Inteligentne Systemy Interaktywne

Piotr Duch

pduch@iis.p.lodz.pl
Instytut Informatyki Stosowanej
Politechnika Łódzka

Lato 2020

Plan wykładu

Informacje ogólne:

- Materiały wykładowe oraz laboratoryjne dostępne są na githubie (https://github.com/iis-siium/ISI).
- Literatura podstawowa:
  - Richard S. Sutton, and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
  - Csaba Szepesvári. *Algorithms for reinforcement learning*. Morgan and Claypool. 2009.
- Wykłady uzupełniające:
  - RL Course by David Silver - https://www.youtube.com
  - CS 188: Artificial Intelligence by Pieter Abbeel (wykład 10 i 11)- https://www.youtube.com/watch?v=IXuHxkpO5E8
- Materiały dodatkowe:
  - Practical RL Course by Yandex School of Data Analysis - https://github.com/yandexdataschool/Practical_RL
  - CS 188: Introduction to Artificial Intelligence by Berkeley University of California - https://inst.eecs.berkeley.edu/ cs188/fa19/project3/

Uczenie ze wzmocnieniem
Wprowadzenie

Uczenie pasywne

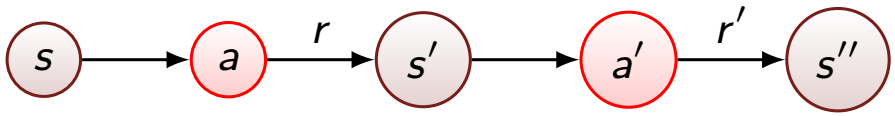
(ang. *model based learning*)

Uczenie aktywne

(ang. *model free learning*)

Uczenie aktywne

Co zrobić, jeżeli nie dysponujemy modelem środowiska?



Sekwencja:

- stany (s),
- akcje (a),
- nagrody (r).

Uczenie aktywne

Algorytmy:

- Monte Carlo.
- Metody różnic tymczasowych (ang. *Temporal Difference learning*):
  - Q-learning,
  - Sarsa.

Uczenie aktywne

Monte Carlo

Cechy algorytmu:

- Algorytm przeznaczony do zadań epizodycznych.
- Nie wymaga modelu środowiska.
- Uczy się na podstawie doświadczenie (ang. *experience*) - sekwencji stan, akcja, nagroda.

Uczenie aktywne

Monte Carlo

Cechy algorytmu:

- Algorytm przeznaczony do zadań epizodycznych.
- Nie wymaga modelu środowiska.
- Uczy się na podstawie doświadczenie (ang. *experience*) - sekwencji stan, akcja, nagroda.

Wersje algorytmu:

- Pierwsza wizyta (ang. *First-visit Monte Carlo*).
- Każda wizyta (ang. *Every-visit Monte Carlo*).

Uczenie aktywne

Monte Carlo

First-visit Monte Carlo method - oszacowanie $V \approx v_\pi$

Wejście: strategia π, która ma być oszacowana.
Inicjalizacja:

- $V(s) \in \mathbb{R}$ - losowe wartości, dla każdego $s \in S$,
- $Returns(s)$ - puste listy, dla każdego $s \in S$.

Nisekończona pętla (dla każdego epizodu):

Wygeneruj sekwencję przejść dla epizodu zgodnie ze strategią π:
$s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_{T-1}, a_{T-1}, r_T$
$G \leftarrow 0$:
Dla każdego kroku w epizodzie, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + r_{t+1}$
Jeżeli stan s_t nie pojawił się wcześniej:
Dodaj G do listy $Returns(s_t)$
$V(s_t) \leftarrow \text{average}(Returns(s_t))$

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Uczenie aktywne

Monte Carlo

First-visit Monte Carlo prediciton - for estimating $V \approx v_\pi$

Input: a policy π to be evaluated
Initialize:

- $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in S$,
- $Returns(s) \leftarrow$ an empty list, for all $s \in S$.

Loop forever (for each episode):
Generate an episode following π: $s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_{T-1}, a_{T-1}, r_T$
$G \leftarrow 0$:
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + r_{t+1}$
Unless s_t appears in $s_0, s_1, \ldots, s_{t+1}$:
Append G to $Returns(s_t)$
$V(s_t) \leftarrow$ average$(Returns(s_t))$

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Uczenie aktywne

Monte Carlo

Co się bardziej przyda:

- V(s),
- Q(s, a).

