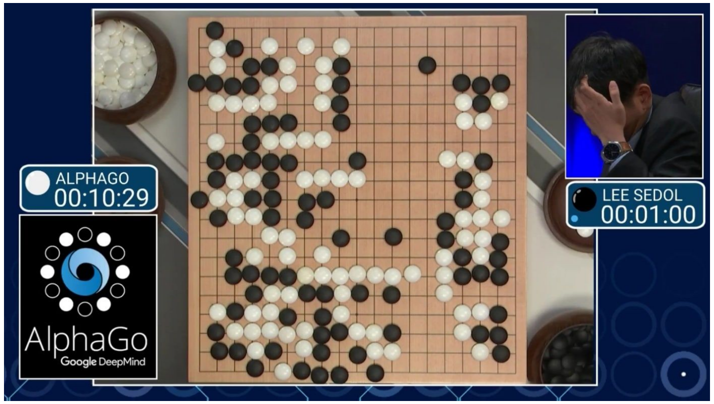
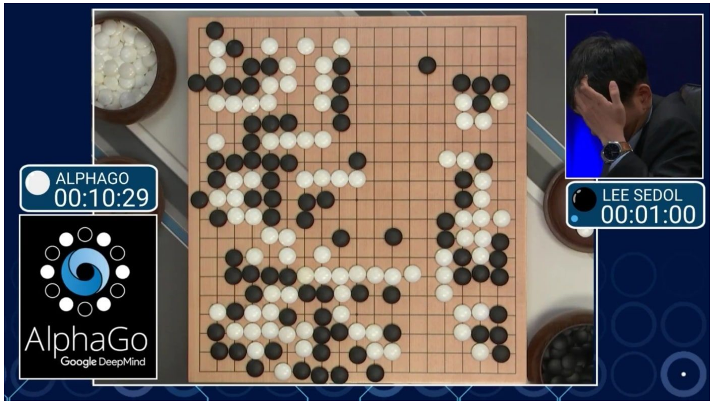
A Brief Introduction To

Reinforcement Learning

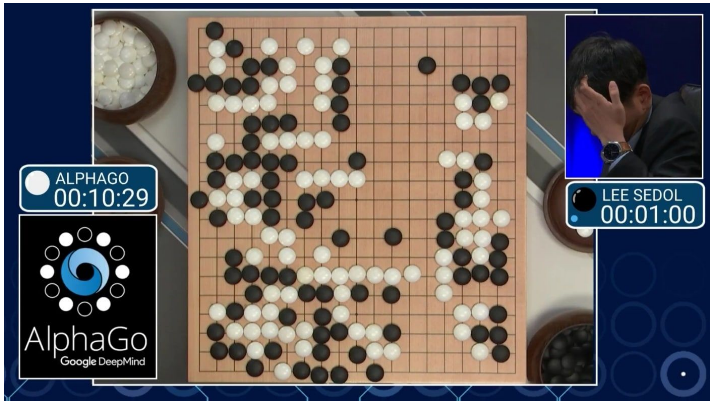
Thomas Moerland

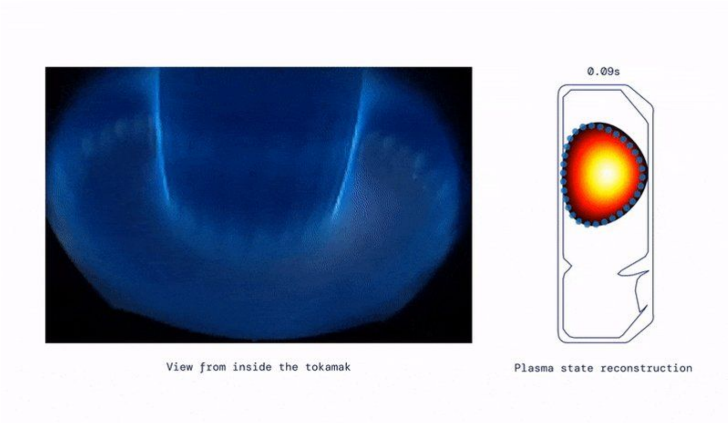

ALPHAGO
00:10:29
AlphaGo
Google DeepMind
LEE SEDOL
00:01:00


The international journal of science / 6 October 2022
nature
MATRIX GAMES
Deep reinforcement learning opens route to faster algorithms for matrix multiplication
Protecting Peru
Can technology help Indigenous groups preserve the Amazon?
Invisible touch
How marine clouds are affected by aerosols emitted from shipping
Preferential practice
US universities favour prestige in faculty hiring and retention

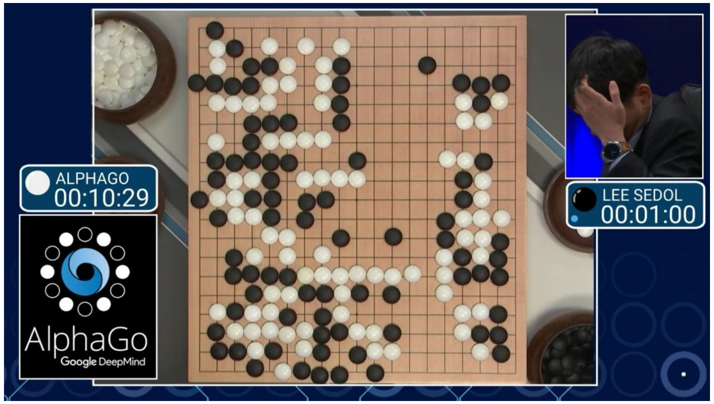

ALPHAGO
00:10:29
AlphaGo
Google DeepMind
LEE SEDOL
00:01:00

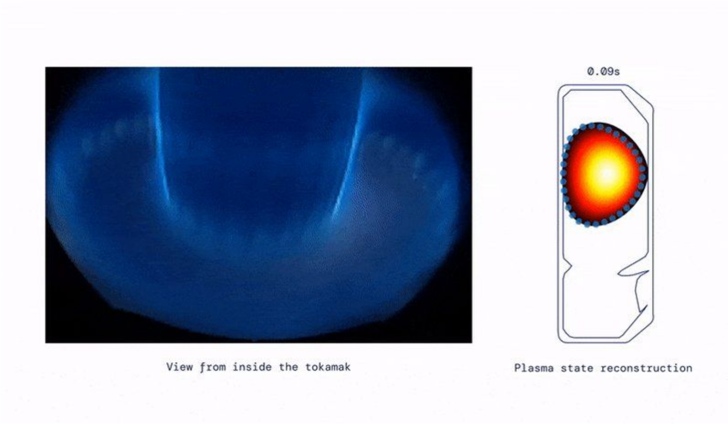

The international journal of science / 6 October 2022
nature
MATRIX GAMES
Deep reinforcement learning opens route to faster algorithms for matrix multiplication
Protecting Peru
Can technology help Indigenous groups preserve the Amazon?
Invisible touch
How marine clouds are affected by aerosols emitted from shipping
Preferential practice
US universities favour prestige in faculty hiring and retention

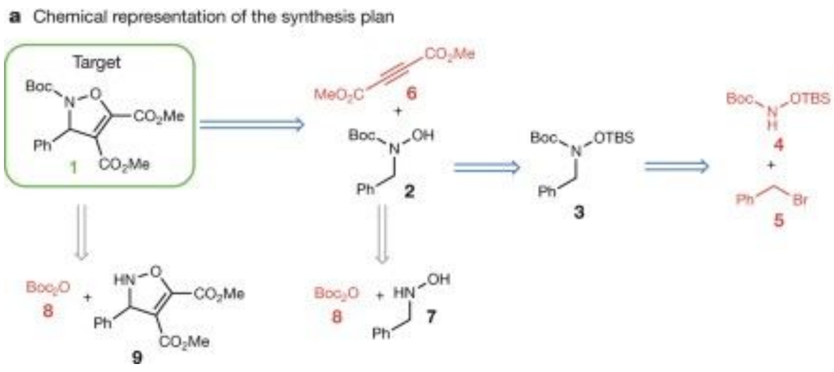

ALPHAGO
00:10:29
AlphaGo
Google DeepMind
LEE SEDOL
00:01:00


0.09s
View from inside the tokamak
Plasma state reconstruction

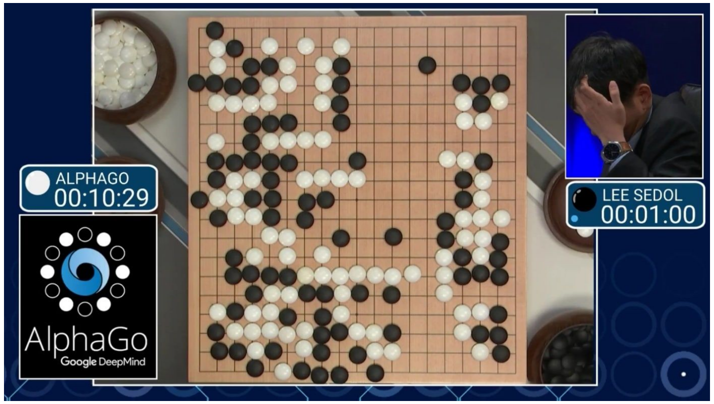

The international journal of science / 6 October 2022
nature
MATRIX GAMES
Deep reinforcement learning opens route to faster algorithms for matrix multiplication
Protecting Peru
Can technology help Indigenous groups preserve the Amazon?
Invisible touch
How marine clouds are affected by aerosols emitted from shipping
Preferential practice
US universities favour prestige in faculty hiring and retention

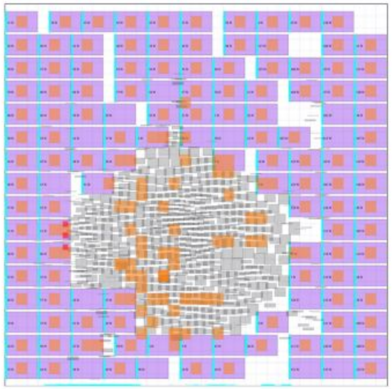

ALPHAGO
00:10:29
AlphaGo
Google DeepMind
LEE SEDOL
00:01:00

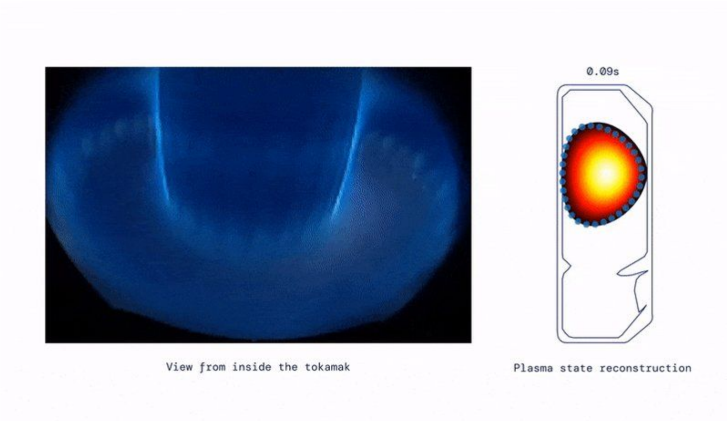

0.09s
View from inside the tokamak
Plasma state reconstruction

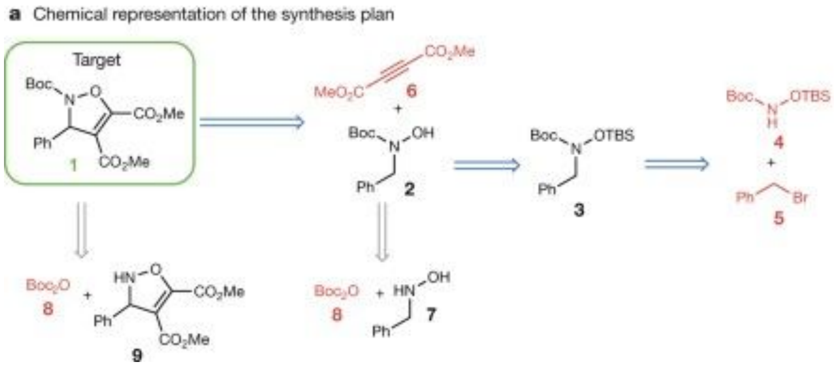

a Chemical representation of the synthesis plan
Target
Boc
CO₂Me
Ph
1
MeO₂C
6
Boc
OH
2
OTBS
3
4
Br
5
Boc₂O
8
HN
9
7

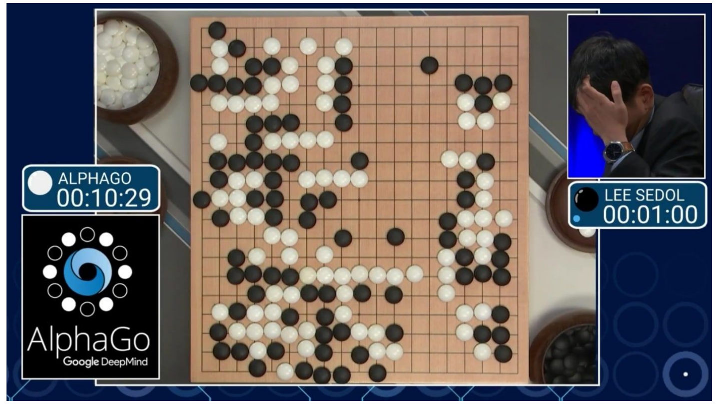

The international journal of science / 6 October 2022
nature
MATRIX GAMES
Deep reinforcement learning opens route to faster algorithms for matrix multiplication
Protecting Peru
Can technology help Indigenous groups preserve the Amazon?
Invisible touch
How marine clouds are affected by aerosols emitted from shipping
Preferential practice
US universities favour prestige in faculty hiring and retention

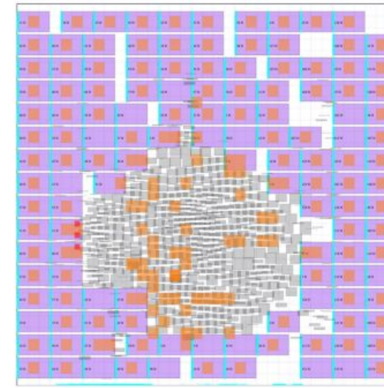

ALPHAGO
00:10:29
AlphaGo
Google DeepMind
LEE SEDOL
00:01:00

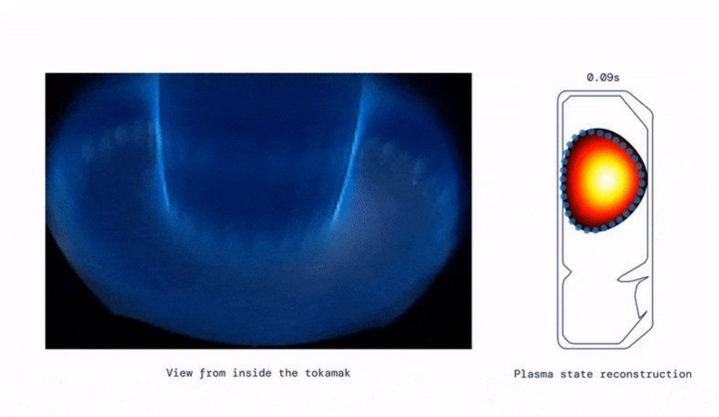


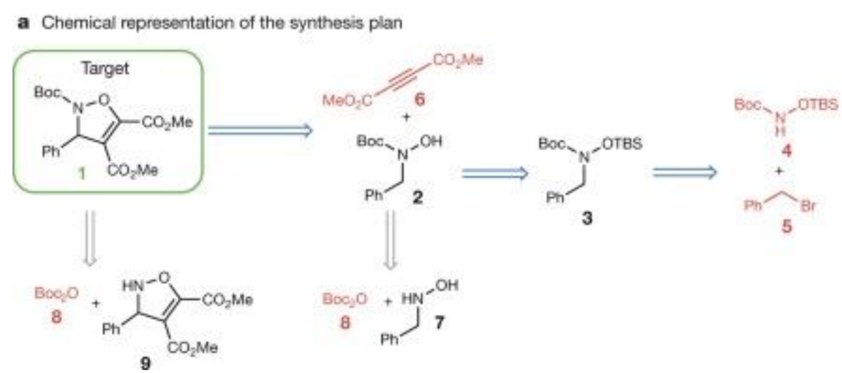

0.09s
View from inside the tokamak
Plasma state reconstruction


a Chemical representation of the synthesis plan
Target
Boc
CO2Me
Ph
1
CO2Me
MeO2C
CO2Me
6
Boc
OH
Ph
2
Boc
OTBS
Ph
3
Boc
OTBS
4
Ph
Br
5
Boc2O
8
HN
CO2Me
Ph
CO2Me
9
Boc2O
8
HN
OH
7
Ph









Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." *Nature* 529.7587 (2016): 484-489.

Fawzi, Alhussein, et al. "Discovering faster matrix multiplication algorithms with reinforcement learning." *Nature* 610.7930 (2022): 47-53.

Degrave, Jonas, et al. "Magnetic control of tokamak plasmas through deep reinforcement learning." *Nature* 602.7897 (2022): 414-419.

Segler, Marwin HS et al. "Planning chemical syntheses with deep neural networks and symbolic AI." *Nature* 555.7698 (2018): 604-610.

Mirhoseini, Azalia, et al. "A graph placement methodology for fast chip design." *Nature* 594.7862 (2021): 207-212.

Many (real-world) problems can be formulated as a

sequential decision-making problem

Many (real-world) problems can be formulated as a

sequential decision-making problem

which may be solved through reinforcement learning.

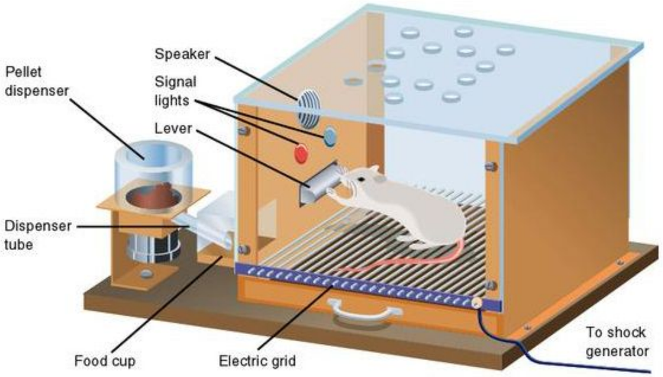
Content

Part I

Introduction

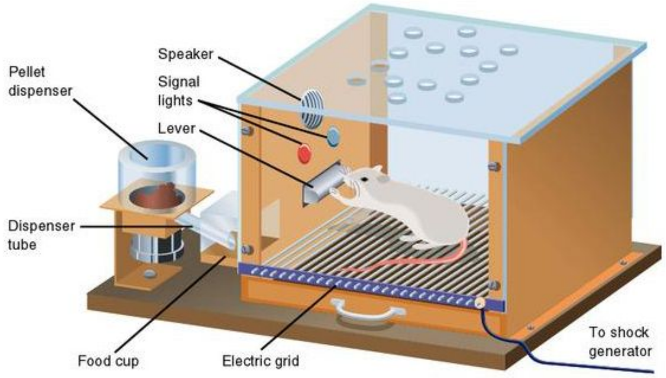
Biology

Biology



Speaker
Pellet dispenser
Signal lights
Lever
Dispenser tube
Food cup
Electric grid
To shock generator

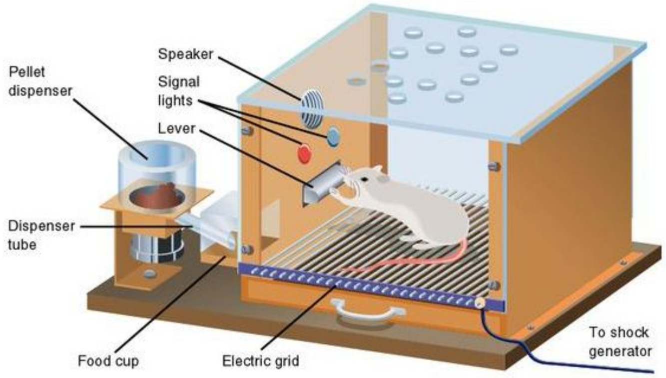
Biology



[*Skinner box*]

[*B.F. Skinner (1904 – 1990)*]

Biology

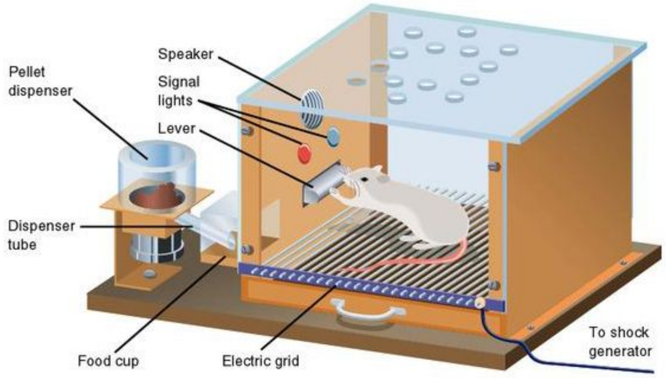


[*Skinner box*]

[*B.F. Skinner (1904 – 1990)*]

Instrumental conditioning:
Learning behaviour based on reward and punishment (trial and error)

Biology



Skinner box

B.F. Skinner (1904 – 1990)

Instrumental conditioning:
Learning behaviour based on reward and punishment (trial and error)

RL is the computational specification of this idea

Supervised versus reinforcement learning

Supervised versus reinforcement learning

	Supervised learning	**Reinforcement learning**
Dataset		
Feedback		

Supervised versus reinforcement learning

	Supervised learning	**Reinforcement learning**
Dataset	Given	
Feedback		

Supervised versus reinforcement learning

	Supervised learning	**Reinforcement learning**
Dataset	Given	Active collection
Feedback		

Supervised versus reinforcement learning

	Supervised learning	**Reinforcement learning**
Dataset	Given	Active collection
Feedback	Full (x with correct y)	

Supervised versus reinforcement learning

	Supervised learning	**Reinforcement learning**
Dataset	Given	Active collection
Feedback	Full (x with correct y)	Partial (~~state with correct action~~) (feedback on some outcomes)

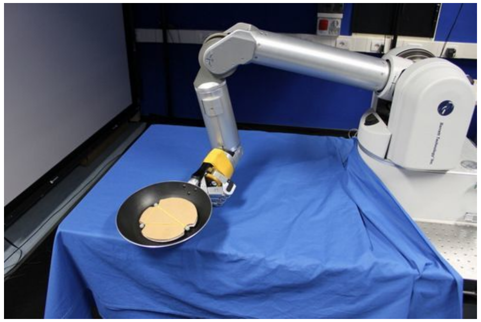
Benefits of Reinforcement Learning

Benefits of Reinforcement Learning

Autonomous behaviour/learning
(only specify goals)

Benefits of Reinforcement Learning

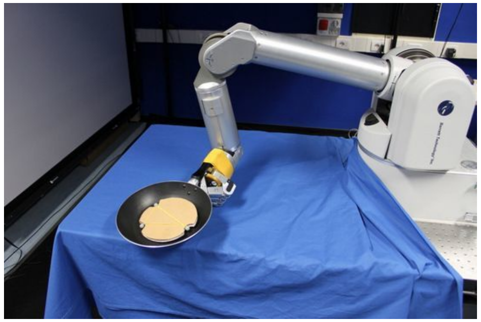
Autonomous behaviour/learning
(only specify goals)



Solve tasks that you can't label
(only need to label the outcome)

Benefits of Reinforcement Learning

Autonomous behaviour/learning
(only specify goals)



Solve tasks that you can't label
(only need to label the outcome)

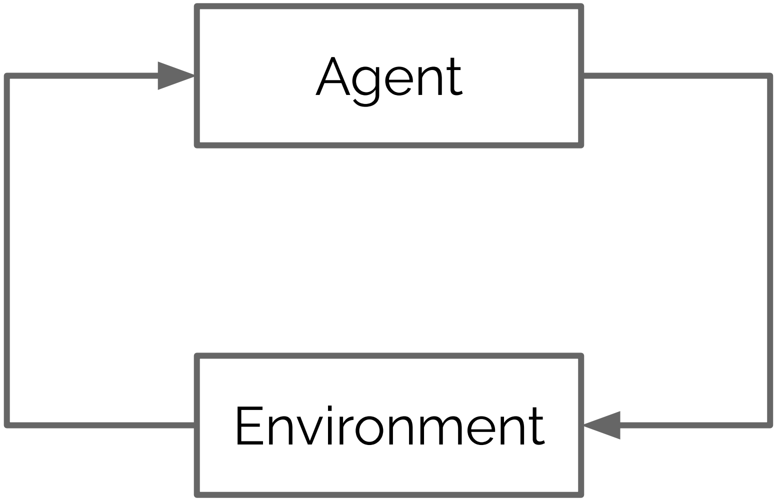
Outperform human solution
(only need to label the outcome)

