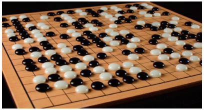
Bits of Reinforcement learning

Philippe Preux
philippe.preux@inria.fr
Scool

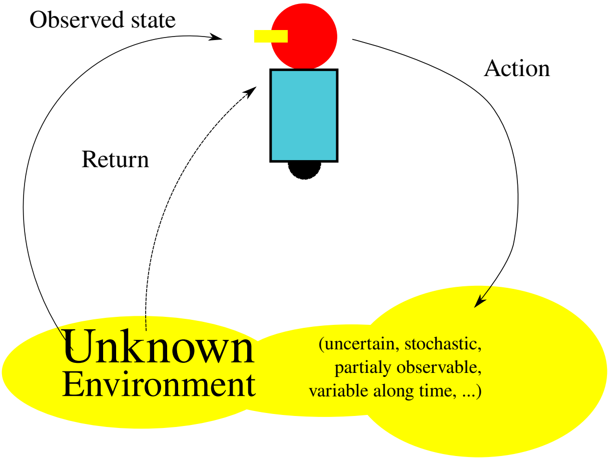
ML in PL, Warsaw

Reinforcement learning

early 1990's, 1 hidden layer made 40 to 80 neurons, self-play, expert level.



The Reinforcement Learning Problem



Learn an optimal behavior.

The Reinforcement Learning Problem

- Markov Decision Problems:

  (states, actions, state dynamics, reward function, objective function).

  known, unknown: the reaction of the environment is unknown.

The Reinforcement Learning Problem

- Markov Decision Problems:

  (states, actions, state dynamics, reward function, objective function).

  known, unknown: the reaction of the environment is unknown.

- Basic principle: to learn, interact with the environment, test actions to observe their consequences (= explore) and use this knowledge (= exploit) increasingly along time.

  Balancing exploration and exploitation is a major key to learn efficiently.

The Reinforcement Learning Problem

- Markov Decision Problems:

  (states, actions, state dynamics, reward function, objective function).

  known, unknown: the reaction of the environment is unknown.

- Basic principle: to learn, interact with the environment, test actions to observe their consequences (= explore) and use this knowledge (= exploit) increasingly along time.

- Basic idea: maintain expectations of the consequences of actions (value). When the observations do not match the expectation, update it: learn when surprised.

The Reinforcement Learning Problem

- Markov Decision Problems:

  (states, actions, state dynamics, reward function, objective function).

  known, unknown: the reaction of the environment is unknown.

- Basic principle: to learn, interact with the environment, test actions to observe their consequences (= explore) and use this knowledge (= exploit) increasingly along time.

- Basic idea: maintain expectations of the consequences of actions (value). When the observations do not match the expectation, update it: learn when surprised.

- 3 main algorithmic branches:

The Reinforcement Learning Problem

- Markov Decision Problems:

  (states, actions, state dynamics, reward function, objective function).

  known, unknown: the reaction of the environment is unknown.

- Basic principle: to learn, interact with the environment, test actions to observe their consequences (= explore) and use this knowledge (= exploit) increasingly along time.

- Basic idea: maintain expectations of the consequences of actions (value). When the observations do not match the expectation, update it: learn when surprised.

- 3 main algorithmic branches:
  - learn the value and deduce a policy from it (Q-learning, DQN, ...).

The Reinforcement Learning Problem

- Markov Decision Problems:

  (states, actions, state dynamics, reward function, objective function).

  known, unknown: the reaction of the environment is unknown.

- Basic principle: to learn, interact with the environment, test actions to observe their consequences (= explore) and use this knowledge (= exploit) increasingly along time.

- Basic idea: maintain expectations of the consequences of actions (value). When the observations do not match the expectation, update it: learn when surprised.

- 3 main algorithmic branches:
  - learn the value and deduce a policy from it (Q-learning, DQN, ...).
  - learn directly a parameterized policy.

The Reinforcement Learning Problem

- Markov Decision Problems:

  (states, actions, state dynamics, reward function, objective function).

  known, unknown: the reaction of the environment is unknown.

- Basic principle: to learn, interact with the environment, test actions to observe their consequences (= explore) and use this knowledge (= exploit) increasingly along time.

- Basic idea: maintain expectations of the consequences of actions (value). When the observations do not match the expectation, update it: learn when surprised.

- 3 main algorithmic branches:
  - learn the value and deduce a policy from it (Q-learning, DQN, ...).
  - learn directly a parameterized policy.
  - combine these two approaches (actor-critic, PPO, SAC, ...).

The Reinforcement Learning Problem

- Markov Decision Problems:

  (states, actions, state dynamics, reward function, objective function).

  known, unknown: the reaction of the environment is unknown.

- Basic principle: to learn, interact with the environment, test actions to observe their consequences (= explore) and use this knowledge (= exploit) increasingly along time.

- Basic idea: maintain expectations of the consequences of actions (value). When the observations do not match the expectation, update it: learn when surprised.

