Lecture 7: Reinforcement Learning

Ziyu Shao

School of Information Science and Technology
ShanghaiTech University

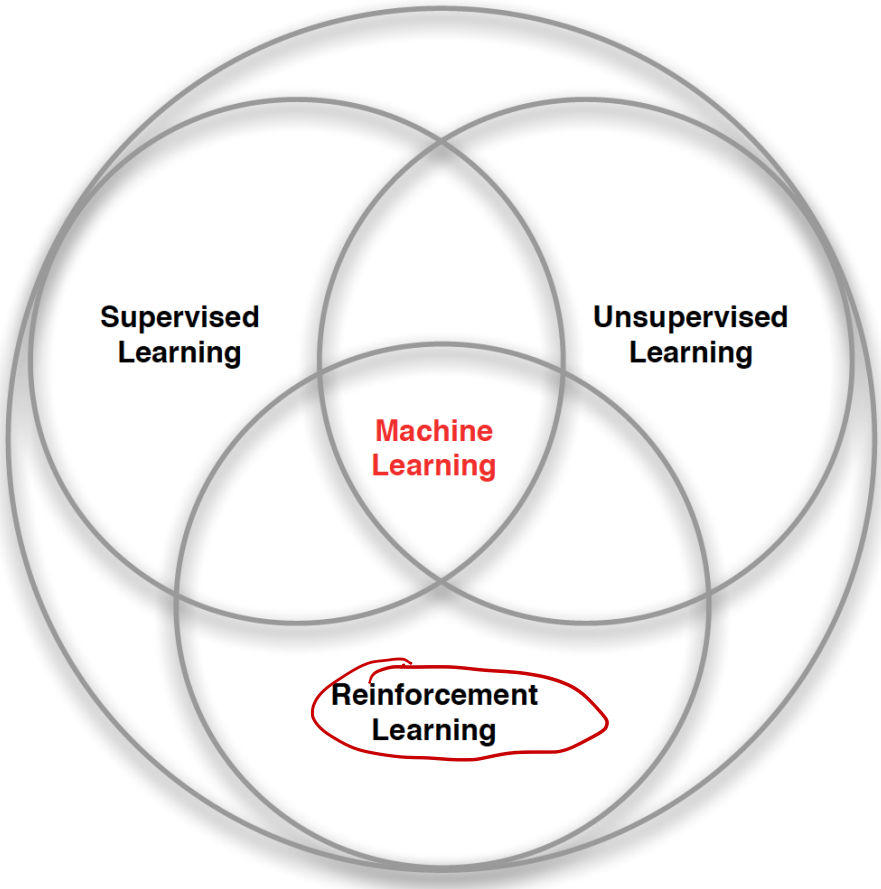
April 20 & 22, 2020

Outline

Outline

What is Reinforcement Learning?

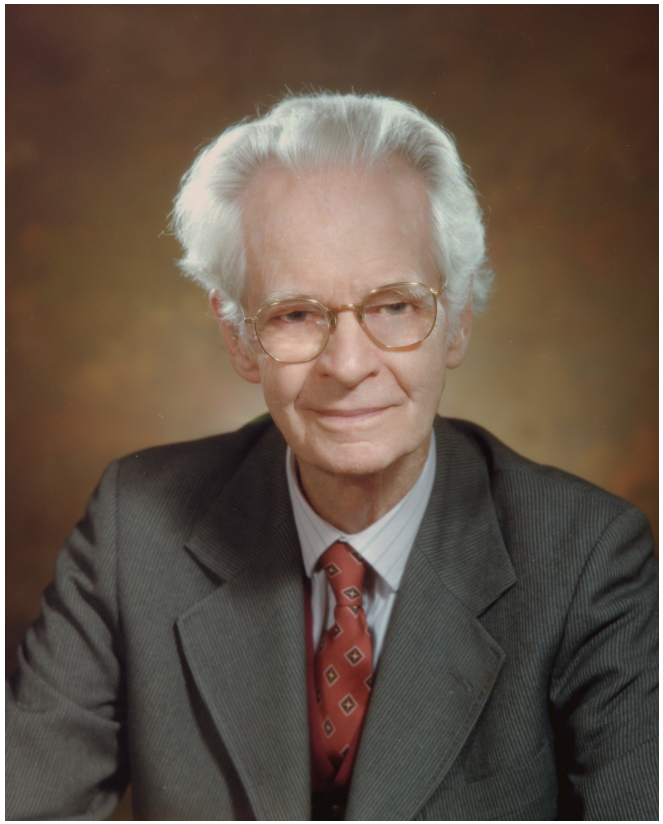
- **Wikipedia**: reinforcement learning is an area of machine learning inspired by **behavioral psychology**, concerned with how software **agents** ought to take **actions** in an environment so as to maximize some notion of cumulative **reward**.



Behavioral Psychology

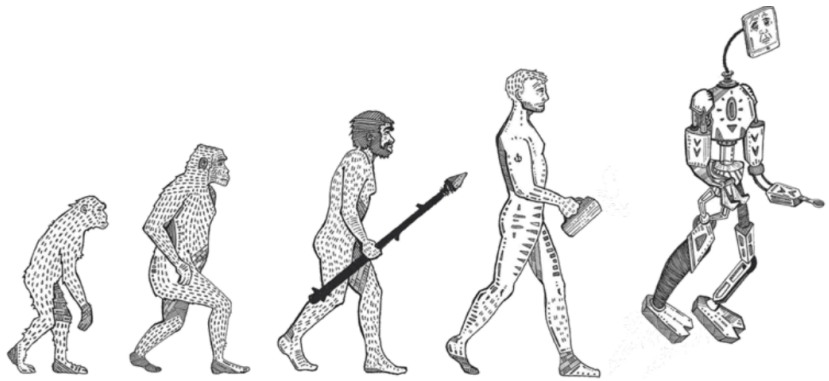
- Behavior is primarily shaped by **reinforcement** rather than free-will
  - behaviors that result in praise/pleasure tend to repeat
  - behaviors that result in punishment/pain tend to become extinct

B.F. Skinner



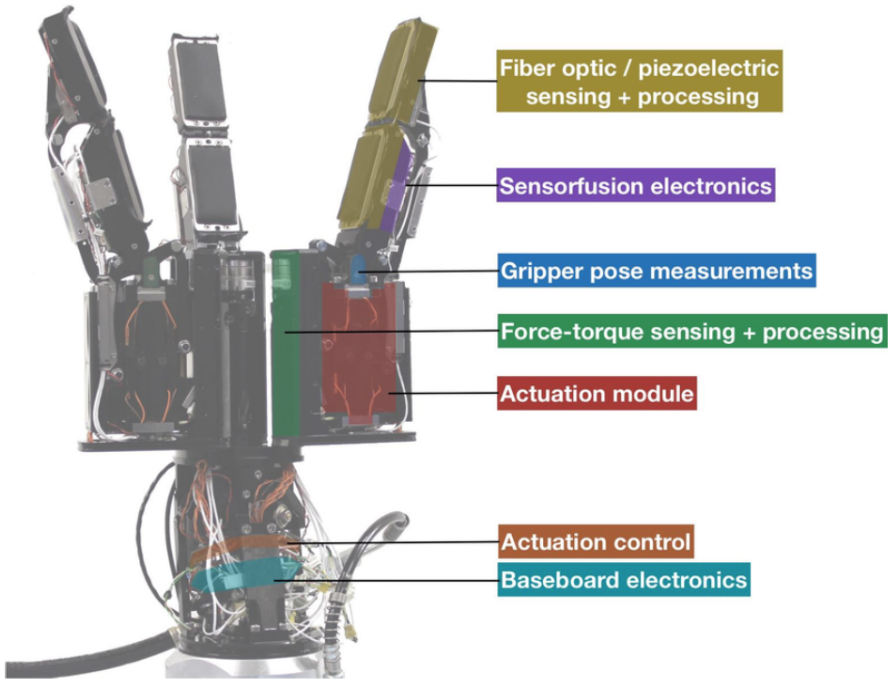
Agent

- An entity (learner & decision maker) that is equipped with
  - sensors, in order to sense the environment
  - end-effectors in order to act in the environment
  - goals that she wants to achieve



Action

- Used by the agent to interact with the environment.
- May have many different temporal granularities and abstractions



Actions can be

- The instantaneous torques applied on the gripper
- The instantaneous gripper translation, rotation, opening
- Instantaneous forces applied to the objects
- Short sequences of the above

Reward: Important Concept

- A **reward** R_t is a scalar feedback signal
- Indicates how well agent is doing at step t
- The agent's job is to maximize cumulative reward

Reinforcement learning is based on the reward hypothesis:

