ITCS 6156/8156 Fall 2023
Machine Learning

Reinforcement Learning

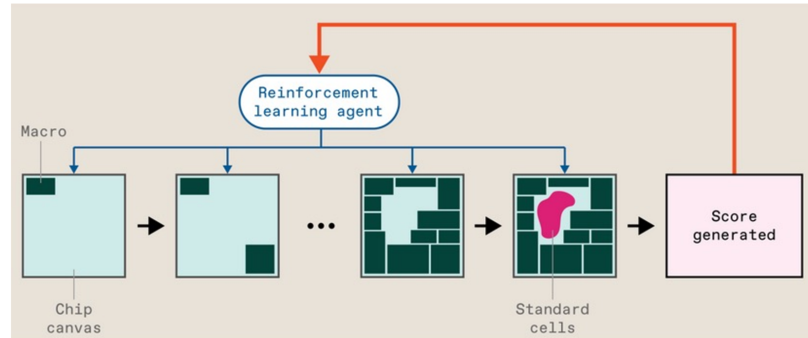
Instructor: Hongfei Xue
Email: hongfei.xue@charlotte.edu
Class Meeting: Mon & Wed, 4:00 PM – 5:15 PM, CHHS 376

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

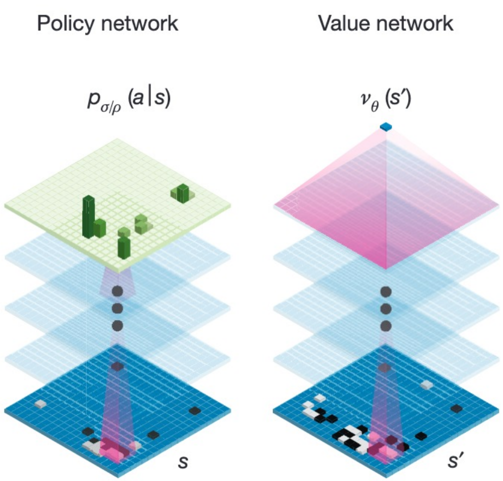
Some content in the slides is based on Dr. Raquel Urtasun’s lecture

