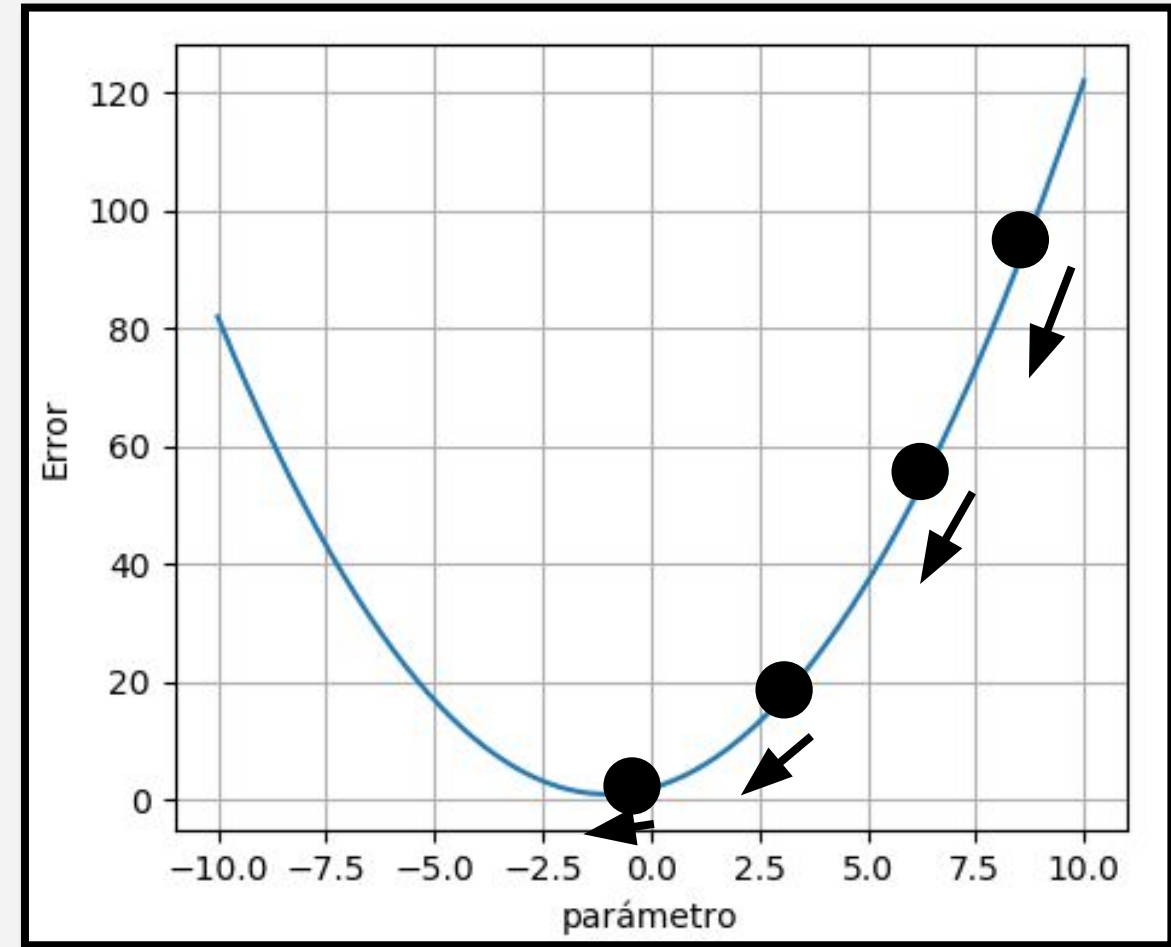


# Descenso de Gradiente

---

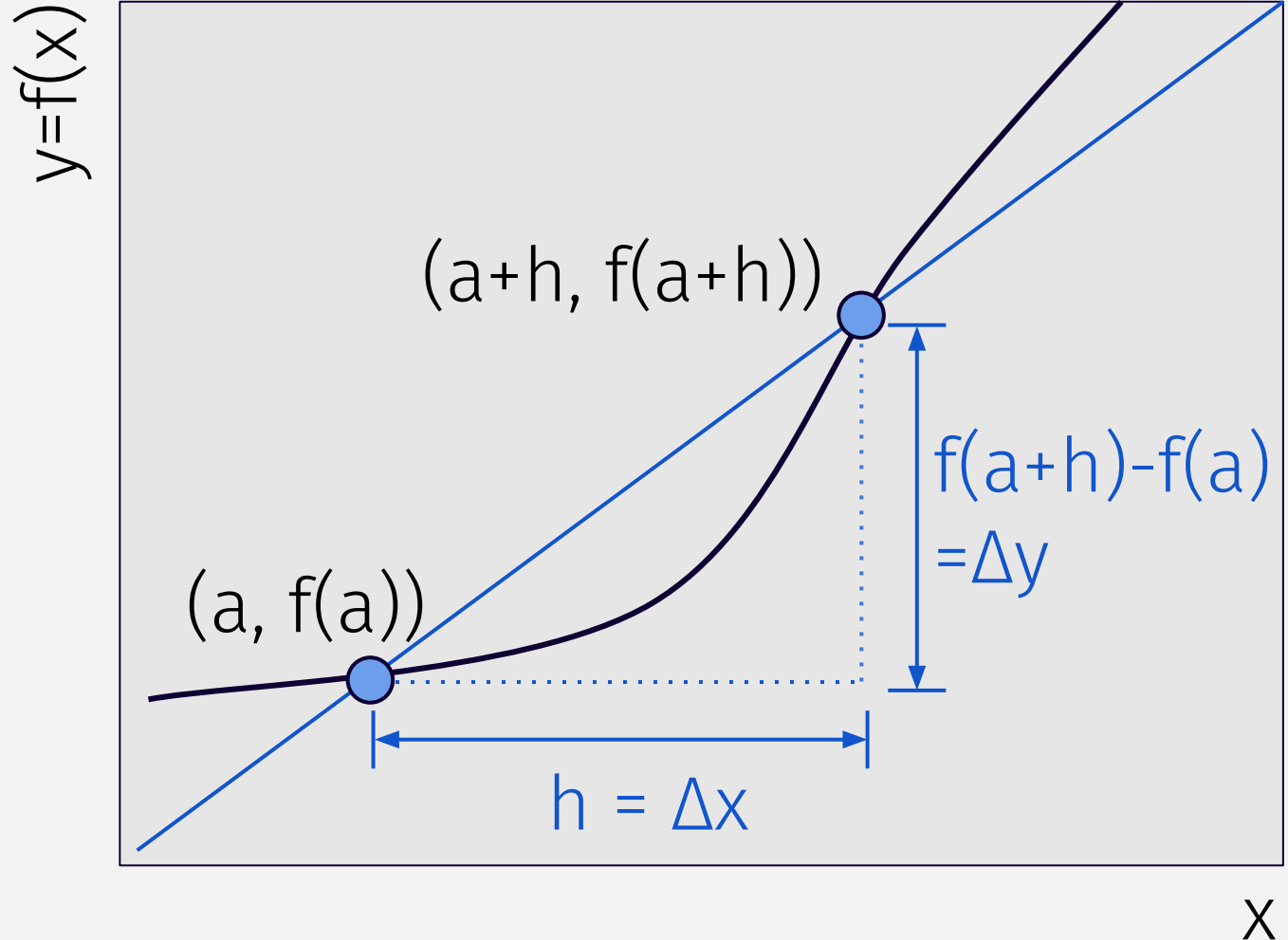
# Descenso de gradiente

- Descenso de gradiente
  - Iterativo
  - Generalizable
    - Regresión Lineal
    - Regresión Logística
    - Redes Neuronales
    - Máquinas de Vectores de Soporte
      - Cualquier modelo con  $E$  derivable
  - Escalable (con modificaciones)
    - Millones de ejemplos



# Derivadas

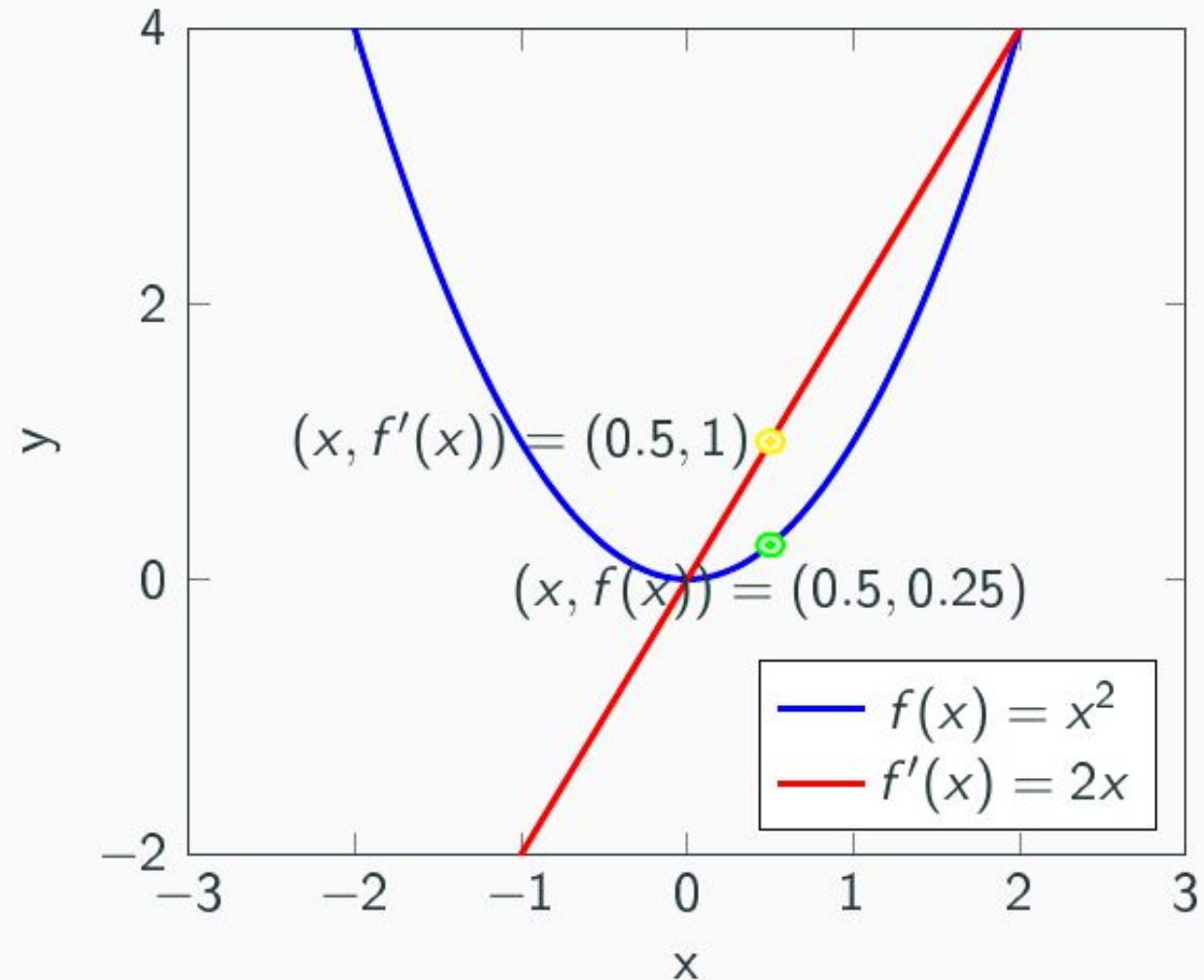
- Función  $f(x) = \dots$ 
  - Alternativamente  $y = f(x)$
  - Derivable  $\rightarrow$  suave
- Derivada
  - Función calculada a partir de otra función
- Derivada 1D:  $y' = f'(x)$ 
  - $f'(\mathbf{x})$  = pendiente a la recta tangente en el punto  $\mathbf{x}$
  - Razón de cambio  **$dy/dx$**
  - Signo indica dirección de crecimiento



# Derivada 1D como indicador de crecimiento/decrecimiento

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} = \begin{cases} > 0 & \text{si } f \text{ crece} \\ < 0 & \text{si } f \text{ decrece} \\ = 0 & \text{pto crítico} \end{cases}$$