- 3 main algorithmic branches:
  - learn the value and deduce a policy from it (Q-learning, DQN, ...).
  - learn directly a parameterized policy.
  - combine these two approaches (actor-critic, PPO, SAC, ...).

  Each comes with countless variants.

The Reinforcement Learning Problem

RL seems the answer to many problems.

The Reinforcement Learning Problem

RL seems the answer to many problems. But,

- Learning to solve a problem with RL remains very long, even for very moderate tasks.

The Reinforcement Learning Problem

RL seems the answer to many problems. But,

- Learning to solve a problem with RL remains very long, even for very moderate tasks.
- Solving large tasks requires computing infrastructures beyond the reach of most academic labs.

The Reinforcement Learning Problem

RL seems the answer to many problems. But,

- Learning to solve a problem with RL remains very long, even for very moderate tasks.
- Solving large tasks requires computing infrastructures beyond the reach of most academic labs.
- Moreover, the training time is so long that this raises methodological issues: brittleness of experimental results.

The Reinforcement Learning Problem

RL seems the answer to many problems. But,

- Learning to solve a problem with RL remains very long, even for very moderate tasks.
- Solving large tasks requires computing infrastructures beyond the reach of most academic labs.
- Moreover, the training time is so long that this raises methodological issues: brittelness of experimental results.
- Last but far from least, the design of the state space and the objective function are crucial. And the action space (may be) too.

Reconciling RL with more real applications

Reconciling RL with more real applications

The situation:

- need an accurate simulator.

Reconciling RL with more real applications

The situation:

- need an accurate simulator.
- Many simulators out there in many scientific domains.

Reconciling RL with more real applications

The situation:

- need an accurate simulator.
- Many simulators out there in many scientific domains.

- Not designed to be used in an interaction loop:
  1) the user describes the simulation to be done in a configuration file,
  2) run the simulator on this configuration file, and
  3) the simulator outputs a result file.

Reconciling RL with more real applications

The situation:

- need an accurate simulator.
- Many simulators out there in many scientific domains.
- Not designed to be used in an interaction loop:
  1) the user describes the simulation to be done in a configuration file,
  2) run the simulator on this configuration file, and
  3) the simulator outputs a result file.
- An RL compliant simulator requires sophisticated modifications.
  The simulator is usually a complex piece of software, resulting from years, decades sometimes, of work, usually written in either Fortran, or C, or C++.

RL meets soft robots

RL meets soft robots

- *Soft* robot vs. rigid robot.

RL meets soft robots

- *Soft* robot vs. rigid robot.
- Infinite degrees of freedom.

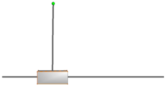
RL meets soft robots

- *Soft* robot vs. rigid robot.
- Infinite degrees of freedom.
- Simulation is much more complex than for rigid robots.

RL meets soft robots

- *Soft* robot vs. rigid robot.
- Infinite degrees of freedom.
- Simulation is much more complex than for rigid robots.
- Example: the famous cartpole turns into a cartStem.

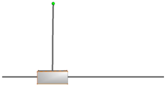
Rewarded when its tip is in the instable equilibrium position



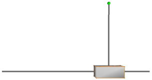
RL meets soft robots

- *Soft* robot vs. rigid robot.
- Infinite degrees of freedom.
- Simulation is much more complex than for rigid robots.
- Example: the famous cartpole turns into a cartStem.

  Rewarded when its tip is in the instable equilibrium position



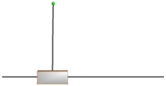
can move to the right:



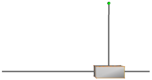
RL meets soft robots

- *Soft* robot vs. rigid robot.
- Infinite degrees of freedom.
- Simulation is much more complex than for rigid robots.
- Example: the famous cartpole turns into a cartStem.

Rewarded when its tip is in the instable equilibrium position



can move to the right:

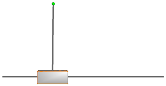


and decelerate:

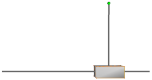
RL meets soft robots

- *Soft* robot vs. rigid robot.
- Infinite degrees of freedom.
- Simulation is much more complex than for rigid robots.
- Example: the famous cartpole turns into a cartStem.

Rewarded when its tip is in the instable equilibrium position



can move to the right:

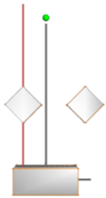


and decelerate:

- An action has a complex non instantaneous outcome,

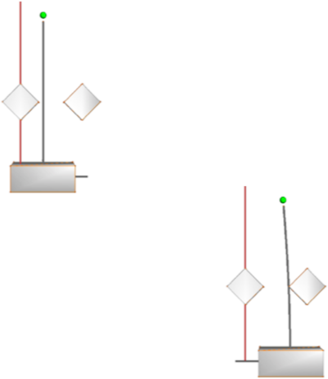
RL meets soft robots

An other example: rewarded when the green tip of the cartStem touches the red line



