Bootcamp Lvl Up

Uczenie ze wzmocnieniem v2

Jakub Łyskawa

January 8, 2022

Setup środowiska

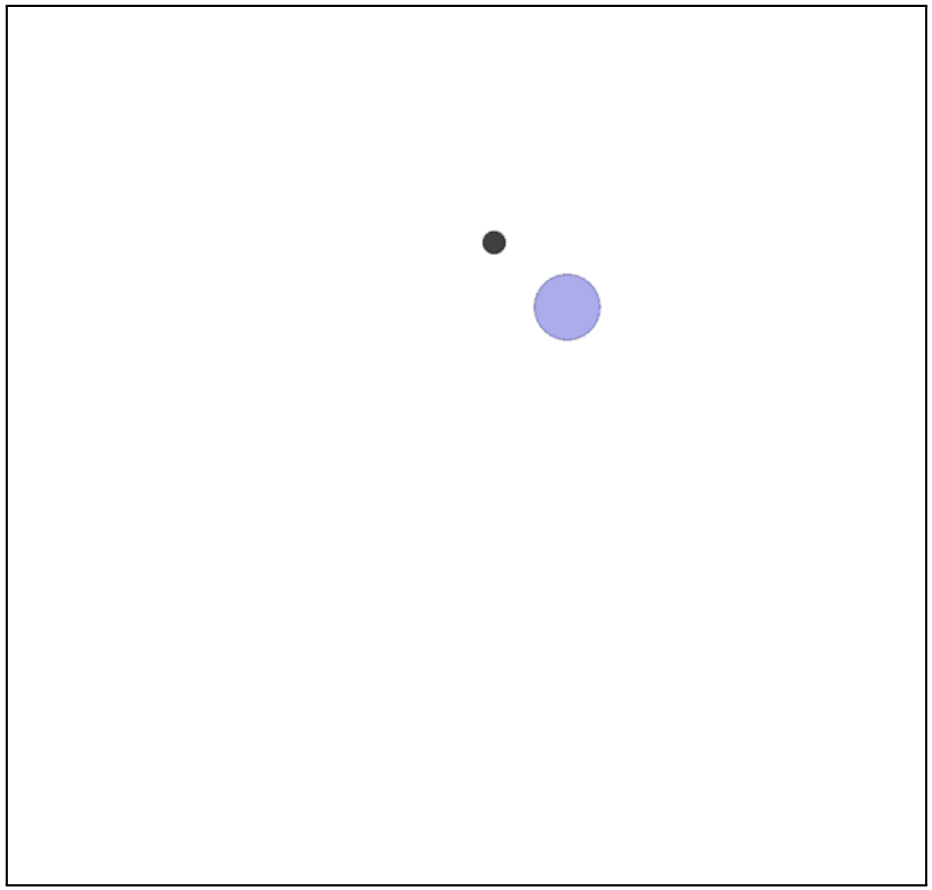
Pakiety Python:

- numpy
- tensorflow
- pyglet
- gym
- pettingzoo[mpe]

