



UNIVERSITY OF  
BATH

## Tópicos en Inteligencia Artificial Deep Learning

### Algoritmos de optimización

Basado en el [ppt 06 – Fitting](#) de  
Prof. Simon Prince  
Adaptado por Prof. Fernando Crema García



# Regresión

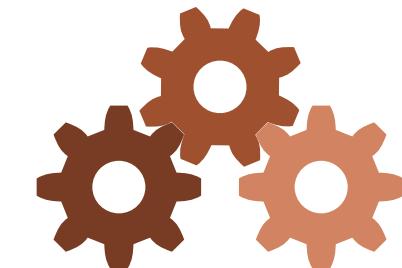
Real world input

6000 square feet,  
4 bedrooms,  
previously sold for  
\$235K in 2005,  
1 parking spot.

Model  
input

$$\begin{bmatrix} 6000 \\ 4 \\ 235 \\ 2005 \\ 1 \end{bmatrix}$$

Model



Supervised learning  
model

Model  
output

$$[340]$$

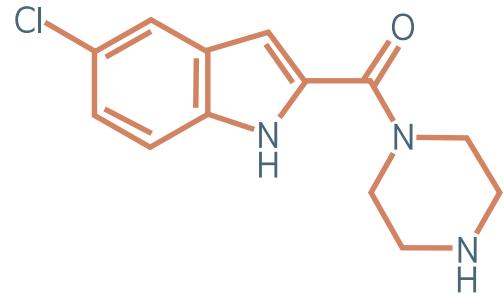
Real world output

Predicted price  
is \$340k

- Problema de regresión univariante (un resultado, valor real)

# Regresión gráfica

Real world input



Model input

$$\begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ 17 \\ 1 \\ 1 \\ \vdots \end{bmatrix}$$

Model



Model output

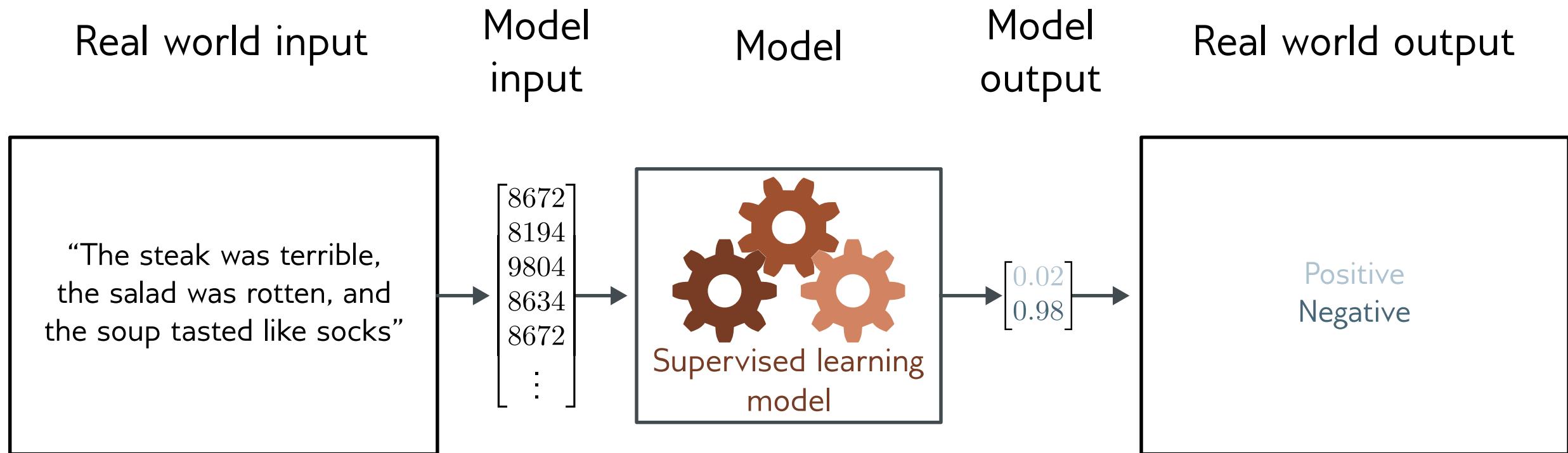
$$\begin{bmatrix} -12.9 \\ 56.4 \end{bmatrix}$$

Real world output

Freezing point  
is  $-12.9^{\circ}\text{C}$   
Boiling point  
is  $56.4^{\circ}\text{C}$

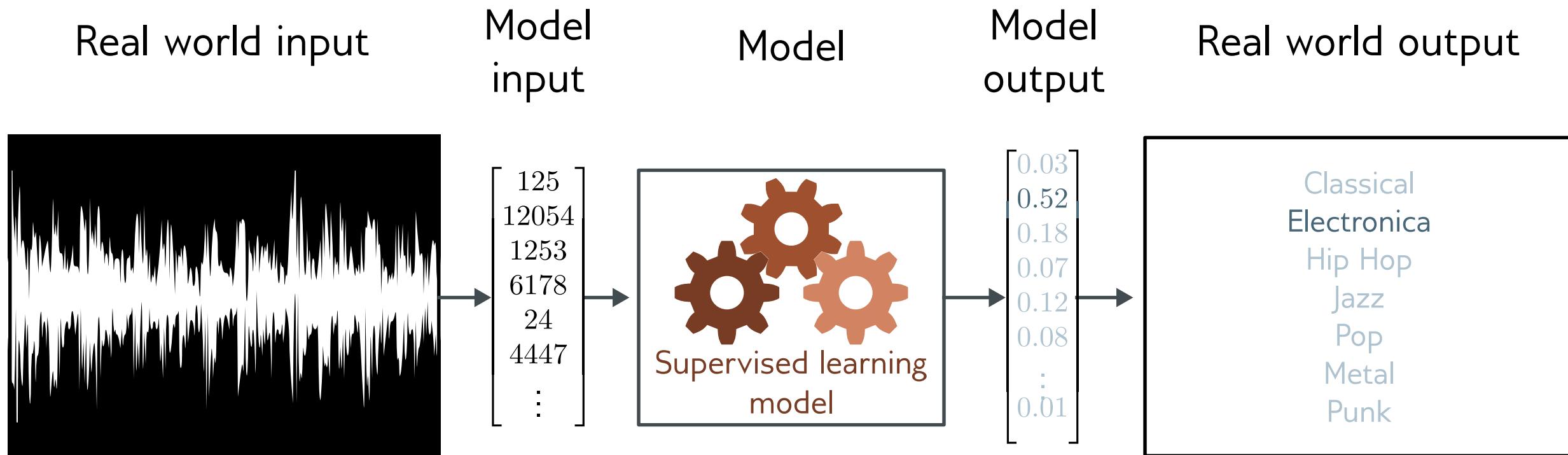
- Problema de regresión multivariante ( $>1$  salida, valor real)

# Clasificación de textos



- Problema de clasificación binaria (dos clases discretas)

# Clasificación de los géneros musicales



- Problema de clasificación multiclas (clases discretas, >2 valores posibles)

# Función de pérdida

- Conjunto de datos de entrenamiento de  $I$  pares de ejemplos de entrada/salida:

$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^I$$

- La función de pérdida o función de coste mide lo malo que es el modelo:

$$L[\phi, f[\mathbf{x}_i, \phi], \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^I]$$

o para abreviar:

$$L[\phi]$$

Devuelve un escalar que es más pequeño cuando el modelo asigna mejor las entradas a las salidas

# Formación

- Función de pérdida:

$$L[\phi]$$

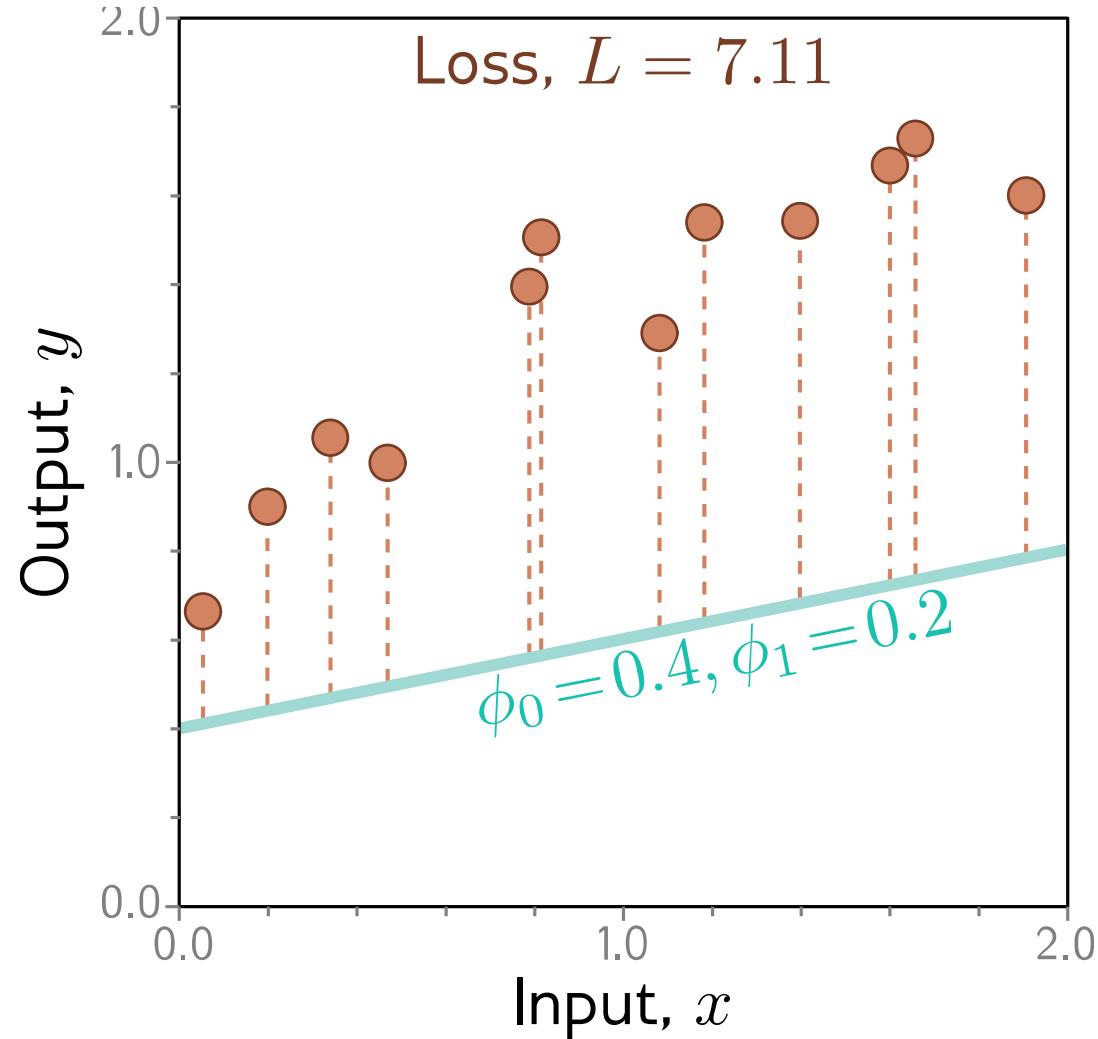


Devuelve un escalar que es más pequeño cuando el modelo asigna mejor las entradas a las salidas

- Encuentra los parámetros que minimizan la pérdida:

$$\hat{\phi} = \operatorname{argmin}_{\phi} [L[\phi]]$$

# Ejemplo: función de pérdida de regresión lineal 1D



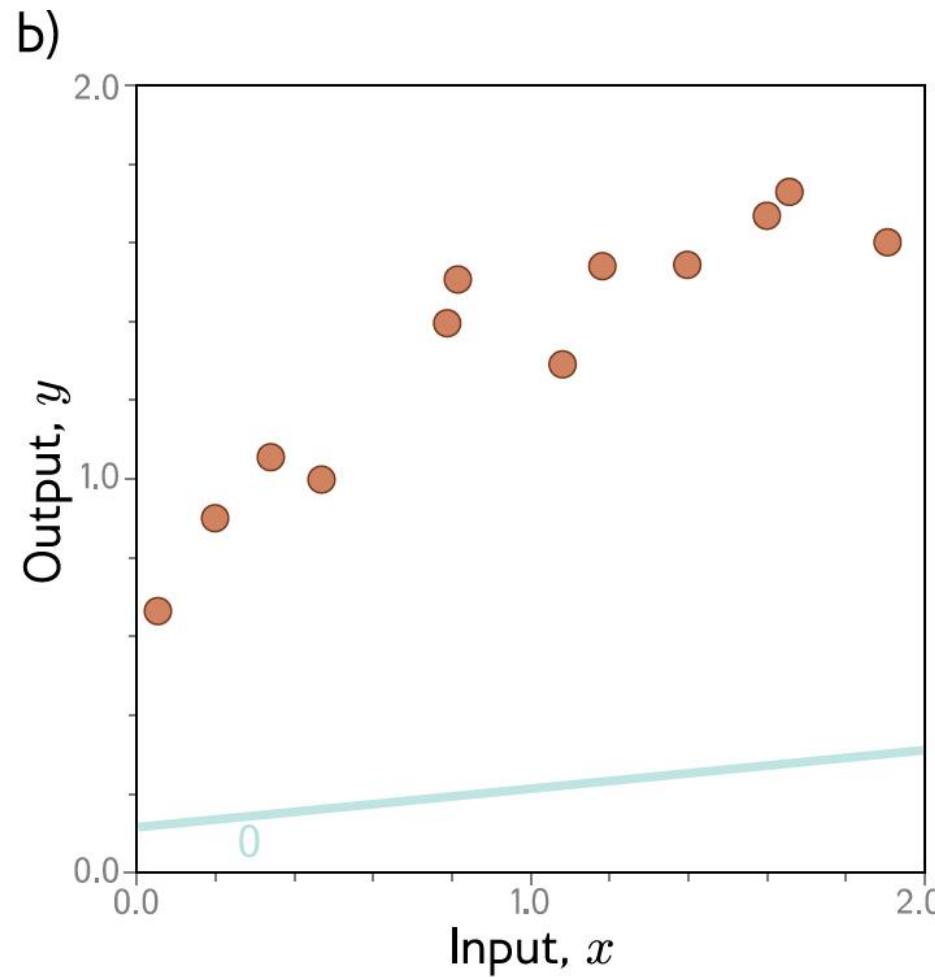
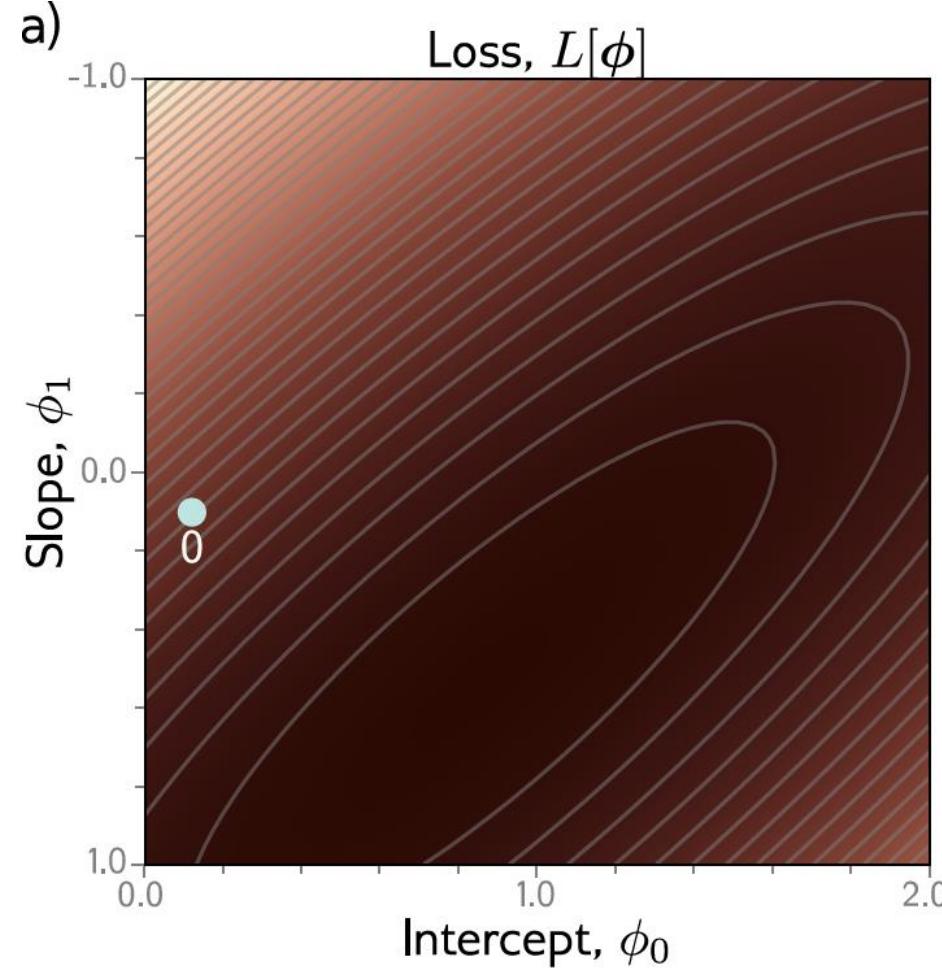
Función de  
pérdida:

$$\begin{aligned} L[\phi] &= \sum_{i=1}^I (f[x_i, \phi] - y_i)^2 \\ &= \sum_{i=1}^I (\phi_0 + \phi_1 x_i - y_i)^2 \end{aligned}$$

"Función de pérdida por  
mínimos cuadrados"

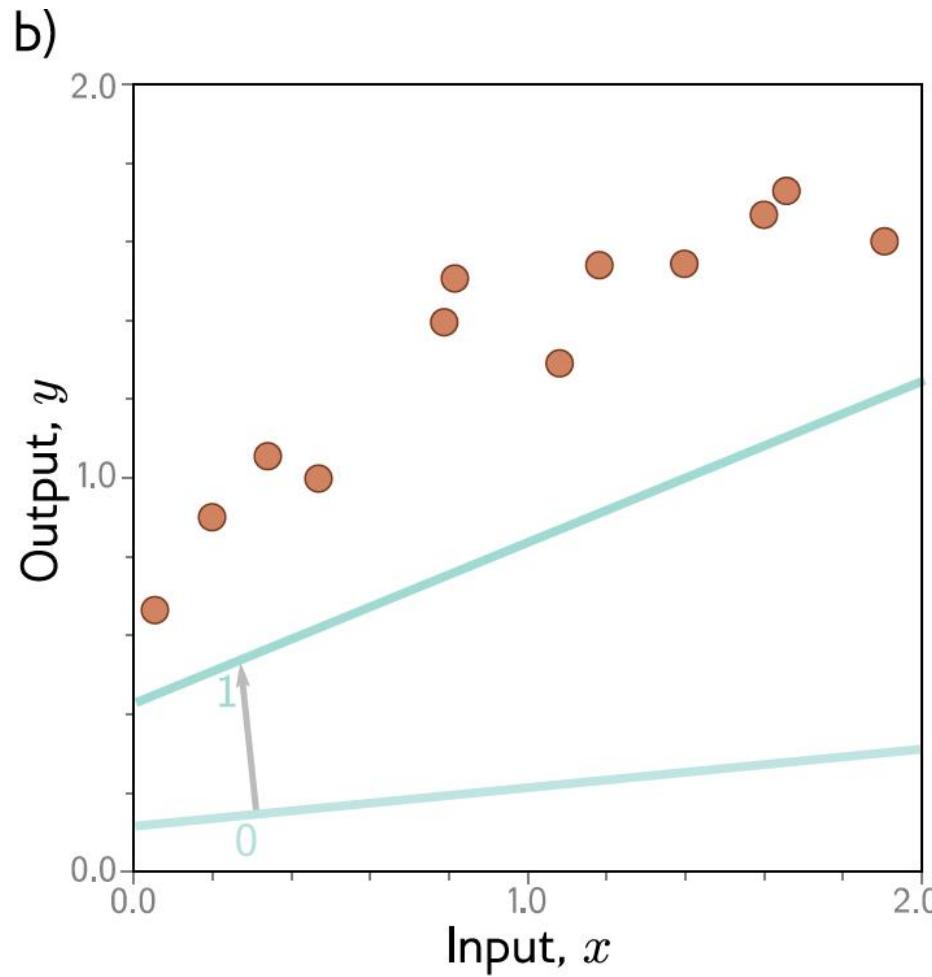
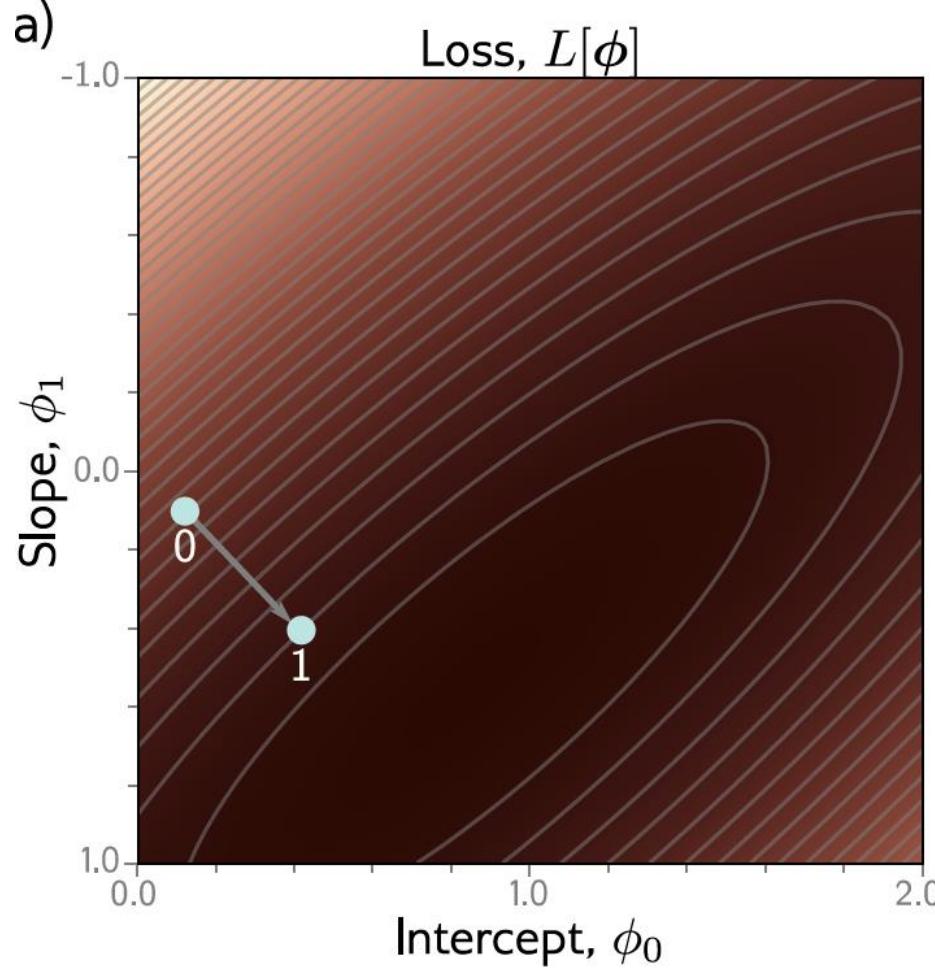
# Ejemplo: entrenamiento de regresión lineal

1D



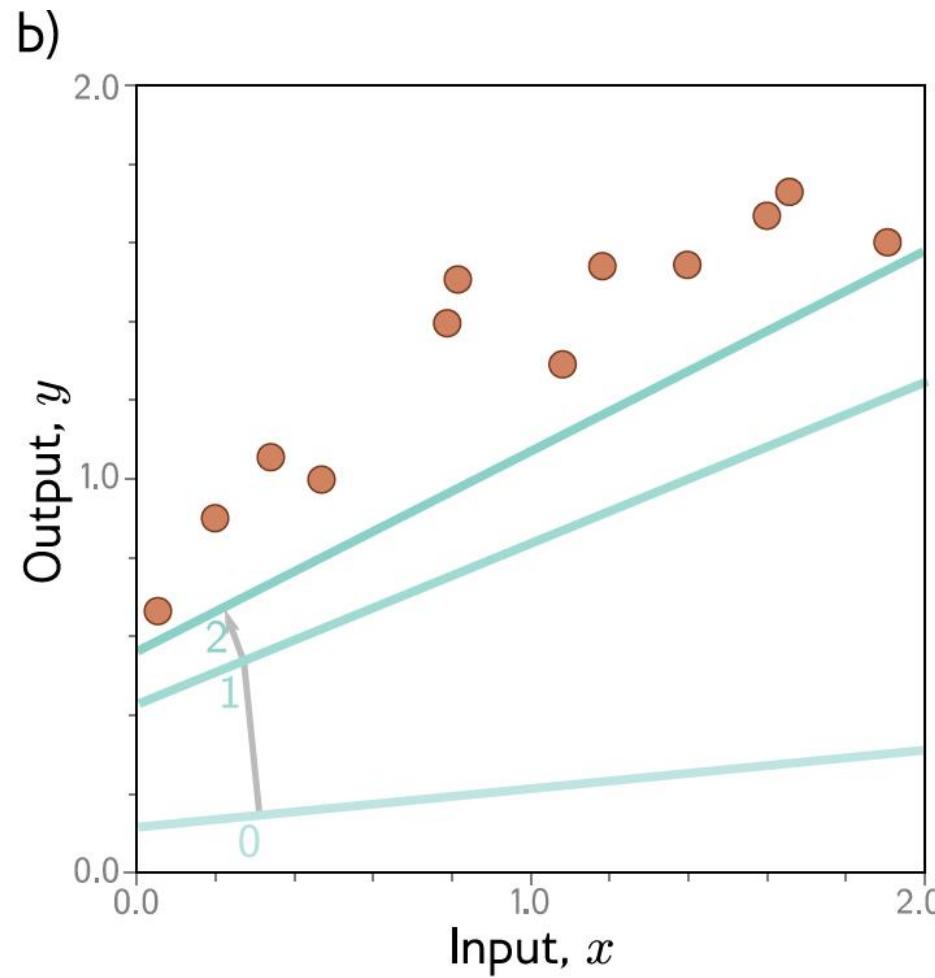
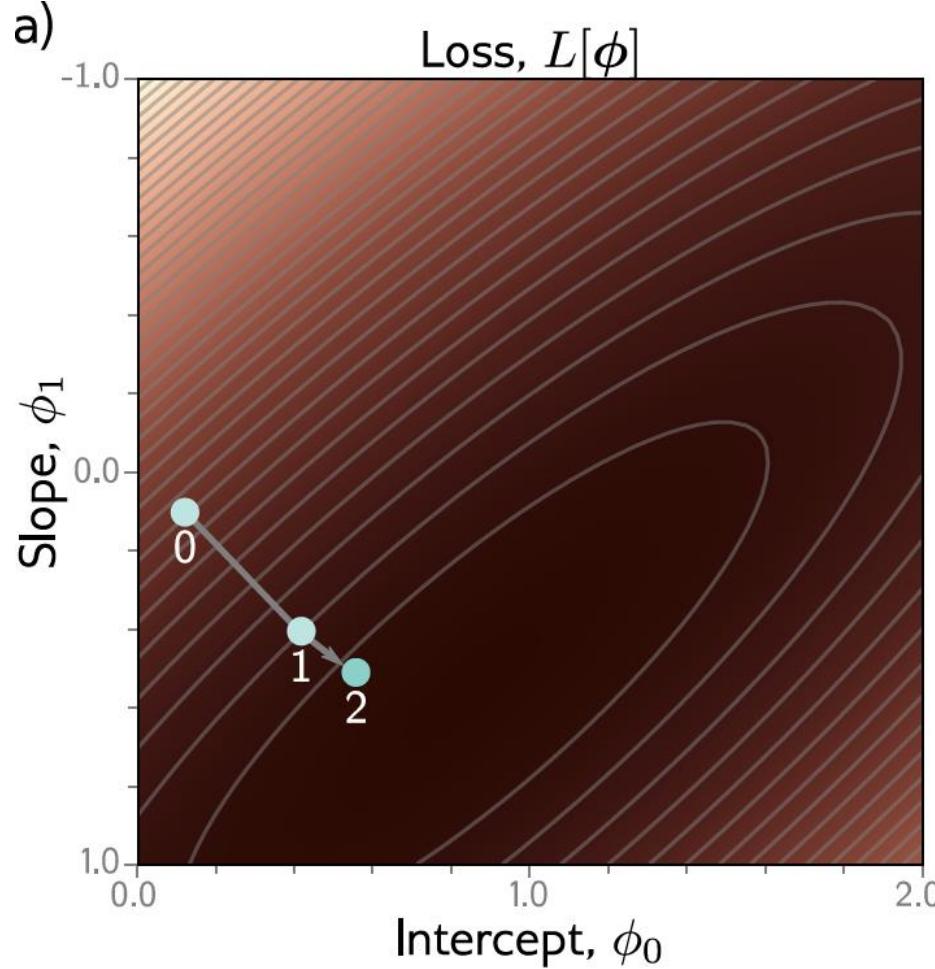
# Ejemplo: entrenamiento de regresión lineal

