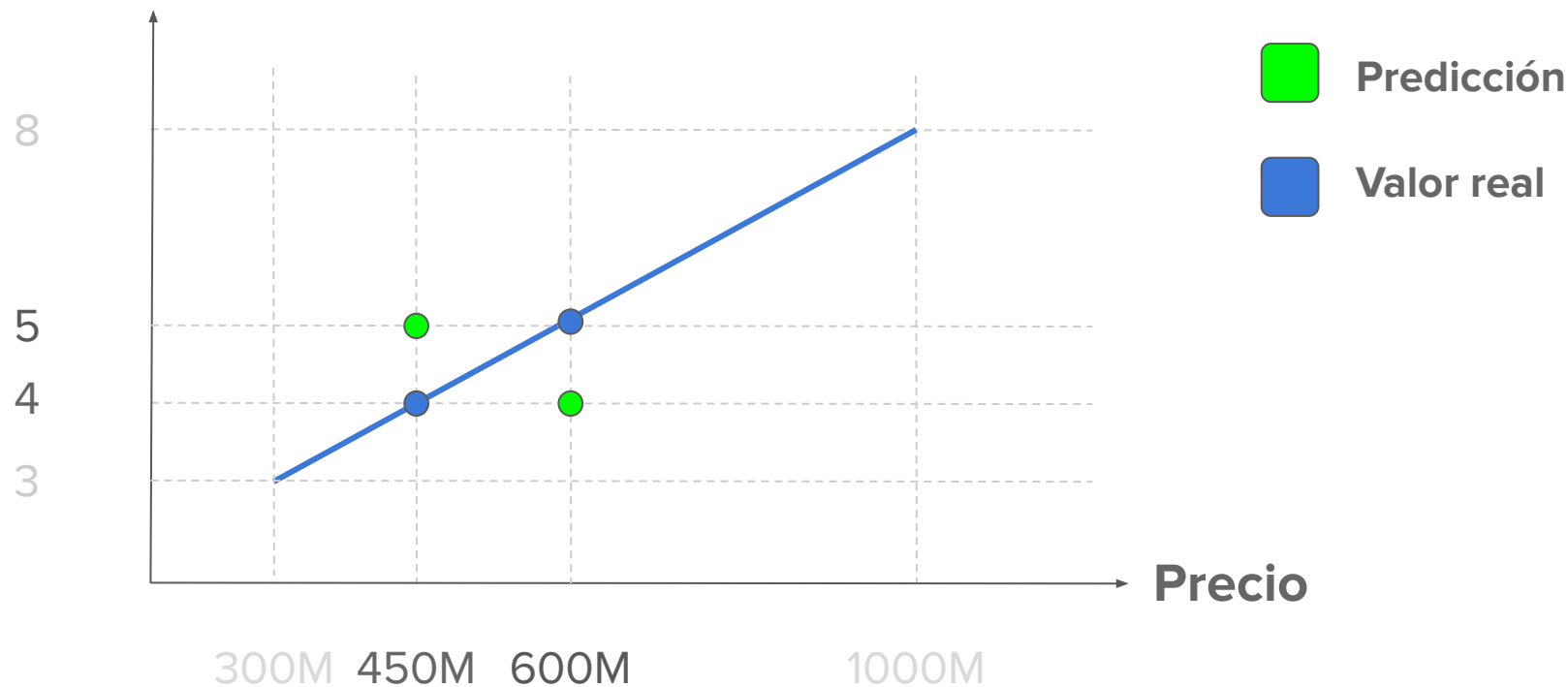


Descenso del gradiente

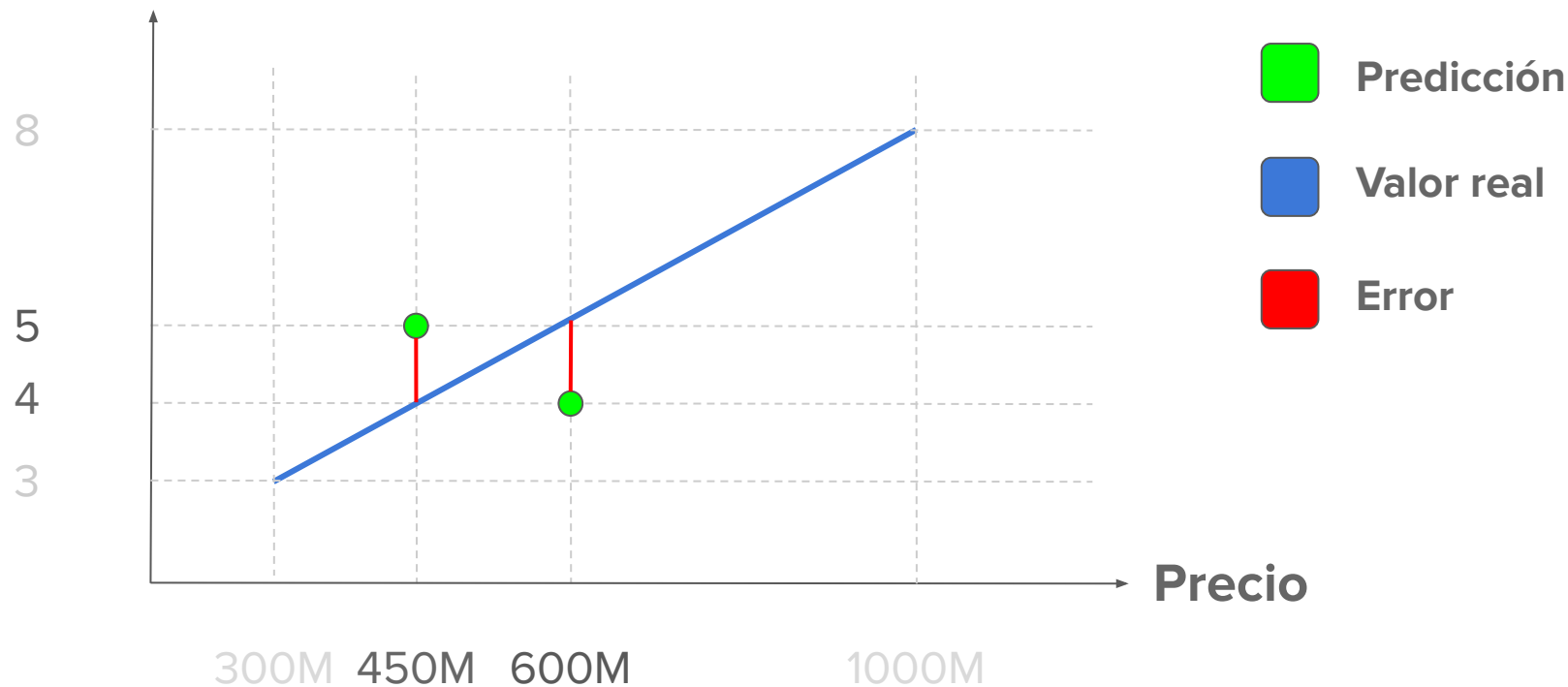
Medir el error en una regresión

Para verificar que el sistema “Aprende”,
podemos medir cuánto se equivoca en cada
predicción e ir corrigiendo

Habitaciones



Habitaciones