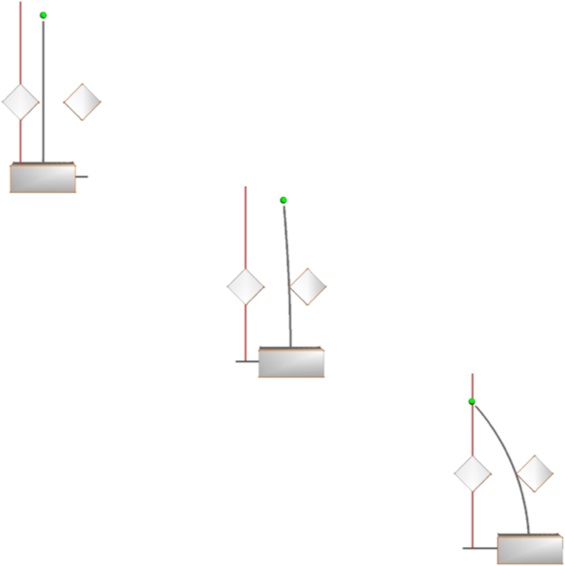
RL meets soft robots

An other example: rewarded when the green tip of the cartStem touches the red line



RL meets soft robots

An other example: rewarded when the green tip of the cartStem touches the red line



RL meets soft robots

Simulating a soft robot

- We use the Simulation Open Framework Architecture (SOFA), a physics-based engine, written in C++.

RL meets soft robots

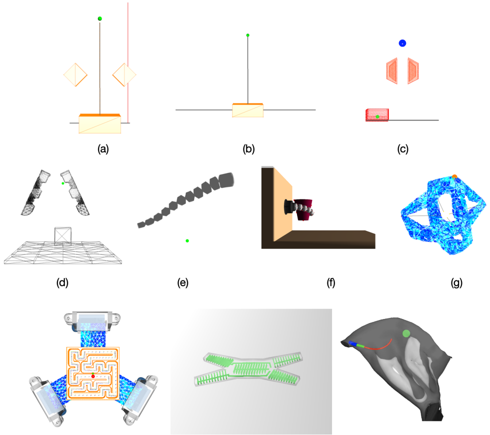
Simulating a soft robot

- We use the Simulation Open Framework Architecture (SOFA), a physics-based engine, written in C++.
- Model the deformation of soft bodies, collisions of soft bodies, ...: complex + computation time raises.

RL meets soft robots

Simulating a soft robot

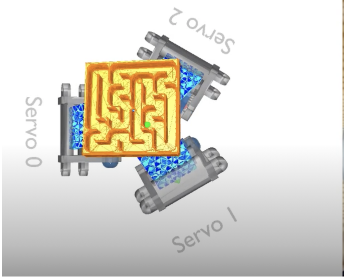
- We use the Simulation Open Framework Architecture (SOFA), a physics-based engine, written in C++.
- Model the deformation of soft bodies, collisions of soft bodies, ...: complex + computation time raises.
- The soft robots we modeled:



RL meets soft robots

Playing with soft robots

- Some are built:



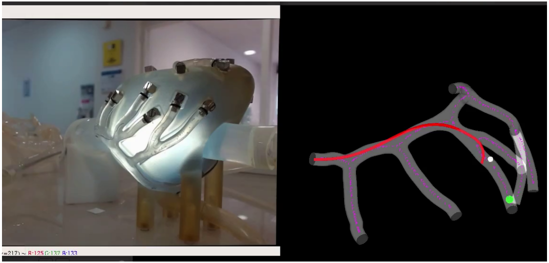
See the tutorial to build your own: Defrost team, Build your own tripod, 2021,

https://handsonsoftrobotics.lille.inria.fr/index.php/tripod

RL meets soft robots

Playing with soft robots

- Some are used for serious matters:



Application to guiding a catheter in coronary arteries.

Controlled by model-based RL.

P. Schegg *et al.*, Automated planning for robotic guidewire navigation in the coronary arteries, *Proc. IEEE 5th International Conference on Soft Robotics* (RoboSoft), April 2022, `https://hal.inria.fr/hal-03778352`.

P. Schegg, *Autonomous Guidewire Navigation for Robotic Percutaneous Coronary Interventions*, PhD dissertation, defended May 2022, Université de Lille, not publicly available.

RL meets soft robots

From SOFA to SofaGym

- Early experiments were made by tinkering SOFA to make it possible to interact with this set of soft robots.

RL meets soft robots

From SOFA to SofaGym

- Early experiments were made by tinkering SOFA to make it possible to interact with this set of soft robots.
- A complete re-design of SOFA core is underway to make it possible to easily interact with any soft robot.

RL meets soft robots

From SOFA to SofaGym

- Early experiments were made by tinkering SOFA to make it possible to interact with this set of soft robots.
- A complete re-design of SOFA core is underway to make it possible to easily interact with any soft robot.
- Gym API. (will move to Gymnasium.)

