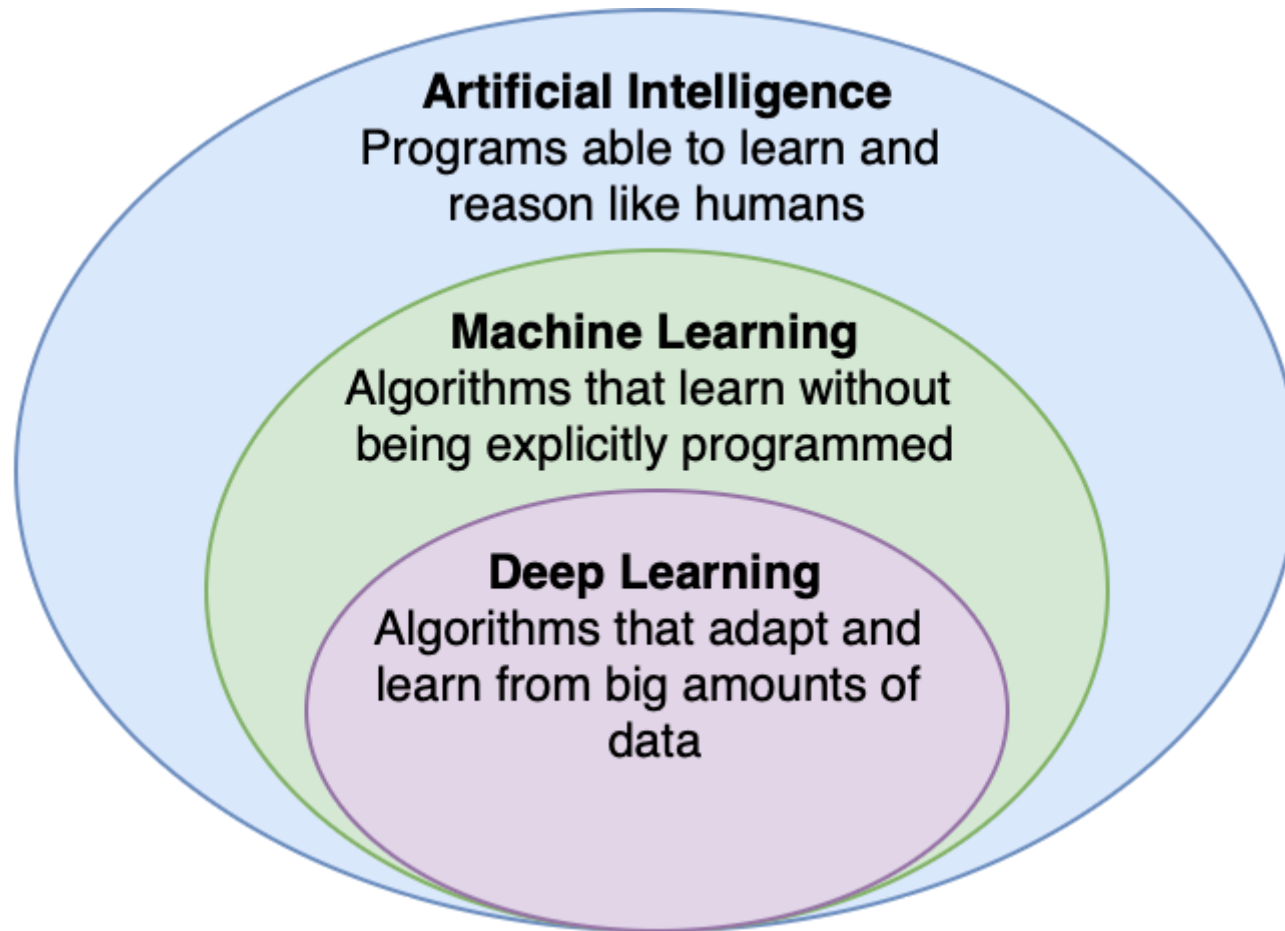


PART 01 - INTRODUCTION TO MACHINE LEARNING AND DEEP LEARNING

Maura Pintor (maura.pintor@unica.it)

AI, ML, AND DL



MACHINE LEARNING

"the use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyse and draw inferences from patterns in data"

Oxford Languages

TYPES OF MACHINE LEARNING

- Supervised learning
- Unsupervised learning
- Reinforcement learning

This course focuses on **supervised learning**, but you can do all three with PyTorch.

