Tema 2 – Optimización y Regularización (Parte 2)

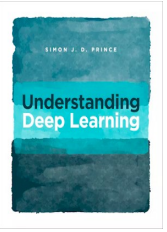
Aprendizaje Automático II - Grado en Inteligencia Artificial
Universidad Rey Juan Carlos

Iván Ramírez Díaz
ivan.ramirez@urjc.es

José Miguel Buenaposada Biencinto
josemiguel.buenaposada@urjc.es

Bibliografía

- Understanding Deep Learning. Capítulo 6.



- Deep Learning: CS 182 2021. Lecture 4.
  Sergey Levine. UC Berkeley.
  Curso en youtube.

Aprendizaje supervisado

- Conjunto de datos de entrenamiento. N pares de muestras entrada/salida:

$$D=\{\boldsymbol{x_i},\boldsymbol{y_i}\}_{i=1}^{N}$$

- Función de coste. Mide cómo de malo es el modelo:

$$J(\boldsymbol{w},\boldsymbol{f}(\cdot;\boldsymbol{w}),\{\boldsymbol{x_i},\boldsymbol{y_i}\}_{i=1}^{N})$$

$$\downarrow$$

$$J(\boldsymbol{w})\in\mathbb{R}$$

Pérdida media

- Función de coste. Mide cómo de malo es el modelo:

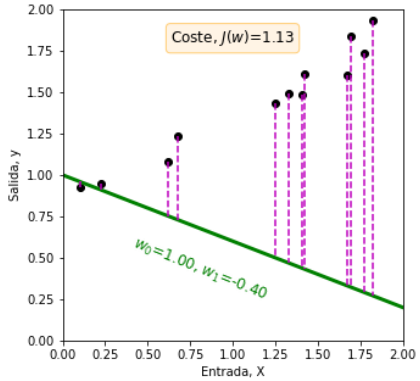
$$J(\boldsymbol{w})=\frac{1}{N}\sum_{i=1}^{N}\underbrace{L(f(\boldsymbol{x_i},\boldsymbol{w}),\boldsymbol{y_i})}_{L_i(\boldsymbol{w})}$$

Entrenamiento. Encontrar los parámetros que minimizan la función de coste:

$$\hat{\boldsymbol{w}}=argmín_{\boldsymbol{w}}\,J(\boldsymbol{w})$$

En general el aprendizaje supervisado minimizará la pérdida media sobre todo el conjunto de entrenamiento D

Ejemplo: regresión lineal 1D



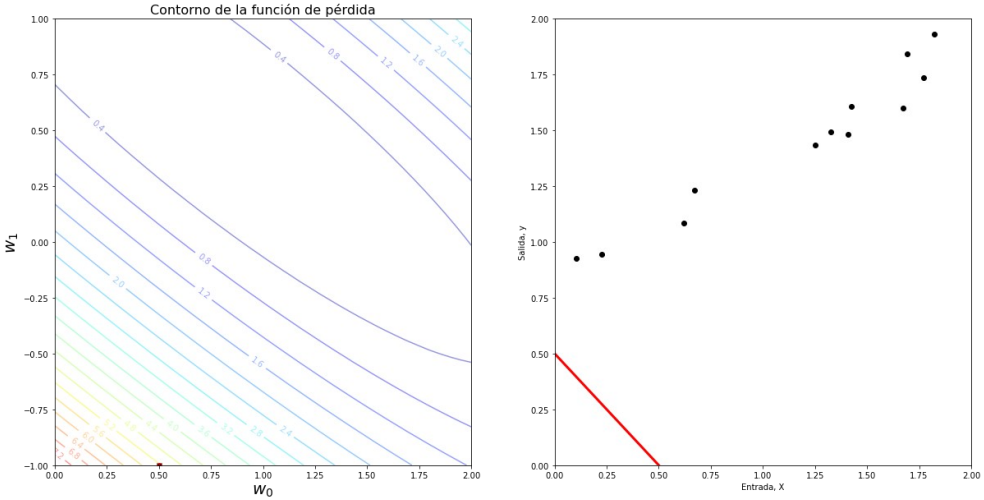
- Función de coste:

$$J(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}\left(f(\boldsymbol{x_i};\boldsymbol{w}) - y_i\right)^2$$
$$= \frac{1}{N}\sum_{i=1}^{N}\left(w_1 x_i + w_0 - y_i\right)^2$$

Función de coste basada en el "error cuadrático medio" ("Mean squared error") o problema de "mínimos cuadrados" ("least squares problem")

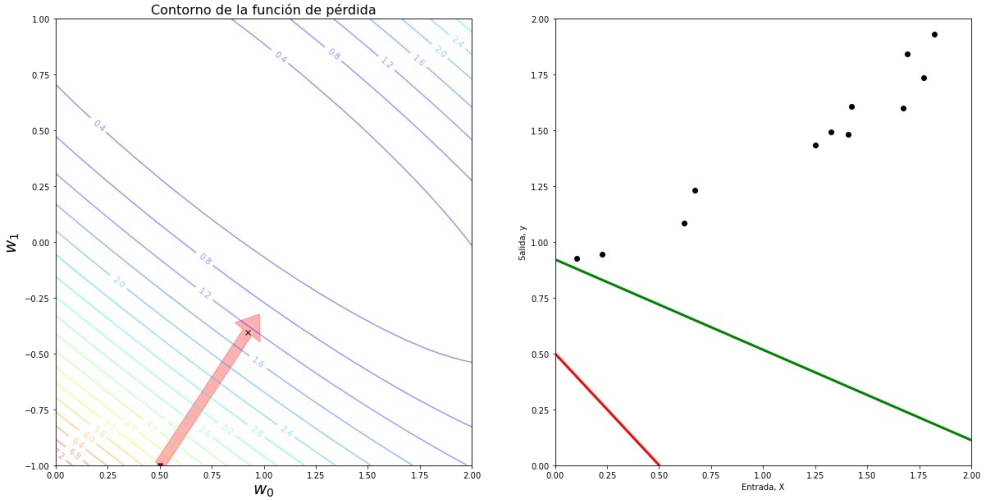
Ejemplo: regresión lineal 1D

Proceso de entrenamiento:



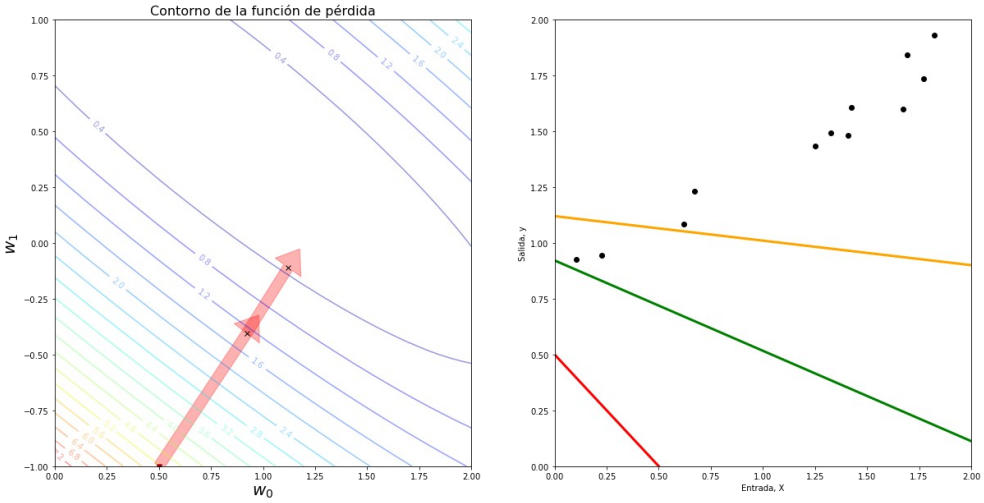
Ejemplo: regresión lineal 1D

Proceso de entrenamiento:



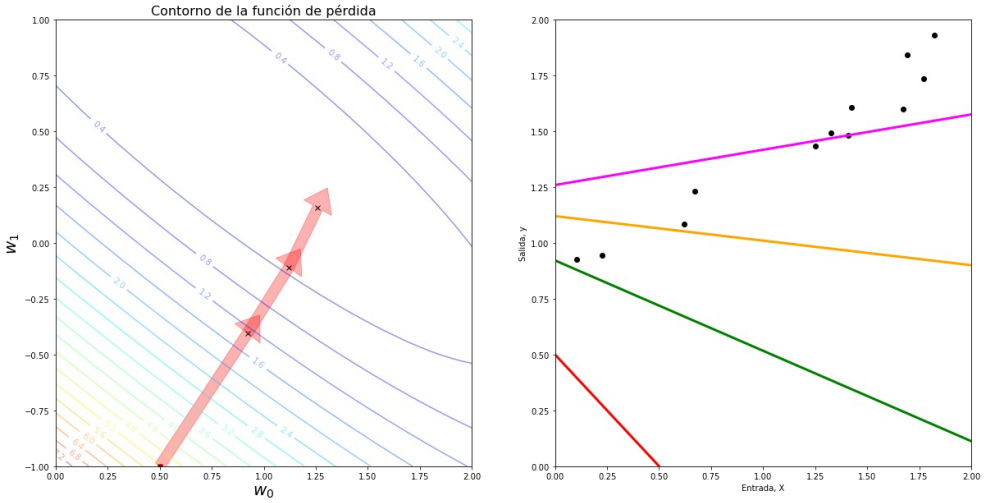
Ejemplo: regresión lineal 1D

Proceso de entrenamiento:



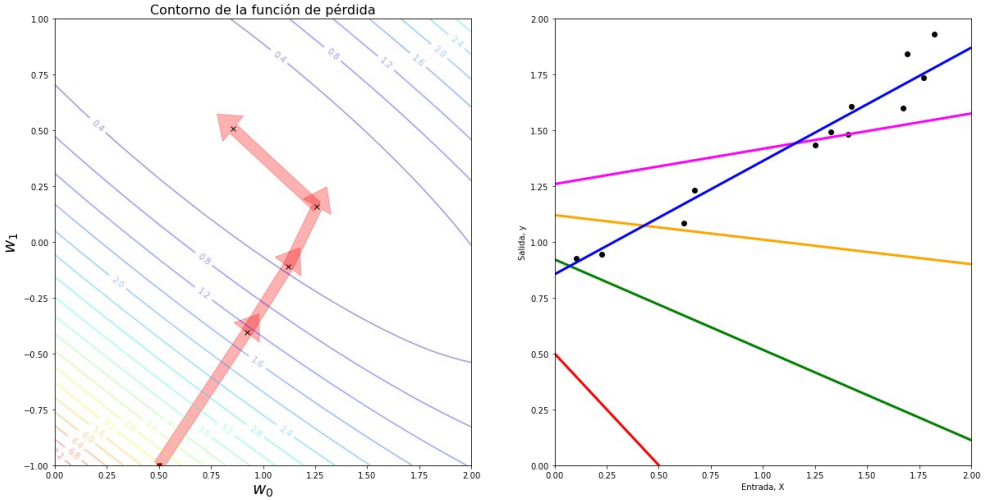
Ejemplo: regresión lineal 1D

Proceso de entrenamiento:



Ejemplo: regresión lineal 1D

Proceso de entrenamiento:



2.3 Entrenamiento de modelos

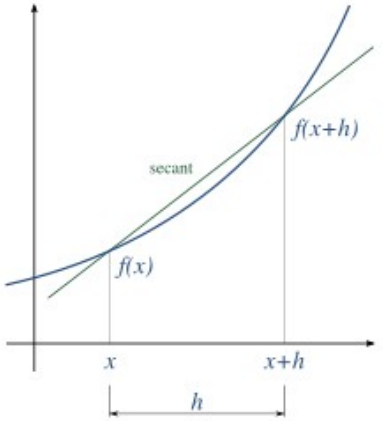
- Conceptos matemáticos
- Algoritmo de descenso de gradiente
- Ejemplo con regresión lineal
- Ejemplo con modelo de Gabor
- Descenso de Gradiente Estocástico (SGD)
- Momento
- Adam

2.3 Entrenamiento de modelos

- Conceptos matemáticos
- Algoritmo de descenso de gradiente
- Ejemplo con regresión lineal
- Ejemplo con modelo de Gabor
- Descenso de Gradiente Estocástico (SGD)
- Momento
- Adam

