



Tópicos en Inteligencia Artificial Deep Learning

Algoritmos de optimización

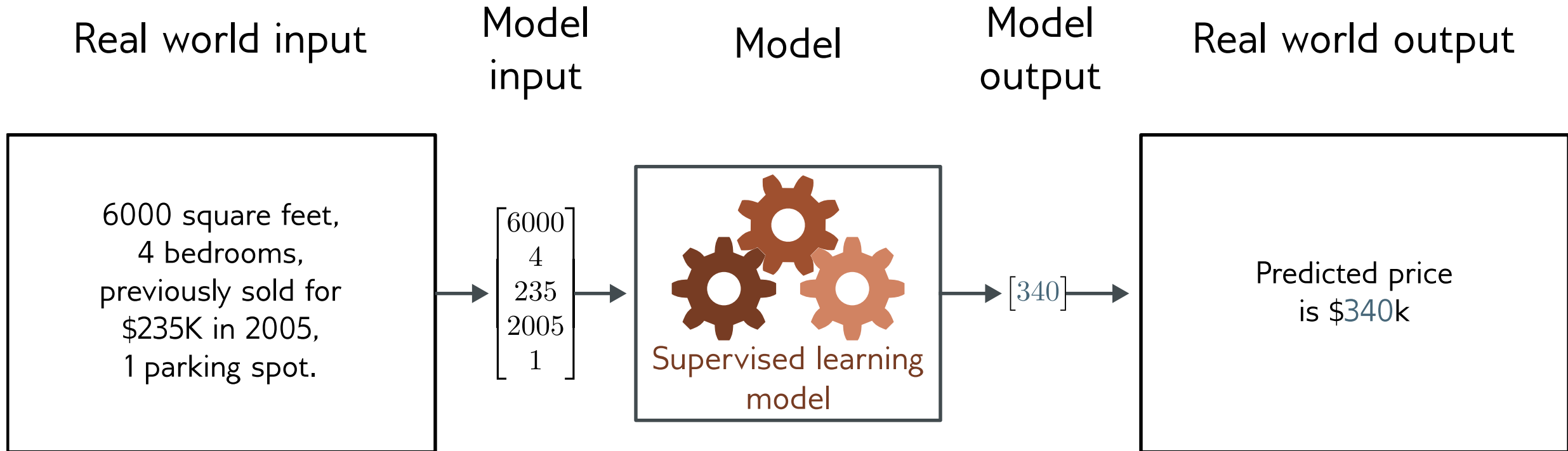
Basado en el ppt 06 – Fitting de

Prof. Simon Prince

Adaptado por Prof. Fernando Crema García

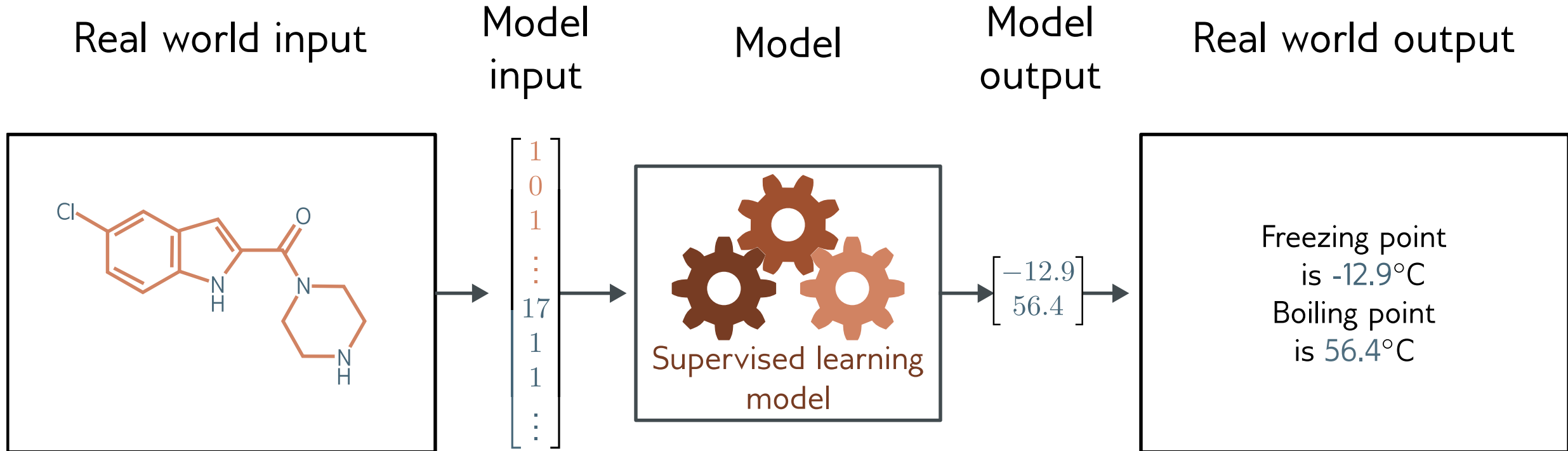


Regresión



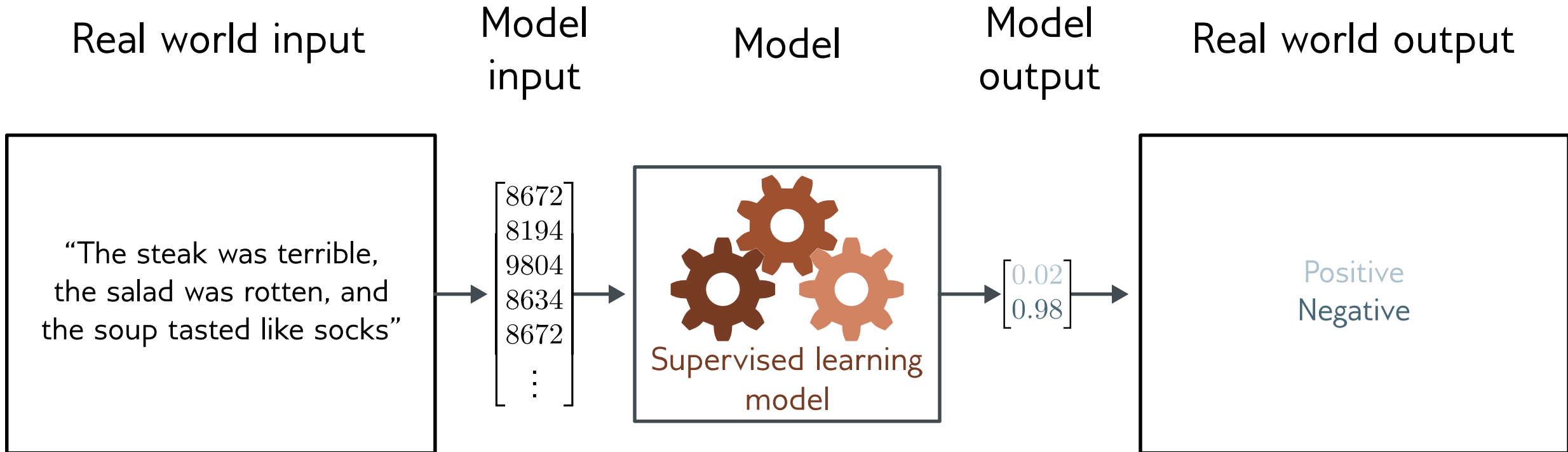
- Problema de regresión univariante (un resultado, valor real)

Regresión gráfica



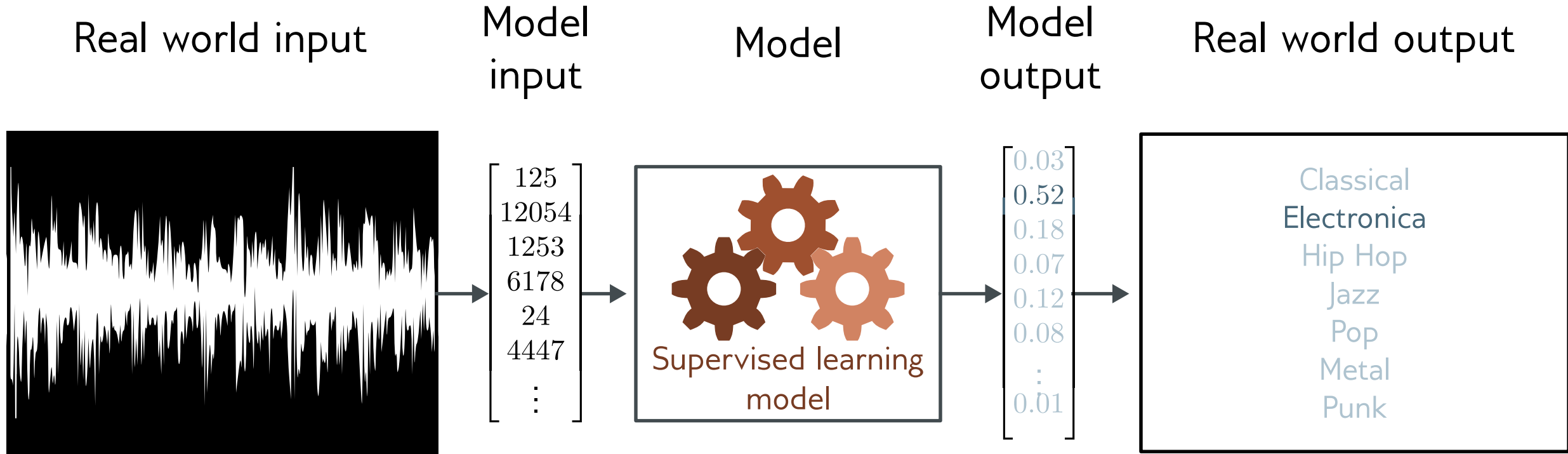
- Problema de regresión multivariante (>1 salida, valor real)

Clasificación de textos



- Problema de clasificación binaria (dos clases discretas)

Clasificación de los géneros musicales



- Problema de clasificación multiclase (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:

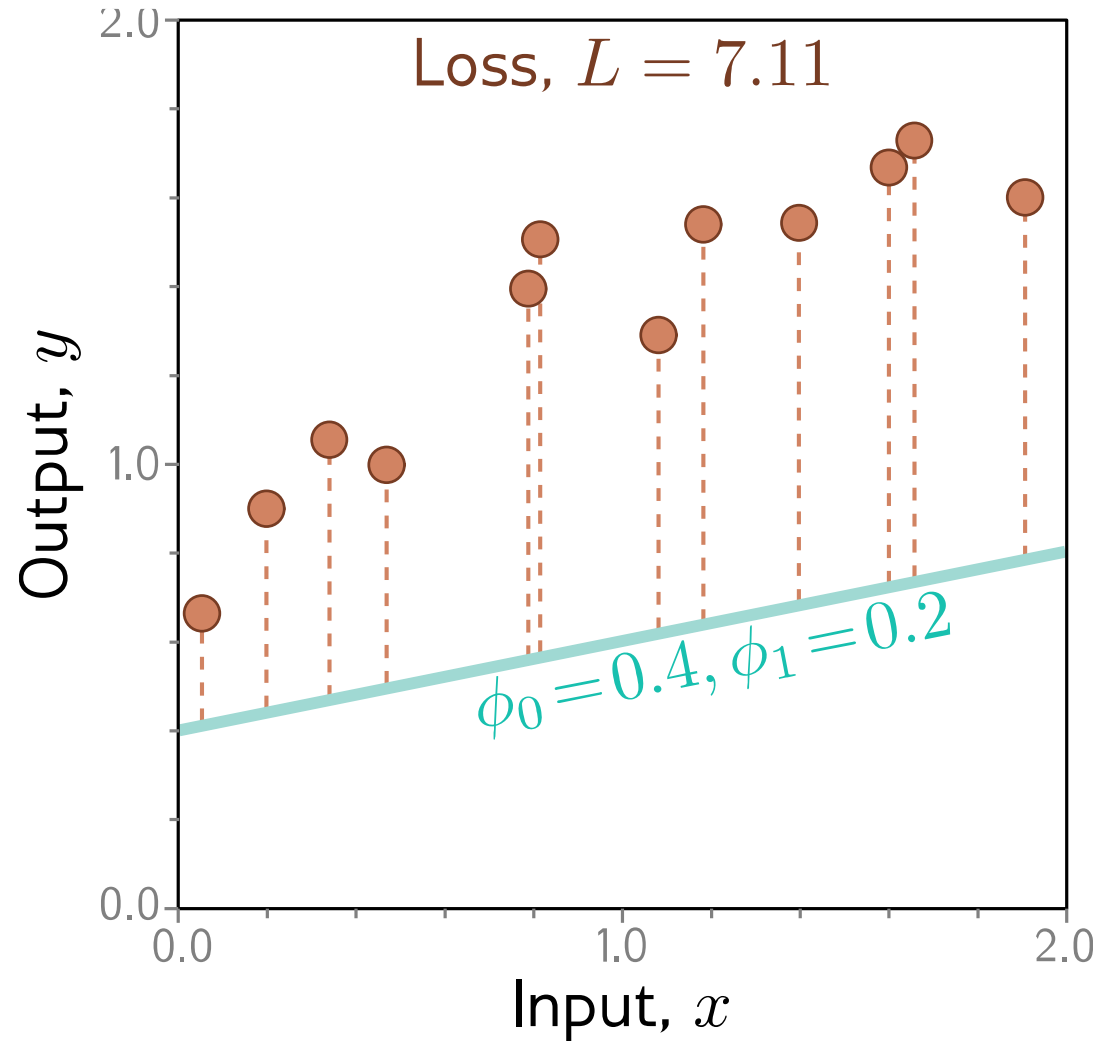
$$L[\phi]$$

← 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



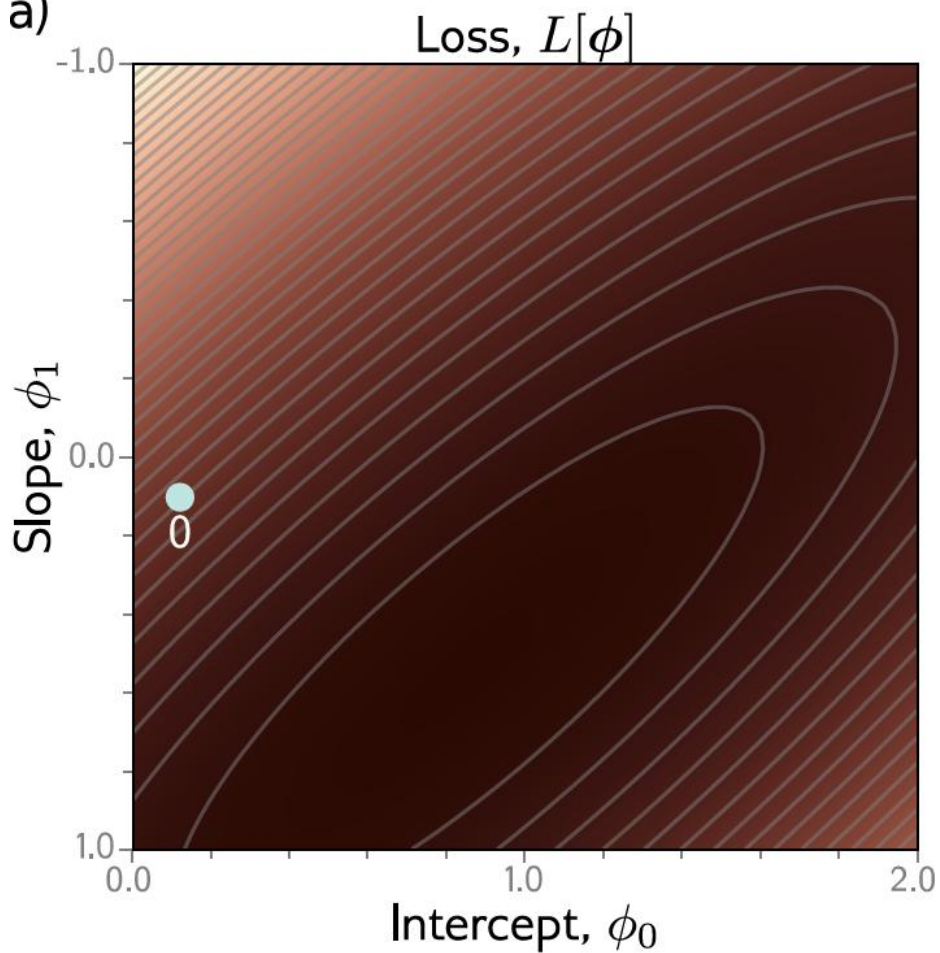
Función de pérdida:

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

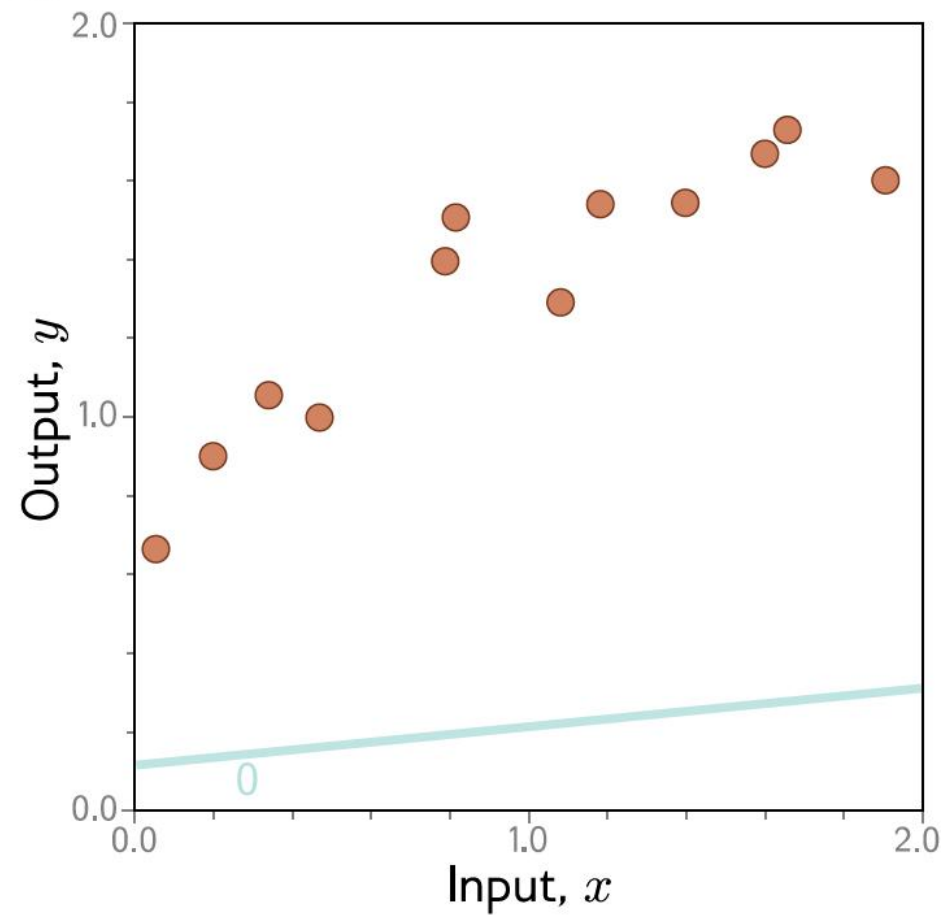
"Función de pérdida por mínimos cuadrados"

Ejemplo: entrenamiento de regresión lineal 1D

a)

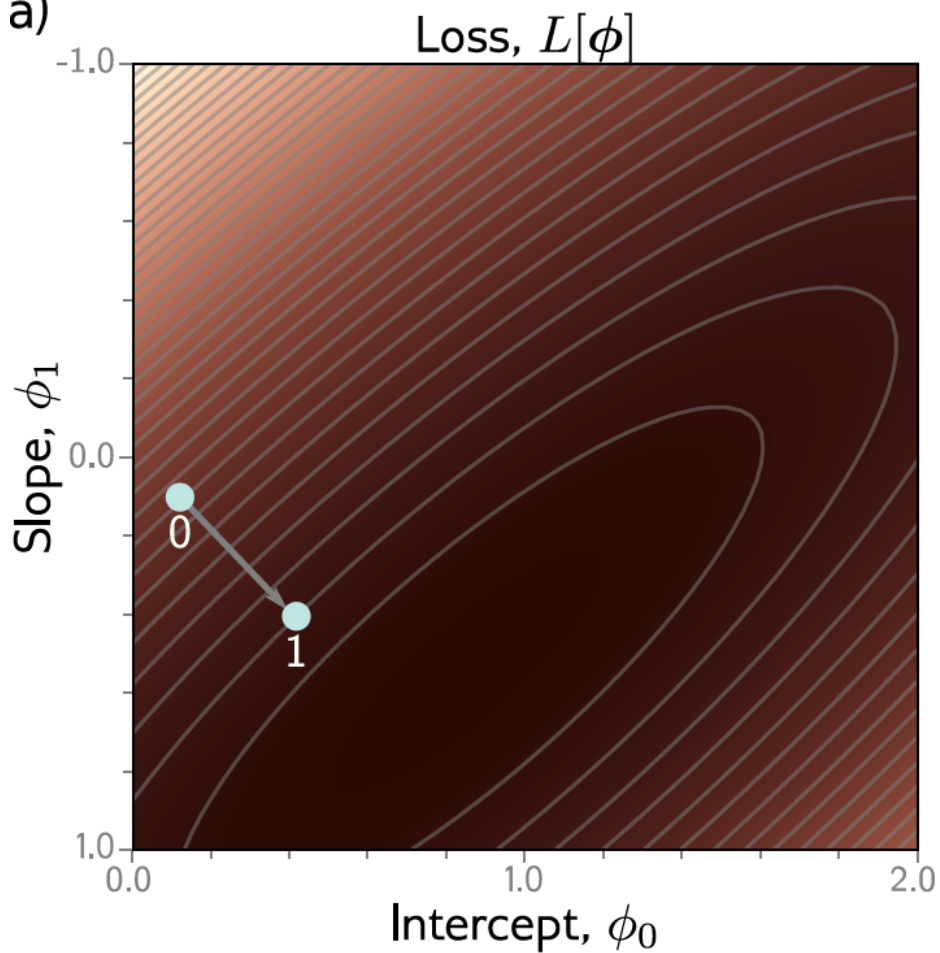


b)

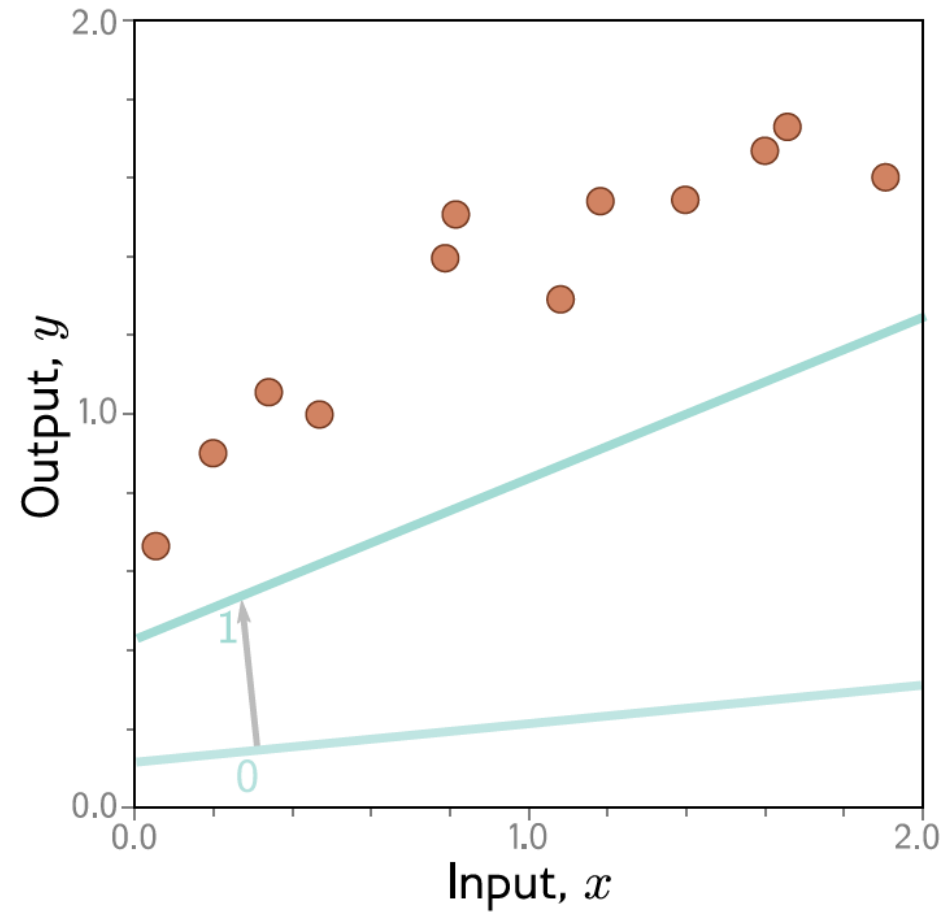


Ejemplo: entrenamiento de regresión lineal 1D

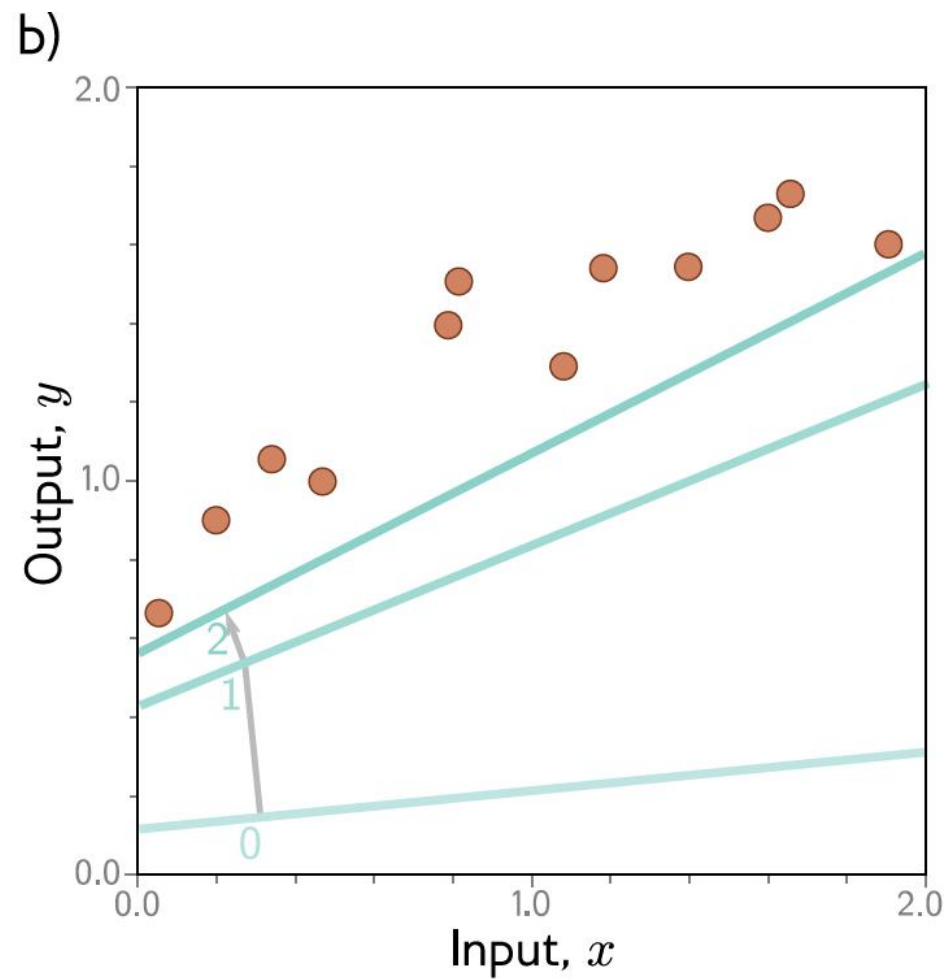
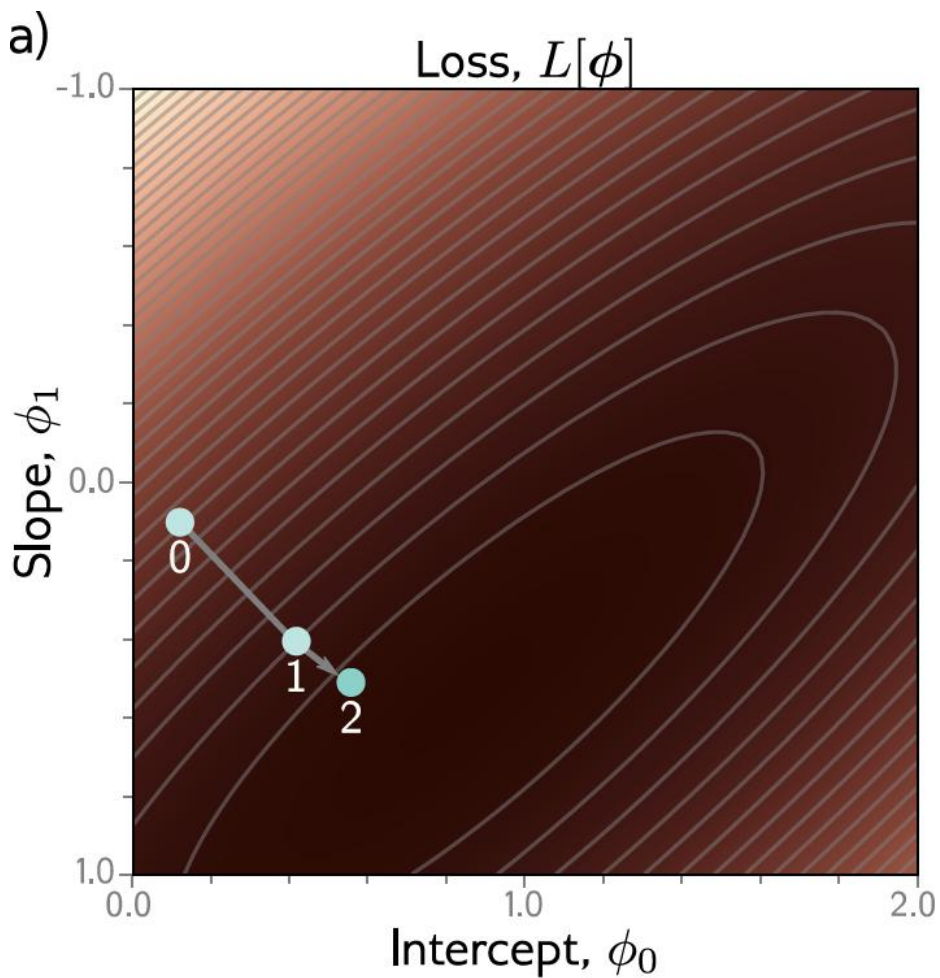
a)



b)

