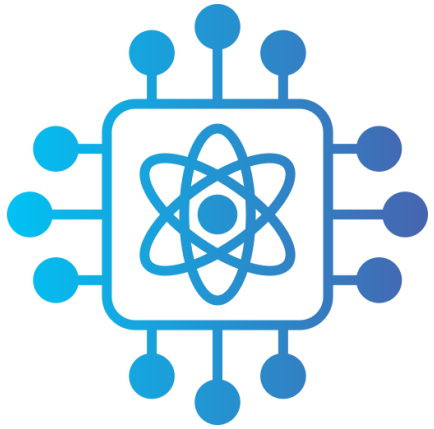


Del Perceptrón al Qubit

Introducción a las Redes Neuronales Artificiales Clásicas y Cuánticas

CIIeC
Centro Interdisciplinario de Investigación y Enseñanza de la Ciencia

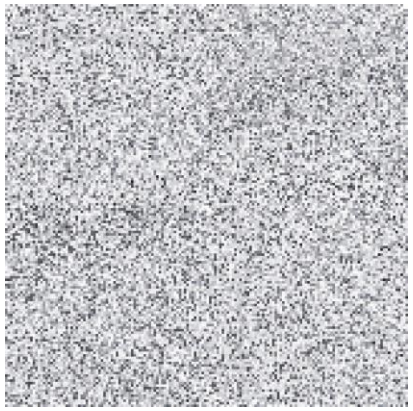
BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA · 1578 ·

DELFÍN
PROGRAMA INSTITUCIONAL PARA EL FORTALECIMIENTO DE LA INVESTIGACIÓN Y EL POSGRADO DEL PACÍFICO
"Por una cultura científica"

¿Qué son las redes neuronales artificiales?

Generación de imágenes

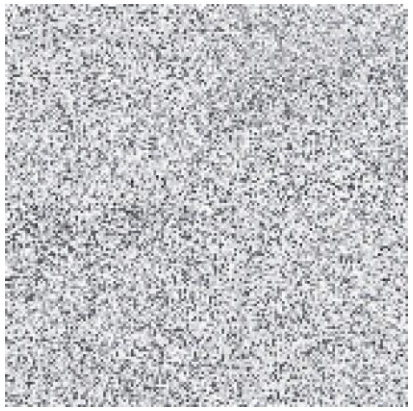
Características (entrada)



¿Qué son las redes neuronales artificiales?

Generación de imágenes

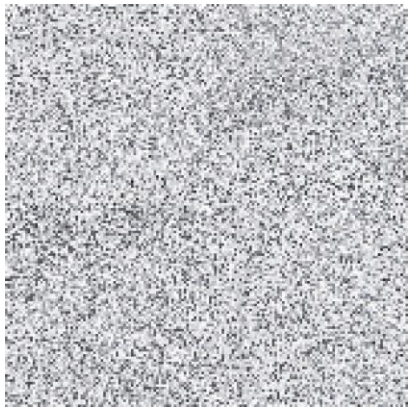
Características (entrada)



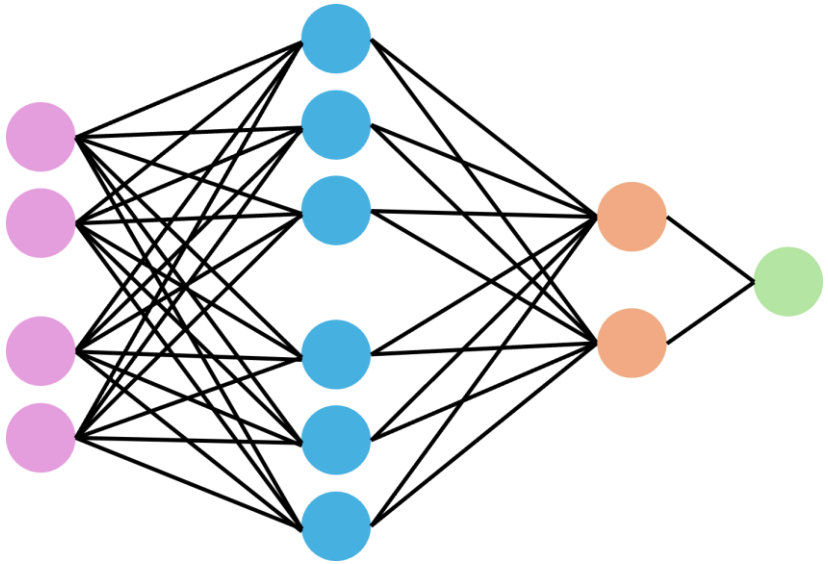
¿Qué son las redes neuronales artificiales?

Generación de imágenes

Características (entrada)



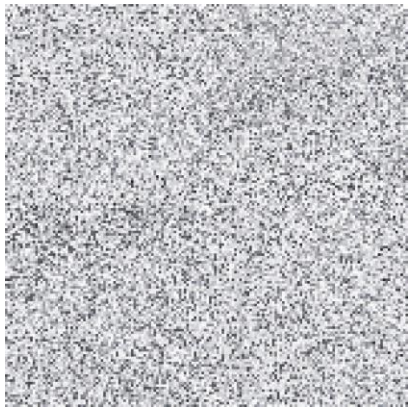
Red neuronal



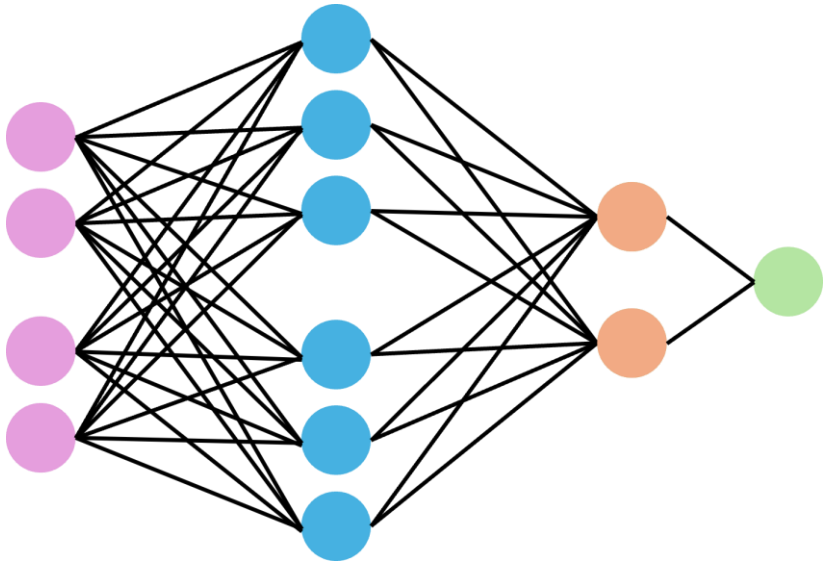
¿Qué son las redes neuronales artificiales?

Generación de imágenes

Características (entrada)



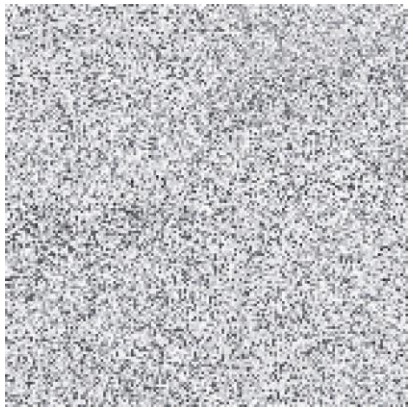
Red neuronal



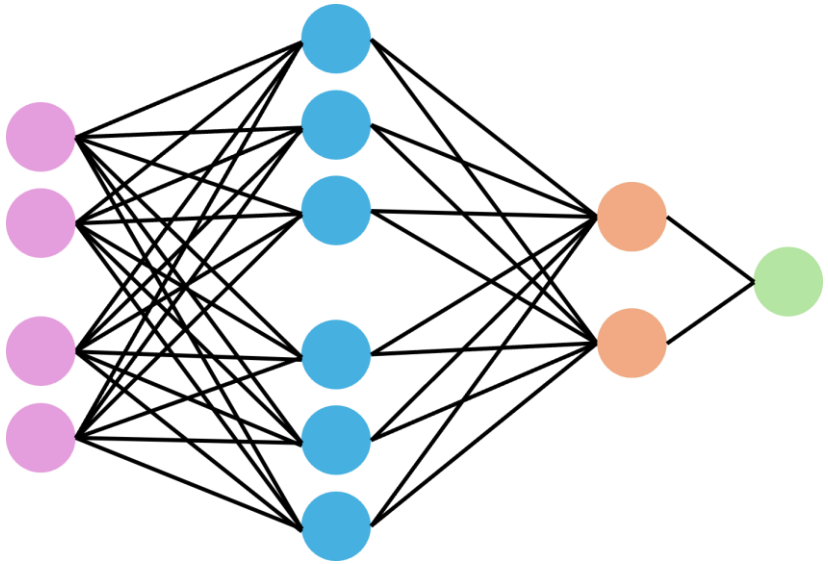
¿Qué son las redes neuronales artificiales?

Generación de imágenes

Características (entrada)



Red neuronal



Salida

¿Qué son las redes neuronales artificiales?

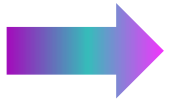
Reconocimiento de imágenes

Características (entrada)

¿Qué son las redes neuronales artificiales?

Reconocimiento de imágenes

Características (entrada)

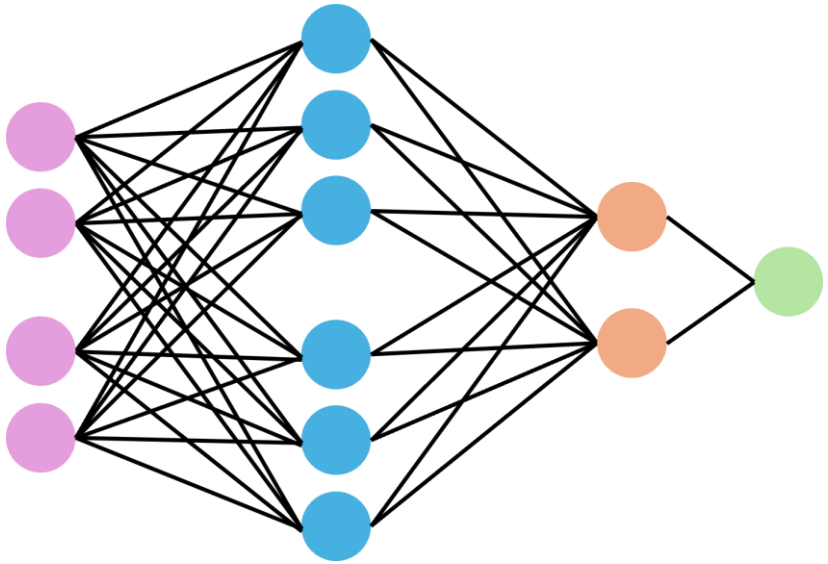


¿Qué son las redes neuronales artificiales?

Reconocimiento de imágenes

Características (entrada)

Red neuronal

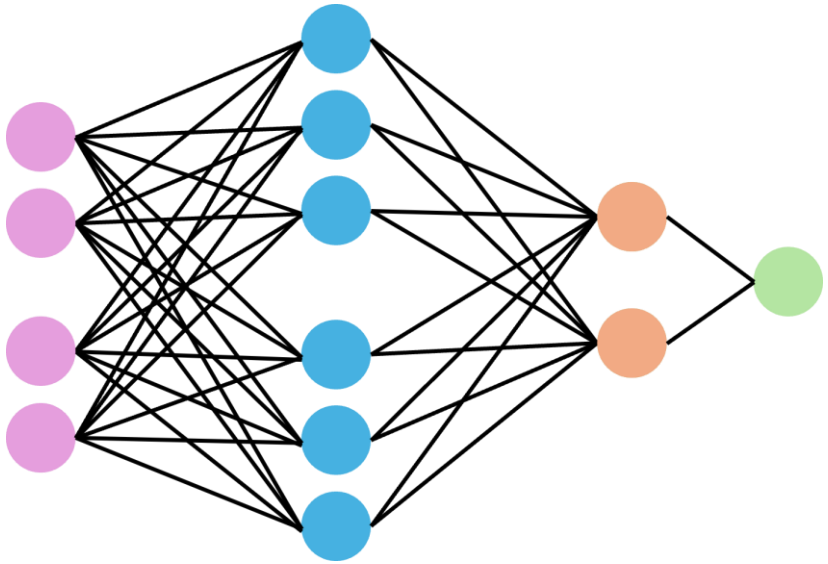


¿Qué son las redes neuronales artificiales?

Reconocimiento de imágenes

Características (entrada)

Red neuronal

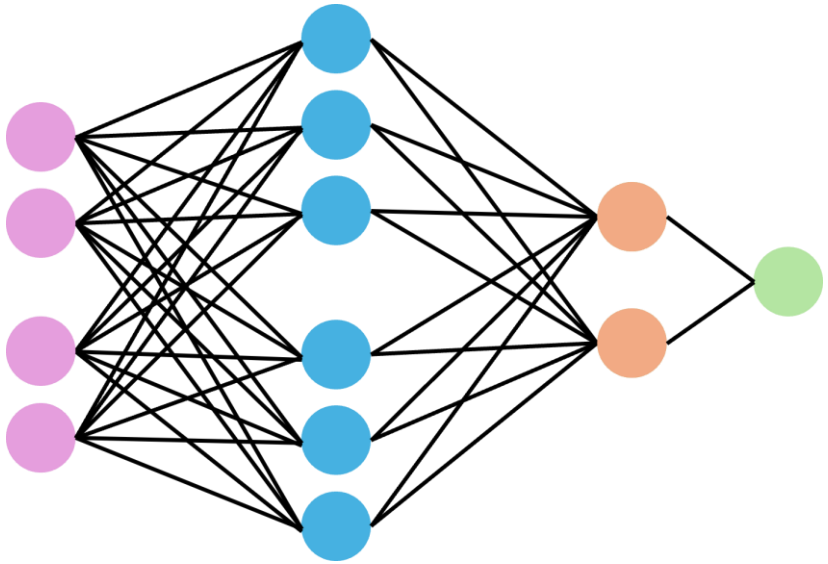


¿Qué son las redes neuronales artificiales?

Reconocimiento de imágenes

Características (entrada)

Red neuronal



Salida

GATO

¿Qué son las redes neuronales artificiales?

Predicción

Características
(entrada)

¿Qué son las redes neuronales artificiales?

Predicción

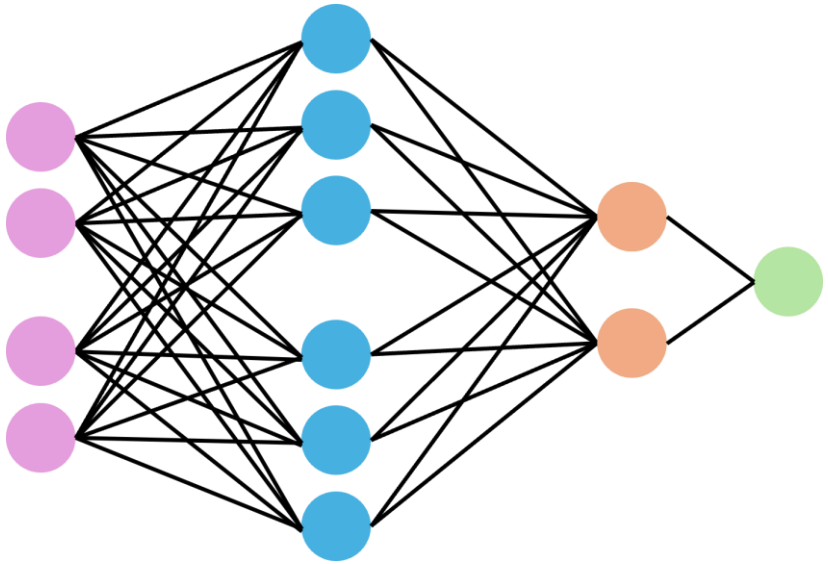
Características
(entrada)

¿Qué son las redes neuronales artificiales?

Predicción

Características
(entrada)

Red neuronal

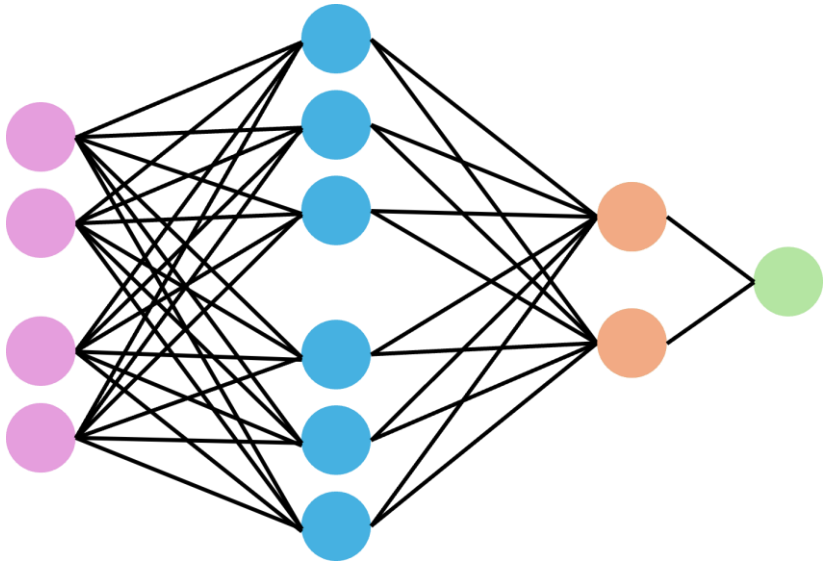


¿Qué son las redes neuronales artificiales?