El concepto de derivada

Función en 1D: $y=f(x)$

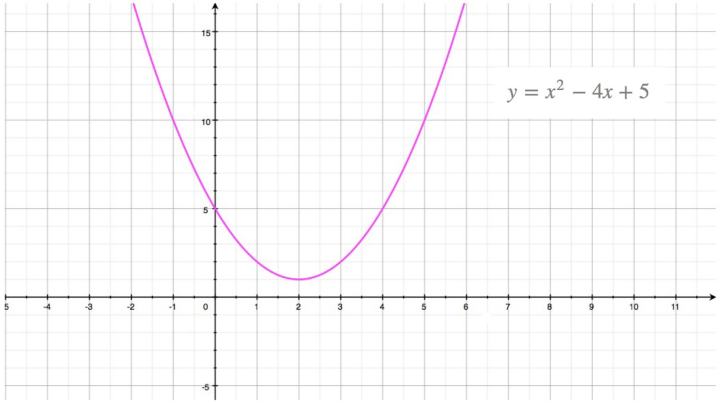


$$\frac{df(x)}{dx}=\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

El concepto de vector gradiente

Función en 1D:

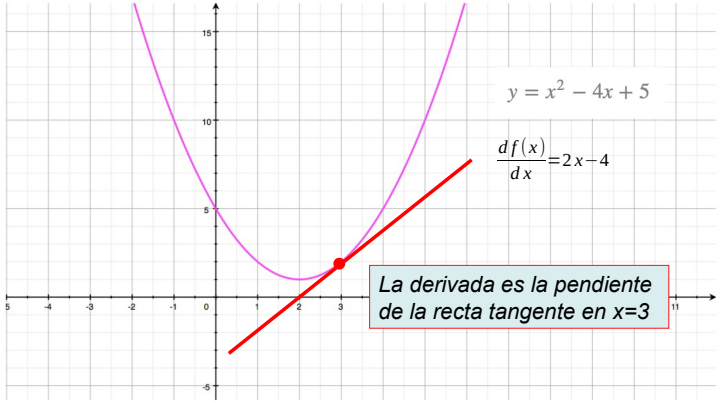
$y=f(x)$



El concepto de vector gradiente

Función en 1D:

$y=f(x)$



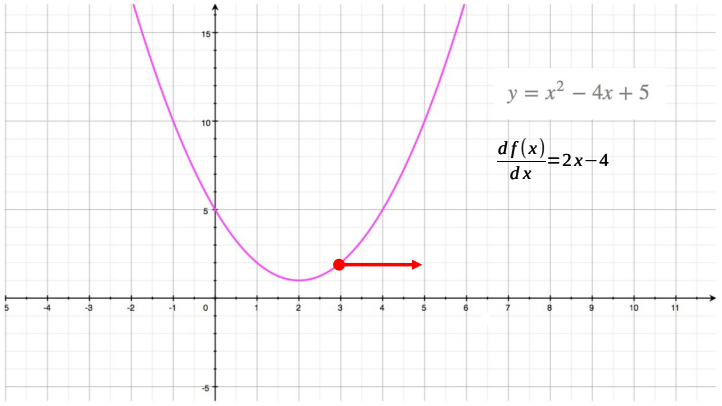
Con x = 3:

$$\left.\frac{df(x)}{dx}\right|_{x=3}=2\cdot 3-4=2$$

El concepto de vector gradiente

Función en 1D:

$y=f(x)$



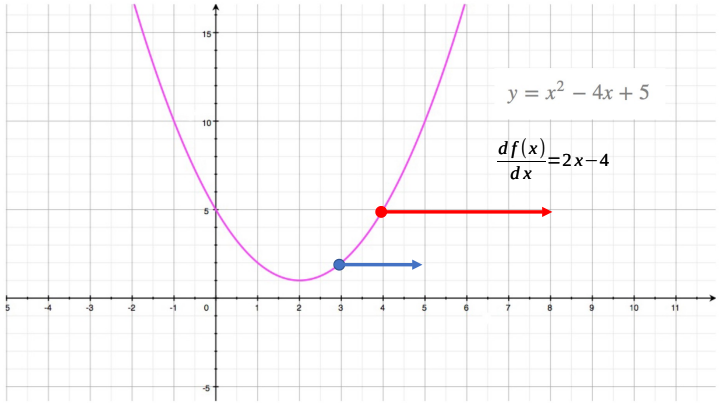
Con x = 3:

$\left.\frac{df(x)}{dx}\right|_{x=3}=2\cdot3-4=2$

El concepto de vector gradiente

Función en 1D:

$y=f(x)$



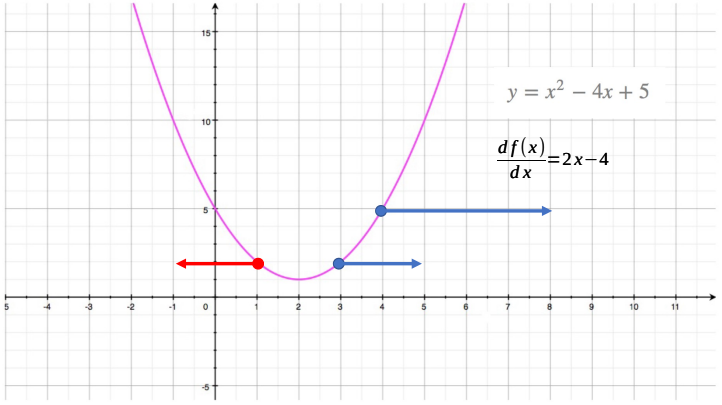
Con x = 4:

$\left.\frac{df(x)}{dx}\right|_{x=4}=2\cdot4-4=4$

El concepto de vector gradiente

Función en 1D:

$y=f(x)$



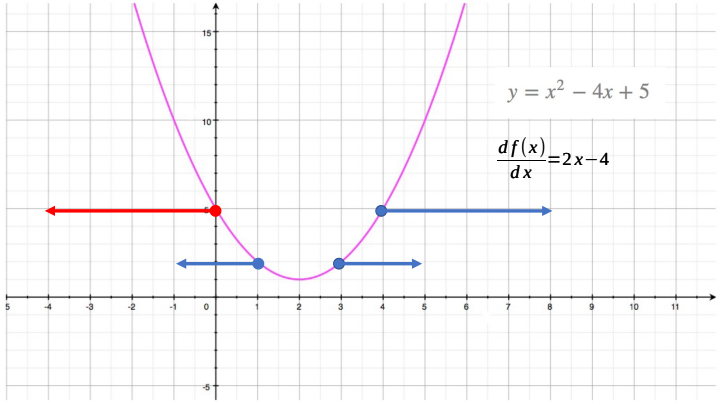
Con x = 1:

$\left.\frac{df(x)}{dx}\right|_{x=1}=2\cdot1-4=-2$

El concepto de vector gradiente

Función en 1D:

$y=f(x)$

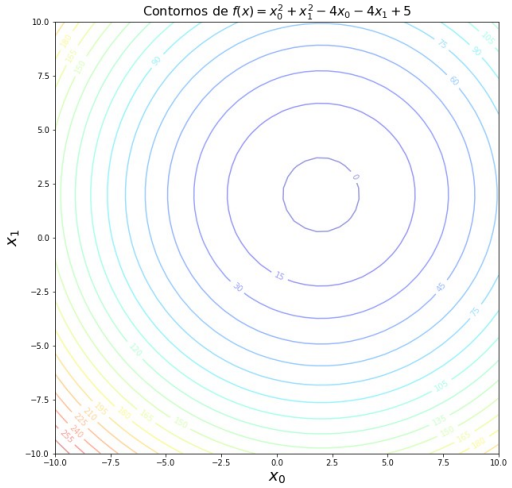


Con x = 0:

$\left.\frac{df(x)}{dx}\right|_{x=0}=2\cdot0-4=-4$

El concepto de vector gradiente

Función en 2D:

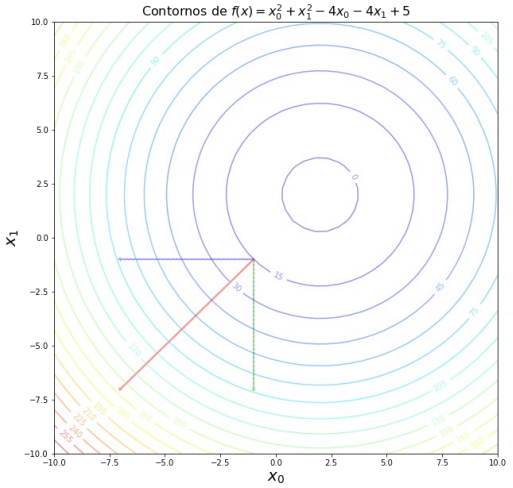


$$f(\boldsymbol{x}) = x_0^2 + x_1^2 - 4x_0 - 4x_1 + 5$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_0} = 2x_0 - 4 \qquad \frac{\partial f(\boldsymbol{x})}{\partial x_1} = 2x_1 - 4$$

El concepto de vector gradiente

Función en 2D:



$$f(\boldsymbol{x}) = x_0^2 + x_1^2 - 4x_0 - 4x_1 + 5$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_0} = 2x_0 - 4 \qquad \frac{\partial f(\boldsymbol{x})}{\partial x_1} = 2x_1 - 4$$

Con x = (-1, -1)^T:

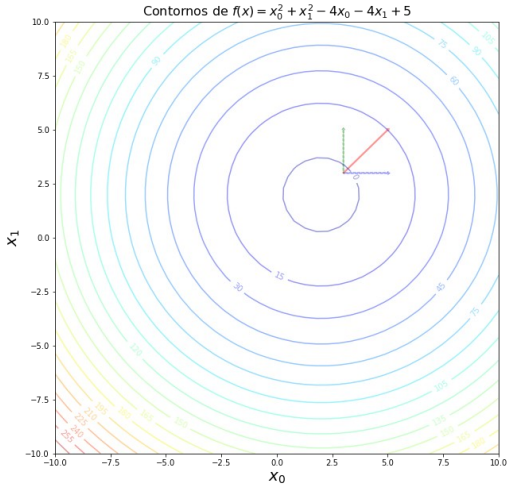
$$\left.\frac{\partial f(\boldsymbol{x})}{\partial x_0}\right|_{x=-(1,-1)^T} = 2\cdot(-1) - 4 = -6$$

$$\left.\frac{\partial f(\boldsymbol{x})}{\partial x_1}\right|_{x=-(1,-1)^T} = 2\cdot(-1) - 4 = -6$$

$$\left.\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=(-1,-1)^T} = \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_0} \\ \frac{\partial f(\boldsymbol{x})}{\partial x_1} \end{bmatrix} = \begin{bmatrix} -6 \\ -6 \end{bmatrix}$$

El concepto de vector gradiente

Función en 2D:



$$f(\boldsymbol{x}) = x_0^2 + x_1^2 - 4x_0 - 4x_1 + 5$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_0} = 2x_0 - 4 \qquad \frac{\partial f(\boldsymbol{x})}{\partial x_1} = 2x_1 - 4$$

Con x = (3, 3)^T:

$$\left.\frac{\partial f(\boldsymbol{x})}{\partial x_0}\right|_{x=-(1,-1)^T} = 2\cdot 3 - 4 = 2$$

$$\left.\frac{\partial f(\boldsymbol{x})}{\partial x_1}\right|_{x=-(1,-1)^T} = 2\cdot 2 - 4 = 2$$

$$\left.\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{x=(3,3)^T} = \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_0} \\ \frac{\partial f(\boldsymbol{x})}{\partial x_1} \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

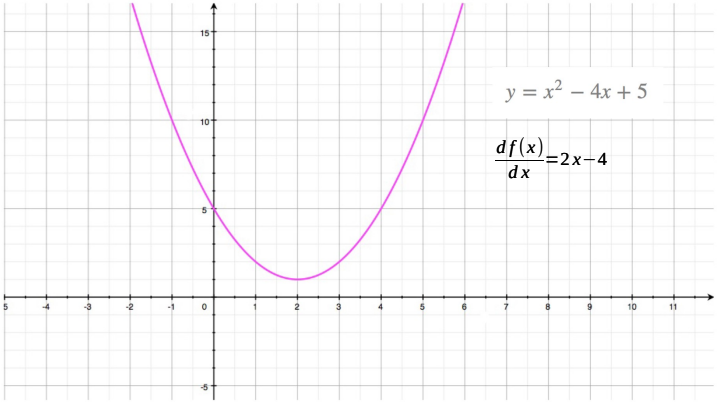
Aproximación numérica al gradiente

Función en 1D:

$$y = f(x)$$

Definición de derivada:

