Introducción a la Optimización de Funciones

Entrenamiento = Aprendizaje = Optimización

- Aprendizaje Supervisado
  - Función de error **E**
  - Ejemplos con:
    - Valores de entrada $\mathbf{x_i}$
    - Valores de salida $\mathbf{y_i}$
  - Buscar ***parámetros óptimos*** en base a ***ejemplos*** y **E**

Ejemplos

Entrada	Salida
2	1
5	3.2
7	4.5
9	6

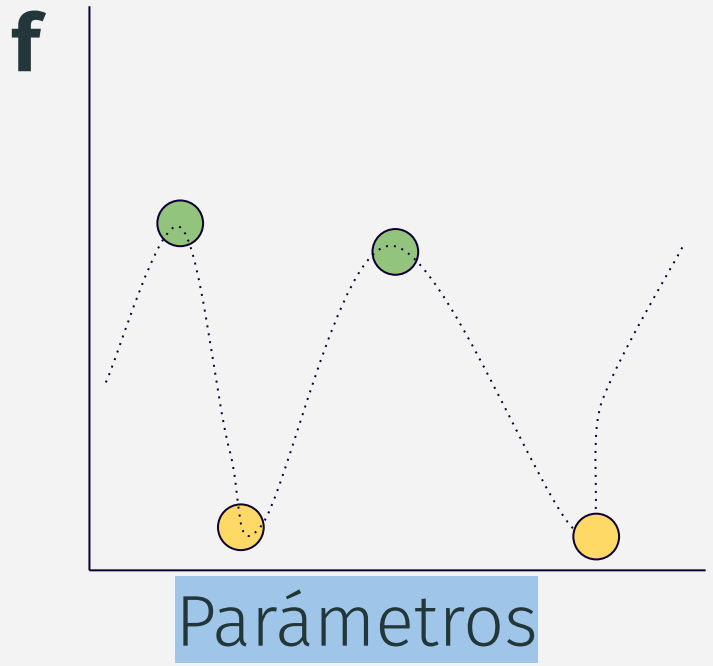
Aprendizaje = **Optimización** de **E**

Modelo

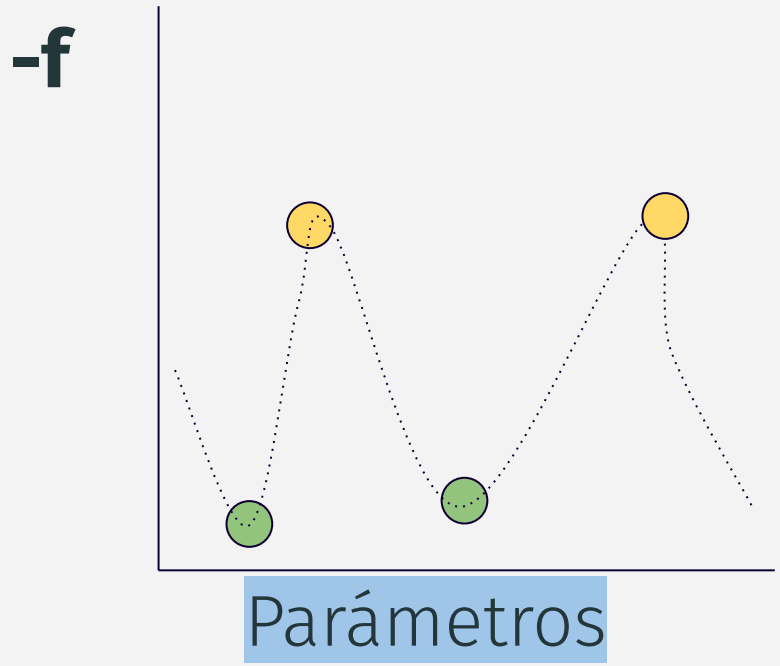
Parámetros óptimos

Optimización de funciones

- ¿Qué significa **optimizar** una función **f**?
  - Buscar **algún** mínimo
  - Variando sus parámetros