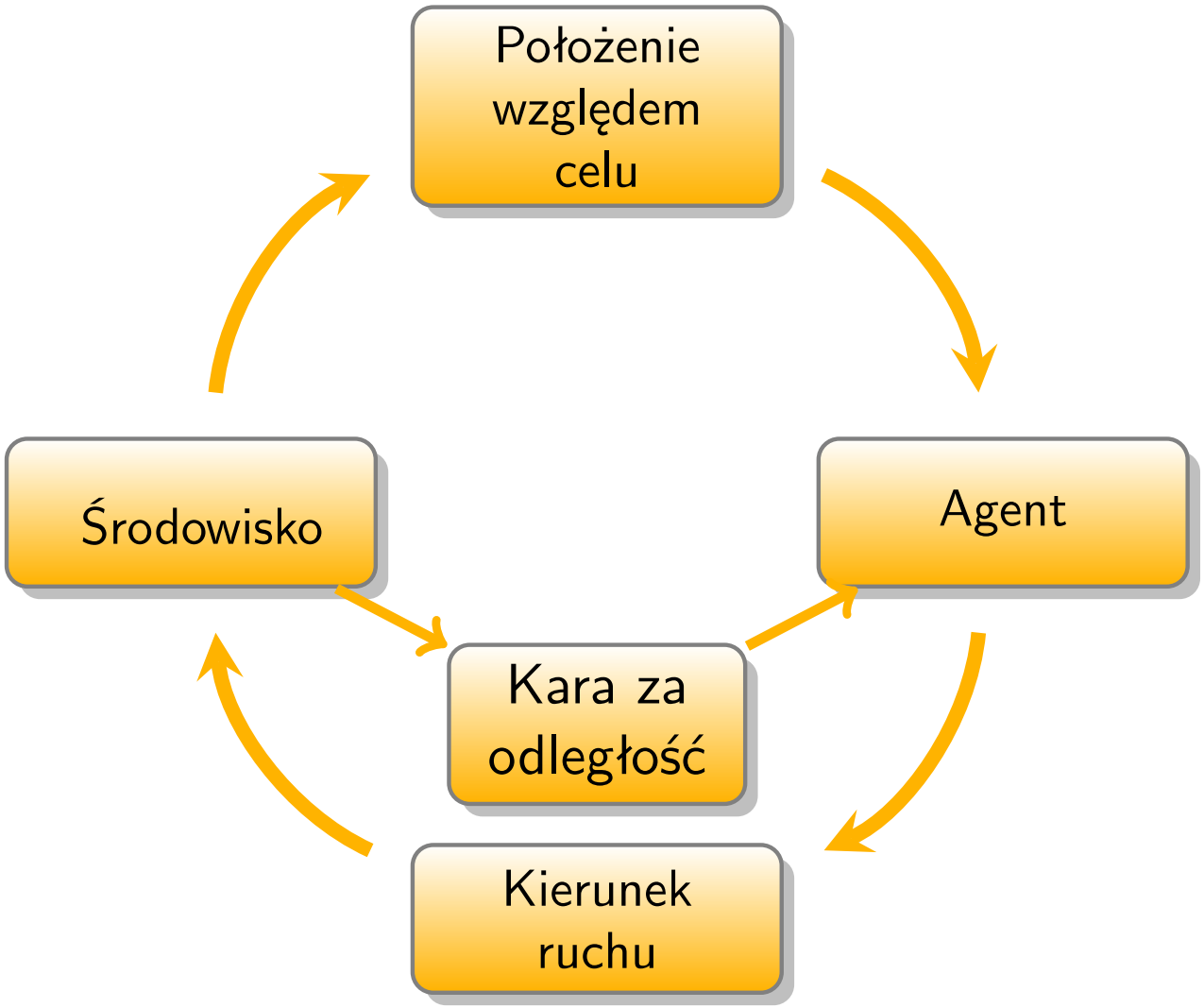
Uczenie ze wzmocnieniem: krótkie przypomnienie

Proces decyzyjny Markowa

Proces decyzyjny Markowa - przykład



Proces decyzyjny Markowa - przykład



Uczenie ze wzmocnieniem

Szukanie polityki decyzyjnej maksymalizującej sumę nagród otrzymywanych przez agenta.

Oznaczenia

s_t

Stan/obserwacja w chwili t.

a_t

Akcja w chwili t.

r_t

Nagroda w chwili t.

π

Polityka decyzyjna - rozkład prawdopodobieństwa akcji

Ważne pojęcia

Zdyskontowana suma nagród

$R_t = \sum_{i=0} \gamma^i r_{t+i}, \gamma \in (0, 1]$

Funkcja wartości

$V^\pi(s) = \mathbb{E}[R_t | s_t = s, \pi]$

Funkcja akcji-wartości

$Q^\pi(s, a) = \mathbb{E}[R_t | s_t = s, a_t = a, \pi]$

Funkcja przewagi

$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s, a)$

Wybrane rodzaje algorytmów

Q-Learning

- Jeden model - funkcja Q
- Polityka - akcja o największej oczekiwanej zdyskontowanej sumie nagród
- Uczenie - uczenie estymacji funkcji Q
- Tylko dla dyskretnych przestrzeni akcji

Wybrane rodzaje algorytmów

Q-Learning

- Jeden model - funkcja Q
- Polityka - akcja o największej oczekiwanej zdyskontowanej sumie nagród
- Uczenie - uczenie estymacji funkcji Q
- Tylko dla dyskretnych przestrzeni akcji

Aktor-krytyk

- Dwa modele - aktor i krytyk
- Aktor - określa politykę
- Krytyk - estymuje zdyskontowaną sumę nagród
- Aktor uczony na podstawie krytyka

Przydatne mechanizmy

Powatrzanie doświadczenia

Dane zebrane w poprzednich krokach są zapamietywane i powtarzane.
Może wymagać uwzględnienia, że zmienia się prawdopodobieństwo otrzymania takiej próbki.

