AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Uczenie ze wzmacnianiem na przykładzie symulacji komputerowej

Bartłomiej Konieczny

Cel projektu

- Prosta symulacja graficzna
- Uczenie ze wzmocnieniem (reinforcement learning)
- Uczący się środowiska agent
- Środowisko wpływające na decyzje robota

Wykorzystane narzędzia

- Java 8 (środowisko IntelliJ Idea)
- LibGdx (symulacja graficzna)
- Project Lombok
- Gradle
- Apache POI(czytanie i zapisywanie do xml)
- Git (https://github.com/kserio/smartSimulation)

Napotkane problemy

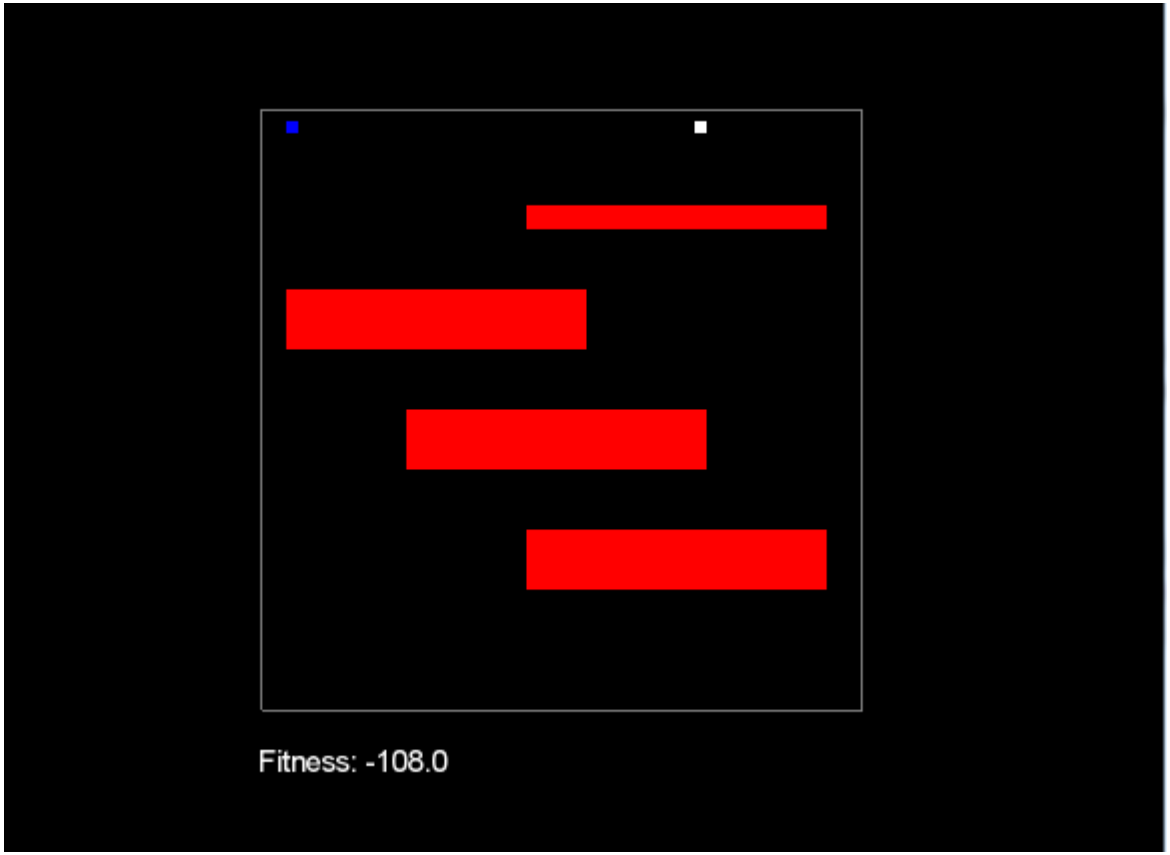
- Reprezentacja stanu
- Testowanie działania algorytmów
- Czas

Symulacja graficzna

Proste prezentacja robota, środowiska (przeszkoda, nagroda) w postaci figur geometrycznych

Zamknięta przestrzeń dwuwymiarowa, po której może się poruszać agent.

Symulacja graficzna



Opis symulacji

- Akcje: MOVE_UP, MOVE_DOWN, MOVE_RIGHT, MOVE_LEFT
- Reprezentacja stanu

  siatka 3x3

Uczenie ze wzmocnieniem

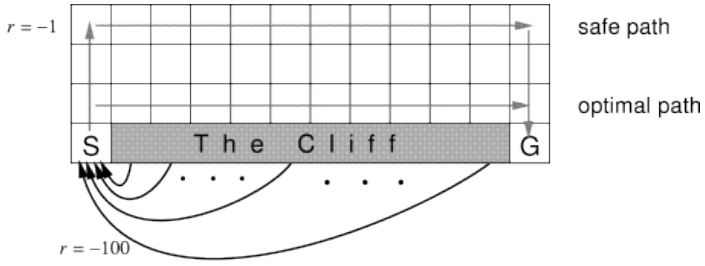
- Dział uczenia maszynowego
- Brak przykładów trenujących
- Nauka na przez interakcję z otoczeniem
- Metoda prób i błędów
- Natychmiastowa nagroda (akcję wpływają na otrzymaną nagrodę)
- Opóźniona nagroda (akcję wpływają również na następne sytuacje i przez to na kolejne nagrody)

Algorytmy

Q-learning(off-policy)

SARSA(on-policy)

Różnice


r = −1
safe path
optimal path
S
The Cliff
G
r = −100


Sarsa
Q-learning
Reward per epsiode
−25
−50
−75
−100
0
100
200
300
400
500
Episodes

Q-learning

Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    Choose a from s using policy derived from Q (e.g., ε-greedy)
    Take action a, observe r, s'
    $Q(s, a) \leftarrow Q(s, a) + \alpha\left[r + \gamma \max_{a'} Q(s', a') - Q(s, a)\right]$
    $s \leftarrow s'$;
  until s is terminal

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

SARSA

Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
  Initialize s
  Choose a from s using policy derived from Q (e.g., ε-greedy)
  Repeat (for each step of episode):
    Take action a, observe r, s'
    Choose a' from s' using policy derived from Q (e.g., ε-greedy)
    $Q(s, a) \leftarrow Q(s, a) + \alpha\big[r + \gamma Q(s', a') - Q(s, a)\big]$
    $s \leftarrow s'$; $a \leftarrow a'$;
  until s is terminal

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Big[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\Big].$$

Bibliografia

- **Richard S. Sutton and Andrew G., "**A Bradford Book**",** The MIT Press
  Cambridge, Massachusetts**,** London, England,
  https://webdocs.cs.ualberta.ca/~sutton/book/ebook/the-book.html
- **Odalric-Ambrym Maillard, Remi Munos, Daniil Ryabko,** „Selecting the State-Representation in Reinforcement Learning”,
  http://papers.nips.cc/paper/4415-selecting-the-state-representation-in-reinforcement-learning.
- **David Poole, Alan Mackworth,** „Artificial Intelligence: Foundations of Computational Agents”, Cambridge University Press, 2010.
- **Lucas Jenß,** „An application of SARSA temporal difference learning to Super Mario”,
  Hamburg University of Applied Sciences
- https://en.wikipedia.org/wiki/Reinforcement_learning
- http://osilek.mimuw.edu.pl/index.php?title=Sztuczna_inteligencja/SI_Modu%C5%82_13_-_Ucz
- https://studywolf.wordpress.com/2013/07/01/reinforcement-learning-sarsa-vs-q-learning/