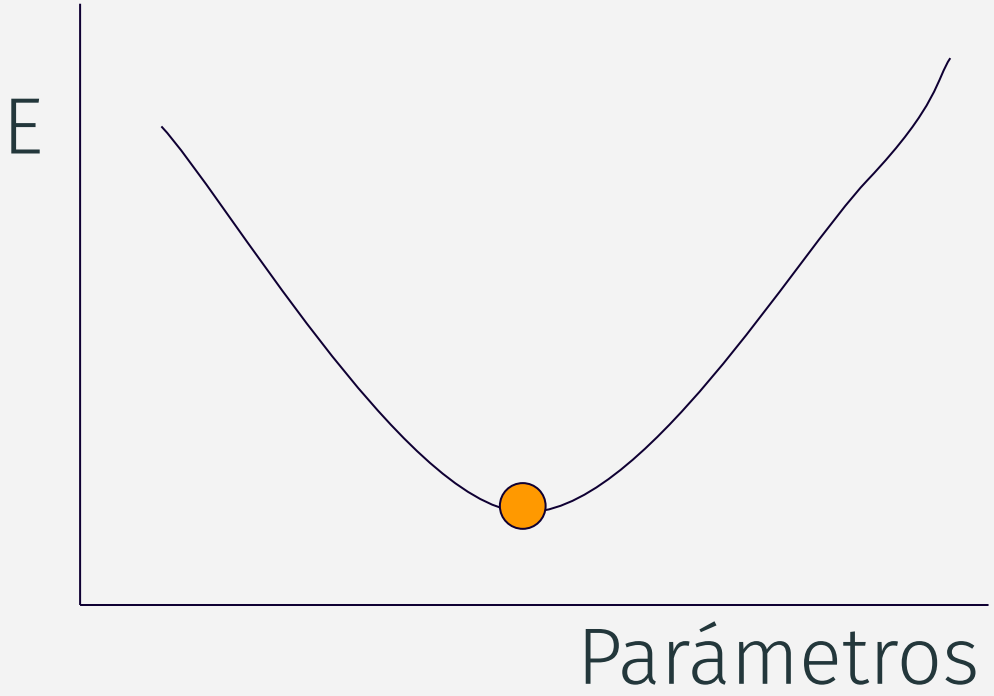


- ¿Por qué *minimizar*?
  - *Maximizar* **f** es equivalente a *minimizar* **-f**
  - Puedo buscar máximos de **f** en los mínimos de **-f**

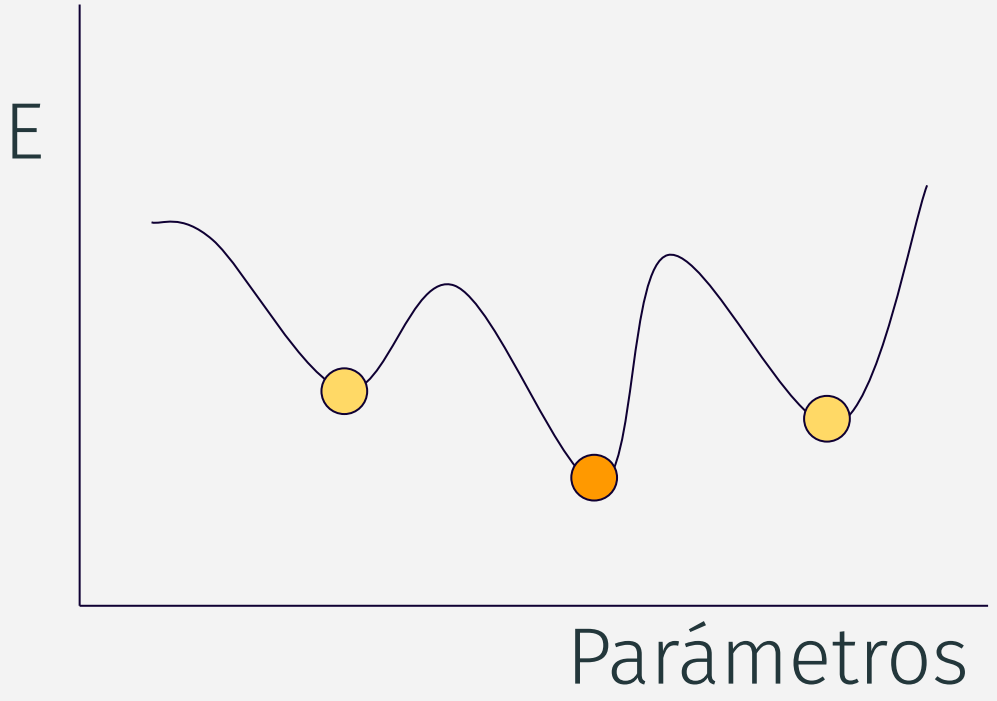


Clasificación de funciones por convexidad

- Función **convexa**
  - **Único** mínimo (global)
  - **E** para **regresión lineal** siempre es convexa :)