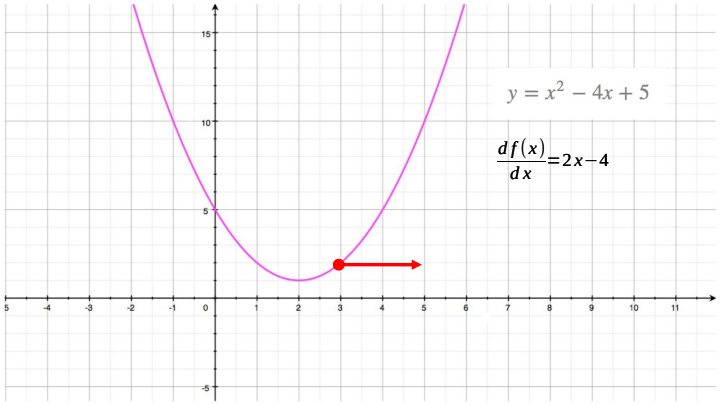
$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h}$$



Aproximación numérica al gradiente

Función en 1D: $y=f(x)$

Definición de derivada: $\frac{df(x)}{dx}=\lim_{h\to 0}\frac{f(x+h)-f(x-h)}{2h}$



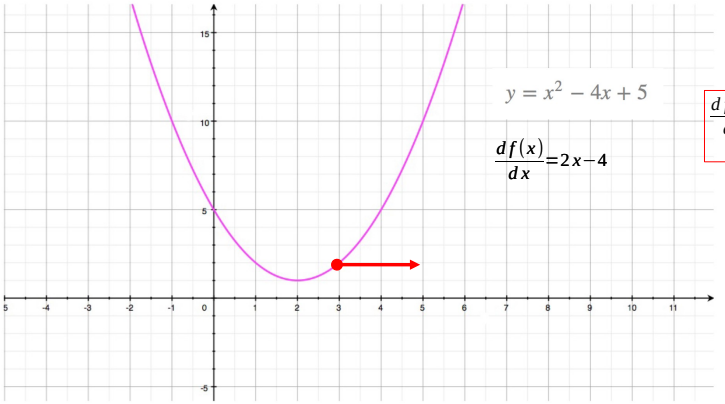
Con x = 3:

$\left.\frac{df(x)}{dx}\right|_{x=3}=2\cdot 3-4=2$

Aproximación numérica al gradiente

Función en 1D: $y=f(x)$

Definición de derivada: $\frac{df(x)}{dx}=\lim_{h\to 0}\frac{f(x+h)-f(x-h)}{2h}$



Con x = 3; h=0,001:

$\left.\frac{df(x)}{dx}\right|_{x=3}=2\cdot 3-4=2$

$\left.\frac{df(x)}{dx}\right|_{x=3}\approx\frac{f(3{,}001)-f(2{,}999)}{2\cdot 0{,}001}=1{,}99999999999978$

Aproximación numérica a la derivada

2.3 Entrenamiento de modelos

- Conceptos matemáticos
- Algoritmo de descenso de gradiente
- Ejemplo con regresión lineal
- Ejemplo con modelo de Gabor
- Descenso de Gradiente Estocástico (SGD)
- Momento
- Adam

Algoritmo de descenso de gradiente

- Paso 0. Inicializar los parámetros $\boldsymbol{\Phi}_0$.
- Repetir:
  - Paso 1. Calcular derivadas de **la función de coste** con respecto a los parámetros $\boldsymbol{\Phi}$

$$\left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}=\begin{bmatrix}\frac{\partial J(\boldsymbol{\Phi})}{\partial \Phi_0}\\ \frac{\partial J(\boldsymbol{\Phi})}{\partial \Phi_1}\\ \vdots\\ \frac{\partial J(\boldsymbol{\Phi})}{\partial \Phi_k}\end{bmatrix}$$

  - Paso 2. Actualizar los parámetros de acuerdo con:

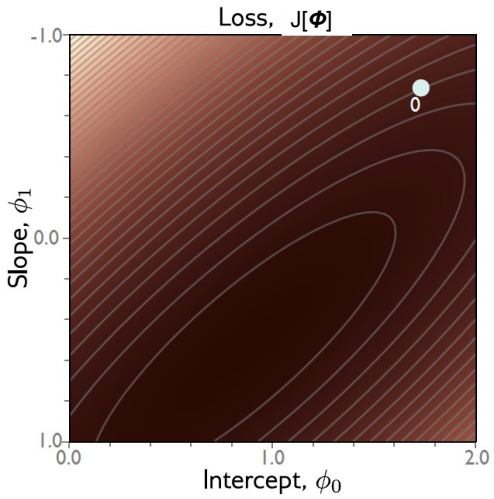
$$\boldsymbol{\Phi}_{t+1}\leftarrow\boldsymbol{\Phi}_t-\alpha\left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

donde el escalar positivo α determina la magnitud del cambio.

2.3 Entrenamiento de modelos

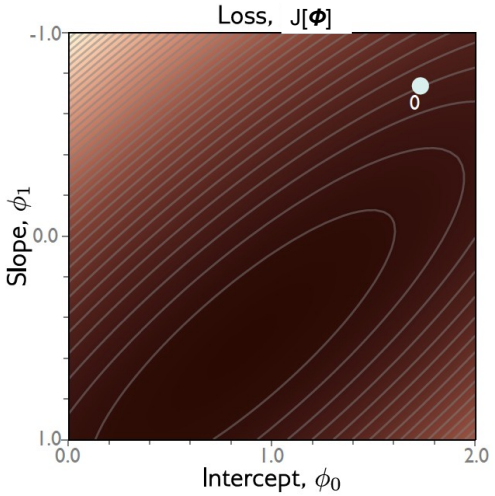
- Conceptos matemáticos
- Algoritmo de descenso de gradiente
- Ejemplo con regresión lineal
- Ejemplo con modelo de Gabor
- Descenso de Gradiente Estocástico (SGD)
- Momento
- Adam

Descenso de gradiente



- Paso 0. Inicializar los parámetros $\boldsymbol{\Phi}_0$.

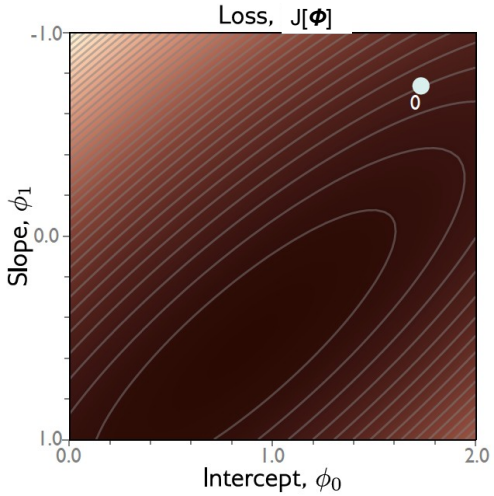
Descenso de gradiente



- Paso 1. Calcular derivadas de la función de coste con respecto a los parámetros

$$J(\boldsymbol{\Phi}) = \sum_{i=1}^{N} \ell_i = \sum_{i=1}^{N} (\mathrm{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$
$$= \sum_{i=1}^{N} (\phi_0 + \phi_1 x_i - y_i)^2$$

Descenso de gradiente

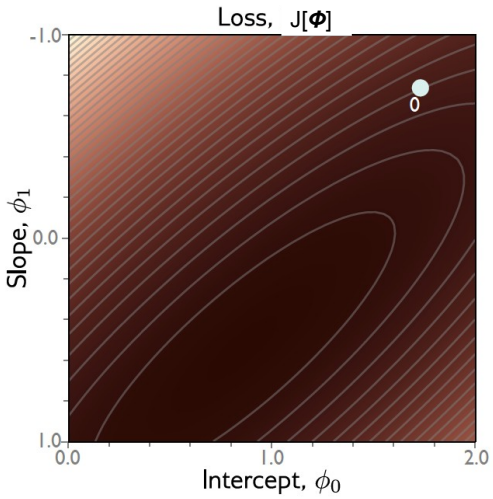


- Paso 1. Calcular derivadas de la función de coste con respecto a los parámetros

$$J(\boldsymbol{\Phi}) = \sum_{i=1}^{N} \ell_i = \sum_{i=1}^{N} (\mathrm{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$
$$= \sum_{i=1}^{N} (\phi_0 + \phi_1 x_i - y_i)^2$$

$$\frac{\partial J}{\partial \boldsymbol{\phi}} = \frac{\partial}{\partial \boldsymbol{\phi}} \sum_{i=1}^{N} \ell_i = \sum_{i=1}^{N} \frac{\partial \ell_i}{\partial \boldsymbol{\phi}}$$

Descenso de gradiente



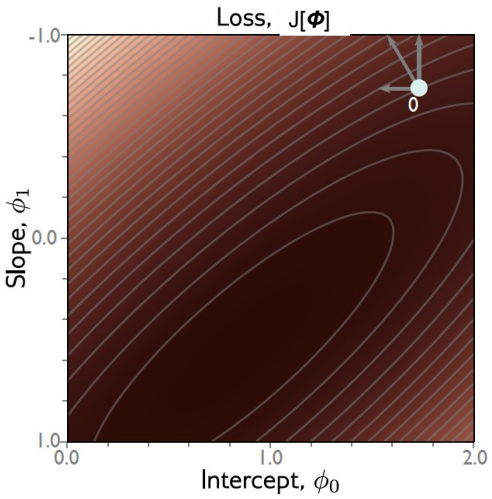
- Paso 1. Calcular derivadas de la función de coste con respecto a los parámetros

$$J(\boldsymbol{\Phi}) = \sum_{i=1}^{N} \ell_i = \sum_{i=1}^{N} (\mathrm{f}[x_i, \boldsymbol{\phi}] - y_i)^2$$
$$= \sum_{i=1}^{N} (\phi_0 + \phi_1 x_i - y_i)^2$$

$$\frac{\partial J}{\partial \boldsymbol{\phi}} = \frac{\partial}{\partial \boldsymbol{\phi}} \sum_{i=1}^{N} \ell_i = \sum_{i=1}^{N} \frac{\partial \ell_i}{\partial \boldsymbol{\phi}}$$

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

Descenso de gradiente

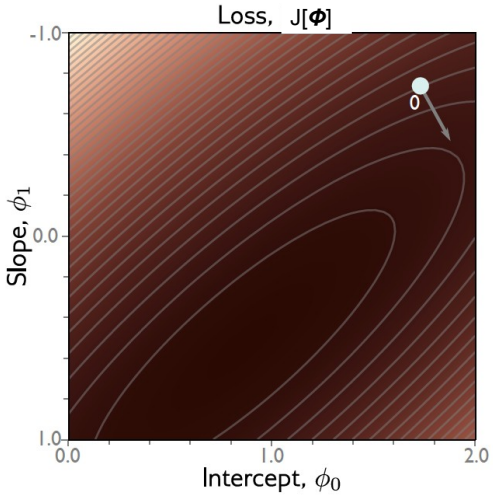


- Paso 1. Calcular derivadas de la función de coste con respecto a los parámetros

$$\frac{\partial J}{\partial \boldsymbol{\phi}} = \frac{\partial}{\partial \boldsymbol{\phi}} \sum_{i=1}^{N} \ell_i = \sum_{i=1}^{N} \frac{\partial \ell_i}{\partial \boldsymbol{\phi}}$$

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

Descenso de gradiente



- Paso 1. Calcular derivadas de la función de coste con respecto a los parámetros

$$\frac{\partial J}{\partial \boldsymbol{\phi}} = \frac{\partial}{\partial \boldsymbol{\phi}} \sum_{i=1}^{N} \ell_i = \sum_{i=1}^{N} \frac{\partial \ell_i}{\partial \boldsymbol{\phi}}$$

