

Técnicas Avanzadas de Data Mining y Sistemas Inteligentes

Maestría en Informática
Escuela de Posgrado
Pontificia Universidad Católica del Perú

2018-2

Machine Learning

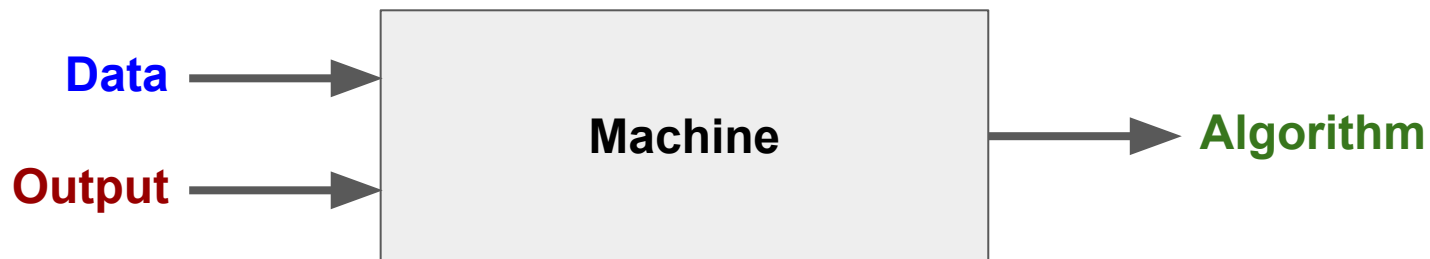
Traditional Programming



Traditional Programming



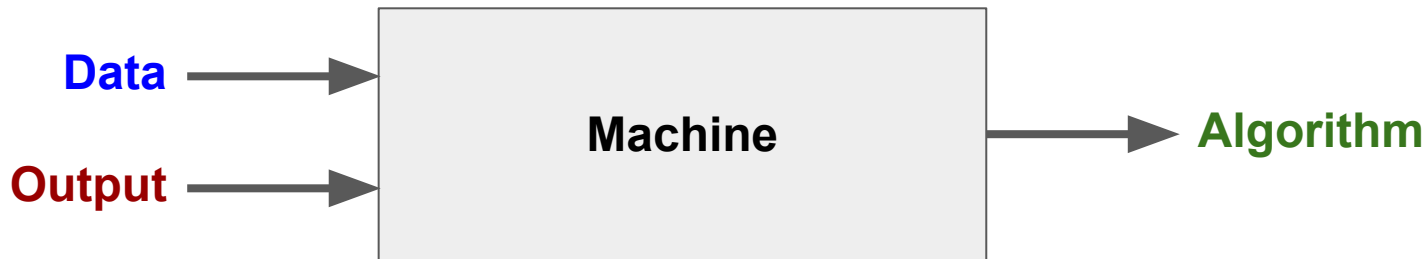
Machine Learning



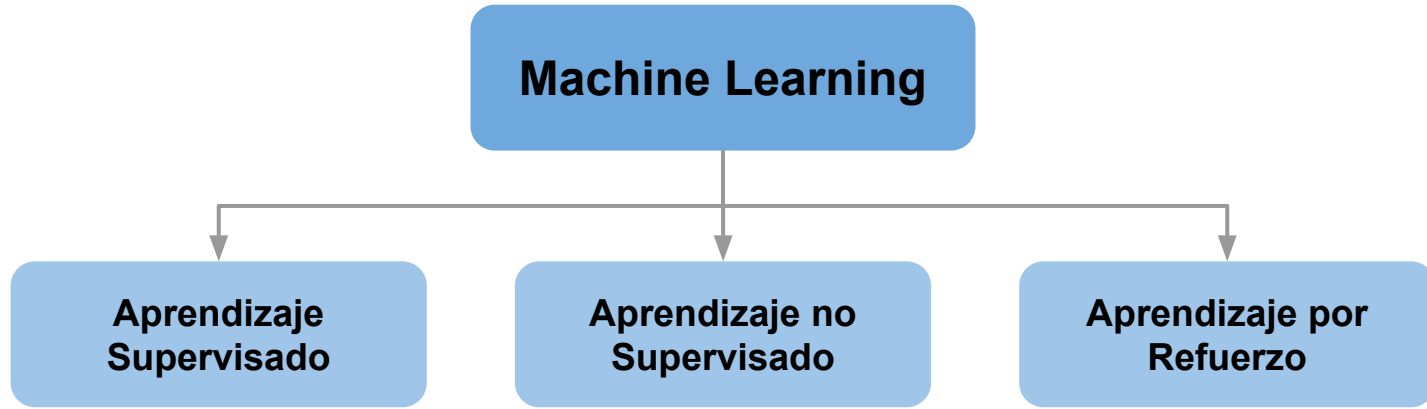
Traditional Programming

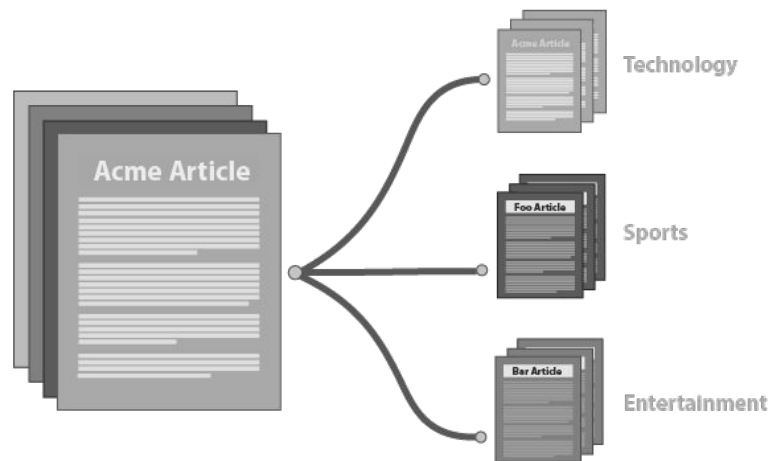
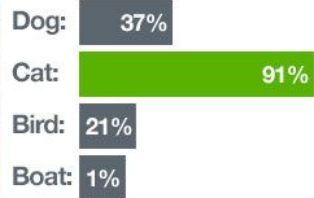
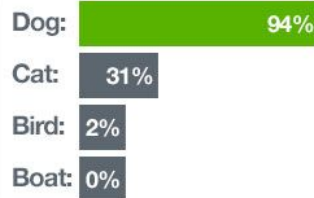


Machine Learning

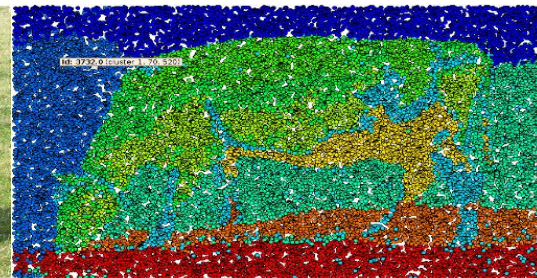


Los algoritmos de machine learning usan métodos computacionales para aprender directamente de la data, sin estar explícitamente programados.





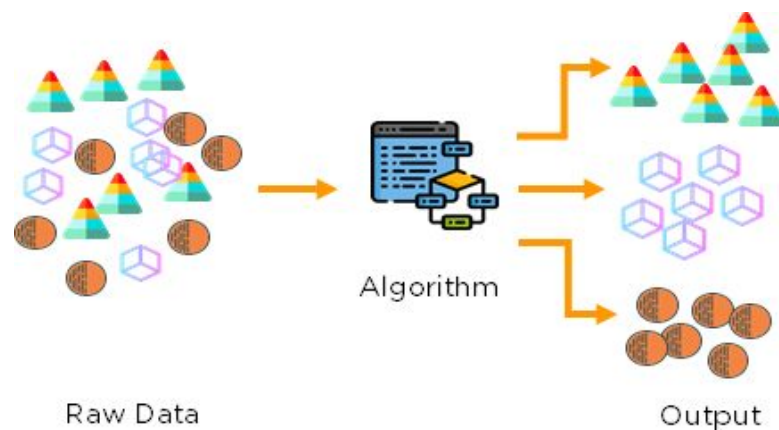
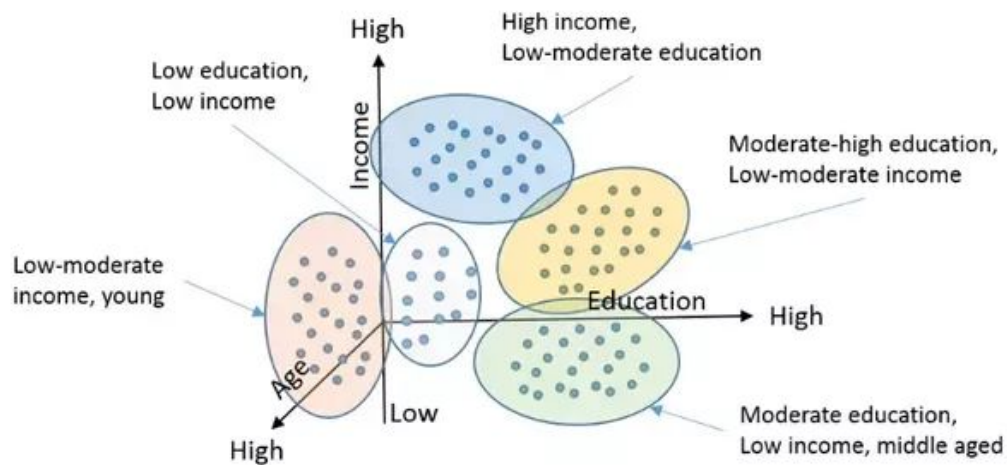
Ma



Aprendizaje
Supervisado

Aprendizaje no
Supervisado

Aprendizaje por
Refuerzo

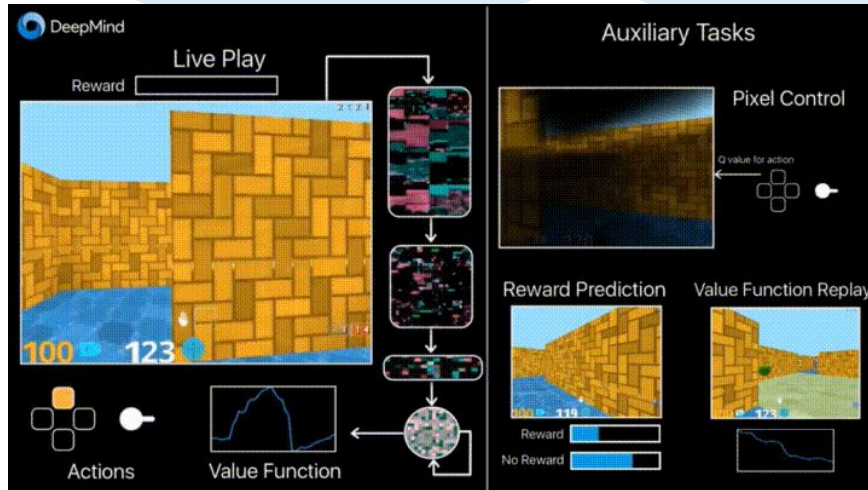


Machine Learning

Aprendizaje Supervisado

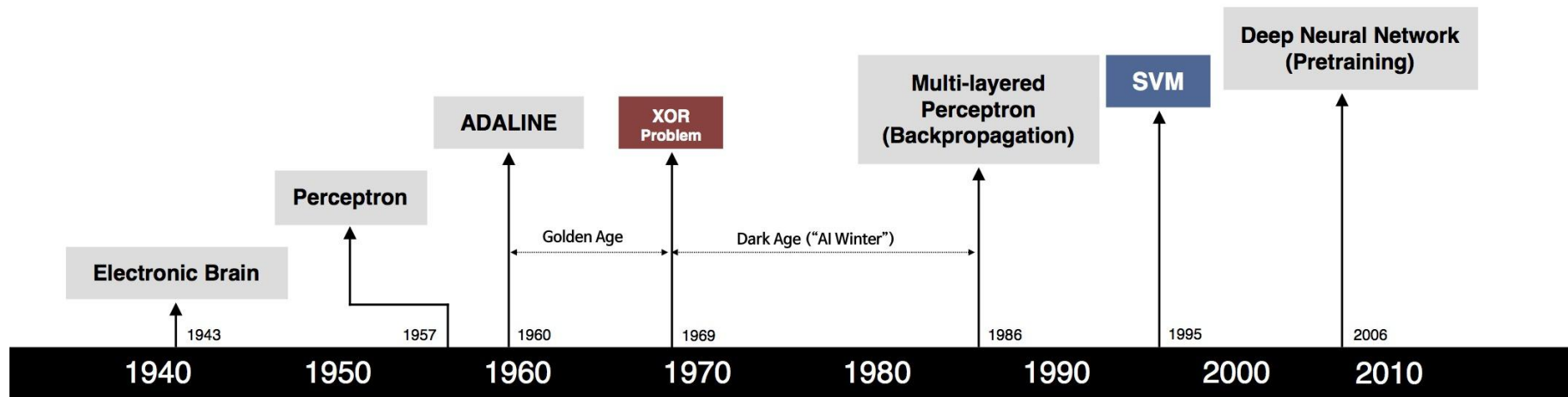
Aprendizaje no Supervisado

Aprendizaje por Refuerzo

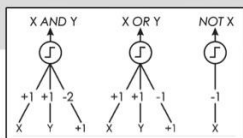


Deep Learning





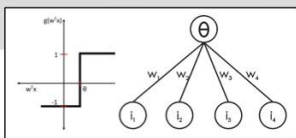
S. McCulloch – W. Pitts



- Adjustable Weights
- Weights are not Learned



F. Rosenblatt



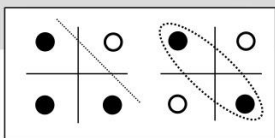
- Learnable Weights and Threshold



B. Widrow – M. Hoff



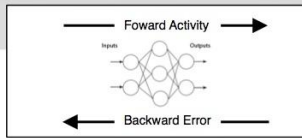
M. Minsky – S. Papert



- XOR Problem



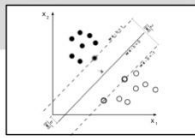
D. Rumelhart – G. Hinton – R. Williams



- Solution to nonlinearly separable problems
- Big computation, local optima and overfitting



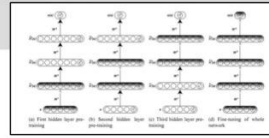
V. Vapnik – C. Cortes



- Limitations of learning prior knowledge
- Kernel function: Human Intervention



G. Hinton – S. Ruslan

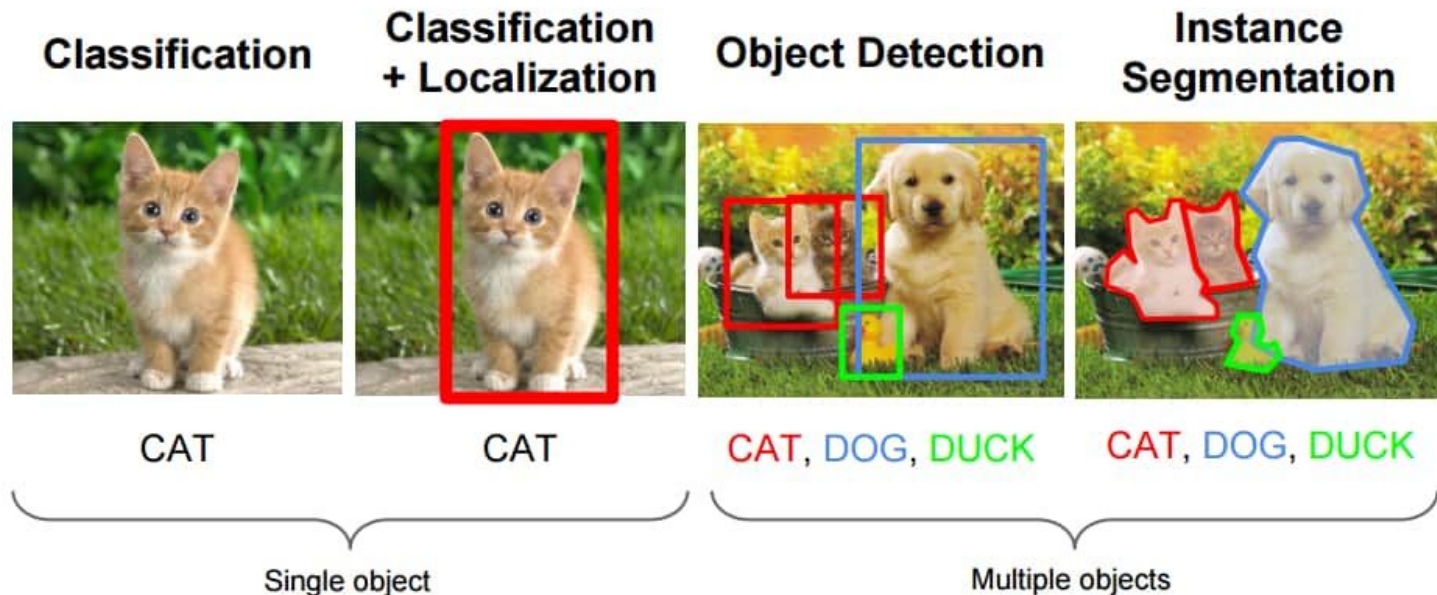


- Hierarchical feature Learning

¿Qué problemas se están resolviendo con DL?

