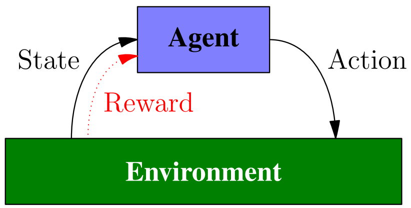
Reinforcement Learning

Reinforcement Learning

Learning of a behavior without
explicit information about correct actions

- A reward gives information about success

Agent–Environment Abstraction:



- Policy — Choice of action, depending on current state
- Learning Objective:
  Develop a policy which maximizes the reward
  ... total reward over the agents lifetime!

Credit Assignment Problems

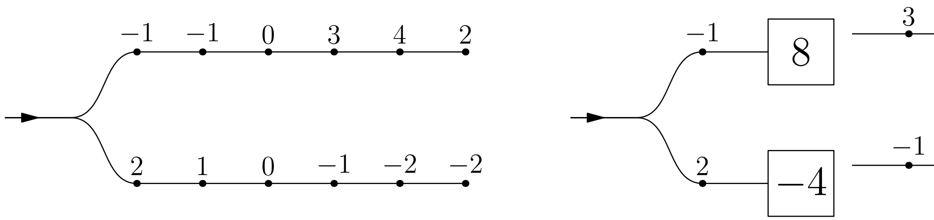
- The reward does not necessarily arrive *when* you do something good
  Temporal credit assignment
- The reward does not say *what* was good
  Structural credit assignment

Consider a *minimal* behavior selection situation:
What is the best action to make?

- Reward — Immediate consequence of our decision
- Planning — Taking future reward into account
- Optimal Behavior — Maximize *total* reward

Can the agent make the right decision without explicitly modeling the future?

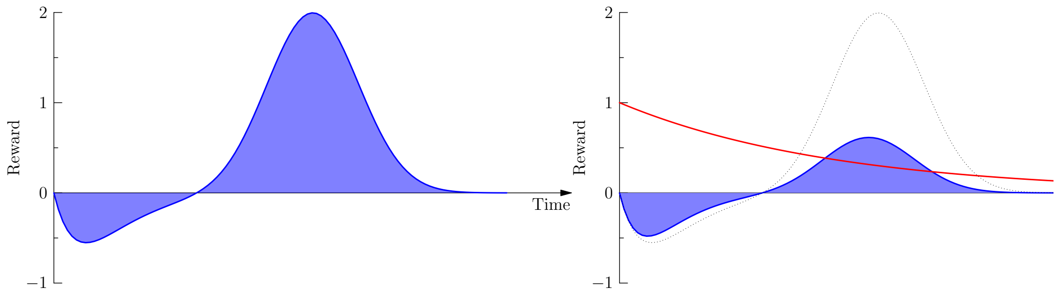
Yes! If the Sum of Future Rewards for each situation is known.



Call this the State Value

- State Values are *subjective*
- Depend on the agents own behavior
- Must be re-adjusted as the agents behavior improves

Planning Horizon — How long is *the future*?



Infinite future allows for *infinite postponing*

- Discounting — Make *early reward* more valuable
- Discount factor (γ) — Time scale of planning

- Finite Horizon

$$\max \left[\sum_{t=0}^{h} r_t \right]$$

- Infinite Horizon

$$\max \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]$$

Requires discount of future reward ($0 < \gamma < 1$)

The Reward function controls which task should be solved

- Game (Chess, Backgammon)
  Reward only at the end: $+1$ when winning, -1 when loosing
- Avoiding mistakes (cycling, balancing, ...)
  Reward -1 at the end (when failing)
- Find a short/fast/cheap path to a goal
  Reward -1 at each step

Simplifying Assumptions

- Discrete time
- Finite number of actions a_i

$$a_i \in a_1, a_2, a_3, \ldots, a_n$$

- Finite number of states s_i

$$s_i \in s_1, s_2, s_3, \ldots, s_m$$

- Environment is a stationary *Markov Decision Process*
  Reward and next state depends only on s, a and chance

Classical model problem: *Grid World*

- Each state is represented by a position in a grid
- The agent acts by moving to other positions

G			
			G

Trivial labyrinth

Reward: -1 at each step until a goal state (G) is reached

The values of a state depends on the current policy.