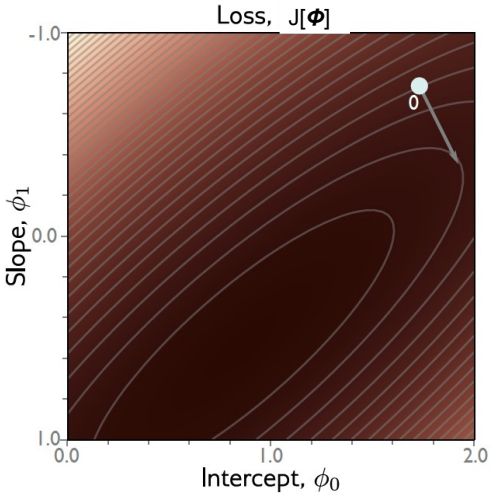
$$\frac{\partial \ell_i}{\partial \boldsymbol{\phi}} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

- Paso 2. Actualizar los parámetros de acuerdo con:

$$\boldsymbol{\Phi}_{t+1} \leftarrow \boldsymbol{\Phi}_t - \alpha \frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\bigg|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

α = tamaño del paso (ó learning rate si tiene valor fijo)

Descenso de gradiente



- Paso 1. Calcular derivadas de la función de coste con respecto a los parámetros

$$\frac{\partial J}{\partial \boldsymbol{\phi}} = \frac{\partial}{\partial \boldsymbol{\phi}} \sum_{i=1}^{N} \ell_i = \sum_{i=1}^{N} \frac{\partial \ell_i}{\partial \boldsymbol{\phi}}$$

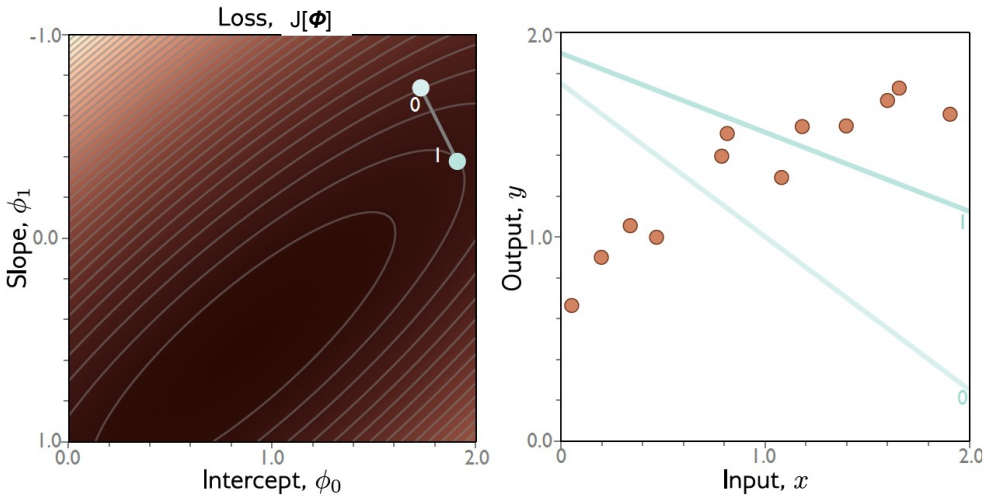
$$\frac{\partial \ell_i}{\partial \boldsymbol{\phi}} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

- Paso 2. Actualizar los parámetros de acuerdo con:

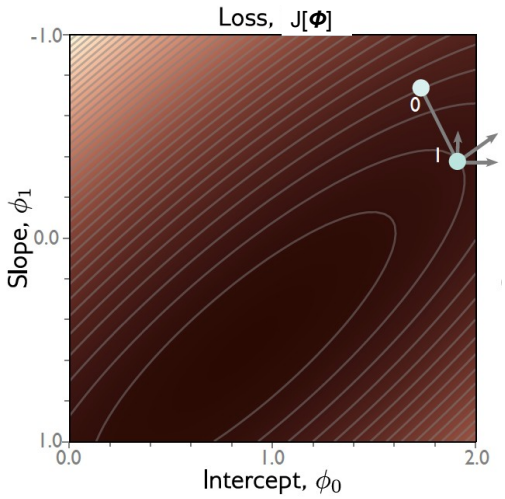
$$\boldsymbol{\Phi}_{t+1} \leftarrow \boldsymbol{\Phi}_t - \alpha \frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\bigg|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

α = tamaño del paso

Descenso de gradiente



Descenso de gradiente



- Paso 1. Calcular derivadas de la función de coste con respecto a los parámetros

$$\frac{\partial J}{\partial \boldsymbol{\phi}} = \frac{\partial}{\partial \boldsymbol{\phi}} \sum_{i=1}^{N} \ell_i = \sum_{i=1}^{N} \frac{\partial \ell_i}{\partial \boldsymbol{\phi}}$$

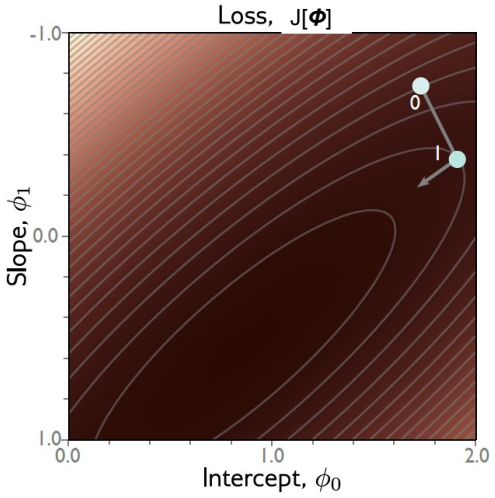
$$\frac{\partial \ell_i}{\partial \boldsymbol{\phi}} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

- Paso 2. Actualizar los parámetros de acuerdo con:

$$\boldsymbol{\Phi}_{t+1} \leftarrow \boldsymbol{\Phi}_t - \alpha \frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\bigg|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

α = tamaño del paso

Descenso de gradiente



- Paso 1. Calcular derivadas de la función de coste con respecto a los parámetros

$$\frac{\partial J}{\partial \boldsymbol{\phi}} = \frac{\partial}{\partial \boldsymbol{\phi}} \sum_{i=1}^{N} \ell_i = \sum_{i=1}^{N} \frac{\partial \ell_i}{\partial \boldsymbol{\phi}}$$

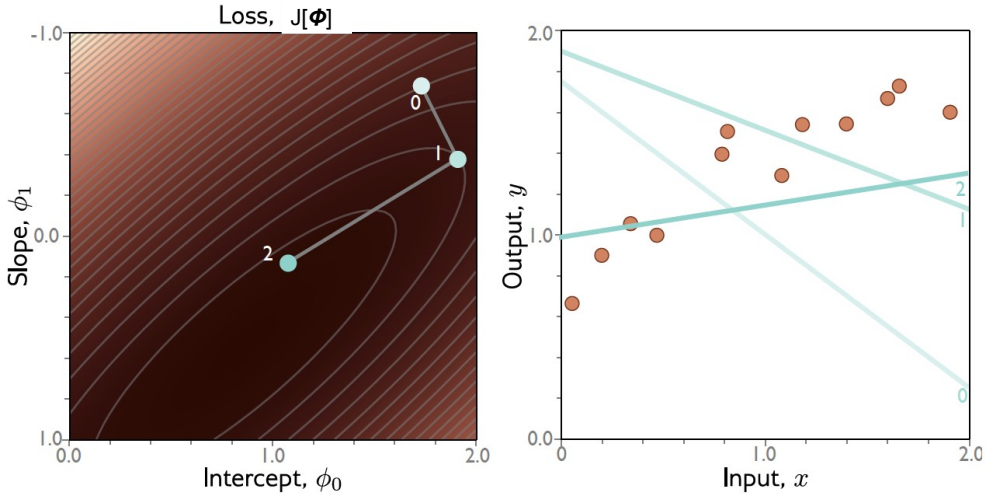
$$\frac{\partial \ell_i}{\partial \boldsymbol{\phi}} = \begin{bmatrix} \frac{\partial \ell_i}{\partial \phi_0} \\ \frac{\partial \ell_i}{\partial \phi_1} \end{bmatrix} = \begin{bmatrix} 2(\phi_0 + \phi_1 x_i - y_i) \\ 2x_i(\phi_0 + \phi_1 x_i - y_i) \end{bmatrix}$$

- Paso 2. Actualizar los parámetros de acuerdo con:

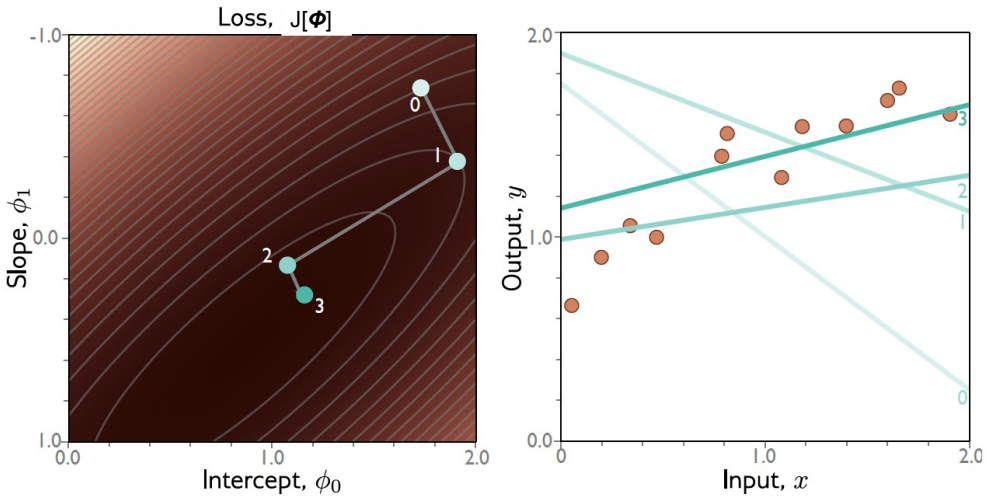
$$\boldsymbol{\Phi}_{t+1} \leftarrow \boldsymbol{\Phi}_t - \alpha \frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\bigg|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

α = tamaño del paso

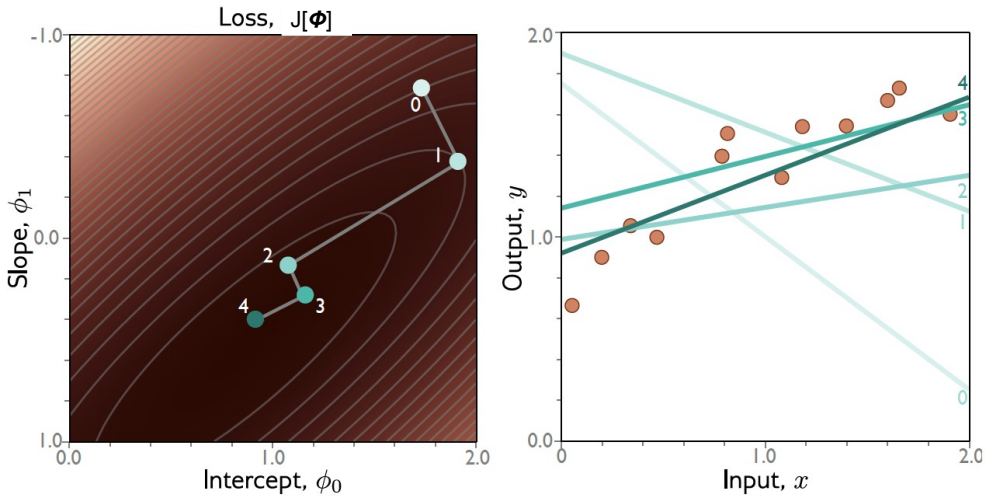
Descenso de gradiente



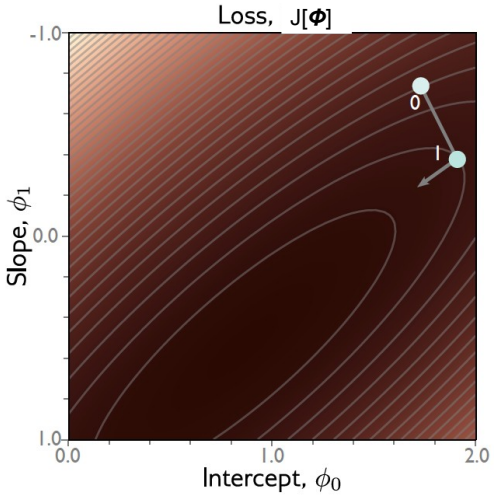
Descenso de gradiente



Descenso de gradiente



Cálculo del paso de descenso (*line search*)