- Visión computacional
 - Reconocimiento de imágenes
 - Segmentación de objetos
 - Reconstrucción de imágenes

Source: Fei-Fei Li, Andrej Karpathy & Justin Johnson
(2016) cs231n, Lecture 8 - Slide 8, Spatial
Localization and Detection (01/02/2016)



¿Qué problemas se están resolviendo con DL?

- Lenguaje natural (NLP)
 - Traducción
 - Parsing
 - Generación de conversaciones (chatbots)
 - Generación de resúmenes



¿Qué problemas se están resolviendo con DL?

- Procesamiento de señales o datos en secuencia

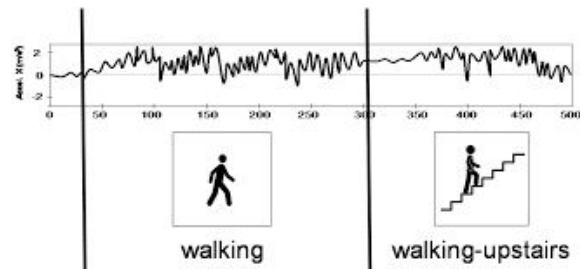
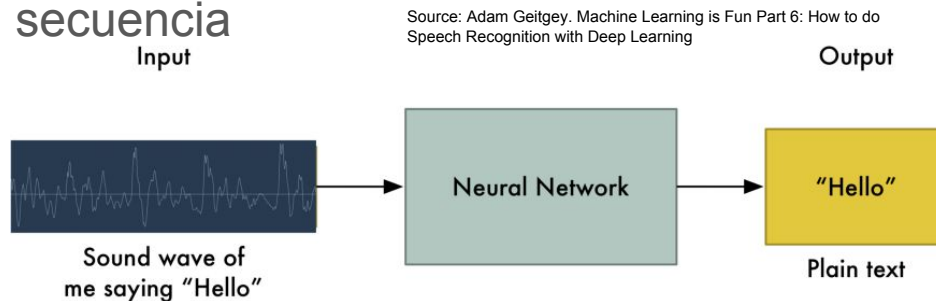
- Reconocimiento de acciones (sensores)
- Reconocimiento de voz

- Tareas combinadas

- Generación de descripciones de imágenes (Image captioning)
- Reconocimiento de videos



Source: Ran Xu, Priyanshu Agarwal, Suren Kumar, Venkat N. Krovi, and Jason J. Corso Combining Skeletal Pose with Local Motion for Human Activity Recognition



Source: Rubén San-Segundo, Juan M. Montero, José Moreno-Pimentel, José M. Pardo. HMM Adaptation for Improving a Human Activity Recognition System

Describes without errors

Describes with minor errors

Somewhat related to the image

Unrelated to the image



A person riding a motorcycle on a dirt road.



Two dogs play in the grass.



A skateboarder does a trick on a ramp.



A dog is jumping to catch a frisbee.



A group of young people playing a game of frisbee.



Two hockey players are fighting over the puck.



A little girl in a pink hat is blowing bubbles.



A refrigerator filled with lots of food and drinks.



A herd of elephants walking across a dry grass field.



A close up of a cat laying on a couch.



A red motorcycle parked on the side of the road.



A yellow school bus parked in a parking lot.

Image Captioning

[Vinyals et al., 2015]

A flower with thin pink petals resting around a cluster of white and yellow stamen.



This flower is white and yellow in colour, with petals that are multi coloured.



A fat round bird with a bright yellow breast and blue on top of the head.



The tan bird has a short beak, with stone black eyes.

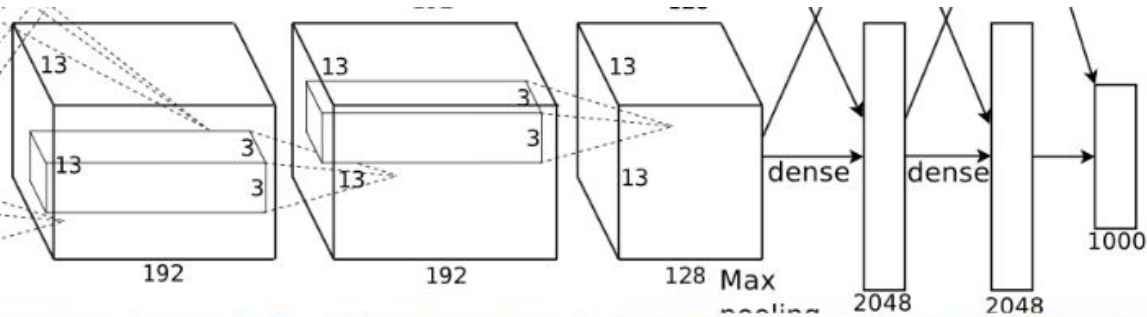


Text to Image Synthesis

[Reed et al., 2016]

Deep Learning Ingredientes

Algorithms



Data

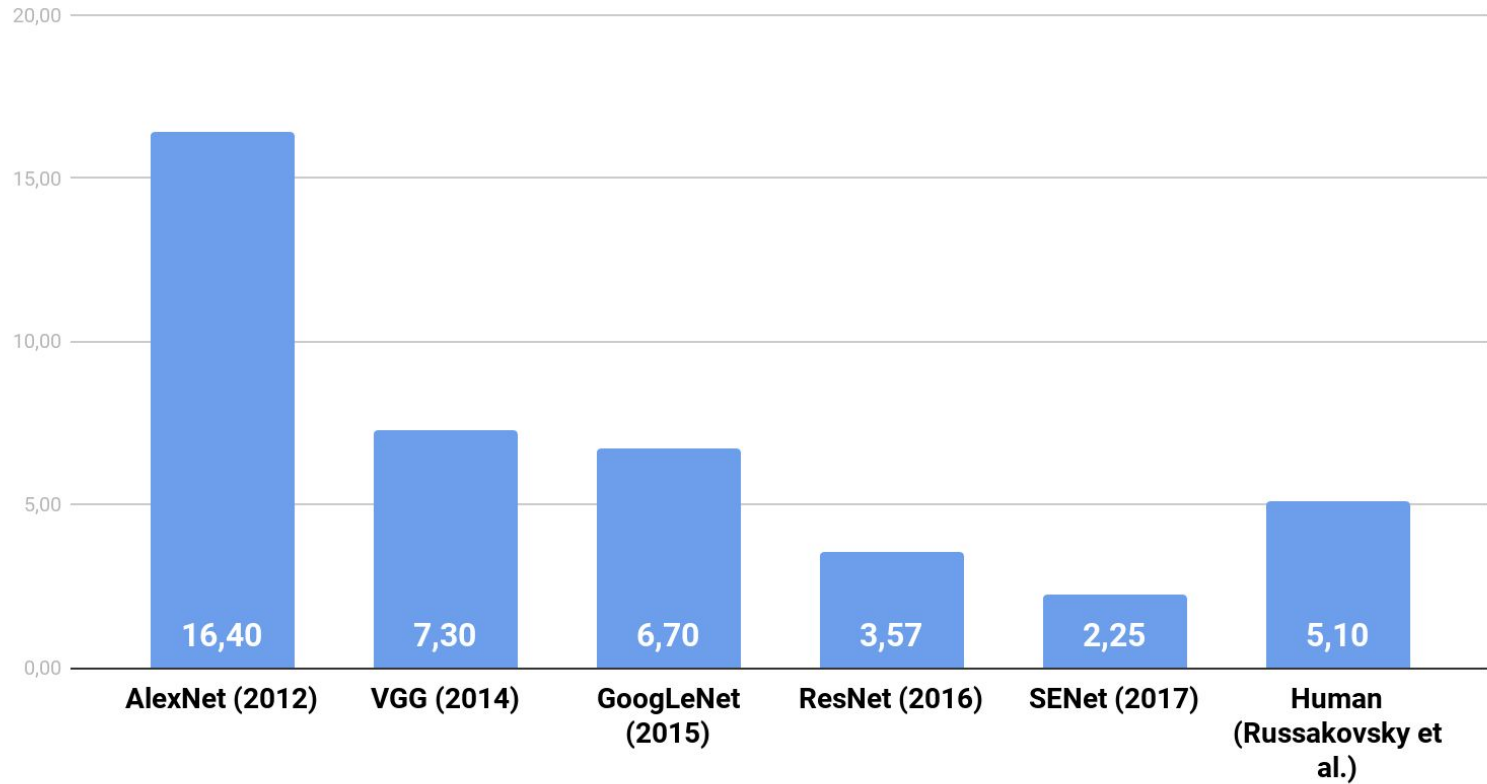
Computation

IMAGENET

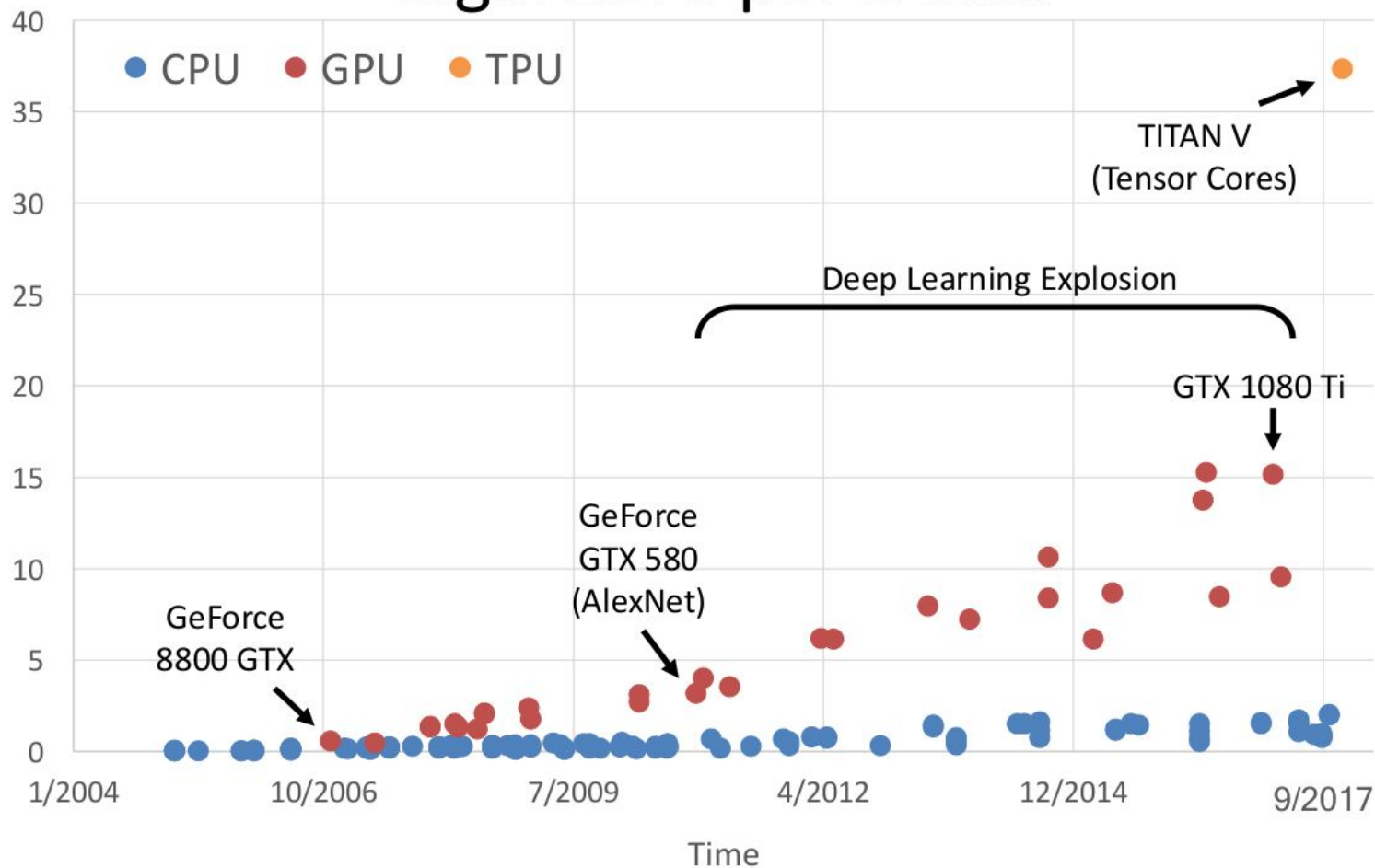
- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



