Hi

school of ai Algiers

Deep Reinforcement Learning

Deep Q-Network

Azzeddine CHENINE
@azzeddineCH_

AI EVERYTIME EVERYWHERE

school of ai Algiers

OpenAI

15 October 2019

school of ai Algiers

Quick Reminder

Q-Learning

Quick Reminder

Learning by experience

Quick Reminder

Learning by experience

Episodic tasks and continuous task

Quick Reminder

Learning by experience

Episodic tasks and continuous task

An action has a reward and results a new state

Quick Reminder

The value based methods try to reach an optimal policy by using a value function

Quick Reminder

The value based methods try to reach an optimal policy by using a value function

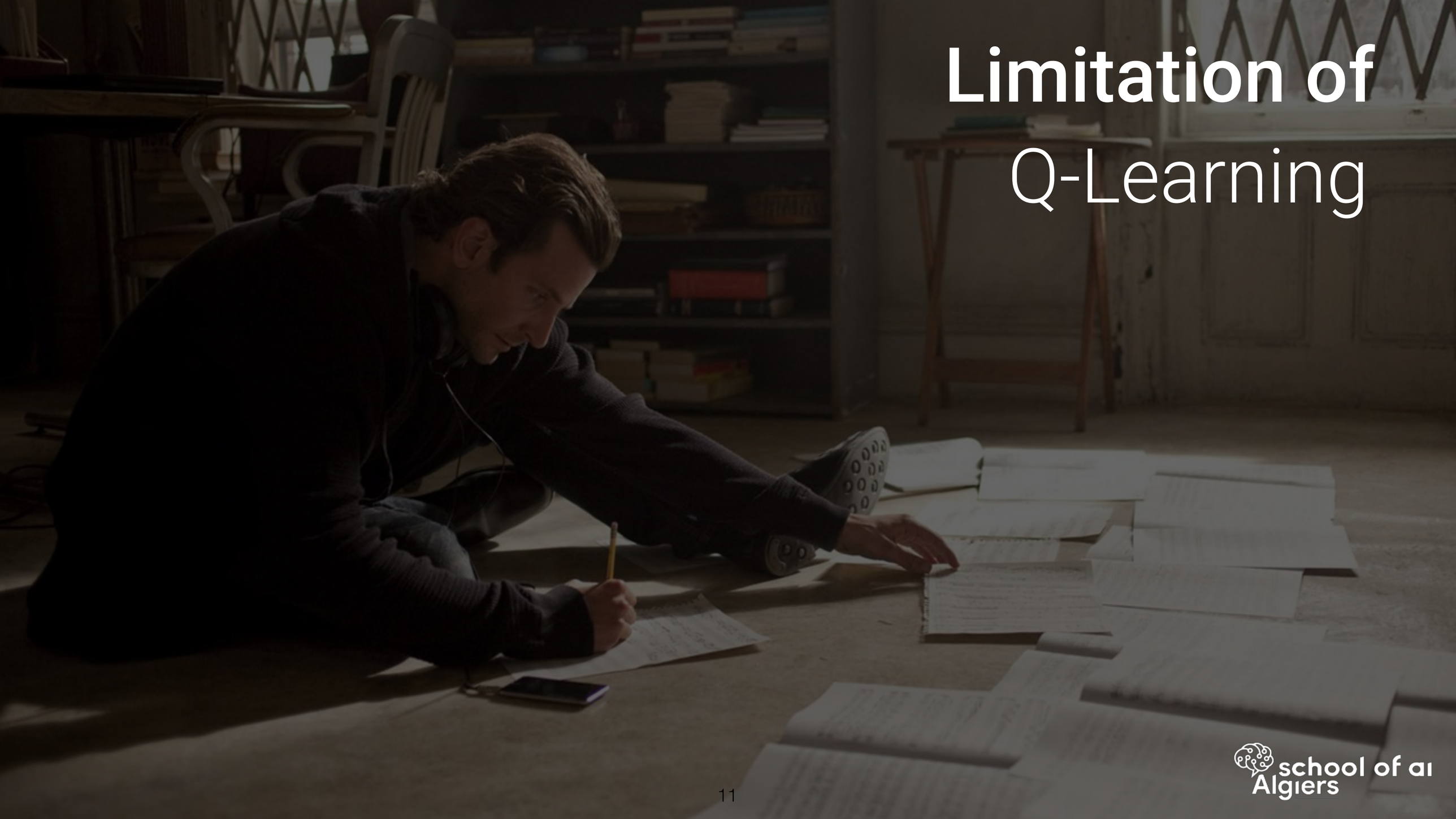
The Q function of an optima policy maximise the expected return G

Quick Reminder

The value based methods try to reach an optimal policy by using a value function

The Q function of an optima policy maximise the expected return G

Epsilon greedy policies solve the exploration and exploitation problem



Limitation of Q-Learning

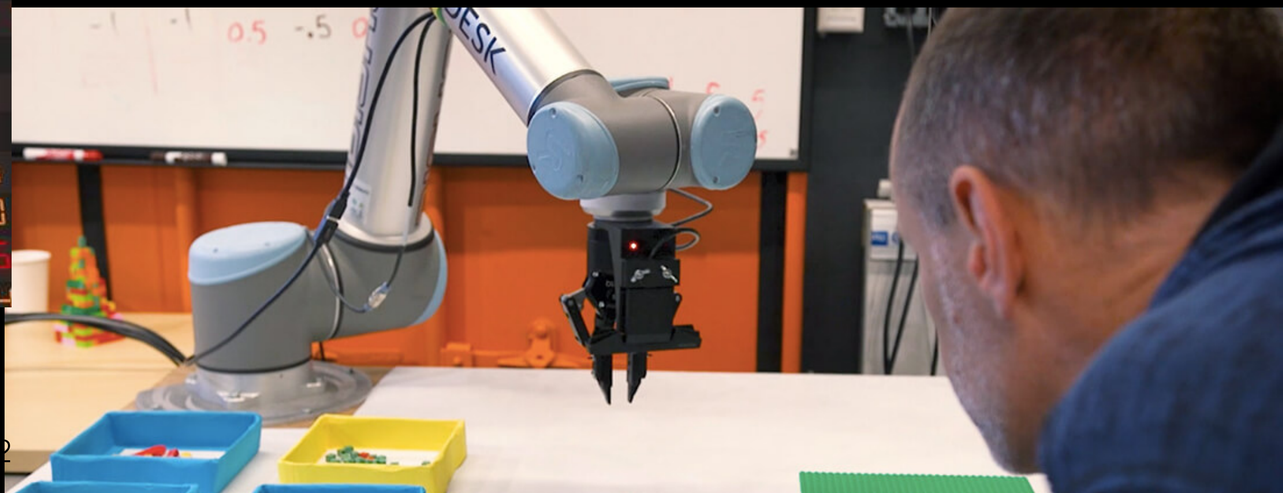
Limitation

Non practical with environments with large or infinite state or action space

Games

Stock market

Robots



Modern problems
require
Modern solutions

Deep Q-Networks

Deepminds

- 2015 -

No more given definitions ….