Definition

All goals can be described by the maximization of expected cumulative reward

Von Neumann - Morgenstern expected utility theorem

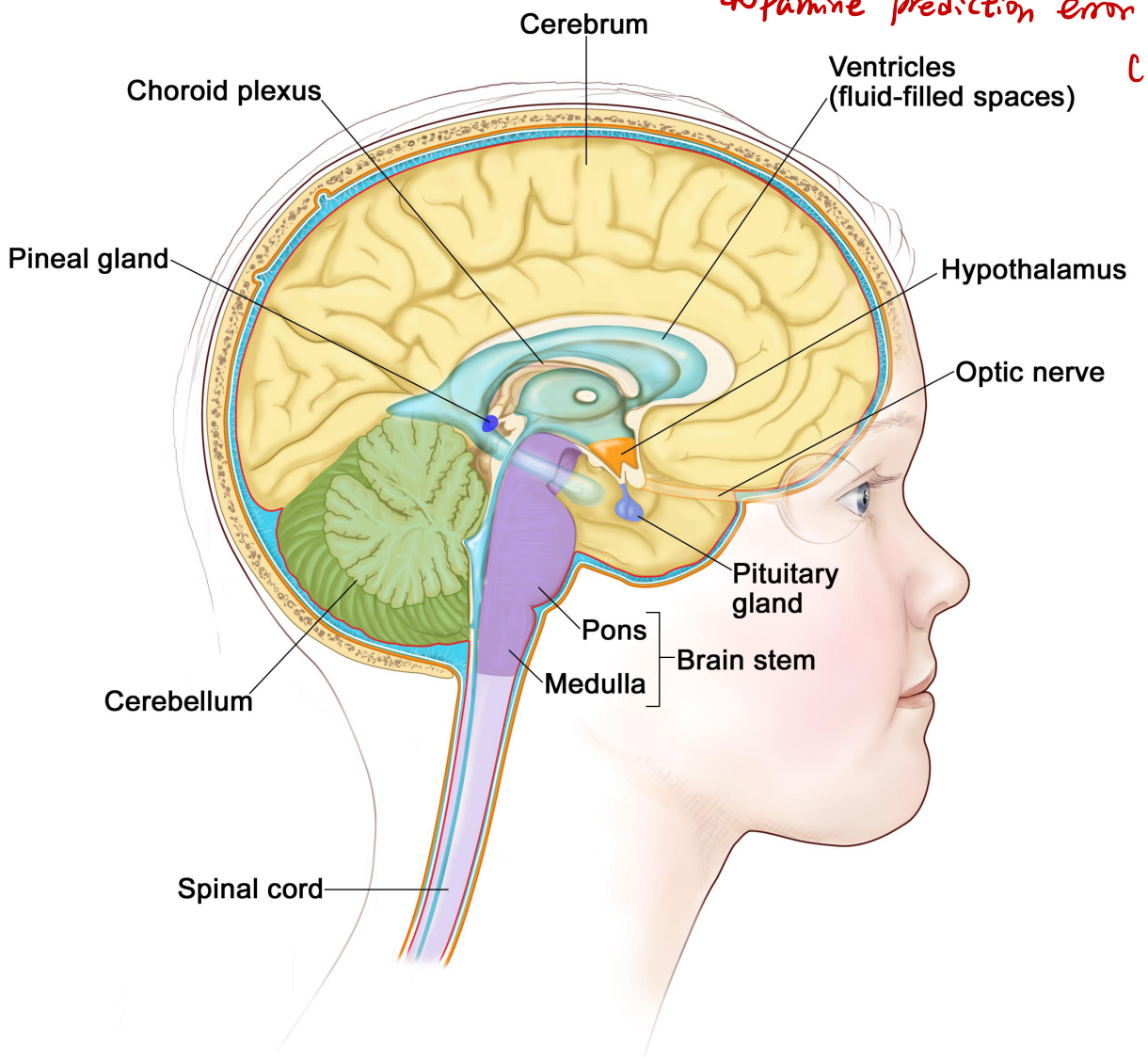
Reward: Brain Perspective

Example: dopamine (role: neurotransmitter)

↓

dopamine prediction error hypothesis

(reward surprise)



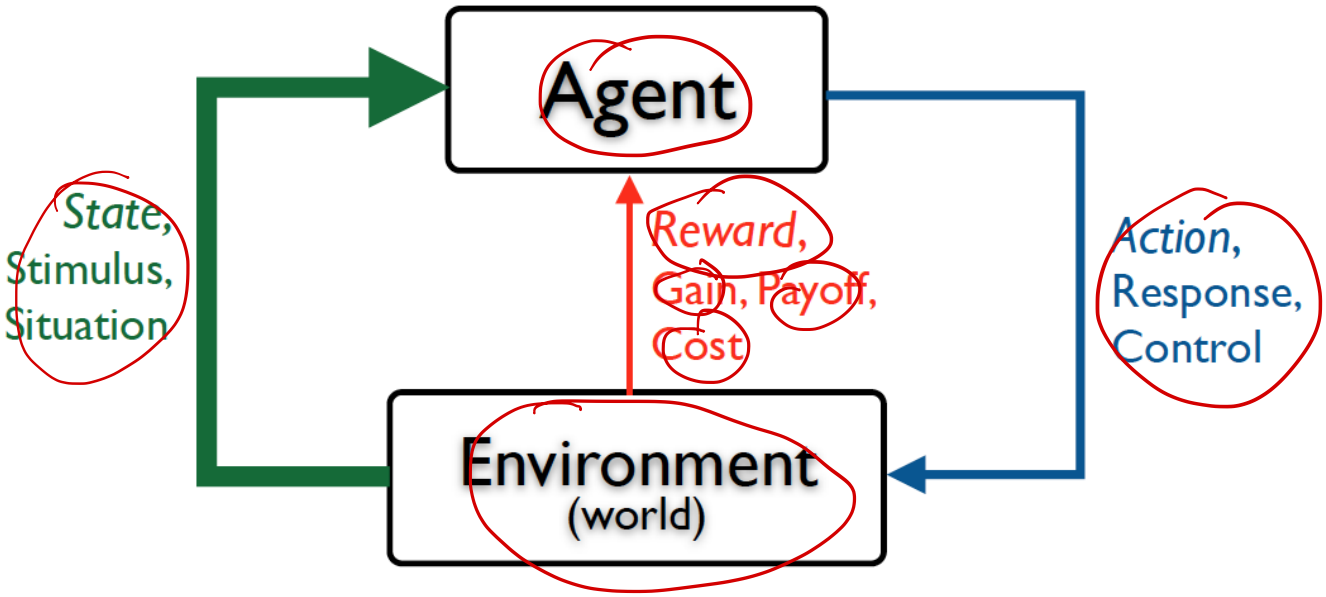
Reward: Animal Psychology

- Negative reinforcements: pain & hunger
- Positive reinforcements: pleasure & food
- Reinforcements used to train animals

Examples of Reward in Engineering

- Fly stunt manoeuvres in a helicopter
  - +ve reward for following desired trajectory
  - −ve reward for crashing
- Defeat the world champion at Backgammon
  - +ve reward for winning a game
  - −ve reward for losing a game
- Control a power station
  - +ve reward for producing power
  - −ve reward for exceeding safety thresholds
- Make a humanoid robot walk
  - +ve reward for forward motion
  - −ve reward for falling over

Inside Reinforcement Learning



- Environment may be unknown, nonlinear, stochastic and complex
- Agent learns a mapping from states to actions: seeking to maximize its cumulative reward in the long run

Inside Reinforcement Learning?

- Agent-oriented learning:
  - explicitly consider the whole problem of goal-directed agent interacting with an uncertain environment
  - more realistic and ambitious than other kinds of machine learning
- Learning by trial and error, with only delayed evaluative feedback (reward)
  - the kind of machine learning most like nature learning
  - learning that can tell for itself when it is right or wrong
- Trade-off between exploration & exploitation

Other Elements of Reinforcement Learning

- **Environment**: the thing agent interacts with, comprising everything outside the agent
- **State**: captures whatever information is available to the agent about its environment
- **Policy**: a mapping from perceived states of the environment to actions to be taken when in those states
- **Value function**: the value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state
- **Efficiently estimating values**: core of almost all reinforcement learning algorithms (Immediate reward vs. long-term value)
- **A model of the environment** (optional)

Reinforcement Learning: Examples

- Game playing (go, atari, backgammon)
- Operations research (pricing, vehicle routing)
- Elevator scheduling
- Helicopter control
- Spoken dialog systems
- Data center energy optimization
- Self-managing network systems
- Autonomous vehicles
- Computational finance

Example: Vehicle Routing

- Agent: vehicle routing software
- Environment: stochastic demand
- State: vehicle location, capacity & depot requests
- Action: vehicle route
- Reward: $-$travel costs




