Artificial Intelligence and Data Analytics for Engineers (AIDAE)

Lecture 8
June, 26th

Andrés Posada

Anas Abdelrazeq

Marco Kemmerling

Vladimir Samsonov

Today's Lecturer

Artificial Intelligence and Data Analytics for Engineers
Overview Lectures 1 – 4

1 **Introduction to Data Analytics and Artificial Intelligence in Engineering**: Organizational matters (e.g. exam, exercises, dates). Goals, Challenges, Obstacles, and Processes.

2 **Introduction into the primary programming language of the lecture, Python**: Syntax, libraries, IDEs etc. Why is Python the *lingua franca* of the Data Scientist?

3 **Data Preparation**: Cleansing and Transformation. How do real world data sets look like and why is cleaning and transformation an integral part of a Data Scientist's workflow?

4 **Data Integration**: Architectures, Challenges, and Approaches. How can you integrate various data sources into an overarching consolidating schema and why is this important?

Artificial Intelligence and Data Analytics for Engineers
Overview Lectures 5 – 8

5	**Data Representation**: Feature Extraction and Selection. How to pick relevant features for the task at hand. Manual vs automatic methods. What is the curse of dimensionality?
6	**Data-Driven Learning**: Supervised (Classification, Regression) methods and algorithms. What is an artificial neural net? What methods are there for evaluation of your model?
7	**Data-Driven Learning**: Unsupervised (Clustering) methods and algorithms. How can machines learn without labels? What methods are there for evaluation of your model?
8	**Environment-Driven Learning:** Reinforcement Learning

Today’s Lecture

Reinforcement Learning

What is it?

Methods

Applying

Learning Objectives

Learning Objective
w.r.t. Knowledge/Understanding.

After successfully completing this lecture, the students will have achieved the following learning outcomes:

- Have an understanding of what reinforcement learning is.
- Understand topics like agents, rewards, action.
- Know about the different types of reinforcement learning.

Motivation and Introduction

Reinforcement Learning

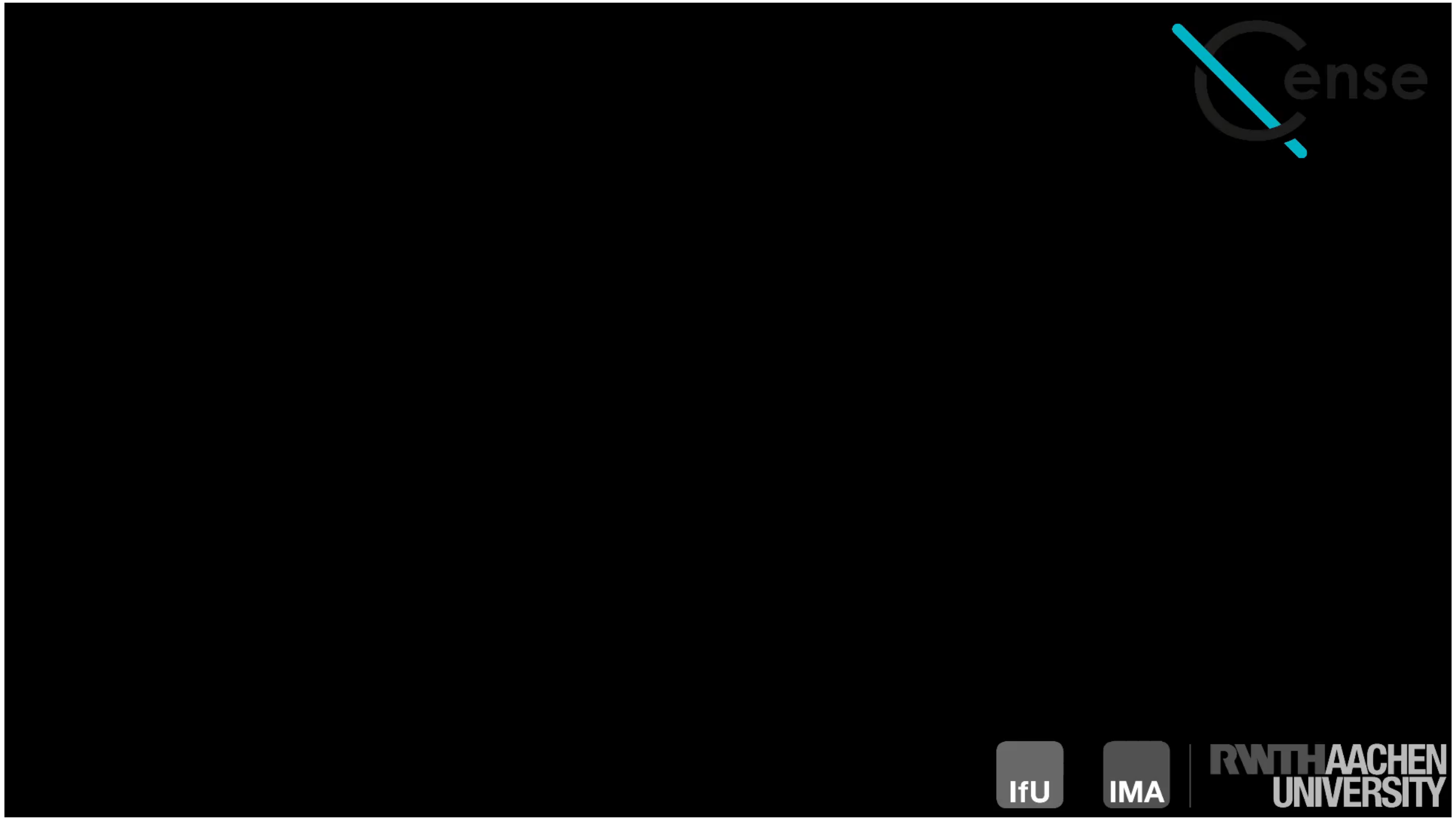
„Artificial Intelligence =
Deep Learning + Reinforcement Learning “,

David Silver, Google DeepMind 2015

- Deep Learning: Learning an abstract representation of data
- Reinforcement Learning: Learning parameters of representation from interaction with the environment

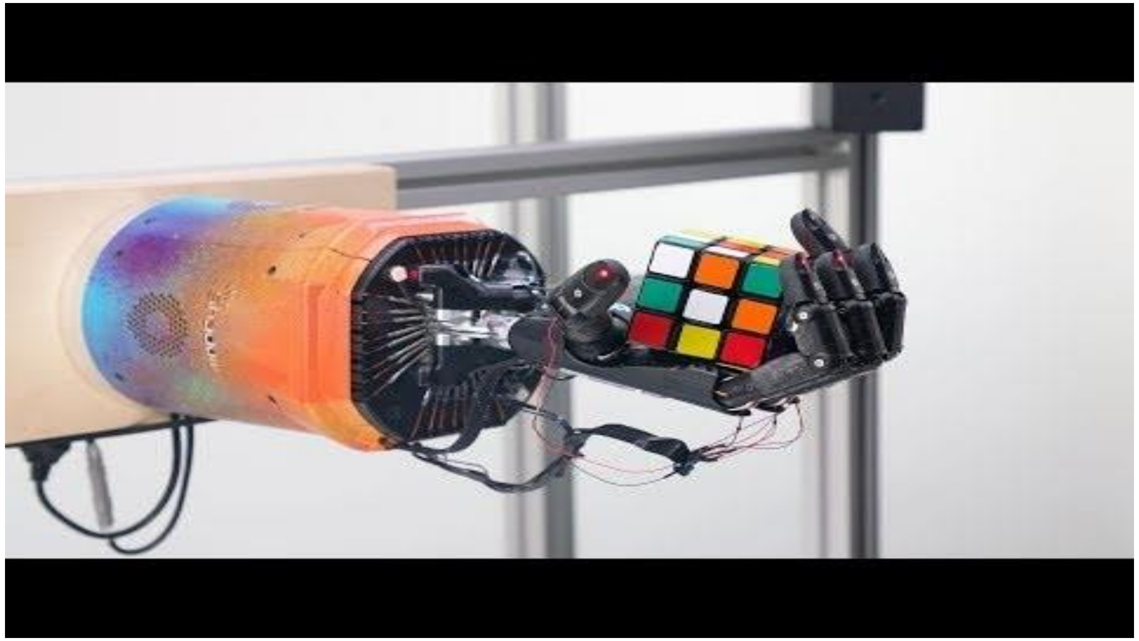
Reinforcement Learning in Action

Adaptive Path Planning

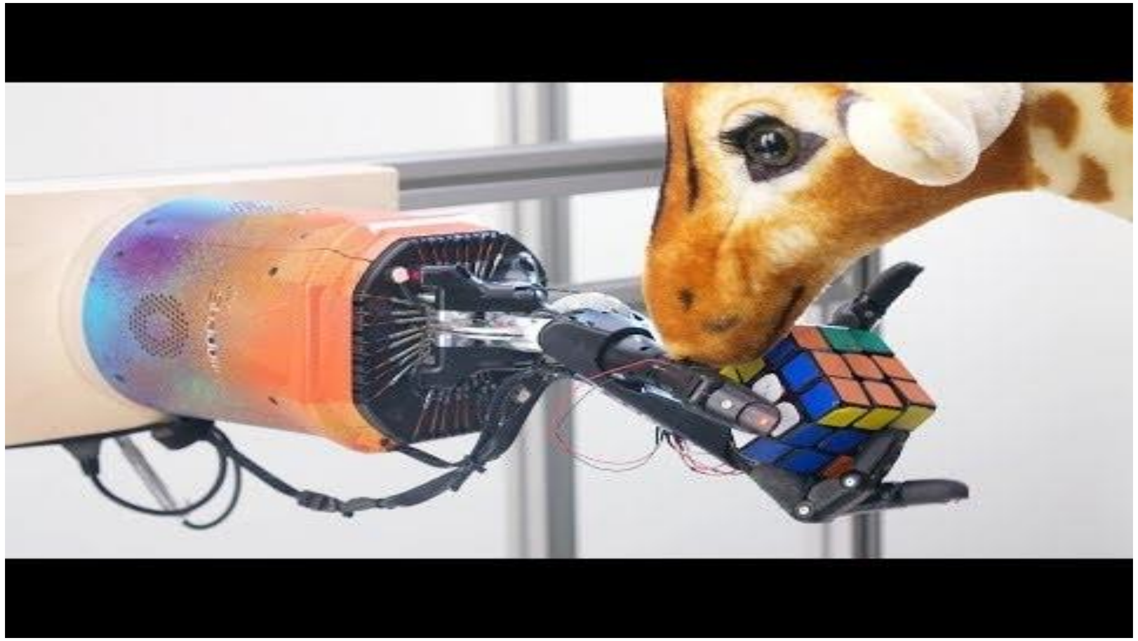


Reinforcement Learning in Action

Human-Level Dexterity with Robotics



Unperturbed



With perturbations

Solving complex manipulation and planning tasks in unknown conditions while been trained in the simulation only