1D



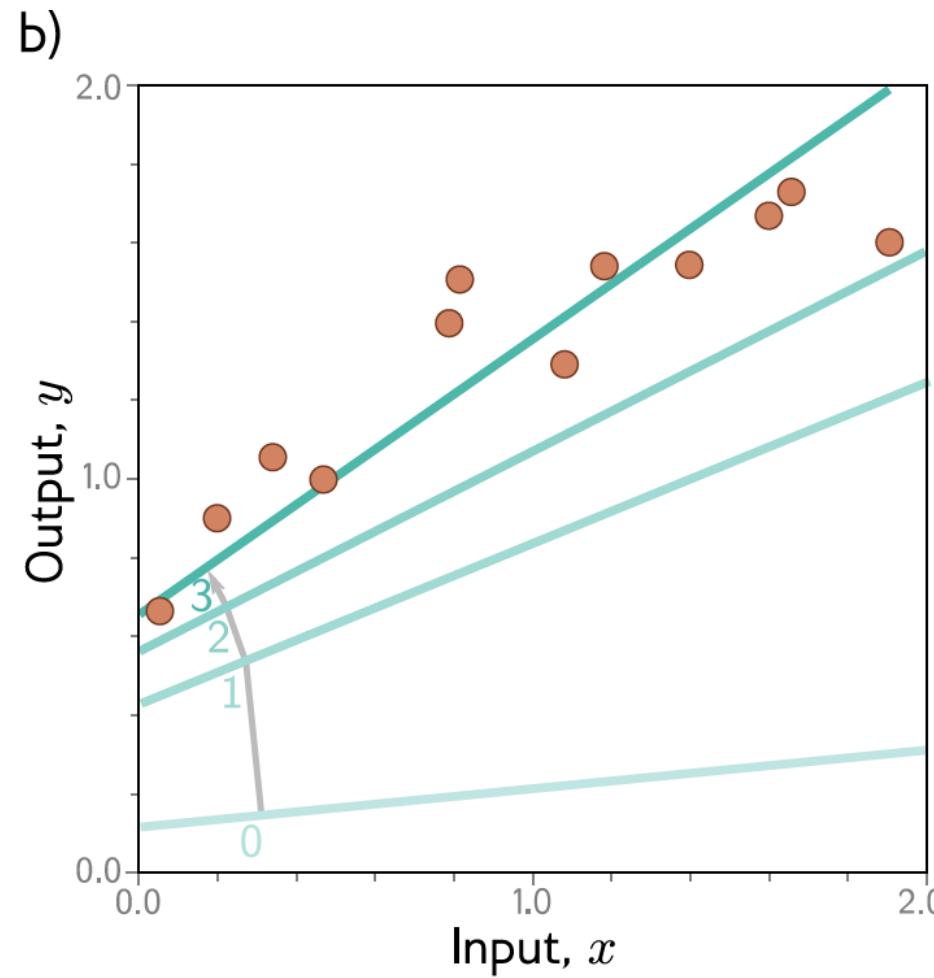
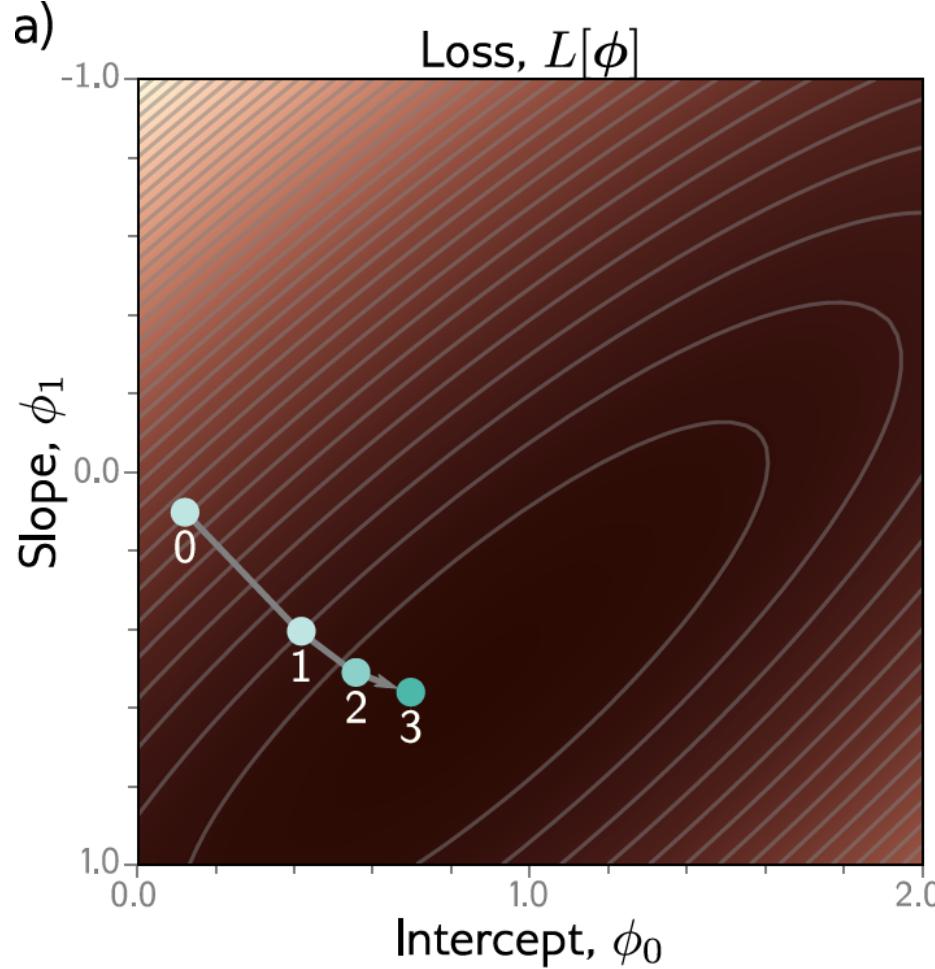
# Ejemplo: entrenamiento de regresión lineal

1D



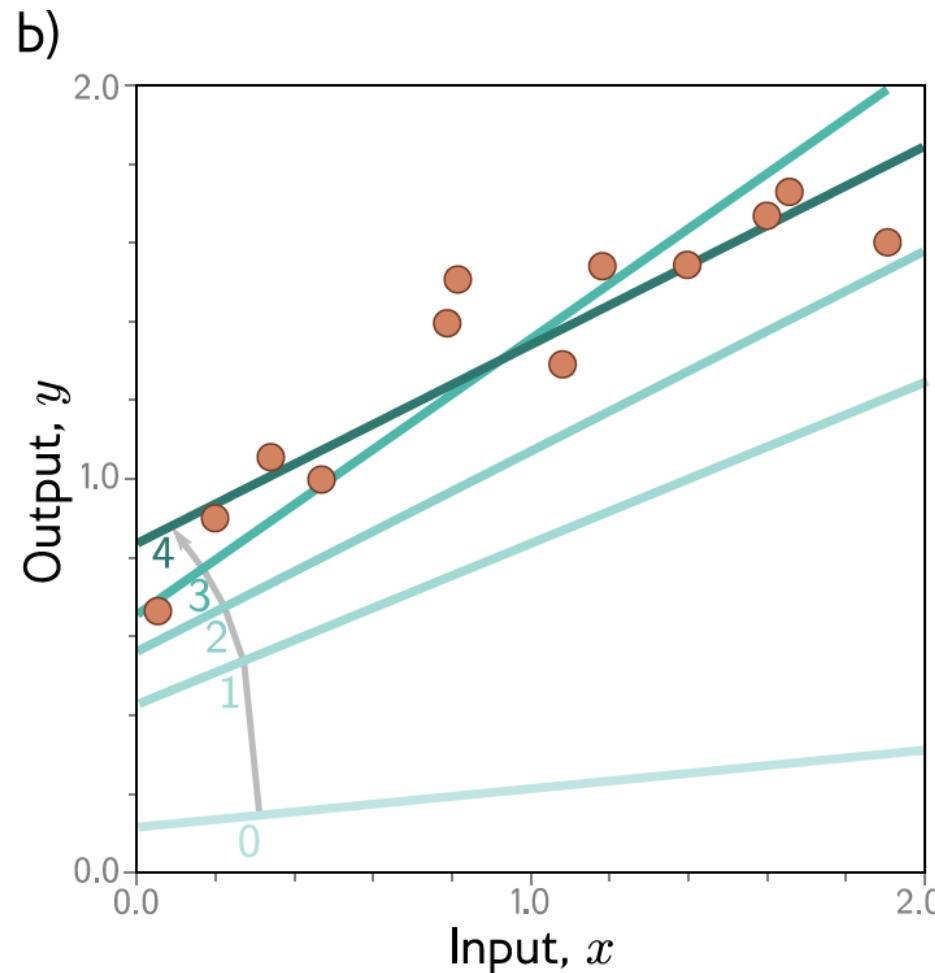
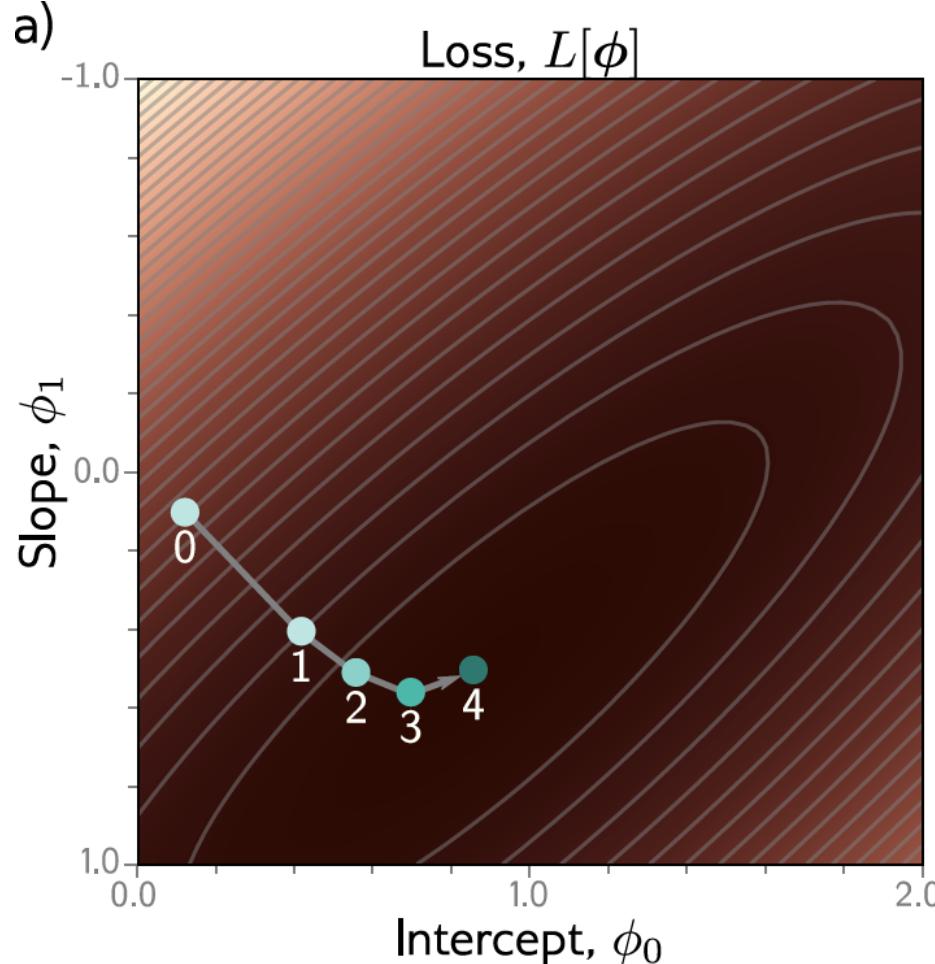
# Ejemplo: entrenamiento de regresión lineal

1D



# Ejemplo: entrenamiento de regresión lineal

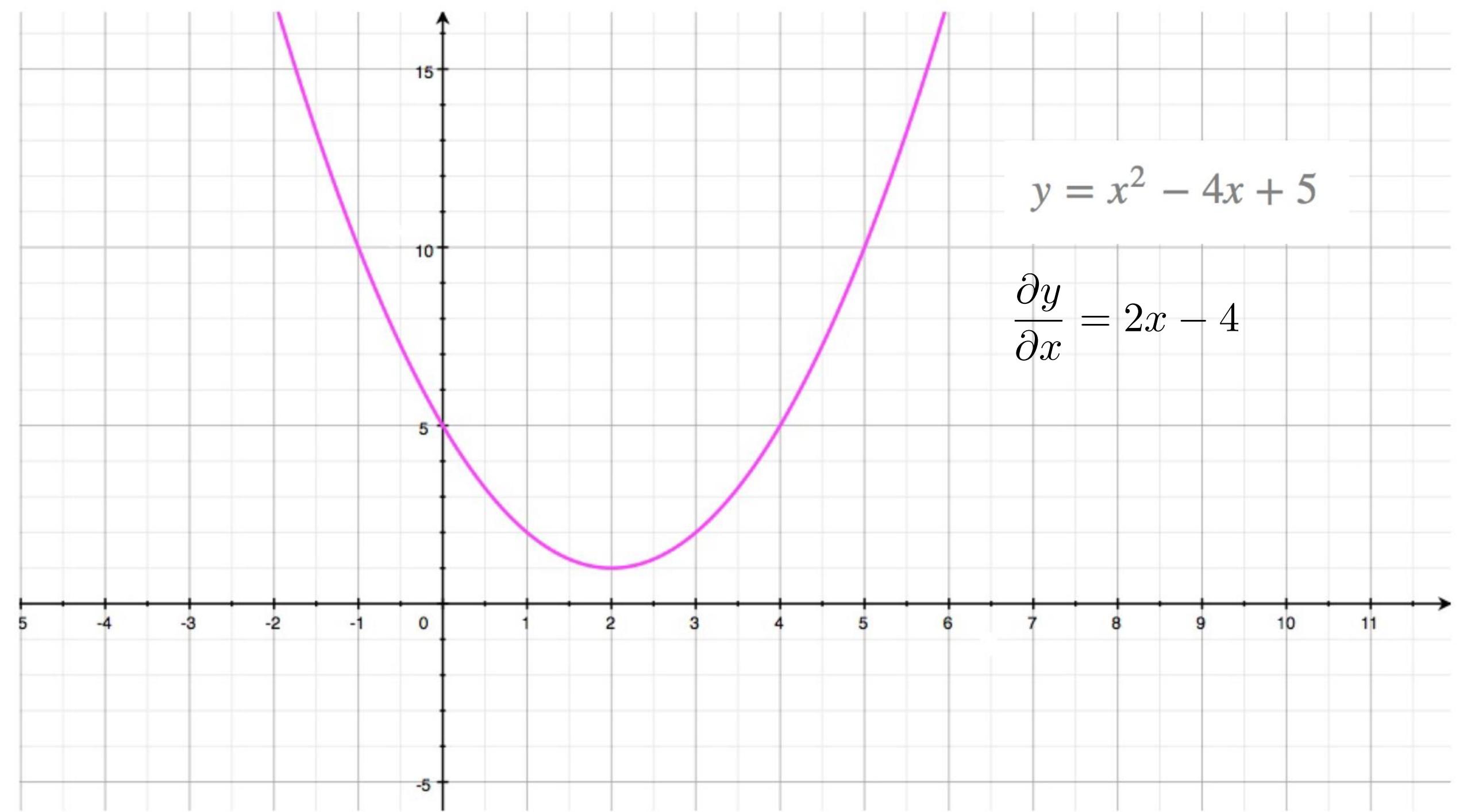
1D

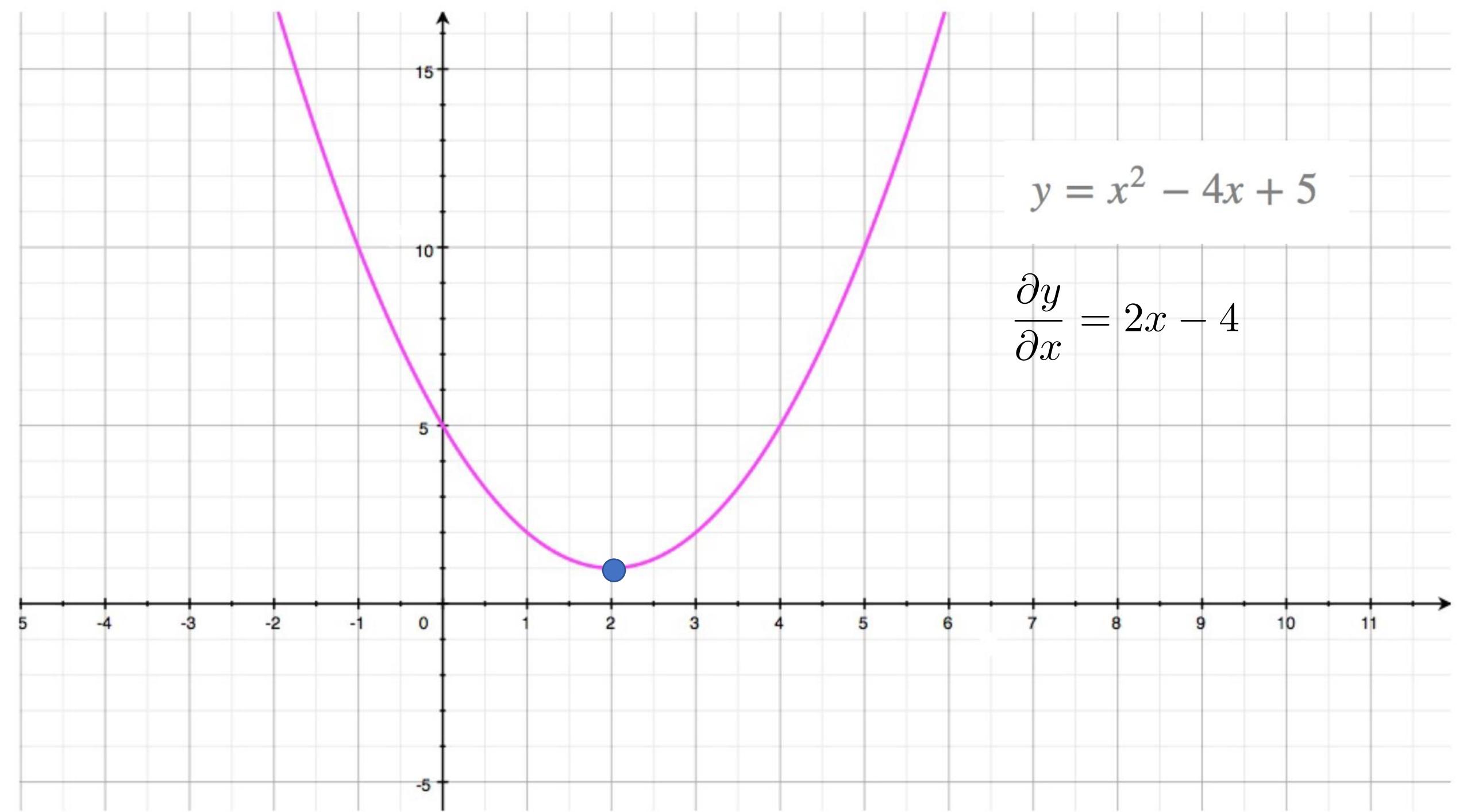


Esta técnica se conoce como **descenso por gradiente**

# Modelos de ajuste

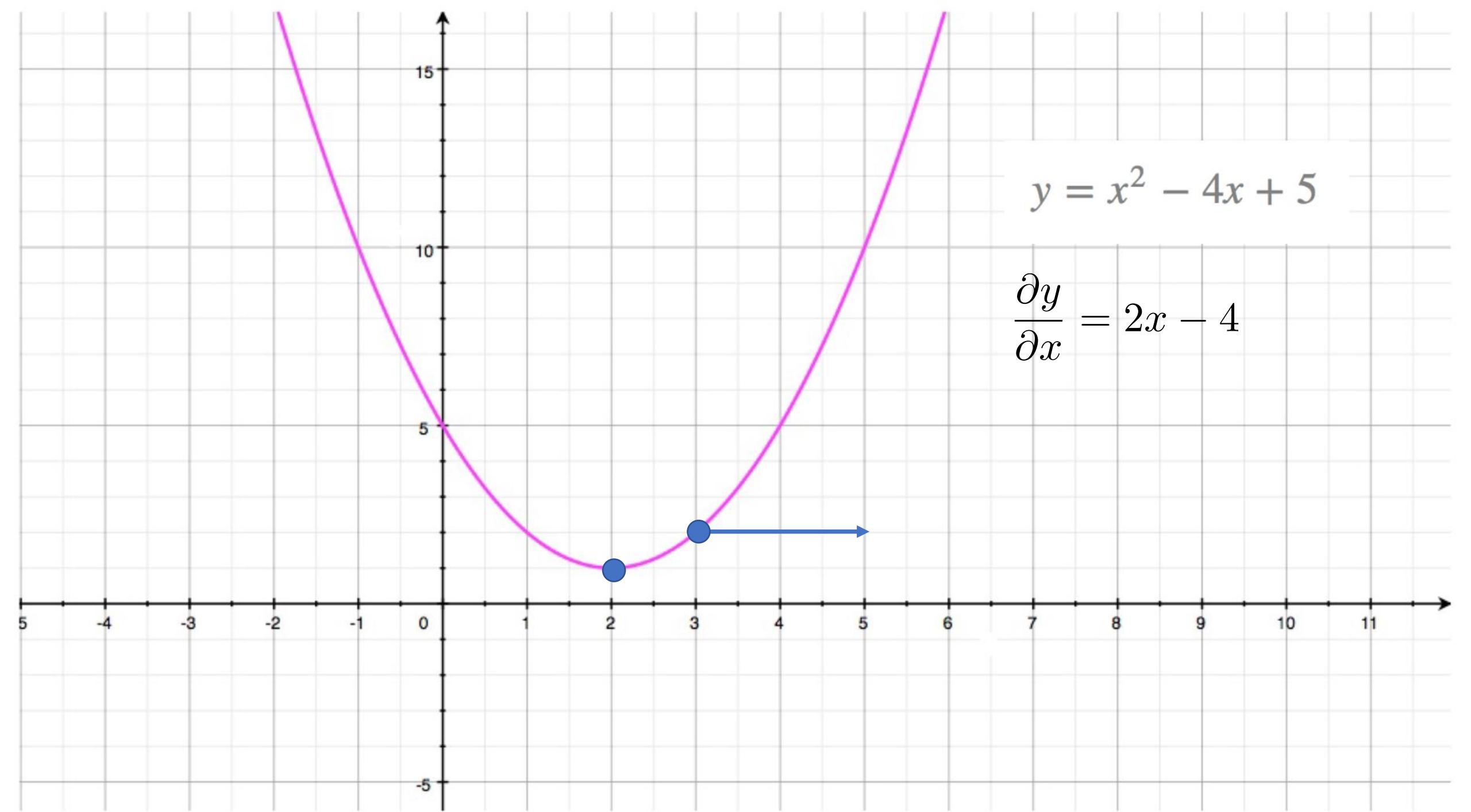
- Matemáticas
- Algoritmo de descenso gradiente
- Ejemplo de regresión lineal
- Ejemplo de modelo de Gabor
- Descenso de gradiente estocástico
- Momentum
- Adam

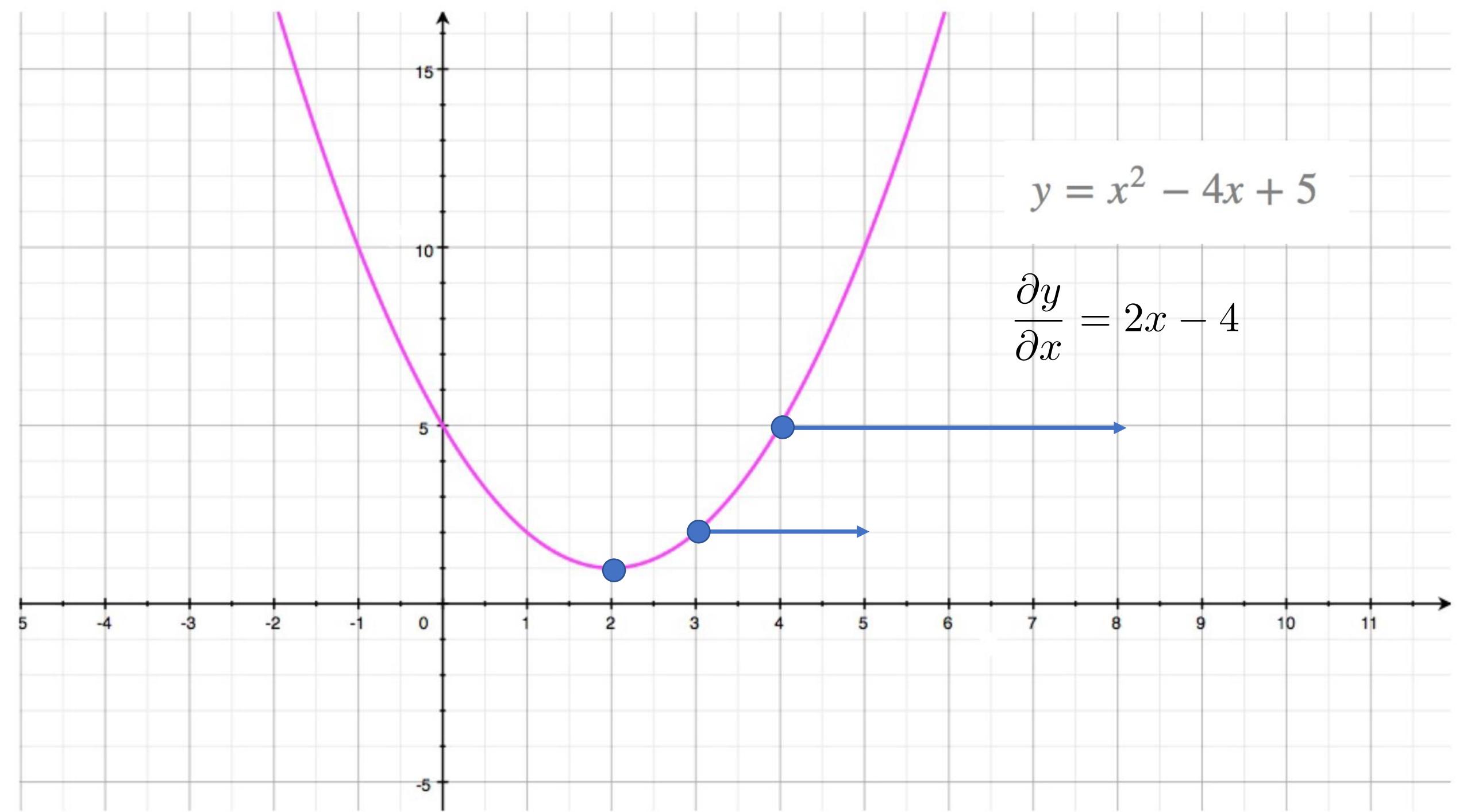


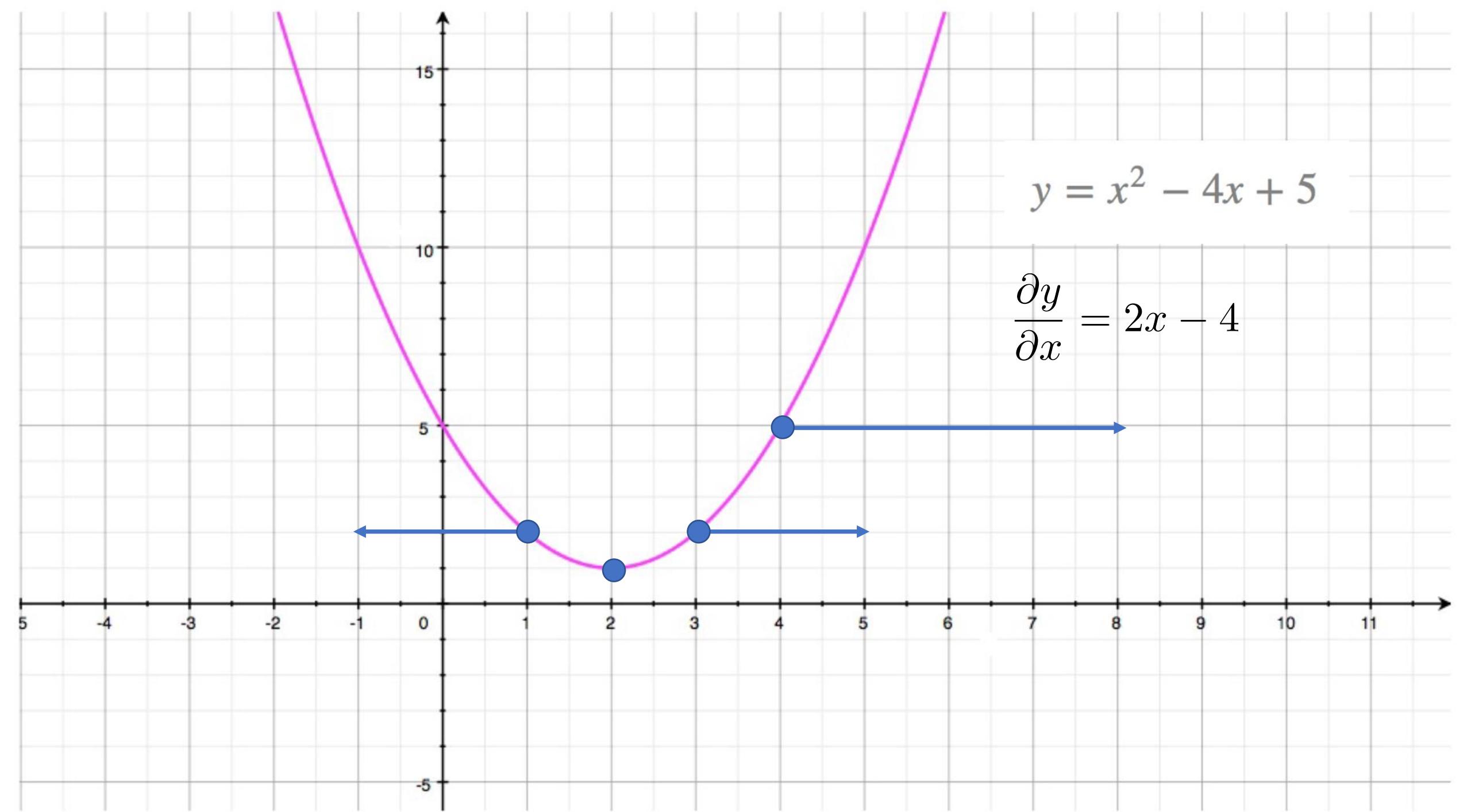


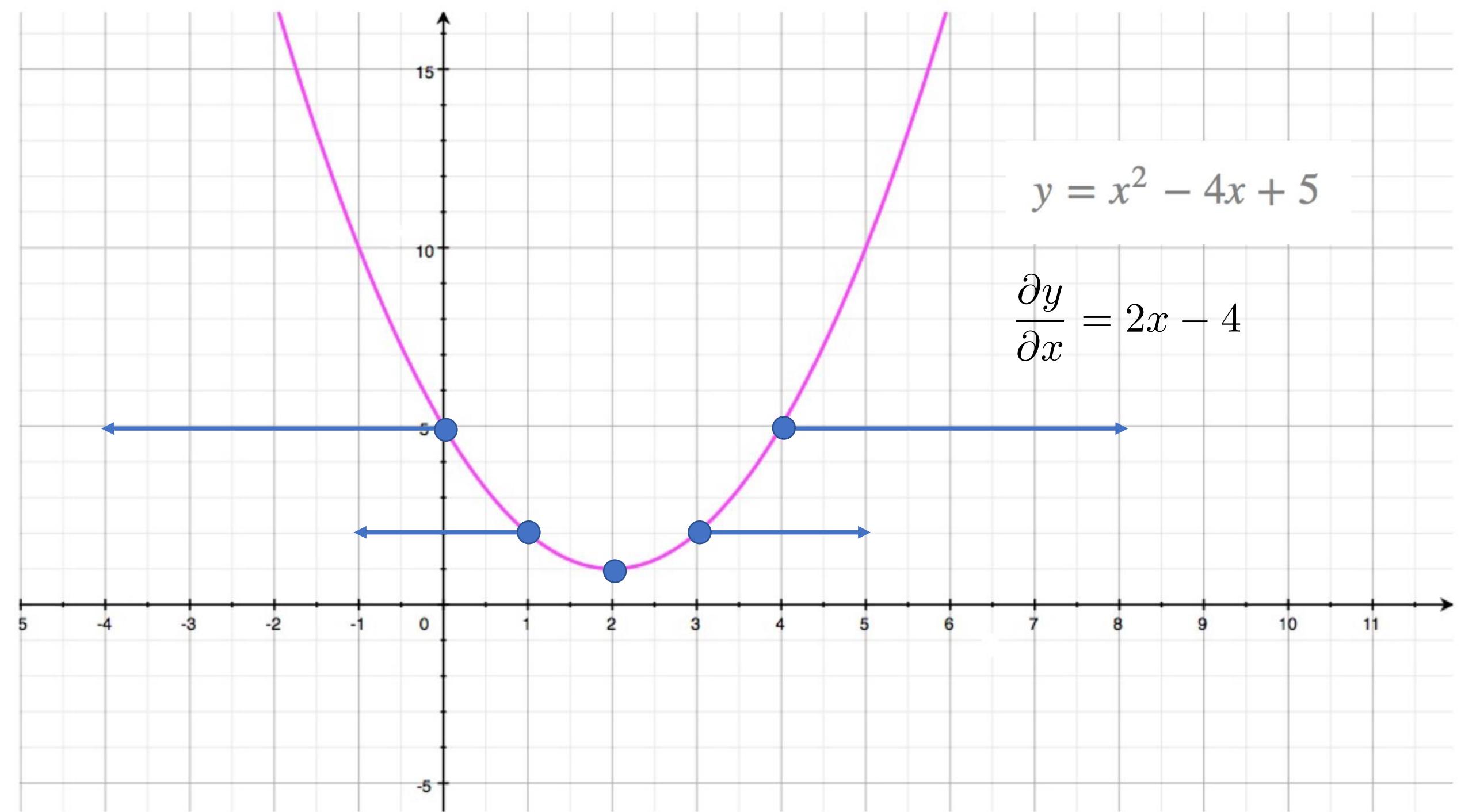
$$y = x^2 - 4x + 5$$

$$\frac{\partial y}{\partial x} = 2x - 4$$









$$y = x^2 - 4x + 5$$

$$\frac{\partial y}{\partial x} = 2x - 4$$

# Modelos de ajuste

- Matemáticas
- Algoritmo de descenso gradiente
- Ejemplo de regresión lineal
- Ejemplo de modelo de Gabor
- Descenso de gradiente estocástico
- Momentum
- Adam