Fancy Applications of RL

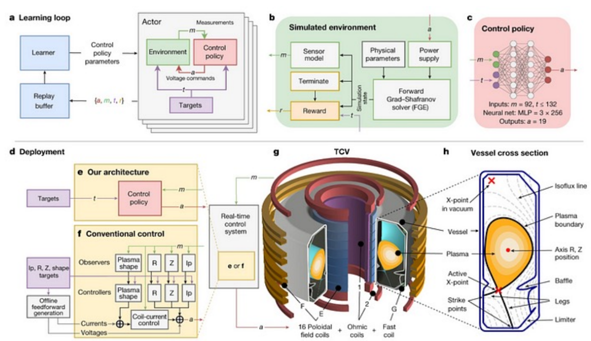
Play Dota



Chip Design



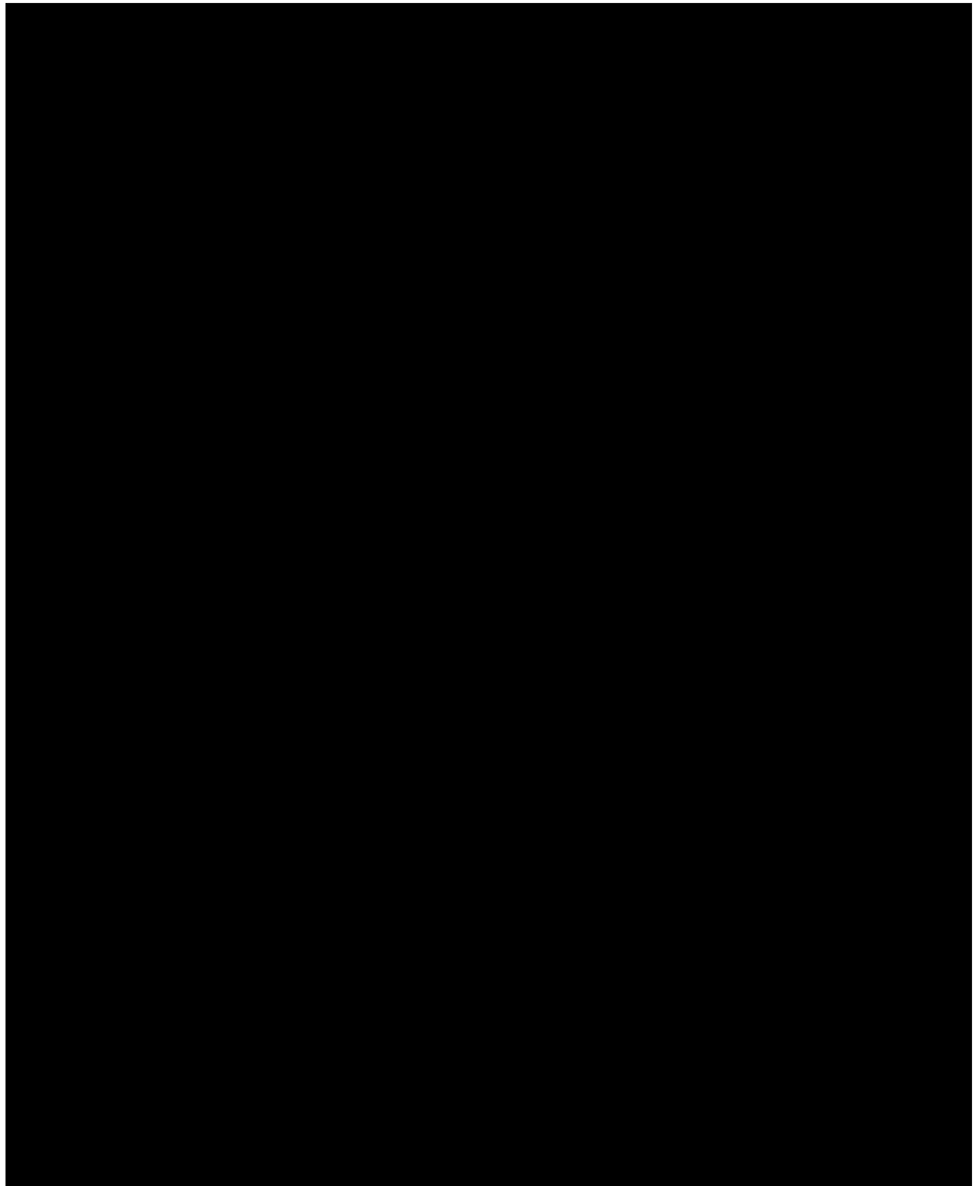
AlphaGo



Nuclear Fusion

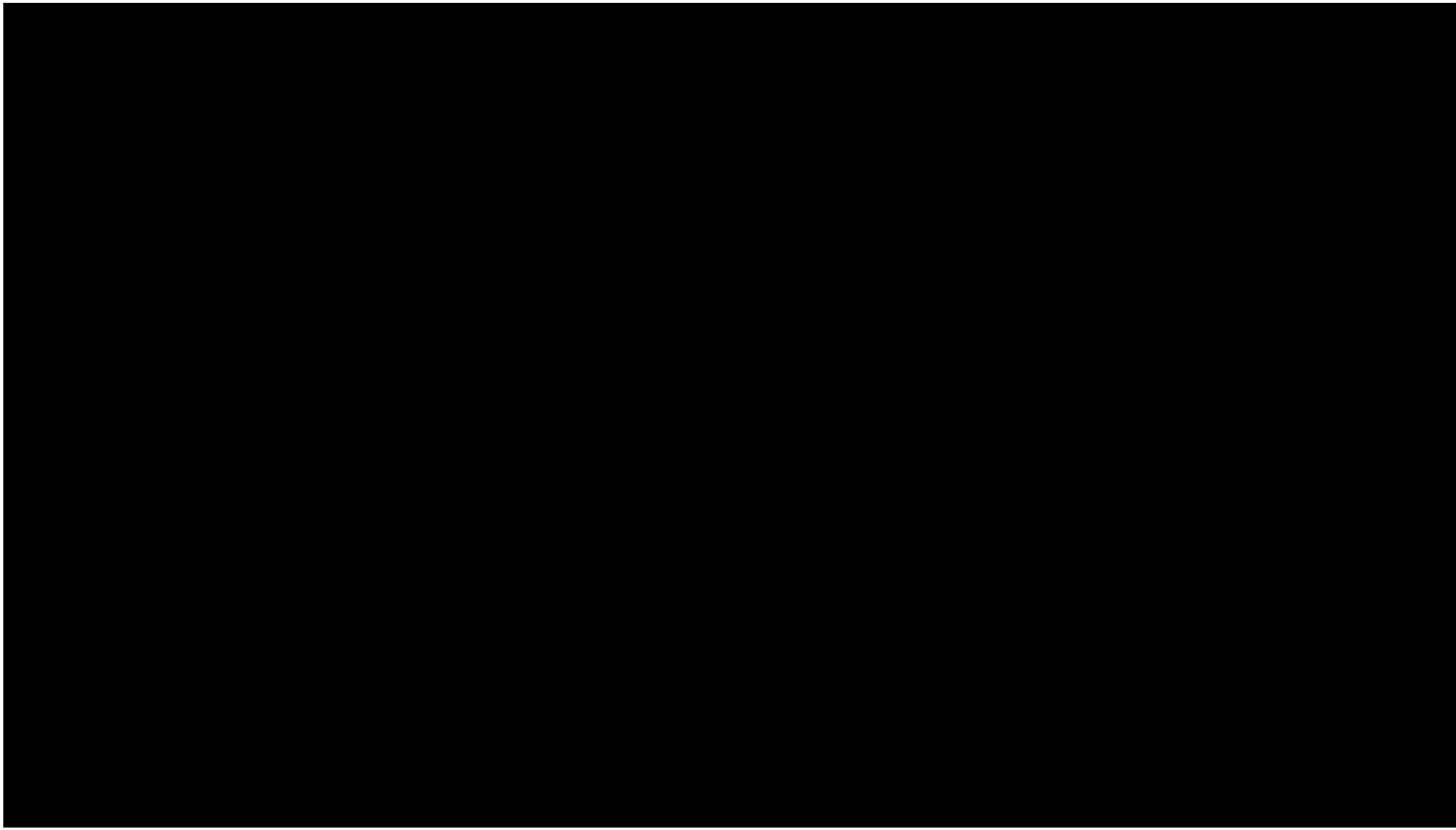
AWS Deep Racer

Playing Games: Atari



From YouTube: https://www.youtube.com/watch?v=V1eYniJ0Rnk