ImageNet Top 5 Error Rate

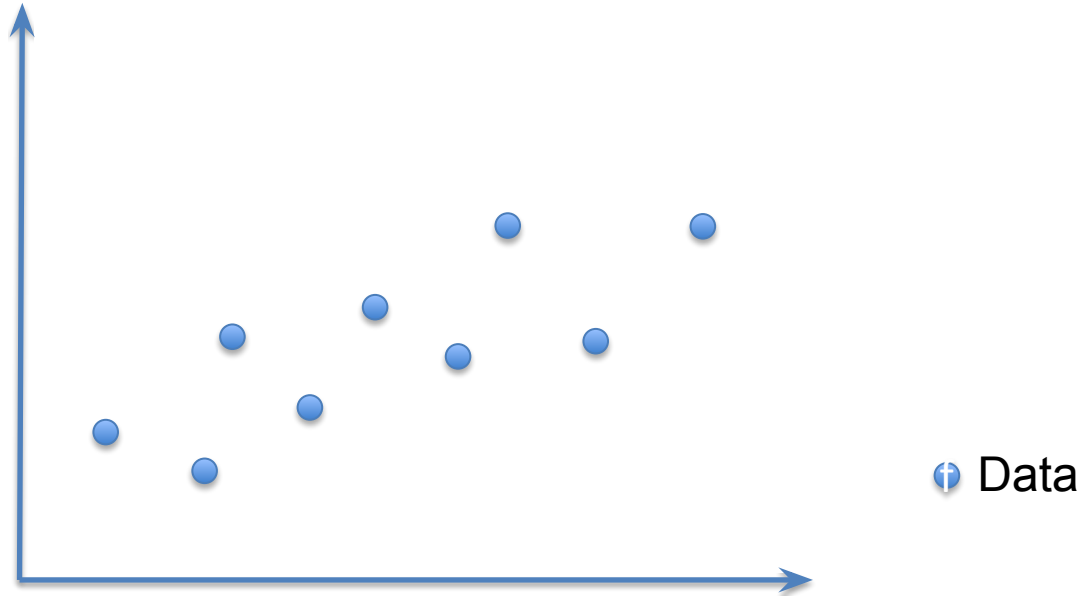


GigaFLOPs per Dollar

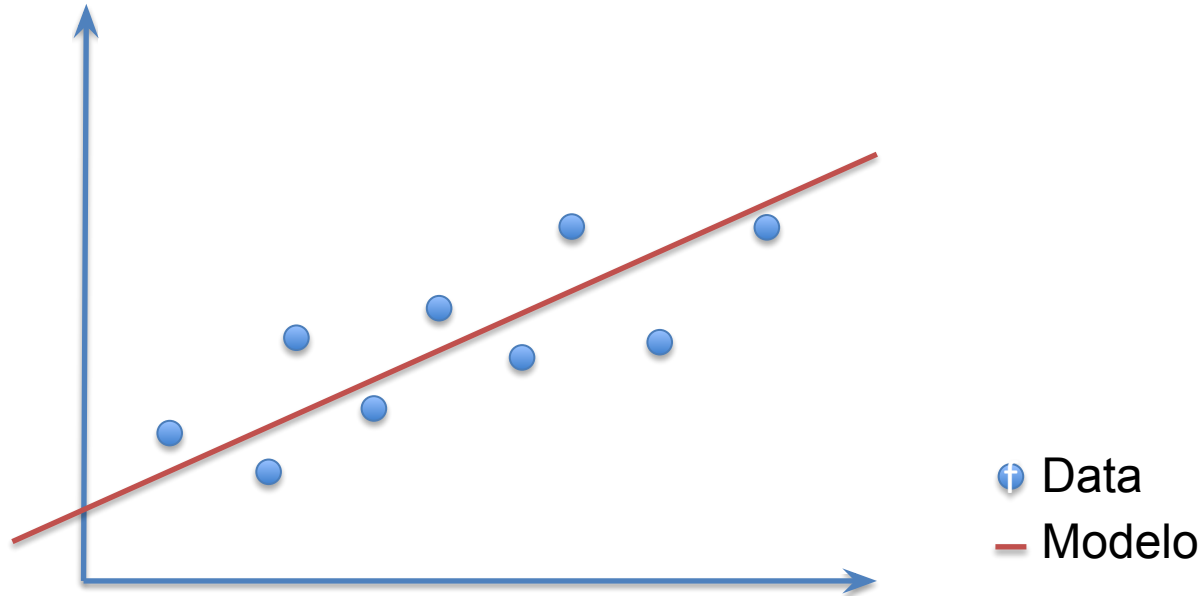


Regresión Lineal Simple

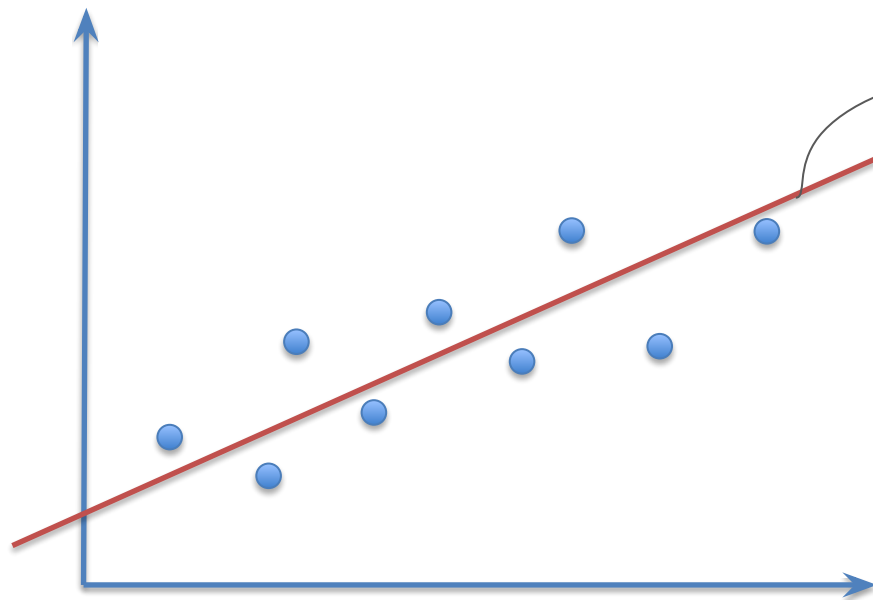
Regresión lineal simple



Regresión lineal simple



Regresión lineal simple



$$f(x) = ax + b$$

Donde:

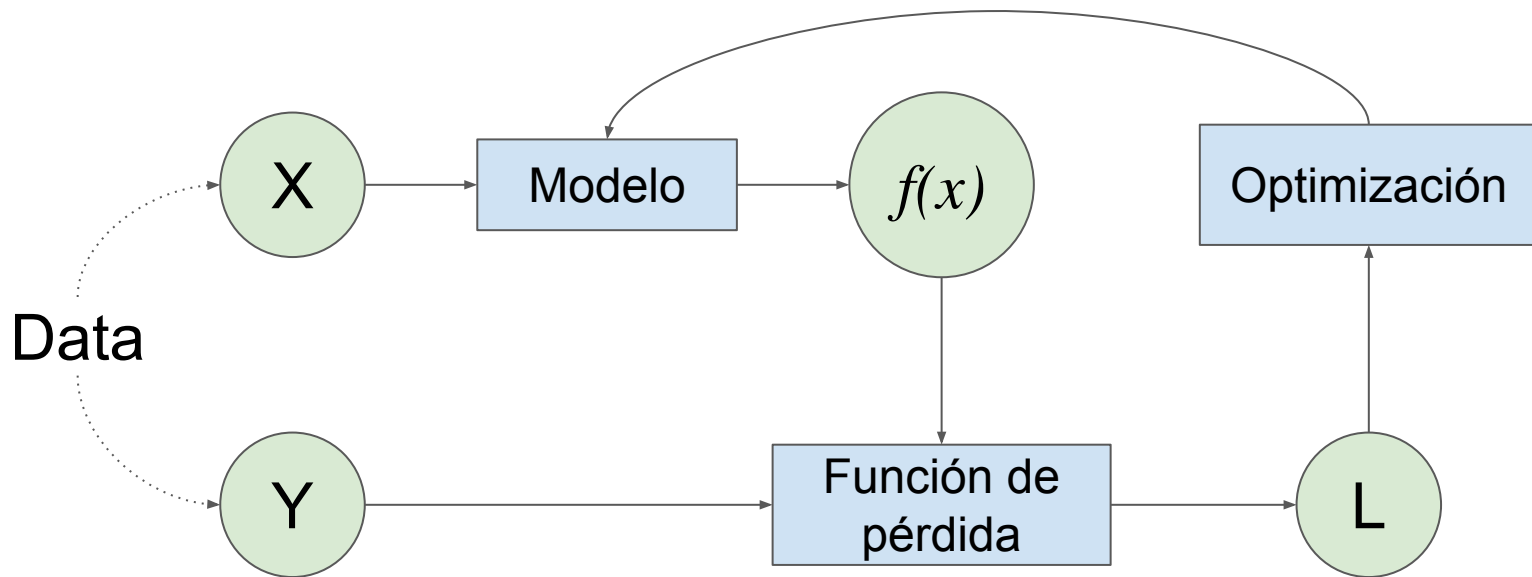
a = pendiente

b = término de intercepción (bias)

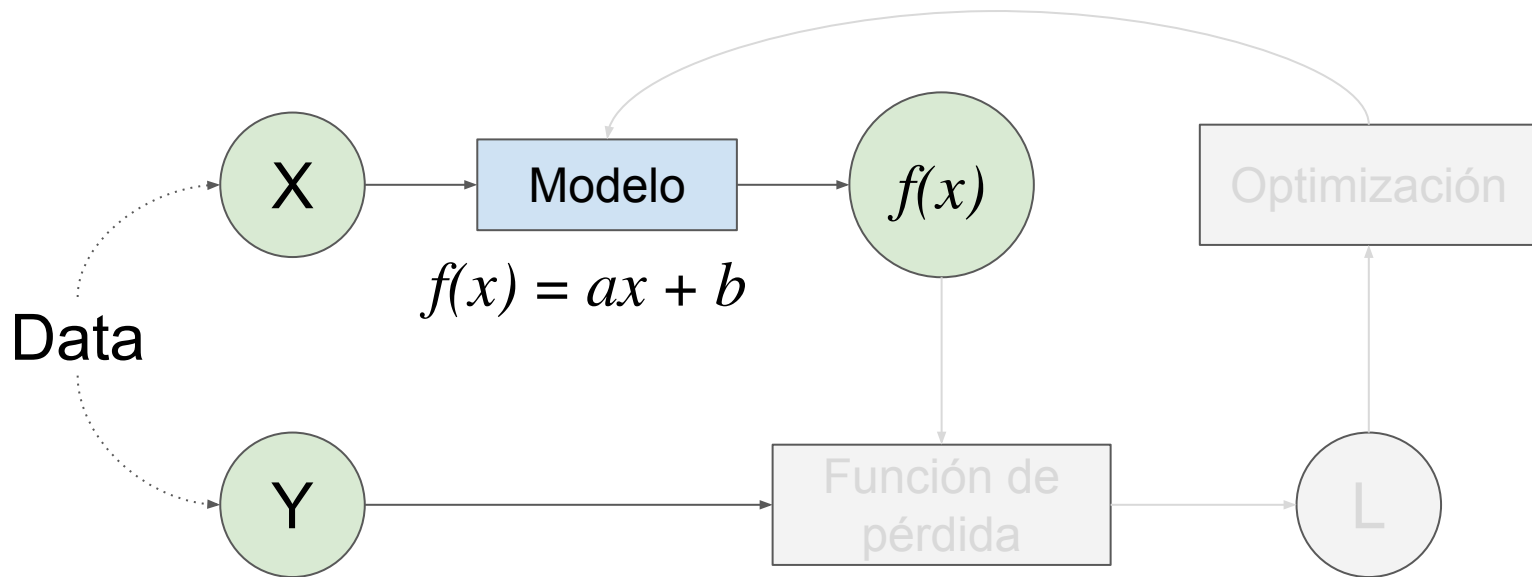
⬢ Data

— Modelo

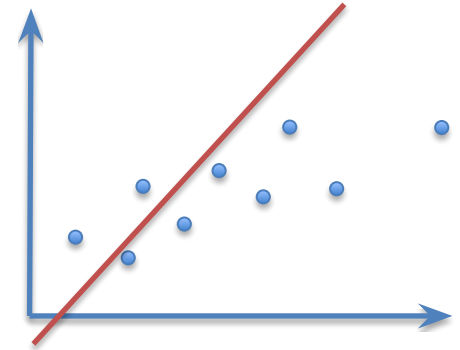
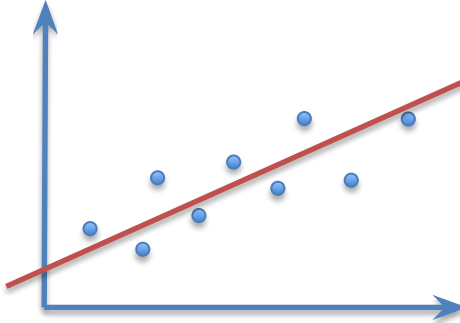
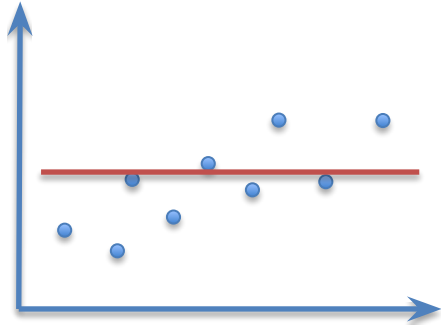
Regresión lineal simple



Regresión lineal simple

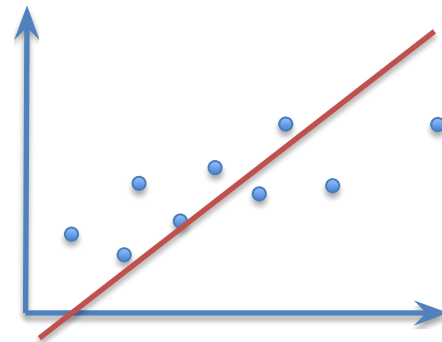
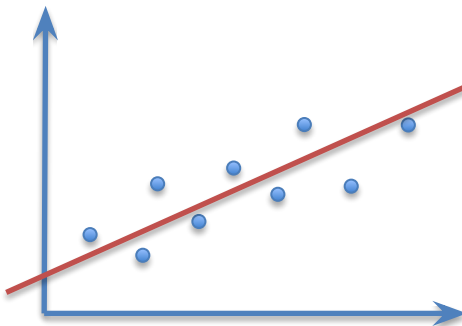
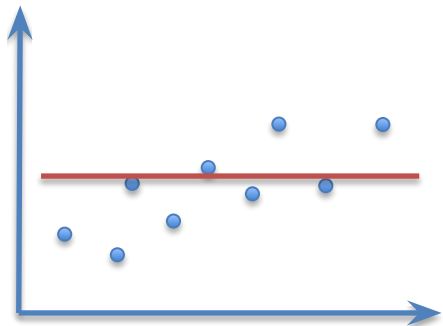


Función de pérdida



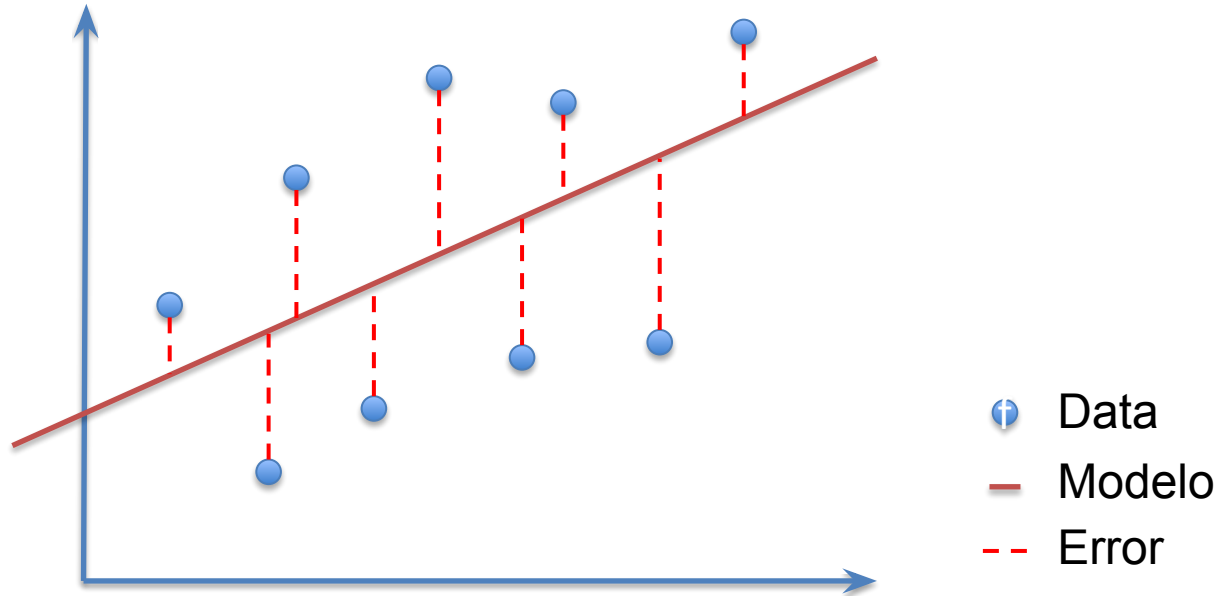
¿Cuál de los modelos describe mejor la data?