source: https://openai.com/blog/solving-rubiks-cube/

Introduction

Introduction

- Brief view into Artificial intelligence and Machine Learning

Artificial Intelligence:
Any technique which enables a computer to mimic human behaviour.

General AI (GAI)
Transfer knowledge across domains.

Narrow AI
Performs a single task extremely well

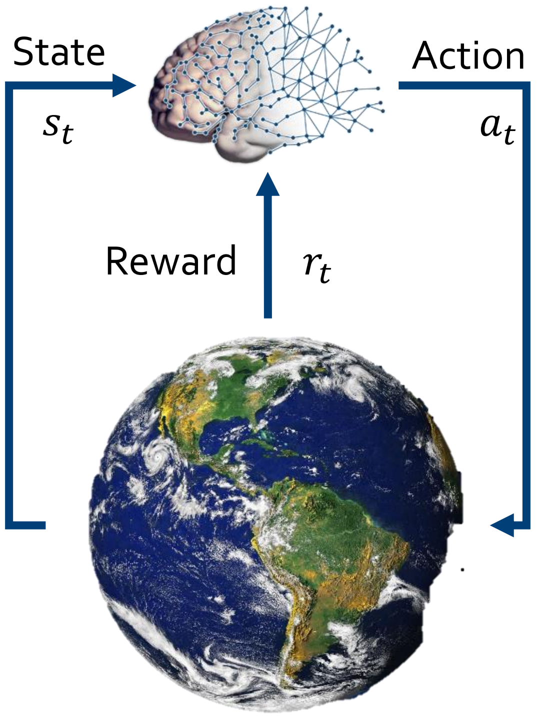
Machine Learning
Algorithms whose performance improve as they are exposed to more data.

- Supervised Learning
- Unsupervised Learning
- Reinforced Learning

What is reinforcement learning?

Main Idea
The agent learns from success and failure through reward and punishment.

- Goal of the agent is defined by the reward function.
- Agent receives Feedback in form of a reward
  With RL, the agent does not know the reward function!
- Agent must (learn) to act in such a way that the expected future reward is maximized
- All learning is based on samples of states, actions and rewards

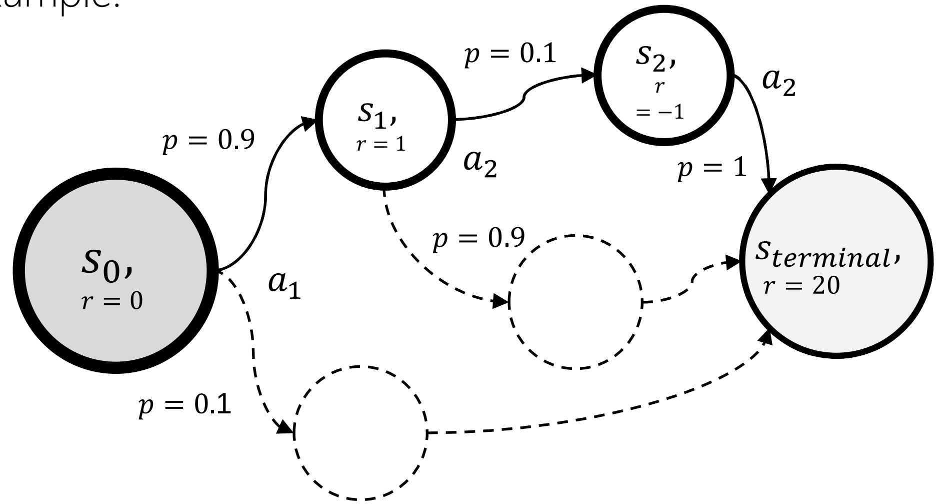


Essential Aspects of Reinforcement Learning

Essential Concepts

Reinforcement learning is formulated as a Markov decision-making process (MDP). This process is a tuple $\langle S, A, P, R, \gamma \rangle$

Example:



- S is a finite set of states
- A is a finite set of actions
- P is a transition matrix
  - $P_{ss'}^{a} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$
  - Probability that the action a leads from the state s to the state s'
- R is a reward function, $R_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$
- γ ist a discount factor $\gamma \in [0,1]$

Essential Concepts

Definition:

Markov Property:

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

Where:

S_t – current state of the agent
S_{t+1} – next state
P – state transition probability

In other words:

Future is Independent of the past given the present

Essential Concepts

Definition:
Expected Return – is a sum of future returns:
$G_t = R_{t+1} + \gamma G_{t+1}$

Where:

G_t – expected return of the current state
R_{t+1} – reward of the next state
γ – discount factor

How to understand this equation:

$G_1 = R_2 + \gamma R_3 \ldots + \gamma^8 R_{10} = R_2 + \gamma (R_3 \ldots + \gamma^7 R_{10})$
$G_2 = R_3 \ldots + \gamma^7 R_{10}$
$G_1 = R_2 + \gamma G_2$

Essential Concepts

Definition:
Value Function (State Value Function) – expected return for policy π in state S at time t

$V_\pi(S_t) = E_\pi\,(G_t \mid S_t)$

Where:

S_t – current state
π – policy RL agent follows
G_t – expected return for the current state

How to understand E_π:

- Actions taken by the agent can be stochastic
- The transition probabilities in real world environment can be stochastic

We need to account for probability of every possible transition in the current state:

$$E_\pi\,(G_t \mid S_t) = \sum_{a \in A} P(a \mid S_t) \sum_{S_{t+1} \in S} \sum_{R_{t+1} \in R} P(S_{t+1}, R_{t+1} \mid S_t, a)\, G_t$$

Where: S_t, S_{t+1} - current and next states; a - agent action, A, S, R- action , state and reward spaces; P denote conditional probabilities

Essential Concepts

Definition:
Q-Function (Action Value Function) – expected return for policy π in state S at time t assuming action a
$Q_{\pi}(S_t, a) = E_{\pi}\,(G_t \,|\, S_t, a)$

Where:

S_t – current state
π – policy RL agent follows
a – agent action
G_t – expected return for the current state

Essential Concepts

Definition:
Bellman Equation – calculating the value of the current state depends only on the possible next states
$V_\pi(S_t) = E_\pi\,(R_{t+1} + \gamma V_\pi(S_{t+1}) | S_t)$

Where:

S_t – current state
S_{t+1} – next state
R_{t+1} – reward of the next state
γ – discount factor
π – policy RL agent follows

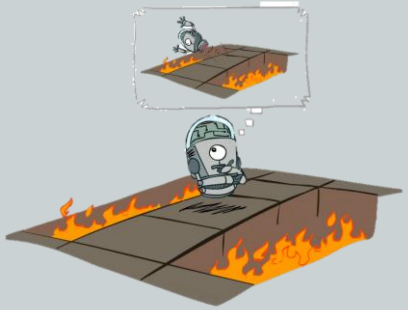
If environment transitions and RL policy are deterministic:

$V_\pi(S_t) = R_{t+1} + \gamma V_\pi\,(S_{t+1}) | S_t$

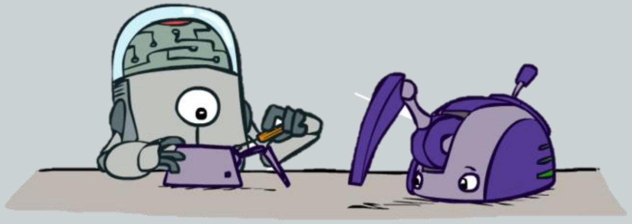
Most of the model-free RL algorithms are just trying to solve Bellman Equation!