# Algoritmo de descenso gradiente

**Step 1.** Compute the derivatives of the loss with respect to the parameters:

$$\frac{\partial L}{\partial \phi} = \begin{bmatrix} \frac{\partial L}{\partial \phi_0} \\ \frac{\partial L}{\partial \phi_1} \\ \vdots \\ \frac{\partial L}{\partial \phi_N} \end{bmatrix}.$$

**Step 2.** Update the parameters according to the rule:

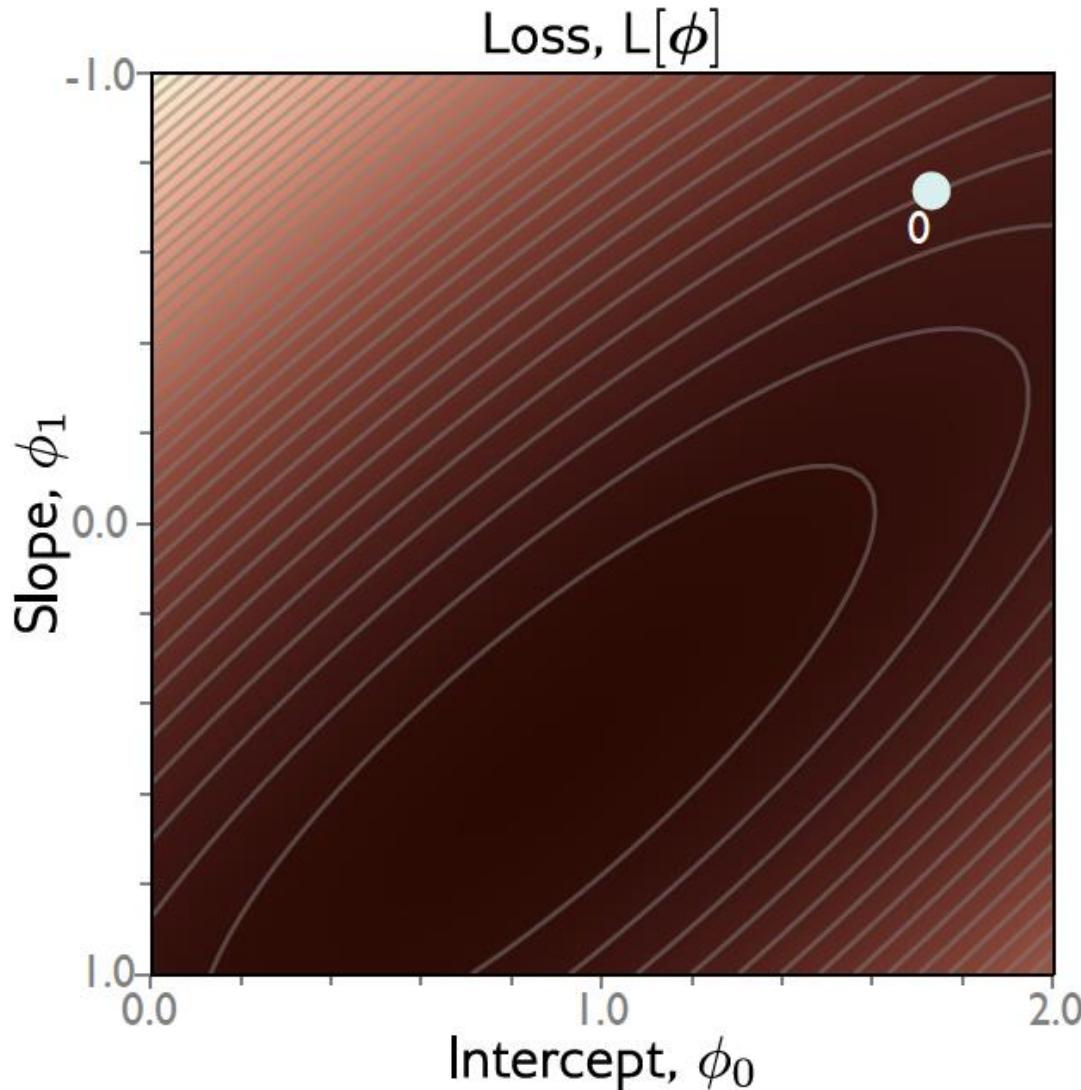
$$\phi \leftarrow \phi - \alpha \frac{\partial L}{\partial \phi},$$

where the positive scalar  $\alpha$  determines the magnitude of the change.

# Modelos de ajuste

- Matemáticas
- Algoritmo de descenso gradiente
- Ejemplo de regresión lineal
- Ejemplo de modelo de Gabor
- Descenso de gradiente estocástico
- Momentum
- Adam

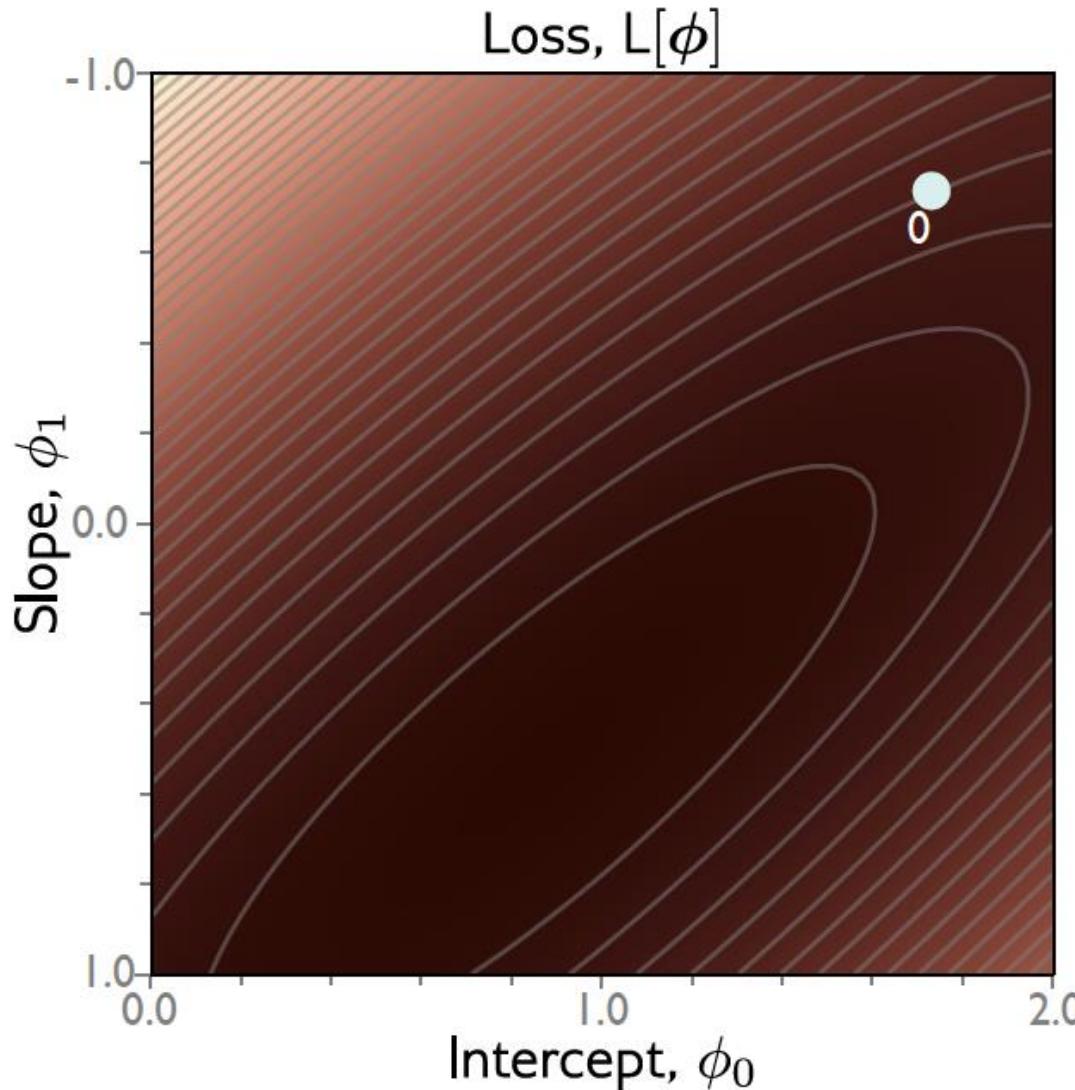
# Descenso gradual



Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

$$\begin{aligned} L[\phi] &= \sum_{i=1}^I \ell_i = \sum_{i=1}^I (f[x_i, \phi] - y_i)^2 \\ &= \sum_{i=1}^I (\phi_0 + \phi_1 x_i - y_i)^2 \end{aligned}$$

# Descenso gradual

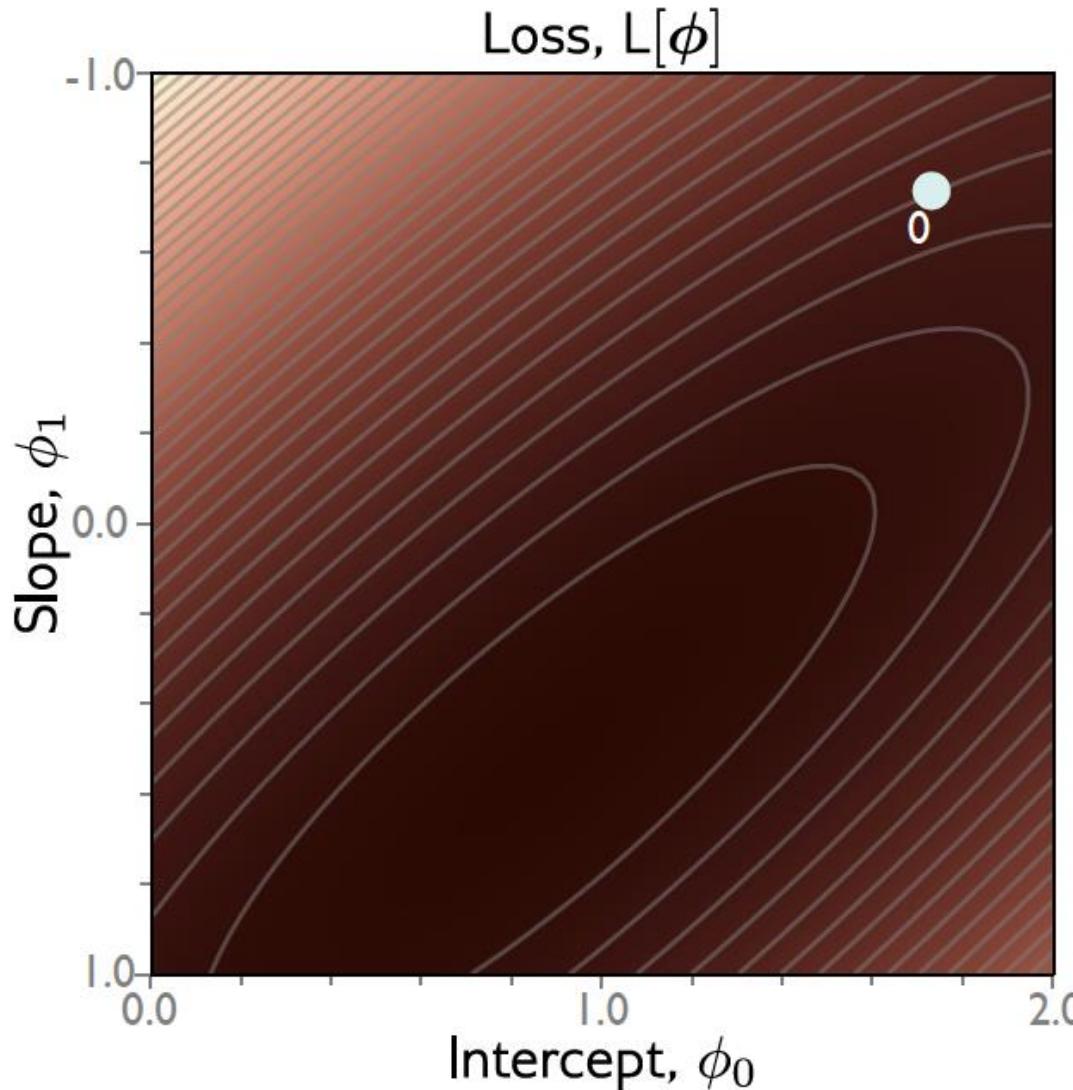


Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

$$\begin{aligned} L[\phi] &= \sum_{i=1}^I \ell_i = \sum_{i=1}^I (f[x_i, \phi] - y_i)^2 \\ &= \sum_{i=1}^I (\phi_0 + \phi_1 x_i - y_i)^2 \end{aligned}$$

$$\frac{\partial L}{\partial \phi} = \frac{\partial}{\partial \phi} \sum_{i=1}^I \ell_i = \sum_{i=1}^I \frac{\partial \ell_i}{\partial \phi}$$

# Descenso gradual



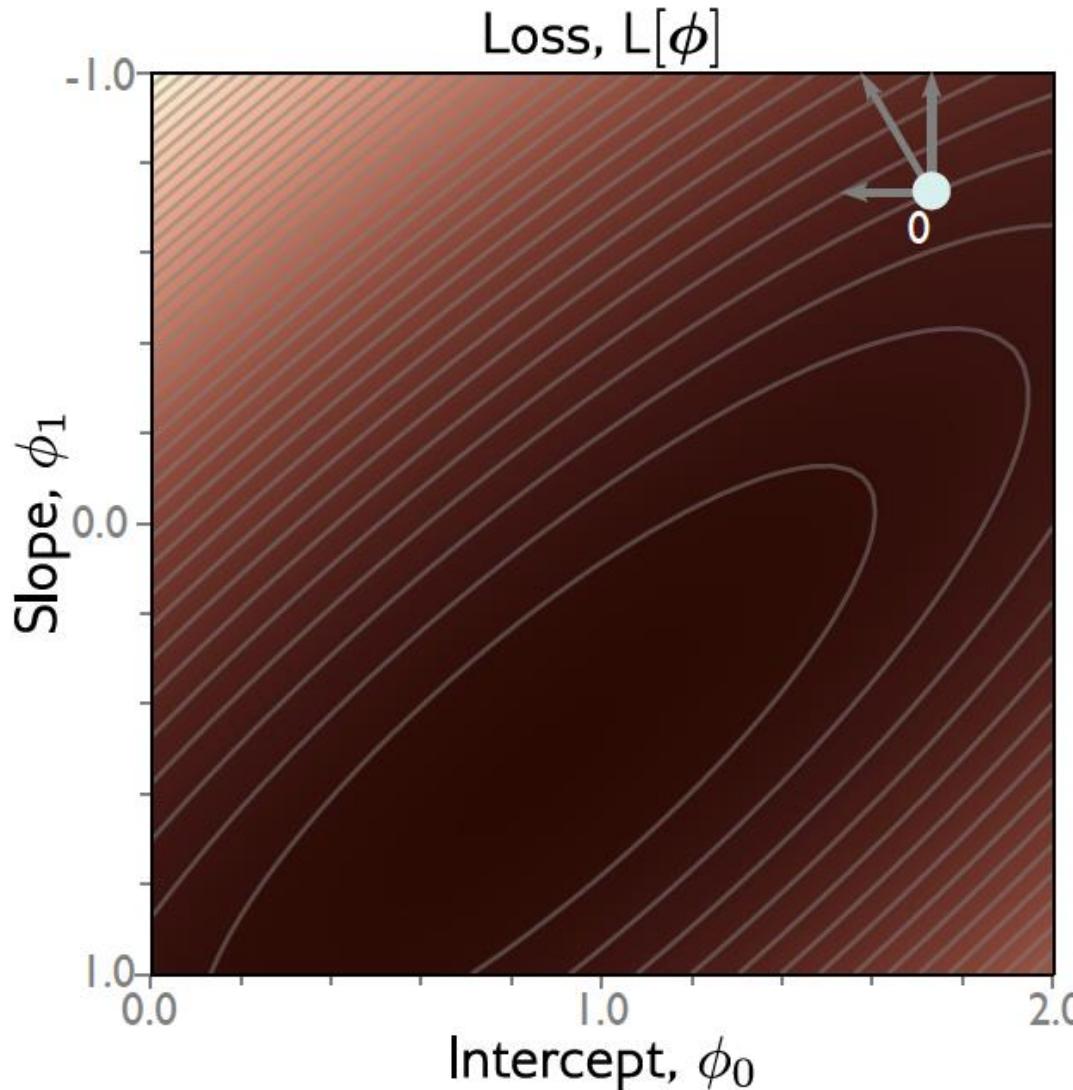
Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

$$\begin{aligned} L[\phi] &= \sum_{i=1}^I \ell_i = \sum_{i=1}^I (f[x_i, \phi] - y_i)^2 \\ &= \sum_{i=1}^I (\phi_0 + \phi_1 x_i - y_i)^2 \end{aligned}$$

$$\frac{\partial L}{\partial \phi} = \frac{\partial}{\partial \phi} \sum_{i=1}^I \ell_i = \sum_{i=1}^I \frac{\partial \ell_i}{\partial \phi}$$

$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

# Descenso gradual

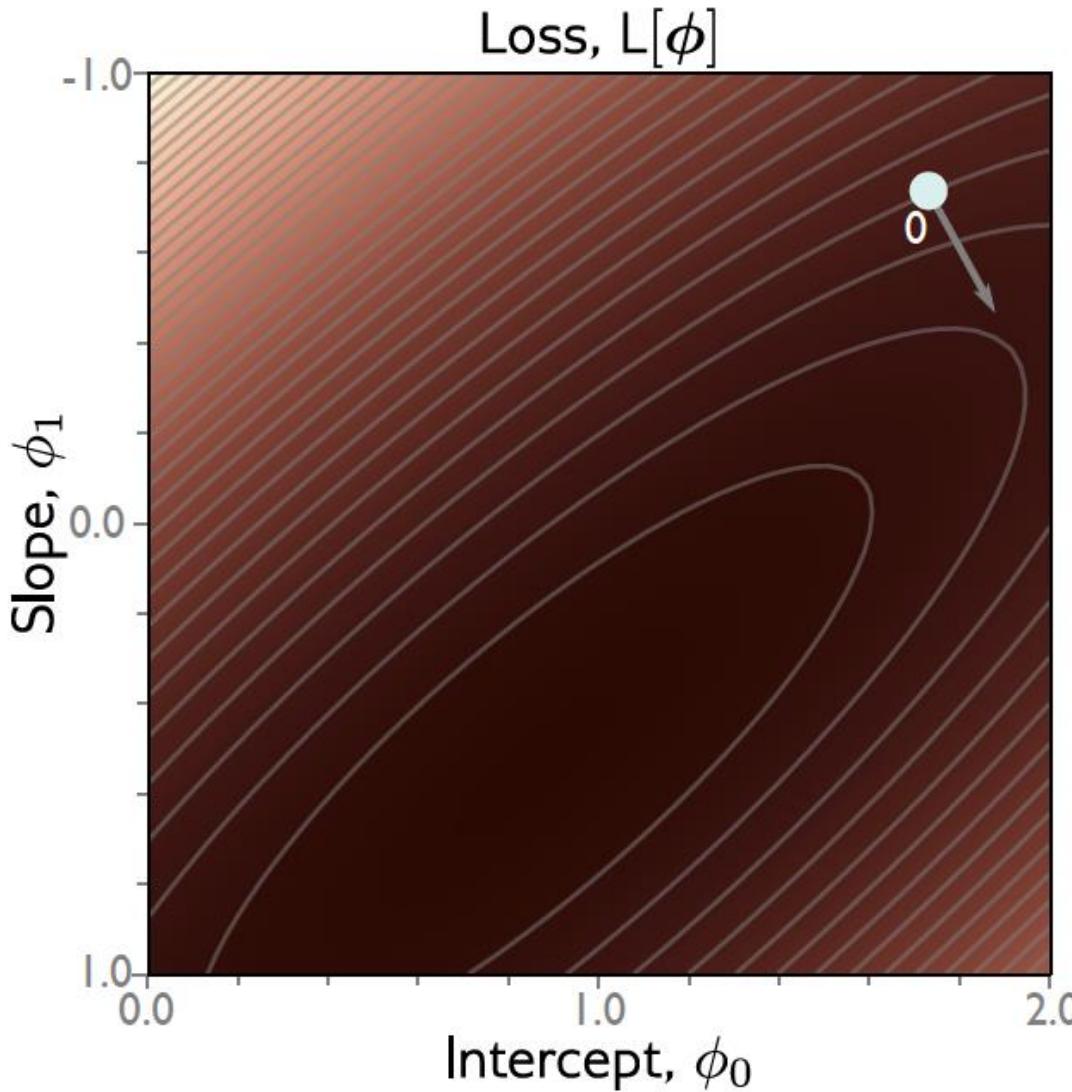


Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

$$\frac{\partial L}{\partial \phi} = \frac{\partial}{\partial \phi} \sum_{i=1}^I \ell_i = \sum_{i=1}^I \frac{\partial \ell_i}{\partial \phi}$$

$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

# Descenso gradual



Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

$$\frac{\partial L}{\partial \phi} = \frac{\partial}{\partial \phi} \sum_{i=1}^I \ell_i = \sum_{i=1}^I \frac{\partial \ell_i}{\partial \phi}$$

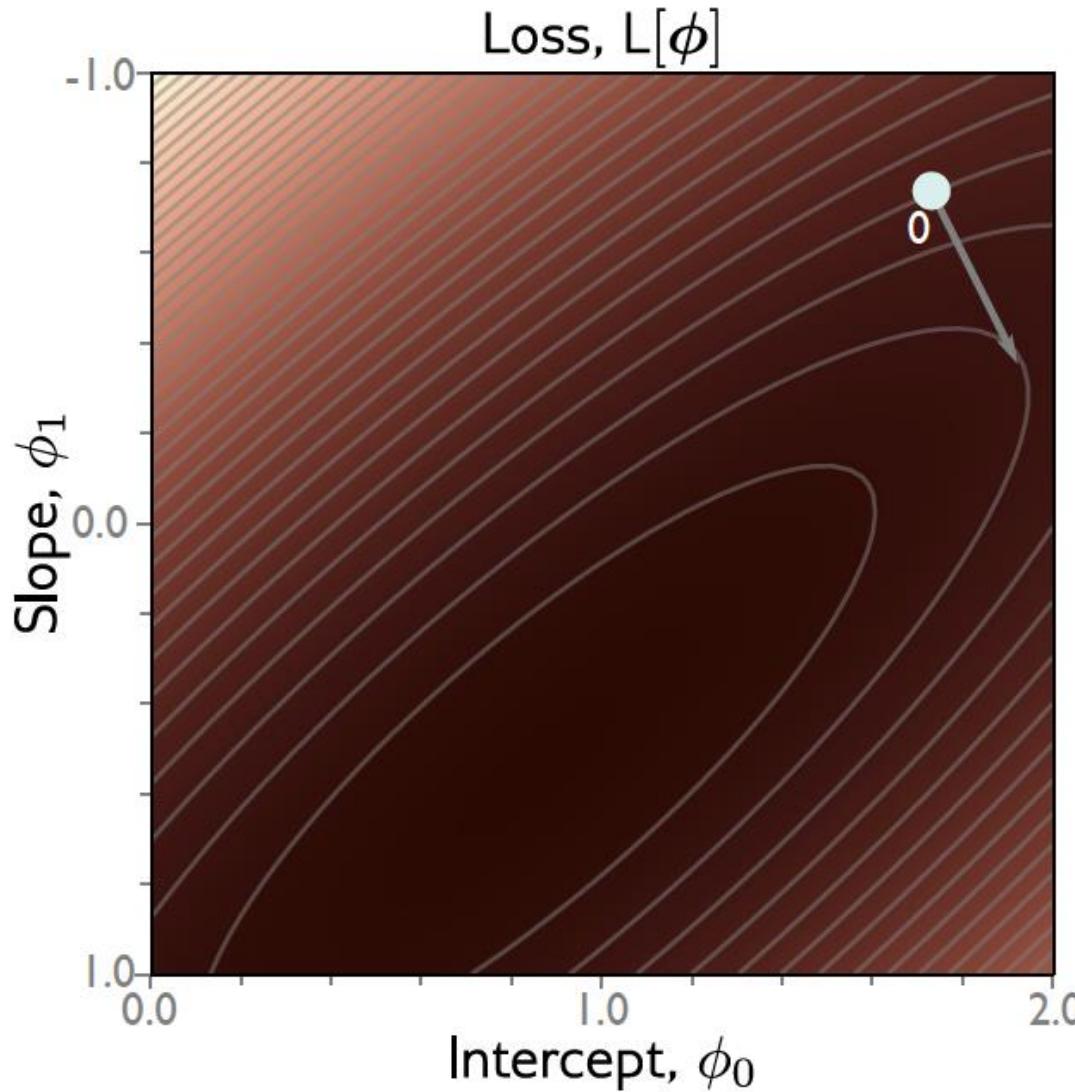
$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

Paso 2: Actualizar los parámetros según la regla

$$\phi \leftarrow \phi - \alpha \frac{\partial L}{\partial \phi}$$

= tamaño del paso o **ritmo de aprendizaje** si es fijo

# Descenso gradual



Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

$$\frac{\partial L}{\partial \phi} = \frac{\partial}{\partial \phi} \sum_{i=1}^I \ell_i = \sum_{i=1}^I \frac{\partial \ell_i}{\partial \phi}$$