Przydatne mechanizmy

Powatrzanie doświadczenia

Dane zebrane w poprzednich krokach są zapamietywane i powtarzane.
Może wymagać uwzględnienia, że zmienia się prawdopodobieństwo otrzymania takiej próbki.

ϵ-greedy exploration

Z prawdopodobieństwem ϵ akcja jest losowana z rozkładu jednostajnego z całej przestrzeni, a z $(1 - \epsilon)$ jest brana najlepsza

Advantage Actor-Critic - nowy algorytm

Struktura

Aktor

- Określa rozkład prawdopodobieństwa akcji w stanie

Krytyk

- Estymator funkcji wartości

Struktura

Aktor

- Określa rozkład prawdopodobieństwa akcji w stanie
- Nauka - minimalizacja
  $L_\pi(t) = -\log \pi(a_t|s_t)A(s_t, a_t)$

Krytyk

- Estymator funkcji wartości
- Nauka - minimalizacja:
  $L_V(t) = (r_t + \gamma V(s_{t+1}) - V(s_t))^2$

Struktura

Aktor

- Określa rozkład prawdopodobieństwa akcji w stanie
- Nauka - minimalizacja $L_\pi(t) = -\log \pi(a_t|s_t)A(s_t, a_t)$
- Funkcja przewagi $A(s_t, a_t)$ estymowana przez $r_t + \gamma V(s_{t+1}) - V(s_t)$

Krytyk

- Estymator funkcji wartości
- Nauka - minimalizacja: $L_V(t) = (r_t + \gamma V(s_{t+1}) - V(s_t))^2$

Zbieranie danych

Agent wykonuje do T kroków w środowisku między kolejnymi krokami uczenia.

W każdym kroku uczenia podaje się zebrane dane jako batch.

Zbieranie danych

Agent wykonuje do T kroków w środowisku między kolejnymi krokami uczenia.

W każdym kroku uczenia podaje się zebrane dane jako batch.

Asynchronous Advantage Actor-Critic (A3C)

- Wiele aktorów i krytyków niezależnie zbierających dane
- Regularna synchronizacja parametrów sieci

Zbieranie danych

Agent wykonuje do T kroków w środowisku między kolejnymi krokami uczenia.

W każdym kroku uczenia podaje się zebrane dane jako batch.

Asynchronous Advantage Actor-Critic (A3C)

- Wiele aktorów i krytyków niezależnie zbierających dane
- Regularna synchronizacja parametrów sieci

Synchronous Advantage Actor-Critic (A2C)