- Paso 1. Calcular derivadas de la función de coste con respecto a los parámetros

$$\frac{\partial J}{\partial \boldsymbol{\phi}} = \frac{\partial}{\partial \boldsymbol{\phi}} \sum_{i=1}^{N} \ell_i = \sum_{i=1}^{N} \frac{\partial \ell_i}{\partial \boldsymbol{\phi}}$$

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

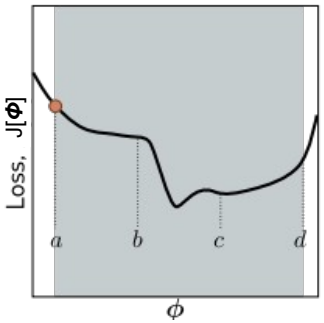
- Paso 2. Actualizar los parámetros de acuerdo con:

$$\boldsymbol{\Phi}_{t+1} \leftarrow \boldsymbol{\Phi}_t - \alpha \frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\bigg|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

α = tamaño del paso

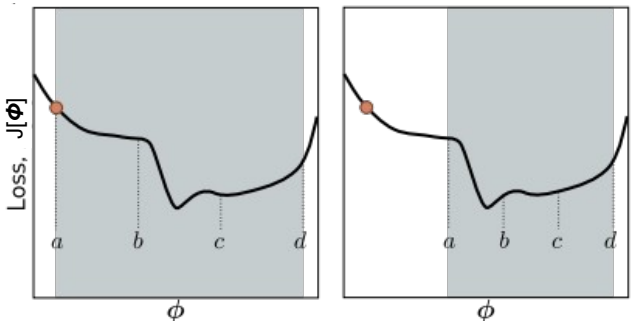
Cálculo del paso de descenso (*line search*)

Algoritmo de acotación (bracketing):



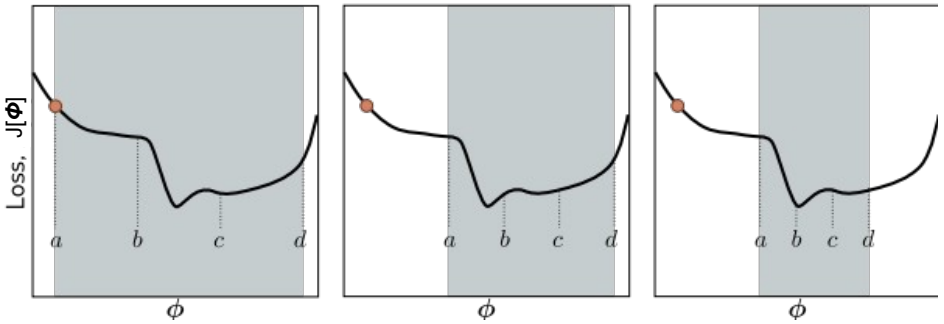
Cálculo del paso de descenso (*line search*)

Algoritmo de acotación (bracketing):

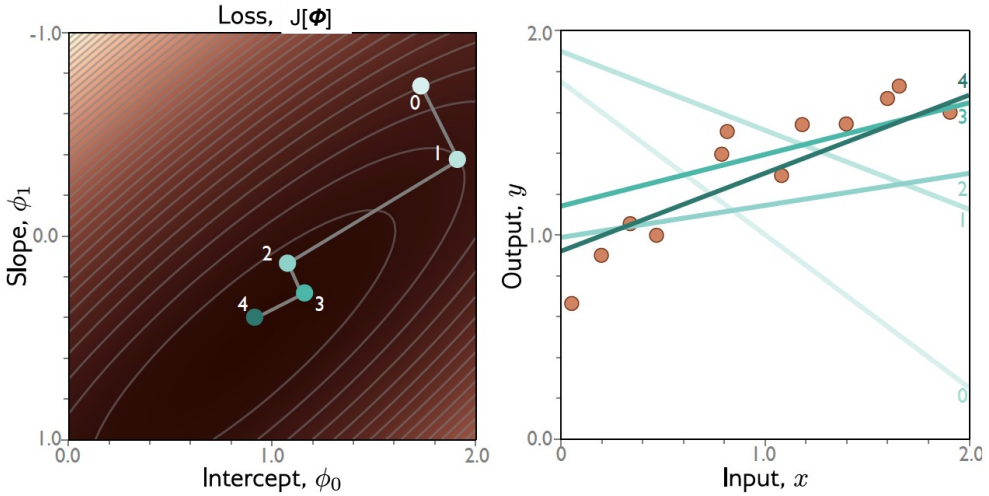


Cálculo del paso de descenso (*line search*)

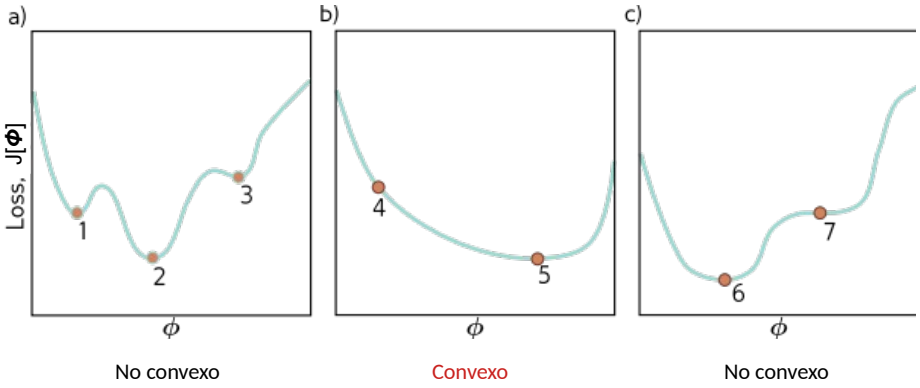
Algoritmo de acotación (bracketing):



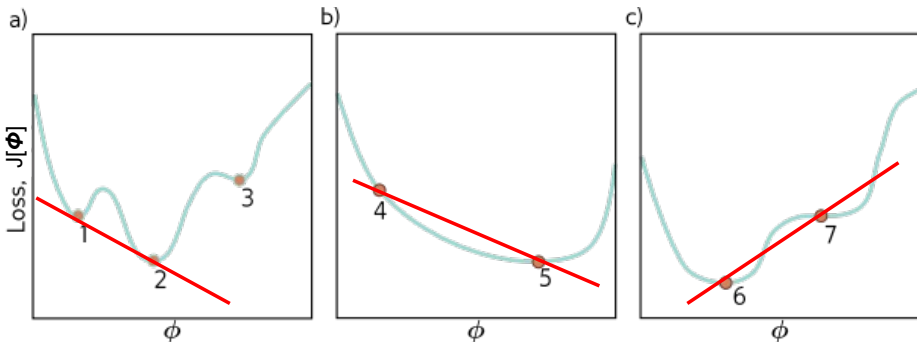
Descenso de gradiente



Problemas de con func. de coste convexa

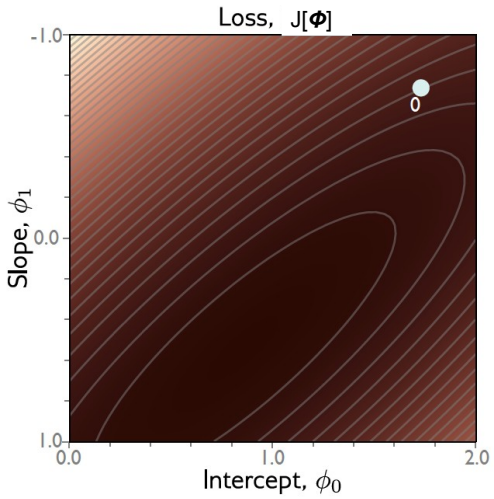


Problemas de con func. de coste convexa



No convexo | Convexo | No convexo

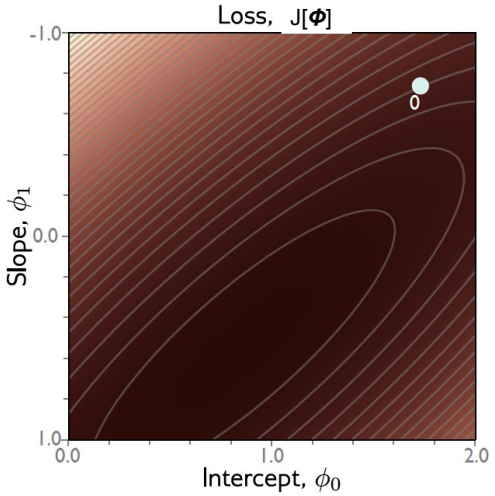
Cálculo del paso de descenso (*line search*)



- Hessiano (matriz de la derivada 2ª):

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

Cálculo del paso de descenso (*line search*)



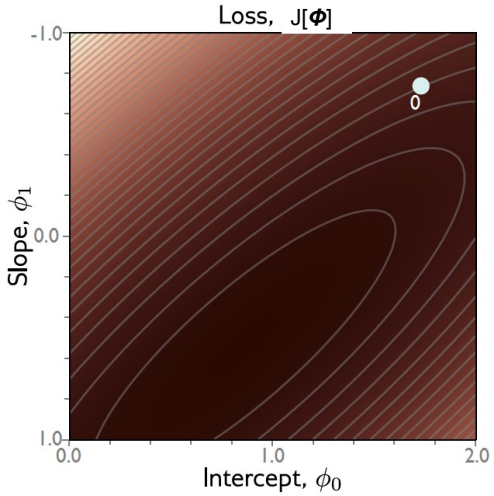
- Hessiano (matriz de la derivada 2ª):

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

- El test de convexidad en 2D es que el determinante del Hessiano es positivo para todo punto $\boldsymbol{\Phi}$:

$$\left|\mathbf{H}[\phi]\right| = \frac{\partial^2 J}{\partial \phi_0^2}\frac{\partial^2 J}{\partial \phi_1^2} - \frac{\partial^2 J}{\partial \phi_0 \partial \phi_1}\frac{\partial^2 J}{\partial \phi_1 \partial \phi_0}$$

Cálculo del paso de descenso (*line search*)



- Hessiano (matriz de la derivada 2ª):

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

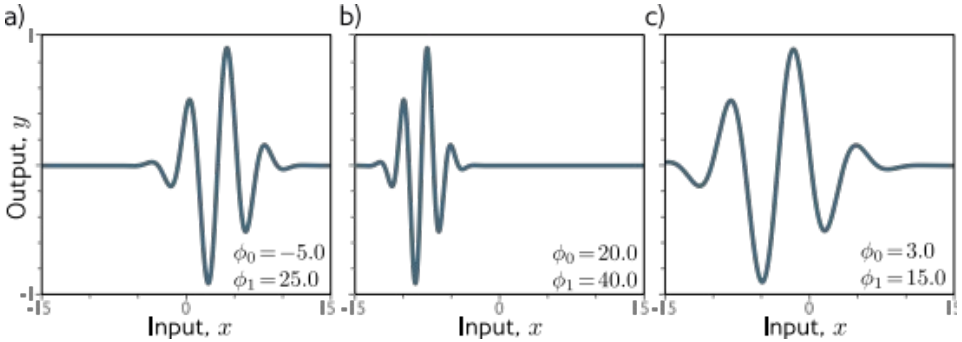
- En un punto, $\boldsymbol{\Phi}$, donde el gradiente es 0 en todas direcciones:
  - Máximo. Si todos los autovalores de H[$\boldsymbol{\Phi}$] son positivos
  - Mínimo. Si todos los autovalores de H[$\boldsymbol{\Phi}$] son negativos
  - Punto de silla (saddle point). Si los autovalores tienen distinto signo.

