

# Reinforcement learning

Michal CHOVANEC, PhD.

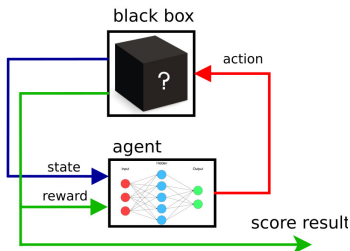
*March 2018*

Fakulta riadenia a informatiky

# Problem definition

- learn to play game with unknow rules
- input : state and reward
- output : action and total score

Agent never see required value (required action)



# Q-learning algorithm

$$Q'(s, a) = R(s, a) + \gamma \max_{a' \in A} Q(s', a')$$

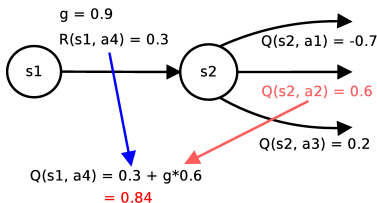
where

$Q(s, a)$  is previous state

$Q(s', a')$  is actual state

$R(s, a)$  is reward obtained in state  $s$  after executing action  $a$

$\gamma$  is discount factor  $\gamma \in \langle 0, 1 \rangle$



# SARSA algorithm

State Action Reward State Action

$$Q'(s, a) = (1 - \alpha)Q(s, a) + \alpha(R(s, a) + \gamma Q(s', a'))$$

where

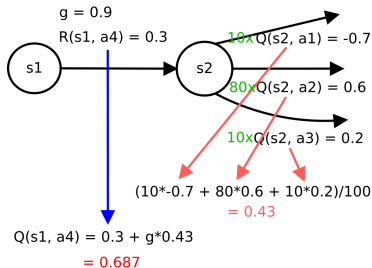
$Q(s, a)$  is previous state

$Q(s', a')$  is actual state

$R(s, a)$  is reward obtained in state  $s$  after executing action  $a$

$\gamma$  is discount factor  $\gamma \in \langle 0, 1 \rangle$

$\alpha$  is learning rate  $\alpha \in (0, 1)$



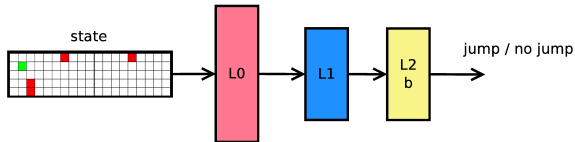
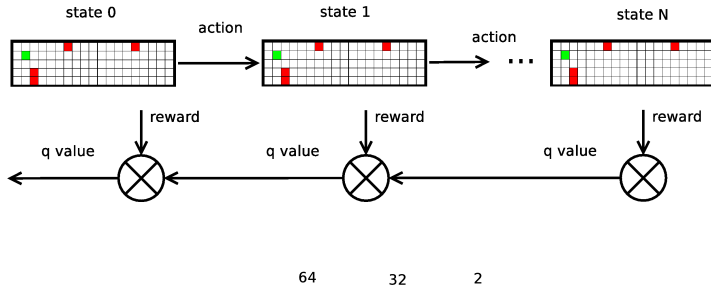
# Storing Q values

- table
- linear combination of basis function (handmade features)
- Kenerva's sparse encoding
- neural network

problems

- state correlations
- nonstationary Q values
- convergence to optimal strategy

# Neural network example - deep reinforcement learning





<https://github.com/michalnand/robotics>

[https://github.com/michalnand/machine\\_learning](https://github.com/michalnand/machine_learning)

[michal.nand@gmail.com](mailto:michal.nand@gmail.com)