- Dane zbierane na wielu środowiskach równocześnie
- Łączone w pojedynczy batch

Aktor-krytyk dla dyskretnej przestrzeni akcji

- Krytyk - jak zwykle

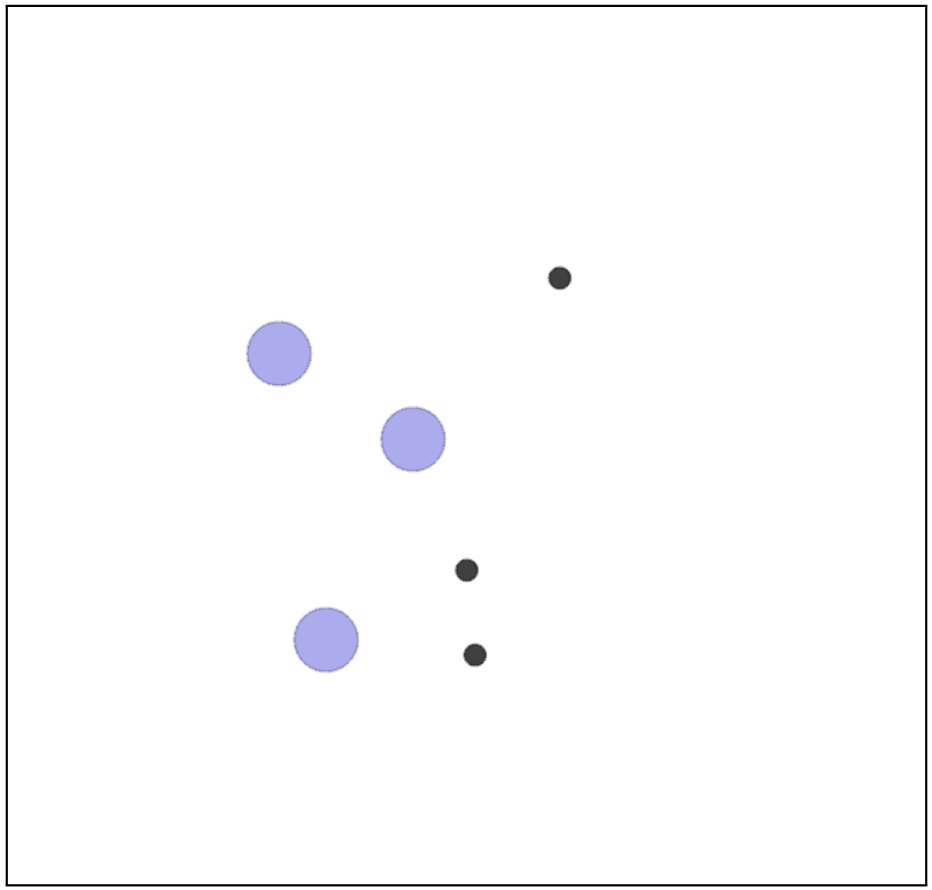
Aktor-krytyk dla dyskretnej przestrzeni akcji

- Krytyk - jak zwykle
- Aktor - sieć neuronowa z softmaxem na wyjściu

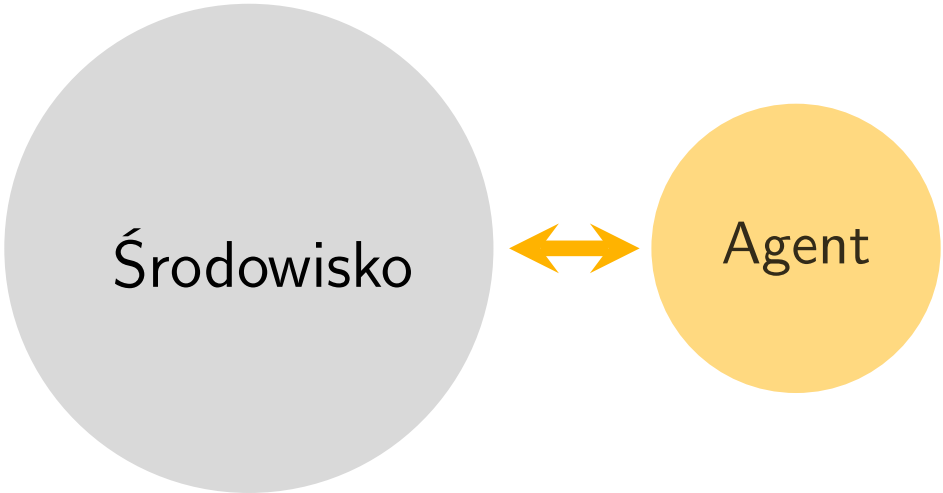
Coding time!