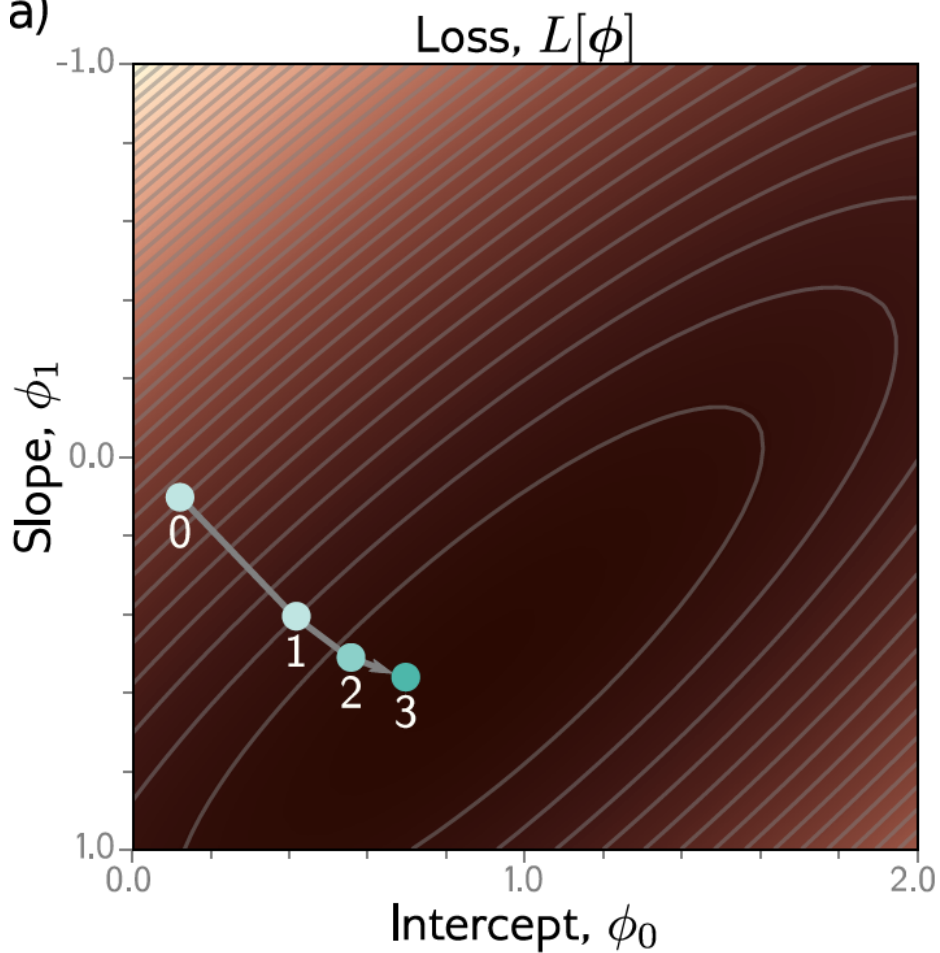


Ejemplo: entrenamiento de regresión lineal 1D

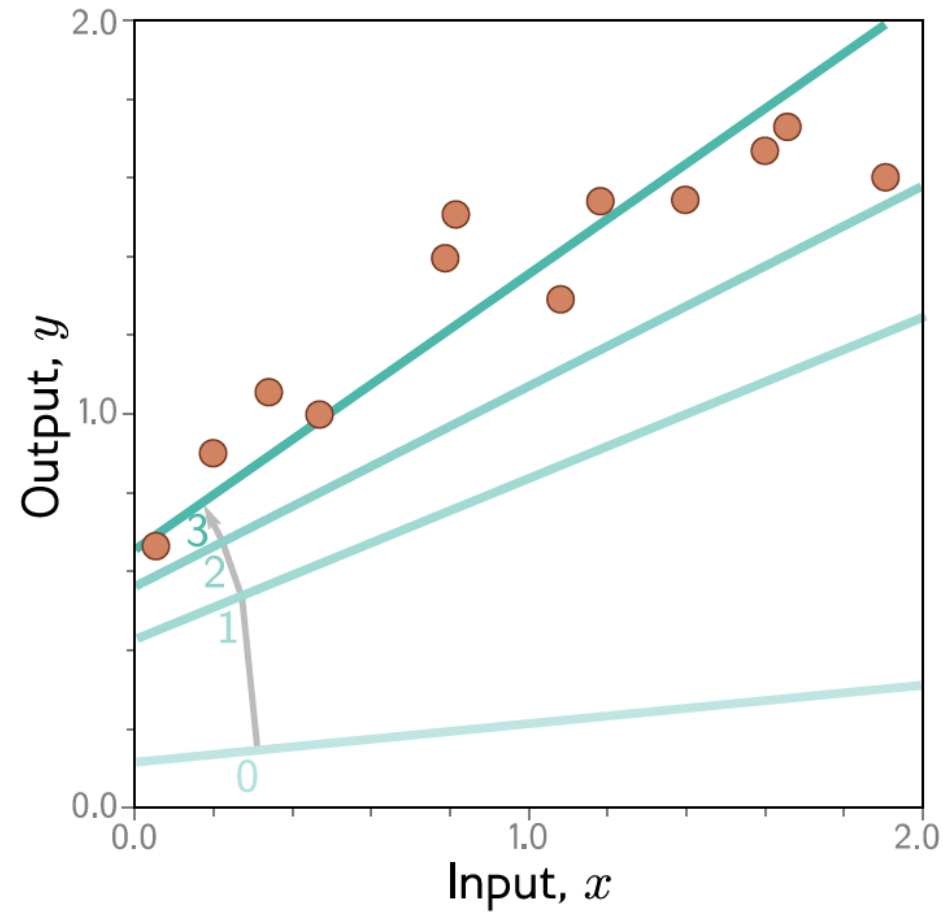


Ejemplo: entrenamiento de regresión lineal 1D

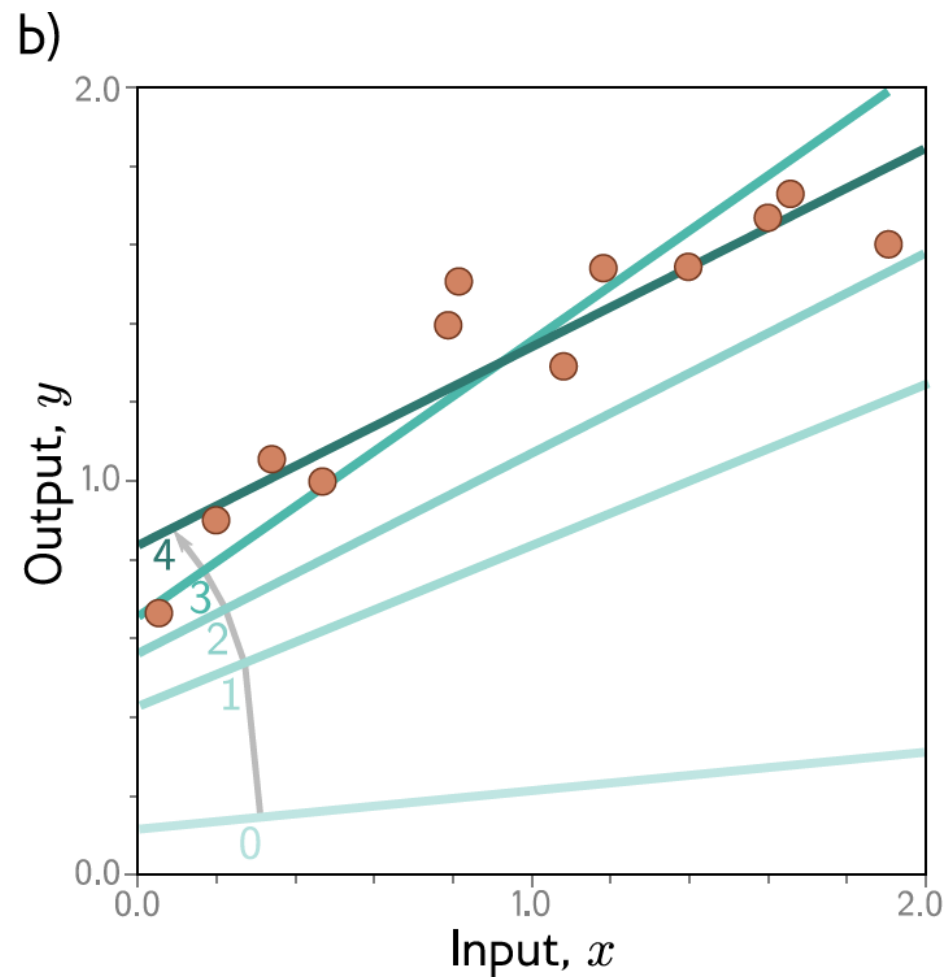
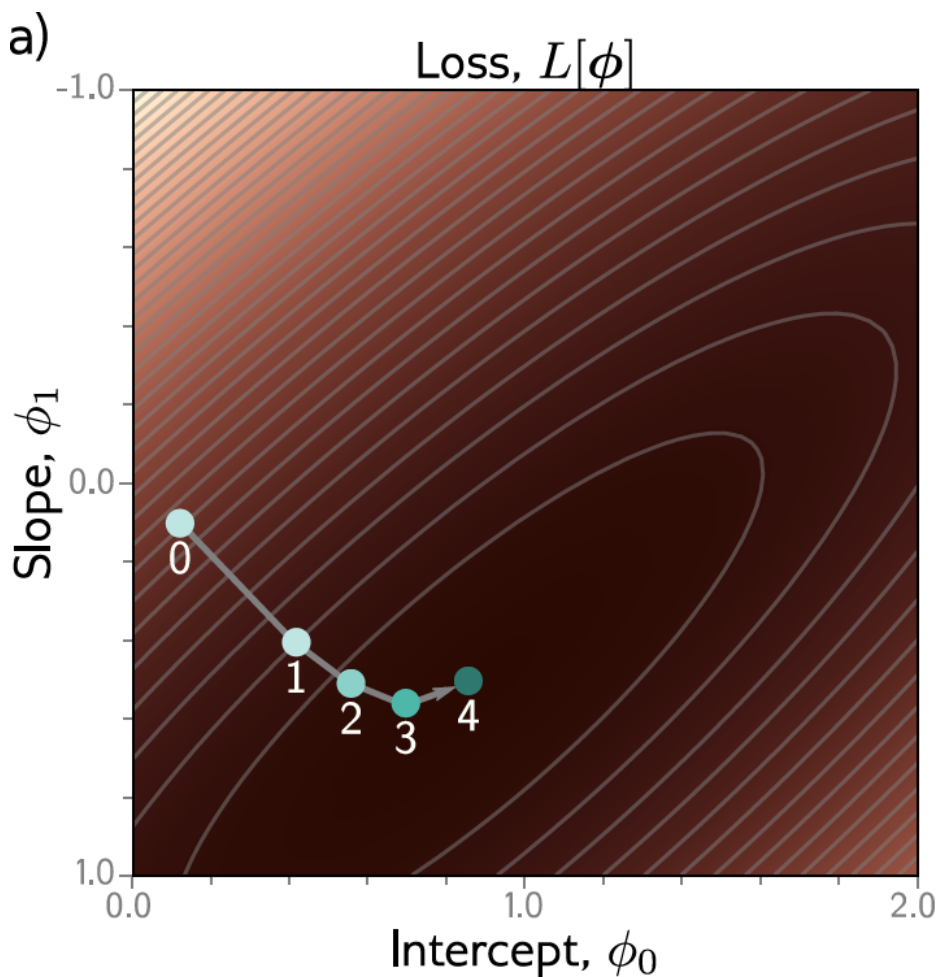
a)



b)



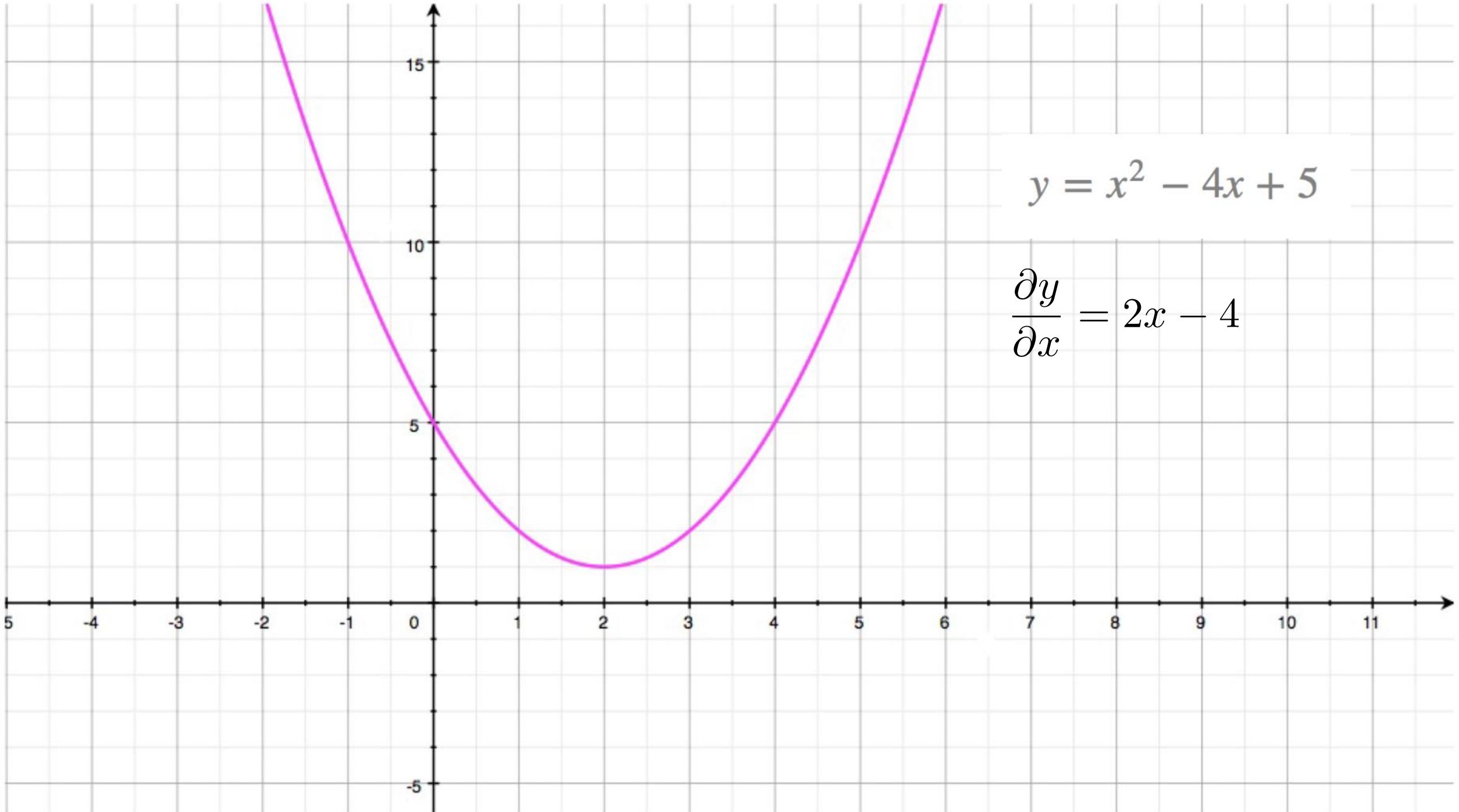
Ejemplo: entrenamiento de regresión lineal 1D



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

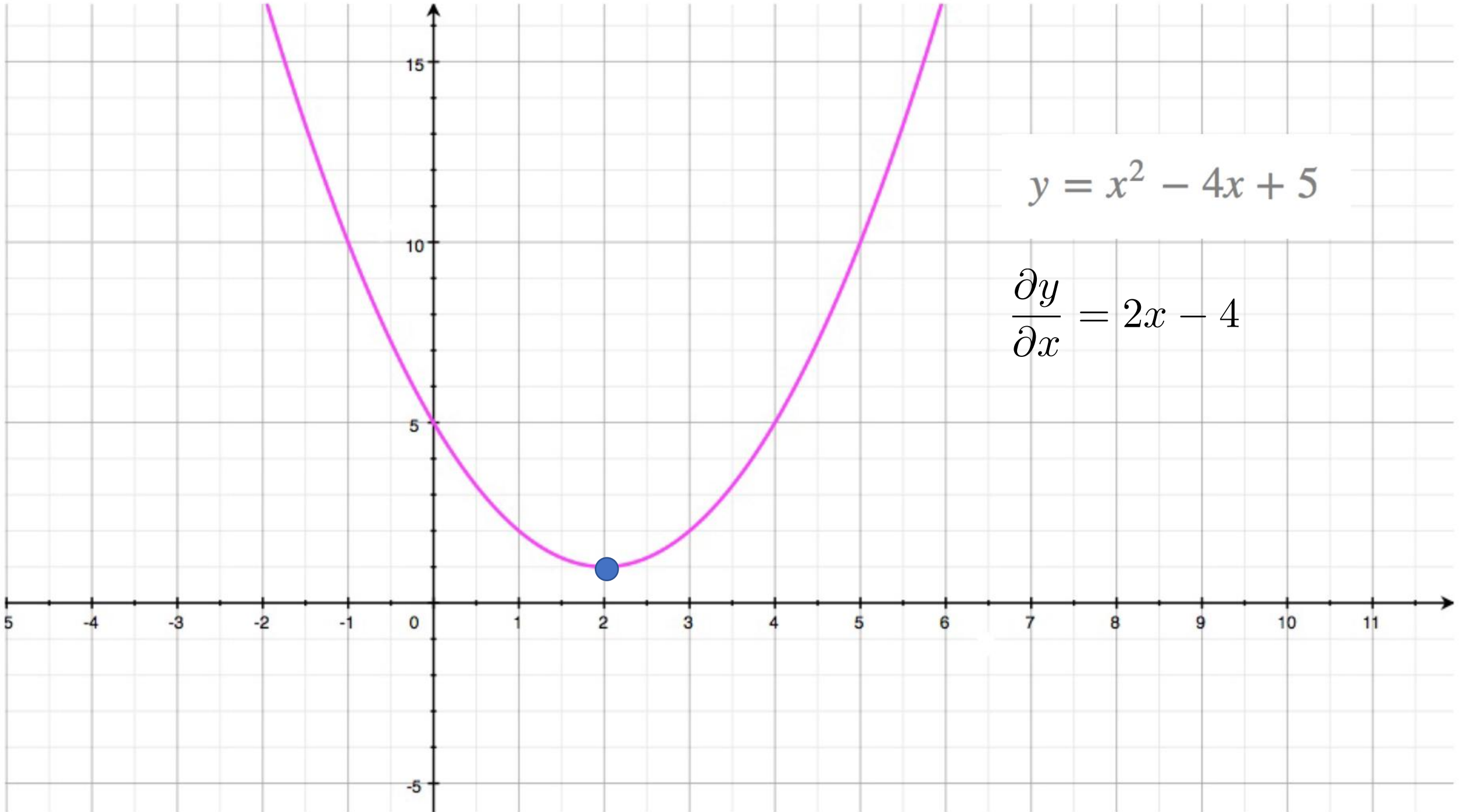
Modelos de ajuste

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



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

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

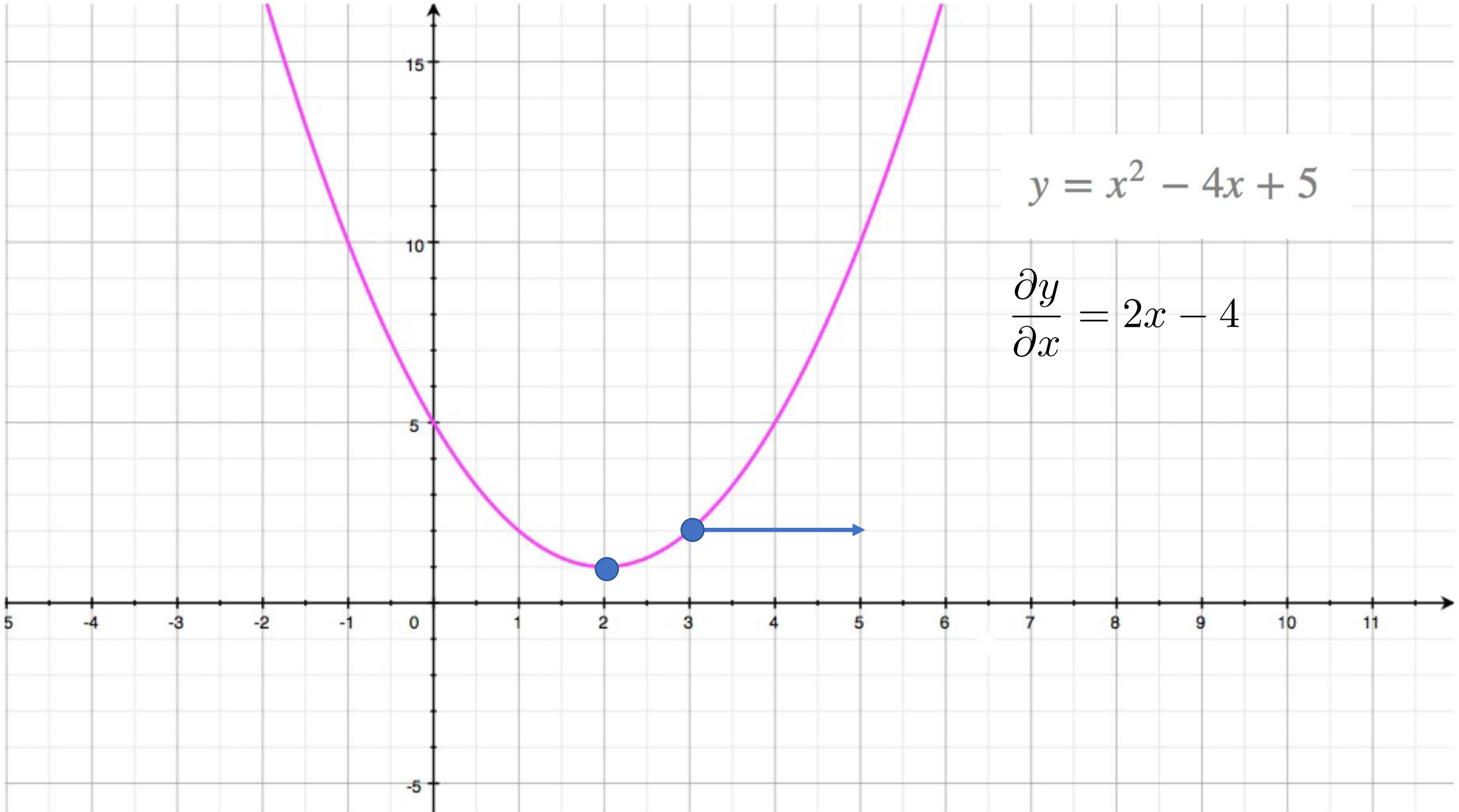


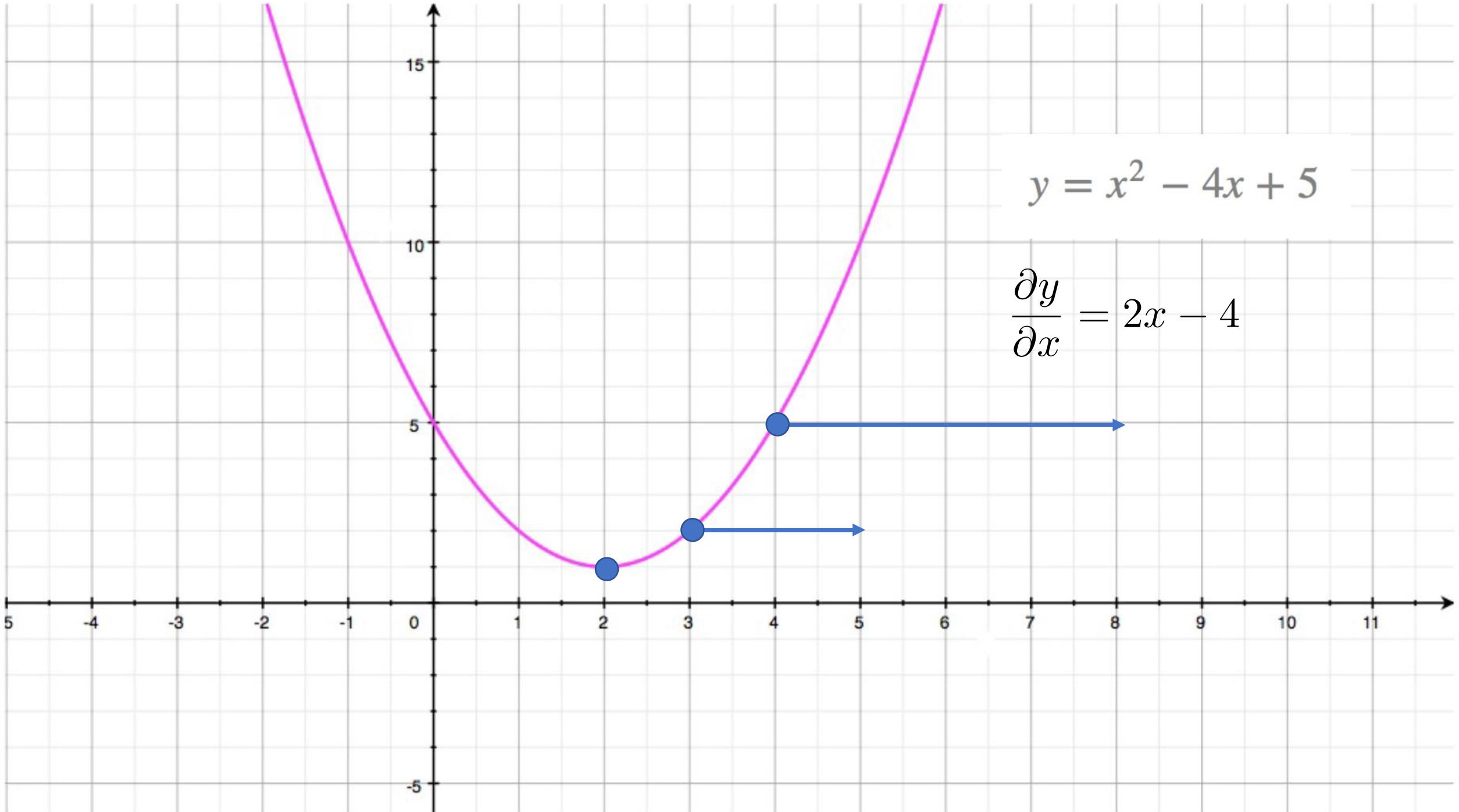
$$y = x^2 - 4x + 5$$

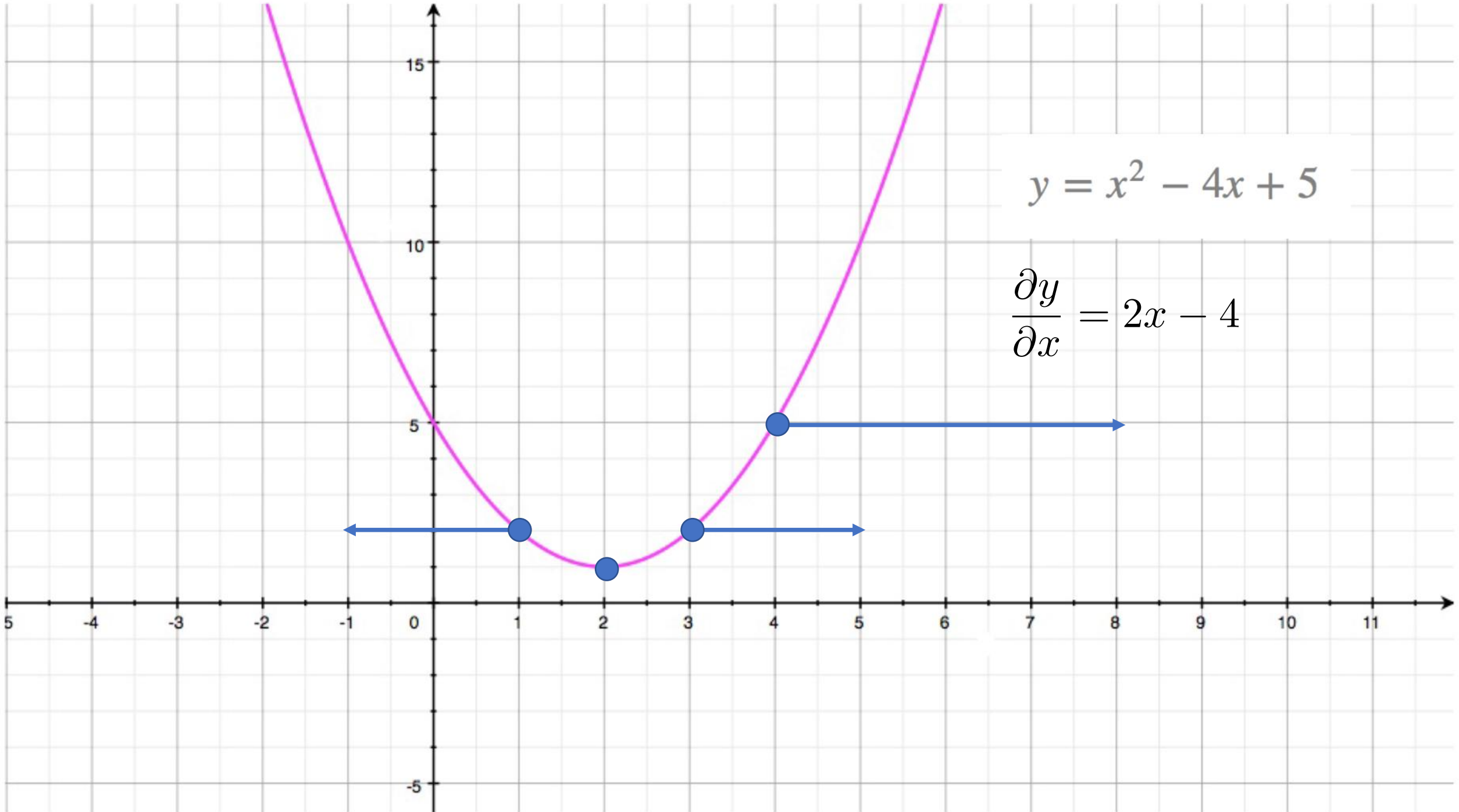
$$\frac{\partial y}{\partial x} = 2x - 4$$