Playing Games: Hide and Seek

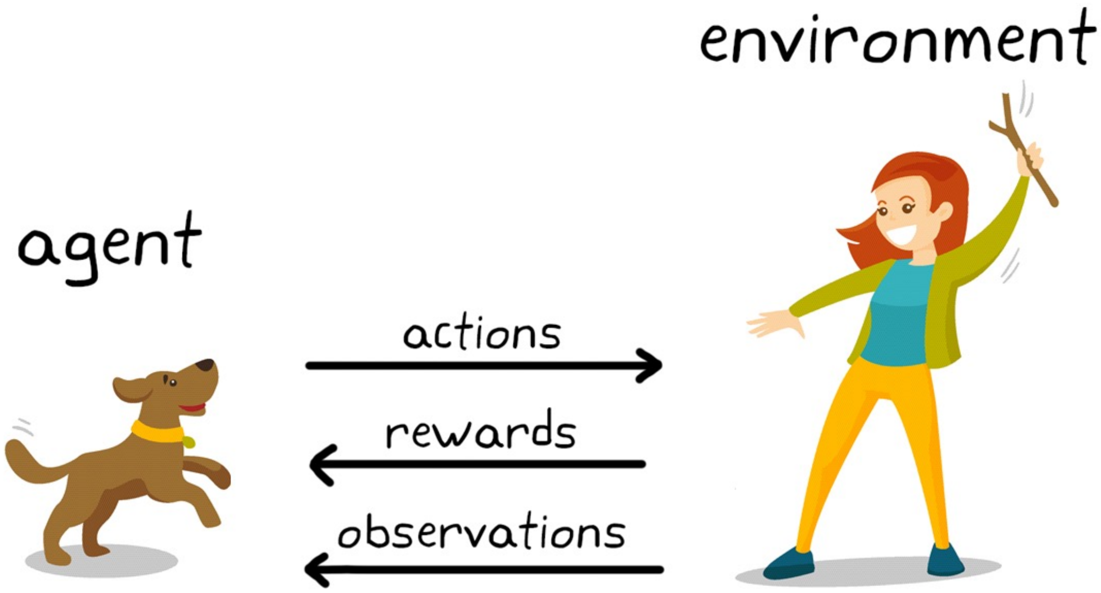


From YouTube: https://www.youtube.com/watch?v=Lu56xVlZ40M

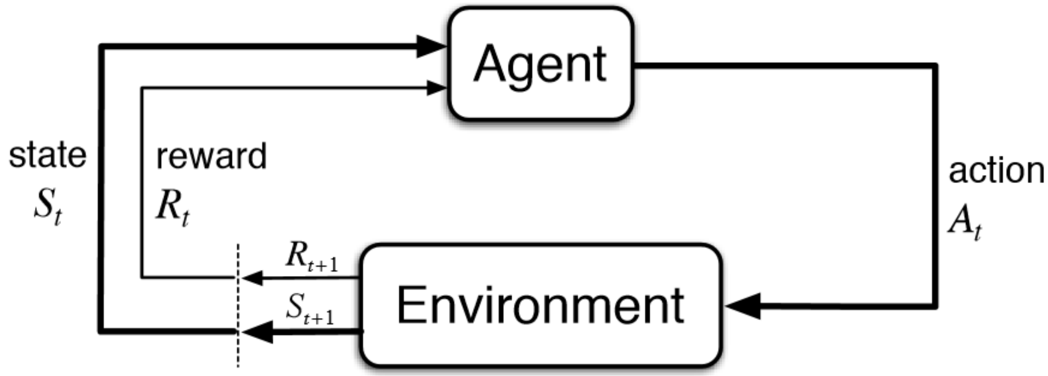
Reinforcement Learning

- Learning algorithms differ in the information available to learner
  - Supervised: correct outputs
  - Unsupervised: no feedback, must construct measure of good output
  - Reinforcement learning
- More realistic learning scenario:
  - Continuous stream of input information, and actions
  - Effects of action depend on state of the world
  - Obtain reward that depends on world state and actions
    - not correct response, just some feedback