MARL

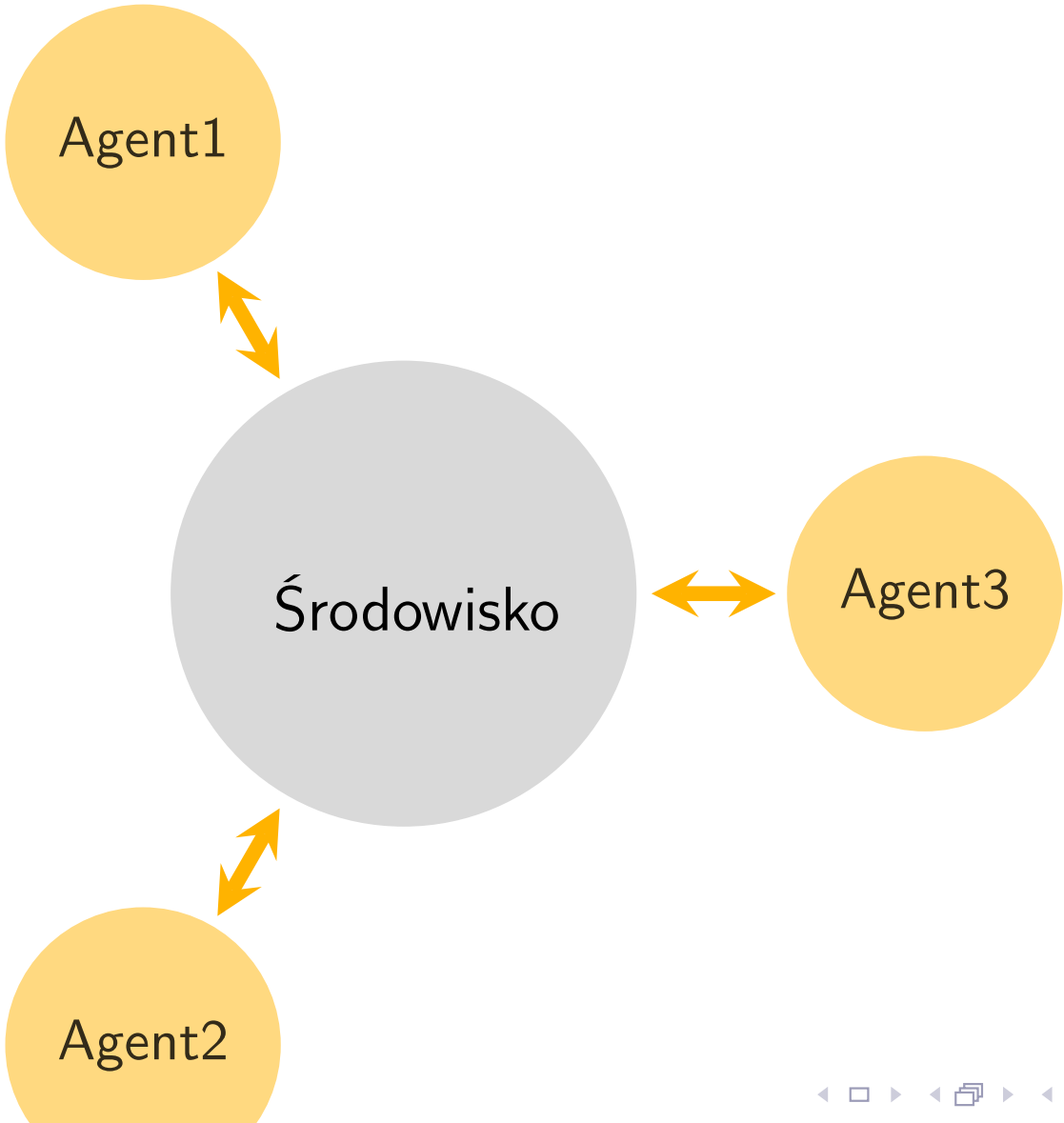
Bardziej złożone środowisko



Proces decyzyjny Markowa



Wieloagentowy proces decyzyjny Markowa



Wieloagentowy proces decyzyjny Markowa

- Wiele agentów, każdy wykonuje akcję, każdy ma swoją obserwację, każdy ma swoją nagrodę

Wieloagentowy proces decyzyjny Markowa

- Wiele agentów, każdy wykonuje akcję, każdy ma swoją obserwację, każdy ma swoją nagrodę
- Cel - maksymalizacja sumy nagród dla (podzbioru) agentów

Wieloagentowy proces decyzyjny Markowa

- Wiele agentów, każdy wykonuje akcję, każdy ma swoją obserwację, każdy ma swoją nagrodę
- Cel - maksymalizacja sumy nagród dla (podzbioru) agentów
- Agenci mogą współpracować lub konkurować

