Introduction to Deep Learning (I2DL)

Tutorial 3: Data

Reminder

- Use Piazza for general and private questions
  - Do not email us personally!
- Office hours started last week
  - Find schedule on Piazza
- Solutions to the exercises
  - Will be published together with the following exercises

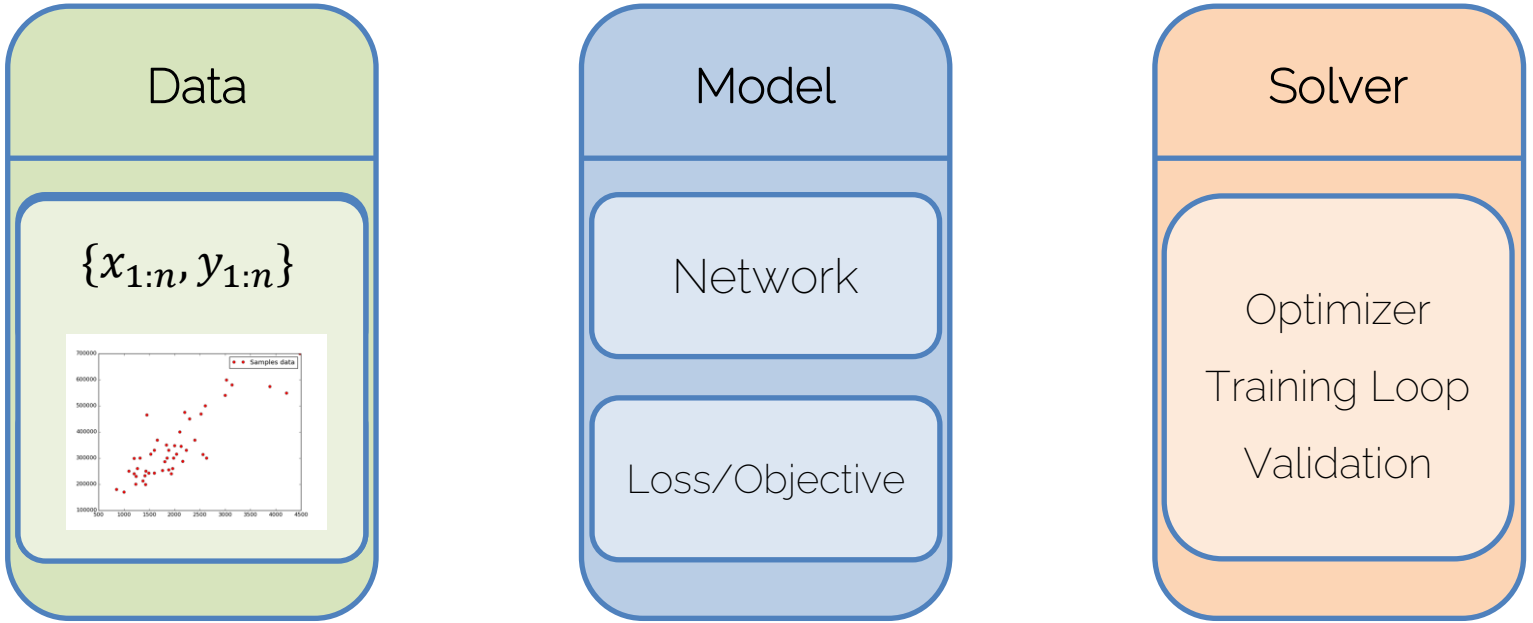
Today's Outline

- Exercise outline
  - Pillars of Deep Learning
  - Reinvent the wheel

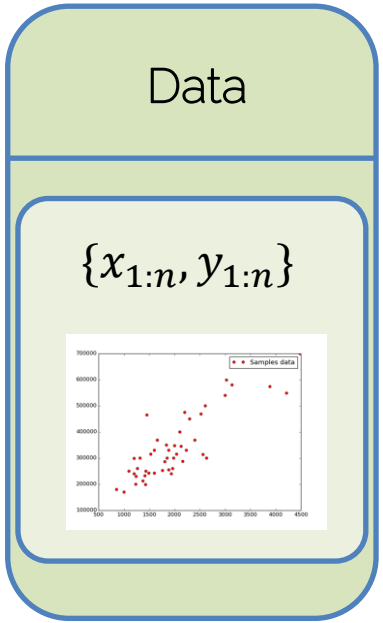
- Contents of
  - Example Datasets & -loader
  - Exercise 3 (Submission #2)