Predicción

Características
(entrada)

Red neuronal

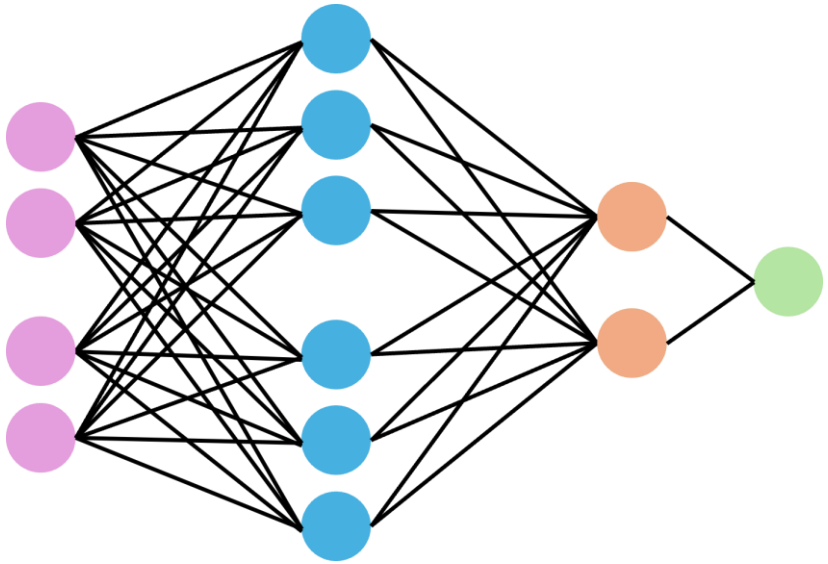


¿Qué son las redes neuronales artificiales?

Predicción

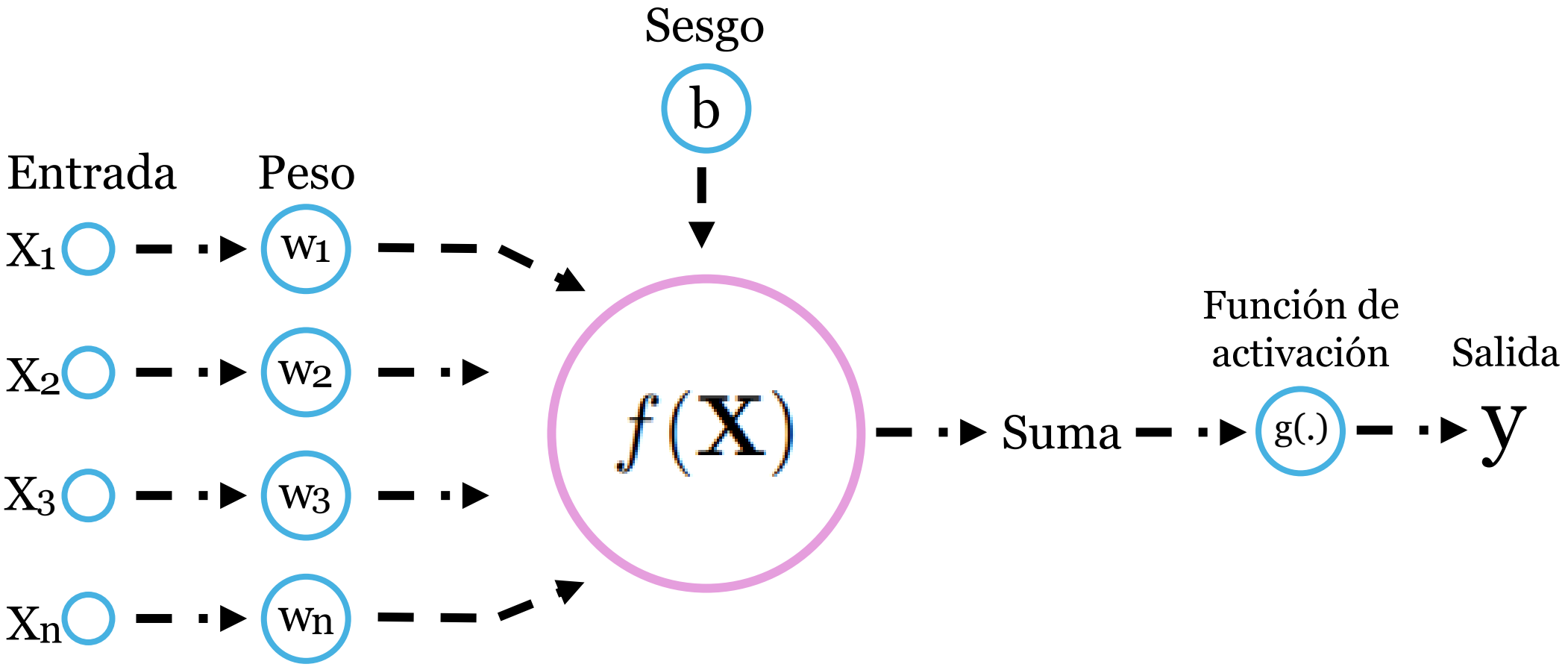
Características (entrada)

Red neuronal

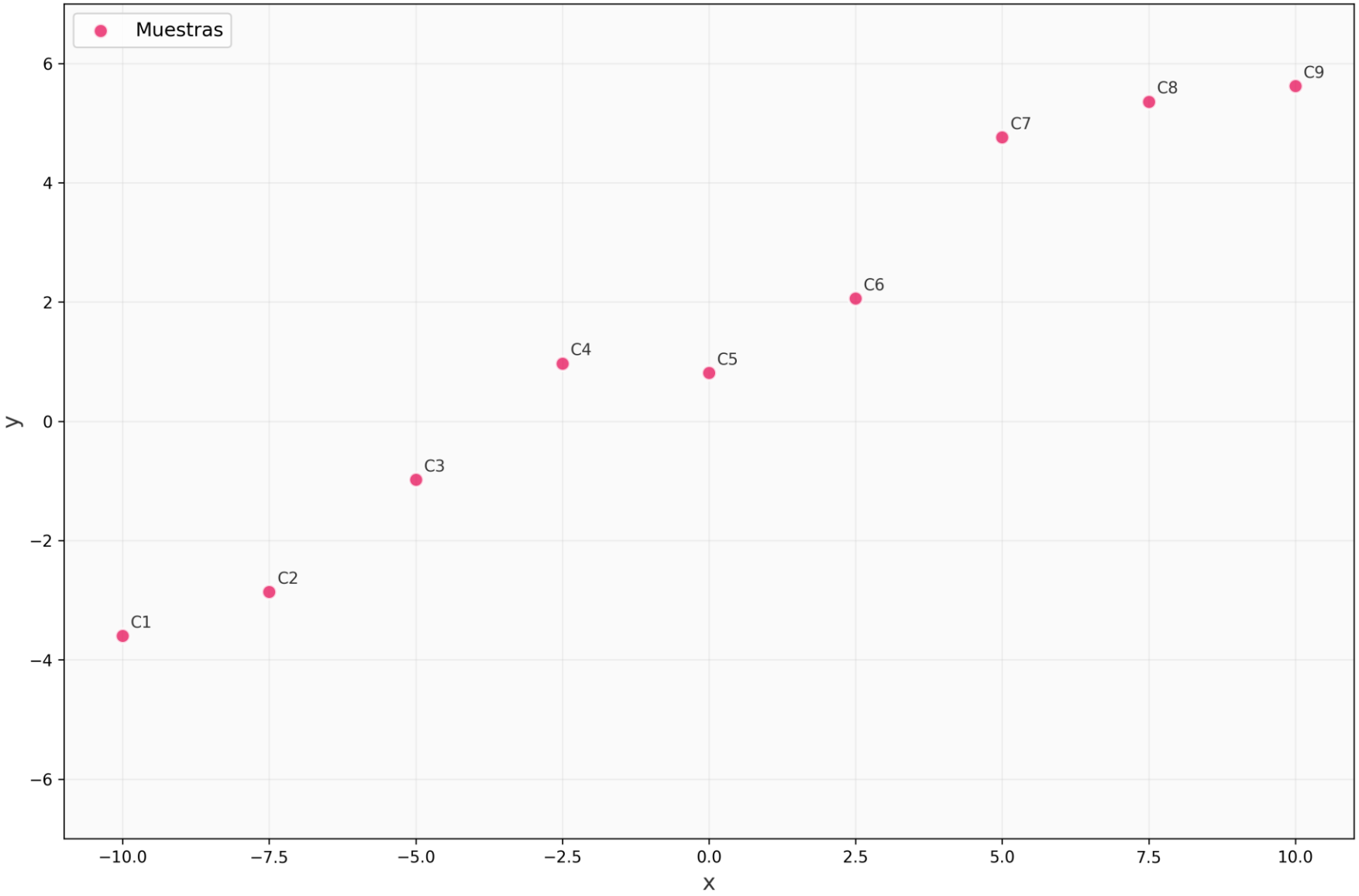


Salida

¿Qué es una neurona?



Regresión lineal

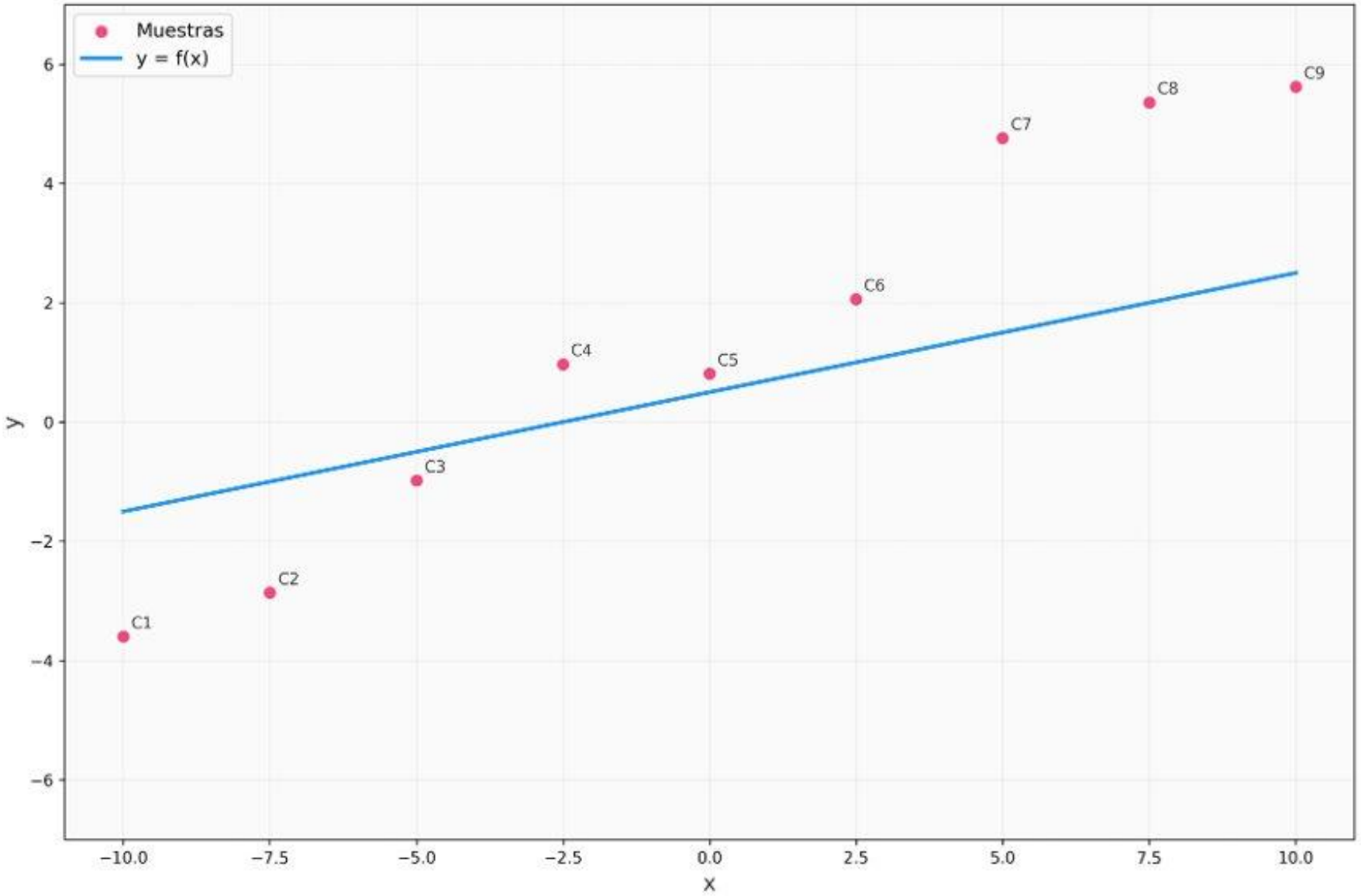


Notación

$$
\begin{aligned}
x &= \text{Característica} \\
y &= \text{Objetivo} \\
m &= \text{Número de muestras} \\
(x, y) &= \text{Punto individual} \\
(x^{(i)}, y^{(i)}) &= i\text{-ésimo punto}
\end{aligned}
$$

X	y
-10.0	-3.60
-7.5	-2.86
-5.0	-0.98
-2.5	0.97
0.0	0.81
2.5	2.06
5.0	4.76
7.5	5.36
10.0	5.62

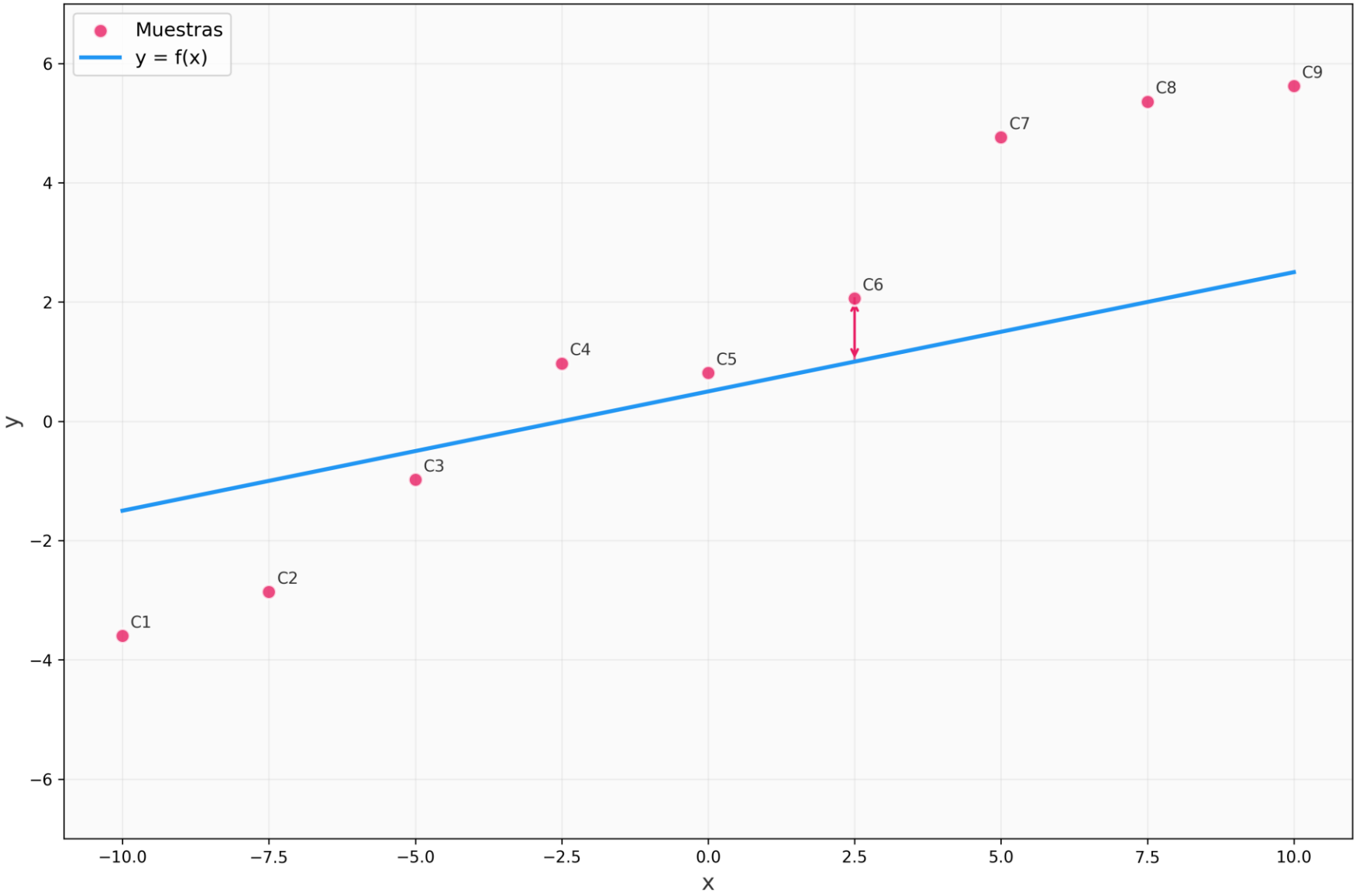
Regresión lineal



Proponemos una función conocida como hipótesis para modelar la relación entre $\mathbf{X}$ e $\mathbf{y}$