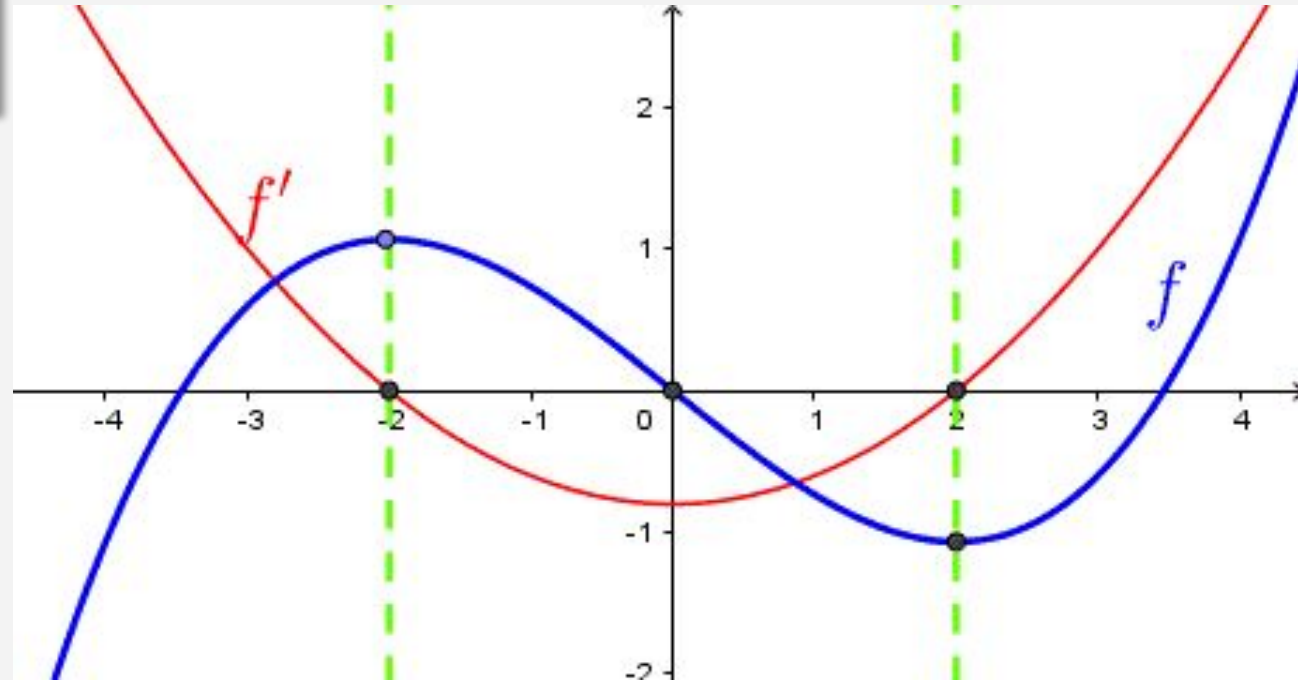
- Cociente incremental
  - Límite por izquierda y por derecha
  - Signo indica si crece o decrece
    - Solo alrededor del punto!



# Derivada 1D como indicador de crecimiento/decrecimiento

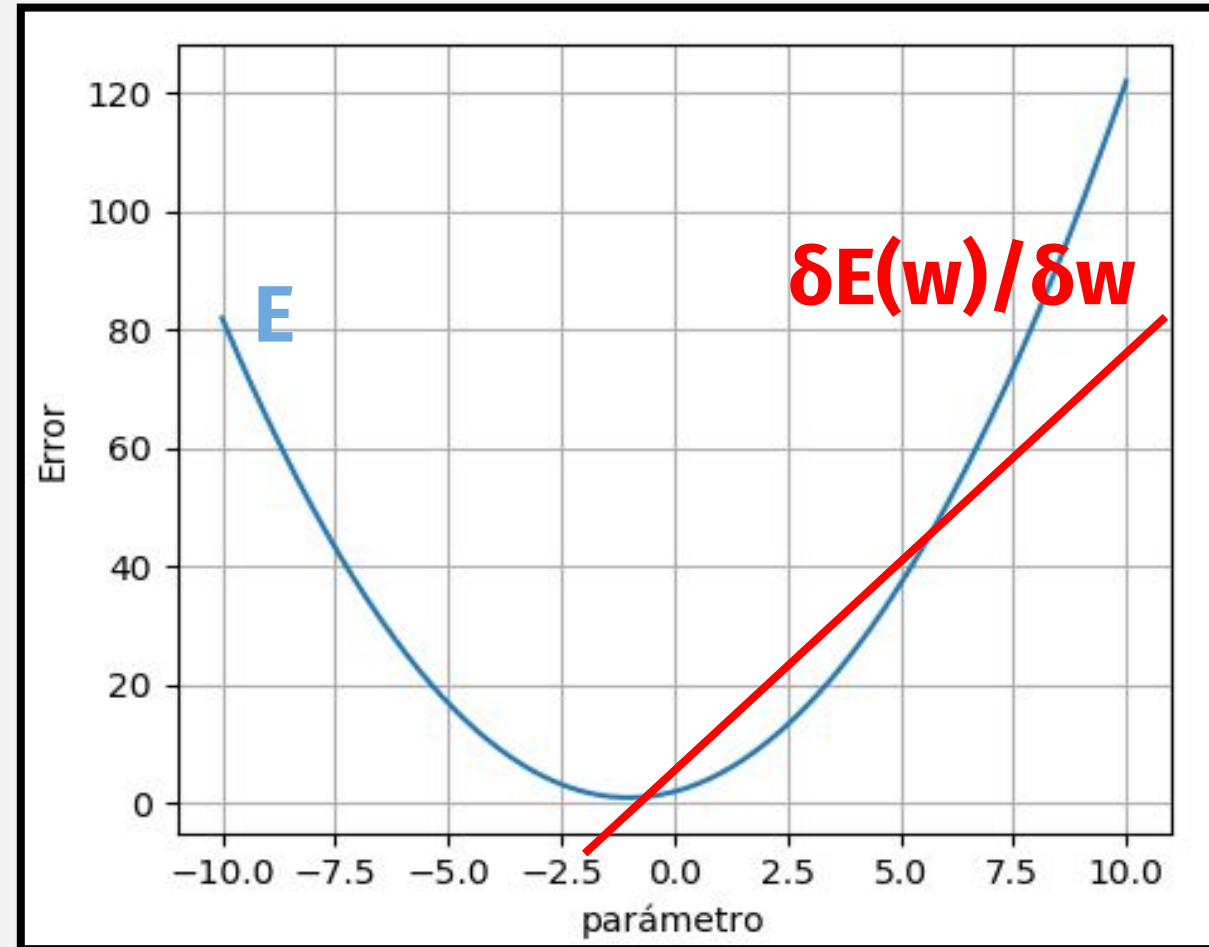
$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} = \begin{cases} > 0 & \text{si } f \text{ crece} \\ < 0 & \text{si } f \text{ decrece} \\ = 0 & \text{pto crítico} \end{cases}$$

- Cociente incremental
  - Límite por izquierda y por derecha
  - Signo indica si crece o decrece
    - Solo alrededor del punto!

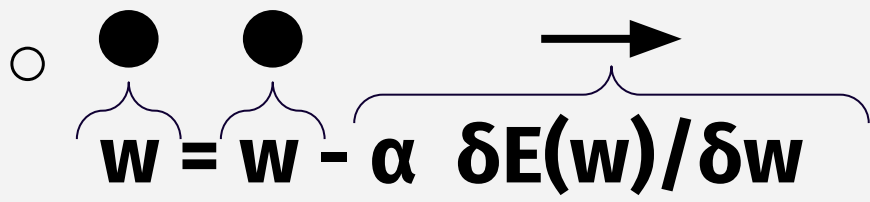


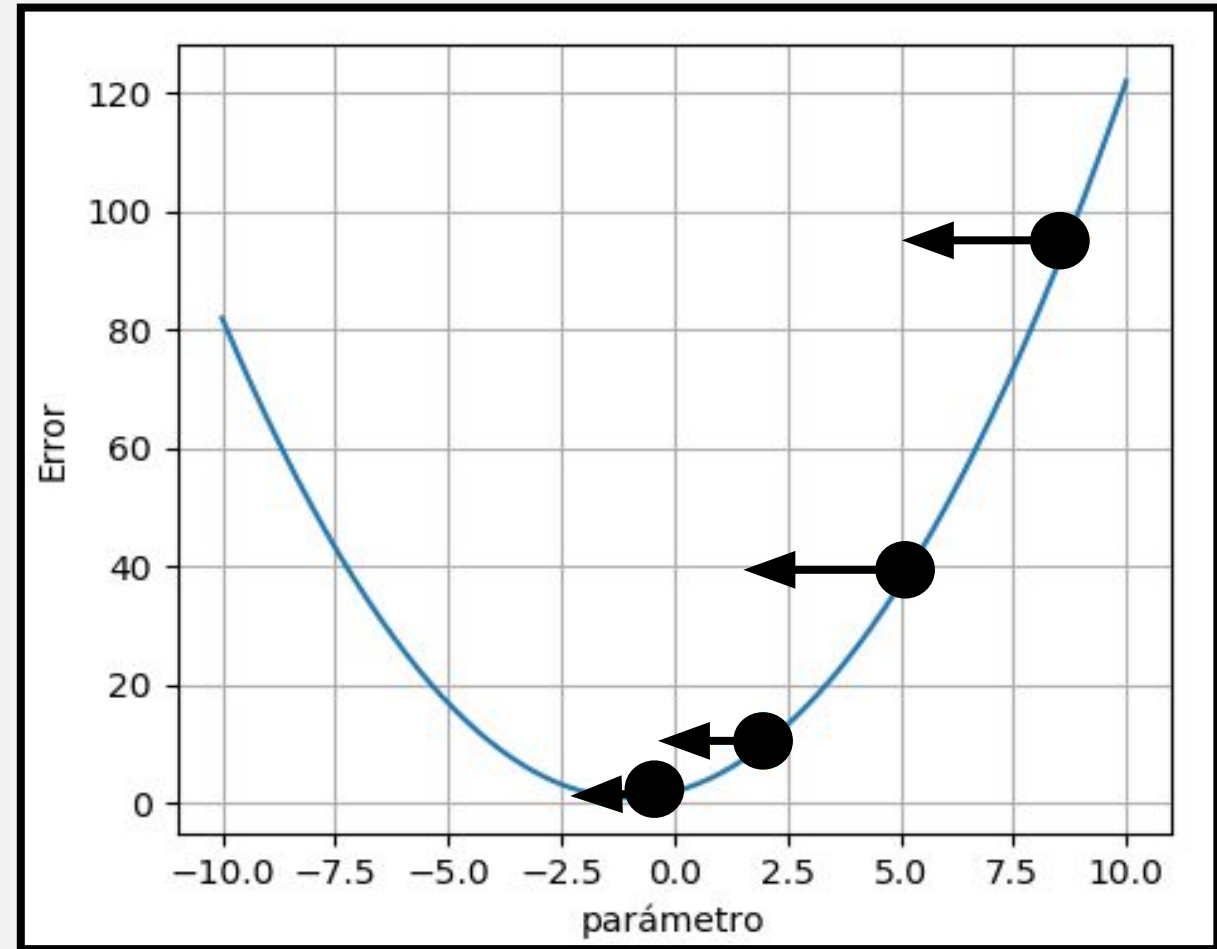
# Descenso de gradiente en 1D

- Datos
  - Función  $E$ 
    - De error u otra
  - Parámetro  $w$ 
    - Valor inicial  $w=w_0$  aleatorio
  - Derivada de  $E$ 
    - Con respecto a  $\mathbf{w}$ 
      - $\delta E(w)/\delta w$
  - Velocidad de aprendizaje  $\alpha$ 
    - Tamaño del **paso**
- Iterar
  - $\mathbf{w} = \mathbf{w} - \alpha \delta E(w)/\delta w$
  - Hasta converger