SUPERVISED LEARNING

Basic concepts:

- samples
- features
- labels

EXAMPLE: CLASSIFICATION



label: elephant

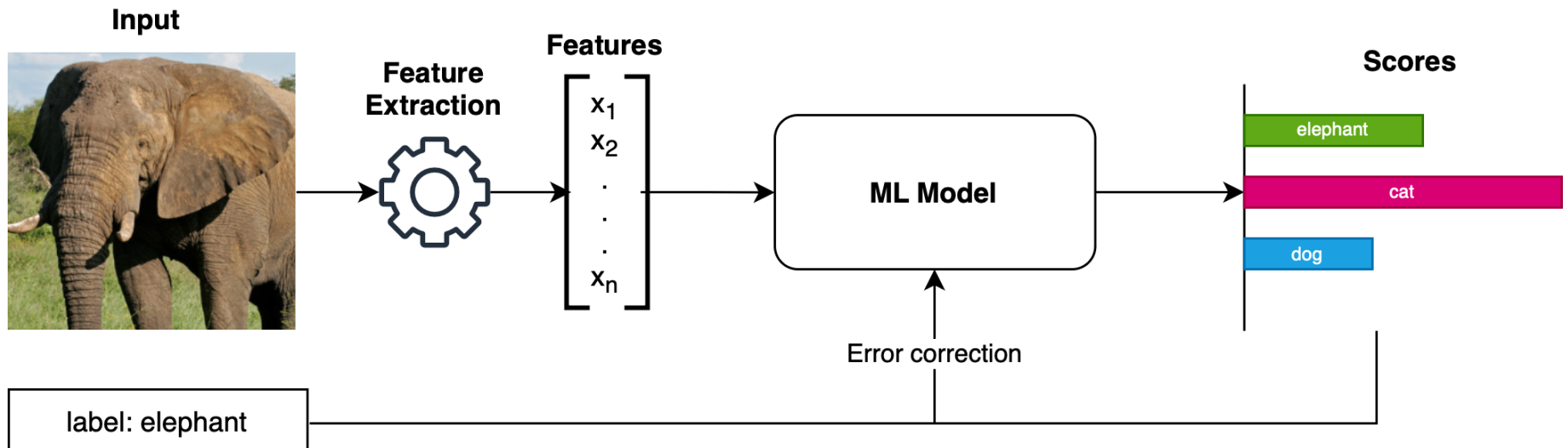
ML Model

elephant

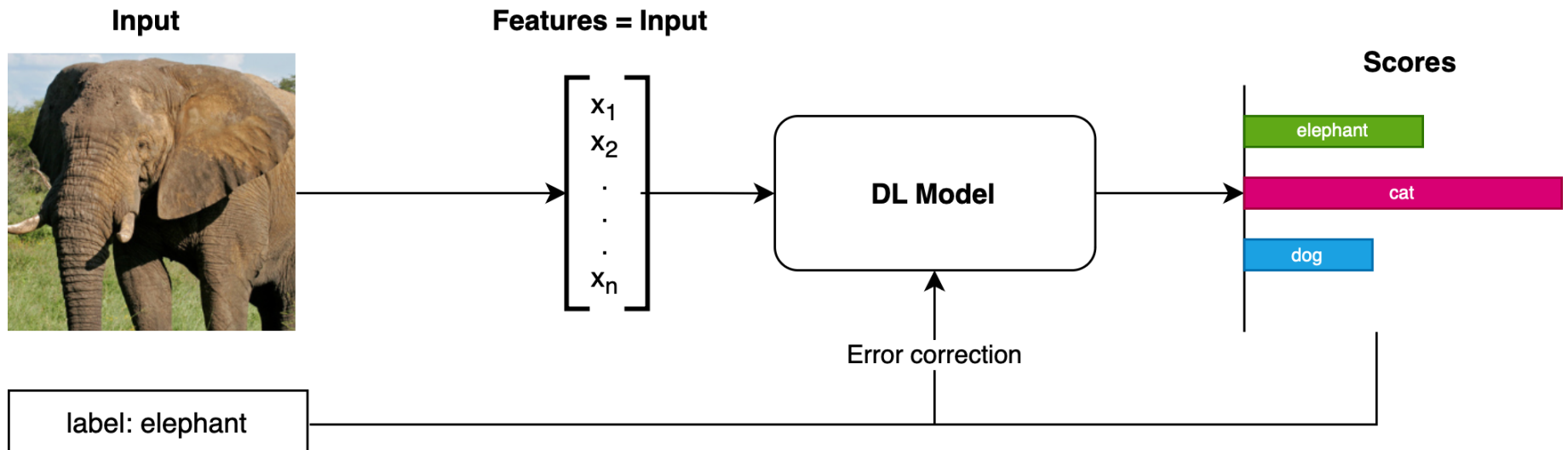
cat

dog

HOW DOES ML LEARN?



HOW DOES DL LEARN?



THE DEEP LEARNING REVOLUTION

- ML model use handcrafted features to characterize the input. ML relied heavily on feature engineering.
- DL model include (and learn directly) feature extraction. DL focuses on finding useful representations automatically.

