Introduction to Reinforcement Learning

Bogdan Ivanyuk-Skulskyi, Dmytro Kuzmenko

Department of Mathematics,
National University of Kyiv-Mohyla Academy

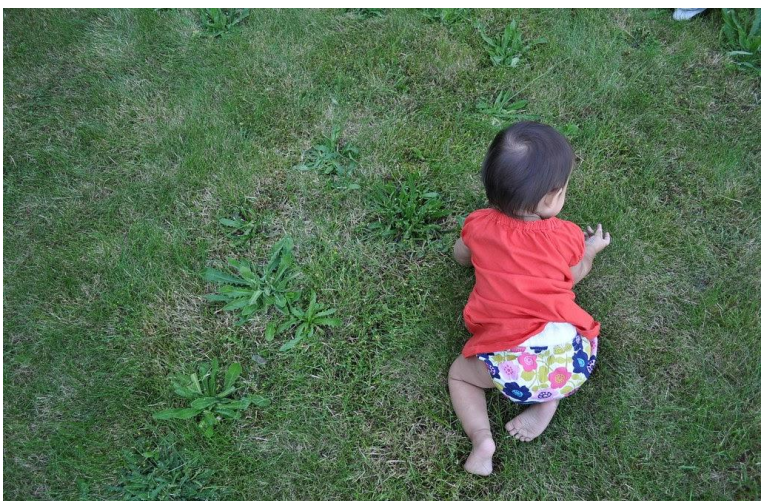
February 6, 2023

Plan

- **Overview of the reinforcement learning**
- Course logistics
- Introduction to sequential decision making under uncertainty

Make good sequence of decisions

Make good sequence of decisions



Reinforcement learning

Fundamental challenge in artificial intelligence and machine learning is learning to make good decisions under uncertainty

2010s: RL IN GAMES

Figure. DeepMind Nature, 2015

2010s: Robotics

Figure. Chelsea Finn, Sergey Levine, Pietel Abbeel

Expanding Reach. Health

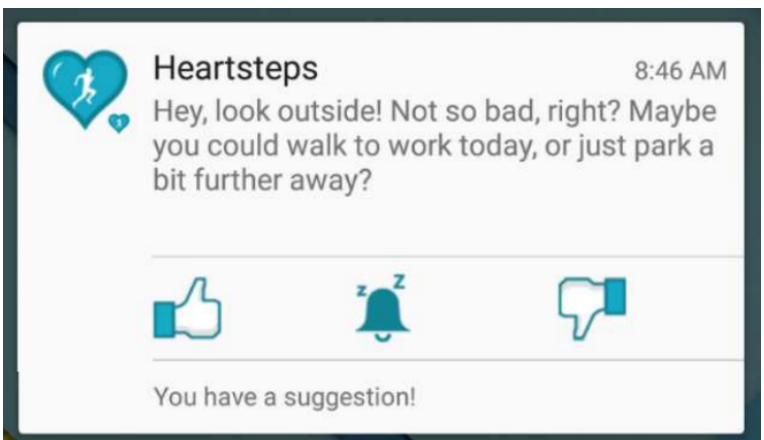


Figure. Personalized HeartSteps: A Reinforcement Learning Algorith for Optimizing Physical Activity. Liao, Greenewald, Klasnja, Murphy 2019 arxiv

Reinforcement Learning Involves

- Optimization
- Delayed consequences
- Exploration
- Generalization

Optimization

- Goal is to find an optimal way to make decisions
  - yield best outcomes or at least very good outcomes
- Explicit notion of utility of decisions

Example: finding the minimum distance route between two cities given network of roads

Delayed Consequences

- Decisions now can impact things much later ...
  - Saving for retirement
  - Finding a key in video game Montezuma's revenge
- Introduces two challenges
  - When planning: decisions involve reasoning about not just immediate benefit of a decision but also its longer term ramifications
  - When learning: temporal credit assignment is hard (what caused later high or low rewards?)

