Introduction to Reinforcement Learning

ML on quantum and classical data

Gorka Muñoz-Gil (**ICFO**)

Master in Photonics (UPC-UAB-UB-ICFO), 2018-2019

Bureaucracy

Homeworks

Prof. Lewenstein homeworks: in hand **22/11** (last lecture).

Then, for each topic (Deep ML, RBM and RL) handle:

- Short summary of your homework (1 or 2 page max.). Comment on the goal, tools you used (ML method, architecture,...) and add results (images and plots asked).
- Notebook (colab or jupyter) with the **commented** code and results.
- Send each of the homeworks before **24/11** to
  - Deep ML: alexandre.dauphin@icfo.eu
  - RBM: patrick.huembeli@icfo.eu
  - RL: gorka.munoz@icfo.eu

Exam (28/11)

Four or five general questions (qualitative) on the topics presented.

Final mark: 0.4 x Practical HW + 0.2 x Lewenstein's HW + 0.4 x Exam

Recap Supervised vs. Unsupervised Learning

Supervised learning

Data is labeled: photo + some information of what is on it (cat or dog?)

Examples:

- Regression methods (SVM,...)
- Image classification with NN

Unsupervised learning

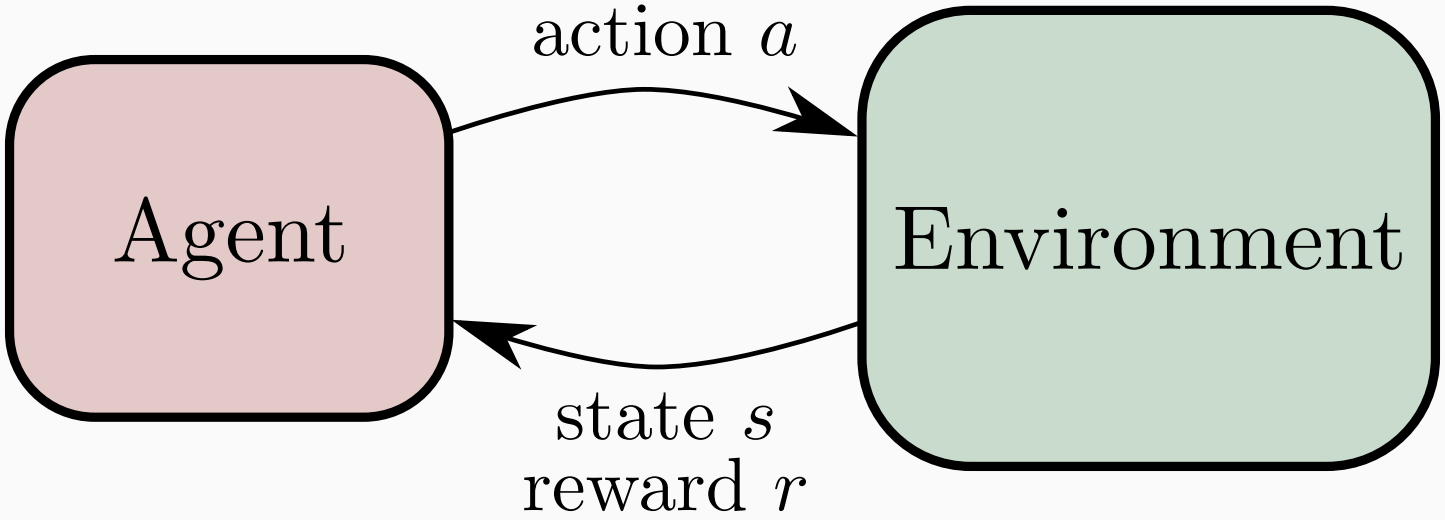
Data is not labeled, we want to find patterns or reconstruct some samples without extra information.

Examples:

- Autoencoders
- Boltzmann machines (energy models)
- Clusterization algorithms (tSNE,...)

Reinforcement Learning

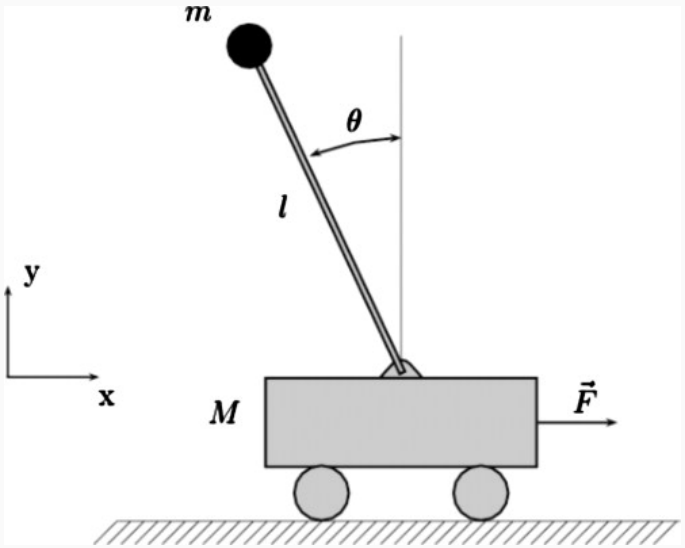
Nor supervised nor unsupervised: a mix of both!



