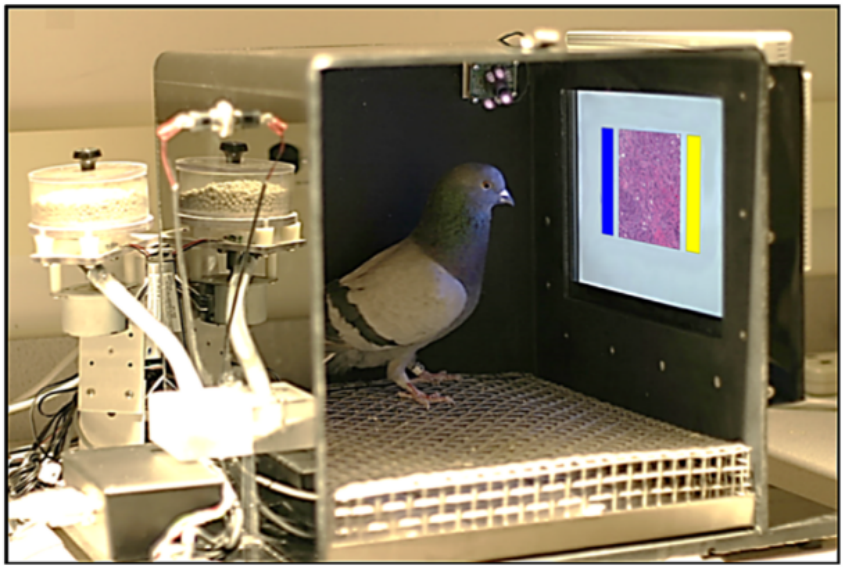
Master 2 Computer Science
RL Course M2 AI

Introduction to Reinforcement Learning

Akka Zemmari

RL: Inspiration



RL: success stories

RL in a nutshell

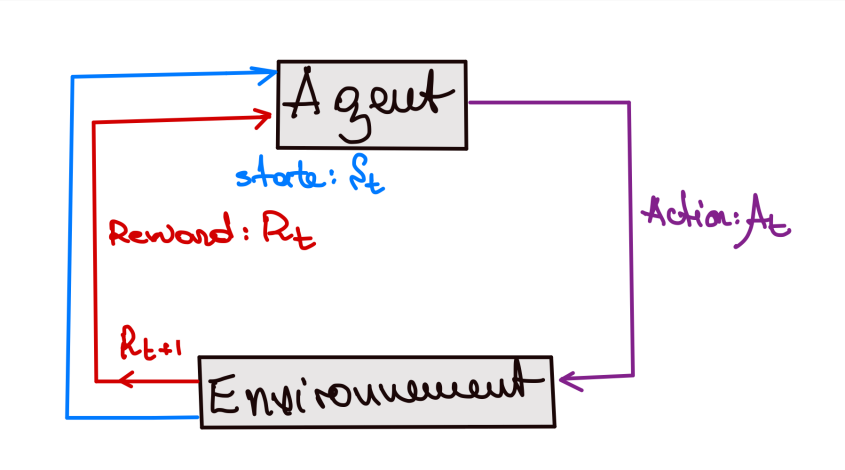


Figure 1: RL: An agent interacting with an environment

Notation: see blackboard ...

RL in a nutshell

- **Reinforcement Learning (RL)**: Focuses on learning how to make sequences of decisions to maximize a reward signal.
- **Agent**: Learns to interact with an environment to achieve a goal.
- **Environment**: Provides feedback to the agent in the form of rewards.
- **Goal**: Learn a policy that maps states to actions to maximize the expected cumulative reward.

RL vs ML

- **Machine Learning (ML)**: Focuses on learning patterns from data to make predictions or decisions.
  - **Supervised Learning**: Learns a mapping from input to output data.

$$f \text{ s.t. } f(x) \approx y$$

  - **Unsupervised Learning**: Learns patterns from input data without explicit output labels.

$$f \text{ s.t. } f(x) \approx x$$

- **Reinforcement Learning (RL)**: Focuses on learning how to make sequences of decisions to maximize a reward signal.

Presentation

Objectives: This course introduces the basic concepts and algorithms for Reinforcement Learning (RL).

Instructors:

- **Guillaume Lagarde** (`guillaume.lagarde@labri.fr`)
- **Akka Zemmari** (`akka.zemmari@labri.fr`)

Course format:

- Lectures: 12 sessions of 2 hours
- Hands-on sessions: 12 sessions of 2 hours
- Project: Implementation of an RL algorithm
- Evaluation: Project report and oral presentation

Content

- Introduction to Reinforcement Learning
- Reminders: Probability Theory, Statistics and Calculus
- Multi-Armed Bandits
- Exploration and Exploitation
- Markov Decision Processes (MDPs)
- Dynamic Programming
- Monte Carlo, Temporal Difference Learning, Sarsa, Q-Learning
- Function Approximation, Gradient Methods, Monte Carlo Tree Search (MCTS)
- Deep Reinforcement Learning (DQN)
- Multi-Agent Reinforcement Learning
- Project

Hands on sessions

- **Python**: Programming language used in this course
- **Jupyter Notebook**: Interactive environment for running Python code
- **OpenAI Gym**: Toolkit for developing and comparing RL algorithms
- **PyTorch**: Deep learning library used in this course
- **RL Baselines3 Zoo**: Collection of pre-trained RL agents