COMPUTER SCIENCE

Lecture 1: Introduction

Xin Chen
Assistant Professor
University of New Mexico

What will we learn?

- The basic algorithms in **traditional learning** and their implementations in Python.
- Basic concepts in **supervised learning**, **unsupervised learning** and **reinforcement learning**.
- Applications of traditional learning algorithms.
- The ML libraries *scikit-learn* and *PyTorch*.
- Building and training **artificial neural networks** in PyTorch.
- Applications of artificial neural networks.

How to learn this course?

- **Taking every class.** *Since we have a comprehensive course content, each lecture is important.*
- **Taking notes in every class.** *The slides will not include everything, please take notes of the content that is not contained in the slides and textbook.*
- **Reading the textbook.** *The slides are not a textbook, you should never expect to understand the course content by only reading the slides.*
- **Reading online materials.** *Some online references will be provided in the slides, however, you are encouraged to proactively find online materials using the skills and knowledge learned in this course.*
- **Reviewing after every class.** *Please read the corresponding chapter(s) and the lecture slides after every class.*
- **Arranging regular team discussion.** *Please discuss with your teammates regularly.*

Somethings you need to know

- During a class, **never use laptop(s), cell phone(s) or other things** that distract you from the class.
- This is not a basic programming course, **we do not help with debugging**.
- Please let me know in advance if you will be late (after 10 min) to a class. Please do not be late to a class frequently.
- If you meet any difficulty in course study, please reach out to the instructor or TA **at the first convenient time**.
- **AI tools**, including any LLMs, are **forbidden** in this course.

What is Machine Learning (ML)?

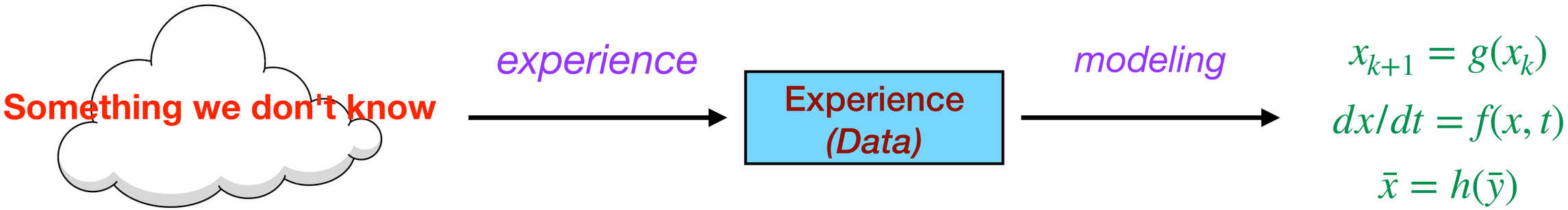
"Learning is any process by which a system improves performance from experience."

- Herbert Simon

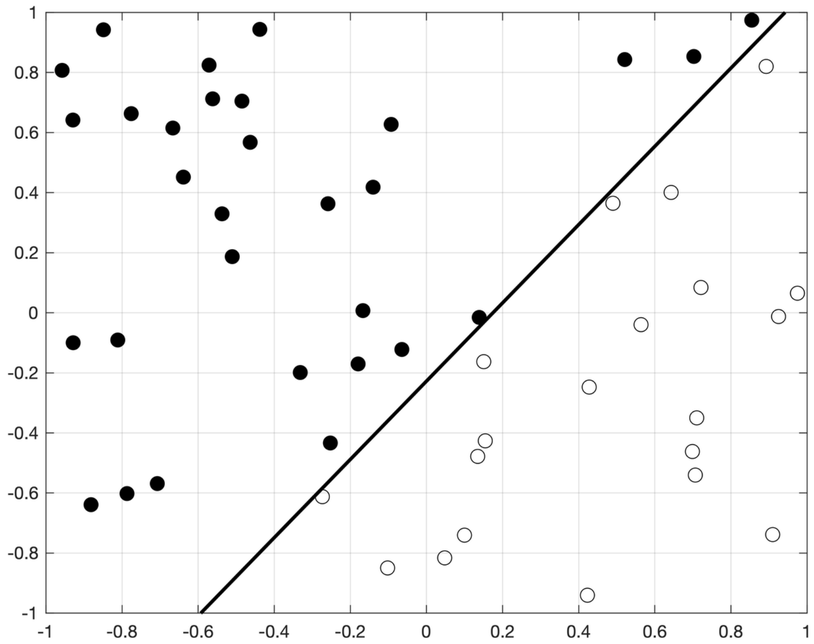
Machine learning is the study of algorithms that

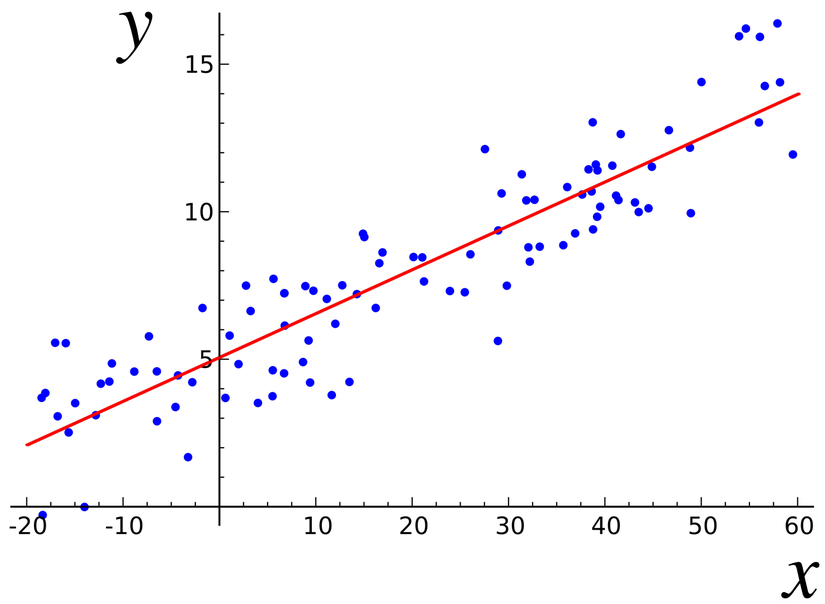
- improve the performance P,
- at some task T,
- and with experience E.