The goal of the agent is to maximize the reward over long interactions with the environment.

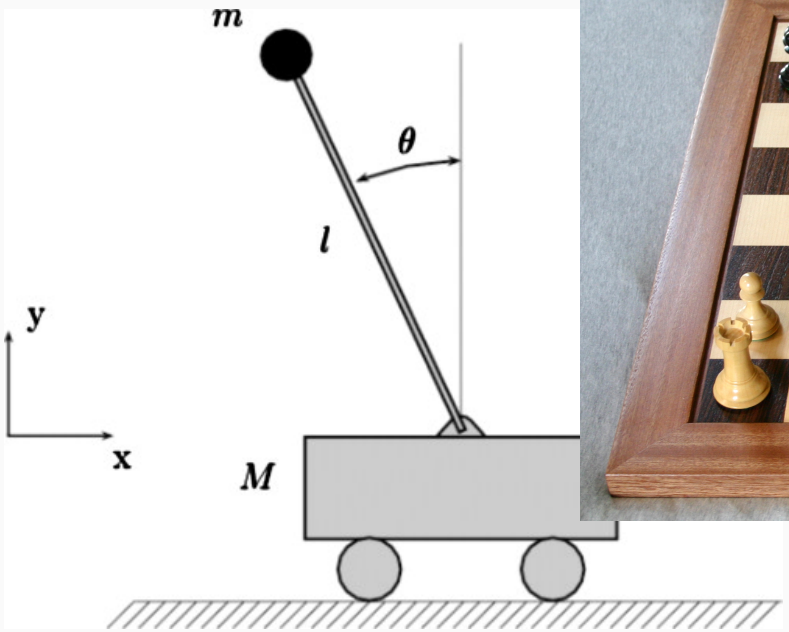
What is an environment?

As simple as



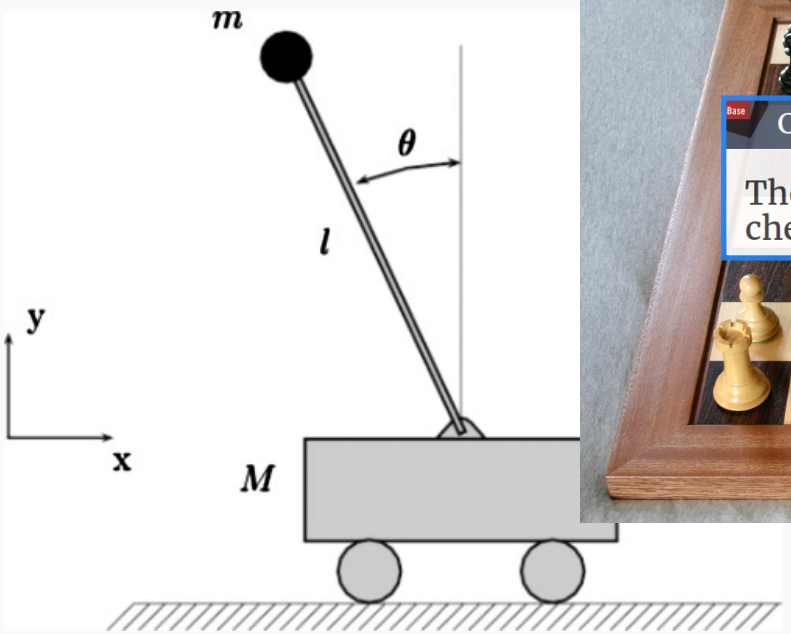
What is an environment?

More complex



What is an environment?

