

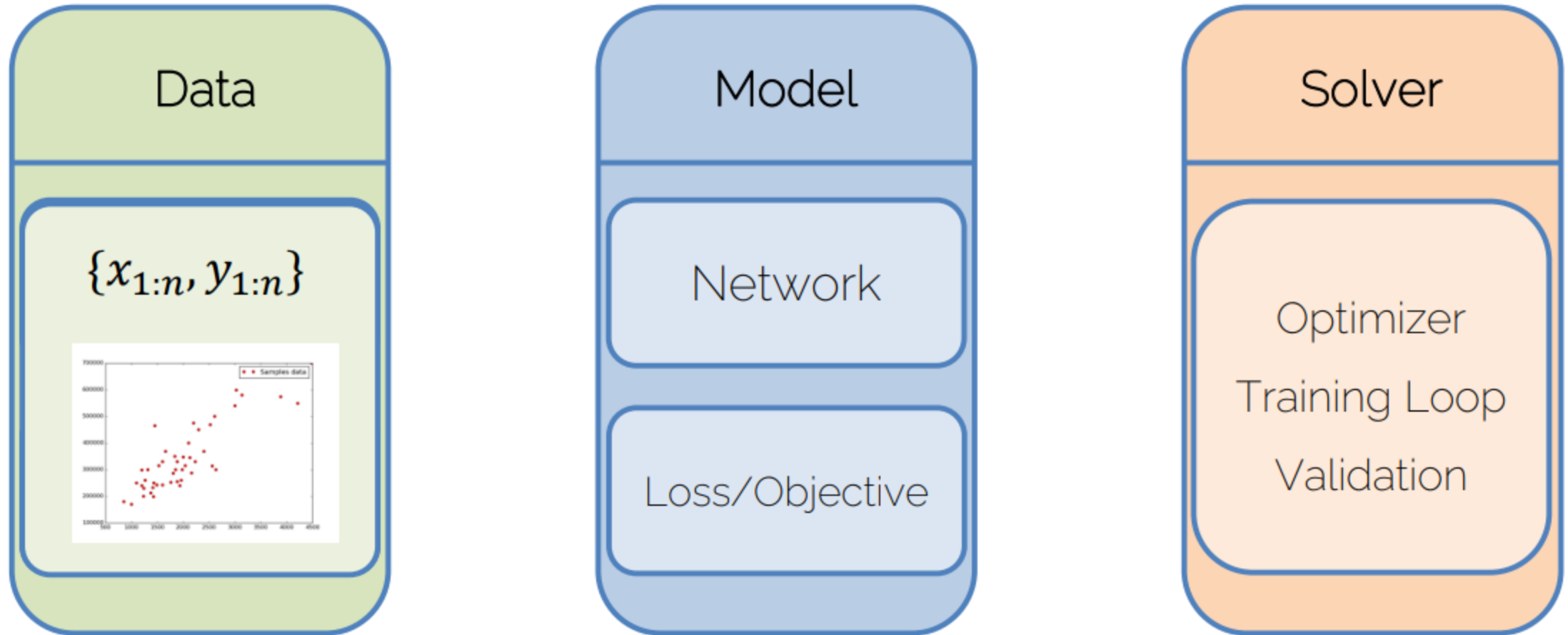


Tutorial 3

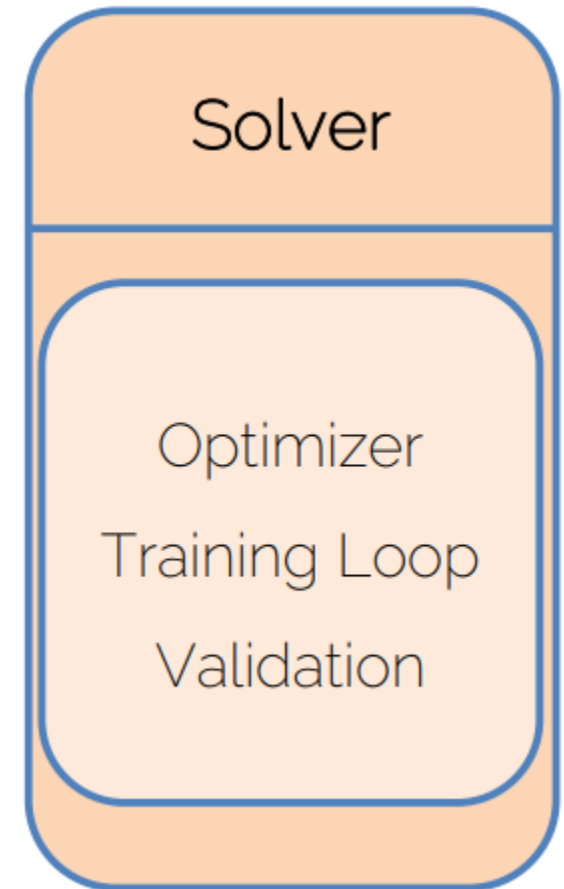