$$y = f(x) = 0.2x + 0.5$$

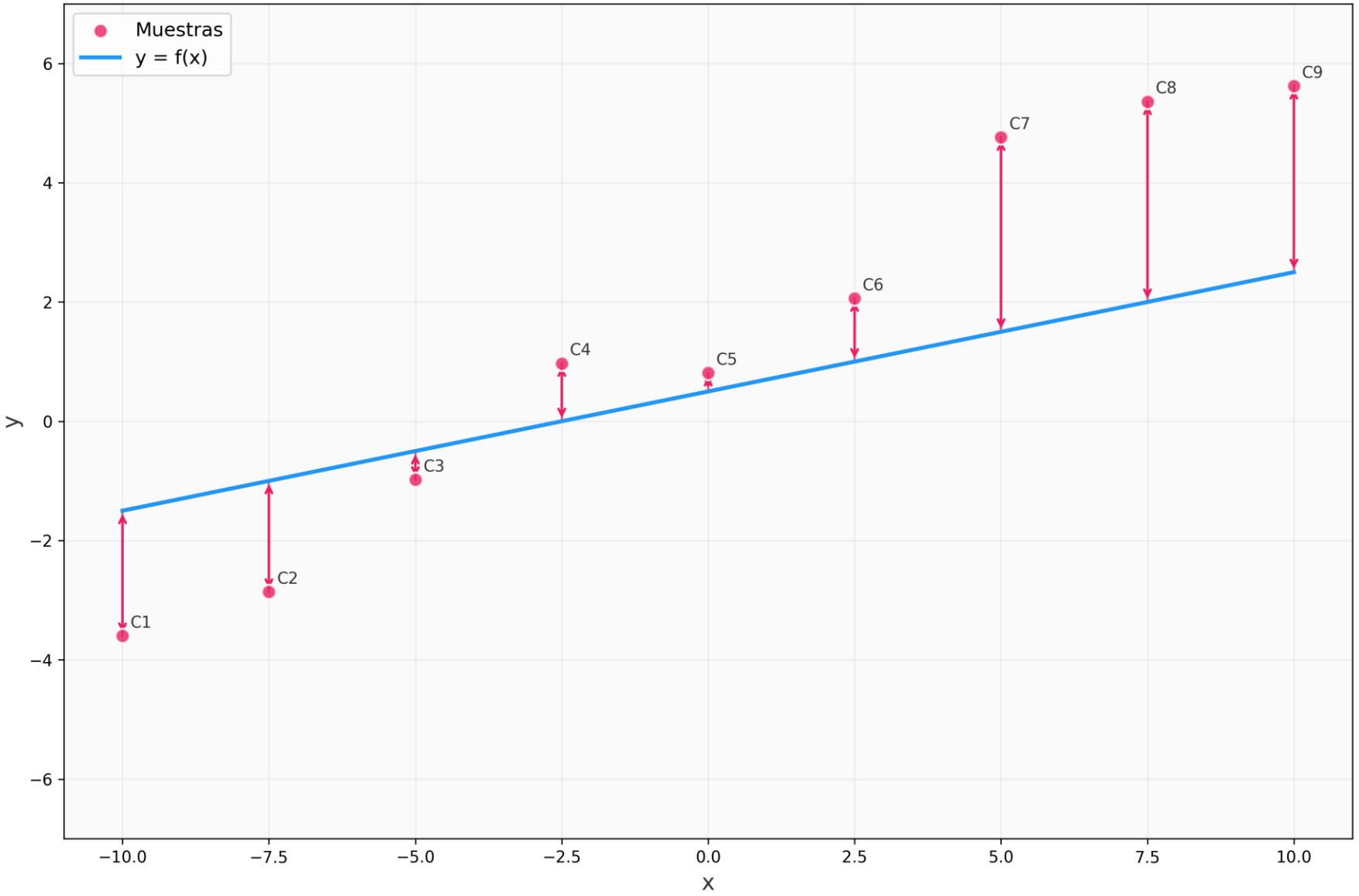
Regresión lineal



Para evaluar qué tan bien se ajusta nuestra hipótesis, calculamos el error entre el valor real y la predicción como:

$$(y^{(i)} - f(x^{(i)}))^2$$

Regresión lineal



El error total estará dado por el promedio de los errores individuales

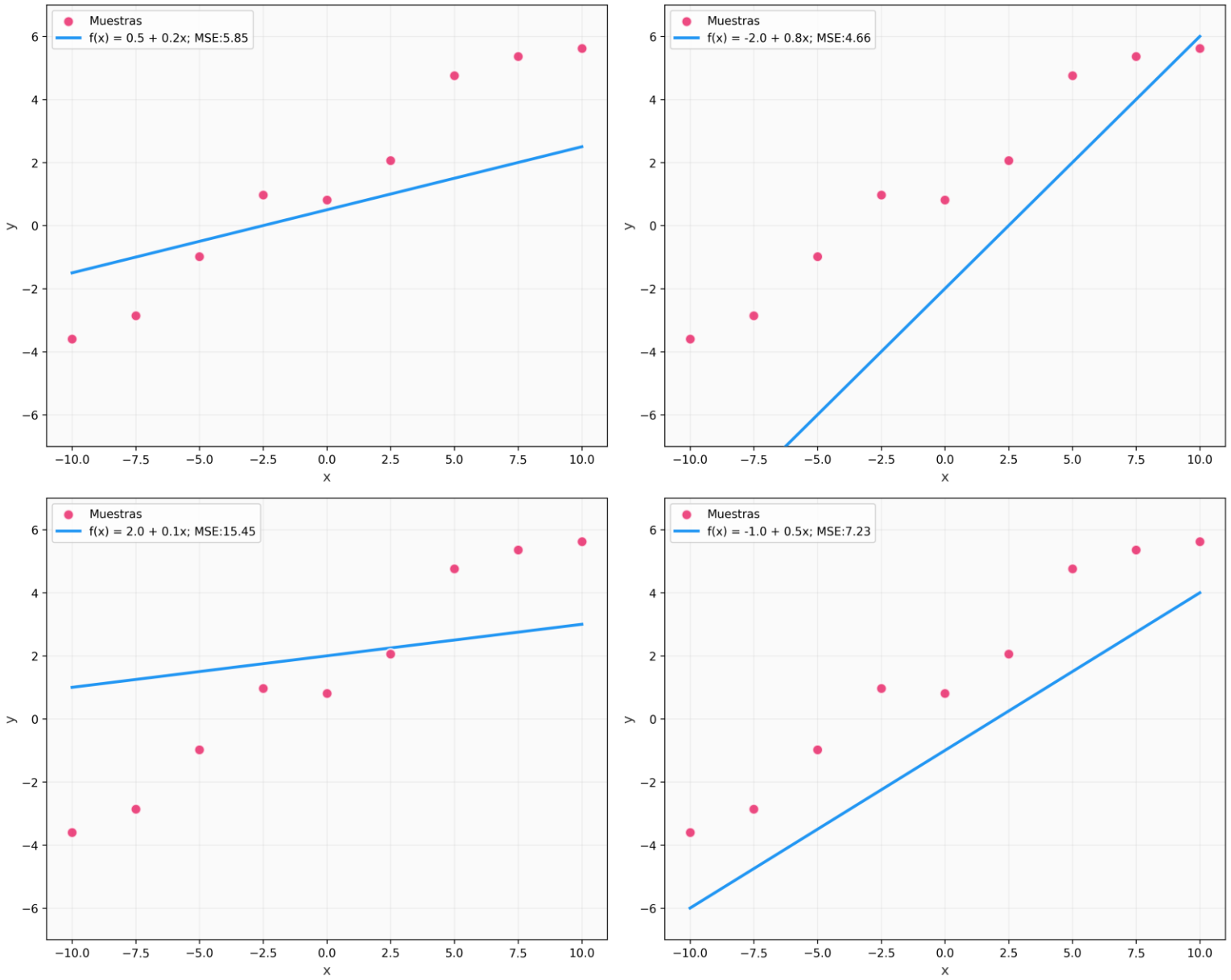
$$\frac{1}{m}\sum_{i=0}^{m}(y^{(i)} - f(x^{(i)}))^2$$

Error Cuadrático Medio

$$MSE = \frac{1}{m}\sum_{i=0}^{m}(y^{(i)} - f(x^{(i)}))^2$$

- Toma únicamente errores positivos
- Es una función diferenciable y convexa
- El ajuste perfecto se alcanza cuando MSE=0

Gradiente descendente

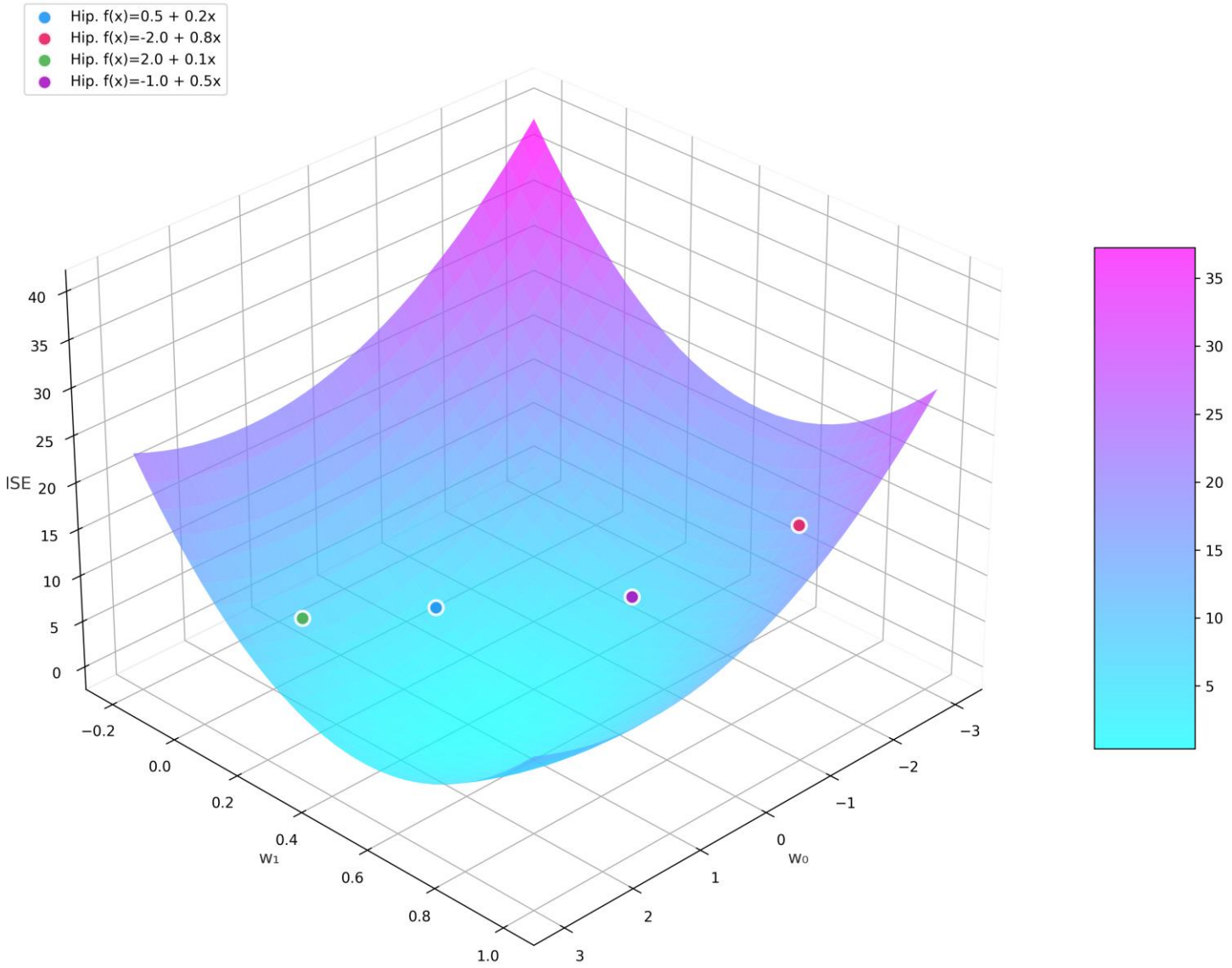


Para una función hipótesis de la forma

$$f(x) = w_0 + w_1 x$$

¿Cómo encontramos los parámetros que minimizan la función de costo?

Gradiente descendente

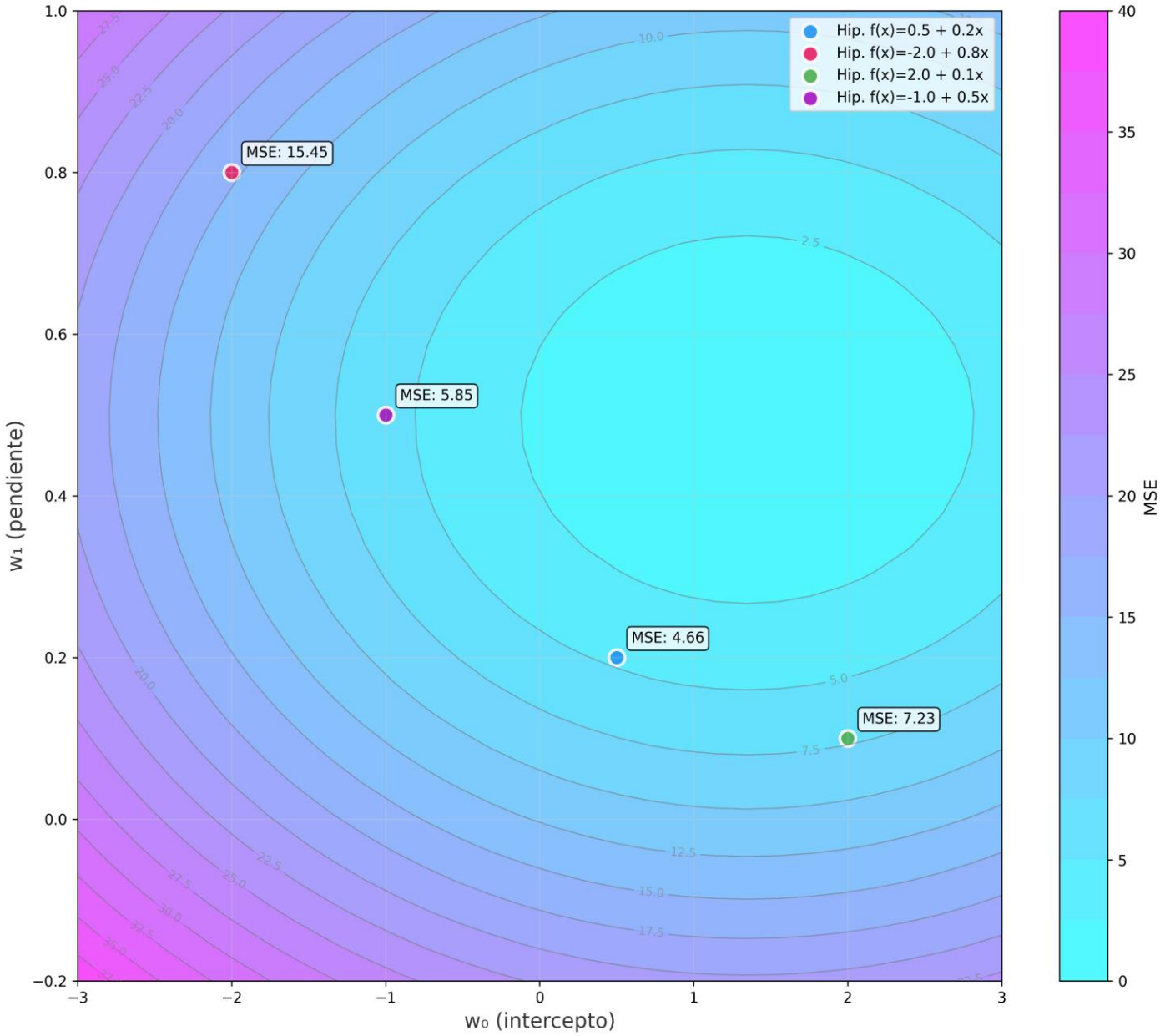


La gráfica de la función de costo respecto a los parámetros se asemeja a un valle cuya altura representa el valor del costo para las coordenadas

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

Objetivo: Encontrar el punto más bajo del valle

Gradiente descendente



Otra forma de visualizar la función de costo es mediante curvas de nivel

¿De qué manera podemos encontrar los parámetros óptimos automáticamente?