More complex

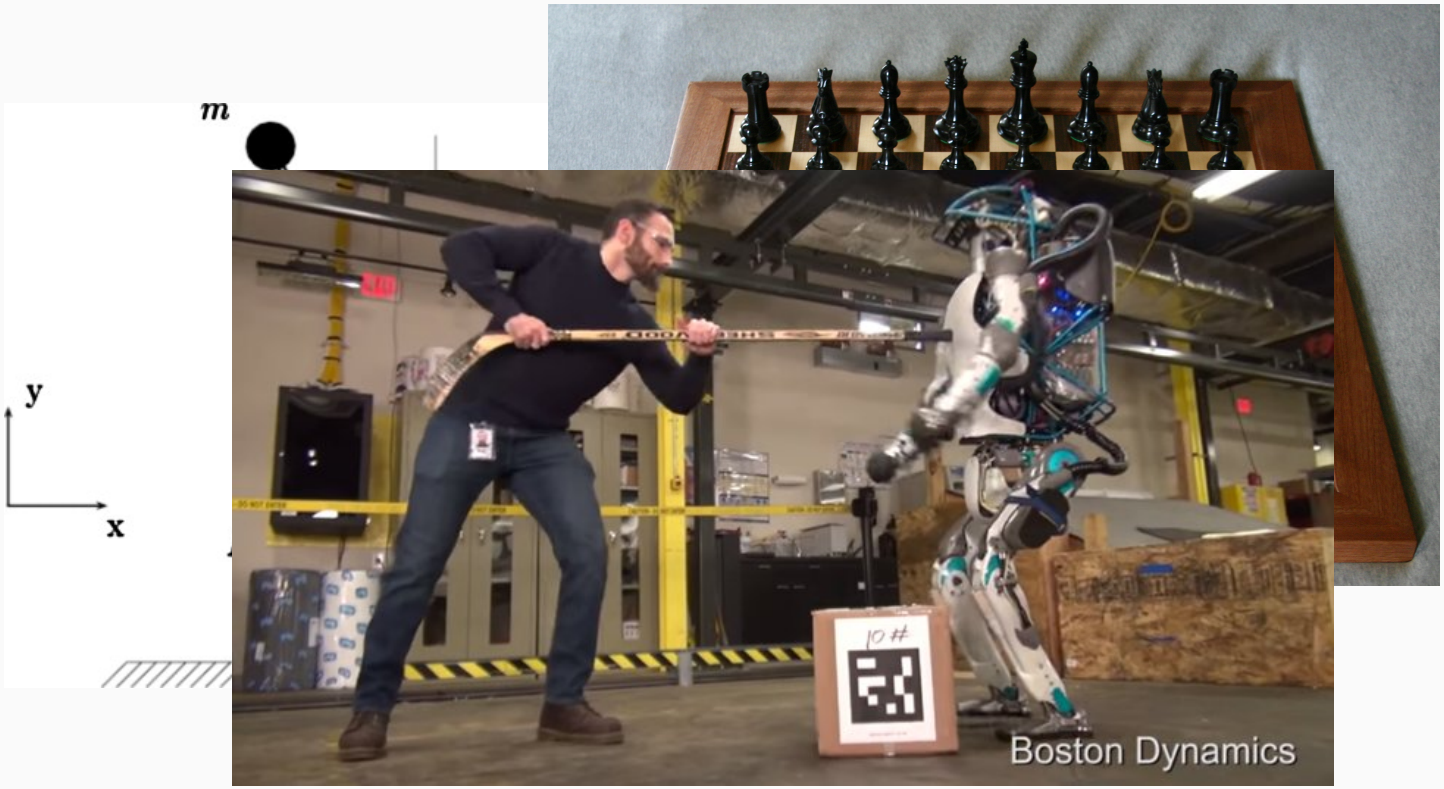


What is an environment?