Uczenie aktywne

Monte Carlo

Metoda Monte Carlo, zmodyfikowana tak, aby wyznaczała $q_\pi(s, a)$ zamiast $v(s)$ będzie wyglądała analogicznie do tej, przedstawionej wcześniej.

Odwiedzony stan będzie określany za pomocą pary stan (s) - akcja wybrana w dany stanie (a).

Metoda *every-visit Monte Carlo* oszacuje wartość w danym stanie jako średnią oczekiwanych nagród ze wszystkich wizyt w danym stanie.

Metoda *first-visit Monte Carlo* oszacuje wartość w danym stanie jako nagrodę otrzymaną przy okazji pierwszej wizyty w danym stanie.

Uczenie aktywne

Monte Carlo

Jak rozwiązać problem nieodwiedzanych stanów:

- eksploracja stanów początkowych (ang. *exploring starts*):
  - wybieramy losowy stan i akcję, dla których rozpoczynamy epizod,
  - nierealistyczne w rzeczywistym świecie, za wyjątkiem symulacji,
- algorytm ϵ-zachłanny (ang. ϵ-*greedy*):
  - wybieramy najlepszą akcję z prawdopodobieństwem $1 - \epsilon + \frac{\epsilon}{|A(s)|}$,
  - wybieramy losową akcję z prawdopodobieństwem $\frac{\epsilon}{|A(s)|}$.

Uczenie aktywne

Monte Carlo

First-visit Monte Carlo method (for ϵ-soft policies) - oszacowanie $\pi \approx \pi_*$

Parametry algorytmu: mała wartość $\epsilon > 0$
Inicjalizacja:

- π losowa ϵ-miękka strategia,
- $Q(s, a) \in \mathbb{R}$ (losowe), dla każdej pary $s \in S$, $a \in A(s)$,
- $Returns(s, a) \leftarrow$ pusta lista, dla każdej pary $s \in S$, $a \in A(s)$.

Pętla nieskończona (dla każdego epizodu):
Wygeneruj sekwencję przejść dla epizodu zgodnie ze strategią π:
$s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_{T-1}, a_{T-1}, r_T$
$G \leftarrow 0$:
Dla każdego kroku w epizodzie, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + r_{t+1}$
Jeżeli para s_t, a_t niepojawiła się wcześniej w sekwencji $s_0, a_0, s_1, a_1, \ldots, s_{t+1}, a_{t+1}$:
Dodaj G do listy $Returns(s_t, a_t)$
$Q(S_t, A_t) \leftarrow \text{average}(Returns(s_t, a_t))$
$a^* \leftarrow \text{argmax}_a Q(s_t, a)$
Dla każdej akcji $a \in A(s_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A(s)|} & \text{if} \quad a = a^* \\ \frac{\epsilon}{|A(s)|} & \text{if} \quad a \neq a^* \end{cases} \tag{1}$$

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Uczenie aktywne

Monte Carlo

First-visit Monte Carlo method (for ϵ-soft policies) - estimates $\pi \approx \pi_*$

Algorithm parameter: small $\epsilon > 0$
Initialize:

- π an arbitrary ϵ-soft policy,
- $Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in S$, $a \in A(s)$,
- $Returns(s, a) \leftarrow$ an empty list, for all $s \in S$, $a \in A(s)$.

Loop forever (for each episode):
Generate an episode following π: $s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_{T-1}, a_{T-1}, r_T$
$G \leftarrow 0$:
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + r_{t+1}$
Unless the pair s_t, a_t appears in $s_0, a_0, s_1, a_1, \ldots, s_{t+1}, a_{t+1}$:
Append G to $Returns(s_t, a_t)$
$Q(s_t, a_t) \leftarrow \text{average}(Returns(s_t, a_t))$
$a^* \leftarrow \text{argmax}_a Q(s_t, a)$
For all $a \in A(s_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A(s)|} & \text{if} \quad a = a^* \\ \frac{\epsilon}{|A(s)|} & \text{if} \quad a \neq a^* \end{cases} \tag{2}$$

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Uczenie aktywne

Metody różnic tymczasowych (ang. *Temporal-Difference (TD) Learning*)

Metody różnic tymczasowych:

- Kombinacja metody Monte Carlo i Programowania Dynamicznego.
- Nie wymagają znajomości modelu środowiska.
- Uaktualnianie przewidywanych wartości następuje natychmiastowo - nie ma koniczeności oczekiwania na zakończenie epizodu.