Gradiente descendente

Problema: Optimización de una función

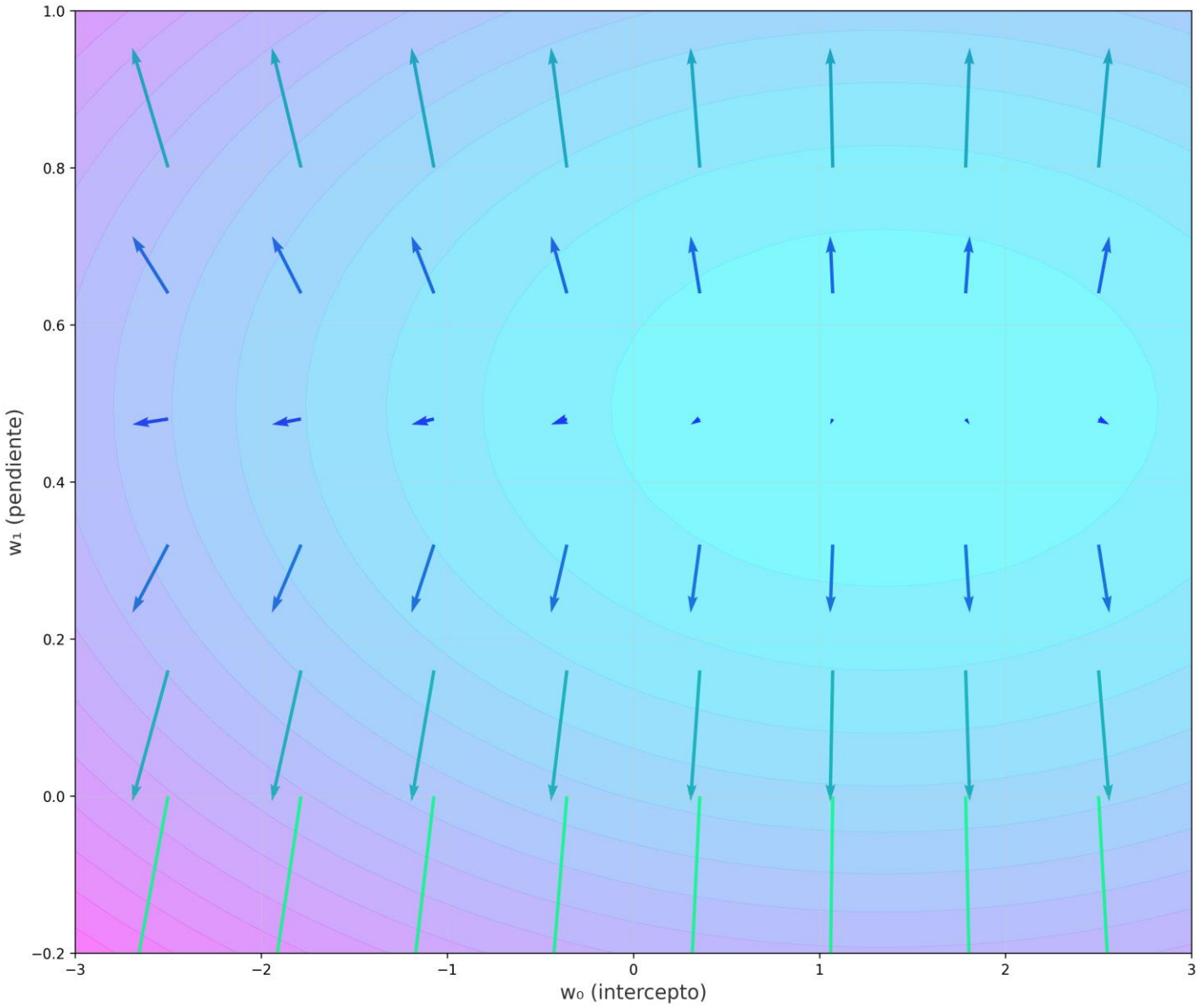
$$J(w_0, w_1) = \frac{1}{m}\sum_{i=0}^{m}(y^{(i)} - f(x^{(i)}))^2$$

Para minimizar $J(w_0, w_1)$, necesitamos calcular las derivadas parciales

$$\begin{aligned}\frac{\partial J}{\partial w_0} &= \frac{\partial}{\partial w_0}\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - w_0 - w_1 x^{(i)}\right)^2\right] \\ &= \frac{1}{m}\sum_{i=1}^{m} 2\left(y^{(i)} - w_0 - w_1 x^{(i)}\right)\cdot(-1) \\ &= \frac{2}{m}\sum_{i=1}^{m}\left(f(x^{(i)}) - y^{(i)}\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial J}{\partial w_1} &= \frac{\partial}{\partial w_1}\left[\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - w_0 - w_1 x^{(i)}\right)^2\right] \\ &= \frac{1}{m}\sum_{i=1}^{m} 2\left(y^{(i)} - w_0 - w_1 x^{(i)}\right)\cdot(-x^{(i)}) \\ &= \frac{2}{m}\sum_{i=1}^{m}\left(f(x^{(i)}) - y^{(i)}\right)x^{(i)}\end{aligned}$$

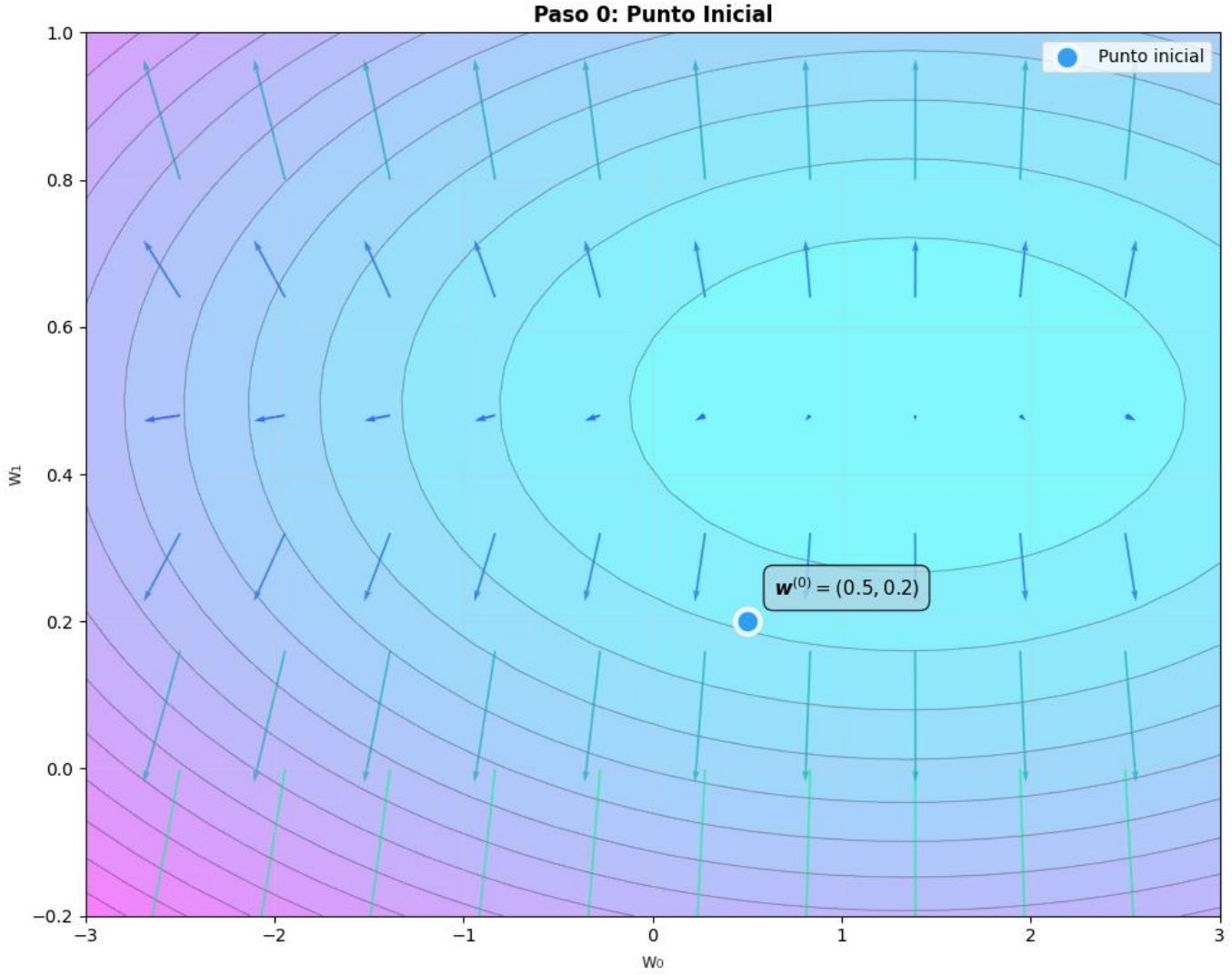
Gradiente descendente

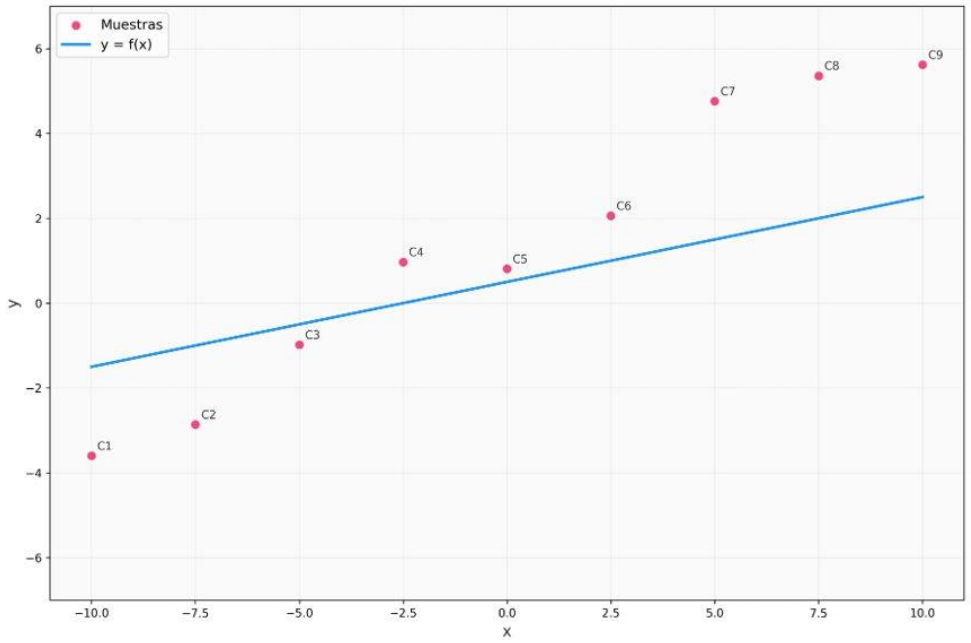


Usando las derivadas obtenemos el vector gradiente, que nos indica la dirección de mayor crecimiento de la función

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial w_0} \\ \frac{\partial J}{\partial w_1} \end{bmatrix}$$

Gradiente descendente

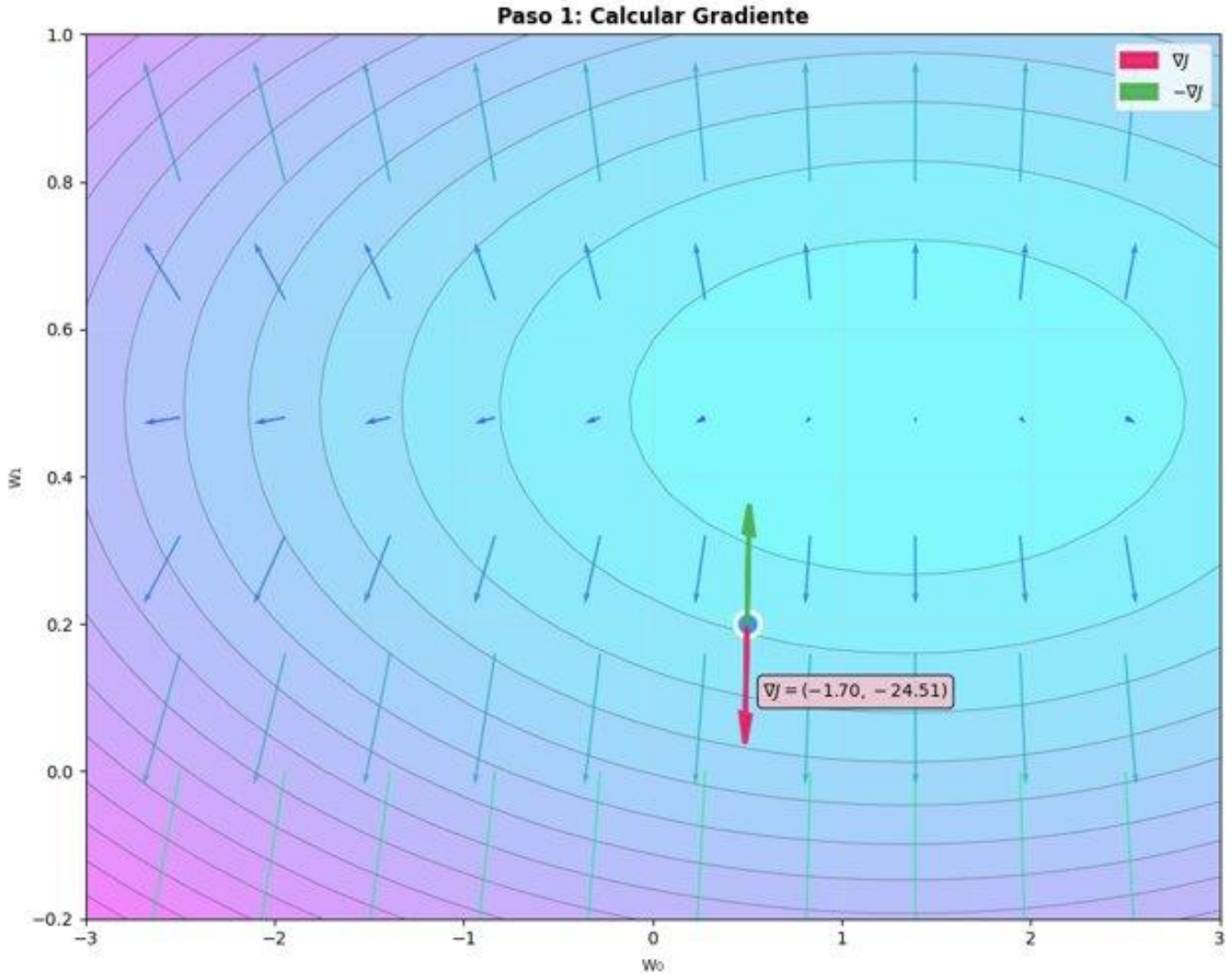




$$f(x) = 0.5 + 0.2x$$

$$\mathbf{w}^{(0)} = \begin{bmatrix} 0.5 \\ 0.2 \end{bmatrix}$$

Gradiente descendente



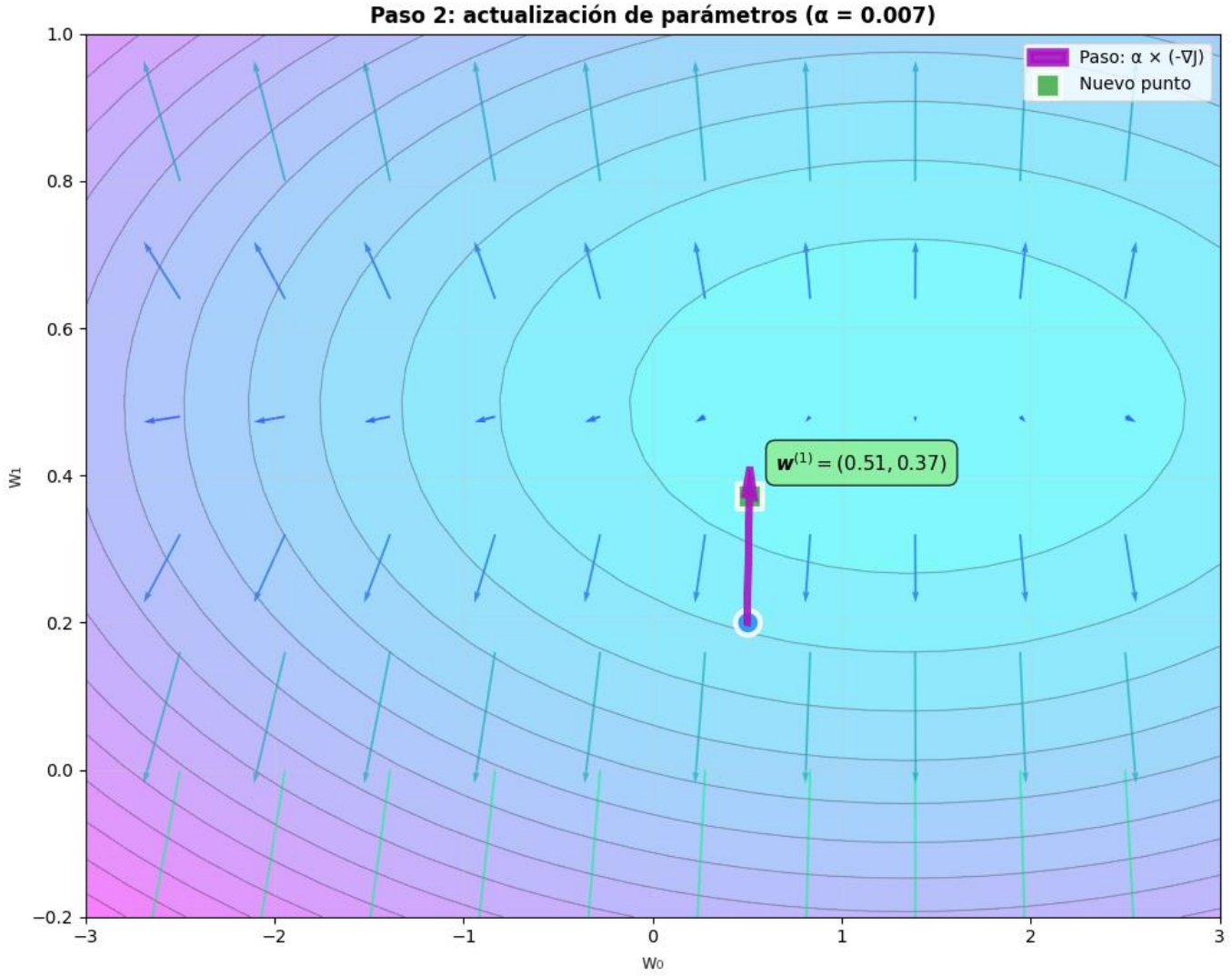
$$\left.\frac{\partial J}{\partial w_0}\right|_{(0.5,0.2)} = \frac{2}{m}\sum_{i=1}^{m}(f(x^{(i)}) - y^{(i)})$$

$$\left.\frac{\partial J}{\partial w_1}\right|_{(0.5,0.2)} = \frac{2}{m}\sum_{i=1}^{m}(f(x^{(i)}) - y^{(i)})x^{(i)}$$