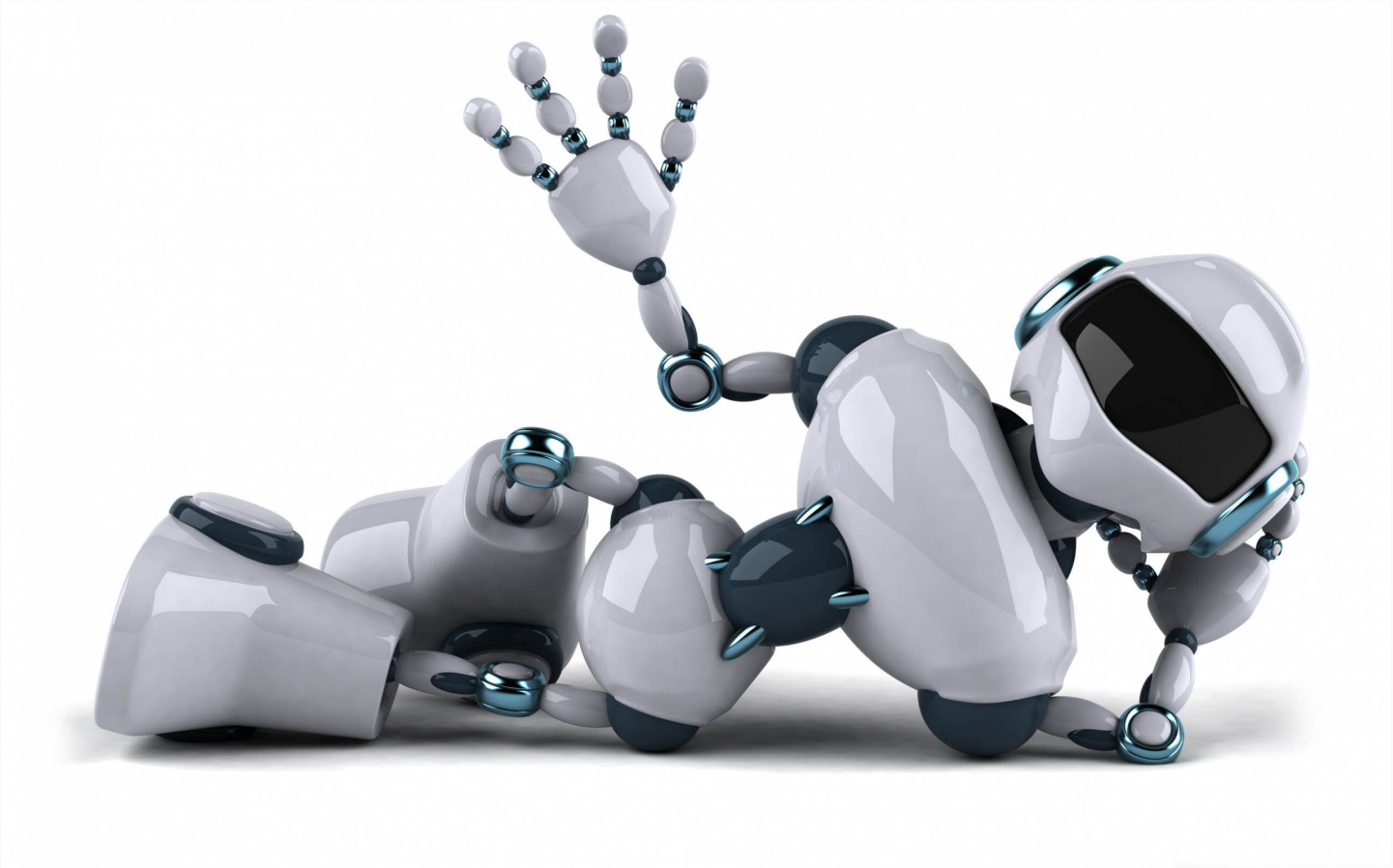
More complex

What is an environment?