Example: Helicopter Control

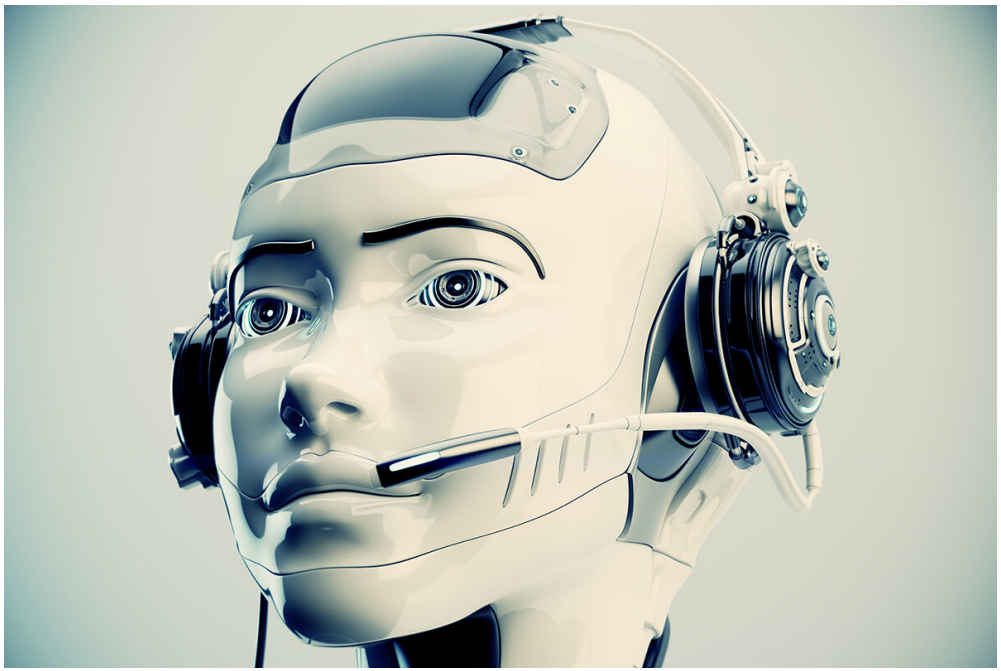
- Agent: controller
- Environment: helicopter
- State: position, orientation, velocity & angular velocity
- Action: collective pitch, cyclic pitch & tail rotor control
- Reward: $-$deviation from desired trajectory
- 2008(Andrew Ng): automated helicopter wins acrobatic competition against humans

Example: Go

- Agent: player
- Environment: opponent
- State: board configuration
- Action: next stone location
- Reward: $+1$ win & -1 loose
- 2016: AlphaGo defeats top player lee Sedol (4-1)
  - Game 2 move 37: AlphaGo plays an unexpected move

Example: Conversational Agent

- Agent: virtual asistant Siri (Artificial Intelligence)
- Environment: user
- State: conversation history
- Action: next utterance
- Reward: points based on task completion & user satisfaction



Main Topics of Reinforcement Learning

- Learning: by trial and error
- Planning: search, reason, thought, cognition
- Prediction: evaluation functions, knowledge
- Control: action selection, decision making
- Dynamics: how the state changes given the actions of the agent
- Model-based RL
  - dynamics are known or are estimated
  - solving RL problems that use models and planning
- Model-free RL
  - unknown dynamics
  - explicitly trial-and-error learners (Sample inefficiency)

Characteristics of Reinforcement Learning

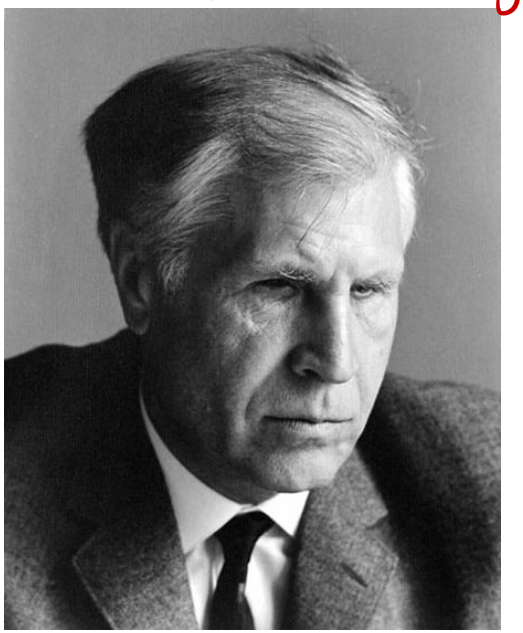
What makes reinforcement learning different from other machine learning paradigms?

- There is no supervisor, only a reward signal
- Feedback is delayed, not instantaneous
- Time really matters (sequential, non i.i.d data)
- Agent's actions affect the subsequent data it receives

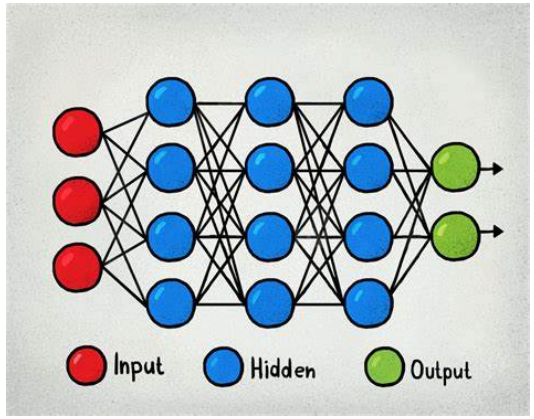
Reinforcement Learning: Interdisciplinary Topic

- Reinforcement learning is also known as
  - Approximate optimal control
  - Approximate dynamic programming ADP
  - Neuro-dynamic programming NDP