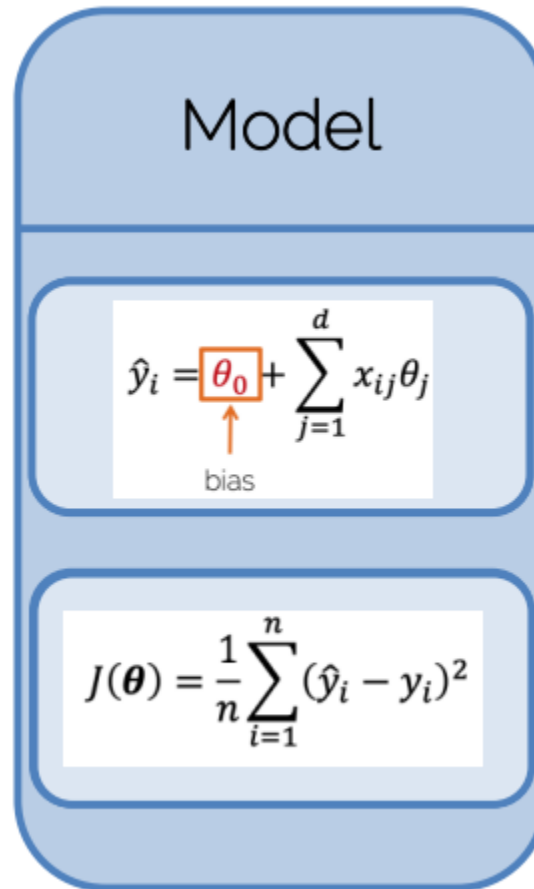
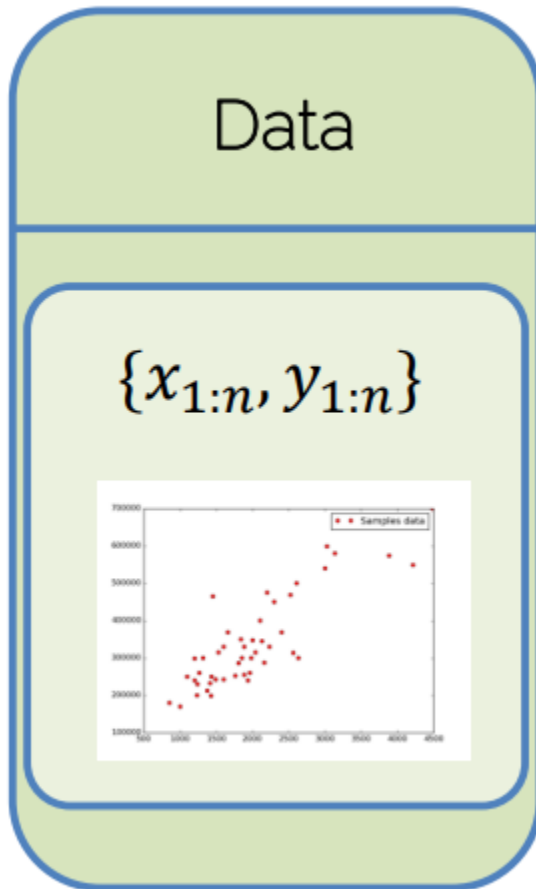