HOW TO MAKE DL SUCCESSFUL

- We need to get useful data;
- We need to feed those data to a model;
- We need to define the model;
- We need an automated way of *training* the DL model.

TOOLS FOR DEEP LEARNING

You can write deep learning with `scikit-learn`, but it is not going to be efficient.

The reason will be explained very soon.

Although we will use **PyTorch** for the course, the concepts will be presented as foundational deep learning tool and can easily be applied to other libraries (e.g., **Tensorflow**).

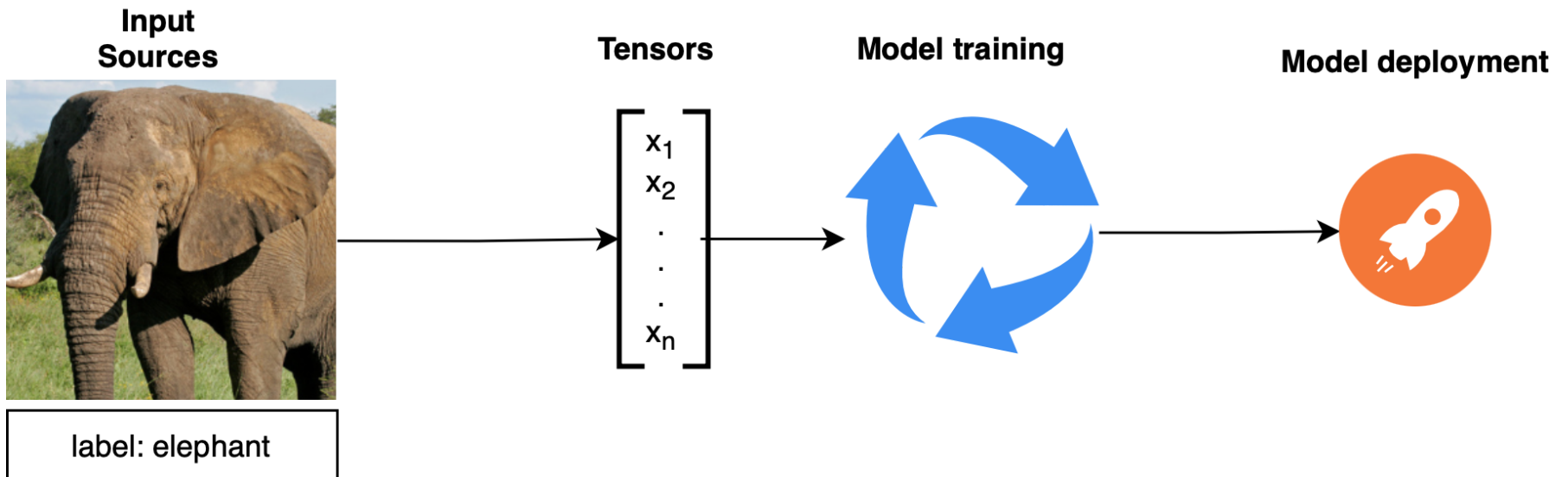
WHY DO WE NEED DL LIBRARIES?

- DL-specific APIs
- support numerical optimization on generic mathematical expressions
- native support for parallel data loaders
- GPU computation ($\sim 50x$ faster than CPU)
- high-performance C++ runtime for deployment