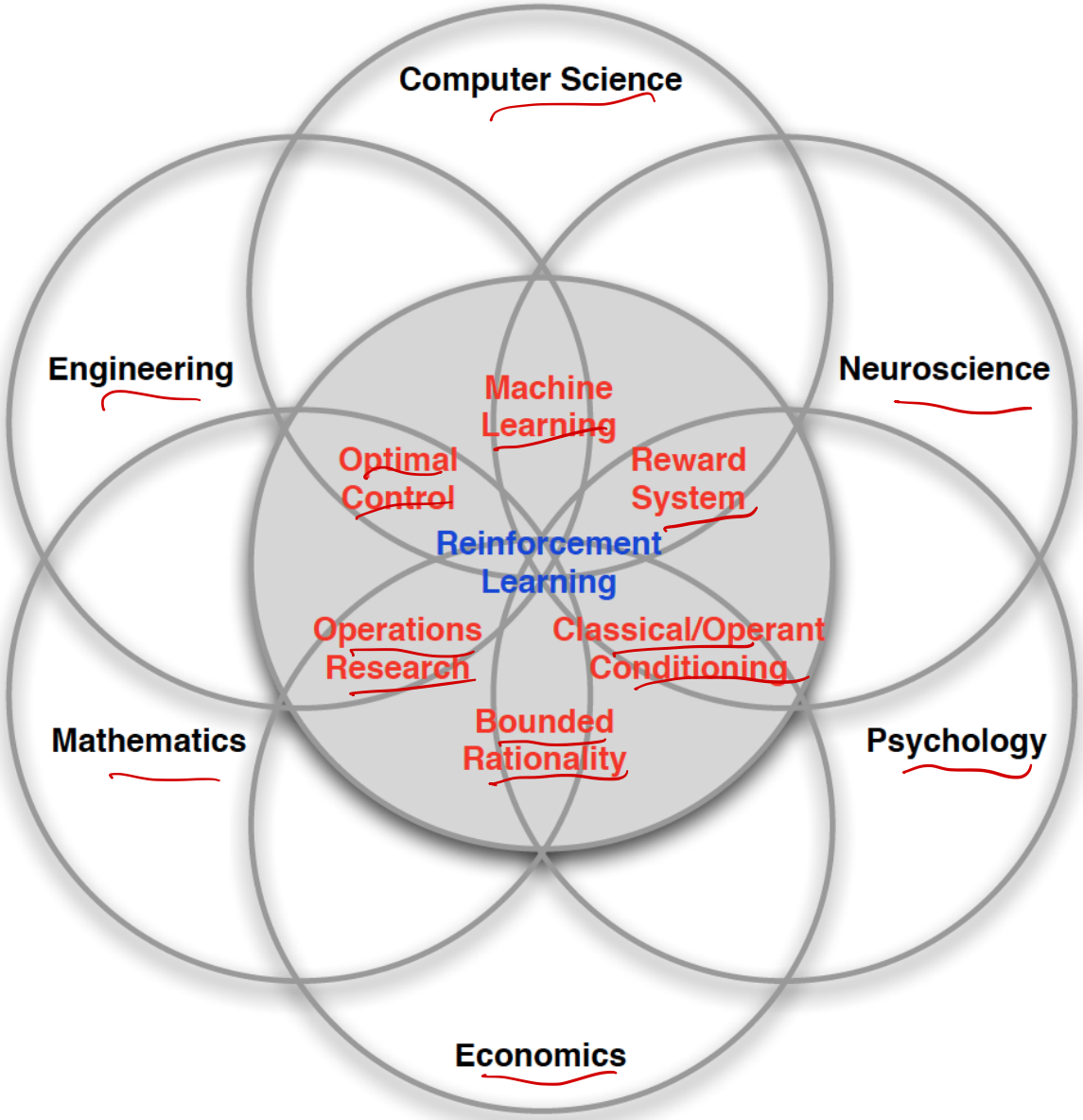
Lev Semenovich Pontriagin



Richard E. Bellman

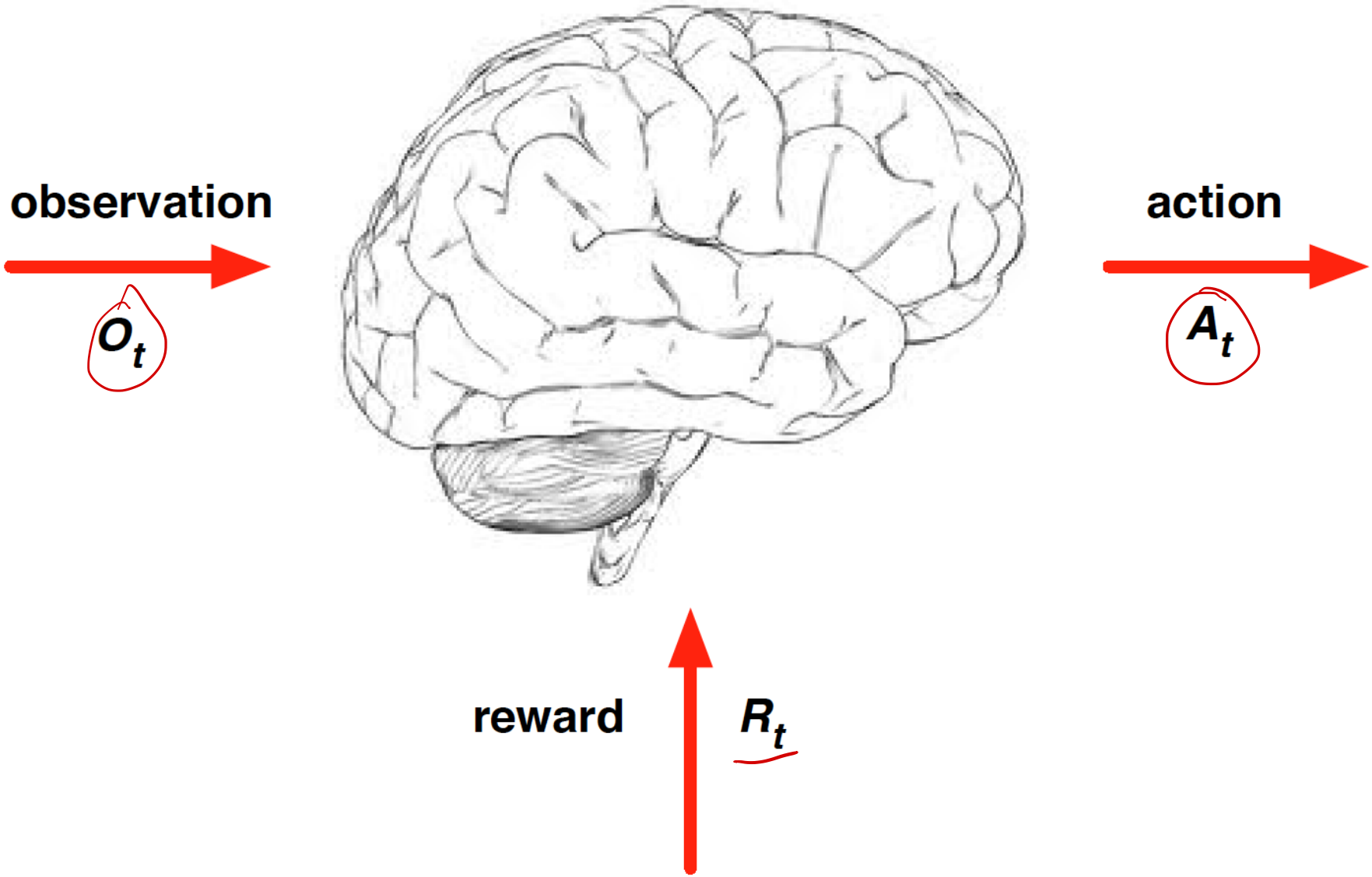


Reinforcement Learning: Interdisciplinary Topic



Outline

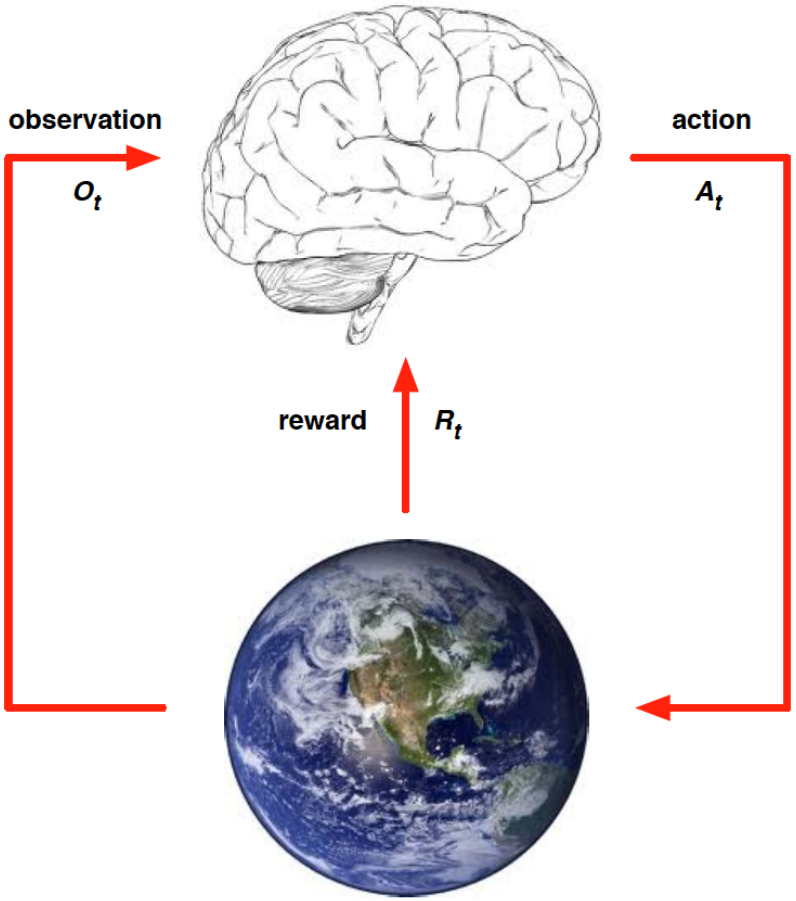
Agent and Environment



Agent and Environment

1°. what if the reward is a vector?

2°. inverse reinforcement learning (IRL)?
: learn an agent's objectives, values or rewards by observing its behavior
(reverse-engineering)

