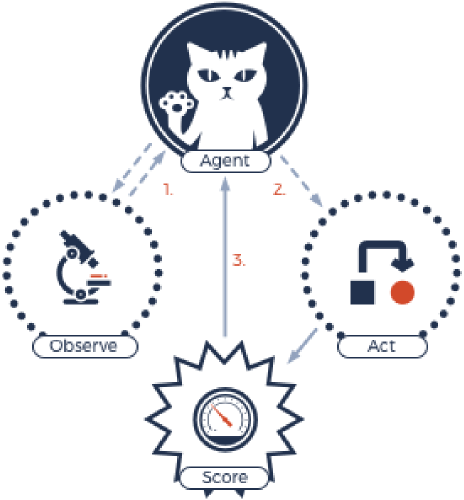
Reinforcement Learning

Carrot or stick?

Reinforcement Learning



Components of RL

- Agent
- Environment
- State
- Actions
- Policy
- Returns
- Exploration & Exploitation

Agent

The model that acts and learns within our environment.

Environment

The limited space that the agent exists within that defines the rules of what is possible.

State

The relevant information about the environment with respect to the agent's actions and results.

- State is the true reality. The agent only has access to observations.
- A state is *fully observed* if the observations represent the full state. *Partially observed* otherwise.

$$\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t).$$

Actions

The agent can perform actions that alter the state.

- Discrete vs Continuous actions

Policy

The set of rules an agent follows to determine the action it should take given its observations about the environment.

Returns

The environment provides a returned value to the agent after it takes an action; reward or punishment.

We evaluate a model based on the total rewards at the end of the simulation or at time τ using,

$$\underset{\textit{finite}}{R(\tau)} = \sum_{t=0}^{T} r_t, \quad \underset{\textit{infinite}}{R(\tau)} = \sum_{t=0}^{\infty} \gamma^t r_t.$$

Exploration vs Exploitation

We want to balance between learning new things (exploration) and utilizing the knowledge we already have (exploitation).

RL Objective

$$V^{\pi}(s) = \underset{\tau \sim \pi}{\mathrm{E}}[R(\tau) | s_0 = s],$$
$$Q^{\pi}(s, a) = \underset{\tau \sim \pi}{\mathrm{E}}[R(\tau) | s_0 = s, a_0 = a].$$

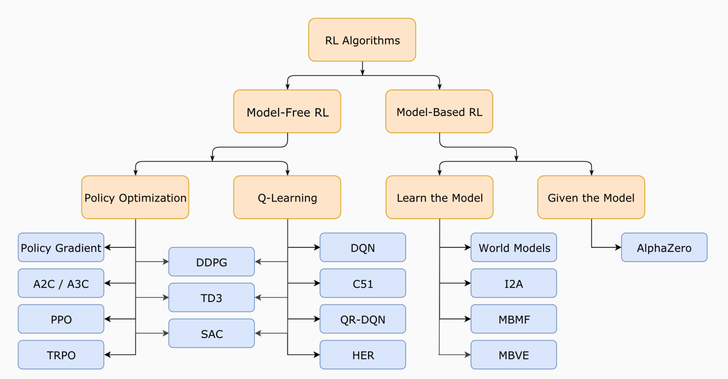
Bellman Equations

$$
\begin{aligned}
V^{\pi}(s_t) &= \mathrm{E}\left[r(s_t, a_t) + \gamma V^{\pi}\left(s_{t+1}\right)\right], \\
Q^{\pi}(s_t, a_t) &= \mathrm{E}\left[r(s_t, a_t) + \gamma \mathrm{E}\left[Q^{\pi}\left(s_{t+1}, a_{t+1}\right)\right]\right].
\end{aligned}
$$

Considerations with RL

- Importance of reward functions - see ”Curiosity Driven Learning”
- Over fitting and learned helplessness

Types of RL


RL Algorithms
Model-Free RL
Model-Based RL
Policy Optimization
Q-Learning
Learn the Model
Given the Model
Policy Gradient
A2C / A3C
PPO
TRPO
DDPG
TD3
SAC
DQN
C51
QR-DQN
HER
World Models
I2A
MBMF
MBVE
AlphaZero

Q Learning

Approximate action-value function $Q(s, a)$

- Create a Q table
- Select actions and update Q values
- Repeat for many iterations
- Select highest scoring action in practice

Q Table

Actions :	↑	→	↓	←
Start				
Nothing / Blank				
Power				
Mines				
END				

ϵ-Greedy Action Selection

$$a_t = \begin{cases} \arg\max_a Q(s_t, a) & P(1-\epsilon) \\ \text{random action} & P(\epsilon) \end{cases}$$

Updating Q

$$Q_{\text{new}}(s, a) = (1 - \alpha)Q(s, a) + \alpha(r_t + \gamma \max_a Q(s_{t+1}, a)),$$

where α is the learning rate and γ is our discount rate.

Using Q Table

Given a state and action, select the highest scoring action.

Questions