Types of RL

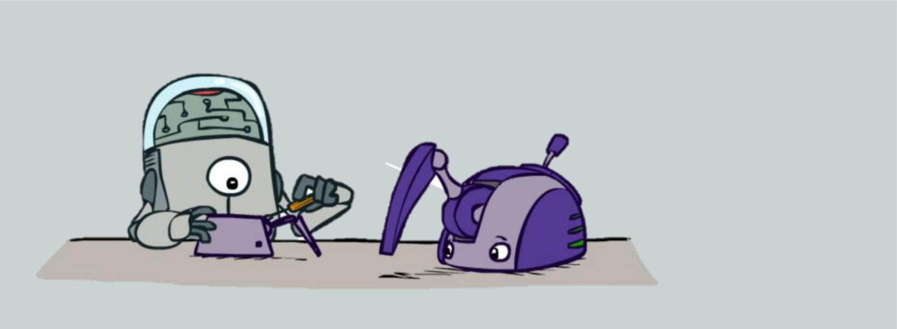
Case Distinctions

Offline Solution (MDPs)	Online Learning (RL)
	
Idea: • The state space, the action model and the transition model are simulated • Definition as MDP	**Idea:** • Agent executes actions (strategy may exist) • Rewards will be experienced • A model may be created
Procedure: • e.g. Dynamic Programming • Synchronous DP • Asynchronous DP	**Procedure:** • Model-based learning • Model-free learning • Active or passive learning

Case Distinctions

Model-based learning	Model-free learning
	
Model-based Idea: • Learning an approximated model from experience • Solve as if the learned values were correct	**Model-free idea:** • Direct learning • Strategy is derived directly • No detour via model identification
Learning of the empirical model: • Count values s' for all s, a • Normalize to obtain an estimate of the transition model $P_{ss'}^{a}$ • Explore any reward R_s^a	**Procedure:** • Active or passive learning

Case Distinctions

Model-based learning	Model-free learning
	

Model-based

Experience

$\{(s,a,r,s')\}$

Learning

Model

$P(s'|a,s)$

$R(s,a)$

Learning

Benefit

$V(s)$

$Q(s,a)$

Policy search

Planning

Choice of action

$\pi(a|s)$

Behavior / Strategy

Policy

Model-free

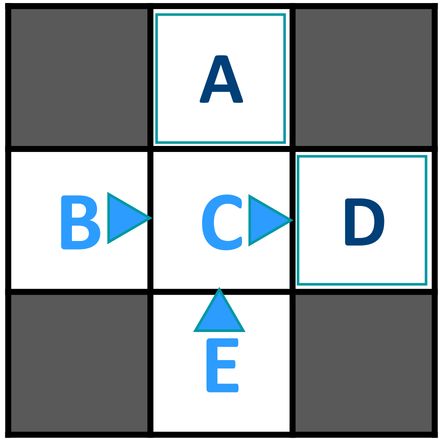
Model-based Reinforcement Learning

Model-based RL, Introduction

- Basic idea:
  - Learning an approximated transition model based on experience
  - Solving the learned MDP
- Step 1: Learning an empirical MDP model
  - Counting the resulting $\boldsymbol{s'}$ for each state action pair $(\boldsymbol{s}, \boldsymbol{a})$
  - Normalization leads to an estimate of the transitional model $\boldsymbol{P^{a}_{ss'}}$
  - Observe the reward $\boldsymbol{R^{a}_{s}}$ for the episode $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s'})$
- Step 2: Solving the learned MDP
  - E.g. Perform value iteration

Model-based RL, Example

In the following GridWorld the transition matrix $P_{ss'}^{a}$ and the reward function R_s^a are to be learned by RL



Assumption: $\gamma = 1$

Final states:

A,D

Strategy π is given

Learned model parameters

$P_{ss'}^{a}$	
$\mathbb{P}[C\|B, East]$	1,00
$\mathbb{P}[D\|C, East]$	0,75
$\mathbb{P}[A\|C, East]$	0,25
…	…

R_s^a	
R_C	-1
R_D	10
R_A	-10
…	…

Observed episodes (training)

Episode 1

s	a	s'	R_s^a
B	East	C	-1
C	East	D	-1
D	Exit		+10

Episode 2

s	a	s'	R_s^a
B	East	C	-1
C	East	D	-1
D	Exit		+10

Episode 3

s	a	s'	R_s^a
E	North	C	-1
C	East	D	-1
D	Exit		+10

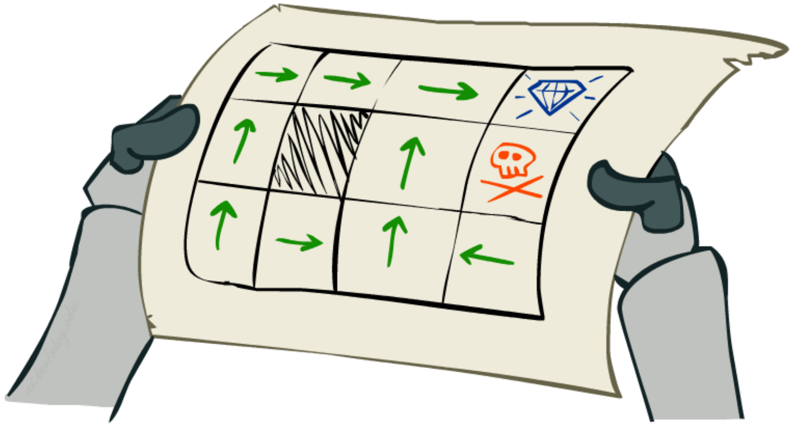
Episode 4

s	a	s'	R_s^a
E	North	C	-1
C	East	A	-1
A	Exit		-10

Model-free Reinforcement Learning

Model-free Passive Reinforcement Learning

- Simplified task: Evaluation of policies
  - Input: A fixed policy $\boldsymbol{\pi(s)}$
  - The transition matrix $\boldsymbol{P^{a}_{ss'}}$ is unknown
  - The reward function $\boldsymbol{R^{a}_{s}}$ is unknown
  - Goal: Learn the state values $\boldsymbol{V(s)}$



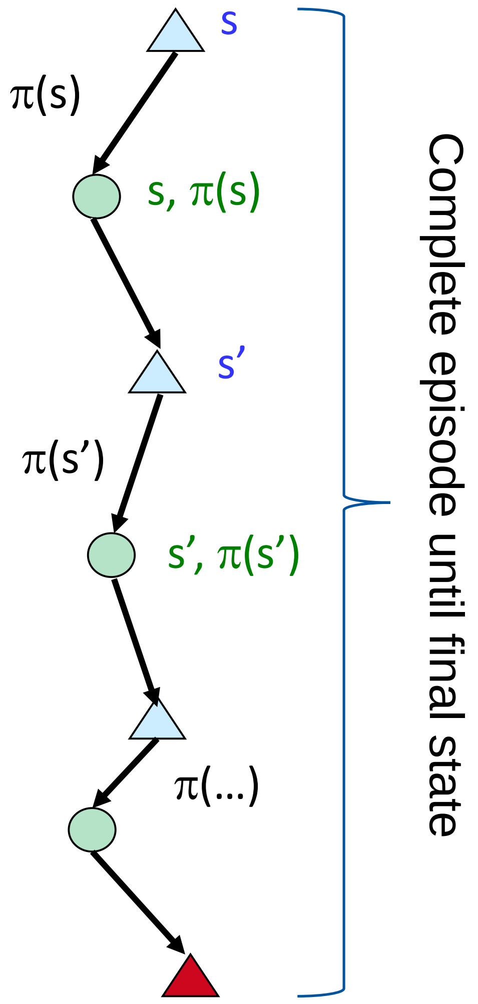
- In this case:
  - The learner runs parallel to the policy execution
  - No choice as to which action to perform
  - Just execute the given strategy and learn from experience

Model-free RL, Monte-Carlo Methods

- **Idea:** estimates values based on averaged sample returns
  - Sample Sequences $(s, a, s', R_{ss'})$ by
    - direct interaction with the environment
    - or in simulation
  - Experience is divided into episodes
  - **Episodic Learning:** Update only after completion of an episode

- Learning values from **episodes**
  - **Incremental –** i.e. episode-by-episode (not step-by-step)
  - **Prediction problem**: Calculation of v_π and q_π for fixed, arbitrary strategy π
  - **Strategy improvement**
  - Ideas from Dynamic Programming

- Methods:
  - **First-visit MC policy evaluation**
  - **Every-visit MC policy evaluation**



Model-free RL, Monte-Carlo Methods - First-visit MC policy evaluation

First-visit MC policy evaluation Algorithm

Initialize

$\pi \leftarrow policy\ to\ be\ evaluated$

$V \leftarrow arbitrary\ state\ value\ function$

$Returns(s), \leftarrow an\ empty\ list, \forall s \in S$

Repeat (for$ever$):

Generate an episode using π

For each state s appearing in the episode:

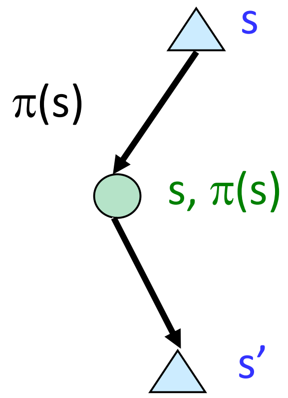
G ← return following the first occurrence of s

Add G to $Returns(s)$

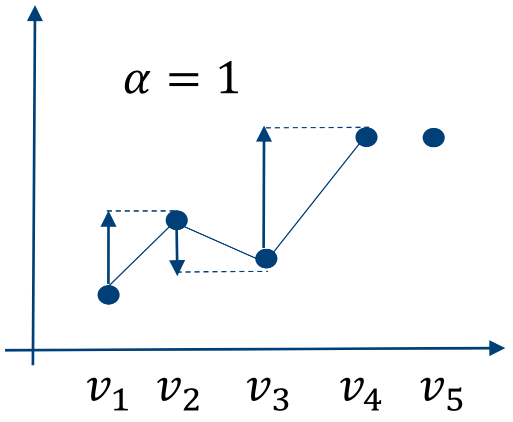
$\mathrm{V}(s) \leftarrow average(Returns(s))$

