RUTGERS

THE STATE UNIVERSITY
OF NEW JERSEY

Machine Learning

Reinforcement Learning

Edgar Granados

Reinforcement Learning

Definition

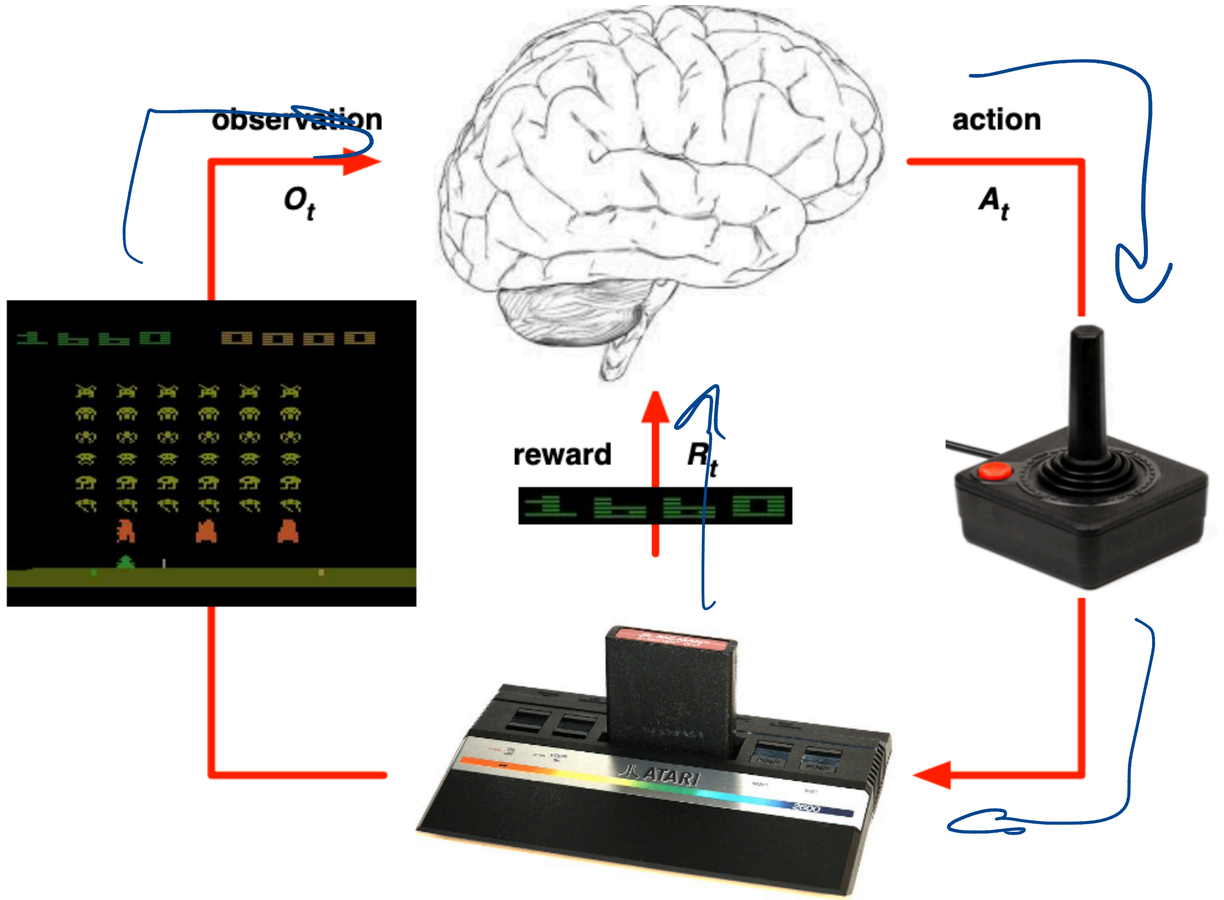
"A way of programming agents by reward and punishment without needing to specify how the task is to be achieved." [L. Kaelbling, M. Littman and A. Moore, 1996]