No more given definitions ….

figuring it out by ourselves

Neural networks can be trained to fit on a linear or a non linear function.

1 - How to replace the Q-table ?

1 - How to replace the Q-table ?

Neural networks can be trained to fit on a linear or a non linear function.

Replace the Q-value function with a deep neural network

1 - How to replace the Q-table ?

Neural networks can be trained to fit on a linear or a non linear function.

Replace the Q-value function with a deep neural network

The outputs layer hold the Q-value of each possible action

$\Longrightarrow$ **make the actions space discrete**

1 - How to replace the Q-table ?

Neural networks can be trained to fit on a linear or a non linear function.

Replace the Q-value function with a deep neural network

The outputs layer hold the Q-value of each possible action

$\Longrightarrow$ **make the actions space discrete**

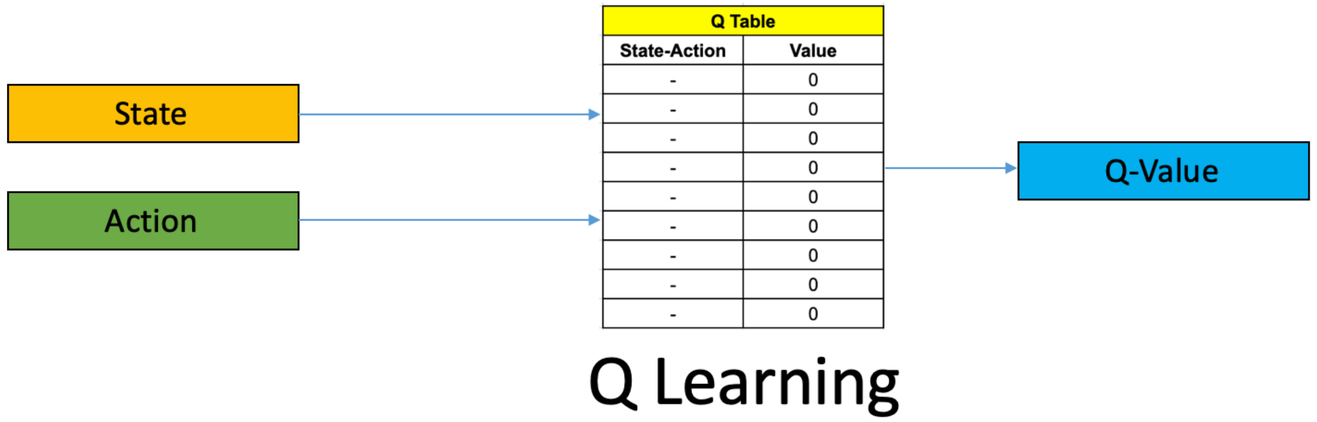
The input layer takes the desired representation of the state

$\Longrightarrow$ **a pipeline of convolutions**

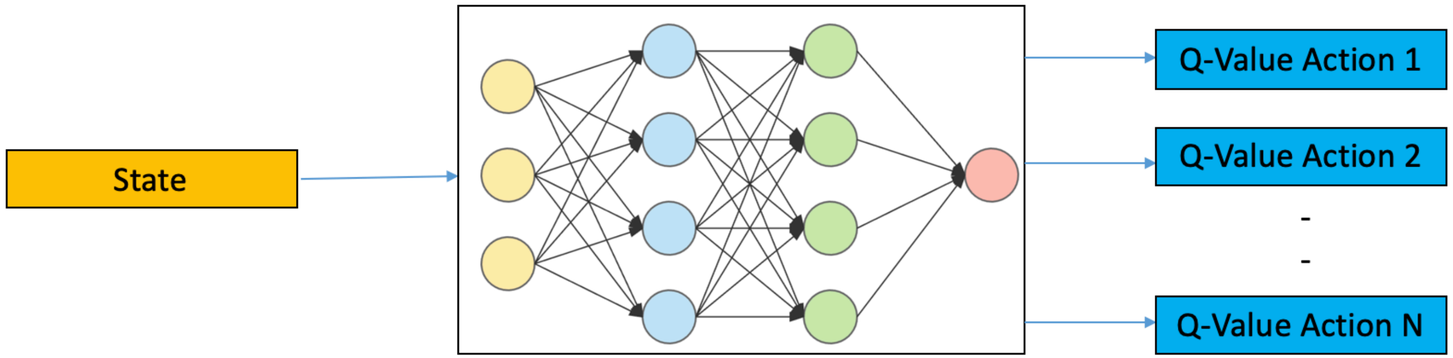
$\Longrightarrow$ **a Dense layer**

$\Longrightarrow$ **a sequence of LSTM/RNN cells**

1 - How to replace the Q-table ✅



Q Learning



Deep Q Learning

How did we used update the Q-table

The agent took an action (a) in a state (S)

1 - How to replace the Q-table ✅

2 - How to optimize the model

How did we used update the Q-table

The agent took an action (a) in a state (S)

The agent moved to the next state (S’) and received a reward (r)

1 - How to replace the Q-table ✅

2 - How to optimize the model

How did we used update the Q-table

The agent took an action (a) in a state (S)

The agent moved to the next state (S') and received a reward (r)

the discount rate δ reflects how much we care about the futur reward

$\delta \approx 1 \Longrightarrow$ we care too much about the future reward

$\delta \approx 0 \Longrightarrow$ we care too much about the immediate reward

1 - How to replace the Q-table ✅

2 - How to optimize the model

How did we used update the Q-table

$$Q'(S, a) \Longleftarrow Q(S, a) + \alpha\,[\,r(S, a, S') + \delta \max Q(S', a') - Q(S, a)\,]$$

A good **Q** value for an action (a) given a state (S) is a good approximation to the expected discounted cumulative reward.