- Función **no-convexa**
  - **Varios** mínimos
    - Global (medio)
    - Local (otros dos)
  - **E** para **redes** es no-convexa :(

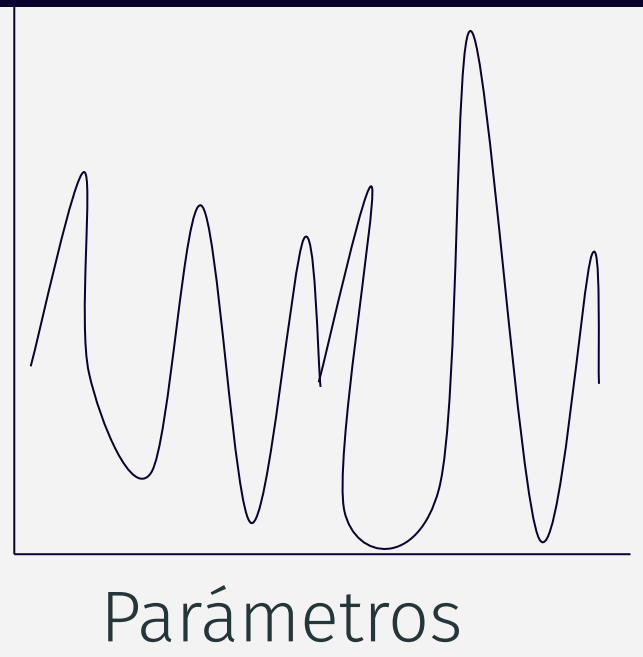


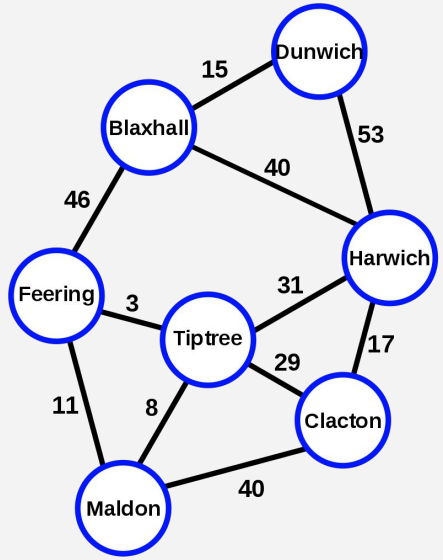
Dificultades principales

- Más de 2 parámetros (99.9% de los problemas)
  - No se puede visualizar!
    - **Algoritmos de Optimización**

Tipos de Algoritmos de Optimización

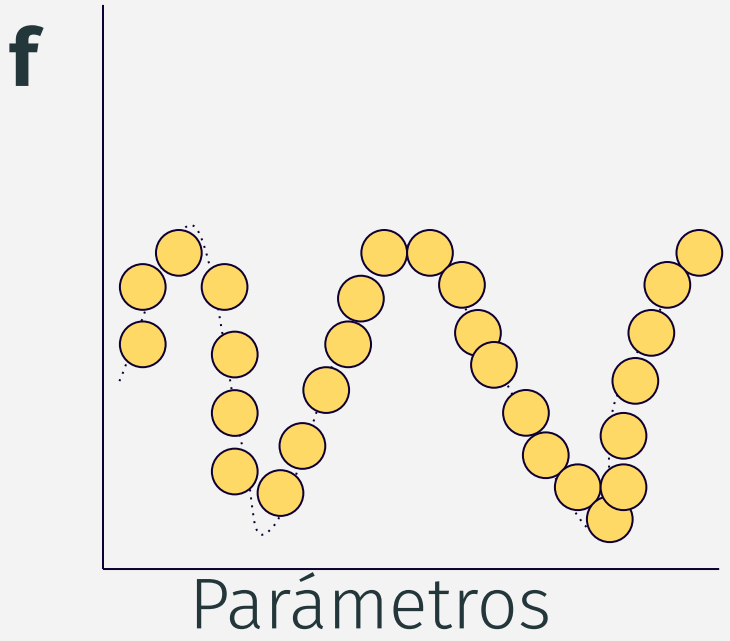
- Generales
  - Pocas asunciones sobre **f**
  - Mayor tiempo de ejecución
  - Pocas garantías
  - Ejemplos
    - Fuerza bruta
    - Búsqueda aleatoria
- Especializados
  - Básicamente, todo lo contrario
  - **Pero no siempre existen/son aplicables**
  - Ejemplo
    - Camino mínimo en un grafo