Part II

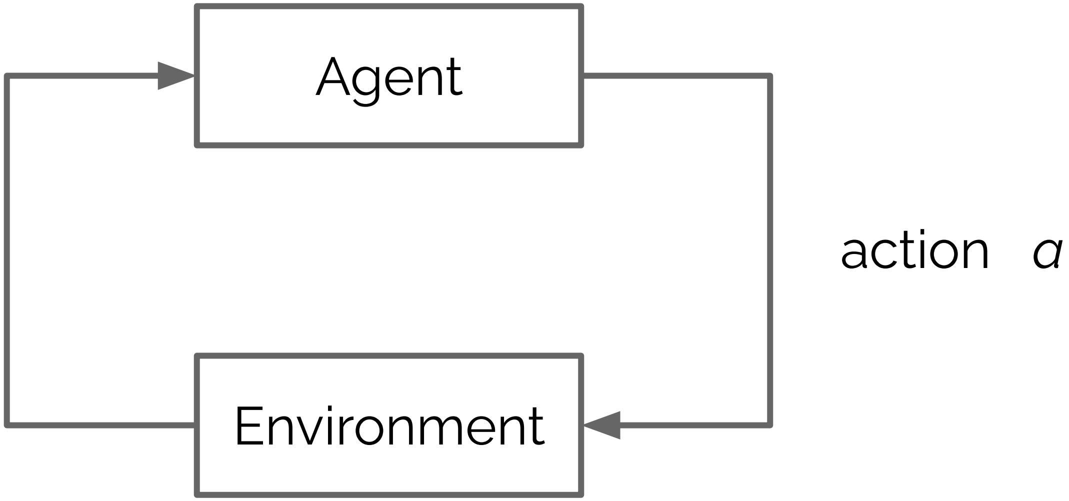
Problem Formulation

Agent-Environment loop

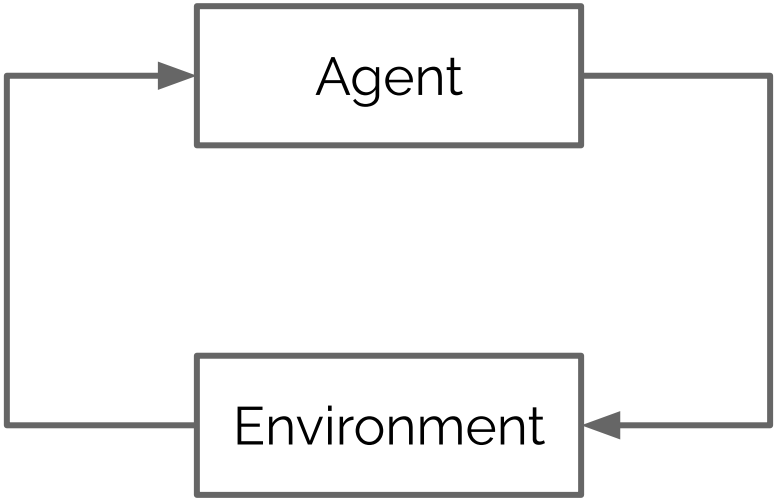
Agent-Environment loop



Agent-Environment loop

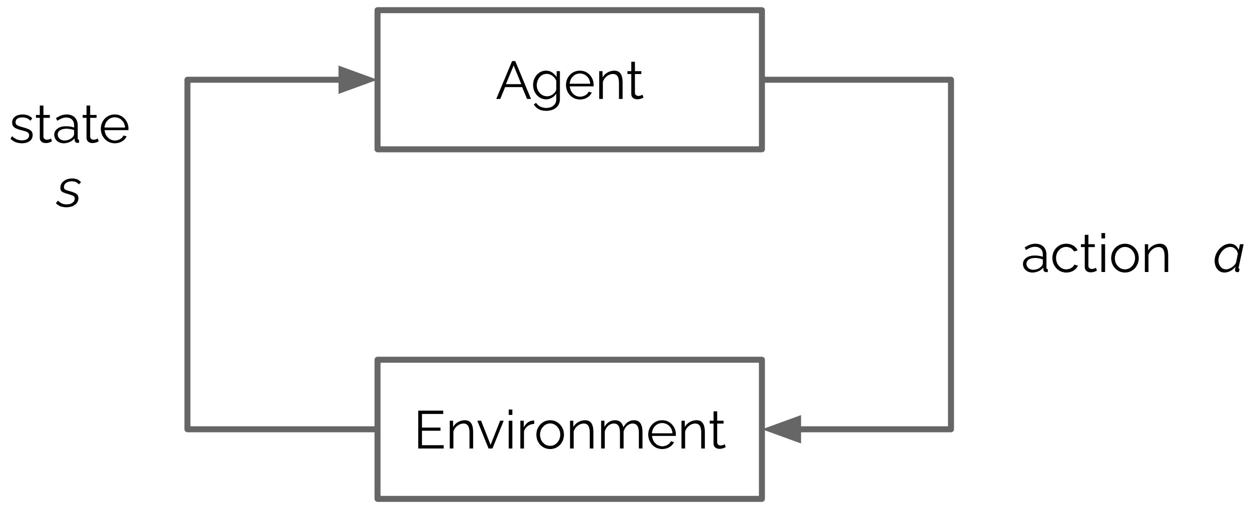


Agent-Environment loop

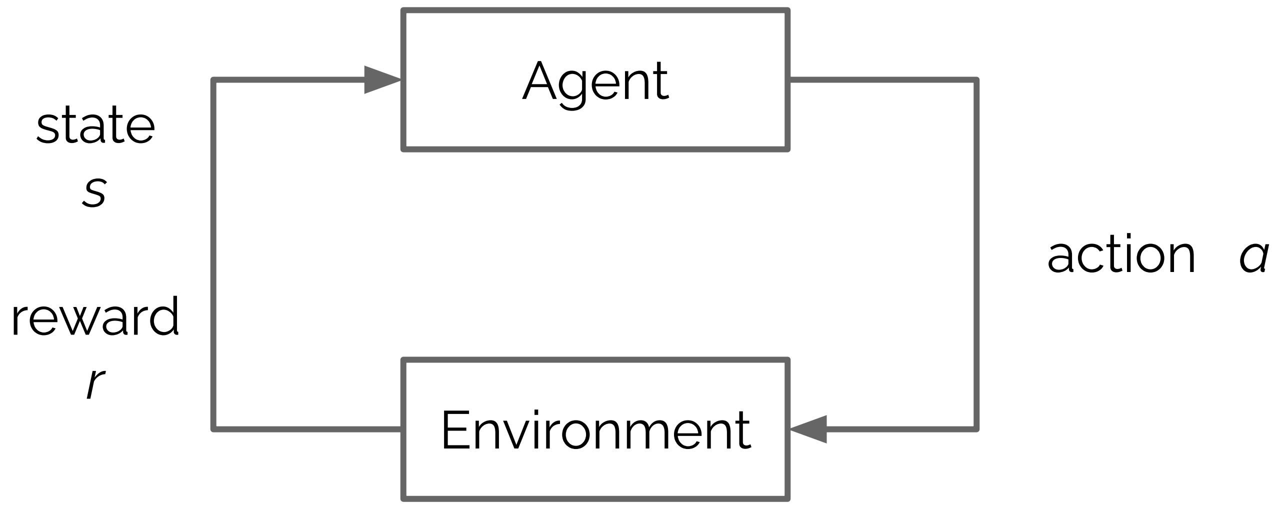


action a

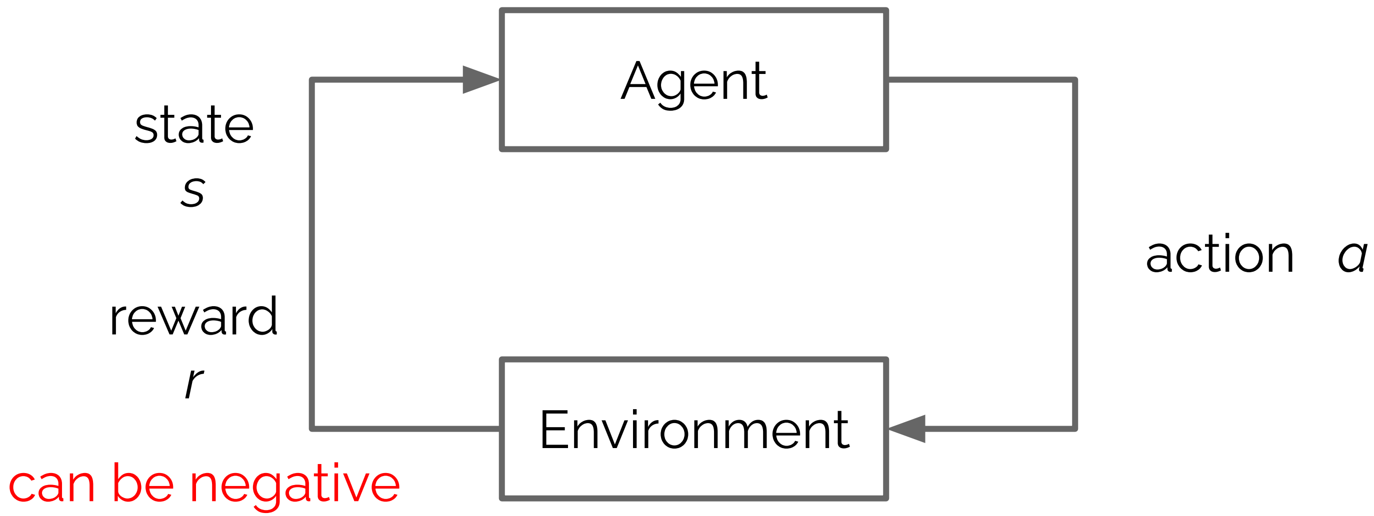
Agent-Environment loop



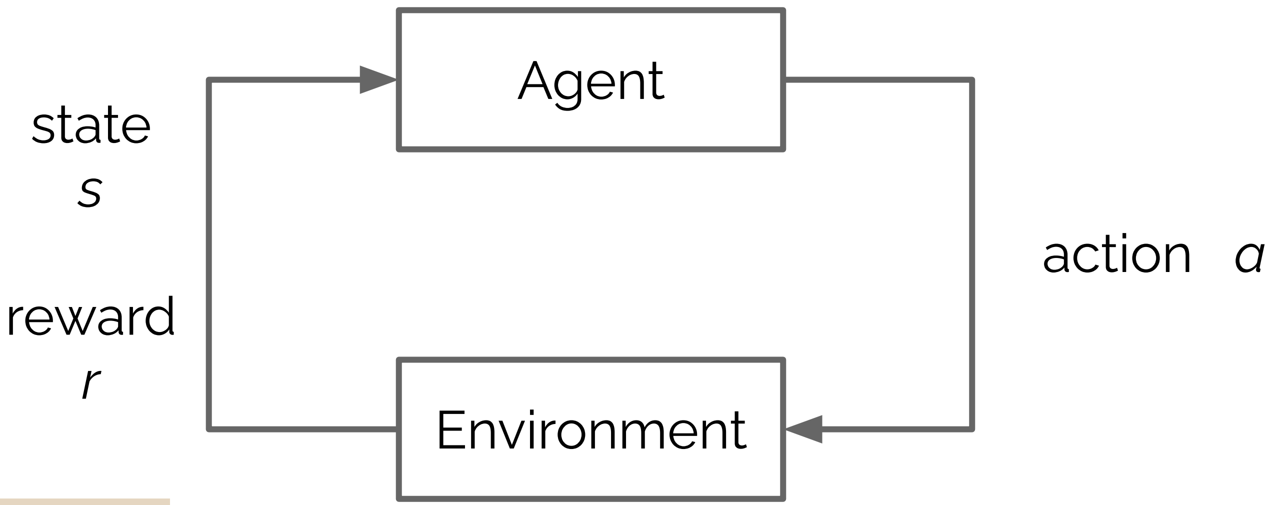
Agent-Environment loop



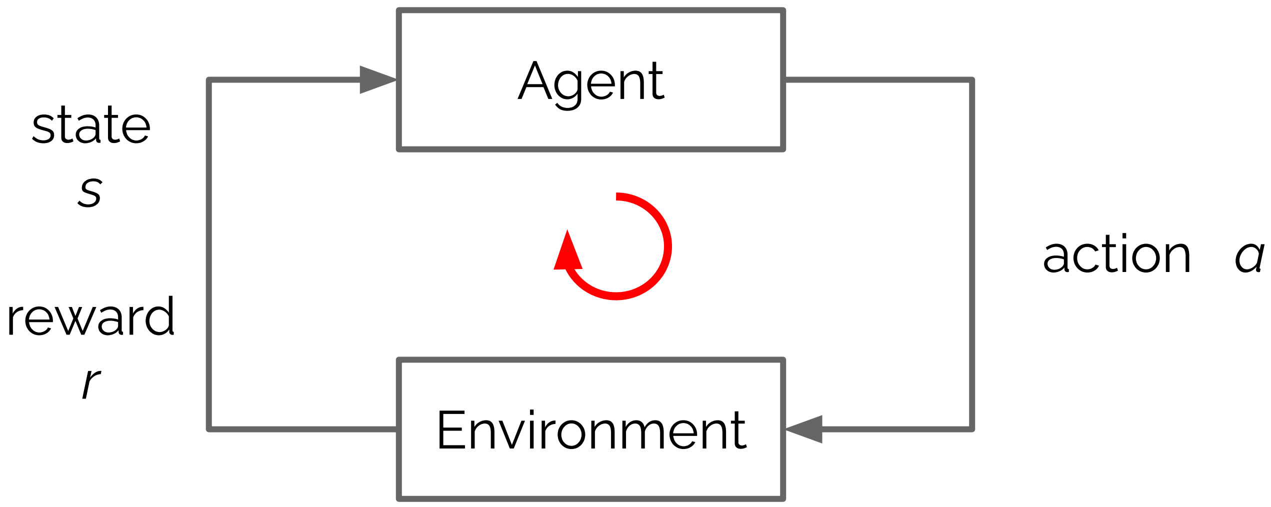
Agent-Environment loop



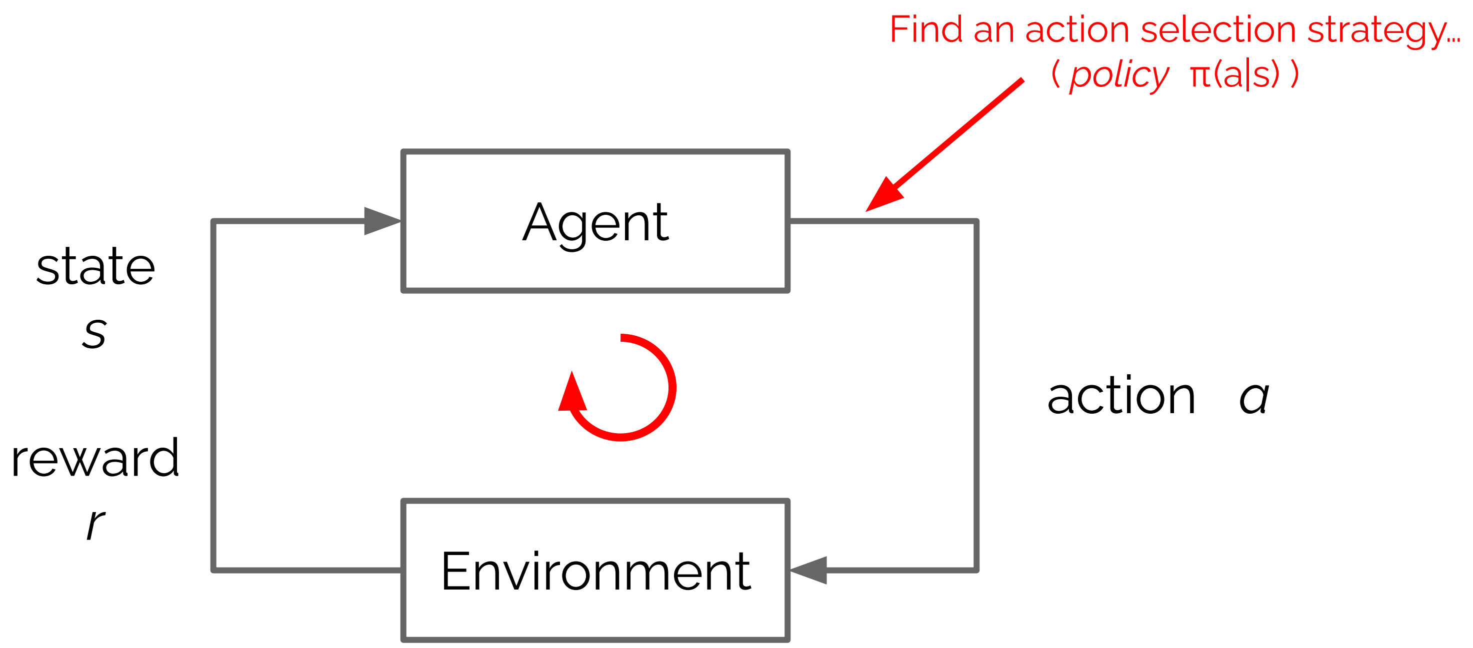
Agent-Environment loop



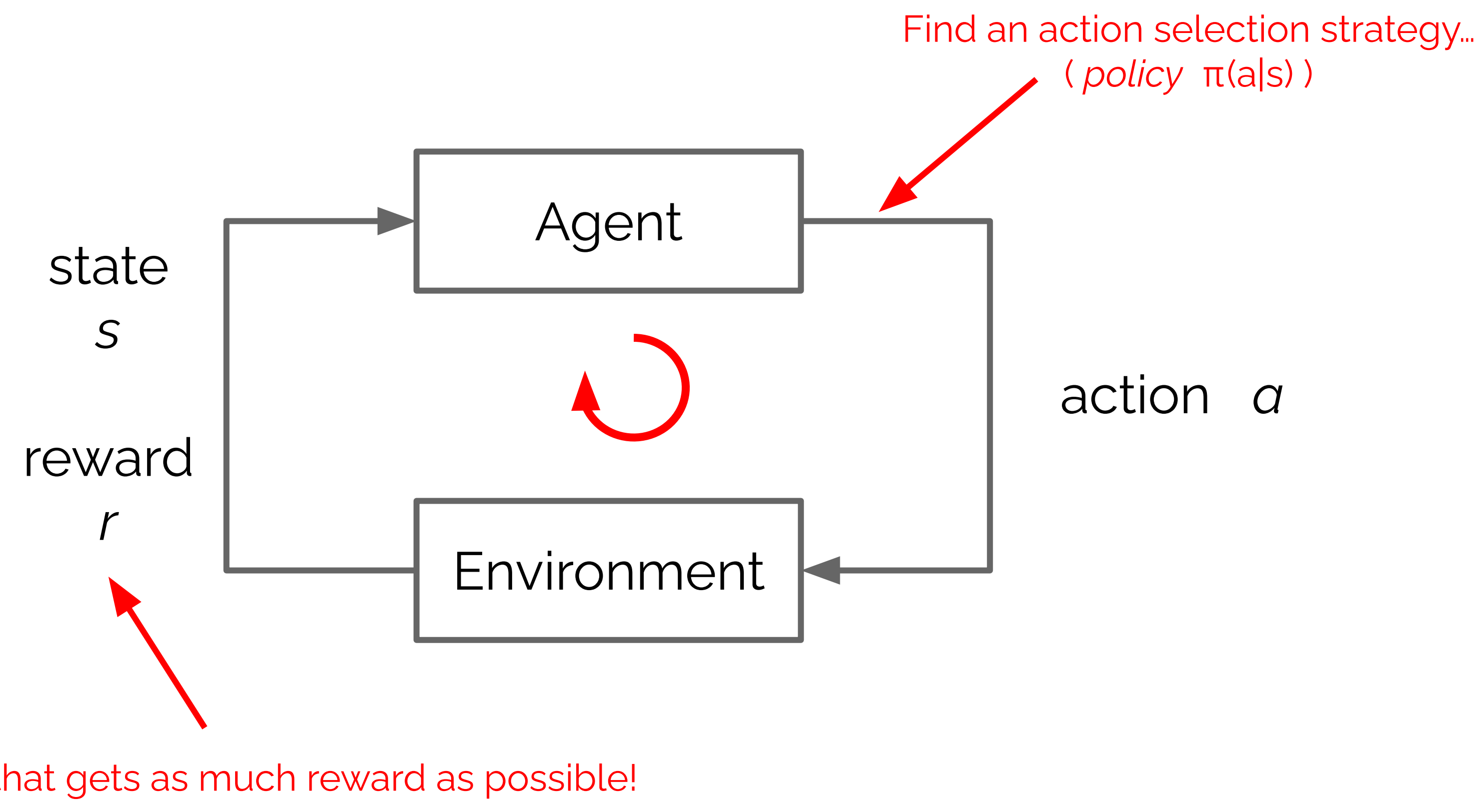
Agent-Environment loop



Agent-Environment loop



Agent-Environment loop



Reward

Reward

Immediate reward

$$r_t$$

Reward

~~Immediate reward~~

Cumulative reward

$$r_t + r_{t+1} + r_{t+2} + \dots$$

Reward

~~Immediate reward~~

~~Cumulative reward~~

Expected cumulative reward

$$\mathbb{E}[r_t + r_{t+1} + r_{t+2} + \ldots \qquad]$$

Reward

~~Immediate reward~~

~~Cumulative reward~~

Expected cumulative reward

$$\mathbb{E}[r_t + r_{t+1} + r_{t+2} + \ldots \qquad\qquad\qquad]$$

Average over stochasticity in 1) environment and 2) own policy.

Reward

~~Immediate reward~~

~~Cumulative reward~~

Expected cumulative reward

= Value

$$Q^{\pi}(s, a) = \mathbb{E}[r_t + r_{t+1} + r_{t+2} + \ldots | s_t = s, a_t = a]$$

Reward

~~Immediate reward~~

~~Cumulative reward~~

Expected cumulative reward

= Value

$$Q^{\pi}(s,a) = \mathbb{E}[r_t + r_{t+1} + r_{t+2} + ...|s_t = s, a_t = a]$$

Q-value: total reward we get on average after taking action a in state s.

Reward

~~Immediate reward~~

~~Cumulative reward~~

Expected cumulative reward

= Value

$$Q^{\boxed{\pi}}(s, a) = \mathbb{E}[r_t + r_{t+1} + r_{t+2} + ...|s_t = s, a_t = a]$$

Q-value: total reward we get on average after taking action a in state s.
- Depends on our own future behaviour π (if we act stupid, reward will be low)

Reward

~~Immediate reward~~

~~Cumulative reward~~

Expected cumulative reward

= Value

$$Q^{\boxed{\pi}}(s,a) = \mathbb{E}[r_t + r_{t+1} + r_{t+2} + \ldots | s_t = s, a_t = a]$$

Q-value: total reward we get on average after taking action a in state s.
- Depends on our own future behaviour π (if we act stupid, reward will be low)

Can show each state-action has one optimal value, denoted by Q*(s,a).
- These are the quantities we want to know!

Illustration: Optimal Value

Illustration: Optimal Value

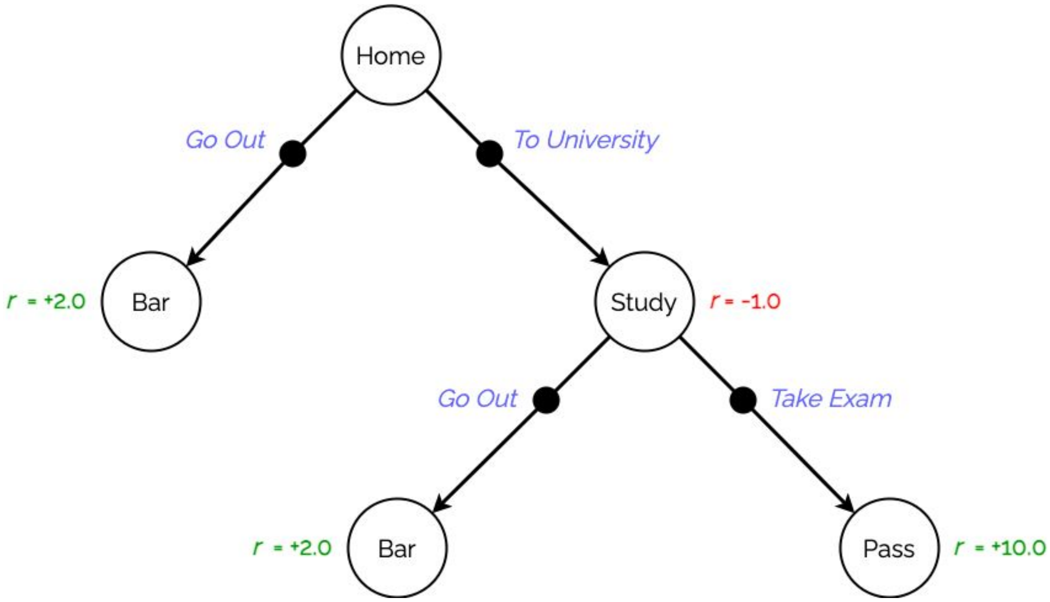
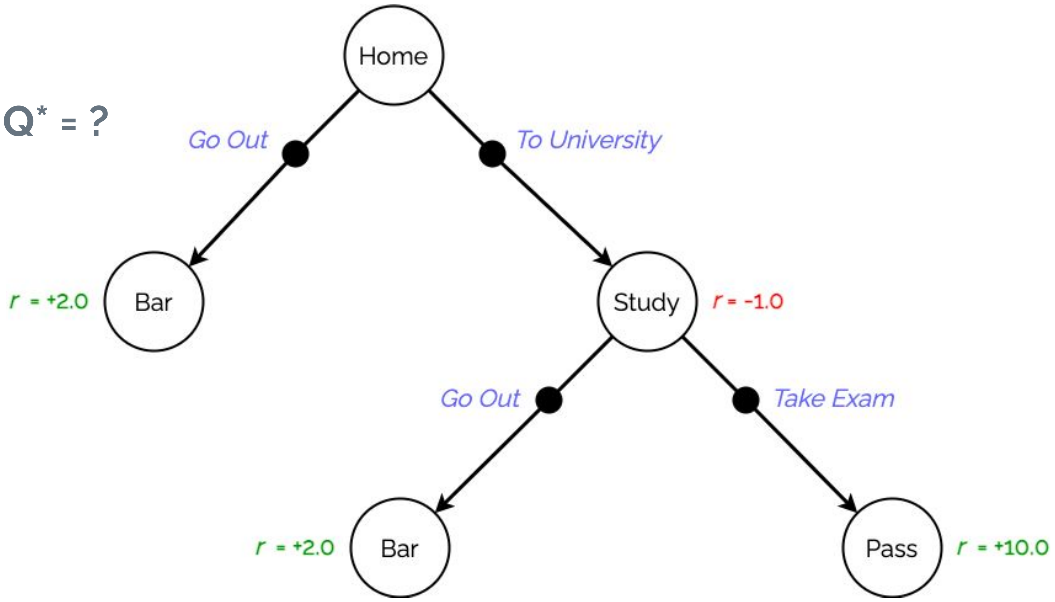


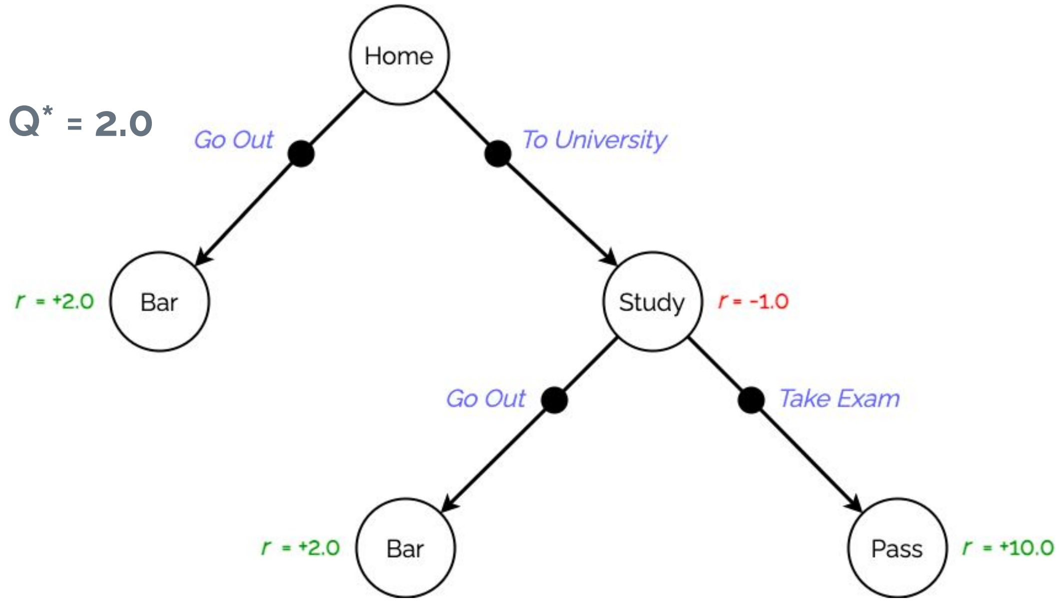
Illustration: Optimal Value

$Q^* = ?$



Question: What is Q*(Home, Go Out)?

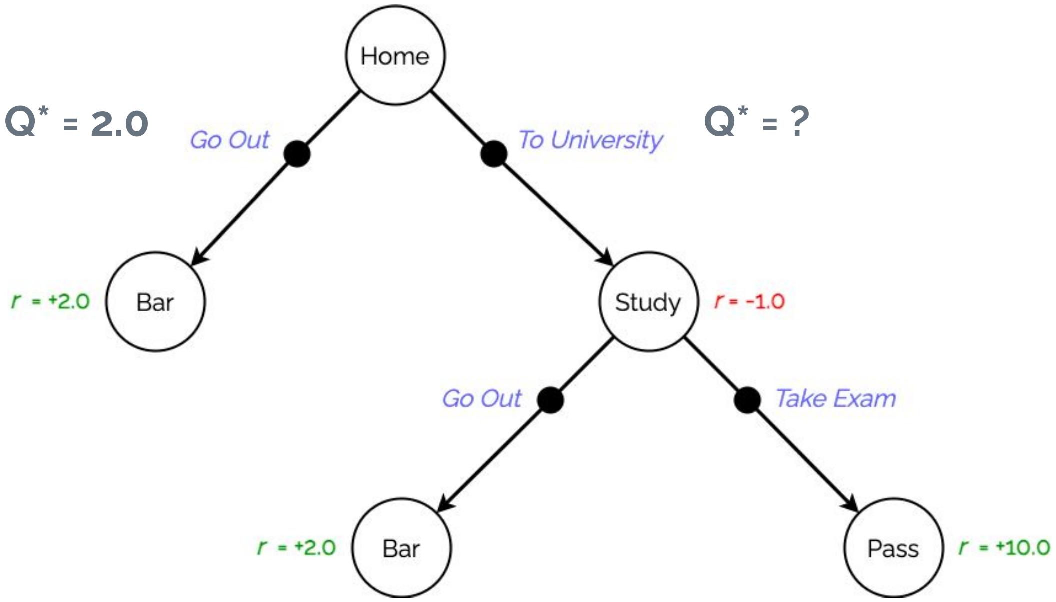
Illustration: Optimal Value



Question: What is Q*(Home, Go Out)?

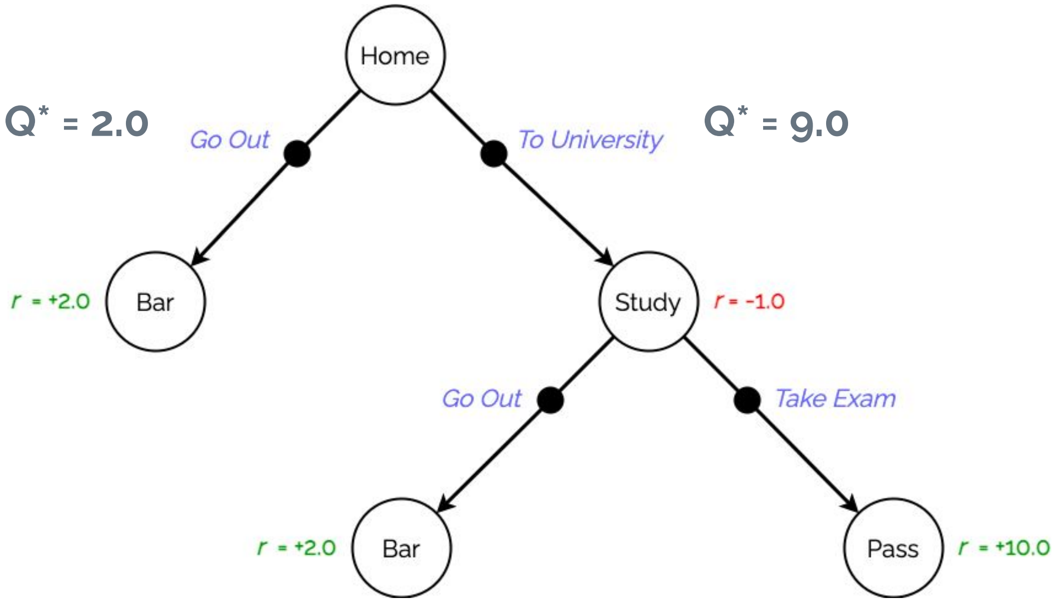
Answer: 2.0

Illustration: Optimal Value



Question: What is Q*(Home, To University)?

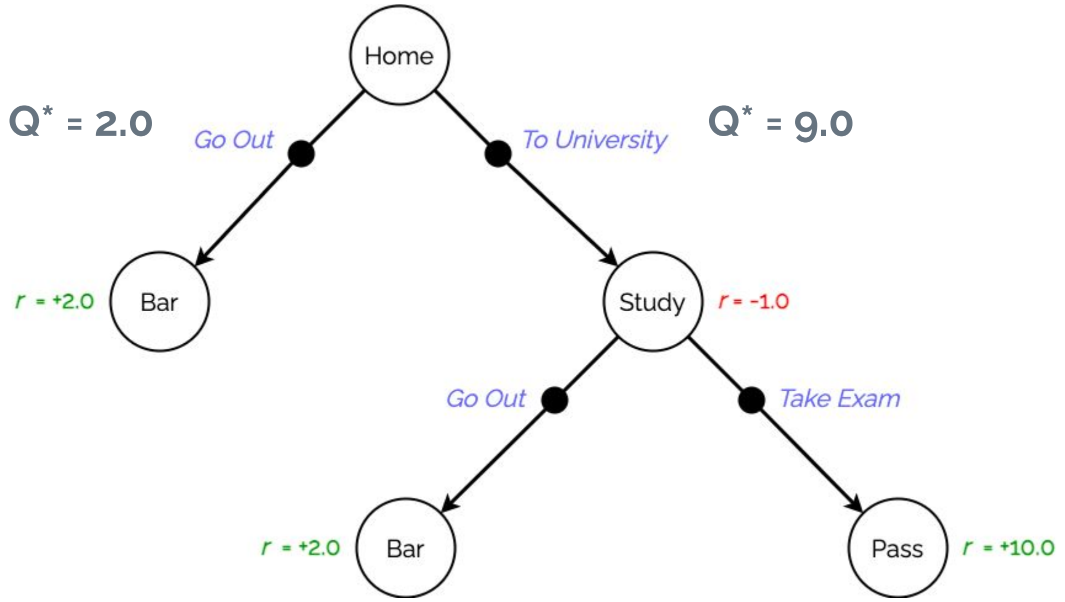
Illustration: Optimal Value



Question: What is Q*(Home, To University)?

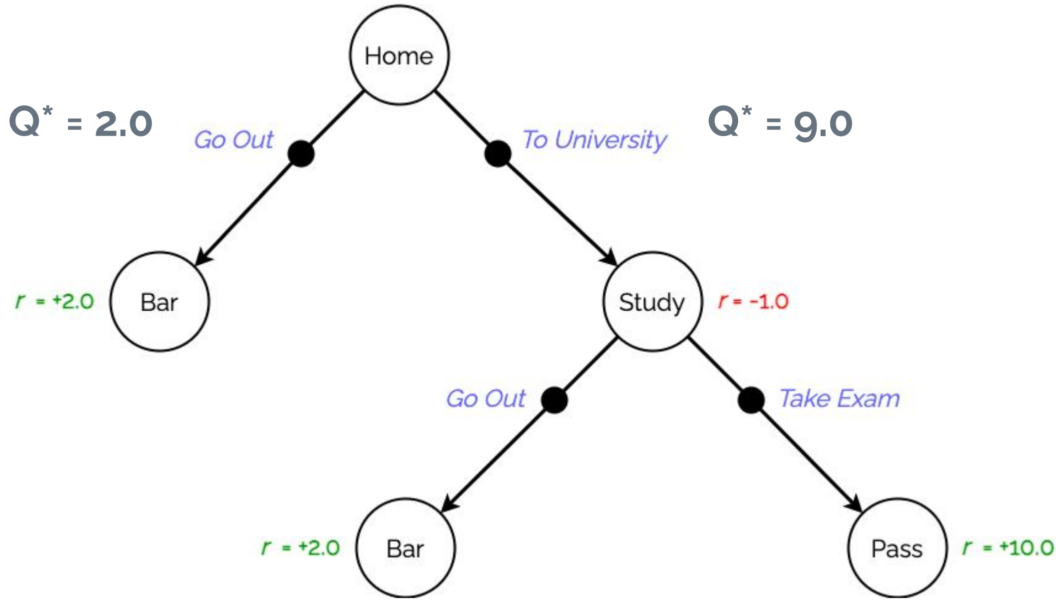
Answer: -1.0 + 10.0 = 9.0

Illustration: Optimal Value



Question: What should you do at Home?

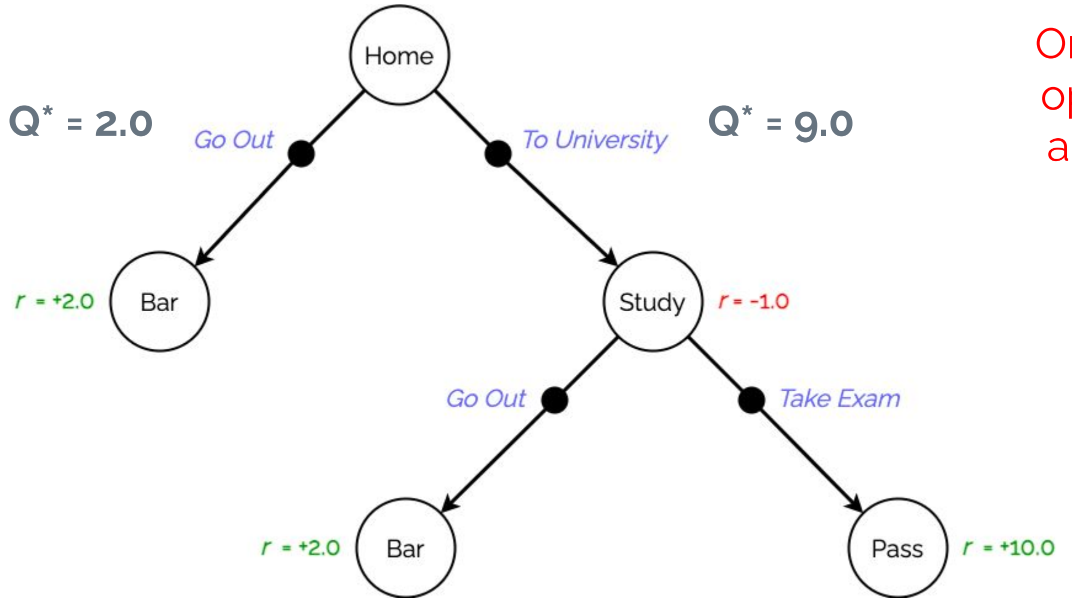
Illustration: Optimal Value



Question: What should you do at Home?

Answer: Come to University!

Illustration: Optimal Value



Once we know the optimal values we also know how to act optimally

Question: What should you do at Home?

Answer: Come to University!

Part III

The Reinforcement Learning Cycle

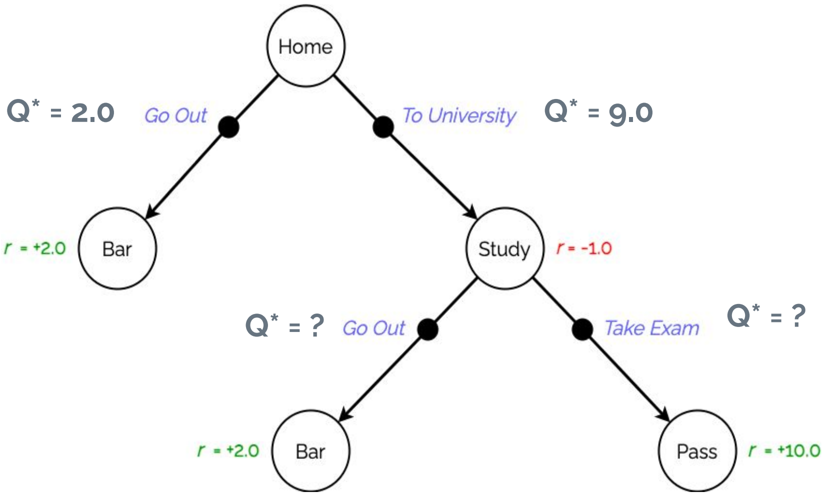
Challenge

Challenge

Problem: In practice we don't know the problem structure and optimal Q-values.