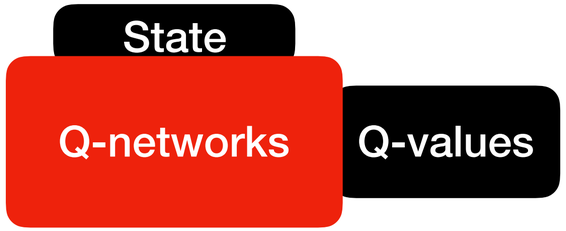
$$\Longrightarrow Q(S, a) \approx r(S, a, S') + \delta \max Q(S', a')$$

1 - How to replace the Q-table ✅

2 - How to optimize the model

How are we going to update the Q-networks weights

$$\Longrightarrow Q(S, a) \approx r(S, a, S') + \delta \max Q(S', a')$$

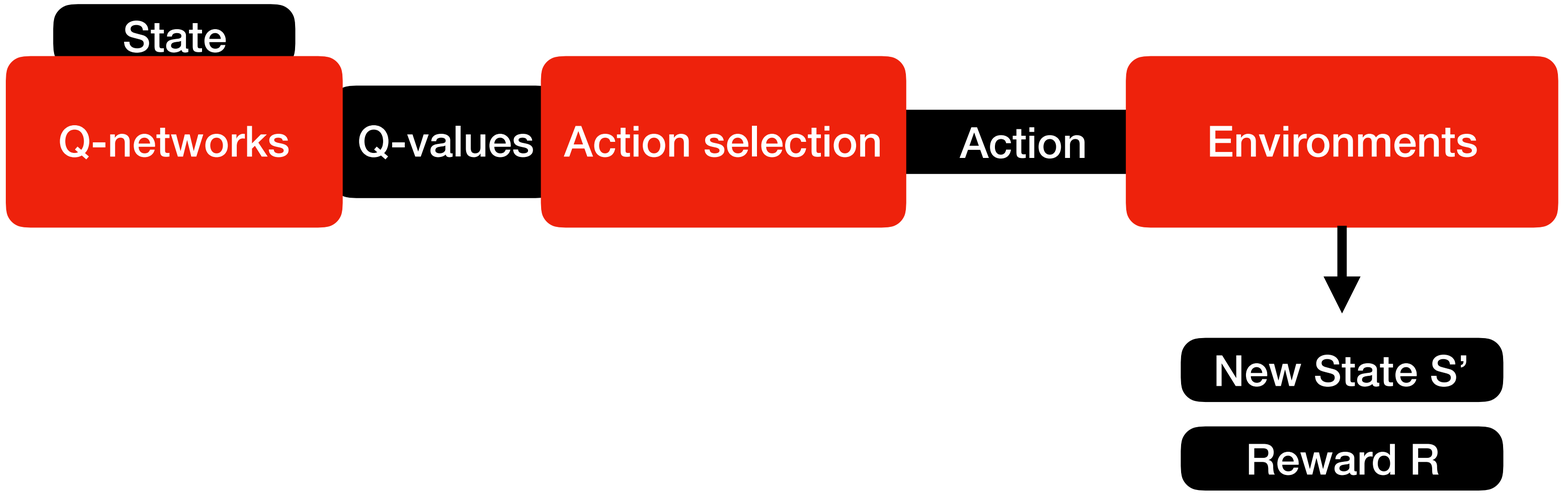


1 - How to replace the Q-table ✅

2 - How to optimize the model

How are we going to update the Q-networks weights

$$\Longrightarrow Q(S, a) \approx r(S, a, S') + \delta \max Q(S', a')$$

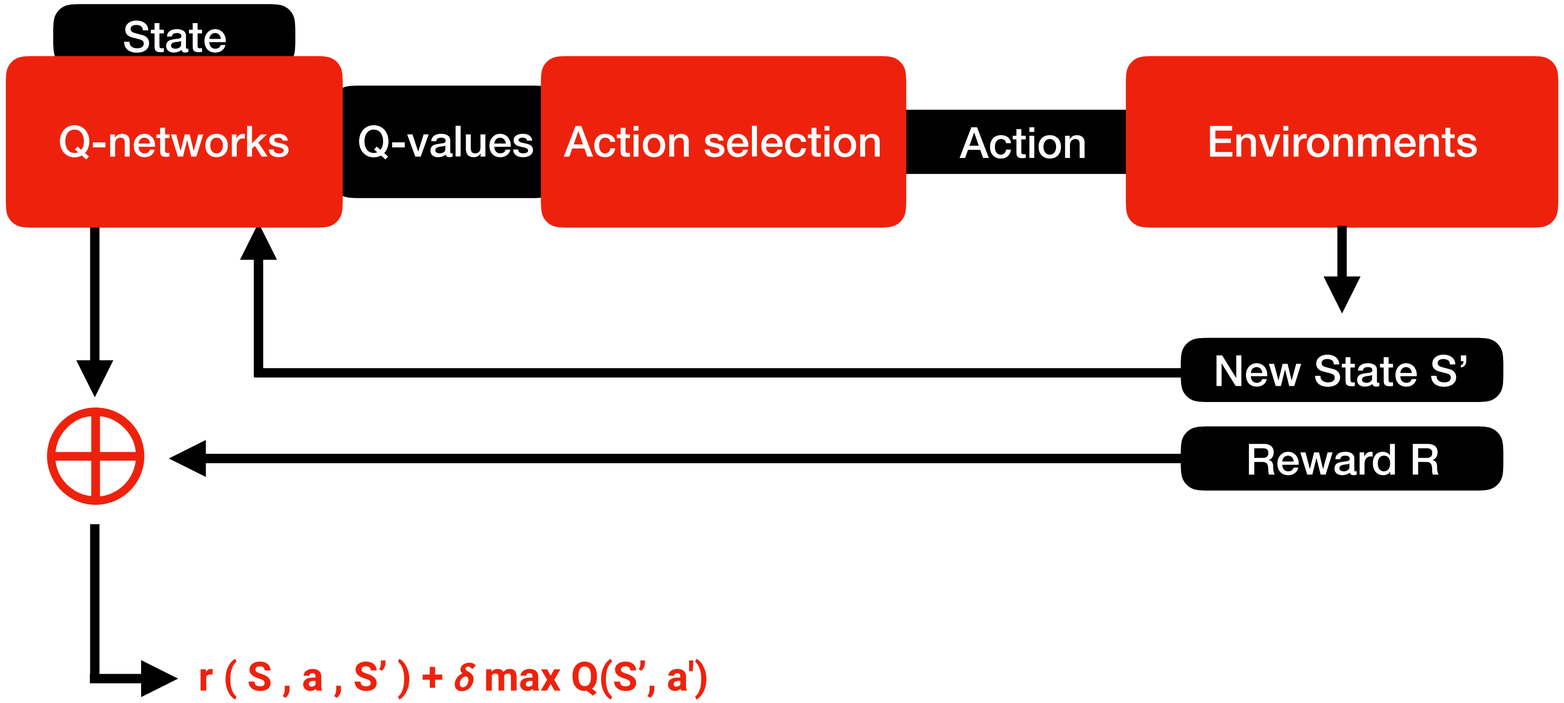


1 - How to replace the Q-table ✅

2 - How to optimize the model