General Exercise Overview

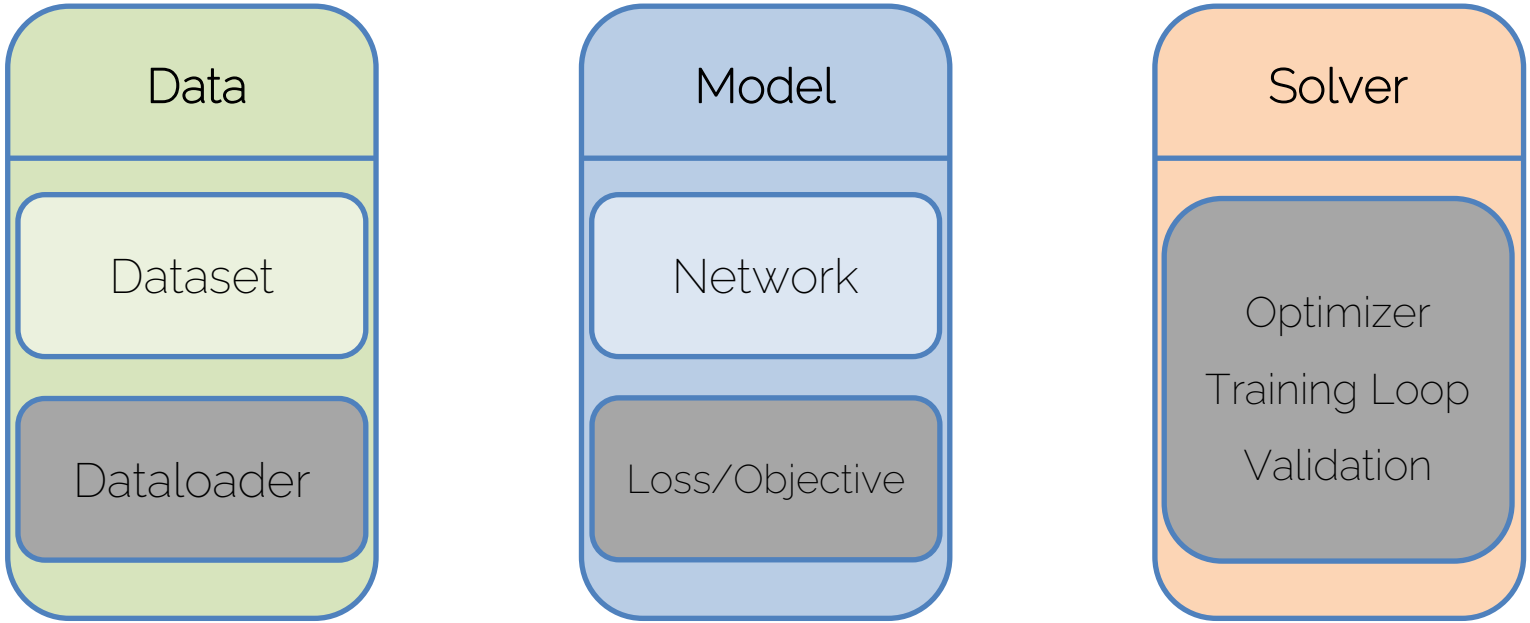
The Pillars of Deep Learning



The Pillars of Deep Learning



The Pillars of Deep Learning



Can be implemented once and used in multiple projects

Your task for exercise 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)

- Implementation of
  - A simple dataset and data loading
  - Regression/classification pipeline using Neural Networks

Exercise 3

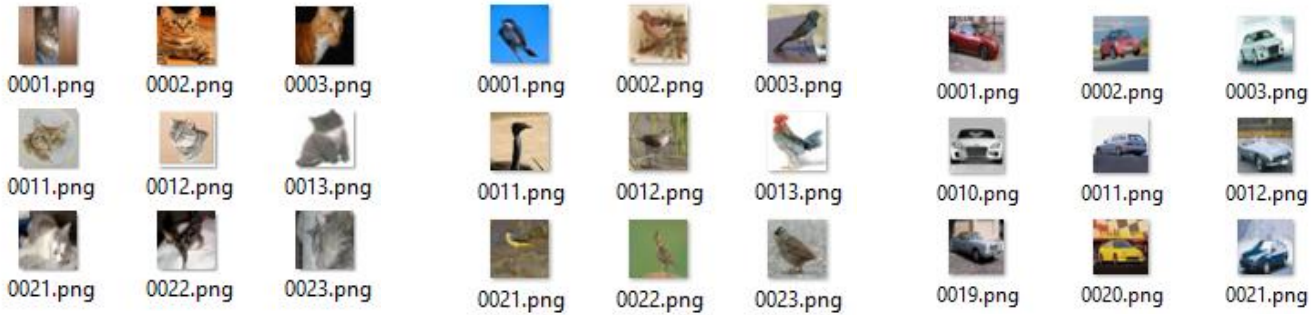
Exercise 3: Dataset

- Reads data and provides a simple way to access it

- Performs on-the-fly data preprocessing / augmentations
  - Preprocessing: e.g. scale image to fixed size
  - Augmentations: e.g. random image flips, crops, etc.