Exploration

- Learning about the world by making decisions
  - Agent as scientist
  - Learn to ride a bike by trying (and failing)
  - Finding a key in Montezuma's revenge
- Censored data
  - Only get a reward (label) for decision made
  - Don't know what would happen if we had taken red pill insted of blue pill
- Decisions impact what we learn about
  - If we choose to go to CS instead of Applied Math, we will have different later experiences...

Generalization

Policy is a mapping from past experience to action
Why not just pre-program a policy?

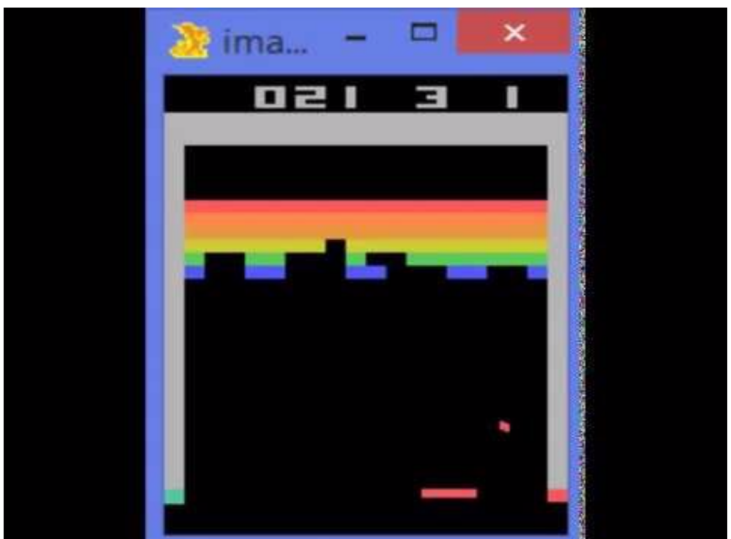


Figure. DeepMind Nature, 2015

How many possible images are there?

Generalization

Policy is a mapping from past experience to action
Why not just pre-program a policy?

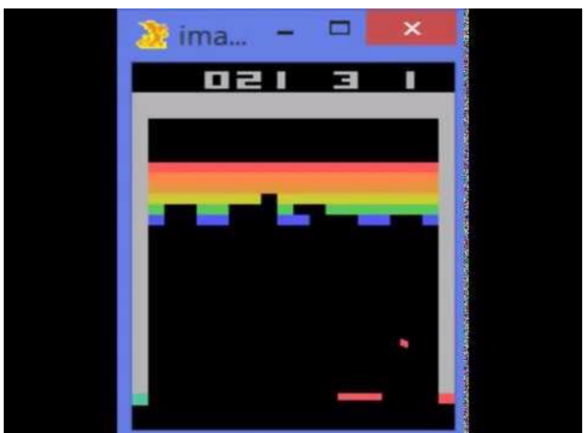


Figure. DeepMind Nature, 2015

How many possible images are there?

- $256^{(100\times200)^3}$

RL vs Other AI and Machine Learning

Supervised learning

	Supervised Learning	Unsupervised Learning	Reinforcement Learning	Imitation Learning
Optimization				
Learns from experience	x			
Generalization	x			
Delayed Consequences				
Exploration				

RL vs Other AI and Machine Learning

Unsupervised learning

	Supervised Learning	Unsupervised Learning	Reinforcement Learning	Imitation Learning
Optimization				
Learns from experience	x	x		
Generalization	x	x		
Delayed Consequences				
Exploration				

RL vs Other AI and Machine Learning

Reinforcement learning

	Supervised Learning	Unsupervised Learning	Reinforcement Learning	Imitation Learning
Optimization			x	
Learns from experience	x	x	x	
Generalization	x	x	x	
Delayed Consequences			x	
Exploration			x	

RL vs Other AI and Machine Learning

Imitation learning

	Supervised Learning	Unsupervised Learning	Reinforcement Learning	Imitation Learning
Optimization			x	x
Learns from experience	x	x	x	x
Generalization	x	x	x	x
Delayed Consequences			x	x
Exploration			x	