2.3 Entrenamiento de modelos

- Conceptos matemáticos
- Algoritmo de descenso de gradiente
- Ejemplo con regresión lineal
- Ejemplo con modelo de Gabor
- Descenso de Gradiente Estocástico (SGD)
- Momento
- Adam

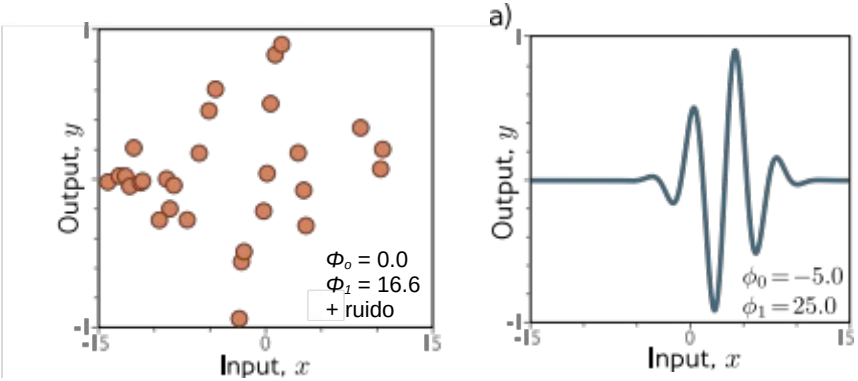
Modelo de Gabor

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

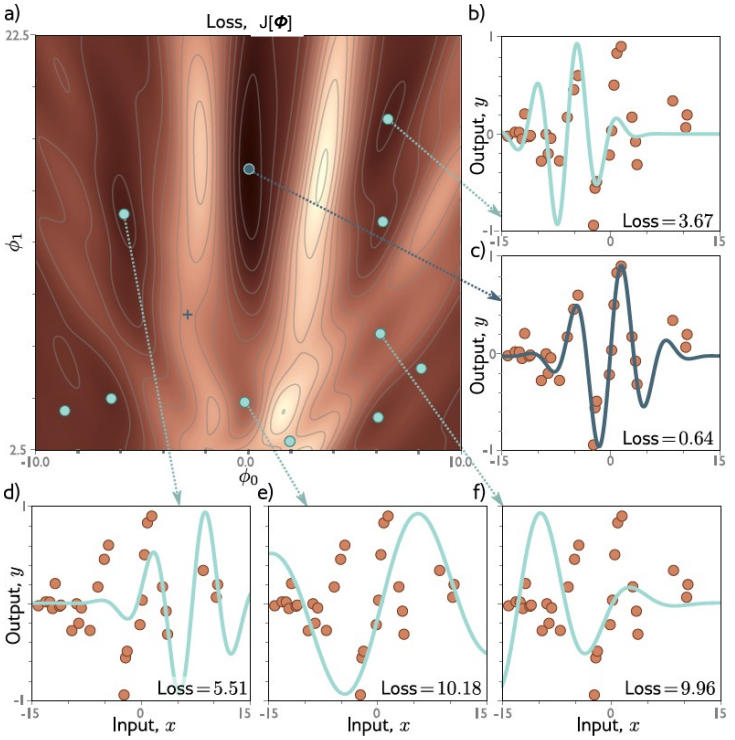


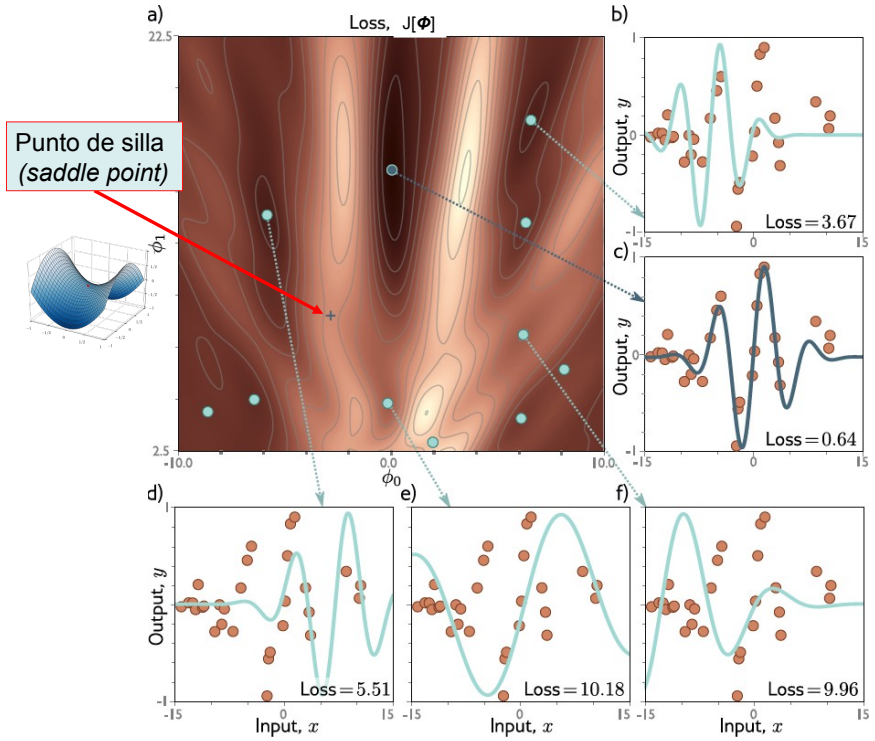
Modelo de Gabor + Error Cuadrático

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

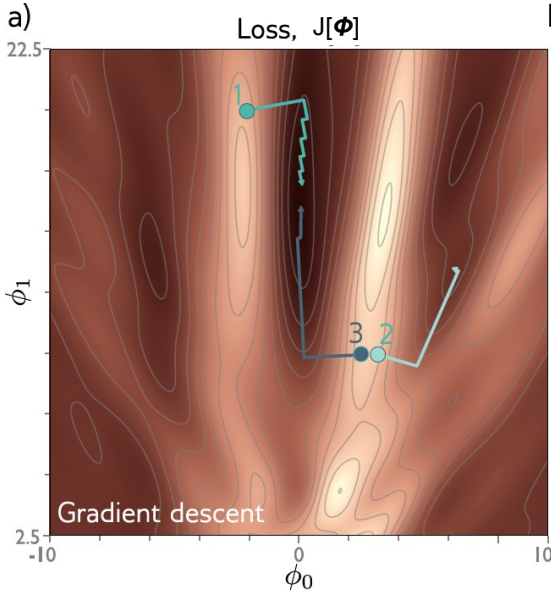


$$J(\boldsymbol{w}) = \sum_{i=1}^{N} \left(f(\boldsymbol{x_i}; \boldsymbol{\Phi}) - y_i\right)^2$$

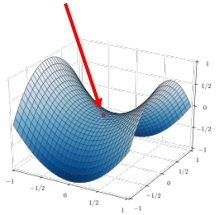




Decenso de gradiente



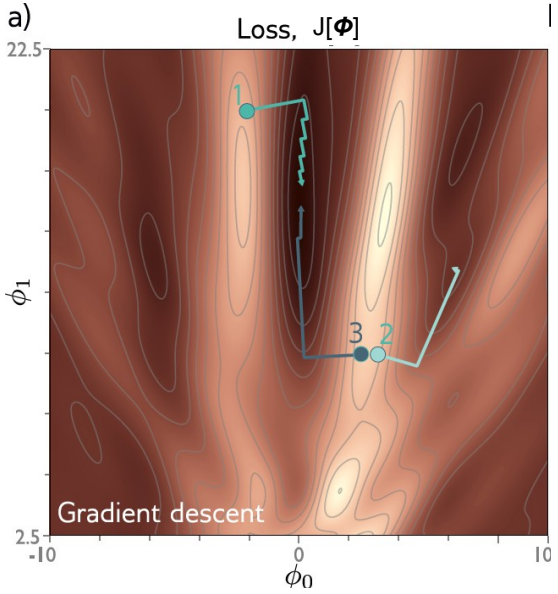
- El descenso de gradiente llega al mínimo global si comenzamos en el "valle" correcto.
- En otro caso, desciende hasta un mínimo local
- O se queda atrapado en un punto de silla (*saddle point*).



2.3 Entrenamiento de modelos

- Conceptos matemáticos
- Algoritmo de descenso de gradiente
- Ejemplo con regresión lineal
- Ejemplo con modelo de Gabor
- Descenso de Gradiente Estocástico (SGD)
- Momento
- Adam

Decenso de Gradiente Estocástico (SGD)



- Hasta ahora (*full batch descent*):

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

- *Stochastic Gradient Descent (SGD)*:

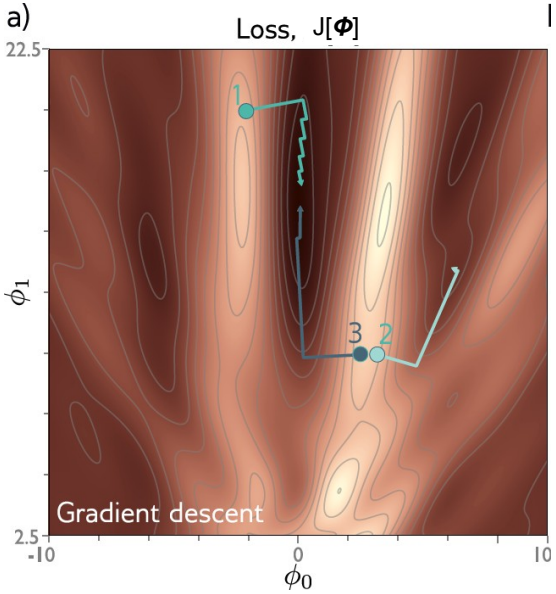
$$\phi_{t+1} \longleftarrow \phi_t - \alpha \frac{\partial \ell_i[\phi_t]}{\partial \phi}$$

Con α, tasa de aprendizaje (*learning rate*), de valor fijo (en el epoch).

¡Un único ejemplo!

L. Bottou 2010. Large-Scale Machine Learning with Stochastic Gradient Descent

Decenso de Gradiente Estocástico (SGD)



- Hasta ahora (*full batch descent*):

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

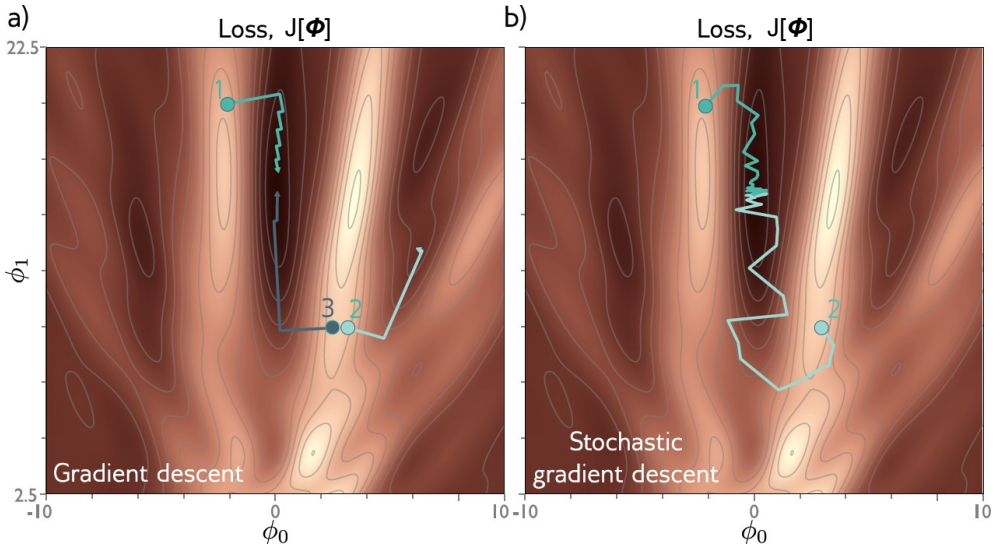
- *Stochastic Gradient Descent (SGD) con mini-batches*:

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

Con α, tasa de aprendizaje (*learning rate*), de valor fijo (en el epoch).

¡Múltiples ejemplos en el mini-batch!