Función de pérdida

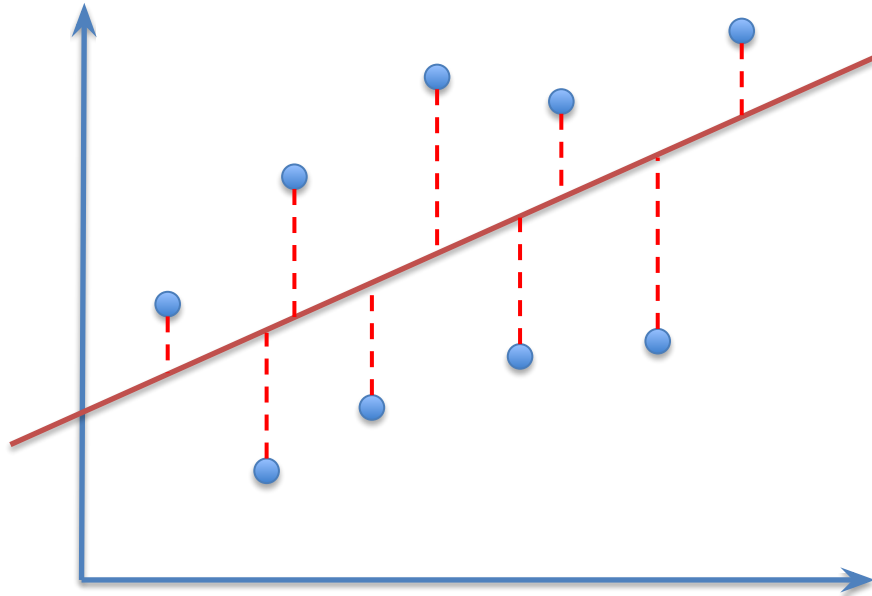


¿Cuál de los modelos describe mejor la data?
Evaluamos cada uno con una **función de pérdida** y escogemos el que tenga la menor pérdida.

Función de pérdida: MSE (Mean squared error)






Función de pérdida: MSE (Mean squared error)

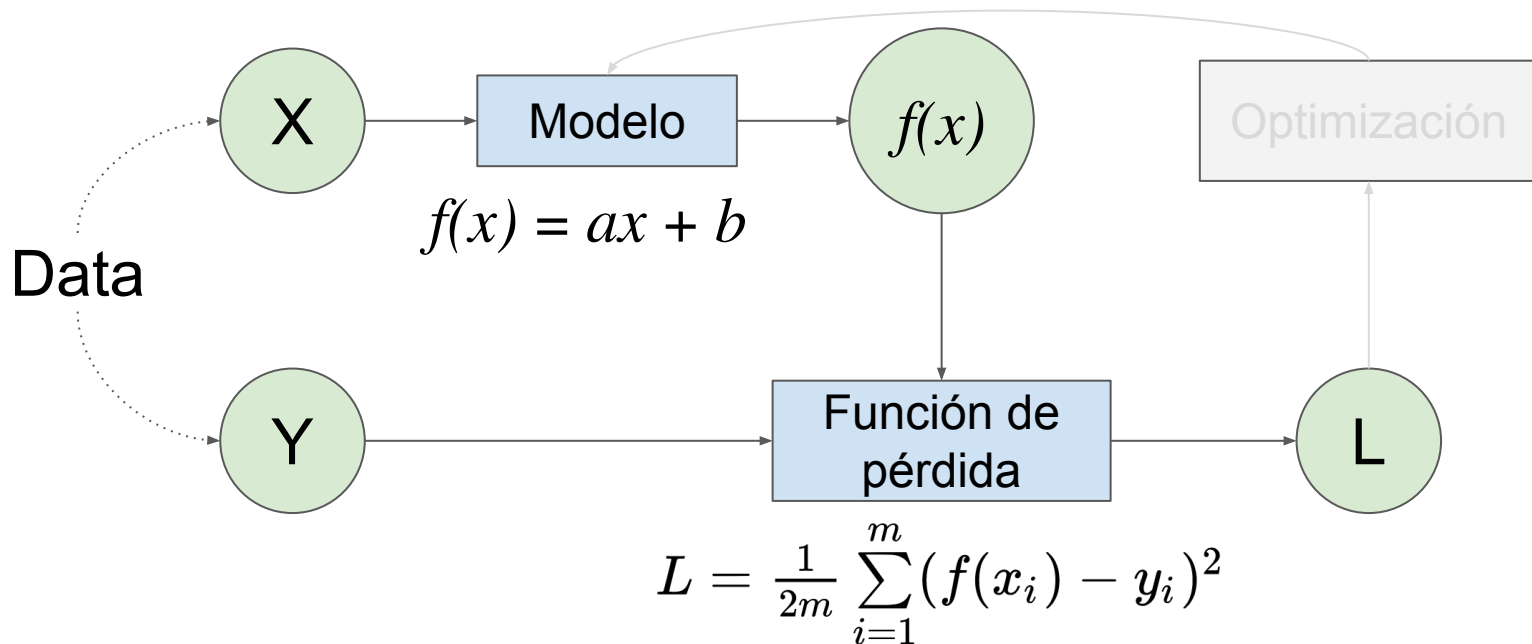


Mean squared error:

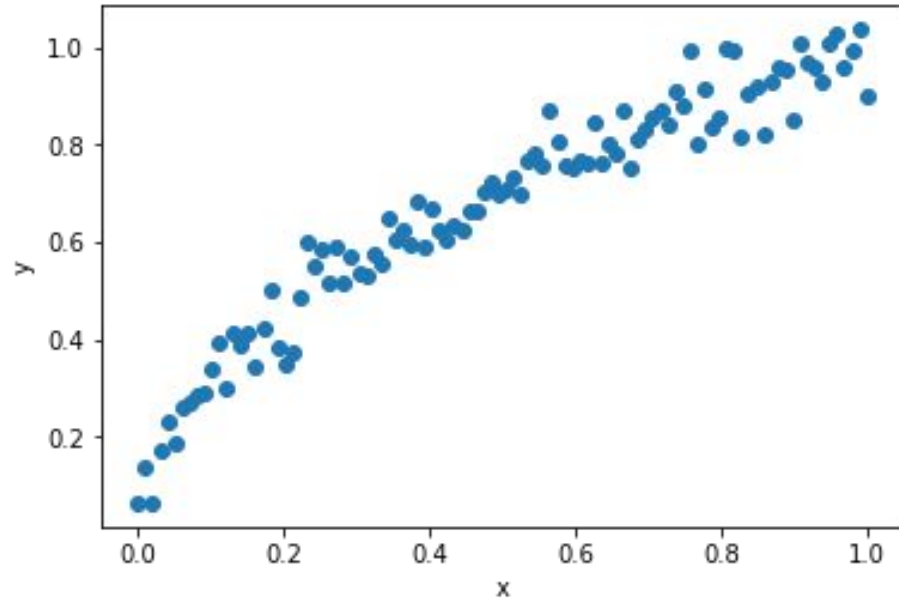
$$L = \frac{1}{2m} \sum_{i=1}^m (f(x_i) - y_i)^2$$

-  Data x
-  Modelo $f(x)$
-  Error $f(x) - y$

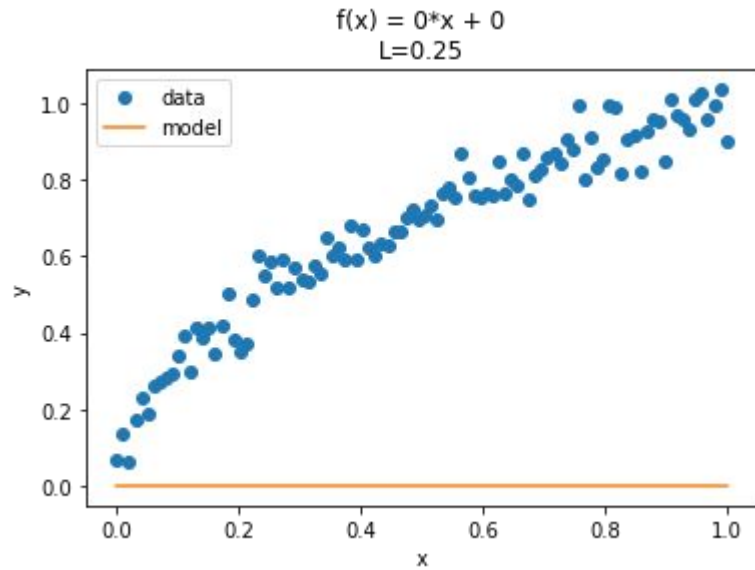
Función de pérdida: MSE (Mean squared error)



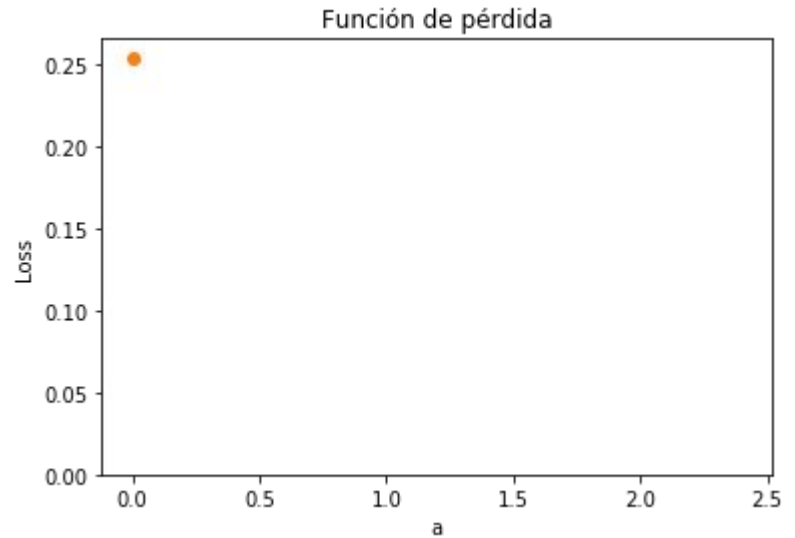
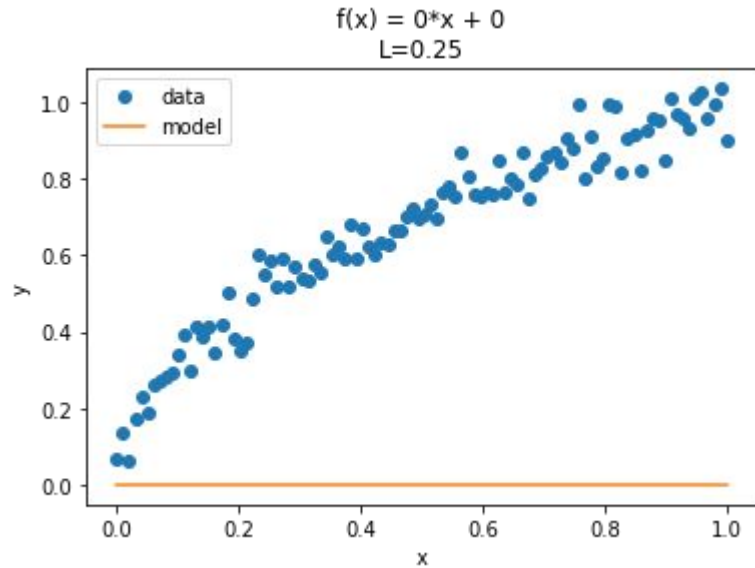
Optimización: Gradient descent



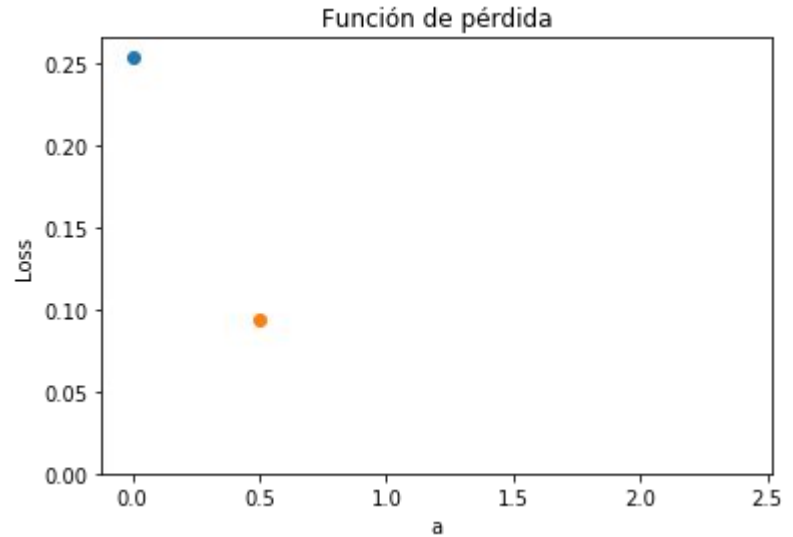
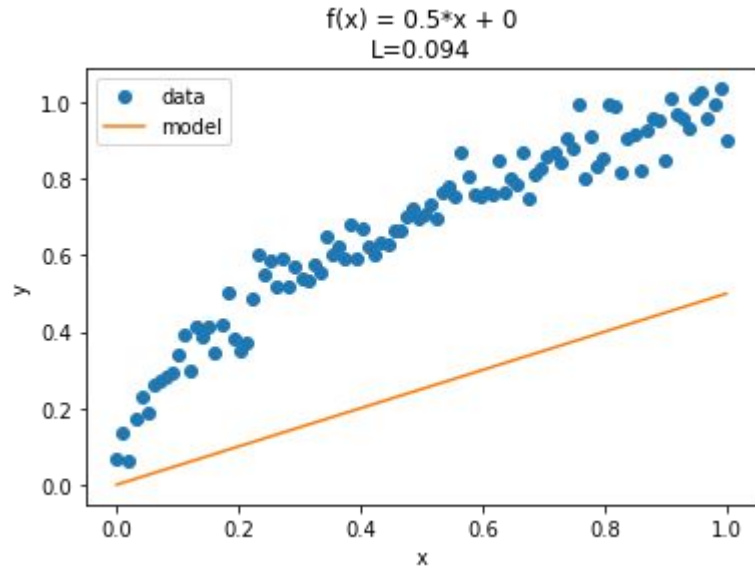
Optimización: Gradient descent