Issues

- Where do rewards come from?
  - And what happens if we get it wrong?
- Robustness / Risk sensitivity
- Multi-agent RL

Plan

- Overview of the reinforcement learning
- **Course logistics**
- Introduction to sequential decision making under uncertainty

Course overview

- Lectures: Monday 18:00
- Practicals: Wednesday 18:00

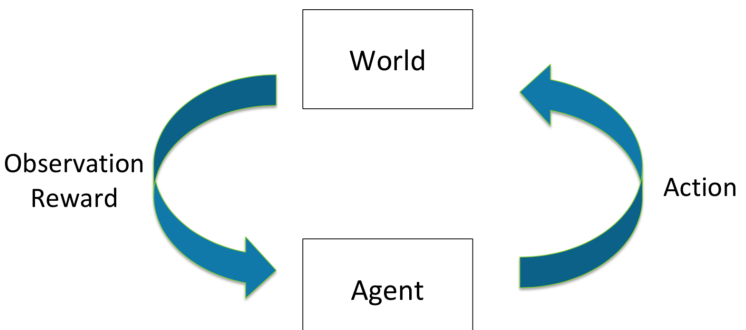
- Materials: GitHub + DistEdu (later)

- Homeworks (code + pen paper)
  - 30 points each
  - 25 points (late policy: 1 week late)
  - 20 points (late policy: 2 weeks late and on)
- Paper review
  - 10 points
  - 15-20 minutes presentation

Plan

- Overview of the reinforcement learning
- Course logistics
- **Introduction to sequential decision making under uncertainty**

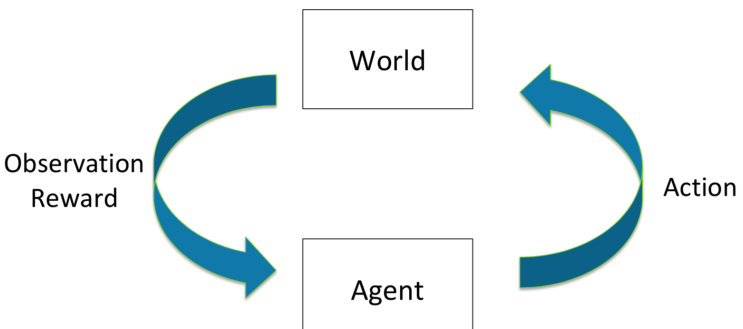
Sequential Decision Making



Goal: select actions to maximize total expected future reward

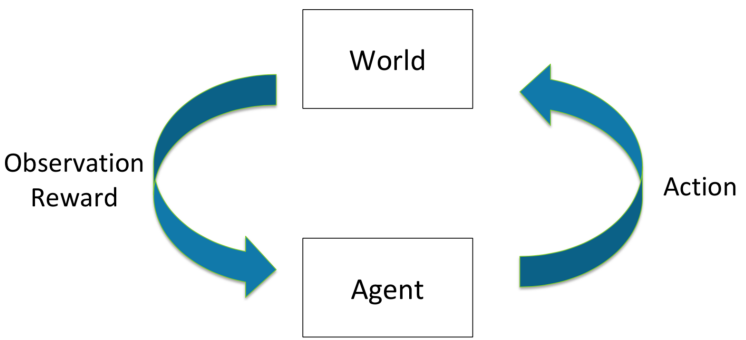
May require balancing immediate long term rewards

TIME



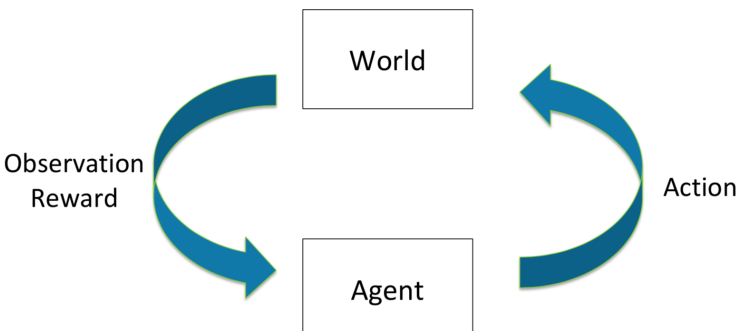
- Each time step t:
  - Agent takes an action a_t
  - World updates given action a_t, emits observation o_t and reward r_t
  - Agent receives observation o_t and reward r_t