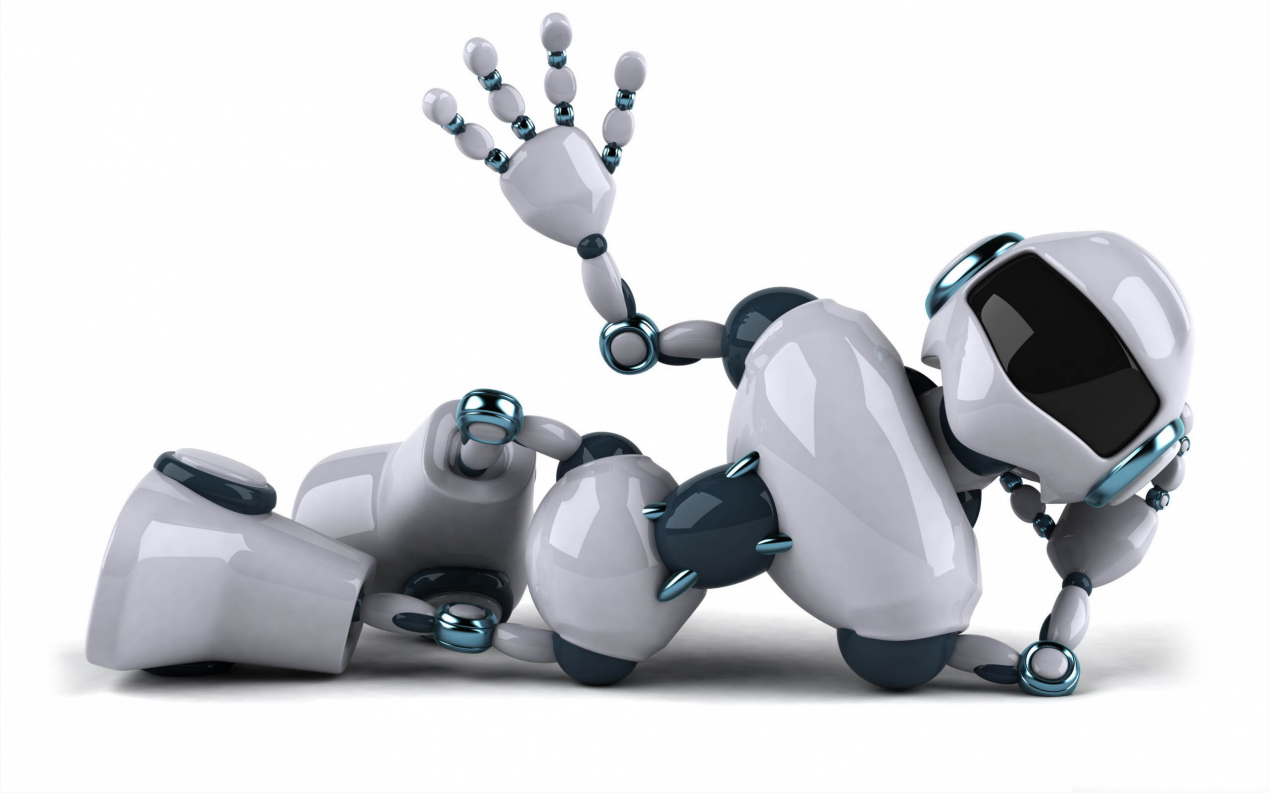
More more complex



What is an agent?



What is an agent?



Indeed, more boring... We will see that it can be from a simple table (Q-Table), a neural network (Q-learning).

Both will be used to approximate a probabilistic function (policy) that tells how to act given the state of the environment .

What problems can we solve?

Classical

- Learn how to play games from the easier (cartpole, Frozen lake) to more complex (Doom, Go, Dota,...).
- Robotics: learn to move, complete tasks, be autonomous,...
- Optimize memory control, personalized web services,...

Quantum

- Huge effort on the generalization of RL in the 'quantum realm' (see V. Dunjko, arxiv:1610.08251)
- Quantum control: prepare physical systems in desired quantum state (1705.00565v2, 1805.00654).
- Preparation of quantum experiments (projectivesimulation.org)
- Certification of many-body properties assisted by machine learning (solvind the quantum marginal problem, soon!)

(Tentative) Plan of the course

1. (Day 1) Introduction to Reinforcement Learning
    - Basisc elements of RL
    - Multi-armed bandits
2. (Day 2) Markov decision processes
3. (Day 2) Q-learning: tables and networks
    - Frozen Lake example: tables and networks

Basic elements of Reinforcement Learning

Beyond the agent and the environment, we identify four main subelements:

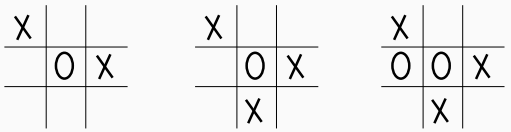
- Policy: ensemble of rules that set the behaviour of the agent.
- Reward signal: defines the goal in a RL problem.
- Value function: specifies what is good on the long run.
- Model: Mimics the responses of the environment.

We are not considering here action and states, which will be explored in the MDPs.

Basic elements of Reinforcement Learning

Example: Tic-tac-toe problem

We construct a vector of values, which is our *value function* V:



$$V \rightarrow \boxed{v(s_k) \mid v(s_j) \mid v(s_i) \mid \cdots}$$

Reward: $v_i \rightarrow 1$ if the current state wins, $v_i \rightarrow 0$ if the current state looses.

Policy: We choose the action that takes us to the state with higher v_i.

How to update V?