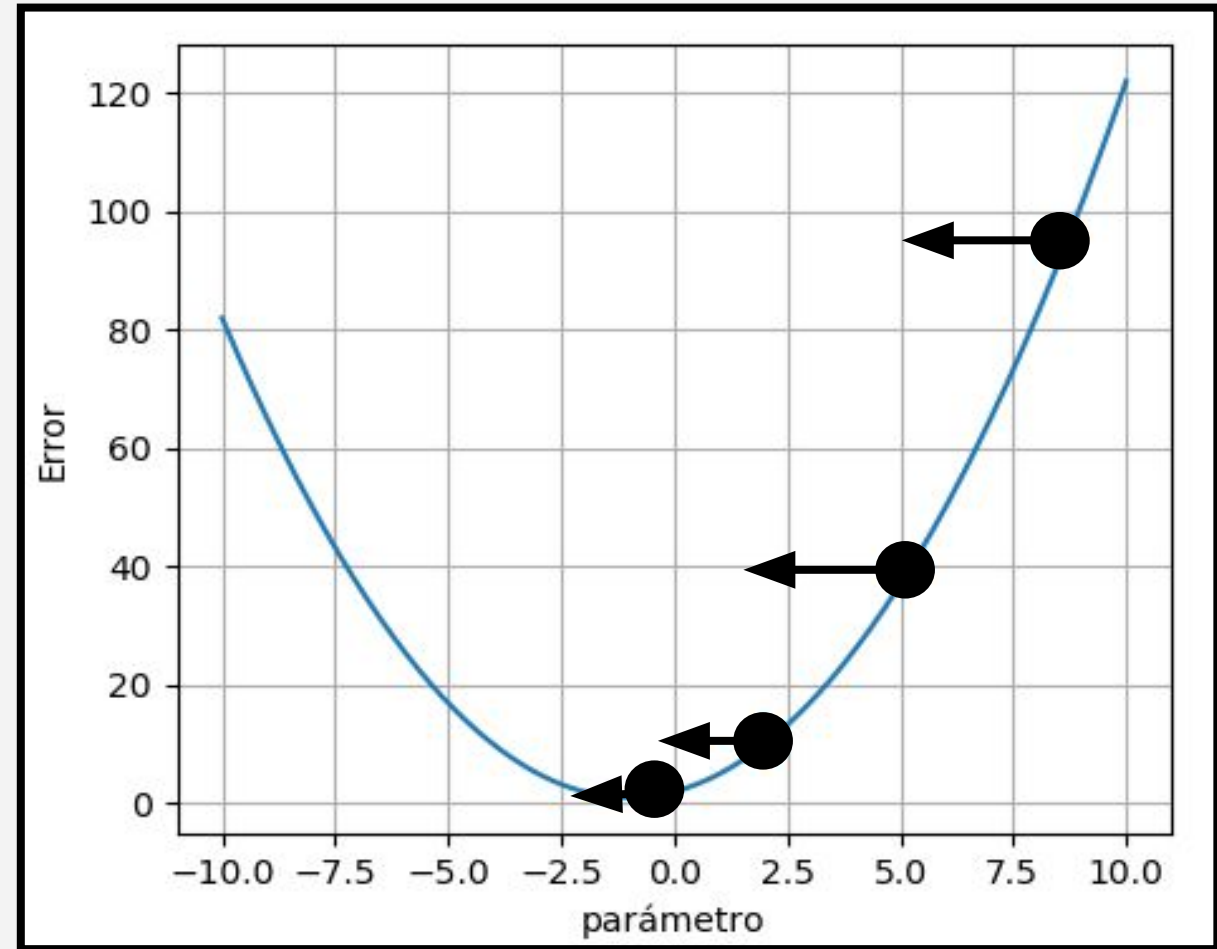
# Descenso de gradiente en 1D

- Iterar
  - 
$$\mathbf{w} = \mathbf{w} - \alpha \frac{\delta E(\mathbf{w})}{\delta \mathbf{w}}$$
- Interpretamos
  - $w = w - \langle \text{algo} \rangle$ 
    - Modificamos  $w$
  - $\delta E(w) / \delta w \rightarrow$ 
    - Dirección y magnitud del cambio
  - $\alpha$  —
    - Magnitud del cambio
  - $\alpha \delta E(w) / \delta w \rightarrow$ 
    - Dirección y magnitud del cambio



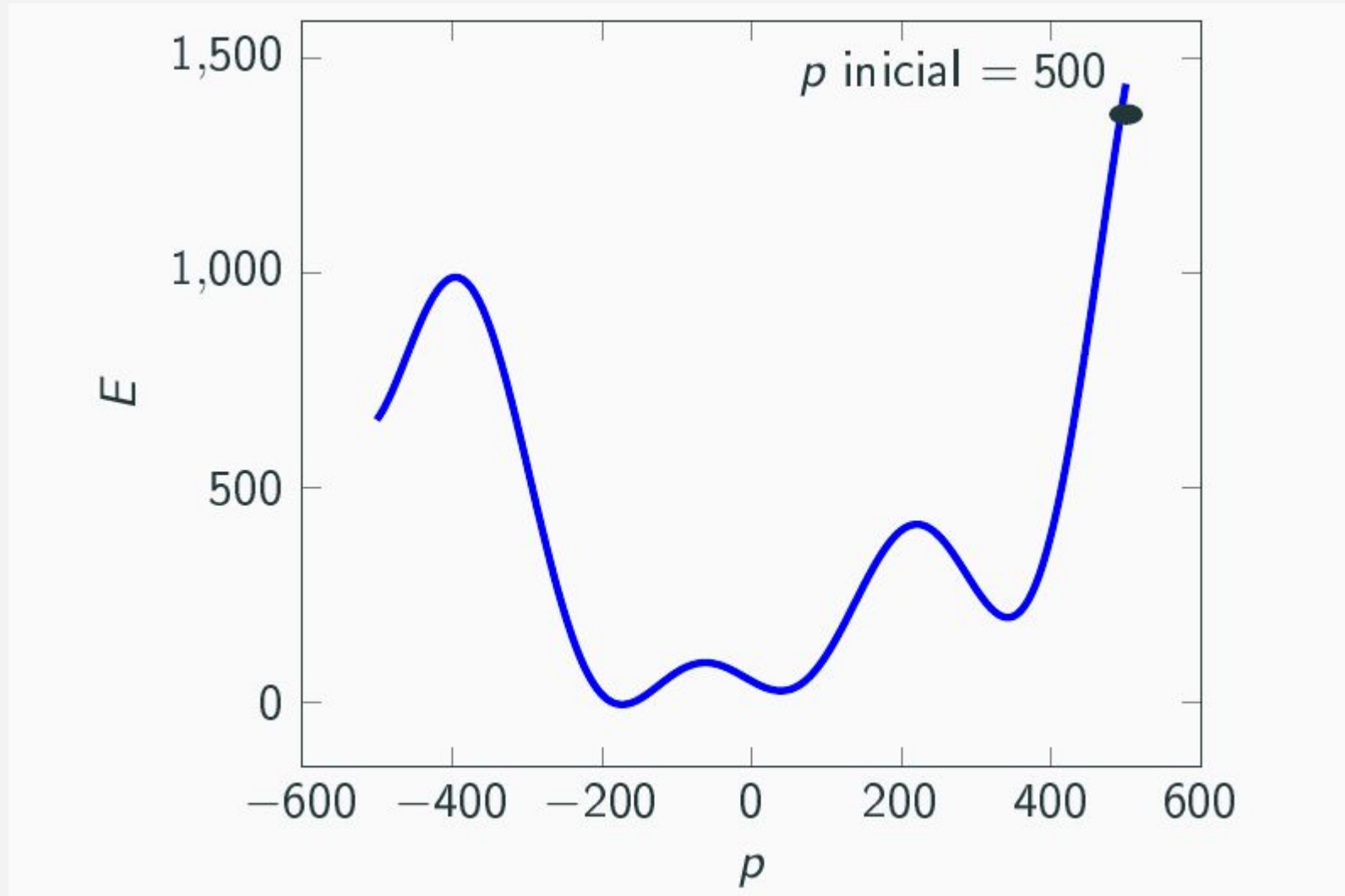
# Descenso de gradiente en 1D

```
def descenso_gradiente(E,w,x,y):  
    w = ...  
    converge=False  
    while not converge:  
         $\delta E \delta w$  = derivada(E,w,x,y)  
        w = w -  $\delta E \delta w$   
        converge = ... (depende)  
    return w
```