History



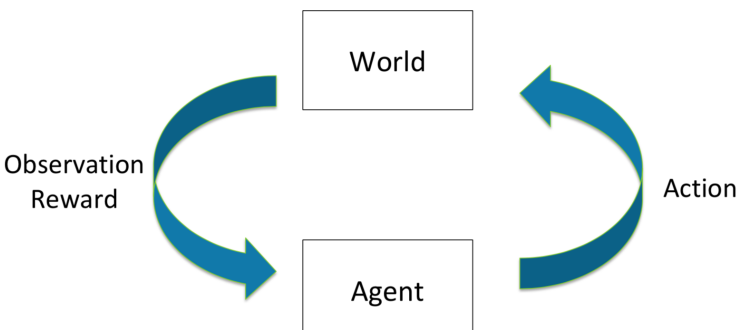
- History $h_t = (a_1, o_1, r_1, ..., a_t, o_t, r_t)$
- Agent chooses action based on history
- State is information assumed to determine what happens next
  - Function of history: $s_t = (h_t)$

World State



- This is true state of the world used to determine how world generated next observation and reward
- Often hidden or unknown to agent
- Even if known may contain information not needed by agent

Agent State: Agent's Internal Representation



- What the agent / algorithm uses to make decisions about how to act
- Generally a function of the history: $s_t = f(h_t)$
- Could include meta information like state of algorithm (how many computations executed, etc) or decision process (how many decisions left until an episode ends)

Markov Assumption

- Information state: sufficient statistic of history
- State s_t is Markov if and only if:

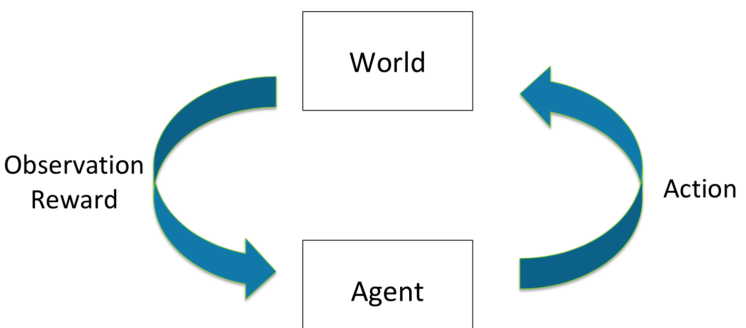
$$p(s_{t+1}|s_t, a_t) = p(s_{t+1}|h_t, a_t) \tag{1}$$

- Future is independent of past given present

Why is Markov Assumption Popular?

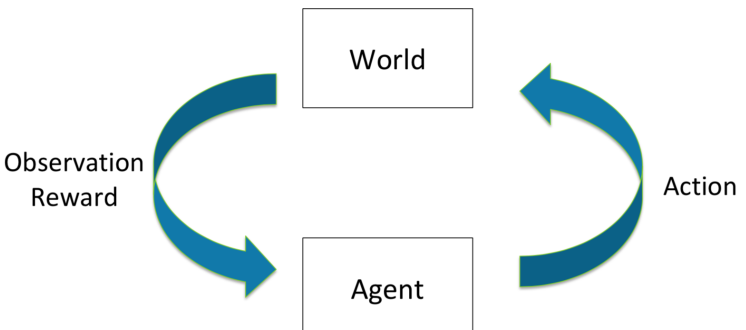
- Can always be satisfied
  - Setting state as history is always Markov: $s_t = h_t$
- In practise often assume most recent observation is sufficient statistic of history: $s_t = o_t$
- State representation has big implications for:
  - Computational complexity
  - Data required
  - Resulting performance

Full Observability / Markov Decision Process (MDP)



Environment and world state $s_t = o_t$

Full Observability / Markov Decision Process (MDP)



- Is state Markov? Is world partially observable? (POMDP)
- Are dynamics deterministic or stochastic?
- Do actions influence only immediate reward or reward and next state?