Descripción General del Ejercicio

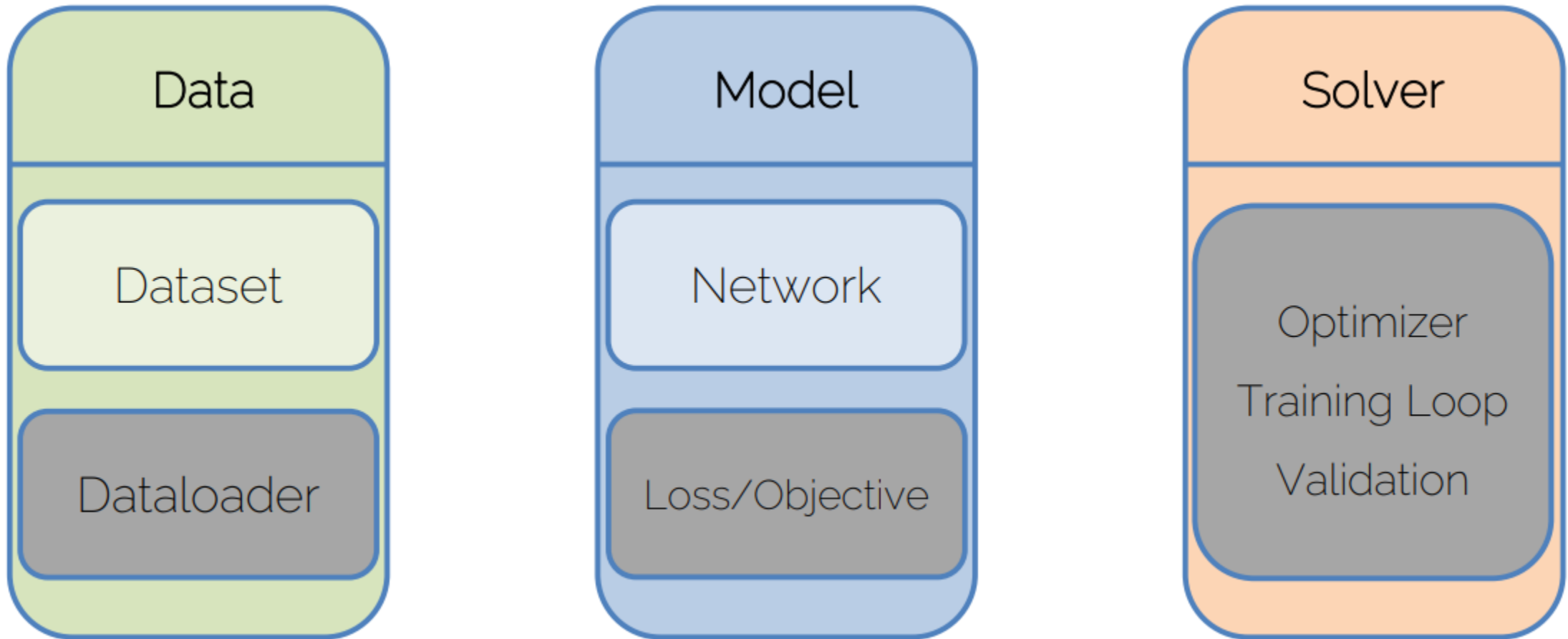
Pilares del Deep Learning



Pilares del Deep Learning



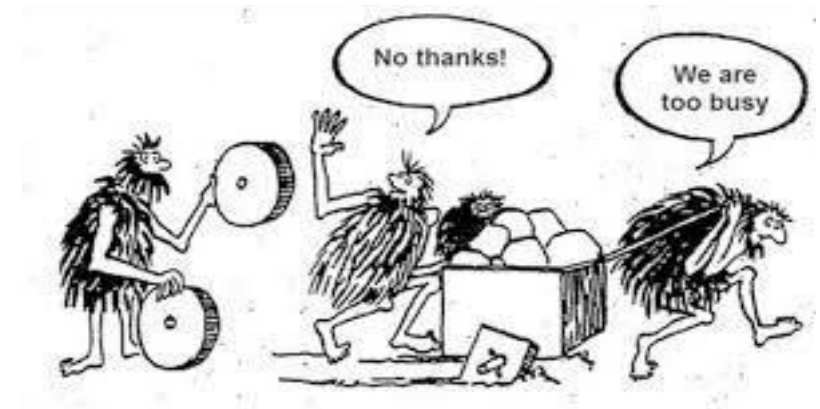
Pilares del Deep Learning



Objetivos para los ejercicios 3-5

Exercise 03: Dataset and Dataloader
Exercise 04: Solver and Linear Regression
Exercise 05: Neural Networks
Exercise 06: Hyperparameter Tuning

Numpy
(Reinvent the wheel)





Ejercicio 3

Ejercicio 3: Dataset

- Lee datos y proporciona una forma sencilla de acceder a ellos
- Preprocesamiento y data augmentation sobre la marcha
- Preprocesamiento: por ejemplo, escalar la imagen a un tamaño fijo
- Augmentation: por ejemplo, giros aleatorios de imágenes, recortes, etc.

Ejemplo: Image Classification Dataset

- Dado: 1 carpeta con 10 sub-carpetas
- Cada sub-carpeta contiene X imágenes para una determinada categoría