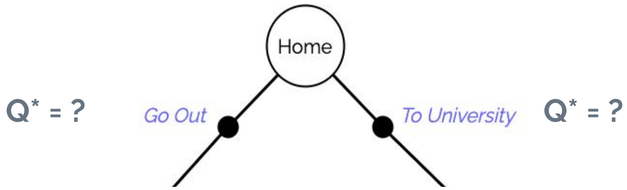
Challenge

Problem: In practice we don't know the problem structure and optimal Q-values.



Challenge

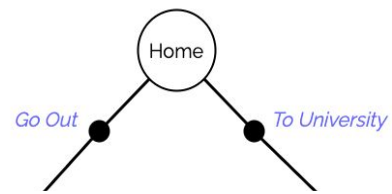
Problem: In practice we don't know the problem structure and optimal Q-values.



Challenge

Problem: In practice we don't know the problem structure and optimal Q-values.

$Q^* = ?$



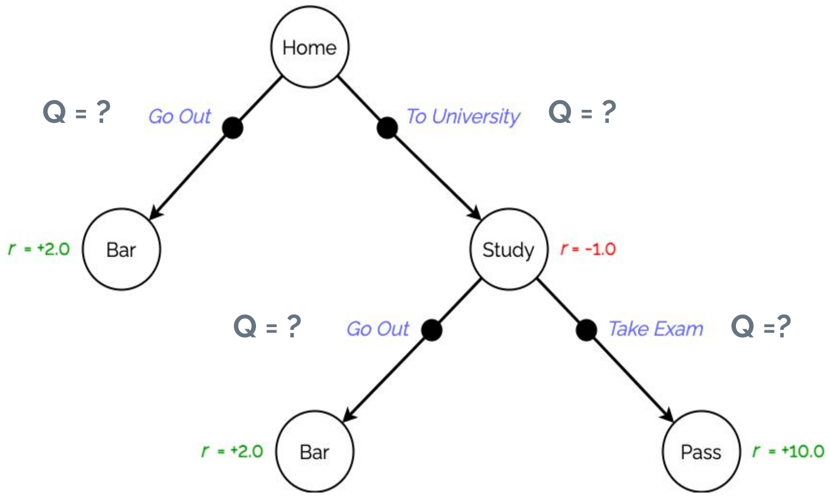
$Q^* = ?$

Solution: Learn through trial and error.

The Reinforcement Learning Cycle

The Reinforcement Learning Cycle

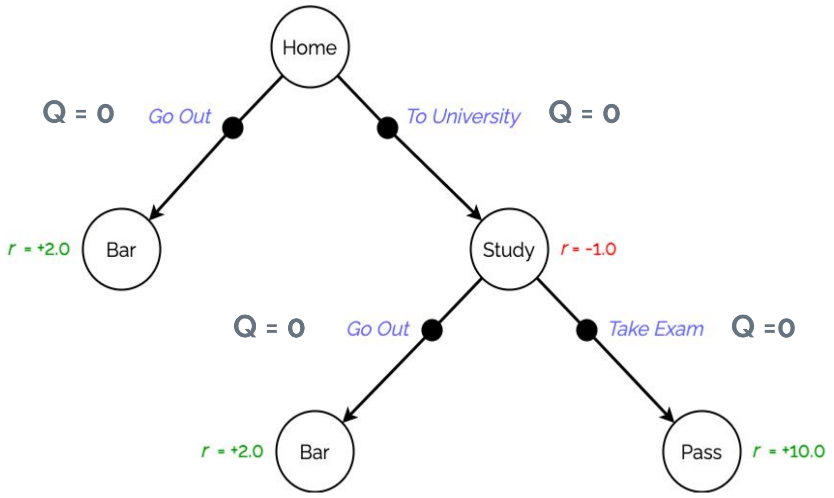
Pseudocode



The Reinforcement Learning Cycle

Pseudocode

Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

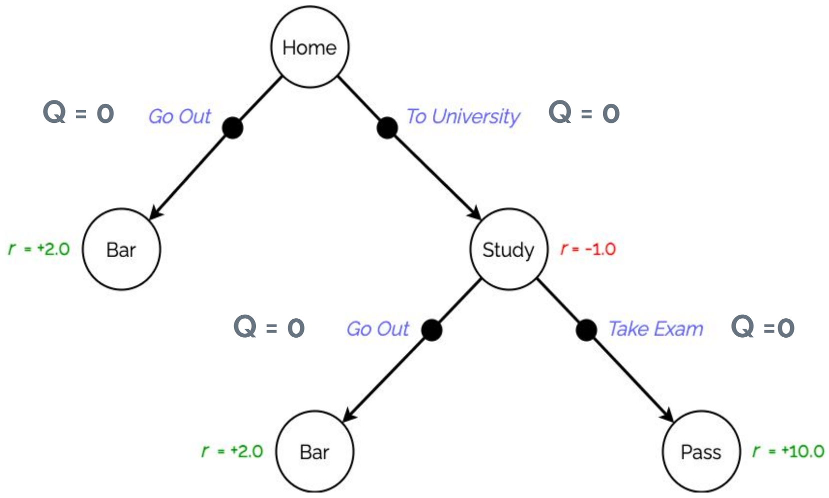


The Reinforcement Learning Cycle

Pseudocode

Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

Repeat:



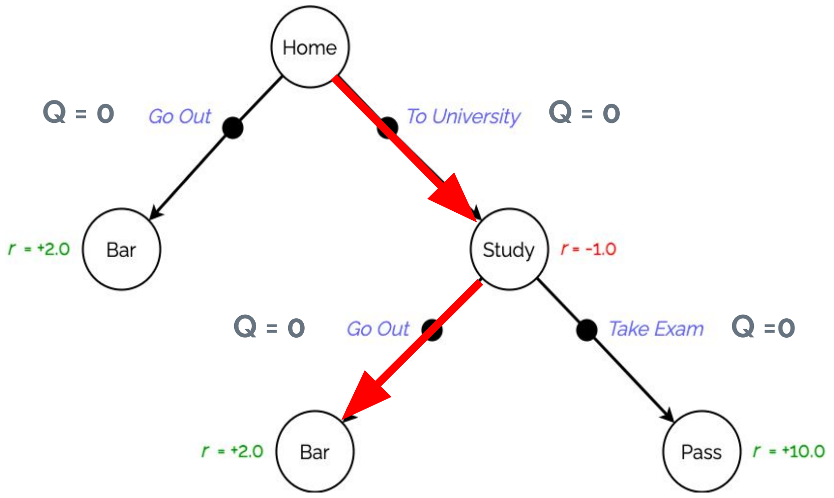
The Reinforcement Learning Cycle

Pseudocode

Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

Repeat:

1) Exploration: Sample a sequence of actions.



The Reinforcement Learning Cycle

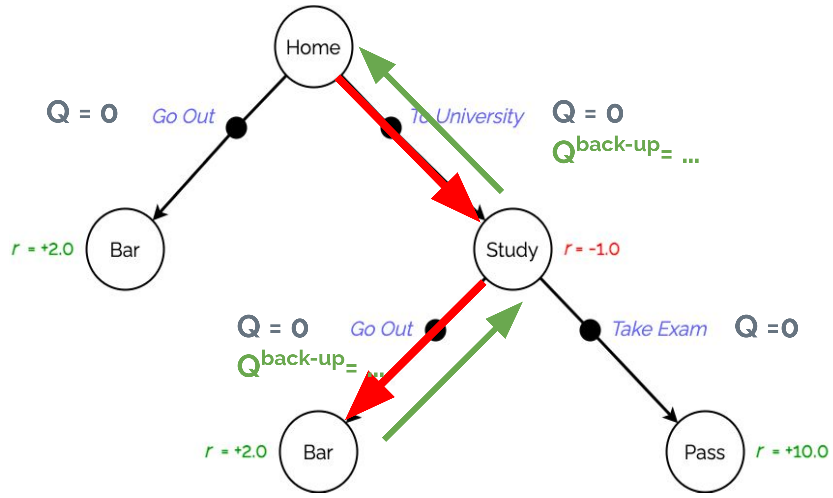
Pseudocode

Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

Repeat:

1) Exploration: Sample a sequence of actions.

2) Credit assignment: Compute new value estimates **$Q^{back\text{-}up}$(s,a)** for all actions along the path.



The Reinforcement Learning Cycle

Pseudocode

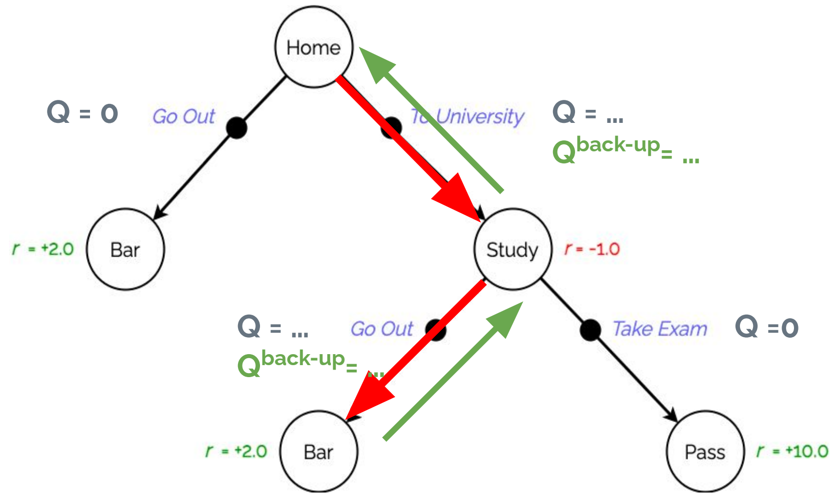
Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

Repeat:

1) Exploration: Sample a sequence of actions.

2) Credit assignment: Compute new value estimates $\mathbf{Q^{back\text{-}up}(s,a)}$ for all actions along the path.

3) Learning update: Adjust our **Q(s,a)** solution based on the back-up estimates $\mathbf{Q^{back\text{-}up}(s,a)}$.



The Reinforcement Learning Cycle

Pseudocode

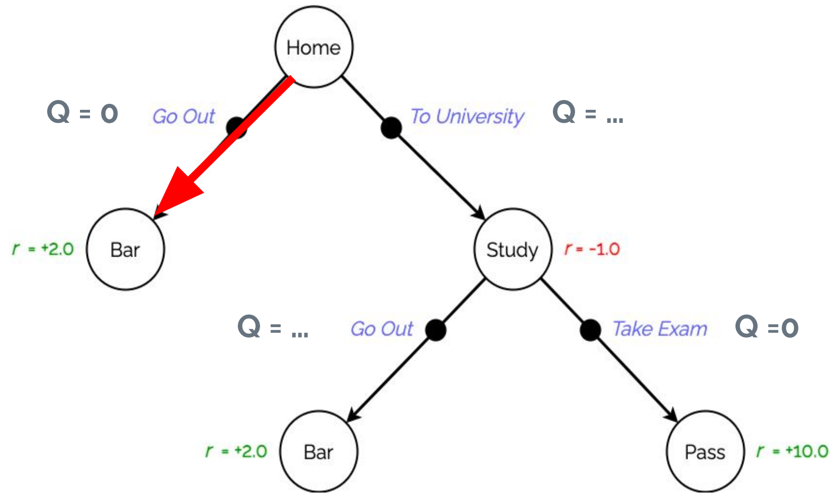
Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

Repeat:

1) **Exploration**: Sample a sequence of actions.

2) Credit assignment: Compute new value estimates $\mathbf{Q^{back\text{-}up}(s,a)}$ for all actions along the path.

3) Learning update: Adjust our **Q(s,a)** solution based on the back-up estimates $\mathbf{Q^{back\text{-}up}(s,a)}$.



The Reinforcement Learning Cycle

Pseudocode

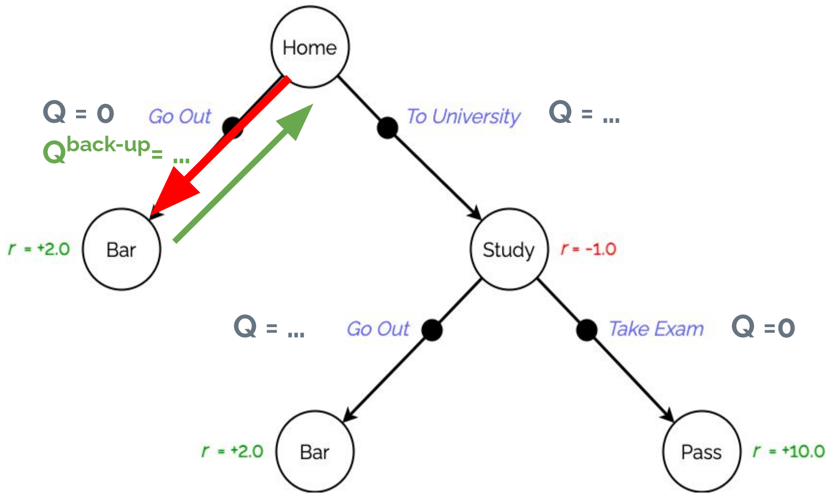
Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

Repeat:

1) Exploration: Sample a sequence of actions.

2) **Credit assignment**: Compute new value estimates $\mathbf{Q^{back\text{-}up}(s,a)}$ for all actions along the path.

3) Learning update: Adjust our **Q(s,a)** solution based on the back-up estimates $\mathbf{Q^{back\text{-}up}(s,a)}$.



The Reinforcement Learning Cycle

Pseudocode

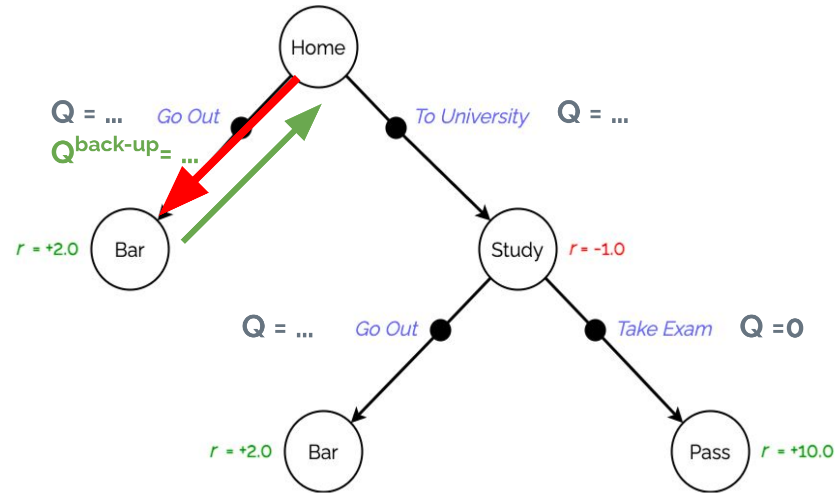
Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

Repeat:

1) Exploration: Sample a sequence of actions.

2) Credit assignment: Compute new value estimates $\mathbf{Q^{back\text{-}up}(s,a)}$ for all actions along the path.

3) **Learning update**: Adjust our **Q(s,a)** solution based on the back-up estimates $\mathbf{Q^{back\text{-}up}(s,a)}$.



The Reinforcement Learning Cycle

Pseudocode

Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

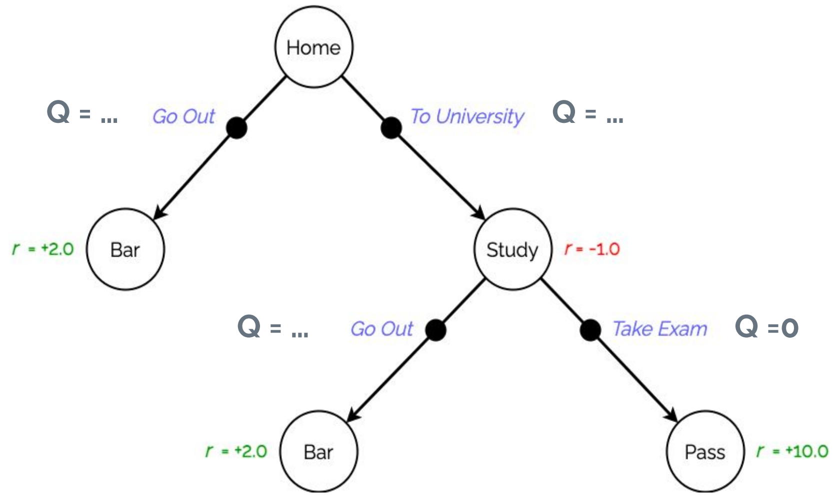
Repeat:

1) Exploration: Sample a sequence of actions.

2) Credit assignment: Compute new value estimates $\mathbf{Q^{back\text{-}up}(s,a)}$ for all actions along the path.

3) **Learning update**: Adjust our **Q(s,a)** solution based on the back-up estimates $\mathbf{Q^{back\text{-}up}(s,a)}$.

etc.



The Reinforcement Learning Cycle

Pseudocode

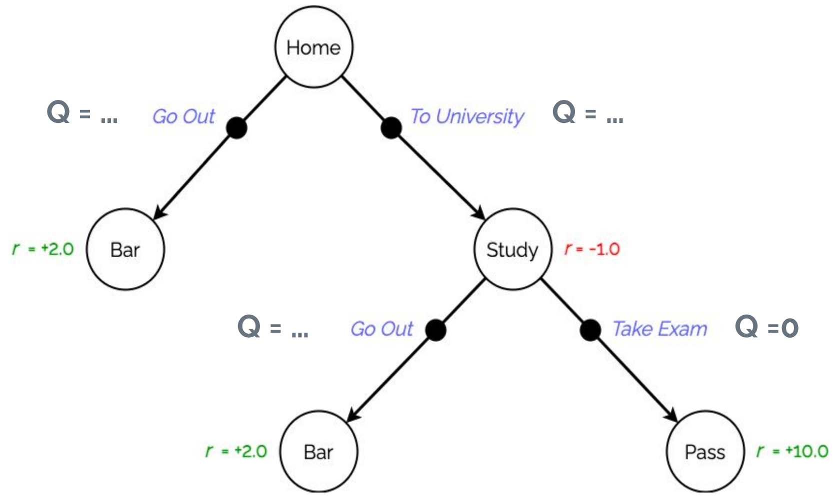
Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

Repeat:

1) Exploration: Sample a sequence of actions.

2) Credit assignment: Compute new value estimates $\mathbf{Q^{back\text{-}up}(s,a)}$ for all actions along the path.

3) **Learning update**: Adjust our **Q(s,a)** solution based on the back-up estimates $\mathbf{Q^{back\text{-}up}(s,a)}$.



We will discuss all three steps, but in reverse order

Part III A

Learning Update

Learning update (tabular)

Learning update (tabular)

(Really a supervised learning topic, we will briefly discuss this in one slide.)

Learning update (tabular)

$$Q(s,a) \leftarrow Q(s,a) + \eta \cdot \Big(Q^{\text{back-up}}(s,a) - Q(s,a) \Big)$$

Learning update (tabular)

$$Q(s,a) \leftarrow Q(s,a) + \boxed{\eta} \cdot \left(Q^{\text{back-up}}(s,a) - Q(s,a)\right)$$

$$\text{learning rate:} \qquad \eta \in [0,1]$$

Learning update (tabular)

$$\boxed{Q(s,a) \leftarrow} Q(s,a) + \eta \cdot \left(Q^{\text{back-up}}(s,a) - Q(s,a) \right)$$

To update our solution…

Learning update (tabular)

$$Q(s,a) \leftarrow \boxed{Q(s,a)} + \eta \cdot \left(Q^{\text{back-up}}(s,a) - Q(s,a)\right)$$

To update our solution we take the current solution…

Learning update (tabular)

$$Q(s,a) \leftarrow Q(s,a) \boxed{+ \eta \cdot} \Big(Q^{\text{back-up}}(s,a) - Q(s,a)\Big)$$

To update our solution we take the current solution and move it a (small) step…

Learning update (tabular)

$$Q(s,a) \leftarrow Q(s,a) + \eta \cdot \boxed{\left(Q^{\text{back-up}}(s,a) - Q(s,a)\right)}$$

To update our solution we take the current solution and move it a (small) step…

…in the direction of the back-up estimate.

Learning update (tabular)

'training target'

$$Q(s,a) \leftarrow Q(s,a) + \eta \cdot \left(Q^{\text{back-up}}(s,a) - Q(s,a) \right)$$

To update our solution we take the current solution and move it a (small) step…

…in the direction of the back-up estimate.

The Reinforcement Learning Cycle

Pseudocode

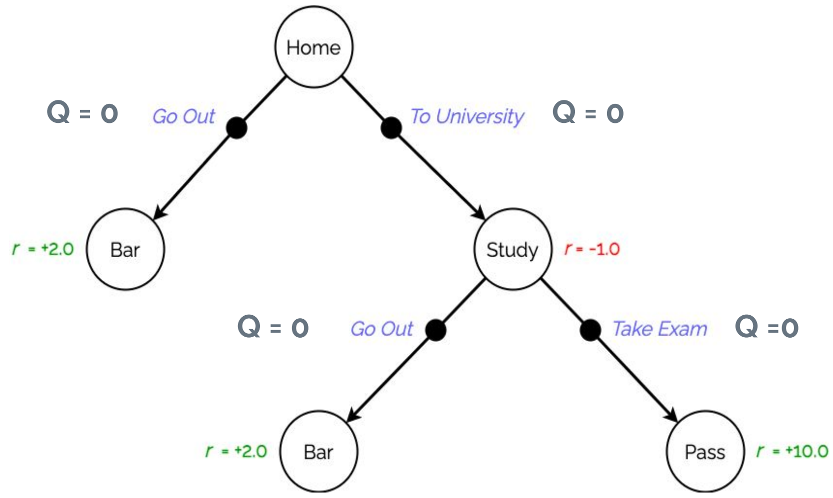
Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

Repeat:

1) Exploration: Sample a sequence of actions.

2) Credit assignment: Compute new value estimates $\mathbf{Q^{back\text{-}up}(s,a)}$ for all actions along the path.

3) Learning update: Adjust our **Q(s,a)** solution based on the back-up estimates $\mathbf{Q^{back\text{-}up}(s,a)}$.



The Reinforcement Learning Cycle

Pseudocode

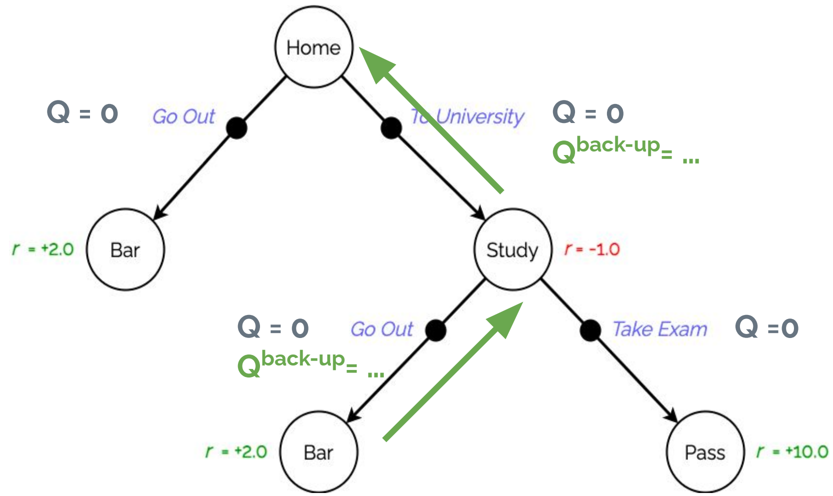
Initialize **Q(s,a)** solution estimates for all states and actions (e.g. to 0)

Repeat:

1) Exploration: Sample a sequence of actions.

2) **Credit assignment**: Compute new value estimates **$Q^{\text{back-up}}(s,a)$** for all actions along the path.

3) Learning update: Adjust our **Q(s,a)** solution based on the back-up estimates **$Q^{\text{back-up}}(s,a)$**.



Part III B

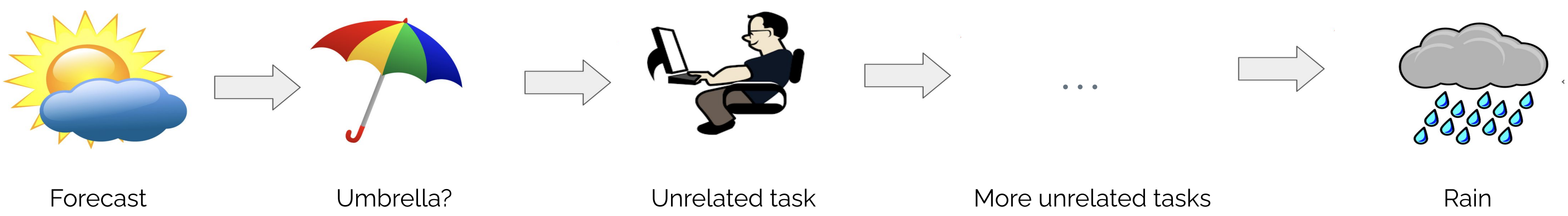
Credit Assignment

Credit assignment

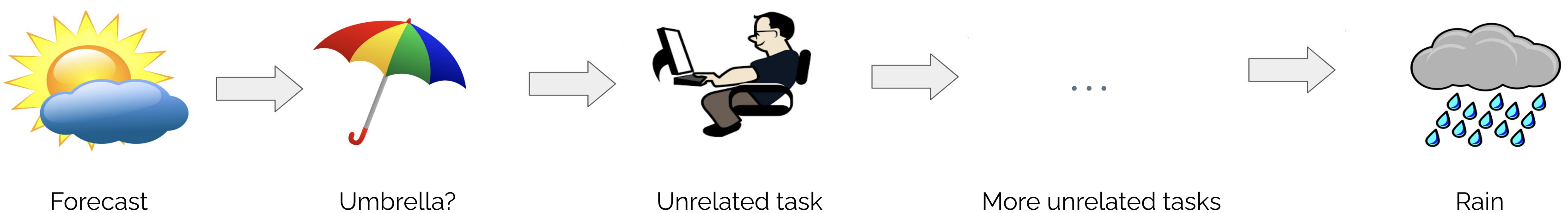
Credit assignment

(Also a topic in neural network training, but then the objective is differentiable)

Credit assignment



Credit assignment



Question: You get the reward (not soaked), but which of your previous actions deserve credit?

Credit assignment

You think this is easy, but humans actually also struggle:

Credit assignment

You think this is easy, but humans actually also struggle:

Credit assignment

You think this is easy, but humans actually also struggle:

Credit assignment

You think this is easy, but humans actually also struggle:

Credit assignment

You think this is easy, but humans actually also struggle:

Credit assignment

You think this is easy, but humans actually also struggle:

Credit assignment

You think this is easy, but humans actually also struggle:

Credit assignment

You think this is easy, but humans actually also struggle:

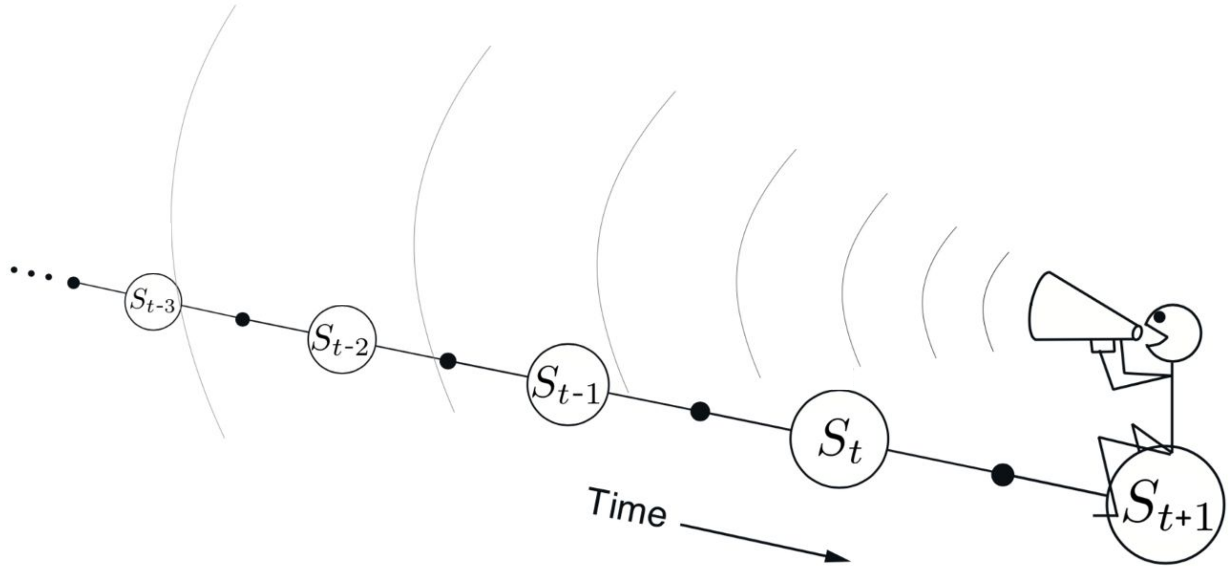
Superstition

=

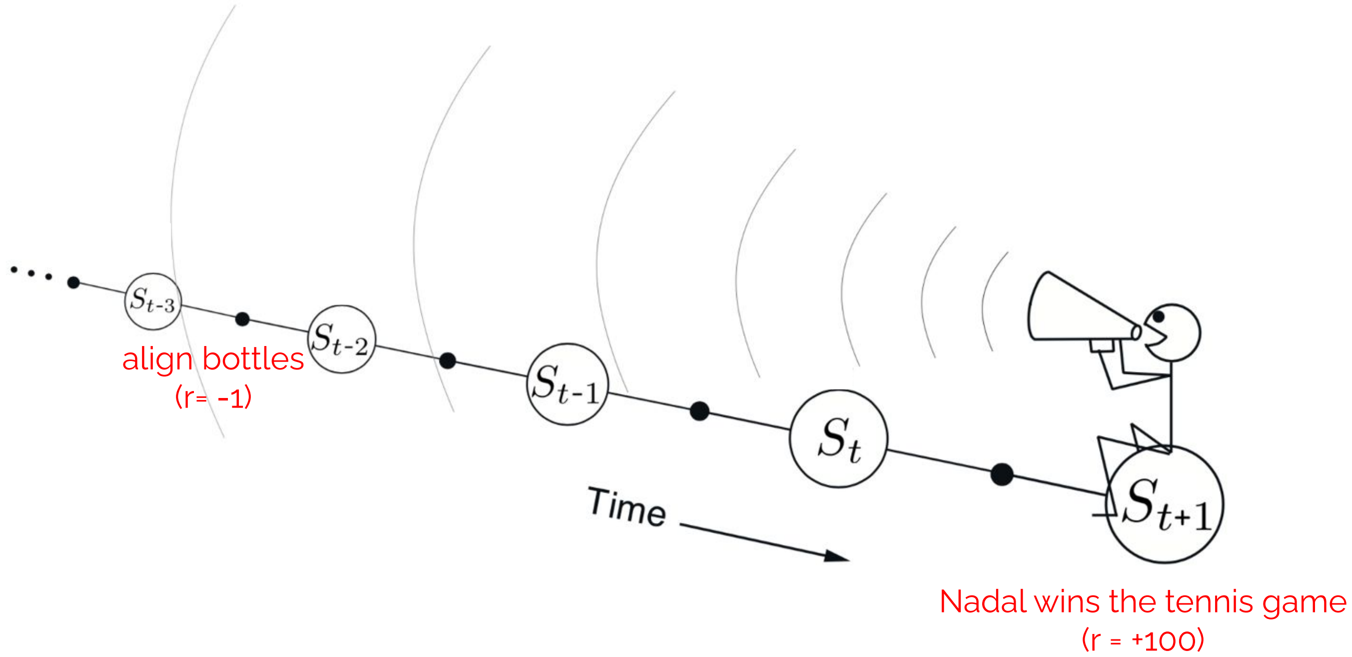
failed credit assignment

Credit assignment

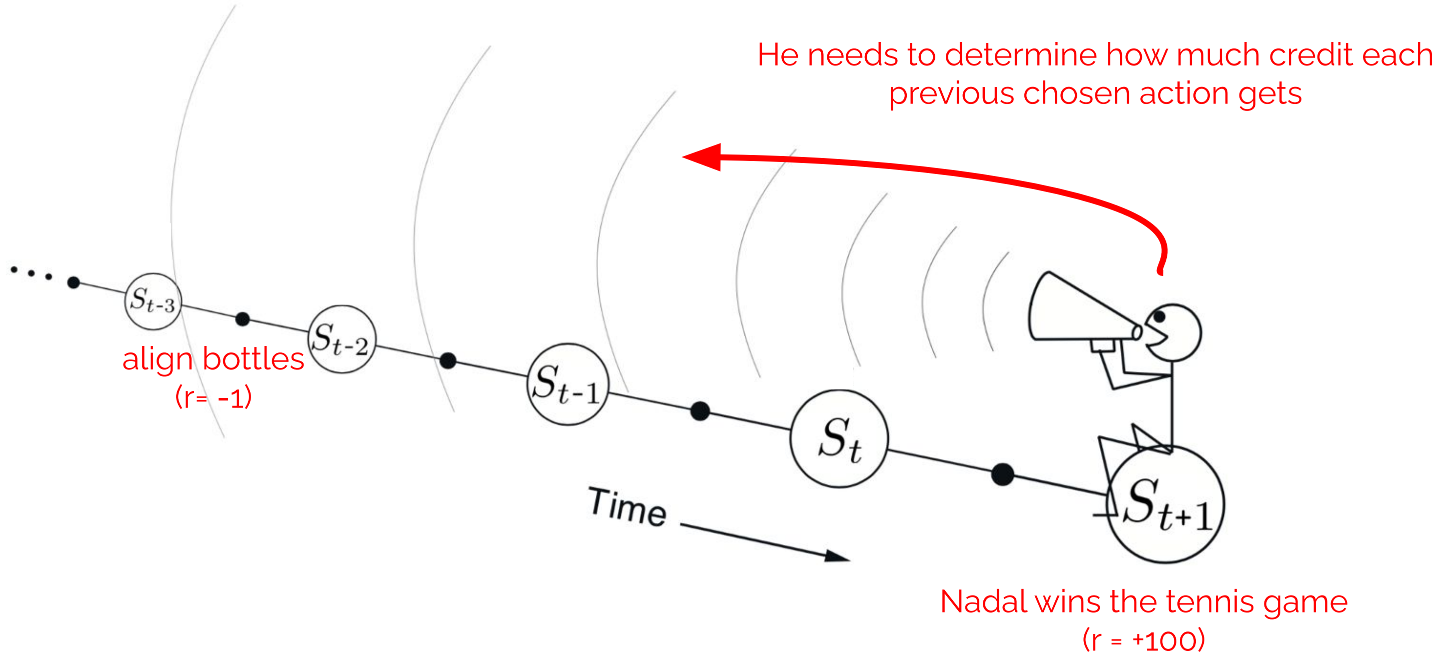
Credit assignment



Credit assignment

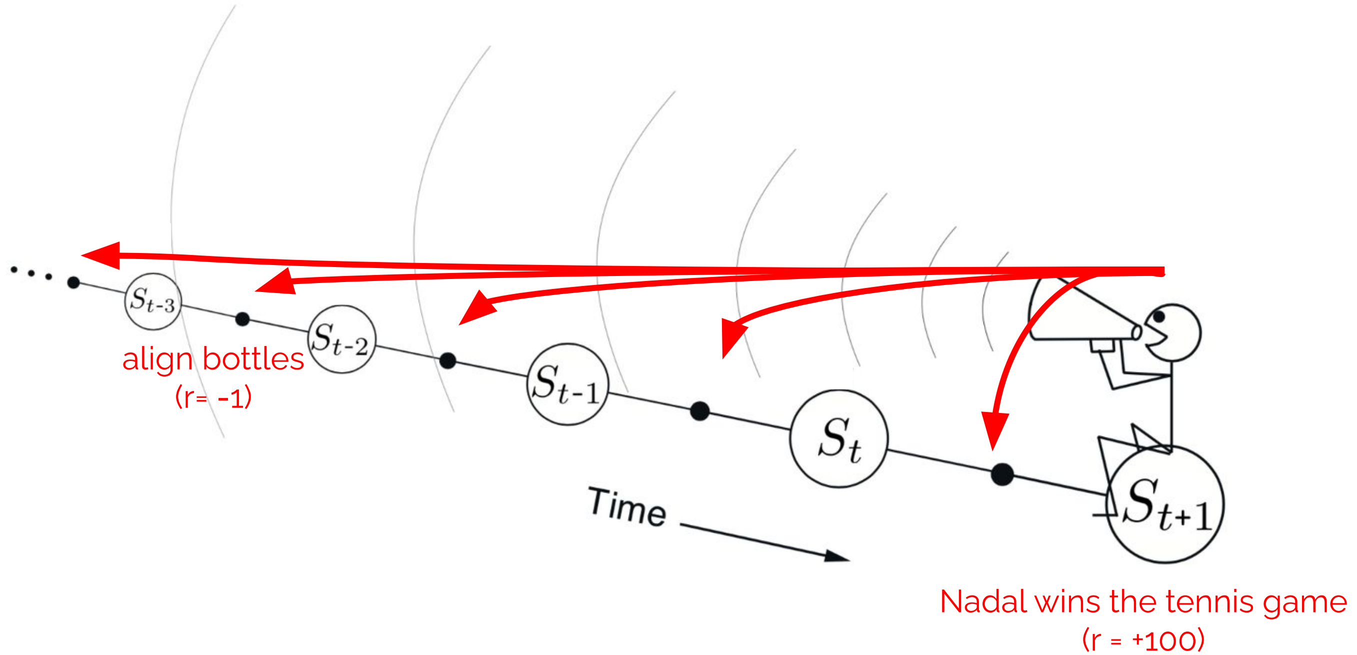


Credit assignment

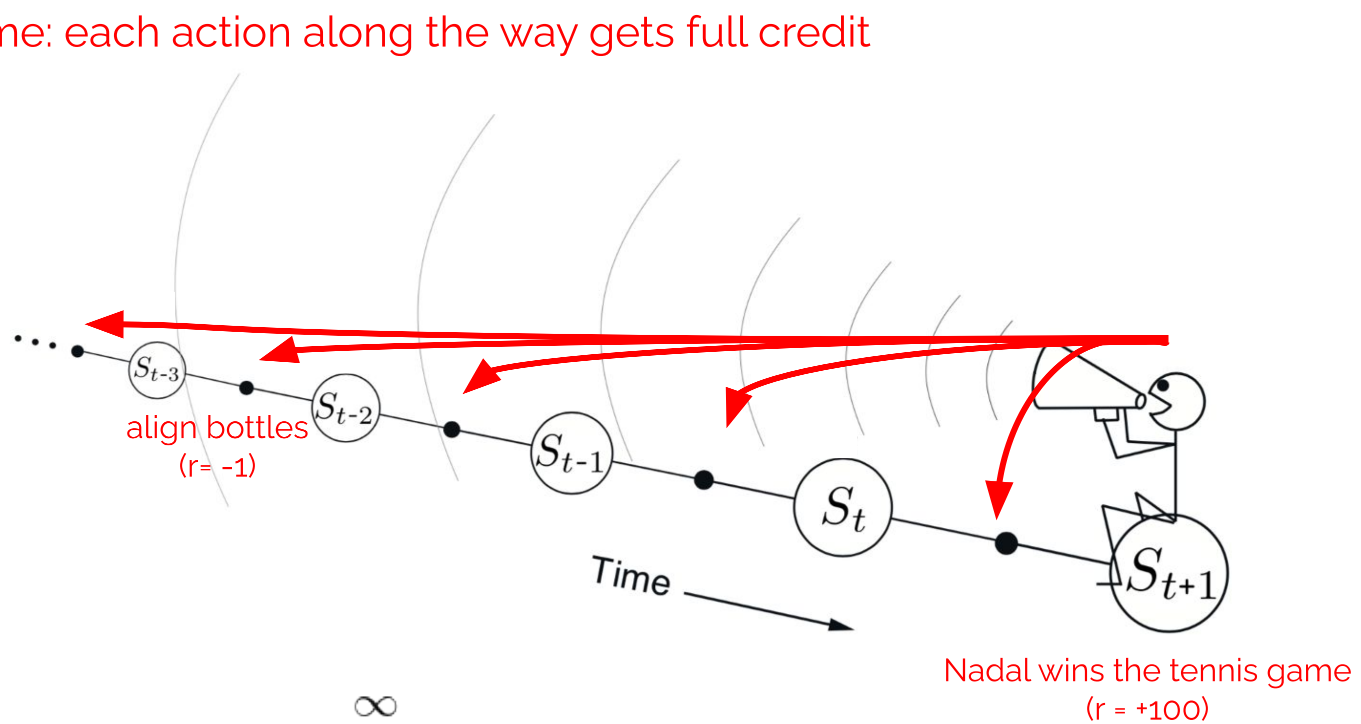


Credit assignment

One extreme: each action along the way gets full credit



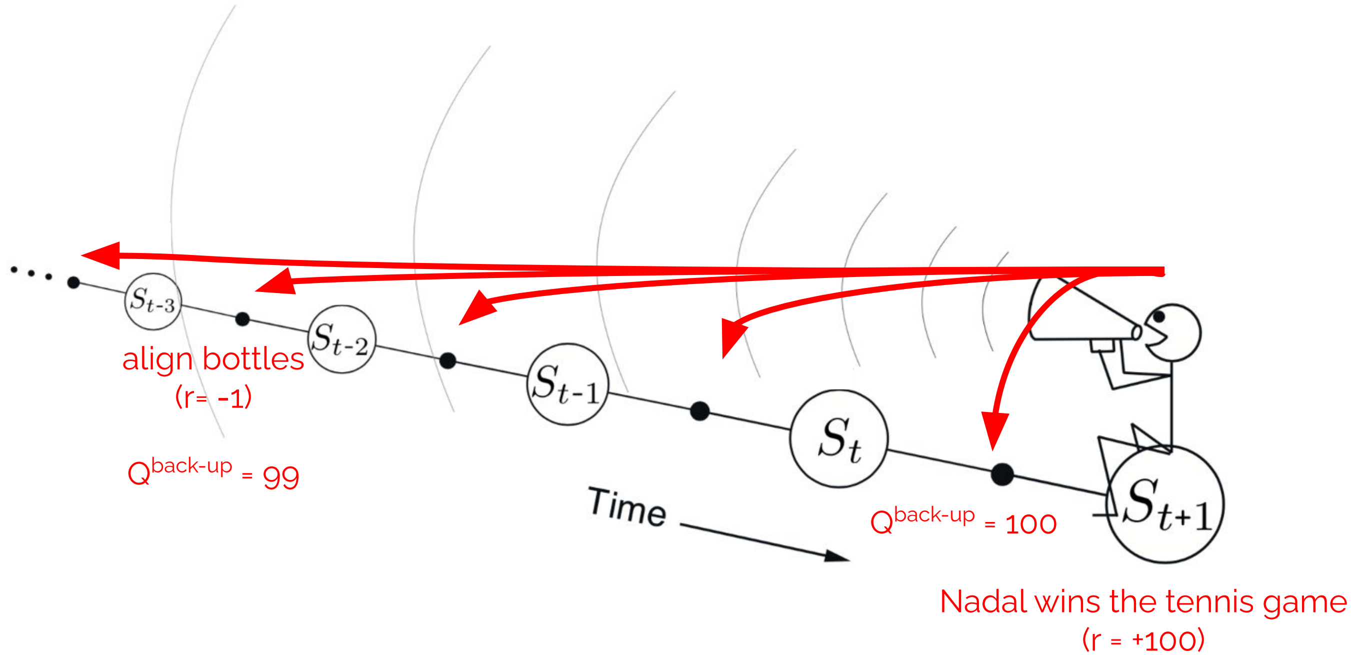
Credit assignment



$$Q^{back-up}(s_t, a_t) = \sum_{i=0}^{\infty} r_{t+i}$$

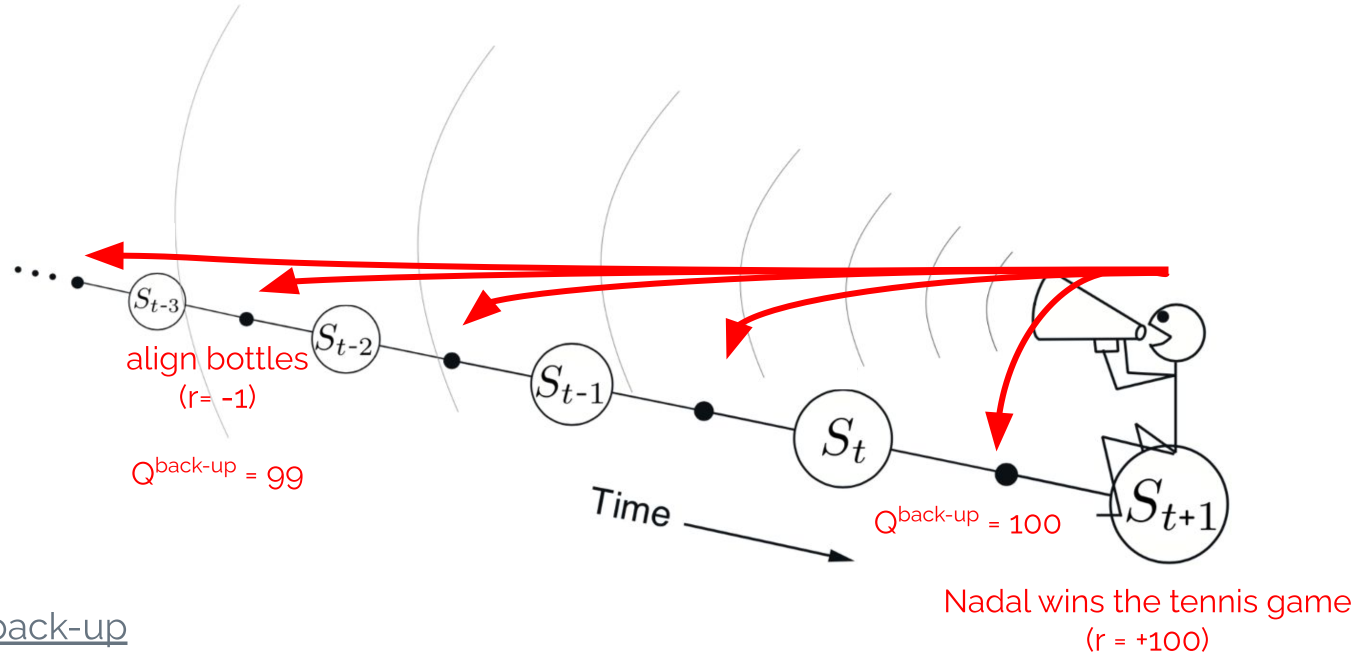
Credit assignment

One extreme: each action along the way gets full credit



Credit assignment

One extreme: each action along the way gets full credit

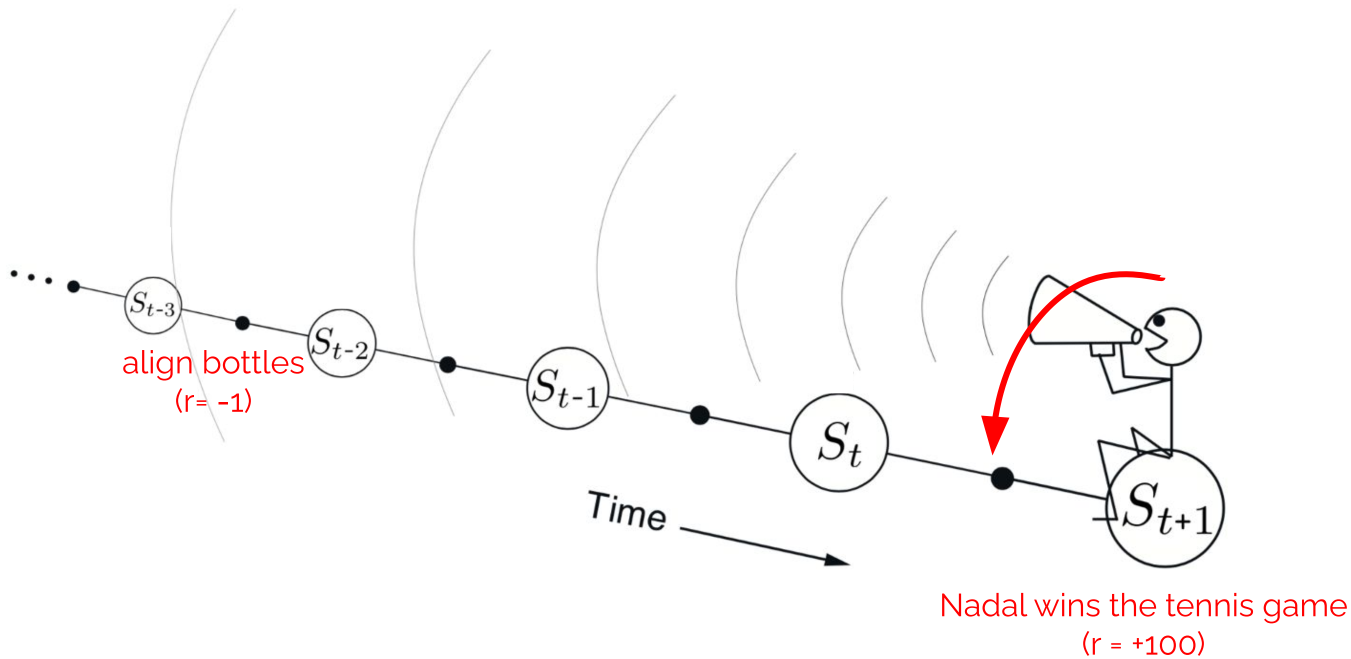


Monte Carlo back-up

+ Fast propagation.
- High variance (action may seem better or worse than it really is)

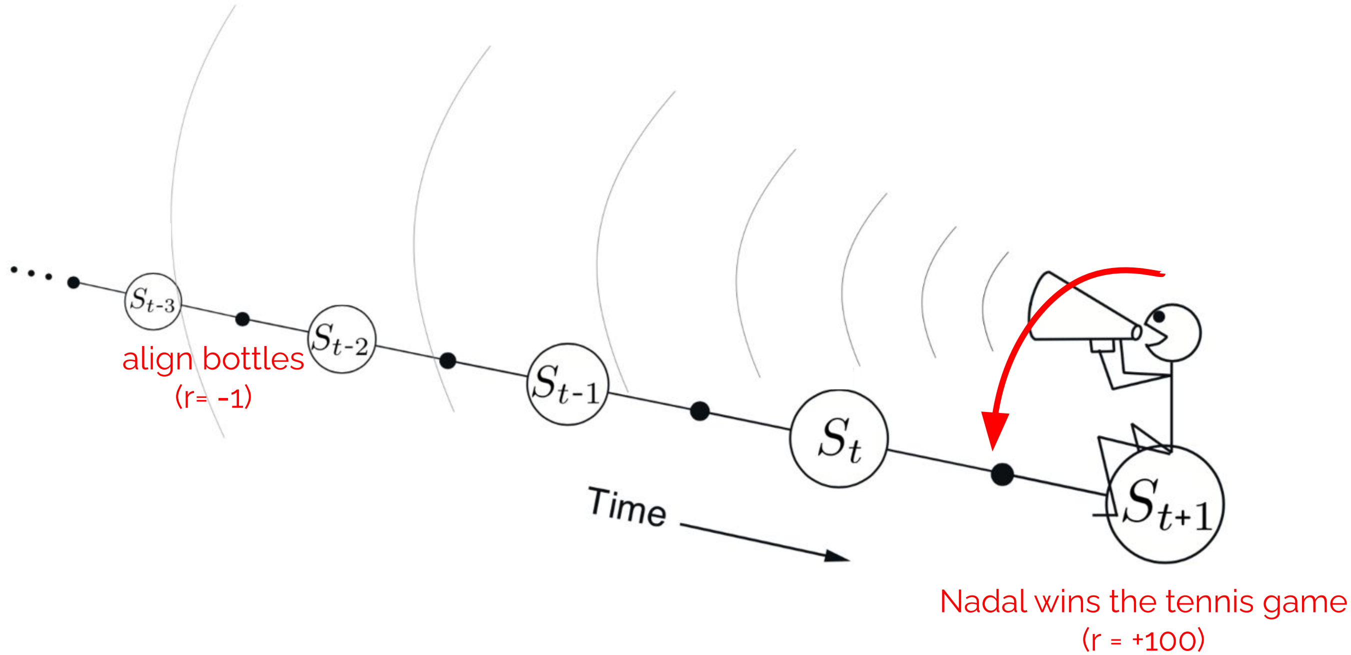
Credit assignment

Other extreme: only last action gets credit (for now).



Credit assignment

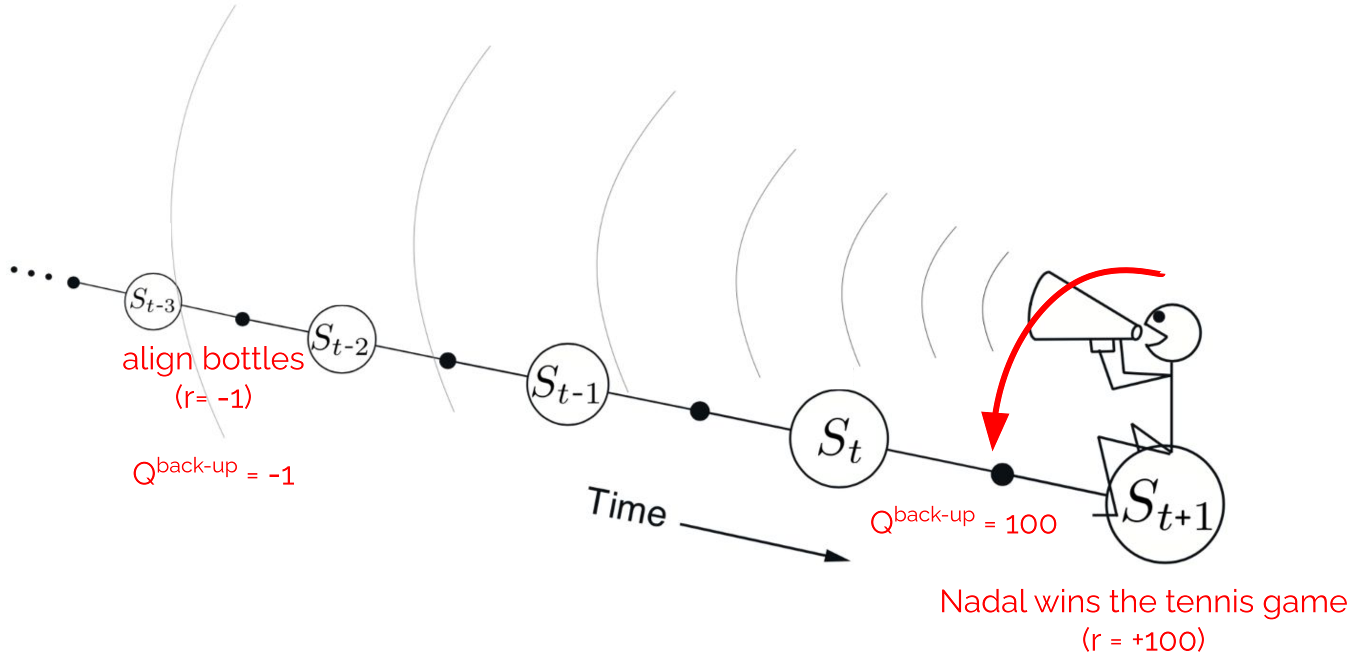
Other extreme: only last action gets credit (for now).



$$Q^{back-up}(s_t, a_t) = r_t + Q(s_{t+1}, a_{t+1})$$

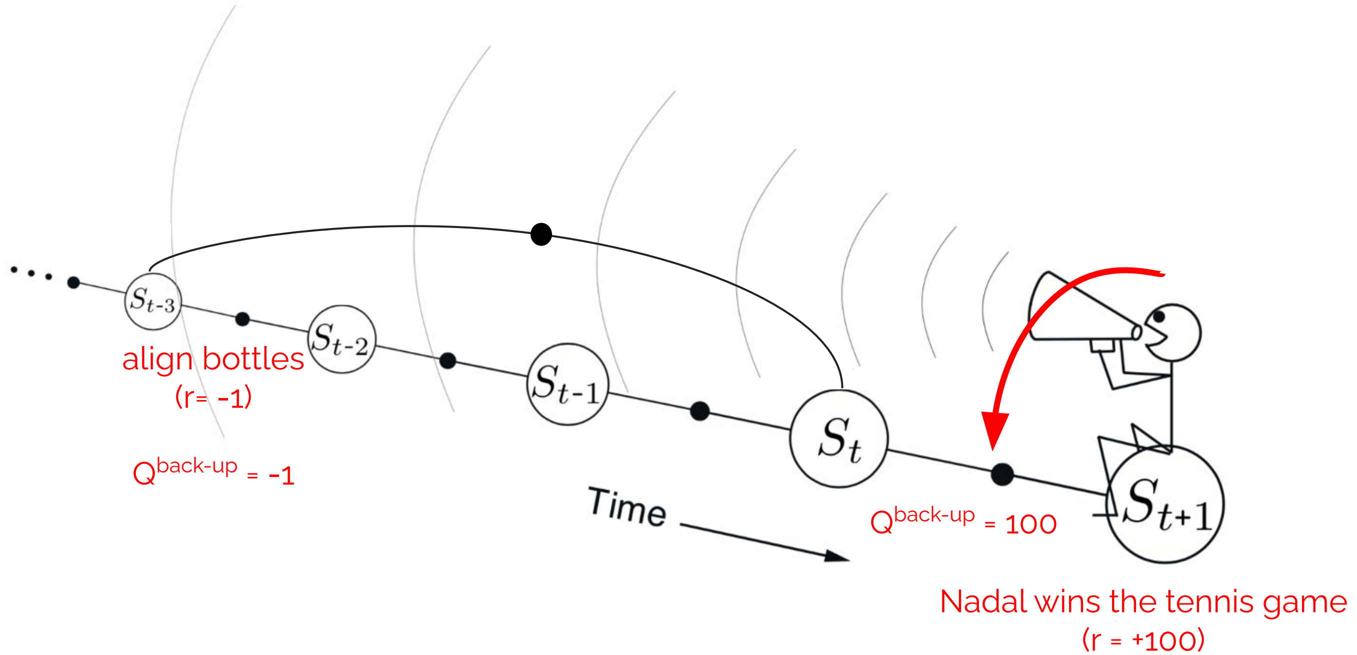
Credit assignment

Other extreme: only last action gets credit (for now).



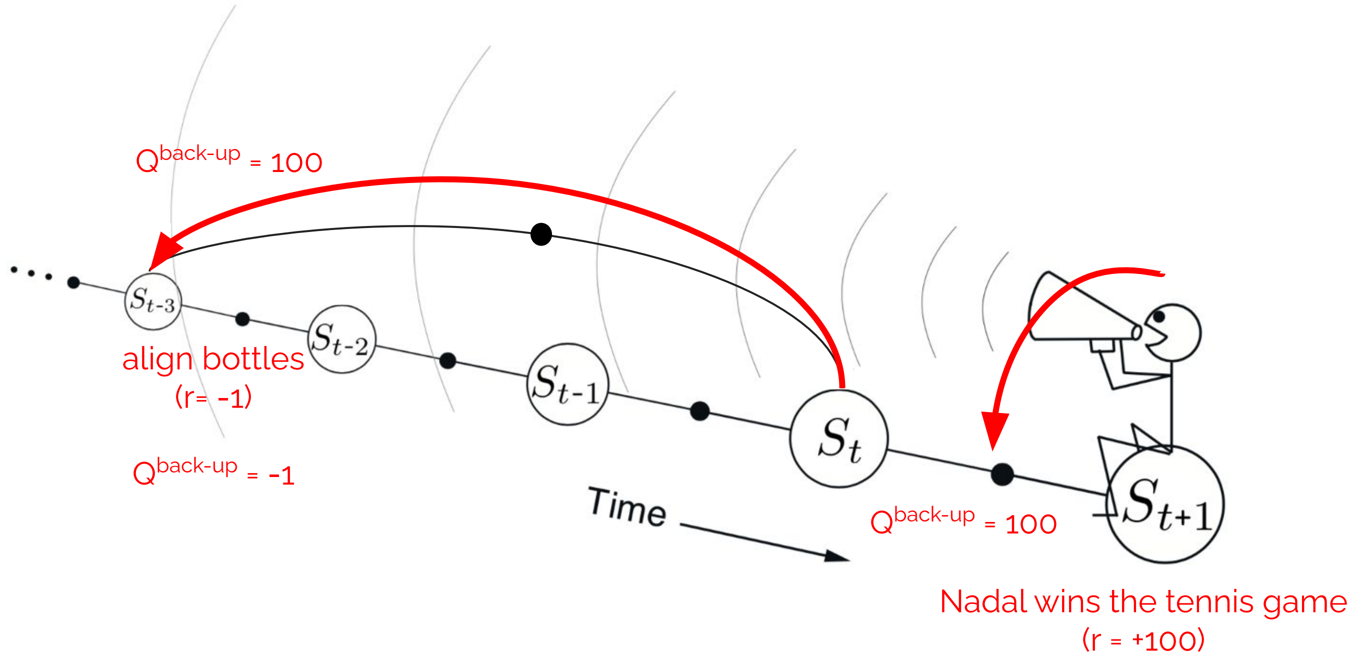
Credit assignment

Other extreme: only last action gets credit (for now).



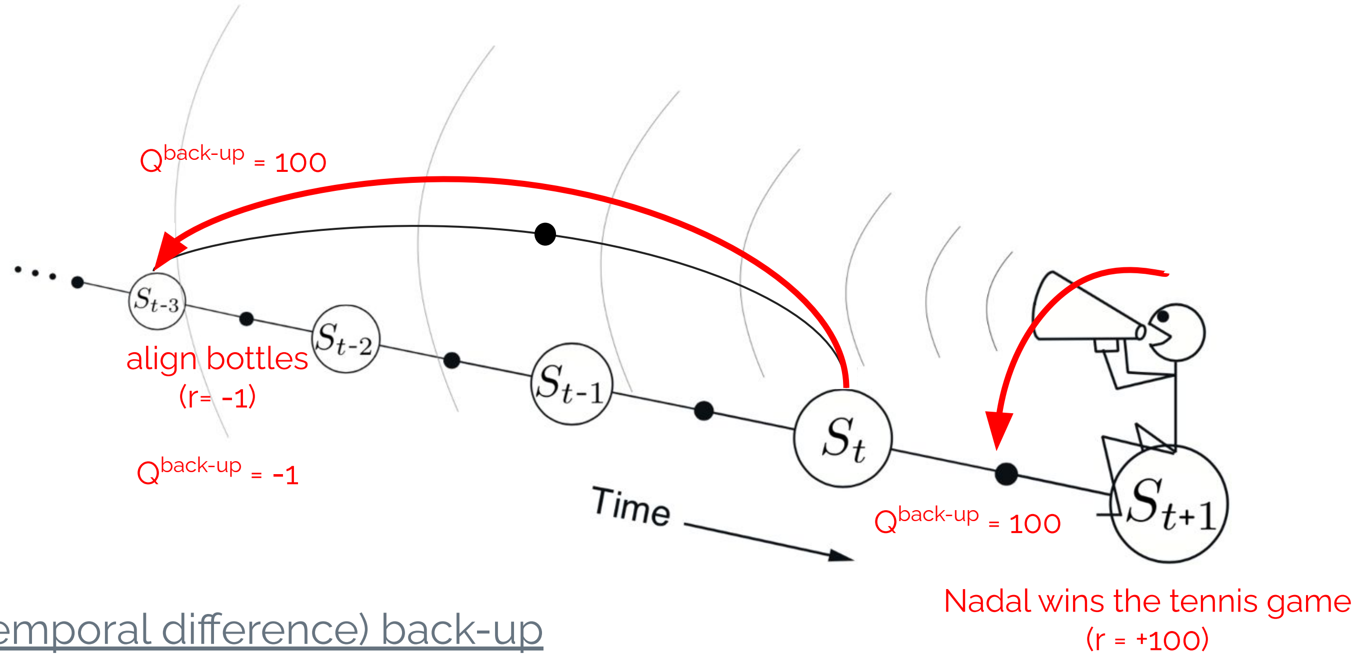
Credit assignment

Other extreme: only last action gets credit (for now).



Credit assignment

Other extreme: only last action gets credit (for now).



One-step (temporal difference) back-up

+ Low variance.
- Slow propagation.