- Tom Mitchell



Example - Tasks of Supervised Learning



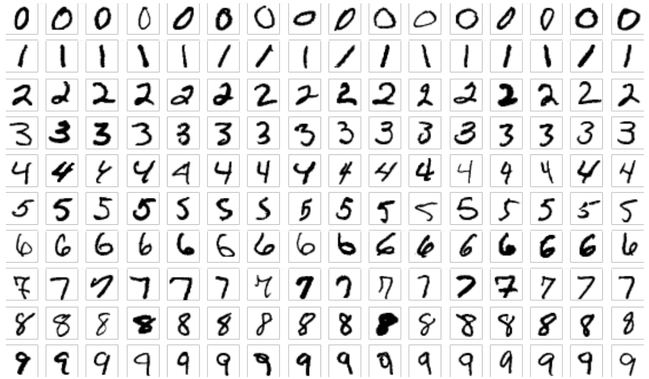


Classification: Finding a *hyperplane* that distinguish the black and white nodes.

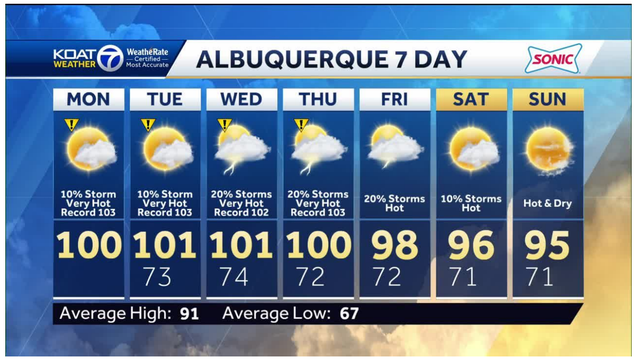
Regression: Finding a *function* that describes the relation between the variables x and y.