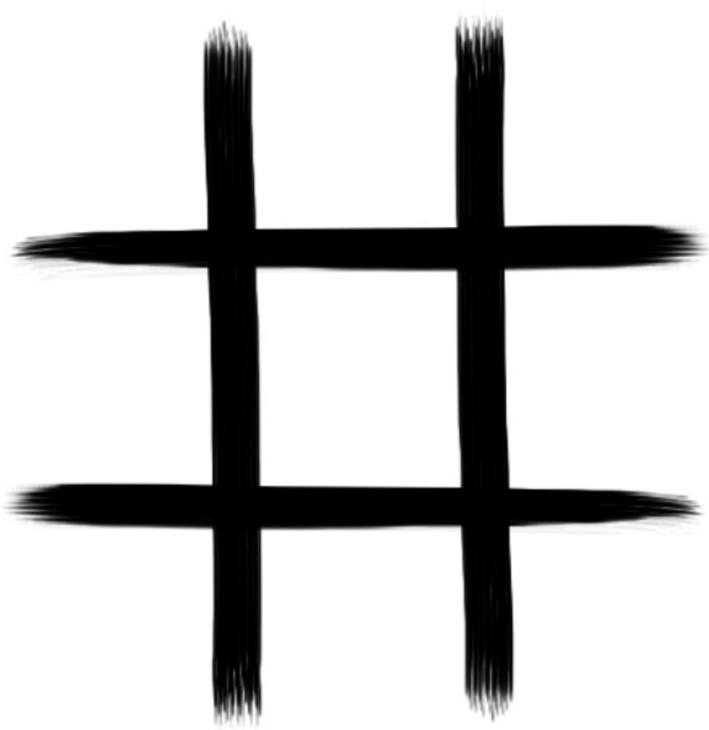
Reinforcement Learning



Reinforcement Learning in Dog Training

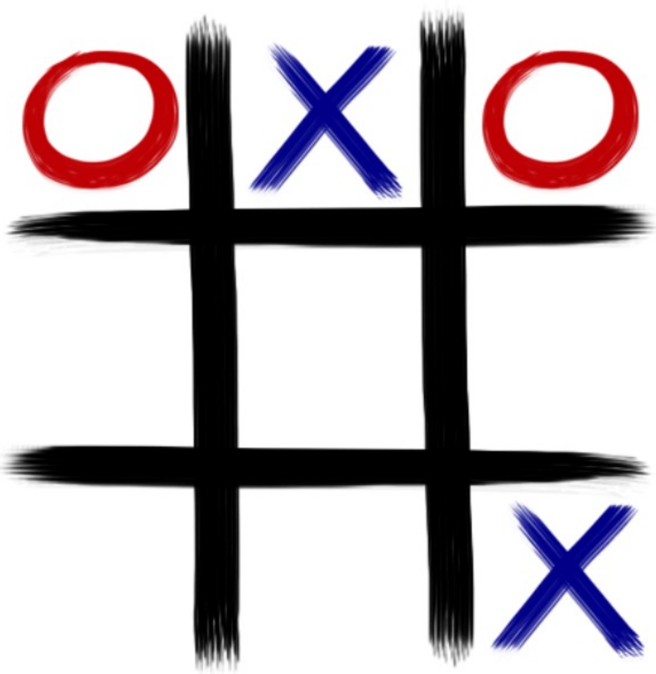


Example: Tic Tac Toe, Notation



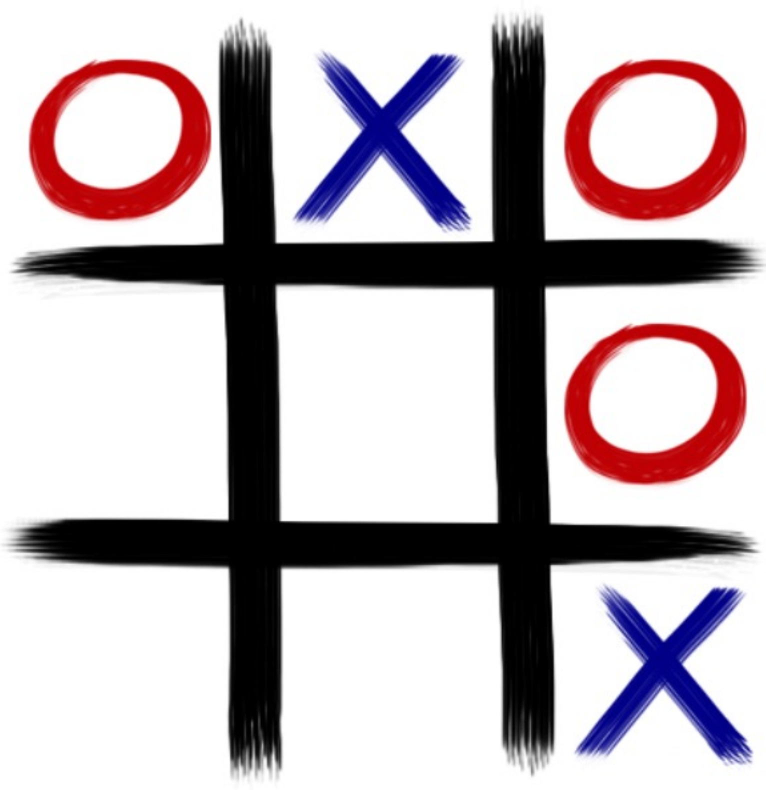
environment

Example: Tic Tac Toe, Notation



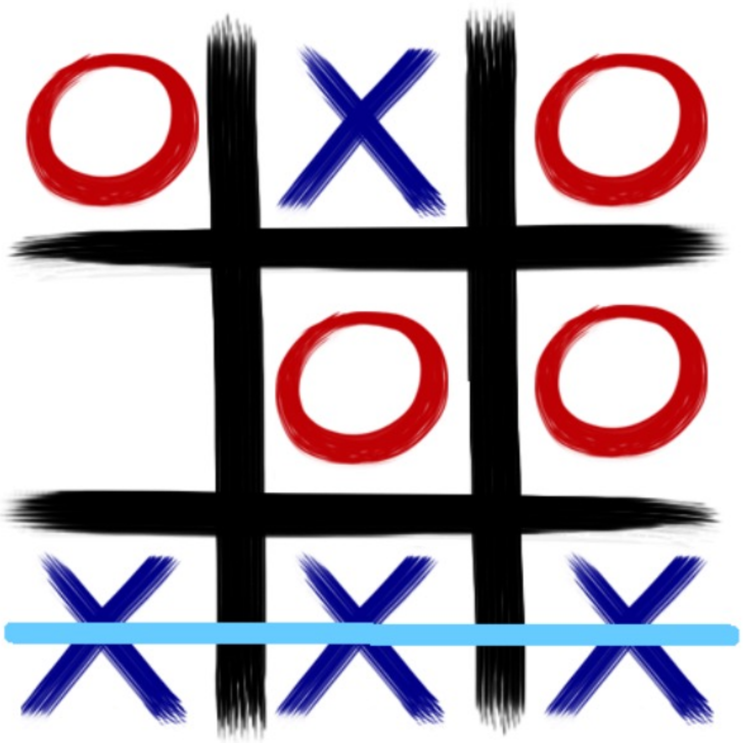
(current)
state

Example: Tic Tac Toe, Notation



action

Example: Tic Tac Toe, Notation



reward

(here: -1)

Formulating Reinforcement Learning

- World described by a discrete, finite set of states and actions
- At every time step t, we are in a state s_t, and we:
  - Take an action a_t (possibly null action)
  - Receive some reward r_{t+1}
  - Move into a new state s_{t+1}
- An RL agent may include one or more of these components:
  - Policy π: agent's behaviour 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 behaviour.
- It's a selection of which action to take, based on the current state
- Deterministic policy: $a = \pi(s)$
- Stochastic policy: $\pi(a|s) = P[a_t = a|s_t = s]$