Wieloagentowy proces decyzyjny Markowa

- Wiele agentów, każdy wykonuje akcję, każdy ma swoją obserwację, każdy ma swoją nagrodę
- Cel - maksymalizacja sumy nagród dla (podzbioru) agentów
- Agenci mogą współpracować lub konkurować
- Liczba agentów może się zmieniać

Wieloagentowy proces decyzyjny Markowa

- Wiele agentów, każdy wykonuje akcję, każdy ma swoją obserwację, każdy ma swoją nagrodę
- Cel - maksymalizacja sumy nagród dla (podzbioru) agentów
- Agenci mogą współpracować lub konkurować
- Liczba agentów może się zmieniać
- Może być ograniczona możliwość komunikacji między agentami podczas uczenia oraz podczas działania

Podejście pierwsze

Każdy agent trenowany niezależnie - Independent A2C

Coding time!

Podejście pierwsze

Każdy agent trenowany niezależnie

Problemy:

- Zmienia się otoczenie w którym agent działa
- Pomija wpływ innych agentów na otrzymane wyniki

Podejście drugie

Agenci trenowani razem, maksymalizując sumaryczną nagrodę

Podejście drugie

Agenci trenowani razem, maksymalizując sumaryczną nagrodę

Czy osobne modele polityki?

Podejście drugie

Agenci trenowani razem, maksymalizując sumaryczną nagrodę

Czy osobne modele polityki?

Centralized A2C

Coding time!

Podejście drugie

Agenci trenowani razem

Problemy:

- Ciężko określić wpływ akcji agenta na nagrodę
- Pojedynczy gorsi agenci mogą "ciągnąć w dół" wszystkich

Multiagent Advantage Actor-Critic

- Każdy agent ma swojego aktora i krytyka

Multiagent Advantage Actor-Critic

- Każdy agent ma swojego aktora i krytyka
- Parametr $\alpha \in [0, 1]$ określa wpływ innych agentów

Multiagent Advantage Actor-Critic

- Każdy agent ma swojego aktora i krytyka
- Parametr $\alpha \in [0, 1]$ określa wpływ innych agentów
- Maksymalizacja sumy nagród $\tilde{r}_t^i = r_t^i + \sum_{j \neq i} \alpha r_t^j$

Multiagent Advantage Actor-Critic

- Każdy agent ma swojego aktora i krytyka
- Parametr $\alpha \in [0, 1]$ określa wpływ innych agentów
- Maksymalizacja sumy nagród $\tilde{r}_t^i = r_t^i + \sum_{j \neq i} \alpha r_t^j$
- Wejście krytyka: $\tilde{s}_t^i = [s_t^i] \cup \alpha [s_t^j]_{j \neq i}$

Multiagent Advantage Actor-Critic

- Każdy agent ma swojego aktora i krytyka
- Parametr $\alpha \in [0, 1]$ określa wpływ innych agentów
- Maksymalizacja sumy nagród $\tilde{r}_t^i = r_t^i + \sum_{j \neq i} \alpha r_t^j$
- Wejście krytyka: $\tilde{s}_t^i = [s_t^i] \cup \alpha [s_t^j]_{j \neq i}$
- Można ograniczyć widoczność agentów

Multiagent Advantage Actor-Critic

- Każdy agent ma swojego aktora i krytyka
- Parametr $\alpha \in [0, 1]$ określa wpływ innych agentów
- Maksymalizacja sumy nagród $\tilde{r}_t^i = r_t^i + \sum_{j \neq i} \alpha r_t^j$
- Wejście krytyka: $\tilde{s}_t^i = [s_t^i] \cup \alpha [s_t^j]_{j \neq i}$
- Można ograniczyć widoczność agentów
- Do straty aktora dodany bonus za entropię: $\sum_{a \in A^i} \pi^i \log \pi^i(a|s_t^i)$

Coding time!

Inne podejścia

- Niezależne agenty
- Scentralizowane uczenie
- Faktoryzacja funkcji wartości

Inne podejścia

- Niezależne agenty
- Scentralizowane uczenie
- Faktoryzacja funkcji wartości
- Konsensus

Inne podejścia

- Niezależne agenty
- Scentralizowane uczenie
- Faktoryzacja funkcji wartości
- Konsensus
- Nauka komunikacji