0	−1	−2	−3
−1	−2	−3	−2
−2	−3	−2	−1
−3	−2	−1	0

V with an optimal policy

0	−14	−20	−22
−14	−18	−22	−20
−20	−22	−18	−14
−22	−20	−14	0

V with a random policy

The Agents Internal Representation

- *Policy*
  The action chosen by the agent for each state

$$\pi(s) \mapsto a$$

- *Value Function*
  Expected total future reward from s when following policy π

$$V^{\pi}(s) \mapsto \Re$$

Representation of the Environment

- Where does an action take us?

$$\delta(s, a) \mapsto s'$$

- How much reward do we receive?

$$r(s, a) \mapsto \Re$$

The values of different states are interrelated
Bellman's Equation:

$$V^{\pi}(s) = r(s, \pi(s)) + \gamma \cdot V^{\pi}(\delta(s, \pi(s)))$$

Normal scenario: $r(s,a)$ and $\delta(s,a)$ are not known

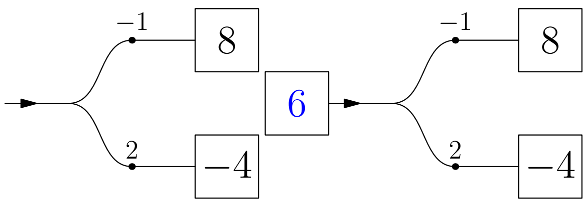
V^{π} must be estimated by experience

Monte-Carlo Method

- Start at a random s
- Follow π, store the rewards and s_t
- When the goal is reached, update $V^{\pi}(s)$-estimation for all visited stated with the future reward we actually received

Painfully slow!

Temporal Difference Learning



Temporal Difference — the difference between:

- Experienced value
  (immediate reward + value of next state)
- Expected value

Measure of unexpected success

- Two sources:
  - Higher *immediate reward* than expected
  - Reached *better situation* than expected

- Expected Value

$$V^{\pi}(s_t)$$

- One-step Experienced Value

$$r_{t+1} + \gamma \cdot V^{\pi}(s_{t+1})$$

- TD-signal — measure of surprise / disappointment

$$TD = r_{t+1} + \gamma \cdot V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$$

- TD Learning

$$V^{\pi}(s_t) \leftarrow V^{\pi}(s_t) + \eta TD$$

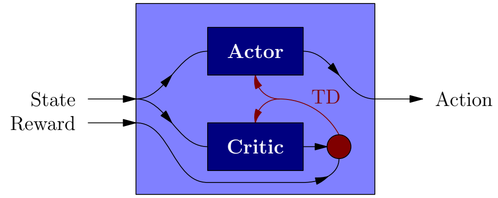
Learning to Act

How do we get a *policy* from the *TD signal*?

Many possibilities...

- Actor–Critic Model
  (Barto, Sutton, Anderson *IEEE Trans. Syst. Man & Cybern.* 1983)
- Q-Learning
  (Watkins, 1989; Watkins & Dayan *Machine Learning* 1992)

Actor–Critic Model



Actor — Associates *states* with *actions* $\pi(\cdot)$
Critic — Associates *states* with their *value* $V(\cdot)$

TD-signal is used as a *reinforcement signal* to update both!

- High TD (unexpected success)
  - Increase value of preceeding state
  - Increase tendency to make same action again
- Low TD (unexpected failure)
  - Decrease value of preceeding state
  - Decrease tendency to make same action again

Q-Learning
Estimate $Q(s,a)$ instead of $V(s)$

$Q(s,a)$: Expected total reward when doing a from s.

$$\pi(s) = \operatorname*{argmax}_a Q(s,a)$$

$$V^\star(s) = \max_a Q^\star(s,a)$$

The Q-function can also be learned using Temporal-Difference

$$Q(s,a) \leftarrow Q(s,a) + \eta \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

s' is the next state.

What do we do when...

- The environment is not deterministic
- The environment is not fully observable
- There are way too many states
- The states are not discrete
- The agent is acting in continuous time

The Exploration–Exploitation dilemma

If an agent strictly follows a greedy policy based on the current estimate of Q, learning is not guaranteed to converge to $Q^\star$

Simple solution:
Use a policy which has a certain probability of "making mistakes"

- ϵ-greedy
  Sometimes (with probability ϵ) make a random action instead of the one that seems best (greedy)
- Softmax
  Assign a probability to choose each action depending on how good they seem

Accelerated learning

Idéa: TD updates can be used to improve not only the last state's value, but also states we have visited earlier.

$$\forall s, a : Q(s, a) \leftarrow Q(s, a) + \eta \left[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right] \cdot e$$

e is a remaining trace (eligibility trace) encoding how long ago we were in s doing a.

Often denoted TD(λ) where λ is the time constant of the trace e