Value Function

- Value function is a prediction of future reward
- Used to evaluate the goodness/badness of states
- Our aim will be to maximize the value function (the total reward we receive over time): find the policy with the highest expected reward
- By following a policy π, the value function is defined as:

$$V^{\pi}(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$$

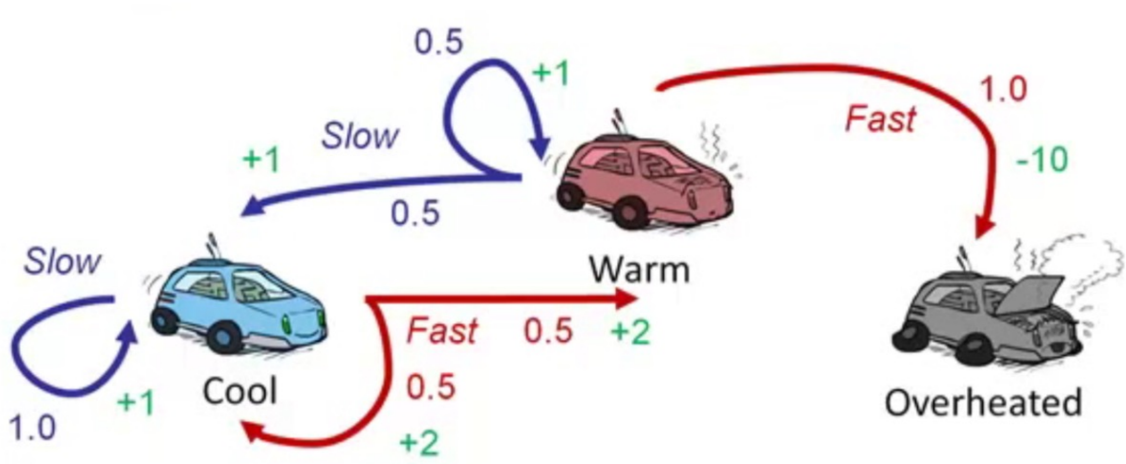
- γ is called a discount rate, and it is always $0 \leq \gamma \leq 1$
- If γ close to 1, rewards further in the future count more, and we say that the agent is "farsighted"
- γ is less than 1 because there is usually a time limit to the sequence of actions needed to solve a task (we prefer rewards sooner rather than later)

Model

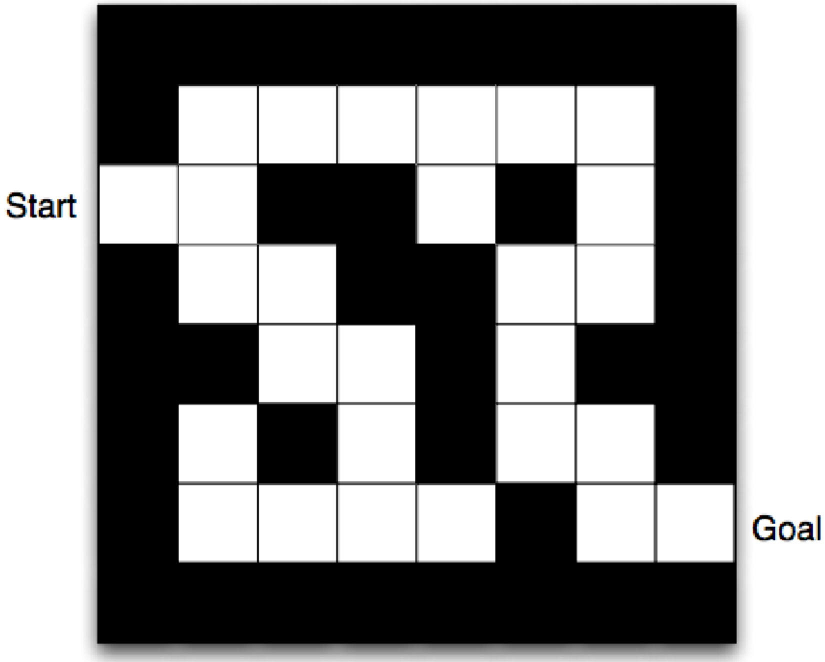
- The model describes the environment by a distribution over rewards and state transitions:

$$P(s_{t+1} = s', r_{t+1} = r' | s_t = s, a_t = a)$$

- We assume the Markov property: the future depends on the past only through the current state

