sea mejor que el anterior



Función objetivo en rojo y posibles soluciones en azul. Error en líneas negras



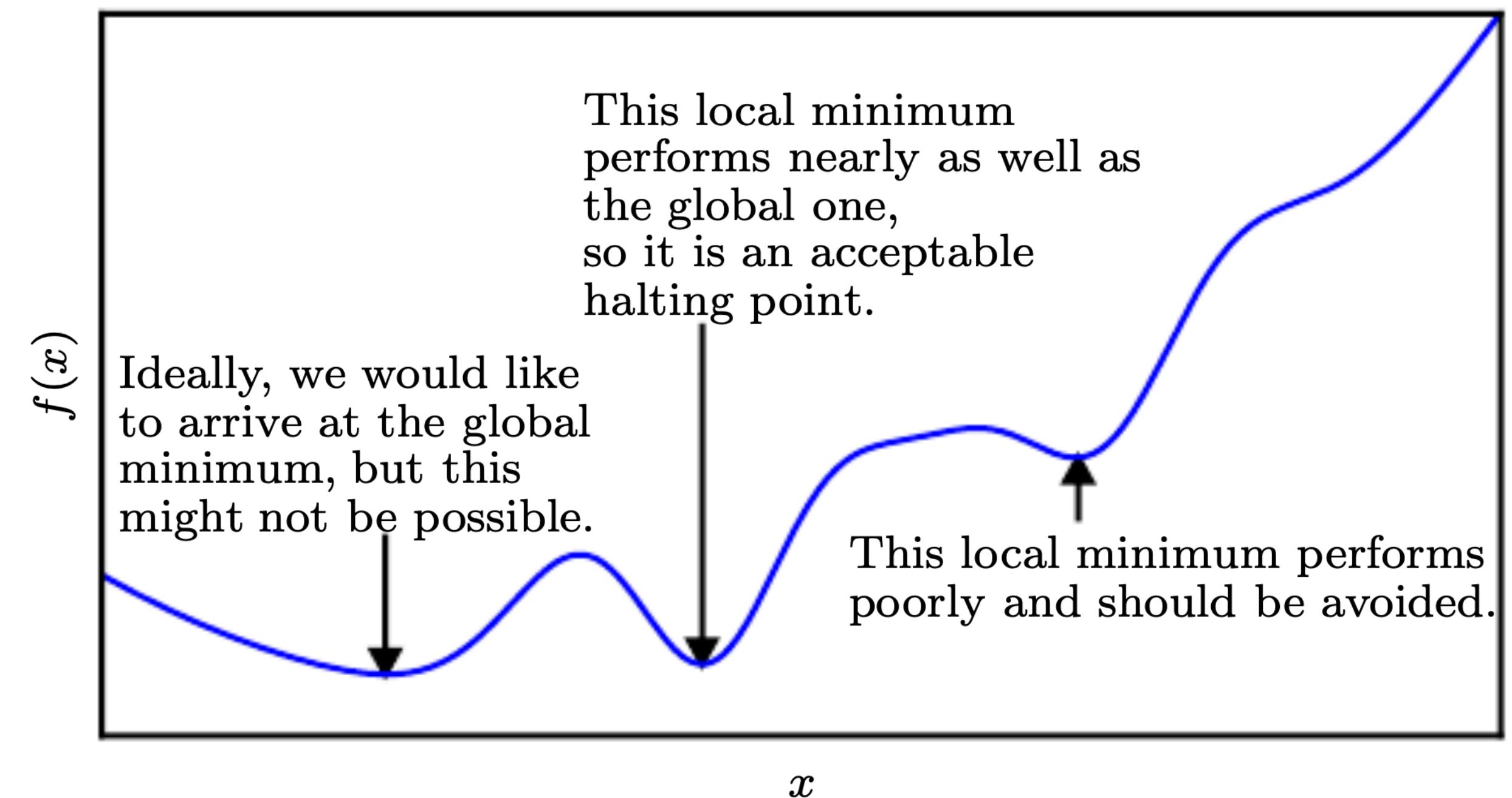
Gráfica de MSE en rojo de las posibles líneas con pendientes  $\theta$ . Se hace 0 en 3