Example: Mars Rover as a MDP

S1	S2	S3	S4	S5	S6	S7
						

- States: Location of rover ($s_1, ..., s_7$)
- Actions: TryLeft or TryRight
- Rewards:
  - +1 in state s_1
  - +10 in state s_7
  - 0 in all other states

RL Algorithm Components

Often includes one or more: Model, Policy, Value Function

MDP Model

Agent's representation of how world changes given agent's action

Transition / dynamics model predicts next agent state

$$p(s_{t+1} = s' | s_t = s, a_t = a) \tag{2}$$

Reward model predicts immediate reward

$$r(s_t = s, a_t = a) = \mathbb{E}[r_t | s_t = s, a_t = a] \tag{3}$$

Example: Mars Rover Stochastic Markov Model

S1	S2	S3	S4	S5	S6	S7
$\hat{r} = 0$	$\hat{r} = 0$	$\hat{r} = 0$	$\hat{r} = 0$	$\hat{r} = 0$	$\hat{r} = 0$	$\hat{r} = 0$

Numbers above show RL agent's reward model

Part of agent's transition model:

- $0.5 = P(s_1|s_1, TryRight) = P(s_2|s_1, TryRight)$
- $0.5 = P(s_2|s_2, TryRight) = P(s_3|s_2, TryRight)...$

Model may be wrong

Policy

- Policy determines how the agent chooses actions
- $\pi : S \to A$, mapping from states to actions
- Deterministic policy:

$$\pi(s) = a \tag{4}$$

- Stochastic policy:

$$\pi(a|s) = P(a_t = a|s_t = s) \tag{5}$$

Value Function

- Value function V^{π}: expected discounted sum of future rewards under a particular policy π

$$V^{\pi}(s_t = s) = \mathbb{E}_{\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t_2} + ...|s_t = s] \tag{6}$$

- Discounted factor γ weights immediate vs future rewards
- Can be good to quantify goodness/badness of states and actions
- And decide how to act by comparing policies

Example: Mars Rover Value Function

S1	S2	S3	S4	S5	S6	S7
$V^{\pi}(s_1) = +1$	$V^{\pi}(s_2) = 0$	$V^{\pi}(s_3) = 0$	$V^{\pi}(s_4) = 0$	$V^{\pi}(s_5) = 0$	$V^{\pi}(s_6) = 0$	$V^{\pi}(s_7) = +10$

- Discounted factor, $\gamma = 0$
- $\pi(s_1) = \pi(s_2) = ... = \pi(s_7) =$TryRight
- Numbers show value $V^{\pi}(s)$ for this policy and this discount factor

Types of RL Agents

- Model-based
  - Explicit: Model
  - May or may not have policy and/or value function
- Model-free
  - Explicit: Value function and/or policy function
  - No model

Evaluation and Control

- Evaluation
  - Estimate/predict the expected rewards from following a given policy
- Control
  - Optimization: find the best policy

Example: Mars Rover Policy Evaluation

S1	S2	S3	S4	S5	S6	S7
$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$

- Discounted factor, $\gamma = 0$
- $\pi(s_1) = \pi(s_2) = ... = \pi(s_7) =$TryRight
- What is the value of this policy?

$$V^{\pi}(s_t = s) = \mathbb{E}_{\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ...|s_t = s]$$

Example: Mars Rover Policy Evaluation

S1	S2	S3	S4	S5	S6	S7
$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$

- Discounted factor, $\gamma = 0$
- $\pi(s_1) = \pi(s_2) = ... = \pi(s_7) =$TryRight
- What is the value of this policy?

$$V^{\pi}(s_t = s) = \mathbb{E}_{\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ...|s_t = s]$$

- Answer:

$$V^{\pi}(s_t = s) = r(s)$$