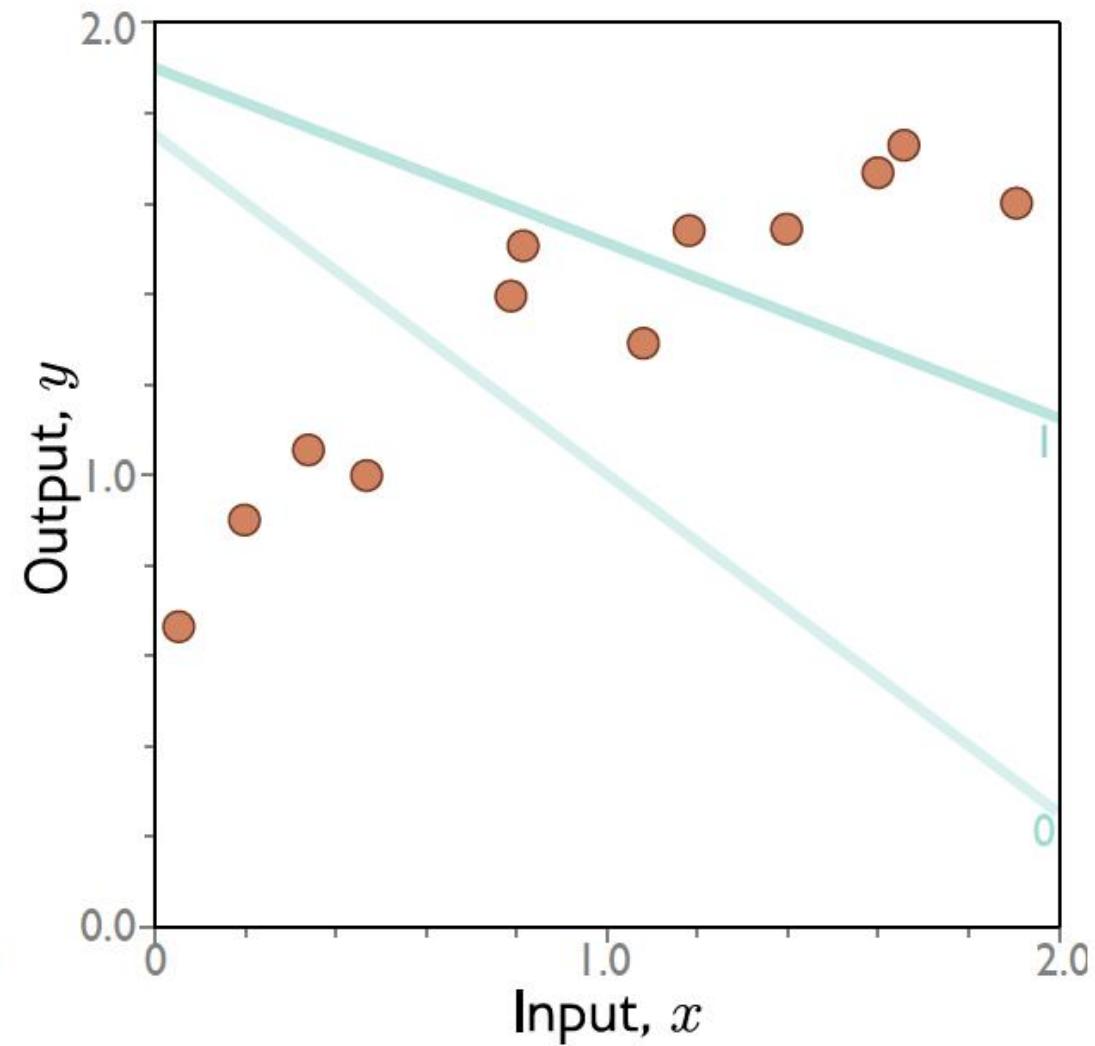
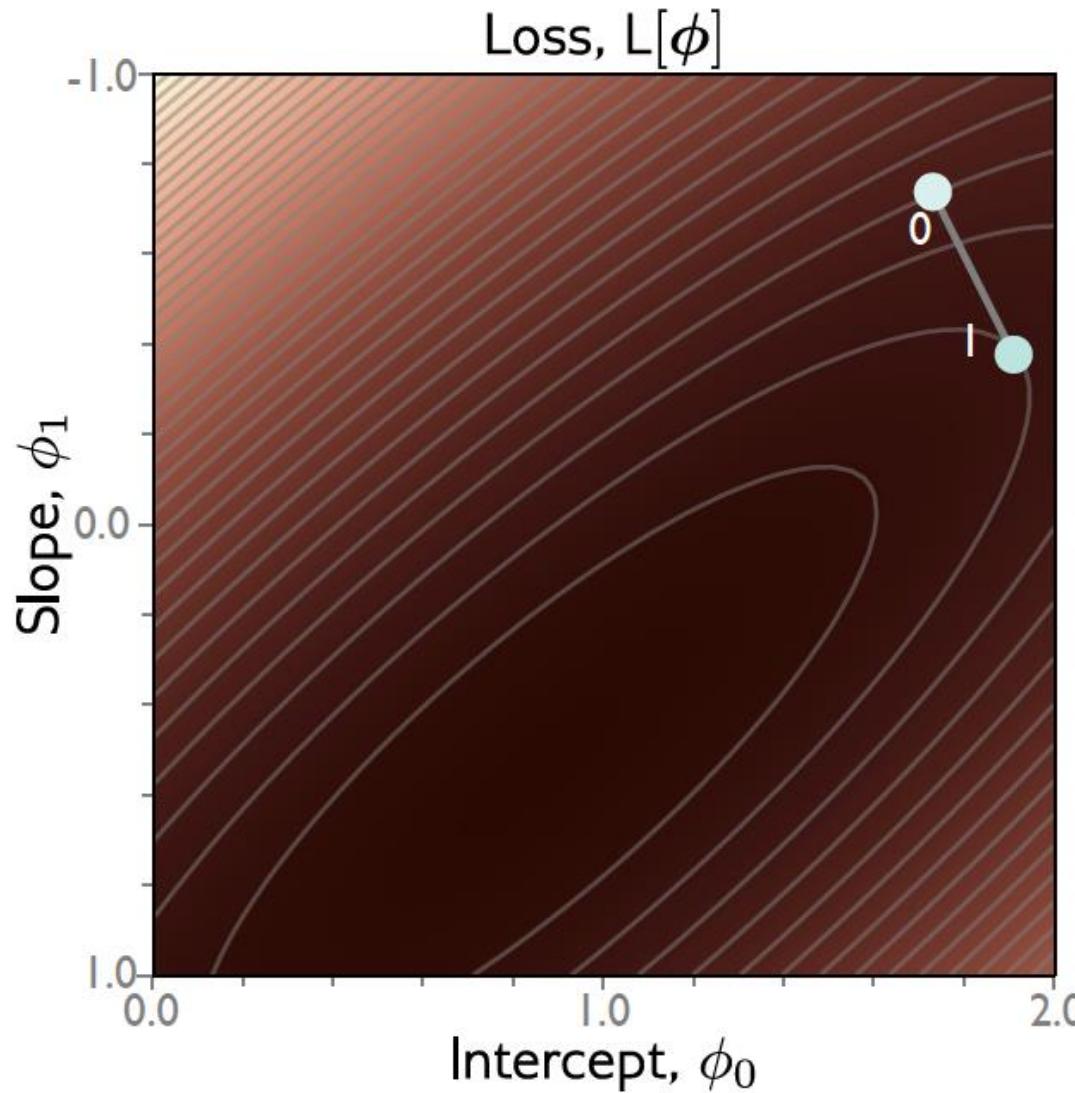
$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

Paso 2: Actualizar los parámetros según la regla

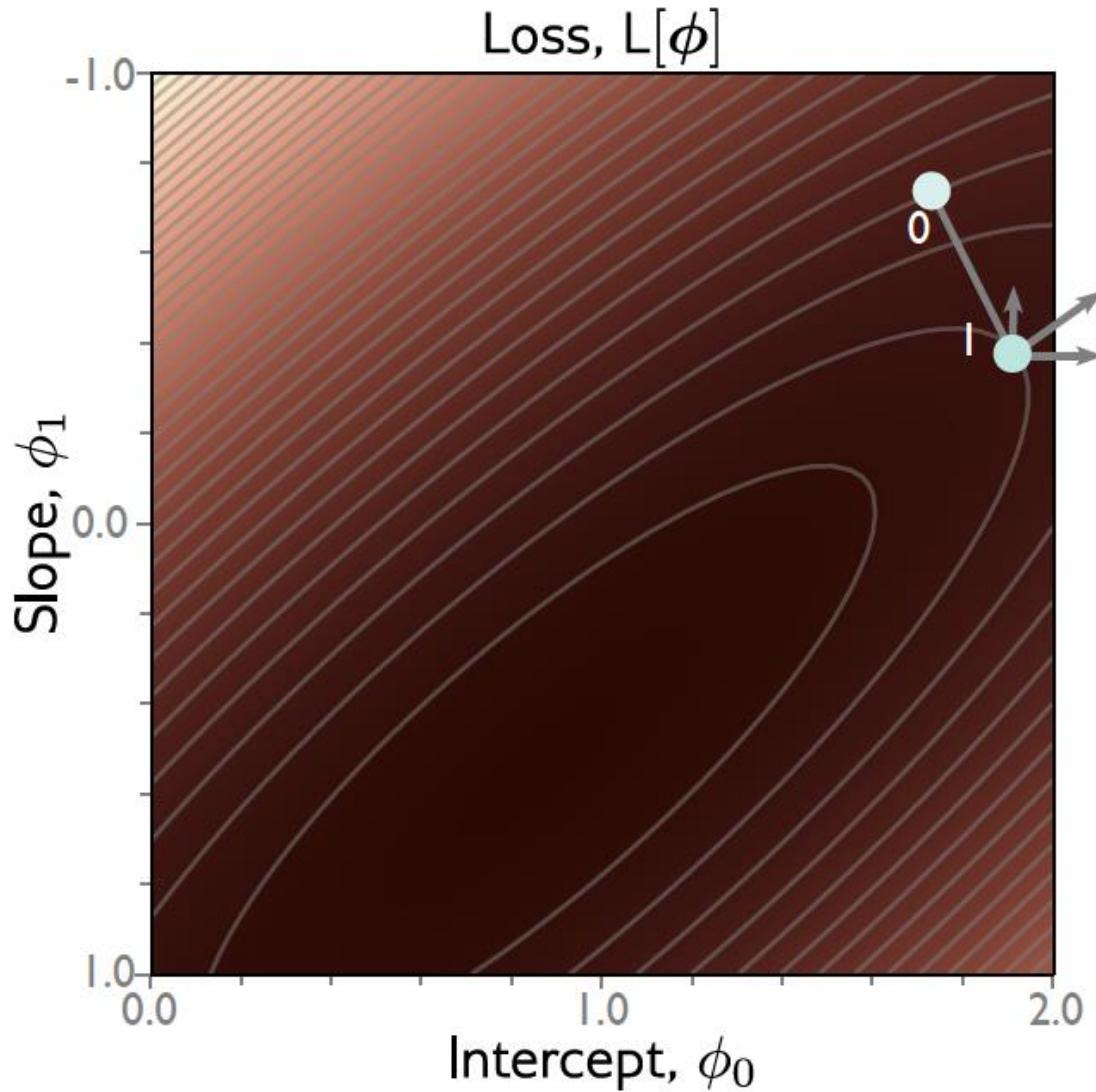
$$\phi \leftarrow \phi - \alpha \frac{\partial L}{\partial \phi}$$

= tamaño del paso

# Descenso gradual



# Descenso gradual



Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

$$\frac{\partial L}{\partial \phi} = \frac{\partial}{\partial \phi} \sum_{i=1}^I \ell_i = \sum_{i=1}^I \frac{\partial \ell_i}{\partial \phi}$$

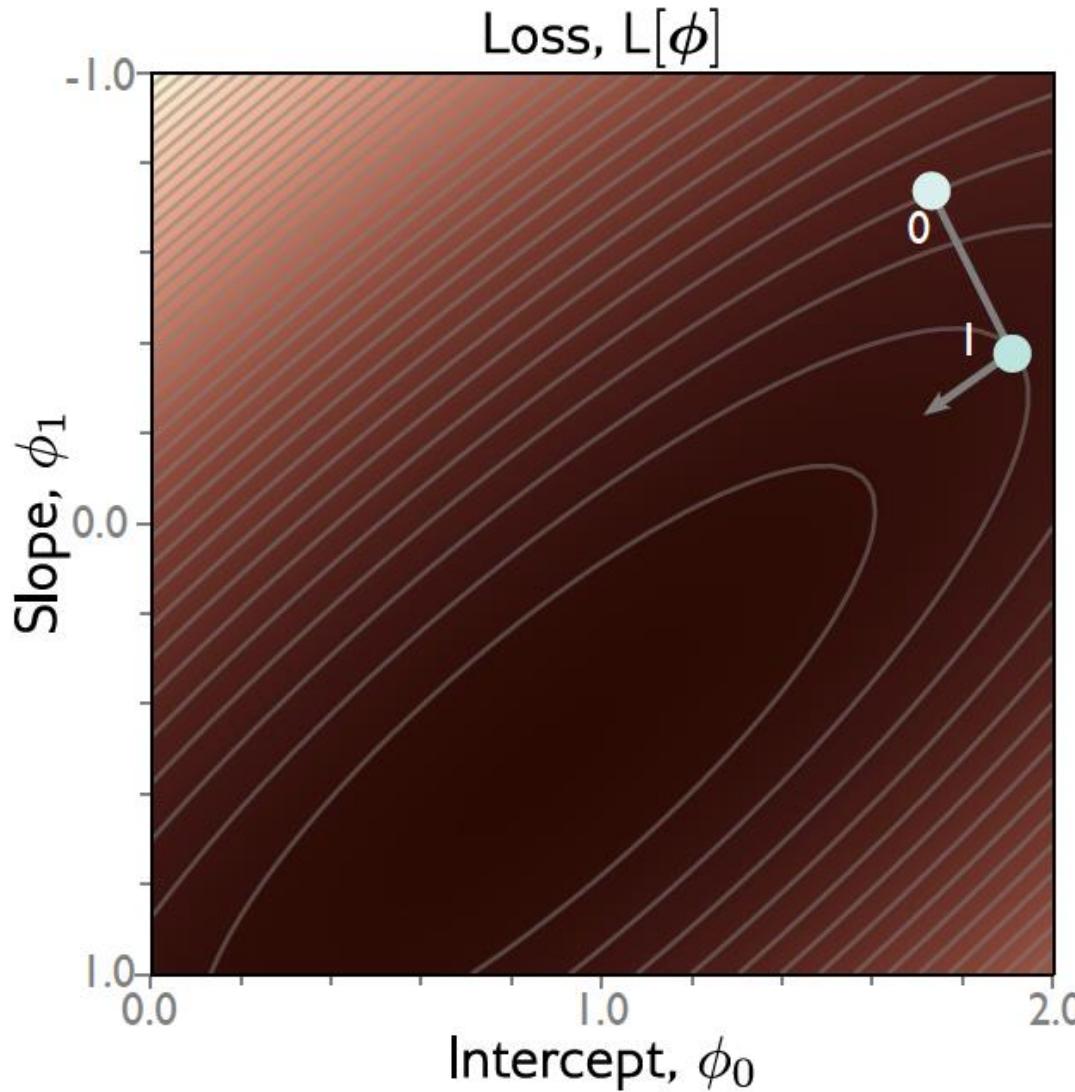
$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

Paso 2: Actualizar los parámetros según la regla

$$\phi \leftarrow \phi - \alpha \frac{\partial L}{\partial \phi}$$

= tamaño del paso

# Descenso gradual



Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

$$\frac{\partial L}{\partial \phi} = \frac{\partial}{\partial \phi} \sum_{i=1}^I \ell_i = \sum_{i=1}^I \frac{\partial \ell_i}{\partial \phi}$$

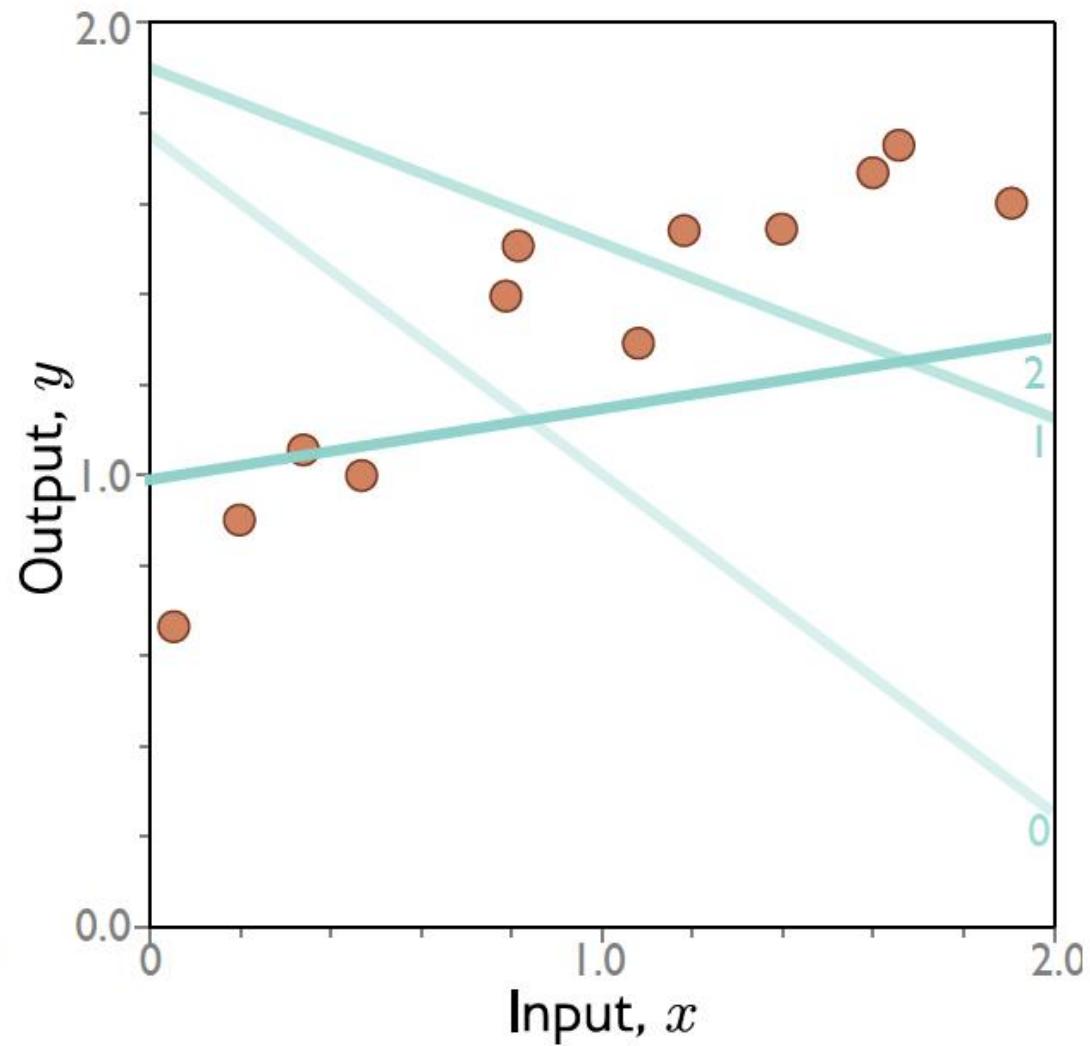
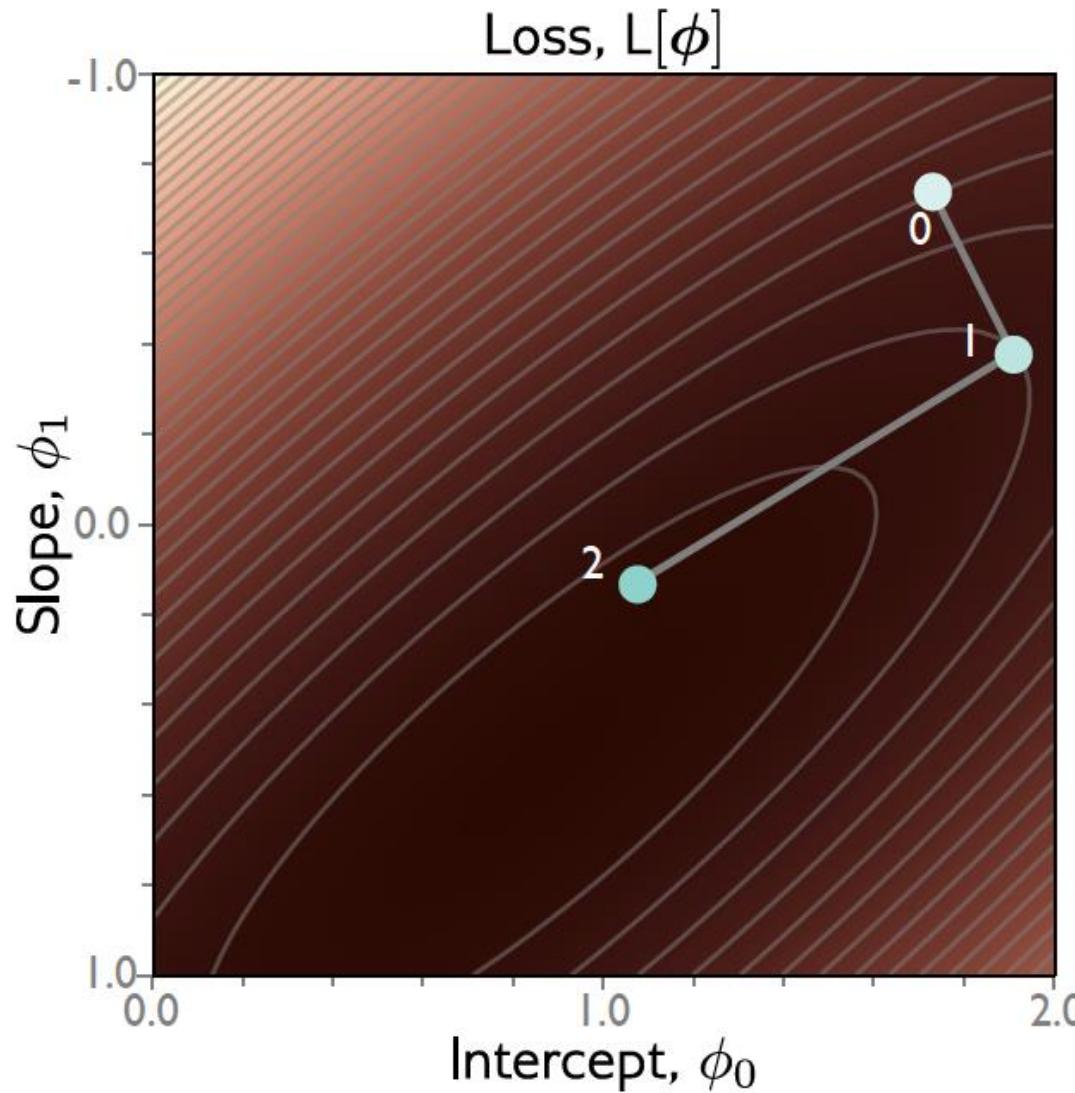
$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

Paso 2: Actualizar los parámetros según la regla

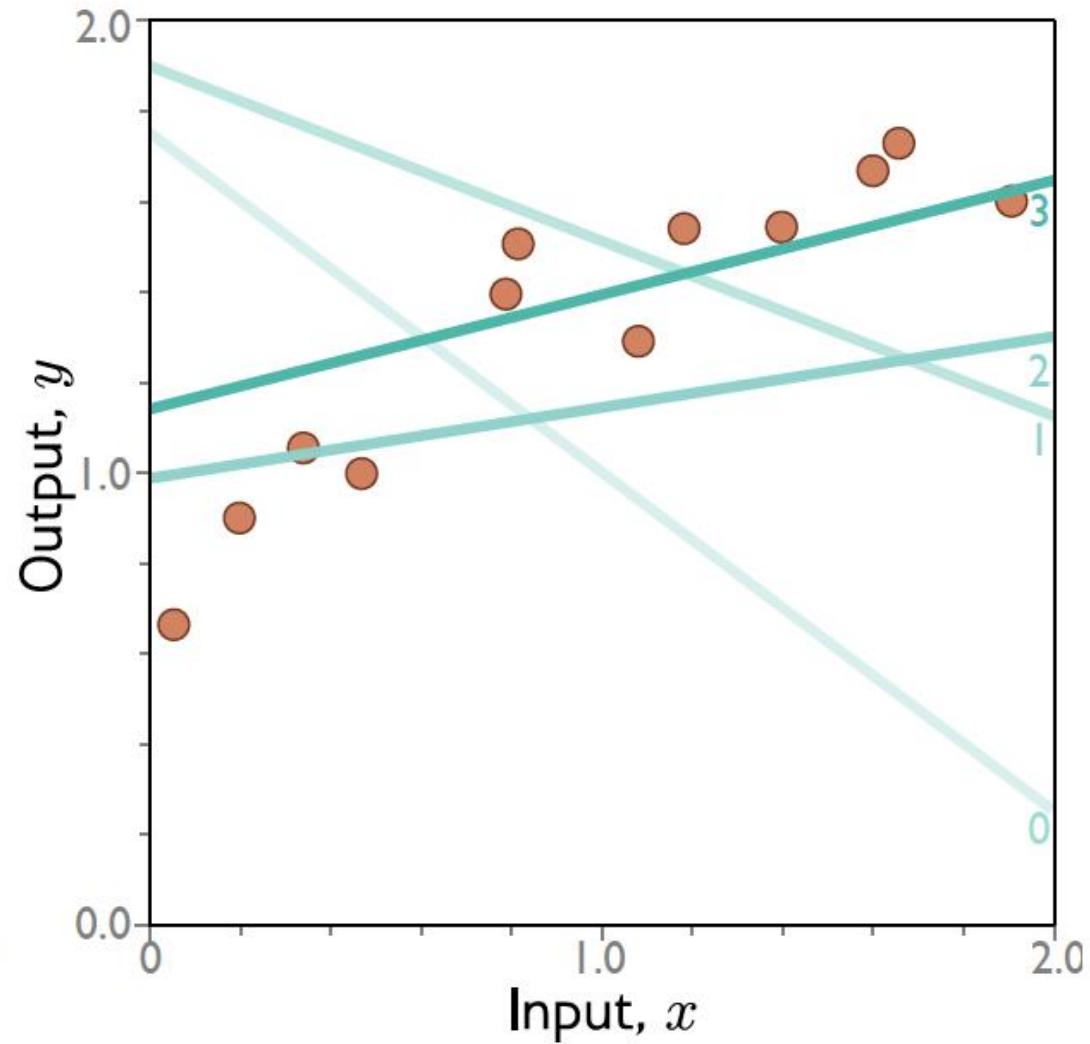
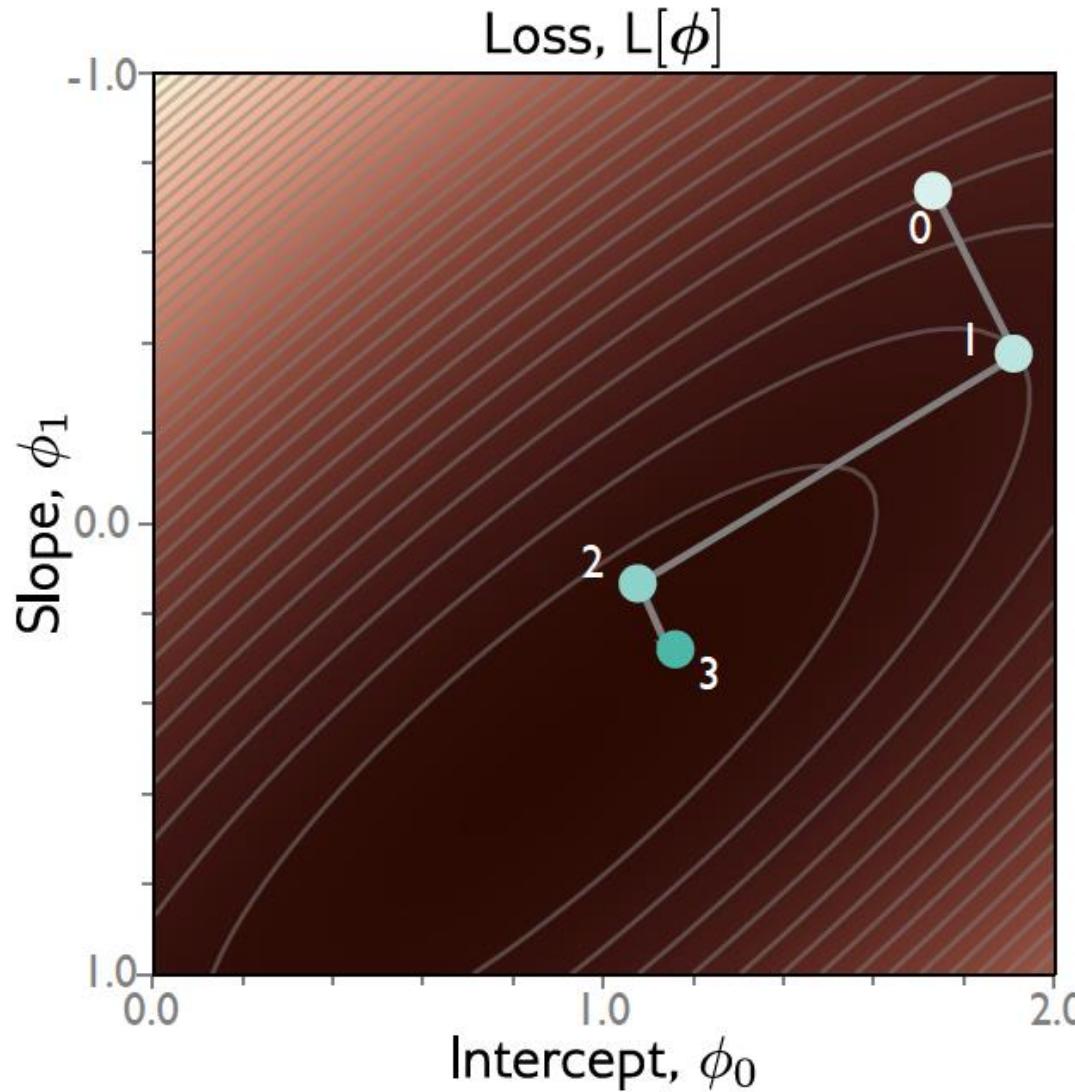
$$\phi \leftarrow \phi - \alpha \frac{\partial L}{\partial \phi}$$