Moverme en el sentido contrario a la derivada de la función de costo

# Gradient Descent

- Optimizamos funciones que pueden tener muchos mínimos locales que no son óptimos
- Todo esto hace que la optimización sea difícil, especialmente cuando la entrada de la función es multidimensional.
- Normalmente nos conformamos con encontrar un valor que sea muy bajo,
- No necesariamente mínimo absoluto



# Gradient Descent

- Para las funciones con múltiples entradas, debemos recurrir al concepto de derivadas parciales .
- La derivada parcial  $\frac{\partial}{\partial x_i} f(x)$  mide como  $f$  cambia con respecto la variable  $x_i$  aumenta en el punto  $x$
- El gradiente generaliza la noción de derivada al caso en que la derivada es con respecto a un vector
- La gradiente de  $f$  es el vector que contiene todas las derivadas parciales, denominadas  $\nabla_x f(x)$

**Derivadas parciales de “One half MSE”**

$$\frac{\partial}{\partial \theta} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta} \frac{1}{2m} \sum_i^m (\theta_0 + \theta_1 x_i - y)^2$$

...

Derivadas parciales por  $\theta$

$$\frac{\partial}{\partial \theta_0} J = \frac{1}{m} \sum_i^m \hat{y}_i - y_i$$

$$\frac{\partial}{\partial \theta_1} J = \frac{1}{m} \sum_i^m (\hat{y}_i - y_i) x_i$$

Donde:

$$\hat{y}_i = \theta_0 + \theta_1 x_i$$

# Gradient Descent

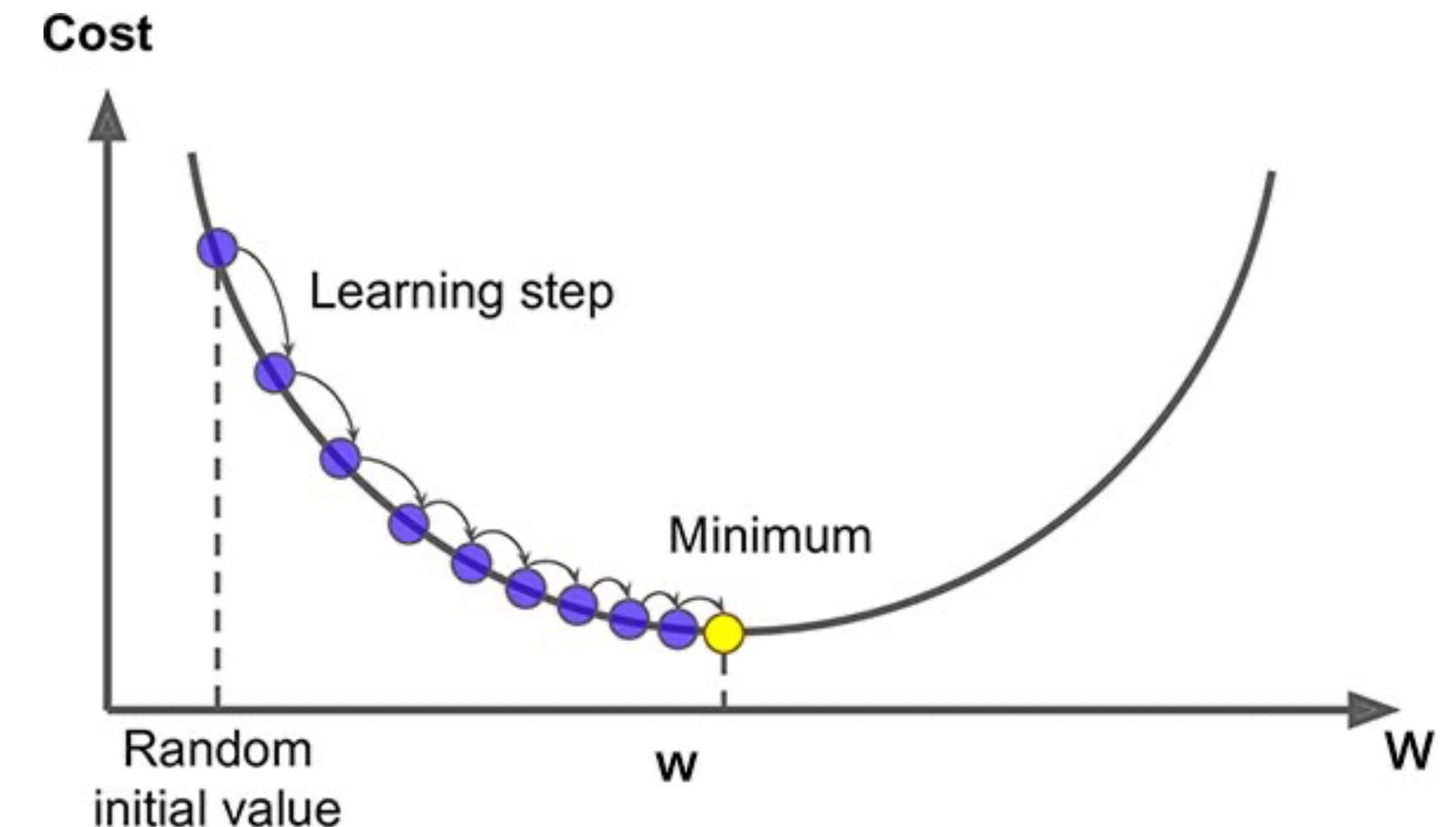
## Algoritmo

1. Inicializar en valores aleatorios cercanos al  $0^*$
2. Repetir hasta convergencia\*

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

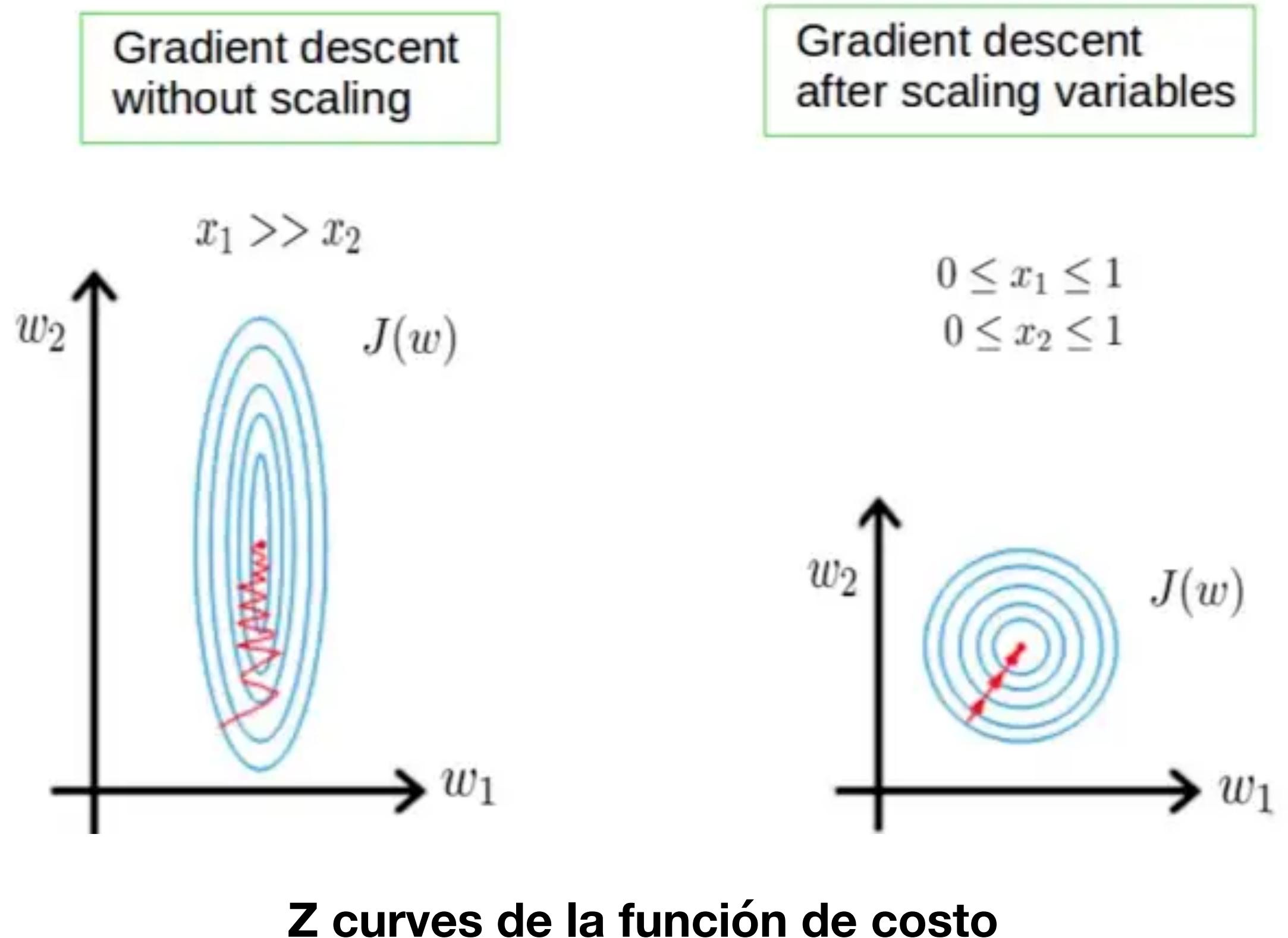
## Learning Rate ( $\alpha$ )