Advanced ML Tasks

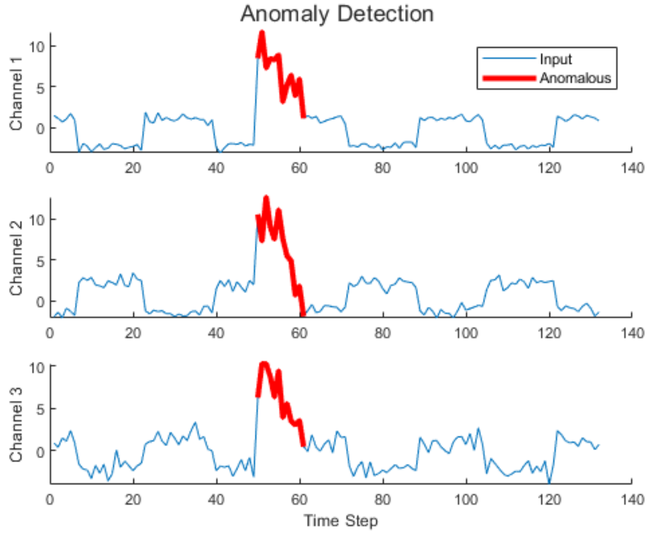
Pattern Recognition



Weather Forecast



Anomaly Detection

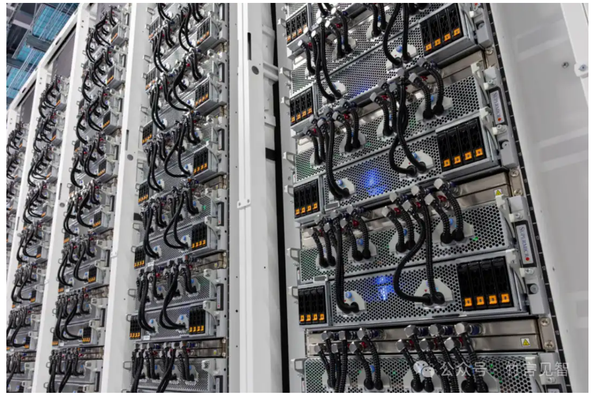


Voice Translation

Motion Planning

Hardware for ML

Training ML Models



Using ML Models

AI/ML in Embedded Systems

Hardware

Raspberry Pi



Nvidia Jetson

D-Robotics RDK

Software

Raspbian

NVIDIA JETSON

debian

ROS
Robot Operating System

ubuntu

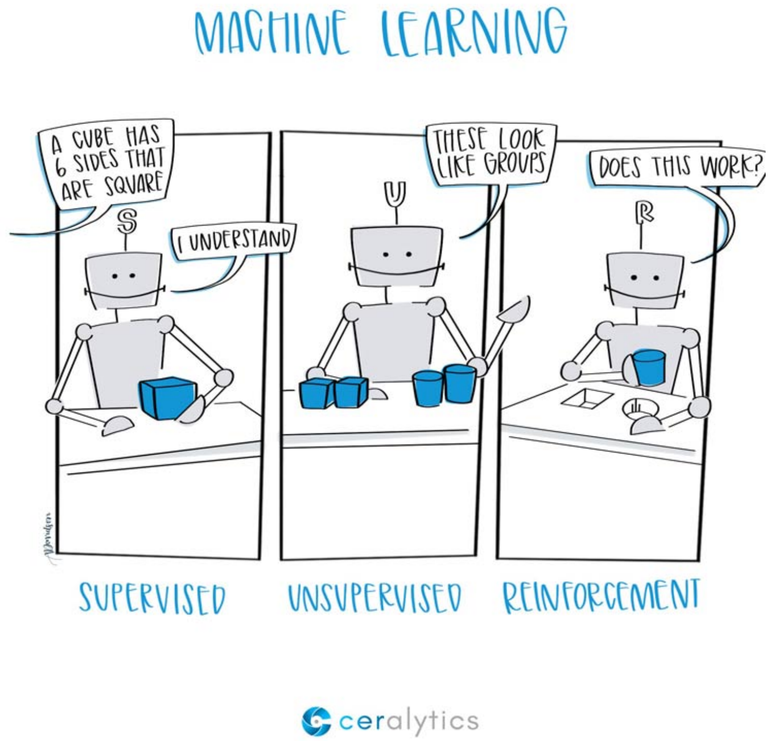
android

Application






Types of Machine Learning



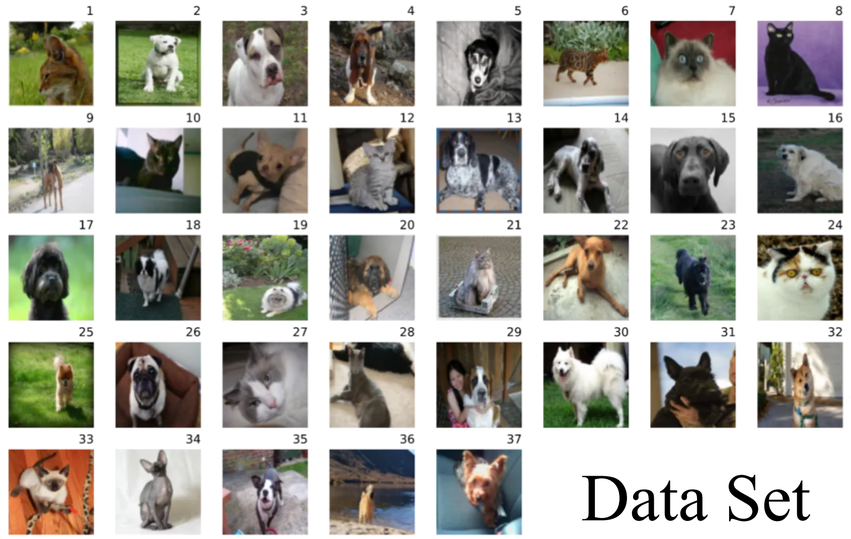
ceralytics

- **Supervised Learning:**
  - *Classification* and *regression*.
- **Unsupervised Learning:**
  - *Clustering*, *dimensionality reduction*, *autoencoders*.
- **Reinforcement Learning:**
  - *Finding a policy based on finite interactions with the environment.*

Supervised Learning

Training Data/Samples

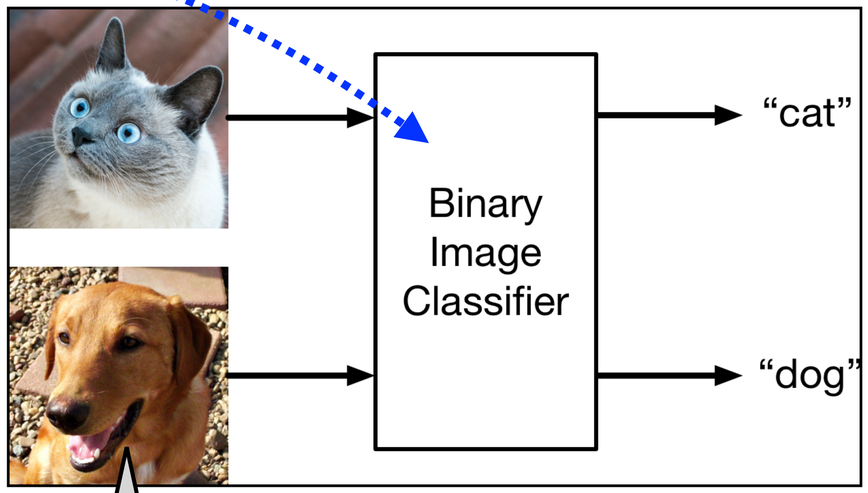
Each sample is correctly labeled. A training data set might not need to be large but is expected to be comprehensive.