Uczenie aktywne

Metody różnic tymczasowych (ang. *Temporal-Difference (TD) Learning*)

Szacowanie funkcji wartości za pomocą metod Monte Carlo:

$$V(s_t) \leftarrow V(s_t) + \alpha[G_t - V(s_t)] \tag{3}$$

Szacowanie funkcji wartości za pomocą metod różnic tymczasowych:

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \tag{4}$$

Uczenie aktywne

Metody różnic tymczasowych (ang. *Temporal-Difference (TD) Learning*)

Tabelaryczny algorytm różnic tymczasowych z krokiem 1 do oszacowania v_π

Wejście: strategia do oszacowania π
Parametr algorytmu: krok uczenia $\alpha \in (0, 1]$
Inicjalizacja tablicy wartości stanów $V(s)$ losowymi wartościami, za wyjątkiem stanu końcowego, któremu przypisana jest wartość 0.
Pętla dla każdego epizodu:
    Inicjalizacja s
    Dla każdego kroku w epizodzie:
        Wybierz ackję a zgodnie ze strategią π dla stanu s
        Wykonaj akcję a i zaobserwuj r oraz s'
        $V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$
        $s \leftarrow s'$
    Dopóki s nie jest stanem końcowym

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Uczenie aktywne

Metody różnic tymczasowych (ang. *Temporal-Difference (TD) Learning*)

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in S^+$, arbitrarily except that $V(terminal) = 0$
Loop for each episode:
    Initialize s
    Loop for each step of episode:
        $a \leftarrow$ action given by π for s
        Take action a, observe r, s'
        $V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$
        $s \leftarrow s'$
    Until s is not terminal

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Uczenie aktywne

Metody różnic tymczasowych (ang. *Temporal-Difference (TD) Learning*)

Błąd:

$$\delta_t \doteq r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{5}$$

Uczenie aktywne

Metody różnic tymczasowych (ang. *Temporal-Difference (TD) Learning*)

Metody różnic tymczasowych nie wymagają znajomości modelu środowiska.

Obliczenia wykonywane są online - brak konieczności oczekiwania na koniec epizodu.

Dla dowolnej stałej strategii π, udowodnione zostało, że metody TD(0) są zbieżne do v_π, w przypadku kiedy wartość parametru uczącego (α) jest stała i dostatecznie mała lub gdy wartość tego parametru zmniejsza się.

Uczenie aktywne

Q-Learning

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

Uczenie aktywne

Q-Learning

Wartość dla strategii π^* - optymalnej strategii

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \, max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

Uczenie aktywne

Q-Learning

Wartość dla strategii π^* - optymalnej strategii

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \, max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

Nagroda otrzymana po wykonaniu akcji a_t w stanie s_t

Uczenie aktywne

Q-Learning

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a)]$$

Uczenie aktywne

Q-Learning

Algorytm Q-Learning do wyznaczenia strategii $\pi \approx \pi_*$

Parametry algorytmu: krok uczenia $\alpha \in (0, 1]$, $\epsilon > 0$ o małej wartości
Inicjalizacja tablicy $Q(s, a)$, dla każdego stanu $s \in S$ i akcji w tym stanie $a \in A(s)$, losowymi wartościami oprócz stanu końcowego $Q(terminal, .) = 0$
Pętla po wszystkich epizodach:
    Inicjalizacja s
    Dla każdego kroku w epizodzie:
        Wybierz akcję a w stanie s wykorzystując strategię opartą o tablicę Q (np., ϵ-zachłanną)
        Wybierz akcję a i zaobserwuj r oraz s'
        $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$
        $s \leftarrow s'$
    Dopóki s nie jest stanem końcowym