= tamaño del paso

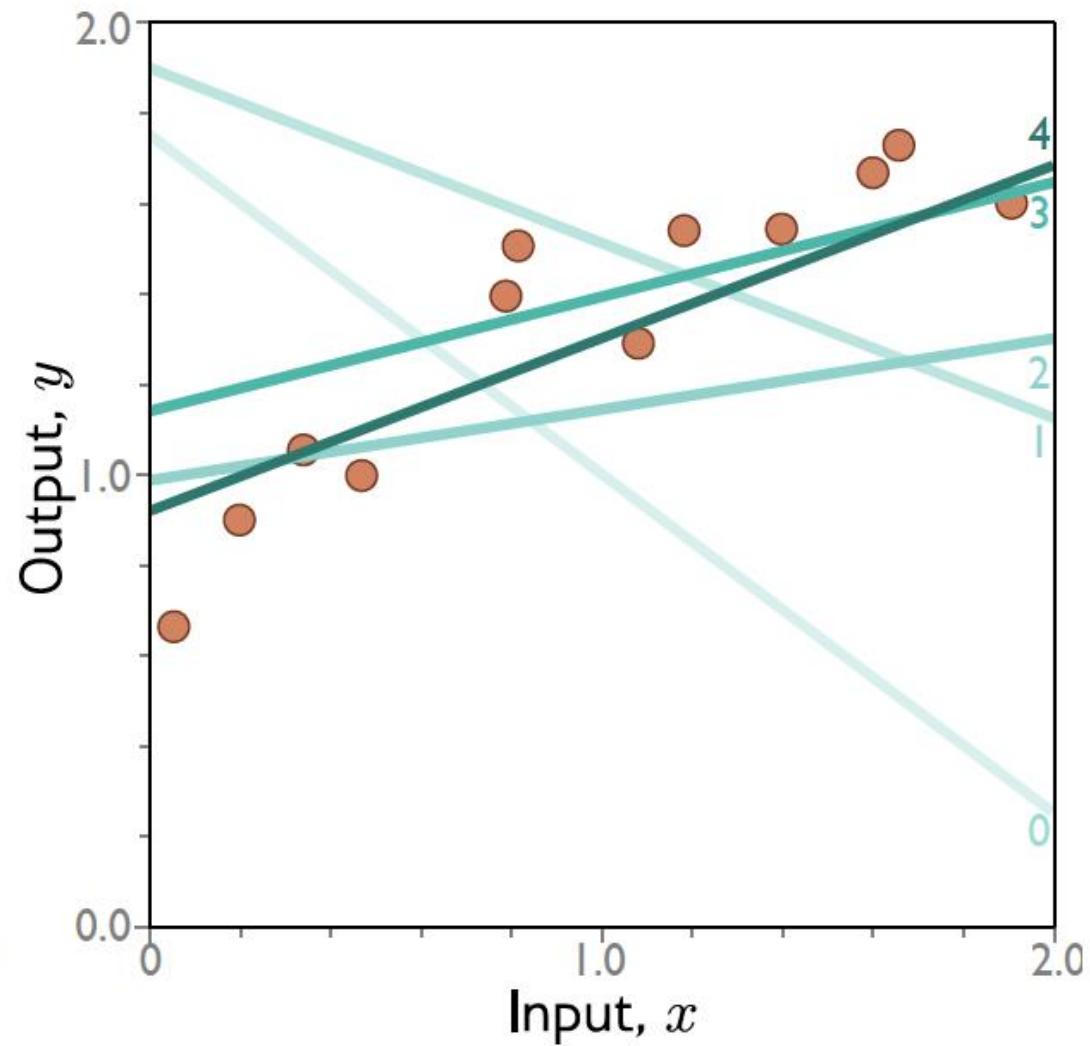
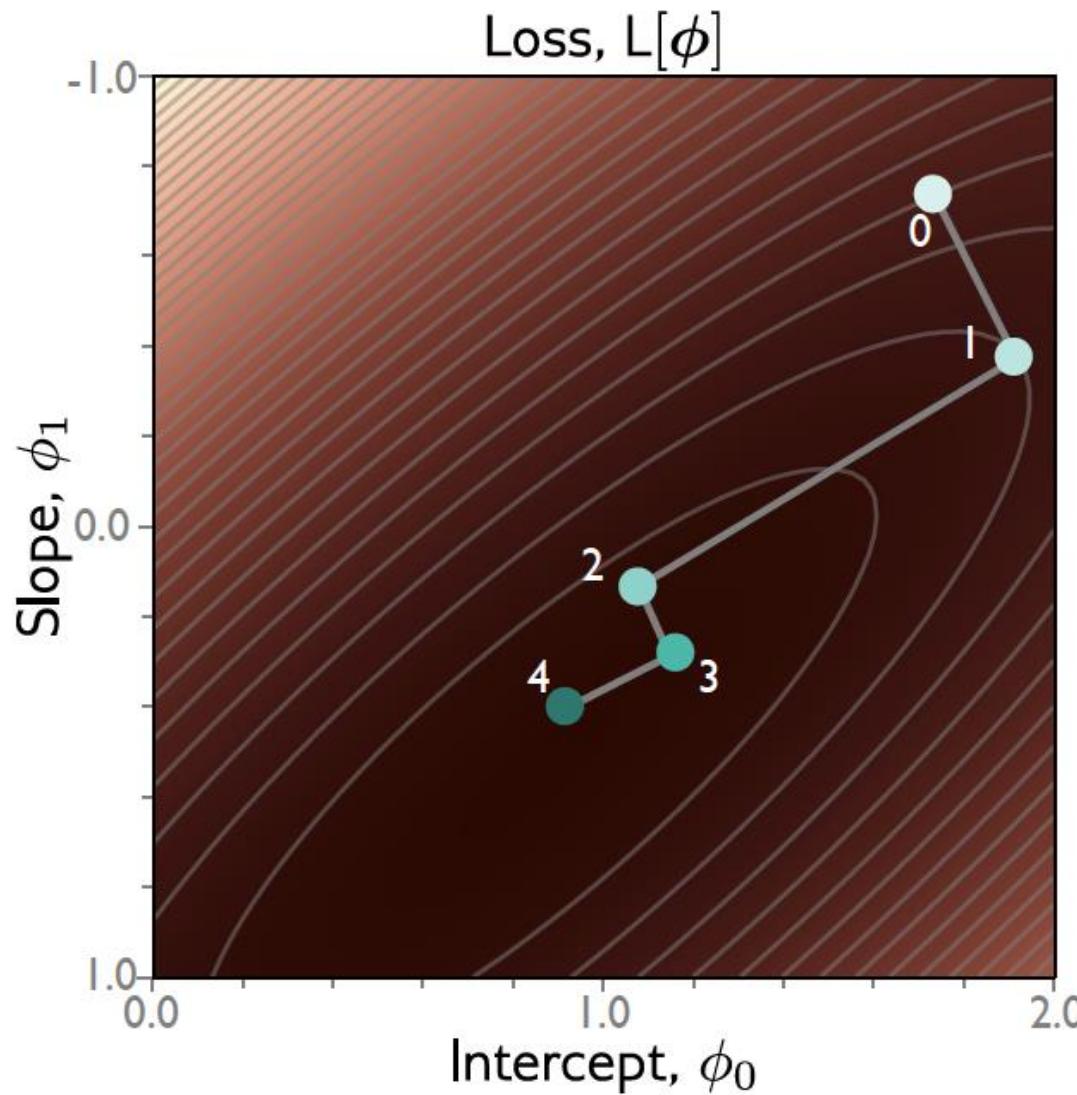
# Descenso gradual



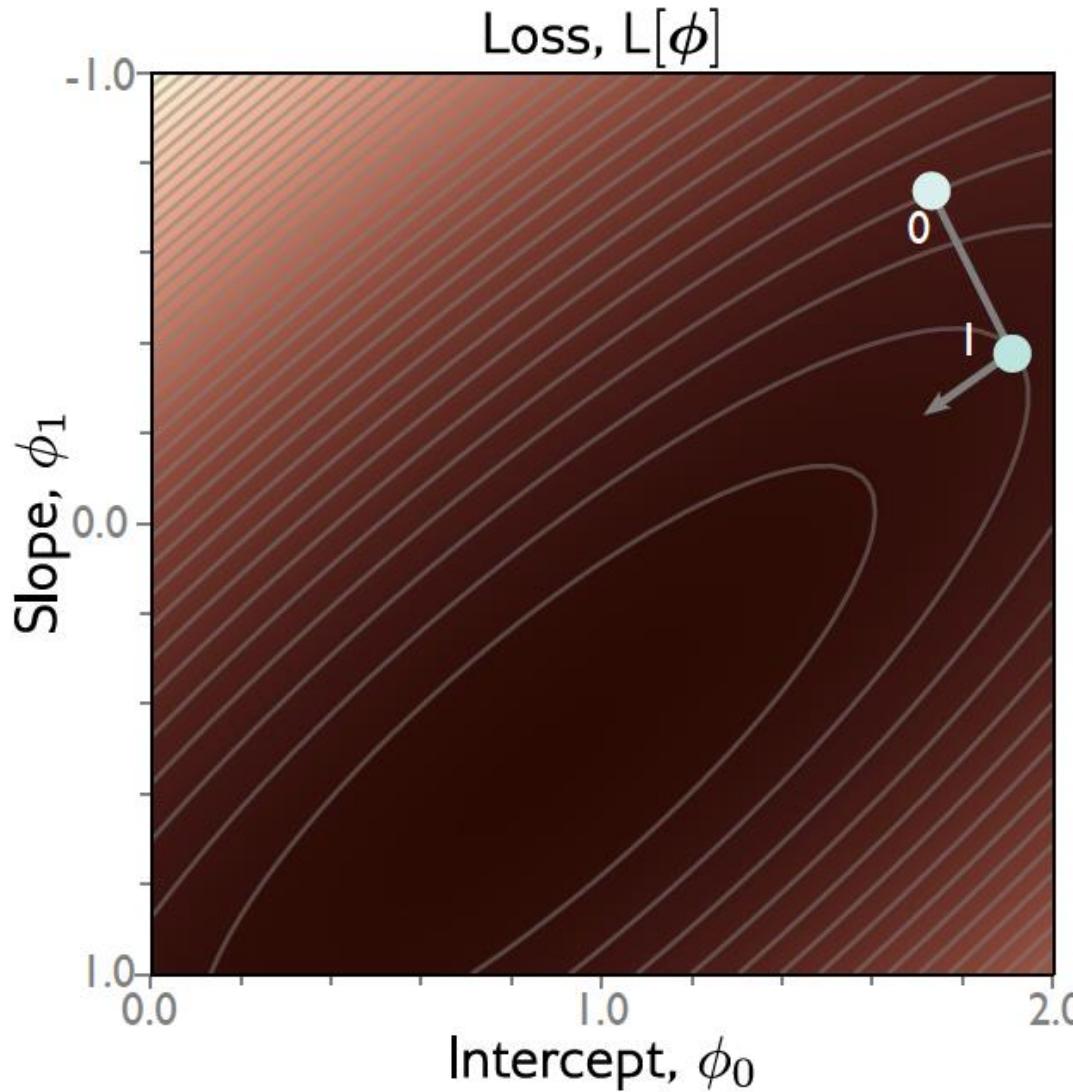
# Descenso gradual



# Descenso gradual



# Búsqueda en línea



Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

$$\frac{\partial L}{\partial \phi} = \frac{\partial}{\partial \phi} \sum_{i=1}^I \ell_i = \sum_{i=1}^I \frac{\partial \ell_i}{\partial \phi}$$

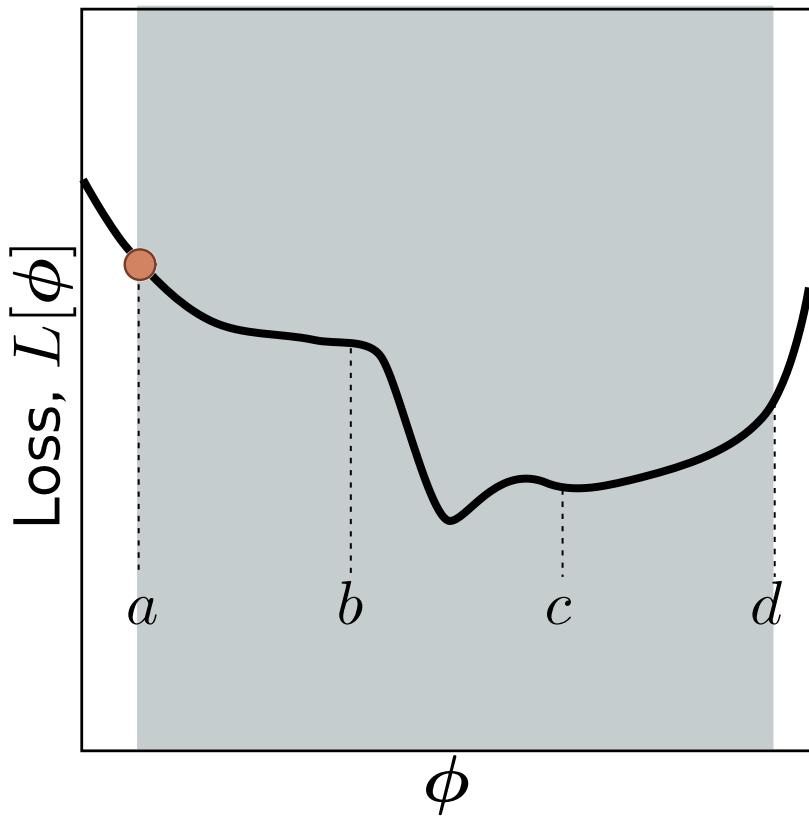
$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

Paso 2: Actualizar los parámetros según la regla

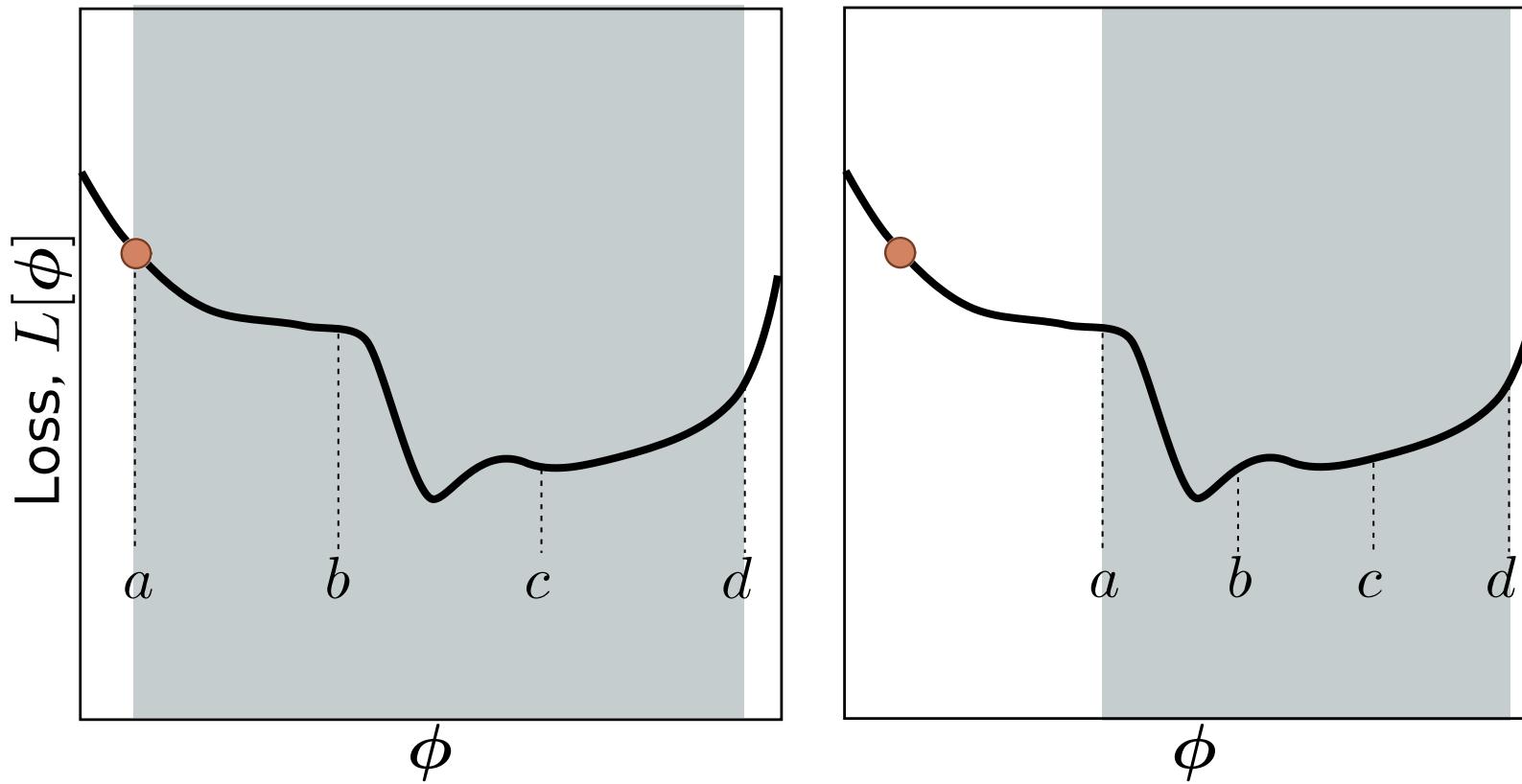
$$\phi \leftarrow \phi - \alpha \frac{\partial L}{\partial \phi}$$

= tamaño del paso

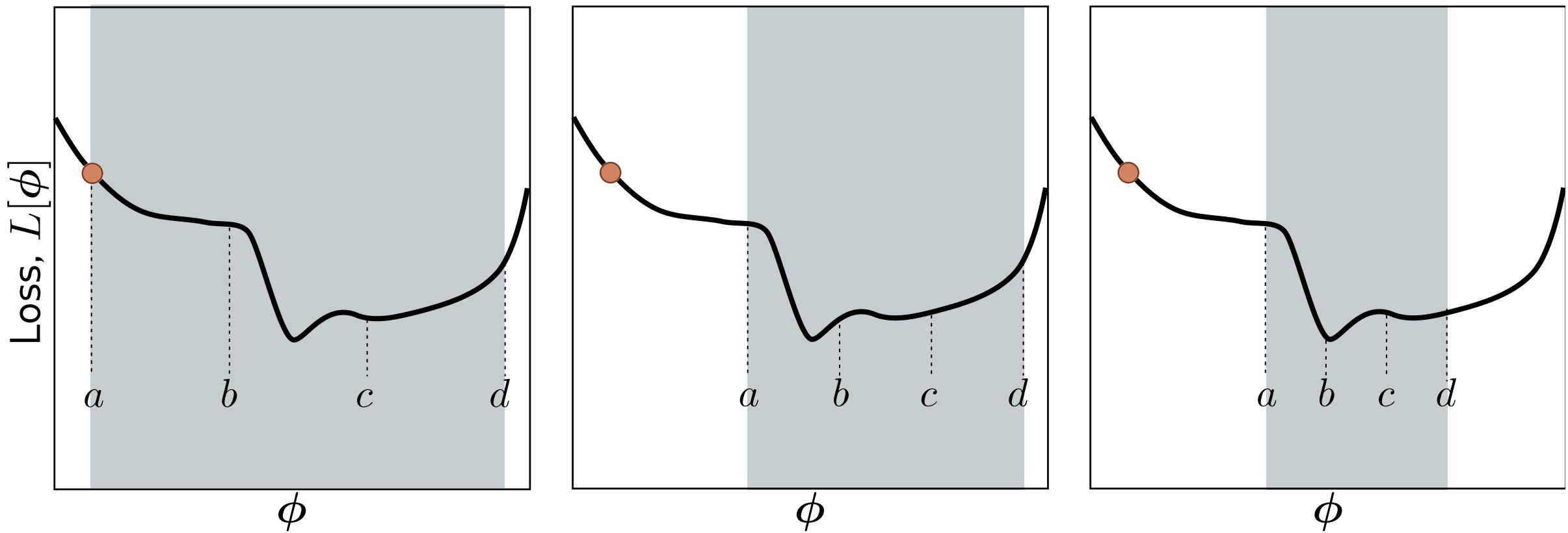
# Búsqueda de líneas (entre corchetes)



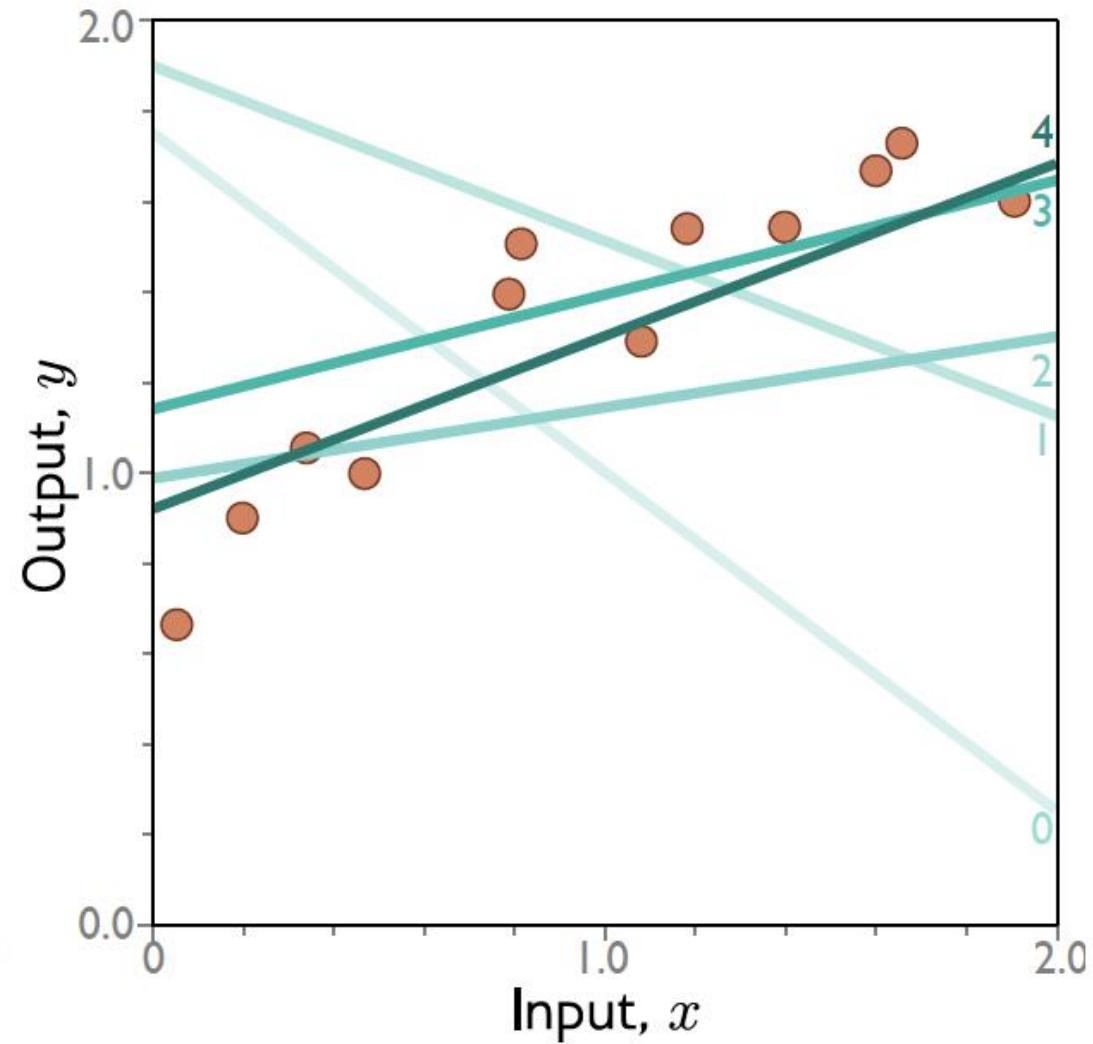
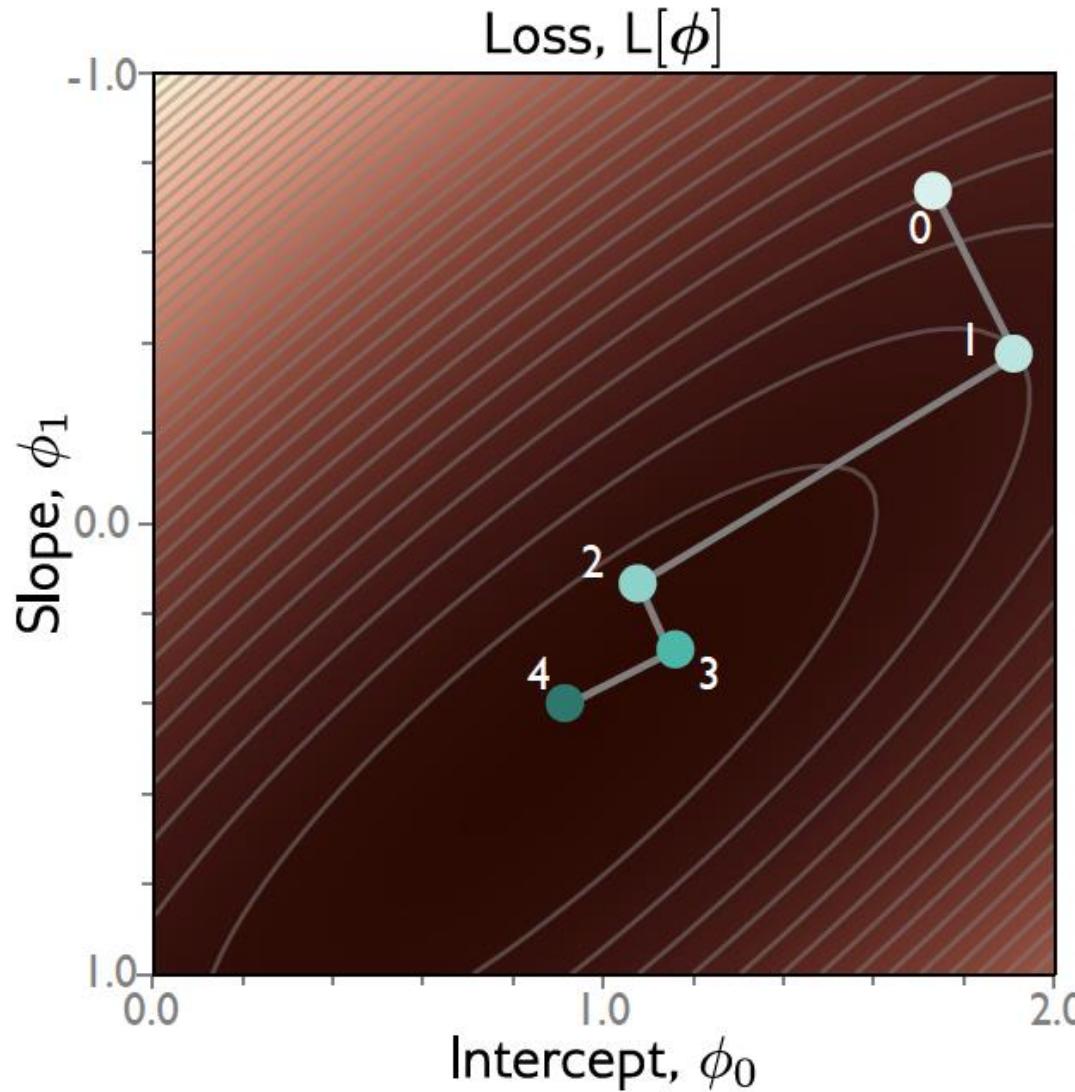
# Búsqueda de líneas (entre corchetes)



