Visualization of Deep Reinforcement Learning

Théo Jaunet
LIRIS, Team SICAL & Imagine

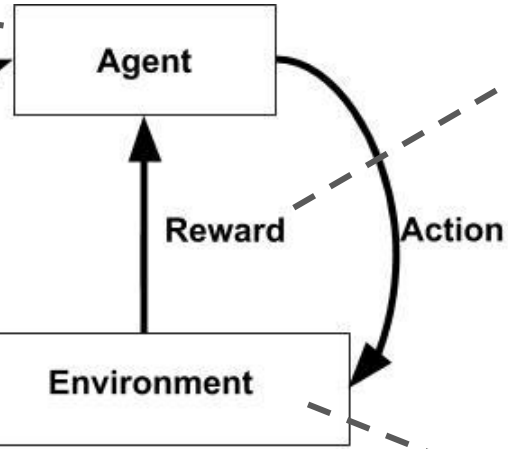
Advisors : **Romain Vuillemot** (SICAL) & **Christian Wolf** (Imagine/INRIA Chroma)

Deep learning and **robotics** have a tremendous number of potential applications in our **daily life**.

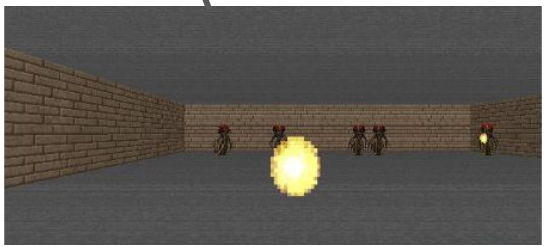
However, **legal** and **trust** issues, among others, restrain its **democratization.**

To tackle those issues, contributions in **Explainable** and **Interpretable** Machine learning started to emerge within ML communities.

We trained robots (we will call **agents**) on the **VizDoom** simulator using Deep Reinforcement Learning (DRL) methods.



+0.01	Living 1 step
-3	dying



Glossary

Model: Neural Network

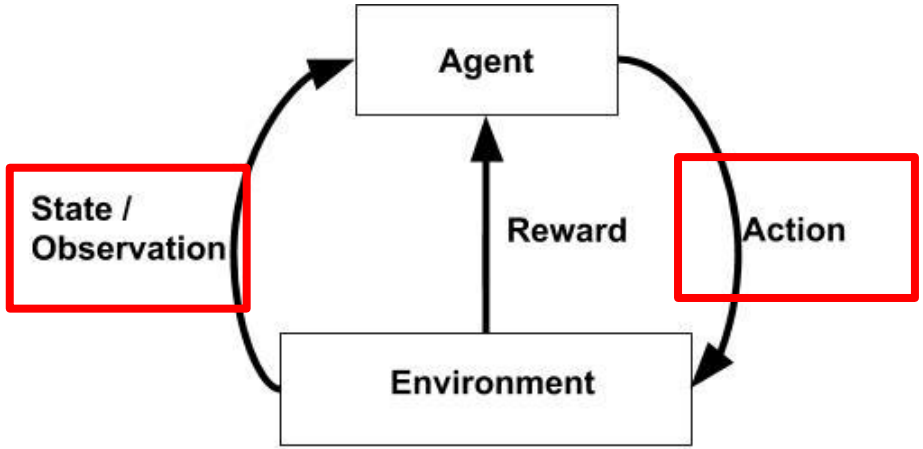
Parameters: Learnable features (e.g., weights)

Hyper-parameters: (Non-learnable)* features configured by the developer, (e.g., number of layers)

Episode: Complete sequence from the beginning of the game until the agent reaches a terminal state (e.i., death, win, stuck, timeout, etc...)

* arguable

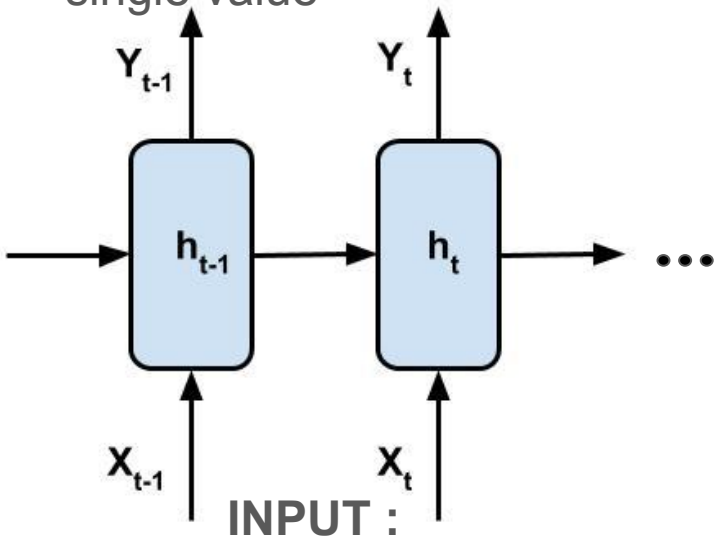
The deep purpose in the Deep Reinforcement Learning.



- Ease the computational needs for fittest action per state calculus.
- Generalize and extract features directly from a picture.

We added a **recurrent layer** as **long term memory**, and therefore, ease the time related information processing.

OUTPUT : Sequence or single value



Hidden state: Vector

INPUT : (Sequence)

- Updated at each steps of an episode.
- Continuous values in range [-1,1].

It is difficult to build those models as experts need to make many design decisions often driven by **intuition**.

- How to set **Hyper-parameters** ?
- How many **layers** ?
- What **dimension** ?

To configure models

- How to **evaluate** a trained agent **performance** ?
- How to **justify** a **decision** ?
- How to **improve** the configuration ?

To understand the performance of a configuration

Where to gather **DRRL model** features such as **inputs**, **saliency maps**, **actions** distribution, inner **states**.