How are we going to update the Q-networks weights

$$\Longrightarrow Q(S, a) \approx r(S, a, S') + \delta \max Q(S', a')$$

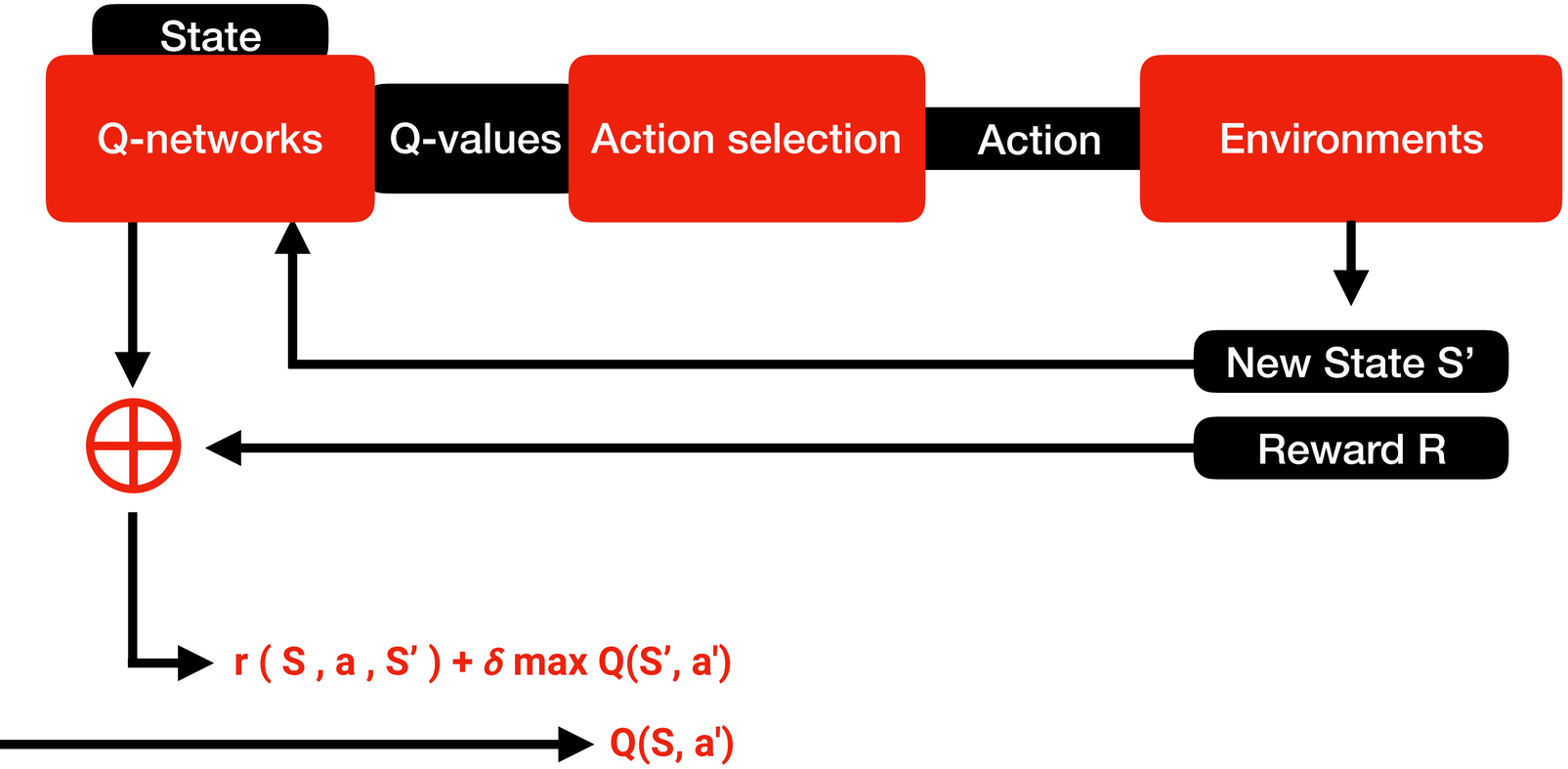


1 - How to replace the Q-table ✅

2 - How to optimize the model

How are we going to update the Q-networks weights

$$\Longrightarrow Q(S, a) \approx r(S, a, S') + \delta \max Q(S', a')$$

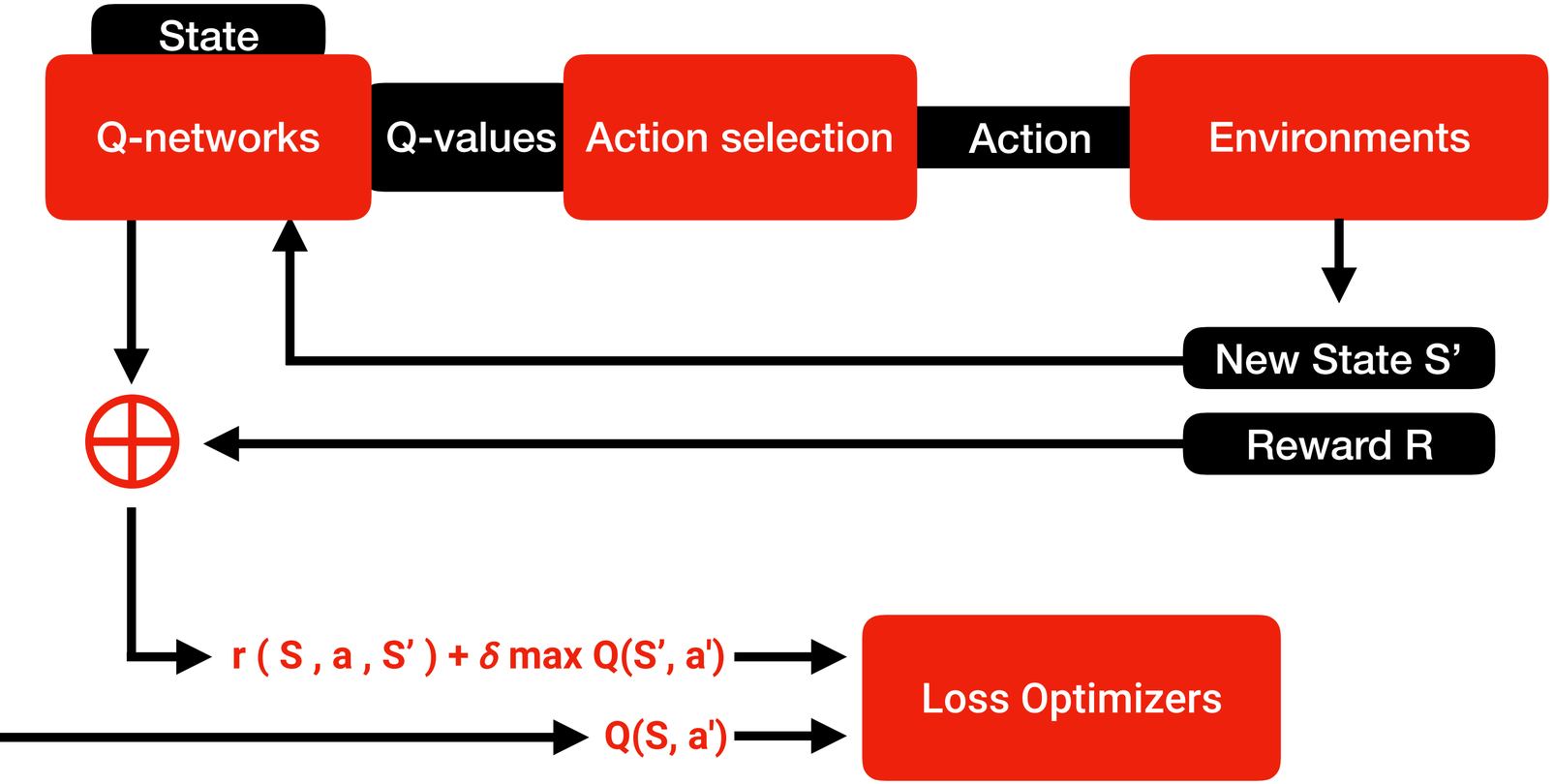


1 - How to replace the Q-table ✅

2 - How to optimize the model

How are we going to update the Q-networks weights

$$\Longrightarrow Q(S, a) \approx r(S, a, S') + \delta \max Q(S', a')$$

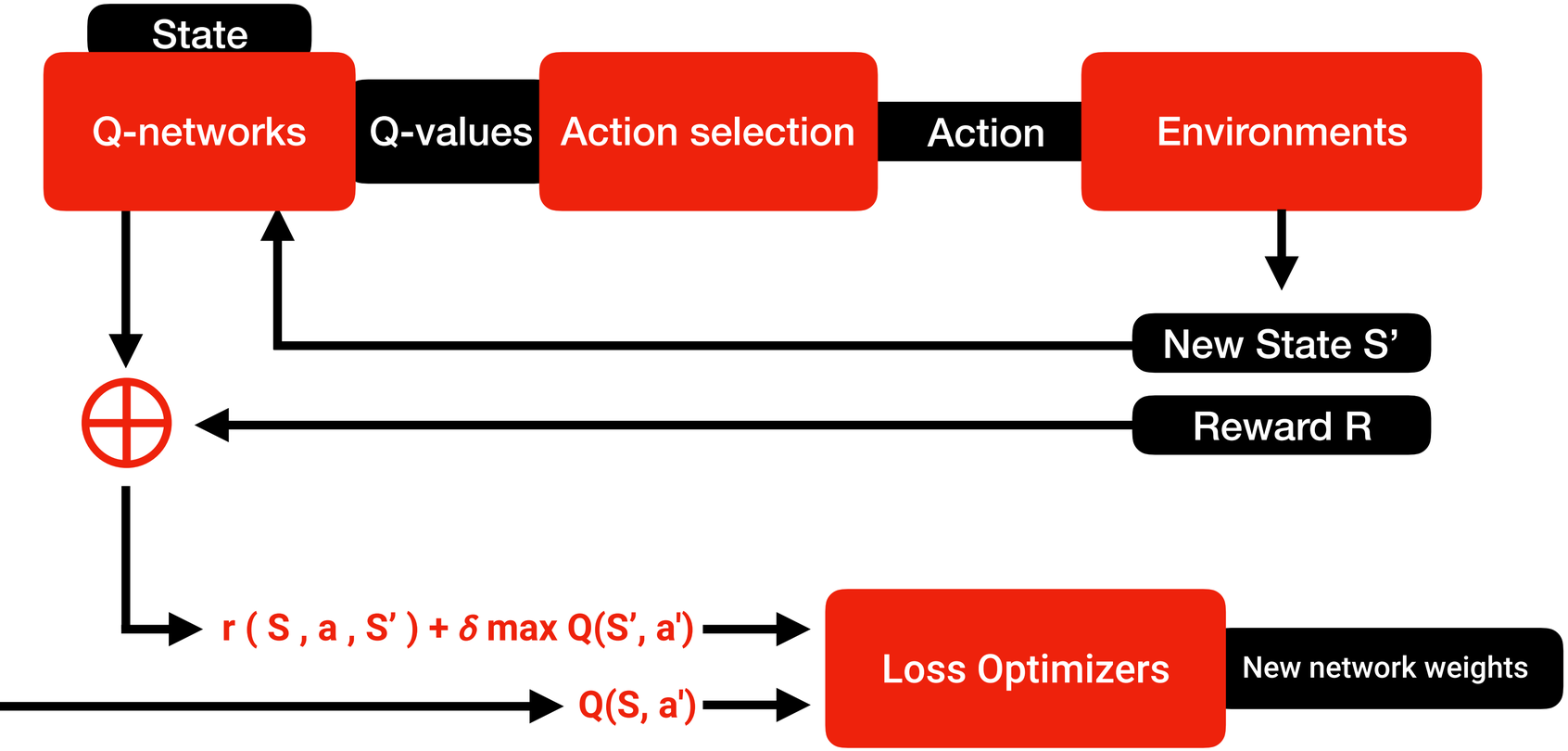


1 - How to replace the Q-table ✅