Data Set

Training an ML Model

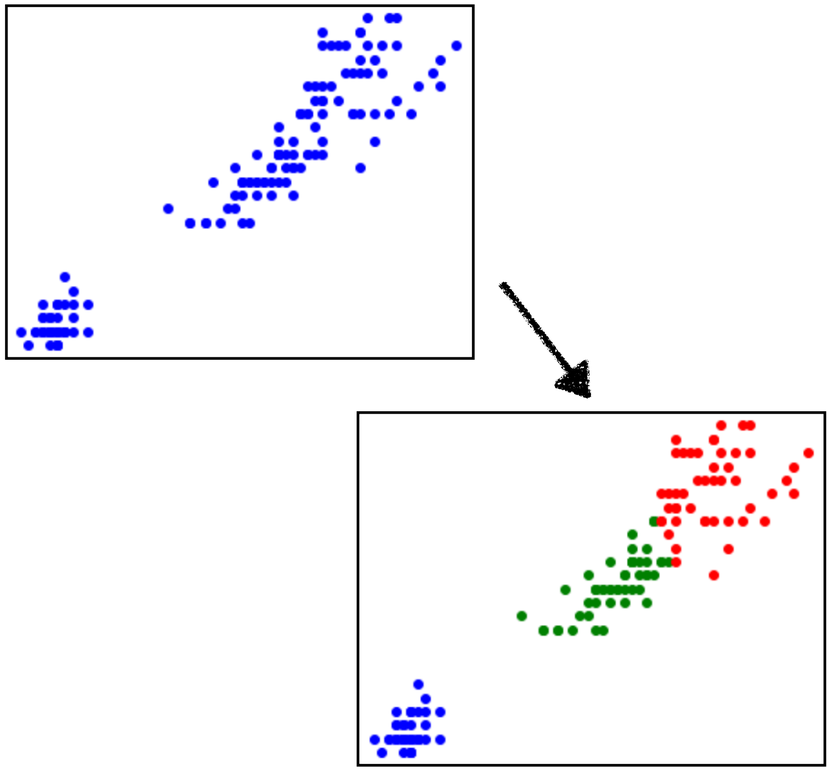
Finding the unknown parameters in a given model to minimize its prediction error on the training samples.



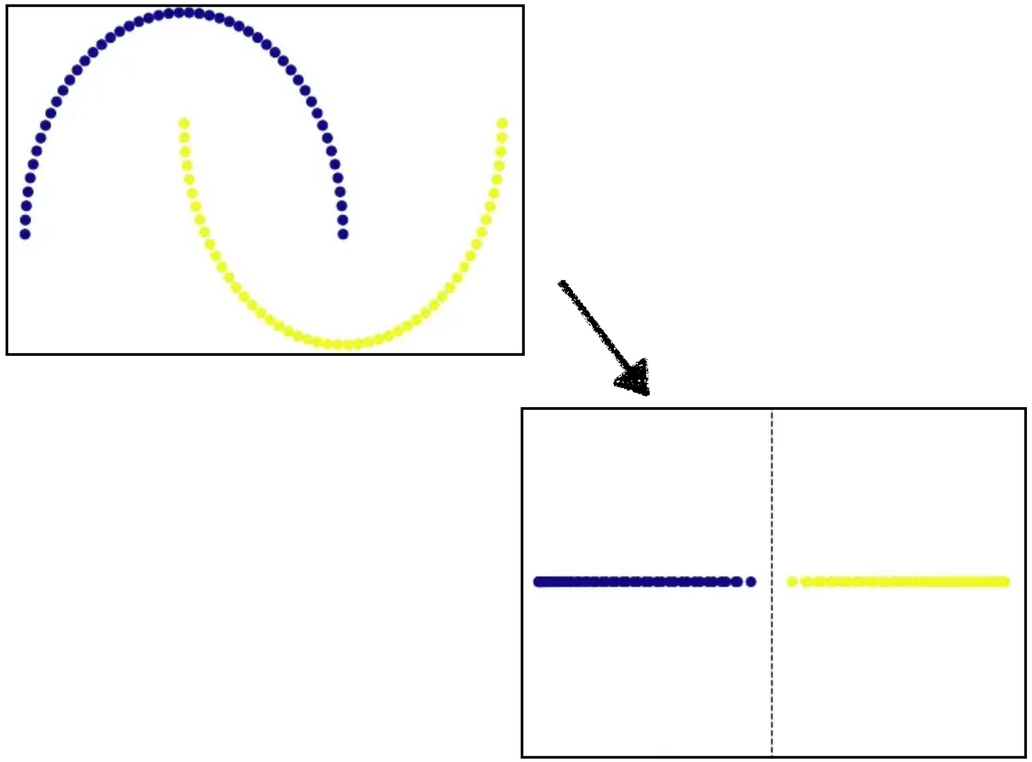
Test Data

Samples used to evaluate the performance of the ML model. These samples should not be in the training data set.