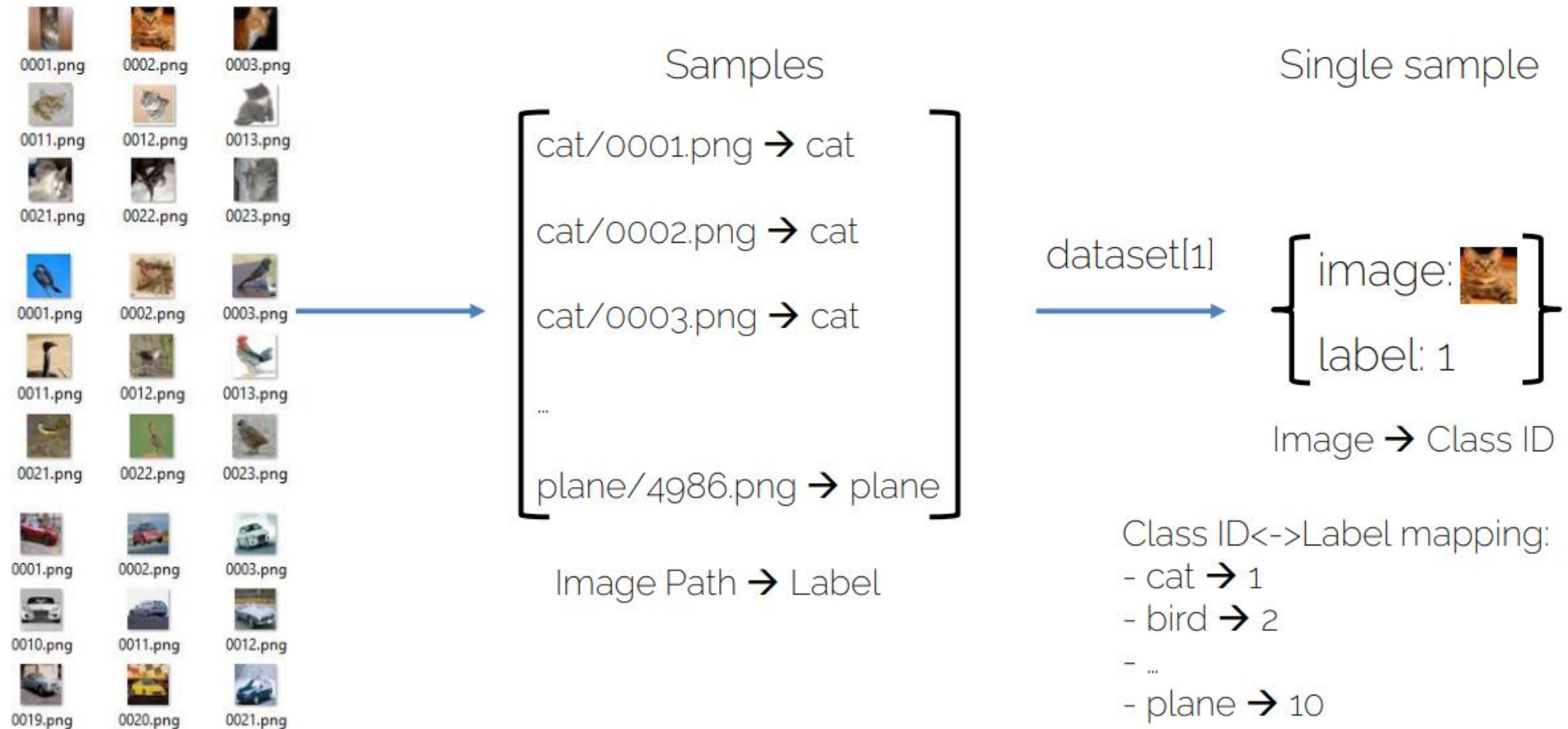
Ejemplo: Image Classification Dataset

- La clase Dataset lee la estructura de esa carpeta
 - ImageDataset(<dataset_root>)
 - > samples = [("cat/0001.png", 1), ..., ("plane/4986.png", 10)]
 - Normalmente, ¡todavía no abre las imágenes!
 - Definir la asignación de clase ID<->etiqueta
- Al acceder/llamar a la clase del conjunto de datos con un índice se obtiene un elemento único:
 - Lee la imagen del disco
 - Preprocesamiento sobre la marcha
 - Realiza augmentation