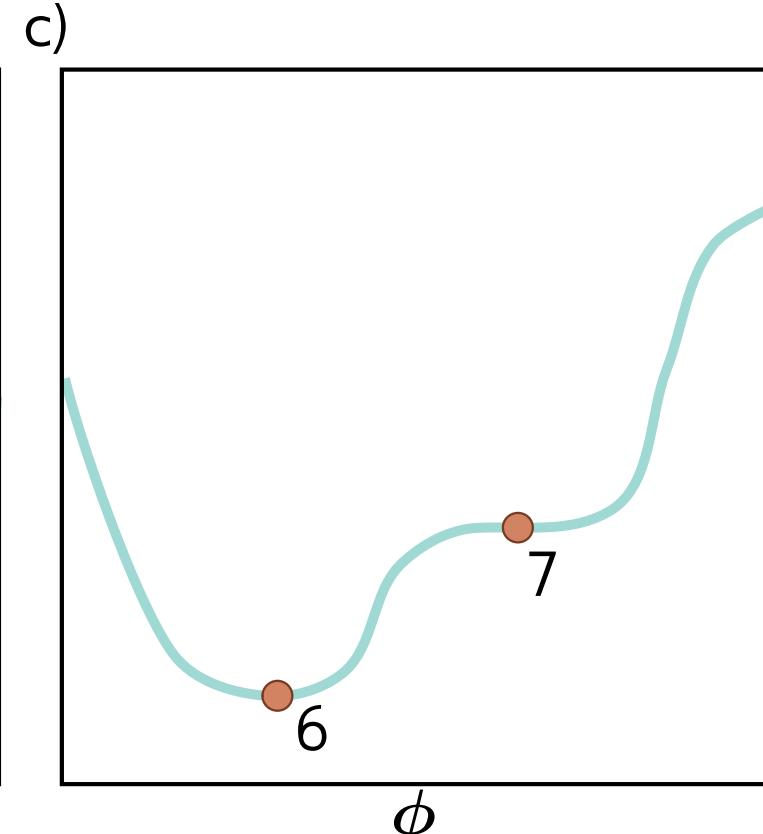
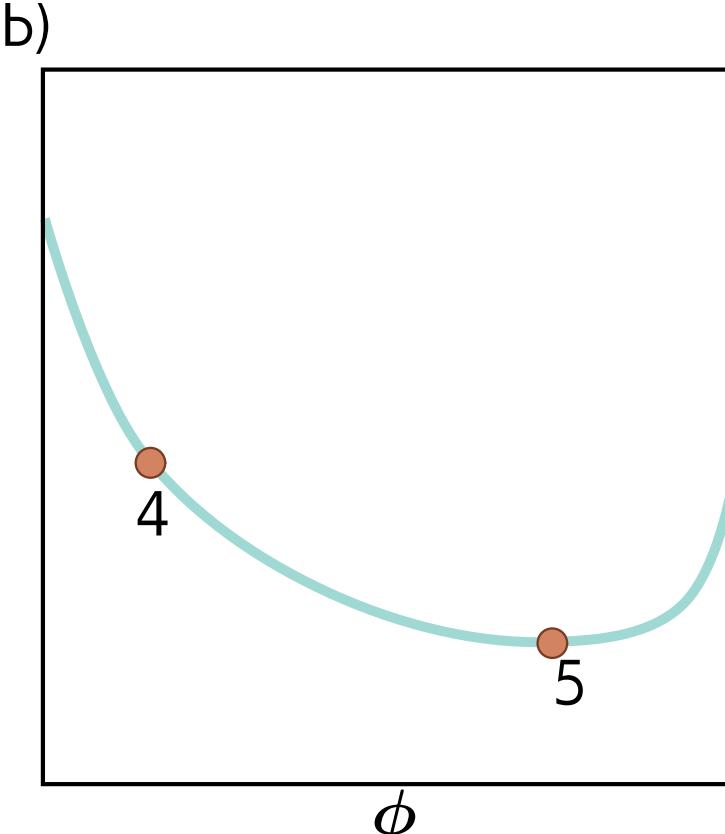
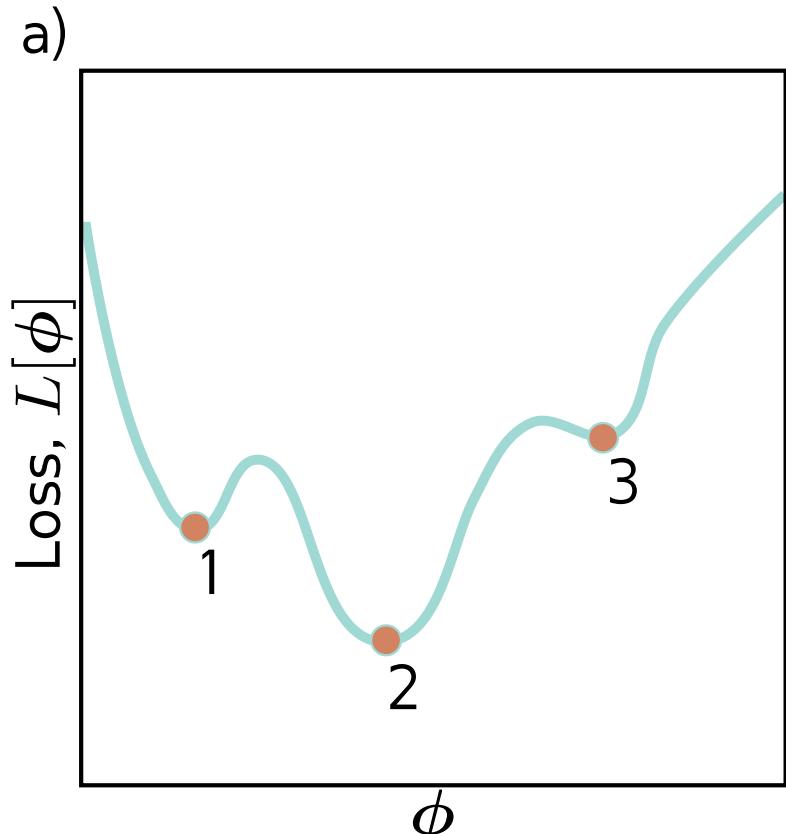
# Búsqueda de líneas (entre corchetes)



# Descenso gradual



# Problemas convexos

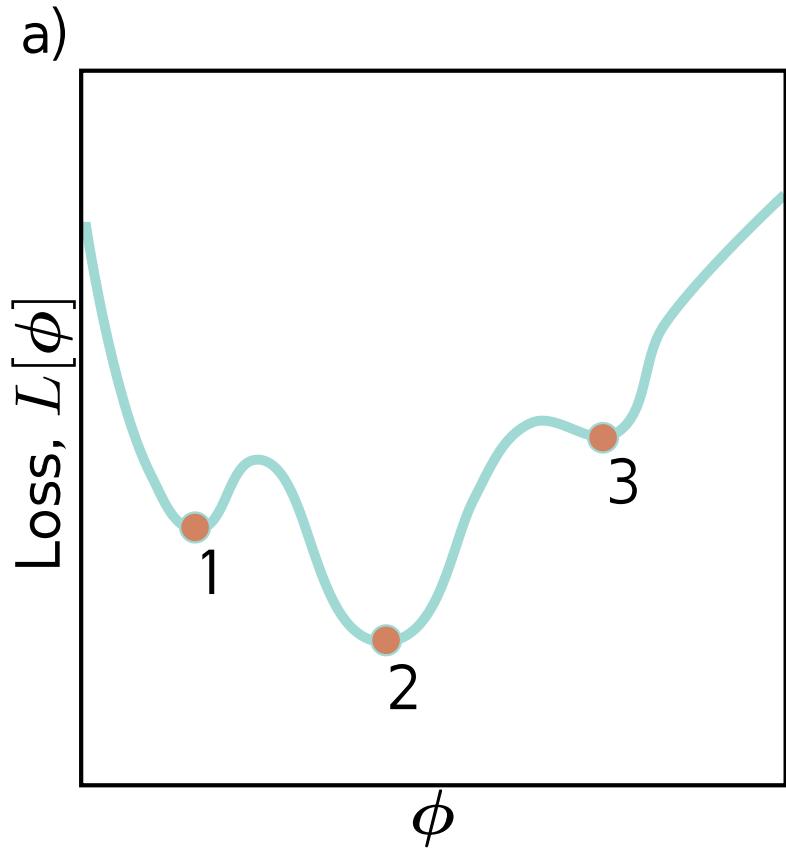


No convexo

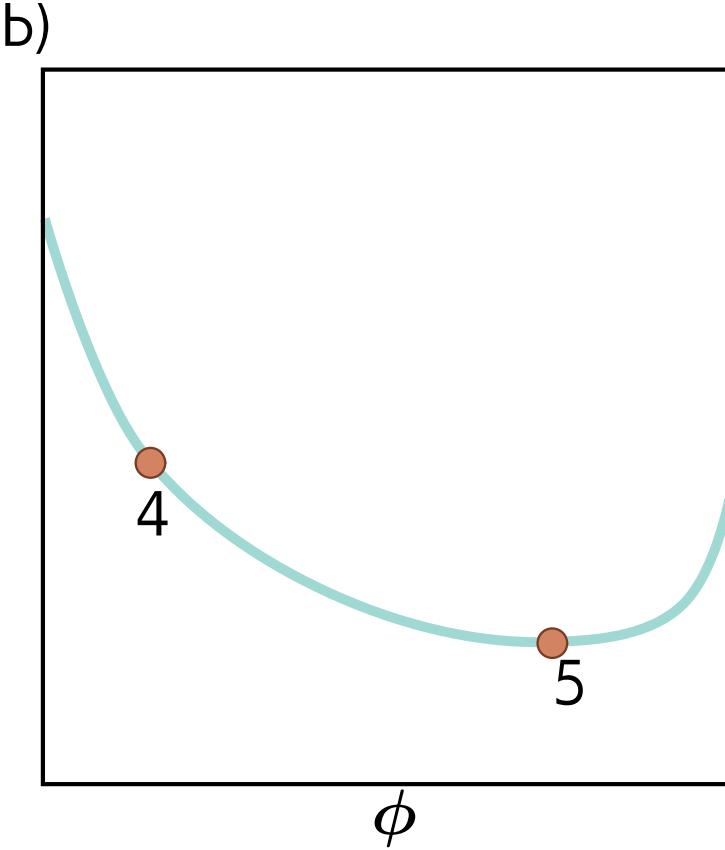
Convexo

No convexo

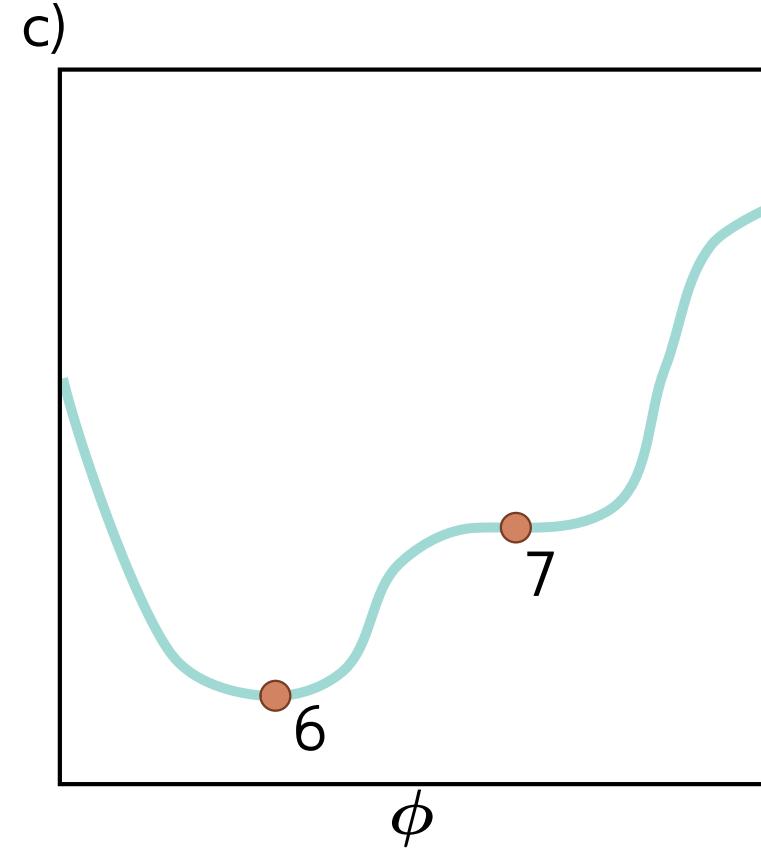
# Problemas convexos



No convexo



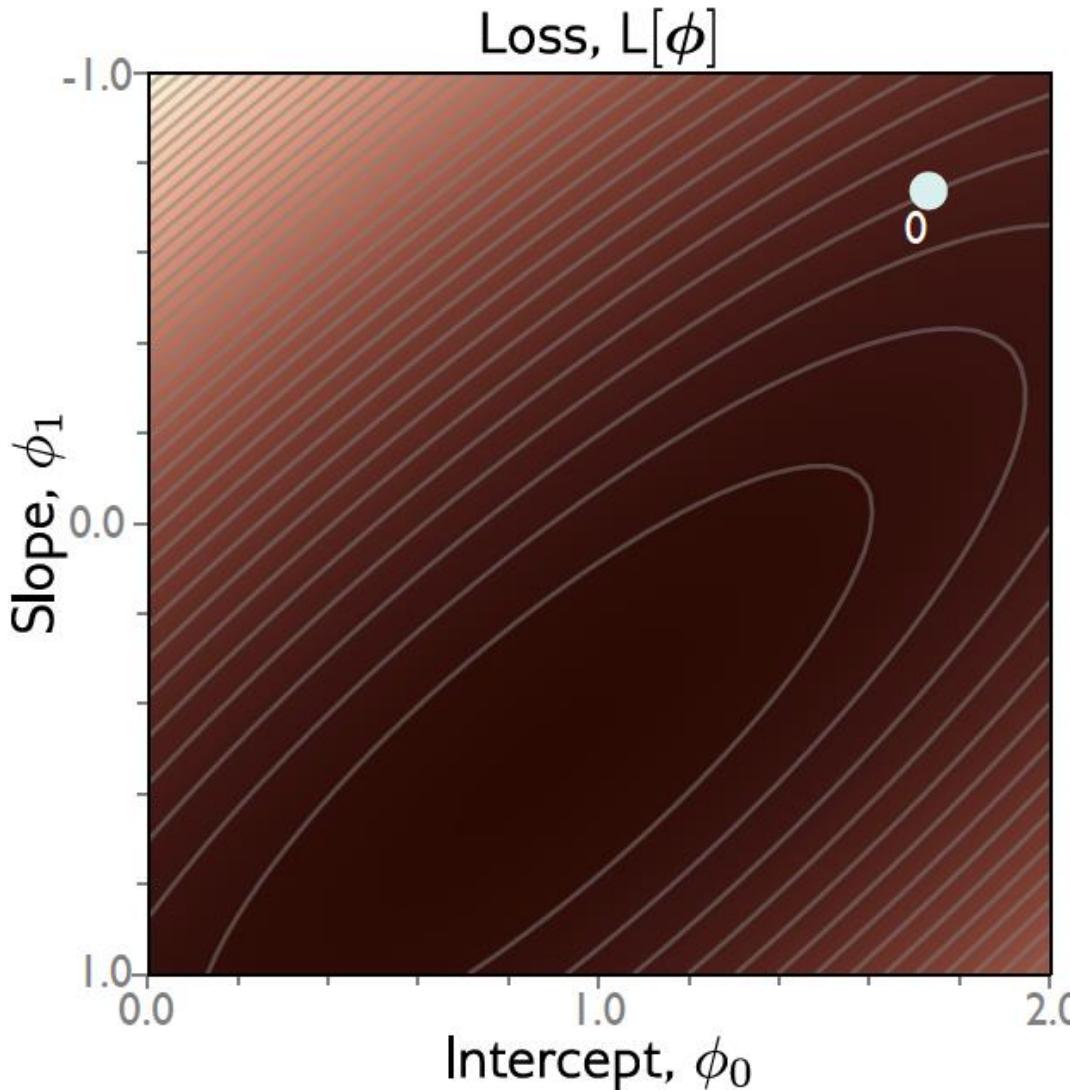
Convexo



No convexo

La prueba de convexidad es que 2<sup>nd</sup> la derivada es positiva en todas partes

# Convexidad en dimensiones superiores



La prueba de convexidad es que el determinante del hessiano (2<sup>da</sup> matriz derivada) es positivo en todas partes.

$$\mathbf{H}[\phi] = \begin{bmatrix} \frac{\partial^2 L}{\partial \phi_0^2} & \frac{\partial^2 L}{\partial \phi_0 \partial \phi_1} \\ \frac{\partial^2 L}{\partial \phi_1 \partial \phi_0} & \frac{\partial^2 L}{\partial \phi_1^2} \end{bmatrix}$$

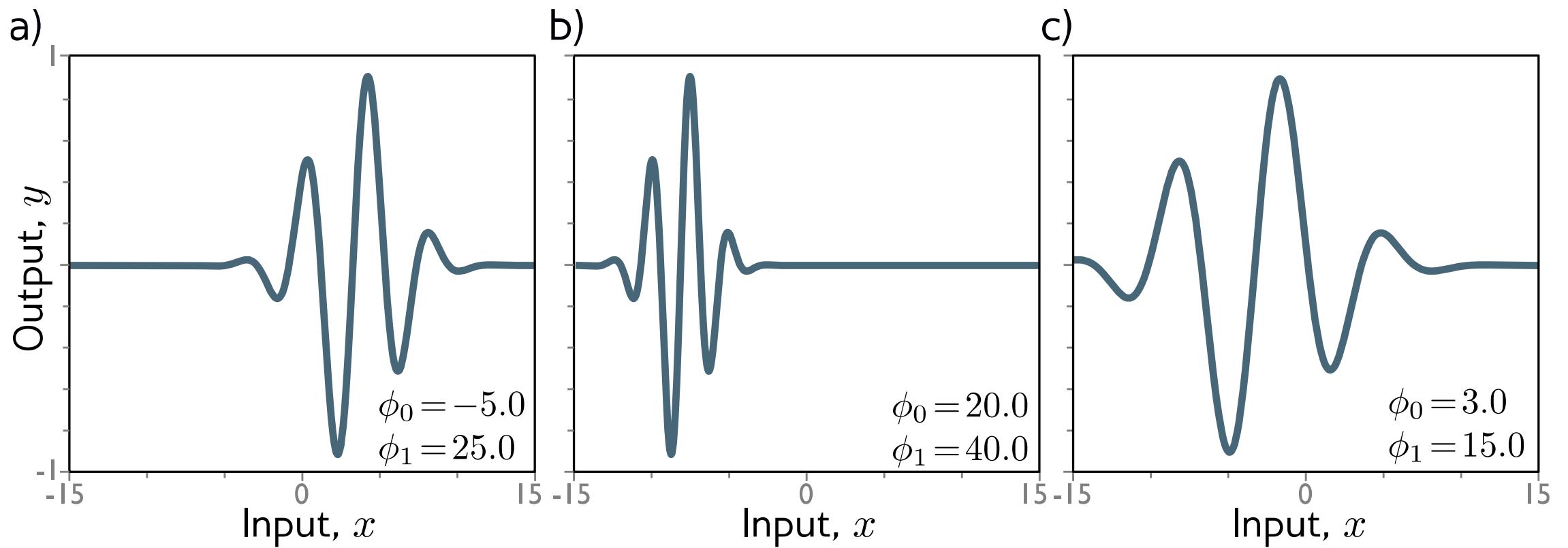
$$\mathbf{H}[\phi] = \frac{\partial^2 L}{\partial \phi_0^2} \frac{\partial^2 L}{\partial \phi_1^2} - \frac{\partial^2 L}{\partial \phi_0 \partial \phi_1} \frac{\partial^2 L}{\partial \phi_1 \partial \phi_0}$$

# Modelos de ajuste

- Matemáticas
- Algoritmo de descenso gradiente
- Ejemplo de regresión lineal
- Ejemplo de modelo de Gabor
- Descenso de gradiente estocástico
- Momentum
- Adam

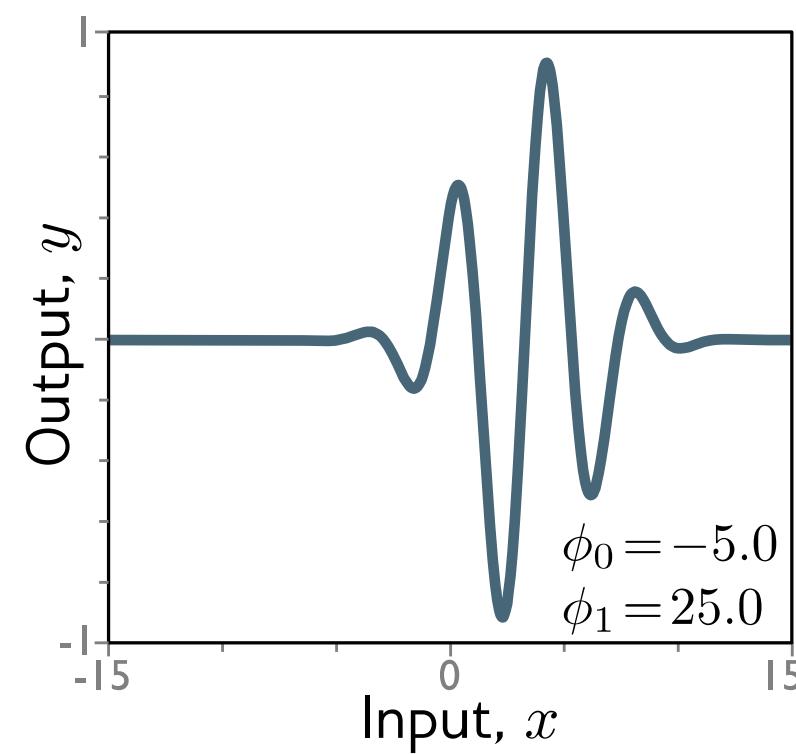
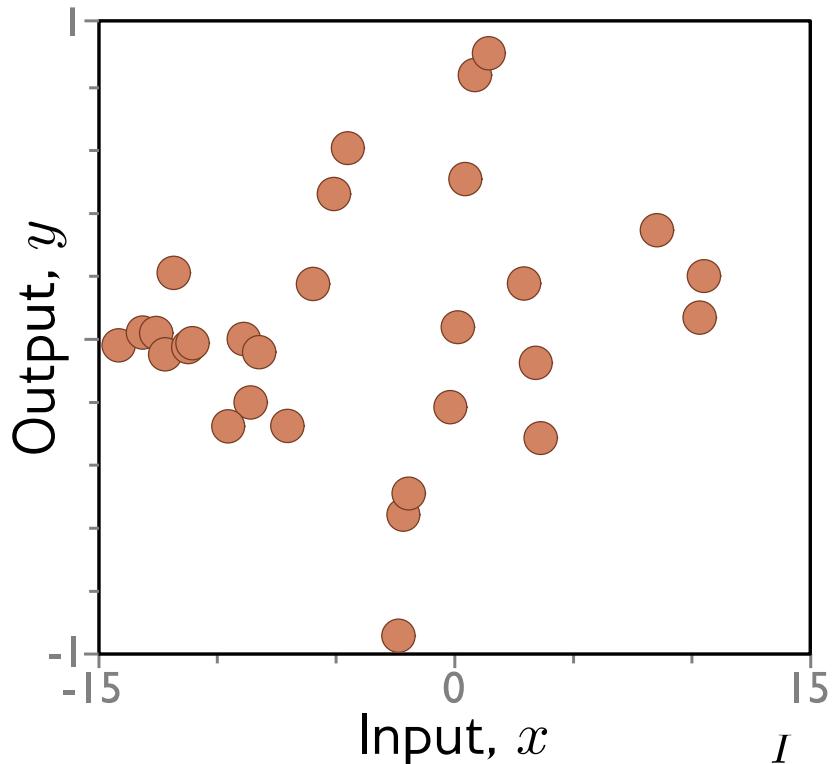
# Modelo Gabor

$$f[x, \phi] = \sin[\phi_0 + 0.06 \cdot \phi_1 x] \cdot \exp\left(-\frac{(\phi_0 + 0.06 \cdot \phi_1 x)^2}{8.0}\right)$$

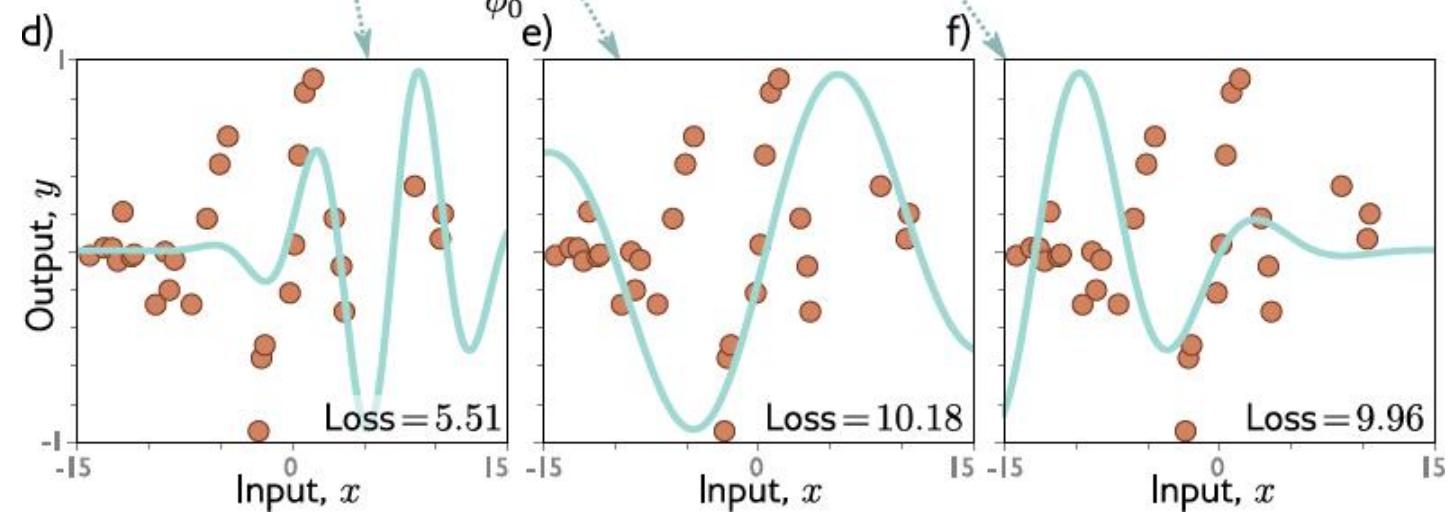
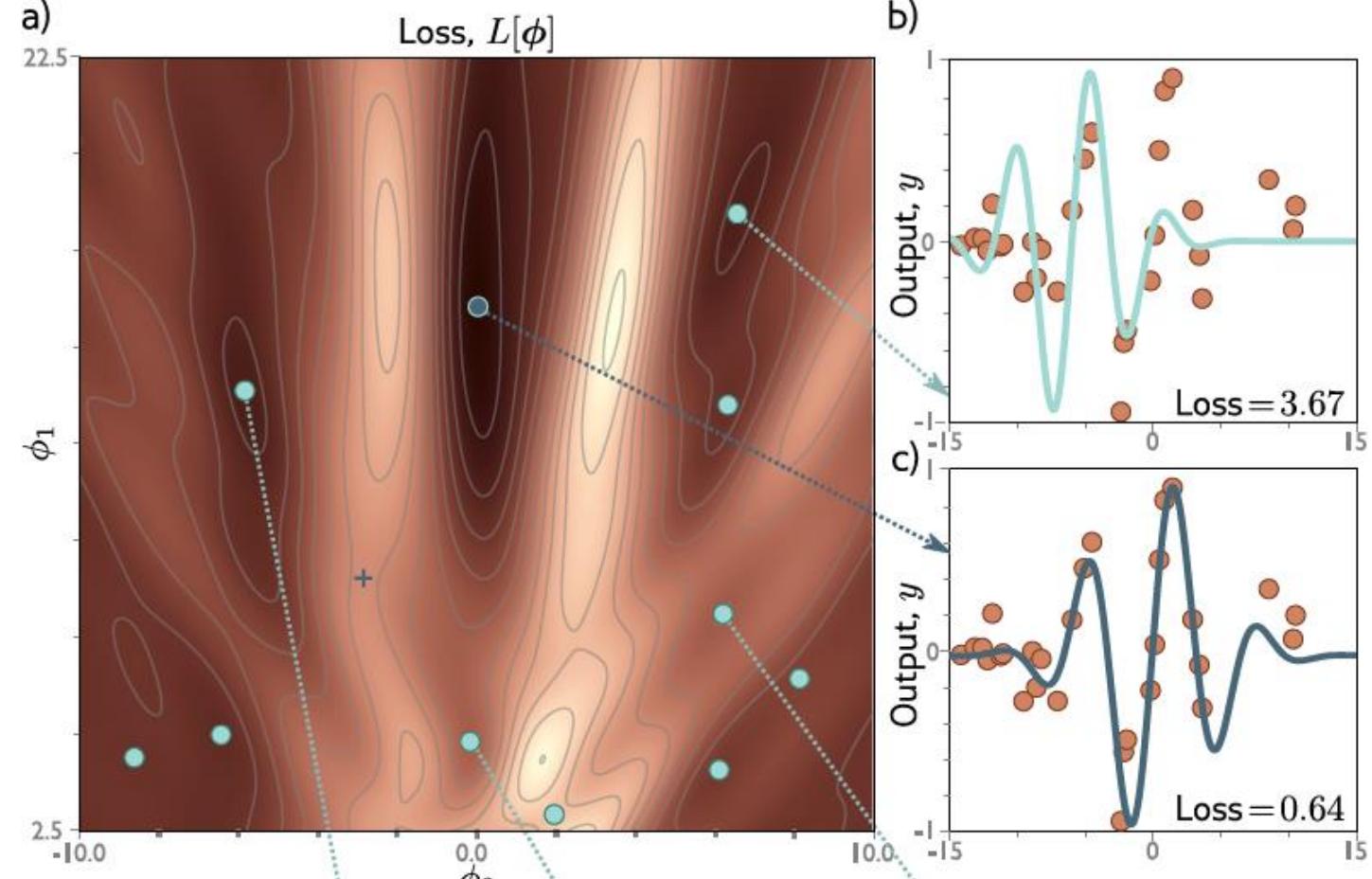