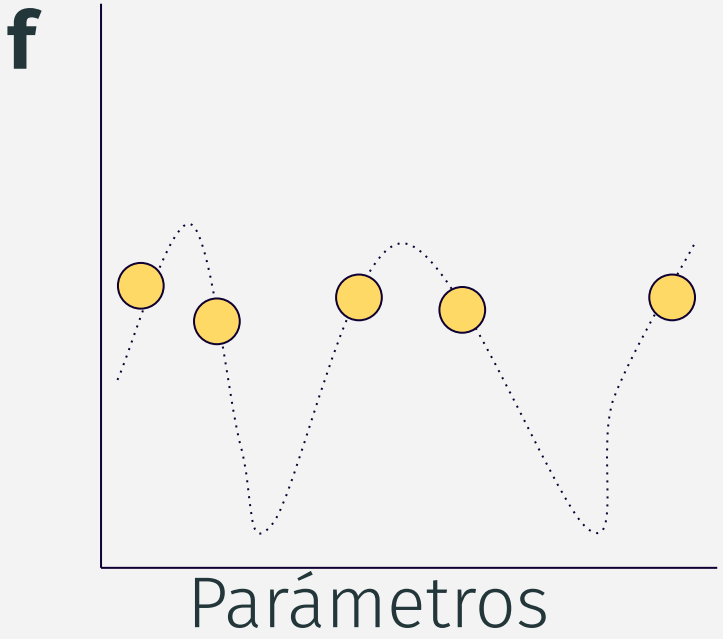


Algoritmos generales **básicos** de optimización

- Fuerza bruta
  - Probar “todos” los valores
  - Quedarse con el mejor
    - ¿Parámetros continuos?
    - ¿Muchos parámetros?