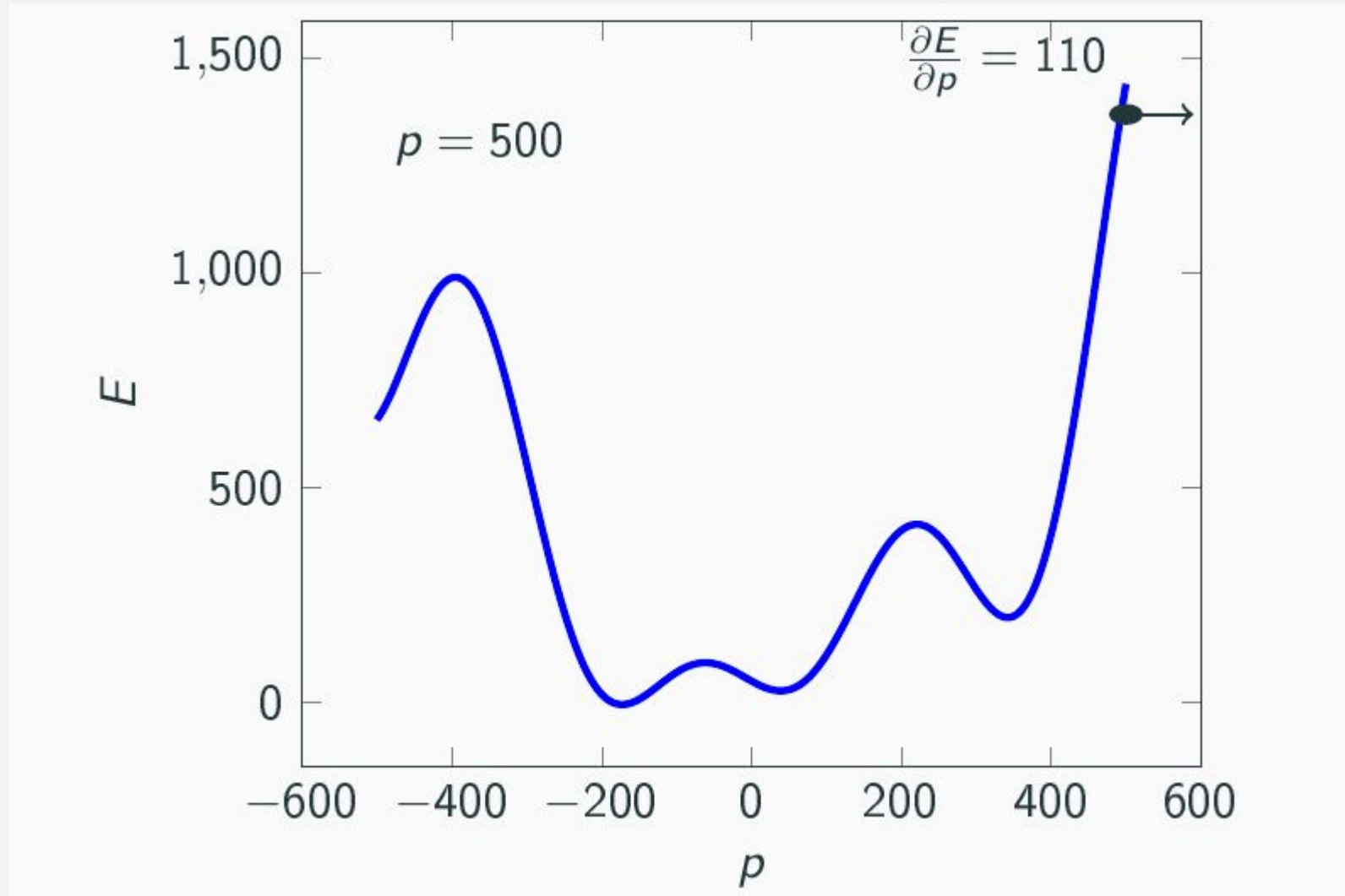




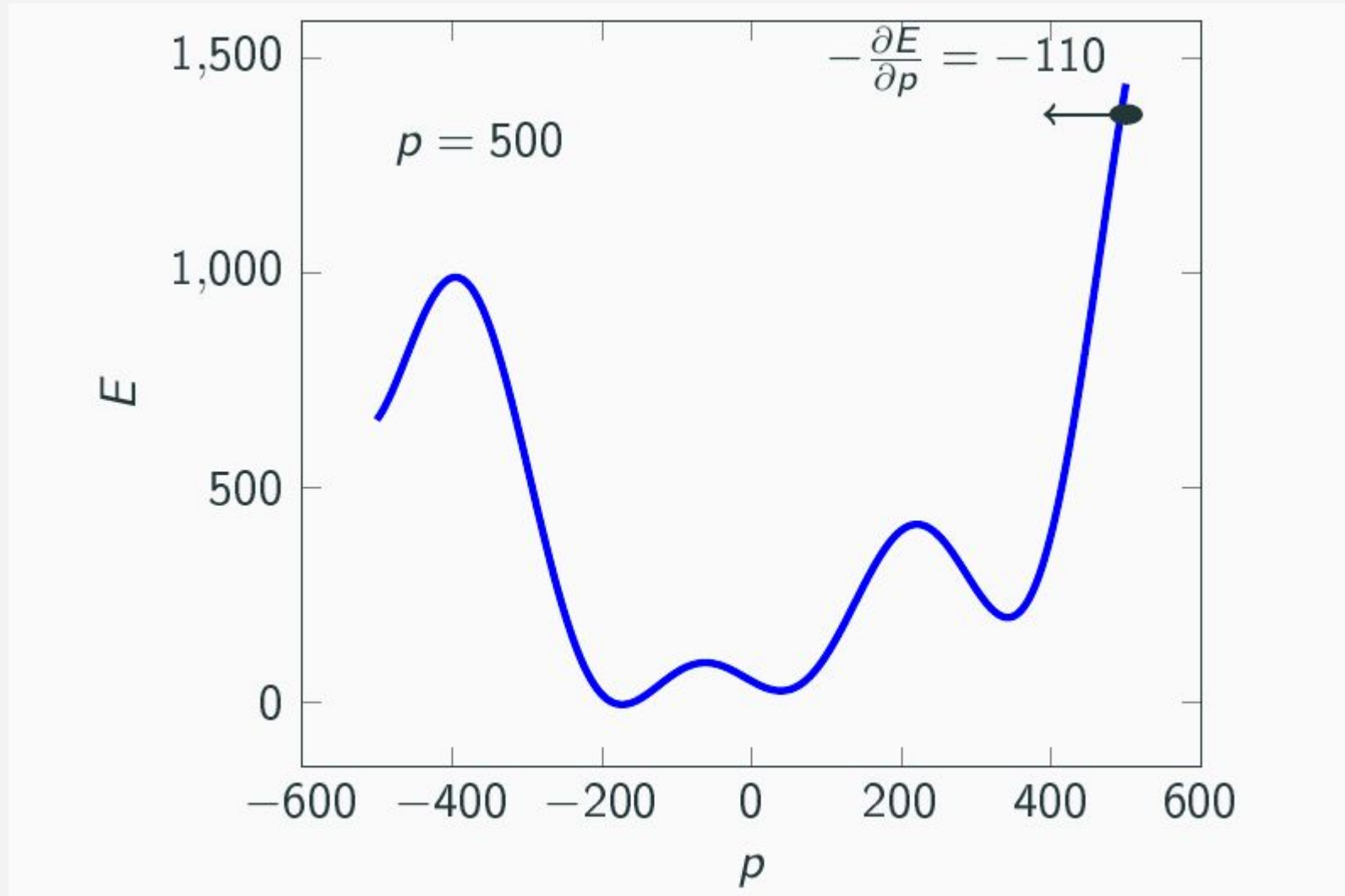
# Descenso de gradiente en 1D



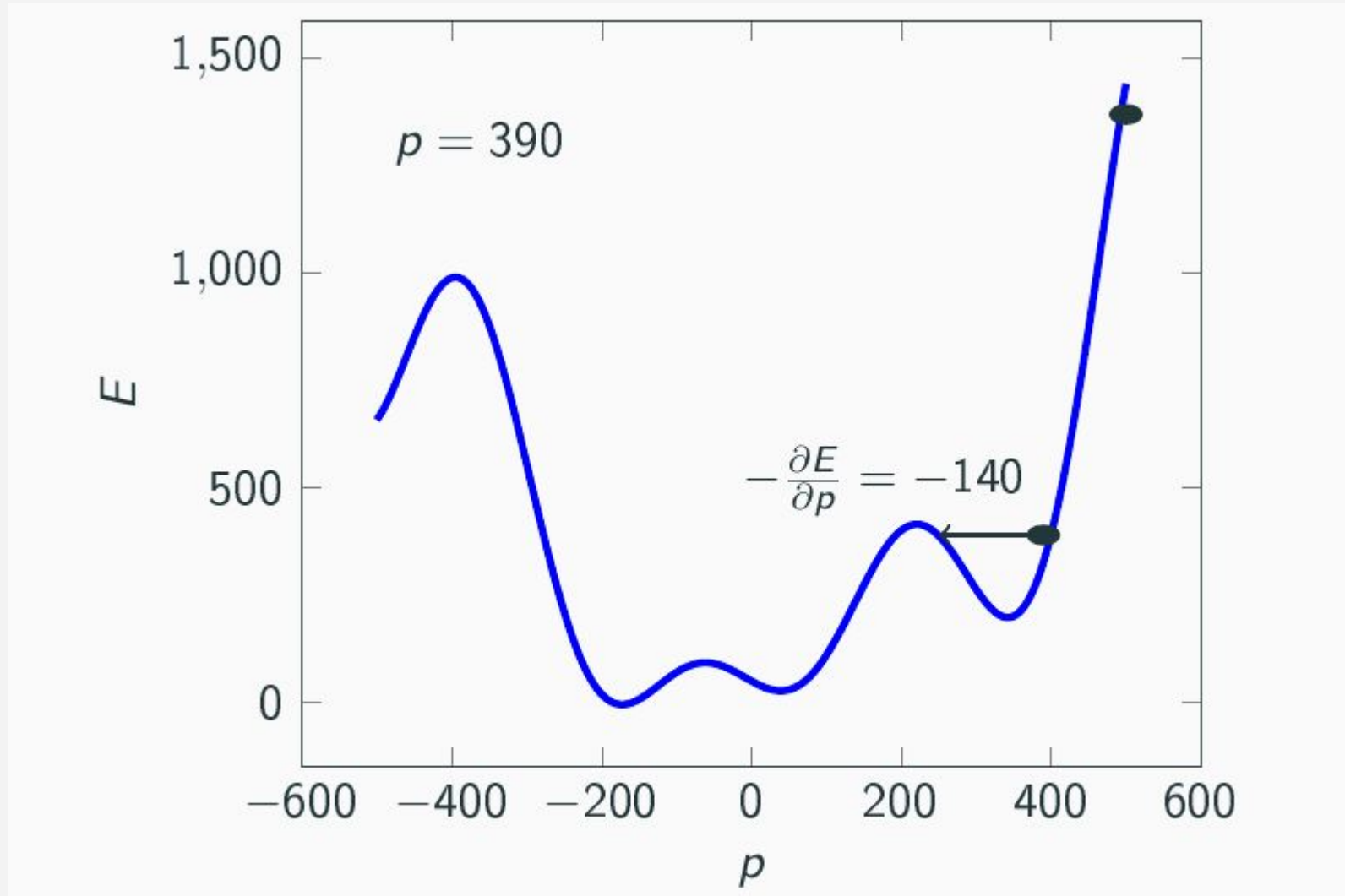
# Descenso de gradiente en 1D



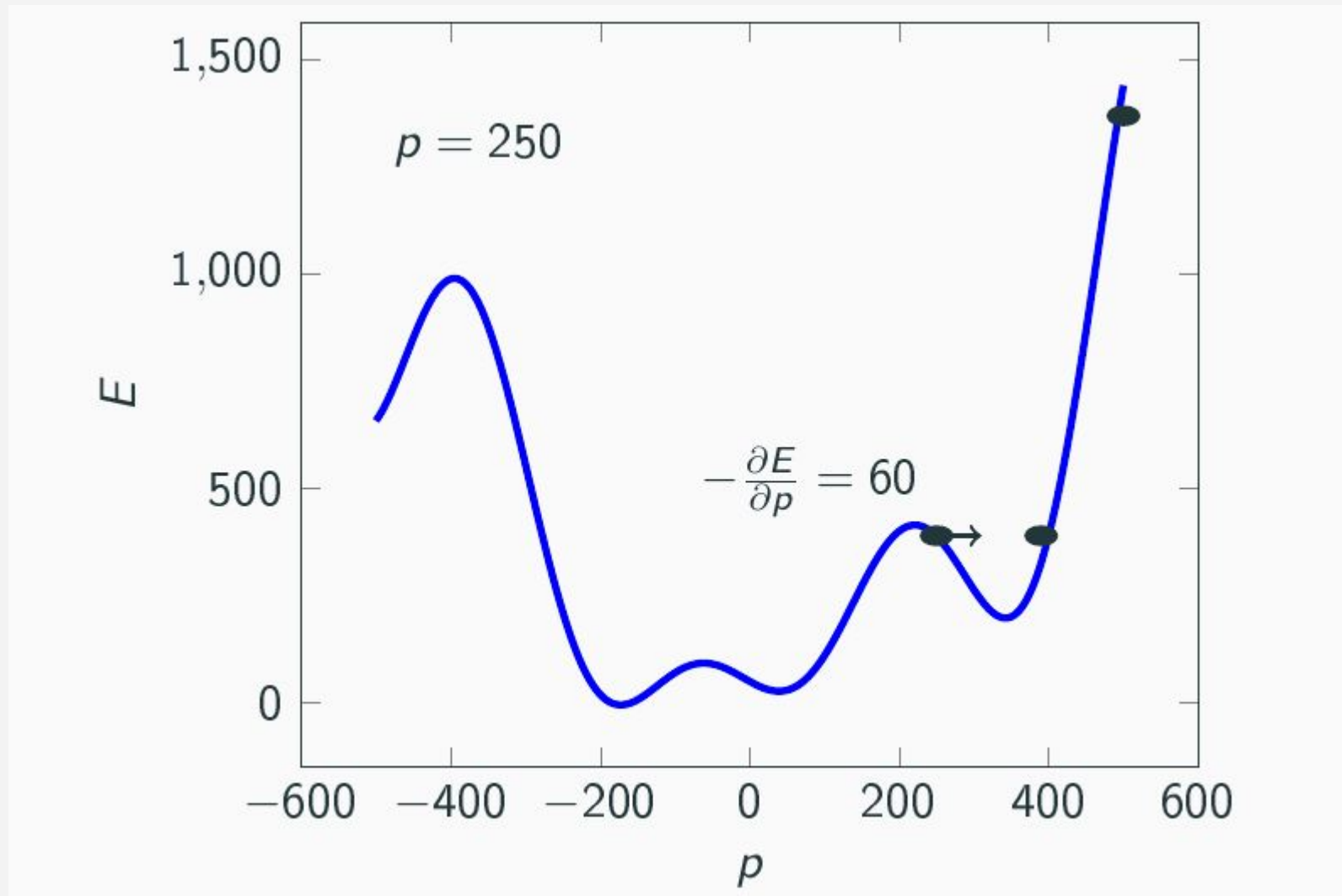
# Descenso de gradiente en 1D



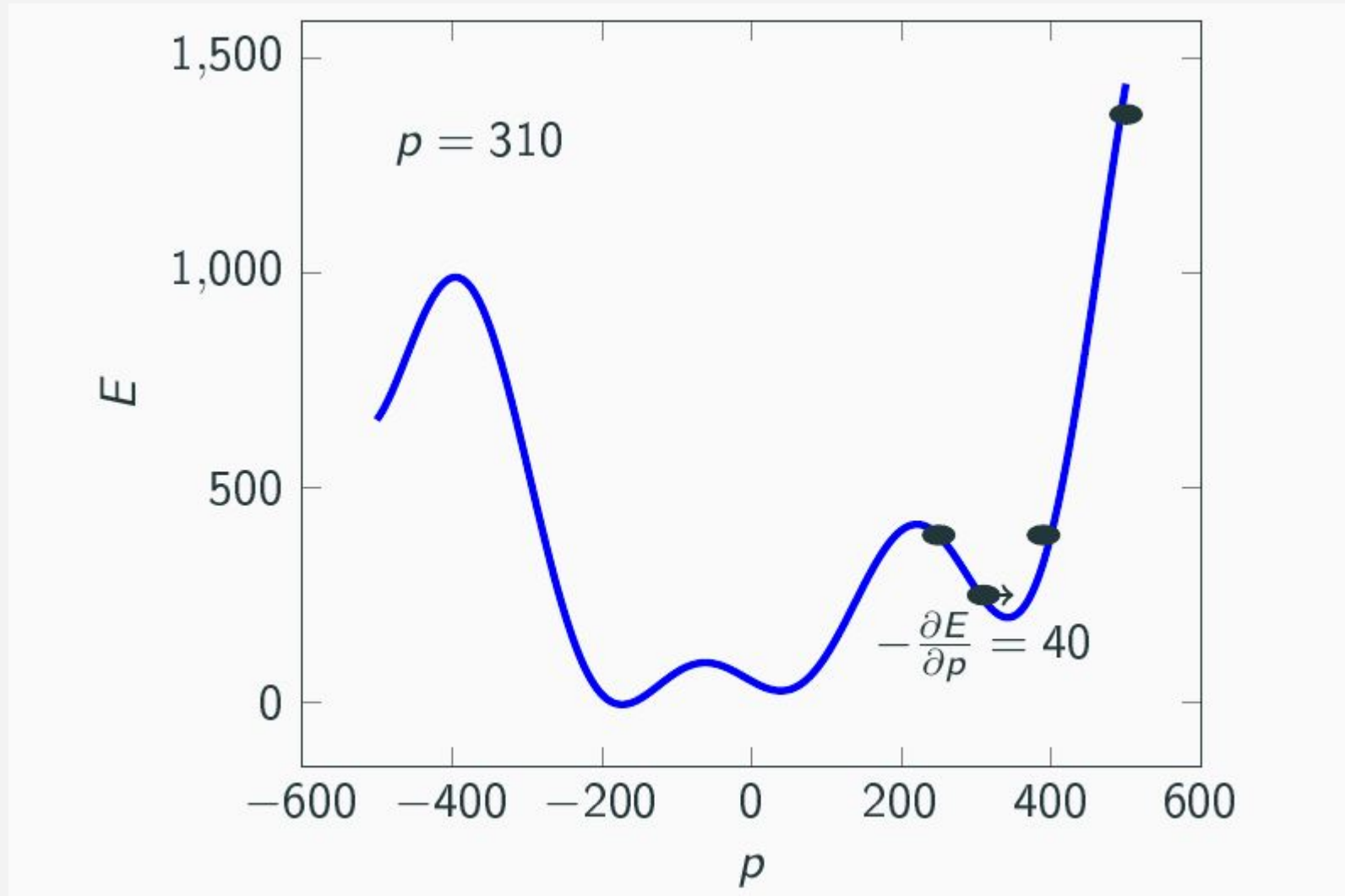
# Descenso de gradiente en 1D



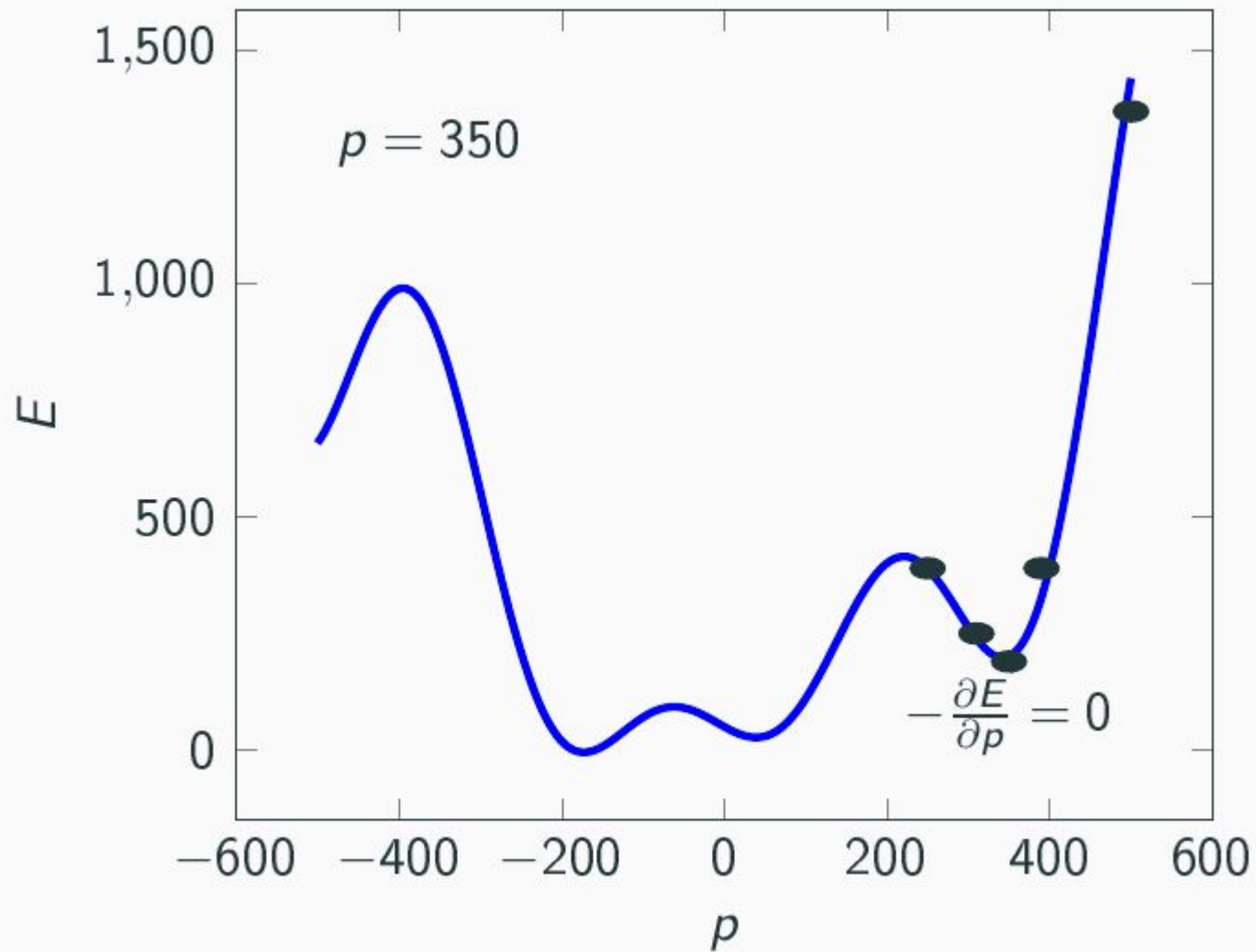
# Descenso de gradiente en 1D



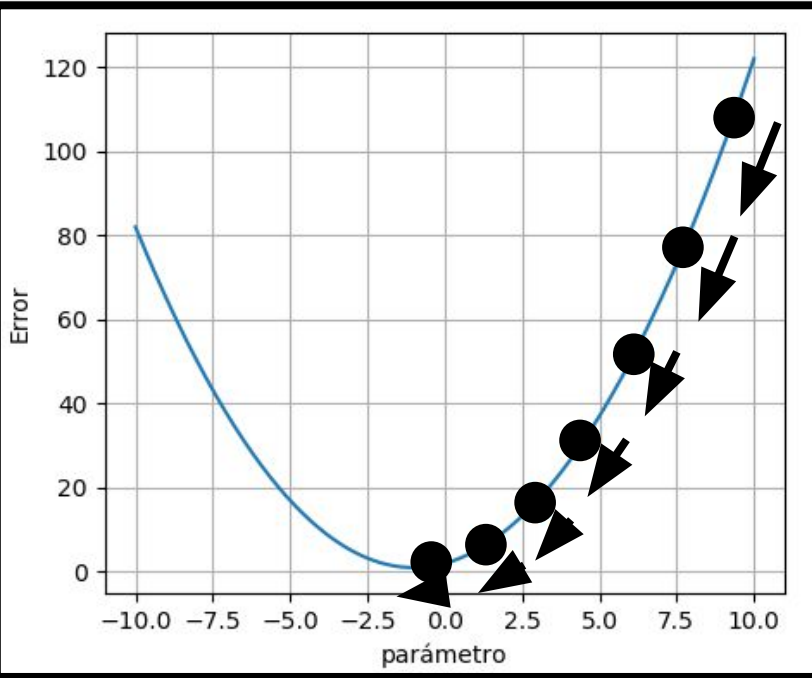