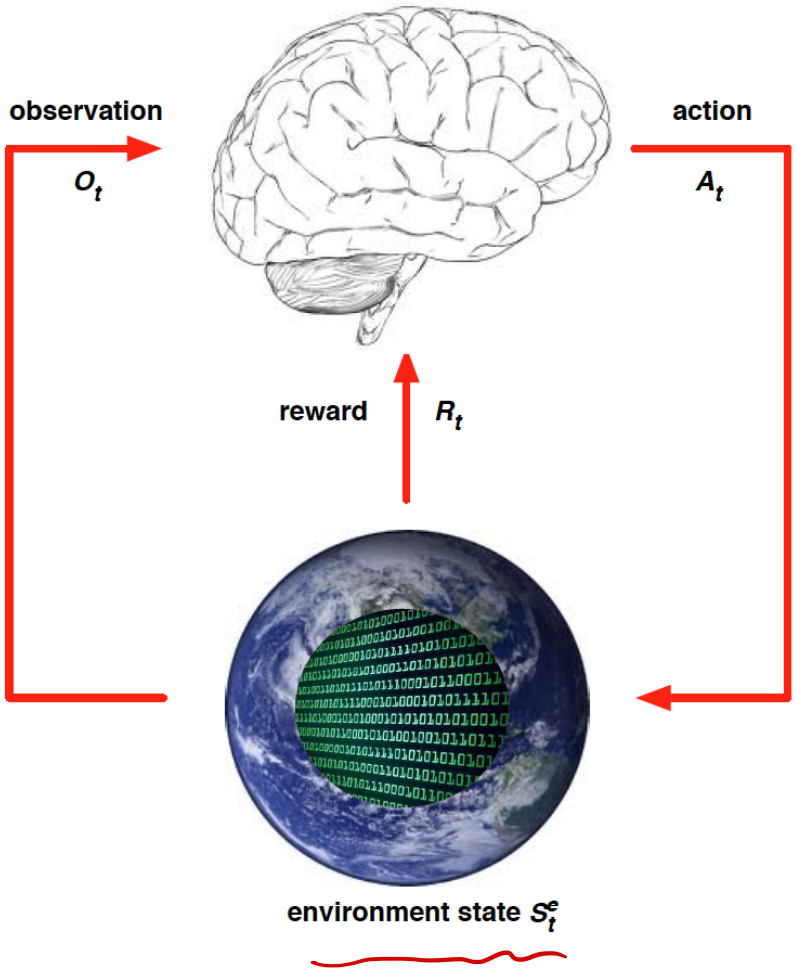


- At each step t the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step

History and State

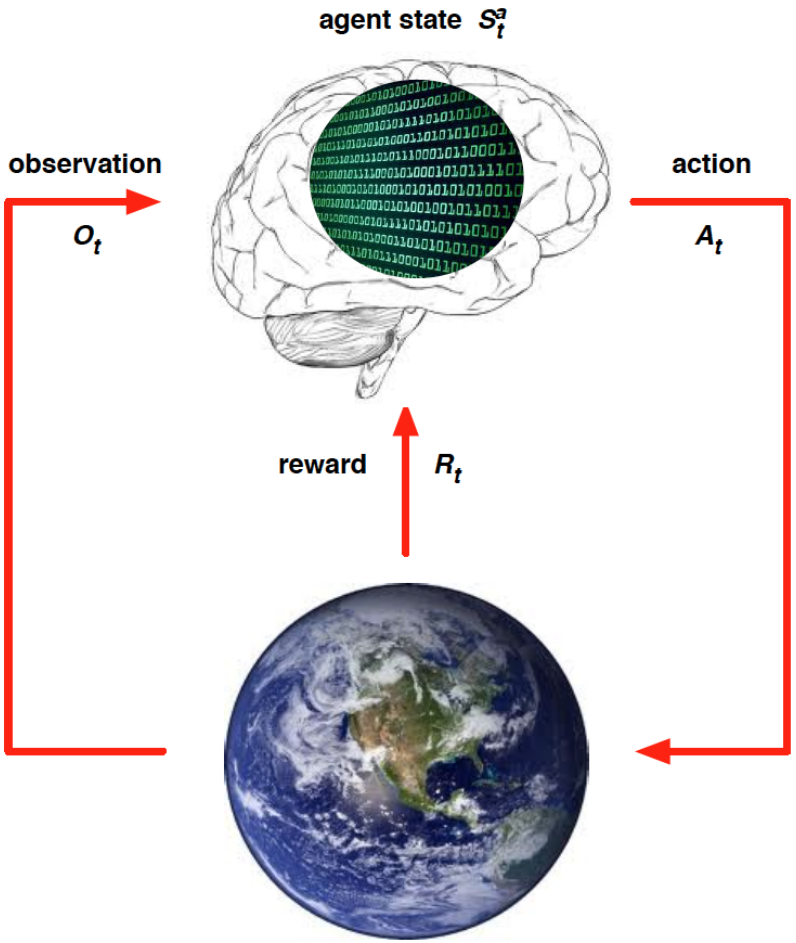
- The history is the sequence of observations, actions, rewards $H_t = O_1, R_1, A_1, \ldots, A_{t-1}, O_t, R_t$
- i.e. all observable variables up to time t
- i.e. the sensorimotor stream of a robot or embodied agent
- What happens next depends on the history:
  - The agent selects actions
  - The environment selects observations/rewards
- State is the information used to determine what happens next
- Formally, state is a function of the history: $S_t = f(H_t)$

Environment State



- The environment state S_t^e is the environment's private representation
- i.e. whatever data the environment uses to pick the next observation/reward
- The environment state is not usually visible to the agent
- Even if S_t^e is visible, it may contain irrelevant information

Agent State



- The agent state S_t^a is the agent's internal representation
- i.e. whatever information the agent uses to pick the next action
- i.e. it is the information used by reinforcement learning algorithms
- It can be any function of history:

$$S_t^a = f(H_t)$$

Information State

An information state (a.k.a. Markov state) contains all useful information from the history.

Definition

A state S_t is Markov if and only if

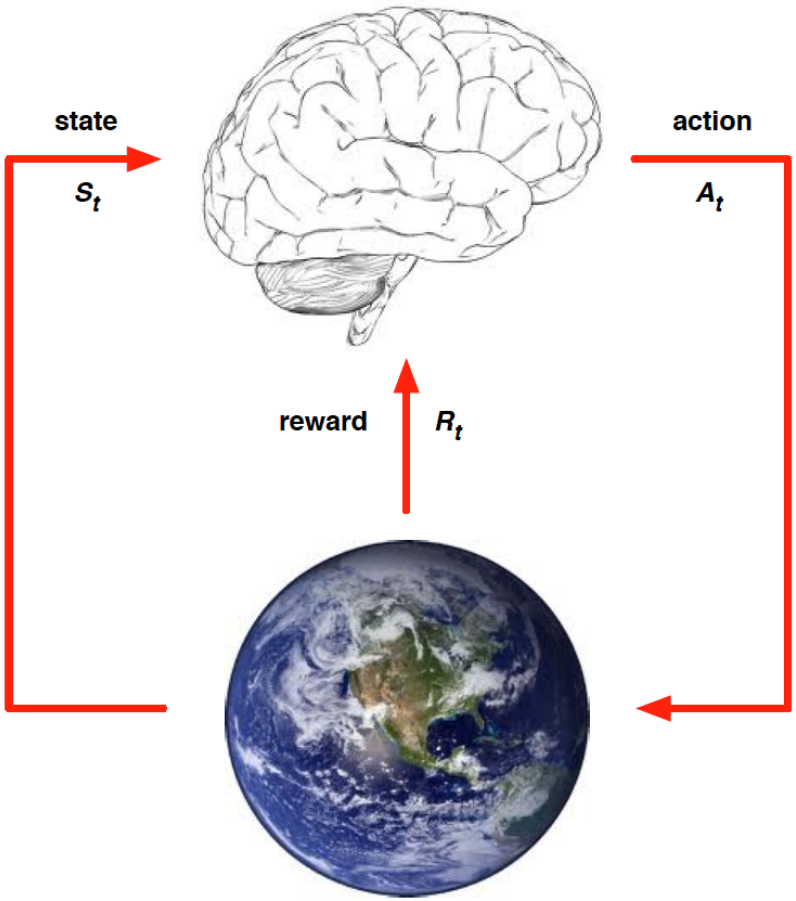
$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, ..., S_t]$$

- "The future is independent of the past given the present"

$$H_{1:t} \to S_t \to H_{t+1:\infty}$$

- Once the state is known, the history may be thrown away
- i.e. The state is a sufficient statistic of the future
- The environment state S_t^e is Markov
- The history H_t is Markov

Fully Observable Environments



Full observability: agent directly observes environment state

$$O_t = S_t^a = S_t^e$$

- Agent state = environment state = information state
- Formally, this is a Markov decision process (MDP)
- (Next lecture and the majority of this course)

Partially Observable Environments

- Partial observability: agent indirectly observes environment:
  - A robot with camera vision isn't told its absolute location
  - A trading agent only observes current prices
  - A poker playing agent only observes public cards
- Now agent state $\neq$ environment state
- Formally this is a partially observable Markov decision process (POMDP)
- Agent must construct its own state representation S_t^a, e.g.
  - Complete history: $S_t^a = H_t$
  - Beliefs of environment state: $S_t^a = (\mathbb{P}[S_t^e = s^1], ..., \mathbb{P}[S_t^e = s^n])$
  - Recurrent neural network: $S_t^a = \sigma(S_{t-1}^a W_s + O_t W_o)$

Can tools from RNN

Major Components of An RL Agent

An RL agent may include one or more of these components:

- Policy: agent's behavior function
- Value function: how good is each state and/or action
- Model: agent's representation of the environment

Policy

- A policy is the agent's behavior
- It is a map from state to action, e.g.
- Deterministic policy: $a = \pi(s)$
- Stochastic policy: $\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]$

$\sum_{a \in \mathcal{A}} \pi(a|s) = 1$

Value Function

- Value function is a prediction of future reward
- Used to evaluate the goodness/badness of states
- And therefore to select between actions, e.g.

$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | S_t = s]$$

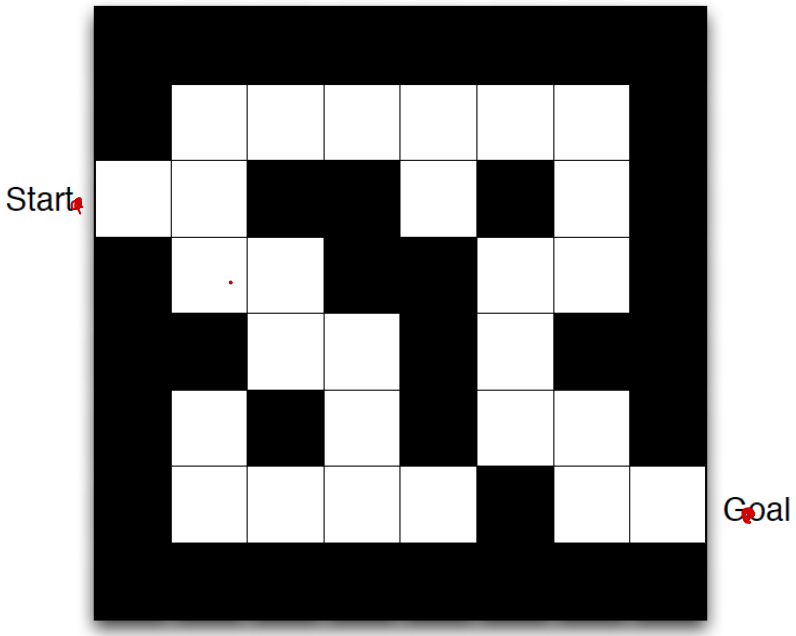
Value of state s following policy π.