Model-free RL, Learning from temporal difference (TD)

- **Idea:** Learning from every experience made!
  - Update V(s) directly if a transition $(s, a, s', R_{ss'})$ was executed
  - Since no larger amount of samples is required, the update takes place more frequently
  - **Model-free learning:** No knowledge of MDP transitions / rewards



- Learning values (episodes) from **temporal difference**
  - **Policy** is fixed, **evaluation** will continue
  - Move the values in the direction of the successor state



- Update rule

$$\mathrm{V}(s) \leftarrow \mathrm{V}(s) + \alpha[R_{ss'} + \gamma V(s') - V(s)]$$

$$\alpha \coloneqq Learning\ rate\ parameters$$

$$R_{ss'} + \gamma V(s') \coloneqq \boldsymbol{TD\ Goal}$$

$$R_{ss'} + \gamma V(s') - V(s) \coloneqq \boldsymbol{TD\ Error}$$

Model-free RL, Learning from temporal difference (TD)

TD-Learning Algorithm

Input: the policy π to be evaluated

Initialize $V(s), \forall s \in S$ *arbitrarily*

Repeat (for each episode e):

Initialize s

Repeat (for each step t of the episode):

$a \leftarrow$ given action by π for s

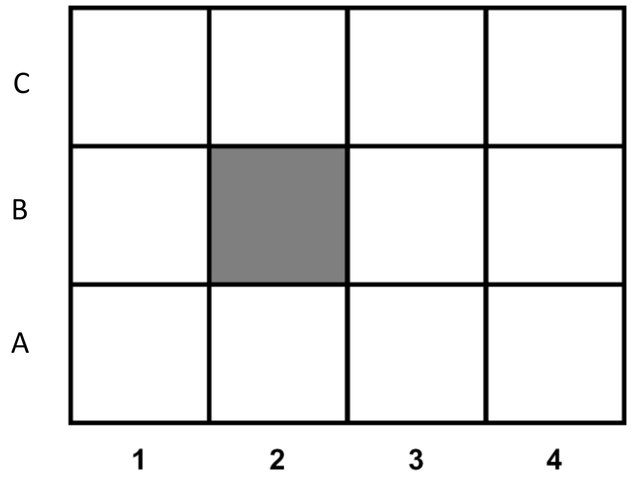
take action a ; observe reward $R_{ss'}$ and next state s'

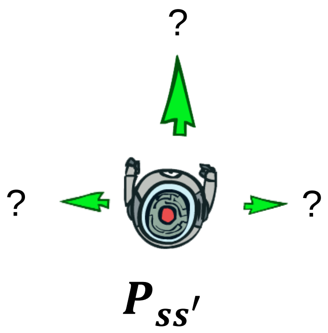
$\mathrm{V}(s) \leftarrow \mathrm{V}(s) + \alpha[R_{ss'} + \gamma V(s') - V(s)]$

$\mathrm{s} \leftarrow s'$;

Until s is final state

Model-free RL, Learning from temporal difference (TD), Example





Initial state A_1

$\pi = S \rightarrow A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

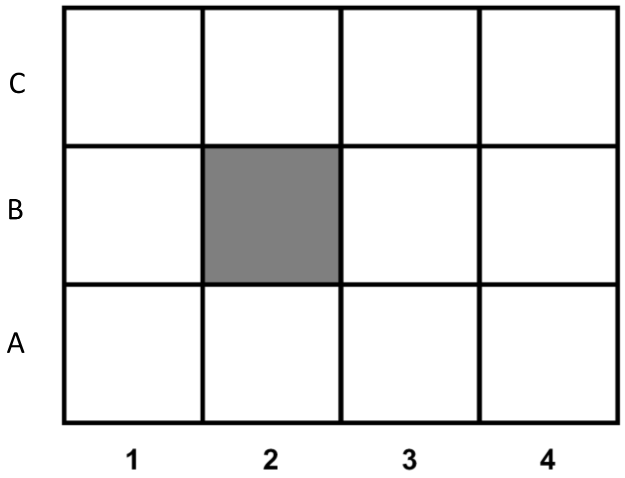
$R_{ss'} = ?$

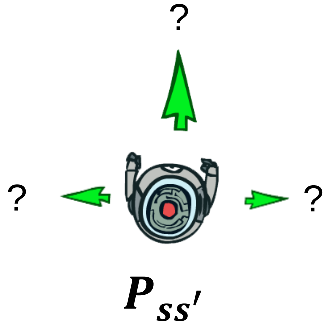
Action executed $\boldsymbol{a}$	Observed state $\boldsymbol{s'}$	Observed reward $R_{ss'}$

$\mathbf{V}(\boldsymbol{s}) \leftarrow \mathbf{V}(\boldsymbol{s}) + \boldsymbol{\alpha}[\boldsymbol{R}_{\boldsymbol{ss'}} + \boldsymbol{\gamma V}(\boldsymbol{s'}) - \boldsymbol{V}(\boldsymbol{s})]$	$\mathbf{V}(\boldsymbol{s})$

$\mathbf{V}(\boldsymbol{A_1})$	$\mathbf{V}(\boldsymbol{A_2})$	$\mathbf{V}(\boldsymbol{A_3})$	$\mathbf{V}(\boldsymbol{A_4})$	$\mathbf{V}(\boldsymbol{B_1})$	$\mathbf{V}(\boldsymbol{B_3})$	$\mathbf{V}(\boldsymbol{B_4})$	$\mathbf{V}(\boldsymbol{C_1})$	$\mathbf{V}(\boldsymbol{C_2})$	$\mathbf{V}(\boldsymbol{C_3})$	$\mathbf{V}(\boldsymbol{C_4})$

Model-free RL, Learning from temporal difference (TD), Example





Initial state A_1

$\pi = S \rightarrow A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

$R_{ss'} = ?$

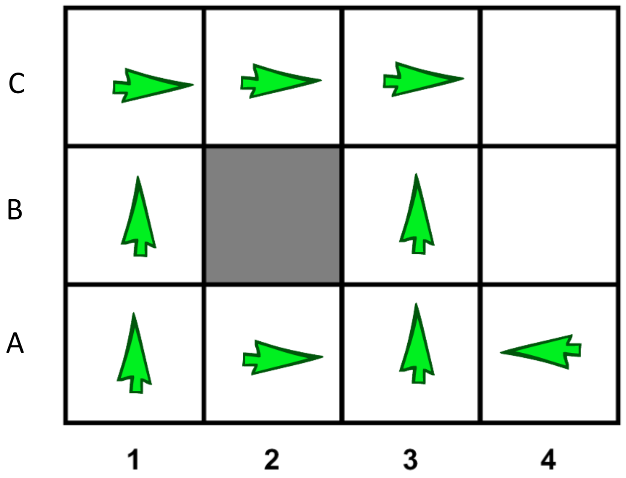
1. Initialize $\boldsymbol{V(s), \forall s \in S}$, **arbitrarily**

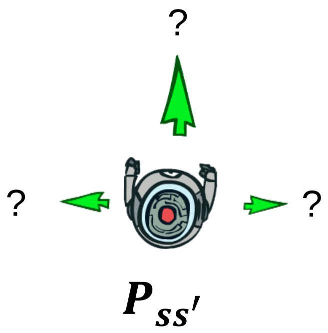
Action executed $\boldsymbol{a}$	Observed state $\boldsymbol{s'}$	Observed reward $R_{ss'}$

$\mathbf{V}(\boldsymbol{s}) \leftarrow \mathbf{V}(\boldsymbol{s}) + \boldsymbol{\alpha}[\boldsymbol{R_{ss'}} + \boldsymbol{\gamma V(s')} - \boldsymbol{V(s)}]$	$\mathbf{V}(\boldsymbol{s})$

$\mathbf{V}(\boldsymbol{A_1})$	$\mathbf{V}(\boldsymbol{A_2})$	$\mathbf{V}(\boldsymbol{A_3})$	$\mathbf{V}(\boldsymbol{A_4})$	$\mathbf{V}(\boldsymbol{B_1})$	$\mathbf{V}(\boldsymbol{B_3})$	$\mathbf{V}(\boldsymbol{B_4})$	$\mathbf{V}(\boldsymbol{C_1})$	$\mathbf{V}(\boldsymbol{C_2})$	$\mathbf{V}(\boldsymbol{C_3})$	$\mathbf{V}(\boldsymbol{C_4})$
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Model-free RL, Learning from temporal difference (TD), Example





Initial state A_1

$\pi = S \rightarrow A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

$R_{ss'} = ?$

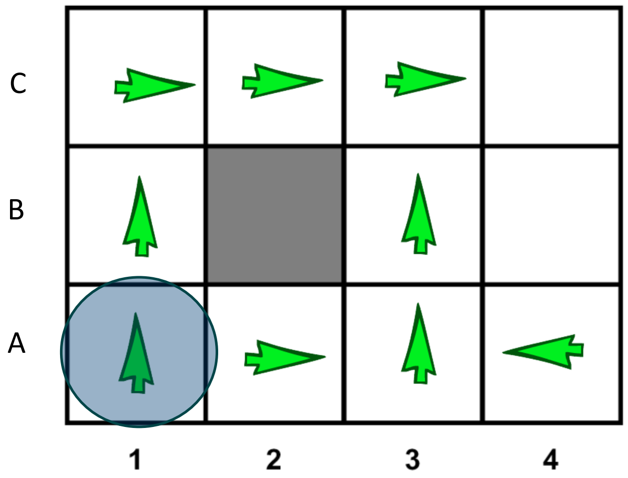
2. Set $\boldsymbol{\pi}$ for the strategy to be evaluated