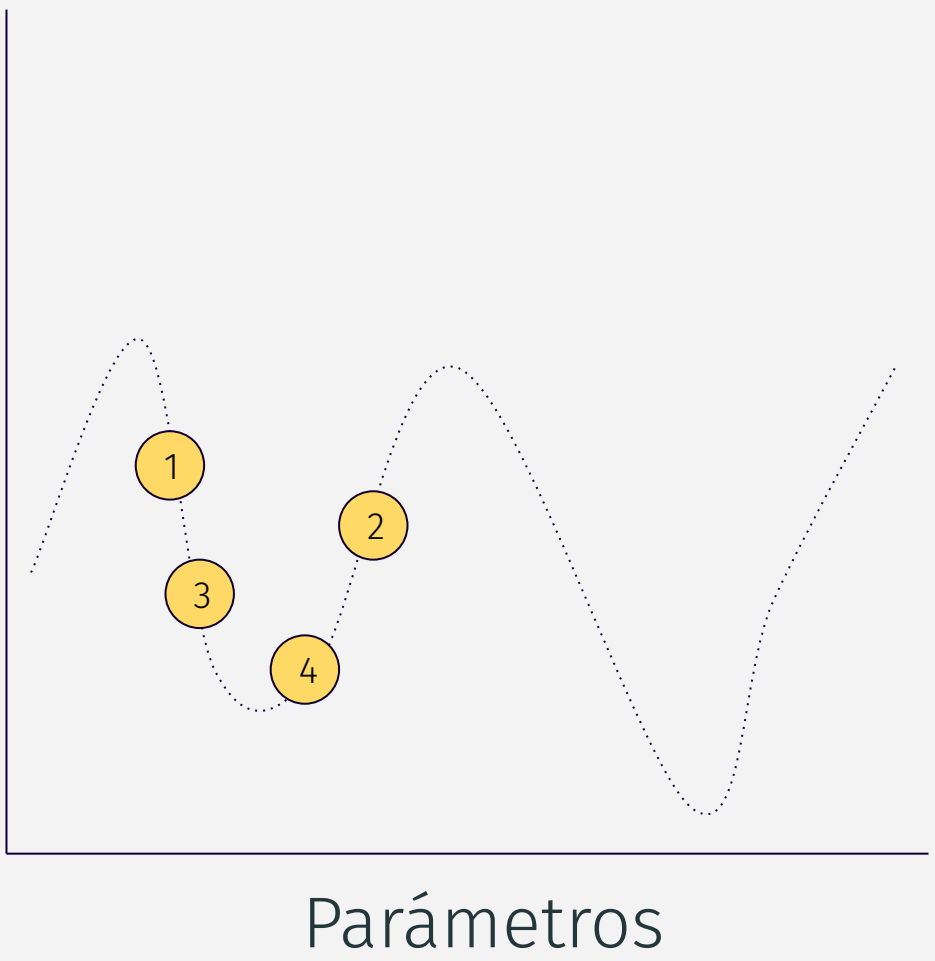


- Búsqueda aleatoria
  - Evaluar valores aleatorios
  - Quedarse con el mejor
    - ¿Cómo generar valores aleatorios?



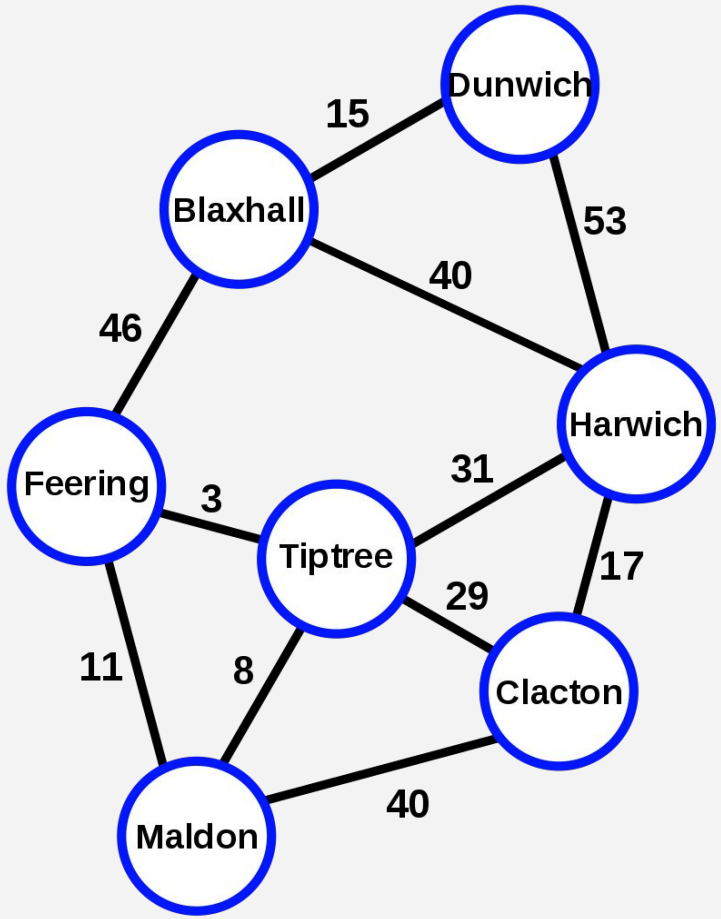
Dificultades principales

- Coste computacional de evaluar la función de error
- Criterio de convergencia
  - Convergencia = terminar
  - Explotación vs exploración
    - 4 evaluaciones
      - ¿continuar o parar?
  - Varios mínimos