Decenso de Gradiente Estocástico



Descenso de Gradiente Estocástico (SGD)

- Paso 0. Inicializar los parámetros $\boldsymbol{\Phi}_0$.
- Repetir:
  - Paso 1. Sacar una muestra de datos B_t de D (el mini-batch)
  - Paso 2. Calcular derivadas de **la función de coste** con respecto a los parámetros en B_t

$$\left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t} \leftarrow \frac{1}{N_B} \sum_{i \in B_t} \left.\frac{\partial J(f(\boldsymbol{x}_i;\boldsymbol{\Phi}), y_i)}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

  - Paso 3. Actualizar los parámetros de acuerdo con:

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{\Phi}_t - \alpha \left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

¡Muestrear aleatoriamente un subconjunto de datos es demasiado lento!
(pensad en el acceso aleatorio a memoria con millones)

Descenso de Gradiente Estocástico (SGD)

- Paso 0. Inicializar los parámetros $\boldsymbol{\Phi}_0$.
- Repetir:
  - Paso 1. Sacar una muestra de datos B_t de D (el mini-batch)
  - Paso 2. Calcular derivadas de **la función de coste** con respecto a los parámetros en B

$$\left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t} \leftarrow \frac{1}{N_B} \sum_{i \in B_t} \left.\frac{\partial J(f(\boldsymbol{x}_i;\boldsymbol{\Phi}), y_i)}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

  - Paso 3. Actualizar los parámetros de acuerdo con:

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{\Phi}_t - \alpha \left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

En la práctica: 1º se barajan los datos aleatoriamente (shuffle):

Descenso de Gradiente Estocástico (SGD)

- Paso 0. Inicializar los parámetros $\boldsymbol{\Phi_0}$.
- Repetir:
  - Paso 1. Sacar una muestra de datos B_t de D (el mini-batch)
  - Paso 2. Calcular derivadas de **la función de coste** con respecto a los parámetros en B

$$\left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t} \leftarrow \frac{1}{N_B}\sum_{i\in B_t}\left.\frac{\partial J(f(\boldsymbol{x_i};\boldsymbol{\Phi}),y_i)}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

  - Paso 3. Actualizar los parámetros de acuerdo con:

$$\boldsymbol{w_{t+1}} \leftarrow \boldsymbol{\Phi_t} - \alpha \left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

En la práctica: 2º se toman los mini-baches secuencialmente desde el principio:

Descenso de Gradiente Estocástico (SGD)

- Paso 0. Inicializar los parámetros $\boldsymbol{\Phi_0}$.
- Repetir:
  - Paso 1. Sacar una muestra de datos B_t de D (el mini-batch)
  - Paso 2. Calcular derivadas de **la función de coste** con respecto a los parámetros en B

$$\left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t} \leftarrow \frac{1}{N_B}\sum_{i\in B_t}\left.\frac{\partial J(f(\boldsymbol{x_i};\boldsymbol{\Phi}),y_i)}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

  - Paso 3. Actualizar los parámetros de acuerdo con:

$$\boldsymbol{w_{t+1}} \leftarrow \boldsymbol{\Phi_t} - \alpha \left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

En la práctica: 2º se toman los mini-baches secuencialmente desde el principio:

Descenso de Gradiente Estocástico (SGD)

- Paso 0. Inicializar los parámetros $\boldsymbol{\Phi_0}$.
- Repetir:
  - Paso 1. Sacar una muestra de datos B_t de D (el mini-batch)
  - Paso 2. Calcular derivadas de **la función de coste** con respecto a los parámetros en B

$$\left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t} \leftarrow \frac{1}{N_B}\sum_{i\in B_t}\left.\frac{\partial J(f(\boldsymbol{x_i};\boldsymbol{\Phi}),y_i)}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

  - Paso 3. Actualizar los parámetros de acuerdo con:

$$\boldsymbol{w_{t+1}} \leftarrow \boldsymbol{\Phi_t} - \alpha \left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

En la práctica: 2º se toman los mini-baches secuencialmente desde el principio:

Descenso de Gradiente Estocástico (SGD)

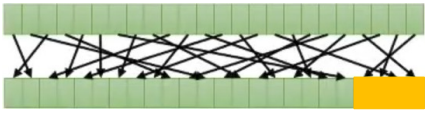
- Paso 0. Inicializar los parámetros $\boldsymbol{\Phi_0}$.
- Repetir:
  - Paso 1. Sacar una muestra de datos B_t de D (el mini-batch)
  - Paso 2. Calcular derivadas de **la función de coste** con respecto a los parámetros en B

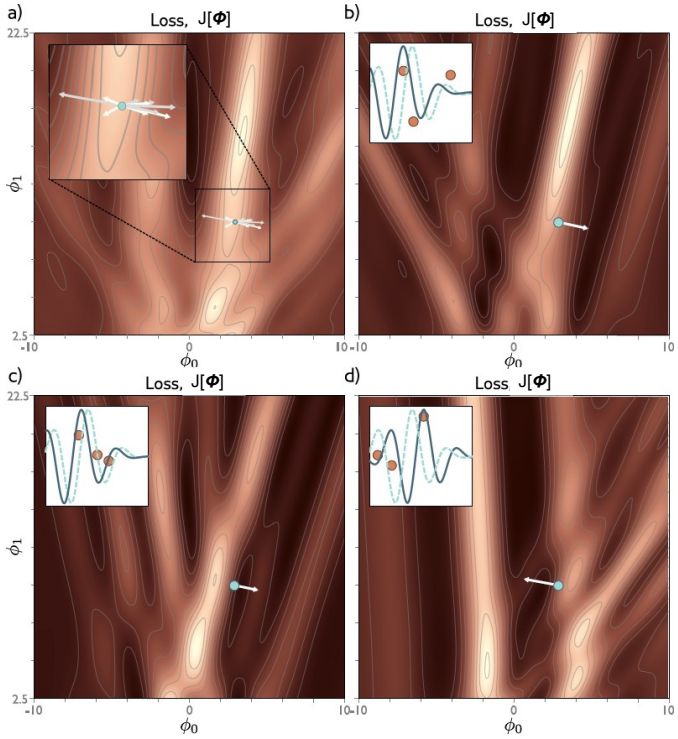
$$\left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t} \leftarrow \frac{1}{N_B}\sum_{i\in B_t}\left.\frac{\partial J(f(\boldsymbol{x_i};\boldsymbol{\Phi}),y_i)}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

  - Paso 3. Actualizar los parámetros de acuerdo con:

$$\boldsymbol{w_{t+1}} \leftarrow \boldsymbol{\Phi_t} - \alpha \left.\frac{\partial J(\boldsymbol{\Phi})}{\partial \boldsymbol{\Phi}}\right|_{\boldsymbol{\Phi}=\boldsymbol{\Phi}_t}$$

En la práctica: 3º Cuando se terminan tenemos una época (epoch) y volvemos al principio



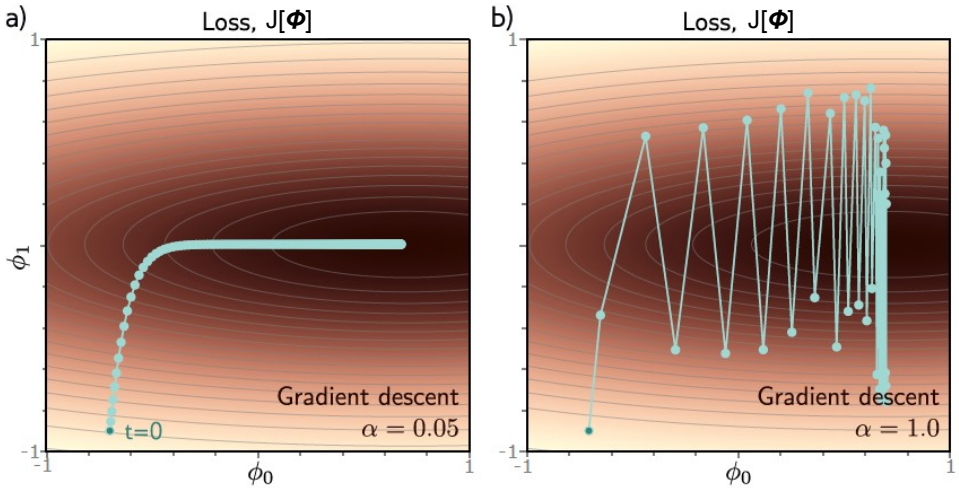


Propiedades del SGD

- Puede escapar de mínimos locales
- Añade ruido, pero realiza actualizaciones razonables basada en parte de los datos
- Utiliza todos los datos por igual
- Menos costoso computacionalmente que la versión *full batch*
- Parece encontrar mejores soluciones

- No converge en el sentido tradicional
- Planificación del learning rate (*learning rate schedule*) – disminuirlo (o aumentarlo) cada cierto número de épocas.

Problema: diferente magnitud en gradientes



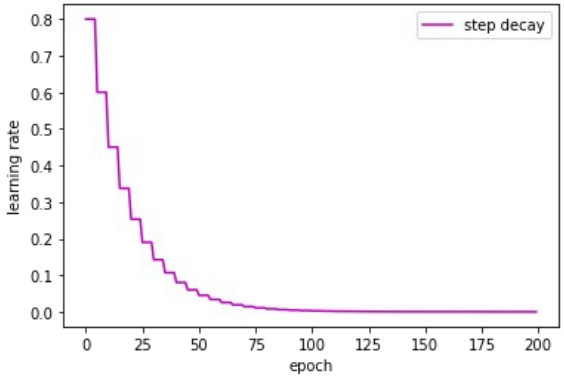
Planificación del learning rate

- Disminuirlo (o aumentarlo) cada cierto número de épocas.

Ejemplo: *step decay*

$$\alpha^{t+1} = \alpha_0 \cdot c^{\lfloor \tau / s \rfloor}$$

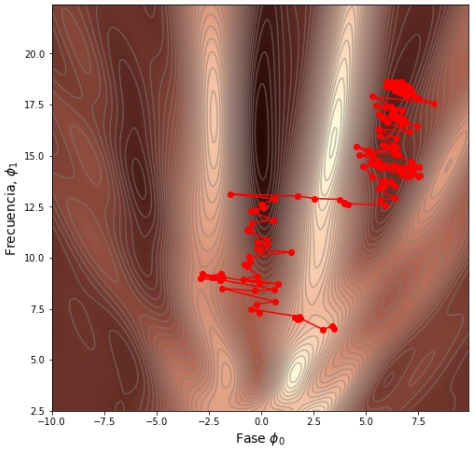
α_0 - learning rate inicial
c – coeficiente de reducción del learning rate
τ – epoch actual
s – cada cuantos epochs se reduce el learning rate multiplicando por c



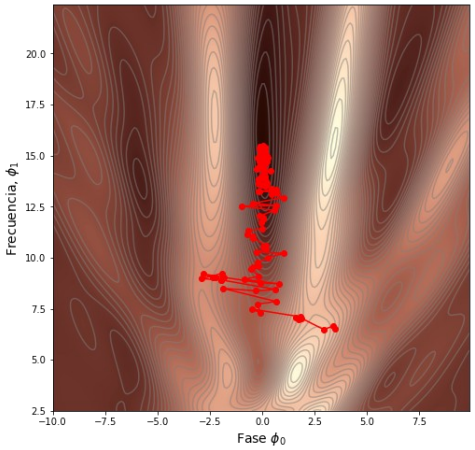
Planificación del learning rate

- SGD puede no converger si no se disminuye α:

Learning rate fijo (α=0.8)



Step decay (α_0=0.8)



2.3 Entrenamiento de modelos

- Conceptos matemáticos
- Algoritmo de descenso de gradiente
- Ejemplo con regresión lineal
- Ejemplo con modelo de Gabor
- Descenso de Gradiente Estocástico (SGD)
- Momento
- Adam

Momento

- Suma ponderada del gradiente con los anteriores

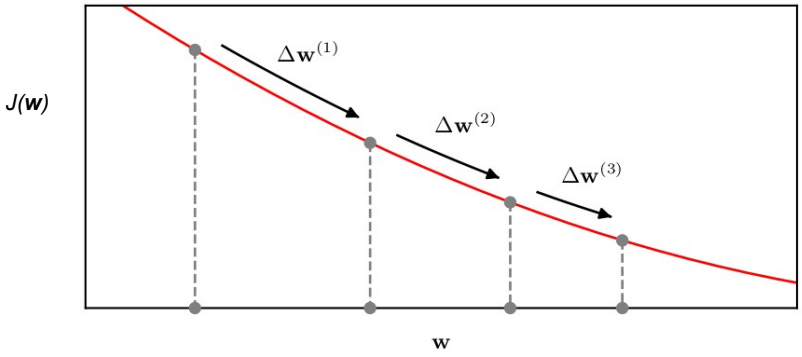
$$\mathbf{m}_{t+1} \leftarrow \beta \cdot \mathbf{m}_t + (1-\beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}}$$

$$\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t - \alpha \cdot \mathbf{m}_{t+1}$$

Típicamente β = 0,9

Con learning rate fijo ...