# Modelo Gabor

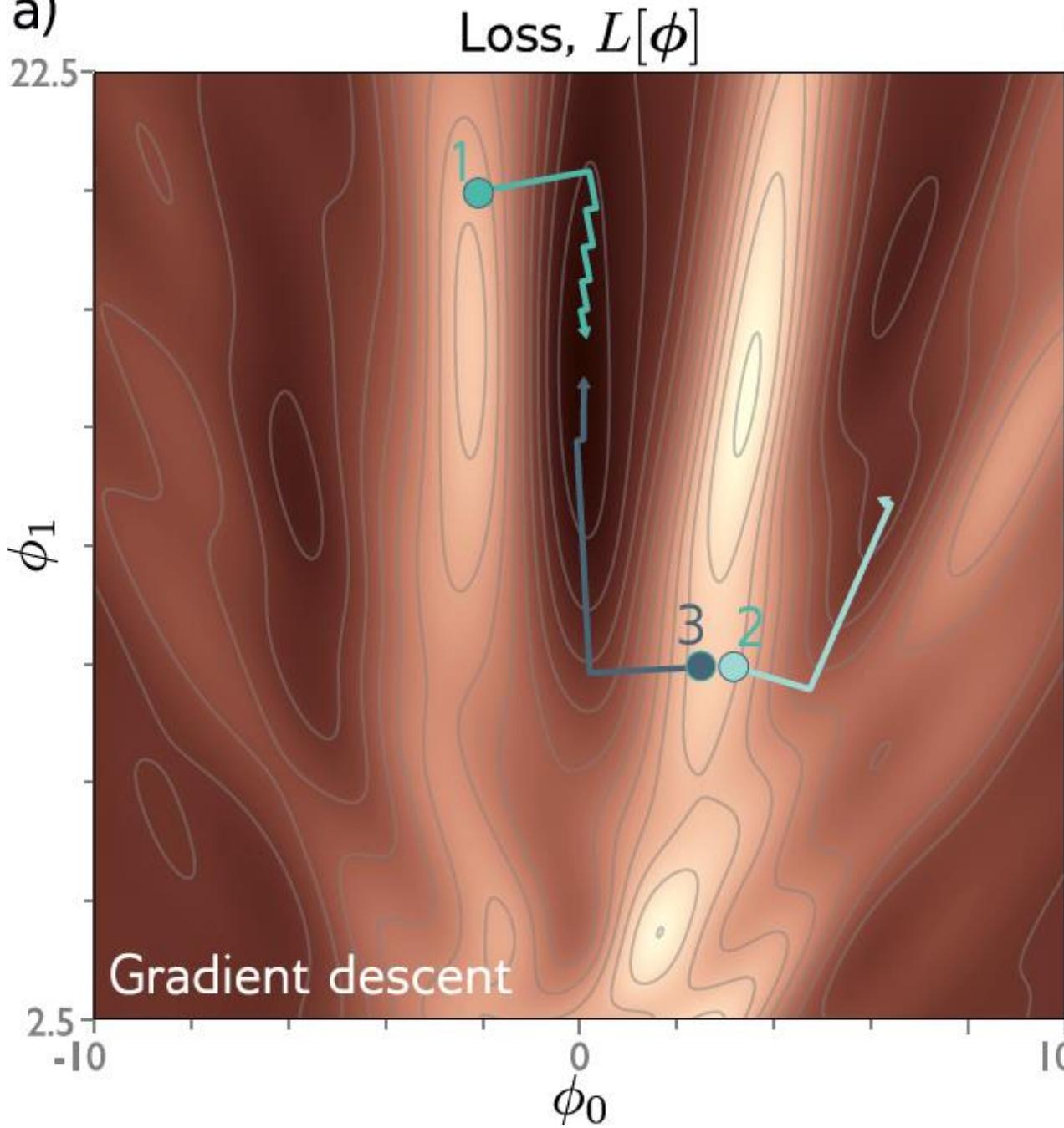
$$f[x, \phi] = \sin[\phi_0 + 0.06 \cdot \phi_1 x] \cdot \exp\left(-\frac{(\phi_0 + 0.06 \cdot \phi_1 x)^2}{8.0}\right)$$



$$L[\phi] = \sum_{i=1}^I (f[x_i, \phi] - y_i)^2$$



a)

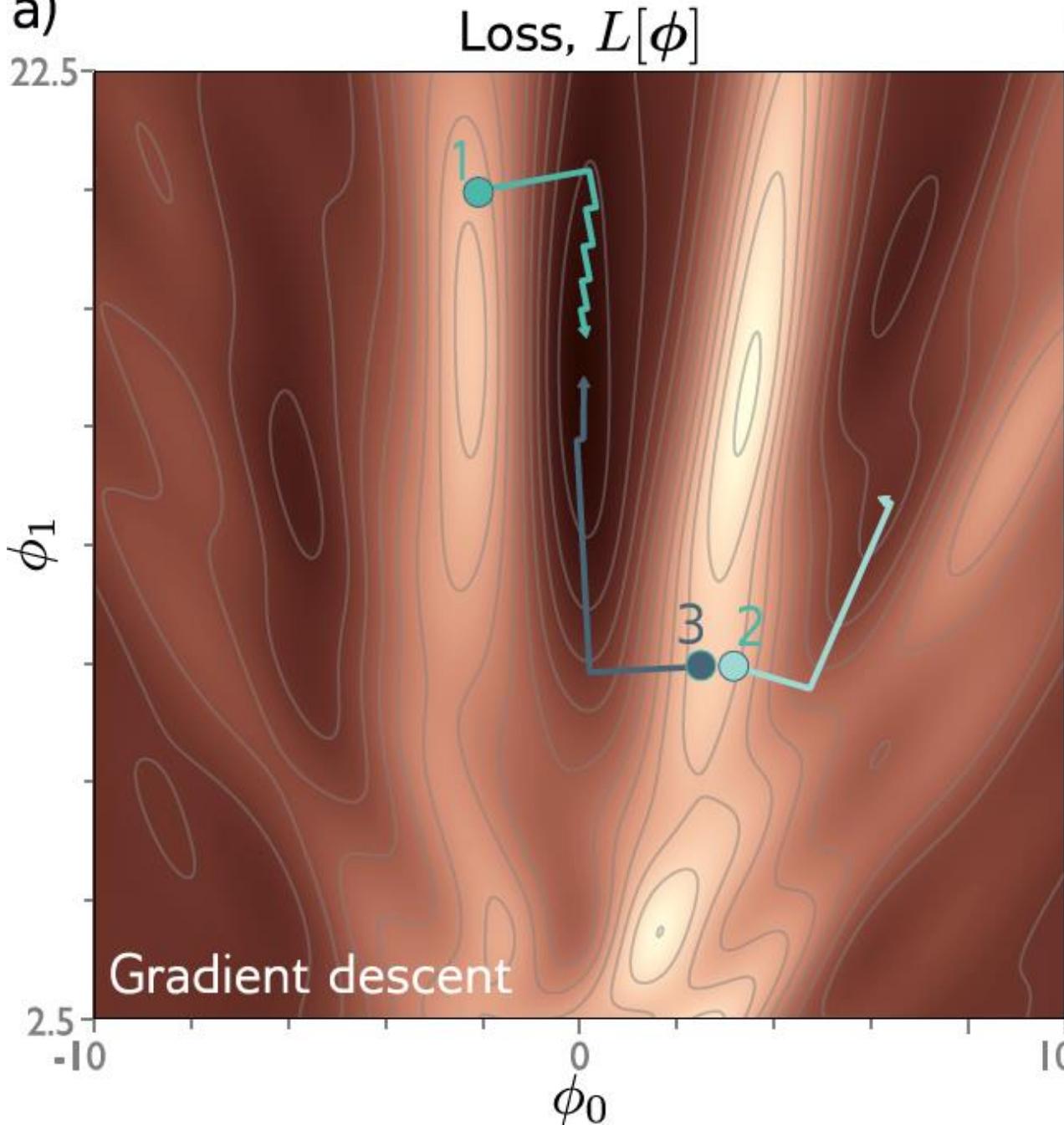


- El descenso gradiente llega al mínimo global si empezamos en el "valle" correcto
- En caso contrario, descenso a un mínimo local
- O quedarse atascado cerca de un punto de silla (*saddle point*) de montar

# Modelos de ajuste

- Matemáticas
- Algoritmo de descenso gradiente
- Ejemplo de regresión lineal
- Ejemplo de modelo de Gabor
- Descenso de gradiente estocástico
- Momentum
- Adam

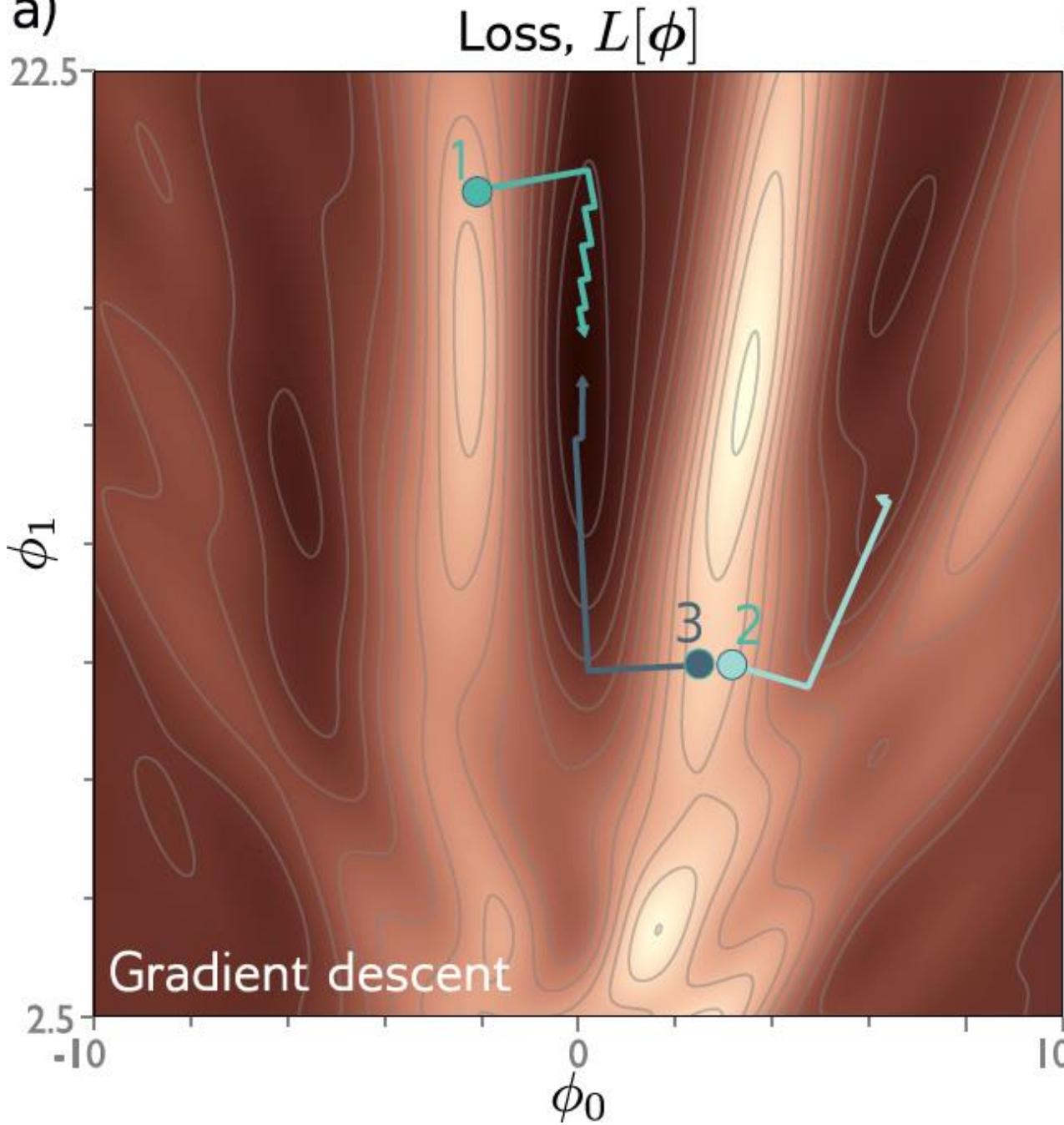
a)



IDEA: añadir ruido

- Descenso de gradiente estocástico
- Calcular el gradiente basándose sólo en un subconjunto de puntos: un **minilote** (*minibatch*)
- Trabajar a través del muestreo de conjuntos de datos sin reemplazo
- Una pasada por los datos se denomina **época** (*epoch*)

a)



Descenso de gradiente estocástico

Antes (descenso por lotes completos)

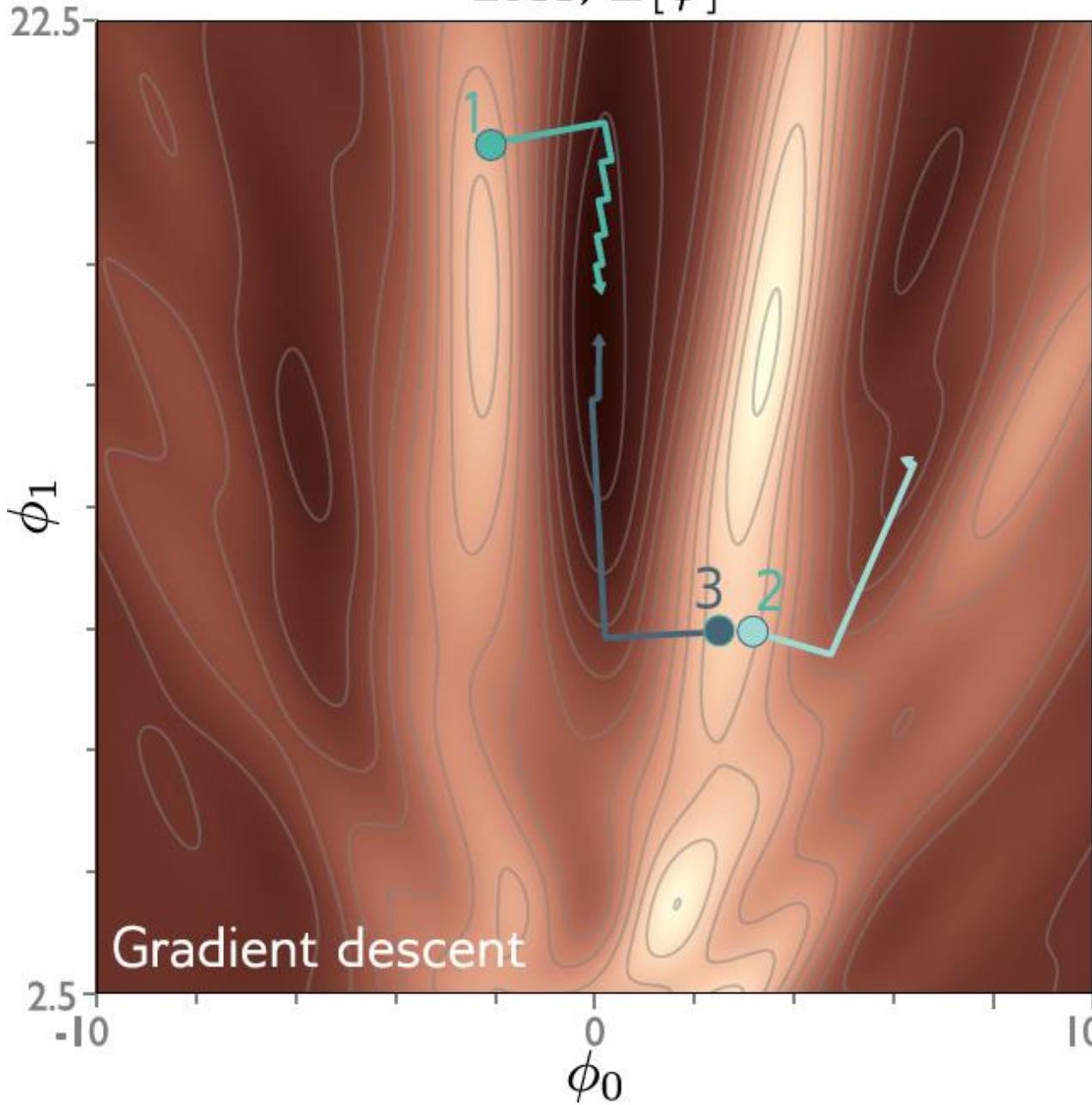
$$\phi_{t+1} \leftarrow \phi_t - \alpha \sum_{i=1}^I \frac{\partial \ell_i[\phi_t]}{\partial \phi},$$

Después (SGD)

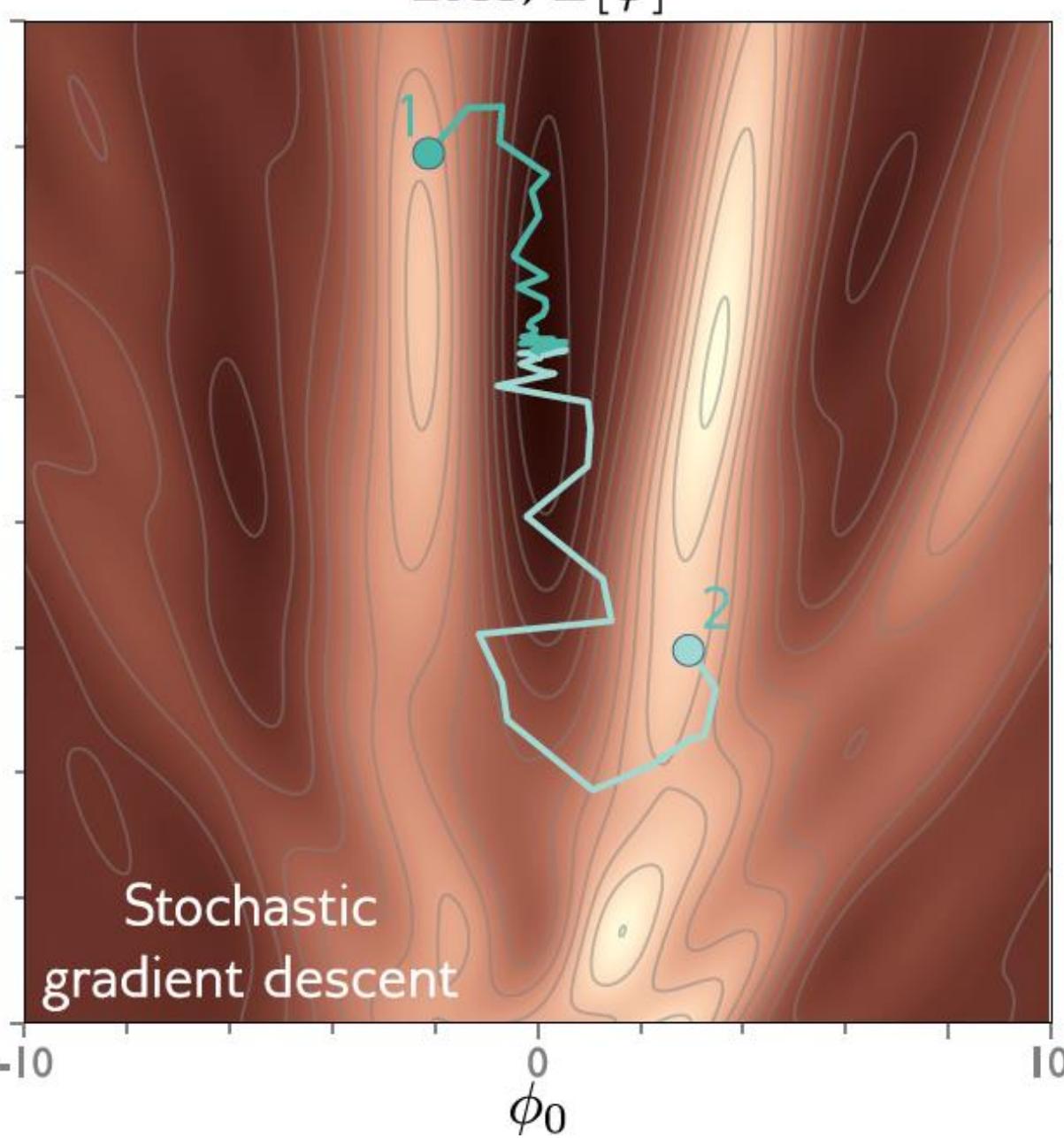
$$\phi_{t+1} \leftarrow \phi_t - \alpha \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\phi_t]}{\partial \phi},$$

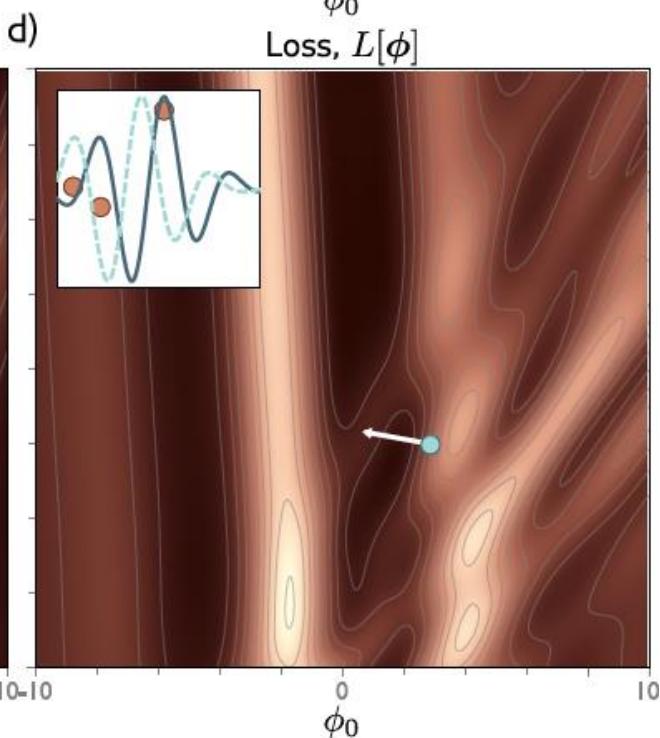
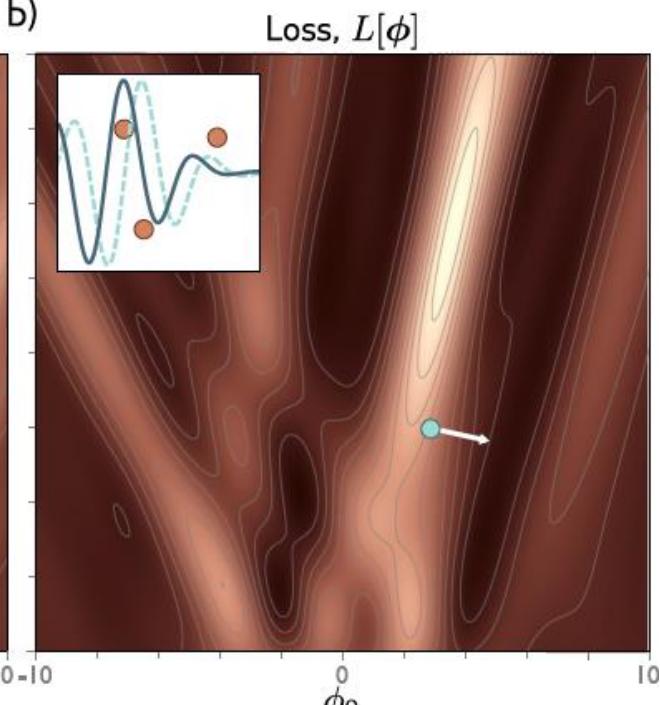
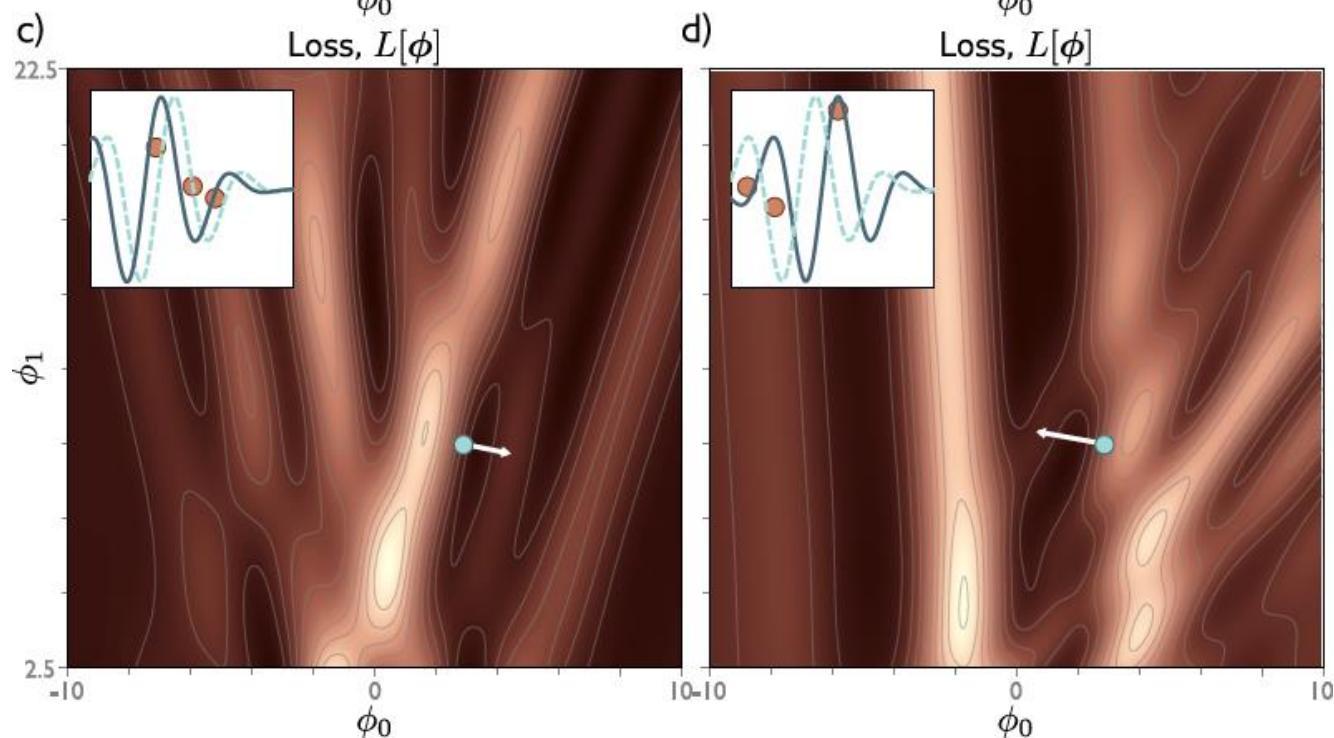
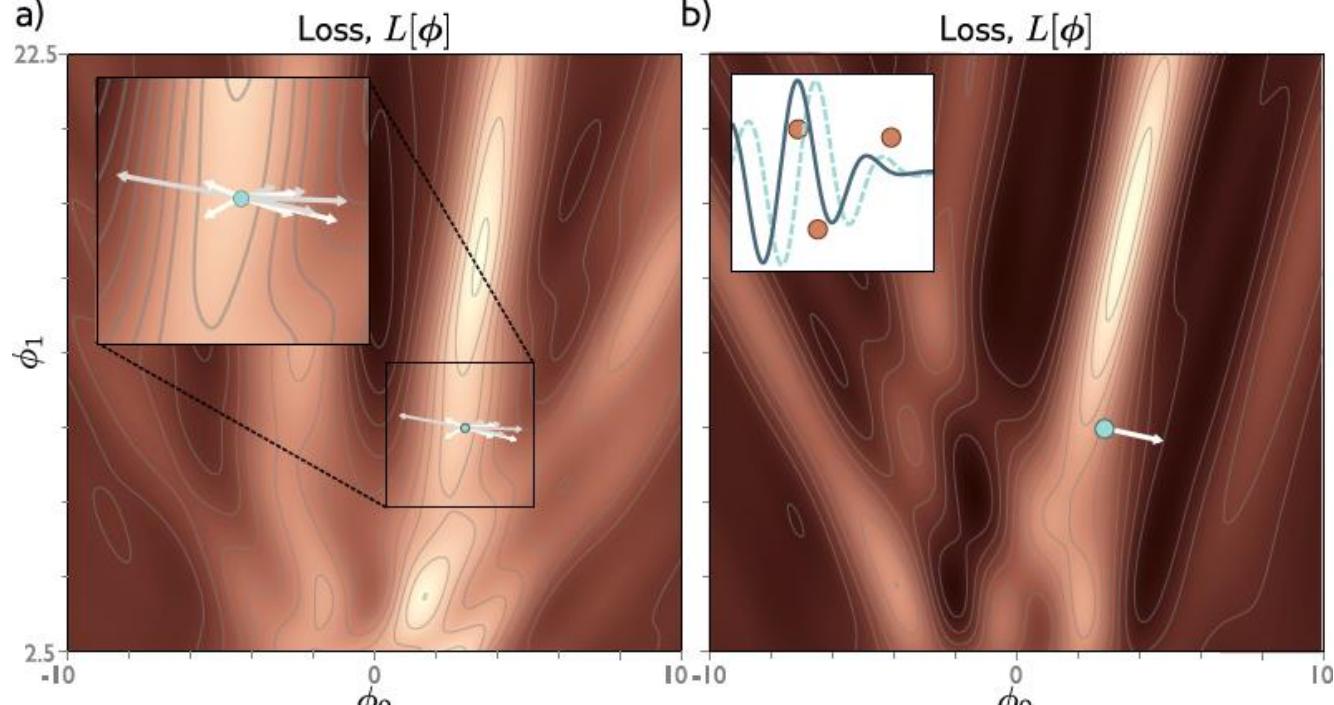
Tasa de aprendizaje fija

a)

Loss,  $L[\phi]$ 

b)

Loss,  $L[\phi]$ 



# Propiedades de SGD

- Puede escapar de mínimos locales
- Añade ruido, pero sigue siendo una actualización razonable, ya que se basa en parte de los datos.
- Utiliza todos los datos por igual
- Menos costoso desde el punto de vista computacional
- Parece encontrar mejores soluciones
- No converge en el sentido tradicional
- **Programación del threshold ( $\alpha$ ) de aprendizaje:** disminución del threshold de aprendizaje a lo largo del tiempo