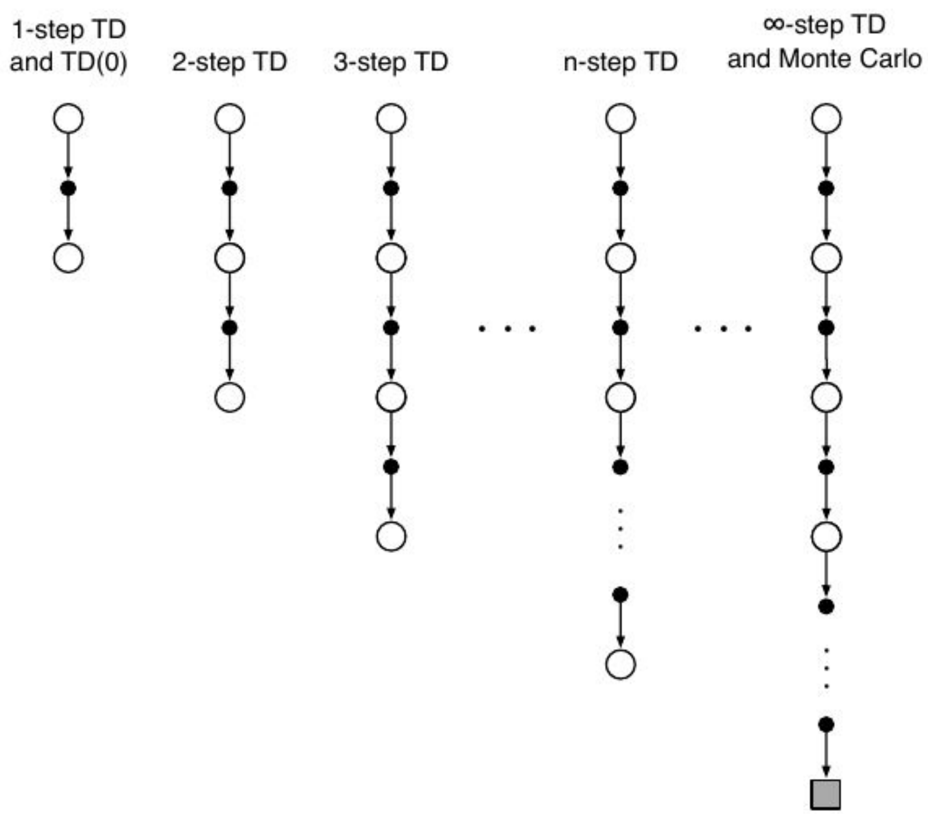
Credit assignment

Credit assignment

Spectrum of back-up estimators

Credit assignment

Spectrum of back-up estimators



The Reinforcement Learning Cycle

Pseudocode

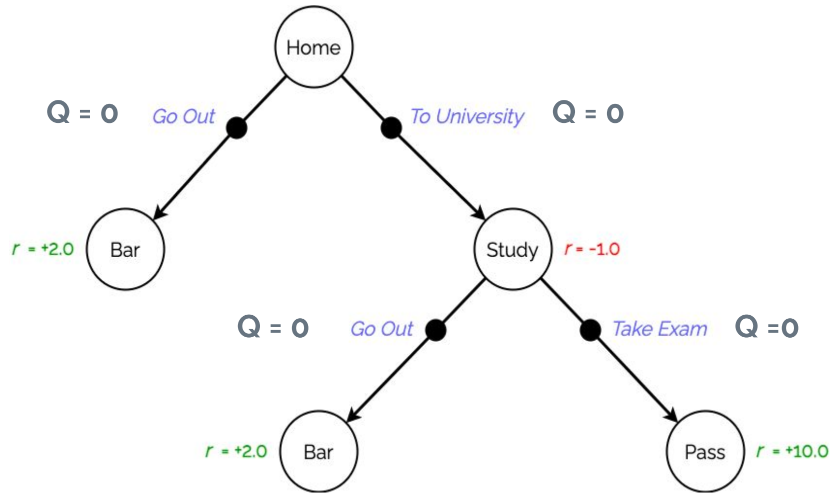
Initialize **Q(s,a)** estimates for all states,actions (e.g. to 0)

Repeat:

1) Exploration: Sample a sequence of actions.

2) Credit assignment: Compute new value estimates $\mathbf{Q^{back\text{-}up}(s,a)}$ for all actions along the path.

3) Update: Adjust our **Q(s,a)** solution based on the new back-up estimates $\mathbf{Q^{back\text{-}up}(s,a)}$.



The Reinforcement Learning Cycle

Pseudocode

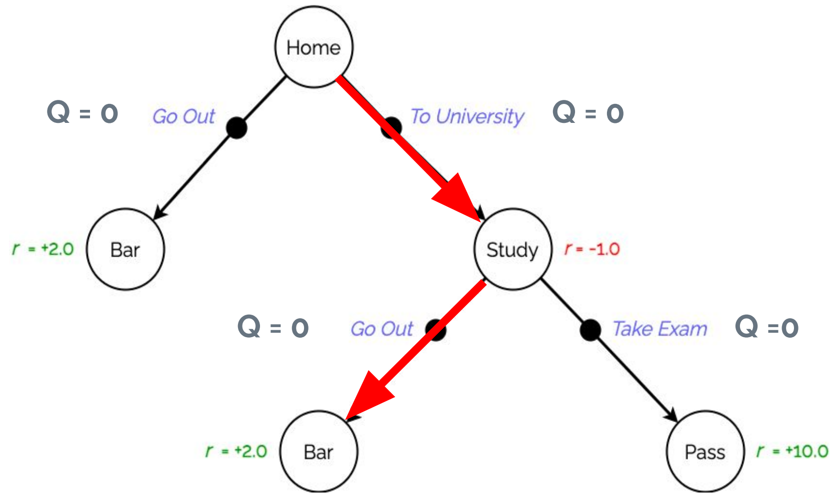
Initialize **Q(s,a)** estimates for all states,actions (e.g. to 0)

Repeat:

1) **Exploration**: Sample a sequence of actions.

2) Credit assignment: Compute new value estimates $\mathbf{Q^{back\text{-}up}(s,a)}$ for all actions along the path.

3) Update: Adjust our **Q(s,a)** solution based on the new back-up estimates $\mathbf{Q^{back\text{-}up}(s,a)}$.

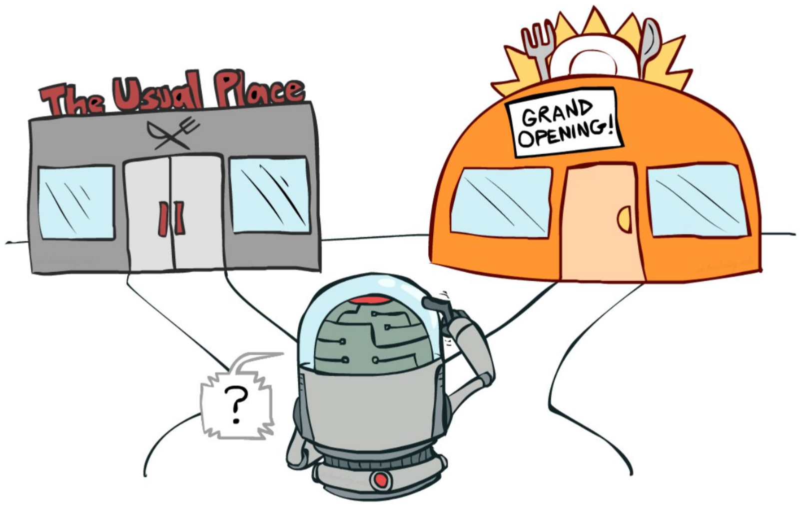


Part III C

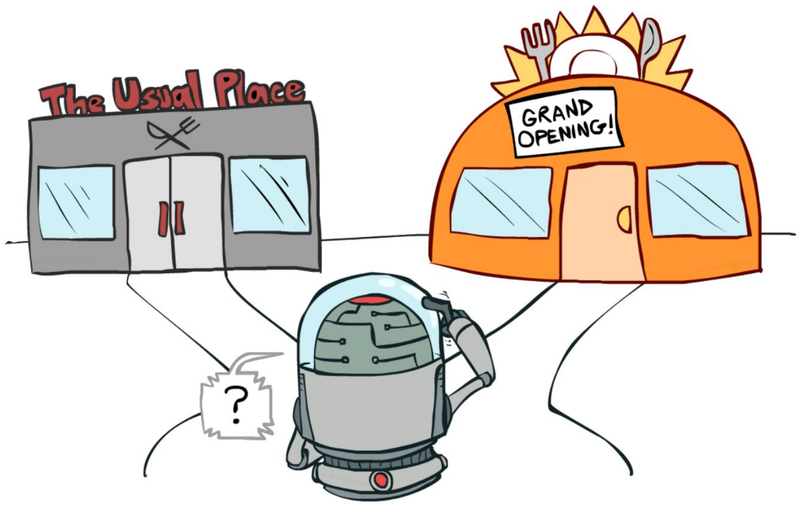
Exploration

Exploration versus exploitation

Exploration versus exploitation

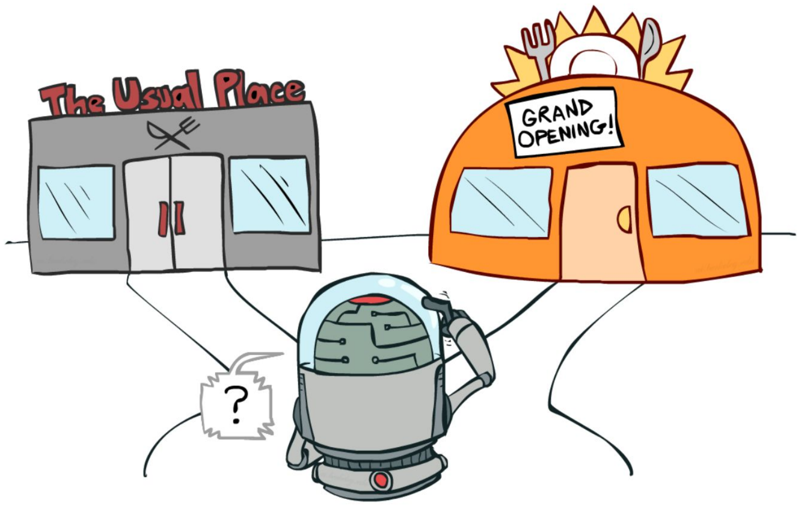


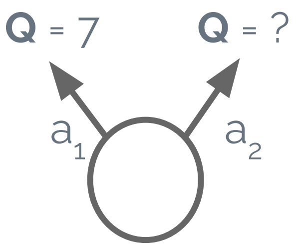
Exploration versus exploitation



Question: The usual place scores a 7/10 on average. Which place would you choose?

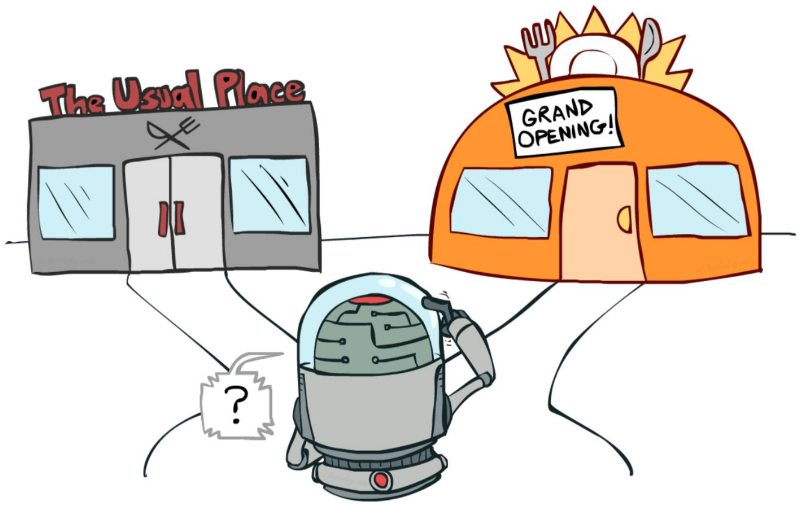
Exploration versus exploitation





Question: The usual place scores a 7/10 on average. Which place would you choose?

Exploration versus exploitation



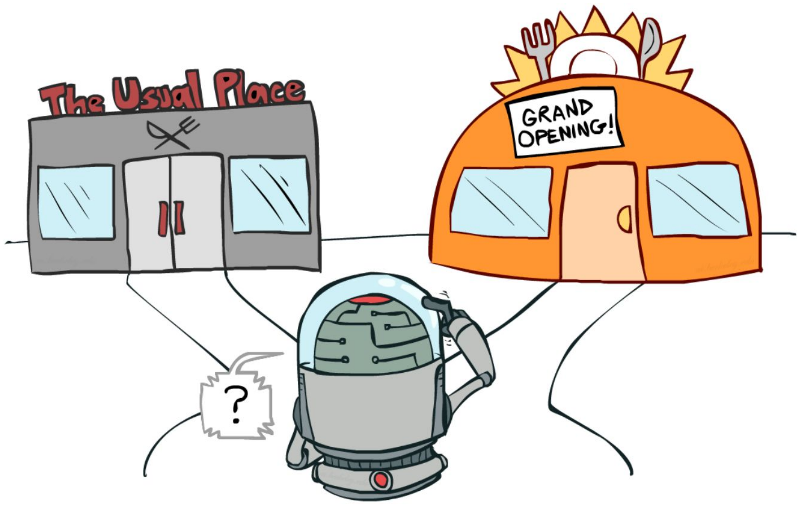
<u>Exploitation</u>

(commit to the current best option)

<u>Exploration</u>

(try something which is new or – currently – seems suboptimal)

Exploration versus exploitation



Exploitation

(commit to the current best option)

Exploration

(try something which is new or – currently – seems suboptimal)

We actually need to balance both

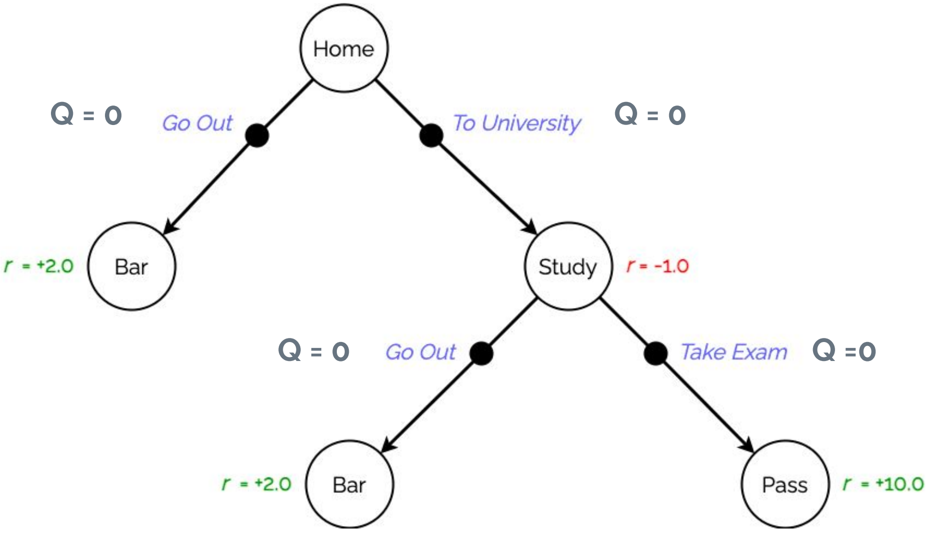
Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

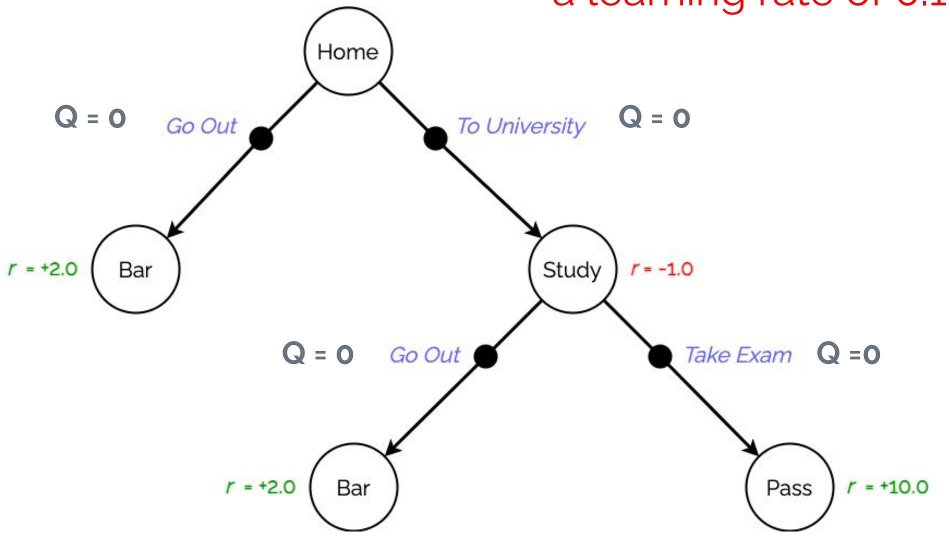


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

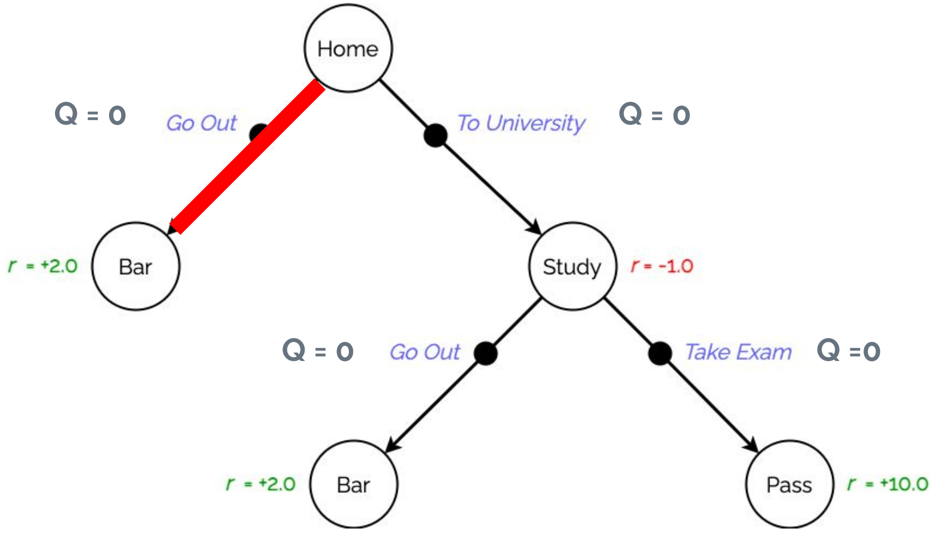
We will use Monte Carlo back-ups and a learning rate of 0.1



Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

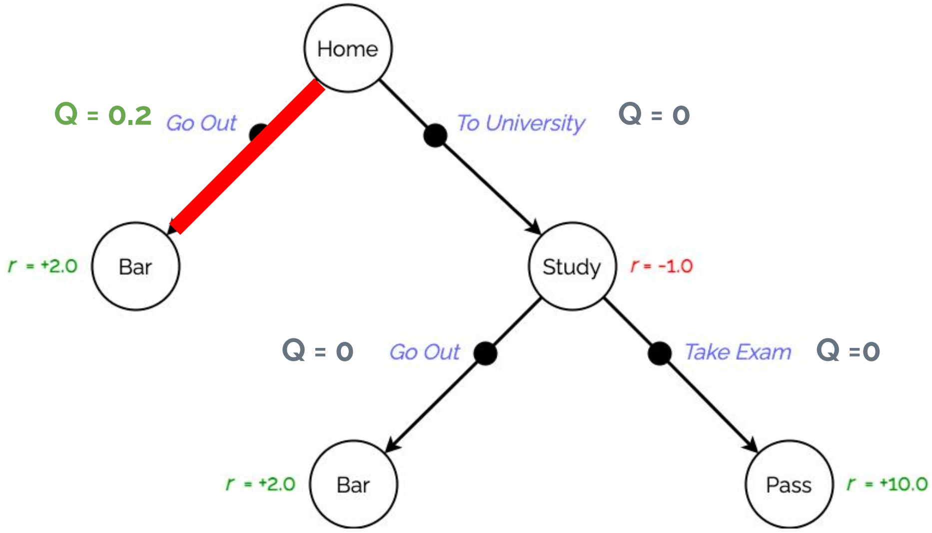
<u>Reasons</u>:



Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

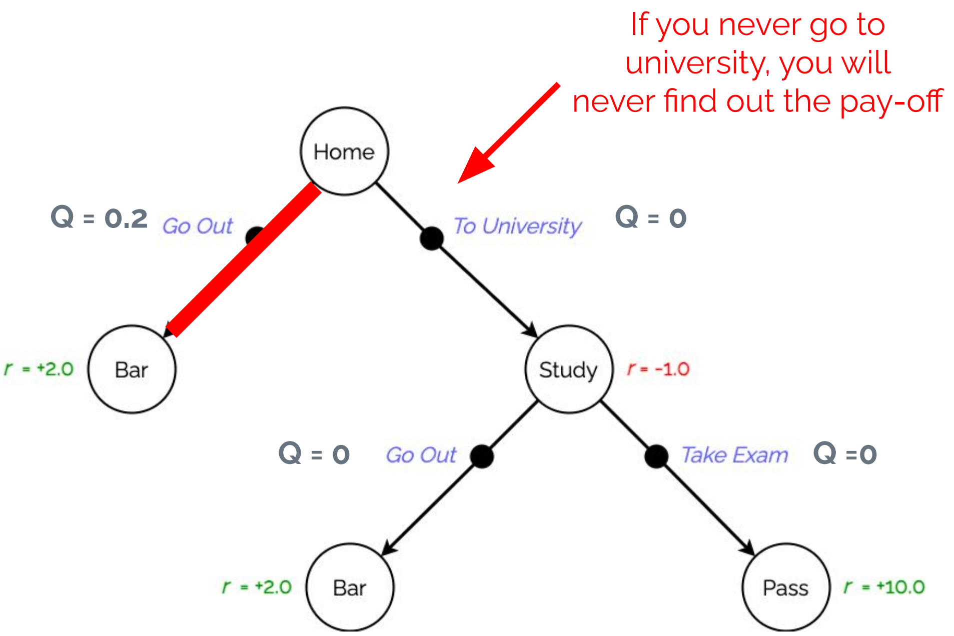
Reasons:



Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

<u>Reasons</u>:

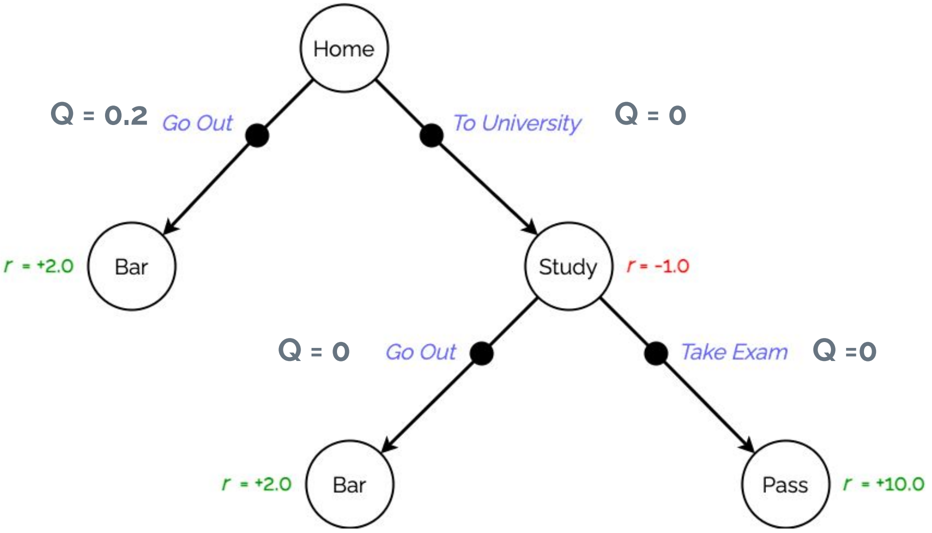


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data

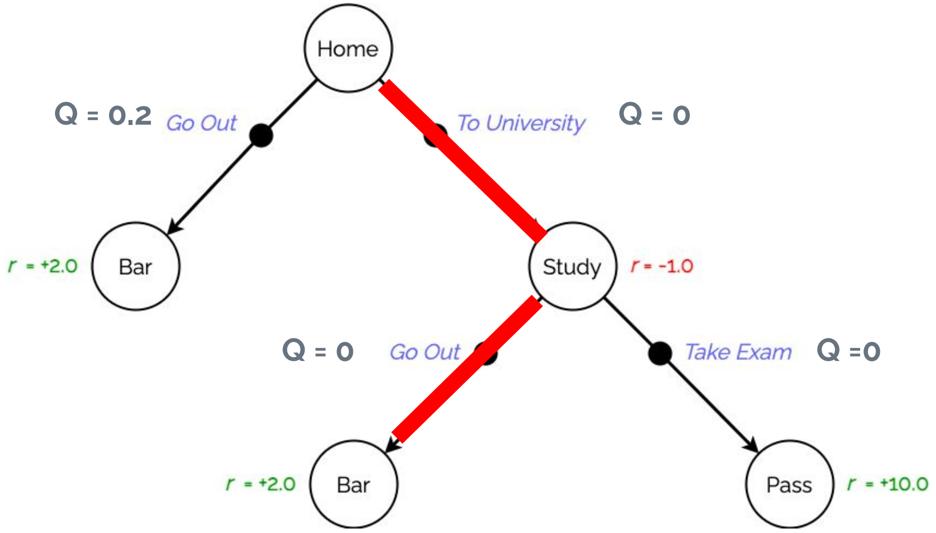


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data

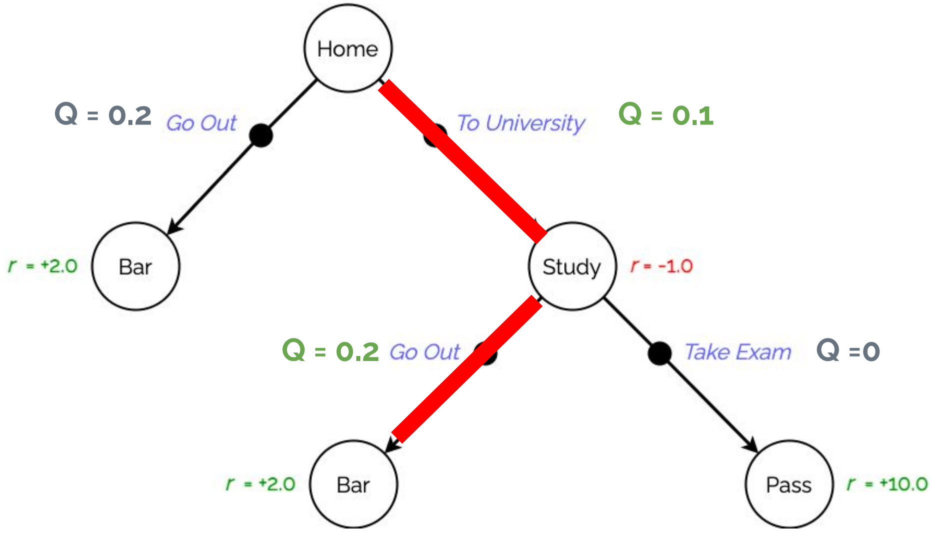


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data



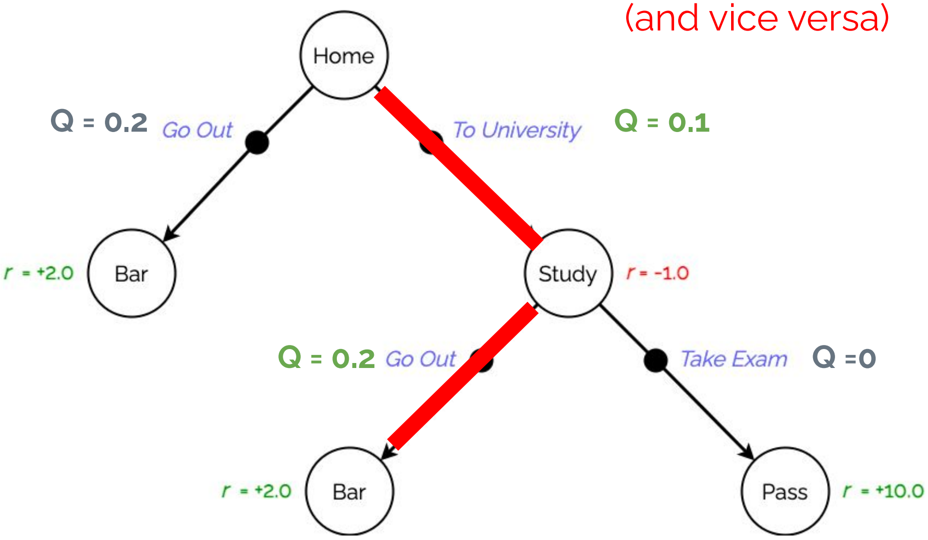
Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

<u>Reasons</u>:

1. We need to collect our own data

A good decision may seem bad if it is followed by bad decisions (and vice versa)

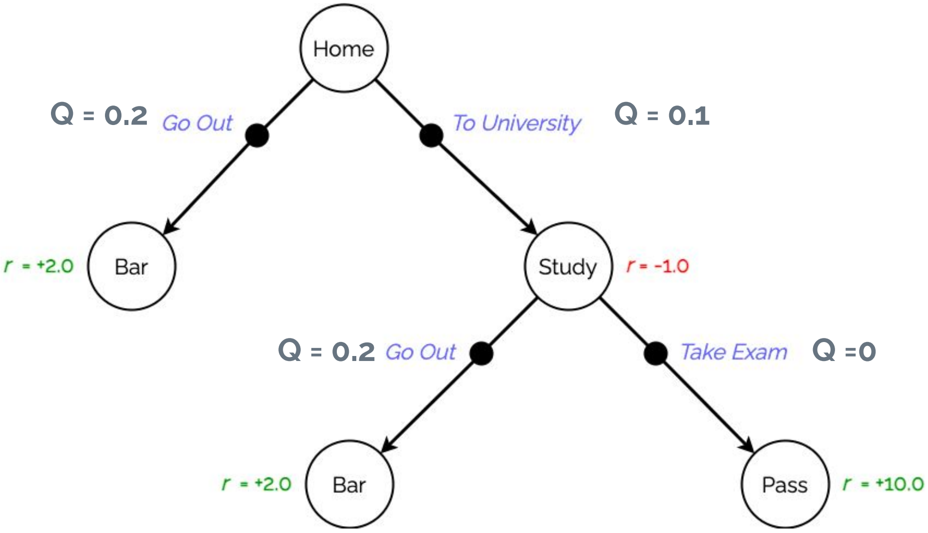


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions

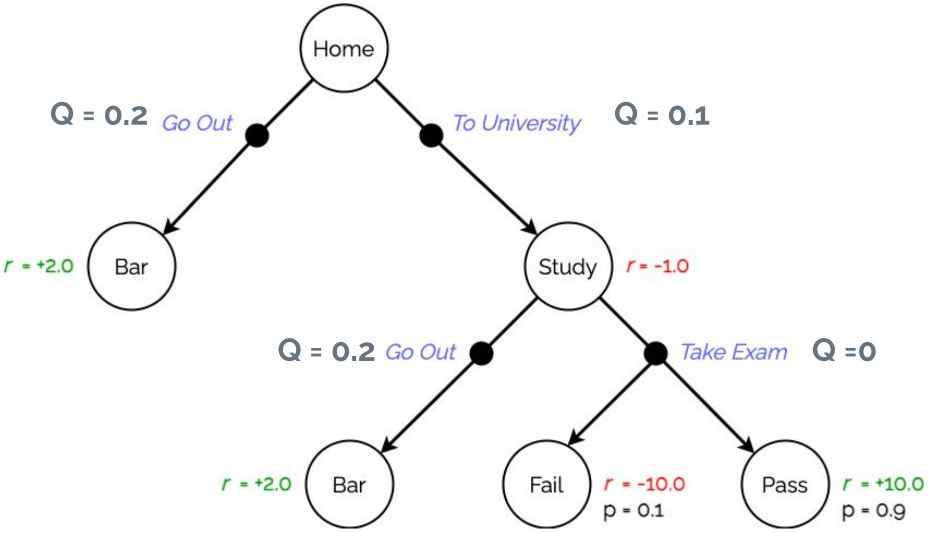


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions

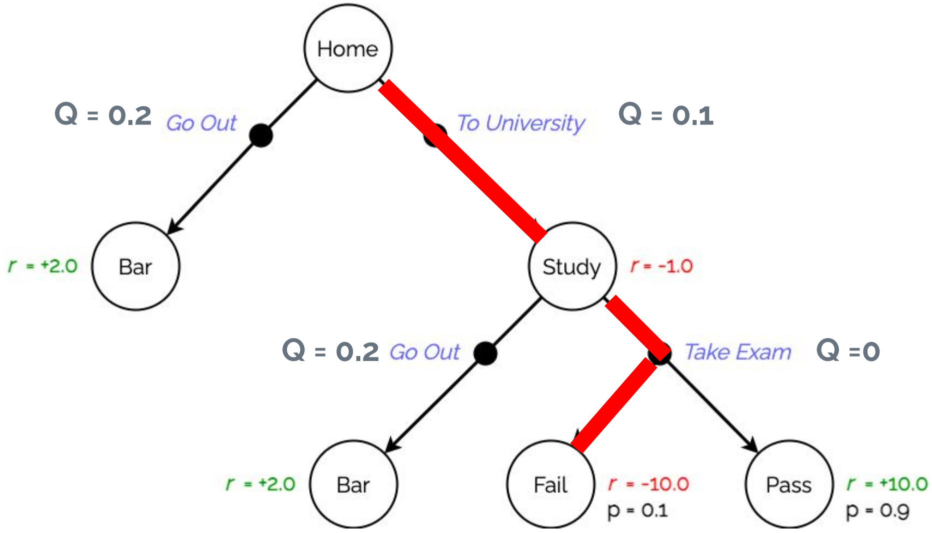


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions

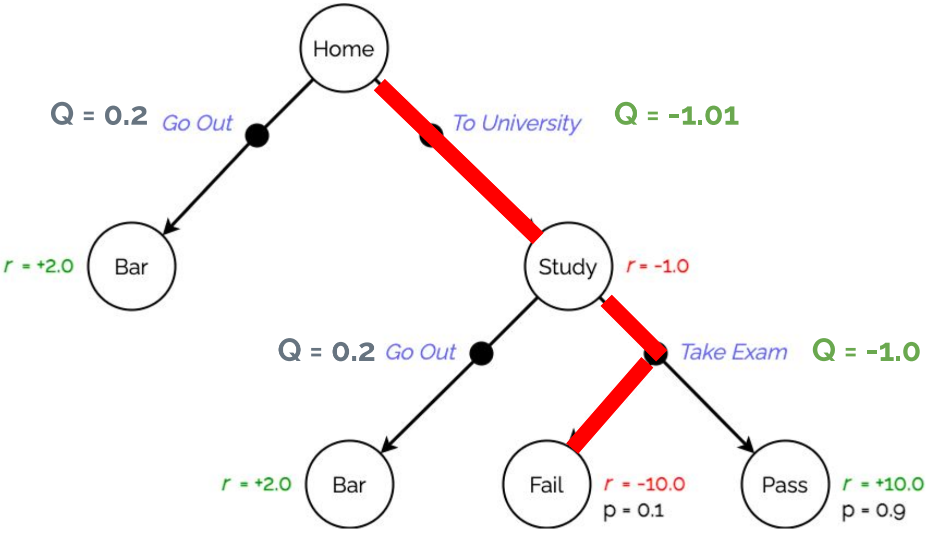


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions



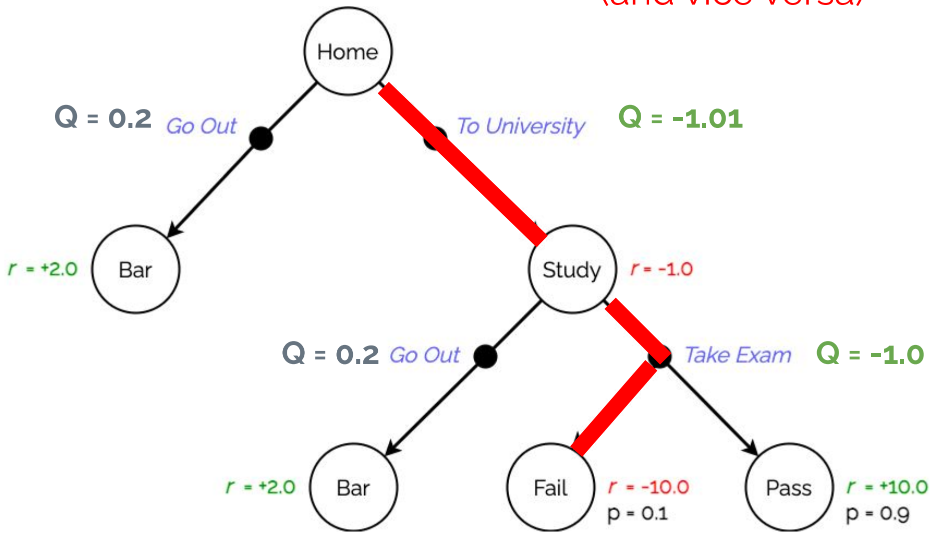
Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions

Action may seem overly bad due to bad luck (and vice versa)

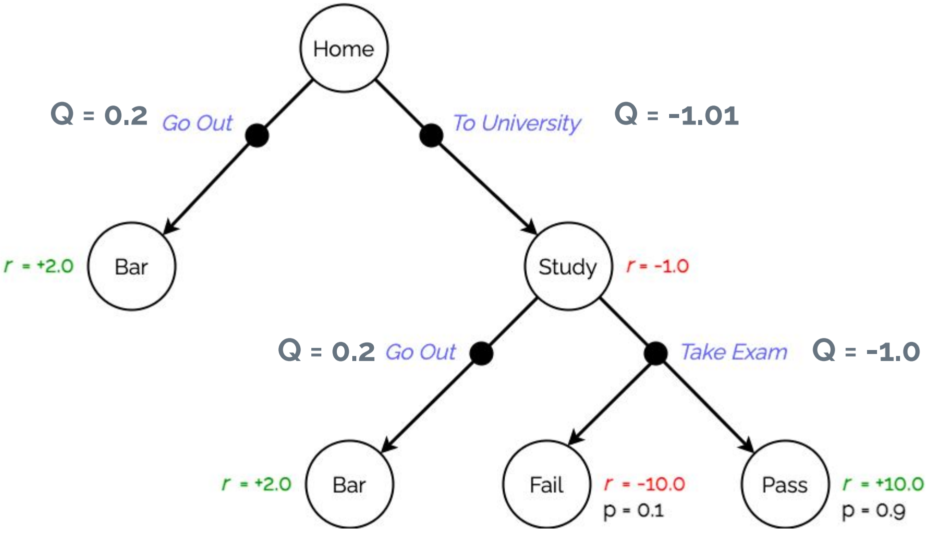


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions
3. Environment can be stochastic



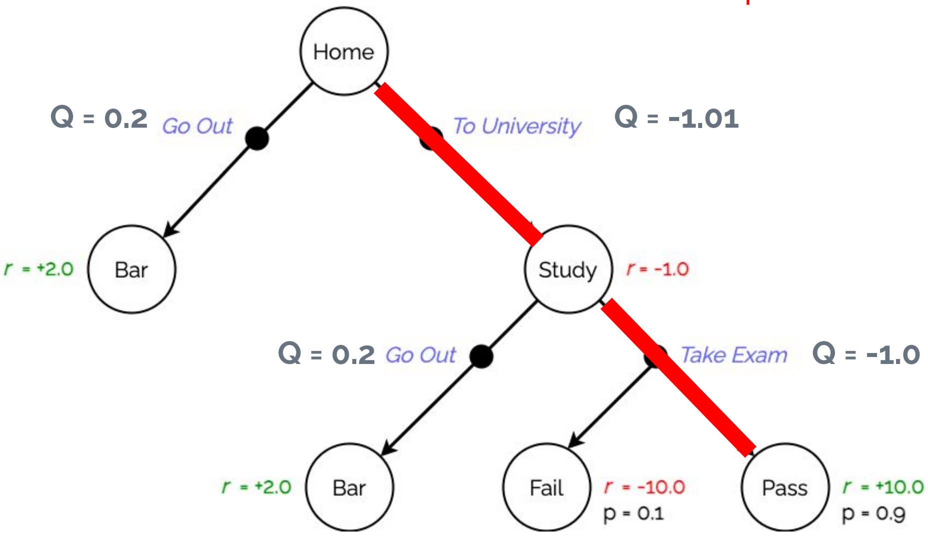
Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

<u>Reasons</u>:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions
3. Environment can be stochastic

You still persist...

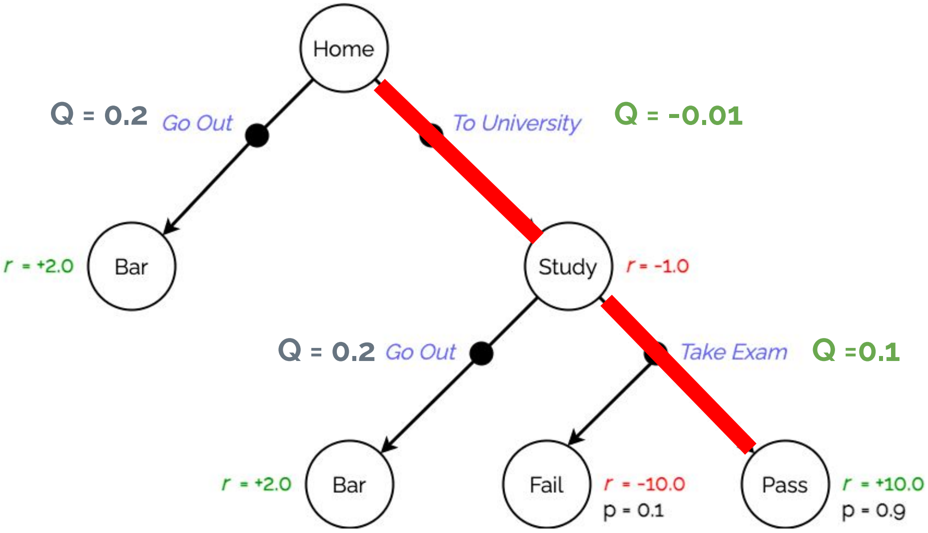


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions
3. Environment can be stochastic

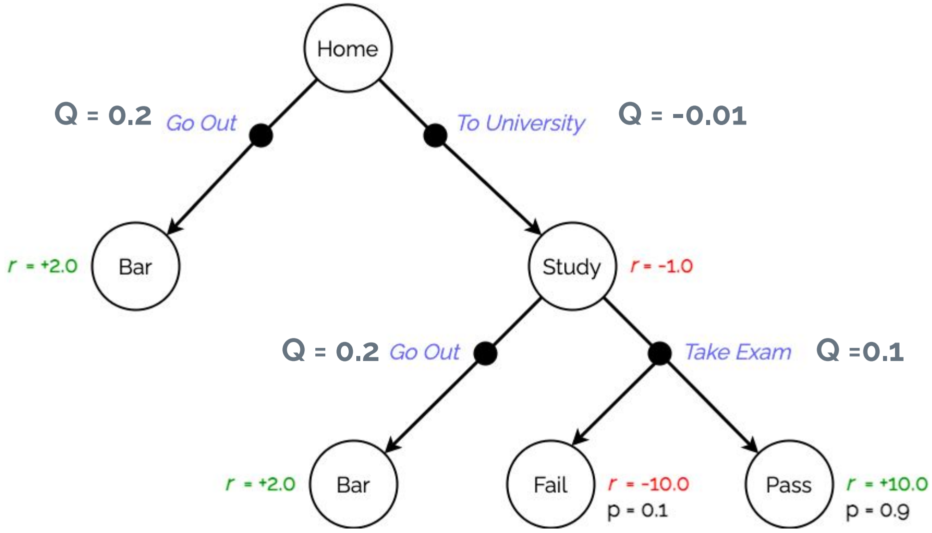


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions
3. Environment can be stochastic

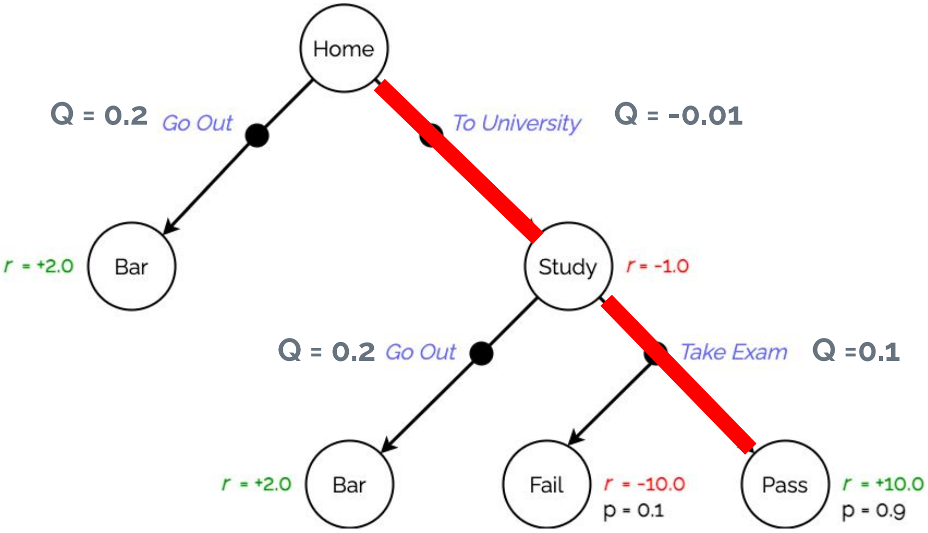


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions
3. Environment can be stochastic

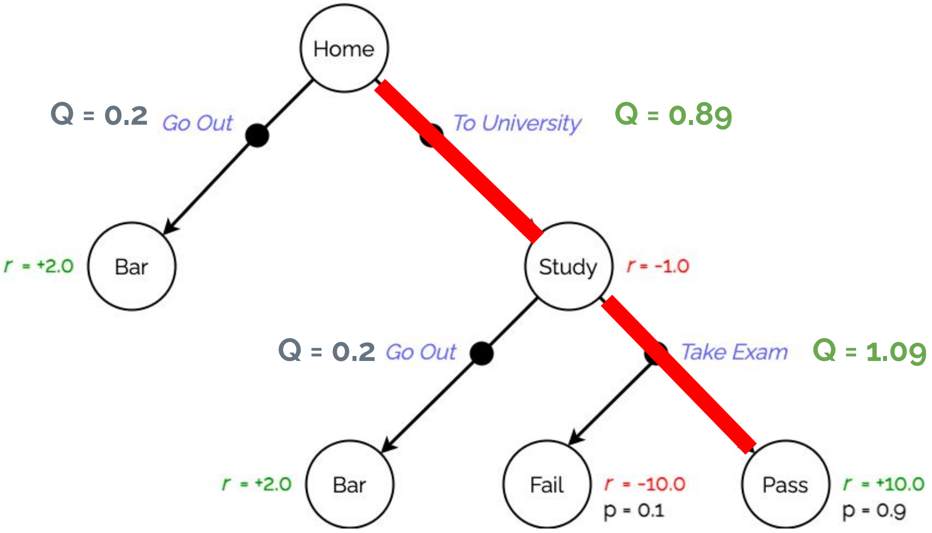


Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions
3. Environment can be stochastic



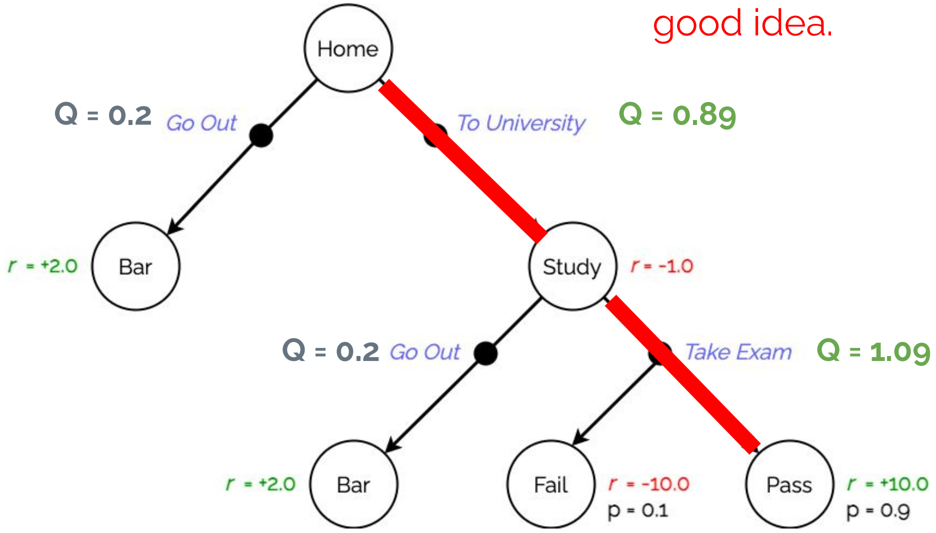
Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions
3. Environment can be stochastic

… and gradually find out coming to university is a good idea.



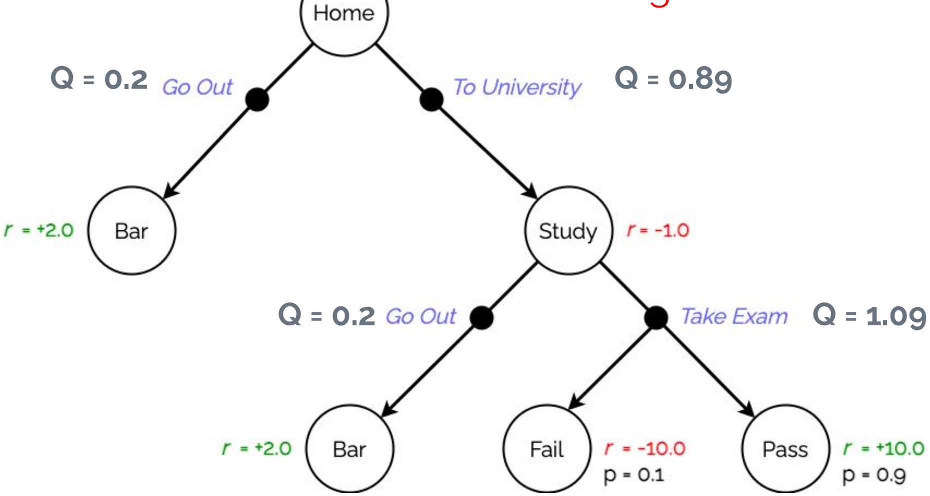
Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

<u>Reasons</u>:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions
3. Environment can be stochastic

… and gradually find out coming to university is a good idea.



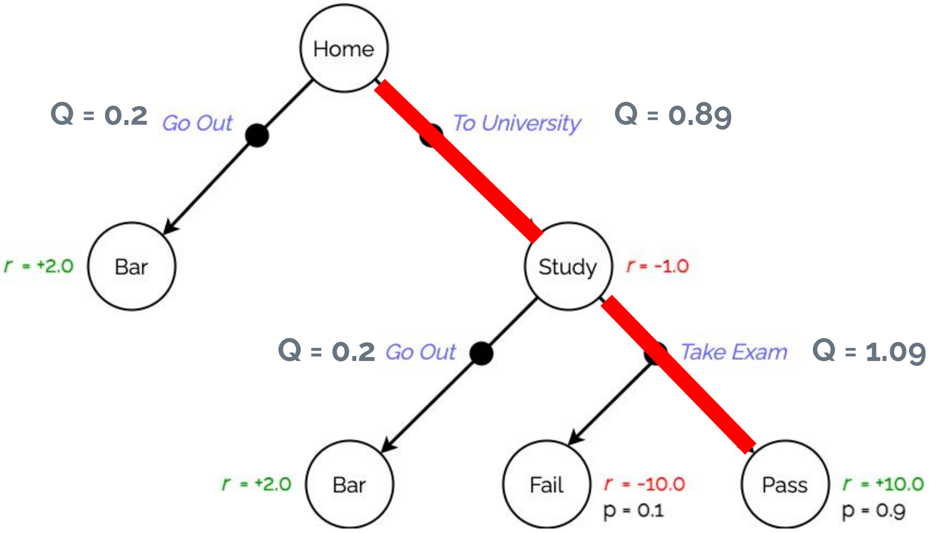
We also need **exploitation**.

Exploration/Exploitation trade-off

We need **exploration** because actions may look worse than they are.

Reasons:

1. We need to collect our own data
2. Good action may seem bad if followed by bad actions
3. Environment can be stochastic



We also need **exploitation**.

Reasons:

1. Want to use what we learned
2. In bigger problems: move in promising directions to further explore.

Exploration/Exploitation strategies

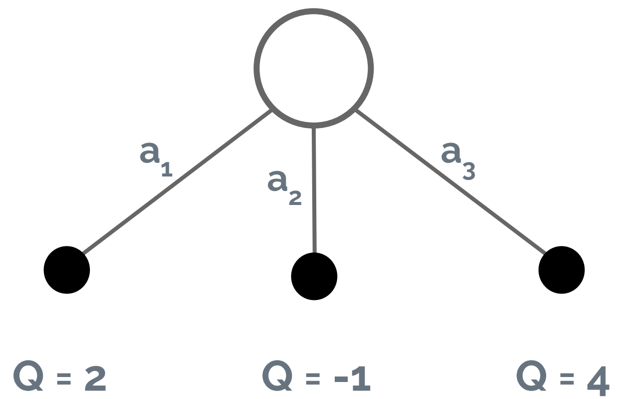
Exploration/Exploitation strategies

Huge amount of strategies, we will here discuss one (simple) example:

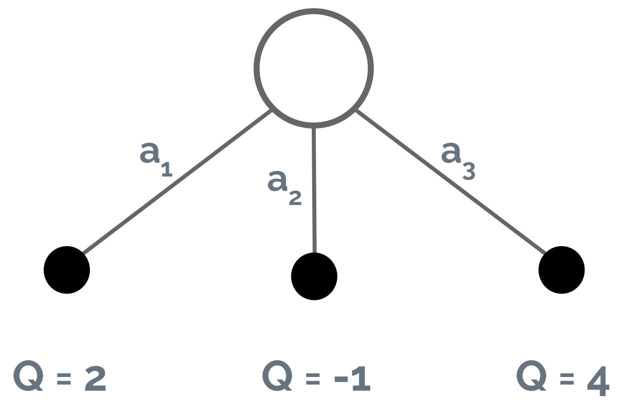
Boltzmann (softmax) exploration

Boltzmann (softmax) exploration

Boltzmann (softmax) exploration



Boltzmann (softmax) exploration



Intuition: give all actions a chance (exploration), but actions with higher Q-estimate deserve a higher probability (exploitation).

Boltzmann (softmax) exploration

Boltzmann (softmax) exploration

$$\pi(a_i|s) = \frac{e^{Q(s,a_i)/\tau}}{\sum_{a\in\mathcal{A}} e^{Q(s,a)/\tau}}$$

Boltzmann (softmax) exploration

To get the probability of selecting action a_i in state s...

$$\pi(a_i|s) = \frac{e^{Q(s,a_i)/\tau}}{\sum_{a \in \mathcal{A}} e^{Q(s,a)/\tau}}$$

Boltzmann (softmax) exploration

To get the probability of selecting action a_i in state s…

… we exponentiate its Q-value…

$$\pi(a_i|s) = \frac{e^{Q(s,a_i)/\tau}}{\sum_{a \in \mathcal{A}} e^{Q(s,a)/\tau}}$$

Boltzmann (softmax) exploration

To get the probability of selecting action a_i in state s…

… we exponentiate its Q-value…

$$\pi(a_i|s) = \frac{e^{Q(s,a_i)/\tau}}{\sum_{a \in \mathcal{A}} e^{Q(s,a)/\tau}}$$

… and normalize over the sum of exponentiated Q-values of all actions (to make it a valid probability distribution).

Boltzmann (softmax) exploration

To get the probability of selecting action a_i in state s…

… we exponentiate its Q-value…

$$\pi(a_i|s) = \frac{e^{Q(s,a_i)\boxed{/\tau}}}{\sum_{a\in\mathcal{A}} e^{Q(s,a)\boxed{/\tau}}}$$

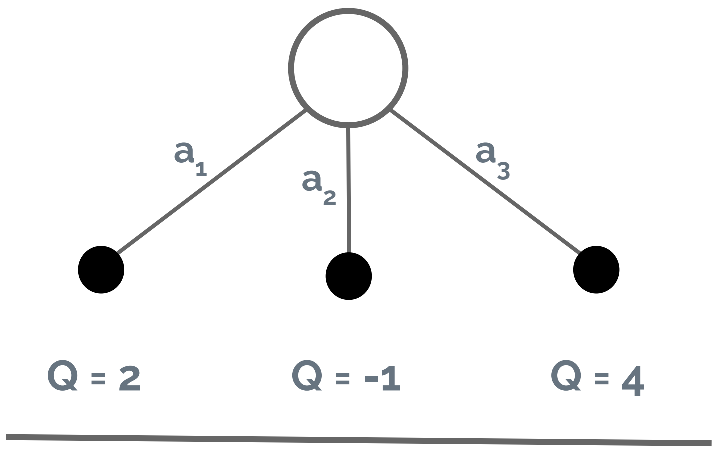
… and normalize over the sum of exponentiated Q-values of all actions (to make it a valid probability distribution).

Temperature **τ** scales the amount of exploration:

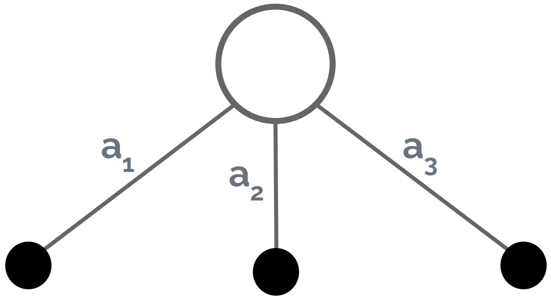
τ → 0 : one-hot (exploit)

τ → ∞ : uniform (explore)

Boltzmann (softmax) exploration



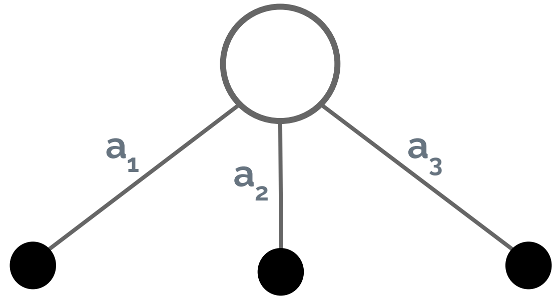
Boltzmann (softmax) exploration



Q = 2	**Q = -1**	**Q = 4**
π=0.12	π=0.01	π=0.87

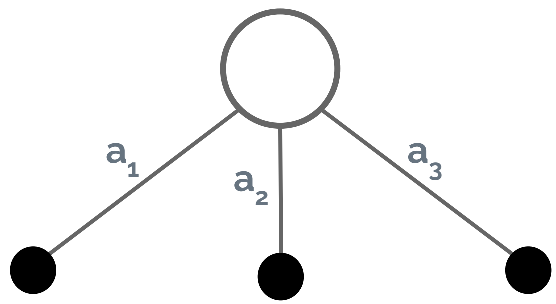
τ=1.0

Boltzmann (softmax) exploration



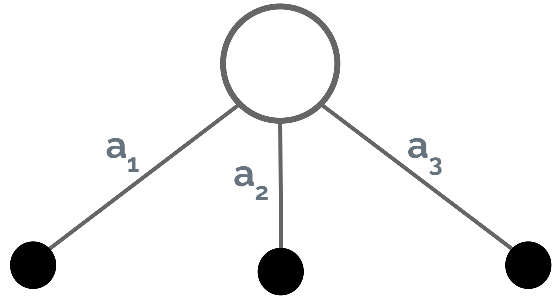
	Q = 2	Q = -1	Q = 4
τ=1.0	π=0.12	π=0.01	π=0.87
τ=0.01	π=0.0	π=0.0	π=1.0

Boltzmann (softmax) exploration



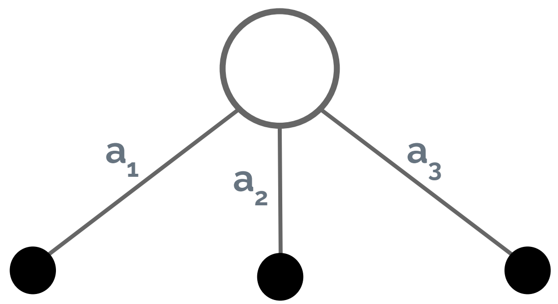
	Q = 2	Q = -1	Q = 4	
τ=1.0	π=0.12	π=0.01	π=0.87	
τ=0.01	π=0.0	π=0.0	π=1.0	**full exploitation**

Boltzmann (softmax) exploration



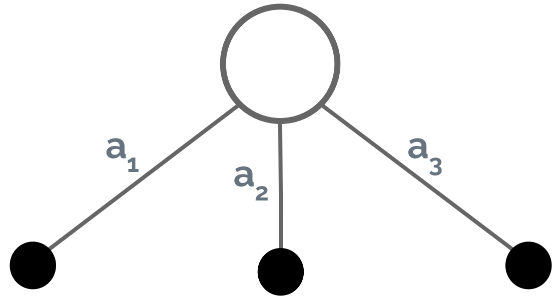
	Q = 2	Q = -1	Q = 4	
τ=1.0	π=0.12	π=0.01	π=0.87	
τ=0.01	π=0.0	π=0.0	π=1.0	**full exploitation**
τ=100	π=0.33	π=0.32	π=0.34	

Boltzmann (softmax) exploration



	Q = 2	Q = -1	Q = 4	
τ=1.0	π=0.12	π=0.01	π=0.87	
τ=0.01	π=0.0	π=0.0	π=1.0	**full exploitation**
τ=100	π=0.33	π=0.32	π=0.34	**full exploration**

Boltzmann (softmax) exploration



Can anneal **τ** during training to gradually transition from exploration to exploitation

	Q = 2	**Q = -1**	**Q = 4**	
τ=1.0	π=0.12	π=0.01	π=0.87	
τ=0.01	π=0.0	π=0.0	π=1.0	**full exploitation**
τ=100	π=0.33	π=0.32	π=0.34	**full exploration**

Exploration

(video)

The Reinforcement Learning Cycle

Pseudocode

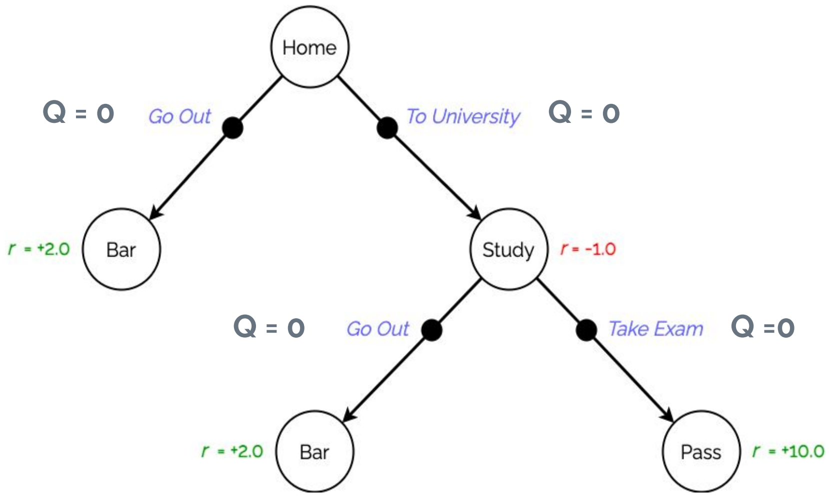
Initialize **Q(s,a)** estimates for all states,actions (e.g. to 0)

Repeat:

1) Exploration:

2) Credit assignment:

3) Update:



The Reinforcement Learning Cycle

Pseudocode

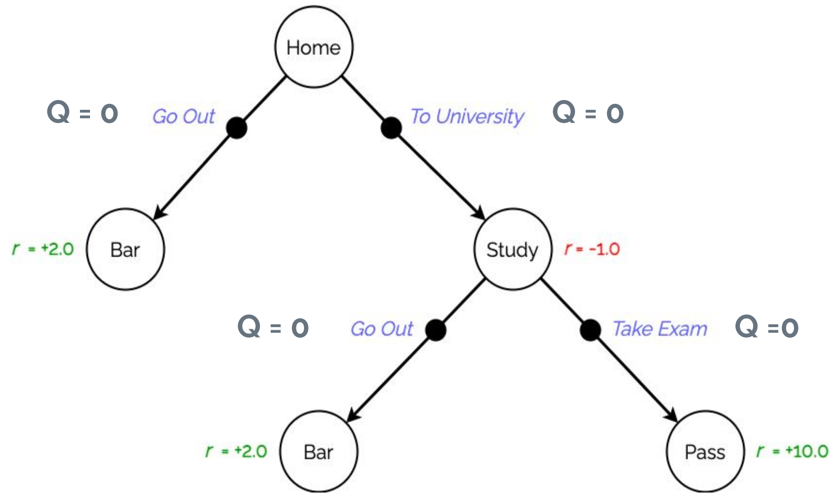
Initialize **Q(s,a**) estimates for all states,actions (e.g. to 0)

Repeat:

1) Exploration: Boltzmann policy with annealing temperature.