Example: Image Classification Dataset

- Given: Path to a folder with 10 sub-folders
  - <dataset_root>
    |- cat
    |- bird
    |- car
    |- …
- Each folder contains X images of a specific category



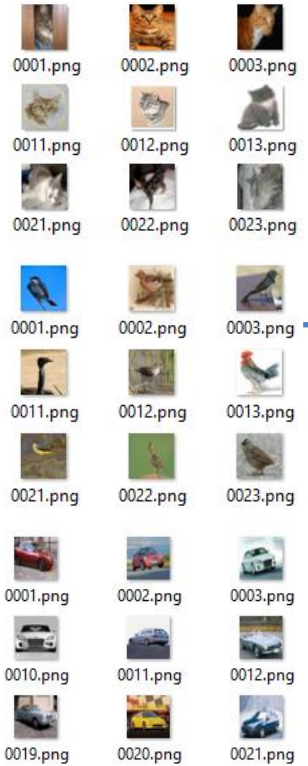
Example: Image Classification Dataset

- Dataset class reads structure of that folder
  - ImageDataset(<dataset_root>)
    → samples = [(“cat/0001.png”, 1), …, (“plane/4986.png”, 10)]
  - Usually, it does not open the images yet!
  - Define class ID<->label mapping
- Accessing/calling the dataset class with an index gives a single element:
  - Reads image from disk
  - Performs on-the-fly preprocessing
  - Preforms augmentations

Example: Image Classification Dataset

Dataset creation

Accessing an element



Samples

cat/0001.png → cat

cat/0002.png → cat

cat/0003.png → cat

…

plane/4986.png → plane

Image Path → Label

dataset[1]

Single sample

image:

label: 1

Image → Class ID

Class ID<->Label mapping:
- cat → 1
- bird → 2
- …
- plane → 10

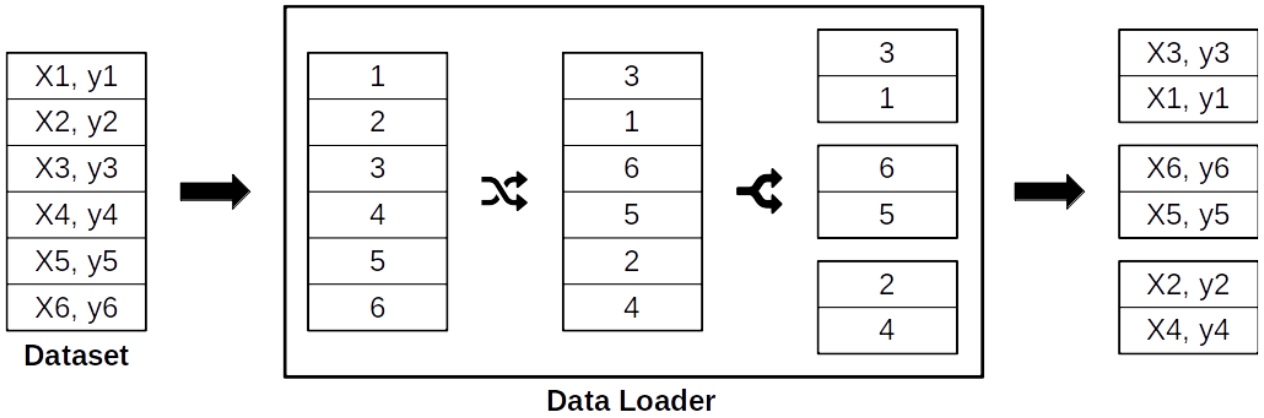
Exercise 3: Dataset

- What we excluded
  - Low level "scripting" details using operating system calls

- Reading every file from disk one-by-one vs loading the entire dataset into memory
  - Usually, datasets are too big to load entirely into memory, but it provides exceptional performance boosts when applicable