# Descenso de gradiente en 1D



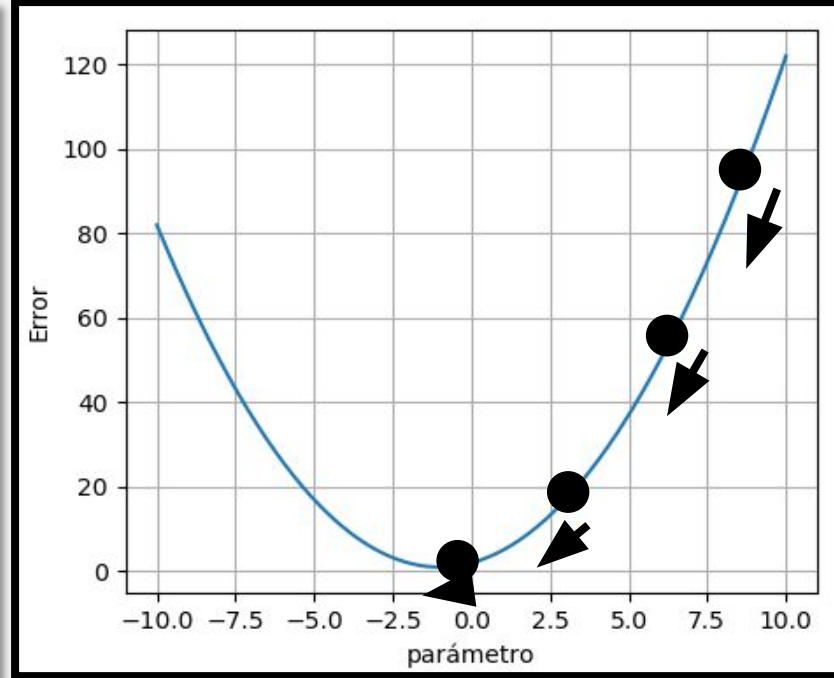
# Descenso de gradiente en 1D



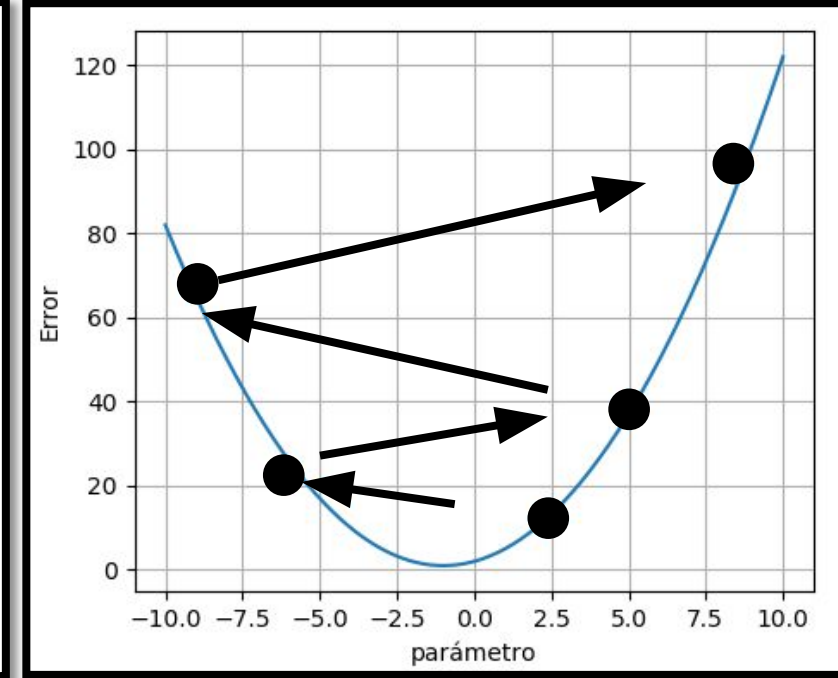
# Efecto de $\alpha$ en $w := w - \alpha (\delta E(w)/\delta w)$



- $\alpha$  demasiado chico
  - Poco avance por iteración
  - Alto coste computacional



- $\alpha$  “correcto”
  - Buen avance por iteración
  - Razonable coste computacional



