RL Basics

Alicja Ziarko

1 RL is Cool

- Learning to play games like Atari https://arxiv.org/abs/1312.5602
- Playing StarCraft https://arxiv.org/abs/1708.04782
- Alignment for LLMs https://arxiv.org/abs/1706.03741
- Enabling reasoning in LLMs - really cool and practical https://arxiv.org/pdf/2412.19437

2 RL basics

2.1 Definition

Definition 1. *Reinforcement Learning is a set of Algorithms for Solving Decision Processes with a temporal component.*

Definition 2. *Reinforcement Learning is a special type of Machine Learning - where datapoints are not sampled independently and they are only indirectly labeled through reward.*

RL Hypothesis 1. *Any goal can be formalized as the outcome of maximizing total rewards.*

RL Hypothesis 2. *Intelligent behavior arises from the actions of an individual seeking to maximize its received rewards signals in a complex and changing world.*

2.2 RL feedback loop

To formalize RL, we can think about some basic objects (illustrated in 1):

- Agent - can perform actions in the environment - for instance a player in a game
- Environment - The world, that the agent can influence by their actions
- State - what agent receives from the environment after performing an action. Could be an image for instance.
- Reward - In addition to a new state, agent also receives the reward for arriving into that new state.

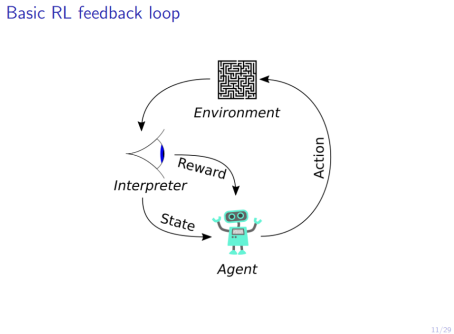

Figure 1: A simple example of an RL feedback loop

3 RL Components in More Details

3.1 Environment

The environment can be:

- A physics simulator
- A simple Atari game
- A complicated game, like StarCraft
- A board game, like Go
- A person who responds to an LLM's messages

Agent interacts with the enviroment, through selecting actions to perform. The environment can be in any state $s \in \mathcal{S}$. The initial state s_0 of the environment is sampled from a distribution $\mathbb{P}_0(\cdot)$. The enviroment accepts any action $a \in \mathcal{A}$. The environment returns to the agent an observation s', which might or might not be the same as the state the environment is in. For instance, observation can be just the image of the game, and might not contain information about velocities of different objects, only their positions. Environment has a transition function $T : (\mathcal{S} \times \mathcal{A}) \to \mathcal{S}$, which might or might not be deterministic. There is moreover a reward function $\mathcal{R} : \mathcal{S} \to \mathbb{R}$, which based on the state of the environment, returns a reward. Environment also has a notion of time, which is incremented by one after each action performed by the agent, starting from 0.

3.2 Agent

Agent uses the policy, in order to decide which actions to select.

3.3 Policy

Policy is the strategy that the agent is using in order to choose an action based on the state received from the environment.

Definition 3. *Policy π is any function mapping a state to a probability distribution on the action space, $\pi : s \mapsto \mathbb{P}(\mathcal{A}|s)$*

3.4 Value Function

Let's denote by a_t the t^{th} action performed by the agent, s_t the state of the enviroment preceeding that action, and $r_t = \mathcal{R}(s_t)$. Furthermore, let's denote the return from time t as $G_t = \sum_{s>t} r_s$ and trajectory $\tau = s_1 a_1 s_2 a_2 \dots$.

Value function is the expected reward starting from a given state, more formally:

Definition 4. *Value function is a function $V : \mathcal{S} \to \mathbb{R}$, defined by $V(s_t) := \mathop{\mathbb{E}}_{\tau}[G_t|s_t]$.*

3.5 Q-Function

State value function is the expected reward starting from a given state action pair, more formally:

Definition 5. *State value function is a function $V : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, defined by $Q(s, a) := \mathop{\mathbb{E}}_{\tau}[G_t|s_t, a_t]$.*

3.6 Discounting

Humans prefer to get a reward fast, instead of getting it in a thousand years. Following this intuition, we introduce the notion of a discount factor γ. Then we calculate the return as $G_t = \sum_{s>t} \gamma^s r_s$, instead of using $G_t = \sum_{s>t} r_s$.