Parámetro que determina que tan grande moverme en el siguiente paso



# Feature Scaling

- El escalado de características en el aprendizaje automático es uno de los pasos más importantes del preprocesamiento
- Poner los valores de las características en el mismo rango
- Eliminar el sesgo de que los números grandes tienen mayor influencia
- GD converge mucho más rápido
- Los dos métodos más comunes de escalado de características son la estandarización y la normalización.



# Feature scaling

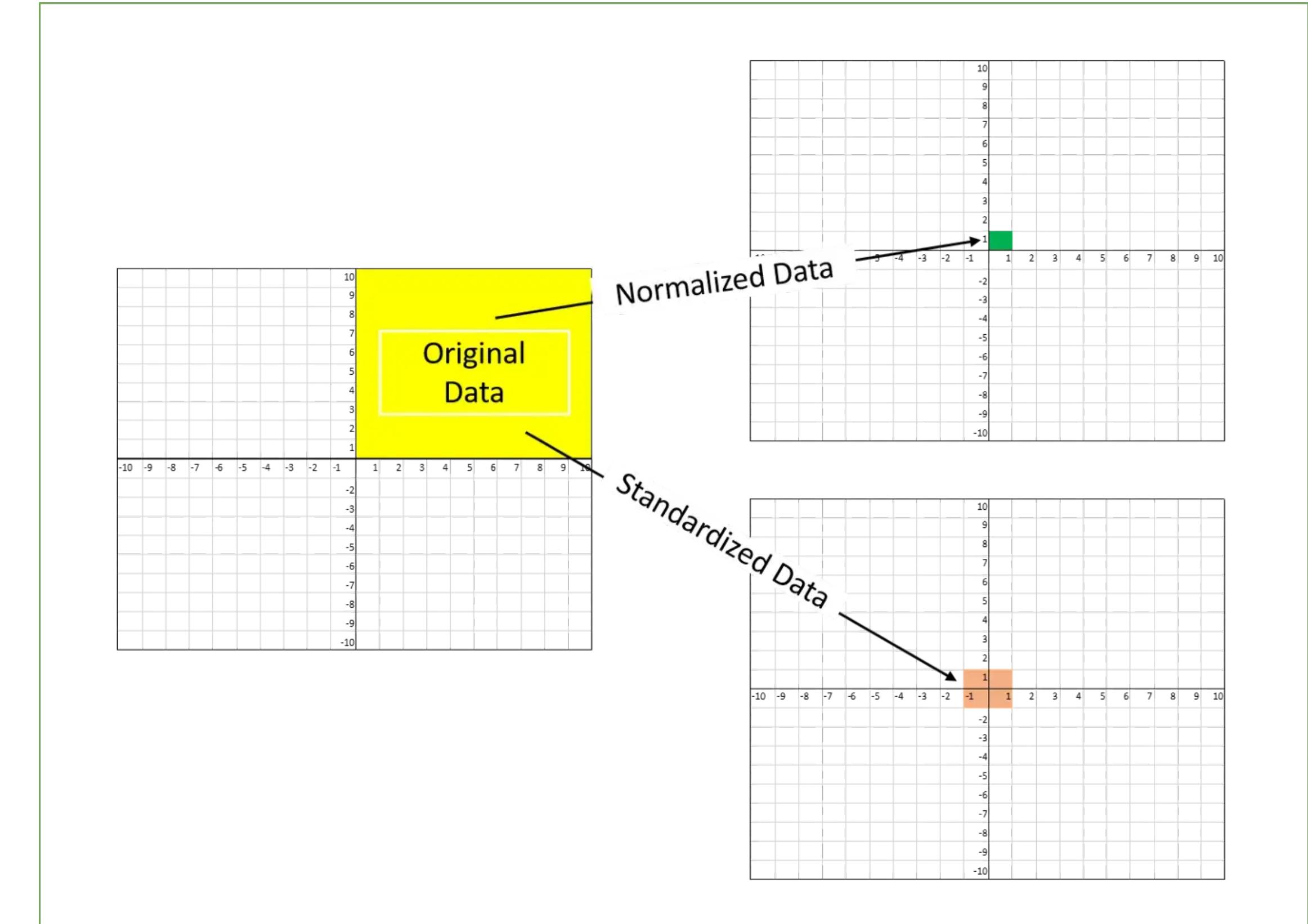
**La normalización** se utiliza cuando queremos acotar nuestros valores entre dos números, típicamente, entre  $[0, 1]$  o  $[-1, 1]$