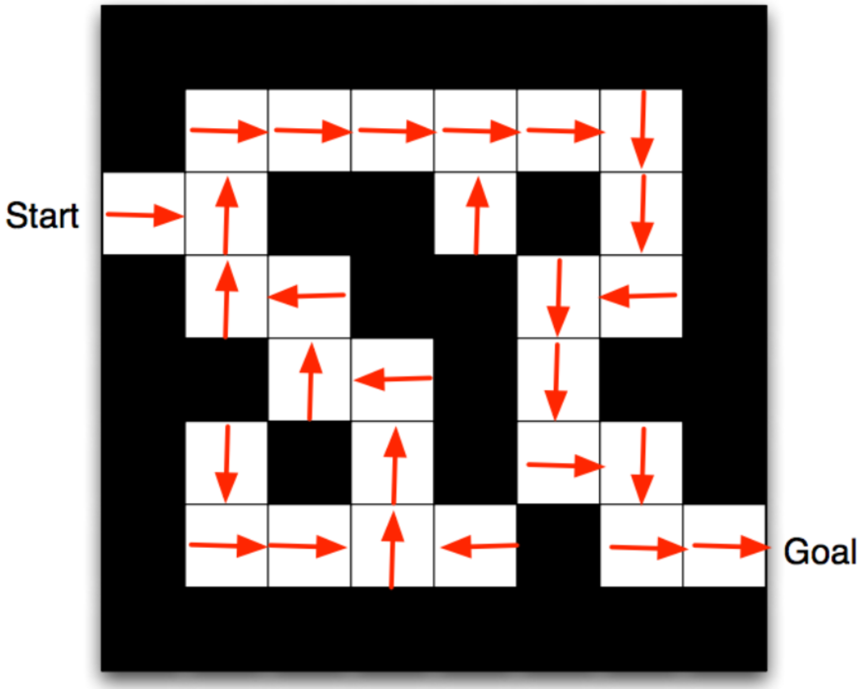


Maze Example



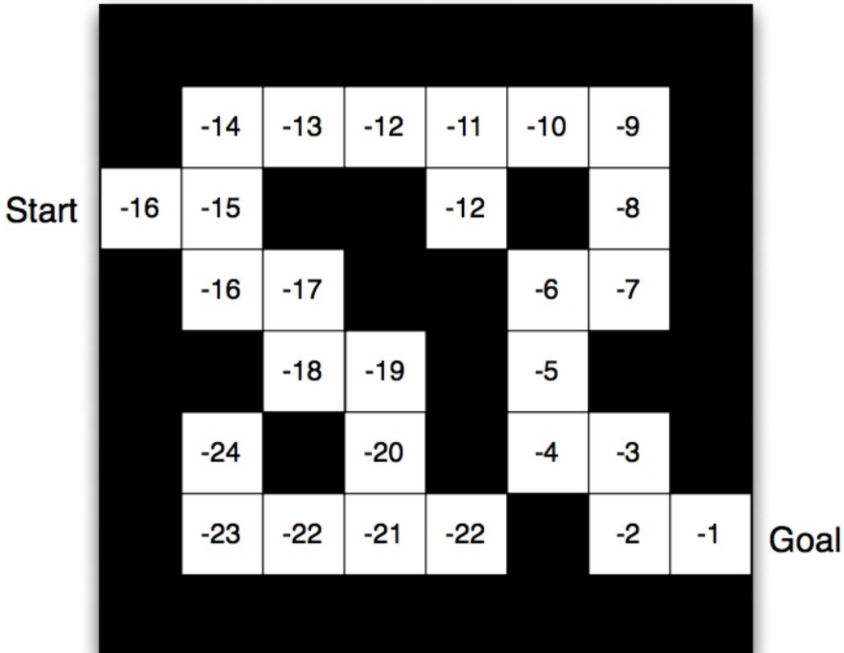
- Rewards: -1 per time-step
- Actions: N, E, S, W
- States: Agent's location

Maze Example



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

Maze Example



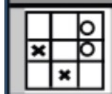
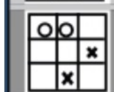
- Numbers represent value $V^{\pi}(s)$ of each state s

Example: Tic-Tac-Toe

- Consider the game tic-tac-toe:
  - reward: win/lose/tie the game $(+1/-1/0)$ [only at final move in given game]
  - state: positions of X's and O's on the board
  - policy: mapping from states to actions
    - based on rules of game: choice of one open position
  - value function: prediction of reward in future, based on current state
- In tic-tac-toe, since state space is tractable, can use a table to represent value function

RL & Tic-Tac-Toe

- Each board position (taking into account symmetry) has some probability

State	Probability of a win (Computer plays "o")
	0.5
	0.5
	1.0
	0.0
	0.5
etc	

- Simple learning process:
  - start with all values = 0.5
  - policy: choose move with highest probability of winning given current legal moves from current state
  - update entries in table based on outcome of each game
  - After many games value function will represent true probability of winning from each state
- Can try alternative policy: sometimes select moves randomly (exploration)