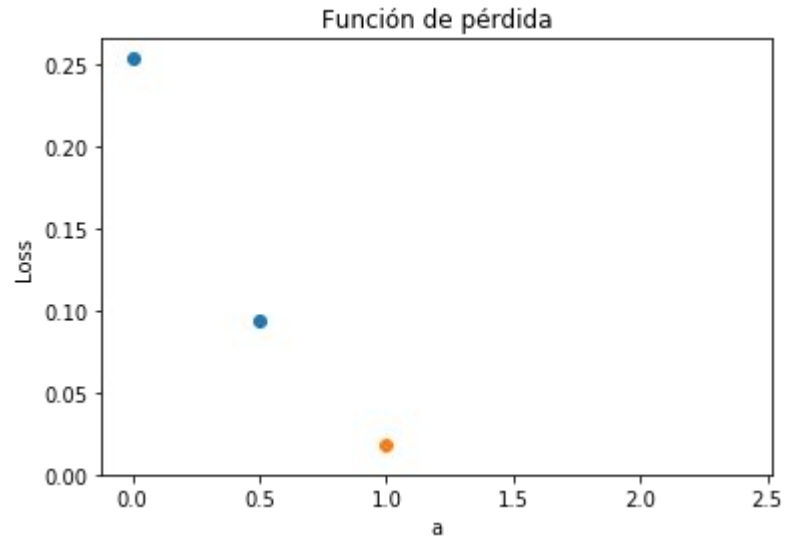
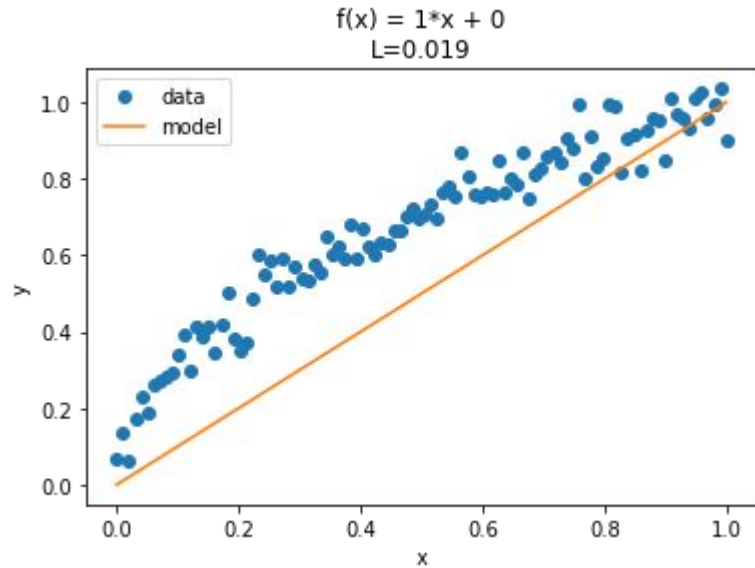
Optimización: Gradient descent



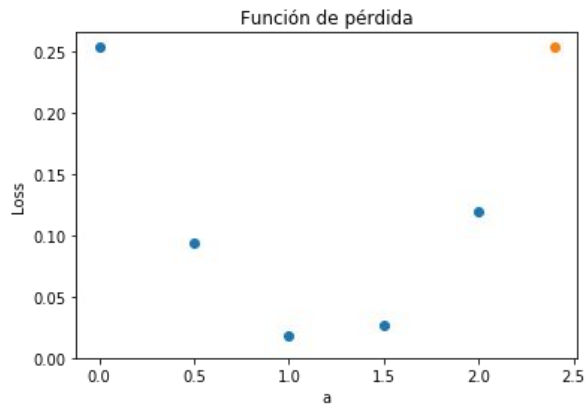
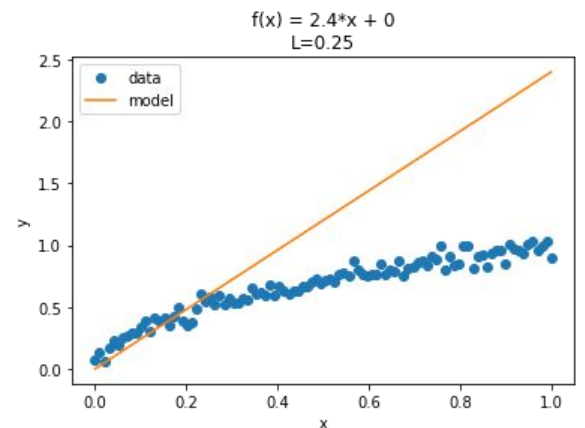
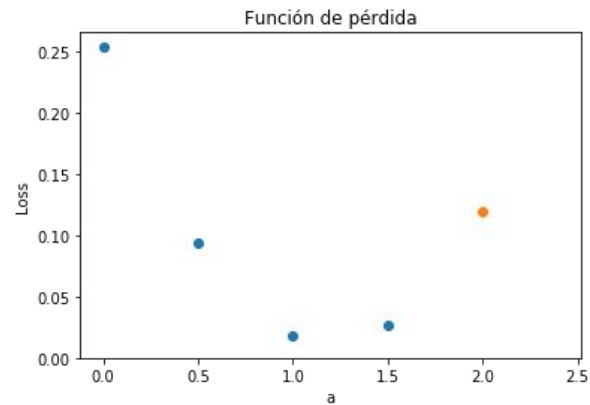
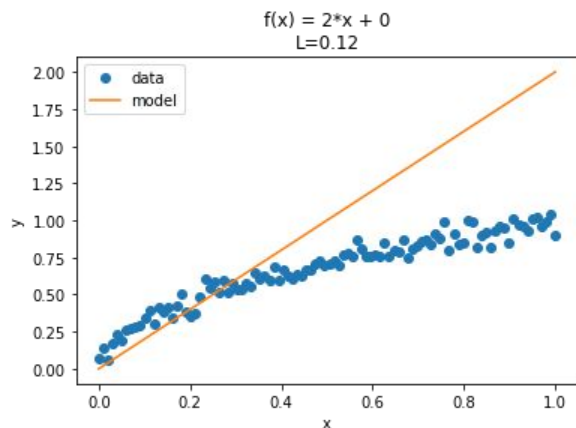
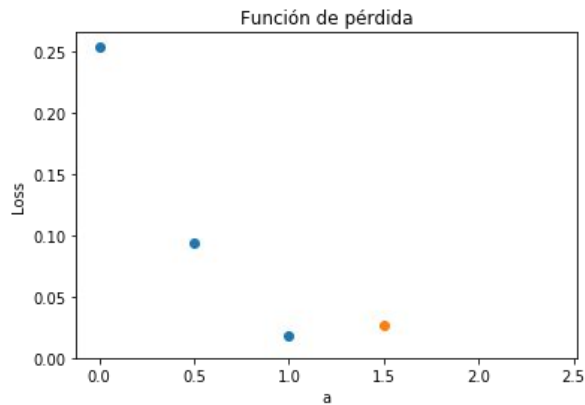
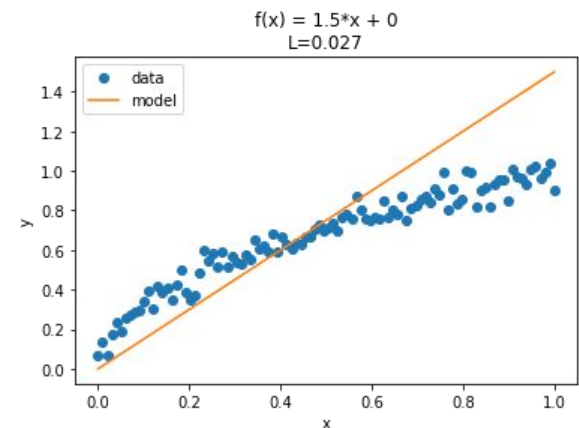
Optimización: Gradient descent