$$V(s) = V(s) + \alpha \left[V(s') - V(s) \right]$$

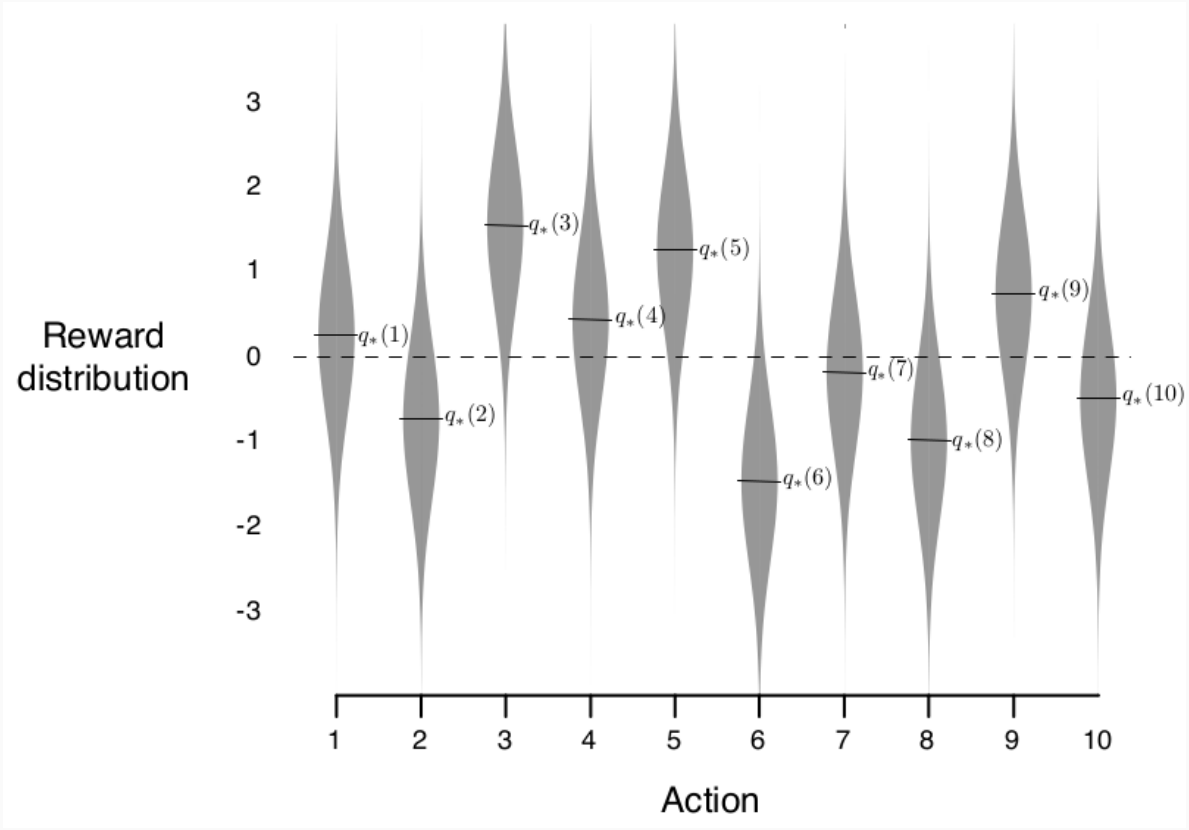
Multi-armed bandits

Multi-armed bandits

SR: 50% SR: 30% SR: 60% SR: 80% SR: 50%

Multi-armed bandits

SR: 50% SR: 30% SR: 60% SR: 80% SR: 50%



Multi-armed bandits

In a k-armed bandit, each of the k actions has an expected reward $\rightarrow$ we will call this the *value* of the action. More formally:

$$q_*(a) = \mathbb{E}\left[R_t | A_t = a\right]$$

We don't know a priori $q_*(a)$. We denote the estimate as $Q(a)$

$\rightarrow$ **Goal:** $Q_t(a)$ to be close to $q_*(a)$!

A simple way of building $Q_t(a)$ would be as follows:

$$Q_t(a) = \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \boxed{\frac{\sum_{i=1}^{t-1} R_i \delta(A_i - a)}{\sum_{i=1}^{t-1} \delta(A_i - a)}}$$

We choose actions using $\boxed{A_t = \text{argmax}_a Q_t(a)}$. To allow exploration, we use ε-greedy method.

Incremental implementation

To avoid memory blow-out, we don't want to calculate $\sum_{i=1}^{t-1} R_i \delta(A_i - a)$ at each step. Instead, we use (now $Q_n(a) \to Q_n$):