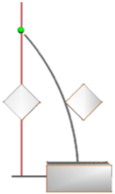
RL meets soft robots

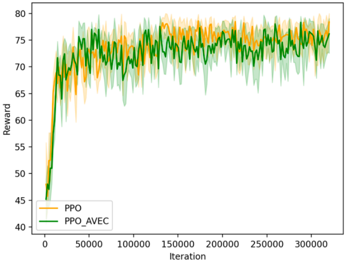
From SOFA to SofaGym

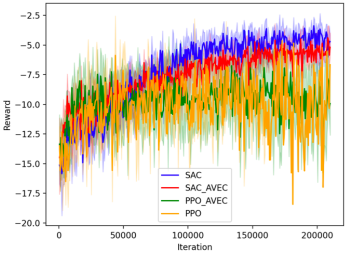
- Early experiments were made by tinkering SOFA to make it possible to interact with this set of soft robots.
- A complete re-design of SOFA core is underway to make it possible to easily interact with any soft robot.
- Gym API. (will move to Gymnasium.)
- SofaGym is developed by soft robots expert research team Defrost at Inria Lille.

RL meets soft robots

Some experimental results







AVEC: replace $\hat{Q}$ by a variance in the AC bias, that is in $\hat{V}(s) - \hat{Q}(s, a)$. Y. Flet-Berliac *et al.*, Learning Value Functions in Deep Policy Gradients using Residual Variance, *ICLR 2021*, arxiv:2010.04440.

RL meets soft robots

A few last words about soft robots

- SofaGym currently provides 11 soft robots.
- This work is still very preliminary.
- Avenues of work to be done: design of the state space, objective function, action space, transfer learning, model-based RL, combination of planning/RL with inverse model, ...

- SofaGym is freely available on `https://github.com/SofaDefrost/SofaGym`.
- P. Schegg *et al.*, SofaGym: An Open Platform for Reinforcement Learning Based on Soft Robot Simulations, *Soft Robotics*, Apr. 2023, pp. 410-430, `http://doi.org/10.1089/soro.2021.0123`.

Learning to manage a crop field

Learning to manage a crop field

Work done in collaboration with Cirad, CGIAR, and BAU (India).

- End goal: be able to recommend what to do next in the field to the farmer.
  In order to fullfill some objective: make money out of his harvest, feed his family, his animals, buy fertilizers, tools, *etc.*,
  while avoiding pollution, soil destruction, *etc.*

Target: small farm holders.

Learning to manage a crop field

Work done in collaboration with Cirad, CGIAR, and BAU (India).

- End goal: be able to recommend what to do next in the field to the farmer.
  In order to fullfill some objective: make money out of his harvest, feed his family, his animals, buy fertilizers, tools, *etc.*,
  while avoiding pollution, soil destruction, *etc.*

  Target: small farm holders.
- Not enough training data available.

Learning to manage a crop field

Work done in collaboration with Cirad, CGIAR, and BAU (India).

- End goal: be able to recommend what to do next in the field to the farmer.
  In order to fullfill some objective: make money out of his harvest, feed his family, his animals, buy fertilizers, tools, *etc.*,
  while avoiding pollution, soil destruction, *etc.*

  Target: small farm holders.
- Not enough training data available.
- Exploration? Very long to make field experiments to collect data.

Learning to manage a crop field

Work done in collaboration with Cirad, CGIAR, and BAU (India).

- End goal: be able to recommend what to do next in the field to the farmer.
  In order to fullfill some objective: make money out of his harvest, feed his family, his animals, buy fertilizers, tools, *etc.*,
  while avoiding pollution, soil destruction, *etc.*

  Target: small farm holders.
- Not enough training data available.
- Exploration? Very long to make field experiments to collect data.
- Simulators exist.

Learning to manage a crop field

- Crop management has been formulated as a Markov Decision Problem since the 1950's.

Learning to manage a crop field

- Crop management has been formulated as a Markov Decision Problem since the 1950's.
- There exists very accurate crop growth simulators.

Learning to manage a crop field

- Crop management has been formulated as a Markov Decision Problem since the 1950's.
- There exists very accurate crop growth simulators.
- The "Decision Support System for Agrotechnology Transfer" (DSSAT):

Learning to manage a crop field

- Crop management has been formulated as a Markov Decision Problem since the 1950's.
- There exists very accurate crop growth simulators.
- The "Decision Support System for Agrotechnology Transfer" (DSSAT):
  - Developed for more than 30 years now, U. Florida, Gainsville.

Learning to manage a crop field

- Crop management has been formulated as a Markov Decision Problem since the 1950's.
- There exists very accurate crop growth simulators.
- The "Decision Support System for Agrotechnology Transfer" (DSSAT):
  - Developed for more than 30 years now, U. Florida, Gainsville.
  - 300 kloc of Fortran.

Learning to manage a crop field

- Crop management has been formulated as a Markov Decision Problem since the 1950's.
- There exists very accurate crop growth simulators.
- The "Decision Support System for Agrotechnology Transfer" (DSSAT):
  - Developed for more than 30 years now, U. Florida, Gainsville.
  - 300 kloc of Fortran.
  - Mechanistic crop model.