$$x' = \frac{x - \bar{x}}{\max(x) - \min(x)}$$

**La estandarización** transforma los datos para que tengan una media 0 y una varianza de 1

$$x' = \frac{x - \bar{x}}{\sigma}$$

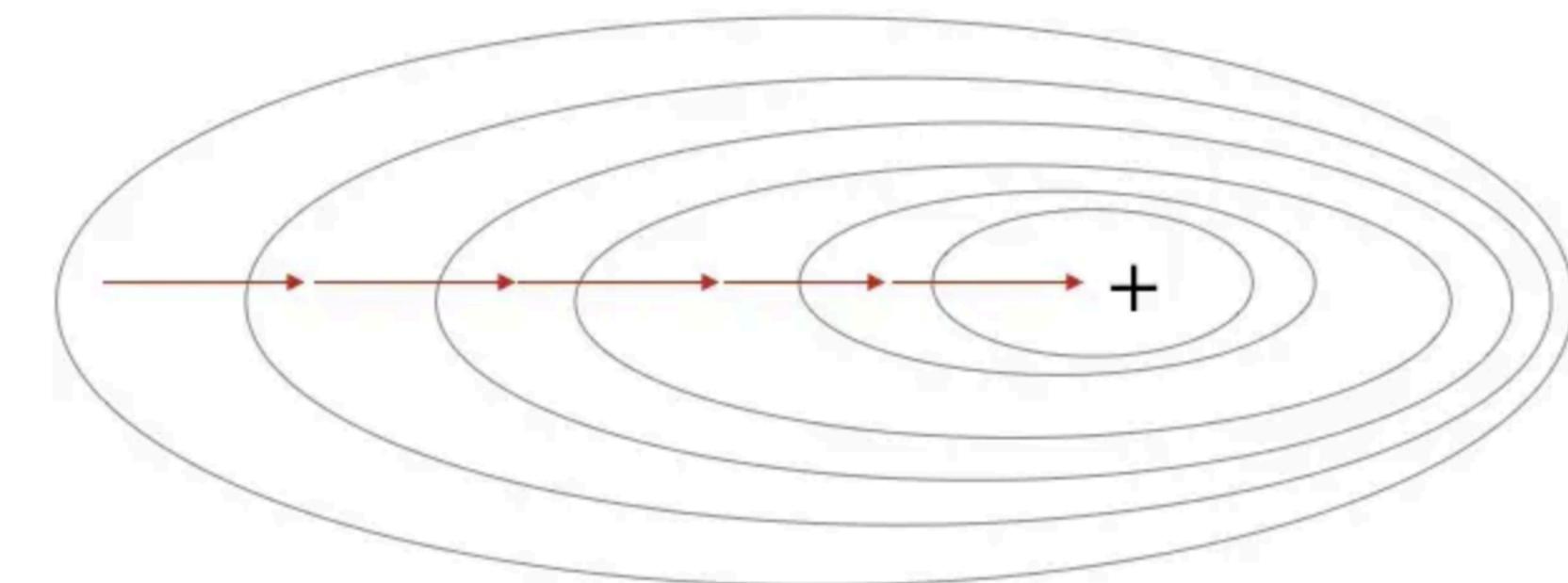
\*Si el modelo usa la distribución de los datos como una característica usar normalización