Unsupervised Learning

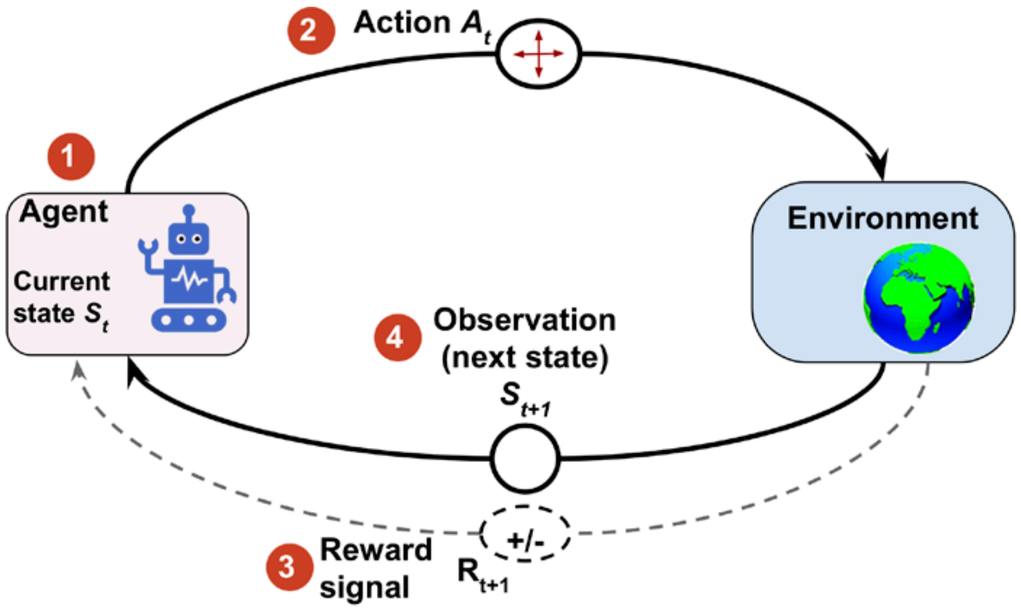


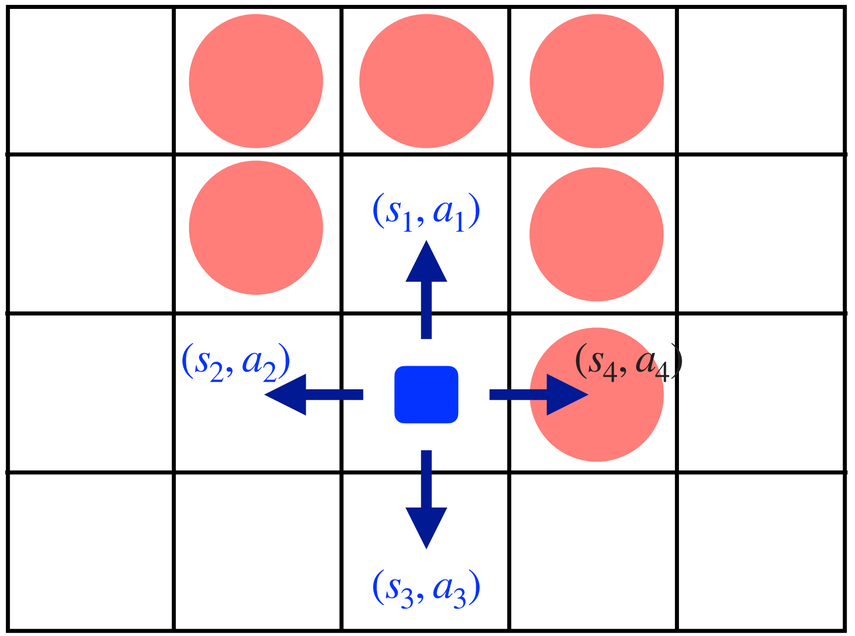
Clustering: Partitioning a set of (unlabeled) samples into homogeneous subsets.



Dimensionality Reduction: Transforming an initial representation of samples into a lower-dimensional representation while preserving some properties of the initial representation.

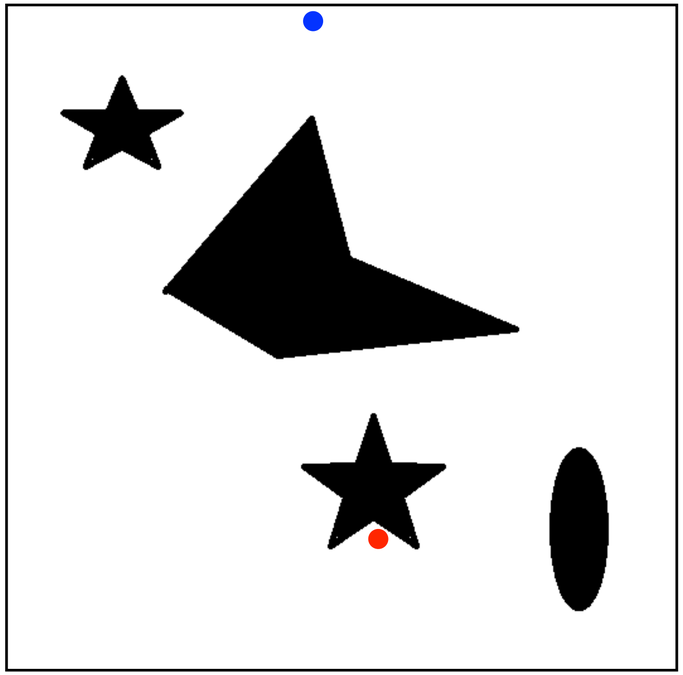
Reinforcement Learning





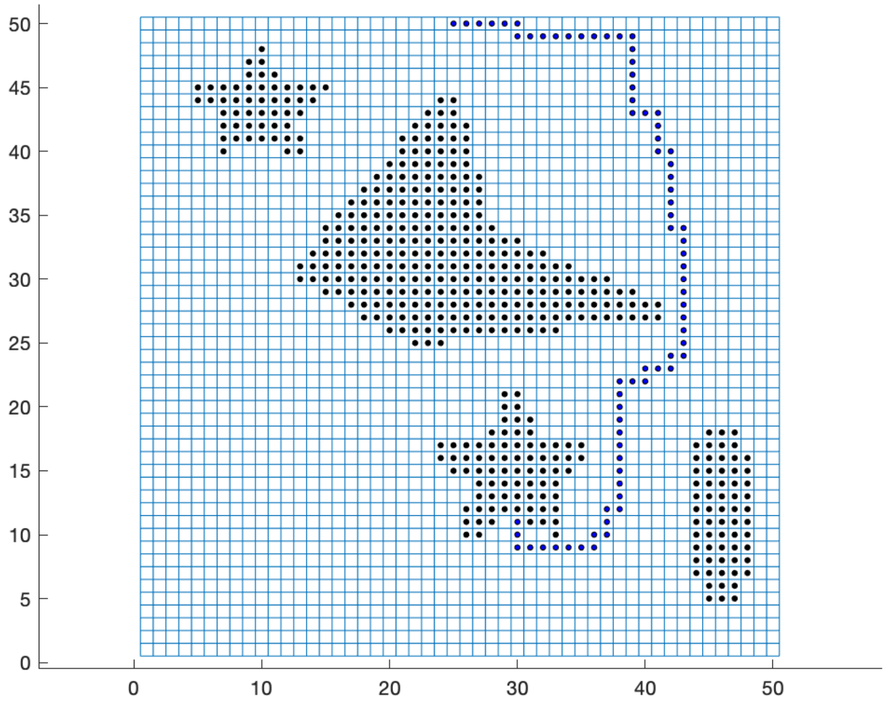
- **State:** The state of the robot: position, moving speed, heading direction, ...
- **Action:** Moving left, right, forward, backward, accelerating, turning, ...
- **Reward:** a positive value for reaching the goal, a negative value for reaching a forbidden position, 0 value for others.