Model

- A model predicts what the environment will do next
- $\mathcal{P}$ predicts the next state
- $\mathcal{R}$ predicts the next (immediate) reward, e.g.

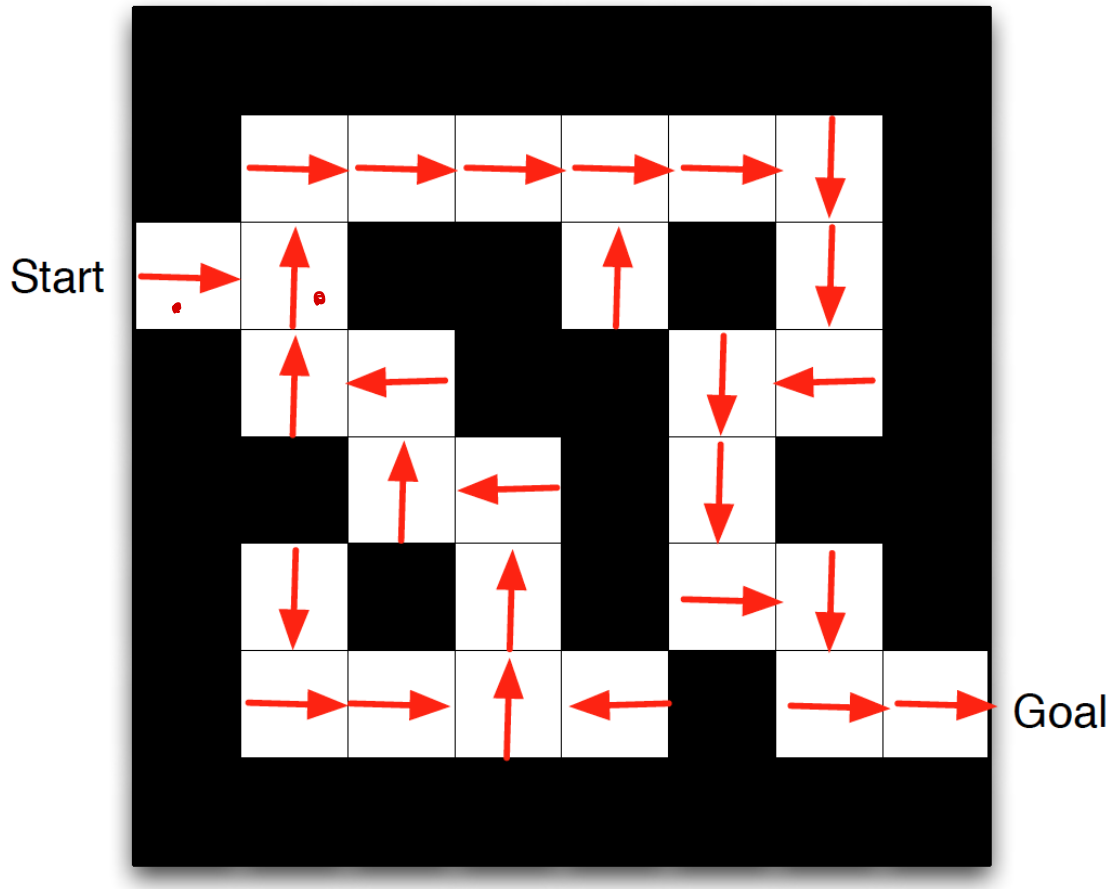
$$\mathcal{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$
$$\mathcal{R}^a_s = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$$

Maze Example



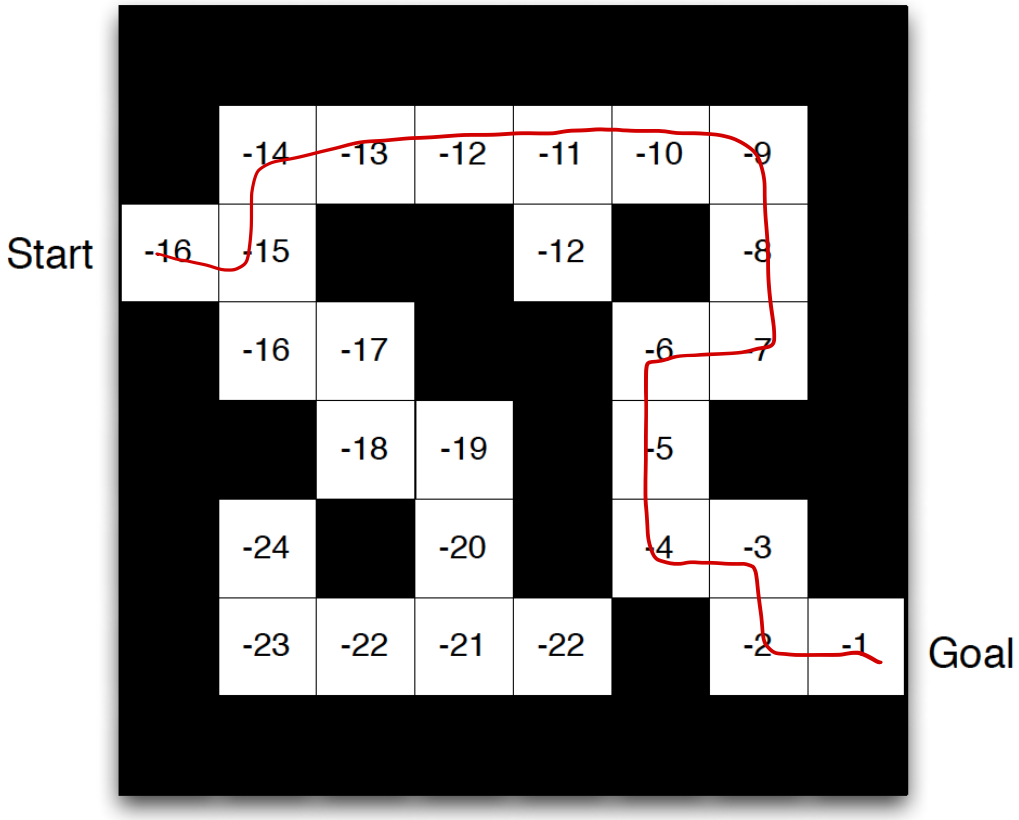
- Rewards: -1 per time-step
- Actions: N, E, S, W
- States: Agent's location

Maze Example: Policy



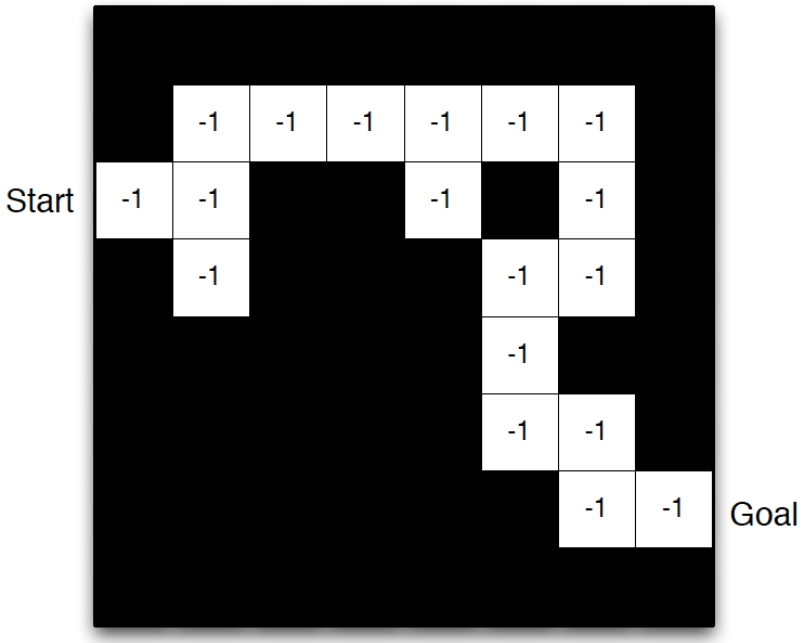
- Arrows represent policy $\pi(s)$ for each state s

Maze Example: Value Function



- Numbers represent value $v_\pi(s)$ of each state s

Maze Example: Model



- Agent may have an internal model of the environment
- Dynamics: how actions change the state
- Rewards: how much reward from each state
- The model may be imperfect

