Reinforcement Learning Lab

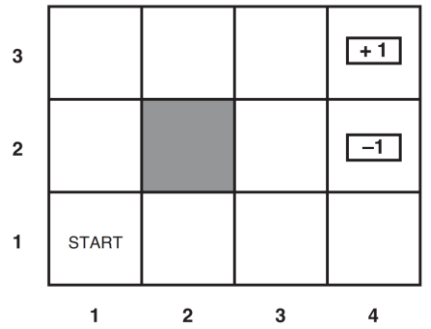
Samuele Bortolotti, Steve Azzolin

Reinforcement Learning

- Teach an agent to perform (optimal) actions in an (unknown) environment
- The agent should learn just from experience

Reinforcement Learning

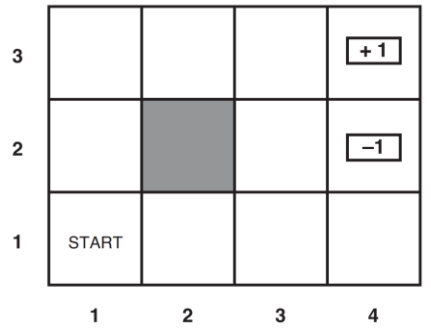
- Teach an agent to perform (optimal) actions in an (unknown) environment
- The agent should learn just from experience



Reinforcement Learning

- Teach an agent to perform (optimal) actions in an (unknown) environment
- The agent should learn just from experience

A *state* is a description of the status of the agent in the environment.

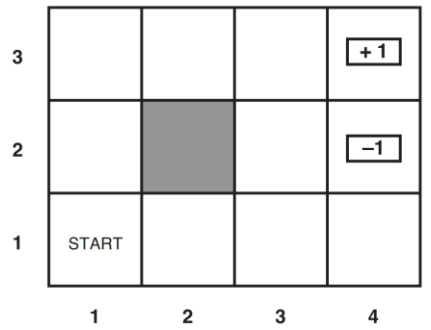


Reinforcement Learning

- Teach an agent to perform (optimal) actions in an (unknown) environment
- The agent should learn just from experience

A *state* is a description of the status of the agent in the environment.

An *agent* can be seen as a function from states to actions π: $S \rightarrow A$

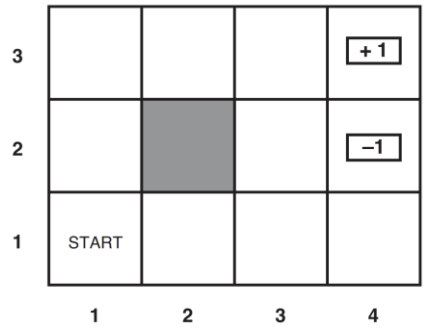


Reinforcement Learning

- Teach an agent to perform (*optimal*) actions in an (*unknown*) environment
- The agent should learn just from experience

$$\pi(s) = argmax_a \sum_{s' \in S} p(s' \mid s, a)\, V(s')$$

State s	V(s)
(1,1)	?
(1,2)	?
(2,1)	?
...	?
(3,4)	+1
(4,3)	-1



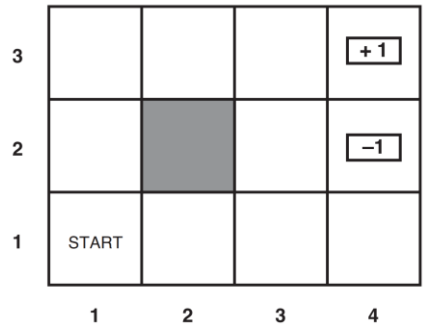
Reinforcement Learning

- Teach an agent to perform (*optimal*) actions in an (*unknown*) environment
- The agent should learn just from experience

$$\pi(s) = argmax_a \sum_{s' \in S} p(s' \mid s, a)\, V(s')$$

Value iteration
Policy iteration
Temporal Difference

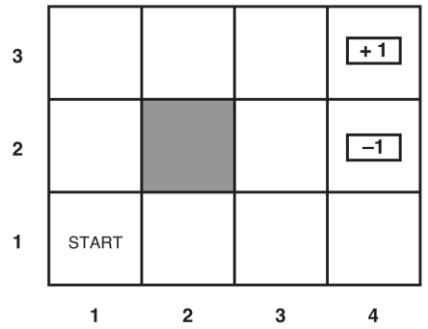
State s	V(s)
(1,1)	?
(1,2)	?
(2,1)	?
...	?
(3,4)	+1
(4,3)	-1



Reinforcement Learning

- Teach an agent to perform (*optimal*) actions in an (*unknown*) environment
- The agent should learn just from experience

State s	Action a	Q(s,a)
(1,1)	Up	?
(1,1)	Right	?
(2,1)	Left	?
(2,1)	Right	?
…		?