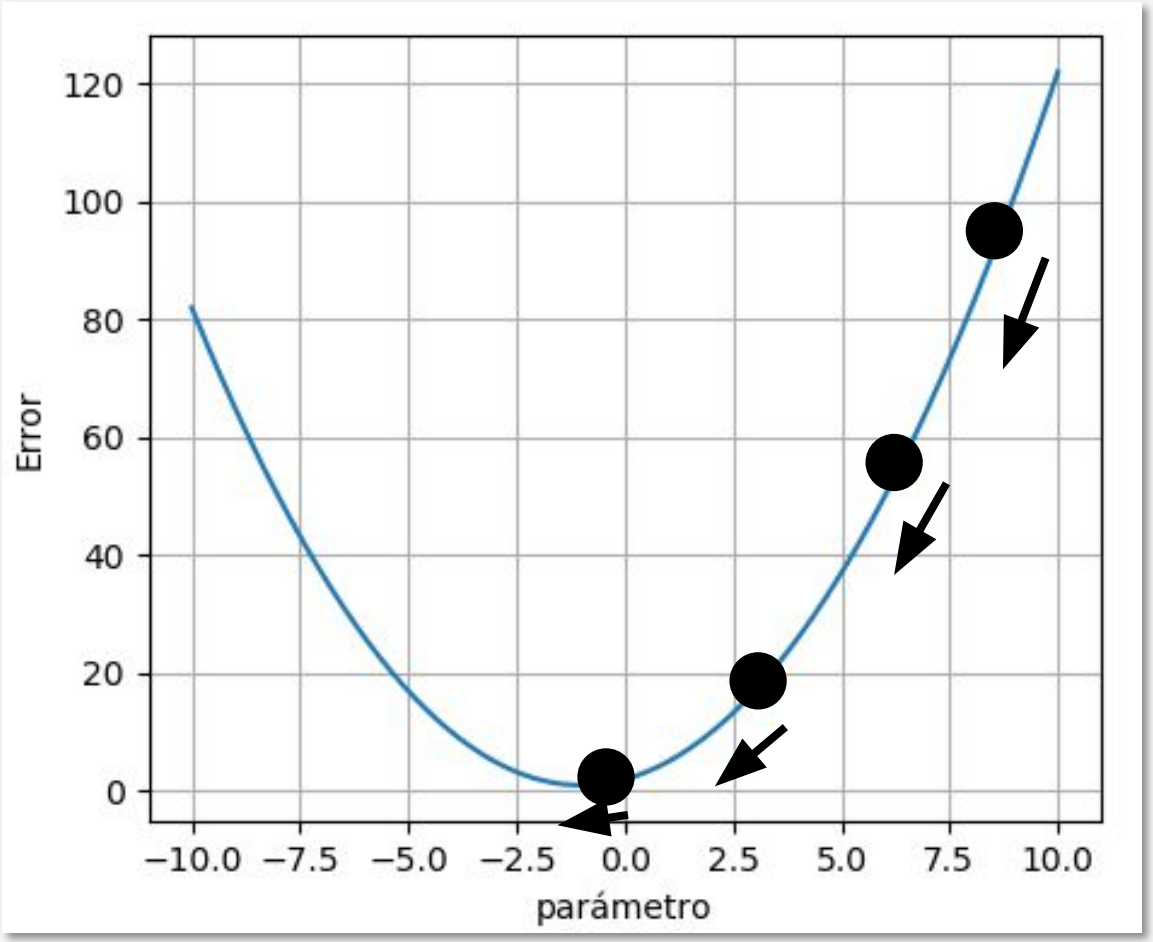
Algoritmos de Optimización Especializados

- Ejemplo: buscar el camino más corto en un grafo
  - Datos: son el grafo **G**
  - Parámetros: el camino **C** a tomar
  - Función de error: Longitud camino **L**(**C**)
    - Encontrar camino **C ∈ G**
      - Con el menor valor de **L**(**C**)
  - Solución **única**
  - *Varios* algoritmos **especializados**
    - Dijkstra
    - Bellman-Ford
    - etc..