$$Q_{n+1} = ... = Q_n + \frac{1}{n}[R_n - Q_n]$$

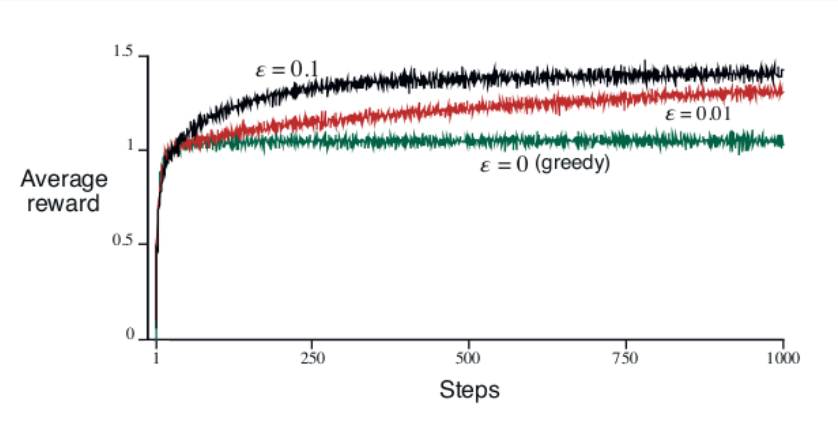
Non-stationary problems

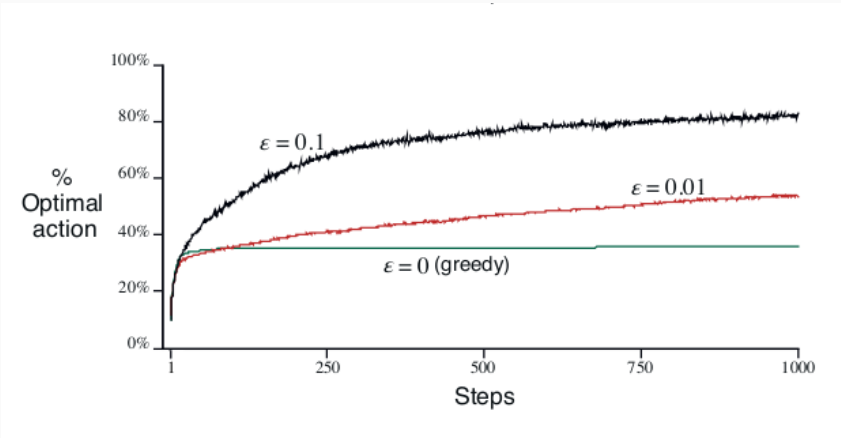
We often encounter problems that evolve in time. Then, it makes more sense to give higher weights to recent rewards than to long-past rewards.

$$Q_{n+1} = Q_n + \alpha[R_n - Q_n] = ... = (1-\alpha)^n Q_i + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

Multi-armed bandits: hands-on problem

Exercise: Solve a 'Gaussian' 10-bandit problem. Compare the results with $\varepsilon = [0, 0.01, 0.1]$. Plot the average reward at each episode over $\sim$2000 runs to find similar results as





(details in github)

Reinforcement Learning and Quantum Physics

Now that we know some basics on RL, let us see how it is connected to quantum. As in all (Q)ML, there are two directions:

- Quantum enhanced RL algorithms (real QML!):
  - *Quantum speedup for active learning agents*, (1401.4997)
  - Implementation: *Speeding-up the decision making of a learning agent using an ion trap quantum processor*, (1709.01366).
  - *Quantum-enhanced reinforcement learning for finite-episode games with discrete state spaces* (1708.09354).
- Classical RL applied to quantum problems:
  - *Active learning machine learns to create new quantum experiments* (1706.00868)
  - *Reinforcement Learning in Different Phases of Quantum Control* (1705.00565)

Quantum enhanced RL algorithms

Goal:

1. Find whether quantum algorithms can give rise to speed up in learning.
2. Construct algorithms which can work directly on quantum data

How is it done? Normally, part of the *classical* algorithm is substituted by its quantum counterpart (matrix inversion, Grover's algorithm,...).

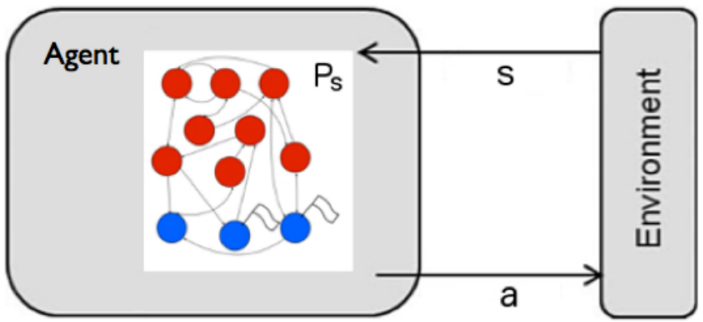
One can also rephrase the problem quadratic unconstrained binary optimization (QUBO), which can be solved in D-Wave or other annealers.

Quantum enhanced RL algorithms

Quantum Speedup for Active Learning Agents

Giuseppe Davide Paparo,[1] Vedran Dunjko,[2,3,4] Adi Makmal,[2,3] Miguel Angel Martin-Delgado,[1] and Hans J. Briegel[2,3]

Can quantum mechanics help us build intelligent learning agents? A defining signature of intelligent behavior is the capacity to learn from experience. However, a major bottleneck for agents to learn in real-life situations is the size and complexity of the corresponding task environment. Even in a moderately realistic environment, it may simply take too long to rationally respond to a given situation. If the environment is impatient, allowing only a certain time for a response, an agent may then be unable to cope with the situation and to learn at all. Here, we show that quantum physics can help and provide a quadratic speedup for active learning as a genuine problem of artificial intelligence. This result will be particularly relevant for applications involving complex task environments.



Nonetheless, while the physical nature of the agent and the environment prohibits speedup through quantum queries of the environment, the physical processes within the agent, which lead to the performed actions, can be significantly improved by employing full quantum mechanics [27]. In particular, the required internal time

The agent's internal "program" is realized by physical processes that correspond to quantum walks. These quantum walks are derived from classical random walks over directed weighted graphs that represent the structure of its episodic memory. We have shown, using quantum coherence and known results from the study of quantum walks, how the agent can explore its episodic memory in superposition in a way that guarantees a provable quadratic speedup in its active learning time over its classical analog.

Classical RL applied to quantum problems

Goal: Use state of the art RL algorithms to solve quantum physics problems (normally very complex).