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

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





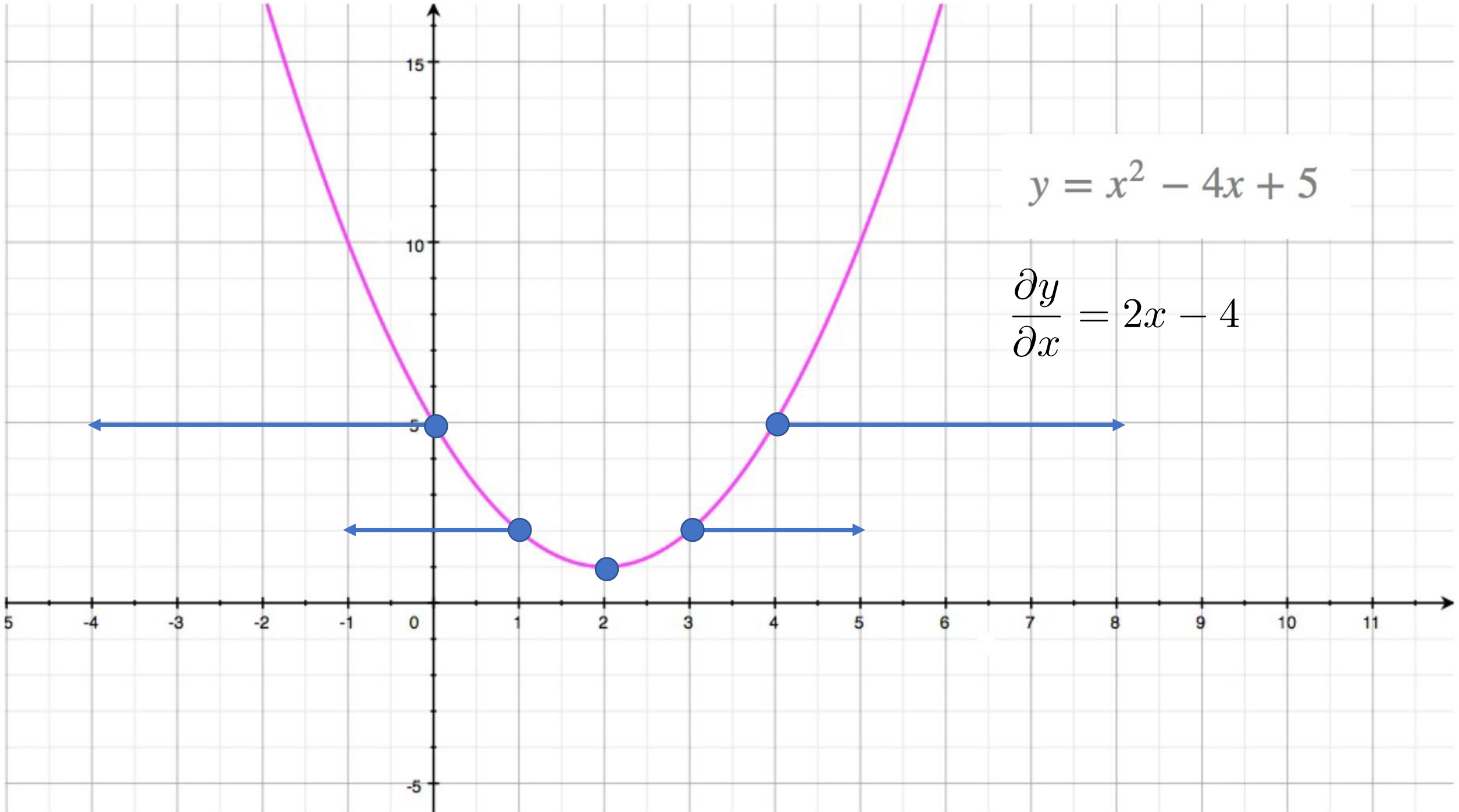


$$y = x^2 - 4x + 5$$

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

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

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



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:

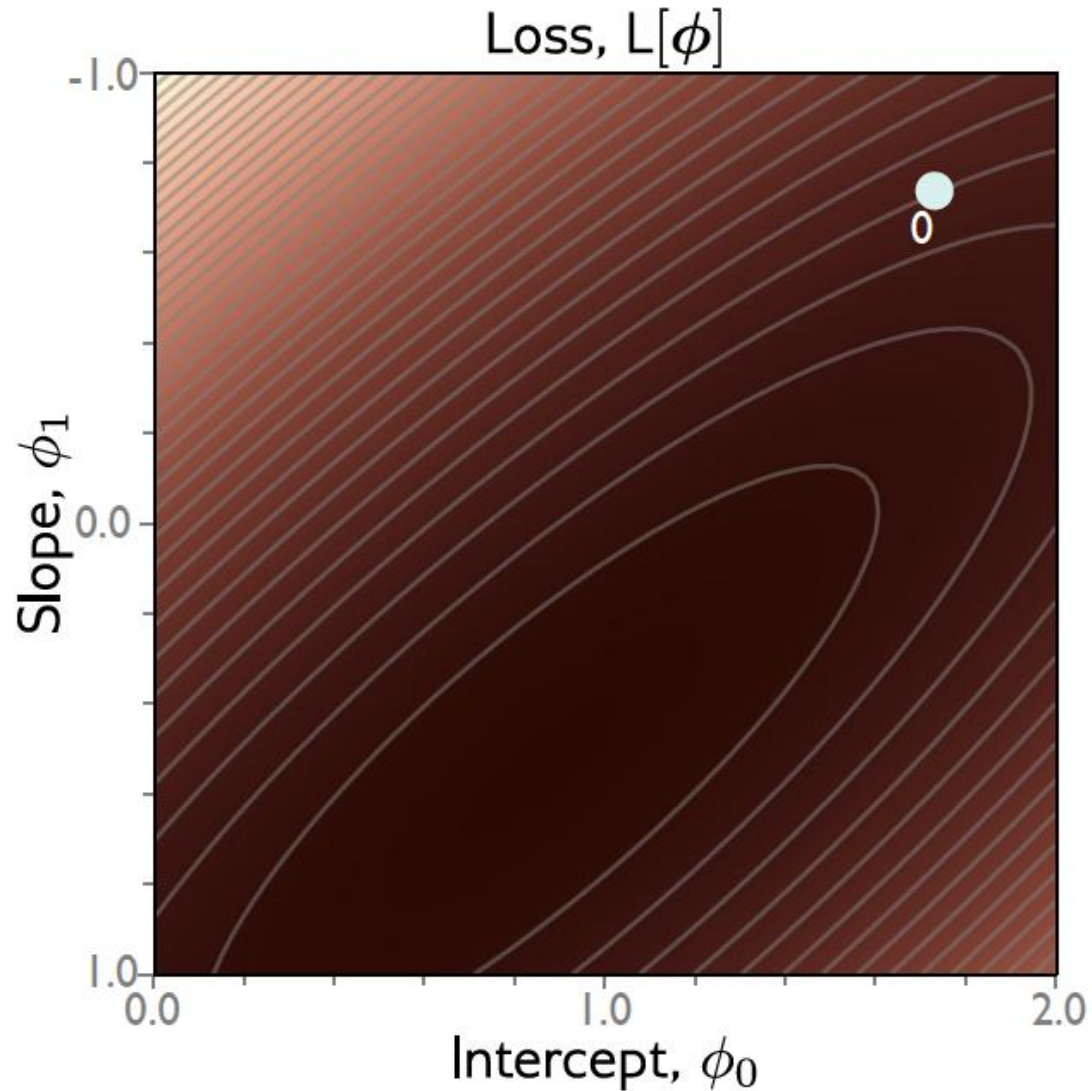
$$\phi \longleftarrow \phi - \alpha \frac{\partial L}{\partial \phi},$$

where the positive scalar α 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

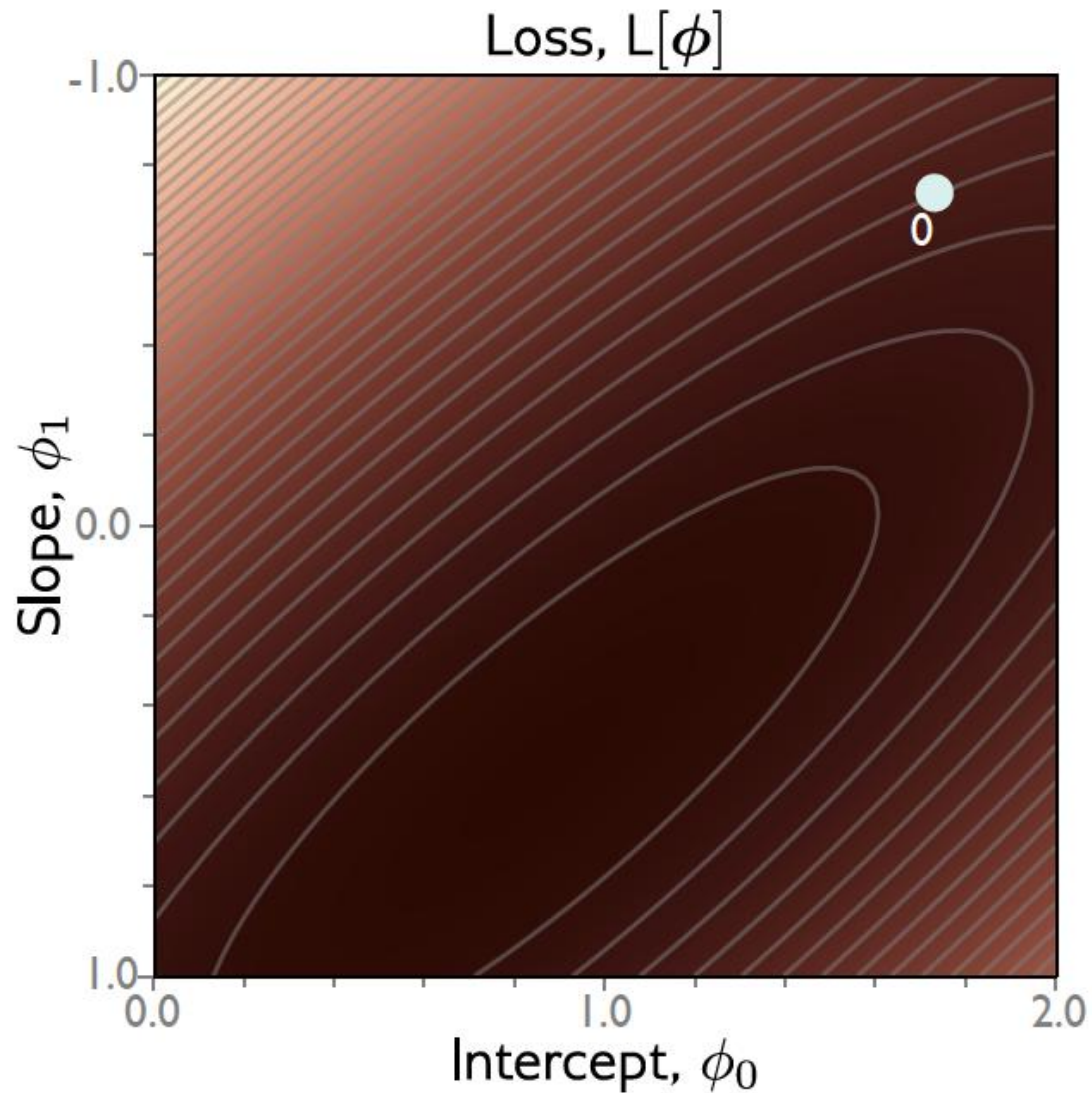
Descenso gradual



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

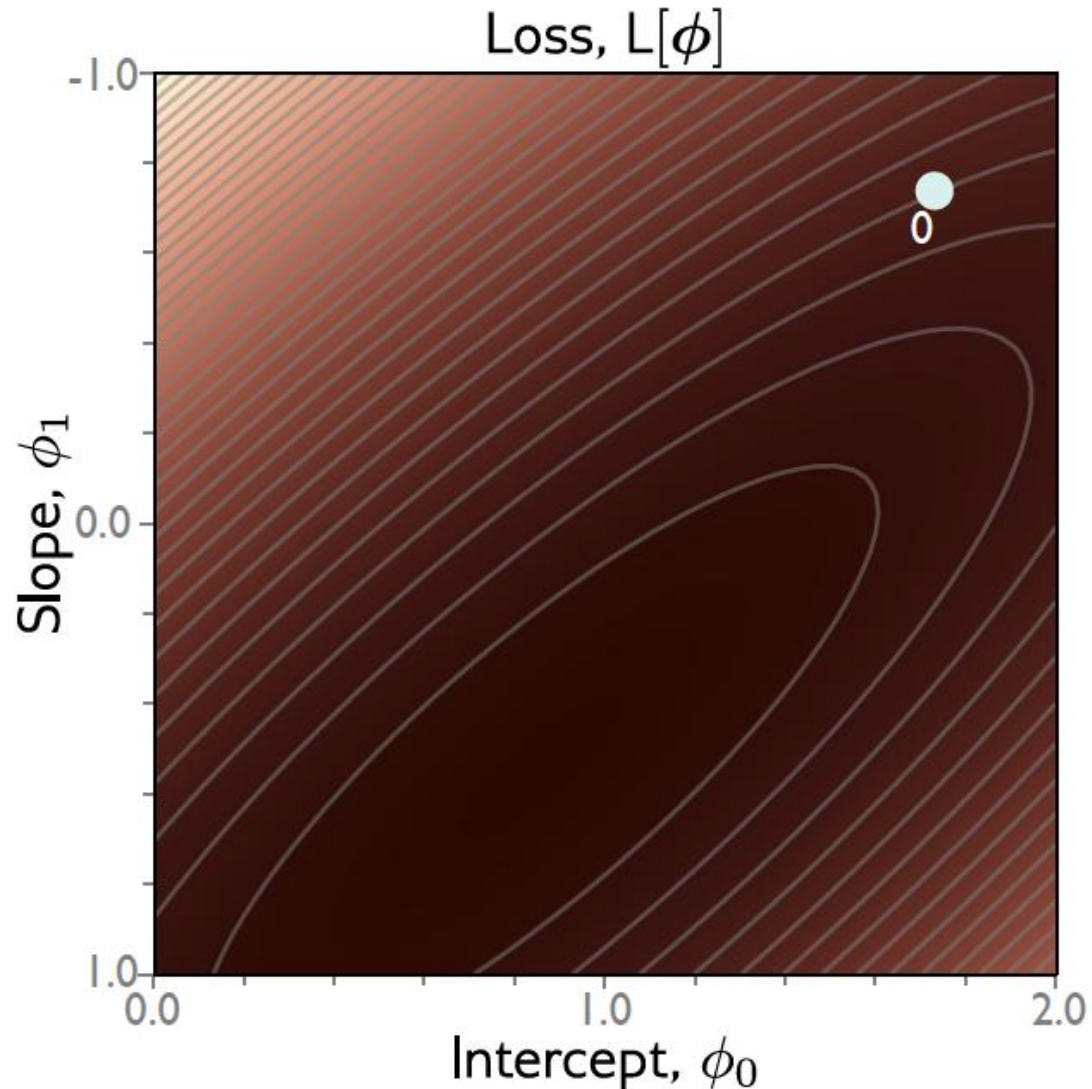


Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

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

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

Descenso gradual



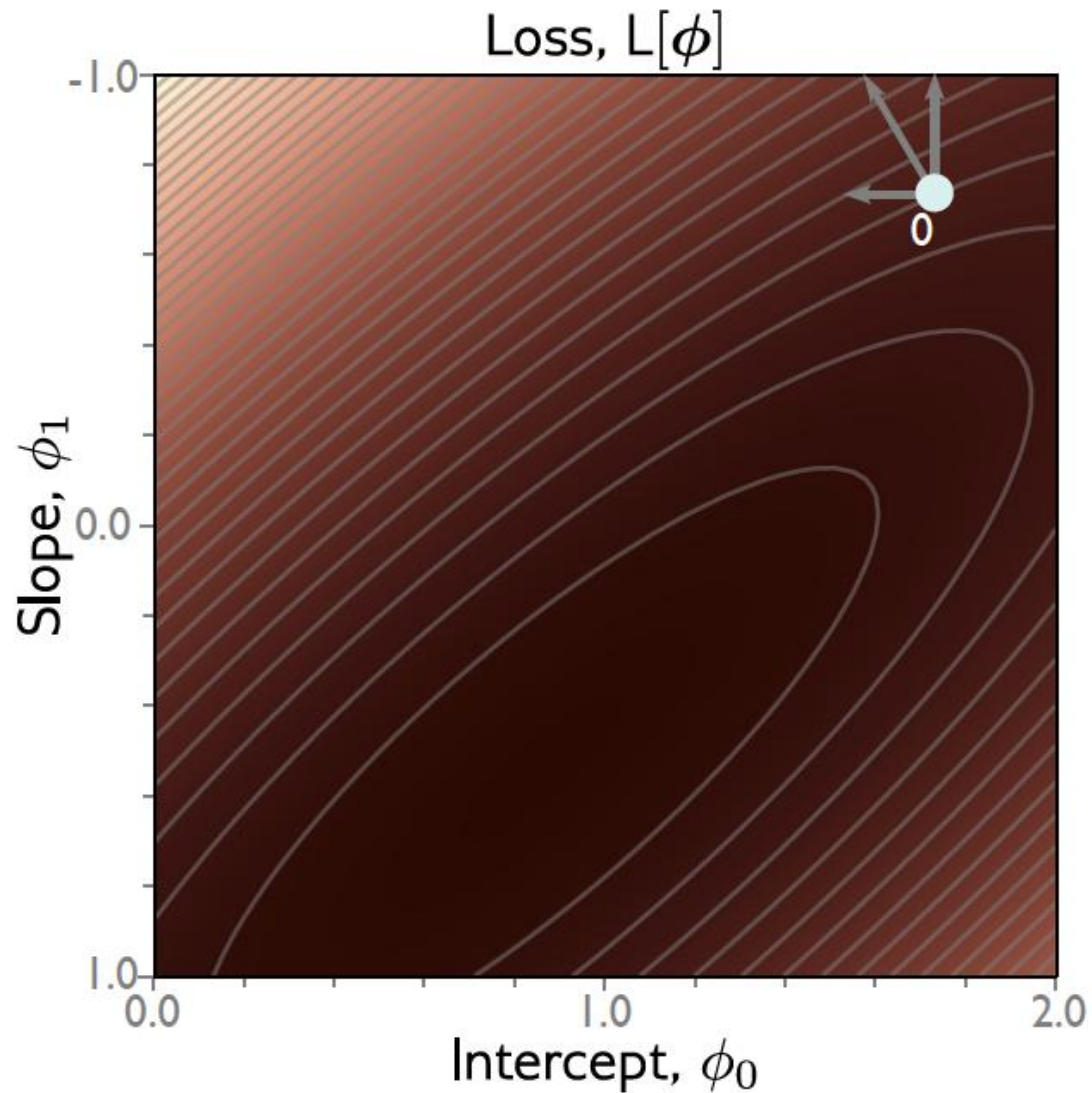
Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

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

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

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

Descenso gradual

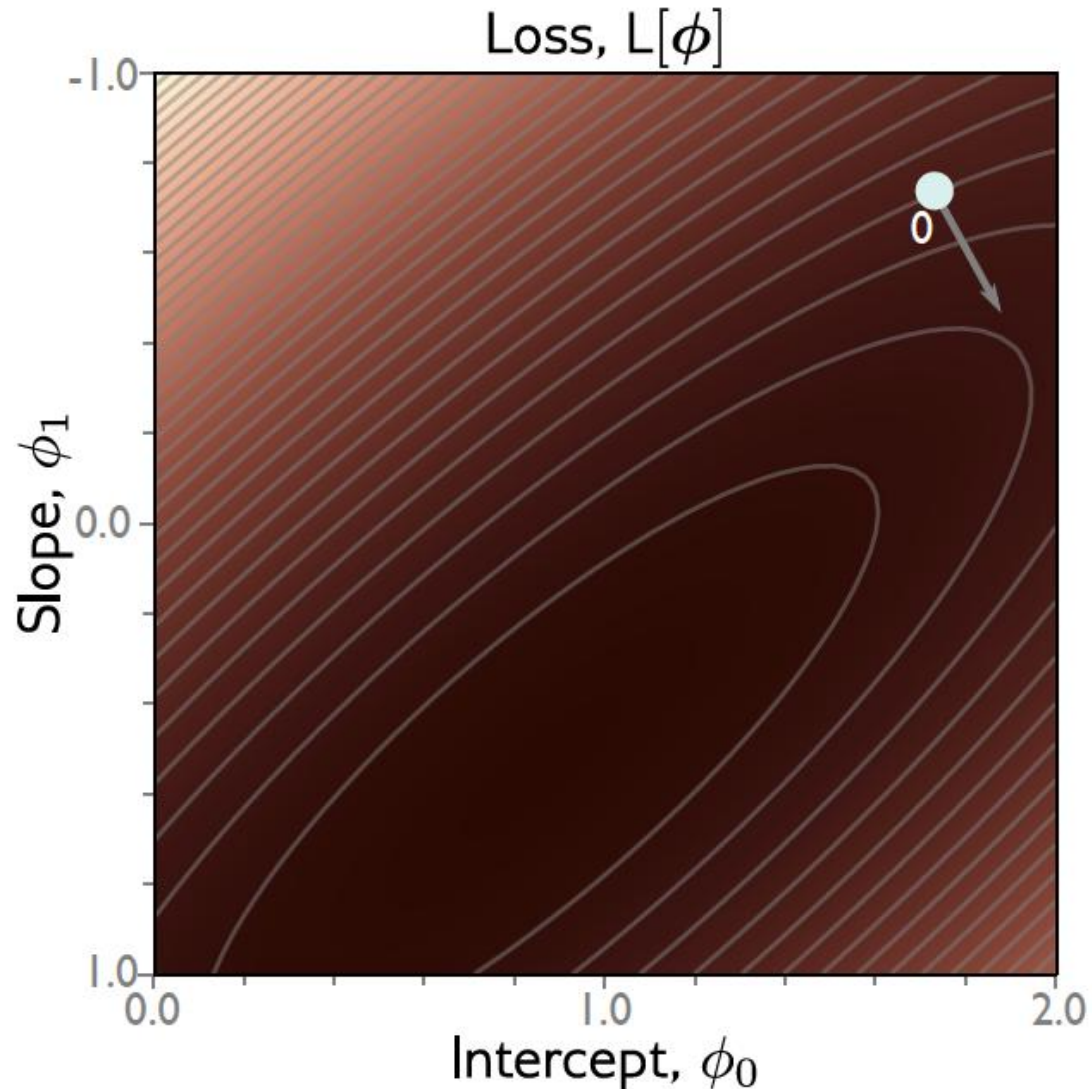


Paso 1: Calcular las derivadas (pendientes de la función) con Respecto a los parámetros

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

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

Descenso gradual



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

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

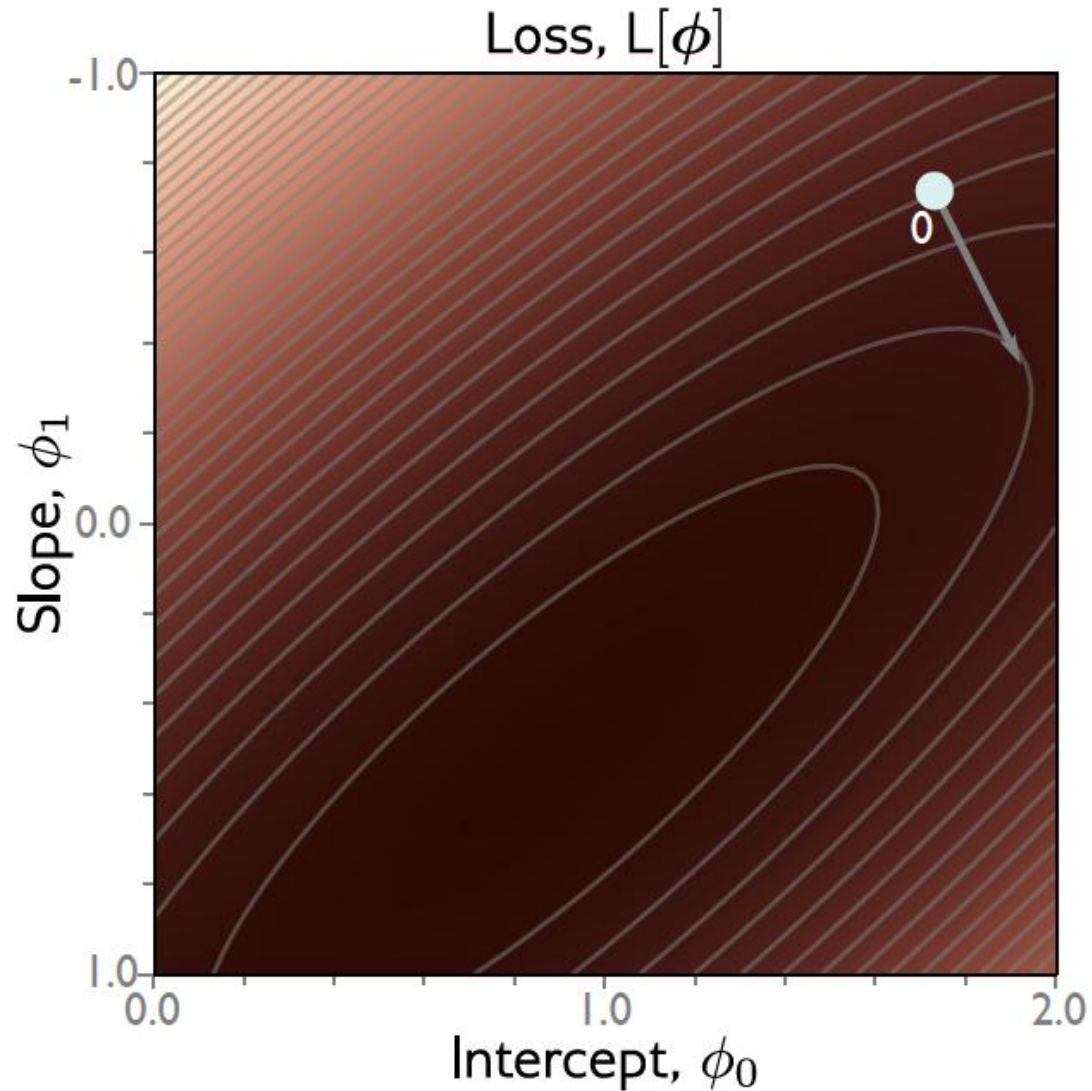
$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