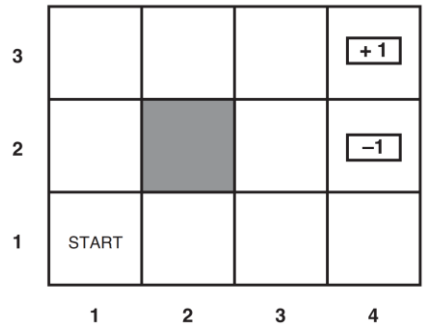
Reinforcement Learning

- Teach an agent to perform (*optimal*) actions in an (*unknown*) environment
- The agent should learn just from experience

$$\pi(s) = argmax_a Q(s, a)$$

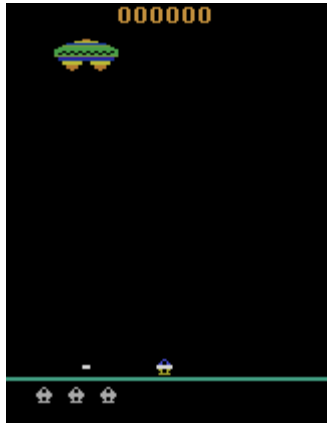
State s	Action a	Q(s,a)
(1,1)	Up	?
(1,1)	Right	?
(2,1)	Left	?
(2,1)	Right	?
…		?