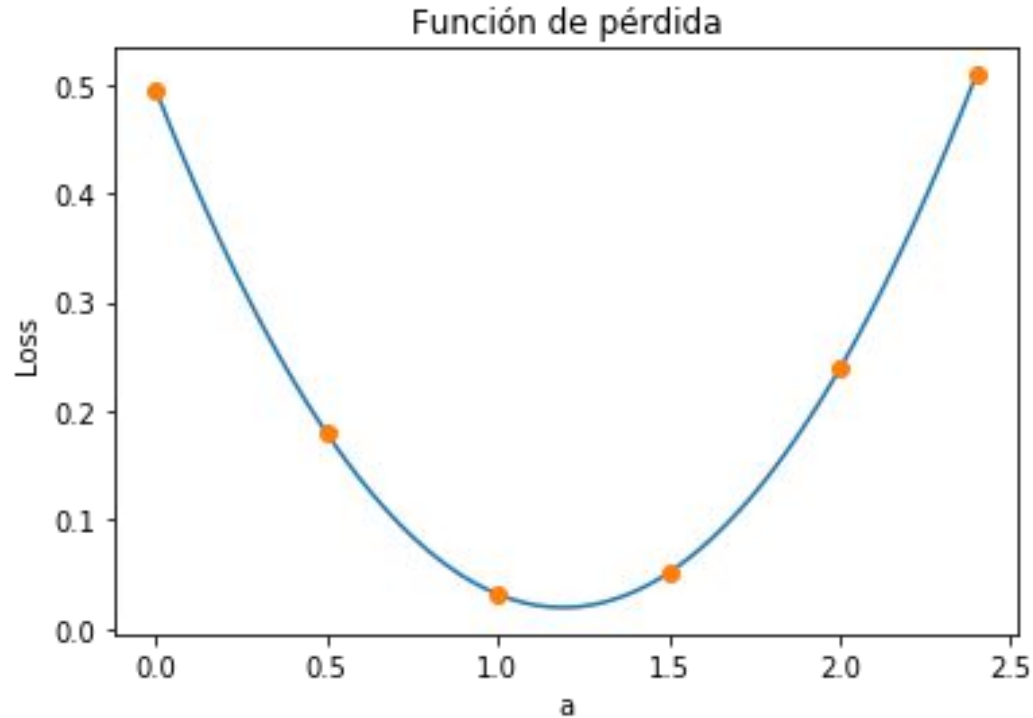
Optimización: Gradient descent



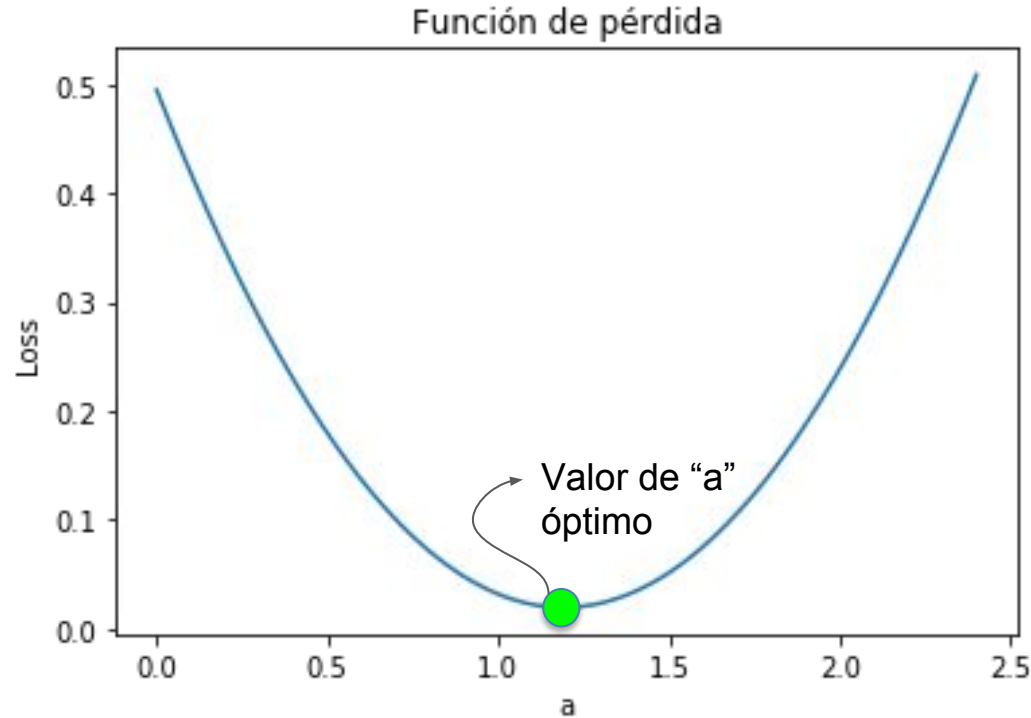
Optimización: Gradient descent



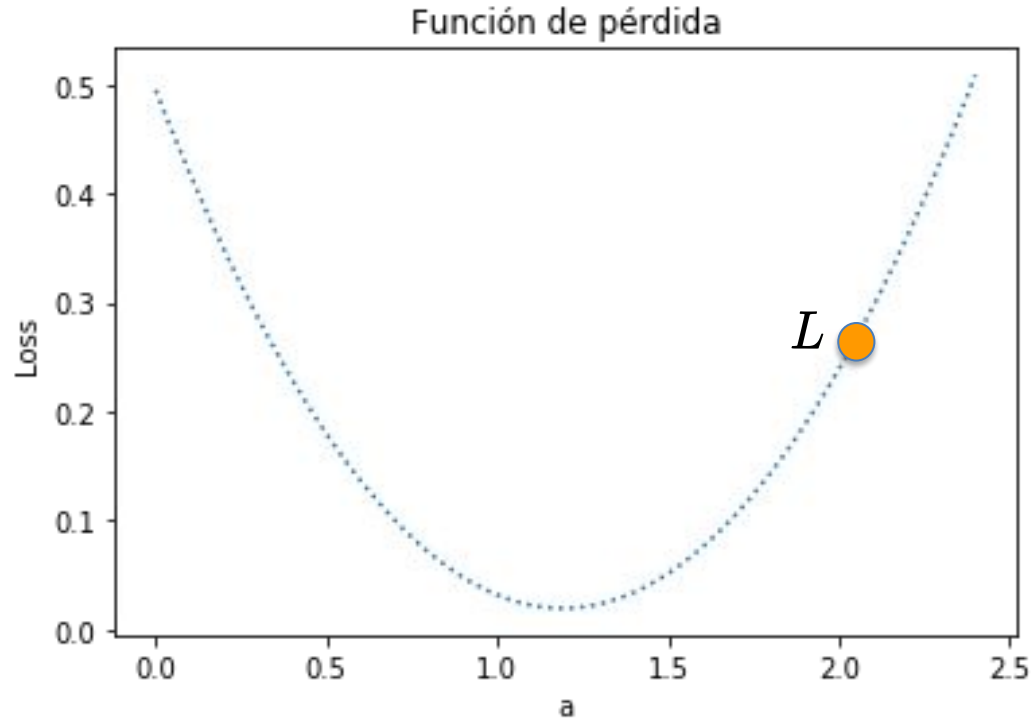
Optimización: Gradient descent



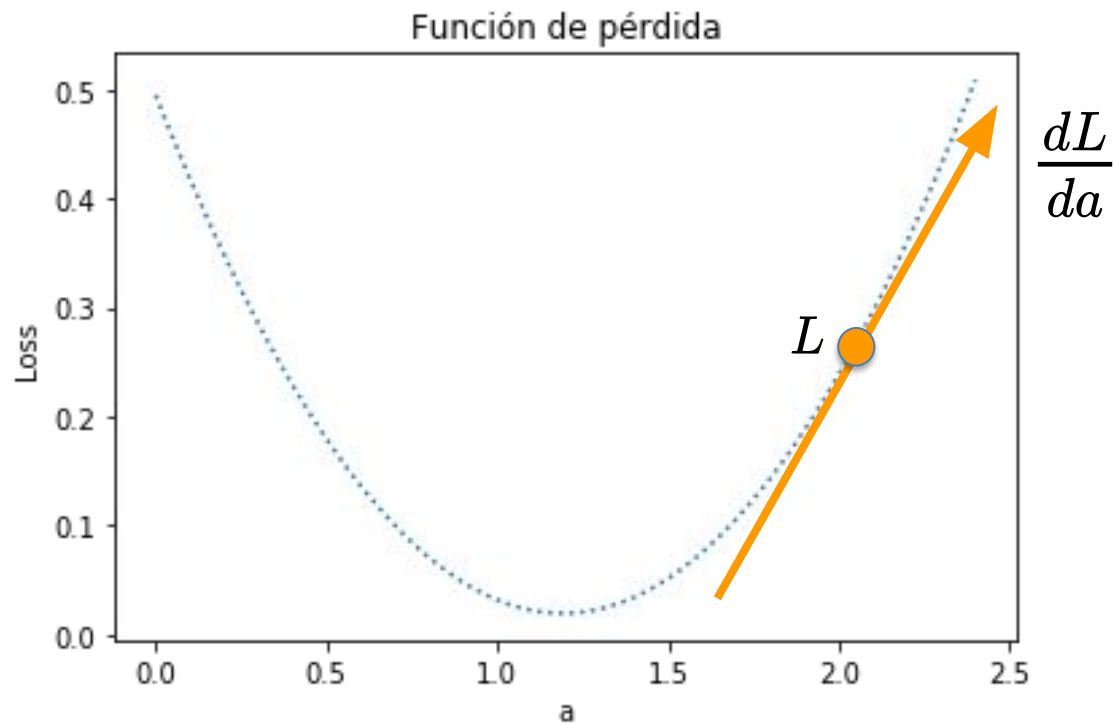
Optimización: Gradient descent



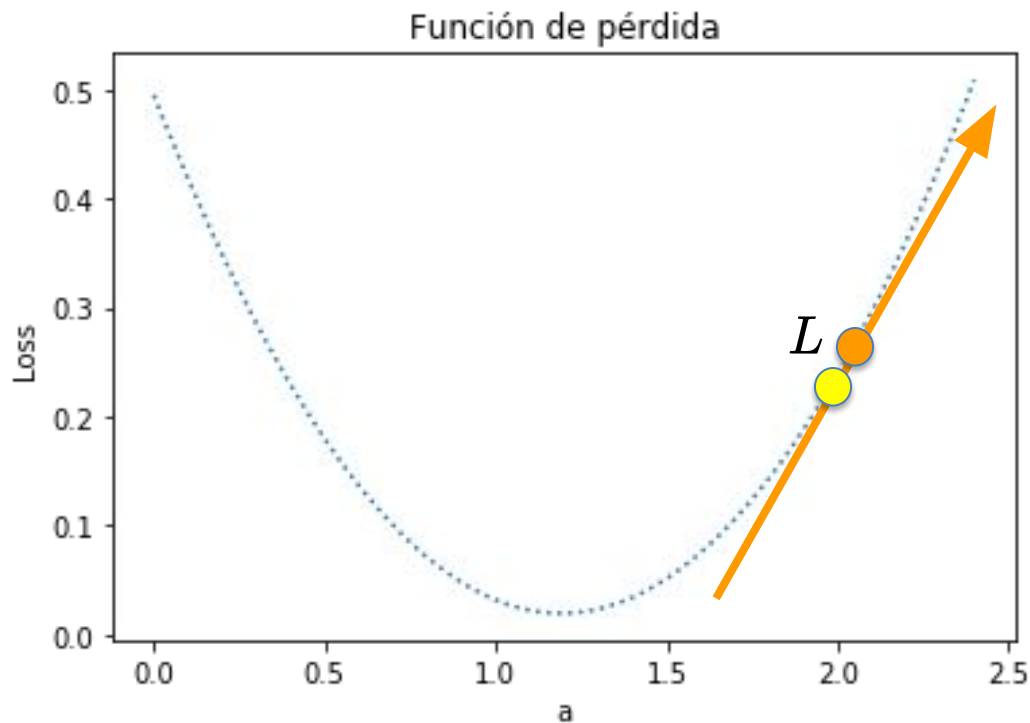
Optimización: Gradient descent



Optimización: Gradient descent



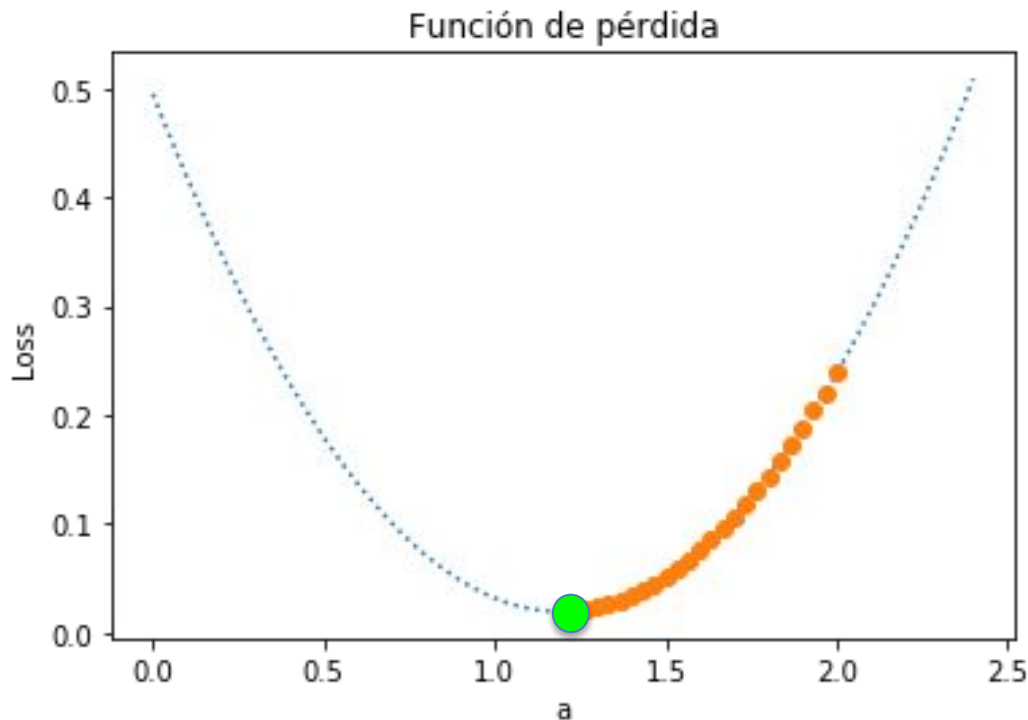
Optimización: Gradient descent



$$\frac{dL}{da}$$

$$new\ a = a - lr * \left(\frac{dL}{da} \right)$$

Optimización: Gradient descent

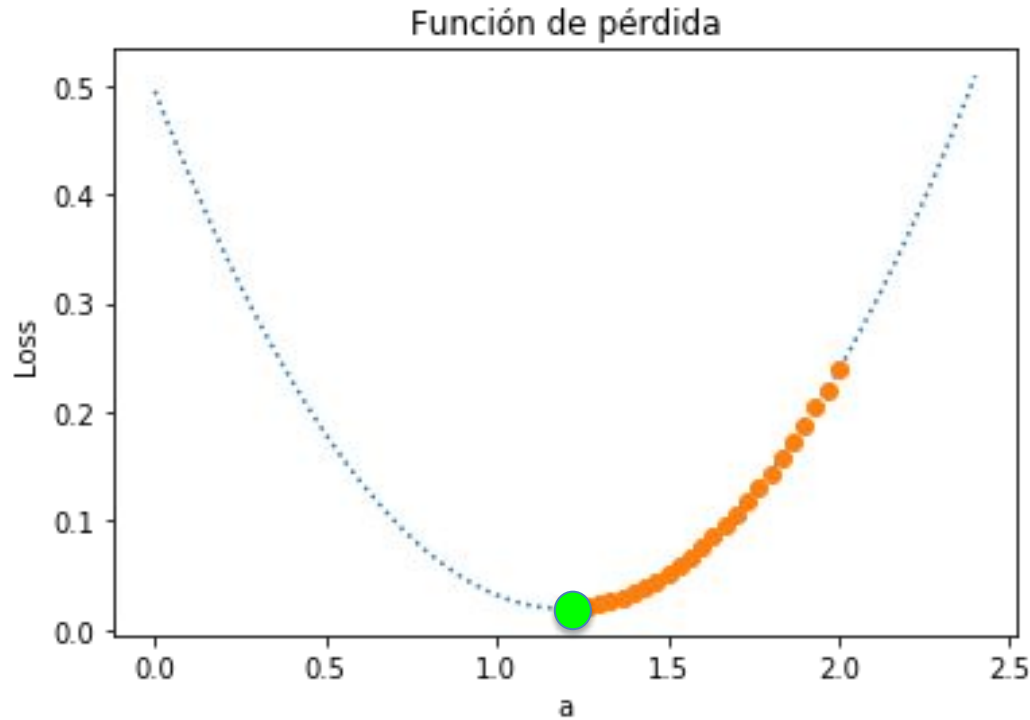


$$\frac{dL}{da}$$

$$new\ a = a - lr * \left(\frac{dL}{da} \right)$$

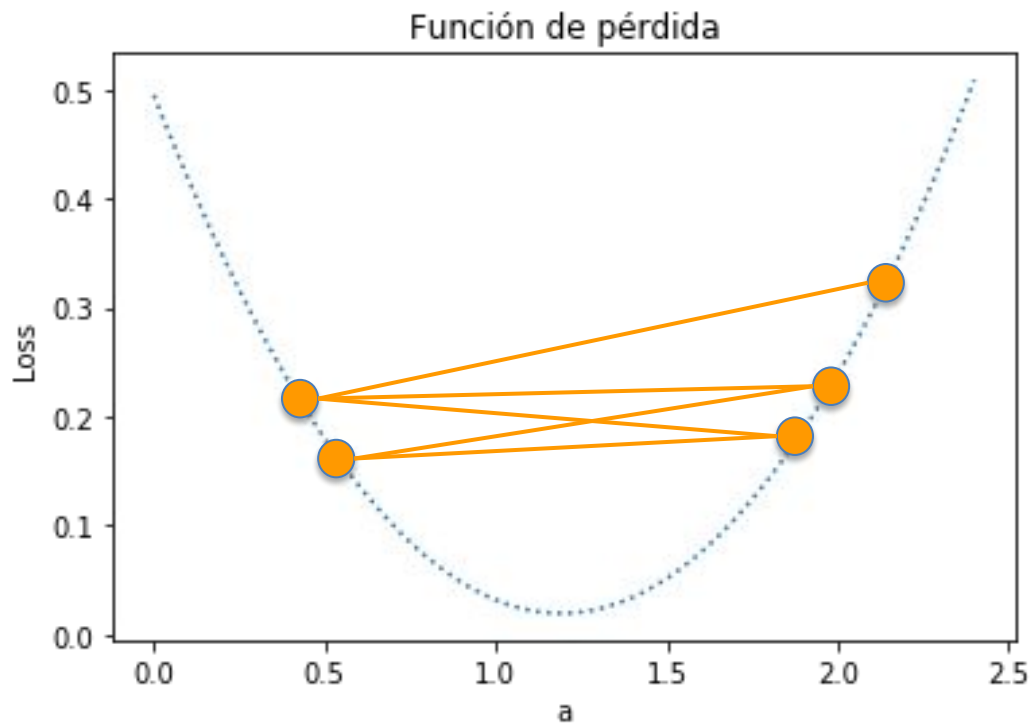
Se itera la optimización
hasta converger.

Learning rate: lr



Con un lr bajo, el proceso de optimización puede ser largo.

Learning rate: lr



Pero con un lr muy alto, llegar al punto óptimo se puede volver imposible.

Optimización: Gradient descent

$$f(x_i) = ax_i + b$$

$$L = \frac{1}{2m} \sum_{i=1}^m (f(x_i) - y_i)^2$$

Optimización: Gradient descent

$$f(x_i) = ax_i + b$$

$$L = \frac{1}{2m} \sum_{i=1}^m (f(x_i) - y_i)^2$$

$$\frac{dL}{da} = \frac{d(\frac{1}{2m} \sum_{i=1}^m (f(x_i) - y_i)^2)}{da} = \frac{1}{2m} \sum_{i=1}^m (2(f(x_i) - y_i) (\frac{d(ax_i + b)}{da})) = \sum_{i=1}^m (f(x_i) - y_i)(x_i)$$

Optimización: Gradient descent

$$f(x_i) = ax_i + b$$

$$L = \frac{1}{2m} \sum_{i=1}^m (f(x_i) - y_i)^2$$

$$\frac{dL}{da} = \frac{d(\frac{1}{2m} \sum_{i=1}^m (f(x_i) - y_i)^2)}{da} = \frac{1}{2m} \sum_{i=1}^m (2(f(x_i) - y_i) (\frac{d(ax_i + b)}{da})) = \sum_{i=1}^m (f(x_i) - y_i)(x_i)$$

$$\frac{dL}{db} = \frac{d(\frac{1}{2m} \sum_{i=1}^m (f(x_i) - y_i)^2)}{db} = \frac{1}{2m} \sum_{i=1}^m (2(f(x_i) - y_i) (\frac{d(ax_i + b)}{db})) = \sum_{i=1}^m (f(x_i) - y_i)$$