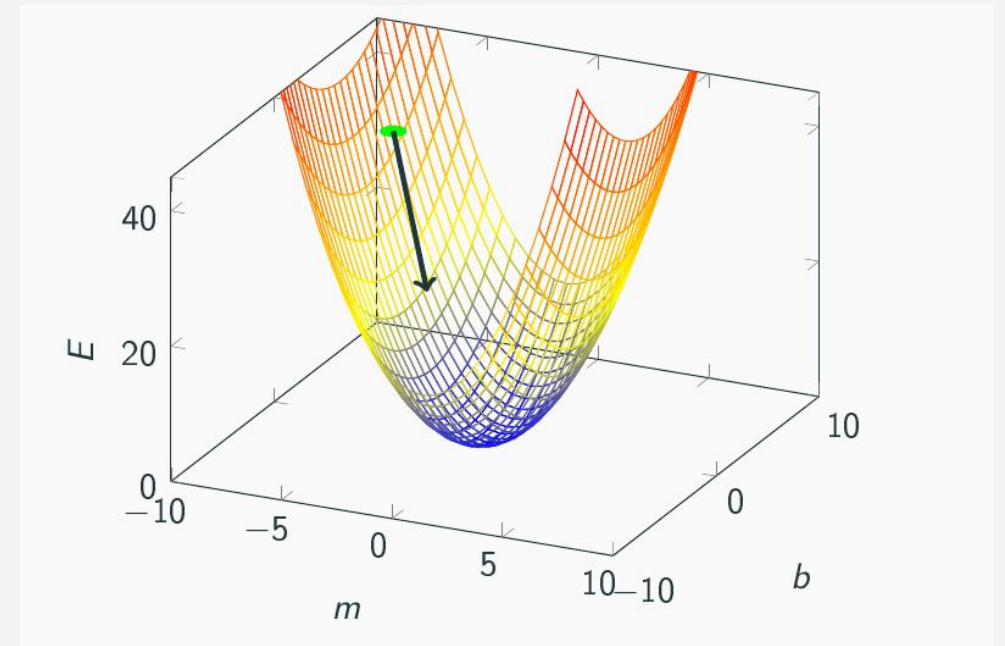
- $\alpha$  demasiado **grande**
  - “saltos” grandes
  - Puede no converger
  - Errores numéricos



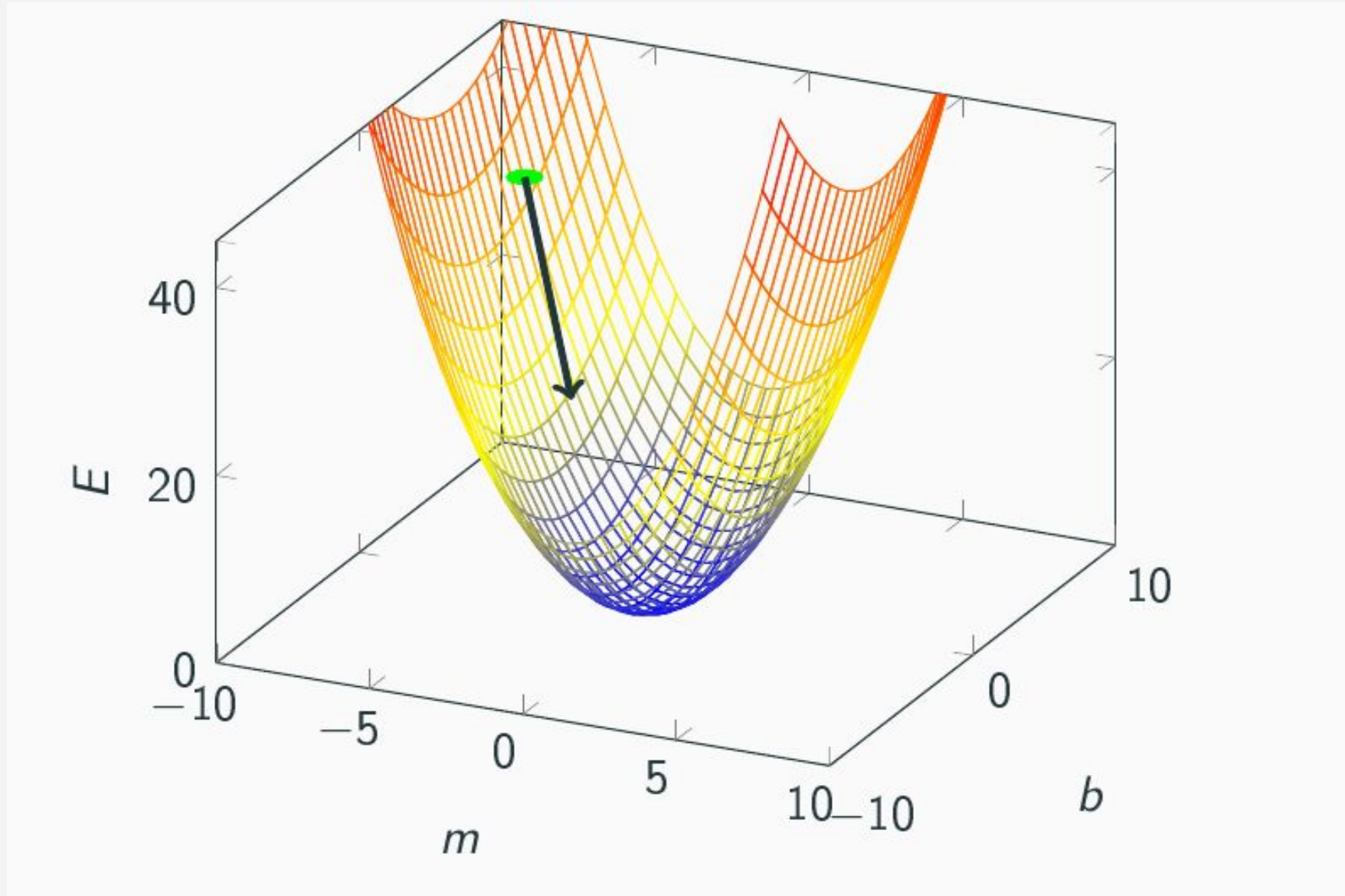
# ¿Por qué descenso de **gradiente**?

- Si tenemos 2 parámetros
  - $w_1$  y  $w_2$ 
    - 2 derivadas parciales
      - $\delta E / \delta w_1$
      - $\delta E / \delta w_2$
  - **Gradiente  $\Delta E$** 
    - Vector de derivadas parciales
    - $\Delta E = (\delta E / \delta w_1, \delta E / \delta w_2)$
- Con P parámetros
  - $w_1, w_2, \dots, w_P$
  - $\Delta E = (\delta E / \delta w_1, \delta E / \delta w_2, \dots, \delta E / \delta w_P)$

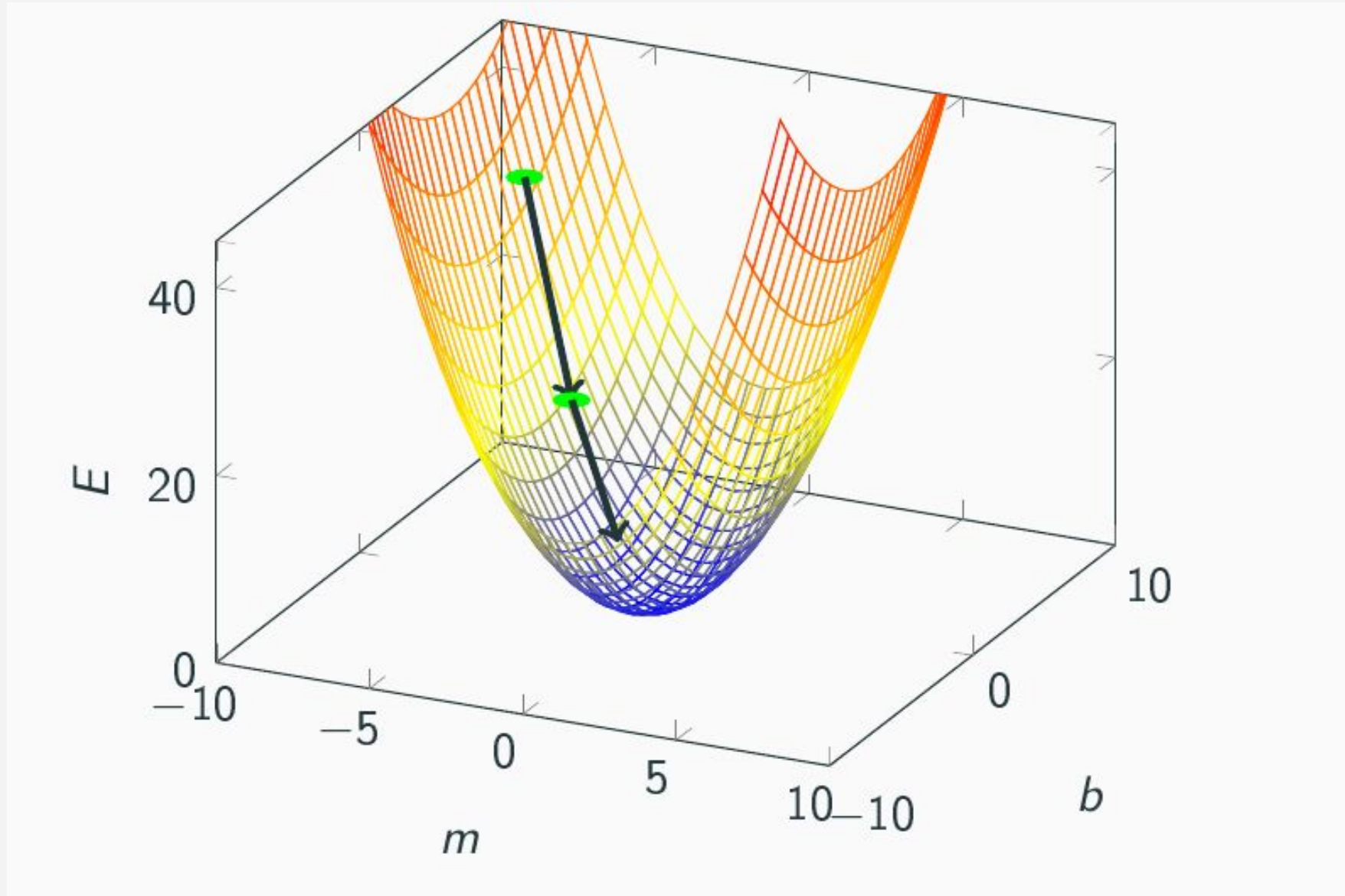
- 1 sólo parámetro
  - $w$
  - **$\Delta E = (\delta E / \delta w)$**
  - Gradiente = derivada