Reinforcement Learning

Videogames

→ Rules unknown

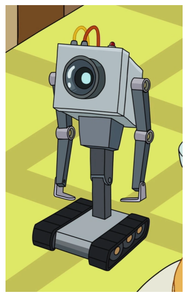
→ Learning from outcomes

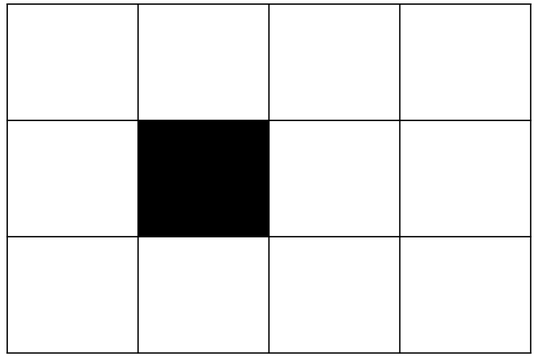


Reinforcement Learning

Behavioral Psychology

Reinforcement Learning





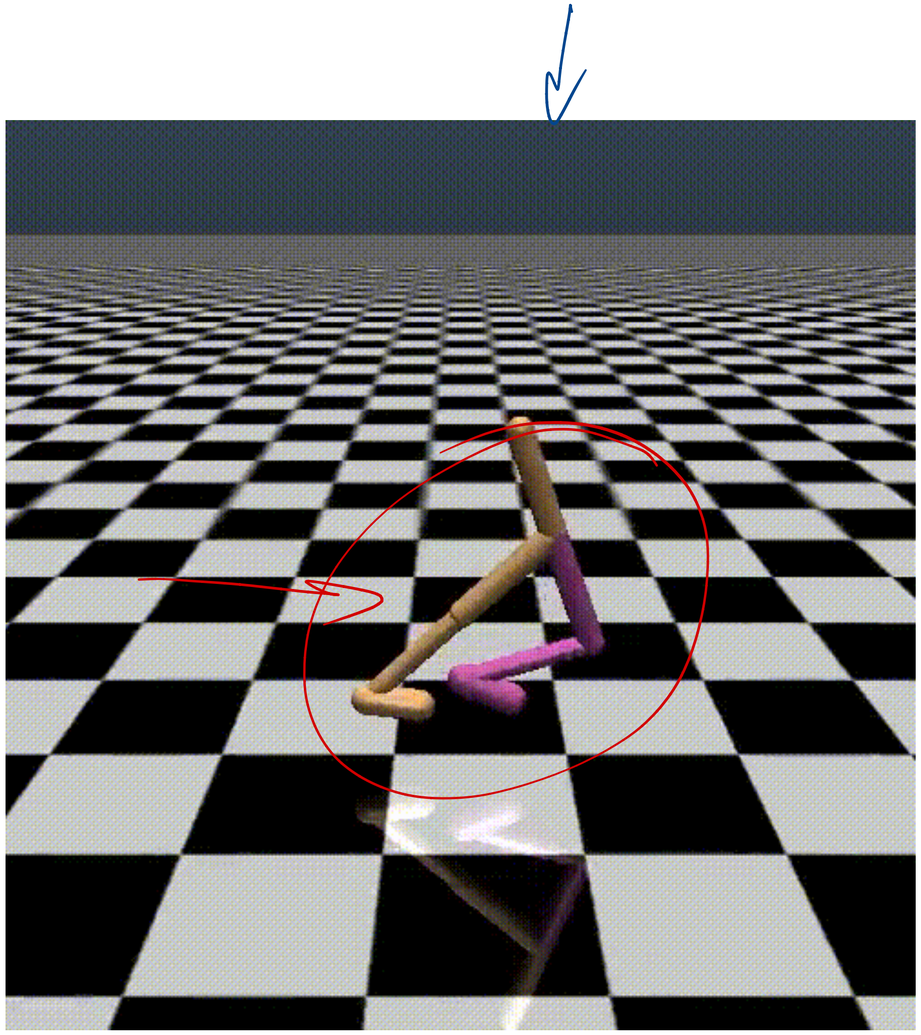
Action

Reward

Observation

Defining Rewards

- Helicopter Maneuvers
- Playing games
- Investment Portfolio
- Power station
- Humanoid walk



Horizons

- Given a reward function, the goal of the agent is to maximize the expected cumulated reward over some number H of steps, called the horizon.

$(s_t, a_t, r_t), (s_{t+1}, a_{t+1}, r_{t+1}) \ldots (s_{t+H-1}, a_{t+H-1}, r_{t+H-1})$

Horizon

Horizons

Finite or infinite?

- If the horizon is infinite, the optimal actions depend only on the state. In our case, the optimal action at any step is to move toward the goal.

$0 < \gamma < 1$

$$r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + (\ldots)$$

Learning with Markov Decision Processes

- Optimal policy without transition function?

- RL:
  - Finding the optimal policy for an MDP with unknown transition function
  - Learn from experience: Act and then observe the receiving rewards

trial and Error

Model-based Approach

- Collect Data → $\{(s_t, a, s'), (\ldots)\}$
- Estimate T:
  - $T(s, a, s') = P(s'|s,a) \approx \dfrac{\#(s,a,s') \in \text{Data}}{\#(s,a,\text{Anything}) \in \text{Data}}$

Get some estimate of T

Do MDP { Policy iter / Value iter

Model-free Approach

- Learn policy directly from rewards
  - No need to learn the transition function

Req More data

Q-value

- A Q-value is the expected sum of rewards that an agent will receive if it executes action a in state s then follows a policy π for the remaining steps

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') V^{\pi}(s')$$

a not necessarily $\pi(s)$

- Value function: (MDP)

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t \geq 0} \gamma^t r_t | s_0 = s, \pi\right]$$

How good is a state?