2 - How to optimize the model

How are we going to update the Q-networks weights

$$\Longrightarrow Q(S, a) \approx r(S, a, S') + \delta \max Q(S', a')$$

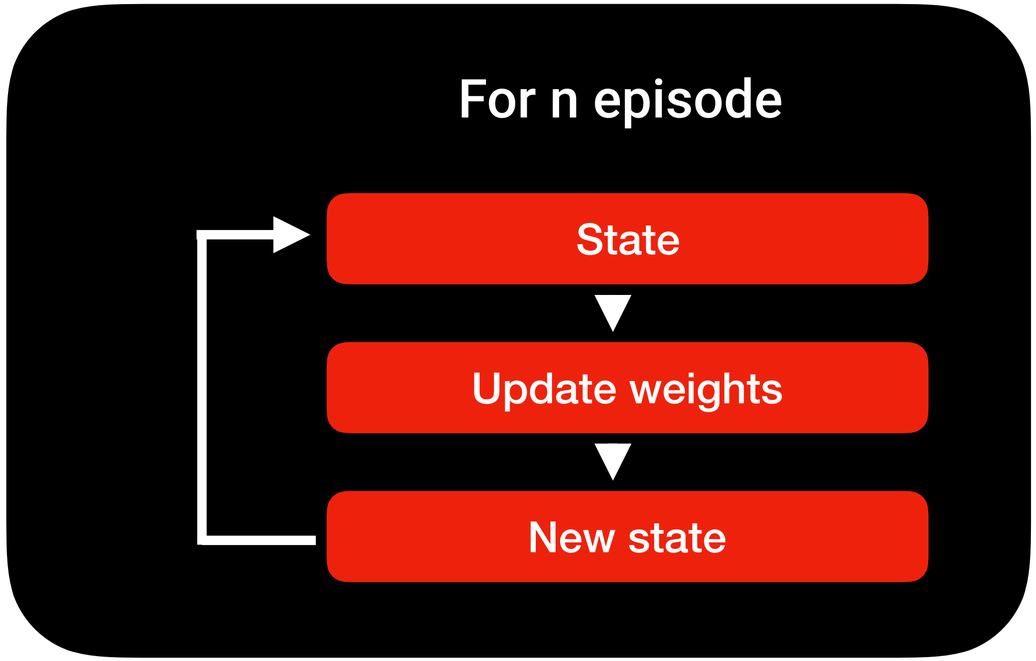


1 - How to replace the Q-table ✅

2 - How to optimize the model

How are we going to update the Q-networks weights

$$\Longrightarrow Q(S, a) \approx r(S, a, S') + \delta \max Q(S', a')$$



1 - How to replace the Q-table ✅

2 - How to optimize the model

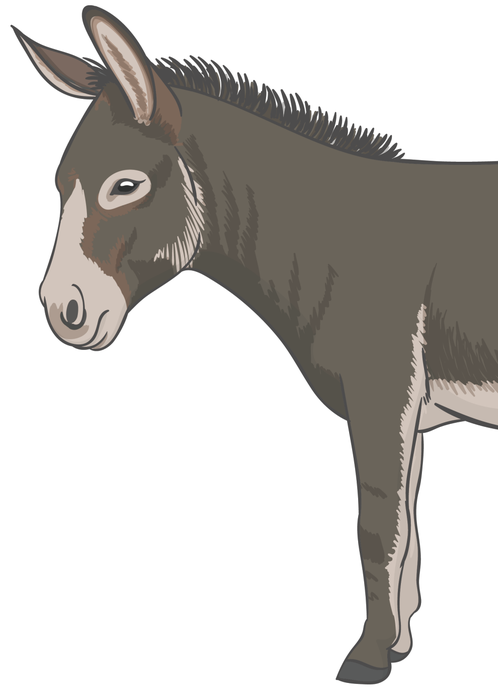
We are chasing a moving target

$r(S, a, S') + \delta \max Q(S', a')$

$Q(S, a)$

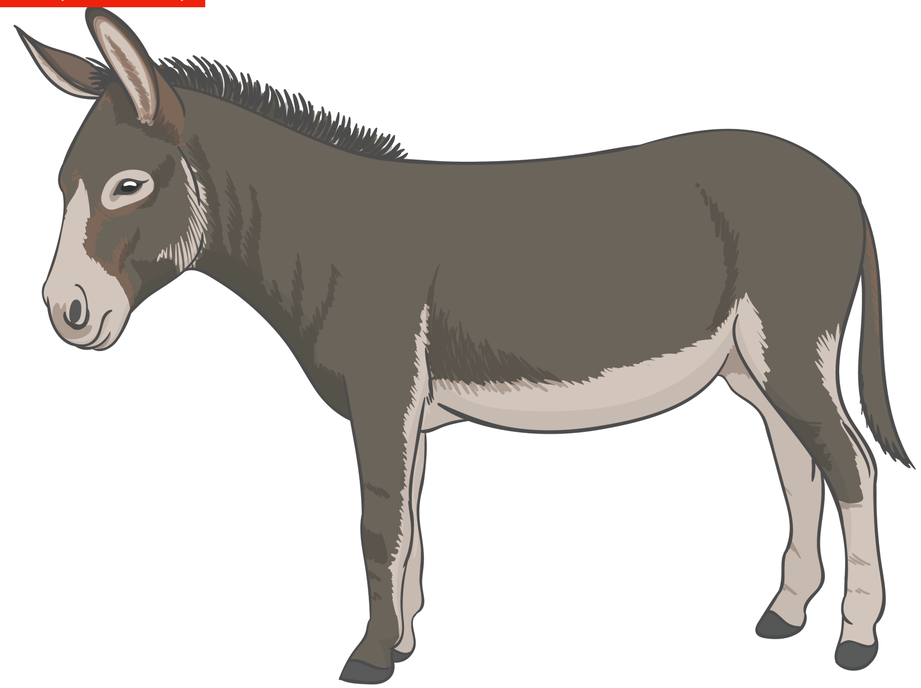