Optimización: Gradient descent

$$f(x_i) = ax_i + b$$

$$L = \frac{1}{2m} \sum_{i=1}^m (f(x_i) - y_i)^2$$

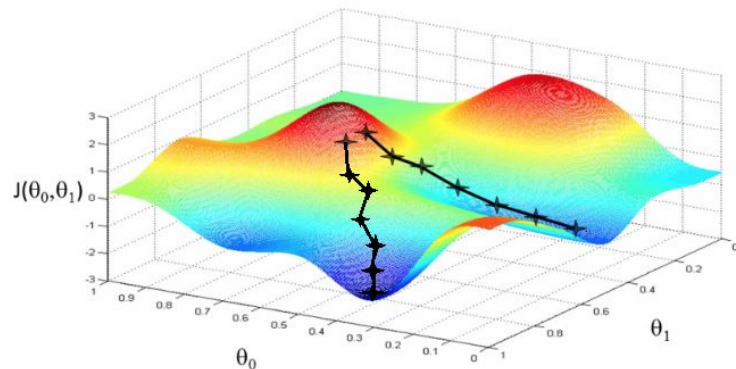
$$\frac{dL}{da} = \sum_{i=1}^m (f(x_i) - y_i)(x_i)$$

$$\frac{dL}{db} = \sum_{i=1}^m (f(x_i) - y_i)$$

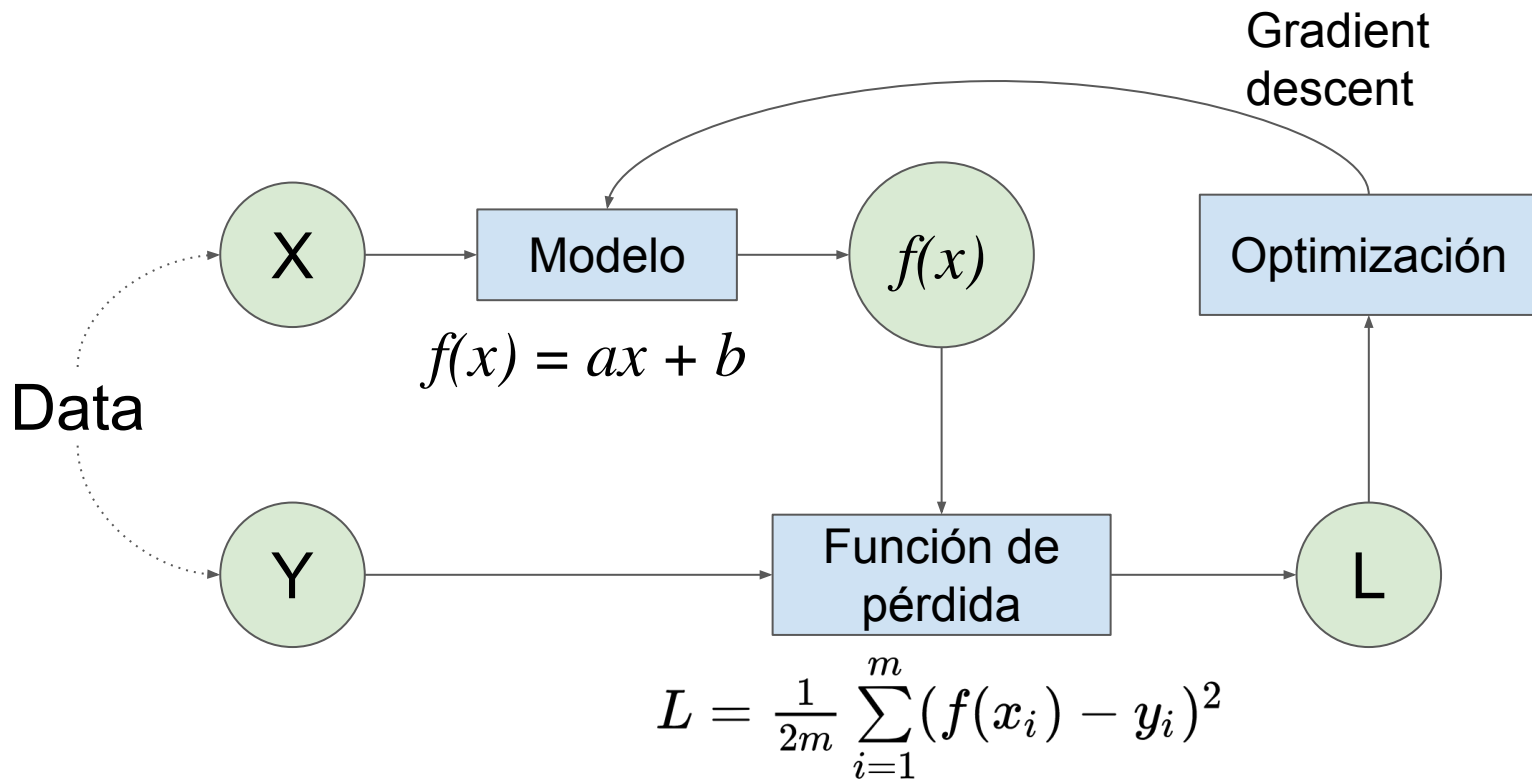
iterar :

$$new\ a = a - lr * \left(\frac{dL}{da} \right)$$

$$new\ b = b - lr * \left(\frac{dL}{db} \right)$$



Optimización: Gradient descent



Código

Herramientas

Ambiente de trabajo

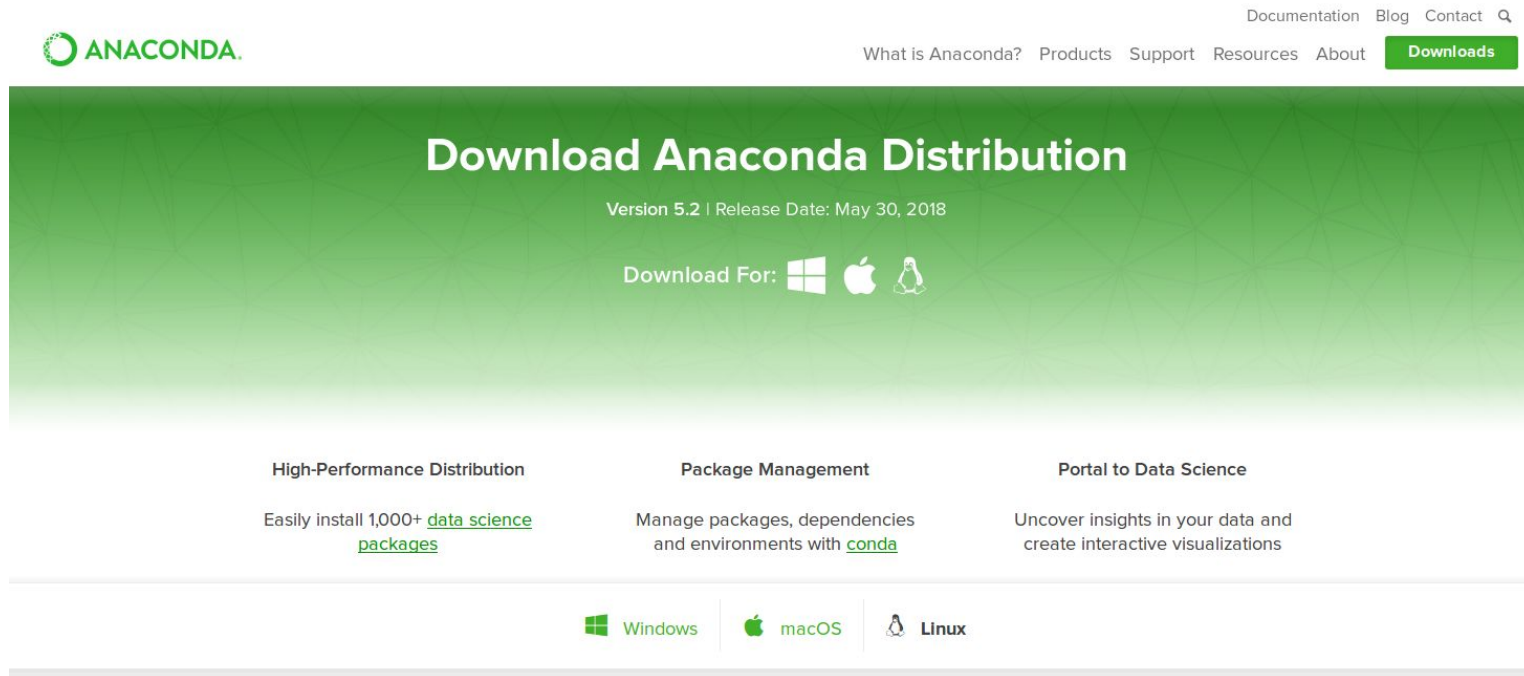


Librerías



Anaconda (package manager)

<https://www.anaconda.com/download>



The screenshot shows the Anaconda website's download page. At the top, the Anaconda logo is on the left, and navigation links for Documentation, Blog, Contact, and a search icon are on the right. Below the navigation, a green banner contains the text 'Download Anaconda Distribution' and 'Version 5.2 | Release Date: May 30, 2018'. Underneath the banner, there are three icons representing Windows, macOS, and Linux. Below this, there are three columns of text: 'High-Performance Distribution' with a link to 'data science packages', 'Package Management' with a link to 'conda', and 'Portal to Data Science'. At the bottom, there are three icons for Windows, macOS, and Linux.




ANACONDA.

Documentation Blog Contact 🔍

What is Anaconda? Products Support Resources About [Downloads](#)

Download Anaconda Distribution

Version 5.2 | Release Date: May 30, 2018

Download For:   

High-Performance Distribution


Easily install 1,000+ [data science packages](#)


Package Management


Manage packages, dependencies and environments with [conda](#)

Portal to Data Science

Uncover insights in your data and create interactive visualizations

 Windows

 macOS

 Linux

Anaconda (package manager)

Listar librerías instaladas:

```
> conda list
```

Instalar librería:

```
> conda install scikit-learn
```

Desinstalar:

```
> conda uninstall scikit-learn
```

Actualizar librerías:

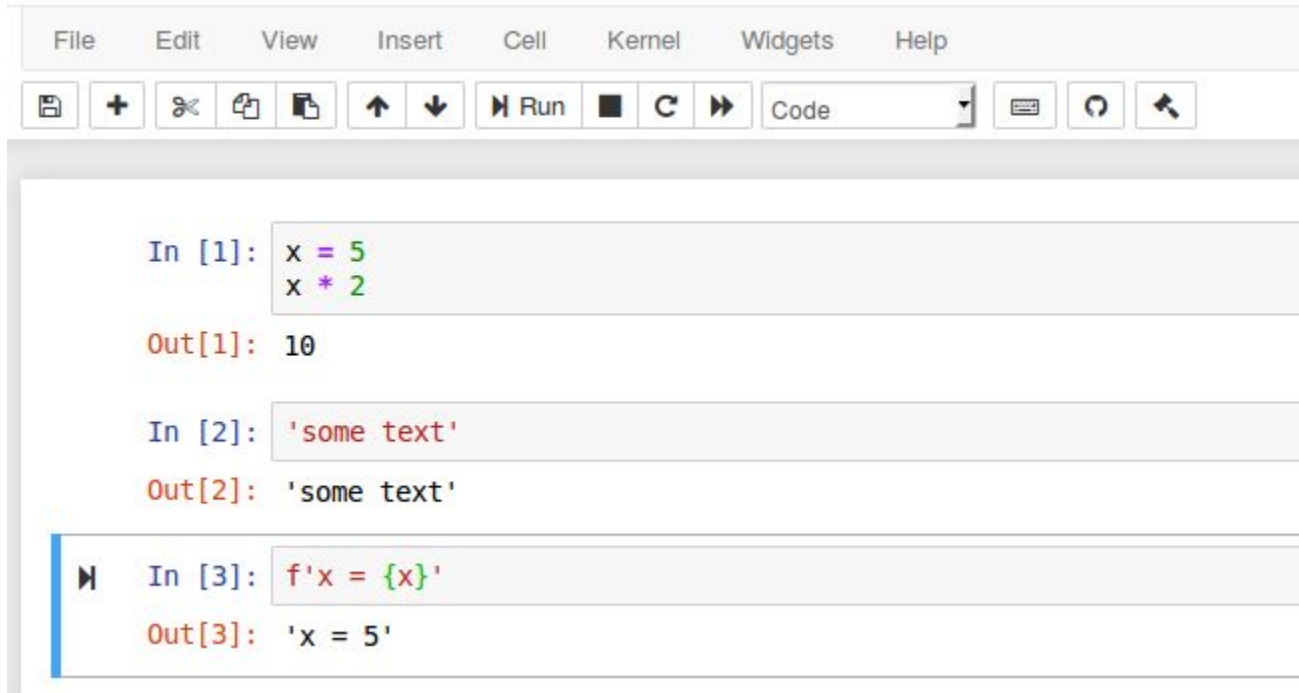
```
> conda update --all
```

Ambiente de trabajo

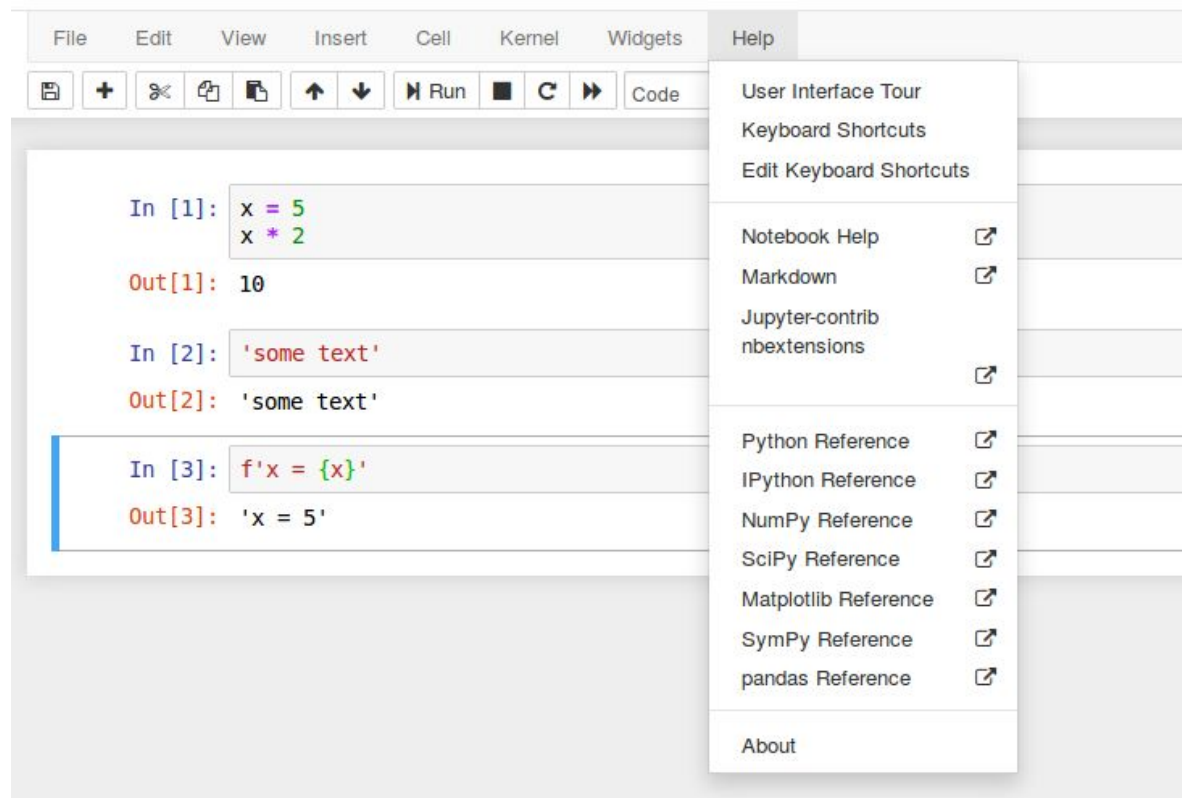
```
Iniciar jupyter:  
> jupyter-notebook
```