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

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

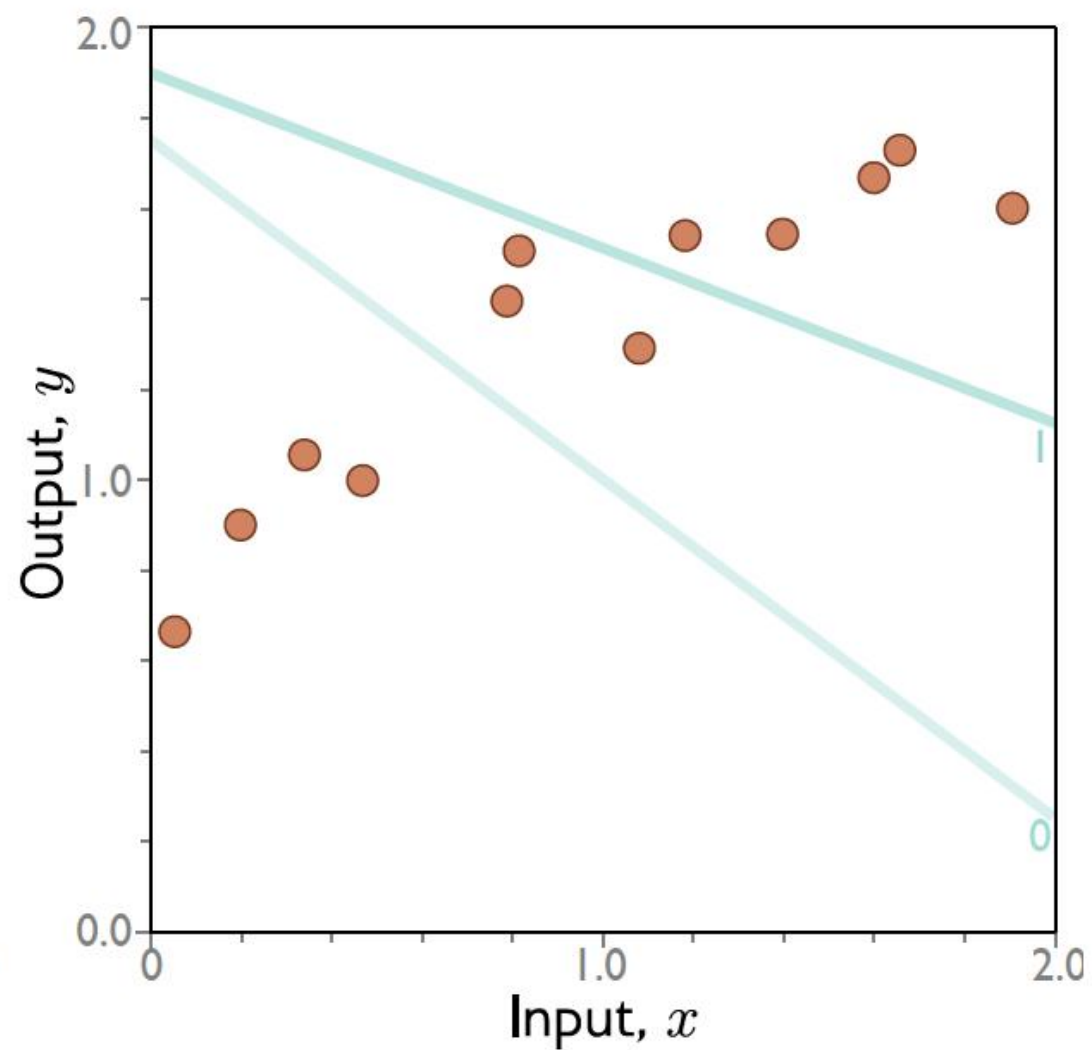
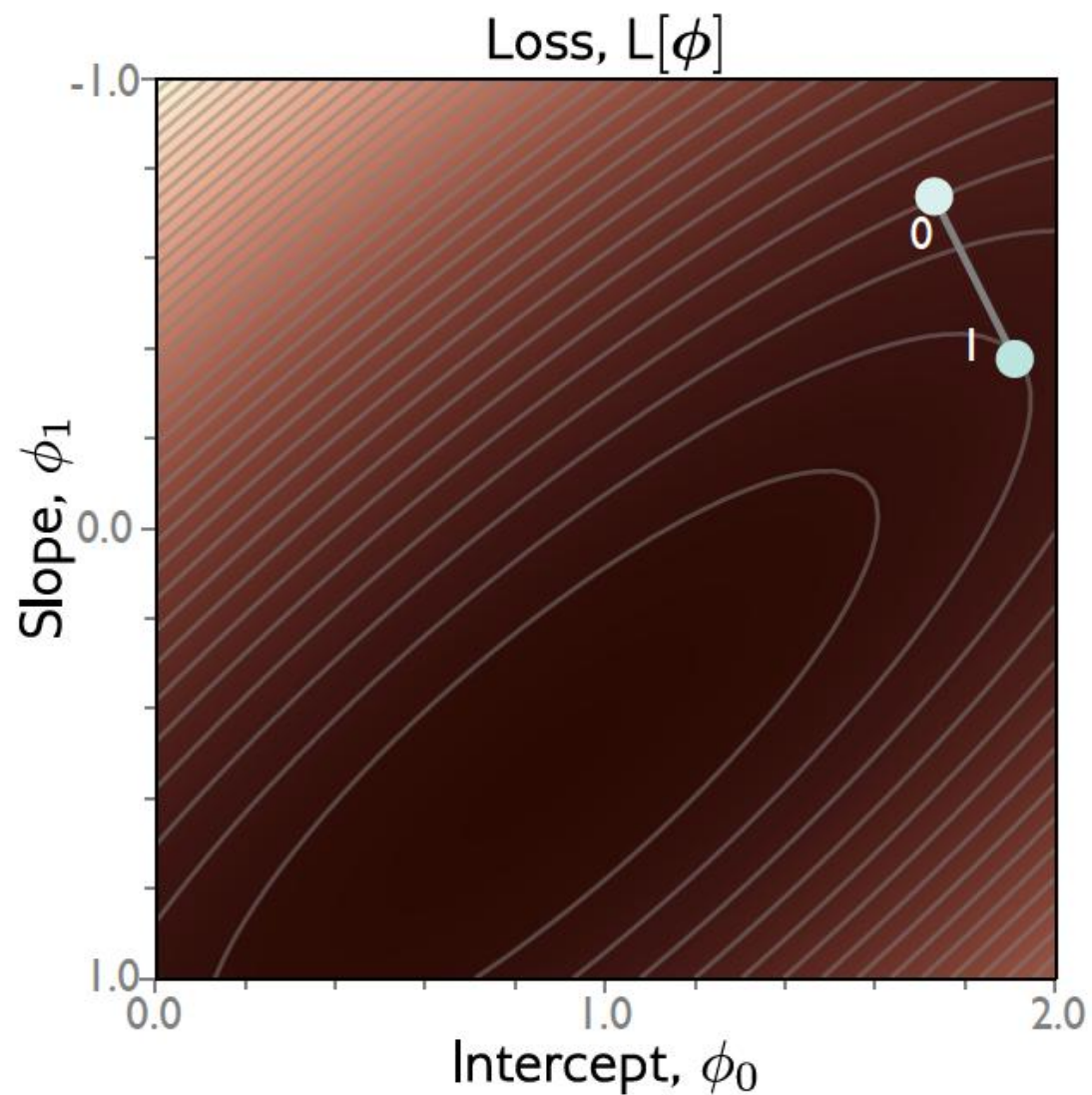
$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

Paso 2: Actualizar los parámetros según la regla

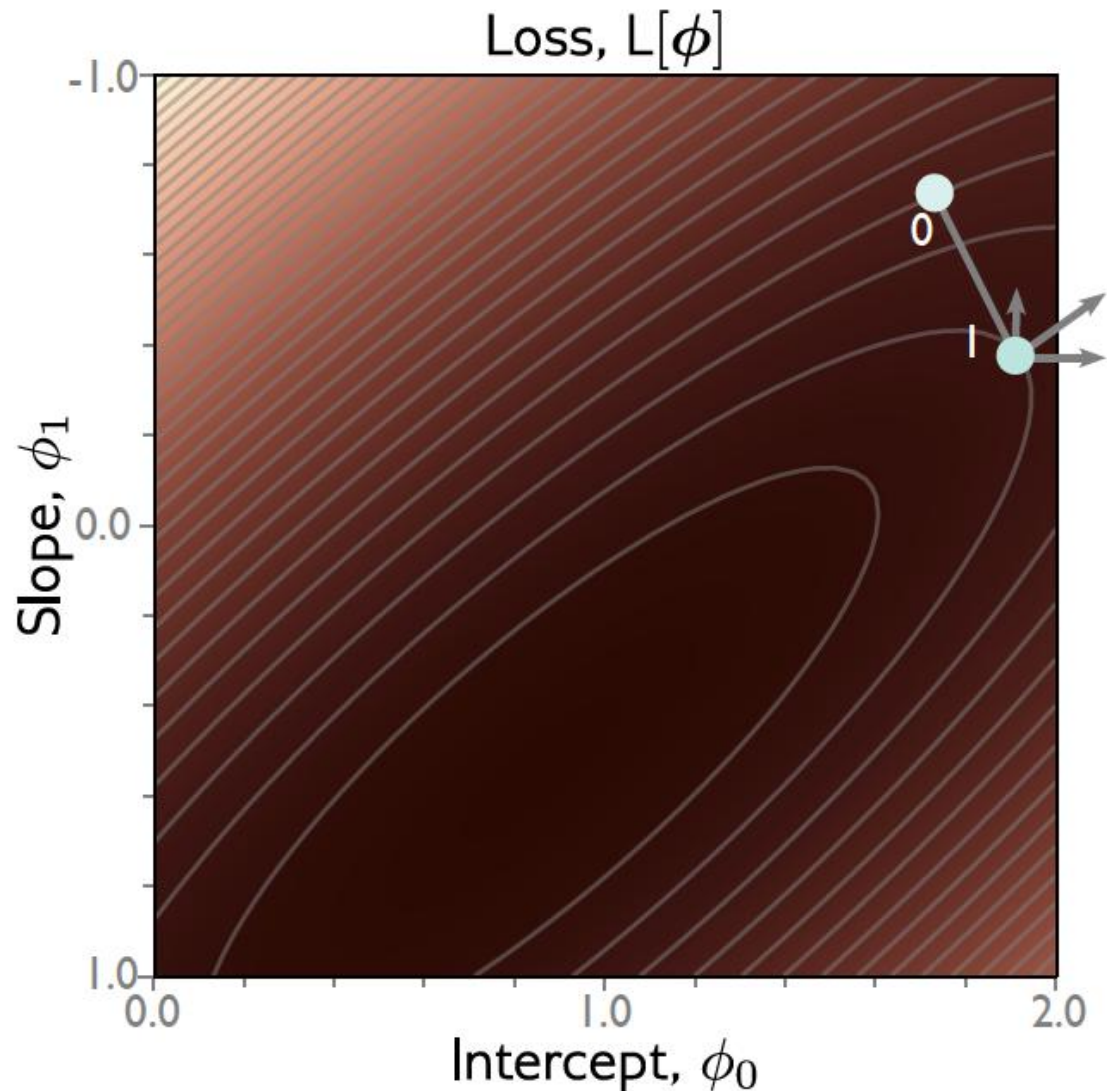
$$\phi \leftarrow \phi - \alpha \frac{\partial L}{\partial \phi}$$

= tamaño del paso

Descenso gradual



Descenso gradual



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

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

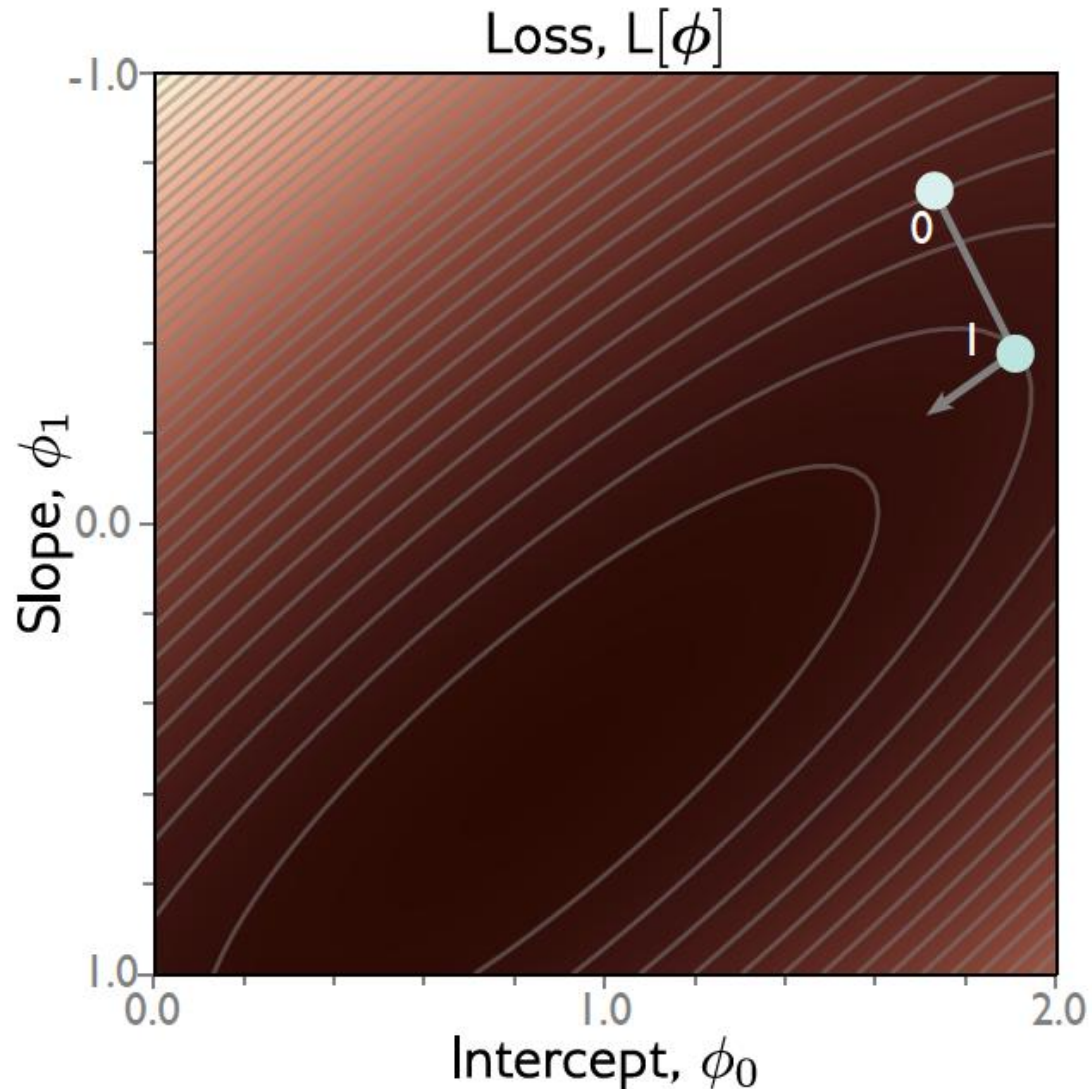
$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

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

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

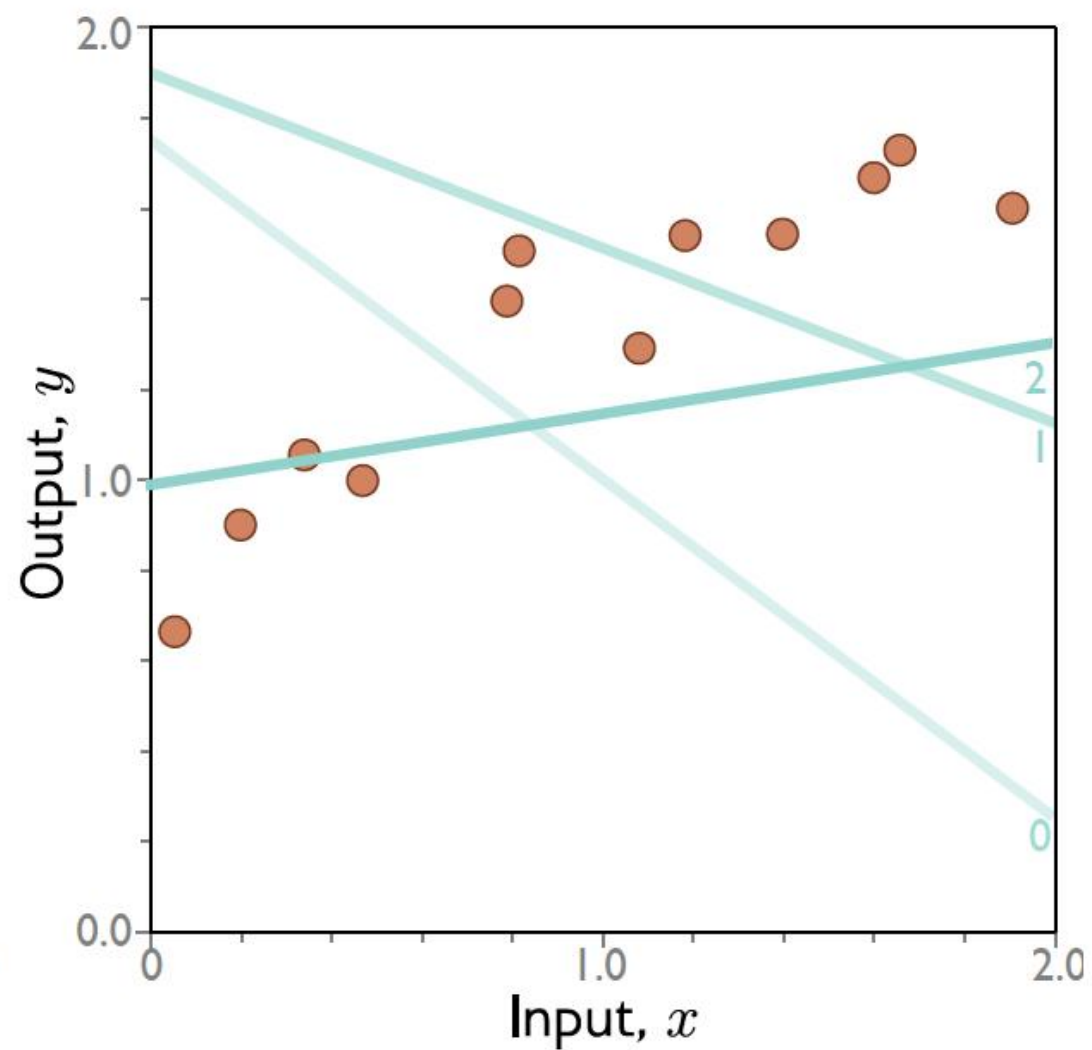
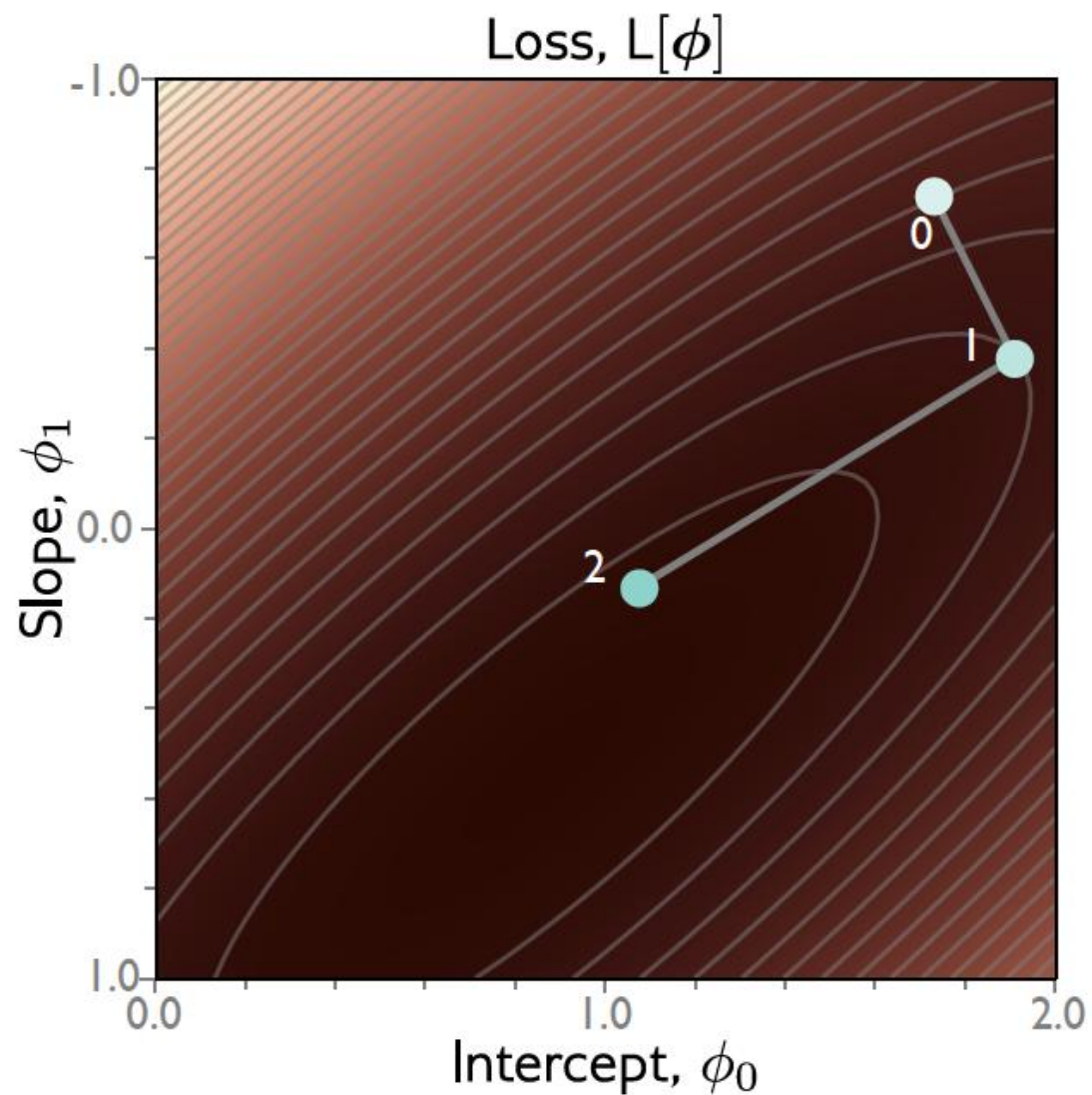
$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

Paso 2: Actualizar los parámetros según la regla

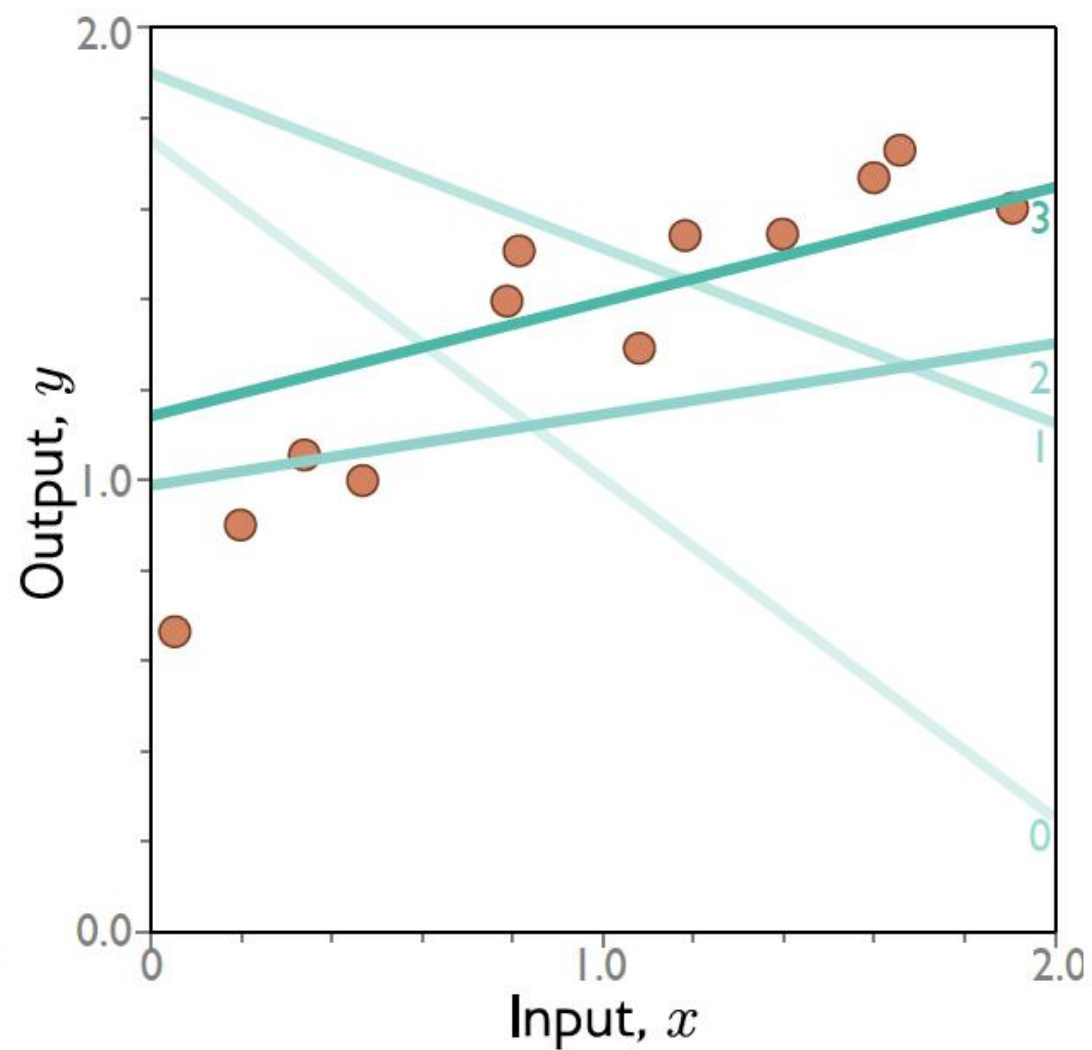
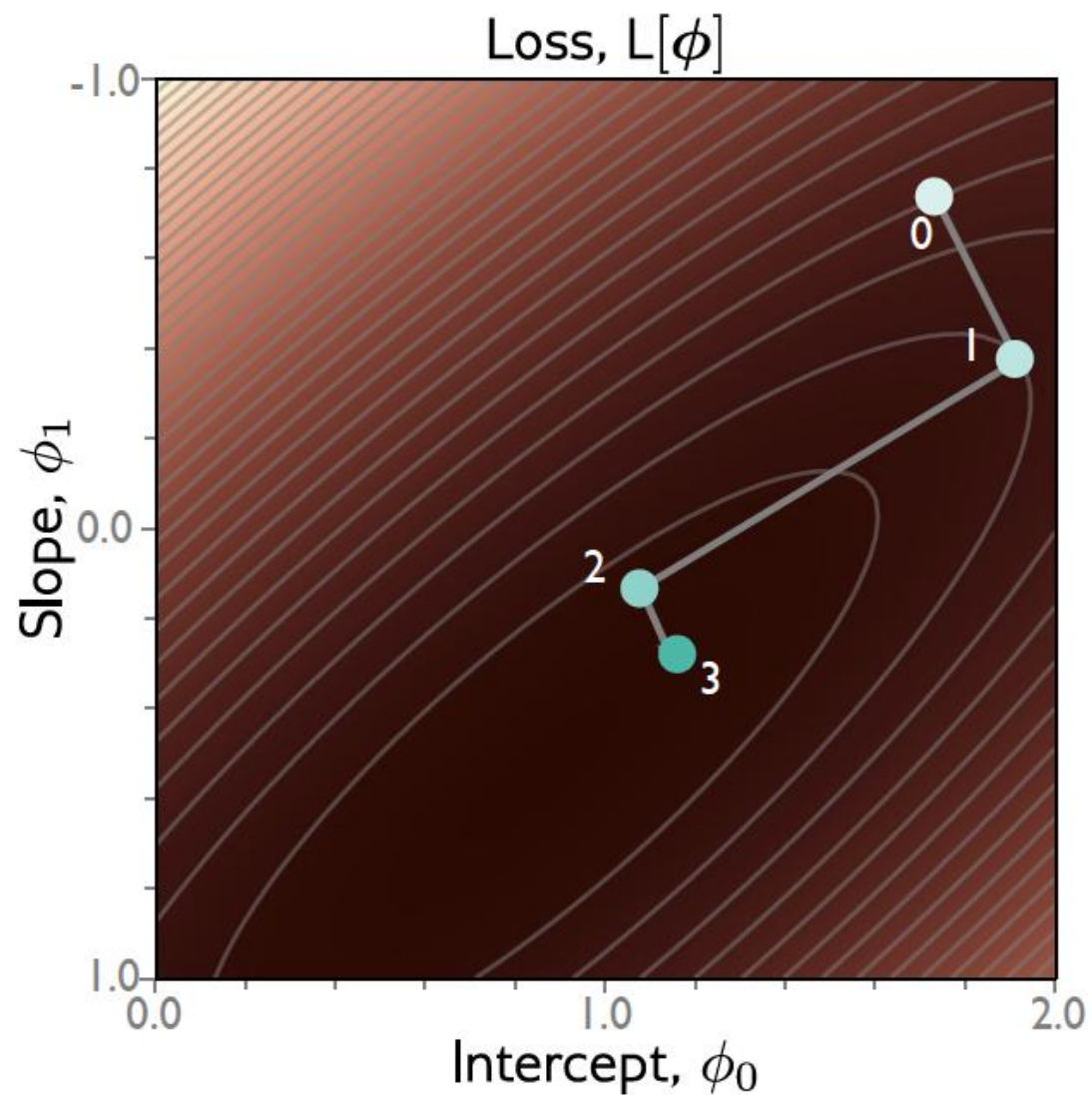
$$\phi \leftarrow \phi - \alpha \frac{\partial L}{\partial \phi}$$