Learning to manage a crop field

- Crop management has been formulated as a Markov Decision Problem since the 1950's.
- There exists very accurate crop growth simulators.
- The "Decision Support System for Agrotechnology Transfer" (DSSAT):
  - Developed for more than 30 years now, U. Florida, Gainsville.
  - 300 kloc of Fortran.
  - Mechanistic crop model.
  - Simulates very accurately the growth of a plant based on the properties of the soil, the cultivar, the weather conditions, initial soil conditions (residue from previous year), ...
    interactions between the soil properties with roots then growth of the plant (PDE integration over time).
    + the actions made in the field: irrigation, fertilization, tillage, ... on a daily basis.

Learning to manage a crop field

From DSSAT to gym-DSSAT

- Crop management has been formulated as a Markov Decision Problem since the 1950's.
- There exists very accurate crop growth simulators.
- The "Decision Support System for Agrotechnology Transfer" (DSSAT).

Learning to manage a crop field

From DSSAT to gym-DSSAT

- Crop management has been formulated as a Markov Decision Problem since the 1950's.
- There exists very accurate crop growth simulators.
- The "Decision Support System for Agrotechnology Transfer" (DSSAT).

- gym interface to DSSAT.

Learning to manage a crop field

From DSSAT to gym-DSSAT

- Crop management has been formulated as a Markov Decision Problem since the 1950's.
- There exists very accurate crop growth simulators.
- The "Decision Support System for Agrotechnology Transfer" (DSSAT).

- gym interface to DSSAT.
- Freely available on `https://gitlab.inria.fr/rgautron/gym_dssat_pdi`.

Learning to manage a crop field

Experiments

- Problem: based on a maize field experiment [Morris *et al.*, 1982]

  How to manage irrigation or fertilization to maximize the yield of a certain cultivar of maize in a certain soil in certain weather conditions?

Learning to manage a crop field

Experiments

- Problem: based on a maize field experiment [Morris *et al.*, 1982]

  How to manage irrigation or fertilization to maximize the yield of a certain cultivar of maize in a certain soil in certain weather conditions?

- Action set: irrigate (volume), fertilize (amount), do-nothing

Learning to manage a crop field

Experiments

- Problem: based on a maize field experiment [Morris *et al.*, 1982]

  How to manage irrigation or fertilization to maximize the yield of a certain cultivar of maize in a certain soil in certain weather conditions?
- Action set: irrigate (volume), fertilize (amount), do-nothing
- Crop growth depends on action, soil nature, weather condition (stochastic), *etc.*

Learning to manage a crop field
Experiments

- Problem: based on a maize field experiment [Morris *et al.*, 1982]

  How to manage irrigation or fertilization to maximize the yield of a certain cultivar of maize in a certain soil in certain weather conditions?
- Action set: irrigate (volume), fertilize (amount), do-nothing
- Crop growth depends on action, soil nature, weather condition (stochastic), *etc.*
- Observation = Collection of measurements amenable to a real farmer $\rightsquigarrow$ partial observability.

Learning to manage a crop field

- Task: How to manage irrigation or fertilization to maximize the yield of a certain cultivar of maize in a certain soil in certain weather conditions?
  Let's focus on the fertilization problem.

Learning to manage a crop field

- Task: How to manage irrigation or fertilization to maximize the yield of a certain cultivar of maize in a certain soil in certain weather conditions?
  Let's focus on the fertilization problem.
- We look for a policy: $\forall$ day: (day, amount of fertilizer)
  which is efficient and effective:
  trades-off yield vs. pollution and cost.

Learning to manage a crop field

- Task: How to manage irrigation or fertilization to maximize the yield of a certain cultivar of maize in a certain soil in certain weather conditions?
  Let's focus on the fertilization problem.
- We look for a policy: $\forall$ day: (day, amount of fertilizer)
  which is efficient and effective:
  trades-off yield vs. pollution and cost.
- The daily return is defined by:
  $r(day) = \text{Nitrogen uptake}(day, day+1) - 0.5 \times \text{fertilizer quantity}(day)$

Learning to manage a crop field

- Task: How to manage irrigation or fertilization to maximize the yield of a certain cultivar of maize in a certain soil in certain weather conditions?
  Let's focus on the fertilization problem.
- We look for a policy: $\forall$ day: (day, amount of fertilizer)
  which is efficient and effective:
  trades-off yield vs. pollution and cost.
- The daily return is defined by:
  $r(day) = \text{Nitrogen uptake}(day, day+1) - 0.5 \times \text{fertilizer quantity}(day)$
- The goal is to maximize $\sum_{day=0}^{day=\text{harvest}} r(day)$.

Learning to manage a crop field

Some results (1/3)

We compare:

1. A null policy which does not fertilize,
2. An expert policy used in the original 1982 field experiment,
3. A policy learned by RL (basic untuned PPO).

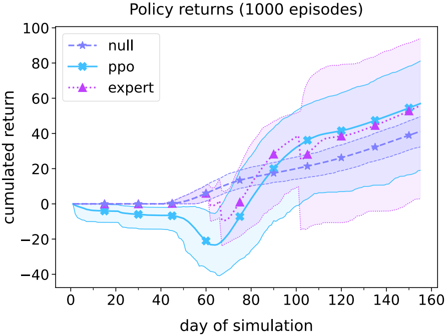
Policies 1. and 2. are fixed and deterministic.
Only the weather is stochastic.