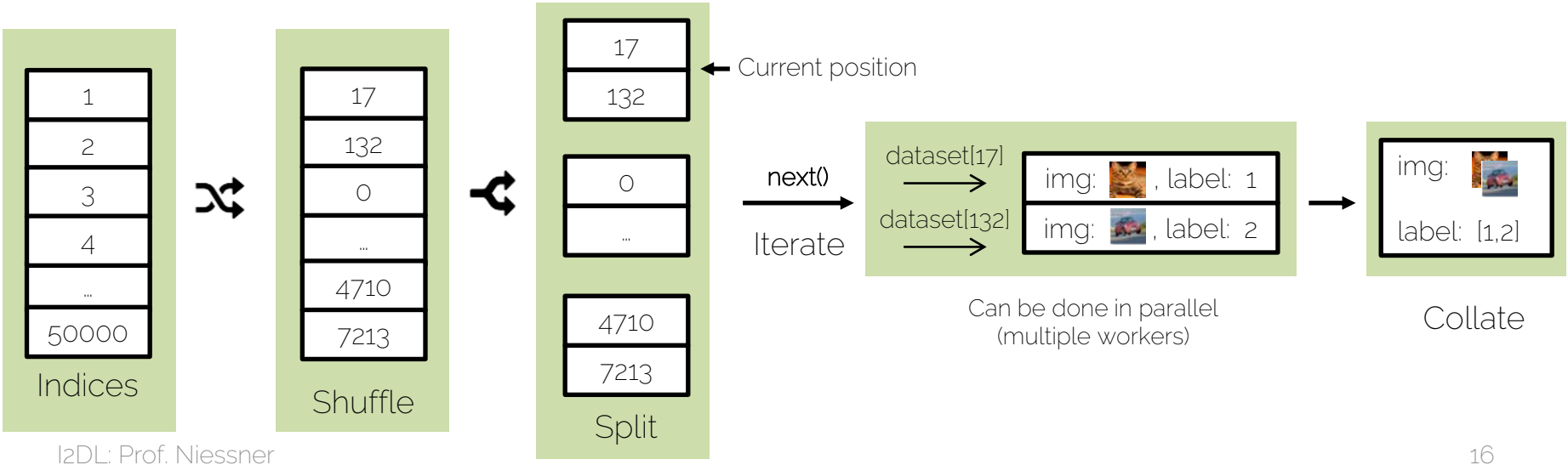
Exercise 3: Dataloader

- Defines how to load the dataset for model training
  - E.g., number of images per batch, number of workers
- Shuffles the dataset
- Splits the dataset into small subsets: (mini) batches



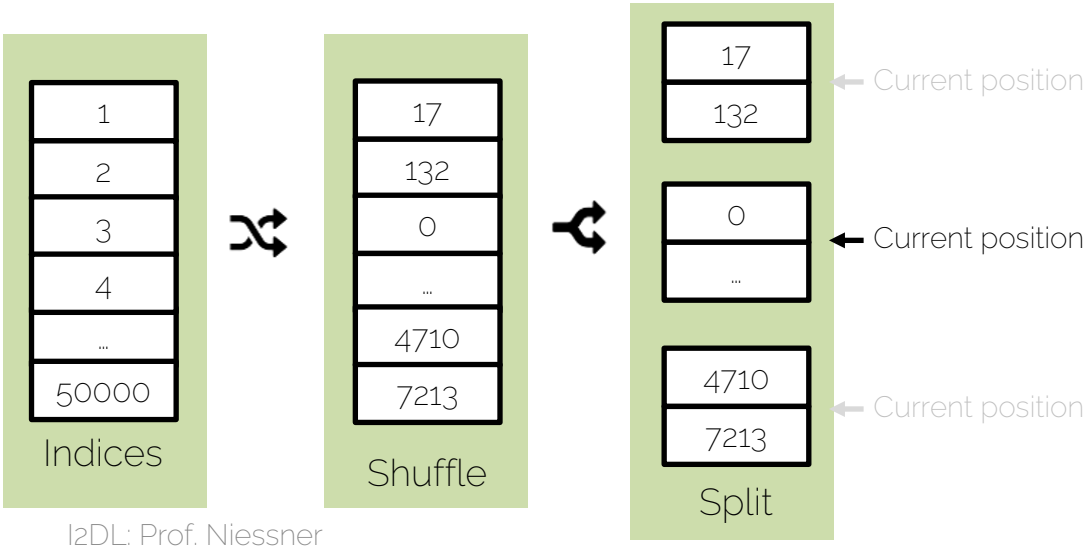
Exercise 3: Dataloader – Iterator & Batching

- Dataloader is an "iterator", not a list
  - Cannot be directly accessed with an index: ~~*dataloader[0]*~~
  - Instead iterate using "next" to get next element: next(dataloader)
  - __iter__() function uses "yield" instead of "return"
- Returns a (mini-) batch of samples in a batched format



Exercise 3: Dataloader – Iterator & Batching

- Dataloader is an "iterator", not a list
  - Cannot be directly accessed with an index: ~~*dataloader[0]*~~
  - Instead iterate using "next" to get next element: next(dataloader)
  - __iter__() function uses "yield" instead of "return"
- Returns a (mini-) batch of samples in a batched format



Overview Exercise 3

- Two notebooks
  - Dataset: CIFAR10
  - Dataloader

- Submission
  1. Implement solution in both notebooks
  2. Single submission zip is created in Dataloader notebook

See you next week