El error de un punto será la diferencia entre el valor real y el valor predicho

El error total será la suma de la media de esos errores

	Predicción	Valor Real	Error (distancia)
Valor 1:	4	5	1
Valor 2:	5	4	1

Error cuadrático medio

Matemáticamente

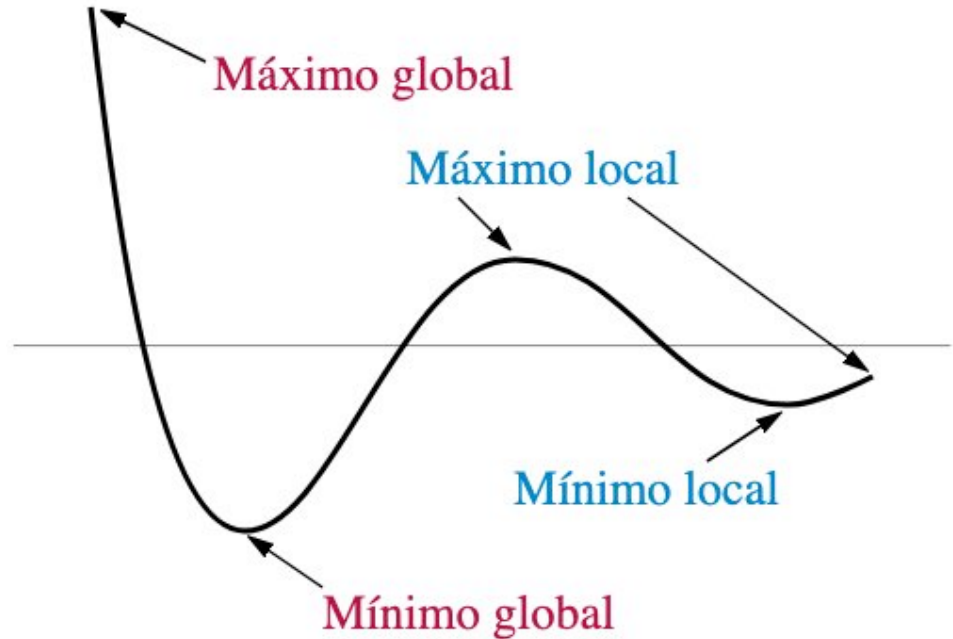
$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