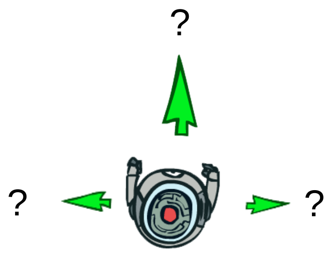
Action executed $\boldsymbol{a}$	Observed state $\boldsymbol{s'}$	Observed reward $R_{ss'}$

$\mathbf{V}(\boldsymbol{s}) \leftarrow \mathbf{V}(\boldsymbol{s}) + \boldsymbol{\alpha}[\boldsymbol{R_{ss'}} + \boldsymbol{\gamma V}(\boldsymbol{s'}) - \boldsymbol{V}(\boldsymbol{s})]$	$\mathbf{V}(\boldsymbol{s})$

$\mathbf{V}(\boldsymbol{A_1})$	$\mathbf{V}(\boldsymbol{A_2})$	$\mathbf{V}(\boldsymbol{A_3})$	$\mathbf{V}(\boldsymbol{A_4})$	$\mathbf{V}(\boldsymbol{B_1})$	$\mathbf{V}(\boldsymbol{B_3})$	$\mathbf{V}(\boldsymbol{B_4})$	$\mathbf{V}(\boldsymbol{C_1})$	$\mathbf{V}(\boldsymbol{C_2})$	$\mathbf{V}(\boldsymbol{C_3})$	$\mathbf{V}(\boldsymbol{C_4})$
0	0	0	0	0	0	0	0	0	0	0

Model-free RL, Learning from temporal difference (TD), Example





$\boldsymbol{P_{ss'}}$

3. Initialize $\boldsymbol{s}$

Initial state A_1

$\pi = S \rightarrow A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

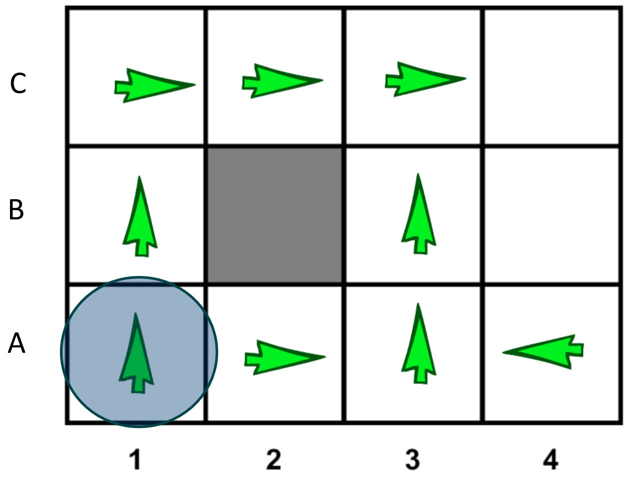
$R_{ss'} = ?$

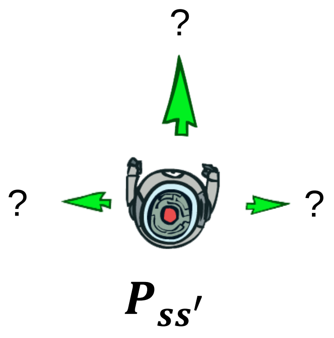
Action executed $\boldsymbol{a}$	Observed state $\boldsymbol{s'}$	Observed reward $R_{ss'}$

$\mathbf{V}(\boldsymbol{s}) \leftarrow \mathbf{V}(\boldsymbol{s}) + \boldsymbol{\alpha}[\boldsymbol{R_{ss'}} + \boldsymbol{\gamma V}(\boldsymbol{s'}) - \boldsymbol{V}(\boldsymbol{s})]$	$\mathbf{V}(\boldsymbol{s})$

$\mathbf{V}(\boldsymbol{A_1})$	$\mathbf{V}(\boldsymbol{A_2})$	$\mathbf{V}(\boldsymbol{A_3})$	$\mathbf{V}(\boldsymbol{A_4})$	$\mathbf{V}(\boldsymbol{B_1})$	$\mathbf{V}(\boldsymbol{B_3})$	$\mathbf{V}(\boldsymbol{B_4})$	$\mathbf{V}(\boldsymbol{C_1})$	$\mathbf{V}(\boldsymbol{C_2})$	$\mathbf{V}(\boldsymbol{C_3})$	$\mathbf{V}(\boldsymbol{C_4})$
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Model-free RL, Learning from temporal difference (TD), Example





Initial state A_1

$\pi = S \rightarrow A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

$R_{ss'} = ?$

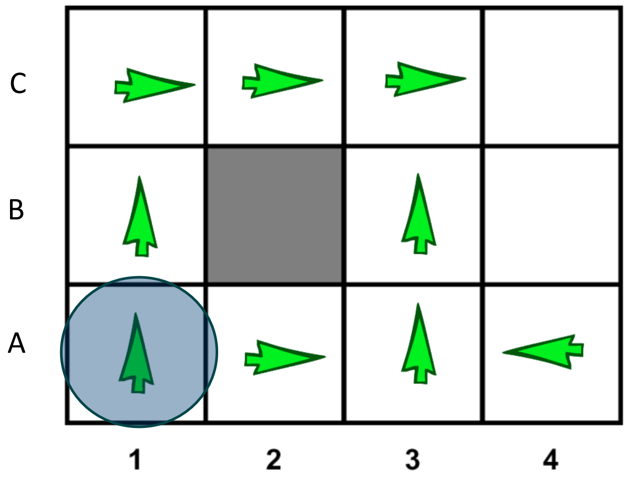
4. $a \leftarrow$ given action π for s

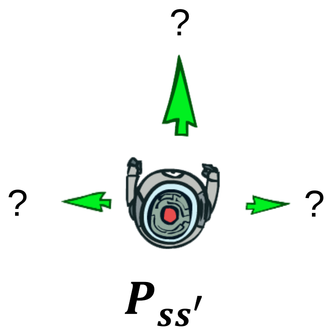
Action executed a	Observed state s'	Observed reward $R_{ss'}$
a_{South}		

$V(s) \leftarrow V(s) + \alpha[R_{ss'} + \gamma V(s') - V(s)]$	$V(s)$

$V(A_1)$	$V(A_2)$	$V(A_3)$	$V(A_4)$	$V(B_1)$	$V(B_3)$	$V(B_4)$	$V(C_1)$	$V(C_2)$	$V(C_3)$	$V(C_4)$
0	0	0	0	0	0	0	0	0	0	0

Model-free RL, Learning from temporal difference (TD), Example





Initial state A_1

$\pi = S \rightarrow A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

$R_{ss'} = ?$

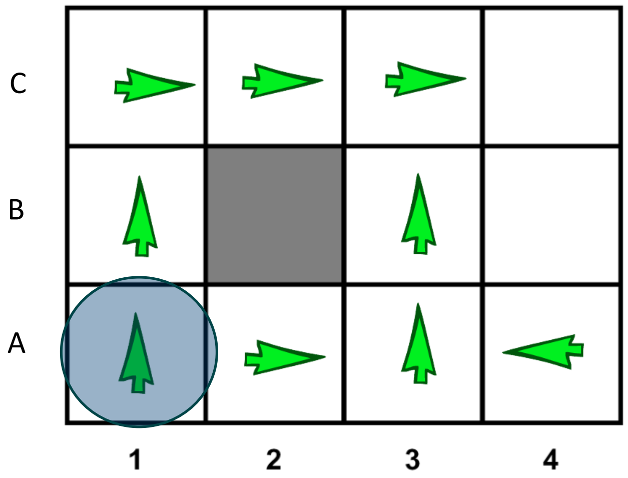
5. Execute $\boldsymbol{a}$; $\boldsymbol{observe\ reward\ R_{ss'}}$ $\boldsymbol{and\ next\ state\ s'}$

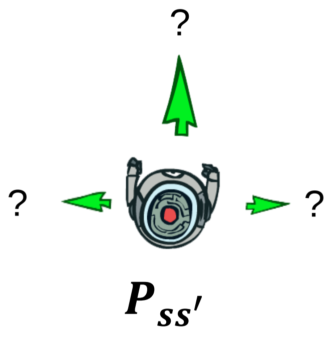
Action executed $\boldsymbol{a}$	Observed state $\boldsymbol{s'}$	Observed reward $R_{ss'}$
a_{North}	B_1	$-0{,}05$

$\mathbf{V}(\boldsymbol{s}) \leftarrow \mathbf{V}(\boldsymbol{s}) + \boldsymbol{\alpha}[\boldsymbol{R_{ss'}} + \boldsymbol{\gamma V}(\boldsymbol{s'}) - \boldsymbol{V}(\boldsymbol{s})]$	$\mathbf{V}(\boldsymbol{s})$

$\mathbf{V}(A_1)$	$\mathbf{V}(A_2)$	$\mathbf{V}(A_3)$	$\mathbf{V}(A_4)$	$\mathbf{V}(B_1)$	$\mathbf{V}(B_3)$	$\mathbf{V}(B_4)$	$\mathbf{V}(C_1)$	$\mathbf{V}(C_2)$	$\mathbf{V}(C_3)$	$\mathbf{V}(C_4)$
0	0	0	0	0	0	0	0	0	0	0