# Modelos de ajuste

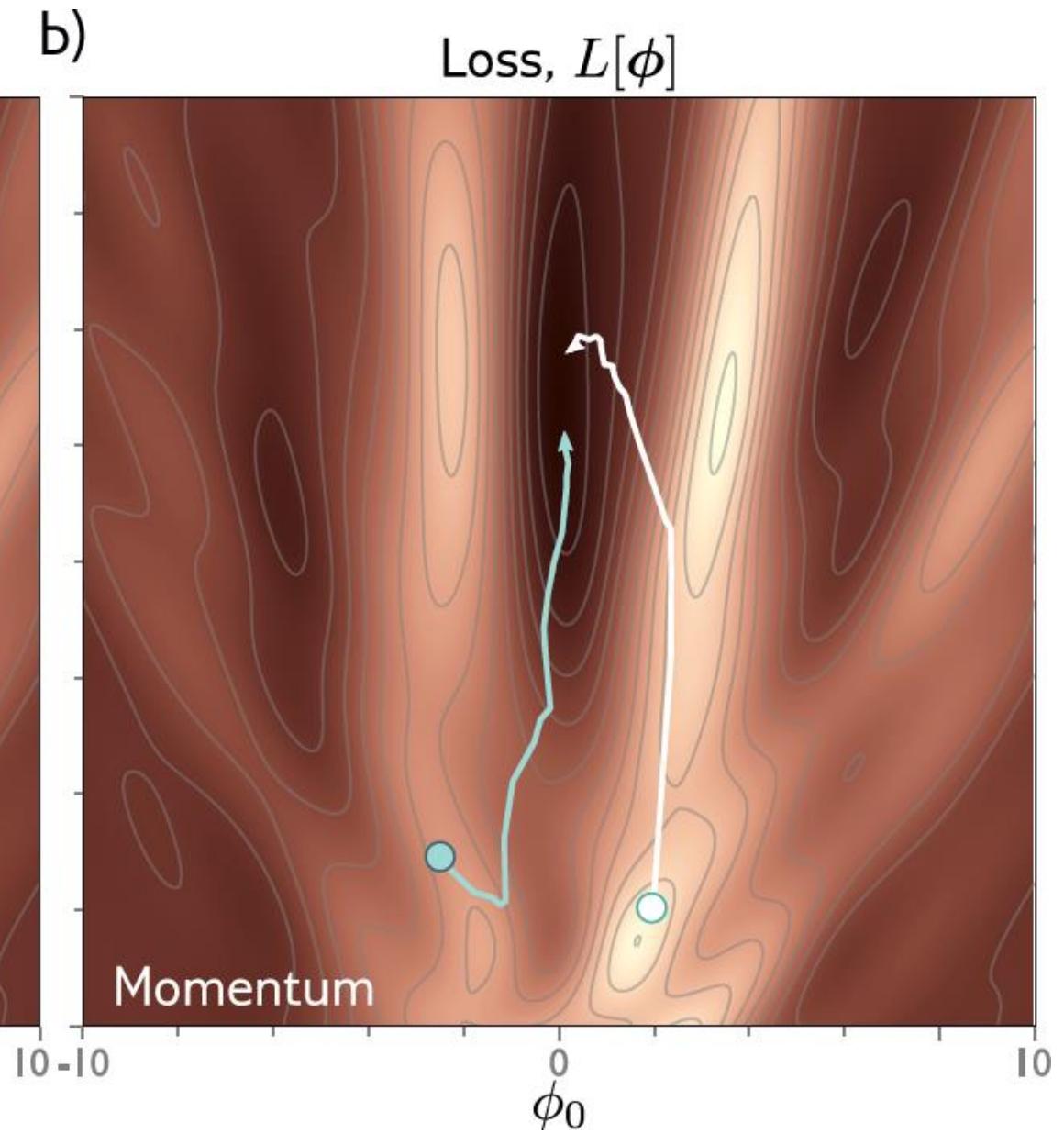
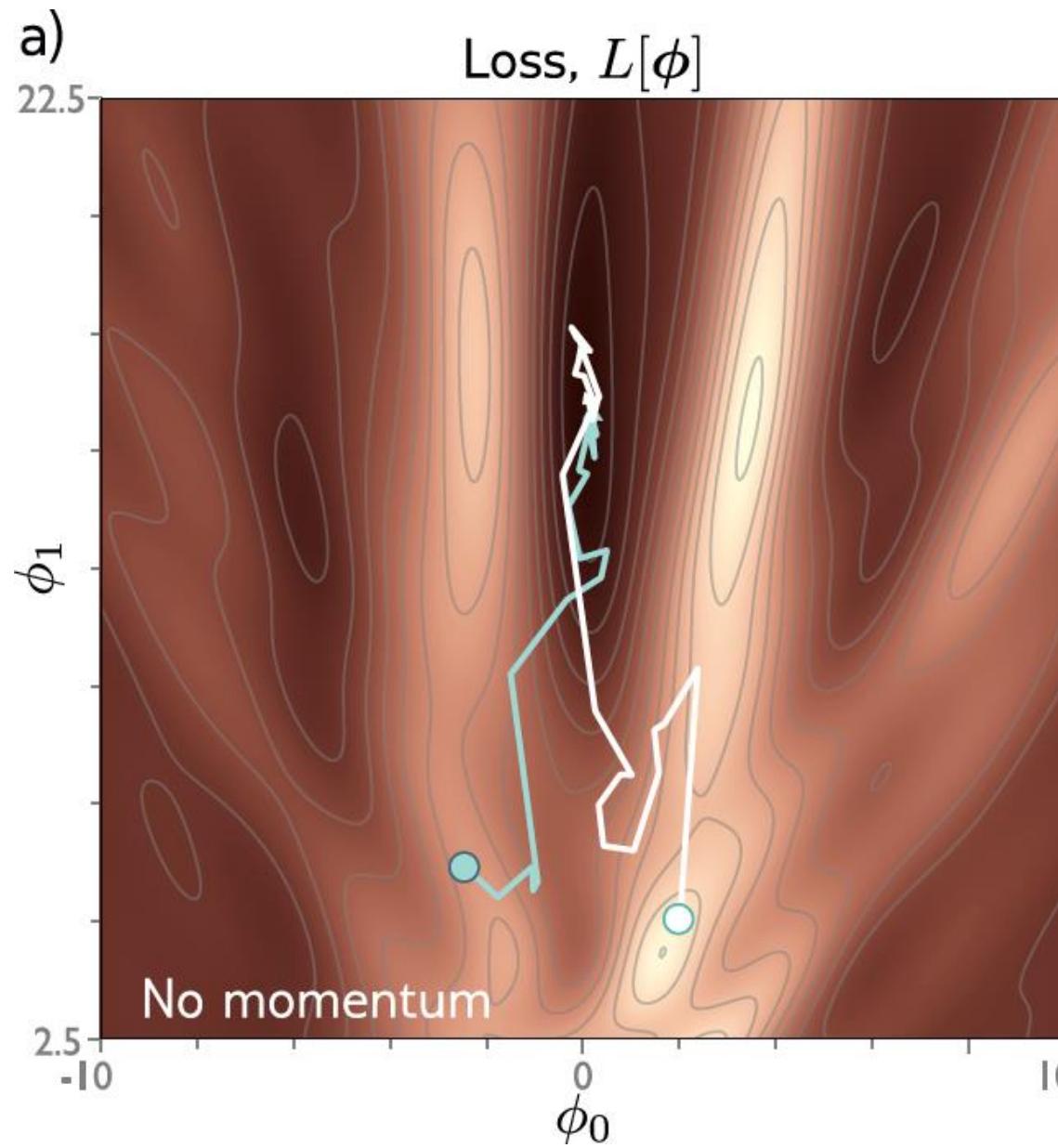
- Matemáticas
- Algoritmo de descenso gradiente
- Ejemplo de regresión lineal
- Ejemplo de modelo de Gabor
- Descenso de gradiente estocástico
- Momentum
- Adam

# Momentum

- Suma ponderada de este gradiente y el gradiente anterior

$$\mathbf{m}_{t+1} \leftarrow \beta \cdot \mathbf{m}_t + (1 - \beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\phi_t]}{\partial \phi}$$

$$\phi_{t+1} \leftarrow \phi_t - \alpha \cdot \mathbf{m}_{t+1}$$



# Nesterov acelera el momentum

- El momentum es una especie de predicción de hacia dónde vamos.

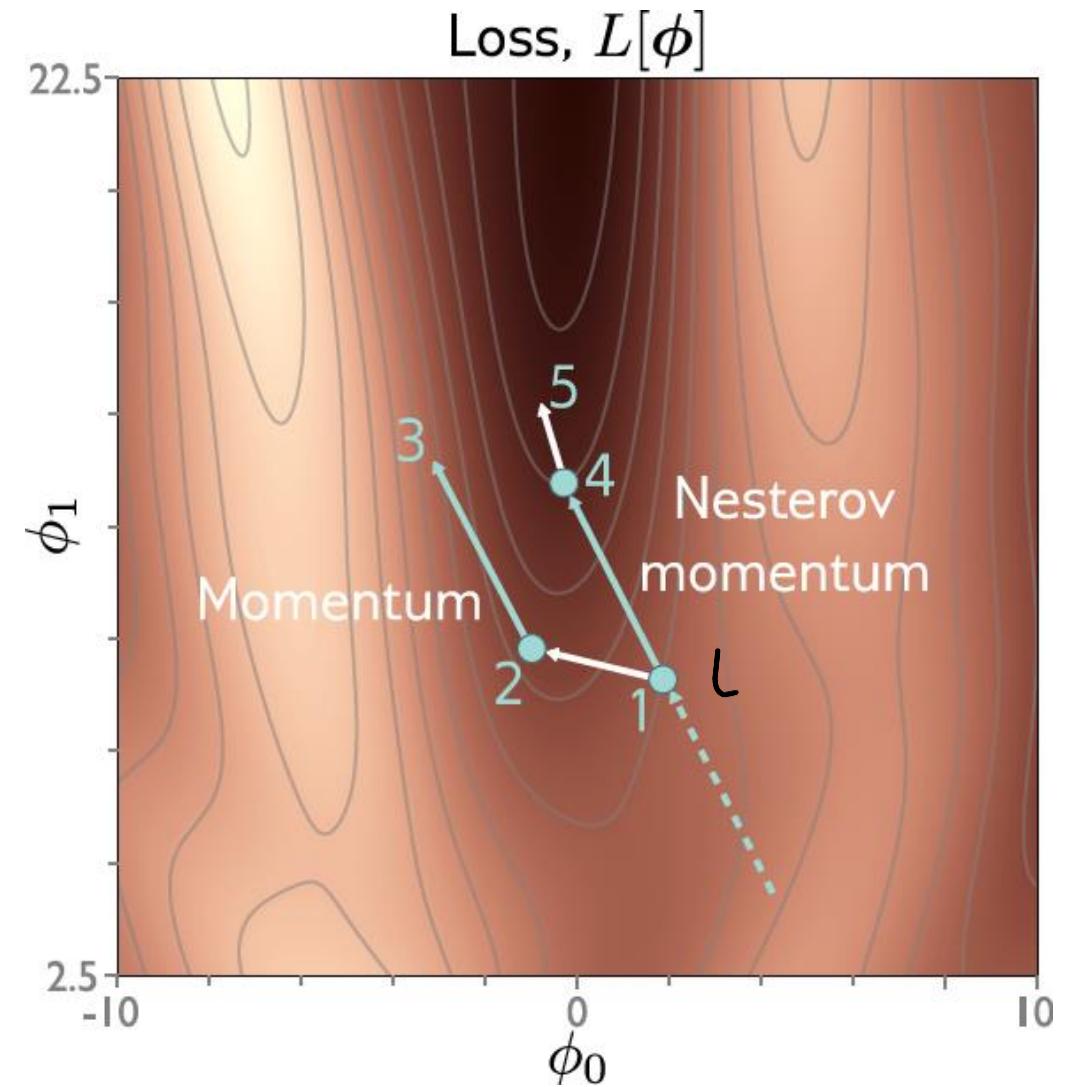
$$\mathbf{m}_{t+1} \leftarrow \beta \cdot \mathbf{m}_t + (1 - \beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\phi_t]}{\partial \phi}$$

$$\phi_{t+1} \leftarrow \phi_t - \alpha \cdot \mathbf{m}_{t+1}$$

- Muévase en la dirección prevista, **ENTONCES**, mida el gradiente

$$\mathbf{m}_{t+1} \leftarrow \beta \cdot \mathbf{m}_t + (1 - \beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\phi_t - \alpha \cdot \mathbf{m}_t]}{\partial \phi}$$

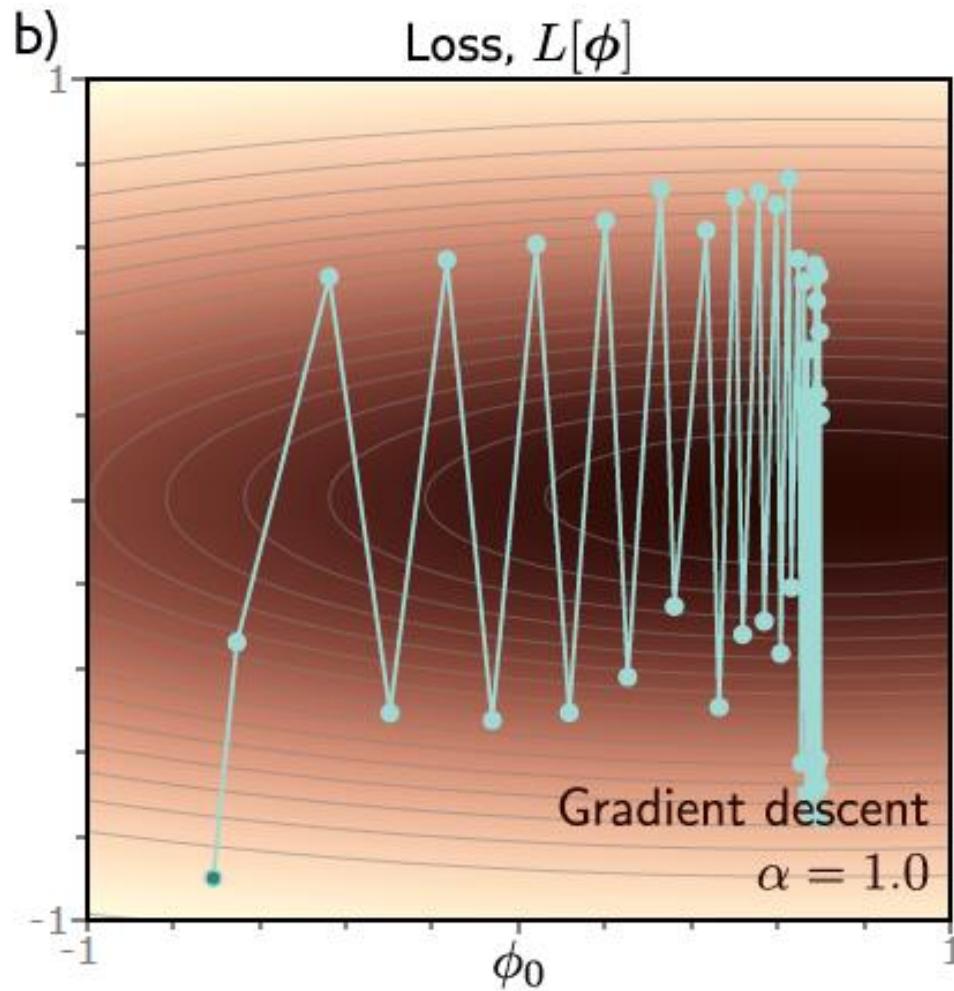
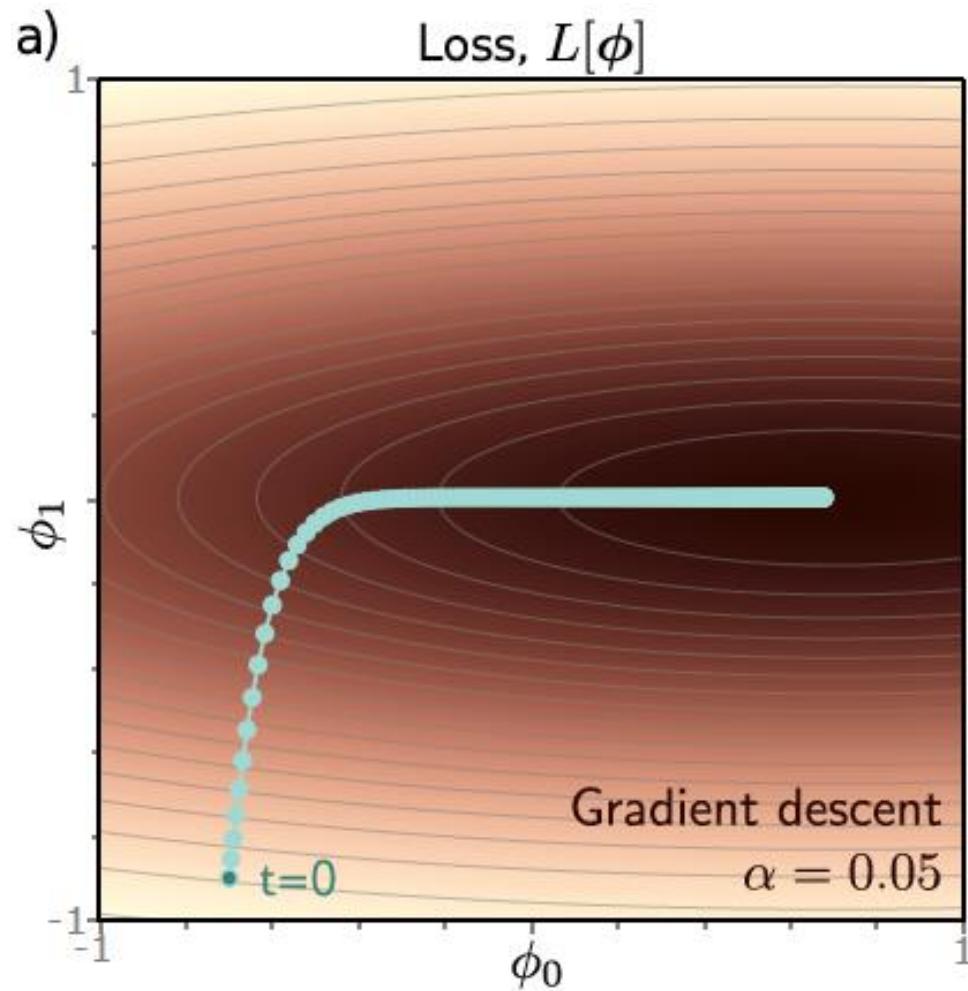
$$\phi_{t+1} \leftarrow \phi_t - \alpha \cdot \mathbf{m}_{t+1}$$



# Modelos de ajuste

- Matemáticas
- Algoritmo de descenso gradiente
- Ejemplo de regresión lineal
- Ejemplo de modelo de Gabor
- Descenso de gradiente estocástico
- Momentum
- Adam

# Estimación adaptativa del momento (Adam)



# Gradientes normalizados

- Medir la media y el gradiente cuadrático puntual

$$\mathbf{m}_{t+1} \leftarrow \frac{\partial L[\phi_t]}{\partial \phi}$$

$$\mathbf{v}_{t+1} \leftarrow \frac{\partial L[\phi_t]}{\partial \phi}^2$$

- Normalizar:

$$\phi_{t+1} \leftarrow \phi_t - \alpha \cdot \frac{\mathbf{m}_{t+1}}{\sqrt{\mathbf{v}_{t+1}} + \epsilon}$$

# Gradientes normalizados

- Medir la media y el gradiente cuadrático puntual

$$\mathbf{m}_{t+1} \leftarrow \frac{\partial L[\phi_t]}{\partial \phi}$$

$$\mathbf{v}_{t+1} \leftarrow \frac{\partial L[\phi_t]}{\partial \phi}^2$$

- Normalizar:

$$\phi_{t+1} \leftarrow \phi_t - \alpha \cdot \frac{\mathbf{m}_{t+1}}{\sqrt{\mathbf{v}_{t+1}} + \epsilon}$$

$$\mathbf{m}_{t+1} = \begin{bmatrix} 3.0 \\ -2.0 \\ 5.0 \end{bmatrix}$$

$$\mathbf{v}_{t+1} = \begin{bmatrix} 9.0 \\ 4.0 \\ 25.0 \end{bmatrix}$$

$$\frac{\mathbf{m}_{t+1}}{\sqrt{\mathbf{v}_{t+1}} + \epsilon} = \begin{bmatrix} 1.0 \\ -1.0 \\ 1.0 \end{bmatrix}$$

# Gradientes normalizados

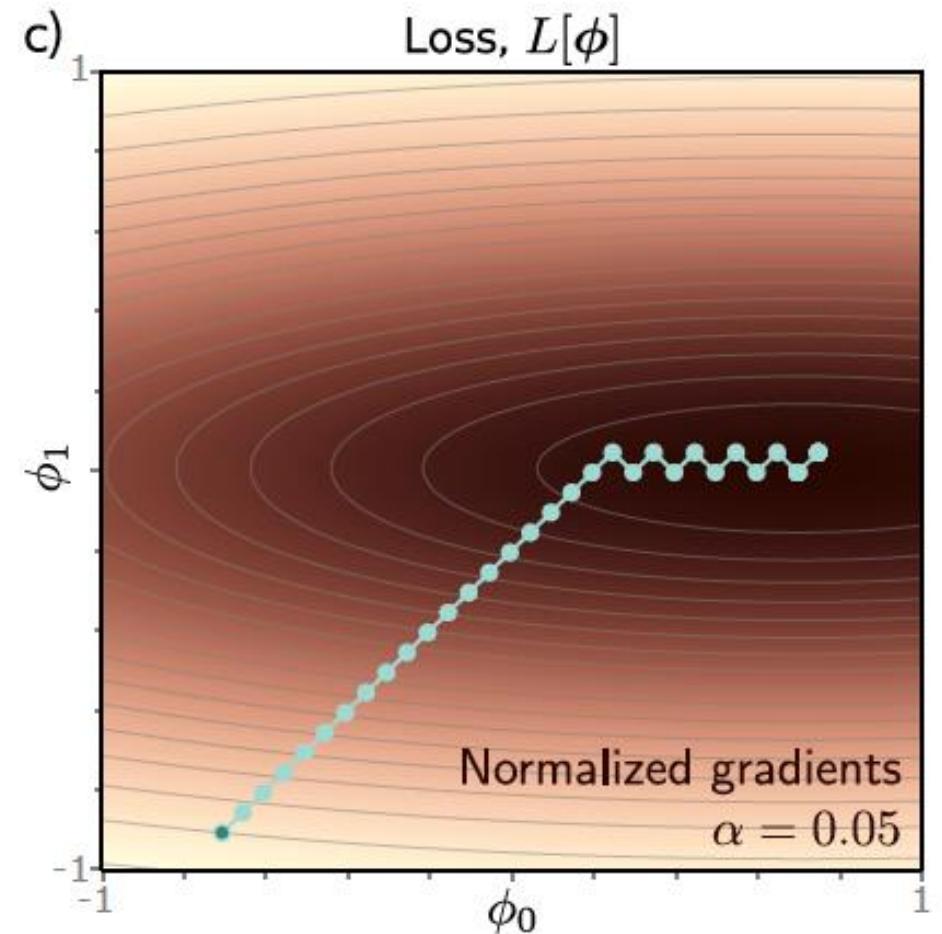
- Medir la media y el gradiente cuadrático puntual

$$\mathbf{m}_{t+1} \leftarrow \frac{\partial L[\phi_t]}{\partial \phi}$$

$$\mathbf{v}_{t+1} \leftarrow \frac{\partial L[\phi_t]}{\partial \phi}^2$$

- Normalizar:

$$\phi_{t+1} \leftarrow \phi_t - \alpha \cdot \frac{\mathbf{m}_{t+1}}{\sqrt{\mathbf{v}_{t+1}}} + \epsilon$$



# Estimación adaptativa del momento (Adam)

- Calcular la media y los gradientes puntuales al cuadrado con el impulso

$$\mathbf{m}_{t+1} \leftarrow \beta \cdot \mathbf{m}_t + (1 - \beta) \frac{\partial L[\phi_t]}{\partial \phi}$$

$$\mathbf{v}_{t+1} \leftarrow \gamma \cdot \mathbf{v}_t + (1 - \gamma) \left( \frac{\partial L[\phi_t]}{\partial \phi} \right)^2$$

- Moderado cerca del inicio de la secuencia

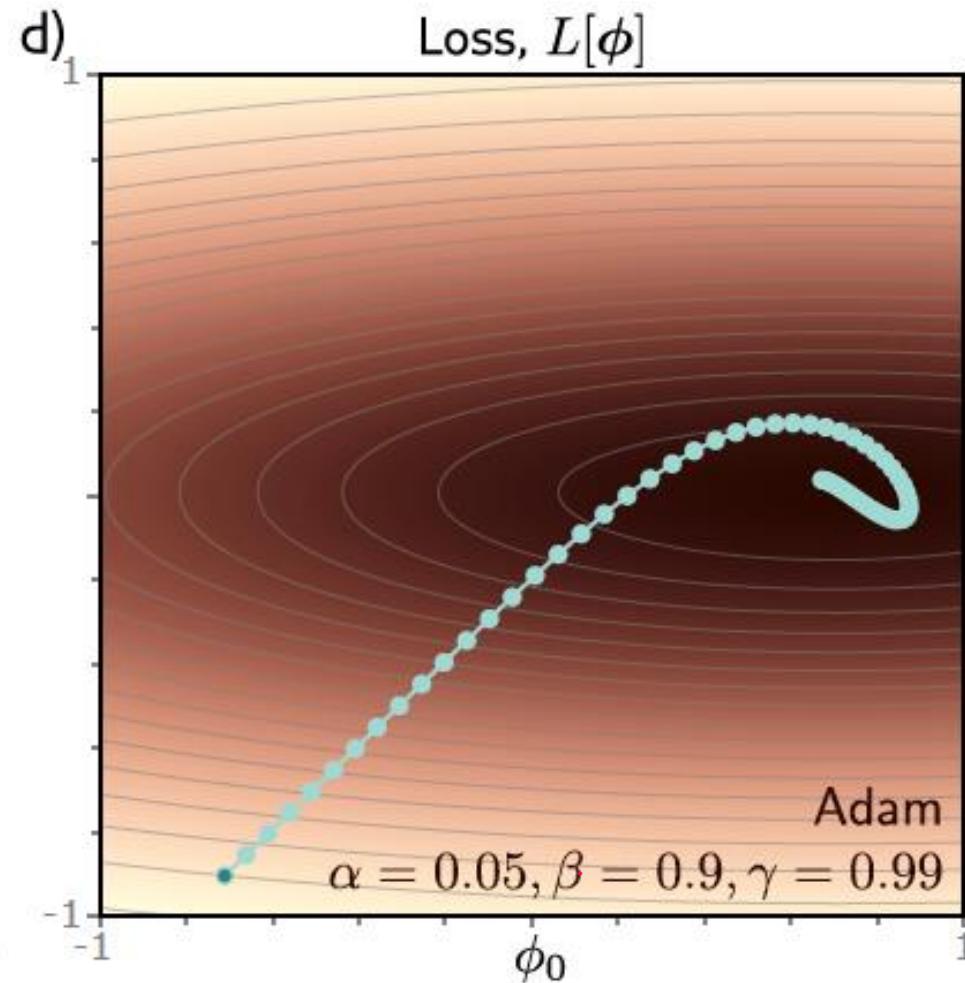
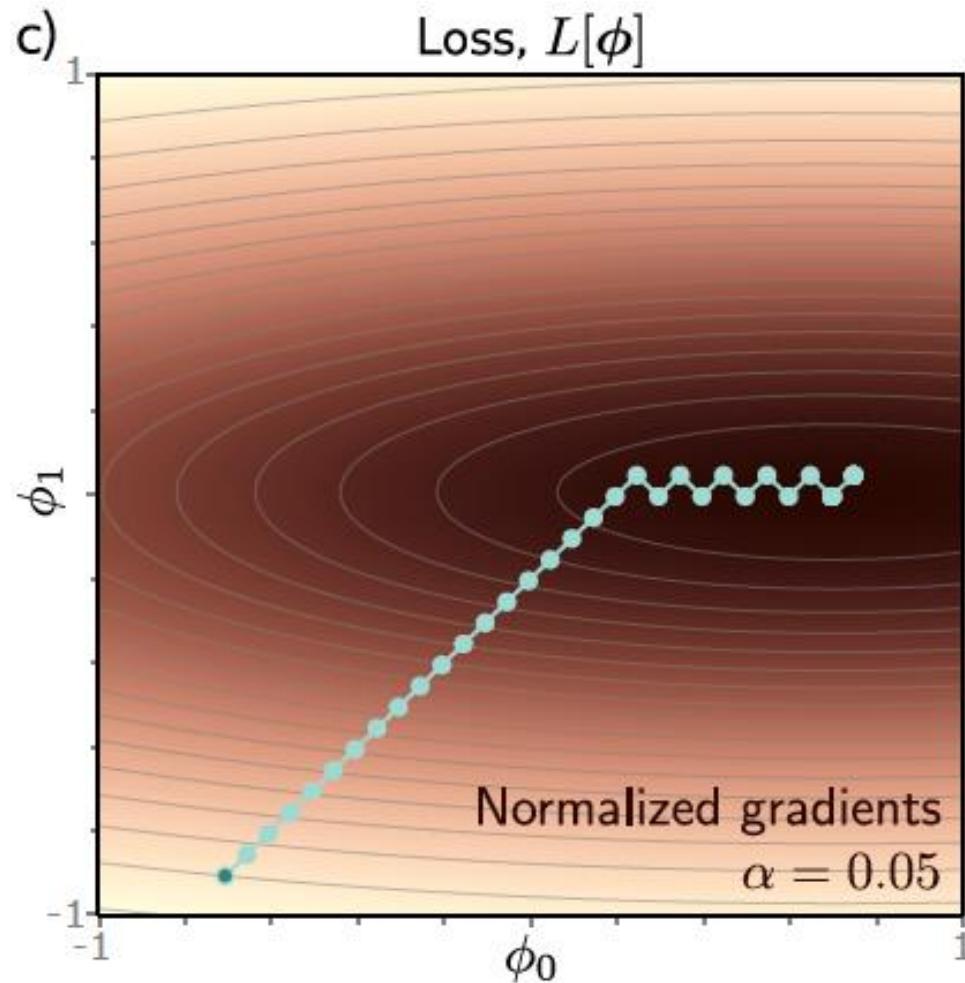
$$\tilde{\mathbf{m}}_{t+1} \leftarrow \frac{\mathbf{m}_{t+1}}{1 - \beta^{t+1}}$$

$$\tilde{\mathbf{v}}_{t+1} \leftarrow \frac{\mathbf{v}_{t+1}}{1 - \gamma^{t+1}}$$

- Actualizar los parámetros

$$\phi_{t+1} \leftarrow \phi_t - \alpha \cdot \frac{\tilde{\mathbf{m}}_{t+1}}{\sqrt{\tilde{\mathbf{v}}_{t+1}} + \epsilon}$$

# Estimación adaptativa del momento (Adam)



# Hiperparámetros

- Elección del algoritmo de aprendizaje
- Tasa de aprendizaje
- Momentum