= tamaño del paso

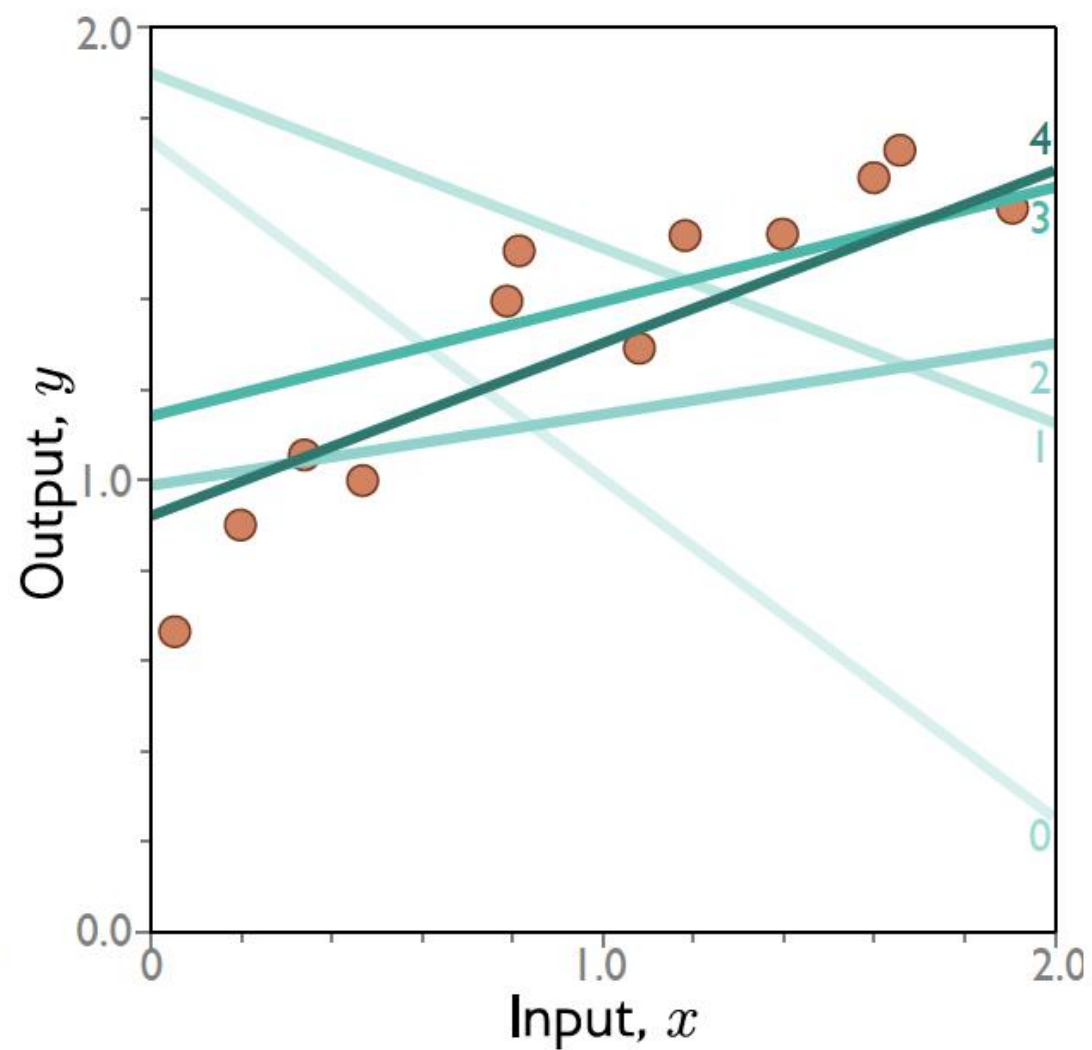
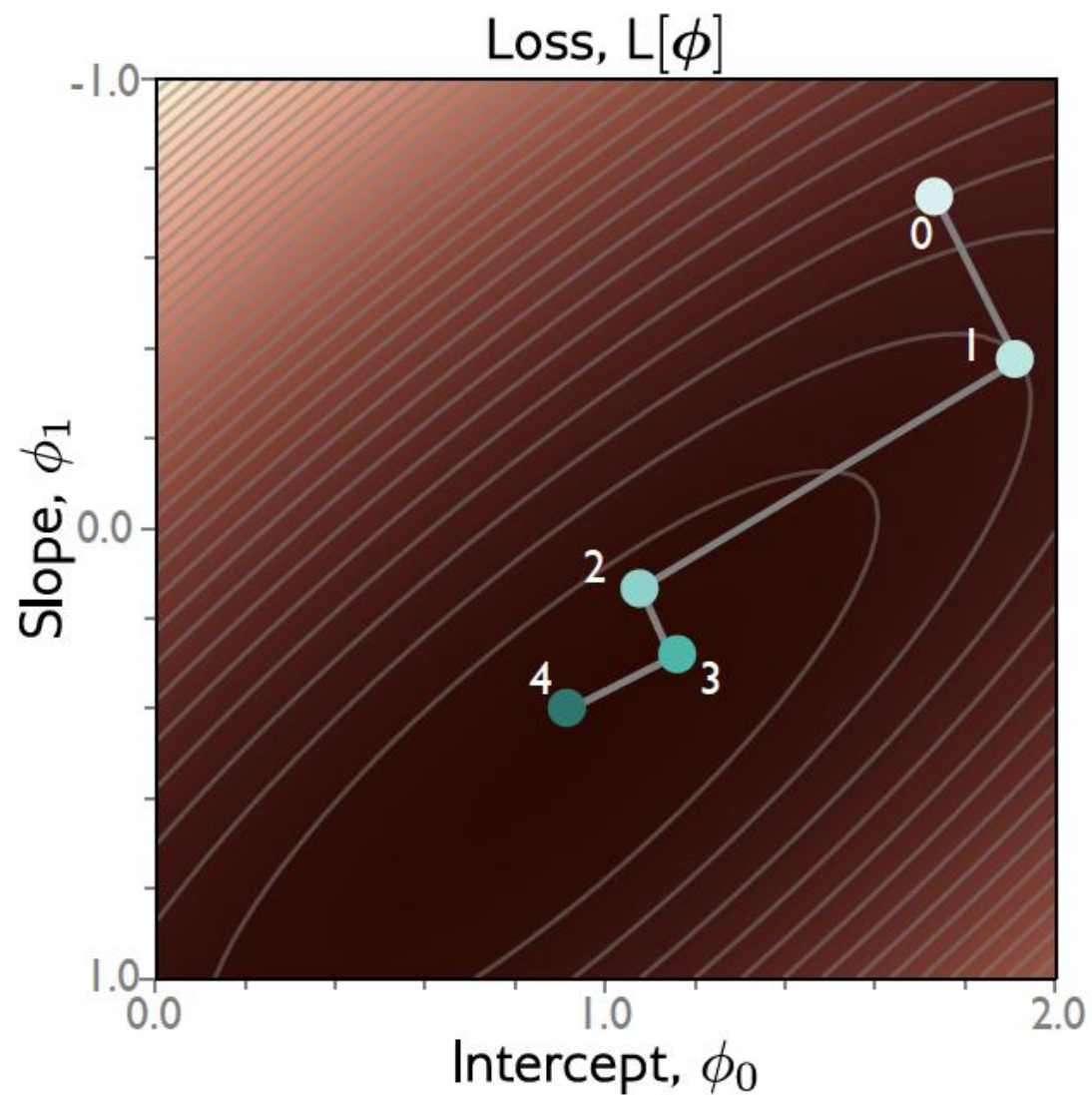
Descenso gradual



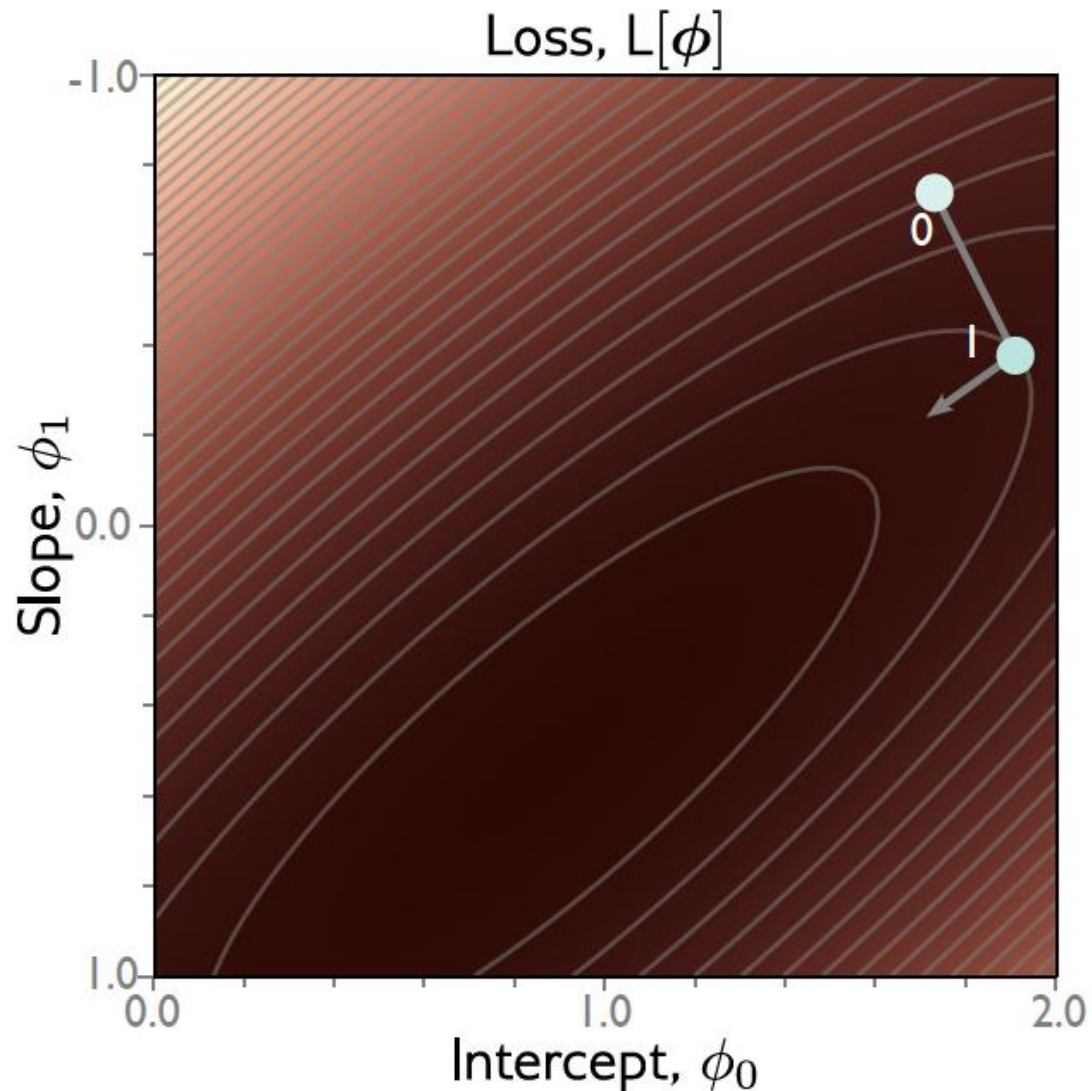
Descenso gradual



Descenso gradual



Búsqueda en línea



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

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

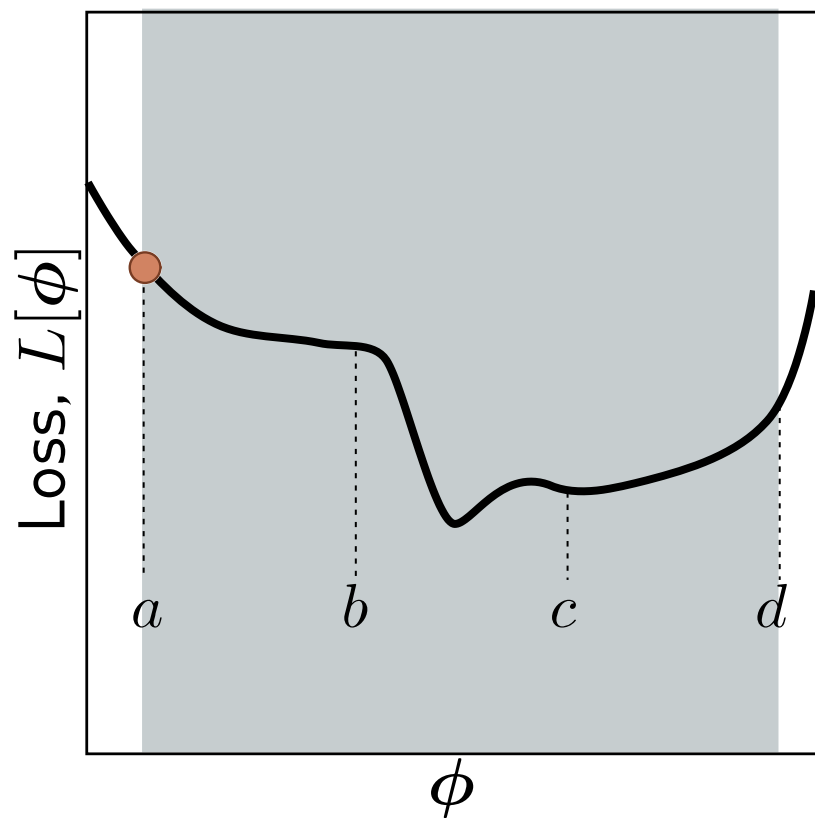
$$\frac{\partial \ell_i}{\partial \phi} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

Paso 2: Actualizar los parámetros según la regla

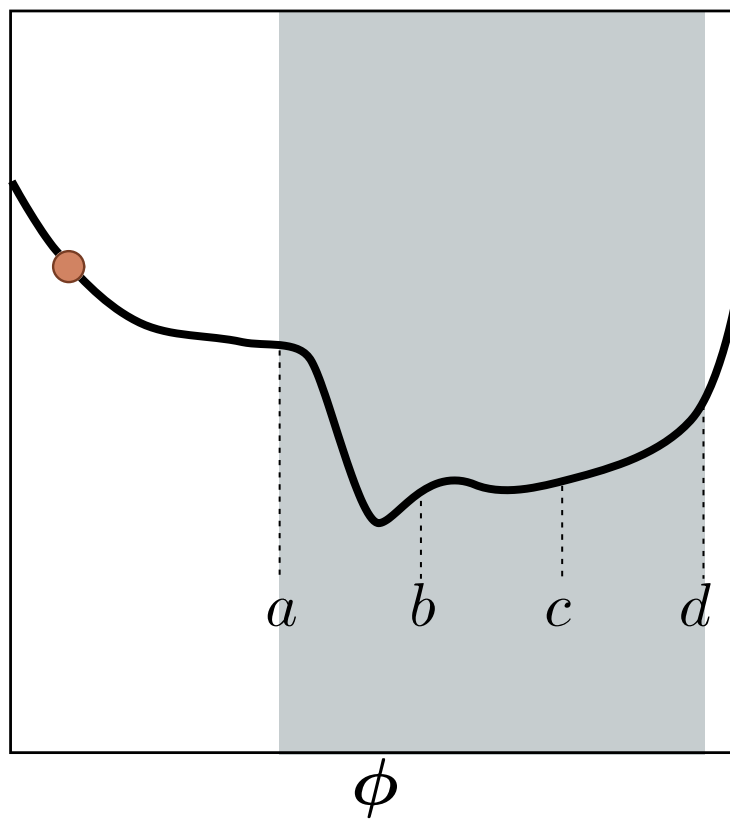
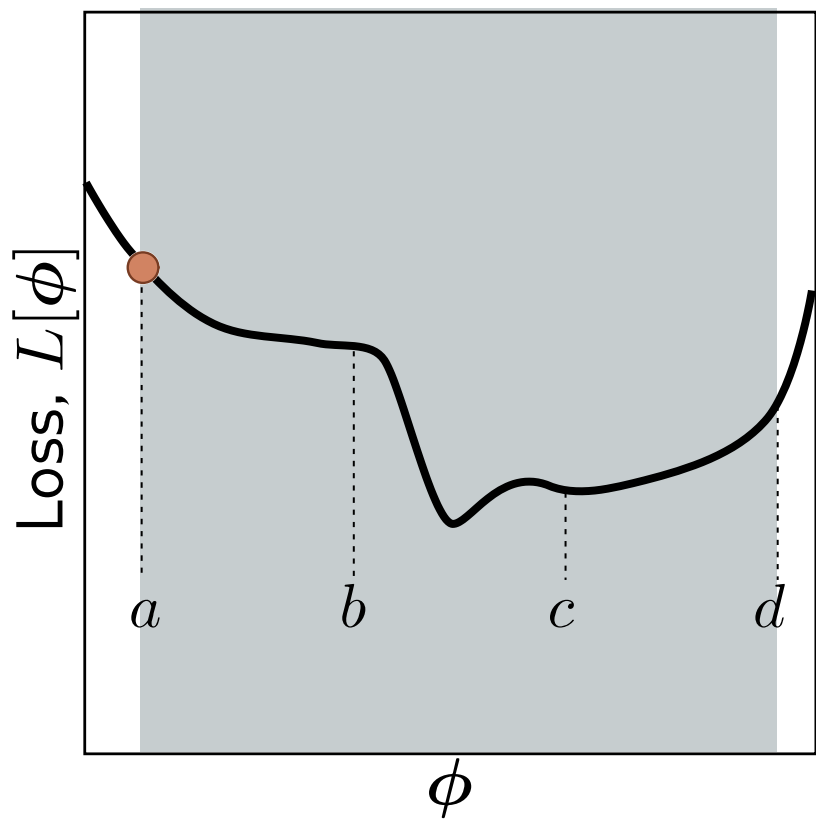
$$\phi \longleftarrow \phi - \alpha \frac{\partial L}{\partial \phi}$$

= tamaño del paso

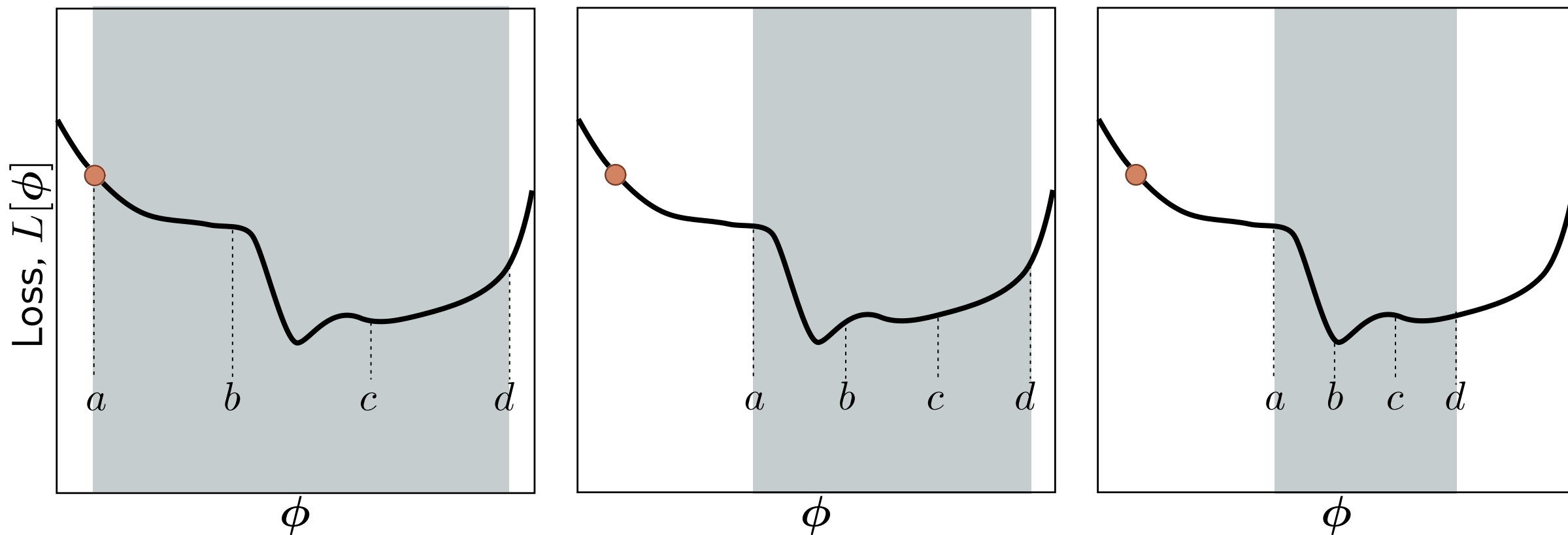
Búsqueda de líneas (entre corchetes)



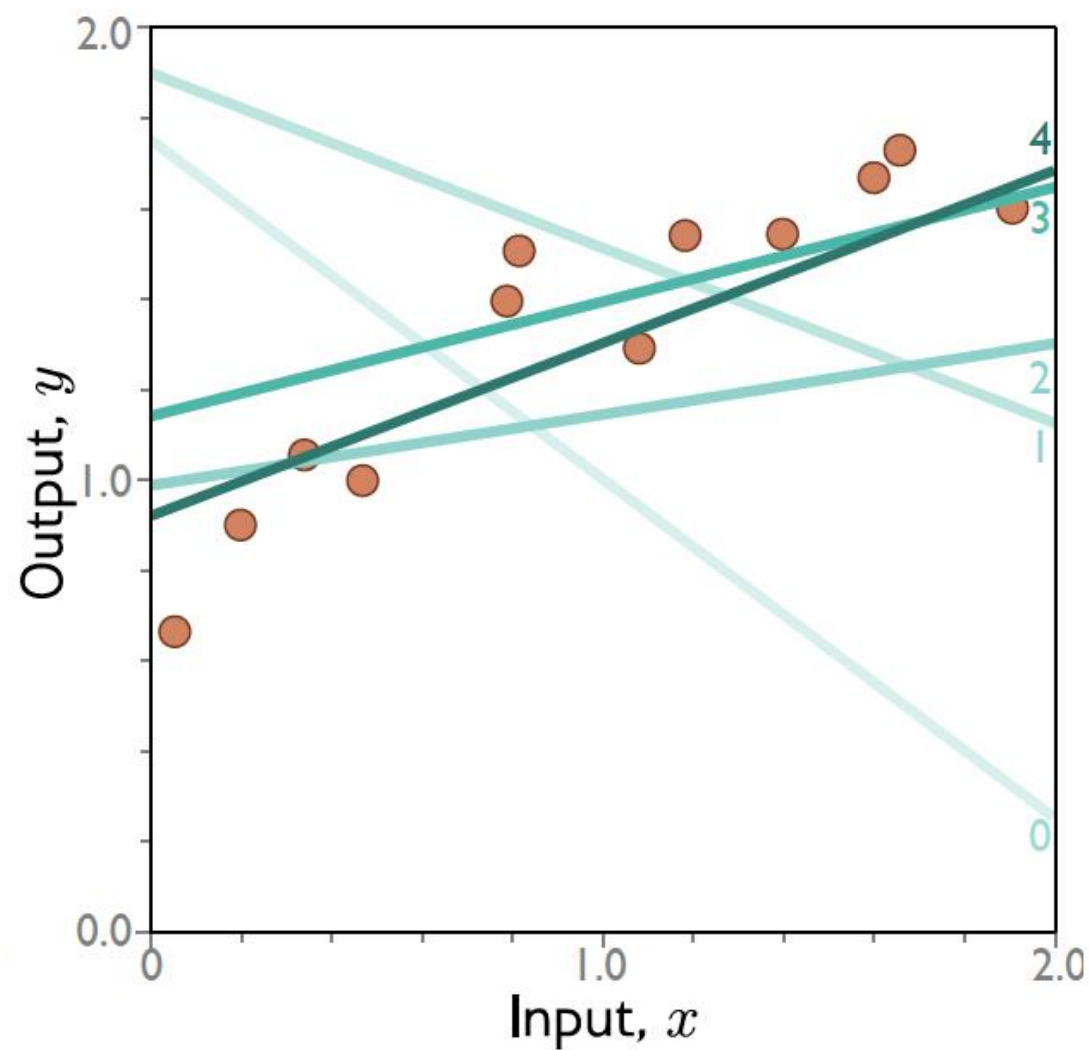
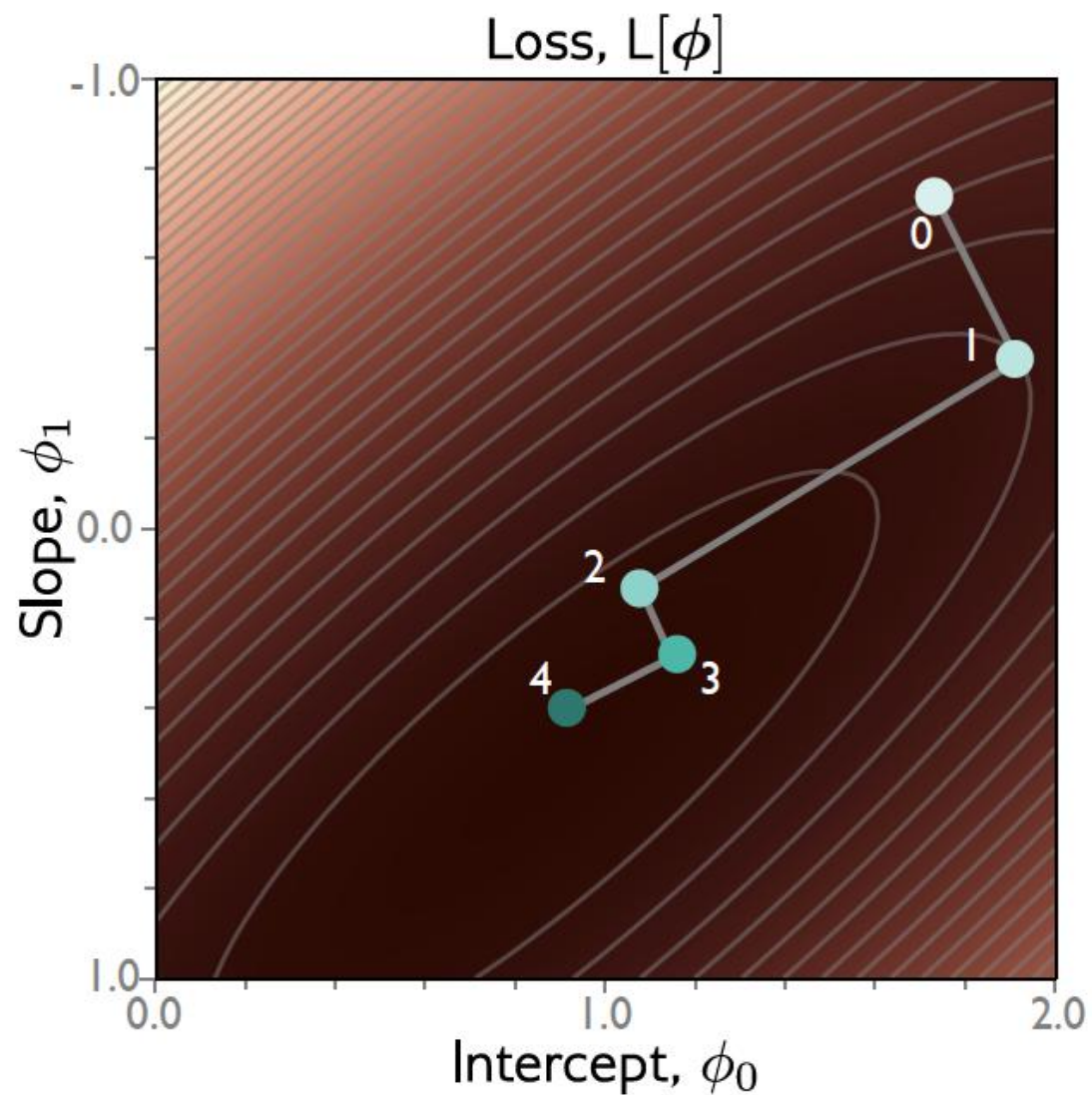
Búsqueda de líneas (entre corchetes)