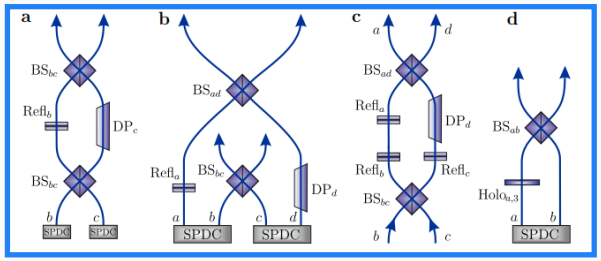
How is it done? There exist a big amount of work in this direction. Main focus is having machines that learn how to interact with a *laboratory*, to produce desired states.

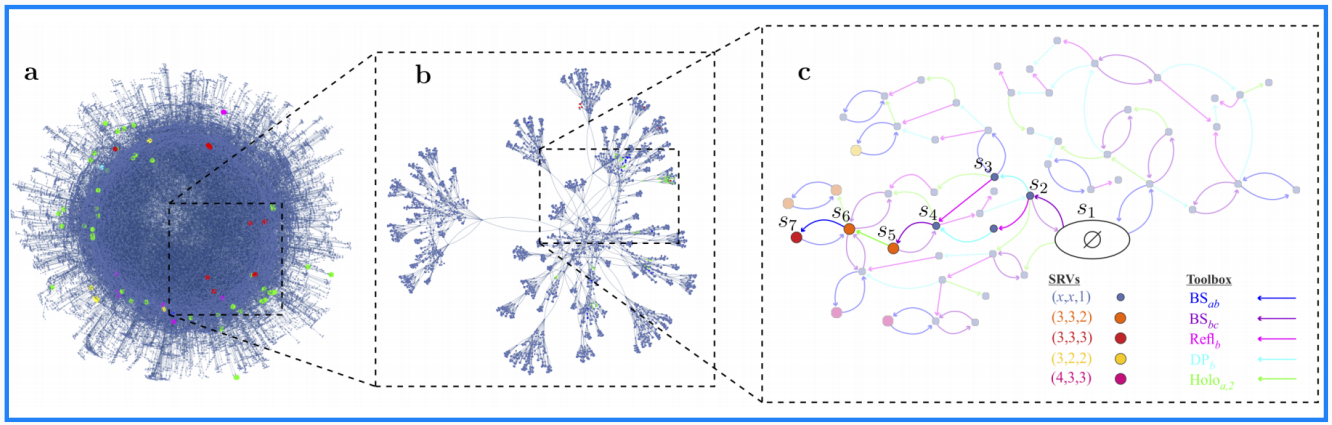
Classical RL applied to quantum problems

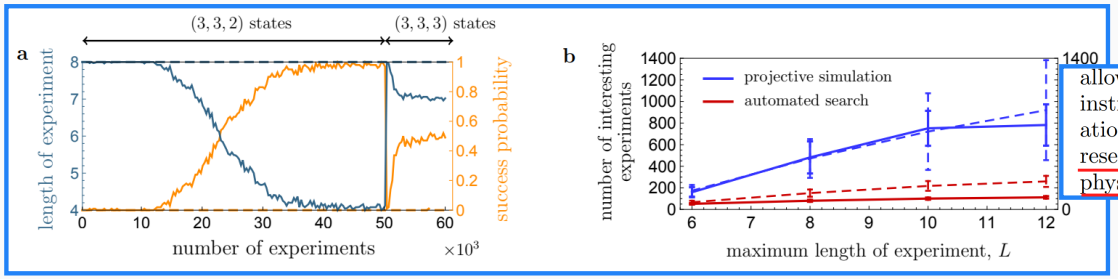
Active learning machine learns to create new quantum experiments

Alexey A. Melnikov,[1,*] Hendrik Poulsen Nautrup,[1,*] Mario Krenn,[2,3] Vedran Dunjko,[1,4] Markus Tiersch,[1] Anton Zeilinger,[2,3] and Hans J. Briegel[1,5]

How useful can machine learning be in a quantum laboratory? Here we raise the question of the potential of intelligent machines in the context of scientific research. A major motivation for the present work is the unknown reachability of various entanglement classes in quantum experiments. We investigate this question by using the projective simulation model, a physics-oriented approach to artificial intelligence. In our approach, the projective simulation system is challenged to design complex photonic quantum experiments that produce high-dimensional entangled multi-photon states, which are of high interest in modern quantum experiments. The artificial intelligence system learns to create a variety of entangled states, and improves the efficiency of their realization. In the process, the system autonomously (re)discovers experimental techniques which are only now becoming standard in modern quantum optical experiments – a trait which was not explicitly demanded from the system but emerged through the process of learning. Such features highlight the possibility that machines could have a significantly more creative role in future research.







allows machines to tackle problems they were not directly instructed or trained to solve. This supports the expectation that AI methodologies will genuinely contribute to research and, very optimistically, the discovery of new physics.