$$r(S, a, S') + \delta \max Q(S', a')$$

$$Q(S, a)$$

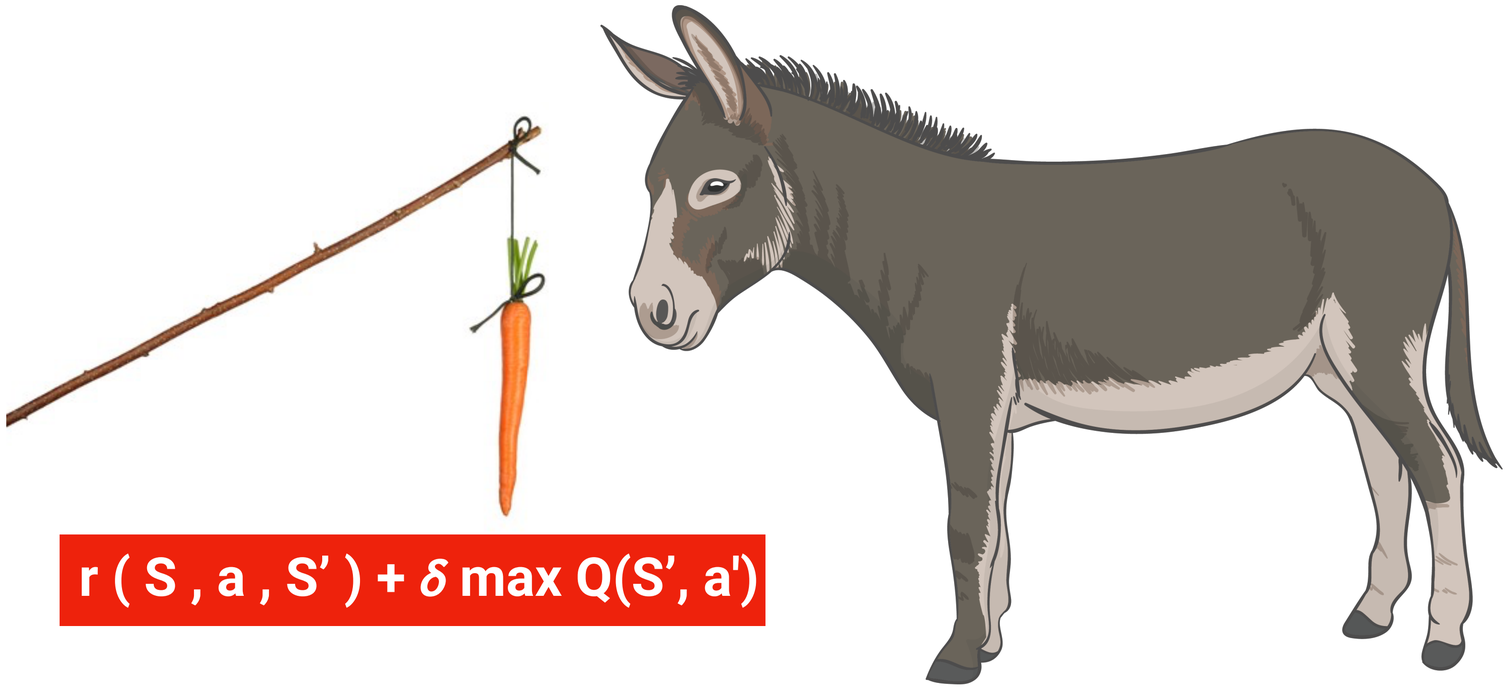


$$r(S, a, S') + \delta \max Q(S', a')$$

$$Q(S, a)$$



$$Q(S, a)$$



$$r(S, a, S') + \delta \max Q(S', a')$$

How are we going to update the Q-networks weights

We use a local network and a target network

1 - How to replace the Q-table ✅

2 - How to optimize the model ✅

3 - T-target

How are we going to update the Q-networks weights

We use a local network and a target network

The target neural network get slightly updated based on the local networks weights

1 - How to replace the Q-table ✅

2 - How to optimize the model ✅

3 - T-target