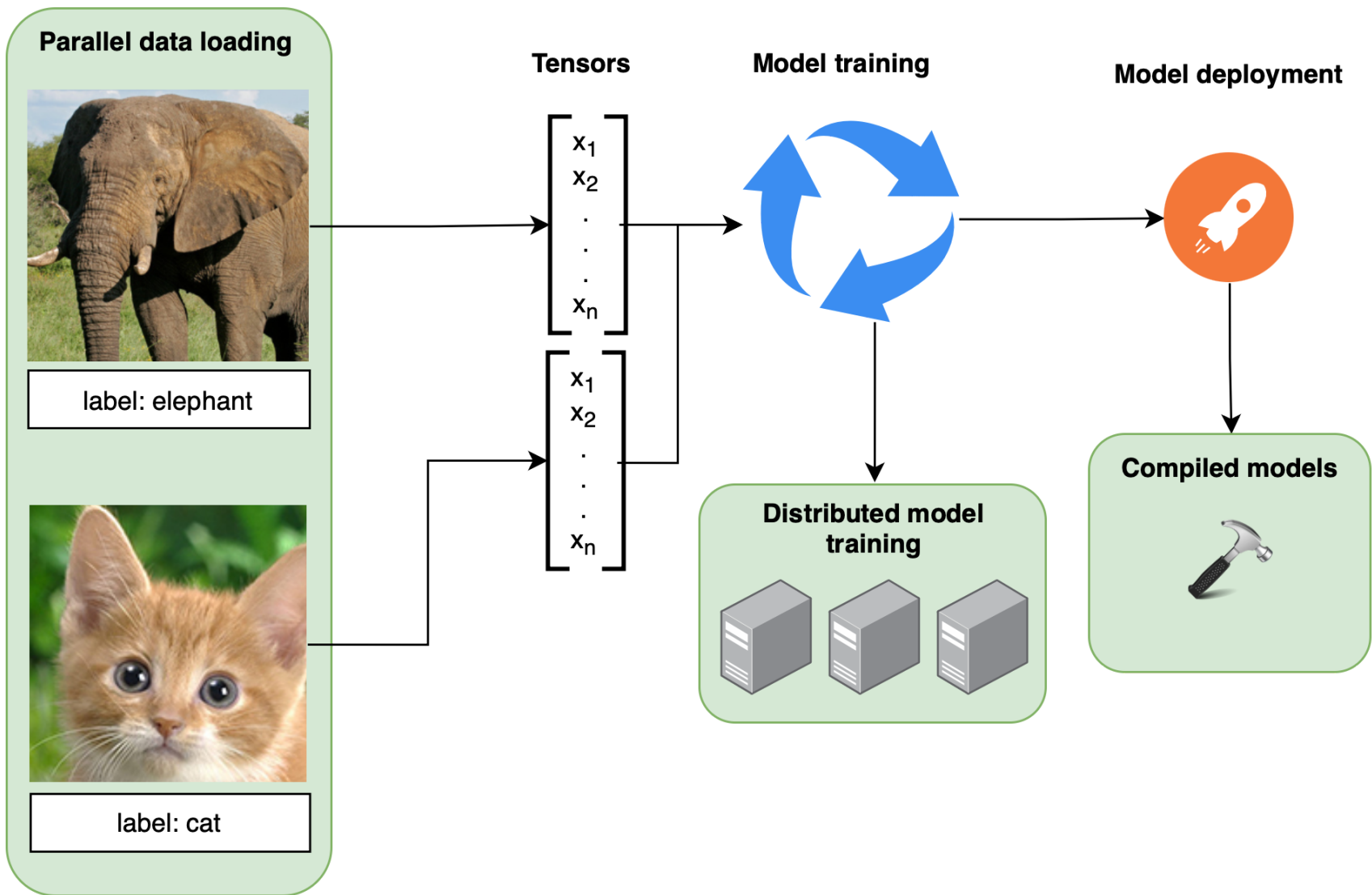
THE PYTORCH LIBRARY

- Most of PyTorch is written in C++ and CUDA
- Python is just used for exposing the APIs
- Tensors keep track of operations and analytically compute derivatives

SIMPLE USAGE OF PYTORCH



REAL ADVANTAGES OF PYTORCH



REAL ADVANTAGES OF PYTORCH

- Parallel data loading: multiple workers can load **chunks** of data into the RAM/GPU
- Distributed model training: multiple machines/GPUs can be seen as a single physical device and work in parallel
- Compiled models: models can be compiled in C++/CUDA operations for runtime

chunks = small and manageable pieces of data (compared, e.g., with 100GB dataset)

HARDWARE AND SOFTWARE REQUIREMENTS FOR THE COURSE

- It is not required to have a GPU
- It is recommended to have unix-based OS (Linux or MacOS), but it is not impossible to follow with a Windows machine
- We won't train large models in our local machines
- Some parts of the course will be demonstrated by using a server

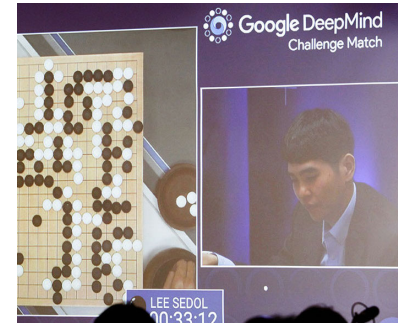
A NOTE ON JUPYTER NOTEBOOKS

- Jupyter Notebooks show a page on the browser, with an interactive Python shell
- They use a kernel, a process that waits for jobs to be run and returns the results, that are displayed on the web interface
- The kernel maintains its state unless it is restarted
 - This means that intermediate results are kept in memory

HOSTED JUPYTER NOTEBOOKS

- Hosted Jupyter notebooks are available at <https://colab.research.google.com>
 - Be aware that they run on a shared runner, hence it is not recommended to launch long-running jobs
- To use the (shared) GPU: `Runtime > Change runtime type > Hardware accelerator > GPU > Save`

EXAMPLES OF DEEP LEARNING TECHNOLOGY



Stable Diffusion:
generate images
by prompt

OpenAI ChatGPT:
multi-purpose
chatbot

AlphaGo: AI able to
play (and win) the
game of Go

... and many others

END OF PART 1

In the next chapter:

- How to represent data for deep learning
- Practical session (prepare your setup)

Maura Pintor (maura.pintor@unica.it)