Model-free RL, Learning from temporal difference (TD), Example





Initial state A_1

$\pi = S \rightarrow A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

$R_{ss'} = ?$

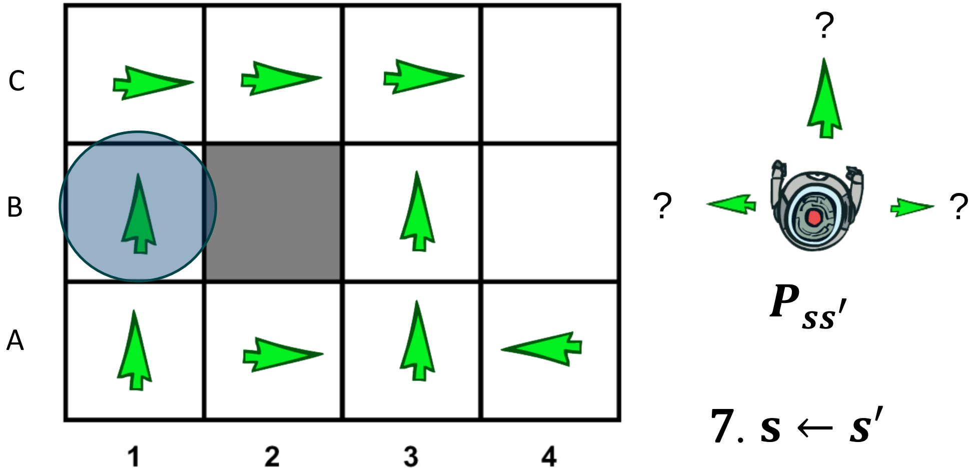
6. $\mathbf{V}(s) \leftarrow \mathbf{V}(s) + \alpha[R_{ss'} + \gamma V(s') - V(s)]$

Action executed a	Observed state s'	Observed reward $R_{ss'}$
a_{North}	B_1	$-0{,}05$

$\mathbf{V}(s) \leftarrow \mathbf{V}(s) + \alpha[R_{ss'} + \gamma V(s') - V(s)]$	$\mathbf{V}(s)$
$0 + 0{,}5[-0{,}05 + 0{,}9 * 0 - 0]$	$-0{,}025$

$\mathbf{V}(A_1)$	$\mathbf{V}(A_2)$	$\mathbf{V}(A_3)$	$\mathbf{V}(A_4)$	$\mathbf{V}(B_1)$	$\mathbf{V}(B_3)$	$\mathbf{V}(B_4)$	$\mathbf{V}(C_1)$	$\mathbf{V}(C_2)$	$\mathbf{V}(C_3)$	$\mathbf{V}(C_4)$
$-0{,}025$	0	0	0	0	0	0	0	0	0	0

Model-free RL, Learning from temporal difference (TD), Example



Initial state A_1

$\pi = S \rightarrow A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

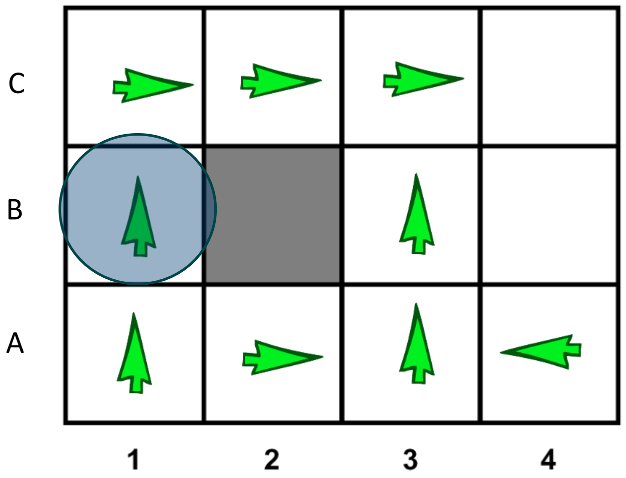
$R_{ss'} = ?$

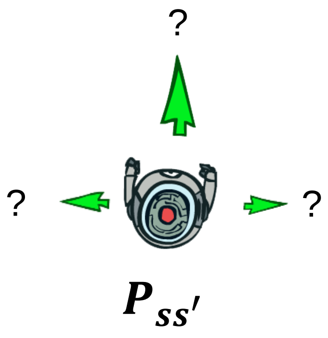
Action executed $\boldsymbol{a}$	Observed state $\boldsymbol{s'}$	Observed reward $R_{ss'}$
a_{North}	B_1	$-0{,}05$

$\mathbf{V}(\boldsymbol{s}) \leftarrow \mathbf{V}(\boldsymbol{s}) + \boldsymbol{\alpha}[\boldsymbol{R_{ss'}} + \boldsymbol{\gamma V}(\boldsymbol{s'}) - \boldsymbol{V}(\boldsymbol{s})]$	$\mathbf{V}(\boldsymbol{s})$
$0 + 0{,}5[-0{,}05 + 0{,}9 * 0 - 0]$	$-0{,}025$

$\mathbf{V}(A_1)$	$\mathbf{V}(A_2)$	$\mathbf{V}(A_3)$	$\mathbf{V}(A_4)$	$\mathbf{V}(B_1)$	$\mathbf{V}(B_3)$	$\mathbf{V}(B_4)$	$\mathbf{V}(C_1)$	$\mathbf{V}(C_2)$	$\mathbf{V}(C_3)$	$\mathbf{V}(C_4)$
$-0{,}025$	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Model-free RL, Learning from temporal difference (TD), Example





Initial state A_1

$\pi = S \rightarrow A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

$R_{ss'} = ?$

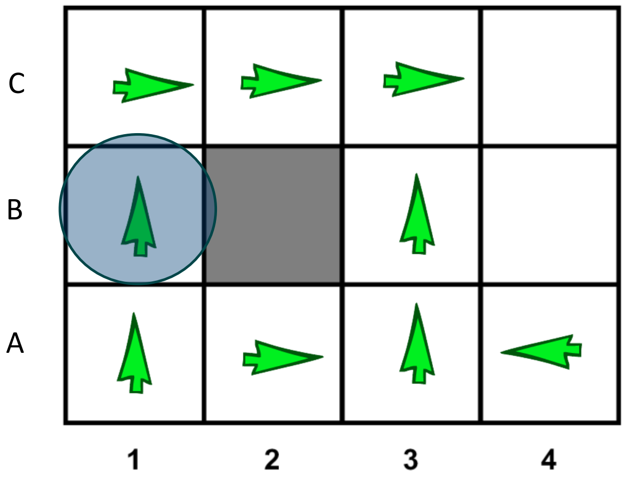
4. $a \leftarrow$ given action by π for s

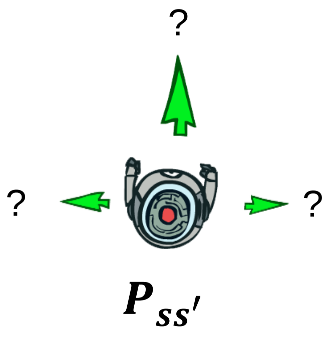
Action executed a	Observed state s'	Observed reward $R_{ss'}$
a_{North}		

$V(s) \leftarrow V(s) + \alpha[R_{ss'} + \gamma V(s') - V(s)]$	$V(s)$

$V(A_1)$	$V(A_2)$	$V(A_3)$	$V(A_4)$	$V(B_1)$	$V(B_3)$	$V(B_4)$	$V(C_1)$	$V(C_2)$	$V(C_3)$	$V(C_4)$
$-0{,}025$	0	0	0	0	0	0	0	0	0	0

Model-free RL, Learning from temporal difference (TD), Example





Initial state A_1

$\pi = S \rightarrow A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

$R_{ss'} = ?$

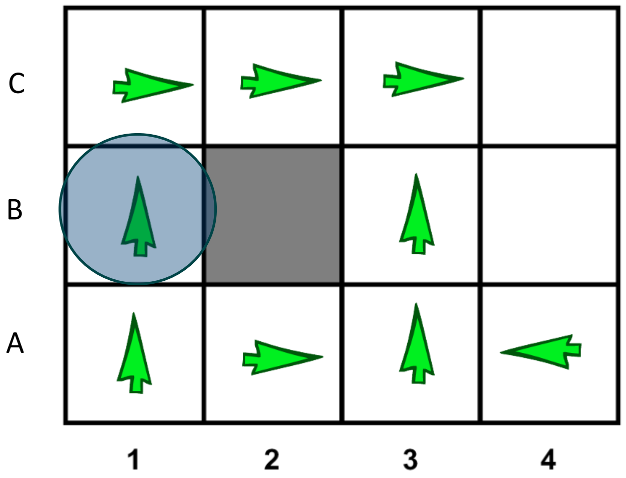
5. Execute $\boldsymbol{a}$; observe reward $\boldsymbol{R_{ss'}}$ *and next state* $\boldsymbol{s'}$

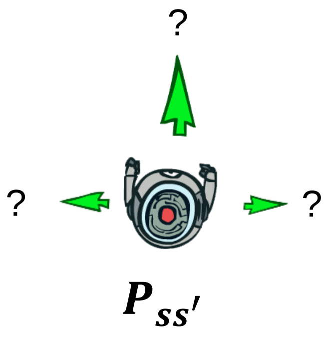
Action executed $\boldsymbol{a}$	Observed state $\boldsymbol{s'}$	Observed reward $R_{ss'}$
a_{North}	B_1	$-0{,}05$

$\mathbf{V}(\boldsymbol{s}) \leftarrow \mathbf{V}(\boldsymbol{s}) + \boldsymbol{\alpha}[\boldsymbol{R_{ss'}} + \boldsymbol{\gamma V}(\boldsymbol{s'}) - \boldsymbol{V}(\boldsymbol{s})]$	$\mathbf{V}(\boldsymbol{s})$