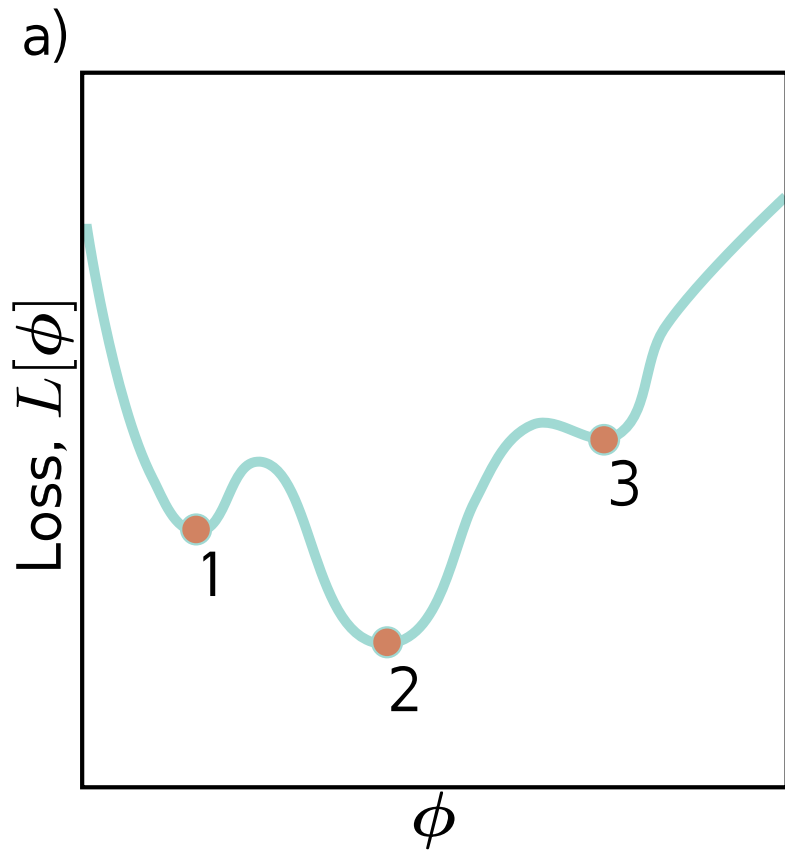
Búsqueda de líneas (entre corchetes)



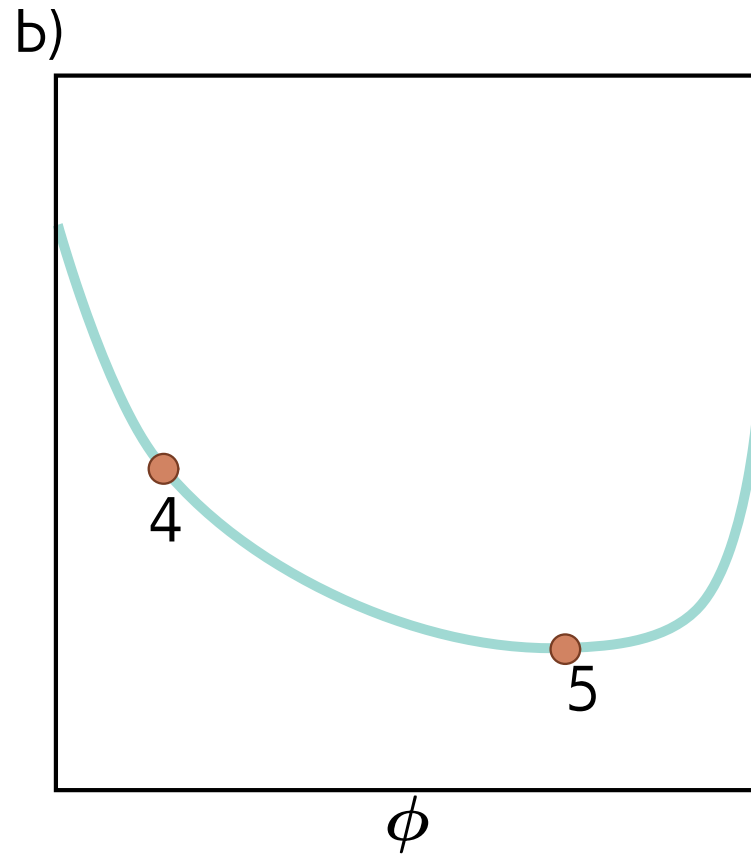
Descenso gradual



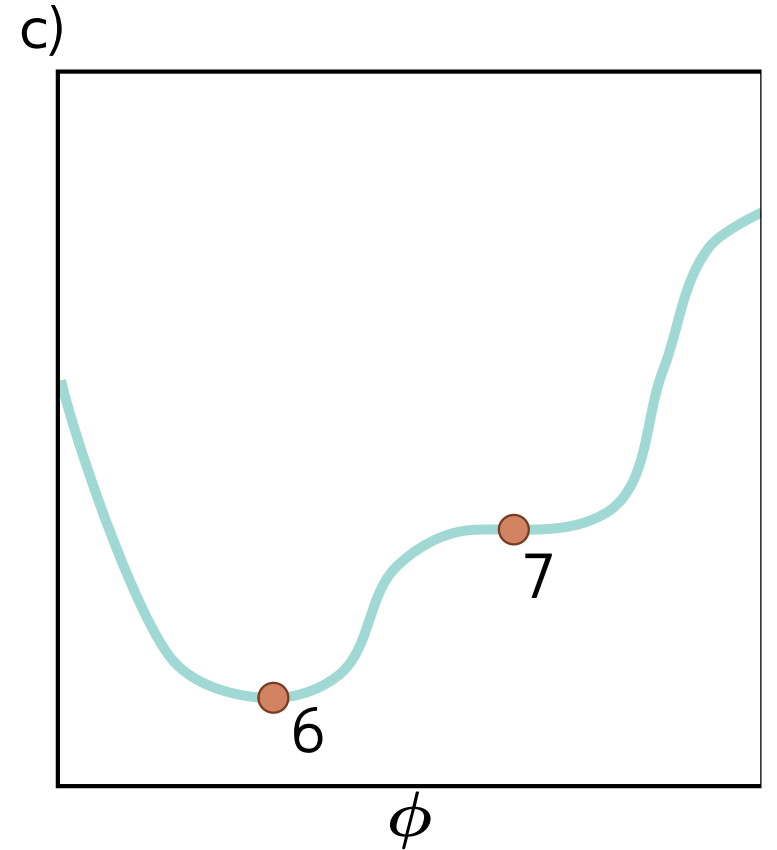
Problemas convexos



No convexo

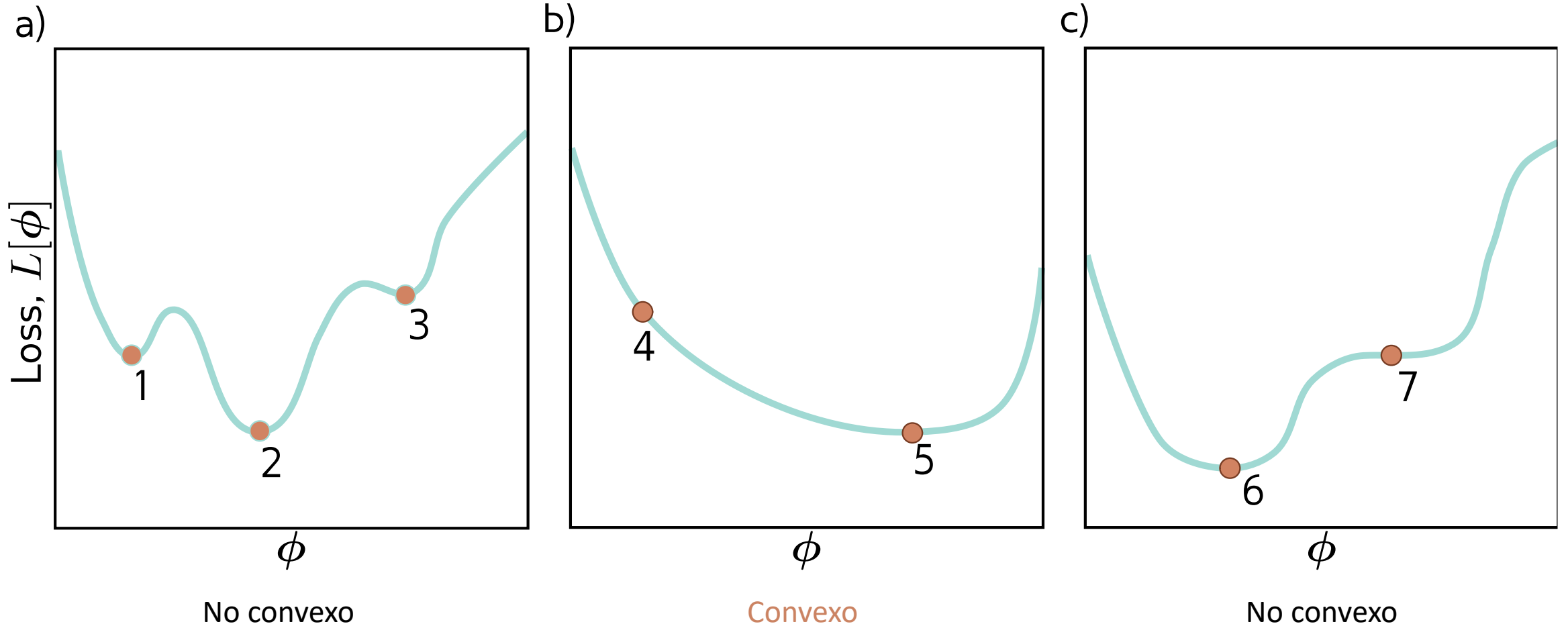


Convexo



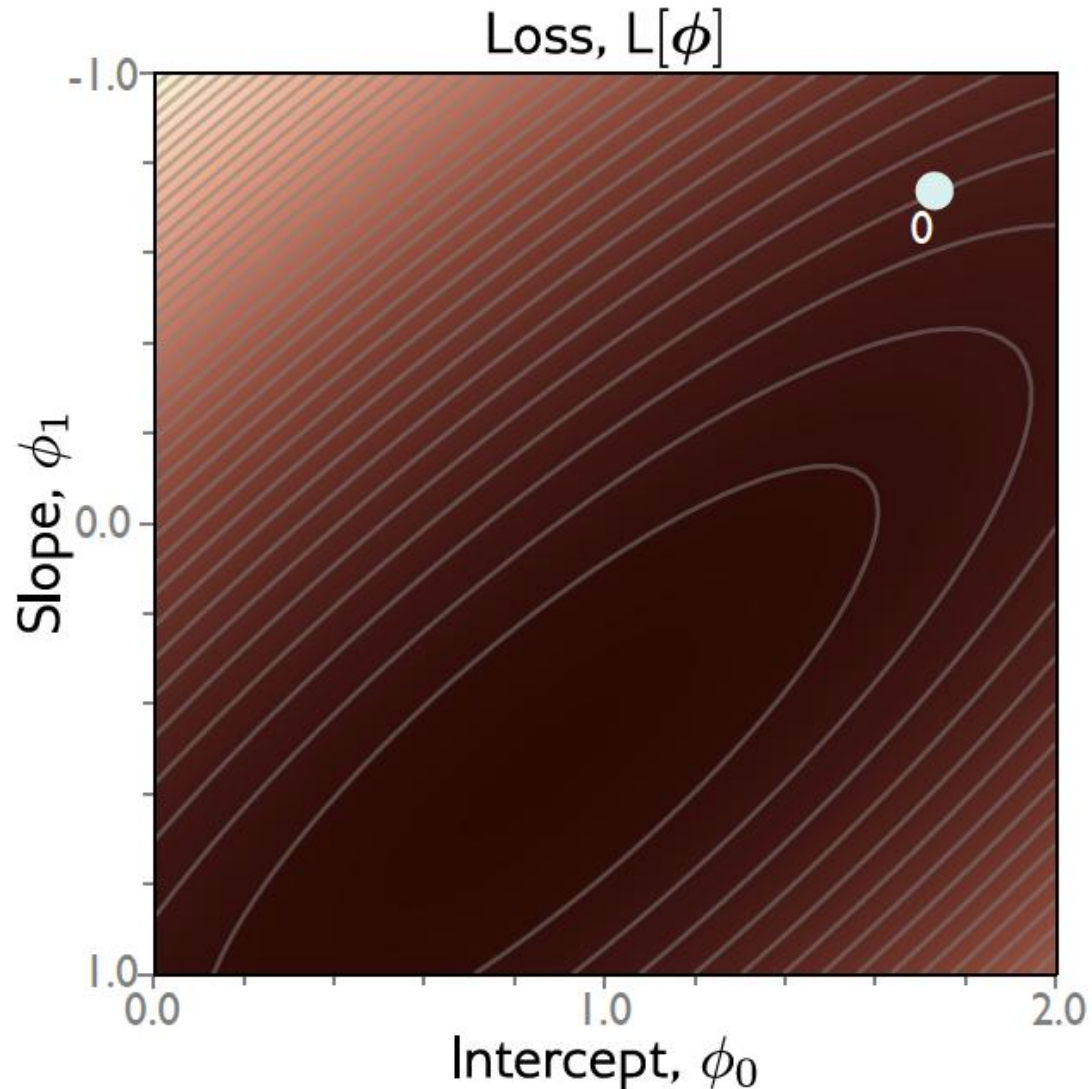
No convexo

Problemas convexos



La prueba de convexidad es que 2nd la derivada es positiva en todas partes

Convexidad en dimensiones superiores



La prueba de convexidad es que el determinante del hessiano (2^{da} matriz derivada) es positivo en todas partes.

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

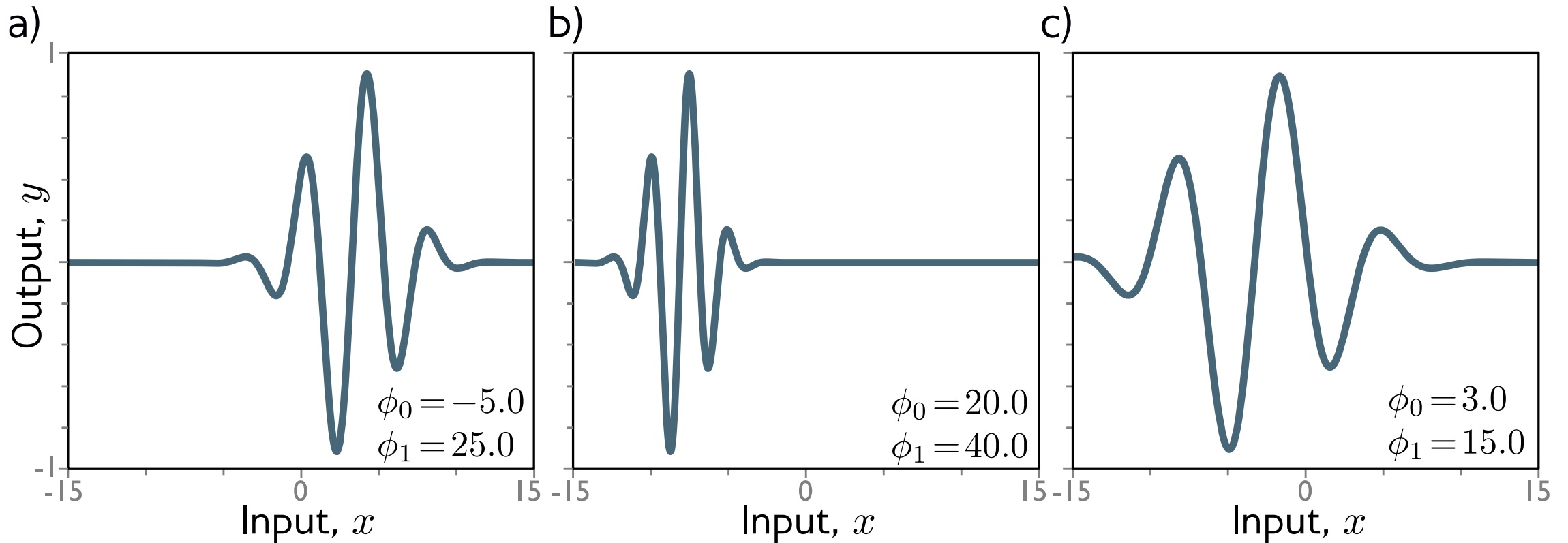
$$\mathbf{H}[\phi] = \frac{\partial^2 L}{\partial \phi_0^2} \frac{\partial^2 L}{\partial \phi_1^2} - \frac{\partial^2 L}{\partial \phi_0 \partial \phi_1} \frac{\partial^2 L}{\partial \phi_1 \partial \phi_0}$$

Modelos de ajuste

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

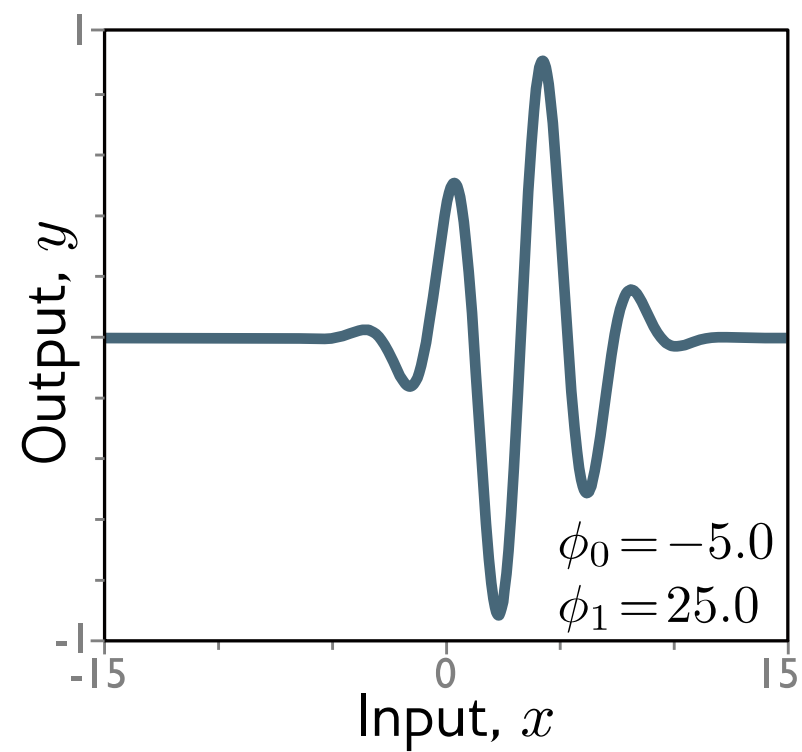
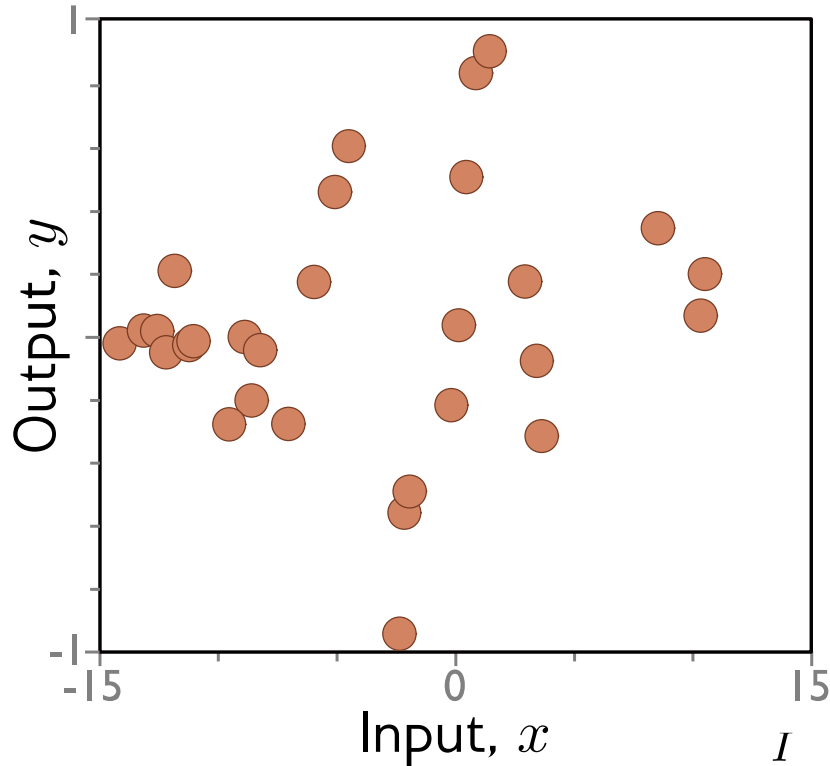
Modelo Gabor

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

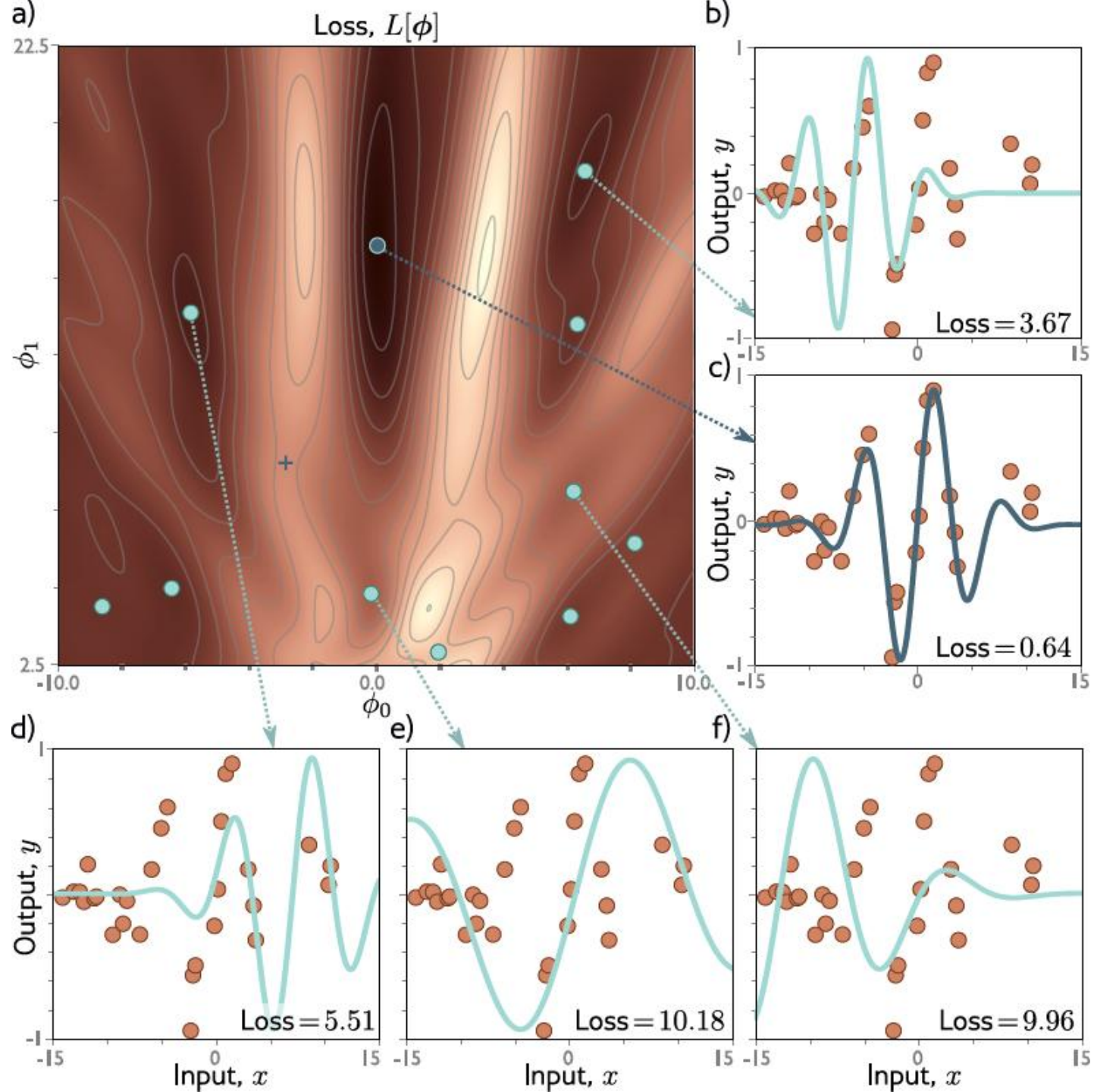


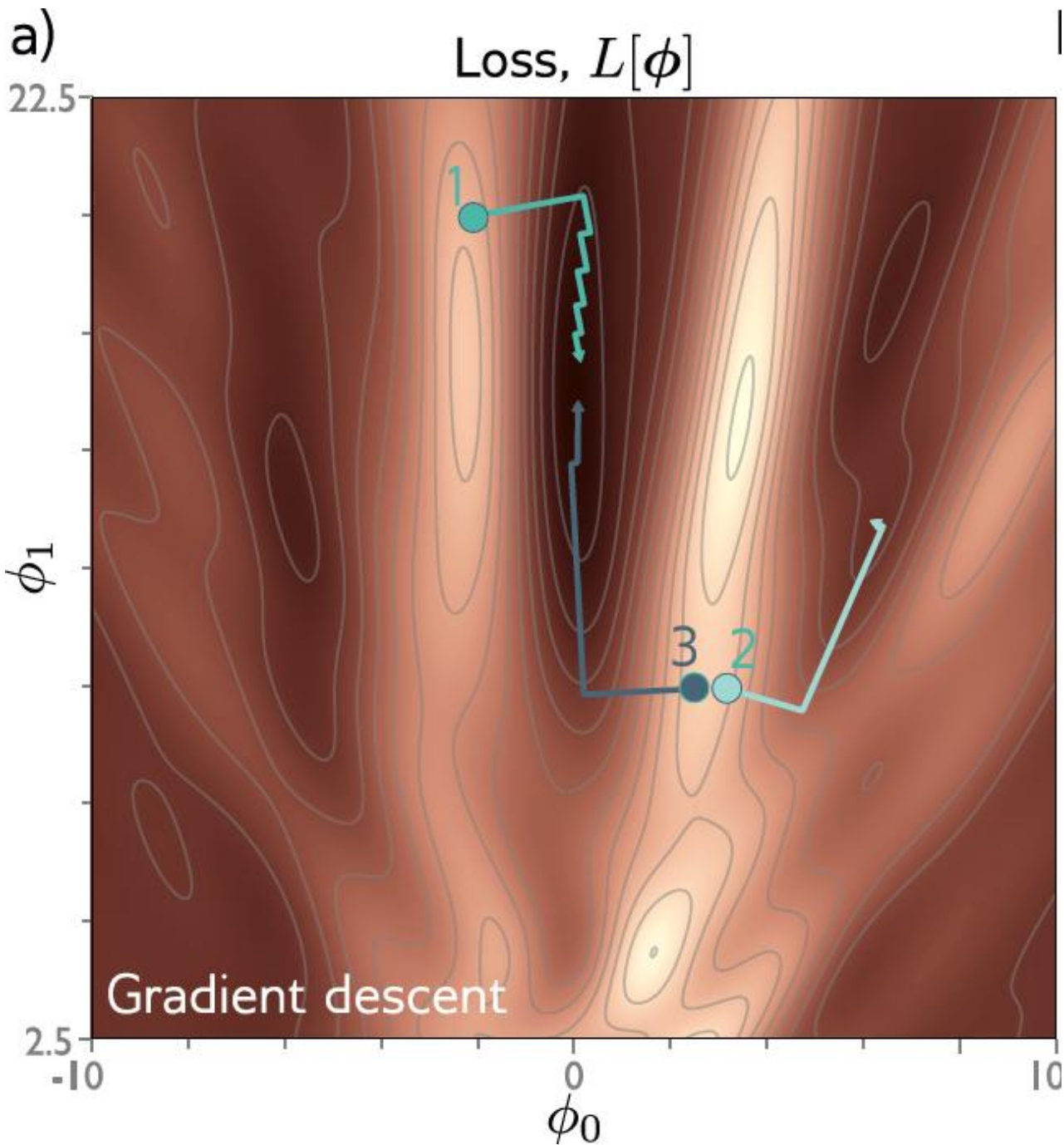
Modelo Gabor

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



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

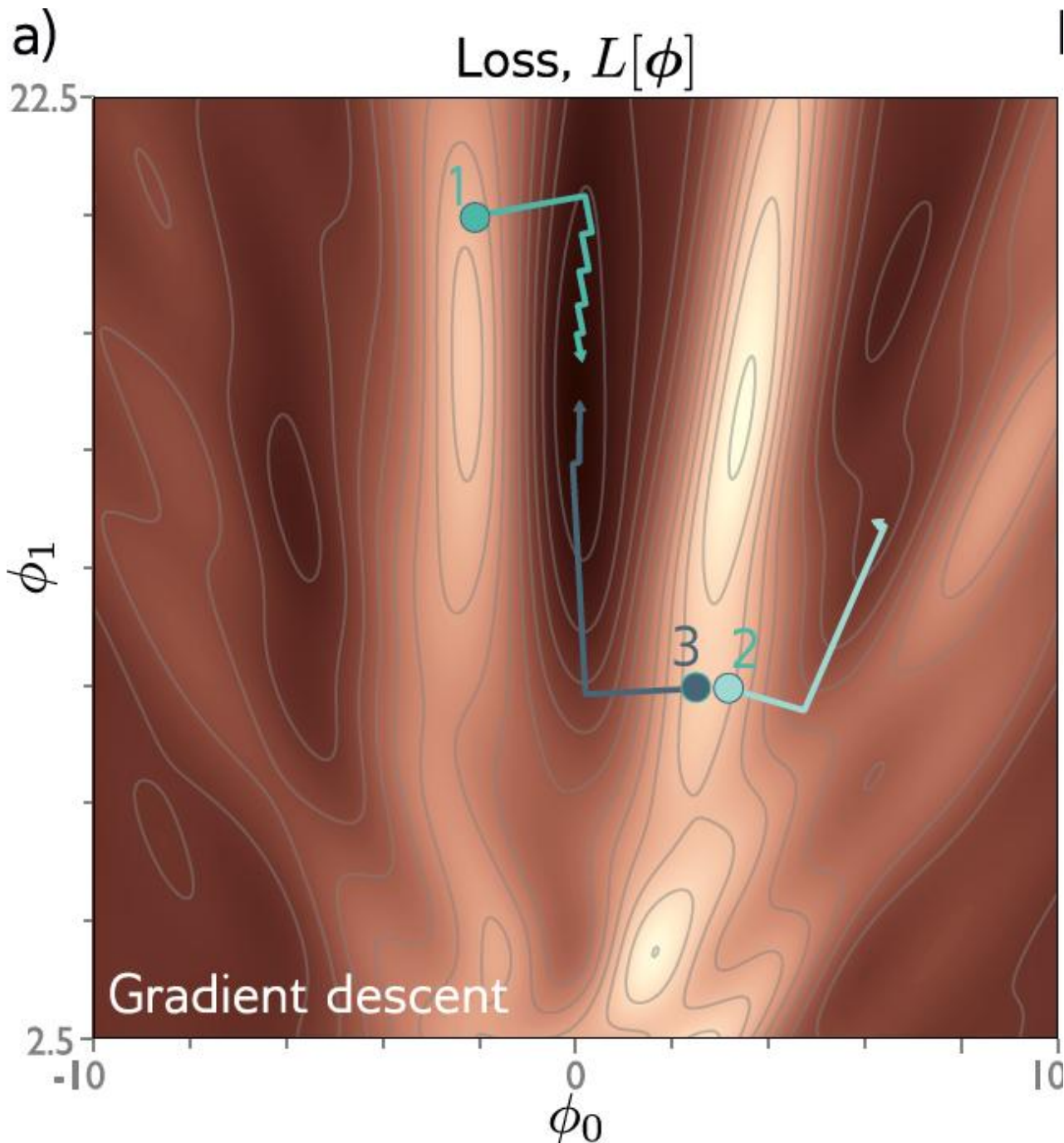




- El descenso gradiente llega al mínimo global si empezamos en el "valle" correcto
- En caso contrario, descenso a un mínimo local
- O quedarse atascado cerca de un punto de silla (*saddle point*) de montar