How are we going to update the Q-networks weights

We use a local network and a target network

The target neural network get slightly updated based on the local networks weights

The local network predict the Q-values for the current state

1 - How to replace the Q-table ✅

2 - How to optimize the model ✅

3 - T-target

How are we going to update the Q-networks weights

We use a local network and a target network

The target neural network get slightly updated based on the local networks weights

The local network predict the Q-values for the current state

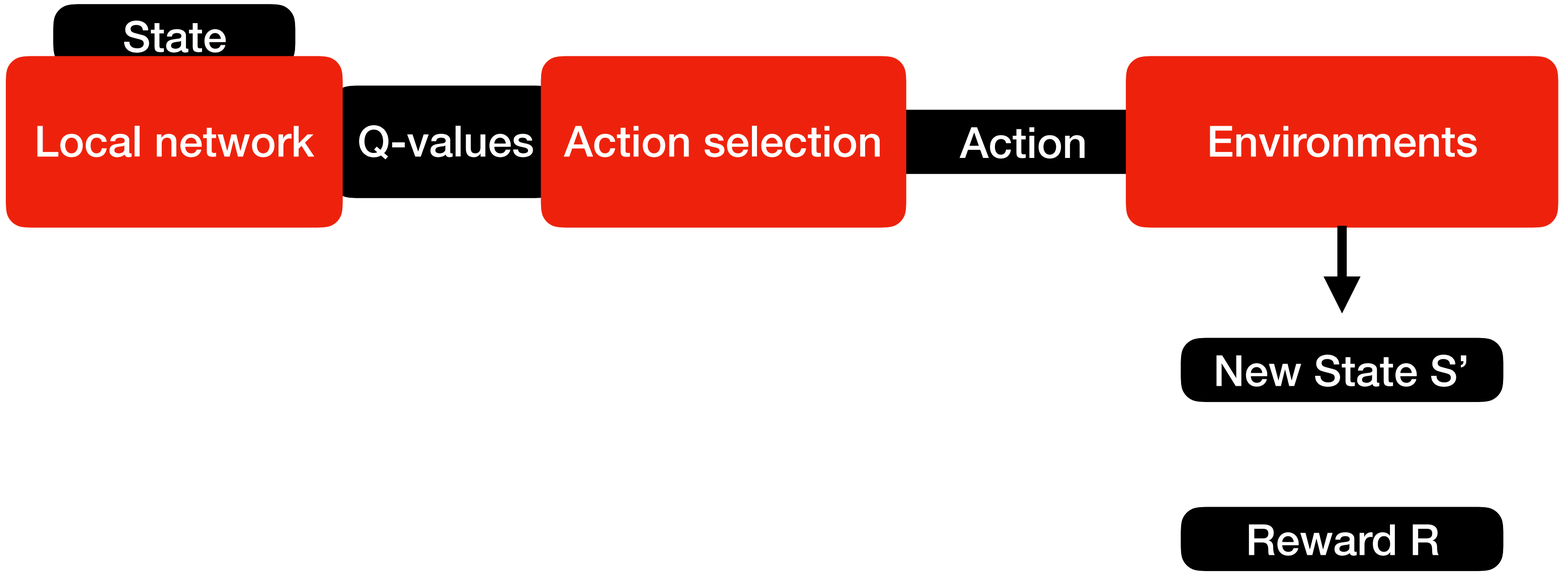
The target network predict the Q-values for the next state