$\mathbf{V}(\boldsymbol{A_1})$	$\mathbf{V}(\boldsymbol{A_2})$	$\mathbf{V}(\boldsymbol{A_3})$	$\mathbf{V}(\boldsymbol{A_4})$	$\mathbf{V}(\boldsymbol{B_1})$	$\mathbf{V}(\boldsymbol{B_3})$	$\mathbf{V}(\boldsymbol{B_4})$	$\mathbf{V}(\boldsymbol{C_1})$	$\mathbf{V}(\boldsymbol{C_2})$	$\mathbf{V}(\boldsymbol{C_3})$	$\mathbf{V}(\boldsymbol{C_4})$
$-0{,}025$	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Model-free RL, Learning from temporal difference (TD), Example





Initial state A_1

$\pi = S \to A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

$R_{ss'} = ?$

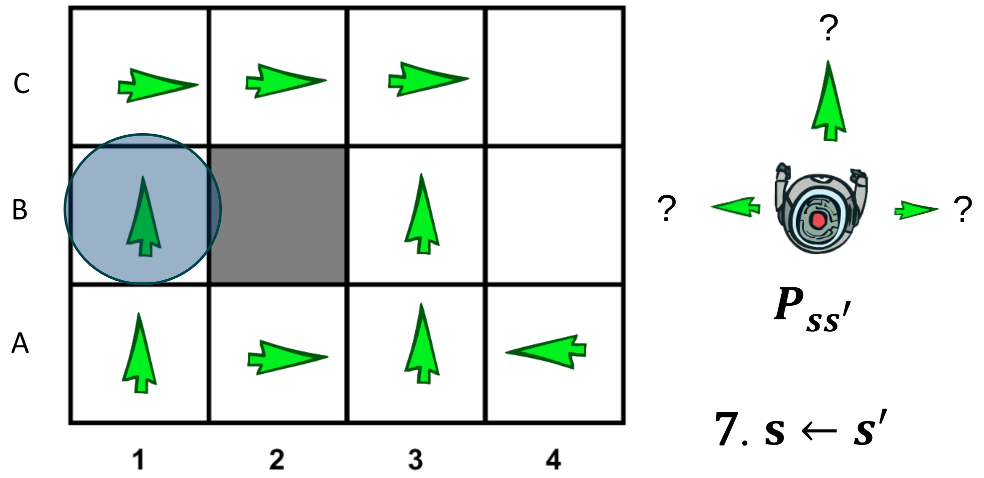
6. $\mathbf{V}(s) \leftarrow \mathbf{V}(s) + \alpha[R_{ss'} + \gamma V(s') - V(s)]$

Action executed a	Observed state s'	Observed reward $R_{ss'}$
a_{North}	B_1	$-0{,}05$

$\mathbf{V}(s) \leftarrow \mathbf{V}(s) + \alpha[R_{ss'} + \gamma V(s') - V(s)]$	$\mathbf{V}(s)$
$0 + 0{,}5[-0{,}05 + 0{,}9 * 0 - 0]$	$-0{,}025$

$V(A_1)$	$V(A_2)$	$V(A_3)$	$V(A_4)$	$V(B_1)$	$V(B_3)$	$V(B_4)$	$V(C_1)$	$V(C_2)$	$V(C_3)$	$V(C_4)$
$-0{,}025$	0	0	0	$-0{,}025$	0	0	0	0	0	0

Model-free RL, Learning from temporal difference (TD), Example



$P_{ss'}$

$7.\ s \leftarrow s'$

Initial state A_1

$\pi = S \rightarrow A$

$\gamma = 0{,}9$

$\alpha = 0{,}5$

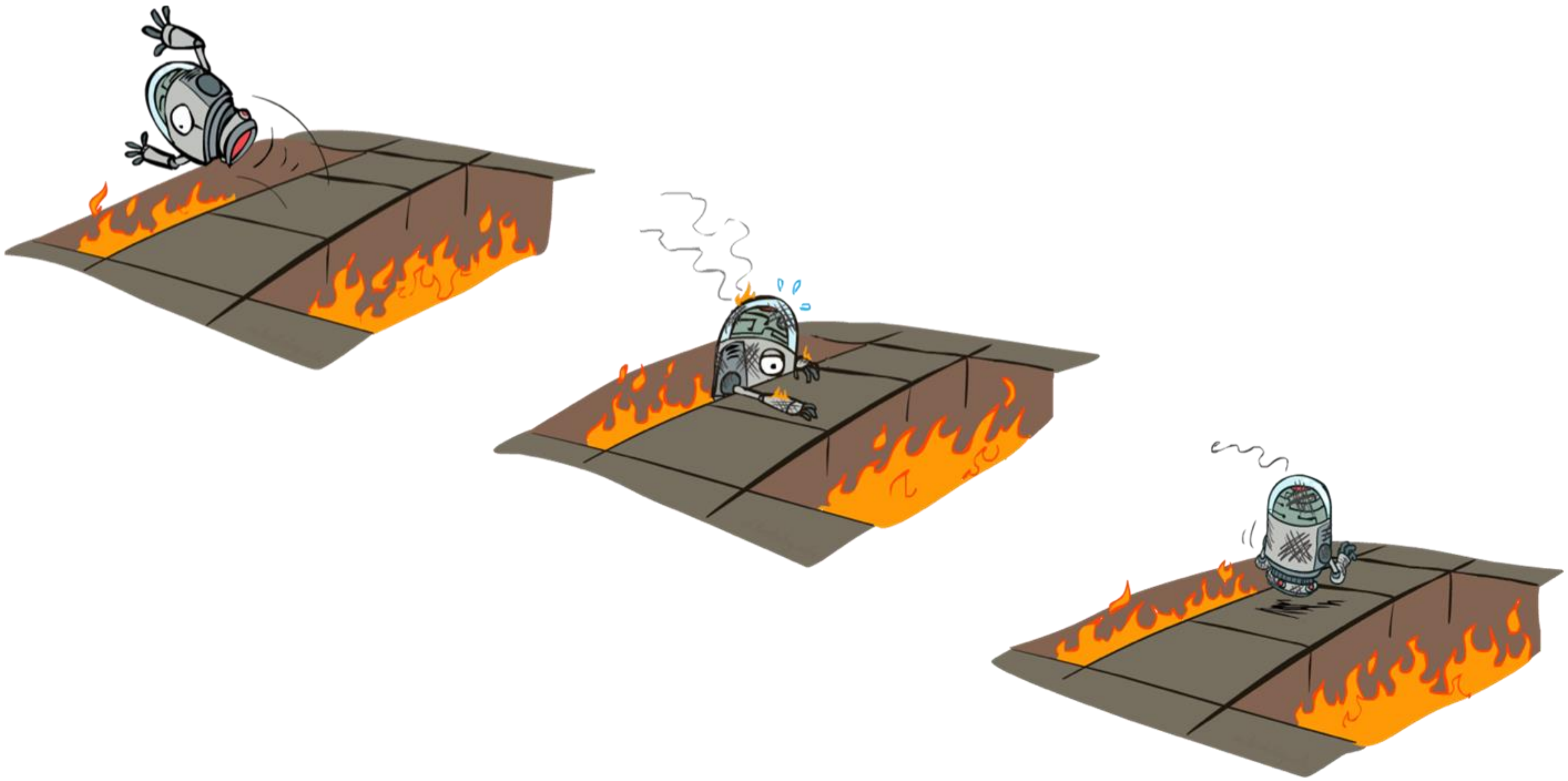
$R_{ss'} = ?$

Action executed a	Observed state s'	Observed reward $R_{ss'}$
a_{North}	B_1	$-0{,}05$

$V(s) \leftarrow V(s) + \alpha[R_{ss'} + \gamma V(s') - V(s)]$	$V(s)$
$0 + 0{,}5[-0{,}05 + 0{,}9 * 0 - 0]$	$-0{,}025$

$V(A_1)$	$V(A_2)$	$V(A_3)$	$V(A_4)$	$V(B_1)$	$V(B_3)$	$V(B_4)$	$V(C_1)$	$V(C_2)$	$V(C_3)$	$V(C_4)$
$-0{,}025$	0	0	0	$-0{,}025$	0	0	0	0	0	0

Model-free Active Reinforcement Learning



Model-free (Active) RL, Q-Learning

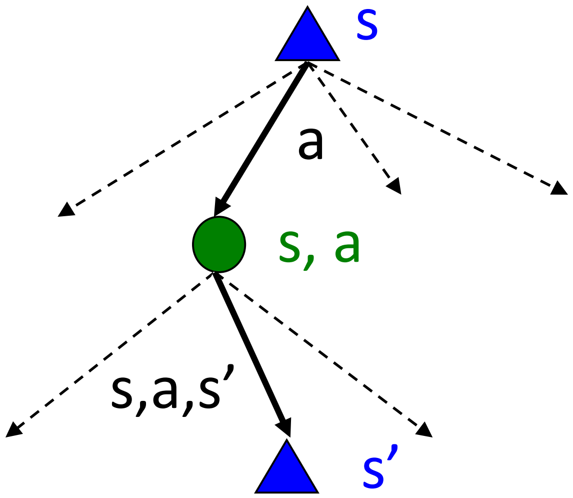
- **Idea:** Learning an action/value representation
  - During TD learning a value representation was learned
  - Neither learning nor action selection requires a model $P_{ss'}^{a}$
  - Therefore, this is a model-free method

- **Procedure**
  - No explicit policy (unlike TD learning) $\Longrightarrow$ active
  - Algorithm always selects the **action with best Q-value**