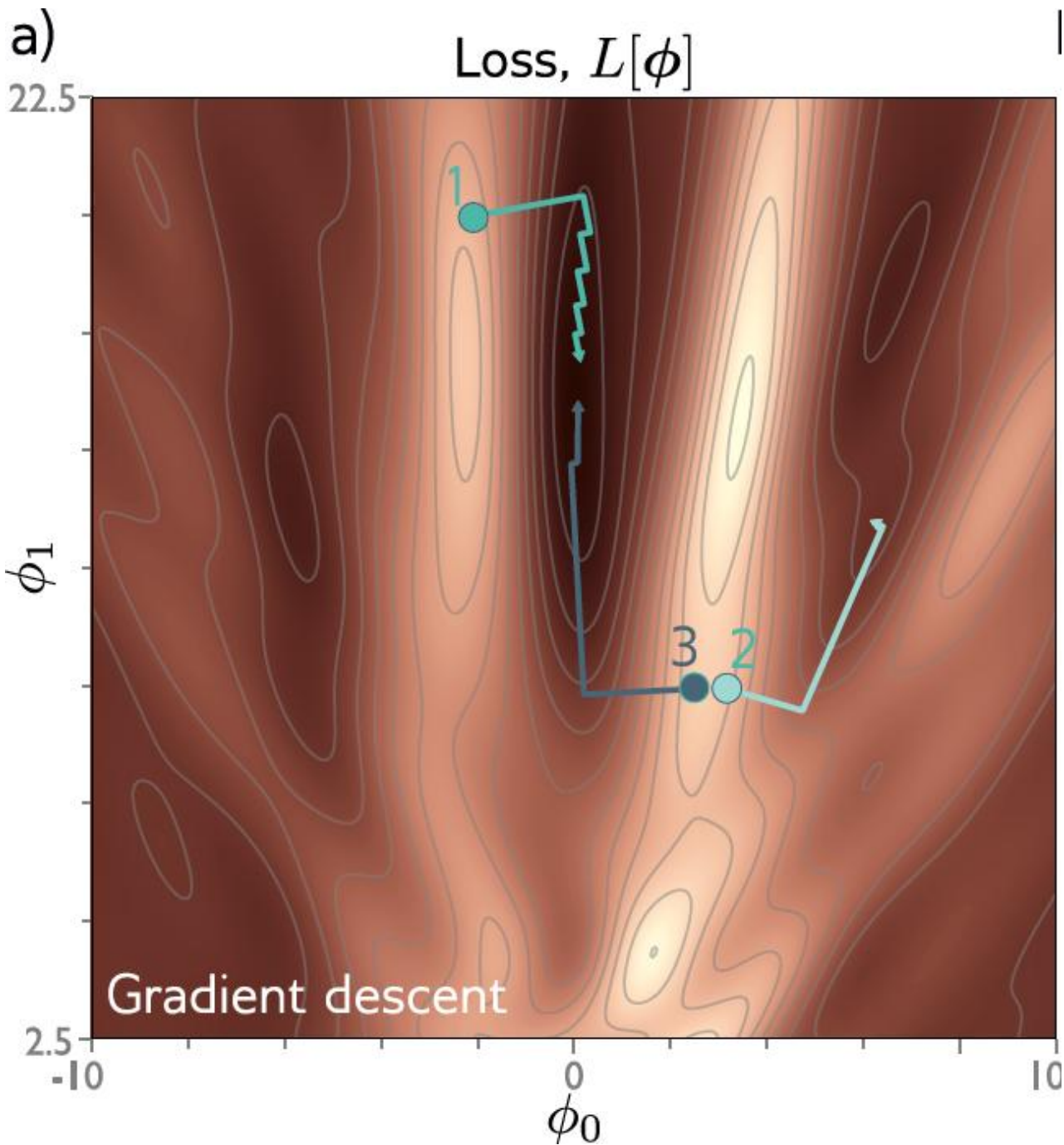
Modelos de ajuste

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



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*)



Descenso de gradiente estocástico

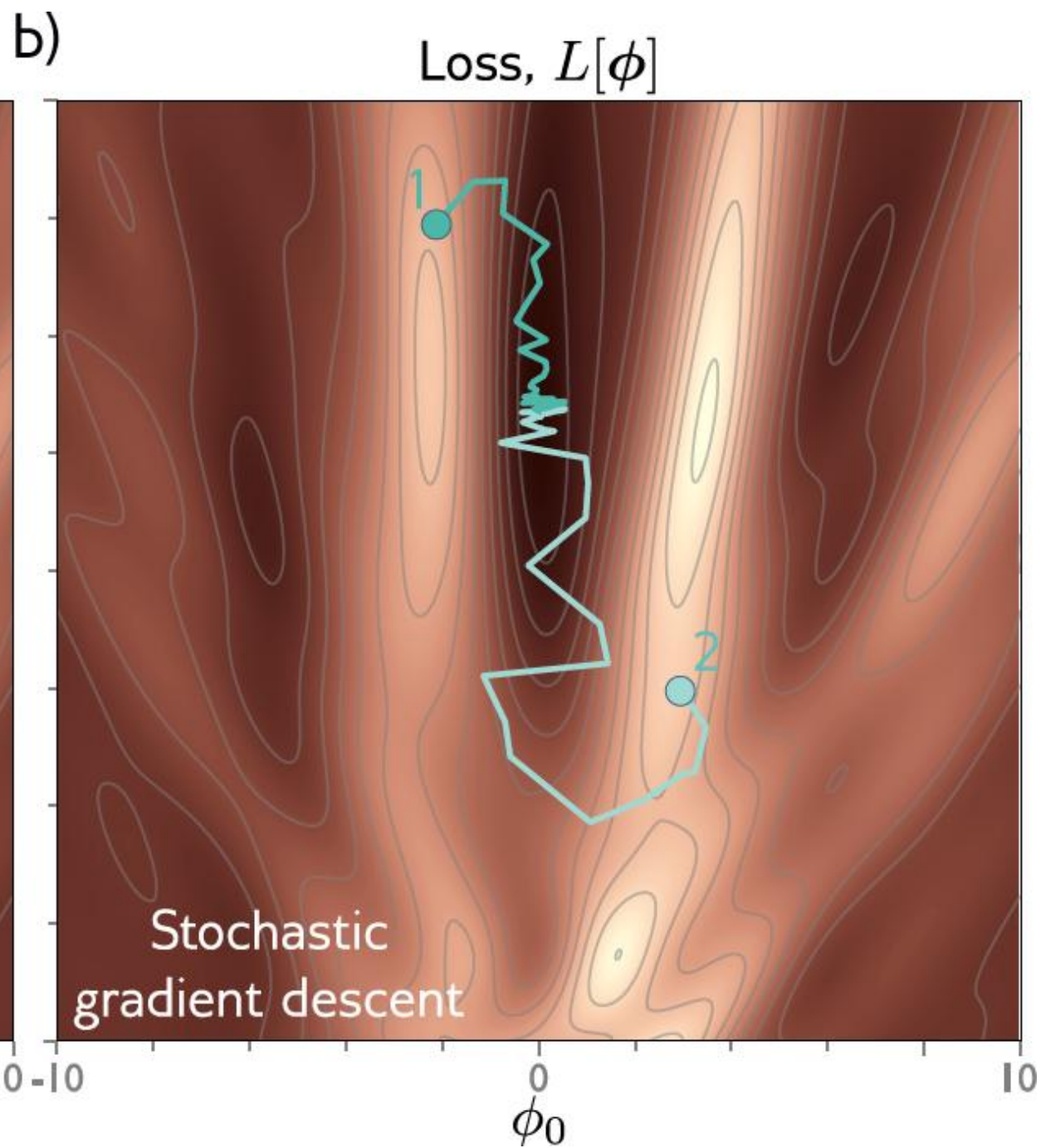
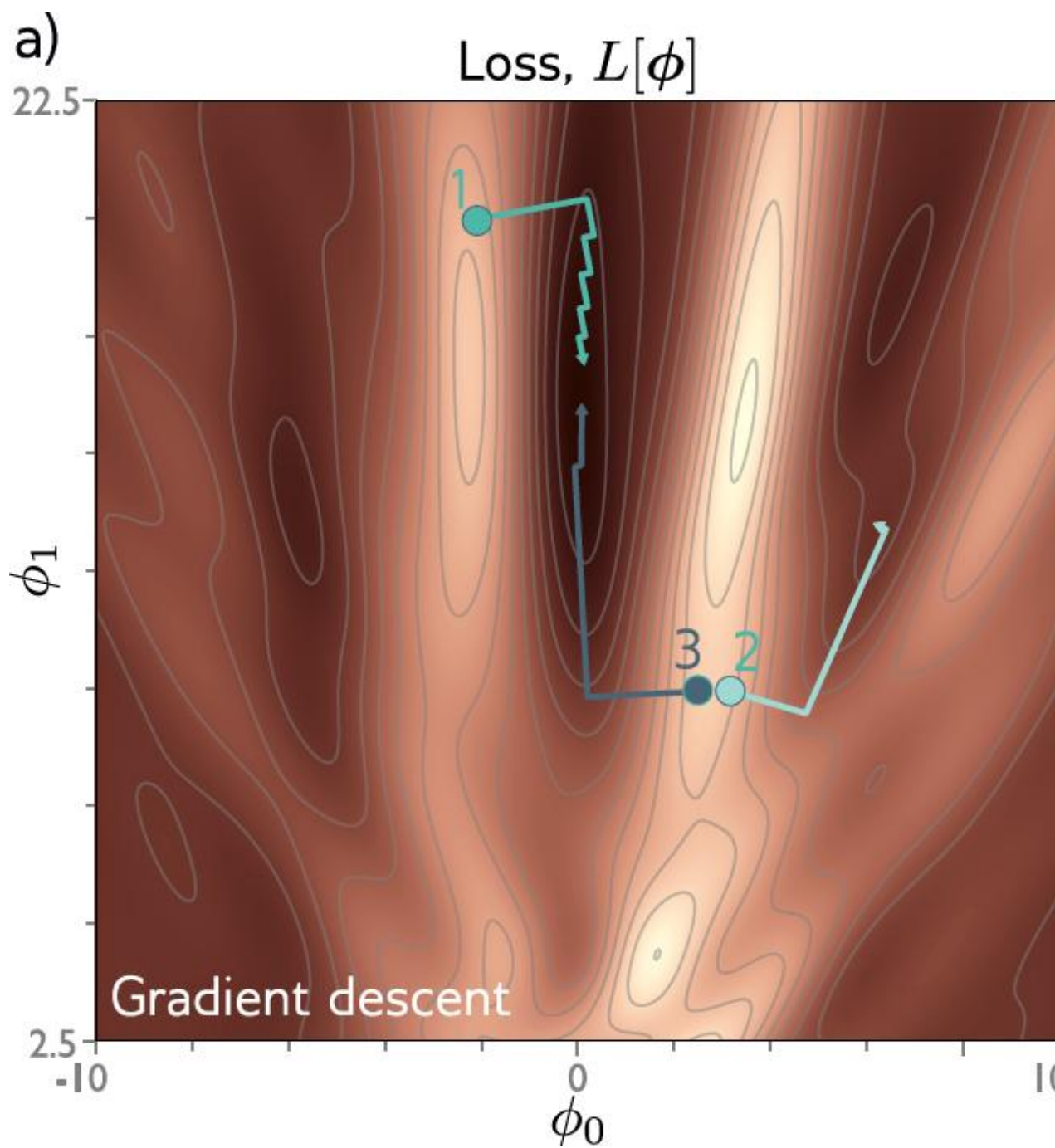
Antes (descenso por lotes completos)

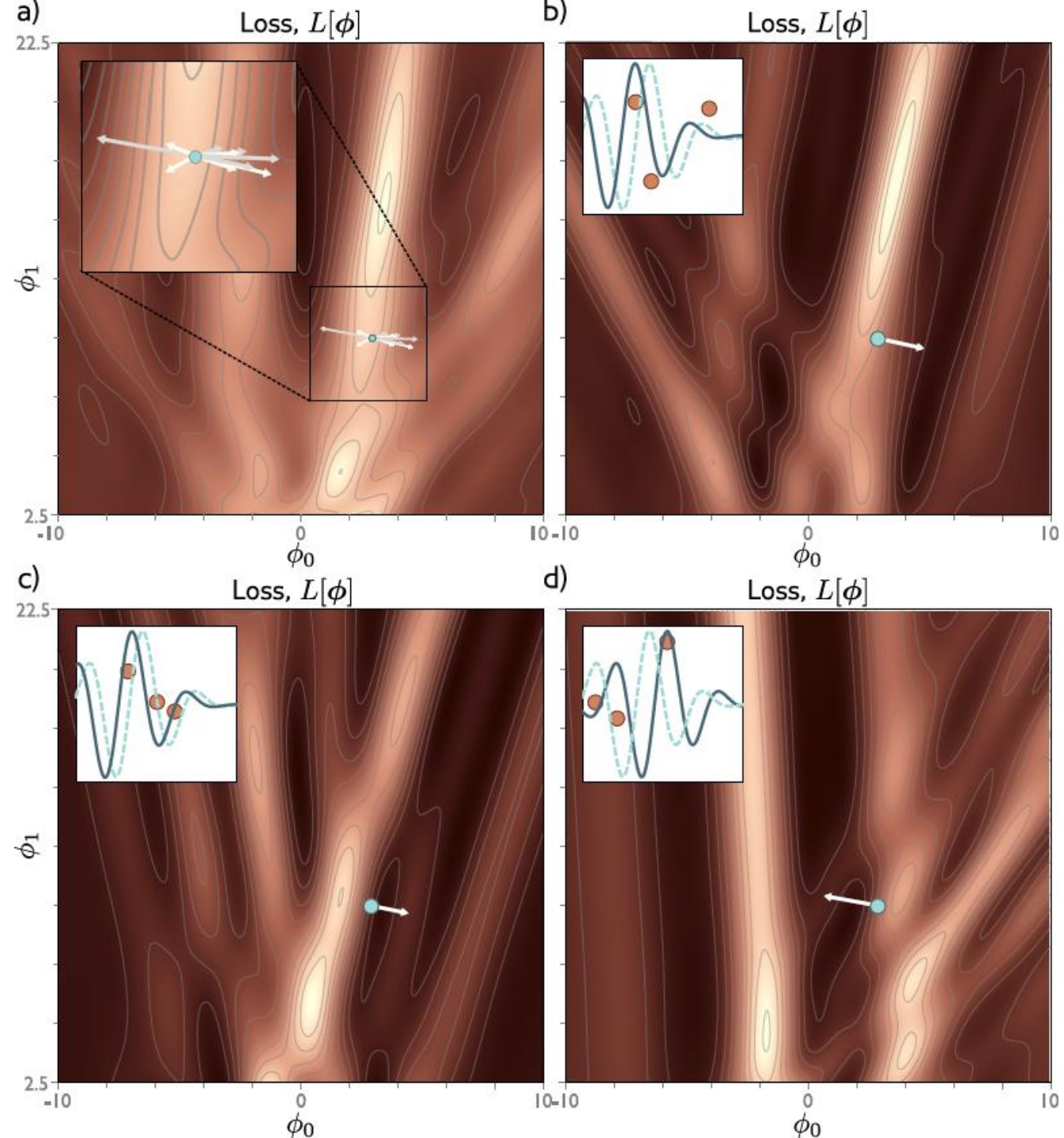
$$\phi_{t+1} \longleftarrow \phi_t - \alpha \sum_{i=1}^I \frac{\partial \ell_i[\phi_t]}{\partial \phi},$$

Después (SGD)

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

Tasa de aprendizaje fija





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 (α) 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

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

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

Momentum

- Suma ponderada de este gradiente y el gradiente anterior

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

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

Momentum Libro

6.3 Momentum

A common modification to stochastic gradient descent is to add a *momentum* term. We update the parameters with a weighted combination of the gradient computed from the current batch and the direction moved in the previous step:

$$\begin{aligned}\mathbf{m}_{t+1} &\leftarrow \beta \cdot \mathbf{m}_t + (1 - \beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\phi_t]}{\partial \phi} \\ \phi_{t+1} &\leftarrow \phi_t - \alpha \cdot \mathbf{m}_{t+1},\end{aligned}\tag{6.11}$$

where $\mathbf{m}_t$ is the momentum (which drives the update at iteration t), $\beta \in [0, 1)$ controls the degree to which the gradient is smoothed over time, and α is the learning rate.

The recursive formulation of the momentum calculation means that the gradient step is an infinite weighted sum of all the previous gradients, where the weights get smaller as we move back in time. The effective learning rate increases if all these gradients are aligned over multiple iterations but decreases if the gradient direction repeatedly changes as the terms in the sum cancel out. The overall effect is a smoother trajectory and reduced oscillatory behavior in valleys (figure 6.7).

Momentum

Libro

6.3 Momentum

A common modification to stochastic gradient descent is to add a *momentum* term. We update the parameters with a weighted combination of the gradient computed from the current batch and the direction moved in the previous step:

$$\begin{aligned}\mathbf{m}_{t+1} &\leftarrow \beta \cdot \mathbf{m}_t + (1 - \beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\phi_t]}{\partial \phi} \\ \phi_{t+1} &\leftarrow \phi_t - \alpha \cdot \mathbf{m}_{t+1},\end{aligned}\tag{6.11}$$

where $\mathbf{m}_t$ is the momentum (which drives the update at iteration t), $\beta \in [0, 1)$ controls the degree to which the gradient is smoothed over time, and α is the learning rate.

The recursive formulation of the momentum calculation means that the gradient step is an infinite weighted sum of all the previous gradients, where the weights get smaller as we move back in time. The effective learning rate increases if all these gradients are aligned over multiple iterations but decreases if the gradient direction repeatedly changes as the terms in the sum cancel out. The overall effect is a smoother trajectory and reduced oscillatory behavior in valleys (figure 6.7).

$\beta \in [0, 1)$ Controla el grado en el cual el gradiente es suavizado a lo largo del tiempo y α es la tasa de aprendizaje

Momentum

Libro

6.3 Momentum

A common modification to stochastic gradient descent is to add a *momentum* term. We update the parameters with a weighted combination of the gradient computed from the current batch and the direction moved in the previous step:

$$\begin{aligned}\mathbf{m}_{t+1} &\leftarrow \beta \cdot \mathbf{m}_t + (1 - \beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\phi_t]}{\partial \phi} \\ \phi_{t+1} &\leftarrow \phi_t - \alpha \cdot \mathbf{m}_{t+1},\end{aligned}\tag{6.11}$$

where $\mathbf{m}_t$ is the momentum (which drives the update at iteration t), $\beta \in [0, 1)$ controls the degree to which the gradient is smoothed over time, and α is the learning rate.

The recursive formulation of the momentum calculation means that the gradient step is an infinite weighted sum of all the previous gradients, where the weights get smaller as we move back in time. The effective learning rate increases if all these gradients are aligned over multiple iterations but decreases if the gradient direction repeatedly changes as the terms in the sum cancel out. The overall effect is a smoother trajectory and reduced oscillatory behavior in valleys (figure 6.7).

La formulación recursive del cálculo del momentum significa que el paso del gradiente es una suma ponderada infinita de todos gradientes previos, donde los pesos se hacen más pequeños a medida que avanza el tiempo

Momentum

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

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

i) Asumimos $m_0 = 0$

$$(ii) \sum_{i \in \beta_t} \frac{\partial \ell_i [\phi_t]}{\partial \phi} = g_t \quad g_0 \sim? \neq 0$$

Precondiciones

$$m_{t+1} = (1 - \beta) \sum_{k=0}^t \beta^{t-k} g_k.$$

El momentum es la suma ponderada de los gradientes

Momentum

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

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

i) Asumimos $m_0 = 0$

$$(ii) \sum_{i \in \beta_t} \frac{\partial \ell_i [\phi_t]}{\partial \phi} = g_t \quad g_0 \sim? \neq 0$$

Precondiciones

$$\mathbf{m}_1 = \beta \mathbf{m}_0 + (1 - \beta) \mathbf{g}_0 = (1 - \beta) \mathbf{g}_0$$

$$\mathbf{m}_2 = \beta \mathbf{m}_1 + (1 - \beta) \mathbf{g}_1 = \beta(1 - \beta) \mathbf{g}_0 + (1 - \beta) \mathbf{g}_1 = (1 - \beta) [\beta \mathbf{g}_0 + \mathbf{g}_1]$$

$$\mathbf{m}_3 = \beta \mathbf{m}_2 + (1 - \beta) \mathbf{g}_2 = (1 - \beta) [\beta^2 \mathbf{g}_0 + \beta \mathbf{g}_1 + \mathbf{g}_2]$$

Expandiendo 3 términos

Momentum

Libro

6.3 Momentum

A common modification to stochastic gradient descent is to add a *momentum* term. We update the parameters with a weighted combination of the gradient computed from the current batch and the direction moved in the previous step:

$$\begin{aligned}\mathbf{m}_{t+1} &\leftarrow \beta \cdot \mathbf{m}_t + (1 - \beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\phi_t]}{\partial \phi} \\ \phi_{t+1} &\leftarrow \phi_t - \alpha \cdot \mathbf{m}_{t+1},\end{aligned}\tag{6.11}$$

where $\mathbf{m}_t$ is the momentum (which drives the update at iteration t), $\beta \in [0, 1)$ controls the degree to which the gradient is smoothed over time, and α is the learning rate.

The recursive formulation of the momentum calculation means that the gradient step is an infinite weighted sum of all the previous gradients, where the weights get smaller as we move back in time. The effective learning rate increases if all these gradients are aligned over multiple iterations but decreases if the gradient direction repeatedly changes as the terms in the sum cancel out. The overall effect is a smoother trajectory and reduced oscillatory behavior in valleys (figure 6.7).

Momentum Libro

$$\phi_1 = \phi_0 - \alpha \mathbf{m}_1$$

$$\phi_2 = \phi_1 - \alpha \mathbf{m}_2 = \phi_0 - \alpha (\mathbf{m}_1 + \mathbf{m}_2)$$

$$\phi_3 = \phi_2 - \alpha \mathbf{m}_3 = \phi_0 - \alpha (\mathbf{m}_1 + \mathbf{m}_2 + \mathbf{m}_3)$$

$$\phi_t = \phi_0 - \alpha \sum_{s=1}^t \mathbf{m}_s$$

$$\mathbf{m}_1 = (1 - \beta) \mathbf{g}_0$$

$$\mathbf{m}_2 = (1 - \beta) [\beta \mathbf{g}_0 + \mathbf{g}_1]$$

$$\mathbf{m}_3 = (1 - \beta) [\beta^2 \mathbf{g}_0 + \beta \mathbf{g}_1 + \mathbf{g}_2]$$

$$\mathbf{m}_1 + \mathbf{m}_2 + \mathbf{m}_3 = (1 - \beta) \left[\underbrace{(1 + \beta + \beta^2)}_{\mathbf{g}_0} \mathbf{g}_0 + \underbrace{(1 + \beta)}_{\mathbf{g}_1} \mathbf{g}_1 + \underbrace{(1)}_{\mathbf{g}_2} \mathbf{g}_2 \right]$$