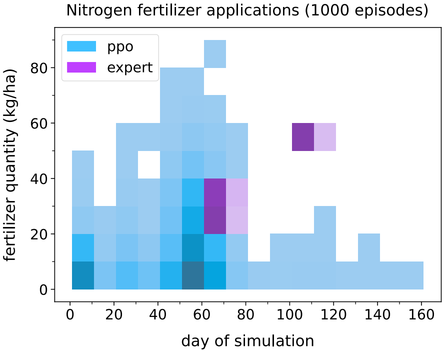
Protocol:

- Null and expert policies are evaluated on 10^3 seasons.
- RL: Trained on 10^6 simulated seasons, then evaluated on 10^3 other seasons.

Learning to manage a crop field

Some results (2/3)





Learning to manage a crop field

Some results (3/3)

	null	expert	PPO
grain yield (kg/ha)	1141.1 (344.0)	**3686.5** (1841.0)	3463.1 (1628.4)
massic nitrogen in grains (%)	1.1 (0.1)	**1.7** (0.2)	1.5 (0.3)
total fertilization (kg/ha)	**0** (0)	115.8 (5.2)	82.8 (15.2)
application number	**0** (0)	3.0 (0.1)	5.7 (1.6)
nitrogen use efficiency (kg/kg)	n.a.	22.0 (14.1)	**28.3** (16.7)
nitrate leaching (kg/ha)	**15.9** (7.7)	18.0 (12.0)	18.3 (11.6)

This table contains the mean (std dev) measured on 10^3 evaluation seasons.

Learning to manage a crop field

Some results (3/3)

	null	expert	PPO
grain yield (kg/ha)	1141.1 (344.0)	**3686.5** (1841.0)	3463.1 (1628.4)
massic nitrogen in grains (%)	1.1 (0.1)	**1.7** (0.2)	1.5 (0.3)
total fertilization (kg/ha)	**0** (0)	115.8 (5.2)	82.8 (15.2)
application number	**0** (0)	3.0 (0.1)	5.7 (1.6)
nitrogen use efficiency (kg/kg)	n.a.	22.0 (14.1)	**28.3** (16.7)
nitrate leaching (kg/ha)	**15.9** (7.7)	18.0 (12.0)	18.3 (11.6)

This table contains the mean (std dev) measured on 10^3 evaluation seasons.

In short: an untuned PPO learns a very good policy that balances the different criteria.

Learning to manage a crop field

Some results (3/3)

	null	expert	PPO
grain yield (kg/ha)	1141.1 (344.0)	**3686.5** (1841.0)	3463.1 (1628.4)
massic nitrogen in grains (%)	1.1 (0.1)	**1.7** (0.2)	1.5 (0.3)
total fertilization (kg/ha)	**0** (0)	115.8 (5.2)	82.8 (15.2)
application number	**0** (0)	3.0 (0.1)	5.7 (1.6)
nitrogen use efficiency (kg/kg)	n.a.	22.0 (14.1)	**28.3** (16.7)
nitrate leaching (kg/ha)	**15.9** (7.7)	18.0 (12.0)	18.3 (11.6)

This table contains the mean (std dev) measured on 10^3 evaluation seasons.

In short: an untuned PPO learns a very good policy that balances the different criteria.

We obtain the same sort of results on the irrigation task.

R. Gautron *et al.*, *gym-DSSAT: a crop model turned into a Reinforcement Learning environment*, Inria Research Report 9460, June 2022, `https://arxiv.org/abs/2207.03270`.

R. Gautron, *Reinforcement learning for rop management support to smallholder farmers in countries of the South: towards risk management*, PhD dissertation, defended Dec. 2022, Université de Montpellier.

Learning to manage a crop field

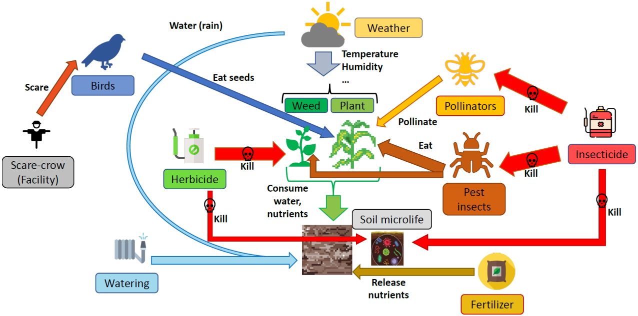
- Many topics remain to be studied.

Learning to manage a crop field

- Many topics remain to be studied.
- gym-DSSAT is great because it is very accurate, but it is hard to manage as a piece of software and limited to DSSAT features.