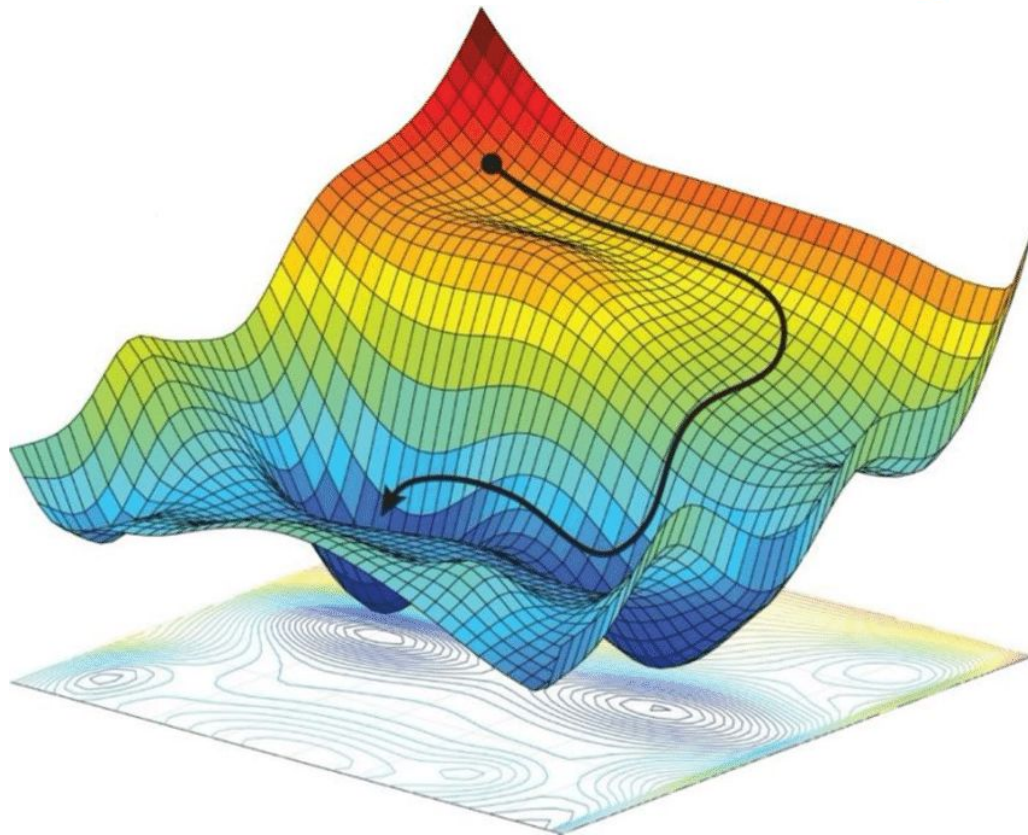
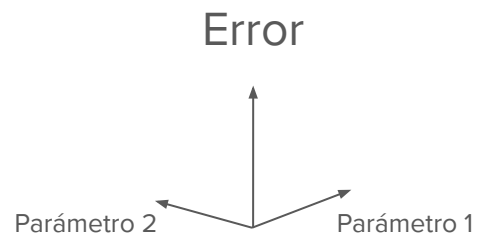
La tarea será encontrar el valor mínimo de esa función, para minimizar el error