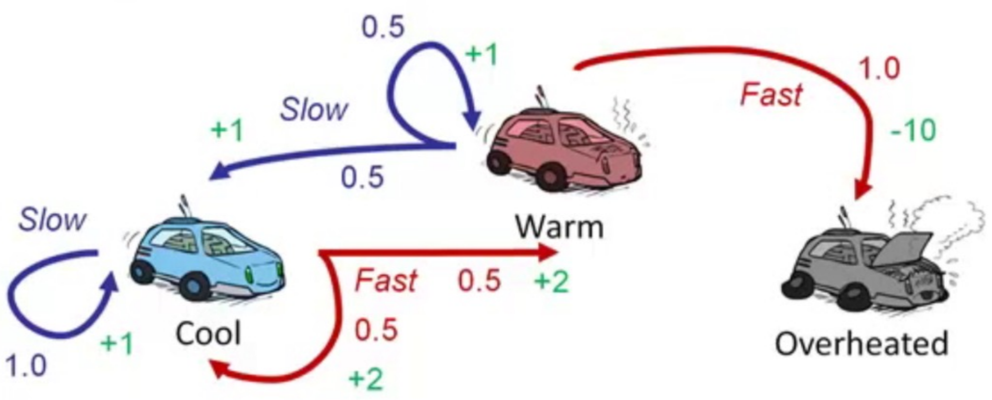
Basic Problems

- Markov Decision Problem (MDP): tuple (S, A, P, γ) where P is

$$P(s_{t+1} = s', r_{t+1} = r' | s_t = s, a_t = a)$$

- Standard MDP problems:

  1. Planning: given complete Markov decision problem as input, compute policy with optimal expected return

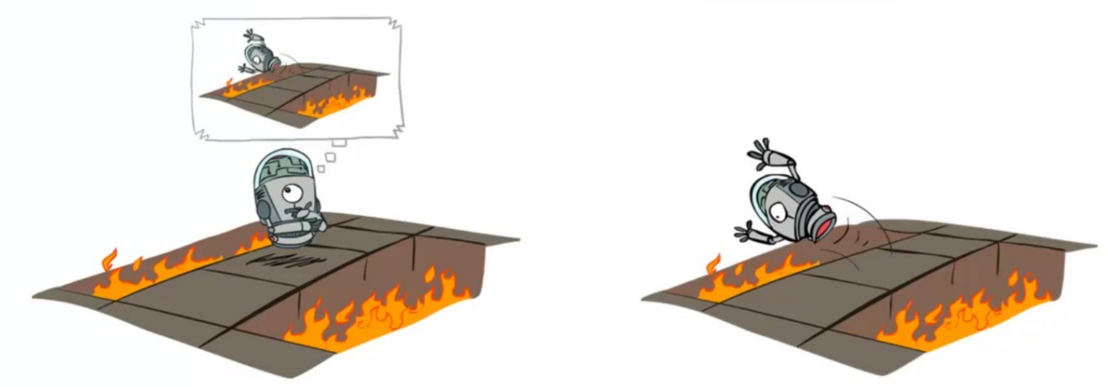


Basic Problems

- Markov Decision Problem (MDP): tuple (S, A, P, γ) where P is

$$P(s_{t+1} = s', r_{t+1} = r' | s_t = s, a_t = a)$$

- Standard MDP problems:
  1. Planning: given complete Markov decision problem as input, compute policy with optimal expected return
  2. Learning: We don't know which states are good or what the actions do. We must try out the actions and states to learn what to do



We will focus on learning, but discuss planning along the way

Exploration vs. Exploitation

- If we knew how the world works (embodied in P), then the policy should be deterministic
  - just select optimal action in each state
- Reinforcement learning is like trial-and-error learning
- The agent should discover a good policy from its experiences of the environment
- Without losing too much reward along the way
- Since we do not have complete knowledge of the world, taking what appears to be the optimal action may prevent us from finding better states/actions
- Interesting trade-off:
  - immediate reward (exploitation) vs. gaining knowledge that might enable higher future reward (exploration)

Examples

- Restaurant Selection
  - Exploitation: Go to your favourite restaurant
  - Exploration: Try a new restaurant
- Online Banner Advertisements
  - Exploitation: Show the most successful advert
  - Exploration: Show a different advert
- Oil Drilling
  - Exploitation: Drill at the best known location
  - Exploration: Drill at a new location
- Game Playing
  - Exploitation: Play the move you believe is best
  - Exploration: Play an experimental move

Questions?