Example - Motion Planning



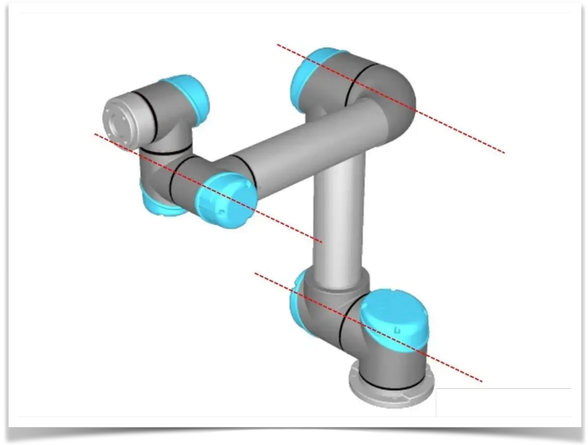
finite abstraction

(gridding the continuous state space)

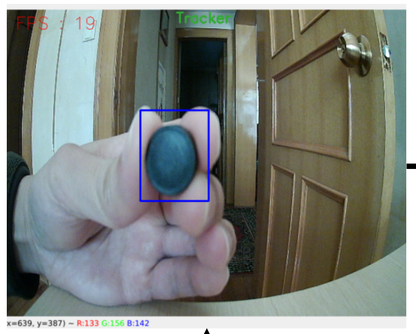


Walking from the start position (red) to the target position (blue) avoid reaching any obstacles. Four actions: **left**, **right**, **up** and **down**. After a reinforcement learning, the agent is going to tell the **best action** to choose in each position (grid).

Motion Planning for Robot Arms



Object Tracking



fetch the image at the current time

1. Detect the Object

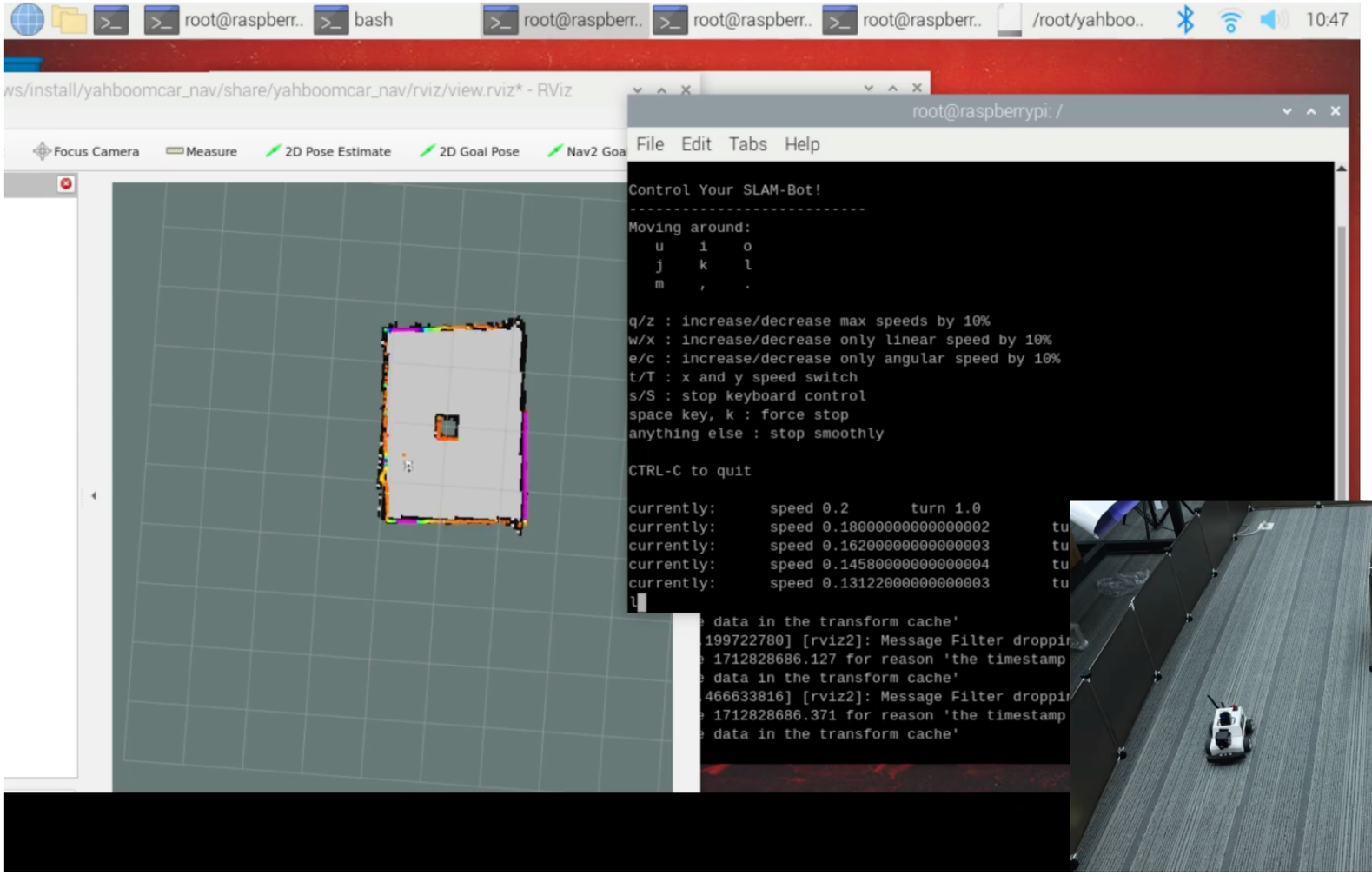
compute the offsets of the blue box in both dimensions