Expandimos 3 términos

Generalizamos para un t

$$\mathbf{m}_{t+1} = (1 - \beta) \sum_{k=0}^t \beta^{t-k} \mathbf{g}_k$$

Reorganizamos la suma para cada g

Momentum Libro

$$\mathbf{m}_1 = (1 - \beta)\mathbf{g}_0$$

$$\mathbf{m}_2 = (1 - \beta) [\beta\mathbf{g}_0 + \mathbf{g}_1]$$

$$\mathbf{m}_3 = (1 - \beta) [\beta^2\mathbf{g}_0 + \beta\mathbf{g}_1 + \mathbf{g}_2]$$

$$\mathbf{m}_1 + \mathbf{m}_2 + \mathbf{m}_3 = (1 - \beta) \left[\underbrace{(1 + \beta + \beta^2)}_{\mathbf{g}_0} \mathbf{g}_0 + \underbrace{(1 + \beta)}_{\mathbf{g}_1} \mathbf{g}_1 + \underbrace{(1)}_{\mathbf{g}_2} \mathbf{g}_2 \right]$$

$$(1 - \beta) (1 + \beta + \dots + \beta^{k-1})$$

$$\mathbf{m}_{t+1} = (1 - \beta) \sum_{k=0}^t \beta^{t-k} \mathbf{g}_k$$

Reorganizamos la suma para cada $\mathbf{g}$

Momentum Libro

$$\mathbf{m}_1 = (1 - \beta)\mathbf{g}_0$$

$$\mathbf{m}_2 = (1 - \beta) [\beta\mathbf{g}_0 + \mathbf{g}_1]$$

$$\mathbf{m}_3 = (1 - \beta) [\beta^2\mathbf{g}_0 + \beta\mathbf{g}_1 + \mathbf{g}_2]$$

$$\mathbf{m}_1 + \mathbf{m}_2 + \mathbf{m}_3 = (1 - \beta) \left[\underbrace{(1 + \beta + \beta^2)}_{\mathbf{g}_0} \mathbf{g}_0 + \underbrace{(1 + \beta)}_{\mathbf{g}_1} \mathbf{g}_1 + \underbrace{(1)}_{\mathbf{g}_2} \mathbf{g}_2 \right]$$

$$(1 - \beta) (1 + \beta + \dots + \beta^{k-1}) = 1 - \beta^k$$

$$\phi_3 = \phi_0 - \alpha [(1 - \beta^3)g_0 + (1 - \beta^2)g_1 + (1 - \beta)g_2]$$

$$\phi_t = \phi_0 - \alpha \sum_{k=0}^{t-1} (1 - \beta^{t-k})g_k$$

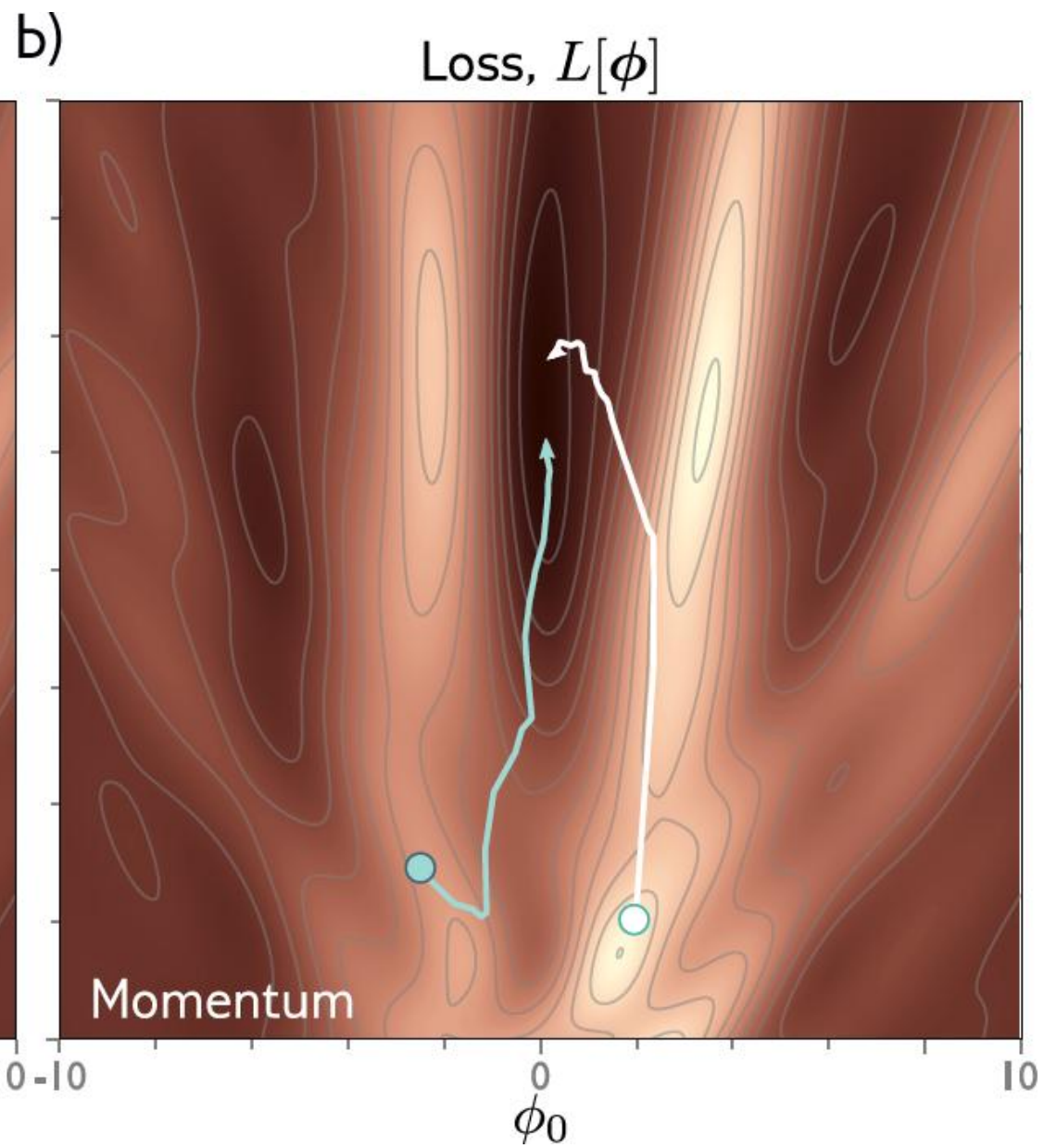
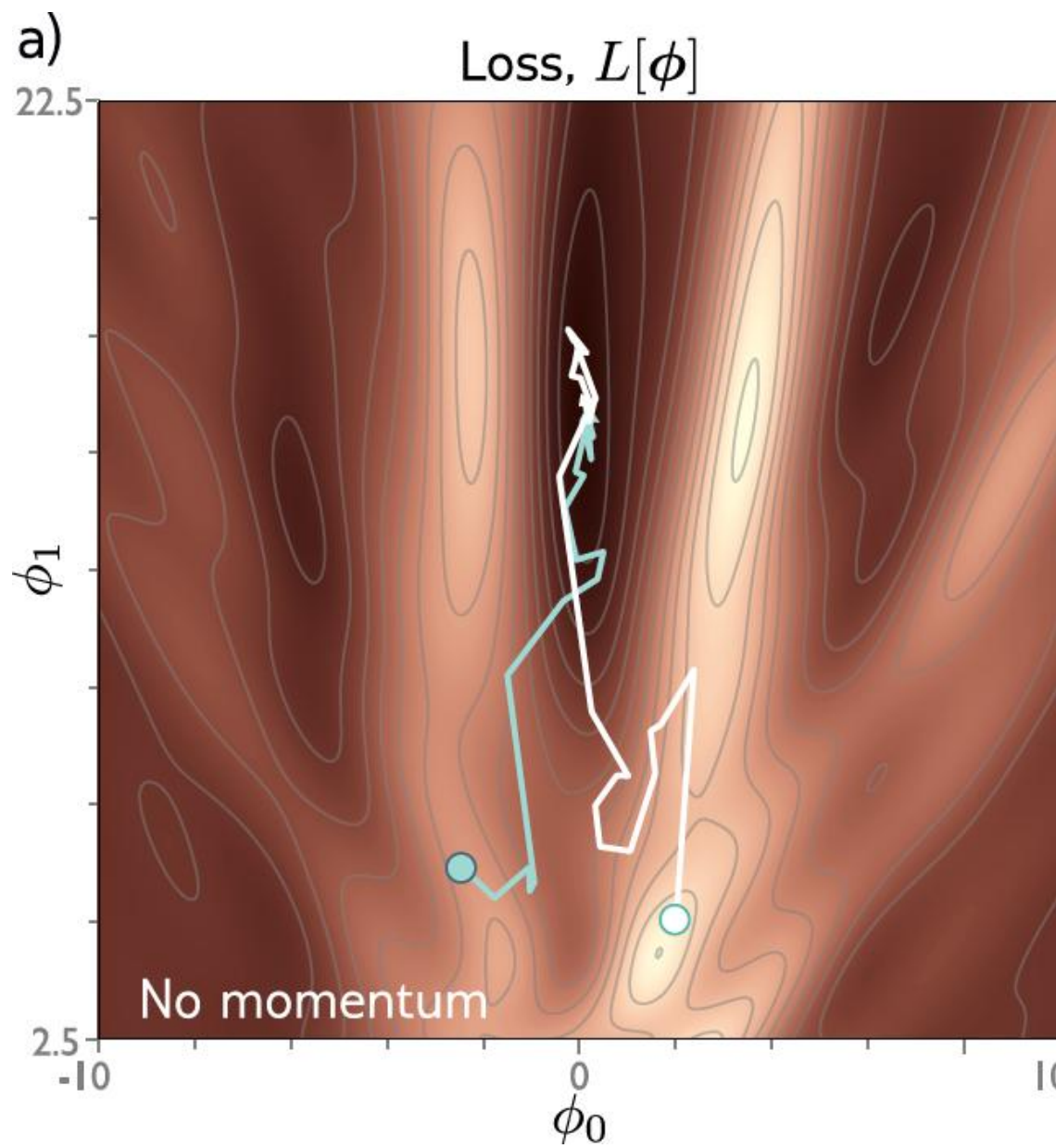
$$\mathbf{m}_{t+1} = (1 - \beta) \sum_{k=0}^t \beta^{t-k} \mathbf{g}_k$$

Reorganizamos la suma para cada $\mathbf{g}$

Cancelando sumandos

Reescribiendo la suma para cada $\mathbf{g}$

Qué pasa cuando t es grande?



Nesterov acelera el momentum

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

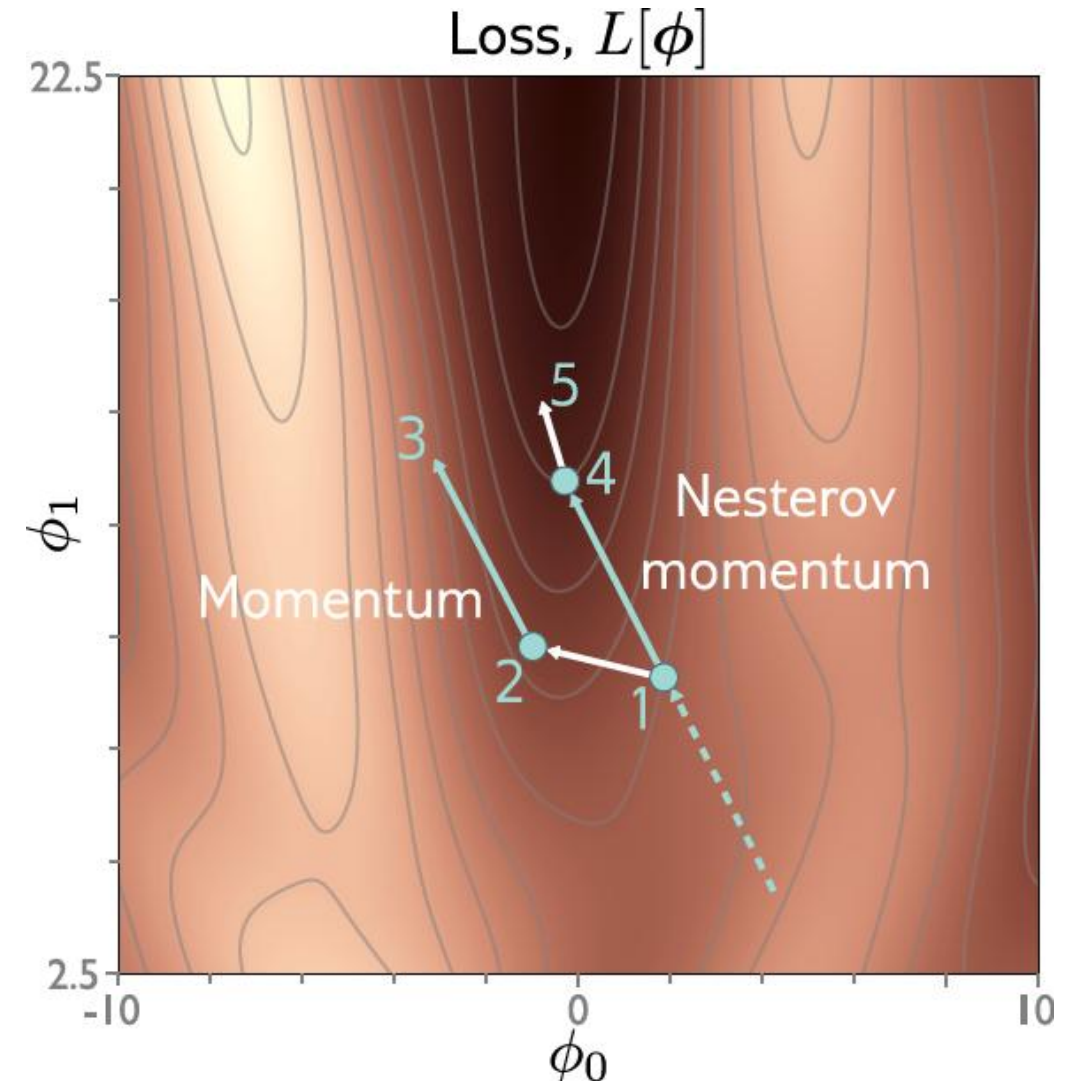
$$\mathbf{m}_{t+1} \leftarrow \beta \cdot \mathbf{m}_t + (1 - \beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\phi_t]}{\partial \phi}$$

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

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

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

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



Nesterov acelera el momentum

- El lookahead point $\phi_t - \alpha \mathbf{m}_t$ se puede entender usando la expansion de Taylor hasta la Segunda componente alrededor de ϕ_t

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

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

$$\frac{\partial \ell_i [\phi_t - \alpha \mathbf{m}_t]}{\partial \phi} = \nabla \ell_i (\phi_t - \alpha \mathbf{m}_t) \approx \nabla \ell_i (\phi_t) - \alpha \nabla^2 \ell_i (\phi_t) \mathbf{m}_t.$$

Nesterov acelera el momentum

- El lookahead point $\phi_t - \alpha \mathbf{m}_t$ se puede entender usando la expansión de Taylor hasta la Segunda componente alrededor de ϕ_t

$$\frac{\partial \ell_i [\phi_t - \alpha \mathbf{m}_t]}{\partial \phi} = \nabla \ell_i (\phi_t - \alpha \mathbf{m}_t) \approx \nabla \ell_i (\phi_t) - \alpha \nabla^2 \ell_i (\phi_t) \mathbf{m}_t.$$

- Esto es en notación con Batches. $\mathbf{H}_i = \nabla^2 \ell_i$ El hessiano

$$\sum_{i \in B_t} \frac{\partial \ell_i [\phi_t - \alpha \mathbf{m}_t]}{\partial \phi} \approx \underbrace{\sum_{i \in B_t} \nabla \ell_i (\phi_t)}_{\text{gradiente ordinario del lote en } \phi_t} - \alpha \underbrace{\left(\sum_{i \in B_t} \mathbf{H}_i (\phi_t) \right)}_{\text{producto Hessiano-vector}} \mathbf{m}_t,$$

Nesterov acelera el momentum

- Esto es en notación con Batches. $\mathbf{H}_i = \nabla^2 \ell_i$ El hessiano

$$\sum_{i \in B_t} \frac{\partial \ell_i [\phi_t - \alpha \mathbf{m}_t]}{\partial \phi} \approx \underbrace{\sum_{i \in B_t} \nabla \ell_i (\phi_t)}_{\text{gradiente ordinario del lote en } \phi_t} - \alpha \underbrace{\left(\sum_{i \in B_t} \mathbf{H}_i (\phi_t) \right) \mathbf{m}_t}_{\text{producto Hessiano-vector}},$$

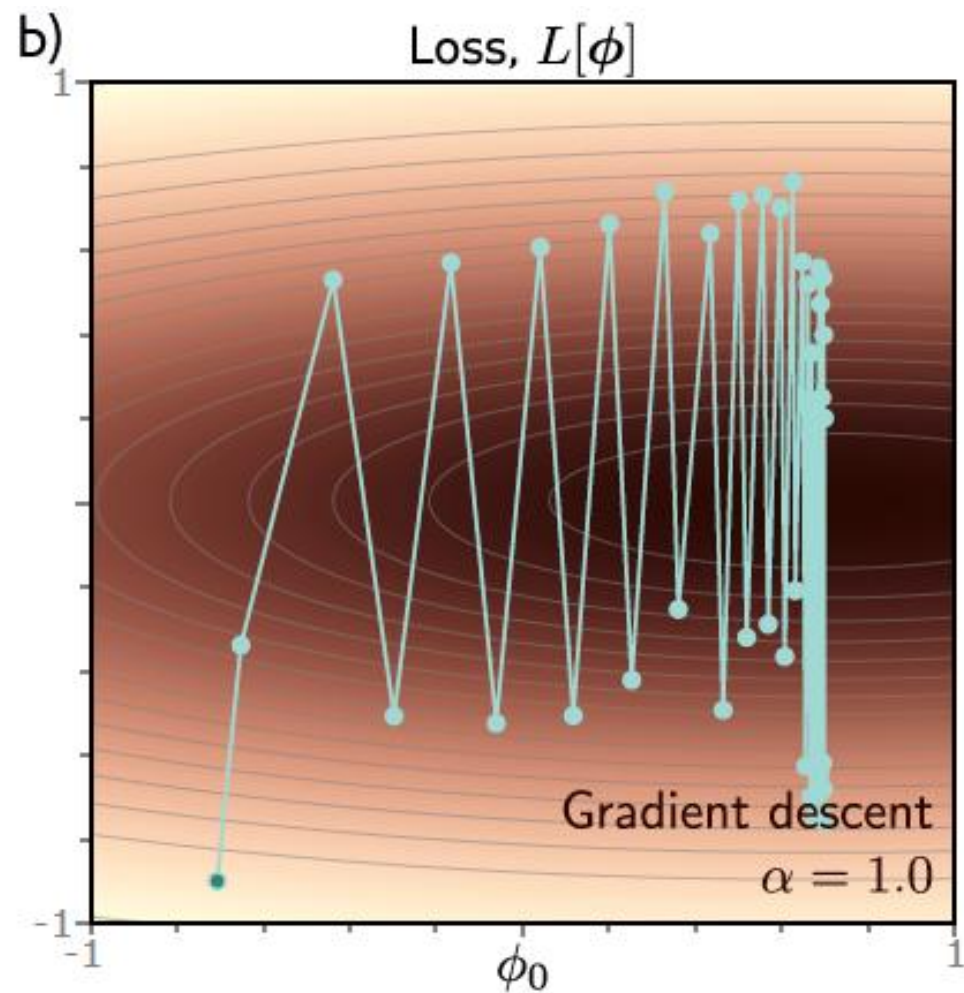
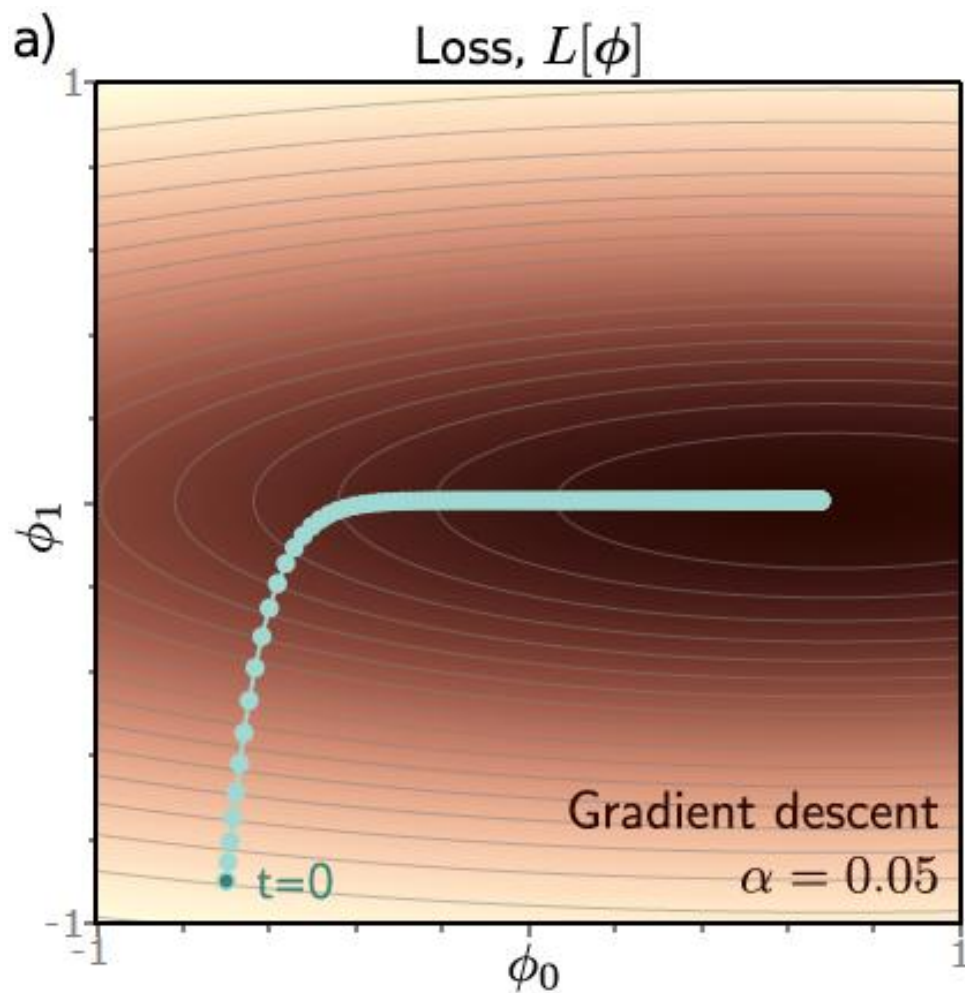
- En realidad, con Nesterov la actualización es entonces:

$$\mathbf{m}_{t+1}^{\text{NAG}} = \mathbf{m}_{t+1}^{\text{std}} - (1 - \beta) \alpha \mathbf{H} \mathbf{m}_t.$$

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

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

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

- 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

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

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

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

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

- Normalizar:

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

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

Gradientes normalizados

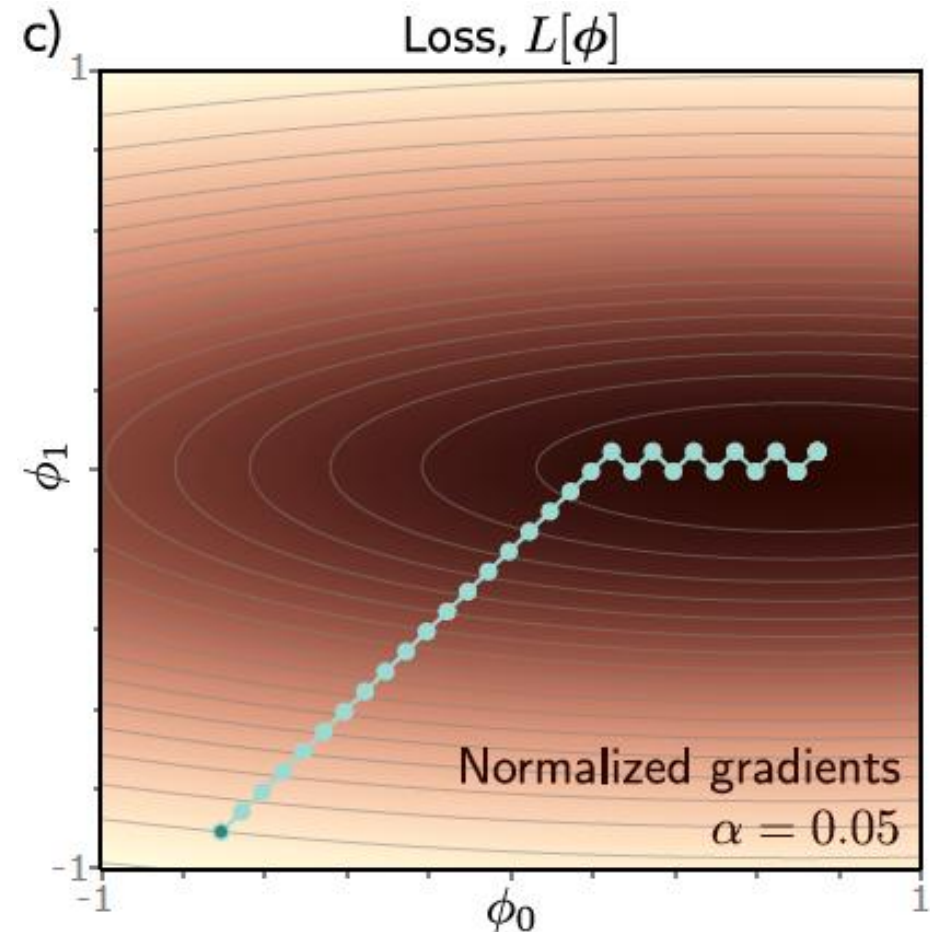
- Medir la media y el gradiente cuadrático puntual

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

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

- 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

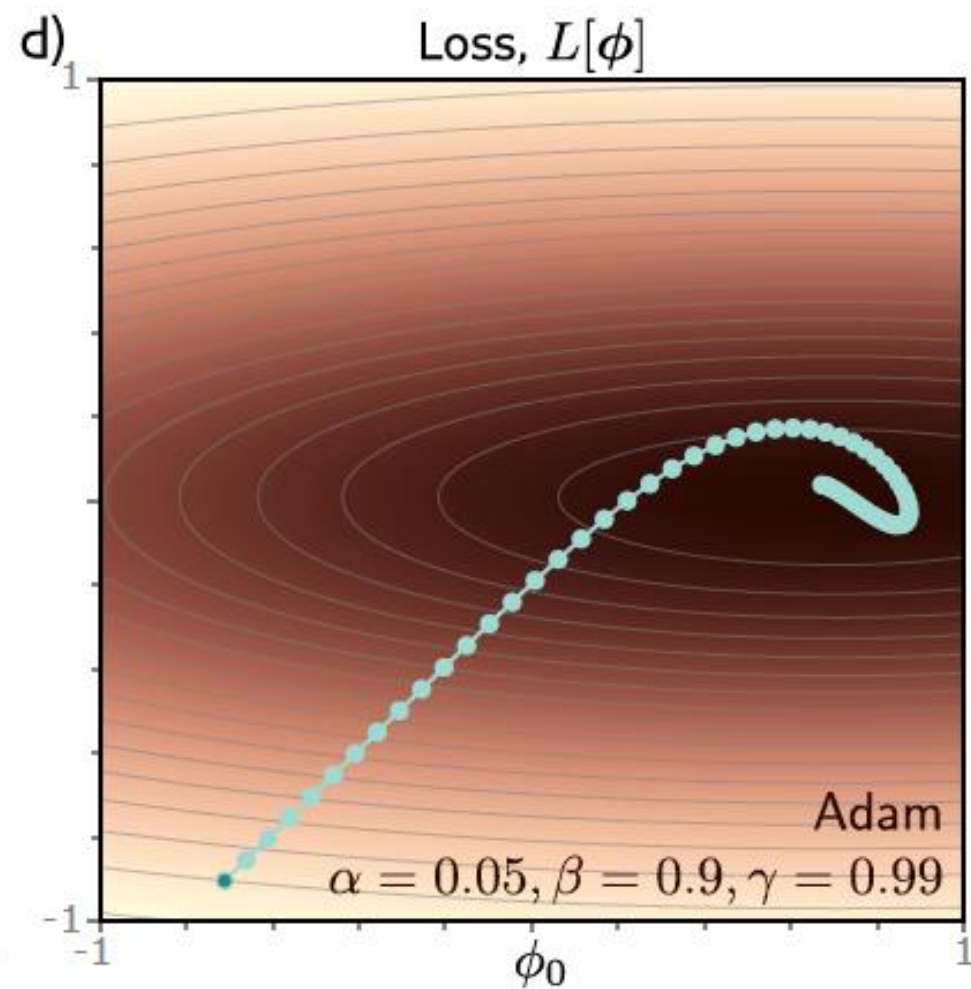
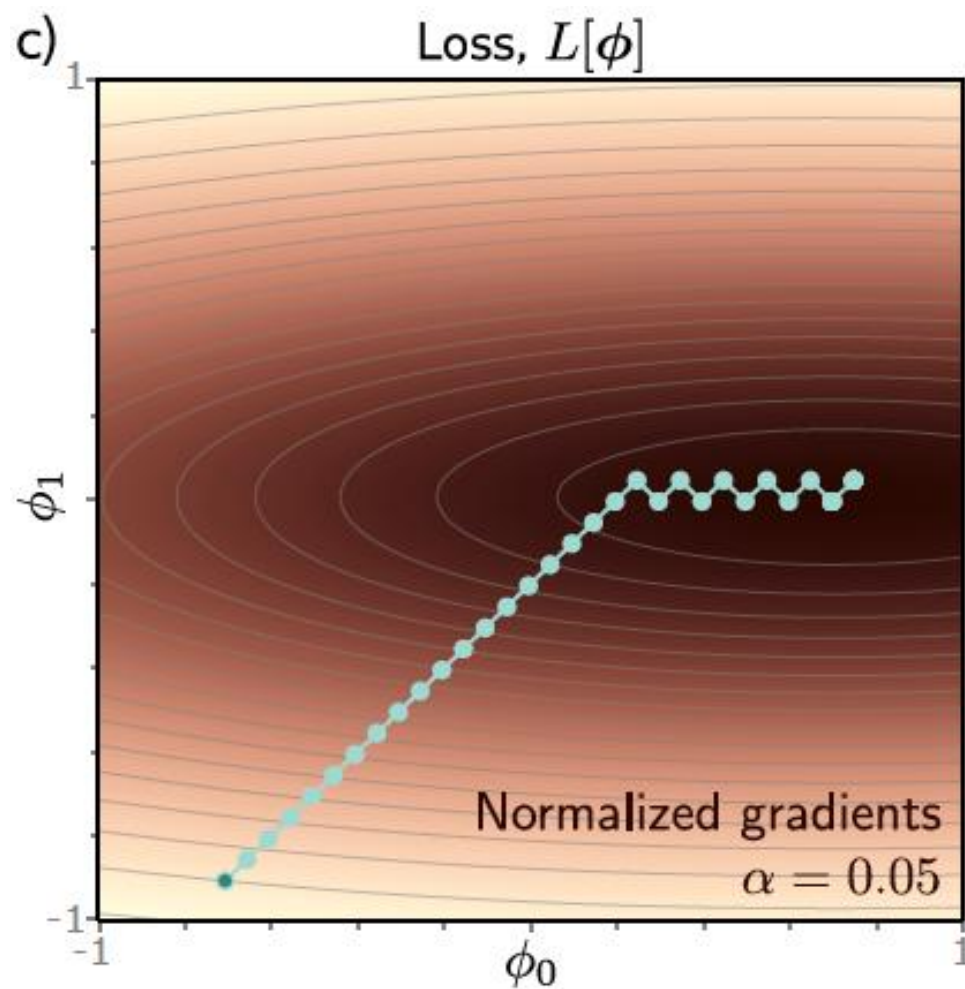
$$\tilde{\mathbf{m}}_{t+1} \leftarrow \frac{\mathbf{m}_{t+1}}{1 - \beta^{t+1}}$$

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

- 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