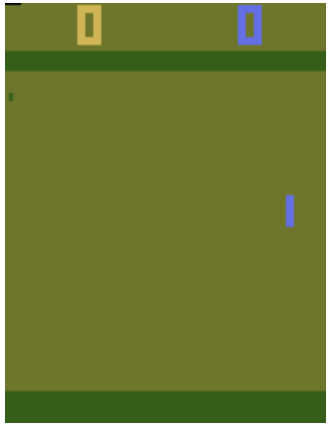
SARSA
Q-learning
Deep Q-learning

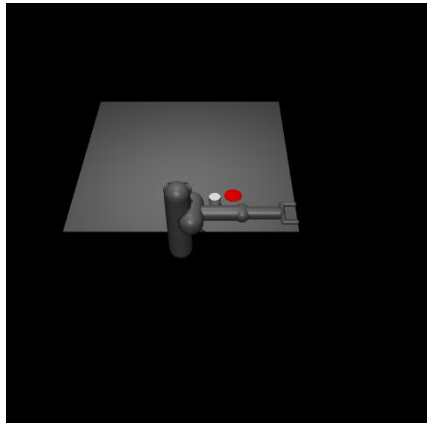


Reinforcement Learning

- The tabular representation of Q(s,a) works only for simple settings

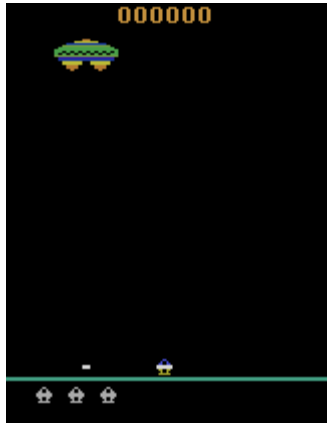


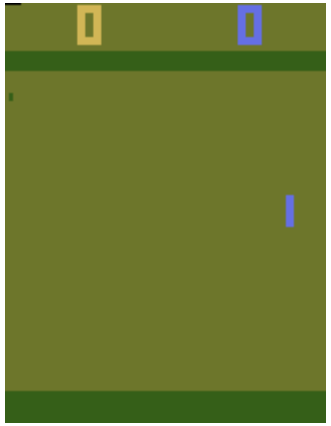


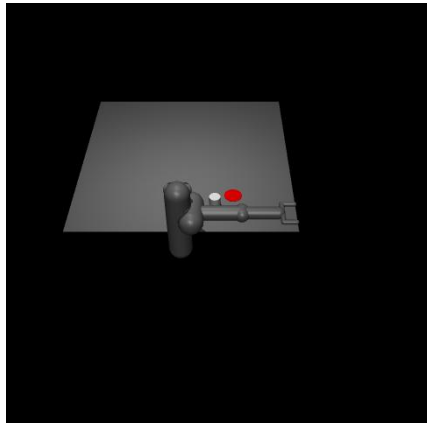


Reinforcement Learning

- The tabular representation of Q(s,a) works only for simple settings



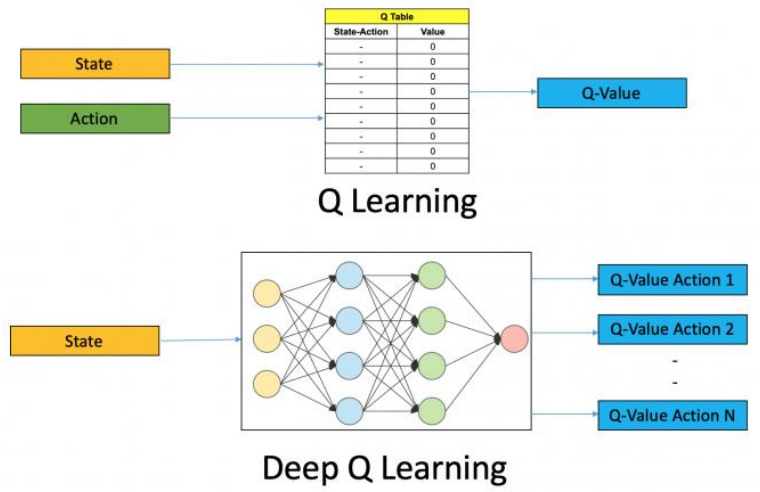




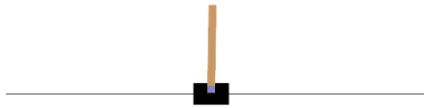
Implement π* with a Neural Network

* *Q(s,a)*

Reinforcement Learning



Code code code code code



Course Information

Instructor: Andrea Passerini
Email: passerini@disi.unitn.it

Teaching assistant: Steve Azzolin
Email: steve.azzolin@unitn.it
Samuele Bortolotti
Email: samuele.bortolotti@unitn.it

Office hours: Arrange by email

Lecture time: Monday 8:30-10:30 (room b109)
Wednesday 11:30-13:30 (room b109)

Communications: Please check the moodle page of the course for news and updates.

Bibliography: R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification (2nd edition)*, Wiley-Interscience, 2001.
D. Koller and N. Friedman, *Probabilistic Graphical Models*, The MIT Press, 2009
J.Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.
I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, The MIT Press, 2016 (online version available here).
S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, 4th Global ed.*, Pearson, 2021.
K. Murphy, *Probabilistic Machine Learning: An Introduction*, The MIT Press, 2021 (online version available here).

Slides: Introduction [slides] [handouts]
Decision Trees [slides] [handouts]
K-nearest neighbours [slides] [handouts]
Linear algebra [slides] [handouts]
Probability theory [slides] [handouts]
Evaluation [slides] [handouts]
Parameter estimation [slides] [handouts]
Bayesian Networks [slides] [handouts]
Learning BN [slides] [handouts]
Naive Bayes [slides] [handouts]
Bayesian Network lab [slides] [data] [software]
Linear discriminant functions [slides] [handouts]
Support Vector Machines [slides] [handouts]
Kernel Machines [slides] [handouts]
Scikit-learn lab [slides] [repository]
Artificial Neural Networks [slides] [handouts]
Artificial Neural Networks lab [slides] [repository]
Ensemble Methods [slides] [handouts]
Unsupervised learning [slides] [handouts]
Unsupervised learning Lab [repository]
Reinforcement learning [slides] [handouts]
Reinforcement learning Lab [repository]

Videos: Registered lectures (from previous year) made available on Moodle