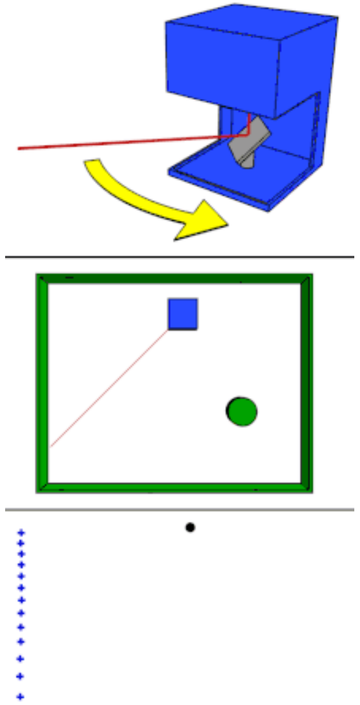
2. Center the Object

Define the move of the camera using a PID controller

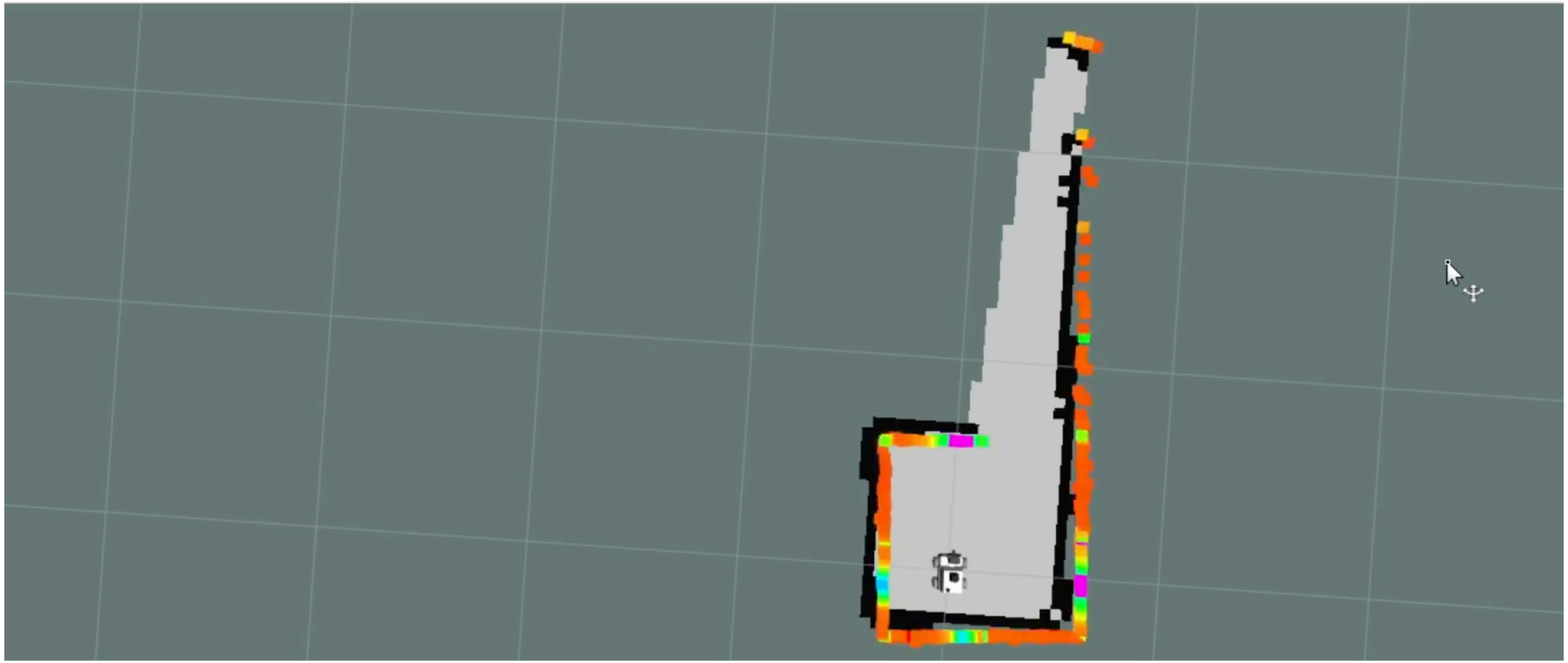
3. Move the Camera

Building a Map from LiDAR Data

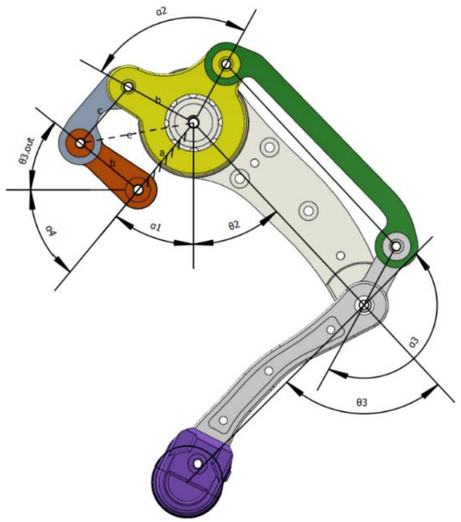


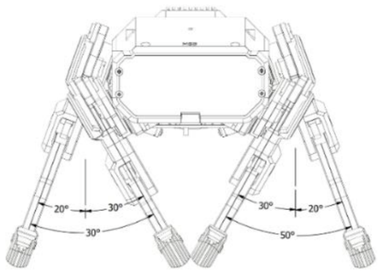


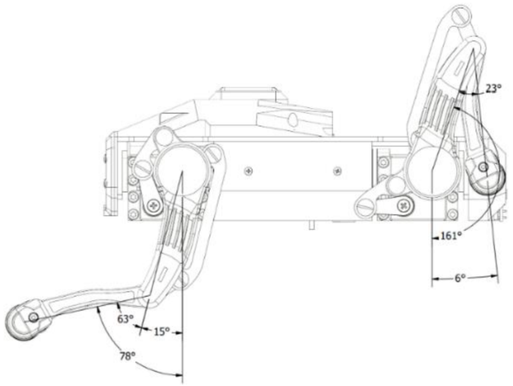
Navigation

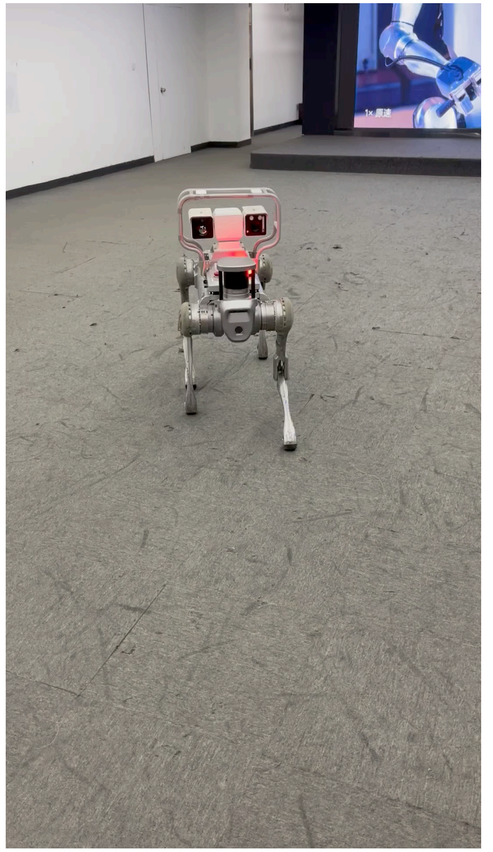


Control of Legged Robots









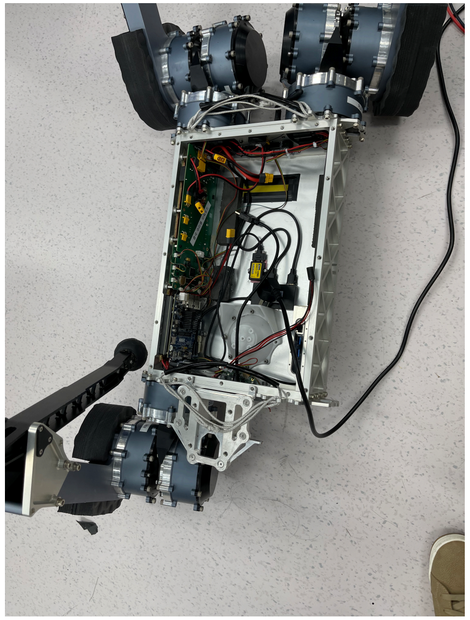
$$e^2 = a^2 + b^2 - 2ab\cos(\pi - \alpha_4 - \theta_{3.out})$$

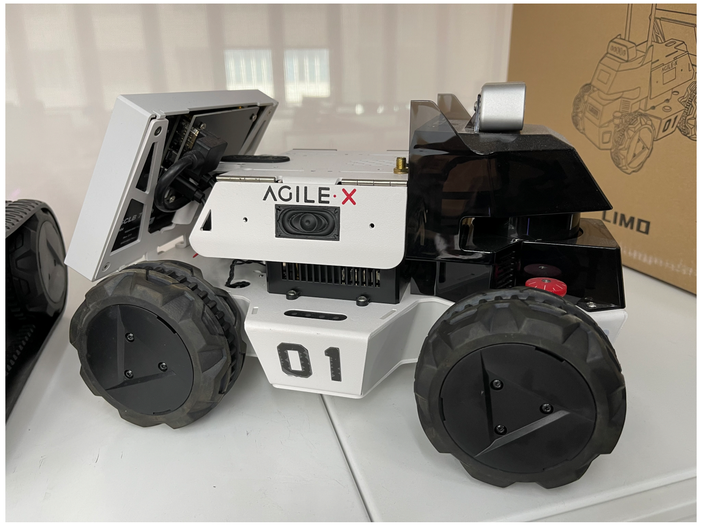
$$\theta_3 = \theta_2 + \cos^{-1}\left(\frac{a^2 + e^2 - b^2}{2ae}\right) + \cos^{-1}\left(\frac{b^2 + e^2 - c^2}{2be}\right) + \alpha_1 + \alpha_2 - \alpha_3$$

https://www.unitree.com/b2


Non-contact Continuous Nocturnal Blood Pressure Monitoring using Fiber Optic Ballistocardiography
BACKGROUND
METHODOLOGY
RESULTS


AGILE·X




AGILE·X
01

[HKUST-GZ]

Most of the ML tasks require a partially defined model which is often called a **parametric model**.

The task of training the model becomes an **optimization problem** which requires to find the unknown parameters that minimize model's prediction error.

We are going to learn how to solve such optimization problems.