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Uczenie aktywne

Q-Learning

Q-Learning for estimating $\pi \approx \pi_*$

Algorithm parameter: step size $\alpha \in (0, 1]$, small $\epsilon > 0$
Initialize $Q(s, a)$, for all $s \in S$, $a \in A(s)$, arbitrarily except that $Q(terminal, .) = 0$
Loop for each episode:
    Initialize s
    Loop for each step of episode:
        Choose a from s using policy derived from Q (e.g., ϵ-greedy)
        Take action a, observe r, s'
        $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$
        $s \leftarrow s'$
    Until s is not terminal

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

Uczenie aktywne

Q-Learning

Algorytm ϵ-zachłanny:

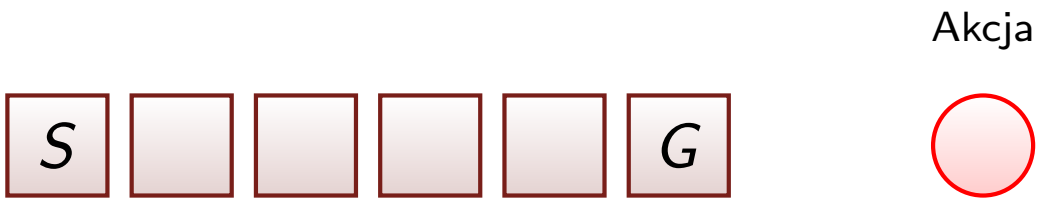
$$a = \begin{cases} \text{argmax}_a Q(s,.) & \text{z prawdopodobieństwem} \quad 1-\epsilon \ * \\ \text{losowa akcja} & \text{z prawdopodobieństwem} \quad \epsilon \end{cases} \tag{6}$$

* w przypadku kilku akcji z taką samą wartością należy wybierać **losową**

Uczenie aktywne

Q-Learning - przykład liczbowy

Nowe środowisko:



Aktualizowanie funkcji wartości dla pary stan-akcja (s_t, a_t):

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

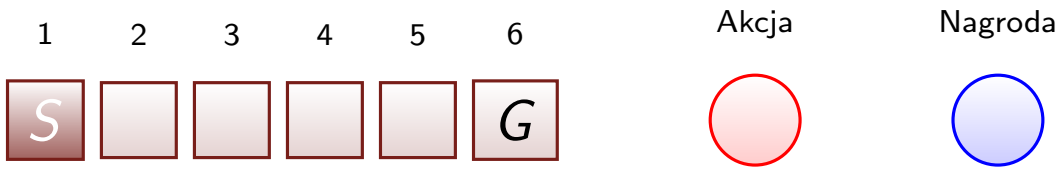
Parametry algorytmu:

- $\alpha = 0.1$,
- $\gamma = 0.9$,
- $r_G = 1$, w pozostałych przypadkach $r = 0$.

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 1:



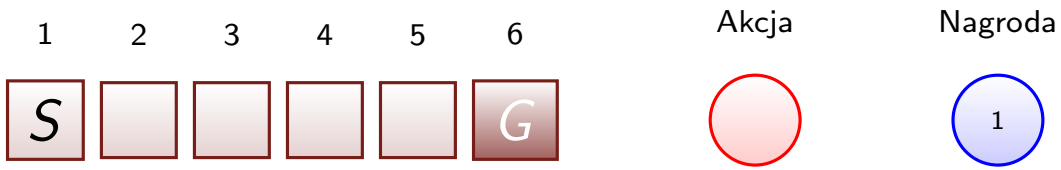
$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 1:



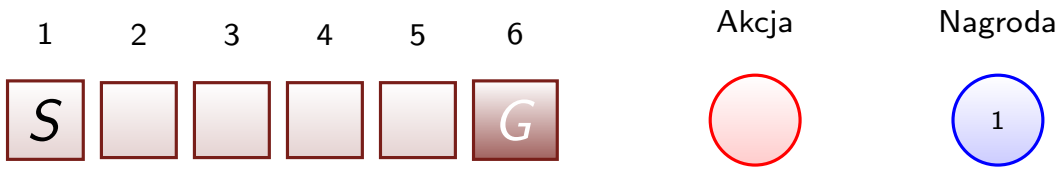
$$Q(5,P) = Q(5,P) + \alpha[r + \gamma max_a Q(6,a) - Q(5,P)]$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 1:



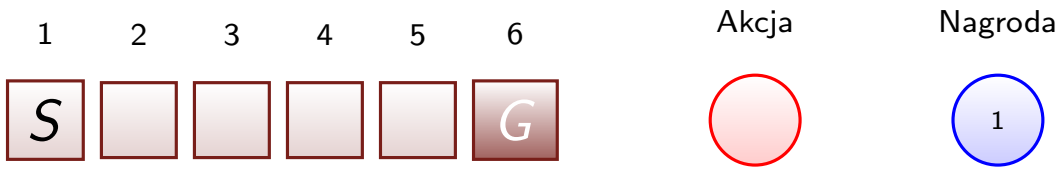
$$Q(5, P) = 0 + 0.1[1 + 0.9 * 0 - 0] = 0.1$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 1:



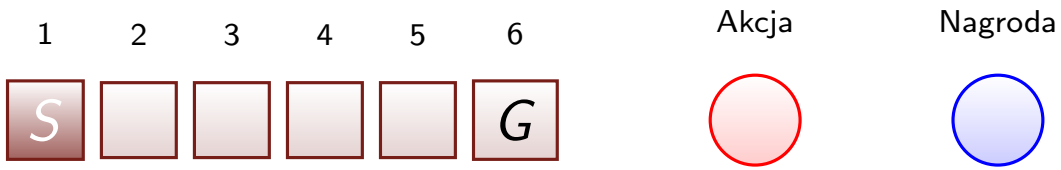
$$Q(5, P) = 0 + 0.1[1 + 0.9 * 0 - 0] = 0.1$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0.1
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 2:



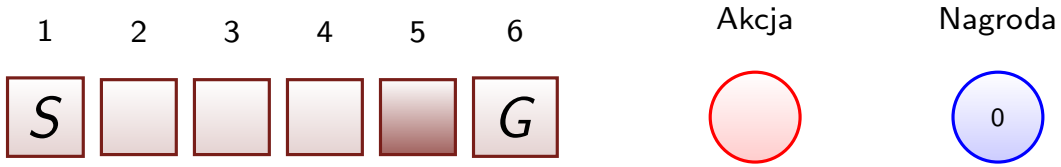
$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0.1
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 2:



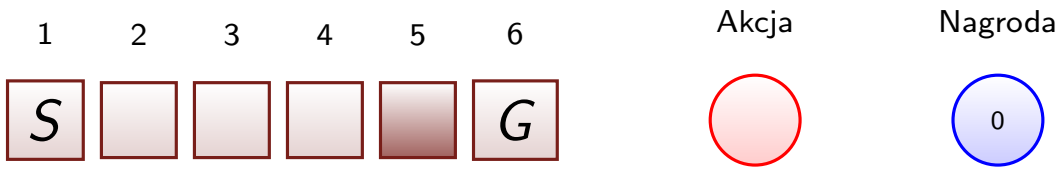
$$Q(4,P) = Q(4,P) + \alpha[r + \gamma max_a Q(5,a) - Q(4,P)]$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0.1
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 2:



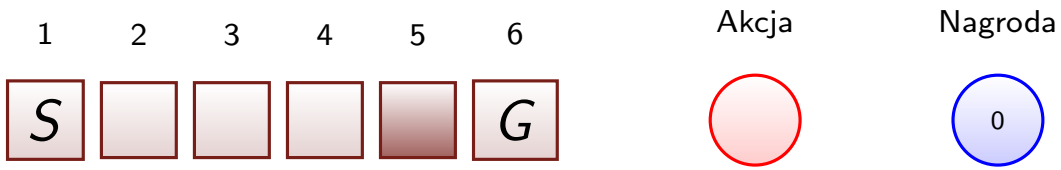
$$Q(4, P) = 0 + 0.1[0 + 0.9 * 0.1 - 0] = 0.009$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0.1
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 2:



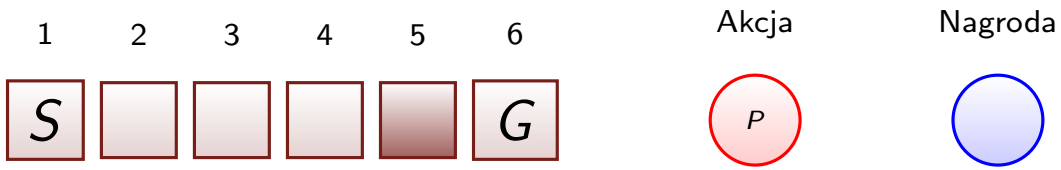
$$Q(4, P) = 0 + 0.1[0 + 0.9 * 0.1 - 0] = 0.009$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0.009
5	0	0.1
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 2:



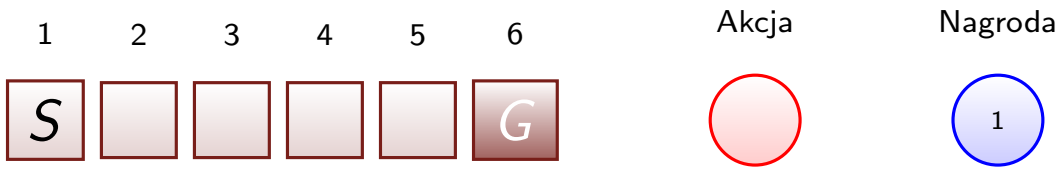
$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0.009
5	0	0.1
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 2:



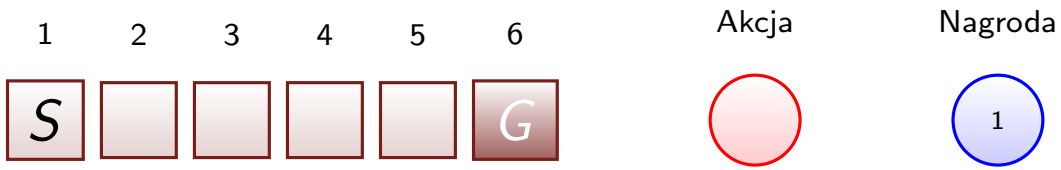
$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0.009
5	0	0.1
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 2:



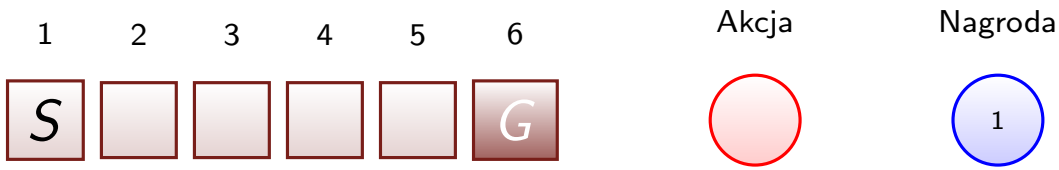
$$Q(5,P) = Q(5,P) + \alpha[r + \gamma max_a Q(6,a) - Q(5,P)]$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0.009
5	0	0.1
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 2:



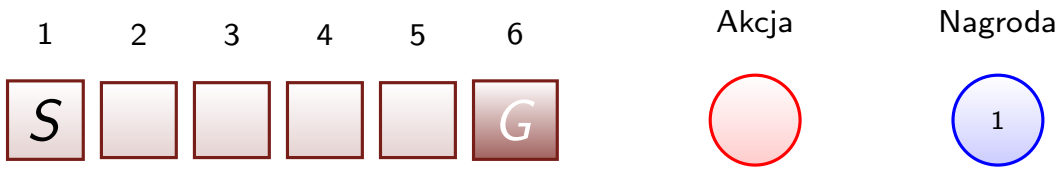
$$Q(5, P) = 0.1 + 0.1[1 + 0.9 * 0 - 0.1] = 0.19$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0.009
5	0	0.1
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.

Epizod 2:

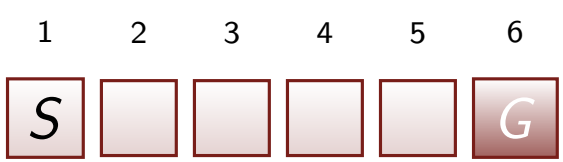


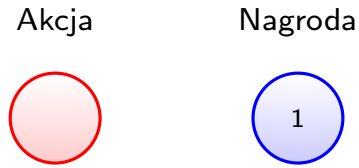
$$Q(5, P) = 0.1 + 0.1[1 + 0.9 * 0 - 0.1] = 0.19$$

Stan	L	P
1	0	0
2	0	0
3	0	0
4	0	0.009
5	0	0.19
6	0	0

Uczenie aktywne

Q-Learning - przykład liczbowy cd.