- Pasos cada vez más pequeños cuando J(**w**) disminuye la curvatura:



Momento

- Suma ponderada del gradiente con los anteriores

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

$$\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t - \alpha \cdot \mathbf{m}_{t+1}$$

El α efectivo (el learning rate):
- *se incrementa si todos los gradientes están alineados en múltiples iteraciones.*
- *decrece si el gradiente cambia de dirección repetidamente (los términos se cancelan).*

Momento

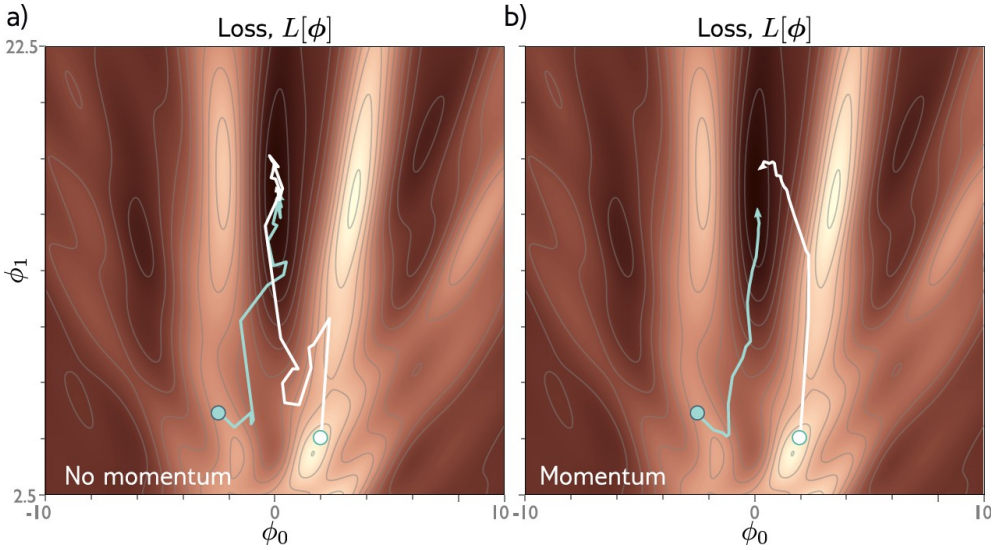
- Suma ponderada del gradiente con los anteriores

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

$$\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t - \alpha \cdot \mathbf{m}_{t+1}$$

Reduce el comportamiento oscilatorio en los valles (la trayectoria es más suave).

SGD + Momento



Momento acelerado de Nesterov

- El momento es una "predicción" de hacia dónde vamos

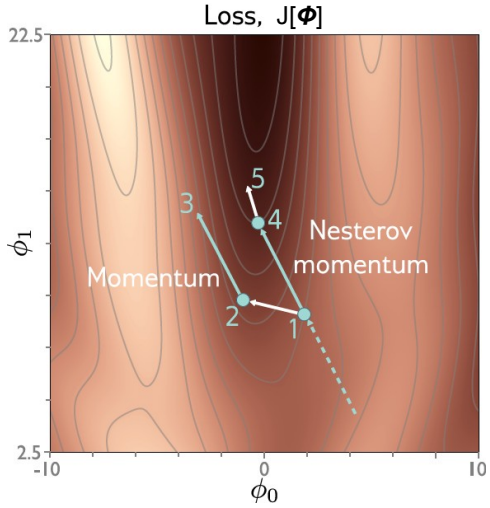
$$\mathbf{m}_{t+1} \leftarrow \beta \cdot \mathbf{m}_t + (1-\beta) \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}}$$

$$\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t - \alpha \cdot \mathbf{m}_{t+1}$$

- Nesterov: moverse primero en la dirección predicha y DESPUÉS, medir el gradiente:

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

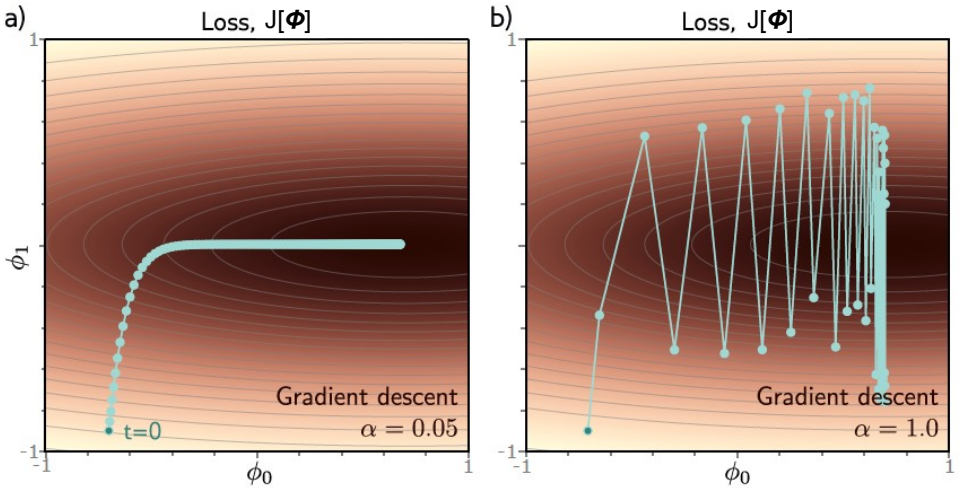
$$\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t - \alpha \cdot \mathbf{m}_{t+1}$$



2.3 Entrenamiento de modelos

- Conceptos matemáticos
- Algoritmo de descenso de gradiente
- Ejemplo con regresión lineal
- Ejemplo con modelo de Gabor
- Descenso de Gradiente Estocástico (SGD)
- Momento
- Adam

Problema: diferente magnitud en gradientes



Idea: Normalizar los gradientes

- Medir el gradiente en el mini-batch y su cuadrado para cada parámetro del modelo:

$$\mathbf{m}_{t+1} \leftarrow \frac{\partial J[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}}$$

$$\mathbf{v}_{t+1} \leftarrow \frac{\partial J[\boldsymbol{\phi}_t]^2}{\partial \boldsymbol{\phi}}$$

- Normalizar el momento:

$$\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t - \alpha \cdot \frac{\mathbf{m}_{t+1}}{\sqrt{\mathbf{v}_{t+1}} + \epsilon}$$

v_{t+1} es diferente para cada parámetro. El learning rate efectivo es diferente para cada parámetro.

Idea: Normalizar los gradientes

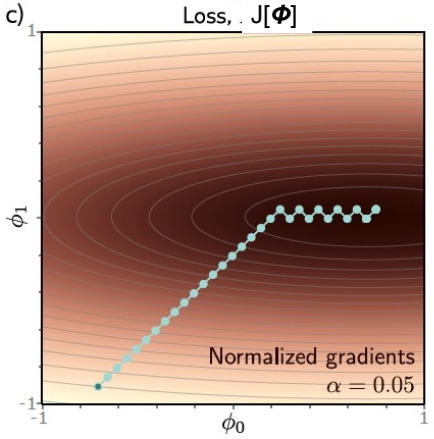
- Medir el gradiente en el mini-batch y su cuadrado para cada parámetro del modelo:

$$\mathbf{m}_{t+1} \leftarrow \frac{\partial J[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}}$$

$$\mathbf{v}_{t+1} \leftarrow \frac{\partial J[\boldsymbol{\phi}_t]^2}{\partial \boldsymbol{\phi}}$$

- Normalizar el momento:

$$\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t - \alpha \cdot \frac{\mathbf{m}_{t+1}}{\sqrt{\mathbf{v}_{t+1}} + \epsilon}$$



Idea: Normalizar los gradientes

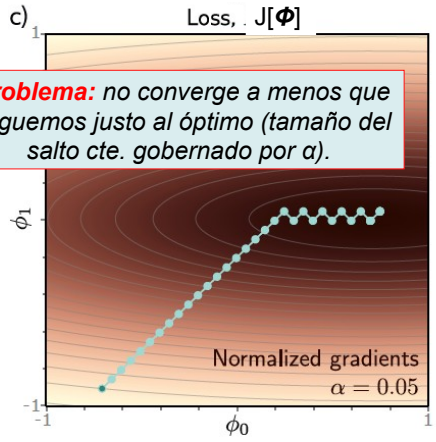
- Medir el gradiente en el mini-batch y su cuadrado para cada parámetro del modelo:

$$\mathbf{m}_{t+1} \leftarrow \frac{\partial J[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}}$$

$$\mathbf{v}_{t+1} \leftarrow \frac{\partial J[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}}^2$$

- Normalizar el momento:

$$\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t - \alpha \cdot \frac{\mathbf{m}_{t+1}}{\sqrt{\mathbf{v}_{t+1}} + \epsilon}$$



Problema: *no converge a menos que lleguemos justo al óptimo (tamaño del salto cte. gobernado por α).*

Adaptive moment estimation (Adam)

- Medir el gradiente en el mini-batch y su cuadrado para cada parámetro del modelo con momento:

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

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

Típicamente β = 0,9; γ = 0,990

Adaptive moment estimation (Adam)

- Medir el gradiente en el mini-batch y su cuadrado para cada parámetro del modelo con momento:

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

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

Típicamente β = 0,9; γ = 0,990

Problema: ***m**ₜ* y ***v**ₜ* *son 0 al comenzar las iteraciones.*

Adaptive moment estimation (Adam)

- Medir el gradiente en el mini-batch y su cuadrado para cada parámetro del modelo con momento:

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

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

- Tener cuidado al comienzo:

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

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

Típicamente β = 0,9; γ = 0,990

Adaptive moment estimation (Adam)

- Medir el gradiente en el mini-batch y su cuadrado para cada parámetro del modelo con momento:

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

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

- Tener cuidado al comienzo:

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

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

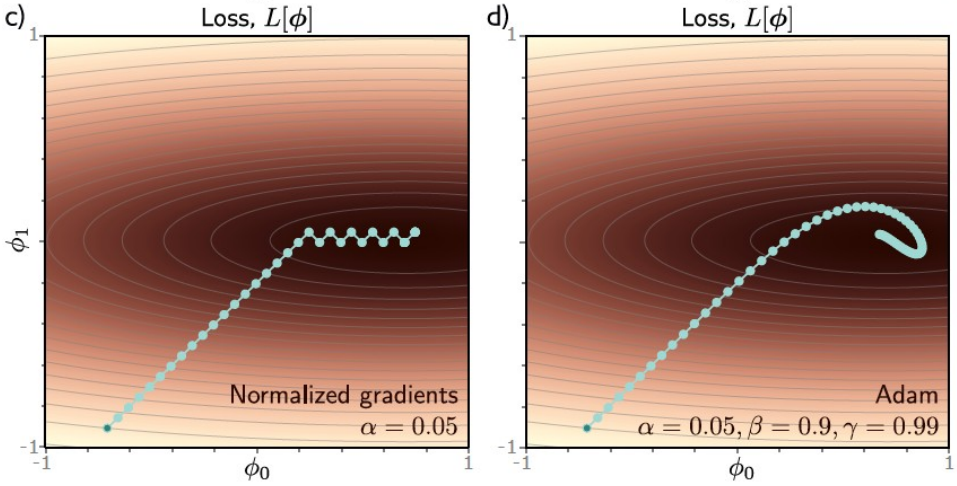
- Actualizar los parámetros:

$$\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t - \alpha \cdot \frac{\mathbf{m}_{t+1}}{\sqrt{\mathbf{v}_{t+1}} + \epsilon}$$

Adaptive moment estimation (Adam)

- Cada parámetro tiene su propio “learning rate efectivo” que se recalcula en cada iteración.
- ¡No es necesario el learning rate schedule! (En SGD sí que es necesario)

Adaptive moment estimation (Adam)



Hiperparámetros

- Elección del algoritmo de optimización
- Tasa de aprendizaje (*learning rate*)
- Momento

¡Son hiperparámetros del algoritmo de aprendizaje (del optimizador!