- Update-Rule

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

$\alpha \coloneqq$ learning rate parameter



Model-free (Active) RL, Q-Learning

Q-Learning Algorithm

Initialize $Q(s,a), \forall s \in S, a \in A(s)$ $arbitrarily$, and $Q(\text{Final State}, \cdot) = 0$

Repeat (for each episode e):

Initialize s

Repeat (for each step t of the episode):

Choose a from s from the policy derived from Q

Take action a and observe $R_{ss'}, s'$

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[R_{ss'} + \gamma \max_a Q(s',a) - Q(s,a)\right]$$

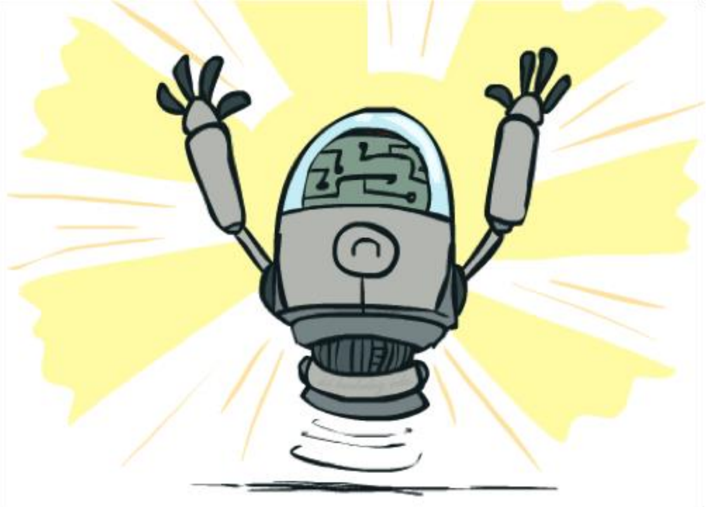
$\mathrm{s} \leftarrow s'$;

Until s is final state

Model-free (Active) RL, Q-Learning

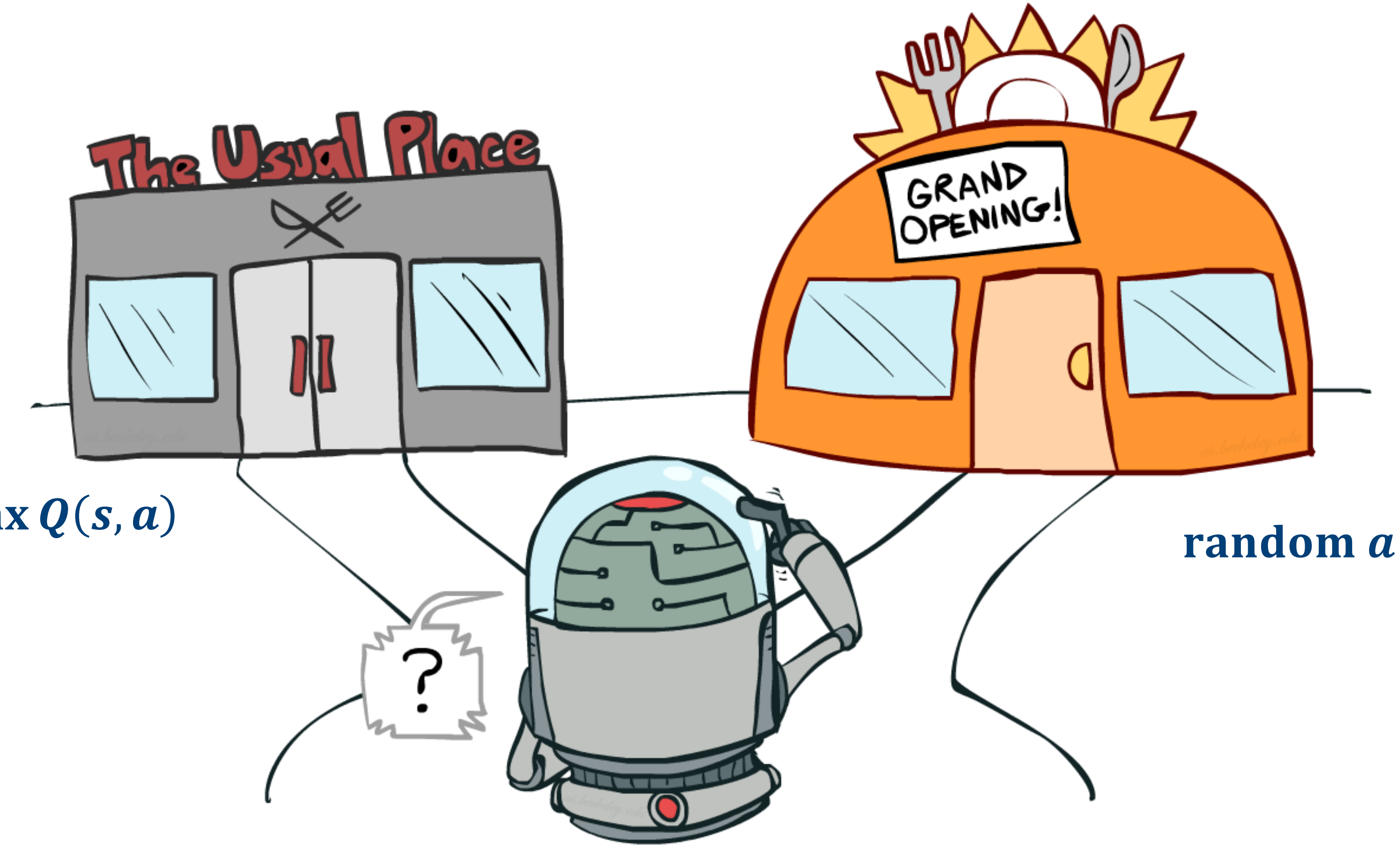
- Impressive results :
  - Q-Learning converges to an optimal strategy - even if the agent behaves suboptimally

- This is called Off-Policy Learning



- Conditional:
  - There has been enough exploration
  - The learning rate parameter may have to decrease over time.
  - ...but this must not happen too quickly.
  - In a nutshell: After enough time, it is not relevant how actions are selected for exploration (!)

Model-free (Active) RL, Exploration vs. Exploitation



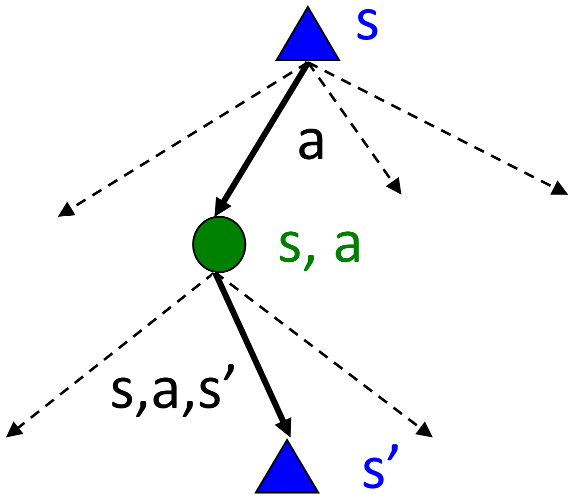
$\mathbf{argmax}_{a}\, Q(s,a)$

random a

Model-free (Active) RL, Exploration vs. Exploitation

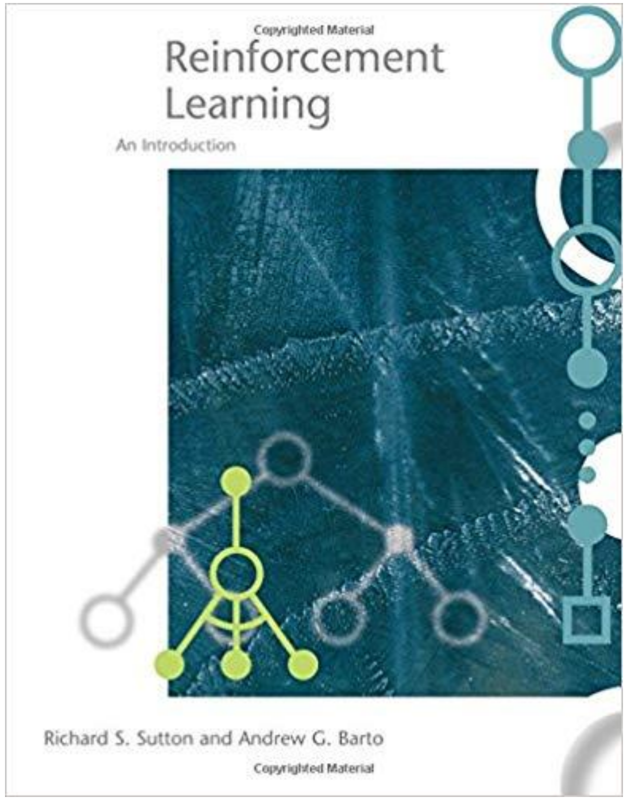
- Easiest: Random actions (ε-greedy)
  - In every time step, toss a coin
  - With (little) probability ε : choose a random action
  - With (high) probability 1- ε : act on the basis of the current strategy

- Problem with ε-greedy
  - Even though the entire state / action space is explored, the agent may still behave suboptimally afterwards
  - A Solution: decrease ε over time



Further Reading Material

- Berkeley Course – Artificial Intelligence (https://inst.eecs.berkeley.edu/~cs188/fa11/lectures.html)
- David Silver – Reinforcement Learning (http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html)



Thank you for your attention!

Lecture Team AIDAE
aidae@ima-ifu.rwth-aachen.de