Learning to manage a crop field

- Many topics remain to be studied.
- gym-DSSAT is great because it is very accurate, but it is hard to manage as a piece of software and limited to DSSAT features.
- $\rightsquigarrow$ Farm-gym: toy farm management environment for RL:



Less accurate but much richer than gym-DSSAT.
Meant to investigate new problem features.

https://github.com/farm-gym/farm-gym

Experimental methodology

Experimental methodology

The big RL experimental failure

- Very poor experimental results.

Experimental methodology

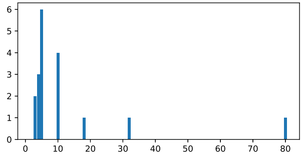
The big RL experimental failure

- Very poor experimental results.
- Most of papers provide conclusions that are not grounded on reliable/significant experimental evidences.

Experimental methodology

The big RL experimental failure

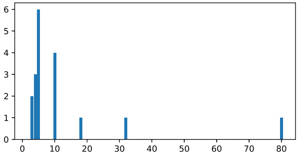
- Very poor experimental results.
- Most of papers provide conclusions that are not grounded on reliable/significant experimental evidences.



Number of runs made in ICML 2022 RL papers.

Experimental methodology
The big RL experimental failure

- Very poor experimental results.
- Most of papers provide conclusions that are not grounded on reliable/significant experimental evidences.

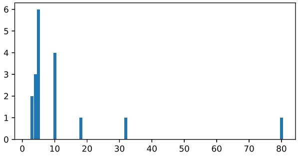


Number of runs made in ICML 2022 RL papers.

- The average of 3 (or 10) runs does not say anything.

Experimental methodology
The big RL experimental failure

- Very poor experimental results.
- Most of papers provide conclusions that are not grounded on reliable/significant experimental evidences.



Number of runs made in ICML 2022 RL papers.

- The average of 3 (or 10) runs does not say anything.
- Computing the standard deviation of 3 (or 10) runs is meaningless.

Experimental methodology

The origins of the big RL experimental failure

- No clear experimental methodology.

Experimental methodology

The origins of the big RL experimental failure

- No clear experimental methodology.
- Vague belief that averaging a set of performances yield a somehow correct way to compare different algorithms.

Experimental methodology

The origins of the big RL experimental failure

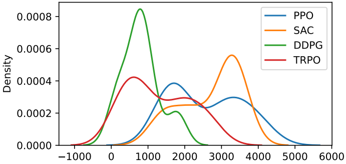
- No clear experimental methodology.
- Vague belief that averaging a set of performances yield a somehow correct way to compare different algorithms.

  This idea is vague, and used to justify the not-too-bad, and the worst.

Experimental methodology

The origins of the big RL experimental failure

- No clear experimental methodology.
- Vague belief that averaging a set of performances yield a somehow correct way to compare different algorithms.
  This idea is vague, and used to justify the not-too-bad, and the worst.
  With 3 runs (or even 10), CLT and other asymptotic results do not apply.



Distribution of performance is not Gaussian at all.

Experimental methodology

The origins of the big RL experimental failure

- No clear experimental methodology.
- Vague belief that averaging a set of performances yield a somehow correct way to compare different algorithms.

  This idea is vague, and used to justify the not-too-bad, and the worst.

  With 3 runs (or even 10), CLT and other asymptotic results do not apply.
- Today, one RL run may take hours.

Experimental methodology

The origins of the big RL experimental failure

- No clear experimental methodology.
- Vague belief that averaging a set of performances yield a somehow correct way to compare different algorithms.

  This idea is vague, and used to justify the not-too-bad, and the worst.

  With 3 runs (or even 10), CLT and other asymptotic results do not apply.
- Today, one RL run may take hours.
- Many RLers are not familiar with basic notions of HPC.

Experimental methodology

The origins of the big RL experimental failure

- No clear experimental methodology.
- Vague belief that averaging a set of performances yield a somehow correct way to compare different algorithms.

  This idea is vague, and used to justify the not-too-bad, and the worst.

  With 3 runs (or even 10), CLT and other asymptotic results do not apply.
- Today, one RL run may take hours.
- Many RLers are not familiar with basic notions of HPC.
- The last day before deadline experiment syndrom.

Experimental methodology

The origins of the big RL experimental failure

- No clear experimental methodology.
- Vague belief that averaging a set of performances yield a somehow correct way to compare different algorithms.

  This idea is vague, and used to justify the not-too-bad, and the worst.

  With 3 runs (or even 10), CLT and other asymptotic results do not apply.
- Today, one RL run may take hours.
- Many RLers are not familiar with basic notions of HPC.
- The last day before deadline experiment syndrom.

We should do much better!

Experimental methodology

What we need

- Goal: compare the performance of a set RL agents.
  An agent is a certain implementation of an algorithm, with certain values for its parameters and hyper-parameters.

Experimental methodology

What we need

- Goal: compare the performance of a set RL agents.
- A statistical test to be able to guarantee that agent A is better than agent B which is better than agent C ... at risk α.

Experimental methodology

What we need

- Goal: compare the performance of a set RL agents.
- A statistical test to be able to guarantee that agent A is better than agent B which is better than agent C ... at risk α.
- A test that runs efficiently.