- Q-value function

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{t \geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi\right]$$

the state-action pair

The Q-learning algorithm

- Q-value updates?

$$Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a') \right]$$

The Q-learning algorithm

- Compute averages as we go:
  - Sample transition (s, a, r, s')
  - $Q(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha [r + \alpha \max_{a'} Q(s', a')]$
  - $\pi(s_t) = \arg\max_{a \in \mathcal{A}} Q^t(s_t, a);$

$Q(s,a) \approx r + \gamma \max_{a'} Q(s',a')$

$0 < \alpha < 1$, decreases over time

Alternative Formulation of the Update Equation

$$Q^{t+1}(s_t, a_t) \leftarrow Q^t(s_t, a_t) + \alpha_t[R(s_t, a_t) + \alpha \max_{a' \in A} Q(s_{t+1}, a') - Q^t(s_t, a_t)]$$

Observed Value

Predicted Value

New Value = Old Value + (learning Rate)(Observed Value − Predicted Value)

Temporal Difference TD

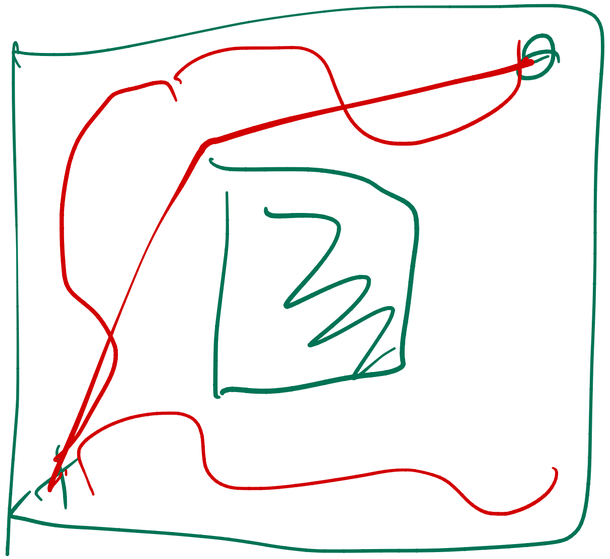
Convergence conditions of tabular Q-learning

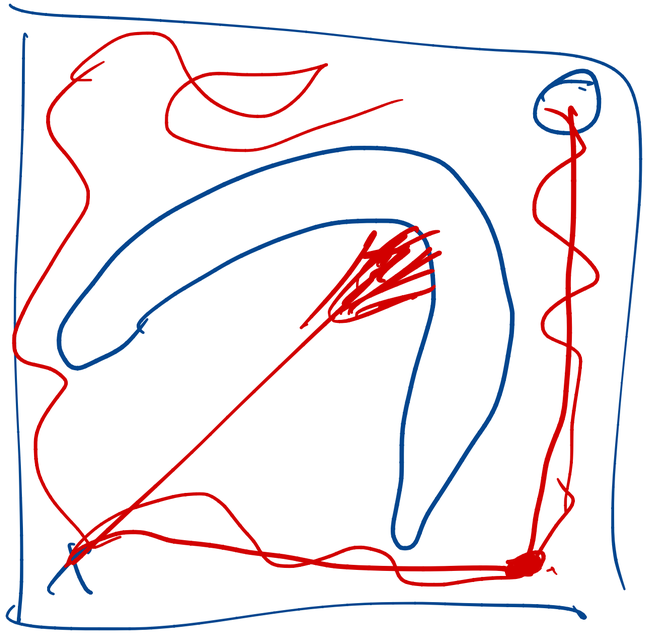
- Robbins-Monro conditions:
  - $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ → learning rate decreases over time
  - $\sum_{t=0}^{\infty} \alpha_t = \infty$ → but slowly

$\alpha_t = \frac{1}{t}$ $\epsilon_t = \frac{1}{\sqrt{t}}$

Exploration vs Exploitation

- Estimate the structure (size/topology) of the state space
- “Finding the right information” —> Difficult





Exploration

- Sometimes take random actions

ε-greedy

$P_{explore} \rightarrow$ Explore

$1 - P_{explore} \rightarrow$ Act on the Policy

Exploration Function

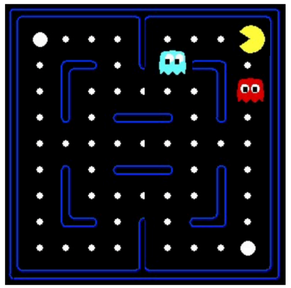
- Takes a value estimate u and a visit count n, and returns an optimistic utility

  - Q-update
  - Modified Q-Update

Basic Q-learning

- Keep table of all q-values

- Generalize



Vector of features

- Features are functions from states to real numbers (often 0/1) that capture important properties of the state

Q-learning with linear Q-functions

$$Q(s,a) \leftarrow Q(s,a) + \alpha(TD)$$

$$w_i \leftarrow w_i + \alpha(TD) f_i(s,a)$$

Deep Q Learning

DQN, Mnih et al., 2013