- Grid layout represents transition model $\mathcal{P}^a_{ss'}$
- Numbers represent immediate reward $\mathcal{R}^a_s$ from each state s (same for all a)

Categorizing RL Agents: I

RL Algorithms

- Value based
  - No Policy (Implicit)
  - Value Function
- Policy Based
  - Policy
  - No Value Function
- Actor-Critic
  - Policy (actor)
  - Value Function (critic)

Q-learning

state-action value function

policy-gradient

$\pi(a|s)$

Actor-critic

Parallel

Optimization

primal

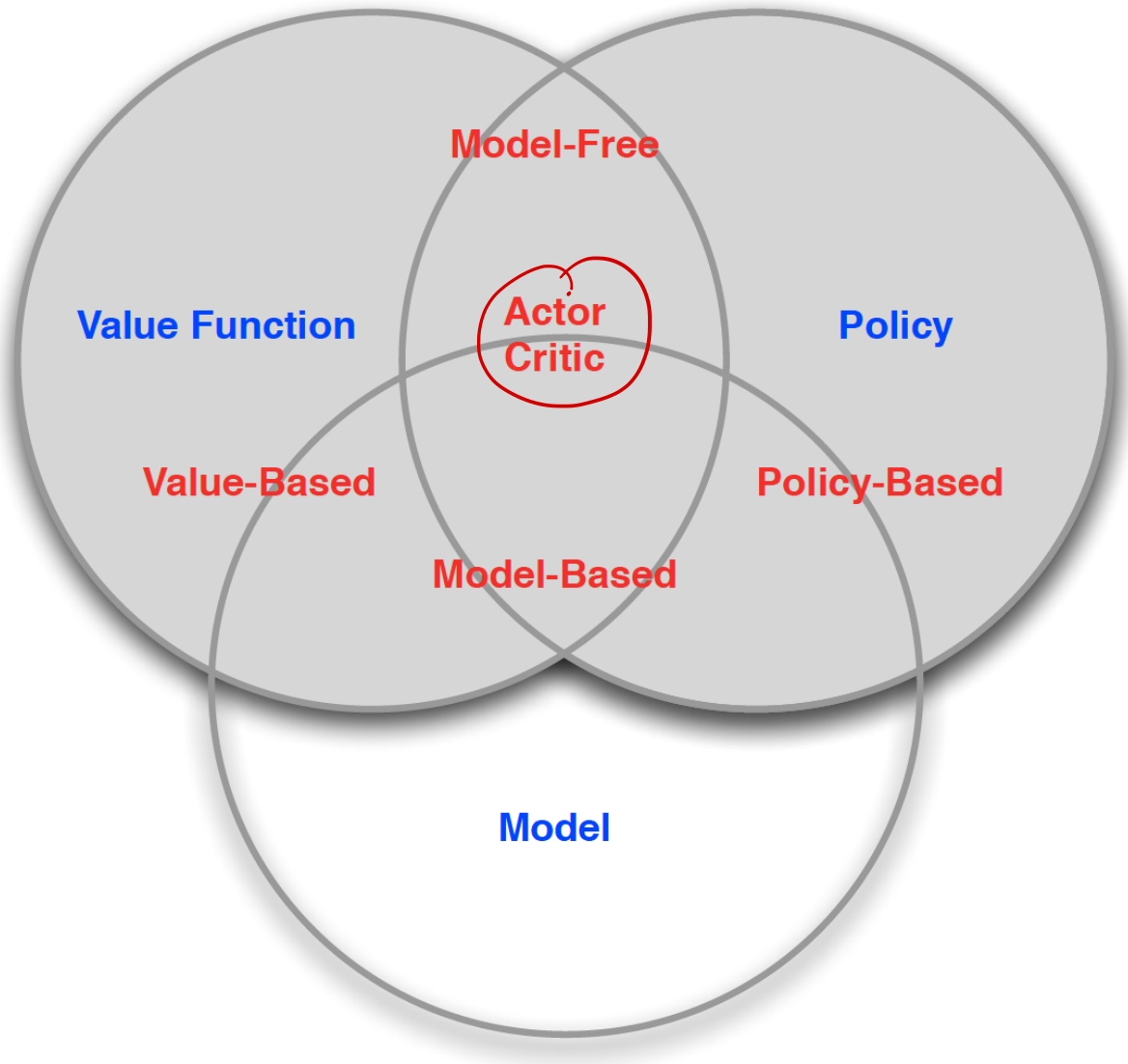
dual

primal-dual.

Categorizing RL Agents: II

- Model Free
  - Policy and/or Value Function
  - No Model
- Model Based
  - Policy and/or Value Function
  - Model

RL Agent Taxonomy



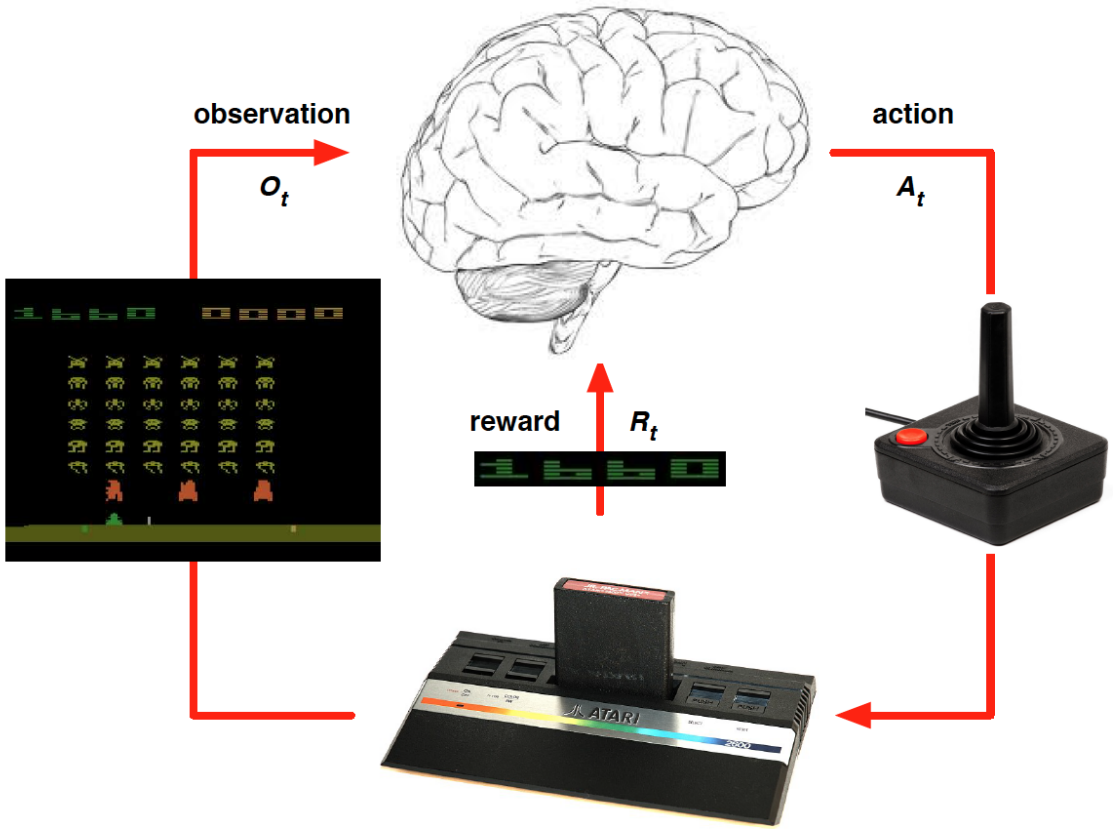
Learning and Planning

Two fundamental problems in sequential decision making

- Reinforcement Learning
  - The environment is initially unknown
  - The agent interacts with the environment
  - The agent improves its policy
- Planning
  - A model of the environment is known
  - The agent performs computations with its model (without any external interaction)
  - The agent improves its policy
  - a.k.a. deliberation, reasoning, introspection, pondering, thought, search

Atari Example: Reinforcement Learning

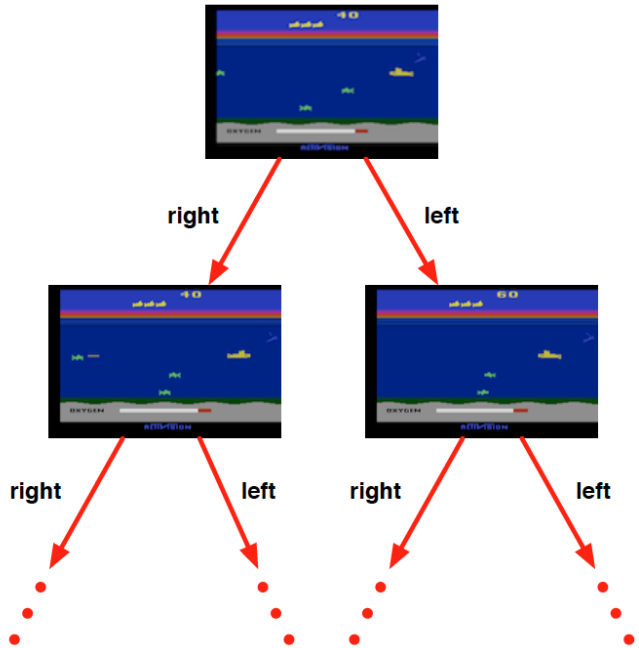
game company.



