Reinforcement Learning and Beyond:
I. Introduction to Reinforcement Learning

Francisco S. Melo

INESC-ID, Portugal
Let Us Start...

... with a game.
Formula for Success

To succeed in the game you had to figure out:

- The game dynamics (how your actions affect the game)
- The goal of the game

To figure these things out, you had to:

- Try out the different actions (explore)

Once you figure these things out you played “optimally”.

In your case, you used several clues:

- Layout of the maze
- Names of the actions

If you were a machine, you would instead see something like:

```
STATE: 1  SCORE: 0
PLEASE CHOOSE ACTION: 1
STATE: 2  SCORE: 0
PLEASE CHOOSE ACTION:
```
Reinforcement Learning
The Reinforcement Learning Setting

WORLD

Agent

WORLD
State
Reward
AGENT
Action
Reward
State

Elements of Reinforcement Learning

- **States** (describe properties of the world that are relevant to the decision process);

- **Actions** (encapsulate the ability of the agent to interact with the world/change the state of the world);

- **Rewards** (provide the agent with evaluative feedback on its actions).
Elements of Reinforcement Learning (cont.)

An example:
Elements of Reinforcement Learning (cont.)

An example:

State: Position in the maze; \{1, \ldots, 20, I, T, G\}
Elements of Reinforcement Learning (cont.)

An example:

State: Position in the maze
{1, \ldots, 20, I, T, G}

Action: Direction of next movement
{N, S, E, W}

Do not change the state

State: Position in the maze
{1, \ldots, 20, I, T, G}

Action: Direction of next movement
{N, S, E, W}
Elements of Reinforcement Learning (cont.)

An example:

- **State:** Position in the maze
  \{1, \ldots, 20, I, T, G\}

- **Action:** Direction of next movement
  \{N, S, E, W\}

- **Reward:** “Desirability” of each state
An example:

**State:** Position in the maze
{1, . . . , 20, I, T, G}

**Action:** Direction of next movement
{N, S, E, W}

**Reward:** “Desirability” of each state

**Goal:** Compute the best action to take in each state.
Is RL...

- ... psychology?
- ... optimal control?
- ... artificial intelligence?
- ... machine learning?
Psychology

**Reinforcement learning**
(in computer science)

Reinforcement signal

AGENT

New (optimal) behavior

**Reinforcement**
(in psychology)

Reinforcement stimuli

INDIVIDUAL

Reinforced behavior
Psychology

Reinforcement learning
(in computer science)

Reinforcement signal

AGENT

New (opt.) behavior

Reinforced behavior

Reinforcement (in psychology)

Reinforcement stimuli

INDIVIDUAL

Reinforced behavior
Optimal Control

Reinforcement learning

- Reinforcement signal
  - AGENT
  - Optimal decision rule

Optimal control

- “Error” signal
  - PLANT/SYSTEM
  - Optimal control law
Optimal Control

Reinforcement learning

\[ r(X_t, A_t) \]

\[ \sum_t \]

\[ X_{t+1} \sim p(X_t, A_t) \]

\[ \max \]

\[ \{A^*_t\} \]

Optimal control

\[ c(x_t, u_t) \]

\[ \sum_t \]

\[ x_{t+1} = f(x_t, u_t, \eta_t) \]

\[ \min \]

\[ \{u^*_t\} \]
Artificial Intelligence

**Reinforcement learning**

- State/reward

  ![AGENT](image)

  Actions

**Artificial intelligence**

- Perceptions/goals

  ![AGENT](image)

  Actions
Reinforcement learning

Sample transitions

Value-function approximation

ALGORITHM

Machine learning

Sample values

Function approximation

ALGORITHM

...to be discussed later.
Some Examples of RL

Helicopter hovering (Ng et al., 2004):

![Helicopter hovering](image_url)
Some Examples of RL

Helicopter manoeuvering (Ng et al., 2004):
Some Examples of RL

Figure-it-out (Ng et al., 2004):
Some Examples of RL

Robot walking (Kohl and Stone, 2004):

Before learning

After learning
Some Examples of RL

Pole balancing (Atkeson and Schaal, 1997):

![Image of pole balancing](image-url)
Reinforcement Learning

Reinforcement learning addresses optimal stochastic control problems in discrete time by trial-and-error and using minimum prior information on the problem.

In this talk:

- The Model → Markov decision process;
- Optimality → Value functions and policies;
- Solutions with model → Dynamic programming;
- Solution without model → Reinforcement learning.
Markov Decision Processes

A Markov decision process (MDP) is a tuple \((X, A, P, r, \gamma)\) where

- \(X\) is the set of possible states;
- \(A\) is the set of possible actions;
- \(P\) represents the dynamics of the process (probabilities of changing state):
  \[
P(x, a, y) = \mathbb{P} [X_{t+1} = y \mid X_t = x, A_t = a]
\]
- \(r\) is a reward function, assigning a reward \(r(x, a)\) for choosing action \(a\) in state \(x\).
MDPs from the Agent’s Perspective

AGENT

WORLD

“Description”

\[ X_t \]

“Evaluation”

\[ R_t \]

“Description”

\[ X_{t+1} \]

Control

\[ A_t \]
Optimality

- Reward function $r$ “encodes” the task of the agent;
- The agent must choose the actions that maximize the total (discounted) reward
  \[
  \sum_{t=0}^{\infty} \gamma^t r_t
  \]
  where $0 < \gamma < 1$ is a discount factor and $R_t$ is the reward at time $t$;

- A policy $\pi$ is a mapping from states to actions (decision-rule);
- We can associate a value to any policy $\pi$:
  \[
  V^\pi(x) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t R_t \mid X_0 = x \right].
  \]
Our Goal in Intro to RL

Learn how to...

- Compute the best policy $\pi^*$ for a *known* MDP;

- Compute the best policy $\pi^*$ for an *unknown* MDP;