2) Credit assignment:

3) Update:



The Reinforcement Learning Cycle

Pseudocode

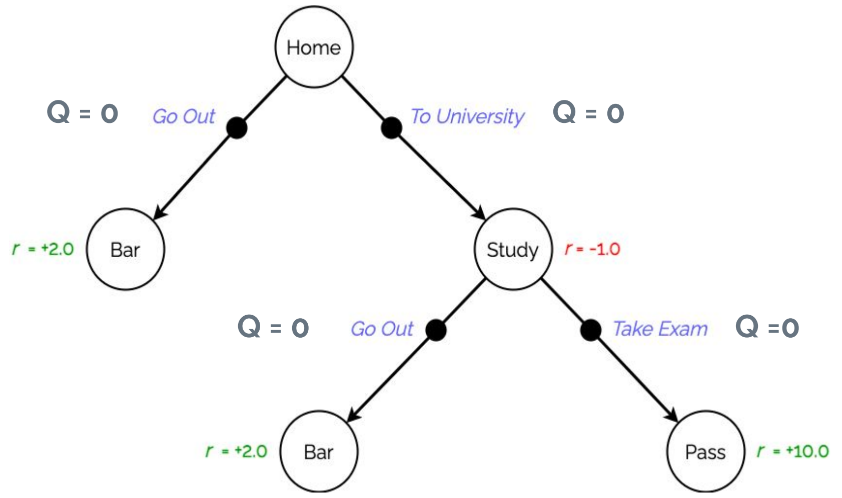
Initialize **Q(s,a)** estimates for all states,actions (e.g. to 0)

Repeat:

1) Exploration: Boltzmann policy with annealing temperature.

2) Credit assignment: Monte Carlo back-up.

3) Update:



The Reinforcement Learning Cycle

Pseudocode

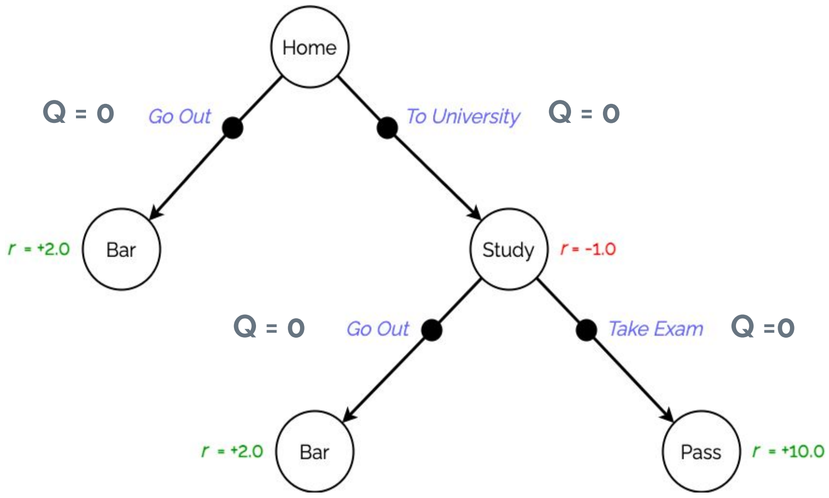
Initialize **Q(s,a**) estimates for all states,actions (e.g. to 0)

Repeat:

1) Exploration: Boltzmann policy with annealing temperature.

2) Credit assignment: Monte Carlo back-up.

3) Update: Tabular learning rule with learning rate 0.1



Part IV

Deep reinforcement learning

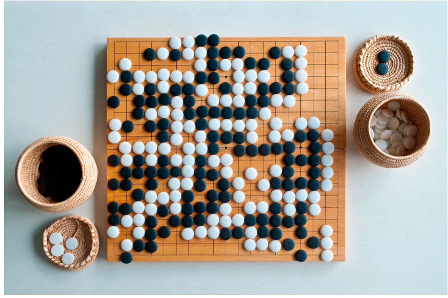
Deep reinforcement learning

Deep reinforcement learning

Deep reinforcement learning = deep learning + reinforcement learning

Deep reinforcement learning

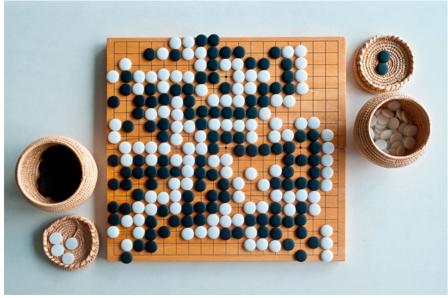
Deep reinforcement learning = deep learning + reinforcement learning



Observation spaces in reinforcement are usually high-dimensional

Deep reinforcement learning

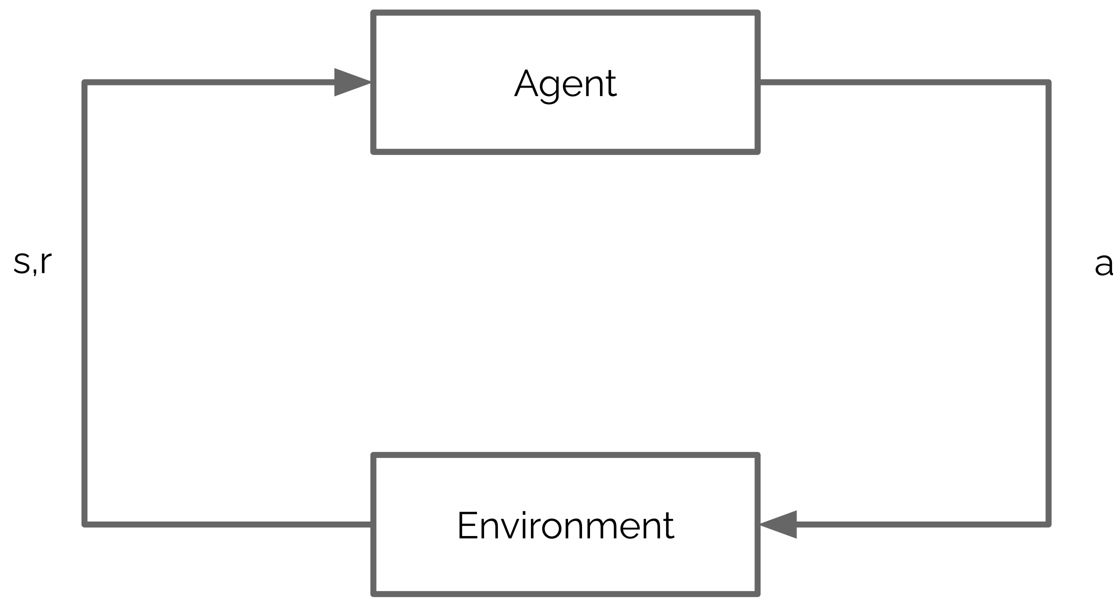
Deep reinforcement learning = deep learning + reinforcement learning



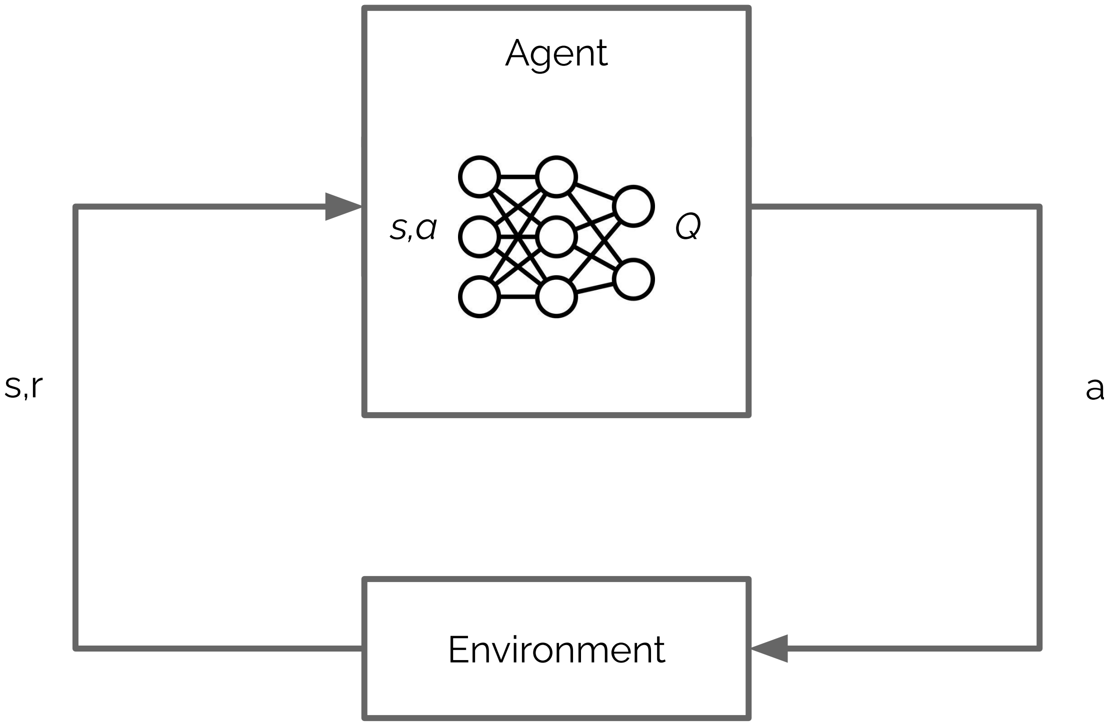
Observation spaces in reinforcement are usually high-dimensional

We need to use *function approximation*, e.g., deep learning, to store our solution
(to fit it in memory & profit from generalization)