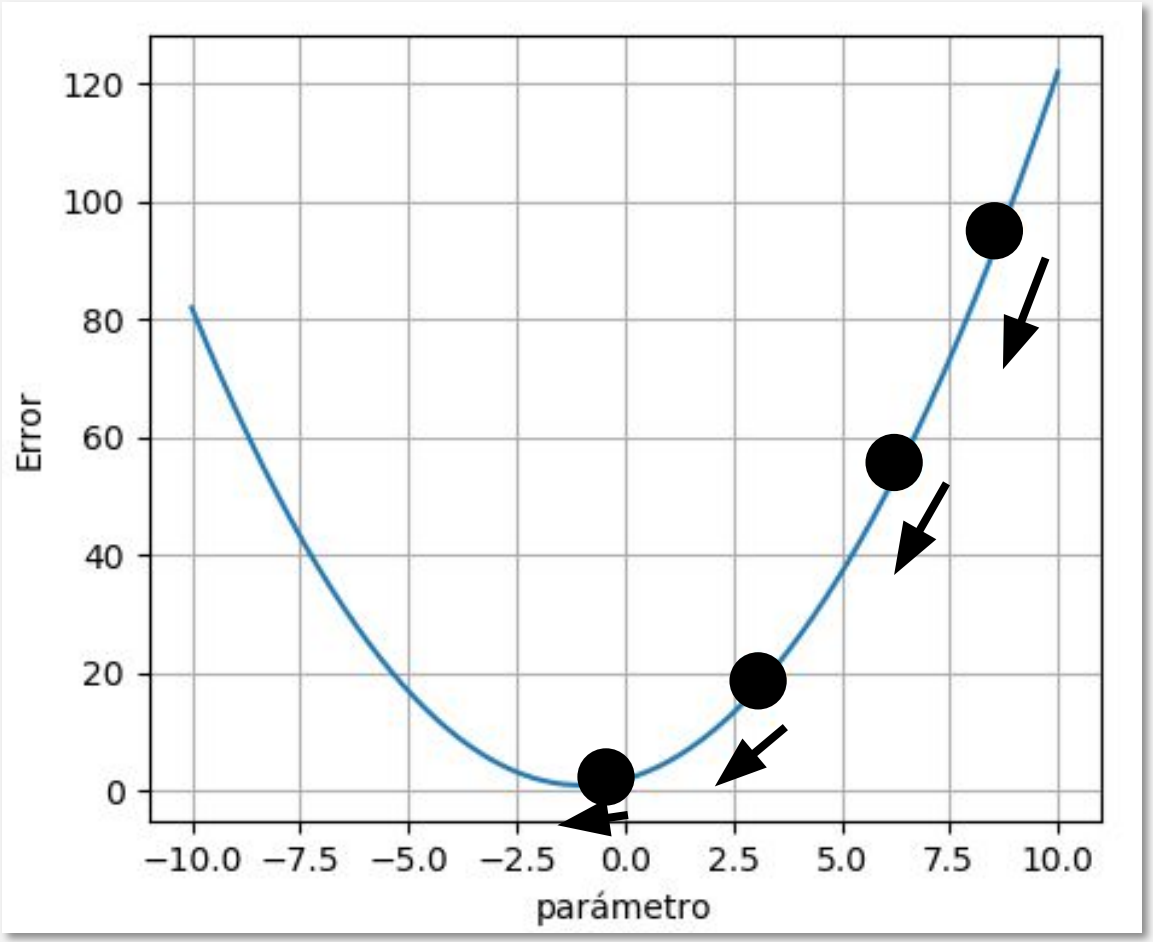
Descenso de Gradiente (DG)

- Algoritmo general
  - Asunción: **f** es **derivable**
    - Puede tener puntos de discontinuidad
  - Utilizar el **gradiente** o **derivada** para guiar la optimización
    - **Gradiente** en un punto
    - **Dirección** de crecimiento de la función
  - Problemas
    - Mínimos locales
  - Escalable a n > millones



Descenso de gradiente (DG) : idea básica

- Empezar en un **valor aleatorio** del parámetro
  - Iterar hasta que derivada=0
    - Calcular **gradiente** o **derivada**
      - Indica dirección de maximización
    - Moverse en dirección opuesta
- Valor final: mínimo local (posiblemente global)



Resumen

- ¿Qué significa **optimizar** una función **f**?
  - Buscar **algún** mínimo
  - Variando sus parámetros
- Algoritmos de Optimización
  - Analíticos vs Iterativos
  - Generales vs Especializados
  - Criterios de convergencia
  - Coste computacional
- Descenso de gradiente
  - Más utilizado para Redes Neuronales