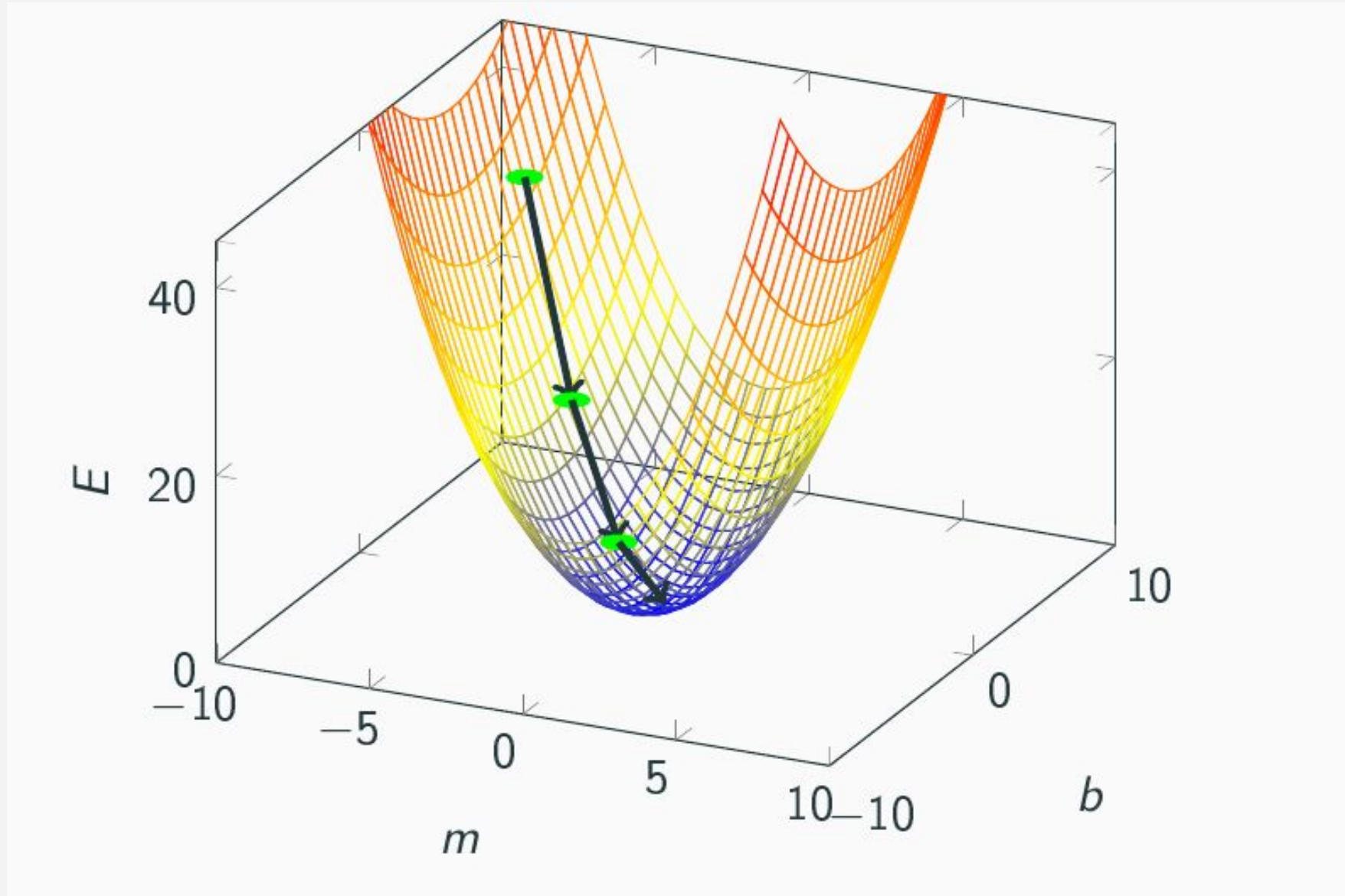
# Descenso de gradiente en 2D



# Descenso de gradiente en 2D

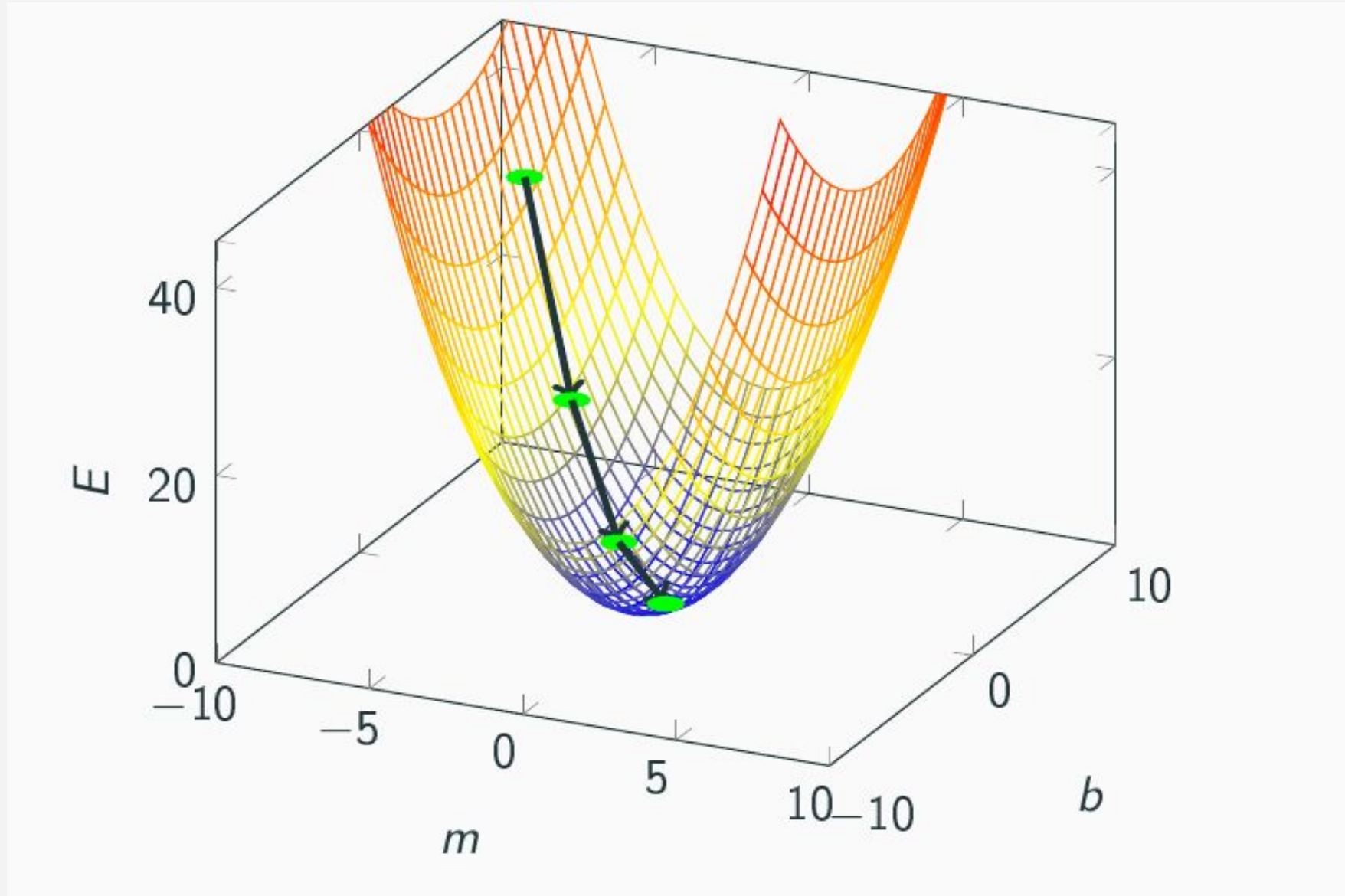


# Descenso de gradiente en 2D



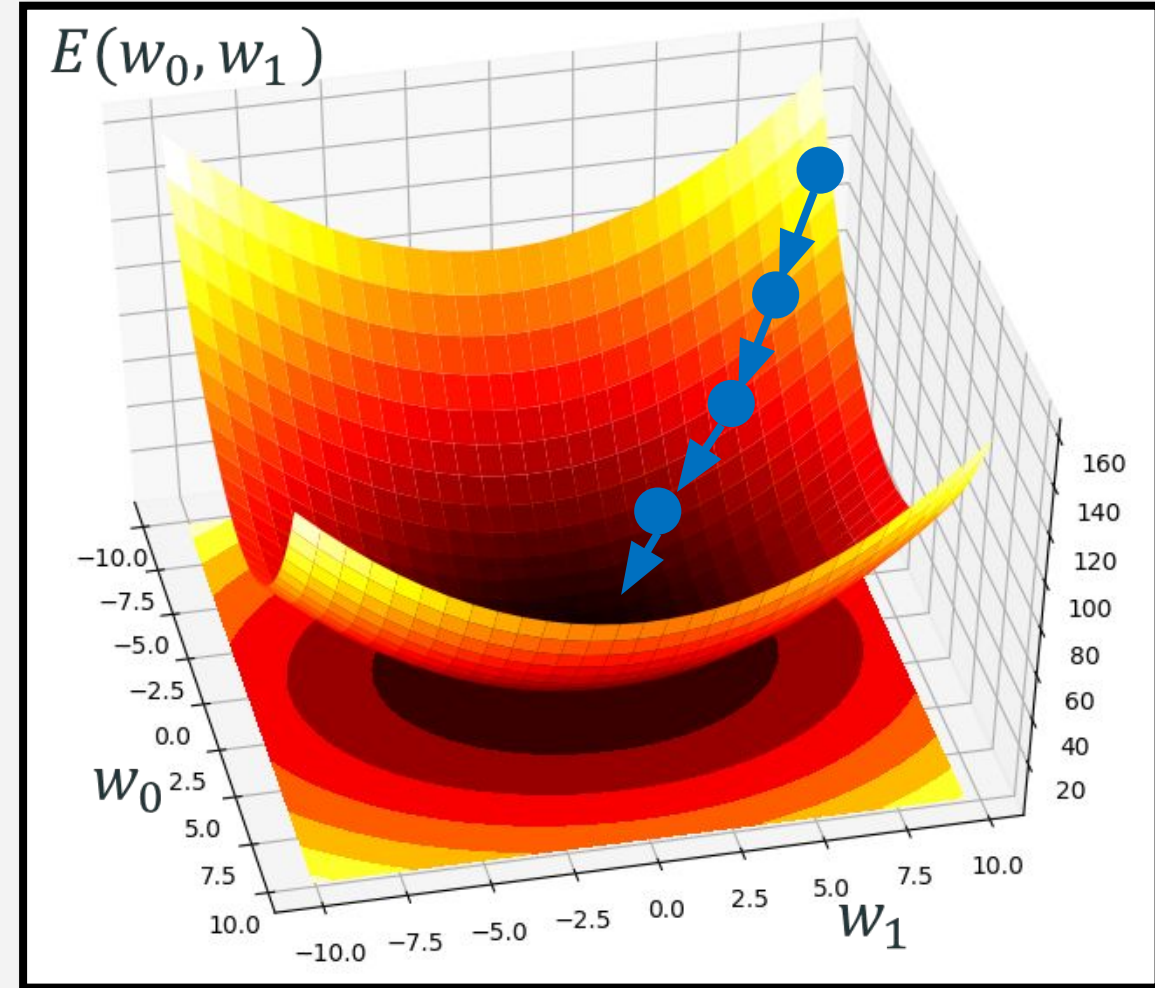


# Descenso de gradiente en 2D



# Descenso de gradiente en 2D

- 2D vs 1D
  - Mismo algoritmo
  - 2 parámetros
    - $E(w_1, w_2) = ..$
    - $\Delta \mathbf{E} = (\delta E / \delta w_1, \delta E / \delta w_2)$
- Ecuaciones
  - $w_1 = w_1 - \alpha \delta E(w_1, w_2) / \delta w_1$
  - $w_2 = w_2 - \alpha \delta E(w_1, w_2) / \delta w_2$
  - En general
    - $w_i = w_i - \alpha \delta E(w_1, w_2) / \delta w_i$



# Descenso de gradiente en ND

Iteración 1

| $x_1$ | $x_1$ | $x_1$ | $x_1$ | $y$ |
|-------|-------|-------|-------|-----|
| 2     | 2     | 2     | 2     | 1   |
| 5     | 5     | 5     | 5     | 3.2 |
| 7     | 7     | 7     | 7     | 4.5 |
| 9     | 9     | 9     | 9     | 6   |
| 10    | 10    | 10    | 10    | 4   |
| 11    | 11    | 11    | 11    | 4.5 |
| 13.4  | 13.4  | 13.4  | 13.4  | 5.5 |
| 14    | 14    | 14    | 14    | 3   |
| 15    | 15    | 15    | 15    | 5   |

$$\begin{aligned} &\delta E / \delta w_1 \\ &\delta E / \delta w_2 \\ &\dots \\ &\delta E / \delta w_p \end{aligned}$$

$$\begin{aligned} w_1 &= w_1 - \delta E / \delta w_1 \\ w_2 &= w_2 - \delta E / \delta w_2 \\ &\dots \\ w_p &= w_p - \delta E / \delta w_p \end{aligned}$$

Iteración 2

| $x_1$ | $x_1$ | $x_1$ | $x_1$ | $y$ |
|-------|-------|-------|-------|-----|
| 2     | 2     | 2     | 2     | 1   |
| 5     | 5     | 5     | 5     | 3.2 |
| 7     | 7     | 7     | 7     | 4.5 |
| 9     | 9     | 9     | 9     | 6   |
| 10    | 10    | 10    | 10    | 4   |
| 11    | 11    | 11    | 11    | 4.5 |
| 13.4  | 13.4  | 13.4  | 13.4  | 5.5 |
| 14    | 14    | 14    | 14    | 3   |
| 15    | 15    | 15    | 15    | 5   |