Obtenemos el vector gradiente

$$\nabla J^{(0)} = \begin{bmatrix} -1.70 \\ -24.51 \end{bmatrix}$$

Gradiente descendente

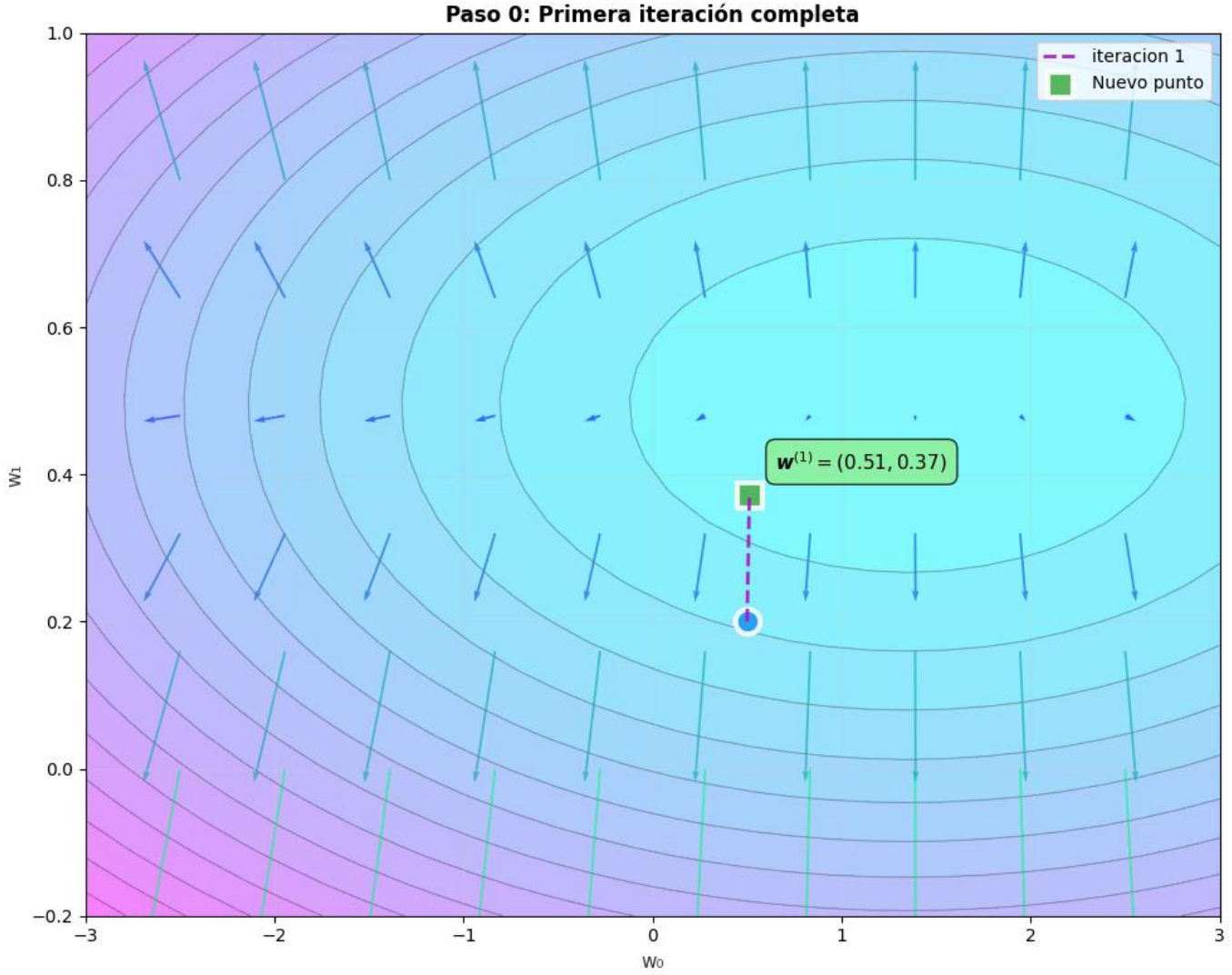


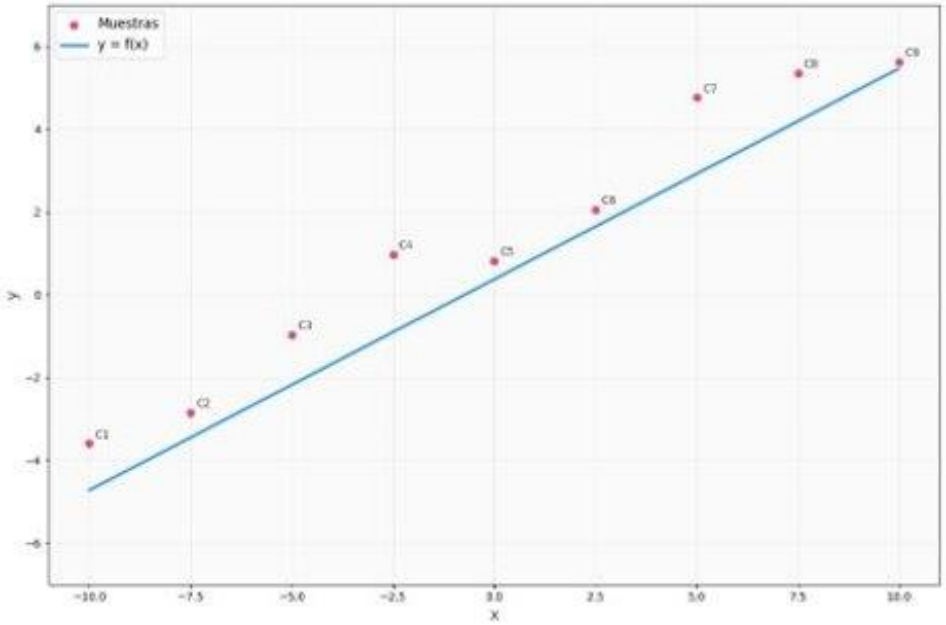
$$w_0^{(1)} = w_0^{(0)} - \alpha \frac{\partial J}{\partial w_0^{(0)}}$$

$$w_1^{(1)} = w_1^{(0)} - \alpha \frac{\partial J}{\partial w_1^{(0)}}$$

Donde es α el factor de aprendizaje, usualmente es elegido un número muy pequeño

Gradiente descendente

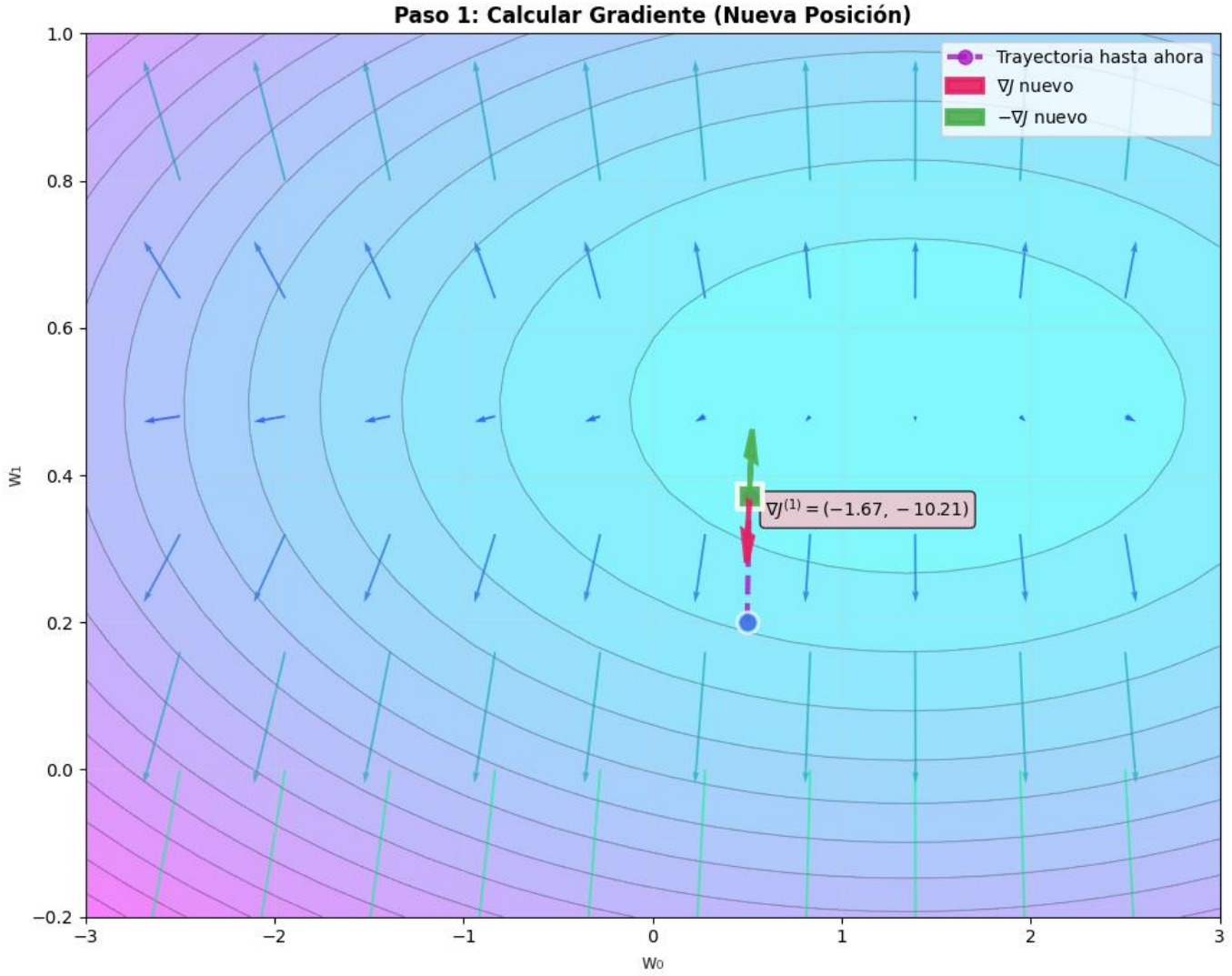




$$f(x) = 0.51 + 0.37x$$

$$\mathbf{w}^{(1)} = \begin{bmatrix} 0.51 \\ 0.37 \end{bmatrix}$$

Gradiente descendente



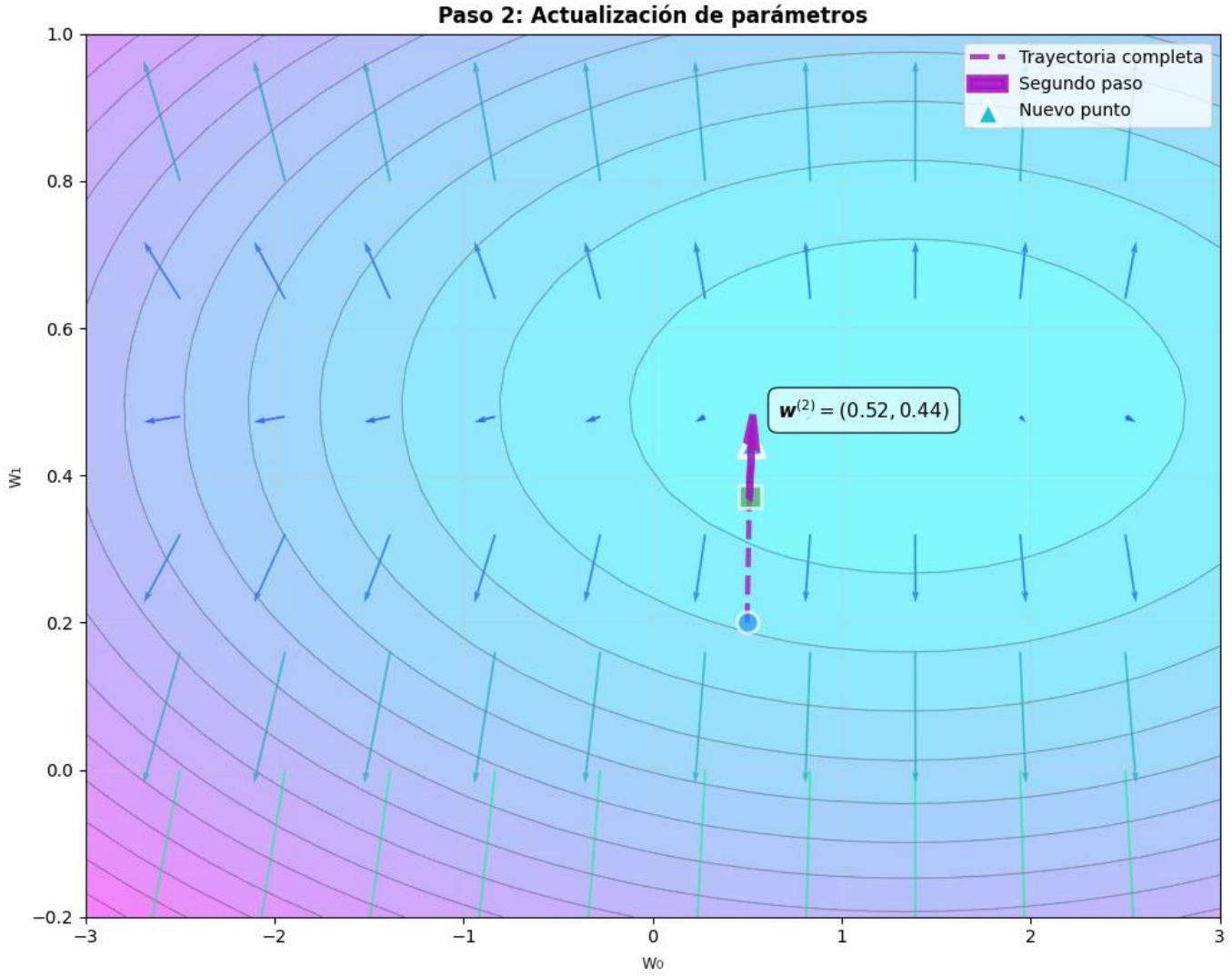
$$\left.\frac{\partial J}{\partial w_0}\right|_{(0.51,0.37)} = \frac{2}{m}\sum_{i=1}^{m}(f(x^{(i)}) - y^{(i)})$$

$$\left.\frac{\partial J}{\partial w_1}\right|_{(0.51,0.37)} = \frac{2}{m}\sum_{i=1}^{m}(f(x^{(i)}) - y^{(i)})x^{(i)}$$

Obtenemos el vector gradiente

$$\nabla J^{(1)} = \begin{bmatrix} -1.67 \\ -10.21 \end{bmatrix}$$

Gradiente descendente



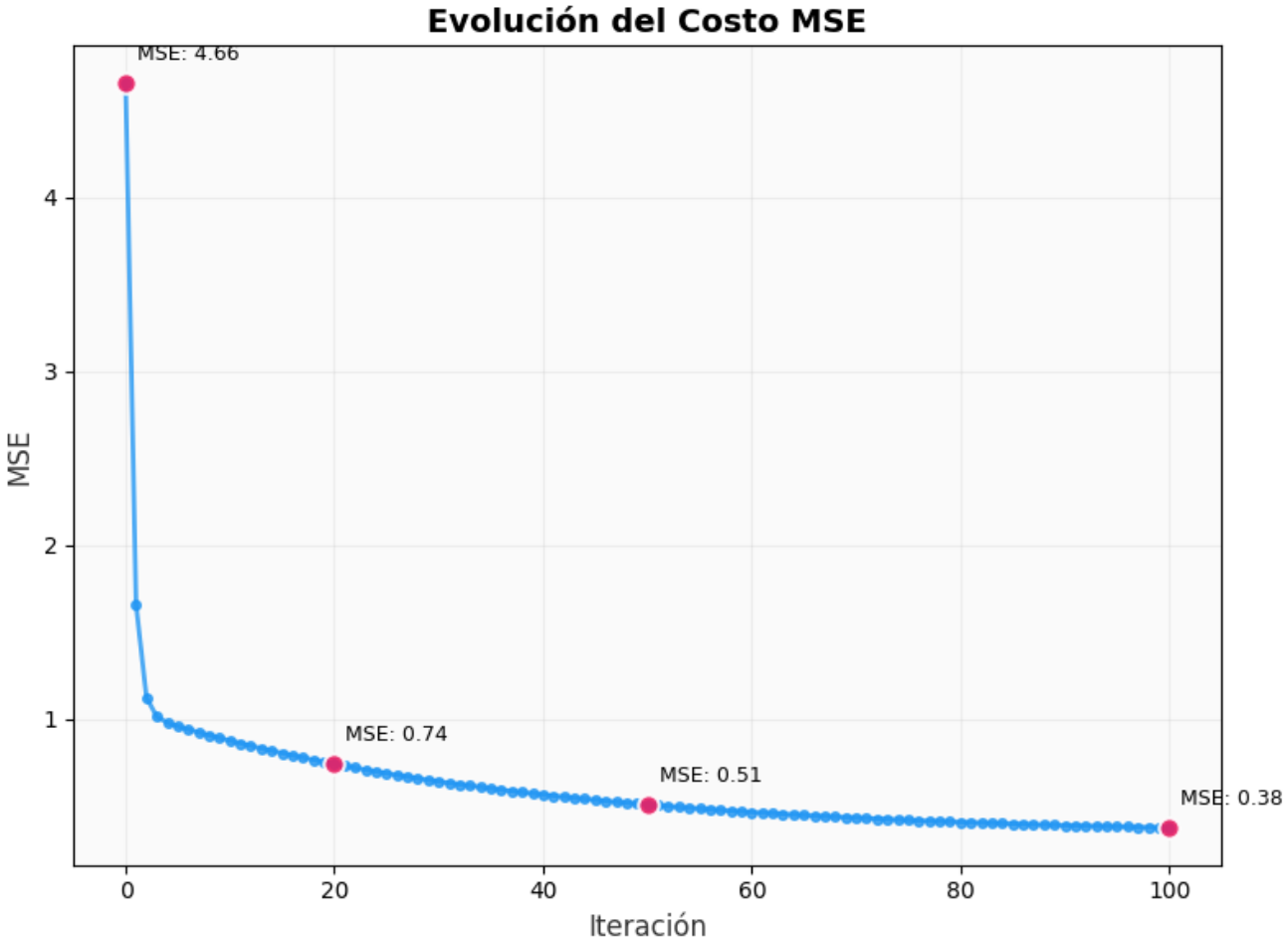
$$w_0^{(2)} = w_0^{(1)} - \alpha \frac{\partial J}{\partial w_0^{(1)}}$$