- Rules of the game are unknown
- Learn directly from interactive game-play
- Pick actions on joystick, see pixels and scores

Atari Example: Planning

- Rules of the game are known
- Can query emulator
  - perfect model inside agent's brain
- If I take action a from state s:
  - what would the next state be?
  - what would the score be?
- Plan ahead to find optimal policy
  - e.g. tree search



Prediction and Control

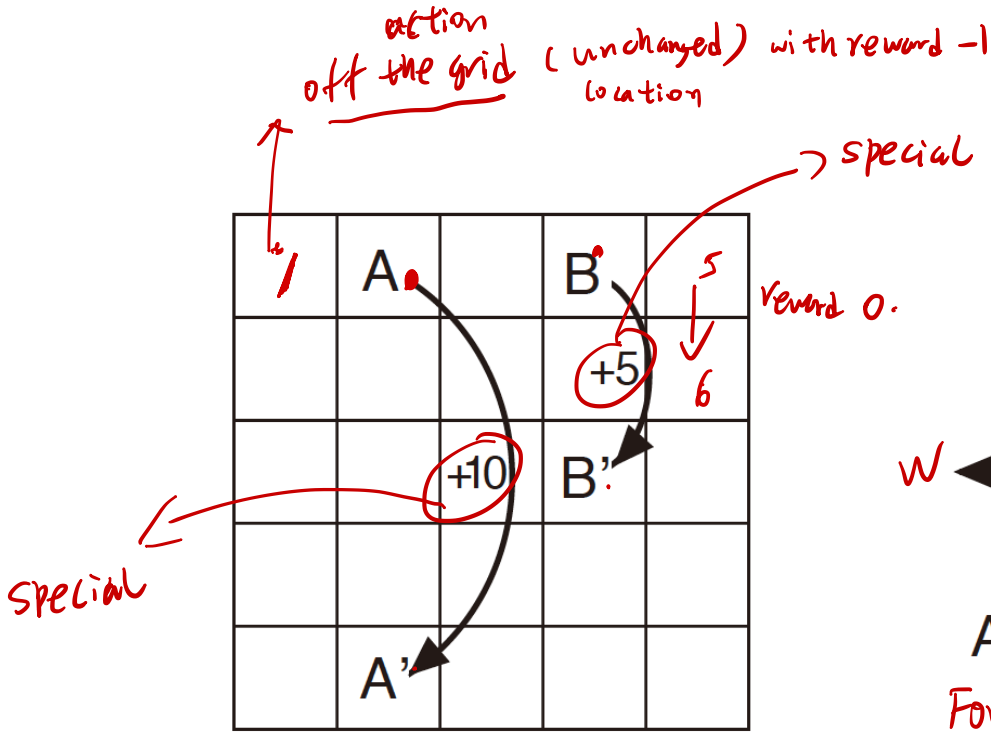
- Prediction: evaluate the future
  - Given a policy $\pi(a|s)$.
- Control: optimize the future
  - Find the best policy

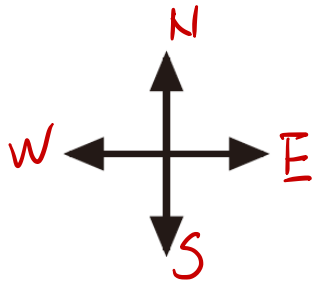
$v_\pi(s)$

$q_\pi(s,a)$

$\pi^* = \arg\max_\pi v_\pi(s)$

Gridworld Example: Prediction





Actions

Four actions

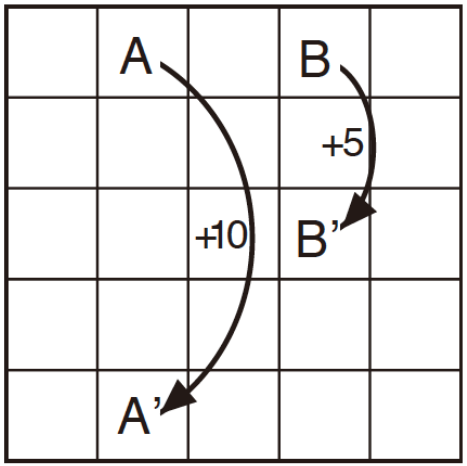
(a)

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

(b)

What is the value function for the uniform random policy?

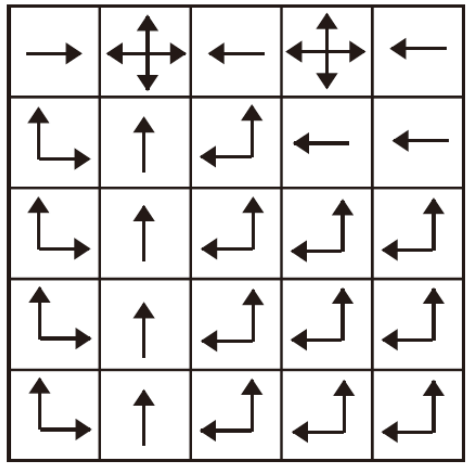
Gridworld Example: Control



a) gridworld

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

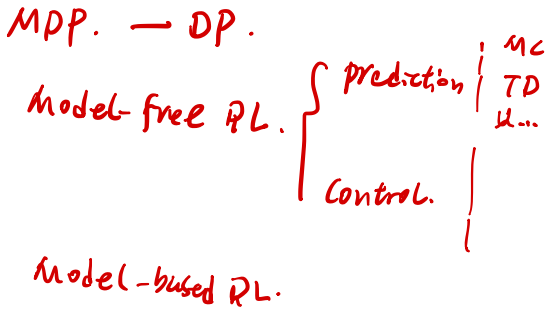
b) v_*



c) π_*

What is the optimal value function over all possible policies?
What is the optimal policy?

Connections with Psychology



- Prediction & Control algorithms in reinforcement learning parallels Classical & Instrumental conditioning in animal learning.
- Environment Models in reinforcement learning parallels Cognitive Maps in animal learning.
  - they can be learned by supervised learning methods without relying on reward signals
  - they can be used later to plan behavior
- Model-free & Model-based algorithms in reinforcement learning parallels Habitual & Goal-directed behavior in pyschology.

Outline

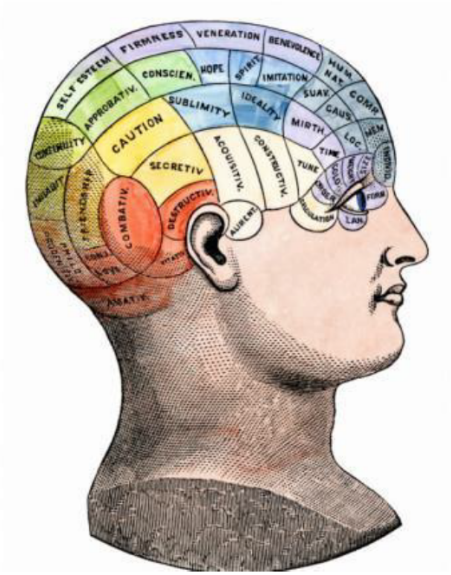
Reinforcement Learning

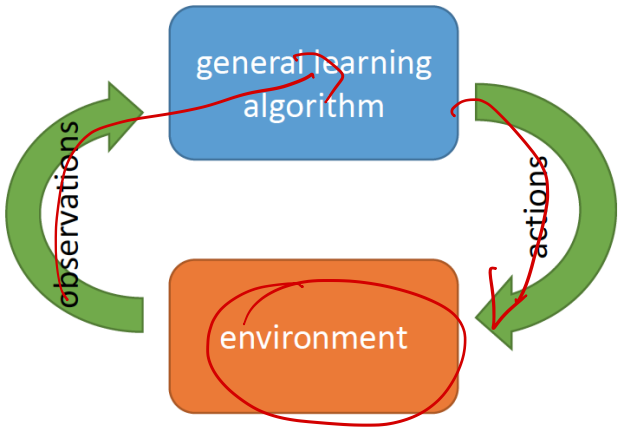
- Comprehensive, but challenging form of machine learning
  - Stochastic environment (usually non-stationary)
  - Incomplete model
  - Interdependent sequence of decisions
  - No supervision
  - Partial and delayed feedback
  - Trail and error with a balance between exploration and exploitation
  - The fleeting nature of time and online data
- Long term goal: lifelong learning & intelligence

Intelligence: Ultimate Goal

Instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain.

- Alan Turing





Outline

Main references

- Reinforcement Learning: An Introduction (second edition), R. Sutton & A. Barto, 2018.
- RL course slides from Richard Sutton, University of Alberta.
- RL course slides from David Silver, University College London.