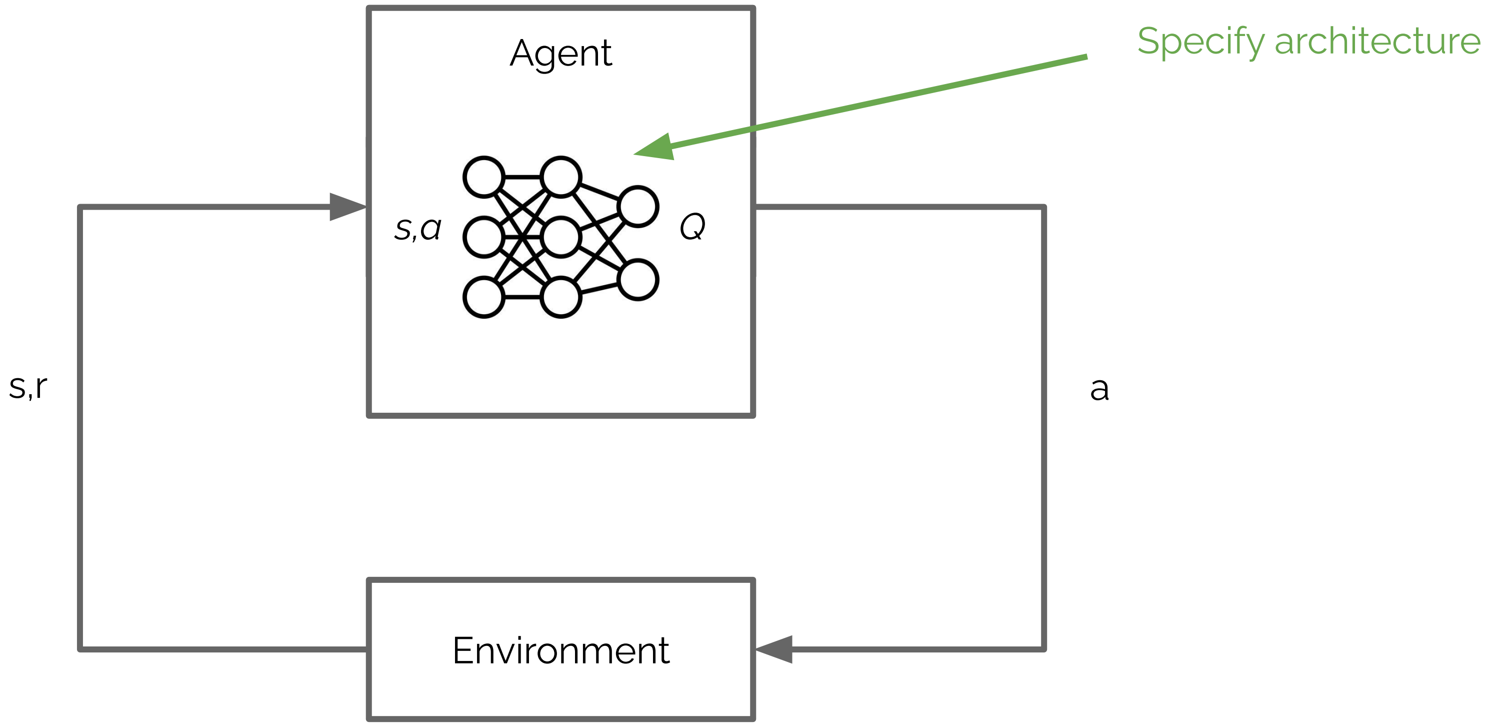
Deep Reinforcement Learning



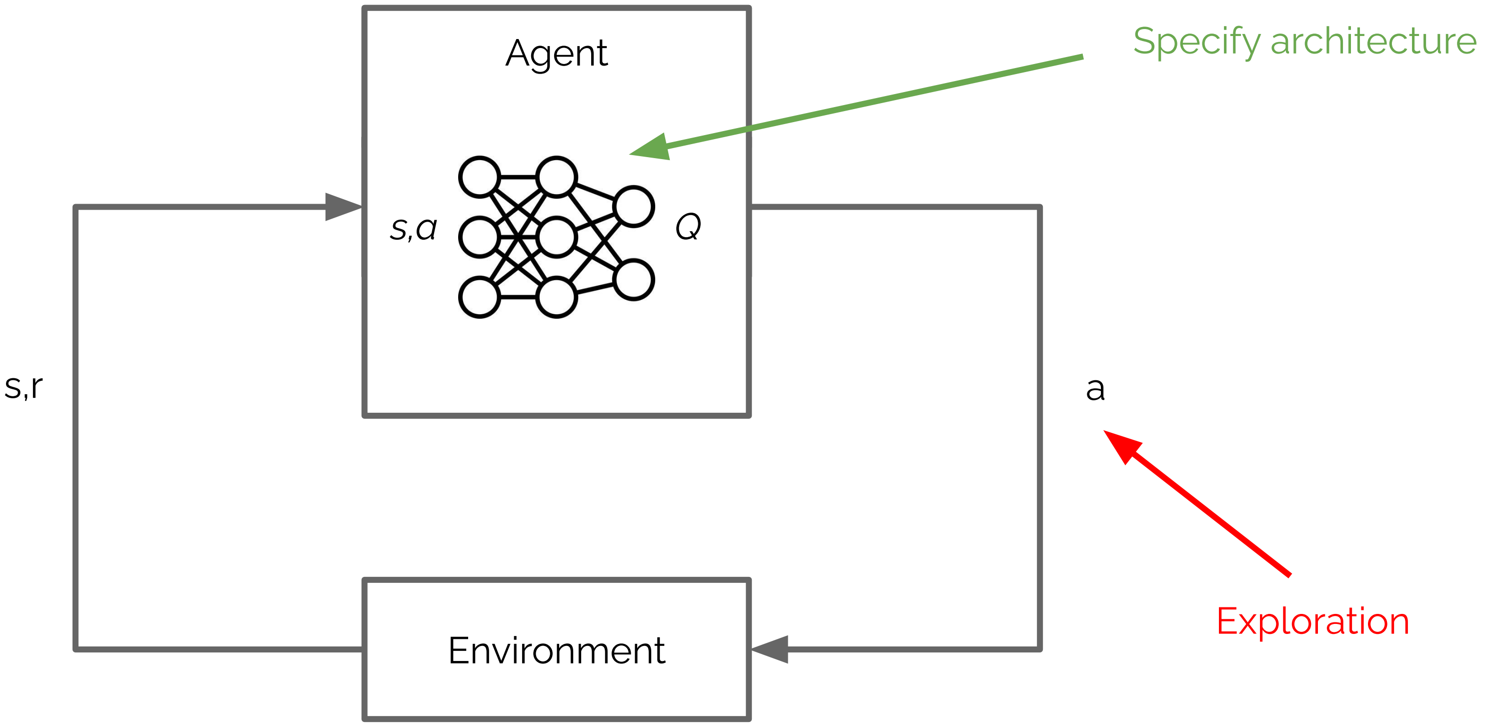
Deep Reinforcement Learning



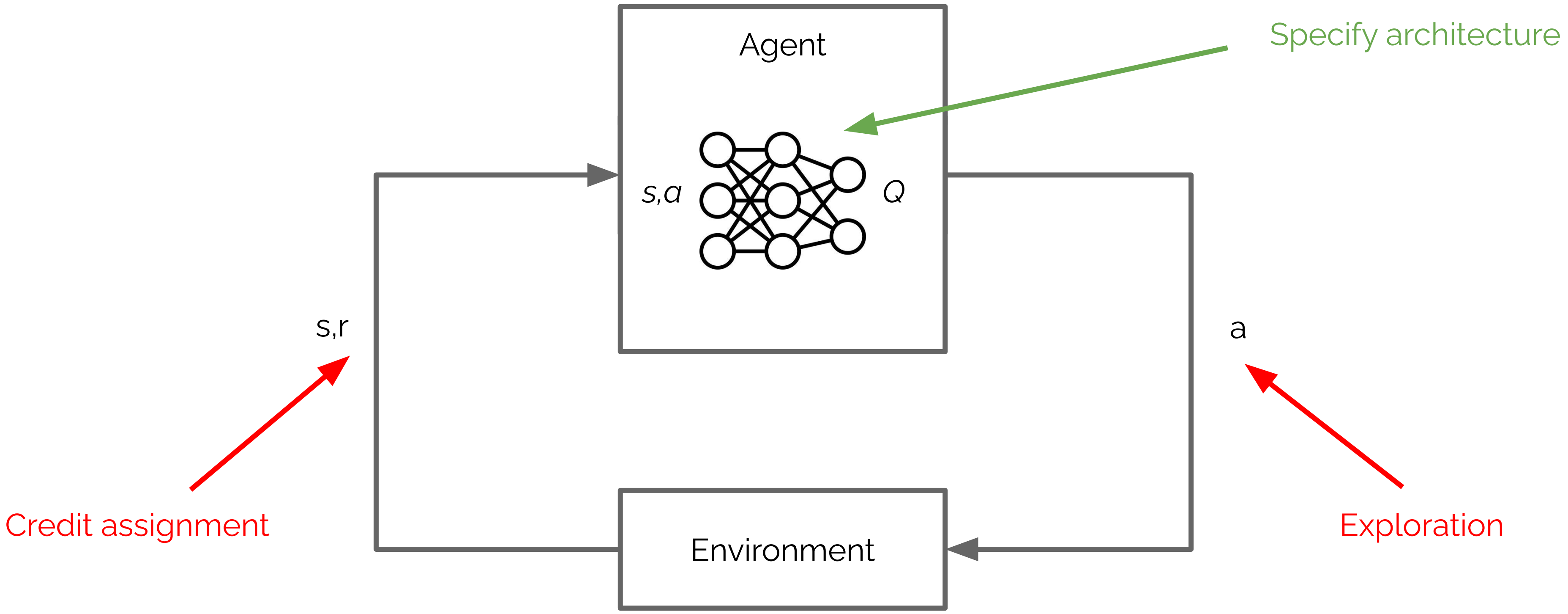
Deep Reinforcement Learning



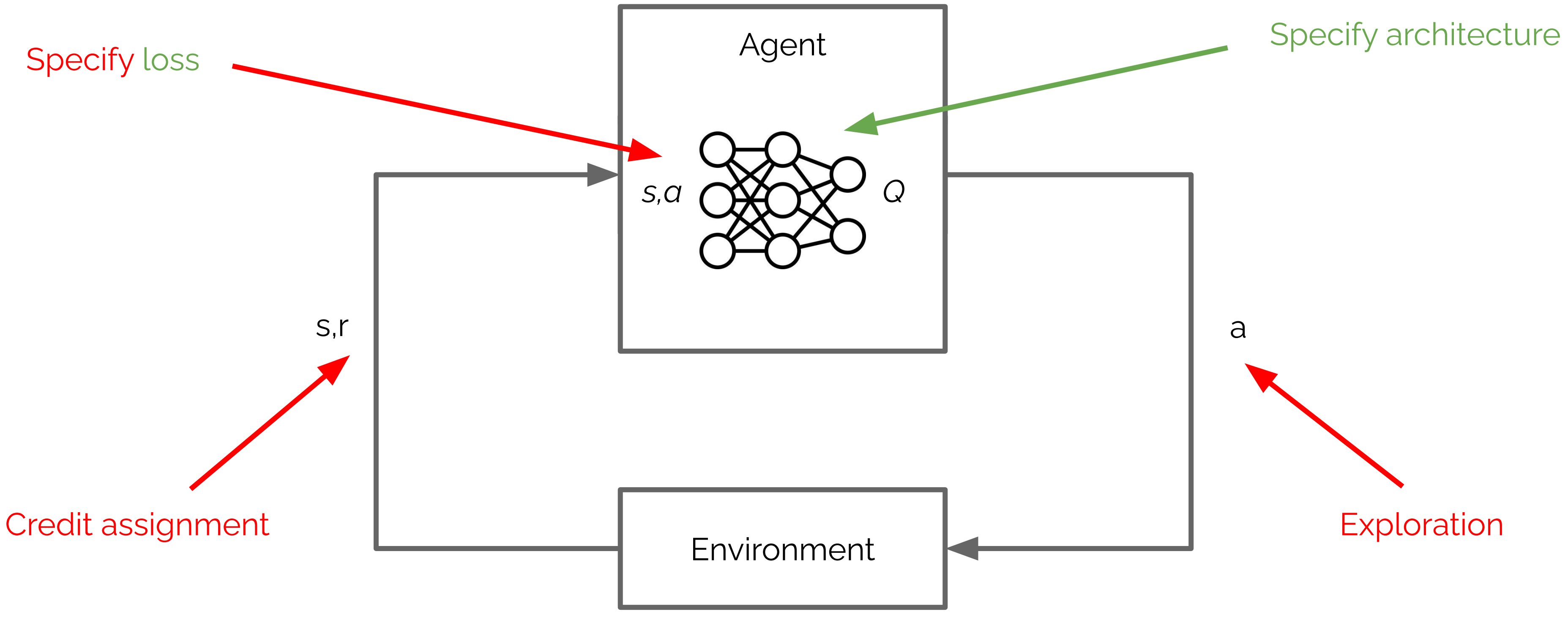
Deep Reinforcement Learning



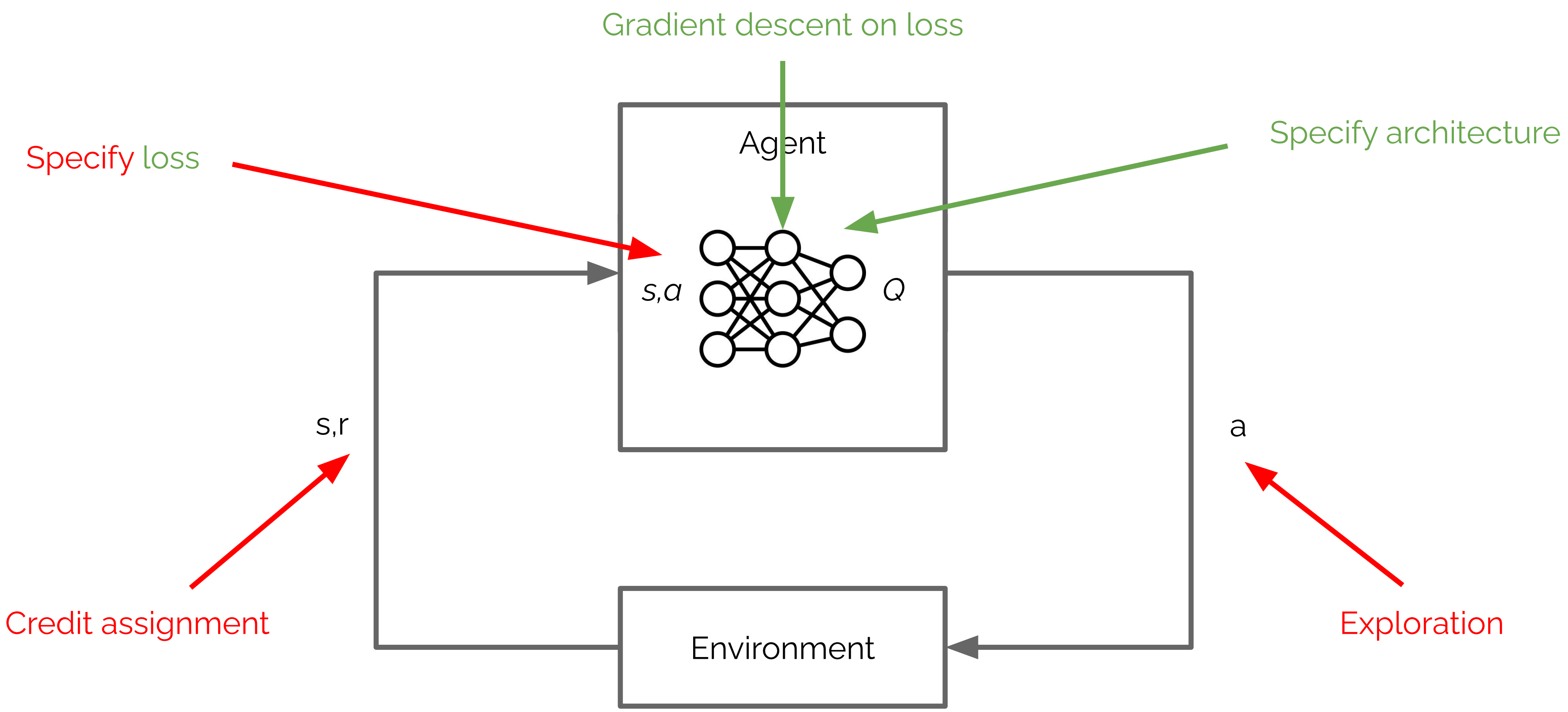
Deep Reinforcement Learning



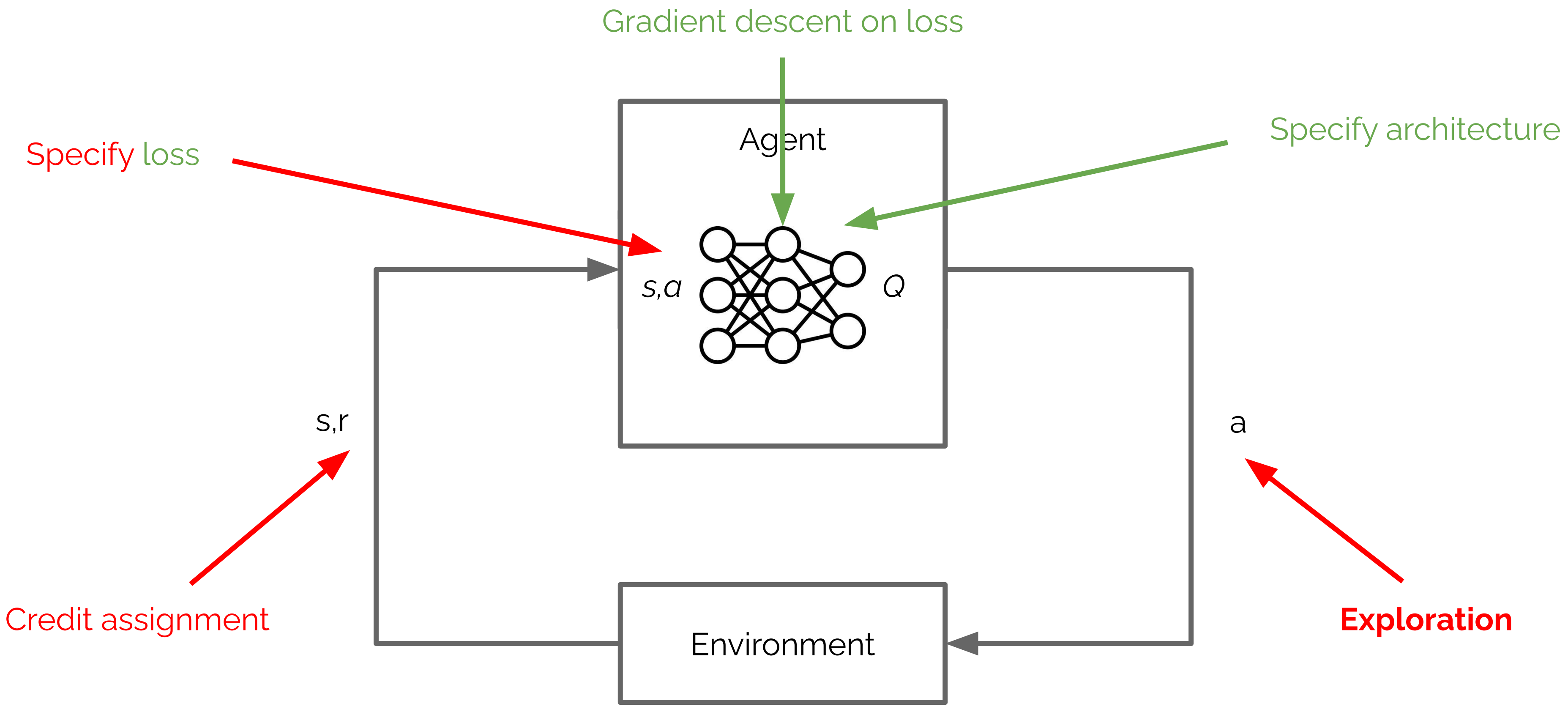
Deep Reinforcement Learning



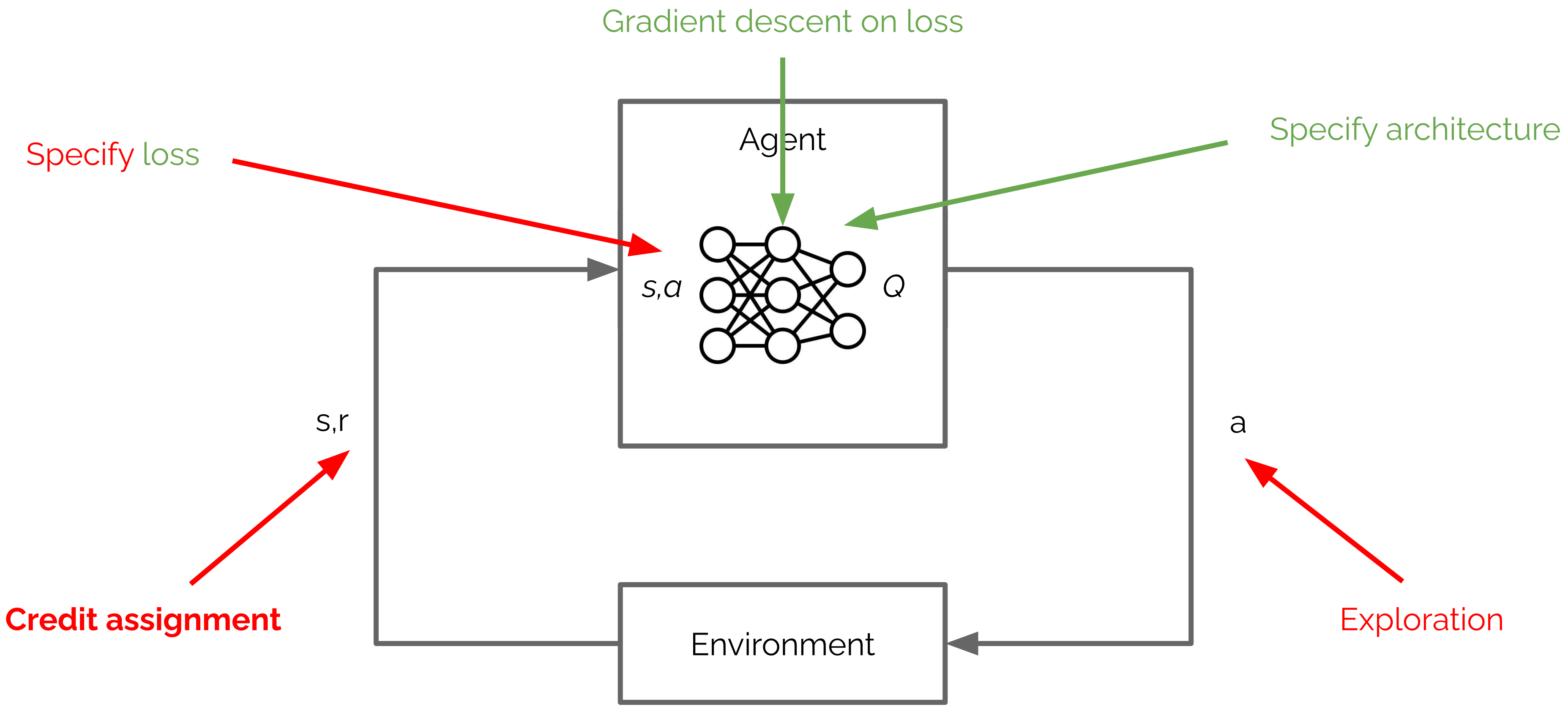
Deep Reinforcement Learning



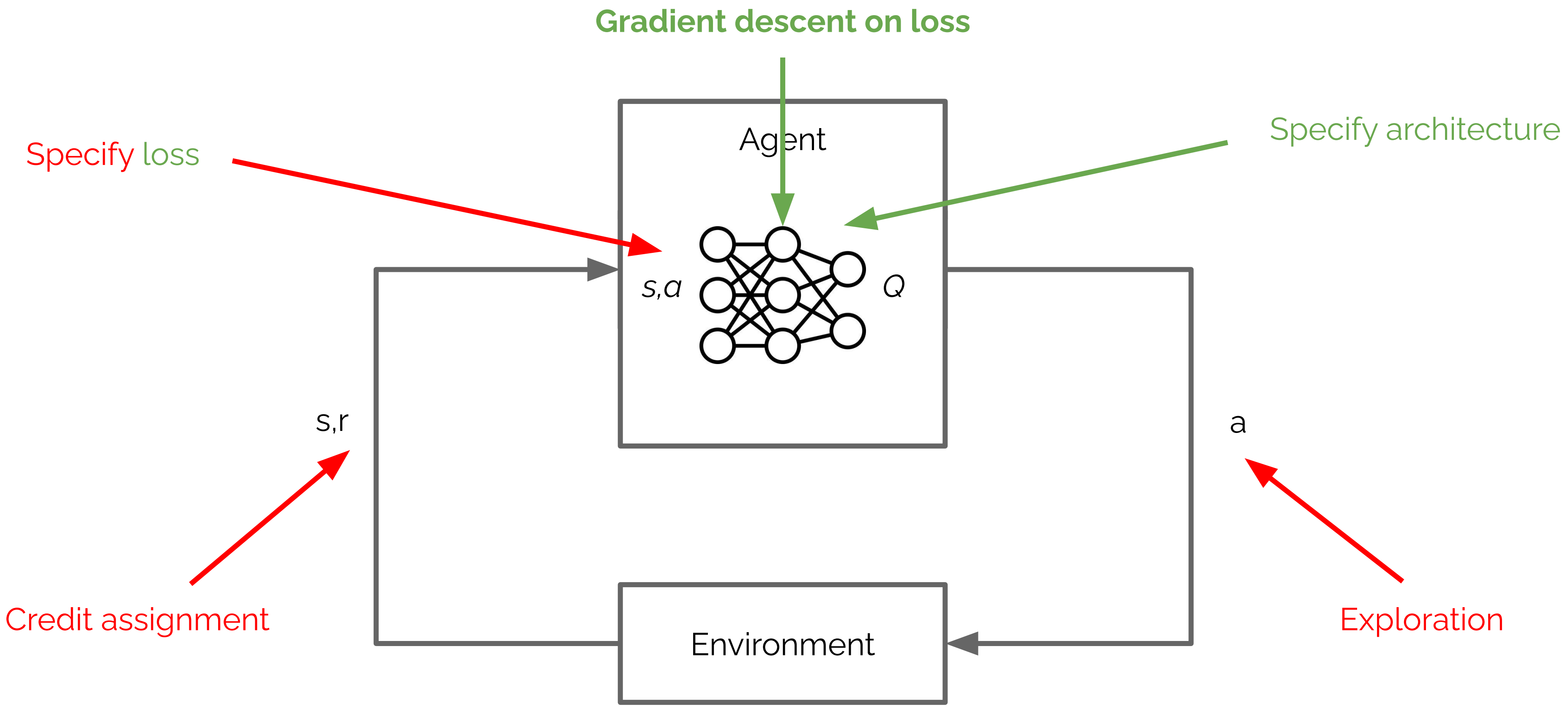
Deep Reinforcement Learning



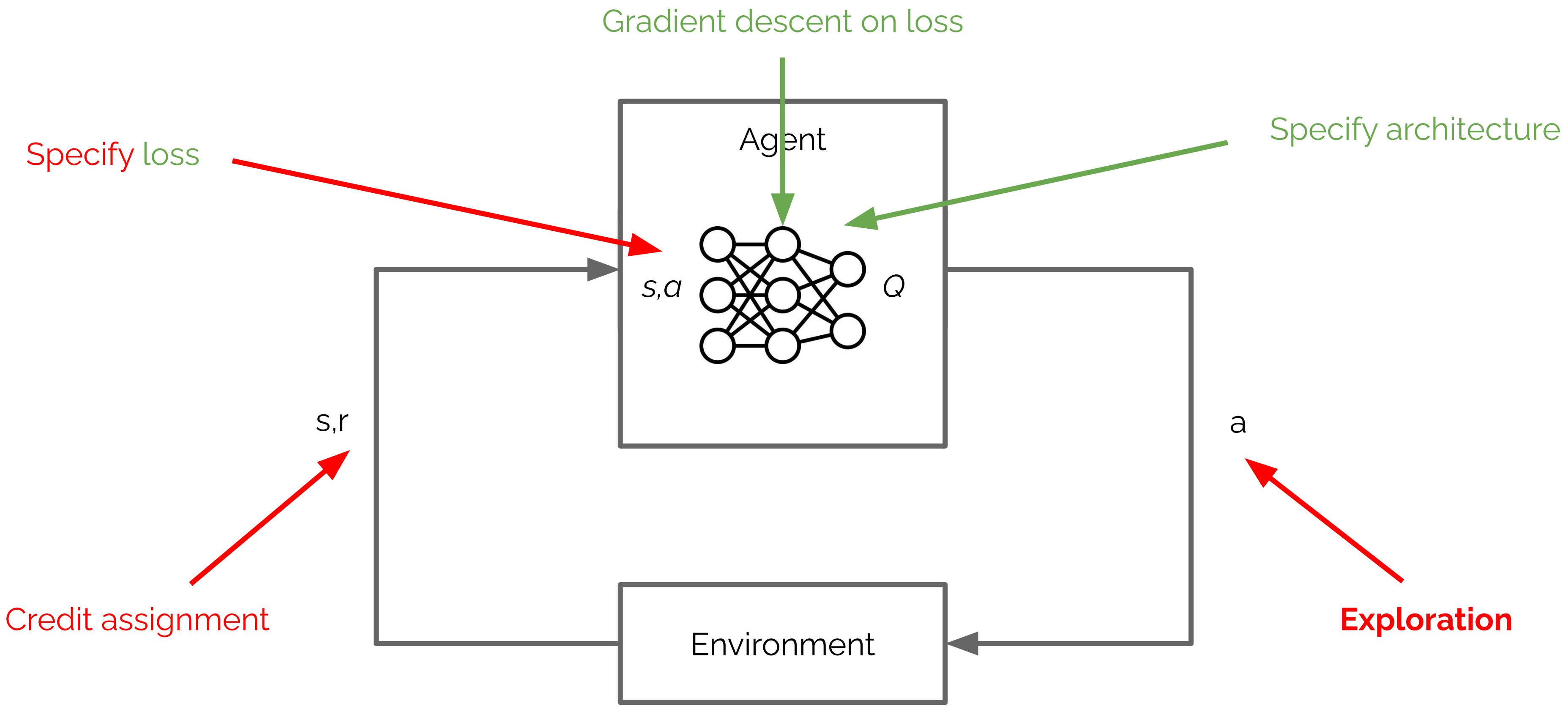
Deep Reinforcement Learning



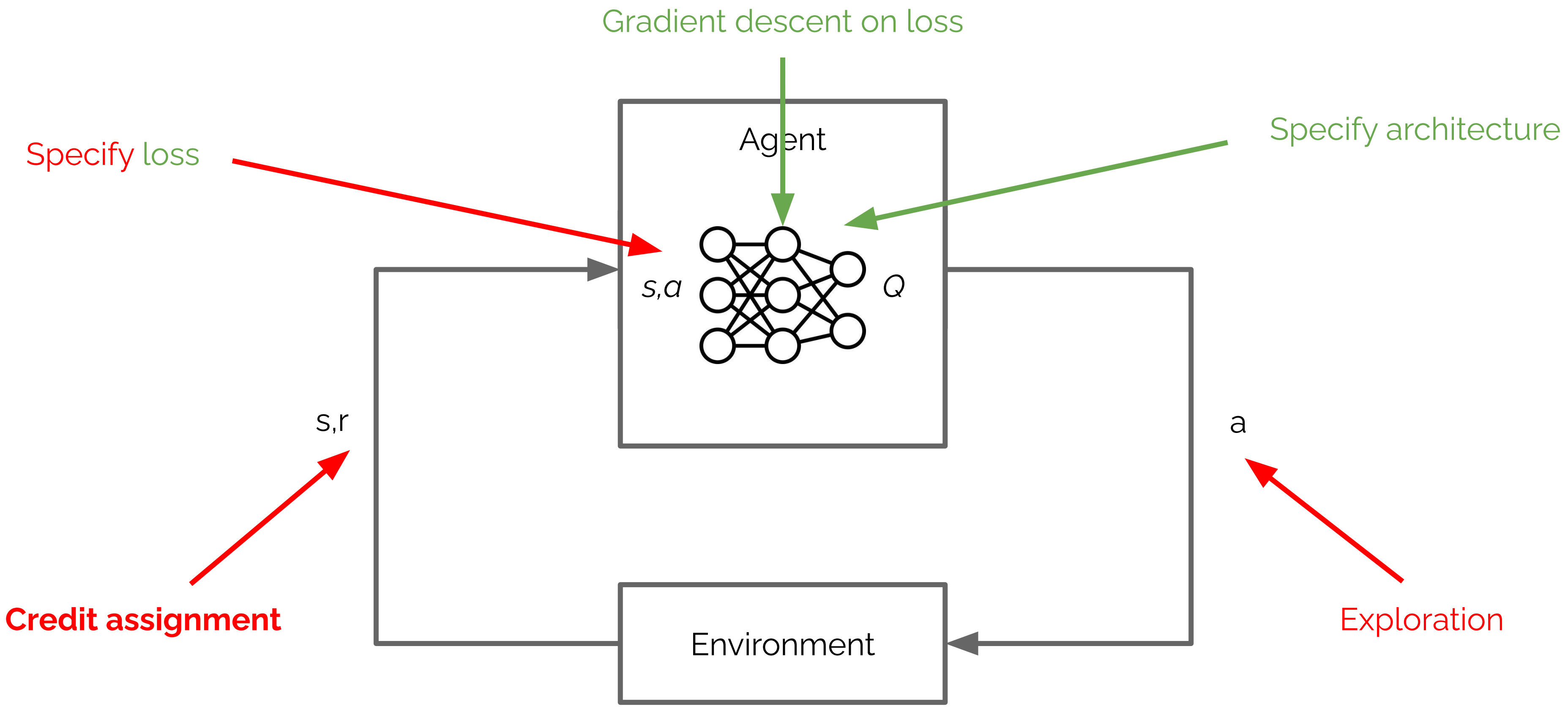
Deep Reinforcement Learning



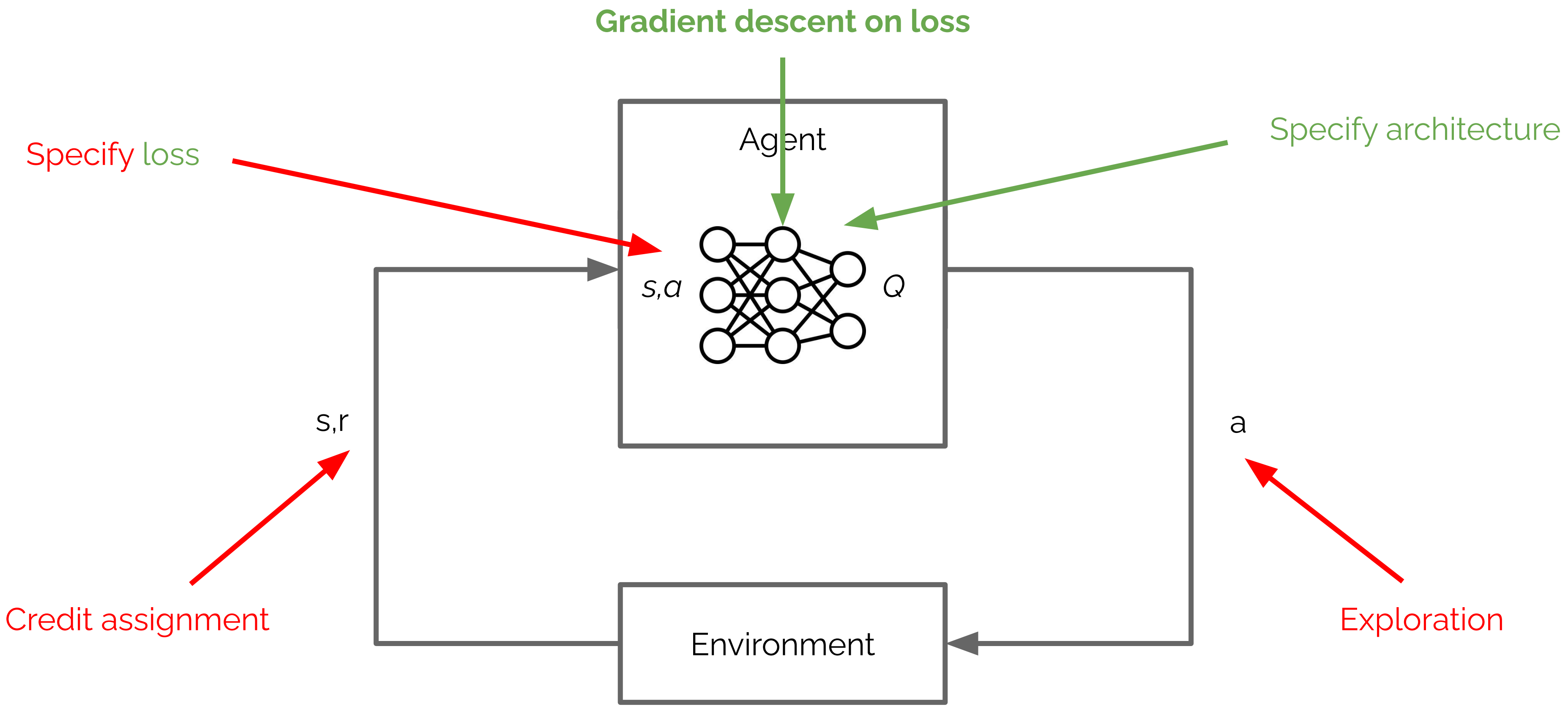
Deep Reinforcement Learning



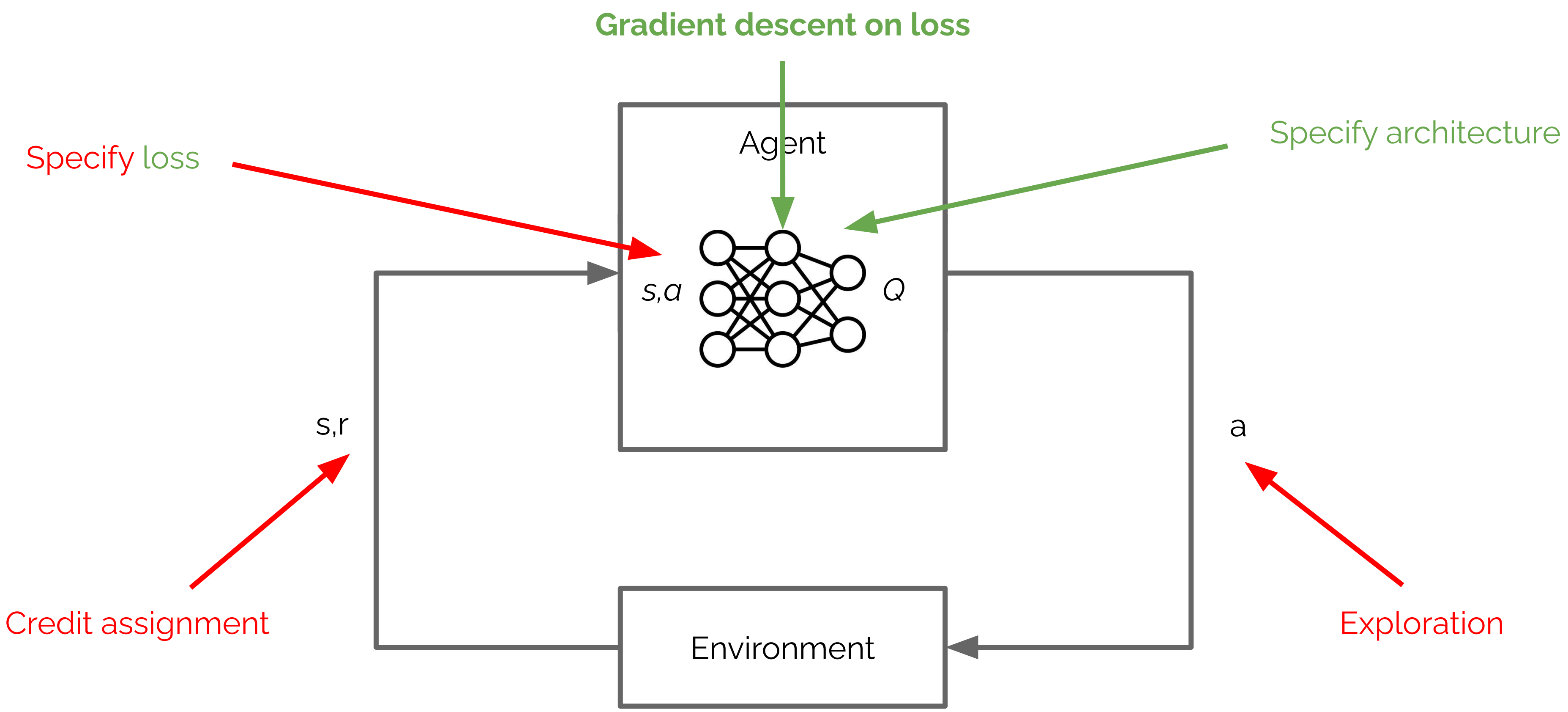
Deep Reinforcement Learning



Deep Reinforcement Learning

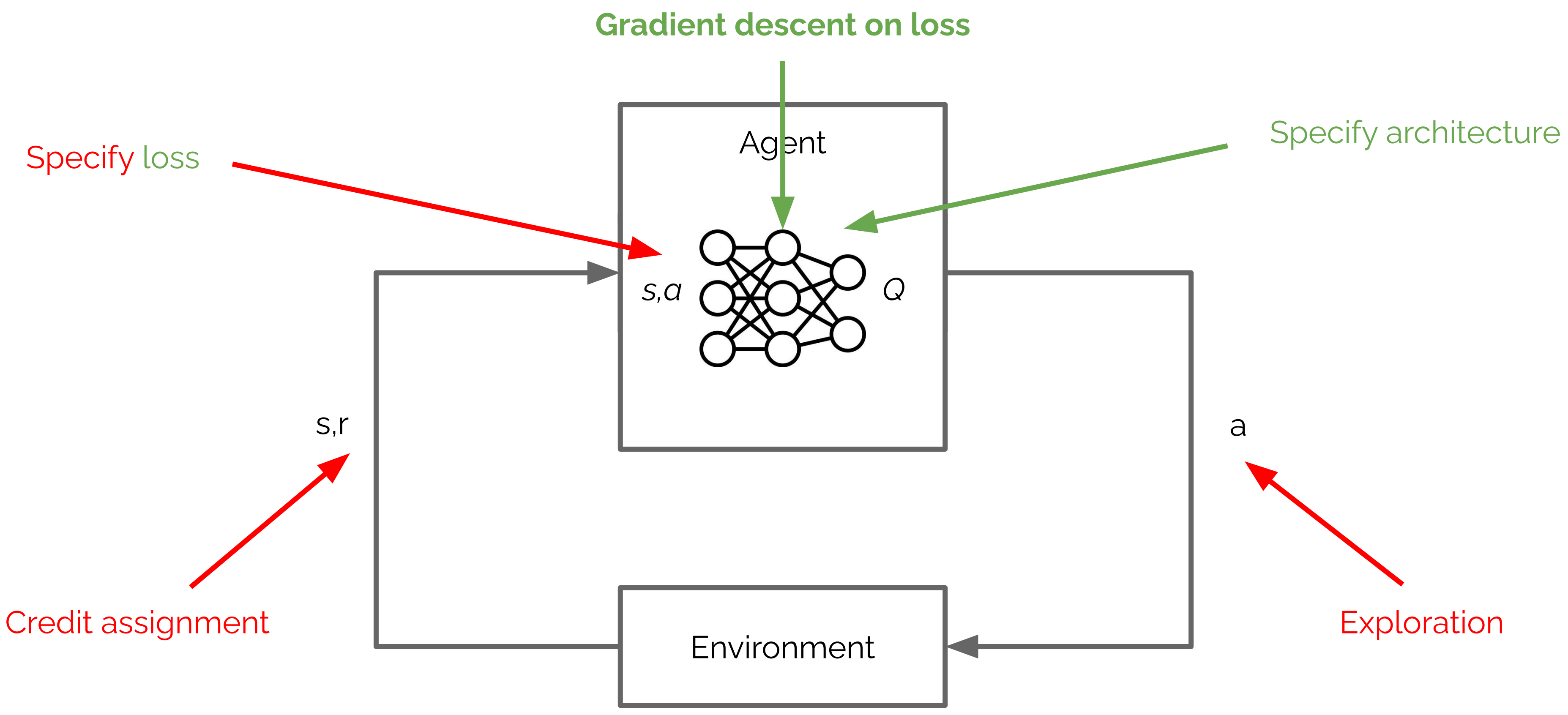


Deep Reinforcement Learning



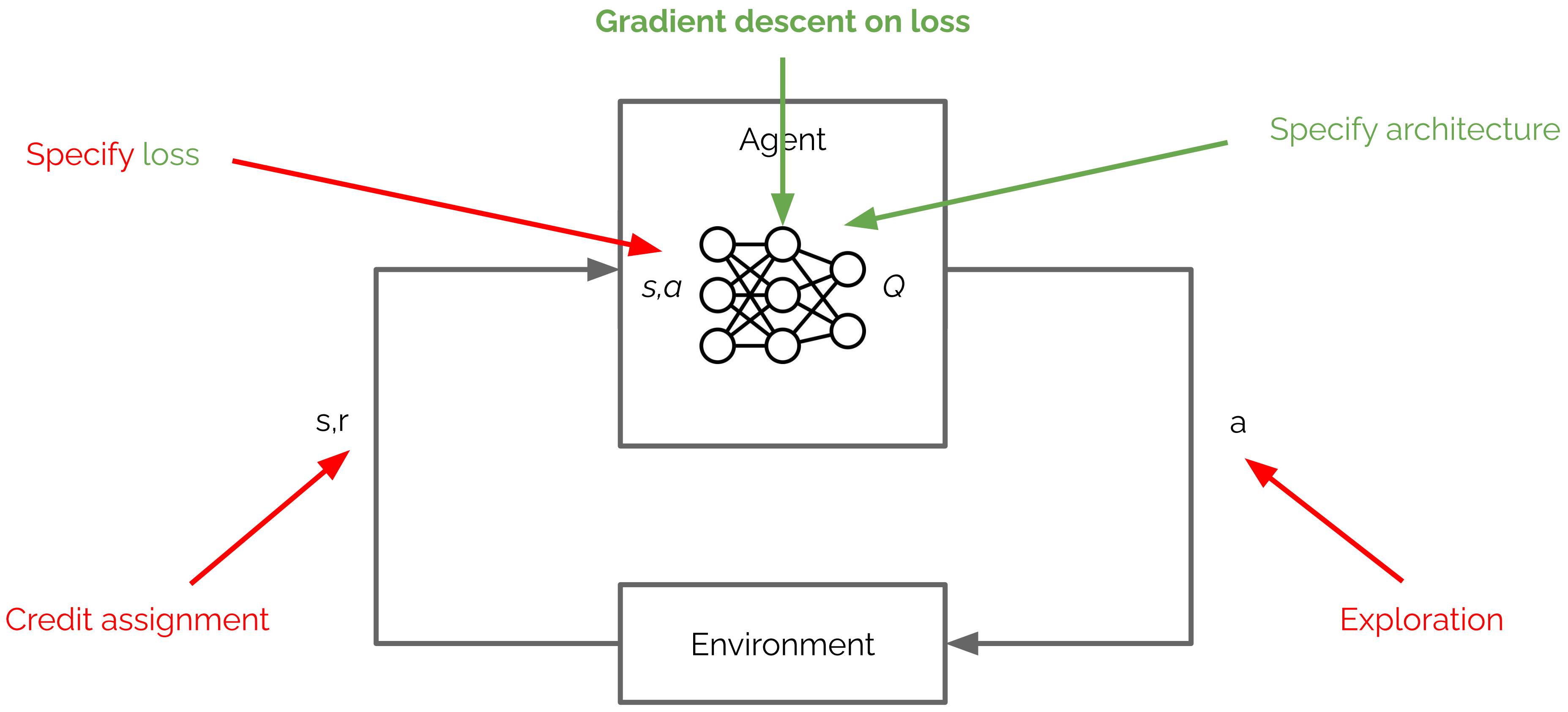
RL is supervised learning

Deep Reinforcement Learning



RL is supervised learning on a moving target function

Deep Reinforcement Learning



RL is supervised learning on a moving target function that influences which data you see.

Conclusion

Conclusion

Conclusion

- Many tasks can be formulated as a sequential decision making problem, for which you can use reinforcement learning (RL).

Conclusion

- Many tasks can be formulated as a sequential decision making problem, for which you can use reinforcement learning (RL).
- The main benefit of RL is that you can learn tasks you only label on outcomes, potentially outperforming human solutions.

Conclusion

- Many tasks can be formulated as a sequential decision making problem, for which you can use reinforcement learning (RL).

- The main benefit of RL is that you can learn tasks you only label on outcomes, potentially outperforming human solutions.

- Key topics in RL are exploration (which action should I try next) and credit assignment (how do I process the obtained reward information).

Conclusion

- Many tasks can be formulated as a sequential decision making problem, for which you can use reinforcement learning (RL).

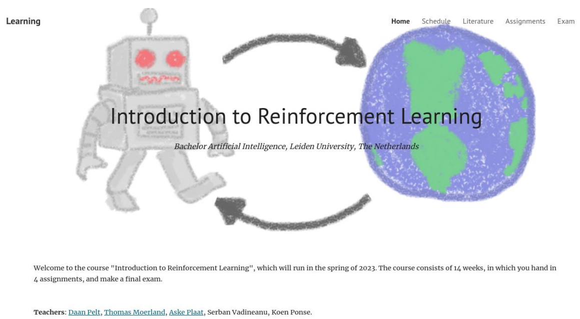
- The main benefit of RL is that you can learn tasks you only label on outcomes, potentially outperforming human solutions.

- Key topics in RL are exploration (which action should I try next) and credit assignment (how do I process the obtained reward information).

- Reinforcement learning is supervised learning on a moving target that influences which data you see.

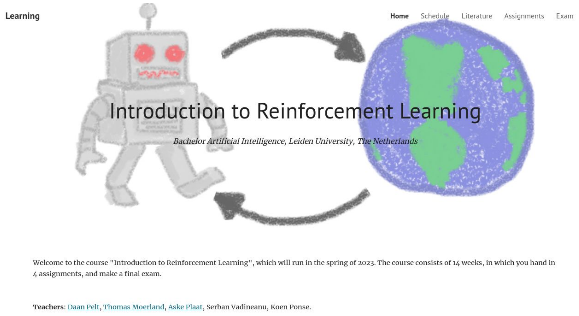
Courses

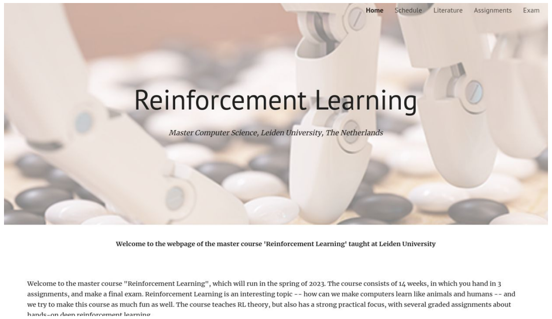
Courses



irl.liacs.nl

Courses

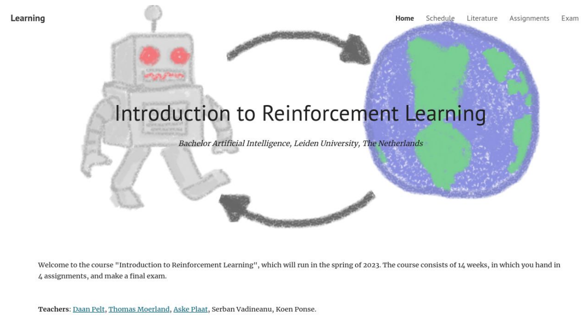


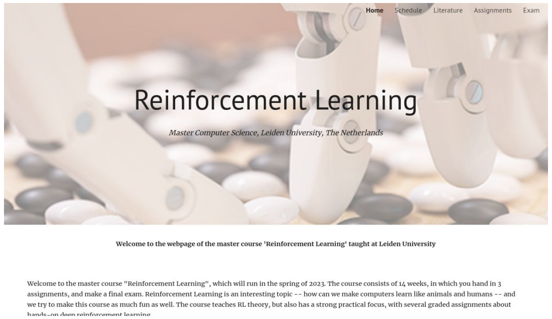


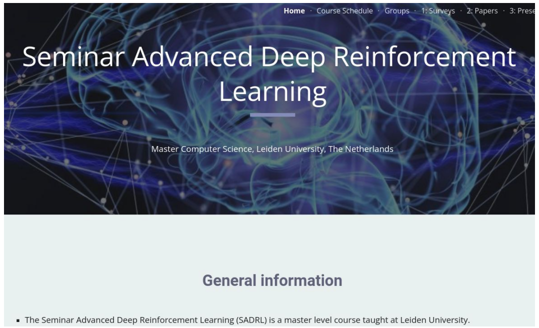
irl.liacs.nl

rl.liacs.nl

Courses

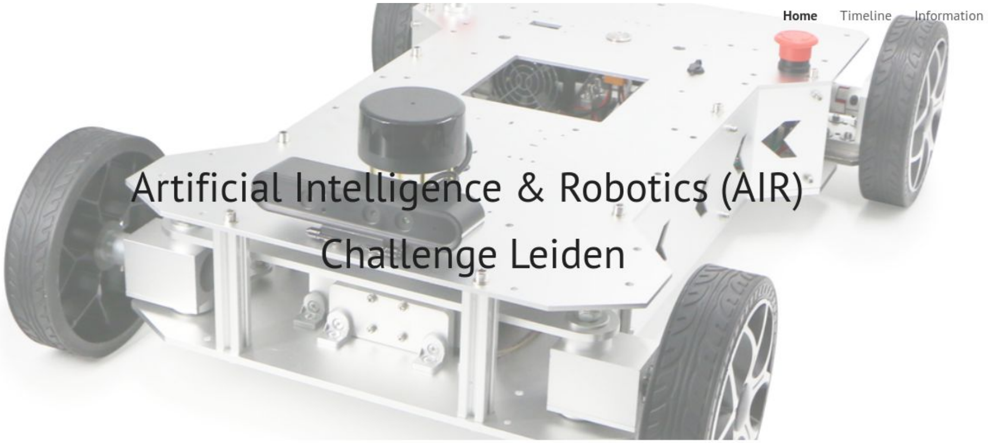






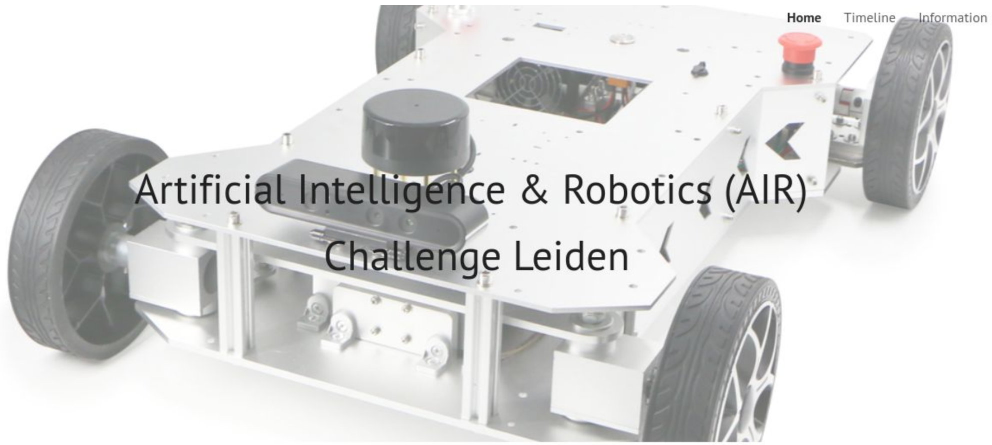
irl.liacs.nl rl.liacs.nl arl.liacs.nl

AI & Robotics challenge



The AI & Robotics Challenge is a yearly bachelor student competition that runs within the Leiden Institute of Advanced Computer Science (LIACS).

AI & Robotics challenge



The AI & Robotics Challenge is a yearly bachelor student competition that runs within the Leiden Institute of Advanced Computer Science (LIACS).

Extra-curricular course (2 ECTS)

Sign-up in September 2023

Questions?