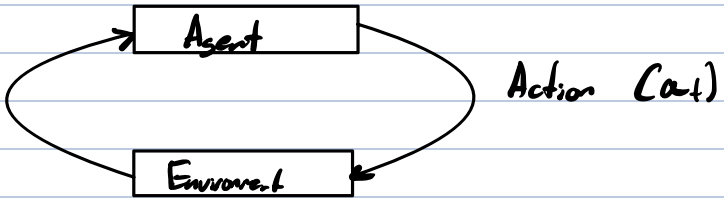
Source: SpinningUp OpenAI

Part 1: Key Concepts in RL

~~What can RL do?~~
↳ Youtube Video (RL in Learning Dexterity)

Key Concepts and Terminology:

Figure 1:



At every step the agent sees a (possible partial) observation of the state of the world.
↳ then then decides on an action to take on it.

Following the action, the environment changes possibly as a result of an algorithm called "self-play".

The agent also perceives reward, a signal demonstrated by the environment — a number that determines how good or bad the current world state is.

The agent's goal is to maximize the cumulative reward called the return.

States and Observations:

A state, denoted by 's' is a complete description of the world.
↳ no information is hidden from the state

An observation 'o' is a partial description of a state, which may omit info.

Note: In deep RL states and observations are represented by real-valued vector, matrix, or higher-order tensor.

ex: A visual observation could be alternatively represented by the RGB matrix of the pixel values.

When an agent is able to observe the complete state of the environment, we say the environment is fully observed.

When the agent can only see a partial observation, we say the environment is partially observed.

Action Spaces:

The set of all valid actions in a given environment is called the action space.

Some environments like Atari and Go have discrete action spaces.
While, others where an agent controls a robot in a physical space have continuous action spaces.

Policies

A policy is a rule used by an agent to decide what actions to take.

When the policy is deterministic it is denoted by μ:

$$a_t = \mu(s_t)$$

When stochastic, denoted by π:

$$a_t \doteq \pi(\cdot|s_t)$$

Policy and agent are often used interchangably, like:
"The policy is trying to maximize the reward."

In deep RL (dRL) we deal with parametrized policies. Policies where outputs are computable functions that depend on a set of parameters (eg. weights and biases) which we can adjust to change the behaviour via some optimization algorithm.

Denoting the parameters of a parametrized policy using θ or ϕ:

$$a_t = \mu_\theta(s_t)$$
$$a_t \doteq \pi_\theta(\cdot|s_t)$$

Deterministic Policies

Stopped here... there are examples that require installation ... installion is incredibly deprecated and a rabbit hole.

Moving onto Jay Alammar's

Hands-On Large Language Models