Experimental methodology
What we need

- Goal: compare the performance of a set RL agents.
- A statistical test to be able to guarantee that agent A is better than agent B which is better than agent C ... at risk α.
- A test that runs efficiently.
- A test that everyone can run, even without understanding the maths.

Experimental methodology

What we need

- Goal: compare the performance of a set RL agents.
- A statistical test to be able to guarantee that agent A is better than agent B which is better than agent C ... at risk α.
- A test that runs efficiently.
- A test that everyone can run, even without understanding the maths.
- A test that fits with the RL community practices.

Experimental methodology

What we need

- Goal: compare the performance of a set RL agents.
- A statistical test to be able to guarantee that agent A is better than agent B which is better than agent C ... at risk α.
- A test that runs efficiently.
- A test that everyone can run, even without understanding the maths.
- A test that fits with the RL community practices.
- Cherry on the cake: a test that requires a minimal number of runs to take its decision (thus ecologically-friendly).

Experimental methodology

Adastop: ingredients

- a test that requires a minimal number of runs to take its decision $\rightsquigarrow$ sequential/adaptive test.

Experimental methodology

Adastop: ingredients

- a test that requires a minimal number of runs to take its decision $\leadsto$ sequential/adaptive test.
- Performance is not Gaussian $\leadsto$ non parametric test: permutation test (compares distributions).

Experimental methodology

Adastop: ingredients

- a test that requires a minimal number of runs to take its decision $\rightsquigarrow$ sequential/adaptive test.
- Performance is not Gaussian $\rightsquigarrow$ non parametric test: permutation test (compares distributions).
- more than 2 agents $\rightsquigarrow$ multiple hypothesis testing.

Experimental methodology

Adastop: ingredients

- a test that requires a minimal number of runs to take its decision $\rightsquigarrow$ sequential/adaptive test.
- Performance is not Gaussian $\rightsquigarrow$ non parametric test: permutation test (compares distributions).
- more than 2 agents $\rightsquigarrow$ multiple hypothesis testing.
- We usually make n runs in parallel $\rightsquigarrow$ group testing.

Experimental methodology

Adastop: ingredients

- a test that requires a minimal number of runs to take its decision $\rightsquigarrow$ sequential/adaptive test.
- Performance is not Gaussian $\rightsquigarrow$ non parametric test: permutation test (compares distributions).
- more than 2 agents $\rightsquigarrow$ multiple hypothesis testing.
- We usually make n runs in parallel $\rightsquigarrow$ group testing.

Idea of the algorithm: given a budget KN:

Initialization: run each agent N times.

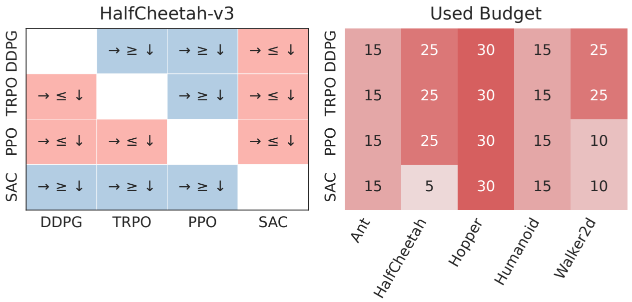
Test: can agents be ranked at risk α?

If some agent can not be ranked, perform an other N runs, merge these new results with previously collected performances, and loop to test step. Stop looping when budget exhausted.

Experimental methodology

Adastop

Comparison of 4 different agents drawn from 4 different RL libraries, on 5 Mujoco tasks.



Experimental methodology

Adastop

- Adastop is the first sound statistical test to compare the experimental performance of a set of agents, trying to minimize the number of runs.
- Performance is whatever you want to compare the agents with.
- Adastop is not restricted to RL. Can be used in many experimental studies, not necessarily related to ML.
- Extensions: how to compare agents performing on a set of tasks (*e.g.* Atari games): open problem for now.

The paper: `https://inria.hal.science/hal-04132861/`
The code: `https://github.com/TimotheeMathieu/adastop`

Few last words

Few last words

- There are plenty of exciting fields of applications of RL, with challenges for theoreticians and pratitioners, fun applications, and serious ones.

Few last words

- There are plenty of exciting fields of applications of RL, with challenges for theoreticians and pratitioners, fun applications, and serious ones.

- We make available open source software to challenge the RL community with real problems: use it!

  We provide other original tasks on our website: see the top of `https://team.inria.fr/scool/publications/` and more to come.

Few last words

- There are plenty of exciting fields of applications of RL, with challenges for theoreticians and pratitioners, fun applications, and serious ones.
- We make available open source software to challenge the RL community with real problems: use it!

  We provide other original tasks on our website: see the top of `https://team.inria.fr/scool/publications/` and more to come.
- Stop the rush: experimental results are important but need to be scientifically established.

Thanks for your attention.

We hire!

profiles ranging from theory to applications.
Interns, PhD. student, post-docs, engineers, permanent staff.

Check out `https://team.inria.fr/scool/`.
Get in touch: `philippe.preux@inria.fr`.