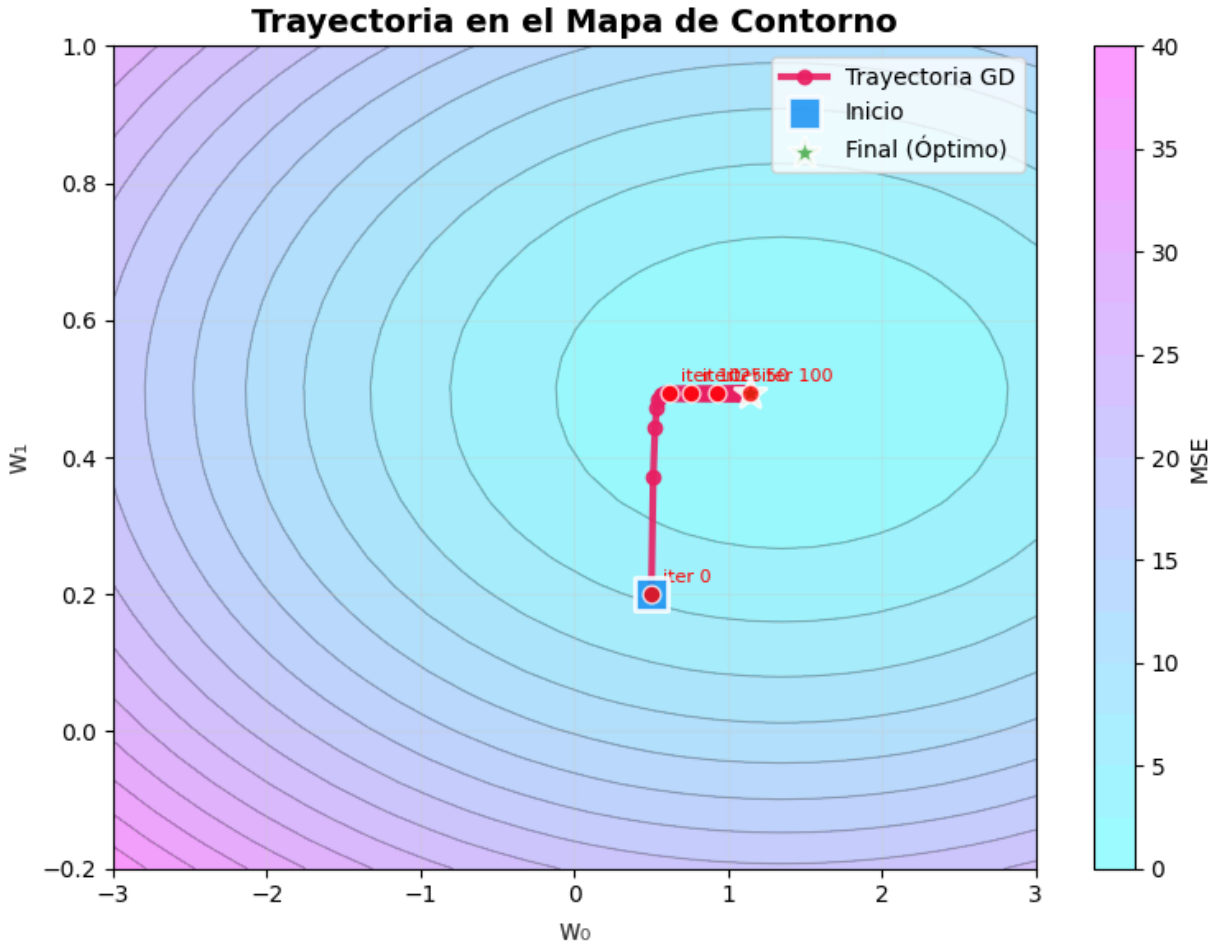
$$w_1^{(2)} = w_1^{(1)} - \alpha \frac{\partial J}{\partial w_1^{(1)}}$$

Generalización del gradiente descendente

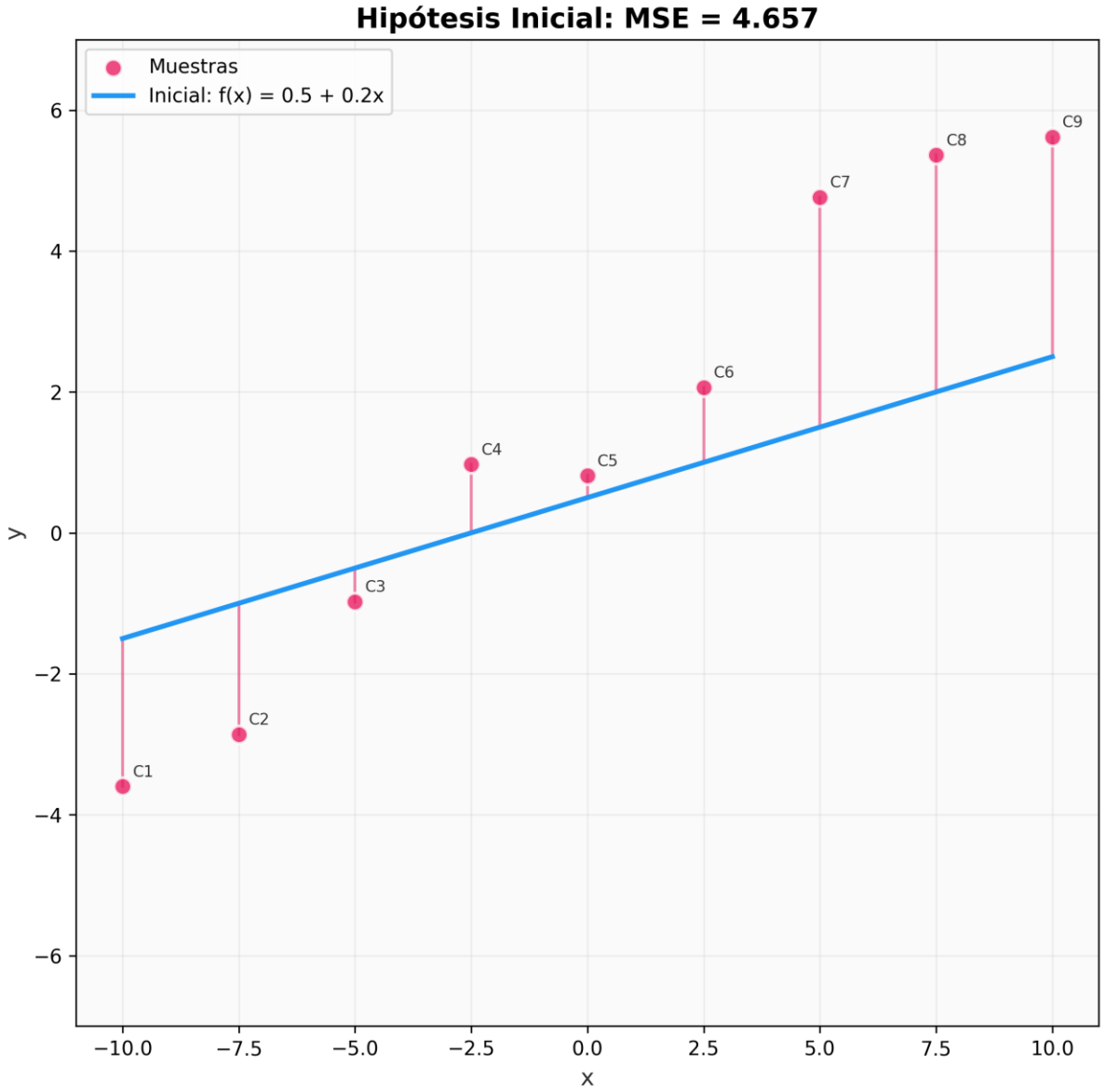
$$\mathbf{w^{(t+1)}} = \mathbf{w^{(t)}} - \alpha \nabla J^{(t)}$$