Epizod 3

Stan	L	P
1	0	0
2	0	0
3	0	0.00081
4	0	0.02520
5	0	0.27100
6	0	0

Epizod 4

Stan	L	P
1	0	0
2	0	0.00007
3	0	0.00300
4	0	0.04707
5	0	0.34390
6	0	0

Uczenie aktywne

Q-Learning

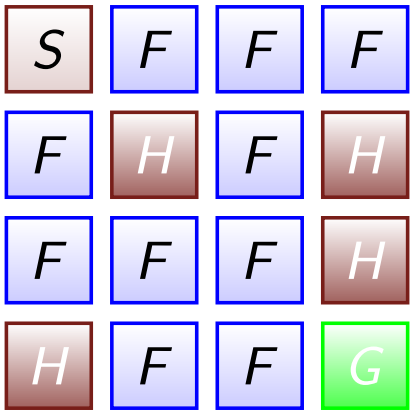
Cechy algorytmu Q-Learning:

- uczy się nie tylko na podstawie swojego doświadczenia, ale także innych ludzi / agentów,
- korzysta z optymalnej strategii nawet w trakcie eksploracji,
- korzysta z wielu strategii podążając tylko jedną.

Uczenie aktywne

Model środowiska

Frozen Lake:



Oznaczenia:

- S - stan początkowy,
- F - zamrożone pole,
- H - dziura (stan końcowy),
- G - cel (stan końcowy).

Nagrody:

- 1 - po dotarciu do pola G,
- 0 - w pozostałych przypadkach.

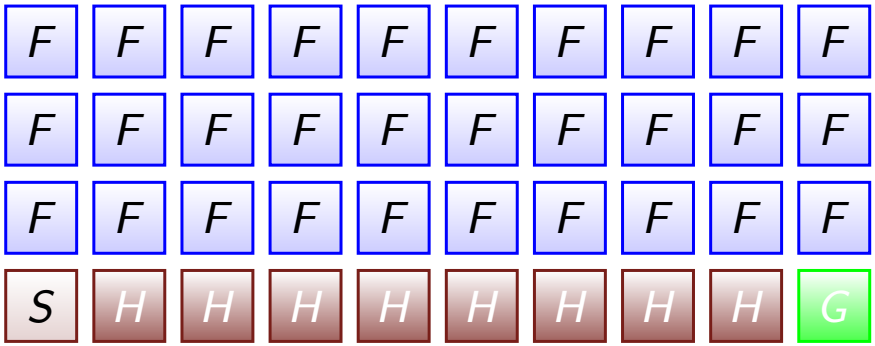
Akcje:

- lewo,
- prawo,
- góra,
- dół.

Uczenie aktywne

Model środowiska

Cliff World:



Akcje:

- lewo,
- prawo,
- góra,
- dół.

Oznaczenia:

- S - stan początkowy,
- F - wolne pole,
- H - dziura (stan końcowy),
- G - cel (stan końcowy).

Nagrody:

- 1 - po dotarciu do pola G,
- -100 - po dotarciu do pola H,
- -1 - w pozostałych przypadkach.

Uczenie aktywne

Materiały uzupełniające

- Książka *Reinforcement Learning: An Introduction*, Richard S. Sutton and Andrew G. Barto, wydanie drugie, 2018.
  - Rozdziały 5.1, 5.2, 5.3 i 5.4 - *Monte Carlo Methods*.
  - Rozdziały 6.1, 6.2, 6.3 i 6.5 - *TD Learning and Q-Learning*.
- Video *Artificial Intelligence Course by Pieter Abbeel - Lecture 10: Reinforcement Learning – od 0:38:00*.

Exploration vs. exploitation

Aproksymacja funkcji wartości stanu

Głębokie uczenie ze wzmocnieniem