1 - How to replace the Q-table ✅

2 - How to optimize the model ✅

3 - T-target

How are we going to update the Q-networks weights

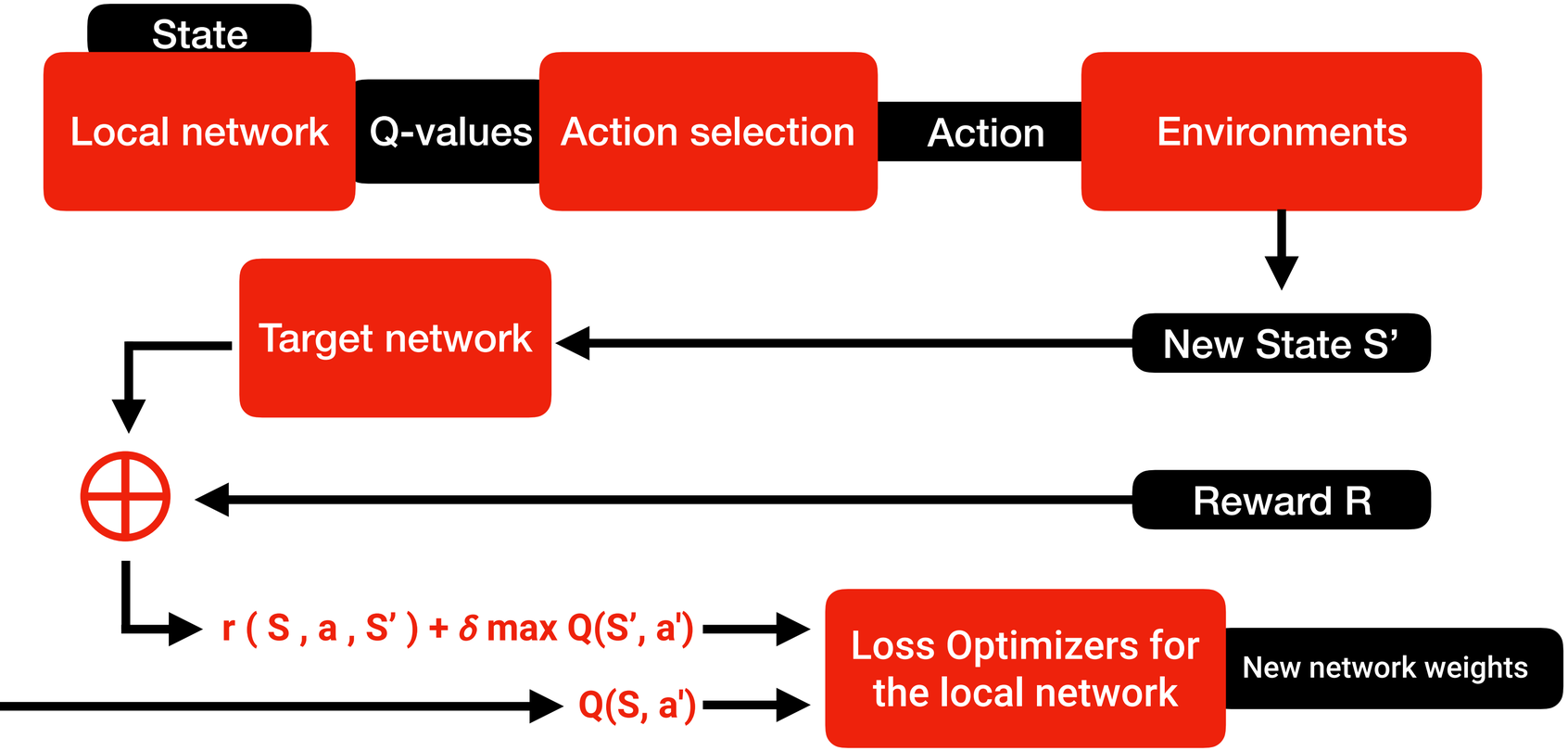


1 - How to replace the Q-table ✅

2 - How to optimize the model ✅

3 - T-target

How are we going to update the Q-networks weights



1 - How to replace the Q-table ✅

2 - How to optimize the model ✅

3 - T-target

Someone to answer the question please

In supervised learning, why we shuffling the dataset gives better accuracy?

Someone to answer the question please

In supervised learning, why we shuffling the dataset gives better accuracy?

If not shuffling data, the data can be sorted or similar data points will lie next to each other, which leads to slow convergence

Someone to answer the question please

In supervised learning, why we shuffling the dataset gives better accuracy?

If not shuffling data, the data can be sorted or similar data points will lie next to each other, which leads to slow convergence

In our case we are having a big correlation between consecutive states which may lead to inefficient learning .

How to break the coloration between states

Stack the experiences in a buffer

After the specified number of steps pick an N random experiences

Proceed with optimizing the loss with the learning batch

1 - How to replace the Q-table ✅

2 - How to optimize the model ✅

3 - T-target ✅

4 - Experience replay

How to break the coloration between states

Stack the experiences in a buffer

After the specified number of steps pick an N random experiences

Proceed with optimizing the loss with the learning batch

1 - How to replace the Q-table ✅

2 - How to optimize the model ✅

3 - T-target ✅

4 - Experience replay ✅

Putting it all together

Build a DQL agent

Saha ftourkoum