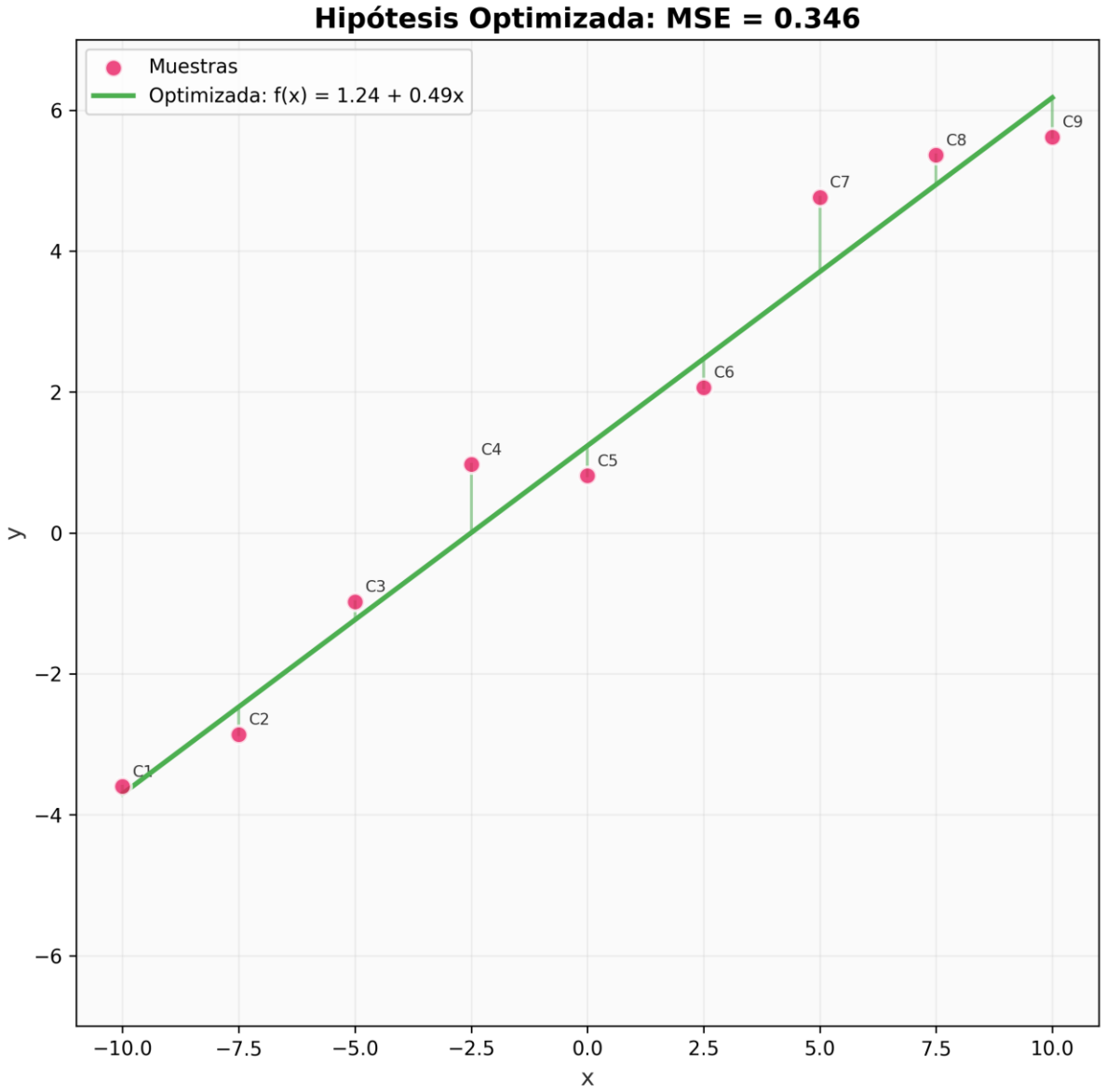
Gradiente descendente



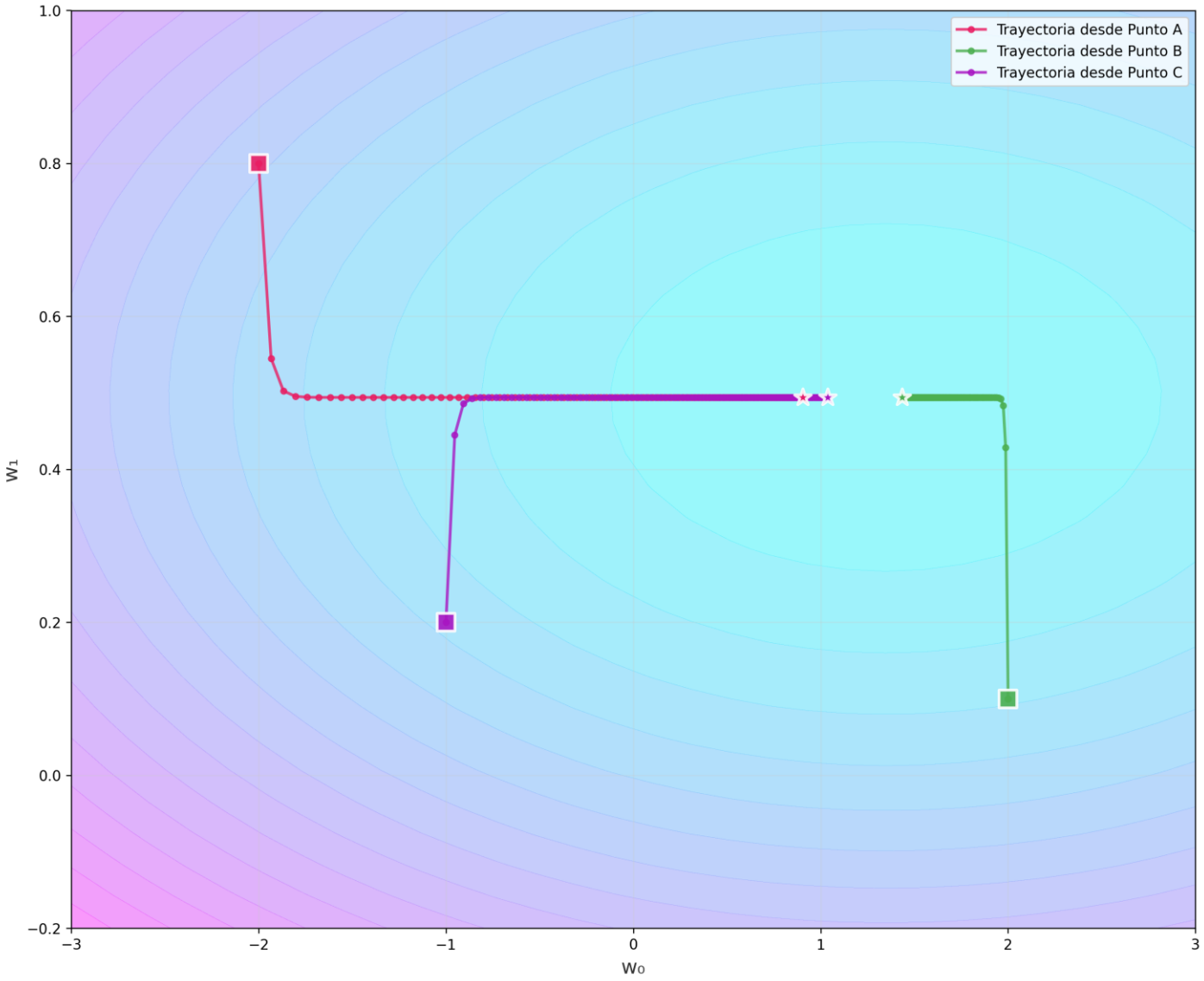


Gradiente descendente



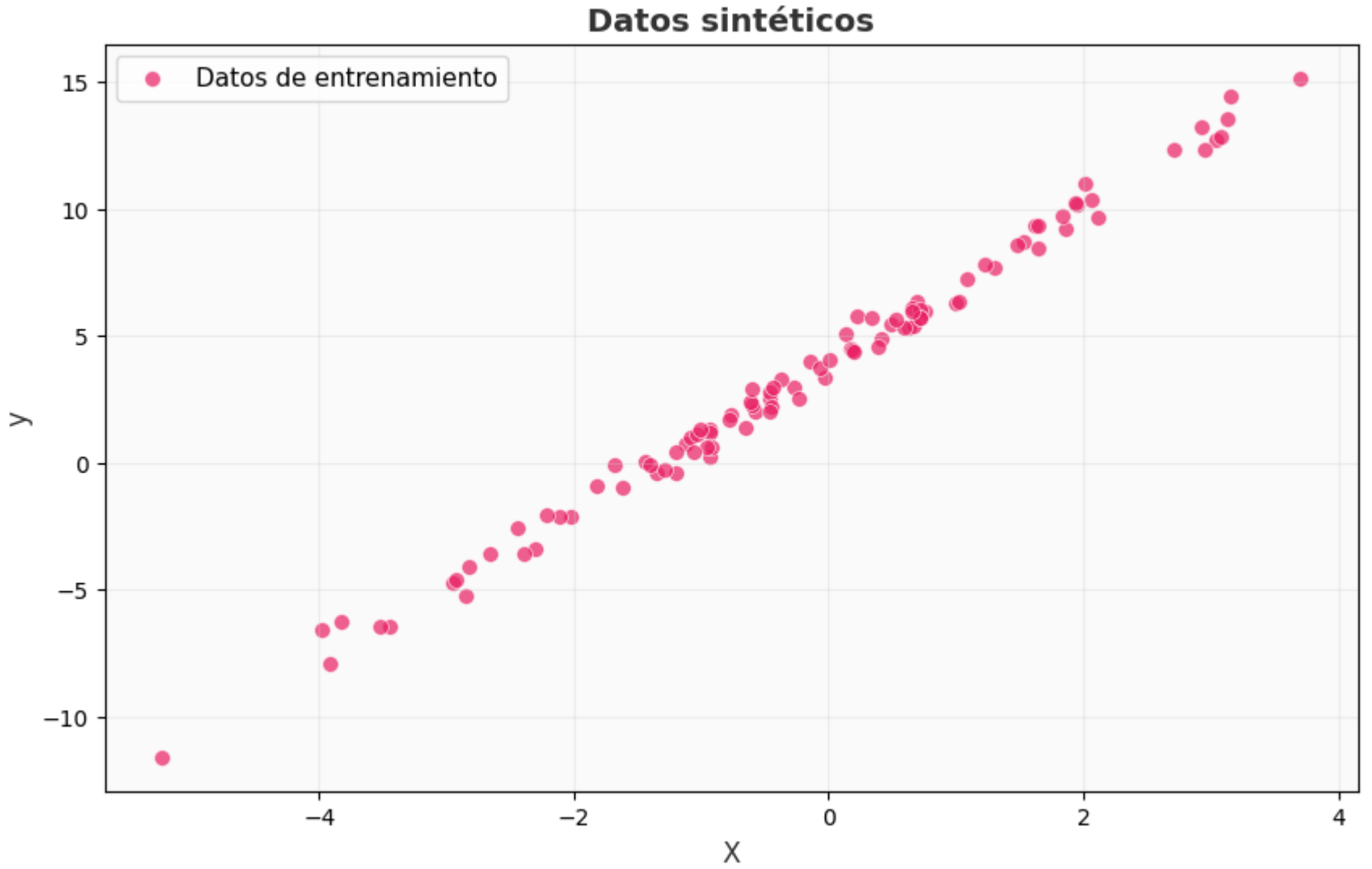


Gradiente descendente

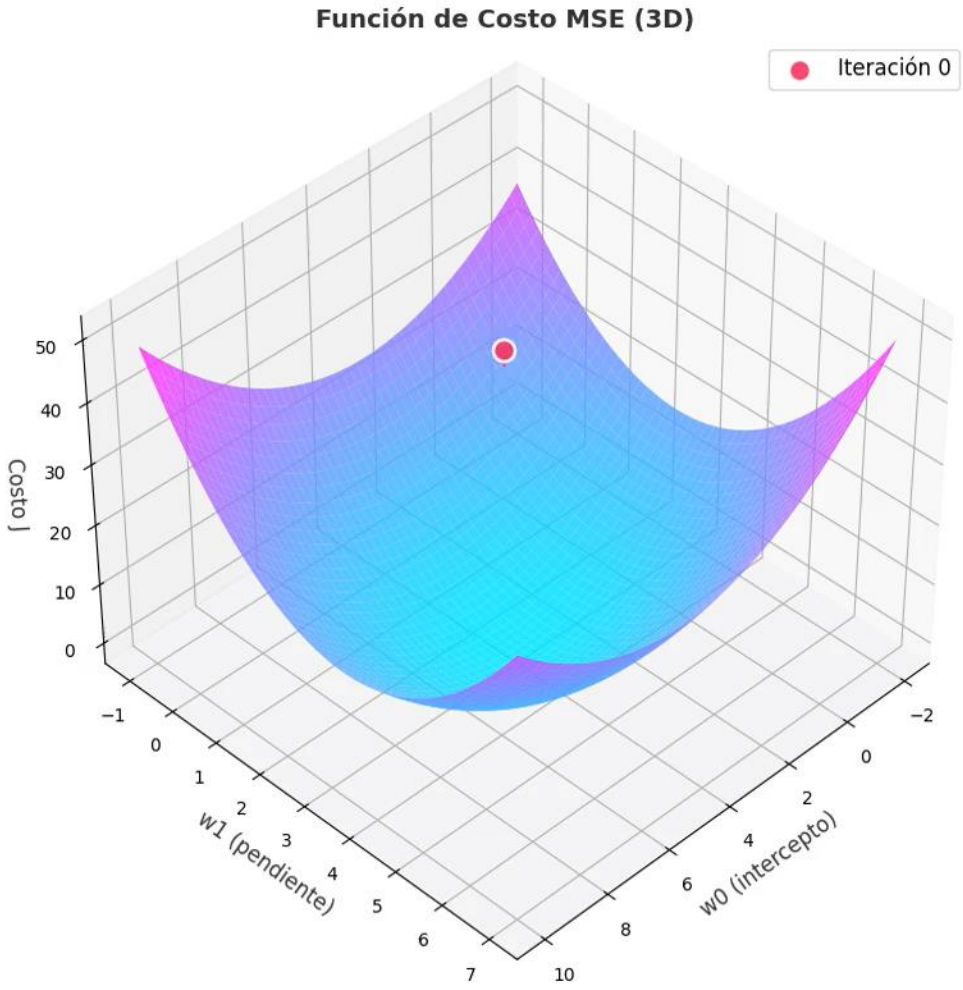


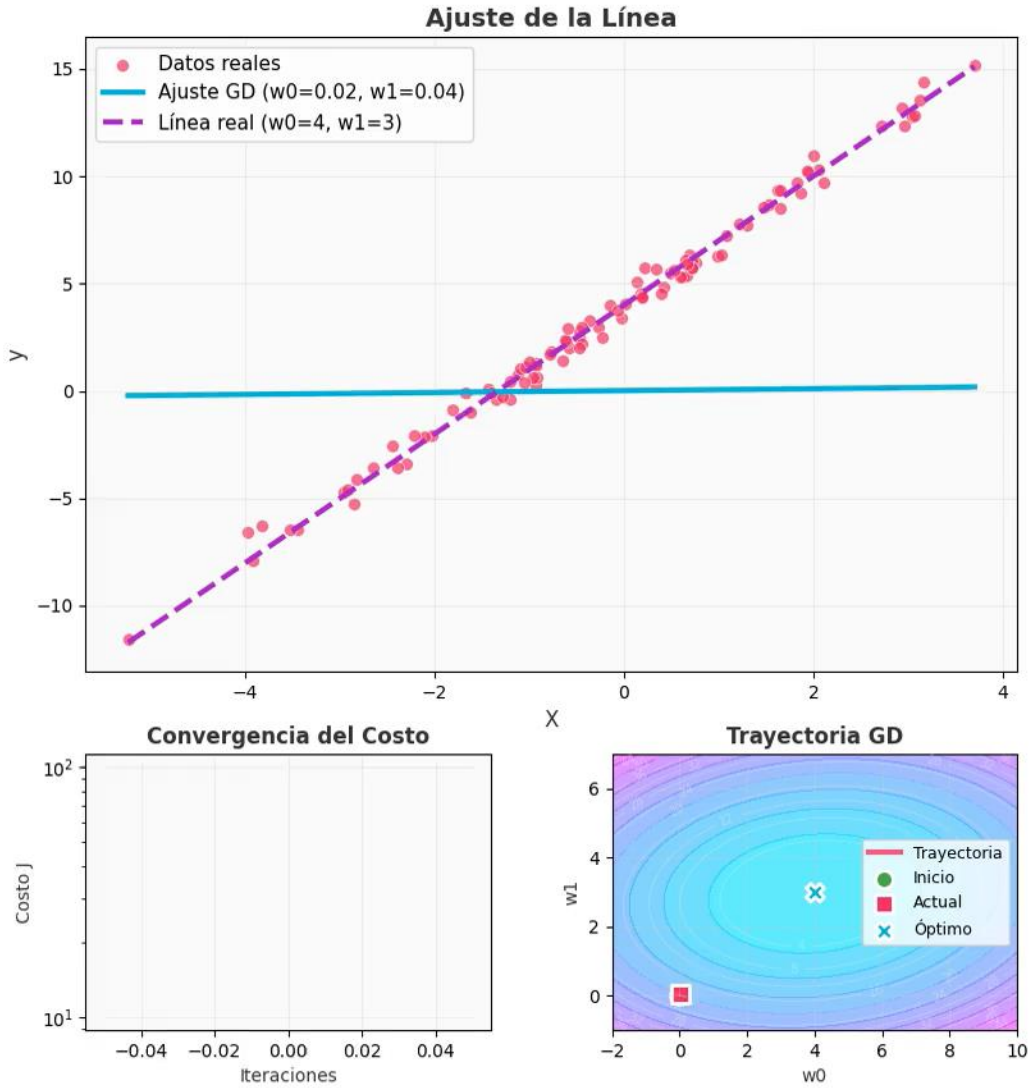
Cualquier coordenada inicial convergerá al mismo punto, el único mínimo local