Vamos a llamarlo
Función de coste

Para encontrar los mínimos de una función, se debe igualar a 0 su derivada

$$f'(x) = 0$$

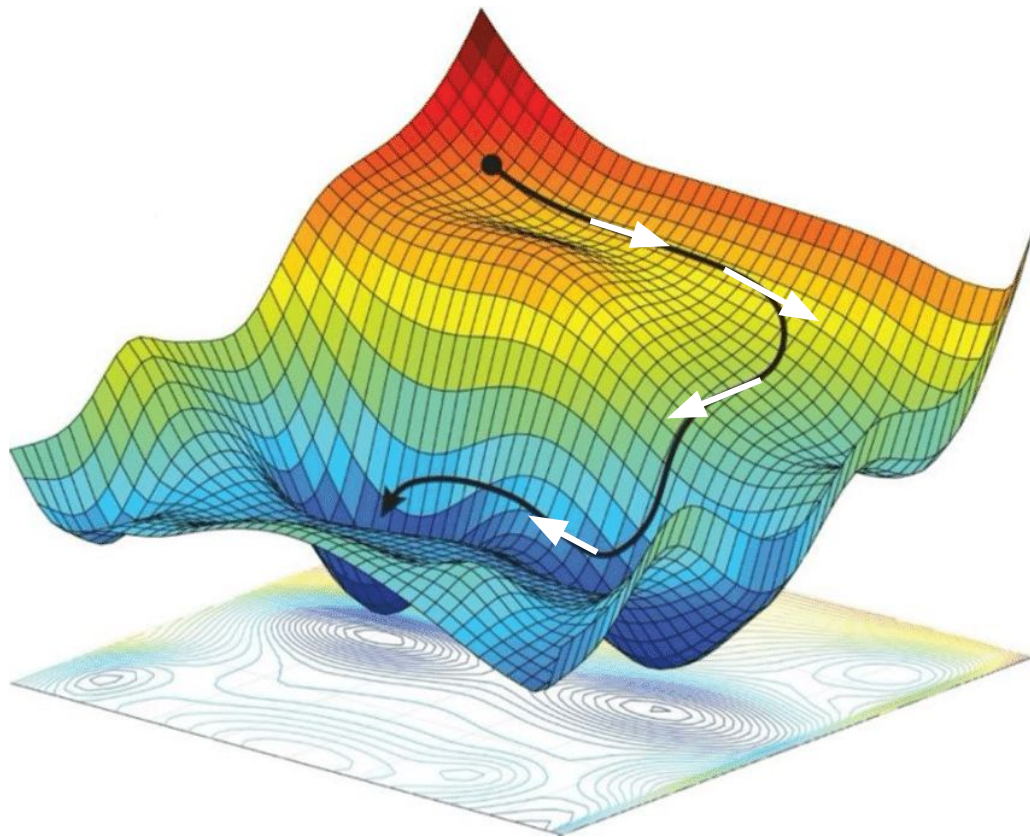




En varias variables se deben calcular las derivadas con respecto a cada variable (Derivadas parciales)

$$\nabla f$$

Para esto se usa el **gradiente**, el cual indica la dirección y magnitud en que deben ajustarse los pesos para minimizar el error.



Se calcula **El Gradiente** y se mueve en la dirección contraria y se repite hasta que se llegue cerca a un mínimo

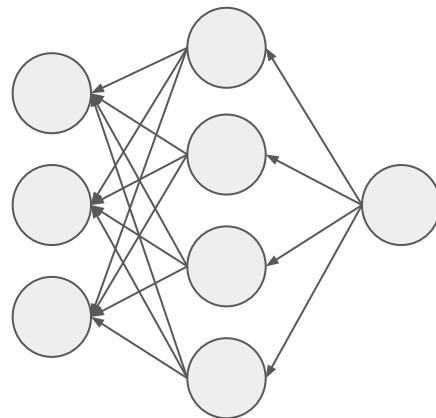
Ejemplo interactivo

Backpropagation

Es un algoritmo para ajustar los pesos de las conexiones **(w)** y **(b)**.

Funciona mediante la propagación hacia atrás del error, desde la salida hasta las entradas.

Durante este proceso, se calcula el gradiente de la función de pérdida **respecto a cada peso**, lo que permite actualizar los pesos en la dirección que reduce el error. En PyTorch es `Backward()`



En resumen...

El **error cuadrático medio** es una medida que calcula el promedio de los cuadrados de las diferencias entre las predicciones de un modelo y los valores reales, indicando qué tan precisas son las predicciones.

El **descenso del gradiente** es un método de optimización que sirve para minimizar la función de coste.