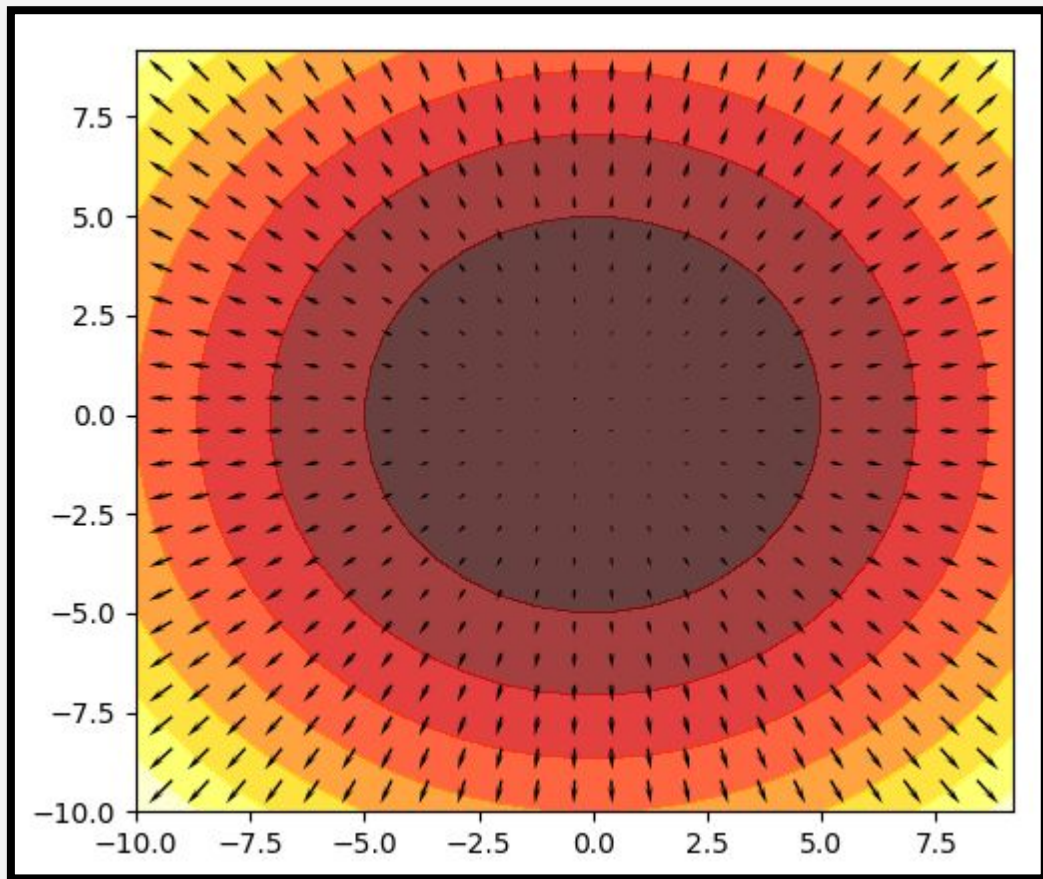
$$\begin{aligned} &\delta E / \delta w_1 \\ &\delta E / \delta w_2 \\ &\dots \\ &\delta E / \delta w_p \end{aligned}$$

$$\begin{aligned} w_1 &= w_1 - \delta E / \delta w_1 \\ w_2 &= w_2 - \delta E / \delta w_2 \\ &\dots \\ w_p &= w_p - \delta E / \delta w_p \end{aligned}$$

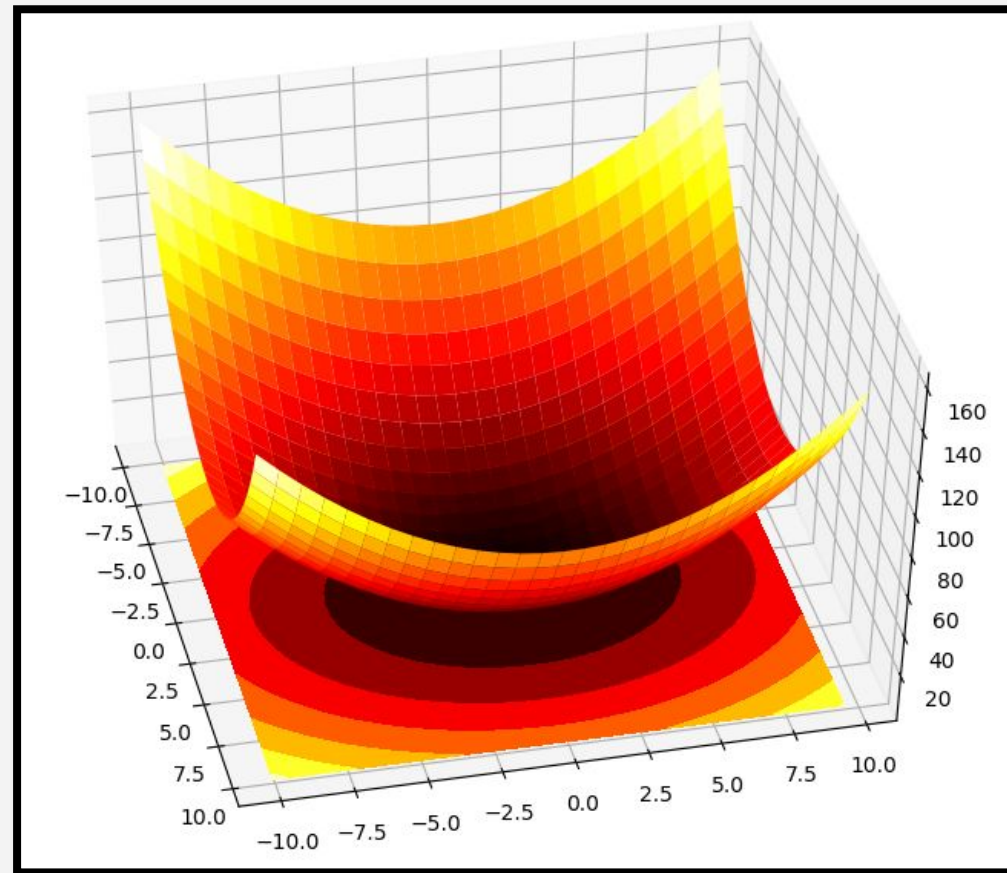
# Diagramas de contorno y gradientes

```
W1 = np.arange(-10, 10, 0.8)
W2 = np.arange(-10, 10, 0.8)
W1, W2 = np.meshgrid(W1, W2)
E = (W1**2 + W2**2)
```

- Flechas: Vectores gradiente



```
plt.contour(W1, W2, E)
dEdW1, dEdW2 = np.gradient(E)
plt.quiver(W1, W2, dEdW1, dEdW2)
```

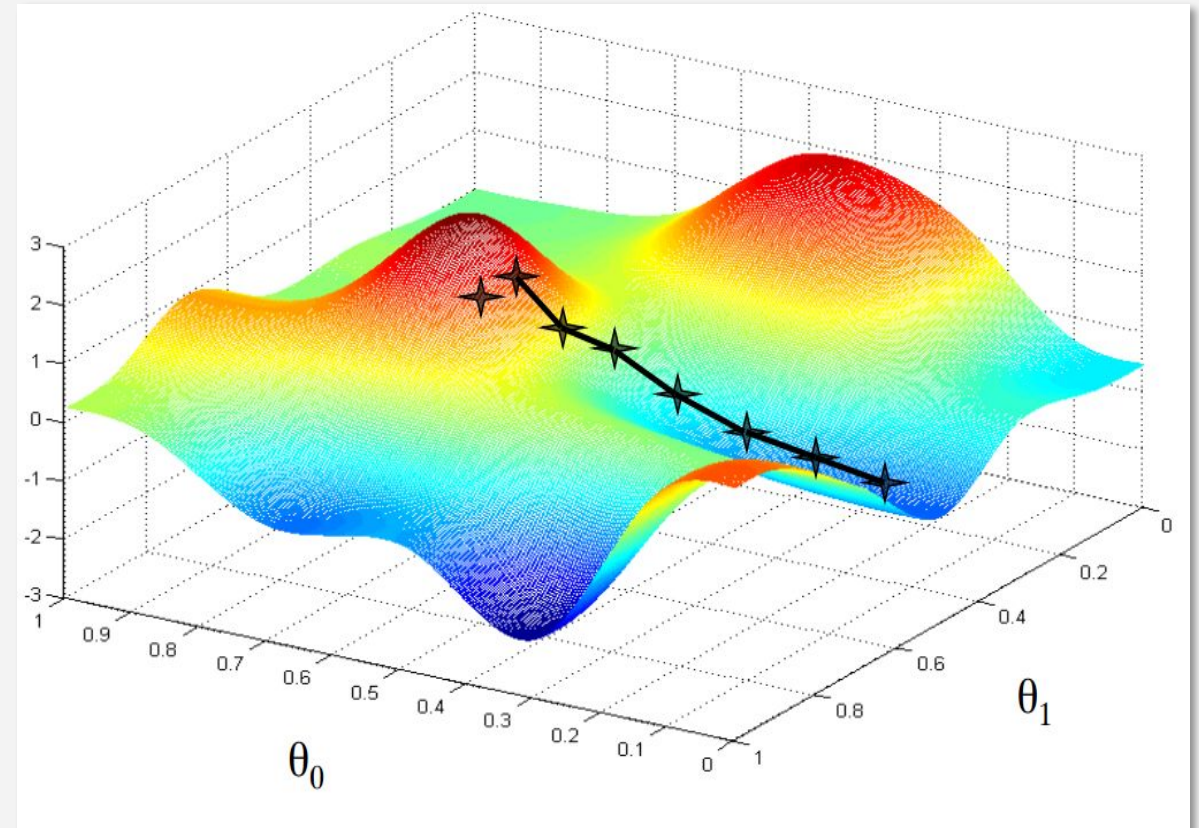
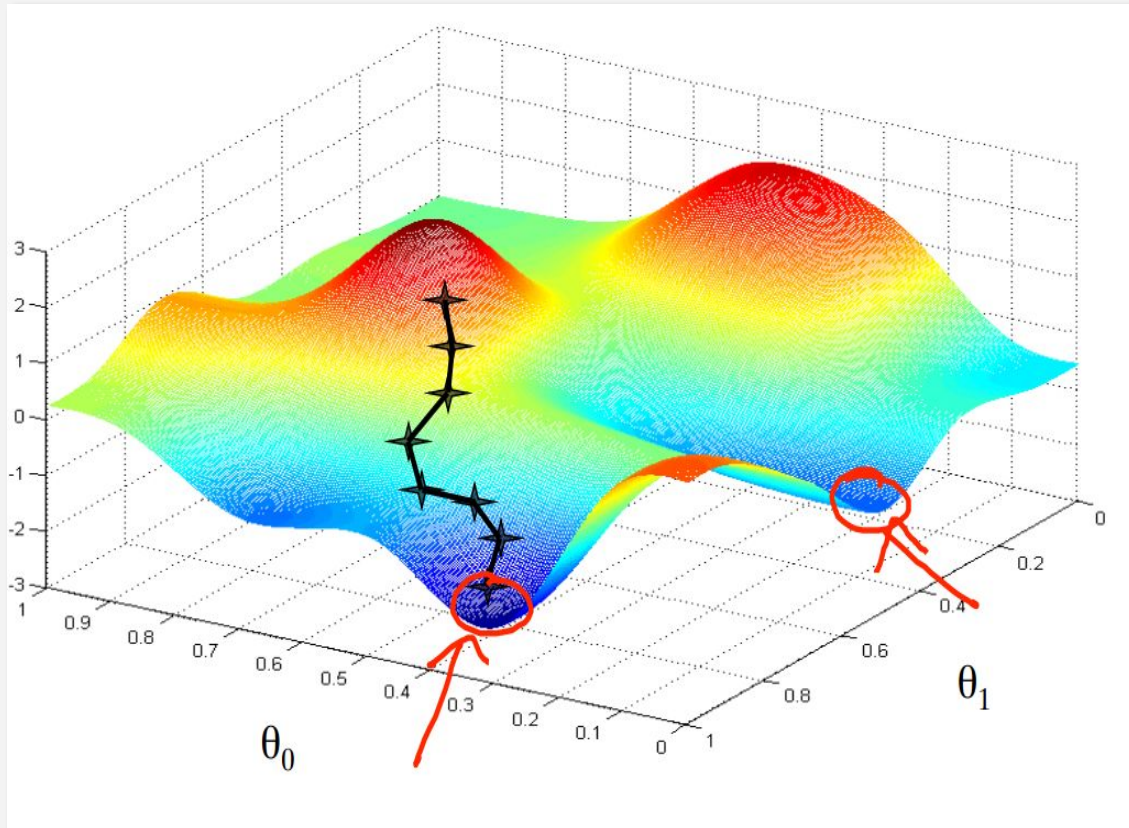


```
ax = Axes3D(fig)
ax.plot_surface(W1, W2, E)
ax.contourf(W1, W2, E)
```



# Funciones no convexas

- Redes Neuronales
  - Varían parámetros iniciales
    - Varían parámetros finales
  - Función de error no convexa → mínimos locales



# Resumen

- Descenso de gradiente
  - Iterativo
  - Generalizable
  - Requiere  $f$  derivable
  - Escalable
    - Millones de ejemplos
      - Con modificaciones
  - Ecuación **fundamental**
    - $\mathbf{w} := \mathbf{w} - \alpha (\delta E(\mathbf{w}_i) / \delta \mathbf{w}_i)$
    - Mueve a  $\mathbf{w}_i$  en la dirección que minimiza  $E$
    - $\alpha$  indica el tamaño del paso