Regresión lineal

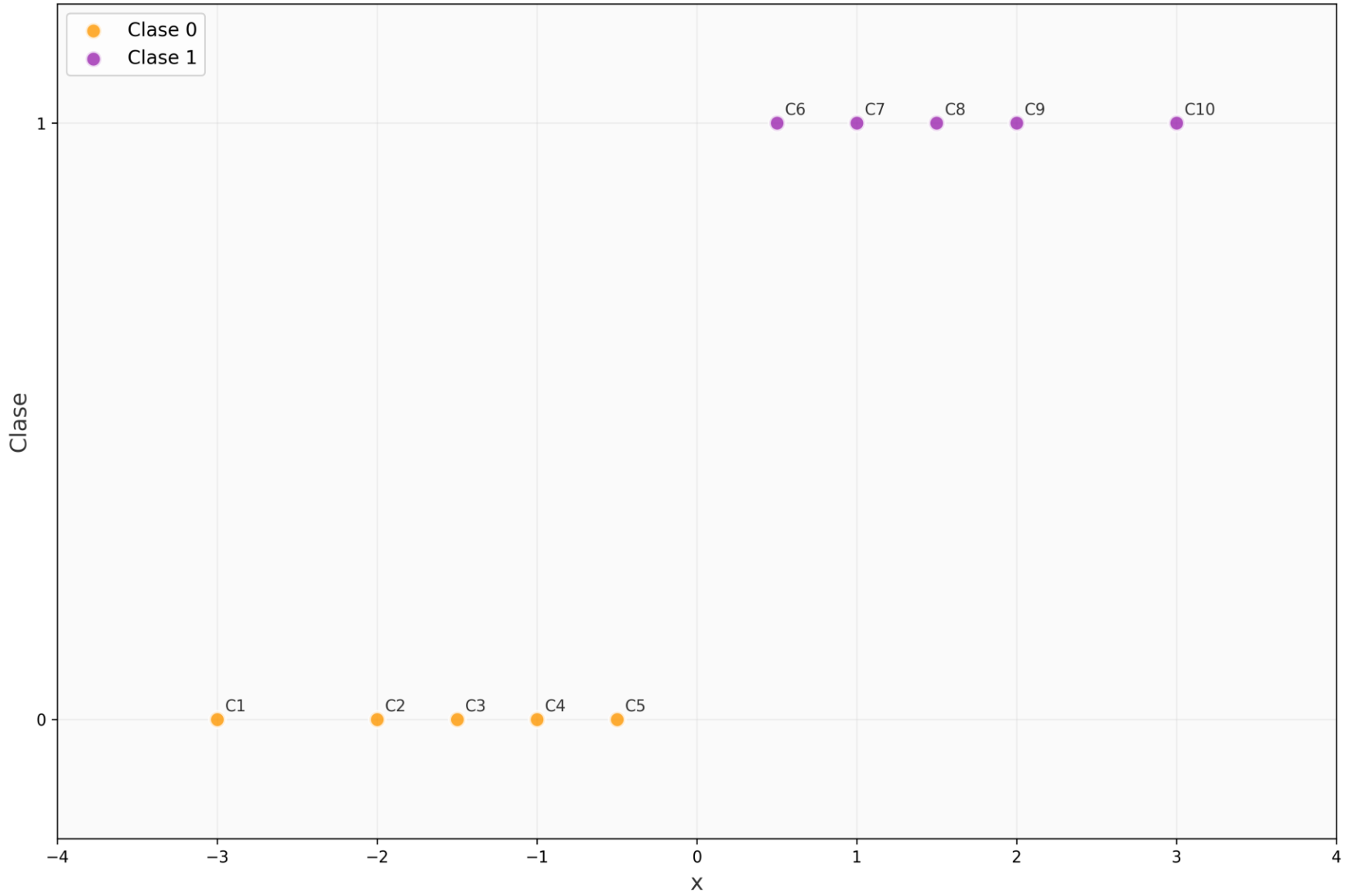


Regresión lineal



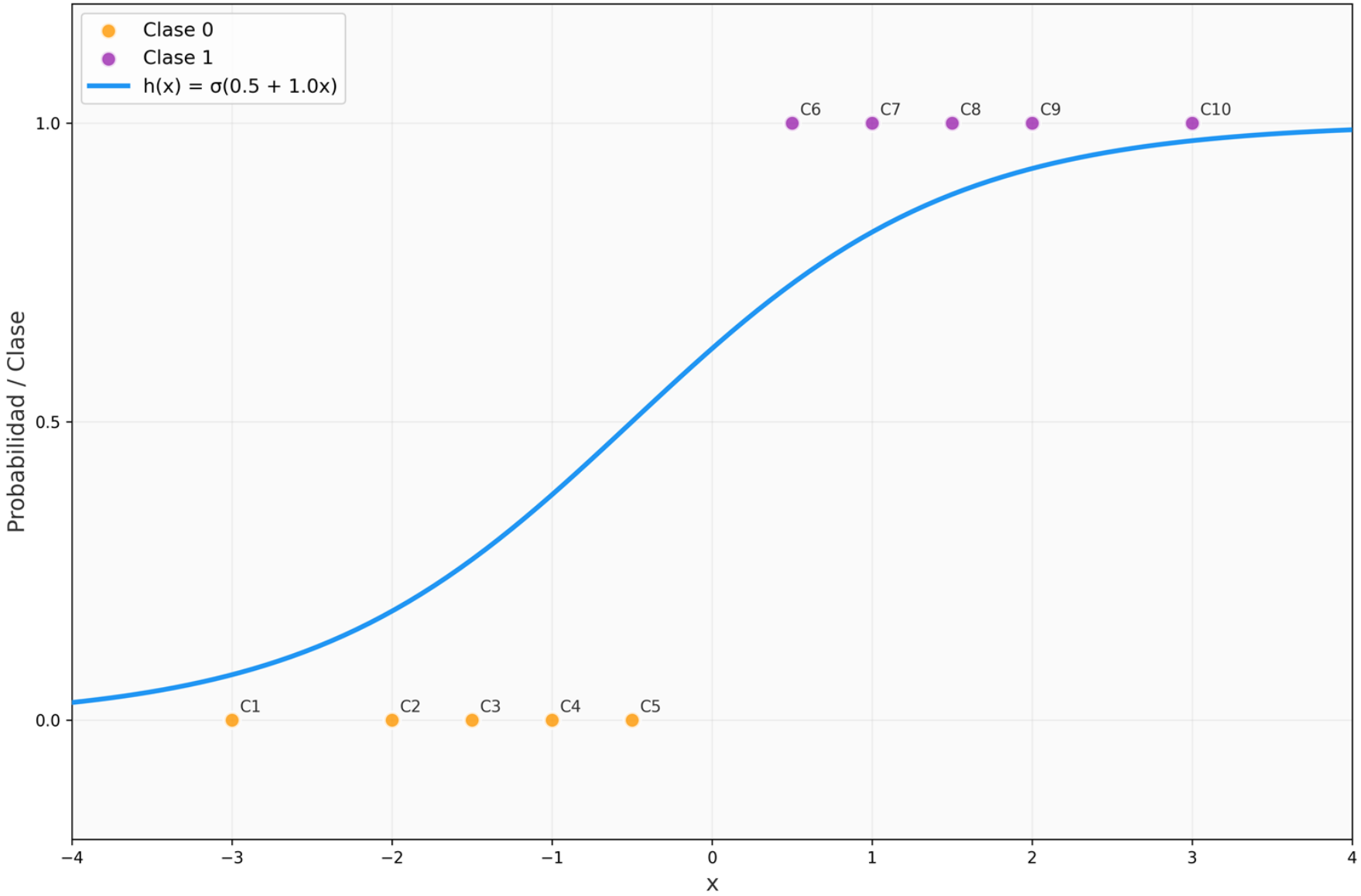


Regresión lineal

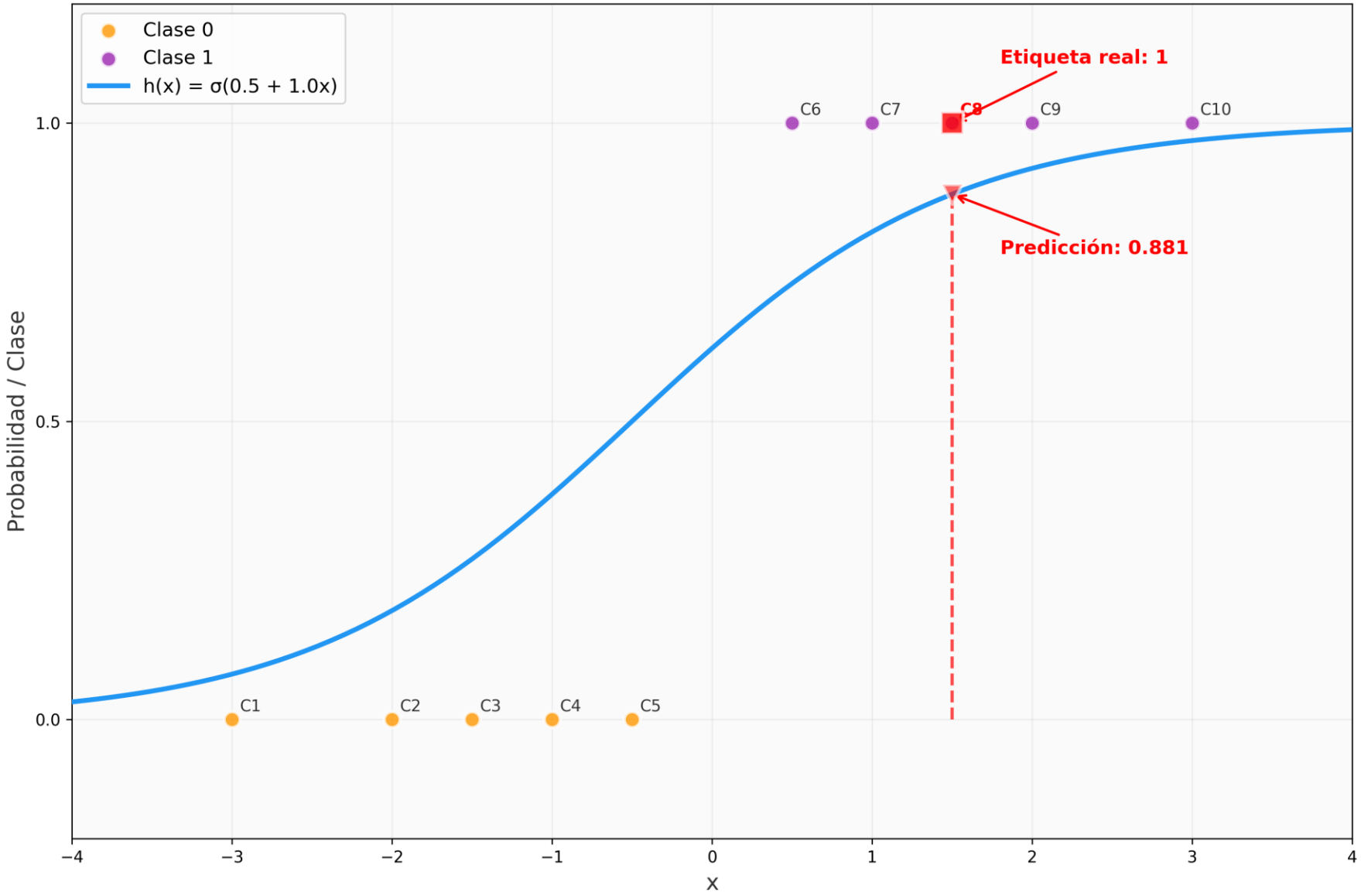


X	Clase (y)
-3.0	0
-2.0	0
-1.5	0
-1.0	0
-0.5	0
0.5	1
1.0	1
1.5	1
2.0	1
3.0	1

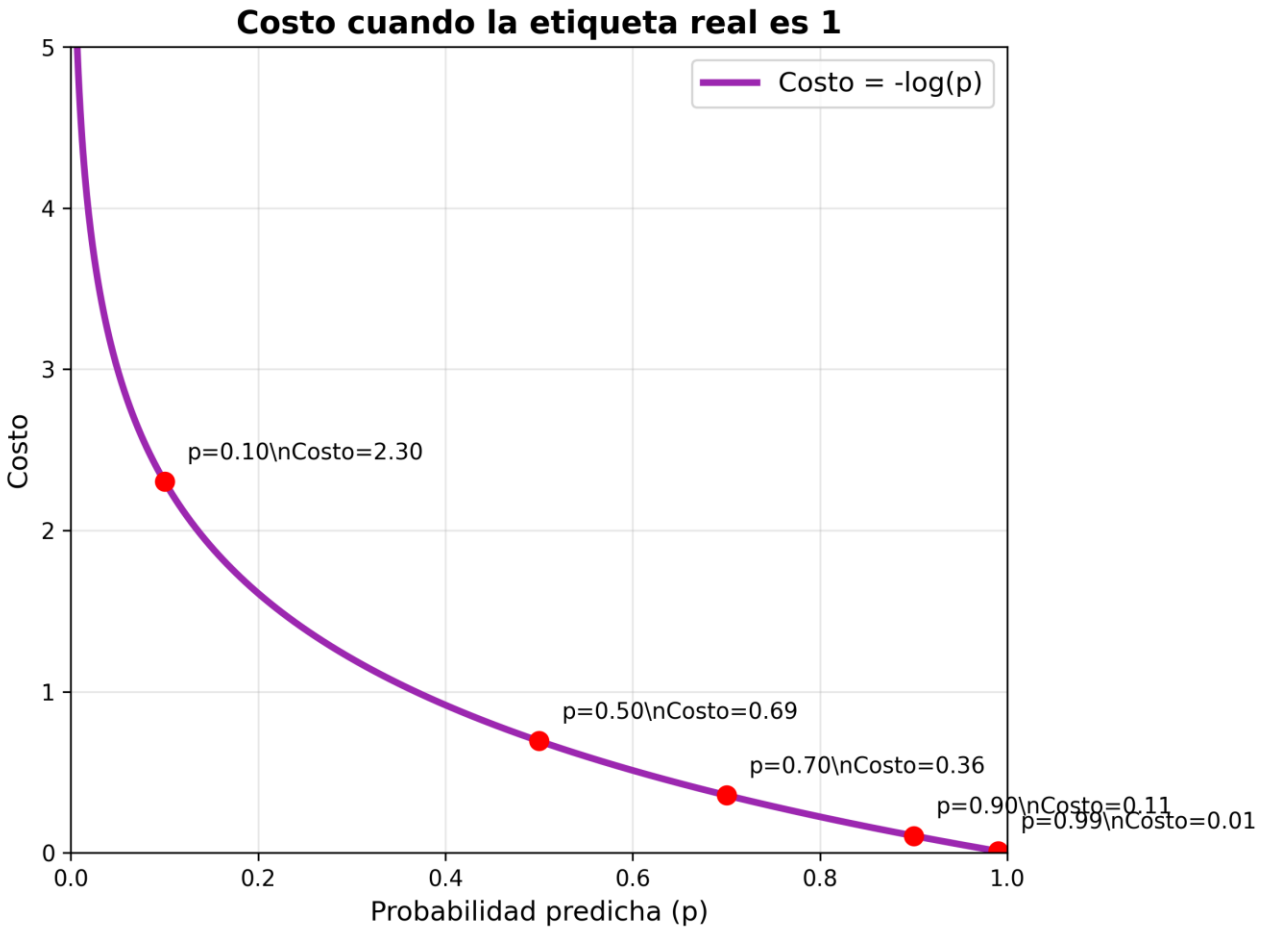
Regresión lineal

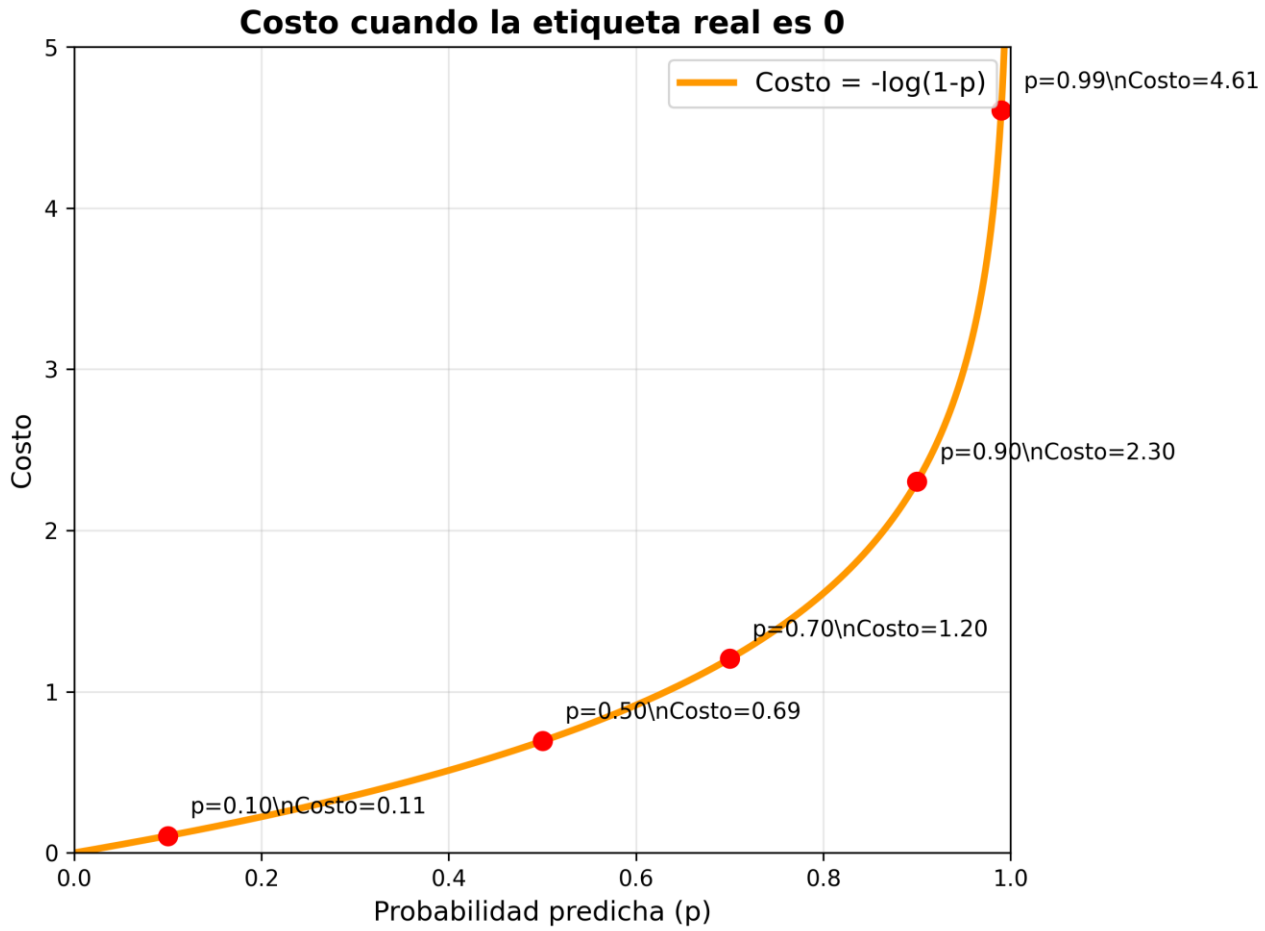


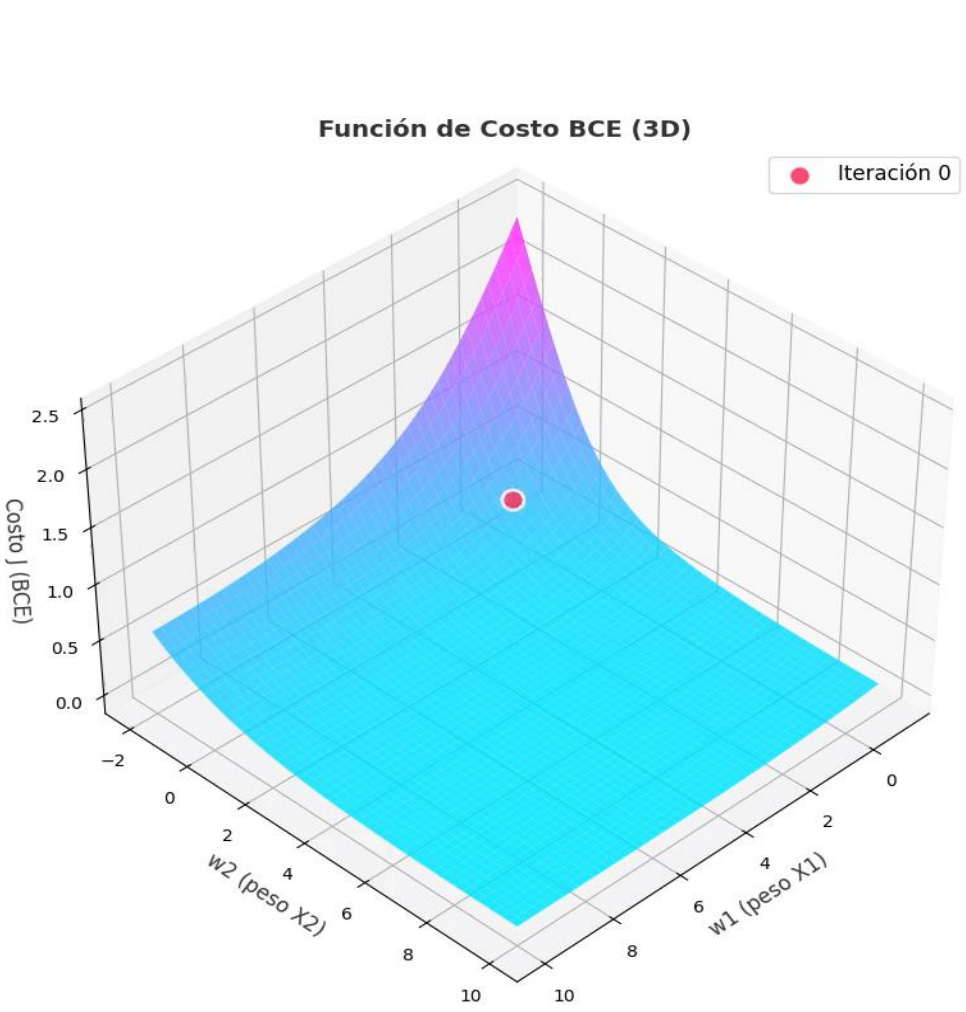
Regresión lineal

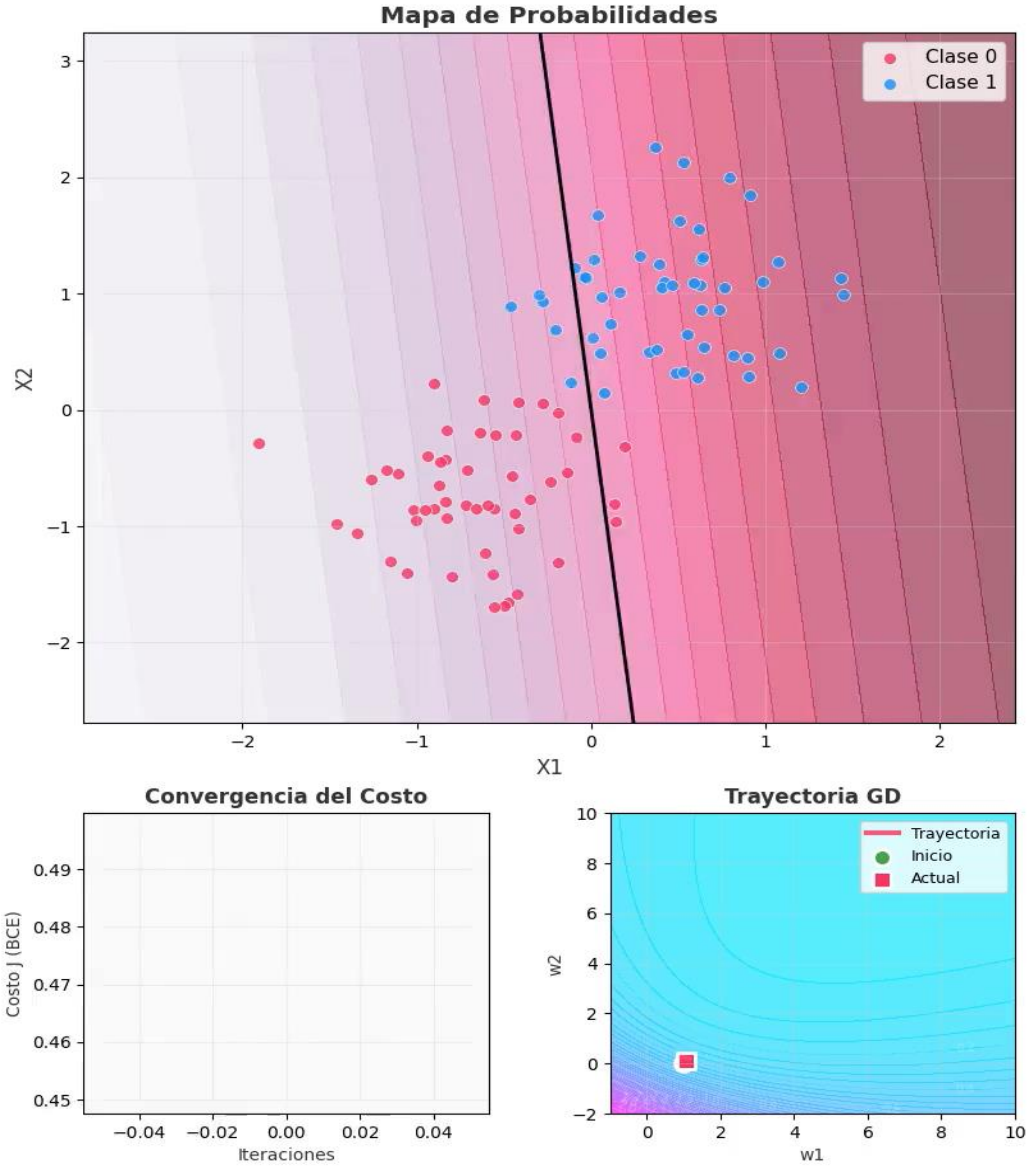


Regresión lineal






Función de Costo BCE (3D)
Iteración 0
Costo J (BCE)
w2 (peso X2)
w1 (peso X1)


Mapa de Probabilidades
Clase 0
Clase 1
X1
X2
Convergencia del Costo
Costo J (BCE)
Iteraciones
Trayectoria GD
Trayectoria
Inicio
Actual
w1
w2