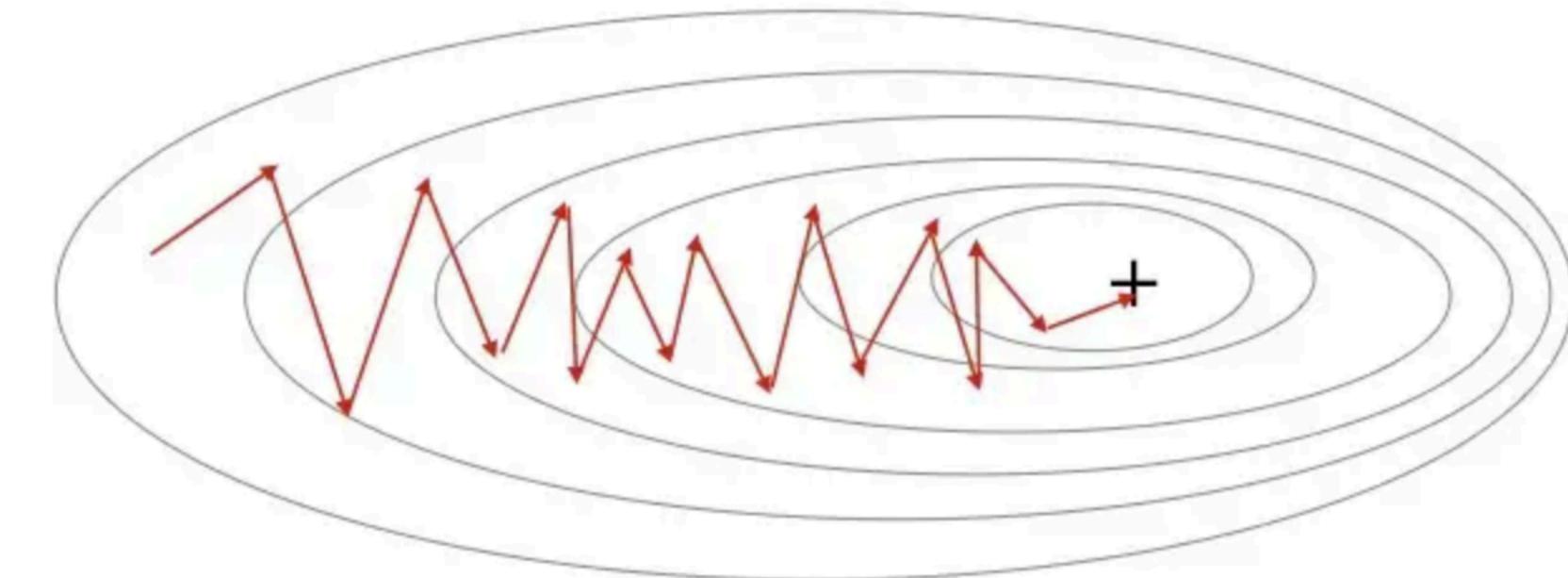
# Stochastic Gradient Descent

- Casi todo el aprendizaje profundo se basa en un algoritmo muy importante: el descenso de gradiente estocástico (SGD).
- Para una buena generalización se necesitan grandes conjuntos de entrenamiento, pero también son más costosos desde el punto de vista informático.
- Tenemos que mirar más de cerca la cantidad de cálculo que hacemos para cada iteración del algoritmo en GD.
- Tomar una muestra aleatoria de observaciones en cada iteración (Mini-batch o Batch)

Gradient Descent



Stochastic Gradient Descent



# Ejercicio

## Implementar:

- A. Cargar datos del dataset (Area vs Precios de casas)
- B. Implementar un modelo de regresión lineal
- C. Entrenar el modelo usando GD
- D. Graficar dataset y línea de predicción