Ejemplo: Image Classification Dataset

Dataset creation

Accessing an element



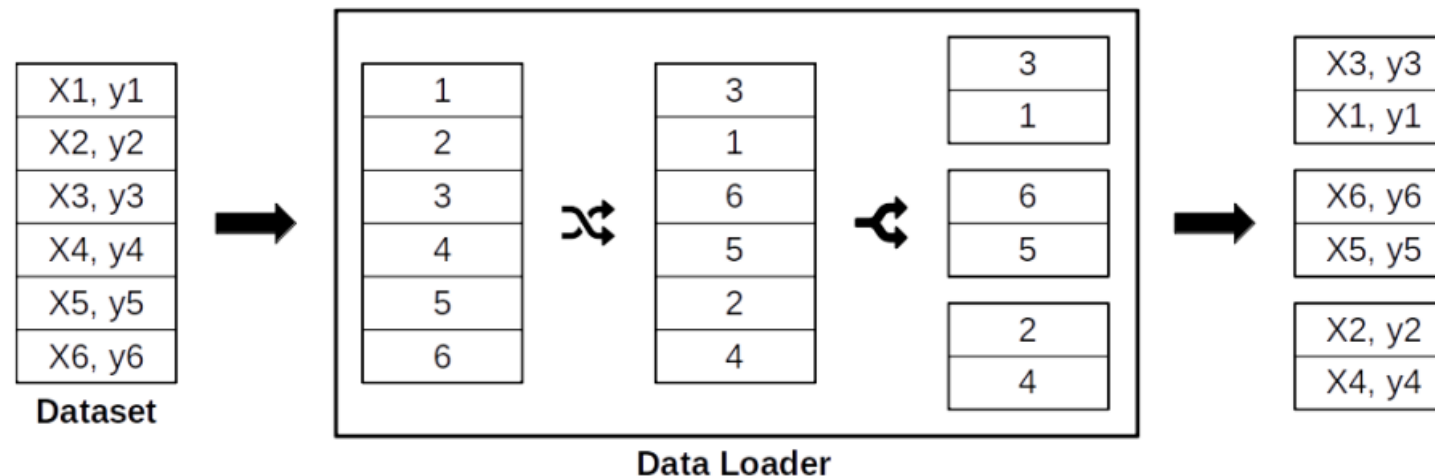


Ejercicio 3: Dataset

- Lectura de cada archivo del disco uno a uno frente a la carga de todo el conjunto de datos en memoria
 - Normalmente, los conjuntos de datos son demasiado grandes para cargarlos por completo en memoria, pero proporciona un aumento excepcional del rendimiento cuando es aplicable

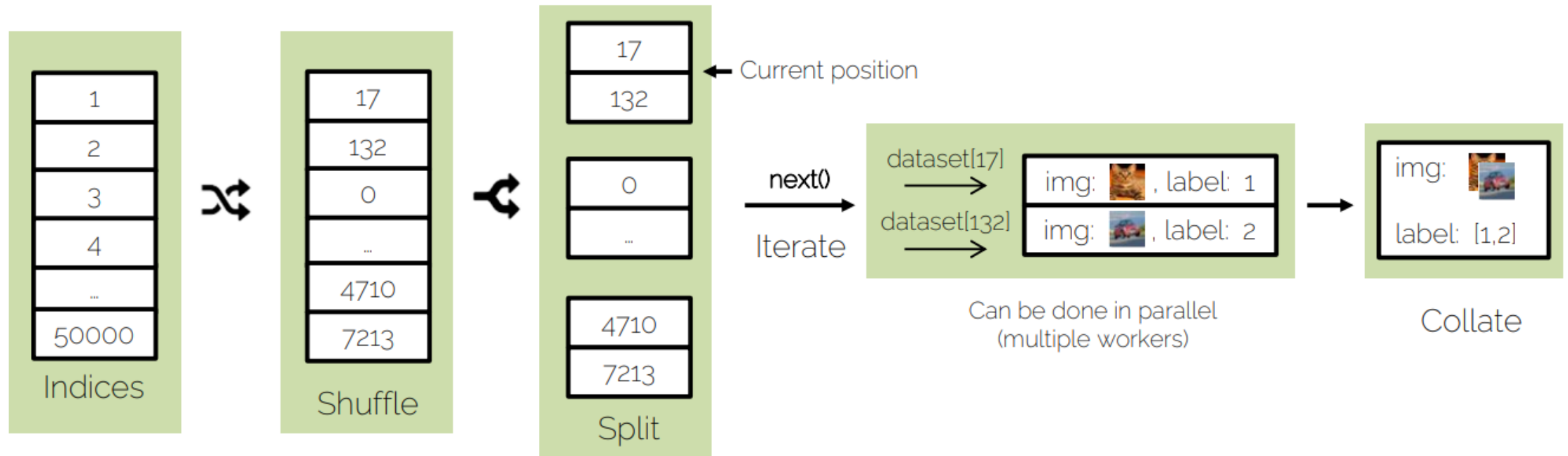
Ejercicio 3: Dataloader

- Define cómo cargar el conjunto de datos para el entrenamiento del modelo
 - Por ejemplo, número de imágenes por lote, número de trabajadores
- Mezcla el conjunto de datos
- Divide el conjunto de datos en pequeños subconjuntos: (mini) batches



Ejercicio 3: Dataloader – Iterator & Batching

- Dataloader es un “iterator”, no una lista
- No se puede acceder directamente con un índice: `dataloader[9]`
- En su lugar, itera utilizando "next" para obtener el siguiente elemento: `next(dataloader)`
 - La función `iter ()` utiliza "yield" en lugar de "return".
- Devuelve un (mini-) batch de muestras en un formato por lotes





Resumen Ejercicio 3

- Dos notebooks
 - Dataset: CIFAR10
 - Dataloader
- Submission
 1. Implementar solución en ambos notebooks
 2. Un solo zip es creado en el Dataloader notebook



Nos vemos el próximo lunes 😊