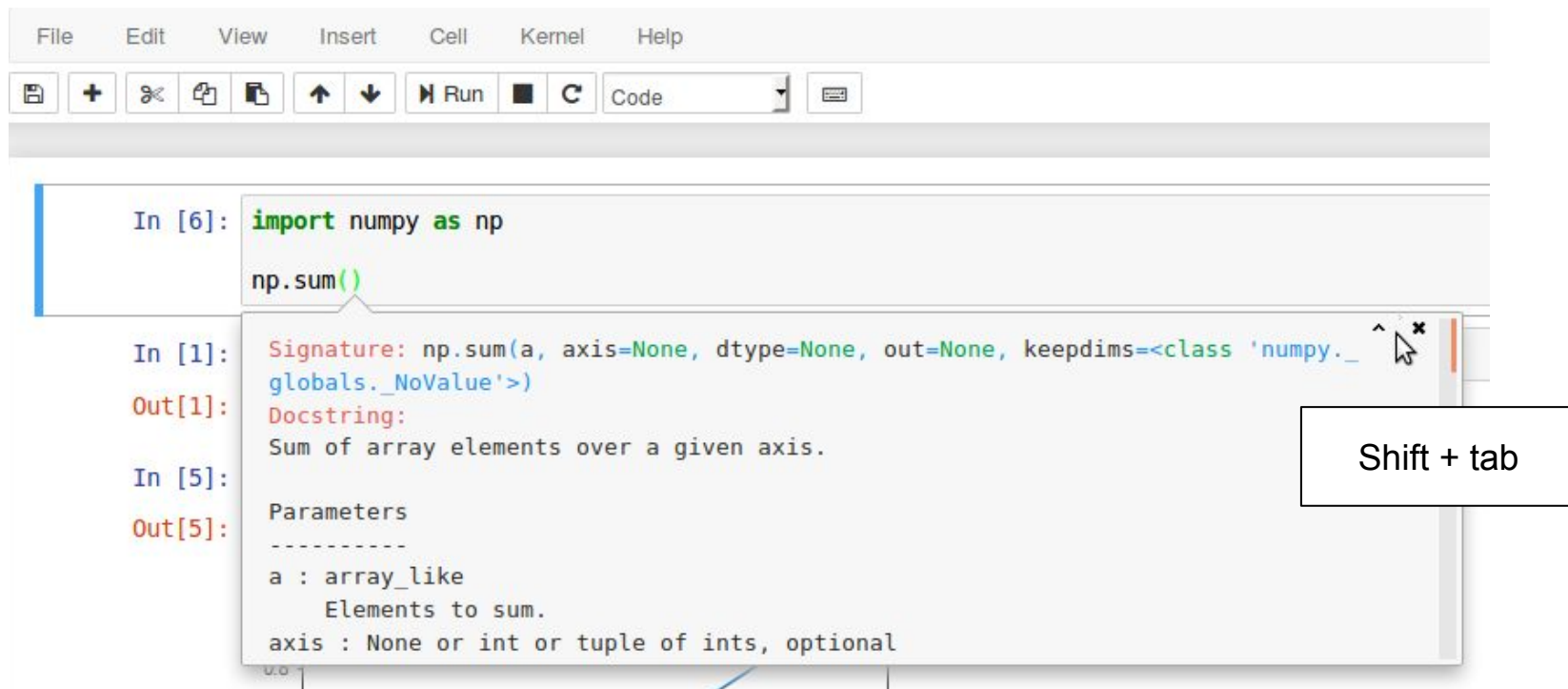
Jupyter notebook



Jupyter notebook



Ver documentación



The image shows a Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar with icons for saving, adding cells, cutting, copying, pasting, navigating, running, and clearing. The main area displays a code cell with the following content:

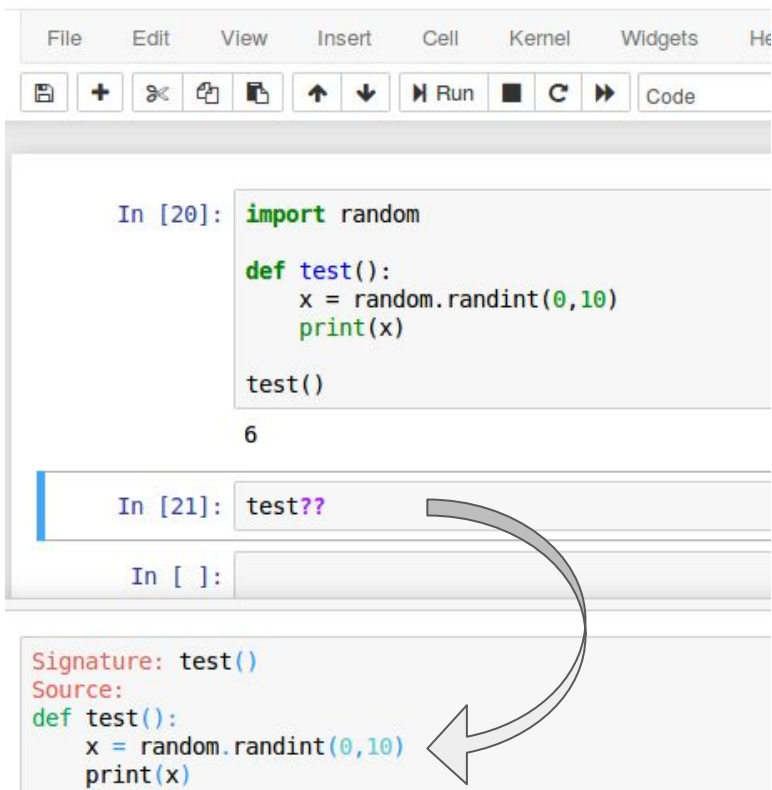
```
In [6]: import numpy as np  
np.sum()
```

A tooltip is visible over the `np.sum()` function call, displaying the following information:

- Signature:** `np.sum(a, axis=None, dtype=None, out=None, keepdims=<class 'numpy._globals._NoValue'>)`
- Docstring:** Sum of array elements over a given axis.
- Parameters:**
 - `a` : array_like
Elements to sum.
 - `axis` : None or int or tuple of ints, optional

A callout box on the right side of the tooltip indicates the keyboard shortcut **Shift + tab** used to toggle the documentation display.

Ver código fuente



The screenshot shows a Jupyter Notebook interface. At the top is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. Below the menu is a toolbar with icons for saving, adding cells, undo, redo, and running code. The main area contains two code cells. The first cell, labeled 'In [20]:', contains the following Python code:

```
import random

def test():
    x = random.randint(0,10)
    print(x)

test()
```

 Below this code, the output '6' is displayed. The second cell, labeled 'In [21]:', contains the text 'test?'. Below this, a third cell labeled 'In []:' is partially visible. At the bottom of the notebook, a tooltip window is open, showing the signature 'Signature: test()' and the source code of the 'test' function:

```
def test():
    x = random.randint(0,10)
    print(x)
```

 A curved arrow points from the 'test??' input in the second cell to the tooltip window.

```
In [20]: import random

def test():
    x = random.randint(0,10)
    print(x)

test()

6

In [21]: test??

In [ ]:
```

Signature: test()
Source:
def test():
 x = random.randint(0,10)
 print(x)

Notebook

Jupyter extensions

```
> conda install -c conda-forge jupyter_contrib_nbextensions
```

Files Running Clusters **Nbextensions**

Select items to perform actions on them. Upload New ▾ ↺

☐ 0 ▾ / github / pucp / winter-school-2018 / semana_1_machine_learning Name ▾ Last Modified File size

<input type="checkbox"/>	..	seconds ago	
<input type="checkbox"/>	data	30 minutes ago	
<input type="checkbox"/>	Images	8 hours ago	
<input type="checkbox"/>	 dia_1_arboles_de_decision.ipynb	2 minutes ago	198 kB
<input type="checkbox"/>	utils.py	6 months ago	5.53 kB

Jupyter extensions

Files Running Clusters Nbextensions

↻

Configurable nbextensions

☒ disable configuration for nbextensions without explicit compatibility (they may break your notebook environment, but can be useful to show for nbextension development)

filter: by description, section, or tags

<input type="checkbox"/> (some) LaTeX environments for Jupyter	<input type="checkbox"/> 2to3 Converter	<input checked="" type="checkbox"/> AddBefore	<input checked="" type="checkbox"/> Autopep8
<input type="checkbox"/> AutoSaveTime	<input checked="" type="checkbox"/> Autoscroll	<input type="checkbox"/> Cell Filter	<input type="checkbox"/> Code Font Size
<input type="checkbox"/> Code prettify	<input type="checkbox"/> Codefolding	<input type="checkbox"/> Codefolding in Editor	<input checked="" type="checkbox"/> CodeMirror mode extensions
<input checked="" type="checkbox"/> Collapsible Headings	<input type="checkbox"/> Comment/Uncomment Hotkey	<input checked="" type="checkbox"/> contrib_nbextensions_help_item	<input type="checkbox"/> datestamper
<input type="checkbox"/> Equation Auto Numbering	<input type="checkbox"/> ExecuteTime	<input type="checkbox"/> Execution Dependencies	<input type="checkbox"/> Exercise
<input type="checkbox"/> Exercise2	<input type="checkbox"/> Export Embedded HTML	<input type="checkbox"/> Freeze	<input checked="" type="checkbox"/> Gist-it
<input type="checkbox"/> Help panel	<input type="checkbox"/> Hide Header	<input type="checkbox"/> Hide input	<input type="checkbox"/> Hide input all
<input type="checkbox"/> Highlight selected word	<input type="checkbox"/> highlighter	<input type="checkbox"/> Hinterland	<input type="checkbox"/> Initialization cells
<input type="checkbox"/> isort formatter	<input checked="" type="checkbox"/> jupyter-js-widgets/extension	<input checked="" type="checkbox"/> jupyter-vega3/index	<input type="checkbox"/> Keyboard shortcut editor
<input checked="" type="checkbox"/> Launch QTConsole	<input type="checkbox"/> Limit Output	<input type="checkbox"/> Live Markdown Preview	<input type="checkbox"/> Load TeX macros
<input type="checkbox"/> Move selected cells	<input type="checkbox"/> Navigation-Hotkeys	<input checked="" type="checkbox"/> Nbextensions dashboard tab	<input checked="" type="checkbox"/> Nbextensions edit menu item
<input type="checkbox"/> nbTranslate	<input type="checkbox"/> Notify	<input type="checkbox"/> Printview	<input type="checkbox"/> Python Markdown
<input type="checkbox"/> Rubberband	<input type="checkbox"/> Ruler	<input type="checkbox"/> Runtools	<input type="checkbox"/> Scratchpad
<input type="checkbox"/> ScrollDown	<input type="checkbox"/> Select CodeMirror Keymap	<input type="checkbox"/> SKILL Syntax	<input checked="" type="checkbox"/> Skip-Traceback
<input type="checkbox"/> Snippets	<input type="checkbox"/> Snippets Menu	<input type="checkbox"/> spellchecker	<input type="checkbox"/> Split Cells Notebook
<input checked="" type="checkbox"/> Table of Contents (2)	<input type="checkbox"/> table_beautifier	<input type="checkbox"/> Toggle all line numbers	<input type="checkbox"/> Tree Filter
<input type="checkbox"/> Variable Inspector	<input type="checkbox"/> zenmode		



Contents ↻ ⚙

- 1 Generamos data
- 2 Definimos el modelo
- 3 Función de pérdida
- ▼ 4 Optimización
 - 4.1 Gradients
 - 4.2 Gradient descent
- 5 Ejercicio 1: Casas Simple
- 6 Multiple linear regression
- 7 Ejercicio 2: Casas Multi
- 8 Usando matrices
- 9 Ejercicio 3: Diabetes Dataset

Algunos hotkeys útiles en jupyter:

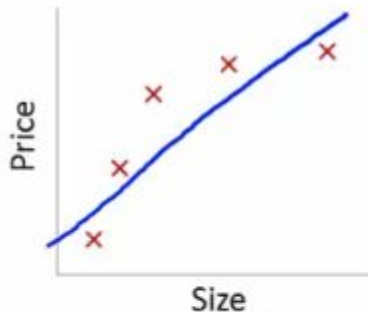
- **control+enter**: correr celda.
- **enter**: editar celda.
- **escape**: salir del modo edición.
- **control+s**: guardar notebook.

Fuera del modo edición se pueden utilizar los siguientes hotkeys:

- **a**: agregar celda arriba.
- **b**: agregar celda abajo.
- **x**: borrar celda.
- **z**: deshacer borrar celda.
- **y**: setear celda en modo python (código).
- **m**: setear celda en modo markdown (texto).
- **o**: mostrar/ocultar resultado.

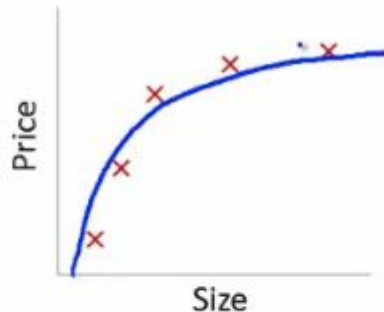
Validación de modelos

Overfitting



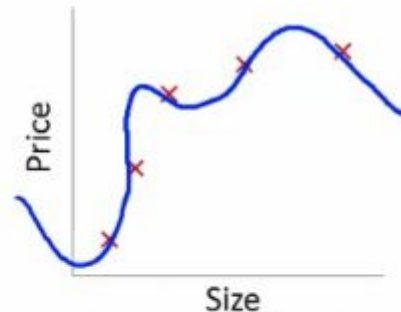
$$\theta_0 + \theta_1 x$$

High bias
(underfit)



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

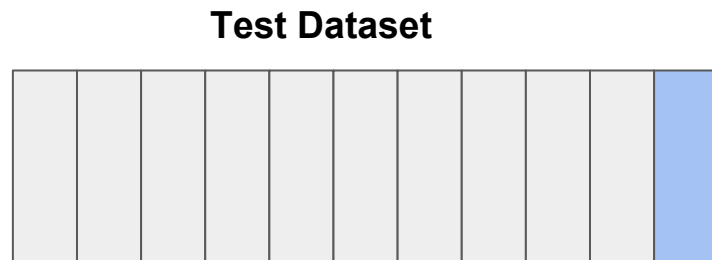
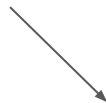
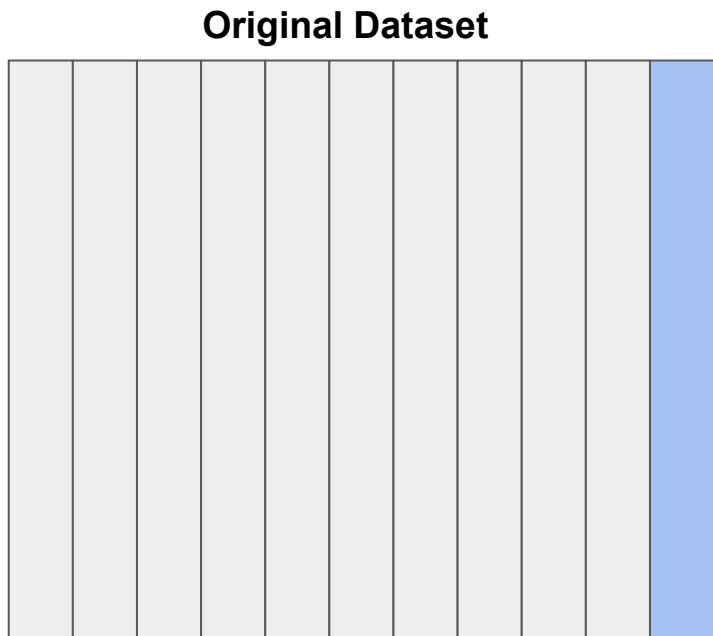
"Just right"



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

High variance
(overfit)

Train and test split



Even better: Train, validation and test split

